

**GigaDevice Semiconductor Inc.**

**GD32E23x**  
**Arm® Cortex®-M23 32-bit MCU**

**用户手册**

1.5 版本

(2022 年 7 月)

# 目录

目录 .....	2
图索引 .....	13
表索引 .....	18
1. 系统及存储器架构 .....	20
1.1. Arm® Cortex®-M23 处理器 .....	20
1.2. 系统架构 .....	21
1.3. 存储器映射 .....	22
1.3.1. 片上 SRAM 存储器 .....	25
1.3.2. 片上闪存 .....	25
1.4. 引导配置 .....	25
1.5. 系统配置寄存器 (SYSCFG) .....	27
1.5.1. 配置寄存器 0 (SYSCFG_CFG0) .....	27
1.5.2. EXTI 源选择寄存器 0 (SYSCFG_EXTISSION0) .....	29
1.5.3. EXTI 源选择寄存器 1 (SYSCFG_EXTISSION1) .....	30
1.5.4. EXTI 源选择寄存器 2 (SYSCFG_EXTISSION2) .....	32
1.5.5. EXTI 源选择寄存器 3 (SYSCFG_EXTISSION3) .....	34
1.5.6. 系统配置寄存器 2 (SYSCFG_CFG2) .....	37
1.5.7. IRQ 延迟寄存器 (SYSCFG_CPU_IRQ_LAT) .....	38
1.6. 设备电子签名 .....	38
1.6.1. 存储容量信息 .....	38
1.6.2. 设备唯一 ID (96 位/位域) .....	39
2. 闪存控制器 (FMC) .....	41
2.1. 简介 .....	41
2.2. 主要特性 .....	41
2.3. 功能描述 .....	41
2.3.1. 闪存结构 .....	41
2.3.2. 读操作 .....	42
2.3.3. FMC_CTL 寄存器解锁 .....	43
2.3.4. 页擦除 .....	43
2.3.5. 整片擦除 .....	44
2.3.6. 主存储闪存块编程 .....	45
2.3.7. OTP 编程 .....	47
2.3.8. 选项字节擦除 .....	47
2.3.9. 选项字节编程 .....	48
2.3.10. 选项字节说明 .....	48

2.3.11. 页擦除/编程保护 .....	49
2.3.12. 安全保护 .....	50
<b>2.4. FMC 寄存器 .....</b>	<b>51</b>
2.4.1. 等待状态寄存器 (FMC_WS) .....	51
2.4.2. 解锁寄存器 (FMC_KEY) .....	51
2.4.3. 选项字节解锁寄存器 (FMC_OBKEY) .....	52
2.4.4. 状态寄存器 (FMC_STAT) .....	52
2.4.5. 控制寄存器 (FMC_CTL) .....	53
2.4.6. 地址寄存器(FMC_ADDR) .....	55
2.4.7. 选项字节状态寄存器 (FMC_OBSTAT) .....	55
2.4.8. 写保护寄存器 (FMC_WP) .....	56
2.4.9. 产品 ID 寄存器 (FMC_PID) .....	56
<b>3. 电源管理单元 (PMU) .....</b>	<b>57</b>
3.1. 简介 .....	57
3.2. 主要特征 .....	57
3.3. 功能说明 .....	57
3.3.1. 电池备份域 .....	58
3.3.2. V <sub>DD</sub> / V <sub>DDA</sub> 电源域 .....	59
3.3.3. 1.2V 电源域 .....	60
3.3.4. 省电模式 .....	60
3.4. PMU 寄存器 .....	63
3.4.1. 控制寄存器 (PMU_CTL) .....	63
3.4.2. 电源控制和状态寄存器 (PMU_CS) .....	64
<b>4. 复位和时钟单元 (RCU) .....</b>	<b>68</b>
4.1. 复位控制单元 (RCTL) .....	68
4.1.1. 简介 .....	68
4.1.2. 功能描述 .....	68
4.2. 时钟控制单元 (CCTL) .....	69
4.2.1. 简介 .....	69
4.2.2. 主要特性 .....	70
4.2.3. 功能描述 .....	71
4.3. RCU 寄存器 .....	74
4.3.1. 控制寄存器 0 (RCU_CTL0) .....	74
4.3.2. 配置寄存器 0 (RCU_CFG0) .....	75
4.3.3. 中断寄存器 (RCU_INT) .....	79
4.3.4. APB2 复位寄存器 (RCU_APB2RST) .....	81
4.3.5. APB1 复位寄存器 (RCU_APB1RST) .....	83
4.3.6. AHB 使能寄存器 (RCU_AHBEN) .....	84
4.3.7. APB2 使能寄存器 (RCU_APB2EN) .....	86
4.3.8. APB1 使能寄存器 (RCU_APB1EN) .....	87

4.3.9. 备份域控制寄存器 (RCU_BDCTL) .....	89
4.3.10. 复位源/时钟寄存器 (RCU_RSTSCK) .....	90
4.3.11. AHB 复位寄存器 (RCU_AHBRST) .....	92
4.3.12. 配置寄存器 1 (RCU_CFG1) .....	93
4.3.13. 配置寄存器 2 (RCU_CFG2) .....	94
4.3.14. 控制寄存器 1 (RCU_CTL1) .....	95
4.3.15. 电源解锁寄存器 (RCU_VKEY) .....	96
4.3.16. 深度睡眠模式电压寄存器 (RCU_DSV) .....	96
<b>5. 中断/事件控制器 (EXTI) .....</b>	<b>97</b>
5.1. 简介 .....	97
5.2. 主要特性 .....	97
5.3. 中断功能描述 .....	97
5.4. 外部中断及事件(EXTI)结构框图 .....	100
5.5. 外部中断及事件功能概述 .....	100
<b>5.6. EXTI 寄存器 .....</b>	<b>103</b>
5.6.1. 中断使能寄存器 (EXTI_INTEN) .....	103
5.6.2. 事件使能寄存器 (EXTI_EVENT) .....	103
5.6.3. 上升沿触发使能寄存器 (EXTI_RTEN) .....	104
5.6.4. 下降沿触发使能寄存器 (EXTI_FTEN) .....	104
5.6.5. 软件中断事件寄存器 (EXTI_SWIEV) .....	105
5.6.6. 挂起寄存器 (EXTI_PD) .....	106
<b>6. 通用和备用输入/输出接口 (GPIO 和 AFIO) .....</b>	<b>107</b>
6.1. 简介 .....	107
6.2. 主要特性 .....	107
6.3. 功能描述 .....	107
6.3.1. GPIO 引脚配置 .....	108
6.3.2. 备用功能 (AF) .....	109
6.3.3. 附加功能 .....	109
6.3.4. 输入配置 .....	109
6.3.5. 输出配置 .....	109
6.3.6. 模拟配置 .....	110
6.3.7. 备用功能 (AF) 配置 .....	110
6.3.8. GPIO 锁定功能 .....	111
6.3.9. GPIO 单周期输出翻转功能 .....	111
<b>6.4. GPIO 寄存器 .....</b>	<b>112</b>
6.4.1. 端口控制寄存器 (GPIOx_CTL, x=A..C,F) .....	112
6.4.2. 端口输出模式寄存器 (GPIOx_OMODE, x=A..C,F) .....	113
6.4.3. 端口输出速度寄存器 (GPIOx_OSPD, x=A..C,F) .....	115
6.4.4. 端口上拉/下拉寄存器 (GPIOx_PUD, x=A..C,F) .....	117

6.4.5.	端口输入状态寄存器 (GPIO <sub>x</sub> _ISTAT, x=A..C,F) .....	118
6.4.6.	端口输出控制寄存器 (GPIO <sub>x</sub> _OCTL, x=A..C,F) .....	119
6.4.7.	端口位操作寄存器 (GPIO <sub>x</sub> _BOP, x=A..C,F) .....	119
6.4.8.	端口配置锁定寄存器 (GPIO <sub>x</sub> _LOCK, x=A,B) .....	120
6.4.9.	备用功能选择寄存器 0 (GPIO <sub>x</sub> _AFSEL0, x=A,B,C) .....	121
6.4.10.	备用功能选择寄存器 1 (GPIO <sub>x</sub> _AFSEL1, x=A,B,C) .....	122
6.4.11.	位清除寄存器 (GPIO <sub>x</sub> _BC, x=A..C,F) .....	123
6.4.12.	端口位翻转寄存器 (GPIO <sub>x</sub> _TG, x=A..C,F) .....	124
<b>7.</b>	<b>循环冗余校验管理单元 (CRC).....</b>	<b>125</b>
7.1.	简介 .....	125
7.2.	主要特性 .....	125
7.3.	功能描述 .....	126
7.4.	<b>CRC 寄存器.....</b>	<b>127</b>
7.4.1.	数据寄存器 (CRC_DATA).....	127
7.4.2.	独立数据寄存器 (CRC_FDATA) .....	127
7.4.3.	控制寄存器 (CRC_CTL) .....	128
7.4.4.	初值寄存器 (CRC_IDATA).....	128
7.4.5.	多项式寄存器 (CRC_POLY).....	129
<b>8.</b>	<b>直接存储器访问控制器 (DMA) .....</b>	<b>130</b>
8.1.	简介 .....	130
8.2.	主要特性 .....	130
8.3.	结构框图 .....	131
8.4.	功能描述 .....	131
8.4.1.	DMA 操作 .....	131
8.4.2.	外设握手 .....	133
8.4.3.	仲裁 .....	133
8.4.4.	地址生成 .....	133
8.4.5.	循环模式 .....	133
8.4.6.	存储器到存储器模式 .....	134
8.4.7.	通道配置 .....	134
8.4.8.	中断 .....	134
8.4.9.	DMA 请求映射 .....	135
8.5.	<b>DMA 寄存器.....</b>	<b>138</b>
8.5.1.	中断标志位寄存器 (DMA_INTF) .....	138
8.5.2.	中断标志位清除寄存器 (DMA_INTC) .....	138
8.5.3.	通道 x 控制寄存器 (DMA_CHxCTL) .....	139
8.5.4.	通道 x 计数寄存器 (DMA_CHxCNT) .....	141
8.5.5.	通道 x 外设基地址寄存器 (DMA_CHxPADDR) .....	142
8.5.6.	通道 x 存储器基地址寄存器 (DMA_CHxMADDR) .....	142

<b>9. 调试 (DBG) .....</b>	<b>144</b>
<b>9.1. 简介 .....</b>	<b>144</b>
<b>9.2. 串行调试接口简介 .....</b>	<b>144</b>
9.2.1. 引脚分配 .....	144
<b>9.3. 调试保持功能描述 .....</b>	<b>144</b>
9.3.1. 低功耗模式调试支持 .....	144
9.3.2. TIMER, I2C, RTC, WWDGT 和 FWDGT 外设调试支持 .....	145
<b>9.4. DBG 寄存器 .....</b>	<b>146</b>
9.4.1. ID 寄存器 (DBG_ID) .....	146
9.4.2. 控制寄存器 0 (DBG_CTL0) .....	146
9.4.3. 控制寄存器 1 (DBG_CTL1) .....	148
<b>10. 模数转换器(ADC).....</b>	<b>150</b>
<b>10.1. 简介 .....</b>	<b>150</b>
<b>10.2. 主要特征 .....</b>	<b>150</b>
<b>10.3. 引脚和内部信号 .....</b>	<b>151</b>
<b>10.4. 功能说明 .....</b>	<b>152</b>
10.4.1. 前置校准功能 .....	152
10.4.2. 双时钟域架构 .....	153
10.4.3. ADC 使能 .....	153
10.4.4. 常规序列 .....	153
10.4.5. 运行模式 .....	153
10.4.6. 转换结果阈值监测功能 .....	156
10.4.7. 数据存储模式 .....	156
10.4.8. 采样时间配置 .....	157
10.4.9. 外部触发配置 .....	157
10.4.10. DMA 请求 .....	158
10.4.11. ADC 内部通道 .....	158
10.4.12. 可编程分辨率(DRES) .....	158
10.4.13. 片上硬件过采样 .....	159
10.4.14. ADC 中断 .....	161
<b>10.5. ADC 寄存器 .....</b>	<b>162</b>
10.5.1. 状态寄存器 (ADC_STAT) .....	162
10.5.2. 控制寄存器 0 (ADC_CTL0) .....	162
10.5.3. 控制寄存器 1 (ADC_CTL1) .....	164
10.5.4. 采样时间寄存器 0 (ADC_SAMPT0) .....	165
10.5.5. 采样时间寄存器 1 (ADC_SAMPT1) .....	166
10.5.6. 看门狗高阈值寄存器 (ADC_WDHT) .....	167
10.5.7. 看门狗低阈值寄存器 (ADC_WDLT) .....	167
10.5.8. 常规序列寄存器 0 (ADC_RSQ0) .....	168
10.5.9. 常规序列寄存器 1 (ADC_RSQ1) .....	168

10.5.10. 常规序列寄存器 2 (ADC_RSQ2) .....	169
10.5.11. 常规数据寄存器 (ADC_RDATA) .....	169
10.5.12. 过采样控制寄存器(ADC_OVSAMPCTL).....	170
<b>11. 比较器 (CMP) .....</b>	<b>172</b>
<b>11.1. 简介.....</b>	<b>172</b>
<b>11.2. 主要特性.....</b>	<b>172</b>
<b>11.3. 功能描述.....</b>	<b>172</b>
11.3.1. 比较器时钟和复位 .....	173
11.3.2. 比较器的 I/O 配置.....	173
11.3.3. 比较器供电模式.....	173
11.3.4. 比较器迟滞 .....	173
11.3.5. 比较器寄存器写保护 .....	174
<b>11.4. 比较器寄存器 .....</b>	<b>175</b>
11.4.1. 控制状态寄存器(CMP_CS) .....	175
<b>12. 看门狗定时器 (WDGT) .....</b>	<b>177</b>
<b>12.1. 独立看门狗定时器 (FWDGT) .....</b>	<b>177</b>
12.1.1. 简介 .....	177
12.1.2. 主要特性.....	177
12.1.3. 功能描述.....	177
12.1.4. FWDGT 寄存器 .....	180
<b>12.2. 窗口看门狗定时器 (WWDGT) .....</b>	<b>183</b>
12.2.1. 简介 .....	183
12.2.2. 主要特性.....	183
12.2.3. 功能描述.....	183
12.2.4. WWDGT 寄存器.....	185
<b>13. 实时时钟 (RTC) .....</b>	<b>187</b>
<b>13.1. 简介.....</b>	<b>187</b>
<b>13.2. 主要特性.....</b>	<b>187</b>
<b>13.3. 功能描述.....</b>	<b>188</b>
13.3.1. 结构框图.....	188
13.3.2. 时钟源和预分频.....	188
13.3.3. 影子寄存器 .....	189
13.3.4. 位域可屏蔽可配置的闹钟 .....	189
13.3.5. RTC 初始化和配置 .....	189
13.3.6. 读取日历.....	191
13.3.7. RTC 复位.....	192
13.3.8. RTC 移位功能 .....	192
13.3.9. RTC 参考时钟检测 .....	193
13.3.10. RTC 数字平滑校准 .....	193

13.3.11. 时间戳功能（仅针对 GD32E230xx 产品） .....	195
13.3.12. 侵入检测 .....	195
13.3.13. 校准时钟输出 .....	196
13.3.14. 闹钟输出 .....	196
13.3.15. RTC 省电模式管理 .....	197
13.3.16. RTC 中断 .....	197
<b>13.4. RTC 寄存器 .....</b>	<b>199</b>
13.4.1. 时间寄存器 (RTC_TIME) .....	199
13.4.2. 日期寄存器 (RTC_DATE) .....	199
13.4.3. 控制寄存器 (RTC_CTL) .....	200
13.4.4. 状态寄存器 (RTC_STAT) .....	203
13.4.5. 预分频寄存器 (RTC_PSC) .....	206
13.4.6. 闹钟 0 时间日期寄存器 (RTC_ALRM0TD) .....	207
13.4.7. 写保护钥匙寄存器 (RTC_WPK) .....	208
13.4.8. 亚秒寄存器 (RTC_SS) .....	209
13.4.9. 移位控制寄存器 (RTC_SHIFTCTL) .....	209
13.4.10. 时间戳时间寄存器 (RTC_TTS) .....	210
13.4.11. 时间戳日期寄存器 (RTC_DTS) .....	211
13.4.12. 时间戳亚秒寄存器 (RTC_SSTS) .....	211
13.4.13. 高精度频率补偿寄存器 (RTC_HRFC) .....	212
13.4.14. 侵入寄存器 (RTC_TAMP) .....	213
13.4.15. 闹钟 0 亚秒寄存器 (RTC_ALRM0SS) .....	217
13.4.16. 备份寄存器 (RTC_BKPx) (x=0..4) .....	218
<b>14. 定时器 (TIMER) .....</b>	<b>219</b>
<b>14.1. 高级定时器 (TIMERx, x=0) .....</b>	<b>220</b>
14.1.1. 简介 .....	220
14.1.2. 主要特征 .....	220
14.1.3. 结构框图 .....	220
14.1.4. 功能说明 .....	221
14.1.5. TIMERx 寄存器 (x=0) .....	246
<b>14.2. 通用定时器 L0 (TIMERx, x=2) .....</b>	<b>271</b>
14.2.1. 简介 .....	271
14.2.2. 主要特征 .....	271
14.2.3. 结构框图 .....	271
14.2.4. 功能说明 .....	272
14.2.5. TIMERx 寄存器 (x=2) .....	285
<b>14.3. 通用定时器 L2 (TIMERx, x=13) .....</b>	<b>307</b>
14.3.1. 简介 .....	307
14.3.2. 主要特征 .....	307
14.3.3. 结构框图 .....	307
14.3.4. 功能说明 .....	307
14.3.5. TIMERx 寄存器 (x=13) .....	315

<b>14.4. 通用定时器 L3 (TIMERx,x=14) .....</b>	<b>325</b>
14.4.1. 简介 .....	325
14.4.2. 主要特性.....	325
14.4.3. 结构框图.....	325
14.4.4. 功能描述.....	326
14.4.5. TIMERx 寄存器(x=14).....	342
<b>14.5. 通用定时器 L4 (TIMERx,x=15,16) .....</b>	<b>361</b>
14.5.1. 简介 .....	361
14.5.2. 主要特性.....	361
14.5.3. 结构框图.....	361
14.5.4. 功能描述.....	362
14.5.5. TIMERx 寄存器(x=15,16).....	375
<b>14.6. 基本定时器 (TIMERx, x=5) .....</b>	<b>390</b>
14.6.1. 简介 .....	390
14.6.2. 主要特征.....	390
14.6.3. 结构框图.....	390
14.6.4. 功能说明.....	390
14.6.5. TIMERx 寄存器(x=5).....	394
<b>15. 红外线接口 (IFRP) .....</b>	<b>399</b>
15.1. 简介.....	399
15.2. 主要特性.....	399
15.3. 功能描述.....	399
<b>16. 通用同步异步收发器 (USART) .....</b>	<b>401</b>
16.1. 简介.....	401
16.2. 主要特性.....	401
16.3. 功能描述.....	402
16.3.1. USART 帧格式 .....	403
16.3.2. 波特率发生 .....	404
16.3.3. USART 发送器 .....	404
16.3.4. USART 接收器 .....	405
16.3.5. DMA 方式访问数据缓冲区.....	407
16.3.6. 硬件流控制 .....	408
16.3.7. 多处理器通信 .....	409
16.3.8. LIN 模式 .....	410
16.3.9. 同步通信模式 .....	410
16.3.10. 串行红外 (IrDA SIR) 编解码功能模块 .....	411
16.3.11. 半双工通信模式 .....	412
16.3.12. 智能卡 (ISO7816-3) 模式 .....	413
16.3.13. ModBus 通信 .....	414
16.3.14. 接收 FIFO .....	415

---

16.3.15. 从 Deepsleep 模式唤醒 .....	415
16.3.16. USART 中断 .....	415
<b>16.4. USART 寄存器 .....</b>	<b>418</b>
16.4.1. USART 控制寄存器 0 (USART_CTL0) .....	418
16.4.2. USART 控制寄存器 1 (USART_CTL1) .....	420
16.4.3. USART 控制寄存器 2 (USART_CTL2) .....	422
16.4.4. USART 波特率寄存器 (USART_BAUD) .....	425
16.4.5. USART 保护时间和预分频器寄存器 (USART_GP) .....	426
16.4.6. USART 接收超时寄存器 (USART_RT) .....	427
16.4.7. USART 请求寄存器 (USART_CMD) .....	427
16.4.8. USART 状态寄存器 (USART_STAT) .....	428
16.4.9. USART 中断标志清除寄存器 (USART_INTC) .....	431
16.4.10. USART 数据接收寄存器 (USART_RDATA) .....	433
16.4.11. USART 数据发送寄存器 (USART_TDATA) .....	433
16.4.12. USART 兼容性控制寄存器 (USART_CHC) .....	434
16.4.13. USART 接收 FIFO 控制和状态寄存器 (USART_RFCS) .....	434
<b>17. 内部集成电路总线接口 (I2C) .....</b>	<b>436</b>
17.1. 简介 .....	436
17.2. 主要特征 .....	436
17.3. 功能说明 .....	436
17.3.1. SDA 线和 SCL 线 .....	437
17.3.2. 数据有效性 .....	437
17.3.3. 开始和停止信号 .....	438
17.3.4. 时钟同步 .....	438
17.3.5. 仲裁 .....	438
17.3.6. I2C 通讯流程 .....	439
17.3.7. 软件编程模型 .....	440
17.3.8. SCL 线控制 .....	447
17.3.9. DMA 模式下数据传输 .....	448
17.3.10. 报文错误校验 .....	448
17.3.11. SMBus 支持 .....	448
17.3.12. SAM_V 支持 .....	450
17.3.13. 状态、错误和中断 .....	450
17.4. I2C 寄存器 .....	451
17.4.1. 控制寄存器 0 (I2C_CTL0) .....	451
17.4.2. 控制寄存器 1 (I2C_CTL1) .....	453
17.4.3. 从机地址寄存器 0 (I2C_SADDR0) .....	454
17.4.4. 从机地址寄存器 1 (I2C_SADDR1) .....	454
17.4.5. 传输缓冲区寄存器 (I2C_DATA) .....	455
17.4.6. 传输状态寄存器 0 (I2C_STAT0) .....	455
17.4.7. 传输状态寄存器 1 (I2C_STAT1) .....	457

17.4.8. 时钟配置寄存器 (I2C_CKCFG) .....	459
17.4.9. 上升时间寄存器 (I2C_RT) .....	459
17.4.10. SAM 控制状态寄存器 (I2C_SAMCS) .....	460
17.4.11. 快速+ 模式配置寄存器 (I2C_FMPCFG) .....	461
<b>18. 串行外设接口/片上音频接口 (SPI/I2S) .....</b>	<b>462</b>
<b>18.1. 简介.....</b>	<b>462</b>
<b>18.2. 主要特性.....</b>	<b>462</b>
18.2.1. SPI 主要特性.....	462
18.2.2. I2S 主要特性 .....	462
<b>18.3. SPI 功能说明 .....</b>	<b>463</b>
18.3.1. SPI 结构框图 .....	463
18.3.2. SPI 信号线描述 .....	463
18.3.3. SPI 时序和数据帧格式 .....	464
18.3.4. 独立发送和接收缓冲区 .....	465
18.3.5. NSS 功能 .....	466
18.3.6. SPI 运行模式.....	468
18.3.7. DMA 功能 .....	476
18.3.8. CRC 功能 .....	476
18.3.9. SPI 中断 .....	477
<b>18.4. I2S 功能说明 .....</b>	<b>479</b>
18.4.1. I2S 结构框图 .....	479
18.4.2. I2S 信号线描述.....	479
18.4.3. I2S 音频标准 .....	479
18.4.4. I2S 时钟 .....	487
18.4.5. 运行 .....	488
18.4.6. DMA 功能 .....	492
18.4.7. I2S 中断 .....	492
<b>18.5. SPI/I2S 寄存器 .....</b>	<b>494</b>
18.5.1. 控制寄存器 0 (SPI_CTL0) .....	494
18.5.2. 控制寄存器 1 (SPI_CTL1) .....	496
18.5.3. 状态寄存器 (SPI_STAT) .....	497
18.5.4. 数据寄存器 (SPI_DATA) .....	499
18.5.5. CRC 多项式寄存器 (SPI_CRCPOLY) .....	500
18.5.6. 接收 CRC 寄存器 (SPI_RCRC) .....	500
18.5.7. 发送 CRC 寄存器 (SPI_TCRC) .....	501
18.5.8. I2S 控制寄存器 (SPI_I2SCTL) .....	502
18.5.9. I2S 时钟预分频寄存器 (SPI_I2SPSC) .....	503
18.5.10. SPI1 四线 SPI 控制寄存器 (SPI_QCTL) .....	504
<b>19. 运算放大器 (OPA) .....</b>	<b>505</b>
<b>19.1. 简介.....</b>	<b>505</b>

---

19.2. 主要特性.....	505
19.3. 功能描述.....	505
19.3.1. 使能运放.....	505
19.3.2. 使能运放.....	505
19.3.3. 使能运放.....	505
20. 附录 .....	506
20.1. 寄存器表中使用的缩写列表 .....	506
20.2. 术语表 .....	506
20.3. 可用外设.....	506
21. 版本历史 .....	507

# 图索引

图 1-1. Cortex®-M23 处理器结构框图 .....	21
图 1-2. GD32E23x 系列器件的系统架构示意图.....	22
图 2-1. 页擦除操作流程 .....	44
图 2-2. 整片擦除操作流程 .....	45
图 2-3. 字编程操作流程 .....	47
图 3-1. 电源域概览 .....	58
图 3-2. 上电/掉电复位波形图.....	59
图 3-3. LVD 阈值波形图.....	60
图 4-1. 系统复位电路 .....	69
图 4-2. 时钟树 .....	70
图 4-3. HXTAL 时钟源 .....	71
图 5-1. EXTI 结构框图.....	100
图 6-1. GPIO 端口位的基本结构 .....	108
图 6-2. 输入配置的基本结构.....	109
图 6-3. 输出配置的基本结构.....	110
图 6-4. 模拟配置的基本结构 .....	110
图 6-5. 备用功能配置的基本结构 .....	111
图 7-1. CRC 管理单元框图.....	125
图 8-1. DMA 结构框图 .....	131
图 8-2. 握手机制.....	133
图 8-3. DMA 中断逻辑图 .....	135
图 8-4. DMA 请求映射 .....	136
图 10-1. ADC 模块框图 .....	152
图 10-2. 单次运行模式 .....	153
图 10-3. 连续运行模式 .....	154
图 10-4. 扫描运行模式，且连续运行模式失能 .....	155
图 10-5. 扫描运行模式，连续运行模式使能.....	155
图 10-6. 间断运行模式 .....	155
图 10-7. 12 位数据存储模式.....	156
图 10-8. 10 位数据存储模式.....	156
图 10-9. 8 位数据存储模式.....	157
图 10-10. 6 位数据存储模式.....	157
图 10-11. 20 位到 16 位的结果截断 .....	159
图 10-12. 右移 5 位和取整的数例 .....	160
图 11-1. 比较器框图 .....	173
图 11-2. 比较器迟滞 .....	174
图 12-1. 独立看门狗定时器框图 .....	178
图 12-2. 窗口看门狗定时器框图 .....	183
图 12-3. 窗口看门狗定时器时序图 .....	184
图 13-1. RTC 结构框图 .....	188

图 14-1. 高级定时器结构框图.....	221
图 14-2. 内部时钟分频为 1 时，计数器的时序图 .....	222
图 14-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图.....	223
图 14-4. 向上计数时序图， $PSC=0/2$ .....	224
图 14-5. 向上计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器的值 .....	224
图 14-6. 向下计数时序图， $PSC=0/2$ .....	225
图 14-7. 向下计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器值.....	226
图 14-8. 中央计数模式计数器时序图.....	227
图 14-9. 中央计数模式下计数器重复时序图.....	228
图 14-10. 在向上计数模式下计数器重复时序图 .....	228
图 14-11. 在向下计数模式下计数器重复时序图 .....	229
图 14-12. 通道输入捕获原理 .....	229
图 14-13. 通道输出比较原理（带有互补输出的通道， $x=0,1,2$ ） .....	230
图 14-14. 通道输出比较原理 .....	231
图 14-15. 三种输出比较模式 .....	232
图 14-16. EAPWM 时序图 .....	233
图 14-17. CAPWM 时序图 .....	233
图 14-18. 带死区时间的互补输出 .....	236
图 14-19. 通道响应中止输入（高电平有效）时，输出信号的行为 .....	237
图 14-20. 在编码器模式 2 且 $CI0FE0$ 极性不反相时计数器行为 .....	238
图 14-21. 在编码器模式 2 且 $CI0FE0$ 极性反相时计数器行为 .....	238
图 14-22. 霍尔传感器用在 BLDC 电机控制中 .....	239
图 14-23. 两个定时器之间的霍尔传感器时序图 .....	240
图 14-24. 复位模式.....	241
图 14-25. 暂停模式.....	241
图 14-26. 事件模式.....	242
图 14-27. 单脉冲模式， $TIMERx_CHxCV = 4$ , $TIMERx_CAR=99$ .....	242
图 14-28. 定时器 0 主/从模式的例子 .....	243
图 14-29. 用定时器 2 的使能信号触发定时器 0 .....	244
图 14-30. 用定时器 2 的 $CI0$ 信号来触发定时器 0 和定时器 2 .....	245
图 14-31. 通用定时器 L0 结构框图 .....	272
图 14-32. 内部时钟分频为 1 时，计数器的时序图 .....	273
图 14-33. 当 PSC 数值从 0 变到 2 时，计数器的时序图.....	274
图 14-34. 向上计数时序图， $PSC=0/2$ .....	275
图 14-35. 向上计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器的值 .....	275
图 14-36. 向下计数时序图， $PSC=0/2$ .....	276
图 14-37. 向下计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器值.....	277
图 14-38. 中央计数模式计数器时序图.....	278
图 14-39. 通道输入捕获原理 .....	279
图 14-40. 通道输出比较原理（ $x=0,1,2,3$ ） .....	280
图 14-41. 三种输出比较模式 .....	281
图 14-42. EAPWM 时序图 .....	282
图 14-43. CAPWM 时序图 .....	282
图 14-44. 通用定时器 L2 结构框图 .....	307

图 14-45. 内部时钟分频为 1 时, 计数器的时序图 .....	308
图 14-46. 当 PSC 数值从 0 变到 2 时, 计数器的时序图 .....	309
图 14-47. 向上计数时序图, PSC=0/2 .....	310
图 14-48. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值 .....	310
图 14-49. 通道输入捕获原理 .....	311
图 14-50. 通道输出比较原理 .....	312
图 14-51. 三种输出比较模式 .....	313
图 14-52. PWM 时序图 .....	314
图 14-53. 通用定时器 L3 结构框图 .....	326
图 14-54. 内部时钟分频为 1 时, 计数器的时序图 .....	327
图 14-55. 当 PSC 数值从 0 变到 2 时, 计数器的时序图 .....	328
图 14-56. 向上计数时序图, PSC=0/2 .....	329
图 14-57. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值 .....	329
图 14-58. 在向上计数模式下计数器重复时序图 .....	330
图 14-59. 通道输入捕获原理 .....	331
图 14-60. 通道输出比较原理 (带有互补输出的通道, x=0) .....	332
图 14-61. 通道输出比较原理 .....	332
图 14-62. 三种输出比较模式 .....	333
图 14-63. PWM 时序图 .....	334
图 14-64. 带死区时间的互补输出 .....	337
图 14-65. 通道响应中止输入 (高电平有效) 时, 输出信号的行为 .....	338
图 14-66. 复位模式下的控制电路 .....	339
图 14-67. 暂停模式下的控制电路 .....	339
图 14-68. 事件模式下的控制电路 .....	340
图 14-69. 单脉冲模式, TIMERx_CHxCV = 4 TIMERx_CAR=99 .....	340
图 14-70. 通用定时器 L4 结构框图 .....	362
图 14-71. 内部时钟分频为 1 时, 计数器的时序图 .....	363
图 14-72. 当 PSC 数值从 0 变到 2 时, 计数器的时序图 .....	363
图 14-73. 向上计数时序图, PSC=0/2 .....	364
图 14-74. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值 .....	365
图 14-75. 在向上计数模式下计数器重复时序图 .....	366
图 14-76. 通道输入捕获原理 .....	367
图 14-77. 通道输出比较原理 (带有互补输出的通道, x=0) .....	368
图 14-78. 三种输出比较模式 .....	369
图 14-79. PWM 时序图 .....	370
图 14-80. 带死区时间的互补输出 .....	372
图 14-81. 通道响应中止输入 (高电平有效) 时, 输出信号的行为 .....	373
图 14-82. 单脉冲模式, TIMERx_CHxCV = 4 TIMERx_CAR=99 .....	374
图 14-83. 基本定时器结构框图 .....	390
图 14-84. 内部时钟分频为 1 时, 计数器的时序图 .....	391
图 14-85. 当 PSC 数值从 0 变到 2 时, 计数器的时序图 .....	391
图 14-86. 向上计数时序图, PSC=0/2 .....	392
图 14-87. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值 .....	393
图 15-1. IFRP 输出时序图 1 .....	399

图 15-2. IFRP 输出时序图 2 .....	400
图 15-3. IFRP 输出时序图 3 .....	400
图 16-1. USART 模块内部框图 .....	403
图 16-2. USART 字符帧（8 数据位和 1 停止位） .....	403
图 16-3. USART 发送步骤 .....	405
图 16-4. 过采样方式接收一个数据位（OSB=0） .....	406
图 16-5. 采用 DMA 方式实现 USART 数据发送配置步骤 .....	407
图 16-6. 采用 DMA 方式实现 USART 数据接收配置步骤 .....	408
图 16-7. 两个 USART 之间的硬件流控制 .....	408
图 16-8. 硬件流控制 .....	409
图 16-9. 空闲状态下检测断开帧 .....	410
图 16-10. 数据传输过程中检测断开帧 .....	410
图 16-11. 同步模式下的 USART 示例 .....	411
图 16-12. 8-bit 格式的 USART 同步通信波形（CLEN=1） .....	411
图 16-13. IrDA SIR ENDEC 模块 .....	412
图 16-14. IrDA 数据调制 .....	412
图 16-15. ISO7816-3 数据帧格式 .....	413
图 16-16. USART 接收 FIFO 结构 .....	415
图 16-17. USART 中断映射框图 .....	417
图 17-1. I2C 模块框图 .....	437
图 17-2. 数据有效性 .....	438
图 17-3. 起始和停止信号 .....	438
图 17-4. 时钟同步 .....	438
图 17-5. SDA 线仲裁 .....	439
图 17-6. 7 位地址的 I2C 通讯流程 .....	439
图 17-7. 10 位地址的 I2C 通讯流程（主机发送） .....	439
图 17-8. 10 位地址的 I2C 通讯流程（主机接收） .....	439
图 17-9. 从机发送模式（10 位地址模式） .....	441
图 17-10. 从机接收模式（10 位地址模式） .....	442
图 17-11. 主机发送模式（10 位地址模式） .....	443
图 17-12. 主机接收使用方案 A 模式（10 位地址模式） .....	445
图 17-13. 主机接收使用方案 B 模式（10 位地址模式） .....	447
图 18-1. SPI 结构框图 .....	463
图 18-2. SPI0 常规模式下的时序图 .....	464
图 18-3. SPI1 常规模式下的时序图 .....	465
图 18-4. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0） .....	465
图 18-5. 发送/接收缓冲区 .....	466
图 18-6. 典型的全双工模式连接 .....	469
图 18-7. 典型的单工模式连接（主机：接收，从机：发送） .....	469
图 18-8. 典型的单工模式连接（主机：只发送，从机：接收） .....	469
图 18-9. 典型的双向线连接 .....	469
图 18-10. 主机 TI 模式在不连续发送时的时序图 .....	471
图 18-11. 主机 TI 模式在连续发送时的时序图 .....	472
图 18-12. 从机 TI 模式时序图 .....	472

图 18-13. NSS 脉冲模式时序图（主机连续发送）	473
图 18-14. SPI 四线模式写操作时序图	474
图 18-15. SPI 四线模式读操作时序图	474
图 18-16. I2S 结构框图	479
图 18-17. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)	480
图 18-18. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)	480
图 18-19. I2S 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)	480
图 18-20. I2S 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)	480
图 18-21. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	481
图 18-22. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	481
图 18-23. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	481
图 18-24. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	481
图 18-25. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)	482
图 18-26. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)	482
图 18-27. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)	482
图 18-28. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)	482
图 18-29. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	482
图 18-30. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	482
图 18-31. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	483
图 18-32. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	483
图 18-33. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	483
图 18-34. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	483
图 18-35. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	484
图 18-36. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	484
图 18-37. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)	484
图 18-38. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)	484
图 18-39. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)	484
图 18-40. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)	485
图 18-41. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)	485
图 18-42. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)	485
图 18-43. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)	485
图 18-44. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)	485
图 18-45. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)	485
图 18-46. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)	486
图 18-47. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)	486
图 18-48. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)	486
图 18-49. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)	486
图 18-50. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)	486
图 18-51. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)	486
图 18-52. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)	487
图 18-53. I2S 时钟生成结构框图	487
图 18-54. I2S 初始化流程	489
图 18-55. I2S 主机接收禁能流程	491

# 表索引

表 1-1. GD32E23x 系列器件的存储器映射表 .....	23
表 1-2. Flash 模块组织 .....	25
表 1-3. 引导模式.....	26
表 2-1. 闪存的基地址和大小.....	41
表 2-2. WSCNT 位域与 AHB 时钟频率之间的关系.....	42
表 2-3. 选项字节 .....	48
表 2-4. OB_WP 位对应页保护 .....	49
表 3-1. 节电模式总结 .....	61
表 4-1. 时钟源的选择 .....	73
表 4-2. 深度睡眠模式下内核电压选择 .....	73
表 5-1. Cortex-M23 中的 NVIC 异常类型 .....	97
表 5-2. 中断向量表 .....	98
表 5-3. EXTI 触发源 .....	100
表 6-1. GPIO 配置表 .....	107
表 8-1. DMA 传输操作 .....	132
表 8-2. 中断事件 .....	134
表 8-3. DMA 各通道请求表 .....	137
表 10-1. ADC 内部输入信号 .....	151
表 10-2. ADC 输入引脚定义 .....	151
表 10-3. ADC 的外部触发源 .....	157
表 10-4. 不同分辨率对应的 $t_{CONV}$ 时间 .....	158
表 10-5. 不同 N 和 M 组合的最大输出值（灰色值表示截断） .....	160
表 12-1. 独立看门狗定时器在 40kHz (IRC40K) 时的最小/最大超时周期 .....	178
表 12-2. 在 72MHz ( $f_{PCLK1}$ ) 时的最大/最小超时值 .....	184
表 13-1. 省电模式管理 .....	197
表 13-2. 中断控制 .....	197
表 14-1. 定时器 (TIMERx) 分为六种类型 .....	219
表 14-2. 由参数控制的互补输出表 .....	234
表 14-3. 不同编码器模式下的计数方向 .....	237
表 14-4. 从模式示例 .....	240
表 14-5. 由参数控制的互补输出表 .....	335
表 14-6. 从模式例子列表 .....	338
表 14-7. 由参数控制的互补输出表 .....	371
表 16-1. USART 重要引脚描述 .....	402
表 16-2. 停止位配置 .....	404
表 16-3. USART 中断请求 .....	415
表 17-1. I2C 总线术语说明（参考飞利浦 I2C 规范） .....	437
表 17-2. 事件状态标志位 .....	450
表 17-3. 错误标志位 .....	450
表 18-1. SPI 信号描述 .....	463

---

表 18-2. SPI 四线信号描述.....	464
表 18-3. 从机模式 NSS 功能 .....	467
表 18-4. 主机模式 NSS 功能 .....	467
表 18-5. SPI 运行模式.....	468
表 18-6. SPI 中断请求.....	478
表 18-7. I2S 比特率计算公式 .....	487
表 18-8. 音频采样频率计算公式.....	487
表 18-9. 各种运行模式下 I2S 接口信号的方向 .....	488
表 18-10. I2S 中断 .....	493
表 20-1. 寄存器功能位访问属性.....	506
表 20-2. 术语 .....	506
表 21-1. 版本历史.....	507

## 1. 系统及存储器架构

GD32E23x系列器件是基于Arm® Cortex®-M23处理器的32位通用微控制器。Cortex®-M23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AHB总线上执行。存储器的组织采用了ARMv8M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®-M23 处理器

Cortex®-M23处理器是一个低功耗32位处理器。适用于需要一个区域优化处理器来进行深度嵌入式应用的场景。Cortex®-M23处理器为开发人员提供了显著的好处，包括：

- 一个简单的体系结构，易于学习和编程；
- 超低功耗、高效节能；
- 优秀的代码密度；
- 确定性、高性能中断处理；
- 向上兼容Cortex-M处理器家族系列；

Cortex®-M23处理器通过精简强大的指令集和广泛优化的设计提供高效处理能力，提供包括单周期乘法器和17周期分频器的高端处理硬件。

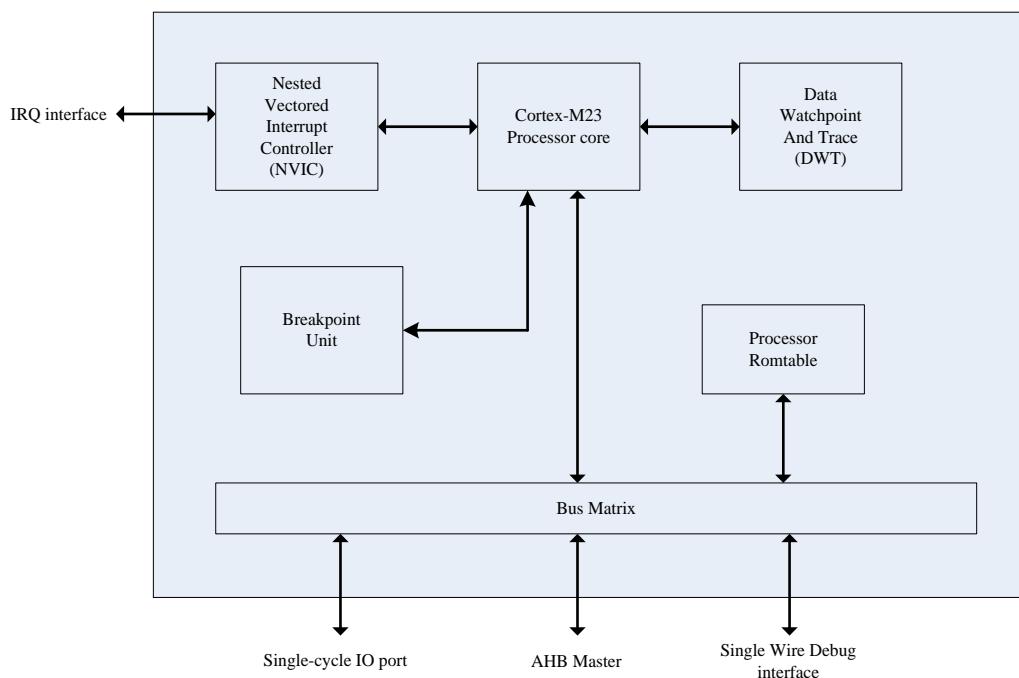
Cortex®-M23处理器高度集成了一个可配置的嵌套矢量中断控制器（NVIC），以提供业界领先的中断性能。

下面列出由Cortex®-M23提供的一些系统外设：

- 低延迟，高速外设I/O端口；
- 向量表偏移寄存器；
- 断点单元；
- 数据观测点；
- 串行调试接口。

下图显示了Cortex®-M23处理器结构框图。欲了解更多信息，请参阅Arm® Cortex®-M23技术参考手册。

图 1-1. Cortex®-M23 处理器结构框图

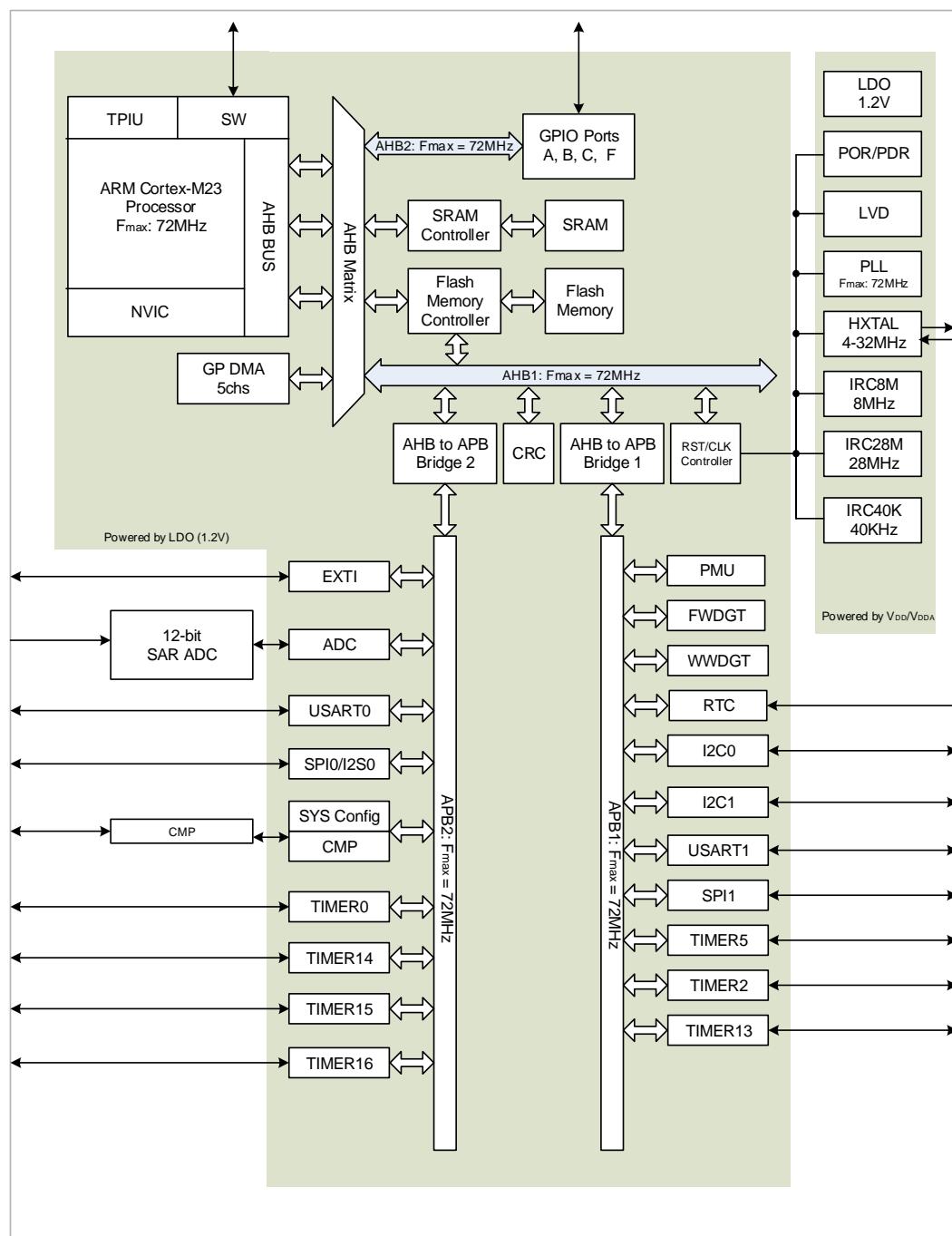


## 1.2. 系统架构

GD32E23x 系列器件的系统架构如下图所示。该 AHB 矩阵是一个基于 AMBA 5.0 AHB-LITE 的多层总线，这个结构使得系统中的多个主机和从机之间的并行通信成为可能。该 AHB 矩阵中包含属于 Cortex®-M23 内核的 AHB 总线，以及内核外的 DMA 共 2 个主机。该 AHB 矩阵还连接了 4 个从机，分别为：FMC、内部 SRAM、AHB1 和 AHB2。

AHB2 连接 GPIO 端口。AHB1 连接 AHB 外设，包括 2 个 AHB-APB 总线桥。AHB-APB 总线桥提供了 AHB1 和两条 APB 总线之间的全同步连接。两条 APB 总线连接了所有的 APB 外设。

图 1-2. GD32E23x 系列器件的系统架构示意图



### 1.3. 存储器映射

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®-M23 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®-M23 处理器提供的规则预先定义的。同时，一部分地址空间由 Arm® Cortex®-M23 的系统外设所占用。下表显示了 GD32E23x 系列器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。几乎每个外设都分配了 1KB 的地址空间，这样可以简化每个外设的地址译码。

表 1-1. GD32E23x 系列器件的存储器映射表

预先定义的地址空间	总线	地址范围	外设
		0xE000 0000 - 0xE00F FFFF	Cortex M23 内部外设
外部设备		0xA000 0000 - 0xDFFF FFFF	保留
外部 RAM		0x60000000 - 0x9FFFFFFF	保留
外设	AHB1	0x5004 0000 - 0x5FFF FFFF	保留
		0x5000 0000 - 0x5003 FFFF	保留
	AHB2	0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	保留
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4002 4400 - 0x47FF FFFF	保留
		0x4002 4000 - 0x4002 43FF	保留
		0x4002 3400 - 0x4002 3FFF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0400 - 0x4002 0FFF	保留
		0x4002 0000 - 0x4002 03FF	DMA
	APB2	0x4001 8000 - 0x4001 FFFF	保留
		0x4001 5C00 - 0x4001 7FFF	保留
		0x4001 5800 - 0x4001 5BFF	DBG
		0x4001 4C00 - 0x4001 57FF	保留
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 4000 - 0x4001 43FF	TIMER14
		0x4001 3C00 - 0x4001 3FFF	保留
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	保留
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	保留
		0x4001 2400 - 0x4001 27FF	ADC
		0x4001 0800 - 0x4001 23FF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG + CMP

预先定义的地址空间	总线	地址范围	外设
APB1	APB1	0x4000 CC00 - 0x4000 FFFF	保留
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	保留
		0x4000 8000 - 0x4000 BFFF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	保留
		0x4000 7400 - 0x4000 77FF	保留
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6400 - 0x4000 6FFF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4800 - 0x4000 53FF	保留
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	保留
		0x4000 3800 - 0x4000 3BFF	SPI1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 1400 - 0x4000 1FFF	保留
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0800 - 0x4000 0FFF	保留
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	保留
SRAM		0x2000 2000 - 0x3FFF FFFF	保留
		0x2000 0000 - 0x2000 1FFF	SRAM
代码		0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option bytes
		0x1FFF EC00 - 0x1FFF F7FF	System memory
		0x0801 0000 - 0x1FFF EBFF	保留
		0x0800 0000 - 0x0800 FFFF	Main Flash memory
		0x0001 0000 - 0x07FF FFFF	保留
		0x00000000 - 0x0000FFFF	Aliased to Flash or system memory

### 1.3.1. 片上 SRAM 存储器

GD32E23x系列微控制器含有高达8KB的片上SRAM，起始地址为0x2000 0000，支持字节、半字(16比特)和整字(32比特)访问。存储器支持奇偶校验来提高鲁棒性。用户可以通过用户选项字节(请参考第2.3.9章节[选项字节编程](#))的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\_CFG2)之中。如果系统配置寄存器2(SYSCFG\_CFG2)的SRAM\_PARITY\_ERROR\_LOCK位置1，错误标志将被连接到定时器0/定时器14/定时器15/定时器16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。

### 1.3.2. 片上闪存

该系列微控制器提供高达64KB的片上闪存。片上闪存包括高达64KB的主存储块和3KB容量的用于存储引导装载程序(*boot loader*)的信息块。主存储块分为64页，每页的容量为1KB。下表显示详细信息。

**表 1-2. Flash 模块组织**

闪存块	名称	地址范围	大小
主存储块	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	0x0801 FC00 - 0x0801 FFFF	1 Kbytes
信息块	系统存储器	0x1FFF EC00 - 0x1FFF F7FF	3 Kbytes
	选项字节	0x1FFF F800 - 0x1FFF F80F	16 bytes

读访问支持字节、半字(16比特)和整字(32比特)；写访问(编程)只支持字(32比特)和双字(64比特)。片上闪存的每一页都可以单独被擦除，整个主存模块(除信息块)也可以同时被擦除。

## 1.4. 引导配置

GD32E23x 系列微控制器提供了三种引导源，可以通过用户选项字节 BOOT1\_n 位(请参考第2.3.9章节[选项字节编程](#))和 BOOT0 引脚进行选择。BOOT0 引脚的电平值是在复位后第4个系统时钟上升沿锁存的。用户可自行选择所需要的引导源，通过设置上电复位或系统复位后的

BOOT1\_n 位和 BOOT0 的引脚电平值。下表描述了详细的引导模式信息。

表 1-3. 引导模式

引导源选择	启动模式选择引脚	
	Boot1	Boot0
主FLASH存储器	x	0
系统存储器	0	1
片上SRAM	1	1

#### 1. Boot1值与BOOT1\_n值相反

上电序列或系统复位后，Arm® Cortex®-M23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址，然后从引导代码的基址开始执行程序。

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。

芯片内嵌的引导装载程序位于系统存储器中，用来对片上闪存的主存进行重编程。该引导装载程序可通过以下串行接口之一工作：USART0或USART1。

## 1.5. 系统配置寄存器 (SYSCFG)

SYSCFG 基地址: 0x4001 0000

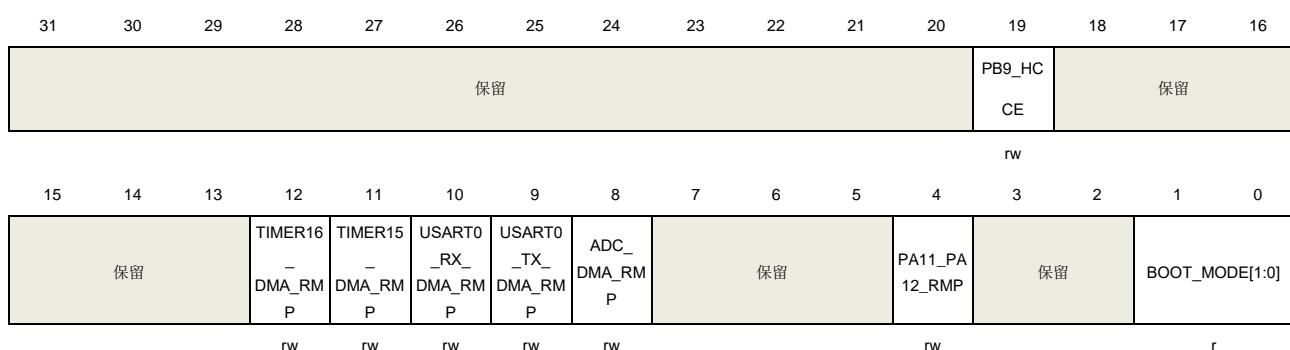
### 1.5.1. 配置寄存器 0 (SYSCFG\_CFG0)

#### GD32E230xx 产品

地址偏移: 0x00

复位值: 0x0000 000X (根据BOOT0引脚的状态和用户选项字节的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



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	TIMER16_DMA_RM	TIMER 16 DMA请求重映射使能 P 0: 不重映射 (TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道0) 1: 重映射 (TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道1)
11	TIMER15_DMA_RM	TIMER 15 DMA请求重映射使能 P 0: 不重映射 (TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道2) 1: 重映射 (TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道3)
10	USART0_RX_DMA_RMP	USART0_RX DMA请求重映射使能 0: 不重映射 (USART0_RX DMA被映射在DMA通道2) 1: 重映射 (USART0_RX DMA被映射在DMA通道4)
9	USART0_TX_DMA	USART0_TX DMA请求重映射使能

	RMP	0: 不重映射 (USART0_TX DMA被映射在DMA通道1) 1: 重映射 (USART0_TX DMA被映射在DMA通道3)
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0) 1: 重映射 (ADC DMA被映射在DMA通道1)
7:5	保留	必须保持复位值
4	PA11_PA12_RMP	小封装PA11/PA12重映射位控制位 (28和20pin封装) 该位由软件设置和清除。它控制PA9/10或PA11/12针对在小引脚数封装上的映射。 0: 不重映射 (PA9/10映射在引脚上) 1: 重映射 (PA11/12映射而不是PA9/10)
3:2	保留	必须保持复位值
1:0	BOOT_MODE[1:0]	引导模式(详细请参考第1.4章节 <a href="#">引导配置</a> ) bit0 映射到BOOT0引脚; bit1 的值与BOOT1_n的值相反 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

### GD32E231xx 产品

地址偏移: 0x00

复位值: 0x0000 000X (根据BOOT0引脚的状态和用户选项字节的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														保留	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIMER16 - DMA_RM P	TIMER15 - DMA_RM P	USART0 _RX_ DMA_RM P	USART0 _TX_ DMA_RM P	ADC_ DMA_RM P	保留	保留	PA11_PA 12_RMP	保留	保留	保留	保留	保留	保留	r

位/位域	名称	描述
31:20	保留	必须保持复位值
19	保留	必须保持复位值
18:13	保留	必须保持复位值
12	TIMER16_DMA_RM	TIMER 16 DMA请求重映射使能 P 0: 不重映射 (TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道0) 1: 重映射 (TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道1)

11	TIMER15_DMA_RM	TIMER 15 DMA请求重映射使能
P		0: 不重映射 (TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道2) 1: 重映射 (TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道3)
10	USART0_RX_DMA_	USART0_RX DMA请求重映射使能
RMP		0: 不重映射 (USART0_RX DMA被映射在DMA通道2) 1: 重映射 (USART0_RX DMA被映射在DMA通道4)
9	USART0_TX_DMA_	USART0_TX DMA请求重映射使能
RMP		0: 不重映射 (USART0_TX DMA被映射在DMA通道1) 1: 重映射 (USART0_TX DMA被映射在DMA通道3)
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0) 1: 重映射 (ADC DMA被映射在DMA通道1)
7:5	保留	必须保持复位值
4	PA11_PA12_RMP	小封装PA11/PA12重映射位控制位 (28和20pin封装) 该位由软件设置和清除。它控制PA9/10或PA11/12针对在小引脚数封装上的映射。 0: 不重映射 (PA9/10映射在引脚上) 1: 重映射 (PA11/12映射而不是PA9/10)
3:2	保留	必须保持复位值
1:0	BOOT_MODE[1:0]	引导模式(详细请参考第1.4章节 <a href="#">引导配置</a> ) bit0 映射到BOOT0引脚; bit1 的值与BOOT1_n的值相反 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. EXTI 源选择寄存器 0 (SYSCFG\_EXTISS0)

