

**GigaDevice Semiconductor Inc.**

**GD32F1x0  
ARM® Cortex™-M3 32-bit MCU**

**用户手册**

**第 2 版**

**(2016 年 1 月)**

## 目 录

目 录 .....	1
图 .....	15
表 .....	22
<b>1. 系统及存储器架构 .....</b>	<b>1</b>
<b>1.1. ARM Cortex-M3 处理器 .....</b>	<b>1</b>
<b>1.2. 系统架构 .....</b>	<b>2</b>
<b>1.3. 存储器映射 .....</b>	<b>5</b>
1.3.1. 位带操作 .....	8
1.3.2. 片上 SRAM .....	8
1.3.3. 片上闪存 .....	8
<b>1.4. 引导配置 .....</b>	<b>9</b>
<b>1.5. 系统配置寄存器 (SYSCFG) .....</b>	<b>10</b>
1.5.1. 系统配置寄存器 1 (SYSCFG_R1) .....	10
1.5.2. 系统配置寄存器 2 (SYSCFG_R2) .....	11
1.5.3. EXTI 源选择寄存器 1 (SYSCFG_EXTI_S1) .....	12
1.5.4. EXTI 源选择寄存器 2 (SYSCFG_EXTI_S2) .....	13
1.5.5. EXTI 源选择寄存器 3 (SYSCFG_EXTI_S3) .....	14
1.5.6. EXTI 源选择寄存器 4 (SYSCFG_EXTI_S4) .....	15
1.5.7. 系统配置寄存器 3 (SYSCFG_R3) .....	17
<b>1.6. 设备电子签名 .....</b>	<b>18</b>
1.6.1. 存储密度信息 .....	18
1.6.2. 设备唯一 ID (96 位/位域) .....	18
<b>2. 电源控制 (PWR) .....</b>	<b>20</b>
<b>2.1. 简介 .....</b>	<b>20</b>
<b>2.2. 主要特性 .....</b>	<b>20</b>
<b>2.3. 功能描述 .....</b>	<b>20</b>
2.3.1. 电池备份域 .....	22
2.3.2. V <sub>DD</sub> /V <sub>DDA</sub> 电源域 .....	23
2.3.3. 1.2V 电源域 (适用于 GD32F130xx 和 GD32F150xx 产品) .....	24
2.3.4. 1.8V 电源域 (适用于 GD32F170xx 和 GD32F190xx 产品) .....	24
2.3.5. 省电模式 .....	25
<b>2.4. PWR 寄存器 .....</b>	<b>26</b>
2.4.1. 电源控制寄存器 (PWR_CTLR) .....	26
2.4.2. 电源状态寄存器 (PWR_STR) .....	29
<b>3. 闪存存储器控制器 (FMC) .....</b>	<b>31</b>

<b>3.1. 简介</b>	<b>31</b>
<b>3.2. 主要特性</b>	<b>31</b>
<b>3.3. 功能描述</b>	<b>32</b>
3.3.1. 闪存存储器结构	32
3.3.2. 读保护	33
3.3.3. FMC_CMR 寄存器解锁	33
3.3.4. 页擦除	33
3.3.5. 整片擦除	34
3.3.6. 主存储块编程	35
3.3.7. 可选字节块擦除	36
3.3.8. 可选字节块编程	37
3.3.9. 可选字节块说明	37
3.3.10. 页擦除/编程保护	38
3.3.11. 安全保护	39
<b>3.4. FMC 寄存器</b>	<b>39</b>
3.4.1. 闪存保留寄存器 (FMC_RESR)	39
3.4.2. 闪存解锁寄存器 (FMC_UKEYR)	40
3.4.3. 闪存可选字节块解锁寄存器 (FMC_OBKEYR)	40
3.4.4. 闪存控制/状态寄存器 (FMC_CSR)	41
3.4.5. 闪存命令寄存器 (FMC_CMRR)	41
3.4.6. 闪存命令地址寄存器 (FMC_AR)	43
3.4.7. 闪存可选字节寄存器 (FMC_OPTR)	43
3.4.8. 闪存页擦除/编程保护寄存器 (FMC_WPR)	44
3.4.9. 闪存等待状态控制寄存器 (FMC_WSCR)	44
3.4.10. 闪存产品保留 ID 寄存器 1 (FMC_RES_ID1)	46
3.4.11. 闪存产品保留 ID 寄存器 2 (FMC_RES_ID2)	46
<b>4. 复位和时钟控制单元 (RCC)</b>	<b>47</b>
<b>4.1. 复位控制单元 (RCU)</b>	<b>47</b>
4.1.1. 简介	47
4.1.2. 功能描述	47
<b>4.2. 时钟控制单元 (CCU)</b>	<b>48</b>
4.2.1. 简介	48
4.2.2. 主要特性	51
4.2.3. 功能描述	51
<b>4.3. RCC 寄存器</b>	<b>55</b>
4.3.1. 总体时钟控制寄存器 (RCC_GCCR)	55
4.3.2. 总体时钟配置寄存器 (RCC_GCFGR)	56
4.3.3. 总体时钟中断寄存器 (RCC_GCIR)	63
4.3.4. APB2 复位控制寄存器 (RCC_APB2RCR)	68
4.3.5. APB1 复位控制寄存器 (RCC_APB1RCR)	70
4.3.6. AHB 时钟控制寄存器 (RCC_AHBCCR)	74

4.3.7.	APB2 时钟控制寄存器 (RCC_APB2CCR) .....	76
4.3.8.	APB1 时钟控制寄存器 (RCC_APB1CCR) .....	78
4.3.9.	备份域控制寄存器 (RCC_BDCR) .....	82
4.3.10.	总体控制/状态寄存器 (RCC_GCSR) .....	85
4.3.11.	AHB 复位控制寄存器 (RCC_AHBRCR) .....	87
4.3.12.	总体时钟配置寄存器 2 (RCC_GCFGR2) .....	88
4.3.13.	总体时钟配置寄存器 3 (RCC_GCFGR3) .....	89
4.3.14.	总体时钟控制寄存器 2 (RCC_GCCR2) .....	91
4.3.15.	总体时钟配置寄存器 4 (RCC_GCFGR4) (仅适用于 GD32F170xx 和 GD32F190xx 产品) .....	92
4.3.16.	附加时钟控制寄存器 (RCC_ACCR) .....	93
4.3.17.	附加复位控制寄存器 (RCC_ARCR) .....	93
4.3.18.	电源解锁寄存器 (RCC_VC_KEY) .....	94
4.3.19.	RCC 深度睡眠模式电压寄存器 (RCC_DEEPSLEEP_VC) .....	94
4.3.20.	RCC 掉电电压选择寄存器 (RCC_PDR_S) (仅适用于 GD32F130xx 和 GD32F150xx 产品) .....	95
<b>5.</b>	<b>通用和备用功能 I/Os (GPIO 和 AFIOs) .....</b>	<b>97</b>
5.1.	简介 .....	97
5.2.	主要特性 .....	97
5.3.	功能描述 .....	97
5.3.1.	GPIO 管脚配置 .....	99
5.3.2.	备用功能 (AF) .....	99
5.3.3.	附加功能 .....	99
5.3.4.	输入配置 .....	99
5.3.5.	输出配置 .....	100
5.3.6.	模拟配置 .....	101
5.3.7.	备用功能(AF) 配置 .....	101
5.3.8.	GPIO 锁定功能 .....	102
<b>5.4.</b>	<b>GPIO 寄存器 .....</b>	<b>102</b>
5.4.1.	GPIO 端口控制寄存器 (GPIOx_CTLR) (x=A..D,F) .....	102
5.4.2.	GPIO 端口输出模式寄存器 (GPIOx_OMODE) (x=A..D,F) .....	104
5.4.3.	GPIO 端口输出速度寄存器 (GPIOx_OSPEED) (x=A..D,F) .....	106
5.4.4.	GPIO 端口上拉/下拉寄存器 (GPIOx_PUPD)(x=A..D,F) .....	107
5.4.5.	GPIO 端口数据输入寄存器 (GPIOx_DIR) (x=A..D,F) .....	109
5.4.6.	GPIO 端口数据输出寄存器 (GPIOx_DOR) (x=A..D,F) .....	109
5.4.7.	GPIO 端口位操作寄存器 (GPIOx_BOR) (x=A..D,F) .....	110
5.4.8.	GPIO 端口配置锁定寄存器 (GPIOx_LOCKR) (x=A,B) .....	110
5.4.9.	GPIO 端口备用功能选择低位寄存器 (GPIOx_AFSL) (x=A,B,C) .....	111
5.4.10.	GPIO 端口选择备用功能高位寄存器 (GPIOx_AFSH) (x=A,B,C) .....	112
5.4.11.	GPIO 端口位清除寄存器 (GPIOx_BCR) (x=A..D,F) .....	114
5.4.12.	GPIO 端口位翻转寄存器 (GPIOx_TGR) (x=A..D,F) (仅适用于 GD32F170xx 和 GD32F190xx 产 品) .....	114
<b>6.</b>	<b>循环冗余校验计算单元 (CRC) .....</b>	<b>115</b>

6.1. 简介 .....	115
6.2. 主要特性 .....	115
6.3. 功能描述 .....	116
6.4. CRC 寄存器 .....	117
6.4.1. CRC 数据寄存器 (CRC_DTR) .....	117
6.4.2. CRC 独立数据寄存器 (CRC_FDTR) .....	117
6.4.3. CRC 控制寄存器 (CRC_CTLR) .....	118
6.4.4. CRC 初值寄存器 (CRC_IDTR) .....	119
7. 中断和事件 .....	120
7.1. 简介 .....	120
7.2. 主要特性 .....	120
7.3. 功能描述 .....	120
7.3.1. NVIC 以及异常/中断处理 .....	120
7.3.2. 外部中断及事件 (EXTI) .....	124
7.4. 中断 EXTI 寄存器 .....	127
7.4.1. 中断使能寄存器 (EXTI_IER) .....	127
7.4.2. 事件使能寄存器 (EXTI_EER) .....	128
7.4.3. 上升沿触发使能寄存器 (EXTI RTE) .....	128
7.4.4. 下降沿触发使能寄存器 (EXTI_FTE) .....	129
7.4.5. 软件中断事件寄存器 (EXTI_SIE) .....	130
7.4.6. 挂起寄存器 (EXTI_PD) .....	130
8. DMA 控制器 (DMA) .....	132
8.1. 简介 .....	132
8.2. 主要特性 .....	132
8.3. 功能描述 .....	132
8.3.1. DMA 传输 .....	132
8.3.2. 通道之间的仲裁 .....	133
8.3.3. 地址生成算法 .....	133
8.3.4. 循环模式 .....	134
8.3.5. 存储器到存储器模式 .....	134
8.3.6. 中断请求 .....	134
8.3.7. DMA 通道配置流程 .....	134
8.3.8. DMA 请求映射 .....	135
8.4. DMA 寄存器 .....	137
8.4.1. DMA 中断状态寄存器 (DMA_IFR) .....	137
8.4.2. DMA 中断标志清除寄存器 (DMA_ICR) .....	138
8.4.3. DMA 通道 x 控制寄存器 (DMA_CTLRx) .....	139
8.4.4. DMA 通道 x 剩余计数器 (DMA_RCNTx) .....	140
8.4.5. DMA 通道 x 外设端基地址寄存器 (DMA_PBARx) .....	141

8.4.6. DMA 通道 x 存储器端基地址寄存器 (DMA_MBArx) .....	141
<b>9. 定时器 (TIMERx).....</b>	<b>142</b>
<b>9.1. 高级定时器 (TIMER1) .....</b>	<b>143</b>
9.1.1. 简介 .....	143
9.1.2. 主要特性 .....	143
9.1.3. 功能描述 .....	143
9.1.4. TIMER1 寄存器 .....	172
<b>9.2. 通用定时器 (TIMER2 和 TIMER3) .....</b>	<b>195</b>
9.2.1. 简介 .....	195
9.2.2. 主要特性 .....	195
9.2.3. 功能描述 .....	195
9.2.4. TIMER2 和 TIMER3 寄存器 .....	218
<b>9.3. 基本定时器 (TIMER6) .....</b>	<b>238</b>
9.3.1. 简介 .....	238
9.3.2. 主要特性 .....	238
9.3.3. 功能描述 .....	238
9.3.4. TIMER6 寄存器 .....	241
<b>9.4. 通用定时器 (TIMER14).....</b>	<b>245</b>
9.4.1. 简介 .....	245
9.4.2. 主要特性 .....	245
9.4.3. 功能描述 .....	245
9.4.4. TIMER14 寄存器 .....	255
<b>9.5. 通用定时器 (TIMER15).....</b>	<b>264</b>
9.5.1. 简介 .....	264
9.5.2. 主要特性 .....	264
9.5.3. 功能描述 .....	264
9.5.4. TIMER15 寄存器 .....	285
<b>9.6. 通用定时器 (TIMER16/TIMER17) .....</b>	<b>302</b>
9.6.1. 简介 .....	302
9.6.2. 主要特性 .....	302
9.6.3. 功能描述 .....	302
9.6.4. TIMER16/TIMER17 寄存器 .....	317
<b>10. 红外线接口 (IFRR) .....</b>	<b>330</b>
<b>10.1. 简介 .....</b>	<b>330</b>
<b>10.2. 主要特性 .....</b>	<b>330</b>
<b>10.3. 功能描述 .....</b>	<b>330</b>
<b>11. 看门狗 (WDG) .....</b>	<b>332</b>
<b>11.1. 独立看门狗 (IWDG).....</b>	<b>332</b>
11.1.1. 简介 .....	332

11.1.2.	主要特性	332
11.1.3.	功能描述	332
11.1.4.	独立看门狗寄存器	334
<b>11.2.</b>	<b>窗口看门狗 (WWDG) .....</b>	<b>337</b>
11.2.1.	简介	337
11.2.2.	主要特性	337
11.2.3.	功能描述	337
11.2.4.	窗口看门狗寄存器	339
<b>12.</b>	<b>模拟数字转换器 (ADC).....</b>	<b>341</b>
<b>12.1.</b>	<b>简介</b>	<b>341</b>
<b>12.2.</b>	<b>主要特性</b>	<b>341</b>
<b>12.3.</b>	<b>功能描述</b>	<b>343</b>
12.3.1.	校准(ADC_CLB)	345
12.3.2.	双时钟域架构	345
12.3.3.	规则组和注入组通道	345
12.3.4.	转换模式	346
12.3.5.	模拟看门狗	348
12.3.6.	注入通道管理	349
12.3.7.	数据对齐	349
12.3.8.	可编程的采样时间	350
12.3.9.	外部触发	350
12.3.10.	DMA 请求	350
12.3.11.	温度传感器和内部参考电压 V <sub>REF</sub>	350
12.3.12.	电池电压监测	351
12.3.13.	ADC 中断	351
12.3.14.	可编程分辨率 (RES) ——快速转换模式	351
12.3.15.	过采样 (仅适用于 GD32F170xx 和 GD32F190xx 产品)	352
<b>12.4.</b>	<b>ADC 寄存器</b>	<b>354</b>
12.4.1.	ADC 状态寄存器 (ADC_STR)	354
12.4.2.	ADC 控制寄存器 1 (ADC_CTLR1)	355
12.4.3.	ADC 控制寄存器 2 (ADC_CTLR2)	358
12.4.4.	ADC 采样时间寄存器 1 (ADC_SPT1)	360
12.4.5.	ADC 采样时间寄存器 2 (ADC_SPT2)	361
12.4.6.	ADC 注入通道数据偏移寄存器 x (ADC_ICOSx) (x=1..4)	362
12.4.7.	ADC 看门狗高阈值寄存器 (ADC_AWHT)	362
12.4.8.	ADC 看门狗低阈值寄存器 (ADC_AWLT)	363
12.4.9.	ADC 规则序列寄存器 1 (ADC_RSQ1)	363
12.4.10.	ADC 规则序列寄存器 2 (ADC_RSQ2)	364
12.4.11.	ADC 规则序列寄存器 3 (ADC_RSQ3)	364
12.4.12.	ADC 注入序列寄存器 (ADC_ISQ)	364
12.4.13.	ADC 注入数据寄存器 x (ADC_IDTRx) (x=1..4)	365

---

12.4.14.	ADC 规则数据寄存器 (ADC_RDTR) .....	365
12.4.15.	ADC 过采样控制寄存器(ADC_OVSCR) （仅适用于 GD32F170xx 和 GD32F190xx 产品） .....	366
<b>13.</b>	<b>数-模转换器 (DAC) . . . . .</b>	<b>368</b>
<b>13.1.</b>	<b>简介 .....</b>	<b>368</b>
<b>13.2.</b>	<b>主要特性 .....</b>	<b>368</b>
<b>13.3.</b>	<b>功能描述 .....</b>	<b>369</b>
13.3.1.	使能 DAC .....	369
13.3.2.	使能 DAC 输出缓存 .....	369
13.3.3.	DAC 数据格式 .....	370
13.3.4.	DAC 转换 .....	370
13.3.5.	DAC 输出电压 .....	370
13.3.6.	DMA 请求 .....	371
13.3.7.	DAC 触发 .....	371
13.3.8.	双 DAC 转换模式 .....	371
<b>13.4.</b>	<b>DAC 寄存器 .....</b>	<b>373</b>
13.4.1.	DAC 控制寄存器 (DAC_CTLR) .....	373
13.4.2.	DAC 软件触发寄存器 (DAC_SWTR) .....	376
13.4.3.	DAC1 12 位右对齐数据保持寄存器 (DAC1_R12DHR) .....	377
13.4.4.	DAC1 12 位左对齐数据保持寄存器 (DAC1_L12DHR) .....	377
13.4.5.	DAC1 8 位右对齐数据保持寄存器 (DAC1_R8DHR) .....	378
13.4.6.	DAC2 12 位右对齐数据保持寄存器 (DAC2_R12DHR) （仅适用于 GD32F170xx 和 GD32F190xx 产 品） .....	378
13.4.7.	DAC2 12 位左对齐数据保持寄存器 (DAC2_L12DHR) （仅适用于 GD32F170xx 和 GD32F190xx 产 品） .....	379
13.4.8.	DAC2 8 位右对齐数据保持寄存器 (DAC2_R8DHR) （仅适用于 GD32F170xx 和 GD32F190xx 产 品） .....	379
13.4.9.	DAC 双模式 12 位右对齐数据保持寄存器 (DACP_R12DHR) （仅适用于 GD32F170xx 和 GD32F190xx 产品） .....	380
13.4.10.	DAC 双模式 12 位左对齐数据保持寄存器(DACP_L12DHR) （仅适用于 GD32F170xx 和 GD32F190xx 产品） .....	380
13.4.11.	DAC 双模式 8 位右对齐数据保持寄存器(DACP_R8DHR) （仅适用于 GD32F170xx 和 GD32F190xx 产品） .....	381
13.4.12.	DAC1 数据输出寄存器 (DAC1_ODR) .....	381
13.4.13.	DAC2 数据输出寄存器 (DAC2_ODR) （仅适用于 GD32F170xx 和 GD32F190xx 产品） .....	382
13.4.14.	DAC 状态寄存器 (DAC_STR) .....	382
<b>14.</b>	<b>I2C 接口 . . . . .</b>	<b>384</b>
<b>14.1.</b>	<b>简介 .....</b>	<b>384</b>
<b>14.2.</b>	<b>主要特性 .....</b>	<b>384</b>
<b>14.3.</b>	<b>功能描述 .....</b>	<b>384</b>
14.3.1.	SDA 线和 SCL 线 .....	385

14.3.2.	数据有效性 .....	386
14.3.3.	开始和停止状态 .....	386
14.3.4.	时钟同步 .....	386
14.3.5.	仲裁 .....	387
14.3.6.	I <sup>2</sup> C 通讯流程 .....	387
14.3.7.	软件编程模型 .....	388
14.3.8.	报文错误校验 .....	396
14.3.9.	SMBus 支持 .....	397
14.3.10.	状态、错误和中断 .....	398
<b>14.4.</b>	<b>I<sup>2</sup>C 寄存器 .....</b>	<b>399</b>
14.4.1.	I <sup>2</sup> C 控制寄存器 1 (I <sup>2</sup> C_CTLR1) .....	399
14.4.2.	I <sup>2</sup> C 控制寄存器 2 (I <sup>2</sup> C_CTLR2) .....	400
14.4.3.	I <sup>2</sup> C 本机地址寄存器 1 (I <sup>2</sup> C_AR1) .....	401
14.4.4.	I <sup>2</sup> C 本机地址寄存器 2 (I <sup>2</sup> C_AR2) .....	402
14.4.5.	I <sup>2</sup> C 传输缓冲区寄存器 (I <sup>2</sup> C_DTR) .....	402
14.4.6.	I <sup>2</sup> C 传输状态寄存器 1 (I <sup>2</sup> C_STR1) .....	402
14.4.7.	I <sup>2</sup> C 传输状态寄存器 2 (I <sup>2</sup> C_STR2) .....	404
14.4.8.	I <sup>2</sup> C 时钟配置寄存器 (I <sup>2</sup> C_CLKR) .....	406
14.4.9.	I <sup>2</sup> C 上升时间寄存器 (I <sup>2</sup> C_RTR) .....	406
14.4.10.	I <sup>2</sup> C SAM 控制状态寄存器 (I <sup>2</sup> C_SAMCSR) （仅适用于 GD32F170xx 和 GD32F190xx 产品）	407
<b>15.</b>	<b>串行外设接口 / 片上音频接口 (SPI/I<sup>2</sup>S) .....</b>	<b>409</b>
<b>15.1.</b>	<b>简介 .....</b>	<b>409</b>
<b>15.2.</b>	<b>主要特性 .....</b>	<b>409</b>
15.2.1.	SPI 特性 .....	409
15.2.2.	I <sup>2</sup> S 特性 .....	410
<b>15.3.</b>	<b>SPI 功能描述 .....</b>	<b>411</b>
15.3.1.	引脚配置 (SPI 单路模式, 默认) .....	411
15.3.2.	引脚配置 (SPI 四路模式) .....	413
15.3.3.	SPI 从模式 .....	414
15.3.4.	SPI 主模式 .....	415
15.3.5.	SPI 四路主模式（仅适用于 GD32F170xx 和 GD32F190xx 产品） .....	415
15.3.6.	SPI 单工通信 .....	417
15.3.7.	数据接收和发送流程 .....	418
15.3.8.	CRC 校验 .....	418
15.3.9.	状态标志和错误标志 .....	419
15.3.10.	停止 SPI .....	420
15.3.11.	DMA 请求 .....	421
15.3.12.	SPI 中断 .....	422
<b>15.4.</b>	<b>I<sup>2</sup>S 功能描述 .....</b>	<b>422</b>
15.4.1.	一般描述 .....	422
15.4.2.	支持的音频标准 .....	423

15.4.3. 时钟生成器 .....	430
15.4.4. 运行 .....	431
15.4.5. DMA 功能 .....	435
<b>15.5. SPI 寄存器 .....</b>	<b>435</b>
15.5.1. SPI 控制寄存器 1 (SPI_CTLR1) .....	435
15.5.2. SPI 控制寄存器 2 (SPI_CTLR2) .....	437
15.5.3. SPI 状态寄存器 (SPI_STR) .....	437
15.5.4. SPI 数据寄存器 (SPI_DTR) .....	439
15.5.5. SPI CRC 多项式寄存器 (SPI_CPR) .....	439
15.5.6. SPI 接收 CRC 寄存器 (SPI_RCR) .....	439
15.5.7. SPI 发送 CRC 寄存器 (SPI_TCR) .....	440
15.5.8. SPI I2S 控制寄存器 (SPI_I2SCTLR) .....	440
15.5.9. SPI I2S 时钟预分频寄存器 (SPI_I2SCKP) .....	442
15.5.10. SPI 四路控制寄存器 (SPI_QWCR) (仅适用于 GD32F170xx 和 GD32F190xx 产品) .....	442
<b>16. 比较器 (CMP) .....</b>	<b>444</b>
16.1. 简介 .....	444
16.2. 主要特性 .....	444
16.3. 功能描述 .....	444
16.3.1. 比较器时钟和复位 .....	446
16.3.2. 比较器输入输出 .....	446
16.3.3. 比较器电源模式 .....	446
16.3.4. 比较器迟滞 .....	446
16.3.5. 比较器寄存器写保护 .....	446
16.4. CMP 寄存器 .....	447
16.4.1. CMP 控制和状态寄存器 (CMP_CSR) .....	447
<b>17. 通用同步异步收发器 (USART) .....</b>	<b>454</b>
17.1. 简介 .....	454
17.2. 主要特性 .....	454
17.3. 功能描述 .....	456
17.3.1. USART 发送器 .....	457
17.3.2. USART 接收器 .....	458
17.3.3. 接收错误 .....	459
17.3.4. 波特率发生 .....	459
17.3.5. 自动波特率检测 .....	459
17.3.6. 多处理器通信 .....	460
17.3.7. ModBus 通信 .....	460
17.3.8. LIN 模式 .....	460
17.3.9. 半双工通信模式 .....	461
17.3.10. 同步通信模式 .....	461
17.3.11. 智能卡(ISO7816)模式 .....	462

17.3.12.	串行红外 (IrDA SIR) 编解码功能模块 .....	464
17.3.13.	硬件流控制 .....	465
17.3.14.	DMA 请求 .....	466
17.3.15.	从 Deep-sleep 模式唤醒 .....	466
17.3.16.	USART 中断 .....	467
<b>17.4.</b>	<b>USART 寄存器 .....</b>	<b>468</b>
17.4.1.	USART 控制寄存器 1 (USART_CTLR1) .....	468
17.4.2.	USART 控制寄存器 2 (USART_CTLR2) .....	470
17.4.3.	USART 控制寄存器 3 (USART_CTLR3) .....	473
17.4.4.	USART 波特率寄存器 (USART_BRR) .....	476
17.4.5.	USART 保护时间和预分频器寄存器 (USART_GTPR) .....	476
17.4.6.	USART 接收超时寄存器 (USART_RTR) .....	477
17.4.7.	USART 请求寄存器 (USART_CMD) .....	478
17.4.8.	USART 状态寄存器 (USART_STR) .....	479
17.4.9.	USART 状态标志清除寄存器 (USART_SCR) .....	482
17.4.10.	USART 数据接收寄存器 (USART_RDTR) .....	484
17.4.11.	USART 数据发送寄存器 (USART_TDTR) .....	484
<b>18.</b>	<b>MCU 调试 (MCUDBG) .....</b>	<b>486</b>
18.1.	简介 .....	486
18.2.	功能描述 .....	486
18.2.1.	省电模式的调试支持 .....	486
18.2.2.	定时器, I2C, RTC, WWDG, IWDG 与 CAN 的调试支持 .....	486
<b>18.3.</b>	<b>MCUDBG 寄存器 .....</b>	<b>486</b>
18.3.1.	MCUDBG ID 寄存器 (MCUDBG_IDR) .....	486
18.3.2.	MCUDBG 控制寄存器 1 (MCUDBG_CTLR1) .....	487
18.3.3.	MCUDBG 控制寄存器 2 (MCUDBG_CTLR2) .....	491
<b>19.</b>	<b>通用串行总线全速设备接口(USB 2.0 FS) .....</b>	<b>493</b>
19.1.	简介 .....	493
19.2.	主要特性 .....	493
19.2.1.	实现 .....	494
19.2.2.	时钟 .....	494
19.2.3.	管脚描述 .....	494
<b>19.3.</b>	<b>功能描述 .....</b>	<b>495</b>
19.3.1.	基本功能模块 .....	495
19.3.2.	缓冲区 .....	497
19.3.3.	端点 .....	498
19.3.4.	中断处理 .....	500
19.3.5.	复位事件 .....	501
19.3.6.	传输流程 .....	502
19.3.7.	电源管理 .....	504

<b>19.4. USB 寄存器 .....</b>	<b>505</b>
19.4.1. USB 控制寄存器 (USB_CTLR).....	505
19.4.2. USB 中断标志寄存器 (USB_IFR).....	507
19.4.3. USB 状态寄存器 (USB_SR).....	507
19.4.4. USB 设备地址寄存器 (USB_AR) .....	508
19.4.5. USB 缓冲器地址寄存器 (USB_BAR) .....	509
19.4.6. USB 端点 n 控制/状态寄存器 (USB_EPnCSR), n=[0..7] .....	509
19.4.7. USB 发送缓冲地址寄存器 n (USB_TXARN) .....	511
19.4.8. USB 发送字节数目寄存器 n (USB_TXCNTn) .....	511
19.4.9. USB 接收缓冲器地址寄存器 n (USB_RXARN) .....	511
19.4.10. USB 接收字节数目寄存器 n (USB_RXCNRn) .....	512
19.4.11. USB 子端点 n 寄存器 (USB_SUBEPPnR), n=[0..7] .....	512
19.4.12. USB LPM 控制寄存器 (USB_LPMCNTR) .....	513
19.4.13. USB LPM 中断状态寄存器 (USB_LPMISTR) .....	513
<b>20. 实时时钟 (RTC) .....</b>	<b>515</b>
<b>20.1. 简介.....</b>	<b>515</b>
<b>20.2. 主要特性 .....</b>	<b>515</b>
<b>20.3. 功能描述 .....</b>	<b>516</b>
20.3.1. RTC 框图.....	516
20.3.2. RTC 管脚.....	517
20.3.3. 时钟和预分频.....	518
20.3.4. 实时时钟和日历 .....	518
20.3.5. 位域可屏蔽可配置的闹钟 .....	519
20.3.6. RTC 初始化和配置.....	519
20.3.7. 读取日历 .....	520
20.3.8. 复位 RTC.....	521
20.3.9. RTC 同步 .....	522
20.3.10. RTC 参考时钟检测.....	522
20.3.11. RTC 平滑数字校准.....	523
20.3.12. 时间戳功能 .....	524
20.3.13. 侵入检测 .....	525
20.3.14. 校准时钟输出 .....	526
20.3.15. 闹钟输出 .....	526
20.3.16. RTC 省电模式管理 .....	526
20.3.17. RTC 中断 .....	526
<b>20.4. RTC 寄存器.....</b>	<b>527</b>
20.4.1. RTC 时间寄存器 (RTC_TSRA) .....	527
20.4.2. RTC 日期寄存器 (RTC_DSR) .....	528
20.4.3. RTC 控制寄存器 (RTC_CTLR) .....	528
20.4.4. RTC 状态和标志寄存器 (RTC_ISTR).....	530
20.4.5. RTC 预分频寄存器 (RTC_PSCR) .....	532
20.4.6. RTC 警报寄存器 (RTC_ALMR) .....	533

20.4.7. RTC 写保护 (RTC_WPR).....	534
20.4.8. RTC 亚秒寄存器 (RTC_SSR).....	534
20.4.9. RTC 移位功能控制寄存器 (RTC_SHIFTCTRLR).....	535
20.4.10. RTC 时间戳时间寄存器 (RTC_TTR) .....	535
20.4.11. RTC 时间戳日期寄存器 (RTC_TDR).....	536
20.4.12. RTC 时间戳亚秒寄存器 (RTC_TSRR).....	537
20.4.13. RTC 校准配置寄存器 (RTC_CCR).....	537
20.4.14. RTC 侵入和复用功能配置寄存器 (RTC_TAFCR) .....	538
20.4.15. RTC 闹钟亚秒寄存器 (RTC_ALMSSR) .....	540
20.4.16. RTC 备份寄存器 (RTC_BKPxR)(x=0,1,2,3,4) .....	541
<b>21. 触摸传感控制器 (TSI) .....</b>	<b>543</b>
<b>21.1. 简介 .....</b>	<b>543</b>
<b>21.2. 主要特性 .....</b>	<b>543</b>
<b>21.3. 功能描述 .....</b>	<b>543</b>
21.3.1. TSI 框图 .....	543
21.3.2. 触摸传感技术概述 .....	544
21.3.3. 电荷转移序列 .....	545
21.3.4. 电荷转移序列状态机 .....	547
21.3.5. 状态时钟和持续时间 .....	548
21.3.6. PIN 模式和 TSI 控制 .....	549
21.3.7. ASW 和迟滞模式 .....	549
21.3.8. TSI 操作流 .....	549
21.3.9. TSI 标志和中断 .....	549
21.3.10. TSI GPIOs .....	550
<b>21.4. TSI 寄存器 .....</b>	<b>550</b>
21.4.1. TSI 控制寄存器 (TSI_CTLR) .....	550
21.4.2. TSI 错误中断或标志中断使能寄存器 (TSI_IER) .....	552
21.4.3. TSI 清错误或标志寄存器 (TSI_CEFR) .....	553
21.4.4. TSI 状态寄存器 (TSI_STR) .....	554
21.4.5. TSI 引脚迟滞模式寄存器 (TSI_PHMR) .....	554
21.4.6. TSI 模拟开关寄存器 (TSI_ASWR) .....	555
21.4.7. TSI 采样引脚寄存器 (TSI_SPR) .....	555
21.4.8. TSI 通道引脚寄存器 (TSI_CPR) .....	556
21.4.9. TSI 组控制寄存器 (TSI_GCTRLR) .....	556
21.4.10. TSI 组 x 周期数寄存器 (TSI_GxCYCNR) .....	557
<b>22. HDMI-CEC 控制器 (HDMI-CEC) .....</b>	<b>558</b>
<b>22.1. 简介 .....</b>	<b>558</b>
<b>22.2. 主要特性 .....</b>	<b>558</b>
<b>22.3. 功能描述 .....</b>	<b>559</b>
22.3.1. CEC 总线引脚 .....	559

22.3.2. 信息说明 .....	559
22.3.3. 位时序说明 .....	560
22.3.4. 仲裁 .....	561
22.3.5. SFTOPT 位说明 .....	561
22.3.6. 错误定义 .....	562
22.3.7. HDMI-CEC 中断 .....	565
<b>22.4. HDMI-CEC 寄存器.....</b>	<b>565</b>
22.4.1. CEC 控制寄存器 (CEC_CTLR) .....	565
22.4.2. CEC 设置寄存器 (CEC_SR) .....	566
22.4.3. CEC 数据发送寄存器 (CEC_TDTR) .....	568
22.4.4. CEC 数据接收寄存器 (CEC_RDTR).....	568
22.4.5. CEC 中断状态寄存器 (CEC_ISTR) .....	568
22.4.6. CEC 中断使能寄存器 (CEC_IER) .....	570
<b>23. LCD 控制器 (LCD) .....</b>	<b>573</b>
23.1. 简介 .....	573
23.2. 主要特性 .....	573
23.3. 功能描述 .....	573
23.3.1. LCD 架构 .....	573
23.3.2. 时钟发生器 .....	574
23.3.3. 闪烁控制 .....	575
23.3.4. SEG/COM 驱动器 .....	575
23.3.5. 双缓冲存储 .....	578
23.3.6. 模拟矩阵 .....	578
23.4. LCD 寄存器 .....	579
23.4.1. LCD 控制寄存器 (LCD_CTLR) .....	579
23.4.2. LCD 配置寄存器 (LCD_CFGR).....	580
23.4.3. LCD 状态标志寄存器 (LCD_SFR) .....	582
23.4.4. LCD 中断清除寄存器 (LCD_ICR) .....	583
23.4.5. LCD 显示数据寄存器 (LCD_DATAx, x=0~7).....	584
<b>24. 运算放大器 (OPAMP)/可编程参考电流和参考电压 (IVREF).....</b>	<b>585</b>
24.1. 运算放大器 (OPAMP) .....	585
24.1.1. 简介 .....	585
24.1.2. 主要特性 .....	585
24.1.3. 功能描述 .....	585
24.1.4. OPAMP 寄存器 .....	588
24.2. 可编程参考电流和参考电压 (IVREF) .....	592
24.2.1. 简介 .....	592
24.2.2. 主要特征 .....	592
24.2.3. 功能描述 .....	592
24.2.4. IVREF 寄存器 .....	593

<b>25. CAN 总线控制器 . . . . .</b>	<b>595</b>
<b>25.1. 简介 . . . . .</b>	<b>595</b>
<b>25.2. 主要特性 . . . . .</b>	<b>595</b>
<b>25.3. 功能描述 . . . . .</b>	<b>596</b>
<b>25.3.1. 工作模式 . . . . .</b>	<b>596</b>
<b>25.3.2. 通信模式 . . . . .</b>	<b>597</b>
<b>25.3.3. 数据发送 . . . . .</b>	<b>598</b>
<b>25.3.4. 数据接收 . . . . .</b>	<b>599</b>
<b>25.3.5. 过滤功能 . . . . .</b>	<b>600</b>
<b>25.3.6. 时间触发通信 . . . . .</b>	<b>603</b>
<b>25.3.7. 通信参数 . . . . .</b>	<b>604</b>
<b>25.3.8. 错误标志 . . . . .</b>	<b>605</b>
<b>25.3.9. 中断 . . . . .</b>	<b>605</b>
<b>25.3.10. bxCAN PHY 模式 . . . . .</b>	<b>606</b>
<b>25.4. CAN 寄存器 . . . . .</b>	<b>607</b>
<b>25.4.1. CAN 控制寄存器 (CAN_CTLR) . . . . .</b>	<b>607</b>
<b>25.4.2. CAN 状态寄存器 (CAN_STR) . . . . .</b>	<b>609</b>
<b>25.4.3. CAN 发送状态寄存器 (CAN_TSTR) . . . . .</b>	<b>610</b>
<b>25.4.4. CAN 接收 FIFO0 寄存器 (CAN_RFRO) . . . . .</b>	<b>612</b>
<b>25.4.5. CAN 接收 FIFO1 寄存器 (CAN_RFR1) . . . . .</b>	<b>612</b>
<b>25.4.6. CAN 中断使能寄存器 (CAN_IER) . . . . .</b>	<b>613</b>
<b>25.4.7. CAN 错误寄存器 (CAN_ER) . . . . .</b>	<b>615</b>
<b>25.4.8. CAN 位时序寄存器 (CAN_BTR) . . . . .</b>	<b>615</b>
<b>25.4.9. CAN 发送邮箱标识符寄存器 (CAN_TMIRx) (x=0..2) . . . . .</b>	<b>616</b>
<b>25.4.10. CAN 发送邮箱属性寄存器 (CAN_TMPRx) (x=0..2) . . . . .</b>	<b>617</b>
<b>25.4.11. CAN 发送邮箱 Data0 寄存器 (CAN_TMD0Rx) (x=0..2) . . . . .</b>	<b>618</b>
<b>25.4.12. CAN 发送邮箱 Data1 寄存器 (CAN_TMD1Rx) (x=0..2) . . . . .</b>	<b>618</b>
<b>25.4.13. CAN 接收 FIFO 邮箱标识符寄存器 (CAN_RFMI Rx) (x=0..1) . . . . .</b>	<b>619</b>
<b>25.4.14. CAN 接收 FIFO 邮箱属性寄存器 (CAN_RFMPRx) (x=0..1) . . . . .</b>	<b>619</b>
<b>25.4.15. CAN 接收 FIFO 邮箱 Data0 寄存器 (CAN_RFMD0Rx) (x=0..1) . . . . .</b>	<b>620</b>
<b>25.4.16. CAN 接收 FIFO 邮箱 Data1 寄存器 (CAN_RFMD1Rx) (x=0..1) . . . . .</b>	<b>620</b>
<b>25.4.17. CAN 过滤器控制寄存器 (CAN_FCTRLR) . . . . .</b>	<b>621</b>
<b>25.4.18. CAN 过滤器模式寄存器 (CAN_FMR) . . . . .</b>	<b>621</b>
<b>25.4.19. CAN 过滤器位宽寄存器 (CAN_FSR) . . . . .</b>	<b>622</b>
<b>25.4.20. CAN 过滤器关联 FIFO 寄存器 (CAN_FAFR) . . . . .</b>	<b>622</b>
<b>25.4.21. CAN 过滤器激活寄存器 (CAN_FWR) . . . . .</b>	<b>623</b>
<b>25.4.22. CAN 过滤器(x) 数据(y)寄存器 (CAN_FxDyR) (x=0..27, y=0..1) . . . . .</b>	<b>623</b>
<b>25.4.23. CAN PHY 控制寄存器(CAN_PHYCR) . . . . .</b>	<b>624</b>
<b>26. 版本历史 . . . . .</b>	<b>625</b>

## 图

图 1-1. Cortex™-M3 框图.....	2
图 1-2. GD32F130xx 和 GD32F150xx 产品的系统结构 .....	3
图 1-3. GD32F170xx 和 GD32F190xx 产品的系统结构 .....	4
图 1-4. GD32F130xx 和 GD32F150xx 产品的存储器映射 .....	6
图 1-5. GD32F170xx 和 GD32F190xx 产品的存储器映射 .....	7
图 2-1. GD32F130xx 和 GD32F150xx 产品的电源域概览 .....	21
图 2-2. GD32F170xx 与 GD32F190xx 产品的电源域概览 .....	22
图 2-3. 上电复位/掉电复位波形图 .....	24
图 2-4. LVD 阈值波形图 .....	24
图 3-1. 页擦除操作流程 .....	34
图 3-2. 整片擦除操作流程 .....	35
图 3-3. 主存储块编程操作流程 .....	36
图 4-1. 系统复位电路 .....	48
图 4-2. GD32F130xx 和 GD32F150xx 产品的时钟树 .....	49
图 4-3. GD32F170xx 和 GD32F190xx 产品的时钟树 .....	50
图 4-4. HSE 接线图 .....	51
图 5-1. 标准 I/O 端口位的基本结构 .....	98
图 5-2. 浮空/上拉/下拉输入配置 .....	100
图 5-3. 输出配置 .....	100
图 5-4. 模拟高阻配置 .....	101
图 5-5. 备用功能配置 .....	102
图 6-1. CRC 计算单元框图 .....	116
图 7-1. EXTI 框图 .....	125
图 8-1. DMA 中断产生逻辑 .....	134
图 8-2. DMA 请求映射 .....	136
图 9-1. 高级定时器框图 .....	144
图 9-2. 当预分频器的参数从 1 变到 2 时, 计数器的时序图 .....	145
图 9-3. 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	145
图 9-4. 计数器时序图, 内部时钟分频系数为 1 .....	146
图 9-5. 计数器时序图, 内部时钟分频系数为 2 .....	146
图 9-6. 计数器时序图, 内部时钟分频系数为 4 .....	147
图 9-7. 计数器时序图, 内部时钟分频系数为 N .....	147
图 9-8. 计数器时序图, 当 ARSE=0 时更新事件 .....	148
图 9-9. 计数器时序图, 当 ARSE=1 时更新事件 .....	148
图 9-10. 计数器时序图, 内部时钟分频系数为 1 .....	149
图 9-11. 计数器时序图, 内部时钟分频系数为 2 .....	150
图 9-12. 计数器时序图, 内部时钟分频系数为 4 .....	150
图 9-13. 计数器时序图, 内部时钟分频系数为 N .....	151
图 9-14. 计数器时序图, 不使用计数器时的更新事件 .....	151
图 9-15. 计数器时序图, 内部时钟分频系数为 1, TIMERx_CARL = 0x5 .....	152
图 9-16. 计数器时序图, 内部时钟分频系数为 2 .....	153

图 9-17. 计数器时序图, 内部时钟分频系数为 4, TIMERx_CARL=0x63.....	153
图 9-18. 计数器时序图, 内部时钟分频系数为 N .....	154
图 9-19. 计数器时序图, 当 ARSE=1 (计数器下溢)时的更新事件.....	154
图 9-20. 计数器时序图, 当 ARSE=1 (计数器上溢)时的更新事件.....	155
图 9-21. 不同模式及不同 TIMERx_CREP 寄存器设置下更新速率的例子 .....	155
图 9-22. 内部时钟分频为 1 时正常模式下的控制电路.....	156
图 9-23. 捕获/比较通道(例如: 通道 1 输入级).....	157
图 9-24. 捕获/比较通道 1 主电路.....	158
图 9-25. 捕获/比较通道输出级 (通道 1 到 3).....	158
图 9-26. 捕获/比较通道输出级 (通道 4).....	158
图 9-27. 输出比较模式, 在 OC1 上切换.....	160
图 9-28. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式.....	160
图 9-29. 在 OC1 上输出比较 PWM 模式 1, 中央对齐计数模式.....	160
图 9-30. 带有死区时间的互补输出 .....	161
图 9-31. 延迟大于负脉冲的死区时间波形 .....	161
图 9-32. 延迟大于正脉冲的死区时间波形 .....	162
图 9-33. 中止功能相应的输出表现 .....	163
图 9-34. 单脉冲模式 .....	164
图 9-35. 编码器接口模式下计数器运行实例 .....	165
图 9-36. 带有 TI1FP1 极性转换的编码器接口模式下的例子 .....	165
图 9-37. 复位模式的控制电路 .....	166
图 9-38. 暂停模式下的控制电路 .....	167
图 9-39. 触发模式下的控制电路 .....	167
图 9-40. 定时器 1 主/从模式的例子 .....	168
图 9-41. 用定时器 2 的使能输出触发定时器 1 .....	169
图 9-42. 用定时器 2 的更新来触发定时器 1 .....	169
图 9-43. 用定时器 2 的使能来选通定时器 1 .....	170
图 9-44. 用定时器 2 的 OC1REF 选通定时器 1 .....	171
图 9-45. 用定时器 2 的 TI1 输入来触发定时器 1 和定时器 2 .....	172
图 9-46. 通用定时器框图(TIMER2 和 TIMER3) .....	196
图 9-47. 当预分频器的参数从 1 变到 2 时, 计数器时序图 .....	196
图 9-48. 当预分频器的参数从 1 变到 4 时, 计数器时序图 .....	197
图 9-49. 计数器时序图, 内部时钟分频系数为 1 .....	198
图 9-50. 计数器时序图, 内部时钟分频系数为 2 .....	198
图 9-51. 计数器时序图, 内部时钟分频系数为 4 .....	199
图 9-52. 计数器时序图, 内部时钟分频系数为 N .....	199
图 9-53. 计数器时序图, 当 ARSE=0 时的更新事件 .....	200
图 9-54. 计数器时序图, 当 ARSE=1 时的更新事件 .....	200
图 9-55. 计数器时序图, 内部时钟分频系数为 1 .....	201
图 9-56. 计数器时序图, 内部时钟分频系数为 2 .....	202
图 9-57. 计数器时序图, 内部时钟分频系数为 4 .....	202
图 9-58. 计数器时序图, 内部时钟分频系数为 N .....	203
图 9-59. 计数器时序图, 不使用计数器时的更新事件 .....	203
图 9-60. 计数器时序图, 内部时钟分频系数为 1, TIMERx_CARL = 0x5.....	204

图 9-61. 计数器时序图, 内部时钟分频系数为 2 .....	205
图 9-62. 计数器时序图, 内部时钟分频系数为 4, TIMERx_CARL=0x63 .....	205
图 9-63. 计数器时序图, 内部时钟分频系数为 N .....	206
图 9-64. 计数器时序图, ARSE=1(计数器下溢) 时的更新事件 .....	206
图 9-65. 计数器时序图, ARSE=1 (计数器上溢) 时的更新事件 .....	207
图 9-66. 内部时钟分频为 1 时正常模式下的控制电路 .....	208
图 9-67. 捕获/比较通道(例如: 通道 1 输入级) .....	209
图 9-68. 捕获/比较通道 1 主电路 .....	210
图 9-69. 捕获/比较通道输出级 (通道 1) .....	210
图 9-70. 输出比较模式, 在 OC1 上切换 .....	211
图 9-71. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式 .....	212
图 9-72. 在 OC1 上输出比较 PWM 模式 1, 中央对齐计数模式 .....	212
图 9-73. 单脉冲模式 .....	213
图 9-74. 编码器接口模式下计数器运行实例 .....	214
图 9-75. 带有 TI1FP1 极性转换的编码器接口模式下的例子 .....	214
图 9-76. 复位模式的控制电路 .....	215
图 9-77. 暂停模式下的控制电路 .....	215
图 9-78. 触发模式下的控制电路 .....	216
图 9-79. 定时器 2 主/从模式的例子 .....	216
图 9-80. 通用定时器框图 (TIMER6) .....	238
图 9-81. 当预分频器的参数从 1 变到 2 时, 计数器时序图 .....	239
图 9-82. 当预分频器的参数从 1 变到 4 时, 计数器时序图 .....	239
图 9-83. 内部时钟分频为 1 时正常模式下的控制电路 .....	240
图 9-84. 通用定时器框图 (TIMER14) .....	245
图 9-85. 当预分频器的参数从 1 变到 2 时, 计数器时序图 .....	246
图 9-86. 当预分频器的参数从 1 变到 4 时, 计数器时序图 .....	246
图 9-87. 计数器时序图, 内部时钟分频系数为 1 .....	247
图 9-88. 计数器时序图, 内部时钟分频系数为 2 .....	248
图 9-89. 计数器时序图, 内部时钟分频系数为 4 .....	248
图 9-90. 计数器时序图, 内部时钟分频系数为 N .....	249
图 9-91. 计数器时序图, 当 ARSE=0 时的更新事件 .....	250
图 9-92. 计数器时序图, 当 ARSE=1 时的更新事件 .....	250
图 9-93. 内部时钟分频为 1 时正常模式下的控制电路 .....	251
图 9-94. 捕获/比较通道(例如: 输入级) .....	252
图 9-95. 捕获/比较通道 1 主电路 .....	252
图 9-96. 捕获/比较通道输出级 .....	253
图 9-97. 输出比较模式, 在 OC1 上切换 .....	254
图 9-98. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式 .....	254
图 9-99. 在 OC1 上输出比较 PWM 模式 1, 中央对齐计数模式 .....	255
图 9-100. 通用定时器框图 (TIMER15) .....	265
图 9-101. 当预分频器的参数从 1 变到 2 时, 计数器时序图 .....	266
图 9-102. 当预分频器的参数从 1 变到 4 时, 计数器时序图 .....	266
图 9-103. 计数器时序图, 内部时钟分频系数为 1 .....	267
图 9-104. 计数器时序图, 内部时钟分频系数为 2 .....	267

图 9-105. 计数器时序图, 内部时钟分频系数为 4.....	268
图 9-106. 计数器时序图, 内部时钟分频系数为 N.....	268
图 9-107. 计数器时序图, 当 ARSE=0 时更新事件.....	269
图 9-108. 计数器时序图, 当 ARSE=1 时更新事件.....	269
图 9-109. 不同模式及不同 TIMERx_CREP 寄存器设置下更新速率的例子.....	270
图 9-110. 内部时钟分频为 1 时正常模式下的控制电路.....	271
图 9-111. 捕获/比较通道(例如: 通道 1 输入级).....	272
图 9-112. 捕获/比较通道 1 主电路.....	272
图 9-113. 捕获/比较通道输出级(通道 1).....	273
图 9-114. 捕获/比较通道输出级(通道 2).....	273
图 9-115. 输出比较模式, 在 OC1 上切换.....	274
图 9-116. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式.....	275
图 9-117. 在 OC1 上输出比较 PWM 模式 1, 中央对齐计数模式.....	275
图 9-118. 带有死区时间的互补输出.....	276
图 9-119. 延迟大于负脉冲的死区时间波形.....	276
图 9-120. 延迟大于正脉冲的死区时间波形.....	276
图 9-121. 响应中止的输出表现 (中止输入高电平有效) .....	277
图 9-122. 单脉冲模式.....	278
图 9-123. 复位模式的控制电路.....	279
图 9-124. 暂停模式下的控制电路.....	279
图 9-125. 触发模式下的控制电路.....	280
图 9-126. 定时器 15 主从模式的例子.....	280
图 9-127. 用定时器 2 的使能输出触发定时器 15 .....	281
图 9-128. 用定时器 2 的更新来触发定时器 15 .....	282
图 9-129. 用定时器 2 的使能来选通定时器 15 .....	282
图 9-130. 用定时器 2 的 OC1REF 选通定时器 15 .....	283
图 9-131. 用定时器 2 的 TI1 输入来触发定时器 15 和定时器 2 .....	284
图 9-132. 通用定时器框图 (TIMER16/17).....	303
图 9-133. 当预分频器的参数从 1 变到 2 时, 计数器时序图 .....	304
图 9-134. 当预分频器的参数从 1 变到 4 时, 计数器时序图 .....	304
图 9-135. 计数器时序图, 内部时钟分频系数为 1.....	305
图 9-136. 计数器时序图, 内部时钟分频系数为 2.....	305
图 9-137. 计数器时序图, 内部时钟分频系数为 4.....	306
图 9-138. 计数器时序图, 内部时钟分频系数为 N.....	306
图 9-139. 计数器时序图, 当 ARSE=0 时的更新事件 .....	307
图 9-140. 计数器时序图, 当 ARSE=1 时更新事件 .....	307
图 9-141. 不同模式及不同 TIMERx_CREP 寄存器设置下更新速率的例子 .....	308
图 9-142. 内部时钟分频为 1 时正常模式下的控制电路.....	309
图 9-143. 捕获/比较通道(例如: 通道 1 输入级).....	310
图 9-144. 捕获/比较通道 1 主电路 .....	310
图 9-145. 捕获/比较通道输出级 (通道 1).....	311
图 9-146. 输出比较模式, 在 OC1 上切换.....	312
图 9-147. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式.....	312
图 9-148. 在 OC1 上输出比较 PWM 模式 1, 中央对齐计数模式.....	313

图 9-149. 带有死区时间的互补输出.....	314
图 9-150. 延迟大于负脉冲的死区时间波形.....	314
图 9-151. 延迟大于正脉冲的死区时间波形.....	314
图 9-152. 响应中止的输出表现（中止输入高电平有效）.....	315
图 9-153. 单脉冲模式.....	316
图 10-1. IFRR 输出时序图 1 .....	330
图 10-2. IFRR 输出时序图 2 .....	331
图 10-3. IFRR 输出时序图 3 .....	331
图 11-1. 独立看门狗框图 .....	333
图 11-2. 窗口看门狗框图 .....	337
图 11-3. 窗口看门狗时序图 .....	338
图 12-1. GD32F130xx 与 GD32F150xx 产品的 ADC 模块图 .....	343
图 12-2. GD32F170xx 与 GD32F190xx 产品的 ADC 模块 .....	344
图 12-3. 单次转换模式.....	346
图 12-4. 连续转换模式，禁用扫描 .....	347
图 12-5. 连续转换模式，使用扫描 .....	347
图 12-6. 扫描转换模式.....	348
图 12-7. 间断转换模式.....	348
图 12-8. 12 位分辨率的数据对齐 .....	349
图 12-9. 20 位到 16 位的结果截取 .....	352
图 12-10. 右移 5 位和四舍五入的数例 .....	353
图 13-1. DAC 通道模块框图 .....	369
图 14-1. I2C 模块框图.....	385
图 14-2. 数据有效性 .....	386
图 14-3. 开始和停止状态 .....	386
图 14-4. 时钟同步 .....	387
图 14-5. SDA 线仲裁 .....	387
图 14-6. 7 位地址的 I2C 通讯流程 .....	388
图 14-7. 10 位地址的 I2C 通讯流程 .....	388
图 14-8. 从机发送模式 .....	390
图 14-9. 从机接收模式 .....	391
图 14-10. 主机发送模式 .....	392
图 14-11. 主机接收使用方案 A 模式 .....	394
图 14-12. 主机接收使用方案 B 模式 .....	396
图 15-1. 单主/单从应用 .....	411
图 15-2. SPI 数据时钟时序图 .....	412
图 15-3. SPI 四路模式下数据时序图（SCKPL=1, SCKPH=1） .....	414
图 15-4. 使用 DMA 发送 .....	421
图 15-5. 使用 DMA 接收 .....	422
图 15-6. I2S 框图 .....	423
图 15-7. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0) .....	424
图 15-8. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1) .....	424
图 15-9. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0) .....	424
图 15-10. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1) .....	424

图 15-11. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0) .....	424
图 15-12. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1) .....	425
图 15-13. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0) .....	425
图 15-14. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1) .....	425
图 15-15. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0) .....	425
图 15-16. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1) .....	425
图 15-17. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0) .....	426
图 15-18. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1) .....	426
图 15-19. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) .....	426
图 15-20. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) .....	426
图 15-21. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) .....	426
图 15-22. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) .....	426
图 15-23. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) .....	427
图 15-24. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) .....	427
图 15-25. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) .....	427
图 15-26. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) .....	427
图 15-27. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0) .....	427
图 15-28. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1) .....	428
图 15-29. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0) .....	428
图 15-30. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1) .....	428
图 15-31. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0) .....	428
图 15-32. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1) .....	428
图 15-33. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0) .....	428
图 15-34. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1) .....	428
图 15-35. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0) .....	429
图 15-36. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1) .....	429
图 15-37. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0) .....	429
图 15-38. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1) .....	429
图 15-39. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0) .....	429
图 15-40. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1) .....	429
图 15-41. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0) .....	429
图 15-42. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1) .....	429
图 15-43. I2S 时钟生成器框图 .....	430
图 16-1. GD32F130xx 与 GD32F150xx 产品的 CMP 框图 .....	445
图 16-2. GD32F170xx 与 GD32F190xx 产品的 CMP 框图 .....	445
图 17-1. USART 模块内部框图 .....	457
图 17-2. USART 字符帧 (9 数据位和 1 停止位) .....	458
图 17-3. LIN 模式下的帧格式错误检测以及断开帧检测 .....	461
图 17-4. 同步模式下的 USART 示例 .....	461
图 17-5. 8-bit 格式的 USART 同步通信波形 (LBCL=1) .....	462
图 17-6. ISO7816-3 帧格式 .....	462
图 17-7. IrDA SIR ENDEC 模块 .....	464
图 17-8. IrDA 数据调制 .....	465
图 17-9. 两个 USARTs 之间的硬件流控制 .....	465

---

图 17-10. 硬件流控.....	466
图 17-11. USART 中断映射框图 .....	468
图 19-1.USB 模块图 .....	495
图 19-2.缓冲描述符表的用法示例（USB_BAR = 0） .....	498
图 19-3.USB 事务与中断事件.....	502
图 21-1. TSI 模块框图.....	544
图 21-2. 一个通道引脚的采样引脚的框图.....	545
图 21-3. 电荷序列传输期间的采样 引脚的电压.....	547
图 21-4. 电荷转移序列的有限状态机的状态转移图.....	547
图 22-1. HDMI-CEC 控制器框图.....	559
图 23-1. LCD 模块框图.....	574
图 23-2. 1/3 偏置, 1/4 占空比.....	575
图 23-3. 1/4 偏置 1/6 占空比.....	577
图 23-4. LCD 死区时间 (1/3 偏置, 1/4 占空比) .....	577
图 23-5. 电阻分压网络.....	578
图 24-1. OPAMP1 信号路由 .....	586
图 24-2. OPAMP2 信号路由 .....	586
图 24-3. OPAMP3 信号路由 .....	586
图 25-1. CAN 模块结构图 .....	596
图 25-2. 发送寄存器 .....	598
图 25-3. 发送邮箱状态转换 .....	598
图 25-4. 接收寄存器 .....	600
图 25-5. 32-bit 位宽过滤器 .....	601
图 25-6. 16-bit 位宽过滤器 .....	601
图 25-7. 32-bit 位宽掩码模式过滤器 .....	601
图 25-8. 32-bit 位宽列表模式过滤器 .....	601
图 25-9. 过滤序号 .....	602
图 25-10. 过滤索引 .....	603
图 25-11. 位时序 .....	604
图 25-12. bxCAN PHY 连接图 .....	607

## 表

表 1-1. Flash 模块组织.....	9
表 1-2. Flash 模块组织.....	9
表 1-3. 引导模式.....	10
表 2-1. 节电模式总结 .....	26
表 3-1. 闪存存储器的基地址和大小 .....	32
表 3-2. 闪存存储器的基地址和大小 .....	32
表 3-3. 可选字节 .....	37
表 3-4. OB_WP 位对应页保护 .....	38
表 4-1. 时钟源的选择 .....	53
表 4-2. 时钟源的选择 .....	53
表 4-3. 深度睡眠模式下内核电压选择 .....	54
表 4-4. 深度睡眠模式下内核电压选择 .....	55
表 5-1. GPIO 配置表 .....	98
表 7-1. Cotrex-M3 中的 NVIC 异常类型 .....	121
表 7-2. GD32F130xx 和 GD32F150xx 的中断向量表 .....	121
表 7-3 GD32F170xx 和 GD32F190xx 的中断向量表 .....	123
表 7-4. GD32F130xx 和 GD32F150xx 的 EXTI 触发源 .....	126
表 7-5 GD32F170xx 和 GD32F190xx 的 EXTI 触发源 .....	127
表 8-1. DMA 传输操作 .....	133
表 8-2. DMA 中断事件 .....	134
表 8-3. 各 DMA 通道请求总结 .....	137
表 9-1. 计数方向对编码器信号 .....	165
表 9-2. 计数方向对编码器信号 .....	214
表 11-1. 独立看门狗在 40 kHz (LSI)时的最小/最大暂停时间.....	333
表 11-2. MHz (fPCLK1)时的最大最小值.....	338
表 12-1. ADC 内部信号 .....	344
表 12-2. GD32F130xx 和 GD32F150xx 产品的 ADC 引脚定义 .....	344
表 12-3. GD32F170xx 和 GD32F190xx 产品的 ADC 引脚定义 .....	344
表 12-4. ADC 用于规则通道的外部触发 .....	350
表 12-5. ADC 用于注入通道的外部触发 .....	350
表 12-6. 不同的分辨率对应的 tSAR 时间参数 .....	352
表 12-7. N 和 M 的最大输出值.....	353
表 13-1. DAC 引脚 .....	369
表 13-2. DAC 的外部触发源 .....	371
表 14-1. I2C 总线术语说明 .....	385
表 14-2. 事件状态标志位 .....	398
表 14-3. I2C 错误标志位 .....	398
表 15-1. SPI 中断请求 .....	422
表 15-2. 比特率计算公式 .....	430
表 15-3. 音频采样率计算公式 .....	430
表 15-4. 音频采样率配置和精度 .....	430

---

表 15-5. 各个运行模式下 I2S 接口信号的方向 .....	431
表 15-6. I2S 中断 .....	433
表 17-1. USART 重要管脚描述 .....	456
表 17-2. 停止位配置 .....	458
表 17-3. USART 中断请求 .....	467
表 19-1. GD32F150xx USB 实现 .....	494
表 19-2. GD32F150xx USB 信号管脚 .....	495
表 19-3. 双缓冲标志定义 .....	499
表 19-4. 双缓冲的用法 .....	499
表 19-5. 接收状态编码 .....	510
表 19-6. 端点类型编码 .....	510
表 19-7. 端点类别编码 .....	510
表 19-8. 发送状态编码 .....	510
表 19-9. 子端点状态 .....	513
表 21-1. 电荷转移序列的详细步骤以及引脚和开关状态 .....	545
表 21-2. 充电扩展状态的持续时间 .....	548
表 21-3. TSI 错误和标志位 .....	549
表 21-4. TSI 引脚 .....	550
表 22-1. 数据位时序参数表 .....	560
表 22-2. 错误处理时序参数表 .....	563
表 22-3. TE 时序参数表 .....	564
表 23-1. 奇数帧电压 .....	575
表 23-2. 偶数帧电压 .....	575
表 23-3. 所有 COM 信号驱动器 .....	576
表 23-4. VLCDrail 连接引脚 .....	579
表 24-1. 工作模式和校准 .....	587
表 26-1. 版本历史 .....	625

## 1. 系统及存储器架构

GD32F1x0系列器件的系统架构包括了ARM® Cortex™-M3处理器，总线架构和存储器组织，将在下面的章节中描述。Cortex™-M3处理器是新一代的处理器核心，它提供了许多新的功能。高度集成以及各种先进的功能使Cortex™-M3处理器适用于需要微控制器具有高性能和低功耗的市场产品。Cortex™-M3处理器包括3条AHB总线分别称为I-CODE总线、D-Code总线和系统总线。Cortex™-M3处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这三个总线上执行。存储器的组织采用了哈佛结构，预先定义的存储器映射和高达4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

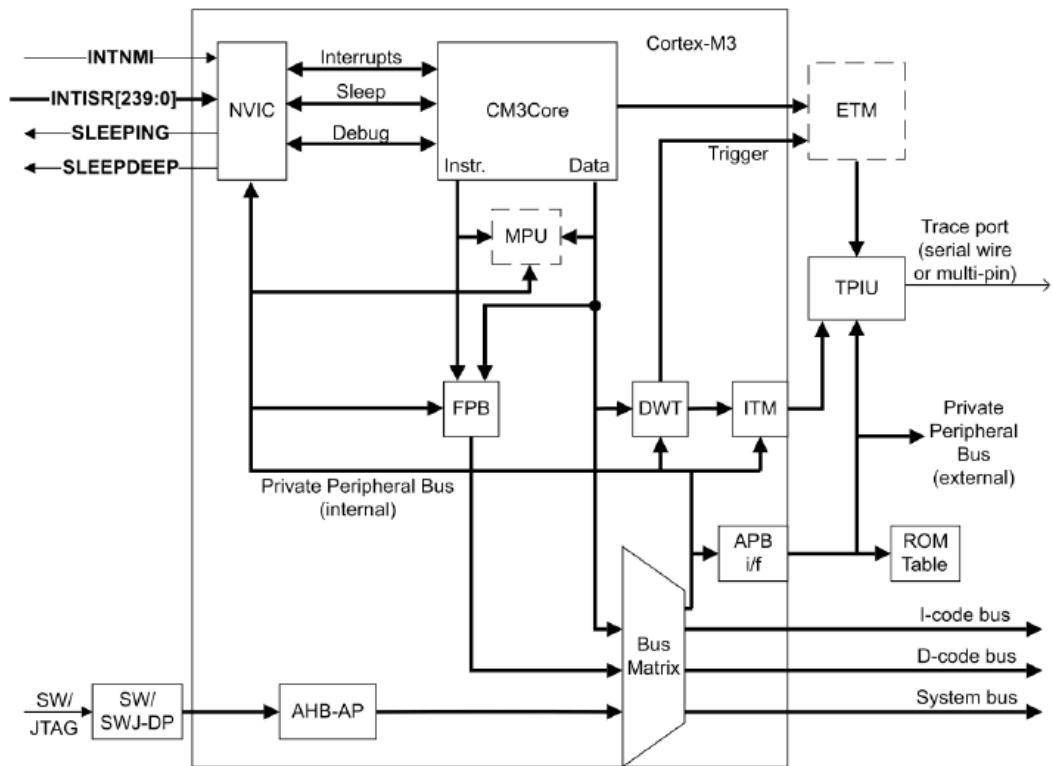
### 1.1. ARM Cortex-M3 处理器

Cortex™-M3处理器是一个通用的32位处理器，非常适合那些兼顾高性能和低功耗的应用领域。它提供了许多新的功能，如Thumb-2指令集，硬件除法器，低延迟的中断响应时间，位带操作和多条总线的同时访问。Cortex™-M3处理器基于ARMv7架构，同时支持Thumb和Thumb-2指令集。下面列出了Cortex™-M3提供的一些系统外设：

- 内部总线矩阵，用于实现I-Code总线、D-Code总线、系统总线、专用总线(PPB)以及调试专用总线(AHB-AP)的互联
- 嵌套式向量型中断控制器 (NVIC)
- 闪存地址重载及断点单元 (FPB)
- 数据观测点及跟踪单元(DWT)
- 指令跟踪宏单元 (ITM)
- 串行线和JTAG调试接口 (SWJ-DP)
- 跟踪端口接口单元 (TPIU)

下图显示了Cortex™-M3处理器框图。欲了解更多信息，请参阅ARM® Cortex™-M3技术参考手册。

图 1-1. Cortex™-M3 框图



## 1.2. 系统架构

GD32F1x0系列器件的系统结构如下图所示。基于AMBA 3.0 AHB-LITE的多层次总线互连矩阵使得系统中的多个主机和从机之间的并行通信成为可能。总线矩阵有四个主机，包括I-CODE总线、D-Code总线、Cortex™-M3内核系统总线和DMA的系统总线。I-CODE总线是指令总线，用于将代码区域(0x0000 0000~0x1FFF FFFF)的向量取到Cortex™-M3内核。D-Code总线用于数据加载和存储，也用于代码区的调试访问。系统总线用于系统区域的访问，包括取指令，数据加载和存储，以及调试访问。系统区域包括内部SRAM区域和外设区域。总线矩阵包含5个从机，包括闪存控制器的I-CODE和D-Code接口、内部SRAM、AHB1和AHB2。

AHB2与GPIO端口相连。AHB1与AHB外设相连。诸多AHB外设之中有两个AHB到APB的桥提供了AHB1和两条APB总线之间的完全同步连接。这两条APB总线连接所有的APB外设。两条APB总线都能够按最高频率72MHz工作。

图 1-2. GD32F130xx 和 GD32F150xx 产品的系统结构

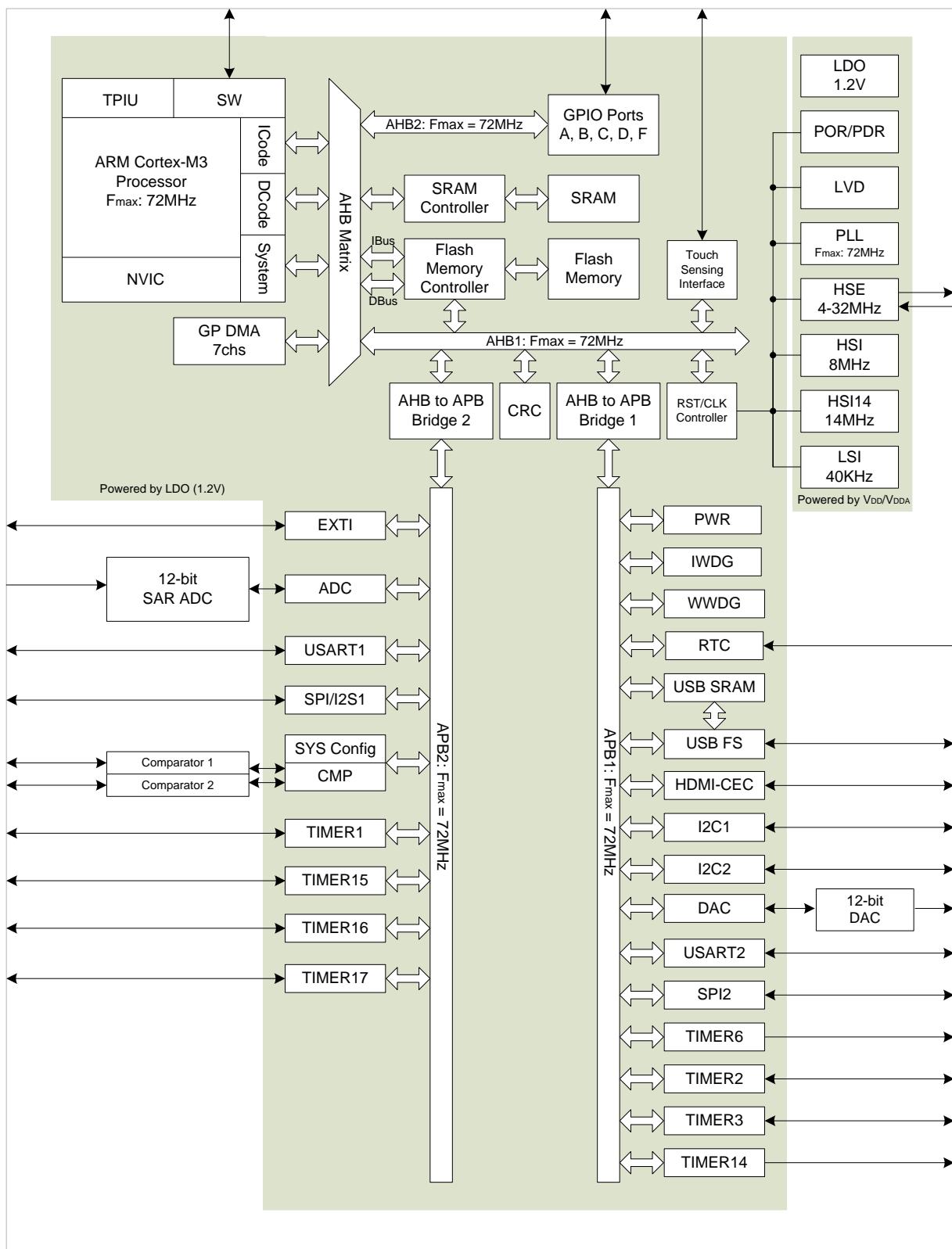
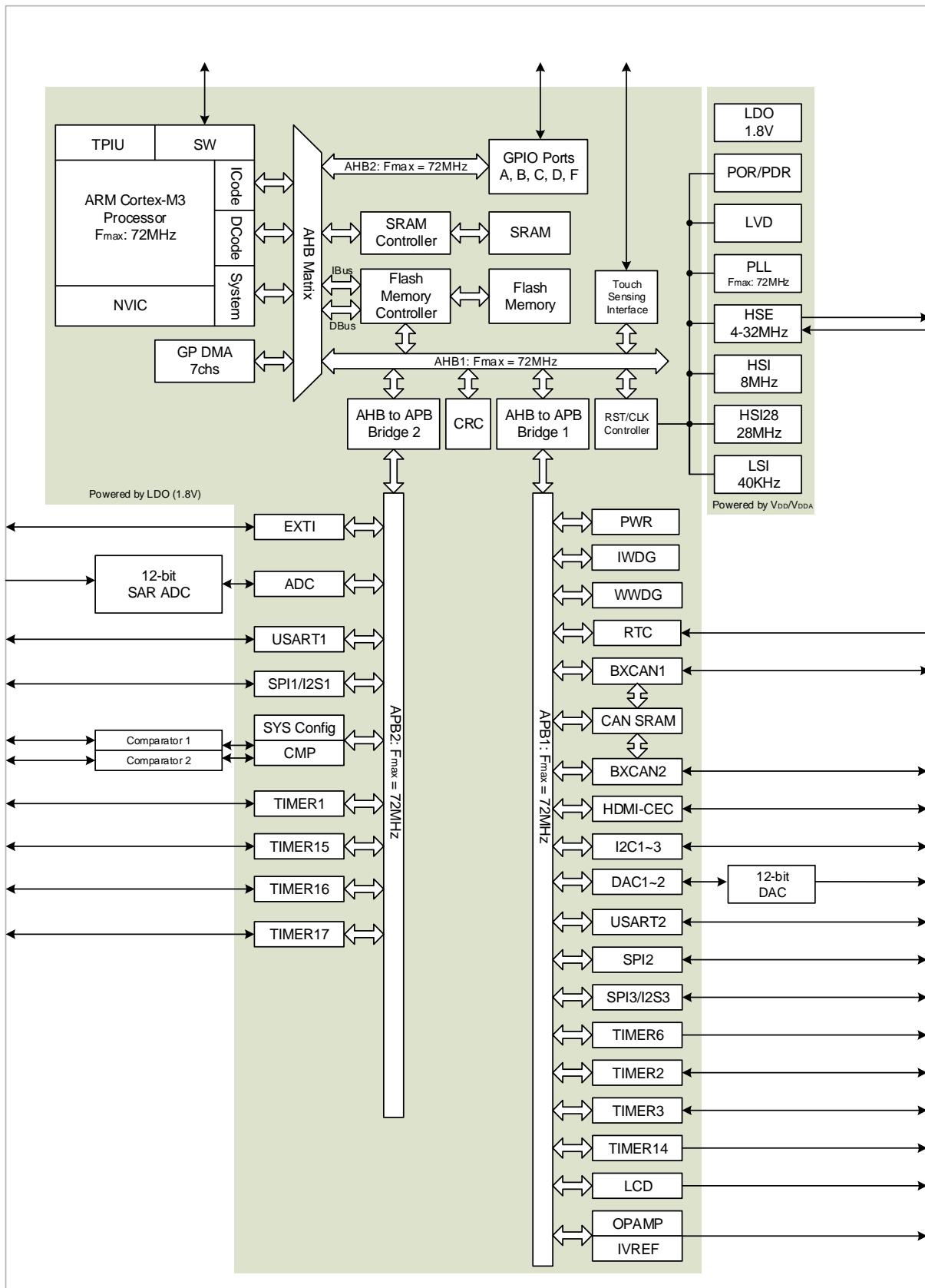


图 1-3. GD32F170xx 和 GD32F190xx 产品的系统结构



### 1.3. 存储器映射

ARM® Cortex™-M3处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和I/O端口都在同一个线性的4 GB的地址空间之内。这是Cortex™-M3的最大地址范围，因为它的地址总线宽度是32位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按Cortex™-M3处理器提供的规则预先定义的。ARM® Cortex™-M3的自带的系统外设也占用了一些地址空间。下图显示了GD32F1x0系列设备的存储映射，包括指令代码、SRAM、外设和其他预先定义的区域。为了简化外设的地址译码，每个外设分配的地址空间都是1KB。APB1外设都位于从0x4000 0000到0x4000 FFFF的地址空间，而APB2外设都位于从0x4001 0000到0x40017FFF的地址区域。从0x4002 0000到0x4002 FFFF的地址区域被AHB1外围设备所使用，从0x48000000到0x4800FFFF的地址区域被AHB2外围设备所使用。

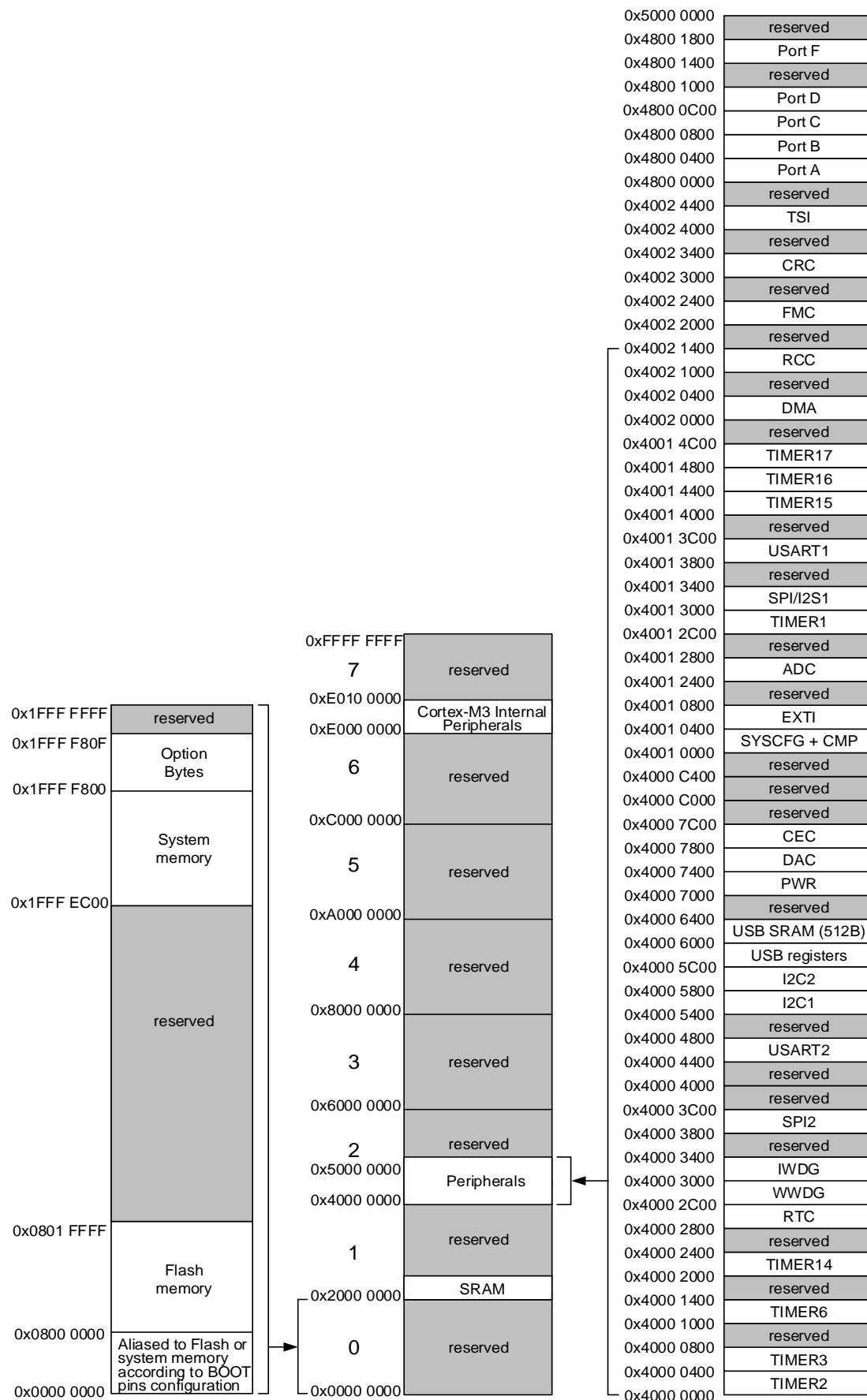
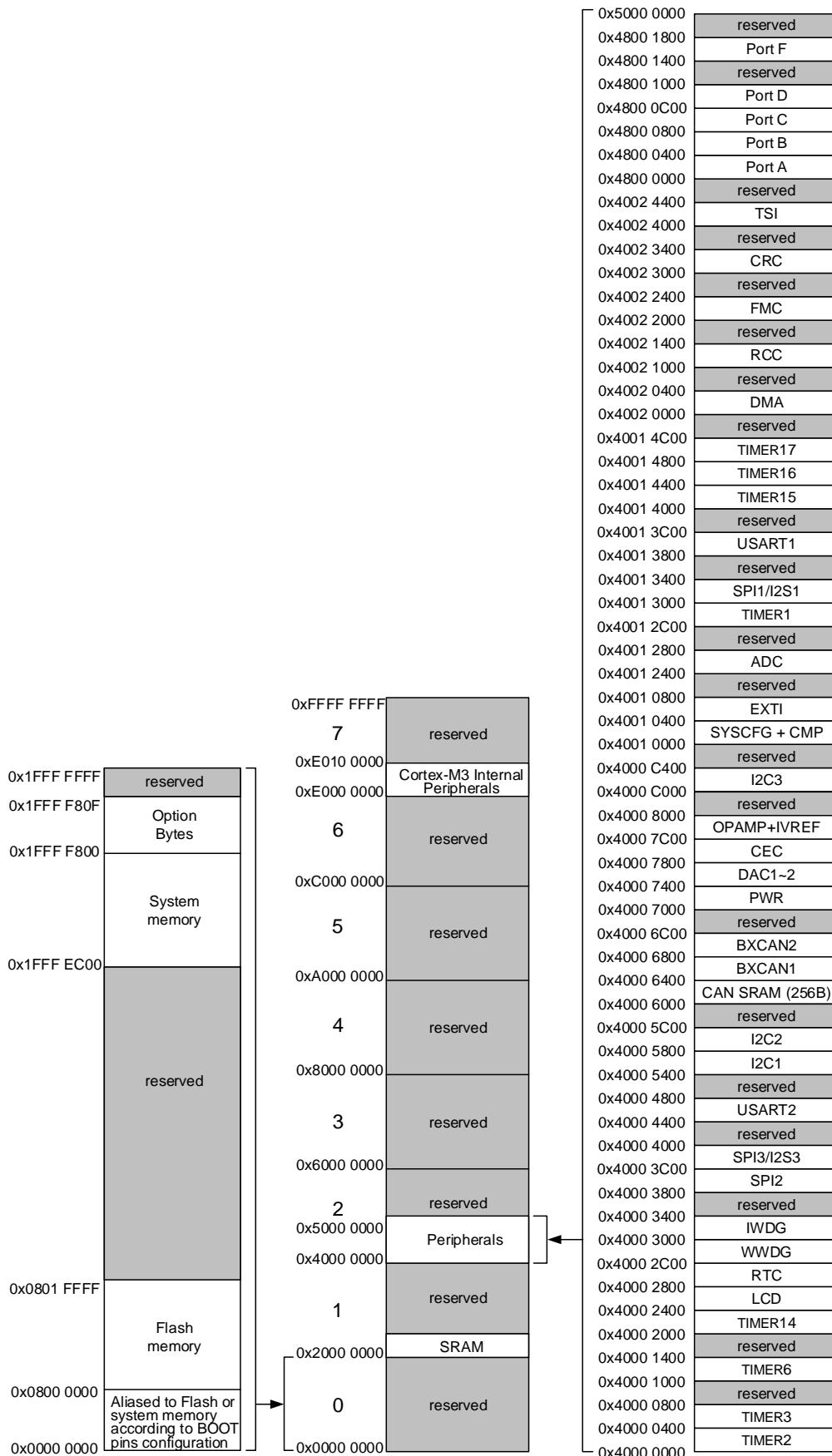
**图 1-4. GD32F130xx 和 GD32F150xx 产品的存储器映射**


图 1-5. GD32F170xx 和 GD32F190xx 产品的存储器映射



### 1.3.1. 位带操作

支持了位带操作后，用户可以使用普通的加载/存储指令来对单一的比特进行读写。Cortex™-M3处理器提供了两个支持位带操作的区域。其中一个是SRAM区的最低1MB范围，第二个是片内外设区的最低1MB范围。这两个区域中的地址除了普通应用外，还有自己的“位带别名区”。位带别名区把每个比特扩展成一个32位的字。当用户访问位带别名区时，就可以达到访问原始比特的目的。

下面的映射公式表明了位带别名区的每个32位字如何对应位带区的某个比特。

$$\text{位带别名区地址} = \text{位带别名区基址} + (\text{字节偏移} \times 32) + (\text{位数} \times 4)$$

其中：

- 位带别名区地址指的是位带区目标比特对应在位带别名区的地址
- 位带别名区基址指的是位带别名区的起始地址
- 字节偏移指的是位带区目标比特所在的字节的字节地址偏移量
- 位数指的是目标比特在对应字节中的位置

例如，要想访问0x2000 0200地址的第7位，可访问的位带别名区地址是：

$$\text{位带别名地址} = 0x2200\ 0000 + (0x200 * 32) + (7 * 4) = 0x2200\ 401C$$

如果对0x2200 401C进行写操作，那么0x2000 0200的第7位将会相应变化；如果对0x2200 401C进行读操作，那么视0x2000 0200的第7位状态而返回0x0000 0001或0x0000 0000。

### 1.3.2. 片上 SRAM

GD32F1x0系列微控制器含有高达8KB的片上SRAM，起始地址为0x20000000。它支持8位、16位和32位访问。为了增加存储器的鲁棒性，支持奇偶校验。用户可以通过用户可选字节(请参考第3.3.9章可选字节)的OB\_SRAM\_PARITY\_CHECK位来启用奇偶校验功能。当启用时，如果校验失败，产生一个NMI中断。SRAM奇偶校验错误标志在系统配置寄存器2(SYSCFG\_R2)之中。如果系统配置寄存器2(SYSCFG\_R2)的SRAM\_PARITY\_ERROR\_LOCK位置1，错误标志将被连接到定时器1/定时器15/定时器16/定时器17的break输入端。

SRAM的真实数据宽度为36位，包括32位数据和4位奇偶校验(每字节1位)位。在写入时，奇偶校验位被计算并存储到SRAM。当读取时，奇偶校验位会用SRAM读出的数据再计算一遍。计算出的奇偶校验位将与读出的奇偶校验位(写入访问期间计算并存储的奇偶校验位)进行比较。如果它们不相同，则奇偶校验失败。

**注意：**如果启用了SRAM奇偶校验，建议通过软件在代码的开始初始化整个SRAM存储器，以避免读取未初始化的位置时，得到的奇偶校验错误。

### 1.3.3. 片上闪存

**GD32F130xx和GD32F150xx产品：**

该系列微控制器提供高达64 KB的片上闪存。片上闪存包括最大高达64KB的主存模块和3KB容量的用于存储引导加载程序的信息模块。主存模块分为64页，每页的容量为1KB。下表显示详细信息。

**表 1-1. Flash 模块组织**

模块	名称	地址	尺寸
主存模块	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbytes
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbytes
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbytes
	.	.	.
	.	.	.
	Page 63	0x0800 FC00 - 0x0800 FFFF	1 Kbytes
	系统存储器	0x1FFF EC00 - 0x1FFF F7FF	3 Kbytes
信息模块	可选字节	0x1FFF F800 - 0x1FFF F80F	16 bytes

针对前32页的读访问可实现每个周期无任何等待状态读出32位数据的效率。读访问支持8位、16位和32位；写访问（编程）只支持16位和32位。片上闪存的每一页都可以单独被擦除，整个主存模块也可以同时被擦除。

#### GD32F170xx和GD32F190xx产品：

该系列微控制器提供高达128KB的片上闪存。片上闪存包括最大高达128KB的主存模块和3KB容量的用于存储引导加载程序的信息模块。主存模块分为128页，每页的容量为1KB。下表显示详细信息。

**表 1-2. Flash 模块组织**

模块	名称	地址	尺寸
主存模块	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbytes
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbytes
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbytes
	.	.	.
	.	.	.
	Page 128	0x0801 FC00 - 0x0801 FFFF	1 Kbytes
	系统存储器	0x1FFF EC00 - 0x1FFF F7FF	3 Kbytes
信息模块	可选字节	0x1FFF F800 - 0x1FFF F80F	16 bytes

针对前32页的读访问可实现每个周期无任何等待状态读出32位数据的效率。读访问支持8位、16位和32位；写访问（编程）只支持16位和32位。片上闪存的每一页都可以单独被擦除，整个主存模块也可以同时被擦除。

## 1.4. 引导配置

GD32F1x0 系列微控制器提供了三种引导源，可以通过用户可选字 OB\_BOOT1\_n 位(请参考第 3.3.9 章选项字节)和 BOOT0 进行选择。BOOT0 指的是复位后第 4 个系统时钟上升沿锁存的 BOOT0 引脚的值。通过 OB\_BOOT1\_n 和 BOOT0 选择引导源的详细信息列于下表中。

表 1-3. 引导模式

启动源选择	启动模式选择引脚	
	Boot1 <sup>(1)</sup>	Boot0
片上闪存的主存	x	0
片上闪存的系统存储器	0	1
片上SRAM	1	1

### 1. BOOT1值与OB\_BOOT1\_n值相反

上电后序列或系统复位后，ARM®Cortex™-M3处理器先从0x0000 0000地址获取栈顶值，再从0x0000 0004地址获得引导代码的基地址，然后从引导代码的基地址开始执行程序。

根据所选的引导源，片上闪存的主存（开始于0x0800 0000的存储空间）或系统存储器（开始于0x1FFF EC00的存储空间）会被映射到引导空间，即从0x0000 0000开始的地址空间。如果片上SRAM（开始于0x2000 0000的存储空间）被选为引导源，用户必须在应用程序初始化代码中通过修改NVIC异常向量表和偏移寄存器将向量表重置到SRAM中。

芯片内嵌的引导加载程序位于系统存储器中，用来对片上闪存的主存编程。在GD32F1x0系列微控制器中，引导装载程序可通过以下串行接口之一工作：USART1或USART2。

## 1.5. 系统配置寄存器 (SYSCFG)

### 1.5.1. 系统配置寄存器 1 (SYSCFG\_R1)

地址偏移：0x00

复位值：0x0000 000X (根据BOOT0引脚的状态和用户可选字节的OB\_BOOT1\_n的值，X表示BOOT\_MODE[1:0]可能为任意值)

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										PB9_HCCE	保留				
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIMER17_DMA_RMP	TIMER16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	保留			BOOT_MODE[1:0]						
	rw	rw	rw	rw	rw										

位/位域	名称	描述
31:20	保留	必须保持复位值
19	PB9_HCCE	PB9引脚大电流能力使能 当该位为1时，PB9引脚可以直接用来控制红外发光二极管。 0: PB9引脚大电流能力关闭

1: PB9引脚大电流能力开启，同时该引脚的速度控制被忽略

18:13	保留	必须保持复位值
12	TIMER17_DMA_RMP	Timer 17 DMA 请求重映射使能 0: 不重映射 (TIMER17_CH1和TIMER17_UP DMA被映射在DMA通道1) 1: 重映射 (TIMER17_CH1和TIMER17_UP DMA被映射在DMA通道2)
11	TIMER16_DMA_RMP	Timer 16 DMA 请求重映射使能 0: 不重映射(TIMER16_CH1和TIMER16_UP DMA被映射在DMA通道3) 1: 重映射 (TIMER16_CH1和TIMER16_UP DMA被映射在DMA通道4)
10	USART1_RX_DMA_R MP	USART1_RX DMA 请求重映射使能 0: 不重映射(USART1_RX DMA被映射在DMA通道3) 1: 重映射 (USART1_RX DMA被映射在DMA通道5)
9	USART1_TX_DMA_R MP	USART1_TX DMA 请求重映射使能 0: 不重映射(USART1_TX DMA被映射在DMA通道2) 1: 重映射 (USART1_TX DMA被映射在DMA通道4)
8	ADC_DMA_RMP	ADC DMA 请求重映射使能 0: 不重映射(ADC DMA被映射在DMA通道1) 1: 重映射 (ADC DMA被映射在DMA通道2)
7:2	保留	必须保持复位值
1:0	BOOT_MODE [1:0]	引导模式 (详细请参考1.4节Boot配置) Bit0对应BOOT0; bit1对应OB_BOOT1_n的值取反。 x0: 从片上闪存的主存引导启动 01: 从片上闪存的系统存储器引导启动 11: 从片上SRAM引导启动

### 1.5.2. 系统配置寄存器 2 (SYSCFG\_R2)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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															

rw

位/位域	名称	描述
------	----	----

---

31:4	保留	必须保持复位值
3:1	LCD_DECA	LCD去耦电容连接 Bit1: 去耦电容是否连接至PB2 Bit2: 去耦电容是否连接至PB12 Bit3: 去耦电容是否连接至PB0
0	保留	必须保持复位值

### 1.5.3. EXTI 源选择寄存器 1 (SYSCFG\_EXTI\_SS1)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXTI3_SS [3:0]		EXTI2_SS [3:0]		EXTI1_SS [3:0]		EXTI0_SS [3:0]								

rw                    rw                    rw                    rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI3_SS[3:0]	EXTI 3源选择 X000: PA3引脚 X001: PB3引脚 X010: PC3引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
11:8	EXTI2_SS[3:0]	EXTI 2源选择 X000: PA2引脚 X001: PB2引脚 X010: PC2引脚 X011: PD2引脚 X100: 保留 X101: 保留 X110: 保留 X111: 保留
7:4	EXTI1_SS[3:0]	EXTI 1源选择

X000: PA1引脚  
 X001: PB1引脚  
 X010: PC1引脚  
 X011: 保留  
 X100: 保留  
 X101: PF1引脚  
 X110: 保留  
 X111: 保留

3:0       EXTI0\_SS[3:0]   EXTI 0 源选择  
 X000: PA0引脚  
 X001: PB0引脚  
 X010: PC0引脚  
 X011: 保留  
 X100: 保留  
 X101: PF0引脚  
 X110: 保留  
 X111: 保留

#### 1.5.4.   EXTI 源选择寄存器 2 (SYSCFG\_EXTISS2)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI7_SS[3:0]	EXTI 7源选择 X000: PA7引脚 X001: PB7引脚 X010: PC7引脚 X011: 保留 X100: 保留 X101: PF7引脚 X110: 保留 X111: 保留

11:8	EXTI6_SS[3:0]	EXTI 6源选择
		X000: PA6引脚
		X001: PB6引脚
		X010: PC6引脚
		X011: 保留
		X100: 保留
		X101: PF6引脚
		X110: 保留
		X111: 保留
7:4	EXTI5_SS[3:0]	EXTI 5源选择
		X000: PA5引脚
		X001: PB5引脚
		X010: PC5引脚
		X011: 保留
		X100: 保留
		X101: PF5引脚
		X110: 保留
		X111: 保留
3:0	EXTI4_SS[3:0]	EXTI 4源选择
		X000: PA4引脚
		X001: PB4引脚
		X010: PC4引脚
		X011: 保留
		X100: 保留
		X101: PF4引脚
		X110: 保留
		X111: 保留

### 1.5.5. EXTI 源选择寄存器 3 (SYSCFG\_EXTISS3)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

位/位域      名称      描述

---

31:16	保留	必须保持复位值
15:12	EXTI11_SS[3:0]	EXTI 11源选择 X000: PA11引脚 X001: PB11引脚 X010: PC11引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
11:8	EXTI10_SS[3:0]	EXTI 10源选择 X000: PA10引脚 X001: PB10引脚 X010: PC10引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
7:4	EXTI9_SS[3:0]	EXTI 9源选择 X000: PA9引脚 X001: PB9引脚 X010: PC9引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
3:0	EXTI8_SS[3:0]	EXTI 8源选择 X000: PA8引脚 X001: PB8引脚 X010: PC8引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留

### 1.5.6. EXTI 源选择寄存器 4 (SYSCFG\_EXTISS4)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI15_SS[3:0]	EXTI 15源选择 X000: PA15引脚 X001: PB15引脚 X010: PC15引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
11:8	EXTI14_SS[3:0]	EXTI 14源选择 X000: PA14引脚 X001: PB14引脚 X010: PC14引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
7:4	EXTI13_SS[3:0]	EXTI 13源选择 X000: PA13引脚 X001: PB13引脚 X010: PC13引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
3:0	EXTI12_SS[3:0]	EXTI 12源选择 X000: PA12引脚 X001: PB12引脚

31:16	保留	必须保持复位值
15:12	EXTI15\_SS[3:0]	EXTI 15源选择 X000: PA15引脚 X001: PB15引脚 X010: PC15引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
11:8	EXTI14\_SS[3:0]	EXTI 14源选择 X000: PA14引脚 X001: PB14引脚 X010: PC14引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
7:4	EXTI13\_SS[3:0]	EXTI 13源选择 X000: PA13引脚 X001: PB13引脚 X010: PC13引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
3:0	EXTI12\_SS[3:0]	EXTI 12源选择 X000: PA12引脚 X001: PB12引脚

X010: PC12引脚

X011: 保留

X100: 保留

X101: 保留

X110: 保留

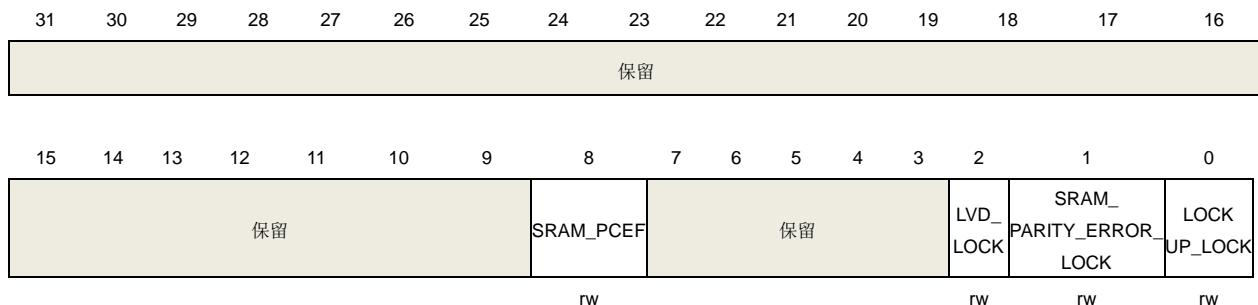
X111: 保留

### 1.5.7. 系统配置寄存器 3 (SYSCFG\_R3)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:9	保留	必须保持复位值
8	SRAM_PCEF	<p>SRAM奇偶校验错误标志</p> <p>当SRAM奇偶校验错误发生时，该位由硬件置1。该位由软件写1清零。</p> <p>0: 未检测到SRAM奇偶校验错误</p> <p>1: 检测到SRAM奇偶校验错误</p>
7:3	保留	必须保持复位值
2	LVD_LOCK	<p>LVD锁定</p> <p>该位由软件置1，在系统复位时才能清零。</p> <p>0: LVD中断从TIMER1/15/16/17的break输入端断开。PWR_CTLR寄存器的LVDE和LVDT[2:0]可以被设置</p> <p>1: LVD中断与TIMER1/15/16/17的break输入端连接。PWR_CTLR寄存器的LVDE和LVDT[2:0]仅仅可读</p>
1	SRAM_PARITY_ERROR_LOCK	<p>SRAM奇偶校验错误锁定</p> <p>该位由软件置1，在系统复位时才能清零。</p> <p>0: SRAM奇偶校验错误从TIMER1/15/16/17的break输入端断开</p> <p>1: SRAM奇偶校验错误与TIMER1/15/16/17的break输入端连接</p>
0	LOCKUP_LOCK	Cortex-M3 LOCKUP输出锁定
		该位由软件置1，在系统复位时才能清零。

0: Cortex-M3 LOCKUP输出从TIMER1/15/16/17的break输入端断开

1: Cortex-M3 LOCKUP输出与TIMER1/15/16/17的break输入端连接

## 1.6. 设备电子签名

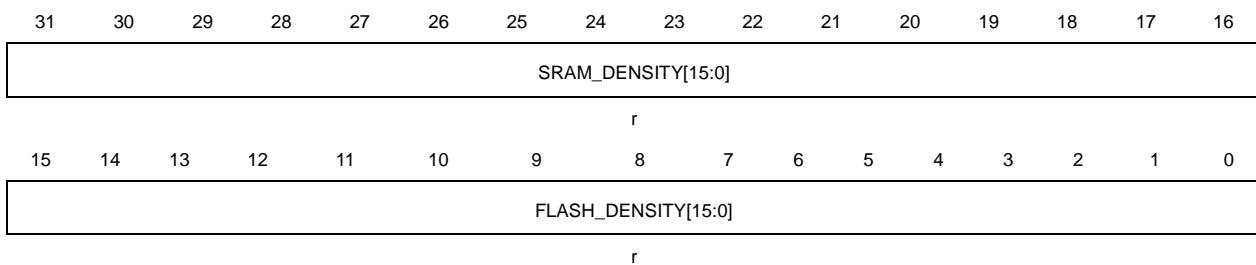
设备的电子签名中包含的存储容量信息和96位的唯一设备ID。它被存储在片上闪存的信息模块中。96位唯一设备ID对于每颗芯片而言都是唯一的。它可以用作序列号，或安全密钥的一部分，等等。

### 1.6.1. 存储密度信息

基址址: 0x1FFF F7E0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问



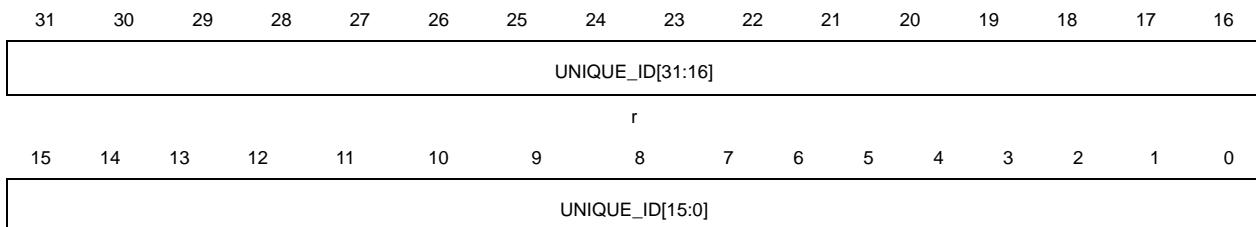
位/位域	名称	描述
31:16	SRAM_DENSITY	SRAM 容量
[15:0]		该值表明芯片的片上SRAM容量，以Kbytes为单位 例如：0x0008表示8Kbytes。
15:0	FLASH_DENSITY	Flash存储器容量
[15:0]		该值表明芯片的片上闪存容量，以Kbytes为单位 例如：0x0020表示32Kbytes。

### 1.6.2. 设备唯一 ID (96 位/位域)

基址址: 0x1FFF F7AC

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问



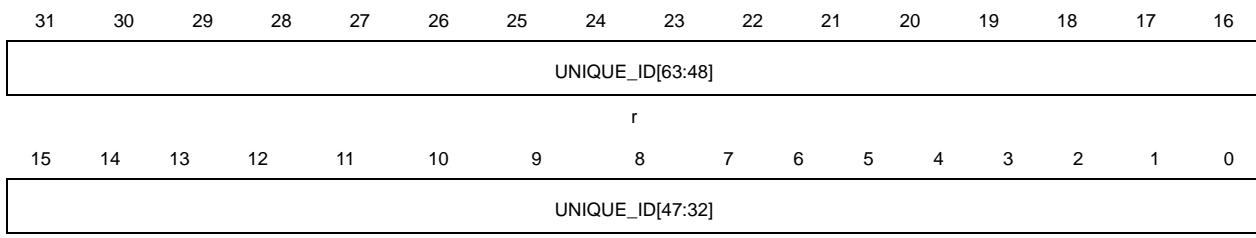
r

位/位域	名称	描述
31:0	UNIQUE_ID[31:0]	设备唯一 ID

基地址: 0x1FFF F7B0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问

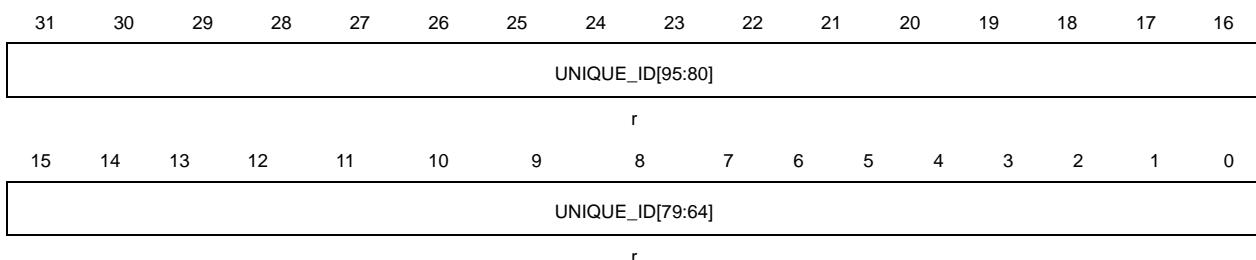


位/位域	名称	描述
31:0	UNIQUE_ID[63:32]	设备唯一 ID

基地址: 0x1FFF F7B4

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:0	UNIQUE_ID[95:64]	唯一设备 ID

## 2. 电源控制 (PWR)

### 2.1. 简介

功耗是GD32F1x0系列设备中最重要的问题之一，因此在电源控制中提供了三种省电模式，包括睡眠模式、深度睡眠模式和待机模式。这些模式使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。对于GD32F130xx和GD32F150xx产品，如下图所示设备有三个电源域，包括V<sub>DD</sub>域、1.2V域和备份域。对于GD32F170xx和GD32F190xx产品，如下图所示设备有三个电源域，包括V<sub>DD</sub>域、1.8V域和备份域。V<sub>DD</sub>域由V<sub>DD</sub>引脚直接供电。在V<sub>DD</sub>域中嵌入了一个LDO，用来产生1.2V/1.8V电压为1.2V/1.8V域供电。备份域中实现了一个电源切换器。当V<sub>DD</sub>电源开启时，备份域由V<sub>DD</sub>引脚供电；当V<sub>DD</sub>电源关闭时，电源切换器可以将备份域的电源切换到V<sub>BAT</sub>引脚，此时备份域由V<sub>BAT</sub>引脚（电池）供电。

### 2.2. 主要特性

- 三个电源域：V<sub>BAT</sub>、V<sub>DD</sub> 和1.2V电源域（GD32F130xx和GD32F150xx产品）或1.8V电源域（GD32F170xx和GD32F190xx产品）
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式
- 内部调压器提供1.2V电源（GD32F130xx和GD32F150xx产品）或1.8V电源（GD32F170xx和GD32F190xx产品）
- 提供由V<sub>BAT</sub>供电的20字节的备份寄存器，在待机模式数据也不会丢失
- 提供低电压检测器，当电压低于所设定的阈值时能发出中断或事件
- 当V<sub>DD</sub>关闭的时，由V<sub>BAT</sub>（电池）为备份域供电
- 用于存储RTC校准值的校准寄存器

### 2.3. 功能描述

下图提供了PWR及相关电源域的结构框图。

图 2-1. GD32F130xx 和 GD32F150xx 产品的电源域概览

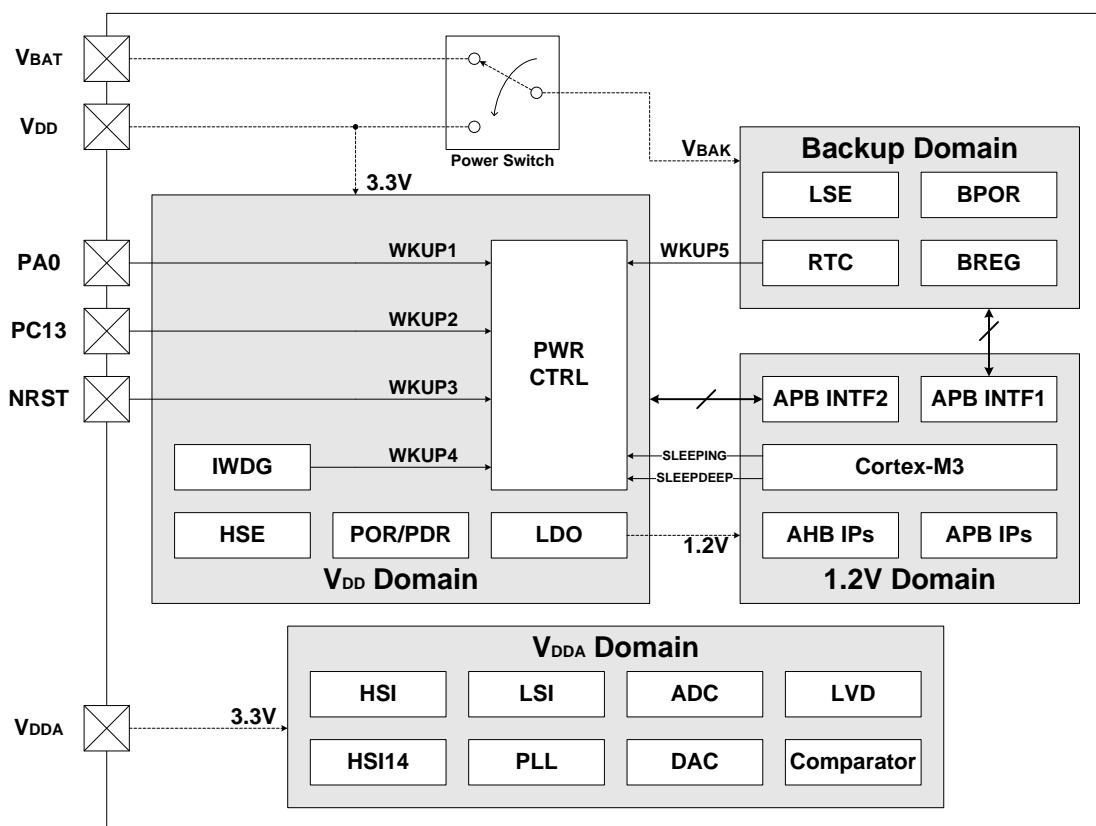
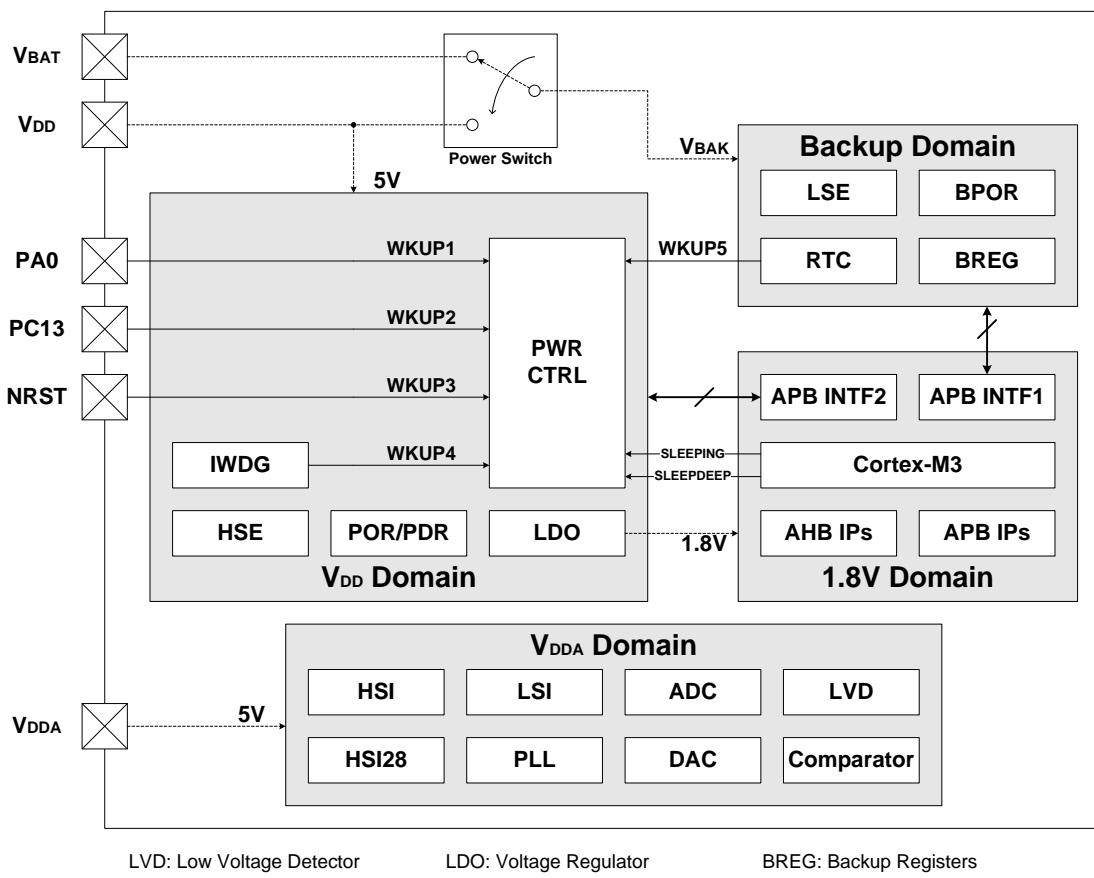


图 2-2. GD32F170xx 与 GD32F190xx 产品的电源域概览



LVD: Low Voltage Detector

LDO: Voltage Regulator

BREG: Backup Registers

POR: Power On Reset

PDR: Power Down Reset

BPOR: VBAK Power On Reset

### 2.3.1. 电池备份域

该备份域包含BPOR(备份域上电复位)、BREG(备份寄存器)、RTC(实时时钟)、LSE(低速外部晶体振荡器)，以及三个PAD，包括PC13、PC14和PC15。它可由内部电源切换器来选择VDD供电或VBAT(电池)供电。为了保留备份寄存器的内容并支持RTC正常工作，当VDD关闭时，VBAT引脚可以连接至电池或其他电源等备份源供电。电源切换器是由VDD域掉电复位电路控制的。对于没有外部电池的应用，建议将VBAT引脚通过100nF的外部陶瓷去耦电容连接到VDD引脚上。

备份域的复位源包括BPOR和备份域软件复位。在VBAK没有完全上电时，BPOR信号强制设备处于复位状态。应用软件可以通过设置RCC\_BDCR寄存器BKPRST位来触发备份域软件复位。

RTC的时钟源可以是低速内部RC振荡器(LSI)或低速外部晶体振荡器(LSE)，或高速外部晶体振荡器(HSE)时钟32分频。当VDD被关闭时，RTC只能选择LSE作时钟源。在通过WFI/WFE指令进入省电模式之前，Cortex™-M3需要通过RTC寄存器设置预期的唤醒时间并启用唤醒功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将报警并唤醒设备。RTC的配置和操作的细节将在RTC章节来描述。

备份域内提供了5个32位寄存器，多达20个字节，用于用户存储应用数据。这些寄存器是由VBAK供电的，即使VDD电源关闭，这些寄存器也不会掉电。备份寄存器的复位源只有被备份域上电复位和备份域软件复位。

当备份域由 $V_{DD}$ 电源供电时 ( $V_{BAK}$ 连接至 $V_{DD}$ )，以下功能可用：

- PC13可以作为通用I/O口、TAMPER引脚、RTC时钟校准和RTC闹钟或秒输出。(参见第20章：实时时钟RTC)
- PC14和PC15可以作为通用I/O口或LSE晶振引脚。

当备份域由 $V_{BAT}$ 电源供电时 ( $V_{BAK}$ 连接至 $V_{BAT}$ )，以下功能可用：

- PC13可以作为TAMPER引脚和RTC闹钟或秒输出。(参见第20章：实时时钟RTC)
- PC14和PC15仅可作为LSE晶振引脚。

**注意：**由于PC13至PC15是通过电源切换器供电的，电源切换器仅可通过小电流，因此当PC13至PC15的GPIO口在输出模式时，其工作的速度不能超过2MHz。

### 2.3.2. $V_{DD}/V_{DDA}$ 电源域

大多数模拟模块都位于 $V_{DDA}$ 电源域，包括HSI(高速内部RC振荡器)、在GD32F130xx和GD32F150xx产品中的HSI14(14MHz频率的高速内部RC振荡器)或在GD32F170xx和GD32F190xx产品中的HSI28(28MHz频率的高速内部RC振荡器)、LSI(低速内部RC振荡器)、PLL(锁相环)、ADC(模数转换器)、DAC(数模转换器)、LVD(低电压检测器)和比较器等。此外，LDO(电压调整器)，POR/PDR(上电/掉电复位)，HSE(高速外部晶体振荡器)，IWDG(独立看门狗)，功耗控制逻辑，除PC13、PC14和PC15之外的所有PAD，都是在 $V_{DD}$ 域中实现的。LDO在GD32F130xx和GD32F150xx产品中用来给1.2V域供电，在GD32F170xx和GD32F190xx产品中用来给1.8V域供电。通常，数字电路是由 $V_{DD}$ 供电的，而大多数模拟电路是由 $V_{DDA}$ 供电的。将模拟电路电源( $V_{DDA}$ )从 $V_{DD}$ 独立出来的好处是，可以让模拟电路达到更好的性能，特别是提高了ADC和DAC的转换精度。 $V_{DDA}$ 的电压可以等于或高于 $V_{DD}$ 。 $V_{DDA}$ 可以从外部连接到 $V_{DD}$ ，但必须通过外部滤波电路连接，以避免 $V_{DDA}$ 上的噪声。否则，如果 $V_{DDA}$ 和 $V_{DD}$ 不同， $V_{DDA}$ 电压必须始终高于 $V_{DD}$ 。在这种情况下，为了安全起见，可在 $V_{DD}$ 和 $V_{DDA}$ 之间使用外部肖特基二极管。

LVD的功能是检测 $V_{DD}$ 或 $V_{DDA}$ 供电电压是否低于低电压检测阈值。该功能由电源控制寄存器(PWR\_CTLR)进行配置。LVDE位控制LVD功能是否开启，而LVDT[2:0]位用来选择低电压检测阈值，阈值范围在GD32F130xx和GD32F150xx产品中为2.2V到2.9V之间，在GD32F170xx和GD32F190xx产品中为2.4V到4.5V之间。低电压状态标志(LVDF)位于电源状态寄存器(PWR\_STR)。另外，低电压检测事件也被连接到了EXTI的第16线，用户可以通过配置EXTI的第16线产生相应的中断。

POR/PDR电路的功能是检测 $V_{DD}/V_{DDA}$ 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。下图显示了供电电压和电源复位信号之间的关系。 $V_{POR}$ 表示上电复位的阈值电压， $V_{PDR}$ 表示掉电复位的阈值电压。在GD32F130xx和GD32F150xx产品中， $V_{PDR}$ 是可配置的，通过RCC\_PDR\_S寄存器的PDR\_S位可以从两个值中选择一个作为 $V_{PDR}$ (参见4.3.20章节RCC寄存器)。当选择较低的值时，强烈建议不要对片上闪存做编程或擦除操作，因为这些操作在电压低至接近该阈值时可能会失败。

图 2-3. 上电复位/掉电复位波形图

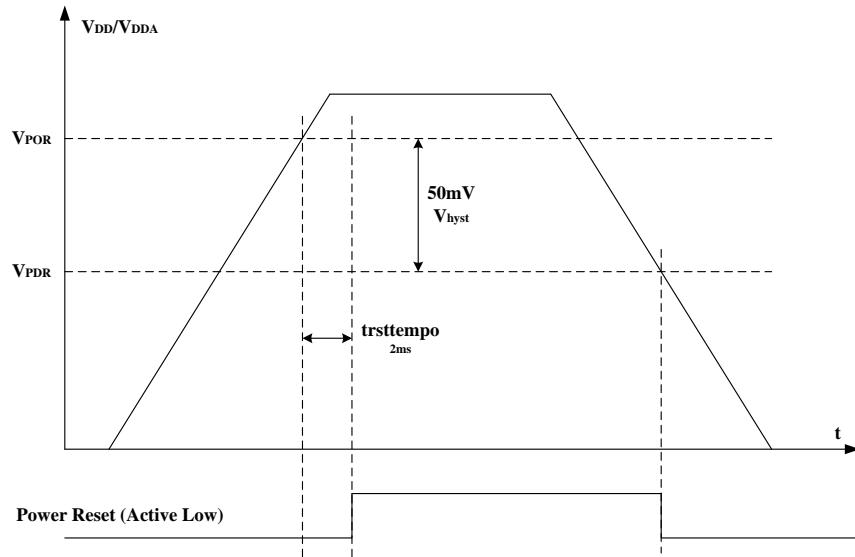
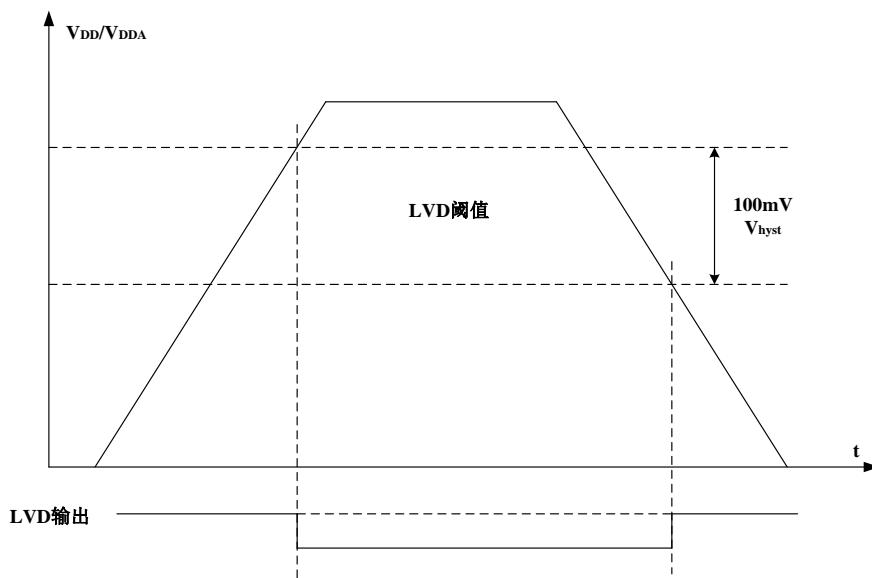


图2-4. LVD阈值波形图



### 2.3.3. 1.2V 电源域（适用于 GD32F130xx 和 GD32F150xx 产品）

Cortex™-M3处理器，AHB/APB外设等主要功能均位于1.2V电源域。一旦1.2V电压上电，POR将在1.2V电源域产生复位序列。

### 2.3.4. 1.8V 电源域（适用于 GD32F170xx 和 GD32F190xx 产品）

Cortex™-M3处理器，AHB/APB外设等主要功能均位于1.8V电源域。一旦1.8V电压上电，POR将在1.8V电源域产生复位序列。

## 2.3.5. 省电模式

系统复位或电源复位后，GD32F1x0的所有电源域全部处于供电状态。用户可以通过减慢系统时钟(HCLK, PCLK1, PCLK2)或关闭未使用的外设的时钟来实现较低的功耗。此外，三种省电模式（睡眠模式、深度睡眠模式和待机模式）可以实现更低的功耗。

### 睡眠模式

睡眠模式与Cortex™-M3的休眠模式相对应。休眠模式只关闭Cortex™-M3的时钟。如需进入睡眠模式，只要清除Cortex™-M3系统控制寄存器中的SLEEPDEEP位，并执行一条WFI或WFE指令即可。如果睡眠模式是通过执行WFI指令进入的，任何中断都可以唤醒系统。如果睡眠模式是通过执行WFE指令进入的，任何唤醒事件都可以唤醒系统。由于无需在进入或退出中断上浪费时间，该模式所需的唤醒时间最短。

根据Cortex™-M3中SCR(系统控制寄存器)的SLEEPONEXIT位，有两种睡眠进入机制可选：

- **Sleep-now:** 如果SLEEPONEXIT位被清零，一旦执行WFI或WFE指令，MCU立即进入睡眠模式。
- **Sleep-on-exit:** 如果SLEEPONEXIT位被置位，当系统从最低优先级的中断处理程序离开后，MCU立即进入睡眠模式。

### 深度睡眠模式

深度睡眠模式与Cortex™-M3的深度睡眠模式相对应。深度睡眠模式会关闭1.2V域(GD32F130xx和GD32F150xx产品)或1.8V域(GD32F170xx和GD32F190xx产品)中的所有时钟以及包括HSI、HSI14(GD32F130xx和GD32F150xx产品)或HSI28(GD32F170xx和GD32F190xx产品)、HSE和PLL的所有时钟生成电路。PWR\_CTLR寄存器的LDOLP位可以控制LDO工作在正常模式或低功耗模式。进入深度睡眠模式的步骤是，先将Cortex™-M3系统控制寄存器的SLEEPDEEP位置1，再清除PWR\_CTLR寄存器的SDBM位，然后执行WFI或WFE指令。来自EXTI的任何中断或唤醒事件可以将系统从深度睡眠模式中唤醒。刚退出深度睡眠模式时，HSI被选中作为系统时钟。请注意，如果LDO工作在低功耗模式，退出深度睡眠模式需要花费更多的时间。

**注意：**为了顺利进入深度睡眠模式，所有EXTI线上的挂起状态(在EXTI\_PD寄存器中)和RTC闹钟标志必须被复位。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

### 待机模式

待机模式是基于Cortex™-M3的深度睡眠模式实现的。待机模式会关闭整个1.2V域(GD32F130xx和GD32F150xx产品)或1.8V域(GD32F170xx和GD32F190xx产品)的供电，同时LDO和包括HSI、HSI14(GD32F130xx和GD32F150xx产品)或HSI28(GD32F170xx和GD32F190xx产品)、HSE和PLL的所有时钟生成电路也会被关闭。进入待机模式的步骤是，先将Cortex™-M3系统控制寄存器的SLEEPDEEP位置1，再将PWR\_CTLR寄存器的SDBM位置1，再清除PWR\_STR寄存器的WUF位，然后执行WFI或WFE指令。待机模式有五个唤醒源，包括来自NRST引脚的外部复位、RTC报警、IWDG复位、WKUP1或WKUP2引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM和寄存器

(除了备份寄存器)的内容都会丢失。退出待机模式时，会发生上电复位，复位之后Cortex™-M3将从0x00000000地址开始执行指令代码。

**表 2-1. 节电模式总结**

模式	睡眠	深度睡眠	待机
<b>Description</b>	仅关闭 CPU 时钟	1. 关闭 1.2V 电源域 (GD32F130xx 和 GD32F150xx 产品) 1.8V 电源域 (GD32F170xx 和 GD32F190xx 产品) 的所有时钟 2. 关闭 HSI、HSI14 (GD32F130xx 和 GD32F150xx 产品) 或 HSI28 (GD32F170xx 和 GD32F190xx 产 品)、HSE 和 PLL	1. 关闭 1.2V 电源域 (GD32F130xx 和 GD32F150xx 产 品) 1.8V 电源域 (GD32F170xx 和 GD32F190xx 产 品) 的供电 2. 关闭 HSI、HSI14 (GD32F130xx 和 GD32F150xx 产 品) 或 HSI28 (GD32F170xx 和 GD32F190xx 产 品)、HSE 和 PLL
<b>LDO 状态</b>	开启	开启或低功耗模式	关闭
<b>配置</b>	SLEEPDEEP = 0	SLEEPDEEP = 1 SDBM = 0	SLEEPDEEP = 1 SDBM = 1, WUFR=1
<b>进入指令</b>	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
<b>唤醒</b>	若通过 WFI 进入，则任何中断均可唤醒；若通过 WFE 进入，则任何事件均可唤醒	来自 EXTI 的任何中断或事件	1. NRST 引脚 2. RTC 报警 3. IWDG 复位 4. WKUP1 引脚 5. WKUP2 引脚
<b>唤醒延迟</b>	无	HSI 唤醒时间 如果 LDO 处于低功耗模式的话，需增加 LDO 唤醒时间	上电序列

## 2.4. PWR 寄存器

### 2.4.1. 电源控制寄存器 (PWR\_CTLR)

#### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x00

复位值: 0x0000 0000 (从待机模式唤醒后复位)

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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				BKPWE	LVDT[2:0]			LVDE	SBFR	WUFR	SDBM	LDOLP			

位/位域	名称	描述
31:9	保留	必须保持复位值
8	BKPWE	备份域写使能 0: 禁止对备份域寄存器的写操作 1: 允许对备份域寄存器的写操作 复位之后, 任何对备份域寄存器的写操作都将被禁止。如需对备份域寄存器做写操作, 需先将该位置1。
7:5	LVDT[2:0]	低电压检测器阈值 000: 2.2V 001: 2.3V 010: 2.4V 011: 2.5V 100: 2.6V 101: 2.7V 110: 2.8V 111: 2.9V
4	LVDE	低电压检测器使能 0: 关闭低电压检测器 1: 开启低电压检测器  注意: 当SYSCFG_R3寄存器里的LVD_LOCK位被置1时, LVDE和LVDT[1:0]仅可读。
3	SBFR	待机标志复位 0: 无影响 1: 复位待机标志 读该位, 始终返回0。
2	WUFR	唤醒标志复位 0: 无影响 1: 复位唤醒标志 读该位, 始终返回0。
1	SDBM	待机模式 0: 当Cortex™-M3 进入深度睡眠模式时, 系统进入深度睡眠模式 1: 当Cortex™-M3 进入深度睡眠模式时, 系统进入待机模式

0	LDOLP	LDO低功耗模式 0: 当系统进入深度睡眠模式时, LDO仍正常工作 1: 当系统进入深度睡眠模式时, LDO进入低功耗模式  <b>注意:</b> 在深度睡眠模式下, 个别外设可能会开启HSI时钟来做一些工作。在这种情况下, 如果LDO正处于低功耗模式, LDO会自动从低功耗模式切换到正常工作模式, 并保持正常工作模式, 直到外设工作完毕。
---	-------	--

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x00

复位值: 0x0000 0000 (从待机模式唤醒后复位)

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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				BKPWE	LVDT[2:0]			LVDE	SBFR	WUFR	SDBM	LDOLP	rw		

位/位域	名称	描述
31:9	保留	必须保持复位值
8	BKPWE	备份域写使能 0: 禁止对备份域寄存器的写操作 1: 允许对备份域寄存器的写操作  复位之后, 任何对备份域寄存器的写操作都将被禁止。如需对备份域寄存器做写操作, 需先将该位置1。
7:5	LVDT[2:0]	低电压检测器阈值 000: 2.4V 001: 2.7V 010: 3.0V 011: 3.3V 100: 3.6V 101: 3.9V 110: 4.2V 111: 4.5V
4	LVDE	低电压检测器使能 0: 关闭低电压检测器 1: 开启低电压检测器  <b>注意:</b> 当SYSCFG_R2寄存器里的LVD_LOCK位被置1时, LVDE和LVDT[1:0]仅可读。

3	SBFR	待机标志复位 0: 无影响 1: 复位待机标志 读该位, 始终返回0。
2	WUFR	唤醒标志复位 0: 无影响 1: 复位唤醒标志 读该位, 始终返回0。
1	SDBM	待机模式 0: 当Cortex™-M3 进入深度睡眠模式时, 系统进入深度睡眠模式 1: 当Cortex™-M3 进入深度睡眠模式时, 系统进入待机模式
0	LDOLP	LDO低功耗模式 0: 当系统进入深度睡眠模式时, LDO仍正常工作 1: 当系统进入深度睡眠模式时, LDO进入低功耗模式  注意: 在深度睡眠模式下, 个别外设可能会开启HSI时钟来做一些工作。在这种情况下, 如果LDO正处于低功耗模式, LDO会自动从低功耗模式切换到正常工作模式, 并保持正常工作模式, 直到外设工作完毕。

#### 2.4.2. 电源状态寄存器 (PWR\_STR)

地址偏移: 0x04

复位值: 0x0000 0000 (从待机模式唤醒后不复位)

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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				WUPE2		WUPE1		保留				LVDF		SBF	
rw								rw							

位/位域	名称	描述
31:10	保留	必须保持复位值
9	WUPE2	WKUP2引脚唤醒使能 0: 关闭WKUP2引脚唤醒功能 1: 开启WKUP2引脚唤醒功能  如果WUPE2在进入省电模式之前置1, WKUP2引脚的上升沿会将系统从省电模式唤醒。由于WKUP2引脚为高电平有效, 当WUPE2为高时, WKUP2引脚将设置为输入下拉模式。
8	WUPE1	WKUP1 引脚唤醒使能

0: 关闭WKUP1引脚唤醒功能

1: 开启WKUP1引脚唤醒功能

如果WUPE1在进入省电模式之前置1, WKUP1引脚的上升沿会将系统从省电模式唤醒。由于WKUP1引脚为高电平有效, 当WUPE1为高时, WKUP1引脚将设置为输入下拉模式。

7:3 保留 必须保持复位值

2 LVDF 低电压状态标志

0: 低电压事件没出现 ( $V_{DD}$ 高于设定的LVD阈值)

1: 低电压事件出现 ( $V_{DD}$ 等于或低于LVD阈值)

**注意:** LVD功能在待机模式下被禁用。

1 SBF 待机标志

0: 设备没进入过待机模式

1: 设备曾进入过待机模式

该位只能由POR/PDR或通过设置PWR\_CTLR寄存器的SBFR位来清零。

0 WUF 唤醒标志

0: 没有收到唤醒事件

1: 收到来自WKUP引脚或RTC报警等的唤醒事件

该位只能由POR/PDR或通过设置PWR\_CTLR寄存器的SBFR位来清零。

### 3. 闪存存储器控制器 (FMC)

#### 3.1. 简介

闪存存储器控制器 (FMC) , 提供了所有需要的片上闪存存储功能。在32K字节空间内, CPU执行指令零等待。FMC也提供了页擦除、整片擦除, 以及闪存存储区的32位字或16位半字编程。

#### 3.2. 主要特性

##### GD32F130xx和GD32F150xx产品:

- 高达64K字节片上闪存存储器可用于存储指令或数据
- 32K字节空间内CPU执行指令零等待
- 从闪存取32K ~ 64K地址内字节数据有比较长的延迟
- 3K字节启动器信息块
- 16字节可选字节块用于用户需求
- 每页1K字节
- 32位字或16位半字编程, 支持页擦除和整片擦除
- 有闪存读保护功能, 阻止非法代码或数据进入
- 有页擦除和页编程保护功能, 阻止意外操作

##### GD32F170xx和GD32F190xx产品:

- 高达128K字节片上闪存存储器可用于存储指令或数据
- 32K字节空间内CPU执行指令零等待
- 从闪存取32K ~ 128K地址内字节数据有比较长的延迟
- 3K字节启动器信息块
- 16字节可选字节块用于用户需求
- 每页1K字节
- 32位字或16位半字编程, 支持页擦除和整片擦除
- 有闪存读保护功能, 阻止非法代码或数据进入
- 有页擦除和页编程保护功能, 阻止意外操作

### 3.3. 功能描述

#### 3.3.1. 闪存存储器结构

**适用于GD32F130xx和GD32F150xx产品**

闪存存储器包括多达64K字节的主闪存，按64页每页1K字节分块，还有一个3K字节的信息块用于启动器。主闪存存储器的64页每页都可以单独擦除。闪存存储器的基址和大小见下表。

**表 3-1. 闪存存储器的基址和大小**

闪存块	名称	地址	大小(字节)
主存储闪存块 主存储块	页 0	0x0800 0000 - 0x0800 03FF	1KB
	页 1	0x0800 0400 - 0x0800 07FF	1KB
	页 2	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	页 63	0x0800 FC00 - 0x0800 FFFF	1KB
	信息块	启动器	3KB
可选字节块	可选字节	0x1FFF F800 - 0x1FFF F80F	16B

**注意：**信息块存储了启动器信息，不能被用户编程或擦除

**适用于GD32F170xx和GD32F190xx产品**

闪存存储器包括多达128K字节的主闪存，按128页每页1K字节分块，还有一个3K字节的信息块用于启动器。主闪存存储器的128页每页都可以单独擦除。闪存存储器的基址和大小见下表。

**表 3-2. 闪存存储器的基址和大小**

闪存块	名称	地址	大小(字节)
主存储闪存块 主存储块	页 0	0x0800 0000 - 0x0800 03FF	1KB
	页 1	0x0800 0400 - 0x0800 07FF	1KB
	页 2	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	页 128	0x0801 FC00 - 0x0801 FFFF	1KB
	信息块	启动器	3KB
可选字节块	可选字节	0x1FFF F800 - 0x1FFF F80F	16B

**注意：**信息块存储了启动器信息，不能被用户编程或擦除

### 3.3.2. 读保护

Flash可以像普遍存储空间一样直接寻址访问。对flash的取指令和取数据都是通过CPU的IBUS或DBUS来完成。

### 3.3.3. FMC\_CMR 寄存器解锁

复位后，FMC\_CMR寄存器进入锁定状态，LK位置为1（OPTR位除外，此位用于重加载可选字节）。通过先后向FMC\_UKEYR寄存器写入0x45670123和0xCDEF89AB，可以使得FMC\_CMР解锁。两次写操作后，FMC\_CMР寄存器的LK位被硬件清0。可以通过软件设置FMC\_CMР寄存器的LK位为1再次锁定FMC\_CMР寄存器。任何对FMC\_UKEYR寄存器的错误操作都会将LK位置1，从而锁FMC\_CMР寄存器，引发一个总线错误。

FMC\_CMР寄存器的OBPG位和OBER位可以被FMC\_OBKEYR寄存器锁定。解锁过程也是两次写操作，向FMC\_UKEYR寄存器先后写入0x45670123和0xCDEF89AB，然后将FMC\_CMР寄存器的OBWE位置1。软件可以将FMC\_CMР的OBWE位清0来锁定FMC\_CMР的OBPG位和OBER位。

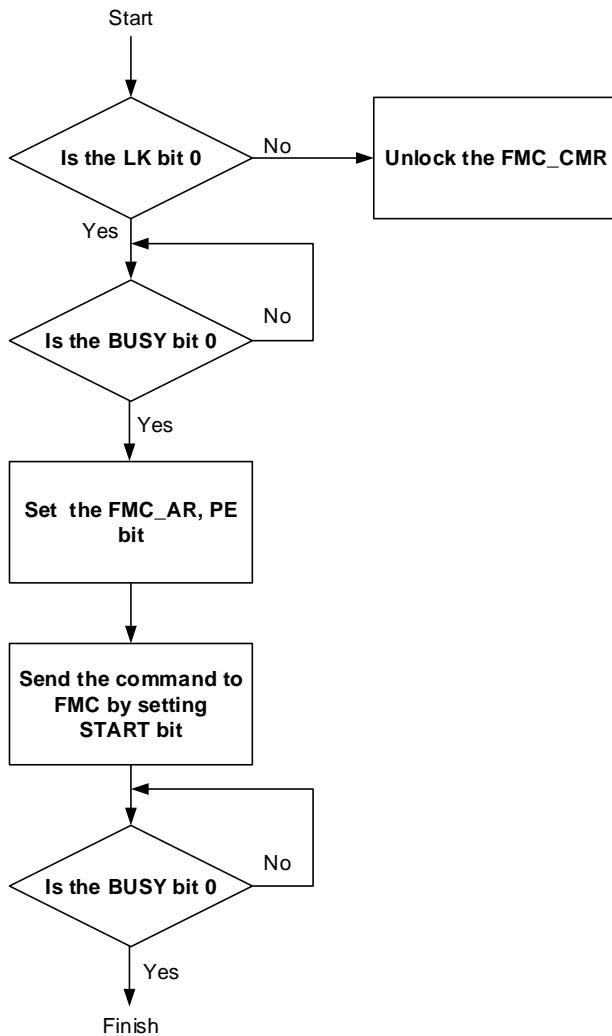
### 3.3.4. 页擦除

FMC的页擦除功能使一个主闪存存储页的内容初始化为高电平。每一页都可以被独立擦除，不影响其他页内容。FMC擦除页步骤如下。

- 确保FMC\_CMР寄存器不处于锁定状态。
- 检查FMC\_CSR寄存器的BUSY位来判定没有闪存存储器是否处于运行状态。
- 当BUSY为0时，写页地址到FMC\_AR寄存器。
- 写页擦除命令到FMC\_CMР寄存器的PE位（置1）。
- 通过将FMC\_CMР寄存器的START位置1来发送页擦除命令到FMC。
- 检查FMC\_CMР寄存器的BUSY位来判断擦除指令是否执行完毕。
- 如果需要，使用DBUS读并验证该页。

当页擦除成功执行，FMC\_CSR寄存器的ENDF位和FMC\_CMР寄存器的ENDIE位被置1，FMC从而将触发一个中断。当目标擦除页被用来取指令或访问数据时，软件可能失去控制，相应的擦除操作无效，但FMC将不提供任何通知，因此确保目标擦除页地址的正确性是很有必要的。如果FMC\_CMР寄存器的ERIE位被置1，FMC将触发一次闪存操作错误中断。软件可以检查FMC\_CSR寄存器的PGEF位来监测中断处理器的状况。FMC\_CSR寄存器的ENDF位预示操作的结束。下图显示了页擦除操作流程。

图3-1. 页擦除操作流程



### 3.3.5. 整片擦除

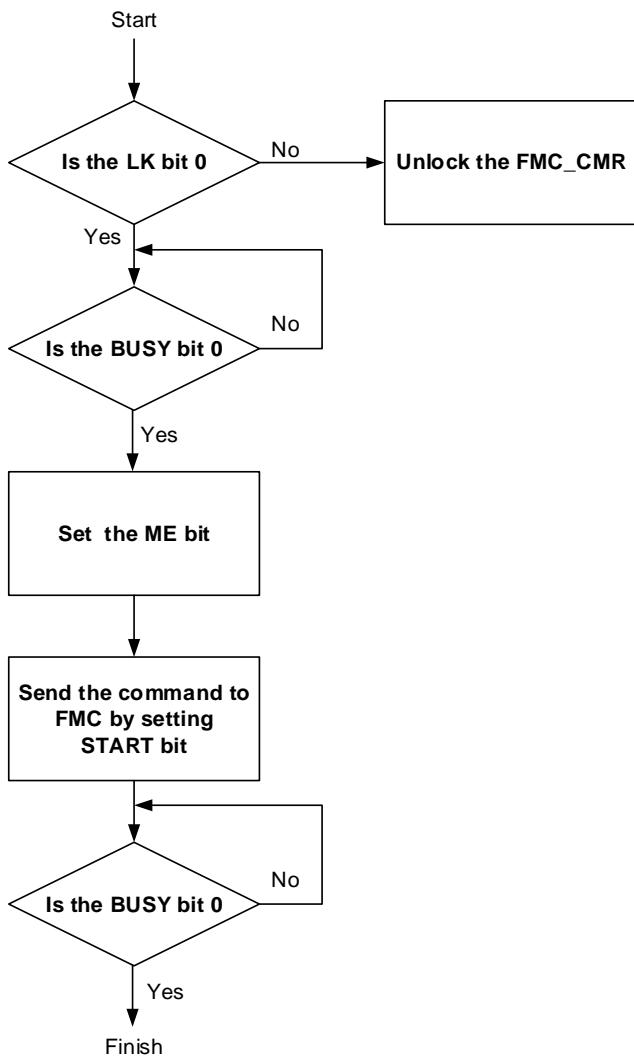
FMC提供了整片擦除功能可以初始化主存储块区内容。具体步骤如下：

- 确保FMC\_CMR寄存器不处于锁定状态。
- 检查FMC\_CSR寄存器的BUSY位来判定没有闪存存储器是否处于运行状态。
- 当BUSY为0时，写整片擦除命令到FMC\_CMR寄存器的ME位（置1）。
- 通过将FMC\_CMR寄存器的START位置1来发送整片擦除命令到FMC。
- 检查FMC\_CMR寄存器的BUSY位来判断擦除指令是否执行完毕。
- 如果需要，使用DBUS读取该页进行验证。

当整片擦除成功执行，FMC\_CSR寄存器的ENDF位和FMC\_CMR寄存器的ENDIE位被置1，FMC从而将触发一个中断。由于所有的闪存数据都将被复位为0xFFFF\_FFFF，可以通过运行在SRAM中的程序或使用调试工具直接访问FMC寄存器来实现整片擦除操作。FMC\_CSR寄存

器的ENDF位预示操作的结束。下图显示了整片擦除操作流程。

图3-2. 整片擦除操作流程



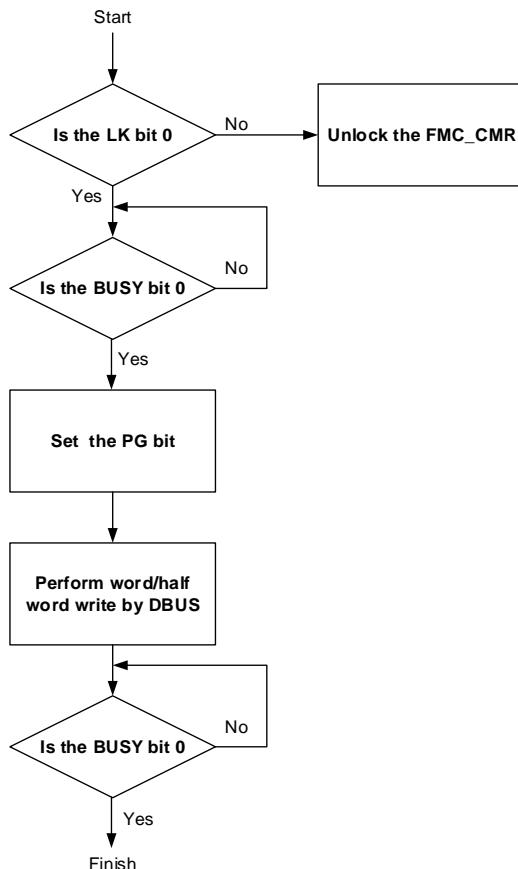
### 3.3.6. 主存储块编程

FMC提供了一个32位字/16位半字编程功能，用来修改闪存主存储块内容。以下步骤显示了字编程操作寄存器过程。

- 确保FMC\_CMCR寄存器不处于锁定状态。
- 检查FMC\_CSR寄存器的BUSY位来判定没有闪存存储器是否处于运行状态。
- 当BUSY为0时，写字编程命令到FMC\_CMCR寄存器的PG位。
- DBUS写一个32位字/16位半字到预期地址。
- 检查FMC\_CMCR寄存器的BUSY位来判断擦除指令是否执行完毕。
- DBUS访问闪存存储器验证操作结果。

当主存储块编程成功执行，FMC\_CSR寄存器的ENDF位和FMC\_CMRR寄存器的ENDIE位被置1，FMC从而将触发一个中断。执行字/半字编程操作需要检查目的地址是否已经擦除。如果地址没有被擦除，PGEF位将被置1，该页上的编程操作无效（当编程内容为0x0时，目的地址没有正常擦除，也可以正常编程）。如果FMC\_CMRR寄存器的ERIE位被置1，FMC将触发一次闪存操作错误中断。软件可以检查FMC\_CSR寄存器的PGEF位来监测中断处理器的状况。FMC\_CSR寄存器的ENDF位预示操作的结束。（主存储块编程操作的起始地址是0x08000000）下图显示了主存储块编程操作流程。

图3-3. 主存储块编程操作流程



### 3.3.7. 可选字节块擦除

FMC提供了一个擦除功能用来初始化闪存中的可选字节块。下面的步骤显示了可选字节块擦除过程。

- 确保FMC\_CMRR寄存器不处于锁定状态。
- FMC\_CMRR寄存器的OBWE位置1。
- 检查FMC\_CSR寄存器的BUSY位来判定没有闪存存储器是否处于运行状态。
- 写可选字节擦除命令到FMC\_CMRR寄存器的OBER位。
- 通过将FMC\_CMRR寄存器的START位置1来发送可选字节擦除命令到FMC。

- 检查FMC\_CMR寄存器的BUSY位来判断擦除指令是否执行完毕。
- DBUS访问闪存存储器验证操作结果。

当可选字节擦除成功执行，FMC\_CSR寄存器的ENDF位和FMC\_CMR寄存器的ENDIE位被置1，FMC从而将触发一个中断。FMC\_CSR寄存器的ENDF位置1预示操作的结束。

### 3.3.8. 可选字节块编程

FMC提供了一个32位字/16位半字编程功能，用来修改可选字节块内容。下面的步骤显示了可选字节编程操作过程。

- 确保FMC\_CMR寄存器不处于锁定状态
- FMC\_CMR寄存器的OBWE位置1。
- 检查FMC\_CSR寄存器的BUSY位来判定没有闪存存储器是否处于运行状态。
- 写可选字节块编程命令到FMC\_CMR寄存器的OBPG位。
- DBUS写一个32位字/16位半字到预期地址。
- 检查FMC\_CMR寄存器的BUSY位来判断擦除指令是否执行完毕。
- DBUS访问闪存存储器验证操作结果。

当可选字节擦除成功执行，FMC\_CSR寄存器的ENDF位和FMC\_CMR寄存器的ENDIE位被置1，FMC从而将触发一个中断。执行字/半字编程操作需要检查目的地址是否已经擦除。如果地址没有被擦除，PGEF位将被置1，该页上的编程操作无效（当编程内容为0x0时，目的地址没有正常擦除，也可以正常编程）。FMC\_CSR寄存器的ENDF位预示操作的结束。

### 3.3.9. 可选字节块说明

每次系统复位或将FMC\_CMR寄存器的OPTR位置1时，闪存存储器的可选字节块重加载到FMC\_OPTR和FMC\_WPR寄存器，可选字节工作。可选字节的补码和可选字节相反。当可选字节重装载，如果可选字节的补码和可选字节不匹配，FMC\_OPTR寄存器的OBER位将被置1，可选字节被设置为0xFF。可选字节详情见下表。

表 3-3. 可选字节

地址	名称	说明
0x1fff f800	OB_RDPT	可选字节安全保护值 0xA5: 无保护 除了0xA5或0xCC的任何值: 低级保护 0xCC: 高级保护
0x1fff f801	OB_RDPT_N	OB_RDPT补码值
0x1fff f802	OB_USER	用户定义的可选字节 [7]: 保留 [6]: OB_SRAM_PARITY_CHECK 0: 使能SRAM奇偶校验

地址	名称	说明
		1: 禁止SRAM奇偶校验 [5]: OB_VDDA_VISOR 0: 禁止VDDA监视器 1: 使能VDDA监视器 [4]: OB_BOOT1_n 0: BOOT1位是1 1: BOOT1位是0 [3]: 保留 [2]: OB_STDBY_RSTn 0: 设置待机模式时产生复位而不是进入待机模式 1: 设置待机模式时进入待机模式而不产生复位 [1]: OB_DEEPSLEEP_RSTn 0: 设置深度睡眠模式时产生复位而不进入深度睡眠模式 1: 设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]: OB_WDG_SW 0: 硬件自动设置窗口看门狗 1: 软件设置窗口看门狗
0x1fff f803	OB_USER_N	OB_USER补码值
0x1fff f804	OB_DATA[7:0]	用户定义数据位7到0
0x1fff f805	OB_DATA_N[7:0]	OB_DATA补码值的7到0位
0x1fff f806	OB_DATA[15:8]	用户定义数据位15到8
0x1fff f807	OB_DATA_N[15:8]	OB_DATA补码值的15到8位
0x1fff f808	OB_WP[7:0]	页擦除/编程保护位的7到0位
0x1fff f809	OB_WP_N[7:0]	OB_WP补码值的7到0位
0x1fff f80a	OB_WP[15:8]	页擦除/编程的保护位15到8位
0x1fff f80b	OB_WP_N[15:8]	OB_WP补码值的15到8位

### 3.3.10. 页擦除/编程保护

FMC的页擦除/编程保护功可以阻止对闪存存储器的意外操作。当FMC对被保护的页进行页擦除或编程操作时，操作本身无效且FMC\_CSR寄存器的WPEF位将被置1。如果WPEF位被置1且ERIE位也被置1来使能相应的中断，FMC将触发闪存操作错误中断来引起CPU重视。配置可选字节的OB\_WP[15:0]位为0可以单独使能每个页保护功能。如果在可选字节区域执行了页擦除操作，所有的闪存存储器页保护功能都将被禁止。当对可选字节的OB\_WP位置1或清0时，需要将FMC\_CMR寄存器的OPTR位置1或系统复位来重加载OB\_WP位。下表显示了将OB\_WP[15:0]位置0使能了哪些页保护。

表 3-4. OB\_WP 位对应页保护

OB_WP bit	被保护的页
OB_WP[0]	页0 ~ 页3
OB_WP[1]	页4 ~ 页7

OB_WP bit	被保护的页
OB_WP[2]	页8 ~ 页11
.	.
OB_WP[14]	页56 ~ 页59
OB_WP[15]	页60 ~ 页64

### 3.3.11. 安全保护

FMC提供了一个安全保护功能来阻止非法代码或数据进入闪存存储器，此功能可以很好地保护软件和固件免受非法用户的操作，有三个保护级别。

无保护：当将OB\_RDPT字节和它的补码设置为0xA55A，不执行保护。主存储块和可选字节块可以被所有操作进入。

低级别保护：当设置OB\_RDPT字节和它的补码值为任何值除了0xA55A或0xCC33，执行低级别保护。主闪存仅仅能被使用者的代码进入。在调试模式下，从SRAM中启动或从启动加载器启动，所有对主存储块的操作都被禁止。如果在调试模式下，从SRAM启动或从启动器启动时，读主存储块将产生总线错误。如果在调试模式下，从SRAM启动或从启动器启动时，对主存储块编程或擦除操作，FMC\_CSR寄存器的PGEF位将被置1。低级别保护下，可选择字节块可以被任意操作进入。如果将OB\_RDPT字节和它的补码设置为0xA55A，进入无保护级别，将执行主存储块全片擦除操作。

高级别保护：当设置OB\_RDPT字节和它的补码值为0xCC33，进入高级别保护，从SRAM启动或从启动器启动都被禁止。所有的操作来源于用户代码。主存储块可以执行所有操作。可选字节块不能被擦除，OB\_RDPT字节和它的补码值不能被重编程。因此，如果设置保护级别为高，不能转换为低级保护和无保护。

## 3.4. FMC 寄存器

### 3.4.1. 闪存保留寄存器 (FMC\_RESR)

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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																WSCNT[2:0]

rw

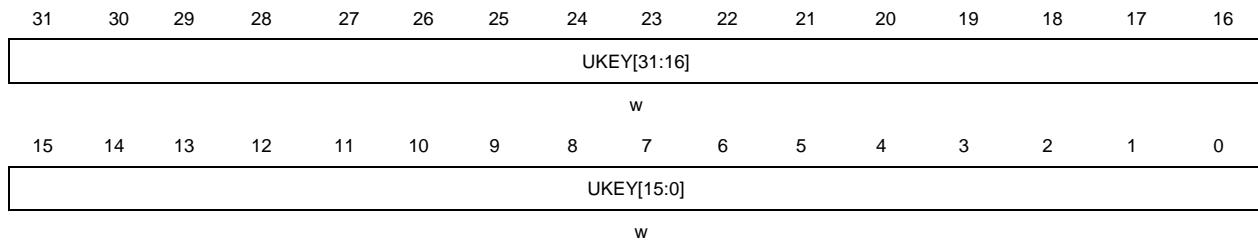
位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	WSCNT[2:0]	等待状态计数寄存器 硬件置1和清0。WSEN位被置1时WSCNT位有效。 000: 不增加等待状态 001: 增加1个等待状态 010: 增加2个等待状态 011 ~ 111: 保留

### 3.4.2. 闪存解锁寄存器 (FMC\_UKEYR)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



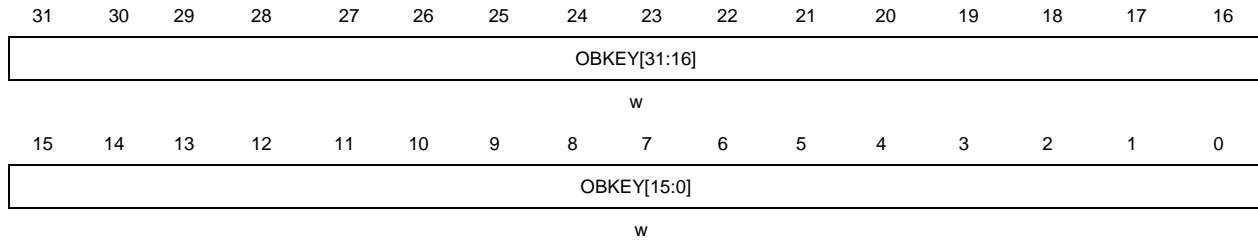
位/位域	名称	描述
31:0	UKEY[31:0]	FMC_CMR解锁寄存器 这些位仅能被软件写 写解锁值到UKEY[31:0]可以解锁FMC_CMR寄存器

### 3.4.3. 闪存可选字节块解锁寄存器 (FMC\_OBKEYR)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:0	OBKEY[31:0]	FMC_CMR可选字节操作解锁寄存器

这些位仅能被软件写

写解锁值到OBKEY[31:0]解锁FMC\_CMCR寄存器的可选字节命令

### 3.4.4. 闪存控制/状态寄存器 (FMC\_CSR)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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								ENDF	WPEF	保留.		PGEF	保留		BUSY
								rw	rw	rw		rw		r	

位/位域	名称	描述
31:6	保留	必须保持复位值
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置1，软件写1清0。
4	WPEF	擦除/编程保护错误标志位 在受保护的页上擦除/编程操作时，此位被硬件置1，软件写1清0。
3	保留	必须保持复位值
2	PGEF	编程错误标志位 当被编程区域状态不为0xFFFF时对闪存编程，此位被硬件置1，软件写1清0。
1	保留	必须保持复位值
0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置1。当操作结束或者出错，此位被清0。

### 3.4.5. 闪存命令寄存器 (FMC\_CMCR)

地址偏移: 0x10

复位值: 0x0000 0080

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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	OPTR	ENDIE	保留	ERIE	OBWE	保留	LK	START	OBER	OBPG	保留	ME	PE	PG
	RW	RW		RW	RW		RW	RW	RW	RW		RW	RW	RW

位/位域	名称	描述
31:14	保留	必须保持复位值
13	OPTR	<p>可选字节重加载位 软件置1。</p> <p>0: 没有作用 1: 强制可选字节重装载, 产生系统复位</p>
12	ENDIE	<p>操作结束中断使能位 软件置1和清0。</p> <p>0: 无硬件中断产生 1: 使能操作结束中断</p>
11	保留	必须保持复位值
10	ERIE	<p>出错中断使能位 软件置1和清0。</p> <p>0: 无硬件中断产生. 1: 使能出错中断</p>
9	OBWE	<p>可选字节擦除/编程使能位 当正确的序列写入FMC_OBKEYR寄存器, 此位由硬件置1。此位可以被软件清0。</p>
8	保留	必须保持复位值
7	LK	<p>锁FMC_CMRR寄存器标志位 当正确的序列写入FMC_UKEYR寄存器, 此位由硬件清0, 此位可以由软件置1。</p>
6	START	<p>发送擦除命令到FMC位 软件置1可以发送擦除命令到FMC。当BUSY位被清0时, 此位由硬件清0。</p>
5	OBER	<p>可选字节擦除命令位 软件置1和清0。</p> <p>0: 无作用 1: 可选字节擦除命令</p>
4	OBPG	<p>可选字节编程命令位 软件置1和清0。</p> <p>0: 无作用 1: 可选字节编程命令</p>
3	保留	必须保持复位值
2	ME	主存储块整片擦除命令位

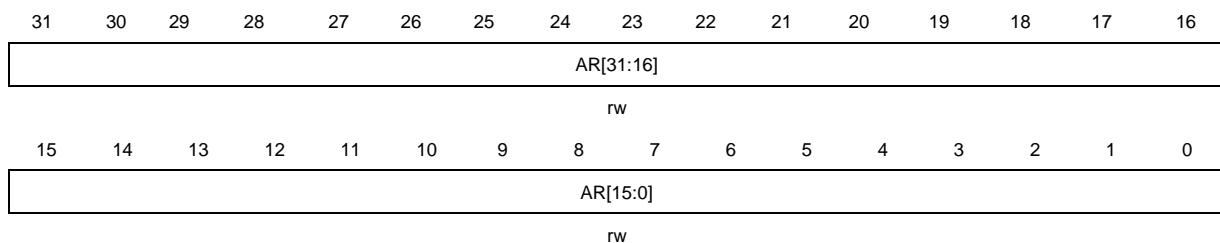
		软件置1和清0。
1	PE	0: 无作用 1: 主存储块整片擦除命令
		主存储块页擦除命令位
		软件置1和清0。
		0: 无作用 1: 主存储块页擦除命令
0	PG	主存储块页编程命令位
		软件置1和清0。
		0: 无作用 1: 主存储块页编程命令

### 3.4.6. 闪存命令地址寄存器 (FMC\_AR)

地址偏移: 0x14

复位值: 0x0000 0080

该寄存器只能按字 (32位) 访问



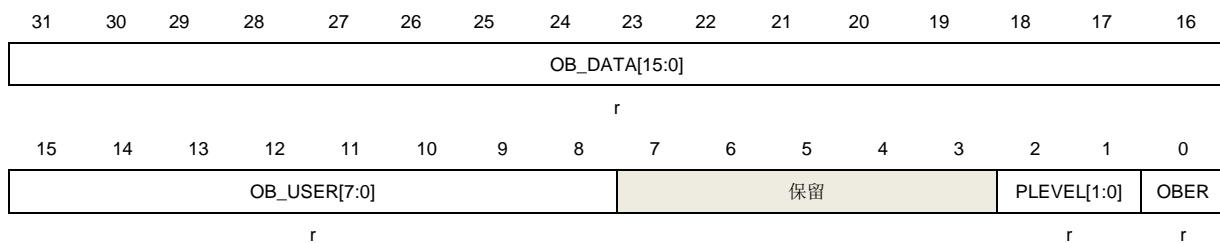
位/位域	名称	描述
31:0	AR[31:0]	闪存命令地址位 软件置位 AR位是闪存擦除命令的地址。

### 3.4.7. 闪存可选字节寄存器 (FMC\_OPTR)

地址偏移: 0x1C

复位值: 0xFFFF XX0X

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:16	OB_DATA[15:0]	系统复位后保存可选字节块的OB_DATA[15:0]部分
15:8	OB_USER[7:0]	系统复位后保存可选字节块的OB_USER字节
7:3	保留	必须保持复位值
2:1	PLEVEL[1:0]	安全保护级别 00: 无保护 01: 低保护级别 11: 高保护级别
0	OBER	可选字节读错误位 当可选字节和它的补码不匹配时此位由硬件置1，可选字节被设置为0xFF。

### 3.4.8. 闪存页擦除/编程保护寄存器 (FMC\_WPR)

地址偏移: 0x20

复位值: 0x0000 XXXX

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	OB_WP[15:0]	系统复位后保存可选字节块的OB_WP[15:0]部分 0: 保护生效 1: 未保护

### 3.4.9. 闪存等待状态控制寄存器 (FMC\_WSCR)

**GD32F130xx 和 GD32F150xx 产品**

地址偏移: 0xFC

复位值: 0xFFFF XXXX

该寄存器只能按字（32位）访问



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

r

位/位域	名称	描述
31:1	保留	必须保持复位值
0	WSEN	<p>FMC等待状态使能寄存器</p> <p>此位由软件置1和清0。此位也被FMC_UKEYR寄存器保护。需要写0x45670123和0xCDEF89AB到FMC_UKEYR寄存器。</p> <p>0: 从闪存取指无等待状态</p> <p>1: 从闪存取指增加等待状态</p>

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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															BPEN WSEN

rw rw

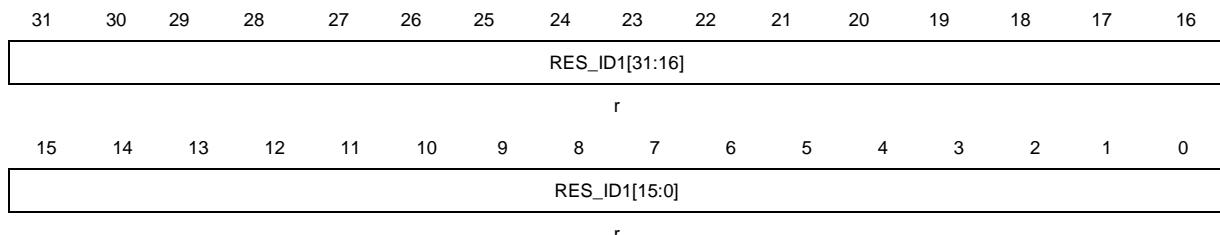
位/位域	名称	描述
31:2	保留	必须保持复位值
1	BPEN	<p>FMC位编程使能寄存器</p> <p>此位由软件置1和清0。</p> <p>0: 无效, 写操作必须检测闪存操作页全为FF</p> <p>1: 写操作不需要检测闪存操作页全为FF, FMC可以按位编程</p>
0	WSEN	<p>FMC等待状态使能寄存器</p> <p>此位由软件置1和清0。</p> <p>此位也被FMC_UKEYR寄存器保护。需要写0x45670123和0xCDEF89AB到FMC_UKEYR寄存器。</p> <p>0: 从闪存取指无等待状态</p> <p>1: 从闪存取指增加等待状态</p>

### 3.4.10. 闪存产品保留 ID 寄存器 1 (FMC\_RES\_ID1)

地址偏移: 0x100

复位值: 0xXXXX XXXX

该寄存器只能按字 (32位) 访问



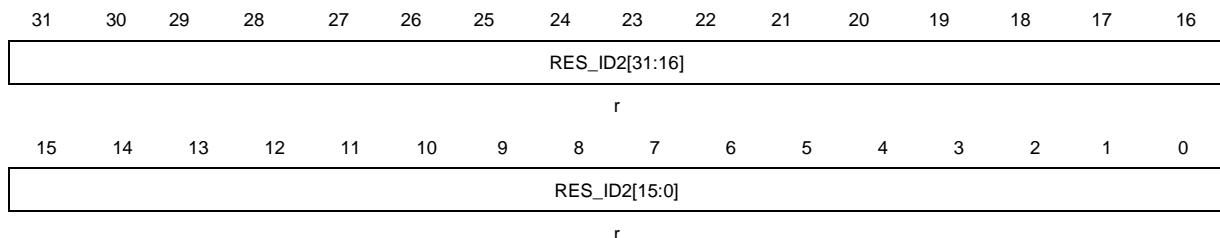
位/位域	名称	描述
31:0	RES_ID1[31:0]	产品保留 ID 寄存器1 这些位只读 上电后这些位始终不能改变，芯片生产以后只能被编程一次。

### 3.4.11. 闪存产品保留 ID 寄存器 2 (FMC\_RES\_ID2)

地址偏移: 0x104

复位值: 0xXXXX XXXX

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:0	RES_ID2[31:0]	产品保留ID寄存器2 这些位只读 上电后这些位始终不能改变，芯片生产以后只能被编程一次。

## 4. 复位和时钟控制单元 (RCC)

### 4.1. 复位控制单元 (RCU)

#### 4.1.1. 简介

GD32F1x0复位控制包括三种复位控制：电源复位、系统复位和备份域复位。电源复位又称为冷复位，电源启动时复位除了备份域的所有系统。除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。备份域复位复位备份区域。复位被外部信号、内部事件和复位发生器触发。接下章节将详细介绍这些复位。

#### 4.1.2. 功能描述

##### 电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位(POR/PDR 复位) 2、从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准将提供好1.2V电压（GD32F130xx和GD32F150xx产品）或1.8V电压（GD32F170xx和GD32F150xx产品）时，电源复位无效。复位入口矢量被固定在地址0x0000\_0004。

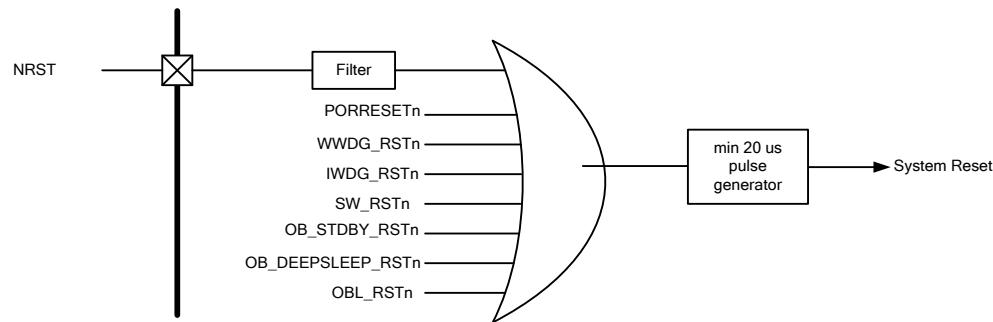
##### 系统复位

当发生以下任一事件时，产生一个系统复位：

- 上电复位 (PORRESETn)
- 外部引脚复位 (NRST)
- 窗口看门狗计数终止 (WWDG\_RSTn)
- 独立看门狗计数终止 (IWDG\_RSTn)
- Cortex™-M3的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’ (SW\_RSTn)
- 可选负载字节复位 (OBL\_RSTn)
- 用户选择字节寄存器OB\_STDBY\_RSTn设置为0，并且进入待机模式时 (OB\_STDBY\_RSTn)
- 用户选择字节寄存器OB\_DEEPSLEEP\_RSTn设置为0，并且进入深度睡眠模式时 (OB\_DEEPSLEEP\_RSTn)

系统复位脉冲发生器保证每一个复位源(外部或内部)都能有至少20μs的低电平脉冲延时。

图 4-1. 系统复位电路



### 备份域复位

当以下事件之一发生时，产生备份域复位。1、设置备份域控制寄存器中的BKPRST位为‘1’；  
2、备份域电源上电复位(在V<sub>DD</sub>和V<sub>BAT</sub>两者掉电的前提下，V<sub>DD</sub>或V<sub>BAT</sub>上电)。

## 4.2. 时钟控制单元 (CCU)

### 4.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部高速8M RC振荡器时钟(HSI)、一个内部高速14M RC 振荡器时钟(HSI14)(GD32F130xx和GD32F150xx产品)或一个内部高速28M RC振荡器时钟(HSI28)(GD32F170xx和GD32F190xx产品)、一个外部高速晶体振荡器时钟(HSE)、一个内部低速RC振荡器时钟(LSI)、一个外部低速晶体振荡器时钟(LSE)、一个锁相环(PLL)、一个HSE时钟监视器、时钟预分频器、时钟多路复用器和时钟选通电路。

AHB、APB和Cortex™-M3时钟都源自系统时钟(CK\_SYS)，系统时钟的时钟源为HSI、HSE或PLL。系统时钟的最大运行时钟频率可以达到72MHz。独立看门狗定时器有独立的时钟源(LSI)，实时时钟(RTC)使用LSI、LSE或HSE/32作为时钟源。

图4-2. GD32F130xx和GD32F150xx产品的时钟树

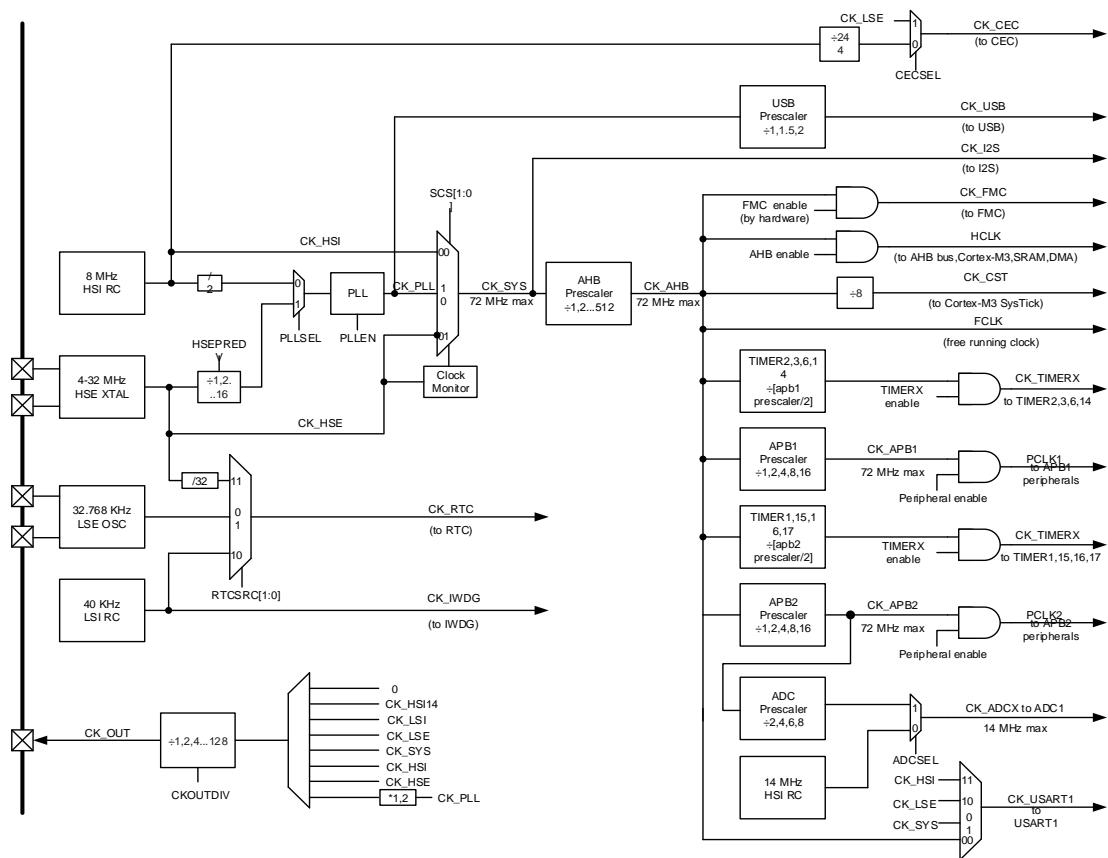
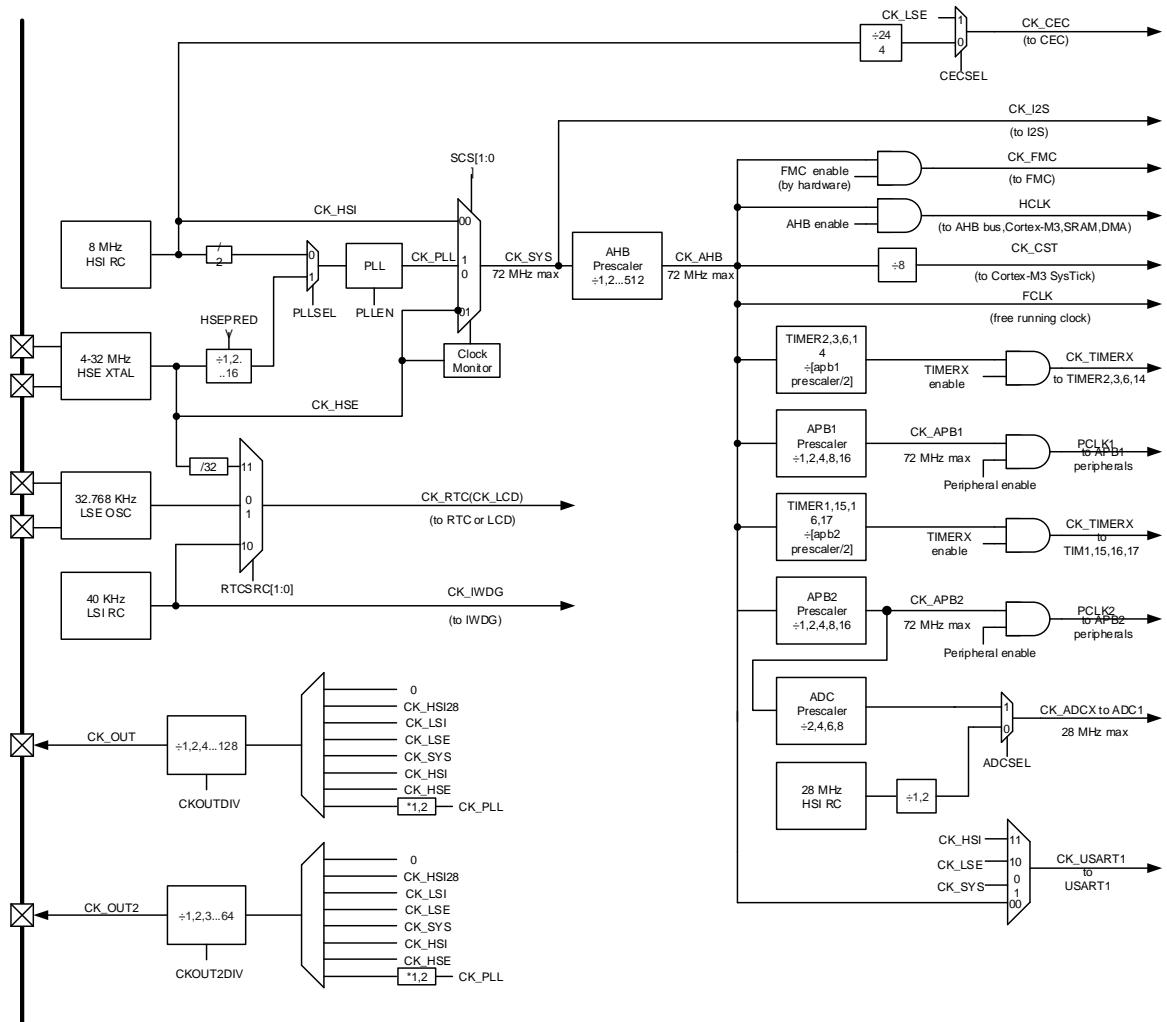


图4-3. GD32F170xx和GD32F190xx产品的时钟树



预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB和APB2/APB1域的最高时钟频率为72MHz。但当使用I2C外设时,APB1时钟需保证不大于36MHz。RCC通过AHB时钟(HCLK)8分频后作为Cortex系统定时器(SysTick)的外部时钟。通过对SysTick控制与状态寄存器的设置,可选择上述时钟或Cortex(HCLK)时钟作为SysTick时钟。

在GD32F130xx和GD32F150xx产品中ADC时钟由APB2时钟经2、4、6、8分频或HSI14时钟获得,在GD32F170xx和GD32F190xx产品中ADC时钟由APB2时钟经2、4、6、8分频或HSI28或HSI2/2时钟获得,它们是通过设置全局时钟配置寄存器3(RCC\_GCFDGR3)的ADCSEL位来选择ADC时钟源的。USART1的时钟可以选择HSI时钟、LSE时钟或APB2时钟,通过设置全局时钟配置寄存器3(RCC\_GCFDGR3)的USART1SEL位来选择。CEC时钟可以选择HSI时钟244分频或LSE时钟,通过设置全局时钟配置寄存器3(RCC\_GCFDGR3)的CECSEL位来选择。

RTC或LCD时钟(仅适用于GD32F170xx和GD32F190xx产品)可以选择LSE时钟、LSI时钟或HSE时钟32分频,通过设置备用域控制寄存器(RCC\_BDCR)的RTCSEL位来选择。

IWDG时钟可以选择LSI时钟,当IWDG启动时强制选择。

如果APB时钟分频系数为1,定时器的时钟频率与所在APB总线频率一致。否则,定时器的时

钟频率被设为与其相连的APB总线频率的2倍。

#### 4.2.2. 主要特性

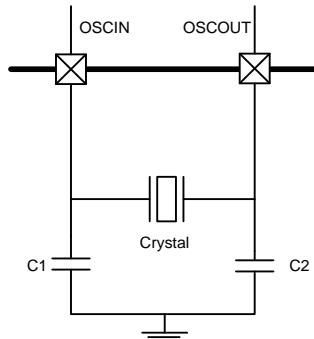
- 4到32 MHz外部高速晶体振荡器 (HSE)
- 8 MHz内部高速RC振荡器 (HSI)
- 14 MHz内部高速RC振荡器 (HSI14) (GD32F130xx和GD32F150xx产品)
- 28 MHz内部高速RC振荡器 (HSI28) (GD32F170xx和GD32F190xx产品)
- 32,768 Hz外部低速晶体振荡器 (LSE)
- 40 kHz内部低速RC振荡器 (LSI)
- PLL 时钟源可以是HSE或HSI
- HSE时钟可监视

#### 4.2.3. 功能描述

##### 高速外部晶体振荡器时钟(HSE)

4到32MHz的外部振荡器可为系统提供更为精确的主时钟。带有特定频率的晶体必须靠近两个HSE的引脚。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

图4-4. HSE接线图



HSE晶体可以通过设置时钟控制寄存器RCC\_GCCR的HSEEN位来启动或关闭，在时钟控制寄存器RCC\_GCCR中的HSESTB位用来指示高速外部振荡器是否已稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间又称启动时间。当HSE时钟稳定后，如果在时钟中断寄存器RCC\_GCIR中的相应中断使能位HSESTBIE位被置‘1’，将会产生相应中断。在这一点上，HSE时钟可以被直接用作系统时钟源或者PLL输入时钟。

将时钟控制寄存器RCC\_GCCR的HSEBPS和HSEEN位置‘1’可以设置外部时钟旁路模式。CK\_HSE等于驱动OSCIN管脚的外部时钟。

##### 高速内部8MHz RC振荡器时钟 (HSI)

高速内部8MHz RC振荡器时钟，简称HSI时钟，拥有8MHz的固定频率，设备上电后CPU默认

选择的时钟源就是HSI时钟。HSI RC振荡器能够在不需要任何外部器件的条件下提供更低成本类型的时钟源。HSI晶体可以通过设置时钟控制寄存器(RCC\_GCCR)中的HSIEN位被启动和关闭。时钟控制寄存器RCC\_GCCR中的HSIRSTB位用来指示HSI内部RC振荡器是否稳定。HSI振荡器的启动时间比HSE晶体振荡器要更短。如果时钟中断寄存器RCC\_GCIR中的相应中断使能位HSISTBIE被置‘1’，在HSI稳定以后，将产生一个中断。HSI时钟也可用作PLL输入时钟。

工厂会校准HSI时钟频率的精度，但是它的精度仍然比HSE时钟要差。用户需求、环境条件和成本将决定选择哪个时钟作为系统时钟源。

如果HSE或者PLL是系统时钟源，为了最大程度减小系统从深度睡眠模式启动的时间，系统从深度睡眠模式初始唤醒的时候硬件强制HSI时钟作为系统时钟。

### 锁相环 (PLL)

内部锁相环PLL通过对输入参考频率为4~32MHz的时钟基准2 ~32倍频，可以提供16~72 MHz的时钟输出。

PLL可以通过设置时钟控制寄存器(RCC\_GCCR)中的PLLEN位被启动和关闭。时钟控制寄存器RCC\_GCCR中的PLLSTB位用来指示PLL时钟是否稳定。如果时钟中断寄存器RCC\_GCIR中的相应中断使能位PLLSTBIE被置‘1’，在PLL稳定以后，将产生一个中断。

### 高速内部14M RC振荡器时钟 (HSI14) (GD32F130xx和GD32F150xx产品)

高速内部14M RC振荡器时钟 (HSI14)有一个固定的频率14MHz，专门用作ADC时钟。HSI14可以通过设置时钟控制寄存器2(RCC\_GCCR2)中的HSI14EN位被启动和关闭。时钟控制寄存器2(RCC\_GCCR2)中的HSI14RSTB位用来指示HSI14时钟是否已稳定。如果时钟中断寄存器RCC\_GCIR中的相应中断使能位HSI14STBIE被置‘1’，在HSI14稳定以后，将产生一个中断。

### 高速内部28M RC振荡器时钟 (HSI28) (GD32F170xx和GD32F190xx产品)

高速内部28M RC振荡器时钟(HSI28)有一个固定的频率28MHz，专门用作ADC时钟。HSI28可以通过设置时钟控制寄存器2(RCC\_GCCR2)中的HSI28EN位被启动和关闭。时钟控制寄存器2(RCC\_GCCR2)中的HSI28RSTB位用来指示HSI28时钟是否已稳定。如果时钟中断寄存器RCC\_GCIR中的相应中断使能位HSI28STBIE被置‘1’，在HSI28稳定以后，将产生一个中断。

### 低速外部晶体振荡器时钟(LSE)

LSE晶体是一个32.768kHz的低速外部晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且精确的时钟源。LSE时钟可以通过设置备份域控制寄存器(RCC\_BDCR)中的LSEEN位被启动和关闭。备份域控制寄存器RCC\_BDCR中的LSESTB位用来指示LSE时钟是否稳定。如果时钟中断寄存器RCC\_GCIR中的相应中断使能位LSESTBIE被置‘1’，在LSE稳定以后，将产生一个中断。

将备份域控制寄存器RCC\_BDCR的LSEBPS和LSEEN位置‘1’可以选择外部时钟旁路模式。CK\_LSE与连到OSC32IN脚上外部时钟信号一致。

### 低速内部RC振荡器时钟(LSI)

LSI RC振荡器时钟担当一个低功耗时钟源的角色，它的时钟频率大约40 kHz，为独立看门狗和实时时钟电路提供时钟。LSI提供低成本的时钟源，因为不需要外部器件。LSI RC振荡器可以通过设置控制/状态寄存器RCC\_GCSR中的LSIEN位被启动和关闭。控制/状态寄存器

RCC\_GCSR中的LSISTB位用来指示LSI时钟是否已稳定。如果控制/状态寄存器RCC\_GCSR中的相应中断使能位LSISTBIE被置‘1’，在LSI稳定以后，将产生一个中断。

### 系统时钟 (CK\_SYS) 选择

系统复位后，HSI时钟被选为系统时钟，改变时钟配置寄存器RCC\_GCFGR 中的系统时钟变换位SCS可以切换系统时钟源为HSE或PLL。当SCS的值改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

### HSE 时钟监视器(CKM)

设置时钟控制寄存器RCC\_GCCR 中的HSE时钟监视使能位CKMEN，HSE可以使能时钟监视功能。该功能必须在HSE启动延迟完毕后使能，在HSE停止后禁止。一旦监测到HSE故障，HSE将自动被禁止，时钟中断寄存器RCC\_GCIR 中的HSE时钟阻塞标志位CKMF将被置‘1’，产生HSE故障事件。这个故障引发的中断和Cortex-M3的不可屏蔽中断相连。如果HSE被选作系统或PLL的时钟源，HSE故障将促使选择HSI为系统时钟源且PLL将被自动禁止。

### 时钟输出功能

#### GD32F130xx和GD32F150xx产品：

时钟输出功能输出从32kHz到54MHz的时钟。通过设置时钟配置寄存器RCC\_GCFGR中的CK\_OUT时钟源选择位CKOUTSRC[2:0]能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O(AFIO)模式来输出选择的时钟信号。

表 4-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	CK_HSI14
010	CK_LSI
011	CK_LSE
100	CK_SYS
101	CK_HSI
110	CK_HSE
111	CK_PLL 或 CK_PLL/2

通过配置时钟配置寄存器RCC\_GCFGR的CKOUTDIV[2:0]位，可以将输出时钟按比例分频，进而降低CK\_OUT频率。

#### GD32F170xx和GD32F190xx产品：

时钟输出功能输出从32kHz到54MHz的时钟。通过设置时钟配置寄存器RCC\_GCFGR中的CK\_OUT时钟源选择位CKOUTSRC[2:0]和时钟配置寄存器4RCC\_GCFGR4中的CK\_OUT2时钟源选择位CKOUT2SRC[2:0]能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O(AFIO)模式来输出选择的时钟信号。

表 4-2. 时钟源的选择

时钟源选择位	时钟源
--------	-----

000	无时钟
001	CK_HSI28
010	CK_LSI
011	CK_LSE
100	CK_SYS
101	CK_HSI
110	CK_HSE
111	CK_PLL 或 CK_PLL/2

通过配置时钟配置寄存器RCC\_GCFGR的CKOUTDIV[2:0]位，可以将输出时钟按比例分频，进而降低CK\_OUT频率。

CK\_OUT2的时钟源通过时钟配置寄存器4RCC\_GCFGR4的CKOUT2SRC[2:0]位配置。通过配置时钟配置寄存器4(RCC\_GCFGR4)的CKOUT2DIV[2:0]位，可以将输出时钟按比例分频，进而降低CK\_OUT频率。

### 深度睡眠模式时钟控制

当MCU工作在深度睡眠模式时，HDMI CEC或USART1能唤醒MCU，前提是它们的时钟是由LSE时钟提供且LSE时钟被使能。

如果HDMI CEC或USART1时钟选择HSI时钟并且工作在深度睡眠模式，它们有能力开启或关闭HSI时钟，HDMI CEC或USART1采用HSI时钟作为工作时钟来唤醒深度睡眠模式。

### 电压控制

#### GD32F130xx和GD32F150xx产品：

掉电选择寄存器(RCC\_PDR\_VC)中的PDR\_S位可以控制掉电复位。如果PDR\_S位是0，VDD低于2.6V的时候电源掉电复位生效。如果PDR\_S位是1，VDD低于1.8V的时候掉电复位生效。在PDR\_S位是1时，并且VDD低于2.6V时，闪存编程和擦除不能使用。

深度睡眠模式电压寄存器(RCC\_DEEPSLEEP\_VC)中的DEEPSLEEP\_VC[2:0]位可以控制内核在深度睡眠模式下的电压。

表 4-3. 深度睡眠模式下内核电压选择

DEEPSLEEP_VC[2:0]	深度睡眠模式电压(V)
000	1.2
001	1.1
010	1.0
011	0.9
100 ~ 111	保留

RCC\_PDR\_VC和RCC\_DEEPSLEEP\_VC寄存器被电源解锁寄存器(RCC\_VC\_KEY)保护。只有在写0x1A2B3C4D到RCC\_VC\_KEY后，RCC\_PDR\_VC和RCC\_DEEPSLEEP\_VC寄存器才能被写入。

#### GD32F170xx和GD32F190xx产品：

深度睡眠模式电压寄存器(RCC\_DEEPSLEEP\_VC)中的DEEPSLEEP\_VC[2:0]位可以控制内

核在深度睡眠模式下的电压。

**表 4-4. 深度睡眠模式下内核电压选择**

DEEPSLEEP_VC[2:0]	深度睡眠模式电压(V)
000	1.8
001	1.6
010	1.4
011	1.2
100 ~ 111	保留

RCC\_PDR\_VC和RCC\_DEEPSLEEP\_VC寄存器被电源解锁寄存器(RCC\_VC\_KEY)保护。只有在写0xA2B3C4D到RCC\_VC\_KEY后, RCC\_PDR\_VC和RCC\_DEEPSLEEP\_VC寄存器才能被写入。

## 4.3. RCC 寄存器

### 4.3.1. 总体时钟控制寄存器 (RCC\_GCCR)

地址偏移: 0x00

复位值: 0x0000 XX83 X表示未定义。

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PLLSTB	PLLEN	保留			CKMEN	HSEBPS	HSESTB	HSEEN	
						r	rw				rw	rw	r	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICALIB[7:0]						HSIADJ[4:0]				保留		HSISTB	HSIEN		
						r				rw		r	rw		

位/位域	名称	描述
31:26	保留	必须保持复位值
25	PLLSTB	PLL 时钟稳定标志位 硬件置'1'来指示PLL输出时钟是否稳定待用。 0: PLL没稳定 1: PLL稳定
24	PLLEN	PLL 使能 软件置位或复位。如果PLL时钟作为系统时钟的时候该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0: PLL被关闭 1: PLL被打开
23:20	保留	必须保持复位值.
19	CKMEN	HSE时钟监视使能 0: 禁止外部4 ~ 32 MHz 晶体振荡器(HSE) 时钟监视器

		1: 使能外部4 ~ 32 MHz 晶体振荡器(HSE) 时钟监视器 当硬件监测到HSE时钟一直停留在低或者高的状态，内部硬件将切换系统时钟到HSI RC时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清CKMF 位。 <b>注意：</b> 使能HSE时钟监视器以后，硬件无视控制位HSIEN的状态，自动使能HSI时钟。
18	HSEBPS	外部晶体振荡器(HSE)时钟旁路模式使能 只有在HSEEN 位为0时HSEBPS位才可写。 0: 禁止HSE旁路模式 1: 使能HSE旁路模式 HSE输出时钟等于输入时钟
17	HSESTB	外部晶体振荡器(HSE)时钟稳定状态标志位 硬件置‘1’来指示HSE振荡器时钟是否稳定待用。 0: HSE 振荡器未稳定 1: HSE 振荡器已稳定
16	HSEEN	外部高速振荡器时钟使能 软件置‘1’或清‘0’。如果HSE时钟或者PLL输入时钟作为系统时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0: 禁止外部4 ~ 32 MHz晶体振荡器 1: 使能外部4 ~ 32 MHz 晶体振荡器
15:8	HSICALIB[7:0]	高速内部振荡器校准值寄存器 上电时自动加载这些位
7:3	HSIADJ[4:0]	高速内部振荡器时钟调整值 这些位由软件置位，最终调整值为HSIADJ当前值加上HSICALIB[7:0]位的值。最终调整值应该调整HSI到8 MHz ± 1%。
2	保留	必须保持复位值.
1	HSISTB	高速内部(HSI)时钟稳定状态标志位 硬件置‘1’来指示HSI振荡器时钟是否稳定待用。 0: HSI 振荡器未稳定 1: HSI 振荡器已稳定
0	HSIEN	高速内部振荡器使能 软件复位置位。如果HSI时钟用作系统时钟时该位不能被复位。当从待机和深度睡眠模式返回或用作系统时钟的HSE振荡器发生故障时，该位由硬件置1来启动HSI振荡器。 0: 内部8 MHz RC振荡器关闭 1: 内部8 MHz RC振荡器开启

#### 4.3.2. 总体时钟配置寄存器 (RCC\_GCFGR)

**GD32F130xx 和 GD32F150xx 产品**

地址偏移： 0x04

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV	CKOUTDIV[2:0]	PLLMF[4]	CKOUTSRC[2:0]	USBPS[1:0]	PLLMF[3:0]	PLLPREDV	PLLSEL								
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPS[1:0]	APB2PS[2:0]	APB1PS[2:0]	AHBPS[3:0]	SCSS[1:0]	SCS[1:0]										
RW	RW	RW	RW	RW	RW	R	RW								

位/位域	名称	描述
31	PLLDV	CK_PLL 1或2分频来用作CK_OUT 0: CK_PLL 2分频用作CK_OUT 1: CK_PLL用作CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT分频器, 来降低CK_OUT频率 CK_OUT的选择参考GCFGR的26:24位。 000: CK_OUT不分频 001: CK_OUT2分频 010: CK_OUT4分频 011: CK_OUT8分频 100: CK_OUT16分频 101: CK_OUT32分频 110: CK_OUT64分频 111: CK_OUT128分频
27	PLLMF[4]	PLLMF 寄存器第4位 见GCFGR的21:18位
26:24	CKOUTSRC[2:0]	CK_OUT时钟源选择 软件置位或清零。 000: 没有时钟被选择 001: 选择高速14M内部振荡器时钟 010: 选择低速内部振荡器时钟 011: 选择外部低速振荡器时钟 100: 选择系统时钟 101: 选择内部高速8M振荡器时钟 110: 选择外部高速振荡器时钟 111: 依赖于PLLDV选择(CK_PLL / 2)或CK_PLL
23:22	USBPS[1:0]	USB时钟预分频选择 软件置位或清零来控制USB时钟预分频值。USB时钟必须为48MHz.。如果USB时钟使能这些位不能被复位。 00: 选择(CK_PLL / 1.5) 01: 选择CK_PLL

---

		10: 选择(CK_PLL / 2.5) 11: 选择(CK_PLL / 2)
21:18	PLLMF[3:0]	PLL倍频因子 软件写这些位包括GCFGR的27位来确定PLL的倍频因子。 00000: (PLL时钟源 x 2) 00001: (PLL时钟源 x 3) 00010: (PLL时钟源 x 4) 00011: (PLL时钟源 x 5) 00100: (PLL时钟源 x 6) 00101: (PLL时钟源 x 7) 00110: (PLL时钟源 x 8) 00111: (PLL时钟源 x 9) 01000: (PLL时钟源 x 10) 01001: (PLL时钟源 x 11) 01010: (PLL时钟源 x 12) 01011: (PLL时钟源 x 13) 01100: (PLL时钟源 x 14) 01101: (PLL时钟源 x 15) 01110: (PLL时钟源 x 16) 01111: (PLL时钟源 x 16) 10000: (PLL时钟源 x 17) 10001: (PLL时钟源 x 18) 10010: (PLL时钟源 x 19) 10011: (PLL时钟源 x 20) 10100: (PLL时钟源 x 21) 10101: (PLL时钟源 x 22) 10110: (PLL时钟源 x 23) 10111: (PLL时钟源 x 24) 11000: (PLL时钟源 x 25) 11001: (PLL时钟源 x 26) 11010: (PLL时钟源 x 27) 11011: (PLL时钟源 x 28) 11100: (PLL时钟源 x 29) 11101: (PLL时钟源 x 30) 11110: (PLL时钟源 x 31) 11111: (PLL时钟源 x 32) 注意: PLL输出频率不能超过72MHz。
17	PLLPREDV	HSE分频器作为PLL输入。该位与时钟配置寄存器2(RCC_GCFGR2)中的HSEPREDV[0]位是一样的。参考RCC_CGFGR2的HSEPREDV位说明。 由软件置1或清0来分频或不分频HSE后，当作PLL输入时钟源。 0: 选择HSE时钟 1: HSE时钟二分频

16	PLLSEL	PLL 时钟源选择 软件置1或清0来控制PLL时钟源 0: 选择HSI二分频为PLL时钟源 1: 选择HSE为PLL时钟源
15:14	ADCPS[1:0]	ADC时钟预分频选择 软件清0和置1。 00: 选择APB2时钟2分频 01: 选择APB2时钟4分频 10: 选择APB2时钟6分频 11: 选择APB2时钟8分频
13:11	APB2PS[2:0]	APB2预分频选择 软件置1和清0来控制APB2时钟分频因子。 0xx: 选择AHB时钟不分频 100: 选择AHB时钟2分频 101: 选择AHB时钟4分频 110: 选择AHB时钟8分频 111: 选择AHB时钟16分频
10:8	APB1PS[2:0]	APB1 预分频选择 软件设置和清除来控制APB1时钟分频因子。 0xx: 选择AHB时钟不分频 100: 选择AHB时钟2分频 101: 选择AHB时钟4分频 110: 选择AHB时钟8分频 111: 选择AHB时钟16分频
7:4	AHBPS[3:0]	AHB预分频选择 软件设置和清除来控制AHB时钟分频因子。 0xxx: 选择CK_SYS系统时钟不分频 1000: 选择CK_SYS系统时钟2分频 1001: 选择CK_SYS系统时钟4分频 1010: 选择CK_SYS系统时钟8分频 1011: 选择CK_SYS系统时钟16分频 1100: 选择CK_SYS系统时钟64分频 1101: 选择CK_SYS系统时钟128分频 1110: 选择CK_SYS系统时钟256分频 1111: 选择CK_SYS系统时钟512分频
3:2	SCSS[1:0]	系统时钟转换状态 硬件设置和清除指示系统当前时钟源 00: 选择CK_HSI作为CK_SYS 系统时钟源 01: 选择CK_HSE作为CK_SYS 系统时钟源 10: 选择CK_PLL作为CK_SYS 系统时钟源 11: 保留

1:0	SCS[1:0]	系统时钟转换
		软件设置选择系统时钟源。由于CK_SYS的改变有固有的延迟，需要软件读SCSS位来确保转换是否结束。在从深度睡眠或待机模式中返回时，或作为系统时钟或PLL时钟源的HSE出现故障时，强制选择HSI作为系统时钟。
00:	选择HSI时钟作为CK_SYS系统时钟源	
01:	选择HSE时钟作为CK_SYS系统时钟源	
10:	选择PLL作为CK_SYS系统时钟源	
11:	保留	

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV	CKOUTDIV[2:0]			PLLMF[4]	CKOUTSRC[2:0]			保留	PLLMF[3:0]				PLLPREDV	PLLSEL	
rw	rw	rw	rw	rw									rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPS[1:0]	APB2PS[2:0]			APB1PS[2:0]			AHBPS[3:0]				SCSS[1:0]	SCS[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	rw

位/位域	名称	描述
31	PLLDV	CK_PLL 1或2分频来用作CK_OUT 0: CK_PLL 2分频用作CK_OUT 1: CK_PLL用作CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT分频器，来降低CK_OUT频率 CK_OUT的选择参考GCFGR的26:24位 000: CK_OUT不分频 001: CK_OUT2分频 010: CK_OUT4分频 011: CK_OUT8分频 100: CK_OUT16分频 101: CK_OUT32分频 110: CK_OUT64分频 111: CK_OUT128分频
27	PLLMF[4]	PLLMF寄存器第4位 见GCFGR的21:18位
26:24	CKOUTSRC[2:0]	CK_OUT时钟源选择 软件置位或清零。 000: 没有时钟被选择 001: 选择高速28M内部振荡器时钟

010: 选择低速内部振荡器时钟  
011: 选择外部低速振荡器时钟  
100: 选择系统时钟  
101: 选择内部高速8M振荡器时钟  
110: 选择外部高速振荡器时钟  
111: 依赖于PLLDV选择(CK\_PLL / 2)或的CK\_PLL

23:22 保留 必须保持复位值.