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: 保留 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: 保留 X011: 保留 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: 保留 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: 保留 X011: 保留 X100: 保留 X101: PF0引脚 X110: 保留 X111: 保留

### 1.5.3. EXTI 源选择寄存器 1 (SYSCFG\_EXTISS1)

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: 保留 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: 保留 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: 保留 X011: 保留 X100: 保留 X101: 保留 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: 保留  
 X011: 保留  
 X100: 保留  
 X101: 保留  
 X110: 保留  
 X111: 保留

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

##### **GD32E230xx 产品**

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: 保留 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: 保留 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: 保留
		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: 保留
		X011: 保留
		X100: 保留
		X101: 保留
		X110: 保留
		X111: 保留

### GD32E231xx 产品

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: 保留 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: 保留
		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: 保留
		X010: 保留
		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: 保留
		X011: 保留
		X100: 保留
		X101: 保留
		X110: 保留
		X111: 保留

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

#### GD32E230xx 产品

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]
rw	rw	rw	rw

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 X010: 保留 X011: 保留 X100: 保留 X101: 保留 X110: 保留

X111: 保留

### GD32E231xx 产品

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]						

rw rw rw rw

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: 保留 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:	保留
	X011:	保留
	X100:	保留
	X101:	保留
	X110:	保留
	X111:	保留

### 1.5.6. 系统配置寄存器 2 (SYSCFG\_CFG2)

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				SRAM_P CEF	保留				LVD_LOCK	SRAM_PARITY_ERROR_LOCK	LOCK_UP_LOCK				

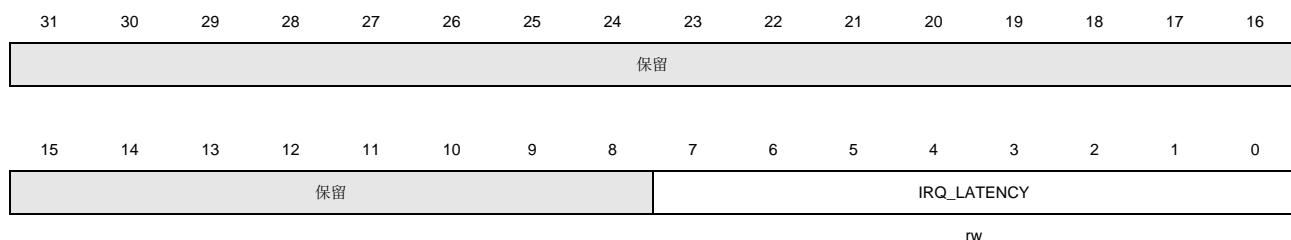
位/位域	名称	描述
31:9	保留	必须保持复位值
8	SRAM_PCEF	<p>SRAM奇偶校验错误标志</p> <p>当SRAM奇偶校验错误发生时，该位由硬件置1。该位由软件写1清零。</p> <p>0: 未检测到SRAM奇偶校验错误</p> <p>1: 检测到SRAM奇偶校验错误</p>
7:3	保留	必须保持复位值
2	LVD_LOCK	<p>LVD锁定</p> <p>该位由软件置1，在系统复位时才能清零。</p> <p>0: LVD中断从TIMER0/14/15/16的break输入端断开。<b>PMU_CTL</b>寄存器的LVDEN和LVDT[2:0]可以被设置</p> <p>1: LVD中断与TIMER0/14/15/16的break输入端连接。<b>PMU_CTL</b>寄存器的LVDEN和LVDT[2:0]仅仅可读</p>
1	SRAM_PARITY	SRAM奇偶校验错误锁定

	<b>ERROR_LOCK</b>	该位由软件置1，在系统复位时才能清零。 0: SRAM奇偶校验错误从TIMER0/14/15/16的break输入端断开 1: SRAM奇偶校验错误与TIMER0/14/15/16的break输入端连接
0	<b>LOCKUP_LOCK</b>	Cortex-M23 LOCKUP输出锁定 该位由软件置1，在系统复位时才能清零。 0: Cortex-M23 LOCKUP输出从TIMER0/14/15/16的break输入端断开 1: Cortex-M23 LOCKUP输出与TIMER0/14/15/16的break输入端连接

### 1.5.7. IRQ 延迟寄存器 (**SYSCFG\_CPU\_IRQ\_LAT**)

地址偏移: 0x100

复位值: 0x0000 0000



位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	IRQ_LATENCY[7:0]	<p>IRQ_LATENCY 指定在 NVIC 中挂起的中断与在 AHB-Lite 接口上发出的该中断的向量读取之间的最小周期数。</p> <p>如果将 IRQ_LATENCY 延迟设置为 0，则尽可能快地中断中断。</p> <p>对于非零值，Cortex-M23 处理器确保在 NVIC 中的中断和正在执行的中断的向量提取之间存在最小的 IRQ_LATENCY + 1 个 hclk 周期。</p>

## 1.6. 设备电子签名

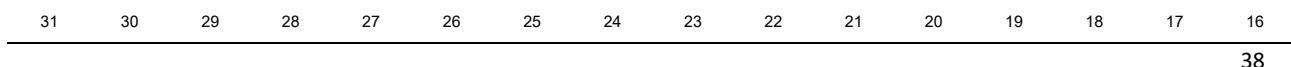
设备的电子签名中包含的存储容量信息和96位的唯一设备ID。它被存储在片上闪存的信息模块中。96位唯一设备ID对于每颗芯片而言都是唯一的。它可以用作序列号，或安全密钥的一部分，等等。

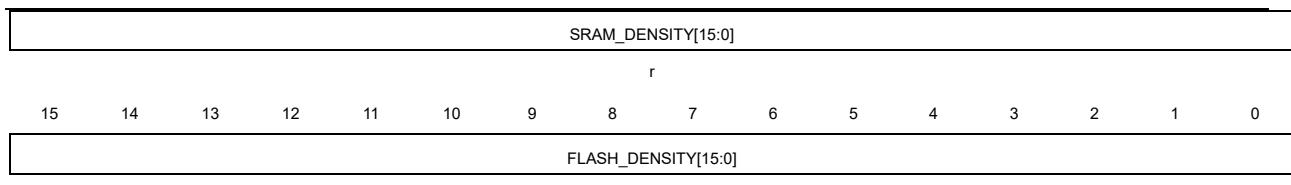
### 1.6.1. 存储容量信息

基地址: 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





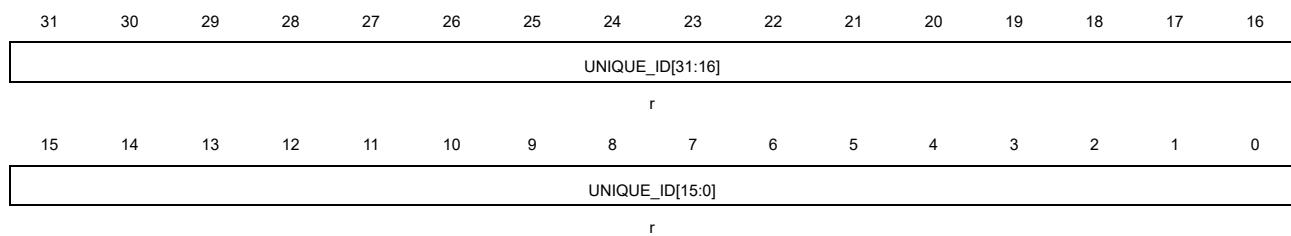
位/位域	名称	描述
31:16	SRAM_DENSITY	SRAM存储器容量 [15:0] 该值表明芯片的片上SRAM存储器容量，以Kbytes为单位 例如：0x0008表示8Kbytes。
15:0	FLASH_DENSITY	Flash存储器容量 [15:0] 该值表明芯片的片上Flash容量，以Kbytes为单位 例如：0x0020表示32Kbytes。

### 1.6.2. 设备唯一 ID (96 位/位域)

基地址：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

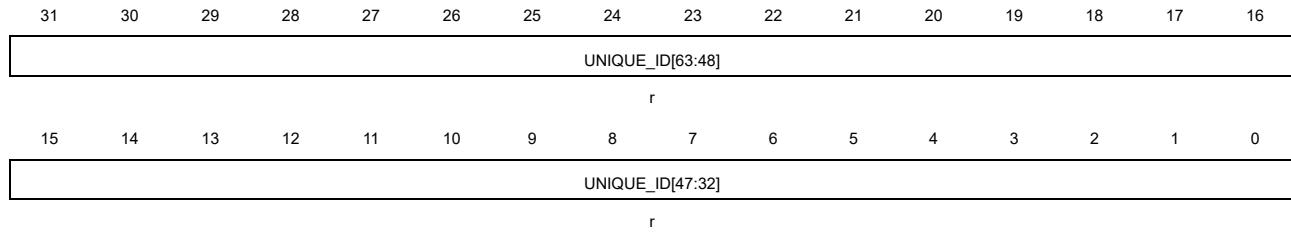


位/位域	名称	描述
31:0	UNIQUE_ID[31:0]	设备唯一 ID

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

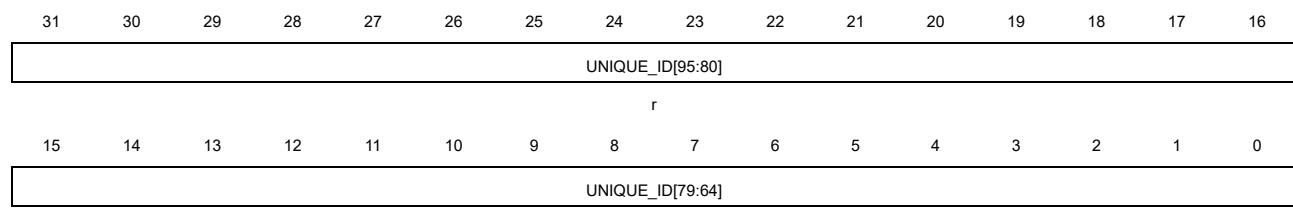


位/位域	名称	描述
31:0	UNIQUE_ID[63:32]	设备唯一 ID

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. 闪存控制器 (FMC)

### 2.1. 简介

闪存控制器 (FMC)，提供了片上闪存需要的所有功能。从闪存的64K字节空间开始，CPU执行指令需要一点等待时间。FMC也提供了页擦除、整片擦除以及32位整字或64位双字编程等操作。

### 2.2. 主要特性

对于 GD32E23x系列：

- 高达64KB字节的片上闪存可用于存储指令或数据；
- 在闪存的前64K字节空间内，CPU执行指令需0~2个等待时间；
- 预取缓冲区用于提高读操作速度；
- 3K字节引导装载程序信息块；
- 16字节的选项字节块用于用户应用需求；
- 每页大小为1K字节；
- 32位字或64位双字编程，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；

### 2.3. 功能描述

#### 2.3.1. 闪存结构

闪存存储器包括一个高达64KB字节的主闪存(按64页每页1KB字节分块)和一个3KB字节的用于引导装载程序的信息块。主闪存存储器的64页每页都可以单独擦除。闪存存储器的基地址和大小见表 2-1. 闪存的基址和大小。

表 2-1. 闪存的基址和大小

闪存块	名称	地址范围	大小(字节)
主闪存块	Page 0	0x0800 0000 - 0x0800 03FF	1KB
	Page 1	0x0800 0400 - 0x0800 07FF	1KB
	Page 2	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	Page 63	0x0800 FC00 - 0x0800 FFFF	1KB
信息块	引导装载程序	0x1FFF EC00 - 0x1FFF F7FF	3KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B

一次性编程块	OTP字节	0x1FFF_7000~0x1FFF_73FF	1KB
--------	-------	-------------------------	-----

**注意:** 信息块存储了引导装载程序(boot loader)，不能被用户编程或擦除。

### 2.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据可以使用CPU的AHB BUS总线。

**等待状态添加:**

根据AHB时钟频率正确配置FMC\_WS寄存器中WSCNT位。WSCNT位域与AHB时钟频率之间的关系如[表 2-2. WSCNT位域与AHB时钟频率之间的关系](#)所示。

**表 2-2. WSCNT位域与AHB时钟频率之间的关系**

AHB时钟频率	WSCNT配置
<= 24MHz	0 (添加0个等待状态)
<= 48MHz	1 (添加1个等待状态)
<= 72MHz	2 (添加2个等待状态)

如果发生系统复位，AHB时钟频率为8MHz，WSCNT的值为0。

**注意:**

1. 如果需要提高AHB的时钟频率，首先，参考[表 2-2. WSCNT位域与AHB时钟频率之间的关系](#)，根据目标AHB时钟频率配置WSCNT位域。然后将AHB时钟频率提高到目标时钟频率。在配置WSCNT位域之前提高AHB时钟频率是不可行的。
2. 如果需要降低AHB的时钟频率，首先，降低目标AHB时钟频率，然后参考[表 2-2. WSCNT位域与AHB时钟频率之间的关系](#)，根据目标AHB时钟频率配置WSCNT位域。在降低AHB时钟频率之前配置WSCNT位域是不可行的。

由于添加了等待状态，读效率非常低（如72MHz时添加两个等待状态）。为了加速读操作，需要用到一些功能。

**当前缓冲区:**

当前缓冲区总是被使能的。每次对闪存存储器进行读操作，将获得一个64位的数据并存储在当前缓冲区中。而每一个读操作，CPU需要32位或16位数据。所以在连续代码的情况下，下一个数据能从当前缓冲区中获取到而不需要重复地从存储器获取。

**预取缓冲区**

当FMC\_WS寄存器中PFEN位被置位时预取缓冲区被使能。在连续代码的情况下，当CPU处理当前缓冲区中数据（64位）时，32位操作至少需要两个时钟，16位操作至少需要4个时钟。在这种情况下，在闪存存储器的下一个双字地址预取数据并存到预取缓冲区。当CPU完成当前缓冲区并需要处理下一个数据，预取缓冲区被命中。

### 2.3.3. FMC\_CTL 寄存器解锁

复位后，FMC\_CTL寄存器不可以用写模式进行访问（OBRLD位除外，此位用于重加载选项字节）并且FMC\_CTL寄存器中的LK位被置为1。一个包含对FMC\_KEY寄存器进行两次写操作的解锁序列可以解锁FMC\_CTL寄存器的访问，分别是先后写入0x45670123和0xCDEF89AB。两次写操作后，FMC\_CTL寄存器的LK位将被硬件清0。可以通过软件设置FMC\_CTL寄存器的LK位为1再次锁定FMC\_CTL寄存器。任何对FMC\_KEY寄存器的错误操作都会将LK位置1，从而锁定FMC\_CTL寄存器，并且引发一个总线错误。

FMC\_CTL寄存器的OBPG位和OBER位可以被FMC\_OBKEY寄存器锁定。解锁序列是向FMC\_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后将FMC\_CTL寄存器的OBWEN位置1。软件可以将FMC\_CTL的OBWEN位清0来锁定FMC\_CTL的OBPG位和OBER位。

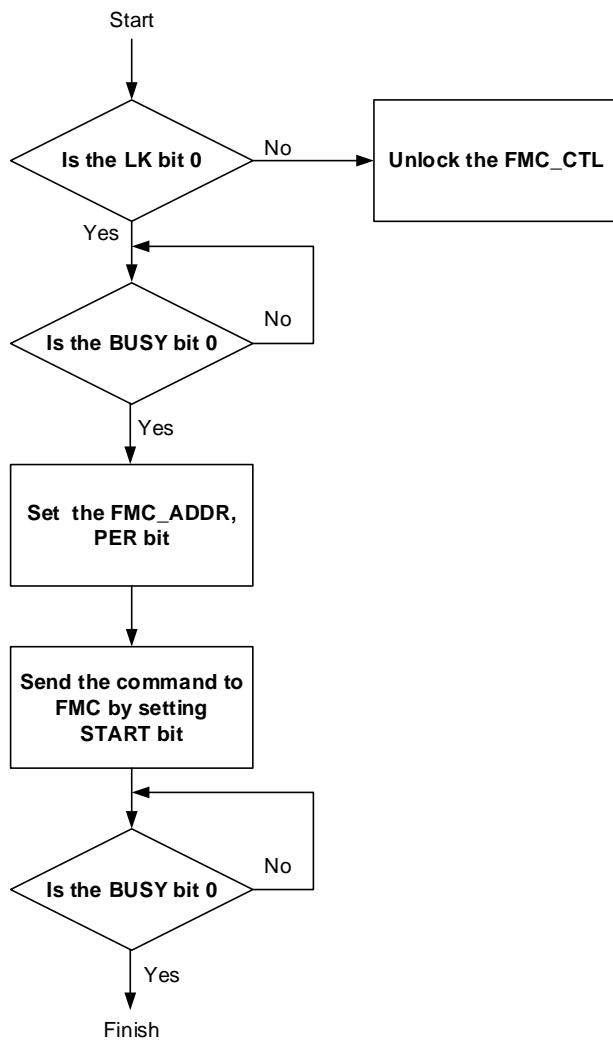
### 2.3.4. 页擦除

FMC的页擦除功能将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，不影响其他页的内容。FMC页擦除的操作步骤如下：

- 确保FMC\_CTL寄存器不处于锁定状态；
- 检查FMC\_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写页地址到FMC\_ADDR寄存器；
- 写页擦除命令到FMC\_CTL寄存器的PER位；
- 通过将FMC\_CTL寄存器的START位置1来发送页擦除命令到FMC；
- 通过检查FMC\_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读来验证该页的内容。

当页擦除成功执行，FMC\_STAT寄存器的ENDF位将会被置1，并且如果FMC\_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。需要注意的是，一定要确保目标页地址的正确性。否则当错误的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。另一方面，在擦除/编程保护的页进行页擦除操作将会无效。如果FMC\_CTL寄存器的ERRIE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\_STAT寄存器的PGERR位来监测中断处理器的状况。[图 2-1. 页擦除操作流程](#)显示了页擦除操作的流程：

图 2-1. 页擦除操作流程



### 2.3.5. 整片擦除

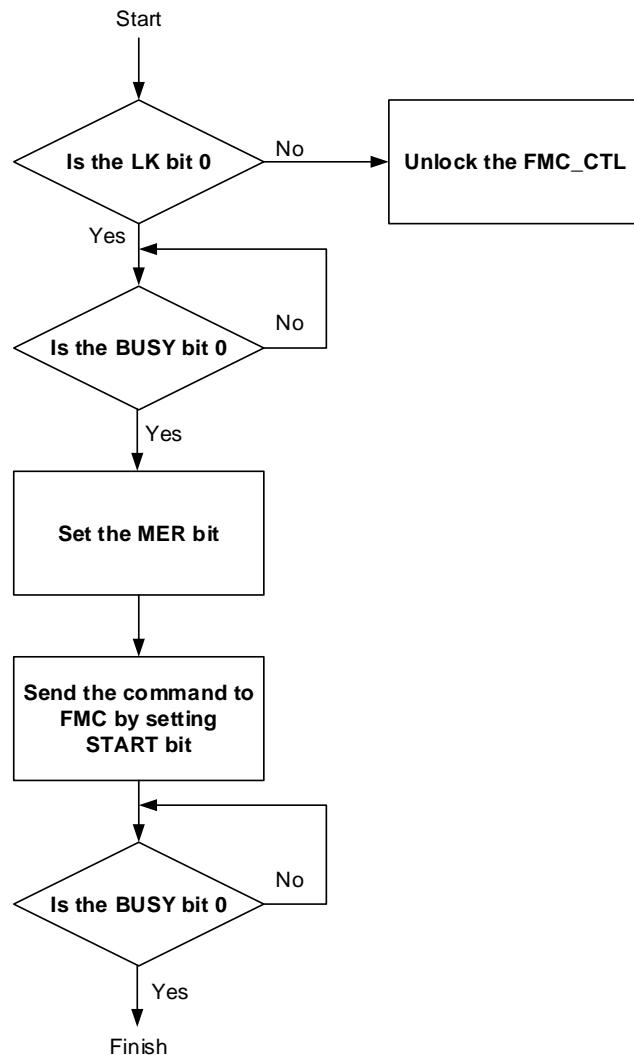
FMC提供了整片擦除功能可以初始化主闪存块的内容。整片擦除操作的寄存器设置具体步骤如下：

- 确保FMC\_CTL寄存器不处于锁定状态；
- 检查FMC\_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写整片擦除命令到FMC\_CTL寄存器的MER位（置1）；
- 通过将FMC\_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 通过检查FMC\_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读和验证该闪存的内容。

当整片擦除成功执行，FMC\_STAT寄存器的ENDF位将会被置1，并且如果FMC\_CTL寄存器的ENDIE位之前已经被置1，那么FMC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\_STAT寄存器的ENDF位指示该操作的结束。（编程操作的起始地址需要是0x0800 0000）。[图 2-2. 整片擦除操作流程](#)显示了整片擦除操作的流程：

[图 2-2. 整片擦除操作流程](#)



### 2.3.6. 主存储闪存块编程

FMC提供了一个DBUS 32位整字/16位半字编程功能，用来修改主闪存存储器的内容。然而实际上，是由FMC\_WS寄存器中PGW位来设置32位或者64位数据被编程到闪存存储器。以下步骤显示了编程操作寄存器的过程：

- 确保FMC\_CTL寄存器不处于锁定状态；
- 检查FMC\_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 如果有需要，设置PGW位；
- 写字编程命令到FMC\_CTL寄存器的PG位；
- DBUS写一个数据到预期绝对地址（0x08XX XXXX）；

如果DBUS写的是一个32位整字并且PGW位被设置成0（32位编程闪存存储器），DBUS写一次数据被编程到闪存存储器。在编程数据时必须是字对齐。

如果DBUS写的是一个32位整字并且PGW位被设置成1（64位编程闪存存储器），DBUS写两次组成一个64位数据，然后再编程到闪存存储器。在编程数据时必须是双字对齐。

如果DBUS写的是一个16位半字并且PGW位被设置成0（32位编程闪存存储器），DBUS写两次组成一个32位数据，然后再编程到闪存存储器。在编程数据时必须是字对齐。

如果DBUS写的是一个16位半字并且PGW位被设置成1（64位编程闪存存储器），DBUS写四次组成一个64位数据，然后再编程到闪存存储器。在编程数据时必须是双字对齐。

为了减少编程时间，假设采用DBUS 32位编程，如果被编程的数据是双字对齐则将PGW位置1，如果被编程的数据是字对齐则将PGW位清零。

- 通过检查FMC\_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读并验证该闪存存储器是否编程成功。

当主存储块编程成功执行，FMC\_STAT 寄存器的 ENDF 位将会被置 1，并且如果 FMC\_CTL 寄存器的 ENDIE 位之前已经被置 1，那么 FMC 将触发一个中断。需要注意的是在执行字/半字编程操作之前需要检查目的地址是否已经被擦除过，如果没有被擦除过，PGERR 位将被置 1，该页上的编程操作无效即使编程内容为 0x0。每个字在擦除后和下次擦除前只能编程一次。

另一方面，在擦除/编程保护的页进行编程操作将会无效。如果 FMC\_CTL 寄存器的 ERRIE 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\_STAT 寄存器的 PGERR 位来监测中断处理器的状况。FMC\_STAT 寄存器的 ENDF 位指示操作的结束。

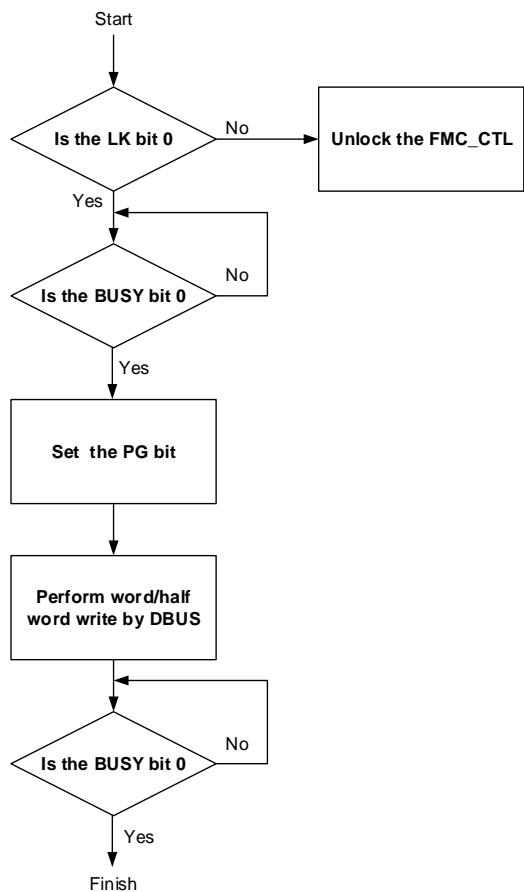
在以下情况中，FMC\_STAT寄存器的PGAERR位将被置位。

- DBUS采用字节编程的方式（而非32位整字或者16位半字编程）。
- DBUS编程宽度与之前不一致。不允许32位和16位编程混合使用。
- DBUS编程不对齐。如果采用DBUS 32位编程并且PGW被置1(即对闪存存储器64位编程)，DBUS第二次编程必须是双字对齐并且属于同一个双字地址。如果采用DBUS 16位编程并且PGW被置0(即对闪存存储器32位编程)，DBUS第二次编程必须是整字对齐并且属于同一个字节地址。如果采用DBUS 16位编程并且PGW被置1(即对闪存存储器64位编程)，第二/三/四次编程必须是双字对齐并且属于同一个双字地址。

**注意：**如果没有完整写完64位或32位（配置FMC\_WS寄存器中PGW位选择编程宽度），并没有任何提示这个数据没有写进闪存存储器。

在这些情况下，如果FMC\_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。软件可以通过检查FMC\_STAT寄存器中PGERR，PGAERR或者WPERR位来检测是哪个条件触发了中断。[图 2-3. 字编程操作流程](#)显示了主存储块字编程操作的流程：

图 2-3. 字编程操作流程



### 2.3.7. OTP 编程

OTP编程方法与主储存闪存编程相同。OTP块只能被编程一次并且不能被擦除。

**注意：**必须确保在OTP编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生意外中断，会有极低的概率使存储在闪存中的数据发生破坏。

### 2.3.8. 选项字节擦除

FMC提供了一个擦除功能用来初始化闪存中的选项字节块。下面的步骤显示了选项字节块的擦除过程：

- 确保FMC\_CTL寄存器不处于锁定状态；
- 确保FMC\_CTL寄存器的OBWEN位处于使能状态；
- 检查FMC\_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写选项字节擦除命令到FMC\_CTL寄存器的OBER位（置1）；
- 通过将FMC\_CTL寄存器的START位置1来发送选项字节擦除命令到FMC；
- 通过检查FMC\_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；

- 如果有需要，可以使用**DBUS**读并验证是否擦除成功。

当选项字节擦除成功执行，**FMC\_STAT**寄存器的**ENDF**位将会被置1，并且如果**FMC\_CTL**寄存器的**ENDIE**位之前已经被置1，那么**FMC**将触发一个中断。**FMC\_STAT**寄存器的**ENDF**位置1指示该操作的结束。

### 2.3.9. 选项字节编程

**FMC**提供了一个32位字/64位双字编程功能，用来修改选项字节块的内容。下面的步骤显示了选项字节编程的操作过程：

- 确保**FMC\_CTL**寄存器不处于锁定状态；
- 确保**FMC\_CTL**寄存器的**OBWEN**位处于使能状态；
- 检查**FMC\_STAT**寄存器的**BUSY**位来确保闪存存储器没有正在进行中的操作，即**BUSY**位为0。否则等待该操作完成；
- 写选项字节编程命令到**FMC\_CTL**寄存器的**OBPG**位；
- **DBUS**写一个32位字/16位半字到预期地址；
- 通过检查**FMC\_STAT**寄存器的**BUSY**位来判断擦除指令是否执行完毕，若未完成则需等待**BUSY**位为0；
- 如果有需要，可以使用**DBUS**读并验证是否编程成功。

当选项字节编程操作成功执行，**FMC\_STAT**寄存器的**ENDF**位将会被置1，并且如果**FMC\_CTL**寄存器的**ENDIE**位之前已经被置1，那么**FMC**将触发一个中断。需要注意的是在执行字/半字编程操作之前需要检查目的地址是否已经被擦除过，如果没有被擦除过，**PGERR**位将被置1，该页上的编程操作无效即使编程内容为0x0。**FMC\_STAT**寄存器的**ENDF**位指示操作的结束。

### 2.3.10. 选项字节说明

每次系统复位或将**FMC\_CTL**寄存器的**OBRLD**位置1时，闪存存储器的选项字节块会被重加载到**FMC\_OBSTAT**和**FMC\_WP**寄存器，之后选项字节生效。选项字节的补码和选项字节相反。当选项字节被重加载，如果选项字节的补码和选项字节不匹配，**FMC\_OBSTAT**寄存器的**OBERR**位将被置1，选项字节将被设置为0xFF。选项字节详情见[表 2-3. 选项字节](#)：

表 2-3. 选项字节

地址	名称	说明
0x1fff f800	OB_SPC	选项字节安全保护码 0xA5：无保护 除0xA5或0xCC之外的任意值：低级别保护 0xCC：高级别保护
0x1fff f801	OB_SPC_N	OB_SPC补码值
0x1fff f802	OB_USER	用户定义的选项字节 [7]: 保留 [6]: SRAM_PARITY_CHECK 0: 使能SRAM奇偶校验 1: 去使能SRAM奇偶校验

地址	名称	说明
		<p>[5]: VDDA_VISOR 0: 去使能V<sub>DDA</sub>监视器 1: 使能V<sub>DDA</sub>监视器</p> <p>[4]: BOOT1_n 0: BOOT1位是1 1: BOOT1位是0</p> <p>[3]: 保留</p> <p>[2]: nRST_STDBY 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是产生复位</p> <p>[1]: nRST_DPSLP 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</p> <p>[0]: nWDG_SW 0: 硬件自动设置独立看门狗定时器 1: 软件设置独立看门狗定时器</p>
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位
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位
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/编程保护位的15到8位
0x1fff f80b	OB_WP_N[15:8]	OB_WP补码值的15到8位

### 2.3.11. 页擦除/编程保护

FMC的页擦除/编程保护功能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\_STAT寄存器的WPERR位将被置1。如果WPERR位被置1且ERRIE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。配置选项字节的OB\_WP[15:0]位为0可以单独使能每个页的保护功能。如果在选项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。当对选项字节的OB\_WP位置1或清0时，需要将FMC\_CTL寄存器的OBRLD位置1或系统复位来重加载OB\_WP位。[表 2-4. OB\\_WP位对应页保护](#)显示了通过设置OB\_WP[15:0]寄存器可以保护哪些页：

**表 2-4. OB\_WP 位对应页保护**

OB_WP位	页保护
OB_WP[0]	页0 ~ 页3
OB_WP[1]	页4 ~ 页7
OB_WP[2]	页8 ~ 页11

OB_WP位	页保护
OB_WP[14]	页56 ~ 页59
OB_WP[15]	页60 ~ 页63

### 2.3.12. 安全保护

FMC提供了一个安全保护功能来阻止闪存上非法的代码或数据访问。此功能可以很好地保护软件和固件免受非法用户的操作。安全保护等级划分为以下三等：

无保护状态：当将OB\_SPC字节和它的补码值设置为0xA55A，则不执行保护。主存储块和选项字节块可以被所有操作访问。

保护等级低：当设置OB\_SPC字节和它的补码值为除了0xA55A或0xCC33之外的任意值，则执行低级别保护。主闪存仅仅能被用户代码访问。在调试模式下，从SRAM或boot loader模式启动时，所有对主存储块的操作都被禁止，并且如果是读操作将会产生一个总线错误，如果是编程或擦除操作将会导致FMC\_STAT寄存器的PGERR位被置1。在低级别保护状态下，选项字节块可以被任意操作访问。如果将OB\_SPC字节和它的补码值设置为0xA55A，进入无保护级别，那么将执行一次主存储块整片擦除操作。

保护等级高：当设置OB\_SPC字节和它的补码值为0xCC33，则执行高级别保护。在调试模式下，从SRAM或boot loader模式启动都将被禁止。主闪存块可由用户代码的所有操作进行访问。选项字节不能被擦除，OB\_SPC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，不能转换为低级别保护和无保护。

**注意：**在读保护情况下，flash中前4k不能被擦写。

## 2.4. FMC 寄存器

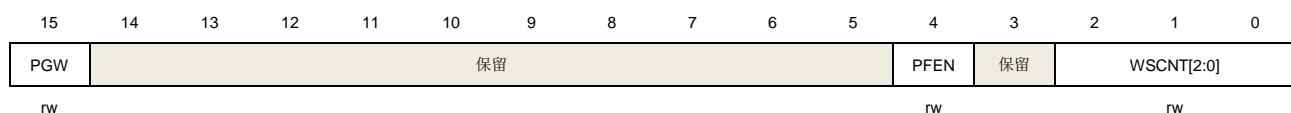
FMC基地址: 0x4002 2000

### 2.4.1. 等待状态寄存器 (FMC\_WS)

地址偏移: 0x00

复位值: 0x0000 0030

该寄存器只能按字(32位)访问



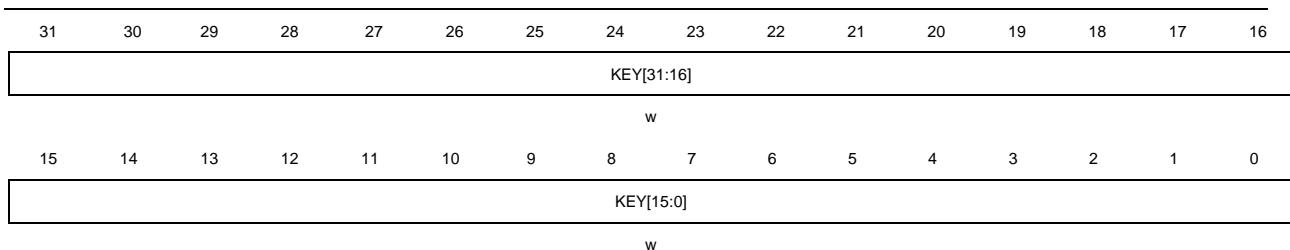
位/位域	名称	描述
31:16	保留	必须保持复位值
15	PGW	闪存存储器编程宽度 0: 闪存存储器32位编程 1: 闪存存储器64位编程
14:5	保留	必须保持复位值
4	PFEN	预取使能 0: 预取失能 1: 预取使能
3	保留	必须保持复位值
2:0	WSCNT[2:0]	等待状态计数寄存器 软件置1和清0。 000: 不增加等待状态 001: 增加1个等待状态 010: 增加2个等待状态 011 ~ 111: 保留

### 2.4.2. 解锁寄存器 (FMC\_KEY)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



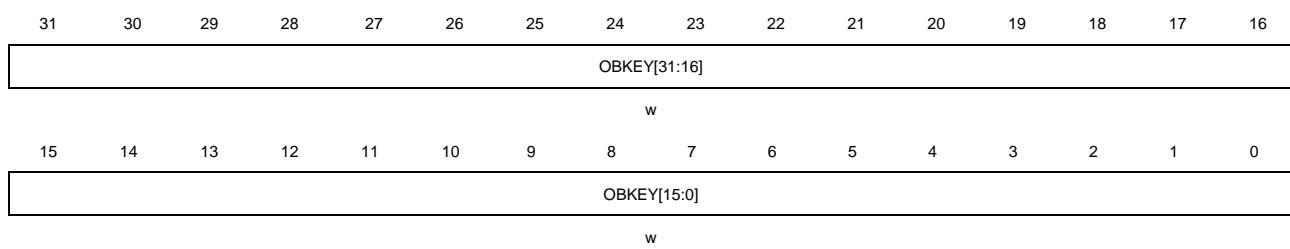
位/位域	名称	描述
31:0	KEY[31:0]	<b>FMC_CTL</b> 解锁寄存器 这些位仅能被软件写。 写解锁值到KEY[31:0]来解锁 <b>FMC_CTL</b> 寄存器。

#### 2.4.3. 选项字节解锁寄存器 (**FMC\_OBKEY**)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



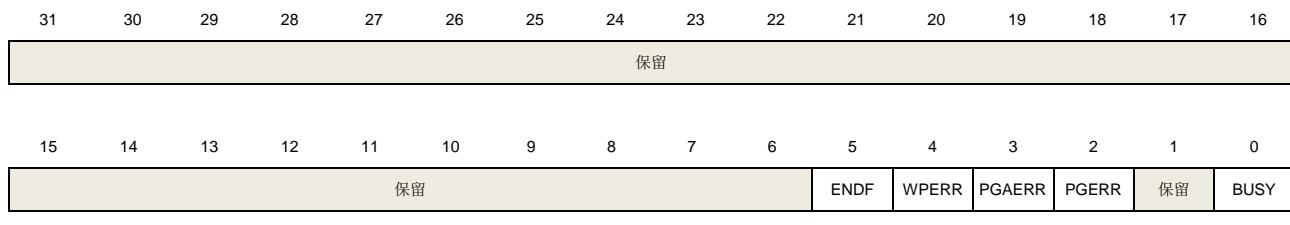
位/位域	名称	描述
31:0	OBKEY[31:0]	<b>FMC_CTL</b> 选项字节解锁寄存器 这些位仅能被软件写 写解锁值到OBKEY[31:0]来解锁 <b>FMC_CTL</b> 寄存器的选项字节命令

#### 2.4.4. 状态寄存器 (**FMC\_STAT**)

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
31:6	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。软件可以通过写1来清0该位。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上进行擦除/编程操作时，此位会被硬件置1。软件可以通过写1来清0该位。
3	PGAERR	编程对齐错误标志位 在DBUS写未对齐时由硬件置1。软件可以通过写1来清0该位。
2	PGERR	编程错误标志位 当在值不为0xFFFF的闪存上进行编程时，此位会被硬件置1。软件可以通过写1来清0该位。
1	保留	必须保持复位值
0	BUSY	闪存忙标志位 当有闪存操作正在进行时，此位被置1。当操作结束或者出错时，此位被清0。

#### 2.4.5. 控制寄存器 (FMC\_CTL)

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	OBRLD	ENDIE	保留	ERRIE	OBWEN	保留	LK	START	OBER	OBPG	保留	MER	PER	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	OBRLD	选项字节重加载位 软件置1。 0: 没有作用 1: 强制选项字节重装载，并产生一次系统复位
12	ENDIE	操作结束中断使能位 软件置1或清0。 0: 无硬件中断产生

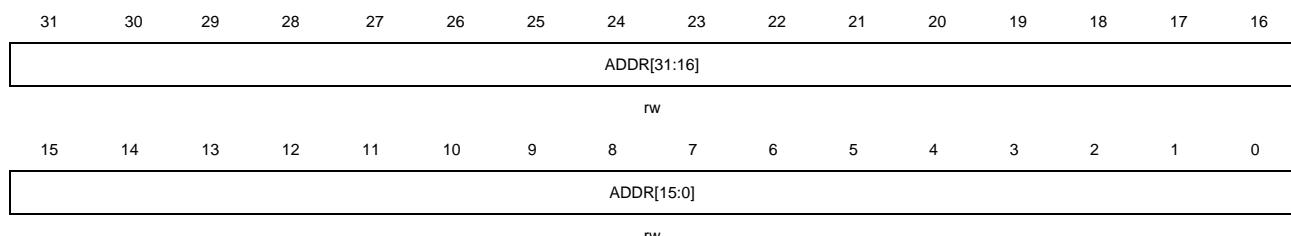
		1: 使能操作结束中断
11	保留	必须保持复位值
10	ERRIE	错误中断使能位 软件置1或清0。 0: 无硬件中断产生 1: 使能错误中断
9	OBWEN	选项字节擦除/编程使能位 当正确的序列写入FMC_OBKEY寄存器，此位由硬件置1。此位可以被软件清0。
8	保留	必须保持复位值
7	LK	锁FMC_CTL寄存器标志位 当正确的序列写入FMC_KEY寄存器，此位由硬件清0。此位可以由软件置1。
6	START	发送擦除命令到FMC位 软件置1后将发送擦除命令到FMC。当BUSY位被清0后，此位将被硬件清0。
5	OBER	选项字节擦除命令位 软件置1和清0。 0: 无作用 1: 选项字节擦除命令
4	OBPG	选项字节编程命令位 软件置1和清0。 0: 无作用 1: 选项字节编程命令
3	保留	必须保持复位值
2	MER	主存储块整片擦除命令位 软件置1和清0。 0: 无作用 1: 主存储块整片擦除命令
1	PER	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	主存储块编程命令位 软件置1和清0 0: 无作用 1: 主存储块编程命令

### 2.4.6. 地址寄存器(FMC\_ADDR)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



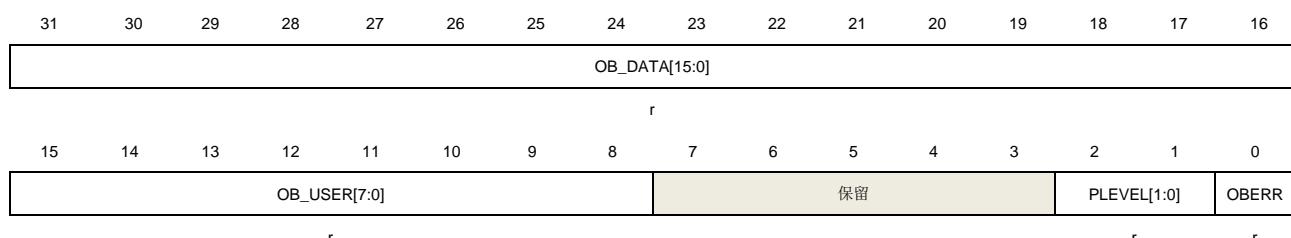
位/位域	名称	描述
31:0	ADDR[31:0]	闪存命令地址位 软件置位 ADDR位是闪存擦除命令的地址。

### 2.4.7. 选项字节状态寄存器 (FMC\_OBSTAT)

地址偏移: 0x1C

复位值: 0xXXXX XX0X

该寄存器只能按字(32位)访问



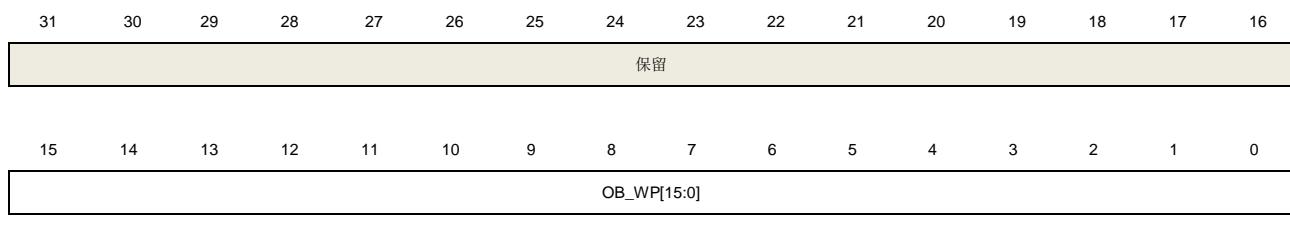
位/位域	名称	描述
31:16	OB_DATA[15:0]	系统复位后保存选项字节块的OB_DATA[15:0]部分
15:8	OB_USER[7:0]	系统复位后保存选项字节块的OB_USER字节
7:3	保留	必须保持复位值
2:1	PLEVEL[1:0]	安全保护级别 00: 无保护 01: 低保护级别 11: 高保护级别
0	OBERR	选项字节读错误位 当选项字节和它的补码不匹配时此位由硬件置1，并且选项字节被设置为0xFF。

### 2.4.8. 写保护寄存器 (FMC\_WP)

地址偏移: 0x20

复位值: 0x0000 XXXX

该寄存器只能按字(32位)访问



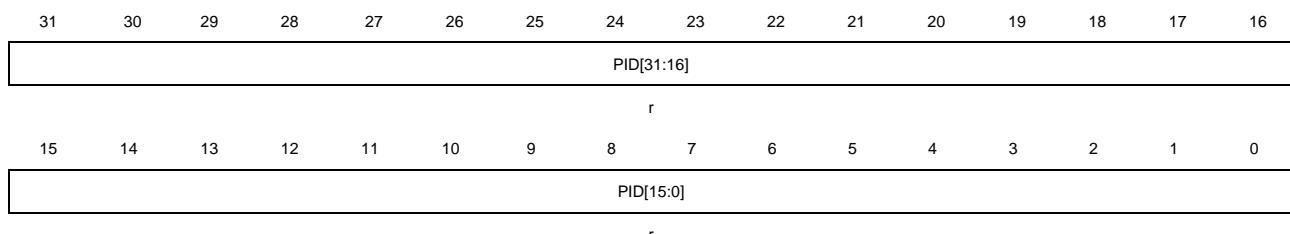
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	OB_WP[15:0]	系统复位后保存选项字节块的OB_WP[15:0]部分 0: 保护生效 1: 未保护

### 2.4.9. 产品 ID 寄存器 (FMC\_PID)

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	PID[31:0]	产品保留ID寄存器 该寄存器为只读 上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。

### 3. 电源管理单元 (PMU)

#### 3.1. 简介

功耗设计是 GD32E23x 系列产品比较注重的问题之一。电源管理单元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。这些模式能减少电源能耗，且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。如 [图3-1. 电源域概览](#) 所示，GD32E23x 系列设备有三个电源域，包括  $V_{DD}$  /  $V_{DDA}$  域，1.2V 域和备份域。 $V_{DD}$  /  $V_{DDA}$  域由电源直接供电。在  $V_{DD}$  /  $V_{DDA}$  域中嵌入了一个LDO，用来为1.2V域供电。备份域由  $V_{DD}$  电源供电。

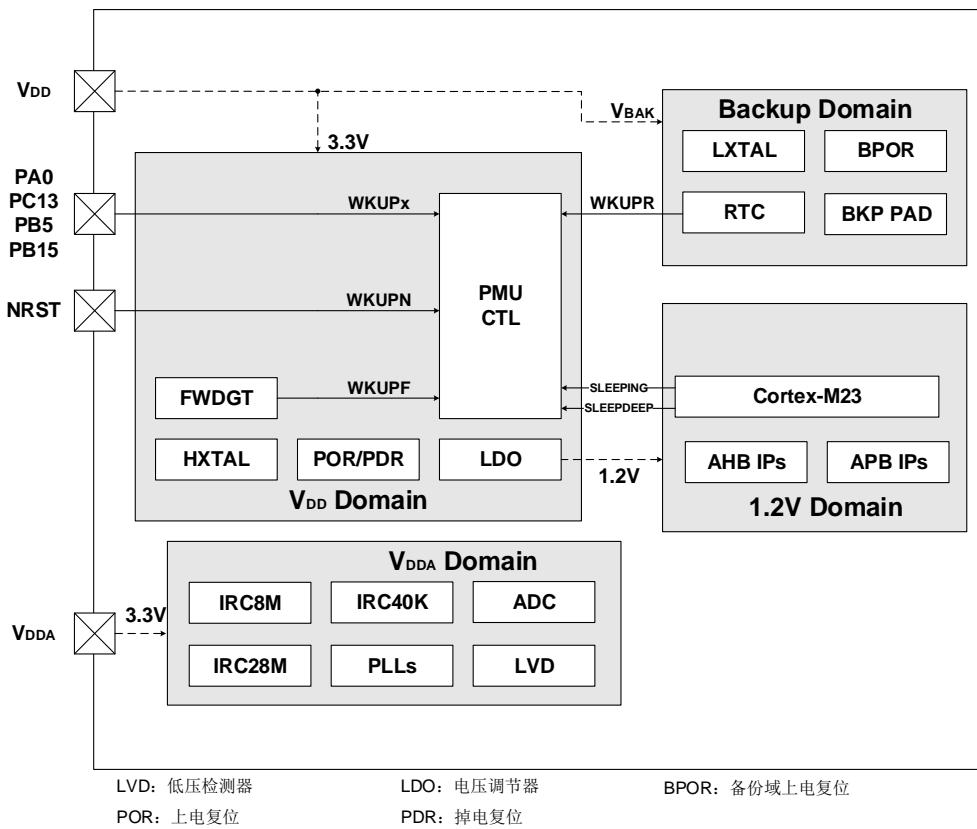
#### 3.2. 主要特征

- 三个电源域：备份域、 $V_{DD}$  /  $V_{DDA}$  域和1.2V电源域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部电压调节器（LDO）提供1.2V电源；
- 提供低电压检测器（LVD），当电压低于所设定的阈值时能发出中断或事件；
- LDO输出电压用于节约能耗；

#### 3.3. 功能说明

[图3-1. 电源域概览](#) 提供了PMU及相关电源域的内部结构框图。

图3-1. 电源域概览



注意: GD32E231系列, 无PC13脚和WKUP1功能。

### 3.3.1. 电池备份域

电池备份域由V<sub>DD</sub>供电, 然后由V<sub>BAK</sub>为备份域供电, 该备份域包含RTC(实时时钟)、LXTAL(低速外部晶体振荡器)、BPOR(备份域上电复位), 以及PC13至PC15共3个PAD(GD32E231系列无PC13脚)。

备份域的复位源包括备份域上电复位和备份域软件复位。在V<sub>BAK</sub>没有完全上电前, BPOR信号强制设备处于复位状态。应用软件可以通过设置RCU\_BDCTL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(IRC40K)或低速外部晶体振荡器(LXTAL), 或高速外部晶体振荡器(HXTAL)时钟32分频。在通过WFI/WFE指令进入省电模式之前, Cortex®-M23需要通过RTC寄存器设置预期的闹钟时间并启用闹钟功能, 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将唤醒设备。RTC的配置和操作的细节将在[实时时钟\(RTC\)](#)来描述。

当备份域由V<sub>DD</sub>供电(V<sub>BAK</sub>连接至V<sub>DD</sub>)时, 以下功能可用:

- PC13可以作为通用I/O口或RTC功能引脚(参见[实时时钟\(RTC\)](#));
- PC14和PC15可以作为通用I/O口或LXTAL晶振引脚。

注意: GD32E231系列无PC13脚。由于PC13至PC15引脚仅可通过小电流, 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(最大负载为30pF)。

### 3.3.2. $V_{DD} / V_{DDA}$ 电源域

$V_{DD} / V_{DDA}$  域包括  $V_{DD}$  域和  $V_{DDA}$  域两部分。 $V_{DD}$  域包括 HXTAL (高速外部晶体振荡器)、LDO (电压调节器)、POR / PDR (上电/掉电复位)、FWDGT (独立看门狗定时器) 和除 PC13、PC14 和 PC15 之外的所有 PAD 等等。 $V_{DDA}$  域包括 ADC (AD 转换器)、IRC8M (内部 8M RC 振荡器)、IRC28M (内部 28M RC 振荡器)、IRC40K (内部 40KHz RC 振荡器) PLLs (锁相环) 和 LVD (低电压检测器) 等等。

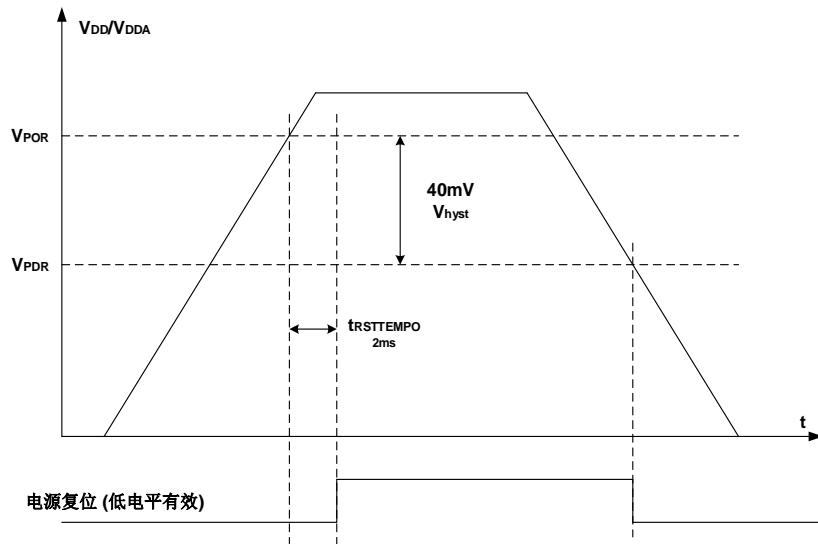
#### $V_{DD}$ 域

为 1.2V 域供电的 LDO (电压调节器)，其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式 (全供电状态)、深度睡眠模式 (全供电或低功耗状态) 和待机模式 (关闭状态)。

POR / PDR (上电/掉电复位) 电路检测  $V_{DD} / V_{DDA}$  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。POR 电路只检测  $V_{DD}$  供电电压，PDR 电路检测  $V_{DD}$  和  $V_{DDA}$  供电电压。[图3-2. 上电/掉电复位波形图](#)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，典型值约为 1.71V， $V_{PDR}$  表示掉电复位的阈值电压，典型值约为 1.67V。迟滞电压  $V_{hyst}$  值约为 40mV。

**注意：**如果用户能确保  $V_{DDA}$  电压大于等于  $V_{DD}$ ，可以通过清零选项字节寄存器 OB\_USER 的位 VDDA\_VISOR，关闭由 PDR 电路实现的  $V_{DDA}$  监视器以减小功耗。

**图3-2. 上电/掉电复位波形图**

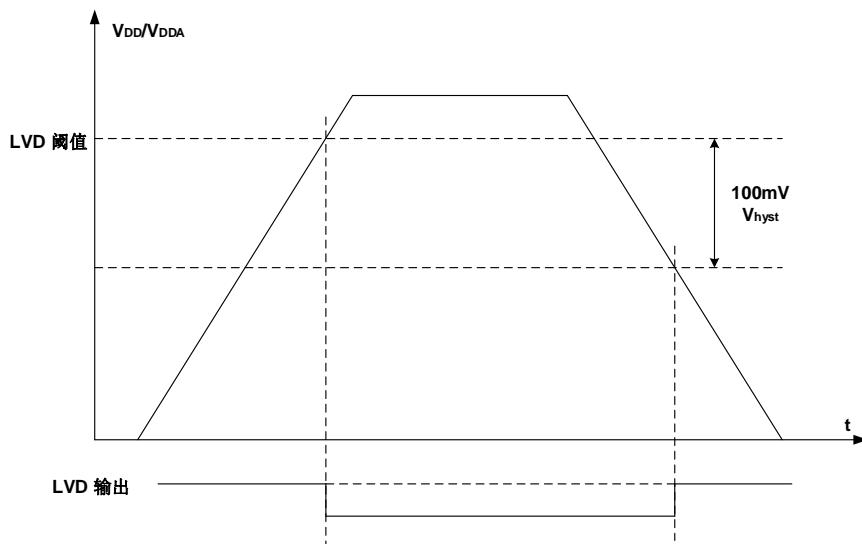


#### $V_{DDA}$ 域

LVD 的功能是检测  $V_{DD} / V_{DDA}$  供电电压是否低于低电压检测阈值，该阈值由电源控制寄存器 (PMU\_CTL) 中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器 (PMU\_CS) 中的 LVDF 位表示低电压事件是否出现，该事件连接至 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。[图 3-3. LVD 阈值波形图](#)显示了  $V_{DD} / V_{DDA}$  供电电压和 LVD 输出信号的关系。(LVD 中断信号依赖于 EXTI 第 16 线的上升或下降沿配置)。迟滞电压  $V_{hyst}$  值为 100mV。

**图3-3. LVD阈值波形图**



一般来说，数字电路由  $V_{DD}$  供电，而大多数的模拟电路由  $V_{DDA}$  供电。为了提高 ADC 的转换精度，为  $V_{DDA}$  独立供电可使模拟电路达到更好的特性。为避免噪声， $V_{DDA}$  通过外部滤波电路连接至  $V_{DD}$ ，相应的  $V_{SSA}$  通过特定电路连接至  $V_{SS}$ 。否则，如果  $V_{DDA}$  和  $V_{DD}$  不同时， $V_{DDA}$  须高于  $V_{DD}$ ，但压差不超过 0.3V。

### 3.3.3. 1.2V 电源域

主要功能包括 Cortex®-M23 内核逻辑、AHB / APB 外设、备份域和  $V_{DD} / V_{DDA}$  域的 APB 接口等。当 1.2V 电压上电后，POR 将在 1.2V 域中产生一个复位序列，复位完成后，要进入指定的省电模式，须先配置相关的控制位，之后一旦执行 WFI 或 WFE 指令，设备便进入该省电模式。关于这方面的详细内容，将在以下章节予以说明。

### 3.3.4. 省电模式

系统复位或电源复位后，GD32E23x MCU 处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有：减慢系统时钟 (HCLK, PCLK1, PCLK2)，关闭未使用的外设的时钟或通过 PMU\_CTL 寄存器的 LDOVS 来配置 LDO 输出电压。LDOVS 只有在 PLL 关闭情况下才可以配置。此外，三种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式和待机模式。

#### 睡眠模式

睡眠模式与 Cortex®-M23 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M23 的时钟。如需进入睡眠模式，只要清除 Cortex®-M23 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(如果 SEVONPEND

为1，任何中断都可以唤醒系统，请参考Cortex®-M23技术手册）。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间最短。

根据Cortex®-M23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®-M23 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，1.2V 域中的所有时钟全部关闭，IRC8M、IRC28M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU\_CTL 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU\_CTL 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M23 技术手册）。刚退出深度睡眠模式时，IRC8M 被选中作为系统时钟。请注意，如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

**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）和相关外设标志位必须被复位，参考[表5-3. EXTI 触发源](#)。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®-M23 的 SLEEPDEEP 模式实现的。在待机模式下，整个 1.2V 域全部停止供电，同时 LDO 和包括 IRC8M、IRC28M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU\_CTL 寄存器的 STBMOD 位置 1，再清除 PMU\_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU\_CS 寄存器的 STBF 位状态表示 MCU 是否曾进入过待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟/时间戳/侵入事件，FWDGT 复位，WKUP 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 1.2V 电源域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®-M23 将从 0x00000000 地址开始执行指令代码。

**表3-1. 节电模式总结**

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1、关闭 1.2V 电源域的所有时钟 2、关闭 IRC8M、IRC28M、HXTAL 和 PLL	1、关闭 1.2V 电源域的供电 2、关闭 IRC8M、IRC28M、HXTAL 和 PLL
LDO 状态	开启（正常功耗模式）	开启（正常功耗模式或者低功耗模式）	关闭