21:18 PLLMF[3:0] PLL倍频因子  
软件写这些位包括GCFGR的27位来确定PLL的倍频因子。

00000: (PLL时钟源 x 2)  
00001: (PLL时钟源 x 3)  
00010: (PLL时钟源 x 4)  
00011: (PLL时钟源 x 5)  
00100: (PLL时钟源 x 6)  
00101: (PLL时钟源 x 7)  
00110: (PLL时钟源 x 8)  
00111: (PLL时钟源 x 9)  
01000: (PLL时钟源 x 10)  
01001: (PLL时钟源 x 11)  
01010: (PLL时钟源 x 12)  
01011: (PLL时钟源 x 13)  
01100: (PLL时钟源 x 14)  
01101: (PLL时钟源 x 15)  
01110: (PLL时钟源 x 16)  
01111: (PLL时钟源 x 16)  
10000: (PLL时钟源 x 17)  
10001: (PLL时钟源 x 18)  
10010: (PLL时钟源 x 19)  
10011: (PLL时钟源 x 20)  
10100: (PLL时钟源 x 21)  
10101: (PLL时钟源 x 22)  
10110: (PLL时钟源 x 23)  
10111: (PLL时钟源 x 24)  
11000: (PLL时钟源 x 25)  
11001: (PLL时钟源 x 26)  
11010: (PLL时钟源 x 27)  
11011: (PLL时钟源 x 28)  
11100: (PLL时钟源 x 29)  
11101: (PLL时钟源 x 30)  
11110: (PLL时钟源 x 31)  
11111: (PLL时钟源 x 32)

注意: PLL输出频率不能超过72 MHz。

---

17	PLLPREDV	HSE分频器作为PLL输入。该位与时钟配置寄存器2(RCC_GCFGR2)中的HSEPREDV[0]位是一样的。参考RCC_CGFGR2的HSEPREDV位说明。 由软件置1或清0来分频或不分频HSE后，当作PLL输入时钟源。 0: 选择HSE时钟 1: HSE时钟二分频
16	PLLSEL	PLL时钟源选择 软件置1或清0来控制PLL时钟源。 0: 选择HSI二分频为PLL时钟源 1: 选择HSE为PLL时钟源
15:14	ADCPS[1:0]	ADC时钟预分频选择 软件清0和置1。 00: 选择APB2时钟2分频 01: 选择APB2时钟4分频 10: 选择APB2时钟6分频 11: 选择APB2时钟8分频
13:11	APB2PS[2:0]	APB2预分频选择 软件置1和清0来控制APB2时钟分频因子。 0xx: 选择AHB时钟不分频 100: 选择AHB时钟2分频 101: 选择AHB时钟4分频 110: 选择AHB时钟8分频 111: 选择AHB时钟16分频
10:8	APB1PS[2:0]	APB1预分频选择 软件设置和清除来控制APB1时钟分频因子。 0xx: 选择AHB时钟不分频 100: 选择AHB时钟2分频 101: 选择AHB时钟4分频 110: 选择AHB时钟8分频 111: 选择AHB时钟16分频
7:4	AHBPS[3:0]	AHB预分频选择 软件设置和清除来控制AHB时钟分频因子。 0xxx: 选择CK_SYS系统时钟不分频 1000: 选择CK_SYS系统时钟2分频 1001: 选择CK_SYS系统时钟4分频 1010: 选择CK_SYS系统时钟8分频 1011: 选择CK_SYS系统时钟16分频 1100: 选择CK_SYS系统时钟64分频 1101: 选择CK_SYS系统时钟128分频 1110: 选择CK_SYS系统时钟256分频 1111: 选择CK_SYS系统时钟512分频
3:2	SCSS[1:0]	系统时钟转换状态

硬件设置和清除指示系统当前时钟源。

- 00: 选择CK\_HIS作为CK\_SYS系统时钟源
- 01: 选择CK\_HSE作为CK\_SYS系统时钟源
- 10: 选择CK\_PLL作为CK\_SYS系统时钟源
- 11: 保留

1:0	SCS[1:0]	系统时钟转换	软件设置选择系统时钟源。由于CK_SYS的改变有固有的延迟，需要软件读SCSS位来确保转换是否结束。在从深度睡眠或待机模式中返回时，或作为系统时钟或PLL时钟源的HSE出现故障时，强制选择HSI作为系统时钟。
			<ul style="list-style-type: none"> <li>00: 选择HSI时钟作为CK_SYS系统时钟源</li> <li>01: 选择HSE时钟作为CK_SYS系统时钟源</li> <li>10: 选择PLL作为CK_SYS系统时钟源</li> <li>11: 保留</li> </ul>

#### 4.3.3. 总体时钟中断寄存器 (RCC\_GCIR)

##### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CKMR	保留	HSI14 STBR	PLL STBR	HSE STBR	HSI STBR	LSE STBR	LSI STBR
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	HSI14 STBIE	PLL STBIE	HSE STBIE	HSI STBIE	LSE STBIE	LSI STBIE	CKMF	保留	HSI14 STBF	PLL STBF	HSE STBF	HSI STBF	LSE STBF	LSI STBF	
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值
23	CKMR	<p>HSE时钟阻塞中断复位 软件写1复位CKMF标志位。 0: 不复位CKMF标志位 1: 复位CKMF标志位</p>
22	保留	必须保持复位值
21	HSI14STBR	<p>HSI14时钟稳定中断复位 软件写1复位HSI14STBF标志位。 0: 不复位HSI14STBF标志位</p>

		1: 复位HSI14STBF标志位
20	PLLSTBR	PLL稳定中断复位 软件写1复位PLLSTBF标志位。 0: 不复位PLLSTBF标志位 1: 复位PLLSTBF标志位
19	HSESTBR	HSE时钟稳定中断复位 软件写1复位HSESTBF标志位。 0: 不复位HSESTBF标志位 1: 复位HSESTBF标志位
18	HSISTBR	HSI时钟稳定中断复位 软件写1复位HSISTBF标志位。 0: 不复位HSISTBF标志位 1: 复位HSISTBF标志位
17	LSESTBR	LSE时钟稳定中断复位 软件写1复位LSESTBF标志位。 0: 不复位LSESTBF标志位 1: 复位LSERDYF标志位
16	LSISTBR	LSI时钟稳定中断复位 软件写1复位LSIRDYF标志位。 0: 不复位LSISTBF标志位 1: 复位LSIRDYF标志位
15:14	保留	必须保持复位值
13	HSI14STBIE	HSI14时钟稳定中断使能 软件置1和清0来使能/禁止HSI14时钟稳定中断。 0: 禁止HSI14时钟稳定中断 1: 使能HSI14时钟稳定中断
12	PLLSTBIE	PLL时钟稳定中断使能 软件置1和清0来使能/禁止PLL时钟稳定中断。 0: 禁止PLL时钟稳定中断 1: 使能PLL时钟稳定中断
11	HSESTBIE	HSE时钟稳定中断使能 软件置1和清0来使能/禁止HSE时钟稳定中断。 0: 禁止HSE时钟稳定中断 1: 使能HSE时钟稳定中断
10	HSISTBIE	HSI时钟稳定中断使能 软件置1和清0来使能/禁止HSI时钟稳定中断。 0: 禁止HSI时钟稳定中断 1: 使能HSI时钟稳定中断

9	LSESTBIE	LSE时钟稳定中断使能 LSE时钟稳定中断使能/禁止控制。 0: 禁止LSE时钟稳定中断 1: 使能LSE时钟稳定中断
8	LSISTBIE	LSI时钟稳定中断使能 LSI时钟稳定中断使能/禁止控制。 0: 禁止LSI时钟稳定中断 1: 使能LSI时钟稳定中断
7	CKMF	HSE时钟阻塞中断标志位 当HSE时钟阻塞时硬件置1。 软件置CKMR=1时清除该位。 0: 时钟运行正常 1: HSE时钟阻塞
6	保留	必须保持复位值.
5	HSI14STBF	HSI14 时钟稳定中断标志位 当HSI14时钟稳定且HSI14STBIE位被置1时由硬件置1。 软件置HSI14STBR=1时清除该位。 0: 无HSI14时钟稳定中断产生 1: HSI14时钟稳定中断发生
4	PLLSTBF	PLL 时钟稳定中断标志位 当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1。 软件置PLLSTBR=1时清除该位。 0: 无PLL时钟稳定中断产生 1: 产生PLL时钟稳定中断
3	HSESTBF	HSE时钟稳定中断标志位 当外部4 ~ 16 MHz晶体振荡器时钟稳定且HSESTBIE位被置1时由硬件置1。 软件置HSESTBR=1时清除该位。 0: 无HSE时钟稳定中断发生 1: 发生HSE时钟稳定中断
2	HSISTBF	HSI时钟稳定中断标志位 当内部8 MHz RC振荡器时钟稳定且HSISTBIE位被置1时由硬件置1。 软件置HSISTBR=1时清除该位。 0: 无HSI时钟稳定中断产生 1: 产生HSI时钟稳定中断
1	LSESTBF	LSE 时钟稳定中断标志位 当外部32.768KHz晶体振荡器时钟稳定且LSESTBIE为被置1时由硬件置1。 软件置LSESTBR=1时清除该位。 0: 无LSE时钟稳定中断发生 1: 发生LSE时钟稳定中断

0	LSISTBF	LSI时钟稳定中断标志位 当内部32kHz RC振荡器时钟稳定且LSISTBIE位被置1时由硬件置1。 软件置LSISTBR =1时清除该位。 0: 无LSI时钟稳定中断产生 1: 产生LSI时钟稳定中断
---	---------	--

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CKMR	保留	HSI28 STBR	PLL STBR	HSE STBR	HSI STBR	LSE STBR	LSI STBR
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	HSI28 STBIE	PLL STBIE	HSE STBIE	HSI STBIE	LSE STBIE	LSI STBIE	CKMF	保留	HSI28 STBF	PLL STBF	HSE STBF	HSI STBF	LSE STBF	LSI STBF	
rw	rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r	r

位	位域	说明
31:24	保留	必须保持复位值
23	CKMR	HSE时钟阻塞中断复位 软件写1复位CKMF标志位。 0: 不复位CKMF标志位 1: 复位CKMF标志位
22	保留	必须保持复位值
21	HSI28STBR	HSI28时钟稳定中断复位 软件写1复位HSI28STBF标志位。 0:不复位HSI28STBF标志位 1:复位HSI28STBF标志位
20	PLLSTBR	PLL稳定中断复位 软件写1复位PLLSTBF标志位。 0: 不复位PLLSTBF标志位 1: 复位PLLSTBF标志位
19	HSESTBR	HSE时钟稳定中断复位 软件写1复位HSESTBF标志位。 0: 不复位HSESTBF标志位 1: 复位HSESTBF标志位

18	HSISTBR	HSI时钟稳定中断复位 软件写1复位HSISTBF标志位。 0: 不复位HSISTBF标志位 1: 复位HSISTBF标志位
17	LSESTBR	LSE时钟稳定中断复位 软件写1复位LSESTBF标志位。 0: 不复位LSESTBF标志位 1: 复位LSERDYF标志位
16	LSISTBR	LSI时钟稳定中断复位 软件写1复位LSIRDYF标志位。 0: 不复位LSISTBF标志位 1: 复位LSIRDYF标志位
15:14	保留	必须保持复位值
13	HSI28STBIE	HSI28时钟稳定中断使能 软件置1和清0来使能/禁止HSI28时钟稳定中断。 0: 禁止HSI28时钟稳定中断 1: 使能HSI28时钟稳定中断
12	PLLSTBIE	PLL时钟稳定中断使能 软件置1和清0来使能/禁止PLL时钟稳定中断。 0: 禁止PLL时钟稳定中断 1: 使能PLL时钟稳定中断
11	HSESTBIE	HSE时钟稳定中断使能 软件置1和清0来使能/禁止 HSE 时钟稳定中断。 0: 禁止HSE时钟稳定中断 1: 使能HSE时钟稳定中断
10	HSISTBIE	HSI时钟稳定中断使能 软件置1和清0来使能/禁止 HSI 时钟稳定中断。 0: 禁止HSI时钟稳定中断 1: 使能HSI时钟稳定中断
9	LSESTBIE	LSE时钟稳定中断使能 LSE时钟稳定中断使能/禁止控制。 0: 禁止LSE时钟稳定中断 1: 使能LSE时钟稳定中断
8	LSISTBIE	LSI时钟稳定中断使能 LSI时钟稳定中断使能/禁止控制。 0: 禁止LSI时钟稳定中断 1: 使能LSI时钟稳定中断
7	CKMF	HSE时钟阻塞中断标志位 当HSE时钟阻塞时硬件置1。

		软件置CKMR=1时清除该位。 0: 时钟运行正常 1: HSE时钟阻塞
6	保留	必须保持复位值.
5	HSI28STBF	<b>HSI28</b> 时钟稳定中断标志位 当HSI28时钟稳定且HSI28STBIE位被置1时由硬件置1。 软件置HSI28STBR=1时清除该位。 0: 无HSI28时钟稳定中断产生 1: HSI28时钟稳定中断发生
4	PLLSTBF	<b>PLL</b> 时钟稳定中断标志位 当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1。 软件置PLLSTBR=1时清除该位 0: 无PLL时钟稳定中断产生 1: 产生PLL时钟稳定中断
3	HSESTBF	<b>HSE</b> 时钟稳定中断标志位 当外部4 ~ 16 MHz晶体振荡器时钟稳定且HSESTBIE位被置1时由硬件置1。 软件置HSESTBR=1时清除该位。 0: 无HSE时钟稳定中断发生 1: 发生HSE时钟稳定中断
2	HSISTBF	<b>HSI</b> 时钟稳定中断标志位 当内部8 MHz RC振荡器时钟稳定且HSISTBIE位被置1时由硬件置1。 软件置HSISTBR=1时清除该位。 0: 无HSI时钟稳定中断产生 1: 产生HSI时钟稳定中断
1	LSESTBF	<b>LSE</b> 时钟稳定中断标志位 当外部32.768KHz晶体振荡器时钟稳定且LSESTBIE为被置1时由硬件置1。 软件置LSESTBR=1时清除该位。 0: 无LSE时钟稳定中断发生 1: 发生LSE时钟稳定中断
0	LSISTBF	<b>LSI</b> 时钟稳定中断标志位 当内部32kHz RC振荡器时钟稳定且LSISTBIE位被置1时由硬件置1。 软件置LSISTBR = 1时清除该位。 0: 无LSI时钟稳定中断产生 1: 产生LSI时钟稳定中断

#### 4.3.4. APB2 复位控制寄存器 (RCC\_APB2RCR)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										TIMER17	TIMER16	TIMER15			
										RST	RST	RST			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART1 RST	保留	SPI1 RST	TIMER1 RST	保留	ADC1 RST	保留						CFG RST		
	rw		rw	rw		rw							rw		

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER17RST	Timer 17定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER17定时器
17	TIMER16RST	Timer 16定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER16定时器
16	TIMER15RST	Timer 15定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER15定时器
14	USART1RST	USART1 复位 由软件置1或清0。 0: 无复位 1: 复位USART1
13	保留	必须保持复位值
12	SPI1RST	SPI1复位 由软件置1或清0。 0: 无复位 1: 复位SPI1
11	TIMER1RST	Timer 1定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER1定时器
10	保留	必须保持复位值

---

9	ADC1RST	ADC 1复位 由软件置1或清0。 0: 无复位 1: 复位ADC1
8:1	保留	必须保持复位值
0	CFGRST	系统配置复位 由软件置1或清0。 0: 无复位 1: 复位系统配置模块

#### 4.3.5. APB1 复位控制寄存器 (RCC\_APB1RCR)

**GD32F130xx 和 GD32F150xx 产品**

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留 RST	CEC RST	DAC RST	PWR RST	保留				USB RST	I2C2 RST	I2C1 RST	保留			USART 2RST	保留
	rw	rw	rw					rw	rw	rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	保留		WWDG RST	保留		TIMER14 RST	保留		TIMER6 RST	保留		TIMER3 RST	TIMER2 RST	
	rw	rw		rw			rw			rw			rw	rw	

---

位/位域	名称	描述
31	保留	必须保持复位值
30	CECRST	HDMI CEC复位 由软件置1或清0。 0: 无复位 1: 复位HDMI CEC单元
29	DACRST	DAC复位 由软件置1或清0。 0: 无复位 1: 复位DAC单元
28	PWRRST	电源控制复位 由软件置1或清0。 0: 无复位

		1: 复位电源控制单元
27:24	保留	必须保持复位值
23	USBRST	USB复位 由软件置1或清0。 0: 无复位 1: 复位USB
22	I2C2RST	I2C2复位 由软件置1或清0。 0: 无复位 1: 复位I2C2
21	I2C1RST	I2C复位 由软件置1或清0。 0: 无复位 1: 复位I2C1
20:18	保留	必须保持复位值
17	USART2RST	USART2复位 由软件置1或清0。 0: 无复位 1: 复位USART2
16	保留	必须保持复位值
15	SPI3RST	SPI3复位 由软件置1或清0。 0: 无复位 1: 复位SPI3
14	SPI2RST	SPI2复位 由软件置1或清0。 0: 无复位 1: 复位SPI2
13:12	保留	必须保持复位值
11	WWDGRST	窗口看门狗复位 由软件置1或清0。 0: 无复位 1: 复位窗口看门狗
10:9	保留	必须保持复位值
8	TIMER14RST	TIMER14定时器复位 由软件置1或清0。 0: 无复位

## 1: 复位TIMER14定时器

7:5	保留	必须保持复位值
4	TIMER6RST	TIMER6定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER6定时器
3:2	保留	必须保持复位值
1	TIMER3RST	TIMER3定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER3定时器
0	TIMER2RST	TIMER2定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER2定时器

**GD32F170xx 和 GD32F190xx 产品**

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPAMPIVREF RST	CEC RST	DAC RST	PWR RST	保留	CAN2 RST	CAN1 RST	保留	I2C2 RST	I2C1 RST	保留	USART 2RST	保留	USART 2RST	保留	USART 2RST
RW	RW	RW	RW		RW	RW		RW	RW		RW		RW		RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	保留	WWDG RST	保留	LCDRS T	TIMER14 RST	保留	TIMER6 RST	保留	TIMER3 RST	保留	TIMER3 RST	TIMER2 RST		
RW	RW		RW		RW	RW		RW		RW		RW		RW	RW

位/位域	名称	描述
31	OPAMPIVREFRST	OPAMP与IVREF单元复位 由软件置1或清0。 0: 无复位 1: 复位OPAMP与IVREF单元
30	CECRST	HDMI CEC复位 由软件置1或清0。 0: 无复位 1: 复位HDMI CEC单元

29	DACRST	DAC复位 由软件置1或清0。 0: 无复位 1: 复位DAC单元
28	PWRRST	电源控制复位 由软件置1或清0。 0: 无复位 1: 复位电源控制单元
27	保留	必须保持复位值
26	CAN2RST	CAN2复位 由软件置1或清0。 0: 无复位 1: 复位CAN2
25	CAN1RST	CAN1复位 由软件置1或清0。 0: 无复位 1: 复位CAN1
24:23	保留	必须保持复位值
22	I2C2RST	I2C2复位 由软件置1或清0。 0: 无复位 1: 复位I2C2
21	I2C1RST	I2C1复位 由软件置1或清0。 0: 无复位 1: 复位I2C1
20:18	保留	必须保持复位值
17	USART2RST	USART2复位 由软件置1或清0。 0: 无复位 1: 复位USART2
16	保留	必须保持复位值
15	SPI3RST	SPI3复位 由软件置1或清0。 0: 无复位 1: 复位SPI3
14	SPI2RST	SPI2复位 由软件置1或清0。

		0: 无复位 1: 复位SPI2
13:12	保留	必须保持复位值
11	WWDGRST	窗口看门狗复位 由软件置1或清0。 0: 无复位 1: 复位窗口看门狗
10	保留	必须保持复位值
9	LCDRST	LCD复位 由软件置1或清0。 0: 无复位 1: 复位LCD模块
8	TIMER14RST	TIMER14定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER14定时器
7:5	保留	必须保持复位值
4	TIMER6RST	TIMER6定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER6定时器
3:2	保留	必须保持复位值
1	TIMER3RST	TIMER3定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER3定时器
0	TIMER2RST	TIMER2定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER2定时器

#### 4.3.6. AHB 时钟控制寄存器 (RCC\_AHBCCR)

地址偏移: 0x14

复位值: 0x0000 0014

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

保留	TSIEN	保留	PFEN	保留	PDEN	PCEN	PBEN	PAEN	保留
		rw		rw		rw		rw	
15      14      13      12      11      10      9      8      7      6      5      4      3      2      1      0		rw		rw		rw		rw	
	保留.	CRC EN	保留	FMC EN	保留	SRAM EN	保留	DMA1 EN	
		rw		rw		rw		rw	

位/位域	名称	描述
31:25	保留	必须保持复位值
24	TSIEN	<p>TSI时钟使能</p> <p>由软件置1或清0。</p> <p>0: TSI时钟关闭</p> <p>1: TSI时钟开启</p>
23	保留	必须保持复位值
22	PFEN	<p>GPIOF时钟使能</p> <p>由软件置1或清0。</p> <p>0: GPIOF时钟关闭</p> <p>1: GPIOF时钟开启</p>
21	保留	必须保持复位值
20	PDEN	<p>GPIOD时钟使能</p> <p>由软件置1或清0。</p> <p>0: GPIOD时钟关闭</p> <p>1: GPIOD时钟开启</p>
19	PCEN	<p>GPIOC时钟使能</p> <p>由软件置1或清0。</p> <p>0: GPIOC时钟关闭</p> <p>1: GPIOC时钟开启</p>
18	PBEN	<p>GPIOB时钟使能</p> <p>由软件置1或清0。</p> <p>0: GPIOB时钟关闭</p> <p>1: GPIOB时钟开启</p>
17	PAEN	<p>GPIOA时钟使能</p> <p>由软件置1或清0。</p> <p>0: GPIOA时钟关闭</p> <p>1: GPIOA时钟开启</p>
16:7	保留	必须保持复位值
6	CRCEN	CRC 时钟使能

		由软件置1或清0。
		0: CRC时钟关闭
		1: CRC时钟开启
5	保留	必须保持复位值
4	FMCEN	FMC时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的FMC时钟。 0: 关闭睡眠模式下的FMC时钟 1: 开启睡眠模式下的FMC时钟
3	保留	必须保持复位值
2	SRAMEN	SRAM接口时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的SRAM时钟。 0: 关闭睡眠模式下的SRAM接口时钟 1: 开启睡眠模式下的SRAM接口时钟
1	保留	必须保持复位值
0	DMA1EN	DMA1时钟使能 由软件置1或清0。 0: 关闭DMA1时钟 1: 开启DMA1时钟

#### 4.3.7. APB2 时钟控制寄存器 (RCC\_APB2CCR)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										TIMER17	TIMER16	TIMER15			
										EN	EN	EN			
										rw	rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART1 EN	保 留	SPI1 EN	TIMER1 EN	保留	ADC1 EN	保留						CFGCMP EN		
	rw		rw	rw		rw								rw	

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER17EN	TIMER17定时器时钟使能 由软件置1或清0。 0: 关闭TIMER17定时器时钟

		1: 开启 <b>TIMER17</b> 定时器时钟
17	<b>TIMER16EN</b>	<b>TIMER16</b> 定时器时钟使能 由软件置1或清0。 0: 关闭 <b>TIMER16</b> 定时器时钟 1: 开启 <b>TIMER16</b> 定时器时钟
16	<b>TIMER15EN</b>	<b>TIMER15</b> 定时器时钟使能 由软件置1或清0。 0: 关闭 <b>TIMER15</b> 定时器时钟 1: 开启 <b>TIMER15</b> 定时器时钟
15	保留	必须保持复位值
14	<b>USART1EN</b>	<b>USART1</b> 时钟使能 由软件置1或清0。 0: 关闭 <b>USART1</b> 时钟 1: 开启 <b>USART1</b> 时钟
13	保留	必须保持复位值
12	<b>SPI1EN</b>	<b>SPI1</b> 时钟使能 由软件置1或清0。 0: 关闭 <b>SPI1</b> 时钟 1: 开启 <b>SPI1</b> 时钟
11	<b>TIMER1EN</b>	<b>TIMER1</b> 定时器时钟使能 由软件置1或清0。 0: 关闭 <b>TIMER1</b> 定时器时钟 1: 开启 <b>TIMER1</b> 定时器时钟
10	保留	必须保持复位值
9	<b>ADC1EN</b>	<b>ADC 1</b> 接口时钟使能 由软件置1或清0。 0: 关闭 <b>ADC 1</b> 接口时钟 1: 开启 <b>ADC 1</b> 接口时钟
8:1	保留	必须保持复位值
0	<b>CFGCMPEN</b>	系统配置与比较器时钟使能 由软件置1或清0。 0: 关闭系统配置与比较器模块时钟 1: 开启系统配置与比较器模块时钟

#### 4.3.8. APB1 时钟控制寄存器 (RCC\_APB1CCR)

##### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CEC EN	DAC EN	PWR EN	保留				USB EN	I2C2 EN	I2C1 EN	保留				USART 2EN
	rw	rw	rw					rw	rw	rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	保留		WWDG EN	保留		TIMER14 EN	保留			TIMER6 EN	保留		TIMER3 EN	TIMER2 EN
	rw			rw			rw				rw			rw	rw

位/位域	名称	描述
31	保留	必须保持复位值
30	CECEN	HDMI CEC接口时钟使能 由软件置1或清0。 0: 关闭HDMI CEC接口时钟 1: 开启HDMI CEC接口时钟
29	DACEN	DAC接口时钟使能 由软件置1或清0。 0: 关闭DAC接口时钟 1: 开启DAC接口时钟
28	PWREN	电源接口时钟使能 由软件置1或清0。 0: 关闭电源接口时钟 1: 开启电源接口时钟
27:24	保留	必须保持复位值
23	USBEN	USB时钟使能 由软件置1或清0。 0: 关闭USB时钟 1: 开启USB时钟
22	I2C2EN	I2C2时钟使能 由软件置1或清0。 0: 关闭I2C2时钟

		1: 开启I2C2时钟
21	I2C1EN	I2C1时钟使能 由软件置1或清0。 0: 关闭I2C1时钟 1: 开启I2C1时钟
20:18	保留	必须保持复位值
17	USART2EN	USART2时钟使能 由软件置1或清0。 0: 关闭USART2时钟 1: 开启USART2时钟
16	保留	必须保持复位值
15	SPI3EN	SPI3时钟使能 由软件置1或清0。 0: 关闭SPI3时钟 1: 开启SPI3时钟
14	SPI2EN	SPI2时钟使能 由软件置1或清0。 0: 关闭SPI2时钟 1: 开启SPI2时钟
13:12	保留	必须保持复位值
11	WWDGEN	窗口看门狗时钟使能 由软件置1或清0。 0: 关闭窗口看门狗时钟 1: 开启窗口看门狗时钟
10:9	保留	必须保持复位值
8	TIMER14EN	TIMER14定时器时钟使能 由软件置1或清0。 0: 关闭TIMER14定时器时钟 1: 开启TIMER14定时器时钟
7:5	保留	必须保持复位值
4	TIMER6EN	TIMER6定时器时钟使能 由软件置1或清0。 0: 关闭TIMER6定时器时钟 1: 开启TIMER6定时器时钟
3:2	保留	必须保持复位值
1	TIMER3EN	TIMER3定时器时钟使能 由软件置1或清0。

		0: 关闭TIMER3定时器时钟 1: 开启TIMER3定时器时钟
0	TIMER2EN	TIMER2定时器时钟使能 由软件置1或清0。 0: 关闭TIMER2定时器时钟 1: 开启TIMER2定时器时钟

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPAMPIVREFEN	CEC EN	DAC EN	PWR EN	保 留	CAN2EN	CAN1EN	保 留	I2C2 EN	I2C1 EN	保 留	USART 2EN	保 留			
rw	rw	rw	rw		rw	rw		rw	rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	保 留	WWDG EN	保 留	LCDEN	TIMER14 EN	保 留	TIMER6 EN	保 留	TIMER3 EN	TIMER2 EN				
rw	rw		rw		rw	rw		rw		rw		rw	rw		

位/位域	名称	描述
31	OPAMPIVREFEN	OPAMP与IVREF时钟使能 由软件置1或清0。 0: 关闭OPAMP与IVREF接口时钟 1: 开启OPAMP与IVREF接口时钟
30	CECEN	HDMI CEC接口时钟使能 由软件置1或清0。 0: 关闭HDMI CEC接口时钟 1: 开启HDMI CEC接口时钟
29	DACEN	DAC接口时钟使能 由软件置1或清0。 0: 关闭DAC接口时钟 1: 开启DAC接口时钟
28	PWREN	电源接口时钟使能 由软件置1或清0。 0: 关闭电源接口时钟 1: 开启电源接口时钟
27	保留	必须保持复位值

26	CAN2EN	CAN2时钟使能 由软件置1或清0。 0: 关闭CAN2时钟 1: 开启CAN2时钟
25	CAN1EN	CAN1时钟使能 由软件置1或清0。 0: 关闭CAN1时钟 1: 开启CAN1时钟
24:23	保留	必须保持复位值
22	I2C2EN	I2C2时钟使能 由软件置1或清0。 0: 关闭I2C2时钟 1: 开启I2C2时钟
21	I2C1EN	I2C1时钟使能 由软件置1或清0。 0: 关闭I2C1时钟 1: 开启I2C1时钟
20:18	保留	必须保持复位值
17	USART2EN	USART2时钟使能 由软件置1或清0。 0:关闭USART2时钟 1:开启USART2时钟
16	保留	必须保持复位值
15	SPI3EN	SPI3时钟使能 由软件置1或清0。 0: 关闭SPI3时钟 1: 开启SPI3时钟
14	SPI2EN	SPI2时钟使能 由软件置1或清0。 0: 关闭SPI2时钟 1: 开启SPI2时钟
13:12	保留	必须保持复位值
11	WWDGGEN	窗口看门狗时钟使能 由软件置1或清0。 0: 关闭窗口看门狗时钟 1: 开启窗口看门狗时钟
10	保留	必须保持复位值

9	LCDEN	LCD 时钟使能 由软件置1或清0。 0: 关闭LCD时钟 1: 使能LCD时钟
8	TIMER14EN	TIMER14定时器时钟使能 由软件置1或清0。 0: 关闭TIMER14定时器时钟 1: 开启TIMER14定时器时钟
7:5	保留	必须保持复位值
4	TIMER6EN	TIMER6定时器时钟使能 由软件置1或清0。 0: 关闭TIMER6定时器时钟 1: 开启TIMER6定时器时钟
3:2	保留	必须保持复位值
1	TIMER3EN	TIMER3定时器时钟使能 由软件置1或清0。 0: 关闭TIMER3定时器时钟 1: 开启TIMER3定时器时钟
0	TIMER2EN	TIMER2定时器时钟使能 由软件置1或清0。 0: 关闭TIMER2定时器时钟 1: 开启TIMER2定时器时钟

#### 4.3.9. 备份域控制寄存器 (RCC\_BDCR)

##### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x20

复位值: 0x0000 0018, 由备份域复位电路复位

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

**注意:** 备份域控制寄存器(BDCR)的LSEEN, LSEBPS, RTCSEL和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器(PWR\_CTLR)中的BKPWE位置1后才能对这些位进行改动。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
保留															BKPRST		
rw																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RTCEN	保留				RTCSRC[1:0]		保留			LSEDR[1:0]		LSEBPS		LSESTB		LSEEN	
rw		rw				rw			rw		rw		r		rw		

位/位域	名称	描述
31:17	保留	必须保持复位值
16	BKPRST	备份域复位 由软件置1或清0。 0: 无复位 1: 复位备份域
15	RTCEN	RTC时钟使能 由软件置1或清0。 0: 关闭RTC时钟 1: 开启RTC时钟
14:10	保留	必须保持复位值
9:8	RTCSRC[1:0]	RTC时钟入口选择 软件置位或清除来控制RTC时钟源。 00: 没有时钟 01: 选择LSE时钟作为RTC时钟源 10: 选择LSI时钟作为RTC时钟源 11: 选择HSE时钟32分频作为RTC时钟源
7:5	保留	必须保持复位值
4:3	LSEDRI[1:0]	LSE驱动能力 软件置位或清除。当复位备份域时，会重装载缺省值。 00: 弱驱动能力 01: 中低驱动能力 10: 中高驱动能力 11: 强驱动能力 (复位后的缺省值) 注: LSEDRI在旁路模式下无效
2	LSEBPS	LSE旁路模式使能 软件置1和清0。 0: 禁止LSE旁路模式 1: 使能LSE旁路模式
1	LSESTB	外部低速振荡器稳定状态位 硬件置1来指示LSE输出时钟是否稳定待用。 0: LSE未稳定 1: LSE已稳定
0	LSEEN	LSE使能 软件置1和清0。 0: 关闭LSE 1: 开启LSE

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x20

复位值: 0x0000 0018, 由备份域复位电路复位

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

**注意:** 备份域控制寄存器(BDCR)的LSEEN, LSEBPS, RTCSEL和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器(PWR\_CTLR)中的BKPWE位置1后才能对这些位进行改动。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留															BKPRST	
rw																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RTCEN	保留			RTCSRC[1:0]		保留			LSEDRI[1:0]		LSEBPS		LSESTB		LSEEN	
rw	rw			rw		rw			rw		r		rw			

位/位域	名称	描述
31:17	保留	必须保持复位值
16	BKPRST	备份域复位 由软件置1或清0。 0: 无复位 1: 复位备份域
15	RTCEN	RTC时钟使能 由软件置1或清0。 0: 关闭RTC时钟 1: 开启RTC时钟
14:10	保留	必须保持复位值
9:8	RTCSRC[1:0]	RTC和LCD时钟入口选择 软件置位或清除来控制RTC和LCD的时钟源。一旦RTC和LCD时钟源被选定，直到下次后备域被复位，它不能被改变。 00: 没有时钟 01: 选择LSE时钟作为RTC/LCD时钟源 10: 选择LSI时钟作为RTC/LCD时钟源 11: 选择HSE时钟32分频作为RTC/LCD时钟源
7:5	保留	必须保持复位值
4:3	LSEDRI[1:0]	LSE驱动能力 软件置位或清除。当复位备份域时，会重装载缺省值。 00: 弱驱动能力 01: 中低驱动能力 10: 中高驱动能力 11: 强驱动能力 (复位后的缺省值)

注：LSEDR在旁路模式下无效

2	LSEBPS	LSE旁路模式使能 软件置1和清0。 0: 禁止LSE旁路模式 1: 使能LSE旁路模式
1	LSESTB	外部低速振荡器稳定状态位 硬件置1来指示LSE输出时钟是否稳定待用。 0: LSE未稳定 1: LSE已稳定
0	LSEEN	LSE使能 软件置1和清0。 0: 关闭LSE 1: 开启LSE

#### 4.3.10. 总体控制/状态寄存器 (RCC\_GCSR)

地址偏移: 0x24

复位值: 0x0C00 0000, 除复位标志外由系统复位清除, 复位标志只能由电源复位清除。

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDG RSTF	IWDG RSTF	SW RSTF	PWR RSTF	EP RSTF	OBL RSTF	RSTFC	V12 RSTF	保留						
rw	rw	rw	rw	rw	rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												LSISTB	LSIEN	r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无低功耗管理复位发生 1: 发生低功耗管理复位
30	WWDGRSTF	窗口看门狗定时器复位标志位 窗口看门狗定时器复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无窗口看门狗复位发生 1: 发生窗口看门狗复位
29	IWDGRSTF	独立看门狗定时器复位标志位

		独立看门狗复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无独立看门狗定时器复位发生 1: 发生独立看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无软件复位发生 1: 发生软件复位
27	PWRRSTF	电源复位标志位 电源复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无电源复位发生 1: 发生电源复位
26	EPRSTF	外部引脚复位标志位 当有外部引脚复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无外部引脚复位发生 1: 发生外部引脚复位
25	OBLRSTF	可选字节装载器复位标志位 可选字节装载器装载字节时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无选项字节装载器复位发生 1: 发生选项字节装载器复位
24	RSTFC	清除复位标志位 由软件置1来清除所有复位标志位。 0: 无作用 1: 清除复位标志位
23	V12RSTF	V12域电源复位标志位 当有V12域电源复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无V12域电源复位发生 1: 发生V12域电源复位
22:2	保留	必须保持复位值.
1	LSISTB	LSI时钟稳定状态位 该位由硬件置1指示LSI输出时钟是否稳定待用。 0: LSI时钟未稳定 1: LSI时钟已稳定
0	LSIEN	LSI时钟使能

软件置1和清0。

0: 关闭LSI时钟

1: 开启LSI时钟

#### 4.3.11. AHB 复位控制寄存器 (RCC\_AHBRCR)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						TSIRST	保留	PFRST	保留	PDRST	PCRST	PBRST	PARST	保留	
rw							rw			rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.															

位/位域	名称	描述
31:25	保留	必须保持复位值
24	TSIRST	TSI单元复位 由软件置1或清0。 0: 无作用 1: 复位TSI单元
23	保留	必须保持复位值
22	PFRST	GPIOF复位 由软件置1或清0。 0: 无作用 1: 复位GPIOF口
21	保留	必须保持复位值
20	PDRST	GPIOD复位 由软件置1或清0. 0: 无作用 1: 复位GPIOD口
19	PCRST	GPIOC复位 由软件置1或清0。 0: 无作用 1: 复位GPIOC口
18	PBRST	GPIOB复位

		由软件置1或清0。
		0: 无作用
		1: 复位GPIOB口
17	PARST	GPIOA复位
		由软件置1或清0。
		0: 无作用
		1: 复位GPIOA口
16:0	保留	必须保持复位值

#### 4.3.12. 总体时钟配置寄存器 2 (RCC\_GCFGR2)

地址偏移: 0x2c

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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														HSEPREDV[3:0]	
rw															

位/位域	名称	描述
31:4	保留	必须保持复位值
3:0	HSEPREDV[3:0]	<p>HSE时钟作为PLL输入源分频因子</p> <p>由软件置1或清0。这些位仅能在PLL关闭时改写。</p> <p><b>注:</b> HSEPREDV的位0 与RCC_GCFGR的位17的功能相同，修改RCC_GCFGR的位17同时改变这里的位0，HSE时钟分频因子为 (HSEPREDV + 1)</p> <p>0000: HSE 作为PLL的输入，不分频</p> <p>0001: HSE 作为PLL的输入2分频</p> <p>0010: HSE 作为PLL的输入3分频</p> <p>0011: HSE 作为PLL的输入4分频</p> <p>0100: HSE 作为PLL的输入5分频</p> <p>0101: HSE 作为PLL的输入6分频</p> <p>0110: HSE 作为PLL的输入7分频</p> <p>0111: HSE 作为PLL的输入8分频</p> <p>1000: HSE 作为PLL的输入9分频</p> <p>1001: HSE 作为PLL的输入10分频</p> <p>1010: HSE 作为PLL的输入11分频</p> <p>1011: HSE 作为PLL的输入12分频</p> <p>1100: HSE 作为PLL的输入13分频</p>

- 1101: HSE 作为PLL的输入14分频
- 1110: HSE 作为PLL的输入15分频
- 1111: HSE 作为PLL的输入16分频

#### 4.3.13. 总体时钟配置寄存器 3 (RCC\_GCFGR3)

##### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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								ADC SEL	保留	CEC SEL	保留			USART1 SEL[1:0]	
								RW	RW	RW				RW	

位/位域	名称	描述
31:9	保留	必须保持复位值
8	ADCSEL	ADC时钟源选择 由软件置1或清0。 0: ADC时钟源选择HSI14时钟 1: ADC时钟源选择APB2时钟2/4/6/8分频
7	保留	必须保持复位值
6	CECSEL	CEC时钟源选择 由软件置1或清0。 0: CEC时钟选择HSI时钟244分频 1: CEC时钟选择LSE时钟
5:2	保留	必须保持复位值
1:0	USART1SEL[1:0]	USART1时钟源选择 由软件置1或清0。 00: USART1时钟选择APB2时钟 01: USART1时钟选择系统时钟 10: USART1时钟选择LSE时钟 11: USART1时钟选择HSI时钟

**GD32F170xx 和 GD32F190xx 产品**

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															HSI28DIV
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							ADC SEL	保留	CEC SEL	保留				USART1 SEL[1:0]	
rw							rw							rw	

位/位域	名称	描述
31:17	保留	必须保持复位值
16	HSI28DIV	CK_HSI28是否2分频 由软件置1或清0。 0: ADC时钟选择HSI28/2 1: ADC时钟选择HSI28
15:9	保留	必须保持复位值
8	ADCSEL	ADC时钟源选择 由软件置1或清0。 0: ADC时钟源选择HSI28时钟或HSI28/2时钟通过HSI28DIV位 1: ADC时钟源选择APB2时钟2/4/6/8分频
7	保留	必须保持复位值
6	CECSEL	CEC时钟源选择 由软件置1或清0。 0: CEC时钟选择HSI时钟244分频 1: CEC时钟选择LSE时钟
5:2	保留	必须保持复位值
1:0	USART1SEL[1:0]	USART1时钟源选择 由软件置1或清0。 00: USART1时钟选择APB2时钟 01: USART1时钟选择系统时钟 10: USART1时钟选择LSE时钟 11: USART1时钟选择HSI时钟

#### 4.3.14. 总体时钟控制寄存器 2 (RCC\_GCCR2)

##### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x34

复位值: 0x0000 XX80 X表示未定义

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSI14CALIB[7:0]								HSI14ADJ[4:0]				保留		HSI14 STB	HSI14 EN
r															
位/位域	名称	描述	r	rw	r	rw									
31:16	保留	必须保持复位值													
15:8	HSI14CALIB[7:0]	HSI14时钟校准 启动时这些位会被自动初始化为出厂校准参数。													
7:3	HSI14ADJ[4:0]	HSI14时钟调整 这些位由软件设定，最终调整值为(HSI14ADJ)位的值加上HSI14CALIB[7:0]位的值。最终调整值应该把HSI14调整到14MHz ±1%。													
2	保留	必须保持复位值。													
1	HSI14STB	HSI14时钟稳定标志位 由硬件置1来指示内部HSI14振荡器已经稳定可用。 0: HSI14振荡器未稳定 1: HSI14振荡器已稳定													
0	HSI14EN	HSI14时钟使能 软件置1和清0。 0: 关闭HSI14时钟 1: 开启HSI14时钟													

##### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x34

复位值: 0x0000 XX80 X表示未定义

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	HSI28CALIB[7:0]	HSI28时钟校准 启动时这些位会被自动初始化为出厂校准参数。
7:3	HSI28ADJ[4:0]	HSI28 时钟调整 这些位由软件设定，最终调整值为(HSI28ADJ)位的值加上HSI28CALIB[7:0]位的值。 最终调整值应该把HSI28 调整到28MHz ±1%。
2	保留	必须保持复位值.
1	HSI28STB	HSI28时钟稳定标志位 由硬件置1来指示内部HSI28振荡器已经稳定可用。 0: HSI28振荡器未稳定 1: HSI28振荡器已稳定
0	HSI28EN	HSI28时钟使能 软件置1和清0。 0: 关闭HSI28时钟 1: 开启HSI28时钟

#### 4.3.15. 总体时钟配置寄存器 4 (RCC\_GCFGR4) (仅适用于 **GD32F170xx and GD32F190xx** 产品)

地址偏移: 0x80

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	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 CKOUT2DIV[5:0] 保留 CKOUT2SRC[2:0]															

位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	CKOUT2DIV[5:0]	CK_OUT2分频器, 来降低CK_OUT2频率 CK_OUT2的选择参考GCFGR4的2:0位。

		0: CK_OUT2 1分频
		1: CK_OUT2 2分频
		2: CK_OUT2 3分频
		...
		63: CK_OUT2 64分频
7:3	保留	必须保持复位值
2:0	CKOUT2SRC[2:0]	CK_OUT2 时钟源选择 软件置位或清零。 000: 没有时钟被选择 001: 选择高速28M内部振荡器时钟 010: 选择低速内部振荡器时钟 011: 选择外部低速振荡器时钟 100: 选择系统时钟 101: 选择内部高速8M振荡器时钟 110: 选择外部高速振荡器时钟 111: 依赖于PLLDV选择(CK_PLL / 2)或的CK_PLL

#### 4.3.16. 附加时钟控制寄存器 (RCC\_ACCR)

地址偏移: 0xF8

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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															
rw															
位/位域	名称	描述													
31:1	保留	必须保持复位值													
0	I2C3EN	I2C3时钟单元使能 由软件置1或清0。 0: 关闭I2C3时钟单元 1: 开启I2C3时钟单元													

#### 4.3.17. 附加复位控制寄存器 (RCC\_ARCR)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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															

rw

位/位域	名称	描述
31:1	保留	必须保持复位值
0	I2C3RST	I2C3单元复位 由软件置1或清0。 0: 无作用 1: 复位I2C3单元

#### 4.3.18. 电源解锁寄存器 (RCC\_VC\_KEY)

地址偏移: 0x100

复位值: 0x0000 0000

该寄存器可以按字节 (8位)、半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
31:0	KEY[31:0]	RCC_PDR_VC 和 RCC_DEEPSLEEP_VC 解锁寄存器 这些位只能被软件写, 读的话全是0。只有在向RCC_VC_KEY 寄存器写 0x1A2B3C4D后, RCC_PDR_VC 和 RCC_DEEPSLEEP_VC 寄存器才能被写。

#### 4.3.19. RCC 深度睡眠模式电压寄存器 (RCC\_DEEPSLEEP\_VC)

##### GD32F130xx 和 GD32F150xx 产品

地址偏移: 0x134

复位值: 0x0000 0000

该寄存器可以按字节 (8位)、半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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

DEEPSLEEP\_VC[2:0]

rw

位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	DEEPSLEEP_VC[2:0]	深度睡眠模式电压寄存器 这些位由软件置位和清除。 000: 在深度睡眠模式下内核电压为1.2V 001: 在深度睡眠模式下内核电压为1.1V 010: 在深度睡眠模式下内核电压为1.0V 011: 在深度睡眠模式下内核电压为0.9V 100~111: 保留

### GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x134

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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															

rw

位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	DEEPSLEEP_VC[2:0]	深度睡眠模式电压寄存器 这些位由软件置位和清除。 000: 在深度睡眠模式下内核电压为1.8V 001: 在深度睡眠模式下内核电压为1.6V 010: 在深度睡眠模式下内核电压为1.4V 011: 在深度睡眠模式下内核电压为1.2V 100~111: 保留

### 4.3.20. RCC 掉电电压选择寄存器 (RCC\_PDR\_S) (仅适用于 GD32F130xx and GD32F150xx 产品)

地址偏移: 0x138

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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

PDR\_S

RW

位	位域	说明
31:1	保留	必须保持复位值
0	PDR_S	掉电电压选择寄存器 由软件置1或清0。 0: 掉电电压为2.6V 1: 掉电电压为1.8V

## 5. 通用和备用功能 I/Os (GPIO 和 AFIOs)

### 5.1. 简介

最多可支持 55 个通用 I/O 管脚, (GPIO), 称为 PA0 ~ PA15 和 PB0 ~ PB15, PC0 ~ PC15, PD2、PF0/PF1, PF4 ~ PF7 设备用其来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。

GPIO 端口和和其他的备用功能(AFs)复用同一管脚, 在特定的封装下获得最大的的灵活性。GPIO 管脚通过配置相关的寄存器可以用作备用功能管脚, 不管是备用功能输入还是备用功能输出。

每一个 GPIO 可以软件配置为输出(推挽或开漏), 输入、外设备用功能或模拟模式。每一个 GPIO 管脚可以配置为上拉、下拉或无上拉/下拉。除了模拟模式外, 所有的管脚 GPIOs 都具备大电流驱动能力。

### 5.2. 主要特性

- 输入/输出方向控制
- 施密特触发器输入功能使能控制
- 每个管脚弱上拉/下拉功能
- 输出推挽/开漏使能控制
- 输出置位/复位控制
- 输出驱动速度选择
- 模拟输入/输出配置
- 备用功能输入/输出配置
- 端口配置锁定

#### GD32F170xx和GD32F190xx产品:

- 单周期输出翻转功能

### 5.3. 功能描述

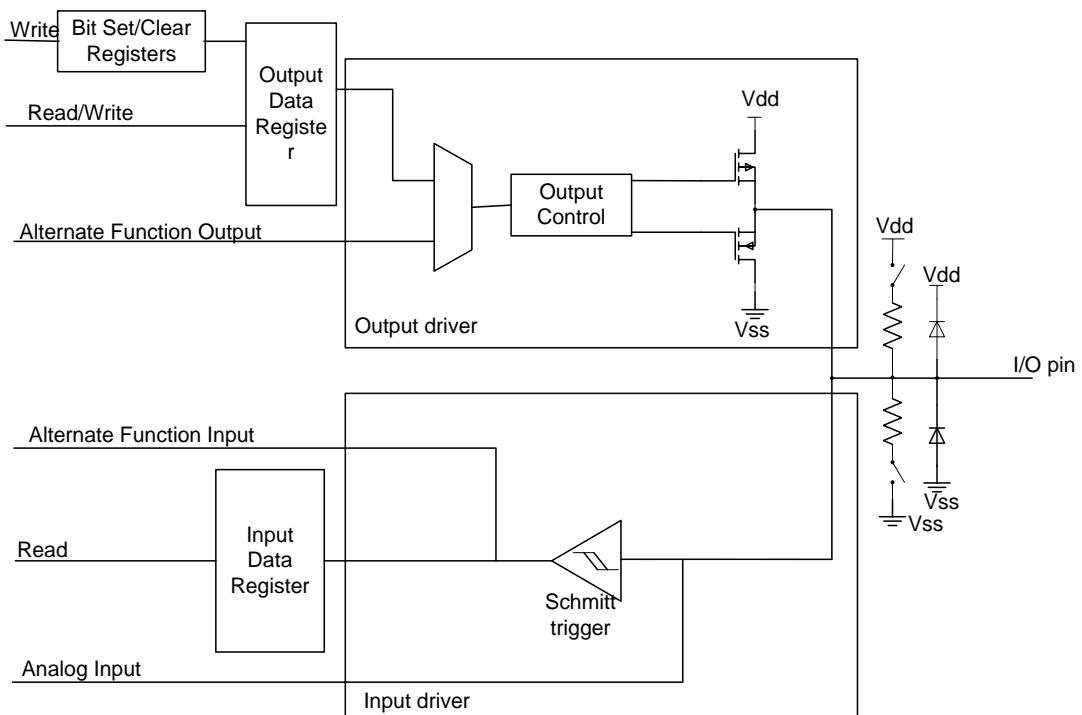
每一个通用I/O端口通过GPIO32位配置寄存器(GPIOx\_CTLR)可以配置为GPIO输入, GPIO输出, AF功能或模拟模式。AFIO输入输出通过AFIO输出使能来选择。当端口是输出(GPIO输出或AFIO输出)的时候, 它可以通过GPIO输出模式寄存器(GPIOx\_OMODE)被配置为推挽或开漏模式。并且这个端口的最大速度可以通过GPIO输出速度寄存器 (GPIOx\_OSPD)配置。每个端口可以通过GPIO上下拉寄存器(GPIOx\_PUPD)配置为悬空(无上拉或下拉), 上拉或下拉功能。

表 5-1. GPIO 配置表

PAD TYPE			CTRLn	OMODEn	PUPDn
GPIO 输入	X	悬空	00	X	00
		上拉			01
		下拉			10
GPIO 输出	推挽	悬空	01	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
AFIO 输入	X	悬空	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	悬空	10	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
ANALOG	X	X	11	X	XX

下图是I/O管脚的基本结构。

图 5-1. 标准 I/O 端口位的基本结构



### 5.3.1. GPIO 管脚配置

在复位期间或复位之后，备用功能并未激活。GPIO 端口都被配置成输入悬空模式，这种输入禁用上拉(PU)/下拉(PD)电阻。但是复位后，串行线调试位于输入 PU/PD 模式。

PA14: SWCLK在输入下拉模式

PA13: SWDIO在输入上拉模式

GPIO管脚可以配置为输入或输出。并且所有的GPIO管脚都有一个内部的弱上拉和弱下拉可以选择。当GPIO管脚可配置为输入管脚时，外部管脚上的数据在每个AHB2时钟周期时都会装载到数据输入寄存器(GPIO\_DIR)。

当GPIO管脚被配置为输出管脚，用户可以配置端口的速度。并选择输出驱动的模式：推挽或开漏模式。输出寄存器的值(GPIO\_DOR)将会在I/O管脚上输出。

当需要对GPIO\_DOR进行按位写操作时不需关中断，用户可以修改一位或几位通过写‘1’到位操作寄存器(GPIO\_BOR，或专用于清0的GPIO\_BCR)，此过程仅需要一个最小的AHB2写访问。其他位不受影响。

### 5.3.2. 备用功能 (AF)

当端口被配置为AFIO(CTLRn到“10”位，CTLRn在GPIOx\_CTLR寄存器里)，这个端口被用作外设备用功能。每个端口可以通过GPIO备用功能选择寄存器(GPIOx\_AFSL、GPIOx\_AFSH)，有16个备用功能可以配置。每个端口的详细的备用功能分配在设备用户手册里详细介绍。

### 5.3.3. 附加功能

一些管脚有附加的功能，他们优先于GPIO寄存器中配置。当用作ADC或DAC复用功能的时候，管脚必须被配置成模拟模式。当用作RTC、WKUPx和振荡器附加功能时，通过相关的RTC、PWR和RCC寄存器，端口类型被自动的设置。当这些端口被用作附加功能时，正常的GPIO功能被禁用。

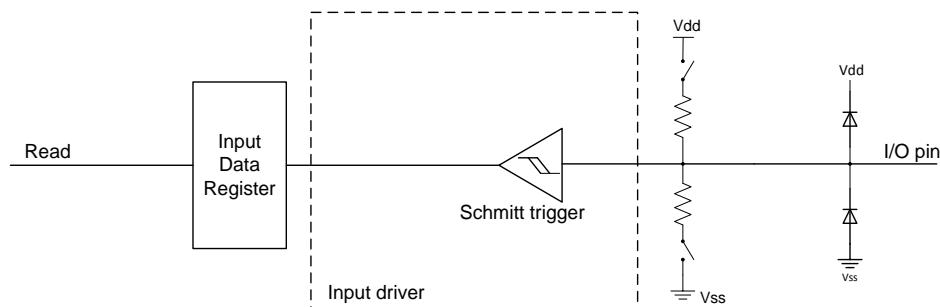
### 5.3.4. 输入配置

当GPIO管脚被用作输入：

- 输入施密特触发器将激活
- 可以选择弱上拉和弱下拉电阻
- I/O 管脚上的数据在每个AHB2周期都会被采样并输入到数据输入寄存器
- 输出缓冲器禁用

下图显示了I/O端口位的配置。

图 5-2. 浮空/上拉/下拉输入配置



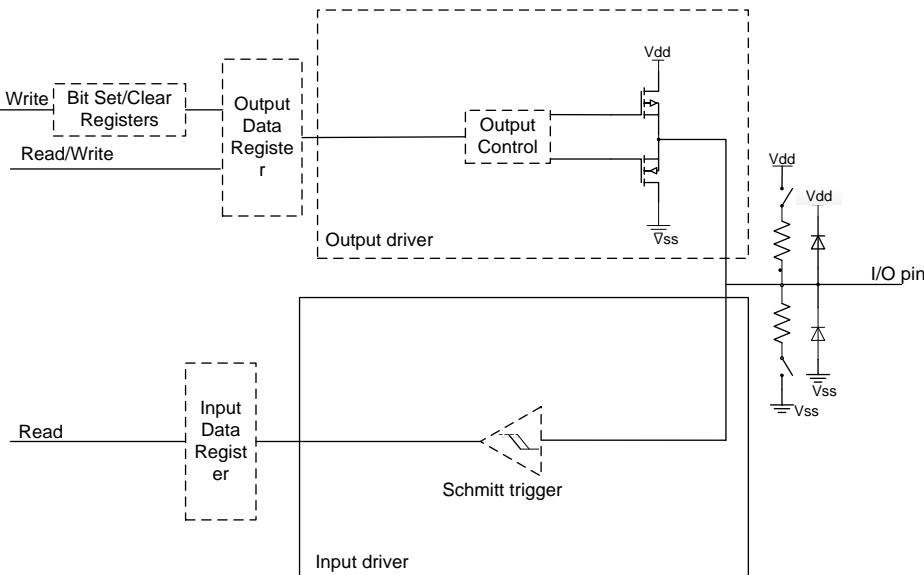
### 5.3.5. 输出配置

当GPIO管脚被用作输出:

- 输入施密特触发器将激活
- 可以选择弱上拉和弱下拉电阻
- 输出缓冲器被使能
  - 开漏模式: 输出寄存器的“0”将使N-MOS管道导通但是输出寄存器“1”将使得管脚处于高阻状态
  - 推挽模式: 输出寄存器的“0”将使N-MOS管道导通但是输出寄存器“1”将使P-MOS管道导通
- 在推挽模式下, 对数据输出寄存器的读访问将返回上次写入的值
- 在开漏模式下, 对数据输入寄存器的访问将返回I/O的状态

下图是 I/O 端口位的输出配置。

图 5-3. 输出配置



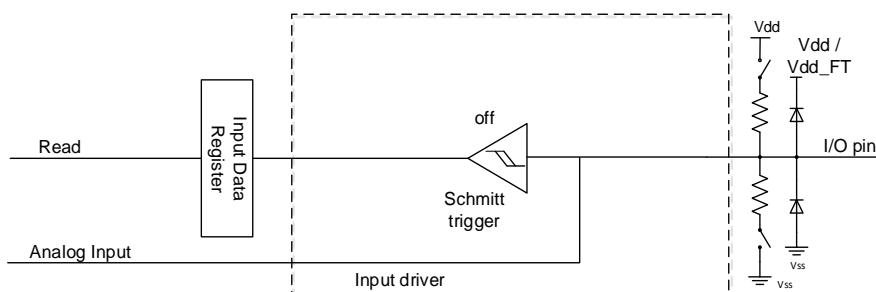
### 5.3.6. 模拟配置

当GPIO管脚被用作模拟配置：

- 弱上拉和下拉电阻禁用
- 输出缓冲器禁用
- 输入施密特触发器未激活
- 读数据输入寄存器将获得“0”

下图是模拟高阻配置

图 5-4. 模拟高阻配置



### 5.3.7. 备用功能(AF) 配置

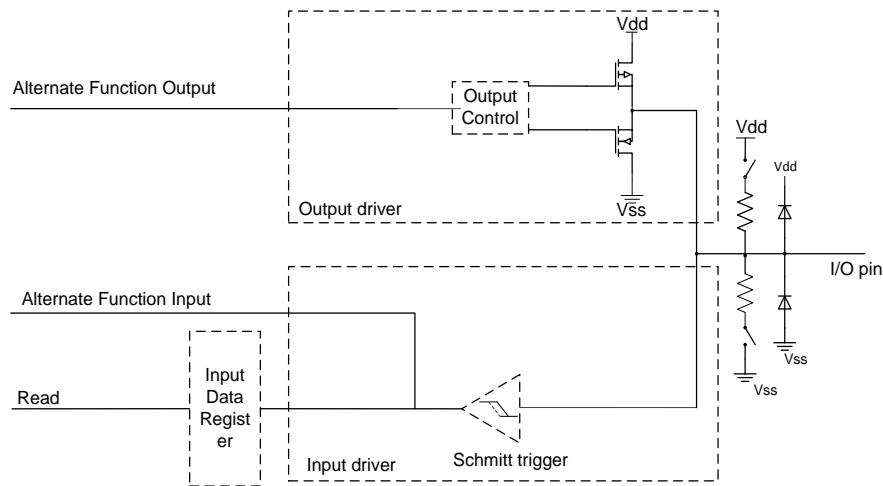
为了适合不同的器件封装，GPIO支持由软件将一些备用功能应用到部分其他管脚上。

当被配置为备用功能：

- 输出缓冲器将开启为开漏或推挽配置
- 输出缓冲器由外设驱动
- 输入施密特触发器将激活
- 可以选择弱上拉和下拉电阻
- I/O管脚上的数据在每个AHB2周期都会被采样并输入数据输入寄存器
- 在开漏模式下，对数据输入寄存器的访问将返回I/O的状态
- 在推挽模式下，对数据输出寄存器的读访问将返回上次写入的值

下图是I/O端口位的备用功能配置

图 5-5. 备用功能配置



### 5.3.8. GPIO 锁定功能

锁定机制可以保护IO的配置。

这个被保护寄存器是 GPIOx\_CTLR, GPIOx\_OMODE, GPIOx\_OSPEED, GPIOx\_PUPD, GPIOx\_AFSEL 和 GPIOx\_AFSH。通过32位锁定寄存器(GPIOx\_LOCKR)可以锁定I/O配置。当LOCK序列已经被应用在相应端口位上，不能改变锁定寄存器的值直到下一次复位。它被推荐用在一个功率模块驱动的配置时。

## 5.4. GPIO 寄存器

### 5.4.1. GPIO 端口控制寄存器 (GPIOx\_CTLR) (x=A..D,F)

地址偏移: 0x00

复位值: 端口 A 0x2800 0000; 其他端口 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTLR15[1:0]	CTLR14[1:0]	CTLR13[1:0]	CTLR12[1:0]	CTLR11[1:0]	CTLR10[1:0]	CTLR9[1:0]	CTLR8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTLR7[1:0]	CTLR6[1:0]	CTLR5[1:0]	CTLR4[1:0]	CTLR3[1:0]	CTLR2[1:0]	CTLR1[1:0]	CTLR0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	CTLR15[1:0]	Pin 15配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述

29:28	CTLR14[1:0]	Pin 14配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
27:26	CTLR13[1:0]	Pin 13配置位 这些位由软件置位和清除。 参照 CTRLO[1:0]的描述
25:24	CTLR12[1:0]	Pin 12配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
23:22	CTLR11[1:0]	Pin 11配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
21:20	CTLR10[1:0]	Pin 10配置位 这些位由软件置位和清除。 参照 CTRLO[1:0]的描述
19:18	CTLR9[1:0]	Pin 9配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
17:16	CTLR8[1:0]	Pin 8配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
15:14	CTLR7[1:0]	Pin 7配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
13:12	CTLR6[1:0]	Pin 6配置位 这些位由软件置位和清除。 参照 CTRLO[1:0]的描述
11:10	CTLR5[1:0]	Pin 5配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
9:8	CTLR4[1:0]	Pin 4配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
7:6	CTLR3[1:0]	Pin 3配置位 这些位由软件置位和清除。 参照CTLR0[1:0]的描述
5:4	CTLR2[1:0]	Pin 2配置位 这些位由软件置位和清除。

参照CTLR0[1:0]的描述

3:2	CTLR1[1:0]	Pin 1配置位 这些位由软件置位和清除。 参照 CTLR0[1:0]的描述
1:0	CTLR0[1:0]	Pin 0配置位 这些位由软件置位和清除。 00: 输入模式 (复位状态) 01: GPIO输出模式 10: 备用功能模式 11: 模拟模式

#### 5.4.2. GPIO 端口输出模式寄存器 (**GPIOx\_OMODE**) (x=A..D,F)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字节 (8位)、半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:16	保留	必须保持复位值
15	OM15	Pin 15 输出模式位 这些位由软件置位和清除。 参照OM0的描述
14	OM14	Pin 14 输出模式位 这些位由软件置位和清除。 参照OM0的描述
13	OM13	Pin 13 输出模式位 这些位由软件置位和清除。 参照OM0的描述
12	OM12	Pin 12 输出模式位 这些位由软件置位和清除。 参照OM0的描述
11	OM11	Pin 11 输出模式位 这些位由软件置位和清除。

参照OM0的描述

10	OM10	Pin 10 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
9	OM9	Pin 9 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
8	OM8	Pin 8 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
7	OM7	Pin 7 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
6	OM6	Pin 6 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
5	OM5	Pin 5 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
4	OM4	Pin 4 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
3	OM3	Pin 3 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
2	OM2	Pin 2 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
1	OM1	Pin 1 输出模式位 这些位由软件置位和清除。 参照 OM0 的描述
0	OM0	Pin 0 输出模式位 这些位由软件置位和清除。 0: 推挽输出模式 (复位) 1: 开漏输出模式

### 5.4.3. GPIO 端口输出速度寄存器 (**GPIOx\_OSPD**) (x=A..D,F)

地址偏移: 0x08

复位值: 端口 A 0x0C00 0000; 其他端口 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	OSPD15[1:0]	Pin 15 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
29:28	OSPD14[1:0]	Pin 14 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
27:26	OSPD13[1:0]	Pin 13 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
25:24	OSPD12[1:0]	Pin 12 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
23:22	OSPD11[1:0]	Pin 11 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
21:20	OSPD10[1:0]	Pin 10 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
19:18	OSPD9[1:0]	Pin 9 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
17:16	OSPD8[1:0]	Pin 8 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
15:14	OSPD7[1:0]	Pin 7 输出最大速度位

		这些位由软件置位和清除。
		参照OSPD0[1:0]的描述
13:12	OSPD6[1:0]	Pin 6 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
11:10	OSPD5[1:0]	Pin 5 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
3:2	OSPD1[1:0]	Pin 1 输出最大速度位 这些位由软件置位和清除。 参照OSPD0[1:0]的描述
1:0	OSPD0[1:0]	Pin 0 输出最大速度位 这些位由软件置位和清除。 x0: 输出最大速度2M (复位状态) 01: 输出最大速度10M 11: 输出最大速度50M

#### 5.4.4. GPIO 端口上拉/下拉寄存器 (**GPIOx\_PUPD**)(x=A..D,F)

地址偏移: 0x0C

复位值: 端口 A0x2400 0000; 其他端口 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]	PUPD14[1:0]	PUPD13[1:0]	PUPD12[1:0]	PUPD11[1:0]	PUPD10[1:0]	PUPD9[1:0]	PUPD8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7[1:0]	PUPD6[1:0]	PUPD5[1:0]	PUPD4[1:0]	PUPD3[1:0]	PUPD2[1:0]	PUPD1[1:0]	PUPD0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
------	----	----

31:30	PUPD15[1:0]	Pin 15 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
29:28	PUPD14[1:0]	Pin 14 上拉或下拉位 这些位由软件置位和清除。 参照 PUPD0[1:0]的描述
27:26	PUPD13[1:0]	Pin 13 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
25:24	PUPD12[1:0]	Pin 12 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
23:22	PUPD11[1:0]	Pin 11 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
21:20	PUPD10[1:0]	Pin 10 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
19:18	PUPD9[1:0]	Pin 9 上拉或下拉位 这些位由软件置位和清除 参照PUPD0[1:0]的描述
17:16	PUPD8[1:0]	Pin 8 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
15:14	PUPD7[1:0]	Pin 7 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
13:12	PUPD6[1:0]	Pin 6 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
11:10	PUPD5[1:0]	Pin 5 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
9:8	PUPD4[1:0]	Pin 4 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
7:6	PUPD3[1:0]	Pin 3 上拉或下拉位 这些位由软件置位和清除。

参照PUPD0[1:0]的描述

5:4	PUPD2[1:0]	Pin 2 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
3:2	PUPD1[1:0]	Pin 1 上拉或下拉位 这些位由软件置位和清除。 参照PUPD0[1:0]的描述
1:0	PUPD0[1:0]	Pin 0 上拉或下拉位 这些位由软件置位和清除。 00: 悬空模式, 无上拉和下拉(复位状态) 01: 上拉模式的端口 10: 下拉模式的端口 11: 保留

#### 5.4.5. GPIO 端口数据输入寄存器 (GPIOx\_DIR) (x=A..D,F)

地址偏移: 0x10

复位值: 0x0000 XXXX

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR1 5	DIR1 4	DIR1 3	DIR1 2	DIR1 1	DIR1 0	DIR 9	DIR 8	DIR 7	DIR 6	DIR 5	DIR 4	DIR 3	DIR 2	DIR 1	DIR 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	的描述s
31:16	保留	必须保持复位值
15:0	DIR[15:0]	Port 输入数据 这些位由硬件置位和清除。 0: 输入信号低电平 1: 输入信号高电平

#### 5.4.6. GPIO 端口数据输出寄存器 (GPIOx\_DOR) (x=A..D,F)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOR15	DOR14	DOR13	DOR12	DOR11	DOR10	DOR9	DOR8	DOR7	DOR6	DOR5	DOR4	DOR3	DOR2	DOR1	DOR0

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DOR[15:0]	Port数据输出 这些位由软件置位和清除。 0: Pin输出低电平 1: Pin输出高电平

#### 5.4.7. GPIO 端口位操作寄存器 (GPIOx\_BOR) (x=A..D,F)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BOR15	BOR14	BOR13	BOR12	BOR11	BOR10	BOR9	BOR8	BOR7	BOR6	BOR5	BOR4	BOR3	BOR2	BOR1	BOR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	CRx[15:0]	Port 清除位 这些位由软件置位和清除。 0: 相应的DORx位无变化 1: 清除相应的DORx位
15:0	BORx[15:0]	Port 置位 这些位由软件置位和清除。 0: 相应的DORx位无变化 1: 置位相应的DORx位

#### 5.4.8. GPIO 端口配置锁定寄存器 (GPIOx\_LOCKR) (x=A,B)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															LKK
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	的描述s
31:17	保留	必须保持复位值
16	LKK	<p>锁定键</p> <p>该位仅仅可以设置Lock Key 写序列置位.总可以读。</p> <p>0: 端口配置锁键没有激活</p> <p>1: 端口配置锁键激活</p> <p>GPIO_LOCKR 寄存器锁定直到一个MCU复位</p> <p>LOCK键写序列</p> <p>写 1→写 0→写1→ 读0→读1</p> <p>注意: LCK[15:0]的值必须保持在 LOCK Key写序列期间</p>
15:0	LKx[15:0]	<p>Port 锁定位 0 ~ 15</p> <p>这些位由软件置位和清除。</p> <p>0: 端口配置没有锁定</p> <p>1: 端口配置锁定</p>

#### 5.4.9. GPIO 端口备用功能选择低位寄存器 (GPIOx\_AFSL) (x=A,B,C)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSL7[3:0]				AFSL6[3:0]				AFSL5[3:0]				AFSL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSL3[3:0]				AFSL2[3:0]				AFSL1[3:0]				AFSL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	AFSL7[3:0]	<p>Pin7 选择备用功能</p> <p>这些位由软件置位和清除。</p> <p>参照AFSL0 [3:0]的描述</p>
27:24	AFSL6[3:0]	Pin 6 选择备用功能

这些位由软件置位和清除。

参照AFSL0 [3:0]的描述

23:20 AFSL5[3:0] Pin 5 选择备用功能

这些位由软件置位和清除。

参照AFSL0 [3:0]的描述

19:16 AFSL4[3:0] Pin 4 选择备用功能

这些位由软件置位和清除。

参照AFSL0 [3:0]的描述

15:12 AFSL3[3:0] Pin 3 选择备用功能

这些位由软件置位和清除。

参照AFSL0 [3:0]的描述

11:8 AFSL2[3:0] Pin 2 选择备用功能

这些位由软件置位和清除。

参照AFSL0 [3:0]的描述

7:4 AFSL1[3:0] Pin 1 选择备用功能

这些位由软件置位和清除。

参照AFSL0 [3:0]的描述

3:0 AFSL0[3:0] Pin 0 选择备用功能

这些位由软件置位和清除。

0000: AF0 被选 (复位值)

0001: AF1 被选

0010: AF2 被选

0011: AF3 被选

0100: AF4 被选 (Port A,B only)

0101: AF5 被选 (Port A,B only)

0110: AF6 被选 (Port A,B only)

0111: AF7 被选 (Port A,B only)

1000 ~ 1111: 保留

#### 5.4.10. GPIO 端口选择备用功能高位寄存器 (GPIOx\_AFSH) (x=A,B,C)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSH15[3:0]				AFSH14[3:0]				AFSH13[3:0]				AFSH12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

AFSH11[3:0]	AFSH10[3:0]	AFSH9[3:0]	AFSH8[3:0]
-------------	-------------	------------	------------

rw

rw

rw

rw

位/位域	名称	描述
31:28	AFSH15[3:0]	Pin 15 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0]的描述
27:24	AFSH14[3:0]	Pin 14 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0]的描述
23:20	AFSH13[3:0]	Pin 13 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0]的描述
19:16	AFSH12[3:0]	Pin 12 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0]的描述
15:12	AFSH11[3:0]	Pin 1 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0]的描述
11:8	AFSH10[3:0]	Pin 10 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0] 的描述
7:4	AFSH9[3:0]	Pin 9 选择备用功能 这些位由软件置位和清除。 参照AFSH8[3:0]的描述
3:0	AFSH8[3:0]	Pin 8 选择备用功能 这些位由软件置位和清除。 0000: AF0 被选 (复位值) 0001: AF1 被选 0010: AF2 被选 0011: AF3 被选 0100: AF4 被选 (Port A,B only) 0101: AF5 被选 (Port A,B only) 0110: AF6 被选 (Port A,B only) 0111: AF7 被选 (Port A,B only)
		1000 ~ 1111: 保留

### 5.4.11. GPIO 端口位清除寄存器 (GPIOx\_BCR) (x=A..D,F)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

W      W      W      W      W      W      W      W      W      W      W      W      W      W      W      W

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CRx[15:0]	Port清除位 这些位由软件置位和清除。 0: 对应的DORx位没改变 1: 清对应的DORx位

### 5.4.12. GPIO 端口位翻转寄存器 (GPIOx\_TGR) (x=A..D,F) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0

W      W      W      W      W      W      W      W      W      W      W      W      W      W      W      W

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TGx[15:0]	Port翻转位 这些位由软件置位和清除。 0: 对应的DORx位没改变 1: 翻转对应的DORx位

## 6. 循环冗余校验计算单元 (CRC)

### 6.1. 简介

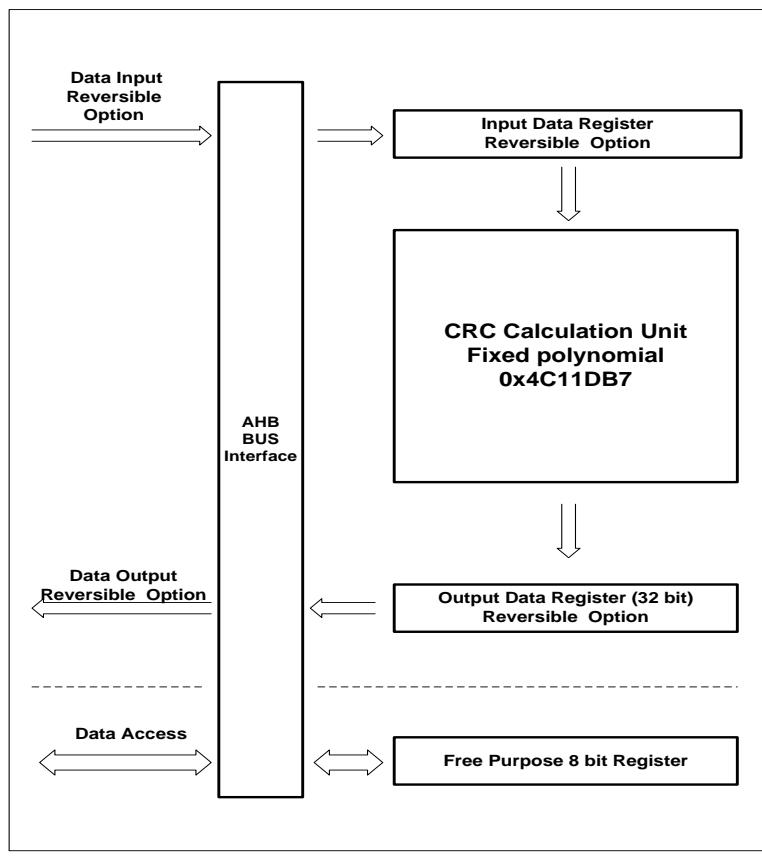
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 计算单元能用固定的多项式来计算 32/16/8 位的 CRC 校验码。

### 6.2. 主要特性

- 32/16/8位数据输入，32位数据输出。从输入数据到得出计算结果，对于32/16/8位的输入数据长度，计算周期分别为4/2/1个AHB时钟周期
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用
- 固定的计算多项式：0x4C11DB7
$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$
该 32 位 CRC 多项式是一个与以太网计算多项式相同的多项式
- 内嵌1个32位的输入缓冲器用于提高总线效率
- 可选I/O数据的逆序与否
- 用户可设置的计算初值，可使计算更灵活

图 6-1. CRC 计算单元框图



### 6.3. 功能描述

该单元有五个功能：

- CRC计算单元可以用来计算32位的原始数据，CRC\_DTR寄存器接收原始数据并存储计算结果。

如果不通过软件设置CRC\_CTLR 寄存器的方式来清除CRC\_DTR 寄存器，新输入的原始数据将会基于前一次CRC\_DTR 寄存器中的结果进行计算。

对于32/16/8位的数据长度，CRC的计算分别要花费4/2/1个AHB的时钟周期。在此期间，因为32位输入缓存的原因，AHB总线将不会被挂起。

- 此模块提供了一个8位的独立寄存器CRC\_FDTR。

CRC\_FDTR与CRC计算无关，任何时候都可以进行独立的读写操作。

- 逆序功能可以交换输入输出数据的位序。

输入数据可选择三种逆序形式。

以原始数据0x1A2B3C4D为例：

- 1) 按字节逆序：

32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x58D43CB2

2) 按半字逆序：

32位数据被分成两组，组内完成颠倒。逆序后的数据为：0xD458B23C

3) 按字逆序：

32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xB23CD458

对于输出数据来说，逆序形式为按字逆序。

例如：当REV\_O=1，计算结果0x22CC4488将被逆序成0x11223344。

■ 多重输入数据大小支持功能可以使用户在组合计算数据上有更大的灵活度。

例如：6位的输入数据能组合成一个字或一个半字，同时也能被组合成3个半字。

■ 用户可配置的初始计算数据功能可让CRC以用户设置值开始计算。

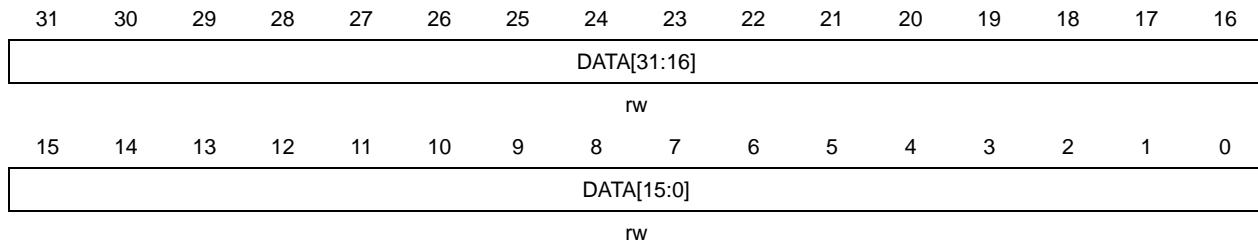
## 6.4. CRC 寄存器

### 6.4.1. CRC 数据寄存器 (CRC\_DTR)

地址偏移：0x00

复位值：0xFFFF FFFF

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:0	DATA[31:0]	<p>CRC 计算结果位</p> <p>软件可读可写。</p> <p>该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次CRC计算的结果。如果输入的计算数据不足4个字节，那么对应的低字节有效。</p>

### 6.4.2. CRC 独立数据寄存器 (CRC\_FDTR)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								FDR[7:0]							

rw

位	位域	描述
31:8	保留	必须保持复位值
7:0	FDR[7:0]	独立数据寄存器位 软件可读可写。 这些位与CRC计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受CRC_CTLR 寄存器的影响。

#### 6.4.3. CRC 控制寄存器 (CRC\_CTLR)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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				Rev_O	Rev_I[1:0]	保留				RESET					

rw rw rs

位	位域	描述
31:8	保留	必须保持复位值
7	Rev_O	翻转输出数据功能 0: 输出数据不翻转 1: 输出数据按32位翻转
6:5	Rev_I[1:0]	翻转输入数据功能 0x0: 输入数据不翻转 0x1: 输入数据按字节翻转 0x2: 输入数据按半字翻转 0x3: 输入数据按字翻转
4:1	保留	必须保持复位值
0	RESET	该位用来复位CRC_DTR寄存器，并将RTC_IDAR寄存器中的更新到CRC_DTR寄存器中，然后自动清零。该位对CRC_FDTR寄存器没有影响。 软件可读写

#### 6.4.4. CRC 初值寄存器 (CRC\_IDTR)

地址偏移: 0x10

复位值: 0xFFFF FFFF

该寄存器只能按字(32位)访问

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

位	位域	描述
31:0	IDAR[31:0]	配置CRC初值  CRC_CTLR寄存器的RESET位置位后, CRC_DTR寄存器的值将被更新为此寄存器的值。

## 7. 中断和事件

### 7.1. 简介

Cortex-M3集成了嵌套式矢量型中断控制器(Nested Vectored Interrupt Controller (NVIC))来实现高效的异常和中断处理。NVIC实现了低延迟的异常和中断处理，以及电源管理控制。它和内核是紧密耦合的。更多关于NVIC的说明请参考《Cortex-M3技术参考手册》。

GD32F1x0 也提供了一个外部中断/事件控制器(EXTI)，包括 23 个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI 有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI 中的每一个边沿检测电路都可以独立配置和屏蔽。

### 7.2. 主要特性

- Cortex-M3系统异常
- 74种可屏蔽的外设中断
- 4位中断优先级配置位—共提供16个中断优先等级
- 高效的中断处理
- 支持异常抢占和咬尾中断
- 将系统从省电模式唤醒
- 外中断中有23个相互独立的边沿检测电路
- 3种触发类型：上升沿触发，下降沿触发和任意沿触发
- 软件中断或事件触发
- 可配置的触发源

### 7.3. 功能描述

#### 7.3.1. NVIC 以及异常/中断处理

ARM Cortex-M3处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序 (ISR)后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。下表列出了所有的异常类型。

表 7-1. Cortex-M3 中的 NVIC 异常类型

异常类型	向量编号	优先级 (a)	向量地址	描述
-	0	-	0x0000_0000	保留
复位	1	-3	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断
硬件故障	3	-1	0x0000_000C	各种硬件级别的故障
存储器管理	4	可编程设置	0x0000_0010	存储器管理
总线故障	5	可编程设置	0x0000_0014	预取指故障, 存储器访问故障
用法故障	6	可编程设置	0x0000_0018	未定义的指令或非法状态
-	7-10	-	0x0000_001C - 0x0000_002B	保留
服务调用	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统的服务调用
调试监控	12	可编程设置	0x0000_0030	调试监控器
-	13	-	0x0000_0034	保留
挂起服务	14	可编程设置	0x0000_0038	可挂起的系统服务请求
系统节拍	15	可编程设置	0x0000_003C	系统节拍定时器
中断	16-89	可编程设置	0x0000_0040 - 0x0000_0164	外设中断 (详见下表)

表 7-2. GD32F130xx 和 GD32F150xx 的中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口式看门狗中断	0x0000_0040
IRQ 1	17	通过 EXTI 线检测的 LVD 中断	0x0000_0044
IRQ 2	18	RTC 全局中断	0x0000_0048
IRQ 3	19	Flash 全局中断	0x0000_004C
IRQ 4	20	RCC 全局中断	0x0000_0050
IRQ 5	21	EXTI 线 0-1 中断	0x0000_0054
IRQ 6	22	EXTI 线 2-3 中断	0x0000_0058
IRQ 7	23	EXTI 线 4-15 中断	0x0000_005C
IRQ 8	24	TSI 全局中断	0x0000_0060
IRQ 9	25	DMA 通道 1 全局中断	0x0000_0064
IRQ 10	26	DMA 通道 2-3 全局中断	0x0000_0068
IRQ 11	27	DMA 通道 4-5 全局中断	0x0000_006C
IRQ 12	28	ADC 和 Comparator 1-2 中断	0x0000_0070

中断编号	向量编号	外设中断描述	向量地址
<b>IRQ 13</b>	29	TIMER1 停止, 更新, 触发和通信中断	0x0000_0074
<b>IRQ 14</b>	30	TIMER1 捕获比较中断	0x0000_0078
<b>IRQ 15</b>	31	TIMER2 全局中断	0x0000_007C
<b>IRQ 16</b>	32	TIMER3 全局中断	0x0000_0080
<b>IRQ 17</b>	33	TIMER6 和 DAC 全局中断	0x0000_0084
<b>IRQ 18</b>	34	保留	0x0000_0088
<b>IRQ 19</b>	35	TIMER14 全局中断	0x0000_008C
<b>IRQ 20</b>	36	TIMER15 全局中断	0x0000_0090
<b>IRQ 21</b>	37	TIMER16 全局中断	0x0000_0094
<b>IRQ 22</b>	38	TIMER17 全局中断	0x0000_0098
<b>IRQ 23</b>	39	I2C1 事件中断	0x0000_009C
<b>IRQ 24</b>	40	I2C2 事件中断	0x0000_00A0
<b>IRQ 25</b>	41	SPI1 全局中断	0x0000_00A4
<b>IRQ 26</b>	42	SPI2 全局中断	0x0000_00A8
<b>IRQ 27</b>	43	USART1 全局中断	0x0000_00AC
<b>IRQ 28</b>	44	USART2 全局中断	0x0000_00B0
<b>IRQ 29</b>	45	保留	0x0000_00B4
<b>IRQ 30</b>	46	CEC 全局中断	0x0000_00B8
<b>IRQ 31</b>	47	保留	0x0000_00BC
<b>IRQ 32</b>	48	I2C1 错误中断	0x0000_00C0
<b>IRQ 33</b>	49	保留	0x0000_00C4
<b>IRQ 34</b>	50	I2C2 错误中断	0x0000_00C8
<b>IRQ 35</b>	51	I2C3 事件中断	0x0000_00CC
<b>IRQ 36</b>	52	I2C3 错误中断	0x0000_00D0
<b>IRQ 37</b>	53	USB 低优先级中断	0x0000_00D4
<b>IRQ 38</b>	54	USB 高优先级中断	0x0000_00D8
<b>IRQ 39-41</b>	55-57	保留	0x0000_00DC- 0x0000_00E4
<b>IRQ 42</b>	58	通过 EXTI 线 18 产生的 USB 唤醒中断	0x0000_00E8
<b>IRQ 43-47</b>	59-63	保留	0x0000_00EC- 0x0000_00FC
<b>IRQ 48</b>	64	DMA 通道 6-7 全局中断	0x0000_0100
<b>IRQ 49-50</b>	65-66	保留	0x0000_0104- 0x0000_0108
<b>IRQ 51</b>	67	SPI3 全局中断	0x0000_010C

表 7-3 GD32F170xx 和 GD32F190xx 的中断向量表

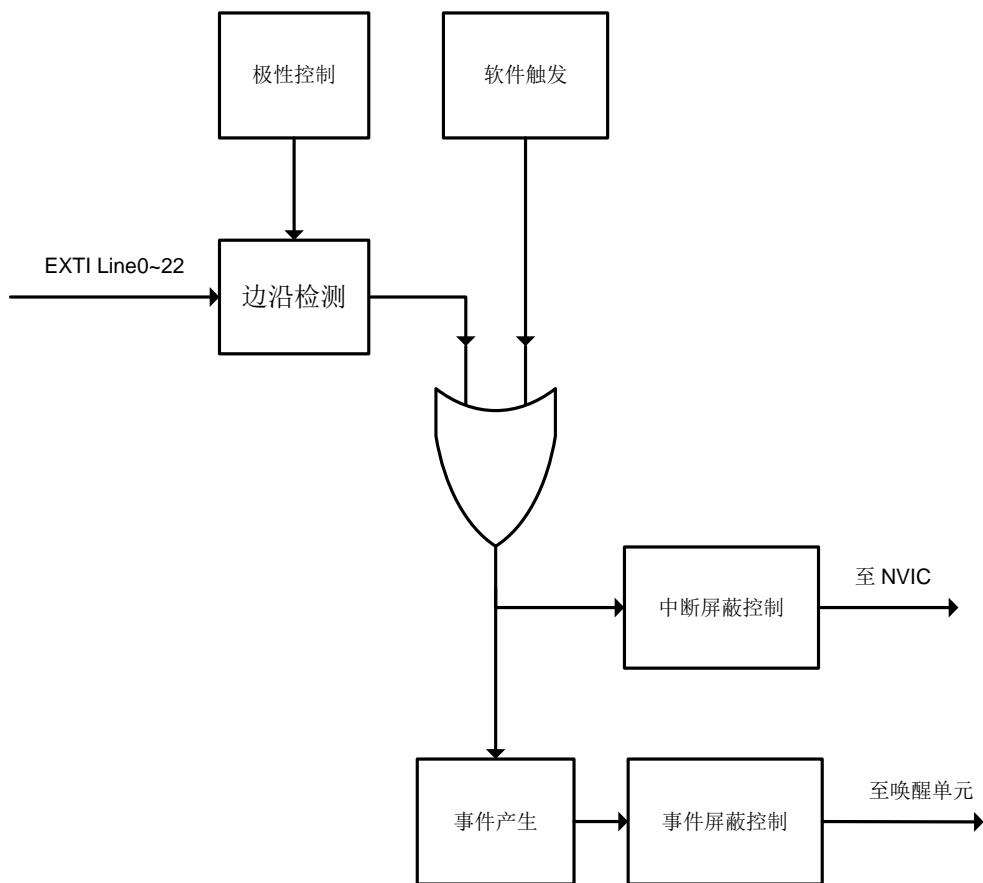
中断编号	向量编 号	外设中断描述	向量地址
<b>IRQ 0</b>	16	窗口式看门狗中断	0x0000_0040
<b>IRQ 1</b>	17	通过 EXTI 线检测的 LVD 中断	0x0000_0044
<b>IRQ 2</b>	18	RTC 全局中断	0x0000_0048
<b>IRQ 3</b>	19	Flash 全局中断	0x0000_004C
<b>IRQ 4</b>	20	RCC 全局中断	0x0000_0050
<b>IRQ 5</b>	21	EXTI 线 0-1 中断	0x0000_0054
<b>IRQ 6</b>	22	EXTI 线 2-3 中断	0x0000_0058
<b>IRQ 7</b>	23	EXTI 线 4-15 中断	0x0000_005C
<b>IRQ 8</b>	24	TSI 全局中断	0x0000_0060
<b>IRQ 9</b>	25	DMA 通道 1 全局中断	0x0000_0064
<b>IRQ 10</b>	26	DMA 通道 2-3 全局中断	0x0000_0068
<b>IRQ 11</b>	27	DMA 通道 4-5 全局中断	0x0000_006C
<b>IRQ 12</b>	28	ADC 和 Comparator 1-2 中断	0x0000_0070
<b>IRQ 13</b>	29	TIMER1 停止, 更新, 触发和通信中 断	0x0000_0074
<b>IRQ 14</b>	30	TIMER1 捕获比较中断	0x0000_0078
<b>IRQ 15</b>	31	TIMER2 全局中断	0x0000_007C
<b>IRQ 16</b>	32	TIMER3 全局中断	0x0000_0080
<b>IRQ 17</b>	33	TIMER6 和 DAC 全局中断	0x0000_0084
<b>IRQ 18</b>	34	保留	0x0000_0088
<b>IRQ 19</b>	35	TIMER14 全局中断	0x0000_008C
<b>IRQ 20</b>	36	TIMER15 全局中断	0x0000_0090
<b>IRQ 21</b>	37	TIMER16 全局中断	0x0000_0094
<b>IRQ 22</b>	38	TIMER17 全局中断	0x0000_0098
<b>IRQ 23</b>	39	I2C1 事件中断	0x0000_009C
<b>IRQ 24</b>	40	I2C2 事件中断	0x0000_00A0
<b>IRQ 25</b>	41	SPI1 全局中断	0x0000_00A4
<b>IRQ 26</b>	42	SPI2 全局中断	0x0000_00A8
<b>IRQ 27</b>	43	USART1 全局中断	0x0000_00AC
<b>IRQ 28</b>	44	USART2 全局中断	0x0000_00B0
<b>IRQ 29</b>	45	保留	0x0000_00B4
<b>IRQ 30</b>	46	CEC 全局中断	0x0000_00B8
<b>IRQ 31</b>	47	保留	0x0000_00BC
<b>IRQ 32</b>	48	I2C1 错误中断	0x0000_00C0
<b>IRQ 33</b>	49	保留	0x0000_00C4
<b>IRQ 34</b>	50	I2C2 错误中断	0x0000_00C8
<b>IRQ 35</b>	51	I2C3 事件中断	0x0000_00CC
<b>IRQ 36</b>	52	I2C3 错误中断	0x0000_00D0
<b>IRQ 37-42</b>	53-58	保留	0x0000_00D4-

中断编号	向量编号	外设中断描述	向量地址
			0x0000_00E8
<b>IRQ43</b>	59	CAN1_TX 中断	0x0000_00EC
<b>IRQ44</b>	60	CAN1_RX0 中断	0x0000_00F0
<b>IRQ45</b>	61	CAN1_RX1 中断	0x0000_00F4
<b>IRQ46</b>	62	CAN1_SCE 中断	0x0000_00F8
<b>IRQ 47</b>	63	LCD 中断	0x0000_00FC
<b>IRQ 48</b>	64	DMA 通道 6-7 全局中断	0x0000_0100
<b>IRQ 49-50</b>	65-66	保留	0x0000_0104- 0x0000_0108
<b>IRQ 51</b>	67	SPI3 全局中断	0x0000_010C
<b>IRQ52-69</b>	68-85	保留	0x0000_0110- 0x0000_0154
<b>IRQ70</b>	86	CAN2_TX 中断	0x0000_0158
<b>IRQ71</b>	87	CAN2_RX0 中断	0x0000_015C
<b>IRQ72</b>	88	CAN2_RX1 中断	0x0000_0160
<b>IRQ73</b>	89	CAN2_SCE 中断	0x0000_0164

### 7.3.2. 外部中断及事件 (EXTI)

EXTI包含23个相互独立的边沿检测电路并且可以向处理器产生28个中断请求或事件唤醒。针对这23个边沿检测电路，EXTI提供3种触发类型：上升沿触发，下降沿触发和任意沿触发。EXTI中每个边沿检测电路都可以分别予以配置或屏蔽。下图为EXTI的内部框图。

图 7-1. EXTI 框图



GD32F130xx和GD32F150xx的EXTI触发源包括来自I/O管脚的16根线以及来自内部模块的8根线。(包括 LVD、RTC、USB、USART、CEC和CMP, 详情请参考表7-4)。GD32F170xx和GD32F190xx的EXTI触发源包括来自I/O管脚的16根线以及来自内部模块的7根线。(包括 LVD、RTC、USART、CEC和CMP, 详情请参考表7-5)。通过配置SYSCFG模块的SYSCFG\_EXTISSx寄存器, 所有的GPIO管脚都可能选作EXTI的触发源, 具体细节请参考第一章中系统配置寄存器的相关内容。

除了中断, EXTI还可以向处理器提供事件信号。The Cortex-M3 内核完全支持等待中断(WFI), 等待事件(WFE)和发送事件(SEV)指令。芯片内部有一个唤醒中断控制器(WIC), 用户可以放心的让处理器和NVIC进入功耗极低的休眠模式, 由WIC来识别中断和事件以及判断优先级。当某些预期的事件发生时, EXTI能唤醒处理器及整个系统, 例如一个特定的I/O 管脚电平翻转或者RTC闹钟动作。

来自内部的触发源CEC和USART也能产生事件以唤醒系统。但是, 为了使处理器能从深度睡眠模式唤醒CPU, 这两个模块要产生一个同步的中断。通过设置EXTI模块的IER和EER寄存器, 可以屏蔽上述两个内部触发线。

### 硬件触发

硬件触发被用来检测外部或内部信号的电压变化。软件需要按如下步骤配置来使用这项功能:

1. 根据应用需要配置SYSCFG模块中的EXTI触发源。

2. 配置EXTI\_RTE寄存器和EXTI\_FTE寄存器以使能相应引脚的上升沿或下降沿检测(软件应当同时配置引脚对应的RTEX和FTEX位以检测该引脚上升沿和下降沿的变化)。
3. 通过配置引脚对应的EXTI\_IER或EXTI\_EER位，使能中断或事件。
4. EXTI开始检测被配置的引脚上的电平变化，当这些引脚上期望的变化被检测到时，相对应的EXTI\_PD寄存器的PDx位将被置位。使能的中断或事件将被触发，软件需要响应该中断或事件并清除相应PDx位。

#### 软件触发

按照如下步骤软件也可以触发EXTI中断或事件：

1. 配置对应的EXTI\_IER或EXTI\_EER位使能中断或事件。
2. 配置EXTI\_SIE寄存器的对应SIE位，对应的PD位将立刻被置1，使能的中断或事件将被触发，软件需要响应该中断或事件并清除相应PDx位。

**表 7-4. GD32F130xx 和 GD32F150xx 的 EXTI 触发源**

EXTI 线编号	触发源	归属
0	PA0 / PB0 / PC0 / PF0	外部
1	PA1 / PB1 / PC1 / PF1	外部
2	PA2 / PB2 / PC2 / PD2	外部
3	PA3 / PB3 / PC3	外部
4	PA4 / PB4 / PC4 / PF4	外部
5	PA5 / PB5 / PC5 / PF5	外部
6	PA6 / PB6 / PC6 / PF6	外部
7	PA7 / PB7 / PC7 / PF7	外部
8	PA8 / PB8 / PC8	外部
9	PA9 / PB9 / PC9	外部
10	PA10 / PB10 / PC10	外部
11	PA11 / PB11 / PC11	外部
12	PA12 / PB12 / PC12	外部
13	PA13 / PB13 / PC13	外部
14	PA14 / PB14 / PC14	外部
15	PA15 / PB15 / PC15	外部
16	LVD	外部
17	RTC 闹钟	外部
18	USB 唤醒	外部
19	RTC 干预和时间戳	外部
20	保留	保留
21	比较器 1 输出	外部
22	比较器 2 输出	外部
23	保留	保留
24	保留	保留
25	USART1 唤醒	内部

EXTI 线编号	触发源	归属
26	保留	保留
27	CEC 唤醒	内部

表 7-5 GD32F170xx 和 GD32F190xx 的 EXTI 触发源

EXTI 线编号	触发源	归属
0	PA0 / PB0 / PC0 / PF0	外部
1	PA1 / PB1 / PC1 / PF1	外部
2	PA2 / PB2 / PC2 / PD2	外部
3	PA3 / PB3 / PC3	外部
4	PA4 / PB4 / PC4 / PF4	外部
5	PA5 / PB5 / PC5 / PF5	外部
6	PA6 / PB6 / PC6 / PF6	外部
7	PA7 / PB7 / PC7 / PF7	外部
8	PA8 / PB8 / PC8	外部
9	PA9 / PB9 / PC9	外部
10	PA10 / PB10 / PC10	外部
11	PA11 / PB11 / PC11	外部
12	PA12 / PB12 / PC12	外部
13	PA13 / PB13 / PC13	外部
14	PA14 / PB14 / PC14	外部
15	PA15 / PB15 / PC15	外部
16	LVD	外部
17	RTC 闹钟	外部
18	保留	保留
19	RTC 干预和时间戳	外部
20	保留	保留
21	比较器 1 输出	外部
22	比较器 2 输出	外部
23	保留	保留
24	保留	保留
25	USART1 唤醒	内部
26	保留	保留
27	CEC 唤醒	内部

## 7.4. 中断 EXTI 寄存器

### 7.4.1. 中断使能寄存器 (EXTI\_IER)

地址偏移: 0x00

复位值: 0x0F90 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		IER27	IER26	IER25	IER24	IER23	IER22	IER21	IER20	IER19	IER18	IER17	IER16
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IER15	IER14	IER13	IER12	IER11	IER10	IER9	IER8	IER7	IER6	IER5	IER4	IER3	IER2	IER1	IER0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	IERx	中断屏蔽控制 0: 第x线中断被屏蔽 1: 第x线中断未被屏蔽

#### 7.4.2. 事件使能寄存器 (EXTI\_EER)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		EER27	EER26	EER25	EER24	EER23	EER22	EER21	EER20	EER19	EER18	EER17	EER16
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EER15	EER14	EER13	EER12	EER11	EER10	EER9	EER8	EER7	EER6	EER5	EER4	EER3	EER2	EER1	EERO
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	EERx	事件屏蔽控制 0: 第x线事件被屏蔽 1: 第x线事件未被屏蔽

#### 7.4.3. 上升沿触发使能寄存器 (EXTI\_RTE)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留						RTE22	RTE21		保留	RTE19	RTE18	RTE17	RTE16
								rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTE15	RTE14	RTE13	RTE12	RTE11	RTE10	RTE9	RTE8	RTE7	RTE6	RTE5	RTE4	RTE3	RTE2	RTE1	RTE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值
22:21	RTE <sub>x</sub>	上升沿触发配置 0: 禁止线 <sub>x</sub> 上升沿触发 1: 允许线 <sub>x</sub> 上升沿触发(中断和事件)
20	保留	必须保持复位值
19	RTE19	上升沿触发配置 0: 禁止线19上升沿触发 1: 允许线19上升沿触发(中断和事件)
18	RTE18	上升沿触发配置 该位仅适用于GD32F150xx产品 0: 禁止线18上升沿触发 1: 允许线18上升沿触发(中断和事件)
17: 0	RTE <sub>x</sub>	上升沿触发配置 0: 禁止线 <sub>x</sub> 上升沿触发 1: 允许线 <sub>x</sub> 上升沿触发(中断和事件)

#### 7.4.4. 下降沿触发使能寄存器 (EXTI\_FTE)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								FTE22	FTE21	保留	FTE19	FTE18	FTE17	FTE16	
								rw	rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTE15	FTE14	FTE13	FTE12	FTE11	FTE10	FTE9	FTE8	FTE7	FTE6	FTE5	FTE4	FTE3	FTE2	FTE1	FTE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31: 23	保留	必须保持复位值
22: 21	FTE <sub>x</sub>	下降沿触发配置 0: 禁止线 <sub>x</sub> 下降沿触发 1: 允许线 <sub>x</sub> 下降沿触发(中断和事件)

---

20	保留	必须保持复位值
19	FTE19	下降沿触发配置 0: 禁止线 <b>19</b> 下降沿触发 1: 允许线 <b>19</b> 下降沿触发(中断和事件)
18	FTE18	下降沿触发配置 该位仅适用于GD32F150xx产品 0: 禁止线 <b>18</b> 下降沿触发 1: 允许线 <b>18</b> 下降沿触发(中断和事件)
17: 0	FTEx	下降沿触发配置 0: 禁止线 <b>x</b> 下降沿触发 1: 允许线 <b>x</b> 下降沿触发(中断和事件)

#### 7.4.5. 软件中断事件寄存器 (EXTI\_SIE)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留										SIE22	SIE21	保留	SIE19	SIE18	SIE17	SIE16
										rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SIE15	SIE14	SIE13	SIE12	SIE11	SIE10	SIE9	SIE8	SIE7	SIE6	SIE5	SIE4	SIE3	SIE2	SIE1	SIE0	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

---

位/位域	名称	描述
31:23	保留	必须保持复位值
22: 21	SIE $\times$	中断/事件软件触发 0: 禁用EXTIx软件中断/事件请求 1: 激活EXTIx软件中断/事件请求
20	保留	必须保持复位值
19: 0	SIE $\times$	中断/事件软件触发 0: 禁用EXTIx软件中断/事件请求 1: 激活EXTIx软件中断/事件请求

#### 7.4.6. 挂起寄存器 (EXTI\_PD)

地址偏移: 0x14

复位值: 未定义

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PD22	PD21	保留	PD19	PD18	PD17	PD16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

位/位域	名称	描述
31: 23	保留	必须保持复位值
22: 21	PDx	中断挂起状态 0: EXIT线x没有发生触发请求 1: EXIT线x已触发, 对这些位写1, 可将其清0。
20	保留	必须保持复位值
19: 0	PDx	中断挂起状态 0: EXIT线x没有发生触发请求 1: EXIT线x已触发, 对这些位写1, 可将其清0。

## 8. DMA 控制器 (DMA)

### 8.1. 简介

直接存储器访问 (DMA) 控制器的功能是自动完成外设与存储器之间的数据传输。通过DMA控制器，数据可以在外设和存储器之间以及存储器与存储器之间快速传输，无需CPU的干涉。从而使CPU可以专注在处理其他系统功能上。DMA控制器有7个通道。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA控制器内部实现了一个仲裁器，用来仲裁多个请求的优先级。

DMA控制器和Cortex™-M3内核均通过系统总线执行数据存取。这两者对总线的竞争使用一种仲裁机制解决。当相同的外设被选中，CPU访问将挂起特定的几个总线周期。总线矩阵中实现了循环仲裁算法来分配DMA与CPU的访问权，它可以确保CPU得到至少一半的系统总线带宽。

### 8.2. 主要特性

- 传输数据长度可编程配置，最大到 65536
- 7 个通道，并且每个通道都可配置
- AHB 和 APB 外设，片上闪存和片上 SRAM 都可以作为访问的源端和目的端
- 每个通道连接固定的硬件 DMA 请求
- DMA 通道请求的优先级由软件配置和通道编号决定
- 支持外设到存储器，存储器到外设，存储器到存储器的数据传输
- 每个通道都具有三个类型的事件标志和一个单独的中断请求
- 每个通道都可配置源端和目的端的传输尺寸

### 8.3. 功能描述

#### 8.3.1. DMA 传输

DMA控制器和外设之间的握手机制是基于请求信号和应答信号的。请求信号表示外设请求DMA访问，而应答信号表示DMA控制器已经访问了外设。当DMA控制器同时收到两个或更多的请求时，哪个请求被服务取决于对应通道优先级高低。每个DMA传输包含两个操作，一是将数据从源端读到DMA，一是将数据从DMA写到目的端。源地址和目的地址是DMA基于DMA\_PBAR<sub>x</sub>, DMA\_MBAR<sub>x</sub>和DMA\_CTLR<sub>x</sub>寄存器的配置自动计算的。详情请参考8.3.3章节。DMA\_RCNT<sub>x</sub> 寄存器控制传输数据的数量。DMA\_CTLR<sub>x</sub>寄存器的PSIZE和MSIZE位，决定每次传输要传多少个字节。假设DMA\_RCNT是4，PNAGA和MNAGA都被置1，结合PSIZE和MSIZE的各种配置，DMA传输的操作详见下表。

表 8-1. DMA 传输操作

Transfer size		Transfer operations	
Source	Destination	Source	Destination
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDDB[15:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

### 8.3.2. 通道之间的仲裁

当同时收到两个或多个请求时，仲裁器根据通道的优先级决定哪一个请求被服务。优先级分两个层次：软件优先级和硬件优先级。软件优先级通过DMA\_CTLRx寄存器的PRIO[1:0]位来配置。软件优先级分“低”、“中”、“高”和“极高”四个等级。而硬件优先级是固定的。编号较低的通道比编号较高的通道有更高的硬件优先级。例如，通道1的硬件优先级高于通道3。仲裁器比较优先级时会优先看软件优先级。如果两个通道的软件优先级不同，软件优先级较高的通道被服务。如果两个通道的软件优先级相同，硬件优先级较高的通道被服务。

### 8.3.3. 地址生成算法

DMA\_CTLRx 寄存器的 PSIZE 和 MSIZE 位用来配置传输数据的尺寸。DMA\_CTLRx 寄存器的 PNAGA 和 MNAGA 位用来配置地址生成算法。DMA 控制器支持两种地址生成算法，分别是固定地址模式和递增地址模式。在固定地址模式下，下一个地址始终等价于当前地址。在递增地址模式下，下一个地址根据传输数据的尺寸在当前地址的基础上递增 1 或 2 或 4。

### 8.3.4. 循环模式

循环模式用于处理循环的数据流（例如，ADC 扫描模式）。通过 DMA\_CTLRx 寄存器的 CIRC 位可以使能该功能。在循环模式下，通道的剩余计数器每当达到 0 时都会自动重装载为初始设定值，所以 DMA 会一直工作。

### 8.3.5. 存储器到存储器模式

通过 DMA\_CTLRx 寄存器的 MEMTOMEM 位可以使能存储器到存储器模式。在此模式下，DMA 通道传输数据时不依赖外设的请求信号。一旦 DMA\_CTLRx 寄存器的 CHEN 位被置 1，DMA 通道就立即开始传输数据，直到 DMA\_RCNTx 寄存器达到 0，DMA 通道才会停止。

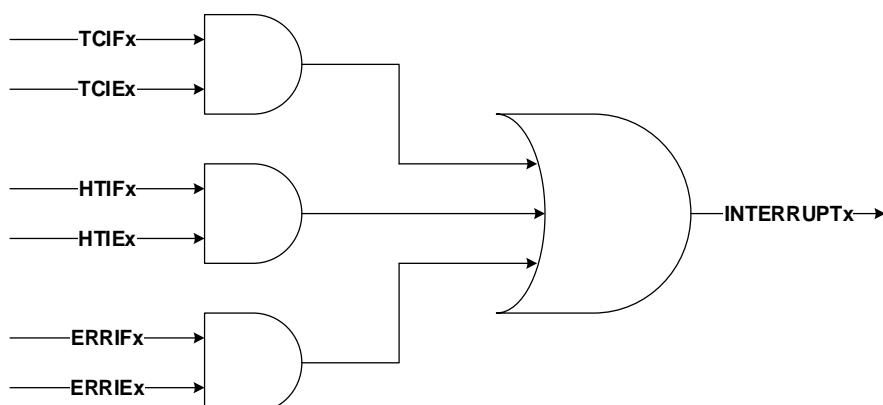
### 8.3.6. 中断请求

每个 DMA 通道都有一个专用的中断。中断事件有三种类型：传输完成，传输完成一半和传输错误。当通道上的任意类型中断事件发生时，如果对应的中断使能位是开启的，就会产生中断。每一个中断事件都在 DMA\_IFR 寄存器中有专用的标志位，在 DMA\_ICR 寄存器中有专用的清除位，在 DMA\_CTLRx 寄存器中有专用的使能位。下表描述了其对应关系。

**表 8-2. DMA 中断事件**

中断事件	标志位	清除位	使能位
传输完成	TCIF	TCIC	TCIE
传输完成一半	HTIF	HTIC	HTIE
传输错误	ERRIF	ERRIC	ERRIE

**图 8-1. DMA 中断产生逻辑**



注意：“x”表示通道编号 ( $x=1\dots 7$ )。

当 DMA 控制器访问保留的地址空间时，传输错误事件发生。此时硬件会自动将 DMA\_CTLRx 寄存器的 CHEN 位清零，通道随之关闭。

### 8.3.7. DMA 通道配置流程

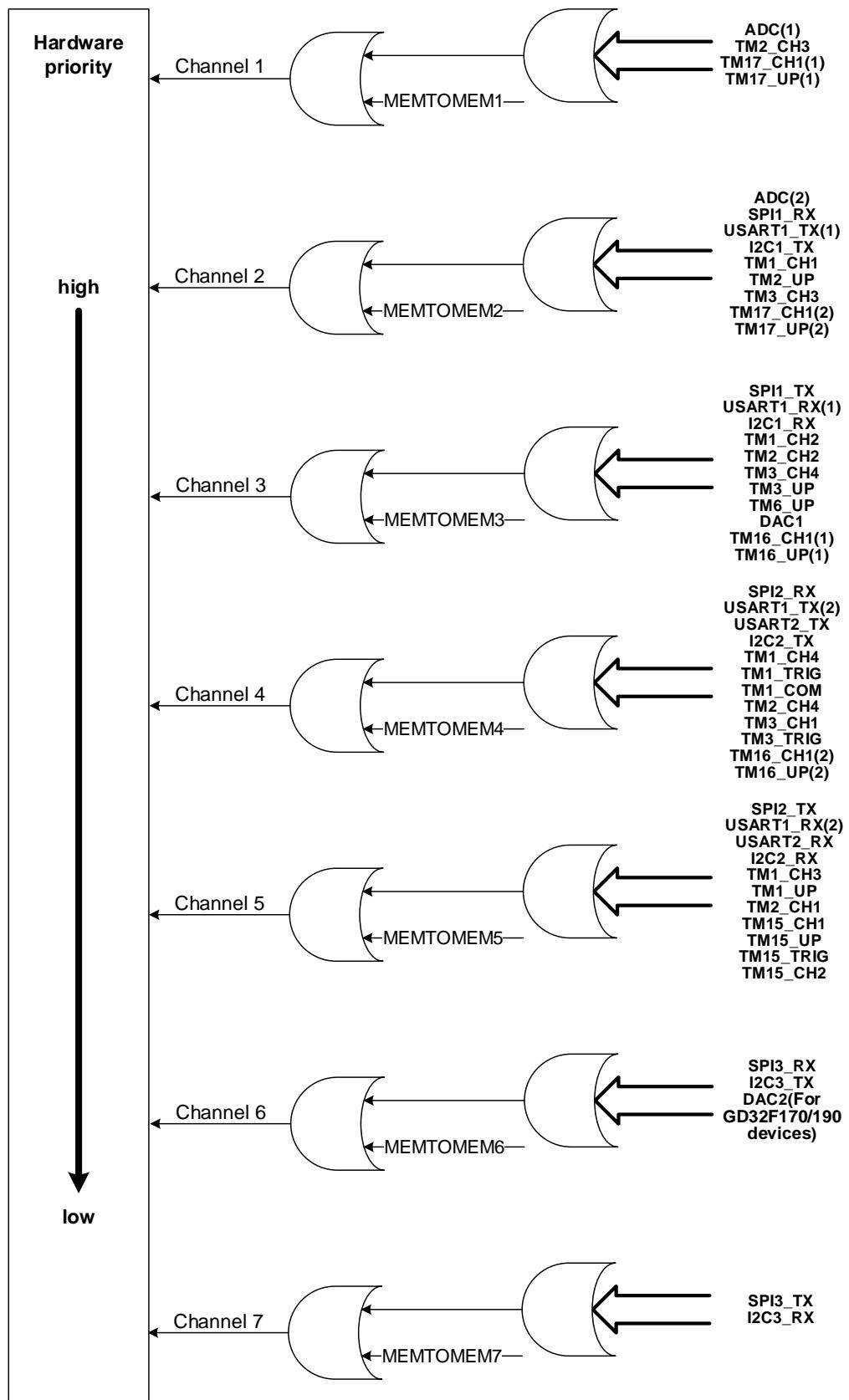
下面给出了配置 DMA 通道的步骤：

1. 通过 DMA\_PBARx 寄存器设置外设地址
2. 通过 DMA\_MBArx 寄存器设置存储器地址
3. 通过 DMA\_RCNTx 寄存器设置传输数据的总数
4. 通过 DMA\_CTLRx 寄存器设置软件优先级, 传输方向, 模式类型, 数据尺寸和中断类型。
5. 将 DMA\_CTLRx 寄存器的 CHEN 位配置成 1

### 8.3.8. DMA 请求映射

几个来自外设的请求可以被映射到同一个 DMA 通道。他们会先经过逻辑或运算再进入 DMA。详情请参考下图。通过设置对应的外设的寄存器，每一个外设的请求都可以独立的开启或关闭。用户必须确保，在同一时间，对于同一通道来说，只有一个外设的请求是开启的。

图 8-2. DMA 请求映射



- 当 SYSCFG\_R1 寄存器中的对应重映射控制位被清 0 时，此请求被映射到该通道

2. 当 SYSCFG\_R1 寄存器中的对应重映射控制位被置 1 时，此请求被映射到该通道

下表列出了每个通道支持的来自外设的请求。

**表 8-3. 各 DMA 通道请求总结**

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
<b>ADC</b>	ADC(1)	ADC(2)	•	•	•	•	•
<b>SPI/I2S</b>	•	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX	SPI3_RX	SPI3_TX
<b>USART</b>	•	USART1_TX(1)	USART1_RX(1)	USART1_RX(2) USART2_TX	USART1_RX(2) USART2_RX	•	•
<b>I<sup>2</sup>C</b>	•	I <sup>2</sup> C1_TX	I <sup>2</sup> C1_RX	I <sup>2</sup> C2_TX	I <sup>2</sup> C2_RX	I <sup>2</sup> C3_TX	I <sup>2</sup> C3_RX
<b>TM1</b>	•	TM1_CH1	TM1_CH2	TM1_CH4 TM1_TRIG TM1_COM	TM1_CH3 TM1_UP	•	•
<b>TM2</b>	TM2_CH3	TM2_UP	TM2_CH2	TM2_CH4	TM2_CH1	•	•
<b>TM3</b>	•	TM3_CH3	TM3_CH4 TM3_UP	TM3_CH1 TM3_TRIG	•	•	•
<b>TM6/DAC</b>	•	•	TM6_UP DAC1	•	•	DAC2(GD32F1 70/190 产品)	•
<b>TM15</b>	•	•	•	•	TM15_CH1 TM15_UP TM15_TRIG TM15_COM	•	•
<b>TM16</b>	•	•	TM16_CH1(1) TM16_UP(1)	TM16_CH1(2) TM16_UP(2)	•	•	•
<b>TM17</b>	TM17_CH1(1) TM17_UP(1)	TM17_CH1(2) TM17_UP(2)	•	•	•	•	•

## 8.4. DMA 寄存器

### 8.4.1. DMA 中断状态寄存器 (DMA\_IFR)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			ERRIF7	HTIF7	TCIF7	GIF7	ERRIF6	HTIF6	TCIF6	GIF6	ERRIF5	HTIF5	TCIF5	GIF5	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF4	HTIF4	TCIF4	GIF4	ERRIF3	HTIF3	TCIF3	GIF3	ERRIF2	HTIF2	TCIF2	GIF2	ERRIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:28	保留	必须保持复位值.
27/23/19/ 15/11/7/3	ERRIFx	通道x (x=1...7) 的错误标志 由硬件置1; 通过配置DMA_ICR寄存器, 软件清0。 0: 通道x没有发生错误 1: 通道x发生错误

26/22/18/	HTIFx	通道x (x=1...7)的传输一半完成标志
14/10/6/2		由硬件置1; 通过配置DMA_ICR寄存器, 软件清0。 0: 通道x一半数目的传输没有完成 1: 通道x一半数目的传输已经完成
25/21/17/	TCIFx	通道x (x=1...7)的传输完成标志
13/9/5/1		由硬件置1; 通过配置DMA_ICR寄存器, 软件清0。 0: 通道x (x=1...7)的传输没有完成 1: 通道x (x=1...7)的传输已经完成
24/20/16/	GIFx	通道x (x=1...7)的全局中断标志
12/8/4/0		由硬件置1; 通过配置DMA_ICR寄存器, 软件清0。 0: 通道x没有发生ERRIF, HTIF或TCIF 1: 通道x至少发生ERRIF, HTIF或TCIF之一

#### 8.4.2. DMA 中断标志清除寄存器 (DMA\_ICR)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
			保留		ERRIC7	HTIC7	TCIC7	GIC7	ERRIC6	HTIC6	TCIC6	GIC6	ERRIC5	HTIC5	TCIC5	GIC5
				w	w	w	w	w	w	w	w	w	w	w	w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ERRIC4	HTIC4	TCIC4	GIC4	ERRIC3	HTIC3	TCIC3	GIC3	ERRIC2	HTIC2	TCIC2	GIC2	ERRIC1	HTIC1	TCIC1	GIC1	
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	

位/位域	名称	描述
31:28	保留	必须保持复位值.
27/23/19/	ERRICx	清除通道x (x=1...7)的错误标志
15/11/7/3		0: 无影响 1: 清零DMA_IFR寄存器的ERRIFx位
26/22/18/	HTICx	清除通道x (x=1...7)的传输一半完成标志
14/10/6/2		0: 无影响 1: 清零DMA_IFR寄存器的HTIFx位
25/21/17/	TCICx	清除通道x (x=1...7)的传输完成标志
13/9/5/1		0: 无影响 1: 清零DMA_IFR寄存器的TCIFx位
24/20/16/	GICx	清除通道x (x=1...7)的全局中断标志
12/8/4/0		0: 无影响 1: 清零DMA_IFR寄存器的GIFx, ERRIFx, HTIFx 和TCIFx位

### 8.4.3. DMA 通道 x 控制寄存器 (DMA\_CTLRx)

$x = 1..7$ , 其中x是通道编号

地址偏移:  $0x08 + 0d20 \times (x - 1)$

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	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	MEMTO MEM	PRIO[1:0]	MSIZE[1:0]	PSIZE[1:0]	MNAGA	PNAGA	CIRC	DIR	ERRIE	HTIE	TCIE	CHEN			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值.
14	MEMTOMEM	存储器到存储器模式 0: 关闭存储器到存储器模式 1: 开启存储器到存储器模式
13:12	PRIO[1:0]	该通道的软件优先级 00: 低 01: 中 10: 高 11: 极高
11:10	MSIZE[1:0]	存储器端的传输数据尺寸 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留
9:8	PSIZE[1:0]	外设端的传输数据尺寸 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留
7	MNAGA	存储器端的地址生成算法 0: 固定地址模式 1: 递增地址模式
6	PNAGA	外设端的地址生成算法 0: 固定地址模式 1: 递增地址模式

5	CIRC	循环模式 0: 关闭循环模式 1: 开启循环模式
4	DIR	通道的数据传输方向 0: 从外设端读数据, 写往存储器端 1: 从存储器端读数据, 写往外设端
3	ERRIE	通道错误中断使能位 0: 关闭通道错误中断 1: 开启通道错误中断
2	HTIE	通道传输一半完成中断使能位 0: 关闭传输一半完成中断 1: 开启传输一半完成中断
1	TCIE	通道传输完成中断使能位 0: 关闭传输完成中断 1: 开启传输完成中断
0	CHEN	通道使能 0: 关闭通道 1: 开启通道

#### 8.4.4. DMA 通道 x 剩余计数器 (DMA\_RCNTx)

$x = 1..7$ , 其中x是通道编号

地址偏移:  $0x0C + 0d20 \times (x - 1)$

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	RCNT[15:0]	剩余计数器 该寄存器表示DMA还剩多少数据没传完。它只在相应通道关闭时才接受写访问。一旦数据传输启动, 它就变为只读寄存器, 在每个DMA传输之后递减1。如果该寄存器为0, 无论通道开启与否, 都不会有数据传输。当全部的DMA传输操作完成后, 根据之前通道重装载的相应配置, RCNT可以被自动重装载为初始值。

#### 8.4.5. DMA 通道 x 外设端基地址寄存器 (DMA\_PBARx)

$x = 1..7$ , 其中x是通道编号

地址偏移:  $0x10 + 0d20 \times (x - 1)$

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

**注意:** 该寄存器应在相应通道关闭时配置。

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

位/位域	名称	描述
31:0	PBAR[31:0]	外设端基地址 当PSIZE为01时, PBAR[0]被忽略, 访问自动与16位地址对齐。 当PSIZE为10时, PBAR[1:0]被忽略, 访问自动与32位地址对齐。

#### 8.4.6. DMA 通道 x 存储器端基地址寄存器 (DMA\_MBArx)

$x = 1..7$ , 其中x是通道编号

地址偏移:  $0x14 + 0d20 \times (x - 1)$

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

**注意:** 该寄存器应在相应通道关闭时配置。

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

位/位域	名称	描述
31:0	MBAR[31:0]	存储器端基地址 当MSIZE为01时, MBAR[0]被忽略, 访问自动与16位地址对齐。 当MSIZE为10时, MBAR[1:0]被忽略, 访问自动与32位地址对齐。

## 9. 定时器 (TIMERx)

定时器(TIMERx)分为六个类型

定时器	TIMER1	TIMER2/3	TIMER6	TIMER14	TIMER15	TIMER16/1 7
类型	高级	通用(1)	基本	通用(2)	通用(3)	通用(4)
预分频器	16 位	16 位	16 位	16 位	16 位	16 位
计数模式	向上, 向下, 中间对齐	向上, 向下, 中间对齐	只有向上	只有向上	只有向上	只有向上
可重复性	•	×	×	×	•	•
捕获/比较 通道数	4	4	0	1	2	1
输出互补 & 时滞	•	×	×	×	•	•
中止输入	•	×	×	×	•	•
单脉冲	•	•	×	×	•	•
正交译码器	•	•	×	×	×	×
从设备控制器	•	•	×	×	•	×
内部连接	• <sup>(1)</sup>	• <sup>(2)</sup>	TRGO TO DAC	×	• <sup>(3)</sup>	×
Debug 模式	•	•	•	•	•	•
DMA	•	•	•	×	•	•

(1)TIMER1: tm15\_trgo->it0;tm2\_trgo->it1;tm3\_trgo->it2;0->it3

(2)TIMER2: tm1\_trgo->it0; tm15\_trgo->it1; tm3\_trgo->it2; 0->it3;

TIMER3: tm1\_trgo->it0; tm2\_trgo->it1; tm15\_trgo->it2; 0->it3;

(3)TIMER15: tm2\_trgo->it0; tm3\_trgo->it1; 0->it2; 0->it3;

## 9.1. 高级定时器 (TIMER1)

### 9.1.1. 简介

高级定时器TIMER1可以用于大量高级控制的场合。它包括一个16位的向上/向下计数器、4个16位的捕捉/比较寄存器(TIMER1\_CHCC)、1个16位的自动装载计数器寄存器(TIMER1\_CARL)和一些控制寄存器。它适合多种用途，包括通用定时器，输入信号脉冲宽度测量或者产生输出波形例如单脉冲输出或PWM输出。TIMER1支持带有两个输入的编码器接口。

高级定时器(TIMER1)和通用定时器(TIMERx)完全独立，它们不共享资源但是可以同步。

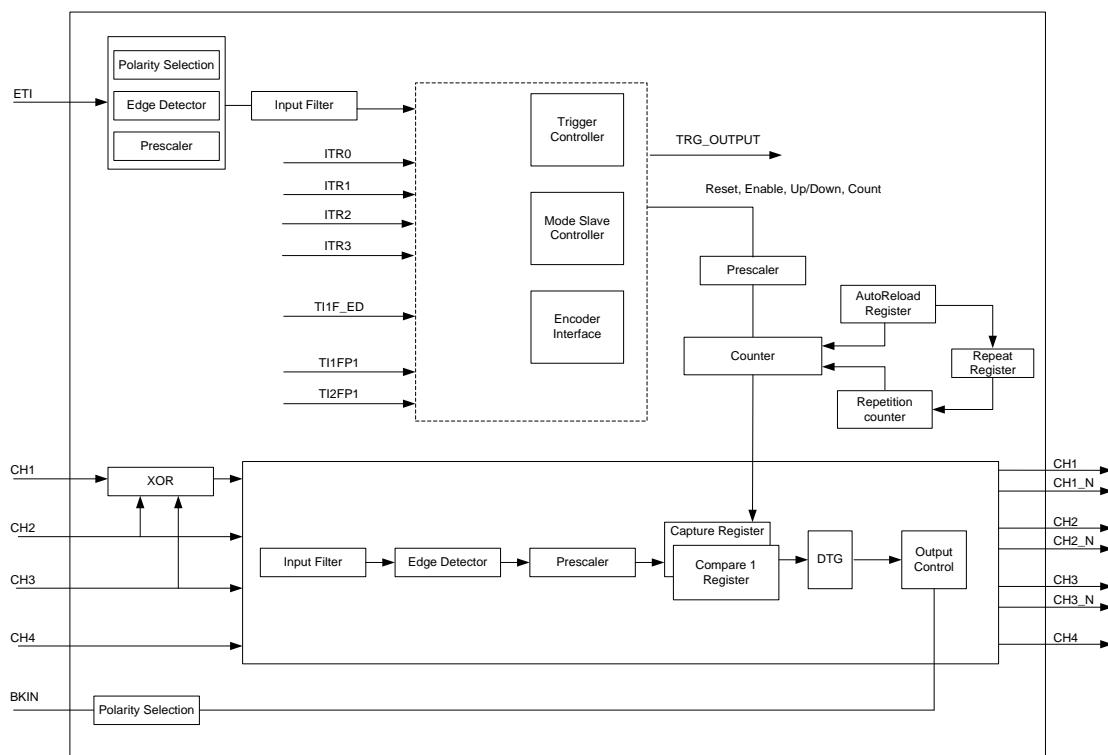
### 9.1.2. 主要特性

- 16位向下、向上、向下/上自动装载计数器
- 16位可编程的预分频器，计数器时钟频率的分频系数为1~65536之间的任意数值
- 多达4个独立通道，支持以下功能：输入捕获，输出比较，PWM波形产生(边缘或中间对齐模式)，单脉冲模式输出
- 允许在指定数目的计数器周期之后更新定时器寄存器并重新计数
- 中止（Break）输入信号可以将模块的输出信号置于一个指定状态
- 更新事件，输入捕获事件，输出比较事件，中止输入，触发事件可以产生中断/DMA
- 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
- 编码器接口控制器带有两个正交输入译码器
- TIMER1主从模式控制器

### 9.1.3. 功能描述

下图提供了高级定时器内部配置细节

图 9-1. 高级定时器框图



### 预分频计数器

预分频器可以将定时器(PCLK)的时钟频率按1到65536之间的任意值分频给计数器(CNT\_CLK)使用。分频系数受预分频寄存器TIMER1\_PSC控制，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 9-2. 当预分频器的参数从 1 变到 2 时，计数器的时序图

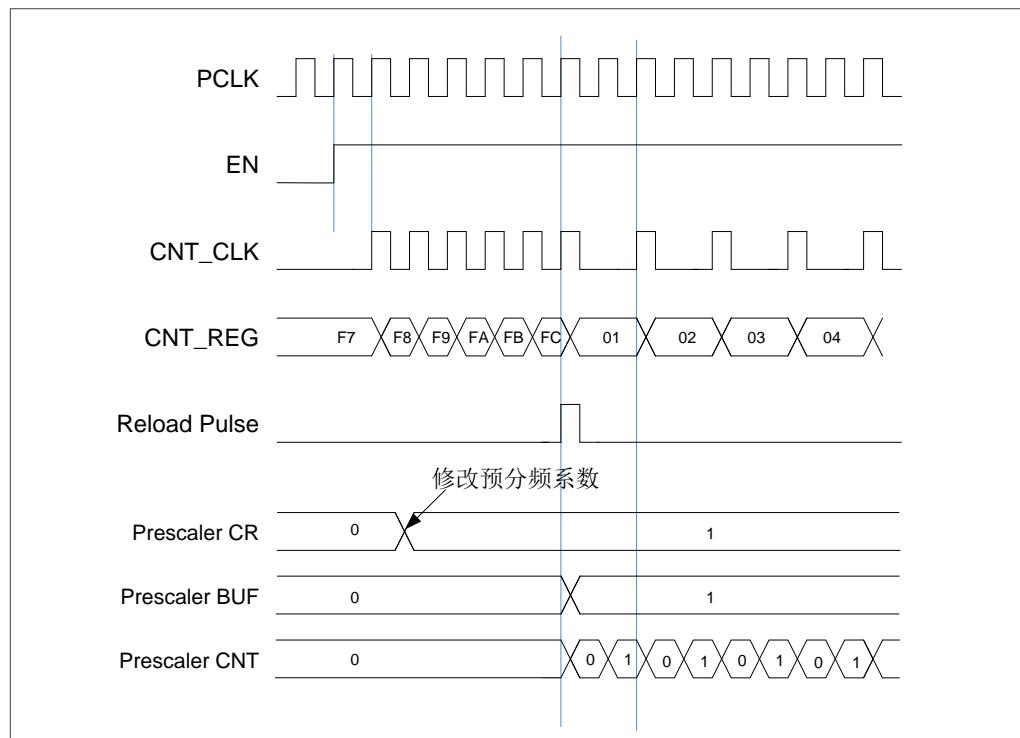
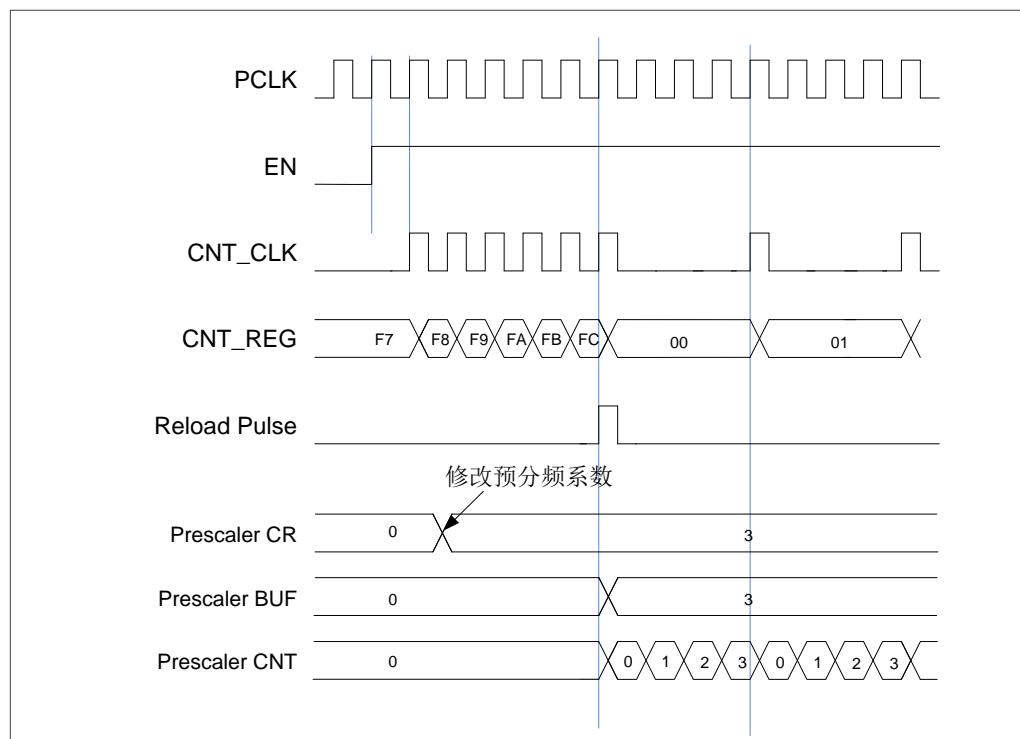


图 9-3. 当预分频器的参数从 1 变到 4 时，计数器的时序图



### 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(TIMER1\_CARL寄存器)，然后重新从0开始计数。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加1(TIMERx\_CREP+1)时，才产生更新事件，否则每次计数器溢出时都会产生更新事件。向上

计数模式中，TIMER1\_CTLR1寄存器中的计数方向控制位应该被设置成0。

可以将TIMER1\_EVG寄存器的UPG位置1来设置更新事件，计数值会被清0，并产生更新事件。

设置TIMER1\_CTLR1寄存器的UPDIS位可以禁止更新事件。

一旦发生更新事件，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作

图 9-4. 计数器时序图，内部时钟分频系数为 1

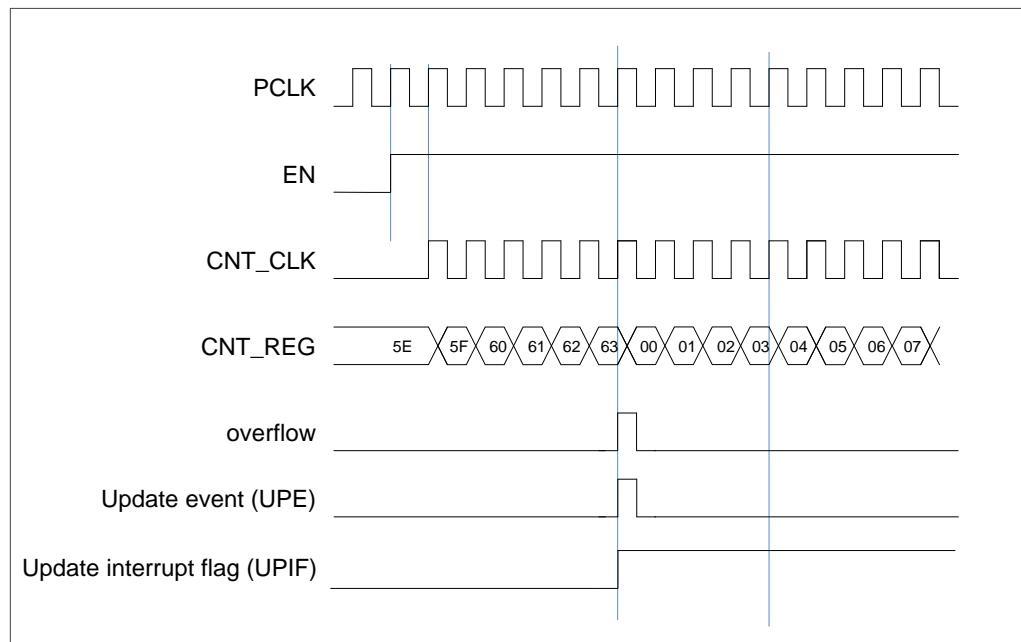


图 9-5. 计数器时序图，内部时钟分频系数为 2

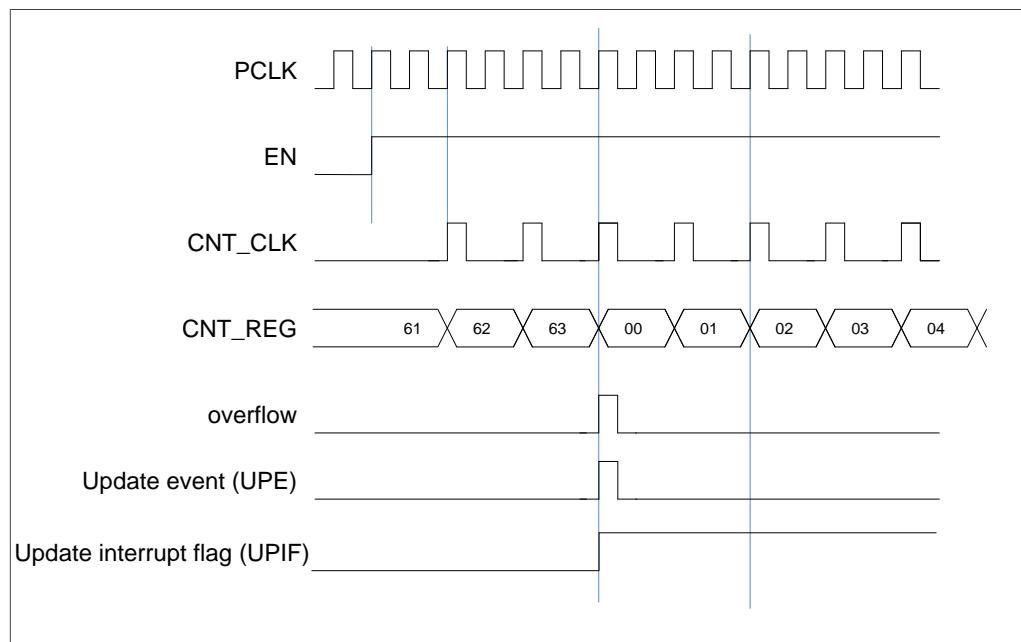


图 9-6. 计数器时序图，内部时钟分频系数为 4

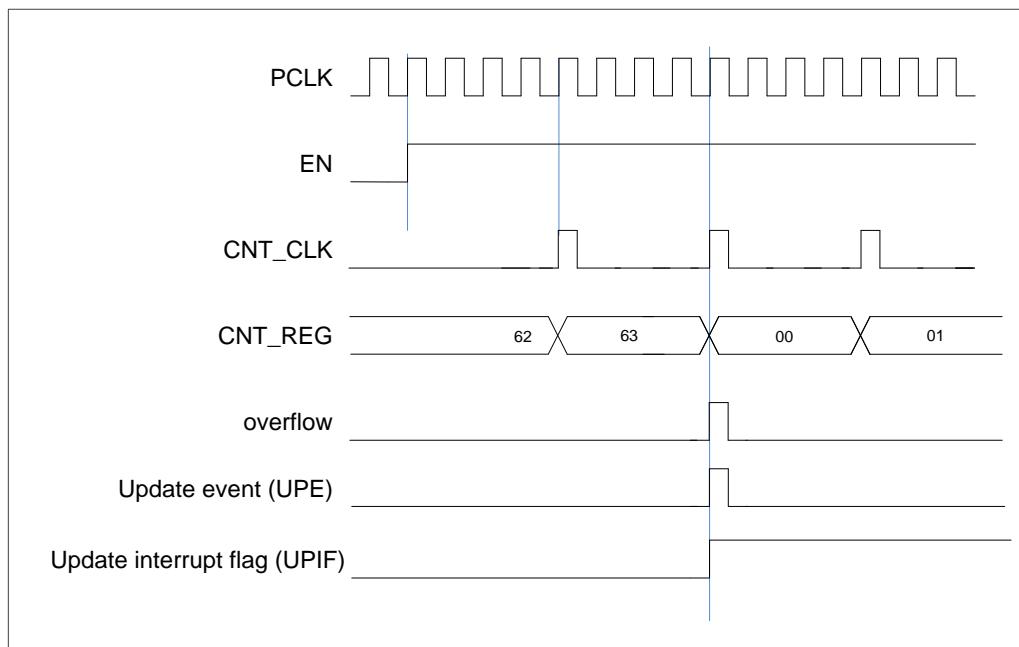


图 9-7. 计数器时序图，内部时钟分频系数为 N

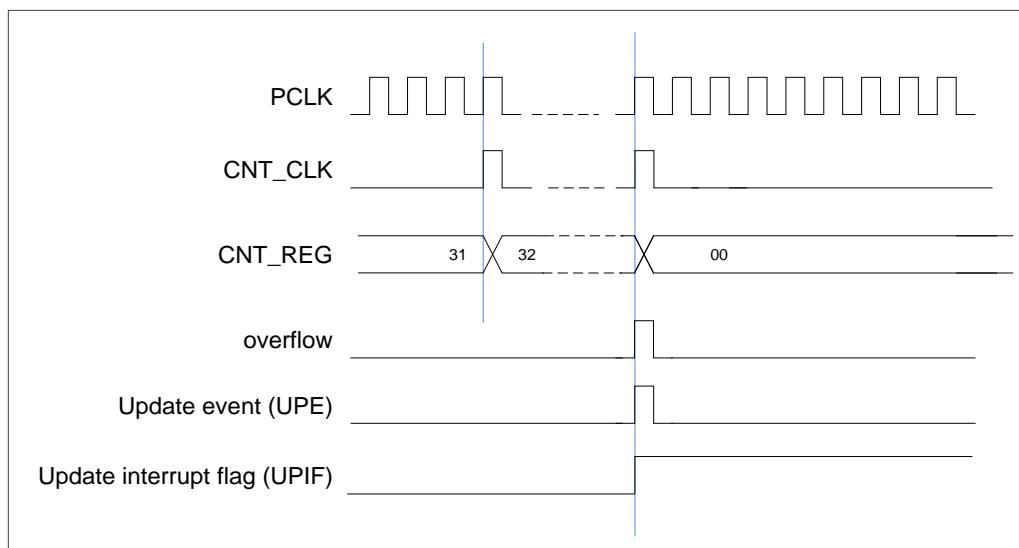


图 9-8. 计数器时序图, 当 ARSE=0 时更新事件

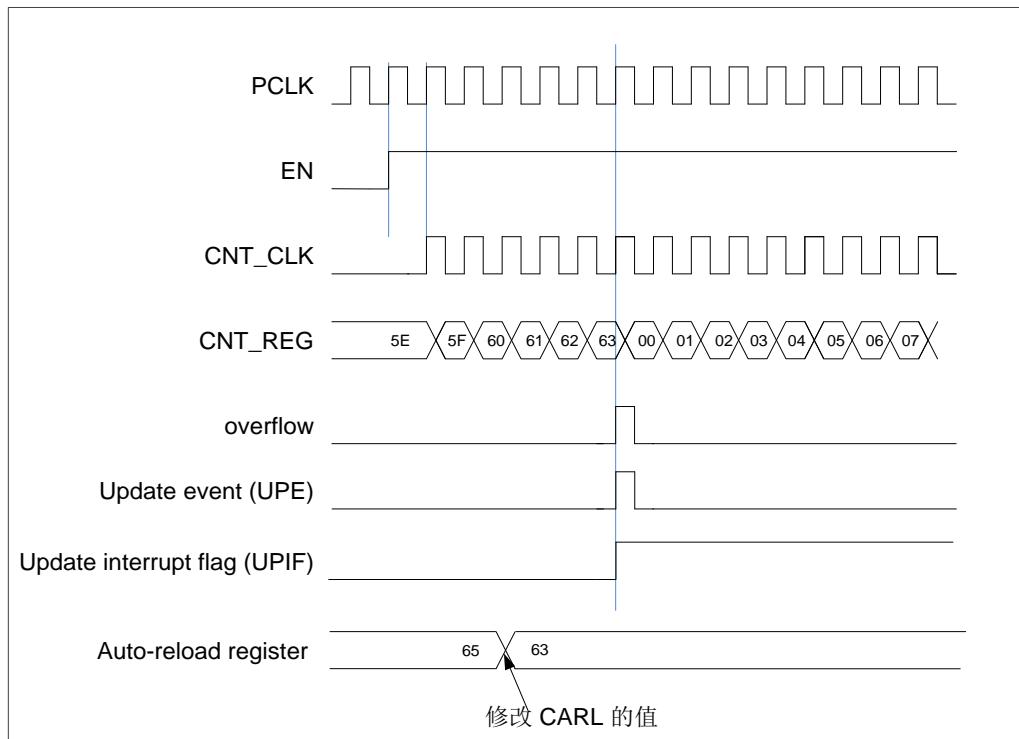
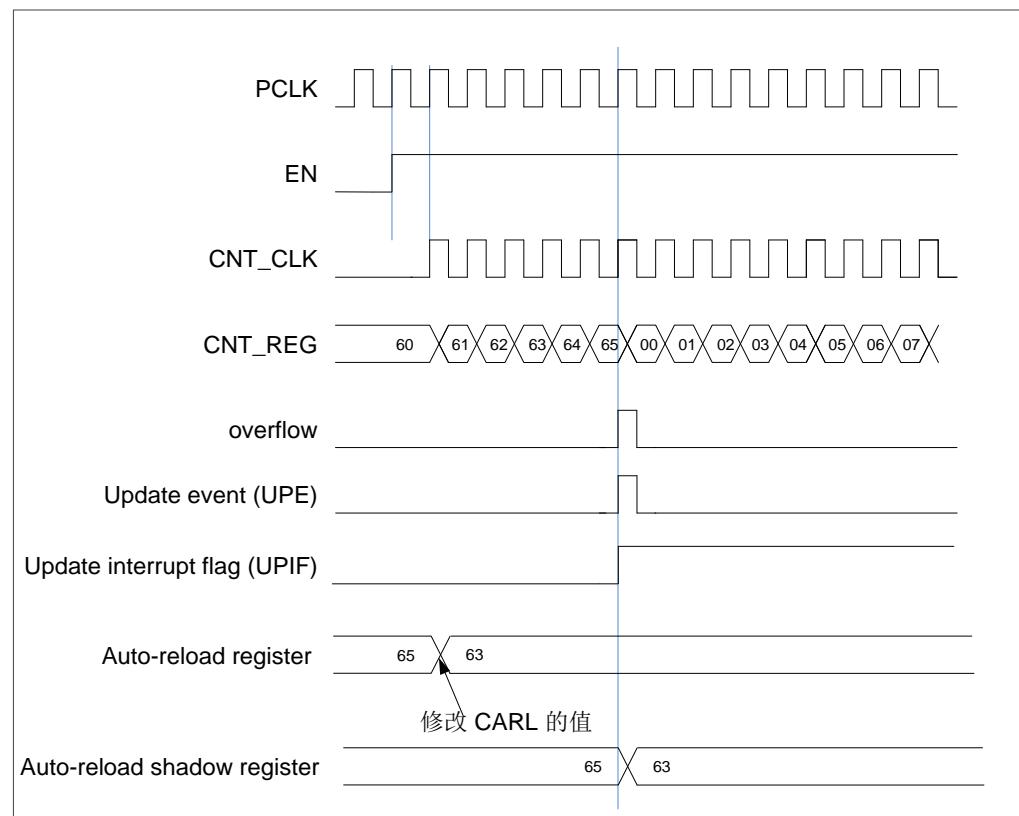


图 9-9. 计数器时序图, 当 ARSE=1 时更新事件



## 向下计数模式

在向下模式中，计数器从自动装入的值(TIMER1\_CARL寄存器的值)开始向下计数到0，然后从自动装入的值重新开始。如果使用了重复计数器，当向下计数达到重复计数寄存器中设定的次数加1 (TIMERx\_CREP+1) 后，才产生更新事件，否则每次计数器下溢时都会产生更新事件。向下计数模式中，TIMER1\_CTLR1寄存器中的计数方向控制位DIR应该被置1。