模式	睡眠	深度睡眠	待机
		耗模式)	
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	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（或 SEVONPEND=1 时的中断）均可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、RTC
唤醒延迟	无	IRC8M 唤醒时间 如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	上电序列

**注意：**在待机模式下，除了 RESET 引脚，配置为 RTC 功能的 PC13，用作 LXTAL 晶振引脚的 PC14 和 PC15，使能的 WKUP 引脚，其他所有 I/O 都处于高阻态。

### 3.4. PMU 寄存器

PMU 基地址: 0x4000 7000

### 3.4.1. 控制寄存器 (PMU CTL)

地址偏移: 0x00

复位值: 0x0000 4000 (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
LDOVS[1:0]	保留				BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP		
rs					RW				RW	RC	RC	RC	RC	RW	RW

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:14	LDOVS[1:0]	<p>选择LDO输出 在PLL关闭时，这些位由软件配置。</p> <p>00: 保留（LDO输出高电压模式） 01: LDO输出高电压模式 1x: LDO输出低电压模式</p>
13:9	保留	必须保持复位值。
8	BKPWEN	<p>备份域写使能 0: 禁止对备份域寄存器的写访问 1: 允许对备份域寄存器的写访问 复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置1。</p>
7:5	LVDT[2:0]	<p>低电压检测器阈值 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111: 3.1V</p>
4	LVDEN	<p>低电压检测器使能 0: 关闭低电压检测器</p>

1: 开启低电压检测器

**注意:** 当SYSCFG\_CFG2寄存器里的LVD\_LOCK位被置1时, LVDEN和LVDT[2:0]仅可读。

3	STBRST	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	WURST	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	STBMOD	待机模式 0: 当Cortex®-M23进入SLEEPDEEP模式时, 系统进入深度睡眠模式 1: 当Cortex®-M23进入SLEEPDEEP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IRC8M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。

### 3.4.2. 电源控制和状态寄存器 (PMU\_CS)

#### GD32E230xx 产品

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	WUPEN6	WUPEN5	保留	保留	WUPEN1	WUPEN0	保留	保留	保留	保留	保留	LVDF	STBF	WUF	r	r

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	WUPEN6	WKUP引脚6 (PB15) 唤醒使能 0: 关闭WKUP引脚6唤醒功能 1: 开启WKUP引脚6唤醒功能

如果WUPEN6在进入省电模式之前置1，WKUP引脚6的上升沿会将系统从省电模式唤醒。由于WKUP引脚6为高电平有效，WKUP引脚6内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。

13	WUPEN5	WKUP引脚5（PB5）唤醒使能 0：关闭WKUP引脚5唤醒功能 1：开启WKUP引脚5唤醒功能 如果WUPEN5在进入省电模式之前置1，WKUP引脚5的上升沿会将系统从省电模式唤醒。由于WKUP引脚5为高电平有效，WKUP引脚5内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
12:10	保留	必须保持复位值。
9	WUPEN1	WKUP引脚1（PC13）唤醒使能 0：关闭WKUP引脚1唤醒功能 1：开启WKUP引脚1唤醒功能 如果WUPEN1在进入省电模式之前置1，WKUP引脚1的上升沿会将系统从省电模式唤醒。由于WKUP引脚1为高电平有效，WKUP引脚1内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
8	WUPENO	WKUP引脚0（PA0）唤醒使能 0：关闭WKUP引脚0唤醒功能 1：开启WKUP引脚0唤醒功能 如果WUPENO在进入省电模式之前置1，WKUP引脚0的上升沿会将系统从省电模式唤醒。由于WKUP引脚0为高电平有效，WKUP引脚0内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
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 <sub>DD</sub> 高于设定的LVD阈值） 1：低电压事件出现（V <sub>DD</sub> 等于或低于LVD阈值） <b>注意：</b> LVD功能在待机模式被禁用。
1	STBF	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PMU_CTL寄存器的STBRST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唤醒事件，包括RTC侵入事件、RTC闹钟事件、RTC时间戳事件。 该位只能由POR / PDR或通过设置PMU_CTL寄存器的WURST位来清零。

## GD32E231xx 产品

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	WUPEN6	WUPEN5	保留	保留	WUPEN0	保留	保留	保留	保留	保留	LVDF	STBF	WUF	r	r	r

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	WUPEN6	WKUP引脚6 (PB15) 唤醒使能 0: 关闭WKUP引脚6唤醒功能 1: 开启WKUP引脚6唤醒功能 如果WUPEN6在进入省电模式之前置1, WKUP引脚6的上升沿会将系统从省电模式唤醒。由于WKUP引脚6为高电平有效, WKUP引脚6内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
13	WUPEN5	WKUP引脚5 (PB5) 唤醒使能 0: 关闭WKUP引脚5唤醒功能 1: 开启WKUP引脚5唤醒功能 如果WUPEN5在进入省电模式之前置1, WKUP引脚5的上升沿会将系统从省电模式唤醒。由于WKUP引脚5为高电平有效, WKUP引脚5内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
12:9	保留	必须保持复位值。
8	WUPENO	WKUP引脚0 (PA0) 唤醒使能 0: 关闭WKUP引脚0唤醒功能 1: 开启WKUP引脚0唤醒功能 如果WUPENO在进入省电模式之前置1, WKUP引脚0的上升沿会将系统从省电模式唤醒。由于WKUP引脚0为高电平有效, WKUP引脚0内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
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) <b>注意:</b> LVD功能在待机模式被禁用。
1	STBF	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PMU_CTL寄存器的STBRST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唤醒事件，包括RTC侵入事件、RTC闹钟事件、RTC时间戳事件。

该位只能由POR / PDR或通过设置PMU\_CTL寄存器的WURST位来清零。

## 4. 复位和时钟单元（RCU）

### 4.1. 复位控制单元（RCTL）

#### 4.1.1. 简介

GD32E23x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电源基准准备好提供1.2V电压给 GD32E23x产品时，电源复位电平将变为无效。复位入口向量被固定在存储器映射地址0x0000\_0004。

##### 系统复位

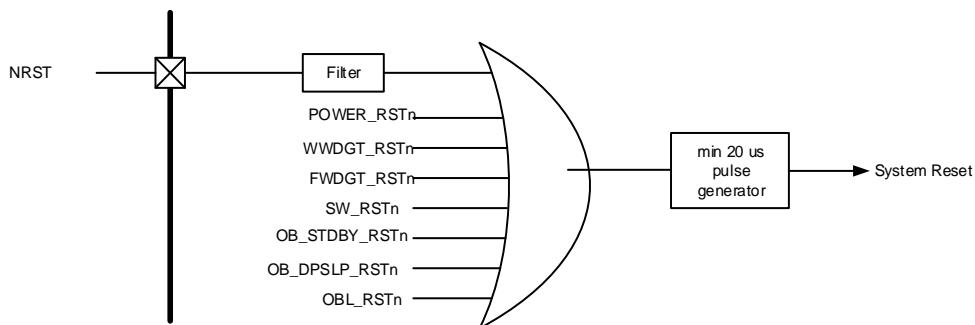
当发生以下任一事件时，产生一个系统复位：

- 电源复位（POWER\_RSTn）；
- 外部引脚复位（NRST）；
- 窗口看门狗定时器计数终止（WWDGT\_RSTn）；
- 独立看门狗定时器计数终止（FWDGT\_RSTn）；
- Cortex®-M23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nRST\_STDBY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nRST\_DPSLP设置为0，并且进入深度睡眠模式时（OB\_DPSLP\_RSTn）。

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。

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（V<sub>DD</sub>重新上电）。

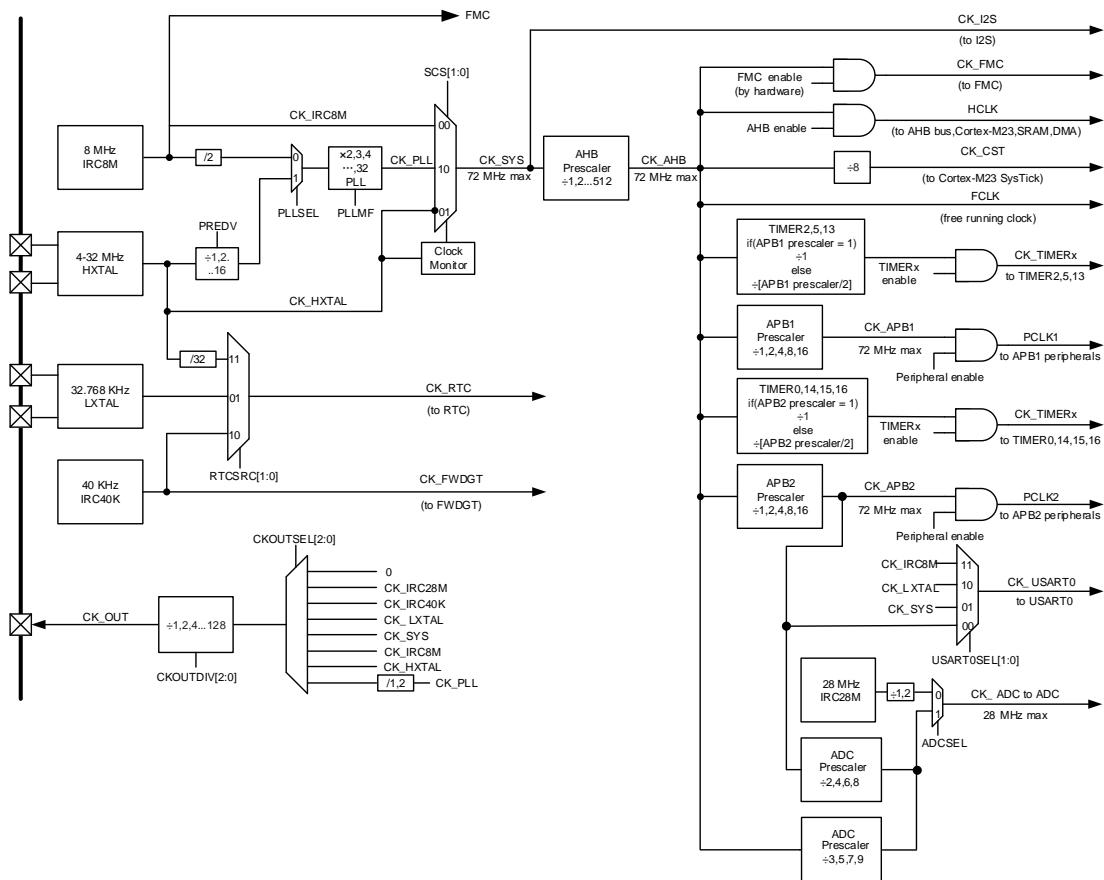
## 4.2. 时钟控制单元（CCTL）

### 4.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部8M RC振荡器时钟（IRC8M）、一个内部28M RC振荡器时钟（IRC28M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部低速RC振荡器时钟（IRC40K）、一个外部低速晶体振荡器时钟（LXTAL）、一个锁相环（PLL）、一个HXTAL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®-M23时钟都源自系统时钟（CK\_SYS），系统时钟的时钟源为IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到72MHz。独立看门狗定时器定时器有独立的时钟源（IRC40K），实时时钟（RTC）使用IRC40K、LXTAL或HXTAL/32作为时钟源。

图4-2. 时钟树



预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2和APB1域的最高时钟频率分别为72MHz、72MHz和72MHz。RCU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GD32E23x产品中ADC时钟由APB2时钟经2、4、6、8分频或由AHB时钟经3、5、7、9分频或IRC28M或IRC28M/2时钟获得，它们是通过设置配置寄存器2（RCU\_CFG2）的ADCSEL位来选择ADC时钟源的。USART0的时钟可以选择IRC8M时钟、LXTAL时钟、系统时钟或APB2时钟，通过设置配置寄存器2（RCU\_CFG2）的USART0SEL位来选择。

RTC时钟可以选择LXTAL时钟、IRC40K时钟或HXTAL时钟32分频，通过设置备用域控制寄存器（RCU\_BDCTL）的RTCSRC位域来选择。

FWDGT时钟可以选择IRC40K时钟，当FWDGT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AHB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（HXTAL）
- 8 MHz内部高速RC振荡器（IRC8M）

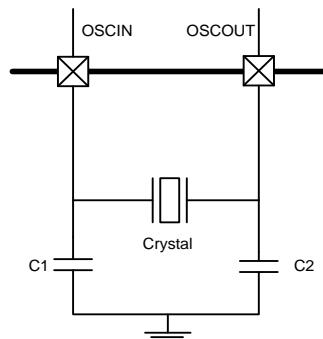
- 28 MHz内部高速RC振荡器（IRC28M）
- 32,768 Hz外部低速晶体振荡器（LXTAL）
- 40 kHz内部低速RC振荡器（IRC40K）
- PLL时钟源可以是HXTAL或IRC8M
- HXTAL时钟监视

#### 4.2.3. 功能描述

##### 高速外部晶体振荡器时钟（HXTAL）

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HXTAL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4-3. HXTAL时钟源



HXTAL晶体可以通过设置时钟控制寄存器RCU\_CTL0的HXTALEN位来启动或关闭，在时钟控制寄存器RCU\_CTL0中的HXTALSTB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HXTAL时钟稳定后，如果在时钟中断寄存器RCU\_INT中的相应中断使能位HXTALSTBIE位被置‘1’，将会产生相应中断。在这一点上，HXTAL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RCU\_CTL0的HXTALBPS和HXTALEN位置‘1’可以设置外部时钟旁路模式。CK\_HXTAL等于驱动OSCIN管脚的外部时钟。

##### 高速内部8MHz RC振荡器时钟（IRC8M）

高速内部8MHz RC振荡器时钟，简称IRC8M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IRC8M时钟。IRC8M 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。IRC8M晶体可以通过设置时钟控制寄存器（RCU\_CTL0）中的IRC8MEN位被启动和关闭。时钟控制寄存器RCU\_CTL0中的IRC8MSTB位用来指示IRC8M内部RC振荡器是否稳定。IRC8M振荡器的启动时间比HXTAL晶体振荡器要短。如果时钟中断寄存器RCU\_INT中的相应中断使能位IRC8MSTBIE被置‘1’，在IRC8M稳定以后，将产生一个中断。IRC8M时钟也可用作PLL输入时钟。

工厂会校准IRC8M时钟频率的精度，但是它的精度仍然比HXTAL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HXTAL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IRC8M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0（RCU\_CTL0）中的PLLEN位被启动和关闭。时钟控制寄存器RCU\_CTL0中的PLLSTB位用来指示PLL时钟是否稳定。如果时钟中断寄存器RCU\_INT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28MHz RC振荡器时钟（IRC28M）

高速内部28MHz RC振荡器时钟（IRC28M）有一个固定的频率28MHz，专门用作ADC时钟。IRC28M可以通过设置时钟控制寄存器1（RCU\_CTL1）中的IRC28MEN位被启动和关闭。时钟控制寄存器1（RCU\_CTL1）中的IRC28MSTB位用来指示IRC28M时钟是否已稳定。如果时钟中断寄存器RCU\_INT中的相应中断使能位IRC28MSTBIE被置‘1’，在IRC28M稳定以后，将产生一个中断。

### 低速外部晶体振荡器时钟（LXTAL）

LXTAL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。LXTAL时钟可以通过设置备份域控制寄存器（RCU\_BDCTL）中的LXTALEN位被启动和关闭。备份域控制寄存器RCU\_BDCTL中的LXTALSTB位用来指示LXTAL时钟是否稳定。如果时钟中断寄存器RCU\_INT中的相应中断使能位LXTALSTBIE被置‘1’，在LXTAL稳定以后，将产生一个中断。

将备份域控制寄存器RCU\_BDCTL的LXTALBPS和LXTALEN位置‘1’可以选择外部时钟旁路模式。CK\_LXTAL与连到OSC32IN脚上外部时钟信号一致。

### 低速内部RC振荡器时钟（IRC40K）

IRC40K 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，为独立看门狗定时器和实时时钟电路提供时钟。IRC40K提供低成本的时钟源，因为不需要外部器件。IRC40K RC振荡器可以通过设置控制/状态寄存器RCU\_RSTSCK中的IRC40KEN位被启动和关闭。控制/状态寄存器RCU\_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果时钟中断寄存器RCU\_INT中的相应中断使能位IRC40KSTBIE被置‘1’，在IRC40K稳定以后，将产生一个中断。

### 系统时钟（CK\_SYS）选择

系统复位后，IRC8M时钟被选为系统时钟，改变时钟配置寄存器RCU\_CFG0中的系统时钟变换位SCS可以切换系统时钟源为HXTAL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。

### HXTAL时钟监视器（CKM）

设置时钟控制寄存器RCU\_CTL0中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL

故障，HXTAL将自动被禁止，时钟中断寄存器RCU\_INT中的HXTAL时钟阻塞标志位CKMIF将被置‘1’，产生HXTAL故障事件。这个故障引发的中断和Cortex-M23的不可屏蔽中断相连。如果HXTAL被选作系统或PLL的时钟源，HXTAL故障将促使选择IRC8M为系统时钟源且PLL将被自动禁止。

### 时钟输出功能

时钟输出功能输出从32kHz到72MHz的时钟。通过设置时钟配置寄存器RCU\_CFG0中的CK\_OUT时钟源选择位CKOUTSEL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_IRC28M
010	CK_IRC40K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M
110	CK_HXTAL
111	CK_PLL 或 CK_PLL/2

通过配置时钟配置寄存器RCU\_CFG0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。

### 深度睡眠模式时钟控制

当MCU工作在深度睡眠模式时，USART0能唤醒MCU，前提是它们的时钟是由LXTAL时钟提供且LXTAL时钟被使能。

如果USART0时钟选择IRC8M时钟并且工作在深度睡眠模式，它们有能力开启或关闭IRC8M时钟，USART0采用IRC8M时钟作为工作时钟来唤醒深度睡眠模式。

### 电压控制

深度睡眠模式电压寄存器（RCU\_DSV）中的DSLPVS[1:0]位可以控制内核在深度睡眠模式下的电压。

**表 4-2. 深度睡眠模式下内核电压选择**

DSLPVS[1:0]	深度睡眠模式电压 (V)
00	1.0
01	0.9
10	0.8
11	1.2

RCU\_DSV寄存器被电源解锁寄存器（RCU\_VKEY）保护。只有在写0xA2B3C4D到RCU\_VKEY后，RCU\_DSV寄存器才能被写入。

## 4.3. RCU 寄存器

RCU基地址: 0x4002 1000

### 4.3.1. 控制寄存器 0 (RCU\_CTL0)

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	
						PLLSTB	PLLEN					CKMEN	HXTALB PS	HXTALST B	HXTALE N	
													RW	RW	R	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
						IRC8MCALIB[7:0]								IRC8MST B	IRC8MEN	
														R	RW	

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	PLLSTB	<b>PLL时钟稳定标志位</b> 硬件置‘1’来指示PLL输出时钟是否稳定待用。 0: PLL不稳定 1: PLL稳定
24	PLLEN	<b>PLL使能</b> 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	<b>HXTAL时钟监视使能</b> 0: 禁止外部4 ~ 32 MHz 晶体振荡器(HXTAL) 时钟监视器 1: 使能外部4 ~ 32 MHz 晶体振荡器(HXTAL) 时钟监视器 当硬件监测到HXTAL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IRC8M 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CKMIF位。 <b>注意：</b> 使能HXTAL时钟监视器以后，硬件无视控制位IRC8MEN的状态，自动使能IRC8M时钟。
18	HXTALBPS	外部晶体振荡器(HXTAL)时钟旁路模式使能

		只有在HXTALEN位为0时， HXTALBPS位才可写。													
		0: 禁止HXTAL旁路模式 1: 使能HXTAL旁路模式， HXTAL输出时钟等于输入时钟													
17	HXTALSTB	外部晶体振荡器（HXTAL）时钟稳定状态标志位 硬件置‘1’来指示HXTAL振荡器时钟是否稳定待用。 0: HXTAL振荡器未稳定 1: HXTAL振荡器已稳定													
16	HXTALEN	外部高速振荡器时钟使能 软件置‘1’或清‘0’。如果HXTAL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	IRC8MCALIB[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	IRC8MADJ[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IRC8MADJ当前值加上IRC8MCALIB[7:0]位的值。 最终调整值应该调整IRC8M到8 MHz ± 1%。													
2	保留	必须保持复位值。													
1	IRC8MSTB	高速内部（IRC8M）时钟稳定状态标志位 硬件置‘1’来指示IRC8M振荡器时钟是否稳定待用。 0: IRC8M振荡器未稳定 1: IRC8M振荡器已稳定													
0	IRC8MEN	高速内部振荡器使能 软件复位置位。如果IRC8M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HXTAL振荡器发生故障时， 该位由硬件置1来启动IRC8M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. 配置寄存器 0 (RCU\_CFG0)

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]	CKOUTSEL[2:0]	保留	PLLMF[3:0]	PLLPRE DV	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

ADCPSC[1:0]	APB2PSC[2:0]	APB1PSC[2:0]	AHBPSC[3:0]	SCSS[1:0]	SCS[1:0]
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RCU_CFG0的26:24位。 000: CK_OUT不分频 001: CK_OUT 2分频 010: CK_OUT 4分频 011: CK_OUT 8分频 100: CK_OUT 16分频 101: CK_OUT 32分频 110: CK_OUT 64分频 111: CK_OUT 128分频
27	PLLMF[4]	PLLMF位域的位4 见RCU_CFG0的21:18位
26:24	CKOUTSEL[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: 选择内部28M RC振荡器时钟 010: 选择内部40K RC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8M RC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	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RCU_CFG0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	<b>PLLPREDV</b>	HXTAL作为PLL输入。该位与时钟配置寄存器1（RCU_CFG1）中的PREDV[0]位是一样的。参考RCU_CFG1的PREDV位说明。 由软件置1或清0来分频或不分频后，当作PLL输入时钟源。 0: 选择HXTAL时钟 1: 选择HXTAL时钟二分频
16	<b>PLLSEL</b>	PLL时钟源选择 软件置1或清0来控制PLL时钟源 0: 选择IRC8M二分频为PLL时钟源 1: 选择HXTAL（RCU_CFG1寄存器的PLLPRESEL位）为PLL时钟源
15:14	<b>ADCPSC[1:0]</b>	ADC时钟预分频选择 软件写三位包括RCU_CFG2的31位来确定ADC时钟分频。 软件清0和置1。 000: 选择APB2时钟2分频 001: 选择APB2时钟4分频 010: 选择APB2时钟6分频 011: 选择APB2时钟8分频 100: 选择AHB时钟3分频

		101: 选择AHB时钟5分频 110: 选择AHB时钟7分频 111: 选择AHB时钟9分频
13:11	APB2PSC[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	APB1PSC[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	AHBPSC[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_IRC8M作为CK_SYS系统时钟源 01: 选择CK_HXTAL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HXTAL出现故障时，强制选择IRC8M作为系统时钟或PLL时钟。  00: 选择IRC8M时钟作为CK_SYS系统时钟源 01: 选择HXTAL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

### 4.3.3. 中断寄存器 (RCU\_INT)

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	
								CKMIC			IRC28M STBIC	PLL STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留	IRC28M STBIE	PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	CKMIF	保留	IRC28M STBIF	PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF		
	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	CKMIC	HXTAL时钟阻塞中断清除 软件写1复位CKMIF标志位。 0: 不复位CKMIF标志位 1: 复位CKMIF标志位
22	保留	必须保持复位值。
21	IRC28MSTBIC	IRC28M时钟稳定中断清除 软件写1复位IRC28MSTBIF标志位。 0: 不复位IRC28MSTBIF标志位 1: 复位IRC28MSTBIF标志位
20	PLLSTBIC	PLL稳定中断清除 软件写1复位PLLSTBIF标志位。 0: 不复位PLLSTBIF标志位 1: 复位PLLSTBIF标志位
19	HXTALSTBIC	HXTAL时钟稳定中断清除 软件写1复位HXTALSTBIF标志位。 0: 不复位HXTALSTBIF标志位 1: 复位HXTALSTBIF标志位
18	IRC8MSTBIC	IRC8M时钟稳定中断清除 软件写1复位IRC8MSTBIF标志位。 0: 不复位IRC8MSTBIF标志位

		1: 复位IRC8MSTBIF标志位
17	LXTALSTBIC	<p>LXTAL时钟稳定中断清除 软件写1复位LXTALSTBIF标志位。</p> <p>0: 不复位LXTALSTBIF标志位 1: 复位LXTALRDYF标志位</p>
16	IRC40KSTBIC	<p>IRC40K时钟稳定中断清除 软件写1复位IRC40KSTBIF标志位。</p> <p>0: 不复位IRC40KSTBIF标志位 1: 复位IRC40KSTBIF标志位</p>
15:14	保留	必须保持复位值。
13	IRC28MSTBIE	<p>IRC28M时钟稳定中断使能 软件置1和清0来使能/禁止IRC28M时钟稳定中断。</p> <p>0: 禁止IRC28M时钟稳定中断 1: 使能IRC28M时钟稳定中断</p>
12	PLLSTBIE	<p>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。</p> <p>0: 禁止PLL时钟稳定中断 1: 使能PLL时钟稳定中断</p>
11	HXTALSTBIE	<p>HXTAL时钟稳定中断使能 软件置1和清0来使能/禁止HXTAL时钟稳定中断。</p> <p>0: 禁止HXTAL时钟稳定中断 1: 使能HXTAL时钟稳定中断</p>
10	IRC8MSTBIE	<p>IRC8M时钟稳定中断使能 软件置1和清0来使能/禁止IRC8M时钟稳定中断。</p> <p>0: 禁止IRC8M时钟稳定中断 1: 使能IRC8M时钟稳定中断</p>
9	LXTALSTBIE	<p>LXTAL时钟稳定中断使能 LXTAL时钟稳定中断使能/禁止控制。</p> <p>0: 禁止LXTAL时钟稳定中断 1: 使能LXTAL时钟稳定中断</p>
8	IRC40KSTBIE	<p>IRC40K时钟稳定中断使能 IRC40K时钟稳定中断使能/禁止控制。</p> <p>0: 禁止IRC40K时钟稳定中断 1: 使能IRC40K时钟稳定中断</p>
7	CKMIF	<p>HXTAL时钟阻塞中断标志位 当HXTAL时钟阻塞时硬件置1。 软件置CKMIC=1时清除该位。</p> <p>0: 时钟运行正常</p>

		1: HXTAL时钟阻塞
6	保留	必须保持复位值。
5	IRC28MSTBIF	<p>IRC28M时钟稳定中断标志位</p> <p>当IRC28M时钟稳定且IRC28MSTBIE位被置1时由硬件置1。</p> <p>软件置IRC28MSTBIC=1时清除该位。</p> <p>0: 无IRC28M时钟稳定中断产生</p> <p>1: IRC28M时钟稳定中断发生</p>
4	PLLSTBIF	<p>PLL时钟稳定中断标志位</p> <p>当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1。</p> <p>软件置PLLSTBIC=1时清除该位。</p> <p>0: 无PLL时钟稳定中断产生</p> <p>1: 产生PLL时钟稳定中断</p>
3	HXTALSTBIF	<p>HXTAL时钟稳定中断标志位</p> <p>当外部4 ~ 32 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1。</p> <p>软件置HXTALSTBIC=1时清除该位。</p> <p>0: 无HXTAL时钟稳定中断发生</p> <p>1: 发生HXTAL时钟稳定中断</p>
2	IRC8MSTBIF	<p>IRC8M时钟稳定中断标志位</p> <p>当内部8 MHz RC振荡器时钟稳定且IRC8MSTBIE位被置1时由硬件置1。</p> <p>软件置IRC8MSTBIC=1时清除该位。</p> <p>0: 无IRC8M时钟稳定中断产生</p> <p>1: 产生IRC8M时钟稳定中断</p>
1	LXTALSTBIF	<p>LXTAL 时钟稳定中断标志位</p> <p>当外部32.768KHz晶体振荡器时钟稳定且LXTALSTBIE为被置1时由硬件置1。</p> <p>软件置LXTALSTBIC=1时清除该位。</p> <p>0: 无LXTAL时钟稳定中断发生</p> <p>1: 发生LXTAL时钟稳定中断</p>
0	IRC40KSTBIF	<p>IRC40K时钟稳定中断标志位</p> <p>当内部32kHz RC振荡器时钟稳定且IRC40KSTBIE位被置1时由硬件置1。</p> <p>软件置IRC40KSTBIC =1时清除该位。</p> <p>0: 无IRC40K时钟稳定中断产生</p> <p>1: 产生IRC40K时钟稳定中断</p>

#### 4.3.4. APB2 复位寄存器 (**RCU\_APB2RST**)

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												TIMER16	TIMER15	TIMER14	
												RST	RST	RST	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0 RST	保留.	SPI0 RST	TIMER0 RST	保留	ADC RST	保留						CFGCMP 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	<b>TIMER16RST</b>	<b>TIMER16定时器复位</b> 由软件置1或清0。 0: 无复位 1: 复位 <b>TIMER16</b> 定时器
17	<b>TIMER15RST</b>	<b>TIMER15定时器复位</b> 由软件置1或清0。 0: 无复位 1: 复位 <b>TIMER15</b> 定时器
16	<b>TIMER14RST</b>	<b>TIMER14定时器复位</b> 由软件置1或清0。 0: 无复位 1: 复位 <b>TIMER14</b> 定时器
15	保留	必须保持复位值。
14	<b>USART0RST</b>	<b>USART0复位</b> 由软件置1或清0。 0: 无复位 1: 复位 <b>USART0</b>
13	保留	必须保持复位值。
12	<b>SPI0RST</b>	<b>SPI0复位</b> 由软件置1或清0。 0: 无复位 1: 复位 <b>SPI0</b>
11	<b>TIMER0RST</b>	<b>TIMER0定时器复位</b> 由软件置1或清0。 0: 无复位 1: 复位 <b>TIMER0</b> 定时器
10	保留	必须保持复位值。
9	<b>ADCRST</b>	<b>ADC复位</b>

由软件置1或清0。

0: 无复位

1: 复位ADC

8:1 保留 必须保持复位值。

0 CFGCMRST 系统配置和比较器复位

由软件置1或清0。

0: 无复位

1: 复位系统配置和比较器模块

#### 4.3.5. APB1 复位寄存器 (**RCU\_APB1RST**)

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	PMU RST		保留		I2C1 RST	I2C0 RST		保留		USART1 RST		保留			
			rw						rw	rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI1 RST	保留	WWDGTR RST	保留	TIMER13 RST	保留		保留		TIMER5 RST	保留	TIMER2 RST	保留		
			rw		rw				rw		rw				rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMURST	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
27:23	保留	必须保持复位值。
22	I2C1RST	I2C1复位 由软件置1或清0。 0: 无复位 1: 复位I2C1
21	I2C0RST	I2C0复位 由软件置1或清0。 0: 无复位 1: 复位I2C0

---

20:18	保留	必须保持复位值。
17	USART1RST	USART1复位 由软件置1或清0。 0: 无复位 1: 复位USART1
16:15	保留	必须保持复位值。
14	SPI1RST	SPI1复位 由软件置1或清0。 0: 无复位 1: 复位SPI1
13:12	保留	必须保持复位值。
11	WWDGTRST	窗口看门狗定时器复位 由软件置1或清0。 0: 无复位 1: 复位窗口看门狗定时器
10:9	保留	必须保持复位值。
8	TIMER13RST	TIMER13定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER13定时器
7:5	保留	必须保持复位值。
4	TIMER5RST	TIMER5定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER5定时器
3:2	保留	必须保持复位值。
1	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
0	保留	必须保持复位值。

#### 4.3.6. AHB 使能寄存器（RCU\_AHBen）

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				PFEN	保留.		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
保留				CRCEN	保留	FMC SPEN	保留	SRAM SPEN	保留	DMAEN					
				rw			rw	rw	rw						

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PFEN	GPIOF时钟使能 由软件置1或清0。 0: GPIOF时钟关闭 1: GPIOF时钟开启
21:20	保留	必须保持复位值。
19	PCEN	GPIOC时钟使能 由软件置1或清0。 0: GPIOC时钟关闭 1: GPIOC时钟开启
18	PBEN	GPIOB时钟使能 由软件置1或清0。 0: GPIOB时钟关闭 1: GPIOB时钟开启
17	PAEN	GPIOA时钟使能 由软件置1或清0。 0: GPIOA时钟关闭 1: GPIOA时钟开启
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	FMCSPEN	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	SRAMSPEN	<b>SRAM</b> 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 <b>SRAM</b> 时钟。 0: 关闭睡眠模式下的 <b>SRAM</b> 接口时钟 1: 开启睡眠模式下的 <b>SRAM</b> 接口时钟
1	保留	必须保持复位值。
0	DMAEN	<b>DMA</b> 时钟使能 由软件置1或清0。 0: 关闭 <b>DMA</b> 时钟 1: 开启 <b>DMA</b> 时钟

#### 4.3.7. APB2 使能寄存器 (**RCU\_APB2EN**)

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								DBGMCU	保留		TIMER16	TIMER15	TIMER14		
					EN				EN	EN	EN				
									RW		RW	RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0 EN	保留	SPI0EN	TIMER0E N	保留	ADCEN	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。
22	DBGMCUEN	<b>DBGMCU</b> 时钟使能 由软件置1或清0。 0: 关闭 <b>DBGMCU</b> 时钟 1: 开启 <b>DBGMCU</b> 时钟
21:19	保留	必须保持复位值。
18	TIMER16EN	<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
17	TIMER15EN	<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TIMER15定时器时钟

16	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
15	保留	必须保持复位值。
14	USART0EN	USART0时钟使能 由软件置1或清0。 0: 关闭USART0时钟 1: 开启USART0时钟
13	保留	必须保持复位值。
12	SPI0EN	SPI0时钟使能 由软件置1或清0。 0: 关闭SPI0时钟 1: 开启SPI0时钟
11	TIMER0EN	TIMER0定时器时钟使能 由软件置1或清0。 0: 关闭TIMER0定时器时钟 1: 开启TIMER0定时器时钟
10	保留	必须保持复位值。
9	ADCEN	ADC接口时钟使能 由软件置1或清0。 0: 关闭ADC接口时钟 1: 开启ADC接口时钟
8:1	保留	必须保持复位值。
0	CFGCMPEN	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 使能寄存器 (**RCU\_APB1EN**)

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	PMUEN	保留				I2C1EN	I2C0EN	保留	USART1EN	保留					
		rw				rw	rw			rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI1EN	保留	WWDGTE	EN	保留	TIMER13E	N	保留	TIMER5E	N	保留	TIMER2E	N	保留	
			rw		rw		rw		rw		rw		rw		

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMUEN	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
27:23	保留	必须保持复位值。
22	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
21	I2C0EN	I2C0时钟使能 由软件置1或清0。 0: 关闭I2C0时钟 1: 开启I2C0时钟
20:18	保留	必须保持复位值。
17	USART1EN	USART1时钟使能 由软件置1或清0。 0:关闭USART1时钟 1:开启USART1时钟
16:15	保留	必须保持复位值。
14	SPI1EN	SPI1时钟使能 由软件置1或清0。 0: 关闭SPI1时钟 1: 开启SPI1时钟
13:12	保留	必须保持复位值。
11	WWDGTE	窗口看门狗定时器时钟使能 由软件置1或清0。 0: 关闭窗口看门狗定时器时钟

1: 开启窗口看门狗定时器时钟

10:9	保留	必须保持复位值。
8	TIMER13EN	TIMER13定时器时钟使能 由软件置1或清0。 0: 关闭TIMER13定时器时钟 1: 开启TIMER13定时器时钟
7:5	保留	必须保持复位值。
4	TIMER5EN	TIMER5定时器时钟使能 由软件置1或清0。 0: 关闭TIMER5定时器时钟 1: 开启TIMER5定时器时钟
3:2	保留	必须保持复位值。
1	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
0	保留	必须保持复位值。

#### 4.3.9. 备份域控制寄存器（RCU\_BDCTL）

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（BDCTL）的LXTALEN, LXTALBPS, RTCSRC和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器（PMU\_CTL）中的BKPWEN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			LXTALDRI[1:0]	LXTALBPS	LXTALSTB	LXTALEN		
											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LXTAL时钟作为RTC时钟源 10: 选择IRC40K时钟作为RTC时钟源 11: 选择HXTAL时钟32分频作为RTC时钟源
7:5	保留	必须保持复位值。
4:3	LXTALDRI[1:0]	LXTAL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） <b>注意：</b> LXTALDRI在旁路模式下无效
2	LXTALBPS	LXTAL旁路模式使能 软件置1和清0。 0: 禁止LXTAL旁路模式 1: 使能LXTAL旁路模式
1	LXTALSTB	外部低速振荡器稳定状态位 硬件置1来指示LXTAL输出时钟是否稳定待用。 0: LXTAL未稳定 1: LXTAL已稳定
0	LXTALEN	LXTAL使能 软件置1和清0。 0: 关闭LXTAL 1: 开启LXTAL

#### 4.3.10. 复位源/时钟寄存器（RCU\_RSTSCK）

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	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	OBLRSTF	RSTFC	V12RSTF							保留
r	r	r	r	r	r	r	rw	r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															保留
															IRC40K STB
															IRC40K EN

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1到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	WWDGTRSTF	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1到RSTFC位来清除该位。 0: 无窗口看门狗定时器复位发生 1: 发生窗口看门狗定时器复位
29	FWDGTRSTF	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1到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1到RSTFC位来清除该位。 0: 无软件复位发生 1: 发生软件复位
27	PORRSTF	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1到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1到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1到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1到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	IRC40KSTB	IRC40K时钟稳定状态位 该位由硬件置1指示IRC40K输出时钟是否稳定待用。 0: IRC40K时钟未稳定 1: IRC40K时钟已稳定
0	IRC40KEN	IRC40K时钟使能 软件置1和清0。 0: 关闭IRC40K时钟 1: 开启IRC40K时钟

#### 4.3.11. AHB 复位寄存器（RCU\_AHBRST）

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		PFRST	保留	PCRST	PBRST	PARST	保留			
									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
31:23	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
21:20	保留	必须保持复位值。
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. 配置寄存器 1 (**RCU\_CFG1**)

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														PREDV[3:0]	
rw															

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PREDV[3:0]	HXTAL或CK_IRC48M作为PLL输入源分频因子 由软件置1或清0。这些位仅能在PLL关闭时改写。 <b>注意:</b> PREDV的位0与RCU_CFG0的位17的功能相同，修改RCU_CFG0的位17同时改变这里的位0，时钟分频因子为 (PREDV + 1) 0000: PLL的输入，不分频 0001: PLL的输入2分频 0010: PLL的输入3分频

- 0011: PLL的输入4分频
- 0100: PLL的输入5分频
- 0101: PLL的输入6分频
- 0110: PLL的输入7分频
- 0111: PLL的输入8分频
- 1000: PLL的输入9分频
- 1001: PLL的输入10分频
- 1010: PLL的输入11分频
- 1011: PLL的输入12分频
- 1100: PLL的输入13分频
- 1101: PLL的输入14分频
- 1110: PLL的输入15分频
- 1111: PLL的输入16分频

#### 4.3.13. 配置寄存器 2 (RCU\_CFG2)

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
ADCPS C[2]	保留														IRC28MD IV
rw															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							ADCSEL	保留							USART0SEL[1:0]
rw															rw

位/位域	名称	描述
31	ADCPSC[2]	ADCPSC的位2 参考RCU_CFG0的位15:14
30:17	保留	必须保持复位值。
16	IRC28MDIV	IRC28M是否分频 0: IRC28M/2作为ADC时钟 1: IRC28M作为ADC时钟
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IRC28M时钟 1: ADC时钟源选择APB2的2,4,6,8分频和AHB时钟3,5,7,9分频

7:2	保留	必须保持复位值。
1:0	USART0SEL[1:0]	USART0时钟源选择 由软件置1或清0。 00: USART0时钟选择APB2时钟 01: USART0时钟选择系统时钟 10: USART0时钟选择LXTAL时钟 11: USART0时钟选择IRC8M时钟

#### 4.3.14. 控制寄存器 1 (RCU\_CTL1)

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



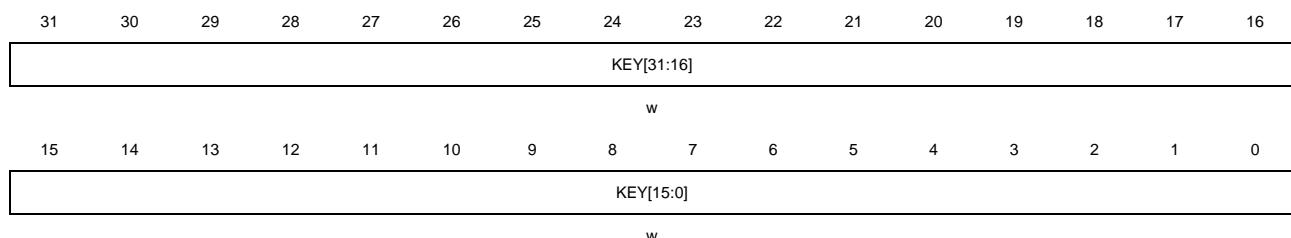
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	IRC28MCALIB[7:0]	IRC28M时钟校准 启动时这些位会被自动初始化为出厂校准参数。
7:3	IRC28MADJ[4:0]	IRC28M时钟调整 这些位由软件设定，最终调整值为(IRC28MADJ)位的值加上IRC28MCALIB[7:0]位的值。最终调整值应该把IRC28M调整到28MHz ±1%。
2	保留	必须保持复位值。
1	IRC28MSTB	IRC28M时钟稳定标志位 由硬件置1来指示内部IRC28M振荡器已经稳定可用。 0: IRC28M振荡器未稳定 1: IRC28M振荡器已稳定
0	IRC28MEN	IRC28M时钟使能 软件置1和清0。 0: 关闭IRC28M时钟 1: 开启IRC28M时钟

#### 4.3.15. 电源解锁寄存器 (RCU\_VKEY)

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



位/位域	名称	描述
31:0	KEY[31:0]	RCU_DSV寄存器解锁 这些位只能被软件写，读的话全是0。只有在向RCU_VKEY 寄存器写0xA2B3C4D后，RCU_DSV寄存器才能被写。

#### 4.3.16. 深度睡眠模式电压寄存器 (RCU\_DSV)

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



位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	DSLPVS[1:0]	深度睡眠模式电压选择 这些位由软件置位和清除。 00: 在深度睡眠模式下内核电压为1.0V 01: 在深度睡眠模式下内核电压为0.9V 10: 在深度睡眠模式下内核电压为0.8V 11: 在深度睡眠模式下内核电压为1.2V

## 5. 中断/事件控制器 (EXTI)

### 5.1. 简介

Cortex-M23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-M23技术参考手册》。

EXTI（中断/事件控制器）包括21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。

### 5.2. 主要特性

- Cortex-M23系统异常；
- 28种可屏蔽的外设中断（GD32E23x产品）；
- 2位中断优先级配置位——共提供4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；
- EXTI中有21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。

### 5.3. 中断功能描述

Arm Cortex-M23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 5-1. Cortex-M23 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
-	4-10	-	0x0000_0010 - 0x0000_002B	保留

异常类型	向量编号	优先级(a)	向量地址	描述
<b>SVCALL 服务调用</b>	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统服务调用
-	12-13	-	0x0000_0030 - 0x0000_0034	保留
<b>PendSV 挂起服务</b>	14	可编程设置	0x0000_0038	可挂起的系统服务请求
<b>系统节拍</b>	15	可编程设置	0x0000_003C	系统节拍定时器

SysTick校准值设为9000, SysTick时钟频率配置为HCLK/8, 此时若HCLK时钟被配置为72MHz, 则SysTick中断会1ms响应一次。

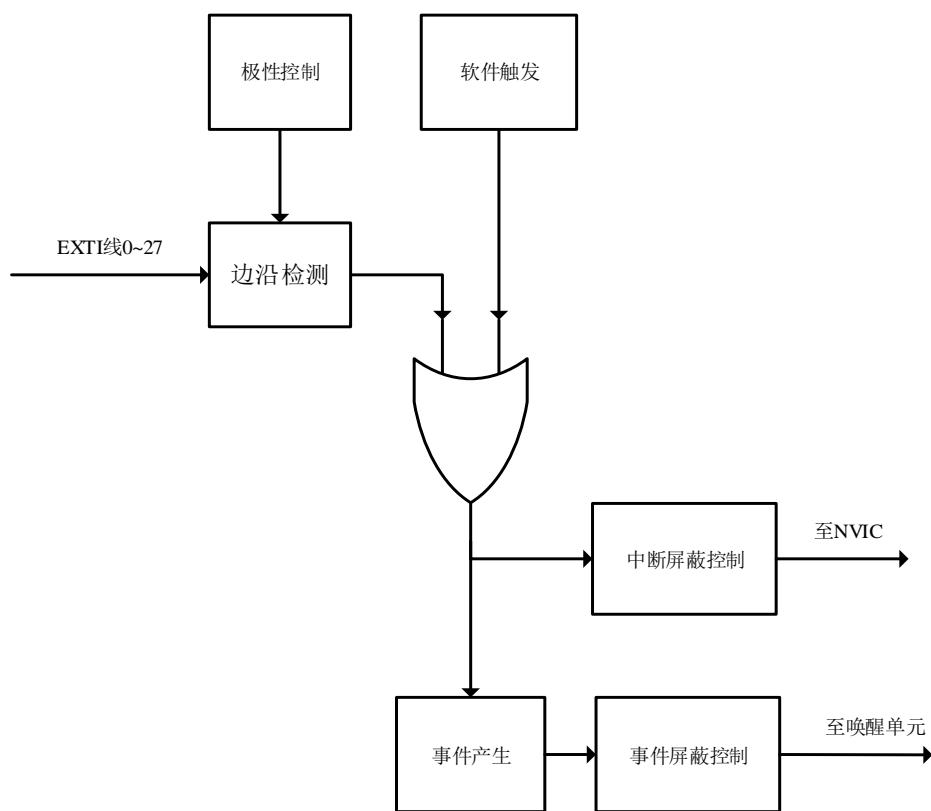
表 5-2. 中断向量表

中断编号	向量编号	外设中断描述	向量地址
<b>IRQ 0</b>	16	窗口看门狗中断	0x0000_0040
<b>IRQ 1</b>	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
<b>IRQ 2</b>	18	RTC 全局中断	0x0000_0048
<b>IRQ 3</b>	19	FMC 全局中断	0x0000_004C
<b>IRQ 4</b>	20	RCU 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	保留	0x0000_0060
<b>IRQ 9</b>	25	DMA 通道 0 全局中断	0x0000_0064
<b>IRQ 10</b>	26	DMA 通道 1-2 全局中断	0x0000_0068
<b>IRQ 11</b>	27	DMA 通道 3-4 全局中断	0x0000_006C
<b>IRQ 12</b>	28	ADC 和 CMP 中断	0x0000_0070
<b>IRQ 13</b>	29	TIMER0 中止, 更新, 触发与通道换相中断	0x0000_0074
<b>IRQ 14</b>	30	TIMER0 捕获比较中断	0x0000_0078
<b>IRQ 15</b>	31	保留	0x0000_007C
<b>IRQ 16</b>	32	TIMER2 全局中断	0x0000_0080
<b>IRQ 17</b>	33	TIMER5 全局中断	0x0000_0084
<b>IRQ 18</b>	34	保留	0x0000_0088
<b>IRQ 19</b>	35	TIMER13 全局中断	0x0000_008C
<b>IRQ 20</b>	36	TIMER14 全局中断	0x0000_0090
<b>IRQ 21</b>	37	TIMER15 全局中断	0x0000_0094
<b>IRQ 22</b>	38	TIMER16 全局中断	0x0000_0098
<b>IRQ 23</b>	39	I2C0 事件中断	0x0000_009C
<b>IRQ 24</b>	40	I2C1 事件中断	0x0000_00A0
<b>IRQ 25</b>	41	SPI0 全局中断	0x0000_00A4
<b>IRQ 26</b>	42	SPI1 全局中断	0x0000_00A8
<b>IRQ 27</b>	43	USART0 全局中断	0x0000_00AC
<b>IRQ 28</b>	44	USART1 全局中断	0x0000_00B0

中断编号	向量编号	外设中断描述	向量地址
<b>IRQ 29</b>	45	保留	0x0000_00B4
<b>IRQ 30</b>	46	保留	0x0000_00B8
<b>IRQ 31</b>	47	保留	0x0000_00BC
<b>IRQ 32</b>	48	I2C0 错误中断	0x0000_00C0
<b>IRQ 33</b>	49	保留	0x0000_00C4
<b>IRQ 34</b>	50	I2C1 错误中断	0x0000_00C8
<b>IRQ 35</b>	51	保留	0x0000_00CC
<b>IRQ 36</b>	52	保留	0x0000_00D0
<b>IRQ 37</b>	53	保留	0x0000_00D4
<b>IRQ 38</b>	54	保留	0x0000_00D8
<b>IRQ 39-41</b>	55-57	保留	0x0000_00DC- 0x0000_00E4
<b>IRQ 42</b>	58	保留	0x0000_00E8
<b>IRQ 43-47</b>	59-63	保留	0x0000_00EC- 0x0000_00FC
<b>IRQ 48</b>	64	保留	0x0000_0100
<b>IRQ 49-50</b>	65-66	保留	0x0000_0104- 0x0000_0108
<b>IRQ 51</b>	67	保留	0x0000_010C
<b>IRQ52-66</b>	68-82	保留	0x0000_0110- 0x0000_0148
<b>IRQ67</b>	83	保留	0x0000_014C

## 5.4. 外部中断及事件(EXTI)结构框图

图 5-1. EXTI 结构框图



## 5.5. 外部中断及事件功能概述

EXTI包含21个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。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。

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5根线，包括LVD、RTC、USART和CMP。通过配置SYSCFG模块的SYSCFG\_EXTISSx寄存器，所有的GPIO管脚都可以被选作EXTI的触发源，具体细节请参考[系统配置寄存器 \(SYSCFG\)](#)。

除了中断，EXTI还可以向处理器提供事件信号。Cortex-M23内核完全支持等待中断(WFI)，等待事件(WFE)和发送事件(SEV)指令。唤醒中断控制器(WIC)可以让处理器和NVIC进入功耗极低的省电模式，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。

表 5-3. EXTI 触发源

GD32E230xx 产品

<b>EXTI 线编号</b>	<b>触发源</b>
0	PA0 / PB0 / PF0
1	PA1 / PB1 / PF1
2	PA2 / PB2
3	PA3 / PB3
4	PA4 / PB4
5	PA5 / PB5
6	PA6 / PB6 / PF6
7	PA7 / PB7 / PF7
8	PA8 / PB8
9	PA9 / PB9
10	PA10 / PB10
11	PA11 / PB11
12	PA12 / PB12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC 闹钟
18	保留
19	RTC 干预和时间戳
20	保留
21	CMP 输出
22	保留
23	保留
24	保留
25	USART0 唤醒
26	保留
27	保留

**GD32E231xx 产品**

<b>EXTI 线编号</b>	<b>触发源</b>
0	PA0 / PB0 / PF0
1	PA1 / PB1 / PF1
2	PA2 / PB2
3	PA3 / PB3
4	PA4 / PB4
5	PA5 / PB5
6	PA6 / PB6 / PF6
7	PA7 / PB7 / PF7
8	PA8 / PB8
9	PA9 / PB9
10	PA10 / PB10

EXTI 线编号	触发源
11	PA11 / PB11
12	PA12 / PB12
13	PA13 / PB13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC 闹钟
18	保留
19	RTC 干预和时间戳
20	保留
21	CMP 输出
22	保留
23	保留
24	保留
25	USART0 唤醒
26	保留
27	保留

## 5.6. EXTI 寄存器

EXTI基地址: 0x4001 0400

### 5.6.1. 中断使能寄存器 (EXTI\_INTEN)

地址偏移: 0x00

复位值: 0x0F94 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				INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	INTENx	中断使能位x(x=0..27) 0: 第x线中断被禁止 1: 第x线中断被使能

### 5.6.2. 事件使能寄存器 (EXTI\_EVEN)

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				EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	EVENx	事件使能位x(x=0..27) 0: 第x线事件被禁止

1: 第x线事件被使能

### 5.6.3. 上升沿触发使能寄存器 (**EXTI\_RTEN**)

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					RTEN21	保留	RTEN19	保留	RTEN17	RTEN16
										rw		rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:22	保留	必须保持复位值
21	RTENx	上升沿触发使能( $x=21$ ) 0: 第x线上升沿触发无效 1: 第x线上升沿触发有效（中断/事件请求）
20	保留	必须保持复位值
19	RTENx	上升沿触发使能( $x=19$ ) 0: 第x线上升沿触发无效 1: 第x线上升沿触发有效（中断/事件请求）
18	保留	必须保持复位值
17:0	RTENx	上升沿触发使能( $x=0..17$ ) 0: 第x线上升沿触发无效 1: 第x线上升沿触发有效（中断/事件请求）

### 5.6.4. 下降沿触发使能寄存器 (**EXTI\_FTEN**)

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					FTEN21	保留	FTEN19	保留	FTEN17	FTEN16
										rw		rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0

rw rw

位/位域	名称	描述
31:22	保留	必须保持复位值
21	FTENx	下降沿触发使能(x=21) 0: 第x线下降沿触发无效 1: 第x线下降沿触发有效 (中断/事件请求)
20	保留	必须保持复位值
19	FTENx	下降沿触发使能(x=19) 0: 第x线下降沿触发无效 1: 第x线下降沿触发有效 (中断/事件请求)
18	保留	必须保持复位值
17:0	FTENx	下降沿触发使能(x=0..17) 0: 第x线下降沿触发无效 1: 第x线下降沿触发有效 (中断/事件请求)

### 5.6.5. 软件中断事件寄存器 (EXTI\_SWIEV)

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										SWIEV21	保留	SWIEV19	保留	SWIEV17	SWIEV16
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:22	保留	必须保持复位值
21	SWIEVx	中断/事件软件触发(x=21) 0: 禁用EXTI线x软件中断/事件请求 1: 激活EXTI线x软件中断/事件请求
20	保留	必须保持复位值
19	SWIEVx	中断/事件软件触发(x=19) 0: 禁用EXTI线x软件中断/事件请求 1: 激活EXTI线x软件中断/事件请求

18	保留	必须保持复位值
17:0	SWIEVx	中断/事件软件触发( $x=0..17$ ) 0: 禁用EXTI线x软件中断/事件请求 1: 激活EXTI线x软件中断/事件请求

### 5.6.6. 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										PD21	保留	PD19	保留	PD17	PD16
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
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
rc_w1															

位/位域	名称	描述
31:22	保留	必须保持复位值
21	PDx	中断挂起状态( $x=21$ ) 0: EXTI线x没有被触发 1: EXTI线x被触发, 对这些位写1, 可将其清0
20	保留	必须保持复位值
19	PDx	中断挂起状态( $x=19$ ) 0: EXTI线x没有被触发 1: EXTI线x被触发, 对这些位写1, 可将其清0
18	保留	必须保持复位值
17:0	PDx	中断挂起状态( $x=0..17$ ) 0: EXTI线x没有被触发 1: EXTI线x被触发, 对这些位写1, 可将其清0

## 6. 通用和备用输入/输出接口（GPIO 和 AFIO）

### 6.1. 简介

GD32E230xx 最多可支持 39 个通用 I/O 引脚 (GPIO)，分别为 PA0 ~ PA15, PB0 ~ PB15, PC13 ~ PC15, PF0 ~ PF1, PF6 ~ PF7；GD32E231xx 最多可支持 37 个通用 I/O 引脚 (GPIO)，分别为 PA0 ~ PA15, PB0 ~ PB8, PB10 ~ PB15, PC14 ~ PC15, PF0 ~ PF1, PF6 ~ PF7。各片上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 端口和其他的备用功能 (AFs) 备用引脚，在特定的封装下获得最大的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能引脚，备用功能输入/输出都可以。

每个 GPIO 引脚可以由软件配置为输出（推挽或开漏）、输入、外设备用功能或者模拟模式。每个 GPIO 引脚都可以配置为上拉、下拉或无上拉/下拉。除模拟模式外，所有的 GPIO 引脚都具备大电流驱动能力。

### 6.2. 主要特性

- 输入/输出方向控制；
- 施密特触发输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 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；
- 端口锁定配置；
- 单周期输出翻转功能。

### 6.3. 功能描述

每个通用I/O端口都可以通过32位控制寄存器 (GPIOx\_CTL) 配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出 (GPIO输出或AFIO输出) 时，可以通过GPIO输出模式寄存器 (GPIOx\_OMODE) 配置为推挽或开漏模式。输出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\_PUD) 配置为浮空 (无上拉或下拉)，上拉或下拉功能。

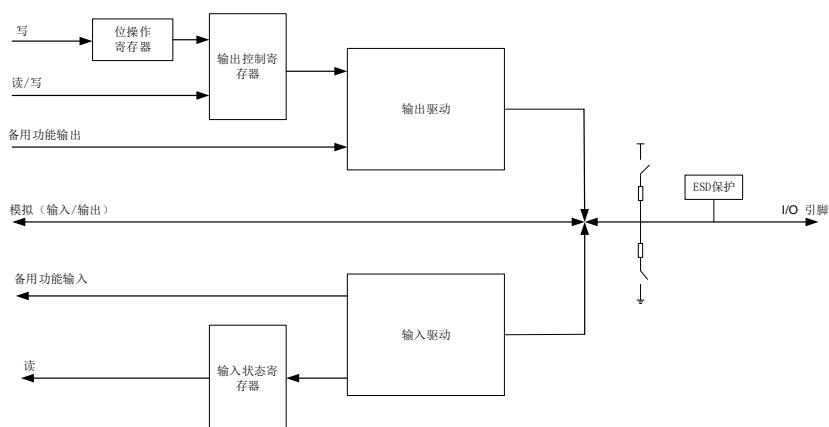
表 6-1. GPIO 配置表

PAD TYPE		CTLn	OMn	PUDn
GPIO	X	悬空	00	X

PAD TYPE			CTLn	OMn	PUDn
输入		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

**图 6-1. GPIO 端口位的基本结构**为标准 I/O 端口位的基本结构图。

**图 6-1. GPIO 端口位的基本结构**



### 6.3.1. GPIO 引脚配置

在复位期间或复位之后，备用功能并未激活，所有 GPIO 端口都被配置成输入浮空模式，这种输入模式禁用上拉(PU)/下拉(PD)电阻。但是复位后，串行线调试为输入 PU/PD 模式。