将TIMER1\_EVG寄存器的UPG位置1可以产生更新事件，更新事件会将计数值初始化为计数器重载值。

将TIMER1\_CTLR1寄存器的UPDIS位置1将禁止更新事件。

一旦产生更新事件，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作。

图 9-10. 计数器时序图，内部时钟分频系数为 1

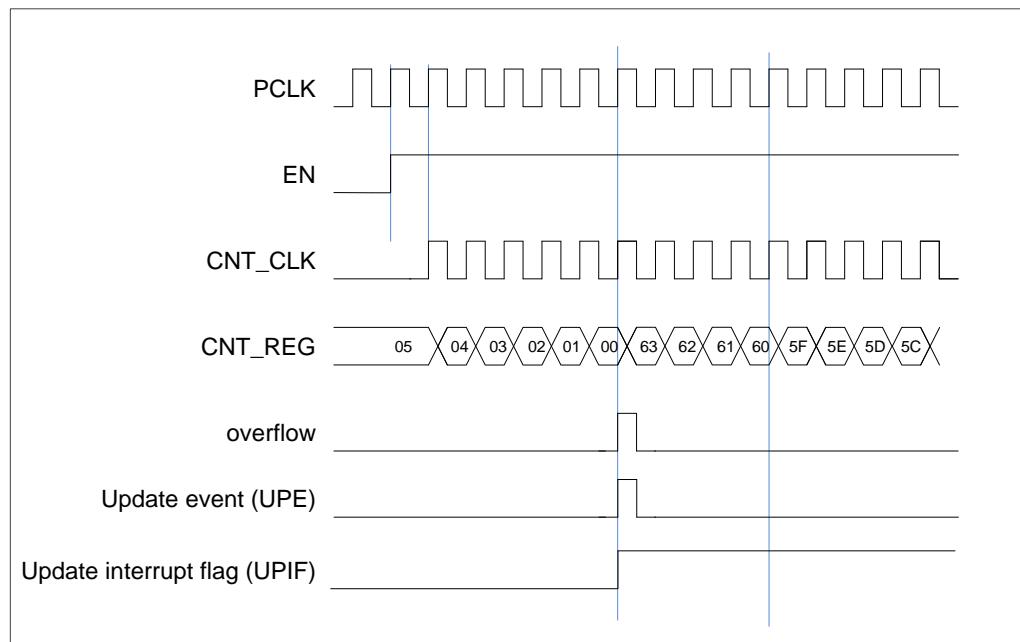


图 9-11. 计数器时序图，内部时钟分频系数为 2

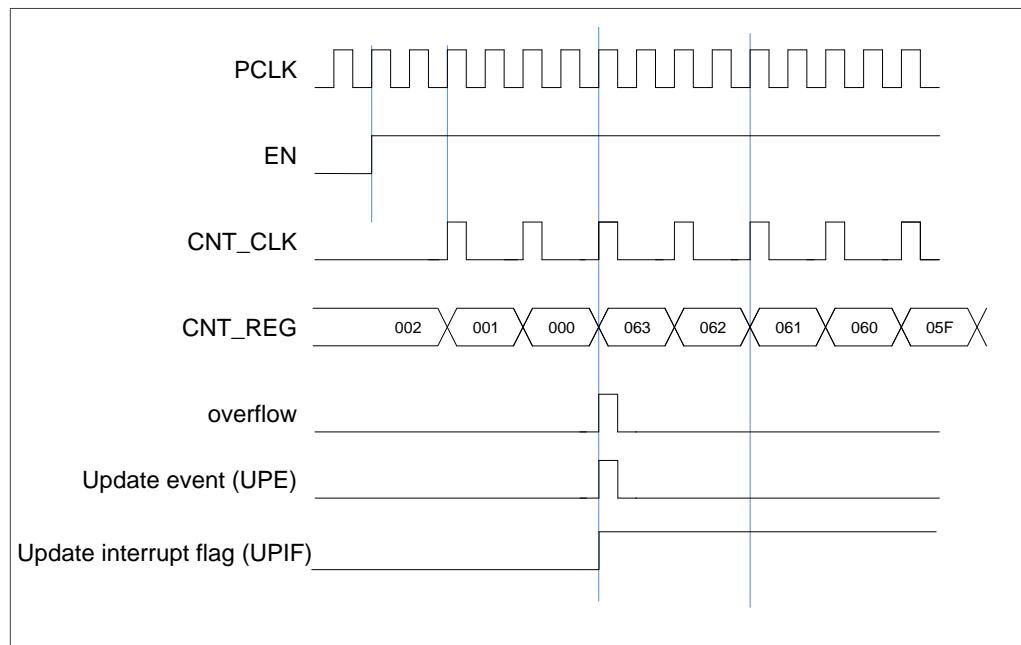


图 9-12. 计数器时序图，内部时钟分频系数为 4

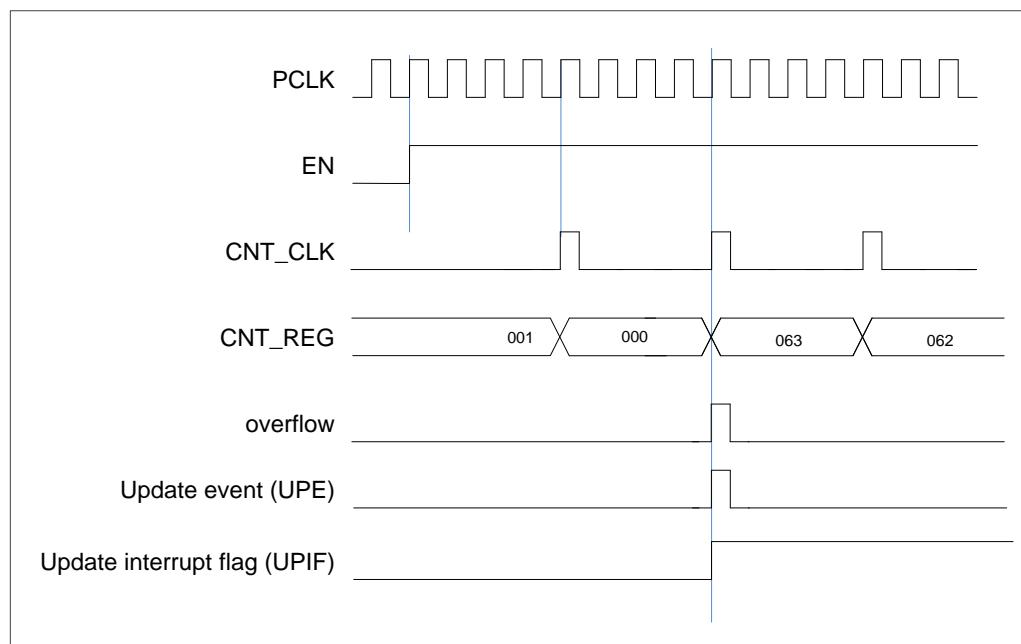


图 9-13. 计数器时序图，内部时钟分频系数为 N

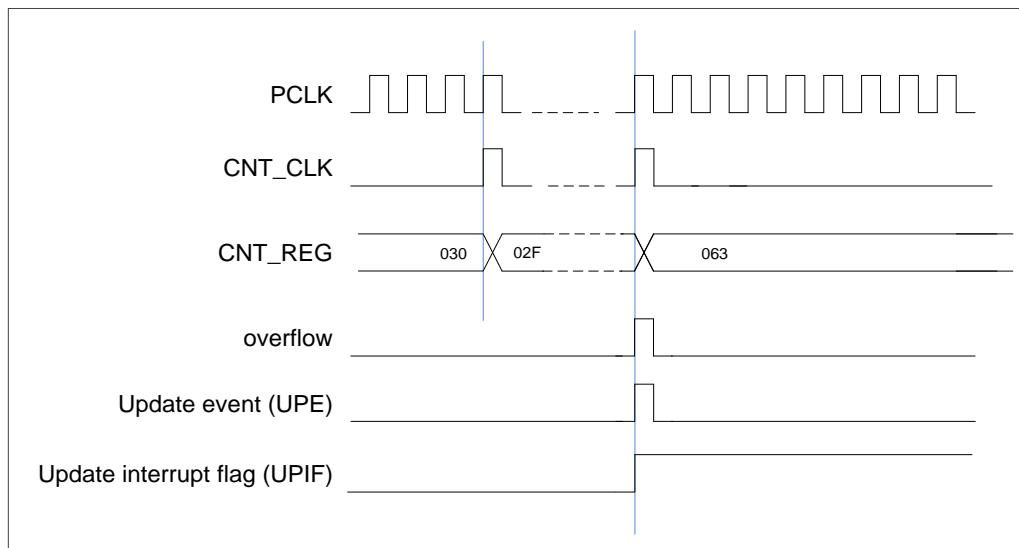
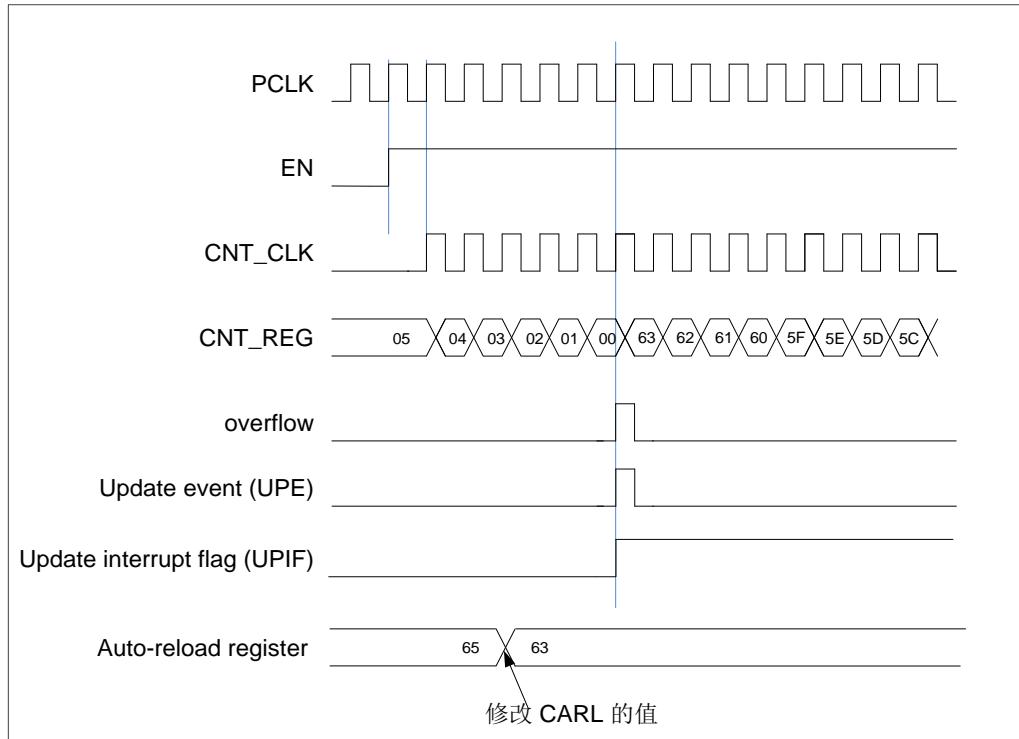


图 9-14. 计数器时序图，不使用计数器时的更新事件



### 中央对齐计数模式

在中央对齐模式，计数器从0开始计数到自动加载的值，然后向下计数到0。向上计数模式中，定时器模块在计数器计数到自动加载值-1产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。TIMER1\_CTLR1寄存器中的计数方向控制位DIR只读，且指示了中间对齐模式下的计数方向。计数方向被硬件自动更新。

将TIMER1\_EVG寄存器的UPG位置1可以初始化计数值为0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

当向下计数(TIMER1\_CTLR1寄存器的CAM为“01”)时发生下溢事件或向上计数时发生上溢事件(TIMER1\_CTLR1寄存器的CAM为“10”), 或者二者同时发生(TIMER1\_CTLR1寄存器的CAM为“11”), TIMER1\_EVG寄存器的UPIF位能被置1。

将TIMER1\_CTLR1寄存器的UPDIS位置1可以禁止更新事件

一旦产生更新事件, 所有的寄存器(重复计数器, 自动重载寄存器, 预分频寄存器)都将被更新。

下图给出一些例子, 当TIMERx\_CARL=0x5时计数器在不同时钟频率下计数器的动作

图 9-15. 计数器时序图, 内部时钟分频系数为 1, TIMERx\_CARL = 0x5

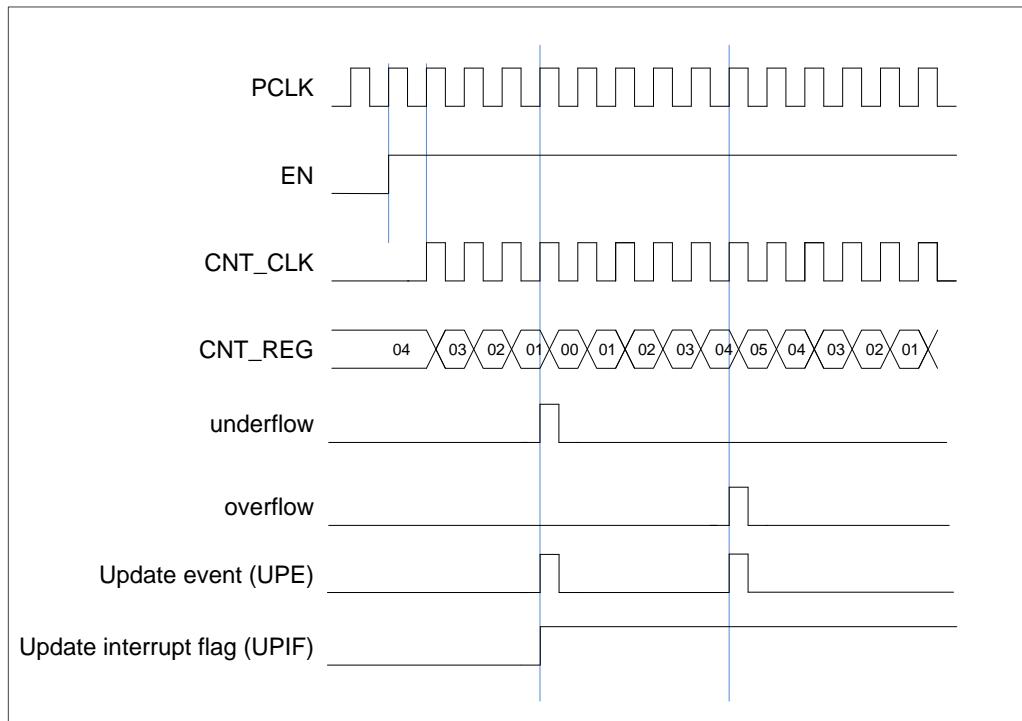


图 9-16. 计数器时序图，内部时钟分频系数为 2

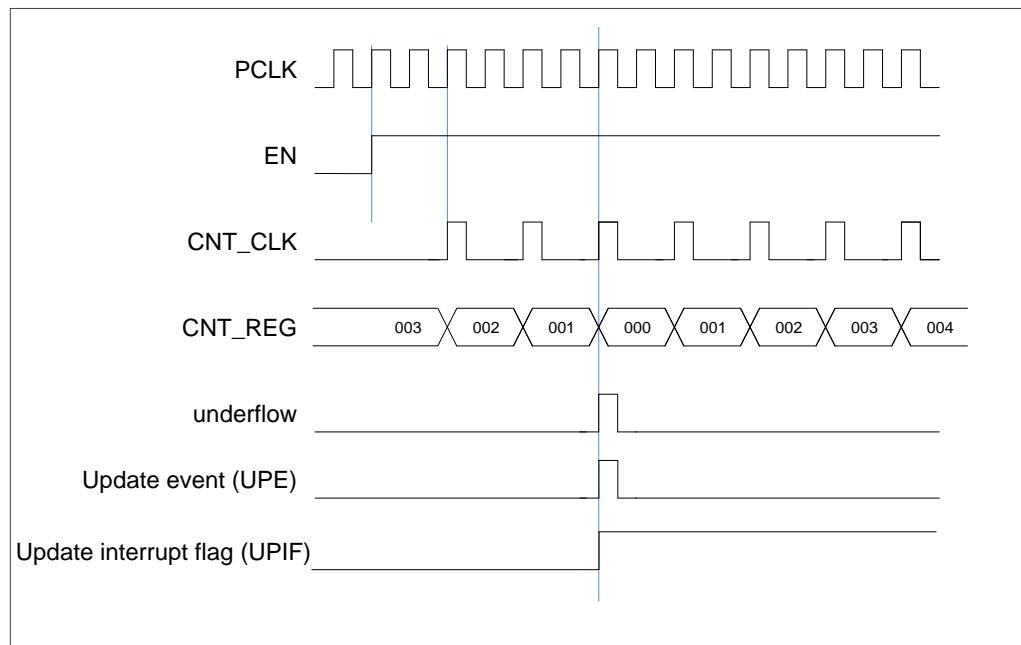


图 9-17. 计数器时序图，内部时钟分频系数为 4, TIMERx\_CARL=0x63

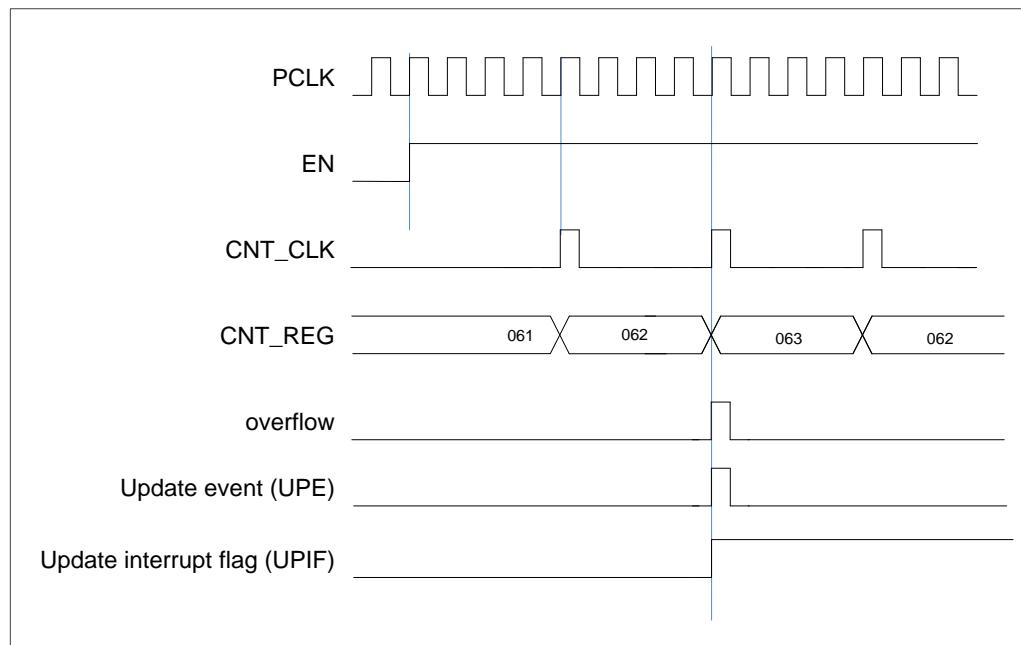


图 9-18. 计数器时序图，内部时钟分频系数为 N

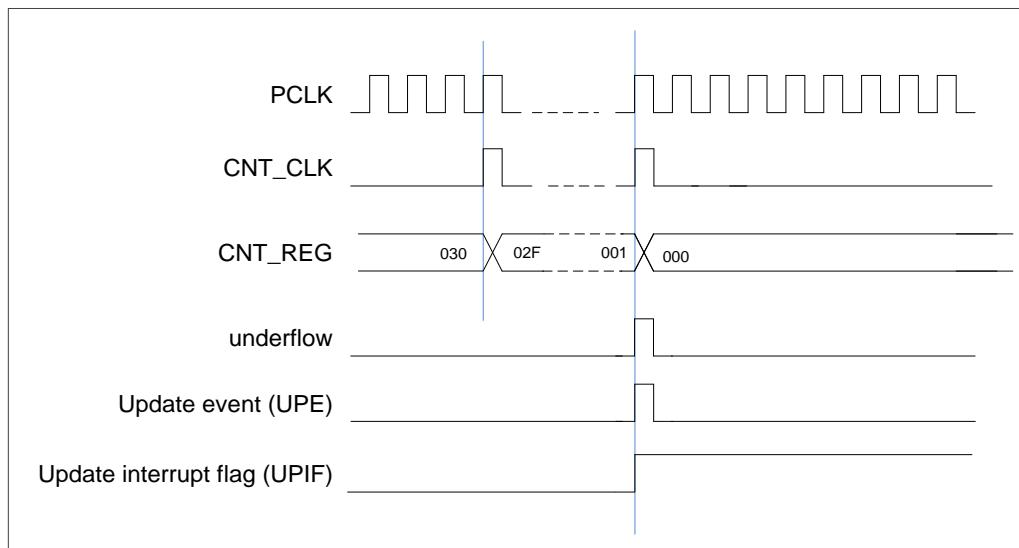


图 9-19. 计数器时序图，当 ARSE=1 (计数器下溢)时的更新事件

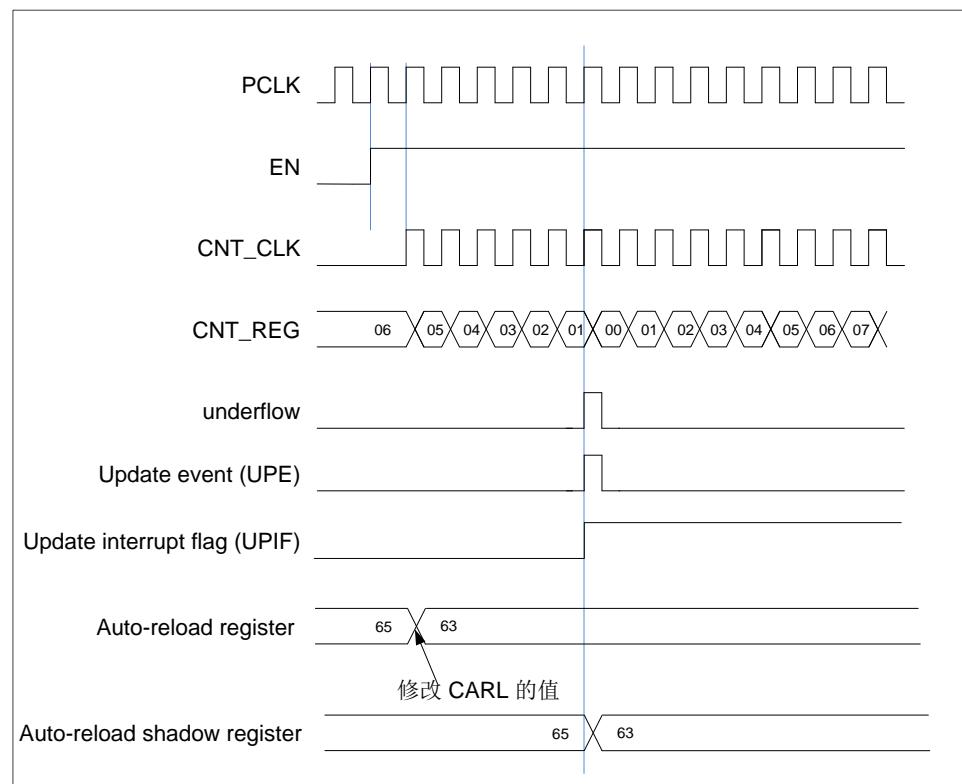
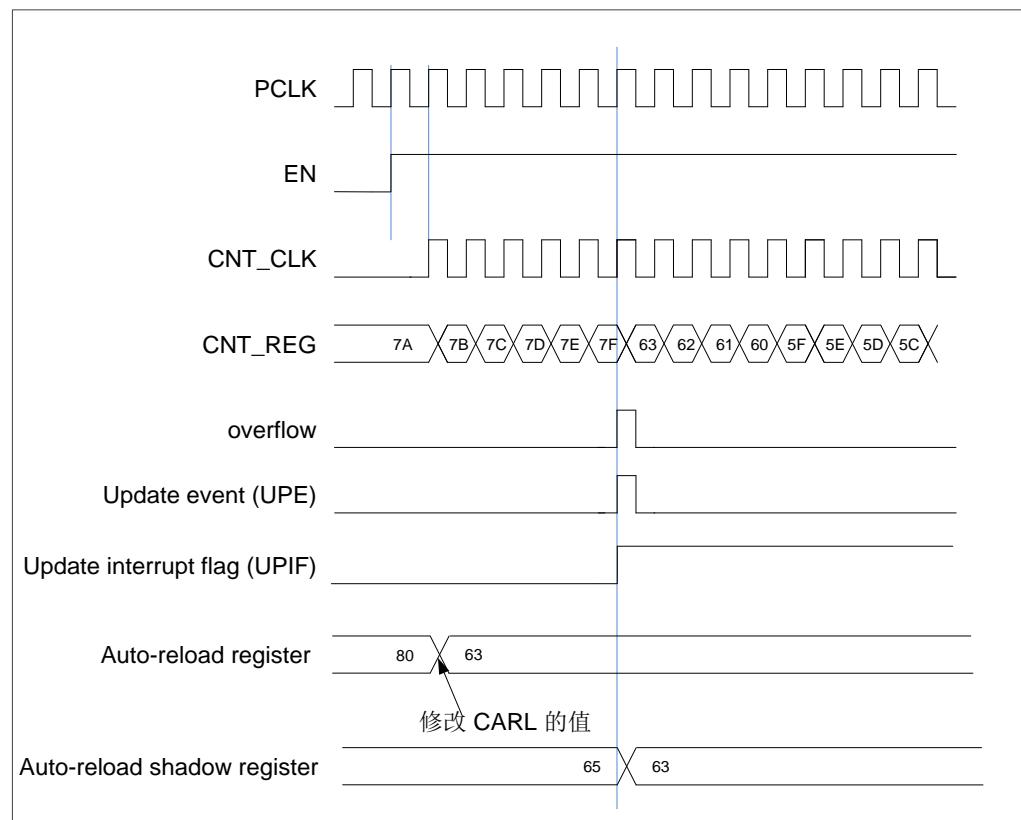


图 9-20. 计数器时序图, 当 ARSE=1 (计数器上溢)时的更新事件

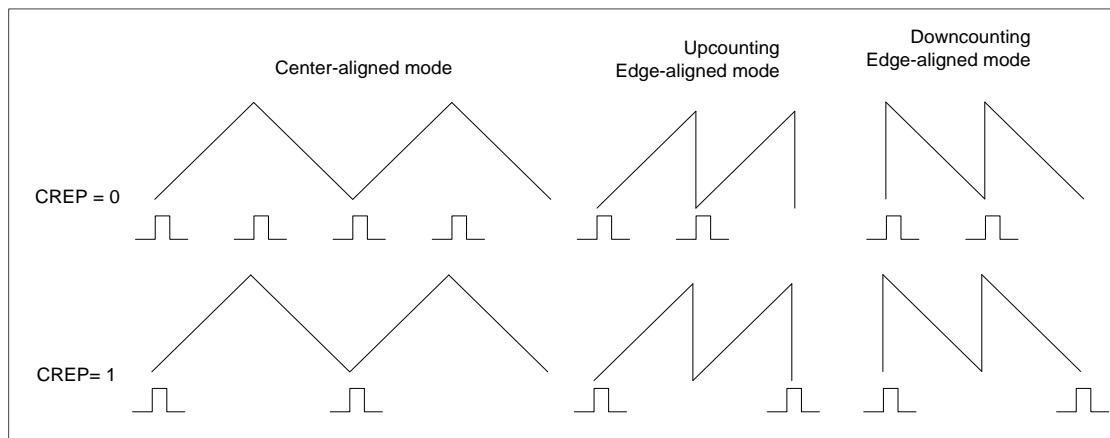


### 重复计数器

重复计数器是用来在N+1个计数周期之后产生更新事件(N为TIMER1\_CREP寄存器的CREP)。向上计数模式下，重复计数器在每次计数器上溢时递减；向下计数模式下，重复计数器在每次计数器下溢时递减；在中央对齐模式下，重复计数器在计数器上溢和下溢时递减。

将TIMER1\_EVG寄存器的UPG位置1可以重载TIMER1\_CREP寄存器中CREP的值并产生一个更新事件。

图 9-21. 不同模式及不同 TIMERx\_CREP 寄存器设置下更新速率的例子



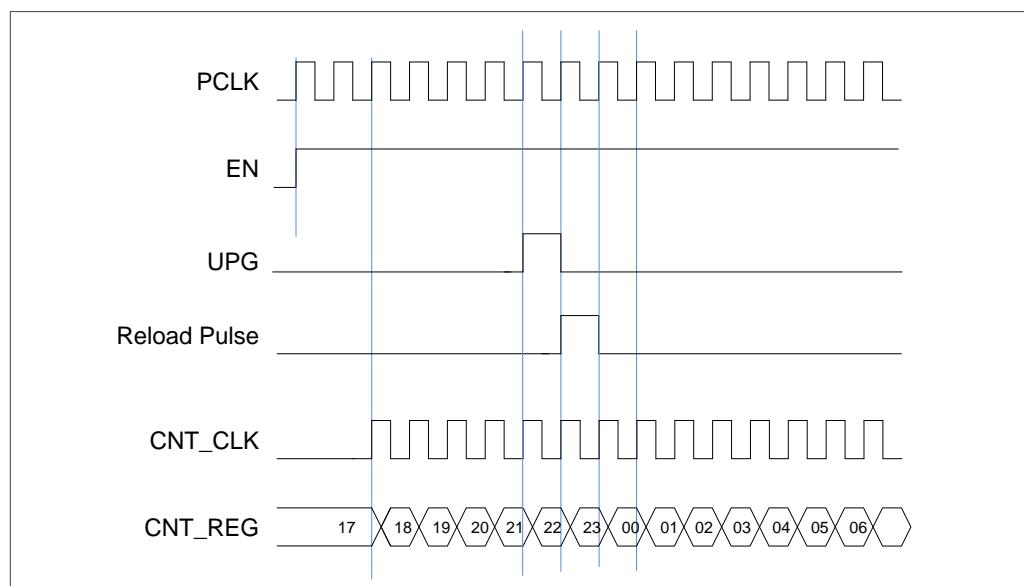
## 时钟选择

决定内部预分频计数器时钟源的定时器模块时钟控制器的描述如下。

### ■ 内部定时器时钟PCLK

如果禁止了从模式，默认用来驱动计数器预分频器的内部时钟源是APB2时钟CK\_APB2。如果使能从模式控制器(将TIMER1\_SMC寄存器的SMC区域设置为包括0x1、0x2、0x3和0x7)，预分频器被其他时钟源(由TIMER1\_SMC寄存器的TRGS区域选择，说明如下)驱动。当从模式选择位SMC被设置为0x4、0x5和0x6，计数器预分频器时钟源由内部时钟PCLK提供。

图 9-22. 内部时钟分频为 1 时正常模式下的控制电路



### ■ 正交译码器

TIMER1\_SMC寄存器的SMC区域应该被设置为0x1, 0x2或0x3来选择正交译码器模式。正交译码器功能使用TIMER1\_CH1和TIMER1\_CH2引脚的两个输入状态产生时钟脉冲驱动计数器预分频器。内部时钟源信号的每次转变时计数方向控制位DIR也被硬件自动改变。输入信号源可以只来源于TIMER1\_CH1引脚，只来源于TIMER1\_CH1引脚或者TIMER1\_CH1和TIMER1\_CH2引脚。

### ■ 内部触发输入 (ITI)

计数器预分频器能在ITI信号的每个上升沿和下降沿计数。此模式可以通过设置TIMER1\_SMC寄存器的SMC区域为0x6来选择，此时计数器将作为一个事件计数器。输入事件ITI能通过设置TRGS区域选择。当ITI信号被选为时钟源，内部边沿监测电路将在每个ITI信号上升沿或下降沿产生一个时钟脉冲来驱动计数器预分频器。

### ■ 外部输入引脚 (TIx)

外部引脚TIMER1\_TIx信号的上升沿和下降沿能驱动计数器预分频器计数。此模式可以通过设置SMC区域为0x7及设置TRGS区域为0x4, 0x5或0x6来选择。注意TIx源于数字滤波器对TIMER1\_TIx采样。

■ 外部输入触发 (ETIF)

外部引脚TIMER1\_ETI信号的每个上升沿或下降沿能驱动计数器预分频器计数。此模式可以通过设置TIMER1\_SMC寄存器的ECM2E位为1来选择。其他选择ETIF信号作为时钟源的方式为设置SMC区域为0x6和设置TRGS区域为0X7。注意DTIF信号源于数字滤波器对TIMER1\_ETI引脚信号的采样。如果选择ETIF信号为时钟源，触发控制器包括边沿监测电路将在每个DTIF信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

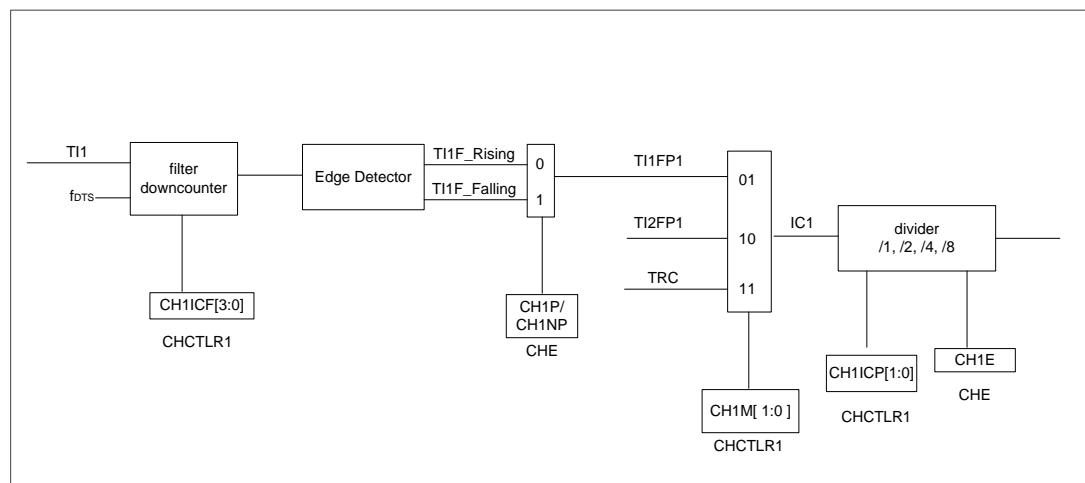
### 捕获/比较通道

TIMER1拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获级

输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。通道1的输入信号(TI1)可以选择来自TIMER1\_CH1信号或者TIMER1\_CH1，TIMER1\_CH2和TIMER1\_CH3信号异或后的信号。通道输入信号(TIx)被一个数字滤波器采样来产生一个滤波输入信号TIxF。然后通道极性和边沿监测模块产生一个TIxFP1或TIxFP2信号，来实现输入捕捉功能。通道输入预分频寄存器(CHx\_ICP)可以设置有效输入事件数目。

图 9-23. 捕获/比较通道(例如：通道 1 输入级)



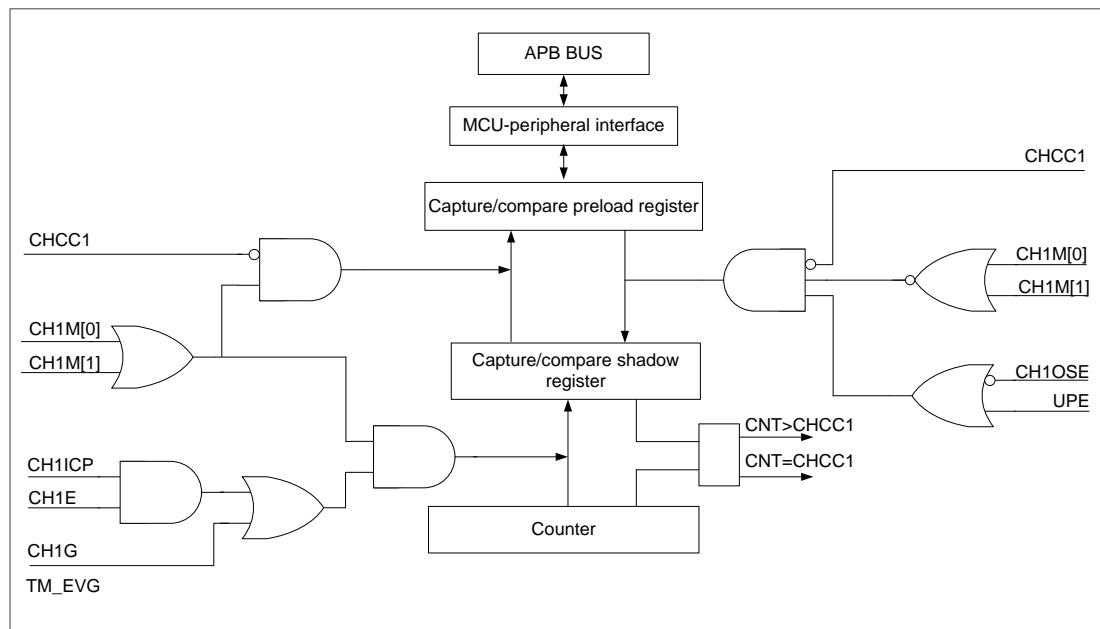
■ 通道控制器

定时器有4个独立的通道用来捕获输入或输出匹配

用于输入捕获模式下，当捕获事件发生时，TIMER1\_CHCCx 影子寄存器首先捕获计数值然后传送到TIMER1\_CHCCx 预装载寄存器中。

用于输出匹配比较模式下，TIMER1\_CHCCx预装载寄存器的值被复制到相应的影子寄存器中，然后计数值和影子寄存器的值比较。

图 9-24. 捕获/比较通道 1 主电路



■ 输出级

TIMER1 有四个用于匹配比较、单脉冲输出和 PWM 输出功能的通道。

图 9-25. 捕获/比较通道输出级 (通道 1 到 3)

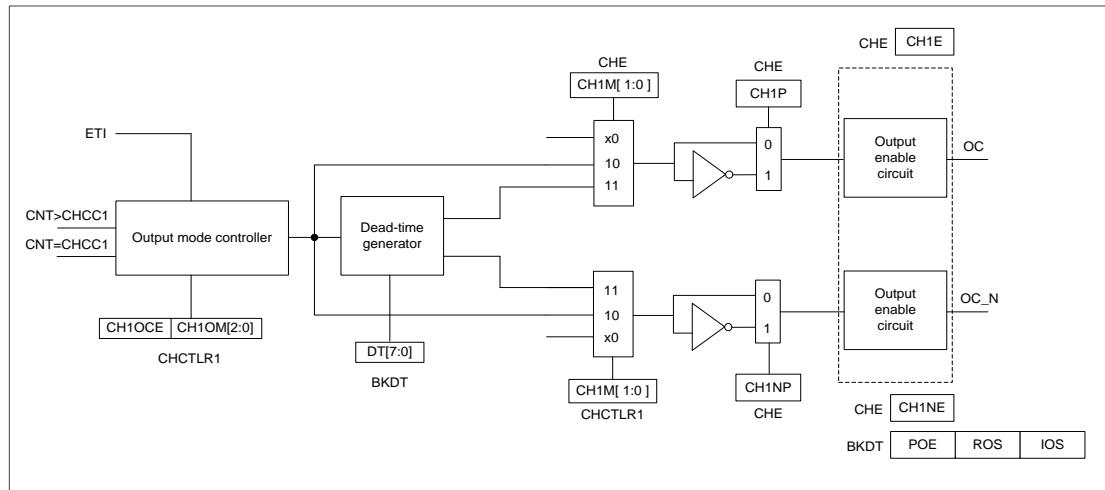
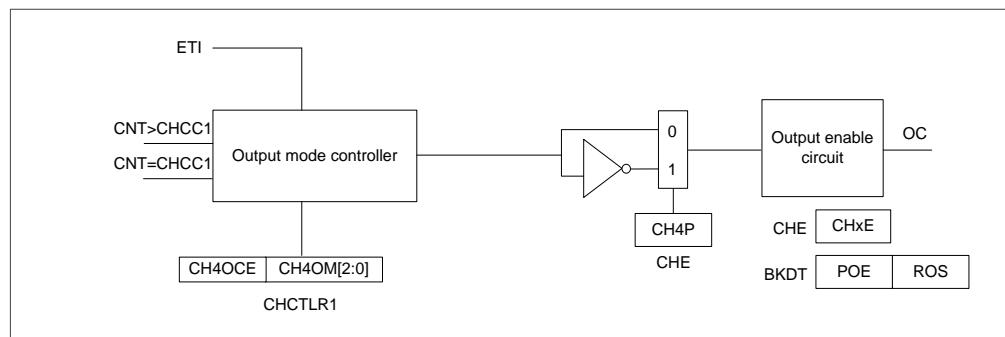


图 9-26. 捕获/比较通道输出级 (通道 4)



## 输入捕获模式

通道用于输入捕获时，当一个有效输入信号的边沿跳变时，计数值被捕获到捕获/比较寄存器(TIMER1\_CHCCx)中。一旦捕获事件发生，TIMER1\_STR寄存器的CHxIF标志位被置1。如果CHxIF位已经被置1，还没来得及被软件清除，此通道上又有另一个捕获事件发生，相应通道捕获溢出标志位CHxOF将被置1。一旦捕获事件发生，CHxDE位决定了是否产生DMA请求，CHxIE位决定了是否产生中断。使用输入捕获模式需按以下步骤配置：

1. 配置通道控制寄存器(TIMER1\_CHCTRLRx)的CCxM位域，选择输入信号
2. 配置通道控制寄存器(TIMER1\_CHCTRLRx)的CHxICF位域，为输入信号添加输入滤波器
3. 配置通道使能寄存器(TIMER1\_CHE)的CHxP/CHxNP位，选择输入捕获的边沿（上升沿或下降沿）
4. 如果输入信号需要分频，配置通道控制寄存器(TIMER1\_CHCTRLRx)的CHxICP位域
5. 如果需要中断，将DMA中断使能寄存器(TIMER1\_DIE)的CHxIE位置位
6. 如果需要DMA请求，将DMA中断使能寄存器(TIMER1\_DIE)的CHxDE位置位
7. 将通道使能寄存器(TIMER1\_CHE)的CCxE位置位，使能通道

按上述配置完寄存器后，该通道上的对应边沿信号将产生输入捕获事件。计数值被捕获到通道捕获/比较寄存器(TIMER1\_CHCCx)中。如果CHxIE位被置位，将会产生中断。如果CHxDE位被置位，将会产生DMA请求。

输入捕获模式也可以用于TIMER1\_CHx(TIx)引脚的信号脉宽测量。例如，PWM信号连接到TI1输入。配置TIMER1\_CHCTRLRx寄存器的CH1M位域为01，将通道1输入捕获信号连接到TI1，并设置上升沿捕获。配置TIMER1\_CHCTRLRx寄存器的CH2M位域位为10，选择通道2捕获信号连接到到TI1，并设置下降沿捕获。计数器配置为通道1上升沿触发后重新计数模式。此时TIMER1\_CHCC1可以测量PWM信号的周期，TIMER1\_CHCC2可以测量PWM信号的占空比。

## 输出比较模式/PWM 模式

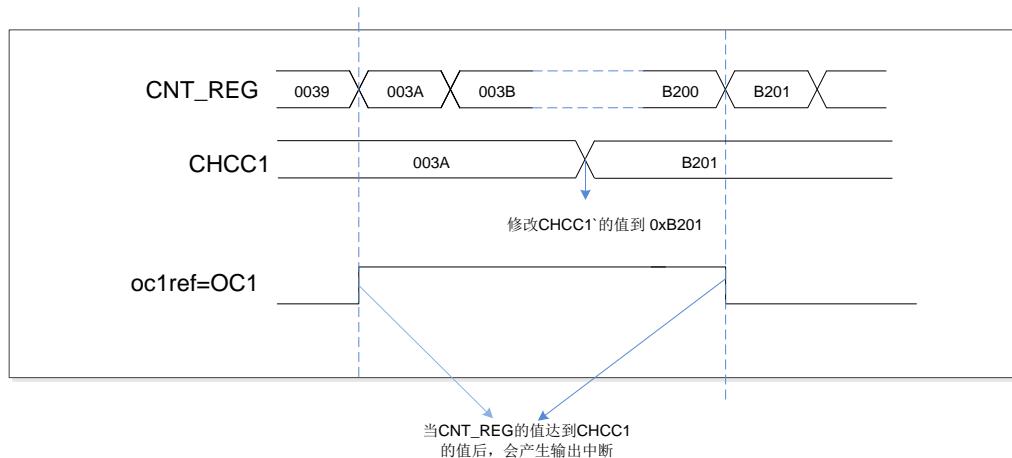
当配置通道控制寄存器(TIMER1\_CHCTRLRx)的CCxM位域为00，使通道工作于输出模式时，通道输出的波形由TIMER1\_CHCTRLRx的CHxOM位域和计数器与TIMER1\_CHCCx寄存器的值比较的结果决定。当比较的值相等时状态寄存器(TIMER1\_STR)的CHxIF位将置1。根据CHxDE或CHxIE位的值，一次DMA请求或中断将被产生。

在输出模式，配置CHxOM位域为101，通道将持续输出有效电平。配置CHxOM位域为100通道将持续输出无效电平。

在输出模式，配置CHxOM位域为001，当计数器与TIMER1\_CHCCx寄存器的值相等时，通道输出有效电平。配置CHxOM位域为010，当计数器与TIMER1\_CHCCx寄存器的值相等时，通道输出无效电平。

在输出比较翻转模式（配置CHxOM位域为011），当计数器与TIMER1\_CHCCx寄存器的值相等时，通道输出电平将翻转。

图 9-27. 输出比较模式，在 OC1 上切换



在PWM输出模式（配置CHxOM位域为110（PWM模式1）或111（PWM模式2））。根据TIMER1\_CAR1寄存器和TIMER1\_CHCCx寄存器的值，通道输出PWM波形。

图 9-28. 在 OC1 上输出比较 PWM 模式 1，向上计数模式

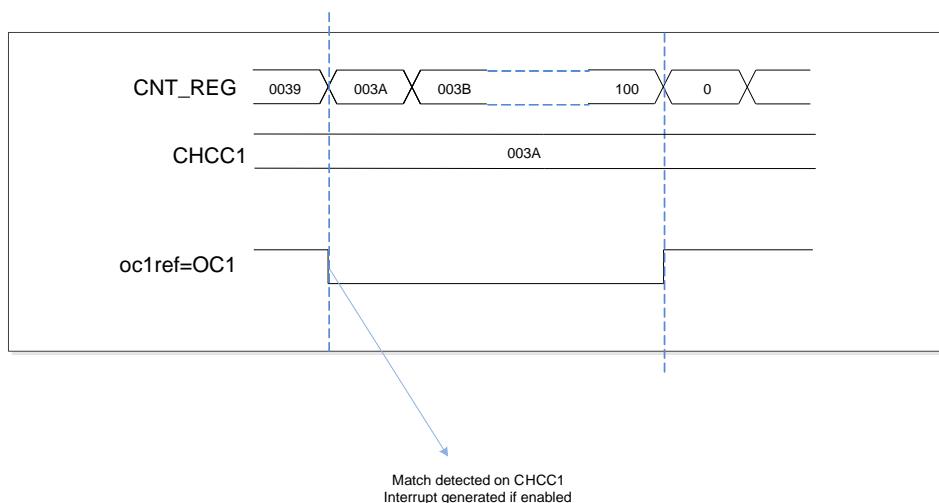
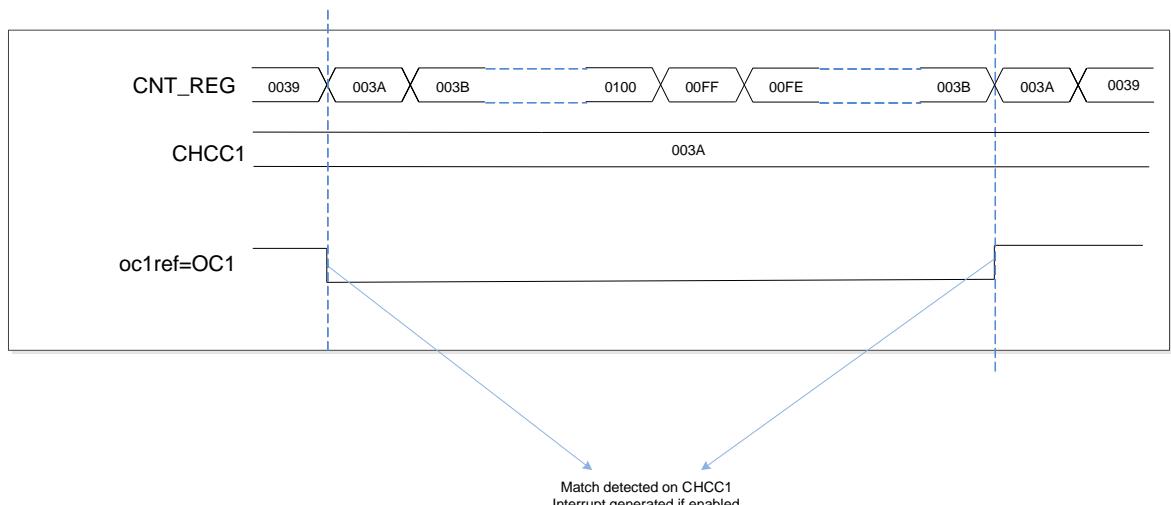


图 9-29. 在 OC1 上输出比较 PWM 模式 1，中央对齐计数模式



### 通道输出参考信号

当TIMER1用于输出匹配比较模式下，设置CHxOM位可以定义OCxREF信号(通道x输出参考信号)类型。OCxREF信号有一些输出功能类型，包括，设置CHxOM=0x00可以保持原始电平，设置CHxOM=0x01可以将OCxREF信号设置为高电平，设置CHxOM=0x02可以将OCxREF信号设置为低电平，在计数值和TIMER1\_CHCCx寄存器的值匹配时，设置CHxOM=0x03可以翻转输出信号。

PWM模式1和PWM模式2输出是另一种类型的OCxREF输出，具体设置是通过设置CHxOM=0x06或0x07来完成的。在这些模式中，OCxREF信号电平的改变取决于计数方向，以及TIMER1\_CHCCx寄存器值和计数值之间的关系。详情参考相应位定义。

设置CHxOM=0x04或0x05可以实现OCxREF信号的强制输出功能。输出比较信号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。

当ETIF信号源于外部TIMER1\_ETI引脚且被设置为高电平(设置TIMER1\_CHCTRL1寄存器的CHxOCE位为1)时，OCxREF信号被强制变为低电平。直到下一次更新事件发生，OCxREF信号才会回到它的有效状态。

### 互补输出及死区时间

TIMER1可以输出两路互补信号。互补信号OCx和OCxN由一些控制位的组合实现：寄存器TIMER1\_CHE的CHxE位和CHxNE位，寄存器TIMER1\_BKDT和TIMER1\_CTLR2的POE，ISOx，ISOxN，IOS和ROS位。输出极性由TIMER1\_CHE寄存器的CHxP位和CHxNP位决定。

如果CHxE，CHxNE和POE位都为1，会插入死区时间。OCx输出的上升沿和OCxREF的上升沿相比有一个延迟；OCxN输出的上升沿和OCxREF的下降沿相比有一个延迟。延迟的值是一个8位的死区时间计数器，由TIMER1\_BKDT寄存器的DT区域决定。如果延迟值大于有效输出(OCx或OCxN)宽度，将不会产生相应信号脉冲。

图 9-30. 带有死区时间的互补输出

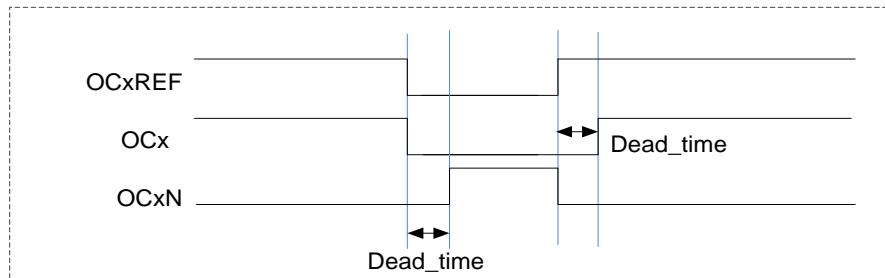


图 9-31. 延迟大于负脉冲的死区时间波形

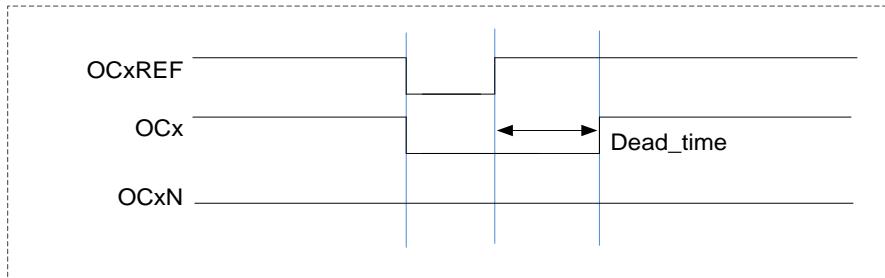
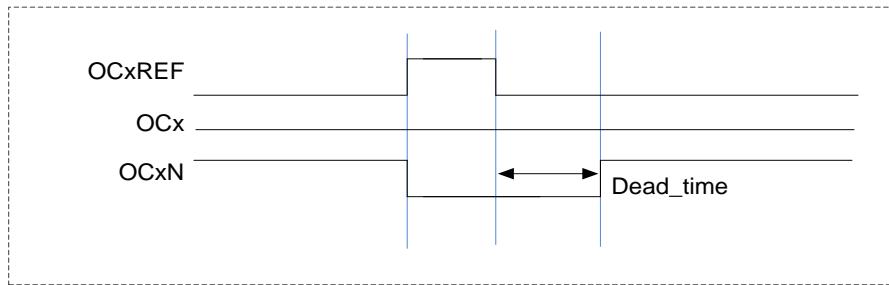


图 9-32. 延迟大于正脉冲的死区时间波形



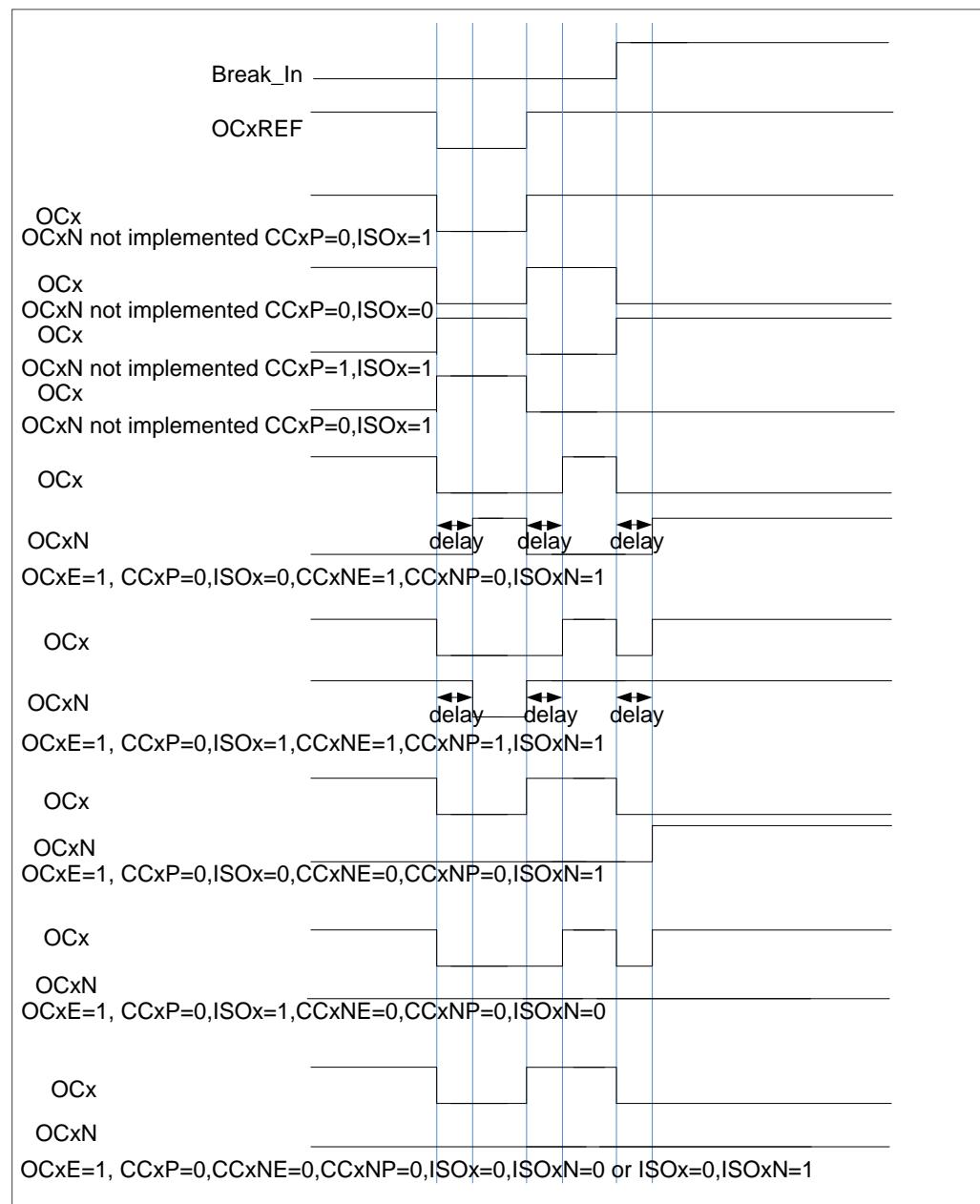
### 中止功能

使用中止功能时，输出OCx和OCxN信号电平被以下位控制，TIMER1\_BKDT寄存器的POE，IOS和ROS位，TIMER1\_CTLR2寄存器的ISOx 和ISOxN位。中止事件发生时，OCx和OCxN信号输出不能在同一时间同时处于有效电平上。中止源既可以是中止输入引脚又可以是一个HSE时钟失败事件。时钟失败事件由RCC中的时钟监视器(CKM)产生。将TIMER1\_BKDT寄存器的BRKE位置1可以使能中止功能。TIMER1\_BKDT寄存器的BRKP位决定了中止输入极性。

发生中止时，POE位被异步清除，一旦POE位为0，配置TIMER1\_CTLR2寄存器的ISOx位驱动输出OCx和OCxN的电平。如果IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，TIMER1\_STR寄存器的BKIF位被置1。如果BKIE=1，中断产生。如果BKDE=1，发送一个DMA请求。

图 9-33. 中止功能相应的输出表现



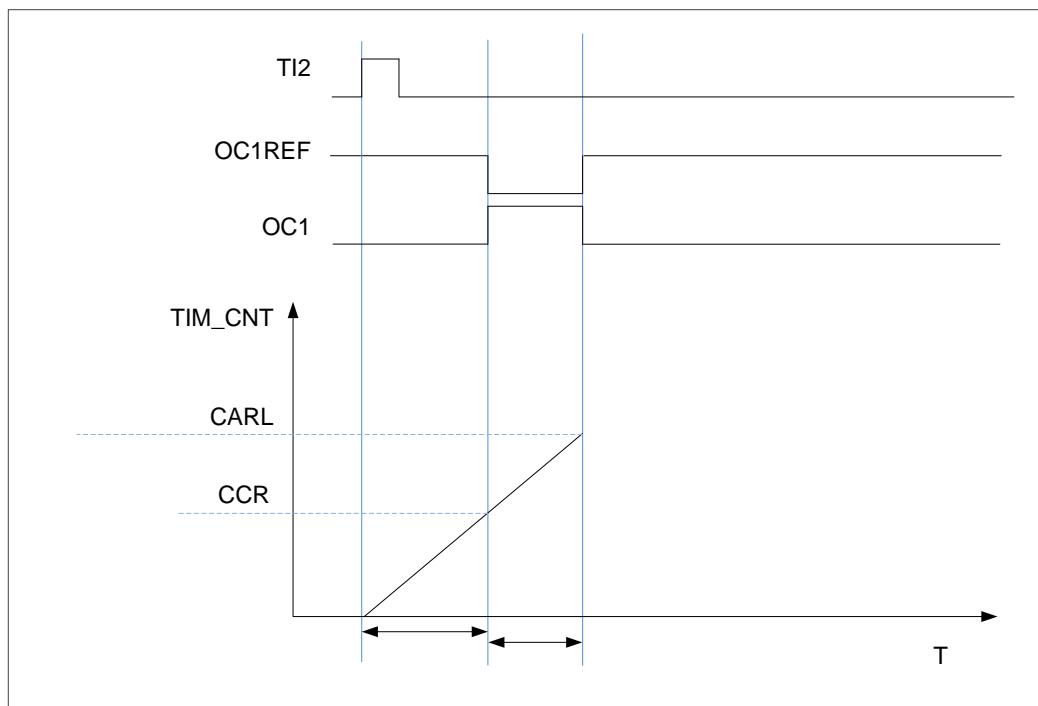
### 单脉冲模式

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMER1\_CTRLR1寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

单脉冲模式下，将CEN位置1产生的触发有效边沿会使能计数器。然而，执行计数值和TIMER1\_CHCCx寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMER1\_CHCTRLR1寄存器的CHxOEF位置1。单脉冲模式下，触发上升沿产生之后，OCxREF信号将被立即强制转换为另一种状态，因为不考虑比较结果的比较匹配事件发生。只有输出通道配置为PWM1或PWM2输出运行模式下时CHxOEF位才可用，触发源来源于触发

信号。

图 9-34. 单脉冲模式



### 正交译码器

正交译码器功能使用两个源于TIMER1\_CH1和TIMER1\_CH2引脚的正交输入TI1和TI2各自相互作用产生计数值。在每个输入源改变期间，DIR位被硬件自动改变。输入源可以是只有TI1，只有TI2或TI1和TI2。选择模式可以设置SMC=0x01, 0x02或0x03来完成。改变计数方向的机制如下图所示。正交译码器可以用作一个带有方向选择的外部时钟，意味着计数器在内部从0到重载值持续计数。因此，用户必须在计数器开始计数前配置TIMER1\_CARL寄存器。

表 9-1. 计数方向对编码器信号

计数模式	电平	TI1FP1		TI2FP2	
		上升	下降	上升	下降
只有 TI1	TI2FP=1	向下	向上	-	-
	TI2FP=0	向上	向下	-	-
只有 TI2	TI1FP=1	-	-	向上	向下
	TI1FP=0	-	-	向下	向上
TI1 和 TI2	TI2FP=1	向下	向上	X	X
	TI2FP=0	向上	向下	X	X
	TI1FP=1	X	X	向上	向下
	TI1FP=0	X	X	向下	向上

注意：“-”意思是“无计数”，“X”意思是不可能。

图 9-35. 编码器接口模式下计数器运行实例

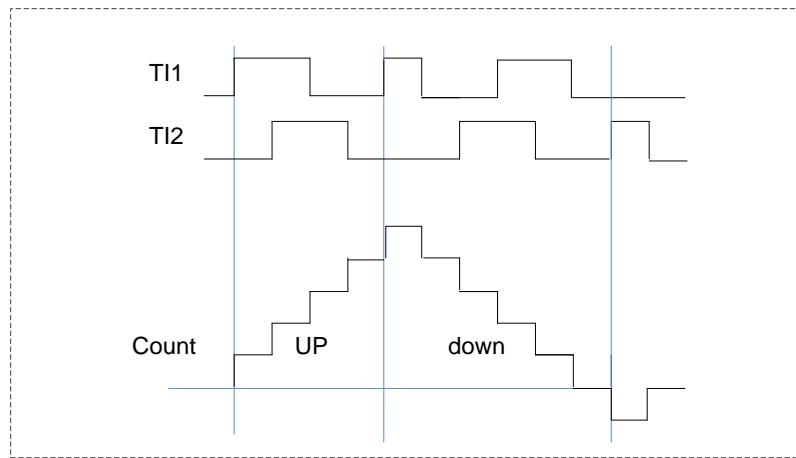
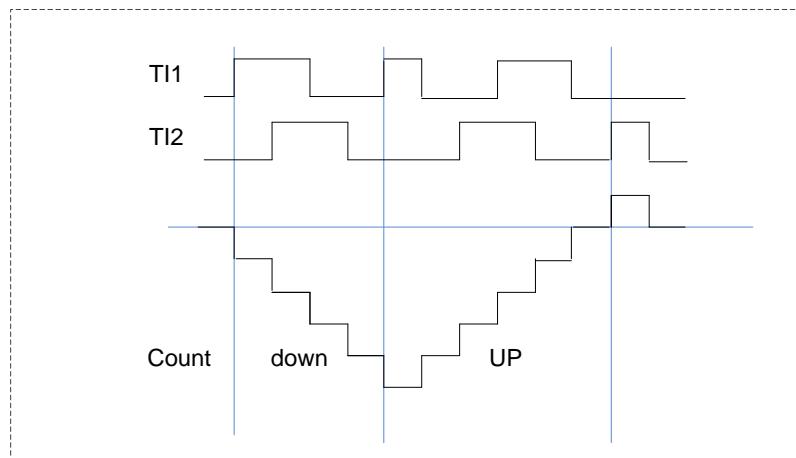


图 9-36. 带有 TI1FP1 极性转换的编码器接口模式下的例子



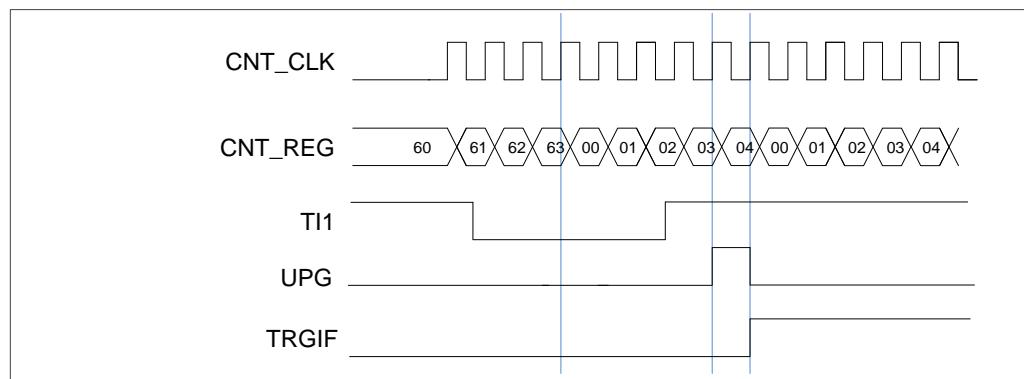
## 从控制器

TIMER1能够在多种模式下和一个外部的触发同步，包括复位模式，暂停模式和触发模式。模式选择通过设置TIMER1\_SMC寄存器的SMC区域来完成。这些模式的触发输入可以通过TIMERx\_SMC寄存器的TRGS区域来选择，下面以TI1信号为例。从控制器运行模式参见附加章节。

### ■ 复位模式

在TI1信号的上升沿来临时，计数器和它的预分频器能够重新被初始化，此时更新事件生成寄存器位UPG将自动被硬件置位，触发事件标志位被置1，然后计数器和它的预分频器被初始化。尽管UPG位被硬件置1，更新事件可能并不会发生，它取决于更新事件禁止控制位UPDIS是否为1。如果UPDIS被置1，将不产生更新事件，然而当TI1上升沿来临时计数器和它的预分频器仍将被初始化。如果TIMER1\_CTLR1寄存器的UPDIS位为0，更新事件在TI1上升沿来临后发生，所有的预加载寄存器将被更新，TI1的上升沿和CNT\_REG被清零之间的延时取决于TI1的同步电路。

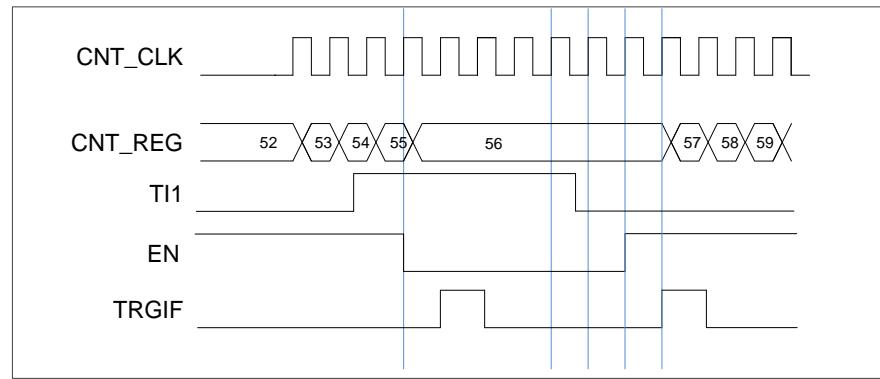
图 9-37. 复位模式的控制电路



### ■ 暂停模式

暂停模式下，选择的TI1输入信号电平用来控制计数器开启/停止运行。当选择的TI1信号为高电平时计数器开始计数，当TI1信号变为低电平时停止计数。此处计数器将保持它的当前值不变且不再被重启。

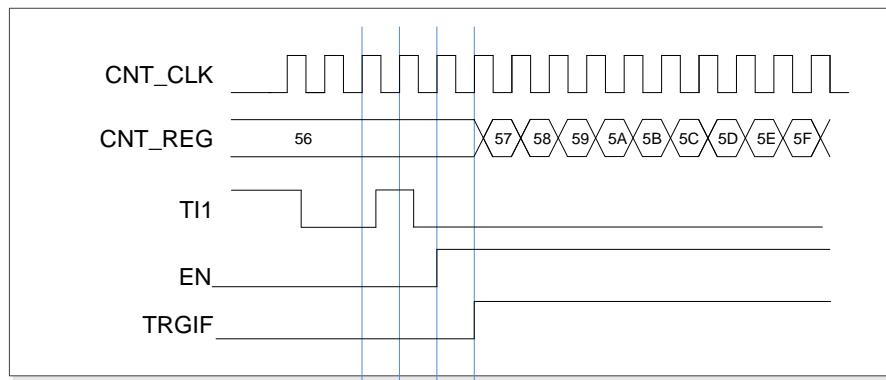
**图 9-38. 暂停模式下的控制电路**



#### ■ 触发模式

计数器被禁止计数之后，当TI1上升沿来临时计数器可以重新开始计数。当一个TI1上升沿来临，计数器从当前值开始计数。注意TI1信号仅用来使能计数器计数，不能控制计数器停止计数。

**图 9-39. 触发模式下的控制电路**

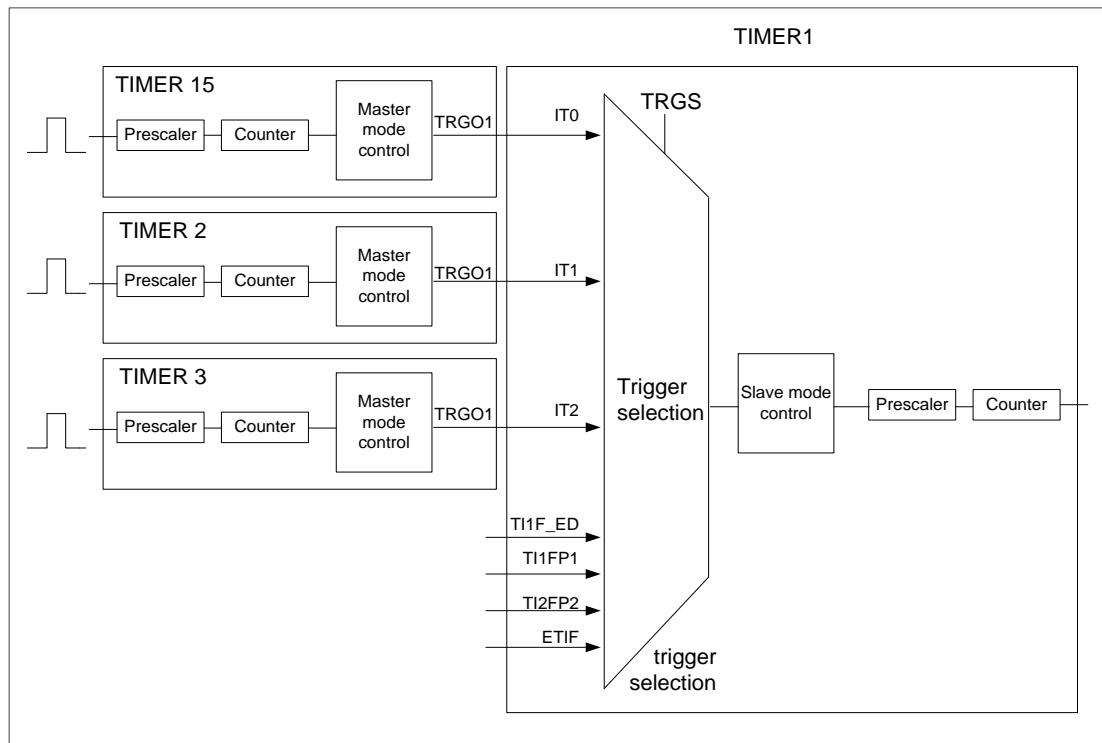


#### 定时器互连

定时器互相之间可以级联或者同步。可以通过配置一个定时器工作在主模式另一个定时器工作在从模式来实现。下图显示了一些主从模式触发选择的例子。

下图显示了当定时器 1 配置为从模式时的触发选择

图 9-40. 定时器 1 主/从模式的例子



其他互连的例子:

■ 定时器 2 作为定时器 1 的预分频器

参考下图连接配置定时器2为定时器1的预分频器，按以下步骤去做：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出 (配置TIMER2\_CTLR2寄存器的MMC=010)。定时器2在每次计数器溢出时，输出一个周期信号。
2. 配置定时器2周期 (TIMER2\_CARL寄存器)。
3. 选择定时器1输入触发源为定时器2 (配置TIMER1\_SMC寄存器的TRGS=001)。
4. 配置定时器1在外部时钟模式1 (配置TIMER1\_SMC寄存器的SMC=111)。
5. 写1到CEN位启动定时器1 (TIMER1\_CTLR1寄存器)。
6. 写1到CEN位启动定时器 2 (TIMER2\_CTLR1寄存器)。

■ 用定时器 2 的使能/更新信号来启动定时器 1

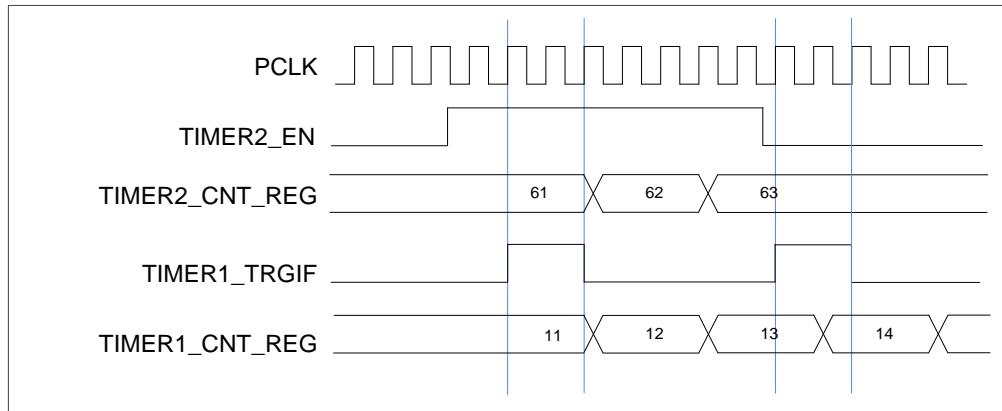
用定时器2的使能信号来启动定时器1，参见下图，在定时器2的使能输出触发的分频内部时钟下，定时器1从当前值开始计数。

当定时器1接收到触发信号，它的CEN位被自动置1，计数器计数直到关闭。两个计数器时钟频率都是PCLK时钟经过预分频器3分频( $f_{CNT\_CLK} = f_{PCLK}/3$ )。按下面步骤进行：

1. 配置定时器2为主模式，发送它的使能信号作为触发输出(配置TIMER2\_CTLR2寄存器的MMC=001)
2. 配置定时器1选择输入触发来自定时器2 (配置TIMER1\_SMC寄存器的TRGS=001)。

3. 配置定时器1在触发模式 (配置TIMER1\_SMC寄存器的SMC=110)。
4. 写1到CEN来开启定时器2 (TIMER2\_CTLR1寄存器)。

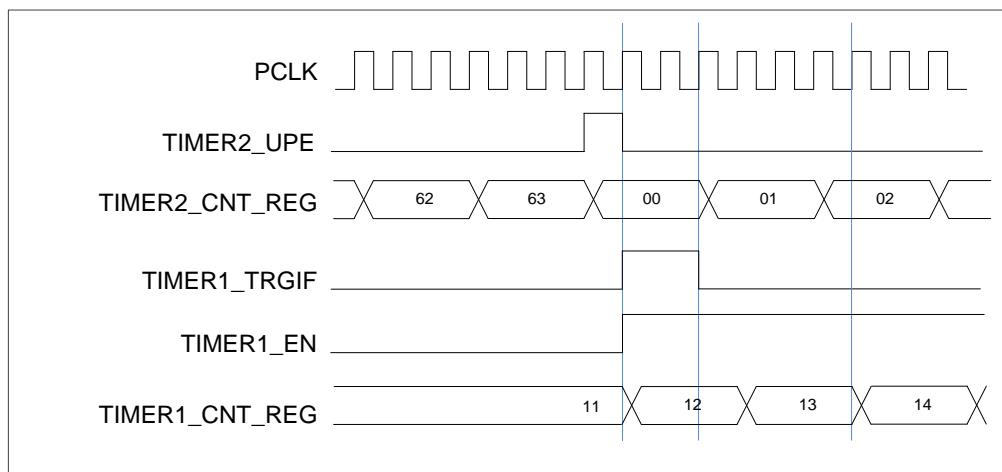
图 9-41. 用定时器 2 的使能输出触发定时器 1



在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。参考下图，按以下步骤进行：

1. 配置定时器2在主模式下，发送它的更新事件(UPE)作为触发输出(配置TIMER2\_CTLR2寄存器的MMC=010)。
2. 配置定时器2周期 (TIMER2\_CARL寄存器)。
3. 配置定时器1获取来自定时器2的输入触发信号 (配置TIMER1\_SMC寄存器的TRGS=001)。
4. 配置定时器1在触发模式 (配置TIMER1\_SMC寄存器的SMC=110)。
5. 写1到CEN位来启动定时器2 (TIMER2\_CTLR1寄存器)。

图 9-42. 用定时器 2 的更新来触发定时器 1

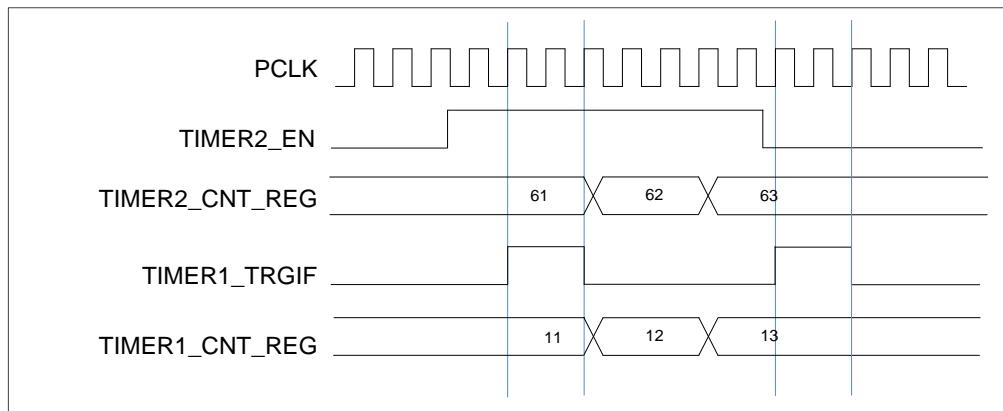


- 使用定时器 2 的使能/OC1 参考信号来使能定时器 1 计数。

在这个例子中，我们使用定时器2的使能输出来控制定时器1的使能。参考下图，在定时器2被使能后，定时器1在内部分频的时钟上开始计数。两个计数器的时钟频率都是由CK\_INT时钟三分频得来( $f_{CK\_CNT} = CK\_INT/3$ )，按以下步骤做：

1. 配置定时器2在主模式，配置其输出使能信号作为触发输出(配置TIMER2\_CTLR2寄存器的MMC=001)。
2. 配置定时器1从定时器2获取输入触发(配置TIMER1\_SMC寄存器的TRGS=001)。
3. 配置定时器1工作在暂停模式 (配置TIMER1\_SMC寄存器的SMC=101)。
4. 写1到CEN位来使能定时器1 (TIMER1\_CTLR1寄存器)。
5. 写1到CEN位来启动定时器2 (TIMER2\_CTLR1寄存器)。
6. 写0到CEN位来关闭Stop 定时器2 (TIMER2\_CTLR1寄存器)。

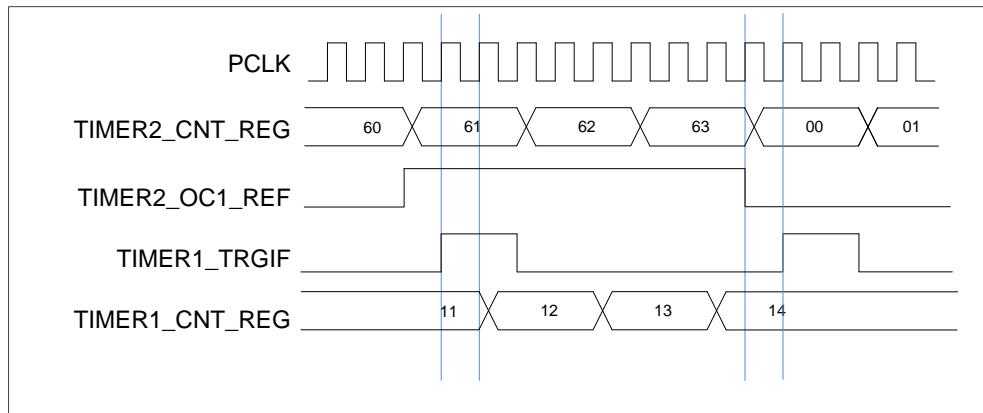
**图 9-43. 用定时器 2 的使能来选通定时器 1**



这个例子中，我们也可以使用OCx\_Ref代替使能信号输出作为触发源。按以下步骤做：

1. 配置定时器2在主模式下，配置其输出比较1参考信号(OC1REF)为触发输出(配置TIMER2\_CTLR2寄存器的MMC=100)。
2. 配置定时器2的OC1REF波形 (TIMER2\_CHCTRLR1寄存器)。
3. 配置定时器1获取来自定时器2的输入触发 (配置TIMER1\_SMC寄存器的TRGS=001)。
4. 配置定时器1工作在暂停模式 (配置TIMER1\_SMC寄存器的SMC=101)。
5. 写1到CEN位来使能定时器1 (TIMER1\_CTLR1寄存器)。
6. 写1到CEN位来开启定时器 2 (TIMER2\_CTLR1寄存器)。

图 9-44. 用定时器 2 的 OC1REF 选通定时器 1



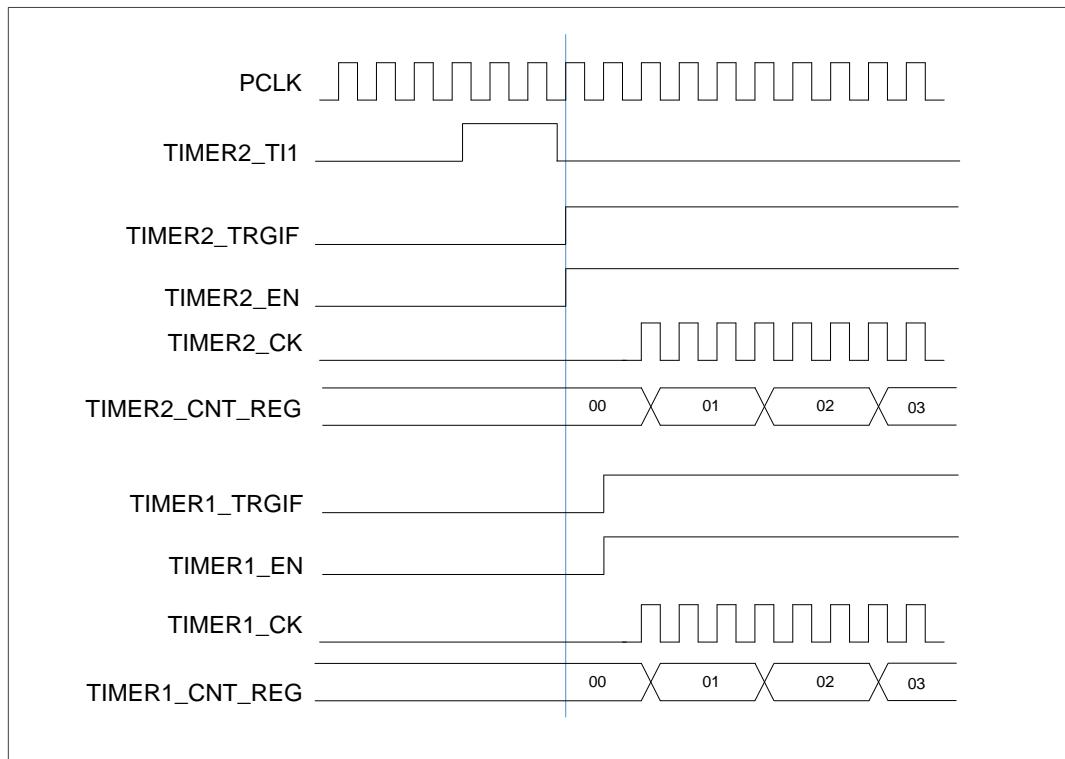
■ 使用一个外部触发来同步两个定时器

配置定时器2的使能信号触发定时器1的开启，配置定时器2的TI1输入信号上升沿来触发定时器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。按以下步骤做：

1. 配置定时器2工作在从模式来获取来自TI1的触发输入(配置TIMER2\_SMC寄存器的TRGS=100)。
2. 配置定时器2工作在触发模式 (配置TIMER2\_SMC 寄存器的SMC=110)。
3. 写MSM=1(TIMER2\_SMC寄存器)来配置定时器2工作在主/从模式。
4. 配置定时器1的触发输入来自定时器2 (配置TIMER1\_SMC 寄存器的TRGS=001)。
5. 配置定时器1工作在触发模式 (配置TIMER1\_SMC 寄存器的SMC=110)。

定时器2的TI1信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的TRGIF标志位都被置1。

图 9-45. 用定时器 2 的 TI1 输入来触发定时器 1 和定时器 2



### 定时器 debug 模式

当微控制器在调试模式，若Cortex™-M3内核停止，且MCUDBG模块中DBG\_TIMERx\_STOP配置位被置1，TIMERx计数器停止。

#### 9.1.4. TIMER1 寄存器

##### TIMER1 控制寄存器 1 (TIMER1\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CDIV[1:0]	ARSE	CAM[1:0]	DIR	SPM	UPS	UPDIS	CEN			

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CDIV[1:0]	时钟分频 软件配置CDIV位规定定时器时钟(PCLK) 和死区时间、采样时钟(DTS)之间的分频系数，该位被死区时间产生器和数字滤波器使用 00: $f_{DTS}=f_{PCLK}$

		01: $f_{DTs} = f_{PCLK}/2$ 10: $f_{DTs} = f_{PCLK}/4$ 11: 保留
7	ARSE	自动重载影子使能 0: 禁止TIMERx_CARL寄存器的影子寄存器 1: 使能TIMERx_CARL寄存器的影子寄存器
6:5	CAM[1:0]	中央对齐模式选择 00: 无中央对齐模式(边沿对齐模式). DIR位指定了计数方向 01: 中央对齐模式1。计数器交替地向上和向下计数。通道的输出比较中断标志位配置为输出模式 (TIMERx_CHCTRLx寄存器中CHxM=00)，只有在计数器向下计数时被置1。 10: 中央对齐模式2。计数器交替地向上和向下计数。通道的输出比较中断标志位配置为输出模式 (TIMERx_CHCTRLx寄存器中CHxM=00)，只有在计数器向上计数时被置1。 11: 中央对齐模式3。计数器交替地向上和向下计数. 通道的输出比较中断标志位配置为输出模式 (TIMERx_CHCTRLx寄存器中CHxM=00)，在计数器向上和向下计数时都被置1。 计数器开启时，不允许从边沿模式转换到中央对齐模式。
4	DIR	方向 0: 向上计数 1: 向下计数 当计数器配置为中央对齐模式或编码器模式时，该位为只读
3	SPM	单脉冲模式. 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
2	UPS	更新请求源 软件通过该位选择更新事件源 0: 使能后，下述任一事件产生更新中断或DMA请求： - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生的更新 1: 使能后只有计数器溢出/ 下溢才产生更新中断或DMA请求。
1	UPDIS	禁止更新 软件通过该位允许/ 禁止更新事件的产生 0: 允许更新事件产生。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生一个更新事件 1: 禁止更新事件。带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控

制器产生一个硬件复位事件，计数器和预分频器被重新初始化

0	CEN	使能计数器 0：禁止计数器 1：使能计数器 在软件将CEN位置1后，外部时钟、暂停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。
---	-----	---

### TIMER1 控制寄存器 2 (TIMER1\_CTLR2)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ISO4	ISO3N	ISO3	ISO2N	ISO2	ISO1N	ISO1	TI1S	MMC[2:0]	DMAS	CCUC	保留.	CCSE		
rw															
位/位域	名称	描述													
15	保留	必须保持复位值													
14	ISO4	通道4的空闲输出状态 参见ISO1位													
13	ISO3N	通道3的互补输出空闲状态 参见ISO1N位													
12	ISO3	通道3的空闲输出状态 参见ISO1位													
11	ISO2N	通道2的互补输出空闲状态 参见ISO1N位													
10	ISO2	通道2的空闲输出状态 参见ISO1位													
9	ISO1N	通道1的互补输出空闲状态 0: 当POE=0, OC1N=0. 1: 当POE=0, OC1N=1 只有当TIMERx_BKDT 寄存器的LK[1:0]位为00时此位才可以被修改													
8	ISO1	通道1的空闲输出状态 0: 当POE=0, OC1=0 1: 当POE=0, OC1=1 如果OC1N生效，一个死区时间后OC1输出改变。此位只有在TIMERx_BKDT 寄存器的LK[1:0]位为00的时候可以被更改													
7	TI1S	通道1触发输入选择 0: 选择TIMERx_CH1引脚输入作为通道1触发输入													

1: TIMERx\_CH1、TIMERx\_CH2 和TIMERx\_CH3 引脚经异或后连到TI1输入

6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择，主模式下发给从定时器用于同步功能 000: 复位。TIMERx_EVG寄存器的UPG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时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TIMERx_EN 被主模式控制器选为触发输出(TRGO)。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。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。 010: 更新。更新事件被主模式控制器选为触发输入。 011: 捕获/比较脉冲.在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲。 100: 比较。OC1REF 信号被用于作为触发输出 101: 比较。OC2REF 信号被用于作为触发输出 110: 比较。OC3REF 信号被用于作为触发输出 111: 比较。OC4REF 信号被用于作为触发输出
3	DMAS	DMA请求源选择 0: 当通道x事件发生，发送通道x的DMA请求 1: 当更新事件发生，发送通道x的DMA请求
2	CCUC	捕获/比较控制影子寄存器更新控制 使能(CCSE=1)捕获/比较控制影子寄存器(CHxE、OCxNE和CMPxM 位)后，该位控制这些影子寄存器更新的时间 0: CCUG位被置1时更新影子寄存器 1: 当CCUG位被置1或检测到TRGI上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值.
0	CCSE	捕获/比较控制影子寄存器使能 0: 影子寄存器CHxE、CHxNE和CHxOM位被禁止 1: 使能影子寄存器CHxE、CHxNE和CHxOM 位，当这些位被写时，只有当CCUG位被置1时这些位才被更新 当通道没有互补输出时，此位无效。

### TIMER1 从模式控制寄存器 (TIM1\_SMC)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETPL	ECM2E	ETPSC[1:0]		ETFC[3:0]		MSM		TRGS[2:0]		OCRC.		SMC[2:0]			

rw rw

位/位域	名称	描述
15	ETPL	<p>外部触发极性</p> <p>该位选择用ETI还是ETI的反相来作为触发操作。</p> <p>0: ETI不反相, 高电平或上升沿有效</p> <p>1: ETI被反相, 低电平或下降沿有效</p>
14	ECM2E	<p>外部时钟模式2使能</p> <p>在外部时钟模式2, 计数器由ETIF信号上的任意有效边沿驱动。</p> <p>0: 禁止外部时钟模式2</p> <p>1: 使能外部时钟模式2</p> <p>将ECM2E 位置1与选择外部时钟模式1并将TRGI连到ETIF(SMC=111和TRGS=111)具有相同功效。</p> <p>下述从模式可以与外部时钟模式2同时使用: 复位模式, 暂停模式和触发模式; 但是,</p> <p>这时TRGI不能是'111'。</p> <p>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ETIF</p>
13:12	ETPSC[1:0]	<p>外部触发预分频</p> <p>外部触发信号ETIP的频率最多是TIMxCLK 频率的1/4。当输入较快的外部时钟时, 可以使用预分频降低ETIP 的频率。</p> <p>00: 禁止预分频</p> <p>01: ETIP频率除以2</p> <p>10: ETIP频率除以4</p> <p>11: ETIP频率除以8</p>
11:8	ETFC[3:0]	<p>外部触发滤波控制</p> <p>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。这些位定义了对ETIP信号采样的频率和对ETIP 数字滤波的带宽。</p> <p>0000: 无滤波器. <math>f_{SAMP} = f_{DTS}</math>, <math>N=1</math>.</p> <p>0001: <math>f_{SAMP} = f_{PCLK}</math>, <math>N=2</math>.</p> <p>0010: <math>f_{SAMP} = f_{PCLK}</math>, <math>N=4</math>.</p> <p>0011: <math>f_{SAMP} = f_{PCLK}</math>, <math>N=8</math>.</p> <p>0100: <math>f_{SAMP}=f_{DTS}/2</math>, <math>N=6</math>.</p> <p>0101: <math>f_{SAMP}=f_{DTS}/2</math>, <math>N=8</math>.</p> <p>0110: <math>f_{SAMP}=f_{DTS}/4</math>, <math>N=6</math>.</p> <p>0111: <math>f_{SAMP}=f_{DTS}/4</math>, <math>N=8</math>.</p> <p>1000: <math>f_{SAMP}=f_{DTS}/8</math>, <math>N=6</math>.</p> <p>1001: <math>f_{SAMP}=f_{DTS}/8</math>, <math>N=8</math>.</p> <p>1010: <math>f_{SAMP}=f_{DTS}/16</math>, <math>N=5</math>.</p> <p>1011: <math>f_{SAMP}=f_{DTS}/16</math>, <math>N=6</math>.</p> <p>1100: <math>f_{SAMP}=f_{DTS}/16</math>, <math>N=8</math>.</p> <p>1101: <math>f_{SAMP}=f_{DTS}/32</math>, <math>N=5</math>.</p> <p>1110: <math>f_{SAMP}=f_{DTS}/32</math>, <math>N=6</math>.</p> <p>1111: <math>f_{SAMP}=f_{DTS}/32</math>, <math>N=8</math>.</p>
7	MSM	主-从模式

主-从模式下输入触发上的事件被延迟了，以允许当前定时器和它的从设备通过TRGO完美同步。如果想要在单个外部事件上同步一些定时器，此模式非常有用。

0: 禁止主-从模式

1: 使能主-从模式

6:4	TRGS[2:0]	触发选择  这些位选择用于同步计数器的触发输入  000: 内部触发输入0 (ITR0) TIMER15 001: 内部触发输入1 (ITR1) TIMER2 010: 内部触发输入2 (ITR2) TIMER3 011: 内部触发输入3 (ITR3) 保留 100: TI1的边沿标志位 (TI1F_ED) 101: 滤波后的通道1输入 (TI1FP1) 110: 滤波后的通道2输入 (TI2FP2) 111: 外部触发输入(ETIF)  从模式被使能后这些位不能改
3	OCRC	OCREF清零信号选择控制  0: OCREF_CLR作为OCREF的清零信号 1: ETIF 作为OCREF的清零信号
2:0	SMC[2:0]	从模式控制  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. 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器 101: 暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止 110: 触发模式. 计数器在触发输入TRGI 的上升沿启动。计数器不能被从模式控制器关闭。 111: 外部时钟模式1. 选中的触发输入的上升沿驱动计数器  如果TI1F_ED 被选为触发输入时，不要使用暂停模式。这是因为，TI1F_ED 在每次TI1F 变化时输出一个脉冲，然而暂停模式是要检查触发输入的电平

## TIMER1 DMA 和中断使能寄存器 (TIMER1\_DIE)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDE	CCUDE	CH4DE	CH3DE	CH2DE	CH1DE	UPDE	BKIE	TRGIE	CCUIE	CH4IE	CH3IE	CH2IE	CH1IE	UPIE

---

位/位域	名称	描述
15	保留	必须保持复位值.
14	TRGDE	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	CCUDE	通道控制DMA更新请求使能 0: 禁止通道控制DMA更新请求 1: 使能通道控制DMA更新请求
12	CH4DE	通道4 DMA请求使能 0: 禁止通道4 DMA请求 1: 使能通道4 DMA请求
11	CH3DE	通道3 DMA请求使能 0: 禁止通道3 DMA请求 1: 使能通道3 DMA请求
10	CH2DE	通道2 DMA请求使能 0: 禁止通道2 DMA请求 1: 使能通道2 DMA请求
9	CH1DE	通道1 DMA请求使能 0: 禁止通道1 DMA请求 1: 使能通道1 DMA请求
8	UPDE	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CCUIE	通道控制更新中断使能 0: 禁止通道控制更新中断 1: 使能通道控制更新中断
4	CH4IE	通道4中断使能 0: 禁止通道4中断 1: 使能通道4中断
3	CH3IE	通道3中断使能 0: 禁止通道3中断

		1: 使能通道3中断
2	CH2IE	通道2中断使能 0: 禁止通道2中断 1: 使能通道2中断
1	CH1IE	通道1中断使能 0: 禁止通道1中断 1: 使能通道1中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### TIMER1 状态寄存器 (TIMER1\_STR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH4OF	CH3OF	CH2OF	CH1OF	保留.	BKIF	TRGIF	CCUIF	CH4IF	CH3IF	CH2IF	CH1IF	UPIF	rc_w0	rc_w0

位/位域	名称	描述
15:13	保留	必须保持复位值
12	CH4OF	通道4重复捕获标志 参见CH1OF描述
11	CH3OF	通道3重复捕获标志 参见CH1OF描述
10	CH2OF	通道2重复捕获标志 参见CH1OF描述
9	CH1OF	通道1重复捕获标志 当通道1被配置为输入模式时，在CH1IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。 0: 无重复捕获中断发生 1: 发生了重复捕获中断
8	保留	必须保持复位值。
7	BKIF	中止中断标志位 一旦中止输入有效，由硬件对该位置‘1’。如果中止输入无效，则该位可由软件清‘0’。 0: 无中止事件产生 1: 中止输入上检测到有效电平

---

6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置1。此位由软件清0。当从模式控制器处于除暂停模式外的其它模式时，在TRGI 输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，TRGI的任意边沿都可以产生触发事件。 0: 无触发事件产生 1: 触发中断产生
5	CCUIF	通道控制更新中断标志 当通道控制更新事件发生时此标志位被硬件置1，此位由软件清0。 0: 无通道控制更新中断发生 1: 通道控制更新中断发生
4	CH4IF	通道4中断标志 参见CH1IF描述
3	CH3IF	通道3中断标志 参见CH1IF描述
2	CH2IF	通道2中断标志 参见CH1IF描述
1	CH1IF	通道1中断标志 此标志由硬件置1软件清0。当通道1在输入模式下时，捕获事件发生时此标志位被置1；当通道1在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道1中断发生 1: 通道1中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

### TIMER1 事件产生寄存器 (TIMER1\_EVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BKG	TRGG	CCUG	CH4G	CH3G	CH2G	CH1G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
15:8	保留	必须保持复位值.
7	BKG	产生中止事件 该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POE位

---

		被清0且BKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。
		0: 不产生中止事件 1: 产生中止事件
6	TRGG	触发事件产生  此位由软件置1，由硬件自动清0。当此位被置1，TIMERx_STR寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。  0: 无触发事件产生 1: 产生触发事件
5	CCUG	通道控制更新事件发生  此位由软件置1，由硬件自动清0。当此位被置1，通道控制寄存器 (CHxE, CHxNE and CHxOM位) 的互补输出被更新。  0: 不产生通道控制更新事件 1: 产生通道控制更新事件
4	CH4G	通道4捕获或比较事件发生  参见CH1G描述
3	CH3G	通道3捕获或比较事件发生  参见CH1G描述
2	CH2G	通道2捕获或比较事件发生  参见CH1G描述
1	CH1G	通道1捕获或比较事件发生  该位由软件置1，用于在通道1产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CC1IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道1配置为输入模式，计数器的当前值被TIMER1_CHCC1寄存器捕获，如果CH1IF标志位已经为1，则CH1OF标志位被置1。  0: 不产生通道1捕获或比较事件 1: 发生通道1捕获或比较事件
0	UPG	此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。  0: 无更新事件产生 1: 产生更新事件

## TIMER1 通道控制寄存器 1 (TIMER1\_CHCTRLR1)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2 OCE	CH2OM[2:0]	CH2 OSE	CH2 OFE	CH2M[1:0]	CH1 OCE	CH1OM[2:0]	CH1OSE	CH1 OFE	CH1M[1:0]						

CH2ICF[3:0]				CH2ICP[1:0]				CH1ICF[3:0]				CH1ICP[1:0]			
rw	rw	rw	rw												

**输出比较模式:**

位/位域	名称	描述
15	CH2OCE	通道2输出比较清0使能 参见CH1OCE描述
14:12	CH2OM[2:0]	通道2输出比较模式 参见CH1OM描述
11	CH2OSE	通道2输出比较影子寄存器使能 参见CH1OSE描述
10	CH2OFE	通道2输出比较快速使能 参见CH1OFE描述
9:8	CH2M[1:0]	通道2模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMER1_CHE寄存器的CH2E位被清0)时这些位才可以写。 00: 通道2配置为输出 01: 通道2配置为输入， IC2映射在TI2上 10: 通道2配置为输入， IC2映射在TI1上 11: 通道2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 TIMER_SMC寄存器的TRGS位选择)。
7	CH1OCE	通道1输出比较清0使能 当此位被置1，当检测到ETIF输入高电平时，OC1参考信号被清0 0: 使能通道1输出比较清零 1: 禁止通道1输出比较清零
6:4	CH1OM[2:0]	通道1输出比较模式 此位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。 OC1REF高电平有效，而OC1、OC1N的有效电平取决于CH1P、CH1NP位。 000: 冻结。输出比较寄存器TIMER1_CHCC1与计数器间的比较对OC1REF不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较寄存器TIMER1_CHCC1相同时，强制OC1REF为高。 010: 匹配时设置为低。当计数器的值与捕获/比较寄存器TIMER1_CHCC1相同时，强制OC1REF为低。 011: 匹配时翻转。当计数器的值与捕获/比较寄存器TIMER1_CHCC1相同时，强制OC1REF翻转。 100: 强制为低。强制OC1REF为低电平 101: 强制为高。强制OC1REF为高电平 110: PWM 模式1。在向上计数时，一旦计数器值小于TIMER1_CHCC1时，OC1REF为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMER1_CHCC1时，OC1REF 为低电平，否则为高电平。 111: PWM 模式2。在向上计数时，一旦计数器值小于TIMER1_CHCC1时，OC1REF

为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMER1\_CHCC1时，OC1REF为高电平，否则为低电平。

在PWM模式1或PWM模式2中，只有当比较结果改变了或者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。

当TIMERx\_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。

3	<b>CH1OSE</b>	通道1输出比较影子寄存器使能  当此位被置1，TIMER1_CHCC1寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。  0: 禁止通道1输出/比较影子寄存器 1: 使能通道1输出/比较影子寄存器  仅在单脉冲模式下(TIMER1_CTLR1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  当TIMERx_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。
2	<b>CH1OFE</b>	通道1输出比较快速使能  该位用于通道配置为PWM模式时加快CC输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，OC被设置为比较电平而与比较结果无关。  0: 禁止通道1输出比较快速. 当触发器的输入有一个有效沿时，激活CC1输出的最小延时为5个时钟周期  1: 使能通道1输出比较快速。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3个时钟周期
1:0	<b>CH1M[1:0]</b>	通道1模式选择  这些位定义了输入的方向和输入信号的选择。只有当通道关闭(TIMER1_CHE寄存器的CH1E位被清0)时这些位才可写。  00: 通道1配置为输出 01: 通道1配置为输入，IC1映射在TI1上 10: 通道1配置为输入，IC1映射在TI2上 11: 通道1配置为输入，IC1映射在TRC上. 此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)

#### 输入捕获模式:

位/位域	名称	描述
15:12	<b>CH2ICF[3:0]</b>	通道2输入捕获滤波控制  参见CH1ICF描述
11:10	<b>CH2ICP[1:0]</b>	通道2输入捕获预分频器  参见CH1ICP描述
9:8	<b>CH2M[1:0]</b>	通道2模式选择  此位定义了通道方向和输入信号的选择。只有在通道关闭(TIMER1_CHE寄存器的CH2E位被清0)时这些位才可写。  00: 通道2配置为输出 01: 通道2配置为输入，IC2映射在TI2上 10: 通道2配置为输入，IC2映射在TI1上

11: 通道2配置为输入, IC2映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER\_SMC寄存器的TRGS位)。

7:4	CH1ICF[3:0]	通道1输入捕获滤波控制 数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。这些位定义了TI1输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$ , N=1 0001: $f_{SAMP} = f_{PCLK}$ , N=2 0010: $f_{SAMP} = f_{PCLK}$ , N=4 0011: $f_{SAMP} = f_{PCLK}$ , N=8 0100: $f_{SAMP} = f_{DTS}/2$ , N=6 0101: $f_{SAMP} = f_{DTS}/2$ , N=8 0110: $f_{SAMP} = f_{DTS}/4$ , N=6 0111: $f_{SAMP} = f_{DTS}/4$ , N=8 1000: $f_{SAMP} = f_{DTS}/8$ , N=6 1001: $f_{SAMP} = f_{DTS}/8$ , N=8 1010: $f_{SAMP} = f_{DTS}/16$ , N=5 1011: $f_{SAMP} = f_{DTS}/16$ , N=6 1100: $f_{SAMP} = f_{DTS}/16$ , N=8 1101: $f_{SAMP} = f_{DTS}/32$ , N=5 1110: $f_{SAMP} = f_{DTS}/32$ , N=6 1111: $f_{SAMP} = f_{DTS}/32$ , N=8
3:2	CH1ICP[1:0]	通道1输入捕获预分频器 这2位定义了CC1 输入的预分频系数。一旦CH1E =0(TIMER1_CHE寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH1M[1:0]	通道1模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMER1_CHE寄存器的CH1E位被清0)时这些位才可写。 00: 通道1配置为输出 01: 通道1配置为输入, IC1映射在TI1上 10: 通道1配置为输入, IC1映射在TI2上 11: 通道1配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)。

### TIMER1 通道控制寄存器 2 (TIMER1\_CHCTRL2)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CH4 OCE	CH4OM[2:0]		CH4 OSE	CH4 OFE	CH4M[1:0]	CH3 OCE	CH3OM[2:0]		CH3 OSE	CH3 OFE	CH3M[1:0]					
CH4ICF[3:0]		CH4ICP[1:0]				CH3ICF[3:0]		CH3ICP[1:0]								
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

**输出比较模式:**

位/位域	名称	描述
15	CH4OCE	通道4输出比较清0使能 参见CH1OCE描述
14:12	CH4OM[2:0]	通道4输出比较模式 参见CH1OM描述
11	CH4OSE	通道4输出比较影子寄存器使能 参见CH1OSE描述
10	CH4OFE	通道4输出比较快速使能 参见CH1OFE描述
9:8	CH4M[1:0]	通道4模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMER1_CHE寄存器的CH4E位被清0)时这些位才可写。 00: 通道4配置为输出 01: 通道4配置为输入, IC4映射在TI4上 10: 通道4配置为输入, IC4映射在TI3上 11: 通道4配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)。
7	CH3OCE	通道3输出比较清0使能 参见CH1OCE描述
6:4	CH3OM[2:0]	通道3输出比较模式 参见CH1OM描述
3	CH3OSE	通道3输出比较影子寄存器使能 参见CH1OSE描述
2	CH3OFE	通道3输出比较快速使能 参见CH1OFE描述
1:0	CH3M[1:0]	通道1模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMER1_CHE寄存器的CH3E位被清0)时这些位才可写。 00: 通道3配置为输出 01: 通道3配置为输入, IC3映射在TI3上 10: 通道3配置为输入, IC3映射在TI4上 11: 通道3配置为输入, IC3映射在TRC上。此模式仅工作在内部触发输入被选中时

(通过设置TIMERx\_SMC寄存器的TRGS位)。

#### 输入捕获模式:

位/位域	名称	描述
15:12	CH4ICF[3:0]	通道4输入捕获滤波控制 参见CH1ICF描述
11:10	CH4ICP[1:0]	通道4输入捕获预分频器 参见CH1ICP描述
9:8	CH4M[1:0]	通道4模式选择 这两位定义了通道的方向和输入信号的选择。只有在通道关闭(TIMER1_CHE寄存器的CH4E位被清0)时这些位才可写。 00: 通道4配置为输出 01: 通道4配置为输入, IC4映射在TI4上 10: 通道4配置为输入, IC4映射在TI3上 11: 通道4配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时 (通过设置TIMER_SMC寄存器的TRGS位)。
7:4	CH3ICF[3:0]	通道3输入捕获滤波控制 参见CH1ICF描述
3:2	CH3ICP[1:0]	通道3输入捕获预分频器 参见CH1ICP描述
1:0	CH3M[1:0]	通道3模式选择 这两位定义了通道的方向和输入信号的选择。只有在通道关闭(TIMER1_CHE寄存器的CH3E位被清0)时这些位才可写。 00: 通道3配置为输出 01: 通道3配置为输入, IC3映射在TI3上 10: 通道3配置为输入, IC3映射在TI4上 11: 通道3配置为输入, IC3映射在TRC上。此模式仅工作在内部触发输入被选中时 (通过设置TIMER_SMC寄存器的TRGS位)。

#### TIMER1 通道使能寄存器 (TIMER1\_CHE)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH4P	CH4E	CH3NP	CH3NE	CH3P	CH3E	CH2NP	CH2NE	CH2P	CH2E	CH1NP	CH1NE	CH1P	CH1E	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
------	----	----

15:14	保留	必须保持复位值。
13	CH4P	通道4极性 参见CH1P描述
12	CH4E	通道4使能 参见CH1E描述
11	CH3NP	通道3互补输出极性 参见CH1NP描述
10	CH3NE	通道3互补输出使能 参见CH1NE描述
9	CH3P	通道3极性 参见CH1P描述
8	CH3E	通道3使能 参见CH1E描述
7	CH2NP	通道2互补输出极性 参见CH1NP描述
6	CH2NE	通道2互补输出使能 参见CH1NE描述
5	CH2P	通道2极性 参见CH1P描述
4	CH2E	通道2使能 参见CH1E描述
3	CH1NP	通道1互补输出极性 当通道1配置为输出模式，此位定义了互补输出信号的极性。 0：通道1高电平有效 1：通道1低电平有效 当通道1配置为输入模式时，此位和CH1P联合使用，作为输入信号的极性选择控制信号，详情请参照CH1P的说明。 当TIMERx_BKDT寄存器的LK[1:0]=11或10 时此位不能被更改。
2	CH1NE	通道1互补输出使能 当通道1配置为输出模式时，将此位置1使能通道1的互补输出。 0：禁止通道1互补输出 1：使能通道1互补输出
1	CH1P	通道1极性 当通道1配置为输出模式时，此位定义了输出信号极性。 0：通道1高电平有效 1：通道1低电平有效 当通道1配置为输入模式时，

[CH1NP, CH1P] 作为输入信号的极性选择控制位，功能如下：

[CH1NP==0, CH1P==0]: 把TIxFP1的上升沿作为捕获/触发的有效信号，并且TIxFP1不会被翻转。

[CH1NP==0, CH1P==1]: 把TIxFP1的下降沿作为捕获/触发的有效信号，并且TIxFP1会被翻转。

[CH1NP==1, CH1P==0]: 保留。不可设置。

[CH1NP==1, CH1P==1]: 把TIxFP1的上升沿和下降沿都作为捕获/触发的有效信号，并且TIxFP1不会被翻转。

当TIMERx\_BKDT寄存器的LK[1:0]=11或10 时此位不能被更改。

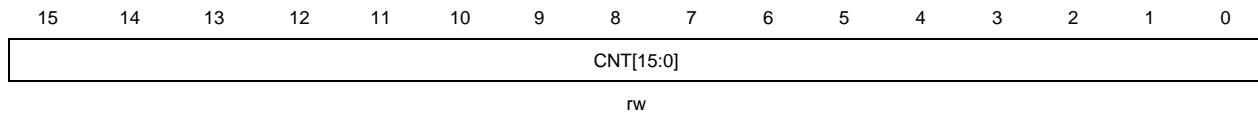
0	CH1E	通道1使能
		当通道1配置为输入模式时，将此位置1使能OC1信号有效。当通道1配置为输出模式时，将此位置1使能通道1上的捕获事件。
	0:	禁止通道1
	1:	使能通道1

## TIMER1 计数器寄存器 (TIMER1\_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



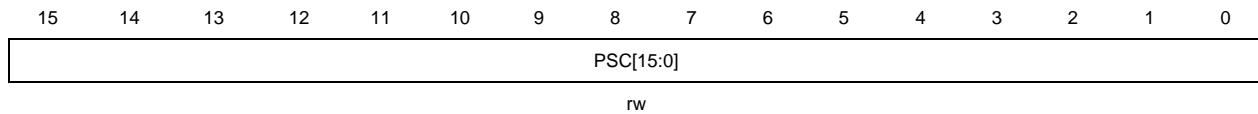
位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

## TIMER1 预分频寄存器 (TIMER1\_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值  计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

### TIMER1 计数器自动重载寄存器 (TIMER1\_CARL)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

### TIMER1 重复计数寄存器 (TIMER1\_CREP)

地址偏移: 0x30

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:8	保留	必须保持复位值.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

### TIMER1 通道 1 捕获/比较寄存器 (TIMER1\_CHCC1)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC1[15:0]															
rw															

位/位域	名称	描述
15:0	CHCC1[15:0]	通道1的捕获或比较值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器

为只读。

当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMER1 通道 2 捕获/比较寄存器 (TIMER1\_CHCC2)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC2[15:0]															
rw															

位/位域	名称	描述
15:0	CHCC2[15:0]	通道2的捕获或比较值 当通道2配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道2配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMER1 通道 3 捕获/比较寄存器 (TIMER1\_CHCC3)

地址偏移: 0x3C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC3[15:0]															
rw															

位/位域	名称	描述
15:0	CHCC3[15:0]	通道3的捕获或比较值 当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMER1 通道 4 捕获/比较寄存器 (TIMER1\_CHCC4)

地址偏移: 0x40

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC4[15:0]															
rw															

位/位域	名称	描述
15:0	CHCC4[15:0]	<p>通道4的捕获或比较值</p> <p>当通道4配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道4配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

### TIMER1 中止和死区时间寄存器 (TIMER1\_BKDT)

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POE	OAE	BRKP	BRKE	ROS	IOS	LK[1:0]									DT[7:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
15	POE	<p>主输出使能</p> <p>根据OAE位的设置值，该位可以由软件清0或被硬件自动置1。一旦中止输入有效，该位被硬件异步清0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHE寄存器的CHxE、CHxNE位)，则开启OC 和OCN 输出。</p> <p>0: 禁止OC 和OCN输出或强制为空闲状态</p> <p>1: 使能通道输出</p>
14	OAE	<p>自动输出使能</p> <p>此位定义了POE位是否可以被硬件自动置1。</p> <p>0: POE位不能被硬件置1</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POE位能被硬件自动置1</p> <p>此位只有在TIMERx_BKDT寄存器的LK[1:0]=00时才可修改。</p>
13	BRKP	<p>中止极性</p> <p>此位定义了中止输入信号的极性。</p> <p>0: 中止输入低电平有效</p> <p>1: 中止输入高电平有效</p>
12	BRKE	<p>中止使能</p> <p>此位置1使能中止事件和CCS时钟失败事件输入。</p> <p>0: 禁止中止输入</p>

		1: 使能中止输入 只有在TIMERx_BKDT寄存器的LK[1:0]=00时此位才可修改。
11	ROS	运行模式下“关闭状态”配置 当POE位被置1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。 0: 当POE位被置1，通道输出信号(OC/OCN)被禁止 1: 当POE位被置1，通道输出信号(OC/OCN)被使能，和TIMER1_CHE寄存器CHxE/CHxNE位有关 此位在TIMERx_BKDT寄存器的LK[1:0]=10或11时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当POE位被清0，此位定义了已经配置为输出模式的通道的输出状态。 0: 当POE位被清0，通道输出信号(OC/OCN)被禁止 1: 当POE位被清0，通道输出信号(OC/OCN)被使能，和TIMER1_CHE寄存器CHxE/CHxNE位有关 此位在TIMERx_BKDT寄存器的LK[1:0]=10或11时不能被更改。
9:8	LK[1:0]	锁定控制 这两位定义了寄存器的写保护特性。 00: 禁止锁定，无写保护 01: 锁定模式1，TIMER1_CTRL2寄存器的ISOx/ISOxN位和TIMER1_BKDT寄存器的BRKE/BRKP/OAE/DT位写保护 10: 锁定模式2，除了锁定模式1下的寄存器，TIMERx_CHE寄存器的CHxP/CHxNP位(如果相应通道配置为输出模式)和TIMER1_BKDT寄存器的ROS/IOS位写保护 11: 锁定模式3，除了锁定模式2下的寄存器，TIMER1_CHCTRLx寄存器的CHxOM/CHxOSE(如果相应通道配置为输出)位写保护 系统复位后这两位只能被写一次，一旦TIMERx_BKDT寄存器被写入，这两位被写保护
7:0	DT[7:0]	死区时间值 这些位定义了插入互补输出之间的死区持续时间。DT值和死区时间的关系如下： DT[7:5]=0xx: DT=DT[7:0]x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> DT[7:5]=10x: DT=(64+DT[5:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> *2 DT[7:5]=110: DT=(32+DT[4:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> *8 DT[7:5]=111: DT=(32+DT[4:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> *16 此位只有在TIMERx_BKDT寄存器的LK[1:0]=00时才可修改。

## TIMER1 DMA 控制寄存器 (TIMER1\_DCTRLR)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

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

位/位域	名称	描述
15:14	保留	必须保持复位值
12:8	DBLTH[4:0]	DMA连续传送长度 这5位定义了DMA 在连续模式下的传送长度( 当对TIMERx_DTRSF寄存器进行读或写时，定时器则进行一次连续传送)。
7:5	保留	必须保持复位值
4:0	DBAR[4:0]	DMA地址 这5位定义了DMA 传送的地址( 当对TIMx_DTRSF寄存器进行读或写时，DBA定义为从TIMx_CTLR1 寄存器所在地址开始的偏移量)。

### TIMER1 DMA 传送寄存器 (TIMER1\_DTRSF)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	DTRSF[15:0]	DMA传送 对TIMER1_DTRSF寄存器的读或写会导致对从TIMERx_CTLR1 寄存器开始的以下地址范围所在寄存器的访问: (DBAR + 连续传送计数器) x 4 连续传送计数器由硬件计算，范围为0到DBLTH。

### TIMER1 配置寄存器 (TIMER1\_CFG) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CCSEL	<p>写捕获比较寄存器选择位 此位由软件写1或清0。</p> <p>1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无效</p>
0	OUTSEL	<p>输出值选择位 此位由软件写1或清0。</p> <p>1: 如果POE位与IOS位均为0，则输出无效 0: 无效</p>

## 9.2. 通用定时器 (TIMER2 和 TIMER3)

### 9.2.1. 简介

通用定时器(TIMER2和TIMER3)由一个16位或32位向上/向下计数器、4个捕获/比较寄存器(TIMERx\_CHCC)、一个自动重载寄存器(TIMERx\_CARL)和一些控制寄存器组成。它适合多种用途，包括通用定时器，输入信号脉宽测量或输出波形产生(单脉冲或PWM输出)。TIMERx支持一个编码器接口，该接口使用一个带有两路输入的译码器。

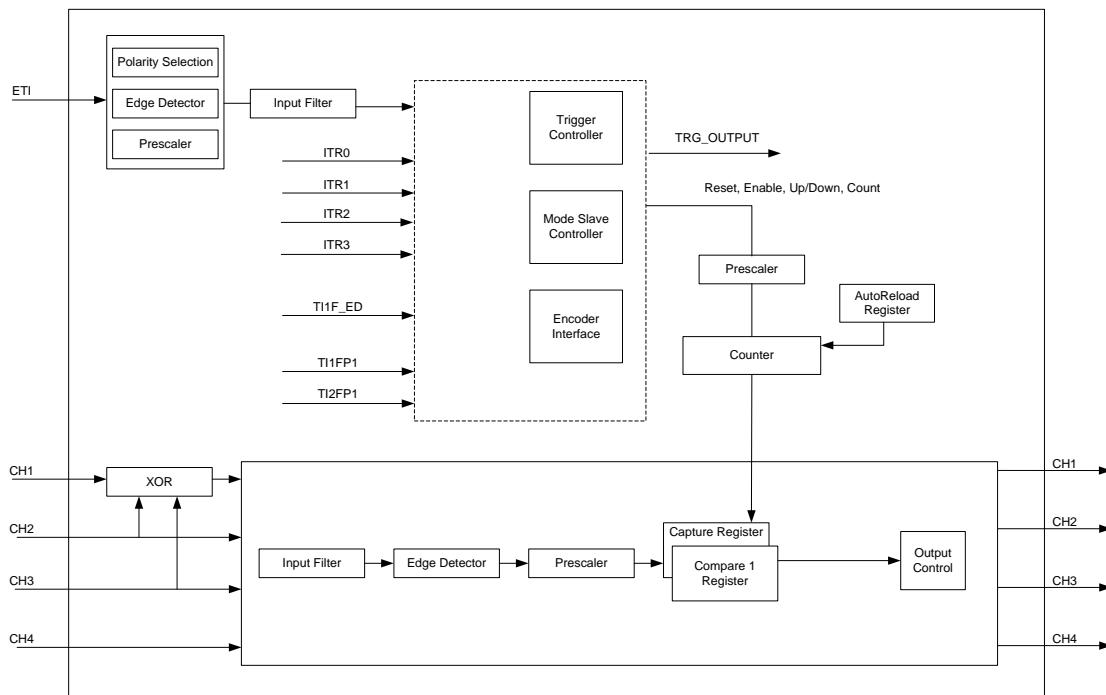
### 9.2.2. 主要特性

- 16位 (TIMER3)或32位(TIMER2)向下、向上及向下/向上自动重载计数器
- 16位可配置的预分频器，计数器时钟频率的分频系数为1到65536的任何数值
- 多达4个独立通道，支持的功能包括输入捕获，比较匹配输出，PWM产生(边沿或中央对齐模式)，以及单脉冲模式输出
- 更新事件，输入捕获事件，输出比较事件，中止输入，触发事件可以产生中断/DMA
- 使用外部信号控制定时器和定时器互连的同步电路
- 带有使用正交译码器的两路输入的编码器接口控制器
- TIMERx 主/从模式控制

### 9.2.3. 功能描述

下图提供了通用定时器的内部配置详情

图 9-46. 通用定时器框图(TIMER2 和 TIMER3)



### 预分频计数器

预分频器可以将定时器(PCLK)的时钟频率按1到65536之间的任意值分频为计数器时钟(CNT\_CLK)。它是受预分频寄存器(TIMERx\_PSC)控制，这个控制寄存器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 9-47. 当预分频器的参数从 1 变到 2 时，计数器时序图

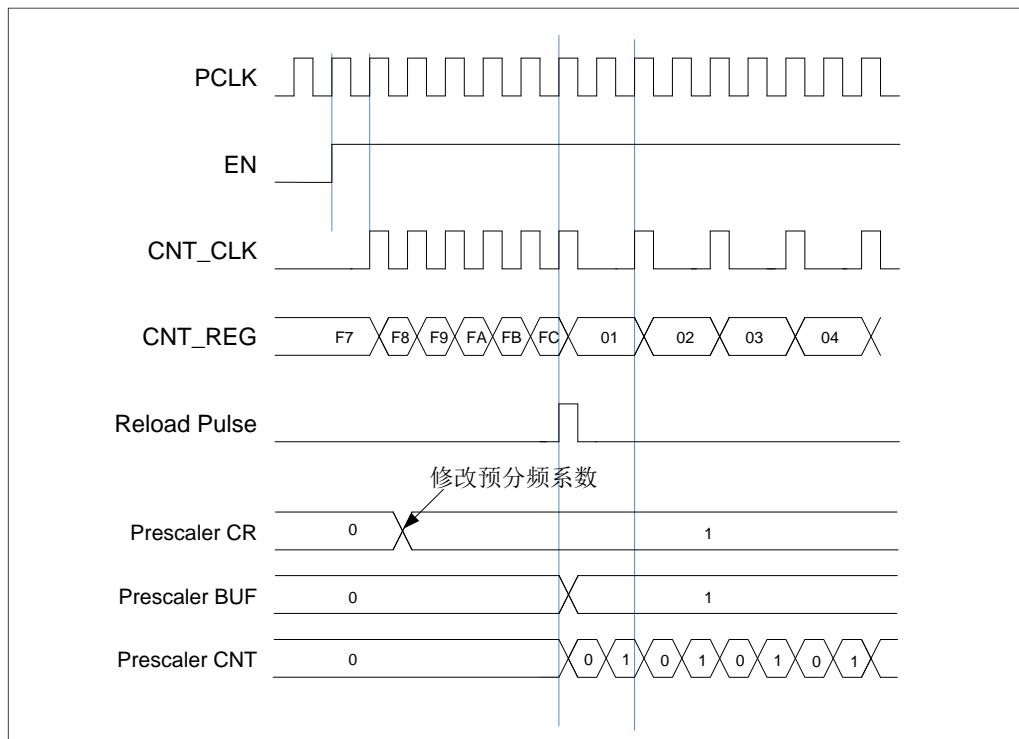
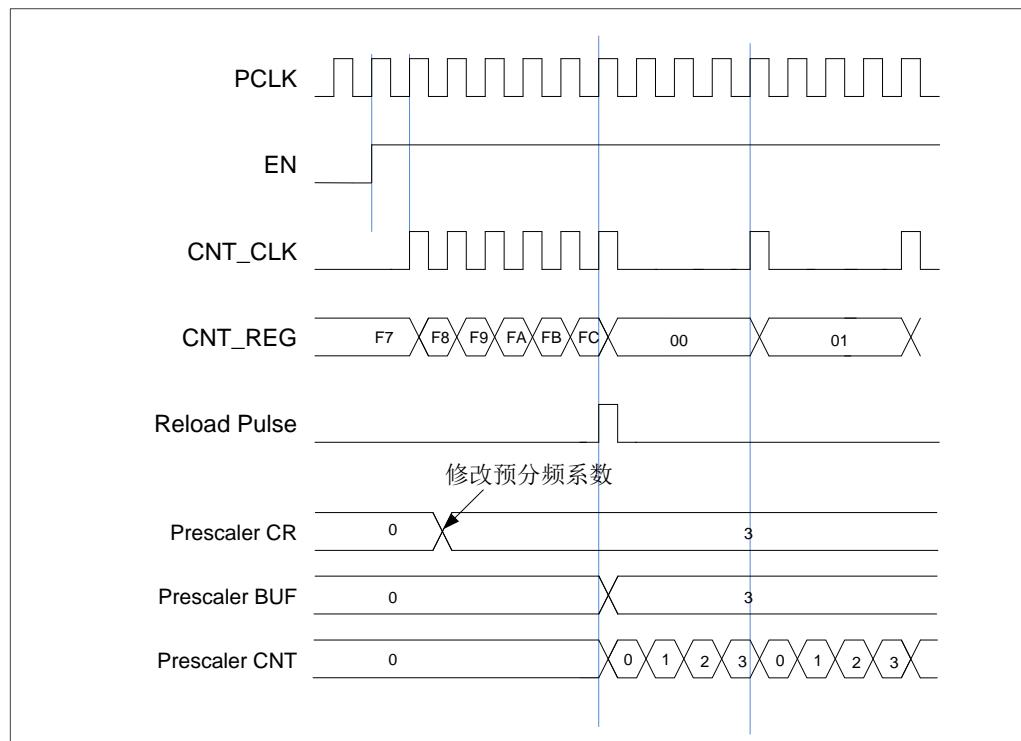


图 9-48. 当预分频器的参数从 1 变到 4 时，计数器时序图



### 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 (TIMERx\_CARL计数器的内容)，然后重新从0开始计数并且产生一个计数器溢出事件。向上计数模式中TIMERx\_CTRLR1寄存器的计数方向控制位DIR应该被设置为0。

通过软件将TIMERx\_EVG寄存器的UPG位置1可以产生更新事件，计数值将被初始化为0然后产生更新事件。

将TIMERx\_CTRLR1寄存器的UPDIS位置1将禁止更新事件。

当更新事件发生时，所有的寄存器(自动重载寄存器，预分频寄存器)都被更新。

下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作。

图 9-49. 计数器时序图，内部时钟分频系数为 1

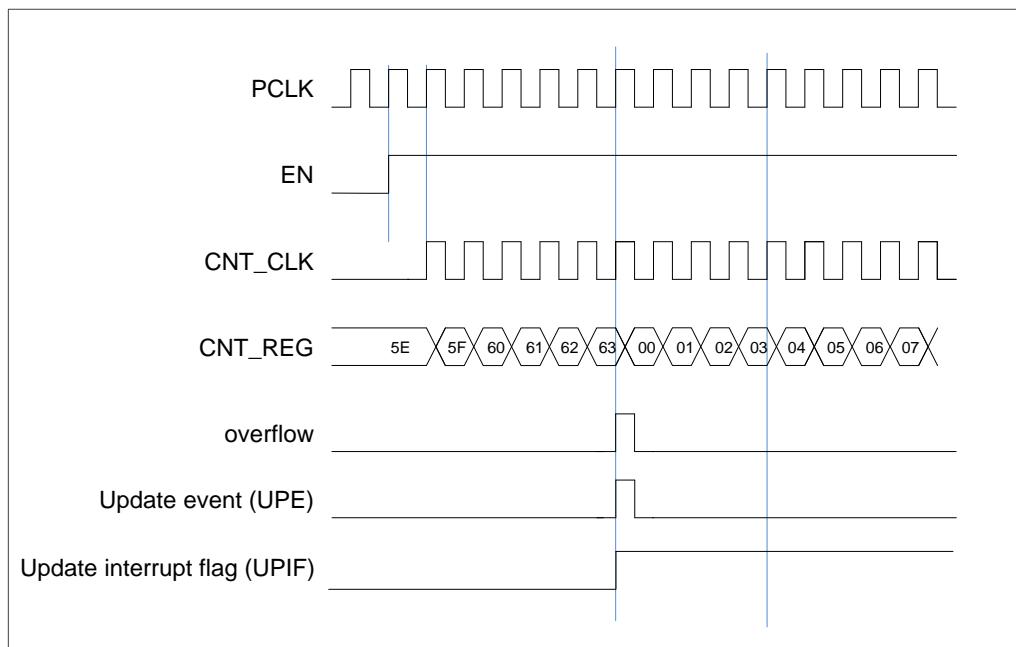


图 9-50. 计数器时序图，内部时钟分频系数为 2

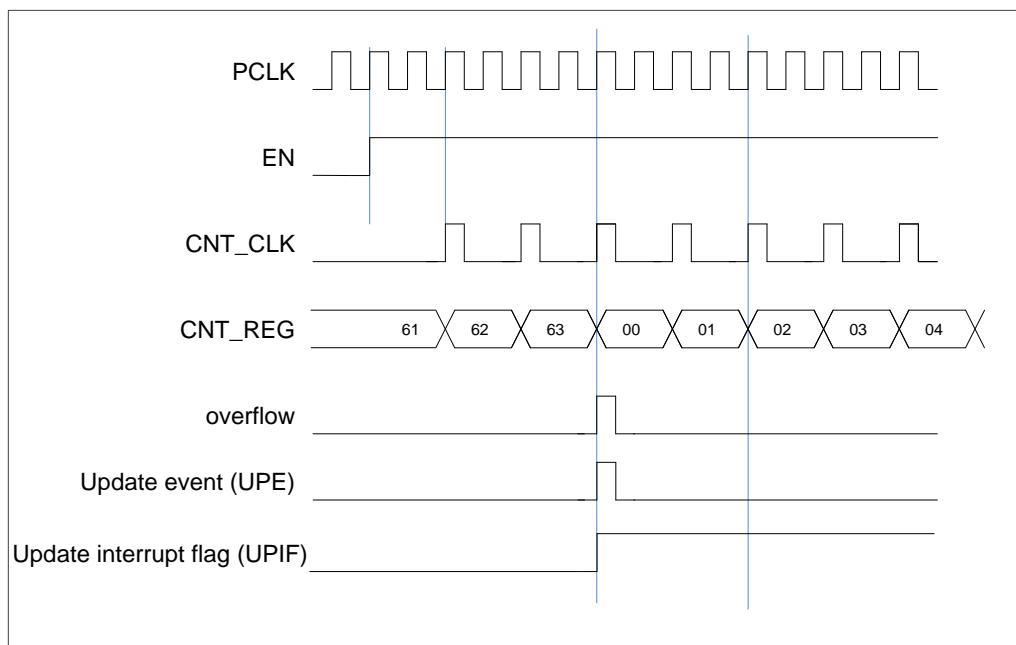


图 9-51. 计数器时序图，内部时钟分频系数为 4

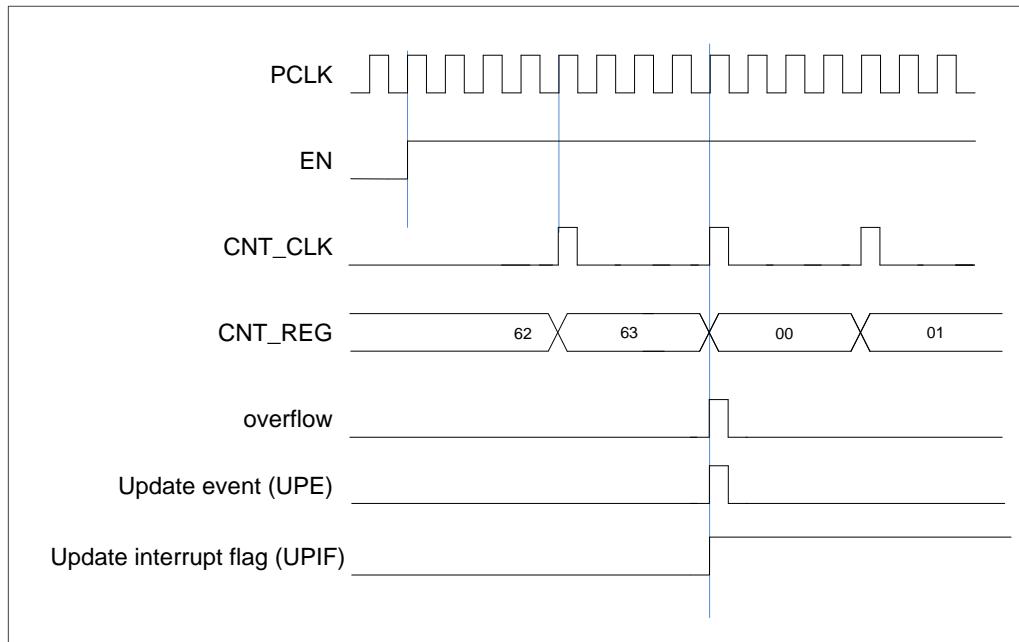


图 9-52. 计数器时序图，内部时钟分频系数为 N

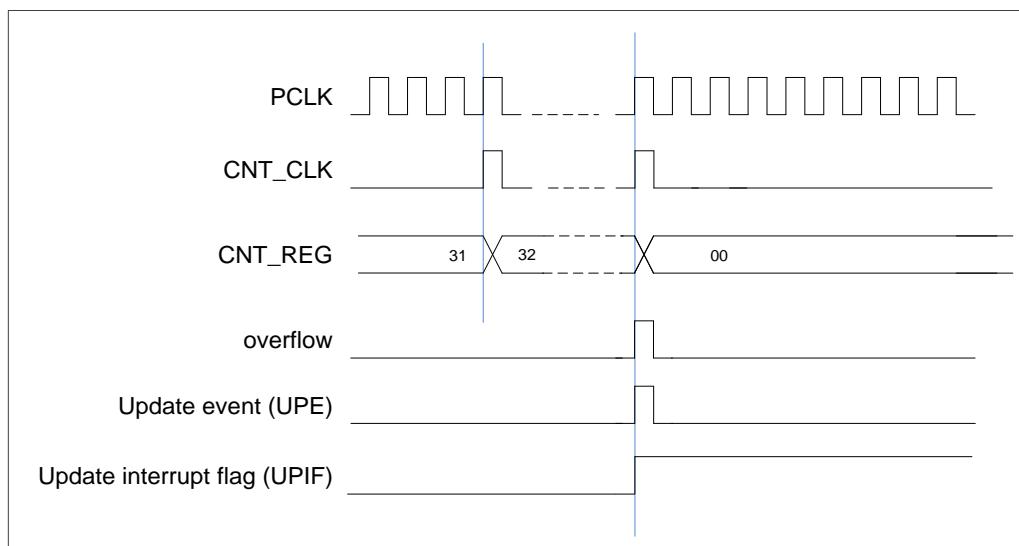


图 9-53. 计数器时序图, 当 ARSE=0 时的更新事件

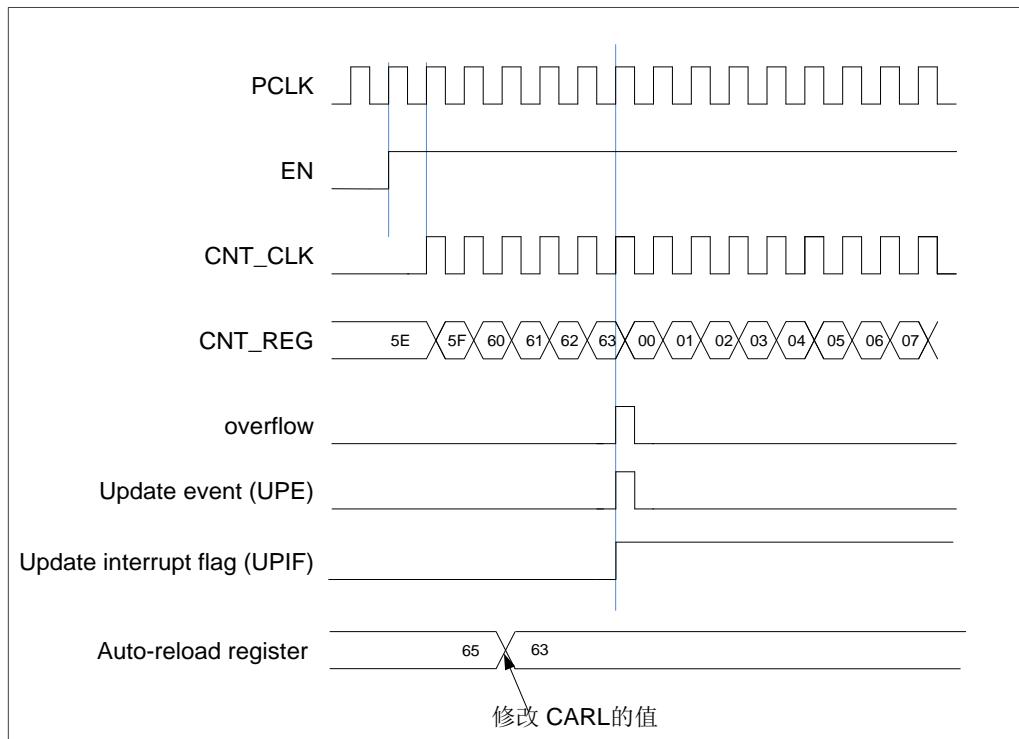
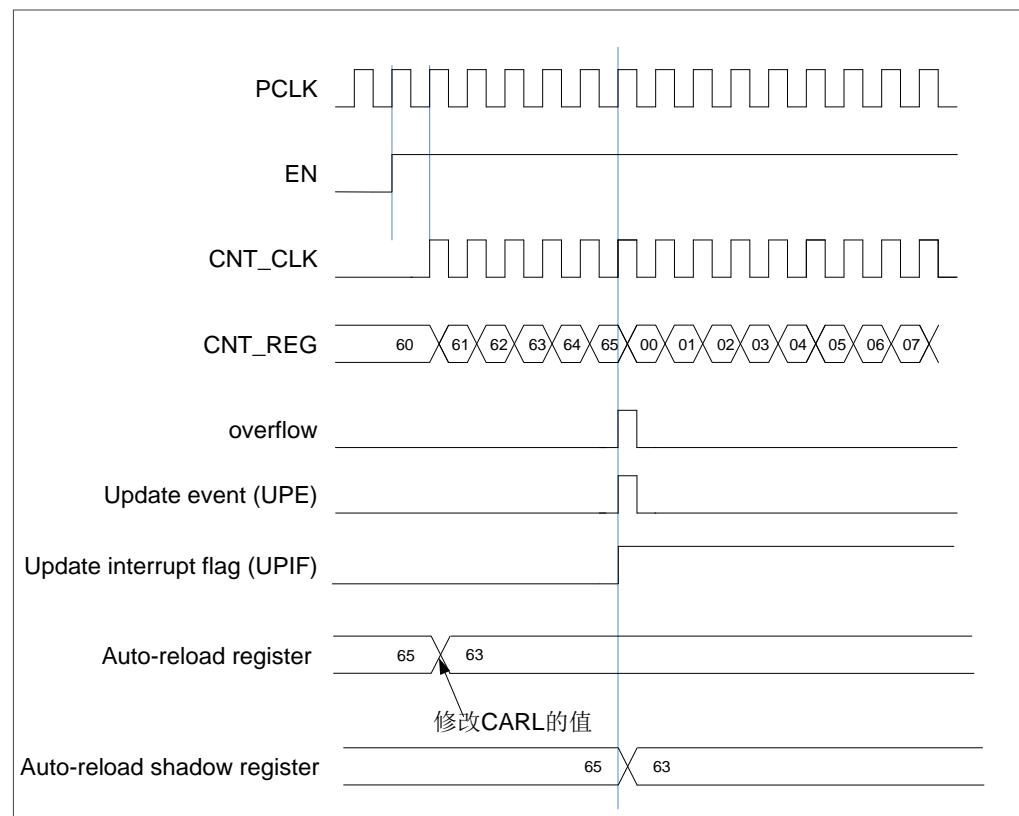


图 9-54. 计数器时序图, 当 ARSE=1 时的更新事件



## 向下计数模式

此模式下，计数器自动从计数重载值(TIMERx\_CARL寄存器的值)计数到0。一旦计数器计数到0，计数器重启，从计数器重载值开始计数。TIMERx\_CTLR1寄存器的计数方向控制位DIR在向下计数模式中应该被置1。

通过软件将TIMERx\_EVG寄存器的UPG位置1也可以产生更新事件，计数值将被初始化为计数器重载值然后产生更新事件。

将TIMERx\_CTLR1寄存器的UPDIS位置1将禁止更新事件。

当更新事件发生时，所有的寄存器(自动重载寄存器，预分频寄存器)都被更新。

下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作。

图 9-55. 计数器时序图，内部时钟分频系数为 1

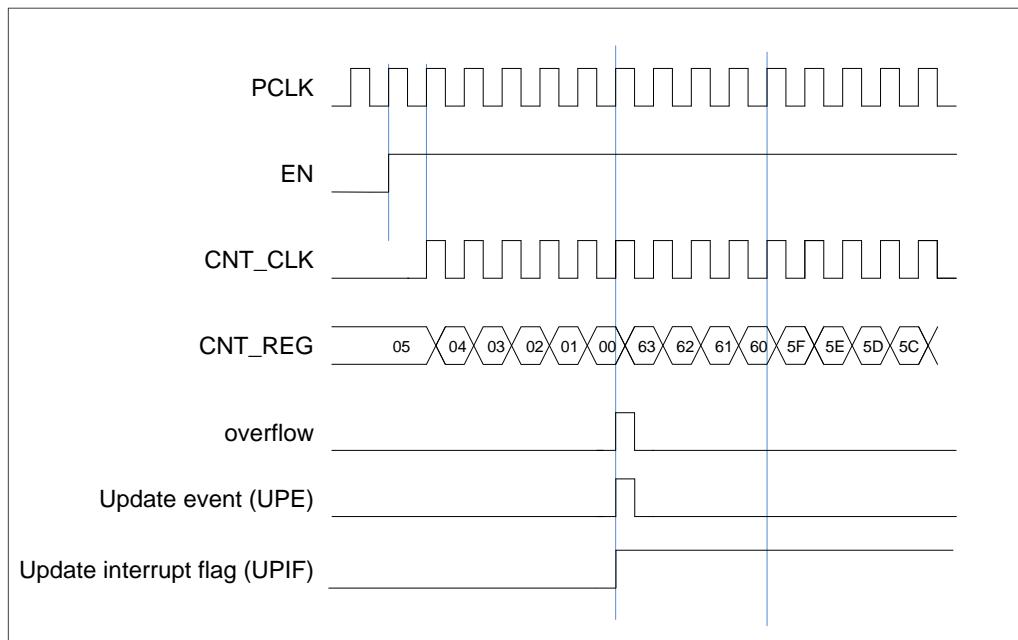


图 9-56. 计数器时序图，内部时钟分频系数为 2

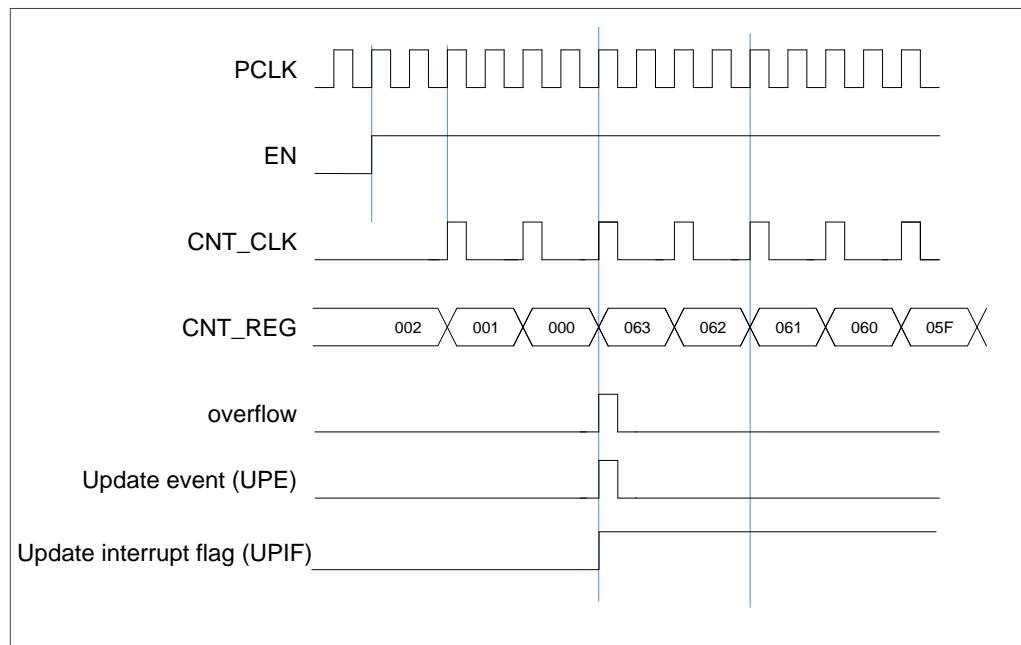


图 9-57. 计数器时序图，内部时钟分频系数为 4

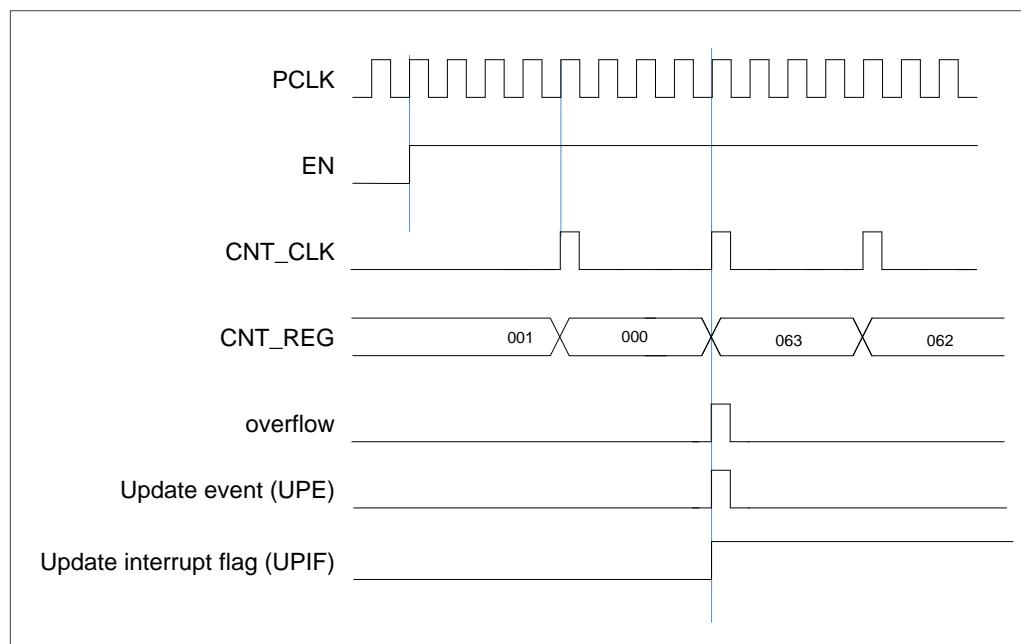


图 9-58. 计数器时序图，内部时钟分频系数为 N

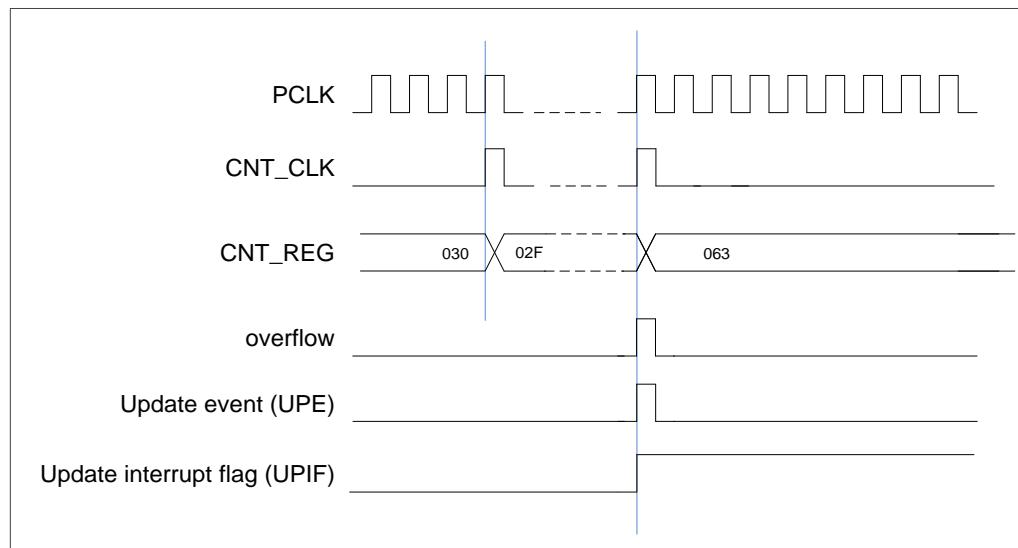
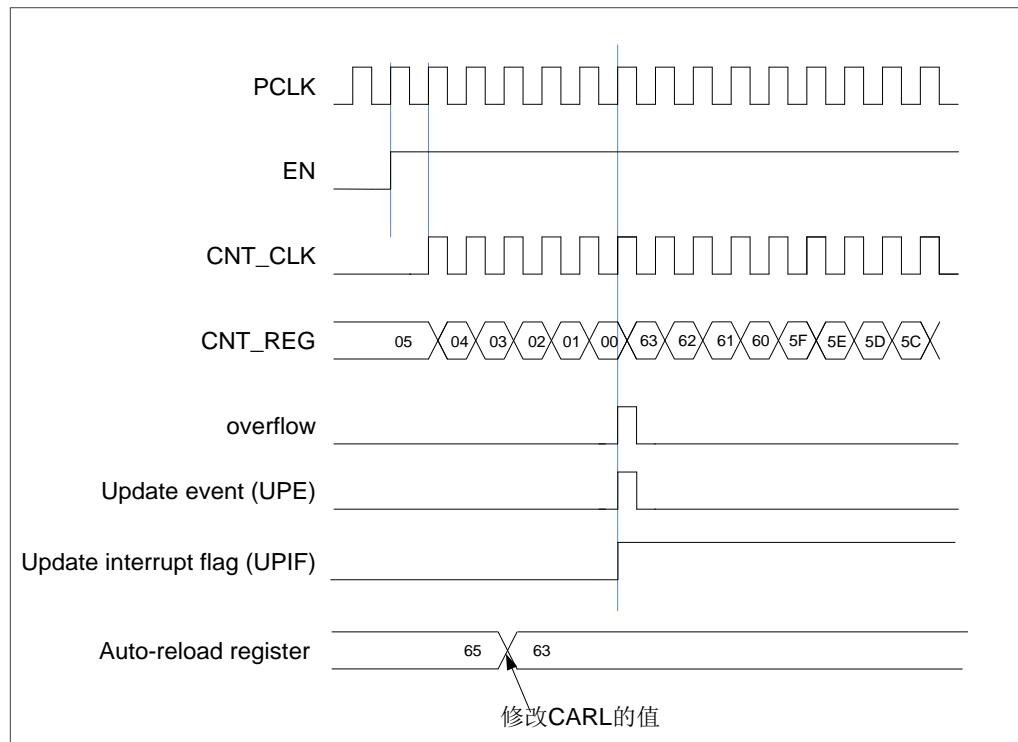


图 9-59. 计数器时序图，不使用计数器时的更新事件



### 中央对齐计数模式

在中央对齐模式，计数器从0开始计数到自动加载的值，然后向下计数到0。向上计数模式中，定时器模块在计数器计数到自动加载值-1产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。TIMERx\_CTRL1寄存器中的计数方向控制位DIR只读，且决定了中间对齐模式下的计数方向。计数方向被硬件自动更新。

将TIMERx\_EVG寄存器的UPG位置1可以初始化计数值为0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

当向下计数(TIMERx\_CTLR1寄存器的CAM为“01”)时发生下溢事件或向上计数时发生上溢事件(TIMERx\_CTLR1寄存器的CAM为“10”), 或者二者同时发生(TIMERx\_CTLR1寄存器的CAM为“11”), TIMERx\_EVG寄存器的UPIF位能被置1。

将TIMERx\_CTLR1寄存器的UPDIS位置1可以禁止更新事件。

一旦产生更新事件, 所有的寄存器(自动重载寄存器, 预分频寄存器)都将被更新。

下图给出一些例子, 当TIMERx\_CARL=0x5时计数器在不同时钟频率下计数器的动作。

图 9-60. 计数器时序图, 内部时钟分频系数为 1, TIMERx\_CARL = 0x5

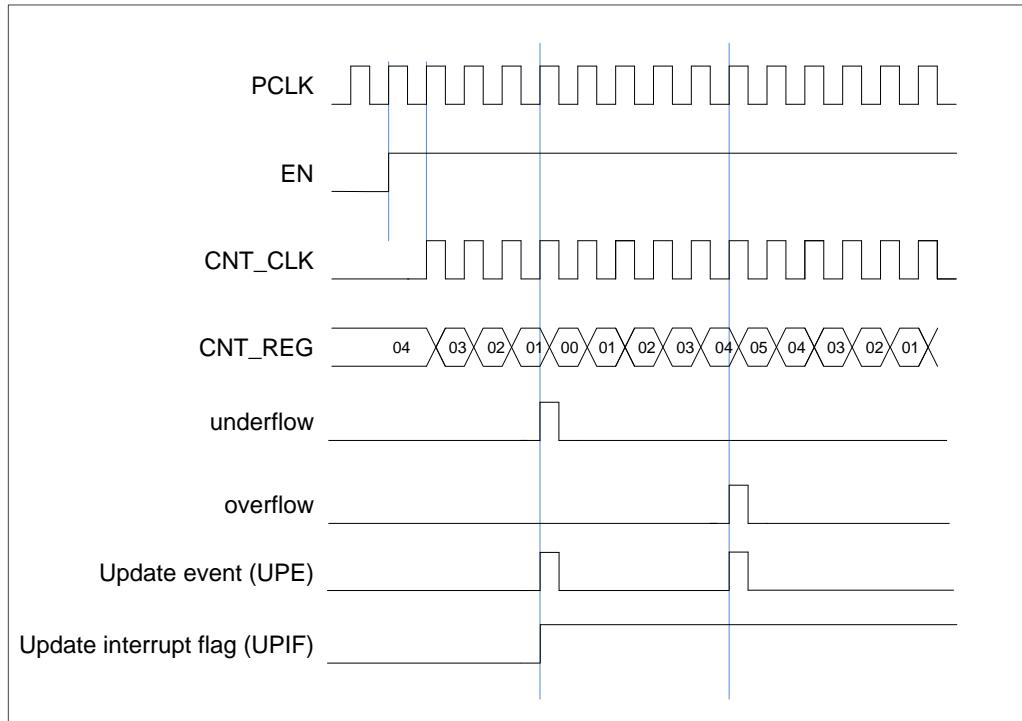


图 9-61. 计数器时序图，内部时钟分频系数为 2

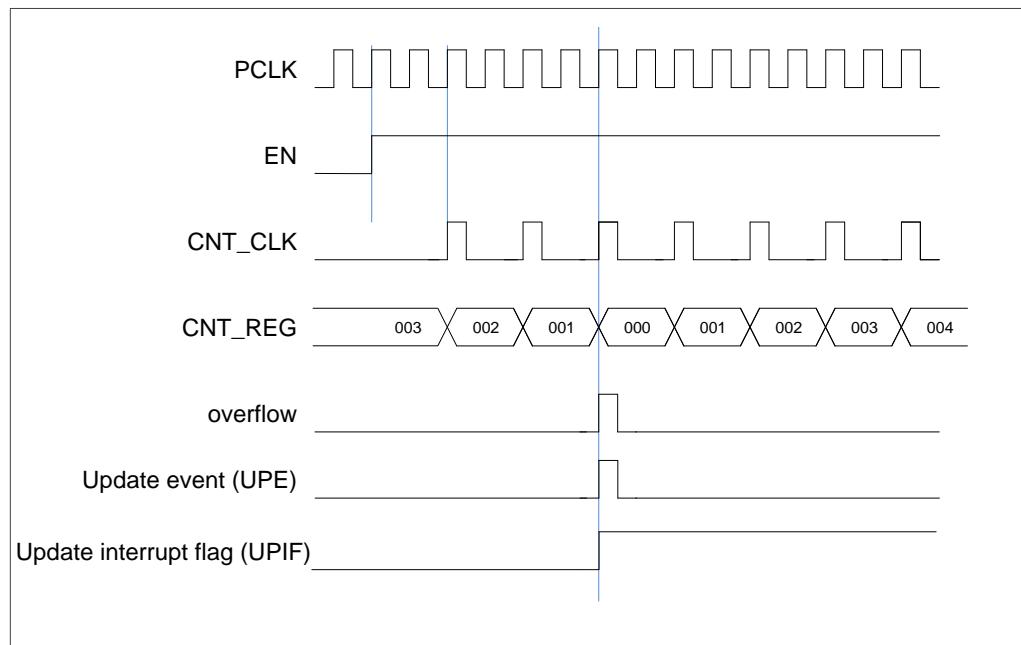


图 9-62. 计数器时序图，内部时钟分频系数为 4, TIMERx\_CARL=0x63

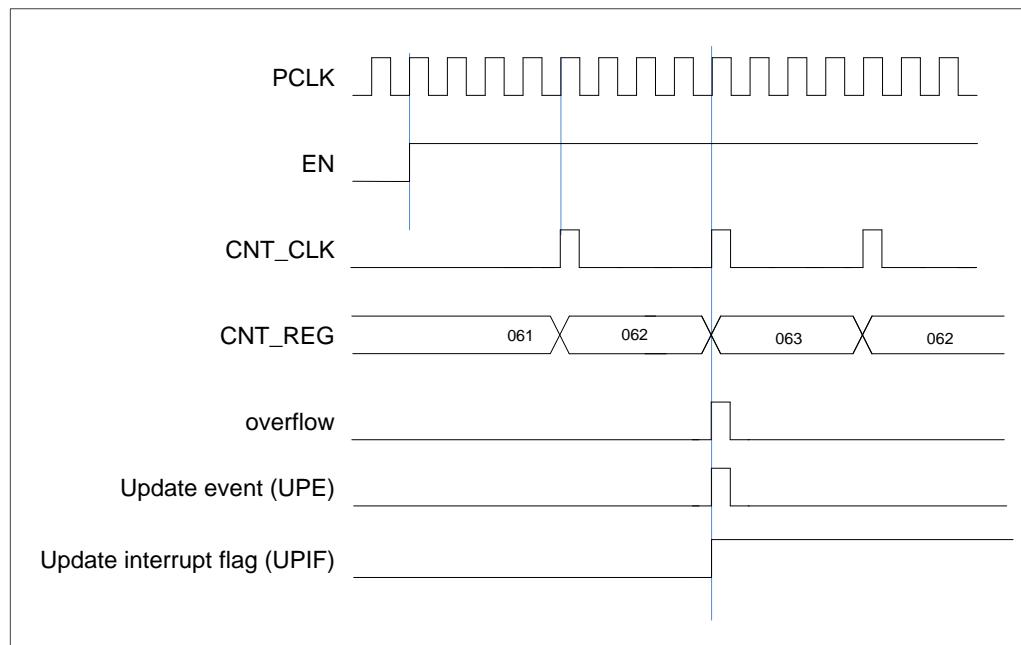


图 9-63. 计数器时序图，内部时钟分频系数为 N

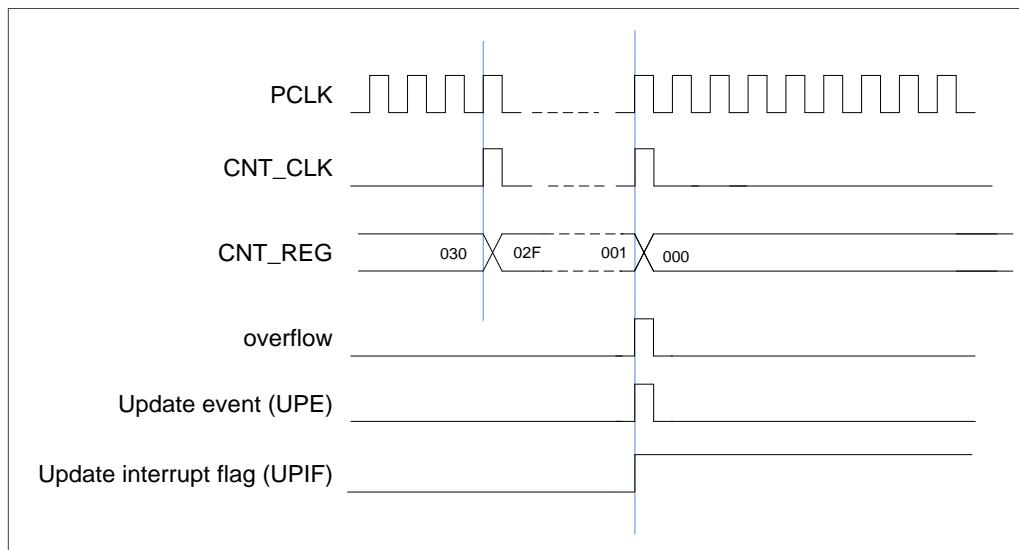


图 9-64. 计数器时序图, ARSE=1(计数器下溢) 时的更新事件

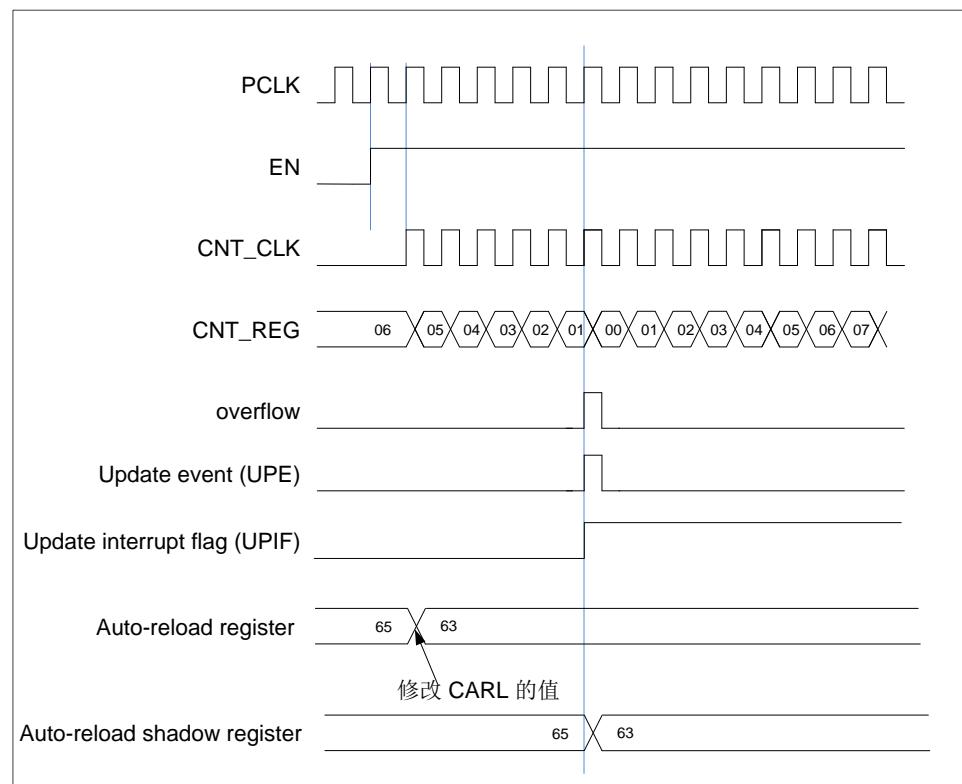
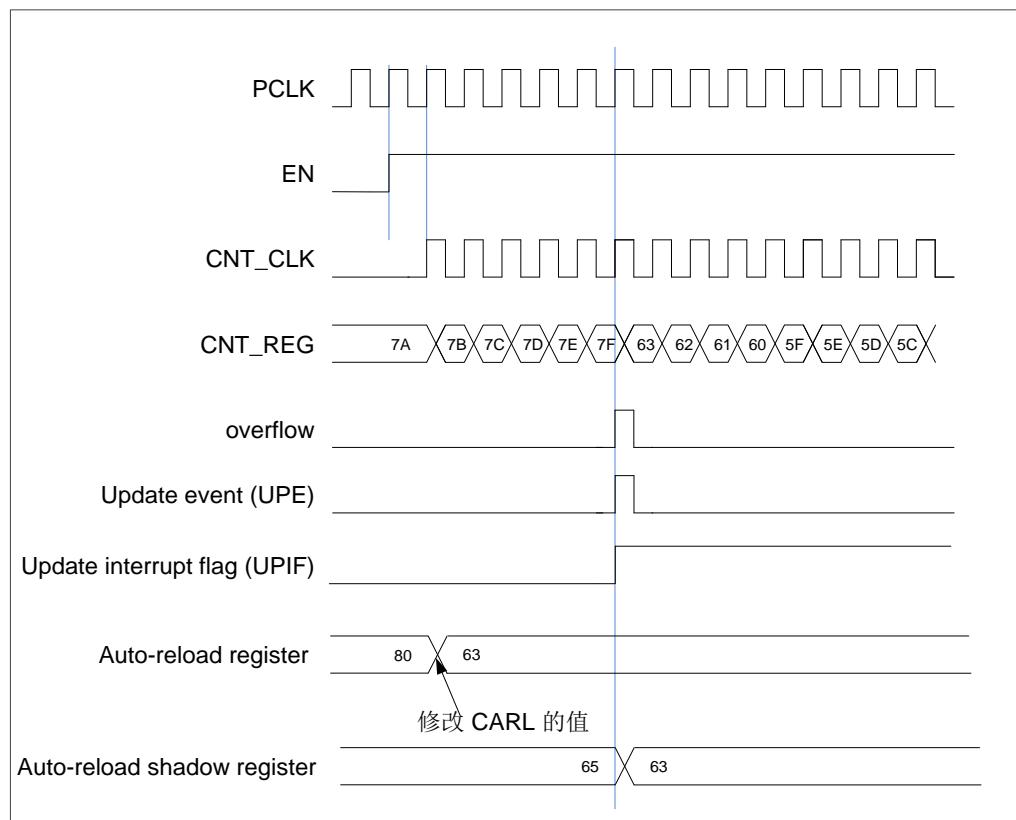


图 9-65. 计数器时序图, ARSE=1 (计数器上溢)时的更新事件



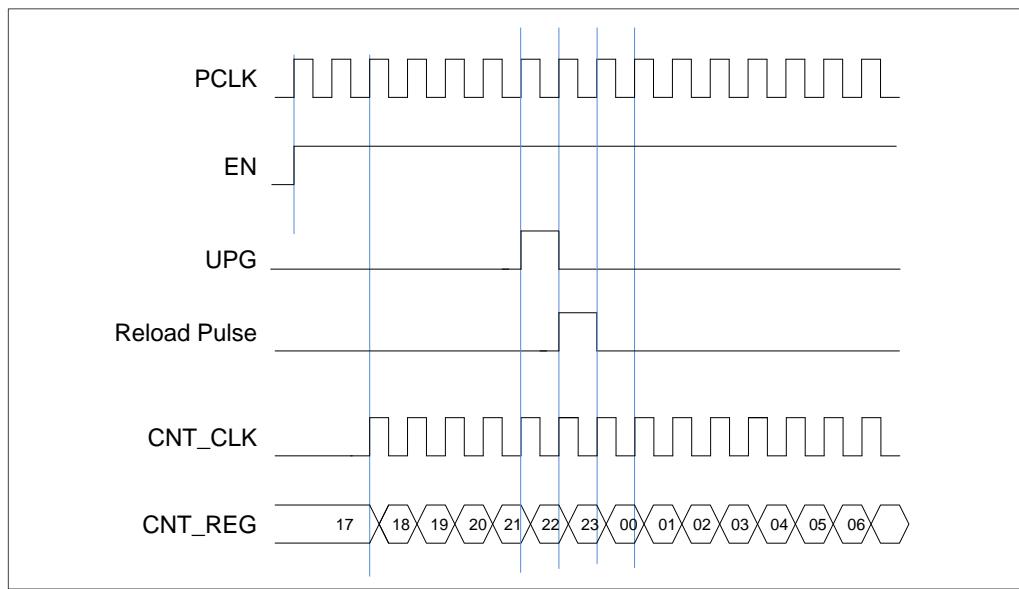
### 时钟选择

决定内部预分频计数器时钟源的定时器模块时钟控制器的描述如下。

- 内部定时器时钟PCLK

如果禁止了从模式，默认用来驱动计数器预分频器的内部时钟源是APB2时钟CK\_APB2。如果使能从模式控制器(将TIMERx\_SMC寄存器的SMC区域设置为包括0x1, 0x2, 0x3和0x7)，预分频器被其他时钟源(由TIMERx\_SMC寄存器的TRGS区域选择，说明如下)驱动。当从模式选择位SMC被设置为0x4, 0x5和0x6，计数器预分频器时钟源由内部时钟PCLK提供。

图 9-66. 内部时钟分频为 1 时正常模式下的控制电路



#### ■ 正交译码器

**TIMERx\_SMC** 寄存器的 **SMC** 区域应该被设置为 0x1, 0x2 或 0x3 来选择正交译码器模式。正交译码器功能使用 **TIMERx\_CH1** 和 **TIMERx\_CH2** 引脚的两个输入状态产生时钟脉冲驱动计数器预分频器。内部时钟源信号的每次转变时计数方向控制位 **DIR** 也被硬件自动改变。输入信号源可以只来源于 **TIMERx\_CH1** 引脚，只来源于 **TIMERx\_CH2** 引脚或者 **TIMERx\_CH1** 和 **TIMERx\_CH2** 引脚。

#### ■ 内部触发输入 (ITI)

计数器预分频器能在 **ITI** 信号的每个上升沿和下降沿计数。此模式可以通过设置 **TIMERx\_SMC** 寄存器的 **SMC** 区域为 0x6 来选择，此时计数器将作为一个事件计数器。输入事件 **ITI** 能通过设置 **TRGS** 区域选择。当 **ITI** 信号被选为时钟源，内部边沿监测电路将在每个 **ITI** 信号上升沿或下降沿产生一个时钟脉冲来驱动计数器预分频器。

#### ■ 外部输入引脚(TIx)

外部引脚 **TIMERx\_TIx** 信号的上升沿和下降沿能驱动计数器预分频器计数。此模式可以通过设置 **SMC** 区域为 0x7 及设置 **TRGS** 区域为 0x4, 0x5 或 0x6 来选择。注意 **TIx** 源于数字滤波器对 **TIMERx\_TIx** 采样。

#### ■ 外部输入触发(ETIF)

外部引脚 **TIMERx\_ETI** 信号的每个上升沿或下降沿能驱动计数器预分频器计数。此模式可以通过设置 **TIMERx\_SMC** 寄存器的 **ECM2E** 位为 1 来选择。其他选择 **ETIF** 信号作为时钟源的方式为设置 **SMC** 区域为 0x6 和设置 **TRGS** 区域为 0X7。注意 **ETIF** 信号源于数字滤波器对 **TIMERx\_ETI** 引脚信号的采样。如果选择 **ETIF** 信号为时钟源，触发控制器包括边沿监测电路将在每个 **ETIF** 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

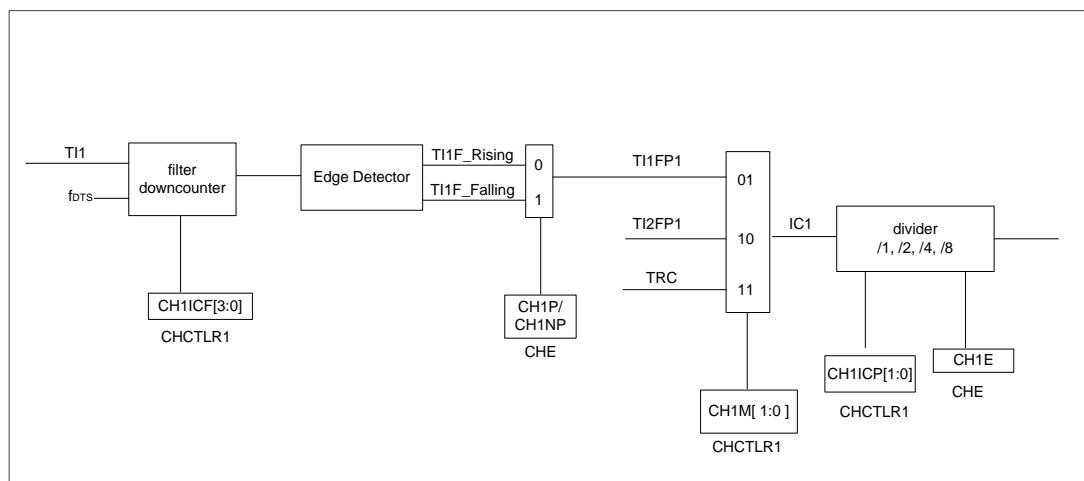
## 捕获/比较通道

TIMER2/3拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级

### ■ 输入捕获级

输入级包括一个数字滤波器，一个通道极性选择，边沿监测和一个通道预分频器。通道0的输入信号(TI1)可以被选择来自 TIMERx\_CH1 信号或者 TIMERx\_CH1， TIMERx\_CH2 和 TIMERx\_CH3信号的异或信号。通道输入信号(TIx)被一个数字滤波器采样来产生一个滤波输入信号TIxF。然后通道极性和边沿监测模块产生一个TIxFP1或TIxFP2信号，来实现输入捕捉功能。通道输入预分频寄存器(CHxICP)可以设置有效输入事件数目。

图 9-67. 捕获/比较通道(例如：通道 1 输入级)



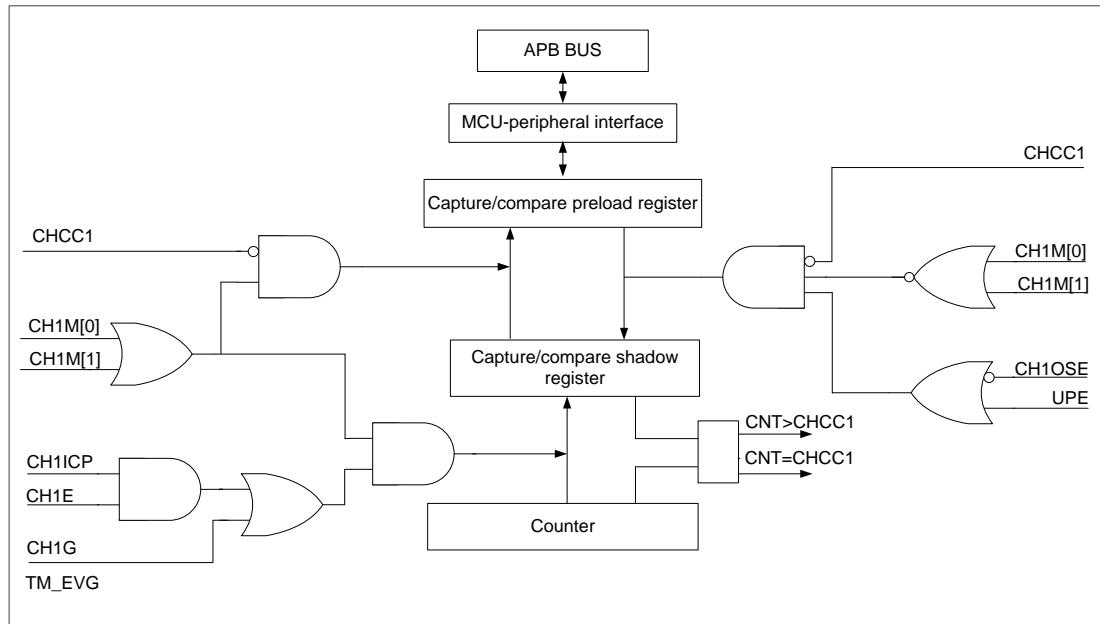
### ■ 通道控制器

TIMER2/3有4个独立的通道用来捕获输入或输出匹配比较。

用于输入捕获模式下，当捕获事件发生时，TIMERx\_CHCCx 影子寄存器首先捕获计数值然后传送到TIMERx\_CHCCx 预装载寄存器中。

用于输出匹配比较模式下，TIMERx\_CHCCx预装载寄存器的值被复制到相应的影子寄存器中，然后计数值和影子寄存器的值比较

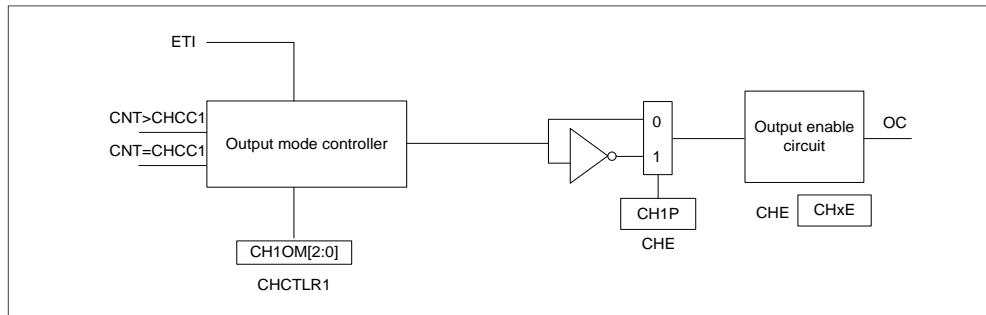
图 9-68. 捕获/比较通道 1 主电路



### ■ 输出级

TIMER2/3 有四个用于匹配比较、单脉冲和 PWM 输出功能的通道。

图 9-69. 捕获/比较通道输出级 (通道 1)



### 输入捕获模式

通道用于输入捕获时，当一个有效输入信号的边沿跳变时，计数值被捕获到捕获/比较寄存器 (TIMERx\_CHCCx) 中。一旦捕获事件发生，TIMERx\_STR 寄存器的 CHxIF 标志位被置 1。如果 CHxIF 位已经被置 1，还没来得及被软件清除，此通道上又有另一个捕获事件发生，相应通道捕获溢出标志位 CHxOF 将被置 1。一旦捕获事件发生，CHxDE 位决定了是否产生 DMA 请求，CHxIE 位决定了是否产生中断。使用输入捕获模式需按以下步骤配置：

8. 配置通道控制寄存器 (TIMERx\_CHCTRLRx) 的 CCxM 位域，选择输入信号
9. 配置通道控制寄存器 (TIMERx\_CHCTRLRx) 的 CHxICF 位域，为输入信号添加输入滤波器
10. 配置通道使能寄存器 (TIMERx\_CHE) 的 CHxP/CHxNP 位，选择输入捕获的边沿（上升沿或下降沿）

11. 如果输入信号需要分频，配置通道控制寄存器(TIMERx\_CHCTRLx)的CHxCP位域
12. 如果需要中断，将DMA中断使能寄存器(TIMERx\_DIE)的CHxIE位置位
13. 如果需要DMA请求，将DMA中断使能寄存器(TIMERx\_DIE)的CHxDE位置位
14. 将通道使能寄存器(TIMERx\_CHE)的CCxE位置位，使能通道

按上述配置完寄存器后，该通道上的对应边沿信号将产生输入捕获事件。计数值被捕获到通道捕获/比较寄存器(TIMERx\_CHCCx)中。如果CHxIE位被置位，将会产生中断。如果CHxDE位被置位，将会产生DMA请求。

输入捕获模式也可以用于TIMERx\_CHx(TIx)引脚的信号脉宽测量。例如，PWM信号连接到TI1输入。配置TIMERx\_CHCTRLx寄存器的CH1M位域为01，将通道1输入捕获信号连接到TI1，并设置上升沿捕获。配置TIMERx\_CHCTRLx寄存器的CH2M位域为10，选择通道2捕获信号连接到TI1，并设置下降沿捕获。计数器配置为通道1上升沿触发后重新计数模式。此时TIMERx\_CHCC1可以测量PWM信号的周期，TIMERx\_CHCC2可以测量PWM信号的占空比。

### 输出比较模式/PWM 模式

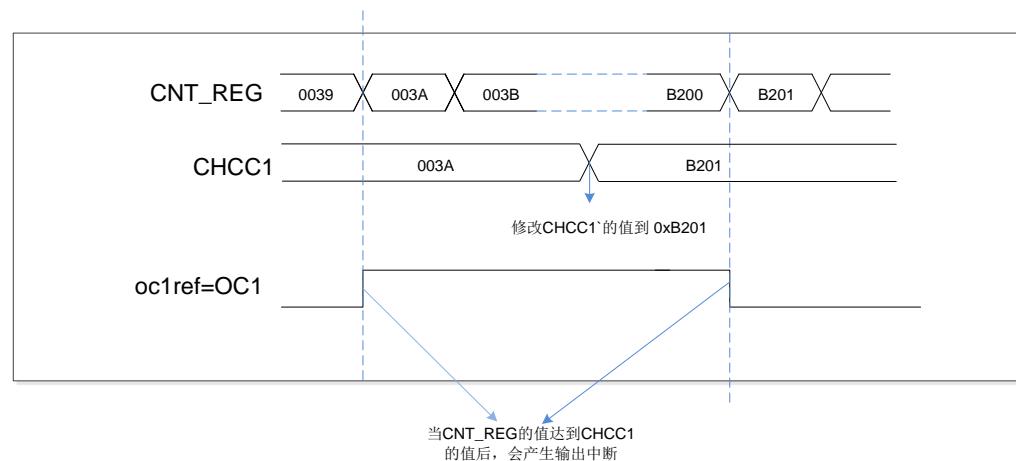
当配置通道控制寄存器(TIMERx\_CHCTRLx)的CCxM位域为00，使通道工作于输出模式时，通道输出的波形由TIMERx\_CHCTRLx的CHxOM位域和计数器与TIMERx\_CHCCx寄存器的值比较的结果决定。当比较的值相等时状态寄存器(TIMERx\_STR)的CHxIF位将置1。根据CHxDE或CHxIE位的值，一次DMA请求或中断将被产生。

在输出模式，配置CHxOM位域为101，通道将持续输出有效电平。配置CHxOM位域为100通道将持续输出无效电平。

在输出模式，配置CHxOM位域为001，当计数器与TIMERx\_CHCCx寄存器的值相等时，通道输出有效电平。配置CHxOM位域为010，当计数器与TIMERx\_CHCCx寄存器的值相等时，通道输出无效电平。

在输出比较翻转模式（配置CHxOM位域为011），当计数器与TIMERx\_CHCCx寄存器的值相等时，通道输出电平将翻转。

**图 9-70. 输出比较模式，在 OC1 上切换**



在PWM输出模式（配置CHxOM位域为110（PWM模式1）或111（PWM模式2））。根据

TIMERx\_CARL寄存器和TIMERx\_CHCCx寄存器的值，通道输出PWM波形。

图 9-71. 在 OC1 上输出比较 PWM 模式 1，向上计数模式

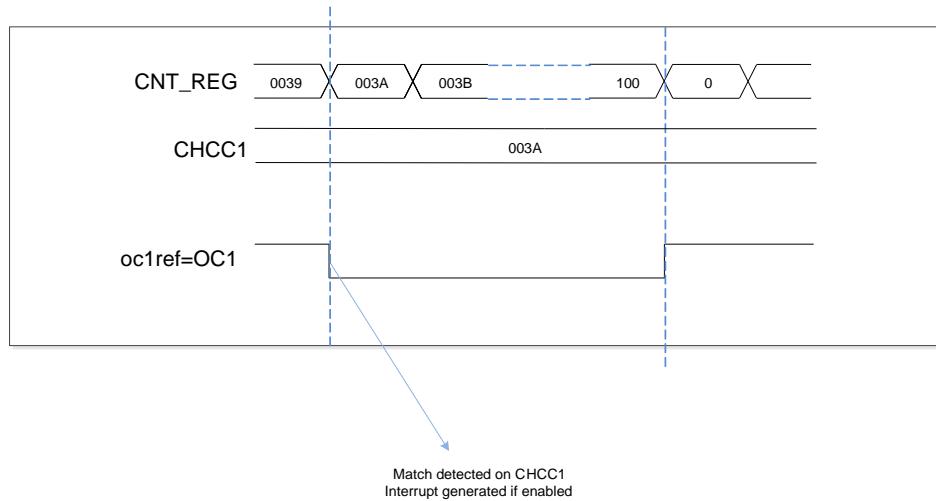
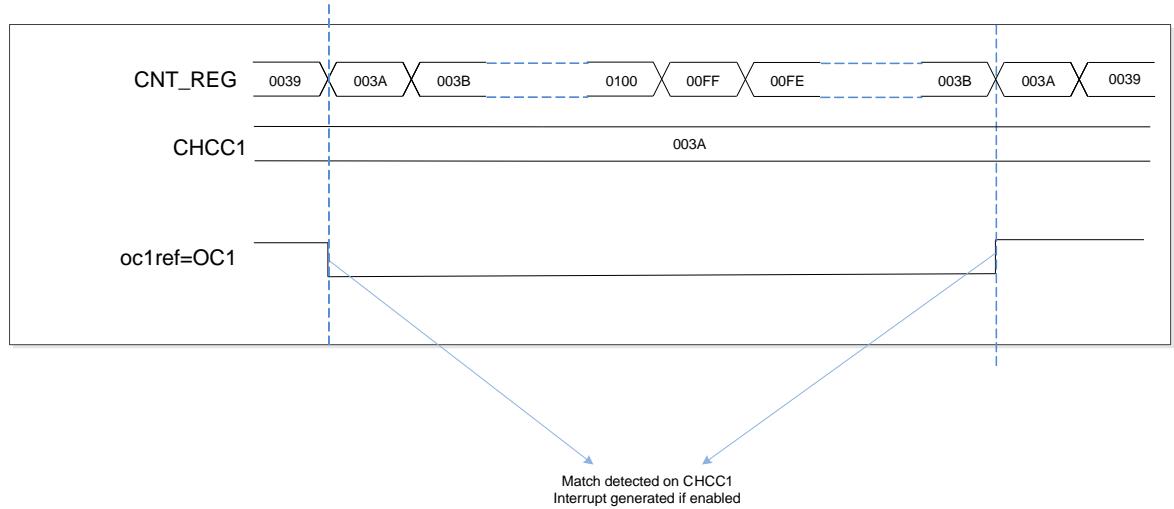


图 9-72. 在 OC1 上输出比较 PWM 模式 1，中央对齐计数模式



### 通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxOM位可以定义OCxREF信号(通道x输出参考信号)类型。OCxREF信号有一些输出功能类型，包括，设置CHxOM=0x00可以保持原始电平，设置CHxOM=0x01可以将OCxREF信号设置为高电平，设置CHxOM=0x02可以将OCxREF信号设置为低电平，在计数值和TIMERx\_CHCCx寄存器的值匹配时，设置CHxOM=0x03可以切换信号。

PWM模式1和PWM模式2输出也是另一种类型的OCxREF输出，具体设置是通过设置CHxOM=0x06或0x07来完成的。在这些模式中，OCxREF信号电平的改变取决于根据计数方向，以及TIMERx\_CHCCx寄存器值和计数值之间的关系。详情参考相应位定义。

设置CHxOM=0x04或0x05可以实现OCxREF信号的强制输出功能。输出比较信号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。

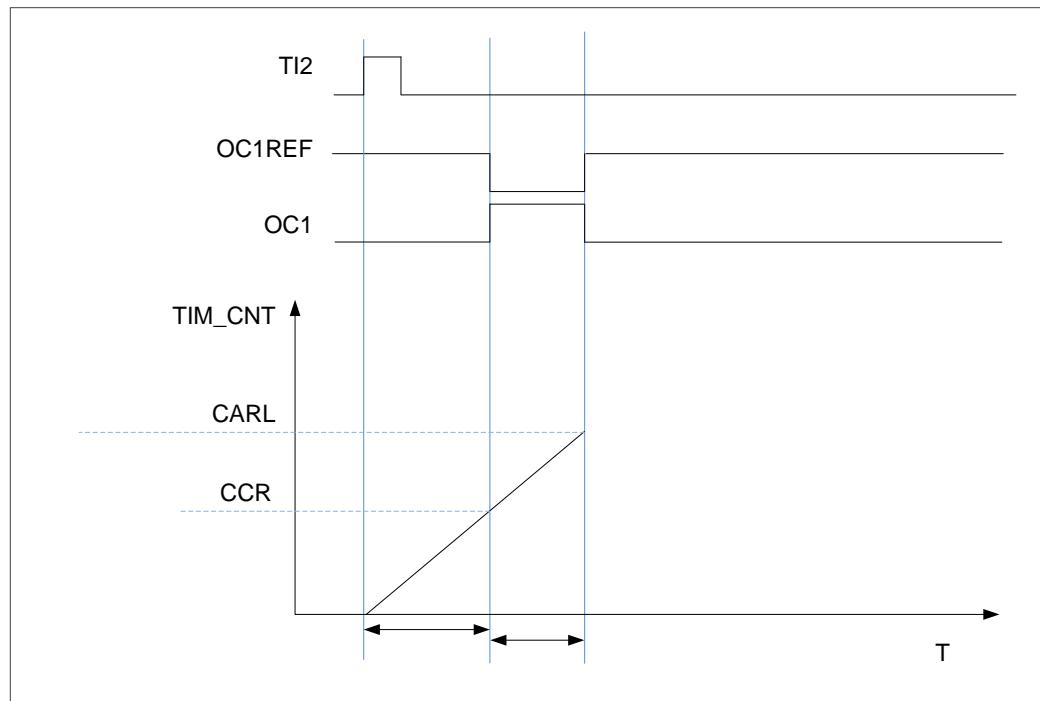
当ETIF信号源于外部TIMERx\_ETI引脚且被设置为高电平(设置TIMERx\_CHCTRL1寄存器的CHxOCE位为1)时，OCxREF信号被强制变为低电平。直到下一次更新事件发生，OCxREF信号才会回到它的有效状态。

### 单脉冲模式

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx\_CTRL1寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

单脉冲模式下，将CEN位置1产生的触发有效边沿将使能计数器。然而，执行计数值和TIMERx\_CHCCx寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx\_CHCTRL1寄存器的CHxOEF位置1。单脉冲模式下，触发上升沿产生之后，OCxREF信号将被立即强制转换为另一种状态，因为不考虑比较结果的比较匹配事件发生。只有输出通道配置为PWM1或PWM2输出运行模式下时 CHxOEF位才可用，触发源来源于触发信号。

图 9-73. 单脉冲模式



### 正交译码器

正交译码器功能使用两个源于TIMER1\_CH1和TIMER1\_CH2引脚的积分输入TI1和TI2各自相互作用产生计数值。在每个输入源改变期间，DIR位被硬件自动改变。输入源可以是只有TI1，只有TI2或TI1和TI2。选择模式可以设置SMC=0x01, 0x02或0x03来完成。改变计数方向的机制如下图所示。正交译码器可以用作一个带有方向选择的外部时钟，意味着计数器在内部从0到重载值持续计数。因此，用户必须在计数器开始计数前配置TIMER1\_CARL寄存器。

表 9-2. 计数方向对编码器信号

计数模式	电平	TI1FP1		TI2FP2	
		向上计数	向下计数	向上计数	向下计数
只有 TI1	TI1FP=1	向下	向上	-	-
	TI1FP=0	向上	向下	-	-
只有 TI2	TI1FP=1	-	-	向上	向下
	TI1FP=0	-	-	向下	向上
TI1 和 TI2	TI2FP=1	向下	向上	X	X
	TI2FP=0	向上	向下	X	X
	TI1FP=1	X	X	向上	向下
	TI1FP=0	X	X	向下	向上

注意: “-”意思是“无计数”; “X”意思是不可能

图 9-74. 编码器接口模式下计数器运行实例

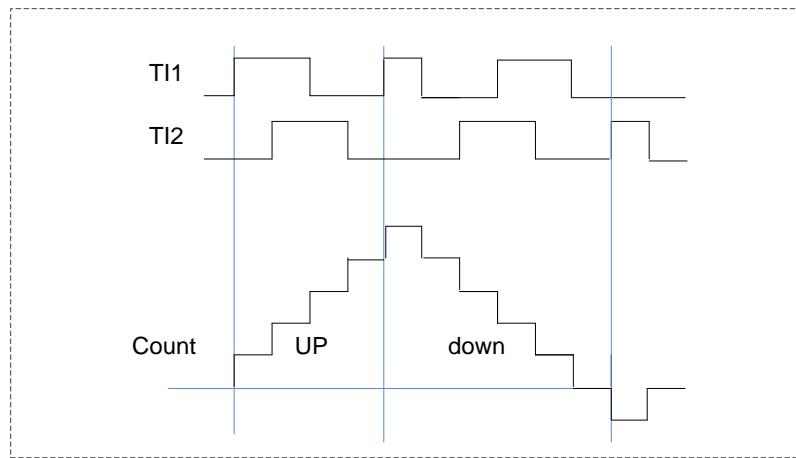
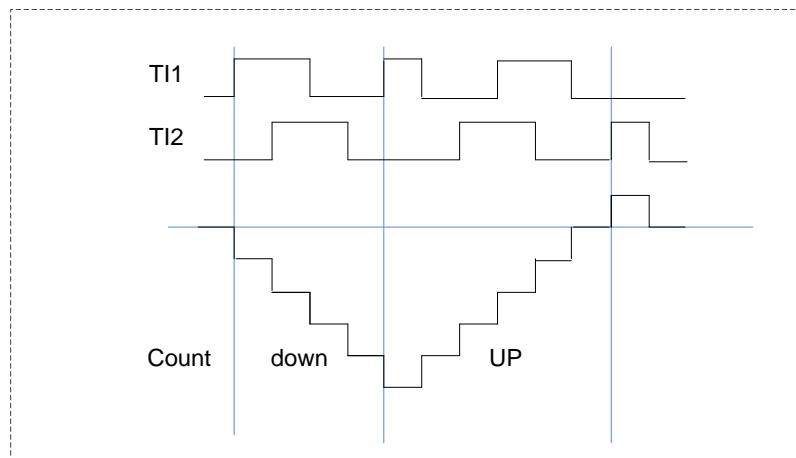


图 9-75. 带有 TI1FP1 极性转换的编码器接口模式下的例子



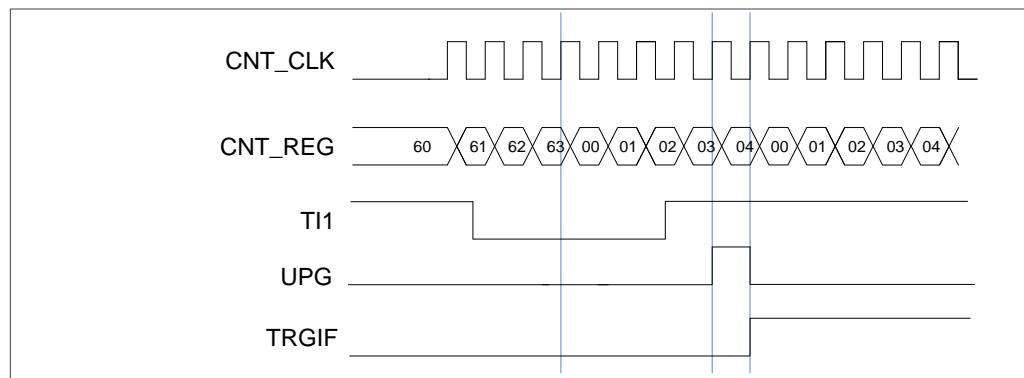
## 从控制器

TIMERx能够在多种模式下和一个外部的触发同步，包括复位模式，暂停模式和触发模式。模式选择通过设置TIMERx\_SMC寄存器的SMC区域来完成。这些模式的触发输入可以通过TIMERx\_SMC寄存器的TRGS区域来选择，下面以TI1信号为例。从控制器运行模式参见附加章节。

### ■ 复位模式

在TI1信号的上升沿来临时，计数器和它的预分频器能够重新被初始化，此时更新事件的软件产生位UPG将自动被硬件声明，触发事件标志位被置1，然后计数器和它的预分频器被重新初始化。尽管UPG位被硬件置1，更新事件并不真的发生，它依赖于更新事件禁止控制位UPDIS被置1与否。如果UPDIS被置1，禁止更新事件发生，将不产生更新事件，然而当TI1上升沿来临时计数器和它的预分频器仍然被重新初始化。如果TIMERx\_CTRL1寄存器的UPDIS位被清0，使能更新事件发生，当TI1上升沿来临时更新事件将发生。所有的预加载寄存器将被更新。

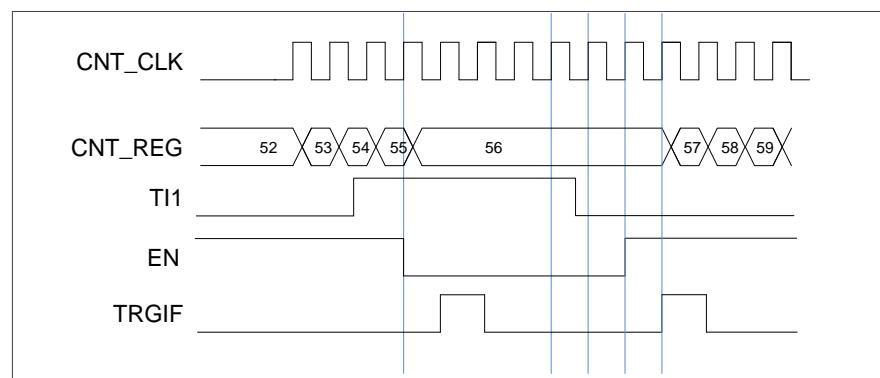
图9-76. 复位模式的控制电路



### ■ 暂停模式

暂停模式下，选择的TI1输入信号电平用来控制计数器开启/停止运行。当选择的TI1信号为高电平时计数器开始计数，当TI1信号变为低电平时停止计数。此处计数器将保持它的当前值且不再被重启。

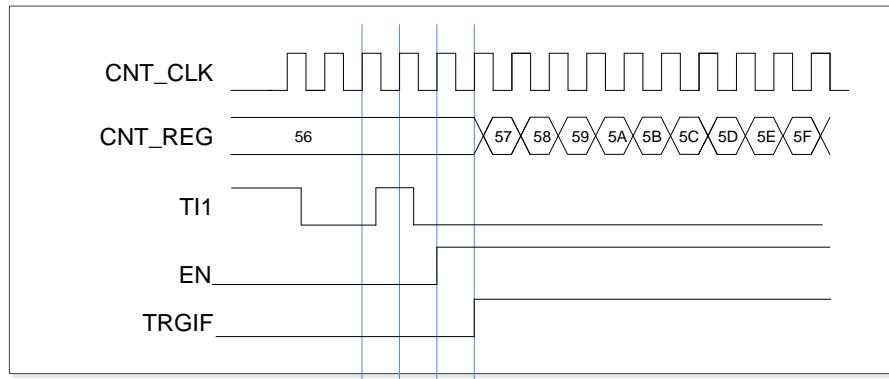
图 9-77. 暂停模式下的控制电路



### ■ 触发模式

计数器被禁止计数之后，当TI1上升沿来临时计数器可以重新开始计数。当一个TI1上升沿来临，计数器从当前值开始计数。注意TI1信号仅用来使能计数器重新计数，不能控制计数器停止计数。

**图 9-78. 触发模式下的控制电路**

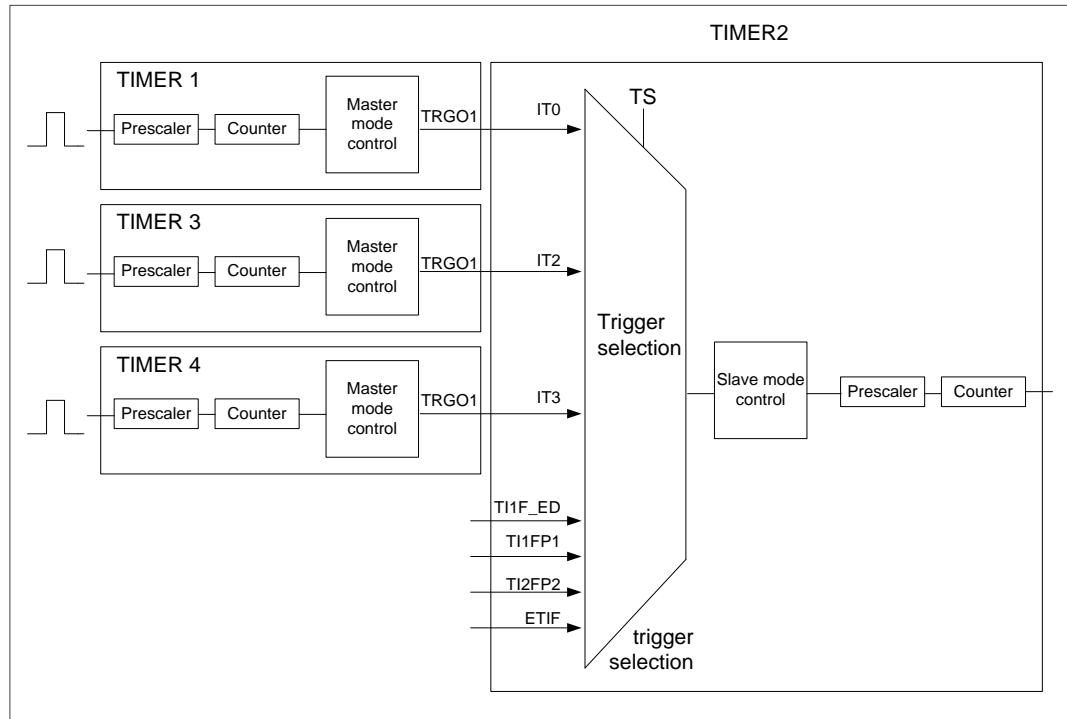


### 定时器互连

定时器互相之间可以级联或者同步。可以通过配置一个定时器工作在主模式另一个定时器工作在从模式来实现。下图显示了一些主从模式触发选择的例子。

下图显示了当定时器 x 配置为从模式时的触发选择。

**图 9-79. 定时器 2 主/从模式的例子**



### 定时器 debug 模式

当微控制器在调试模式，若Cortex™-M3内核停止，且MCUDBG模块中DBG\_TIMERx\_STOP

---

配置位被置1， TIMERx计数器停止。

### 9.2.4. TIMER2 和 TIMER3 寄存器

#### TIMERx 控制寄存器 1 (TIMERx\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN		

rw            rw            rw            rw            rw            rw            rw            rw            rw

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CDIV[1:0]	<p>时钟分频</p> <p>软件配置CDIV位规定定时器时钟(PCLK) 和死区时间、采样时钟(DTS)之间的分频系数，该位被死区时间产生器和数字滤波器使用</p> <p>00: <math>f_{DTS}=f_{PCLK}</math></p> <p>01: <math>f_{DTS}=f_{PCLK}/2</math></p> <p>10: <math>f_{DTS}=f_{PCLK}/4</math></p> <p>11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁止TIMERx_CARL寄存器的影子寄存器</p> <p>1: 使能TIMERx_CARL寄存器的影子寄存器</p>
6:5	CAM[1:0]	<p>中央对齐模式选择</p> <p>00: 无中央对齐模式(边沿对齐模式)。 DIR位指定了计数方向</p> <p>01: 中央对齐模式1。计数器交替地向上和向下计数。通道的输出比较中断标志位配置为输出模式 (TIMERx_CHCTRLRx寄存器中CHxM=00)，只有在计数器向下计数时被置1。</p> <p>10: 中央对齐模式2。计数器交替地向上和向下计数。通道的输出比较中断标志位配置为输出模式 (TIMERx_CHCTRLRx寄存器中CHxM=00)，只有在计数器向上计数时被置1。</p> <p>11: 中央对齐模式3。计数器交替地向上和向下计数。通道的输出比较中断标志位配置为输出模式 (TIMERx_CHCTRLRx寄存器中CHxM=00)，只有在计数器同时向上和向下计数时被置1。</p> <p>计数器开启时，不允许从边沿模式转换到中央对齐模式。</p>
4	DIR	<p>方向</p> <p>0: 向上计数</p> <p>1: 向下计数</p> <p>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。</p>

---

3	SPM	单脉冲模式. 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
2	UPS	更新请求源 软件通过该位选择更新事件源 0: 使能后, 下述任一事件产生更新中断或DMA请求: - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生的更新 1: 使能后只有计数器溢出/ 下溢才产生更新中断或DMA请求。
1	UPDIS	禁止更新 软件通过该位允许/ 禁止更新事件的产生 0: 允许更新事件产生。当以下事件之一发生时, 更新事件产生, 具有缓存的寄存器被装入它们的预装载值: - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生一个更新事件 1: 禁止更新事件。带有缓存的寄存器保持原有值, 如果UPG位被置1或者从模式控制器产生一个硬件复位事件, 计数器和预分频器被重新初始化
0	CEN	使能计数器 0: 禁止计数器 1: 使能计数器 在软件将CEN 位后, 外部时钟、暂停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。

### **TIMERx 控制寄存器 2 (TIMERx\_CTLR2)**

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							TI1S		MMC[2:0]		DMAS		保留		

rw                    rw                    rw

---

位/位域	名称	描述
15:8	保留	必须保持复位值
7	TI1S	通道1触发输入选择 0: 选择TIMERx_CH1引脚输入作为通道1触发输入 1: TIMERx_CH1、TIMERx_CH2 和TIMERx_CH3 引脚经异或后连到TI1输入
6:4	MMC[2:0]	主模式控制

这些位控制TRGO信号的选择，主模式下发给从定时器用于同步功能。

000: 复位。TIMERx\_EVG寄存器的UPG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。

001: 使能。此模式可用于同时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TIMERx\_EN 被主模式控制器选为触发输出(TRGO)。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。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。

010: 更新。更新事件被主模式控制器选为触发输入。

011: 捕获/比较脉冲。在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲

100: 比较。OC1REF信号被用于作为触发输出

101: 比较。OC2REF信号被用于作为触发输出

110: 比较。OC3REF信号被用于作为触发输出

111: 比较。OC4REF信号被用于作为触发输出

3	DMAS	DMA请求源选择
		0: 当通道x事件发生，发送通道x的DMA请求 1: 当更新事件发生，发送通道x的DMA请求
2:1	保留	必须保持复位值

## **TIMERx 从模式控制寄存器 (TIMx\_SMC)**

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETPL	ECM2E	ETPSC[1:0]		ETFC[3:0]		MSM		TRGS[2:0]		OCRC.		SMC[2:0]			

位/位域	名称	描述
15	ETPL	外部触发极性 该位选择用ETI还是ETI的反相来作为触发操作 0: ETI不反相，高电平或上升沿有效 1: ETI被反相，低电平或下降沿有效
14	ECM2E	外部时钟模式2使能 在外部时钟模式2，计数器由ETIF信号上的任意有效边沿驱动 0: 禁止外部时钟模式2 1: 使能外部时钟模式2 将ECM2E位置1与选择外部时钟模式1并将TRGI连到ETIF(SMC=111和TRGS=111)具有相同功效。 下述从模式可以与外部时钟模式2同时使用：复位模式，暂停模式和触发模式；但是，

---

		这时TRGI不能是‘111’。
		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ETIF
13:12	ETPSC[1:0]	<p>外部触发预分频</p> <p>外部触发信号ETIP的频率最多是TIMxCLK 频率的1/4。当输入较快的外部时钟时，可以使用预分频降低ETIP的频率。</p> <ul style="list-style-type: none"> <li>00: 禁止预分频</li> <li>01: ETIP频率除以2</li> <li>10: ETIP频率除以4</li> <li>11: ETIP频率除以8</li> </ul>
11:8	ETFC[3:0]	<p>外部触发滤波控制</p> <p>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。这些位定义了对ETIP信号采样的频率和对ETIP 数字滤波的带宽。</p> <ul style="list-style-type: none"> <li>0000: 无滤波器. <math>f_{SAMP}=f_{DTS}</math>, N=1.</li> <li>0001: <math>f_{SAMP}=f_{PCLK}</math>, N=2.</li> <li>0010: <math>f_{SAMP}=f_{PCLK}</math>, N=4.</li> <li>0011: <math>f_{SAMP}=f_{PCLK}</math>, N=8.</li> <li>0100: <math>f_{SAMP}=f_{DTS}/2</math>, N=6.</li> <li>0101: <math>f_{SAMP}=f_{DTS}/2</math>, N=8.</li> <li>0110: <math>f_{SAMP}=f_{DTS}/4</math>, N=6.</li> <li>0111: <math>f_{SAMP}=f_{DTS}/4</math>, N=8.</li> <li>1000: <math>f_{SAMP}=f_{DTS}/8</math>, N=6.</li> <li>1001: <math>f_{SAMP}=f_{DTS}/8</math>, N=8.</li> <li>1010: <math>f_{SAMP}=f_{DTS}/16</math>, N=5.</li> <li>1011: <math>f_{SAMP}=f_{DTS}/16</math>, N=6.</li> <li>1100: <math>f_{SAMP}=f_{DTS}/16</math>, N=8.</li> <li>1101: <math>f_{SAMP}=f_{DTS}/32</math>, N=5.</li> <li>1110: <math>f_{SAMP}=f_{DTS}/32</math>, N=6.</li> <li>1111: <math>f_{SAMP}=f_{DTS}/32</math>, N=8.</li> </ul>
7	MSM	<p>主-从模式</p> <p>主-从模式下输入触发上的事件被延迟了，以允许当前定时器和它的从设备通过TRGO 完美同步。如果想要在单个外部事件上同步一些定时器，此模式非常有用。</p> <ul style="list-style-type: none"> <li>0: 禁止主-从模式</li> <li>1: 使能主-从模式</li> </ul>
6:4	TRGS[2:0]	<p>触发选择</p> <p>这些位选择用于同步计数器的触发输入。</p> <ul style="list-style-type: none"> <li>000: 内部触发输入0 (ITR0)</li> <li>001: 内部触发输入1 (ITR1)</li> <li>010: 内部触发输入2 (ITR2)</li> <li>011: 内部触发输入3 (ITR3)</li> <li>100: TI1的边沿标志位 (TI1F_ED)</li> <li>101: 滤波后的通道1输入 (TI1FP1)</li> <li>110: 滤波后的通道2输入 (TI2FP2)</li> </ul>

111: 外部触发输入(ETIF)

从模式被使能后这些位不能改

从 TIMER	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
<b>TIMER2</b>	定时器 1	定时器 15	定时器 3	保留
<b>TIMER3</b>	定时器 1	定时器 2	定时器 15	保留

3 OCRC OCREF清零信号选择控制

0: OCREF\_CLR作为OCREF的清零信号

1: ETIF 作为OCREF的清零信号

2:0 SMC[2:0] 从模式控制

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。选中的触发输入的上升沿重新初始化计数器, 并且更新影子寄存器

101: 暂停模式。当触发输入为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止

110: 触发模式。计数器在触发输入TRGI 的上升沿启动。计数器不能被从模式控制器关闭。

111: 外部时钟模式1。选中的触发输入的上升沿驱动计数器

如果TI1F\_ED 被选为触发输入时, 不要使用暂停模式。这是因为, TI1F\_ED 在每次TI1F 变化时输出一个脉冲, 然而暂停模式是要检查触发输入的电平。

### TIMERx DMA 和中断使能寄存器 (TIMERx\_DIE)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDE	保留	CH4DE	CH3DE	CH2DE	CH1DE	UPDE	保留	TRGIE	保留	CH4IE	CH3IE	CH2IE	CH1IE	UPIE

rw                    rw

位/位域	名称	描述
15	保留	必须保持复位值
14	TRGDE	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	保留	必须保持复位值
12	CH4DE	通道4 DMA请求使能

		0: 禁止通道4 DMA请求 1: 使能通道4 DMA请求
11	CH3DE	通道3 DMA请求使能 0: 禁止通道3 DMA请求 1: 使能通道3 DMA请求
10	CH2DE	通道2 DMA请求使能 0: 禁止通道2 DMA请求 1: 使能通道2 DMA请求
9	CH1DE	通道1 DMA请求使能 0: 禁止通道1 DMA请求 1: 使能通道1 DMA请求
8	UPDE	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	保留	必须保持复位值
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	保留	必须保持复位值.
4	CH4IE	通道4中断使能 0: 禁止通道4中断 1: 使能通道4中断
3	CH3IE	通道3中断使能 0: 禁止通道3中断 1: 使能通道3中断
2	CH2IE	通道2中断使能 0: 禁止通道2中断 1: 使能通道2中断
1	CH1IE	通道1中断使能 0: 禁止通道1中断 1: 使能通道1中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### TIMERx DMA 和中断状态寄存器 (TIMERx\_STR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH4OF	CH3OF	CH2OF	CH1OF	保留	保留	TRGIF	保留	CH4IF	CH3IF	CH2IF	CH1IF	UPIF		
	rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0						

位/位域	名称	描述
15:13	保留	必须保持复位值
12	CH4OF	通道4重复捕获标志 参见CH1OF描述
11	CH3OF	通道3重复捕获标志 参见CH1OF描述
10	CH2OF	通道2重复捕获标志 参见CH1OF描述
9	CH1OF	通道1重复捕获标志 当通道1被配置为输入模式时，在CH1IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。 0: 无重复捕获中断发生 1: 发生了重复捕获中断
8:7	保留	必须保持复位值
6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置1。此位由软件清0。当从模式控制器处于除暂停模式外的其它模式时，在TRGI 输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，TRGI的任意边沿都可以产生触发事件。 0: 无触发事件产生 1: 触发中断产生
5	保留	必须保持复位值
4	CH4IF	通道4中断标志 参见CH1IF描述
3	CH3IF	通道3中断标志 参见CH1IF描述
2	CH2IF	通道2中断标志 参见CH1IF描述
1	CH1IF	通道1中断标志 此标志由硬件置1软件清0。当通道1在输入模式下时，捕获事件发生时此标志位被置1；当通道1在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道1中断发生

1: 通道1中断发生

0	<b>UPIF</b>	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断
---	-------------	--

### **TIMERx 事件产生寄存器 (TIMERx\_EVG)**

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				BKG	TRGG	保留	CH4G	CH3G	CH2G	CH1G	UPG				
				w	w		w	w	w	w	w				w

位/位域	名称	描述
15:8	保留	必须保持复位值
7	<b>BKG</b>	产生中止事件 该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POE位被清0且BKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0: 不产生中止事件 1: 产生中止事件
6	<b>TRGG</b>	触发事件产生 此位由软件置1，由硬件自动清0。当此位被置1，TIMERx_STR寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0: 无触发事件产生 1: 产生触发事件
5	保留	必须保持复位值
4	<b>CH4G</b>	通道4捕获或比较事件发生 参见CH1G描述
3	<b>CH3G</b>	通道3捕获或比较事件发生 参见CH1G描述
2	<b>CH2G</b>	通道2捕获或比较事件发生 参见CH1G描述
1	<b>CH1G</b>	通道1捕获或比较事件发生 该位由软件置1，用于在通道1产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CC1IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道1配置为输入模式，计数器的当前值被TIMERx_CHCC1寄存器捕

获，如果CH1IF标志位已经为1，则CH1OF标志位被置1。

0: 不产生通道1捕获或比较事件

1: 发生通道1捕获或比较事件

0	UPG	<p>此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。</p> <p>0: 无更新事件产生 1: 产生更新事件</p>
---	-----	---

### TIMERx 通道控制寄存器 1 (TIMERx\_CHCTRLR1)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2 OCE		CH2OM[2:0]		CH2 OSE	CH2 OFE		CH2M[1:0]	CH1 OCE		CH1OM[2:0]		CH1OSE	CH1 OFE		CH1M[1:0]
		CH2ICF[3:0]			CH2ICP[1:0]					CH1ICF[3:0]			CH1ICP[1:0]		

rw rw

输出比较模式:

位/位域	名称	描述
15	CH2OCE	通道2输出比较清0使能 参见CH1OCE描述
14:12	CH2OM[2:0]	通道2输出比较模式 参见CH1OM描述
11	CH2OSE	通道2输出比较影子寄存器使能 参见CH1OSE描述
10	CH2OFE	通道2输出比较快速使能 参见CH1OFE描述
9:8	CH2M[1:0]	通道2模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHE寄存器的CH2E位被清0)时这些位才可以写。 00: 通道2配置为输出 01: 通道2配置为输入，IC2映射在TI2上 10: 通道2配置为输入，IC2映射在TI1上 11: 通道2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 TIMER_SMC寄存器的TRGS位选择)
7	CH1OCE	通道1输出比较清0使能 当此位被置1，当检测到ETIF输入高电平时，OC1参考信号被清0。

		0: 禁止通道1输出比较清零 1: 使能通道1输出比较清零
6:4	CH1OM[2:0]	通道1输出比较模式  此位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。 OC1REF高电平有效，而OC1、OC1N的有效电平取决于CH1P、CH1NP位。 000: 冻结。输出比较寄存器TIMERx_CHCC1与计数器间的比较对OC1REF不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时，强制OC1REF为高 010: 匹配时设置为低。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时，强制OC1REF为低 011: 匹配时翻转。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时，强制OC1REF翻转 100: 强制为低。强制OC1REF为低电平 101: 强制为高。强制OC1REF为高电平 110: PWM模式1。在向上计数时，一旦计数器值小于TIMERx_CHCC1时，OC1REF为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CHCC1时，OC1REF为低电平，否则为高电平 111: PWM 模式2。在向上计数时，一旦计数器值小于TIMERx_CHCC1时，OC1REF为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CHCC1时，OC1REF为高电平，否则为低电平。 在PWM模式1或PWM模式2中，只有当比较结果改变了或者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。 当TIMERx_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。
3	CH1OSE	通道1输出比较影子寄存器使能  当此位被置1，TIMERx_CHCC1寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0: 禁止通道1输出/比较影子寄存器 1: 使能通道1输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTLR1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。 当TIMERx_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。
2	CH1OFE	通道1输出比较快速使能  该位用于通道配置为PWM模式时加快CC输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，OC被设置为比较电平而与比较结果无关。 0: 禁止通道1输出比较快速。当触发器的输入有一个有效沿时，激活CC1输出的最小延时为5个时钟周期 1: 使能通道1输出比较快速。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3个时钟周期
1:0	CH1M[1:0]	通道1模式选择  这些位定义了输入的方向和输入信号的选择。只有当通道关闭(TIMERx_CHE寄存器的CH1E位被清0)时这些位才可写。

- 00: 通道1配置为输出  
 01: 通道1配置为输入, IC1映射在TI1上  
 10: 通道1配置为输入, IC1映射在TI2上  
 11: 通道1配置为输入, IC1映射在TRC上. 此模式仅工作在内部触发输入被选中时  
 (通过设置TIMER\_SMC寄存器的TRGS位)

**输入捕获模式:**

位/位域	名称	描述
15:12	CH2ICF[3:0]	通道2输入捕获滤波控制  参见CH1ICF描述
11:10	CH2ICP[1:0]	通道2输入捕获预分频器  参见CH1ICP描述
9:8	CH2M[1:0]	通道2模式选择  此位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH2E位被清0)时这些位才可写.  00: 通道2配置为输出 01: 通道2配置为输入, IC2映射在TI2上 10: 通道2配置为输入, IC2映射在TI1上 11: 通道2配置为输入, IC2映射在TRC上. 此模式仅工作在内部触发输入被选中时 (通过设置TIMER_SMC寄存器的TRGS位).
7:4	CH1ICF[3:0]	通道1输入捕获滤波控制  数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。 这些位定义了TI1输入信号的采样频率和数字滤波器的长度。  0000: 无滤波器, $f_{SAMP}=f_{DTS}$ , N=1 0001: $f_{SAMP}=f_{PCLK}$ , N=2 0010: $f_{SAMP}=f_{PCLK}$ , N=4 0011: $f_{SAMP}=f_{PCLK}$ , N=8 0100: $f_{SAMP}=f_{DTS}/2$ , N=6 0101: $f_{SAMP}=f_{DTS}/2$ , N=8 0110: $f_{SAMP}=f_{DTS}/4$ , N=6 0111: $f_{SAMP}=f_{DTS}/4$ , N=8 1000: $f_{SAMP}=f_{DTS}/8$ , N=6 1001: $f_{SAMP}=f_{DTS}/8$ , N=8 1010: $f_{SAMP}=f_{DTS}/16$ , N=5 1011: $f_{SAMP}=f_{DTS}/16$ , N=6 1100: $f_{SAMP}=f_{DTS}/16$ , N=8 1101: $f_{SAMP}=f_{DTS}/32$ , N=5 1110: $f_{SAMP}=f_{DTS}/32$ , N=6 1111: $f_{SAMP}=f_{DTS}/32$ , N=8
3:2	CH1ICP[1:0]	通道1输入捕获prescaler  通道1输入捕获预分频器

这2位定义了CC1 输入的预分频系数。一旦CH1E =0(TIMERx\_CHE寄存器中), 则预分频器复位。

00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获

01: 每2个事件触发一次捕获

10: 每4个事件触发一次捕获

11: 每8个事件触发一次捕获

1:0 CH1M[1:0] 通道1模式选择

这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx\_CHE寄存器的CH1E位被清0)时这些位才可写。

00: 通道1配置为输出

01: 通道1配置为输入, IC1映射在TI1上

10: 通道1配置为输入, IC1映射在TI2上

11: 通道1配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER\_SMC寄存器的TRGS位)。

### TIMER1 通道控制寄存器 2 (TIMER1\_CHCTRL2)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4 OCE	CH4OM[2:0]			CH4 OSE	CH4 OFE	CH4M[1:0]	CH3 OCE	CH3OM[2:0]			CH3 OSE	CH3 OFE	CH3M[1:0]		
CH4ICF[3:0]	CH4ICP[1:0]			CH3ICF[3:0]			CH3ICP[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式:

位/位域	名称	描述
15	CH4OCE	通道4输出比较清0使能 参见CH1OCE描述
14:12	CH4OM[2:0]	通道4输出比较模式 参见CH1OM描述
11	CH4OSE	通道4输出比较影子寄存器使能 参见CH1OSE描述
10	CH4OFE	通道4输出比较快速使能 参见CH1OFE描述
9:8	CH4M[1:0]	通道4模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH4E位被清0)时这些位才可写。 00: 通道4配置为输出 01: 通道4配置为输入, IC4映射在TI4上

---

		10: 通道4配置为输入, IC4映射在TI3上 11: 通道4配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)。
7	CH3OCE	通道3输出比较清0使能 参见CH1OCE描述
6:4	CH3OM[2:0]	通道3输出比较模式 参见CH1OM描述
3	CH3OSE	通道3输出比较影子寄存器使能 参见CH1OSE描述
2	CH3OFE	通道3输出比较快速使能 参见CH1OFE描述
1:0	CH3M[1:0]	通道1模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH3E位被清0)时这些位才可写。 00: 通道3配置为输出 01: 通道3配置为输入, IC3映射在TI3上 10: 通道3配置为输入, IC3映射在TI4上 11: 通道3配置为输入, IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)。

#### 输入捕获模式:

---

位/位域	名称	描述
15:12	CH4ICF[3:0]	通道4输入捕获滤波控制 参见CH1ICF描述
11:10	CH4ICP[1:0]	通道4输入捕获预分频器 参见CH1ICP描述
9:8	CH4M[1:0]	通道4模式选择 这两位定义了通道的方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH4E位被清0)时这些位才可写。 00: 通道4配置为输出 01: 通道4配置为输入, IC4映射在TI4上 10: 通道4配置为输入, IC4映射在TI3上 11: 通道4配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)。
7:4	CH3ICF[3:0]	通道3输入捕获滤波控制 参见CH1ICF描述
3:2	CH3ICP[1:0]	通道3输入捕获预分频器 参见CH1ICP描述

1:0	CH3M[1:0]	通道3 模式选择 这两位定义了通道的方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH3E位被清0)时这些位才可写。 00: 通道3配置为输出 01: 通道3配置为输入, IC3映射在TI3上 10: 通道3配置为输入, IC3映射在TI4上 11: 通道3配置为输入, IC3映射在TRC上. 此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位).
-----	-----------	---

### TIMERx 通道使能寄存器 (TIMERx\_CHE)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4NP	保留	CH4P	CH4E	CH3NP	保留	CH3P	CH3E	CH2NP	保留	CH2P	CH2E	CH1NP	保留	CH1P	CH1E

位/位域	名称	描述
15	CH4NP	通道4极性 参见CH1NP描述
14	保留	必须保持复位值
13	CH4P	通道4极性 参见CH1P描述
12	CH4E	通道4使能 参见CH1E描述
11	CH3NP	通道3极性 参见CH1NP描述
10	保留	必须保持复位值
9	CH3P	通道3极性 参见CH1P描述
8	CH3E	通道3使能 参见CH1E描述
7	CH2NP	通道2极性 参见CH1NP描述
6	保留	必须保持复位值
5	CH2P	通道2极性

参见CH1P描述

4	CH2E	通道2使能 参见CH1E描述
3	CH1NP	通道1极性 当通道1配置为输出模式时，此位必须被设置为0。 当通道1配置为输入模式时，此位和CH1P联合使用，作为输入信号的极性选择控制信号，详情请参照CH1P的说明。
2	保留	必须保持复位值
1	CH1P	通道1极性 当通道1配置为输出模式时，此位定义了输出信号极性。 0：通道1高电平有效 1：通道1低电平有效 当通道1配置为输入模式时， [CH1NP, CH1P] 作为输入信号的极性选择控制位，功能如下： [CH1NP==0, CH1P==0]: 把TIxFP1的上升沿作为捕获/触发的有效信号，并且TIxFP1不会被翻转。 [CH1NP==0, CH1P==1]: 把TIxFP1的下降沿作为捕获/触发的有效信号，并且TIxFP1会被翻转。 [CH1NP==1, CH1P==0]: 保留。不可设置。 [CH1NP==1, CH1P==1]: 把TIxFP1的上升沿和下降沿都作为捕获/触发的有效信号，并且TIxFP1不会被翻转。 当TIMERx_BKDT寄存器的LK[1:0]=11或10时此位不能被更改。
0	CH1E	通道1使能 当通道1配置为输入模式时，将此位置1使能OC1信号有效。当通道1配置为输出模式时，将此位置1使能通道1上的捕获事件。 0：禁止通道1 1：使能通道1

### TIMERx 计数器寄存器 (TIMERx\_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16] 仅 TIMER2 有高 16bit。															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	CNT[31:16]	这些位是当前的计数值，写操作能改变计数器值。仅TIMER2有高16bit。
15:0	CNT[15:0]	这些位是当前的计数值，写操作能改变计数器值。

### TIMERx 预分频寄存器 (TIMERx\_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

### TIMERx 计数器自动重载寄存器 (TIMERx\_CARL)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARL [31:16] 仅 TIMER2 有高 16bit。															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL [15:0]															
rw															

位/位域	名称	描述
31:16	CARL [31:16]	计数器自动重载值，仅TIMER2有高16bit。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

### TIMERx 通道 1 捕获/比较寄存器 (TIMERx\_CHCC1)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHCC1 [31:16] 仅 TIMER2 有高 16bit。															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC1 [15:0]															

rw

位/位域	名称	描述
31:16	CHCC1	通道1的捕获或比较值，仅TIMER2有高16bit。 [31:16]
15:0	CHCC1[15:0]	通道1的捕获或比较值低16位值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMERx 通道 2 捕获/比较寄存器 (TIMERx\_CHCC2)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHCC2 [31:16] 仅 TIMER2 有高 16bit。															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC2 [15:0]															

rw

位/位域	名称	描述
31:16	CHCC2	通道2的捕获或比较值，仅TIMER2有高16bit。 [31:16]
15:0	CHCC2[15:0]	通道2的捕获或比较值 当通道2配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道2配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMERx 通道 3 捕获/比较寄存器 (TIMERx\_CHCC3)

地址偏移: 0x3C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

CHCC3 [31:16] 仅 TIMER2 有高 16bit。
----------------------------------

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

CHCC3 [15:0]
--------------

rw

位/位域	名称	描述
31:16	CHCC3[31:16]	通道3的捕获或比较值，仅TIMER2有高16bit。
15:0	CHCC3[15:0]	通道3的捕获或比较值 当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMERx 通道 4 捕获/比较寄存器 (TIMERx\_CHCC4)

地址偏移: 0x40

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

CHCC4 [31:16] 仅 TIMER2 有高 16bit。
----------------------------------

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

CHCC4 [15:0]
--------------

rw

位/位域	名称	描述
31:16	CHCC4[31:16]	通道4的捕获或比较值，仅TIMER2有高16bit。
15:0	CHCC4[15:0]	通道4的捕获或比较值 当通道4配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道4配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMERx DMA 控制寄存器 (TIMERx\_DCTLR)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

rw

位/位域	名称	描述
15:14	保留	必须保持复位值
12:8	DBLTH[4:0]	DMA连续传送长度 这5位定义了DMA 在连续模式下的传送长度（当对TIMERx_DTRSF寄存器进行读或写时，定时器则进行一次连续传送）。
7:5	保留	必须保持复位值
4:0	DBAR[4:0]	DMA基地址 这5位定义了DMA 传送的基地址（当对TIMEx_DTRSF寄存器进行读或写时，DBA定义为从TIMERx_CTRLR1 寄存器所在地址开始的偏移量）。

### TIMERx DMA 传送寄存器 (TIMERx\_DTRSF)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:0	DTRSF[15:0]	DMA传送 对TIMER1_DTRSF寄存器的读或写会导致对从TIMERx_CTRLR1 寄存器开始的以下地址范围所在寄存器的访问： (DBAR + 连续传送计数器) × 4 连续传送计数器由硬件计算，范围为0到DBLTH。

### TIMERx 配置寄存器 (TIMERx\_CFGR) (仅适用于 GD32F170xx and GD32F190xx 产品)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:2	保留	必须保持复位值

---

1

CCSEL

写捕获比较寄存器选择位

此位由软件写1或清0。

1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效

0: 无效

0

保留

必须保持复位值

## 9.3. 基本定时器 (TIMER6)

### 9.3.1. 简介

通用定时器(TIMER6)由1个16位的计数自动重载寄存器 (TIMERx\_CARL)和一些控制寄存器组成。可以被用作通用定时器和为DAC (数字到模拟转换器)提供时钟。TIMER6的触发连接到DAC，可以触发DAC。

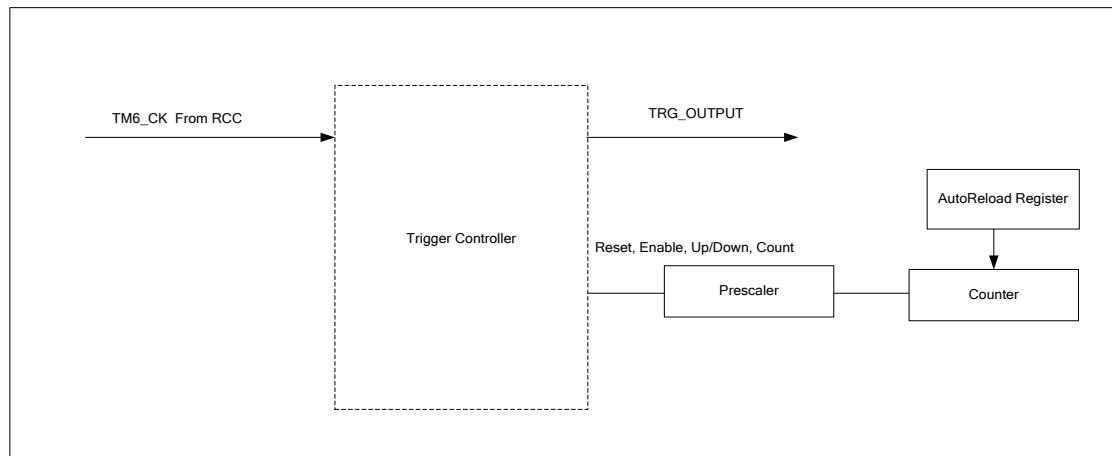
### 9.3.2. 主要特性

- 16位向上自动重载计数器
- 16位可编程的预分频器，计数器时钟频率的分频系数为1~65536之间的任意数值
- 触发DAC的同步电路
- 更新事件产生中断/DMA

### 9.3.3. 功能描述

下图提供了基本定时器内部配置细节图

图 9-80. 通用定时器框图 (TIMER6)



### 预分频计数器

预分频器可以将定时器(PCLK)的时钟频率按1到65536之间的任意值分频给计数器(CNT\_CLK)使用。分频系数受预分频寄存器TIMERx\_PSC控制，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 9-81. 当预分频器的参数从 1 变到 2 时，计数器时序图

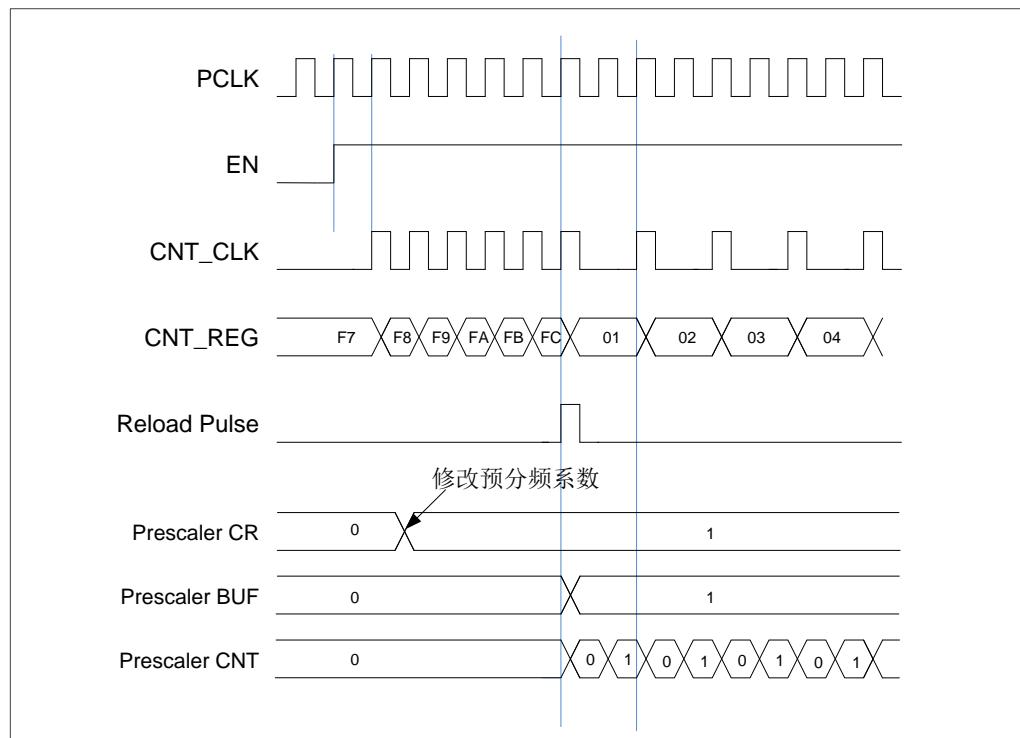
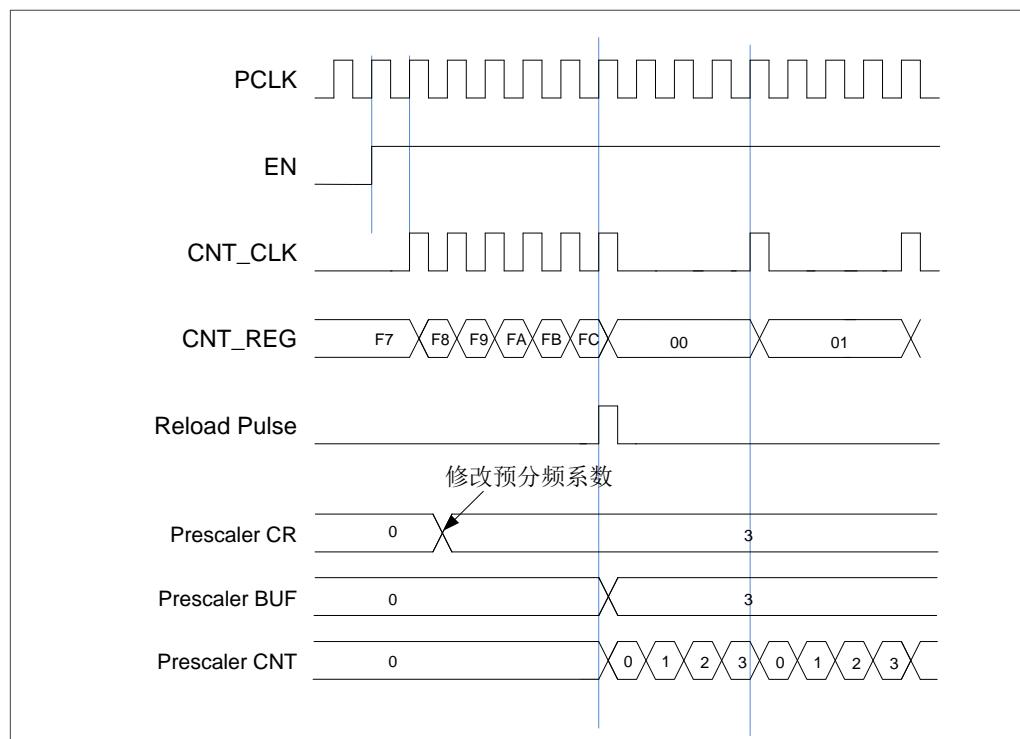


图 9-82. 当预分频器的参数从 1 变到 4 时，计数器时序图



### 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(TIMERx\_CARL寄存器的内容)，然后重新从0开始计数。每次计数器溢出时都会产生更新事件。

可以将TIMERx\_EVG寄存器的UPG位置1来设置更新事件，计数值被清0产生更新事件。

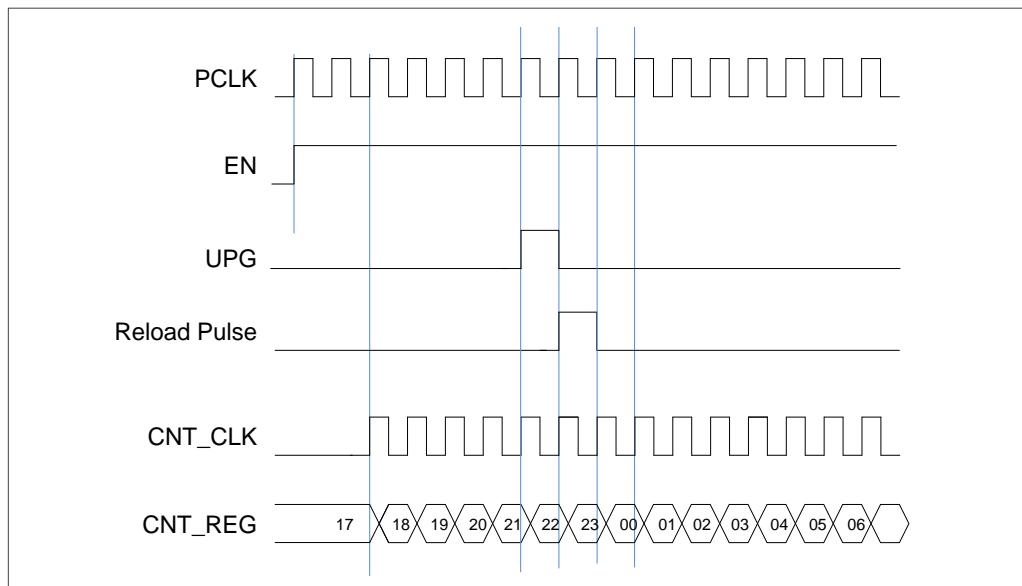
设置TIMERx\_CTLR1寄存器的UPDIS位可以禁止更新事件

一旦发生更新事件，所有的寄存器(重复计数器，重载寄存器，预分频寄存器)都将被更新。

### 时钟选择

基本定时器只有唯一的时钟源RCC，为基本定时器提供内部时钟PCLK。计数器和预分频计数器被这个内部时钟PCLK驱动。UPG如果被置位，将初始化计数器和预分频计数器。

图 9-83. 内部时钟分频为 1 时正常模式下的控制电路



### 定时器 debug 模式

当微控制器在调试模式，若Cortex™-M3内核停止，且MCUDBG模块中DBG\_TIMERx\_STOP配置位被置1，TIMERx计数器停止。

### 9.3.4. TIMER6 寄存器

#### TIMER6 控制寄存器 1 (TIMER6\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		ARSE		保留		SPM		UPS		UPDIS		CEN			
								rw				rw		rw	

位/位域	名称	描述
15:8	保留	必须保持复位值
7	ARSE	自动重载影子使能 0: 禁止TIMERx_CARL寄存器的影子寄存器 1: 使能TIMERx_CARL寄存器的影子寄存器
3	SPM	单脉冲模式。 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
2	UPS	更新请求源 软件通过该位选择更新事件源 0: 使能后, 下述任一事件产生更新中断或DMA请求: - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生的更新 1: 使能后只有计数器溢出/下溢才产生更新中断或DMA请求。
1	UPDIS	禁止更新 软件通过该位允许/禁止更新事件的产生。 0: 允许更新事件产生。当以下事件之一发生时, 更新事件产生, 具有缓存的寄存器被装入它们的预装载值: - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生一个更新事件 1: 禁止更新事件。带有缓存的寄存器保持原有值, 如果UPG位被置1或者从模式控制器产生一个硬件复位事件, 计数器和预分频器被重新初始化。
0	CEN	使能计数器 0: 禁止计数器 1: 使能计数器 在软件将CEN位置1后, 外部时钟、暂停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。

### TIMER6 控制寄存器 2 (TIMER6\_CTLR2)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:7	保留	必须保持复位值
6:4	MMC[2:0]	<p>主模式控制</p> <p>这些位控制TRGO信号的选择，主模式下发给从定时器用于同步功能。</p> <p>000: 复位. TIMERx_EVG寄存器的UG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。</p> <p>001: 使能. 此模式可用于同时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TIMERx_EN 被主模式控制器选为触发输出(TRGO)。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。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。</p> <p>010: 更新。更新事件被主模式控制器选为触发输入</p> <p>011: 捕获/比较脉冲。在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲</p> <p>100: 比较。OC1REF 信号被用于作为触发输出</p> <p>101: 比较。OC2REF 信号被用于作为触发输出</p> <p>110: 比较。OC3REF 信号被用于作为触发输出</p> <p>111: 比较。OC4REF 信号被用于作为触发输出</p>
3:0	保留	必须保持复位值

### TIMER6 DMA 和中断使能寄存器 (TIMER6\_DIE)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述

---

15:9	保留	必须保持复位值
8	UPDE	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### TIMER6 状态寄存器 (TIMER6\_STR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rc\_w0

---

位/位域	名称	描述
15:1	保留	必须保持复位值
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

### TIMER6 事件产生寄存器 (TIMER6\_EVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

w

---

位/位域	名称	描述
15:1	保留	必须保持复位值
0	UPG	此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。

0: 无更新事件产生

1: 产生更新事件

### **TIMER6 计数器寄存器 (TIMER6\_CNT)**

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

### **TIMER6 预分频寄存器 (TIMER6\_PSC)**

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

### **TIMER6 计数器自动重载寄存器 (TIMER6\_CARL)**

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值

这些位定义了计数器的自动重载值。

## 9.4. 通用定时器 (TIMER14)

### 9.4.1. 简介

通用定时器 (TIMER14) 包括一个16位的向上计数器、1个捕捉/比较寄存器(TIMERx\_CHCC)、1个16位的自动装载计数器寄存器(TIMERx\_CARL)和一些控制寄存器。它适合多种用途，包括通用定时器，输入信号脉冲宽度测量或者产生输出波形例如单脉冲输出或PWM输出。

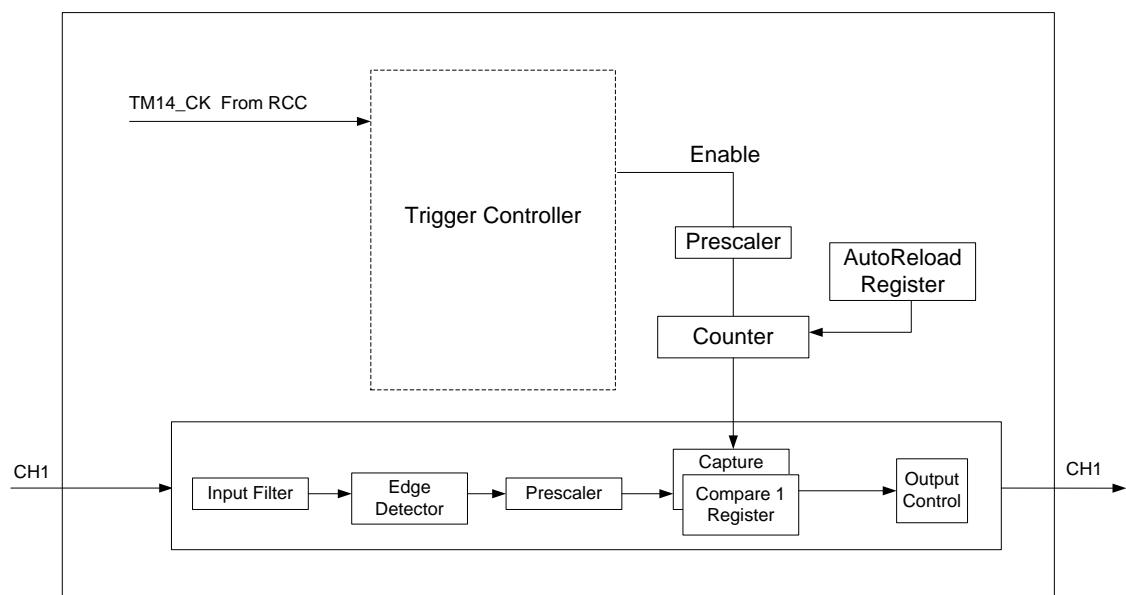
### 9.4.2. 主要特性

- 16位向上自动重载计数器
- 16位可编程的预分频器，计数器时钟频率的分频系数为1~65536之间的任意数值
- 1个独立通道，支持以下功能：输入捕获，输出比较，PWM波形产生(边缘或中间对齐模式)
- 更新事件，输入捕获事件，输出比较事件可以产生中断/DMA

### 9.4.3. 功能描述

下图提供了通用定时器内部配置细节图

图 9-84. 通用定时器框图 (TIMER14)



## 预分频计数器

预分频器可以将定时器(PCLK)的时钟频率按1到65536之间的任意值分频给计数器(CNT\_CLK)使用。分频系数受预分频寄存器TIMERx\_PSC控制，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 9-85. 当预分频器的参数从 1 变到 2 时，计数器时序图

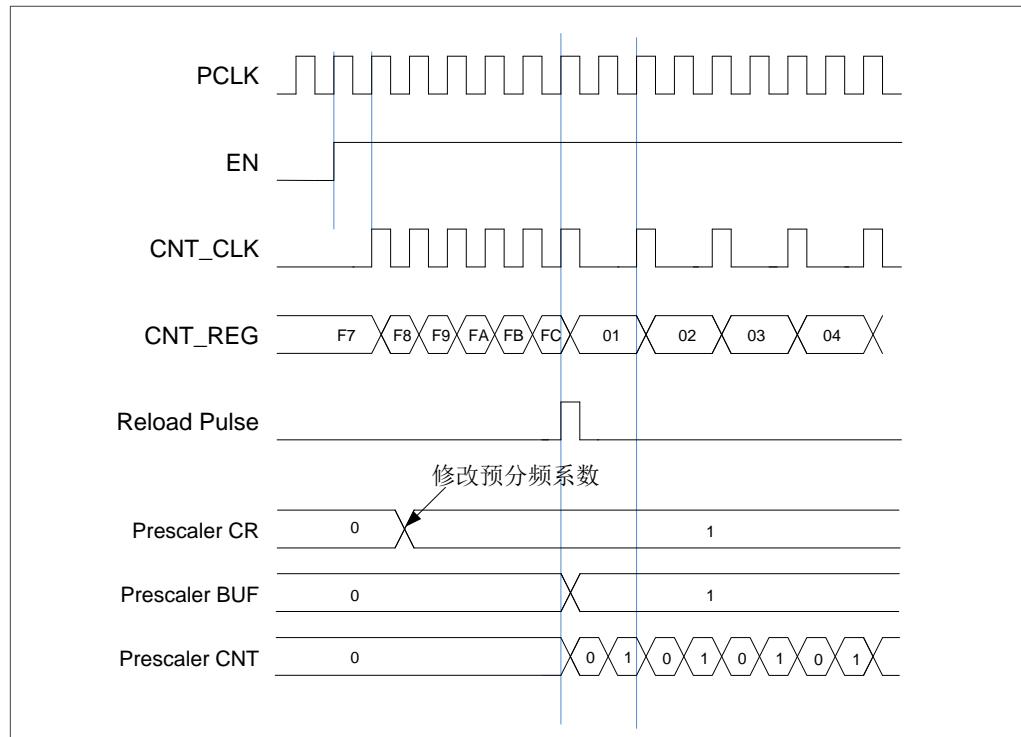
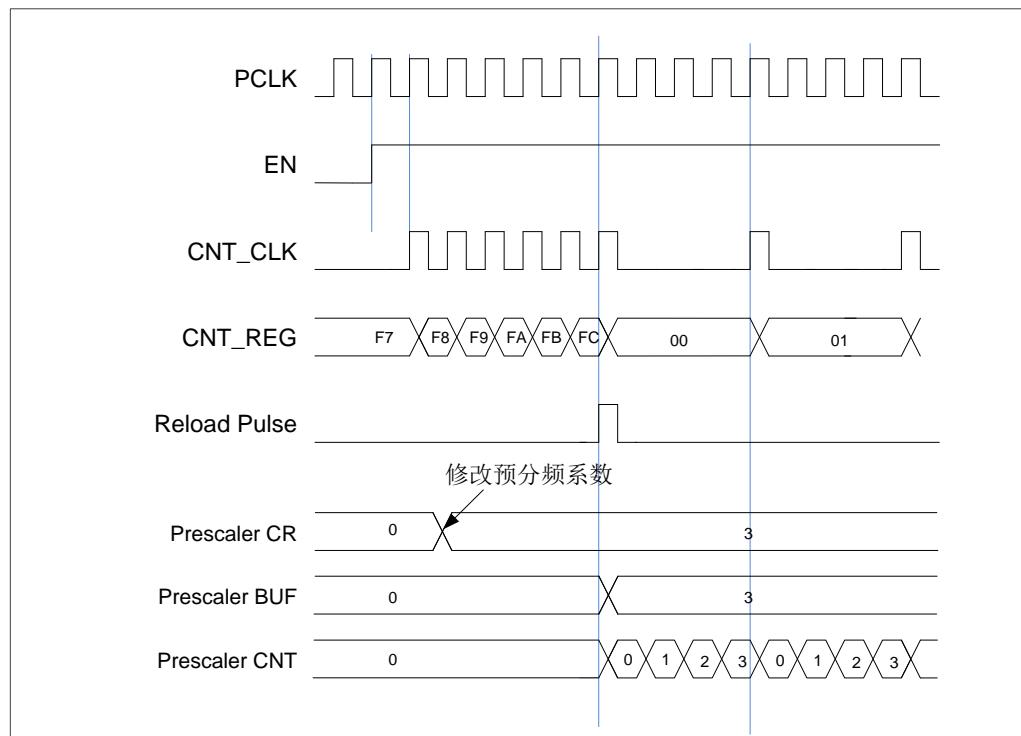


图 9-86. 当预分频器的参数从 1 变到 4 时，计数器时序图



## 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，然后重新从0开始计数。每次计数器溢出时都会产生更新事件。

可以将TIMERx\_EVG寄存器的UPG位置1来设置更新事件，计数值被清0产生更新事件。

设置TIMERx\_CTLR1寄存器的UPDIS位可以禁止更新事件

一旦发生更新事件，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作。

图 9-87. 计数器时序图，内部时钟分频系数为 1

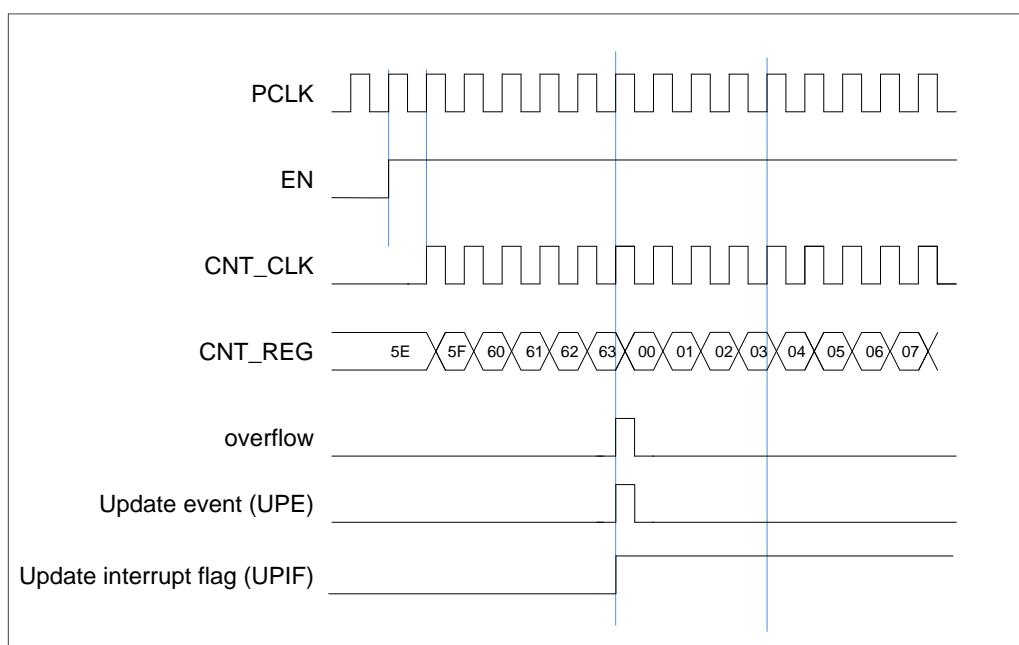


图 9-88. 计数器时序图，内部时钟分频系数为 2

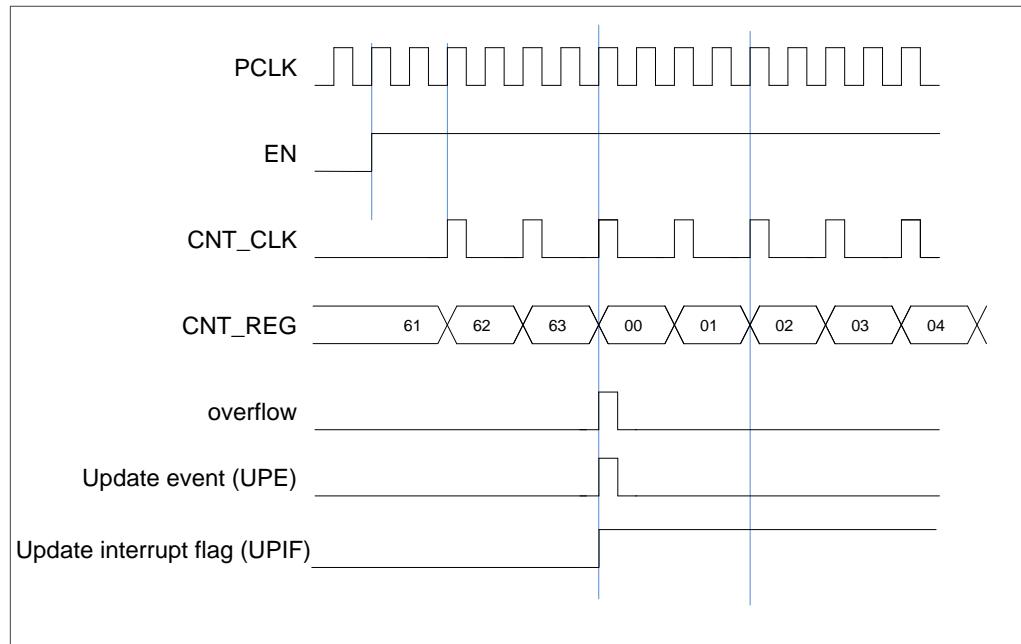


图 9-89. 计数器时序图，内部时钟分频系数为 4

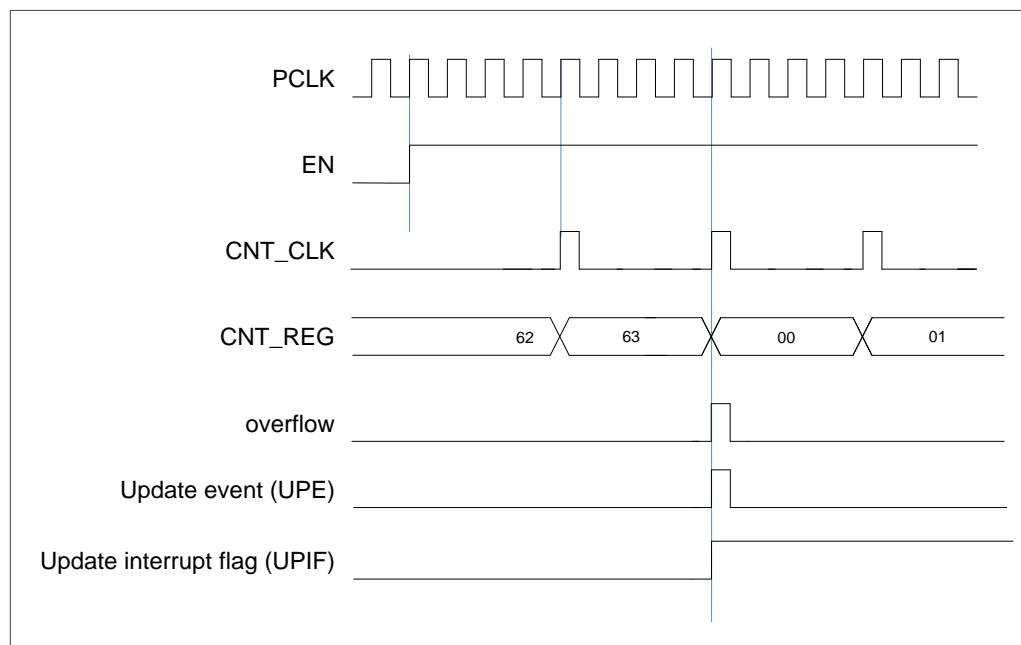


图 9-90. 计数器时序图，内部时钟分频系数为 N

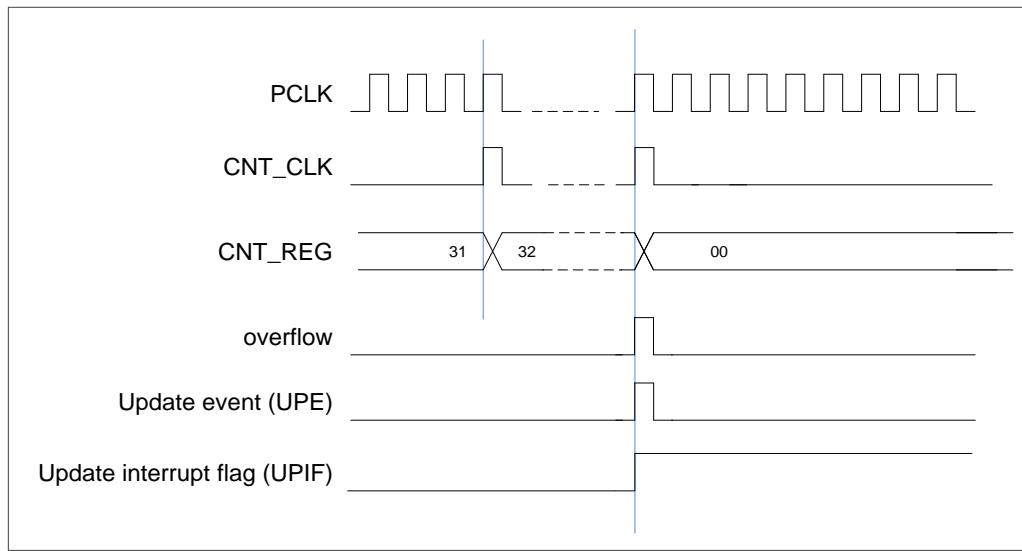


图 9-91. 计数器时序图, 当 ARSE=0 时的更新事件

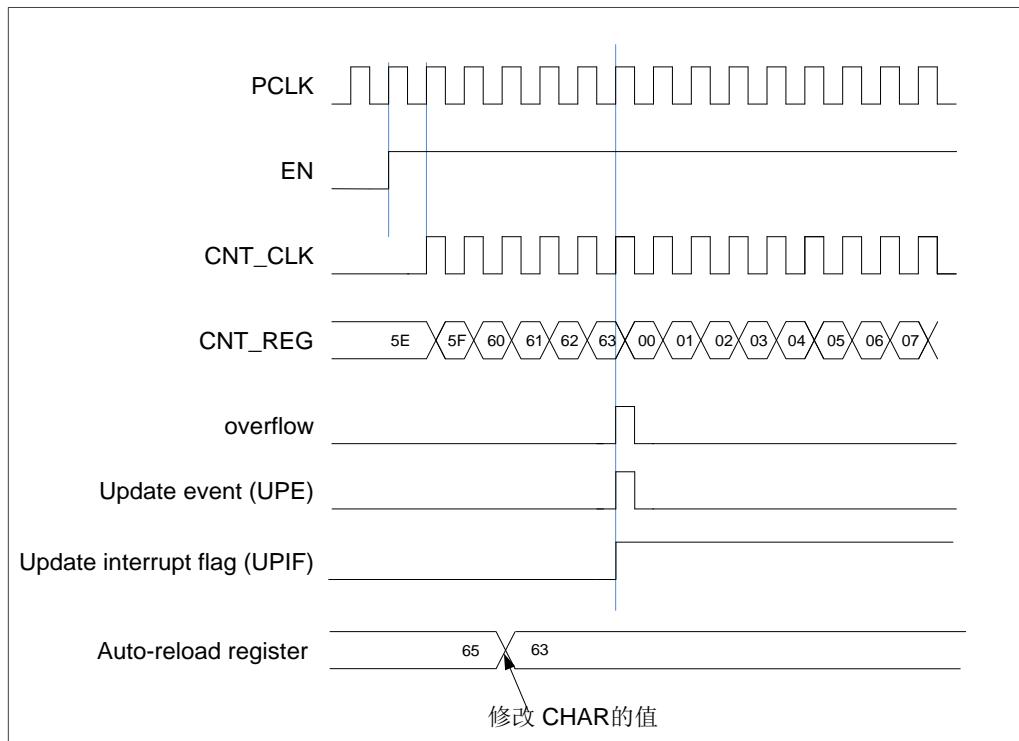
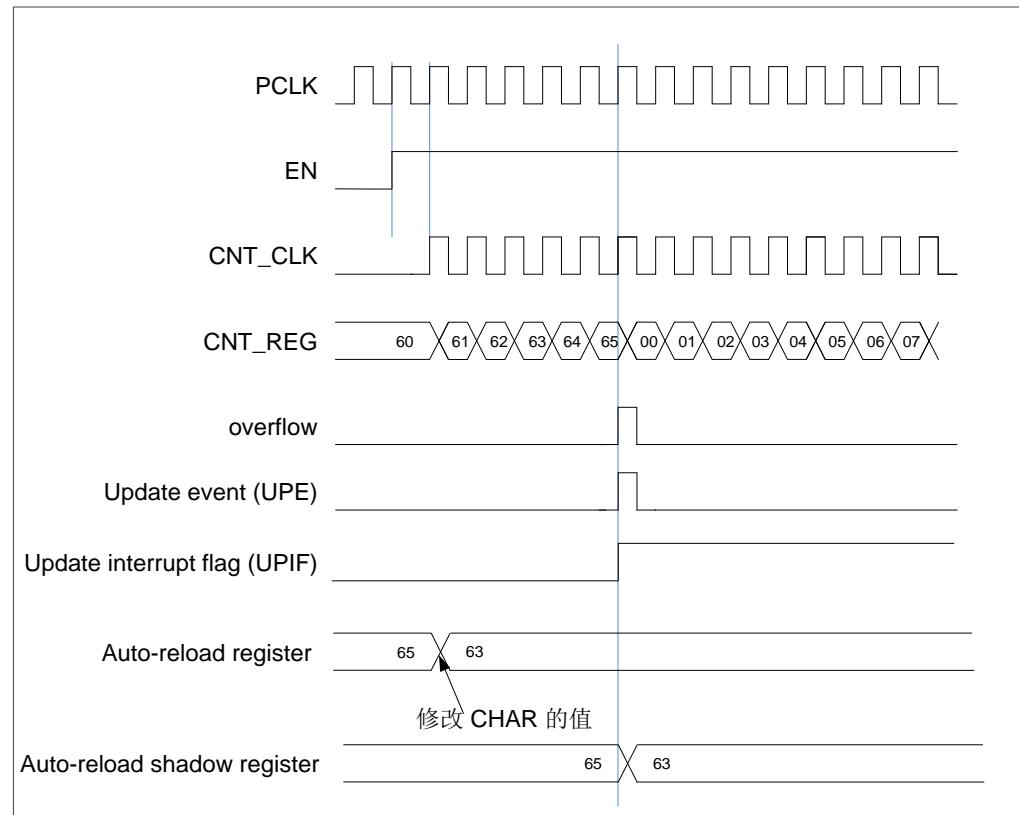


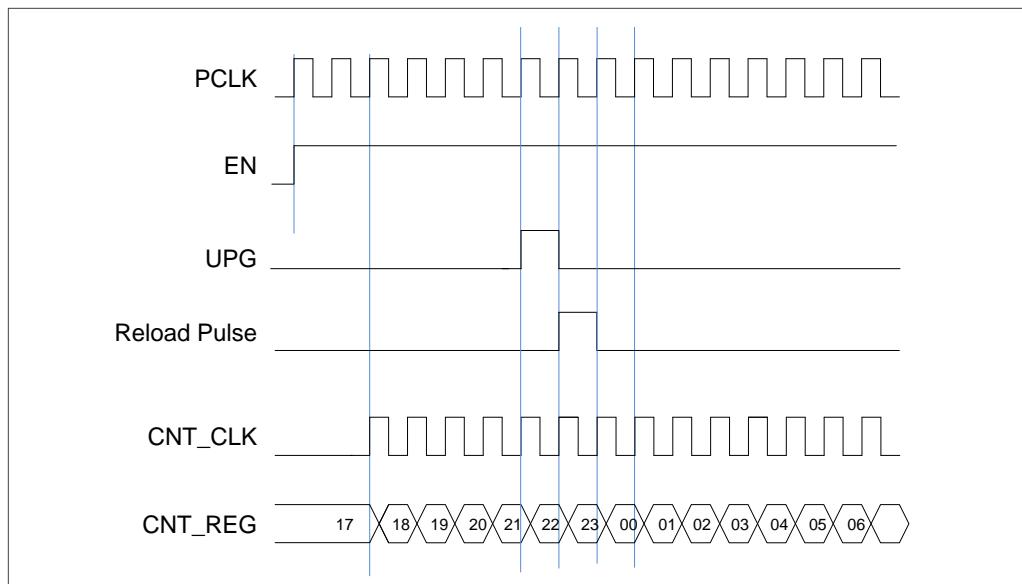
图 9-92. 计数器时序图, 当 ARSE=1 时的更新事件



### 时钟选择

通用定时器只有唯一的时钟源RCC，为定时器提供内部时钟PCLK。计数器和预分频计数器被这个内部时钟PCLK驱动。UPG如果被置位，将初始化计数器和预分频计数器。

图 9-93. 内部时钟分频为 1 时正常模式下的控制电路



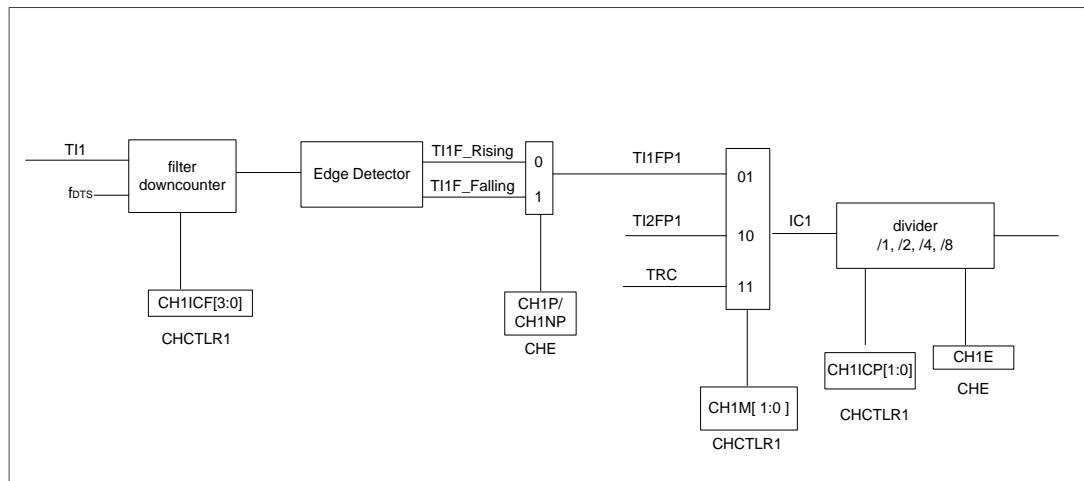
### 捕获/比较通道

TIMER14拥有1个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级、通道控制器和输出级。

#### ■ 输入捕获级

输入级包括一个数字滤波器，一个通道极性选择，边沿监测和一个通道预分频器。通道输入信号(TIx)可以被选择来自TIMERx\_CH1信号。通道输入信号(TIx)被一个数字滤波器采样来产生一个滤波输入信号TIxF。然后通道极性和边沿监测模块产生一个TIxFP1或TIxFP2信号，来实现输入捕捉功能。通道输入预分频寄存器(CHxICP)可以设置有效输入事件数目。

图 9-94. 捕获/比较通道(例如：输入级)



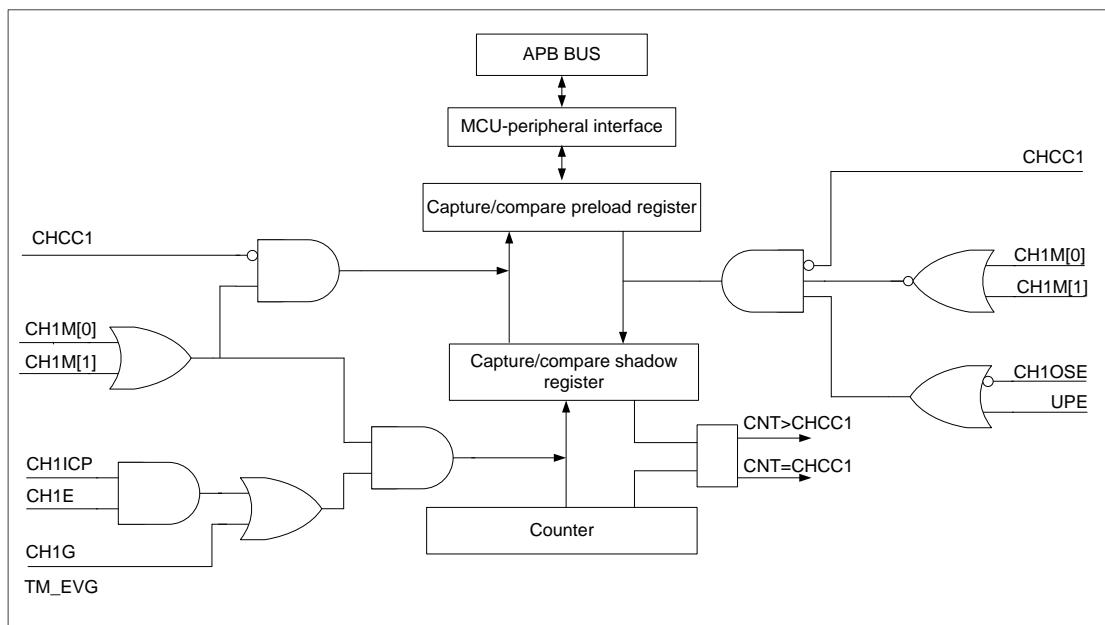
### ■ 通道控制器

TIMER14有1个独立的通道用来捕获输入或输出匹配比较

用于输入捕获模式下，当捕获事件发生时，**TIMERx\_CHCCx**影子寄存器首先捕获计数值然后传送到**TIMER14\_CHCCx**预装载寄存器中。

用于输出匹配比较模式下，**TIMERx\_CHCCx**预装载寄存器的值被复制到相应的影子寄存器中，然后计数值和影子寄存器的值比较。

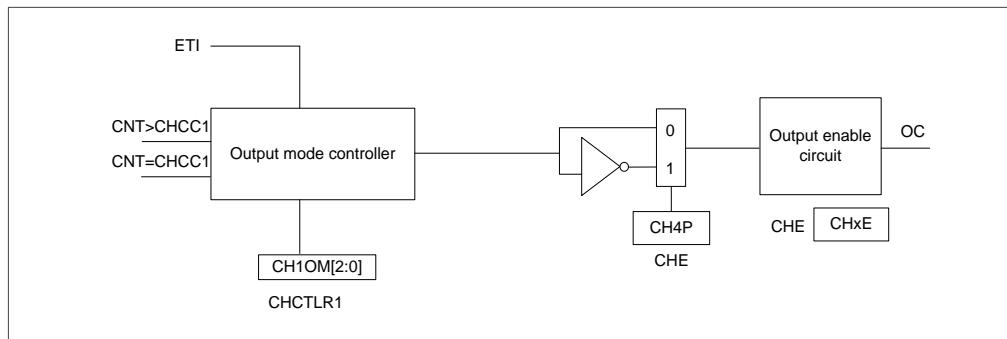
图 9-95. 捕获/比较通道 1 主电路



### ■ 输出级

TIMER14有1个用于匹配比较或PWM输出功能的通道。

图 9-96. 捕获/比较通道输出级



### 输入捕获模式

通道用于输入捕获时，当一个有效输入信号的边沿跳变时，计数值被捕获到捕获/比较寄存器(TIMERx\_CHCCx)中。一旦捕获事件发生，TIMERx\_STR寄存器的CHxIF标志位被置1。如果CHxIF位已经被置1，还没来得及被软件清除，此通道上又有另一个捕获事件发生，相应通道捕获溢出标志位CHxOF将被置1。一旦捕获事件发生，CHxDE位决定了是否产生DMA请求，CHxIE位决定了是否产生中断。使用输入捕获模式需按以下步骤配置：

1. 配置通道控制寄存器(TIMERx\_CHCTRLRx)的CCxM位域，选择输入信号
2. 配置通道控制寄存器(TIMERx\_CHCTRLRx)的CHxICF位域，为输入信号添加输入滤波器
3. 配置通道使能寄存器(TIMERx\_CHE)的CHxP/CHxNP位，选择输入捕获的边沿（上升沿或下降沿）
4. 如果输入信号需要分频，配置通道控制寄存器(TIMERx\_CHCTRLRx)的CHxICP位域
5. 如果需要中断，将DMA中断使能寄存器(TIMERx\_DIE)的CHxIE位置位
6. 如果需要DMA请求，将DMA中断使能寄存器(TIMERx\_DIE)的CHxDE位置位
7. 将通道使能寄存器(TIMERx\_CHE)的CCxE位置位，使能通道

按上述配置完寄存器后，该通道上的对应边沿信号将产生输入捕获事件。计数值被捕获到通道捕获/比较寄存器(TIMERx\_CHCCx)中。如果CHxIE位被置位，将会产生中断。如果CHxDE位被置位，将会产生DMA请求。

输入捕获模式也可以用于TIMERx\_CHx(TIx)引脚的信号脉宽测量。例如，PWM信号连接到TI1输入。配置TIMERx\_CHCTRLRx寄存器的CH1M位域为01，将通道1输入捕获信号连接到TI1，并设置上升沿捕获。配置TIMERx\_CHCTRLRx寄存器的CH2M位域为10，选择通道2捕获信号连接到TI1，并设置下降沿捕获。计数器配置为通道1上升沿触发后重新计数模式。此时TIMERx\_CHCC1可以测量PWM信号的周期，TIMERx\_CHCC2可以测量PWM信号的占空比。

### 输出比较模式/PWM 模式

当配置通道控制寄存器(TIMERx\_CHCTRLRx)的CCxM位域为00，使通道工作于输出模式时，

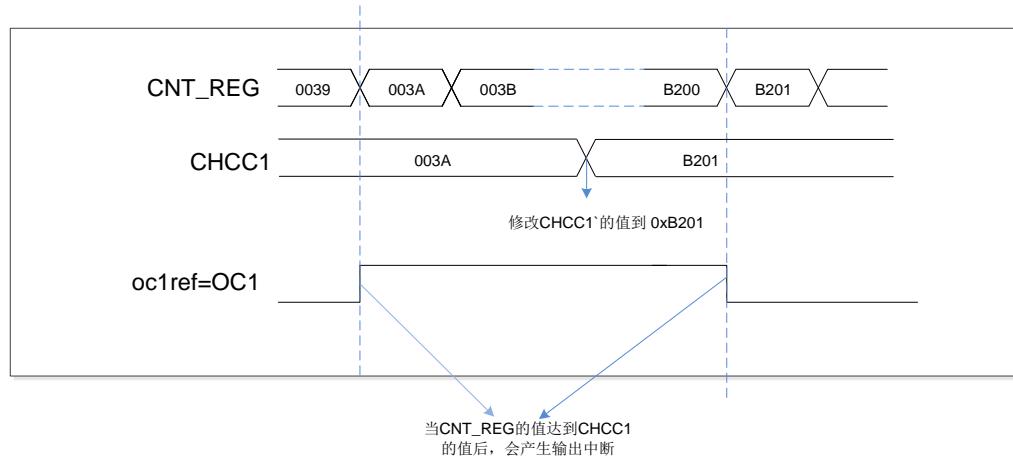
通道输出的波形由 **TIMERx\_CHCTRLx** 的 **CHxOM** 位域和计数器与 **TIMERx\_CHCCx** 寄存器的值比较的结果决定。当比较的值相等时状态寄存器(**TIMERx\_STR**)的 **CHxIF** 位将置 1。根据 **CHxDE** 或 **CHxIE** 位的值，一次 DMA 请求或中断将被产生。

在输出模式，配置 **CHxOM** 位域为 101，通道将持续输出有效电平。配置 **CHxOM** 位域为 100 通道将持续输出无效电平。

在输出模式，配置 **CHxOM** 位域为 001，当计数器与 **TIMERx\_CHCCx** 寄存器的值相等时，通道输出有效电平。配置 **CHxOM** 位域为 010，当计数器与 **TIMERx\_CHCCx** 寄存器的值相等时，通道输出无效电平。

在输出比较翻转模式（配置 **CHxOM** 位域为 011），当计数器与 **TIMERx\_CHCCx** 寄存器的值相等时，通道输出电平将翻转。

图 9-97. 输出比较模式，在 OC1 上切换



在 PWM输出模式（配置**CHxOM**位域为110（PWM模式1））或111（PWM模式2）。根据 **TIMERx\_CARL**寄存器和**TIMERx\_CHCCx**寄存器的值，通道输出PWM波形。

图 9-98. 在 OC1 上输出比较 PWM 模式 1，向上计数模式

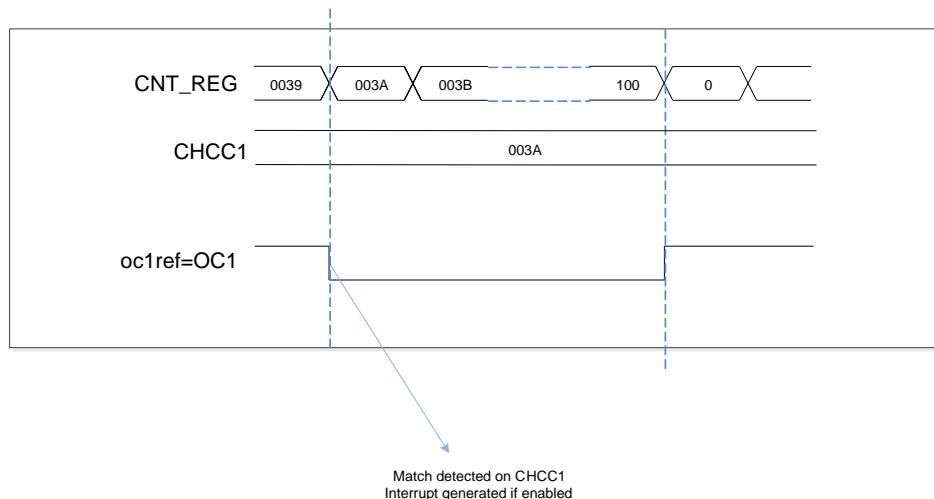
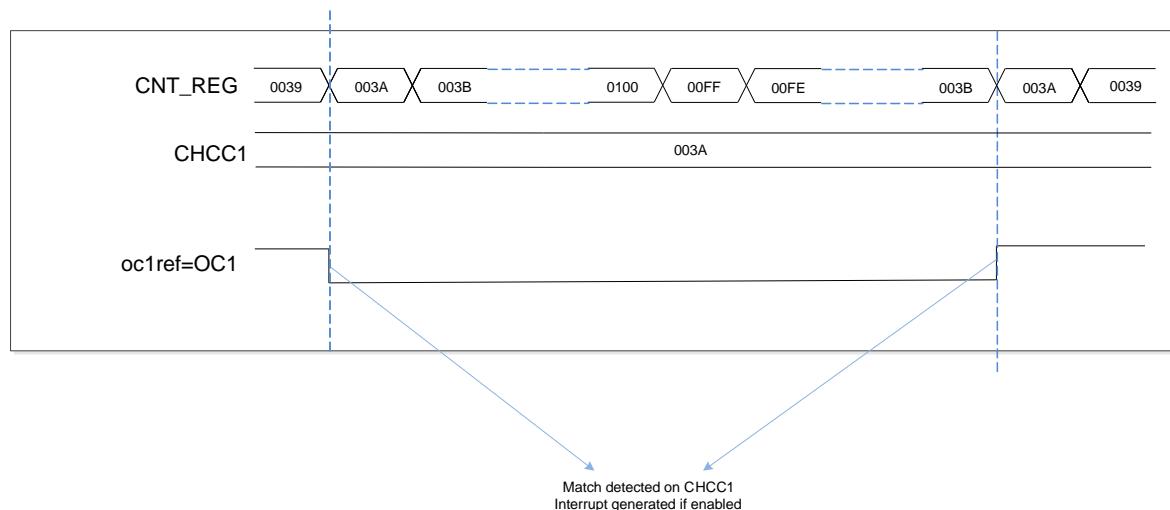


图 9-99. 在 OC1 上输出比较 PWM 模式 1，中央对齐计数模式



### 通道输出参考信号

当TIMER14用于输出匹配比较模式下，设置CHxOM位可以定义OCxREF信号(通道x输出参考信号)类型。OCxREF信号有一些输出功能类型，包括，设置CHxOM=0x00可以保持原始电平，设置CHxOM=0x01可以将OCxREF信号设置为高电平，设置CHxOM=0x02可以将OCxREF信号设置为低电平，在计数值和TIMER1\_CHCCx寄存器的值匹配时，设置CHxOM=0x03可以切换信号。

PWM模式1和PWM模式2输出也是另一种类型的OCxREF输出，具体设置是通过设置CHxOM =0x06或0x07来完成的。在这些模式中，OCxREF信号级别的改变取决于根据计数方向，以及TIMERx\_CHCCx寄存器值和计数值之间的关系。详情参考相应位定义。

设置CHxOM=0x04或0x05可以实现OCxREF信号的强制输出功能。输出比较信号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。

当ETIF信号源于外部TIMERx\_ETI引脚且被设置为高电平(设置TIMERx\_CHCTRL1寄存器的CHxOCE位为1)时，OCxREF信号被强制变为低电平。直到下一次更新事件发生，OCxREF信号才会回到它的有效状态。

### 定时器 debug 模式

当微控制器在调试模式，若Cortex™-M3内核停止，且MCUDBG模块中DBG\_TIMERx\_STOP配置位被置1，TIMERx计数器停止。

#### 9.4.4. TIMER14 寄存器

##### TIMERx 控制寄存器 1 (TIMERx\_CTRL1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					保留		CDIV[1:0]		ARSE		保留		UPS	UPDIS	CEN
								rw		rw			rw	rw	rw

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CDIV[1:0]	<p>时钟分频</p> <p>软件配置CDIV位规定定时器时钟(PCLK) 和死区时间、采样时钟(DTS)之间的分频系数，该位被死区时间产生器和数字滤波器使用。</p> <ul style="list-style-type: none"> <li>00: <math>f_{DTS}=f_{PCLK}</math></li> <li>01: <math>f_{DTS}=f_{PCLK}/2</math></li> <li>10: <math>f_{DTS}=f_{PCLK}/4</math></li> <li>11: 保留</li> </ul>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁止TIMERx_CARL寄存器的影子寄存器</p> <p>1: 使能TIMERx_CARL寄存器的影子寄存器</p>
6:3	保留	必须保持复位值
2	UPS	<p>更新请求源</p> <p>软件通过该位选择更新事件源。</p> <p>0: 使能后，下述任一事件产生更新中断:</p> <ul style="list-style-type: none"> <li>- UPG位被置1</li> <li>- 计数器溢出/下溢</li> <li>- 从模式控制器产生的更新</li> </ul> <p>1: 使能后只有计数器溢出/下溢才产生更新中断</p>
1	UPDIS	<p>禁止更新</p> <p>软件通过该位允许/ 禁止更新事件的产生</p> <p>0: 允许更新事件产生。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值:</p> <ul style="list-style-type: none"> <li>- UPG位被置1</li> <li>- 计数器溢出/下溢</li> <li>- 从模式控制器产生一个更新事件</li> </ul> <p>1: 禁止更新事件。带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化</p>
0	CEN	<p>使能计数器</p> <p>0: 禁止计数器</p> <p>1: 使能计数器</p> <p>在软件将CEN位置1后，外部时钟、暂停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。</p>

### TIMERx 中断使能寄存器 (TIMERx\_DIE)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH1IE	UPIE
rw														rw	rw

位/位域	名称	描述
15:2	保留	必须保持复位值.
1	CH1IE	通道1中断使能 0: 禁止通道1中断 1: 使能通道1中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### TIMERx DMA 和中断状态寄存器 (TIMERx\_STR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH1OF	保留				CH1IF	保留				rc_w0	rc_w0
rc_w0														rc_w0	rc_w0

位/位域	名称	描述
15:10	保留	必须保持复位值
9	CH1OF	通道1重复捕获标志 当通道1被配置为输入模式时, 在CH1IF标志位已经被置1后, 捕获事件再次发生时, 该标志位可以由硬件置1。该标志位由软件清0。 0: 无重复捕获中断发生 1: 发生了重复捕获中断
8:2	保留	必须保持复位值
1	CH1IF	通道1中断标志 此标志由硬件置1软件清0。当通道1在输入模式下时, 捕获事件发生时此标志位被置1; 当通道1在输出模式下时, 此标志位在一个比较事件发生时被置1。 0: 无通道1中断发生

1: 通道1中断发生

0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0. 0: 无更新中断发生 1: 发生更新中断
---	------	--

### TIMERx 事件发生寄存器 (TIMERx\_EVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH1G	UPG
W														W	

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CH1G	通道1捕获或比较事件发生 该位由软件置1，用于在通道1产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CC1IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道1配置为输入模式，计数器的当前值被TIMERx_CHCC1寄存器捕获，如果CH1IF标志位已经为1，则CH1OF标志位被置1。 0: 不产生通道1捕获或比较事件 1: 发生通道1捕获或比较事件
0	UPG	此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0: 无更新事件产生 1: 产生更新事件

### TIMERx 通道控制寄存器 1 (TIMERx\_CHCTRL1)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CH1OCE	CH1OM[2:0]		CH1OSE	CH1OFE	CH1M[1:0]		
								CH1ICF[3:0]		CH1ICP[1:0]					
								RW	RW	RW	RW	RW	RW	RW	RW

## 输出比较模式:

位/位域	名称	描述
15:8	保留	必须保持复位值.
7	CH1OCE	<p>通道1输出比较清0使能</p> <p>当此位被置1, 当检测到ETIF输入高电平时, OC1参考信号被清0</p> <p>0: 禁止通道1输出比较清零</p> <p>1: 使能通道1输出比较清零</p>
6:4	CH1OM[2:0]	<p>通道1输出比较模式</p> <p>此位定义了输出参考信号OC1REF的动作, 而OC1REF决定了OC1的值。OC1REF高电平有效, 而OC1的有效电平取决于CH1P位。</p> <p>000: 冻结。输出比较寄存器TIMERx_CHCC1与计数器间的比较对OC1REF不起作用</p> <p>001: 匹配时设置为高。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时, 强制OC1REF为高。</p> <p>010: 匹配时设置为低。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时, 强制OC1REF为低。</p> <p>011: 匹配时翻转。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时, 强制OC1REF翻转。</p> <p>100: 强制为低。强制OC1REF为低电平</p> <p>101: 强制为高。强制OC1REF为高电平</p> <p>110: PWM 模式1。在向上计数时, 一旦计数器值小于TIMERx_CHCC1时, OC1REF为高电平, 否则为低电平。在向下计数时, 一旦计数器的值大于TIMERx_CHCC1时, OC1REF 为低电平, 否则为高电平。</p> <p>111: PWM 模式2。在向上计数时, 一旦计数器值小于TIMERx_CHCC1时, OC1REF为低电平, 否则为高电平。在向下计数时, 一旦计数器的值大于TIMERx_CHCC1时, OC1REF 为高电平, 否则为低电平。</p> <p>在PWM模式1或PWM模式2中, 只有当比较结果改变了或者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>
3	CH1OSE	<p>通道1输出比较影子寄存器使能</p> <p>当此位被置1, TIMERx_CHCC1寄存器的影子寄存器被使能, 影子寄存器在每次更新事件时都会被更新。</p> <p>0: 禁止通道1输出/比较影子寄存器</p> <p>1: 使能通道1输出/比较影子寄存器</p> <p>仅在单脉冲模式下(TIMERx_CTLR1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</p>
2	CH1OFE	<p>通道1输出比较快速使能</p> <p>该位用于通道配置为PWM模式时加快CC输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配, OC 被设置为比较电平而与比较结果无关</p> <p>0: 禁止通道1输出比较快速. 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: 使能通道1输出比较快速。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3个时钟周期</p>

---

1:0	CH1M[1:0]	通道1模式选择
		这些位定义了输入的方向和输入信号的选择。只有当通道关闭(TIMERx_CHE寄存器的CH1E位被清0)时这些位才可写。
	00:	通道1配置为输出
	01:	通道1配置为输入， IC1映射在TI1上
	10:	通道1配置为输入， IC1映射在TI2上
	11:	通道1配置为输入， IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMC寄存器的TRGS位)。
<b>输入捕获模式:</b>		
位/位域	名称	描述
15:8	保留	必须保持复位值
7:4	CH1ICF[3:0]	通道1输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。 这些位定义了TI1输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP}=f_{DTS}$ , N=1 0001: $f_{SAMP}=f_{PCLK}$ , N=2 0010: $f_{SAMP}=f_{PCLK}$ , N=4 0011: $f_{SAMP}=f_{PCLK}$ , N=8 0100: $f_{SAMP}=f_{DTS}/2$ , N=6 0101: $f_{SAMP}=f_{DTS}/2$ , N=8 0110: $f_{SAMP}=f_{DTS}/4$ , N=6 0111: $f_{SAMP}=f_{DTS}/4$ , N=8 1000: $f_{SAMP}=f_{DTS}/8$ , N=6 1001: $f_{SAMP}=f_{DTS}/8$ , N=8 1010: $f_{SAMP}=f_{DTS}/16$ , N=5 1011: $f_{SAMP}=f_{DTS}/16$ , N=6 1100: $f_{SAMP}=f_{DTS}/16$ , N=8 1101: $f_{SAMP}=f_{DTS}/32$ , N=5 1110: $f_{SAMP}=f_{DTS}/32$ , N=6 1111: $f_{SAMP}=f_{DTS}/32$ , N=8
3:2	CH1ICP[1:0]	通道1输入捕获预分频器 这2位定义了CC1 输入的预分频系数。一旦CH1E =0 (TIMER14_CHE寄存器中)，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH1M[1:0]	通道1模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH1E位被清0)时这些位才可写。

- 00: 通道1配置为输出
- 01: 通道1配置为输入, IC1映射在TI1上
- 10: 通道1配置为输入, IC1映射在TI2上
- 11: 通道1配置为输入, IC1映射在TRC上. 此模式仅工作在内部触发输入被选中时(通过设置TIMER\_SMC寄存器的TRGS位).

### **TIMERx 通道使能寄存器 (TIMERx\_CHE)**

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH1NP	保留.
rw														CH1P	CH1E

位/位域	名称	描述
15:4	保留	必须保持复位值
3	CH1NP	通道1互补输出极性 当通道1配置为输入模式, 此位定义了互补输出信号的极性。 0: 通道1高电平有效 1: 通道1低电平有效
2	保留	必须保持复位值
1	CH1P	通道1极性 当通道1配置为输入模式时, 此位定义了IC1信号极性。当通道1配置为输出模式时, 此位定义了输出信号极性。 0: 通道1高电平有效 1: 通道1低电平有效 当TIMERx_BKDT寄存器的LK[1:0]=11或10时此位不能被更改。
0	CH1E	通道1使能 当通道1配置为输入模式时, 将此位置1使能OC1信号有效。当通道1配置为输出模式时, 将此位置1使能通道1上的捕获事件。 0: 禁止通道1 1: 使能通道1

### **TIMERx 计数器寄存器 (TIMERx\_CNT)**

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

CNT[15:0]
-----------

rw

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

### TIMERx 预分频寄存器 (TIMERx\_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

### TIMERx 计数器自动重载寄存器 (TIMERx\_CARL)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

### TIMERx 通道 1 捕获/比较寄存器 (TIMERx\_CHCC1)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CHCC1[15:0]																

rw

位/位域	名称	描述
15:0	CHCC1[15:0]	<p>通道1的捕获或比较值</p> <p>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

### TIMERx 通道输入重映射寄存器 (TIMERx\_RMP)

地址偏移: 0x50

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH1RMP[1:0]	

rw

位/位域	名称	描述
15:2	保留	必须保持复位值.
1:0	CH1RMP[1:0]	<p>通道1输入重映射</p> <p>00: 通道1输入连接到GPIO(TIMER14_CH1_I)</p> <p>01: 通道1输入连接到RTCCLK</p> <p>10: 通道1输入连接到HSE/32时钟</p> <p>11: 通道1输入连接到CKOUTSEL(受RCC_GCFGR控制)</p>

### TIMERx 配置寄存器 (TIMERx\_CFG) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
15:2	保留	必须保持复位值.
1	CCSEL	写捕获比较寄存器选择位

此位由软件写1或清0

1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效

0: 无效

0 保留 必须保持复位值

## 9.5. 通用定时器 (TIMER15)

### 9.5.1. 简介

通用定时器 (TIMER15) 可以用于多种用途，它包括一个16位的向上计数器、2个捕捉/比较寄存器(TIMERx\_CHCC)、1个16位的自动装载计数器寄存器(TIMERx\_CARL)和一些控制寄存器。它适合多种用途，包括通用定时器，输入信号脉冲宽度测量或者产生输出波形例如单脉冲输出或PWM输出。

通用 (TIMER15) 定时器完全独立，它们不共享资源但是可以同步。

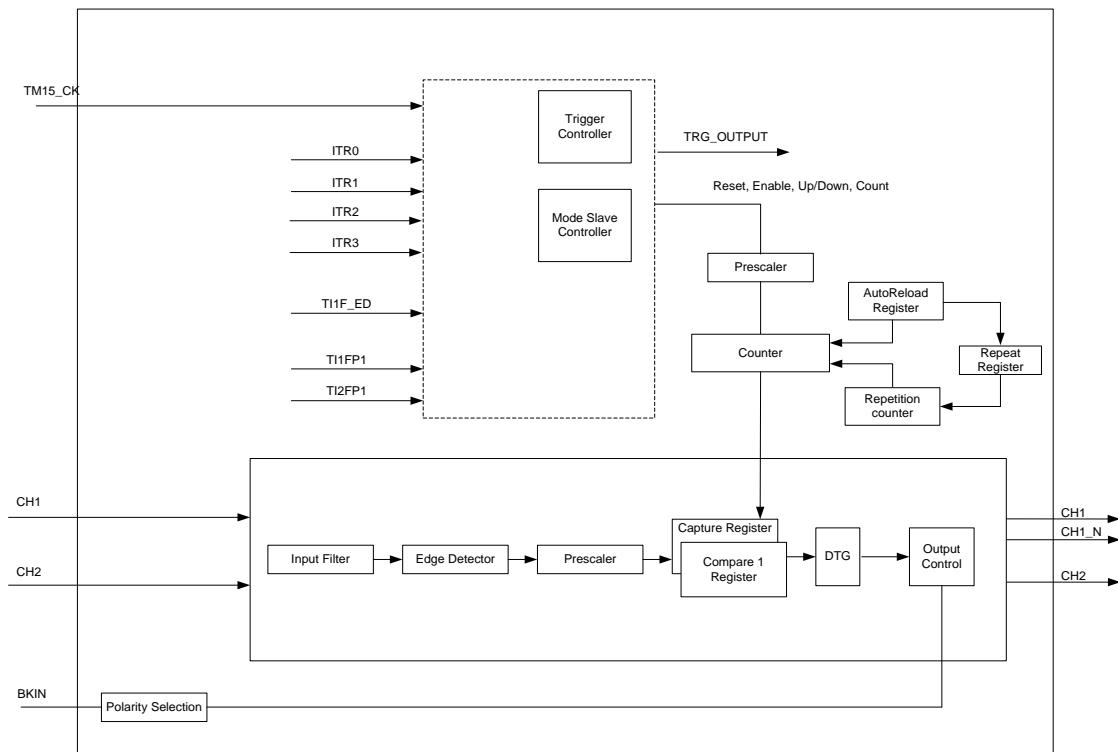
### 9.5.2. 主要特性

- 16位向上自动重载计数器
- 16位可编程的预分频器，计数器时钟频率的分频系数为1~65536之间的任意数值。
- 2个独立通道，支持以下功能：输入捕获，输出比较，PWM波形产生(边缘或中间对齐模式)和单脉冲模式输出
- 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
- 中止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
- 更新事件，输入捕获事件，输出比较事件，中止输入，触发事件可以产生中断/DMA
- 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
- TIMER15 主从模式控制器

### 9.5.3. 功能描述

下图提供了高级定时器内部配置细节

图 9-100. 通用定时器框图 (TIMER15)



### 预分频计数器

预分频器可以将定时器(PCLK)的时钟频率按1到65536之间的任意值分频给计数器(CNT\_CLK)使用。分频系数受预分频寄存器TIMERx\_PSC控制，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 9-101. 当预分频器的参数从 1 变到 2 时，计数器时序图

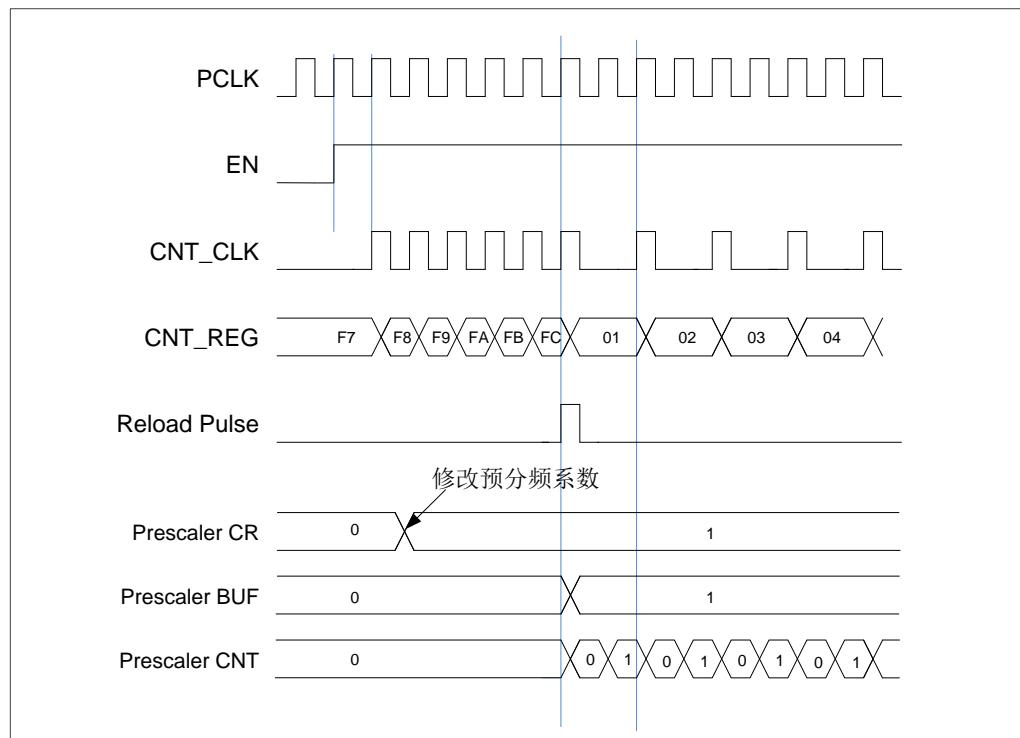
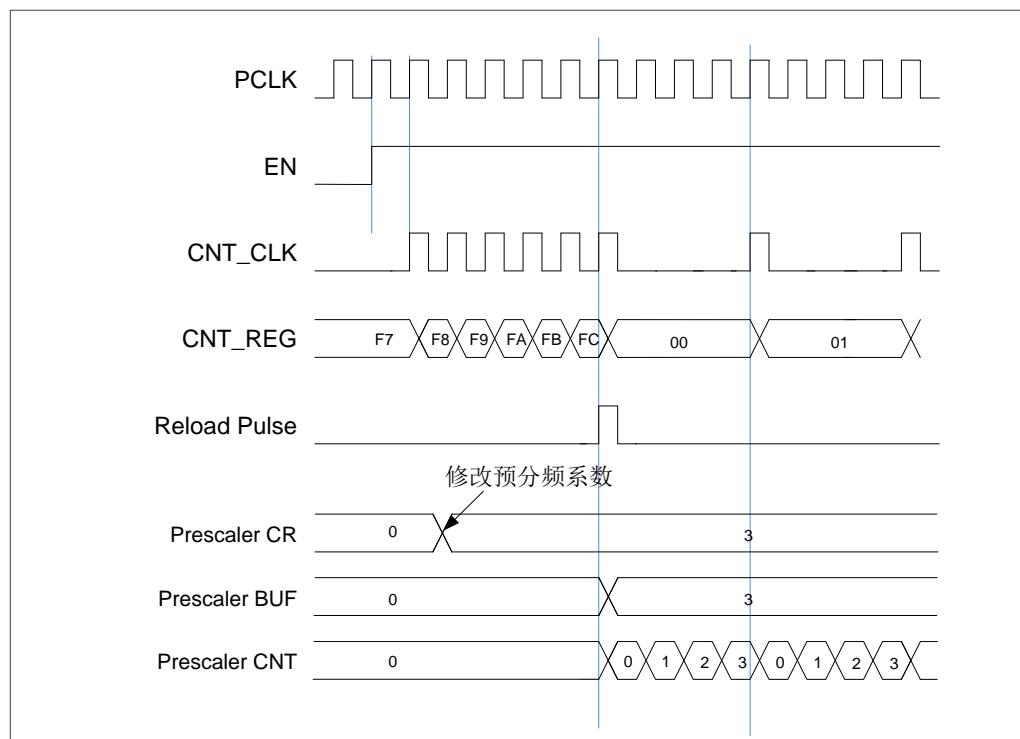


图 9-102. 当预分频器的参数从 1 变到 4 时，计数器时序图



### 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(**TIMERx\_CARL**寄存器的内容)，然后重新从0开始计数。如果使用了重复计数器功能吗，在向上计数达到设置的重复计数次数加1(**TIMERx\_CREP+1**)时，才产生更新事件，否则每次计数器溢出时都会产生更新事件。

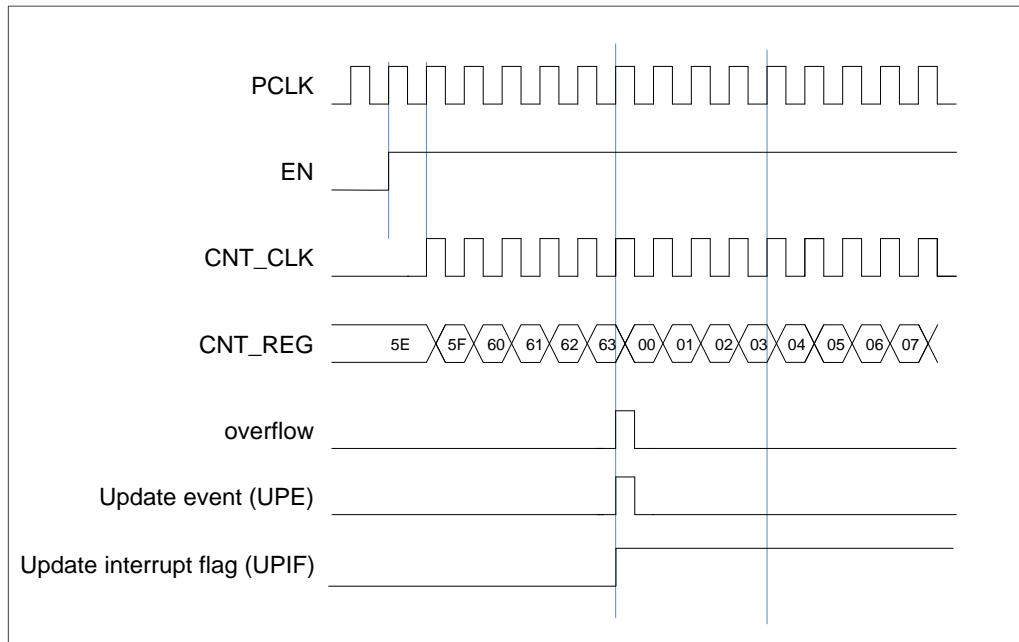
可以将TIMERx\_EVG寄存器的UPG位置1来设置更新事件，计数值被清0产生更新事件。

设置TIMERx\_CTLR1寄存器的UPDIS位可以禁止更新事件。

一旦发生更新事件，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作。

**图 9-103. 计数器时序图，内部时钟分频系数为 1**



**图 9-104. 计数器时序图，内部时钟分频系数为 2**

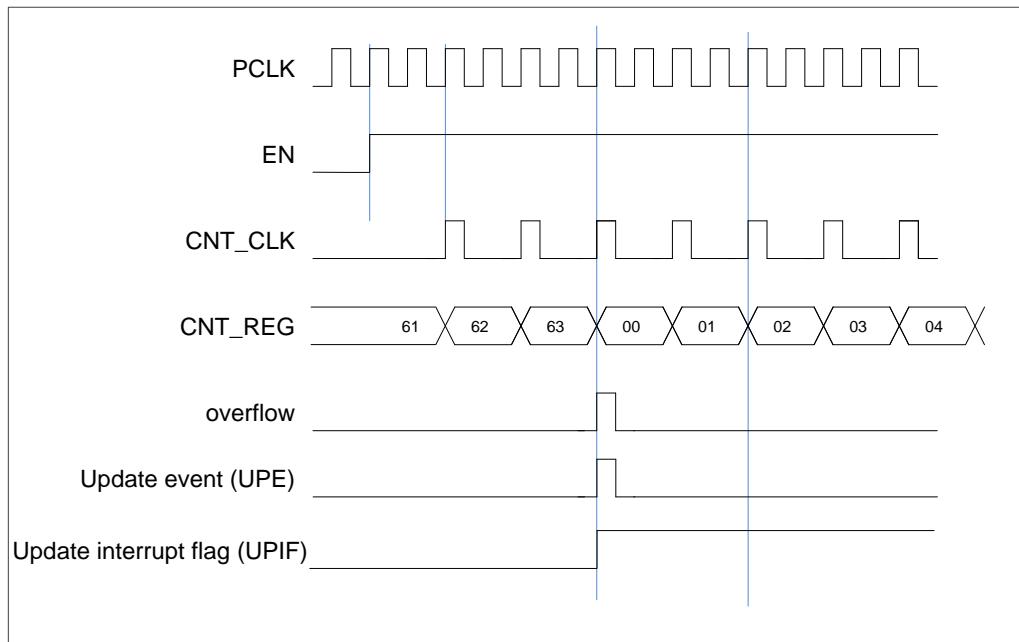


图 9-105. 计数器时序图，内部时钟分频系数为 4

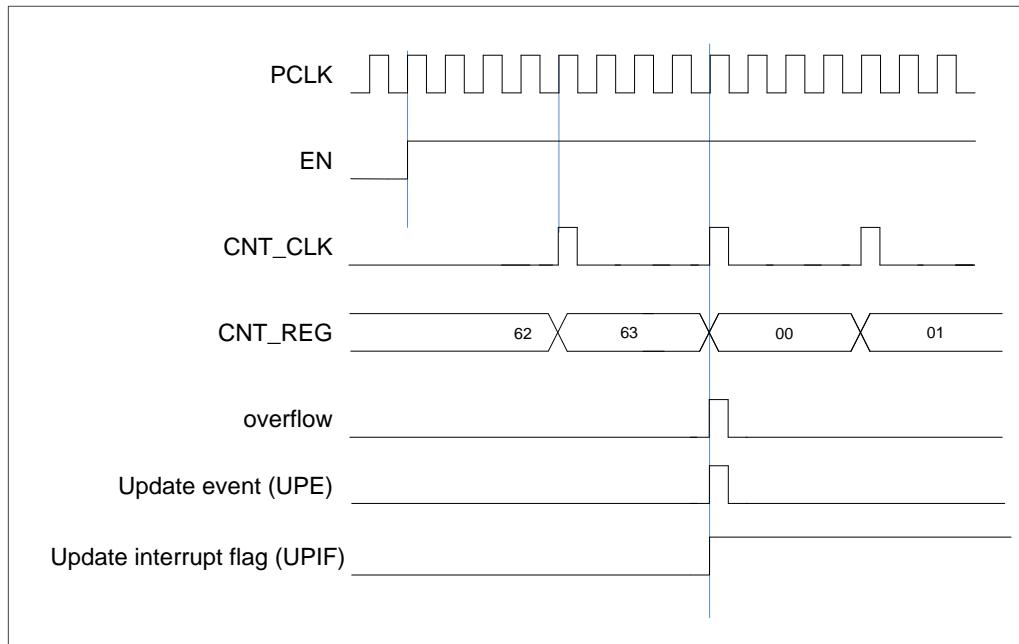


图 9-106. 计数器时序图，内部时钟分频系数为 N

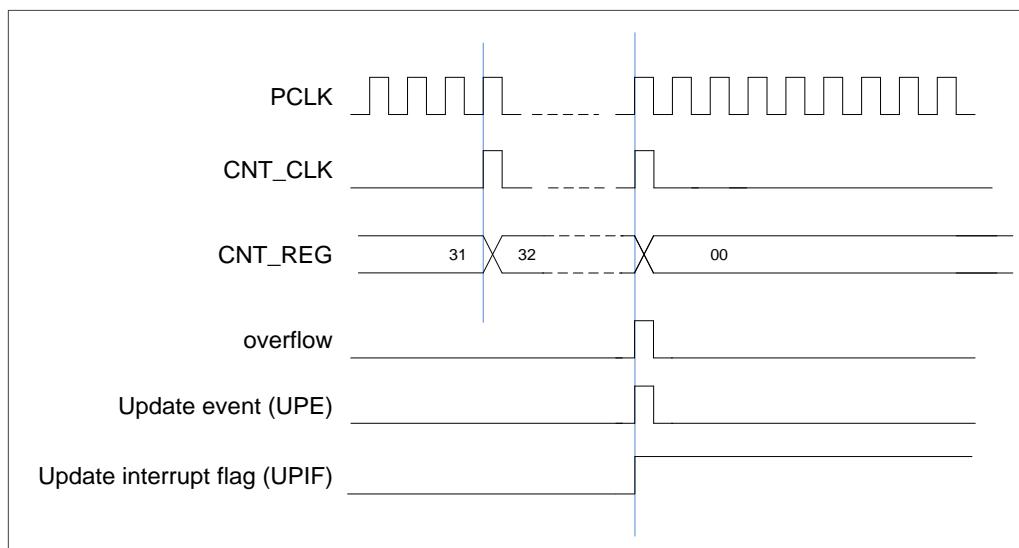


图 9-107. 计数器时序图, 当 ARSE=0 时更新事件

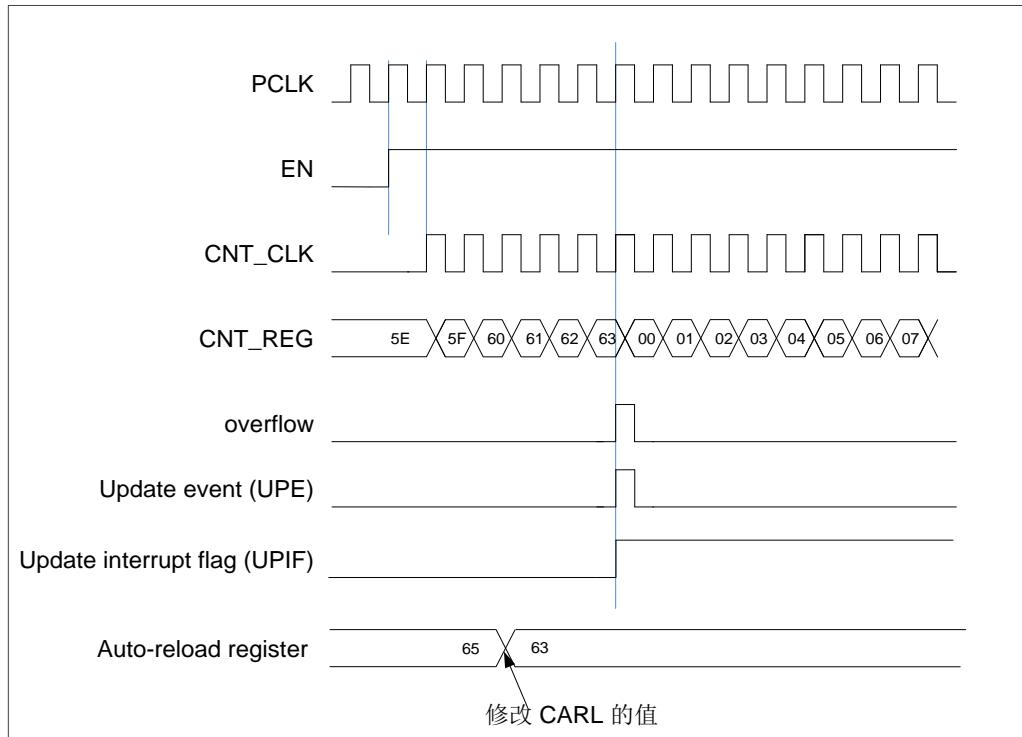
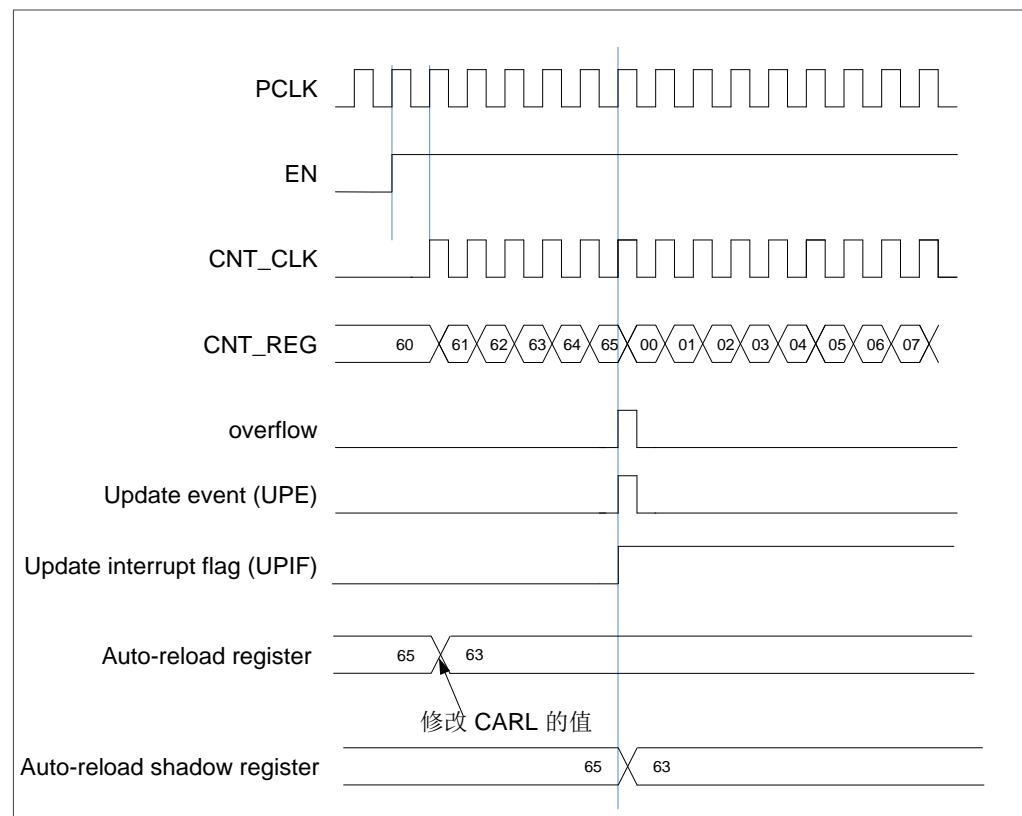


图 9-108. 计数器时序图, 当 ARSE=1 时更新事件



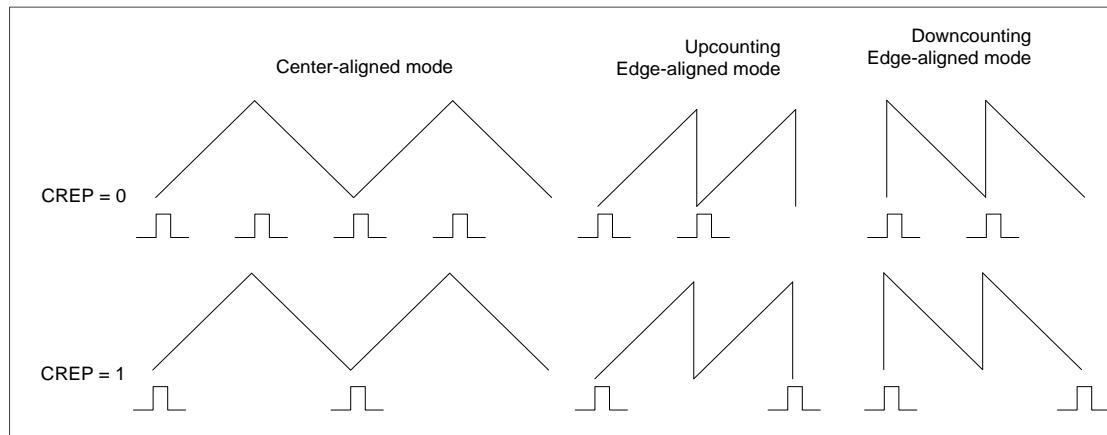
## 重复计数器

仅仅在一个给定数字N+1(N为TIMERx\_CREP寄存器的CREP)个循环周期的计数后，重复计数器被用来产生更新事件或者更新定时器寄存器。向上计数模式下，重复计数器在每次计数器上溢时递减；向下计数模式下，重复计数器在每次计数器下溢时递减；在中央对齐模式下，重复计数器在计数器上溢和下溢时递减。

将TIMERx\_EVG寄存器的UPG位置1可以重载TIMERx\_CREP寄存器中CREP的值并产生一个更新事件。

**图 9-109. 不同模式及不同 TIMERx\_CREP 寄存器设置下更新速率的例子**

TIMER15 只有向上计数边沿对齐模式。



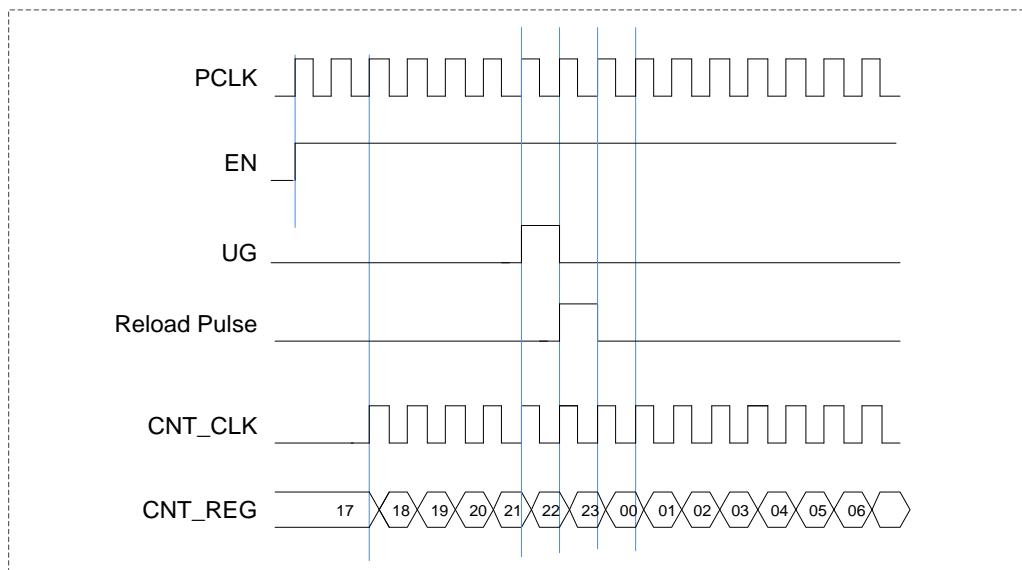
## 时钟选择

决定内部预分频计数器时钟源的定时器模块时钟控制器的描述如下。

### ■ 内部定时器时钟PCLK

如果禁止了从模式，默认用来驱动计数器预分频器的内部时钟源是APB2时钟CK\_APB2。如果使能从模式控制器(将TIMERx\_SMC寄存器的SMC区域设置为包括0x1、0x2、0x3和0x7)，预分频器被其他时钟源(由TIMERx\_SMC寄存器的TRGS区域选择，说明如下)锁定。当从模式选择位SMC被设置为0x4、0x5或0x6，计数器预分频器时钟源由内部时钟PCLK提供。

图 9-110. 内部时钟分频为 1 时正常模式下的控制电路



#### ■ 内部触发输入(ITI)

计数器预分频器能在ITI信号的每个上升沿和下降沿计数。此模式可以通过设置TIMERx\_SMC寄存器的SMC区域为0x6来选择，此时计数器将作为一个事件计数器。输入事件ITI能通过设置TRGS区域选择。当ITI信号被选为时钟源，内部边沿监测电路将在每个ITI信号上升沿或下降沿产生一个时钟脉冲来驱动计数器预分频器。

#### ■ 外部输入引脚(TIx)

外部引脚TIMERx\_TIx信号的上升沿和下降沿能驱动计数器预分频器计数。此模式可以通过设置SMC区域为0x7及设置TRGS区域为0x4、0x5或0x6来选择。注意TIx源于数字滤波器对TIMERx\_TIx采样。

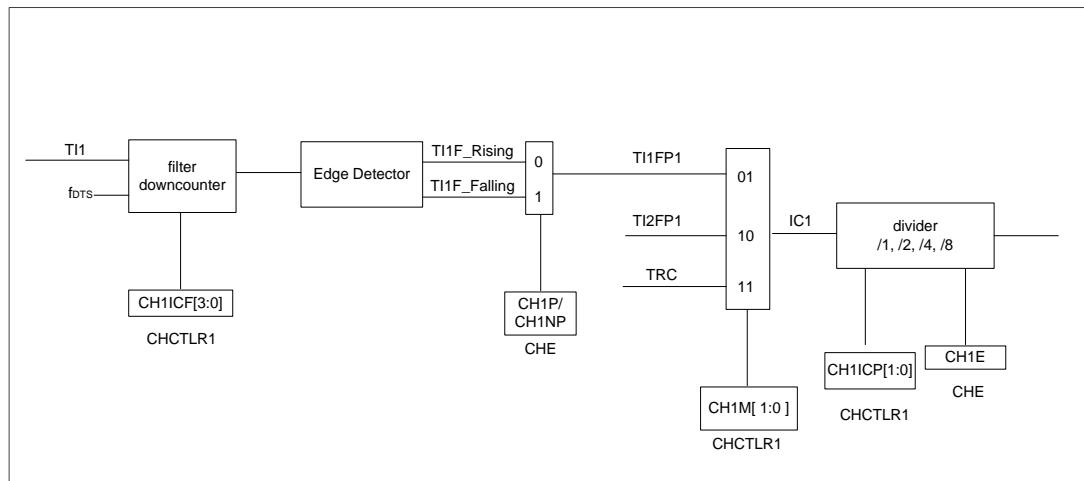
### 捕获/比较通道

TIMER15拥有2个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

#### ■ 输入捕获级

输入级包括一个数字滤波器，一个通道极性选择，边沿监测和一个通道预分频器。通道输入信号(TIx)被一个数字滤波器采样来产生一个滤波输入信号TIxF。然后通道极性和边沿监测模块产生一个TI1FP1或TI2FP1信号，来实现输入捕捉功能。通道输入预分频寄存器(CHxICP)可以设置有效输入事件数目。

图 9-111. 捕获/比较通道(例如：通道 1 输入级)



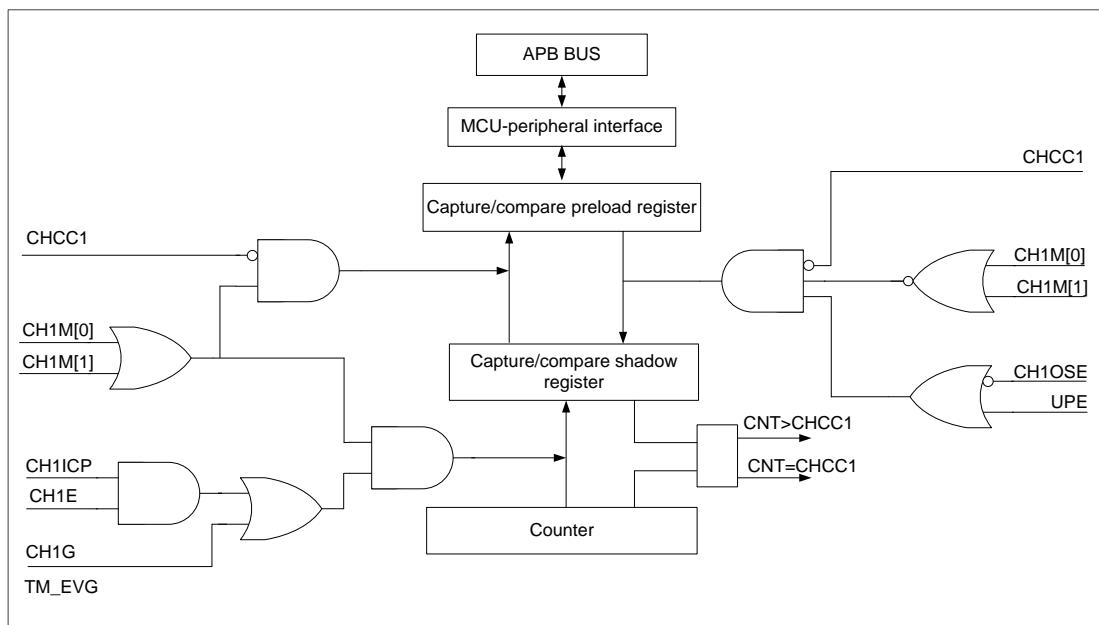
### ■ 通道控制器

TIMER15有2个独立的通道用来捕获输入或输出匹配比较

用于输入捕获模式下，当捕获事件发生时，**TIMERx\_CHCCx**影子寄存器首先捕获计数值然后传送到**TIMERx\_CHCCx**预装载寄存器中。

用于输出匹配比较模式下，**TIMERx\_CHCCx**预装载寄存器的值被复制到相应的影子寄存器中，然后计数值和影子寄存器的值比较。

图 9-112. 捕获/比较通道 1 主电路



### ■ 输出级

TIMER15有2个用于匹配比较、单脉冲和PWM输出功能的通道。

图 9-113. 捕获/比较通道输出级(通道 1)

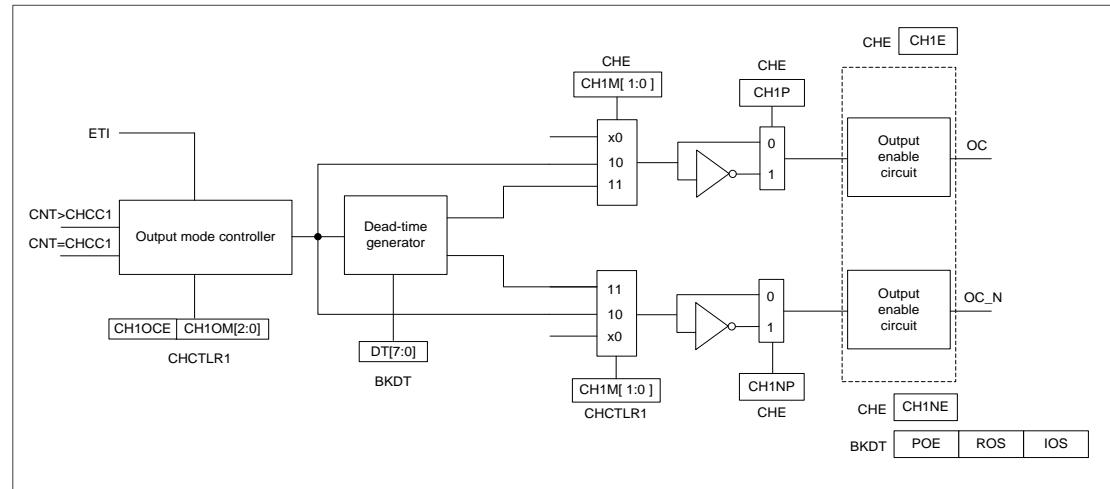
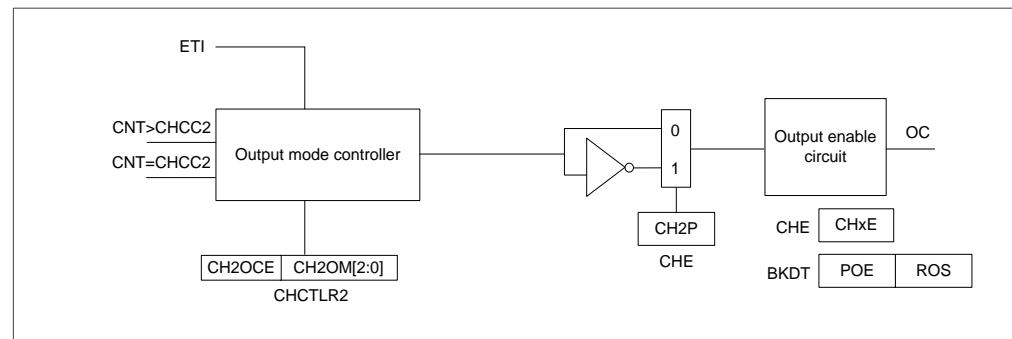


图 9-114. 捕获/比较通道输出级(通道 2)



### 输入捕获模式

通道用于输入捕获时，当一个有效输入信号的边沿跳变时，计数值被捕获到捕获/比较寄存器(TIMERx\_CHCCx)中。一旦捕获事件发生，TIMERx\_STR寄存器的CHxIF标志位被置1。如果CHxIF位已经被置1，还没来得及被软件清除，此通道上又有另一个捕获事件发生，相应通道捕获溢出标志位CHxOF将被置1。一旦捕获事件发生，CHxDE位决定了是否产生DMA请求，CHxIE位决定了是否产生中断。使用输入捕获模式需按以下步骤配置：

1. 配置通道控制寄存器(TIMERx\_CHCTRLx)的CCxM位域，选择输入信号
2. 配置通道控制寄存器(TIMERx\_CHCTRLx)的CHxICF位域，为输入信号添加输入滤波器
3. 配置通道使能寄存器(TIMERx\_CHE)的CHxP/CHxNP位，选择输入捕获的边沿（上升沿或下降沿）
4. 如果输入信号需要分频，配置通道控制寄存器(TIMERx\_CHCTRLx)的CHxICP位域
5. 如果需要中断，将DMA中断使能寄存器(TIMERx\_DIE)的CHxIE位置位
6. 如果需要DMA请求，将DMA中断使能寄存器(TIMERx\_DIE)的CHxDE位置位

## 7. 将通道使能寄存器(TIMERx\_CHE)的CCxE位置位，使能通道

按上述配置完寄存器后，该通道上的对应边沿信号将产生输入捕获事件。计数值被捕获到通道捕获/比较寄存器(TIMERx\_CHCCx)中。如果CHxE位被置位，将会产生中断。如果CHxDE位被置位，将会产生DMA请求。

输入捕获模式也可以用于TIMERx\_CHx(TIx)引脚的信号脉宽测量。例如，PWM信号连接到TI1输入。配置TIMERx\_CHCTRLRx寄存器的CH1M位域为01，将通道1输入捕获信号连接到TI1，并设置上升沿捕获。配置TIMERx\_CHCTRLRx寄存器的CH2M位域为10，选择通道2捕获信号连接到TI1，并设置下降沿捕获。计数器配置为通道1上升沿触发后重新计数模式。此时TIMERx\_CHCC1可以测量PWM信号的周期，TIMERx\_CHCC2可以测量PWM信号的占空比。

## 输出比较模式/PWM 模式

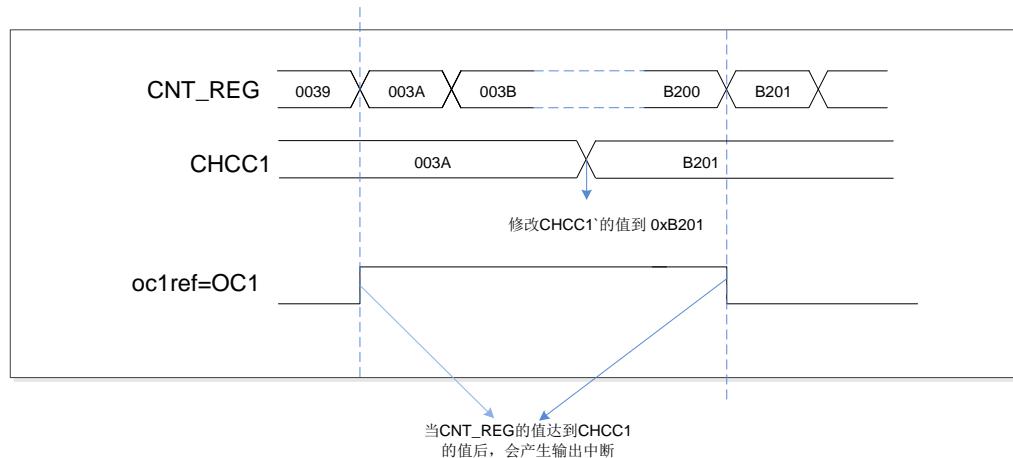
当配置通道控制寄存器(TIMERx\_CHCTRLRx)的CCxM位域为00，使通道工作于输出模式时，通道输出的波形由TIMERx\_CHCTRLRx的CHxOM位域和计数器与TIMERx\_CHCCx寄存器的值比较的结果决定。当比较的值相等时状态寄存器(TIMERx\_STR)的CHxIF位将置1。根据CHxDE或CHxE位的值，一次DMA请求或中断将被产生。

在输出模式，配置CHxOM位域为101，通道将持续输出有效电平。配置CHxOM位域为100通道将持续输出无效电平。

在输出模式，配置CHxOM位域为001，当计数器与TIMERx\_CHCCx寄存器的值相等时，通道输出有效电平。配置CHxOM位域为010，当计数器与TIMERx\_CHCCx寄存器的值相等时，通道输出无效电平。

在输出比较翻转模式（配置CHxOM位域为011），当计数器与TIMERx\_CHCCx寄存器的值相等时，通道输出电平将翻转。

图 9-115. 输出比较模式，在OC1 上切换



在PWM输出模式（配置CHxOM位域为110（PWM模式1）或111（PWM模式2））。根据TIMERx\_CARL寄存器和TIMERx\_CHCCx寄存器的值，通道输出PWM波形。

图 9-116. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式

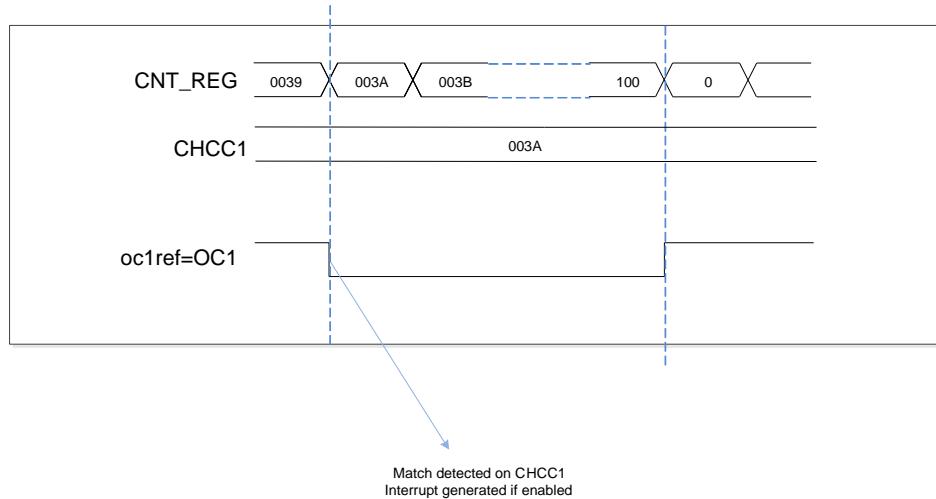
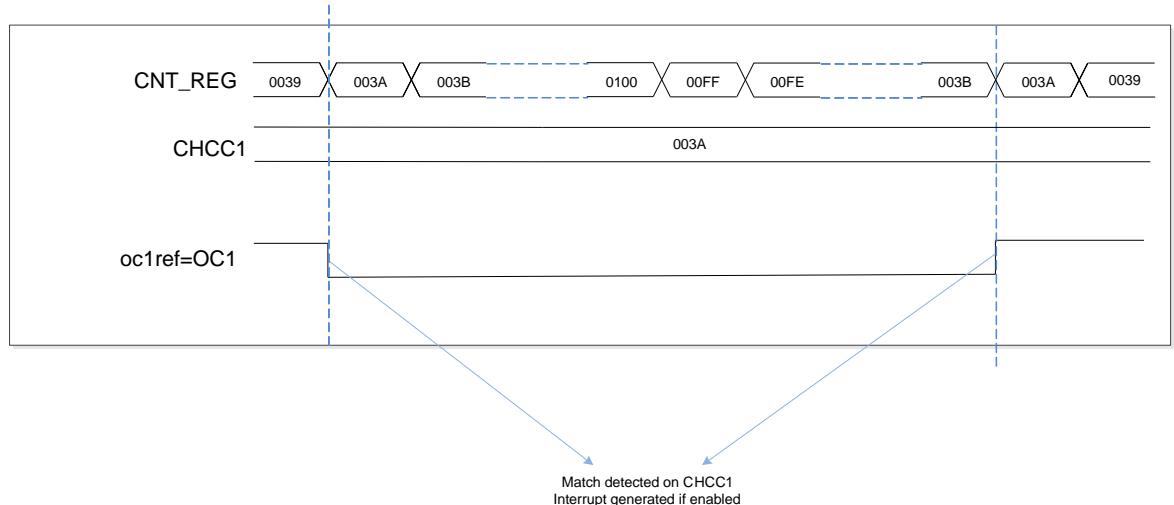


图 9-117. 在 OC1 上输出比较 PWM 模式 1, 中央对齐计数模式



### 通道输出参考信号

当TIMER15用于输出匹配比较模式下，设置CHxOM位可以定义OCxREF信号(通道x输出参考信号)类型。OCxREF信号有一些输出功能类型，包括，设置CHxOM=0x00可以保持原始电平，设置CHxOM=0x01可以将OCxREF信号设置为高电平，设置CHxOM=0x02可以将OCxREF信号设置为低电平，在计数值和TIMERx\_CHCCx寄存器的值匹配时，设置CHxOM=0x03可以切换信号。

PWM模式1和PWM模式2输出也是另一种类型的OCxREF输出，具体设置是通过设置CHxOM =0x06或0x07来完成的。在这些模式中，OCxREF信号级别的改变取决于根据计数方向，以及TIMERx\_CHCCx寄存器值和计数值之间的关系。详情参考相应位定义。

设置CHxOM=0x04或0x05可以实现OCxREF信号的强制输出功能。输出比较信号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。

当ETIF信号源于外部TIMERx\_ ETI引脚且被设置为高电平(设置TIMERx\_CHCTRL1寄存器的

CHxOCE位为1)时，OCxREF信号被强制变为低电平。直到下一次更新事件发生，OCxREF信号才会回到它的有效状态。

### 互补输出及死区时间

TIMER15可以输出两路互补信号。互补信号OCx和OCxN的生效由一些控制位的组合实现：寄存器TIMERx\_CHE的CHxE位和CHxNE位，寄存器TIMERx\_BKDT和TIMERx\_CTRL2的POE, ISOx, ISOxN, IOS和ROS位。输出极性由TIMERx\_CHE寄存器的CHxP位和CHxNP位决定。如果CHxE, CHxNE和POE位都为1，应该插入死区时间。OCx输出的上升沿和OCxREF的上升沿相比有一个延迟；OCxN输出的上升沿和OCxREF的下降沿相比有一个延迟。延迟的值是一个8位的死区时间计数器，由TIMER15\_BKDT寄存器的DT区域决定。如果延迟值大于有效输出(OCx或OCxN)宽度，将不会产生相应信号脉冲。

图 9-118. 带有死区时间的互补输出

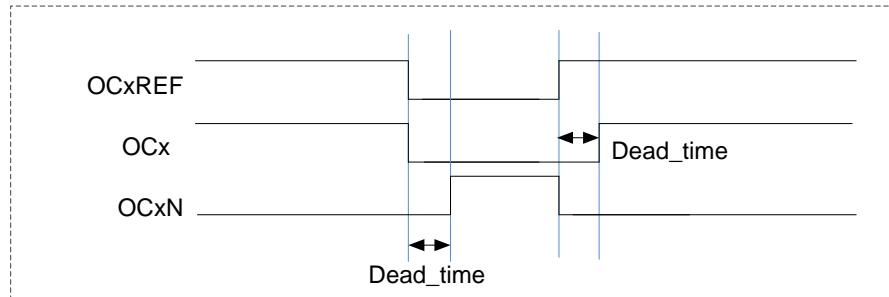


图 9-119. 延迟大于负脉冲的死区时间波形

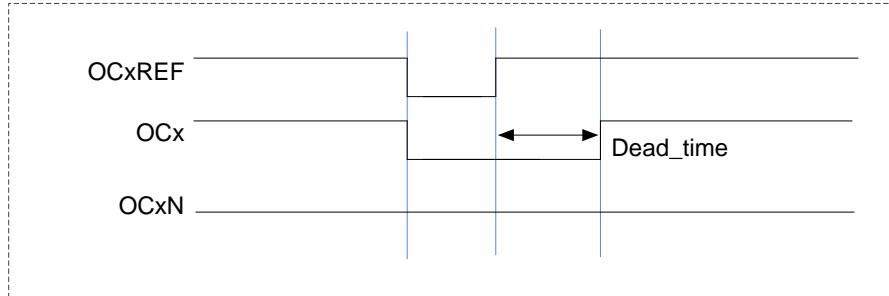
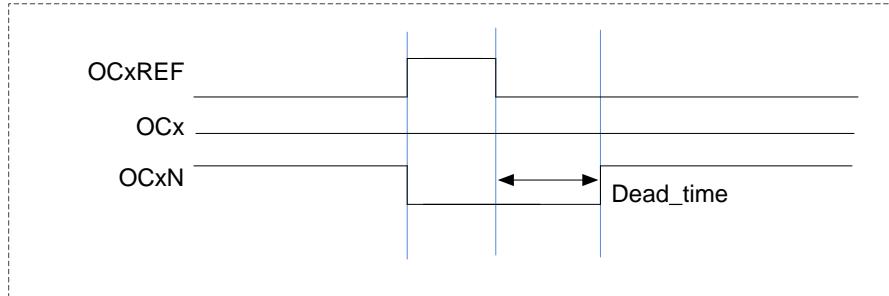


图 9-120. 延迟大于正脉冲的死区时间波形



### 中止功能

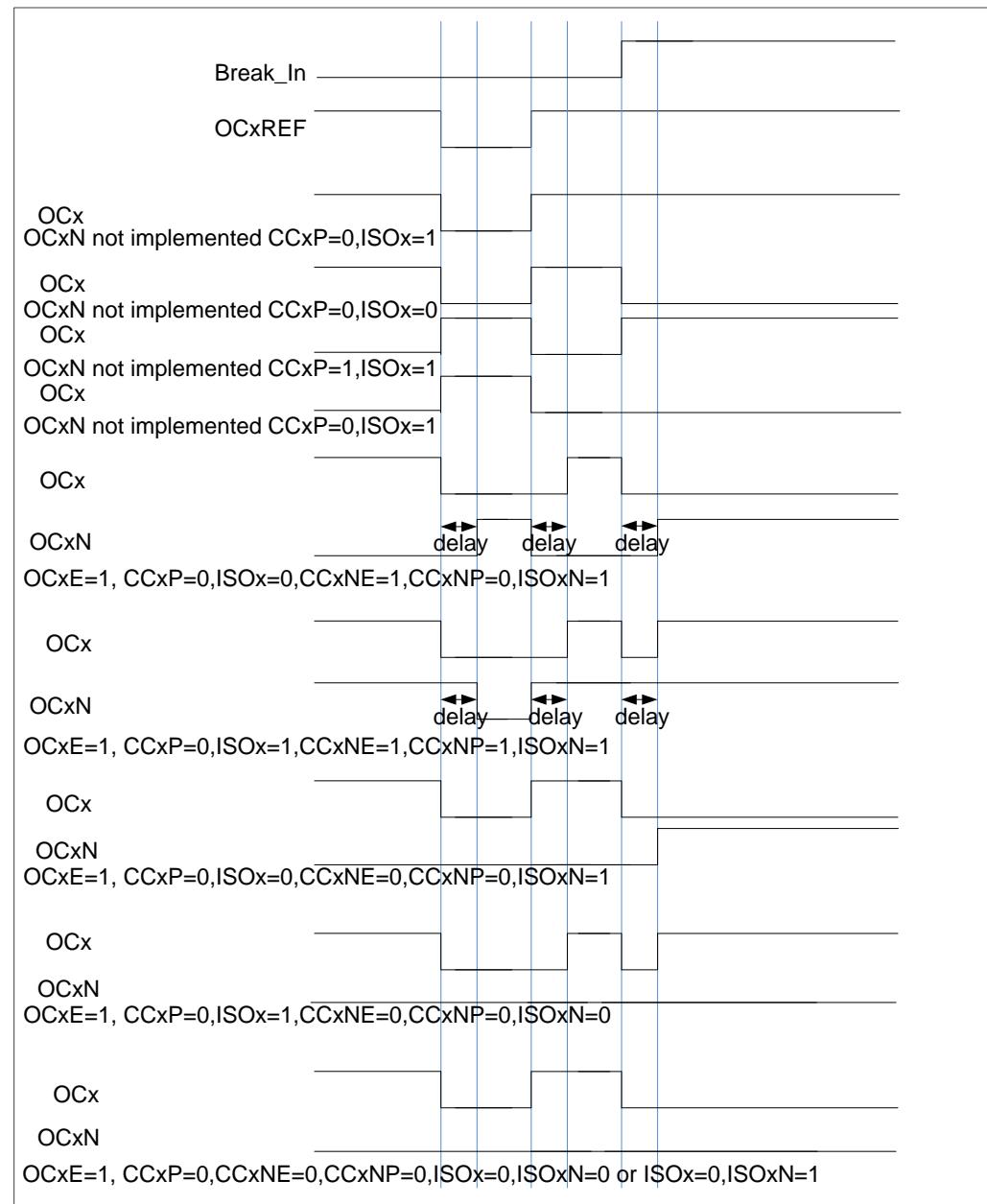
使用中止功能时，输出OCx和OCxN信号电平被以下位控制，TIMERx\_BKDT寄存器的POE, IOS和ROS 位，TIMERx\_CTRL2寄存器的ISOx和ISOxN位。中止事件发生时，OCx和OCxN

信号输出不能在同一时间同时处于有效电平上。中止源既可以是中止输入引脚又可以是一个HSE时钟失败事件。时钟失败事件由RCC中的时钟监视器(CKM)产生。将TIMERx\_BKDT寄存器的BRKE位置1可以使能中止功能。TIMERx\_BKDT寄存器的BRKP位决定了中止输入极性。

发生中止时，POE位被异步地清除，一旦POE位为0，配置TIMERx\_CTLR2寄存器的ISOx位驱动输出OCx和OCxN的电平。如果IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，TIMERx\_STR寄存器的BKIF位被置1。如果BKIE=1，中断产生。如果BKDE=1，发送一个DMA请求。

图 9-121. 响应中止的输出表现（中止输入高电平有效）

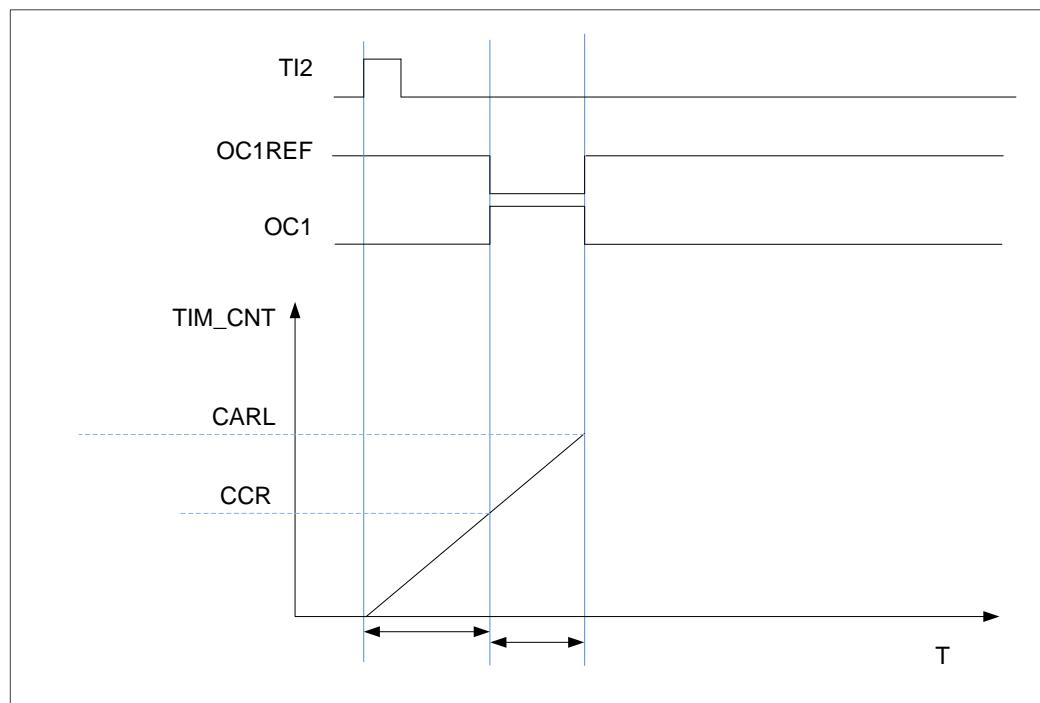


### 单脉冲模式

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx\_CTLR1寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