PA14: SWCLK为AF下拉模式

PA13: SWDIO为AF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AHB时钟周期时都会装载

到端口输入状态寄存器（**GPIOx\_ISTAT**）。

当**GPIO**引脚配置为输出引脚，用户可以配置端口的输出速度和选择输出驱动模式：推挽或开漏模式。端口输出控制寄存器（**GPIOx\_OCTL**）的值将会从相应I/O引脚上输出。

当需要对**GPIOx\_OCTL**进行按位写操作时不需关中断，用户可以通过写‘1’到位操作寄存器（**GPIOx\_BOP**，或用于清0的**GPIOx\_BC**，或用于翻转操作的**GPIOx\_TG**）修改一位或几位，该过程仅需要一个最小的AHB写访问周期，而其他位不受影响。

### 6.3.2. 备用功能（AF）

当端口配置为AFIO（设置**GPIOx\_CTL**寄存器中的**CTLy**值为“0b10”）时，该端口用作外设备功能。通过配置**GPIO**备用功能选择寄存器（**GPIOx\_AFSELy(y=0..1)**），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

### 6.3.3. 附加功能

有些引脚具有附加功能，它们优先于标准**GPIO**寄存器中的配置。当用作ADC附加功能时，引脚必须配置成模拟模式。当引脚用作RTC、WKUPx和振荡器附加功能时，端口类型通过相关的RTC、PMU和RCU寄存器自动设置。当附加功能禁用时，这些端口可用作普通**GPIO**。

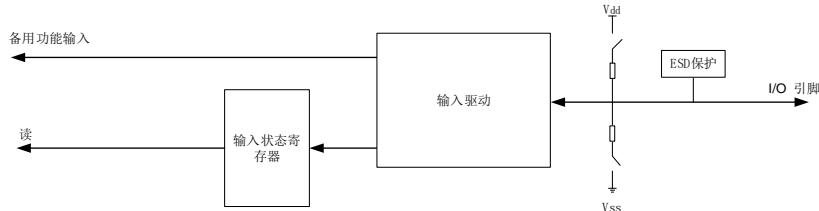
### 6.3.4. 输入配置

当**GPIO**引脚配置为输入时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 当前I/O引脚上的数据在每个AHB时钟周期都会被采样并存入端口输入状态寄存器；
- 输出缓冲器禁用。

**图 6-2. 输入配置的基本结构**显示 I/O 引脚的输入配置。

**图 6-2. 输入配置的基本结构**



### 6.3.5. 输出配置

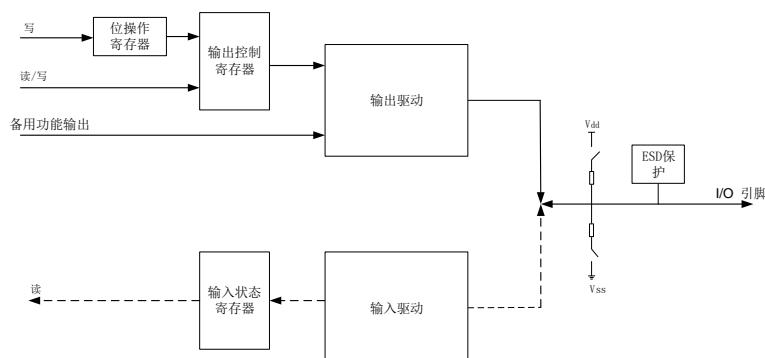
当**GPIO**配置为输出时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 输出缓冲器使能；

- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出“0”；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出“0”；输出控制寄存器设置为“1”，相应引脚输出“1”。
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O的状态。

**图 6-3. 输出配置的基本结构**是 I/O 端口的输出配置。

**图 6-3. 输出配置的基本结构**



### 6.3.6. 模拟配置

当GPIO引脚用于模拟模式时：

- 弱上拉和下拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发输入禁用；
- 读端口输入状态寄存器返回“0”。

**图 6-4. 模拟配置的基本结构**是 I/O 端口的模拟模式配置。

**图 6-4. 模拟配置的基本结构**



### 6.3.7. 备用功能 (AF) 配置

为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

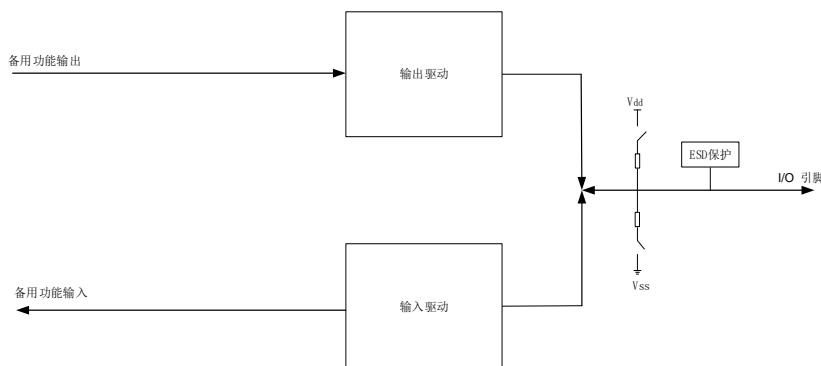
当引脚配置为备用功能时：

- 输出缓冲器启用开漏或者推挽功能；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；

- 可选择的弱上拉/下拉电阻；
- I/O引脚上的数据在每个AHB时钟周期采样并存入端口输入状态寄存器；
- 在开漏模式下对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 在推挽输出模式下对端口输出控制寄存器进行读操作，将返回上次写入的值。

**图 6-5. 备用功能配置的基本结构**是 I/O 端口备用功能配置图。

**图 6-5. 备用功能配置的基本结构**



### 6.3.8. GPIO 锁定功能

GPIO的锁定机制可以保护I/O端口的配置。

被保护的寄存器有：GPIOx\_CTL，GPIOx\_OMODE，GPIOx\_OSPEED，GPIOx\_PUD 和 GPIOx\_AFSEL<sub>y</sub>(y=0..1)。通过配置32位锁定寄存器(GPIOx\_LOCK)可以锁定I/O端口的配置。通过32位锁定寄存器(GPIOx\_LOCK)可以锁定I/O配置。当LOCK序列已经被应用在相应端口位上，直到下一次复位前，不能改变锁定寄存器的值。建议在电源驱动模块驱动的配置时使用锁定功能。

### 6.3.9. GPIO 单周期输出翻转功能

通过将GPIOx\_TG寄存器中对应的位写1，GPIO可以在一个AHB时钟周期内翻转I/O的输出电平。输出信号的频率可以达到AHB时钟的一半。

## 6.4. GPIO 寄存器

GPIOA 基地址: 0x4800 0000

GPIOB 基地址: 0x4800 0400

GPIOC 基地址: 0x4800 0800

GPIOF 基地址: 0x4800 1400

### 6.4.1. 端口控制寄存器 (**GPIOx\_CTL**, x=A..C,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
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]		CTL12[1:0]		CTL11[1:0]		CTL10[1:0]		CTL9[1:0]		CTL8[1:0]			
rw	rw	rw		rw		rw		rw		rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]		CTL4[1:0]		CTL3[1:0]		CTL2[1:0]		CTL1[1:0]		CTL0[1:0]			
rw	rw	rw		rw		rw		rw		rw		rw			

位/位域	名称	描述
31:30	CTL15[1:0]	Pin 15配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
25:24	CTL12[1:0]	Pin 12配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
21:20	CTL10[1:0]	Pin 10配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
19:18	CTL9[1:0]	Pin 9配置位

		该位由软件置位和清除。 参照CTL0[1:0]的描述
17:16	CTL8[1:0]	Pin 8配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
15:14	CTL7[1:0]	Pin 7配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
13:12	CTL6[1:0]	Pin 6配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
11:10	CTL5[1:0]	Pin 5配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
9:8	CTL4[1:0]	Pin 4配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
7:6	CTL3[1:0]	Pin 3配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
5:4	CTL2[1:0]	Pin 2配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
3:2	CTL1[1:0]	Pin 1配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
1:0	CTL0[1:0]	Pin 0配置位 该位由软件置位和清除。 00: GPIO输入模式（复位值） 01: GPIO输出模式 10: 备用功能模式 11: 模拟模式（输入和输出）

#### 6.4.2. 端口输出模式寄存器 (**GPIOx\_OMODE, x=A..C,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

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 该位由软件置位和清除。 参考OM0的描述
14	OM14	Pin 14输出模式位 该位由软件置位和清除。 参考OM0的描述
13	OM13	Pin 13输出模式位 该位由软件置位和清除。 参考OM0的描述
12	OM12	Pin 12输出模式位 该位由软件置位和清除。 参考OM0的描述
11	OM11	Pin 11输出模式位 该位由软件置位和清除。 参考OM0的描述
10	OM10	Pin 10输出模式位 该位由软件置位和清除。 参考OM0的描述
9	OM9	Pin 9输出模式位 该位由软件置位和清除。 参考OM0的描述
8	OM8	Pin 8输出模式位 该位由软件置位和清除。 参考OM0的描述
7	OM7	Pin 7输出模式位 该位由软件置位和清除。 参考OM0的描述
6	OM6	Pin 6输出模式位 该位由软件置位和清除。

## 参考OM0的描述

5	OM5	Pin 5输出模式位 该位由软件置位和清除。 参考OM0的描述
4	OM4	Pin 4输出模式位 该位由软件置位和清除。 参考OM0的描述
3	OM3	Pin 3输出模式位 该位由软件置位和清除。 参考OM0的描述
2	OM2	Pin 2输出模式位 该位由软件置位和清除。 参考OM0的描述
1	OM1	Pin 1输出模式位 该位由软件置位和清除。 参考OM0的描述
0	OM0	Pin 0输出模式位 该位由软件置位和清除。 0: 输出推挽模式（复位值） 1: 输出开漏模式

**6.4.3. 端口输出速度寄存器 (**GPIOx OSPD, x=A..C,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 该位由软件置位和清除。 参考OSPD0[1:0]的描述
29:28	OSPD14[1:0]	Pin 14输出最大速度位 该位由软件置位和清除。

参考OSPD0[1:0]的描述

27:26	OSPD13[1:0]	Pin 13输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
25:24	OSPD12[1:0]	Pin 12输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
23:22	OSPD11[1:0]	Pin 11输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
21:20	OSPD10[1:0]	Pin 10输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
19:18	OSPD9[1:0]	Pin 9输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
17:16	OSPD8[1:0]	Pin 8输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
15:14	OSPD7[1:0]	Pin 7输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
13:12	OSPD6[1:0]	Pin 6输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
11:10	OSPD5[1:0]	Pin 5输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述

---

3:2	OSPD1[1:0]	Pin 1输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
1:0	OSPD0[1:0]	Pin 0输出最大速度位 该位由软件置位和清除。 x0: 输出最大速度2M (复位值) 01: 输出最大速度10M 11: 输出最大速度50M

#### 6.4.4. 端口上拉/下拉寄存器 (**GPIOx\_PUD, x=A..C,F**)

地址偏移: 0x0C

复位值: 端口 A 0x2400 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
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[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
PUD7[1:0]	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[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	PUD15[1:0]	Pin 15上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
29:28	PUD14[1:0]	Pin 14上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
27:26	PUD13[1:0]	Pin 13上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
25:24	PUD12[1:0]	Pin 12上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
23:22	PUD11[1:0]	Pin 11上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
21:20	PUD10[1:0]	Pin 10上拉或下拉位 该位由软件置位和清除。

参照PUDO[1:0]的描述

19:18	PUD9[1:0]	Pin 9上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
17:16	PUD8[1:0]	Pin 8上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
15:14	PUD7[1:0]	Pin 7上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
13:12	PUD6[1:0]	Pin 6上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
11:10	PUD5[1:0]	Pin 5上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
9:8	PUD4[1:0]	Pin 4上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
7:6	PUD3[1:0]	Pin 3上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
5:4	PUD2[1:0]	Pin 2上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
3:2	PUD1[1:0]	Pin 1上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
1:0	PUD0[1:0]	Pin 0上拉或下拉位 该位由软件置位和清除。 00: 悬空模式，无上拉和下拉（复位值） 01: 端口上拉模式 10: 端口下拉模式 11: 保留

#### 6.4.5. 端口输入状态寄存器 (**GPIOx\_ISTAT**, x=A..C,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
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ISTATy	端口输入状态位(y=0..15) 这些位由软件置位和清除。 0: 引脚输入信号为低电平 1: 引脚输入信号为高电平

#### 6.4.6. 端口输出控制寄存器 (GPIOx\_OCTL, x=A..C,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
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	OCTLy	端口输出控制位(y=0..15) 该位由软件置位和清除。 0: 引脚输出低电平 1: 引脚输出高电平

#### 6.4.7. 端口位操作寄存器 (GPIOx\_BOP, x=A..C,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
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
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	CRy	端口清除位y(y=0..15) 该位由软件置位和清除。 0: 相应的OCTLy位没有改变 1: 清除相应的OCTLy位为0
15:0	BOPy[15:0]	端口置位位y(y=0..15) 该位由软件置位和清除。 0: 相应的OCTLy位没有改变 1: 设置相应的OCTLy位为1

#### 6.4.8. 端口配置锁定寄存器 (GPIOx\_LOCK, x=A,B)

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
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	描述
31:17	保留	必须保持复位值
16	LKK	锁定键 该位只能通过Lock Key写序列置位，始终可读。 0: GPIOx_LOCK寄存器和端口配置没有锁定 1: 直到下一次MCU复位前，GPIOx_LOCK寄存器被锁定 LOCK key写序列： 写1→写0→写1→读0→读1 <b>注意：</b> 在LOCK Key写序列期间，LK y(y=0..15)的值必须保持。
15:0	LKy	端口锁定位y(y=0..15)

该位由软件置位和清除。

0: 端口配置没有锁定

1: 端口配置锁定

#### 6.4.9. 备用功能选择寄存器 0 (GPIOx\_AFSEL0, 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
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL7[3:0]	Pin 7选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
27:24	SEL6[3:0]	Pin 6选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
23:20	SEL5[3:0]	Pin 5选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
19:16	SEL4[3:0]	Pin 4选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0选择备用功能

该位由软件置位和清除。

0000: 选择AF0功能 (复位值)

0001: 选择AF1功能

0010: 选择AF2功能

0011: 选择AF3功能

0100: 选择AF4功能 (Port A,B only)

0101: 选择AF5功能 (Port A,B only)

0110: 选择AF6功能 (Port A,B only)

0111: 选择AF7功能 (Port A,B only)

1000 ~ 1111: 保留

#### **6.4.10. 备用功能选择寄存器 1 (GPIOx\_AFSEL1, 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
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			rw

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
27:24	SEL14[3:0]	Pin 14选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 1选择备用功能 该位由软件置位和清除。

参照SEL8[3:0]的描述

11:8	<b>SEL10[3:0]</b>	Pin 10选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
7:4	<b>SEL9[3:0]</b>	Pin 9选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
3:0	<b>SEL8[3:0]</b>	Pin 8选择备用功能 该位由软件置位和清除。 0000: 选择AF0功能 (复位值) 0001: 选择AF1功能 0010: 选择AF2功能 0011: 选择AF3功能 0100: 选择AF4功能 (Port A,B only) 0101: 选择AF5功能 (Port A,B only) 0110: 选择AF6功能 (Port A,B only) 0111: 选择AF7功能 (Port A,B only)
		1000 ~ 1111: 保留

#### 6.4.11. 位清除寄存器 (**GPIOx\_BC**, x=A..C,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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	<b>CRy</b>	端口清除位(y=0..15) 该位由软件置位和清除。 0: 相应OCTLy位没有改变 1: 清除相应的OCTLy位

### 6.4.12. 端口位翻转寄存器 (**GPIOx\_TG**, **x=A..C,F**)

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TGy	端口翻转位y(y=0..15) 该位由软件置位和清除。 0: 相应OCTLy位没有改变 1: 翻转相应的OCTLy位

## 7. 循环冗余校验管理单元 (CRC)

### 7.1. 简介

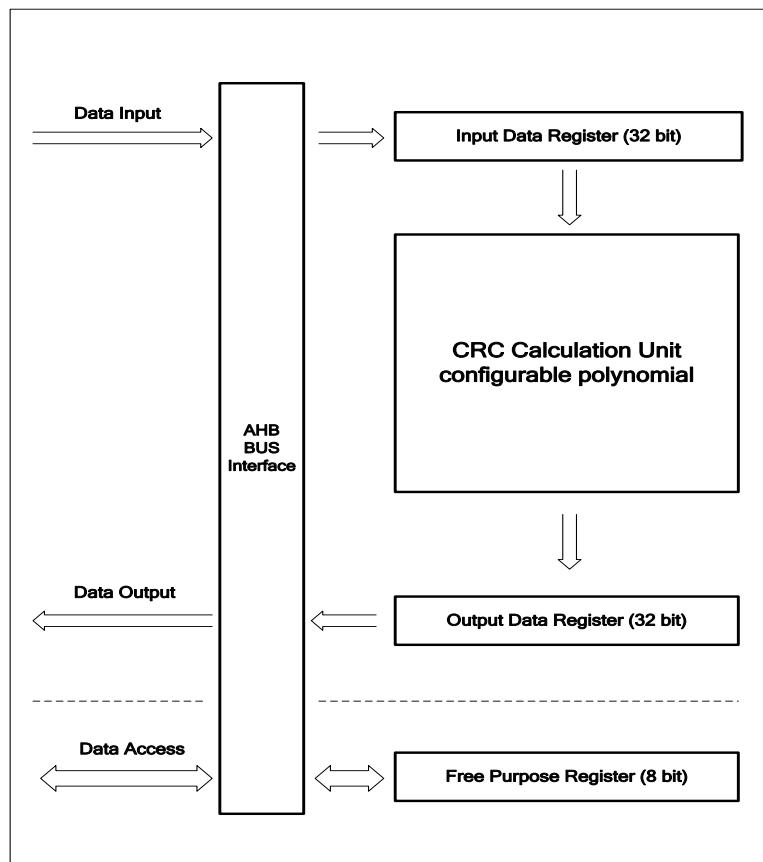
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 管理单元能用用户配置的多项式来计算 7/8/16/32 位的 CRC 校验码。

### 7.2. 主要特性

- 支持7/8/16/32位数据输入；
- 对于7(8)/16/32位的输入数据长度，计算周期分别为1/2/4个AHB时钟周期；
- 用户可以配置多项式长度；
- CRC复位后，用户可以配置计算初值；
- 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；
- 用户可设置的计算初值，可使计算更灵活。

图 7-1. CRC 管理单元框图



### 7.3. 功能描述

- CRC管理单元可以用来计算32位的原始数据，**CRC\_DATA**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\_CTL**寄存器的方式来清除**CRC\_DATA**寄存器，新输入的原始数据将会基于前一次**CRC\_DATA**寄存器中的结果进行计算；  
对于32/16/8(7)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\_FDATA**，**CRC\_FDATA**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0x3456CDEF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：0x2C6AB3F7
  - 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：0x6A2CF7B3
  - 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：0xF7B36A2C  
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0x3344CCDD将被逆序成0xBB3322CC。
- 用户可配置的初始计算数据。  
当RST位置位或对**CRC\_IDATA**寄存器进行写操作时，**CRC\_DATA**寄存器将自动初始化为**CRC\_IDATA**寄存器中的值。
- 用户配置多项式。  
通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。当PS[1:0]或多项式改变后，需要复位CRC。

## 7.4. CRC 寄存器

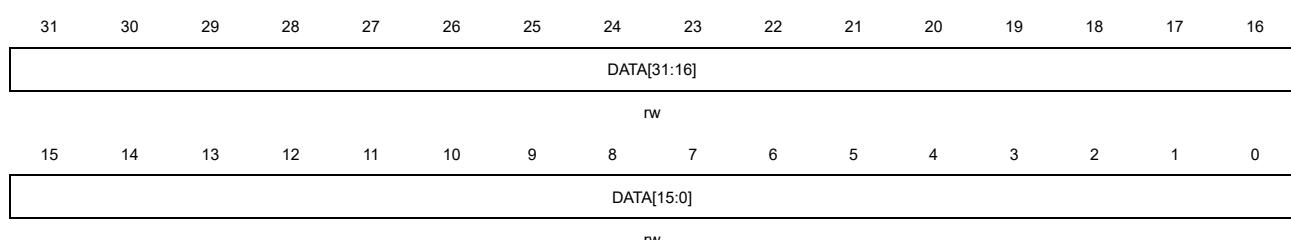
CRC基地址: 0x4002 3000

### 7.4.1. 数据寄存器 (CRC\_DATA)

地址偏移: 0x00

复位值: 0xFFFF FFFF

该寄存器只能按字(32位)访问



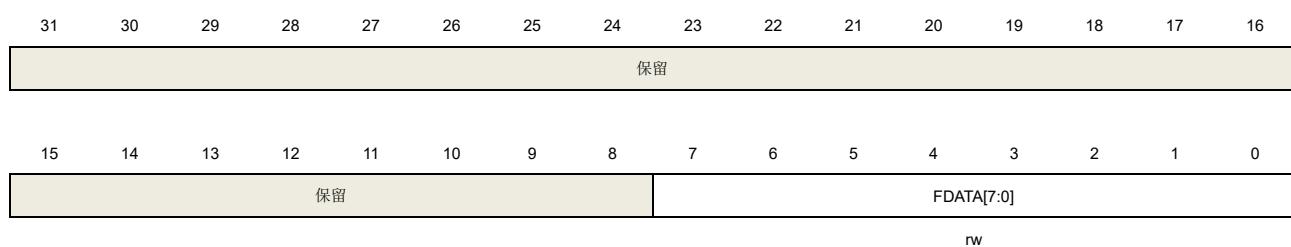
位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次CRC计算的结果。

### 7.4.2. 独立数据寄存器 (CRC\_FDATA)

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
31:8	保留	必须保持复位值
7:0	FDATA[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\_CTL寄存器的影响。

### 7.4.3. 控制寄存器 (CRC\_CTL)

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]		PS[1:0]		保留		RST			
				rw		rw		rw		rw		rs			

位/位域	名称	描述
31:8	保留	必须保持复位值
7	REV_O	按位顺序翻转输出数据功能 0: 输出数据不翻转 1: 输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0: 输入数据不翻转 1: 输入数据按字节翻转 2: 输入数据按半字翻转 3: 输入数据按字翻转
4:3	PS[1:0]	多项式长度 0: 32位 1: 16 (POLY[15:0]用于计数)位 2: 8 (POLY[7:0]用于计数)位 3: 7 (POLY[6:0]用于计数)位
2:1	保留	必须保持复位值
0	RST	该位用来复位CRC_DATA寄存器，CRC_DATA寄存器将自动初始化为CRC_IDATA寄存器中的值，然后自动清零。该位对CRC_FDATA寄存器没有影响。 软件可读写

### 7.4.4. 初值寄存器 (CRC\_IDATA)

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
IDATA [31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDATA[15:0]															
rw															

位/位域	名称	描述
31:0	IDATA[31:0]	配置CRC初值 CRC_CTL寄存器的RST位置位后，CRC_DATA寄存器的值将被更新为此寄存器的值。

#### 7.4.5. 多项式寄存器 (CRC\_POLY)

地址偏移: 0x14

复位值: 0x04C1 1DB7

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
POLY [31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLY[15:0]															
rw															

位/位域	名称	描述
31:0	POLY[31:0]	配置多项式值 配合PS[1: 0]使用。

## 8. 直接存储器访问控制器（DMA）

### 8.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 5 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 DMA 请求的优先级。

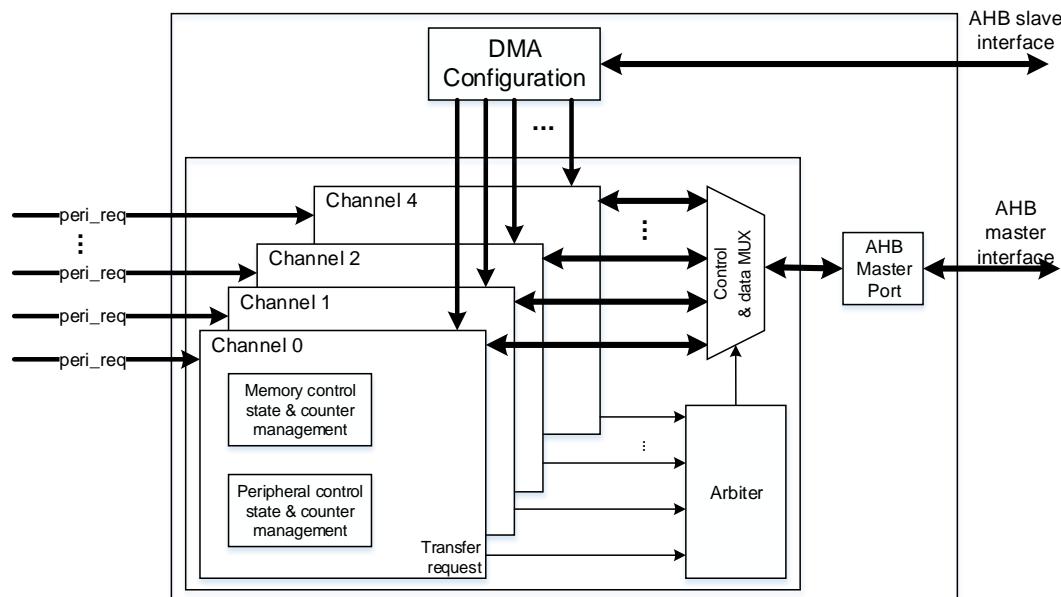
DMA 控制器和 Cortex®-M23 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

### 8.2. 主要特性

- 传输数据长度可编程配置，最大到 65536；
- 5 个通道，并且每个通道都可配置；
- AHB 和 APB 外设，片上闪存和 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；
- 支持软件优先级（低、中、高、极高）和硬件优先级（通道号越低，优先级越高）；
- 存储器和外设的数据传输宽度可配置：字节，半字，字；
- 存储器和外设的数据传输支持固定寻址和增量式寻址；
- 支持循环传输模式；
- 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；
- 每个通道有 3 种类型的事件标志和独立的中断，支持中断的使能和清除；
- 支持中断使能和清除。

## 8.3. 结构框图

图 8-1. DMA 结构框图



由 [图 8-1. DMA 结构框图](#) 所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA
- AHB 主接口进行数据传输
- 仲裁器进行 DMA 请求的优先级管理
- 数据处理和计数

## 8.4. 功能描述

### 8.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA\_CHxPADDR、DMA\_CHxMADDR、DMA\_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA\_CHxCNT 寄存器用于控制传输的次数。DMA\_CHxCTL 寄存器的 PWIDHT 和 MWIDHT 位域决定每次发送和接收的字节数（字节/半字/字）。

假设 DMA\_CHxCNT 寄存器的值为 4，并且 PNAGA 和 MNAGA 位均置位。结合 PWIDHT 和 MWIDHT 的各种配置，DMA 传输的操作详见[表 8-1. DMA 传输操作](#)。

**表 8-1. DMA 传输操作**

传输宽度		传输操作	
源	目标	源	目标
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[7:0] @0x0 2: Write B5B4[7:0] @0x2 3: Write B9B8[7:0] @0x4 4: Write BDBC[7:0] @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

DMA\_CHxCNT 寄存器的 CNT 位域必须在 CHEN 位置位前被配置，其控制传输的次数。在传输过程中，CNT 位域的值表示还有多少次数据传输将被执行。

将 DMA\_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输。

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
  - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输。
  - 在重新使能 DMA 通道前，对任意相关寄存器进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

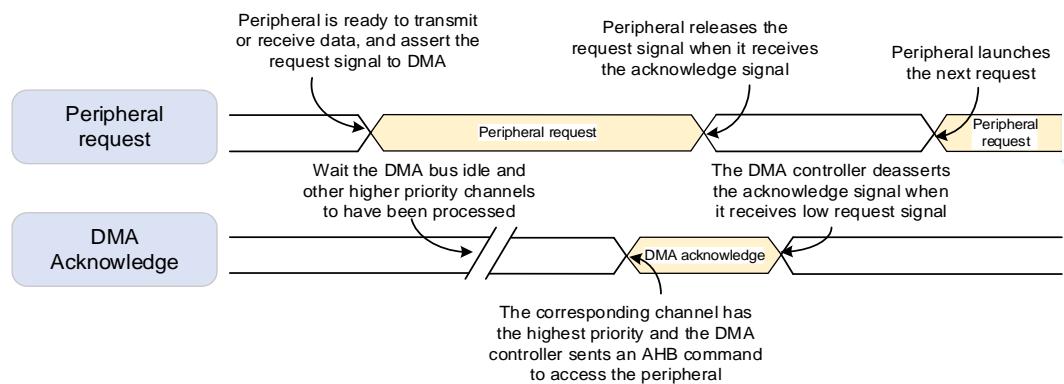
### 8.4.2. 外设握手

为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应答信号：

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；
- 应答信号：由 DMA 控制器响应，表明 DMA 控制器已经发送 AHB 命令去访问外设。

[图8-2. 握手机制](#)中详细描述了DMA控制器与外设之间的握手机制。

图 8-2. 握手机制



### 8.4.3. 仲裁

当DMA控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级，优先级规则如下：

- 软件优先级：分为4级，低，中，高和极高。可以通过寄存器DMA\_CHxCTL的PRIO位域来配置。
- 硬件优先级：当通道具有相同的软件优先级时，编号低的通道优先级高。例：通道0和通道2配置为相同的软件优先级时，通道0的优先级高于通道2。

### 8.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器DMA\_CHxCTL的PNAGA和MNAGA位用来设置存储器和外设的地址生成算法。

在固定模式中，地址一直固定为初始化的基地址（DMA\_CHxPADDR, DMA\_CHxMADDR）。

在增量模式中，下一次传输数据的地址是当前地址加1（或者2, 4），这个值取决于数据传输宽度。

### 8.4.5. 循环模式

循环模式用来处理连续的外设请求(如ADC扫描模式)。将DMA\_CHxCTL寄存器的CMEN位置

位可以使能循环模式。

在循环模式中，当每次DMA传输完成后，CNT值会被重新载入，且传输完成标志位会被置1。DMA会一直响应外设的请求，直到通道使能位（DMA\_CHxCTL寄存器的CHEN位）被清0。

#### **8.4.6. 存储器到存储器模式**

将DMA\_CHxCTL寄存器的M2M位置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\_CHxCTL寄存器的CHEN位被置1，DMA通道就立即开始传输数据，直到DMA\_CHxCNT寄存器达到0，DMA通道才会停止。

#### **8.4.7. 通道配置**

要启动一次新的DMA数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，如果为 1（通道已使能），清零该位。当 CHEN 为 0 时，请按照下列步骤配置 DMA 开始新的传输。
2. 配置 DMA\_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式。
3. 配置 DMA\_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式。
4. 配置 DMA\_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级。
5. 通过 DMA\_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法。
6. 通过 DMA\_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位。
7. 通过 DMA\_CHxPADDR 寄存器配置外设地址。
8. 通过 DMA\_CHxMADDR 寄存器配置存储器地址。
9. 通过 DMA\_CHxCNT 寄存器配置数据传输总量。
10. 将 DMA\_CHxCTL 寄存器的 CHEN 置位 1，使能 DMA 通道。

#### **8.4.8. 中断**

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，半传输完成和传输错误。

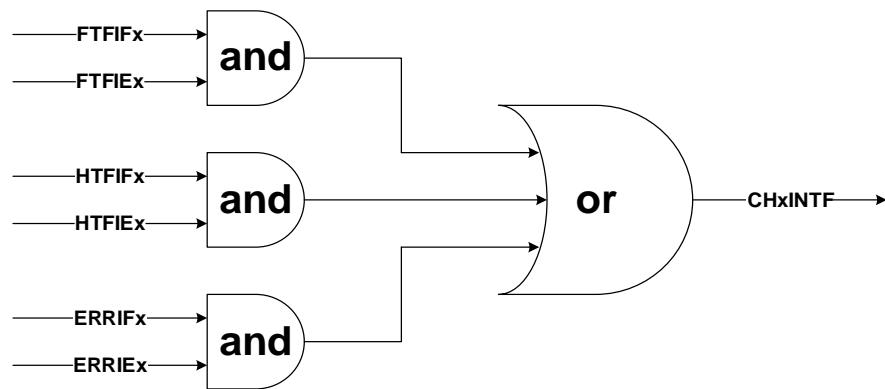
每一个中断事件在DMA\_INTF寄存器中有专用的标志位，在DMA\_INTC寄存器中有专用的清除位，在DMA\_CHxCTL寄存器中有专用的使能位。[表8-2. 中断事件](#)描述了其对应关系。

**表 8-2. 中断事件**

中断事件	标志位	清除位	使能位
	DMA_INTF	DMA_INTC	DMA_CHxCTL
传输完成	FTFIF	FTFIFC	FTFIE
传输半完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如[图8-3. DMA中断逻辑图](#)所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

**图 8-3. DMA 中断逻辑图**



注意：“x”表示通道数（对应x=0...4）

#### 8.4.9. DMA 请求映射

多个外设请求被映射到同一个 DMA 通道。这些请求信号在经过逻辑或后进入 DMA。详情可见[图8-4. DMA 请求映射](#)。通过配置对应外设的寄存器，每个外设的请求均可以独立的开启或关闭。用户必须确保同一时间，在同一个通道上仅有 一个外设的请求被开启。[表8-3. DMA 各通道请求表](#)列举了 DMA 的每个通道所支持的外设请求。

图 8-4. DMA 请求映射

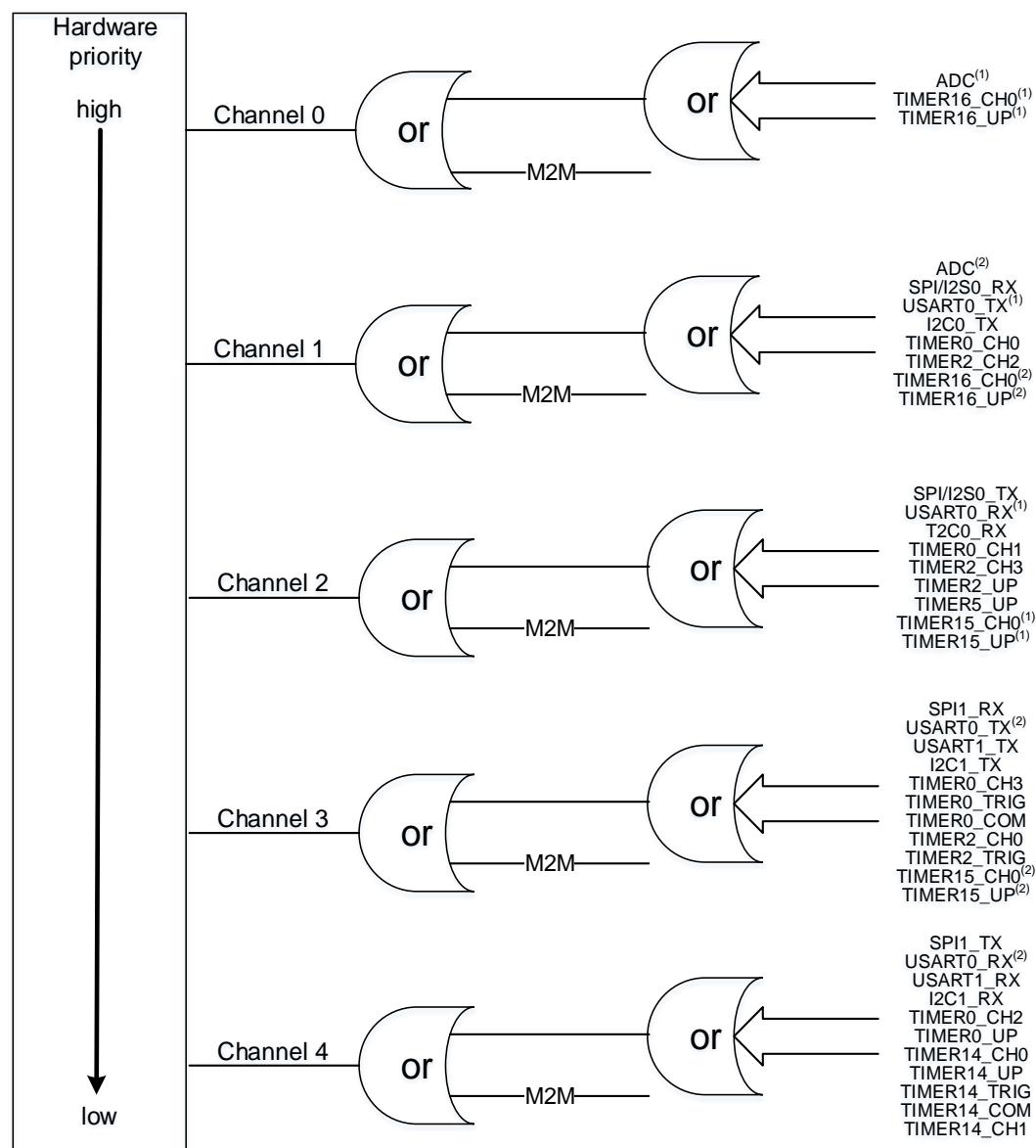


表8-3. DMA各通道请求表

外设	通道 0	通道 1	通道 2	通道 3	通道 4
<b>ADC</b>	ADC <sup>(1)</sup>	ADC <sup>(2)</sup>	•	•	•
<b>SPI/I2S</b>	•	SPI/I2S0_RX	SPI/I2S0_TX	SPI1_RX	SPI1_TX
<b>USART</b>	•	USART0_TX <sup>(1)</sup>	USART0_RX <sup>(1)</sup>	USART0_TX <sup>(2)</sup> USART1_TX	USART0_RX <sup>(2)</sup> USART1_RX
<b>I2C</b>	•	I2C0_TX	I2C0_RX	I2C1_TX	I2C1_RX
<b>TIMER0</b>	•	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TRIG TIMER0_COM	TIMER0_CH2 TIMER0_UP
<b>TIMER2</b>	•	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	TIMER2_CH0 TIMER2_TRIG	•
<b>TIMER5</b>	•	•	TIMER5_UP	•	•
<b>TIMER14</b>	•	•	•	•	TIMER14_CH0 TIMER14_UP TIMER14_TRIG TIMER14_COMM TIMER14_CH1
<b>TIMER15</b>	•	•	TIMER15_CH0 <sup>(1)</sup> TIMER15_UP <sup>(1)</sup>	TIMER15_CH0 <sup>(2)</sup> TIMER15_UP <sup>(2)</sup>	•
<b>TIMER16</b>	TIMER16_CH0 <sup>(1)</sup> TIMER16_UP <sup>(1)</sup>	TIMER16_CH0 <sup>(2)</sup> TIMER16_UP <sup>(2)</sup>	•	•	•

(1). 当 SYSCFG\_CFG0 寄存器的相应重映射位被清零时，请求被映射到该通道。

(2). 当 SYSCFG\_CFG0 寄存器的相应重映射位被置位时，请求被映射到该通道。

## 8.5. DMA 寄存器

DMA 基址: 0x4002 0000

### 8.5.1. 中断标志位寄存器 (DMA\_INTF)

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														ERRIF4	HTFIF4	FTFIF4	GIF4
														r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIFO	HTFIFO	FTFIFO	GIFO		
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r		

位/位域	名称	描述
31:20	保留	必须保持复位值
19/15/11/7/3	ERRIFx	通道x错误标志位(x=0...4) 硬件置位, 软件写DMA_INTC相应位为1清零 0: 通道x未发生传输错误 1: 通道x发生传输错误
18/14/10/6/2	HTFIFx	通道x半传输完成标志位(x=0...4) 硬件置位, 软件写DMA_INTC相应位为1清零 0: 通道x半传输未完成 1: 通道x半传输完成
17/13/9/5/1	FTFIFx	通道x传输完成标志位(x=0...4) 硬件置位, 软件写DMA_INTC相应位为1清零 0: 通道x传输未完成 1: 通道x传输完成
16/12/8/4/0	GIFx	通道x全局中断标志位(x=0...4) 硬件置位, 软件写DMA_INTC相应位为1清零 0: 通道x ERRIF, HTFIF或FTFIF标志位未置位 1: 通道x至少发生ERRIF, HTFIF或FTFIF之一置位

### 8.5.2. 中断标志位清除寄存器 (DMA\_INTC)

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												ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
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
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIFC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:20	保留	必须保持复位值
19/15/11/7/3	ERRIFCx	清除通道x(x=0...4)的错误标志位 0: 无影响 1: 清零DMA_INTF寄存器的ERRIFx位
18/14/10/6/2	HTFIFCx	清除通道x(x=0...4)的半传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的HTFIFx位
17/13/9/5/1	FTFIFCx	清除通道x(x=0...4)的传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的FTFIFx位
16/12/8/4/0	GIFCx	清除通道x(x=0...4)的全局中断标志位 0: 无影响 1: 清零DMA_INTF寄存器的GIFx, ERRIFx, HTFIFx和FTFIFx位

### 8.5.3. 通道 x 控制寄存器 (DMA\_CHxCTL)

x = 0...4, x 为通道序号

地址偏移: 0x08 + 0x14 × x

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	M2M	PRIO[1:0]	MWIDTH[1:0]	PWIDTH[1:0]	MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN			
rw	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	M2M	存储器到存储器模式 软件置位和清零

---

		0: 禁止存储器到存储器模式 1: 使能存储器到存储器模式 <b>CHEN位为1时，该位不能被配置</b>
13:12	<b>PRIO[1:0]</b>	软件优先级 软件置位和清零 00: 低 01: 中 10: 高 11: 极高 <b>CHEN位为1时，该位域不能被配置</b>
11:10	<b>MWIDTH[1:0]</b>	存储器的传输数据宽度 软件置位和清零 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 <b>CHEN位为1时，该位域不能被配置</b>
9:8	<b>PWIDTH[1:0]</b>	外设的传输数据宽度 软件置位和清零 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 <b>CHEN位为1时，该位域不能被配置</b>
7	<b>MNAGA</b>	存储器的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式 <b>CHEN位为1时，该位不能被配置</b>
6	<b>PNAGA</b>	外设的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式 <b>CHEN位为1时，该位不能被配置</b>
5	<b>CMEN</b>	循环模式使能 软件置位和清零 0: 禁止循环模式 1: 使能循环模式 <b>CHEN位为1时，该位不能被配置</b>
4	<b>DIR</b>	传输方向

		软件置位和清零
0:	从外设读出并写入存储器	
1:	从存储器读出并写入外设	
	CHEN位为1时，该位不能被配置	
3	<b>ERRIE</b>	通道错误中断使能位 软件置位和清零 0: 禁止通道错误中断 1: 使能通道错误中断
2	<b>HTFIE</b>	通道半传输完成中断使能位 软件置位和清零 0: 禁止通道半传输完成中断 1: 使能通道半传输完成中断
1	<b>FTFIE</b>	通道传输完成中断使能位 软件置位和清零 0: 禁止通道传输完成中断 1: 使能通道传输完成中断
0	<b>CHEN</b>	通道使能 软件置位和清零 0: 禁止该通道 1: 使能该通道

#### 8.5.4. 通道 x 计数寄存器 (**DMA\_CHxCNT**)

**x = 0...4, x 为通道序号**

地址偏移: **0x0C + 0x14 × x**

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
15:0	CNT[15:0]	传输计数 CHEN位为1时，该位域不能被配置 该寄存器标明还有多少数据等待被传输。一旦通道使能，该寄存器为只读的，并在每个DMA传输之后值减1。如果该寄存器的值为0，无论通道开启与否，都不会有数据

传输。如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会被自动重装载为初始设置值。

### 8.5.5. 通道 x 外设基地址寄存器 (DMA\_CHxPADDR)

$x = 0 \dots 4$ ,  $x$  为通道序号

地址偏移:  $0x10 + 0x14 \times x$

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
PADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PADDR[15:0]															
rw															

位/位域	名称	描述
31:0	PADDR[31:0]	外设地址 CHEN位为1时，该位域不能被配置 当PWIDTH位域的值为01 (16-bit),, PADDR[0]被忽略，访问自动与16位地址对齐。 当PWIDTH位域的值为10 (32-bit), PADDR [1:0]被忽略，访问自动与32位地址对齐。

### 8.5.6. 通道 x 存储器基地址寄存器 (DMA\_CHxMADDR)

$x = 0 \dots 4$ ,  $x$  为通道序号

地址偏移:  $0x14 + 0x14 \times x$

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
MADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADDR[15:0]															
rw															

位/位域	名称	描述
31:0	MADDR[31:0]	存储器基地址 CHEN位为1时，该位域不能被配置 当MWIDTH位域的值为01 (16-bit)时，MADDR [0]被忽略，访问自动与16位地址对齐。 当MWIDTH位域的值为10 (32-bit)时，MADDR [1:0]被忽略，访问自动与32位地址对齐

齐。

## 9. 调试 (DBG)

### 9.1. 简介

GD32E23x系列产品提供了各种各样的调试，跟踪和测试功能。这些功能通过Arm CoreSight™组件的标准配置和链状连接的TAP控制器来实现的。调试和跟踪功能集成在Arm Cortex-M23内核中。调试系统支持串行调试（SWD）和跟踪功能。调试和跟踪功能请参考下列文档：

- Cortex-M23技术参考手册；
- Arm调试接口v5结构规范。

调试系统帮助调试者在低功耗模式下调试以及一些外设调试，包括：TIMER、I2C、RTC、WWDGT和FWDGT。当相应的位被置1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、I2C、RTC、WWDGT和FWDGT。

### 9.2. 串行调试接口简介

调试工具可以通过串行调试接口（SWD）来访问调试功能。

#### 9.2.1. 引脚分配

串行调试（SWD）提供两个引脚的接口：数据输入输出引脚（SWDIO）和时钟引脚（SWCLK）。

调试引脚分配：

PA14 : SWCLK

PA13 : SWDIO

如果SWD没有使用，这两个引脚均释放作为普通GPIO功能。两个引脚具体配置请参考[通用和备用输入输出接口（GPIO和AFIO）](#)。

### 9.3. 调试保持功能描述

#### 9.3.1. 低功耗模式调试支持

当DBG控制寄存器0（DBG\_CTL0）的STB\_HOLD位置1并且进入待机模式，AHB总线时钟和系统时钟由CK\_IRC8M提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当DBG控制寄存器0（DBG\_CTL0）的DSLP\_HOLD位置1并且进入深度睡眠模式，AHB总线时钟和系统时钟由CK\_IRC8M提供，可以在深度睡眠模式下调试。

当DBG控制寄存器0（DBG\_CTL0）的SLP\_HOLD位置1并且进入睡眠模式，AHB总线时钟没有关闭，可以在睡眠模式下调试。

### 9.3.2. TIMER, I2C, RTC, WWDGT 和 FWDGT 外设调试支持

当内核停止，并且DBG控制寄存器0（DBG\_CTL0）或DBG控制寄存器1（DBG\_CTL1）中的相应位置1。对于不同外设，有不同动作：

对于**TIMER**外设，**TIMER**计数器停止并进行调试；

对于**I2C**外设，**SMBUS**保持状态并进行调试；

对于**RTC**外设，计数器停止并进行调试；

对于**WWDGT**或者**FWDGT**外设，计数器时钟停止并进行调试。

## 9.4. DBG 寄存器

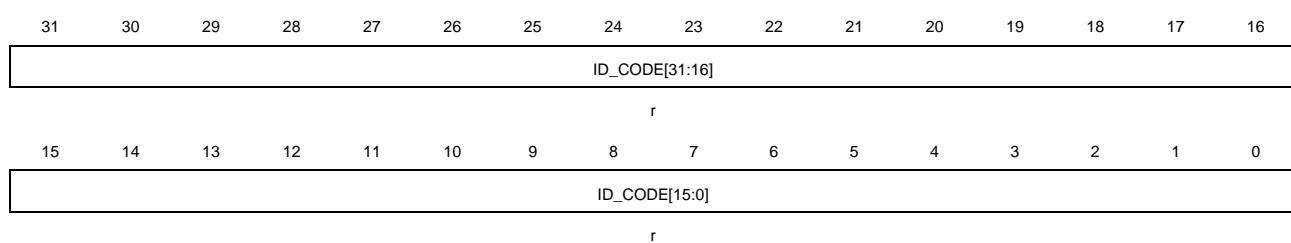
DBG 基地址: 0x4001 5800

### 9.4.1. ID 寄存器 (DBG\_ID)

地址偏移: 0x00

只读寄存器

该寄存器只能按字(32位)访问



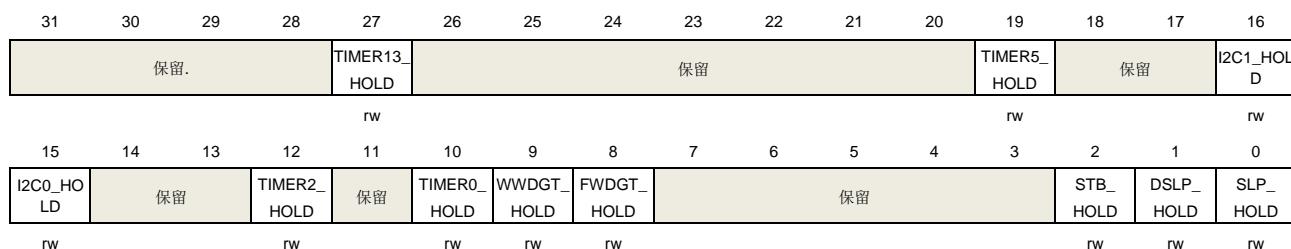
位/位域	名称	描述
31:0	ID_CODE[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数

### 9.4.2. 控制寄存器 0 (DBG\_CTL0)

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	TIMER13_HOLD	TIMER 13 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 13 计数器不变, 用于调试
26:20	保留	必须保持复位值

19	TIMER5_HOLD	<p><b>TIMER 5 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 TIMER 5 计数器不变, 用于调试</p>
18:17	保留	必须保持复位值
16	I2C1_HOLD	<p><b>I2C1 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 I2C1 的 SMBUS 状态不变, 用于调试</p>
15	I2C0_HOLD	<p><b>I2C0 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 I2C0 的 SMBUS 状态不变, 用于调试</p>
14:13	保留	必须保持复位值
12	TIMER2_HOLD	<p><b>TIMER 2 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 TIMER 2 计数器不变, 用于调试</p>
11	保留	必须保持复位值
10	TIMER0_HOLD	<p><b>TIMER 0 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 TIMER 0 计数器不变, 用于调试</p>
9	WWDGT_HOLD	<p><b>WWDGT 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 WWDGT 计数器时钟, 用于调试</p>
8	FWDGT_HOLD	<p><b>FWDGT 保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 当内核停止时保持 FWDGT 计数器时钟, 用于调试</p>
7:3	保留	必须保持复位值
2	STB_HOLD	<p><b>待机模式保持位</b></p> <p>该位由软件置位和复位</p> <p>0: 无影响</p> <p>1: 在待机模式下系统时钟和 AHB 时钟由 CK_IRC8M 提供, 当退出待机模式时, 产生系统复位</p>
1	DSLP_HOLD	深度睡眠模式保持位

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 AHB 时钟由 CK\_IRC8M 提供, 当退出待机模式时, 产生系统复位

0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位 0: 无影响 1: 在睡眠模式下, AHB 时钟继续运行
---	----------	--

### 9.4.3. 控制寄存器 1 (DBG\_CTL1)

地址偏移: 0x08

复位值: 0x0000 0000, 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.															
													TIMER16	TIMER15	TIMER14
													_HOLD	_HOLD	_HOLD
rw rw rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留							
rw															

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER16_HOLD	<p><b>TIMER 16 保持位</b> 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 16 计数器不变, 用于调试</p>
17	TIMER15_HOLD	<p><b>TIMER 15 保持位</b> 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 15 计数器不变, 用于调试</p>
16	TIMER14_HOLD	<p><b>TIMER 14 保持位</b> 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 14 计数器不变, 用于调试</p>
15:11	保留	必须保持复位值
10	RTC_HOLD	<p><b>RTC 保持位</b> 该位由软件置位和复位</p>

0: 无影响

1: 当内核停止时保持 RTC 计数器不变，用于调试

9:0

保留

必须保持复位值

## 10. 模数转换器(ADC)

### 10.1. 简介

MCU片上集成了12位逐次逼近式模数转换器模块（ADC），可以采样来自于10个外部通道、2个内部通道的模拟信号。这12个ADC采样通道都支持多种运行模式，采样转换后，转换结果可以按照最低有效位对齐或最高有效位对齐的方式保存在相应的数据寄存器中。片上的硬件过采样机制可以通过减少MCU的相关计算负担来提高性能。

### 10.2. 主要特征

- 高性能：
  - ADC 采样分辨率：12 位、10 位、8 位或 6 位分辨率；
  - 前置校准功能；
  - 可编程的采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - DMA 请求。
- 双时钟域架构（APB 时钟和 ADC 时钟）。
- 模拟输入通道：
  - 10 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 ( $V_{REFINT}$ )。
- 转换开始的发起：
  - 软件方式；
  - 硬件触发。
- 运行模式：
  - 转换单个通道，或者扫描一组通道；
  - 单次运行模式，每次触发转换一次选择的输入通道；
  - 连续运行模式，连续转换所选择的输入通道；
  - 间断运行模式。
- 中断的产生：
  - 常规序列转换结束；
  - 模拟看门狗事件。
- 转换结果阈值监测器功能：模拟看门狗。
- 模块供电要求：2.4V 到 3.6V，一般电源电压为 3.3V。
- 过采样：
  - 16 位的数据寄存器；
  - 可调整的过采样率，范围从 2x 到 256x；
  - 高达 8 位的可编程数据移位。
- 通道输入范围： $V_{SSA} \leq V_{IN} \leq V_{DDA}$ 。

### 10.3. 引脚和内部信号

[图 10-1. ADC 模块框图](#)给出了 ADC 模块框图。[表 10-1. ADC 内部输入信号](#)和[表 10-2. ADC 输入引脚定义](#)给出了 ADC 内部信号和引脚定义。

**表 10-1. ADC 内部输入信号**

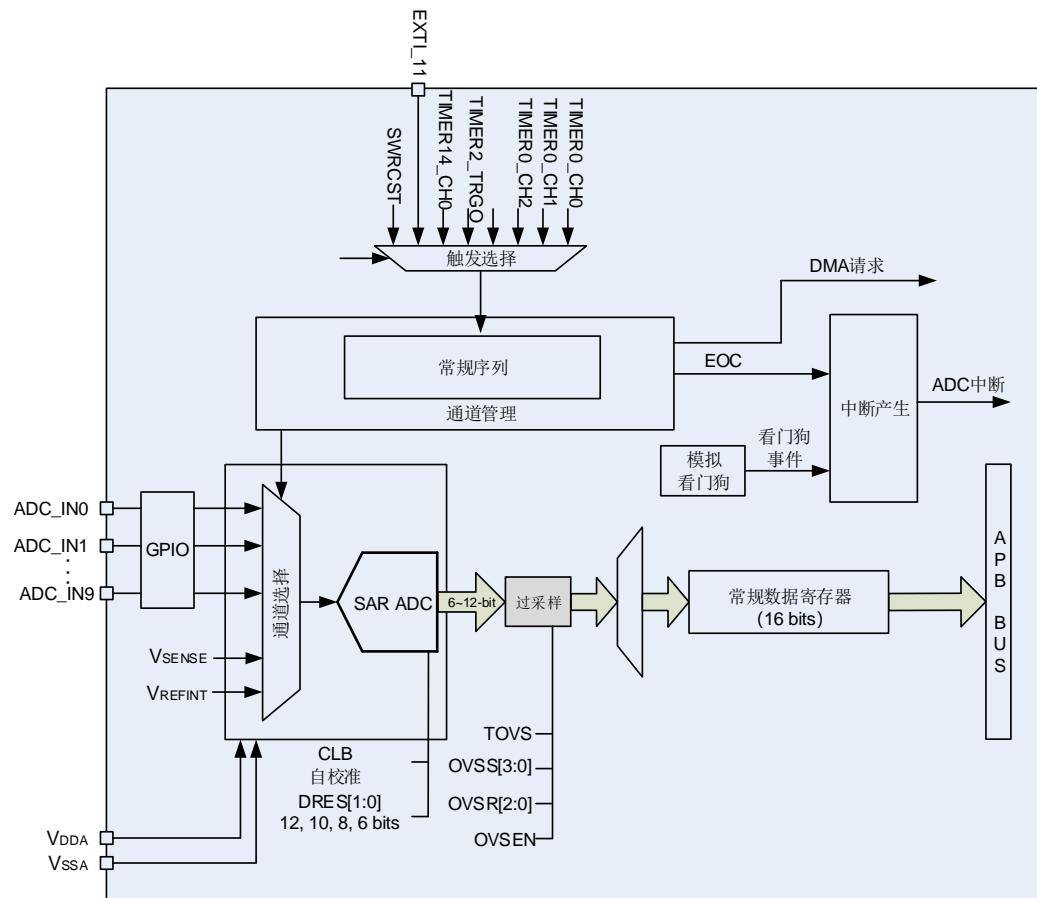
内部信号名称	说明
V <sub>SENSE</sub>	内部温度传感器电压输出
V <sub>REFINT</sub>	内部参考电压输出

**表 10-2. ADC 输入引脚定义**

名称	说明
V <sub>DDA</sub>	模拟电源输入等于 V <sub>DD</sub> , 2.4 V ≤ V <sub>DDA</sub> ≤ 3.6 V
V <sub>SSA</sub>	模拟地, 等于 V <sub>SS</sub>
ADCx_IN [9:0]	多达 10 路外部通道

## 10.4. 功能说明

图 10-1. ADC 模块框图



### 10.4.1. 前置校准功能

在前置校准期间，ADC计算一个校准因子，这个因子应用于ADC的内部，它直到ADC下次掉电才无效。在校准期间，应用程序不能使用ADC，必须等到校准完成。在开始A/D转换前应执行校准操作。通过软件设置CLB=1启动校准。在校准期间CLB位会一直保持1。一旦校准完成，该位由硬件清0。

当ADC运行条件改变(例如供电电压V<sub>DDA</sub>、温度等)时，建议重新执行一次校准操作。

内部的模拟校准可以通过设置ADC\_CTL1寄存器的RSTCLB位来重置。

软件校准过程：

1. 确保 ADCON=1;
2. 延迟 14 个 CK\_ADC 以等待 ADC 稳定;
3. 设置 RSTCLB (该步骤是可选的);
4. 设置 CLB=1;
5. 等待直到 CLB =0。