单脉冲模式下，将CEN位置1产生的触发有效边沿将使能计数器。然而，执行计数值和TIMERx\_CHCCx寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx\_CHCTRL1寄存器的CHxOEF位置1。单脉冲模式下，触发上升沿产生之后，OCxREF信号将被立即强制转换为另一种状态，因为不考虑比较结果的比较匹配事件发生。只有输出通道配置为PWM1或PWM2输出运行模式下时 CHxOEF位才可用，触发源来源于触发信号。

图 9-122. 单脉冲模式



### 从控制器

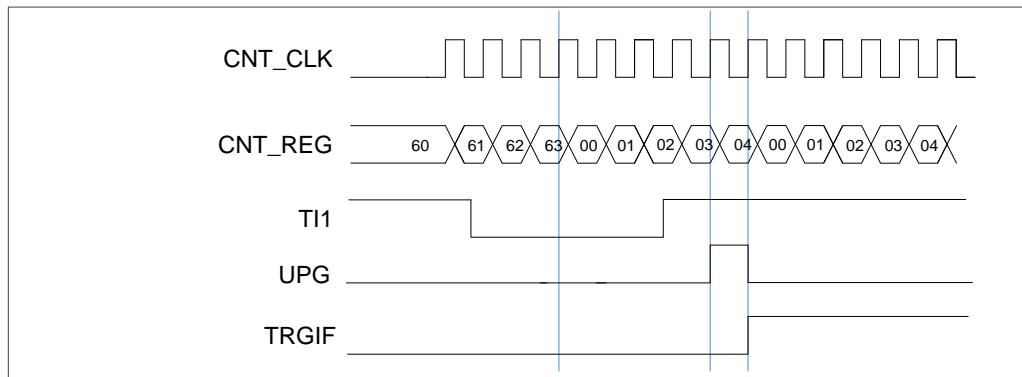
TIMER15能够在多种模式下和一个外部的触发同步，包括复位模式，暂停模式和触发模式。模式选择通过设置TIMERx\_SMC寄存器的SMC区域来完成。这些模式的触发输入可以通过TIMERx\_SMC寄存器的TRGS区域来选择，下面以TI1信号为例。从控制器运行模式参见附加章节。

- 复位模式

在TI1信号的上升沿来临时，计数器和它的预分频器能够重新被初始化，此时更新事件生成寄存器位UPG将自动被硬件声明，触发事件标志位被置1，然后计数器和它的预分频器被重新初始化。尽管UPG位被硬件置1，更新事件并不真的发生，它依赖于更新事件禁止控制位UPDIS被置1与否。如果UPDIS被置1，禁止更新事件发生，将不产生更新事件，然而当TI1上升沿来

临时计数器和它的预分频器仍然被重新初始化。如果TIMERx\_CTRL1寄存器的UPDIS位被清0，使能更新事件发生，当TI1上升沿来临时更新事件将发生。所有的预加载寄存器将被更新。

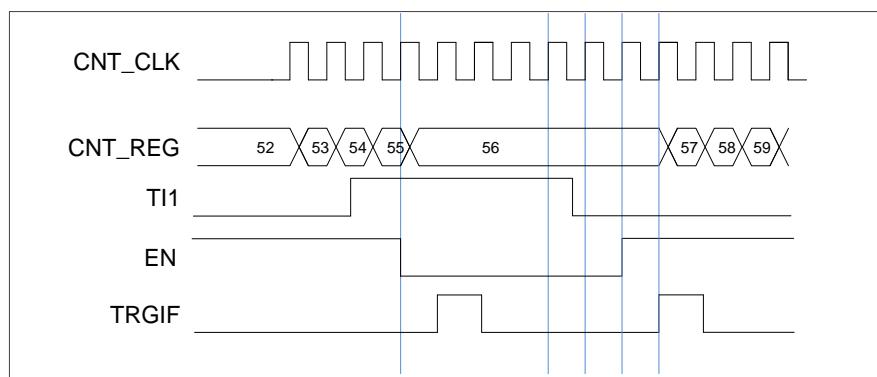
图 9-123. 复位模式的控制电路



#### ■ 暂停模式

暂停模式下，选择的TI1输入信号电平用来控制计数器开启/停止运行。当选择的TI1信号为高电平时计数器开始计数，当TI1信号变为低电平时停止计数。此处计数器将保持它的当前值且不再被重启。

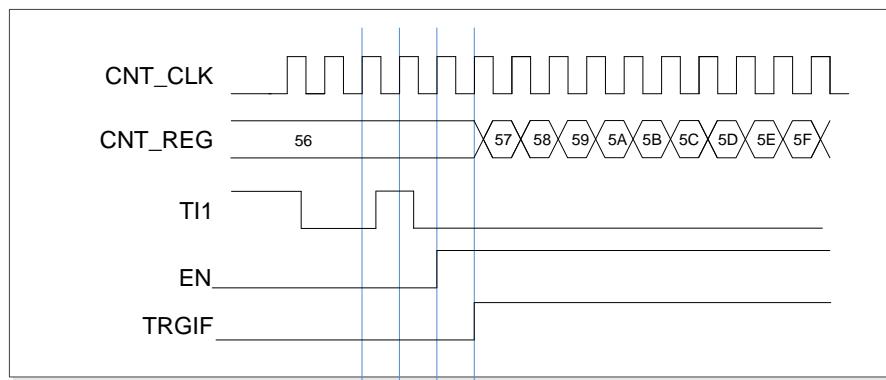
图 9-124. 暂停模式下的控制电路



#### ■ 触发模式

计数器被禁止计数之后，当TI1上升沿来临时计数器可以重新开始计数。当一个TI1上升沿来临，计数器从当前值开始计数。注意TI1信号仅用来使能计数器重新计数，不能控制计数器停止计数。

图 9-125. 触发模式下的控制电路

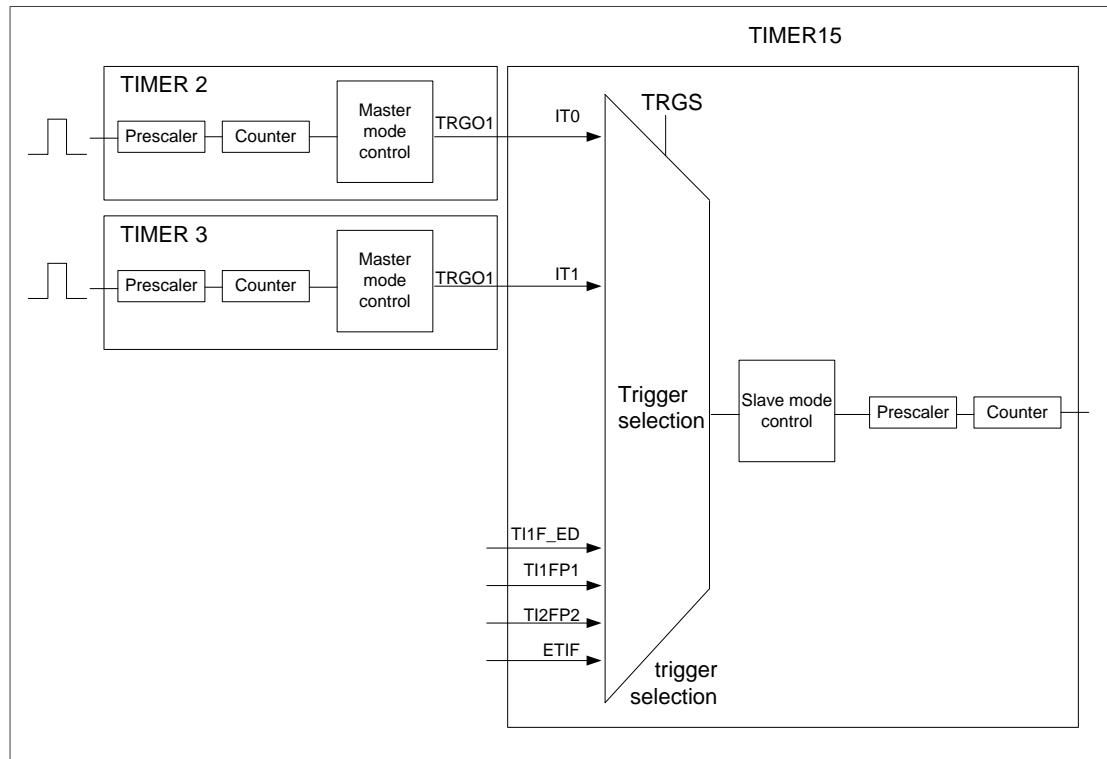


### 定时器互连

定时器互相之间可以级联或者同步。可以通过配置一个定时器工作在主模式另一个定时器工作在从模式来实现。下图显示了一些主从模式触发选择的例子。

下图显示了当定时器15配置为从模式时的触发选择

图 9-126. 定时器 15 主从模式的例子



其他互连的例子:

- 定时器 2 作为定时器 15 的预分频器

参考下图连接配置定时器2为定时器15的预分频器，按以下步骤去做：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出 (配置TIMER2\_CTLR2寄存器)

的MMC=010)。定时器2在每次计数器溢出时，输出一个周期信号。

2. 配置定时器2周期 (TIMER2\_CARL寄存器)。
3. 选择定时器15输入触发源为定时器2 (配置TIMER15\_SMC寄存器的TRGS=001)。
4. 配置定时器15在外部时钟模式1 (配置TIMER15\_SMC寄存器的SMC=111)。
5. 写1到CEN位启动定时器15 (TIMER15\_CTLR1寄存器)。
6. 写1到CEN位启动定时器2 (TIMER2\_CTLR1寄存器)。

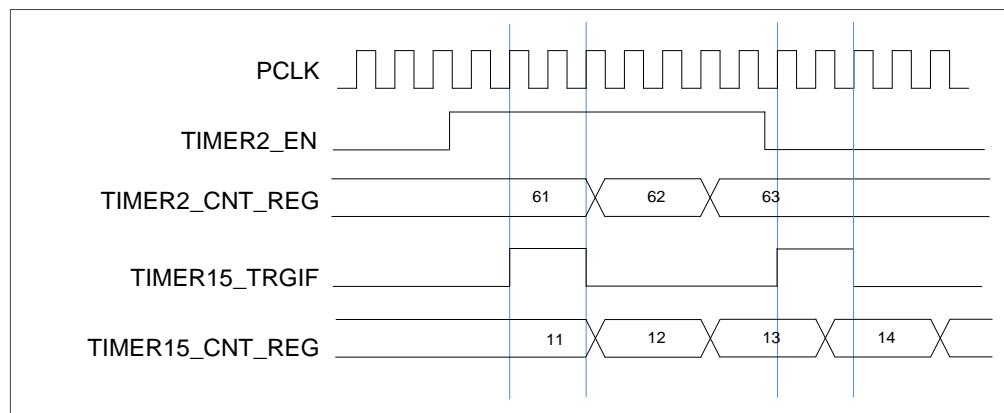
#### ■ 用定时器2的使能/更新信号来启动定时器15

首先，用定时器2的使能输出来使能定时器15，参见下图，在定时器2的使能输出触发的分频内部时钟下，定时器15从当前值开始计数。

当定时器15接收到触发信号，它的CEN位被自动置1，计数器计数直到关闭。两个计数器时钟频率都是PCLK时钟经过预分频器3分频( $f_{CNT\_CLK} = f_{PCLK}/3$ )。按下面步骤进行：

1. 配置定时器2为主模式，发送它的使能信号作为触发输出(配置TIMER2\_CTLR2寄存器的MMC=001)。
2. 配置定时器15选择输入触发来自定时器2 (配置TIMER15\_SMC寄存器的TRGS=000)。
3. 配置定时器15在触发模式 (配置TIMER15\_SMC寄存器的SMC=110)。
4. 写1到CEN来开启定时器2 (TIMER2\_CTLR1寄存器)。

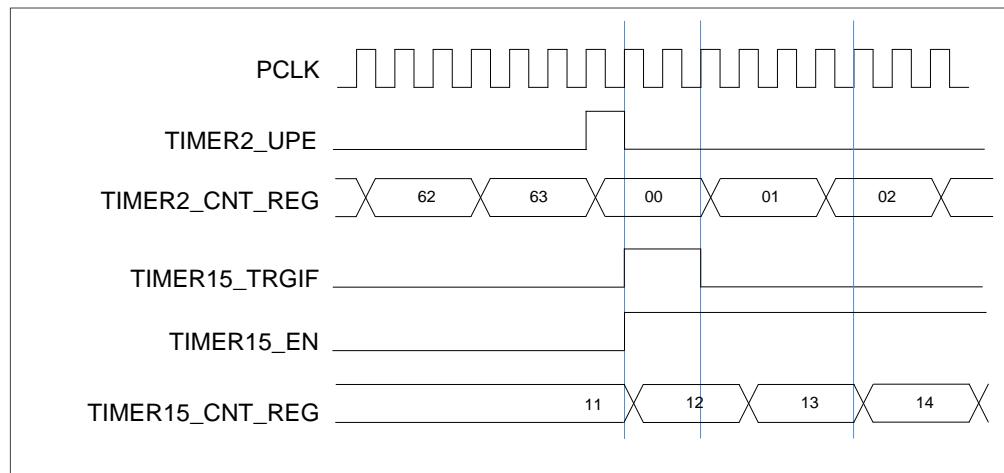
图 9-127. 用定时器2的使能输出触发定时器15



在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。参考下图，按以下步骤进行：

1. 配置定时器2在主模式下，发送它的更新事件(UPE)作为触发输出(配置TIMER2\_CTLR2寄存器的MMC=010)。
2. 配置定时器2周期 (TIMER2\_CARL寄存器)。
3. 配置定时器15获取来自定时器2的输入触发信号 (配置TIMER15\_SMC寄存器的TRGS=001)。
4. 配置定时器15在触发模式 (配置TIMER15\_SMC寄存器的SMC=110)。
5. 写1到CEN位来启动定时器2 (TIMER2\_CTLR1寄存器)。

图 9-128. 用定时器 2 的更新来触发定时器 15

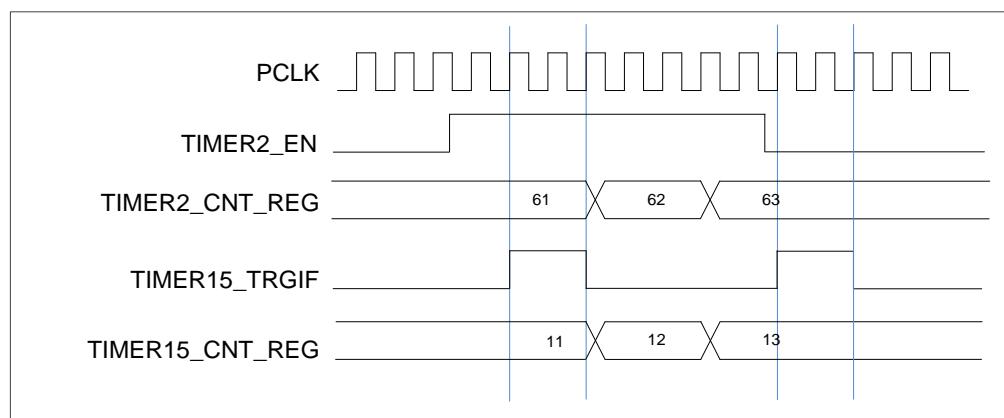


- 使用定时器 2 的使能/OC1 参考信号来使能定时器 15 计数。

在这个例子中，我们使用定时器2的使能输出来控制定时器15的使能。参考下图，在定时器2被使能后，定时器15在内部分频的时钟上开始计数。两个计数器的时钟频率都是由PCLK时钟三分频得来( $f_{CNT\_CLK}=f_{PCLK}/3$ )，按以下步骤做：

1. 配置定时器2在主模式，配置其输出使能信号作为触发输出(配置TIMER2\_CTLR2寄存器的MMC=001)。
2. 配置定时器15从定时器2获取输入触发 (配置TIMER15\_SMC寄存器的TRGS=001)。
3. 配置定时器15工作在暂停模式 (配置TIMER15\_SMC寄存器的SMC=101)。
4. 写1到CEN位来使能定时器15 (TIMER15\_CTLR1寄存器)。
5. 写1到CEN位来启动定时器2 (TIMER2\_CTLR1寄存器)。
6. 写0到CEN位来停止定时器2 (TIMER2\_CTLR1寄存器)。

图 9-129. 用定时器 2 的使能来选通定时器 15

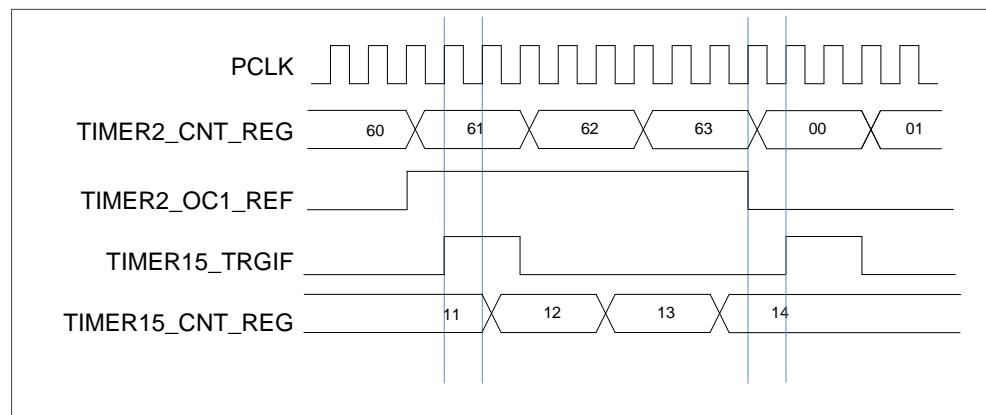


这个例子中，我们也可以使用OCx\_Ref代替使能信号输出作为触发源。按以下步骤做：

1. 配置定时器2在主模式下，配置其输出比较1参考信号(OC1REF)为触发输出(配置TIMER2\_CTLR2寄存器的MMC=100)。
2. 配置定时器2的OC1REF波形 (TIMER2\_CHCTRLR1寄存器)。

3. 配置定时器15获取来自定时器2的输入触发 (配置TIMER15\_SMC寄存器的TRGS=001)。
4. 配置定时器15工作在暂停模式 (配置TIMER15\_SMC寄存器的SMC=101)。
5. 写1到CEN位来使能定时器15 (TIMER15\_CTLR1寄存器)。
6. 写1到CEN位来开启定时器 2 (TIMER2\_CTLR1寄存器)。

图 9-130. 用定时器 2 的 OC1REF 选通定时器 15



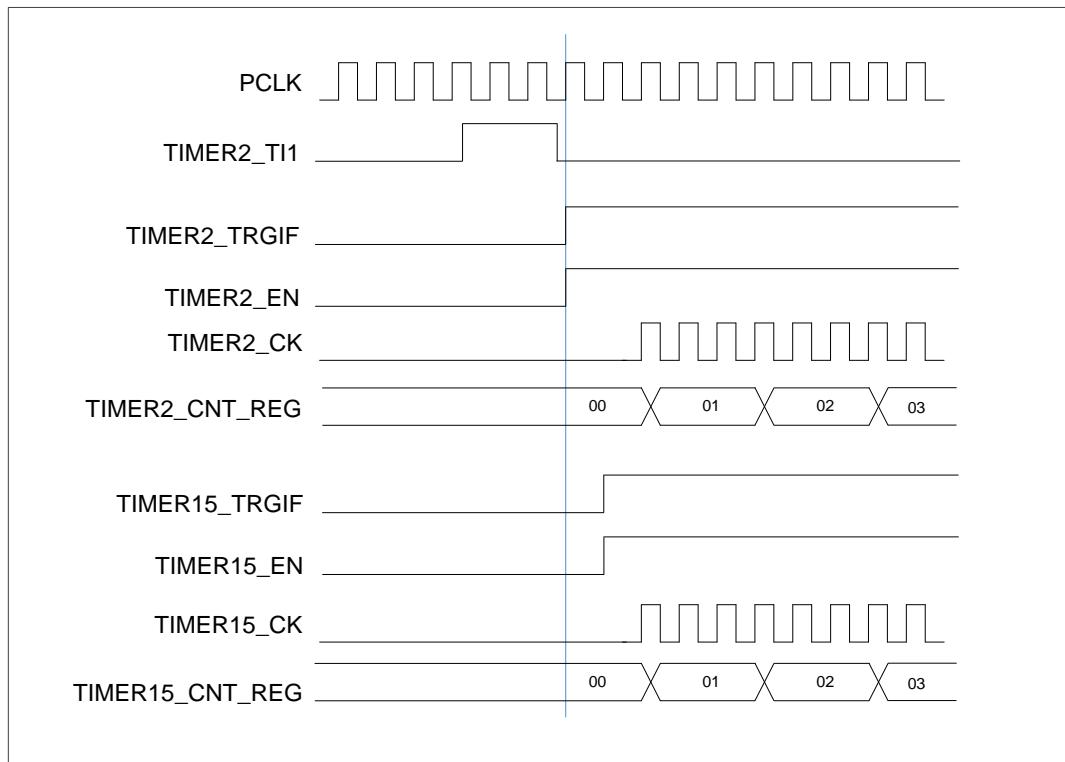
#### ■ 使用一个外部触发来同步两个定时器

配置定时器2的使能触发定时器15的开启，配置定时器2的TI1输入信号上升沿来触发定时器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。按以下步骤做：

1. 配置定时器2工作在从模式来获取来自TI1的触发输入(配置TIMER2\_SMC寄存器的TRGS=100)。
2. 配置定时器2工作在触发模式 (配置TIMER2\_SMC 寄存器的SMC=110)。
3. 写MSM=1(TIMER2\_SMC寄存器)来配置定时器2工作在主/从模式。
4. 配置定时器15的触发输入来自定时器2 (配置TIMER15\_SMC 寄存器的TRGS=001)。
5. 配置定时器15工作在触发模式 (配置TIMER2\_SMC 寄存器的SMC=110)。

定时器2的TI1信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的TRGIF标志位都被置1。

图 9-131. 用定时器 2 的 TI1 输入来触发定时器 15 和定时器 2



### 定时器 debug 模式

当微控制器在调试模式，若Cortex™-M3内核停止，且MCUDBG模块中DBG\_TIMERx\_STOP配置位被置1，TIMERx计数器停止。

### 9.5.4. TIMER15 寄存器

#### TIMERx 控制寄存器 1 (TIMERx\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CDIV[1:0]		ARSE		保留		SPM		UPS		UPDIS	CEN

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CDIV[1:0]	<p>时钟分频</p> <p>软件配置CDIV位规定定时器时钟(PCLK) 和死区时间、采样时钟(DTS)之间的分频系数，该位被死区时间产生器和数字滤波器使用。</p> <ul style="list-style-type: none"> <li>00: <math>f_{DTS}=f_{PCLK}</math></li> <li>01: <math>f_{DTS}=f_{PCLK}/2</math></li> <li>10: <math>f_{DTS}=f_{PCLK}/4</math></li> <li>11: 保留</li> </ul>
7	ARSE	<p>自动重载影子使能</p> <ul style="list-style-type: none"> <li>0: 禁止TIMERx_CARL寄存器的影子寄存器</li> <li>1: 使能TIMERx_CARL寄存器的影子寄存器</li> </ul>
6:4	保留	必须保持复位值
3	SPM	<p>单脉冲模式</p> <ul style="list-style-type: none"> <li>0: 在发生更新事件时，计数器不停止</li> <li>1: 在发生下一次更新事件（清除CEN位）时，计数器停止</li> </ul>
2	UPS	<p>更新请求源</p> <p>软件通过该位选择更新事件源。</p> <p>0: 使能后，下述任一事件产生更新中断或DMA请求：</p> <ul style="list-style-type: none"> <li>- UPG位被置1</li> <li>- 计数器溢出/下溢</li> <li>- 从模式控制器产生的更新</li> </ul> <p>1: 使能后只有计数器溢出/下溢才产生更新中断或DMA请求</p>
1	UPDIS	<p>禁止更新</p> <p>软件通过该位允许/ 禁止更新事件的产生</p> <p>0: 允许更新事件产生。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值：</p> <ul style="list-style-type: none"> <li>- UPG位被置1</li> </ul>

- 计数器溢出/下溢
  - 从模式控制器产生一个更新事件
- 1: 禁止更新事件。带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化

0	CEN	使能计数器
0:	禁止计数器	
1:	使能计数器	在软件将CEN位置1后，外部时钟、暂停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。

### TIMERx 控制寄存器 2 (TIMERx\_CTLR2)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ISO2	ISO1N	ISO1	保留.		MMC[2:0]	DMAS	CCUC	保留.	CCSE		
				rw	rw	rw			rw	rw	rw		rw		rw

位/位域	名称	描述
15:11	保留	必须保持复位值
10	ISO2	通道2的空闲输出状态 参见ISO1位
9	ISO1N	通道1的互补输出空闲状态 0: 当POE=0, OC1N=0 1: 当POE=0, OC1N=1 只有当TIMERx_BKDT 寄存器的LK[1:0]位为00时此位才可以被修改。
8	ISO1	通道1的空闲输出状态 0: 当POE=0, OC1=0 1: 当POE=0, OC1=1 如果OC1N生效，一个死区时间后OC1输出改变。此位只有在TIMERx_BKDT 寄存器的LK[1:0]位为00的时候可以被更改。
7	保留	必须保持复位值
6:4	MMC[2:0]	这些位控制TRGO信号的选择，主模式下发给从定时器用于同步功能 000: 复位。TIMERx_EVG寄存器的UG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟 001: 使能。此模式可用于同时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CNT_EN 被主模式控制器选为触发输出(TRGO)。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。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

010: 更新。更新事件被主模式控制器选为触发输入

011: 捕获/比较脉冲.在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲

100: 比较。OC1REF信号被用于作为触发输出

101: 比较。OC2REF 信号被用于作为触发输出

110: 比较。OC3REF 信号被用于作为触发输出

111: 比较。OC4REF 信号被用于作为触发输出

3	DMAS	DMA请求源选择
0:	当通道x事件发生，发送通道x的DMA请求	
1:	当更新事件发生，发送通道x的DMA请求	
2	CCUC	捕获/比较控制影子寄存器更新控制  使能(CCSE=1)捕获/比较控制影子寄存器(CHxE、CHxNE和CHxOM 位)后，该位控制这些影子寄存器更新的时间  0: CCUG位被置1时更新影子寄存器 1: 当CCUG位被置1或检测到TRGI上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值.
0	CCSE	捕获/比较控制影子寄存器使能  0: 影子寄存器CHxE、CHxNE和CHxOM 位被禁止 1: 使能影子寄存器CHxE、CHxNE和CHxOM 位，当这些位被写时，只有当CCUG位被置1时这些位才被更新 当通道没有互补输出时，此位无效。

### TIMERx 从模式控制寄存器 (TIMERx\_SMC)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:8	保留	必须保持复位值
7	MSM	主-从模式  主-从模式下输入触发上的事件被延迟了，以允许当前定时器和它的从设备通过TRGO完美同步。如果想要在单个外部事件上同步一些定时器，此模式非常有用。 0: 禁止主-从模式 1: 使能主-从模式

6:4	TRGS[2:0]	触发选择 这些位选择用于同步计数器的触发输入。 000: 内部触发输入0 (ITR0) TIMER2 001: 内部触发输入1 (ITR1) TIMER3 010: 内部触发输入2 (ITR2) 保留 011: 内部触发输入3 (ITR3) 保留 100: TI1的边沿标志位 (TI1F_ED) 101: 滤波后的通道1输入 (TI1FP1) 110: 滤波后的通道2输入 (TI2FP2) 111: 外部触发输入(ETIF) 从模式被使能后这些位不能改
3	保留	必须保持复位值
2:0	SMC[2:0]	从模式控制 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: 保留, 不要设置这个值。 010: 保留, 不要设置这个值。 011: 保留, 不要设置这个值。 100: 复位模式。选中的触发输入的上升沿重新初始化计数器, 并且更新影子寄存器 101: 暂停模式。当触发输入为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止 110: 触发模式。计数器在触发输入TRGI 的上升沿启动。计数器不能被从模式控制器关闭。 111: 外部时钟模式1。选中的触发输入的上升沿驱动计数器 如果TI1F_ED 被选为触发输入时, 不要使用暂停模式。这是因为, TI1F_ED 在每次TI1F 变化时输出一个脉冲, 然而暂停模式是要检查触发输入的电平。

### TIMERx DMA 和中断使能寄存器 (TIMERx\_DIE)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留	TRGDE	保留		CH2DE	CH1DE	UPDE	BKIE	TRGIE	CCUIE	保留	CH2IE	CH1IE	UPIE		

位/位域	名称	描述
15	保留	必须保持复位值.
14	TRGDE	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求

---

13:11	保留	必须保持复位值
10	CH2DE	通道2 DMA请求使能 0: 禁止通道2 DMA请求 1: 使能通道2 DMA请求
9	CH1DE	通道1 DMA请求使能 0: 禁止通道1 DMA请求 1: 使能通道1 DMA请求
8	UPDE	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CCUIE	通道控制更新中断使能 0: 禁止通道控制更新中断 1: 使能通道控制更新中断
4:3	保留	必须保持复位值.
2	CH2IE	通道2中断使能 0: 禁止通道2中断 1: 使能通道2中断
1	CH1IE	通道1中断使能 0: 禁止通道1中断 1: 使能通道1中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### TIMERx 状态寄存器 (TIMERx\_STR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH2OF	CH1OF	保留	BKIF	TRGIF	CCUIF	保留.	CH2IF	CH1IF	UPIF		

rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0

位/位域	名称	描述
15:11	保留	必须保持复位值
10	CH2OF	通道2重复捕获标志 参见CH1OF描述
9	CH1OF	通道1重复捕获标志 当通道1被配置为输入模式时，在CH1IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。 0：无重复捕获中断发生 1：发生了重复捕获中断
8	保留	必须保持复位值
7	BKIF	中止中断标志位 一旦中止输入有效，由硬件对该位置‘1’。如果中止输入无效，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置1。此位由软件清0。当从模式控制器处于除暂停模式外的其它模式时，在TRGI 输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，TRGI的任意边沿都可以产生触发事件。 0：无触发事件产生 1：触发中断产生
5	CCUIF	通道控制更新中断标志 当通道控制更新事件发生时此标志位被硬件置1，此位由软件清0。 0：无通道控制更新中断发生 1：通道控制更新中断发生
4:3	保留	必须保持复位值
2	CH2IF	通道2中断标志 参见CH1IF描述
1	CH1IF	通道1中断标志 此标志由硬件置1软件清0。当通道1在输入模式下时，捕获事件发生时此标志位被置1；当通道1在输出模式下时，此标志位在一个比较事件发生时被置1。 0：无通道1中断发生 1：通道1中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0：无更新中断发生 1：发生更新中断

### TIMERx 事件发生寄存器 (TIMERx\_EVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				BKG	TRGG	CCUG	保留				CH2G	CH1G	UPG		
				w	w	w					w	w	w		

位/位域	名称	描述
15:8	保留	必须保持复位值.
7	BKG	<p>产生中止事件</p> <p>该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POE位被清0且BKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。</p> <p>0: 不产生中止事件</p> <p>1: 产生中止事件</p>
6	TRGG	<p>触发事件产生</p> <p>此位由软件置1，由硬件自动清0. 当此位被置1，TIMERx_STR寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。</p> <p>0: 无触发事件产生</p> <p>1: 产生触发事件</p>
5	CCUG	<p>通道控制更新事件发生</p> <p>此位由软件置1，由硬件自动清0. 当此位被置1，通道控制寄存器 (CHxE, CHxNE和CHxOM 位) 的互补输出被更新。</p> <p>0: 不产生通道控制更新事件</p> <p>1: 产生通道控制更新事件</p>
4:3	保留	必须保持复位值
2	CH2G	<p>通道2捕获或比较事件发生</p> <p>参见CH1G描述</p>
1	CH1G	<p>通道1捕获或比较事件发生</p> <p>该位由软件置1，用于在通道1产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CC1IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道1配置为输入模式，计数器的当前值被TIMERx_CHCC1寄存器捕获，如果CH1IF标志位已经为1，则CH1OF标志位被置1。</p> <p>0: 不产生通道1捕获或比较事件</p> <p>1: 发生通道1捕获或比较事件</p>
0	UPG	<p>此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。</p>

0: 无更新事件产生

1: 产生更新事件

### TIMERx 通道控制寄存器 1 (TIMERx\_CHCTRL1)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CH2 OCE	CH2OM[2:0]		CH2 OSE	CH2 OFE	CH2M[1:0]	CH1 OCE	CH1OM[2:0]		CH1OSE	CH1 OFE	CH1M[1:0]	CH1ICF[3:0]	CH1ICP[1:0]	CH1ICF[3:0]	CH1ICP[1:0]	CH1M[1:0]
	CH2ICF[3:0]			CH2ICP[1:0]			CH1ICF[3:0]			CH1ICP[1:0]						
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

#### 输出比较模式:

位/位域	名称	描述
15	CH2OCE	通道2输出比较清0使能 参见CH1OCE描述
14:12	CH2OM[2:0]	通道2输出比较模式 参见CH1OM描述
11	CH2OSE	通道2输出比较影子寄存器使能 参见CH1OSE描述
10	CH2OFE	通道2输出比较快速使能 参见CH1OFE描述
9:8	CH2M[1:0]	通道2模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHE寄存器的CH2E位被清0)时这些位才可以写。 00: 通道2配置为输出 01: 通道2配置为输入, IC2映射在TI2上 10: 通道2配置为输入, IC2映射在TI1上 11: 通道2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 TIMER_SMC寄存器的TRGS位选择)。
7	CH1OCE	通道1输出比较清0使能 当此位被置1, 当检测到ETIF输入高电平时, OC1参考信号被清0。 0: 禁止通道1输出比较清零 1: 使能通道1输出比较清零
6:4	CH1OM[2:0]	通道1输出比较模式 此位定义了输出参考信号OC1REF的动作, 而OC1REF决定了OC1、OC1N的值。 OC1REF高电平有效, 而OC1、OC1N的有效电平取决于CH1P、CH1NP位。 000: 冻结。输出比较寄存器TIMERx_CHCC1与计数器间的比较对OC1REF不起作用

001: 匹配时设置为高。当计数器的值与捕获/比较寄存器TIMERx\_CHCC1相同时，强制OC1REF为高。

010: 匹配时设置为低。当计数器的值与捕获/比较寄存器TIMERx\_CHCC1相同时，强制OC1REF为低。

011: 匹配时翻转。当计数器的值与捕获/比较寄存器TIMERx\_CHCC1相同时，强制OC1REF翻转。

100: 强制为低。强制OC1REF为低电平

101: 强制为高。强制OC1REF为高电平

110: PWM模式1。在向上计数时，一旦计数器值小于TIMERx\_CHCC1时，OC1REF为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx\_CHCC1时，OC1REF为低电平，否则为高电平。

111: PWM模式2。在向上计数时，一旦计数器值小于TIMERx\_CHCC1时，OC1REF为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx\_CHCC1时，OC1REF为高电平，否则为低电平。

在PWM模式1或PWM模式2中，只有当比较结果改变了或者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。

当TIMERx\_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。

3	CH1OSE	通道1输出比较影子寄存器使能 当此位被置1，TIMERx_CHCC1寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0: 禁止通道1输出/比较影子寄存器 1: 使能通道1输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTLR1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。 当TIMERx_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。
2	CH1OFE	通道1输出比较快速使能 该位用于通道配置为PWM模式时加快CC1输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，OC1被设置为比较电平而与比较结果无关 0: 禁止通道1输出比较快速。当触发器的输入有一个有效沿时，激活CC1输出的最小延时为5个时钟周期 1: 使能通道1输出比较快速。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3个时钟周期
1:0	CH1M[1:0]	通道1模式选择 这些位定义了输入的方向和输入信号的选择。只有当通道关闭(TIMERx_CHE寄存器的CH1E位被清0)时这些位才可写。 00: 通道1配置为输出 01: 通道1配置为输入，IC1映射在TI1上 10: 通道1配置为输入，IC1映射在TI2上 11: 通道1配置为输入，IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMC寄存器的TRGS位)。

## 输入捕获模式:

位/位域	名称	描述
15:12	CH2ICF[3:0]	通道2输入捕获滤波控制 参见CH1ICF描述
11:10	CH2ICP[1:0]	通道2输入捕获预分频器 参见CH1ICP描述
9:8	CH2M[1:0]	通道2模式选择 此位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH2E位被清0)时这些位才可写。 00: 通道2配置为输出 01: 通道2配置为输入, IC2映射在TI2上 10: 通道2配置为输入, IC2映射在TI1上 11: 通道2配置为输入, IC2映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)。
7:4	CH1ICF[3:0]	通道1输入捕获滤波控制 数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。 这些位定义了TI1输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP}=f_{DTS}$ , N=1 0001: $f_{SAMP}=f_{PCLK}$ , N=2 0010: $f_{SAMP}=f_{PCLK}$ , N=4 0011: $f_{SAMP}=f_{PCLK}$ N=8 0100: $f_{SAMP}=f_{DTS}/2$ , N=6 0101: $f_{SAMP}=f_{DTS}/2$ , N=8 0110: $f_{SAMP}=f_{DTS}/4$ , N=6 0111: $f_{SAMP}=f_{DTS}/4$ , N=8 1000: $f_{SAMP}=f_{DTS}/8$ , N=6 1001: $f_{SAMP}=f_{DTS}/8$ , N=8 1010: $f_{SAMP}=f_{DTS}/16$ , N=5 1011: $f_{SAMP}=f_{DTS}/16$ , N=6 1100: $f_{SAMP}=f_{DTS}/16$ , N=8 1101: $f_{SAMP}=f_{DTS}/32$ , N=5 1110: $f_{SAMP}=f_{DTS}/32$ , N=6 1111: $f_{SAMP}=f_{DTS}/32$ , N=8
3:2	CH1ICP[1:0]	通道1输入捕获预分频器 这2位定义了CC1 输入的预分频系数。一旦CH1E =0(TIMER15_CHE寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH1M[1:0]	通道1模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的

CH1E位被清0)时这些位才可写。

00: 通道1配置为输出

01: 通道1配置为输入, IC1映射在TI1上

10: 通道1配置为输入, IC1映射在TI2上

11: 通道1配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置TIMER\_SMC寄存器的TRGS位)。

### TIMERx 通道使能寄存器 (TIMERx\_CHE)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				保留			CH2NP	保留.	CH2P	CH2E	CH1NP	CH1NE	CH1P	CH1E	

rw                    rw                    rw                    rw                    rw                    rw                    rw

位/位域	名称	描述
15:8	保留	必须保持复位值
7	CH2NP	通道2互补输出极性 参见CH1NP描述
6	保留	必须保持复位值
5	CH2P	通道2极性 参见CH1P描述
4	CH2E	通道2使能 参见CH1E描述
3	CH1NP	通道1互补输出极性 当通道1配置为输入模式, 此位定义了互补输出信号的极性。 0: 通道1高电平有效 1: 通道1低电平有效 当TIMERx_BKDT寄存器的LK[1:0]=11或10时此位不能被更改。
2	CH1NE	通道1互补输出使能 当通道1配置为输出模式时, 将此位置1使能通道1的互补输出。 0: 禁止通道1互补输出 1: 使能通道1互补输出
1	CH1P	通道1极性 当通道1配置为输入模式时, 此位定义了IC1信号极性。当通道1配置为输出模式时, 此位定义了输出信号极性。 0: 通道1高电平有效 1: 通道1低电平有效

当TIMERx\_BKDT寄存器的LK[1:0]=11或10时此位不能被更改。

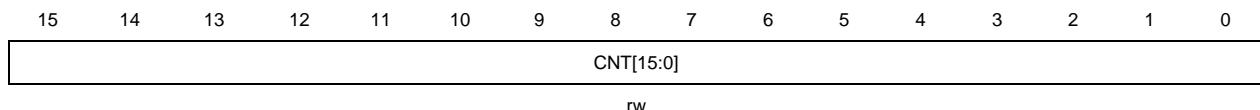
0	CH1E	通道1使能
		当通道1配置为输入模式时，将此位置1使能OC1信号有效。当通道1配置为输出模式时，将此位置1使能通道1上的捕获事件。
0:	禁止通道1	
1:	使能通道1	

### TIMERx 计数器寄存器 (TIMERx\_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



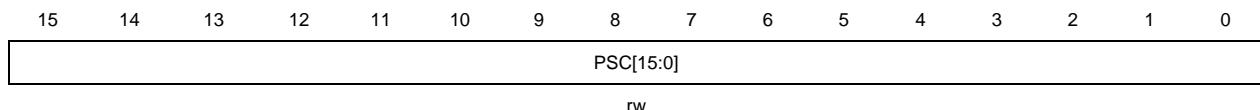
位/位域	名称	描述
15:0	CNT	这些位是当前的计数值。写操作能改变计数器值。

### TIMERx 预分频寄存器 (TIMERx\_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC的值被装入当前预分频寄存器。

### TIMERx 计数器自动重载寄存器 (TIMERx\_CARL)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



CARL[15:0]

rw

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

### TIMERx 重复计数寄存器 (TIMERx\_CREP)

地址偏移: 0x30

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

位/位域	名称	描述
15:8	保留	必须保持复位值
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

### TIMERx 通道 1 捕获/比较寄存器 (TIMERx\_CHCC1)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC1[15:0]															

rw

位/位域	名称	描述
15:0	CHCC1[15:0]	通道1的捕获或比较值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

### TIMERx 通道 2 捕获/比较寄存器 (TIMERx\_CHCC2)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC2[15:0]															
rw															

位/位域	名称	描述
15:0	CHCC2[15:0]	<p>通道2的捕获或比较值</p> <p>当通道2配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道2配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

### TIMERx 中止和死区时间寄存器 (TIMERx\_BKDT)

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POE	OAE	BRKP	BRKE	ROS	IOS	LK[1:0]									DT[7:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
15	POE	<p>主输出使能</p> <p>根据OAE位的设置值，该位可以由软件清0或被硬件自动置1。一旦中止输入有效，该位被硬件异步清0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHE寄存器的CHxE、CHxNE位)，则开启OC 和OCN 输出。</p> <p>0: 禁止OC 和OCN输出或强制为空闲状态</p> <p>1: 使能通道输出</p>
14	OAE	<p>自动输出使能</p> <p>此位定义了POE位是否可以被硬件自动置1。</p> <p>0: POE位不能被硬件置1</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POE位能被硬件自动置1。</p> <p>此位只有在TIMERx_BKDT寄存器的LK[1:0]=00时才可修改。</p>
13	BRKP	<p>中止极性</p> <p>此位定义了中止输入信号的极性。</p> <p>0: 中止输入低电平有效</p> <p>1: 中止输入高电平有效</p>
12	BRKE	<p>中止使能</p> <p>此位置1使能中止事件和CCS时钟失败事件输入。</p>

		0: 禁止中止输入 1: 使能中止输入 只有在TIMERx_BKDT寄存器的LK[1:0]=00时此位才可修改。
11	ROS	运行模式下“关闭状态”配置 当POE位被置1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。 0: 当POE位被置1，通道输出信号(OC/OCN)被禁止 1: 当POE位被置1，通道输出信号(OC/OCN)被使能，和TIMERx_CHE寄存器CHxE/CHxNE位有关 此位在TIMERx_BKDT寄存器的LK[1:0]=10或11时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当POE位被清0，此位定义了已经配置为输出模式的通道的输出状态。 0: 当POE位被清0，通道输出信号(OC/OCN)被禁止 1: 当POE位被清0，通道输出信号(OC/OCN)被使能，和TIMERx_CHE寄存器CHxE/CHxNE位有关 此位在TIMERx_BKDT寄存器的LK[1:0]=10或11时不能被更改。
9:8	LK[1:0]	锁定控制 这两位定义了寄存器的写保护特性。 00: 禁止锁定，无写保护 01: 锁定模式1，TIMERx_CTLR2寄存器的ISOx/ISOxN位和TIMERx_BKDT寄存器的BRKE/BRKP/OAE/DT位写保护 10: 锁定模式2。除了锁定模式1下的寄存器，TIMERx_CHE寄存器的CHxP/CHxNP位(如果相应通道配置为输出模式)和TIMERx_BKDT寄存器的ROS/IOS位写保护 11: 锁定模式3。除了锁定模式2下的寄存器，TIMERx_CHCTRLRx寄存器的CHxOM/CHxOSE(如果相应通道配置为输出)位写保护。 系统复位后这两位只能被写一次，一旦TIMERx_BKDT寄存器被写入，这两位被写保护。
7:0	DT[7:0]	死区时间值 这些位定义了插入互补输出之间的死区持续时间。DT值和死区时间的关系如下： DT [7:5] =0xx: DT= DT [7:0] x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> . DT [7:5] =10x: DT= (64+DT [5:0]) x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> *2. DT [7:5] =110: DT = (32+DT [4:0]) x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> *8. DT [7:5] =111: DT = (32+DT [4:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTs</sub> *16. 此位只有在TIMERx_BKDT寄存器的LK[1:0]=00时才可修改。

### TIMERx DMA 控制寄存器 (TIMERx\_DCTLR)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

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

rw

rw

位/位域	名称	描述
15:14	保留	必须保持复位值
12:8	DBLTH[4:0]	DMA连续传送长度 这5位定义了DMA 在连续模式下的传送长度( 当对TIMERx_DTRSF寄存器进行读或写时, 定时器则进行一次连续传送)。
7:5	保留	必须保持复位值
4:0	DBAR[4:0]	DMA地址 这5位定义了DMA 传送的地址( 当对TIMRx_DTRSF寄存器进行读或写时, DBA定义为从TIMERx_CTRLR1 寄存器所在地址开始的偏移量)。

### TIMERx DMA 传送寄存器 (TIMERx\_DTRSF)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

位/位域	名称	描述
15:0	DTRSF[15:0]	DMA传送 对TIMERx_DTRSF寄存器的读或写会导致对从TIMERx_CTRLR1 寄存器开始的以下地址范围所在寄存器的访问: (DBAR + 连续传送计数器) x 4 连续传送计数器由硬件计算, 范围为0到DBLTH。

### TIMERx 配置寄存器 (TIMERx\_CFG) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

rw

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CCSEL	<p>写捕获比较寄存器选择位 此位由软件写1或清0。</p> <p>1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无效</p>
0	OUTSEL	<p>输出值选择位 此位由软件写1或清0。</p> <p>1: 如果POE位与IOS位均为0，则输出无效 0: 无效</p>

## 9.6. 通用定时器 (TIMER16/TIMER17)

### 9.6.1. 简介

通用定时器TIMER16/TIMER17可以用于大量场合。它包括一个16位的向上计数器、2个16位的捕捉/比较寄存器(TIMERx\_CHCC)、1个16位的自动装载计数器寄存器(TIMERx\_CARL)和一些控制寄存器。它适合多种用途，包括通用定时器，输入信号脉冲宽度测量或者产生输出波形例如单脉冲输出或PWM输出。

通用 (TIMER16/TIMER17) 定时器完全独立，它们不共享资源但是可以同步

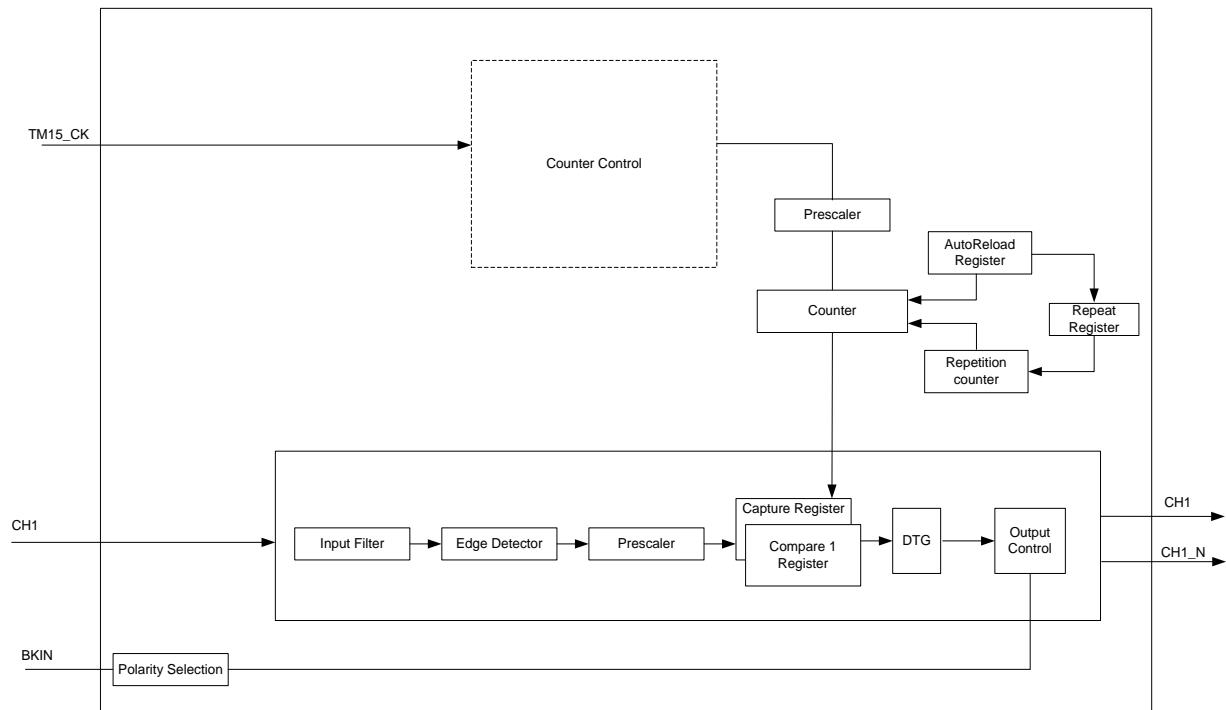
### 9.6.2. 主要特性

- 16位向上自动装载计数器
- 16位可编程的预分频器，计数器时钟频率的分频系数为1~65536之间的任意数值
- 1个独立通道，支持以下功能：输入捕获，输出比较， PWM波形产生(边缘或中间对齐模式)，单脉冲模式输出
- 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
- 中止输入信号可以将定时器输出信号置于一个已知状态
- 更新事件，输入捕获事件，输出比较事件，中止输入可以产生中断/DMA
- 可编程的死区时间用于输出匹配

### 9.6.3. 功能描述

下图提供了通用定时器内部配置细节.

图 9-132. 通用定时器框图 (TIMER16/17)



### 预分频计数器

预分频器可以将定时器(PCLK)的时钟频率按1到65536之间的任意值分频给计数器(CNT\_CLK)使用。分频系数受预分频寄存器TIMERx\_PSC控制，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 9-133. 当预分频器的参数从 1 变到 2 时，计数器时序图

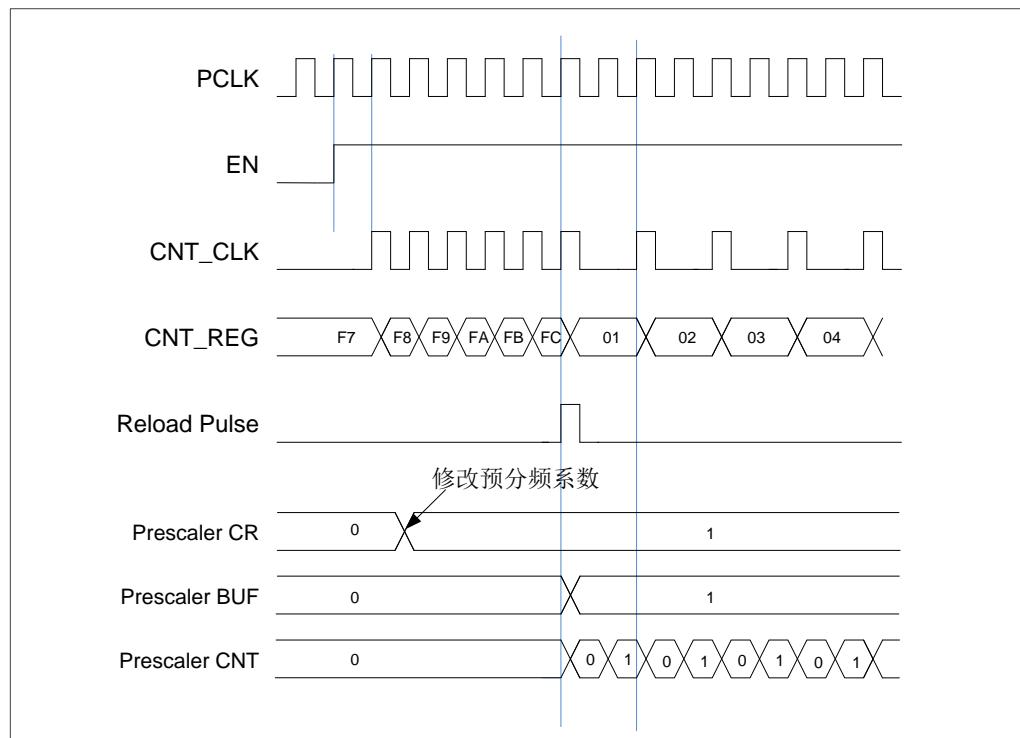
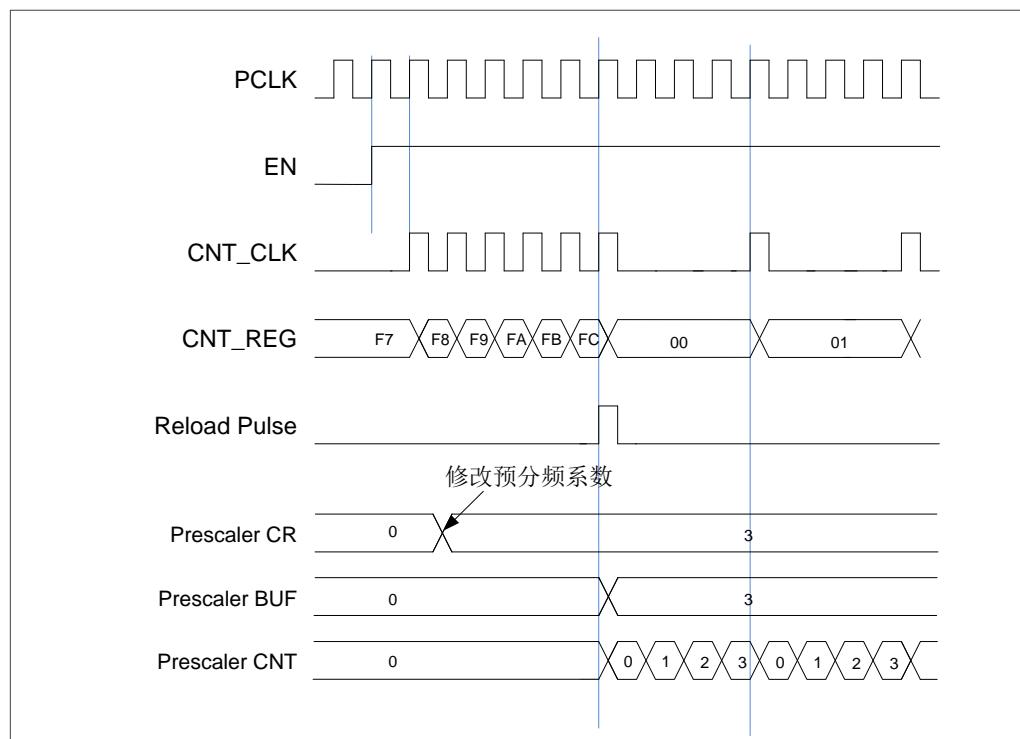


图 9-134. 当预分频器的参数从 1 变到 4 时，计数器时序图



### 向上计数模式

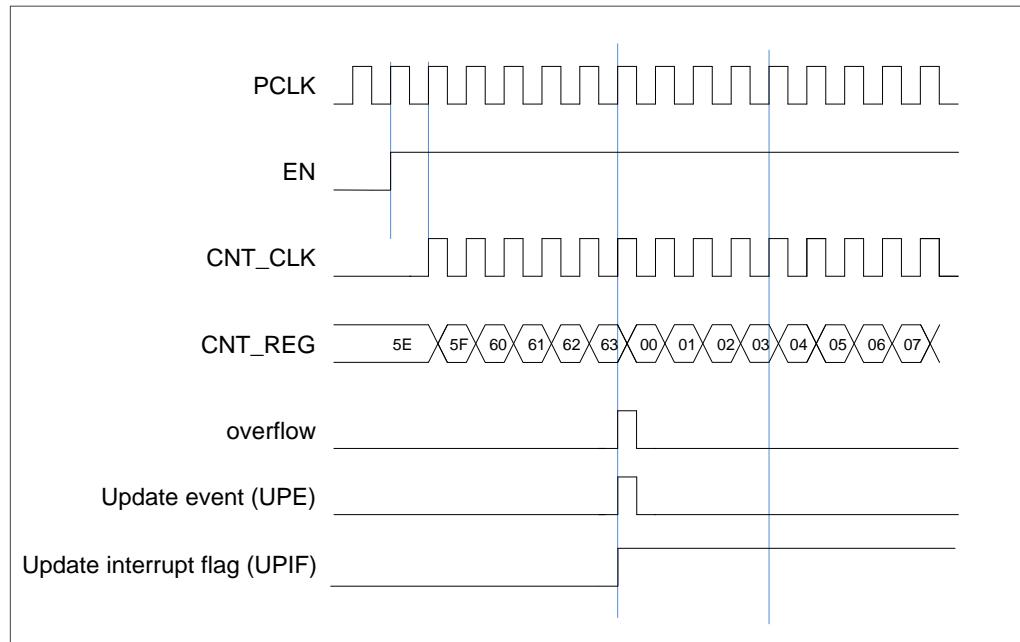
在向上计数模式中，计数器从0计数到自动加载值(TIMERx\_CARL寄存器的内容)，然后重新从0开始计数。如果使用了重复计数器功能吗，在向上计数达到设置的重复计数次数加1(TIMERx\_CREP+1)时，才产生更新事件，否则每次计数器溢出时都会产生更新事件。

可以将TIMERx\_EVG寄存器的UPG位置1来设置更新事件，计数值被清0产生更新事件。

设置TIMERx\_CTLR1寄存器的UPDIS位可以禁止更新事件

一旦发生更新事件，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。  
下图给出一些例子，当TIMERx\_CARL=0x63时计数器在不同时钟频率下计数器的动作

**图 9-135. 计数器时序图，内部时钟分频系数为 1**



**图 9-136. 计数器时序图，内部时钟分频系数为 2**

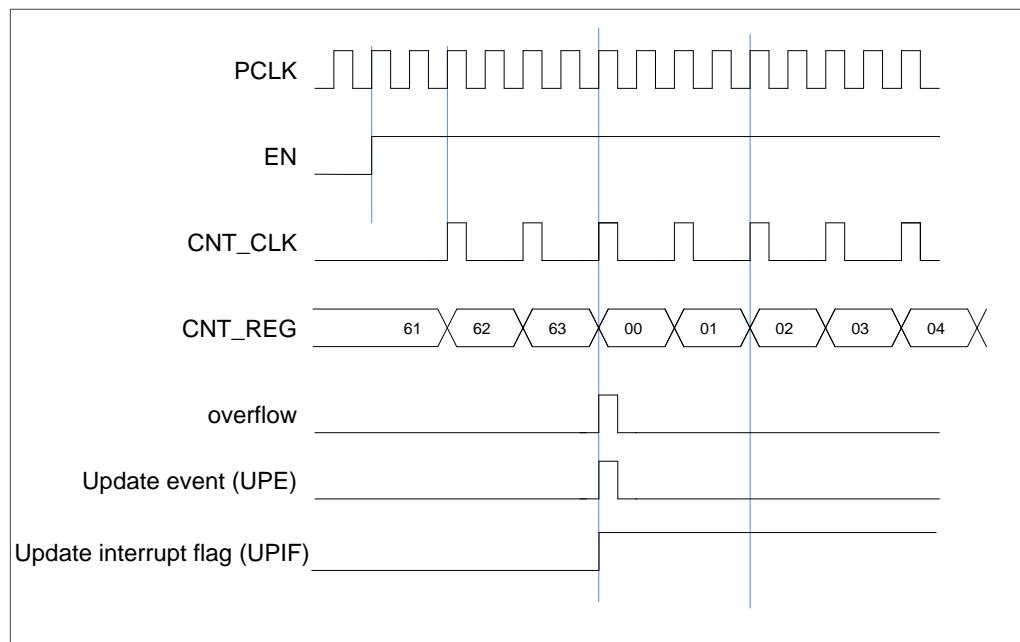


图 9-137. 计数器时序图，内部时钟分频系数为 4

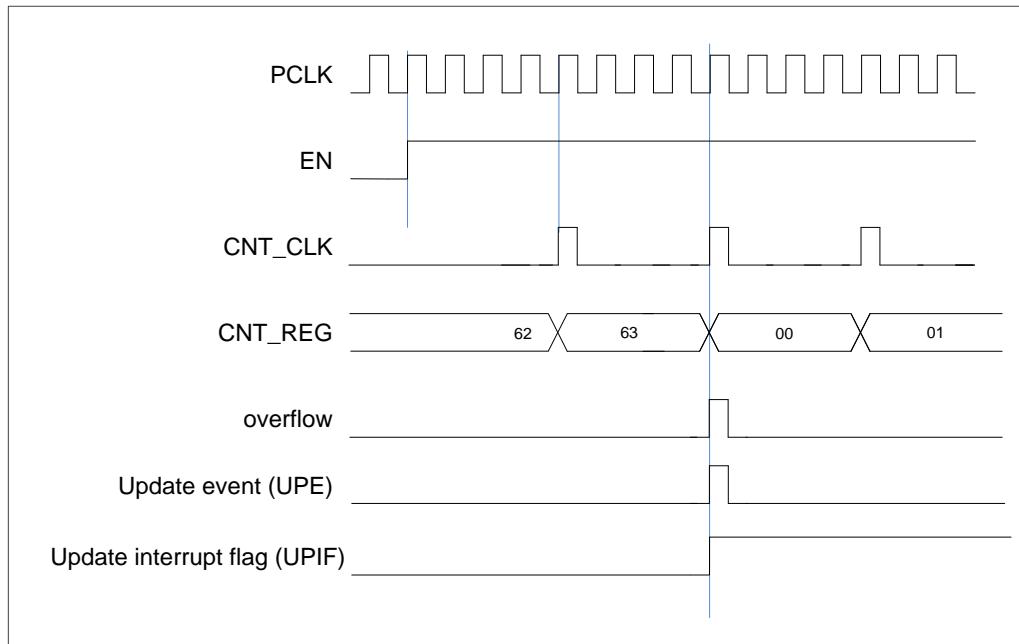


图 9-138. 计数器时序图，内部时钟分频系数为 N

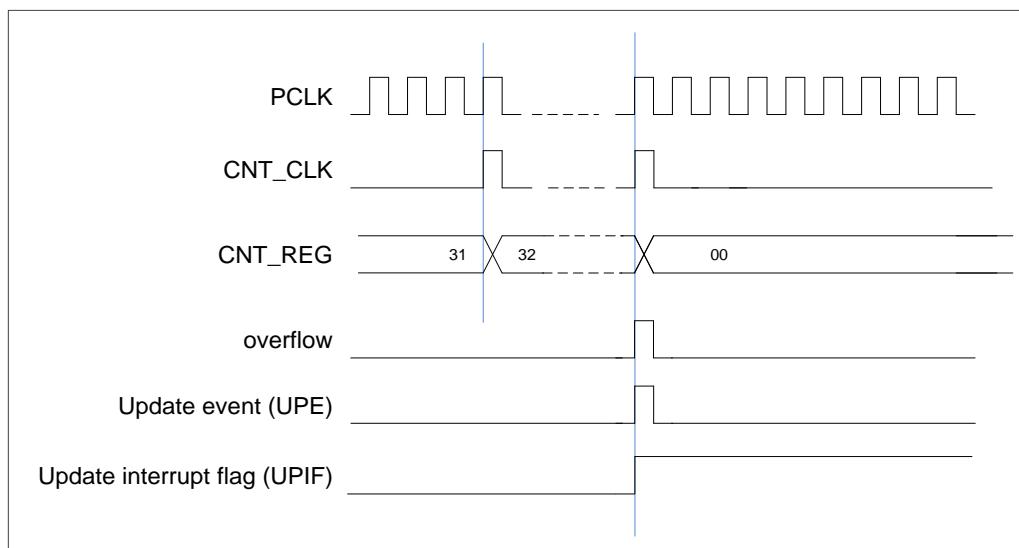


图 9-139. 计数器时序图, 当 ARSE=0 时的更新事件

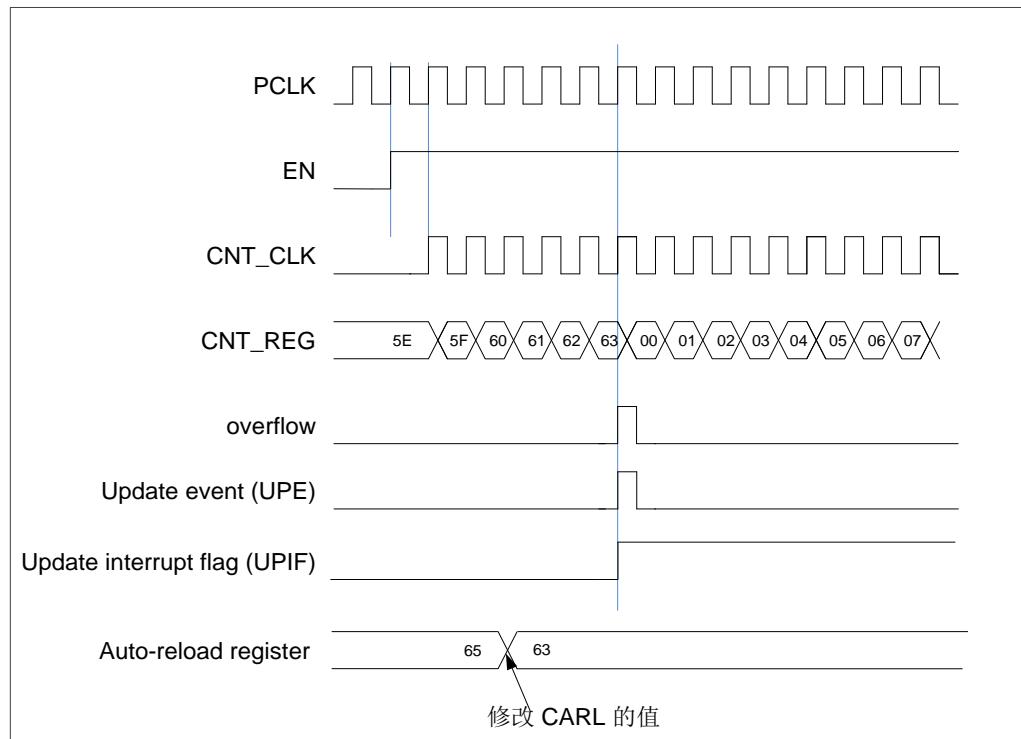
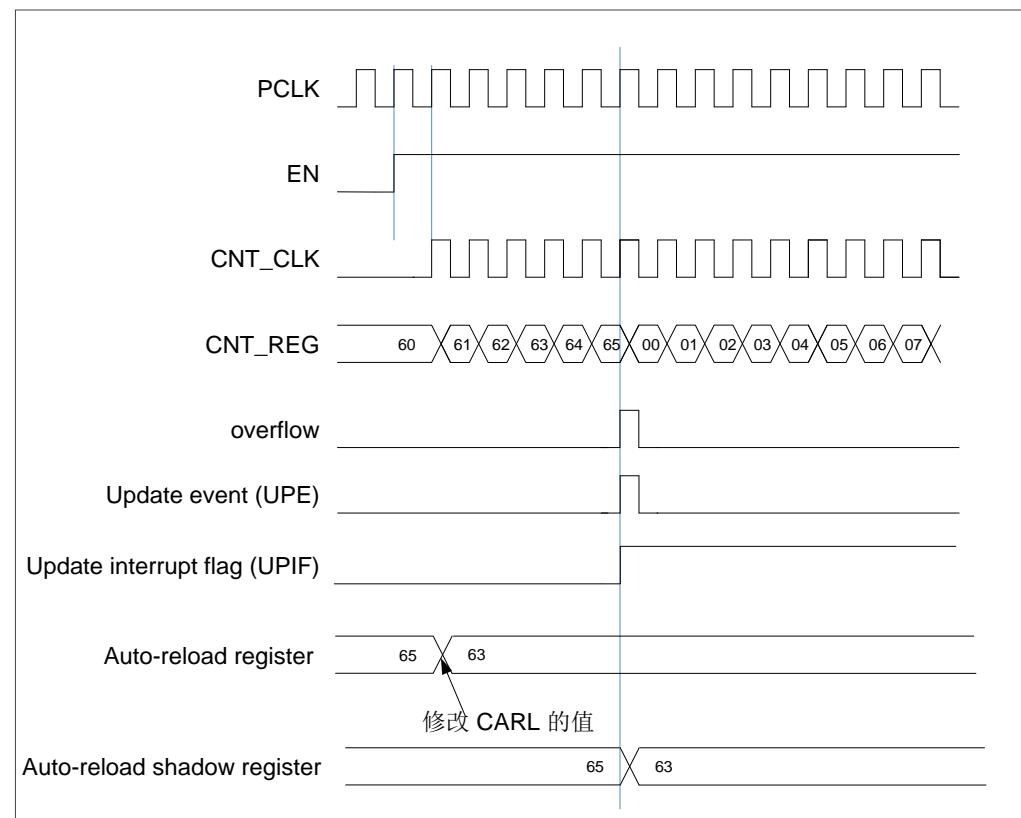


图 9-140. 计数器时序图, 当 ARSE=1 时更新事件



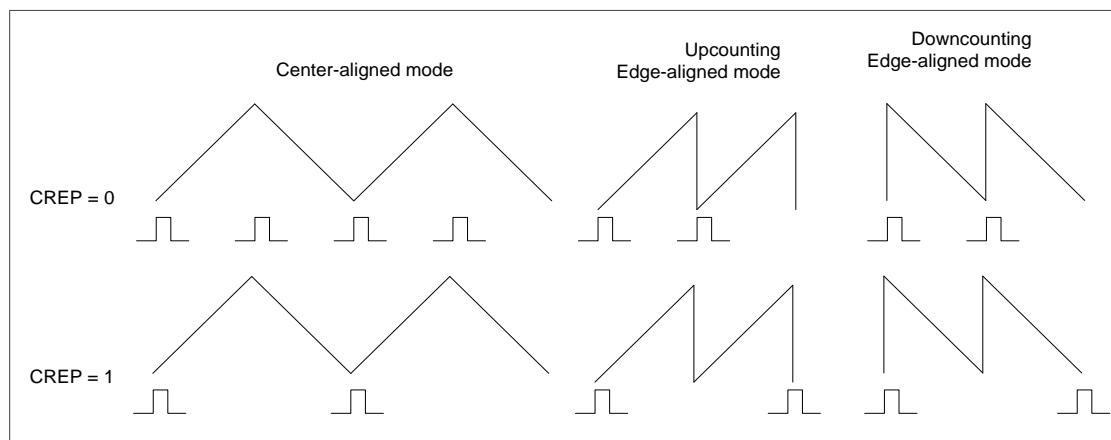
## 重复计数器

仅仅在一个给定数字N+1(N为TIMERx\_CREP寄存器的CREP)个循环周期的计数后，重复计数器被用来产生更新事件或者更新定时器寄存器。向上计数模式下，重复计数器在每个计数器上溢时递减；向下计数模式下，重复计数器在每个计数器下溢时递减；在中央对齐模式下，重复计数器在计数器上溢和下溢时递减。

将TIMERx\_EVG寄存器的UPG位置1可以重载TIMERx\_CREP寄存器中CREP的值并产生一个更新事件。

图 9-141. 不同模式及不同 TIMERx\_CREP 寄存器设置下更新速率的例子

TIMER16/TIMER17 只有向上计数边沿对齐模式



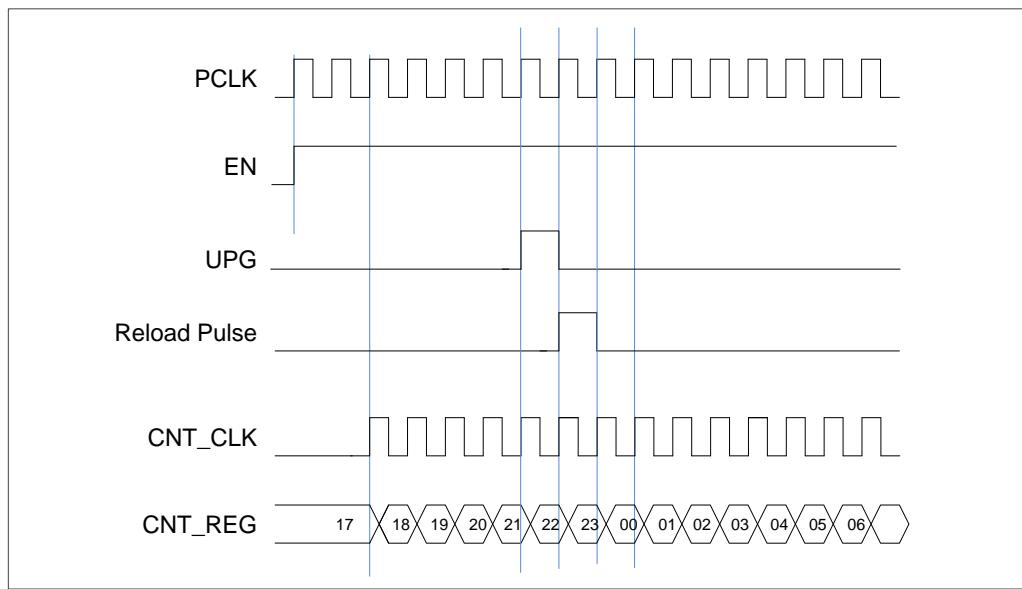
## 时钟选择

决定内部预分频计数器时钟源的定时器模块时钟控制器的描述如下。

### ■ 内部定时器时钟PCLK

由于没有从控制器，定时器唯一的时钟是PCLK。

图 9-142. 内部时钟分频为 1 时正常模式下的控制电路



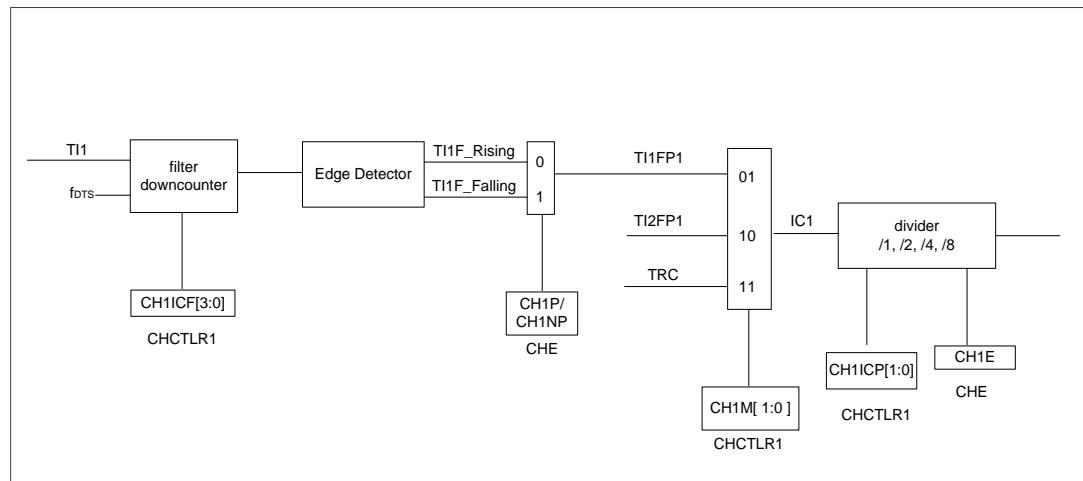
### 捕获/比较通道

TIMER16/TIMER17拥有1个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

#### ■ 输入捕获级

输入级包括一个数字滤波器，一个通道极性选择，边沿监测和一个通道预分频器。通道输入信号(TIx)被一个数字滤波器采样来产生一个滤波输入信号TIxF。然后通道极性和边沿监测模块产生一个TI1FP1或TI2FP1信号，来实现输入捕捉功能。通道输入预分频寄存器(CHxICP)可以设置有效输入事件数目。

图 9-143. 捕获/比较通道(例如：通道 1 输入级)



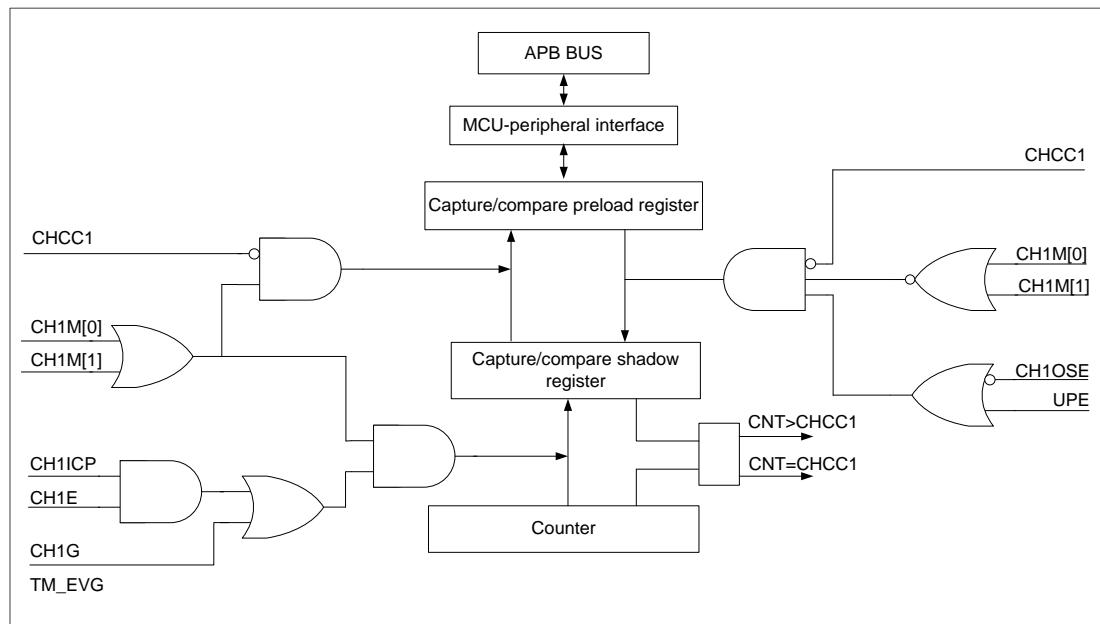
### ■ 通道控制器

TIMER16/17有1个独立的通道用来捕获输入或输出匹配比较

用于输入捕获模式下，当捕获事件发生时，**TIMERx\_CHCCx** 影子寄存器首先捕获计数值然后传递到**TIMERx\_CHCCx** 预装载寄存器中。

用于输出匹配比较模式下，**TIMERx\_CHCCx**预装载寄存器的值被复制到相应的影子寄存器中，然后计数值和影子寄存器的值比较。

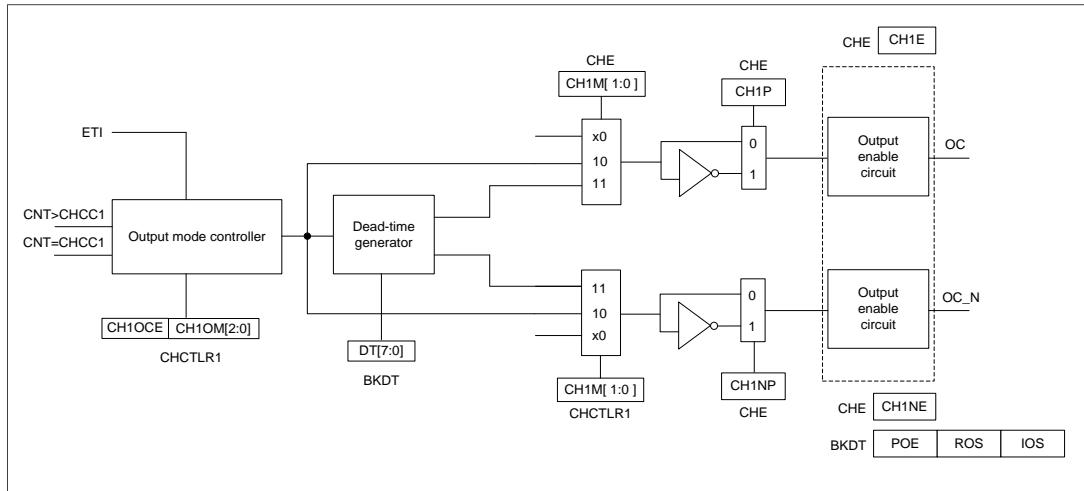
图 9-144. 捕获/比较通道 1 主电路



■ 输出级

TIMER16/17有1个用于匹配比较、单脉冲和PWM输出功能的通道。

图 9-145. 捕获/比较通道输出级 (通道 1)



### 输入捕获模式

通道用于输入捕获时，当一个有效输入信号的边沿跳变时，计数值被捕获到捕获/比较寄存器(TIMERx\_CHCCx)中。一旦捕获事件发生，TIMERx\_STR寄存器的CHxIF标志位被置1。如果CHxIF位已经被置1，还没来得及被软件清除，此通道上又有另一个捕获事件发生，相应通道捕获溢出标志位CHxOF将被置1。一旦捕获事件发生，CHxDE位决定了是否产生DMA请求，CHxIE位决定了是否产生中断。使用输入捕获模式需按以下步骤配置：

1. 配置通道控制寄存器(TIMERx\_CHCTLRx)的CCxM位域，选择输入信号
2. 配置通道控制寄存器(TIMERx\_CHCTLRx)的CHxICF位域，为输入信号添加输入滤波器
3. 配置通道使能寄存器(TIMERx\_CHE)的CHxP/CHxNP位，选择输入捕获的边沿（上升沿或下降沿）
4. 如果输入信号需要分频，配置通道控制寄存器(TIMERx\_CHCTLRx)的CHxICP位域
5. 如果需要中断，将DMA中断使能寄存器(TIMERx\_DIE)的CHxIE位置位
6. 如果需要DMA请求，将DMA中断使能寄存器(TIMERx\_DIE)的CHxDE位置位
7. 将通道使能寄存器(TIMERx\_CHE)的CCxE位置位，使能通道

按上述配置完寄存器后，该通道上的对应边沿信号将产生输入捕获事件。计数值被捕获到通道捕获/比较寄存器(TIMERx\_CHCCx)中。如果CHxIE位被置位，将会产生中断。如果CHxDE位被置位，将会产生DMA请求。

输入捕获模式也可以用于TIMERx\_CHx(TIx)引脚的信号脉宽测量。例如，PWM信号连接到TI1输入。配置TIMERx\_CHCTLRx寄存器的CH1M位域为01，将通道1输入捕获信号连接到TI1，并设置上升沿捕获。配置TIMERx\_CHCTLRx寄存器的CH2M位域为10，选择通道2捕获信号连接到TI1，并设置下降沿捕获。计数器配置为通道1上升沿触发后重新计数模式。此时TIMERx\_CHCC1可以测量PWM信号的周期，TIMERx\_CHCC2可以测量PWM信号的占空比。

### 输出比较模式/PWM 模式

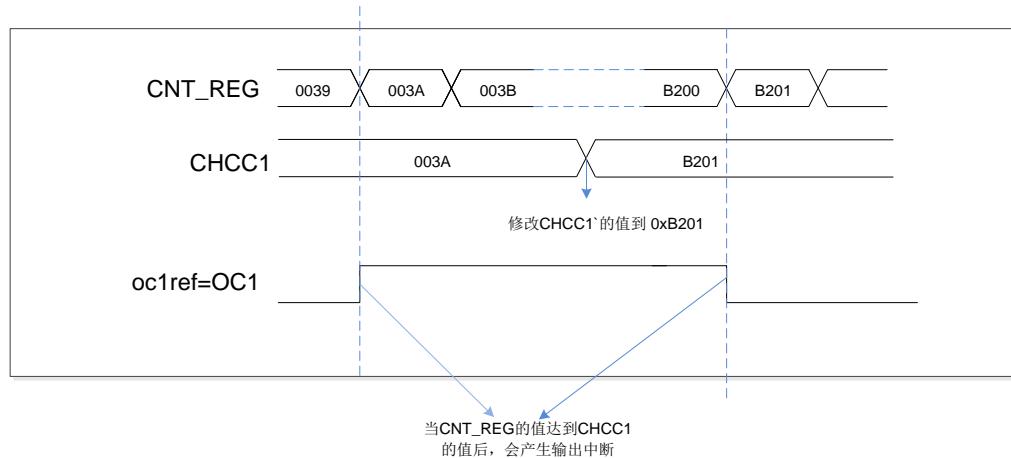
当配置通道控制寄存器(TIMERx\_CHCTRLRx)的CCxM位域为00,使通道工作于输出模式时,通道输出的波形由TIMERx\_CHCTRLRx 的CHxOM位域和计数器与 TIMERx\_CHCCx 寄存器的值比较的结果决定。当比较的值相等时状态寄存器(TIMERx\_STR)的CHxIF位将置1。根据CHxDE或CHxIE位的值,一次DMA请求或中断将被产生。

在输出模式,配置CHxOM位域为101,通道将持续输出有效电平。配置CHxOM位域为100通道将持续输出无效电平。

在输出模式,配置CHxOM位域为001,当计数器与TIMERx\_CHCCx寄存器的值相等时,通道输出有效电平。配置CHxOM位域为010,当计数器与TIMERx\_CHCCx寄存器的值相等时,通道输出无效电平。

在输出比较翻转模式(配置CHxOM位域为011),当计数器与TIMERx\_CHCCx寄存器的值相等时,通道输出电平将翻转。

图 9-146. 输出比较模式, 在 OC1 上切换



在PWM输出模式(配置CHxOM位域为110(PWM模式1))或111(PWM模式2))。根据TIMERx\_CARL寄存器和TIMERx\_CHCCx寄存器的值,通道输出PWM波形。

图 9-147. 在 OC1 上输出比较 PWM 模式 1, 向上计数模式

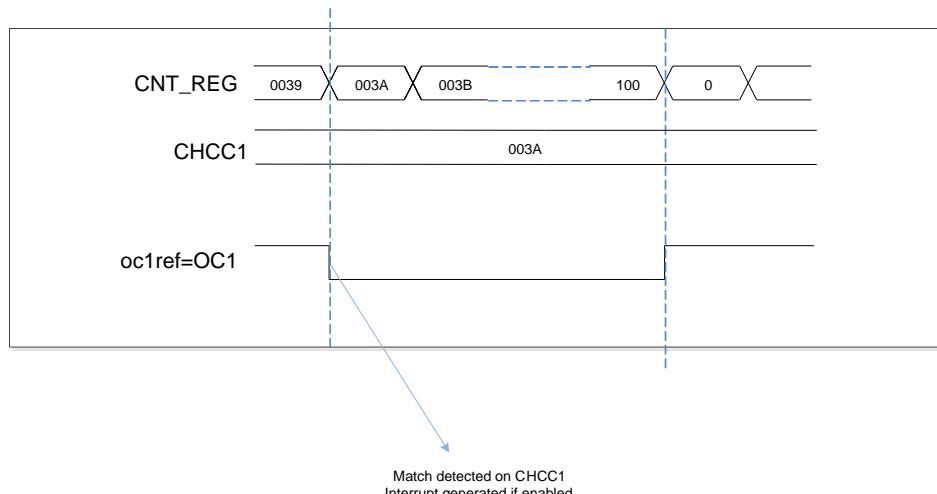
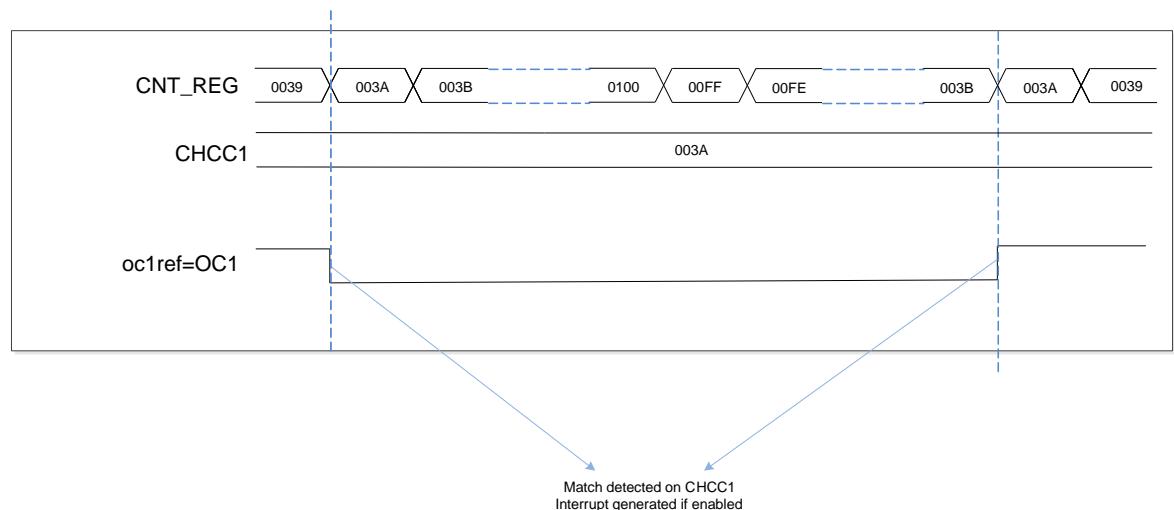


图 9-148. 在 OC1 上输出比较 PWM 模式 1，中央对齐计数模式



### 通道输出参考信号

当TIMER16/TIMER17用于输出匹配比较模式下，设置CHxOM位可以定义OCxREF信号(通道x输出参考信号)类型。OCxREF信号有一些输出功能类型，包括，设置CHxOM=0x00可以保持原始电平，设置CHxOM=0x01可以将OCxREF信号设置为高电平，设置CHxOM=0x02可以将OCxREF信号设置为低电平，在计数值和TIMERx\_CHCCx寄存器的值匹配时，设置CHxOM=0x03可以切换信号。

PWM模式1和PWM模式2输出也是另一种类型的OCxREF输出，具体设置是通过设置CHxOM=0x06或0x07来完成的。在这些模式中，OCxREF信号级别的改变取决于根据计数方向，以及TIMERx\_CHCCx寄存器值和计数值之间的关系。详情参考相应位定义。

设置CHxOM=0x04或0x05可以实现OCxREF信号的强制输出功能。输出比较信号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。

当ETIF信号源于外部TIMERx\_ETI引脚且被设置为高电平(设置TIMERx\_CHCTRL1寄存器的CHxOCE位为1)时，OCxREF信号被强制变为低电平。直到下一次更新事件发生，OCxREF信号才会回到它的有效状态。

### 互补输出及死区时间

TIMER16/TIMER17可以输出两路互补信号。互补信号OCx和OCxN的生效由一些控制位的组合实现：寄存器TIMERx\_CHE的CHxE位和CHxNE位，寄存器TIMERx\_BKDT和TIMERx\_CTRL2的POE，ISOx，ISOxN，IOS和ROS位。输出极性由TIMERx\_CHE寄存器的CHxP位和CHxNP位决定。

如果CHxE，CHxNE和POE位都为1，应该插入死区时间。OCx输出的上升沿和OCxREF的上升沿相比有一个延迟；OCxN输出的上升沿和OCxREF的下降沿相比有一个延迟。延迟的值是一个8位的死区时间计数器，由TIMERx\_BKDT寄存器的DT区域决定。如果延迟值大于有效输出(OCx或OCxN)宽度，将不会产生相应信号脉冲。

图 9-149. 带有死区时间的互补输出

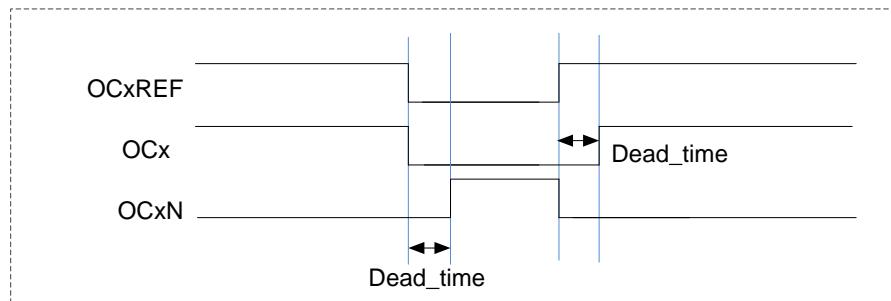


图 9-150. 延迟大于负脉冲的死区时间波形

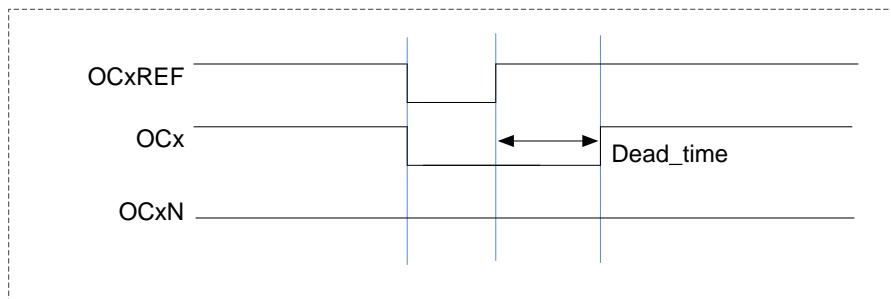
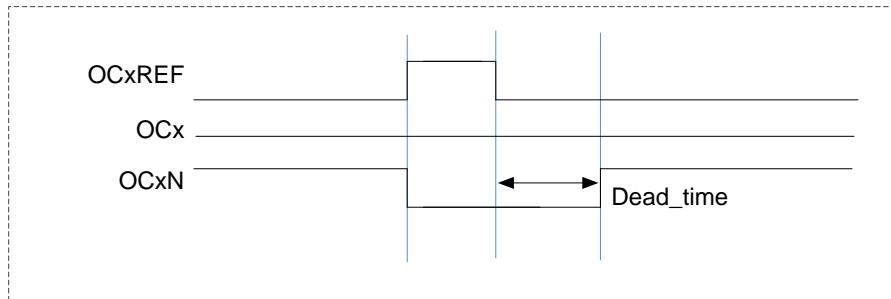


图 9-151. 延迟大于正脉冲的死区时间波形



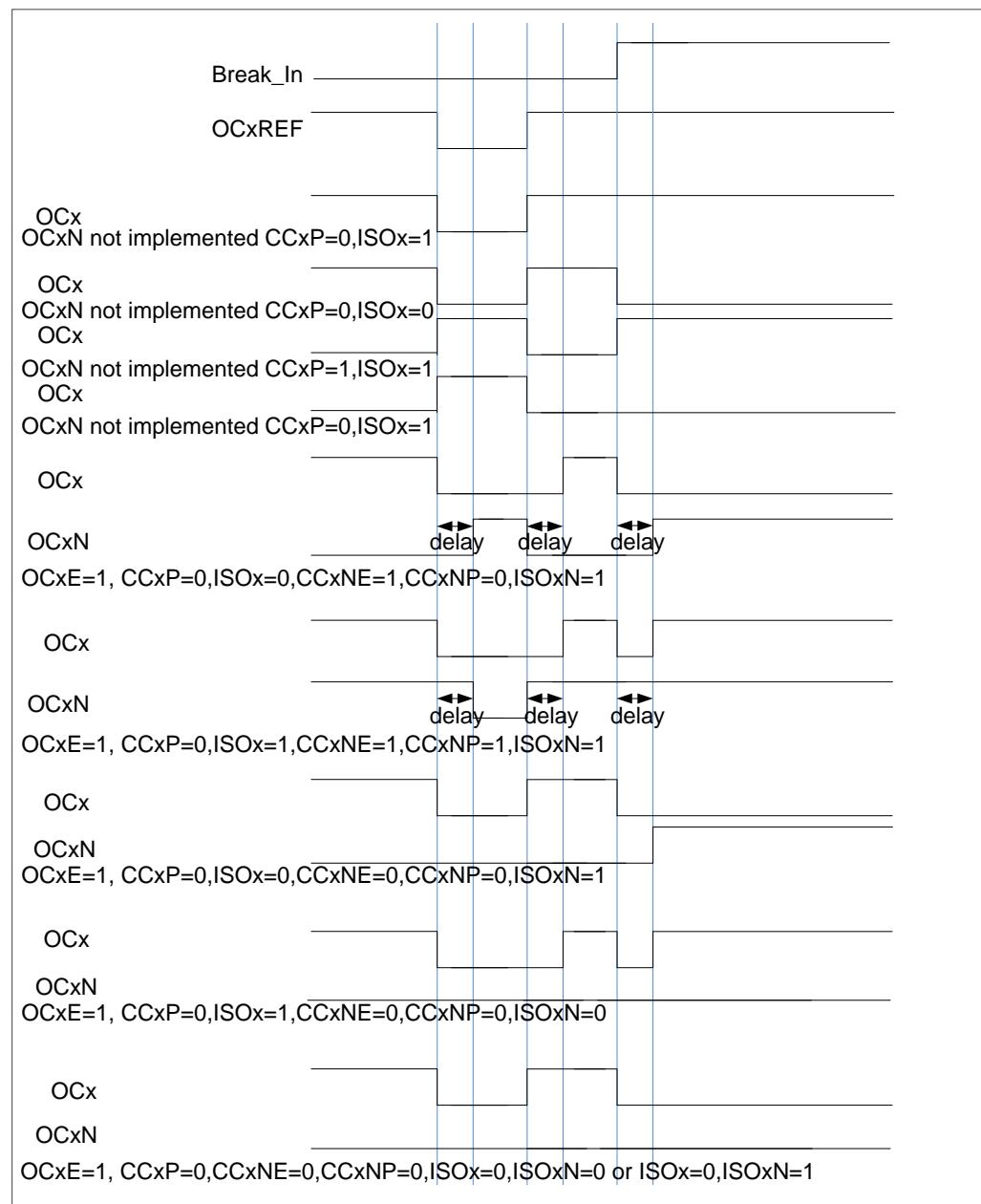
## 中止功能

使用中止功能时，输出OCx和OCxN信号电平被以下位控制，**TIMERx\_BKDT**寄存器的POE, IOS和ROS 位，**TIMERx\_CTLR2**寄存器的ISOx 和ISOxN位。中止事件发生时，OCx和OCxN信号输出不能在同一时间同时处于有效电平上。中止源既可以是中止输入引脚又可以是一个HSE时钟失败事件。时钟失败事件由RCC中的时钟监视器(CKM)产生。将**TIMERx\_BKDT**寄存器的BRKE位置1可以使能中止功能。**TIMERx\_BKDT**寄存器的BRKP位决定了中止输入极性。

发生中止时，POE位被异步地清除，一旦POE位为0，配置**TIMERx\_CR2**寄存器的ISOx位驱动输出OCx和OCxN的电平。如果IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，**TIMERx\_STR**寄存器的BKIF位被置1。如果BKIE=1，中断产生。如果BKDE=1，发送一个DMA请求。

图 9-152. 响应中止的输出表现（中止输入高电平有效）



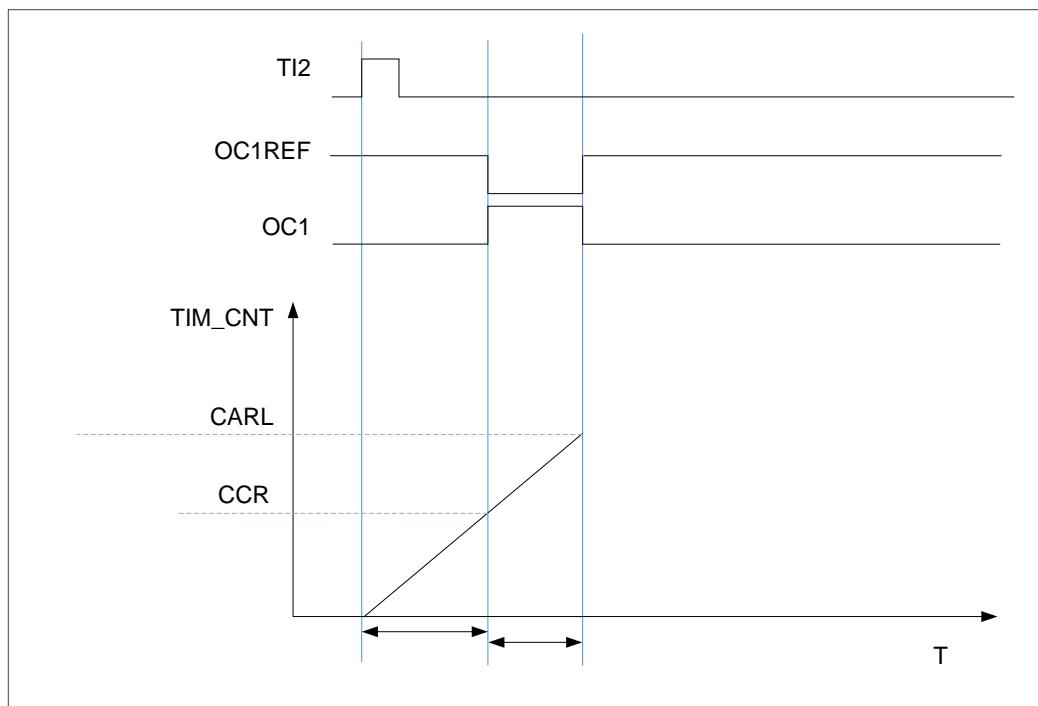
### 单脉冲模式

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx\_CTRL1寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

单脉冲模式下，将CEN位置1产生的触发有效边沿将使能计数器。然而，执行计数值和TIMERx\_CHCCx寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx\_CHCTRL1寄存器的CHxOEF位置1。单脉冲模式下，触发上升沿产生之后，OCxREF信号将被立即强制转换为另一种状态，因为不考虑比较结果的比较匹配事件发生。只有输出通道配置为PWM1或PWM2输出运行模式下时 CHxOEF位才可用，触发源来源于触发

信号。

图 9-153. 单脉冲模式



### 定时器 debug 模式

当微控制器在调试模式，若Cortex™-M3内核停止，且MCUDBG模块中DBG\_TIMERx\_STOP配置位被置1，TIMERx计数器停止。

### 9.6.4. TIMER16/TIMER17 寄存器

#### TIMERx 控制寄存器 1 (TIMERx\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留				CDIV[1:0]		ARSE		保留		SPM		UPS		UPDIS		CEN
						rw		rw			rw		rw		rw	

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CDIV[1:0]	<p>时钟分频</p> <p>软件配置CDIV位规定定时器时钟(PCLK) 和死区时间、采样时钟(DTS)之间的分频系数，该位被死区时间产生器和数字滤波器使用。</p> <ul style="list-style-type: none"> <li>00: <math>f_{DTRGS}=f_{PCLK}</math></li> <li>01: <math>f_{DTRGS}=f_{PCLK}/2</math></li> <li>10: <math>f_{DTRGS}=f_{PCLK}/4</math></li> <li>11: 保留</li> </ul>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁止TIMERx_CARL寄存器的影子寄存器</p> <p>1: 使能TIMERx_CARL寄存器的影子寄存器</p>
6:4	保留	必须保持复位值
3	SPM	<p>单脉冲模式</p> <p>0: 在发生更新事件时，计数器不停止</p> <p>1: 在发生下一次更新事件( 清除CEN 位)时，计数器停止</p>
2	UPS	<p>更新请求源</p> <p>软件通过该位选择更新事件源。</p> <p>0: 使能后，下述任一事件产生更新中断或DMA请求：</p> <ul style="list-style-type: none"> <li>- UPG位被置1</li> <li>- 计数器溢出/下溢</li> <li>- 从模式控制器产生的更新</li> </ul> <p>1: 使能后只有计数器溢出/下溢才产生更新中断或DMA 请求。</p>
1	UPDIS	<p>禁止更新</p> <p>软件通过该位允许/ 禁止更新事件的产生</p> <p>0: 允许更新事件产生。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值：</p> <ul style="list-style-type: none"> <li>- UPG位被置1</li> </ul>

- 计数器溢出/下溢
  - 从模式控制器产生一个更新事件
- 1: 禁止更新事件。带有缓存的寄存器保持原有值，如果UG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化

0	CEN	使能计数器
0:	禁止计数器	
1:	使能计数器	在软件将CEN位置1后，外部时钟、暂停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。

### TIMERx 控制寄存器 2 (TIMERx\_CTLR2)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					ISO1N	ISO1	保留			DMAS	CCUC	保留	CCSE		
					rw	rw				rw	rw		rw		

位/位域	名称	描述
15:11	保留	必须保持复位值
9	ISO1N	通道1的互补输出空闲状态 0: 当POE=0, OC1N=0. 1: 当POE=0, OC1N=1 只有当TIMERx_BKDT 寄存器的LK[1:0]位为00时此位才可以被修改。
8	ISO1	通道1的空闲输出状态 0: 当POE=0, OC1=0 1: 当POE=0, OC1=1 如果OC1N生效，一个死区时间后OC1输出改变。此位只有在TIMERx_BKDT 寄存器的LK[1:0]位为00的时候可以被更改。
7:4	保留	必须保持复位值
3	DMAS	DMA请求源选择 0: 当通道x事件发生，发送通道x的DMA请求 1: 当更新事件发生，发送通道x的DMA请求
2	CCUC	捕获/比较控制影子寄存器更新控制 使能(CCSE=1)捕获/比较控制影子寄存器(CHxE、OCxNE和CHXOM 位)后，该位控制这些影子寄存器更新的时间。 0: CCUG位被置1时更新影子寄存器 1: 当CCUG位被置1或检测到TRGI上升沿时，影子寄存器更新

当通道没有互补输出时，此位无效。

1	保留	必须保持复位值
0	CCSE	捕获/比较控制影子寄存器使能 0: 影子寄存器CHxE、OCxNE和CHXOM 位被禁止 1: 使能影子寄存器CHxE、OCxNE和CHXOM 位，当这些位被写时，只有当CCUG 位被置1时这些位才被更新 当通道没有互补输出时，此位无效。

### TIMERx DMA 和中断使能寄存器 (TIMERx\_DIE)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							CH1DE	UPDE	BKIE	保留	CCUIE		保留	CH1IE	UPIE
							rw	rw	rw		rw			rw	rw

位/位域	名称	描述
15:10	保留	必须保持复位值.
9	CH1DE	通道1 DMA请求使能 0: 禁止通道1 DMA请求 1: 使能通道1 DMA请求
8	UPDE	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	保留	必须保持复位值.
5	CCUIE	通道控制更新中断使能 0: 禁止通道控制更新中断 1: 使能通道控制更新中断
4:2	保留	必须保持复位值.
1	CH1IE	通道1中断使能 0: 禁止通道1中断 1: 使能通道1中断
0	UPIE	更新中断使能 0: 禁止更新中断

1: 使能更新中断

### TIMERx 状态寄存器 (TIMERx\_STR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH1OF	保留	BKIF	保留.	CCUIF	保留				CH1IF	UPIF	rc_w0 rc_w0

位/位域	名称	描述
15:11	保留	必须保持复位值
10	CH2OF	通道2重复捕获标志 参见CH1OF描述
9	CH1OF	通道1重复捕获标志 当通道1被配置为输入模式时，在CH1IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。 0: 无重复捕获发生 1: 发生了重复捕获中断
8	保留	必须保持复位值
7	BKIF	中止中断标志位 一旦中止输入有效，由硬件对该位置‘1’。如果中止输入无效，则该位可由软件清‘0’。 0: 无中止事件产生 1: 中止输入上检测到有效电平
6	保留	必须保持复位值
5	CCUIF	通道控制更新中断标志 当通道控制更新事件发生时此标志位被硬件置1，此位由软件清0。 0: 无通道控制更新中断发生 1: 通道控制更新中断发生
4:2	保留	必须保持复位值
1	CH1IF	通道1中断标志 此标志由硬件置1软件清0。当通道1在输入模式下时，捕获事件发生时此标志位被置1；当通道1在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道1中断发生 1: 通道1中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。

0: 无更新中断发生

1: 发生更新中断

### TIMERx 事件发生寄存器 (TIMERx\_EVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					保留			BKG	保留	CCUG		保留		CH1G	UPG

W                    W                    W                    W

位/位域	名称	描述
15:8	保留	必须保持复位值
7	BKG	<p>产生中止事件</p> <p>该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POE位被清0且BKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。</p> <p>0: 不产生中止事件</p> <p>1: 产生中止事件</p>
6	保留	必须保持复位值
5	CCUG	<p>通道控制更新事件发生</p> <p>此位由软件置1，由硬件自动清0。当此位被置1，通道控制寄存器 (CHxE, OCxNE和CHXOM 位) 的互补输出被更新。</p> <p>0: 不产生通道控制更新事件</p> <p>1: 产生通道控制更新事件</p>
4:2	保留	必须保持复位值
1	CH1G	<p>通道1捕获或比较事件发生</p> <p>该位由软件置1，用于在通道1产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CC1IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道1配置为输入模式，计数器的当前值被TIMER1_CHCC1寄存器捕获，如果CH1IF标志位已经为1，则CH1OF标志位被置1。</p> <p>0: 不产生通道1捕获或比较事件</p> <p>1: 发生通道1捕获或比较事件</p>
0	UPG	<p>此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。</p> <p>0: 无更新事件产生</p> <p>1: 产生更新事件</p>

### TIMERx 通道控制寄存器 1 (TIMERx\_CHCTRL1)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CH1OCE	CH1OM[2:0]		CH1OSE	CH1OFE	CH1M[1:0]		
CH1ICF[3:0]								CH1ICP[1:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

#### 输出比较模式:

位/位域	名称	描述
15:8	保留	必须保持复位值
7	CH1OCE	<p>通道1输出比较清0使能</p> <p>当此位被置1，当检测到ETIF输入高电平时，OC1参考信号被清0</p> <p>0: 使能通道1输出比较清零</p> <p>1: 禁止通道1输出比较清零</p>
6:4	CH1OM	<p>通道1输出比较模式</p> <p>此位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。</p> <p>OC1REF高电平有效，而OC1、OC1N的有效电平取决于CH1P、CH1NP位。</p> <p>000: 冻结。输出比较寄存器TIMERx_CHCC1与计数器间的比较对OC1REF不起作用</p> <p>001: 匹配时设置为高。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时，强制OC1REF为高。</p> <p>010: 匹配时设置为低。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时，强制OC1REF为低。</p> <p>011: 匹配时翻转。当计数器的值与捕获/比较寄存器TIMERx_CHCC1相同时，强制OC1REF翻转。</p> <p>100: 强制为低。强制OC1REF为低电平</p> <p>101: 强制为高。强制OC1REF为高电平</p> <p>110: PWM模式1。在向上计数时，一旦计数器值小于TIMERx_CHCC1时，OC1REF为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CHCC1时，OC1REF为低电平，否则为高电平。</p> <p>111: PWM模式2。在向上计数时，一旦计数器值小于TIMERx_CHCC1时，OC1REF为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CHCC1时，OC1REF 为高电平，否则为低电平。</p> <p>在PWM模式1或PWM模式2中，只有当比较结果改变了或者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> <p>当TIMERx_BKDT寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。</p>
3	CH1OSE	<p>通道1输出比较影子寄存器使能</p> <p>当此位被置1，TIMERx_CHCC1寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。</p>

0: 禁止通道1输出/比较影子寄存器

1: 使能通道1输出/比较影子寄存器

仅在单脉冲模式下(**TIMERx\_CTR1**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。

当**TIMERx\_BKDT**寄存器的LK[1:0]=11且CH1M=00时此位不能被改变。

2	<b>CH1OFE</b>	通道1输出比较快速使能 该位用于通道配置为 <b>PWM</b> 模式时加快CC输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，OC被设置为比较电平而与比较结果无关。 0: 禁止通道1输出比较快速. 当触发器的输入有一个有效沿时，激活CC1输出的最小延时为5个时钟周期 1: 使能通道1输出比较快速。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3个时钟周期
1:0	<b>CH1M</b>	通道1模式选择 这些位定义了输入的方向和输入信号的选择。只有当通道关闭( <b>TIMERx_CHE</b> 寄存器的 <b>CH1E</b> 位被清0)时这些位才可写。 00: 通道1配置为输出 01: 通道1配置为输入，IC1映射在TI1上 10: 通道1配置为输入，IC1映射在TI2上 11: 通道1配置为输入，IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(通过设置 <b>TIMER_SMC</b> 寄存器的 <b>TRGS</b> 位)。

#### 输入捕获模式:

位/位域	名称	描述
15:8	保留	必须保持复位值
7:4	<b>CH1ICF</b>	通道1输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。 这些位定义了TI1输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器， $f_{SAMP}=f_{DTS}$ , N=1 0001: $f_{SAMP}=f_{PCLK}$ , N=2 0010: $f_{SAMP}=f_{PCLK}$ , N=4 0011: $f_{SAMP}=f_{PCLK}$ , N=8 0100: $f_{SAMP}=f_{DTS}/2$ , N=6 0101: $f_{SAMP}=f_{DTS}/2$ , N=8 0110: $f_{SAMP}=f_{DTS}/4$ , N=6 0111: $f_{SAMP}=f_{DTS}/4$ , N=8 1000: $f_{SAMP}=f_{DTS}/8$ , N=6 1001: $f_{SAMP}=f_{DTS}/8$ , N=8 1010: $f_{SAMP}=f_{DTS}/16$ , N=5 1011: $f_{SAMP}=f_{DTS}/16$ , N=6 1100: $f_{SAMP}=f_{DTS}/16$ , N=8 1101: $f_{SAMP}=f_{DTS}/32$ , N=5

1110:  $f_{SAMP}=f_{DTS}/32$ , N=6

1111:  $f_{SAMP}=f_{DTS}/32$ , N=8

3:2	CH1CP	通道1输入捕获预分频器 这2位定义了CC1 输入的预分频系数。一旦CH1E =0(TIMERx_CHE寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH1M	通道1模式选择 这两位定义了通道方向和输入信号的选择。只有在通道关闭(TIMERx_CHE寄存器的CH1E位被清0)时这些位才可写。 00: 通道1配置为输出 01: 通道1配置为输入, IC1映射在TI1上 10: 通道1配置为输入, IC1映射在TI2上 11: 通道1配置为输入, IC1映射在TRC上. 此模式仅工作在内部触发输入被选中时(通过设置TIMER_SMC寄存器的TRGS位)

### TIMERx 通道使能寄存器 (TIMERx\_CHE)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CH1NP	CH1NE	CH1P	CH1E
												rw	rw	rw	rw

位/位域	名称	描述
15:4	保留	必须保持复位值
3	CH1NP	通道1互补输出极性 当通道1配置为输入模式, 此位定义了互补输出信号的极性。 0: 通道1高电平有效 1: 通道1低电平有效 当TIMERx_BKDT寄存器的LK[1:0]=11或10时此位不能被更改。
2	CH1NE	通道1互补输出使能 当通道1配置为输出模式时, 将此位置1使能通道1的互补输出。 0: 禁止通道1互补输出 1: 使能通道1互补输出
1	CH1P	通道1极性 当通道1配置为输入模式时, 此位定义了IC1信号极性。当通道1配置为输出模式时,

此位定义了输出信号极性。

0: 通道1高电平有效

1: 通道1低电平有效

当TIMERx\_BKDT寄存器的LK[1:0]=11或10时此位不能被更改。

0	CH1E	通道1使能
		当通道1配置为输入模式时，将此位置1使能OC1信号有效。当通道1配置为输出模式时，将此位置1使能通道1上的捕获事件。
		0: 禁止通道1
		1: 使能通道1

### TIMERx 计数器寄存器 (TIMERx\_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

### TIMERx 预分频寄存器 (TIMERx\_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

### TIMERx 计数器自动重载寄存器 (TIMERx\_CARL)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

### **TIMERx 重复计数寄存器 (TIMERx\_CREP)**

地址偏移: 0x30

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:8	保留	必须保持复位值
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

### **TIMERx 通道 1 捕获/比较寄存器 (TIMERx\_CHCC1)**

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHCC1[15:0]															
rw															

位/位域	名称	描述
15:0	CHCC1[15:0]	通道1的捕获或比较值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

**TIMERx 中止和死区时间寄存器 (TIMERx\_BKDT)**

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POE	OAE	BRKP	BRKE	ROS	IOS	LK[1:0]									DT[7:0]

rw      rw      rw      rw      rw      rw      rw                                                        rw

位/位域	名称	描述
15	POE	主输出使能 根据OAE位的设置值，该位可以由软件清0或被硬件自动置1。一旦中止输入有效，该位被硬件异步清0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHE寄存器的CHxE、CHxNE位)，则开启OC 和OCN 输出。 0: 禁止OC 和OCN输出或强制为空闲状态 1: 使能通道输出
14	OAE	自动输出使能 此位定义了POE位是否可以被硬件自动置1。 0: POE位不能被硬件置1 1: 如果中止输入无效，下一次更新事件发生时，POE位能被硬件自动置1。 此位只有在TIMERx_BKDT寄存器的LK[1:0]=00时才可修改。
13	BRKP	中止极性 此位定义了中止输入信号的极性。 0: 中止输入低电平有效 1: 中止输入高电平有效
12	BRKE	中止使能 此位置1使能中止事件和CCS时钟失败事件输入。 0: 禁止中止输入 1: 使能中止输入 只有在TIMERx_BKDT寄存器的LK[1:0]=00时此位才可修改。
11	ROS	运行模式下“关闭状态”配置 当POE位被置1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。 0: 当POE位被置1，通道输出信号(OC/OCN)被禁止 1: 当POE位被置1，通道输出信号(OC/OCN)被使能，和TIMERx_CHE寄存器CHxE/CHxNE位有关 此位在TIMERx_BKDT寄存器的LK[1:0]=10或11时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当POE位被清0，此位定义了已经配置为输出模式的通道的输出状态。 0: 当POE位被清0，通道输出信号(OC/OCN)被禁止 1: 当POE位被清0，通道输出信号(OC/OCN)被使能，和TIMERx_CHE寄存器CHxE/CHxNE位有关

CHxE/CHxNE位有关。

此位在TIMERx\_BKDT寄存器的LK[1:0]=10或11时不能被更改。

9:8	LK[1:0]	锁定控制 这两位定义了寄存器的写保护特性。 00: 禁止锁定, 无写保护 01: 锁定模式1, TIMERx_CTLR2寄存器的ISOx/ISOxN位和TIMERx_BKDT寄存器的BRKE/BRKP/OAE/DT位写保护 10: 锁定模式2, 除了锁定模式1下的寄存器, TIMERx_CHE寄存器的CHxP/CHxNP位 (如果相应通道配置为输出模式) 和TIMERx_BKDT寄存器的ROS/IOS 位写保护 11: 锁定模式3, 除了锁定模式2下的寄存器, TIMERx_CHCTRLRx寄存器的CHxOM/CHxOSE (如果相应通道配置为输出)位写保护。 系统复位后这两位只能被写一次, 一旦TIMERx_BKDT寄存器被写入, 这两位被写保护。
7:0	DT[7:0]	死区时间值 这些位定义了插入互补输出之间的死区持续时间。DT值和死区时间的关系如下: DT[7:5]=0xx: DT=DT[7:0]x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTRGS</sub> . DT[7:5]=10x: DT=(64+DT[5:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTRGS</sub> *2. DT[7:5]=110: DT =(32+DT[4:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTRGS</sub> *8. DT[7:5]=111: DT =(32+DT[4:0])x t <sub>DT</sub> , t <sub>DT</sub> =t <sub>DTRGS</sub> *16. 此位只有在TIMERx_BKDT寄存器的LK[1:0]=00时才可修改。

### TIMERx DMA 控制寄存器 (TIMERx\_DCTRL)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:14	保留	必须保持复位值
12:8	DBLTH[4:0]	DMA连续传送长度 这5位定义了DMA在连续模式下的传送长度( 当对TIMERx_DTRSF寄存器进行读或写时, 定时器则进行一次连续传送)。
7:5	保留	必须保持复位值
4:0	DBAR[4:0]	DMA地址 这5位定义了DMA 传送的地址( 当对TIMx_DTRSF寄存器进行读或写时, DBA定义为从TIMERx_CTLR1 寄存器所在地址开始的偏移量)。

### TIMERx DMA 传送寄存器 (TIMERx\_DTRSF)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	DTRSF[15:0]	<p>DMA传送</p> <p>对TIMERx_DTRSF寄存器的读或写会导致对从TIMERx_CTRL1 寄存器开始的以下地址范围所在寄存器的访问:</p> <p>(DBAR + 连续传送计数器) × 4</p> <p>连续传送计数器由硬件计算, 范围为0到DBLTH。</p>

### TIMERx 配置寄存器 (TIMERx\_CFGR) (仅适用于 GD32F170xx and GD32F190xx 产品)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CCSEL	<p>写捕获比较寄存器选择位</p> <p>此位由软件写1或清0。</p> <p>1: 当写入捕获比较寄存器的值与寄存器当前值相等时, 写入操作无效</p> <p>0: 无效</p>
0	OUTSEL	<p>输出值选择位</p> <p>此位由软件写1或清0。</p> <p>1: 如果POE位与IOS位均为0, 则输出无效</p> <p>0: 无效</p>

## 10. 红外线接口 (IFRR)

### 10.1. 简介

红外线接口(IFRR)用来控制发射红外光的LED，该LED可发射红外数据来实现红外遥控。

该模块无寄存器，受TIMER16定时器和TIMER17定时器控制。通过设置GPIO引脚为高速模式，你可以提高模块的输出电流能力。

### 10.2. 主要特性

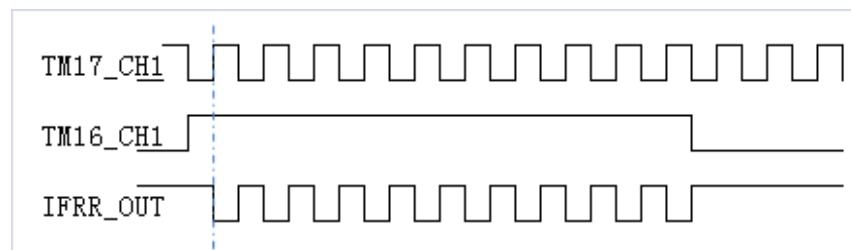
- 通过使能 GPIO 口的备用功能控制位来进行使能 IFRR 的输出功能
- IFRR 输出信号由 TIMER16 定时器的通道 1 和 TIMER17 定时器的通道 1 决定
- 为了获取正确的红外信号，TIMER16 定时器应该产生低频调制包络信号，TIMER17 应该产生高频载波信号
- PB9 能够提供高电流输出驱动 LED 接口

### 10.3. 功能描述

IFRR模块整合了TIMER16定时器和TIMER17定时器的输出来产生红外信号。

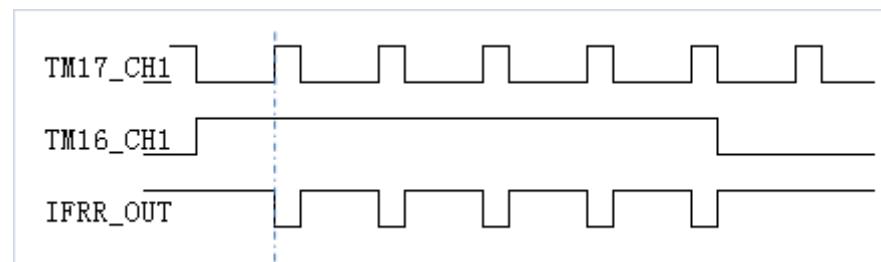
1. 通过对TIMER16定时器的通道1编程输出低频PWM信号来产生调制包络信号，对TIMER17定时器的通道1编程输出高频PWM信号来产生载波信号。产生信号之前需要开启这些通道。
2. 配置GPIO口为复用，并使能这些引脚。
3. 如果你想获得更高的驱动电流输出，需要将IFRR\_OUT映射到PB9口上，并且在SYS\_CFG模式下通过相应寄存器设置PB9口为高速模式。

图 10-1. IFRR 输出时序图 1



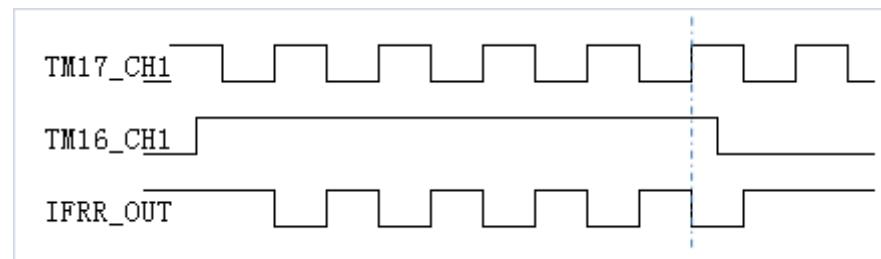
注意：IFRR\_OUT比TIMER17时钟通道1输出信号有一个APB时钟延迟。

图 10-2. IFRR 输出时序图 2



注意：载波的占空比可以被改变，在TIMER16定时器的通道1输出电平为高时，IFRR\_OUT和TIMER17定时器的通道1输出信号反相。

图 10-3. IFRR 输出时序图 3



注意：IFRR\_OUT将保持TIMER17定时器通道1输出信号的完整，即使TIMER16定时器的包络信号无效。

## 11. 看门狗 (WDG)

### 11.1. 独立看门狗 (IWDG)

#### 11.1.1. 简介

看门狗定时器是一个硬件计时电路，用来监测由软件故障导致的系统故障。GD32F1x0 有两个看门狗外设，独立看门狗和窗口看门狗。它们使用灵活，并提供了很高的安全水平和精准的时间控制。两个看门狗都是用来解决软件故障问题的。当它的定时计数值达到了预设的门限的时候，它会触发一个系统复位(对于窗口看门狗来说是个中断)。当处理器工作在调试模式的时候看门狗定时计数器可以停止计数。独立看门狗寄存器的写保护功能可以防止意料之外的配置更改。

独立看门狗(IWDG)有独立的时钟源 (LSI)。因此就算是主时钟失效了，它仍然能保持工作状态，这非常适合于需要独立环境而对计时精度要求不高的场合。窗口看门狗 (WWDG)适用于需要精确计时的场合。

另外，独立看门狗有一个可配置的窗口值，如果软件在计数器达到窗口值之前重装载计数器，将产生一个复位请求。

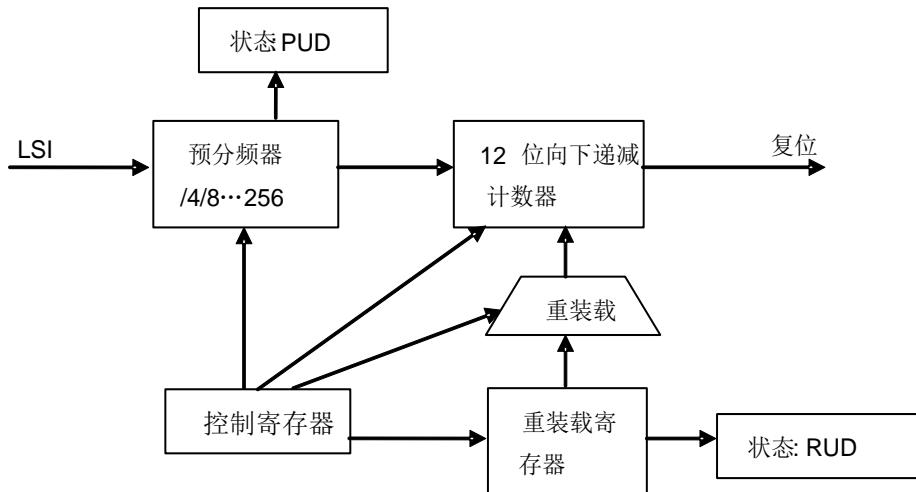
#### 11.1.2. 主要特性

- 自由运行的 12 位向下计数器
- 如果看门狗被使能，那么当向下计数器的值达到 0 时产生系统复位
- 如果看门狗被使能，当向下计数器的值处于窗口值之外时，重装载会导致产生系统复位
- 独立时钟源，独立看门狗在主时钟故障(例如待机和深度睡眠模式下)时仍能工作
- 独立看门狗硬件控制位，可以用来控制是否在上电时自动启动独立看门狗
- 独立看门狗调试模式配置位，可以用来控制当微控制器进入调试模式时(内核被暂停)，独立看门狗计数器是否继续运行

#### 11.1.3. 功能描述

独立看门狗带有一个 8 级预分频器和一个 12 位的向下递减计数器。参考下图的独立看门狗的功能模块。

图 11-1. 独立看门狗框图



向控制寄存器(IWDG\_CTLR)中写 0xCCCC 可以开启独立看门狗，计数器开始向下计数。当计数器记到 0x000，产生一次复位。

在任何时候向控制寄存器中写 0xAAAA 都可以重装载计数器，重装载值来源于 IWDG\_RLDR 寄存器。软件可以在计数器计数值达到 0x000 之前可以通过重装载计数器来阻止看门狗复位。

独立看门狗也能够工作在窗口看门狗模式下，只要在 IWDG\_WND 寄存器中设置适当的值即可。如果重加载操作执行的同时，看门狗计数器的值超出了窗口寄存器(IWDG\_WND) 中存储的值，也会引起系统复位。IWDG\_WND 的默认值是 0x00000FFF，所以如果没有改写它，那么窗口选项默认是关闭的。窗口值一旦改变，立即就会引起看门狗计数器的一次重加载动作，将向下递减计数器置为 IWDG\_RLDR 中的值，并复位预分频计数器。

如果在可选字节中(OB\_USER)打开了"硬件看门狗" 功能，那么在上电的时候看门狗就被自动打开。软件应该在计数器达到 0x000 之前重装载计数器。

IWDG\_PSR 寄存器、IWDG\_RLDR 寄存器和 IWDG\_WND 寄存器有写保护功能。想要写入数据到这些寄存器之前，需要写 0x5555 到控制寄存器中。写其他任何值到控制寄存器中将会再次启动对这些寄存器的写保护。当预分频器寄存器 (IWDG\_PSR) 或者重装载寄存器 (IWDG\_PSR) 或者窗口寄存器 (IWDG\_WND) 更新时，IWDG\_STR 寄存器的状态位应该被置 1。

如果在 MCU 调试模块中的 IWDG\_HOLD 位被清 0，即使 Cortex™-M3 内核停止(调试模式下)独立看门狗依然工作。如果 IWDG\_HOLD 位被置 1，独立看门狗将在调试模式下停止工作。

表 11-1. 独立看门狗在 40 kHz (LSI)时的最小/最大暂停时间

预分频系数	PS[2:0] 位	最小暂停时间(ms) RL[11:0]=0x000	最大暂停时间 (ms) RL[11:0]=0xFFFF
1/4	000	0.1	409.6
1/8	001	0.2	819.2
1/16	010	0.4	1638.4
1/32	011	0.8	3276.8
1/64	100	1.6	6553.6
1/128	101	3.2	13107.2

预分频系数	PS[2:0] 位	最小暂停时间(ms) RL[11:0]=0x000	最大暂停时间 (ms) RL[11:0]=0xFFFF
1/256	110 or 111	6.4	26214.4

校准 LSI 时钟可以使独立看门狗暂停时间更加精确。

#### 11.1.4. 独立看门狗寄存器

##### 控制寄存器 (IWDG\_CTLR)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器可以按半字 (16 位) 或字 (32 位) 访问

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

w0

位/位域	名称	说明
31:16	保留	必须保持复位值
15:0	CT[15:0]	只可写, 写入不同的值来产生不同的功能 0x5555: 关闭IWDG_PSR, IWDG_RLDR和IWDG_WND寄存器的写保护 0xCCCC: 开启独立看门狗计数器。计数减到0时产生中断 0xAAAA: 重装载计数器

##### 预分频寄存器 (IWDG\_PSR)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	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															

rw

位/位域	名称	说明
------	----	----

31:3	保留	必须保持复位值
2:0	PS[2:0]	独立看门狗计时预分频选择。写这些位之前要通过向IWDG_CTLR寄存器写0x5555去除写保护。在改写这个寄存器的过程中，IWDG_STR寄存器的PUD位被置1，此时读取此寄存器的值都是无效的。
	000: 1/4	001: 1/8
	011: 1/32	100: 1/64
	110: 1/256	111: 1/256
	如果应用需要使用一些预分频系数，改变预分频值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零除非进入省电模式。	

### 重加载寄存器 (IWDG\_RLDR)

地址偏移: 0x08

复位值: 0x0000 0FFF

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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				RLD [11:0]											

rw

位/位域	名称	说明
31:12	保留	必须保持复位值
11:0	RLD[11:0]	<p>独立看门狗计数器重加载值，向IWDG_KR寄存器写入0xFFFF的时候，这个值会被更新到看门狗计数器中。</p> <p>这些位有写保护功能。在写这些位之前需向IWDG_CTLR寄存器中写0x5555解除写保护。在改写这个寄存器的过程中，IWDG_STR寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。</p> <p>如果应用需要使用一些预分频系数，改变重加载值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零除非进入省电模式。</p>

### 状态寄存器 (IWDG\_STR)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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												WUD	RUD	PUD	
												ro	ro	ro	

位/位域	名称	说明
31:2	保留	必须保持复位值
2	WUD	看门狗计数器窗口值更新 IWDG_WND寄存器写操作时，该位被置1，此时读取IWDG_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗计数器重装载值更新 IWDG_RLDR寄存器写操作时，该位被置1，此时读取IWDG_RLDR寄存器的任何值都是无效的。
0	PUD	独立看门狗预分频值更新 IWDG_PSR寄存器写操作时，该位被置1，此时读取IWDG_PSR寄存器的任何值都是无效的。

### 窗口寄存器 (IWDG\_WND)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	说明
31:12	保留	必须保持复位值
11:0	WND[11:0]	看门狗计数器窗口值。这些位包含的是窗口值。当计数器计数值超过该寄存器中的值时，重装载操作将会产生复位。在改变重装载值时，IWDG_SR寄存器中的WUD位必须位于0状态。 这些位有写保护功能。在写这些位之前需向IWDG_CTLR寄存器中写0x5555解除写保护。 如果应用需要使用窗口值，改变窗口值之前必须等到PUD位被清0。更新了窗口寄存器中的值后，在代码持续执行之前不必等待PUD值被清零除非进入省电模式。

## 11.2. 窗口看门狗 (WWDG)

### 11.2.1. 简介

窗口看门狗(WWDG)用来监测由软件故障导致的系统故障。窗口看门狗开启后，7位向下递减计数器值逐渐减小。计数值达到 0x3F 时会产生看门狗复位(T6 位被清 0)。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生看门狗复位。因此软件需要在给定的区间内更新计数器。窗口看门狗有一个提前唤醒中断，当计数器计数值达到 0x40 时，在中断使能的前提下看门狗将产生一次中断。

窗口看门狗时钟是由 APB1 时钟预分频而来。

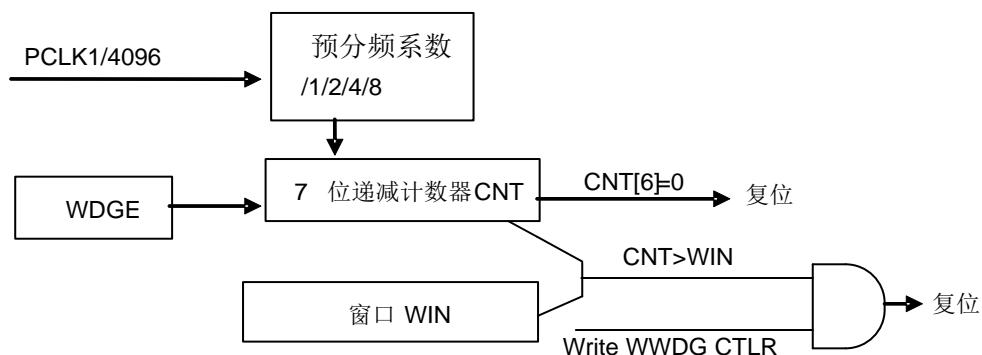
### 11.2.2. 主要特性

- 可编程的 7 位自由运行向下递减计数器
- 当窗口看门狗使能以后有条件的复位。当计数值达到 0x3F，或者计数器在计数值比窗口寄存器的值更高的时候更新的时候，都会产生看门狗复位。
- 提前唤醒中断(EWI)：如果看门狗打开，中断允许，计数值达到 0x40 的时候会产生中断。
- 窗口看门狗调试模式配置位，通过配置该位，可以使窗口看门狗在调试模式下选择停止还是继续工作。

### 11.2.3. 功能描述

如果窗口看门狗使能(将 WWDG\_CTLR 寄存器的 WDGEN 位置 1)，计数值达到 0x3F 的时候产生看门狗复位(CNT[6]位被清 0)。在计数值达到窗口寄存器的值时，如果更新计数器也会产生看门狗复位。

图 11-2. 窗口看门狗框图



软件应该通过写 WWDG\_CTLR 寄存器的方式重装载递减计数器来阻止 MCU 复位。递减计数器的值降到 0x3F 之前并且比窗口值更小的时候应该写 WWDG\_CTLR 寄存器。重装载值必须在 0xFF 和 0xC0 之间。

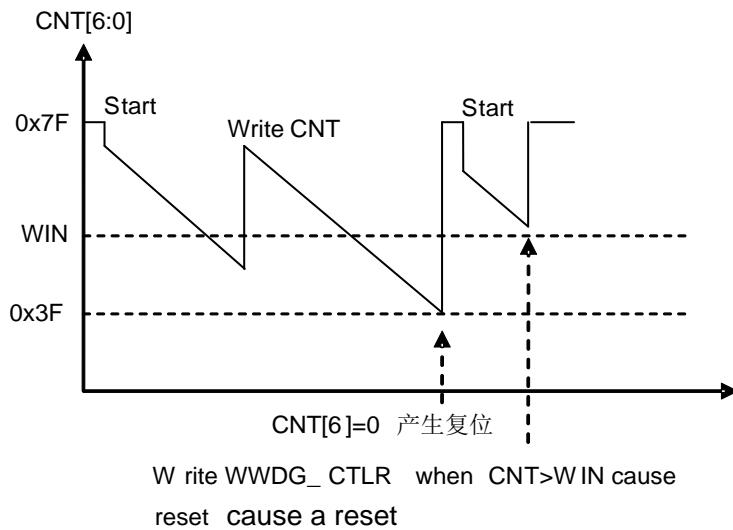
复位之后看门狗总是关闭的。软件可以向 WWDG\_CTLR 的 WDGEN 写 1 开启看门狗。窗口

看门狗打开后，计数器始终递减计数，计数器的值应该大于 0x3F，也就是说 T6 位应该被置位。CNT[5:0] 决定了两次重加载的最大间隔时间。递减速度取决于 APB1 时钟和预分频器 (WWDG\_CFR 寄存器的 PS[1:0] 位)。配置寄存器 (WWDG\_CFR) 包含了窗口值 (WIN[6:0])，递减计数器必须在它的值小于窗口值大于 0x3F 的时候重装载，否则产生看门狗复位。

对 WWDG\_CFR 寄存器的 EWI 位置 1 可以使能提前唤醒中断 (EWI)，当计数值达到 0x40 的时候该中断产生。同时可以用相应的中断服务程序 (ISR) 来触发特定的行为 (例如通信或数据记录)，来分析软件故障的原因以及在器件复位的时候挽救重要数据。此外，软件重装载 ISR 中的计数器来管理软件系统检查等等。在这种情况下，窗口看门狗将永远不会复位但是可以用于其他地方。

向 WWDG\_STR 寄存器的 EWIF 位写 0 可以清 EWI 中断。

图 11-3. 窗口看门狗时序图



计算窗口看门狗超时的公式如下：

$$t_{\text{WWDG}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{PS}} \times (\text{cnt}[5:0] + 1) \text{ (ms)}$$

其中：

$t_{\text{WWDG}}$ : WWDG 超时时间

$t_{\text{PCLK1}}$ : APB1 以 ms 为单位的时钟周期

$t_{\text{WWDG}}$  的最大值和最小值请参考下表。

表 11-2. MHz (fPCLK1) 时的最大最小值

预分频系数	PS[1:0]	最小超时 CNT[6:0]=0x40	最大超时 CNT[6:0]=0x7F
1/1	00	113 μs	7.28 ms
1/2	01	227 μs	14.56 ms
1/4	10	455 μs	29.12 ms
1/8	11	910 μs	58.25 ms

如果 MCU 调试模块中的 WWDG\_HOLD 位被清 0，即使 Cortex™-M3 内核停止工作 (调试模

式下), 窗口看门狗也可以继续工作。当 WWDG\_HOLD 位被置 1 时, 窗口看门狗在调试模式下停止。

### 11.2.4. 窗口看门狗寄存器

#### 控制寄存器 (WWDG\_CTLR)

地址偏移: 0x00

复位值: 0x0000 007F

该寄存器可以按半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	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								WDGEN	CNT[6:0]							
rs								rw								

位/位域	名称	说明
31:8	保留	必须保持复位值
7	WDGEN	开启窗口看门狗, 硬件复位的时候清0, 写0无效。 0: 关闭窗口看门狗 1: 开启窗口看门狗
6:0	CNT[6:0]	看门狗计数器的值。当计数值从0x40降到0x3F时, 产生看门狗复位。当计数器值高于窗口值的时候, 写计数器可以产生看门狗复位。

#### 配置寄存器 (WWDG\_CFR)

地址偏移: 0x04

复位值: 0x0000 007F

该寄存器可以按半字 (16 位) 或字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	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								EWI	PS[1:0]	WIN[6:0]							
rs								rw									

位/位域	名称	说明
31:10	保留	必须保持复位值

9	EWI	提前唤醒中断。如果该位被置1，计数值达到0x40时触发中断。该位由硬件复位清0，写0没有任何作用。
8:7	PS[1:0]	看门狗计数器的预分频系数 00: PCLK1/4096/1 01: PCLK1/4096/2 10: PCLK1/4096/4 11: PCLK1/4096/8
6:0	WIN[6:0]	窗口值，当看门狗计数器的值大于窗口值时，写看门狗计数器(WWDG_CTLR的T位)会产生看门狗复位。

### 状态寄存器 (WWDG\_STR)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	说明
31:1	保留	必须保持复位值
0	EWIF	提前唤醒中断标志位。当计数值达到0x40，即使中断没有被使能(WWDG_CFR中的EWI位被清除)该位也会被硬件置1。这个bit可以通过写0清零，写1无效。

## 12. 模拟数字转换器 (ADC)

### 12.1. 简介

12 位 ADC 是一种采用逐次逼近方式的模拟数字转换器。它有 19 个多路复用通道，可以测量来自 16 个外部和 3 个内部信号源的信号。模拟看门狗允许应用程序来检测输入电压是否超出用户设定的高低阈值。各种通道的 A/D 转换可以配置成单次、连续、扫描或间断转换模式。ADC 转换的结果可以按照左对齐或右对齐的方式存储在 16 位数据寄存器中。

### 12.2. 主要特性

#### GD32F130xx和GD32F150xx产品:

- 12位分辨率
- ADC转换时间: 1.0us(1MS/s)
- 双时钟域架构(APB时钟和ADC时钟)
- 自校准时间: 83个ADC时钟周期
- 可编程采样时间
- 数据寄存器可配置数据对齐方式
- 支持规则数据转换的DMA请求
- 模拟输入通道: 16个外部模拟输入通道, 1个内部温度传感通道( $V_{SENSE}$ ), 1个内部参考电压输入通道( $V_{REFINT}$ ), 1个外部监测电池 $V_{BAT}$ 供电引脚输入通道
- 转换模式: 单次模式、扫描模式、连续模式和间断模式
- 模拟看门狗
- ADC供电要求: 2.6V到3.6V
- ADC输入范围:  $V_{SSA} \leq V_{IN} \leq V_{DDA}$
- 规则转换和注入组转换完成, 模拟看门狗都可以产生中断
- 规则通道和注入通道外部触发

#### GD32F170xx和GD32F190xx产品:

- 高性能
  - 可配置12位、10位、8位、或者6位分辨率
  - ADC转换时间: 12位分辨率为0.5us (2MS/s), 10位分辨率为0.43us, 6位分辨率为0.286us。分辨率越低, 转换越快

- 自校准时间：84个ADC时钟周期
- 可编程采样时间
- 数据寄存器可配置数据对齐方式
- 支持规则数据转换的DMA请求
- 双时钟域架构(APB时钟和ADC时钟)
- 模拟输入通道
  - 16个外部模拟输入通道
  - 1个内部温度传感通道( $V_{SENSE}$ )
  - 1个内部参考电压输入通道( $V_{REFINT}$ )
  - 1个外部监测电池 $V_{BAT}$ 供电引脚输入通道
- 转换开始的发起
  - 软件
  - 配置极性进行硬件触发(来自TIMER1、TIMER2、TIMER3和TIMER15内部定时器事件)
  - 规则通道和注入通道外部触发
- 转换模式
  - 转换单个通道，或者扫描多个通道
  - 单次模式，每次触发转换一次选择的输入
  - 连续模式，连续转换所选择的输入
  - 间断模式
- 规则转换和注入组转换完成，模拟看门狗都可以产生中断
- 模拟看门狗
- ADC供电要求：3V到5.5V，一般电源电压为5V
- 过采样
  - 16位的数据寄存器
  - 可调整的过采样率，从2x到256x
  - 高达8位的可编程数据移位
- ADC输入范围： $V_{SSA} \leq V_{IN} \leq V_{DDA}$

## 12.3. 功能描述

图 12-1 和图 12-2 分别给出了对应设备的 ADC 模块图。

**图 12-1. GD32F130xx 与 GD32F150xx 产品的 ADC 模块图**

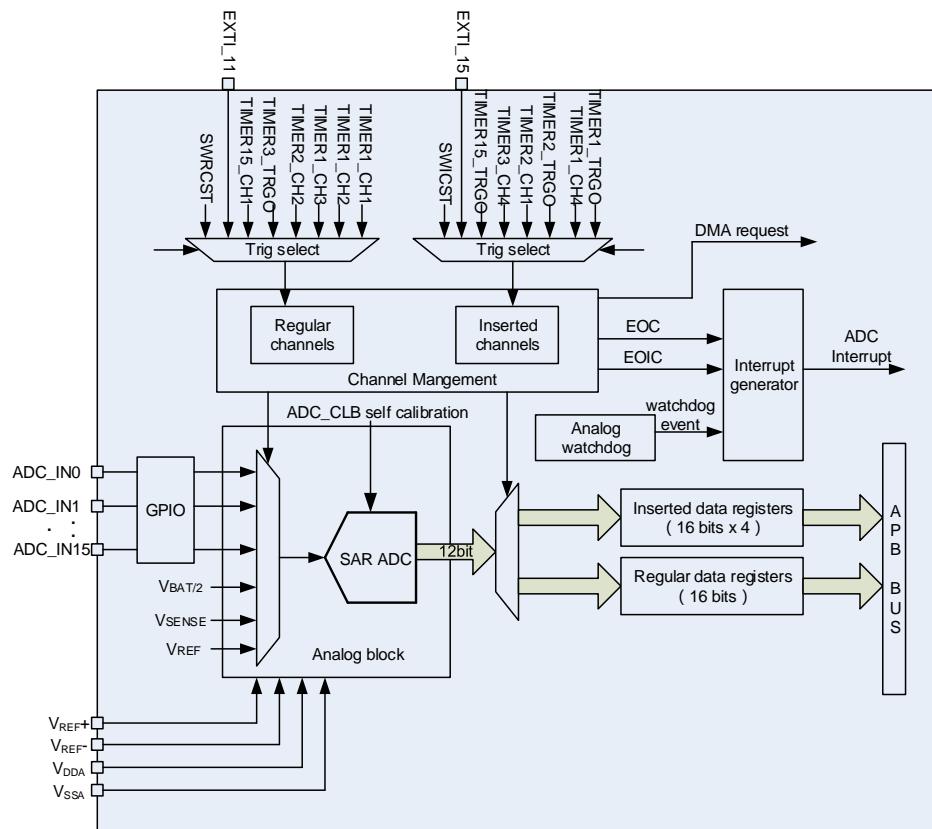


图 12-2. GD32F170xx 与 GD32F190xx 产品的 ADC 模块

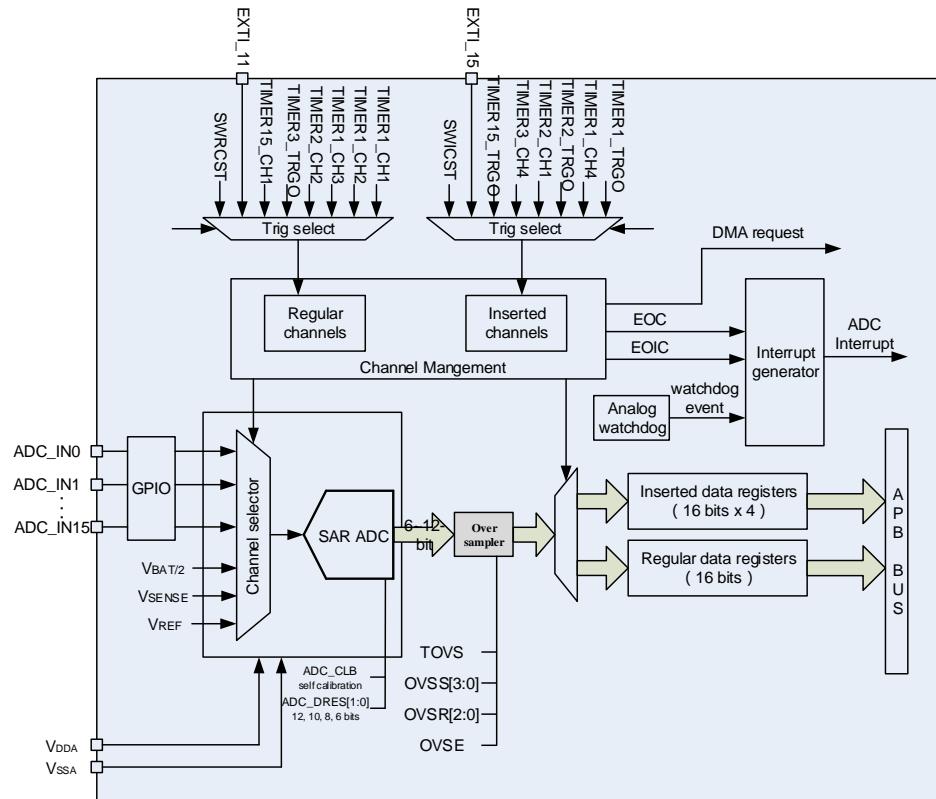


表 12-2 和表 12-3 分别给出了相应设备的 ADC 引脚和内部信号说明。

表 12-1. ADC 内部信号

内部信号名称	信号类型	说明
V <sub>SENSE</sub>	输入	内部温度传感器输出电压
V <sub>REFINT</sub>	输入	内部参考输出电压
V <sub>BAT</sub> /2	输入	V <sub>BAT</sub> 引脚输入电压除以 2

表 12-2. GD32F130xx 和 GD32F150xx 产品的 ADC 引脚定义

名称	信号类型	注释
V <sub>DDA</sub> <sup>(1)</sup>	输入, 模拟供电电源	模拟电源输入等于 V <sub>DD</sub> , 2.6V ≤ V <sub>DDA</sub> ≤ 3.6V
V <sub>SSA</sub> <sup>(1)</sup>	输入, 模拟电源地	模拟地, 等于 V <sub>ss</sub>
ADCx_IN[15: 0]	模拟信号	16 路模拟输入通道

表 12-3. GD32F170xx 和 GD32F190xx 产品的 ADC 引脚定义

名称	信号类型	注释
V <sub>DDA</sub> <sup>(1)</sup>	输入, 模拟供电电源	模拟电源输入等于 V <sub>DD</sub> , 3V ≤ V <sub>DDA</sub> ≤ 5.5V

V <sub>SSA</sub> <sup>(1)</sup>	输入, 模拟电源地	模拟地, 等于 V <sub>ss</sub>
ADCx_IN[15: 0]	模拟信号	16 路模拟输入通道

1. V<sub>DDA</sub> 和 V<sub>SSA</sub> 必须分别连接到 V<sub>DD</sub> 和 V<sub>ss</sub>。

### 12.3.1. 校准(ADC\_CLB)

ADC带有校准功能。在ADC校准期间和校准未完成前，应用不能使用ADC模块。在A/D转换前应执行校准操作，对于GD32F130xx和GD32F150xx产品，清除比较器的偏移误差和电容不匹配误差的校准过程需要83个时钟周期，而对于GD32F170xx和GD32F190xx产品，需要84个时钟周期。由于生产过程存在中的差异，每一片芯片的这些误差都各不相同。

校准是由软件设置ADC\_CTLR2中CLB=1来实现初始化。在校准期间，CLB位必须保持为‘1’，当校准完成后，该位被硬件清零。

当ADC运行条件改变(V<sub>DDA</sub>改变是ADC偏移变化的主要因素，其次是温度的改变)，建议重新做一次校准操作。

内部的模拟校准通过设置ADC\_CTLR2寄存器的RSTCLB位来重置。

软件校准过程：

1. 确保ADCON=1
2. 设置RSTCLB (可选的)
3. 设置CLB=1
4. 等待直到CLB=0

### 12.3.2. 双时钟域架构

除了APB接口时钟，ADC的子模块时钟靠ADC时钟提供，该ADC时钟和APB时钟异步并独立于APB时钟。

用户能够在低功耗运行时降低PLCK时钟频率，但是ADC仍能保持最佳运行状态。

想要更多ADC时钟产生的信息可以参考RCC4.2.1部分。

### 12.3.3. 规则组和注入组通道

ADC支持19个多路通道，可以把转换组织成两组：一个规则组通道和一个注入组通道。

规则组由多达16个转换组成，转换顺序由寄存器控制。ADC\_RSQ1~ADC\_RSQ3寄存器规定了规则组的通道选择。ADC\_RSQ1寄存器的RL[3:0]位规定了全部的转换序列长度。

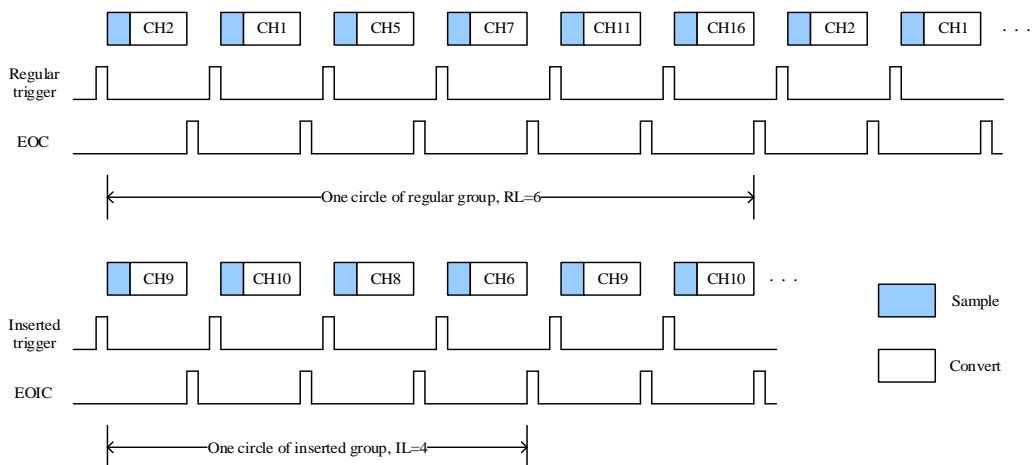
注入组由多达4个转换组成，转换顺序由寄存器控制，ADC\_ISQ寄存器规定了注入组的通道选择。ADC\_ISQ寄存器的IL[1:0]位规定了全部的转换序列长度。

### 12.3.4. 转换模式

#### 单次转换模式

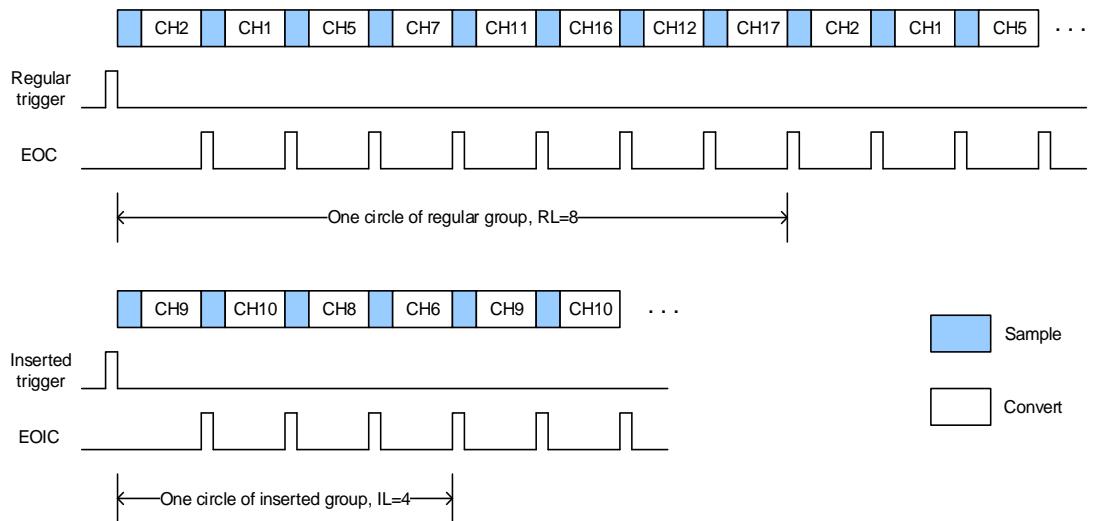
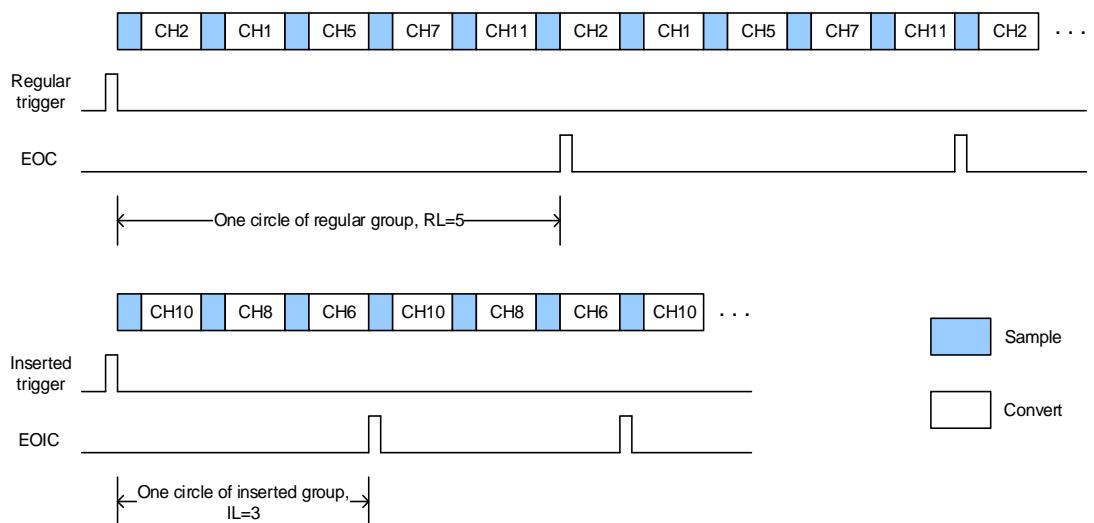
单次转换模式下，ADC\_RSQ1~ADC\_RSQ3寄存器组或者ADC\_ISQ寄存器规定了通道的转换顺序。一旦ADCON位被置‘1’，当相应软件触发或者外部触发发生，单通道的规则组或注入组完成一次采样和转换。单次转换结束后，转换数据将被存放于ADC\_RDTR或ADC\_IDTRx寄存器中，EOC或者EOIC位将被置‘1’。如果EOCIE或EOICIE位被置‘1’，将产生一次中断。

图 12-3. 单次转换模式



#### 连续转换模式

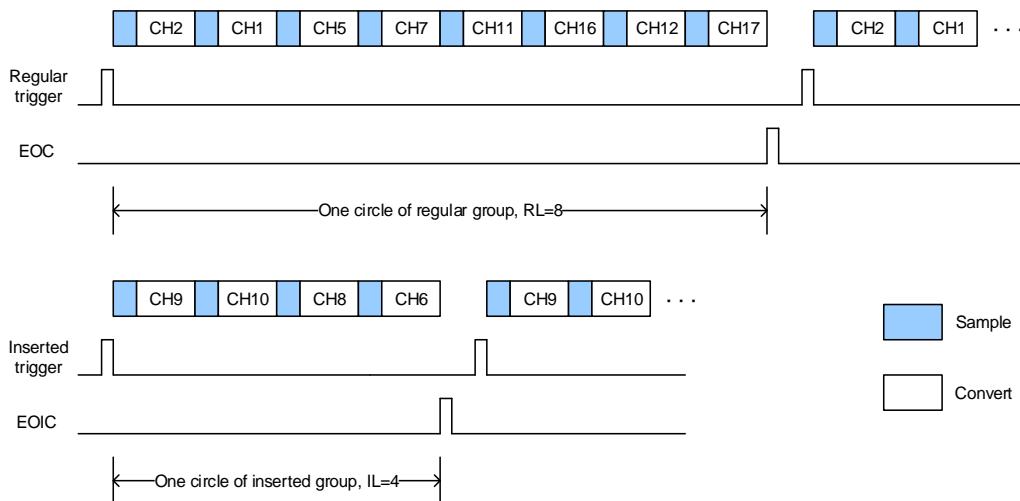
对ADC\_CTLR2寄存器的CTN位置‘1’可以使能连续转换模式。在此模式下，ADC执行由ADC\_RSQ1~ADC\_RSQ3寄存器组或者ADC\_ISQ寄存器规定的转换通道序列。一旦ADCON位被置‘1’，当相应软件触发或者外部触发产生时，规则组或者注入组的AD采样和转换按照指定的通道一个接一个完成。转换数据保存在ADC\_RDTR或ADC\_IDTRx寄存器中。如果扫描模式被禁止，在每个通道转换结束之后EOC或EOIC位将被置‘1’。如果使能扫描模式，在每个规则组或者注入组每次转换周期结束后EOC或EOIC位将被置‘1’。如果EOCIE或EOICIE位被置‘1’，将产生中断。

**图 12-4. 连续转换模式, 禁用扫描**

**图 12-5. 连续转换模式, 使用扫描**


## 扫描转换模式

扫描转换模式可以通过将ADC\_CTLR1寄存器的SM位置‘1’来使能。在此模式下，ADC扫描转换所有被ADC\_RSQ1~ADC\_RSQ3寄存器或ADC\_ISQ寄存器选中的所有通道。一旦ADCON位被置‘1’，当相应软件触发或者外部触发产生，规则组或注入组通道的ADC采样和转换一个接一个地完成，直到规则组或注入组转换结束。转换数据存储在ADC\_RDTR或ADC\_IDTRx寄存器中。规则组或注入组转换结束后，EOC或者EOIC位将被置‘1’。如果EOCIE或EOICIE位被置‘1’，将产生中断。ADC\_CTLR2寄存器的DMA位在扫描模式下必须设置为‘1’。

图 12-6. 扫描转换模式



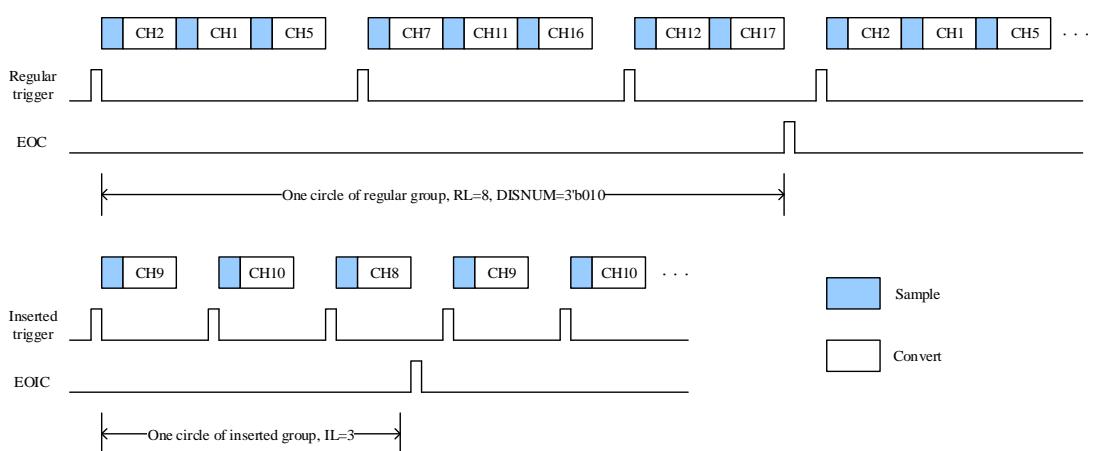
### 间断模式

对于规则组来说，ADC\_CTLR1寄存器的DISRC位置‘1’使能间断转换模式。该模式下可以执行一个短序列的n次转换( $n \leq 8$ )，此转换是ADC\_RSQ1~RSQ3寄存器所选择的转换序列的一部分。数值n由ADC\_CTLR1寄存器的DISCNUM[2:0]位给出。一个软件触发或者外部触发信号可以启动ADC\_RSQ1~ADC\_RSQ3寄存器中选择的下一轮n次采样和转换，直到此序列所有的转换完成为止。每个规则组转换周期结束后，EOC位将被置‘1’。EOCIE位被置‘1’将产生一个中断。

对于注入组来说，ADC\_CTLR1寄存器的DISIC位置‘1’使能间断转换模式。该模式下可以执行ADC\_ISQ寄存器所选择的转换序列的一个通道进行转换。一个软件触发或者外部触发信号可以启动ADC\_ISQ寄存器中选择的下一个通道的转换，直到序列中所有的转换完成为止。每个注入组转换周期结束后，EOIC位将被置‘1’。EOCIE位被置‘1’将产生一个中断。

必须避免同时为规则组和注入组设置间断模式，间断模式只能作用于一组转换。

图 12-7. 间断转换模式



### 12.3.5. 模拟看门狗

ADC\_CTLR1寄存器的AWREN和AWIEN位置‘1’将分别使能规则组和注入组的模拟看门狗的

功能。如果ADC的模拟转换电压低于低阈值或高于高阈值时，ADC\_STR状态寄存器的AWE位将被置‘1’。如果AWEIE位被置‘1’，将产生中断。ADC\_AWHT和ADC\_AWLT寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与ADC\_CTLR2寄存器的DAL位确定的对齐方式无关。ADC\_CTLR1寄存器的AWREN，AWIEN，AWSSM和AWCS[4:0]位可以用来选择模拟看门狗监控单一通道或者多通道。

### 12.3.6. 注入通道管理

#### 自动注入

如果将ADC\_CTLR1寄存器的ICA位置‘1’，在规则组通道之后，注入组通道被自动转换。该模式下注入组通道的外部触发不能被使能。该模式可以转换ADC\_RSQ1~ADC\_RSQ3和ADC\_ISQ寄存器中设置的多至20个转换序列。除了ICA位之外，如果CTN位也被置‘1’，注入组通道将在规则组通道之后被自动转换。

不能同时使用自动注入和间断模式。

#### 触发注入

清除ICA位并将SM位置‘1’，在规则组通道转换期间如果软件触发或者外部触发发生，则启动触发注入转换。这种情况下，ADC取消当前转换，注入通道序列被以扫描模式进行转换。注入通道组转换结束后，规则组转换从上次被取消的转换处重新开始。

### 12.3.7. 数据对齐

ADC\_CTLR2寄存器的DAL位确定转换后数据存储的对齐方式。

注入组通道转换的数据值已经减去了在ADC\_ICOSx寄存器中定义的偏移量，因此结果可能是一个负值。符号值是一个扩展值。

**图 12-8. 12 位分辨率的数据对齐**

Regular group data

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	-----	-----	----	----	----	----	----	----	----	----	----	----

Inserted group data

Sign	Sign	Sign	Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
------	------	------	------	-----	-----	----	----	----	----	----	----	----	----	----	----

DAL=0

Regular group data

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
-----	-----	----	----	----	----	----	----	----	----	----	----	---	---	---	---

Inserted group data

Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0
------	-----	-----	----	----	----	----	----	----	----	----	----	----	---	---	---

DAL=1

### 12.3.8. 可编程的采样时间

ADC 使用若干个ADC\_CLK周期对输入电压采样，采样周期数目可以通过ADC\_SPT1和ADC\_SPT2寄存器的SPTn[2:0]位更改。每个通道可以用不同的时间采样。在12位分辨率的情况下，总转换时间=采样时间+12.5个ADC\_CLK周期。

### 12.3.9. 外部触发

如果将ADC\_CTLR2寄存器的ETERC或ETEIC位置‘1’，外部触发信号的上升沿可以触发一次规则组或注入组转换。ETSRC[2:0]和ETSIC[2:0]控制位可以用来确定8个可能事件中的哪一个，可以触发规则和注入组的转换。

**表 12-4. ADC 用于规则通道的外部触发**

ETSRC[2:0]	触发源	触发类型
000	TIMER1_CH1	来自片上定时器的内部信号
001	TIMER1_CH2	
010	TIMER1_CH3	
011	TIMER2_CH2	
100	TIMER3_TRGO	
101	TIMER15_CH1	
110	EXTI_11	外部信号
111	SWRCST	软件控制位

**表 12-5. ADC 用于注入通道的外部触发**

ETSIC[2:0]	触发源	触发类型
000	TIMER1_TRGO	来自片上定时器的内部信号
001	TIMER1_CH4	
010	TIMER2_TRGO	
011	TIMER2_CH1	
100	TIMER3_CH4	
101	TIM15_TRGO	
110	EXTI_15	外部信号
111	SWICST	软件控制位

### 12.3.10. DMA 请求

因为规则通道转换的值储存在一个仅有的数据寄存器中，所以当转换多个规则通道时需要使用DMA。ADC在规则组一个通道转换结束后产生一个DMA请求，DMA接受到请求后可以将转换的数据从ADC\_RDTR寄存器传输到用户指定的目的地址。

### 12.3.11. 温度传感器和内部参考电压 V<sub>REF</sub>

将ADC\_CTLR2寄存器的TSVREN位置‘1’可以使能温度传感器通道(ADC1\_IN16)和V<sub>REF</sub>通道(ADC1\_IN17)。温度传感器可以用来测量器件周围的温度。传感器输出电压能被ADC转换成数

字量。建议设置温度传感器的采样时间为 $17.1\mu s$ 。温度传感器不用时，复位TSVREN位可以将其置于掉电模式。

温度传感器的输出电压随温度线性变化，由于生产过程的多样化，温度变化曲线的偏移在不同的芯片上会有不同(最多相差 $45^{\circ}C$ )。内部温度传感器更适合于检测温度的变化，而不是测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

内部电压参考( $V_{REF}$ )提供了一个稳定的(带隙基准)电压输出给ADC和比较器。 $V_{REF}$ 内部连接到ADC\_IN17输入通道。

使用温度传感器：

1. 配置温度传感器通道(ADC1\_IN16)的转换序列和采样时间 $17.1\mu s$ 。
2. 置位ADC\_CTLR2寄存器的TSVREN位使能温度传感器。
3. 置位ADC\_CTLR2寄存器的ADCON位或者由外部触发启动ADC转换。
4. 读取温度传感器数据 $V_{temperature}$ 并由下面公式计算出实际温度：

$$\text{温度}({}^{\circ}\text{C}) = \{(V_{30} - V_{temperature(digit)}) / \text{Avg\_Slope}\} + 30$$

$V_{30}$ : 温度传感器在 $30^{\circ}\text{C}$ 下的电压，典型值 $1.43\text{V}$ 。

Avg\_Slope: 温度与温度传感器电压曲线的均值斜率，典型值 $4.3\text{mV}/{}^{\circ}\text{C}$ 。

### 12.3.12. 电池电压监测

$V_{BAT}$ 通道由于监测从 $V_{BAT}$ 引脚过来的备份电池电压。当ADC\_CTLR2寄存器中的VBATEN位置1时，使能 $V_{BAT}$ 通道(ADC\_IN18)，同时一个集成在 $V_{BAT}$ 引脚上的2分压桥也随之自动被使能。由于 $V_{BAT}$ 可能比 $V_{DDA}$ 高，所以使用这个2分压桥用来确保ADC正确操作。因此，转换的值是 $V_{BAT}$ 数据的一半。为了防止不必要的电池能量消耗，推荐仅在需要时才使能2分压桥。

### 12.3.13. ADC 中断

规则组和注入组转换结束和模拟看门狗状态位被置‘1’的时候都能引发中断。独立的中断使能位用于灵活设置ADC中断。

### 12.3.14. 可编程分辨率 (RES) —— 快速转换模式

本节内容适用于GD32F170xx和GD32F190xx产品。

通过降低ADC的分辨率，可能获得较快的转换时间( $t_{ADC}$ )。

对寄存器ADC\_CTLR1中的DRES[1:0]位进行编程即可配置分辨率为6、8、10、12位。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。

只有在ADCON比特为0时，才能修改DRES[1:0]的值。

ADC转换的结果只有12位，其余没有被用到的低位读出来都是为0。

较低的分辨率能够减少转换时间。

**表 12-6. 不同的分辨率对应的 tSAR 时间参数**

DRES[1:0] bits	t <sub>SAR</sub> (ADC clock cycles)	t <sub>SAR(ns)</sub> at f <sub>ADC</sub> =28MHz	t <sub>SMPL</sub> (ADC clock cycles)	t <sub>ADC</sub> (ADC clock cycles)	t <sub>ADC(ns)</sub> at f <sub>ADC</sub> =28MHz
12	12.5	446ns	1.5	14	500ns
10	10.5	375ns	1.5	12	429ns
8	8.5	304ns	1.5	10	357ns
6	6.5	232ns	1.5	8	286ns

### 12.3.15. 过采样（仅适用于 GD32F170xx 和 GD32F190xx 产品）

过采样单元可以通过设置ADC\_OVSCR寄存器的OVSE位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。

过采样单元执行数据预处理以减轻CPU负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个16位的数据。

其结果值根据如下公式计算得出，其中N和M的值可以被调整：

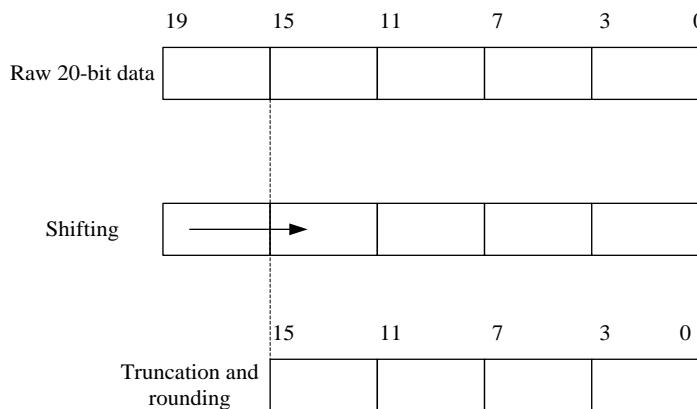
$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{n=N-1} \text{Conversion}(t_n)$$

平均值、数据降低率、信噪比改善和基本过滤这些功能都是由硬件执行的。

过采样率N是在ADC\_OVSCR寄存器的OVSR[2:0]位定义，它的取值范围为2x到256x。除法系数M由一个最多右移8位的值组成，它通过ADC\_OVSCR寄存器OVSS[3:0]位进行配置。

求和单元能够生成一个多达20位(256\*12位)的值。将这个值首先要进行移位，通过四舍五入为一个近似值，然后高位会被截断，仅保留最低16位有效位作为最终值传入对应的数据寄存器中。

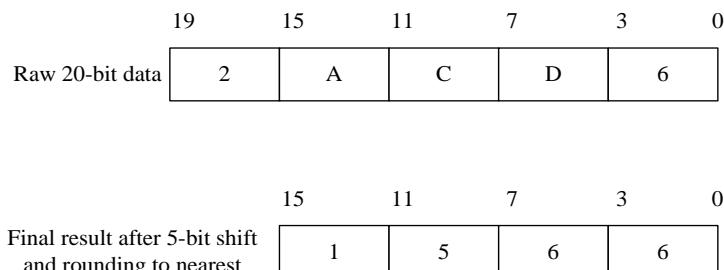
**图 12-9. 20 位到 16 位的结果截取**



**注意：**如果移位后的中间结果还是超过16位，那么该结果的高位就会被直接截掉。

图 12-10 描述一个从原始 20 位的累积数值处理成 16 位结果值的例子。

**图 12-10. 右移 5 位和四舍五入的数例**



下表给出了N和M各种组合的数据格式，初始转换值为0xFFFF。

**表 12-7 N 和 M 的最大输出值**

Over sa mplin g ratio	Max Raw data	No- shift OVSS	1-bit shift OVSS	2-bit shift OVSS	3-bit shift OVSS	4-bit shift OVSS	5-bit shift OVSS	6-bit shift OVSS	7-bit shift OVSS	8-bit shift OVSS
		= <b>0000</b>	= <b>0001</b>	= <b>0010</b>	= <b>0011</b>	= <b>0100</b>	= <b>0101</b>	= <b>0110</b>	= <b>0111</b>	= <b>1000</b>
2x	0x1FF E	0x1F FE	0x0F FF	0x080 0	0x040 0	0x020 0	0x010 0	0x008 0	0x004 0	0x002 0
4x	0x3FF C	0x3F FC	0x1F FE	0x0F FF	0x080 0	0x040 0	0x020 0	0x010 0	0x008 0	0x004 0
8x	0x7FF 8	0x7F F8	0x3F FC	0x1F FE	0x0F FF	0x080 0	0x040 0	0x020 0	0x010 0	0x008 0
16x	0xFFFF 0	0xFF F0	0x7F F8	0x3F FC	0x1F FE	0x0F FF	0x080 0	0x040 0	0x020 0	0x010 0
32x	0x1FF E0	<b>0xFF</b> <b>E0</b>	0xFF F0	0x7F F8	0x3F FC	0x1F FE	0x0F FF	0x080 0	0x040 0	0x020 0
64x	0x3FF C0	<b>0xFF</b> <b>C0</b>	<b>0xFF</b> <b>E0</b>	0xFF F0	0x7F F8	0x3F FC	0x1F FE	0x0F FF	0x080 0	0x040 0
128x	0x7FF 80	<b>0xFF</b> <b>80</b>	<b>0xFF</b> <b>C0</b>	<b>0xFF</b> <b>E0</b>	0xFF F0	0x7F F8	0x3F FC	0x1F FE	0x0F FF	0x080 0
256x	0xFFFF 00	<b>0xFF</b> <b>00</b>	<b>0xFF</b> <b>80</b>	<b>0xFF</b> <b>C0</b>	<b>0xFF</b> <b>E0</b>	0xFF F0	0x7F F8	0x3F FC	0x1F FE	0x0F FF

和标准的转换模式相比，过采样模式的转换时间不会改变，在整个过采样序列的过程中采用时间仍然保持相等。每N个转换就会产生一个新的数据，一个等价的延迟为 $N \times t_{ADC} = N \times (t_{SMPL} + t_{SAR})$ 。

### 过采样配合 ADC 转换模式

当过采样使能时，大多数 ADC 工作模式也都是可用的。

- 规则通道或注入通道
- 由软件或外部触发来开始 ADC 转换
- 单通道或扫描模式，连续或间断模式
- 可编程的采样时间
- 模拟看门狗

只有当 ADCON=0 时，才可以改变过采样的配置。并且要保证在设置 ADCON=1 之前要对过采样进行配置。

## 12.4. ADC 寄存器

### 12.4.1. ADC 状态寄存器 (ADC\_STR)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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										STRC	STIC	EOIC	EOC	AWE	
rc_w0				rc_w0				rc_w0				rc_w0			

位/位域	名称	描述
31:5	保留	必须保持复位值
4	STRC	规则组转换开始标志 0: 规则组转换没有开始 1: 规则组转换开始 规则组转换开始时硬件置位。 软件写'0'清除。
3	STIC	注入组转换开始标志 0: 注入组转换没有开始 1: 注入组转换开始 注入组转换开始时硬件置位。 软件写'0'清除。
2	EOIC	注入组转换结束标志 0: 注入组转换没有结束 1: 注入组转换结束 注入组转换结束时硬件置位。

软件写‘0’清除。

1	EOC	组转换结束标志 0: 组转换没有结束 1: 组转换结束 注入组或规则组转换结束时硬件置位。 软件写‘0’或读ADC_RDTR寄存器清除。
0	AWE	模拟看门狗事件标志 0: 没有模拟看门狗事件 1: 产生模拟看门狗事件 转换电压超过ADC_AWLT和ADC_AWHT寄存器设定的阈值时由硬件置‘1’，软件写‘0’清除。

### 12.4.2. ADC 控制寄存器 1 (ADC\_CTLR1)

**GD32F130xx 和 GD32F150xx 产品：**

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
保留								AWREN	AWIEN	保留							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DISNUM[2:0]	DISIC	DISRC	ICA	AWSSM	SM	EOICIE	AWEIE	EOCIE	AWCS[4:0]								
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

位/位域	名称	描述
31:24	保留	必须保持复位值
23	AWREN	规则组看门狗使能 0: 规则组看门狗禁止 1: 规则组看门狗使能
22	AWIEN	注入组看门狗使能 0: 注入组看门狗禁止 1: 注入组看门狗使能
21:16	保留	必须保持复位值
15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成DISNUM[2:0]+1
12	DISIC	注入组间断模式 0: 注入组间断模式禁止

		1: 注入组间断模式使能
11	DISRC	规则组间断模式 0: 规则组间断模式禁止 1: 规则组间断模式使能
10	ICA	注入组自动转换 0: 注入组自动转换禁止 1: 注入组自动转换使能
9	AWSSM	扫描模式下, 模拟看门狗在单通道有效 0: 模拟看门狗在所有通道有效 1: 模拟看门狗在单通道有效
8	SM	扫描模式 0: 扫描模式使能 1: 扫描模式禁止
7	EOICIE	EOIC中断使能 0: EOIC中断禁止 1: EOIC中断使能
6	AWEIE	AWE中断使能 0: AWE中断禁止 1: AWE中断使能
5	EOCIE	EOC中断使能 0: EOC中断禁止 1: EOC中断使能
4:0	AWCS[4:0]	模拟看门狗通道选择 00000: ADC通道0 00001: ADC通道1 00010: ADC通道2 ..... 01111: ADC通道15 1xx00: ADC通道16 1xx01: ADC通道17 1xx1x: ADC通道18 “x”代表0或1

#### GD32F170xx 和 GD32F190xx 产品:

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

保留								DRES[1:0]	AWREN	AWIEN	保留							
rw								rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
DISNUM[2:0]	DISIC	DISRC	ICA	AWSSM	SM	EOICIE	AWEIE	EOCIE		AWCS[4:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:26	保留	必须保持复位值
25:24	DRES[1:0]	ADC分辨率 00: 12位 01: 10位 10: 8位 11: 6位
23	AWREN	规则组看门狗使能 0: 规则组看门狗禁止 1: 规则组看门狗使能
22	AWIEN	注入组看门狗使能 0: 注入组看门狗禁止 1: 注入组看门狗使能
21:16	保留	必须保持复位值
15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成DISNUM[2:0]+1
12	DISIC	注入组间断模式 0: 注入组间断模式禁止 1: 注入组间断模式使能
11	DISRC	规则组间断模式 0: 规则组间断模式禁止 1: 规则组间断模式使能
10	ICA	注入组自动转换 0: 注入组自动转换禁止 1: 注入组自动转换使能
9	AWSSM	扫描模式下, 模拟看门狗在单通道有效 0: 模拟看门狗在所有通道有效 1: 模拟看门狗在单通道有效
8	SM	扫描模式 0: 扫描模式使能 1: 扫描模式禁止
7	EOICIE	EOIC中断使能

		0: EOIC中断禁止 1: EOIC中断使能
6	AWEIE	AWE中断使能 0: AWE中断禁止 1: AWE中断使能
5	EOCIE	EOC中断使能 0: EOC中断禁止 1: EOC中断使能
4:0	AWCS[4:0]	模拟看门狗通道选择 00000: ADC通道0 00001: ADC通道1 00010: ADC通道2 ..... 01111: ADC通道15 1xx00: ADC通道16 1xx01: ADC通道17 1xx1x: ADC通道18 “x”代表0或1

### 12.4.3. ADC 控制寄存器 2 (ADC\_CTLR2)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			VBATEN	TSVREN	SWRCST	SWICST	ETERC	ETSRC[2:0]			保留				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETEIC	ETSIC[2:0]	DAL	保留	DMA	保留				RSTCLB	CLB	CTN	ADCON			
rw	rw	rw		rw					rw	rw	rw	rw			

位/位域	名称	描述
31:25	保留	必须保持复位值
24	VBATEN	软件使能或禁止V <sub>BAT</sub> 通道 0: V <sub>BAT</sub> 通道禁止 1: V <sub>BAT</sub> 通道使能
23	TSVREN	ADC1的通道16和17使能 0: ADC1的通道16和17禁止 1: ADC1的通道16和17使能

22	SWRCST	规则组转换开始 如果ETSRC[2:0]是111，该位置'1'开启规则组转换。转换开始后，软件置位，软硬件均可清零。
21	SWICST	注入组转换开始 如果ETSIC[2:0]是111，该位置'1'开启注入组转换。转换开始后，软件置位，软硬件均可清零。
20	ETERC	规则组外部触发使能 0: 规则组外部触发禁止 1: 规则组外部触发使能
19:17	ETSRC[2:0]	规则组外部触发选择 000: 定时器1 CC1 001: 定时器1 CC2 010: 定时器1 CC3 011: 定时器2 CC2 100: 定时器3 TRGO 101: 定时器15 CC1 110: 中断线 11 111: SWRCST
16	保留	必须保持复位值
15	ETEIC	注入组外部触发使能 0: 注入组外部触发禁止 1: 注入组外部触发使能
14:12	ETSIC[2:0]	注入组外部触发选择 000: 定时器1 TRGO 001: 定时器1 CC4 010: 定时器2 TRGO 011: 定时器2 CC1 100: 定时器3 CC4 101: 定时器15 TRGO 110: 中断线15 111: SWICST
11	DAL	数据对齐 0: 最低有效位对齐 1: 最高有效位对齐
10:9	保留	必须保持复位值
8	DMA	DMA请求使能 0: DMA请求禁止 1: DMA请求使能

---

7:4	保留	必须保持复位值
3	RSTCLB	校准复位 在校准寄存器初始化后该位可以软件置位和硬件清零。 0: 校准寄存器初始化结束. 1: 校准寄存器初始化开始
2	CLB	ADC校准 0: 校准结束 1: 校准开始
1	CTN	连续模式 0: 禁止连续模式 1: 使能连续模式
0	ADCON	开启ADC。该位从'0'变成'1'将唤醒ADC。当该位被置位以后，不改变寄存器的其他位仅仅对该位写'1',将开启转换。 0: 禁止ADC关闭电源 1: 使能ADC

#### 12.4.4. ADC 采样时间寄存器 1 (ADC\_SPT1)

GD32F130xx 和 GD32F150xx 产品:

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SPT17[2:0]		SPT16[2:0]		SPT15[2:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	rw			SPT14[2:0]	rw			SPT13[2:0]	rw			SPT12[2:0]	rw		

位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	SPTn[2:0]	通道n的采样时间 000: 1.5个周期 001: 7.5个周期 010: 13.5个周期 011: 28.5个周期 100: 41.5个周期 101: 55.5个周期 110: 71.5个周期

111: 239.5 周期

注意: GD32F130xx和GD32F150xx产品, 通道17和通道18的采样时间都是通过ADC\_SPT1的SPT17[2:0]设置。

### GD32F170xx 和 GD32F190xx 产品:

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]		
rw	rw			rw			rw			rw			rw		

位/位域	名称	描述
31:27	保留	必须保持复位值
26:0	SPTn[2:0]	通道n的采样时间 000: 1.5个周期 001: 7.5个周期 010: 13.5个周期 011: 28.5个周期 100: 41.5个周期 101: 55.5个周期 110: 71.5个周期 111: 239.5个周期

### 12.4.5. ADC 采样时间寄存器 2 (ADC\_SPT2)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]	SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]		
rw	rw														

位/位域	名称	描述
------	----	----

31:30	保留	必须保持复位值
29:0	SPTn[2:0]	通道n的采样时间 000: 1.5个周期 001: 7.5个周期 010: 13.5个周期 011: 28.5个周期 100: 41.5个周期 101: 55.5个周期 110: 71.5个周期 111: 239.5个周期

#### 12.4.6. ADC 注入通道数据偏移寄存器 x (ADC\_ICOSx) (x=1..4)

地址偏移: 0x14 - 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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				ICOSn[11:0]											

rw

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	ICOSn[11:0]	注入通道n的数据偏移 当转换注入通道时, 这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在ADC_IDTRx寄存器中读出。

#### 12.4.7. ADC 看门狗高阈值寄存器 (ADC\_AWHT)

地址偏移: 0x24

复位值: 0x0000 0FFF

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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				AWHT[11:0]											

rw

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	AWHT[11:0]	模拟看门狗高阈值 这些位定义了模拟看门狗的高阈值。

#### 12.4.8. ADC 看门狗低阈值寄存器 (ADC\_AWLT)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



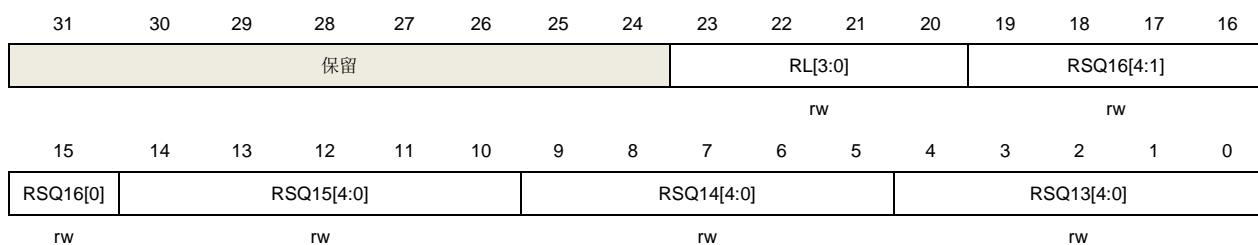
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	AWLT[11:0]	模拟看门狗低阈值 这些位定义了模拟看门狗的低阈值.

#### 12.4.9. ADC 规则序列寄存器 1 (ADC\_RSQ1)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	RL[3:0]	规则通道序列长度 规则通道转换序列中的总的通道数目为RL[3:0]+1。
19:0	RSQn[4:0]	通道编号(0..18)写入这些位来选择规则通道的第n个转换的通道

### 12.4.10. ADC 规则序列寄存器 2 (ADC\_RSQ2)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ12[4:0]				RSQ11[4:0]				RSQ10[4:1]					
rw															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ10[0]		RSQ9[4:0]				RSQ8[4:0]				RSQ7[4:0]					
rw															rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29:0	RSQn[4:0]	通道编号(0..18) 写入这些位来选择规则通道的第n个转换的通道。

### 12.4.11. ADC 规则序列寄存器 3 (ADC\_RSQ3)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ6[4:0]				RSQ5[4:0]				RSQ4[4:1]					
rw															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ4[0]		RSQ3[4:0]				RSQ2[4:0]				RSQ1[4:0]					
rw															rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29:0	RSQn[4:0]	通道编号(0..18) 写入这些位来选择规则通道的第n个转换的通道

### 12.4.12. ADC 注入序列寄存器 (ADC\_ISQ)

地址偏移: 0x38

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						IL[1:0]		ISQ4[4:1]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ISQ4[0]	ISQ3[4:0]				ISQ2[4:0]				ISQ1[4:0]								
rw				rw				rw				rw					

位/位域	名称	描述
31:22	保留	必须保持复位值
21:20	IL[1:0]	注入通道序列长度 注入通道转换序列中的总的通道数目为IL[1:0]+1
19:0	ISQn[4:0]	通道编号(0..18) 写入这些位来选择注入通道的第n个转换的通道 和规则通道转换序列不同的是，如果IL[1:0] 长度不足4，注入通道转换从(4-IL[1:0])开始。 IL 11 ISQ1>>ISQ2>>ISQ3>>ISQ4 10 ISQ2>>ISQ3>>ISQ4 01 ISQ3>>ISQ4 00 ISQ4

#### 12.4.13. ADC 注入数据寄存器 x (ADC\_IDTRx) (x= 1..4)

地址偏移: 0x3C - 0x48

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	IDTn[15:0]	注入转换的数据n 这些位包含了注入通道的转换结果，只读。

#### 12.4.14. ADC 规则数据寄存器 (ADC\_RDTR)

地址偏移: 0x4C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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



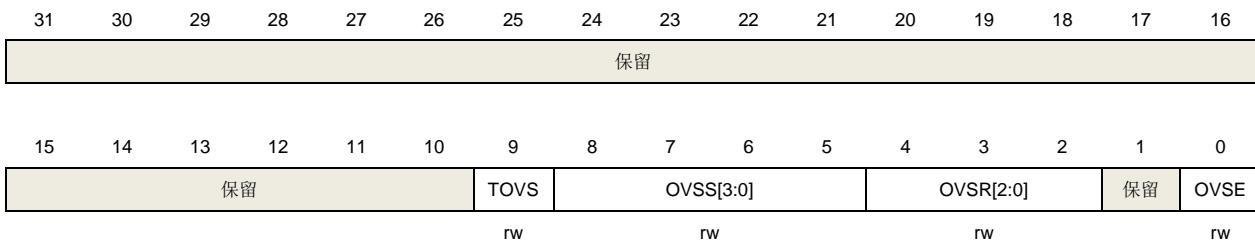
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RDTR[15:0]	规则通道数据 这些位包含了规则通道的转换结果，只读。

#### 12.4.15. ADC 过采样控制寄存器(ADC\_OVSCR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x80

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:10	保留	必须保持复位值
9	TOVS	过采样触发 该位通过软件置位和清除。 0: 在一次触发后连续执行通道的所有过采样转换 1: 通道的每次过采样转换都需要一次触发 <b>注意:</b> 只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。
8:5	OVSS[3:0]	过采样移位 该位通过软件置位和清除。 0000: 不移位 0001: 移1位 0010: 移2位 0011: 移3位 0100: 移4位 0101: 移5位

0110: 移6位

0111: 移7位

1000: 移8位

其余值都保留

**注意:** 只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。

4:2            OVSR[2:0]        过采样率

这些位定义了过采样率的大小。

000: 2x

001: 4x

010: 8x

011: 16x

100: 32x

101: 64x

110: 128x

111: 256x

**注意:** 只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。

1            保留        必须保持复位值

0            OVSE        过采样使能  
该位通过软件置位和清除。

0: 过采样失能

1: 过采样使能

**注意:** 只有在ADCON=0的时候才允许通过软件对该位进行写(确保没有转换正在执行)。

## 13. 数-模转换器 (DAC)

### 13.1. 简介

数字-模拟转换模块(DAC)是 12 位电压输出的数字-模拟转换器。DAC 可以配置成 8 位或 12 位模式，也可以与 DMA 配合使用。DAC 的输入数据可以被设置为左对齐或右对齐。可选择通过输出缓冲器来实现较高的驱动电流。

对于 GD32F130xx 和 GD32F150xx 产品，DAC 模块只有一个 DAC。

对于 GD32F170xx 和 GD32F190xx 产品，DAC 模块有 2 个 DAC，每个 DAC 都有自己的转换器。在双 DAC 模式下，当两个 DAC 组合在一起用于同步更新操作时，转换可以独立或同步进行。

### 13.2. 主要特性

DAC 主要特性如下：

- 12 位分辨率，数据左对齐或者右对齐
- DMA 功能，支持 DMA 欠载出错检测
- 转换同步更新
- 外部触发转换
- 可配置的内部缓存
- 输入参考电压： $V_{DDA}$

对于 GD32F170xx 和 GD32F190xx 产品，还有以下这些特征：

- 两个 DAC 转换器：每个都有一个输出通道
- 双 DAC 通道独立或同时转换

DAC 通道的框图见图 13-1，

表 13-1 给出了引脚的说明。

图 13-1. DAC 通道模块框图

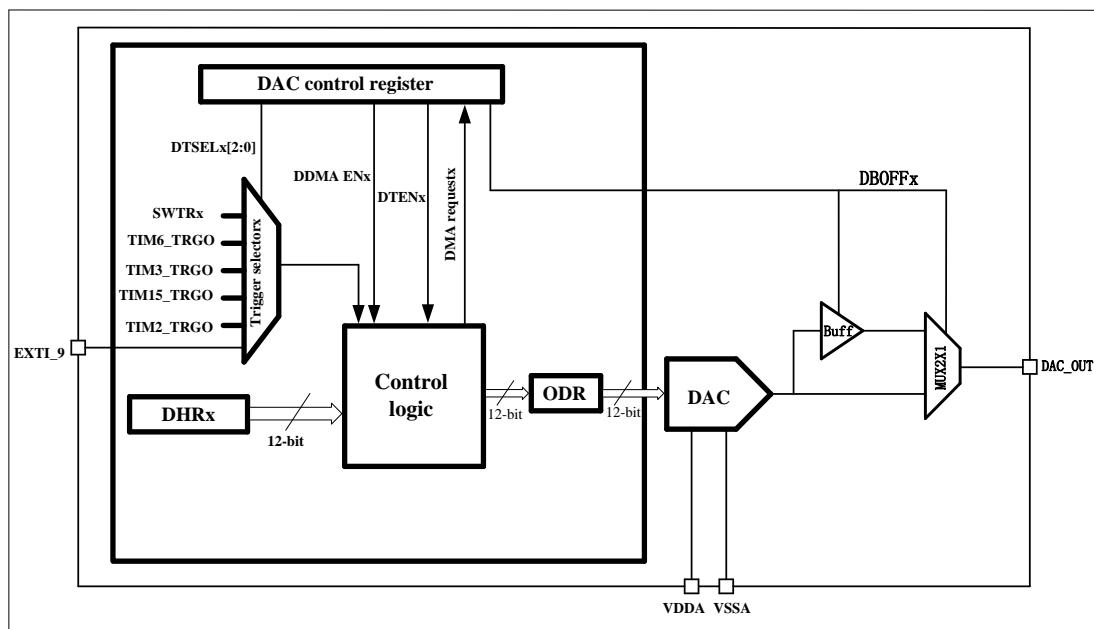


表 13-1. DAC 引脚

名称	注释	信号类型
V <sub>DDA</sub>	模拟电源	输入, 模拟电源
V <sub>SSA</sub>	模拟电源的地线	输入, 模拟电源地
DAC_OUTx	DAC x的模拟输出	模拟输出信号

**注意:** 一旦 DAC 通道使能, 相关联的引脚会自动连接到模拟转换输出引脚(DAC\_OUTx), 引脚会首先被配置为模拟输入 (为了避免寄生干扰和额外的功耗)。

对于 GD32F130xx 和 GD32F150xx 产品, 其引脚为 PA4。

对于 GD32F170xx 和 GD32F190xx 产品, 其引脚为 PA4 或 PA5。

## 13.3. 功能描述

### 13.3.1. 使能 DAC

对 DAC\_CTLR 寄存器的 **DENx** 位置‘1’即可打开对对应的 DAC 供电。经过一段启动时间 **tWAKEUP**, DAC 将被使能。

### 13.3.2. 使能 DAC 输出缓存

DAC 集成了输出缓存 (对于 GD32F130xx 和 GD32F150xx 产品, 只有一个输出缓存; 对于 GD32F170xx 和 GD32F190xx 产品, 有两个缓存), 可以用来减少输出阻抗, 无需外部运放即可直接驱动外部负载。DAC 通道输出缓存可以通过设置 DAC\_CTLR 寄存器的 **DBOFFx** 位来

启用或者关闭。当 DAC 负载重的时候，用户可以使能这个输出缓冲区。

### 13.3.3. DAC 数据格式

单个 DAC，有以下 3 种可能：

1. 8 位数据右对齐：DHRx[11:4]位被写入寄存器 DACx\_R8DHR [7:0]位。
2. 12 位数据右对齐：DHRx[11:0]被写入寄存器 DACx\_R12DHR [11:0]位。
3. 12 位数据左对齐：DHRx[11:0]被写入寄存器 DAC\_L12DHR [15:4]位。

根据 DACx\_yyDHR 寄存器，数据经过移位，然后会被存入 DHRx (DHRx，为数据保持寄存器，它是内部非存储器映射的寄存器)。DHRx 寄存器将会或被自动装载到 DACx\_ODR 寄存器，或通过软件触发或是某个内部事件触发。

双 DAC，有以下 3 种可能：(仅针对 GD32F170xx 和 GD32F190xx 产品)

1. 8 位数据右对齐：DAC1 通道的数据，DHR1[11:4]被写入寄存器 DACD\_R8DHR [7:0]位，DAC2 通道的数据，DHR2[11:4]被写入寄存器 DACD\_R8DHR [15:8]位。
2. 12 位数据右对齐：DAC1 通道的数据，DHR1[11:0]被写入寄存器 DACD\_R12DHR [11:0]位，DAC2 通道的数据，DHR2[11:0]被写入寄存器 DACD\_R12DHR [27:16]位。
3. 12 位数据左对齐：DAC1 通道的数据，DHR1[11:0]被写入寄存器 DACD\_L12DHR [15:4]位，DAC2 通道的数据，DHR2[11:0]被写入寄存器 DACD\_L12DHR [31:20]位。

根据 DACD\_yyDHR 寄存器，数据通过移位后被存储到 DHR1 和 DHR2。DHR1 和 DHR2 会自动，或者软件触发或外部事件触发后分别加载到 DAC1\_ODR 和 DAC2\_ODR。

### 13.3.4. DAC 转换

任何输出到 DAC 通道等待转换的数据必须首先存在 DHR 寄存器里(DAC1\_R8DHR, DAC1\_R12DHR, DAC1\_L12DHR, DAC2\_R8DHR, DAC2\_R12DHR, DAC2\_L12DHR, DACD\_R8DHR, DACD\_R12DHR, DACD\_L12DHR)。

如果没有选择硬件触发(寄存器 DAC\_CTLR 的 DTENx 位置‘0’)，存入 DACx\_yyDHR 寄存器的数据将会自动传至寄存器 DACx\_ODR。如果选择硬件触发(寄存器 DAC\_CTLR 的 DTENx 位置‘1’)，数据传输将在相应触发以后完成。

当 DACx\_yyDHR 寄存器的数据加载到 DACx\_ODR 寄存器，经过时间 tSETTLING 以后，模拟输出有效，tSETTLING 随着电源电压和模拟输出负载的不同会有所变化。

### 13.3.5. DAC 输出电压

DAC 通道引脚的模拟输出电压满足以下关系：

$$\text{DAC 输出} = V_{DDA} * \text{ODR} / 4095$$

数字输入被线性地转化为模拟输出电压，其范围为 0 到 V<sub>DDA</sub>。

### 13.3.6. DMA 请求

每个 DAC 都有一个 DMA 功能。对于 GD32F170xx 和 GD32F190xx 产品，两个 DMA 通道分别用于两个 DAC 的 DMA 请求。

如果 DDMAEN<sub>x</sub> 位置“1”，当有外部触发的时候（不是软件触发），则产生一个 DAC DMA 请求。然后，DAC<sub>x</sub>\_yyDHR 寄存器中的值被传送到 DAC<sub>x</sub>\_ODR 寄存器。

由于 DAC<sub>x</sub> 的 DMA 请求不能排队等待，因此如果第二个外部触发在第一个外部触应答之前达到，将不能处理第二个触发。之后 DAC\_STR 寄存器的 DDMAXUDR 位（DAC<sub>x</sub> 的 DMA 欠载标志位）被置位，错误状态被报告。DMA 数据传输功能被禁止并且不再处理 DMA 请求。DAC<sub>x</sub> 将继续转换旧的数据。

#### GD32F170xx 和 GD32F190xx 产品：

在双 DAC 模式下，如果两个 DAC 的 DDMAEN<sub>x</sub> 位为“1”，则产生两个 DMA 请求。

如果只需要一个 DMA 请求，只需要设置相关的 DDMAEN<sub>x</sub> 位。程序可以在只使用一个 DMA 请求和一个独立 DMA 通道的情况下，处理工作在双 DAC 模式的 2 个 DAC 通道。

### 13.3.7. DAC 触发

如果 DTEN<sub>x</sub> 控制位被置‘1’，DAC 转换可以由某一外部事件触发(定时器计数器，外部中断线)。

每次 DAC 接口检测到选定的触发输入的上升沿，最近存放在 DAC<sub>x</sub>\_yyDHR 寄存器里的数据会被传送到 DAC<sub>x</sub>\_ODR 寄存器中去。

如果选择软件触发，一旦 SWTR<sub>x</sub> 位被置‘1’，转换即开始。一旦寄存器 DAC<sub>x</sub>\_DHR 中的数据加载到寄存器 DAC<sub>x</sub>\_ODR，SWTR<sub>x</sub> 位会被硬件清零。

表 13-2. DAC 的外部触发源

DTSEL <sub>x</sub> [2:0]	触发源	触发类型
000	定时器6 TRGO事件	片上内部定时信号
001	定时器3 TRGO事件	
010	保留	
011	定时器15 TRGO事件	
100	定时器2 TRGO事件	
101	保留	
110	EXTI线路9	外部信号
111	SWTRIG	软件触发

### 13.3.8. 双 DAC 转换模式

本节内容适用于 **GD32F170xx** 和 **GD32F190xx** 产品。

为了更有效的利用总线带宽在两个 DAC 通道同时工作的情况下，DAC 有三个双 DAC 通道模式下的寄存器：DACP\_R8DHR, DACP\_R12DHR 和 DACP\_L12DHR。只需要访问一个寄存器即可同时驱动两个 DAC 通道。

对于双 DAC 通道和这些特殊寄存器，有几种转换模式可用。这些模式在只使用一个 DAC 通道的情况下，仍然可通过独立的 DHR<sub>x</sub> 寄存器操作。

### 独立触发

按照下面顺序设置 DAC 的这种转换模式：

- 分别设置两个 DAC 通道触发使能位 DTEN1 和 DTEN2
- 通过设置 DTSEL1[2:0] 和 DTSEL2[2:0] 位为不同的值，分别设置 2 个 DAC 通道的不同触发源
- 将双 DAC 通道转换的数据装入所需的 DACD\_DHR (DACD\_R8DHR, DACD\_R12DHR 或者 DACD\_L12DHR)

当 DAC1 通道触发到来时，DHR1 寄存器的值传入 DAC1\_ODR (三个 APB1 时钟周期后)。

当 DAC2 通道触发到来时，DHR2 寄存器的值传入 DAC2\_ODR (三个 APB1 时钟周期后)。

### 同时软件启动

按照下面顺序设置 DAC 在此转换模式：

- 将双 DAC 通道转换的数据装入所需的 DACD\_DHR (DACD\_R8DHR, DACD\_R12DHR 或者 DACD\_L12DHR)
- 在这种配置，一个 APB1 时钟周期后，DHR1 和 DHR2 寄存器的值立即传送到 DAC1\_ODR 和 DAC2\_ODR。

### 同时触发

按照下面顺序设置 DAC 在此转换模式：

- 分别设置两个 DAC 通道触发使能位 DTEN1 和 DTEN2
- 通过设置 DTSEL1[2:0] 和 DTSEL2[2:0] 位为相同的值，设置 2 个 DAC 的相同的触发源；
- 将双 DAC 转换的数据装入所需的 DACD\_DHR (DACD\_R8DHR, DACD\_R12DHR 或者 DACD\_L12DHR)
- 当一个触发到来时，DHR1 和 DHR2 寄存器的值立即传送到 DAC1\_ODR 和 DAC2\_ODR (三个 APB1 时钟周期后)。

## 13.4. DAC 寄存器

### 13.4.1. DAC 控制寄存器 (DAC\_CTLR)

**GD32F130xx 和 GD32F150xx 产品:**

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	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		DDMAUDRIE1	DDMAEN1	保留				DTSEL1[2:0]	DTEN1	DBOFF1	DEN1				
		rw	rw					rw	rw	rw	rw				

位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDMAUDRIE1	DAC1 DMA 欠载中断使能 0: DAC1 DMA 欠载中断关闭 1: DAC1 DMA 欠载中断使能
12	DDMAEN1	DAC1 DMA 使能 0: DAC1 DMA 模式关闭 1: DAC1 DMA 模式使能
11:6	保留	必须保持复位值
5:3	DTSEL1[2:0]	DAC1 触发选择 该位只能在位 DTEN1=1 时使用，用于选择外部触发事件。 000: 定时器 6 TRGO 事件 001: 定时器 3 TRGO 事件 010: 保留 011: 定时器 15 TRGO 事件 100: 定时器 2 TRGO 事件 101: 保留 110: 外部中断线 9 111: 软件触发
2	DTEN1	DAC1 触发使能 该位由软件来置位和清除，用来使能和关闭 DAC1 触发 0: DAC1 触发关闭 1: DAC1 触发使能

---

1	DBOFF1	关闭 DAC1 输出缓存 该位由软件来置位和清除，用来使能和关闭 DAC1 输出缓存。 0: 使能 DAC1 输出缓存 1: 关闭 DAC1 输出缓存
0	DEN1	DAC1 使能 该位由软件来置位和清除，用来使能和关闭 DAC1。. 0: 关闭 DAC1 1: 使能 DAC1

### GD32F170xx 和 GD32F190xx 产品：

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DDMAUDRIE2	DDMAEN2		保留				DTSEL2[2:0]		DTEN2	DBOFF2		DEN2		
	rw		rw						rw		rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DDMAUDRIE1	DDMAEN1		保留				DTSEL1[2:0]		DTEN1	DBOFF1		DEN1		
	rw		rw						rw		rw		rw		rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	DDMAUDRIE2	DAC2 DMA 欠载中断使能 0: DAC2 DMA 欠载中断关闭 1: DAC2 DMA 欠载中断使能
28	DDMAEN2	DAC2 DMA 使能 0: DAC2 DMA 模式关闭 1: DAC2 DMA 模式使能
27:22	保留	必须保持复位值
21:19	DTSEL2[2:0]	DAC2 触发选择 该位只能在位 DTEN2=1 时使用，用于选择外部触发事件。 000: 定时器 6 TRGO 事件 001: 定时器 3 TRGO 事件 010: 保留 011: 定时器 15 TRGO 事件 100: 定时器 2 TRGO 事件 101: 保留 110: 外部中断线 9 111: 软件触发

18	DTEN2	DAC2 触发使能 该位由软件来置位和清除，用来使能和关闭 DAC2 触发。 0: DAC2 触发关闭 1: DAC2 触发使能
17	DBOFF2	关闭 DAC2 输出缓存 该位由软件来置位和清除，用来使能和关闭 DAC2 输出缓存。 0: 使能 DAC2 输出缓存 1: 关闭 DAC2 输出缓存
16	DEN2	DAC2 使能 该位由软件来置位和清除，用来使能和关闭 DAC2。 0: 关闭 DAC2 1: 使能 DAC2
15:14	保留	必须保持复位值
13	DDMAUDRIE1	DAC1 DMA 欠载中断使能 0: DAC1 DMA 欠载中断关闭 1: DAC1 DMA 欠载中断使能
12	DDMAEN1	DAC1 DMA 使能 0: DAC1 DMA 模式关闭 1: DAC1 DMA 模式使能
11:6	保留	必须保持复位值
5:3	DTSEL1[2:0]	DAC1 触发选择 该位只能在位 DTEN1=1 时使用，用于选择外部触发事件。 000: 定时器 6 TRGO 事件 001: 定时器 3 TRGO 事件 010: 保留 011: 定时器 15 TRGO 事件 100: 定时器 2 TRGO 事件 101: 保留 110: 外部中断线 9 111: 软件触发
2	DTEN1	DAC1 触发使能 该位由软件来置位和清除，用来使能和关闭 DAC1 触发 0: DAC1 触发关闭 1: DAC1 触发使能
1	DBOFF1	关闭 DAC1 输出缓存 该位由软件来置位和清除，用来使能和关闭 DAC1 输出缓存。 0: 使能 DAC1 输出缓存 1: 关闭 DAC1 输出缓存
0	DEN1	DAC1 使能

该位由软件来置位和清除，用来使能和关闭 DAC1。

0: 关闭 DAC1

1: 使能 DAC1

### 13.4.2. DAC 软件触发寄存器 (DAC\_SWTR)

**GD32F130xx 和 GD32F150xx 产品：**

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	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															SWTR1

w

位/位域	名称	描述
31:1	保留	必须保持复位值
0	SWTR1	DAC1 软件触发，由硬件清 0 0: 关闭软件触发 1: 使能软件触发

**GD32F170xx 和 GD32F190xx 产品：**

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	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															SWTR2 SWTR1

w w

位/位域	名称	描述
31:2	保留	必须保持复位值
1	SWTR2	DAC2 软件触发，由硬件清 0 0: 关闭软件触发

1: 使能软件触发

0	SWTR1	DAC1 软件触发, 由硬件清 0 0: 关闭软件触发 1: 使能软件触发
---	-------	---

### 13.4.3. DAC1 12 位右对齐数据保持寄存器 (DAC1\_R12DHR)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



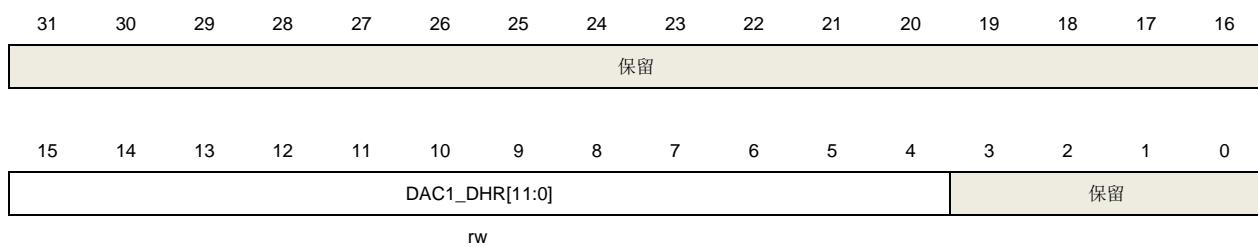
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC1_DHR[11:0]	DAC1 的 12 位 DAC 右对齐数据 这些位由软件写入, 表示 DAC1 的 12 位数据。

### 13.4.4. DAC1 12 位左对齐数据保持寄存器 (DAC1\_L12DHR)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	DAC1_DHR[11:0]	DAC1 的 12 位 DAC 左对齐数据 这些位由软件写入, 表示 DAC1 的 12 位数据。

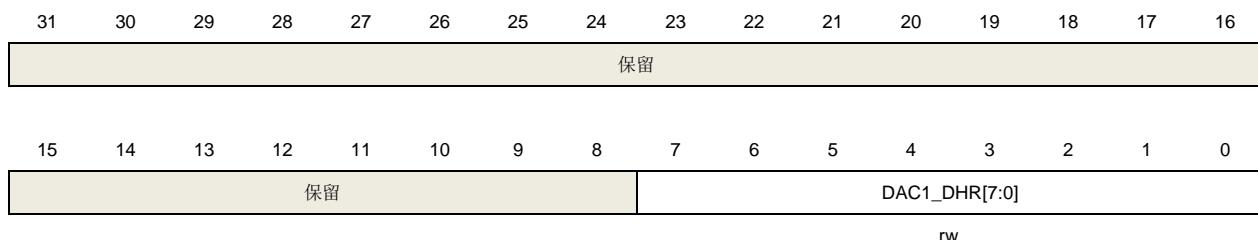
3:0 保留 必须保持复位值

### 13.4.5. DAC1 8 位右对齐数据保持寄存器 (DAC1\_R8DHR)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DAC1_DHR[7:0]	DAC1 的 8 位右对齐数据 这些位由软件写入, 表示 DAC1 的 8 位数据。

### 13.4.6. DAC2 12 位右对齐数据保持寄存器 (DAC2\_R12DHR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC2_DHR[11:0]	DAC2 12位右对齐数据 这些位由软件写入, 表示 DAC2 的 12 位数据

### 13.4.7. DAC2 12 位左对齐数据保持寄存器 (DAC2\_L12DHR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC2_DHR[11:0]										保留					

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	DAC2_DHR[11:0]	DAC2 12 位左对齐数据 这些位由软件写入, 表示 DAC2 的 12 位数据。
3:0	保留	必须保持复位值

### 13.4.8. DAC2 8 位右对齐数据保持寄存器 (DAC2\_R8DHR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								DAC2_DHR[7:0]							

rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DAC2_DHR[7:0]	DAC2 8位右对齐数据 这些位由软件写入, 表示 DAC2 的 8 位数据。

### 13.4.9. DAC 双模式 12 位右对齐数据保持寄存器 (Dacd\_R12DHR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				Dac2_Dhr[11:0]											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				Dac1_Dhr[11:0]											
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	Dac2_Dhr[11:0]	DAC2 12 位右对齐数据 这些位由软件写入, 表示 DAC2 的 12 位数据
15:12	保留	必须保持复位值
11:0	Dac1_Dhr[11:0]	DAC1 的 12 位 DAC 右对齐数据 这些位由软件写入, 表示 DAC1 的 12 位数据。

### 13.4.10. DAC 双模式 12 位左对齐数据保持寄存器(Dacd\_L12DHR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Dac2_Dhr[11:0]								保留							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dac1_Dhr[11:0]								保留							
rw															

位/位域	名称	描述
31:20	Dac2_Dhr[11:0]	DAC2 12位左对齐数据 这些位由软件写入, 表示 DAC2 的 12 位数据

---

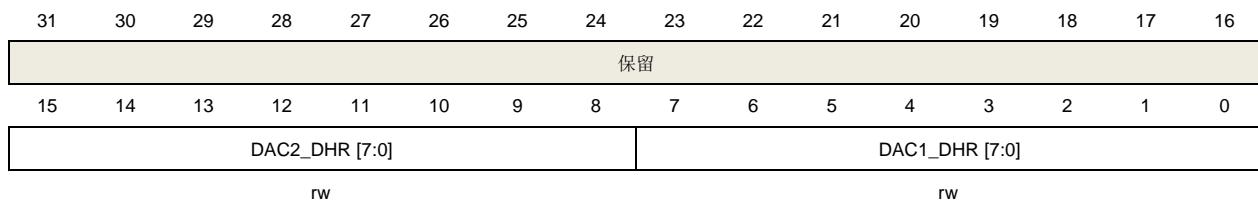
19:16	保留	必须保持复位值
15:4	DAC1_DHR[11:0]	DAC1 的 12 位 DAC 左对齐数据 这些位由软件写入，表示 DAC1 的 12 位数据。
3:0	保留	必须保持复位值

### 13.4.11. DAC 双模式 8 位右对齐数据保持寄存器(DACD\_R8DHR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	DAC2_DHR[7:0]	DAC2 8位右对齐数据 这些位由软件写入，表示 DAC2 的 8 位数据
7:0	DAC1_DHR[7:0]	DAC1 8 位右对齐数据 这些位由软件写入，表示 DAC1 的 8 位数据

### 13.4.12. DAC1 数据输出寄存器 (DAC1\_ODR)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:12	保留	必须保持复位值

11: 0      DAC1\_ODR[11:0]      DAC1 输出数据  
 只读，表示 DAC1 的输出数据。

### 13.4.13. DAC2 数据输出寄存器 (DAC2\_ODR) (仅适用于 GD32F170xx 和 GD32F190xx 产品)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:12	保留	必须保持复位值
11: 0	DAC2_ODR[11:0]	DAC2 输出数据 只读，表示 DAC2 的输出数据。

### 13.4.14. DAC 状态寄存器 (DAC\_STR)

GD32F130xx 和 GD32F150xx 产品:

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDMA1UDR	DAC1 的 DMA 欠载标志 该位由硬件置‘1’，软件清‘0’(通过写‘1’)。

0: 无 DMA 欠载出错发生

1: DMA 欠载出错发生 (DAC 的转换选择触发频率高于 DMA 的响应速度)

12:0 保留 必须保持复位值

### **GD32F170xx 和 GD32F190xx 产品:**

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DDMA2UDR	保留													
rc_w1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DDMA1UDR	保留													
rc_w1															

位/位域	名称	描述
31:30	保留	必须保持复位值
29	DDMA2UDR	DAC2 的 DMA 欠载标志 该位由硬件置‘1’，软件清‘0’(通过写‘1’)。 0: 无 DMA 欠载出错发生 1: DMA 欠载出错发生 (DAC 的转换选择触发频率高于 DMA 的响应速度)
28:14	保留	必须保持复位值
13	DDMA1UDR	DAC1 的 DMA 欠载标志 该位由硬件置‘1’，软件清‘0’(通过写‘1’)。 0: 无 DMA 欠载出错发生 1: DMA 欠载出错发生 (DAC 的转换选择触发频率高于 DMA 的响应速度)
12:0	保留	必须保持复位值

## 14. I2C 接口

### 14.1. 简介

I2C (集成电路间总线)模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式和快速模式，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)和 PMBus(电源管理总线)，此外还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

### 14.2. 主要特性

- 并行总线至 I2C 总线协议的转换及接口
- 同一接口既可实现主机功能又可实现从机功能
- 主从机之间的双向数据传输
- 支持 7 位和 10 位的地址模式和广播寻址
- 支持 I2C 多主机模式
- 支持标速(最高 100 kHz)和快速(最高 400 kHz)
- 从机模式下可配置的 SCL 主动拉低
- 支持 DMA 模式
- 兼容 SMBus 2.0 和 PMBus
- 两个中断：字节成功发送中断和错误事件中断
- 可选择的 PEC (报文错误校验)生成和校验

对于 GD32F170xx 和 GD32F190xx 产品，还有以下这些特征：

- 支持 SAM\_V 模式

### 14.3. 功能描述

I2C 接口的内部结构如下图所示。

图 14-1. I2C 模块框图

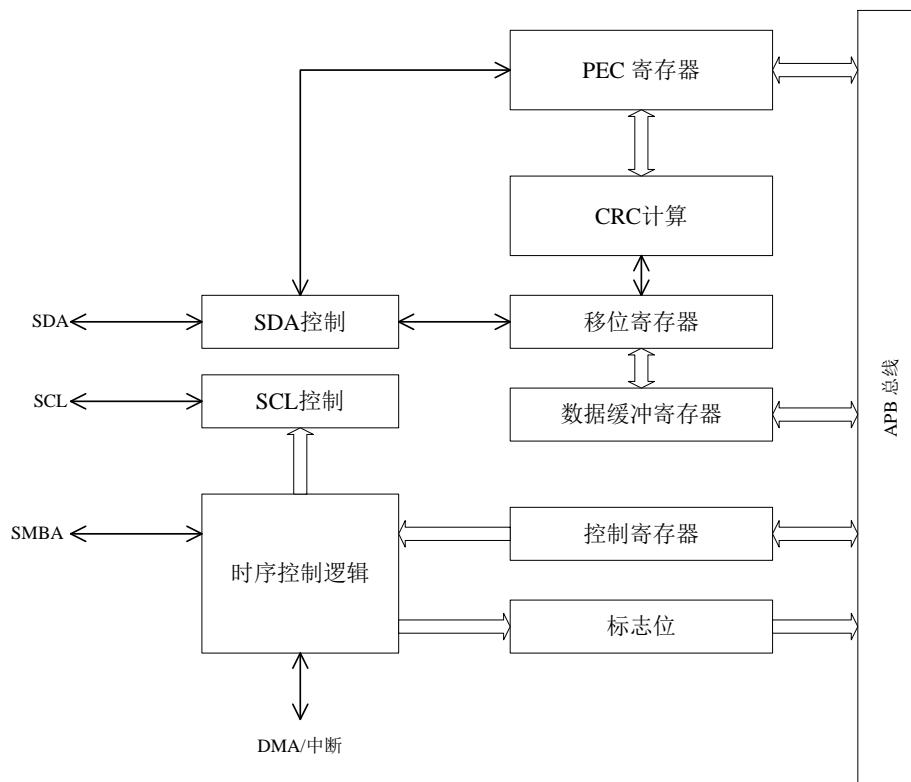


表 14-1. I2C 总线术语说明

术语	说明
发送器	发送数据到总线的设备
接收器	从总线接收数据的设备
主机	初始化数据传输，产生时钟信号和结束数据传输的设备
从机	由主机寻址的设备
多主	不破坏信息的前提下同时控制总线的多个主机
同步	同步两个或更多设备之间的时钟信号的过程
仲裁	如果超过一个主机同时试图控制总线，只有一个主机被允许，且获胜主机的信息不被破坏，保证上述的过程叫仲裁

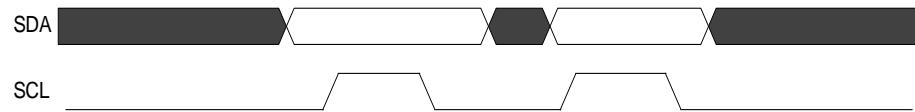
### 14.3.1. SDA 线和 SCL 线

I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须带开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 kbit/s，在快速模式下可以达到 400 kbit/s。由于 I2C 总线上可能会连接不同工艺的设备，逻辑‘0’和逻辑‘1’的电平并不是固定的，取决于 V<sub>DD</sub> 的实际电平。

### 14.3.2. 数据有效性

时钟信号的高电平期间 SDA 线上的数据必须稳定。只有在时钟信号 SCL 变低的时候数据线 SDA 的电平状态才能跳变(如图 14-2)。每个数据比特传输需要一个时钟脉冲。

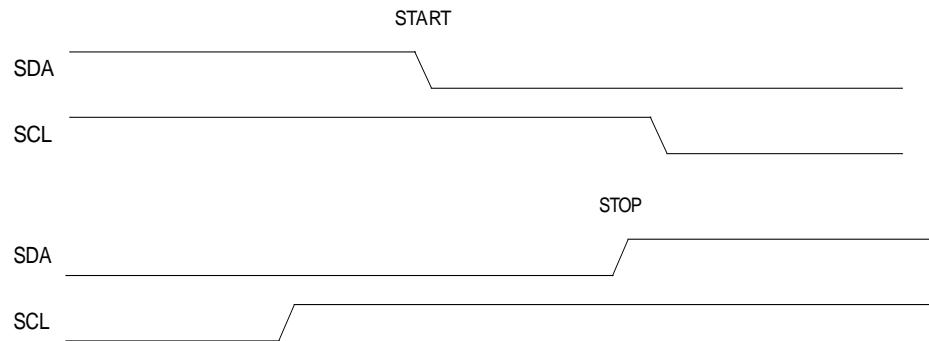
图 14-2. 数据有效性



### 14.3.3. 开始和停止状态

所有的数据传输起始于一个 START(S)结束于一个 STOP(P)(参见图 14-3)。START 起始位定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 结束位定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 14-3. 开始和停止状态

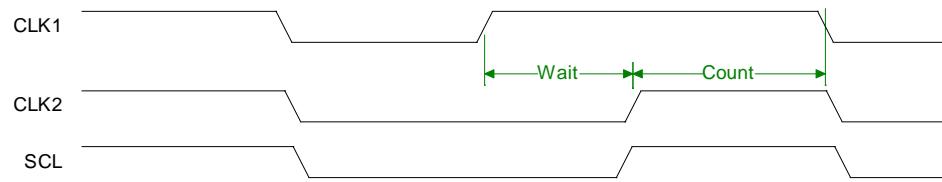


### 14.3.4. 时钟同步

两个主机可以同时在空闲总线上开始传送数据，因此必须通过一些机制来决定哪个主机获取总线的控制权，这一般是通过时钟同步和仲裁来完成的。单主机系统下不需要时钟同步和仲裁机制。

时钟同步通过 SCL 线的线与来实现。这就是说 SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦主器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平(参见图 14-4)。但是如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此 SCL 线被有最长低电平周期的器件保持低电平。此时低电平周期短的器件会进入高电平的等待状态。

图 14-4. 时钟同步



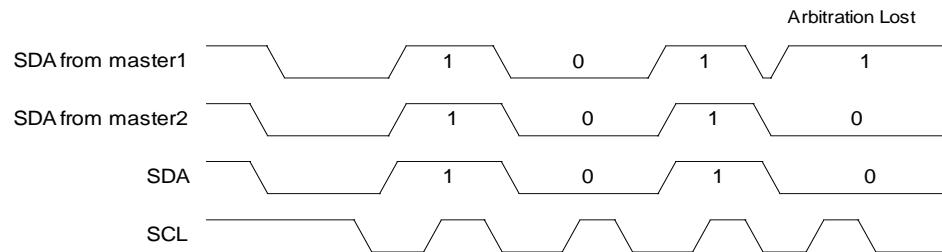
### 14.3.5. 仲裁

仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。

只有在总线空闲的时候主机才可以启动传输。两个主机可能在START起始位的最短保持时间内在总线上产生一个有效的START起始位，这种情况下需要仲裁来决定由哪个主机来完成传输。

仲裁逐位进行，在每一位的仲裁期间，当SCL为高时，每个主机都检查SDA电平是否和自己发送的相同。仲裁的过程需要持续很多位。理论上讲，如果两个主机所传输的内容完全相同，那么他们能够成功传输而不出现错误。如果一个主机发送高电平但检测到SDA电平为低，则认为自己仲裁失败并关闭自己的SDA输出驱动，而另一个主机则继续完成自己的传输。

图 14-5. SDA 线仲裁



### 14.3.6. I2C 通讯流程

每个I2C设备(不管是微控制器，LCD驱动，存储器或者键盘接口)都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。

I2C从机检测到I2C总线上的START起始位之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址（通过软件编程）进行比较，一旦两个地址相同，I2C从机将发送一个确认应答(ACK)，并响应总线的后续命令：发送或接受所要求的数据。此外，如果软件开启了广播呼叫，则I2C从机始终对一个广播地址 (0x00)发送确认应答。I2C模块始终支持7位和10位的地址。

I2C主机负责产生START起始位和STOP结束位来开始和结束一次传输，并且负责产生SCL时钟。

图 14-6. 7 位地址的 I2C 通讯流程

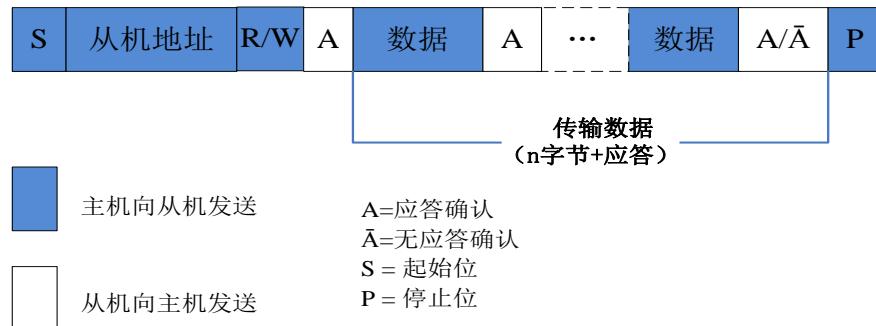
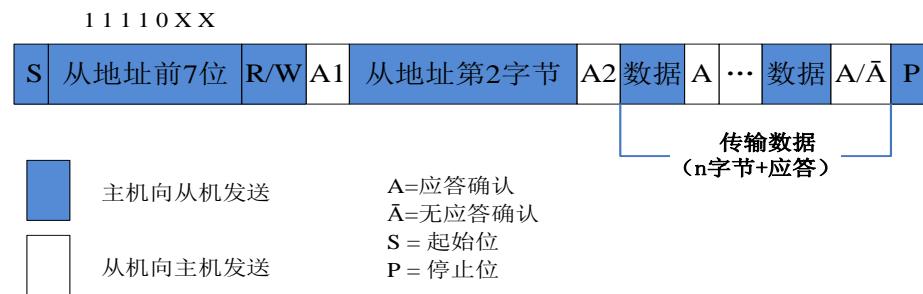


图 14-7. 10 位地址的 I2C 通讯流程



### 14.3.7. 软件编程模型

一个I2C设备例如LCD驱动器可能只是作为一个接收器，但是一个存储器既可以接收数据，也能发送数据。除了按照发送/接收方来区分，I2C设备也分为数据传输的主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。

不管I2C设备是主机还是从机，都可以发送或接收数据，因此，I2C设备有以下4种运行模式：

- 主机发送方
- 主机接收方
- 从机发送方
- 从机接收方

I2C模块支持以上四种模式。系统复位以后，I2C默认工作在从机模式下。通过软件配置使I2C在总线上发送一个START起始位之后，I2C变为主机模式，软件配置在I2C总线上发送STOP结束位后，I2C又变回从机模式。

#### 从机发送模式下的软件流程

在从机模式下要发送数据到I2C总线，软件应该按照下面的步骤来运行操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C\_CTLR2中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位和地址。

2. 接收一个START起始位及随后的地址，地址可以是7位格式也可以是10位格式，I2C硬件将I2C\_STR1寄存器的ADDSEND位置1，此位应该被软件查询或者中断监视，发现置位后，软件应该读I2C\_STR1寄存器然后读I2C\_STR2寄存器来清除ADDSEND位。如果地址是10位格式，I2C主机应该接着再产生一个START (Sr) 并发送一个地址头到I2C总线。从机在检测到START (Sr) 和紧接着的地址头之后会继续将ADDSEND位置1。软件也可以通过读I2C\_STR1寄存器和接着读I2C\_STR2寄存器来第二次清除ADDSEND位。
3. 现在I2C进入数据发送状态，由于移位寄存器和数据寄存器I2C\_DTR都是空的，硬件将TBE位置1。软件此时可以写入第一个字节数据到I2C\_DTR寄存器，但是TBE位并没有被清0，因为写入I2C\_DTR寄存器的字节被立即移入内部移位寄存器。当移位寄存器非空的时候，I2C开始发送数据到I2C总线。
4. 第一个字节的发送期间，软件可以写第二个字节到I2C\_DTR，此时TBE位被清0，因为I2C\_DTR寄存器和移位寄存器都不是空。
5. 第一个字节的发送完成之后，TBE被再次置起，软件可以写第三个字节到I2C\_DTR，同时TBE位被清0。在此之后，任何时候TBE被置1，只要依然有数据待被发送，软件都可以写入一个字节到I2C\_DTR寄存器。
6. 倒数第二个字节发送期间，软件写最后一个数据到I2C\_DTR寄存器来清除TBE标志位，之后就再不用关心TBE的状态。TBE位会在倒数第二个字节发送完成后置起，直到检测到STOP结束位时被清0。
7. 根据I2C协议，I2C主机将不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I2C从机的AE会置起以通知软件发送结束。软件写0到AE位可以清除此位。

**图 14-8. 从机发送模式**

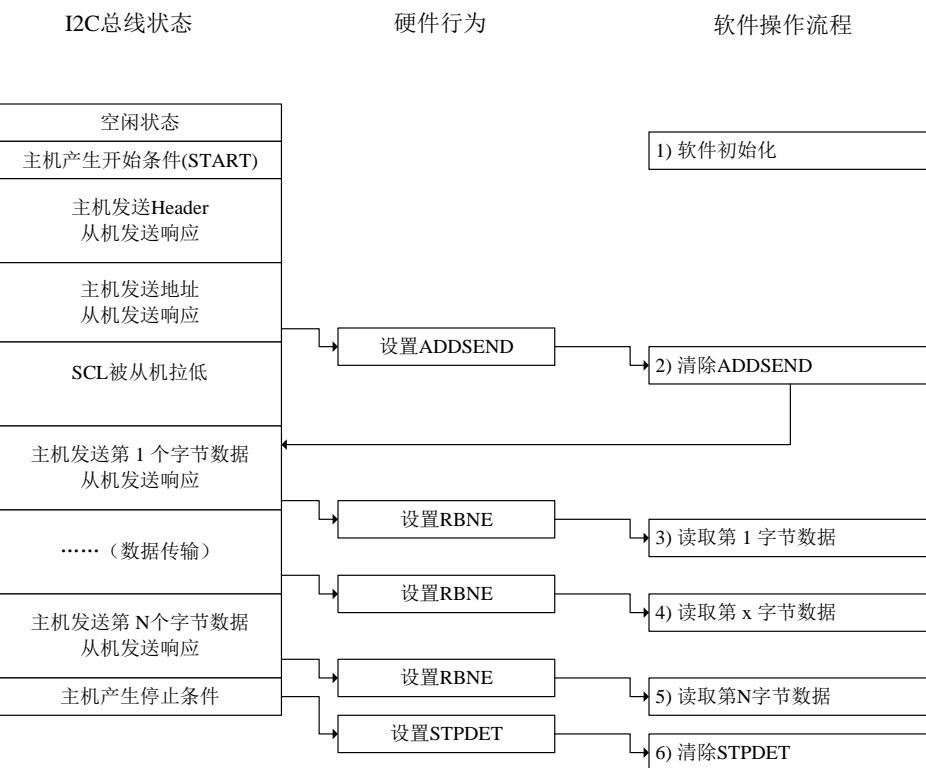

### 从机接收模式下的软件流程

如下图所示，在从机模式下接收数据时，软件应该遵循这些步骤来操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C\_CTLR2中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位以及地址。
2. 在接收到START起始条件和匹配的7位或10地址之后，I2C硬件将I2C状态寄存器的ADDSEND位置1，此位应该通过软件轮询或者中断来检测，发现置起后，软件通过先读I2C\_STR1寄存器然后读I2C\_STR2寄存器来清除ADDSEND位。一旦ADDSEND位被清0，I2C就开始接收来自I2C总线的数据。
3. 一旦接收到第一个字节，RBNE位被硬件置1，软件可以读取I2C\_DTR寄存器的第一个字节，此时RBNE位也被清0。
4. 任何时候RBNE被置1，软件可以从I2C\_DTR寄存器读取一个字节。
5. 接收到最后一个字节后，RBNE被置1，软件可以读取最后的字节。
6. 当I2C检测到I2C总线上一个STOP结束位，STPDET位被置1，软件通过先读I2C\_STR1寄

存器再写I2C\_CTLR1寄存器来清除STPDET位。

**图 14-9. 从机接收模式**



### 主机发送模式下的软件流程

如下图所示，在主机模式下发送数据到I2C总线时，软件应该遵循这些步骤来运行I2C模块：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C\_CTLR2中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位，随后等待I2C总线寻址。
2. 软件将GENSTA位置1，在I2C总线上产生一个START起始位。
3. 发送一个START起始位后，I2C硬件将I2C\_STR1的SBSEND位置1然后进入主机模式。现在软件应该读I2C\_STR1寄存器然后写一个7位地址位或带有地址头的10位地址位到I2C\_DTR寄存器来清除SBSEND位。一旦SBSEND位被清0，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位带地址头的地址，硬件在发送地址头的时候会将ADD10SEND位置1，软件应该通过读I2C\_STR1寄存器然后写10位低地址到I2C\_DTR来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该清除ADDSEND位（通过读I2C\_STR1寄存器然后读I2C\_STR2寄存器）。
5. I2C进入数据发送状态，因为移位寄存器和数据寄存器I2C\_DTR都是空的，所以硬件将TBE位置1。此时软件可以写第一个字节数据到I2C\_DTR寄存器，但是TBE位此时不会被清零，因为写入I2C\_DTR寄存器的字节被立即移入内部移位寄存器。一旦移位寄存器非空，I2C就开始发送数据到总线。

6. 在第一个字节的发送过程中，软件可以写第二个字节到I2C\_DTR，此时TBE会被清零。
7. 任何时候TBE被置1，软件都可以向I2C\_DTR寄存器写入一个字节，只要还有数据待发送。
8. 在倒数第二个字节发送过程中，软件写入最后一个字节数据到I2C\_DTR来清除TBE标志位，此后就不用关心TBE位的状态。TBE位会在倒数第二个字节发送完成后被置起，直到发送STOP结束位时被清零。
9. 最后一个字节发送结束后，I2C主机将BTC位置起，因为移位寄存器和I2C\_DTR寄存器此时都为空。软件此时应该配置GENSTP来发送一个STOP结束位，此后TBE和BTC状态位都将被清0。

**图 14-10. 主机发送模式**



### 主机接收模式下的软件流程

在主机接收模式下，主机需要为最后一个字节接收产生NACK，然后发送STOP结束位。因此，需要额外注意以确保最后接收到数据的正确性。下面提供了两种针对主机接收模式的软件编程方案，方案A需要保证软件能对I2C的中断进行快速响应，方案B则不需要。

## 方案 A

1. 首先，软件应该使能I2C外设时钟，以及配置I2C\_CTLR2中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位，随后等待I2C总线寻址。
2. 软件将GENSTA位置1，从而在I2C在总线上产生一个START起始位
3. 发送一个START起始位后，I2C硬件将I2C\_STR1的SBSEND位置1然后进入主机模式。现在软件应该读I2C\_STR1寄存器然后写一个7位地址位或带有地址头的10位地址位到I2C\_DTR寄存器来清除SBSEND位。一旦SBSEND位被清0，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位带地址头的地址，硬件在发送地址头的时候会先将ADD10SEND位置1，软件应该通过读I2C\_STR1寄存器然后写10位低地址到I2C\_DTR来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该清除ADDSEND位，通过读I2C\_STR1寄存器然后读I2C\_STR2寄存器。如果地址是10位格式，软件应该再次将GENSTA位置1来重新产生一个START (Sr)。在START产生后，SBSEND位会被置1。软件应该通过先读I2C\_STR1然后写地址头到I2C\_DTR来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C\_STR1然后读I2C\_STR2来清除ADDSEND位。
5. 一旦接收到第一个字节，硬件会将RBNE位置1。此时软件可以从I2C\_DTR 寄存器读取第一个字节，之后RBNE位被清0。
6. 此后任何时候RBNE被置1，软件就可以从I2C\_DTR寄存器读取一个字节。
7. 接收完倒数第二个字节(N-1)数据之后，软件应该立即将ACKEN位清0，并将GENSTP位置1，这一过程需要在最后一个字节接收完毕之前完成，以确保NACK发送给最后一个字节。
8. 最后一个字节接收完毕后，RBNE位被置1，软件可以读取最后一个字节。由于ACKEN已经在前一步骤中被清0，I2C不再为最后一个字节发送ACK，并在最后一个字节发送完毕后产生一个STOP结束位。

以上步骤要求字节数目N>1，如果N=1，步骤7应该在步骤4之后就执行，且需要在字节接收完成之前完成。

图 14-11. 主机接收使用方案 A 模式



## 方案 B

1. 首先，软件应该使能I2C外设时钟，配置I2C\_CTLR2中时钟相关寄存器来确保正确的I2C时序。初始化完成之后，I2C运行在默认的从机模式状态，等待START起始位和地址。
2. 软件将GENSTA位置1从而产生一个起始位
3. 发送一个START起始位后，I2C硬件将I2C\_STR1的SBSEND位置1然后进入主机模式。现在软件应该读I2C\_STR1寄存器然后写一个7位地址位或带有地址头的10位地址位到I2C\_DTR寄存器来清除SBSEND位。一旦SBSEND位被清0，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位带地址头的地址，硬件在发送地址头的时候会先将ADD10SEND位置1，软件应该通过读I2C\_STR1寄存器然后写10位低地址到I2C\_DTR来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该清除ADDSEND

位，通过读I2C\_STR1寄存器然后读I2C\_STR2寄存器。如果地址是10位格式，软件应该接着将GENSTA位再次置1来产生一个开始条件（Sr），Sr被发送出去以后SBSEND位被再次置1。软件应该通过先读I2C\_STR1然后写地址头到I2C\_DTR来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C\_STR1然后读I2C\_STR2来清除ADDSEND位。

5. 一旦第一个字节被接收，RBNE位会被硬件置1。此时软件可从I2C\_DTR寄存器读取出第一个字节，同时RBNE位被清0。
6. 此后任何时候，一旦RBNE位被置1，软件就可以从I2C\_DTR寄存器读取一个字节的数据，直到主机接收了N-3个字节。
7. 如图14-12所示，第N-2个字节还没被软件读出，之后第N-1个字节被接收，此时BTC和RBNE都被置位，总线就会被主机锁死以阻止最后一个字节的接收。然后软件应该清除ACKEN位。
8. 软件从I2C\_DTR读出倒数第三个（N-2）字节数据，同时也将BTC位清0。此后第N-1个字节从移位寄存器被移到I2C\_DTR，总线得到释放然后开始接收最后一个字节，由于ACKEN已经被清除，因此主机不会给最后一个字节数据发送ACK响应。
9. 最后一个字节接收完毕后，硬件再次把BTC位和RBNE置1，并拉低SCL，软件将GENSTP位置1，主机发出一个STOP结束位。
10. 软件读取第N-1个字节，清除BTC。此后最后一个字节从移位寄存器被移动到I2C\_DTR。
11. 软件读取最后一个字节，清除RBNE。

以上步骤需要字节数字N>2，N=1和N=2的情况近似。

#### N=1

在第4步，软件应该在清除ADDSEND位之前将ACK位清0，在清除ADDSEND位之后将GENSTP位置1。当N=1时步骤5是最一步。

#### N=2

在第2步，软件应该在GENSTA置1之前将POAP置1。在第4步，软件应该在清除ADDSEND位之前将ACK位清0。在第5步，软件应该一直等到BTC位被置1然后将GENSTP位置1且读取I2C\_DTR两次。

图 14-12. 主机接收使用方案 B 模式



### DMA模式下的配置模式

按照前面的软件流程，每当TBE位和RBNE位被置1之后，软件都应该写或读一个字节，这样将导致CPU的负荷较重。I2C的DMA功能可以在TBE或RBNE位置1时，自动进行一次写或读操作，从而减轻了CPU的负荷，具体DMA的配置请参看DMA相关章节。

#### 14.3.8. 报文错误校验

I2C模块中有一个PEC模块，它使用CRC-8计算器来执行I2C数据的报文校验，CRC多项式为x8

$x_8 + x_7 + x_6 + x_5 + x_4 + x_3 + x_2 + x_1 + 1$ , 和SMBus协议兼容。将PECEN位置1就可以使能PEC功能。PEC将会计算所有通过I2C总线发送的数据（包括地址）。在非DMA模式下，软件可以通过配置PECTRANS来控制I2C在最后一个字节发送完毕后发送PEC值，或者在接收完成后检查接收到的PEC值是否正确。在DMA模式下，如果PECEN位被置1，I2C将自动发送或者检查PEC值。

### 14.3.9. SMBus 支持

系统管理总线（System Management Bus，简写为SMBus或SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus最常见于计算机主板，主要用于电源传输ON/OFF指令的通信。SMBus是I2C的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见Smart BatteryData）。

#### SMBus 协议

SMBus上每个报文交互都遵从SMBus协议中预定义的格式。SMBus是I2C规范中数据传输格式的子集。只要I2C设备可通过SMBus协议之一进行访问，便视为兼容SMBus规范。不符合这些协议的I2C设备，将无法被SMBus和ACPI规范所定义的标准方法访问。

#### 地址解析协议

SMBus采用了I2C硬件以及I2C的硬件寻址方式，但在I2C的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是SMBus规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。协议中有个非常特别之处在于：系统的主机和所有其它设备能够有定义其名称和功能。

#### 超时特性

SMBus有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为10kHz——为了防止长时间锁死总线。I2C在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束后，将可以继续I2C会话。I2C总线协议中并没有限制这个延时的上限，但在SMBus系统中，这个时间被限定为35ms。按照SMBus协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种（问题）状态。这样就并不允许从设备将时钟拉低太长时间。

#### 报文错误校验

SMBus 2.0以及1.1都采用了报文错误校验（Packet Error Checking，缩写为PEC）。在这种模式中，每次会话最后都将传输PEC（报文错误码）字节。该字节是按照CRC-8校验和的方式计算的，计算范围包括整个报文，包括地址以及读/写位。所采用的多项式为 $x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ （CRC-8-ATMHEC算法，初始化为0）。

### SMBus 警报

SMBus 还有一个额外的共享的中断信号，称为SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。SMBus中还定义了较少见的“主机提醒协议（Host Notify Protocol）”，基于I2C 多主模式实现类似的提醒功能，但是可以传递更多数据。

### SMBus 通讯流程

SMBus的通讯流程和标准I2C的流程相似。如果一个应用要使用SMBus模式，那么在程序中需要配置几个SMBus特定的寄存器、响应一些SMBus特定标志位、实现那些在SMBus手册中介绍的上层协议。

1. 在通信之前，需要设置I2C\_CTLR1中SMBEN=1，并且根据需求，配置SMBSEL和ARPEN的值。
2. 为了支持ARP协议（ARPEN=1），在SMBus主机模式下（SMBTYPE=1），软件需要响应标志位HSTSMB（在SMBus从机模式下，响应DEFSMB标志位），并实现ARP协议中的功能。
3. 为了支持SMBus警告模式，软件应该响应SMBALTS标志位，并实现相应的功能。

#### 14.3.10. 状态、错误和中断

I2C有一些状态和错误标志位和中断，通过设置一些寄存器位，便可以从这些标志触发中断（详情参见I2C寄存器）。

表 14-2. 事件状态标志位

事件标志位名称	说明
SBSEND	主机发送 START 起始位
ADDSEND	地址发送和接收
ADD10SEND	10 位地址模式中地址头发送
STPDET	监测到 STOP 结束位
BTC	字节发送结束
TBE	发送时 I2C_DTR 为空
RBNE	接收时 I2C_DTR 非空

表 14-3. I2C 错误标志位

I2C 错误名称	说明
BE	总线错误
LOSTARB	仲裁丢失
RXORE	当禁用 SCL 拉低后，发生了溢出或下溢
AE	没有接收到应答
PECE	CRC 值不相同
SMBTO	SMBus 模式下总线超时
SMBALTS	SMBus 警报

## 14.4. I2C 寄存器

### 14.4.1. I2C 控制寄存器 1 (I2C\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRESET	保留	SALT	PECTRANS	POAP	ACKEN	GENSTP	GENSTA	DISSTRC	GCEN	PECEN	ARPEN	SMBSEL	保留	SMBEN	I2CEN
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
15	SRESET	软件复位I2C， 软件应该在I2C总线被释放时复位I2C 0: I2C未复位 1: I2C复位
14	保留	必须保持复位值
13	SALT	软件置1和清0， 硬件清0。 0: 不通过SMBA发布警告 1: 通过SMBA引脚发送警告
12	PECTRANS	PEC传输 软件置1和清0， 硬件在以下条件下清除此位：PEC传输完成， 监测到START/STOP结束位， I2CEN=0。 0: 不传输PEC值 1: 传输的PEC值
11	POAP	ACK/PEC的位置含义 软件置1和清0， 当I2CEN=0时， 硬件清0。 0: ACKEN位决定对当前正在接收的字节是否发送ACK； PECTRANS位表明PEC是否处于移位寄存器中 1: ACKEN位决定是否对下一个字节发送ACK， PECTRANS位表明下一个即将被接收的字节是PEC
10	ACKEN	是否发送ACK 软件置1和清0， 当I2CEN=0时硬件清0。 0: 不发送ACK 1: 发送ACK
9	GENSTP	I2C总线上产生一个STOP结束位 软件置1和清0， SMBus超时时， 硬件置1， 监测到STOP结束位时， 硬件清0。 0: 不发送STOP 1: 发送STOP

8	GENSTA	I2C总线上产生一个START起始位 软件置1和清0，当监测到START起始位或I2CEN=0时由硬件清0。 0: 不发送START 1: 发送START
7	DISSTRC	在从机模式下数据未就绪是否将SCL拉低 软件置1和清0。 0: 拉低SCL 1: 不拉低SCL
6	GCEN	是否响应对地址(0x00)的广播呼叫 0: 从机不响应广播呼叫 1: 从机将响应广播呼叫
5	PECEN	PEC计算开关 0: PEC计算关闭 1: PEC计算打开
4	ARPEN	SMBus下ARP协议开关 0: 关闭ARP 1: 开启ARP
3	SMBSEL	SMBus类型选择 0: 从机 1: 主机
2	保留	必须保持复位值
1	SMBEN	SMBus/I2C模式开关 0: I2C模式 1: SMBus模式
0	I2CEN	I2C外设使能 0: 禁用I2C 1: 使能I2C

#### 14.4.2. I2C 控制寄存器 2 (I2C\_CTLR2)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DMALST	DMAON	BIE	EE	EIE	保留	I2CCLK[5:0]								

rw rw rw rw rw rw rw

位/位域	名称	描述
------	----	----

15:13	保留	必须保持复位值
12	DMALST	DMA最后传输标志位 0: 下一个DMA EOT不是最后传输 1: 下一个DMA EOT是最后传输
11	DMAON	DMA模式开关 0: DMA模式关 1: DMA模式开
10	BIE	缓冲区中断使能 0: 当TBE = 1或RBNE = 1时无中断 1: 如果ITEVTEN=1, 当TBE = 1或RBNE = 1时产生中断
9	EE	事件中断使能 0: 禁用事件中断 1: 使能事件中断, 意味着当SBSEND、ADDSEND、ADD10SEND、STPDET或BTC标志位有效或当BIE=1时TBE=1或RBNE=1时产生中断。
8	EIE	错误中断使能 0: 禁用错误中断 1: 使能错误中断, 意味着当BE、LOSTARB、AE、RXORE、PECE、SMBTO或SMBALTS标志位生效时产生中断。
7:6	保留	必须保持复位值
5:0	I2CCLK[5:0]	I2C外设时钟频率 I2CCLK[5:0]应该是输入APB时钟频率, 最低2MHz。 0h - 1h: 无时钟 2h - 36h: 2 MHz~36MHz 37h - 63h: 由于APB时钟限制, 无时钟

#### 14.4.3. I2C 本机地址寄存器 1 (I2C\_AR1)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDFORMAT	保留				ADDRESS[9:8]	ADDRESS[7:1]				ADDRESS0					

位/位域	名称	描述
15	ADDFORMAT	I2C从机地址模式 0: 7位地址 1: 10位地址

---

14:10	保留	必须保持复位值
9:8	ADDRESS[9:8]	10位地址的最高两位
7:1	ADDRESS[7:1]	7位地址或者10位地址的第7-1位
0	ADDRESS0	10位地址的第0位

#### 14.4.4. I2C 本机地址寄存器 2 (I2C\_AR2)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						ADDRESS2[7:0]						DUADEN			
rw														rw	

位/位域	名称	描述
15:8	保留	必须保持复位值
7:1	ADDRESS2[7:0]	从机在双重地址模式下第二个I2C地址
0	DUADEN	双重地址模式开关 0: 双重地址模式关 1: 双重地址模式开

#### 14.4.5. I2C 传输缓冲区寄存器 (I2C\_DTR)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:8	保留	必须保持复位值
7:0	TRB[7:0]	数据发送接收缓冲区

#### 14.4.6. I2C 传输状态寄存器 1 (I2C\_STR1)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALTS	SMBTO	保留	PECE	RXORE	AE	LOSTARB	BE	TBE	RBNE	保留	STPDET	ADD10SEND	BTC	ADDSEND	SBSEND
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

位/位域	名称	描述
15	SMBALTS	SMBus警报状态 硬件置1，软件写0清0。 0: SMBA引脚未被拉低(从机模式)或未监测到警报(主机模式) 1: SMBA引脚被拉低(从机模式)或监测到警报(主机模式)
14	SMBTO	SMBus模式下超时信号 硬件置1，软件写0清0。 0: 无超时错误 1: 超时事件发生(SCL被拉低达25ms)
13	保留	必须保持复位值
12	PECE	接收数据时PEC错误 硬件置1，软件写0清0。 0: 接收到PEC且校验正确 1: 接收到PEC但校验错误，此时I2C将无视ACKEN位直接发送NACK
11	RXORE	当禁用SCL 拉低功能后，在从机模式下发生了过载或欠载事件。在从机接收模式下，假如DTR 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生过载。在从机发送模式下，假如当前字节已经发送完成，而DTR 仍然为空，就会发生欠载。 硬件置1，软件写0清0。 0: 无溢出和欠载错误发生 1: 发生溢出或欠载错误
10	AE	应答错误 硬件置1，软件写0清0。 0: 未发生应答错误 1: 发生了应答错误
9	LOSTARB	主机模式下仲裁丢失 硬件置1，软件写0清0。 0: 无仲裁丢失 1: 发生仲裁丢失，I2C模块返回从机模式。
8	BE	总线错误，表示I2C总线上发生了预料之外的START起始位或STOP结束位。 硬件置1，软件写0清0。 0: 无总线错误 1: 发生了总线错误
7	TBE	发送期间I2C_DTR为空

硬件从I2C_DTR寄存器移动一个字节到移位寄存器之后将此位置1，软件写一个字节到I2C_DTR寄存器清除该位。如果移位寄存器和I2C_DTR寄存器都是空的，写I2C_DTR寄存器将不会清除TBE位（详见主机/从机发送模式下的软件操作流程）		
0: I2C_DTR非空 1: I2C_DTR空，软件可以写		
6 RBNE 接收期间I2C_DTR非空 硬件从移位寄存器移动一个字节到I2C_DTR寄存器之后将此位置1，读此位可以清除此位。如果BTC和RBNE都被置1，读I2C_DTR将不会清除RBNE，因为移位寄存器的字节已经被立即移到I2C_DTR. (详见主机/从机接收模式下的软件操作流程) 0: I2C_DTR为空 1: I2C_DTR非空，软件可以读		
5 保留 必须保持复位值		
4 STPDET 从机模式下监测到STOP结束位 此位被硬件置1，先读I2C_STR1然后写CTLR1可以清除此位。 0: 从机模式下未监测到STOP结束位 1: 从机模式下监测到STOP结束位		
3 ADD10SEND 主机模式下10位地址地址头被发送 该位由硬件置1，软件读I2C_STR1和写I2C_DTR清除此位。 0: 主机模式下未发送10位地址的地址头 1: 主机模式下发送10位地址的地址头		
2 BTC 字节发送结束 接收模式下，如果一个字节已经被移位寄存器接收但是此时I2C_DTR寄存器仍然是满的；或者发送模式下，一个字节已经被移位寄存器发送但是I2C_DTR寄存器仍然是空的，如果使能了SCL拉低功能，硬件就会置起BTC标志位。 此位由硬件置1，软件读I2C_STR1和读或写I2C_DTR寄存器清除此位。 0: 未发生BTC 1: 发生了BTC		
1 ADDSEND 主机模式下：成功发送了地址 从机模式下：接收到地址并且和自身的地址匹配 此位由硬件置1，软件读I2C_STR1寄存器和读STR2清0。 0: 无地址被发送或接收 1: 地址在主机模式下被发送或从机模式下接收到匹配地址		
0 SBSEND 主机模式下发送START起始位 此位由硬件置1，软件读I2C_STR1和写I2C_DTR清0。 0: 未发送START条件 1: START条件被发送		

#### 14.4.7. I2C 传输状态寄存器 2 (I2C\_STR2)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ECV[7:0]		DUMODF	HSTSMB	DEFSMB	RXGC	保留	TRS	I2CBSY	MASTER	

r                    r                    r                    r                    r                    r                    r                    r                    r

位/位域	名称	描述
15:8	ECV[7:0]	当PEC使能后硬件计算出的PEC值。
7	DUMODF	从机模式下双标志位表明哪个地址和双地址模式匹配 STOP或START起始位产生后或I2CEN=0时此位由硬件清0。 0: 地址和OAR1匹配 1: 地址和OAR2匹配
6	HSTSMB	从机模式下监测到SMBus主机地址头 STOP或START起始位产生后或I2CEN=0时此位由硬件清0。 0: 未监测到SMBus主机地址头 1: 监测到SMBus主机地址头
5	DEFSMB	从机模式下SMBus主机地址头 STOP或START起始位产生后或I2CEN=0时此位由硬件清0。 0: SMBus设备没有缺省地址 1: 从SMBus设备接收到一个缺省的地址
4	RXGC	是否接收到广播地址(00h) STOP或START起始位产生后或I2CEN=0时此位由硬件清0。 0: 未接收到广播呼叫地址 1: 接收到广播呼叫地址(00h)
3	保留	必须保持复位值
2	TRS	I2C作发送端还是接收端 STOP或START起始位产生后或I2CEN或LOSTARB=0时此位由硬件清0。 0: 接收端 1: 发送端
1	I2CBSY	忙标志 STOP结束位后硬件清0。 0: 无I2C通讯 1: I2C正在通讯
0	MASTER	表明I2C时钟在主机模式还是从机模式的标志位 STOP或START起始位产生后或I2CEN或LOSTARB=0时此位由硬件清0。 0: 从机模式 1: 主机模式

#### 14.4.8. I2C 时钟配置寄存器 (I2C\_CLKR)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15	FAST	主机模式下I2C速度选择 0: 标准速度 1: 快速
14	DTCY	快速模式下占空比 0: $T_{low}/T_{high} = 2$ 1: $T_{low}/T_{high} = 16/9$
13:12	保留	必须保持复位值
11:0	CLKC[11:0]	主机模式下I2C时钟控制 标准速度模式下: $T_{high} = T_{low} = CCR * T_{PCLK1}$ 如果 DTCY=0, 快速模式下: $T_{high} = CCR * T_{PCLK1}$ , $T_{low} = 2 * CCR * T_{PCLK1}$ 如果 DTCY=1, 快速模式下: $T_{high} = 9 * CCR * T_{PCLK1}$ , $T_{low} = 16 * CCR * T_{PCLK1}$

#### 14.4.9. I2C 上升时间寄存器 (I2C\_RTR)

地址偏移: 0x20

复位值: 0x0002

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:6	保留	必须保持复位值
5:0	RISETIME[5:0]	主机模式下最大上升时间 RISETIME 值应该为 SCL 最大上升时间加 1

#### 14.4.10. I2C SAM 控制状态寄存器 (I2C\_SAMCSR) (仅适用于 GD32F170xx and GD32F190xx 产品)

地址偏移：0x80

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFR	RFF	TFR	TFF	保留	RXF	TXF	RFRI E	RFFI E	TFRI E	TFFI E	保留	STOE	SAM E		
r_w0	r_w0	r_w0	r_w0		r	r	rW	rW	rW	rW		rW	rW		

位/位域	名称	描述
15	RFR	接收帧上升沿标志, 由软件写0清除
14	RFF	接收帧下降沿标志, 由软件写0清除
13	TFR	发送帧上升沿标志, 由软件写0清除
12	TFF	发送帧下降沿标志, 由软件写0清除
11:10	保留	必须保持复位值
9	RXF	接收帧信号电平
8	TXF	发送帧信号电平
7	RFRIE	接收帧上升沿中断使能 0: 失能 1: 使能
6	RFFIE	接收帧下降沿中断使能 0: 失能 1: 使能
5	TFRIE	发送帧上升沿中断使能 0: 失能 1: 使能
4	TFFIE	发送帧下降沿中断使能 0: 失能 1: 使能
3:2	保留	必须保持复位值
1	STOE	SAM_V接口超时检测使能 0: 失能 1: 使能

0

SAME

SAM\_V 接口使能

0: 失能

1: 使能

## 15. 串行外设接口 / 片上音频接口 (**SPI/I2S**)

### 15.1. 简介

该模块可通过 SPI 协议或 I2S 协议与外部设备进行通信。

串行外设接口 (Serial Peripheral Interface, 缩写为 SPI) 提供了 SPI 协议的数据发送及接收功能，可工作在主机或从机模式下。对于 GD32F130xx 和 GD32F150xx 产品，SPI 接口支持单路配置下主机和从机模式；对于 GD32F170xx 和 GD32F190xx 产品，SPI 接口不仅支持单路配置下主机和从机模式，还支持四路配置下主机模式。在 SPI 单路配置下，SPI 接口用到 4 根管脚，分别是串行数据输入脚 MISO、串行数据输出脚 MOSI、串行时钟脚 SCK 以及从机选择脚 NSS。在 SPI 四路配置下，SPI 使用六个管脚：MOSI、MISO、IO2、IO3、SCK、NSS。总线上的一个 SPI 设备作为主机工作，通过 NSS 以及 SCK 信号控制数据流，指示数据通信的开始，并控制数据采样速率。要接收一个数据字节，数据位流将在指定的时钟边沿锁存，并将结果保存到数据寄存器中。数据发送的执行方式类似，但顺序与之相反。模式故障检测功能实现了多主功能。SPI 接口可用于多种应用，包括只使用两线的单工同步通信（其中一根数据线可能双向），或采用 CRC 校验的可靠数据通信。

片上音频接口 (Inter-IC Sound, 缩写为 I2S) 支持四种音频标准，分别是 I2S 飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

### 15.2. 主要特性

#### 15.2.1. SPI 特性

- 主设备或从设备模式
- 可配置的时钟位速率
- 可配置的时钟极性和时钟相位
- 16 位宽、相互独立的发送接收缓冲器
- 可配置的 8 或 16 位数据帧大小
- 可配置高位在前或者低位在前的数据格式
- 硬件 CRC 计算，可自动发送 CRC 错误校验
- 可通过 3 线实现全双工同步传输
- 可通过 2 线实现单工同步传输
- 主机和从机都可配置 NSS 工作于软件模式或硬件模式
- SPI 总线忙状态标志

- 发送和接收标志，并可映射至中断
- 主机模式故障、过载以及 CRC 校验错误标志，并可映射至中断
- 发送和接收支持 DMA 模式

对于 **GD32F170xx** 和 **GD32F190xx** 产品，还有以下这些特征：

- 支持 SPI 四路配置下的主设备模式(SPI2)

### 15.2.2. I2S 特性

- 支持的 I2S 标准：
  - I2S 飞利浦标准
  - MSB 对齐标准
  - LSB 对齐标准
  - PCM 标准 (支持短帧同步和长帧同步两种模式)
- 支持的运行模式：
  - 主机发送
  - 主机接收
  - 从机发送
  - 从机接收
- 数据长度可以是 16 位、24 位或 32 位
- 通道长度可以是 16 位或 32 位
- 16 位的移位寄存器用来发送和接收数据
- 数据方向始终是 MSB 在前
- 8 位可配置的线性预分频器，可实现精确的从 8kHz 到 192kHz 的音频采样率
- 可配置的空闲状态时钟极性
- 可以输出主机时钟驱动外部音频器件
- 提供发送欠载错误标志(TXURE)和接收过载错误标志(RXORE)
- 发送和接收都带有 DMA 功能

## 15.3. SPI 功能描述

### 15.3.1. 引脚配置（SPI 单路模式，默认）

在不同模式下，SPI 通过 4 个引脚与外部设备相连：

**MISO**: 此引脚用于在从模式下的数据发送和主模式下的数据接收

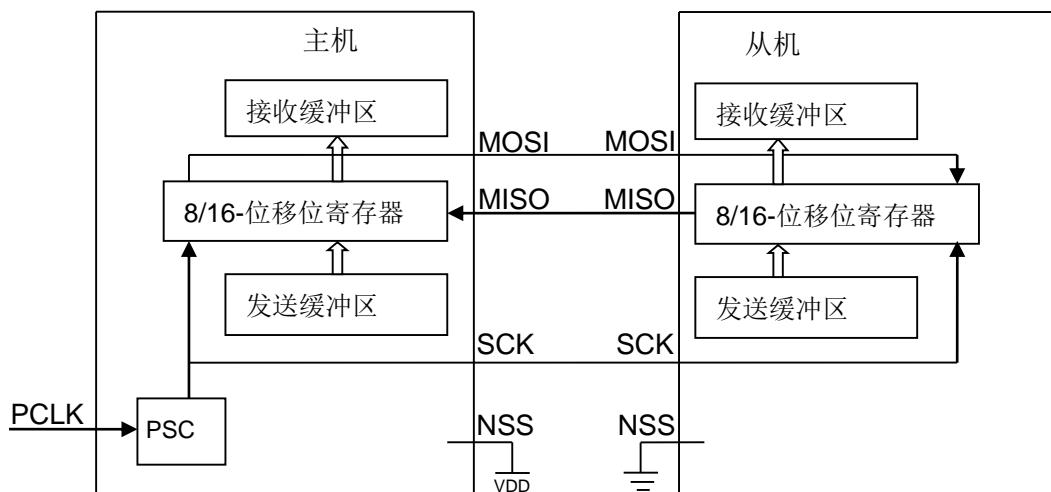
**MOSI**: 此引脚用于在从模式下的数据接收和主模式下的数据发送

**SCK**: SPI 串行时钟输出引脚，作为主设备的输出，从设备的输入

**NSS**: 在硬件模式下（SWNSSEN 清零），NSS 管脚可作为输入脚使用。主机模式下 NSS 管脚应拉高，从机模式下 NSS 管脚应拉低。假如在主机模式下将 NSS 拉低，则将产生错误，此时 CONFN 将会置位，硬件自动将 MSTMODE 清零。假如在从机模式下 NSS 为高电平，意味着该芯片未被选中，SPI 模块要等到 NSS 管脚被外部拉低后才能工作。在软件模式下（SWNSSEN 置位），SSI 位将替代 NSS 管脚的功能，NSS 可用作标准 I/O 口。此外，在主机模式下 NSS 可用作输出。若 NSSDRV 置位，当 SPI 通信开始后 NSS 管脚将拉低。

单主和单从模式内部典型连接图：

图 15-1. 单主/单从应用



主从设备的 MOSI、MISO、SCK 引脚都相互连接。数据通过 MOSI 线从主设备传输到从设备，通过 MISO 线从从设备传输到主设备。主设备产生时钟信号通过 SCK 线传输给从设备。在硬件模式(SWNSSEN 位被清 0)下，NSS 在主模式下被拉高，在从模式下被拉低；在软件模式(SWNSSEN 位被置 1)下，NSS 引脚不用。

主设备通过时钟控制通讯。当主设备通过 MOSI 引脚发送数据给从设备，它也会同时通过 SCK 引脚发送时钟给从设备。从设备通过 MOSI 引脚接收数据，然后根据时钟通过 MISO 发送数据。

通过设置 SPI\_CTR1 寄存器中的 SWNSSEN 位，NSS 既可以在软件模式下也可用在硬件模式下。

## ■ 硬件 NSS 模式 (**SWNSSEN = 0**)

取决于 SPI\_CTLR2 寄存器的 NSSDRV 位，NSS 引脚能被用作输入或者输出(仅在主模式下)。

- **NSS 输出使能 (SWNSSEN = 0, NSSDRV = 1)**

该配置仅能在主模式下使用。主设备启动通讯(开启发送时钟)的时候 NSS 信号被拉低，一直保持低电平直到 SPI 禁止。

- **NSS 输出禁止(SWNSSEN = 0, NSSDRV = 0)**

NSS 引脚用作输入。在主模式下，NSS 引脚应被拉高；在从模式下，NSS 引脚用作片选，当 NSS 低电平的时候从设备被选中，反之当 NSS 高电平的时候从设备未被选中。

## ■ 软件 NSS 模式(**SWNSSEN = 1**)

SPI\_CTLR1 寄存器中的 SWNSS 位代替了 NSS 的功能。在主模式下，SWNSS 位应该被置 1；在从模式下，从设备在 SWNSS 被清 0 的时候被选中，当 SWNSS 被置 1 的时候取消选中。在 SPI 通讯中 NSS 引脚不用。

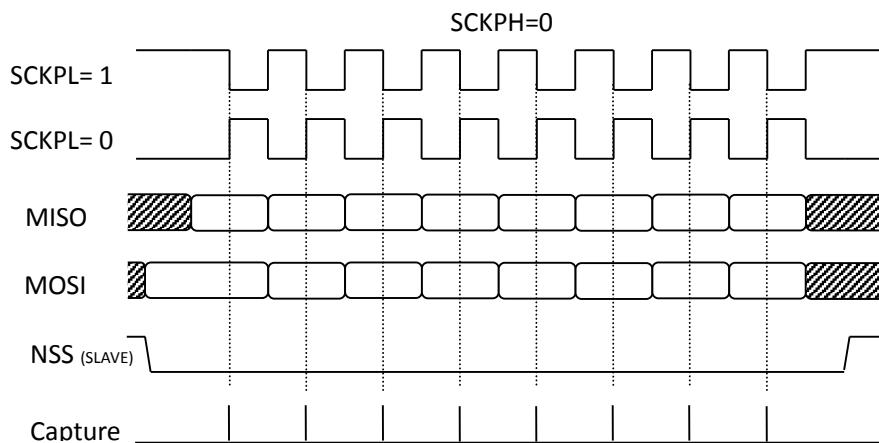
### 时钟相位和时钟极性

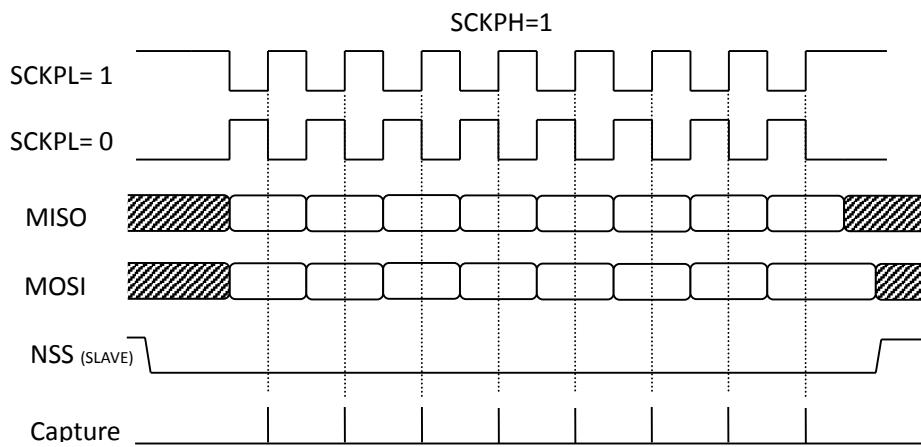
可以通过 SPI\_CTLR1 寄存器的 SCKPL 和 SCKPH 位用软件选择四种可能的时序关系。SCKPL 位控制着没有数据发送时的空闲状态的时钟输出电平。如果 SCKPL 被清 0，空闲时 SCK 引脚为低电平；如果 SCKPL 被置 1，SCK 引脚在空闲时间为高电平。

SCKPH 决定着数据的捕获时间。如果 SCKPH 被清 0，则 SPI 在 SCK 引脚上的第 1 个跳变沿采样数据；如果 SCKPH 被置 1，则 SPI 在 SCK 引脚上的第 2 个跳变沿采样数据。

下图显示了 SPI 传输四种组合的时序关系：

**图 15-2. SPI 数据时钟时序图**





### 数据帧格式

SPI\_CTRLR1 寄存器的 LF 位可以设置 MSB 在前或 LSB 在前。

取决于 SPI\_CTRLR1 寄存器的 LF 位的值, MSB(LF 被清 0)或 LSB(LF 被置 1)最初被发送。SPI\_CTRLR1 寄存器的 FF16 位决定了数据是 8 位还是 16 位。

#### 15.3.2. 引脚配置 (SPI 四路模式)

本节内容适用于 GD32F170xx 和 GD32F190xx 产品。

SPI 引脚默认配置为单路模式。当 SPI\_QWCR 寄存器中的 QMOD 位被置 1 时, 进入 SPI 四路模式(只适用于 SPI2)。在 SPI 四路模式下, SPI 使用六个引脚与外部设备连接:

**MOSI:** 该引脚在 SPI 四路模式读操作时接收数据,写操作时发送数据。

**MISO:** 该引脚在 SPI 四路模式读操作时接收数据,写操作时发送数据。

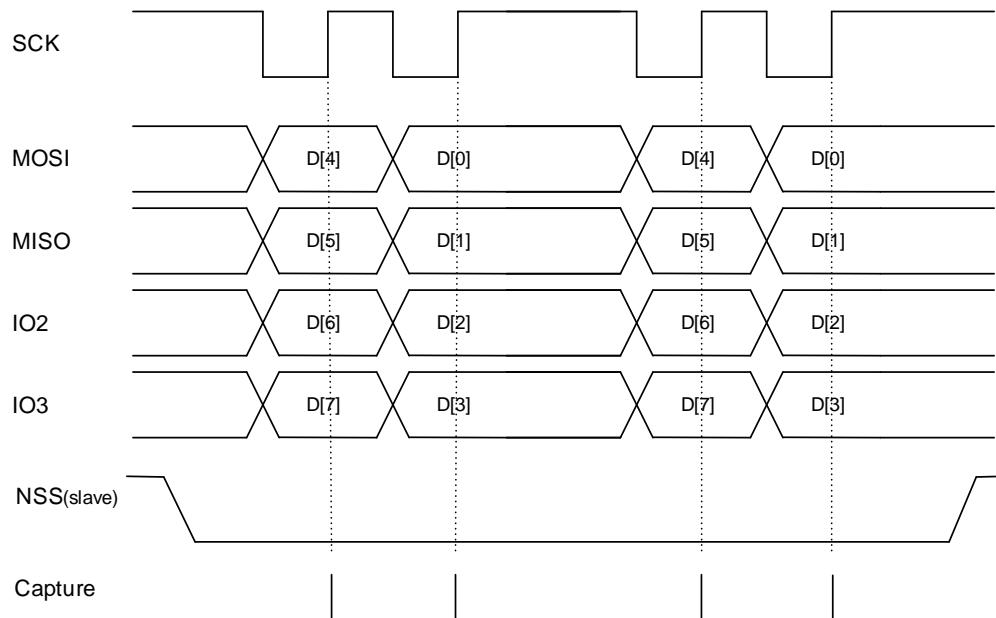
**IO2:** 该引脚在 SPI 四路模式读操作时接收数据,写操作时发送数据。

**IO3:** 该引脚在 SPI 四路模式读操作时接收数据,写操作时发送数据。

**SCK:** 在 SPI 四路模式下, 每个数据帧只有两个时钟周期, 其他的配置与 SPI 单路情况下相同。

**数据帧格式:** 在 SPI 四路模式下, 数据长度固定为 8 位。

图 15-3. SPI 四路模式下数据时序图 (SCKPL=1, SCKPH=1)



### 15.3.3. SPI 从模式

在从模式下，串行时钟由主设备提供，**SPI\_CTLR1** 寄存器的 PSC[2:0]位没有作用。通讯过程由主设备控制，在主设备发送时钟之前，从设备必须已经使能。

SPI 从设备配置步骤：

- 配置数据格式 (**SPI\_CTLR1** 寄存器的 FF16 位)。
- 配置时序关系(**SPI\_CTLR1** 寄存器中的 SCKPL 位和 SCKPH 位)。这两位必须和主设备配置一致。
- 配置帧格式 (**SPI\_CTLR1** 寄存器的 LF 位)，必须和主设备配置一致。
- 配置 NSS 模式 (**SPI\_CTLR1** 寄存器的 SWNSSEN 位)。硬件模式下 (SWNSSEN=0)，发送过程中 NSS 引脚必须拉低。软件模式中(SWNSSEN=1)，**SPI\_CTLR1** 寄存器中的 SWNSS 位在发送过程中必须被清 0。
- 设置从模式 (清 MSTMODE 位)。使能 SPI (**SPIEN** 位置 1)。

#### 发送顺序

当从设备通过 SCK 引脚接收到时钟的时候从设备的发送开始，首先将最高位或最低位放在 MOSI 引脚，其他位从发送缓冲区加载到移位寄存器。TBE 位被置 1，软件根据需要写第二个数据到发送缓冲区。硬件根据接收到的时钟发送移位寄存器中的数据位。

#### 接收顺序

在最后一次采样时钟沿后，数据发送结束，移位寄存器中的数据会被复制到接收缓冲区，

SPI\_STR 寄存器中的 RBNE 位被置 1。读 SPI\_DTR 寄存器会返回接收缓冲区的数据，同时 RBNE 位被清 0。

### 15.3.4. SPI 主模式

主模式配置中，串行时钟由 SPI 产生，从 SCK 引脚发出。

主模式下，PCLK(SPI1 是 PCLK2 时钟或 SPI2 是 PCLK1 时钟)产生串行时钟。PSC[2:0]位决定了传输的波特率。

SPI 主模式配置过程：

- 配置 PSC[2:0]位确定波特率
- 配置数据格式(SPI\_CTLR1 寄存器中的 FF16 位)
- 配置时序关系(SPI\_CTLR1 寄存器的 SCKPL 和 SCKPH 位)。必须和从设备配置相同。
- 配置帧格式(SPI\_CTLR1 寄存器 LF 位)，必须和从设备配置相同。
- 配置 NSS 模式 (SPI\_CTLR1 寄存器 SWNSSEN 位)。如果 NSS 引脚用作输入，在硬件模式下(SWNSSEN=0)，NSS 必须被拉高。在软件模式下 (SWNSSEN=1)，SPI\_CTLR1 寄存器 SWNSS 位必须被置 1。如果 NSS 引脚用作输出，NSSDRV 位必须被置 1，NSS 引脚在发生开始的时候将被拉低。
- 设置主模式(MSTMODE 位置 1)，使能 SPI (SPIEN 位置 1)。

#### 发送顺序

当数据写入发送缓冲后，主设备开始发送，首先将最低位或最高位发送到 MOSI 管脚上，剩余各位将从发送缓冲加载到移位寄存器中。当数据加载完成后，TBE 置位。后续数据被放入发送缓冲区将会接着被发送。当 TBE 位被置 1 时写 DTR 寄存器。

#### 接收顺序

在最后一次采样时钟沿后，数据发送结束，移位寄存器中的数据被复制到接收缓冲区，SPI\_STR 寄存器中的 RBNE 位被置 1。读 SPI\_DTR 寄存器会返回接收缓冲区的数据，RBNE 位被清 0。

### 15.3.5. SPI 四路主模式（仅适用于 GD32F170xx 和 GD32F190xx 产品）

SPI 四路模式用于控制四路 SPI Flash 外设，包括但不限于 GD25QXX 串行 flash 芯片。

要配置成 SPI 四路模式，首先确认 TBE 位置 1 并且 TRANS 位清 0，然后将 SPI\_QWCR 寄存器中的 QMOD 位置 1。在 SPI 四路模式下，SPI\_CTLR1 寄存器中的标志位 BDM、BDOE、CRCEN、CRCNT、FF16、RO 和 LF 保持清 0，并且 MSTMODE 置 1 保证 SPI 作为主设备。SPIEN、PSC、SCKPL 和 SCKPH 根据需要配置。

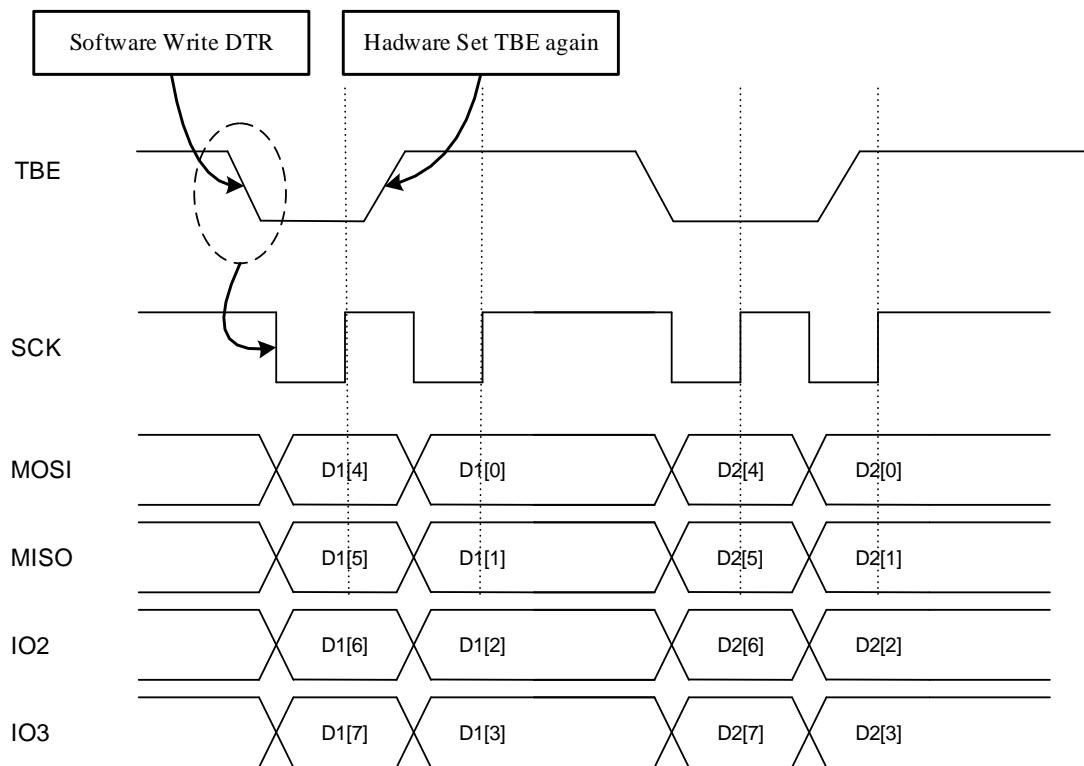
在 SPI 四路模式下有读操作和写操作，取决于 SPI\_QWCR 寄存器中的 QRD 位。

## 四路写操作

当 SPI\_QWCR 寄存器中 QMOD 位被置 1 并且 QRD 位清 0 时，SPI 进行写操作。写操作时 MOSI、MISO、IO2、IO3 引脚用于输出。在 SCK 产生时钟信号后，一旦 TBE 位清 0 并且 SPIEN 位置 1，通过四个引脚发送数据。开始数据传输后，每发送一个数据帧检测 SPIEN 和 TBE 标志位，若不能满足条件停止传输。

四路模式下发送操作流程：

1. 根据应用需求，配置 SPI\_STLR1 和 SPI\_CTLR2 中的时钟预分频、时钟极性、相位等。
2. 设置 SPI\_QWCR 寄存器中的 QMOD，然后通过设置 SPI\_CTLR1 中的 SPIEN=1 来使能 SPI。
3. 往 SPI\_DTR 中写入一个字节的数据，TBE 标志位会被清除。
4. 等待 TBE 由硬件再次置位，然后写下下一个字节。

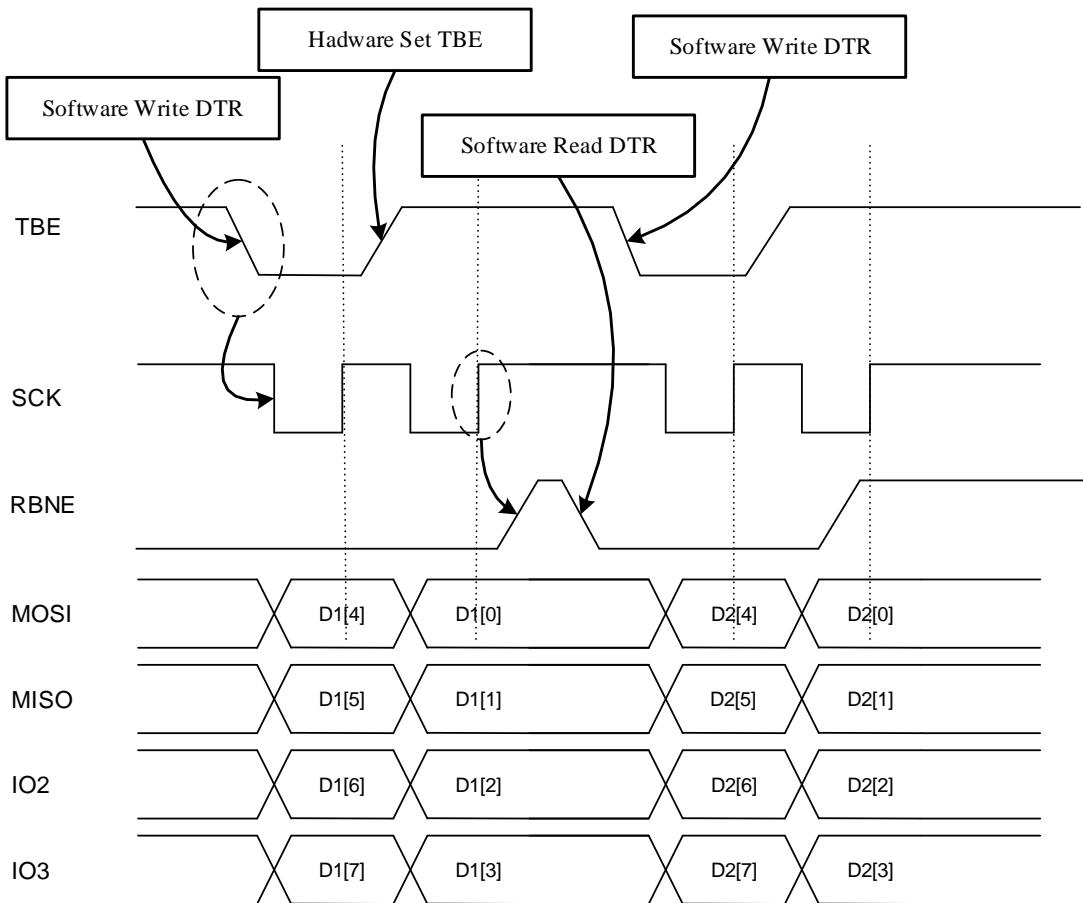


## 四路读操作

当 SPI\_QWCR 寄存器中 QMOD 位和 QRD 位被置 1 时，SPI 进行读操作。读操作时 MOSI、MISO、IO2、IO3 引脚用于输入。TBE 位清 0 时，发送数据产生 SCK 时钟信号，并且 SPIEN 位置 1，通过四个引脚接收数据。开始数据传输后，每发送一个数据帧检测 SPIEN 和 TBE 标志位，若不能满足条件停止传输。所以，需要一直向 DTR 中写空闲数据，保证 SPI 产生 SCK 并接收数据。

四路模式下接收操作流程：

1. 根据应用需求，配置 SPI\_STLR1 和 SPI\_CTLR2 中的时钟预分频、时钟极性、相位等。
2. 设置 SPI\_QWCR 寄存器中的 QMOD 和 QRD，然后通过设置 SPI\_CTLR1 中的 SPIEN=1 来使能 SPI。
3. 往 SPI\_DTR 中随意写入一个字节的数据（例如 0xFF）。
4. 等待 RBNE 标志位置位，通过读 SPI\_DTR 来获取接收的字节
5. 为了接收下一个字节，软件需往 SPI\_DTR 随意写入一个字节的数据（例如 0xFF）



#### 离开 SPI 四路模式或者关闭 SPI

在离开 SPI 四路模式或者关闭 SPI 时，需要先确保 TBE 位置 1，并且 TRANS 位清 0，然后清除 SPI\_QWCR 寄存器中的 QMOD 位离开 SPI 四路模式或者清除 SPI\_CTLR1 寄存器中的 SPIEN 位关闭 SPI。

#### 15.3.6. SPI 单工通信

有两种配置可以使 SPI 工作在单工模式。

将 SPI\_CTLR1 寄存器的 BDM 位置 1。时钟从主设备的 SCK 引脚传输到从设备的 SCK 引脚。数据从主设备的 MOSI 引脚发送到从设备的 MISO 引脚。数据传输方向由 SPI\_CTLR1 寄存器

的 BDOE 位决定。BDOE 位在主从设备中相反。如果 BDOE 在主设备中被置 1 从设备中被清 0，数据从主设备发送到从设备，反之数据从从设备发送到主设备。

将 SPI\_CTLR1 寄存器的 BDM 位置 0。此时 RO 位在主设备和从设备中必须配置相反：如果 RO 位在主设备中被清 0 从设备中被置 1，数据从主设备发送到从设备，主设备的 MOSI 引脚发送数据且从设备的 MOSI 接收数据，MISO 引脚未被使用。如果 RO 位在主设备中被置 1 从设备中被清 0，数据从从设备发送到主设备，从设备的 MISO 引脚发送数据且主设备的 MISO 引脚接收数据，MOSI 引脚未被使用。但是，时钟总是由主设备产生通过 SCK 引脚传输给从设备。

在单工通信过程中，如果主设备配置为只接收模式且从设备配置为只发送模式，当 SPI 被使能时主设备自动接收数据。从设备必须在主设备使能 SPI 之前已经准备好发送，并且应当在限定时间内将数据写入发送缓冲。主设备收到倒数第二个数据之后应禁用 SPI（将 SPIEN 清零）；此时最后一个数据正在传输中，SPI 模块会在接收完最后一个数据后自动禁用。

### 15.3.7. 数据接收和发送流程

每个 SPI 模式都有两个缓冲区：发送缓冲区和接收缓冲区。写 SPI\_DTR 寄存器将会把数据保存在发送缓冲区，读 SPI\_DTR 寄存器将会返回接收缓冲区的数据。

上一帧数据发送结束，发送缓存区的数据会被复制到移位寄存器，TBE 位被硬件置 1，然后软件可以在必要的时候通过写 SPI\_DTR 寄存器把下一个数据写入发送缓冲区。如果 SPI\_CTLR2 寄存器的 TBEIE 位被置 1，在 TBE 位被置 1 的时候将产生中断。写 SPI\_DTR 寄存器可以清除 TBE 位。当 TBE 位被清 0 的时候写 SPI\_DTR 寄存器将会覆盖掉存储在发送缓冲区的数据。

当数据的最后一位在采样时钟沿上被采集完成后，数据将被接收并存储在移位寄存器中，硬件会将移位寄存器中的数据复制到接收缓冲区并将 RBNE 位置 1，以备软件读取。如果 SPI\_CTLR2 寄存器的 TBEIE 位被置 1，在 RBNE 位被置 1 的时候将产生中断。读 SPI\_DTR 寄存器可以清除 RBNE 位。如果在 RBNE 被置 1 的时候一个数据被接收(最后的数据没有被读取)，RXORE 位被置 1 来指示此错误。

### 15.3.8. CRC 校验

CRC 校验能够提高通信的可靠性。SPI 模块中包含两个 CRC 计算单元，分别用于发送数据和接收数据。CRC 计算单元能够逐位计算 CRC 值。计算所用到的多项式是可配置的，保存在 SPI\_CPR 寄存器中。

设置 SPI\_CTLR1 寄存器中的 CRCEN 位使能 CRC 计算。在全双工模式或只发送模式下，软件应该在最后一个数据被写入 SPI\_DTR 寄存器后立即将 CRCNT 位置 1。最后一个数据以后 SPI\_TCR 中的值将被发送。如果数据通过 DMA 方式发送，SPI\_TCR 中的数据将自动被硬件发送，CRCNT 不用配置。当 SPI 正在传输 SPI\_TCR 中的值时，会把此时接收到的数据视为 CRC 值，如果此值和 SPI\_RCR 的值不同则计算器将被关闭，错误位 CRCE 位将被置 1。

在只接收模式下，倒数第二个数据被接收后(此时正在接收最后一个数据)软件应该立即写 CRCNT 位，那么在接收完最后一个数据之后就会继续接收数据的 CRC 值，SPI 将此值和 SPI\_RCR 寄存器中的值进行比较，如果不同错误位 CRCE 将被置 1。

只有在发送缓冲区空的时候才可以发送 CRC 数据。在 CRC 数据发送期间 CRC 计算器被关闭。

使用 CRC 的 SPI 通信：

- 配置 SCKPL, SCKPH, LF, PSC, SWNSSEN, SWNSS 和 MSTMODE 的值。
- 配置 SPI\_CPR 寄存器中的多项式。
- 将 SPI\_CTLR1 寄存器的 CRCEN 位置 1 使能 CRC 计算。
- 将 SPI\_CTLR1 寄存器的 SPIEN 位置 1 使能 SPI。
- 开启通信。
- 在全双工或只发送模式下，在最后一个数据被写入 SPI\_DTR 后立即将 CRCNT 位置 1。在只接收模式下，在倒数第二个数据接收完毕后将 CRCNT 位置 1。CRC 传输的时候 CRC 计算器被关闭。
- 最后一个数据传输结束后 SPI 将发送 CRC 值，并把接收到的 CRC 值和 SPI\_RCR 中的值比较，如果不同 CRCE 位被置 1。
- 从模式下如果 CRCEN 位被置 1，即使 NSS 引脚被拉高 CRC 计算器依然会继续工作。主机侧和从设备侧的 CRC 值必须及时清除，以确保主设备和从设备的 CRC 计算能重新同步。SPIEN 位或者 CRCEN 位被清 0 的时候 CRC 值 (SPI\_RCR 和 SPI\_TCR) 被清除。

### 15.3.9. 状态标志和错误标志

#### 状态标志

##### 发送缓冲区空标志(TBE)

发送缓冲区空闲的时候该位被置 1，软件通过写 SPI\_DTR 寄存器将下一帧数据写入发送缓冲区。如果 TBEIE 位被置 1，TBE 被置 1 的时候产生中断。写 SPI\_DTR 寄存器会清除 TBE 位。

##### 接收缓冲区非空标志(RBNE)

接收缓冲区非空的时候该位被置 1，一帧数据被接收并存储在接收缓冲区里，软件可以通过读 SPI\_DTR 寄存器来读该数据。如果 RBEIE 位被置 1，RBNE 被置 1 的时候产生中断。读 SPI\_DTR 寄存器会清除 RBNE 位。

##### SPI 传输忙标志(TRANS)

TRANS 标志被硬件置 1 和清 0。它表示 SPI 正处于数据传输过程中。如果软件想要禁用 SPI，那么可通过检测 TRANS 标志判断传输是否结束，只有这样才能避免破坏最后一个数据的传输。因此，软件必须严格按照下述步骤操作。

除了主设备的双向只接收模式 (MSTMODE=1 和 BDM=1 和 BDOE=0)之外，发送开始的时候 TRANS 位会被置 1。当 SPI 关闭或者配置错误发生(CONFE=1)后 TRANS 位会被清 0。不连续通讯的时候，在每次通信的时候 TRANS 位为 0；连续通讯的时候，在从模式下 TRANS 位在每次通信之间为 0，在主模式下的通讯中 TRANS 位始终为 1。

## 错误标志

### 配置错误标志(CONFE)

NSS 硬件模式中，若 NSSDRV 未被使能，则当 NSS 引脚被拉低，CONFE 位被置 1。在 NSS 软件模式中，当 SWNSS 位为 0 时 CONFE 位被置 1。当 CONFE 被置 1 时，如果 ERRIE 为 1，将产生中断；于此同时硬件将 SPIEN 位和 MSTMODE 位清 0，SPI 被关闭，设备强制转换成从模式。

可以使用下面的软件序列来清除 CONFE 位：

- 对 SPI\_STR 寄存器读或写访问
- 写 SPI\_CTLR1 寄存器
- SPIEN 和 MSTMODE 位被写保护直到 CONFE 被清 0。从设备的 CONFE 位不能被置 1。  
在多主机的配置中，设备可以在 CONFE 位被置 1 的时候进入从模式，这也意味着发生了系统控制的多主冲突。

### 接收过载错误(RXORE)

在 RBNE 位被置 1 时，如果再有数据被接收，RXORE 位将被置 1。这意味着，上一帧数据还没有被读出来，新的数据已经接收了。接收缓冲区的内容将会被新接收到的数据覆盖，上一帧数据将会丢失。如果 ERRIE 位被置 1，当 RXORE 位被置 1 时将产生中断。

可以使用如下的软件序列来清除 RXORE 位：

读 SPI\_DTR 寄存器，然后读 SPI\_STR 寄存器。

### CRC 错误(CRCE)

当 CRCEN 位被置 1 时，SPI\_RCR 寄存器中的接收到的数据的 CRC 计算值将会和紧随着最后一帧数据接收到的 CRC 的值比较，当二者不同时 CRCE 位被置 1，对 CRCE 位写 0 可以清除该位，写 1 到 CRCE 位无效。

## 15.3.10. 停止 SPI

每当一次传输结束，软件通过清除 SPIEN 位来停止 SPI。在一些配置中，SPIEN 被清 0 以后最后一次传输还在进行。为了避免破坏最后的传输，软件应该遵守以下步骤：

### 全双工模式

等待 RBNE=1 以接收最后的数据  
等待 TBE=1  
等待 TRANS=0  
禁用 SPI (SPIEN=0)，进入深度睡眠模式或关闭外设时钟。

### 只发送模式

最后的数据被写入 SPI\_DTR 寄存器  
等待 TBE=1  
等待 TRANS=0  
禁用 SPI (SPIEN=0)，进入深度睡眠模式或关闭外设时钟。

### 主设备的只接收模式

等待倒数第二次RBNE=1

在禁用 SPI(SPIEN=0)之前等待一个SPI时钟周期(使用软件循环)

在进入深度睡眠模式或关闭外设时钟之前等待最后的RBNE=1

### 从设备的只接收模式

可以在任何时候禁用 SPI (写SPIEN=1)，当前传输结束后SPI将被有效的禁止。

等待TRANS = 0之后才能进入深度睡眠模式或关闭外设时钟。

## 15.3.11. DMA 请求

采用 DMA 来发送或接收数据，才能真正发挥出 SPI 的最高速度。此时读写 SPI\_DTR 的速度足够快，且发送的数据也相当紧凑。

当 SPI\_CTLR2 寄存器的 DMATE 或 DMARE 位被置 1 时，即可进行 DMA 访问。每当 TBE 被置 1，SPI 就会通过发送通道发出一个 DMA 请求。DMA 写一帧数据到 SPI\_DTR 寄存器后 TBE 位将被清 0。当 RBNE 位被置 1，SPI 通过接收通道将会发出一个 DMA 请求。DMA 从 SPI\_DTR 寄存器中读取一帧数据以后 RBNE 位将被清 0。

当 SPI 仅用来发送数据，可以只使能 SPI 的发送 DMA 通道。在这种情况下，RXORE 标志位被置 1 因为接收到的数据没有被读取。

当 SPI 仅用来接收数据，可以只使能 SPI 的接收 DMA 通道。

在发送模式下，当 DMA 已经写入所有将被发送的数据(DMA\_IFR 寄存器的 TCIF 标志位被置 1)，为确保 SPI 通讯结束，软件需要监视 TRANS 标志位来判断通信是否完成，这样做是为了在禁用 SPI 或进入深度睡眠模式之前避免破坏最后一次传输。软件必须首先等到最后一个 TBE=1 然后再等到 TRANS=0。

图 15-4. 使用 DMA 发送

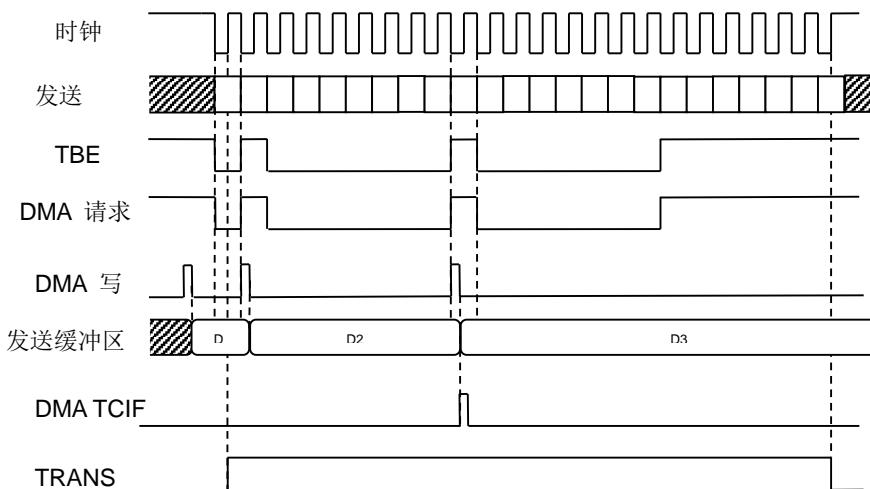
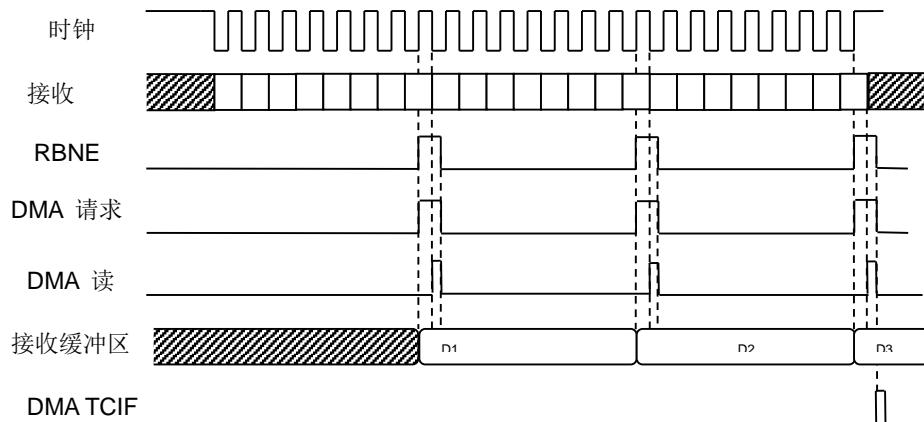


图 15-5. 使用 DMA 接收



在 SPI 通讯中使用 DMA 时，如果 CRCEN 位被置 1，则 SPI 模块会在最后一个数据之后自动发送和接收 CRC 值。此时 CRCNT 控制位将失去意义。接收缓冲中的 CRC 值应当及时读出，以便清零 RBNE 位。

### 15.3.12. SPI 中断

表 15-1. SPI 中断请求

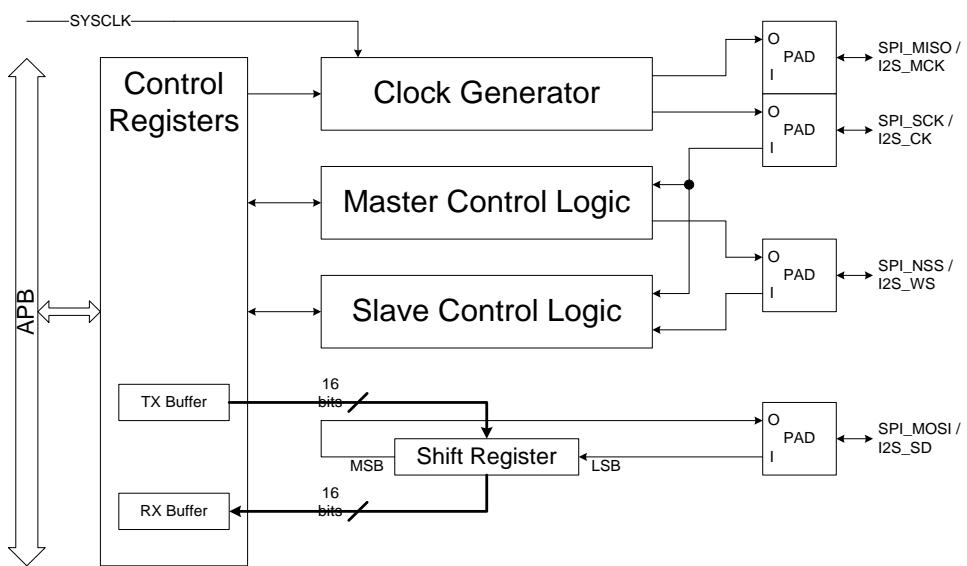
中断事件	标志位	使能控制位
发送缓冲区空	TBE	TBEIE
接收缓冲区有数据	RBNE	RBNEIE
配置出错	CONFE	ERRIE
接收过载错误	RXORE	
CRC错误	CRCE	

## 15.4. I2S 功能描述

### 15.4.1. 一般描述

I2S 框图如下所示。

图 15-6. I2S 框图



I2S 与 SPI 共享相同的引脚、标志位、中断、数据缓冲区和移位寄存器。当 SPI\_I2SCTL 寄存器的 I2SSEL 位被置 1 时，这些资源被 I2S 占用。否则，这些资源被 SPI 占用。

I2S 接口有 4 个引脚，分别是 I2S\_CK、I2S\_WS、I2S\_SD 和 I2S\_MCK。I2S\_CK 是串行时钟信号，与 SPI\_SCK 共享一个引脚。I2S\_WS 是数据控制信号，与 SPI\_NSS 共享一个引脚。I2S\_SD 是串行数据信号，与 SPI\_MOSI 共享一个引脚。I2S\_MCK 是主时钟信号，与 SPI\_MISO 共享一个引脚。I2S\_MCK 对于 I2S 接口而言是个可选信号，它提供了一个 256 倍于 Fs 的时钟频率，其中 Fs 是音频采样率。

I2S 功能有 5 个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来产生 I2S 通讯时钟。主机控制逻辑用来在主机模式下生成 I2S\_WS 信号并控制通信。从机控制逻辑根据接收到的 I2SCK 和 I2S\_WS 来控制从机模式下的通信。移位寄存器控制 I2S\_SD 上的串行数据发送和接收。

#### 15.4.2. 支持的音频标准

I2S 音频标准是通过设置 SPI\_I2SCTL 寄存器的 I2SSTD 位来选择的。有四种音频标准可供选择，分别是 I2S 飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准。除 PCM 之外的所有标准都是两个通道(左通道和右通道)的音频数据分时复用 I2S 接口的，并通过 I2S\_WS 信号来区分当前数据属于哪个通道。对于 PCM 标准来说，I2S\_WS 信号表示帧同步信息。

数据长度和通道长度可以通过 SPI\_I2SCTL 寄存器的 DTLEN 和 CHLEN 位来设置。由于通道长度必须大于等于数据长度，所以有四种数据长度和通道长度的组合可供选择。它们是 16 位数据打包成 16 位数据帧，16 位数据打包成 32 位数据帧，24 位数据打包成 32 位数据帧，32 位数据打包成 32 位数据帧。用于发送和接收的数据缓冲区都是 16 位宽的。所以，要完成数据长度为 24 位或 32 位的数据帧，SPI\_DTR 寄存器需要被访问 2 次；要完成数据长度为 16 位的数据帧，SPI\_DTR 寄存器只需被访问 1 次。如需将 16 位数据打包成 32 位数据帧，硬件会自动插入 16 位 0。

对于所有标准和帧格式来说，数据的最高位总是最先被发送的。对于所有基于两通道分时复用的标准来说，左通道总是先发送的，然后是右通道。

### I2S 飞利浦标准

对于 I2S 飞利浦标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的下降沿变化。各种配置情况的时序图如下所示。

图 15-7. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

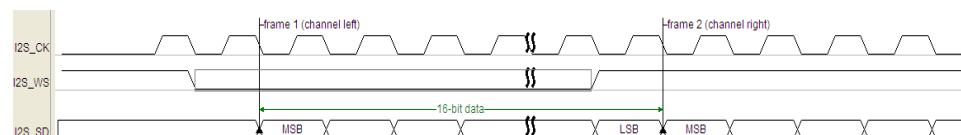
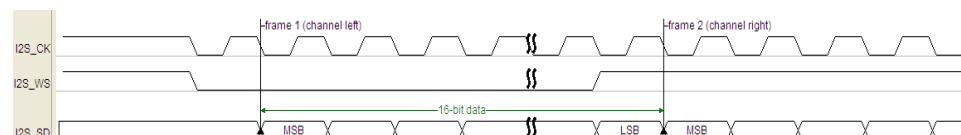


图 15-8. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

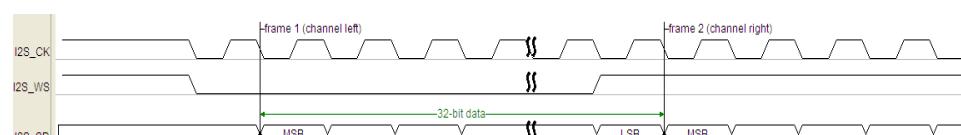


对于 16 位数据打包成 16 位数据帧的帧格式来说，每完成 1 帧需要访问 SPI\_DTR 寄存器 1 次。

图 15-9. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)



图 15-10. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)



对于 32 位数据打包成 32 位数据帧的帧格式来说，每完成 1 帧需要访问 SPI\_DTR 寄存器 2 次。在发送模式下，如要发送 0x8899AABB，写入 SPI\_DTR 寄存器的第一个数据应该是 0x8899，第二个数据应该是 0xAABB。在接收模式下，如要接收 0x8899AABB，第一个从 SPI\_DTR 寄存器读到的数据应该是 0x8899，第二个数据应该是 0xAABB。

图 15-11. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

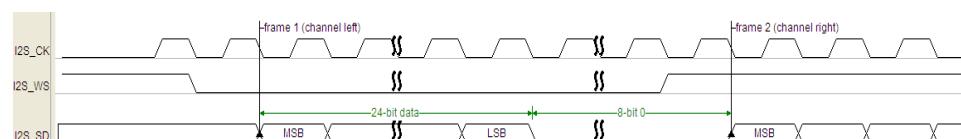
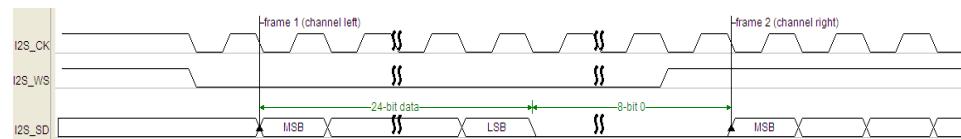


图 15-12. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



对于 24 位数据打包成 32 位数据帧的帧格式来说，每完成 1 帧需要访问 SPI\_DTR 寄存器 2 次。在发送模式下，如要发送 0x8899AA，写入 SPI\_DTR 寄存器的第一个数据应该是 0x8899，第二个数据应该是 0xAAXX（低 8 位可以是任意值，它将被硬件强制变为 0x00）。在接收模式下，如要接收 0x8899AA，第一个从 SPI\_DTR 寄存器读到的数据应该是 0x8899，第二个数据应该是 0xAA00。

图 15-13. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

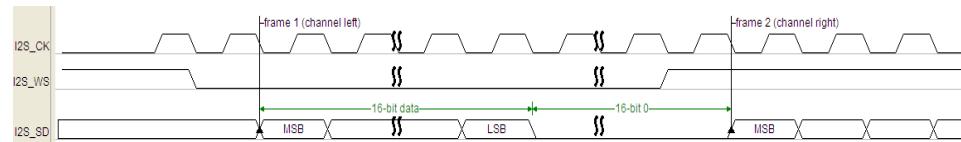
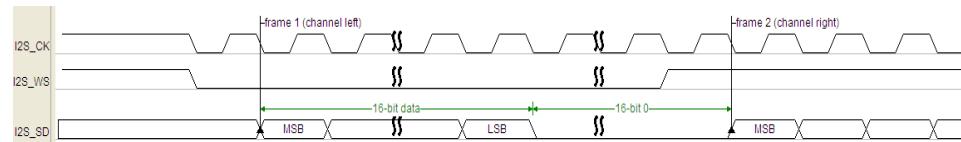


图 15-14. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



对于 16 位数据打包成 32 位数据帧的帧格式来说，每完成 1 帧需要访问 SPI\_DTR 寄存器 1 次。为扩展成 32 位的数据帧格式，剩下的 16 位被硬件强制填为 0x0000。

### MSB 对齐标准

对于 MSB 对齐标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的下降沿变化。SPI\_DTR 寄存器的控制方式与 I2S 飞利浦标准完全相同。各个配置的时序图如下。

图 15-15. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

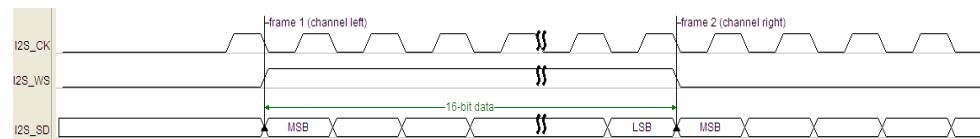


图 15-16. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

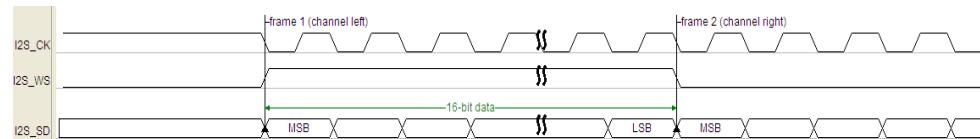


图 15-17. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)



图 15-18. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)

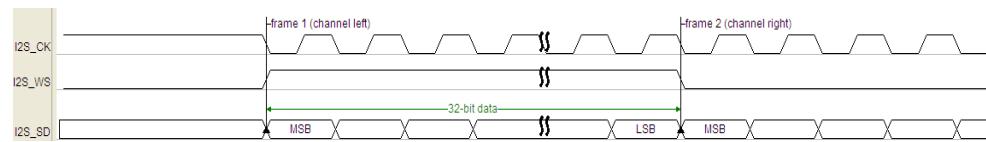


图 15-19. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

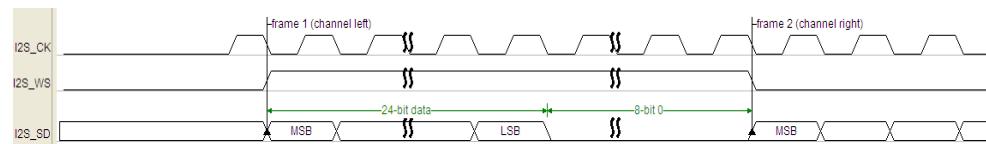


图 15-20. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)

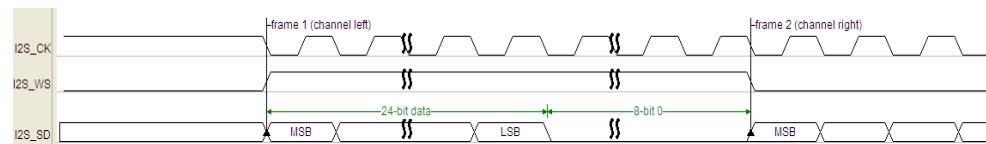


图 15-21. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

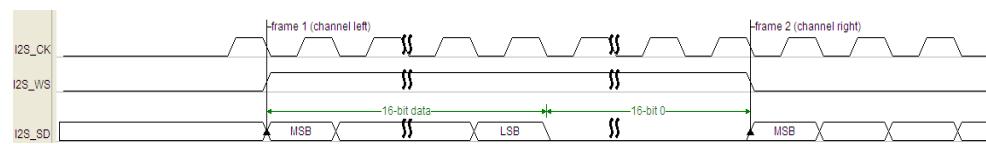
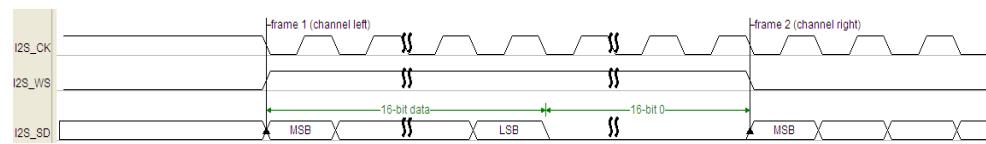


图 15-22. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



### LSB 对齐标准

对于 LSB 对齐标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的下降沿变化。在通道长度与数据长度相同的情况下，LSB 对齐标准和 MSB 对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB 对齐标准的有效数据与最低位对齐，而 MSB 对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 15-23. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

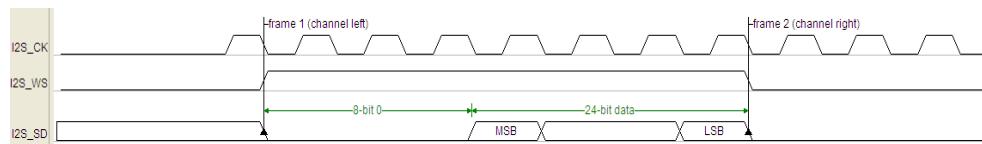
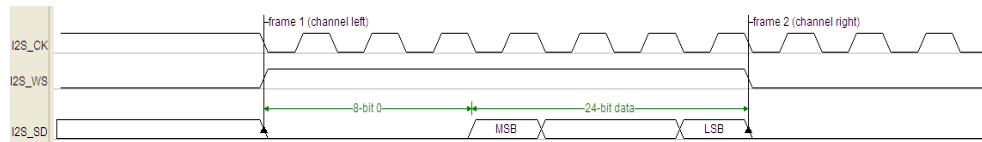


图 15-24. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



对于 24 位数据打包成 32 位数据帧的帧格式来说，每完成 1 帧需要访问 SPI\_DTR 寄存器 2 次。在发送模式下，如要发送 0x8899AA，写入 SPI\_DTR 寄存器的第一个数据应该是 0xXX88 (高 8 位可以是任意值，它将被硬件强制变为 0x00)，第二个数据应该是 0x99AA。在接收模式下，如要接收 0x8899AA，第一个从 SPI\_DTR 寄存器读到的数据应该是 0x0088，第二个数据应该是 0x99AA。

图 15-25. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

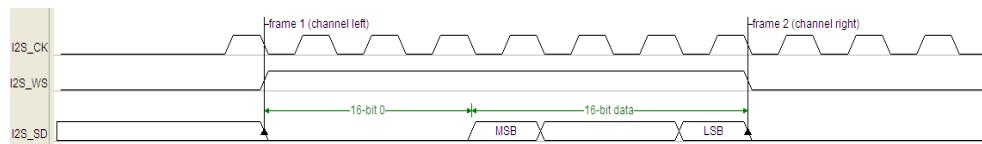
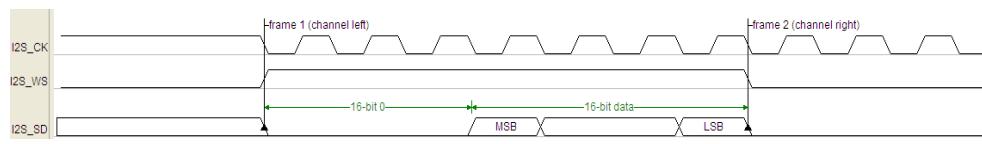


图 15-26. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)

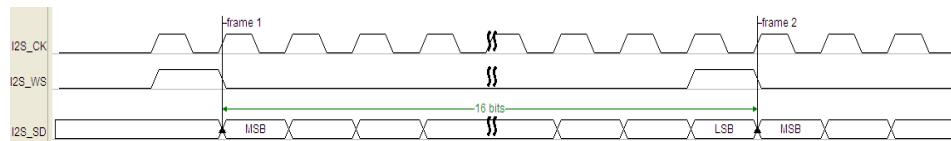


对于 16 位数据打包成 32 位数据帧的帧格式来说，每完成 1 帧需要访问 SPI\_DTR 寄存器 1 次。为扩展成 32 位的数据帧格式，剩下的 16 位被硬件强制填为 0x0000。

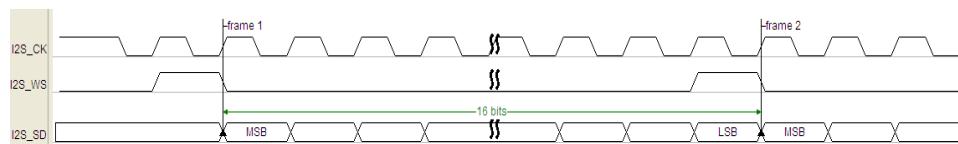
## PCM 标准

对于 PCM 标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的上升沿变化，I2S\_WS 信号表示帧同步信息。短帧同步模式和长帧同步模式都支持，可通过 SPI\_I2SCTLR 寄存器的 PCMSM 位来选择。SPI\_DTR 寄存器的控制方式与 I2S 飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

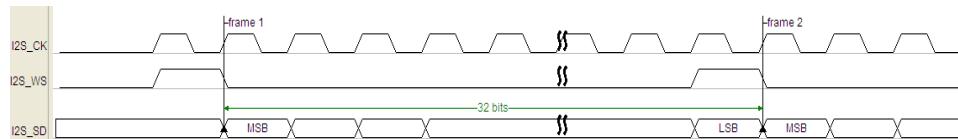
图 15-27. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)



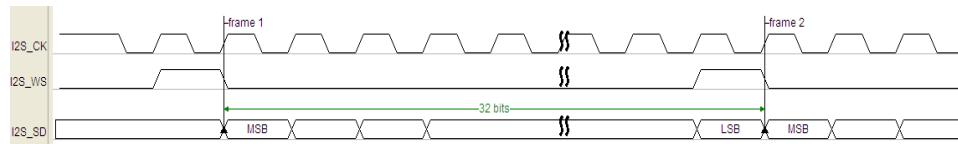
**图 15-28. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)**



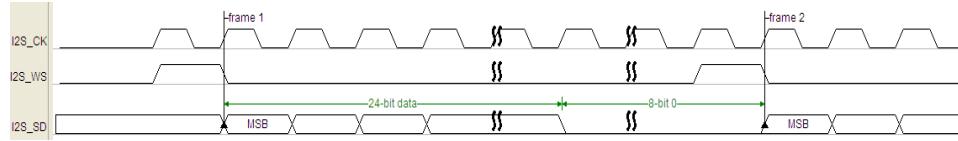
**图 15-29. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)**



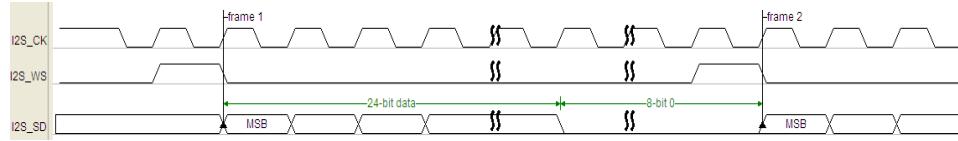
**图 15-30. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)**



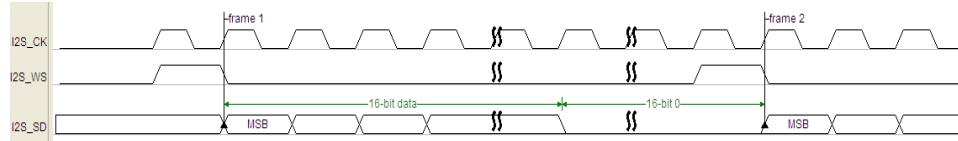
**图 15-31. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)**



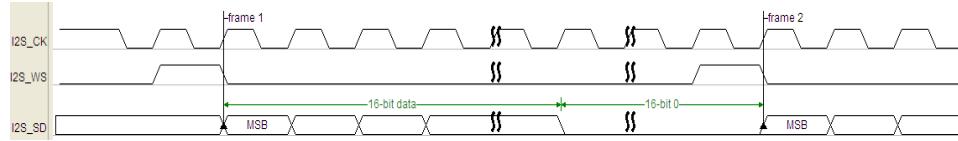
**图 15-32. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)**



**图 15-33. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)**



**图 15-34. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)**



长帧同步模式的各种配置情况时序图如下所示。

图 15-35. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

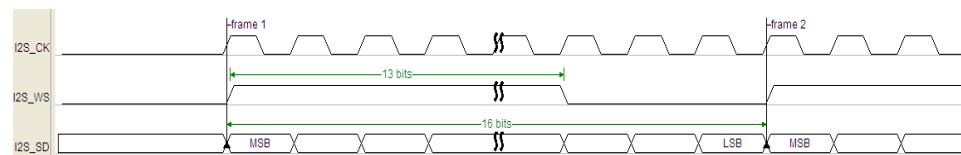


图 15-36. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

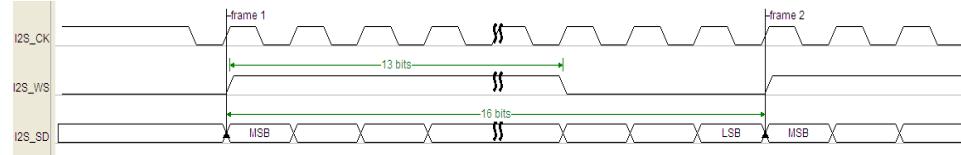


图 15-37. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

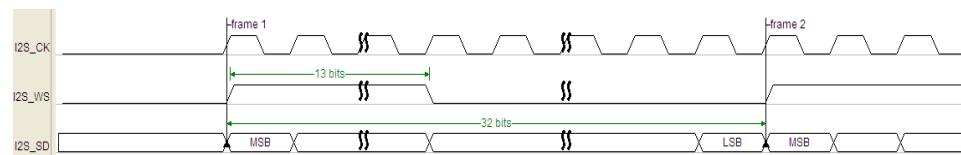


图 15-38. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

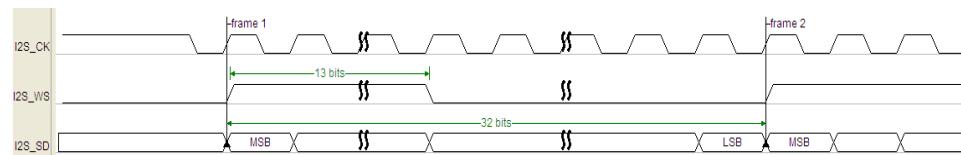


图 15-39. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

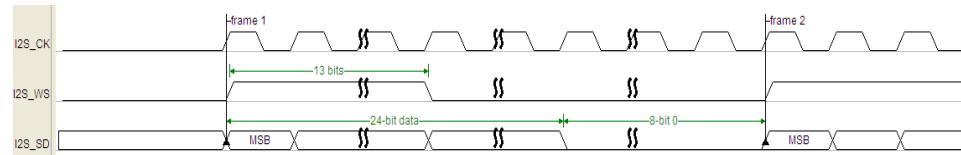


图 15-40. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

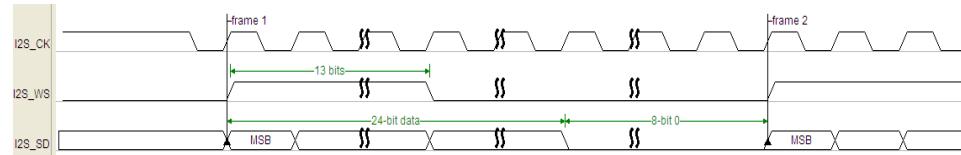


图 15-41. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

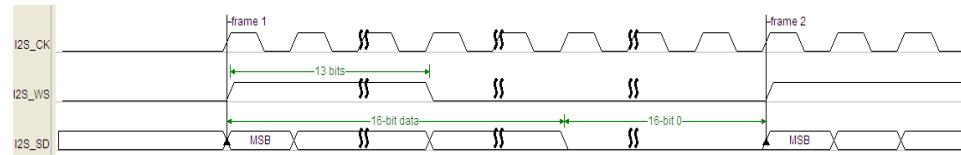
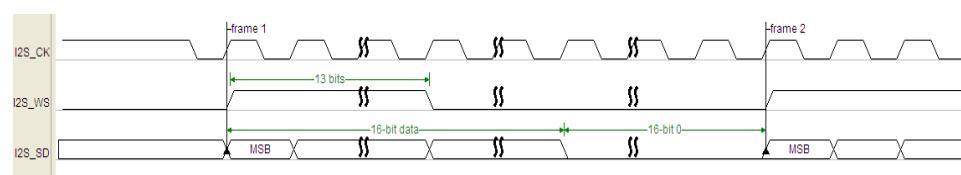
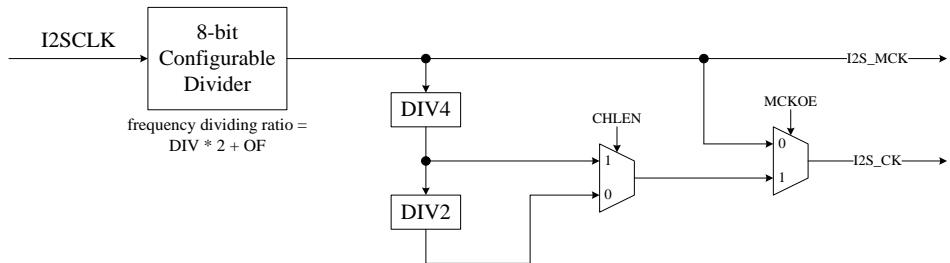


图 15-42. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



### 15.4.3. 时钟生成器

图 15-43. I2S 时钟生成器框图



I2S 时钟生成器框图如上所示。I2S 接口时钟是通过 SPI\_I2SCKP 寄存器的 DIV 位, OF 位, MCKOE 位以及 SPI\_I2SCTLR 寄存器的 CHLEN 位来配置的。I2SCLK 选择 SYSCLK。I2S 比特率可以通过下表所示的公式计算。

表 15-2. 比特率计算公式

MCKOE	CHLEN	公式
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

音频采样率(Fs)和 I2S 比特率的关系按如下公式定义。

$$Fs = I2S \text{ 比特率} / (\text{通道长度} * \text{通道数})$$

所以, 为了得到期望的音频采样率, 时钟生成器需要按下表所列的公式进行配置。

表 15-3. 音频采样率计算公式

MCKOE	CHLEN	公式
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

常用音频采样率的配置方法和能达到的精度在下表中列出。表中的精度是在使用标准 8MHz HSE 时钟的情况下计算得到的。

表 15-4. 音频采样率配置和精度

Target Fs(Hz)	SYSCLK (MHz)	MCKOE	CHLEN = 0				CHLEN = 1			
			DIV	OF	Real Fs(Hz)	Error	DIV	OF	Real Fs(Hz)	Error
96000	72	No	11	1	97826.09	1.90%	6	0	93750	2.34%
96000	72	Yes	1	1	93750	2.34%	1	1	93750	2.34%
96000	48	No	8	0	93750	2.34%	4	0	93750	2.34%
96000	48	Yes	1	0	93750	2.34%	1	0	93750	2.34%
48000	72	No	23	1	47872.34	0.27%	11	1	48913.04	1.90%

Target Fs(Hz)	SYSCLK (MHz)	MCKOE	CHLEN = 0				CHLEN = 1			
			DIV	OF	Real Fs(Hz)	Error	DIV	OF	Real Fs(Hz)	Error
48000	72	Yes	3	0	46875	2.34%	3	0	46875	2.34%
48000	48	No	15	1	48387.1	0.81%	8	0	46875	2.34%
48000	48	Yes	2	0	46875	2.34%	2	0	46875	2.34%
44100	72	No	25	1	44117.65	0.04%	13	0	43269.23	1.88%
44100	72	Yes	3	0	46875	6.29%	3	0	46875	6.29%
44100	48	No	17	0	44117.65	0.04%	8	1	44117.65	0.04%
44100	48	Yes	2	0	46875	6.29%	2	0	46875	6.29%
32000	72	No	35	0	32142.86	0.44%	17	1	32142.86	0.44%
32000	72	Yes	4	1	31250	2.34%	4	1	31250	2.34%
32000	48	No	23	1	31914.89	0.27%	11	1	32608.7	1.90%
32000	48	Yes	3	0	31250	2.34%	3	0	31250	2.34%
22050	72	No	51	0	22058.82	0.04%	25	1	22058.82	0.04%
22050	72	Yes	6	1	21634.61	1.88%	6	1	21634.61	1.88%
22050	48	No	34	0	22058.82	0.04%	17	0	22058.82	0.04%
22050	48	Yes	4	1	20833.33	5.52%	4	1	20833.33	5.52%
16000	72	No	70	1	15675.75	0.27%	35	0	16071.43	0.45%
16000	72	Yes	9	0	15625	2.34%	9	0	15625	2.34%
16000	48	No	47	0	15957.45	0.27%	23	1	15957.45	0.27%
16000	48	Yes	6	0	15625	2.34%	6	0	15625	2.34%
11025	72	No	102	0	11029.41	0.04%	51	0	11029.41	0.04%
11025	72	Yes	13	0	10817.3	1.88%	13	0	10817.3	1.88%
11025	48	No	68	0	11029.41	0.04%	34	0	11029.41	0.04%
11025	48	Yes	8	1	11029.41	0.04%	8	1	11029.41	0.04%
8000	72	No	140	1	8007.11	0.09%	70	1	7978.72	0.27%
8000	72	Yes	17	1	8035.71	0.45%	17	1	8035.71	0.45%
8000	48	No	94	0	7978.72	0.27%	47	0	7978.72	0.27%
8000	48	Yes	11	1	8152.17	1.90%	11	1	8152.17	1.90%

#### 15.4.4. 运行

##### 运行模式

运行模式是通过 SPI\_I2SCTRLR 寄存器的 I2SOM 位来选择的。共有四种运行模式可供选择，分别是主机发送模式，主机接收模式，从机发送模式和从机接收模式。各个运行模式下 I2S 接口信号的方向如下表所示。

表 15-5. 各个运行模式下 I2S 接口信号的方向

运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU <sup>(1)</sup>	输出	输出	输出
主机接收	输出或 NU <sup>(1)</sup>	输出	输出	输入

从机发送	输入或 NU <sup>(1)</sup>	输入	输入	输出
从机接收	输入或 NU <sup>(1)</sup>	输入	输入	输入

NU 表示该引脚没有被 I2S 使用，可以给其他功能使用。

## 状态位和中断

SPI\_STR 寄存器中有 6 个可用的标志位，分别是 TBE、RBNE、TRANS、I2SCH、TXURE 和 RXORE。用户通过这些标志位可以全面监视 I2S 总线的状态。

- 发送缓冲区空标志 (TBE)

发送缓冲区空的时候，该位被置 1。此时，如果 SPI\_CTLR2 寄存器的 TBEIE 位为 1，将产生中断。软件可以通过写 SPI\_DTR 寄存器将下一个数据写入发送缓冲区。在 SPI\_DTR 寄存器被写的时候，TBE 位被清零。

- 接收缓冲区非空标志 (RBNE)

接收缓冲区非空的时候，该位被置 1。此时，如果 SPI\_CTLR2 寄存器的 RBNEIE 位为 1，被产生中断。这表示有效数据已经被接收并存储在接收缓冲区。软件可以通过读 SPI\_DTR 寄存器来读取此数据。在 SPI\_DTR 寄存器被读的时候，RBNE 位被清零。

- 通信进行中标志(TRANS)

该位由硬件置 1 和清零，它表明 I2S 接口的状态。TRANS 标志可以用来监测数据通信的结尾，通常用于关闭 I2S 的软件程序段。可以避免破坏数据通信的最后一帧。要使用此标志位，下面描述的过程必须非常重视。传输开始时，TRANS 位会变为 1。主机接收模式除外。在主机接收模式下，TRANS 位在整个接收过程中都保持为 0。I2S 关闭或传输完成时，TRANS 位会变为 0。非连续通信时，每次通讯之间 TRANS 位都是 0。连续通讯时，在主机发送模式下，TRANS 位始终保持为 1；在从机模式下，TRANS 位会在每次传输之间变低一个 I2S 时钟周期。

- I2S 通道标志(I2SCH)

在发送模式下，I2SCH 标志在每次 TBE 标志由 0 变 1 时更新，表示即将要发送的数据属于哪个通道。在接收模式下，I2SCH 标志在每次 RBNE 标志由 0 变 1 时更新，表示刚接收到的数据属于哪个通道。注意，在出错的时候(TXURE 或 RXORE)，该标志不再可信，在恢复通信之前，I2S 需要被关闭再开启。此外，该标志对于 PCM 标准来说没有意义。

- 发送欠载错误标志(TXURE)

在从机发送模式下，如果收到数据传输的第一个时钟时发送缓冲区仍为空，TXURE 标志会置 1。此时，如果 SPI\_CTLR2 寄存器的 ERRIE 位为 1，将产生中断。在 SPI\_STR 寄存器被读的时候，TXURE 位被清零。

- 接收过载错误标志(RXORE)

在接收模式下，如果接收到新的数据时前一个数据仍没有从 SPI\_DTR 寄存器中读走，RXORE 标志会置 1。此时，如果 SPI\_CTLR2 寄存器的 ERRIE 位为 1，将产生中断。在这种情况下，接收缓冲区的数据将不会被新接收的数据更新。读 SPI\_DTR 寄存器将返回先前接收的正确数据。后续接收到的所有数据都将被丢弃。RXORE 标志会被如下的软件操作序列清零：先读

SPI\_DTR 寄存器，随后读 SPI\_STR 寄存器。

下表总结了 I2S 中断事件和相应的使能位。

表 15-6. I2S 中断

中断事件	标志位	使能位
发送缓冲区空	TBE	TBEIE
接收缓冲区非空	RBNE	RBNEIE
发送欠载错误	TXURE	ERRIE
接收过载错误	RXORE	ERRIE

### 初始化顺序

I2S 初始化序列包含以下五个步骤。如果要初始化 I2S 工作在主机模式，所有的五个步骤都需要被执行。如果要初始化 I2S 工作在从机模式，只需要步骤 2、3、4、5。

- 步骤 1：通过配置 SPI\_I2SCKP 寄存器的 DIV[7:0]位，OF 位和 MCKOE 位，定义 I2S 的比特率和 I2S\_MCK 是否需要提供。
- 步骤 2：通过配置 SPI\_I2SCTLR 寄存器的 CKPL 位，定义空闲状态时钟的极性。
- 步骤 3：通过配置 SPI\_I2SCTLR 寄存器的 I2SSEL 位，I2SSTD[1:0]位，PCMSM 位，I2SOM[1:0]位，DTLEN[1:0]位和 CHLEN 位，定义 I2S 特性。
- 步骤 4：通过配置 SPI\_CTLR2 寄存器的 TBEIE 位，RBNEIE 位，ERRIE 位，DMATE 位和 DMARE 位，选择中断源和 DMA 功能。此步骤可选。
- 步骤 5：将 SPI\_I2SCTLR 寄存器的 I2SEN 位置 1，来启动 I2S。

### 主机发送流程

TBE 标志被用来控制发送流程。如前文所述，TBE 标志表示发送缓冲区空了，此时，如果 SPI\_CTLR2 寄存器的 TBEIE 位为 1，将产生中断。刚开始时，发送缓冲区为空(TBE 为 1)，移位寄存器中没有发送序列。当 16 位数据被写入 SPI\_DTR 寄存器时(TBE 变为 0)，数据立即从发送缓冲区装载到移位寄存器中 (TBE 变为 1)。同时，发送序列开始。数据是并行地装载到 16 位移位寄存器中的，然后串行地从 I2S\_SD 引脚发出 (高位先发)。下一个数据应该在 TBE 为 1 时被写入 SPI\_DTR 寄存器。数据写入 SPI\_DTR 寄存器之后，TBE 变为 0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后 TBE 标志变回 1。为保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入 SPI\_DTR 寄存器。对于除 PCM 标准外的所有标准，I2SCH 标志被用来区别数据所属的通道。I2SCH 标志在每次 TBE 标志由 0 变 1 的时候更新。起初，I2SCH 标志为 0，表示左通道的数据应该被写入 SPI\_DTR 寄存器。

为关闭 I2S，I2SEN 位必须在 TBE 标志变成 1 且 TRANS 标志变成 0 之后清零。

### 主机接收流程

RBNE 标志被用来控制接收序列。如前文所述，RBNE 标志表示接收缓冲区非空，如果 SPI\_CTLR2 寄存器的 RBNEIE 位为 1，将产生中断。当 SPI\_I2SCTLR 寄存器的 I2SEN 位被

置 1 时，接收流程立即开始。起初，接收缓冲区为空(RBNE 为 0)。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区(RBNE 变为 1)。当 RBNE 为 1 时，用户应该将数据从 SPI\_DTR 寄存器中读走。读完以后，RBNE 变成 0。必须在下一次接收结束之前读走 SPI\_DTR 寄存器中的数据，否则将发生接收过载错误。此时 RXORE 标志位会被置 1，如果 SPI\_CTLR2 寄存器的 ERRIE 位为 1，将会发出中断。这种情况下，必须先关闭 I2S 再打开 I2S，然后再恢复通讯。

对于除 PCM 之外的所有标准来说，I2SCH 标志用来区分数据所属的通道。I2SCH 标志在每次 RBNE 标志由 0 变 1 时更新。

为关闭 I2S，需要采取特定的操作步骤来确保 I2S 正常结束当前传输并且没再发起新的传输。操作步骤依赖于所选的音频标准以及配置的数据长度和通道长度。每种情况的操作步骤如下所示。

- 数据长度为16位，通道长度为32位，LSB对齐标准 (DTLEN = 00, CHLEN = 1, I2SSTD = 10)
  1. 等待倒数第二个RBNE
  2. 等待17个I2S时钟周期
  3. 清除I2SEN位
- 数据长度为16位，通道长度为32位，除LSB对齐标准之外的其他标准 (DTLEN = 00, CHLEN = 1, I2SSTD不等于10)
  1. 等待最后一个RBNE
  2. 等待1个I2S时钟周期
  3. 清除I2SEN位
- 其他所有情况
  1. 等待倒数第二个RBNE
  2. 等待1个I2S时钟周期
  3. 清除I2SEN位

## 从机发送流程

从机发送流程与主机发送流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前开启。当外部主机开始发送时钟信号且 I2S\_WS 信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入 SPI\_DTR 寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入 SPI\_DTR 寄存器。否则会产生发送欠载错误。此时 TXURE 标志会置 1，如果 SPI\_CTLR2 寄存器的 ERRIE 位为 1，将会发出中断。这种情况下，必须先关闭 I2S 再打开 I2S，然后再恢复通讯。从机模式下，I2SCH 标志是根据外部主机发来的 I2S\_WS 信号而变化的。

为关闭 I2S，必须在 TBE 标志变为 1 且 TRANS 标志变为 0 之后才能清除 I2SEN 位。

### 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前开启。当外部主机开始发送时钟信号且 I2S\_WS 信号请求传输数据时，接收流程开始。从机模式下，I2SCH 标志是根据外部主机发来的 I2S\_WS 信号而变化的。

为关闭 I2S，必须在收到最后一个 RBNE 之后立即清除 I2SEN 位。

## 15.4.5. DMA 功能

DMA 的工作方式与 SPI 模式完全相同。唯一不同的地方是 I2S 模式不支持 CRC 功能。

## 15.5. SPI 寄存器

### 15.5.1. SPI 控制寄存器 1 (SPI\_CTLR1)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDM	BDOE	CRCEN	CRCNT	FF16	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC [2:0]	MSTMODE	SCKPL	SCKPH		

rw      rw

位/位域	名称	描述
15	BDM	双向数据模式使能 0: 2线各自单向传输模式 1: 1线双向传输模式。数据在主机的MOSI 脚和从设备的MISO 脚之间传输
14	BDOE	双向模式输出使能 当BDM被置1，该位决定数据的传输方向。 0: 只收模式 1: 只发模式
13	CRCEN	CRC计算使能 0: CRC计算禁用 1: CRC计算使能
12	CRCNT	下一次传输CRC 0: 下一个发送值为数据 1:下一个发送值为CRC值(TCR)

当数据传输被DMA管理，CRC值被硬件传输，此位应该被清0。

在全双工模式或只发模式下，当最后一个数据写入SPI\_DTR 寄存器后应将本位置位。

在只收模式下，应当在接收完倒数第二个数据后将本位置位。

11	<b>FF16</b>	数据帧格式 0: 8位数据帧格式 1: 16位数据帧格式
10	<b>RO</b>	只接收 当BDM位被清0，此位决定数据的传输方向。 0: 全双工(发送和接收) 1: 只收模式
9	<b>SWNSSEN</b>	NSS软件模式选择 0: NSS硬件模式，NSS输入引脚依赖IO脚 1: NSS软件模式，NSS输入引脚依赖于SWNSS位
8	<b>SWNSS</b>	NSS软件模式下NSS引脚选择 0: NSS引脚拉低 1: NSS引脚拉高 只有在SWNSSEN位被置1时此位有效
7	<b>LF</b>	数据高低位顺序 0: 先发送MSB，高位在前 1: 先发送LSB，低位在前
6	<b>SPIEN</b>	SPI使能 0: 禁止SPI设备 1: 开启SPI设备
5:3	<b>PSC[2:0]</b>	主设备时钟预分频选择 000: PCLK/2    100: PCLK/32 001: PCLK/4    101: PCLK/64 010: PCLK/8    110: PCLK/128 011: PCLK/16    111: PCLK/256 使用SPI1时，PCLK =PCLK2； 使用SPI2时，PCLK =PCLK1。
2	<b>MSTMODE</b>	主从模式选择 0: 配置为从设备 1: 配置为主设备
1	<b>SCKPL</b>	时钟极性选择 0: SPI空闲状态时，CLK保持低电平 1: SPI空闲状态时，CLK保持高电平
0	<b>SCKPH</b>	时钟相位选择 0: 在第一个时钟跳变沿采集第一个数据 1: 在第二个时钟跳变沿采集第一个数据

### 15.5.2. SPI 控制寄存器 2 (SPI\_CTLR2)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TBEIE	RBNEIE	ERRIE	保留		NSSDRV	DMATE	DMARE
								rw	rw	rw			rw	rw	rw

位/位域	名称	描述
15:8	保留	必须保持复位值
7	TBEIE	发送缓冲区空中断使能 0: TBE中断关闭 1: TBE中断使能, TBE位被置1将产生中断
6	RBNEIE	接收缓冲区非空中断使能 0: RBNE中断关闭 1: RBNE中断使能, 当RBNE位被置1时产生中断
5	ERRIE	错误中断使能 0: 错误中断关闭 1: 错误中断使能, CRCE位或CONFE位或RXORE位或TXURE位被置1的时候产生中断
4:3	保留	必须保持复位值
2	NSSDRV	NSS输出使能 0: NSS输出被禁止 1: NSS输出使能, 如果NSS引脚配置为输出, 当SPI使能时NSS引脚在主模式下被拉低 如果NSS引脚配置为输入, NSS引脚在主模式下应该被拉高, 此时这一位无效
1	DMATE	发送缓冲区DMA使能 0: 发送缓冲区DMA禁止 1: 发送缓冲区DMA使能, SPI_STR寄存器的TBE位被置1时, 相应DMA通道产生一个DMA请求
0	DMARE	接收缓冲区DMA使能 0: 接收缓冲区DMA禁止 1: 接收缓冲区DMA使能, SPI_STR寄存器的RBNE位被置1时, 相应DMA通道产生一个DMA请求

### 15.5.3. SPI 状态寄存器 (SPI\_STR)

地址偏移: 0x08

复位值: 0x0002

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					保留			TRANS	RXORE	CONFE	CRCE	TXURE	I2SCH	TBE	RBNE
								r	r	r	rc_w0	r	r	r	r

位/位域	名称	描述
15:8	保留	必须保持复位值
7	TRANS	通信进行中标志 0: SPI或I2S空闲 1: SPI或I2S目前正在发送或者接收数据，或者发送缓冲区非空 硬件置1和清0。
6	RXORE	接收过载错误标志 0: 没有发生接收过载错误 1: 发生接收过载错误 此标志位由硬件置1和软件序列清0，软件序列为先读SPI_DTR寄存器然后读SPI_STR。
5	CONFE	SPI配置错误标志 0: 无配置错误发生 1: 发生配置错误 (在主模式下，在硬件NSS模式下NSS引脚被拉低或软件NSS模式下SWNSS位为0都会产生CONFE错误) 此标志位由硬件置1和软件序列清0，软件序列为先读或写SPI_STR寄存器然后写SPI_CTLR1寄存器。 I2S模式下该位无用。
4	CRCE	SPI CRC错误表示 0: SPI_RCR值等于最后接收到的CRC值 1: SPI_RCR值不等于最后接收到的CRC值 此位由硬件置1和软件写0清0。 I2S模式下该位无用。
3	TXURE	发送欠载错误标志 0: 无发送欠载错误 1: 发生了发送欠载错误 该位由硬件置1，通过写SPI_STR寄存器清0。 SPI模式下该位无用。
2	I2SCH	I2S通道标志 0: 将要发送或者已经接收的数据属于左通道 1: 将要发送或者已经接收的数据属于右通道 硬件置1和清0。 SPI模式下该位无用，在I2S PCM模式下该位无意义。

---

1	TBE	发送缓冲区空标志 0: 发送缓冲区非空 1: 发送缓冲区空
0	RBNE	接收缓冲区非空标志 0: 接收缓冲区空 1: 接收缓冲区非空

#### 15.5.4. SPI 数据寄存器 (SPI\_DTR)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	DTR[15:0]	数据传输寄存器, 硬件有两个缓冲区: 发送缓冲区和接收缓冲区。向DTR写数据将把数据保存在发送缓冲区, 从DTR读0, 数据将从接收缓冲区中得到数据。如果数据帧格式被设置为8位, DTR[15:8]强制变为0。DTR[7:0]被用来发送和接收数据, 同时发送和接收缓冲区都变成8位。如果数据帧格式为16位, DTR[15:0]被用来发送和接收, 同时发送和接收缓冲区也为16位。

#### 15.5.5. SPI CRC 多项式寄存器 (SPI\_CPR)

地址偏移: 0x10

复位值: 0x0007

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CPR[15:0]	该寄存器包含CRC多项式, 用于CRC计算。缺省值为0007h.