### 10.4.2. 双时钟域架构

除了APB接口时钟，ADC的子模块时钟还可以由ADC时钟提供。ADC时钟和APB时钟异步，并独立于APB时钟。

应用程序能够在低功耗运行时，降低PLCK时钟频率，同时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RCU章节的[4.2.1](#)部分。

### 10.4.3. ADC 使能

ADC\_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果该位为0，则ADC模块保持复位状态。为了省电，当ADCON位为0时，ADC模拟子模块将会进入掉电模式。ADC使能后需等待tsu时间后才能采样，tsu数值详见芯片数据手册。

### 10.4.4. 常规序列

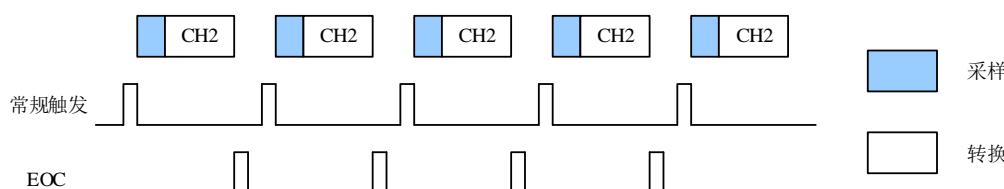
通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多16个通道，每个通道成为常规通道。ADC\_RSQ0寄存器的RL[3:0]位规定了整个常规序列转换序列的长度。ADC\_RSQ0~ADC\_RSQ2寄存器规定了常规序列的通道选择。

### 10.4.5. 运行模式

#### 单次运行模式

单次运行模式下，ADC\_RSQ2寄存器的RSQ0[4:0]位规定了ADC的转换通道。当ADCON位被置1时，一旦相应软件触发或者外部触发发生，ADC就会采样和转换一个通道。

**图 10-2. 单次运行模式**



常规通道单次转换结束后，转换数据将被存放于ADC\_RDATA寄存器中，EOC将会置1。如果EOCIE位被置1，将产生一个中断。

常规组单次运行模式的软件流程：

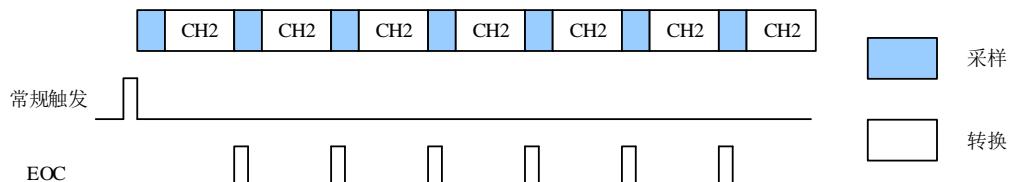
1. 确保ADC\_CTL0寄存器的DISRC位和SM位以及ADC\_CTL1寄存器中的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC\_SAMPTx寄存器；
4. 如果有需要，可以配置ADC\_CTL1寄存器中的ETERC位和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等到EOC位置1；
7. 从ADC\_RDATA寄存器中读ADC转换结果；

8. 写0清除EOC标志位。

### 连续运行模式

对ADC\_CTL1寄存器中的CTN位置1，可以使能连续运行模式。在此模式下，ADC执行由RSQ0[4:0]规定的转换通道。当ADCON位被置1，一旦相应软件触发或者外部触发产生，ADC就会采样和转换规定的通道。转换数据保存在ADC\_RDATA寄存器中。

**图 10-3. 连续运行模式**



常规序列连续运行模式的软件流程：

1. 设置ADC\_CTL1寄存器中的CTN位为1；
2. 根据模拟通道编号来配置RSQ0；
3. 配置ADC\_SAMPTx寄存器；
4. 如果有需要，配置ADC\_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者给常规序列产生一个外部触发信号；
6. 等待EOC标志位置1；
7. 从ADC\_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位；
9. 如果需要进行连续转换，重复步骤6~8。

可以使用DMA来传输转换数据，不需循环查询EOC标志位：

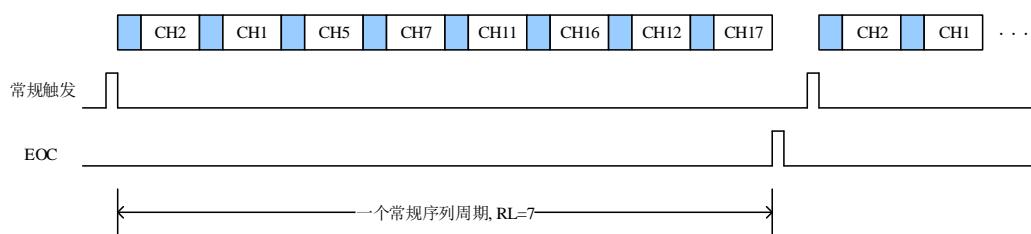
1. 设置ADC\_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC\_SAMPTx寄存器；
4. 如果有需要，配置ADC\_CTL1寄存器的ETERC位和ETSRC位；
5. 准备DMA模块，用于传输来自ADC\_RDATA寄存器的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

### 扫描运行模式

扫描运行模式可以通过将ADC\_CTL0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\_RSQ0~ADC\_RSQ2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，ADC就会一个接一个的采样和转换常规序列通道。转换数据存储在ADC\_RDATA寄存器中。常规序列转换结束后，EOC位将被置1。如果EOCIE位被置1，将产生中断。当常规序列工作在扫描模式下时，ADC\_CTL1寄存器的DMA位必须设置为1。

如果ADC\_CTL1寄存器的CTN位也被置1，则在常规序列转换完之后，转换自动重新开始。

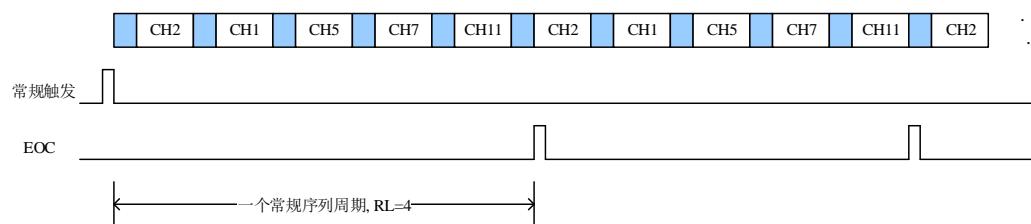
图 10-4. 扫描运行模式，且连续运行模式失能



常规序列扫描运行模式的软件流程：

1. 设置 ADC\_CTL0 寄存器的 SM 位和 ADC\_CTL1 寄存器的 DMA 位为 1;
2. 配置 ADC\_RSQx 和 ADC\_SAMPTx 寄存器;
3. 如果有需要，配置 ADC\_CTL1 寄存器中的 ETERC 和 ETSRC 位;
4. 准备 DMA 模块，用于传输来自 ADC\_RDATA 寄存器的数据;
5. 设置 SWRCST 位，或者给常规序列产生一个外部触发;
6. 等待 EOC 标志位置 1;
7. 写 0 清除 EOC 标志位。

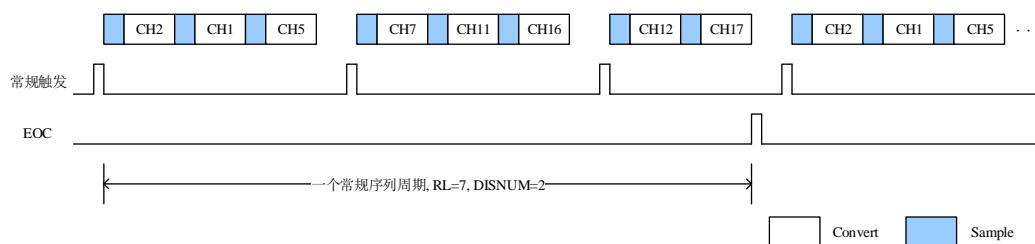
图 10-5. 扫描运行模式，连续运行模式使能



### 间断运行模式

当 ADC\_CTL0 寄存器的 DISRC 位被置 1 时，常规序列使能间断运行模式被使能。该模式下，可以执行一次 n 个通道的短序列转换(n 不超过 8)，这个短序列是 ADC\_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC\_CTL0 寄存器的 DISNUM[2:0]位配置。当相应的软件触发或外部触发发生，ADC 就会采样和转换在 ADC\_RSQ0~RSQ2 寄存器所选择通道中接下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EOC 位将被置 1。如果 EOCIE 位被置 1 将产生一个中断。

图 10-6. 间断运行模式



常规序列间断运行模式的软件流程：

1. 设置 ADC\_CTL0 寄存器的 DISRC 位和 ADC\_CTL1 寄存器的 DMA 位为 1;
2. 配置 ADC\_CTL0 寄存器的 DISNUM[2:0]位;
3. 配置 ADC\_RSQx 和 ADC\_SAMPTx 寄存器;

4. 如果有需要，配置 ADC\_CTL1 寄存器中的 ETERC 位和 ETSRC 位；
5. 准备 DMA 模块，用于传输来自 ADC\_RDATA 寄存器中的数据；
6. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
7. 如果需要，重复步骤 6；
8. 等待 EOC 标志位置 1；
9. 写 0 清除 EOC 标志位。

#### 10.4.6. 转换结果阈值监测功能

ADC\_CTL0 寄存器中的 RWDEN 位置 1 时，将使能常规序列的模拟看门狗功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 的模拟转换电压低于低阈值或高于高阈值时，ADC\_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC\_WDHT 和 ADC\_WDLT 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，因此阈值与 ADC\_CTL1 寄存器中的 DAL 位确定的对齐方式无关。ADC\_CTL0 寄存器的 RWDEN，WDSC 和 WDCHSEL[4:0]位可以用来选择模拟看门狗监控单一通道或多通道。

#### 10.4.7. 数据存储模式

ADC\_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

在最高有效位对齐中，12/10/8 位数据按半字方式对齐，而 6 位数据按照字节的方式对齐的，如下 [图 10-7. 12 位数据存储模式](#)，[图 10-8. 10 位数据存储模式](#)，[图 10-9. 8 位数据存储模式](#) 和 [图 10-10. 6 位数据存储模式](#) 所示。

**图 10-7. 12 位数据存储模式**

常规通道数据															
0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
DAL=0															

常规通道数据															
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
DAL=1															

**图 10-8. 10 位数据存储模式**

常规通道数据															
0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
DAL=0															

常规通道数据															
D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
DAL=1															

图 10-9. 8 位数据存储模式

常规通道数据															
0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0	
DAL=0															

常规通道数据															
D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0
DAL=1															

图 10-10. 6 位数据存储模式

常规通道数据															
0	0	0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0
DAL=0															

常规通道数据															
0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0
DAL=1															

#### 10.4.8. 采样时间配置

ADC 使用多个 CK\_ADC 周期对输入电压采样，采样周期数目可以通过 ADC\_SAMPT0 和 ADC\_SAMPT1 寄存器的 SPTn[2:0]位设置。每个通道可以使用不同的时间采样。例如，在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 CK\_ADC 周期。

例如：

$CK_{ADC} = 28MHz$ , 采样时间为 1.5 个周期，那么总的转换时间为：“1.5+12.5”个 CK\_ADC 周期，即 0.500us。

#### 10.4.9. 外部触发配置

外部触发输入的上升沿可以触发常规序列的转换。常规序列的外部触发源由 ADC\_CTL1 寄存器的 ETSRC[2:0]位控制。

表 10-3. ADC 的外部触发源

ETSRC[2:0]	触发源	触发类型
000	TIMER0_CH0	硬件触发
001	TIMER0_CH1	
010	TIMER0_CH2	
011	保留	
100	TIMER2_TRGO	
101	TIMER14_CH0	
110	EXTI11	

ETSRC[2:0]	触发源	触发类型
111	SWRCST	软件触发

#### 10.4.10. DMA 请求

DMA请求，可以通过设置ADC\_CTL1寄存器的DMA位来使能，用来传输常规序列多个通道的转换结果。ADC在常规序列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\_RDATA寄存器传输到用户指定的目的地址。

#### 10.4.11. ADC 内部通道

将ADC\_CTL1寄存器的TSVREN位置1，可以使能温度传感器通道(ADC\_IN16)和V<sub>REFINT</sub>通道(ADC\_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。建议温度传感器的采样时间至少设置为17.1μs。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会发生线性变化，由于芯片生产过程的多样化，温度变化曲线的偏差在芯片间会有不同(最多相差45°C)。内部温度传感器更适用于检测温度的变化，而不是用于测量绝对温度。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来校准这个偏移错误。

内部电压参考(V<sub>REFINT</sub>)提供了一个稳定的(带隙基准)电压输出给ADC和比较器。V<sub>REFINT</sub>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(ADC\_IN16)的转换序列和采样时间为17.1us
2. 置位ADC\_CTL1寄存器中的TSVREN位，使能温度传感器
3. 置位ADC\_CTL1寄存器的ADCON位，或者由外部触发ADC转换
4. 读取内部温度传感器输出电压V<sub>temperature</sub>，并由下面公式计算出实际温度：

$$\text{温度 } (\text{°C}) = \{(V_{25} - V_{\text{temperature}}) / \text{Avg\_Slope}\} + 25$$

V<sub>25</sub>: 温度传感器在25°C下的电压，典型值请参考datasheet。

Avg\_Slope: 温度与温度传感器电压曲线的均值斜率，典型值请参考datasheet。

#### 10.4.12. 可编程分辨率(DRES)

ADC分辨率可以通过寄存器ADC\_CTL0中的DRES[1:0]位进行配置。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位为0时，才能修改DRES[1:0]的值。较低的分辨率能够减少转换时间，如表10-4. 不同分辨率对应的t<sub>CONV</sub>时间所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间t<sub>ADC</sub>。

表10-4. 不同分辨率对应的t<sub>CONV</sub>时间

DRES [1:0] bits	t <sub>CONV</sub> (ADC 时钟周期)	t <sub>CONV</sub> (ns) (f <sub>ADC</sub> =28MHz)	t <sub>SMPL</sub> (ADC 时 钟周期)	t <sub>ADC</sub> (ADC 时 钟周期)	t <sub>ADC(ns)</sub> (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

#### 10.4.13. 片上硬件过采样

片上硬件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 位宽的数据。

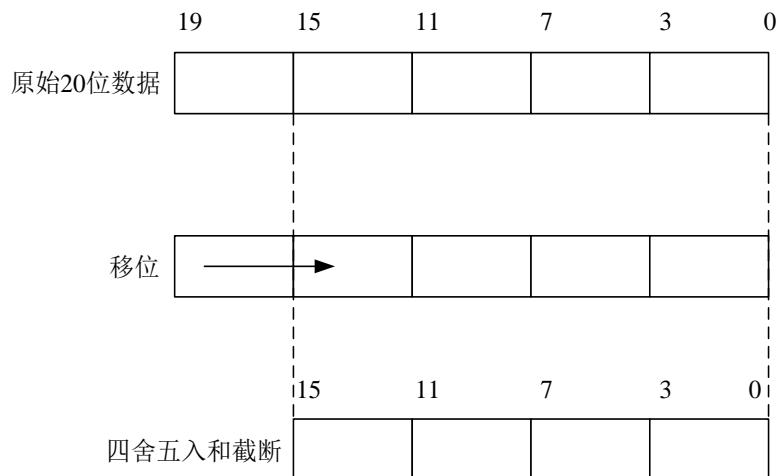
其结果根据如下公式计算得出，其中，N 和 M 的值可以被调整，过采样单元可以通过设置 ADC\_OVSAMPCTL 寄存器中的 OVSEN 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。D<sub>out</sub>(n)是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{n=N-1} D_{out}(n) \quad (11-1)$$

片上硬件过采样单元执行两个功能：求和和位右移。过采样率 N 是在 ADC\_OVSAMPCTL 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 定义了一个多达 8 位的右移，它通过 ADC\_OVSAMPCTL 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。首先，将这个值进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位截断，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。

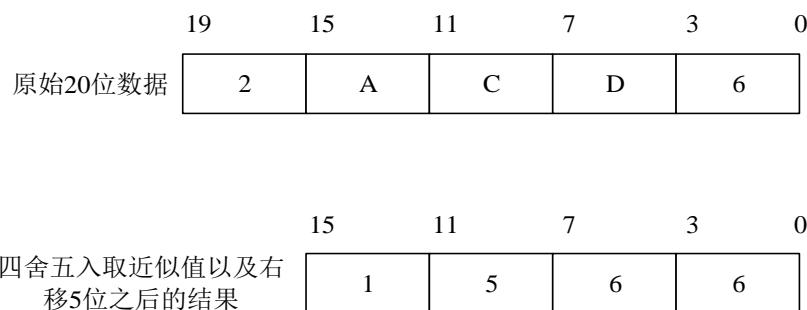
图 10-11. 20 位到 16 位的结果截断



**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 10-12. 右移 5 位和取整的数例](#)描述了一个从原始 20 位的累积数值处理成 16 位结果值的例子。

图 10-12. 右移 5 位和取整的数例



**表 10-5. 不同 N 和 M 组合的最大输出值 (灰色值表示截断)** 给出了 N 和 M 的各种组合的数据格式，初始转换值为 0xFFFF。

表 10-5. 不同 N 和 M 组合的最大输出值 (灰色值表示截断)

过采样率	最大原始数据	无移位 OVSS=0000	1 位移位 OVSS=0001	2 位移位 OVSS=0010	3 位移位 OVSS=0011	4 位移位 OVSS=0100	5 位移位 OVSS=0101	6 位移位 OVSS=0110	7 位移位 OVSS=0111	8 位移位 OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFFF0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFFF0	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

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样时间仍然保持相等。每 N 个转换就会产生一个新的数据，一个等价的延迟为：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (11-2)$$

### 过采样配合 ADC 工作模式

当过采样使能时，大多数 ADC 工作模式都是可用的。

- 常规序列通道
- 由软件触发或外部触发开始 ADC 转换
- 单次或扫描运行模式，连续或间断运行模式
- 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。

#### 10.4.14. ADC 中断

以下任一个事件发生都可以产生中断：

- 常规序列转换结束；
- 模拟看门狗事件；

单独的中断使能位用于灵活设置ADC中断。

## 10.5. ADC 寄存器

ADC基地址: 0x4001 2400

### 10.5.1. 状态寄存器 (ADC\_STAT)

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		保留		EOC		rc_w0	

位/位域	名称	描述
31:5	保留	必须保持复位值
4	STRC	常规序列转换开始标志 0: 转换没有开始 1: 转换开始 常规序列转换开始时硬件置位。软件写0清除。
3:2	保留	必须保持复位值。
1	EOC	常规序列转换结束标志 0: 转换没有结束 1: 转换结束 常规序列转换结束时硬件置位。 软件写0或读ADC_RDATA寄存器清除。
0	WDE	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_WDLT和ADC_WDHT寄存器中设定的阈值时由硬件置1，软件写0清除。

### 10.5.2. 控制寄存器 0 (ADC\_CTL0)

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								DRES [1:0]		RWDEN		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]		保留	DISRC	保留	WDSC	SM	保留	WDEIE	EOCIE	WDCHSEL [4:0]									
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	RWDEN	常规序列模拟看门狗使能 0: 常规序列模拟看门狗禁止 1: 常规序列模拟看门狗使能
22:16	保留	必须保持复位值
15:13	DISNUM [2:0]	间断模式下的转换转换数目 触发后即将被转换的通道数目将变成DISNUM[2:0]+1。
12	保留	必须保持复位值。
11	DISRC	常规序列间断模式 0: 间断运行模式禁止 1: 间断运行模式使能
10	保留	必须保持复位值。
9	WDSC	扫描模式下, 模拟看门狗在通道配置 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: 扫描运行模式禁止 1: 扫描运行模式使能
7	保留	必须保持复位值。
6	WDEIE	WDE中断使能 0: 中断禁止 1: 中断使能
5	EOCIE	EOC中断使能 0: 中断禁止 1: 中断使能

---

4:0	WDCHSEL [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 ..... 01000: ADC通道8 01001: ADC通道9 10000: ADC通道16 10001: ADC通道17
<b>注意:</b> ADC的模拟输入通道16和通道17分别连接到温度传感器和V <sub>REFINT</sub> 。		

### 10.5.3. 控制寄存器 1 (ADC\_CTL1)

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				TSVREN	SWRCST	保留	ETERC	ETSRC [2:0]				保留	保留		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DAL	保留	DMA	保留				RSTCLB	CLB	CTN	ADCON	保留			
rw								rw				rw			

位/位域	名称	描述
31:25	保留	必须保持复位值
23	TSVREN	ADC的通道16和17使能 0: ADC的通道16和17禁止 1: ADC的通道16和17使能
22	SWRCST	软件触发常规序列转换开始 如果ETSRC是111, 该位置'1'开启常规序列转换。该位由软件置位, 软件清零或转换开始由硬件清零。
21	保留	必须保持复位值。
20	ETERC	常规序列外部触发使能 0: 常规序列外部触发禁止 1: 常规序列外部触发使能
19:17	ETSRC [2:0]	常规序列通道外部触发选择 000: TIMER0 CH0 001: TIMER0 CH1 010: TIMER0 CH2 011: 保留

		100: TIMER2 TRGO
		101: TIMER14 CH0
		110: 中断线 11
		111: 软件触发SWRCST
16:12	保留	必须保持复位值
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 软件置位，在校准寄存器初始化后，该位硬件清零。 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: 连续运行模式禁止 1: 连续运行模式使能
0	ADCON	开启ADC。该位从'0'变成'1'将在稳定时间结束后唤醒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: 禁能ADC并掉电 1: 使能ADC

#### 10.5.4. 采样时间寄存器 0 (ADC\_SAMPT0)

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]				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
23:21	SPT17[2:0]	参考SPT16[2:0]的描述
20:18	SPT16[2:0]	通道采样时间 000: 通道采样时间为1.5周期 001: 通道采样时间为7.5周期 010: 通道采样时间为13.5周期 011: 通道采样时间为28.5周期 100: 通道采样时间为41.5周期 101: 通道采样时间为55.5周期 110: 通道采样时间为71.5周期 111: 通道采样时间为239.5周期
17:0	保留	必须保持复位值

### 10.5.5. 采样时间寄存器 1 (ADC\_SAMPT1)

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]					
		rw		rw		rw									
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		rw									

位/位域	名称	描述
31:30	保留	必须保持复位值
29:27	SPT9[2:0]	参考SPT0[2:0]的描述
26:24	SPT8[2:0]	参考SPT0[2:0]的描述
23:21	SPT7[2:0]	参考SPT0[2:0]的描述
20:18	SPT6[2:0]	参考SPT0[2:0]的描述
17:15	SPT5[2:0]	参考SPT0[2:0]的描述
14:12	SPT4[2:0]	参考SPT0[2:0]的描述
11:9	SPT3[2:0]	参考SPT0[2:0]的描述
8:6	SPT2[2:0]	参考SPT0[2:0]的描述

5:3	SPT1[2:0]	参考SPT0[2:0]的描述
2:0	SPT0[2:0]	通道采样时间 000: 通道采样时间为1.5周期 001: 通道采样时间为7.5周期 010: 通道采样时间为13.5周期 011: 通道采样时间为28.5周期 100: 通道采样时间为41.5周期 101: 通道采样时间为55.5周期 110: 通道采样时间为71.5周期 111: 通道采样时间为239.5周期

### 10.5.6. 看门狗高阈值寄存器 (ADC\_WDHT)

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															WDHT [11:0]
rw															

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	WDHT [11:0]	模拟看门狗高侧阈值 这些位定义了模拟看门狗的高侧阈值。

### 10.5.7. 看门狗低阈值寄存器 (ADC\_WDLT)

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
保留															WDLT [11:0]
rw															

位/位域	名称	描述
------	----	----

31:12	保留	必须保持复位值
11:0	WDLT [11:0]	模拟看门狗低侧阈值 这些位定义了模拟看门狗的低侧阈值。

### 10.5.8. 常规序列寄存器 0 (ADC\_RSQ0)

地址偏移: 0x2C

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						RL [3:0]				RSQ15[4:1]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]		RSQ14[4:0]						RSQ13[4:0]				RSQ12[4:0]			
rw			rw					rw				rw			

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	RL [3:0]	常规序列长度 常规通道转换序列中的总通道数目为 RL[3:0]+1。
19:15	RSQ15[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考RSQ0[4:0]的描述

### 10.5.9. 常规序列寄存器 1 (ADC\_RSQ1)

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		RSQ11[4:0]						RSQ10[4:0]				RSQ9[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ9[0]		RSQ8[4:0]						RSQ7[4:0]				RSQ6[4:0]			
rw			rw					rw				rw			

位/位域	名称	描述

31:30	保留	必须保持复位值
29:25	RSQ11[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ10[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ9[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ8[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考RSQ0[4:0]的描述

### 10.5.10. 常规序列寄存器 2 (ADC\_RSQ2)

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		RSQ5[4:0]				RSQ4[4:0]				RSQ3[4:1]					
rw															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ3[0]		RSQ2[4:0]				RSQ1[4:0]				RSQ0[4:0]					
rw															rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	RSQ5[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ0[4:0]	通道编号(0..9,16,17)写入这些位来选择常规通道的第n个转换的通道。

### 10.5.11. 常规数据寄存器 (ADC\_RDATA)

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
保留															

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RDATA [15:0]	常规通道转换数据 这些位包含了常规通道的转换结果，只读。

### 10.5.12. 过采样控制寄存器(ADC\_OVSAMPCTL)

地址偏移: 0x80

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		TOVS		OVSS [3:0]		OVSR [2:0]		保留		OVSEN					
		rw		rw		rw		rw		rw					

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 该位通过软件设置和清除。 0: 在一个触发后，连续执行过采样通道的所有转换 1: 对于过采样通道的每次转换都需要一次触发，触发次数由过采样率 (OVSR[2:0]) 决定 <b>注意:</b> 当ADCON= 0时软件才允许写该位(确定没有转换正在进行)。
8:5	OVSS [3:0]	过采样移位 该位通过软件设置和清除。 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

## 其它位保留

**注意:** 只有在ADCON=0的时候, 才允许通过软件对该位进行写操作(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 <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操作(确保没有转换正在执行)。
1	保留	必须保持复位值
0	OVSEN	过采样使能 该位通过软件置位和清除。 0: 过采样禁止 1: 过采样使能 <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操作(确保没有转换正在执行)。

## 11. 比较器 (CMP)

### 11.1. 简介

通用比较器可独立工作，其输出端可用于 I/O 口，也可和定时器结合使用。

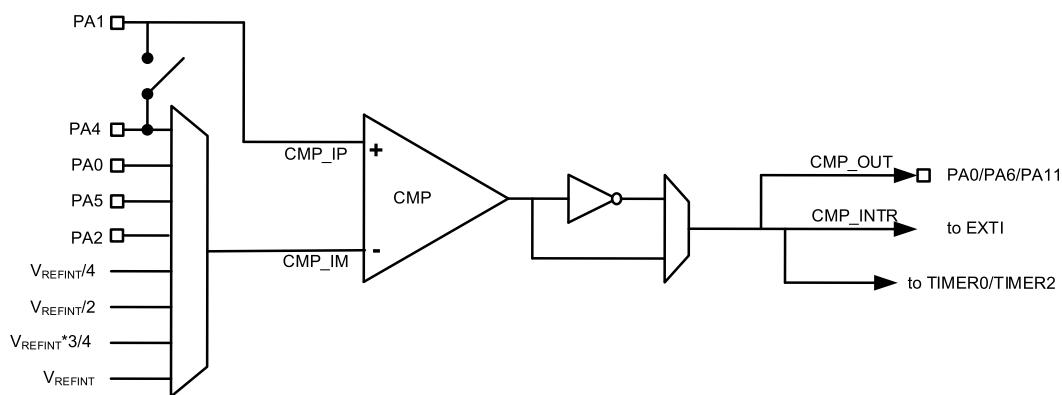
### 11.2. 主要特性

- 轨对轨比较器
- 迟滞可配置
- 速度、功耗可配置
- 比较器可配置以下模拟信号作为输入源
  - 比较器输入引脚
  - 0.25、0.5、0.75、1 倍的内部参考电压
- 输出到 I/O 口
- 作为触发源输出到定时器
- 输出到 EXTI

### 11.3. 功能描述

比较器的框图展示如下：

图 11-1. 比较器框图



注意 : VREFINT 是 1.2V.

### 11.3.1. 比较器时钟和复位

比较器的时钟与 PCLK 同步，它与 SYSCFG 分享共同的复位和时钟使能位。

### 11.3.2. 比较器的 I/O 配置

在被选为比较器输入端之前，相应管脚必须配置为模拟模式。

参考 Datasheet 的引脚定义，比较器输出必须连接到对应的复用 I/O 口。

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器输入通道。
- CMP 输出连接到定时器中止功能。
- CMP 输出连接到定时器 OCPRE\_CLR。

为了在深度睡眠模式下工作，比较器端口的极性选择和输出可以独立与 PCLK 工作。

比较器的输出可同时实现内部和外部输出。

每个比较器都有其自己的 EXTI 线，都可用于产生中断或事件，并且可以从省电模式中退出。

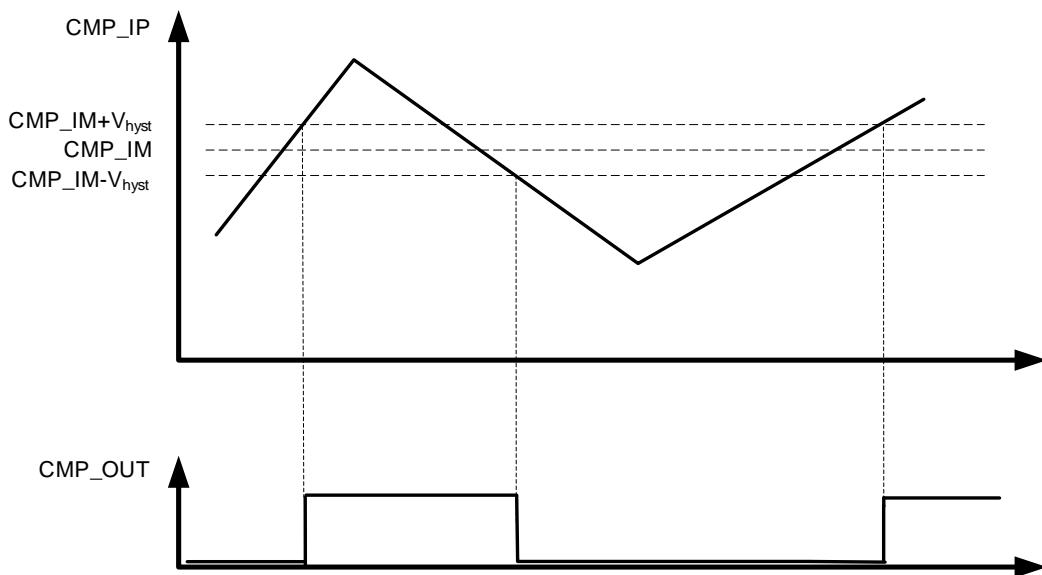
### 11.3.3. 比较器供电模式

对于给定的程序，在比较器功耗和传输迟滞之间存在着协定，可通过寄存器 CMP\_CS 的位 CMPM 的配置进行调整。当 CMPM 为 2'b 00 时，比较器以运行速度最快和功耗最大模式工作，但当 CMPM 位 2'b 11 时，比较器以运行速度最慢和功耗最小的模式工作。

### 11.3.4. 比较器迟滞

为了避免噪声信号所引起的假输出，比较器的迟滞可配置，该功能可以在无需要时关闭

图 11-2. 比较器迟滞



### 11.3.5. 比较器寄存器写保护

比较器的控制状态寄存器（ $CMP\_CS$ ）可通过设置  $CMPLK$  位为 1 来进行保护， $CMP\_CS$  寄存器，包含  $CMPLK$  位，就会变为只读位，只有在 MCU 复位时才可以复位。

## 11.4. 比较器寄存器

CMP 基址: 0x4001 001C

### 11.4.1. 控制状态寄存器(CMP\_CS)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器通过字访问(32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPLK	CMPO	CMPHST[1:0]	CMPPL	CMPOSEL[2:0]	保留	CMPMSEL[2:0]	CMPM[1:0]	CMPSW	CMPEN						
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	

位/位域	名称	描述
31:16	保留	必须为保留值。
15	CMPLK	<b>CMP lock</b> 该位可将 CMP 的各控制位设为只读, 该位可写一次, 通过系统复位清除, 可通过软件置位 0: CMP_CS[15:0] 是可读可写位 1: CMP_CS[15:0] 是只读位
14	CMPO	<b>CMP 输出</b> 该位即 CMP 输出状态, 是只读位 0: 正相输入端低于反相输入端, 输出为低电平 1: 正相输入端高于反相输入端, 输出为高电平
13:12	CMPHST[1:0]	<b>CMP 迟滞</b> 该域用于控制迟滞水平 00: 无迟滞 01: 低迟滞 10: 中迟滞 11: 高迟滞
11	CMPPL	<b>CMP 输出极性</b> 该位是用来控制 CMP 输出极性 0: 输出是正相的 1: 输出是反相的
10:8	CMPOSEL[2:0]	<b>比较器的输出选择</b> 该域用于控制 CMP 输出选择 000: 无选择 001: 定时器 0 中止输入

		010: 定时器 0 通道 0 输入捕获 011: 定时器 0 OCPRE_CLR 输入 100: 保留 101: 保留 110: 定时器 2 通道 0 输入捕获 111: 定时器 2 OCPRE_CLR 输入
7	保留	必须为保留值
6:4	CMPMSEL[2:0]	CMP_IM 输入选择 该位用来选择 CMP 的输入端 CMP_IM 的输入源 000: VREFINT/4 001: VREFINT/2 010: VREFINT*3/4 011: VREFINT 100: PA4 101: PA5 110: PA0 111: PA2
3:2	CMPM[1:0]	CMP 模式 该位用来控制 CMP 的运行模式 00: 高速/全功耗 01: 中速/中功耗 10: 低速/低功耗 11: 超低速/超低功耗
1	CMPSW	CMP 开关 该位用于开关 CMP 正相输入端 PA1 与 PA4 之间的连接 0: 开关断开 1: 开关闭合
0	CMPEN	CMP 使能 0: CMP 除能 1: CMP 使能

## 12. 看门狗定时器 (WDGT)

看门狗定时器 (WDGT) 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。片上有两个看门狗定时器外设，独立看门狗定时器 (FWDGT) 和窗口看门狗定时器 (WWDGT)。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。两个看门狗定时器都是用来解决软件故障问题的。

看门狗定时器在内部计数值达到预设门限的时候，会触发一个复位。当处理器工作在调试模式的时候看门狗定时器定时计数器可以停止计数。

### 12.1. 独立看门狗定时器 (FWDGT)

#### 12.1.1. 简介

独立看门狗定时器 (FWDGT) 有独立的时钟源 (IRC40K)。即使主时钟失效，FWDGT依然能保持正常工作状态，适用于需要独立环境且对计时精度要求不高的场合。

当内部向下计数器的计数值达到0，独立看门狗会产生一个系统复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

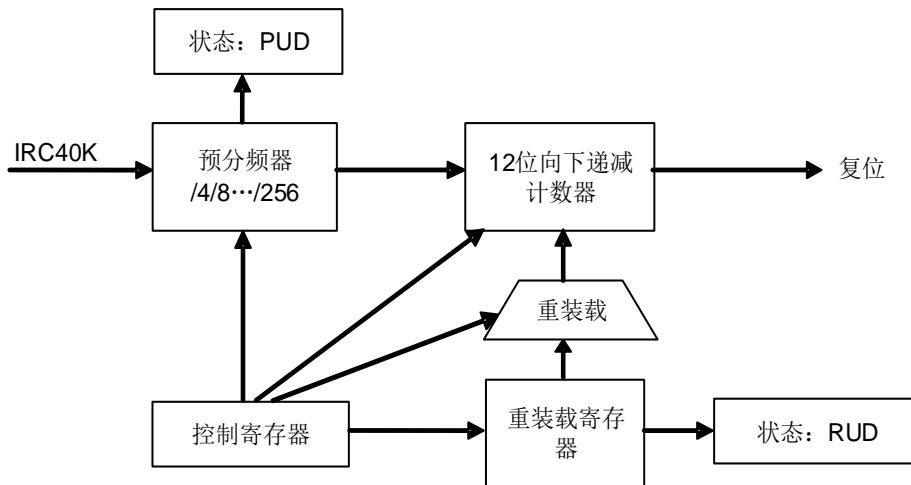
#### 12.1.2. 主要特性

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，有以下两种情况下会产生复位：
  - 当计数器到0时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 独立时钟源，独立看门狗定时器在主时钟故障(例如待机和深度睡眠模式下)时仍能工作；
- 独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

#### 12.1.3. 功能描述

独立看门狗定时器带有一个8级预分频器和一个12位的向下递减计数器。参考[图12-1. 独立看门狗定时器框图](#)为独立看门狗定时器的功能模块。

图 12-1. 独立看门狗定时器框图



向控制寄存器（FWDGT\_CTL）中写0xCCCC可以开启独立看门狗定时器，计数器开始向下计数。当计数器记到0x000，产生一次系统复位。

在任何时候向FWDGT\_CTL中写0xAAAA都可以重装载计数器，重装载值来源于重装载寄存器（FWDGT\_RLD）。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定时器产生系统复位。

独立看门狗定时器也能够作为窗口看门狗定时器运行，只要在窗口寄存器（FWDGT\_WND）寄存器中设置适当的窗口值即可。当如果重装载操作执行的同时，如果看门狗定时器计数器的值大于超出了窗口寄存器（FWDGT\_WND）中存储的值，也将会引起系统复位。FWDGT\_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定时器计数器的一次重加载动作，将向下递减计数器置为 FWDGT\_RLD 中的值，并复位预分频计数器。

如果在选项字节中打开了“硬件看门狗定时器”功能，那么在上电的时候看门狗定时器就被自动打开。为了避免系统复位，软件应该在计数器达到0x000之前重装载计数器。

预分频寄存器（FWDGT\_PSC）寄存器和FWDGT\_RLD寄存器都有写保护功能。在写数据到这些寄存器之前，需要写0x5555到控制寄存器（FWDGT\_CTL）中。写其他任何值到控制寄存器 FWDGT\_CTL 中将会再次启动对这些寄存器的写保护。当预分频寄存器（FWDGT\_PSC）或者重装载寄存器（FWDGT\_RLD）更新时，FWDGT\_STAT寄存器相应的状态位被置1。

如果DBG中控制寄存器0（DBG\_CTL0）中的FWDGT\_HOLD位被清0，即使Cortex™-M3内核停止（调试模式下）独立看门狗定时器依然工作。如果FWDGT\_HOLD位置1，独立看门狗定时器将在调试模式下停止工作。

表 12-1. 独立看门狗定时器在 40kHz (IRC40K) 时的最小/最大超时周期

预分频系数	PSC[2:0]位	最小超时 (ms)	
		RLD[11:0]=0x000	RLD[11:0]=0xFFFF
1 / 4	000	0.025	409.525
1 / 8	001	0.025	819.025
1 / 16	010	0.025	1638.025

预分频系数	PSC[2:0]位	最小超时 (ms) RLD[11:0]=0x000	最大超时 (ms) RLD[11:0]=0xFFFF
1 / 32	011	0.025	3276.025
1 / 64	100	0.025	6552.025
1 / 128	101	0.025	13104.025
1 / 256	110 or 111	0.025	26208.025

通过校准IRC40K可以使独立看门狗定时器超时更加精确。

**注意：**当执行完喂狗reload操作之后，如需要立即进入deepsleep / standby模式时，必须通过软件设置，在reload命令及deepsleep / standby模式命令中间插入（3个以上）IRC40K时钟间隔。

### 12.1.4. FWDGT 寄存器

FWDGT 基地址: 0x4000 3000

#### 控制寄存器 (FWDGT\_CTL)

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
CMD[15:0]															

w

位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写, 写入不同的值来产生不同的功能 0x5555: 关闭FWDGT_PSC、FWDGT_RLD和FWDGT_WND的写保护 0xCCCC: 开启独立看门狗定时器定时计数器。计数减到0时产生复位 0xAAAA: 重装载计数器

#### 预分频寄存器 (FWDGT\_PSC)

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														PSC[2:0]	
														rw	

位/位域	名称	说明
31:3	保留	必须保持复位值
2:0	PSC[2:0]	独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中, FWDGT_STAT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

010: 1 / 16

011: 1 / 32

100: 1 / 64

101: 1 / 128

110: 1 / 256

111: 1 / 256

如果应用需要使用几个预分频系数，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（在进入省电模式前需等待PUD值清零）。

### **重装载寄存器 (FWDGT\_RLD)**

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>独立看门狗定时器定时计数器重装载值，向FWDGT_CTL寄存器写入0xFFFF的时候，这个值会被更新到独立看门狗定时器计数器中。</p> <p>这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT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>如果应用需要使用几个重装载值，改变重装载值之前必须等到RUD位被清0。更新了重装载值后，在代码持续执行之前不必等待RUD值被清零（在进入省电模式前需等待RUD值清零）。</p>

### **状态寄存器 (FWDGT\_STAT)**

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

r r r

位/位域	名称	说明
31:2	保留	必须保持复位值。
2	WUD	独立看门狗定时计数器窗口值更新。 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器计数器重装载值更新 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。
0	PUD	独立看门狗定时器预分频值更新 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。

### 窗口寄存器（FWDGT\_WND）

地址偏移: 0x10

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				WND[11:0]											
rw															

位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WND[11:0]	独立看门狗定时计数器窗口值。这些位将用来将窗口值的上限值与向下递减计数器进行比较。当计数值大于WND[11:0]中值，重装载操作会引起复位，若要改变重装载值，FWDGT_STAT寄存器中的WUD位必须保持复位状态。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。 如果应用需要使用几个窗口值，改变窗口值之前必须等到WUD位被清0。除了在进入低功耗模式下，更新了窗口值后，在代码持续执行之前不必等待WUD值被清零。

## 12.2. 窗口看门狗定时器 (WWDGT)

### 12.2.1. 简介

窗口看门狗定时器 (WWDGT) 用来监测由软件故障导致的系统故障。窗口看门狗定时器开启后，7位向下递减计数器值逐渐减小。计数值达到0x3F时会产生系统复位 (CNT[6]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系统复位。因此软件需要在给定的区间内更新计数器。窗口看门狗定时器在计数器计数值达到0x40，会产生一个提前唤醒标志，如果使能中断将会产生提前唤醒中断。

窗口看门狗定时器时钟是由APB1时钟预分频而来。窗口看门狗定时器适用于需要精确计时的场合。

### 12.2.2. 主要特性

- 可编程的7位自由运行向下递减计数器；
- 当窗口看门狗使能后，有以下两种情况会产生复位：
  - 当计数器达到0x3F时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断 (EWI)：看门狗定时器打开，中断使能，计数值达到0x40时会产生中断；
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

### 12.2.3. 功能描述

如果窗口看门狗定时器使能（将WWDGT\_CTL寄存器的WDGTEN位置1），计数值达到0x3F的时候产生系统复位 (CNT[6]位被清0) 或者在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

图 12-2. 窗口看门狗定时器框图



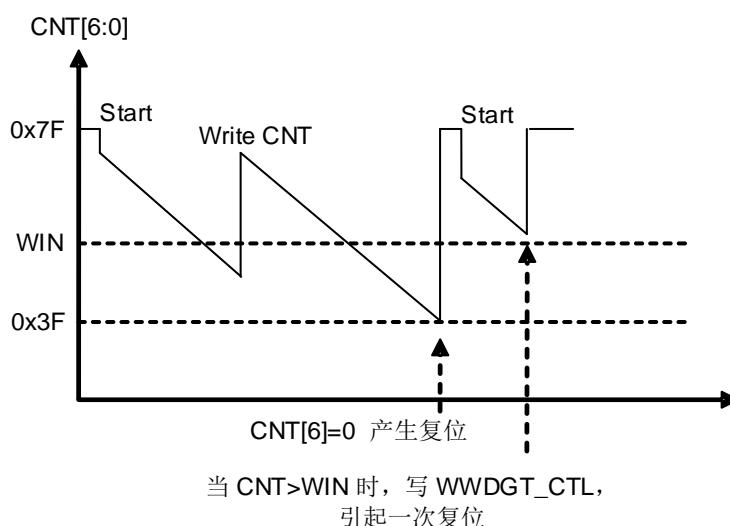
上电复位之后窗口看门狗定时器总是关闭的。软件可以向WWDGT\_CTL的WDGTEN写1开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于APB1时钟和预分频器 (WWDGT\_CFG寄存器的PSC[1:0]位)。

配置寄存器 (WWDGT\_CFG) 中的 WIN[6:0] 位用来设定窗口值。当计数器的值小于窗口值，且大于 0x3F 的时候，重装载向下计数器可以避免复位，否则在其他时候进行重加载就会引起复位。

对 WWDGT\_CFG 寄存器的 EWIE 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，在 ISR 中软件可以重装载计数器来管理软件系统检查等。在这种情况下，窗口看门狗定时器将永远不会复位但是可以用于其他地方。

通过将 WWDGT\_STAT 寄存器的 EWIF 位写 0 可以清除 EWI 中断。

图 12-3. 窗口看门狗定时器时序图



窗口看门狗定时器超时的计算公式如下：

$$t_{WWDGT} = t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0] + 1) \quad (ms) \quad (12-1)$$

其中：

$t_{WWDGT}$ : 窗口看门狗定时器的超时时间

$t_{PCLK1}$ : APB1 以 ms 为单位的时钟周期

$t_{WWDGT}$  的最大值和最小值请参考 [表 12-2. 在 72MHz \(f<sub>PCLK1</sub>\) 时的最大/最小超时值](#)。

表 12-2. 在 72MHz (f<sub>PCLK1</sub>) 时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时 $CNT[6:0] = 0x00$	最大超时 $CNT[6:0] = 0x7F$
1/1	00	56 μs	3.64 ms
1/2	01	113 μs	7.28 ms
1/4	10	227 μs	14.56 ms
1/8	11	455 μs	29.12 ms

如果 MCU 调试模块中的 WWDGT\_HOLD 位被清 0，即使 Cortex®-M23 内核停止工作（调试模式下），窗口看门狗定时器也可以继续工作。当 WWDGT\_HOLD 位被置 1 时，窗口看门狗定时器在调试模式下停止。

### 12.2.4. WWDGT 寄存器

WWDGT 基地址: 0x4000 2C00

#### 控制寄存器 (WWDGT\_CTL)

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。



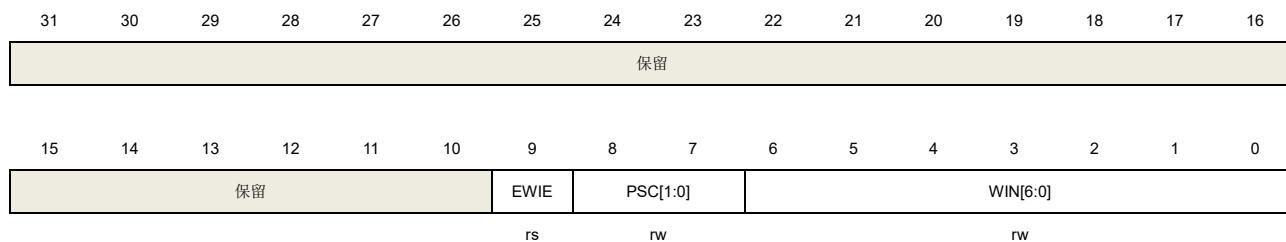
位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器, 硬件复位的时候清0, 写0无效。 0: 关闭窗口看门狗定时器。 1: 开启窗口看门狗定时器。
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时, 产生看门狗定时器复位。 当计数器值高于窗口值的时候, 写计数器可以产生看门狗定时器系统复位。

#### 配置寄存器 (WWDGT\_CFG)

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。



位/位域	名称	说明
31:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。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, 或通过置位RCU模块的WWDGTRST位进行软件复位。写0没有任何作用。

8:7	PSC[1:0]	预分频器，看门狗定时器计数器的时间基准。 00: PCLK1 / 4096 / 1 01: PCLK1 / 4096 / 2 10: PCLK1 / 4096 / 4 11: PCLK1 / 4096 / 8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器(WWDGT_CTL的CNT位)会产生系统复位。

### 状态寄存器 (WWDGT\_STAT)

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。

31	30	29	28	27	26	25	24	23	22	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															EWIF

rw

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(WWDGT_CFG中的EWIE位为0)该位也会被硬件置1。这个位可以通过写0清零，写1无效。

## 13. 实时时钟 (RTC)

### 13.1. 简介

RTC模块提供了一个包含日期（年/月/日）和时间（时/分/秒/亚秒）的日历功能。除亚秒用二进制码显示外，时间和日期都以BCD码的形式显示。RTC可以进行夏令时补偿。RTC可以工作在省电模式下，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。

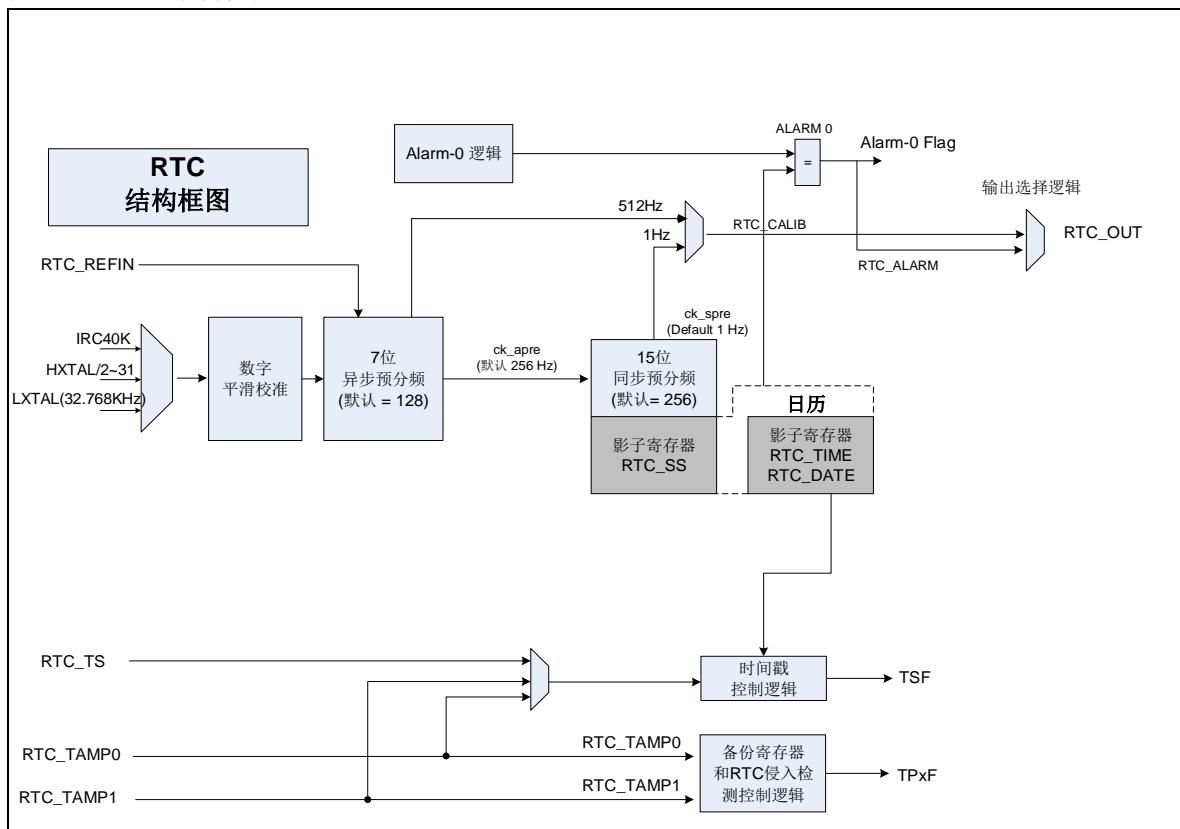
### 13.2. 主要特性

- 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)来提高日历精度。
- 数字校准功能：通过调整最小时间单位（最大可调精度0.95ppm）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。
- 两个模式可配置的独立的侵入检测（GD32E231xx产品为1个）。
- 可编程的日历和一个位域可屏蔽的闹钟。
- 可屏蔽的中断源：
  - 闹钟 0；
  - 时间戳检测（GD32E231xx 产品无此功能）；
  - 侵入检测；
- 5个32位 (共20字节) 通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位。

### 13.3. 功能描述

#### 13.3.1. 结构框图

图 13-1. RTC 结构框图



RTC 单元包括：

- 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（GD32E231xx产品无此功能）：
  - 512Hz (默认预分频值): RTC\_OUT;
  - 1Hz(默认预分频值) : RTC\_OUT;
  - 闹钟事件(极性可配置) : RTC\_OUT。
- 可选的RTC输入功能:
  - 时间戳事件检测(RTC\_TS) (GD32E231xx 产品无此功能);
  - 侵入事件检测 0(RTC\_TAMP0) (GD32E231xx 产品无此功能);
  - 侵入事件检测 1(RTC\_TAMP1);
  - 参考时钟输入 RTC\_REFIN(50 或 60Hz)。

#### 13.3.2. 时钟源和预分频

RTC 单元有三个可选的独立时钟源：LXTAL、IRC40K和HXTAL的32分频后的时钟。

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：

$$f_{ck\_apre} = \frac{f_{rtcclk}}{\text{FACTOR\_A} + 1} \quad (\text{式13-1})$$

$$f_{ck\_spre} = \frac{f_{ck\_apre}}{\text{FACTOR\_S} + 1} = \frac{f_{rtcclk}}{(\text{FACTOR\_A} + 1) * (\text{FACTOR\_S} + 1)} \quad (\text{式13-2})$$

`ck_apre` 用于为 RTC\_SS 亚秒寄存器自减计数器提供时钟，该寄存器值为二进制，表示到达下一秒时间，该寄存器自减到 0 时，自动加载 `FACTOR_S` 的值。`ck_spre` 用于为日历寄存器提供时钟，每个时钟增加一秒。

### 13.3.3. 影子寄存器

当APB总线访问RTC日历寄存器RTC\_DATE、RTC\_TIME和RTC\_SS时，`BPSHAD`位决定是访问影子寄存器还是真实日历寄存器。默认情况下`BPSHAD`为0，APB总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`RSYNF`位也会再次置位。在Deep-sleep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`RSYNF`位。如果想要在`BPSHAD=0`的情况下读日历寄存器的值，须等待`RSYNF`置1（最大的等待时间是2个RTC时钟周期）。

**注意：**在 `BPSHAD=0` 下，读日历寄存器(RTC\_SS, RTC\_TIME, RTC\_DATE)的APB时钟的频率( $f_{appb}$ )必须至少是RTC时钟频率( $f_{rtcclk}$ )的七倍。

系统复位将复位影子寄存器。

### 13.3.4. 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\_CTL寄存器中的`ALRMxEN` ( $x=0$ ) 位控制。当`ALRMxEN=1`并且闹钟所有位域的值与对应的日子时间值匹配，`ALRMxF`标志位将会置位。

**注意：**当秒字段未被屏蔽时(RTC\_ALRMxTD 寄存器的 `MSKS=0`)，为确保正常运行，RTC\_PSC 寄存器的同步预分频系数 (`FACTOR_S`) 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 `ALRMxEN` 位被置位 3 个 RTC 时钟周期后，`ALRMxF` 位将置位。

### 13.3.5. RTC 初始化和配置

#### RTC 寄存器写保护

在默认情况下，PMU\_CTL寄存器的`BKPWEN`位被清0。所以写RTC寄存器前需要软件提前设置`BKPWEN`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\_WPK寄存器；
2. 写 ‘0x53’到RTC\_WPK寄存器。

写一个错误的值到RTC\_WPK会使写保护再次生效。写保护状态不受系统复位的影响。被写保护的寄存器如下：

RTC\_TIME, RTC\_DATE, RTC\_CTL, RTC\_STAT, RTC\_PSC, RTC\_ALRMxTD,  
RTC\_SHIFTCTL, RTC\_HRFC, RTC\_ALRMxSS。

## 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 INITM 位为 1 进入初始化模式。等待 INITF 位被置 1。
2. 在 RTC\_PSC 寄存器中，设置同步和异步预分频器的分频系数。
3. 在影子寄存器(RTC\_TIME 和 RTC\_DATE)中写初始的日历值，并且通过设置 RTC\_CTL 寄存器的 CS 位来配置时间的格式 (12 或 24 小时制)。
4. 清除 INITM 位退出初始化模式。

大约4个RTC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(BPSHAD=0)，软件应该确保RSYNF位已经置1。

YCM标志表明日历是否完成初始化，该标志会硬件检查日历的年份值。

## 夏令时

通过S1H, A1H和DSM位配置，RTC模块可以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，可以软件配置DSM位来记录这个调节操作。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\_CTL 的 ALRMxEN 位 ( $x=0$ )，禁用闹钟；
2. 设置 Alarm 寄存器(RTC\_ALRMxTD/RTC\_ALRMxSS)；
3. 设置寄存器 RTC\_CTL 的 ALRMxEN 位，使能闹钟功能。

### 13.3.6. 读取日历

#### 当 BPSHAD=0 时，读日历寄存器

当BPSHAD=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。

RSYNF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日历时间和日期。

为了确保这3个值(RTC\_SS, RTC\_TIME, RTC\_DATE)为同一时间，硬件上采取了如下一致性机制：

1. 读RTC\_SS锁定RTC\_TIME和RTC\_DATE的更新；
2. 读RTC\_TIME锁定RTC\_DATE的更新；
3. 读 RTC\_DATE 解锁 RTC\_TIME 和 RTC\_DATE 的更新。

如果想在一个很短的时间间隔内（少于2个RTCCLK）读取日历，应先清除RSYNF位并等待其置位后再读取。

下面几种情况，软件须等待RSYNF置位后才能读日历寄存器（RTC\_SS, RTC\_TIME, RTC\_DATE）：

1. 系统复位之后；
2. 日历初始化之后；
3. 一次移位操作之后。

特别是从低功耗模式唤醒后，软件必须清除RSYNF位并等待RSYNF再次置位后才能读取日历寄存器。

#### 当 BPSHAD=1 时，读日历寄存器

当BPSHAD=1，RSYNF位会被硬件清0，读日历寄存器不需考虑RSYNF位。当前真实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(Deep-sleep/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RSYNF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\_SS/RTC\_TIME/RTC\_DATE)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，那么这个值就是一致的且准确的。

### 13.3.7. RTC 复位

在RTC单元，有两个复位源可用：系统复位和备份域复位。

当系统复位有效时，日历影子寄存器和RTC\_STAT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 真实的日历寄存器；
- RTC 控制寄存器 (RTC\_CTL)；
- RTC 预分频寄存器 (RTC\_PSC)；
- RTC 高精度频率补偿寄存器 (RTC\_HRFC)；
- RTC 移位控制寄存器(RTC\_SHIFTCTL)；
- RTC 时间戳寄存器 (RTC\_SSTS/RTC\_TTS/RTC\_DTS)；
- RTC 侵入寄存器 (RTC\_TAMP)；
- RTC 备份寄存器 (RTC\_BKPx)；
- RTC 闹钟寄存器 (RTC\_ALRMxSS/RTC\_ALRMxTD)。

当系统复位或者进入省电模式的时候，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。

### 13.3.8. RTC 移位功能

当用户有一个高精度的远程时钟而且RTC 1hz 时钟 (ck\_spre) 和远程时钟只有一个亚秒级的偏差，RTC单元提供一个称作移位的功能去消除这个偏差来提高秒钟的精确性。

以二进制格式显示亚秒值，RTC运行时该值是递减计数。因此通过增加RTC\_SHIFTCTL 寄存器的SFS[14: 0]的值到RTC\_SS同步预分频器计数器值SSC[15: 0])或通过增加SFS[14: 0]的值到同步预分频器计数器SSC[15: 0]并且同时置位A1S位，能分别延迟或提前下一秒到达的时间。

RTC\_SS的最大值取决于RTC\_PSC寄存器的FACTOR\_S的值。FACTOR\_S越大，调整的精度也就越高。

因为1Hz的时钟(ck\_spre) 由FACTOR\_A和FACTOR\_S共同产生，越高的FACTOR\_S值就意味着越低的FACTOR\_A值，同时越低的FACTOR\_A意味着越高的功耗。

**注意：**在使用移位功能之前，软件必须检查 RTC\_SS 中 SSC 的第 15 位(SSC[15])并确保该位为 0。写 RTC\_SHIFTCTL 寄存器之后，RTC\_STAT 寄存器的 SOPF 位将会再次置位。当同步移位操作完成时，SOPF 位被硬件清 0。系统复位不影响 SOPF 位。当 REFEN=0 时，移位操作才能正确的工作。如果 REFEN=1，软件禁止写入 RTC\_SHIFTCTL。

### 13.3.9. 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LXTAL有更高精度的外部参考时钟源(50Hz或60Hz)。

使能这项功能之后(REFEN=1)，每一个秒更新的时钟(1Hz)边沿将与最近的RTC\_REFIN参考时钟沿进行对比。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LXTAL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REFEN=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，时间窗时长也不同。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时长的时间窗，当检测状态处于边沿对齐操作时，使用3个ck\_apre时长的时间窗。

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。但是当两个时钟边沿没有对齐时，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时长时间窗内没有发现参考时钟边沿)，日历也能通过LXTAL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个ck\_apre时长时间窗口去检测参考时钟，然后用3个ck\_apre时长时间窗口去调节ck\_spre(1Hz)时钟边沿。

**注意：**使能参考时钟检测功能之前(REFEN=1)，软件必须配置FACTOR\_A为0x7F，FACTOR\_S为0xFF。

待机模式下和数字粗校准时，参考时钟检测功能不可用。

### 13.3.10. RTC 数字平滑校准

RTC平滑校准是一种用于校准RTC频率的方法，该方法通过调整校准周期内的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在一次校准周期内，RTC时钟的脉冲个数增加或者减少了一定的数目。这种校准的分辨率为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。

高精度频率补偿寄存器(RTC\_HRFC)指定了在校准周期内要屏蔽的RTC时钟数目，CMSK[8:0]位能屏蔽0到511个RTC时钟，这样RTC的频率最多降低487.1PPM。

为了提高RTC频率可以设置FREQI位。如果FREQI位被置位，将会有512个额外的RTC时钟周期增加到校准周期(32/16/8秒)时间期间，这意味着每 $2^{11}/2^{10}/2^9$ RTC时钟插入一个RTC时钟周期。

因此使用FREQI可以使RTC频率增加488.5ppm。

同时使用CMSK和FREQI，每个周期时间可以调整-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。

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{\text{cal}} = f_{\text{rtcclk}} \times \left(1 + \frac{FREQI \times 512 - CMSK}{2^N + CMSK - FREQI \times 512}\right) \quad (\text{式13-3})$$

注意： N=20/19 / 18 (32/16/8 秒)校准时间周期。

### 当 FACTOR\_A < 3 时校准：

当异步预分频器值(FACTOR\_A)被设置小于3时，若要使用校准功能，软件不能将FREQI位设置为1。当FACTOR\_A<3， FREQI位设置将会被忽略。

当FACTOR\_A小于3时，FACTOR\_S值应小于标称值。假设RTC时钟频率是正常的 32.768KHz，对应的FACTOR\_S应该按下面所示设置：

FACTOR\_A = 2: FACTOR\_S减少2(8189)

FACTOR\_A = 1: FACTOR\_S减少4(16379)

FACTOR\_A = 0: FACTOR\_S减少8(32759)

当FACTOR\_A小于3， CMSK为0x100，校准频率公式如下：

$$f_{\text{cal}} = f_{\text{rtcclk}} \times \left(1 + \frac{256 - CMSK}{2^N + CMSK - 256}\right) \quad (\text{式 13-4})$$

注意： N=20/19 / 18 (32/16/8 秒)校准时间周期。

### 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(在32秒周期内0.5个RTCCLK)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(在16秒周期内0.5个RTCCLK)之内。

- 校准周期设为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误差在1.907ppm(在8秒周期内0.5个RTCCLK)之内。

## 运行中重校准

当INITF位是0，用下面的步骤，软件可以更新RTC\_HRFC：

- 1). 等待SCPF位置0；
- 2). 写一个新的值到RTC\_HRFC寄存器；
- 3). 3个ck\_apre时钟周期之后，新的校准设置开始生效。

### 13.3.11. 时间戳功能（仅针对 GD32E230xx 产品）

时间戳功能由RTC\_TS管脚输入，通过配置TSEN位来使能。

当RTC\_TS管脚检测到时间戳事件发生时，会将日历的值保存在时间戳寄存器中( RTC\_DTS/RTC\_TTS/RTC\_SSTS)，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，来增加时间戳事件的触发源：设置TPTS=1，使得侵入检测功能的侵入事件同时也作为时间戳事件的输入源。

**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。

### 13.3.12. 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滤波功能的电平检测模式。

#### RTC 备份寄存器( (RTC\_BKPx)

RTC备份寄存器处于V<sub>DD</sub>备份域中。从待机模式唤醒或系统复位操作都不会影响这些寄存器。

只有当被检测到有侵入事件和备份域复位时，这些寄存器复位。.