#### 15.5.6. SPI 接收 CRC 寄存器 (SPI\_RCR)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

RCR[15:0]

r

位/位域	名称	描述
15:0	RCR[15:0]	当SPI_CTRLR1寄存器的CRCEN 置位时，硬件将会对接收的数据计算CRC值，并将其保存在RCR寄存器中。假如数据帧格式为8位，CRC 计算将采用CRC8标准，并且将结果保存在RCR[7:0]中；假如数据帧格式为16 位，CRC计算将采用CRC16标准，并且将结果保存在RCR[15:0]中。硬件在收到每个位后都会计算CRC。当TRANS置位时，读取本寄存器有可能返回的是计算过程产生的中间值。SPI_CTRLR1寄存器中的CRCEN位或SPIEN位被清0时复位此寄存器。

### 15.5.7. SPI 发送 CRC 寄存器 (SPI\_TCR)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

r

位/位域	名称	描述
15:0	TCR[15:0]	当SPI_CTRLR1寄存器的CRCEN位被置1，硬件计算发送出去的字节的CRC值后保存到TCR寄存器。如果是8位数据帧格式，CRC计算基于CRC8的标准进行，保存数据到TCR[7:0]，如果是16位数据帧格式，CRC计算基于CRC16的标准进行，保存数据到TCR[15:0].硬件在发送出每个位后都会计算CRC值，如果TRANS位被置1，读此寄存器将返回一个中间值。不同的数据高低位顺序(SPI_CTRLR1寄存器的LF位)也将返回不同的CRC值。SPI_CTRLR1寄存器中的CRCEN位或SPIEN位被清0时复位此寄存器。

### 15.5.8. SPI I2S 控制寄存器 (SPI\_I2SCTLR)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	I2SEL	I2SEN	I2SOM[1:0]	PCMSM	保留	I2SSSTD[1:0]	CKPL	DTLEN[1:0]	CHLEN						

rw rw

位/位域	名称	描述
15:12	保留	必须保持复位值

11	I2SEL	I2S模式选择 0: SPI模式 1: I2S模式 应该在SPI和I2S都关闭时配置此位。
10	I2SEN	I2S使能 0: I2S关闭 1: I2S开启 SPI模式不使用此位。
9:8	I2SOM[1:0]	I2S运行模式 00: 从机发送模式 01: 从机接收模式 10: 主机发送模式 11: 主机接收模式 应该在I2S关闭时配置此位。 SPI模式不使用此位。
7	PCMSM	PCM帧同步模式 0: 短帧同步 1: 长帧同步 只有在PCM标准下，此位才有意义。 应该在I2S关闭时配置此位。 SPI模式不使用此位。
6	保留	必须保持复位值
5:4	I2SSTD[1:0]	I2S标准选择 00: I2S飞利浦标准 01: MSB对齐标准 10: LSB对齐标准 11: PCM标准 应该在I2S关闭时配置此位。 SPI模式不使用此位。
3	CKPL	空闲状态时钟极性 0: I2S时钟空闲状态为低电平 1: I2S时钟空闲状态为高电平 应该在I2S关闭时配置此位。 SPI模式不使用此位。
2:1	DTLEN[1:0]	数据长度 00: 16位 01: 24位 10: 32位 11: 保留 应该在I2S关闭时配置此位。

SPI模式不使用此位。

0	CHLEN	通道长度 0: 16位 1: 32位 通道长度不能小于数据长度。 应该在I2S关闭时配置此位。 SPI模式不使用此位。
---	-------	--

### 15.5.9. SPI I2S 时钟预分频寄存器 (SPI\_I2SCKP)

地址偏移: 0x20

复位值: 0x0002

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:10	保留	必须保持复位值
9	MCKOE	I2S_MCK输出使能 0: I2S_MCK输出关闭 1: I2S_MCK输出开启 应该在I2S关闭时配置此位。 SPI模式不使用此位。
8	OF	预分频电路的奇系数 0: 实际分频系数为DIV * 2 1: 实际分频系数为DIV * 2 + 1 应该在I2S关闭时配置此位。 SPI模式不使用此位。
7:0	DIV[7:0]	预分频电路的分频系数 实际分频系数为DIV * 2 + OF。 DIV不能为0。 应该在I2S关闭时配置此位。 SPI模式不使用此位。

### 15.5.10. SPI 四路控制寄存器 (SPI\_QWCR) (仅适用于 GD32F170xx and GD32F190xx 产品)

地址偏移: 0x80

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												IO23_DRV	QRD	QMOD	
												RW	RW	RW	

位/位域	名称	描述
15:3	保留	必须保持复位值
2	IO23_DRV	IO2 和 IO3 输出使能 0: 单路模式下IO2和IO3输出关闭 1: 单路模式下IO2和IO3输出高电平 仅适用于 <b>SPI2</b> 。
1	QRD	四路读写位 0: SPI四路模式写操作 1: SPI四路模式读操作 该位仅能在SPI未通信时配置(TRANS位清0)。 仅适用于 <b>SPI2</b> 。
0	QMOD	四路模式使能 0: SPI工作在单路模式 1: SPI工作在四路模式 该位仅能在SPI未通信时配置(TRANS位清0)。 仅适用于 <b>SPI2</b> 。

## 16. 比较器 (CMP)

### 16.1. 简介

通用比较器 CMP1 和 CMP2，可独立使用(所有的终端都可以接到 I/O 口)，也可与定时器结合使用。它们可用于多种功能包括由通过模拟信号触发从省电模式中唤醒，模拟信号调理，以及与 DAC 和定时器输出的 PWM 相结合，组成逐周期的电流控制回路。

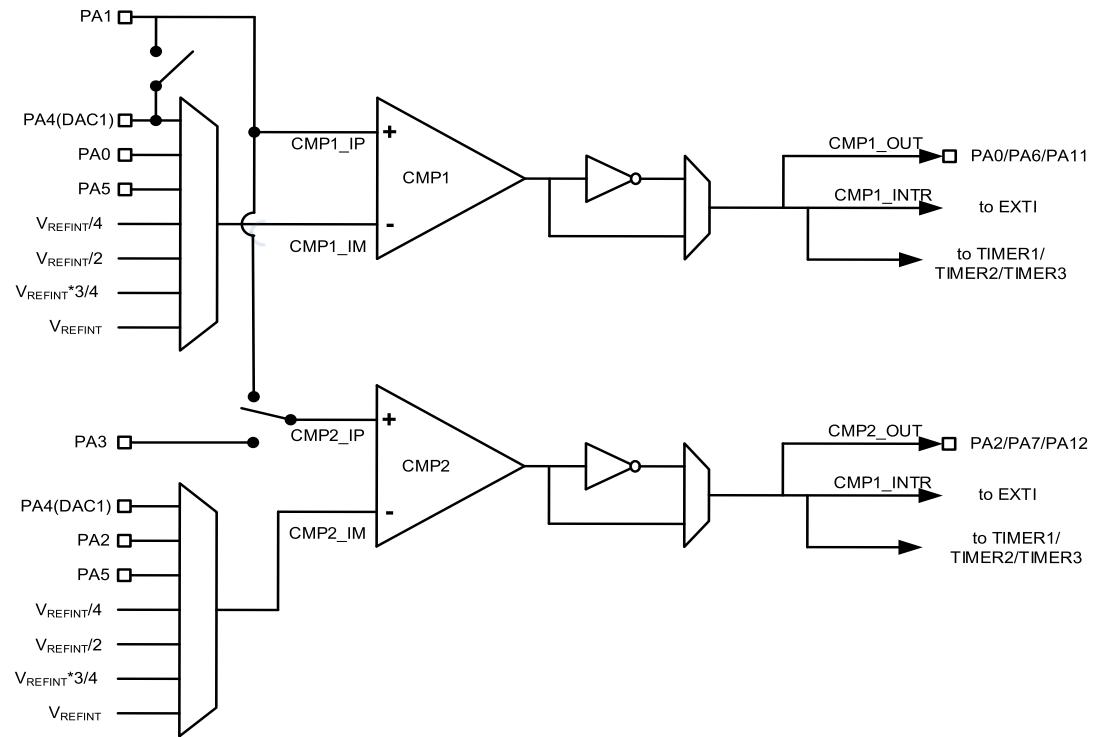
### 16.2. 主要特性

- 轨对轨比较器
- 可配置迟滞
- 可配置的速率和损耗
- 每个比较器有可配置的模拟输入源
  - DAC
  - 3 个 I/O 引脚
  - 内部参考电压及其等分电压值
- 窗口比较器
- 输出到 I/O 口
- 输出到定时器可以触发其他事件
- 输出到 EXTI

### 16.3. 功能描述

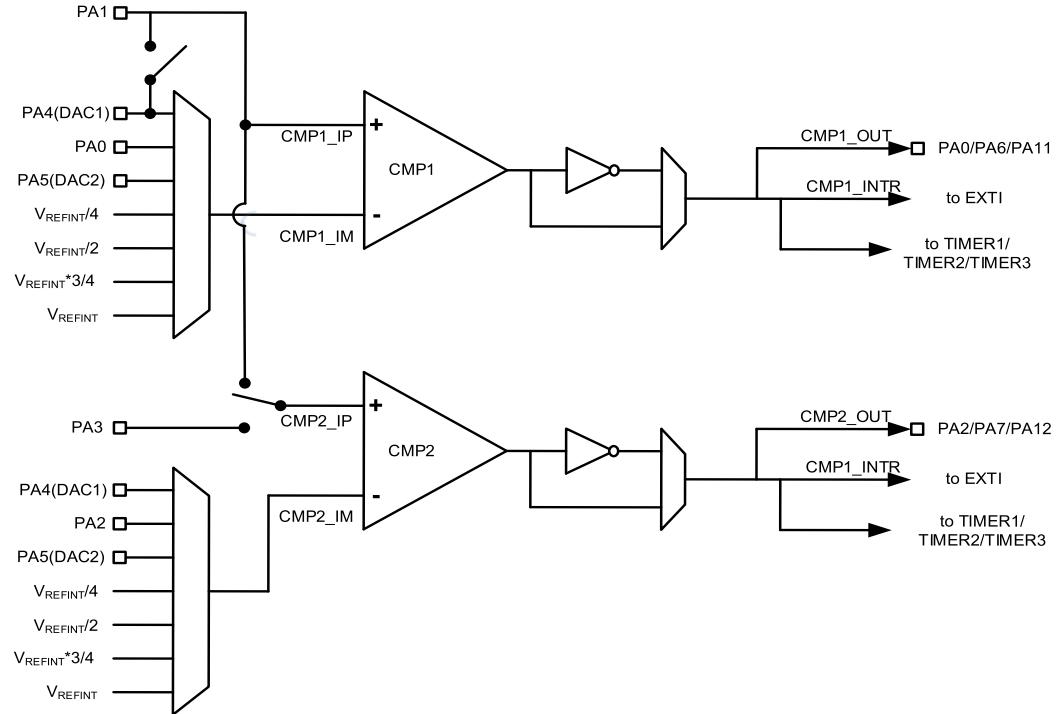
CMP 框图如下。

图 16-1. GD32F130xx 与 GD32F150xx 产品的 CMP 框图



注意: V<sub>REFINT</sub> 为 1.2V。

图 16-2. GD32F170xx 与 GD32F190xx 产品的 CMP 框图



注意: V<sub>REFINT</sub> 为 1.2V。

### 16.3.1. 比较器时钟和复位

CMP 时钟由时钟控制器提供，与 PCLK 同步。CMP 和 SYSCFG 共享复位使能和时钟使能位。

### 16.3.2. 比较器输入输出

在被选为比较器输入口之前，I/O 口必须由 GPIO 寄存器配置成模拟输入模式。

参照 Datasheet 中的引脚定义，比较器输出端必须接到具备相应备用功能 I/O 脚上。

各种定时器的输入端可以在内部与 CMP 的输出端相连确保以下功能：

- 定时测量的输入捕获
- 使用 BKIN 对 PWM 信号紧急关闭
- 使用 OCREF\_CLR 输入实现逐周期电流控制

极性选择逻辑与输出端口的重定向工作独立于 PCLK 时钟，这使得比较器可以在深度睡眠模式下工作。

比较器的输出可以在片内和片外同时重定向。

比较器的输出端口可以内部连接到扩展中断和事件控制器。每个比较器都有自己的 EXTI 线路并且能够产生中断或事件。从省电模式返回也是相同的机制。

### 16.3.3. 比较器电源模式

对于一个给定的应用，比较器通过电源消耗和传输延迟的折中可以调整到最适宜的状态，这一过程通过配置 CMP\_CSR 寄存器的 CMPxM[1:0]位来实现。当 CMPxM = 00 时，比较器工作速度最快，并且电源消耗也最大；当 CMPxM = 11 时，比较器工作速度最慢，电源消耗也最小。

### 16.3.4. 比较器迟滞

为了避免噪声信号引起的转换输出的假象，比较器包含了一个可编程的迟滞，用以强制使用外部期间的迟滞值。在不需要的时候(例如从省电模式返回)，这个功能可以关闭。

### 16.3.5. 比较器寄存器写保护

出于对应用的安全考虑，例如过流保护或热保护，及其他特殊功能保护需求，有必要保证比较器的配置在假的寄存器访问或者程序计数器崩溃的情况下不会被改写。

出于这个考虑，应该在配置结束以后立即通过设置 CMPxLK 位为 1 可以使比较器控制和状态寄存器进入写保护状态。总体的 CMP\_CSR 寄存器将变成只读，包括 CMPxLK 位。

只有 MCU 的复位才能复位 CMPxLK 位。

## 16.4. CMP 寄存器

### 16.4.1. CMP 控制和状态寄存器 (CMP\_CSR)

**GD32F130xx 和 GD32F150xx 产品：**

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP2LK	CMP2O	CMP2HST[1:0]	CMP2PL	CMP2OSEL[2:0]	WNDEN	CMP2MSEL[2:0]	CMP2M[1:0]	保留	CMP2EN						
rwo	r	rw/r		rw/r		rw/r		rw/r		rw/r		rw/r		rw/r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1LK	CMP1O	CMP1HST[1:0]	CMP1PL	CMP1OSEL[2:0]	保留	CMP1MSEL[2:0]	CMP1M[1:0]	CMP1SW	CMP1EN						
rwo	r	rw/r		rw/r		rw/r		rw/r		rw/r		rw/r		rw/r	

位/位域	名称	描述
31	CMP2LK	比较器 2 锁定 该位可以使比较器 2 的所有控制位只读。该位只能被写一次。一旦被软件置 1 该位只能通过系统复位来清 0。 0: CMP_CSR[31:16]位可读可写 1: CMP_CSR[31:16]位只读
30	CMP2O	比较器 2 输出 该位反映比较器 2 的输出状态, 只读。 0: 低输出 (同相输入低于反相输入) 1: 高输出 (同相输入高于反相输入)
29:28	CMP2HST[1:0]	比较器 2 迟滞 这些位控制比较器 2 的迟滞程度。 00: 没有迟滞 01: 低度迟滞 10: 中度迟滞 11: 高度迟滞
27	CMP2PL	比较器 2 输出极性 该位用于切换比较器 2 输出极性。 0: 同相输出 1: 反相输出
26:24	CMP2OSEL[2:0]	比较器 2 输出选择 这些位用来选择比较器输出方向。 000: 无选择

		001: 定时器 1 刹车输入 010: 定时器 1 输入捕捉 1 011: 定时器 1 OCREF_CLR 输入 100: 定时器 2 输入捕捉 4 101: 定时器 2 OCREF_CLR 输入 110: 定时器 3 输入捕捉 1 111: 定时器 3 OCREF_CLR 输入
23	WNDEN	窗口模式使能  该位使 CMP2_IP 端与 PA3 断开，并且与和 CMP1_IP 端相连。 0: CMP2_IP 连接到 PA3 1: CMP2_IP 连接到 CMP1_IP
22:20	CMP2MSEL[2:0]	比较器 2 反相输入选择  这些位用于选择连接到比较器 2 的反相输入的信号源。 000: V <sub>REFINT</sub> /4 001: V <sub>REFINT</sub> /2 010: V <sub>REFINT</sub> *3/4 011: V <sub>REFINT</sub> 100: PA4 (DAC1) 101: PA5 110: PA2 111: 保留
19:18	CMP2M[1:0]	比较器 2 模式  比较器 2 的工作模式控制位，允许调整速率和损耗。 00: 高速/高功耗 01: 中速/中等功耗 10: 低速/低功耗 11: 极低速率/超低功耗
17	保留	必须保持复位值
16	CMP2EN	比较器 2 使能 0: 比较器 2 关闭 1: 比较器 2 打开
15	CMP1LK	比较器 1 锁定  该位只能写一次，由软件置 1，由系统复位清零。 它令比较器 1 的所有控制位为只读。 0: CMP_CSR[15:0] 位可读可写 1: CMP_CSR[15:0] 位只读
14	CMP1O	比较器 1 输出  只读，反映应比较器 1 输出状态。 0: 低输出（同相输入低于反相输入） 1: 高输出（同相输入高于反相输入）

---

13:12	CMP1HST[1:0]	比较器 1 迟滞 这些位用于控制比较器 1 的迟滞程度。 00: 无迟滞 01: 低度迟滞 10: 中度迟滞 11: 高度迟滞
11	CMP1PL	比较器 1 输出极性 该位用于切换比较器 1 输出极性。 0: 非反相输出 1: 反相输出
10:8	CMP1OSEL[2:0]	比较器 1 输出选择 这些位用来选择比较器 1 的输出方向。 000: 无选择 001: 定时器 1 刹车输入 010: 定时器 1 输入捕捉 1 011: 定时器 1 OCREF_CLR 输入 100: 定时器 2 输入捕捉 4 101: 定时器 2 OCREF_CLR 输入 110: 定时器 3 输入捕捉 1 111: 定时器 3 OCREF_CLR 输入
7	保留	必须保持复位值
6:4	CMP1MSEL[2:0]	比较器 1 反相输入选择 这些位用于选择连接到比较器 1 的反相输入的信号源。 000: V <sub>REFINT</sub> /4 001: V <sub>REFINT</sub> /2 010: V <sub>REFINT</sub> *3/4 011: V <sub>REFINT</sub> 100: PA4 (DAC1) 101: PA5 110: PA0 111: 保留
3:2	CMP1M[1:0]	比较器 1 模式 比较器 1 的工作模式控制位，允许调整速率和损耗。 00: 高速/全功耗 01: 中速/中等功耗 10: 低速/低功耗 11: 极低速/超低功耗
1	CMP1SW	比较器 1 开关 该位关闭 PA0 上比较器 1 的同相输入端和 PA4(DAC1)的 I/O 之间的开关。 0: 开关断开

1: 开关闭合

0            CMP1EN            比较器 1 使能  
               0: 比较器 1 关闭  
               1: 比较器 1 打开

### GD32F170xx 和 GD32F190xx 产品:

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP2LK	CMP2O	CMP2HST[1:0]	CMP2PL	CMP2OSEL[2:0]	WNDEN	CMP2MSEL[2:0]	CMP2M[1:0]	保留	CMP2EN						
rwo	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1LK	CMP1O	CMP1HST[1:0]	CMP1PL	CMP1OSEL[2:0]	保留	CMP1MSEL[2:0]	CMP1M[1:0]	CMP1SW	CMP1EN						
rwo	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位/位域	名称	描述
31	CMP2LK	比较器 2 锁定  该位可以使比较器 2 的所有控制位只读。该位只能被写一次。一旦被软件置 1 该位只能通过系统复位来清 0。  0: CMP_CSR[31:16]位可读可写 1: CMP_CSR[31:16]位只读
30	CMP2O	比较器 2 输出  该位反映比较器 2 的输出状态, 只读。  0: 低输出 (同相输入低于反相输入) 1: 高输出 (同相输入高于反相输入)
29:28	CMP2HST[1:0]	比较器 2 迟滞  这些位控制比较器 2 的迟滞程度。  00: 没有迟滞 01: 低度迟滞 10: 中度迟滞 11: 高度迟滞
27	CMP2PL	比较器 2 输出极性  该位用于切换比较器 2 输出极性。  0: 同相输出 1: 反相输出
26:24	CMP2OSEL[2:0]	比较器 2 输出选择  这些位用来选择比较器输出方向。

		000: 无选择 001: 定时器 1 刹车输入 010: 定时器 1 输入捕捉 1 011: 定时器 1 OCREF_CLR 输入 100: 定时器 2 输入捕捉 4 101: 定时器 2 OCREF_CLR 输入 110: 定时器 3 输入捕捉 1 111: 定时器 3 OCREF_CLR 输入
23	WNDEN	窗口模式使能  该位使 CMP2_IP 端与 PA3 断开，并且与和 CMP1_IP 端相连。 0: CMP2_IP 连接到 PA3 1: CMP2_IP 连接到 CMP1_IP
22:20	CMP2MSEL[2:0]	比较器 2 反相输入选择  这些位用于选择连接到比较器 2 的反相输入的信号源。 000: V <sub>REFINT</sub> /4 001: V <sub>REFINT</sub> /2 010: V <sub>REFINT</sub> *3/4 011: V <sub>REFINT</sub> 100: PA4 (DAC1) 101: PA5 (DAC2) 110: PA2 111: 保留
19:18	CMP2M[1:0]	比较器 2 模式  比较器 2 的工作模式控制位，允许调整速率和损耗。 00: 高速/高功耗 01: 中速/中等功耗 10: 低速/低功耗 11: 极低速率/超低功耗
17	保留	必须保持复位值
16	CMP2EN	比较器 2 使能  0: 比较器 2 关闭 1: 比较器 2 打开
15	CMP1LK	比较器 1 锁定  该位只能写一次，由软件置 1，由系统复位清零。 它令比较器 1 的所有控制位为只读。 0: CMP_CSR[15:0] 位可读可写 1: CMP_CSR[15:0] 位只读
14	CMP1O	比较器 1 输出  只读，反映应比较器 1 输出状态。 0: 低输出（同相输入低于反相输入）

		1: 高输出（同相输入高于反相输入）
13:12	CMP1HST[1:0]	比较器 1 迟滞 这些位用于控制比较器 1 的迟滞程度。 00: 无迟滞 01: 低度迟滞 10: 中度迟滞 11: 高度迟滞
11	CMP1PL	比较器 1 输出极性 该位用于切换比较器 1 输出极性。 0: 非反相输出 1: 反相输出
10:8	CMP1OSEL[2:0]	比较器 1 输出选择 这些位用来选择比较器 1 的输出方向。 000: 无选择 001: 定时器 1 刹车输入 010: 定时器 1 输入捕捉 1 011: 定时器 1 OCREF_CLR 输入 100: 定时器 2 输入捕捉 4 101: 定时器 2 OCREF_CLR 输入 110: 定时器 3 输入捕捉 1 111: 定时器 3 OCREF_CLR 输入
7	保留	必须保持复位值
6:4	CMP1MSEL[2:0]	比较器 1 反相输入选择 这些位用于选择连接到比较器 1 的反相输入的信号源。 000: $V_{REFINT}/4$ 001: $V_{REFINT}/2$ 010: $V_{REFINT} \cdot 3/4$ 011: $V_{REFINT}$ 100: PA4 (DAC1) 101: PA5 (DAC2) 110: PA0 111: 保留
3:2	CMP1M[1:0]	比较器 1 模式 比较器 1 的工作模式控制位，允许调整速率和损耗。 00: 高速/全功耗 01: 中速/中等功耗 10: 低速/低功耗 11: 极低速/超低功耗
1	CMP1SW	比较器 1 开关 该位关闭 PA0 上比较器 1 的同相输入端和 PA4(DAC1)的 I/O 之间的开关。

0: 开关断开

1: 开关闭合

0            CMP1EN      比较器 1 使能

0: 比较器 1 关闭

1: 比较器 1 打开

## 17. 通用同步异步收发器 (USART)

### 17.1. 简介

通用同步异步收发器(USART)提供了一个灵活方便的全双工数据交换接口，支持同步或异步数据传输方式。USART 用来在串行接口之间传递数据，也广泛的用于 RS232 标准的通信。USART 提供了可编程的波特率发生器，能对系统时钟进行分频产生 USART 发送和接收所需的特定频率。

它支持半双工单线同步通信，LIN，智能卡协议和红外编码规范。它还支持 Modem 流控操作 (CTS/RTS) 和多处理器通信。

USART 还支持 DMA 功能，来实现高速率的数据通信。

### 17.2. 主要特性

- 全双工异步通信
- 半双工单线通信
- NRZ 标准格式(Mark/Space)
- 双时钟域
  - 互为异步关系的APB时钟和USART时钟
  - 不依赖于PCLK设置的波特率设置
- 可编程的波特率产生器，当时钟频率为72MHz，过采样为8倍，最高速度可达9 MBits/s
- 完全可编程的串口特性：
  - 数据位(8或9位)低位或高位在前
  - 偶校验位，奇校验位，无校验位的生成/检测
  - 产生1, 1.5或者2个停止位
- 可互换的Tx/Rx引脚
- 可配置的数据极性
- 自动检测波特率
- 支持硬件Modem流控操作(CTS/RTS)和RS485驱动使能
- 借助集中式DMA，可实现可配置的多级缓存通信
- 发送器和接收器可分别使能
- 传输检测标志：

- 接收缓存满
  - 发送缓存空
  - 发送结束标志
- 奇偶校验位控制：
    - 发送奇偶校验位
    - 检测接收到字节的奇偶校验位
  - 错误检测：过载，噪声，帧格式错误和奇偶校验错误
  - LIN断开帧的产生和检测
  - 支持红外数据协议
  - 同步传输模式以及为同步传输输出发送时钟
  - 支持兼容ISO7816-3 (T=0和T=1)的智能卡接口
  - 多处理器通信
    - 假如地址不匹配，始终保持静默模式
    - 通过线路空闲或者地址掩码检测从静默模式唤醒
  - 支持 ModBus 通信
    - 超时功能
    - CR/LF字符识别
  - 从深度睡眠模式唤醒
    - 通过标准RBEN中断
    - 通过WUF中断
- 14个带标志的中断源：
    - CTS变化
    - LIN暂停帧检测
    - 发送数据寄存器空
    - 发送完成
    - 接受数据寄存器满
    - 空闲线检测
    - 过载错误
    - 帧错误

- 噪声错误
- 奇偶校验错误
- 地址/字节匹配
- 接收超时中断
- 块结束中断
- 从深度睡眠模式唤醒

USART1 完全实现上述功能，但是 USART2 只实现了上面所介绍功能的部分，下面这些功能在 USART2 中没有实现：

- 自动波特率检测
- 智能卡模式
- IrDA SIR ENDEC模块
- LIN模式
- 双时钟域和从深度睡眠模式唤醒
- 接收超时中断
- Modbus 通信

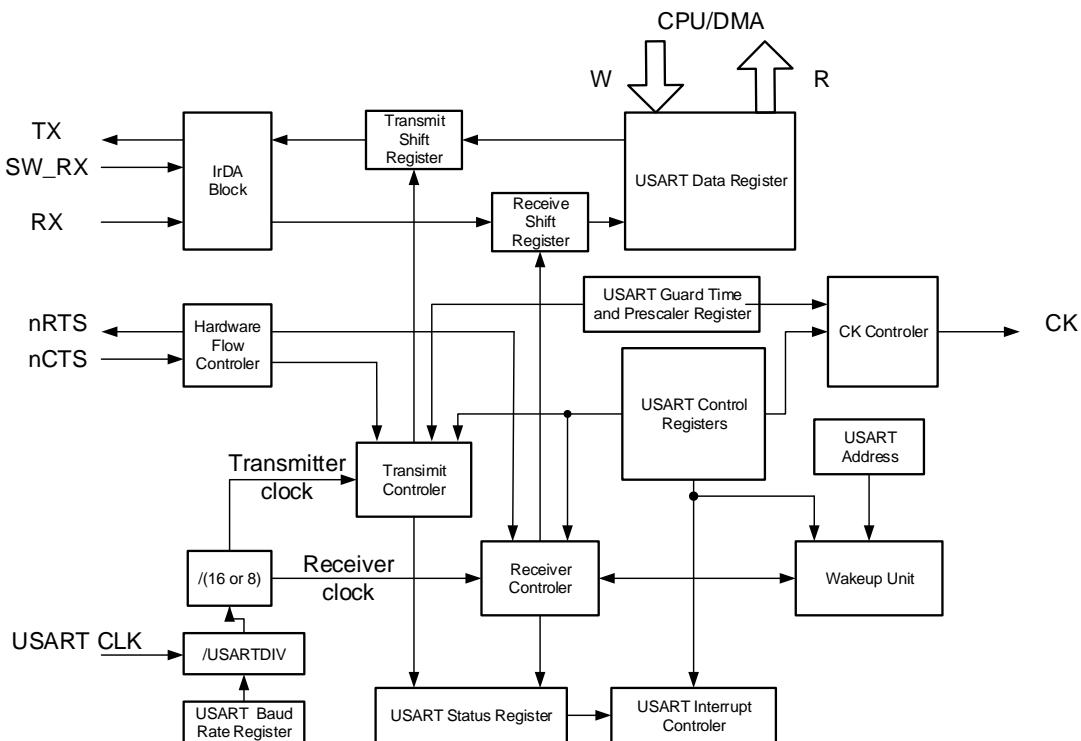
### 17.3. 功能描述

USART 接口通过以下主要管脚从外部连接到其他设备

表 17-1. USART 重要管脚描述

管脚名称	电气类型	描述
RX	输入	接收数据
TX	输出 I/O (单线模式/智能卡模式)	发送数据 当 USART 使能后，若无数据发送， 默认为高电平
CK	输出	串行时钟 用于同步通信
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

图 17-1. USART 模块内部框图



### 17.3.1. USART 发送器

当 USART\_CTLR1 寄存器的使能位(TEN)置位后，波特率发生器将产生传送器时钟脉冲并且通过 CK 管脚输出。之后串行码流将按控制寄存器中的设置，在一个空闲帧之后，由发送器开始传输。

在传送被破坏的情况下，只要传送还在继续，便不得禁用 TEN 位。

在 TBE 位处于置位状态时，数据可以在不覆盖前一个数据的情况下写入 USART\_TDTR 寄存器。当数据写入 USART\_TDTR 寄存器，该位将被清 0。

假如一帧数据已经发送出去，并且 TBE 位已经置位，那么 TC 位将被置 1。假如 USART\_CTLR1 寄存器中的中断使能位(TCIE)为 1，将会有中断产生。

**USART** 发送应按照下列步骤进行：

1. 写 USART\_CTLR1 寄存器的 WL 去设置字长
  2. 在 USART\_CTLR2 寄存器中设置停止位的长度
  3. 如果选择了多级缓存通信方式，应该在 USART\_CTLR3 寄存器中使能 DMA (DENT 位)
  4. USART\_BRR 寄存器中设置波特率
  5. 在 USART\_CTLR1 寄存器中设置 UEN 位，使能 USART
  6. 在 USART\_CTLR1 寄存器中设置 TEN 位

7. 等待 TBE 置位
8. 向 USART\_TDTR 寄存器写数据
9. 等待 TC=1，发送完成

在禁用 USART 或进入低功耗状态之前，必须等待 TC 置位。

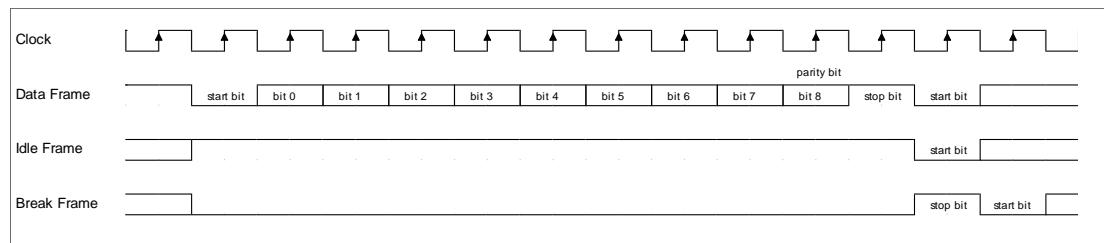
先读 USART\_STR 然后再写 USART\_TDTR 可将 TC 位清 0。在多级缓存通信方式下，直接向 TC 写 0，也能清 TC。

当 SBKCMD 置位时，会发送一个断开帧，发送完成后，SBKCMD 将清 0。

**表 17-2. 停止位配置**

停止位长度 (bit)	描述
1	默认值
1.5	智能卡模式
2	标准 USART，单线以及调制解调模式

**图 17-2. USART 字符帧 (9 数据位和 1 停止位)**



如果开启了奇偶检验位，则最高位被用来做奇偶校验位。

假如数据位(除奇偶校验位)中“1”的个数为偶数个，奇偶校验位应当为 ‘0’ (偶校验) 或‘1’ (奇校验)。

### 17.3.2. USART 接收器

当检测到有效的开始脉冲后，接收器开始接收码流。在此期间还会进行过载，奇偶校验，帧格式错误和断开帧检测。

如果一帧接收完成，并且 RBNE 位置位，那么就可以读 USART\_RDTR 寄存器和 USART\_STR 的相关位。若 USART\_CTLR1 寄存器的中断使能位(RBNEIE)被置位，就会产生中断。

RBNE 位必须在下一帧数据到达之前清 0，否则就会造成过载错误。

在数据接收被破坏的情况下，只要接收还在继续，便不得禁用 REN 位。

在多级缓存通信模式下直接读 USART\_RDTR，可以将 RBNE 位清 0。在多缓冲模式下的 DMA 读操作也会将 RBNE 位清 0。

USART 接收应按下面步骤进行：

1. 写 USART\_CTLR1 寄存器的 WL 去设置字长

2. 在 USART\_CTLR2 寄存器中设置停止位的长度
3. 如果选择了多级缓存通信方式，因该在 USART\_CTLR3 寄存器中使能 DMA (DENT 位)
4. USART\_BRR 寄存器中设置波特率
5. 在 USART\_CTLR1 寄存器中设置 UEN 位，使能 USART
6. 在 USART\_CTLR1 中设置 REN 位

### 17.3.3. 接收错误

若 RBNE 已经置位的情况下或者前一个 DMA 请求还未被处理的情况下又接收到一个新数据，将会产生过载错误。此时 USART\_STR 寄存器的 ORE 位被置位。

USART 模块会按照内部波特率参考时钟对 RX 状态线进行过采样。如果检测到噪声错误，USART\_STR 的 NE 位将在 RBNE 位的上升沿置位并且从移位寄存器收到无效数据。

当在预期时间未检测到停止位，将产生帧格式错误。USART\_STR 的 FE 位将在 RBNE 位的上升沿置位，并且从移位寄存器收到无效数据。

通过写 USART\_SCR 寄存器的 OREC, NEC 和 FEC 可分别的将以上三个错误标志位清 0。

### 17.3.4. 波特率发生

波特率分频系数是一个 16 位的数字，包含 12 位整数部分和 4 位小数部分。波特率发生器使用这两位组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使 USART 能够产生所有标准波特率。

波特率分频系数 (USARTDIV) 与系统时钟具有如下关系：

如果过采样率是 16，公式如下：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}}$$

如果过采样率是 8，公式如下：

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}}$$

USART 时钟(UCLK)的选择是通过时钟控制系统来实现的(参考 RCC 部分)。在使能 USART 之前，必须选好时钟源 (通过设置 UEN 位)。

### 17.3.5. 自动波特率检测

USART 能够基于接收到的一个字符自动检测和设置 USART\_BRR 寄存器的值。通过设置 USART\_CTLR2 寄存器的 ABDM 位，有两种自动波特率检测方法可以选择。这两种方法是：

1. 以 1 开始的任一字节。在这种情况下，USART 将开始测量起始位的长度(下降沿到上升沿)。

2. 以 10xx 开头的任何字符，这种情况下 USART 将测量起始位和第一个数据位的总长度，通过下降沿到下降沿，以减少信号斜率对测量精度的影响。

### 17.3.6. 多处理器通信

在多处理器通信中，未被寻址的接收方将进入静默模式。在接收期间，只有目标接收器才会激活来接收随后的数据，与此同时，那些不相干的接收器将处于静默模式。通过空闲总线检测和地址掩码检测可以使器件进入或退出静默模式。

在空闲总线检测期间，当 USART\_CMD 寄存器的 MMCMD 被置位时，USART 进入静默模式并且将 USART\_STR 的 RWU 置位。一旦检测到空闲帧，会立即退出静默模式。USART\_STR 寄存器的 RWU 位被清 0，但是 USART\_STR 的 IDLEF 位没有置 1。

在地址掩码检测期间，最高位为 1 的字节会被认为是寻址字节。寻址字节的低 4 位或低 7 位即为目标接收器的地址。接收器将寻址字节与他们自己的地址进行比较，然后根据结果进入或退出静默模式(置位或复位 RWU 位)。

### 17.3.7. ModBus 通信

通过实现块尾检测功能，USART 提供对 Modbus/RTU 和 Modbus/ASCII 协议实现的基本支持。

Modbus/RTU 这个模式下，块尾通过一个超过 2 个字符长度的空闲状态来识别。这个功能是通过一个可设置的超时检测功能来实现的。

为了检测空闲状态，USART\_CTLR2 寄存器的 RTEN 位和 USART\_CTLR1 寄存器的 RTIE 位必须置位。RTR 寄存器必须设置成和 2 个字节超时所对应的时间。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件现在接收块已经完成。

Modbus/ASCII 在这个模式下，块尾被认为是一个特定的字符(CR/LF)串。USART 用字符匹配机制实现这个功能。具体是通过将 LF 的 ASCII 码配置到 ADDR 区域并激活地址匹配中断(AMIE=1)来实现的。软件将在收到 LF 或可以在 DMA 缓存中查找到 CR/LF 时得到提示。

### 17.3.8. LIN 模式

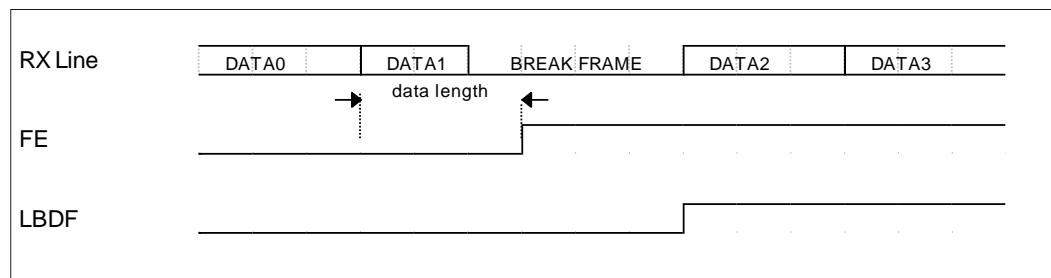
将 USART\_CTLR2 的 LMEN 置位即可使能本地互联网络模式。在 LIN 模式下，USART\_CTLR2 的 CKEN, STB 以及 USART\_CTLR3 的 SCEN, HDEN, IREN 位都应该被清 0。

LIN 发送过程与普通发送过程基本相同。数据位的长度只能是 8，断开帧为连续的 13 个 0。断开帧检测是完全独立于 USART 接收器的。因此在空闲状态下或者帧传输状态下都可以进行断开帧检测。

在使能接收器和检测到起始位后，电路将开始采样下一位。

在断开检测期间，如果遇到了帧格式错误，那么断开检测将要停止，直到 RX 线变成高电平，再次开始断开检测。之后重新开始检测起始位。假如在分隔符(高电平)之前检测到连续的 10/11 (根据 USART\_CTLR2 寄存器中的 LBDL 设置) 个'0'，那么在 USART\_STR 中的 LIN 断开标志位也会被置位。

图 17-3. LIN 模式下的帧格式错误检测以及断开帧检测



### 17.3.9. 半双工通信模式

通过设置 USART\_CTRLR3 寄存器的 HDEN 位，可以使能半双工模式。在半双工通信模式下，USART\_CTRLR2 寄存器的 LMEN, CKEN 位和 USART\_CTRLR3 寄存器的 SCEN, IREN 位清零。

半双工模式下仅用单线通信。TX 和 RX 将从内部连接到一起。TX 管脚应该被配置为 IO 管脚。通信冲突应由软件负责处理。当 TEN 被置位时，在数据寄存器中的数据将会被发送。

### 17.3.10. 同步通信模式

USART 支持主机模式下的全双工同步串行通信，可以通过置位 USART\_CTRLR2 的 CKEN 位来使能。在同步模式下，USART\_CTRLR2 的 LMEN 和 USART\_CTRLR3 的 SCEN, HDEN, IREN 位应该被清 0。CK 管脚作为 USART 同步发送器的时钟输出，仅仅当 TEN 位被使能时，它才被激活。在起始位和停止位传送期间，不会从 CK 管脚输出时钟脉冲。USART\_CTRLR2 的 LBCP 位用来决定在最低位(地址索引位)发送期间是否有时钟信号输出。在空闲状态和断开帧发送期间也不会有时钟信号产生。USART\_CTRLR2 的 CPH 位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART\_CTRLR2 的 CPL 位用来决定在 USART 同步模式空闲状态下，时钟管脚的电平。

在发送器或接收器使能的情况下，不能改变这 3 位(CPL, CPH, LBCP) 的值。

时钟与已发送的数据同步。同步模式下的接收器按照发送器的时钟进行采样，没有任何过采样。

图 17-4. 同步模式下的 USART 示例

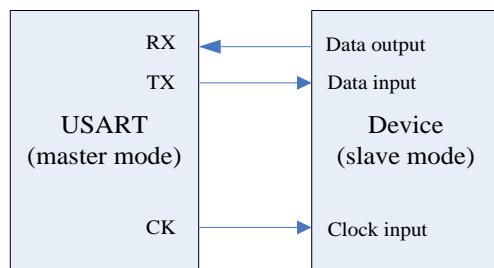
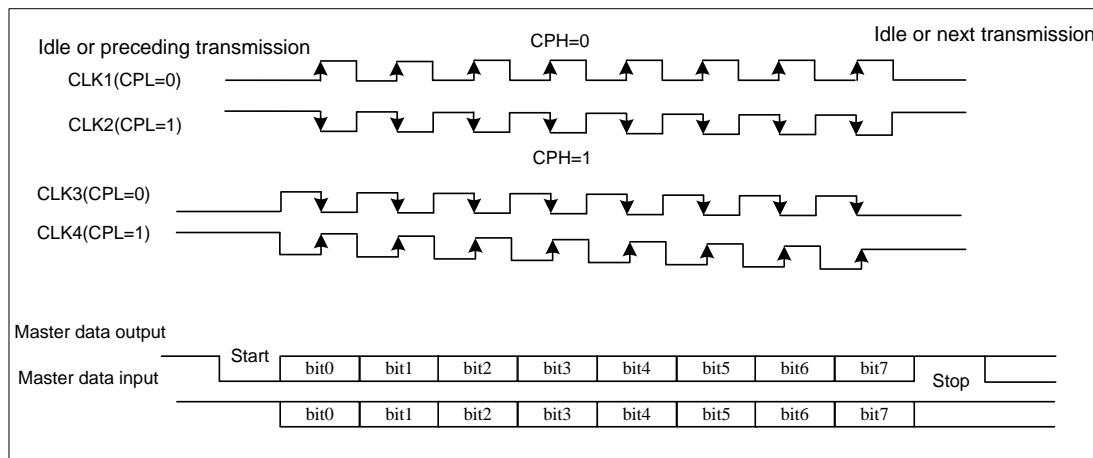


图 17-5. 8-bit 格式的 USART 同步通信波形 (LBCL=1)



### 17.3.11. 智能卡(ISO7816)模式

智能卡模式是一种异步通信模式，将 USART\_CTLR3 寄存器的 SCEN 位置位即可使能智能卡模式。在智能卡模式下，USART\_CTLR2 的 LMEN 位和 USART\_CTLR3 的 HDEN, IREN 位应该清 0。

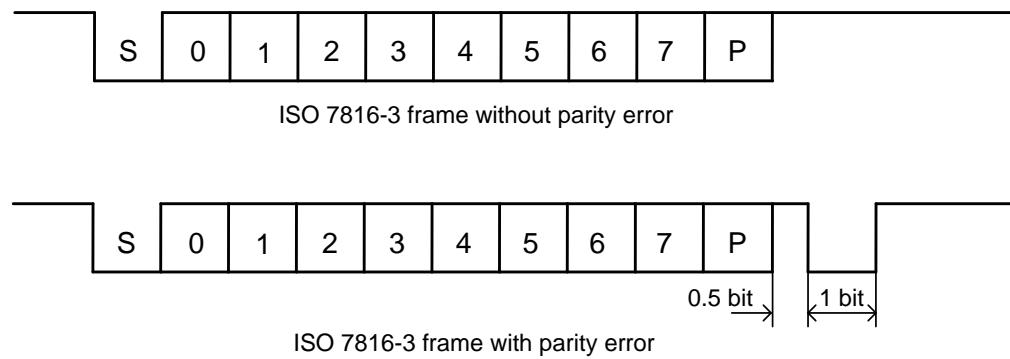
假如 CKEN 位被置位，USART 将向智能卡提供一个时钟。该时钟可以分频用于其他用途。

智能卡模式下的帧格式为：1 起始位+9 数据位(包括 1 奇偶校验位)+1.5 停止位。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX 管脚需要被设置成开漏模式，这个管脚将会与智能卡驱动同一条双向连线。

#### T=0 模式

图 17-6. ISO7816-3 帧格式



相较于正常操作模式下的时序，从发送移位寄存器到 TX 管脚的传递时间延迟了半个波特率时钟，并且 TC 标志的置位将根据保护时间寄存器的设置延迟某一特定时间。根据的协议，USART 能自动的再次发送数据，重发次数在 SCRTNUM 中设置。在上一个重复的字节接收结束，TC 位将会立即置位而没有一个保护时间。如果 USART 连续收到 NACK 信号，那么在设定好的重新发送次数后它将会停止发送并将该错误标注为帧格式错误。USART\_CMD 寄存器

的 TXFCMD 位可将 TBE 位清 0。

在 USART 发送期间，在一帧数据的接收过程中，如果检测到有奇偶校验错误，TX 线将拉低一个波特率时钟。这个信号是发送‘NACK’(应答错误)信号到智能卡。之后智能卡一侧就会产生帧格式错误。如果接收到的字符是错误的，RBNE 中断和接收 DMA 请求都不会被激活。根据协议，智能卡将要重新发送数据。如果在最大的重新发送次数后(这个次数的具体值在 SCRTNUM 位域)，接收到的字符仍然是错误的，USART 停止发送 NACK 信号和标注这个错误为奇偶校验错误。

若 USART\_CTLR3 寄存器的 NACK 被置位，‘NACK’信号将被发送到 USART。并且 USART 不会将‘NACK’误当作起始位。

空闲帧和断开帧在智能卡模式下不适用。

### T=1 模式 (块模式)

在 T=1(块模式)下，UART\_CTLR3 寄存器的 NACK 位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将 RTR 寄存器设置为 BWT (块等待) – 11 的值并且 RBNEIE 必须置位。如果这个时间到了，还没有从卡收到应答，将引起超时中断。如果在超时之前收到了第一个字节，则会引起 RBNE 中断。块模式下，如果用 DMA 从智能卡读取数据，也只能在第一个字节接收好后再去使能 DMA。

在接收到第一个字节之后(RBNE 中断)必须将 RTR 寄存器设置为 CWT (字节等待时间) – 11 的值(这个时间以波特时间作为单位)，这是为了在两个连续的字符之间自动检测最大等待时间。如果智能卡在前一个字符发送结束后到设定的 CWT 周期之间没有发送字符，USART 会通过 RTF 标志提醒软件，当 RTIE 被置位时，会引起中断。

USART 用一个块长度计数器去统计收到的所有字符的长度。这个计数器在 USART 开始发送的时候自动清 0(TBE=0)。这个块长度信息位于智能卡发出数据的第三个字节(序言部分)。这个值必须写入 USART\_RTR 寄存器的 BL。当使用 DMA 模式时，在块开始之前，这个寄存器必须被设定为最小值(0x0)。为了得到这个值，在收到第四个字节后，会引起一个中断。软件必须从接收缓冲区读取第三个字节作为块长度。

在中断驱动接收模式，块的长度可以由软件提取出来并做检测或者通过设置 BL 的值得到。但是在块开始之前，BL(0xFF)可以被设置为最大值。实际值则要在接收到第三个字节后写到寄存器中。

整个块的长度(包括序言区，收尾区和信息区)等于 BL+4。块尾通过 EBF 标志和相应中断提醒给软件(当 EBIE 位置 1 时)。如果块长度出错，将会引起一个 RT 中断。

### 直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换

直接转换定义如下：低位在前，逻辑值位为 1 对应传输线高电平，采用偶校验。要使用这种转换方式，下列控制位必须被设置为：MSBF=0，DINV=0 (默认值)。

反向转换定义如下：高位在前，逻辑值位为 1 对应传输线低电平，采用奇校验。要使用这种转

换方式，下列控制为必须被设置为：MSBF=1，DINV=1

### 17.3.12. 串行红外（IrDA SIR）编解码功能模块

通过 USART\_CTRLR3 寄存器的 IREN 位置位，使能模式。在 IrDA 模式，USART\_CTRLR2 寄存器的 LMEN，STB，CKEN 位和 USART\_CTRLR3 寄存器的 HDEN，SCEN 位将被清 0。

按照 IrDA SIR 物理层的定义，一个红外光脉冲(RTZ 信号)代表逻辑‘0’。其脉冲宽度通常占一个位时间的 3/16。若脉冲宽度小于一个 1 个 PSC 时钟，IrDA 则无法检测到。如果脉冲宽度大于 1 但是小于 2 倍 PSC 时钟，IrDA 则无法可靠的检测到。

USART 数据帧由 SIR 发送编码器进行调制。调制后的信号经由红外 LED 进行发送。对于编码器而言，波特率应小于 115200。

红外检测器接收到调制后的信号并且输出解码后的数据帧。编码器所发送的调制信号的极性与解码器所接收信号的极性是相反的。解码器输入通常为高电平当输入为低脉冲时，解码得到一个起始位。

在 IrDA SIR ENDEC 模块发送和接收不得同时进行。

对于省电模式：

- 发送器：脉冲宽度可以为低功耗波特率的 3 倍
- 接收器：与正常模式相同

**图 17-7. IrDA SIR ENDEC 模块**

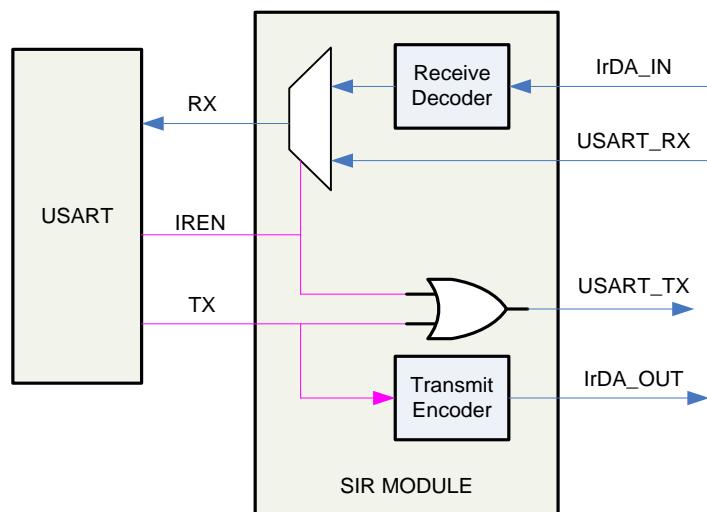
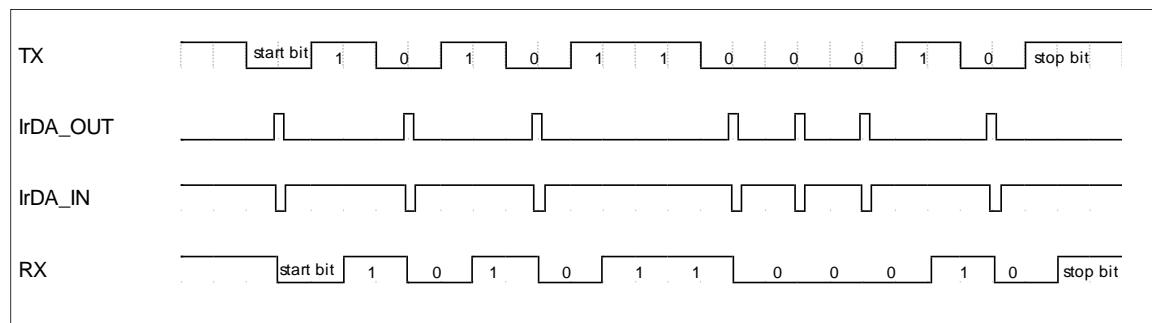


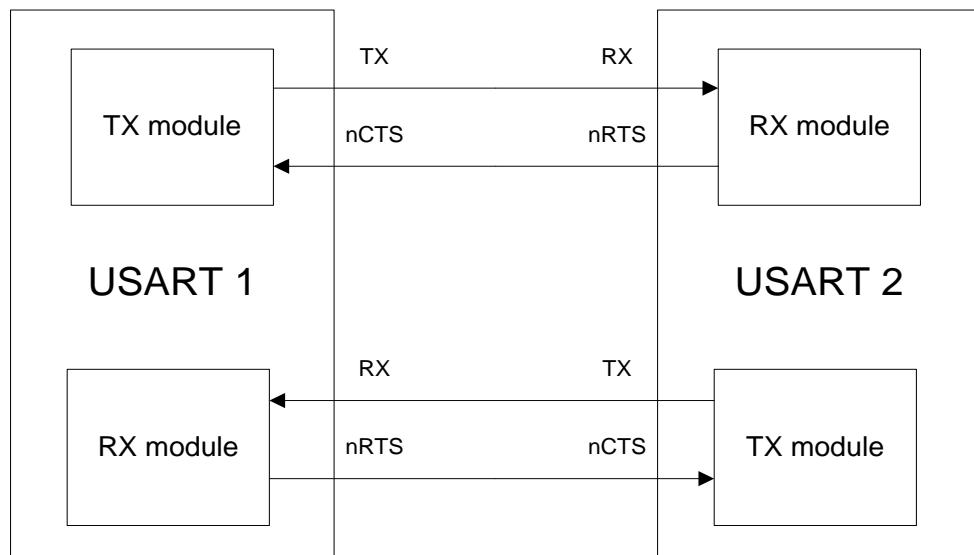
图 17-8. IrDA 数据调制



### 17.3.13. 硬件流控制

利用 nCTS 输入信号和 nRTS 输出信号，去控制串行数据流被称为硬件流控制。通过向 USART\_CTRLR3 寄存器的 RTSEN 位写‘1’使能 RTS 流控制，通过向 USART\_CTRLR3 寄存器的 CTSEN 位写‘1’使能 CTS 流控制。

图 17-9. 两个 USARTs 之间的硬件流控制



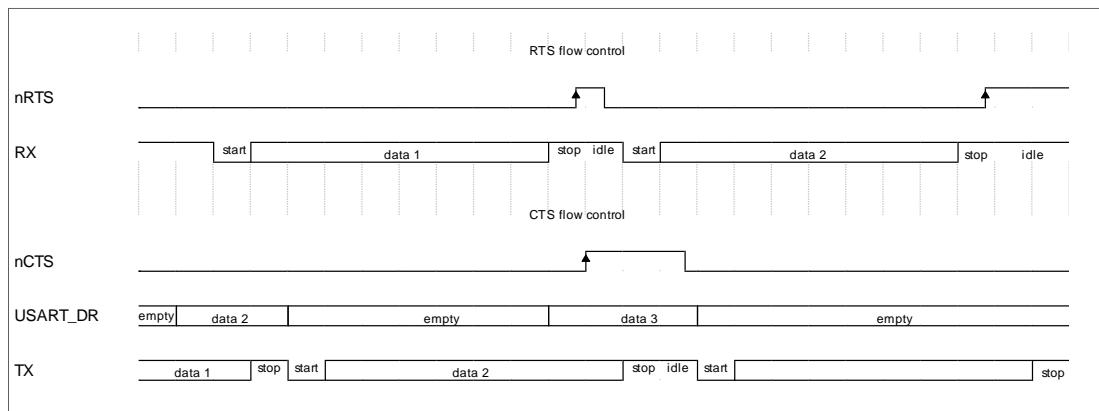
#### RTS 流控

仅当 nRTS 信号低电平时，USART 接收器才能接收数据。在接收期间，信号保持低电平不会变高。接收完成之后，电平变高。当 nRTS 信号再次变低时，开始下一次接受。如果接收寄存器已满，信号将保持高电平。

#### CTS 流控

如果 USART\_STR 寄存器的 TBE 位是‘0’并且 nCTS 信号是低电平，发送器发送数据帧。在发送期间，若 nCTS 信号变为高电平，发送器在发送完当前帧数据后停止发送。

图 17-10. 硬件流控



### RS485 驱动使能

驱动使能功能通过设置 **USART\_CTLR3** 控制寄存器的 **DEM** 位来打开。它允许用户通过 **DE** (Driver Enable)信号激活外部收发器控制。提前时间是驱动使能信号和第一个字节的起始位之间的时间间隔。这个时间可以在 **USART\_CTLR1** 控制器的 **DEA[4:0]** 位域进行设置。滞后时间是一个发送信息最后一个字节的停止位与释放 **DE** 信号之间的时间间隔。这个时间可以在 **USART\_CTLR1** 控制器的 **DED[4:0]** 位域进行设置。**DE** 信号的极性可以通过 **USART\_CTLR3** 控制寄存器的 **DEP** 位来设置。

#### 17.3.14. DMA 请求

DMA 可实现 USART 的连续通信。**USART\_CTLR3** 寄存器 **DENT** 位被用来使能 DMA 发送，**USART\_CTLR3** 寄存器的 **DENR** 被用来使能 DMA 接收。

DMA 发送配置如下：

1. 配置 DMA 寄存器，包括目标地址(**USART\_TDTR** 寄存器的地址)，源地址(存储器地址)，传输总字节数，通道优先级，DMA 中断等
2. 向 **USART\_SCR** 的 **TCC** 位写‘0’
3. 向 DMA 控制寄存器 **DENT** 位写‘1’以使能 DMA 通道

当 **DMA\_IFR** 寄存器的 **TCIF** 标志被置位后，之后 **USART\_STR** 寄存器的 **TC** 标志位才会置位。在数据发送期间，该标志保持为 0，USART 传输结束后置 1。当 **TC** 未置 1 时，进入深度睡眠模式或禁用 USART 将会导致传输被破坏。

**DMA 接收配置：**配置 DMA 寄存器，包括目标地址(**USART\_RDTR** 寄存器地址)，源地址(存储器)，传输总字节数，通道优先级，DMA 中断等

只要接收到了数据，就会触发 **RBNE** 事件

#### 17.3.15. 从 Deep-sleep 模式唤醒

通过标准 **RBNE** 中断或 **WUM** 中断 USART 能从深度睡眠模式唤醒 MCU。

UESM 位必须置 1 并且 USART 时钟必须设置为 HSI 或 LSE(请参考 RCC 部分)。

当使用 RBNE 标准中断时，必须在进入深度睡眠模式前将 RBNEIE 位置位。

当使用 WUIE 中断时，WUIE 中断源可以通过 WUM 位来选择。

在进入深度睡眠模式前，必须禁用 DMA。在进入深度睡眠模式前，软件必须检测 USART 没有正在传送数据。这可以通过 USART\_STR 寄存器中的 BSY 标志来判断。REA 位必须被检测以确保 USART 实际上是使能的。

当检测到唤醒事件时，无论 MCU 工作在深度睡眠模式还是正常模式，WUF 标志位被通过硬件置 1，并且在 WUIE 被置位的情况下，触发一个唤醒中断。

### 17.3.16. USART 中断

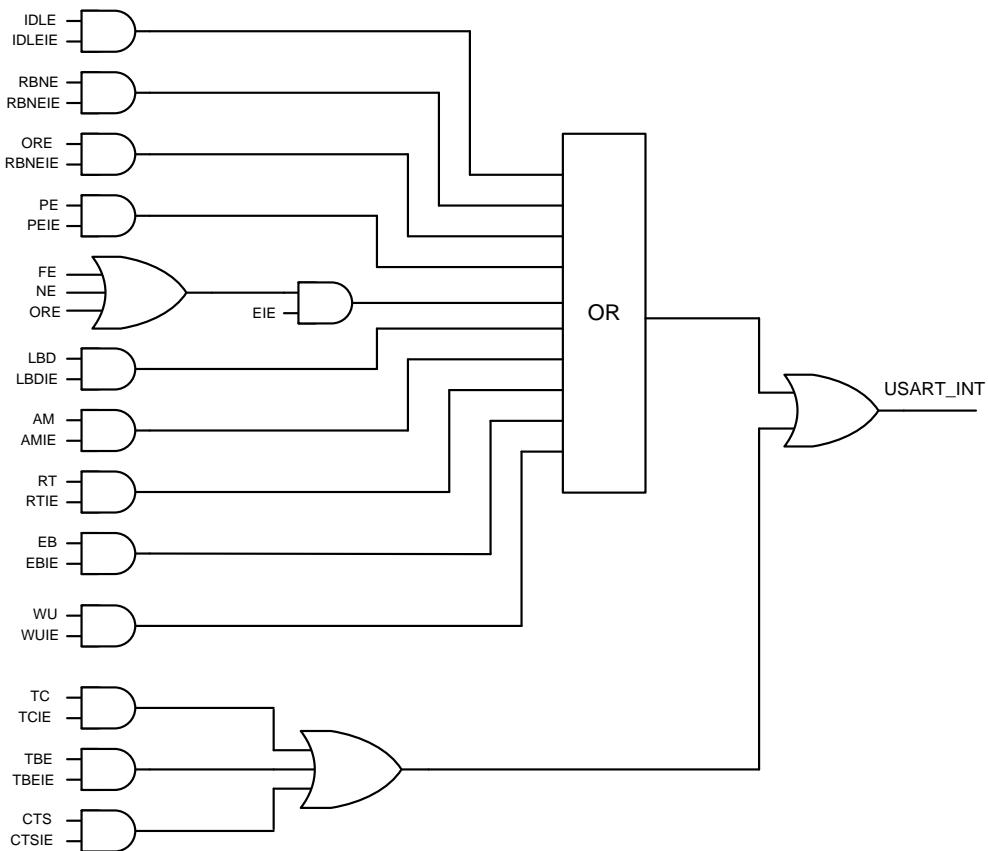
USART 中断事件和标志如下表所示：

表 17-3. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE
CTS 标志	CTSF	CTSIE
发送结束	TC	TCIE
接收到的数据可以读取	RBNE	RBNEIE
检测到过载错误	ORE	
检测到线路空闲	IDLEF	IDLEIE
奇偶校验错误	PE	PEIE
LIN 模式下，检测到断开标志	LBDF	LBDIE
DMA 接收时，出现接收错误(噪声标志，过载错误，帧格式错误)	NE or ORE or FE	EIE
字符匹配	AMF	AMIE
接收超时错误	RTF	RTIE
发现块尾	EBF	EBIE
从Deepsleep模式唤醒	WUF	WUIE

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 USART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

图 17-11. USART 中断映射框图



## 17.4. USART 寄存器

### 17.4.1. USART 控制寄存器 1 (USART\_CTLR1)

地址偏移: 0x00

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EBIE	RTIE	DEA[4:0]				DED[4:0]					
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM	AMIE	MEN	WL	WM	PCEN	PM	PEIE	TBEIE	TCIE	RBNEIE	IDIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31:28	保留	必须保持复位值
27	EBIE	块尾中断使能 0: 中断禁止 1: 中断使能 在USART2, 该位保留。
26	RTIE	接收超时中断使能 0: 中断禁止 1: 中断使能 在USART2, 该位保留。
25:21	DEA[4:0]	驱动使能置位时间 这些数字用来定义DE(驱动使能)信号的置位与第一个字节的起始位之间的时间间隔。 它以采样时间为单位(1/8或1/16位时间), 可以通过OM位来配置。 当USART被使能(UEN=1)时, 该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间 这些位用来定义一个发送信息最后一个字节的停止位与置低DE(驱动使能)信号之间的时间间隔。它以采样时间为单位(1/8或1/16位时间), 可以通过OM位来配置。 当USART被使能(UEN=1)时, 该位域不能被改写。
15	OM	过采样模式 0: 16倍过采样 1: 8倍过采样 在LIN、IrDA 和智能卡模式, 这些位保持清0。 当USART被使能(UEN=1)时, 该位域不能被改写。
14	AMIE	ADDR字符匹配中断使能 0: ADDR字符匹配中断禁用 1: ADDR字符匹配中断使能
13	MEN	MEN: 静默模式使能 0: 静默模式禁用 1: 静默模式被使能
12	WL	字长 0: 8 数据位, 1: 9 数据位 当USART被使能(UEN=1)时, 该位域不能被改写。
11	WM	从静默模式唤醒方法 0: 空闲线 1: 地址标记 当USART被使能(UEN=1)时, 该位域不能被改写。
10	PCEN	校验控制使能 0: 校验控制禁用 1: 校验控制被使能

当USART被使能(UEN=1)时，该位域不能被改写。

9	PM	校验模式 0: 偶校验 1: 奇校验 当USART被使能(UEN=1)时，该位域不能被改写。
8	PEIE	校验错误中断使能 0: 校验错误中断禁用 1: 当USART_STR寄存器的PE位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能 0: 中断禁止 1 : 当USART_STR寄存器的TBE位置位时，将触发中断。
6	TCIE	发送完成中断使能 0: 发送完成中断禁用 1: 当USART_STR寄存器的TC位置位时，将触发中断。
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能 0: 读数据缓冲区非空中断和过载错误中断禁用 1: 当USART_STR寄存器的ORE或RBNE位置位时，将触发中断。
4	IDIE	IDLE线检测中断使能 0: IDLE 线检测中断禁用 1: 当USART_STR寄存器的IDLEF位置位时，将触发中断。
3	TEN	发送器使能 0: 发送器关闭 1: 发送器打开
2	REN	接收器使能 0: 接收器关闭 1: 接收器打开并且开始搜索起始位。
1	UESM	USART在深度睡眠模式下使能 0: USART不能从深度睡眠模式唤醒MCU 1: USART能从深度睡眠模式唤醒MCU. 条件是USART的时钟源必须是HSI或LSE. 在USART2，该位保留
0	UEN	USART使能 0: USART预分频器和输出禁用 1: USART预分频器和输出被使能

### 17.4.2. USART 控制寄存器 2 (USART\_CTLR2)

地址偏移: 0x04

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								RTEN	ABDM[1:0]	ABDEN	MSBF	DINV	TINV	RINV	
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]	CKEN	CPL	CPH	LBCP	保留	LBDIE	LBDL	ADDM		保留			
rw	rw	rw	rw	rw	rw	rw		rw	rw	rw					

位/位域	名称	描述
31:24	ADDR[7:0]	<p>USART的节点地址</p> <p>这些位给出USART的节点地址。</p> <p>在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址标记的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM位被置位时，仅仅ADDR[3:0]被用来比较。</p> <p>在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符 (8-bit)与ADD[7:0]的值进行比较并且AMF标志设为正在匹配。</p> <p>当接收器(REN=1)和 USART (UEN=1) 被使能时，该位域不能被改写。</p>
23	RTEN	<p>接收器超时使能</p> <p>0: 接收器超时检测功能禁用</p> <p>1: 接收器超时检测功能被使能</p> <p>在USART2，该位保留。</p>
22:21	ABDM[1:0]	<p>自动波特率检测模式</p> <p>00: 下降沿到上升沿的测量 (测量开始位)</p> <p>01: 下降沿对下降沿的测量 (接收到的帧必须是一个这种格式的帧10xxxxxx)</p> <p>10: 保留</p> <p>11: 保留</p> <p>当USART被使能时，该位域不能改写(UEN=1)。</p> <p>在USART2，该位保留。</p>
20	ABDEN	<p>自动波特率检测使能</p> <p>0: 自动波特率检测禁用</p> <p>1: 自动波特率检测被使能</p> <p>在USART2，该位保留。</p>
19	MSBF	<p>高位在前</p> <p>0: 数据发送/接收，采用低位在前</p> <p>1: 数据发送/接收，采用高位在前</p> <p>USART被使能(UEN=1)时，该位域不能被改写。</p>
18	DINV	<p>数据位反转</p> <p>0: 数据位信号值没有反转</p> <p>1: 数据位信号值被反转</p> <p>USART被使能(UEN=1)时，该位域不能被改写。</p>

---

17	TINV	TX 管脚电平反转 0: TX管脚信号值没有反转 1: TX管脚信号值被反转 USART被使能(UEN=1)时，该位域不能被改写。
16	RINV	RX管脚电平反转 0: RX管脚信号值没有反转 1: RX管脚信号值被反转 USART被使能(UEN=1)时，该位域不能被改写。
15	STRP	交换TX/RX管脚 0: TX和RX管脚功能没被交换 1: TX和RX管脚功能被交换 当USART 被使能时，该位域不能被改写(UEN=1)。
14	LMEN	LIN模式使能 0: LIN模式关闭 1: LIN模式开启 USART被使能(UEN=1)时，该位域不能被改写。 在USART2，该位保留。
13:12	STB[1:0]	STOP位长 00: 1停止位 01: 保留 10: 2停止位 11: 1.5停止位 USART被使能(UEN=1)时，该位域不能被改写。
11	CKEN	CK管脚使能 0: CK管脚禁用 1: CK管脚被使能 USART被使能(UEN=1)时，该位域不能被改写。 在USART2，该位保留
10	CPL	时钟极性 0: 在同步模式下，CK管脚不对外发送时保持为低电平 1: 在同步模式下，CK管脚不对外发送时保持为高电平 USART被使能(UEN=1)时，该位域不能被改写。
9	CPH	时钟相位 0: 在同步模式下，在首个时钟边沿采样第一个数据 1: 在同步模式下，在第二个时钟边沿采样第一个数据 USART被使能(UEN=1)时，该位域不能被改写。
8	LBCP	末位时钟脉冲 0: 在同步模式下，最后一位(MSB)的时钟脉冲不输出到CK管脚 1: 在同步模式下，最后一位(MSB)的时钟脉冲输出到CK管脚

USART被使能(UEN=1)时，该位域不能被改写。

7	保留	必须保持复位值
6	LBDIE	<p>LIN断开信号检测中断使能 0: 断开信号检测中断禁用 1: 当USART_STR 的LBDF位置位，将产生中断。 在USART2, 该位保留</p>
5	LBDL	<p>LIN暂停帧长度 0: 检测10位暂停帧 1: 检测11位暂停帧 USART被使能(UEN=1)时，该位域不能被改写。 在USART2, 该位保留</p>
4	ADDM	<p>地址检测模式 这位用来选择4位地址检测还是全位地址检测 0: 4位地址检测 1: 全位地址检测在7位，8位和9位数据模式下，地址检测分别按 6位，7位和8位地址 (ADD[5:0], ADD[6:0] 和ADD[7:0]) 。 USART被使能(UEN=1)时，该位域不能被改写。</p>
3:0	保留	必须保持复位值

### 17.4.3. USART 控制寄存器 3 (USART\_CTLR3)

地址偏移: 0x08

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								WUIE	WUM[1:0]	SCRTNUM[2:0]			保留		
										rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSBM	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NACK	HDEN	IRLP	IREN	ERIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值
22	WUIE	<p>从深度睡眠模式唤醒中断使能 0: 从深度睡眠模式唤醒中断禁用 1: 从深度睡眠模式唤醒中断被使能 在USART2, 该位保留。</p>
21:20	WUM[1:0]	<p>从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位USART_ISR寄存器中的WU(从深度睡眠唤醒标志)</p>

		标志。
		00: WU在地址匹配的时候置位。如何实现地址匹配需要在ADDR和ADDM中定义。
		01: 保留。
		10: WU在检测到起始位时置位
		11: WU在检测到RBNE时置位
		USART被使能(UEN=1)时，该位域不能被改写。
		在USART2，该位保留。
19:17	SCRTNUM[2:0]	智能卡自动重试数目寄存器 在智能卡模式下，这些位用来指定在发送和接收时重试的次数。在发送模式下，它指的是在产生发送错误(FE bit set)之前自动重试的发送次数。 在接收模式下，它指的是在产生接收错误(RBNE位和PE位置位)之前自动重试的接收次数。 当这些位被设置为0x0时，在发送模式下这些位将不会自动发送。 USART被使能(UEN=1)时，该位域不能被改写。 在USART2，该位保留。
16	保留	必须保持复位值
15	DEP	驱动使能的极性选择模式 0: DE信号高有效 1: DE信号低有效 USART被使能(UEN=1)时，该位域不能被改写。
14	DEM	驱动使能模式 用户使能该位以后，可以通过DE信号对外部收发器进行控制。DE信号是从RTS 管脚输出的。 0: DE功能禁用 1: DE功能开启 USART被使能(UEN=1)时，该位域不能被改写。
13	DDRE	在接收错误时禁止DMA 0: 在发生接收错误的情况下，不禁用DMA。所有的错误数据不会产生DMA请求，以确保错误的数据不会被传输，但是下一个接收到的正确的数据会被传输。RBNE位保持0以阻止过载错误，但是相应的错误标志位会被置位。在智能卡模式下这种模式可以使用。 1: 在接收错误的情况下，DMA被关闭。 DMA请求会被屏蔽，直到相应的标志位被清0。RBNE标志和相应的错误标志位会被设置。软件在清错误标志之前，必须首先关DMA 请求(DMAR = 0) 或清RBNE。 USART被使能(UEN=1)时，该位域不能被改写。
12	OVRD	溢出禁止 0: 溢出功能被使能。当接收到的数据在新数据到达前没有被读走，ORE错误标志位将被置位，并且新数据将会丢失 1: 溢出功能禁止。当接收到的数据在新数据到达前没有被读走，ORE错误标志位将不会被置位，新数据会将USART_RDTR寄存器以前的内容覆盖

USART被使能(UEN=1)时，该位域不能被改写。

11	OSBM	单次采样方式 0: 三次采样方法 1: 一次采样方法 USART被使能(UEN=1)时，该位域不能被改写。
10	CTSIE	CTS中断使能 0: CTS中断屏蔽 1: 当USART_STR寄存器的CTS位置位时，会产生中断。
9	CTSEN	CTS中断使能 0: CTS 硬件流控禁用 1: CTS 硬件流控被使能 USART被使能(UEN=1)时，该位域不能被改写。
8	RTSEN	RTS使能 0: RTS硬件流控禁用 1: RTS硬件流控被使能，只有当接收缓冲区有空间的时候，才会请求下一个数据。 USART被使能(UEN=1)时，该位域不能被改写。
7	DENT	DMA发送使能 0: 关闭DMA发送模式 1: 开启DMA发送模式
6	DENR	DMA接收使能 0: 关闭DMA接收模式 1: 开启DMA接收模式
5	SCEN	智能卡模式使能 0: 智能卡模式禁用 1: 智能卡模式使能 USART被使能(UEN=1)时，该位域不能被改写)。 在USART2中，该位保留。
4	NACK	在智能卡模式NACK使能 0:当出现校验错误时不发送NACK 1: 当出现校验错误时发送NACK USART被使能(UEN=1)时，该位域不能被改写)。 在USART2中，该位保留。
3	HDEN	半双工使能 0: 禁用半双工模式 1: 开启半双工模式 USART被使能(UEN=1)时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0: 正常模式 1: 低功耗模式

USART被使能(UEN=1)时，该位域不能被改写。

1	IREN	IrDA 模式使能 0: IrDA 禁用 1: IrDA 被使能 USART被使能(UEN=1)时，该位域不能被改写。 在USART2中，该位保留。
0	ERIE	多级缓存通信模式的错误中断使能 0: 禁用错误中断 1: 在多级缓存通信时，当USART_STR寄存器的FE位，ORE位或NE位被置位时，会产生中断。

#### 17.4.4. USART 波特率寄存器 (USART\_BRR)

地址偏移: 0xC

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

当USART(UEN=1)被使能时，这一寄存器不能改写。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	BRR[11:0]	波特率分频系数的整数部分 $DIV\_INT[11:0] = BRR[15:4]$
3:0	BRR [3:0]	波特率分频系数的小数部分 如果OM = 0, USARTDIV [3:0] = BRR [3:0]; 如果OM = 1, USARTDIV [3:1] = BRR [2:0], BRR [3]必须被置0。

#### 17.4.5. USART 保护时间和预分频器寄存器 (USART\_GTPR)

地址偏移: 0x10

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

USART被使能(UEN=1)时，这一位域不能被改写  
在USART2中，这一寄存器保留

31	30	29	28	27	26	25	24	23	22	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								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	GT[7:0]	在智能卡模式下的保护时间值
7:0	PSC[7:0]	<p>对系统时钟进行的预分频值</p> <p>在红外低功耗模式下，对系统时钟进行分频以获得低功耗模式下的频率。寄存器的值是分频系数。</p> <p>00000000: 保留 – 不设置这个值</p> <p>00000001: 1分频</p> <p>00000010: 2分频</p> <p>...</p> <p><b>在红外正常模式下</b></p> <p>00000001: 仅仅能设为这个值</p> <p><b>在智能卡模式下</b>，预分频值被存储在PSC[4:0]中，PSC[7:5]必须保持复位值。分频系数是寄存器中值的两倍。</p> <p>00000: 保留 -不设置这个值</p> <p>00001: 2分频</p> <p>00010: 4分频</p> <p>00011: 6分频</p> <p>...</p>

#### 17.4.6. USART 接收超时寄存器 (USART\_RTR)

地址偏移: 0x14

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

在USART2中，这一寄存器保留

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

位/位域	名称	描述
31:24	BL[7:0]	<p>块长度</p> <p>这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度(1-LEC/2-CRC) - 1.</p> <p>这个值可以在块接收开始去设置(用于需要从块的序言提取块的长度的情形)，这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。</p> <p>在其他模式下，当REN=0 (禁用接收器)并且/或者当EOBCF 位被写1时块的长度计数器被清0。</p>
23:0	RT[23:0]	<p>接收器超时门限</p> <p>该位域指定接收超时值，单位是波特时钟的时长</p> <p>标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，RTF标志被置位</p> <p>在智能卡模式，这个值被用来实现CWT和BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始算的</p> <p>这些位可以在工作时改写。假如一个新数据到来的时间比RT规定的晚，RTF标志会被置位。对于每个接收字符，这个值只能改写一次。</p>

#### 17.4.7. USART 请求寄存器 (USART\_CMD)

地址偏移: 0x18

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								TXFC MD	RXFC MD	MMC MD	SBKC MD	ABDC MD			
w								w	w	w	w	w			

位/位域	名称	描述
31:5	保留	必须保持复位值
4	TXFCMD	<p>发送数据清空请求</p> <p>向该位写1去置位TBE标志位，取消发送数据。</p> <p>在USART2中，该位保留。</p>
3	RXFCMD	<p>接收数据清空请求</p> <p>向该位写1来清除RBNE标志位，在读接收数据的情况下丢弃该数据。</p>
2	MMCMD	<p>静默模式请求</p> <p>向该位写1使USART进入静默模式并且置位RWU标志位。</p>

---

1	SBKCMD	发送断开帧请求 向该位写1置位SBKF标志并使USART在空闲时发送一个断开帧。
0	ABDCMD	自动波特率检测请求 向该位写1会清除USART_ISR寄存器的ABRF位，并且在下一个数据接收帧开始自动检测波特率。 在USART2中，该位保留。

#### 17.4.8. USART 状态寄存器 (USART\_STR)

地址偏移: 0x1C

复位值: 0x0000\_00C0

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														REA	TEA
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABDF	ABDE	保留	EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORE	NE	FE	PE
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值
22	REA	接收使能通知标志 这位反映了USART核心逻辑的接收使能状态，该位可以通过硬件设置。 0: USART核心接收逻辑禁用 1: USART核心接收逻辑被使能
21	TEA	发送使能通知标志 这位反映了USART核心逻辑的发送使能状态，该位可以通过硬件设置。 0: USART核心发送逻辑禁用 1: USART核心发送逻辑被使能
20	WUF	从深度睡眠模式唤醒标志 0: 没有从深度睡眠模式唤醒 1: 已从深度睡眠模式唤醒，如果在USART_CTLR3寄存器的WUFIE=1并且MCU处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，这一位通过硬件置位，这个事件在WUS位域被定义。 当UESM被清0时，该位清0。 在USART2中，该位保留
19	RWU	接收器从静默模式唤醒 这位表示USART处于静默模式。 0: 接收器在工作状态

		1: 接收器在静默状态 当在唤醒和静默模式切换时，它通过硬件清0或者置1。静默模式控制（地址帧还是空闲帧）是用USART_CTLR1寄存器的WAKE位来选择的。 如果选择空闲信号唤醒，只能通过向USART_CMD寄存器的MMCMD位写1来置位该位了。
18	SBF	断开信号发送标识 0: 没发送断开字符 1: 将要发送断开字符 该位表示一个断开发送信号被请求。 通过向USART_CMD寄存器的SBKCMD写1来置位。 当断开帧的停止位发送期间，硬件清0。
17	AMF	ADDR匹配标志 0: ADDR和接收到的字符不匹配 1: ADDR和接收到的字符匹配，如果USART_CTLR1寄存器的AMIE=1，将引发一个中断。 当接收到ADDR [7:0]中定义的字符时，硬件置位。 通过向USART_SCR寄存器的AMC写1清0。
16	BSY	忙标志 0: USART处于空闲 1: USART正在接收
15	ABDF	自动波特率检测标志 0: 没有自动波特率检测完成 1: 自动波特率检测完成 当自动波特率检测完成时，硬件置位。 通过向USART_CMD寄存器的ABDCMD写1清0，并请求新的波特率检测。 在USART2中，该位保留。
14	ABDE	自动波特率检测错误 0: 没有自动波特率检测错误出现 1: 自动波特率检测错误出现 如果波特率超出范围或者字符对比失败，硬件置位。 通过向USART_CTLR3寄存器的ABDRQ位写1清0。 在USART2中，该位保留。
13	保留	必须保持复位值
12	EBF	块结束标志 0: 块没有结束 1: 块结束已经到了(足够的字节数)，如果USART_CTLR2寄存器的EBIE=1，将引发一个中断 当接收到的字节数(从块开始，包括序言部分)等于或大于BLEN + 4，硬件置位。 通过向USART_SCR寄存器的EBC写1清0。 在USART2中，该位保留。

11	RTF	<p>接收超时标志</p> <p>0: 尚未超时</p> <p>1: 已经超时, 如果USART_CTLR2寄存器的RTIE被置位, 将会引发中断。</p> <p>如果空闲的时间已经超过了在RTR寄存器中设定的RT值, 通过硬件置1。</p> <p>通过向USART_SCR寄存器的RTC位写1清0。</p> <p>在智能卡模式, 这个超时相当于CWT或BWT计时。</p> <p>在USART2中, 该位保留。</p>
10	CTS	<p>CTS电平</p> <p>该值等于nCTS输入脚电平的反向拷贝。</p> <p>0: nCTS输入脚高电平</p> <p>1: nCTS输入脚低电平</p>
9	CTSF	<p>CTS变化标志</p> <p>0: nCTS 状态线没有变化</p> <p>1: nCTS 状态线发生变化 如果USART_CTLR3寄存器的CTSIE位置位, 将引发中断。</p> <p>当nCTS 输入变化时, 由硬件置位。</p> <p>通过向USART_SCR寄存器的CTSC位写1, 清零该位。</p>
8	LBDF	<p>LIN 断开检测标志</p> <p>0: 没有检测到LIN断开字符</p> <p>1: 检测到LIN断开字符。当USART_CTLR2寄存器的LBDIE位被置位时, 将会有中断产生。.</p> <p>当LIN断开帧被检测到的时候, 硬件置位。</p> <p>通过向USART_SCR寄存器的LBDC位写1, 清零该位。</p> <p>在USART2中, 该位保留。</p>
7	TBE	<p>发送数据寄存器空</p> <p>0: 数据没有发送到移位寄存器</p> <p>1: 数据发送到移位寄存器。如果USART_CTLR1寄存器的TBEIE位置位, 将会有中断产生。</p> <p>当USART_TDTR 寄存器的内容已经被转移到移位寄存器或者向USART_CMD 寄存器的TXFCMD位写1时, 由硬件置位。</p> <p>通过向USART_TDTR寄存器中写数据来清0。</p>
6	TC	<p>发送完成</p> <p>0: 发送没有完成</p> <p>1: 发送完成.如果USART_CTLR1寄存器的TCIE被置位, 将会有中断产生</p> <p>如果一个包含数据的帧的发送完成而且TBE位被置位, 由硬件置位。</p> <p>通过向USART_SCR寄存器的TCC位写1清0。</p>
5	RBNE	<p>读数据缓冲区非空</p> <p>0: 没有接收到数据</p> <p>1: 已接收到数据并且可以读取。当寄存器USART_CTLR1的RBNEIE位被置位, 将会有中断产生。</p> <p>当接收移位寄存器的内容已经被转移到寄存器USART_RDTR, 由硬件置位。</p>

通过读USART\_RDTR寄存器或向USART\_CMD寄存器的RXFCMD位写1清0。

4	IDLEF	<p>空闲线检测标志</p> <p>0: 没检测到空闲线</p> <p>1: 检测到空闲线。,如果USART_CTLR1寄存器的IDLEIE位置1, 将会有中断产生。</p> <p>当检测到空闲线时, 通过硬件置位。直到RBNE位置位, 它不会再次置位。</p> <p>向USART_SCR寄存器的IDLEC位写1清0。</p>
3	ORE	<p>溢出错误</p> <p>0: 没有检测到溢出错误</p> <p>1: 检测到溢出错误。在多级缓存通信中, 如果寄存器USART_CTLR1的RBNEIE位置位, 将会引发中断。如果寄存器USART_CTLR3的ERIE位置位也会引发中断。</p> <p>在RBNE置位的情况下, 如果接收移位寄存器的数据传递给USART_RDTR寄存器, 将会由硬件置位。</p> <p>向USART_SCR寄存器的OREC位写1清0。</p>
2	NE	<p>噪声错误标志</p> <p>0: 没检测到噪声错误</p> <p>1: 检测到噪声错误。在多级缓存通信中, 如果寄存器 USART_CTLR3的ERIE位置位, 将会有中断产生。</p> <p>在接收帧的时候检测到噪声, 将会由硬件置位。</p> <p>向寄存器USART_SCR的NEC位写1清0。</p>
1	FE	<p>帧错误</p> <p>0: 没检测到帧错误</p> <p>1: 检测到帧错误或者断开字符。在多级缓存通信中, 如果寄存器 USART_CTLR3的ERIE位置位, 将会有中断产生。</p> <p>当一个不同步, 强噪声或者断开字符被检测到时, 在智能卡模式下, 当发送次数达到上限, 仍然没有收到成功发送应答 (卡一直响应NACKs), 该位也将被置位。</p> <p>向USART_SCR寄存器的FEC位写1清0。</p>
0	PE	<p>校验错误</p> <p>0: 没检测到校验错误</p> <p>1: 检测到校验错误 在多级缓存通信中, 如果寄存器 USART_CTLR1的PEIE位置位, 将会有中断产生。</p> <p>当在接收模式的时候检测到校验错误, 将会由硬件置位。</p> <p>向USART_SCR寄存器的PEC位写1清0。</p>

## 17.4.9. USART 状态标志清除寄存器 (USART\_SCR)

地址偏移: 0x20

复位值: 0x0000\_0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	EBC	RTC	保留	CTSC	LBDC	保留	TCC	保留	IDLEC	OREC	NEC	FEC	PEC	W	W

W                    W                    W                    W                    W                    W                    W                    W                    W                    W                    W                    W                    W                    W

位/位域	名称	描述
31:21	保留	必须保持复位值
20	WUC	从深度睡眠模式唤醒标志的清除 向这一位写1清0USART_STR寄存器的WUF位。 在USART2中，该位保留。
19:18	保留	必须保持复位值
17	AMC	ADDR匹配标志清除清除 向该位写1清0USART_STR寄存器的AM位。
16:13	保留	必须保持复位值
12	EBC	块结束标志清除 向该位写1清0 USART_STR寄存器的EB位。 在USART2中，该位保留。
11	RTC	接收超时标志清除 向这一位写1清0 USART_STR寄存器的RT标志。 在USART2中，该位保留。
10	保留	必须保持复位值
9	CTSC	CTS变化标志清除 向该位写1清0 USART_STR寄存器的CTSF位。
8	LBDC	LIN断开字符检测标志清除 向该位写1清0 USART_STR寄存器的LBDF标志位。 在USART2中，该位保留。
7	保留	必须保持复位值
6	TCC	发送完成标志清除 向该位写1清0 USART_STR寄存器的TC位。
5	保留	必须保持复位值
4	IDLEC	空闲线检测标志清除 向该位写1清0 USART_STR寄存器的IDLE位。
3	OREC	溢出标志清除 向该位写1清0 USART_STR寄存器的ORE位。
2	NEC	噪声检测清除

向该位写1清0USART\_STR寄存器的NE位。

- |   |            |   |
|---|------------|---|
| 1 | <b>FEC</b> | 帧格式错误标志清除<br>向该位写1清0 USART_STR 寄存器的FE位。 |
| 0 | <b>PEC</b> | 校验错误标志清除<br>向该位写1清0 USART_STR寄存器的PE位。   |

#### 17.4.10. USART 数据接收寄存器 (USART\_RDTR)

地址偏移: 0x24

复位值: 未定义

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								RDTR[8:0]							

r

位/位域	名称	描述
31:9	保留	必须保持复位值
8:0	<b>RDTR[8:0]</b>	接收数据的值 包含接收到的数据字节。 如果接收到的数据打开了奇偶校验位(USART_CTLR1寄存器的PE置1), 那么接收到的数据的最高位(第7位或8位, 取决于数据的长度)是奇偶校验位。

#### 17.4.11. USART 数据发送寄存器 (USART\_TDTR)

地址偏移: 0x28

复位值: 未定义

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								TDTR[8:0]							

rw

位/位域	名称	描述
31:9	保留	必须保持复位值

8:0

TDTR[8:0]

发送数据的值

包含发送到的数据字节。

如果发送到的数据打开了奇偶校验位(**USART\_CTLR1**寄存器的**PE**置1), 那么发送的数据的最高位(第7位或8位取决于数据的长度)将会被奇偶校验位替代。

只有当**USART\_STR**寄存器的**TBE**位被置位时, 该寄存器才可以改写。

## 18. MCU 调试 (MCUDBG)

### 18.1. 简介

MCUDBG模块可帮助调试器调试省电模式、定时器、I2C、RTC、WWDG、IWDG与bxCAN。当相应位置位，在省电模式下提供时钟或保持TIMER，WWDG、IWDG、RTC、I2C或bxCAN的当前状态（bxCAN模块的内容仅针对GD32F170xx和GD32F190xx产品）。

### 18.2. 功能描述

#### 18.2.1. 省电模式的调试支持

当MCUDBG控制寄存器1(MCUDBG\_CTRLR1)的STDBY\_HOLD位置位，并进入待机模式，AHB总线时钟和系统时钟是由CK\_HSI提供，并且调试器可以在待机模式下进行调试。当退出待机模式时，产生系统复位。

当MCUDBG控制寄存器1(MCUDBG\_CTRLR1)的DEEPSLEEP\_HOLD位置位，并进入深度睡眠模式，AHB总线时钟和系统时钟是由CK\_HSI提供，并且调试器可以在深度睡眠模式下进行调试。

当MCUDBG控制寄存器1(MCUDBG\_CTRLR1)的SLEEP\_HOLD位置位，并进入睡眠模式，CPU的AHB总线时钟未关闭，并且调试器可以在睡眠模式下进行调试。

#### 18.2.2. 定时器，I2C，RTC，WWDG，IWDG 与 CAN 的调试支持

当内核停止，并且 MCUDBG 控制寄存器 1(MCUDBG\_CTRLR1) 或 MCUDBG 控制寄存器 2(MCUDBG\_CTRLR2) 的相应的位置位，将如下动作：

对于定时器，定时器计数器停止并进行调试

对于I2C，SMBUS保持状态进行调试

对于WWDG或IWDG，计数器时钟停止进行调试

对于RTC，计数器停止进行调试

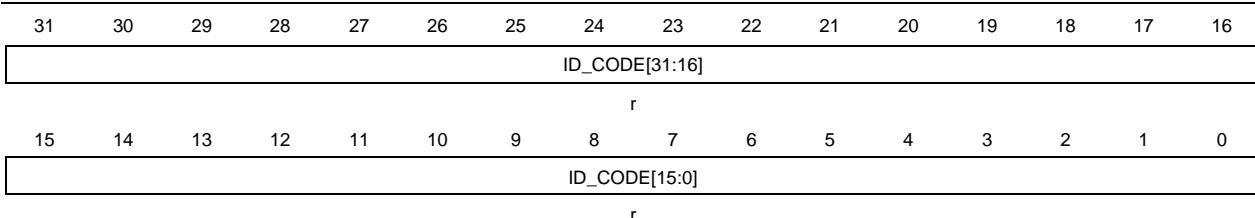
对于CAN，接收寄存器停止计数进行调试

### 18.3. MCUDBG 寄存器

#### 18.3.1. MCUDBG ID 寄存器 (MCUDBG\_IDR)

地址偏移：0x00

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:0	ID_CODE[15:0]	MCUDBG ID 寄存器 这些位由软件读取，这些位是不变的常数。

### 18.3.2. MCUDBG 控制寄存器 1 (MCUDBG\_CTRLR1)

**GD32F130xx 和 GD32F150xx 产品：**

地址偏移: 0x04

复位值: 0x0000 0000, 仅上电复位

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:28	保留	必须保持复位值
27	TM14_HOLD	Timer 14 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 14 计数器不变, 用于调试
26:20	保留	必须保持复位值
19	TM6_HOLD	Timer 6 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 6 计数器不变, 用于调试
18	保留	必须保持复位值
17	I2C3_HOLD	I2C3 保持寄存器 该位由软件置位和复位。

		0: 无影响 1: 当内核停止时保持 I2C3 SMBUS 状态不变, 用于调试
16	I2C2_HOLD	I2C2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 I2C2 SMBUS 状态不变, 用于调试
15	I2C1_HOLD	I2C1 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 I2C1 SMBUS 状态不变, 用于调试
14:13	保留	必须保持复位值
12	TM3_HOLD	Timer 3 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 3 计数器不变, 用于调试
11	TM2_HOLD	Timer 2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 2 计数器不变, 用于调试
10	TM1_HOLD	Timer 1 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 1 计数器不变, 用于调试
9	WWDG_HOLD	WWDG 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 wwdg counter 不变, 用于调试
8	IWDG_HOLD	IWDG 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 iwdg counter 不变, 用于调试
7:3	保留	必须保持复位值
2	STDBY_HOLD	待机模式保持寄存器 该位由软件置位和复位。 0: 无影响 1: 在待机模式下, 系统时钟和 HCLK 由 CK_HSI 提供, 当退出待机模式时, 产生系统复位
1	DEEPSLEEP_HOLD	深度睡眠模式保持寄存器

该位由软件置位和复位。

0: 无影响

1: 在深度睡眠模式下, 系统时钟和 HCLK 由 CK\_HSI 提供

0	SLEEP_HOLD	睡眠保持寄存器 该位由软件置位和复位 0: 无影响 1: 在睡眠模式下, HCLK 继续运行
---	------------	---

GD32F170xx 和 GD32F190xx 产品

地址偏移: 0x04

复位值: 0x0000 0000, 仅仅上电复位

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				TIMER14_ HOLD	保留				CAN2_ HOLD	保留	TIMER6_ HOLD	保留	I2C3_ HOLD	I2C2_ HOLD	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1_ HOLD	CAN1_ HOLD	保留	TIMER3_ HOLD	TIMER2_ HOLD	TIMER1_ HOLD	WWDG_ HOLD	IWDG_ HOLD	保留				STDBY_ HOLD	DEEPSLEEP_ HOLD	SLEEP_ HOLD	
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	

位/位域	名称	描述
31:28	保留	必须保持复位值
27	TIMER14_HOLD	Timer 14 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 14 计数器不变, 用于调试
26:22	保留	必须保持复位值
21	CAN2_HOLD	CAN 2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 CAN 2 计数器不变, 用于调试
20	保留	必须保持复位值
19	TIMER6_HOLD	Timer 6 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 6 计数器不变, 用于调试
18	保留	必须保持复位值
17	I2C3_HOLD	I2C3 保持寄存器

		该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 I2C3 SMBUS 状态不变, 用于调试
16	I2C2_HOLD	I2C2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 I2C2 SMBUS 状态不变, 用于调试
15	I2C1_HOLD	I2C1 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 I2C1 SMBUS 状态不变, 用于调试
14	CAN1_HOLD	CAN 1 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 CAN 1 计数器不变, 用于调试
13	保留	必须保持复位值
12	TIMER3_HOLD	Timer 3 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 3 计数器不变, 用于调试
11	TIMER2_HOLD	Timer 2 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 2 计数器不变, 用于调试
10	TIMER1_HOLD	Timer 1 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 1 计数器不变, 用于调试
9	WWDG_HOLD	WWDG 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 wwdg counter 不变, 用于调试
8	IWDG_HOLD	IWDG 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 iwdg counter 不变, 用于调试
7:3	保留	必须保持复位值
2	STDBY_HOLD	待机模式保持寄存器

该位由软件置位和复位。

0: 无影响

1: 在待机模式下, 系统时钟和 HCLK 由 CK\_HSI 提供, 当退出待机模式时, 产生系统复位

1	DEEPSLEEP_HOLD	深度睡眠模式保持寄存器 该位由软件置位和复位。 0: 无影响 1: 在深度睡眠模式下, 系统时钟和 HCLK 由 CK_HSI 提供
0	SLEEP_HOLD	睡眠保持寄存器 该位由软件置位和复位。 0: 无影响 1: 在睡眠模式下, HCLK 继续运行

### 18.3.3. MCUDBG 控制寄存器 2 (MCUDBG\_CTLR2)

地址偏移: 0x08

复位值: 0x0000 0000, 仅仅上电复位

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留														TIMER17_HOLD	TIMER16_HOLD	TIMER15_HOLD
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留					RTC_HOLD	保留										

rw

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER17_HOLD	Timer 17 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 17 计数器不变, 用于调试
17	TIMER16_HOLD	Timer 16 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 16 计数器不变, 用于调试
16	TIMER15_HOLD	Timer 15 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持定时器 15 计数器不变, 用于调试

---

15:11	保留	必须保持复位值
10	RTC_HOLD	<p>RTC 保持寄存器 该位由软件置位和复位。 0: 无影响 1: 当内核停止时保持 RTC 计数器不变, 用于调试</p>
9:0	保留	必须保持复位值

## 19. 通用串行总线全速设备接口(USB 2.0 FS)

本章内容适用于 GD32F150xx 产品，其主要描述了 USB 设备控制器及其接口。由于此模块的特性与其他串口不同，其复杂性较高，所以要求开发人员必须对其有所了解。如果开发人员需要了解相关的最新信息，请参考由 USB 设计论坛（USB Implementers Forum, USB-IF）出版的官方规范的 USB 2.0 版本。

### 19.1. 简介

通用串行总线（USB）是一个四线制总线，其主要使用电源线VBus、数据线D+(DP)、数据线D-(DM)和地线GND这4根线，其能够支持主机和USB控制器实现的功能设备之间的通信。USB主机通过基于令牌的协议分配USB带宽到各个连接的设备，通过轮询的方式进行数据通讯。USB总线支持设备的热拔插和动态配置。所有的交互过程均由主机发起。

USB主机和设备之间的每一次传输都是通过事务完成的，事务可分为3种类型：Setup事务，Data IN事务和Data Out事务，而事务又由3种数据包组成：令牌包，数据包和可选的握手包。

USB设备接口实现了全速USB 2.0总线和APB1总线之间的通讯，其可以处理所有USB数据包：其按照USB标准请求检测令牌包、处理数据包的传输并进行握手包的处理。事务的格式化是通过USB控制器硬件处理的，包括CRC的生成与校验。按照USB协议，每个设备最多有16个逻辑端点或者32个物理端点，而USB控制器支持最多8个双向端点或者16个单向端点。

每个USB端点都有一个对应的缓冲区，这个缓冲区位于一个专用的512字节SRAM中。主机PC端和系统存储器之间的数据传递均通过这个数据缓冲区实现。端点进行数据的收发必须等到端点状态可用才可以在自己的缓冲区中进行。对于要求速率稳定的同步传输和要求高吞吐量的批量传输，USB控制器提供了特殊的支持。它实现了一个双缓冲用法，这允许同步端点或者批量端点可以连续地收发数据而不用等待端点状态变为可用。因为对于USB设备而言这些端点总是有一个可用的缓冲去处理而微处理器同时可以使用另外一个缓冲。

在任何时候，只要有需要都可以通过写USB控制寄存器（USB\_CTLR）使USB模块进入低功耗模式（挂起模式）。此时，应该避免所有静态功耗并降低或者关闭USB时钟。在低功耗模式下，当检测到USB总线上有活动时将恢复到正常模式。

### 19.2. 主要特性

- USB 2.0全速设备控制器
- 支持USB 2.0链接电源管理（LPM）
- 最多支持8个可配置的端点
- 支持双缓冲的批量传输端点/同步传输端点
- 端点都支持控制、批量、同步和中断传输类型（端点0除外）
- 支持挂起/恢复操作

- 一个512字节的专用SRAM用于数据包缓冲
- 集成的USB物理层

### 19.2.1. 实现

表19-1描述了在GD32F150xx设备上的USB实现

表 19-1. GD32F150xx USB 实现

USB 特征	GD32F1x0 USB
端点数	8 个双向/16 个单向端点
专用缓冲区大小 (SRAM)	512 字节
专用数据缓冲区访问方式	2×16 比特每字
USB 2.0 连接电源管理 (LPM) 支持	是

### 19.2.2. 时钟

按照USB标准定义，USB全速模块使用固定的48MHz时钟。使用USB模块时，需要打开两个时钟，一个是USB控制器时钟，需要配置其频率为48MHz，另一个是USB的APB1接口时钟，其实际为当前的APB1总线时钟，其频率可以大于或者小于48MHz。

**注意：**为了满足USB数据传输率和分组缓冲区接口的系统需求，APB1总线时钟的频率必须大于24MHz，以避免数据缓冲区的上下溢出。

USB控制器的48MHz时钟可以通过MCU的内部晶振或者外部晶振经过PLL倍频得到：

- 8MHz的内部晶振2分频后作为PLL的输入，再进行12倍频得到
- 8MHz的外部晶振直接作为PLL的输入，先进行倍频，再经过USB分频器分频得到

当通过外部晶振产生USB时钟时，需要注意的是USB分频系数只有4个值：1分频、1.5分频、2分频和2.5分频（用不到，MCU主频无法达到120MHz）。所以，为了获得48MHz的时钟，PLL倍频后只能为48MHz、72MHz或者96MHz。

**注意：**无论使用外部晶振还是内部晶振产生USB时钟，其时钟准确度都必须要达到±500ppm。如果USB时钟的准确度下降，传输数据可能无法满足USB规范要求，甚至直接导致USB无法运行。

### 19.2.3. 管脚描述

USB控制器以设备模式运行时总共要求使用3个管脚：DP，DM和VBus。管脚DP与DM为数据管脚，主要用来收发数据，在控制器上是固定设置的。在使能USB模块之前，这些管脚在复位后默认为普通的GPIO，一旦使能USB后便成为USB管脚。当USB设备为总线供电设备时，VBus管脚主要用来为控制器供电，当为自供电设备时没有被用到。

管脚DP和DM是不需要配置的，因为一旦USB模块被使能，他们会自动连接到USB内部的收发器上。

信号管脚如下表所示：

**表 19-2. GD32F150xx USB 信号管脚**

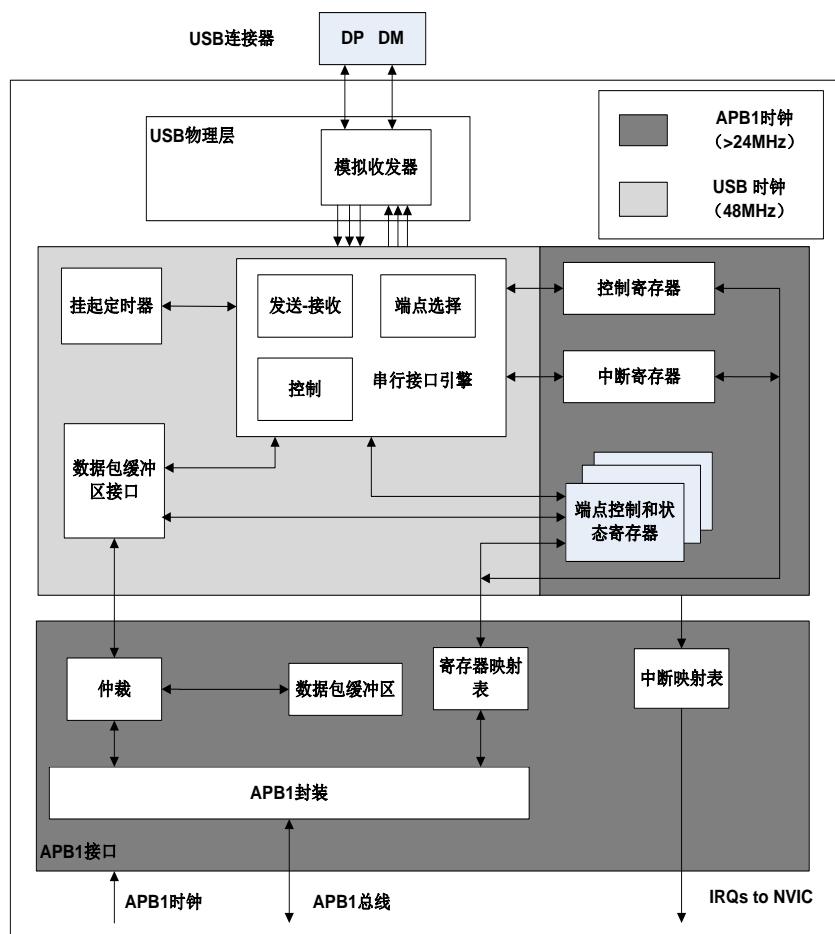
USB管脚	GPIO引脚	GPIO配置
USBDP	PA12	一旦USB被使能，这些引脚自动连接到USB内部的收发器上
USBDM	PA11	

## 19.3. 功能描述

### 19.3.1. 基本功能模块

图19-1显示了USB模块图

**图 19-1.USB 模块图**



### 控制器模块

USB控制器包括以下的模块：

- 内置模拟收发器（ATX）：USB ATX连到USB引脚DP和DM上，能够发送和接收USB总线上的双向差分信号。

- **串行接口引擎 (SIE) :** SIE实现全速USB协议层。考虑到速度问题，它完全是硬件实现的，不需要任何固件干涉。SIE主要负责处理端点缓冲和USB总线之间的数据传输。该模块的功能包括：编解码串行数据并进行误差修正、位填充/解除填充、同步模式识别、CRC校验/产生、PID验证/生成、并行/串行转换、地址识别和握手信号的评估/生成。
- **挂起定时器：** 挂起定时器产生起始帧锁定时钟脉冲并当USB总线上3ms没有活动时检测全局性挂起（来自于主机）。
- **数据包缓冲接口：** 这个模块主要用来管理数据包缓冲区，以便端点进行数据的发送与接收。它可以根据SIE的要求为端点选择并分配合适的缓冲区并按照端点寄存器指定的地址在数据包缓冲区中定位正确的存储地址。在每一个字节交换后它会递增地址，直到数据包的结尾，同时，它会跟踪交换字节数从而阻止缓冲区溢出。
- **端点控制和状态寄存器：** 每个端点都有一个相关的寄存器包含它的控制信息和当前状态。对于单向/单缓冲端点而言，一个端点寄存器能够实现两个方向的端点。端点寄存器总共有8个，允许达到16个单向/单缓冲端点或者7个双缓冲端点，并可以随意组合。例如，USB外设可以通过编程使其拥有2个双缓冲端点和12个单向/单缓冲端点。
- **控制寄存器：** 控制寄存器包含整个USB设备的状态与控制信息。
- **中断寄存器：** 中断寄存器包含了中断标志与事件记录。它们可以用于查询中断的原因、中断状态或者去清除一个挂起的中断。

## APB1 接口模块

USB APB1接口包含以下模块：

- **数据包缓冲区：** 此模块由专用的512字节SRAM实现，主要作为USB的数据缓冲区来使用。每个已实现的USB端点在数据缓冲区中都有一个保留空间。所需总的缓冲区空间由使用的端点数、端点的最大包长度以及端点是否支持双缓冲来决定。此模块由数据包缓冲接口使用并创建相应的数据结构。应用程序可以直接访问该缓冲区。它由256个16位的字构成，但是由于MCU的访问方式为32位，所以其实际占用空间为1024字节。
- **仲裁器：** 由于USB模块是通过APB1接口部件与APB1总线相连的，所以当APB1总线与USB接口都发生存储器请求时将产生冲突。仲裁器就是用来解决此冲突的，其会赋予APB1总线更高的优先权，并总是保留一半的存储器带宽供USB完成传输。通过这种时分复用的策略其实现了虚拟的双端口SRAM，即在USB传输的同时，允许应用程序访问存储器。同时，在传输过程中允许任意长度的多字节APB1传输。
- **寄存器映射表：** 此部件将USB模块的各种字节宽度和位宽度的寄存器映射成能被APB1寻址的16位宽度的内存集合。
- **APB1封装：** 此部件为缓冲区和寄存器提供了到APB1的接口，并将整个USB模块映射到APB1地址空间。
- **中断映射表：** 这个模块将产生中断的事件映射到NVIC不同的中断线上。

### 19.3.2. 缓冲区

USB数据通过USB缓冲区在MCU内核和SIE之间传送，USB缓冲区是通过一个特殊的虚拟双端口SRAM存储器实现的，其被映射到APB1外设存储空间中，从地址0x4000 6000到0x4000 6400，总容量为1KB，但实际USB只使用了512B。

#### 缓冲区描述符表

每个USB端点都有一个与其相关的缓冲区描述符，专用于控制端点缓冲区。在应用中，往往将其视为特殊功能寄存器。这些寄存器并不是硬件映射的，其地址也不是固定的，其主要控制端点相关的存储区域位置、缓冲区大小、传送字节数。如果与缓冲区描述符对应的端点未使能，则相应的寄存器将不能使用，而成为可用的SRAM空间。这些寄存器的集合就被称为缓冲区描述符表，其为用户构建和控制各种长度和配置的端点缓冲区提供了灵活的方法。

缓冲区描述符表是由一个个定义了缓冲区地址和长度的数据结构的表项所组成。第一个表项是由USB\_BAR定义的。一个有效的端点通常有两个报文缓冲区，分别用来发送和接收，所以相应地对应了两个表项。每个表项包括4个16位字，因此按照8字节边界对齐。

#### 双缓冲

通常情况下，USB的每个端点都只有一个发送缓冲区和一个接收缓冲区。所以，对于USB端点的发送和接收都只是单缓冲。但是为了满足以下两种USB传输类型，USB模块实现了双缓冲的数据传输方式：

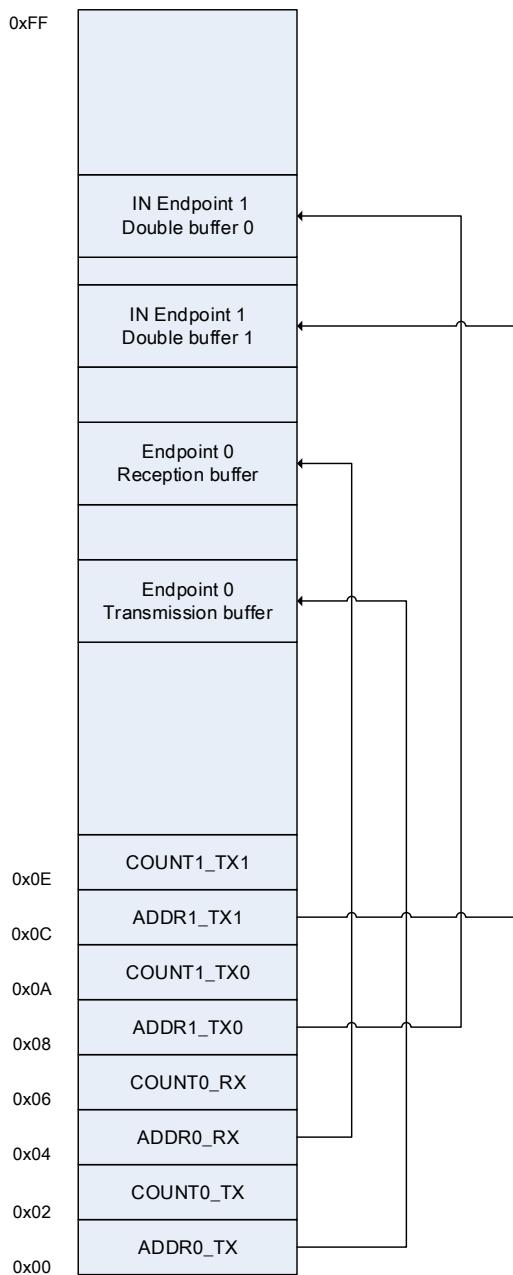
- 批量传输：其要求较高的数据吞吐率，需要保证在总线空闲的状态下，达到速率最快。
- 同步传输：其为实时的流传输方式，要求数据必须以恒定速率传输或在特定时间限制内到达。

由于一个端点寄存器可以用于此端点的发送和接收，所以其实际控制着两个缓冲区。而为了实现双缓冲，需要使得一个端点寄存器所控制的发送和接收缓冲区都变为发送缓冲区或者都变为接收缓冲区，这样就实现了双缓冲，但是如何进行具体的控制却相当复杂。详情请参考“双缓冲端点”一节。

图19-2描述了缓冲区描述表与数据包缓冲区之间的关系，其中包括使用单缓冲的端点0和使用双缓冲的端点1。

**注意：**此图未按实际比例绘出，且是通过USB总线16位数据方式进行寻址的。

图 19-2. 缓冲描述符表的用法示例（USB\_BAR = 0）



### 19.3.3. 端点

USB模块支持端点的动态配置，除了端点0（只可以配置成控制传输方式），每个端点都可以配置成四种USB传输类型中的任一种。至于缓冲区个数配置，端点0只能配置成单缓冲，其余端点既可以配置成单缓冲，也可以配置成双缓冲。

#### 单缓冲端点

通常情况下，应用程序使用的端点都被配置成单缓冲端点，在一个传输方向上都只有一个缓冲区。一次事务完成后，端点状态会被硬件自动置为NAK状态，此时USB设备需要处理前一次的

数据传输。然而，如果此时USB端点又收到新的数据分组，它将回复NAK分组，这导致主机端不断地重发同样的数据分组直到设备可以再次处理数据时才回应ACK分组。这样的重传占用了很多带宽，对于要求高吞吐量的批量传输和实时性较高的同步传输而言，这影响了传输速率。因此，引入了双缓冲端点。

### 双缓冲端点

双缓冲端点使用了两个缓冲区来进行数据收发，这两个缓冲区一个由硬件使用，另一个由应用程序来使用。在使用过程中，USB外设需要知道哪一个缓冲区正在被应用软件使用，从而避免发生冲突。既然在USB\_EPnCSR寄存器中有两个数据翻转位，而USB外设只使用一位来进行硬件数据处理（由于双缓冲功能所需的单向约束）。那么，应用程序可以使用另外一位来表明当前正在使用哪个缓冲区。这个新的缓冲区标志位称作SW\_BUF。

下表解释了USB\_EPnCSR寄存器位与DTOG/SW\_BUF定义之间的对应关系。

表 19-3. 双缓冲标志定义

缓冲标志	发送端点	接收端点
DTOG	TX_DTG(USB_EPnCSR第6位)	RX_DTG(USB_EPnCSR第14位)
SW_BUF	USB_EPnCSR第14位	USB_EPnCSR第6位

DTOG位和SW\_BUF位负责流控。当一次传输完成的时候，USB外设翻转DTOG位；当数据被复制后，应用软件需要翻转SW\_BUF位。除了首次传输，如果DTOG位的值等于SW\_BUF位的值，传输将会暂停，并且向主机发送NAK数据包。当这两位不相等的时候，传输会继续。

为了使能双缓冲器功能，EP\_CTL和EP\_KCTL位都需要被配置：

- 设置EP\_CTL = 00
- 设置EP\_KCTL = 1

表 19-4. 双缓冲的用法

端点类型	DTOG	SW_BUF	USB外设所使用的包缓冲	应用软件所使用的包缓冲
OUT	0	1	RXARn_0 / RXCNTRn_0 缓冲 描述符列表地址	RXARn_1 / RXCNTRn_1 缓冲 描述符列表地址
	1	0	RXARn_1 / RXCNTRn_1 缓冲 描述符列表地址	RXARn_0 / RXCNTRn_0 缓冲 描述符列表地址
IN	0	1	TXARn_0 / TXCNTRn_0 缓冲 描述符列表地址	TXARn_1 / TXCNTRn_1 缓冲 描述符列表地址
	1	0	TXARn_1 / TXCNTRn_1 缓冲 描述符列表地址	TXARn_0 / TXCNTRn_0 缓冲 描述符列表地址

### 端点初始化

USB端点在使用前必须初始化，初始化过程如下：

- 初始化TXARn/RXARn寄存器，填写发送/接收数据缓冲区地址；

- 按照端点的用途，配置USB\_EPnCSR寄存器的EP\_CTL和EP\_KCTL位；
- 对于发送，初始化TXCNTn并使能TX\_STA位；对于接收，初始化RXCNTRn, BLKSIZ以及BLKNUM位，然后使能RX\_STA。

对于同步端点和双缓冲批量端点，发送和接收的相关位域都必须初始化。一旦端点被使能，寄存器USB\_EPnCSR，缓冲区地址以及COUNT位域都不能通过应用软件修改。

### 端点请求

如果同时有多个端点发生中断请求，硬件会先响应优先级最高的端点号。端点的优先级定义方法如下：

- 首先依据端点属性排列，同步端点和双缓冲端点具有高优先级，其他端点为低优先级
- 接着根据同属性端点的端点号排列，端点号越小，优先级越高

应用程序应该根据上述的优先级策略顺序处理端点的中断请求。

#### 19.3.4. 中断处理

USB中断事件从对象上来分，主要有两类：设备中断，端点中断。设备中断事件较多，包括复位、挂起、唤醒等等中断；而端点中断只包括正确传输中断。所有的中断事件都通过中断标志寄存器来捕获，当有中断条件产生时，相应的中断状态位将被硬件置位（不管中断使能位是否设置）。软件可以编程中断使能寄存器的对应位，从而将特定的中断条件连接到嵌套的向量中断控制器NVIC中来触发USB中断事件。NVIC中设置了三类USB中断，具体如下：

- USB低优先级中断：可被所有USB事件触发
- USB高优先级中断：只能被同步和双缓冲批量传输的正确传输事件触发
- USB唤醒中断：可被所有的唤醒事件触发

### USB 低优先级中断处理

在此中断服务程序中需要检查中断标志寄存器的每个位，从而来确定具体的中断事件，在执行相应的操作后必须清除中断标志位，否则同样的中断会再次被触发。即使多个中断标志被置位，也只会触发一次中断。在中断服务程序中检查中断标志寄存器各个位的顺序决定了各个中断事件的优先级，处理程序会按照此优先级来处理中断事件。

### USB 高优先级中断处理

在此中断服务程序中只处理同步传输和双缓冲批量传输。当然，这两种传输也可以通过低优先级中断处理流程来处理，但是如果配置了高优先级中断，将优先通过此中断处理流程来处理，而不再通过低优先级处理流程来处理。由于在此中断服务程序中只有正确传输中断，所以其处理速度较低优先级中断要快一些。

## USB 唤醒中断处理

通常MCU睡眠时都是通过外部中断事件进行唤醒的，但是USB的唤醒除了通过外部事件进行唤醒外，还可以通过内部事件唤醒。外部中断源都是通过触发中断线18的中断来唤醒USB的；内部唤醒事件主要通过信号来唤醒。具体的唤醒中断处理过程都是直接通过USB低优先级中断处理服务程序来完成的。

在唤醒中断服务程序中，会恢复USB时钟，清除中断标志位。

### 19.3.5. 复位事件

#### 系统和上电复位

一旦系统或上电复位，此时USB控制器与接口都处于关闭状态。此时所有端点都处于禁止状态，USB模块不会响应任何分组。应用程序需要进行如下的初始化流程：

首先配置USB控制器为48MHz，打开APB1接口时钟，从而激活寄存器单元时钟，再清除复位信号；

其次开启USB收发器相关的模拟部分，其中打开内部电压需要一段启动时间；

最后配置初始化需要的寄存器和缓冲区描述表，从而使USB模块能进行正常的中断处理与数据传输。

USB固件需要做以下工作：

- 复位USB\_CTLR寄存器中的CLOSE位
- 等待内部参考电压稳定
- 清除USB\_CTLR寄存器的SETRST位
- 清除IFR寄存器以移除冗余的挂起中断，然后使能其他单元

#### 复位中断

当这个事件发生时，USB模块会复位内部协议状态机，发送和接收部分将被禁止，直到复位中断被清除。所有的配置寄存器会保持原状态而不会被复位，以保证复位后传输还可以立即正确执行。但是，设备的地址和端点寄存器会被复位。

USB固件需要做以下工作：

- 在10ms内设定AR寄存器的USBEN位来使能USB模块
- 初始化EP0CSR寄存器和它相关的数据包缓冲

### 19.3.6. 传输流程

#### USB 事务概述

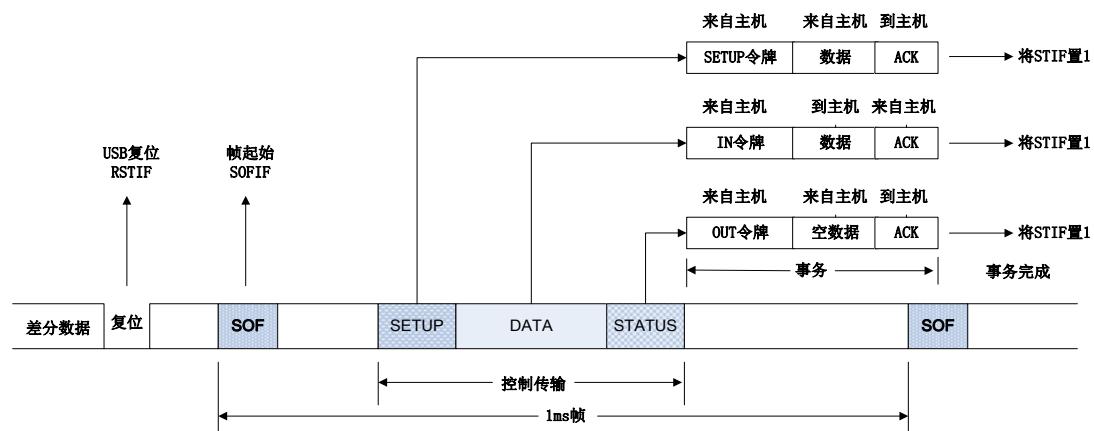
USB模块通过事务在设备端点和主机之间传输数据。主机定义了事务的方向：**IN**和**OUT**都是相对于主机而言的，**IN**事务是将数据从设备传送到主机，**OUT**事务是将数据从主机传送到设备。所有的事务都由主机控制器启动。

一次事务的过程大体如下：当主机发送一个可被USB模块识别的有效功能/端点对令牌分组后，如果USB设备端正确接收了并且需要发送或者接收数据，将随之发生相关的数据传输过程。USB模块通过缓冲区描述块和端点寄存器实现专用缓冲区与端口的数据交换，然后通过端口实现与主机的数据交换。在所有的数据传输完成后，如果需要，根据传输方向，发送或接收相应的握手分组。

在每次事务成功结束后，都会触发与端点相关的中断。在使能端点相关中断的情况下，可以读取USB中断标志寄存器来处理传输中相应端点的中断事件。

下图19-3描述了USB事务与中断事件的关系：

**图 19-3.USB 事务与中断事件**



#### 数据发送 (IN 事务)

当一个有效端点收到一个IN令牌包，SIE将会从端点相关的TXARn缓冲区开始读取TXCNTn个字节并行数据，在IN分组传输结束后，这些数据会装载到输出移位寄存器中，并同时将其转换为串行数据通过USB ATX传输到USB总线上，从而发给主机。在数据传输结束后，硬件会将计算好的CRC发送到主机。如果IN分组指示的端点无效，USB模块会根据TX\_STA的值发送NAK或STALL握手包。USB外设是按照LSB顺序从TXARn表明的地址开始发送数据包的。

当收到来自主机发送的ACK包，USB外设将翻转TX\_DTG，设置TX\_STA=10(NAK)以及将TX\_ST位置位。应用软件将通过检查USB\_IFR寄存器的EPNUM和DIR位确认哪个端点已经完成传输。当数据包存储器填满数据后应用软件通过设置TX\_STA=11(VALID)再次使能端点从而开始下一次传输。

## 数据接收 (OUT 和 SETUP 事务)

当一个有效端点接收到一个OUT或SETUP令牌包时，USB ATX会接收USB总线上的双向差分数据。SIE从ATX中接收串行数据并将其转换为并行数据流。并行数据流会被USB控制器存储到用RXARn规定的缓冲区，同时，控制器会根据实际接收到的字节数更新RXCNTRn位域。此域中的BLKSIZ和BLKNUM的值被用来计算BUF\_COUNT，BUF\_COUNT用来检测缓冲区是否溢出。USB外设以LSB的顺序接收到来自主机的数据，然后计算CRC。当检测到DATA包传输结束时，计算的CRC的值将和接收到的CRC值进行比较。如果没有错误发生，将向主机发送一个ACK握手包。

如果接收期间出现了任何错误，USB设备置位ERRIF位并且继续将数据复制到报文缓冲区，但不发送ACK包。USB设备自己能够从接收错误中恢复并且继续处理下一个数据传递。USB设备的访问从不超出由BLKSIZ和BLKNUM定义的包缓冲区。接收到紧跟在数据字节之后的2字节的CRC也会复制到数据包缓冲区。如果数据的长度比实际分配的长度大，超出的数据不被复制。这种情况称为缓冲过载。此时将发送STALL握手包，当前会话失败。

如果一个被寻址的端点是无效的，将按照USB\_EPnCSR寄存器RX\_STA位的设置发送NAK或STALL握手包而不是ACK，并且不向接收缓冲区写入数据。接收缓冲按RXARn中的地址开始写入，写入的字节数取决于接收到的数据包的长度，CRC被包含在内（也就是有效数据载荷+2），但不能超过BLKSIZ和BLKNUM定义的缓冲区的长度，超过范围的数据不会被写入缓冲区。通过这种方式，USB外设绝不会占用超出被分配的接收缓冲区域。如果数据包载荷的长度（应用程序实际使用的字节数）超出分配的缓冲区，USB外设即会检测到一个缓冲区过载情况。在这种情况下，将发送一个STALL握手包代替通常的ACK，以提醒主机这一问题，不会有中断产生并且该次会话被认为是失败的。

如果没有错误发生，USB外设将会发送ACK握手包到主机，翻转RX\_DTG位，设置RX\_STA='10(NAK)并且置位RX\_ST位。应用软件检测USB\_IFR寄存器的EPNUM和DIR位去判断哪一个端点触发了RX\_ST。处理完接收到的数据之后，应用软件可以设置RX\_STA=11(Valid)启动下一个会话。

## 控制传输

控制传输要求从一个主机发送到设备的SETUP会话开始，此会话描述设备应当执行的访问控制类型。SETUP会话后面跟着0个或者若干个携带被请求访问特定信息的数据会话。最后，一个状态会话完成控制传输并且允许端点返回控制传输的状态到客户端软件。在状态会话之后，控制传输完成，主机可以操作该端点的下一个控制传输。

要初始化控制传输，TX\_DTG和RX\_DTG位分别被设置成1和0，TX\_STA和RX\_STA被设置成'10(NAK)。根据SETUP的内容，应用软件可以判断下一个会话是IN或OUT。当RX\_ST事件发生在控制传输期间，必须先检查SETUP位。如果它是1，它就是一个SETUP会话，否则就是一个普通的OUT。USB外设通过解析SETUP事务的内容获取其关心的数据量和传输方向，并且要求将未使用的方向置为STALL，除了最后一个数据阶段。

在最后一个数据阶段，应用软件将控制端点相反方向的状态设置为NAK。这将使主机等待控制操作的完成。如果操作成功完成，软件将会把NAK改成VALID，否则改成STALL。如果状态阶段是一个OUT事务，STATUS\_OUT位将会被置位，这样携带非零数据的状态会话将被应答STALL以表明已发生了错误。

USB规范要求主机的SETUP分组具有自动重传功能，也就是说设备不能以非ACK包（NAK包或STALL包）来响应主机的SETUP请求。当SETUP分组传输失败时，会触发主机的下一次SETUP分组传输。

控制端点的RX\_STA位不可被置为'00(禁用)状态，当其正确收到一个SETUP令牌时，USB接收这个数据，执行被要求的数据传输并且发送回一个ACK握手包。如果该端点之前有RX\_ST请求未被应用软件处理，USB将丢弃当前的SETUP事务，不管状态如何都不应答任何握手包，以此来模拟一个接收错误并且强迫主机再次发送一个SETUP令牌。这样做是为了避免丢失紧随一次RX\_ST中断之后的又一个SETUP分组传输。

### 同步传输

同步传输可以保证固定的传输速率以及固定的延迟，但当总线发生错误时并不支持数据重发。此时只有接收方才能确定是否有错误发生。USB底层协议并不允许同步通道中的握手包被回送给发送方。通常来说，握手包不会被发送以通知发送方数据包是否被成功接收。因此，同步协议没有一个握手阶段，在数据包发送之后没有ACK包。它也不支持数据翻转，DATA0 PID仅仅被用来开始一个数据包的发送。

通过设置EP\_CTL位为'10，该端点被定义为同步端点。STA位有两个可能值：'00(禁用)和'11(有效)，其他的任何值是非法的。应用软件可以实现双缓冲以提高性能。通过在每个会话时交换发送和接收包缓冲，应用软件可以将数据复制进缓冲或复制出缓冲，同时USB外设能在另一个缓冲区中处理数据的发送或接收。通过查询DTOG位即可知道USB外设现在正在使用哪一个缓冲区。

应用软件按照要用的首个缓冲区去初始化DTOG。在每一个事务的结尾，RX\_ST或TX\_ST位被置位，这取决于使能方向，而忽略CRC错误或缓冲过载情况（如果错误发生，ERRIF位将被置位）。同时，USB外设将会翻转DTOG位，但是不影响STA位。

## 19.3.7. 电源管理

### 电源模式

USB应用往往都有几种不同的电源要求和配置。最常见的电源模式如下：

1. 总线供电模式：USB协议一直强调由USB从设备进行电源管理，这要求设备从总线获得应用所需的电源。总线供电设备必须满足下述限制：
  - 处于非配置状态的从设备，从USB总线最多获取100mA的电流
  - 已配置的设备只能按照配置描述符的MaxPower位域的设置获得电流且最大值不超过500 mA
  - 已挂起设备最多获取500uA电流
2. 自供电模式：在这种模式下，USB应用为自身提供电源，只有很小一部分电源来自USB
3. 以自供电为主的双电源供电模式：某些应用可能需要双电源供电，在这种模式下，应用主要使用内部电源，但是在没有内部电源可用的时候切换到总线供电

本USB模块既支持总线供电也支持自供电模式，这两种模式之间的切换需要手动控制。

### 挂起与唤醒

假如在USB总线上没有活动超过3ms，设备将进入挂起状态。由于全速模式下USB主机会一毫秒发送一个帧起始包（SOF），所以USB模块可以通过检测3个连续的SOF分组丢失事件来确定进入挂起状态。如果有来自主机的恢复信号，将唤醒一个挂起的设备。恢复信号来自于恢复序列，恢复序列可以由主机发起，也可以由设备自身触发。在挂起状态下，硬件不再检测挂起信号直到唤醒过程结束。

USB外设还支持软件初始化的远程唤醒。远程唤醒通常指设备唤醒主机，其要求先唤醒USB模块，再唤醒主机。为了启动远程唤醒功能，应用软件必须使能所有的时钟，并清除挂起位。这将导致硬件产生一个远程唤醒信号的上行数据流。

通过将SETSPS位设为1，即可使能挂起模式并且禁用对于SOF接收的检查。将LOWM位设为1将关闭USB模拟收发器的静态功耗，但是此时仍能检测到恢复信号。

通过将USB\_CTLR寄存器的RSREQ位置位，然后在10ms内再将其清为0可以启动恢复序列（10ms的时间间隔可以用ESOF中断来实现，该中断在内核运行时每1ms发生一次）。

### 链接电源管理（LPM）级别 L1

当USB子端点寄存器的SUB\_STA位是0x3(Valid)，如果主机发送EXT PID和LPM SubPID，设备将进入LPM L1状态（睡眠状态）。

将SETSPS位设为1将使能低功耗模式，并且禁用对SOF接收的检查。将LOWM位设为1将关闭USB模拟收发器的静态功耗，但是此时仍能检测到恢复信号。当检测到恢复信号的时候，退出睡眠状态。

## 19.4. USB 寄存器

### 19.4.1. USB 控制寄存器 (USB\_CTLR)

地址偏移: 0x40

复位值: 0x0003

该寄存器可以按半字（16位）或者字（32位）的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIE	PMOUIE	ERRIE	WKUPIE	SPSIE	RSTIE	SOFIE	ESOFIE	保留		RSREQ	SETSPS	LOWM	CLOSE	SETRST	

位/位域	名称	描述
15	STIE	成功传输中断使能 0: 禁用成功传输中断 1: 当USB_IFR寄存器的STIF位被置位，产生中断

14	PMOUIE	包缓冲上溢/下溢中断使能 0: 当包缓冲上溢/下溢不产生中断 1: 当USB_IFR寄存器的PMOUIF位被置位，产生中断.
13	ERRIE	错误中断使能 0: 禁用错误中断 1: 当USB_IFR寄存器的ERRIF位被置位，产生中断
12	WKUPIE	唤醒中断使能 0: 禁用唤醒中断 1: 当USB_IFR寄存器的WKUPIF位被置位，产生中断
11	SPSIE	挂起状态中断使能 0: 禁用挂起状态中断 1: 当USB_IFR寄存器的SPSIF位被置位，产生中断
10	RSTIE	USB复位中断使能 0: 禁用USB复位中断 1: 当USB_IFR寄存器的RSTIF位被置位，产生中断
9	SOFIE	帧起始中断使能 0: 禁用帧起始中断 1: 当USB_IFR寄存器的SOFIF位被置位，产生中断
8	ESOFIE	期望的帧起始中断使能 0: 禁用预期的帧起始中断 1: 当USB_IFR寄存器的ESOFIF位被置位，产生中断
7:5	保留	必须保持复位值
4	RSREQ	软件向USB主机发送一个恢复请求，USB主机应该按USB规范驱动这个恢复序列 0: 没有恢复请求 1: 发送恢复请求
3	SETSPS	当USB_IFR寄存器的SPSIF位被置位时，软件应该设置挂起状态 0: 没有设置挂起状态 1: 设置挂起状态
2	LOWM	当置位这一位时，USB在挂起状态进入低功耗模式。如果从挂起状态恢复，硬件会复位这一位。 0: 无影响 1: 在挂起模式进入低功耗模式
1	CLOSE	当这一位被置位的时候，USB进入关闭状态，并且完全关闭USB，同主机断开 0: 不在关断状态 1: 在关断状态
0	SETRST	当这位置位，USB外设应该被复位 0: 无影响 1: 发生复位

### 19.4.2. USB 中断标志寄存器 (USB\_IFR)

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字 (16 位) 或者字 (32 位) 的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIF	PMOUIF	ERRIF	WKUPIF	SPSIF	RSTIF	SOFIF	ESOFIF	保留		DIR		EPNUM[3:0]			

r            rc\_w0            rc\_w0            rc\_w0            rc\_w0            rc\_w0            rc\_w0            rc\_w0            r            r            r            r

位/位域	名称	描述
15	STIF	成功传输中断标志 当一个会话成功完成时，硬件置位该位。
14	PMOUIF	包缓冲上溢/下溢中断标志 硬件置位该位表示包缓冲区存储不下所有所传输的数据。软件写0清该位。
13	ERRIF	错误中断标志 当在会话期间有错误发生时，硬件置位该位。软件写0清该位。
12	WKUPIF	唤醒中断标志 在SUSPEND状态下，当总线上有活动被检测到时，硬件置位该位。软件写0清该位。
11	SPSIF	挂起状态中断标志 当USB总线无任何活动超过3ms时，硬件置位该位，表明有SUSPEND请求。软件写0清该位。
10	RSTIF	USB复位中断标志 当检测到USB RESET信号时硬件置位该位。软件写0清该位。
9	SOFIF	帧起始中断标志 一个新的SOF包到达时硬件置位该位。软件写0清该位。
8	ESOFIF	期望的帧起始中断标志 硬件置位表示一个SOF被期望但是还没有到达。软件写0清该位。
7:5	保留	必须保持复位值
4	DIR	会话传输方向 硬件置位表示会话的传输方向 0: IN类型 1: OUT 类型
3:0	EPNUM[3:0]	端点号 硬件置位确认当前会话所关联的端点。

### 19.4.3. USB 状态寄存器 (USB\_SR)

地址偏移: 0x48

复位值: 0x0XXX 这里 X 是未定义的

该寄存器可以按半字（16 位）或者字（32 位）的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_DP	RX_DM	LOCK	SOFLN[1:0]	FCNT[10:0]											
r	r	r	r						r						

位/位域	名称	描述
15	RX_DP	接收数据D+线状态 代表DP线的状态。
14	RX_DM	接收数据D-线状态 代表DM线的状态。
13	LOCK	锁定USB 硬件置位表明接收到了至少两个连续SOF包。
12:11	SOFLN[1:0]	丢失SOF 当每次发生ESOFIF事件时，硬件递增此位，一旦再次接收到SOF则清除该位。
10:0	FCNT[10:0]	帧编号计数器 每次收到SOF，帧编号计数器将会增加。

#### 19.4.4. USB 设备地址寄存器 (USB\_AR)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字（16 位）或者字（32 位）的方式访问

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

位/位域	名称	描述
15:8	保留	必须保持复位值
7	USBEN	USB设备使能 通过软件设置该位使能USB设备。 0: USB设备禁用。没有会话要处理 1: USB设备使能
6:0	USBADDR[6:0]	USB设备地址 总线复位之后，地址被复位为0x00。若USB使能位被置位，则从设备会响应功能地址DEV_ADDR的报文。

### 19.4.5. USB 缓冲器地址寄存器 (USB\_BAR)

地址偏移: 0x50

复位值: 0x0000

该寄存器可以按半字 (16 位) 或者字 (32 位) 的方式访问

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

位/位域	名称	描述
15:3	BAR[15:3]	缓冲器地址 所分配缓冲器(512byte on-chip SRAM)的起始地址, 用来保存缓冲描述符表以及包缓冲。
2:0	保留	必须保持复位值

### 19.4.6. USB 端点 n 控制/状态寄存器 (USB\_EPnCSR), n=[0..7]

地址偏移: 0x00 to 0x1C

复位值: 0x0000

该寄存器可以按半字 (16 位) 或者字 (32 位) 的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_ST	RX_DTG	RX_STA[1:0]	SETUP	EP_CTL[1:0]	EP_KCTL	TX_ST	TX_DTG	TX_STA[1:0]	EP_AR[3:0]						
rc_w0	t	t	r	rw	rw	rc_w0	t	t	rw						

位/位域	名称	描述
15	RX_ST	正确接收 当一个成功的OUT/SETUP会话完成时, 硬件置位此位。 通过软件写0清该位。
14	RX_DTG	接收数据PID翻转位 本标志位代表非同步端点的翻转数据位(0=DATA0, 1=DATA1)。 用来实现双缓冲端点的流控功能。 用于同步端点的缓冲区交换。
13:12	RX_STA[1:0]	接收状态位 通过软件写1翻转。 写0保持不变。 参考下表
11	SETUP	Setup会话完成 当一个SETUP会话完成时, 硬件置位此位。
10:9	EP_CTL[1:0]	端点类型控制

参考下表

8	EP_KCTL	端点类别控制 其具体含义取决于端点类型的设置。 参考下表
7	TX_ST	正确发送 当一个IN会话成功完成时，硬件置位此位。 软件清0
6	TX_DTG	发送数据PID翻转位 本标志位代表非同步端点的翻转数据位(0=DATA0, 1=DATA1)。 用来实现双缓冲端点的流控功能。 用于同步端点的缓冲区交换。
5:4	TX_STA[1:0]	发送状态位 参考下表
3:0	EP_AR	端点地址 用来指示会话的目标端点

表 19-5. 接收状态编码

RX_STA[1:0]	含义
00	<b>DISABLED:</b> 忽略此端点的所有接受请求
01	<b>STALL:</b> 握手状态为STALL
10	<b>NAK:</b> 握手状态为 NAK
11	<b>VALID:</b> 使能端点的接收

表 19-6. 端点类型编码

EP_CTL[1:0]	含义
00	BULK(批量传输)
01	CONTROL(控制传输)
10	ISO(同步传输)
11	INTERRUPT(中断传输)

表 19-7. 端点类别编码

EP_CTL[1:0]	EP_KCTL 含义
00	DBL_BUF
01	STATUS_OUT

表 19-8. 发送状态编码

TX_STA[1:0]	含义
00	<b>DISABLED:</b> 忽略此端点的所有发送请求
01	<b>STALL:</b> 握手状态为STALL
10	<b>NAK:</b> 握手状态为NAK
11	<b>VALID:</b> 使能端点的发送

#### 19.4.7. USB 发送缓冲地址寄存器 n (USB\_TXARn)

地址偏移: [USB\_BAR] + n\*16

USB 本地地址: [USB\_BAR] + n\*8

该寄存器可以按半字 (16 位) 或者字 (32 位) 的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXARn[15:1]														TXARn0	
rw														rw	

位/位域	名称	描述
15:1	TXARn[15:1]	发送缓冲地址 在收到下一个IN分组时，需要发送的数据所在的缓冲区起始地址。
0	TXARn[0]	必须设为0

#### 19.4.8. USB 发送字节数目寄存器 n (USB\_TXCNTn)

地址偏移: [USB\_BAR] + n\*16 + 4

USB 本地地址: [USB\_BAR] + n\*8 + 2

该寄存器可以按半字 (16 位) 或者字 (32 位) 的方式访问

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

位/位域	名称	描述
15:10	保留	必须保持复位值
9:0	TXCNTn[9:0]	发送字节数 在收到下一个IN令牌后，将发送的字节数。

#### 19.4.9. USB 接收缓冲器地址寄存器 n (USB\_RXARn)

地址偏移: [USB\_BAR] + n\*16 + 8

USB 本地地址: [USB\_BAR] + n\*8 + 4

该寄存器可以按半字 (16 位) 或者字 (32 位) 的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXARn[15:1]														RXARn0	
rw														rw	

位/位域	名称	描述

---

15:1	RXARn[15:1]	接收缓冲器地址 收到下一个OUT或者SETUP分组时，用于保存数据的缓冲区起始地址。
0	RXARn[0]	必须设为0

#### 19.4.10. USB 接收字节数目寄存器 n (USB\_RXCNTRn)

地址偏移: [USB\_BAR] + n\*16 + 12

USB 本地地址: [USB\_BAR] + n\*8 + 6

该寄存器可以按半字（16位）或者字（32位）的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKSIZ	BLKNUM[4:0]				RXCNTRn[9:0]										

位/位域	名称	描述
15	BLKSIZ	块的大小 0: 块大小是2字节 1: 块大小是32字节
14:10	BLKNUM[4:0]	块数目 包缓冲区所分配的块的数目。
9:0	RXCNTRn[9:0]	接收字节数 在收到下一个OUT/SETUP令牌后，接收到数据的字节数。

#### 19.4.11. USB 子端点 n 寄存器 (USB\_SUBEPnR), n=[0..7]

地址偏移: 0x100到0x11C

复位值: 0x0000

该寄存器可以按半字（16位）或者字（32位）的方式访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUB_ST	Reserved	SUB_STA[1:0]	Reserved	SUBPID_ATTR											

位/位域	名称	描述
15	SUB_ST	LPM令牌成功接收 当一个LPM令牌接收成功完成，硬件置位。 通过软件写0清该位。
14	保留	必须保持复位值
13:12	SUB_STA[1:0]	当收到LPM subpid后，发送的握手包状态位。 通过软件写1翻转。

写0保持不变。

参考下表

11	保留	必须保持复位值
10:0	SUBPID_ATTR	LPM令牌的bmAttribute域 这些位由软件读，硬件进行写操作。

表 19-9. 子端点状态

SUB_STA[1:0]	含义
00	<b>DISABLED:</b> 忽略此端点的所有 LPM 令牌请求
01	<b>STALL:</b> 握手状态为STALL
10	<b>NYET:</b> 握手状态为 NYET
11	<b>VALID:</b> 使能端点的 LPM 令牌功能

#### 19.4.12. USB LPM 控制寄存器 (USB\_LPMCNTR)

地址偏移: 0x140

复位值: 0x0000

该寄存器可以按半字（16 位）或者字（32 位）的方式访问

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

rw

位/位域	名称	描述
15	LPMSTIE	LPM token成功传输中断使能 该位由软件进行置位和复位 0: 没有中断产生 1: 当LPM令牌成功传输时，产生中断
14:0	保留	必须保持复位值

#### 19.4.13. USB LPM 中断状态寄存器 (USB\_LPMISTR)

地址偏移: 0x144

复位值: 0x0000

该寄存器可以按半字（16 位）或者字（32 位）的方式访问

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

rw

位/位域	名称	描述
15	LPMSTIF	LPM令牌正确传输中断标志 该位由软件进行置位和复位 0: 无影响 1: LPM令牌正确传输
14:0	保留	必须保持复位值

## 20. 实时时钟 (RTC)

### 20.1. 简介

RTC 模块提供了一个包含年/月/日的日历以及时/分/秒/亚秒的时间的时钟功能。

除亚秒外用二进制码显示外，时间和日历都以 BCD 码的形式显示。

RTC 可以进行夏令时补偿。

RTC 可以工作在低功耗模式下，并通过软件配置来智能唤醒。

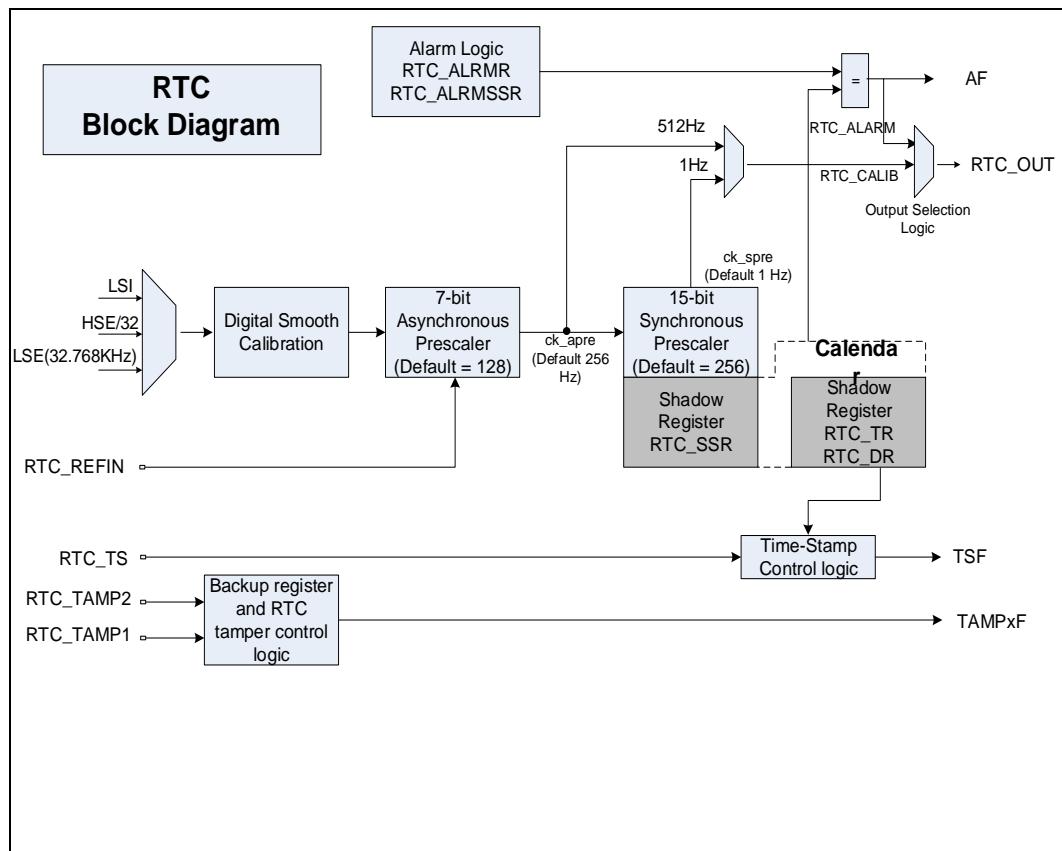
RTC 支持外接更高精度的低频时钟，用以达到更高的日历精度。

### 20.2. 主要特性

- 通过软件设置来实现夏令时补偿
- 参考时钟检测功能：通过外接更高精度的低频率时钟源(50Hz或60Hz)来提供日历精度
- 数字校准功能：通过调整最小时间（最大可调精度0.95ppm）单位来进行日历校准
- 通过移位功能进行亚秒级调整
- 记录事件时间的时间戳功能
- 模式可配置的两个独立的侵入源
- 可编程的日历和位域可屏蔽的闹钟
- 可屏蔽的中断源：
  - 闹钟
  - 时间戳检测
  - 侵入检测
- 5个32位通用备份寄存器，能够在低功耗模式下保留数据。当有外部事件侵入时，备份寄存器将会复位

## 20.3. 功能描述

### 20.3.1. RTC 框图



RTC 模块包括:

- 闹钟事件/中断
- 侵入事件/中断
- 32位备份寄存器
- 可选的RTC输出功能:
  - 512Hz (默认预分频值)
  - 1Hz(默认预分频值)
  - 闹钟事件(输出极性可配置)
- 可选的RTC输入功能:
  - 时间戳事件检测: RTC\_TS
  - 侵入事件检测1: RTC\_TAMP1
  - 侵入事件检测2: RTC\_TAMP2

- 参考时钟输入：RTC\_REFIN(50或60 Hz)

### 20.3.2. RTC 管脚

PC13, PC14 和 PC15 管脚的功能可以通过配置寄存器 RTC\_TAFCR 和 RTC\_CTLR 来直接控制。

PC13 的功能优先级如下表：

PC13功能	OS	COEN	TAP1EN	TSEN	PC13 MODE	PC13 VALUE
RTC_ALARM OD输出	1	X	X	X	X	0
RTC_ALRM PP输出	1	X	X	X	X	1
RTC_CALIB PP输出	0	1	X	X	X	X
RTC_TAMP1 浮空输入	0	0	1	0	X	X
RTC_TS and RTC_TAMP1 浮空输入	0	0	1	1	X	X
RTC_TS 浮空输入	0	0	0	1	X	X
强制PP输出	0	0	0	0	1	PC13 value bit
唤醒引脚 或标准GPIO	0	0	0	0	0	X

注意：(1) OD 表示开漏，PP 表示推挽

(2) ‘X’ 无影响

PC14 的功能优先级如下表：

PC14功能	LSEEN in RCC_BDCR	LSEBPS in RCC_BDCR	PC14 MODE	PC14 VALUE
LSE 晶振 模拟输入	1	0	X	X
LSE 旁路 模拟输入	1	1	X	X
强制PP输出	0	X	1	PC14 value bit
标准GPIO	0	X	0	X

注意：(1) OD 表示开漏，PP 表示推挽

(2) ‘X’ 无影响

PC15 的功能优先级如下表：

PC15功能	LSEEN in RCC_BDCR	LSEBPS in RCC_BDCR	PC15 MODE	PC15 VALUE
LSE 晶振模拟输入	1	0	X	X
强制推挽输出	1	1	1	PC15 value bit
	0	X		
标准GPIO	0	X	0	X

注意：(1) OD 表示开漏，PP 表示推挽

(2) ‘X’ 无影响

### 20.3.3. 时钟和预分频

RTC 单元有三个可选的独立时钟源：LSE，LSI和32分频后的HSE。

在RTC单元，有两个预分频器用来实现日历功能和其他功能。一个分频器是7位异步预分频器，另一个是15位同步预分频器。异步分频器主要用来降低功率消耗。如果两个分频器都被使用，建议异步分频器的值尽可能大。

两个预分频器的频率计算公式如下：

$$f_{\text{ck_apre}} = \frac{f_{\text{rtcclk}}}{\text{PREDIV\_A} + 1}$$

$$f_{\text{ck_spre}} = \frac{f_{\text{ck_apre}}}{\text{PREDIV\_S} + 1} = \frac{f_{\text{rtcclk}}}{(\text{PREDIV\_A} + 1) * (\text{PREDIV\_S} + 1)}$$

ck\_apre时钟为二进制RTC\_SSR亚秒递减计数器提供时钟。当RTC\_SSR递减为0时，其值将自动重载为PREDIV\_S的值。

ck\_spre时钟为日历寄存器提供驱动时钟，一个时钟将产生一个秒脉冲。

### 20.3.4. 实时时钟和日历

当APB总线访问RTC日历寄存器RTC\_DSR，RTC\_TSR和RTC\_SSR时，BYPUSHAD位决定他们值是访问影子寄存器还是真实日历寄存器。默认情况下BYPUSHAD为0，APB线访问影子日历寄存器。

每两个RTC 时钟，影子日历寄存器值会更新为真实日历寄存器的值，与此同时RSF位也会再次置位。在DeepSleep和Standby模式下，影子寄存器不会更新。退出这两种模式时，软件必须清除RSF位。如果软件想要在BYPUSHAD=0的情况下读日历寄存器的值，须等RSF置1之后（最大的等待时间是2个RTC时钟周期）。

**注意：**在 BYPUSHAD=0下，读日历寄存器(RTC\_SSR,RTC\_TSR,RTC\_DSR)的APB时钟的频率(f<sub>APB</sub>)必须至少是RTC时钟频率(f<sub>RTCCCLK</sub>)的七倍。

系统复位将复位影子寄存器。

### 20.3.5. 位域可屏蔽可配置的闹钟

RTC闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC闹钟功能的使能由RTC\_CTLR寄存器中的AE位控制。当AE=1并且闹钟所有位域的值与对应的日历时间值匹配，AF标志位将会置位。

**注意：**当秒字段未被屏蔽时( RTC\_ALRMR 寄存器的 SMSK=0)，为确保正常运行,RTC\_PSCR寄存器的同步预分频系数应大于等于3。

如果一个位域被屏蔽，这个位域被认为在逻辑上匹配的。如果所有的位域被屏蔽在AE位被置位3个RTC时钟周期后，AF位将置位。

### 20.3.6. RTC 初始化和配置

#### RTC 寄存器写保护

在默认情况下，PWR\_CTLR寄存器的BKPWE位被清0。所以写RTC寄存器前需要软件提前设置BKPWE位。

上电复位后，大多数RTC寄存器是被写保护的。写入这些寄存器的第一步是解锁这些保护。

通过下面的步骤，可以解锁这些保护：

1. 写‘0xCA’到RTC\_WPR寄存器
2. 写‘0x53’到RTC\_WPR寄存器

写一个错误的值到RTC\_WPR会使写保护再次生效。

写保护状态不受系统复位的影响。

#### 日历初始化和配置

通过以下步骤可以设置日历和预分频器的值：

1. 设置INIT位为1进入初始化模式。在这一模式下，日历计数器停止，它的值可以被更新。
2. 循环读INITE位。如果INITE位为1，确保已经真正进入到初始化模式。这大概需要2个RTCCLK时钟周期 (由于时钟同步的原因)。
3. 在RTC\_PSCR寄存器中，设置同步和异步预分频器的分频系数。
4. 在影子寄存器(RTC\_TSR和RTC\_DSR)中写初始的日历值，并且通过设置RTC\_CTLR寄存器的FMT位来配置时间的格式 (12或24小时)。
5. 清除INIT位退出初始化模式。

大约4 RTCCLK 时钟周期后，真正的日历寄存器将从影子寄存器载入时间和日期的设定值，同时日历计数器将要重新开始运行。

**注意：**初始化以后如果要读取日历寄存器(BYPUSHAD=0)，软件应该确保RSF位已经置1。

INITS标志通过检测年值是否为0x00来标记日历是否已初始化。如果年值为0，则INITS=0；如果年值不为0，则INITS=1。

### 夏令时补偿

通过S1H,A1H和BKP位，RTC模块支持夏令时补偿调节功能。

当日历正在运行时，S1H和A1H能使日历减去或加上1小时。S1H和A1H功能可以重复设置，BKP位能记录这个调节操作。设置S1H或A1H位后，减或加1小时将在下一秒钟到来时生效。

### 闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

1. 清除寄存器RTC\_CTLR的AE位禁用闹钟。
2. 设置Alarm寄存器(RTC\_ALRMSSR/RTC\_ALRMR)。
3. 设置寄存器RTC\_CTLR的AE位使能Alarm功能。

## 20.3.7. 读取日历

### 当 BYPSHAD=0 时，读日历寄存器

当BYPUSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基本要求：APB1总线时钟频率必须大于或等于RTC时钟频率的7倍。在任何情况下APB1总线时钟的频率都不能低于RTC的时钟频率。

当APB1总线时钟频率低于7倍RTC时钟频率时，日历的读取应该遵守以下流程：

1. 读取两次日历时间和日期寄存器。
2. 如果两次的值相等，那么这个值就是正确的。
3. 如果这两次的值不相等，应该再读一次。
4. 第三次的值可以认为是正确的。

RSF每2个RTC时钟周期被置位一次。在这时，影子日历寄存器会更新为真实的日历时间和日期。

为了确保这3个值(RTC\_SSR,RTC\_TSR,RTC\_DSR)为同一时间，硬件上采取了如下一致性机制：

1. 读RTC\_SSR锁定RTC\_TSR和RTC\_DSR的更新
2. 读RTC\_TSR锁定RTC\_DSR的更新
3. 读RTC\_DSR解锁RTC\_TSR 和 RTC\_DSR 的更新

如果想在一个很短的时间间隔内（少于2RTCCLK）读取日历，应先清除RSF位并等待其置位后再读取。

下面几种情况，软件须等待**RSF**置位后才能读日历寄存器(**RTC\_SSR**,**RTC\_TSR**,**RTC\_DSR**):

1. 系统复位之后
2. 日历初始化之后
3. 一次同步之后

特别是从低功耗模式唤醒后，软件必须清除**RSF**位并等待**RSF**再次置位后才能读取日历寄存器。

### 当 **BYPSHAD=1** 时，读日历寄存器

当**BYPSHAD=1**，**RSF**位会被硬件清0，读日历寄存器不需考虑**RSF**位。当前真实的日历寄存器值会被直接读取。如此配置的好处是当从低功耗模式(Deepsleep/Standy模式)唤醒后，软件可以立即获取当前日历寄存器的值而无需加入任何等待延迟(此延迟最大为2个RTC时钟周期)。

由于没有**RSF**位周期性的置位，如果两次读日历寄存器之间出现**ck\_apre**时钟边沿，不同寄存器(**RTC\_SSR**/**RTC\_TSR**/**RTC\_DSR**)的值可能并非同一时刻。

另外，如果日历寄存器的值正在发生变化的时刻被APB总线读取，那么有可能APB总线读取的值是不准确的。

为了确保日历值的正确性和一致性，读取时软件须如下操作：连续读取所有日历寄存器的值两次，如果上两次的值是一样的，那么这个值就是相关且准确的。

### 20.3.8. 复位 RTC

在 RTC 单元，有两个复位源可用：系统复位和备份域复位。

当系统复位有效地时候，日历影子寄存器和**RTC\_STR**寄存器的某些位将要复位到默认值。

备份域复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

- RTC 当前实时日历寄存器
- RTC 控制寄存器 (**RTC\_CTLR**)
- RTC 预分频寄存器 (**RTC\_PSCR**)
- RTC 校准配置寄存器 (**RTC\_CCR**)
- RTC 移位寄存器 (**RTC\_SHIFTCTRLR**)
- RTC 时间戳寄存器 (**RTC\_TSSSR**/**RTC\_TTR**/**RTC\_TDR**)
- RTC 侵入和复用功能配置寄存器(**RTC\_TAFCR**)
- RTC 备份寄存器(**RTC\_BKPxR**)
- RTC 闹钟寄存器(**RTC\_ALRMSSR**/**RTC\_ALRMR**)

当系统复位或者进入低功耗模式的时候，RTC单元将会继续运行。但是如果备份域复位，RTC将会停止计数并且所有的寄存器会复位。

### 20.3.9. RTC 同步

当用户有一个高精度的远程时钟而且RTC时间和远程时钟只有一个亚秒级的偏差，RTC单元提供一个称作同步移位的功能去消除这个偏差来提高秒钟的精确性。

RTC\_SSR 寄存器以二进制格式显示亚秒值，RTC运行时该值是递减计数。因此增加亚秒的值(通过增加SFS[14:0]的值到同步预分频器计数器SS[15:0])或减少(通过增加SFS[14:0]的值到同步预分频器计数器SS[15:0]并且同时置位A1S位)亚秒的值能延迟或提前下一秒到达的时间。

RTC\_SSR的最大值取决于RTC\_PSCR寄存器的PREDIV\_S的值。PREDIV\_S越大，调整的精度也就越高。

因为1Hz的时钟(ck\_spre)由PREDIV\_A和PREDIV\_S共同产生，越高的PREDIV\_S值就意味着越低的PREDIV\_A值，同时越低的PREDIV\_A意味着越高的功耗。

**注意:**在使用同步移位功能之前，软件必须检查RTC\_SSR中SS的第15位(SS[15])并确保该位为0。写RTC\_SHIFTCTRL寄存器之后，RTC\_ISR寄存器的SHPF位将会再次置位。当同步移位操作完成时，SHPF位被硬件清0。系统复位不影响SHPF位。当REFCLKEN=0时，同步移位操作才能正确的工作。如果REFCLKEN=1，软件禁止写入RTC\_SHIFTCTRL。

### 20.3.10. RTC 参考时钟检测

RTC参考时钟是另外一种提高RTC秒级精度的方法。为了使能这项功能，需要有一个相对于LSE有更高精度的外部时钟源 (50Hz or 60 Hz)。

使能这项功能之后(REFCLKEN=1)，每一个秒更新的时钟边沿(1Hz)将与最近的RTC\_REFIN时钟沿进行对比。在大多数情况下，这两个时钟沿是对齐的。但当两个时钟沿由于LSE准确度的原因没有对齐的时候，RTC参考时钟的检测功能会偏移1Hz时钟沿一点相位使得下一个1Hz时钟沿和参考时钟沿对齐。

当REFCLKEN=1，每一秒前后都会有一个进行检测的时间窗，处于不同的检测状态时间窗周期也不同。当检测状态处于检测第一个参考时钟边沿时，使用7个ck\_apre的时间窗周期，当检测状态处于边沿对齐操作时，使用3个ck\_apre时间窗周期。

无论使用哪一种时间窗，当参考时钟在时间窗中被检测到的时候，同步预分频计数器会被强制重载。当两个时钟(ck\_spre和参考时钟)边沿是对齐的，这个重载操作对1Hz 日历更新没有任何影响。但是当两个始终边沿没有对齐时，这个重载操作将会移动 ck\_spre 时钟边沿以使得ck\_spre(1Hz) 时钟边沿和参考时钟边沿对齐。

当参考检测功能正在运行中但外部参考时钟消失(在3个 ck\_apre 时间窗内没有发现参考时钟边沿)，日历也能通过LSE继续自动更新。如果这个参考时钟重新恢复，参考时钟检测功能会先用7 -ck\_apre窗口去检测参考时钟，然后用3-ck\_apre 窗口去调节ck\_spre(1Hz)时钟边沿。

**注意:** 使能参考时钟检测功能之前(REFCLKEN=1)，软件必须配置PREDIV\_A为0x7F和

PREDIVE\_S为0xFF。

待机模式下，参考时钟检测功能不可用。

### 20.3.11. RTC 平滑数字校准

RTC校准功能是一种用于校准RTC频率的方法，该方法通过调整用户配置的校准周期内的RTC时钟脉冲个数的方式来实现校准。

完成一次这种校准相当于执行了如下行为：

在一次校准周期内，RTC时钟的脉冲个数通过增加或者减少了一定的数目。这种校准的分辨率大约为0.954ppm，范围为从-487.1ppm 到+488.5ppm。

校准周期的时间可以配置到 $2^{20}/2^{19}/2^{18}$  RTC 时钟周期，如果RTC的输入频率是32.768KHz，这些校准周期时间分别代表32/16/8秒。

校准配置寄存器(RTC\_CCR)指定了在校准周期内要屏蔽的RTCCLK时钟数目：

- 设置CMSK[0]为1使得1个周期被屏蔽在周期时间里
- 设置CMSK[1]为1使得2个额外的周期被屏蔽
- 设置CMSK[2]为1使得4个额外的周期被屏蔽
- 设置CMSK[8]为1使得256个额外的周期被屏蔽

因此使用CMSK能屏蔽从0到511个的时钟周期，RTC频率能被最多降低到487.1ppm。

为提高RTC频率可以设置CALP位。如果CALP位被置位，将会有512个额外的周期增加到周期时间期间，这意味着每 $2^{11}/2^{10}/2^9$ (32/16/8 秒) RTC时钟插入一个周期

因此使用CALP可以使RTC频率增加488.5ppm。

同时使用CMSK和CALP每个周期时间可以做到调整-511到+512个RTC时钟周期。这意味着在0.954ppm分辨率的情况下，调整范围为从-487.1ppm到+488.5ppm。

当校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal} = f_{rtcclk} * [1 + (CALP \times 512 - CALM) / (2^N + CALM - CALP \times 512)]$$

注意：N=20/19/18 for 32/16/8 秒校准时间周期。

**当 PREDIV\_A < 3 时校准：**

当异步预分频器值(PREDIV\_A)被设置小于3时，当正在使用校准功能时，软件不应该将CALP位设置为1。当PREDIV\_A<3，CALP位设置将会被忽略。

当PREDIV\_A小于3时，PREDIV\_S值应小于标称值。假设RTC时钟频率是正常的32.768KHz，对应的PREDIV\_S应该按下面所示设置：

PREDIV\_A = 2: PREDIV\_S减少2(8189 with 32.768KHz)

PREDIV\_A = 1: PREDIV\_S减少4(16379 with 32.768KHz)

PREDIV\_A = 0: PREDIV\_S减少8(32759 with 32.76KHz)

当PREDIV\_A小于3,频率校准公式如下:

$$f_{cal} = f_{rtcclk} * [1 + (256 - CALM) / (2^N + CALM - 256)]$$

**注意:** 如果PREDIV\_A小于3, 若RTC时钟为精确的32.768KHz, CMSK的校准中点是0x100。

### 验证 RTC 校准

提供1Hz校准时钟的输出用于协助软件测量并验证RTC的精度。

在有限的测量周期内测量RTC的频率, 最高可能发生2个RTCCLK的测量误差。

为了消除这一测量误差, 测量周期应该和校准周期一致。

- 校准周期设为32秒(默认配置)

用准确的32秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.477ppm(0.5 RTCCLK 在32秒周期内)之内。

- 校准周期设为16秒(通过设置CWND16位)

使用此配置, CMSK[0]被硬件置0。

用准确的16秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.954ppm(0.5 RTCCLK 在16秒周期内)之内。

- 校准周期设8秒(通过设置CWND8位)

使用此配置, CMSK[1: 0]被硬件置0。

用准确的8秒周期去测量1Hz校准输出的准确性能保证这个测量在1.907ppm(0.5 RTCCLK 周期超过8秒)之内。

### 运行中重校准

当INITE位是0, 用下面的步骤, 软件可以更新RTC\_CCR:

- 1). 等待RECPF位置0
- 2). 写一个新的值到RTC\_CCR寄存器
- 3). 3个ck\_apre时钟周期之后, 新的校准设置开始生效。

### 20.3.12. 时间戳功能

时间戳功能由RTC\_TS管脚输入, 通过配置TSEN位来使能。

当RTC\_TS管脚有时间戳事件发生时, 会将日历的值保存在时间戳寄存器中(RTC\_TDR/RTC\_TTR/RTC\_TSSSR), 同时时间戳标志(TSF)也将由硬件置1。

如果时间戳中断使能被启用(TSIE), 时间戳事件会产生一个中断。

时间戳寄存器只会在时间戳事件第一次发生的时刻 (TSF=0) 记录日历时间, 而当TSF=1

时，时间戳事件的值不会被记录。

RTC模块提供了一个可选的功能特性来扩大时间戳事件的触发源：设置TAPTS=1来让侵入检测功能的侵入事件同时也作为时间戳的输入源。

**注意：**因为同步机制的原因，当时间戳事件发生时，TSF会延迟2个ck\_apre周期置位。

### 20.3.13. 侵入检测

RTC\_TAMPx管脚可以作为侵入事件检测功能输入管脚，检测模式有两种可供用户选择：边沿检测模式或者是带可配置过滤功能的电平检测模式。

#### RTC 备份寄存器(RTC\_BKPxR)

RTC备份寄存器处于V<sub>DD</sub>备份域中，即使V<sub>DD</sub>电源被切断，该区域的寄存器的电源还可由V<sub>BAT</sub>提供。从待机模式唤醒或系统复位操作都不会影响这些寄存器。

当被检测到有侵入事件或者Flash读保护等级由1变成0的时候，备份域的寄存器将会复位。

#### 初始化侵入检测功能

TAPxE位可以独立使能对应于不同管脚上的RTC侵入检测功能。

使能TAPxE位启动侵入检测功能之前，需要设置好侵入检测的配置。

当检测到侵入事件，相应的标志位(TAMPxF)将会置位。

如果侵入事件中断使能被启用(TAPIE)，侵入事件会产生一个中断。

任何侵入事件都会导致备份寄存器(RTC\_BKPxR)复位。

#### 侵入事件源的时间戳

使能TAPTS位，能让侵入检测功能被用作时间戳功能。如果这位被设置为1，当检测到侵入事件时TSF也将会被置位，如同使能了时间戳功能。当检测到侵入事件时，无论TAPTS位的值如何，TAMPxF位将置位。

#### 侵入事件检测为边沿检测模式

当FLIT位为0x0时，侵入检测被设置成边沿检测模式，TAPxTRIG位决定检测沿是上升沿还是下降沿。当侵入检测配置为边沿检测模式时候，侵入检测输入管脚上的上拉电阻将会被禁用。

由于检测侵入事件会复位备份寄存器 (RTC\_BKPxR)，因此对备份寄存器写操作时应该确保侵入事件导致的复位和写操作不会同时发生。避免这种情形的推荐方法是先关闭侵入检测功能，在完成写操作后再重新启动该功能。

**注意：**PC13上的侵入检测功能即使VDD电源被关掉也依然可以运行。

### 侵入事件检测为带可配置滤波功能的电平检测模式

当FLIT位没有被设置成0x0时，侵入检测被设置成电平检测模式，TAMPxTRG位决定连续采样次数(2, 4 or 8)。

当DISPU被设置成0(默认值)，内部的上拉电阻将会在每一次采样前预充电侵入管脚，这样侵入事件的输入管脚上就允许连接更大的电容。预充电的时间可以通过PRCH位来配置。越大的电容，所需的充电时间越长。

电平检测模式下每次采样之间的时间间隔是可配置的。通过调整采样频率(FREQ)，软件能在功耗和检测延迟之间取得一个平衡。

### 20.3.14. 校准时钟输出

如果COEN位设置为1，PC13管脚会输出参考校准时钟。

当COS位设置为0(默认值)并且异步预分频器(PREDIV\_A)设为0x7F时，RTC\_CALIB的频率是 $f_{RTCCLK}/64$ 。因此若RTCCLK的频率为32.768KHz, RTC\_CALIB对应的输出为512Hz。因为下降沿存在轻微的抖动，因此推荐使用RTC\_CALIB输出的上升沿。

当COS位设置为1时，RTC\_CALIB的频率计算公式为 $f_{RTCCLK}/((PREDIV_S+1)*(PREDIV_A+1))$ 。若RTCCLK为32.768KHz，如果预分频器是默认值，那么RTC\_CALIB对应的输出是1Hz。

### 20.3.15. 闹钟输出

当OS控制位被设置为0x01时，RTC\_ALARM 输出功能被启用。这个功能将直接输出RTC\_ISR寄存器的AF位的值。

RTC\_CTLR寄存器中的OPOL位可以配置AF位输出时候的极性，因此RTC\_ALARM的输出电平有可能与AF位相反。

### 20.3.16. RTC 省电模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC 中断
深度睡眠模式	当时钟源是LSE或LSI时可以工作	RTC闹钟/侵入事件/时间戳事件/ 唤醒事件
待机模式	当时钟源是LSE或LSI时可以工作	RTC闹钟/侵入事件/时间戳事件/ 唤醒事件

### 20.3.17. RTC 中断

所有的RTC中断都被连接到EXTI控制器。

如果想使用RTC闹钟/侵入/时间戳中断，应按下面步骤操作：

- 1). 设置并使能对应的EXTI中连接到RTC闹钟/侵入/时间戳的中断线，然后配置该线为上升沿

触发模式

2). 配置并使能RTC全局中断

3). 配置并使能RTC闹钟/侵入/时间戳功能

中断	事件标志	控制位	退出睡眠	退出停止	退出待机
闹钟	AF	AIE	Y	Y(*)	Y(*)
时间戳	TSF	TSIE	Y	Y(*)	Y(*)
侵入 1	TAMP1F	TAPIE	Y	Y(*)	Y(*)
侵入 2	TAMP2F	TAPIE	Y	Y(*)	Y(*)

\* : 仅仅当RTC时钟源是LSE或LSI时有效。

## 20.4. RTC 寄存器

### 20.4.1. RTC 时间寄存器 (RTC\_TSR)

偏移地址: 0x00

系统复位值: 当BYPUSHAD = 0, 0x0000 0000 。

当BYPUSHAD = 1, 无影响。

写保护寄存器, 仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								APM	HT[1:0]	HU[3:0]					
								rW	rW	rW					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MT[2:0]		MU[3:0]			保留		ST[2:0]		SU[3:0]					
	rW		rW			保留		rW		rW					

位/位域	名称	描述
31:23	保留	必须保持复位值
22	APM	AM/PM 标志 0: AM 或 24 小时制 1: PM
21:20	HT[1:0]	时钟十位值, 以 BCD 码形式存储
19:16	HU[3:0]	时钟个位值, 以 BCD 码形式存储
15:	保留	必须保持复位值
14:12	MT[2:0]	分钟十位值, 以 BCD 码形式存储
11:8	MU[3:0]	分钟个位值, 以 BCD 码形式存储

---

7	保留	必须保持复位值
6:4	ST[2:0]	秒钟十位值, 以 BCD 码形式存储
3:0	SU[3:0]	秒钟个位值, 以 BCD 码形式存储

#### 20.4.2. RTC 日期寄存器 (RTC\_DSR)

偏移地址: 0x04

系统复位值: 当 BYPSHAD = 0, 0x0000 2101。

当 BYPSHAD = 1, 无影响

写保护寄存器, 仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						YT[3:0]				YU[3:0]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD[2:0]		MT	MU[3:0]				保留		DT[1:0]	DU[3:0]					
rw		rw		rw						rw			rw		

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	YT[3:0]	年份十位值, 以 BCD 码形式存储
19:16	YU[3:0]	年份个位值, 以 BCD 码形式存储
15:13	WD[2:0]	星期 0x0: 保留 0x1: 星期一 ... 0x7: 星期日
12	MT	月份十位值, 以 BCD 码形式存储
11:8	MU[2:0]	月份个位值, 以 BCD 码形式存储
7:6	保留	必须保持复位值
5:4	DT[1:0]	日期十位值, 以 BCD 码形式存储
3:0	DU[3:0]	日期个位值, 以 BCD 码形式存储

#### 20.4.3. RTC 控制寄存器 (RTC\_CTLR)

偏移地址: 0x08

系统复位: 无影响

备份域复位值: 0x0000 0000

## 写保护寄存器

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				COEN	OS[1:0]		OPOL	COS	BKP	S1H	A1H				
					rw		rw		rw		rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	保留	AIE	TSEN	保留	AE	保留	FMT	BYPSHAD	REFCLKEN	TSEDGE	保留				
	rw		rw		rw		rw		rw		rw				

位/位域	名称	描述
31:24	保留	必须保持复位值
23	COEN	校准输出使能 0: 关闭校准输出 1: 使能校准输出
22:21	OS[1:0]	输出选择 该位用来选择 RTC_ALARM 输出的标志源。 0x0: 禁用 RTC_ALARM 输出 0x1: 启用 alarm 标志输出 0x2: 保留 0x3: 保留
20	OPOL	输出极性 该位用来反转 RTC_ALARM 输出。 0: 禁用反转 RTC_ALARM 输出 1: 启用反转 RTC_ALARM 输出
19	COS	校准输出选择 仅当 COEN=1 并且预分频器是默认值时有效。 0: 校准输出是 512Hz 1: 校准输出是 1Hz
18	BKP	备份位 该位可以通过软件灵活使用。常用来记录夏令时调整操作。
17	S1H	减 1 小时(冬季时间变化) 当前时间非零的情况下，将当前时间减去一个小时。 0: 没有影响 1: 在下一个秒改变时，将减少一个小时
16	A1H	增加 1 小时(夏季时间变化) 将当前时间增加一个小时。 0: 没有影响 1: 在下一个秒改变时，将增加一个小时
15	TSIE	时间戳中断使能 0: 禁用时间戳中断

		1: 启用时间戳中断
14:13	保留	必须保持复位值
12	AIE	RTC 闹钟中断使能 0: 禁用闹钟中断 1: 启用闹钟中断
11	TSEN	时间戳功能使能 0: 禁用时间戳功能 1: 启用时间戳功能
10:9	保留	必须保持复位值
8	AE	闹钟功能使能 0: 禁用闹钟功能 1: 启用闹钟功能
7	保留	必须保持复位值
6	FMT	显示时间格式 0: 24 小时制 1: 12 小时制 <b>注意:</b> 仅能在初始化状态进行写入
5	BYPSHAD	旁路影子寄存器 0: 读取的日历的值来自影子日历寄存器 1: 读取的日历的值来自真正日历寄存器 <b>注意:</b> 如果 APB1 时钟的频率小于 RTCCLK 频率的 7 倍, 该位必须设为 1.
4	REFCLKEN	参考时钟检测功能使能 0: 禁用参考时钟检测功能 1: 启用参考时钟检测功能 <b>注意:</b> 仅能在初始化状态进行写入
3	TSEDGE	时间戳事件有效检测边沿 0: 上升沿是时间戳事件有效检测沿 1: 下降沿是时间戳事件有效检测沿
2:0	保留	必须保持复位值

#### 20.4.4. RTC 状态和标志寄存器 (RTC\_ISTR)

偏移地址: 0x0C

系统复位: 仅 INIT, INITE 和 RSYNF 位被置 0, 其他位无影响。

备份域复位值: 0x0000 0007

写保护寄存器, 除 RTC\_ISTR[14:8] 外

该寄存器只能按字(32位)访问

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

保留															RECPF
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TAMP2F	TAMP1F	TSOVF	TSF	保留	AF	INIT	INITE	RSYNF	INITSF	SOPF	保留	AWF		

rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rw      r      rc\_w0      r      rc\_w0      rc\_w0      r

位/位域	名称	描述
31:17	保留	必须保持复位值
16	RECPF	校准挂起标志 当软件向 RTC_CCR 进行写操作时，该位被硬件置 1。当校准设置开始执行后，该位被硬件清零 0。
15	保留	必须保持复位值
14	TAMP2F	RTC_TAMP2 事件标志 当在 tamper2 输入管脚检测到侵入事件时，该位硬件置 1。软件可以通过向该位写 0 来清除。
13	TAMP1F	RTC_TAMP1 事件标志 当在 tamper1 输入管脚检测到侵入事件时，该位硬件置 1。软件可以通过向该位写 0 来清除。
12	TSOVF	时间戳事件溢出标志 如果 TSF 位已经置位，当再次检测到时间戳事件时，该位会通过硬件置 1。软件可以通过向该位写 0 来清除。
11	TSF	时间戳事件标志 当检测到一个时间戳事件时，该位会通过硬件置 1。软件可以通过向该位写 0 来清除。
10:9	保留	必须保持复位值
8	AF	Alarm 发生标志 当现在的时间/日期与闹钟设置的时间/日期匹配的时候，该位会通过硬件置 1。软件可以通过向该位写 0 来清除。
7	INIT	进入初始化模式 0: 自由运行模式 1: 进入初始化模式设置时间/日期，预分频计数器将停止运行
6	INITE	初始化状态标志 该位被硬件置 1，初始化状态时可以设置日历寄存器和预分频器。 0: 日历寄存器和预分频器的值不能改变 1: 日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影

子日历寄存器。初始化模式(INIT)，移位操作挂起标志(SOPF)或者旁路模式(BYPSHAD)会清除该位。该位也可以通过软件写 0，清除。

- 0: 影子寄存器未同步
- 1: 影子寄存器已同步

4	INITSF	日历初始化状态标志 硬件置位和清零。当日历寄存器的年份值不为 0 时置位，为 0 时清除。 0: 日历尚未初始化 1: 日历已经初始化
3	SOPF	移位功能操作挂起标志 0: 移位操作没有挂起 1: 移位操作挂起
2:1	保留	必须保持复位值
0	AWF	Alarm 配置位可写标志位 硬件置位和清零。AE=0 时，标记 alarm 是否可写。 0: 不允许修改 Alarm 寄存器设置 1: 允许修改 Alarm 寄存器设置

#### 20.4.5. RTC 预分频寄存器 (RTC\_PSCR)

偏移地址: 0x10

系统复位: 无影响

备份域复位值: 0x007F 00FF

寄存器仅在初始化状态可写

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:23	保留	必须保持复位值
22:16	PREDIV_A[6:0]	同步预分频系数 $ck_{apre}$ 频率 = RTCCLK 频率/(PREDIV_A+1)
15	保留	必须保持复位值
14:0	PREDIV_S[14:0]	同步预分频系数 $ck_{spre}$ 频率 = $ck_{apre}$ 频率/(PREDIV_S+1)

### 20.4.6. RTC 警报寄存器 (RTC\_ALMR)

偏移地址: 0x1C

系统复位: 无影响

备份域复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMSK	WDSEL	DT[1:0]		DU[3:0]			HMSK	APMF	HT[1:0]		HU[3:0]				
		rw		rw			rw	rw	rw		rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSK	MT[2:0]		MU[3:0]			SMSK	ST[2:0]		SU[3:0]						
	rw		rw			rw	rw		rw		rw				

位/位域	名称	描述
31	DMSK	警报日期位域屏蔽位 0: 不屏蔽日期/天位域 1: 屏蔽日期/天位域
30	WDSEL	星期选择 0: 此时 DU[3:0] 代表日期 1: 此时 DU[3:0] 代表星期几, 此时 DT[3:0]无意义
29:28	DT[1:0]	日期十位值, 以 BCD 码格式存储
27:24	DU[3:0]	日期个位值或星期天数, 以 BCD 码格式存储
23	HMSK	警报小时位域屏蔽位 0: 不屏蔽小时位域 1: 屏蔽小时位域
22	APMF	AM/PM 标志 0: AM 或 24 小时制 1: PM
21:20	HT[1:0]	时钟十位值, 以 BCD 码形式存储
19:16	HU[3:0]	时钟个位值, 以 BCD 码形式存储
15	MMSK	警报分钟位域屏蔽位 0: 不屏蔽分钟位域 1: 屏蔽分钟位域
14:12	MT[2:0]	分钟十位值, 以 BCD 码形式存储
11:8	MU[3:0]	分钟个位值, 以 BCD 码形式存储
7	SMSK	秒位域屏蔽位 0: 不屏蔽秒位域

**1: 屏蔽秒位域**

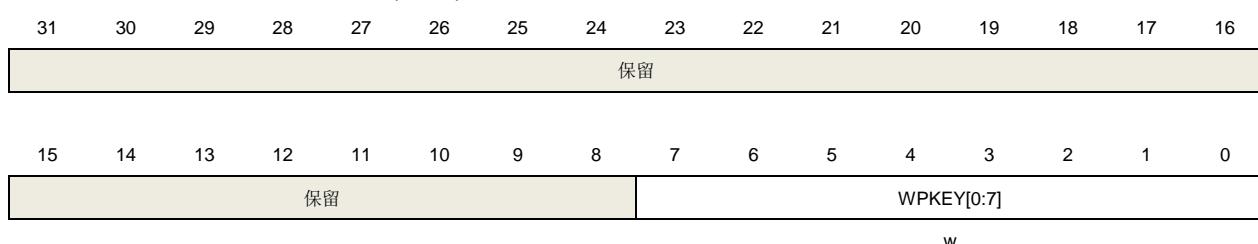
6:4	ST[2:0]	秒钟十位值, 以 BCD 码形式存储
3:0	SU[3:0]	秒钟个位值, 以 BCD 码形式存储

### 20.4.7. RTC 写保护 (RTC\_WPR)

偏移地址: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WPKEY[0:7]	写保护的解锁值

### 20.4.8. RTC 亚秒寄存器 (RTC\_SSR)

偏移地址: 0x28

系统复位值: 当BYPUSHAD = 0, 0x0000 0000。

当BYPUSHAD = 1, 无影响。

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SS[15:0]	亚秒值 该位值是同步预分频计数器的值。秒的小数部分由下面公式给出: 秒的小数部分 = ( PREDIV_S - SS ) / ( PREDIV_S + 1 )

#### 20.4.9. RTC 移位功能控制寄存器 (RTC\_SHIFTCTRL)

偏移地址: 0x2C

复位值: 0x0000 0000

仅当SHPF=0, 该寄存器可写

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A1S	保留														
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SFS[14:0]														
w															

位/位域	名称	描述
31	A1S	增加一秒 0: 无影响 1: 增加一秒到时钟/日历 该位与 SFS 位一起使用, 增加小于一秒到当前时间。
30:15	保留	必须保持复位值
14:0	SFS[14:0]	减去小于一秒的一段时间 这位的值将增加到同步预分频计数器 当仅用 SFS 时, 由于同步与分频器是一个递减计数器, 所以时钟将会延迟。 延迟(秒) = $SFS / (PREDIV_S + 1)$ 当 A1S 和 SFS 一起使用时, 时钟将会提前 提前(秒) = $(1 - (SFS / (PREDIV_S + 1)))$

注意: 写入此寄存器会导致 RSYNF 位被清 0。

#### 20.4.10. RTC 时间戳时间寄存器 (RTC\_TTR)

偏移地址: 0x30

备份域复位值: 0x0000 0000

系统复位: 无影响

当TSF被置1, 该位用来记录日历时间

清除TSF位也会清除此寄存器

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								APM	HT[1:0]	HT[3:0]					
w						r	r	r	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MT[2:0]			MU[3:0]			保留	ST[2:0]			SU[3:0]				
w							r								

位/位域	名称	描述
31:23	保留	必须保持复位值
22	APM	AM/PM 标记 0: AM 或 24 小时制 1: PM
21:20	HT[1:0]	时钟十位值, 以 BCD 码形式存储
19:16	HU[3:0]	时钟个位值, 以 BCD 码形式存储
15	保留	保持复位值
14:12	MT[2:0]	分钟十位值, 以 BCD 码形式存储
11:8	MU[3:0]	分钟个位值, 以 BCD 码形式存储
7	保留	保持复位值
6:4	ST[2:0]	秒钟十位值, 以 BCD 码形式存储
3:0	SU[3:0]	秒钟个位值, 以 BCD 码形式存储

#### 20.4.11. RTC 时间戳日期寄存器 (RTC\_TDR)

偏移地址: 0x34

备份域复位值: 0x0000 0000

系统复位: 无影响

当TSF被置1, 该位用来记录日历时间

清除TSF位也会清除此寄存器

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD[2:0]	MT		MU[3:0]		保留		DT[1:0]		DU[3:0]						
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:13	WD[2:0]	星期数
12	MT	月份十位值, 以 BCD 码形式存储
11:8	MU[3:0]	月份个位值, 以 BCD 码形式存储
7:6	保留	保持复位值

5:4	DT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DU[3:0]	日期个位值，以 BCD 码形式存储

#### 20.4.12. RTC 时间戳亚秒寄存器 (RTC\_TSSSR)

偏移地址: 0x38

备份域复位: 0x0000 0000

系统复位: 无影响

当TSF被置1，该位用来记录日历时间

清除TSF位也会清除此寄存器

该寄存器只能按字(32位)访问

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SS[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

#### 20.4.13. RTC 校准配置寄存器 (RTC\_CCR)

偏移地址: 0x3C

备份域复位: 0x0000 0000

系统复位: 无影响

写保护寄存器

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CWND8	CWND16	保留				CMSK[8:0]								

rw

rw

rw

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CALP	RTC 频率增加 488.5ppm

		0: 无影响 1: 每 $2^{11}$ 个脉冲增加一个 RTCCLK 脉冲 该位需与 CMSK 位一起使用。如果输入时钟频率是 32.768KHz, 在 32s 校准窗期间, 增加的 RTCCLK 脉冲数是 $(512 * CALP) - CMSK$
14	CWND8	采用 8 秒校准周期 0: 无影响 1: 采用 8 秒校准周期 <b>注意:</b> 当 CWND8=1, CMSK[1:0] 被锁定在“00”。
13	CWND16	采用 16 秒校准周期 0: 无影响 1: 采用 16 秒校准周期 <b>注意:</b> 当 CWND16=1, CMSK[0] 被锁定在“0”。
12:9	保留	必须保持复位值
8:0	CMSK[8:0]	校准周期 RTCCLK 脉冲屏蔽数 在 $2^{20}$ RTCCLK 脉冲之内屏蔽的脉冲数 此项功能可以以 0.9537 ppm 的分辨率来降低日历频率

#### 20.4.14. RTC 侵入和复用功能配置寄存器 (RTC\_TAFCR)

偏移地址: 0x40

备份域复位: 0x0000 0000

系统复位: 无影响

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PC15MDE	PC15VAL	PC14MDE	PC14VAL	PC13MDE	PC13VAL	保留			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]	FLIT[1:0]	FREQ[2:0]	TAPTS	保留		TAP2TRIG	TAP2EN	TAPIE	TAP1TRIG	TAP1EN				
rw	rw	rw	rw	rw			rw	rw	rw	rw	rw				

位/位域	名称	描述
31:24	保留	必须保持复位值
23	PC15MDE	PC15 模式 0: PC15 受 GPIO 配置控制 1: 如果 LSEEN=0, PC15 被强制设置为推挽输出
22	PC15VAL	PC15 值 当 PC15MODE=1 且 LSEEN=0 时, PC15 输出该位的值
21	PC14MDE	PC14 模式 0: PC14 受 GPIO 配置控制

		1: 如果 LSEEN=0, PC14 被强制设置为推挽输出
20	PC14VAL	<p>PC14 值</p> <p>当 PC14MODE=1 且 LSEEN=0 时, PC14 输出该位值。</p>
19	PC13MDE	<p>0: PC13 受 GPIO 配置控制</p> <p>1: 如果所有的 RTC 复用功能关闭, PC13 被强制设置为推挽输出</p>
18	PC13VAL	<p>Alarm 输出类型控制/PC13 输出值</p> <p>当 RTC 使能复用功能时, 输出 RTC_ALARM:</p> <p>0: RTC_ALARM 用开漏模式输出</p> <p>1: RTC_ALARM 用推挽模式输出</p> <p>当所有的 RTC 复用功能禁用且 PC13MODE=1 时, PC13 输出该位值</p>
17:16	保留	必须保持复位值
15	DISPU	<p>RTC_TAMPx 上拉禁用位</p> <p>0: 使能内部 RTC_TAMPx 引脚上的上拉电阻并在采样前进行预充电</p> <p>1: 禁用预充电功能</p>
14:13	PRCH[1:0]	<p>RTC_TAMPx 的预充电时间</p> <p>该位设置决定了每次采样前的预充电时间</p> <p>0x0: 1 RTC clock</p> <p>0x1: 2 RTC clock</p> <p>0x2: 4 RTC clock</p> <p>0x3: 8 RTC clock</p>
12:11	FLIT[1:0]	<p>RTC_TAMPx 过滤器计数设置</p> <p>该位决定了侵入事件检测模式和在电平检测模式下连续采样的次数。</p> <p>0x0: 用边沿模式检测侵入事件, 预充电功能被自动禁用。</p> <p>0x1: 用电平模式检测侵入事件。连续采样到 2 个有效电平时认为发生侵入事件</p> <p>0x2: 用电平模式检测侵入事件。连续采样到 4 个有效电平时认为发生侵入事件</p> <p>0x3: 用电平模式检测侵入事件。连续采样到 8 个有效电平时认为发生侵入事件</p>
10:8	FREQ[2:0]	<p>侵入事件电平模式检测的采样频率</p> <p>0x0: 每次采样间隔 32768 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 1Hz)</p> <p>0x1: 每次采样间隔 16384 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 2Hz)</p> <p>0x2: 每次采样间隔 8192 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 4Hz)</p> <p>0x3: 每次采样间隔 4096 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 8Hz)</p> <p>0x4: 每次采样间隔 2048 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 16Hz)</p> <p>0x5: 每次采样间隔 1024 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 32Hz)</p> <p>0x6: 每次采样间隔 512 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 64Hz)</p> <p>0x7: 每次采样间隔 256 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 128Hz)</p>
7	TAPTS	<p>将侵入检测功能同时置位时间戳标记</p> <p>0: 无影响</p> <p>1: 当检测到侵入事件时, 即使 TSEN=0, TSF 也会被置位</p>

6:5	保留	必须保持复位值
4	TAP2TRIG	对应于 RTC_TAMP2 输入管脚的侵入事件检测触发沿设置 如果侵入检测处于边沿模式(FLIT =0): 0: 上升沿触发一个侵入检测事件 1: 下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLIT ≠0): 0: 低电平触发一个侵入检测事件 1: 高电平触发一个侵入检测事件
3	TAP2EN	Tamper 2 检测使能位 0: 禁用 tamper 2 检测功能 1: 启用 tamper 2 检测功能
2	TAPIE	侵入检测中断使能 0: 禁用侵入中断 1: 启用侵入中断
1	TAP1TRIG	对应于 RTC_TAMP1 输入管脚的侵入事件检测触发沿设置 如果侵入检测处于边沿模式(FLIT =0): 0: 上升沿触发一个侵入检测事件 1: 下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLIT≠0): 0: 低电平触发一个侵入检测事件 1: 高电平触发一个侵入检测事件
0	TAP1EN	Tamper 1 检测使能位 0: 禁用 tamper 1 检测功能 1: 启用 tamper 1 检测功能

注意：在改变侵入检测配置之前，应该复位 TAMPxE 位。

#### 20.4.15. RTC 闹钟亚秒寄存器 (RTC\_ALMSSR)

偏移地址: 0x44

备份域复位: 0x0000 0000

系统复位: 无影响

写保护寄存器，仅当AE=0或INIT=1，可以进行写操作

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
------	----	----

---

31:28	保留	必须保持复位值
27:24	MSKSS[3:0]	亚秒位域的屏蔽控制位 0x0: 屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1: SS[0]位用于时间匹配，其他位被忽略。 0x2: SS[1:0] 位用于时间匹配，其他位被忽略。 0x3: SS[2:0] 位用于时间匹配，其他位被忽略。 0x4: SS[3:0] 位用于时间匹配，其他位被忽略。 0x5: SS[4:0] 位用于时间匹配，其他位被忽略。 0x6: SS[5:0] 位用于时间匹配，其他位被忽略。 0x7: SS[6:0] 位用于时间匹配，其他位被忽略。 0x8: SS[7:0] 位用于时间匹配，其他位被忽略。 0x9: SS[8:0] 位用于时间匹配，其他位被忽略。 0x10: SS[9:0] 位用于时间匹配，其他位被忽略。 0x11: SS[10:0] 位用于时间匹配，其他位被忽略。 0x12: SS[11:0] 位用于时间匹配，其他位被忽略。 0x13: SS[12:0] 位用于时间匹配，其他位被忽略。 0x14: SS[13:0] 位用于时间匹配，其他位被忽略。 0x15: SS[14:0] 位用于时间匹配，其他位被忽略。 <b>注意：</b> 同步计数器的第 15 位(RTC_SSR 寄存器中的 SS[15])从不被匹配。
23:15	保留	必须保持复位值
14:0	SS[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSS 位控制。

#### 20.4.16. RTC 备份寄存器 (RTC\_BKPxR)(x=0,1,2,3,4)

偏移地址: 0x50到0x60

备份域复位: 0x0000 0000

系统复位: 无影响

该寄存器只能按字(32位)访问

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

---

位/位域	名称	描述
31:0	BKP[31:0]	备份域寄存器。 软件可读写寄存器。由于此寄存器可由 V <sub>BAT</sub> 供电，因此寄存器值在省电模式下依然



保持有效。当侵入检测标志位 TAMP<sub>x</sub> 置 1 或者禁用 Flash 读保护的时候此寄存器值将复位。

## 21. 触摸传感控制器 (TSI)

### 21.1. 简介

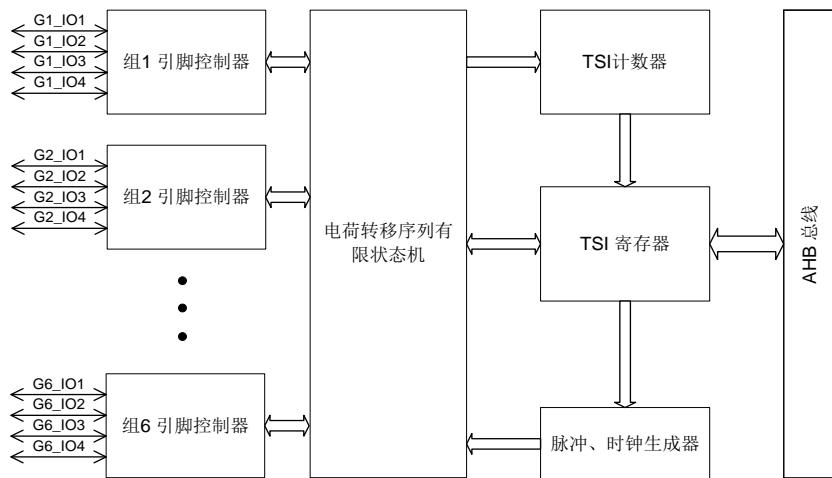
触摸传感控制器 (TSI) 为按键、触摸检测等应用提供了简易的解决方案。控制器基于电荷转移方法，当一个手指接近电极时会引起整个系统的电容变化，而TSI可以通过电荷转移的方法来检测到这种变化从而感知到手指接近这一行为。

### 21.2. 主要特性

- 内置的电荷转移序列完全由硬件控制
- 包含6个完全并行的引脚组
- 同时最多可配置18个通道引脚和6个采样引脚
- 电荷转移序列的频率可配置
- 可以由用户实现自定义的电荷转移序列
- 可产生序列结束、错误标志和中断
- 内置的电荷转移序列中包含可配置的扩频功能

### 21.3. 功能描述

#### 21.3.1. TSI 框图

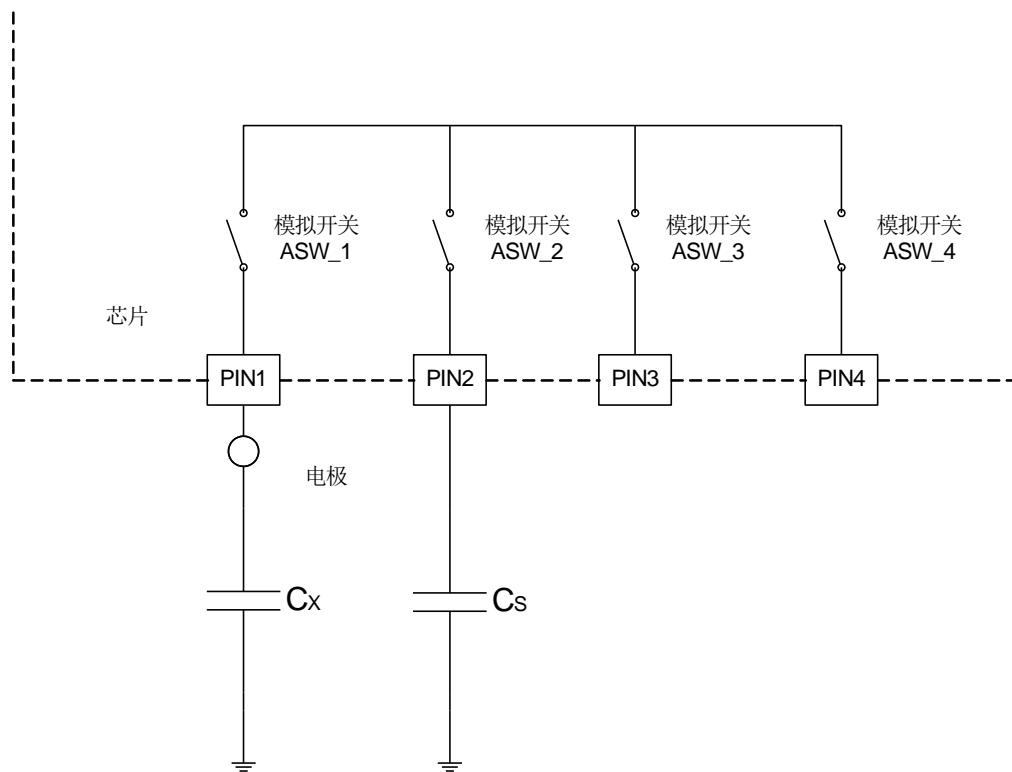
**图 21-1. TSI 模块框图**


### 21.3.2. 触摸传感技术概述

针对触摸感知有很多不同的技术手段，例如光学、电阻式、电容式及应变式等等，其中检测系统的变化是这些技术的关键问题和目标。TSI模块使用电荷转移的方法来检测由于触摸带来的系统的变化，尤其是触摸电极上电容的变化。为了检测到这种变化，TSI会运行内置的或用户自定义的电荷转移序列，这个序列包括数次充电、转移的步骤直至满足终止条件，这一过程中的充电、转移次数就表征了系统的电容特性，因此应用程序可以通过监测这一次数值来感知系统的变化。

如下图中，由4个引脚组成了一个引脚组，4个引脚各自通过一个模拟开关连接到一个公共点上，这些模拟开关是实现电荷转移序列的关键组成部分。每一个引脚组中应该配置一个引脚为采样引脚，其他的3个引脚中应该至少配置一个为通道引脚。例如在下图中，PIN1就被配置为通道引脚，PIN2则配置为采样引脚，PIN3和PIN4没有使用。PIN1作为通道引脚，它会连接一个电极，这个电极通常设计在PCB板上，而采样引脚PIN2上需要连接一个采样电容  $C_s$ 。现在通道引脚 PIN1 的电容包括  $C_x$  和通过电极引入的电容，所以当手指触摸电极时，PIN1的电容会增大而 PIN2的电容保持不变。TSI模块会使用电荷转移序列来检测到这一变化，具体实现方法会在下一节中详细说明。

图 21-2. 一个通道引脚的采样引脚的框图



### 21.3.3. 电荷转移序列

TSI模块会执行电荷转移序列来测量一个通道引脚的电容变化。整个序列的详细步骤如表21-1所示。(表21-1假设系统是按照上图来配置的，即PIN1是通道引脚，PIN2是采样引脚)

表 21-1. 电荷转移序列的详细步骤以及引脚和开关状态

步骤序号	名称	ASW1 状态	ASW2 状态	PIN1 状态	PIN2 状态
1	放电	闭合	闭合	浮空输入	下拉
2	缓冲时间 1	断开	断开	浮空输入	浮空输入
3	充电	断开	断开	输出高电平	浮空输入
4	充电扩展	断开	断开	输出高电平	浮空输入
5	缓冲时间 2	断开	断开	浮空输入	浮空输入
6	电荷转移	闭合	闭合	浮空输入	浮空输入
7	缓冲时间 3	断开	断开	浮空输入	浮空输入
8	对比	断开	断开	浮空输入	浮空输入

#### 1. 放电

闭合 ASW1 和 ASW2 并且将PIN2配置成下拉，使 $C_x$ 和  $C_s$ 放电。这个步骤是初始化的操作，应该由软件在传输序列开始之前通过配置寄存器来实现。该步骤的放电时间需要保证 $C_x$  和 $C_s$  的电压被彻底放电至0。

#### 2. 缓冲时间 1

ASW1 和ASW2 断开，PIN1 被设置成浮空输入。

### 3. 充电

充电过程中，通道引脚PIN1被配置成输出高电平，给  $C_x$  充电，ASW1和ASW2仍然断开。充电时间应该配置为可确保（细节参看寄存器部分） $C_x$  充电到VDD。

### 4. 充电扩展

这是在电荷转移序列中的一个可选步骤，目的是实现扩频。此步骤中所有引脚和模拟开关的状态与步骤3完全相同，唯一差别是持续时间，该步骤的持续时间在每次循环都会变化，最大值可以TSI寄存器配置。

### 5. 缓冲时间 2

ASW1和ASW2断开，PIN1被设置成浮空输入。

### 6. 电荷转移

这一步骤中ASW1和ASW2闭合，而PIN1被设置成浮空输入，电荷将从  $C_x$  向  $C_s$  转移。这个转移时间应该被合理配置（细节参看寄存器部分）以确保转移后采样引脚和通道引脚的电压值相等。

### 7. 缓冲时间 3

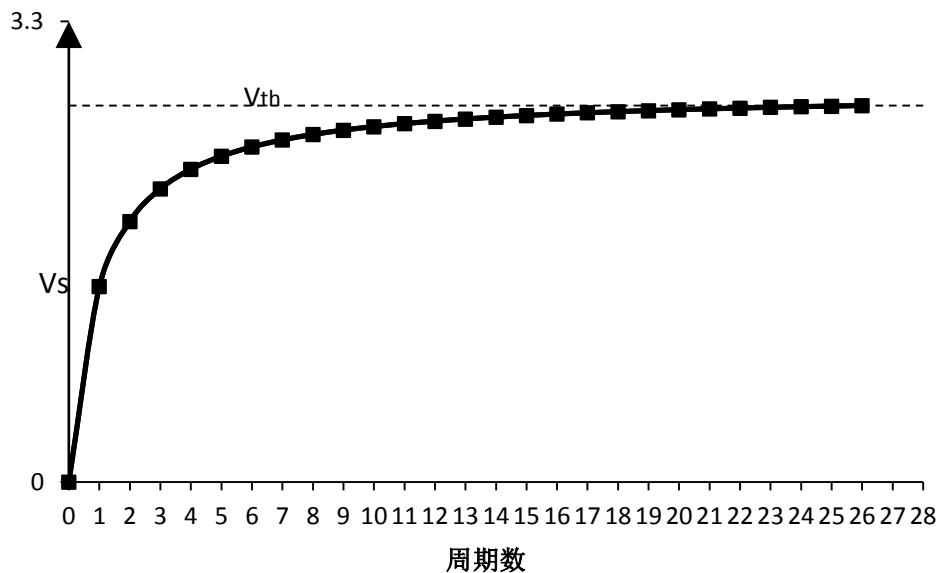
ASW1 和ASW2 断开，PIN1 被设置成浮空输入。

### 8. 对比

ASW1、ASW2 和 PIN1 仍然保留上一步骤的配置。在这一步将采样引脚PIN2 的电压  $V_s$  和一个门限电压  $V_{th}$  进行比较。如果  $V_s$  比  $V_{th}$  低，那么整个序列返回到步骤2开始新的循环，否则整个序列结束。

在步骤1完成之后，采样引脚的电压  $V_s$  是0，之后每经过一个充电、转移周期都会增大一些，如下图所示。根据电气学知识，如果电极上的电容越大，则每个周期  $V_s$  增大的值就会越大。电荷转移序列的停止条件是  $V_s$  达到  $V_{th}$ ，系统中有一个计数器来记录在一个电荷转移序列中具体的充电、转移周期次数，如果每个充电、转移周期  $V_s$  增加的值越大，那么总次数就会越小，软件可以从寄存器读出这个次数值来检测系统电容的变化。

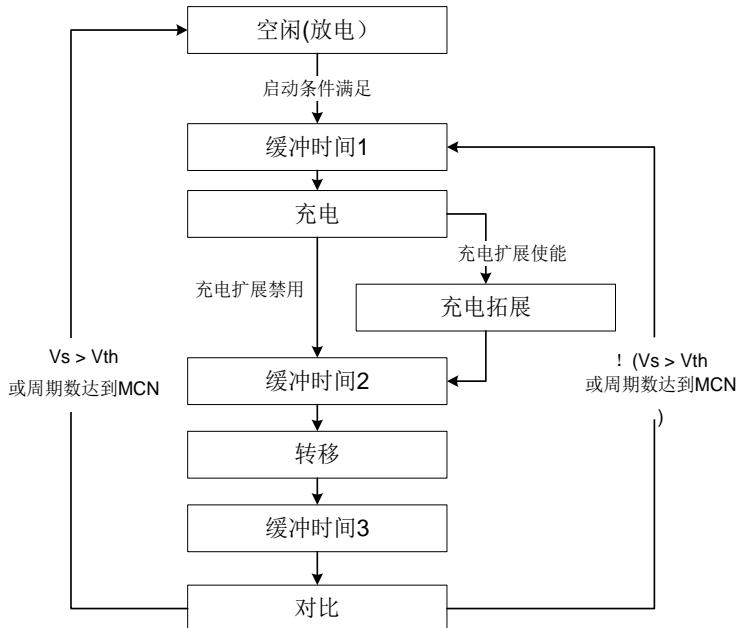
图 21-3. 电荷序列传输期间的采样 引脚的电压



#### 21.3.4. 电荷转移序列状态机

TSI内部有一个硬件实现的有限状态机，来执行如前一节中描述的电荷转移序列，如下图所示列。

图 21-4. 电荷转移序列的有限状态机的状态转移图



复位之后，状态机处于默认的IDLE状态。状态机有2种启动条件（由TSI\_CTLR寄存器的TMSEL位定义）：

**TMSEL = 0:** 软件触发模式。在这一模式下，软件通过将 TSI\_CTLR寄存器的TSIST位写1来启动状态机。

**TMSEL = 1:** 硬件触发模式。在这种模式下，状态机会监测TSITG引脚，当发现上升沿或下降

沿（极性可由寄存器配置）时自动启动。

状态机一旦启动就会按照上图所描述的跳转流程来运行。状态机如果在当前状态下的持续时间一旦达到配置值就会离开当前状态，并根据当前的条件进入下一个状态。只有当ECEN位被置1，才会出现充电扩展这一状态，这个状态用于实现扩频功能。

在比较状态，将状态机会比较每个已启用的引脚组中的采样引脚电压和阈值电压。如果所有采样引脚的电压都达到了阈值，状态机返回空闲状态并停止，否则，状态机返回到缓冲时间1状态，继续下一个周期，像上图所示，在27个周期之后， $V_s$ （采样引脚的电压）到达  $V_{th}$ （阈值电压）。

在TSI\_CTLR寄存器的MCN位也定义了一个最大周期数，当周期数达到 MCN，状态机也将返回 IDLE状态并停止，不管此时  $V_s$  是否达到了  $V_{th}$ 。

### 21.3.5. 状态时钟和持续时间

TSI模块里有3个时钟：HCLK、CTCLK和ECCLK。HCLK为系统时钟，它驱动TSI的寄存器和状态机。CTCLK是从HCLK分频得到的，（分频系数由CTCDIV定义），它是用来计算充电和转移持续时间的时钟。ECCLK也是从HCLK分频得到（分频系数由ECCDIV定义），它用来计算充电扩展的持续时间。除了充电扩展，其他的状态持续时间在每一次循环周期中都是固定的。其中，缓冲时间1、缓冲时间2和缓冲时间3的持续时间固定为2 HCLK周期。充电、传输状态的持续时间由CDT与CTDT寄存器定义。充电扩展状态的持续时间每个周期会发生变化，它的最大值由TSI\_CTLR寄存器的ECDT[6:0]位域定义。充电扩展状态在每个周期的持续时间变化规律如表21-2所示：

表 21-2. 充电扩展状态的持续时间

周期数	充电扩展状态持续时间
1	0
2	1
...	
ECDT	ECDT-1
ECDT+1	ECDT
ECDT+2	ECDT+1
ECDT+3	ECDT
ECDT+4	ECDT-1
...	...
2*ECDT+1	2
2*ECDT+2	1
2*ECDT+3	0
2*ECDT+4	1
2*ECDT+5	2
...	...

### 21.3.6. PIN 模式和 TSI 控制

每一个引脚组包括4个引脚，每个这些引脚任意一个都能够被用作采样引脚或通道引脚。但是在一个引脚组内，只有一个引脚应配置为采样引脚，通道引脚可以是多于一个，但任何情况下采样引脚和通道引脚都不能配置为同一个引脚。

一旦在GPIO中将一个引脚配置为TSI引脚，并在TSI中将其配置为采样或通道引脚，则在电荷转移序列中，该引脚的模式就由TSI控制。一般情况下，每个引脚有3种模式：输入、输出高和输出低。一个通道引脚或采样引脚在电荷转移序列期间的模式已在表21-1中具体描述，它们在序列空闲时的模式由TSI\_CTLR寄存器中的PINMOD位定义。此外，如果一个引脚在GPIO中配置为TSI引脚，但在TSI模块配置中它既不是采样引脚，也不是通道引脚，我们称之为自由引脚，该引脚的模式也由TSI\_CTLR寄存器的PINMOD位定义。

### 21.3.7. ASW 和迟滞模式

当状态机正在运行时，一个通道或采样引脚的模拟开关由电荷转移序列控制，如表21-1所示。当状态机处于空闲状态时，这些引脚的模拟开关由TSI\_ASW寄存器中对应的GxPy位控制，此外所有的自由引脚的模拟开关也由GxPy位控制（自由引脚的定义见上一节）。

值得注意的是，TSI模块会始终控制引脚的模拟开关，即使这些引脚在GPIO中没有配置成TSI引脚。用户可以利用这个特性，通过软件读写GxPy位来控制这些模拟开关，同时通过GPIO方式直接配置引脚的输出/输入模式，这样就可以实现用户自定义的电荷转移序列。TSI每个引脚的施密特触发器迟滞模式是由TSI\_PHMR寄存器GxPy位配置。

### 21.3.8. TSI 操作流

TSI的正常软件运行流程如下所示：

系统初始化，如系统时钟配置，TSI相关的GPIO配置等等。

按要求编程TSI\_CTLR, TSI\_CP, TSI\_IER, TSI\_SP和GEx位/位域TSI\_GCTRLR。

通过设置TSI\_CTLR寄存器的TSIEN位使能TSI。

通过设置TSI\_CEFR寄存器清除CTCF和MCE中断标志位。

选择触发方式：软件触发模式，设置TSIST位以开始充电传输序列；硬件触发模式，TSI由触发引脚的下降沿/上升沿启动。

等待TSI\_STR寄存器中的CTCF或MCE中断标志位置位。

读出TSI\_GxCYCNR寄存器的CYCN位。

### 21.3.9. TSI 标志和中断

表 21-3. TSI 错误和标志位

标识名称	描述	清零
------	----	----

CTCF	TSI 停止因为所有引脚组的采样电压 $V_s$ 都达到了阈值电压 $V_{th}$ .	TSI_CEFR 寄存器的 CCTCF 位
MNE	TSI 停止由于周期数到达最大值	TSI_CEFR 寄存器的 CMNE 位

### 21.3.10. TSI GPIOs

表 21-4. TSI 引脚

TSI 组	TSI 引脚	GPIO 引脚
第 1 引脚组	PIN1	PA0
	PIN2	PA1
	PIN3	PA2
	PIN4	PA3
第 2 引脚组	PIN1	PA4
	PIN2	PA5
	PIN3	PA6
	PIN4	PA7
第 3 引脚组	PIN1	PC5
	PIN2	PB0
	PIN3	PB1
	PIN4	PB2
第 4 引脚组	PIN1	PA9
	PIN2	PA10
	PIN3	PA11
	PIN4	PA12
第 5 引脚组	PIN1	PB3
	PIN2	PB4
	PIN3	PB6
	PIN4	PB7
第 6 引脚组	PIN1	PB11
	PIN2	PB12
	PIN3	PB13
	PIN4	PB14

## 21.4. TSI 寄存器

### 21.4.1. TSI 控制寄存器 (TSI\_CTLR)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDT[3:0]				CTDT[3:0]				ECDT[6:0]				ECEN			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECDIV	CTCDIV[2:0]			保留				MCN[2:0]		PINMOD	ES	TM	TSIST	TSIEN	
	rw			rw				rw		rw	rw	rw	rw	rw	

位/位域	名称	描述
31:28	CDT[3:0]	充电状态持续时间 CDT[3:0] 由软件置位和清除。这些位控制在电荷转移序列中充电状态的持续时间。 0000: $1 \times t_{CTCLK}$ 0001: $2 \times t_{CTCLK}$ 0010: $3 \times t_{CTCLK}$ .... 1111: $16 \times t_{CTCLK}$
27:24	CTDT[3:0]	电荷转移状态持续时间 CTDT[3:0] 由软件置位和清除。 这些位控制在电荷转移序列电荷转移状态的持续时间。 0000: $1 \times t_{CTCLK}$ 0001: $2 \times t_{CTCLK}$ 0010: $3 \times t_{CTCLK}$ .... 1111: $16 \times t_{CTCLK}$
23:17	ECDT[6:0]	充电扩展状态最大持续时间 ECDT[6:0] 由软件置位和清除。这些位控制充电扩展状态的最大持续时间。注：仅当 TSI_CTRL寄存器的ECEN位置1时才有充电扩展状态出现。 0000000: $1 \times t_{ECCLK}$ 0000001: $2 \times t_{ECCLK}$ 0000010: $3 \times t_{ECCLK}$ .... 1111111: $128 \times t_{ECCLK}$
16	ECEN	充电扩展状态使能 0: 充电扩展禁用 1: 充电扩展启用
15	ECCDIV	ECCLK时钟分频系数 ECCLK由HCLK分频得到， ECDIV定义了分频系数。 0: $f_{ECCLK} = f_{HCLK}$ 1: $f_{ECCLK} = 0.5f_{HCLK}$
14:12	CTCDIV[2:0]	CTCLK 时钟分频系数 CTCLK由HCLK分频得到， CTCDIV定义了分频系数。 000: $f_{CTCLK} = f_{HCLK}$

		001: $f_{CTCLK} = f_{HCLK}/2$ 010: $f_{CTCLK} = f_{HCLK}/4$ 011: $f_{CTCLK} = f_{HCLK}/8$ .... 111: $f_{CTCLK} = f_{HCLK}/128$
11:8	保留	必须保持复位值
7:5	MCN[2:0]	<p>一个序列的最大周期数</p> <p>MCN[2:0] 定义电荷转移序列的最大充电、转移周期数，一旦达到这个值，不论采样电压是否达到阈值电压序列都会停止。</p> <p>000: 255 001: 511 010: 1023 011: 2047 100: 4096 101: 8191 110: 16383 111: 保留</p>
4	PINMOD	<p>引脚模式</p> <p>该位定义在状态机为空闲状态时TSI引脚的模式，同时也定义自由引脚的模式。</p> <p>0: 输出低电平 1: 悬空输入</p>
3	ES	<p>边沿类型选择</p> <p>该位定义硬件触发模式下的边沿类型。</p> <p>0: 下降沿触发 1: 上升沿触发</p>
2	TM	<p>触发模式选择</p> <p>0: 软件触发模式，当TSIST位置位后，序列将开始 1: 硬件触发模式，序列将启动，在触发引脚检测到一个上升沿或下降沿</p>
1	TSIST	<p>TSI 启动</p> <p>在软件触发模式下，由软件置1，用于启动一次电荷转移序列，序列结束后硬件自动清零。软件也可以手动清零它来手动停止序列。</p> <p>0: TSI不启动 1: TSI启动</p>
0	TSIEN	<p>TSI 使能</p> <p>0: TSI模块启用 1: TSI模块禁用</p>

### 21.4.2. TSI 错误中断或标志中断使能寄存器 (TSI\_IER)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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															

rw rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	MNEIE	最大循环次数错误中断使能 0: MNE 中断禁用 1: MNE 中断被使能
0	CRCFIE	电荷转移完成标志中断使能 0: CCTCF 中断禁用 1: CCTCF 中断被使能

#### 21.4.3. TSI 清错误或标志寄存器 (TSI\_CEFR)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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															

w w

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CMNE	清除最大循环次数错误 0: 保留 1: 清MNE
0	CCTCF	清除电荷转移完成标志 0: 保留 1: 清CTCF

#### 21.4.4. TSI 状态寄存器 (TSI\_STR)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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 r

位/位域	名称	描述
31:2	保留	必须保持复位值
1	MNE	最大循环次数错误 因为到达MCN[2:0]定义的最大周期数，电荷转移序列停止。该位由硬件置位，软件可通过向TSI_CEFR寄存器的CMNE位写1来清零。 0: 没有最大循环次数错误 1: 最大循环次数错误
0	CTCF	电荷转移完成标志 因为所有的使能的组的采样引脚达到电压阈值或到达MCN[2:0]定义的最大周期数，电荷转移序列停止。该位由硬件置位，软件可通过向TSI_CEFR寄存器的CCTCF位写1来清零。 0: 电荷转移没有完成 1: 电荷转移完成

#### 21.4.5. TSI 引脚迟滞模式寄存器 (TSI\_PHMR)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rw															
G4P4	G4P3	G4P2	G4P1	G3P4	G3P3	G3P2	G3P1	G2P4	G2P3	G2P2	G2P1	G1P4	G1P3	G1P2	G1P1

---

31:24	保留	必须保持复位值
23:0	GxPy	引脚迟滞模式 该位由软件置位和清除。 0: Pin GxPy 施密特触发迟滞模式禁用 1: Pin GxPy 施密特触发迟滞模式使能

#### 21.4.6. TSI 模拟开关寄存器 (TSI\_ASWR)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
保留										G6P4	G6P3	G6P2	G6P1	G5P4	G5P3	G5P2	G5P1
										rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
G4P4	G4P3	G4P2	G4P1	G3P4	G3P3	G3P2	G3P1	G2P4	G2P3	G2P2	G2P1	G1P4	G1P3	G1P2	G1P1		
rw																	

---

位/位域	名称	描述
------	----	----

31:24	保留	必须保持复位值
23:0	GxPy	模拟开关状态 该位由软件置位和清除。 0: GxPy 的模拟开关断开 1: GxPy 的模拟开关闭合

#### 21.4.7. TSI 采样引脚寄存器 (TSI\_SPR)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
保留										G6P4	G6P3	G6P2	G6P1	G5P4	G5P3	G5P2	G5P1
										rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
G4P4	G4P3	G4P2	G4P1	G3P4	G3P3	G3P2	G3P1	G2P4	G2P3	G2P2	G2P1	G1P4	G1P3	G1P2	G1P1		
rw																	

---

位/位域	名称	描述
------	----	----

---

31:24	保留	必须保持复位值
23:0	GxPy	采样引脚模式 该位由软件置位和清除。 0: GxPy 引脚不是采样引脚 1: GxPy 引脚是采样引脚

#### 21.4.8. TSI 通道引脚寄存器 (TSI\_CPR)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4P4	G4P3	G4P2	G4P1	G3P4	G3P3	G3P2	G3P1	G2P4	G2P3	G2P2	G2P1	G1P4	G1P3	G1P2	G1P1
rw															

位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	GxPy	通道引脚模式 该位由软件置位和清除。 0: GxPy 引脚不是通道引脚 1: GxPy 引脚是通道引脚

#### 21.4.9. TSI 组控制寄存器 (TSI\_GCTLR)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	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/位域	名称	描述

---

31:24	保留	必须保持复位值
21:16	GCx	组完成 当一个使能组的电荷转移序列完成的时候，该位由硬件置位。当一个新的电荷转移序列开始的时候，由硬件清0。 0: 组 x 电荷转移没有完成 1: 组 x 电荷转移完成
15:6	保留	必须保持复位值
5:0	GEx	引脚组使能 该位由软件置位和清除。 0: 引脚组 x 禁用 1: 引脚组 x 使能

#### 21.4.10. TSI 组 x 周期数寄存器 (TSI\_GxCYCNR)

地址偏移: 0x30 + 0x04 x Group Number x(1-6)

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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		CYCN[13:0]													
rw															

---

位/位域	名称	描述
31:14	保留	必须保持复位值
13:0	CYCN[13:0]	周期数目 这些位反映了电荷转移序列完成时组x执行的周期数。当一个新的电荷转移序列开始的时候，由硬件清0。

## 22. HDMI-CEC 控制器 (HDMI-CEC)

### 22.1. 简介

消费电子控制 (CEC)是 HDMI(高清多媒体接口)标准的一部分。CEC 作为一种协议，提供了在用户环境中各种音像制品之间的高层次的控制功能。CEC 协议从 HDMI-CEC 控制器获得硬件支持。

### 22.2. 主要特性

- 符合HDMI-CEC1.4规范
- 32.768KHz 的CEC时钟具有两个时钟源可供选择：
  - 1) LSE 振荡器
  - 2) 固定分频后的HSI 振荡器 (HSI/244)
- 为了超低功耗应用，CEC控制器可工作在Deepsleep模式
- 可配置仲裁优先级的的SFT(信号空闲时间)：
  - 1) 用户配置
  - 2) 控制器根据HDMI-CEC协议规格自动配置
- 可编程的私有地址(OADR)
- 支持监听模式，在不干扰CEC总线情况下接收CEC总线上的数据
- RX位宽容度扩展功能支持更高的兼容性
- 位错误检测
  - 1) 短位错误(RSBPE)
  - 2) 长位错误(RLBPE)
  - 3) 位上升沿错误(RBRE)
- 可配置的错误位生成条件
  - 1) SBPE检测总会生成错误位
  - 2) LBPE检测只有在LBPEGEN=1的时候生成错误位
  - 3) BRE检测只在BREGEN=1的时候生成错误位
- 传输错误检测(TE)

- 传输欠载检测(TU)
- 接收过载检测(RO)
- 仲裁丢失检测(LSTAR), 带自动发送重试

## 22.3. 功能描述

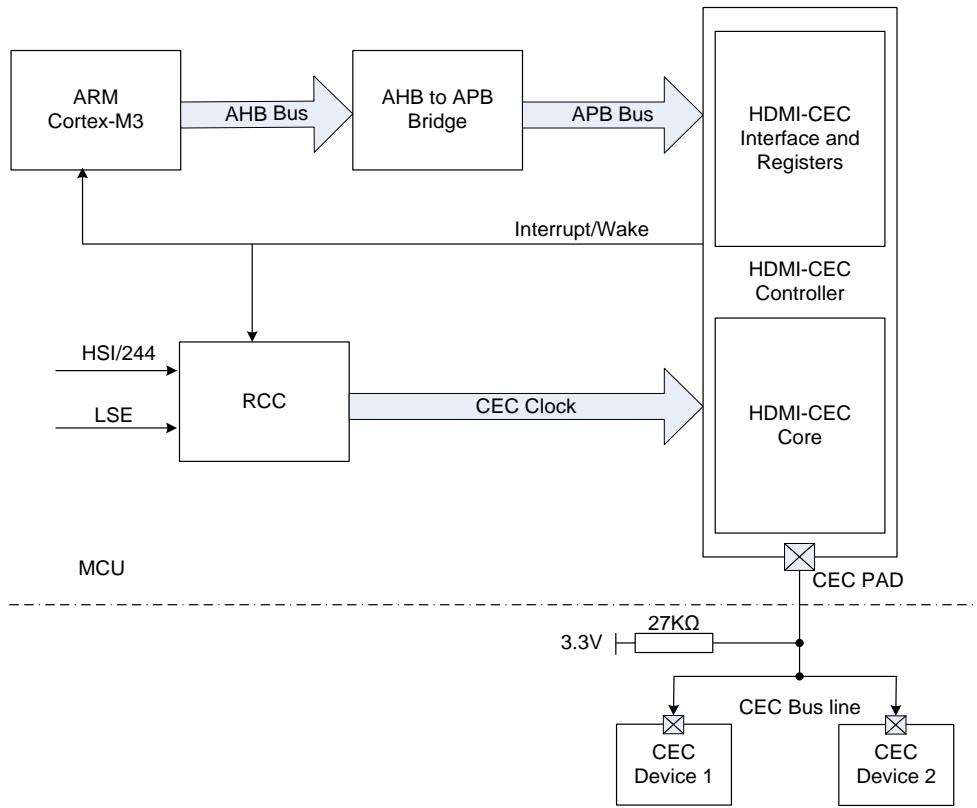
### 22.3.1. CEC 总线引脚

CEC控制器用单根双向线来发送和接收数据, CEC引脚通过外接一个 $27K\Omega$  的上拉电阻连接到 $+3.3V$ 电源电压。

控制器的输出级, 必须用漏极开路或集电极开路方式, 以实现逻辑与的线路连接。

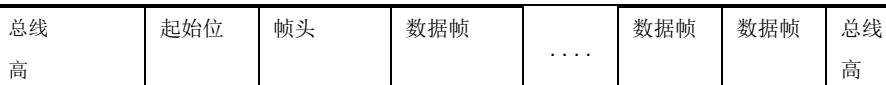
使用HDMI-CEC控制器时, 需要配置CEC引脚为AF Open drain模式, 并且需外接 $27K\Omega$ 的上拉电阻。

图 22-1. HDMI-CEC 控制器框图



### 22.3.2. 信息说明

完整的信息包括一帧或者多帧, 信息结构如下:



帧有两种类型：

- 1) **帧头**: 信息中紧接着起始位的第一帧，包含信号源的逻辑地址和信号目的地的逻辑地址
- 2) **数据帧**: 信息中除了帧头的其他帧

所有的帧长度都是10位并且有相同的基本结构如下：

帧结构									
7	6	5	4	3	2	1	0		
信息位								EOM	ACK

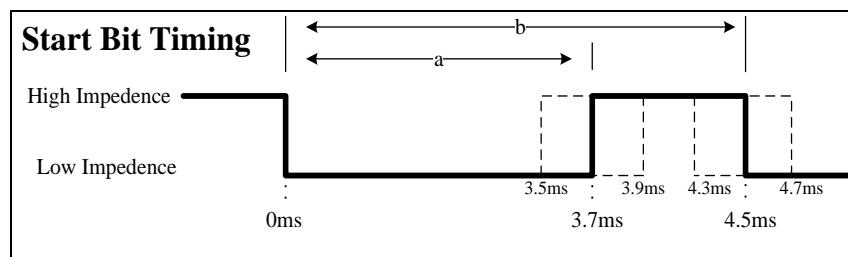
信息位是数据、指令还是地址取决于应用环境。

控制位EOM和ACK在一帧中会始终出现，并且所代表的意义相同。

### 22.3.3. 位时序说明

信息中所有位的时序都被分成两种类型：起始位和数据位

- 1) 起始位：起始位通过一段低电平持续时间(a)的来表示有效，其总周期(b)如下：



- 2) 数据位：有效数据位的时序约束如下：

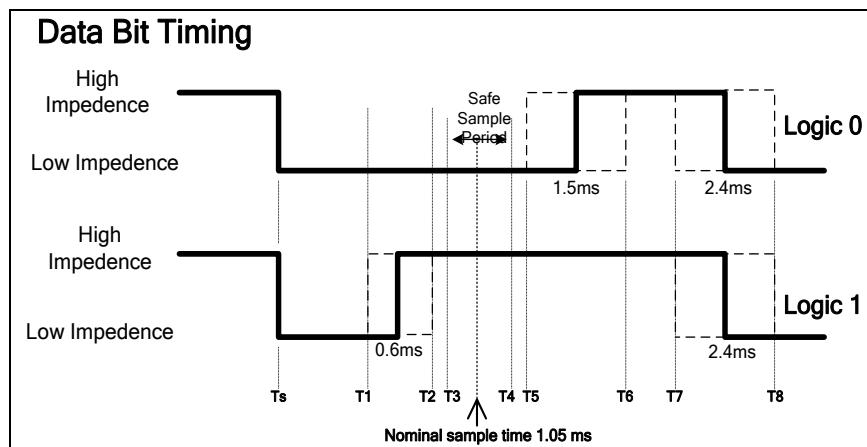


表 22-1. 数据位时序参数表

Ts	时间 (ms)	位起始
T1	0.4ms	传输逻辑1时, T1为从低电平到高电平过渡的最早时间
T2	0.8ms	传输逻辑1时, T2为从低电平到高电平过渡的最晚时间
T3	0.85ms	可以对信号线安全采样的最早时间

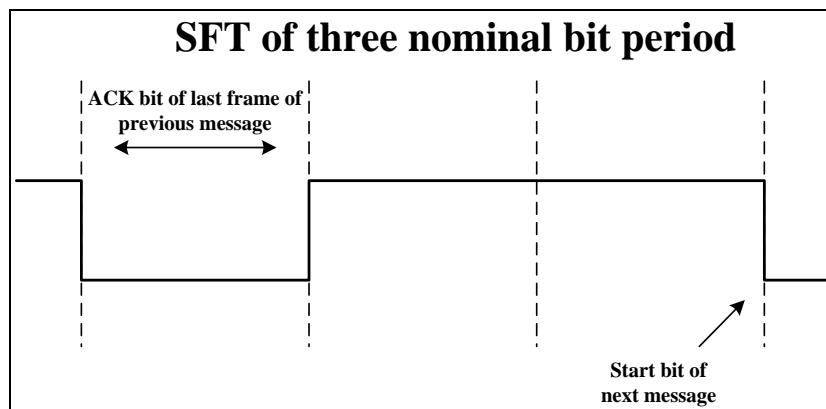
Ts	时间 (ms)	位起始
T4	1.25ms	可以对信号线安全采样的最晚时间
T5	1.3ms	T5为允许设备返回高阻态(逻辑0)的最早时间
T6	1.7ms	T6为允许设备返回高阻态(逻辑0)的最晚时间
T7	2.05ms	T7为下一位开始的最早时间
	2.4ms	额定数据位周期总长
T8	2.75ms	T8为下一位开始的最晚时间

### 22.3.4. 仲裁

CEC线仲裁阶段起始于起始位的前沿，结束于启动器地址位的末端。在此期间，启动器监视 CEC线。如果此时它监测到非自身驱动的低阻态，那么就认为失去了仲裁。



在试图传输或者再次传输一帧之前，CEC设备应该确保CEC总线信号已经空闲了大量时间。这个信号空闲时间的起点为先前帧结束位的开始时刻。



需要的信号空闲时间长度取决于当前控制信号线的状态和设备的初始值。

如果 SFT=0x0，HDMI-CEC控制器的SFT将有如下表现：

应用场景	信号空闲时间 (额定信号位周期)
先前的启动器在发送之前信息后想立即发送另一个信息	$\geq 7$
新的启动器想发送一个信息	$\geq 5$
先前的信息发送不成功	$\geq 3$

这意味着在当前设备完成发送当前信息之后，其他设备有机会在以上空闲时间内进入CEC线并发送自己的信息。

如果SFT不是0x0，会执行相应的用户所配置的SFT时间

### 22.3.5. SFTOPT 位说明

SFT选项位可通过设置更多的SFT计数器的启动时间点来节省总线闲置时间。

当SFTOPT = 0时， SOM位生效时， SFT定时器将在控制器处于闲置状态下启动。

当SFTOPT = 1时， SFT定时器将在CEC总线处于空闲状态下启动。此时如果在SFT结束后配置SOM，会节省SFT时间，因为控制器启动传输的时候不再需要等待时间。

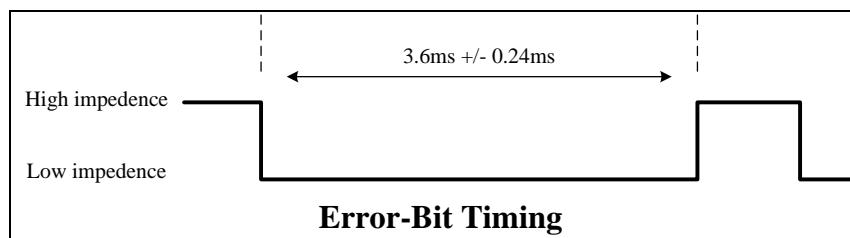
当SFTOPT = 1时，某些事件也可能启动SFT计数器：

- 常规TX/RX收发结束(TEND/REND有效)
- 传输没有结束例如TE, TAE或TU有效
- 接收过程中，如果监测到错误且产生错误位，SFT定时器在输出错误位结束后启动

### 22.3.6. 错误定义

#### 错误位

如果发生错误且相应产生配置使能，HDMI-CEC 控制器将在CEC引脚上产生一个错误位来提示错误。错误位周期定义如下：



#### 帧错误

CEC协议规定信息的每一帧需要受到确定信号来确保传输成功。对于广播(目的地址=0xF)来说，ACK位应该为逻辑1；对于单播(目的地址<0xF)来说，ACK位应该为逻辑0，否则将产生帧错误(TAE/RAE标志位生效)

另一个帧错误产生的情况是当HDMI-CEC控制器处于启动器阶段时，CEC总线电平和CEC控制器输出电平不同(TE会置位)。

#### 位上升错误(RBRE)

如果在BRE检查窗口的时候监测到上升沿，BRE标志位会置位；如果BREIE=1，BRE标志位也会产生CEC中断。

如果BRESTP=1，控制器将停止接受信息，RBREGEN=1，错误位将产生。

如果BRESTP=1，在广播时BRE标志位会置位，并且会产生错误位来通知启动器有错误发生。如果不产生监测BRE的错误位，可以配置BREGEN=0且BCNG=1。

**注：**不可同时配置BRESTP=0和BREGEN=1

#### 短位错误 (RSBPE)

当相邻的下降沿周期比预期短的时候，SBPE被置1。

如果SBPEIE=1， SBPE标志位也会产生CEC中断。

如果SBPE错误标志置位，一定会输出错误位，除了以下情况：

- (1). BCNG = 1
- (2). LSTNM = 1
- (3). 接收广播

### 长位错误(RLBPE)

当相邻下降沿周期比预期长时， LBPE位被置1。

如果LBPEIE=1， LBPE标志位也会产生CEC中断。

如果LBPE标志位置位，控制器将会停止接收信息并产生错误位除非以下情况发生：，

- (1). 单次播放和广播模式下LBPEGEN=1
- (2). 广播模式下BCNG=0

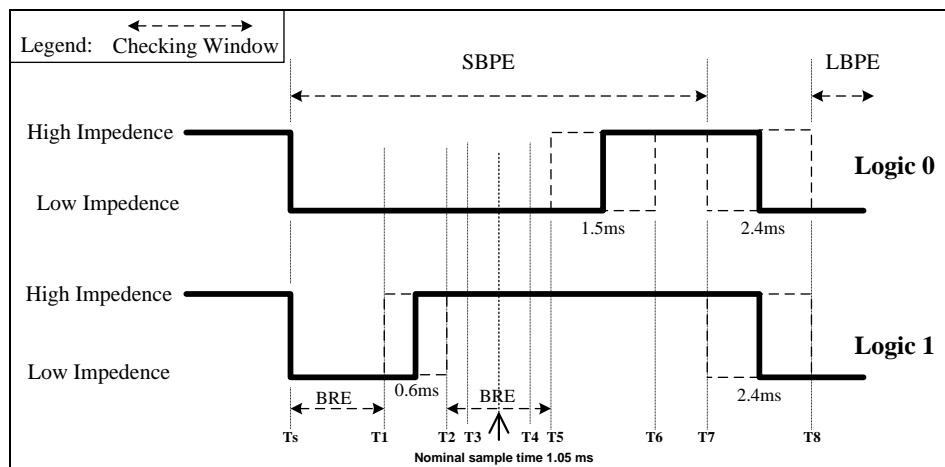


表 22-2. 错误处理时序参数表

时间	RTOL	ms	说明
Ts	-	0ms	开始传输时刻
T1	1	0.3ms	传输逻辑1的低电平到高电平的转换最早时间
	0	0.4ms	
T2	0	0.8ms	传输逻辑1的低电平到高电平转换的最晚时间
	1	0.9ms	
T3	-	0.85ms	对信号线安全采样的最早时间
T4	-	1.25ms	对信号线安全采样的最晚时间
T5	1	1.2ms	允许设备返回到高阻态(逻辑0)的最早时间
	0	1.3ms	
T6	0	1.7ms	允许设备返回到高阻态(逻辑0)的最晚时间
	1	1.8ms	
T7	1	1.85ms	下一位开始传输的最早时间

时间	RTOL	ms	说明
	0	2.05ms	
		2.4ms	额定数据位周期
T8	0	2.75ms	下一位开始传输的最晚时间
	1	2.95ms	

### 传输错误监测(TE)

当启动器在CEC总线传输高组态的时候监测到总线呈低阻态时，TE被置1。

如果TEIE=1时，TE也会产生CEC中断。

当TE置位时，传输停止，软件可以重启传输。

TE检查窗口依赖于帧的不同位状态如下：

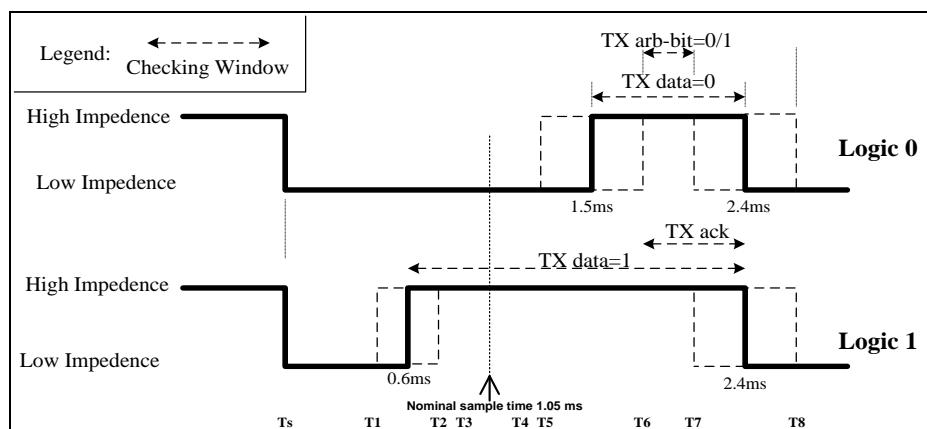


表 22-3. TE 时序参数表

时间	RTOL	ms	说明
Ts	-	0ms	传输开始位
T1	1	0.3ms	传输逻辑1时低电平到高电平的转换最早时间
	0	0.4ms	
T2	0	0.8ms	传输逻辑1时低电平到高电平转换的最晚时间
	1	0.9ms	
T3	-	0.85ms	信号线安全采样的最早时间
T4	-	1.25ms	信号线安全采样的最晚时间
T5	1	1.2ms	允许设备返回到高阻态(逻辑0)的最早时间
	0	1.3ms	
T6	0	1.7ms	允许设备返回到高阻态(逻辑0)的最晚时间
	1	1.8ms	
T7	1	1.85ms	下一位开始传输的最早时间
	0	2.05ms	
		2.4ms	额定数据位周期
T8	0	2.75ms	下一位开始传输的最晚时间
	1	2.95ms	

### 22.3.7. HDMI-CEC 中断

HDMI-CEC控制器中有13个中断，每个中断由相应的标志位和中断使能位组成。

编号	HDMI-CEC中断事件	事件标志位	中断使能位
1	丢失仲裁	LSTARB	LSTARBIE
2	传输字节请求	TBR	TBRIE
3	传输结束	TEND	TENDIE
4	传输字节缓冲区欠载	TU	TUIE
5	传输错误	TE	TEIE
6	传输确认错误	TAE	TAEIE
7	接收字节请求	RBR	RBRIE
8	接收结束	REND	RENDIE
9	接收过载	RO	ROIE
10	接收位上升沿错误	RBRE	RBREIE
11	接收位短位错误	RSBPE	RSBPEIE
12	接收位长位错误	RLBPE	RLBPEIE
13	接收位确认错误	RAE	RAEIE

注： HDMI-CEC的任何中断都将使芯片从深度睡眠模式唤醒。

## 22.4. HDMI-CEC 寄存器

### 22.4.1. CEC 控制寄存器 (CEC\_CTLR)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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										EOM	SOM	CECON	rs	rs	rw

位/位域	名称	描述
31:3	保留	必须保持复位值
2	EOM	发送模式下下一帧的EOM位值 当CECON=1时EOM只能被软件写入。EOM被硬件清0，SOM也一样。 0: 在EOM位下一帧发送0 1: 在EOM位下一帧发送1

1	SOM	启动发送一帧信息 当CECON=1时SOM只能被软件写入。在以下任意标志位置位以后，SOM被硬件清0：TEND, TU, TAE, TE 和CECON=0 如果信息只有一帧，配置txdata之前EOM应该被置1。SOM被置1以后，SFT计数器启动，当SFT计数结束，Start-bit将输出到CEC总线上。软件可以在SOM=1的时候通过清除CECON位来取消发送。 0：无CEC传输 1：CEC传输挂起或者正在进行
0	CECON	使能/禁止HDMI-CEC控制位。 CECON位由软件配置。 0：禁止HDMI-CEC控制器，取消任何信息发送并清除EOM/SOM位 1：使能CEC控制器，如果SOM=0，进入接收状态

## 22.4.2. CEC 设置寄存器 (CEC\_SR)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32位）访问

**注意：**该寄存器只有在 CECON=0 的时候才能写入

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LSTNM	OADR [14:0]															
rw															rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留								SFT MOPT	BCNG	RLBPEGEN	RBRE GEN	RBRE STP	R TOL	SFT[2:0]		
rw																

位/位域	名称	描述
31	LSTNM	监听模式使能位 软件置1和清0。 0：只接收OADR中带有正确ACK的单播信息或广播信息 1：接收OADR的信息时给出正确ACK：接收不是OADR的信息，不发送ACK
30:16	OADR[14:0]	自身地址 OADR的每一位代表一个目的地址。例如，如果oar[0]=1，控制器将接收发送目的地址为0x0的信息。这意味着控制器可以配置为多个私有地址。广播信息始终被接收。 接收到目的地址(帧头的后4位)以后，如果在oar中声明了，控制器将收到积极反馈信息，如果目的地址不在oar中但LSTN=1，控制器将接收信息但不反馈ACK。如果在oar中没有声明且LSTN=0，控制器将不会接收该信息。
15:9	保留	必须保持复位值
8	SFTMOPT	SFT选项位

		软件置1和清0。
		0: SOM置1后SFT计数器开始计数 1: 发收/接收结束后SFT计数器自动启动
7	BCNG	广播信息模式下不产生错误位  软件置1和清0。  0: 广播模式下, 当LSTN=1时BRE和LBPE将在CEC总线上产生错误位, SBPE也将产生错误位 1: 以上条件下不产生错误位
6	RLBPEGEN	在单播模式下监测到LBPE的时候产生错误位  软件置1和清0。  0: 在单播模式下监测到LBPE时不产生错误位 1: 在单播模式下监测到LBPE时产生错误位
5	RBREGEN	在单次传播模式下监测到BRE的时候产生错误位  软件置1和清0。  0: 在单播模式下监测到BRE的时候不产生错误位 1: 在单播模式下监测到BRE的时候产生错误位
4	RBRESTP	监测到BRE时是否停止接收信息  软件置1和清0。  0: 不停止接收BRE下的信息, 数据位按额定时间采样 1: 停止接收BRE下的信息.
3	RTOL	接收位时间宽容度  软件置1和清0  0: 标准接收位时间宽容度 1: 扩展接收位时间宽容度.
2:0	SFT[2:0]	信号空闲时间  软件置1和清0。  如果SFT=0x0, SFT时间跟HDMI-CEC协议描述的一样, 否则, SFT时间被软件固定配置。ACK位传输的下降沿启动计时。  0x0: <ul style="list-style-type: none"><li>- 3个额定数据位周期。如果SFT计数器是由于传输不成功而启动((LSTARB=1,TE=1,TU=1或TAE=1))</li></ul> 0x1: 1.5个额定数据位周期 0x2: 2.5个额定数据位周期 0x3: 3.5个额定数据位周期 0x4: 4.5个额定数据位周期 0x5: 5.5个额定数据位周期 0x6: 6.5个额定数据位周期 0x7: 7.5个额定数据位周期

### 22.4.3. CEC 数据发送寄存器 (CEC\_TDTR)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	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								TXDATA[7:0]							

w

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	TXDATA[7:0]	发送数据寄存器, 只写。

### 22.4.4. CEC 数据接收寄存器 (CEC\_RDTR)

地址偏移: 0xC

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	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								RXDATA[7:0]							

r

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	RXDATA[7:0]	接收数据寄存器 只读, 该位包含从CEC线接收的最后数据字节。

### 22.4.5. CEC 中断状态寄存器 (CEC\_ISTR)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	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	TAE	TE	TU	TEND	TBR	LSTAR	RAE	RLBPE	RSBPE	RBRE	RO	REND	RBR		

rc\_w1      rc\_w1

位/位域	名称	描述
31:13	保留	必须保持复位值
12	TAE	<p>发送ACK错误标志位</p> <p>硬件置1，软件写1清0。</p> <p>单播模式下ACK位作为1被接收及广播模式下作为0被接收将使该标志位生效。TAE将停止发送信息并清除SOM和EOM。</p>
11	TE	<p>发送错误</p> <p>该位由硬件置1，软件写1清零。</p> <p>如果控制器在启动器阶段下且监测到CEC线上非控制器输出的低阻态，TE置位。TE将停止发送信息且清除SOM和EOM。</p>
10	TU	<p>发送数据欠载</p> <p>该位由硬件置1，软件写1清零。</p> <p>在发送下一字节之前如果软件没有写数据，TU生效。TU将停止发送信息并且清除SOM和EOM</p>
9	TEND	<p>发送成功结束</p> <p>该位由硬件置1，软件写1清零。</p> <p>如果信息的所有帧成功传输，TEND位有效，TEND将清除SOM和EOM位。</p>
8	TBR	<p>发送字节数据请求</p> <p>该位由硬件置1，软件写1清零。</p> <p>当前帧的第四位传输结束后，TBR位生效，软件需要在6个额定数据位周期内写数据到txdata。</p>
7	LSTAR	<p>仲裁丢失</p> <p>该位由硬件置1，软件写1清零。</p> <p>以下任一情况发生时ARBLST位生效：外部CEC设备在控制器处于SFT状态时为了发送起始位拉低CEC线，或者控制器和CEC设备同时发送起始位但是控制器的启动器地址优先级更低。</p> <p>如果ARBLST生效，控制器将进入接收状态，接收信息结束后控制器将重新尝试发送消息。发送和接收期间，SOM始终被置1。</p>
6	RAE	<p>接收ACK错误</p> <p>该位由硬件置1，软件写1清零。</p> <p>以下情况RAE生效，在广播模式下如果ACK=0；或者LSTN=1且目的地址不是OADR的单播模式下，ACK=1。RAE将停止接收信息。</p>

5	RLBPE	长位周期错误 该位由硬件置1，软件写1清零。 数据位周期超过最大时间范围的时候LBPE生效。LBPE将停止接收信息且在以下情况产生错误位：单次传播模式下LBPEGEN=1，或广播模式下BCNG=0。
4	RSBPE	短位周期错误 该位由硬件置1，软件写1清零。 数据位周期小于最小周期的时候SBPE位生效。
3	RBRE	位上升错误 该位由硬件置1，软件写1清零。 期望时间外产生上升沿时，BRE生效。
2	RO	接收过载 该位由硬件置1，软件写1清零。 接收新的字节且RBR被置1时RO生效。 RO将停止接收信息并发送错误ACK位。
1	REND	接收结束 该位由硬件置1，软件写1清零。 当控制器完整接收到带有成功ACK反馈的全部信息的时候REND置位。REND和RBR同时生效。
0	RBR	接收字节数据接收 该位由硬件置1，软件写1清零。 当控制器接收到带有成功ACK反馈的数据的时候RBR生效，此时rxdata有效。

#### 22.4.6. CEC 中断使能寄存器 (CEC\_IER)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	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	TAEIE	TEIE	TUIE	TXENDIE	TBRIE	LSTARBIE	RAEIE	RLBPEIE	RSBPEIE	RBREIE	ROIE	RENDIE	RBRIE		

位/位域	名称	描述
31:13	保留	必须保持复位值
12	TAEIE	TAE中断使能 软件置1和清0。 0: TAE中断禁止 1: TAE中断允许

11	TEIE	TE中断使能 软件置1和清0。 0: TE中断禁止 1: TE中断允许
10	TUIE	TU中断使能 软件置1和清0。 0: TU中断禁止 1: TU中断允许
9	TENDIE	TEND中断使能 软件置1和清0。 0: TEND中断禁止 1: TEND中断允许
8	TBRIE	TBR中断使能 软件置1和清0。 0: TBR中断禁止 1: TBR中断允许
7	LSTARBIE	ALRLST中断使能 软件置1和清0。 0: ARBLST中断禁止 1: ARBLST中断允许
6	RAEIE	RAE中断使能 软件置1和清0。 0: RAE中断禁止 1: RAE中断允许
5	RLBPEIE	LBPE中断使能 软件置1和清0。 0: LBPE中断禁止 1: LBPE中断允许
4	RSBPEIE	SBPE中断使能 软件置1和清0。 0: SBPE中断禁止 1: SBPE中断允许
3	RBREIE	BRE中断使能 软件置1和清0。 0: BRE中断禁止 1: BRE中断允许
2	ROIE	RO中断使能 软件置1和清0。 0: RO中断禁止

1: RO中断允许

1	RENDIE	RENDE中断使能 软件置1和清0。 0: REND中断禁止 1: REND中断允许
0	RBRIE	RBR中断使能 软件置1和清0。 0: RBR中断禁止 1: RBR中断允许

## 23. LCD 控制器 (LCD)

本章内容适用于 GD32F170xx 和 GD32F190xx 产品

### 23.1. 简介

LCD驱动器通过自动产生SEG和COM交流电压信号来直接驱动LCD显示。该驱动器可以驱动单色液晶显示器（LCD），这是一种由若干段（像素或完整的符号）构成的，有可见和不可见两种状态的显示屏。LCD驱动器支持最大32个SEG和8个COM。

### 23.2. 主要特性

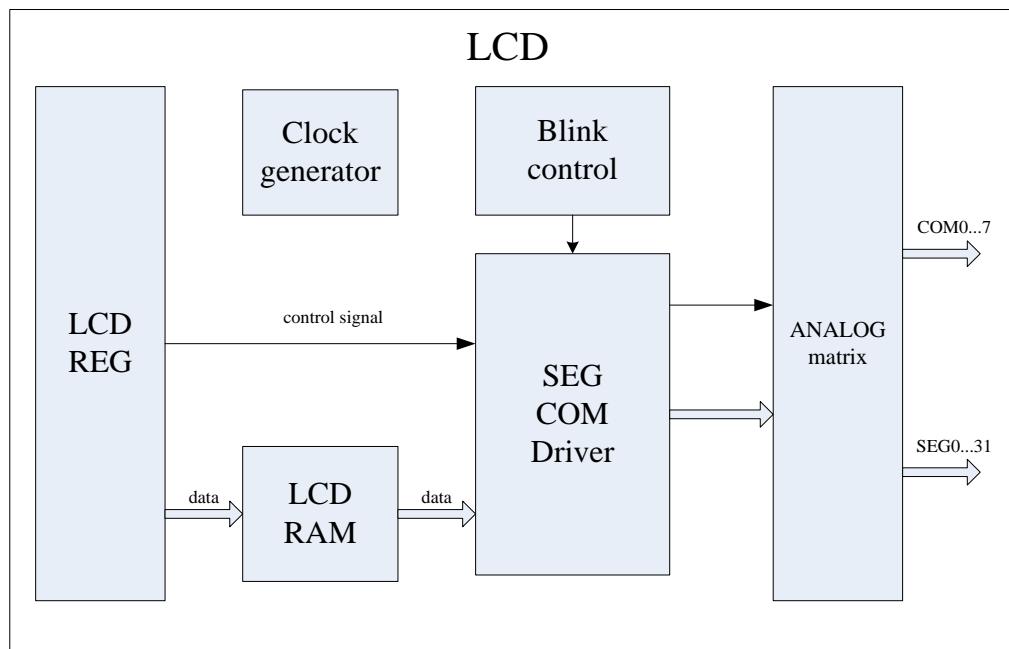
- 可配置帧率
- 单个SEG或所有SEG的闪烁
- 支持静态、1/2、1/3、1/4、1/6和1/8占空比
- 支持1/2、1/3和1/4偏置
- 双路缓冲器可多达8x32位寄存器来存储LCD\_DATAx
- 对比度也可通过配置死区时间来调整
- $V_{LCD_{rail}}$ 解耦能力

### 23.3. 功能描述

#### 23.3.1. LCD 架构

LCD控制器框图如下所示：

图 23-1. LCD 模块框图



LCD REG是LCD控制器的寄存器，包括LCD\_CTLR、LCD\_CFGR、LCD\_SFR、LCD\_ICR和LCD\_DATAx五个寄存器，它们可通过APB总线配置，且可使CPU产生中断。

时钟发生器可以从输入时钟产生LCD时钟，LCD时钟可以驱动闪烁控制和SEG/COM驱动器。闪烁控制可以产生闪烁频率和闪烁像素，SEG/COM驱动器可产生SEG和COM信号输送到ANALOG矩阵，且ANALOG矩阵可实现SEG和COM电压。

### 23.3.2. 时钟发生器

LCD输入时钟与RTCCLK共用，允许三种不同的时钟源：通过配置RCC\_BDCR寄存器的RTCSRC位域，可以选择LSE、LSI或HSE32分频。输入时钟频率变化范围为32KHz到1MHz。

LCD控制器可使用从集成时钟分频器输出的LCD时钟信号来产生SEG和COM线的时序。LCD时钟信号来自于RCCU输入时钟，输入时钟频率范围为32KHz到1MHz。LCD时钟频率可在LCD\_CTLR寄存器中的PRE位和DIV位来选择，由此产生的LCD时钟频率计算结果如下：

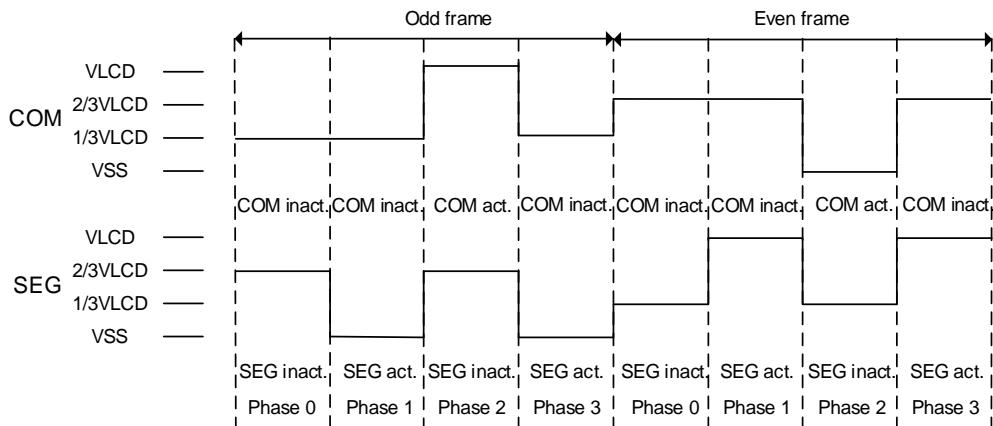
$$f_{LCD} = \frac{f_{in\_clk}}{2^{PRE} \times (DIV + 16)}$$

LCD时钟作为LCD控制器的时钟基准。LCD的时钟频率相当于相频率。一个LCD帧是一个奇数帧或一个偶数帧，它们拥有与有效的COM一样多的相。占空比定义为1/(LCD屏显示需要的COM数)。因此帧频率计算结果如下：

$$f_{frame} = f_{LCD} \times Duty$$

在帧起始，LCD\_SFR寄存器的SOF位由硬件置位，如果LCD\_CFGR寄存器的SOFIE位被置位，LCD中断将被执行。向LCD\_ICR寄存器的SOFC位写1将清除SOF位。

图 23-2. 1/3 偏置, 1/4 占空比



### 23.3.3. 闪烁控制

LCD控制器也支持闪烁功能。闪烁模式可通过LCD\_CFGR寄存器中的BLKM位来控制，BLKM = 01表示允许在SEG0和COM0上闪烁单个段，BLKM = 10表示允许闪烁所有COM和SEG0，BLKM = 11表示允许闪烁所有COM和所有SEG，BLKM = 00表示禁用闪烁。

闪烁频率源于LCD时钟，可通过LCD\_CFGR中BLKDIV位来选择，由此产生的BLINK频率计算结果如下：

$$f_{\text{BLINK}} = \frac{f_{\text{LCD}}}{2(\text{BLKDIV}+3)}$$

在选择BLKM = 01, 10或11中的一种闪烁模式后，使能的SEG或所有SEG都会在下一帧分界变成空白，且会停留半个BLKCLK周期，然后它们会在一个帧分界再次变白之前完成下一帧分界激活并继续停留另半个BLKCLK周期时间。

### 23.3.4. SEG/COM 驱动器

SEG/COM驱动器可以产生SEG和COM信号。

**偏置发生器：**

偏置可通过LCD\_CTLR寄存器中BIAS位来选择，仅在对应的一个帧周期的段内可以达到最大幅值VLCD或VSS。奇数帧电压和偶数帧电压如下表所示：

表23-1. 奇数帧电压

偏置	静态	1/2 偏置	1/3 偏置	1/4 偏置
<b>COM 有效</b>	VLCD	VLCD	VLCD	VLCD
<b>COM 无效</b>	/	1/2 VLCD	1/3 VLCD	1/4 VLCD
<b>SEG 有效</b>	VSS	VSS	VSS	VSS
<b>SEG 无效</b>	VLCD	VLCD	2/3 VLCD	1/2 VLCD

表23-2. 偶数帧电压

偏置	静态	1/2 偏置	1/3 偏置	1/4 偏置
<b>COM 有效</b>	VSS	VSS	VSS	VSS

<b>COM 无效</b>	/	1/2 VLCD	2/3 VLCD	3/4 VLCD
<b>SEG 有效</b>	VLCD	VLCD	VLCD	VLCD
<b>SEG 无效</b>	VSS	VSS	1/3 VLCD	1/2 VLCD

### COM信号:

COM信号可通过LCD\_CTLR寄存器中的DUTY位来选择。当DUTY是000时，静态占空比被选择，仅COM[0]被使用，且仅一段在奇数帧或偶数帧内，COM[0]驱动器信号一直有效。当DUTY是001时，仅COM[1:0]和2段被使用。当DUTY是010时，仅COM[2:0]和3段被使用。当DUTY是011时，仅COM[3:0]和4段被使用。当DUTY是100时，COM[7:0]和8段被使用。当DUTY是101时，COM[5:0]和6段被使用。所有的COM信号驱动器如下表所示：

表23-3. 所有COM信号驱动器

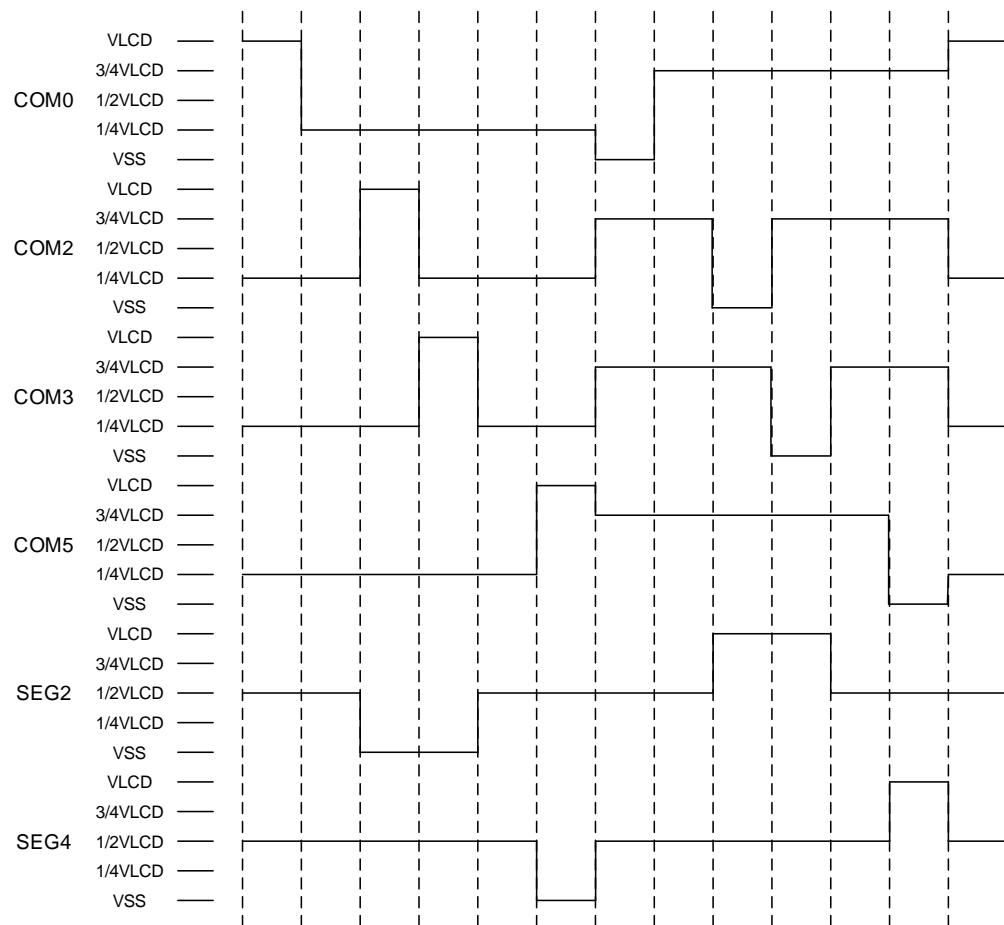
段	1	2	3	4	5	6	7	8
<b>COM0</b>	有效	无效						
<b>COM1</b>	无效	有效	无效	无效	无效	无效	无效	无效
<b>COM2</b>	无效	无效	有效	无效	无效	无效	无效	无效
<b>COM3</b>	无效	无效	无效	有效	无效	无效	无效	无效
<b>COM4</b>	无效	无效	无效	无效	有效	无效	无效	无效
<b>COM5</b>	无效	无效	无效	无效	无效	有效	无效	无效
<b>COM6</b>	无效	无效	无效	无效	无效	无效	有效	无效
<b>COM7</b>	无效	有效						

### SEG信号:

SEG信号从LCD\_DATAx寄存器中读取。LCD\_DATAx表示相x的SEG信号数据。当该位为1时，对应的SEG驱动有效信号，当该位为0时，对应的SEG驱动无效信号。

例如，应用程序需要激活像素COM2-SEG2, COM3-SEG2和COM5-SEG4，则LCD\_DATA2寄存器的bit2、LCD\_DATA3寄存器的bit2、LCD\_DATA5寄存器的bit4应被置位。这样SEG2信号将在每个奇数帧与偶数帧的第三、四相变为有效，SEG4信号将在每个奇数帧与偶数帧的第六相变为有效。有效与无电压请参考表23-1和表23-2。图23-3展现了当配置为1/4偏置，1/6占空比时的SEG信号。

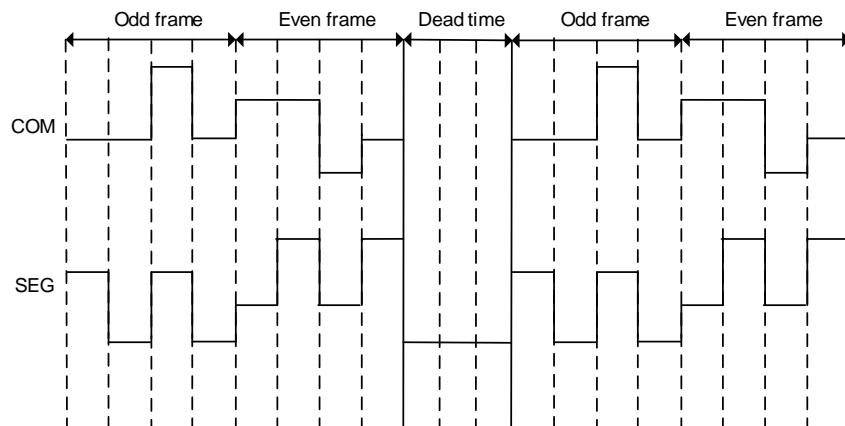
图 23-3. 1/4 偏置 1/6 占空比



#### 死区时间:

死区时间通过LCD\_CFGR寄存器中DD位来配置，它在每个偶数帧后插入VSS，插入的段数时间由DD位定义。应用程序可以通过配置死区时间调节对比度。

图 23-4. LCD 死区时间 (1/3 偏置, 1/4 占空比)



### 23.3.5. 双缓冲存储

双缓冲存储用于确保显示信息的连续性。

应用程序通过修改 `LCD_DATAx` 寄存器组访问第一缓冲区。在将显示信息写入 `LCD_DATAx` 寄存器组后，应用程序需要将 `LCD_SR` 寄存器的 `UPRF` 位置位，之后硬件将数据从第一缓冲区传入第二缓冲区，在传输的这段时间内，`UPRF` 位将保持置位，同时 `LCD_DATAx` 寄存器组将写保护。当传输结束后，`UPRF` 位被清除同时 `UPDF` 位将被硬件置位，如果 `UPDIE` 位被置位，将会产生一个中断。`SEG` 信号由第二缓冲区内的数据驱动，因此写 `LCD_DATAx` 寄存器组将不会对显示信息造成影响。

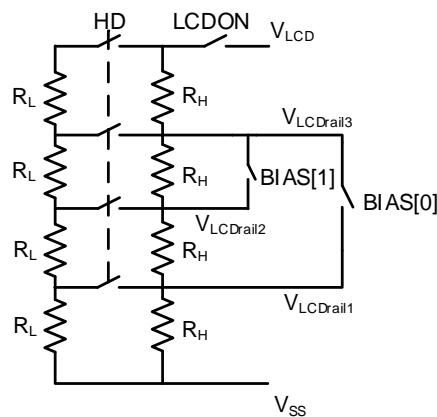
如果 `UPRF` 位在显示失能 (`LCDON=0`) 时被置位，传输将不会发生，直到 `LCDON` 置位。

### 23.3.6. 模拟矩阵

模拟矩阵提供LCD电压。LCD电压电平可由VLCD引脚或内部电压升压转换器产生（由 `LCD_CTLR` 寄存器中 `VSRC` 位选择）。当使用内部电压源时，`VLCD` 的值可以通过配置 `LCD_CFGR` 寄存器的 `CONR[2:0]` 位域从 `VLCD0` 到 `VLCD7` 中选择 (`VLCDx` 的值请参考产品数据手册)。应用程序可以通过改变 `VLCD` 的值调节对比度。另外一种调节对比度的方法是使用死区时间。

模拟矩阵通过如图23-5所示的内部电阻分压网络提供 `VSS` 和 `VLCD` 中间的电压电平 (1/3 `VLCD`、2/3 `VLCD` 或 1/4 `VLCD`、2/4 `VLCD` 和 3/4 `VLCD`)。

图 23-5. 电阻分压网络



在转换期间，为了快速到达静态状态，低值电阻 ( $R_L$ ) 被使用以增加电流。之后低值电阻被关闭，高值电阻 ( $R_H$ ) 被使用以减小功耗。 $R_L$  被使用的时间长度依据 `LCD_CFGR` 寄存器的 `PULSE[2:0]` 位域的值。通过配置 `LCD_CFGR` 寄存器的 `HD` 位， $R_L$  可以一直被使用。

#### 外部解耦

通过配置 `SYSCFG_R2` 寄存器的 `LCD_DECA` 位域，`VLCD` 内部电压轨道 (`VLCDrail1`, `VLCDrail2`, `VLCDrail3` 如图 23-5) 能够被连接到 GPIO 口。在这些 GPIO 引脚增加去耦电容可以帮助稳定电压以获得更高的对比度，此时 `PULSE[2:0]` 被允许选择更低的值以减少功耗。

**注意：**当使用 `VLCDrail` 功能时，GPIO 应该被配置为模拟输入模式。`SEG` 复用功能与 `VLCDrail` 解耦功能不能同时被使用。

表23-4. VLCDrail 连接引脚

	偏置			引脚
	1/2	1/3	1/4	
<b>VLCDrail1</b>	1/2VLCD	1/3VLCD	1/4VLCD	PB12
<b>VLCDrail2</b>	1/2VLCD	2/3VLCD	1/2VLCD	PB2
<b>VLCDrail3</b>	1/2VLCD	2/3VLCD	3/4VLCD	PB0

## 23.4. LCD 寄存器

### 23.4.1. LCD 控制寄存器 (LCD\_CTLR)

偏移地址: 0x00

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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								COM S	BIAS[1:0]	DUTY[2:0]	VSR C	LCD N				
								rw	rw	rw	rw	rw				

位/位域	名称	描述
31:8	保留	必须保持复位值
7	COMS	<p>COM/SEG引脚选择</p> <p>该位用于COM/SEG引脚的选择。当占空比选择1/8或1/6时，LCD_COM[7:4]总是作为LCD_COM[7:4]功能，无论该位有无被置位。</p> <p>0: LCD_COM[7:4]引脚选择LCD_COM[7:4] 1: LCD_COM[7:4]引脚选择LCD_SEG[31:28]</p>
6:5	BIAS[1:0]	<p>偏置选择</p> <p>偏置为驱动LCD时的电压水平参数。它被定义为1/（驱动LCD显示屏能够使用的电压水平总数-1）。</p> <p>00: 1/4偏置（5个电压水平: VSS, 1/4VLCD, 1/2VLCD, 3/4VLCD, VLCD） 01: 1/2偏置（3个电压水平: VSS, 1/2VLCD, VLCD） 10: 1/3偏置（4个电压水平: VSS, 1/3VLCD, 2/3VLCD, VLCD） 11: 保留</p>
4:2	DUTY[2:0]	<p>占空比选择</p> <p>这些位决定占空比。占空比定义为1/（LCD显示屏需要的COM数）</p> <p>000: 静态占空比 001: 1/2占空比</p>

010: 1/3占空比

011: 1/4占空比

100: 1/8占空比

101: 1/6占空比

110: 保留

111: 保留

1 VSRC LCD电压源

配置该位决定LCD电压源。

0: 内部电压源

1: 外部电压源(VLCD引脚)

0 LCDON LCD控制器开始

通过软件置1该位开始LCD控制器。通过软件清零该位停止LCD控制器，且LCD控制器在下一帧开始时停止。

0: LCD控制器停止

1: LCD控制器开始

### 23.4.2. LCD 配置寄存器 (LCD\_CFGR)

偏移地址: 0x04

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				PRE[3:0]				DIV[3:0]				BLKM[1:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKDIV[2:0]		CONR[2:0]		DD[2:0]		PULSE[2:0]		UPDIE		保留		SOFIE		HD	
rw		rw		rw		rw		rw		rw		rw		rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值
25:22	PRE[3:0]	LCD时钟预分频器 配置这些位定义LCD时钟预分频器。 0000: $f_{PRE} = f_{in\_clk}$ 0001: $f_{PRE} = f_{in\_clk}/2$ 0010: $f_{PRE} = f_{in\_clk}/4$ ... 1111: $f_{PRE} = f_{in\_clk}/32768$
21:18	DIV[3:0]	LCD时钟分频器 配置这些位定义DIV分频器的分频因子。 0000: $f_{LCD} = f_{PRE}/16$ 0001: $f_{LCD} = f_{PRE}/17$

		0010: $f_{LCD} = f_{PRE}/18$ ... 1111: $f_{LCD} = f_{PRE}/31$
17:16	BLKM[1:0]	闪烁模式 00: 不闪烁 01: 闪烁SEG[0]、COM[0]（1像素） 10: 闪烁SEG[0]和所有COM（由可编程占空比可实现最大8像素） 11: 闪烁所有SEG和所有COM（所有像素）
15:13	BLKDIV[2:0]	闪烁分频器 000: $f_{BLINK} = f_{LCD}/8$ 001: $f_{BLINK} = f_{LCD}/16$ 010: $f_{BLINK} = f_{LCD}/32$ 011: $f_{BLINK} = f_{LCD}/64$ 100: $f_{BLINK} = f_{LCD}/128$ 101: $f_{BLINK} = f_{LCD}/256$ 110: $f_{BLINK} = f_{LCD}/512$ 111: $f_{BLINK} = f_{LCD}/1024$
12:10	CONR[2:0]	对比度 当选择内部电压源（VSRC=0）时，这些位表示VLCD电压，其范围从VLCD0到VLCD7 (典型值为2.75V到5.20V)，VLCDx值请参考数据手册。当使用外部电压源时 (VSRC=1) 这些位无效。 000: VLCD0 001: VLCD1 010: VLCD2 011: VLCD3 100: VLCD4 101: VLCD5 110: VLCD6 111: VLCD7
9:7	DD[2:0]	死区时间 配置这些位定义帧间死区时间的长度。 000: 无死区时间 001: 1相周期死区时间 010: 2相周期死区时间 ... 111: 7相周期死区时间
6:4	PULSE[2:0]	脉冲持续时间 配置这些位根据PRE脉冲来定义脉冲持续时间。 000: 0 001: 1/f <sub>PRE</sub> 010: 2/f <sub>PRE</sub>

		011: 3/f <sub>PRE</sub>
		100: 4/f <sub>PRE</sub>
		101: 5/f <sub>PRE</sub>
		110: 6/f <sub>PRE</sub>
		111: 7/f <sub>PRE</sub>
3	UPDIE	LCD更新完成中断使能 该位可被软件置1和清零。 0: 禁用LCD更新完成中断 1: 使能LCD更新完成中断
2	保留	必须保持复位值
1	SOFIE	帧开始中断使能 该位可被软件置1和清零。 0: 禁用 LCD 帧开始中断 1: 使能LCD帧开始中断
0	HD	高驱动使能 该位可被软件置1和清零。 0: 禁用持久的高驱动。RL使能的持续时间通过PULSE[2:0]配置 1: 使能持久的高驱动。RL总是被使能，PULSE[2:0]位无效

### 23.4.3. LCD 状态标志寄存器 (LCD\_SFR)

偏移地址: 0x08

复位值: 0x0000 0020

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	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										SYNF	VRDYF	UPDF	UPRF	SOF	ONF
<b>位/位域</b>			<b>名称</b>			<b>描述</b>									
31:6	保留			必须保持复位值											
5	SYNF			LCD_CFG寄存器同步标志 当LCD_CFG寄存器更新到LCD时钟域时，该位置1。当写LCD_CFG寄存器时，通过硬件清零该位。 0: LCD_CFG寄存器尚未同步 1: LCD_CFG寄存器已与LCD时钟域同步										r	r
4	VRDYF			LCD电压就绪标志 该位根据LCD电压由硬件置位或清零。										r	r

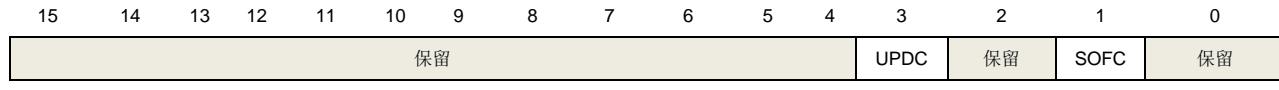
		0: LCD电压未就绪 1: 升压转换器已使能并准备提供准确电压
3	UPDF	更新LCD数据完成标志  当更新完成LCD数据时，该位通过硬件置位。通过向LCD_ICR寄存器的UPDC位写1来清零该位。  0: 无影响 1: LCD数据更新完成
2	UPRF	更新LCD数据请求标志  通过LCD_DATAx寄存器组修改第一缓冲区后，应用程序应当置位该位将数据传输到第二缓冲区中。该位将保持置位直到传输完成，在这段时间LCD_DATAx寄存器组为写保护状态。  0: 无影响 1: 请求LCD数据更新
1	SOF	帧开始标志  在一个新帧开始时，该位通过硬件置1。通过往LCD_ICR寄存器中SOFC位写1来清零该位。  0: 无影响 1: 帧开始标志
0	ONF	LCD控制器开启标志  当LCDON为1时，该位通过硬件置1。在LCDON位被清零并且最后的帧被显示后，该位通过硬件清零。  0: LCD控制器关闭 1: LCD控制器开启

#### 23.4.4. LCD 中断清除寄存器 (LCD\_ICR)

偏移地址: 0x0C

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
31:4	保留	必须保持复位值
3	UPDC	LCD数据更新完成清除位  置1该位清除LCD_SFR寄存器中的UPDF标志。  0: 无影响

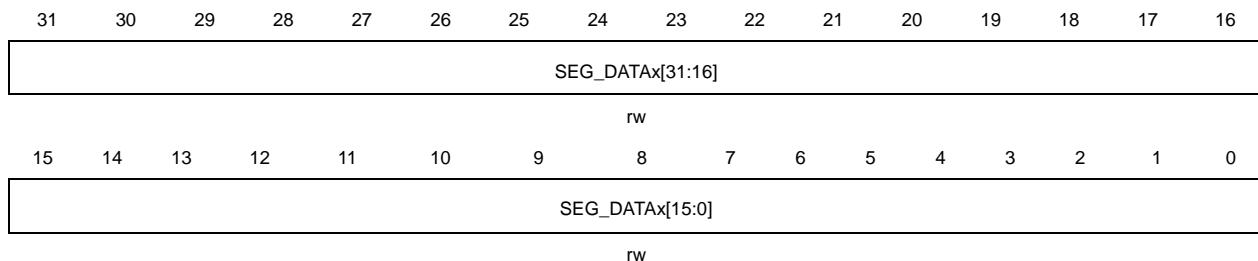
		1: 清除UPDF标志
2	保留	必须保持复位值
1	SOFC	帧开始标志清除 置1该位清除LCD_SFR寄存器中的SOF标志。 0: 无影响 1: 清除UPDF标志
0	保留	必须保持复位值

### 23.4.5. LCD 显示数据寄存器 (LCD\_DATAx, x=0~7)

偏移地址: 0x14+0x08\*x

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
31:0	SEG_DATAx[31:0]	每一位对应一个像素来显示。 0: 该像素无效 1: 该像素有效

## 24. 运算放大器 (OPAMP)/可编程参考电流和参考电压 (IVREF)

本章内容适用于 GD32F170xx 和 GD32F190xx 产品

### 24.1. 运算放大器 (OPAMP)

#### 24.1.1. 简介

具有外部和内部跟随路由功能的三路运放集成在 MCU 内（或对外部元件放大和滤波功能）。一个外部 ADC 通道测量一路运放的输出。

#### 24.1.2. 主要特性

- 轨到轨的输入输出电压范围
- 低输入偏置电流
- 低输入失调电压
- 低功耗模式

#### 24.1.3. 功能描述

##### 信号路由

与专用I/O口的连接情况如下所示：

- OPAMP1\_VINP——>PA1
- OPAMP1\_VINM——>PA2
- OPAMP1\_VOUT——>PA3
- OPAMP2\_VINP——>PA6
- OPAMP2\_VINM——>PA7
- OPAMP2\_VOUT——>PB0
- OPAMP3\_VINP——>PC1
- OPAMP3\_VINM——>PC2
- OPAMP3\_VOUT——>PC3

图24-1. OPAMP1信号路由

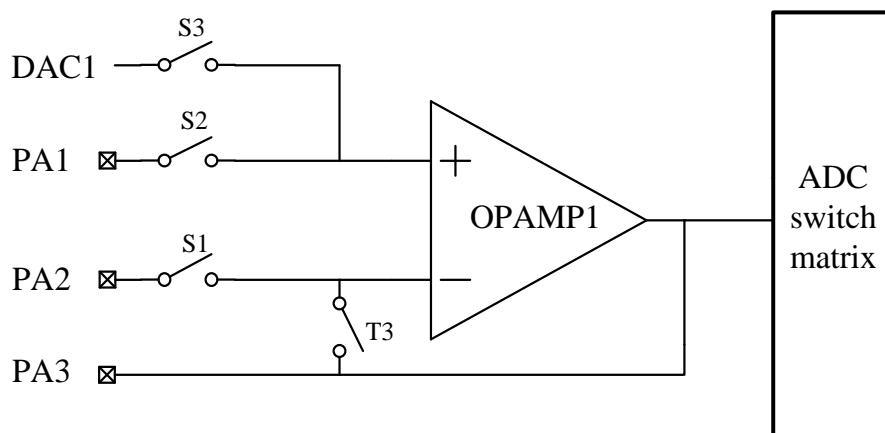


图24-2. OPAMP2信号路由

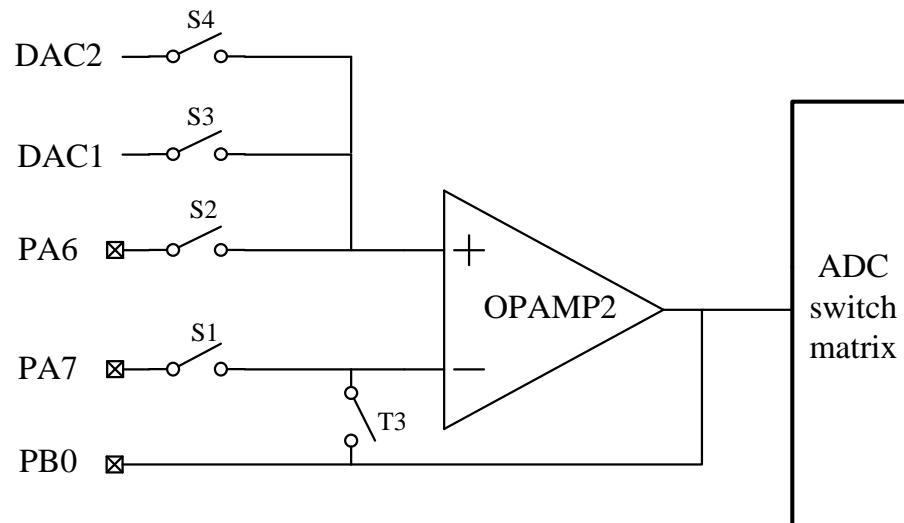
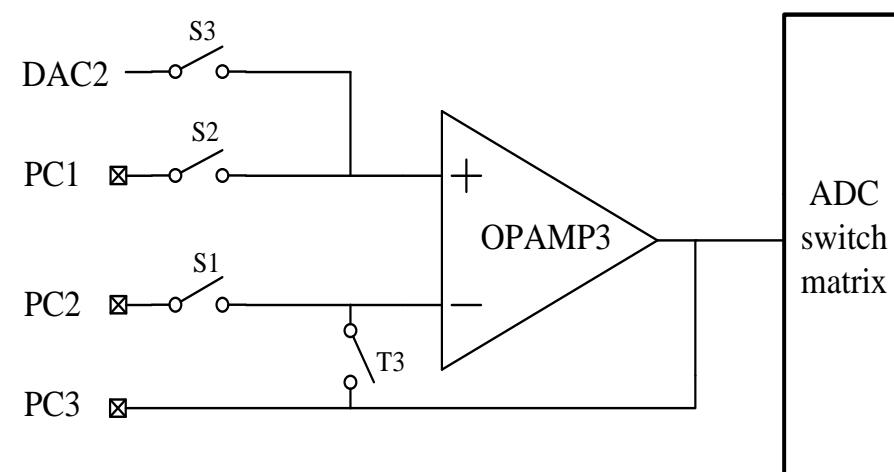


图24-3. OPAMP3信号路由



通过配置OPAMP\_CTRL寄存器可以选择三路运放的路由。在三路运放中，DAC1/DAC2通过S3/S4连接至同相输入端。OPAx\_PD位可以禁用所有运放。

## 校准

在出厂前，存储在非易失存储器中的默认出厂校准值可以用来初始化运放的失调，且芯片支持软件使用用户设定值来校准运放失调。

在校准操作时，所有与运放输入有关的开关必须断开。

每个运放都有正常模式和低功耗模式两种工作模式。不同的模式在不同寄存器里可以配置为不同的用户校准值。30位的校准值应用于每个运放的不同模式中，对OT\_USER位写1可以切换校准值为用户设定值。以上对所有运放都起作用。

以下步骤为对于一路运放的推荐流程：

- 1) 通过配置控制寄存器断开与运放连接的所有开关；
- 2) OT\_USER位置1；
- 3) 选择下表中的一种校准模式：

例如：运放1，正常模式，失调校准低  
配置如下：

OPA1PD=0, OPA1LPM=0, OPA1CAL\_H=0, OPA1CAL\_L=1

- 4) 在OPA1\_TRIM\_LOW中写入校准值。写入值应该从00000b开始至第一个值中，这会致OPA1CALOUT位置1。

注：在往校准寄存器写校准值后和检查 CALOUT 位前，软件应该等待在 datasheet 中指定的  $t_{offtrimmax}$  延时。

当获取到准确的校准值，这个值必须保持在校准寄存器里。

重复步骤3和4来完成整个运放校准流程：

- 正常模式校准高，
- 低功耗模式校准低，
- 低功耗模式校准高。

**注意：**在整个校准程序中，与运放输出外部连接的上拉或下拉电流不能大于500uA。

表24-1. 工作模式和校准

Mode	Control Bits				Output	
	OPAxPD	OPAxLPM	OPAxCAL_H	OPAxCAL_L	V <sub>OUT</sub>	CALOUT
<b>Normal</b>	0	0	0	0	analog	0
			1	1		
<b>Low-power</b>	0	1	0	0	analog	0
			1	1		
<b>Power-down</b>	1	X	X	X	Z	0
<b>Offset cal-high</b>	0	X	1	0	analog	X
<b>Offset cal-</b>	0	X	0	1	analog	X

low							
-----	--	--	--	--	--	--	--

#### 24.1.4. OPAMP 寄存器

##### OPAMP 控制寄存器 (OPAMP\_CTLR)

地址偏移: 0x5C

复位值: 0x0001 0101

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA3CAL OUT	OPA2CAL OUT	OPA1CAL OUT	OPA_RANG E	S4OPA 2	保留			OPA3 LPM	OPA3 CAL_H	OPA3CA L_L	S3O PA3	S2O PA3	S1O PA3	T3O PA3	OPA3 PD
r	r	r	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPA2LP M	OPA2CAL _H	OPA2CAL _L	S3OP A2	S2OP A2	S1OP A2	T3OP A2	OPA 2PD	OPA1 LPM	OPA1CA L_H	OPA1 CAL_L	S3O PA1	S2O PA1	S1O PA1	T3O PA1	OPA1 PD
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	OPA3CALOUT	OPAMP3校准输出 校准模式中, 如果失调被校准, 标志位会生效。
30	OPA2CALOUT	OPAMP2校准输出 校准模式中, 如果失调被校准, 标志位会生效。
29	OPA1CALOUT	OPAMP1校准输出 校准模式中, 如果失调被校准, 标志位会生效。
28	OPA_RANGE	当运放关闭时, 该位可被软件置位或清零。可选择运放稳定供电范围。 0: 低压范围(VDDA < 3.3 V) 1: 高压范围(VDDA > 3.3 V)
27	S4OPA2	OPAMP2中S4开关使能 0: S4开关打开 1: S4开关闭合
26:24	保留	必须保持复位值
23	OPA3LPM	OPAMP3低功耗模式 0: OPAMP3低功耗模式关闭 1: OPAMP3低功耗模式开启
22	OPA3CAL_H	OPAMP3中N差分失调校准 0: OPAMP3中N差分失调校准关闭 1: OPAMP3中N差分失调校准开启(OPA3CAL_L = 0)
21	OPA3CAL_L	OPAMP3中P差分失调校准

		0: OPAMP3中P差分失调校准关闭 1: OPAMP3中P差分失调校准开启(OPA3CAL_H = 0)
20	S3OPA3	OPAMP3中S3开关使能 0: S3开关打开 1: S3开关闭合
19	S2OPA3	OPAMP3中S2开关使能 0: S2开关打开 1: S2开关闭合
18	S1OPA3	OPAMP3中S1开关使能 0: S1开关打开 1: S1开关闭合
17	T3OPA3	OPAMP3中T3开关使能 0: T3开关打开 1: T3开关闭合
16	OPA3PD	OPAMP3关闭 0: OPAMP3开启 1: OPAMP3关闭
15	OPA2LPM	OPAMP2低功耗模式 0: OPAMP2低功耗模式关闭 1: OPAMP2低功耗模式开启
14	OPA2CAL_H	OPAMP2中N差分失调校准 0: OPAMP2中N差分失调校准关闭 1: OPAMP2中N差分失调校准开启(OPA2CAL_L = 0)
13	OPA2CAL_L	OPAMP2中P差分失调校准 0: OPAMP2中P差分失调校准关闭 1: OPAMP2中P差分失调校准开启(OPA2CAL_H = 0)
12	S3OPA2	OPAMP2中S3开关使能 0: S3开关打开 1: S3开关闭合
11	S2OPA2	OPAMP2中S2开关使能 0: S2开关打开 1: S2开关闭合
10	S1OPA2	OPAMP2中S1开关使能 0: S1开关打开 1: S1开关闭合
9	T3OPA2	OPAMP2中T3开关使能 0: T3开关打开

		1: T3开关闭合
8	OPA2PD	OPAMP2关闭 0: OPAMP2开启 1: OPAMP2关闭
7	OPA1LPM	OPAMP1低功耗模式 0: OPAMP1低功耗模式关闭 1: OPAMP1低功耗模式开启
6	OPA1CAL_H	OPAMP1中N差分失调校准 0: OPAMP1中N差分失调校准关闭 1: OPAMP1中N差分失调校准开启(OPA1CAL_L = 0)
5	OPA1CAL_L	OPAMP2中P差分失调校准 0: OPAMP1中P差分失调校准关闭 1: OPAMP1中P差分失调校准开启(OPA1CAL_H = 0)
4	S3OPA1	OPAMP1中S3开关使能 0: S3开关打开 1: S3开关闭合
3	S2OPA1	OPAMP1中S2开关使能 0: S2开关打开 1: S2开关闭合
2	S1OPA1	OPAMP1中S1开关使能 0: S1开关打开 1: S1开关闭合
1	T3OPA1	OPAMP1中T3开关使能 0: T3开关打开 1: T3开关闭合
0	OPA1PD	OPAMP1关闭 0: OPAMP1开启 1: OPAMP1关闭

### OPAMP 正常模式的失调校准寄存器 (OPAMP\_OTR)

地址偏移: 0x60

复位值: 0x0000 0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OT_USER	保留	OA3_TRIM_HIGH[4:0]				OA3_TRIM_LOW[4:0]				OA2_TRIM_HIGH[4:1]					
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OA2_TRIM_HIGH[0]	OA2_TRIM_LOW[4:0]	OA1_TRIM_HIGH[4:0]	OA1_TRIM_LOW[4:0]
------------------	-------------------	--------------------	-------------------

位/位域	名称	描述
31	OT_USER	读该位始终返回0，且可通过软件置位与清零。OPAMPx失调可通过该位选择出厂预设校准值或用户编程校准值来校准。 0：使用默认出厂值校准OPAMP失调 1：使用用户编程值校准OPAMP失调
30	保留	必须保持复位值
29:25	OA3_TRIM_HIGH[4:0] ]	OPAMP3, NMOS对正常模式下5-bit失调校准值
24:20	OA3_TRIM_LOW[4:0]	OPAMP3, PMOS对正常模式下5-bit失调校准值
19:15	OA2_TRIM_HIGH[4:0] ]	OPAMP2, NMOS对正常模式下5-bit失调校准值
14:10	OA2_TRIM_LOW[4:0]	OPAMP2, PMOS对正常模式下5-bit失调校准值
9:5	OA1_TRIM_HIGH[4:0] ]	OPAMP1, NMOS对正常模式下5-bit失调校准值
4:0	OA1_TRIM_LOW[4:0]	OPAMP1, PMOS对正常模式下5-bit失调校准值

### OPAMP 低功耗模式的失调校准寄存器 (OPAMP\_LPOTR)

地址偏移: 0x64

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		OA3_TRIM_LP_HIGH[4:0]				OA3_TRIM_LP_LOW[4:0]				OA2_TRIM_LP_HIGH[4:1]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2_LP_HIGH[0]		OA2_TRIM_LP_LOW[4:0]				OA1_TRIM_LP_HIGH[4:0]				OA1_TRIM_LP_LOW[4:0]					

位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	OA3_TRIM_LP_HIGH[4:0]	OPAMP3, NMOS对低功耗模式下5-bit失调校准值
24:20	OA3_TRIM_LP_LOW[4:0]	OPAMP3, PMOS对低功耗模式下5-bit失调校准值

---

19:15	OA2_TRIM_LP_HIGH[4:0]	OPAMP3, NMOS对低功耗模式下5-bit失调校准值
14:10	OA2_TRIM_LP_LOW[4:0]	OPAMP3, PMOS对低功耗模式下5-bit失调校准值
9:5	OA1_TRIM_LP_HIGH[4:0]	OPAMP3, NMOS对低功耗模式下5-bit失调校准值
4:0	OA1_TRIM_LP_LOW[4:0]	OPAMP3, PMOS对低功耗模式下5-bit失调校准值

## 24.2. 可编程参考电流和参考电压 (IVREF)

### 24.2.1. 简介

MCU可提供低功耗模式和大电流模式这两种可编程参考电流输出模式。在低功耗模式下，电流步长为1uA，最大电流为63uA。在大电流模式下，电流步长为8uA，最大电流为504uA。

MCU可提供一种参考电压，也可提供2.5V电压。

### 24.2.2. 主要特征

参考电流特征：

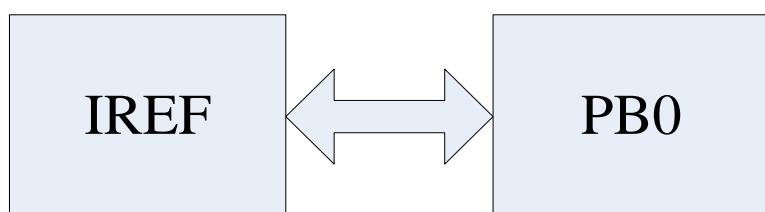
- 可编程电流
- 可编程源电流或灌电流
- 低功耗模式和大电流模式

参考电压特征：

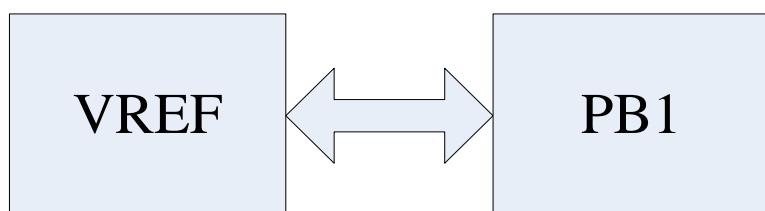
- 用户可编程校准

### 24.2.3. 功能描述

信号路由



在使用IREF时，PB0引脚应该被配置为模拟输入模式。



在使用VREF时，PB1引脚应该被配置为模拟I/O模式，且推荐使用外部VREF去耦电容。

用户校准

用户可以通过编程CPT[4:0]来校准IREF输出电流和通过编程VPT[4:0]来校准VREF电压。

#### 24.2.4. IVREF 寄存器

## IVREF 控制寄存器 (IVREF\_CTLR)

偏移地址: 0x300

复位值: 0x1000 0F00

该寄存器可以按半字（16位）或字（32位）访问

---

13	保留	必须保持复位值
12:8	CPT[4:0]	电流精度校准 00000: -15% .... 11111: +16%
7	SCMO	灌电流模式 0: 源电流 1: 灌电流
6	保留	必须保持复位值
5:0	CSDA[5:0]	电流步长数据 000000: 默认值 000001: Step * 1 .... 111111: Step * 63

## 25. CAN 总线控制器

本章内容适用于 GD32F170xx 和 GD32F190xx 产品

### 25.1. 简介

CAN (Controller Area Network) 总线是一种常见的工业控制总线，本章的 CAN 总线控制器遵循 CAN 总线协议 2.0A 和 2.0B。

CAN 总线控制器可以处理总线上的数据收发，通过过滤器后，应用程序就可以得到希望的数据帧。CAN 总线控制器具有 28 个过滤器。应用程序通过 3 个发送邮箱将发送数据送至总线，通过 2 个深度为 3 的接收 FIFO 获取总线数据。CAN 总线控制器同时也可以支持时间触发 CAN 通信（Time-trigger communication）。

### 25.2. 主要特性

- CAN 总线协议 2.0A 和 2.0B
- 通信波特率最大为 1Mbit/s
- 支持时间触发 CAN 通信（Time-trigger communication）

#### 发送功能

- 3 个发送邮箱
- 支持优先级发送
- 支持发送时间戳

#### 接收功能

- 2 个深度为 3 的接收 FIFO
- 具有 28 个过滤器
- FIFO 锁定功能

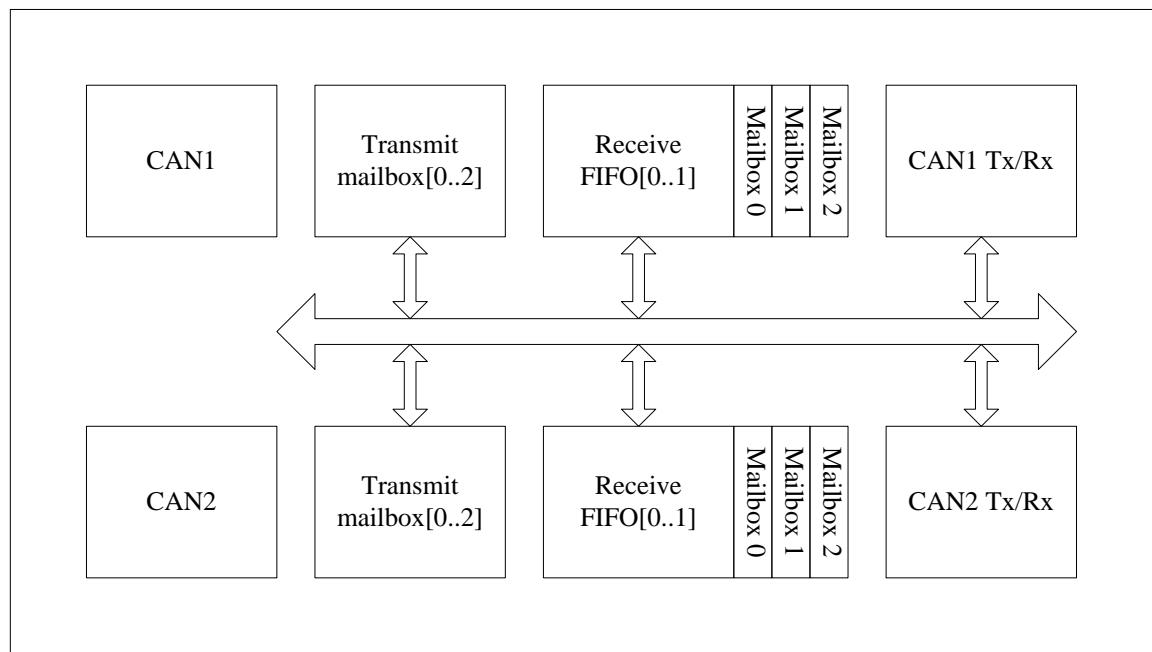
#### 时间触发通信

- 在时间触发通信模式下取消自动重传
- 16 位定时器
- 接收时间戳
- 发送时间戳

## 25.3. 功能描述

CAN 模块结构图如下所示。

图 25-1. CAN 模块结构图



### 25.3.1. 工作模式

CAN 总线控制器有 3 种工作模式：

- 睡眠工作模式
- 初始化工作模式
- 正常工作模式

#### 睡眠工作模式

芯片复位后，CAN总线控制器处于睡眠工作模式。同时睡眠工作模式是一种低功耗模式。

将CAN\_CTLR寄存器中的SWM置1，使CAN总线控制器进入睡眠工作模式，以CAN\_STR寄存器中的SWS置1来表示。

将CAN\_CTLR寄存器中的AWK置1可以使CAN总线控制器由硬件自动地离开睡眠工作模式。将CAN\_CTLR寄存器中的SWM清0，可以手动离开睡眠工作模式。

#### 初始化工作模式

初始化工作模式可以进行 CAN 总线通信参数调整。将 CAN\_CTLR 寄存器中的 IWM 置 1，使

CAN 总线控制器进入初始化工作模式，将其清 0 离开初始化工作模式。

### 正常工作模式

在初始化工作模式中设定 CAN 总线通信参数后，将 CAN\_CTLR 寄存器中的 IWM 清 0 进入正常工作模式，与 CAN 总线网络中的接点进行正常通信。

## 25.3.2. 通信模式

CAN 总线控制器有 4 种通信模式：

- 静默（Silent）通信模式
- 回环（Loopback）通信模式
- 回环静默（Loopback and Silent）通信模式
- 正常（Normal）通信模式

### 静默（Silent）通信模式

在静默通信模式下，可以从 CAN 总线接收数据，不向总线发送任何数据。将 CAN\_BTR 寄存器中的 SCM 置 1，使 CAN 总线控制器进入静默通信模式，将其清 0 可以离开静默通信模式。

静默通信模式可以用来监控 CAN 网络数据。

### 回环（Loopback）通信模式

在回环通信模式下，由 CAN 总线控制器发送的数据又可以被自己接收，同时这些发送数据也送至 CAN 网络。将 CAN\_BTR 寄存器中的 LCM 置 1，使 CAN 总线控制器进入回环通信模式，将其清 0 可以离开回环通信模式。

回环通信模式通常用来进行 CAN 通信自测。

### 回环静默（Loopback and Silent）通信模式

在回环静默通信模式下，CAN 总线控制器既不从 CAN 网络接收数据，也不向 CAN 网络发送数据，由其发送的数据又可以被自己接收。将 CAN\_BTR 寄存器中的 LCM 和 SCM 置 1，使 CAN 总线控制器进入回环静默通信模式，将它们清 0 可以离开回环静默通信模式。

回环通信模式通常用来进行 CAN 通信自测。

### 正常（Normal）通信模式

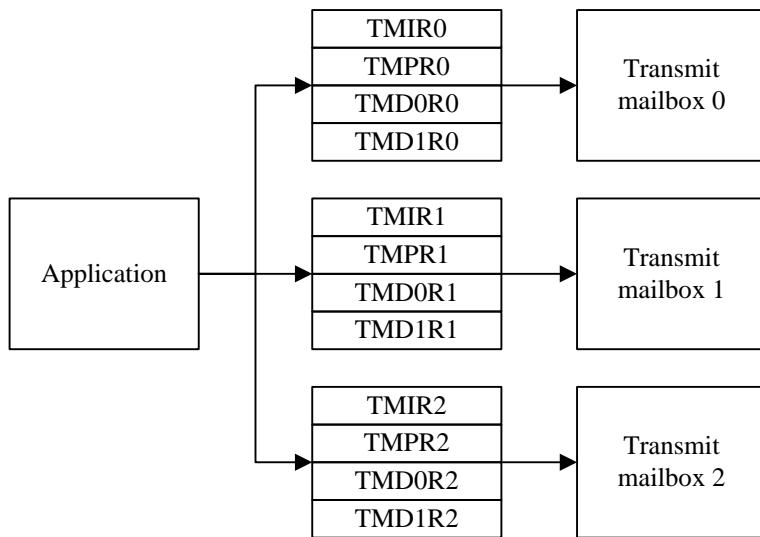
CAN 总线控制器通常工作在正常通信模式下，可以从 CAN 总线接收数据，也可以向 CAN 总线发送数据。这时需要将 CAN\_BTR 寄存器中的 LCM 和 SCM 清 0。

### 25.3.3. 数据发送

#### 发送寄存器

数据发送通过3个发送邮箱进行，可以通过寄存器CAN\_TMIRx, CAN\_TMPRx, CAN\_TMD0Rx和CAN\_TMD1Rx对发送邮箱进行配置。如下图所示。

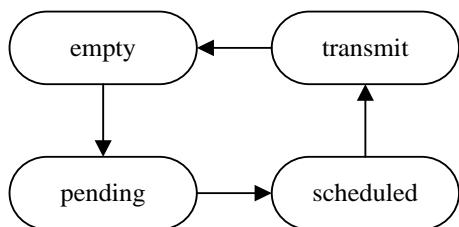
图 25-2. 发送寄存器



#### 发送邮箱状态转换

当发送邮箱处于**empty**状态时，应用程序才可以对邮箱进行配置。当邮箱被配置完成后，可以将CAN\_TMIRx寄存器TE置1，从而向CAN总线控制器提交发送请求，这时发送邮箱处于**pending**状态。当超过1个邮箱处于**pending**状态时，需要对多个邮箱进行调度，这时发送邮箱处于**scheduled**状态。当调度完成后，发送邮箱中的数据开始向CAN总线发送数据，这时发送邮箱处于**transmit**状态。当数据发送完成，邮箱变为空闲，可以再次交给应用程序使用，这时发送邮箱重新变为**empty**状态。如下图所示。

图 25-3. 发送邮箱状态转换



#### 发送状态和错误信息

CAN\_TSTR寄存器中的MTF, MTFNE, MAL和MTE用来说明发送状态和错误信息。

- **MTF**: 发送完成标志位。当数据发送完成时，MTF 置 1。

- **MTFNE**: 无错误发送完成标志位。当数据发送完成且没有错误时，MTFNE置1。
- **MAL**: 仲裁失败标志位。当发送数据过程中出现仲裁失败时，MAL置1。
- **MTE**: 发送错误标志位。当发送过程中检测到总线错误时，MTE置1。

## 数据发送步骤

数据发送步骤如下：

第1步：选择一个空闲发送邮箱。

第2步：根据应用程序要求，配置4个发送寄存器。

第3步：将CAN\_TMIRx寄存器的TE置1。

第4步：检测发送状态和错误信息。典型情况是检测到MTF和MTFNE置1，说明数据被成功发送。

## 发送选项

### 中止数据发送

将CAN\_TSTR寄存器的MST置1，可以中止数据发送。

当发送邮箱处于**pending**和**scheduled**状态，CAN\_TSTR寄存器的MST置1可以立即中止数据发送。

当发送邮箱处于**transmit**状态，则面临两种情况。一种情况是数据发送被成功地完成，MTF和MTFNE为1，这时发送邮箱将转换为**empty**状态。相对的，如果数据发送过程中出现了问题，这时发送邮箱将转换为**scheduled**状态，这时数据发送被中止。

### 发送优先级

当有2个及其以上发送邮箱等待发送时，寄存器CAN\_CTLR的TFO可以决定发送顺序。

当TFO为1，所有等待发送的邮箱按照先来先发送（FIFO）的顺序进行。

当TFO为0，具有最小标识符（Identifier）的邮箱最先发送。如果所有的标识符（Identifier）相等，具有最小邮箱编号的邮箱最先发送。

## 25.3.4. 数据接收

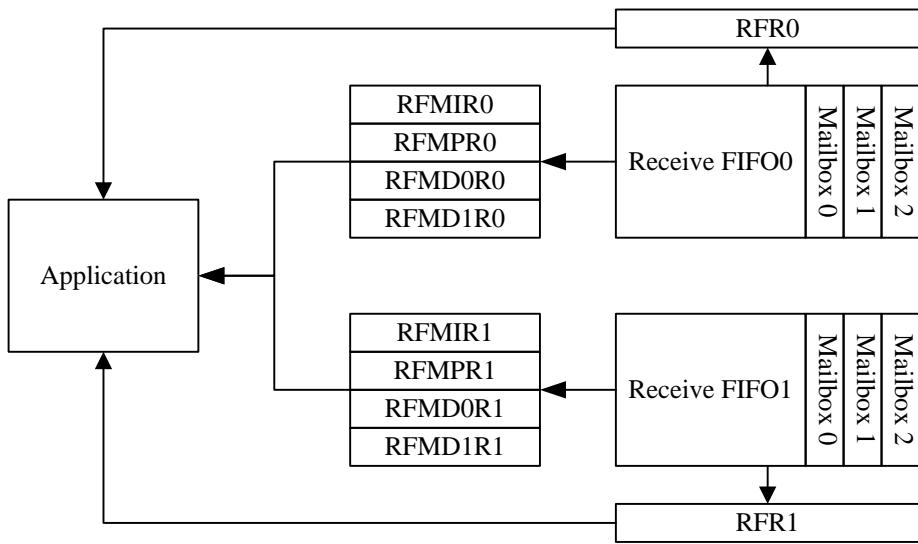
### 接收寄存器

应用程序通过2个深度为3的FIFO接收来自CAN网络的数据。

寄存器CAN\_RFRx可以操作FIFO，也包含FIFO状态。寄存器CAN\_RFMRx，CAN\_RFMPRx，CAN\_RFMD0Rx和CAN\_RFMD1Rx可以表示接收数据帧。

如下图所示。

图 25-4. 接收寄存器



### 接收 FIFO

每个接收FIFO包含3个接收邮箱，用来存储接收数据帧。这些邮箱按照先进先出方式进行组织，最早从CAN网络接收的数据，最早被应用程序处理。

寄存器CAN\_RFRx包含FIFO状态信息和帧的数量。当FIFO中包含数据时，可以通过寄存器CAN\_RFMI Rx, CAN\_RFMPRx, CAN\_RFMD0Rx和CAN\_RFMD1Rx读取数据，之后将寄存器CAN\_RFRx的RFD置1释放邮箱。

### 接收 FIFO 状态信息

接收FIFO状态信息包含在寄存器CAN\_RFRx中。

RFL: FIFO中包含的帧数量。FIFO为空时，RFL为0；FIFO为满时，RFL为3。

RFF: FIFO满状态标志位。这时RFL为3。

RFO: FIFO溢出标志位。当FIFO已经包含了3个数据帧时，新的数据帧到来使FIFO发生溢出。

### 数据接收步骤

第1步：查看FIFO中帧的数量。

第2步：通过CAN\_RFMI Rx, CAN\_RFMPRx, CAN\_RFMD0Rx和CAN\_RFMD1Rx读取数据

第3步：将寄存器CAN\_RFRx的RFD置1释放邮箱，并且等待其由硬件自动清0。

### 25.3.5. 过滤功能

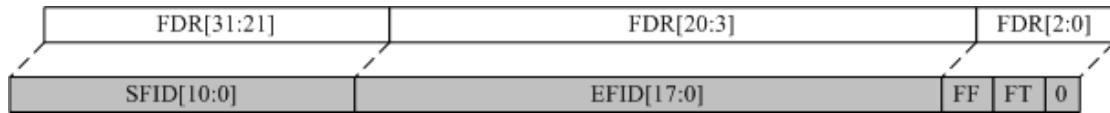
一个待接收的数据帧会根据其标识符（Identifier）进行过滤：通过过滤的帧，送至FIFO；没有通过过滤的帧，直接丢弃。

## 过滤器位宽

每一个过滤器单元有2个寄存器CAN\_FDR0和CAN\_FDR1，它们可以配置为2种位宽：32-bit位宽和16-bit位宽。

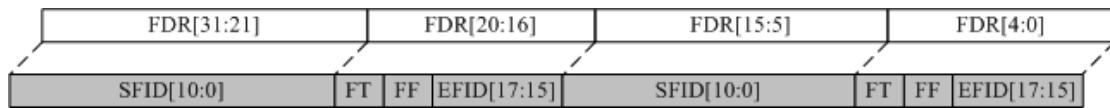
32-bit位宽CAN\_FDRx包含字段：SFID[10:0], EFID[17:0], FF和FT。如下图所示。

图 25-5. 32-bit 位宽过滤器



16-bit位宽CAN\_FDRx包含字段：SFID[10:0], FT, FF和EFID[17:15]。如下图所示。

图 25-6. 16-bit 位宽过滤器

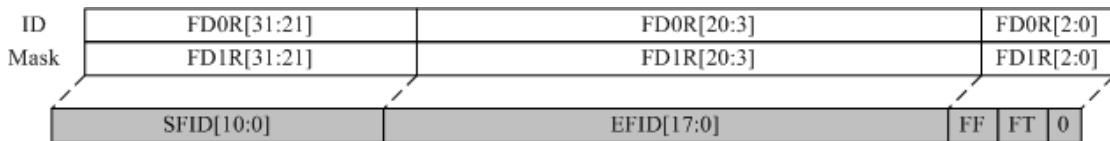


## 掩码模式

对于一个接收数据帧的标识符（Identifier），掩码模式用来指定哪些位必须与预设的标识符相同，哪些位无需判断。

一个32-bit位宽掩码模式过滤器如下图所示。

图 25-7. 32-bit 位宽掩码模式过滤器



## 列表模式

对于一个接收数据帧的标识符（Identifier），列表模式用来表示与预设的标识符列表中能够匹配则通过，否则丢弃。

一个32-bit位宽列表模式过滤器如下图所示。

图 25-8. 32-bit 位宽列表模式过滤器



## 过滤序号

过滤器由若干过滤单元（Bank）组成，每个过滤单元因为位宽和模式的选择不同，而具有不同的过滤效果。例如下图所示的2个过滤单元，Bank0是32-bit位宽掩码模式，Bank1是32-bit位宽列表模式。

图 25-9. 过滤序号

Filter Bank	Filter Data Register	Filter Number
0	F0D0R-32bit-ID	0
	F0D1R-32bit-Mask	
1	F1D0R-32bit-ID	1
	F1D1R-32bit-ID	

Bank0中的寄存器F0D0R和F0D1R共同组成一个可以被过滤器通过的标识符集合，它的过滤序号（Filter Number）是0。

Bank1中的寄存器F1D0R包含了一个可以被过滤器通过的标识符集合，而F1D1R也包含了一个可以被过滤器通过的标识符集合，它们的过滤序号分别是1和2。

## 过滤器关联的 FIFO

过滤单元可以关联FIFO0或FIFO1。一个关联到FIFO0的过滤单元，通过这个过滤单元的帧被投送到FIFO0中存储。

## 过滤器激活控制

一个过滤单元是否工作，是否可以起到过滤作用，可以通过CAN\_FWR寄存器进行控制。例如将寄存器CAN\_FWR中的FW0置1，Bank0被设置为工作状态。

## 过滤索引

一个包含过滤序号（Filter Number）N的过滤单元通过了某个帧，则该帧的过滤索引（Filtering Index）为N。这时CAN\_RFMPRx寄存器中FI等于N。下图是一个过滤索引的例子。

在下图中，如果一个帧通过了FIFO0中过滤序号10（Filter Number=10）的过滤单元，那么该帧的过滤索引为10。这时CAN\_RFMPRx寄存器中FI等于10。

过滤序号不关心对应的过滤单元（Bank）是否处于工作状态。例如Bank3被关联到FIFO0，且为“不激活”状态，但它仍然包含过滤序号3和4。

图 25-10. 过滤索引

<i>Filter Bank</i>	<i>FIFO0</i>	<i>Active</i>	<i>Filter Number</i>	<i>Filter Bank</i>	<i>FIFO1</i>	<i>Active</i>	<i>Filter Number</i>
0	F0D0R-32bit-ID	Yes	0	2	F2D0R[15:0]-16bit-ID	Yes	0
	F0D1R-32bit-Mask				F2D0R[31:16]-16bit-Mask		1
1	F1D0R-32bit-ID	Yes	1		F2D1R[15:0]-16bit-ID	No	2
	F1D1R-32bit-ID		2		F2D1R[31:16]-16bit-Mask		3
3	F3D0R[15:0]-16bit-ID	No	3	4	F4D0R-32bit-ID	No	4
	F3D0R[31:16]-16bit-Mask				F4D1R-32bit-Mask		5
	F3D1R[15:0]-16bit-ID		4	5	F5D0R-32bit-ID	No	6
	F3D1R[31:16]-16bit-Mask				F5D1R-32bit-ID		7
7	F7D0R[15:0]-16bit-ID	No	5	6	F6D0R[15:0]-16bit-ID	Yes	8
	F7D0R[31:16]-16bit-ID		6		F6D0R[31:16]-16bit-ID		9
	F7D1R[15:0]-16bit-ID		7		F6D1R[15:0]-16bit-ID		10
	F7D1R[31:16]-16bit-ID		8		F6D1R[31:16]-16bit-ID		11
8	F8D0R[15:0]-16bit-ID	Yes	9	10	F10D0R[15:0]-16bit-ID	No	12
	F8D0R[31:16]-16bit-ID		10		F10D0R[31:16]-16bit-Mask		13
	F8D1R[15:0]-16bit-ID		11		F10D1R[15:0]-16bit-ID		14
	F8D1R[31:16]-16bit-ID		12		F10D1R[31:16]-16bit-Mask		15
9	F9D0R[15:0]-16bit-ID	Yes	13	11	F11D0R[15:0]-16bit-ID	No	16
	F9D0R[31:16]-16bit-Mask				F11D0R[31:16]-16bit-ID		11
	F9D1R[15:0]-16bit-ID		14		F11D1R[15:0]-16bit-ID		12
	F9D1R[31:16]-16bit-Mask				F11D1R[31:16]-16bit-ID		13
12	F12D0R-32bit-ID	Yes	15	13	F13D0R-32bit-ID	Yes	14
	F12D1R-32bit-Mask				F13D1R-32bit-ID		15

### 优先级

过滤器优先级顺序如下：

- 1、32-bit位宽模式高于16-bit位宽模式
- 2、列表模式高于掩码模式
- 3、较小的过滤序号（Filter Number）具有较高的优先级

### 25.3.6. 时间触发通信

时间触发通信是CAN通信应用协议，CAN网络中的所有节点都按照一个预先设定的时间序列进行通信，尤其适合于时间周期性应用和时间确定性应用。

在这种通信模式下，一个内部的16-bit计数器开始工作，在每一个CAN位时间（Bit time）增1。这个内部计数器为数据发送和数据接收提供时间戳，这些时间戳存放在寄存器CAN\_RFMPRx 和CAN\_TMPRx中。

在这种通信模式下，自动重发功能是禁止的。

### 25.3.7. 通信参数

#### 自动重发禁止模式

在时间触发通信模式下，要求自动重发必须是禁止的，可以通过将CAN\_CTLR寄存器的ARD置1满足要求。这时数据只会被发送一次，如果因为仲裁失败或者总线错误而导致发送失败，CAN总线控制器不会像通常那样进行数据自动重发。

发送结束时，寄存器CAN\_TSTR中的MTF置1，而发送状态信息可以通过MTFNE，MAL和MTE获得。

#### 位时序 (Bit timing)

CAN总线控制器将位时间分为3个部分。

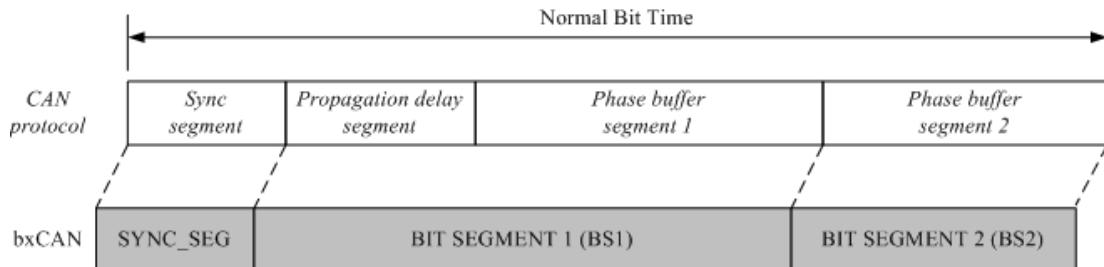
同步段 (Synchronization segment)，记为SYNC\_SEG。该段占用1个时间单元 ( $1 \times t_q$ )。

位段1 (Bit segment 1)，记为BS1。该段占用1到16个时间单元。相对于CAN协议而言，BS1相当于传播时间段 (Propagation delay segment) 和相位缓冲段1 (Phase buffer segment 1)。

位段2 (Bit segment 2)，记为BS2。该段占用1到8个时间单元。相对于CAN协议而言，BS2相当于传播时间段 (Propagation delay segment) 和相位缓冲段1 (Phase buffer segment 1)。

对比与CAN协议，位时序如下图所示。

图 25-11. 位时序



再同步补偿宽度SJW (reSynchronization Jump Width) 对CAN网络节点同步误差进行补偿，占用1到4个时间单元。

#### 波特率

波特率计算公式如下：

$$BaudRate = \frac{1}{Normal\ Bit\ Time}$$

$$Normal\ Bit\ Time = t_{SYNC\_SEG} + t_{BS1} + t_{BS2}$$

其中：

$$t_{SYNC\_SEG} = 1 \times t_q$$

$$t_{BS1} = (1 + BTR.TS1) \times t_q$$

$$t_{BS2} = (1 + BTR.TS2) \times t_q$$

$$t_q = (1 + BTR.BRP) \times t_{PCLK1}$$

### 25.3.8. 错误标志

CAN总线的状态可以通过发送错误计数值（Transmit Error Counter，记为TEC）和接收错误计数值（Receive Error Counter，记为REC）反映。同时寄存器CAN\_ER还可以表明当前错误状态，这些错误状态在寄存器CAN\_IER控制下产生中断。

#### 离线恢复

当TEC大于255时，CAN总线控制器进入离线状态，这时寄存器CAN\_ER中的BOE置1，并且发送和接收失效。

根据寄存器CAN\_CTLR中的ABOR配置，离线恢复有2种方式。如果ABOR为1，处于离线状态的CAN总线控制器检测到CAN协议所定义的离线恢复序列（在CAN\_RX检测到128次连续11个位的隐性位）时，会自动恢复。如果ABOR为0，则将CAN\_CTLR中的IWM置1进入初始化模式，然后进入正常工作模式。

### 25.3.9. 中断

CAN总线控制器占用4个中断向量，通过寄存器CAN\_IER进行控制。这4个中断向量对应4类中断源：

- 发送中断
- FIFO0 中断
- FIFO1 中断
- 错误和状态改变中断

#### 发送中断

发送中断包括：

- 寄存器CAN\_TSTR中的MTF0置1：发送邮箱0变为空闲。
- 寄存器CAN\_TSTR中的MTF1置1：发送邮箱1变为空闲。
- 寄存器CAN\_TSTR中的MTF2置1：发送邮箱2变为空闲。

### FIFO0 中断

FIFO0中断包括:

- 寄存器CAN\_RFR0中的RFL0不为0: FIFO0中包含待接收数据。
- 寄存器CAN\_RFR0中的RFF0为1: FIFO0满。
- 寄存器CAN\_RFR0中的RFO0为1: FIFO0溢出。

### FIFO1 中断

FIFO1中断包括:

- 寄存器CAN\_RFR1中的RFL1不为0: FIFO1中包含待接收数据。
- 寄存器CAN\_RFR1中的RFF1为1: FIFO1满。
- 寄存器CAN\_RFR1中的RFO1为1: FIFO1溢出。

### 错误和状态改变中断

错误和状态改变中断包括:

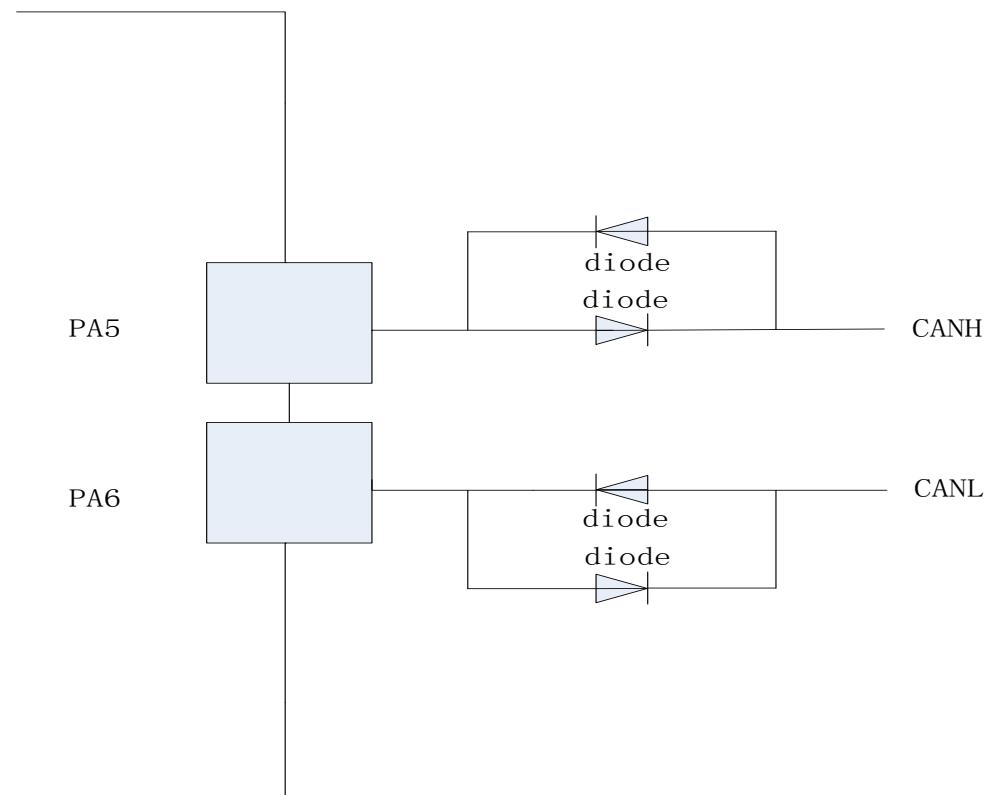
- 寄存器CAN\_ER 中的EN在1至6之间, BOE、PE和WE为1: 发生错误。
- 寄存器CAN\_STR中的WIF为1: 处于睡眠工作模式的CAN总线控制器检测到唤醒信号。
- 寄存器CAN\_STR中的SEIF为1: CAN总线控制器进入睡眠工作模式。

## 25.3.10. bxCAN PHY 模式

若将CAN\_PHYCR寄存器的PHYEN位置1, CAN PHY功能将使能。内部的收发器将进入工作状态, 此时硬件上可以不再外接收发器。该功能仅bxCAN1模块可以使用。

MCU芯片外部连接图如下所示:

图 25-12. bxCAN PHY 连接图



1. 将PA5/PA6配置成模拟输入模式。
2. 使能CAN时钟。
3. 在CAN\_PHYCR寄存器中，配置PHYEN和PDMODE位。
4. 配置CAN相关寄存器。

## 25.4. CAN 寄存器

### 25.4.1. CAN 控制寄存器 (CAN\_CTLR)

地址偏移: 0x00

复位值: 0x0001 0002

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														DFZ	
														RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SR	保留							TTC	ABOR	AWK	ARD	RFOD	TFO	SWM	IWM
RS								RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:17	保留	必须保持复位值
16	DFZ	调试冻结 0: 处于Debug时, CAN接收和发送正常工作 1: 处于Debug时, CAN接收和发送停止
15	SR	软件复位 0: 正常操作 1: 复位CAN并进入睡眠工作模式 该位会自动清0。
14:8	保留	必须保持复位值
7	TTC	时间触发通信 0: 禁用时间触发通信 1: 使能时间触发通信
6	ABOR	自动离线恢复 0: 通过软件手动地从离线状态恢复 1: 通过硬件自动的从离线状态恢复
5	AWK	自动唤醒 0: 通过软件手动的从睡眠工作模式唤醒 1: 通过硬件自动的从睡眠工作模式唤醒
4	ARD	自动重发禁止 0: 使能自动重发 1: 禁用自动重发
3	RFOD	禁用接收FIFO满时覆盖 0: 使能接收FIFO满时覆盖。当接收FIFO满时, FIFO中的数据被新来的数据覆盖。 1: 禁用接收FIFO满时覆盖。当接收FIFO满时, 新来的数据被丢弃, FIFO中的数据保持不变。
2	TFO	发送FIFO顺序 0: 标识符 (Identifier) 较小的帧先发送 1: 所有等待发送的邮箱按照先进先出 (FIFO) 的顺序发送
1	SWM	睡眠工作模式 0: 禁用睡眠工作模式 1: 使能睡眠工作模式
0	IWM	初始化工作模式 0: 禁用初始化工作模式 1: 使能初始化工作模式

### 25.4.2. CAN 状态寄存器 (CAN\_STR)

地址偏移: 0x04

复位值: 0x0000 0C02

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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	LASTRX	RS	TS	保留		SEIF	WIF	EIF	SWS	IWS	rc_w1	rc_w1	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值
11	RX	RX引脚电平
10	LASTRX	RX引脚最近一次的采样值
9	RS	接收状态 0: CAN当前不是接收器 1: CAN当前是接收器
8	TS	发送状态 0: CAN当前不是发送器 1: CAN当前是发送器
7:5	保留	必须保持复位值
4	SEIF	进入睡眠工作模式的状态改变中断标志 0: CAN没有进入睡眠工作模式 1: CAN进入睡眠工作模式。如果相应的中断使能位为1，则发生中断。
3	WIF	从睡眠工作模式唤醒的状态改变中断标志 0: 没有检测到唤醒信号 1: 发现唤醒信号。如果相应的中断使能位为1，则发生中断。
2	EIF	错误中断标志 0: 没有错误。 1: 发生错误。如果相应的中断使能位为1，则发生中断。
1	SWS	睡眠工作状态 0: CAN没有处于睡眠工作状态 1: CAN处于睡眠工作状态
0	IWS	初始化工作状态 0: CAN没有处于初始化工作状态

1: CAN处于初始化工作状态

### 25.4.3. CAN 发送状态寄存器 (CAN\_TSTR)

地址偏移: 0x08

复位值: 0x1C00 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]	MST2	保留	MTE2	MAL2	MTFNE2	MTF2			
r	r	r	r	r	r	r	rs		rc_w1	rc_w1	rc_w1	rc_w1			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	保留		MTE1	MAL1	MTFNE1	MTF1	MST0	保留	MTE0	MAL0	MTFNE0	MTF0			
rs			rc_w1	rc_w1	rc_w1	rc_w1	rs		rc_w1	rc_w1	rc_w1	rc_w1			

位/位域	名称	描述
31	TMLS2	在发送FIFO中邮箱2最后发送 该位为1表明, 当有2个及其以上帧等待发送时, 发送邮箱2具有最后的发送顺序。
30	TMLS1	在发送FIFO中邮箱1最后发送 该位为1表明, 当有2个及其以上帧等待发送时, 发送邮箱1具有最后的发送顺序。
29	TMLS0	在发送FIFO中邮箱0最后发送 该位为1表明, 当有2个及其以上帧等待发送时, 发送邮箱0具有最后的发送顺序。
28	TME2	发送邮箱2空闲 0: 发送邮箱2不空闲 1: 发送邮箱2空闲
27	TME1	发送邮箱1空闲 0: 发送邮箱1不空闲 1: 发送邮箱1空闲
26	TME0	发送邮箱0空闲 0: 发送邮箱0不空闲 1: 发送邮箱0空闲
25:24	NUM[1:0]	当发送FIFO不满时, NUM表示下一个将要发送的邮箱号。 当发送FIFO满时, NUM表示最后一个将要发送的邮箱号。
23	MST2	邮箱2停止发送 将其置1, 将停止邮箱2的发送过程。 当邮箱2变为empty状态时, 该位被硬件自动清0。
22:20	保留	必须保持复位值
19	MTE2	邮箱2发送错误

		当发生错误时该位被置1
18	MAL2	邮箱2仲裁失败 当发生仲裁失败时该位被置1
17	MTFNE2	邮箱2无错发送完成 当发送完成并且没有错误时该位被置1
16	MTF2	邮箱2发送完成 当发送完成时该位被置1。 对该位写1可以将该位清0，也可以通过将寄存器CAN_TMR2中的TE置1而将MTF2清0。另外，当MTF2被清0时，MTE2，MAL2和MTFNE2也会被同时清0。
15	MST1	邮箱1停止发送 将其置1，将停止邮箱1的发送过程。 当邮箱1变为empty状态时，该位被硬件自动清0。
14:12	保留	必须保持复位值
11	MTE1	邮箱1发送错误 当发生错误时该位被置1
10	MAL1	邮箱1仲裁失败 当发生仲裁失败时该位被置1
9	MTFNE1	邮箱1无错发送完成 当发送完成并且没有错误时该位被置1
8	MTF1	邮箱1发送完成 当发送完成时该位被置1。 对该位写1可以将该位清0，也可以通过将寄存器CAN_TMR1中的TE置1而将MTF1清0。另外，当MTF1被清0时，MTE1，MAL1和MTFNE1也会被同时清0。
7	MST0	邮箱0停止发送 将其置1，将停止邮箱0的发送过程。 当邮箱0变为empty状态时，该位被硬件自动清0。
6:4	保留	必须保持复位值
3	MTE0	邮箱0发送错误 当发生错误时该位被置1。
2	MAL0	邮箱0仲裁失败 当发生仲裁失败时该位被置1。
1	MTFNE0	邮箱0无错发送完成 当发送完成并且没有错误时该位被置1。
0	MTF0	邮箱0发送完成 当发送完成时该位被置1。 对该位写1可以将该位清0，也可以通过将寄存器CAN_TMR0中的TE置1而将MTF0清0。

清0。另外，当MTFO被清0时，MTE0，MALO和MTFNE0也会被同时清0。

#### 25.4.4. CAN 接收 FIFO0 寄存器 (CAN\_RFR0)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								RFDO	RFO0	RFF0	保留	RFL0[1:0]	rs      rc_w1      rc_w1		

位/位域	名称	描述
31:6	保留	必须保持复位值
5	RFDO	从接收FIFO0读取数据 该位被置1，将从FIFO0读取数据。 数据读取完毕并释放相应的数据空间后，该位被清0。
4	RFO0	接收FIFO0溢出 0: 接收FIFO0没有溢出 1: 接收FIFO0溢出
3	RFF0	接收FIFO0满 0: 接收FIFO0不满 1: 接收FIFO0满
2	保留	必须保持复位值
1:0	RFL0[1:0]	接收FIFO0中帧的数量

#### 25.4.5. CAN 接收 FIFO1 寄存器 (CAN\_RFR1)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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								RFDO1	RFO1	RFF1	保留	RFL1[1:0]			

位/位域	名称	描述
31:6	保留	必须保持复位值
5	RFD1	从接收FIFO1读取数据 该位被置1，将从FIFO1从读取数据。 数据读取完毕并释放相应的数据空间后，该位被清0。
4	RFO1	接收FIFO1溢出 0：接收FIFO1没有溢出 1：接收FIFO1溢出
3	RFF1	接收FIFO1满 0：接收FIFO1不满 1：接收FIFO1满
2	保留	必须保持复位值
1:0	RFL1[1:0]	接收FIFO1中帧的数量

#### 25.4.6. CAN 中断使能寄存器 (CAN\_IER)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													SWIE	WIE	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIE	保留	ENIE	BOIE	PEIE	WEIE	保留	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE		
rw		rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值
17	SWIE	睡眠中断使能 0：禁用睡眠中断 1：使能睡眠中断
16	WIE	唤醒中断使能 0：禁用唤醒中断 1：使能唤醒中断
15	EIE	错误中断使能 0：禁用错误中断

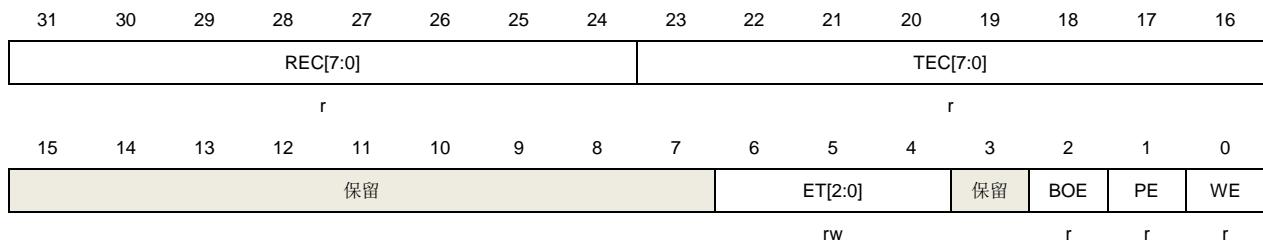
		1: 使能错误中断
14:12	保留	必须保持复位值
11	ENIE	错误种类中断使能 0: 禁用错误种类中断 1: 使能错误种类中断
10	BOIE	离线中断使能 0: 禁用离线中断 1: 使能离线中断
9	PEIE	被动错误中断使能 0: 禁用被动错误 1: 使能被动错误
8	WEIE	警告错误中断使能 0: 禁用警告错误中断 1: 使能警告错误中断
7	保留	必须保持复位值
6	RFOIE1	接收FIFO1溢出中断使能 0: 禁用接收FIFO1溢出中断 1: 使能接收FIFO1溢出中断
5	RFFIE1	接收FIFO1满中断使能 0: 禁用接收FIFO1满中断 1: 使能接收FIFO1满中断
4	RFNEIE1	接收FIFO1非空中断使能 0: 禁用接收FIFO1非空中断 1: 使能接收FIFO1非空中断
3	RFOIE0	接收FIFO0溢出中断使能 0: 禁用接收FIFO0溢出中断 1: 使能接收FIFO0溢出中断
2	RFFIE0	接收FIFO0满中断使能 0: 禁用接收FIFO0满中断 1: 使能接收FIFO0满中断
1	RFNEIE0	接收FIFO0非空中断使能 0: 禁用接收FIFO0非空中断 1: 使能接收FIFO0非空中断
0	TMEIE	发送邮箱空中断使能 0: 禁用发送邮箱空中断 1: 使能发送邮箱空中断

### 25.4.7. CAN 错误寄存器 (CAN\_ER)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:24	REC[7:0]	接收错误计数值
23:16	TEC[7:0]	发送错误计数值
15:7	保留	必须保持复位值
6:4	ET[2:0]	错误种类 ET由硬件更新，可以反映位传输过程中的错误情况。当位传输成功没有错误时，ET为0。软件可以设置ET为0b111。 000: 无错误 001: 填充错误 010: 格式错误 011: ACK错误 100: 位隐性错 101: 位显性错误 110: CRC错误 111: 软件设置值
3	保留	必须保持复位值
2	BOE	离线错误 当CAN总线控制器离线时，该位被置1。
1	PE	被动错误 当TEC或者REC大于127时，该位由硬件置1。
0	WE	警告错误 当TEC或REC大于等于96时，该位由硬件置1。

### 25.4.8. CAN 位时序寄存器 (CAN\_BTR)

地址偏移: 0x1C

复位值: 0x0123 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCM	LCM	保留			SJW[1:0]	保留	BS2[2:0]	BS1[3:0]							
rw	rw				rw				rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			BRP[9:0]												rw

位/位域	名称	描述
31	SCM	静默通信模式 0: 禁用静默通信模式 1: 使能静默通信模式
30	LCM	回环通信模式 0: 禁用回环通信模式 1: 使能回环通信模式
29:26	保留	必须保持复位值
25:24	SJW[1:0]	再同步补偿宽度 再同步补偿占用的时间单元数量 = SJW[1:0]+1
23	保留	必须保持复位值
22:20	BS2[2:0]	位段2 位段2占用的时间单元数量 = BS2[2:0]+1
19:16	BS1[3:0]	位段1 位段1占用的时间单元数量 = BS1[3:0]+1
15:10	保留	必须保持复位值
9:0	BRP[9:0]	波特率分频系数

#### 25.4.9. CAN 发送邮箱标识符寄存器 (CAN\_TMRx) (x=0..2)

地址偏移: 0x180, 0x190, 0x1A0

复位值: 0xXXXX XXXX (bit0=0)

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFID[10:0]/EFID[28:18]								EFID[17:13]							
rw										rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFID[12:0]								FF FT TE				rw	rw	rw	
rw															

位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	TE	发送使能 0: 禁用发送 1: 使能发送 当应用程序想要发送数据时，该位被置1将启动发送过程。当发送结束，发送邮箱为空时，该位由硬件清0。

#### 25.4.10. CAN 发送邮箱属性寄存器 (CAN\_TMPRx) (x=0..2)

地址偏移: 0x184, 0x194, 0x1A4

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:16	TS[15:0]	时间戳 发送时间戳
15:9	保留	必须保持复位值
8	TSE	时间戳使能 0: 禁用时间戳 1: 使能时间戳。时间戳TS[15:0]将放在寄存器CAN_TMD1R的DB6和DB7中。

只有当寄存器CAN\_CTLR中的TTC为1时，该位才有效。

7:4	保留	必须保持复位值
3:0	DLC[3:0]	数据长度 DLC[3:0]表示帧内数据长度。

#### 25.4.11. CAN 发送邮箱 Data0 寄存器 (CAN\_TMD0Rx) (x=0..2)

地址偏移: 0x188, 0x198, 0x1A8

复位值: 0xXXXX XXXX

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB3[7:0]								DB2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							

位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

#### 25.4.12. CAN 发送邮箱 Data1 寄存器 (CAN\_TMD1Rx) (x=0..2)

地址偏移: 0x18C, 0x19C, 0x1AC

复位值: 0xXXXX XXXX

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB7[7:0]								DB6[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							

位/位域	名称	描述
31:24	DB7[7:0]	字节7

---

23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

#### 25.4.13. CAN 接收 FIFO 邮箱标识符寄存器 (CAN\_RFMIRx) (x=0..1)

地址偏移: 0x1B0, 0x1C0

复位值: 0xFFFF XXXX

该寄存器只能按字(32 位)访问



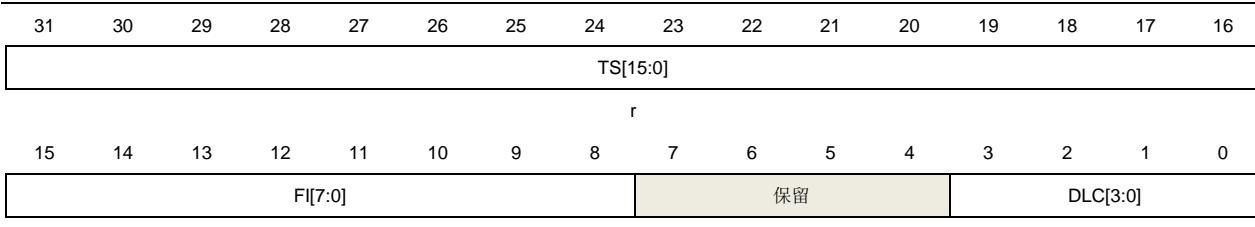
位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	保留	必须保持复位值

#### 25.4.14. CAN 接收 FIFO 邮箱属性寄存器 (CAN\_RFMPRx) (x=0..1)

地址偏移: 0x1B4, 0x1C4

复位值: 0xFFFF XXXX

该寄存器只能按字(32 位)访问



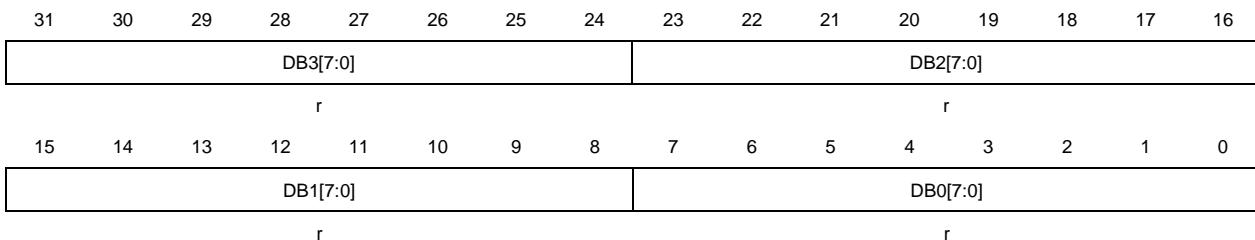
位/位域	名称	描述
31:16	TS[15:0]	时间戳 接收时间戳。
15:8	FI[7:0]	过滤索引 帧通过过滤器时的过滤序号。
7:4	保留	必须保持复位值
3:0	DLC[3:0]	数据长度 DLC[3:0]表示帧内数据长度。

#### 25.4.15. CAN 接收 FIFO 邮箱 Data0 寄存器 (CAN\_RFMD0Rx) (x=0..1)

地址偏移: 0x1B8, 0x1C8

复位值: 0xFFFF XXXX

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

#### 25.4.16. CAN 接收 FIFO 邮箱 Data1 寄存器 (CAN\_RFMD1Rx) (x=0..1)

地址偏移: 0x1BC, 0x1CC

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB7[7:0]								DB6[7:0]							
r								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
r								r							

位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

#### 25.4.17. CAN 过滤器控制寄存器 (CAN\_FCTRL)

地址偏移: 0x200

复位值: 0x2A1C 0E01

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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		HBC2F[5:0]								保留							
rw										rw							

位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	HBC2F[5:0]	CAN2过滤器单元起始位置
7:1	保留	必须保持复位值
0	FLD	过滤器锁禁用 0: 使能过滤器锁 1: 禁用过滤器锁

#### 25.4.18. CAN 过滤器模式寄存器 (CAN\_FMR)

地址偏移: 0x204

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		FM27	FM26	FM25	FM24	FM23	FM22	FM21	FM20	FM19	FM18	FM17	FM16
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FM15	FM14	FM13	FM12	FM11	FM10	FM9	FM8	FM7	FM6	FM5	FM4	FM3	FM2	FM1	FM0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	FMx	过滤器模式 0: 掩码模式 1: 列表模式

#### 25.4.19. CAN 过滤器位宽寄存器 (CAN\_FSR)

地址偏移: 0x20C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	FSx	过滤器位宽 0: 16-bit位宽 1: 32-bit位宽

#### 25.4.20. CAN 过滤器关联 FIFO 寄存器 (CAN\_FAFR)

地址偏移: 0x214

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	FAFx	过滤器关联FIFO 0: 关联FIFO0 1: 关联FIFO1

#### 25.4.21. CAN 过滤器激活寄存器 (CAN\_FWR)

地址偏移: 0x21C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	FWx	<p>过滤器激活</p> <p>0: 没有激活</p> <p>1: 激活工作</p>

25.4.22. CAN 过滤器(x) 数据(y)寄存器 (CAN\_FxDyR) (x=0..27, y=0..1)

地址偏移: 0x240..0x31C

复位值: 0xXXXX XXXX

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	FDx	<p>过滤器数据</p> <p>掩码模式下：</p> <ul style="list-style-type: none"> <li>0：标识符的Bit(x)不需参与比较</li> <li>1：标识符的Bit(x)需要参与比较</li> </ul> <p>列表模式下：</p> <ul style="list-style-type: none"> <li>0：标识符的Bit(x)必须为0</li> <li>1：标识符的Bit(x)必须为1</li> </ul>

#### 25.4.23. CAN PHY 控制寄存器(CAN\_PHYCR)

地址偏移: 0x3FC

复位值: 0x0000 0300

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	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				POMODE[1:0]				保留				PHYEN			
rw								rw							

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	POMODE[1:0]	<p>CAN PHY输出驱动控制位</p> <p>该位由软件置1清0。</p> <ul style="list-style-type: none"> <li>00: 低斜率模式</li> <li>01: 中斜率模式</li> <li>10: 高斜率模式</li> <li>11: 高速模式 (默认值)</li> </ul>
7:1	保留	必须保持复位值
0	PHYEN	<p>PHY 使能位.</p> <p>该位由软件置1清0。该位控制BXCAN1是否使用CAN PHY。</p> <ul style="list-style-type: none"> <li>0: 无CAN PHY模式</li> <li>1: BXCAN1的CAN PHY模式使能</li> </ul>

## 26. 版本历史

表 26-1. 版本历史

版本号.	说明	日期
1.0	初稿发布	2014 年 3 月
2.0	添加 GD32F170/190 产品	2016 年 1 月