#### 初始化侵入检测功能

TPxEN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TPxEN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(TPxF)将会置位。如果侵入事件中断使能被启用(TPIE)，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\_BKPx)复位。

#### 侵入事件源的时间戳

使能TPTS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也将被置位，如果使能了时间戳功能。当检测到侵入事件时，无论TPTS位的值如何，TPxF位将置位。

### 侵入事件检测为边沿检测模式

当FLT位为0x0时，侵入检测被设置成边沿检测模式，TPxE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，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\_BKPx)，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（仅针对 GD32E230xx 产品，GD32E231xx 产品无 PC13 引脚）。

### 侵入事件检测为带可配置滤波功能的电平检测模式

当FLT位没有被设置成0x0时，侵入检测被设置成电平检测模式，FLT位决定有效电平需连续采样的次数(2, 4或者 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。

#### 13.3.13. 校准时钟输出

如果COEN位设置为1，RTC\_OUT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(FACTOR\_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：

$$f_{rtc\_calib} = \frac{f_{rtcclk}}{(FACTOR\_A+1) \times (FACTOR\_S+1)} \quad (式 13-5)$$

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。

#### 13.3.14. 闹钟输出

当OS控制位被设置为0x01时，RTC\_ALARM复用输出功能被启用。这个功能将直接输出RTC\_STAT寄存器的ALRMxF(x=0)值。

RTC\_CTL寄存器中的OPOL位可以配置ALRMxF位或者WTF位输出时候的极性，因此RTC\_ALARM的输出电平有可能与相应的位值相反。

### 13.3.15. RTC 省电模式管理

表 13-1. 省电模式管理

GD32E230xx 产品

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LXTAL或IRC40K时可以工作	RTC闹钟/侵入事件/时间戳事件
待机模式	当时钟源是LXTAL或IRC40K时可以工作	RTC闹钟/侵入事件/时间戳事件

GD32E231xx 产品

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LXTAL或IRC40K时可以工作	RTC闹钟/侵入事件
待机模式	当时钟源是LXTAL或IRC40K时可以工作	RTC闹钟/侵入事件

### 13.3.16. 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/侵入事件/时间戳中断，应按下面步骤操作：

1. 设置并使能对应的 EXTI 中接到 RTC 闹钟/侵入事件/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 闹钟/侵入事件/时间戳的全局中断；
3. 配置并使能 RTC 闹钟/侵入事件/时间戳功能。

表 13-2. 中断控制

GD32E230xx 产品

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式	退出待机模式
闹钟 0	ALRM0F	ALRM0IE	Y	Y(*)	Y(*)
时间戳	TSF	TSIE	Y	Y(*)	Y(*)
侵入 0	TP0F	TPIE	Y	Y(*)	Y(*)

侵入 1	TP1F	TPIE	Y	Y(*)	Y(*)
------	------	------	---	------	------

\* : 仅当RTC时钟源是LXTAL或IRC40K时有效。

#### GD32E231xx 产品

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式	退出待机模式
闹钟 0	ALRM0F	ALRM0IE	Y	Y(*)	Y(*)
侵入 1	TP1F	TPIE	Y	Y(*)	Y(*)

\* : 仅当RTC时钟源是LXTAL或IRC40K时有效。

## 13.4. RTC 寄存器

RTC 基地址: 0x4000 2800

### 13.4.1. 时间寄存器 (RTC\_TIME)

偏移地址: 0x00

系统复位值: 当 BPSHAD = 0, 0x0000 0000

当 BPSHAD = 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								PM	HRT[1:0]		HRU[3:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MNT[2:0]			MNU[3:0]			保留	SCT[2:0]			SCU[3:0]				
	rw			rw				rw			rw				

位/位域	名称	描述
31:23	保留	必须保持复位值
22	PM	AM/PM 标志 0: AM 或 24 小时制 1: PM
21:20	HRT[1:0]	小时十位值, 以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值, 以 BCD 码形式存储
15:	保留	必须保持复位值
14:12	MNT[2:0]	分钟十位值, 以 BCD 码形式存储
11:8	MNU[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	SCT[2:0]	秒钟十位值, 以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值, 以 BCD 码形式存储

### 13.4.2. 日期寄存器 (RTC\_DATE)

偏移地址: 0x04

系统复位值: 当 BPSHAD = 0, 0x0000 2101

当 BPSHAD = 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								YRT[3:0]				YRU[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]	MONT	MONU[3:0]				保留		DAYT[1:0]		DAYU[3:0]					
rw	rw	rw				rw		rw		rw					

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	YRT[3:0]	年份十位值，以 BCD 码形式存储
19:16	YRU[3:0]	年份个位值，以 BCD 码形式存储
15:13	DOW[2:0]	星期 0x0: 保留 0x1: 星期一 ... 0x7: 星期日
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[3:0]	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	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

### 13.4.3. 控制寄存器 (RTC\_CTL)

#### GD32E230xx 产品

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	DSM	S1H	A1H	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	保留	ALRMOIE	TSEN	保留	ALRMOE_N	保留	CS	BPSHAD	REFEN	TSEG	保留				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	w			

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 该位用来选择输出的标志源。 0x0: 禁用 RTC_ALARM 输出 0x1: 启用闹钟 0 标志输出
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	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
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	ALRM0IE	RTC 闹钟 0 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	ALRMOEN	闹钟 0 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	CS	时间格式 0: 24 小时制 1: 12 小时制 注意: 仅能在初始化状态进行写入
5	BPSHAD	禁止影子寄存器 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 注意: 如果 APB1 时钟的频率小于 RTCCLK 频率的 7 倍, 该位必须设为 1
4	REFEN	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 注意: 仅能在初始化状态进行写入并且 FACTOR_S 必须为 0x00FF
3	TSEG	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

**GD32E231xx 产品**

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														DSM	S1H
rw w w														A1H	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ALRMOIE	保留	ALRMOEN	保留	CS	BPSHAD	REFEN	保留	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:19	保留	必须保持复位值

18	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
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
15:13	保留	必须保持复位值
12	ALRMOIE	RTC 闹钟 0 中断使能 0: 禁用闹钟中断 1: 启用闹钟中断
11:9	保留	必须保持复位值
8	ALRM0EN	闹钟 0 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	CS	时间格式 0: 24 小时制 1: 12 小时制 注意：仅能在初始化状态进行写入
5	BPSHAD	禁止影子寄存器 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 注意：如果 APB1 时钟的频率小于 RTCCLK 频率的 7 倍，该位必须设为 1
4	REFEN	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 注意：仅能在初始化状态进行写入并且 FACTOR_S 必须为 0x00FF
3:0	保留	必须保持复位值

#### 13.4.4. 状态寄存器 (RTC\_STAT)

**GD32E230xx 产品**

偏移地址: 0x0C

系统复位：仅 INITM, INITF 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\_STAT[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															SCPF
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TP1F	TP0F	TSOVRF	TSF	保留	ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	保留	ALRM0WF		
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rw	r	rc_w0	r	r				r

位/位域	名称	描述
31:17	保留	必须保持复位值
16	SCPF	平滑校准挂起标志 在未进入初始化模式时向 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑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	TP1F	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。可以通过向该位软件写 0 来清除。
13	TP0F	RTC_TAMP0 事件标志 当在 tamper0 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
12	TSOVRF	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。 可以通过向该位软件写 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。可以通过向该位软件写 0 来清除。
10:9	保留	必须保持复位值
8	ALRM0F	Alarm0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
7	INITM	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/日期和预分频，计数器将停止运行
6	INITF	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(INITM)，移位操作挂起标志(SOPF)或者禁止影子寄存器模式(BPSHAD = 1)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	YCM	年份配置标志 当日历寄存器的年份值不为 0 时硬件置 1 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	ALRM0WF	Alarm0 配置可写标志 硬件置位和清零。ALRM0EN=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

### GD32E231xx 产品

偏移地址: 0x0C

系统复位: 仅 INITM, INITF 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\_STAT[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															SCPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TP1F		保留		ALRM0F	INITM	INITF	RSYNF	YCM	SOPF		保留		ALRM0WF	
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r					r

位/位域	名称	描述
31:17	保留	必须保持复位值
16	SCPF	平滑校准挂起标志 在未进入初始化模式时向 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑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	TP1F	<p><b>RTC_TAMP1</b> 事件标志</p> <p>当在 <b>tamper1</b> 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。</p>
13:9	保留	必须保持复位值
8	ALRM0F	<p><b>Alarm0</b> 发生标志</p> <p>当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。</p>
7	INITM	<p>进入初始化模式</p> <p>0: 自由运行模式</p> <p>1: 进入初始化模式设置时间/日期和预分频，计数器将停止运行</p>
6	INITF	<p>初始化状态标志</p> <p>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。</p> <p>0: 日历寄存器和预分频器的值不能改变</p> <p>1: 日历寄存器和预分频器的值可以改变</p>
5	RSYNF	<p>寄存器同步标志</p> <p>每 2 个 <b>RTCCLK</b> 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(<b>INITM</b>)，移位操作挂起标志(<b>SOPF</b>)或者禁止影子寄存器模式(<b>BPSHAD = 1</b>)会清除该位。该位也可以通过软件写 0 清除。</p> <p>0: 影子寄存器未同步</p> <p>1: 影子寄存器已同步</p>
4	YCM	<p>年份配置标志</p> <p>当日历寄存器的年份值不为 0 时硬件置 1</p> <p>0: 日历尚未初始化</p> <p>1: 日历已经初始化</p>
3	SOPF	<p>移位功能操作挂起标志</p> <p>0: 移位操作没有挂起</p> <p>1: 移位操作挂起</p>
2:1	保留	必须保持复位值
0	ALRM0WF	<p><b>Alarm0</b> 配置可写标志</p> <p>硬件置位和清零。 <b>ALRM0EN=0</b> 时，标记 <b>alarm</b> 是否可写。</p> <p>0: 不允许修改 <b>Alarm</b> 寄存器设置</p> <p>1: 允许修改 <b>Alarm</b> 寄存器设置</p>

### 13.4.5. 预分频寄存器 (**RTC\_PSC**)

偏移地址: 0x10

系统复位: 无影响

备份域复位值: 0x007F 00FF

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

位/位域	名称	描述
31:23	保留	必须保持复位值
22:16	FACTOR_A[6:0]	异步预分频系数 $ck\_apre$ 频率 = RTCCLK 频率/(FACTOR_A+1)
15	保留	必须保持复位值
14:0	FACTOR_S[14:0]	同步预分频系数 $ck\_spre$ 频率 = $ck\_apre$ 频率/(FACTOR_S+1)

### 13.4.6. 闹钟 0 时间日期寄存器 (RTC\_ALRM0TD)

偏移地址: 0x1C

系统复位: 无影响

备份域复位值: 0x0000 0000

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	
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]			MSKH	PM	HRT[1:0]		HRU[3:0]					
rw	rw	rw		rw			rw	rw	rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MSKM	MNT[2:0]		MNU[3:0]			MSKS	SCT[2:0]		SCU[3:0]							
rw	rw	rw		rw			rw	rw	rw		rw					

位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0: 不屏蔽日期/天位域 1: 屏蔽日期/天位域
30	DOWS	星期选择 0: 此时 DAYU[3: 0] 代表日期个位值 1: 此时 DAYU[3: 0] 代表星期几, 此时 DAYT[1: 0]无意义
29:28	DAYT[1:0]	日期十位值, 以 BCD 码格式存储

27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0: 不屏蔽小时位域 1: 屏蔽小时位域
22	PM	AM/PM 标志 0: AM 或 24 小时制 1: PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0: 不屏蔽分钟位域 1: 屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0: 不屏蔽秒位域 1: 屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

### 13.4.7. 写保护钥匙寄存器 (RTC\_WPK)

偏移地址: 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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								WPK[7:0]							

w

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WPK[7:0]	写保护的解锁值

### 13.4.8. 亚秒寄存器(RTC\_SS)

偏移地址: 0x28

系统复位值: 当BPSHAD = 0, 0x0000 0000。

当BPSHAD = 1, 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															

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SSC[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: $\text{秒的小数部分} = (\text{FACTOR\_S} - \text{SSC}) / (\text{FACTOR\_S} + 1)$

### 13.4.9. 移位控制寄存器 (RTC\_SHIFTCTL)

偏移地址: 0x2C

系统复位: 无影响

备份域复位值: 0x0000 0000

写保护寄存器, 仅当SOPF=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, 由于同步预分频器是一个递减计数器, 所以时钟将会延迟。

**延迟(秒) = SFS / ( FACTOR\_S + 1 )**

当 **A1S** 和 **SFS** 一起使用时, 时钟将会提前

**提前(秒) = ( 1 - ( SFS / ( FACTOR\_S + 1 ) ) )**

**注意:** 写入此寄存器会导致 **RSYNF** 位被清 0。

#### 13.4.10. 时间戳时间寄存器 (**RTC\_TTS**)

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								PM	HRT[1:0]		HRU[3:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MNT[2:0]		MNU[3:0]			保留	SCT[2:0]		SCU[3:0]						
	r			r					r			r		r	

位/位域	名称	描述
31:23	保留	必须保持复位值
22	PM	AM/PM 标记 0: AM 或 24 小时制 1: PM
21:20	HRT[1:0]	小时十位值, 以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值, 以 BCD 码形式存储
15	保留	必须保持复位值
14:12	MNT[2:0]	分钟十位值, 以 BCD 码形式存储
11:8	MNU[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	SCT[2:0]	秒钟十位值, 以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值, 以 BCD 码形式存储

### 13.4.11. 时间戳日期寄存器 (RTC\_DTS)

偏移地址: 0x34

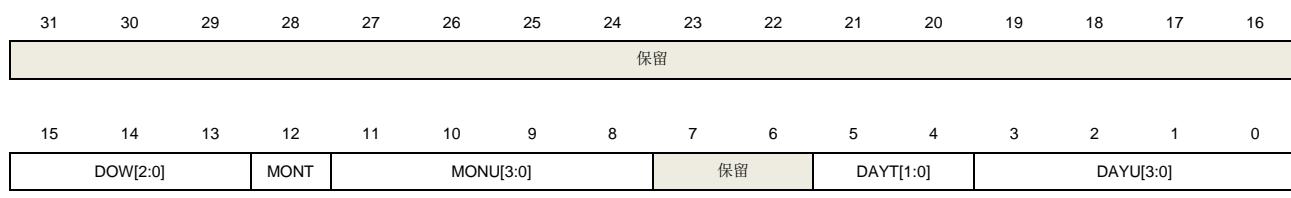
备份域复位值: 0x0000 0000

系统复位: 无影响

当TSF被置1, 该位用来记录日历日期

清除TSF位也会清除此寄存器

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
15:13	DOW[2:0]	星期数
12	MONT	月份十位值, 以 BCD 码形式存储
11:8	MONU[3:0]	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	DAYT[1:0]	日期十位值, 以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值, 以 BCD 码形式存储

### 13.4.12. 时间戳亚秒寄存器 (RTC\_SSTS)

偏移地址: 0x38

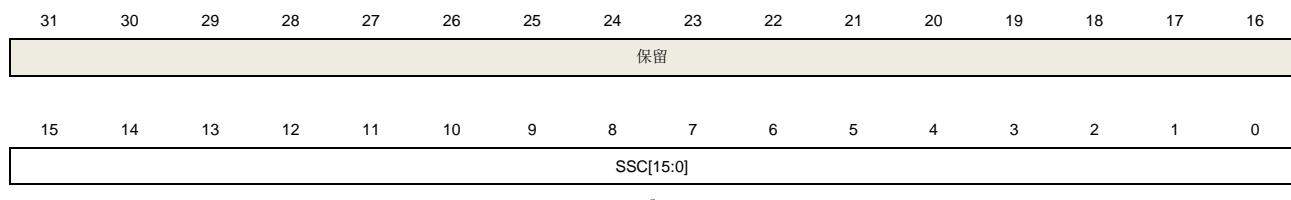
备份域复位值: 0x0000 0000

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



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SSC[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

### 13.4.13. 高精度频率补偿寄存器 (RTC\_HRFC)

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
FREQI	CWND8	CWND16	保留				CMSK[8:0]								rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	FREQI	RTC 频率增加 488.5ppm 0: 无影响 1: 每 211 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 * FREQI) - CMSK
14	CWND8	采用 8 秒校准周期 0: 无影响 1: 采用 8 秒校准周期 注意: 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 注意: 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 220 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

### 13.4.14. 侵入寄存器 (RTC\_TAMP)

#### GD32E230xx 产品

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				保留	
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
DISPU	PRCH[1:0]	FLT[1:0]	FREQ[2:0]	TPTS	保留		TP1EG	TP1EN	TP1IE	TP0EG	TP0EN				
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	PC15MDE	PC15 模式 0: 无影响 1: 如果 LXTAL 禁用, 强制 PC15 推挽输出
22	PC15VAL	PC15 值 当 LXTAL 禁用且 PC15MDE=1 时, PC15 输出该位数据
21	PC14MDE	PC14 模式 0: 无影响 1: 如果 LXTAL 禁用, 强制 PC14 推挽输出
20	PC14VAL	PC14 值 当 LXTAL 禁用且 PC14MDE=1 时, PC14 输出该位数据
19	PC13MDE	PC13 模式 0: 无影响 1: 当 RTC 所有备用功能禁用时, PC13 输出该位数据
18	PC13VAL	PC13 值或闹钟输出类型值 PC13 输出闹钟 0: PC13 开漏输出 1: PC13 推挽输出 当 RTC 所有备用功能禁用且 PC13MDE=1 时 0: PC13 输出 0 1: PC13 输出 1
17:16	保留	必须保持复位值

15	DISPU	RTC_TAMPx 上拉禁用位 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 1: 禁用预充电功能
14:13	PRCH[1:0]	RTC_TAMPx 的预充电时间 该位设置决定了每次采样前的预充电时间 0x0: 1 个 RTC 时钟 0x1: 2 个 RTC 时钟 0x2: 4 个 RTC 时钟 0x3: 8 个 RTC 时钟
12:11	FLT[1:0]	RTC_TAMPx 过滤器计数设置 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。 0x0: 用边沿模式检测侵入事件，预充电功能被自动禁用。 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 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 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
10:8	FREQ[2:0]	侵入事件电平模式检测的采样频率 0x0: 每次采样间隔 32768 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 1Hz) 0x1: 每次采样间隔 16384 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 2Hz) 0x2: 每次采样间隔 8192 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 4Hz) 0x3: 每次采样间隔 4096 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 8Hz) 0x4: 每次采样间隔 2048 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 16Hz) 0x5: 每次采样间隔 1024 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 32Hz) 0x6: 每次采样间隔 512 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 64Hz) 0x7: 每次采样间隔 256 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 128Hz)
7	TPTS	侵入事件时触发时间戳 0: 无影响 1: 当检测到侵入事件时, 即使 TSEN=0, TSF 也会被置位
6:5	保留	必须保持复位值
4	TP1EG	TAMP1 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式( <b>FLT =0</b> ): 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( <b>FLT !=0</b> ): 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	TP1EN	Tamper1 检测使能位 0: 禁用 Tamper1 检测功能 1: 启用 Tamper1 检测功能
2	TPIE	侵入检测中断使能 0: 禁用侵入中断

**1: 启用侵入中断**

1	TP0EG	TAMPO 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =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(FLT !=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	TP0EN	Tamper0 检测使能位 0: 禁用 Tamper0 检测功能 1: 启用 Tamper0 检测功能

**注意:** 强烈建议在改变侵入检测配置之前, 应该复位 TPxEN 位。

**GD32E231xx 产品**

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	保留			
rw								rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]	FLT[1:0]	FREQ[2:0]	TPTS	保留		TP1EG	TP1EN	TP1IE	保留					
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: 无影响 1: 如果 LXTAL 禁用, 强制 PC15 推挽输出
22	PC15VAL	PC15 值 当 LXTAL 禁用且 PC15MDE=1 时, PC15 输出该位数据
21	PC14MDE	PC14 模式 0: 无影响 1: 如果 LXTAL 禁用, 强制 PC14 推挽输出
20	PC14VAL	PC14 值 当 LXTAL 禁用且 PC14MDE=1 时, PC14 输出该位数据

19:16	保留	必须保持复位值
15	DISPU	<p><b>RTC_TAMPx</b> 上拉禁用位</p> <p>0: 使能内部 <b>RTC_TAMPx</b> 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><b>RTC_TAMPx</b> 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 个 RTC 时钟</p> <p>0x1: 2 个 RTC 时钟</p> <p>0x2: 4 个 RTC 时钟</p> <p>0x3: 8 个 RTC 时钟</p>
12:11	FLT[1:0]	<p><b>RTC_TAMPx</b> 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	TPTS	<p>侵入事件时触发时间戳</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	TP1EG	<p><b>TAMP1</b> 输入管脚的侵入事件检测触发沿</p> <p>如果侵入检测处于边沿模式(<b>FLT</b> =0):</p> <p>0: 上升沿触发一个侵入检测事件</p> <p>1: 下降沿触发一个侵入检测事件</p> <p>如果侵入检测处于电平模式(<b>FLT</b> !=0):</p> <p>0: 低电平触发一个侵入检测事件</p> <p>1: 高电平触发一个侵入检测事件</p>
3	TP1EN	<p><b>Tamper1</b> 检测使能位</p> <p>0: 禁用 <b>Tamper1</b> 检测功能</p> <p>1: 启用 <b>Tamper1</b> 检测功能</p>

---

2	TPIE	侵入检测中断使能
		0: 禁用侵入中断
		1: 启用侵入中断

1:0 保留 必须保持复位值

**注意：** 强烈建议在改变侵入检测配置之前，应该复位 **TPxEN** 位。

### 13.4.15. 闹钟 0 亚秒寄存器 (RTC\_ALRM0SS)

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ALRM0EN=0或INITM=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				MSKSSC[3:0]				保留							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		SSC[14:0]				rw									

位/位域	名称	描述
31:28	保留	必须保持复位值
27:24	MSKSSC[3:0]	<p>亚秒位域的屏蔽控制位</p> <p>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。</p> <p>0x1: SSC[0]位用于时间匹配，其他位被忽略。</p> <p>0x2: SSC[1: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x3: SSC[2: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x4: SSC[3: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x5: SSC[4: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x6: SSC[5: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x7: SSC[6: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x8: SSC[7: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x9: SSC[8: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x10: SSC[9: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x11: SSC[10: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x12: SSC[11: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x13: SSC[12: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x14: SSC[13: 0] 位用于时间匹配，其他位被忽略。</p> <p>0x15: SSC[14: 0] 位用于时间匹配，其他位被忽略。</p>

注意：同步预分频计数器的第 15 位(RTC\_SS 寄存器中的 SSC[15])从不被匹配。

---

23:15	保留	必须保持复位值
14:0	<b>SSC[14:0]</b>	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 <b>MSKSSC</b> 位控制。

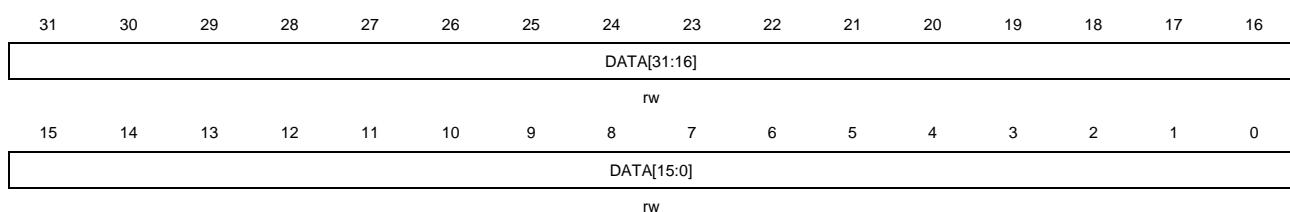
### 13.4.16. 备份寄存器 (**RTC\_BKPx**) (x=0..4)

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




---

位/位域	名称	描述
31:0	<b>DATA[31:0]</b>	数据 软件可读写寄存器。寄存器值在省电模式下依然保持有效。当侵入检测标志位 <b>TPxF</b> 置 1，这些寄存器会被复位。当 <b>FMC</b> 读保护功能禁用时，这些寄存器会被复位。

## 14. 定时器 (TIMER)

表 14-1. 定时器 (TIMERx) 分为六种类型

定时器	定时器 0	定时器 2	定时器 13	定时器 14	定时器 15/16	定时器 5
类型	高级	通用 L0	通用 L2	通用 L3	通用 L4	基本
预分频器	16 位	16 位	16 位	16 位	16 位	16 位
计数器	16 位	16 位	16 位	16 位	16 位	16 位
计数模式	向上, 向下, 中央对齐	向上, 向下, 中央对齐	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	1	2	1	0
互补和 死区时间	●	×	×	●	●	×
中止输入	●	×	×	●	●	×
单脉冲	●	●	×	●	●	●
正交译码器	●	●	×	×	×	×
主-从管理	●	●	×	●	×	×
内部连接	● <sup>(1)</sup>	● <sup>(2)</sup>	×	● <sup>(3)</sup>	×	×
DMA	●	●	×	●	●	● <sup>(4)</sup>
Debug 模式	●	●	●	●	●	●

(1) TIMER0 ITIO: TIMER14\_TRGO ITI1: 0 ITI2: TIMER2\_TRGO ITI3: 0

(2) TIMER2 ITIO: TIMER0\_TRGO ITI1: 0 ITI2: TIMER14\_TRGO ITI3: 0

(3) TIMER14 ITIO: 0 ITI1: TIMER2\_TRGO ITI2: 0 ITI3: 0

(4) 只有更新事件可以产生 DMA 请求, 定时器 5 中没有 DMAS 位(DMA 请求源选择位)。

## 14.1. 高级定时器 (TIMERx,x=0)

### 14.1.1. 简介

高级定时器 (TIMER0) 是四通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。高级定时器含有一个16位无符号计数器。

高级定时器是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

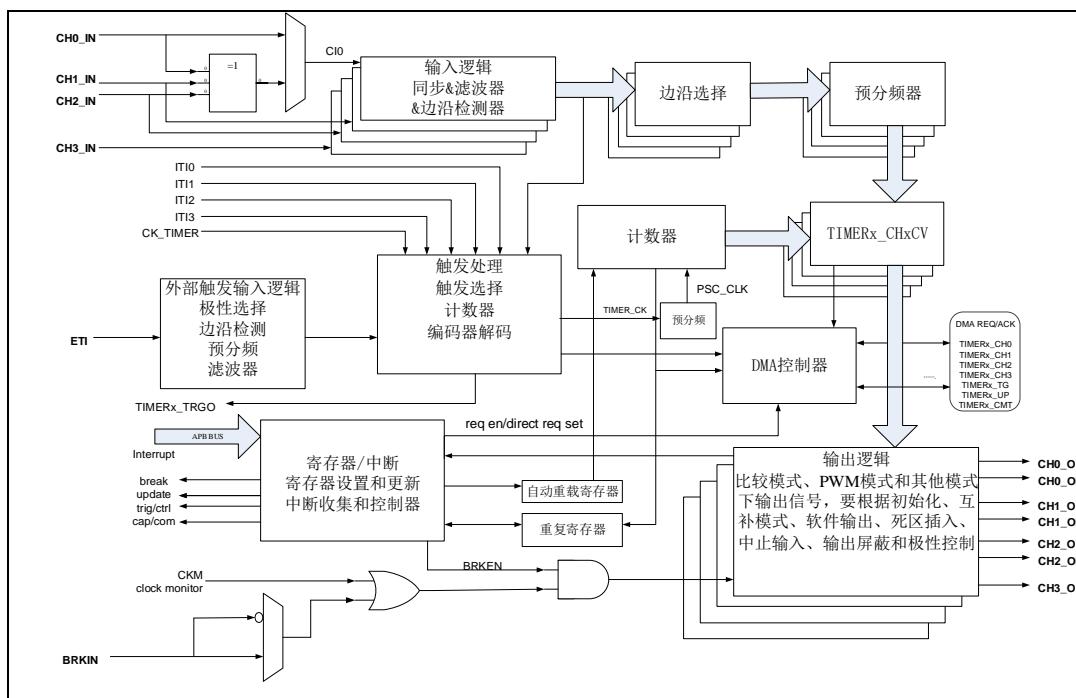
### 14.1.2. 主要特征

- 总通道数： 4;
- 计数器宽度： 16位;
- 定时器时钟源可选： 内部时钟，内部触发，外部输入，外部触发;
- 多种计数模式： 向上计数，向下计数和中央计数;
- 正交编码器接口： 用来追踪运动和分辨旋转方向和位置;
- 霍尔传感器接口： 用来做三相电机控制;
- 可编程的预分频器： 16位。运行时可以被改变;
- 每个通道可配置： 输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和DMA请求： 更新事件，触发事件，比较/捕获事件，换相事件和中止事件;
- 多个定时器的菊花链使得一个定时器可以同时启动多个定时器;
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数;
- 定时器主-从管理。

### 14.1.3. 结构框图

[图14-1. 高级定时器结构框图](#)提供了高级定时器的内部配置细节。

图 14-1. 高级定时器结构框图



#### 14.1.4. 功能说明

##### 时钟源配置

高级定时器的时钟源可以是内部时钟源CK\_TIMER，或者是由SMC (TIMERx\_SMCFG寄存器位[2:0]) 位确定的时钟源。

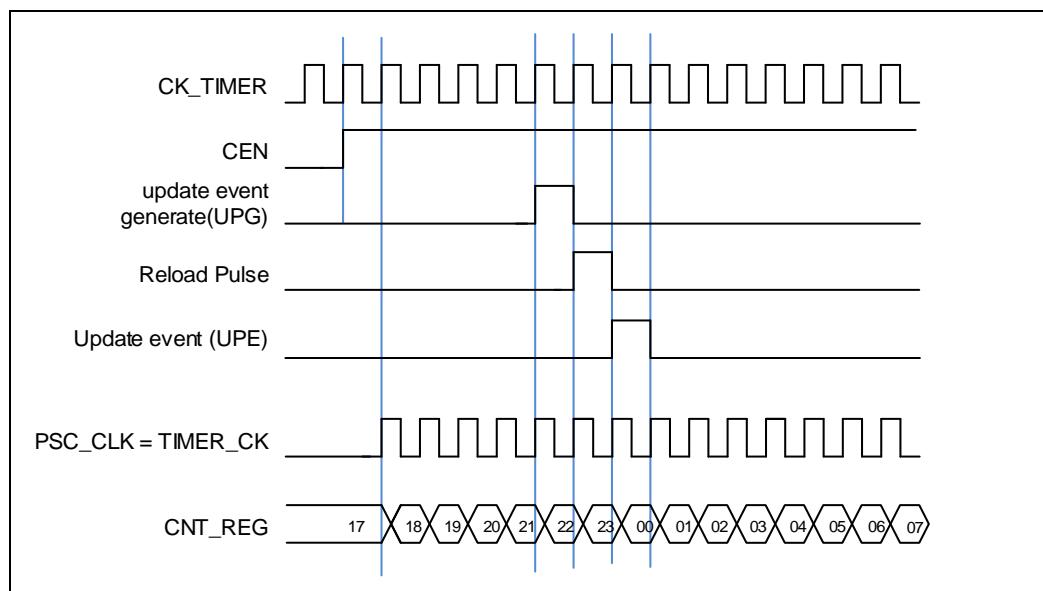
- SMC[2:0]=3'b000, 定时器选择内部时钟源（连接到RCU模块的CK\_TIMER）

如果SMC[2:0]=3'b000，默认用来驱动计数器预分频器的是内部时钟源CK\_TIMER。当CEN置位，CK\_TIMER经过预分频器（预分频值由TIMERx\_PSC寄存器确定）产生PSC\_CLK。

这种模式下，驱动预分频器计数的TIMER\_CK等于来自于RCU模块的CK\_TIMER。

如果将TIMERx\_SMCFG寄存器的SMC[2:0]设置为0x1、0x2、0x3和0x7，预分频器被其他时钟源(由TIMERx\_SMCFG寄存器的TRGS[2:0]区域选择)驱动，在下文说明。当SMC位被设置为0x4、0x5和0x6，计数器预分频器时钟源由内部时钟CK\_TIMER驱动。

图 14-2. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]=3'b111 (外部时钟模式0)，定时器选择外部输入引脚作为时钟源。

计数器预分频器可以在TIMERx\_CH0/TIMERx\_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。Clx是TIMERx\_Clx通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号ITI0/1/2/3的上升沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x0, 0x1, 0x2 或者0x3。

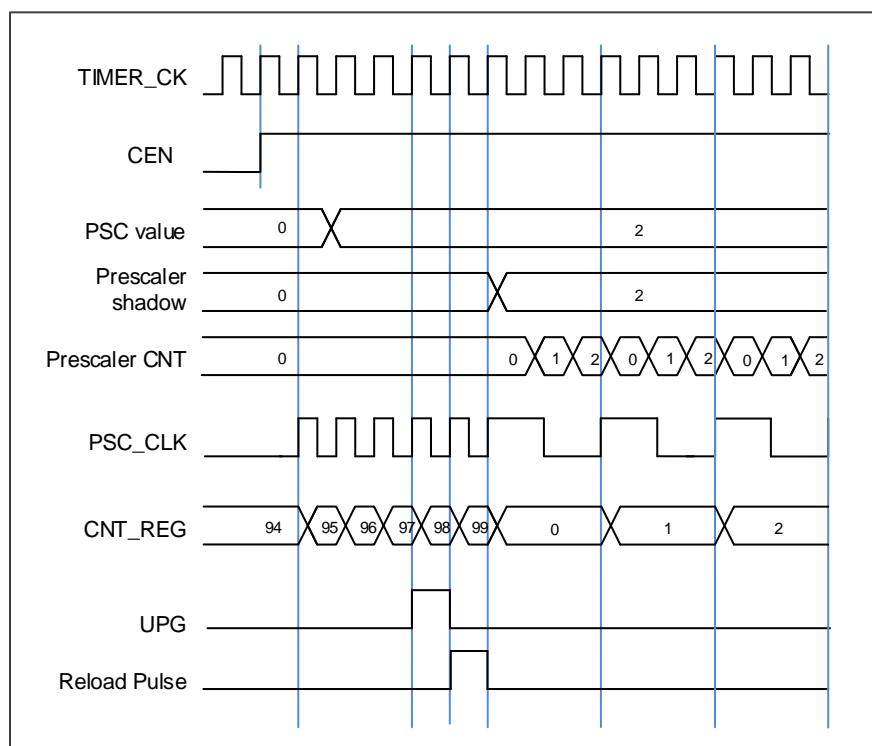
- SMC1=1'b1 (外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源。

计数器预分频器可以在外部引脚ETI的每个上升沿或下降沿计数。这种模式可以通过设置TIMERx\_SMCFG寄存器中的SMC1位为1来选择。另一种选择ETI信号作为时钟源方式是，设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x7。注意ETI信号是通过数字滤波器采样ETI引脚得到的。如果选择ETI信号为时钟源，触发控制器包括边沿监测电路将在每个ETI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。

## 时钟预分频器

预分频器可以将定时器的时钟(TIMER\_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC\_CLK驱动计数器计数。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 14-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图



### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到计数器自动加载值（定义在 **TIMERx\_CAR** 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数，并且产生上溢事件。另外，在(**TIMERx\_CREP+1**)次上溢后将会产生更新事件。在向上计数模式中，**TIMERx\_CTL0** 寄存器中的计数方向控制位 **DIR** 应该被设置成 0。

当通过**TIMERx\_SWEVG**寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果**TIMERx\_CTL0**寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

[图14-4. 向上计数时序图, PSC=0/2](#) 和 [图14-5. 向上计数时序图, 在运行时改变TIMERx\\_CAR寄存器的值](#)给出了一些例子，当 **TIMERx\_CAR=0x99** 时，计数器在不同预分频因子下的行为。

图 14-4. 向上计数时序图, PSC=0/2

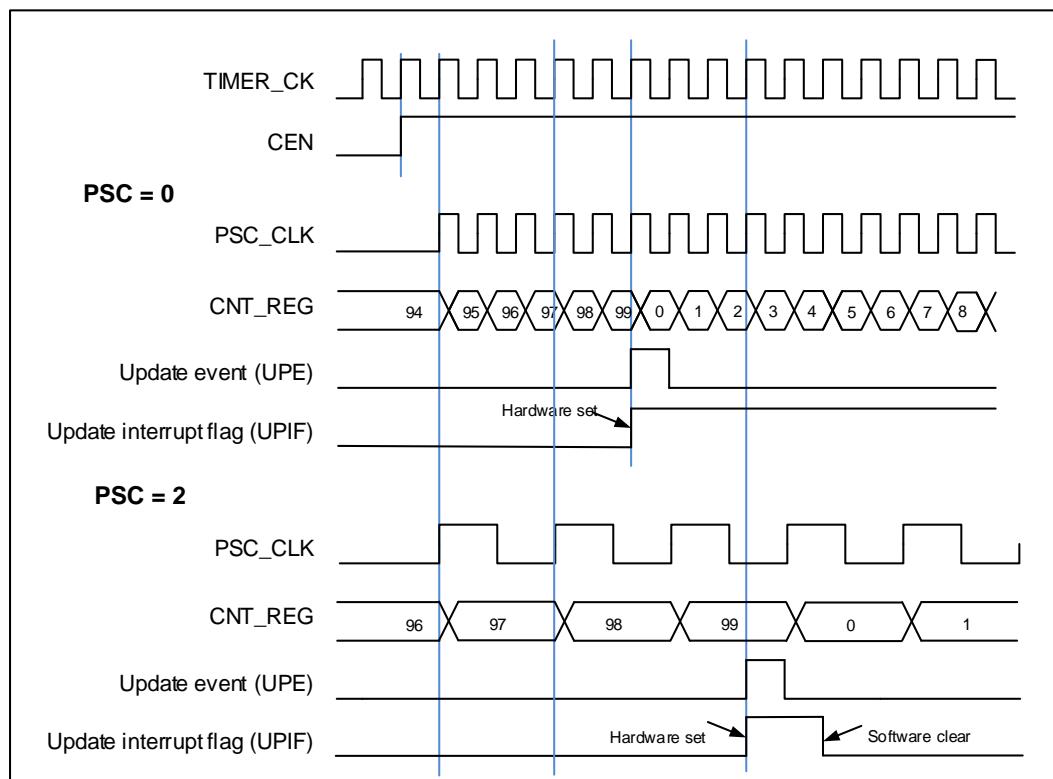
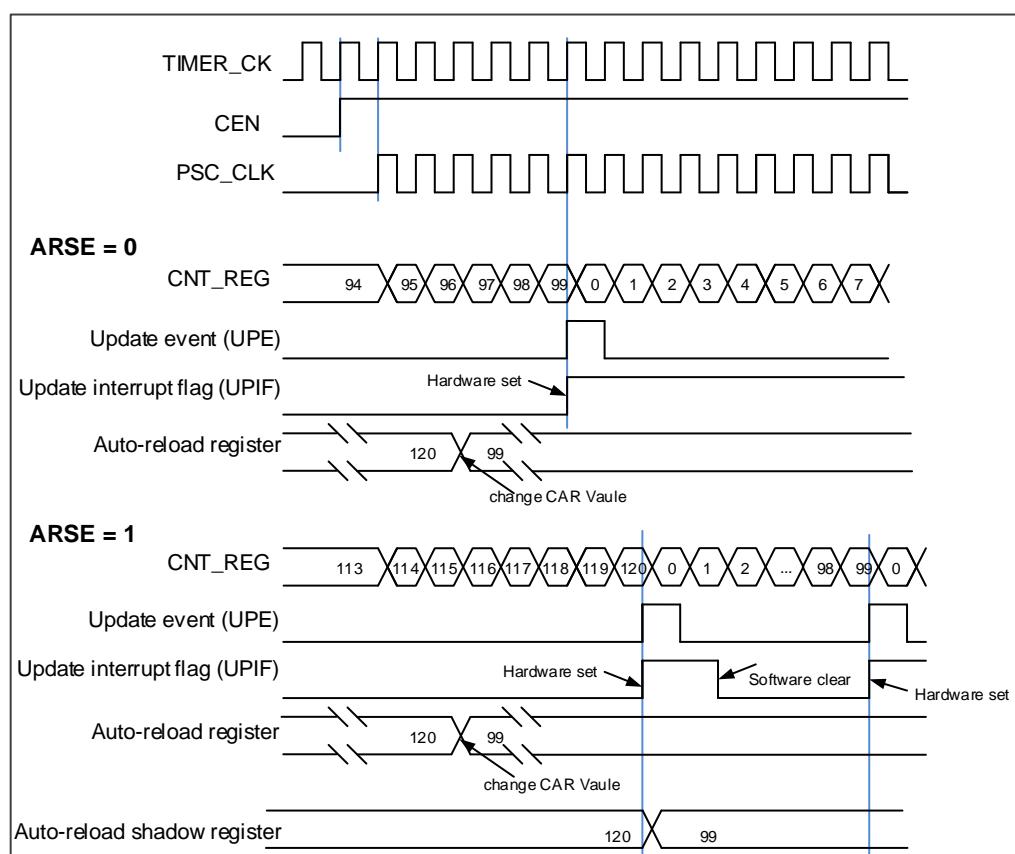


图 14-5. 向上计数时序图, 在运行时改变 TIMERx\_CAR 寄存器的值



## 计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx\_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数。如果设置了重复计数器，在(TIMERx\_CREP+1)次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，TIMERx\_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

当通过TIMERx\_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果TIMERx\_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

[图 14-6. 向下计数时序图，PSC=0/2](#) 和 [图 14-7. 向下计数时序图，在运行时改变 TIMERx CAR 寄存器值](#)给出了一些例子，当 TIMERx\_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 14-6. 向下计数时序图，PSC=0/2

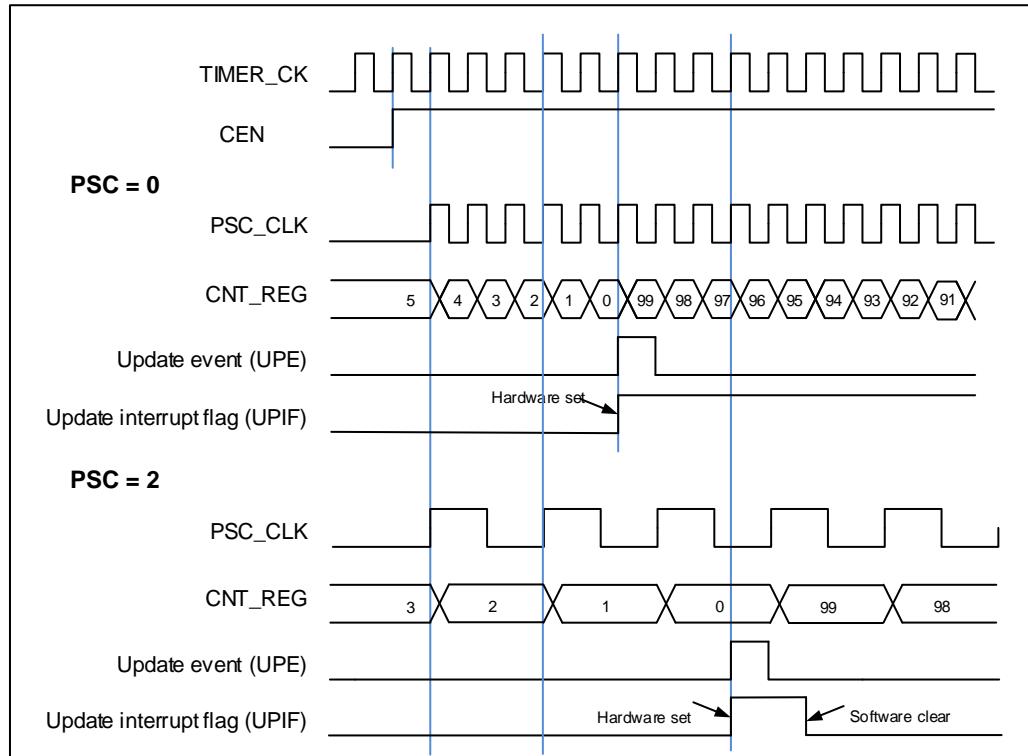
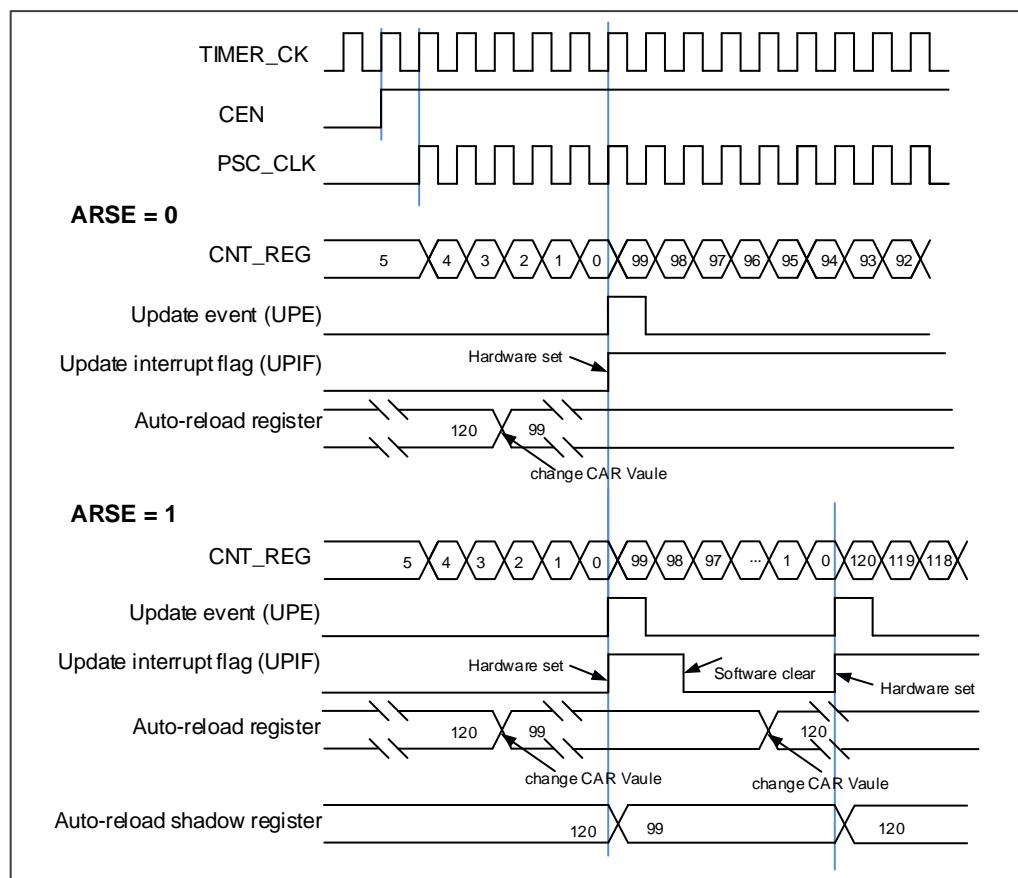


图 14-7. 向下计数时序图，在运行时改变 TIMERx\_CAR 寄存器值



### 计数器中央对齐模式

在中央对齐计数模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到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。在中央对齐计数模式中，TIMERx\_CTL0寄存器中的计数方向控制位DIR只读，指示了当前的计数方向。

将TIMERx\_SWEVG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。

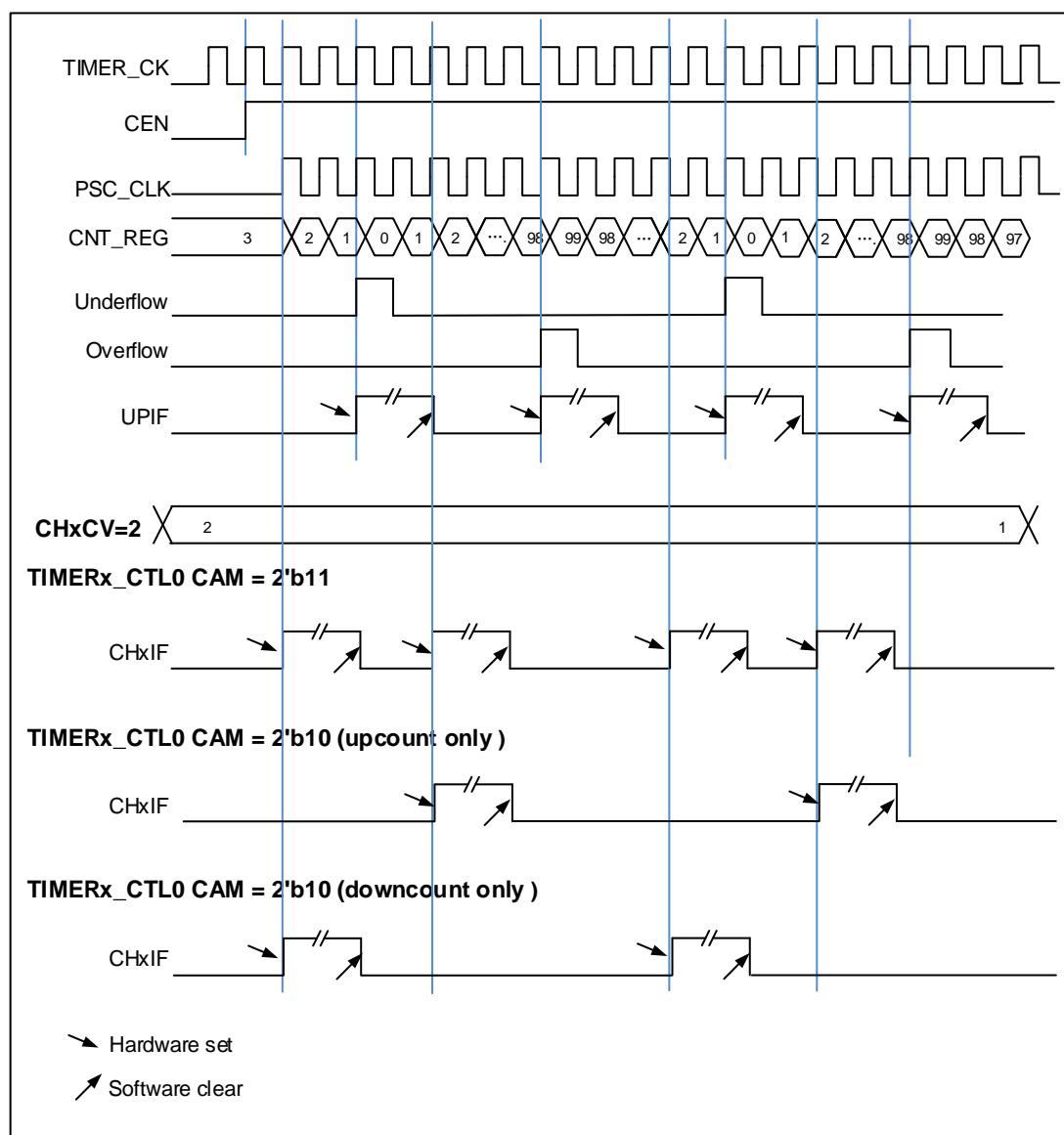
上溢或者下溢时，TIMERx\_INTF寄存器中的UPIF位都会被置1。但是CHxIF位是否置1与TIMERx\_CTL0寄存器中CAM的值有关。具体细节参考[图14-8. 中央计数模式计数器时序图](#)。

如果TIMERx\_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

[图14-8. 中央计数模式计数器时序图](#)给出了一些例子，当TIMERx\_CAR=0x99，TIMERx\_PSC=0x0时，计数器的时序图。

图 14-8. 中央计数模式计数器时序图



### 更新事件（来自上溢/下溢）频率配置

更新事件的生成频率（来自上溢和下溢事件）可以通过**TIMEx\_CREP**寄存器进行配置。重复计数器是用来在N+1个计数周期之后产生更新事件，更新定时器的寄存器，N为**TIMEx\_CREP**寄存器的CREP。重复计数器在每次计数器上溢和下溢时递减（向上计数模式中不存在下溢事件；向下计数模式中不存在上溢事件）。

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

新写入的 CREP 值将在下一次更新事件到来时生效。当 CREP 的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的 CREP 值何时生效。如果在写入奇数到 CREP 寄存器后由软件生成更新事件，则在下溢时产生更新事件。如果在写入奇数到 CREP 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

图 14-9. 中央计数模式下计数器重复时序图

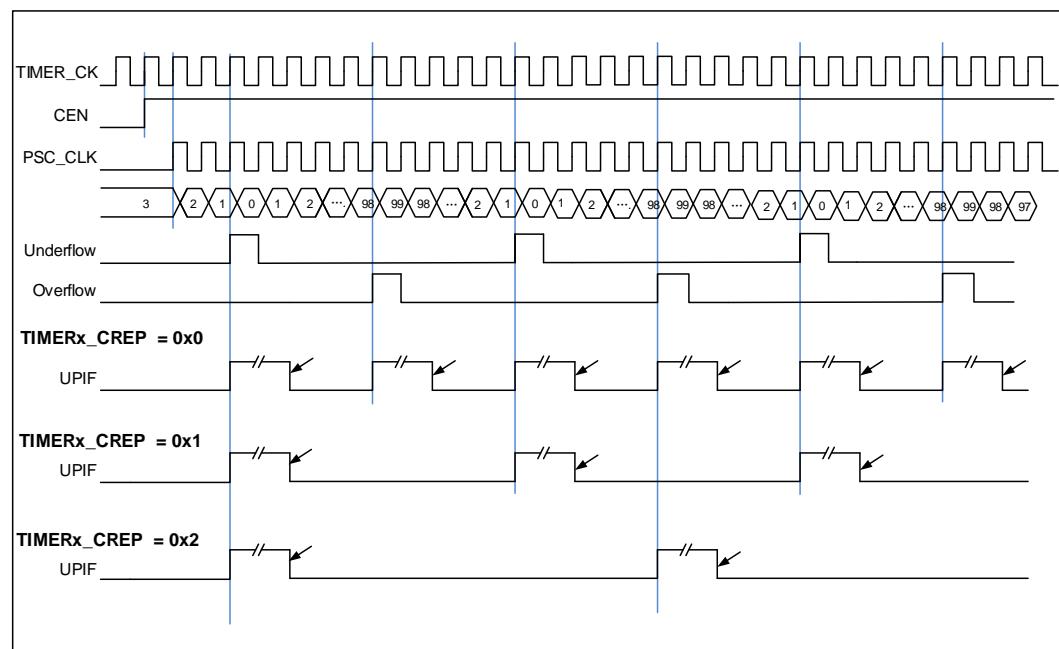


图 14-10. 在向上计数模式下计数器重复时序图

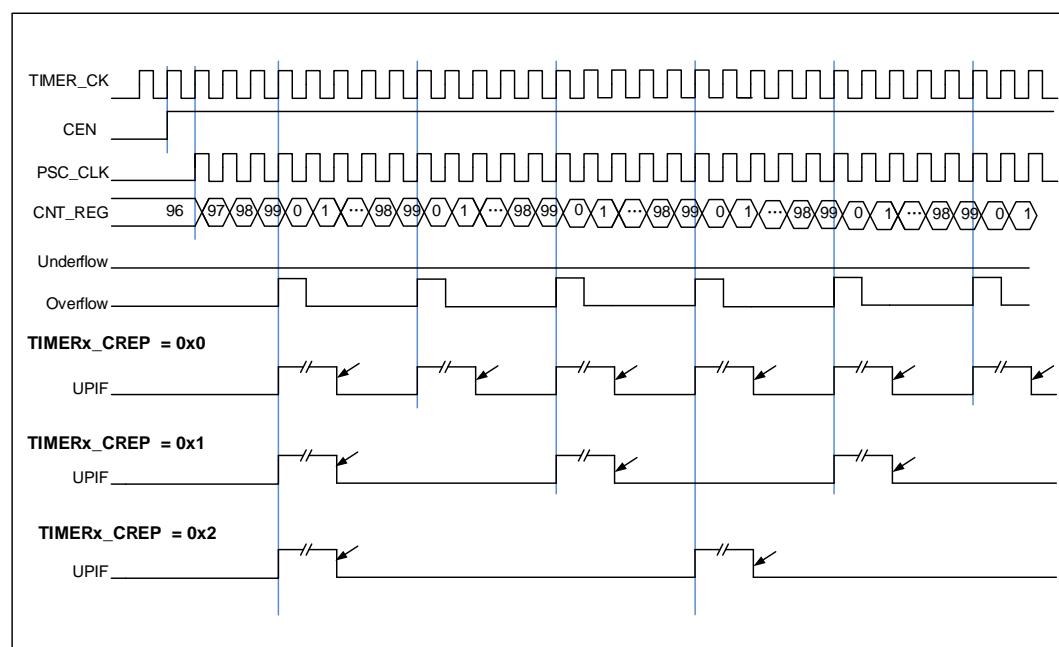
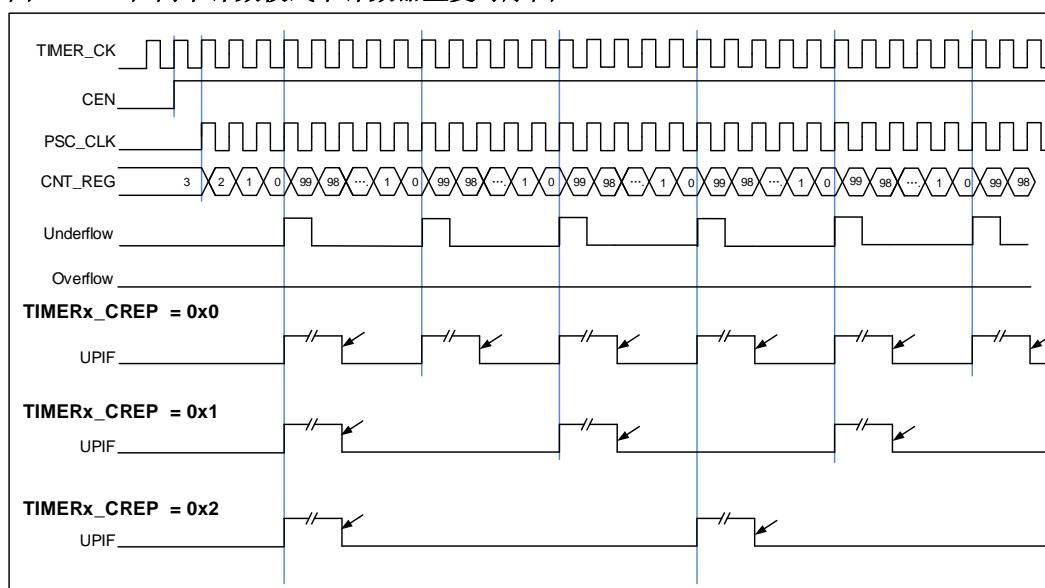


图 14-11. 在向下计数模式下计数器重复时序图



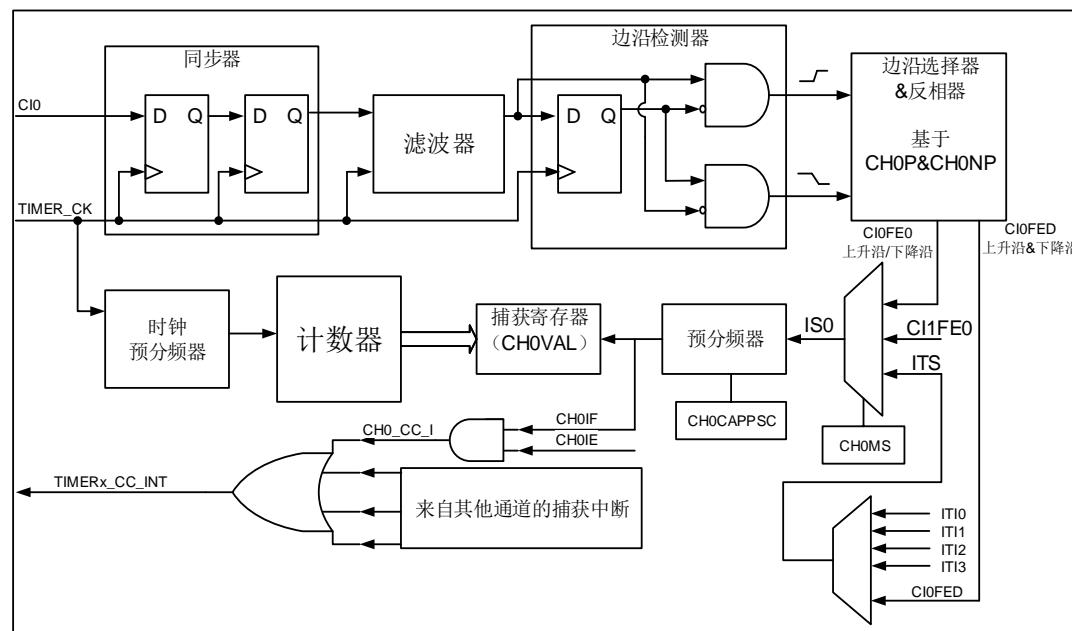
### 输入捕获和输出比较通道

高级定时器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。

#### ■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形的时序，频率，周期和占空比等。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。如果在输入引脚上出现被选择的边沿，**TIMERx\_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，若**CHxIE=1**则产生通道中断。

图 14-12. 通道输入捕获原理



通道输入信号  $\text{Cl}_x$  有两种选择，一种是  $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}$  信号，另一种是  $\text{TIMER}_{\text{x}}\text{-CH}_0$ ， $\text{TIMER}_{\text{x}}\text{-CH}_1$  和  $\text{TIMER}_{\text{x}}\text{-CH}_2$  异或之后的信号。通道输入信号  $\text{Cl}_x$  先被  $\text{TIMER\_CK}$  信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置  $\text{CH}_{\text{x}}\text{P}$  选择使用上升沿或者下降沿。通过配置  $\text{CH}_{\text{x}}\text{MS}$ ，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。配置  $\text{IC}$  预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}\text{CV}$  存储计数器的值。

配置步骤如下：

**第一步：滤波器配置（ $\text{TIMER}_{\text{x}}\text{-CHCTL0}$  寄存器中  $\text{CH}_{\text{x}}\text{CAPFLT}$ ）：**

根据输入信号和请求信号的质量，配置相应的  $\text{CH}_{\text{x}}\text{CAPFLT}$ 。

**第二步：边沿选择（ $\text{TIMER}_{\text{x}}\text{-CHCTL2}$  寄存器中  $\text{CH}_{\text{x}}\text{P}/\text{CH}_{\text{x}}\text{NP}$ ）：**

配置  $\text{CH}_{\text{x}}\text{P}/\text{CH}_{\text{x}}\text{NP}$  选择上升沿或者下降沿。

**第三步：捕获源选择（ $\text{TIMER}_{\text{x}}\text{-CHCTL0}$  寄存器中  $\text{CH}_{\text{x}}\text{MS}$ ）：**

一旦通过配置  $\text{CH}_{\text{x}}\text{MS}$  选择输入捕获源，必须确保通道配置在输入模式 ( $\text{CH}_{\text{x}}\text{MS} \neq 0x0$ )，而且  $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}\text{CV}$  寄存器不能再被写。

**第四步：中断使能（ $\text{TIMER}_{\text{x}}\text{-DMAINTEN}$  寄存器中  $\text{CH}_{\text{x}}\text{IE}$  和  $\text{CH}_{\text{x}}\text{DEN}$ ）：**

使能相应中断，可以获得中断和 DMA 请求。

**第五步：捕获使能（ $\text{TIMER}_{\text{x}}\text{-CHCTL2}$  寄存器中  $\text{CH}_{\text{x}}\text{EN}$ ）。**

**结果：**当期望的输入信号发生时， $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}\text{CV}$  被设置成当前计数器的值， $\text{CH}_{\text{x}}\text{IF}$  位置 1。

如果  $\text{CH}_{\text{x}}\text{IF}$  位已经为 1，则  $\text{CH}_{\text{x}}\text{OF}$  位置 1。根据  $\text{TIMER}_{\text{x}}\text{-DMAINTEN}$  寄存器中  $\text{CH}_{\text{x}}\text{IE}$  和  $\text{CH}_{\text{x}}\text{DEN}$  的配置，判断相应的中断和 DMA 请求是否被提出。

**直接产生：**软件设置  $\text{CH}_{\text{x}}\text{G}$  位，会直接产生中断和 DMA 请求。

通道输入捕获功能也可用来测量  $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}$  引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到  $\text{CIO}$ 。配置  $\text{TIMER}_{\text{x}}\text{-CHCTL0}$  寄存器中  $\text{CH}_0\text{MS}$  为  $2'b01$ ，选择通道 0 的捕获信号为  $\text{CIO}$ ，同时设置上升沿捕获。配置  $\text{TIMER}_{\text{x}}\text{-CHCTL0}$  寄存器中  $\text{CH}_1\text{MS}$  为  $2'b10$ ，选择通道 1 捕获信号为  $\text{CIO}$ ，同时设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。 $\text{TIMER}_{\text{x}}\text{-CH}_0\text{CV}$  寄存器测量 PWM 的周期值， $\text{TIMER}_{\text{x}}\text{-CH}_1\text{CV}$  寄存器测量 PWM 占空比值。

#### ■ 通道输出比较功能

图 14-13. 通道输出比较原理（带有互补输出的通道， $x=0,1,2$ ）

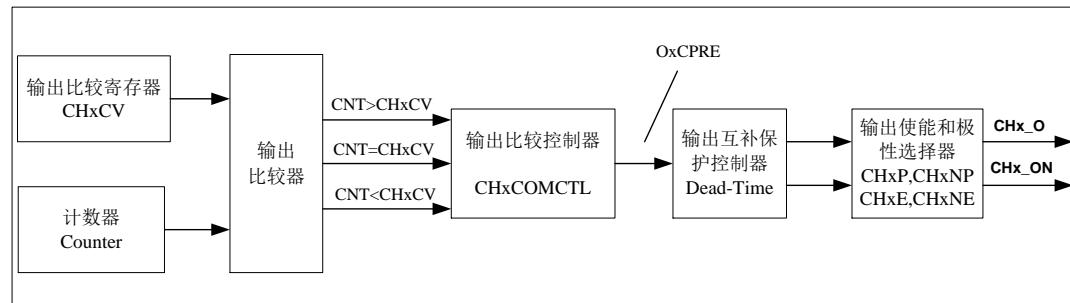
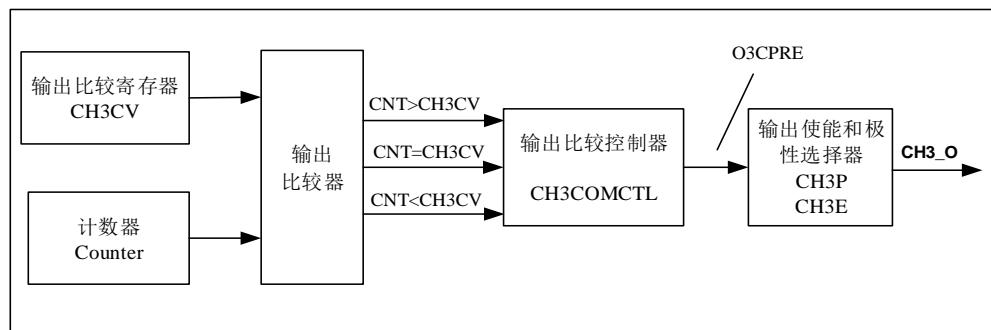


图 14-14. 通道输出比较原理



[图14-13. 通道输出比较原理（带有互补输出的通道， \$x=0,1,2\$ ）](#) 和 [图14-14. 通道输出比较原理](#) 分别给出了输出比较的原理电路。通道输出信号  $CHx\_O/CHx\_ON$  与  $OxCPRE$  信号（详情请见 [通道输出准备信号](#)）的关系描述如下： $OxCPRE$  信号高电平有效， $CHx\_O/CHx\_ON$  的输出情况与  $OxCPRE$  信号， $CHxP/CHxNP$  位和  $CHxE/CHxNE$  位有关（具体情况请见 [TIMERx\\_CHCTL2](#) 寄存器中的描述）。例如：

1) 当设置  $CHxP=0$  ( $CHx\_O$  高电平有效，与  $OxCPRE$  输出极性相同)、 $CHxE=1$  ( $CHx\_O$  输出使能) 时：

- 若  $OxCPRE$  输出有效 (高) 电平，则  $CHx\_O$  输出有效 (高) 电平；
- 若  $OxCPRE$  输出无效 (低) 电平，则  $CHx\_O$  输出无效 (低) 电平。

2) 当设置  $CHxNP=1$  ( $CHx\_ON$  低电平有效，与  $OxCPRE$  输出极性相反)、 $CHxNE=1$  ( $CHx\_ON$  输出使能) 时：

- 若  $OxCPRE$  输出有效 (高) 电平，则  $CHx\_ON$  输出有效 (低) 电平；
- 若  $OxCPRE$  输出无效 (低) 电平，则  $CHx\_ON$  输出无效 (高) 电平。

当  $CH0\_O$  和  $CH0\_ON$  同时输出时， $CH0\_O$  和  $CH0\_ON$  的具体输出情况还与 [TIMERx\\_CCHP](#) 寄存器中的相关位 (ROS、IOS、POE 和 DTCFG 等位) 有关。详情请见 [通道输出互补PWM](#)。

在通道输出比较功能，[TIMERx](#) 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 [TIMERx\\_CHxCV](#) 寄存器与计数器的值匹配时，根据 [CHxCOMCTL](#) 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 [TIMERx\\_CHxCV](#) 寄存器的值匹配时，[CHxIF](#) 位被置 1，如果  $CHxIE = 1$  则会产生中断，如果  $CxDCE=1$  则会产生 DMA 请求。

配置步骤如下：

**第一步：时钟配置：**

配置定时器时钟源，预分频器等。

**第二步：比较模式配置：**

- 设置  $CHxCOMSEN$  位来配置输出比较影子寄存器；
- 设置  $CHxCOMCTL$  位来配置输出模式 (置高电平/置低电平/反转)；
- 设置  $CHxP/CHxNP$  位来选择有效电平的极性；
- 设置  $CHxEN$  使能输出。

**第三步：通过  $CHxIE/CxDCE$  位配置中断/DMA 请求使能。**

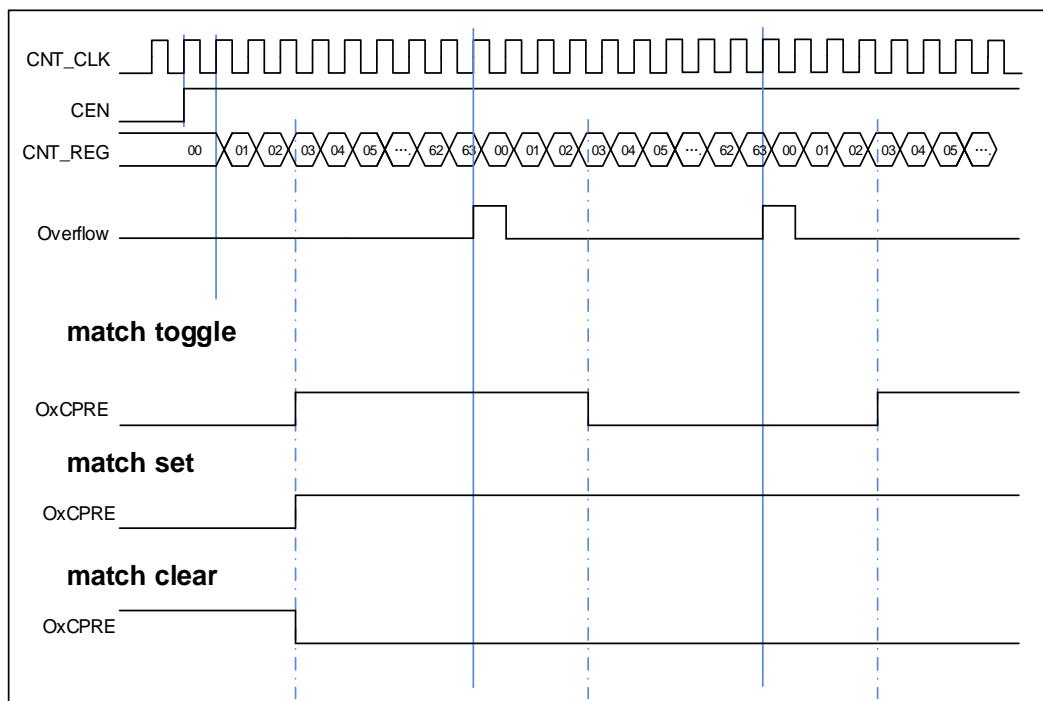
**第四步：通过 [TIMERx\\_CAR](#) 寄存器和 [TIMERx\\_CHxCV](#) 寄存器配置输出比较时基：**

TIMERx\_CHxCV可以在运行时根据你所期望的波形而改变。

**第五步：**设置CEN位使能定时器。

[图14-15. 三种输出比较模式](#)显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 14-15. 三种输出比较模式



## 输出 PWM 功能

在PWM输出模式下（PWM模式0是配置CHxCOMCTL为3'b110，PWM模式1是配置CHxCOMCTL为3'b111），通道根据TIMERx\_CAR寄存器和TIMERx\_CHxCV寄存器的值，输出PWM波形。

根据计数模式，可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

EAPWM的周期由TIMERx\_CAR寄存器值决定，占空比由TIMERx\_CHxCV寄存器值决定。[图14-16. EAPWM时序图](#)显示了CAPWM的输出波形和中断。

CAPWM的周期由(2\*TIMERx\_CAR寄存器值)决定，占空比由(2\*TIMERx\_CHxCV寄存器值)决定。[图14-17. CAPWM时序图](#)显示了CAPWM的输出波形和中断。

在向上计数模式中，PWM模式0下（CHxCOMCTL=3'b110），如果TIMERx\_CHxCV寄存器的值大于TIMERx\_CAR寄存器的值，通道输出一直为无效电平；PWM模式1下（CHxCOMCTL=3'b111），如果TIMERx\_CHxCV寄存器的值大于TIMERx\_CAR寄存器的值，通道输出一直为有效电平。

图 14-16. EAPWM 时序图

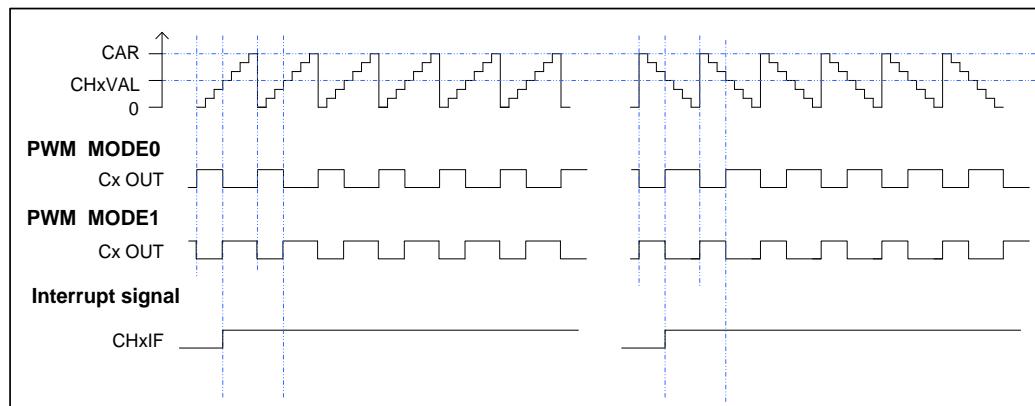
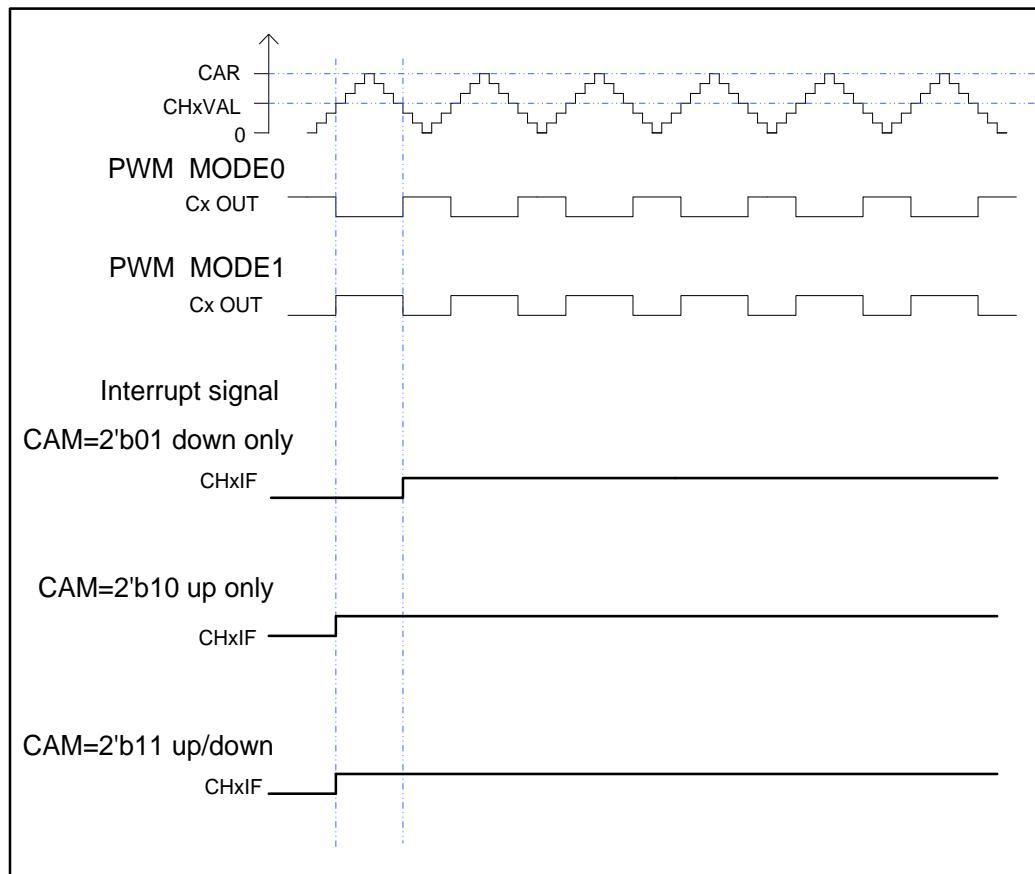


图 14-17. CAPWM 时序图



### 通道输出准备信号

根据[图14-13. 通道输出比较原理（带有互补输出的通道， \$x=0,1,2\$ ）](#)所示，当TIMERx用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号OxCPre信号(通道x输出准备信号)。设置CHxCOMCTL位可以定义OxCPre信号类型。OxCPre信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPre信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPre信号设置为低电平；设置

**CHxCOMCTL=0x03**, 在计数器值和**TIMERx\_CHxCV**寄存器的值匹配时, 可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型, 设置**CHxCOMCTL**位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中, 根据计数器值和**TIMERx\_CHxCV**寄存器值的关系以及计数方向, OxCPR<sub>E</sub>信号改变其电平。具体细节描述, 请参考相应的位。

设置**CHxCOMCTL=0x04**或**0x05**可以实现OxCPR<sub>E</sub>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**TIMERx\_CHxCV**的值和计数器值之间的比较结果。

设置**CHxCOMCEN=1**, 当由外部ETI引脚信号产生的ETIFP信号为高电平时, OxCPR<sub>E</sub>被强制为低电平。在下一次更新事件到来时, OxCPR<sub>E</sub>信号才会回到有效电平状态。

### 通道输出互补 PWM

**CHx\_O**和**CHx\_ON**是一对互补输出通道, 这两个信号不能同时有效。**TIMERx**有四路通道, 只有前三路有互补输出通道。互补信号**CHx\_O**和**CHx\_ON**是由一组参数来决定:  
**TIMERx\_CHCTL2**寄存器中的**CHxEN**和**CHxNEN**位, **TIMERx\_CCHP**寄存器中的**POEN**、**ROS**和**IOS**位, **TIMERx\_CTL1**寄存器中的**ISOx**和**ISOxN**位。输出极性由**TIMERx\_CHCTL2**寄存器中的**CHxP**和**CHxNP**位来决定。

表 14-2. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁用	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出禁用。	
			1	0	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1		
		1	0	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用。	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出使能。	
			1	0	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1		
1	0	0/1	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON 输出禁用。	
				1	CHx_O = LOW	CHx_ON=OxCPR <sub>E</sub> ⊕CHxNP CHx_ON 输出使能
			1	0	CHx_O=OxCPR <sub>E</sub> ⊕CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁用。

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
				1	CHx_O=OxCPRE $\oplus$ CHxP CHx_O输出使能	CHx_ON=(!OxCPRE) $\oplus$ CHxNP CHx_ON输出使能
					CHx_O = CHxP CHx_O输出禁用。	CHx_ON = CHxNP CHx_ON输出禁用。
				0	CHx_O = CHxP CHx_O输出使能	CHx_ON=OxCPRE $\oplus$ CHxNP CHx_ON输出使能
					CHx_O=OxCPRE $\oplus$ CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能
				1	CHx_O=OxCPRE $\oplus$ CHxP CHx_O输出使能	CHx_ON=(!OxCPRE) $\oplus$ CHxNP CHx_ON输出使能
					CHx_O=OxCPRE $\oplus$ CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能

### 互补 PWM 插入死区时间

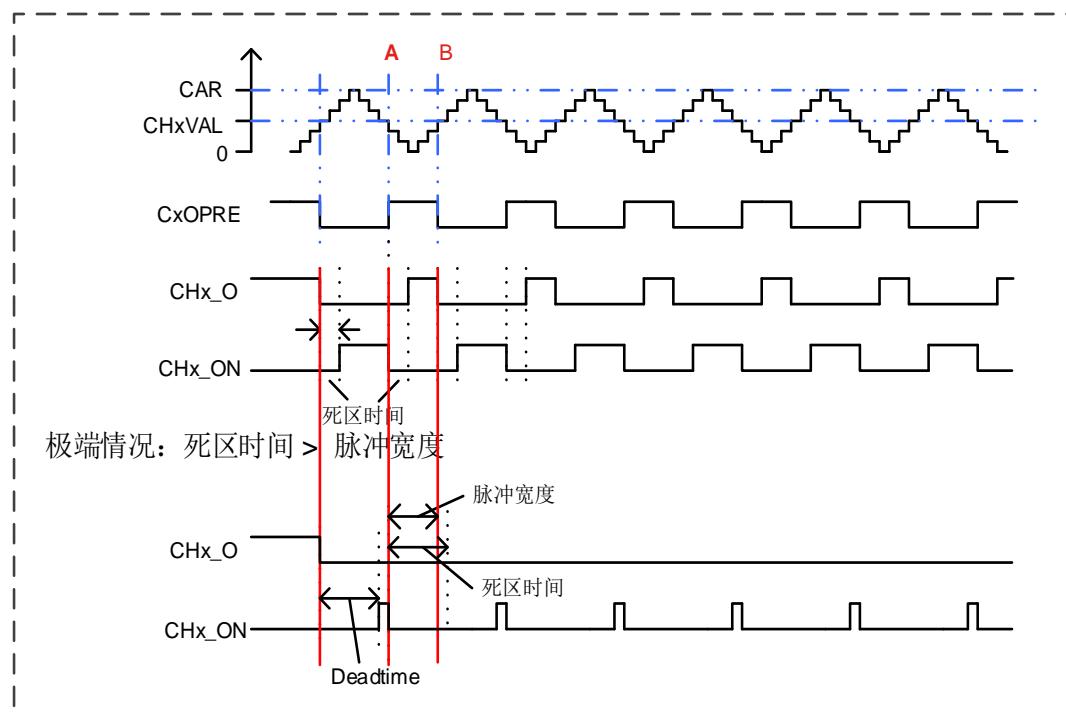
设置CHxEN和CHxNEN为1'b1的同时，设置POEN为1，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对除了通道3以外的通道有效。死区时间的细节，请参考TIMERx\_CCHP寄存器。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在PWM模式0，当通道x匹配事件发生时（TIMERx计数器=CHxVAL），OxCPRE反转。在图14-18. 带死区时间的互补输出中的A点，CHx\_O信号在死区时间内为低电平，直到死区时间过后才变为高电平，而CHx\_ON信号立刻变为低电平。同样，在B点，通道x匹配事件再次发生（TIMERx计数器=CHxVAL），OxCPRE信号被清0，CHx\_O信号被立即清零，CHx\_ON信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些极端事件发生，例如：如果死区延时大于或者等于CHx\_ON信号的占空比，CHx\_ON信号一直为无效值。

图 14-18. 带死区时间的互补输出



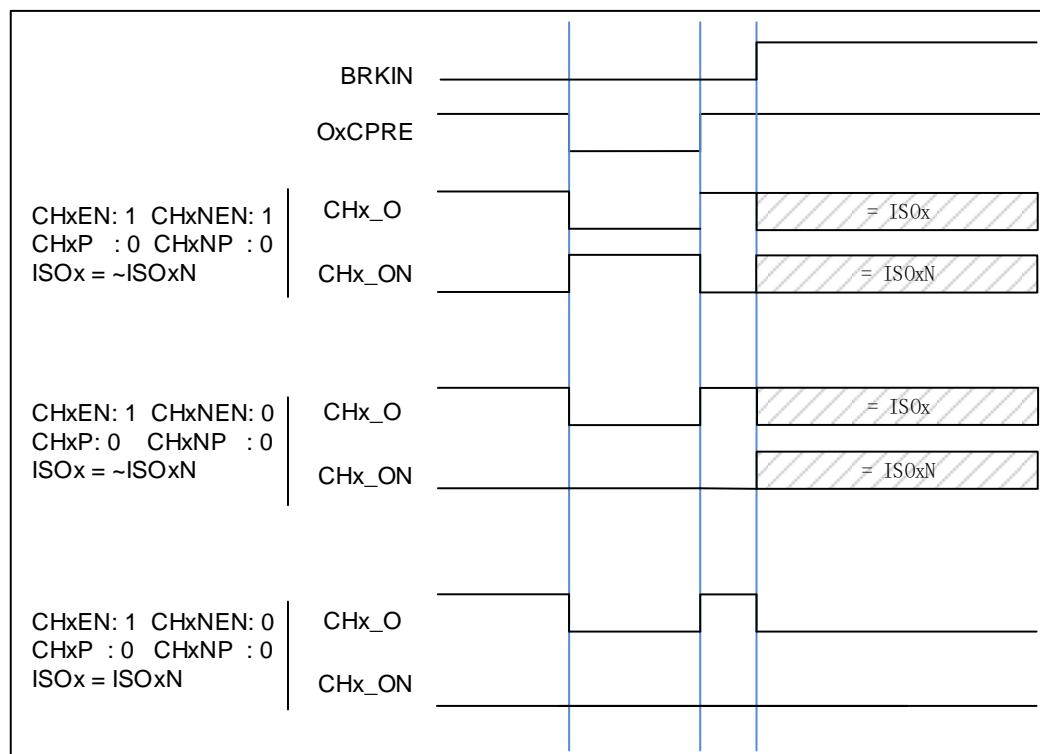
### 中止模式

使用中止模式时，输出CHx\_O和CHx\_ON的信号电平被以下位控制，**TIMERx\_CCHP**寄存器的POEN，IOS和ROS位，**TIMERx\_CTL1**寄存器的ISOx和ISOxN位。任何情况下，CHx\_O和CHx\_ON信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择HXTAL时钟失效事件，时钟失效事件由RCU中的时钟监视器(CKM)产生。将**TIMERx\_CCHP**寄存器的BRKEN位置1可以使能中止功能。**TIMERx\_CCHP**寄存器的BRKP位决定了中止输入极性。

发生中止时，POEN位被异步清除，一旦POEN位为0，CHx\_O和CHx\_ON的输出电平由**TIMERx\_CTL1**寄存器中的ISOx位和ISOxN位决定。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\_INTF**寄存器的BRKIF位被置1。如果BRKIE=1，中断产生。

图 14-19. 通道响应中止输入（高电平有效）时，输出信号的行为



### 正交译码器

正交译码器功能使用由TIMERx\_CH0和TIMERx\_CH1引脚生成的CI0FE0和CI1FE1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位会发生改变。输入源可以是只有CI0FE0，可以只有CI1FE1，或者可以同时有CI0FE0和CI1FE1，通过设置SMC=0x01, 0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制如[表14-3. 不同编码器模式下的计数方向](#)所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在0和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置TIMERx\_CAR寄存器。

表 14-3. 不同编码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
编码器模式0 SMC[2:0]=3'b001	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
编码器模式1 SMC [2:0]=3'b010	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
编码器模式2 SMC [2:0]=3'b011	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意：“-”意思是“无计数”；“X”意思是不可能。“0”意思是低电平，“1”意思是高电平。

图 14-20.在编码器模式 2 且 CI0FE0 极性不反相时计数器行为

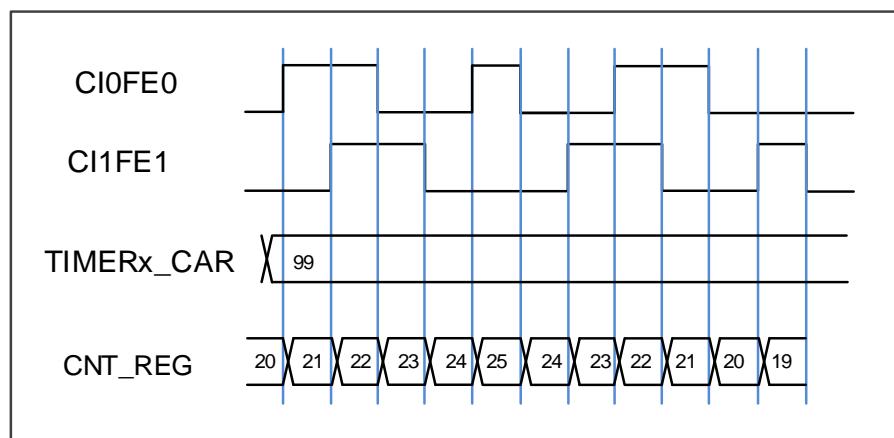
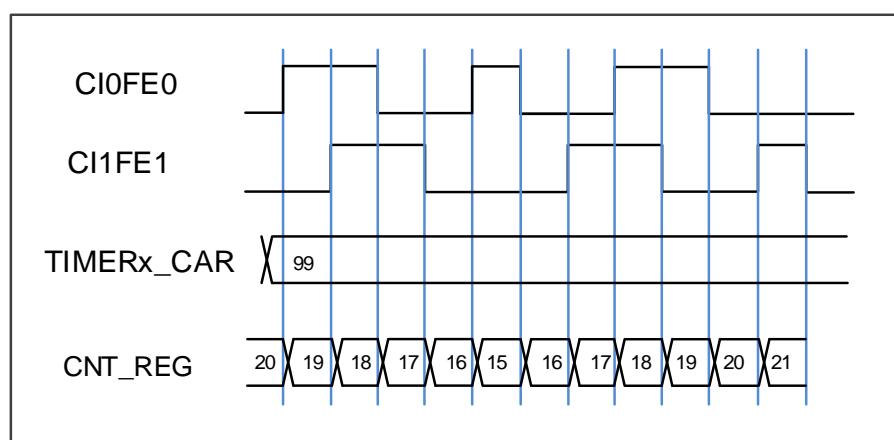


图 14-21.在编码器模式 2 且 CI0FE0 极性反相时计数器行为



### 霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制BLDC电机。

**图14-22. 霍尔传感器用在BLDC电机控制中**是定时器和电机的连接示意图。众所周知，我们要两个定时器。**TIMER\_in**定时器（可以是高级定时器或者通用L0定时器）接收霍尔传感器的三路信号。

三个霍尔传感器信号与 **TIMER\_in** 定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接功能（TRGO-ITIx），**TIMER\_in**定时器和**TIMER\_out**定时器可以连接在一起。**TIMER\_out**定时器根据ITIx触发信号输出PWM波，驱动BLDC电机，控制BLDC电机的速度。这样，**TIMER\_in**定时器和**TIMER\_out**定时器的连接形成了一个反馈电路，可以根据需求改变配置。

高级定时器和通用L0定时器具有输入异或功能，可作为**TIMER\_in**定时器。同时，高级定时器具备互补输出和死区插入功能，可作为**TIMER\_out**定时器。

另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如：

**TIMER\_in (TIMER2) -> TIMER\_out (TIMER0 ITI2)**

选择好合适的互连定时器，线路也已经连接好，就可以配置定时器。有以下关键配置：

- 通过设置TI0S，来使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 通过设置CCUC和CCSE，来选择ITIx触发换相。
- 根据需求配置PWM参数。

图 14-22. 霍尔传感器用在 BLDC 电机控制中

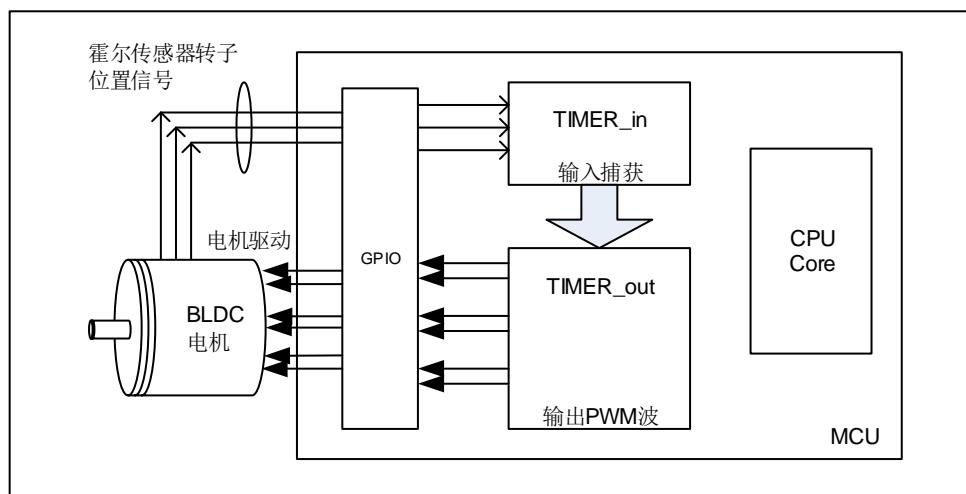
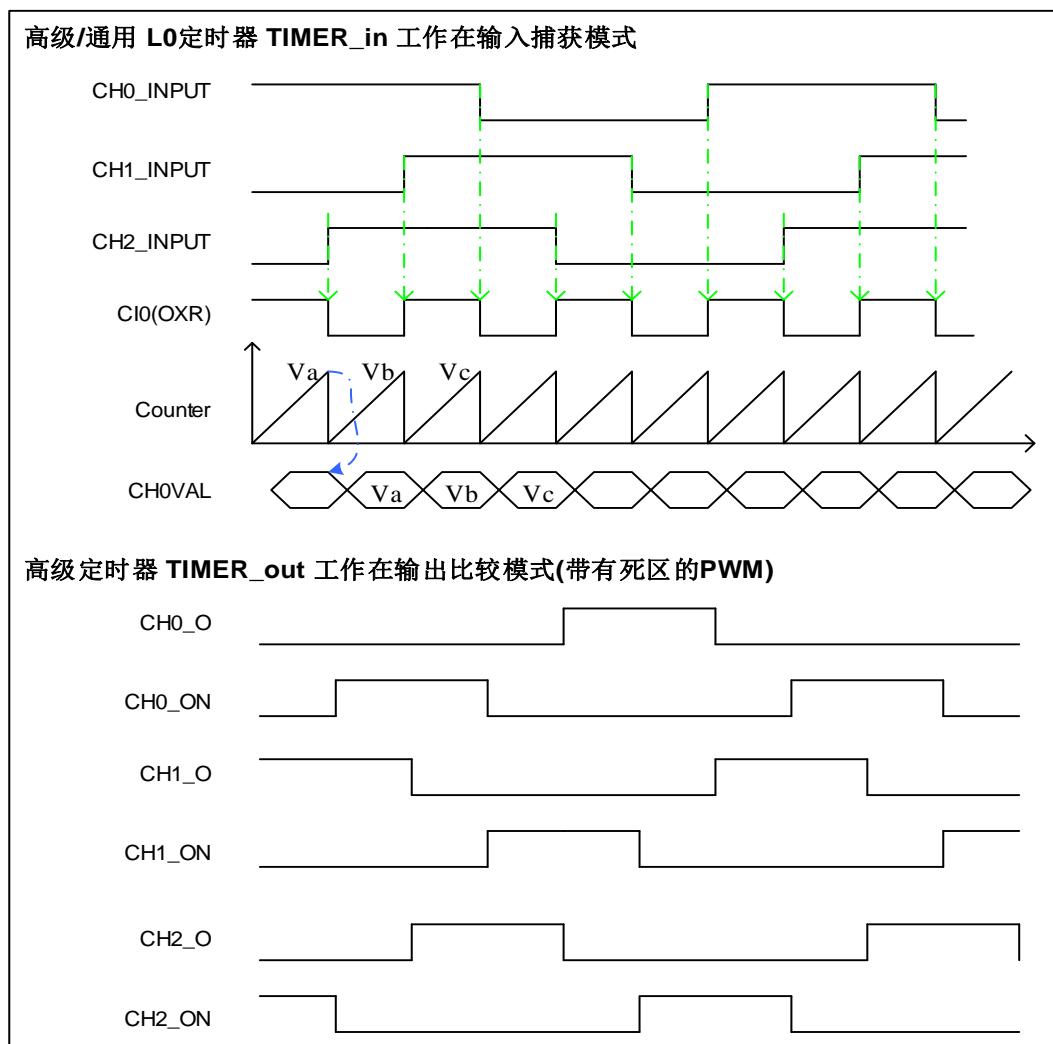


图 14-23. 两个定时器之间的霍尔传感器时序图

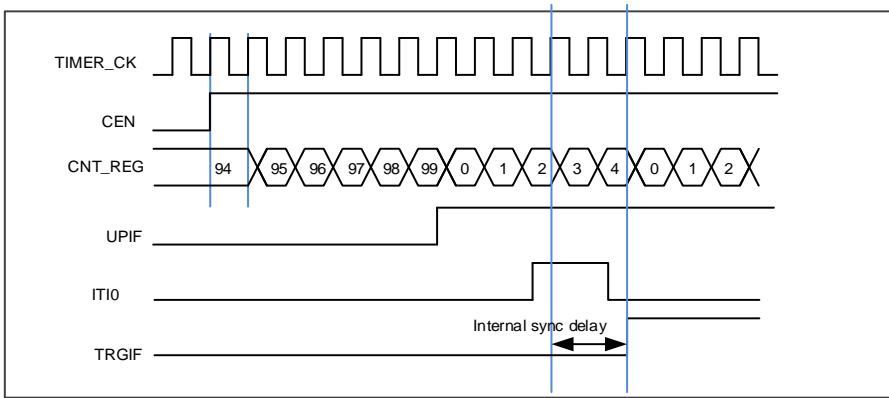
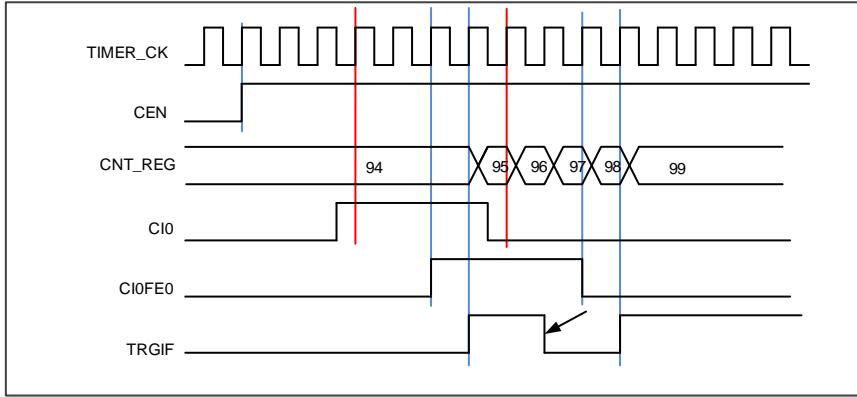


### 主-从管理

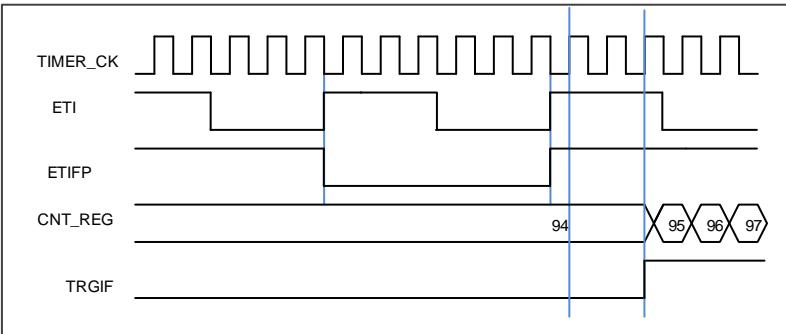
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx\_SMCFG寄存器中的SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx\_SMCFG寄存器中的TRGS[2:0]来选择。

表 14-4. 从模式示例

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相。 如果触发源是ETIFFP，配置ETP选择极性和反相。	若触发源为ITIx，滤波和预分频不可用。 若触发源为CIx，可配置CHxCAPFLT设置滤波，预分频不可用。 若触发源为ETIFFP，滤波和预分频均可用。

	模式选择	触发源选择	极性选择	滤波和预分频
		111: ETIFP		
	<b>复位模式</b> 当触发输入上升沿到来时，计数器清零重启。	TRGS[2:0]=3'b000 选择ITIO为触发源。	若触发源是ITIO，极性选择不可用。	若触发源是ITIO，滤波和预分频不可用。
<b>图 14-24. 复位模式</b>				
例1				 <p>The diagram illustrates the timing sequence for the reset mode. The TIMER_CK signal is a square wave. The CEN signal is high during the first four timer cycles. The CNT_REG signal shows the count from 94 to 1, then back to 94. The UPIF signal is high during the first four cycles. The ITIO signal is high during the first four cycles. The TRGIF signal is high after a short internal sync delay following the fourth cycle.</p>
	<b>暂停模式</b> 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TRGS[2:0]=3'b101 选择CI0FE0为触发源。	TIOS=0 (非异或) [CH0NP=0, CH0P=0] CI0FE0不反相。捕获发生在上升沿。	在这个例子中滤波被旁路。
例2				 <p>The diagram illustrates the timing sequence for stop mode. The TIMER_CK signal is a square wave. The CEN signal is high during the first four timer cycles. The CNT_REG signal shows the count from 94 to 99. The CI0 signal is high during the first four cycles. The CI0FE0 signal is high during the first four cycles. The TRGIF signal is high after a short internal sync delay following the fourth cycle.</p>
例3	<b>事件模式</b> 触发输入的上升沿计数器开始计数。	TRGS[2:0]=3'b111 选择ETIFP为触发源。	ETP = 0, ETI极性不变。	ETPSC = 1, ETI 2分频。 ETFC = 0, ETI 无滤波。

	模式选择	触发源选择	极性选择	滤波和预分频
图 14-26. 事件模式				



### 单脉冲模式

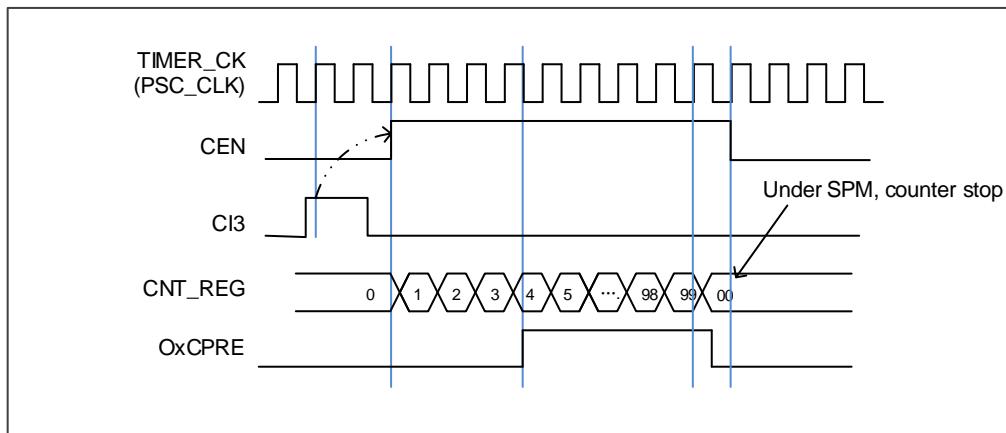
设置TIMERx\_CTL0寄存器的SPM位置1，使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置CHxCOMCTL配置TIMERx为PWM模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx\_CTL0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。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx\_CHxCV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\_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM模式0或PWM模式1时CHxCOMFEN位才可用，触发源来源于触发信号。

[图14-27. 单脉冲模式，TIMERx\\_CHxCV = 4, TIMERx\\_CAR=99](#)展示了一个例子。

图 14-27. 单脉冲模式，TIMERx\_CHxCV = 4, TIMERx\_CAR=99

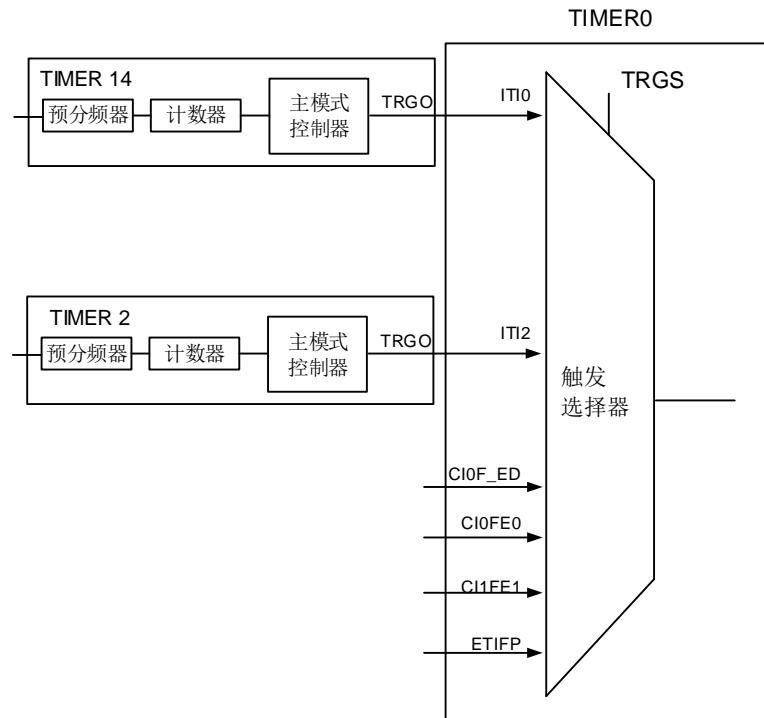


## 定时器互连

定时器之间可配置为内部级联，一个定时器配置为主模式输出TRGO信号，另一个定时器配置为从模式，TRGO信号包括复位事件、使能事件、更新事件、捕获比较脉冲事件、比较事件。从定时器接收到ITIx信号，并执行对应的操作，包括内部时钟模式、正交编码模式、复位模式、暂停模式、事件模式、外部时钟模式。

[图14-28. 定时器0主/从模式的例子](#) 显示了当定时器0配置为从模式时的触发选择。

图 14-28. 定时器 0 主/从模式的例子



其他定时器互连的例子：

- 定时器2作为定时器0的预分频器

参考[图14-28. 定时器0主/从模式的例子](#)连接配置定时器2为定时器0的预分频器，步骤如下：

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\_CTL1寄存器的MMC=3'b010)。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期(TIMER2\_CAR寄存器)；
3. 选择定时器0输入触发源为定时器2 (配置TIMERx\_SMCFG寄存器的TRGS=3'b010)；
4. 配置定时器0在外部时钟模式0(配置TIMERx\_SMCFG寄存器的SMC=3'b111)；
5. 写1到CEN位启动定时器0 (TIMER0\_CTL0寄存器)；
6. 写1到CEN位启动定时器2 (TIMER2\_CTL0寄存器)。

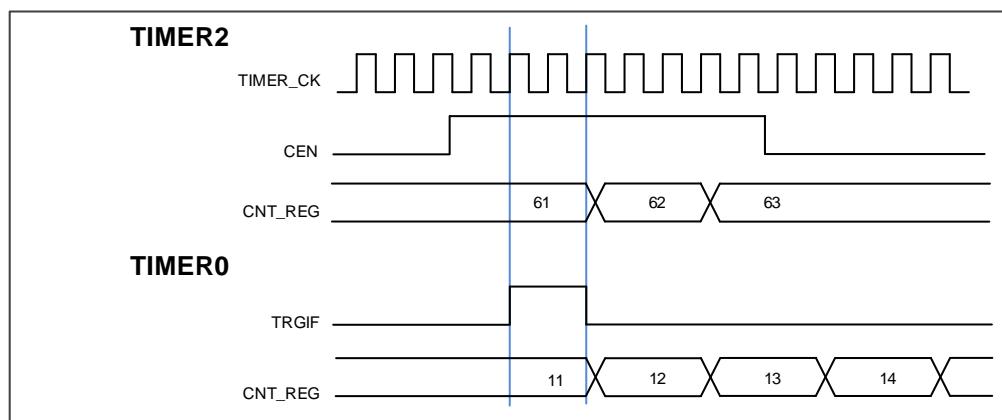
- 用定时器2的使能/更新信号来启动定时器0

用定时器2的使能信号来启动定时器0，见[图14-29. 用定时器2的使能信号触发定时器0](#)。在定时器2使能信号输出后，定时器0按照分频后的内部时钟从当前值开始计数。

当定时器0接收到触发信号，它的CEN位被置1，计数器计数直到禁能定时器0。两个定时器的计数器频率都是TIMER\_CK经过预分频器3分频后的频率( $f_{PSC\_CLK} = f_{TIMER\_CK}/3$ )。步骤如下：

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\_CTL1寄存器的MMC=3'b001);
2. 配置定时器0选择输入触发来自定时器2 (配置TIMERx\_SMCFG寄存器的TRGS=3'b010);
3. 配置定时器0在事件模式 (配置TIMERx\_SMCFG寄存器的SMC=3'b 110);
4. 写1到CEN来开启定时器2 (TIMER2\_CTL0寄存器)。

**图 14-29. 用定时器 2 的使能信号触发定时器 0**



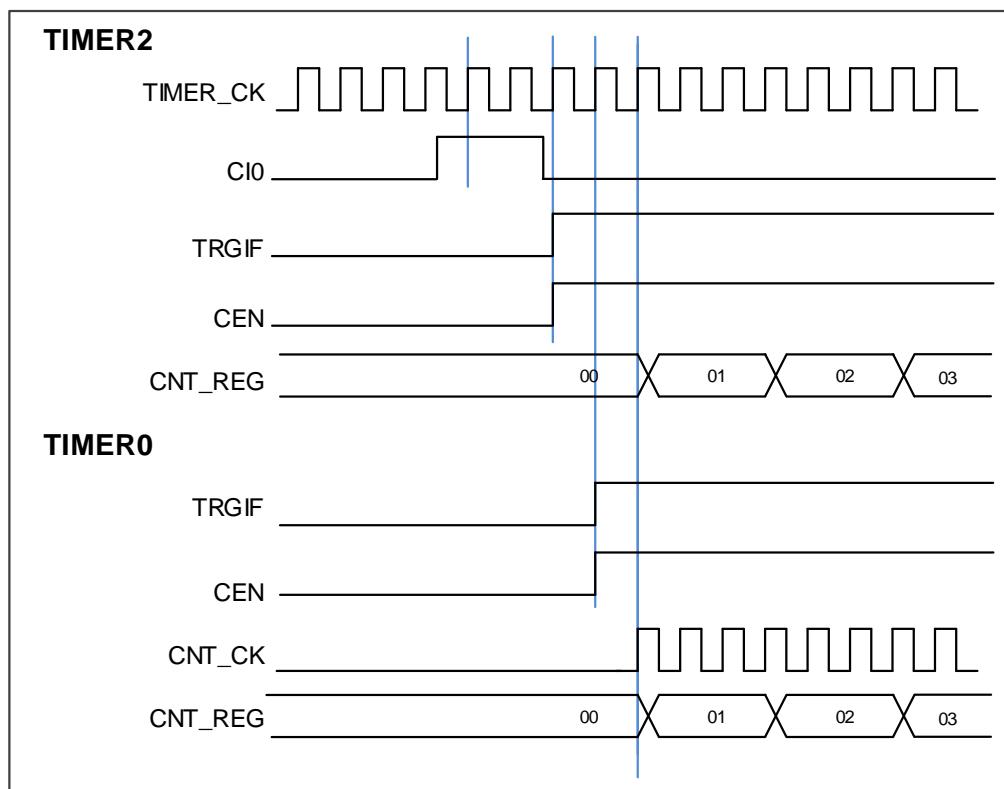
■ 使用一个外部触发来同步两个定时器。

配置定时器2的使能信号触发定时器0的开启，配置定时器2的CI0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。步骤如下：

1. 配置定时器2工作在从模式，并选择CI0\_ED作为触发输入(配置TIMER2\_SMCFG寄存器的TRGS=3'b100);
2. 配置定时器2工作在事件模式(配置TIMER2\_SMCFG寄存器的SMC=3'b110);
3. 写MSM=1(TIMER2\_SMCFG寄存器)来配置定时器2工作在主/从模式;
4. 配置定时器0的触发输入为定时器2 (配置TIMERx\_SMCFG 寄存器的TRGS=3'b010);
5. 配置定时器0工作在事件模式(配置TIMER0\_SMCFG寄存器的SMC=3'b110)。

当定时器2的CI0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 14-30. 用定时器 2 的 CI0 信号来触发定时器 0 和定时器 2



### 定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMERx\_DMACFG和TIMERx\_DMATB。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P（传输方向为从内存到外设）模式，PADDR（外设基地址）为TIMERx\_DMATB寄存器地址，DMA就会访问TIMERx\_DMATB寄存器。实际上，TIMERx\_DMATB寄存器只是一个缓冲，定时器会将TIMERx\_DMATB映射到一个内部寄存器，这个内部寄存器由TIMERx\_DMACFG寄存器中的DMATA来指定。如果TIMERx\_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器发送1个DMA请求就可以完成。如果TIMERx\_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMERx\_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4，DMATA+0x8，DMATA+0xC寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

### 定时器调试模式

当Cortex®-M23内核停止，DBG\_CTL0寄存器中的TIMERx\_HOLD配置位被置1，定时器计数器停止。

### 14.1.5. TIMERx 寄存器(x=0)

TIMER0基地址: 0x4001 2C00

#### 控制寄存器 0 (TIMERx\_CTL0)

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				CKDIV[1:0]		ARSE		CAM[1:0]		DIR		SPM		UPS		UPDIS		CEN	
				rw		rw		rw		rw		rw		rw		rw			

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。</p> <p>00: <math>f_{DTS}=f_{CK\_TIMER}</math></p> <p>01: <math>f_{DTS}=f_{CK\_TIMER}/2</math></p> <p>10: <math>f_{DTS}=f_{CK\_TIMER}/4</math></p> <p>11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能TIMERx_CAR寄存器的影子寄存器</p> <p>1: 使能TIMERx_CAR寄存器的影子寄存器</p>
6:5	CAM[1:0]	<p>计数器对齐模式选择</p> <p>00: 无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向</p> <p>01: 中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向下计数时，CHxF位置1</p> <p>10: 中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向上计数时，CHxF位置1</p> <p>11: 中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)，在向上和向下计数时，CHxF位都会置1</p> <p>当计数器使能以后，该位不能从0x00切换到非0x00</p>
4	DIR	<p>方向</p> <p>0: 向上计数</p> <p>1: 向下计数</p>

当计数器配置为中央对齐计数模式或编码器模式时，该位只读。

3	SPM	单脉冲模式
		0: 单脉冲模式禁能。更新事件发生后，计数器继续计数
		1: 单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源
		软件配置该位，选择更新事件源。
		0: 以下事件均会产生更新中断或DMA请求：
		UPG位被置1
		计数器溢出/下溢
		复位模式产生的更新
		1: 下列事件会产生更新中断或DMA请求：
		计数器溢出/下溢
1	UPDIS	禁止更新。
		该位用来使能或禁能更新事件的产生
		0: 更新事件使能。更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：
		UPG位被置1
		计数器溢出/下溢
		复位模式产生的更新
		1: 更新事件禁能。
		注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能
		0: 计数器禁能
		1: 计数器使能
		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。

### 控制寄存器 1 (TIMERx\_CTL1)

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	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]	DMAS	CCUC	保留	CCSE		
	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	ISO3	通道3的空闲状态输出 参考ISO0位
13	ISO2N	通道2的互补通道空闲状态输出 参考ISO0N位
12	ISO2	通道2的空闲状态输出 参考ISO0位
11	ISO1N	通道1的互补通道空闲状态输出 参考ISO0N位
10	ISO1	通道1的空闲状态输出 参考ISO0位
9	ISO0N	通道0的互补通道空闲状态输出 0: 当POEN位复位时, CH0_ON设置低电平 1: 当POEN位复位时, CH0_ON设置高电平 此位只有在TIMERx_CCHP寄存器的PROT[1:0]位为00的时候可以被更改。
8	ISO0	通道0的空闲状态输出 0: 当POEN位复位时, CH0_O设置低电平 1: 当POEN位复位时, CH0_O设置高电平 如果CH0_ON生效, 一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP寄存器的PROT[1:0]位为00的时候可以被更改。
7	TIOS	通道0触发输入选择 0: 选择TIMERx_CH0引脚作为通道0的触发输入 1: 选择TIMERx_CH0, TIMERx_CH1和TIMERx_CH2引脚异或的结果作为通道0的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择, TRGO信号由主定时器发给从定时器用于同步功能。 000: 当产生一个定时器复位事件后, 输出一个TRGO信号, 定时器复位源为: 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001: 当产生一个定时器使能事件后, 输出一个TRGO信号, 定时器使能源为: CEN位置1 在暂停模式下, 触发输入置1 010: 当产生一个定时器更新事件后, 输出一个TRGO信号, 更新事件源由UPDIS和UPS位决定 011: 当通道0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: 当产生一次比较事件时, 输出一个TRGO信号, 比较事件源来自O0CPRE 101: 当产生一次比较事件时, 输出一个TRGO信号, 比较事件源来自O1CPRE 110: 当产生一次比较事件时, 输出一个TRGO信号, 比较事件源来自O2CPRE 111: 当产生一次比较事件时, 输出一个TRGO信号, 比较事件源来自O3CPRE

3	DMAS	DMA请求源选择 0: 当通道捕获/比较事件发生时, 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	换相控制影子寄存器更新控制 当换相控制影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 使能(CCSE=1), 这些影子寄存器更新控制如下: 0: CMTG位被置1时, 更新影子寄存器 1: 当CMTG位被置1或检测到TRIGI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	换相控制影子使能 0: 影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 禁能 1: 影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 使能 如果这些位已经被写入了, 换相事件到来时这些位才被更新。 当通道没有互补输出时, 此位无效。

### 从模式配置寄存器 (**TIMERx\_SMCFG**)

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
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]		MSM		TRGS[2:0]		OCRC		SMC[2:0]			

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定ETI信号的极性 0: ETI高电平或上升沿有效 1: ETI低电平或下降沿有效
14	SMC1	SMC的一部分使能外部时钟模式1 在外部时钟模式1, 计数器由ETIFP信号上的任意有效边沿驱动 0: 外部时钟模式1禁能 1: 外部时钟模式1使能 当从模式配置为复位模式, 暂停模式和事件模式时, 定时器仍然可以工作在外部时钟模式1。但是 TRGS 必须不能为 3'b111。

如果外部时钟模式 0 和外部时钟模式 1 同时被配置，外部时钟的输入是 ETIF

注意：外部时钟模式0使能在寄存器的SMC[2:0]位域。

13:12	<b>ETPSC[1:0]</b>	<p>外部触发预分频</p> <p>外部触发信号ETIFF的频率不能超过TIMER_CK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ETIFF的频率。</p> <p>00: 预分频禁能</p> <p>01: 2 分频</p> <p>10: 4 分频</p> <p>11: 8分频</p>																																											
11:8	<b>ETFC[3:0]</b>	<p>外部触发滤波控制</p> <p>外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 <b>fSAMP</b> 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #d3d3d3; text-align: left; padding: 2px;"><b>EXTFC[3:0]</b></th><th style="background-color: #d3d3d3; text-align: center; padding: 2px;">次数</th><th style="background-color: #d3d3d3; text-align: center; padding: 2px;"><b>fSAMP</b></th></tr> </thead> <tbody> <tr> <td style="text-align: left; padding: 2px;">4'b0000</td><td style="text-align: center; padding: 2px;"></td><td style="text-align: center; padding: 2px;">滤波器禁能.</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0001</td><td style="text-align: center; padding: 2px;">2</td><td 4"="" style="text-align: center; padding: 2px; vertical-align: middle; rowspan="><math>f_{\text{TIMER\_CK}}</math></td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0010</td><td style="text-align: center; padding: 2px;">4</td><td rowspan="3" style="text-align: center; padding: 2px;"></td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0011</td><td style="text-align: center; padding: 2px;">8</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0100</td><td style="text-align: center; padding: 2px;">6</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0101</td><td style="text-align: center; padding: 2px;">8</td><td 4"="" style="text-align: center; padding: 2px; vertical-align: middle; rowspan="><math>f_{\text{DTS\_CK}}/2</math></td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0110</td><td style="text-align: center; padding: 2px;">6</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b0111</td><td style="text-align: center; padding: 2px;">8</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1000</td><td style="text-align: center; padding: 2px;">6</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1001</td><td style="text-align: center; padding: 2px;">8</td><td 4"="" style="text-align: center; padding: 2px; vertical-align: middle; rowspan="><math>f_{\text{DTS\_CK}}/4</math></td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1010</td><td style="text-align: center; padding: 2px;">5</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1011</td><td style="text-align: center; padding: 2px;">6</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1100</td><td style="text-align: center; padding: 2px;">8</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1101</td><td style="text-align: center; padding: 2px;">5</td><td 4"="" style="text-align: center; padding: 2px; vertical-align: middle; rowspan="><math>f_{\text{DTS\_CK}}/16</math></td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1110</td><td style="text-align: center; padding: 2px;">6</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1111</td><td style="text-align: center; padding: 2px;">8</td></tr> <tr> <td style="text-align: left; padding: 2px;">4'b1111</td><td style="text-align: center; padding: 2px;">8</td></tr> </tbody> </table>	<b>EXTFC[3:0]</b>	次数	<b>fSAMP</b>	4'b0000		滤波器禁能.	4'b0001	2	$f_{\text{TIMER\_CK}}$	4'b0010	4		4'b0011	8	4'b0100	6	4'b0101	8	$f_{\text{DTS\_CK}}/2$	4'b0110	6	4'b0111	8	4'b1000	6	4'b1001	8	$f_{\text{DTS\_CK}}/4$	4'b1010	5	4'b1011	6	4'b1100	8	4'b1101	5	$f_{\text{DTS\_CK}}/16$	4'b1110	6	4'b1111	8	4'b1111	8
<b>EXTFC[3:0]</b>	次数	<b>fSAMP</b>																																											
4'b0000		滤波器禁能.																																											
4'b0001	2	$f_{\text{TIMER\_CK}}$																																											
4'b0010	4																																												
4'b0011	8																																												
4'b0100	6																																												
4'b0101	8	$f_{\text{DTS\_CK}}/2$																																											
4'b0110	6																																												
4'b0111	8																																												
4'b1000	6																																												
4'b1001	8	$f_{\text{DTS\_CK}}/4$																																											
4'b1010	5																																												
4'b1011	6																																												
4'b1100	8																																												
4'b1101	5	$f_{\text{DTS\_CK}}/16$																																											
4'b1110	6																																												
4'b1111	8																																												
4'b1111	8																																												
7	<b>MSM</b>	<p>主-从模式</p> <p>该位用来同步被选择的定时器同时开始计数。TRGI用做启动事件，通过TRGO，定时器被连接在一起。</p> <p>0: 主从模式禁能</p> <p>1: 主从模式使能</p>																																											
6:4	<b>TRGS[2:0]</b>	<p>触发选择</p> <p>该位域用来指定选择哪一个信号作为用来同步定时器的触发输入源</p> <p>000: ITI0</p> <p>001: ITI1</p> <p>010: ITI2</p> <p>011: ITI3</p> <p>100: CI0F_ED</p>																																											

101: CI0FE0

110: CI1FE1

111: ETIFP

从模式被使能后这些位不能改。

3	OCRC	OCPRE清除源选择
		0: OCPRE_CLR_INT连接到OCPRE_CLR输入
		1: OCPRE_CLR_INT连接到ETIF
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0。根据CI0FE0的电平，计数器在CI1FE1的边沿向上/下计数。
		010: 编码器模式1。根据CI1FE1的电平，计数器在CI0FE0的边沿向上/下计数。
		011: 编码器模式2。根据另一个信号的输入电平，计数器在CI0FE0和CI1FE1的边沿向上/下计数。
		100: 复位模式。在选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。
		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，则计数器时钟停止。
		110: 事件模式。计数器在触发输入的上升沿启动。
		111: 外部时钟模式0。选中的触发输入的上升沿驱动计数器。

### DMA 和中断使能寄存器 (**TIMERx\_DMAINTEN**)

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	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CHOIE	UPIE
	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	TRGDEN	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	CMTDEN	换相DMA更新请求使能 0: 禁止换相DMA更新请求 1: 使能换相DMA更新请求

---

12	CH3DEN	通道3比较/捕获DMA请求使能 0: 禁止通道3比较/捕获DMA请求 1: 使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获DMA请求使能 0: 禁止通道2比较/捕获DMA请求 1: 使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获DMA请求使能 0: 禁止通道1比较/捕获DMA请求 1: 使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BRKIE	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	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4	CH3IE	通道3比较/捕获中断使能 0: 禁止通道3中断 1: 使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0: 禁止通道2中断 1: 使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0: 禁止通道1中断 1: 使能通道1中断
1	CHOIE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断

1: 使能更新中断

### 中断标志寄存器 (TIMERx\_INTF)

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	CH3OF	CH2OF	CH1OF	CH0OF	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CHOIF	UPIF		

rc\_w0      rc\_w0

位/位域	名称	描述
31:13	保留	必须保持复位值
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时，在CH0IF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: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	保留	必须保持复位值
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，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，此标志会置1，此位由软件清0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0: 无触发事件产生 1: 触发中断产生

5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时，此标志位被硬件置1，此位由软件清0。 0：无通道换相更新中断发生 1：通道换相更新中断发生
4	CH3IF	通道3比较/捕获中断标志 参见CH0IF描述
3	CH2IF	通道2比较/捕获中断标志 参见CH0IF描述
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1，软件清0。 当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0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0中断发生 1：通道0中断发生
0	UPIF	更新中断标志 此位在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

### 软件事件产生寄存器 (**TIMERx\_SWEVG**)

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								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
w								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值
7	BRKG	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，POEN位被清0且BRKIF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	<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，<b>TIMERx_INTF</b>寄存器的<b>TRGIF</b>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	CMTG	<p>通道换相更新事件发生</p> <p>此位由软件置1，由硬件自动清0。当此位被置1，根据CCSE位(<b>TIMERx_CTL1</b>寄存器中)的值，通道捕获/比较控制寄存器(<b>CHxEN</b>, <b>CHxNEN</b> 和<b>CHxCOMCTL</b>)的互补输出被更新。</p> <p>0: 不产生通道换相更新事件</p> <p>1: 产生通道换相更新事件</p>
4	CH3G	<p>通道3捕获或比较事件发生</p> <p>参见<b>CH0G</b>描述</p>
3	CH2G	<p>通道2捕获或比较事件发生</p> <p>参见<b>CH0G</b>描述</p>
2	CH1G	<p>通道1捕获或比较事件发生</p> <p>参见<b>CH0G</b>描述</p>
1	CH0G	<p>通道0捕获或比较事件发生</p> <p>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，<b>CH0IF</b>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0配置为输入模式，计数器的当前值被捕获到<b>TIMERx_CH0CV</b>寄存器，如果<b>CH0IF</b>标志位已经为1，则<b>CH0OF</b>标志位被置1。</p> <p>0: 不产生通道0捕获或比较事件</p> <p>1: 发生通道0捕获或比较事件</p>
0	UPG	<p>更新事件产生</p> <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>

### 通道控制寄存器 0 (**TIMERx\_CHCTL0**)

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
CH1COM CEN	CH1COMCTL[2:0]	CH1COM SEN	CH1COM FEN	CH1MS[1:0]	CH0COM CEN	CH0COMCTL[2:0]	CH0COM SEN	CH0COM FEN	CH0MS[1:0]						
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]				CH0CAPFLT[3:0]	CH0CAPPSC[1:0]									

rw rw rw rw rw rw

### 输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH1COMCEN	通道1输出比较清0使能 参见CH0COMCEN描述
14:12	CH1COMCTL[2:0]	通道1输出比较模式 参见CH0COMCTL描述
11	CH1COMSEN	通道1输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH1COMFEN	通道1输出比较快速使能 参见CH0COMFEN描述
9:8	CH1MS[1:0]	通道 1 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_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00: 通道 1 配置为输出 01: 通道 1 配置为输入, IS1 映射在 CI1FE1 上 10: 通道 1 配置为输入, IS1 映射在 CI0FE1 上 11: 通道 1 配置为输入, IS1 映射在 ITS 上 注意: 当CH1MS[1:0]=11时, 需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1, 当检测到 ETIFP 信号输入高电平时, O0CPRE 参考信号被清 0 0: 禁止通道 0 输出比较清零 1: 使能通道0输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式, 而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外, O0CPRE 高电平有效, 而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000: 时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 为低。

011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 **TIMERx\_CH0CV** 相同时，强制 **O0CPRE** 翻转。

100: 强制为低。强制 **O0CPRE** 为低电平

101: 强制为高。强制 **O0CPRE** 为高电平

110: PWM 模式 0。在向上计数时，一旦计数器值小于 **TIMERx\_CH0CV** 时，**O0CPRE** 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\_CH0CV** 时，**O0CPRE** 为低电平，否则为高电平。

111: PWM 模式 1。在向上计数时，一旦计数器值小于 **TIMERx\_CH0CV** 时，**O0CPRE** 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\_CH0CV** 时，**O0CPRE** 为高电平，否则为低电平。

如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，**O0CPRE** 电平才改变。

当 **TIMERx\_CCHP** 寄存器的 **PROT[1:0]=11** 且 **CH0MS=00** (比较模式) 时此位不能被改变。

3	<b>CH0COMSEN</b>	通道 0 输出比较影子寄存器使能 当此位被置 1， <b>TIMERx_CH0CV</b> 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 0 输出/比较影子寄存器 1: 使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下( <b>SPM =1</b> )，可以在未确认影子寄存器的情况下使用 PWM 模式 当 <b>TIMERx_CCHP</b> 寄存器的 <b>PROT [1:0]=11</b> 且 <b>CH0MS =00</b> 时此位不能被改变。
2	<b>CH0COMFEN</b>	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较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， <b>CH0_O</b> 被设置为比较电平而与比较结果无关。 0: 禁止通道 0 输出比较快速。 1: 使能通道 0 输出比较快速。
1:0	<b>CH0MS[1:0]</b>	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 ( <b>TIMERx_CHCTL2</b> 寄存器的 <b>CH0EN</b> 位被清 0) 时这些位才可写。 00: 通道 0 配置为输出 01: 通道 0 配置为输入，ISO 映射在 <b>CI0FE0</b> 上 10: 通道 0 配置为输入，ISO 映射在 <b>CI1FE0</b> 上 11: 通道 0 配置为输入，ISO 映射在 <b>ITS</b> 上 注意：当 <b>CH0MS[1:0]=11</b> 时，需要通过 <b>TRGS</b> 位（位于 <b>TIMERx_SMCFG</b> 寄存器）选择内部触发输入。

#### 输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	<b>CH1CAPFLT[3:0]</b>	通道 1 输入捕获滤波控制

参见CH0CAPFLT描述

11:10 CH1CAPPSC[1:0] 通道1输入捕获预分频器  
参见CH0CAPPSC描述

9:8 CH1MS[1:0] 通道1模式选择  
与输出模式相同

7:4 CH0CAPFLT[3:0] 通道0输入捕获滤波控制  
**C10** 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。  
数字滤波器的基本原理：根据  $f_{SAMP}$  对 C10 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。  
滤波器参数配置如下：

<b>CH0CAPFLT [3:0]</b>	<b>采样次数</b>	<b><math>f_{SAMP}</math></b>
4'b0000		无滤波器
4'b0001	2	$f_{CK\_TIMER}$
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] 通道0输入捕获预分频器  
这2位定义了通道0输入的预分频系数。当TIMERx\_CHCTL2寄存器中的CH0EN=0时，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 CH0MS[1:0] 通道0模式选择  
与输出比较模式相同

### 通道控制寄存器 1 (TIMERx\_CHCTL1)

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
CH3COM CEN	CH3COMCTL[2:0]	CH3COM SEN	CH3COM FEN	CH3MS[1:0]	CH2COM CEN	CH2COMCTL[2:0]	CH2COM SEN	CH2COM FEN	CH2MS[1:0]						
CH3CAPFLT[3:0]	CH3CAPPSC[1:0]	CH2CAPFLT[3:0]	CH2CAPPSC[1:0]												
rw		rw		rw		rw		rw		rw		rw		rw	

### 输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH3COMCEN	通道3输出比较清0使能 参见CH0COMCEN描述
14:12	CH3COMCTL[2:0]	通道3输出比较模式 参见CH0COMCTL描述
11	CH3COMSEN	通道3输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH3COMFEN	通道3输出比较快速使能 参见CH0COMFEN描述
9:8	CH3MS[1:0]	通道 3 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_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 00: 通道 3 配置为输出 01: 通道 3 配置为输入, IS3 映射在 CI3FE3 上 10: 通道 3 配置为输入, IS3 映射在 CI2FE3 上 11: 通道 3 配置为输入, IS3 映射在 ITS 上 注意: 当CH3MS[1:0]=11时, 需要通过TRGS位(位于TIMERx_SMCFG 寄存器)选择内部触发输入。
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1, 当检测到 ETIPP 输入高电平时, O2CPRE 参考信号被清 0 0: 使能通道 2 输出比较清零 1: 禁止通道2输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式, 而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外, O2CPRE 高电平有效, 而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000: 时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用

001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 **TIMERx\_CH2CV** 相同时，强制 **O2CPRE** 为高。

010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 **TIMERx\_CH2CV** 相同时，强制 **O2CPRE** 为低。

011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 **TIMERx\_CH2CV** 相同时，强制 **O2CPRE** 翻转。

100: 强制为低。强制 **O2CPRE** 为低电平

101: 强制为高。强制 **O2CPRE** 为高电平

110: PWM 模式 0。在向上计数时，一旦计数器值小于 **TIMERx\_CH2CV** 时，**O2CPRE** 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\_CH2CV** 时，**O2CPRE** 为低电平，否则为高电平。

111: PWM 模式 1。在向上计数时，一旦计数器值小于 **TIMERx\_CH2CV** 时，**O2CPRE** 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\_CH2CV** 时，**O2CPRE** 为高电平，否则为低电平。

如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，**O2CPRE** 电平才改变。

当 **TIMERx\_CCHP** 寄存器的 **PROT[1:0]=11** 且 **CH2MS=00** (比较模式) 时此位不能被改变。

3	<b>CH2COMSEN</b>	<p>通道 0 输出比较影子寄存器使能</p> <p>当此位被置 1，<b>TIMERx_CH2CV</b> 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 2 输出/比较影子寄存器</p> <p>1: 使能通道 2 输出/比较影子寄存器</p> <p>仅在单脉冲模式下(<b>SPM =1</b>)，可以在未确认影子寄存器情况下使用 PWM 模式</p> <p>当 <b>TIMERx_CCHP</b> 寄存器的 <b>PROT [1:0]=11</b> 且 <b>CH2MS =00</b> 时此位不能被改变。</p>
2	<b>CH2COMFEN</b>	<p>通道 2 输出比较快速使能</p> <p>当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较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，<b>CH2_O</b> 被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道 2 输出比较快速.</p> <p>1: 使能通道2输出比较快速。</p>
1:0	<b>CH2MS[1:0]</b>	<p>通道 2 I/O 模式选择</p> <p>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (<b>TIMERx_CHCTL2</b> 寄存器的 <b>CH2EN</b> 位被清 0) 时这些位才可写。</p> <p>00: 通道 2 配置为输出</p> <p>01: 通道 2 配置为输入，IS2 映射在 CI2FE2 上</p> <p>10: 通道 2 配置为输入，IS2 映射在 CI3FE2 上</p> <p>11: 通道 2 配置为输入，IS2 映射在 ITS 上.</p> <p>注意：当 <b>CH2MS[1:0]=11</b> 时，需要通过 <b>TRGS</b> 位（位于 <b>TIMERx_SMCFG</b> 寄存器）选择内部触发输入。</p>

## 输入捕获模式：

位/位域	名称	描述																																										
31:16	保留	必须保持复位值																																										
15:12	CH3CAPFLT[3:0]	通道3输入捕获滤波控制 参见CH0CAPFLT描述																																										
11:10	CH3CAPPSC[1:0]	通道3输入捕获预分频器 参见CH0CAPPSC描述																																										
9:8	CH3MS[1:0]	通道3模式选择 与输出模式相同																																										
7:4	CH2CAPFLT[3:0]	通道 2 输入捕获滤波控制  CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 $f_{SAMP}$ 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：																																										
		<table border="1"> <thead> <tr> <th>CH2CAPFLT [3:0]</th><th>采样次数</th><th><math>f_{SAMP}</math></th></tr> </thead> <tbody> <tr> <td>4'b0000</td><td></td><td>无滤波器</td></tr> <tr> <td>4'b0001</td><td>2</td><td rowspan="3"><math>f_{CK\_TIMER}</math></td></tr> <tr> <td>4'b0010</td><td>4</td></tr> <tr> <td>4'b0011</td><td>8</td></tr> <tr> <td>4'b0100</td><td>6</td><td rowspan="2"><math>f_{DTS}/2</math></td></tr> <tr> <td>4'b0101</td><td>8</td></tr> <tr> <td>4'b0110</td><td>6</td><td rowspan="2"><math>f_{DTS}/4</math></td></tr> <tr> <td>4'b0111</td><td>8</td></tr> <tr> <td>4'b1000</td><td>6</td><td rowspan="2"><math>f_{DTS}/8</math></td></tr> <tr> <td>4'b1001</td><td>8</td></tr> <tr> <td>4'b1010</td><td>5</td><td rowspan="3"><math>f_{DTS}/16</math></td></tr> <tr> <td>4'b1011</td><td>6</td></tr> <tr> <td>4'b1100</td><td>8</td></tr> <tr> <td>4'b1101</td><td>5</td><td rowspan="3"><math>f_{DTS}/32</math></td></tr> <tr> <td>4'b1110</td><td>6</td></tr> <tr> <td>4'b1111</td><td>8</td></tr> </tbody> </table>	CH2CAPFLT [3:0]	采样次数	$f_{SAMP}$	4'b0000		无滤波器	4'b0001	2	$f_{CK\_TIMER}$	4'b0010	4	4'b0011	8	4'b0100	6	$f_{DTS}/2$	4'b0101	8	4'b0110	6	$f_{DTS}/4$	4'b0111	8	4'b1000	6	$f_{DTS}/8$	4'b1001	8	4'b1010	5	$f_{DTS}/16$	4'b1011	6	4'b1100	8	4'b1101	5	$f_{DTS}/32$	4'b1110	6	4'b1111	8
CH2CAPFLT [3:0]	采样次数	$f_{SAMP}$																																										
4'b0000		无滤波器																																										
4'b0001	2	$f_{CK\_TIMER}$																																										
4'b0010	4																																											
4'b0011	8																																											
4'b0100	6	$f_{DTS}/2$																																										
4'b0101	8																																											
4'b0110	6	$f_{DTS}/4$																																										
4'b0111	8																																											
4'b1000	6	$f_{DTS}/8$																																										
4'b1001	8																																											
4'b1010	5	$f_{DTS}/16$																																										
4'b1011	6																																											
4'b1100	8																																											
4'b1101	5	$f_{DTS}/32$																																										
4'b1110	6																																											
4'b1111	8																																											
3:2	CH2CAPPSC[1:0]	通道2输入捕获预分频器  这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时，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	CH2MS[1:0]	通道2模式选择 与输出比较模式相同																																										

### 通道控制寄存器 2 (TIMERx\_CHCTL2)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN	

位/位域	名称	描述
31:14	保留	必须保持复位值
13	CH3P	通道3极性 参考CH0P描述
12	CH3EN	通道3使能 参考CH0EN描述
11	CH2NP	通道2互补输出极性 参考CH0NP描述
10	CH2NEN	通道2互补输出使能 参考CH0NEN描述
9	CH2P	通道2极性 参考CH0P描述
8	CH2EN	通道2使能 参考CH0EN描述
7	CH1NP	通道1互补输出极性 参考CH0NP描述
6	CH1NEN	通道1互补输出使能 参考CH0NEN描述
5	CH1P	通道1极性 参考CH0P描述
4	CH1EN	通道1使能 参考CH0EN描述
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0: 通道0互补输出高电平为有效电平

**1:** 通道0互补输出低电平为有效电平

当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号Cl0的极性选择控制信号。

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

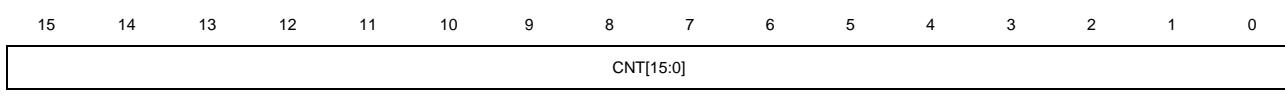
2	CH0NEN	通道0互补输出使能 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。  0: 禁止通道0互补输出 1: 使能通道0互补输出
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。  0: 通道0高电平为有效电平 1: 通道0低电平为有效电平 当通道0配置为输入模式时，此位定义了Cl0信号极性。 CH0P将选择Cl0FE0或者Cl1FE0的有效边沿或者捕获极性。 CH0P=0: 把ClxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且ClxFE0不会被翻转。 CH0P=1: 把ClxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且ClxFE0会被翻转。 当TIMERx_CCHP寄存器的PROT[1:0]=11或10 时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。  0: 禁止通道0 1: 使能通道0

### 计数器寄存器 (TIMERx\_CNT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



rw

位/位域	名称	描述
31:16	保留	必须保持复位值
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。

### 预分频寄存器 (TIMERx\_PSC)

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
PSC[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (TIMERx\_CAR)

地址偏移: 0x2C

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
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。 <b>注意:</b> 在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非0值(例如0xFFFF)。

### 重复计数寄存器 (TIMERx\_CREP)

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								CREP[7:0]							

rw

位/位域	名称	描述
31:8	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)。

### 通道 0 捕获/比较寄存器 (**TIMERx\_CH0CV**)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道0的捕获或比较值 当通道0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0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。

### 通道 1 捕获/比较寄存器 (**TIMERx\_CH1CV**)

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

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

CH1VAL[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[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>

### 通道 2 捕获/比较寄存器 (TIMERx\_CH2CV)

地址偏移: 0x3C

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

CH2VAL[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH2VAL[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>

### 通道 3 捕获/比较寄存器 (TIMERx\_CH3CV)

地址偏移: 0x40

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
CH3VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[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。

### 互补通道保护寄存器 (**TIMERx\_CCHP**)

地址偏移: 0x44

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
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]									DTCFG[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
31:16	保留	必须保持复位值
15	POEN	所有的通道输出使能 该位通过以下方式置 1: -写 1 置位 -如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0: -写 0 清 0 -有效的中止输入（异步） 如果一个通道配置为输出模式，如果设置了相应的使能位（ <b>TIMERx_CHCTL2</b> 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。 0: 禁止通道输出 1: 使能通道输出 注意：仅当 CHxMS[1:0]=2'b00 时该位有效
14	OAEN	自动输出使能

		0: POEN 位只能使用软件方式置 1。 1: 如果中止输入无效, 下一次更新事件发生时, POEN 位将会置 1 此位只有在 TIMERx_CCHP 寄存器的 PROT [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 BRKIN 的极性。 0: 中止输入低电平有效 1: 中止输入高电平有效
12	BRKEN	中止使能 此位置 1 使能中止事件和 CKM 时钟失败事件输入。 0: 禁能中止输入 1: 使能中止输入 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下“关闭状态”配置 当 POEN 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 POEN 位被置 1, 通道输出信号 (CHx_O/ CHx_ON)被禁止。 1: 当 POEN 位被置 1, 通道输出信号 (CHx_O / CHx_ON)被使能, 和 TIMER0_CHCTL2 寄存器 CHxEN/CHxNEN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [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 POEN 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 POEN 位被清 0, 通道输出信号(CHx_O/ CHx_ON)被禁止。 1: 当 POEN 位被清 0, 通道输出信号(CHx_O / CHx_ON)被使能, 和 TIMERx_CHCTL2 寄存器 CHxEN/CHxNEN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00: 禁能保护模式。无写保护。 01: PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位, TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。 10: PROT 模式 1.除了 PROT 模式 0 下的寄存器写保护外, 还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位 (如果相应通道配置为输出模式), TIMERx_CCHP 寄存器中 ROS/IOS 位写保护。 11: PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外, 还有 TIMERx_CHCTRL0/1 中 CHxCOMCTL/CHxCOMSEN 位 (如果相关通道配置为输出模式) 写保护。 系统复位后这两位只能被写一次, 一旦 TIMERx_CCHP 寄存器被写入, 这两位被写保护。
7:0	DTCFG[7:0]	死区时间控制 DTCFG 值和死区时间的关系如下:

DTCFG[7:5]	The duration of dead-time
3'b0xx	DTCFG[7:0] * t <sub>DTS_CK</sub>

3'b10x	$(64 + \text{DTCFG}[5:0]) * t_{\text{DTS\_CK}} * 2$
3'b110	$(32 + \text{DTCFG}[4:0]) * t_{\text{DTS\_CK}} * 8$
3'b111	$(32 + \text{DTCFG}[4:0]) * t_{\text{DTS\_CK}} * 16$

注意：

1.  $t_{\text{DTS\_CK}}$  是 DTS\_CK 的周期，由 TIMERx\_CTL0 中的 CKDIC[1:0] 定义。
2. 此位只有在 TIMERx\_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

### DMA 配置寄存器 (TIMERx\_DMACFG)

地址偏移：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
保留		DMATC[4:0]		保留		DMATA [4:0]									rw

位/位域	名称	描述
31:14	保留	必须保持复位值
12:8	DMATC[4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMATB寄存器的数量n, $n = (\text{DMATC } [4:0] + 1)$ . DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值
4:0	DMATA[4:0]	DMA传输起始地址 该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当第一次访问TIMERx_DMATB寄存器时，实际访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问（起始地址+0x4）。

### DMA 发送缓冲区寄存器 (TIMERx\_DMATB)

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DMATB[15:0]	DMA发送缓冲 对这个寄存器的读或写，从（起始地址）到（起始地址+传输次数*4）地址范围内的寄存器会被访问。传输次数由硬件计算，范围为0到DMATC。

### 配置寄存器 (TIMERx\_CFG)

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														CHVSEL	OUTSEL

rw rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	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: 无影响。
0	OUTSEL	输出值选择位 此位由软件写1或清0。 1: 如果POEN位与IOS位均为0，则输出无效。 0: 无影响。

## 14.2. 通用定时器 L0 (TIMERx, x=2)

### 14.2.1. 简介

通用定时器L0 (TIMER2) 是4通道定时器，支持输入捕获，输出比较，产生PWM信号控制电机和电源管理。通用定时器L0计数器是16位无符号计数器。

通用定时器L0是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

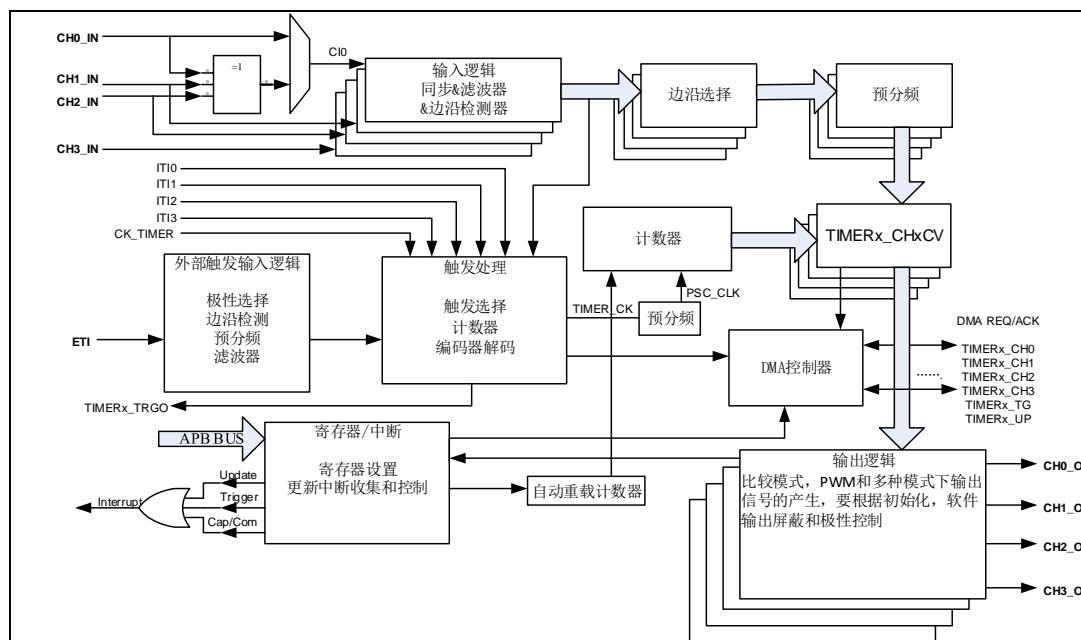
### 14.2.2. 主要特征

- 总通道数：4;
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交编码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件；
- 多个定时器的菊花链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

### 14.2.3. 结构框图

[图14-31. 通用定时器L0结构框图](#)提供了通用定时器L0的内部细节。

图 14-31. 通用定时器 L0 结构框图



#### 14.2.4. 功能说明

##### 时钟源配置

通用定时器L0可以是内部时钟源CK\_TIMER，或者是由SMC(TIMERx\_SMCFG寄存器位[2:0])位确定的时钟源。

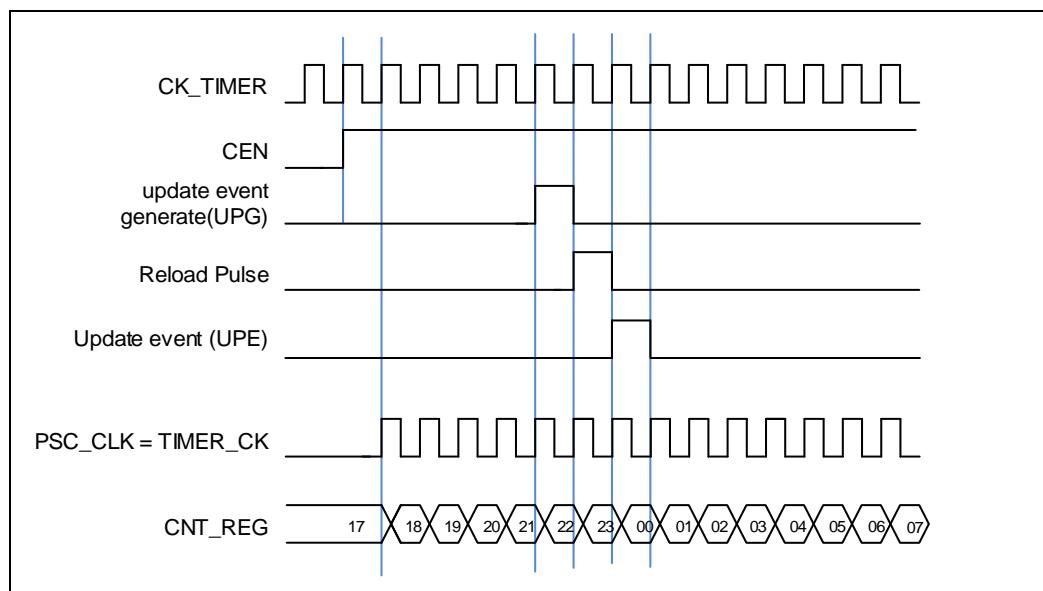
- SMC[2:0]=3'b000, 定时器选择内部时钟源（连接到RCU模块的CK\_TIMER）

如果SMC[2:0]=3'b000, 默认用来驱动计数器预分频器的是内部时钟源CK\_TIMER。当CEN置位, CK\_TIMER经过预分频器（预分频值由TIMERx\_PSC寄存器确定）产生PSC\_CLK。

这种模式下, 驱动预分频器计数的TIMER\_CK等于来自于RCU模块的CK\_TIMER。

如果将TIMERx\_SMCFG寄存器的SMC[2:0]设置为0x1、0x2、0x3和0x7, 预分频器被其他时钟源(由TIMERx\_SMCFG寄存器的TRGS[2:0]区域选择)驱动, 在下文说明。当SMC位被设置为0x4、0x5和0x6, 计数器预分频器时钟源由内部时钟CK\_TIMER驱动。

图 14-32. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]=3'b111 (外部时钟模式0)，定时器选择外部输入引脚作为时钟源。

计数器预分频器可以在TIMERx\_CH0/TIMERx\_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。Clx是TIMERx\_Cl通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号ITIO/1/2/3的上升沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x0, 0x1, 0x2 或者0x3。

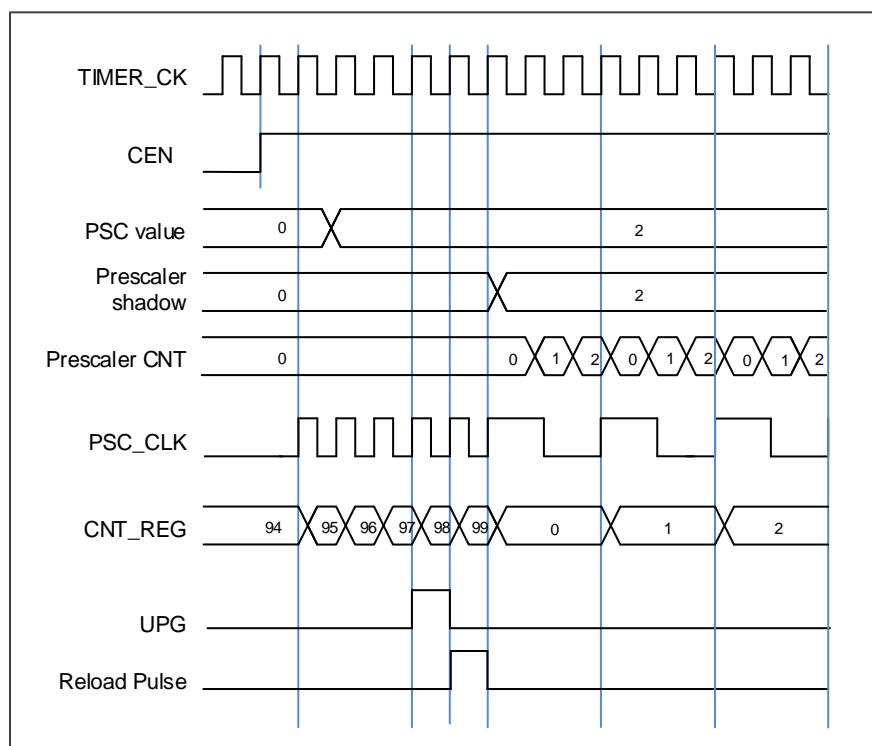
- SMC1=1'b1 (外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源。

计数器预分频器可以在外部引脚ETI的每个上升沿或下降沿计数。这种模式可以通过设置TIMERx\_SMCFG寄存器中的SMC1位为1来选择。另一种选择ETI信号作为时钟源方式是，设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x7。注意ETI信号是通过数字滤波器采样ETI引脚得到的。如果选择ETI信号为时钟源，触发控制器包括边沿监测电路将在每个ETI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。

### 时钟预分频器

预分频器可以将定时器的时钟(TIMER\_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC\_CLK驱动计数器计数。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 14-33. 当 PSC 数值从 0 变到 2 时，计数器的时序图



### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在`TIMERx_CAR`寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，`TIMERx_CTL0`寄存器中的计数方向控制位`DIR`应该被设置成0。

当通过`TIMERx_SWEVG`寄存器的`UPG`位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果`TIMERx_CTL0`寄存器的`UPDIS`置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

[图 14-34. 向上计数时序图, PSC=0/2](#) 和 [图 14-35. 向上计数时序图, 在运行时改变 TIMERx CAR 寄存器的值](#)给出了一些例子，当 `TIMERx_CAR=0x99` 时，计数器在不同预分频因子下的行为。

图 14-34. 向上计数时序图, PSC=0/2

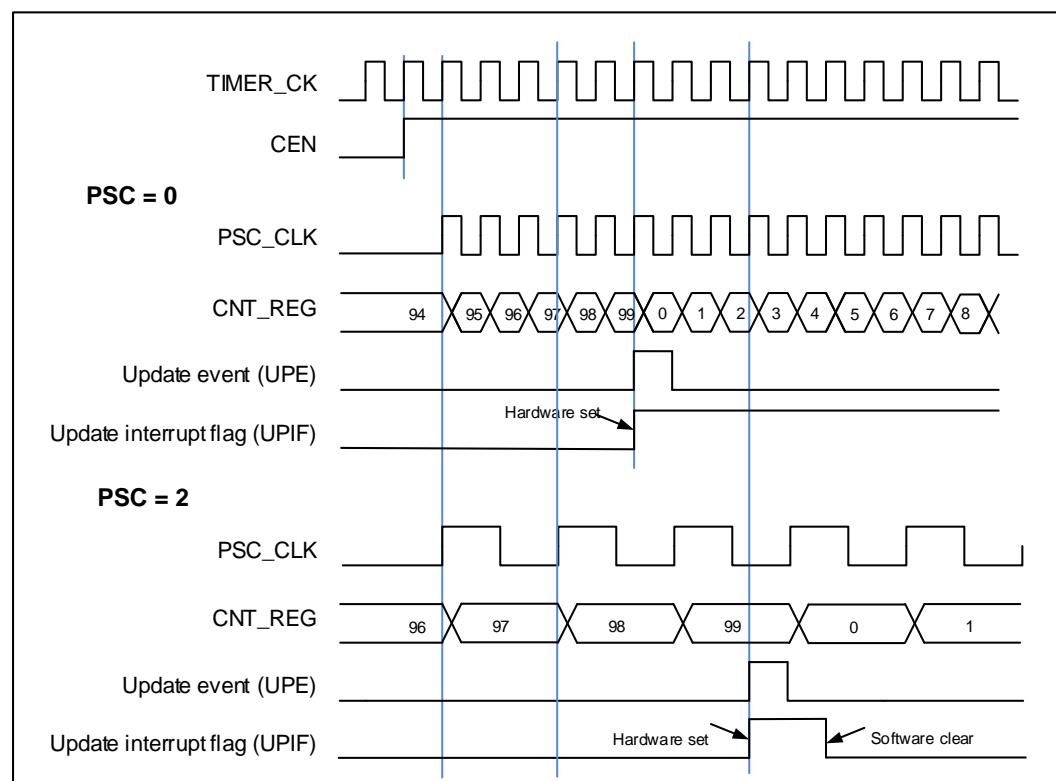
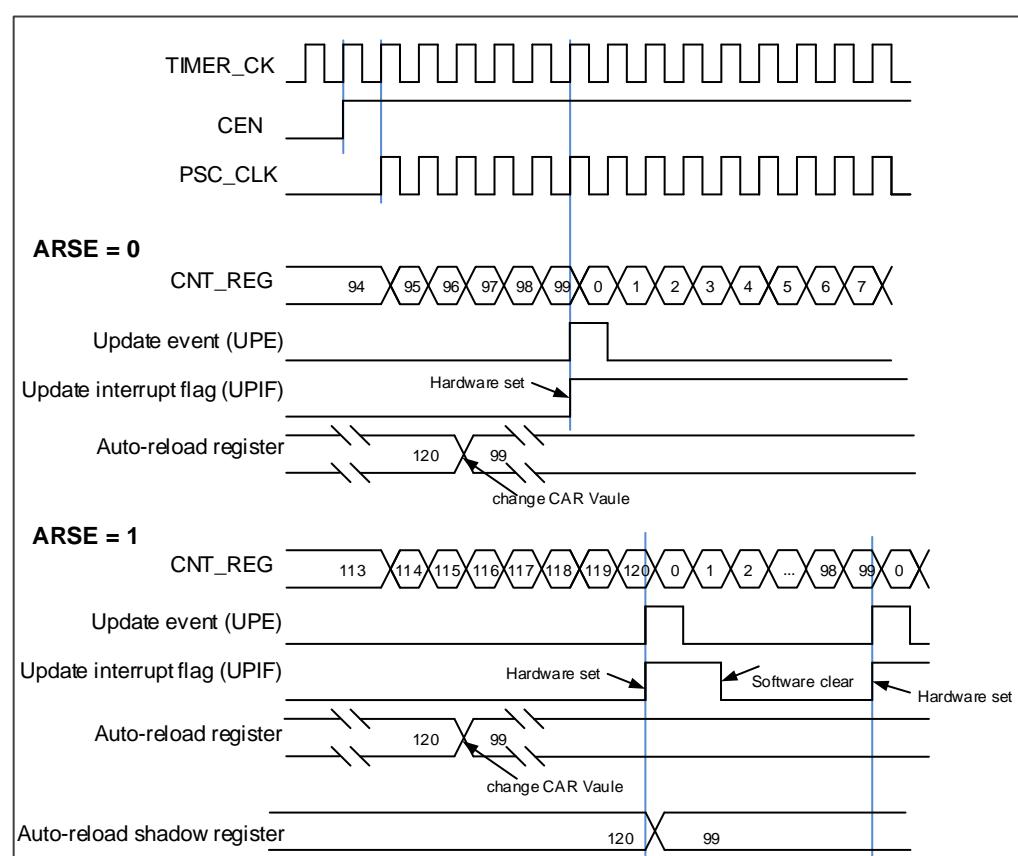


图 14-35. 向上计数时序图, 在运行时改变 TIMERx\_CAR 寄存器的值



## 计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx\_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数并产生下溢。在向下计数模式中，TIMERx\_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

当通过TIMERx\_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果TIMERx\_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

[图 14-36. 向下计数时序图，PSC=0/2](#) 和 [图 14-37. 向下计数时序图，在运行时改变 TIMERx CAR 寄存器值](#)给出了一些例子，当 TIMERx\_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 14-36. 向下计数时序图，PSC=0/2

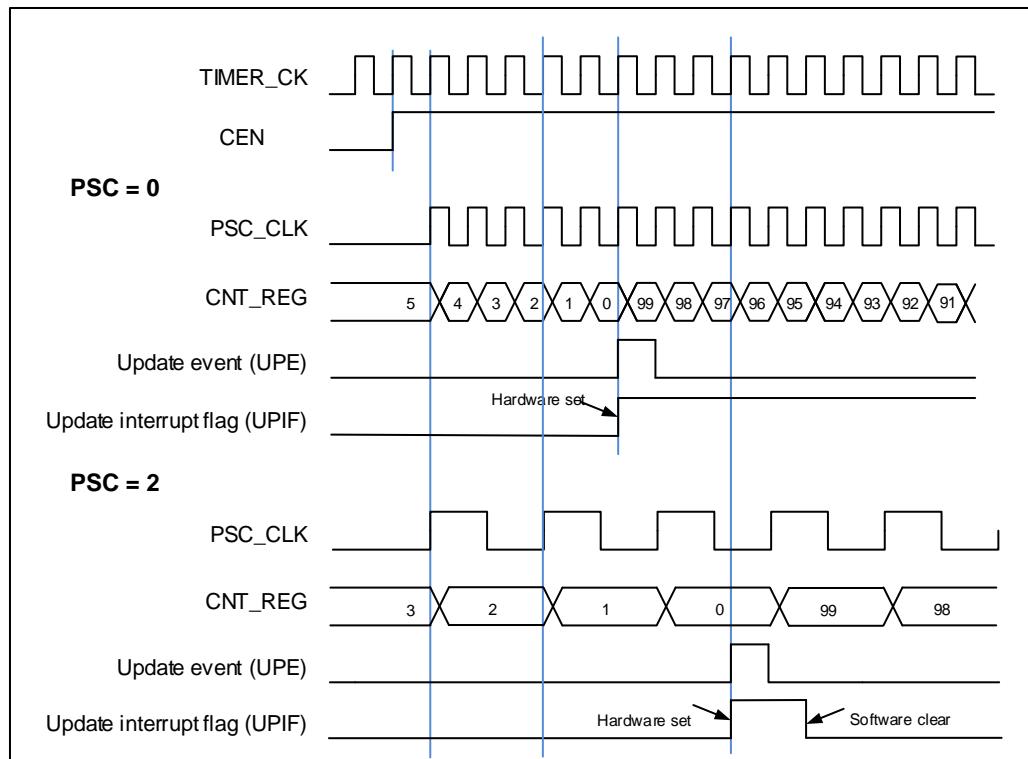
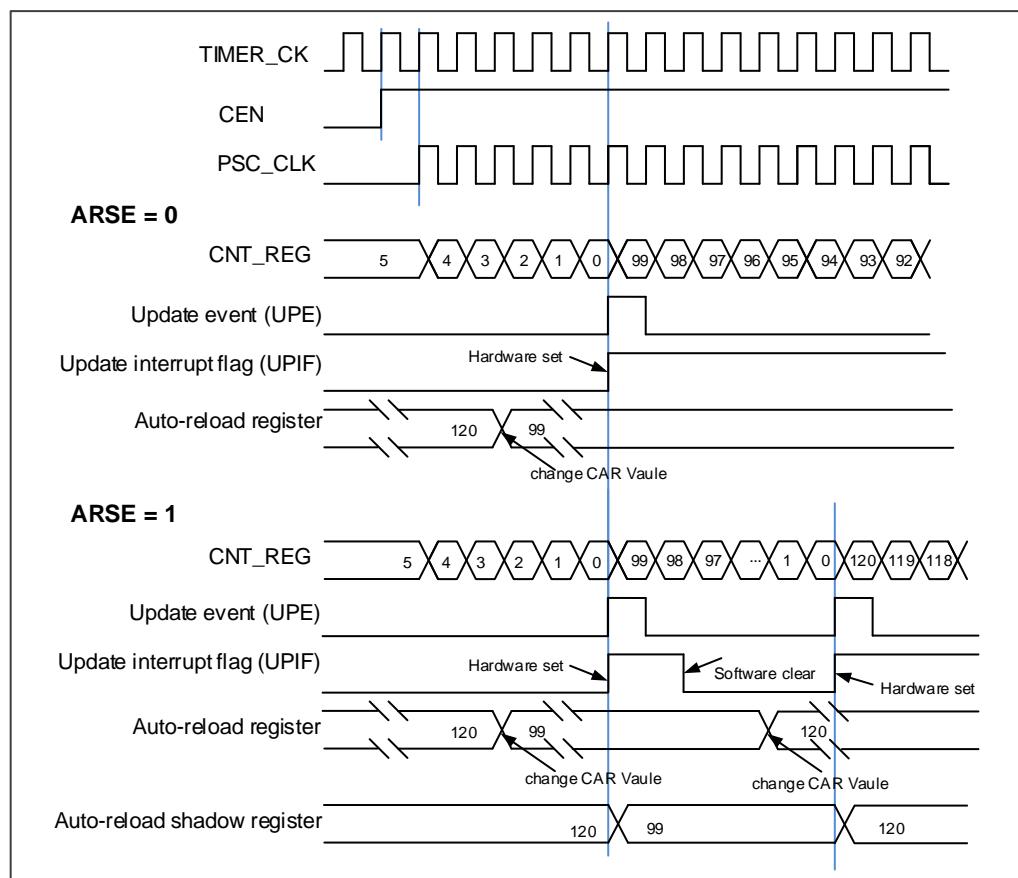


图 14-37. 向下计数时序图，在运行时改变 TIMERx\_CAR 寄存器值



### 计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 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。在中央计数模式中，TIMERx\_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。

将 TIMERx\_SWEVG 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。

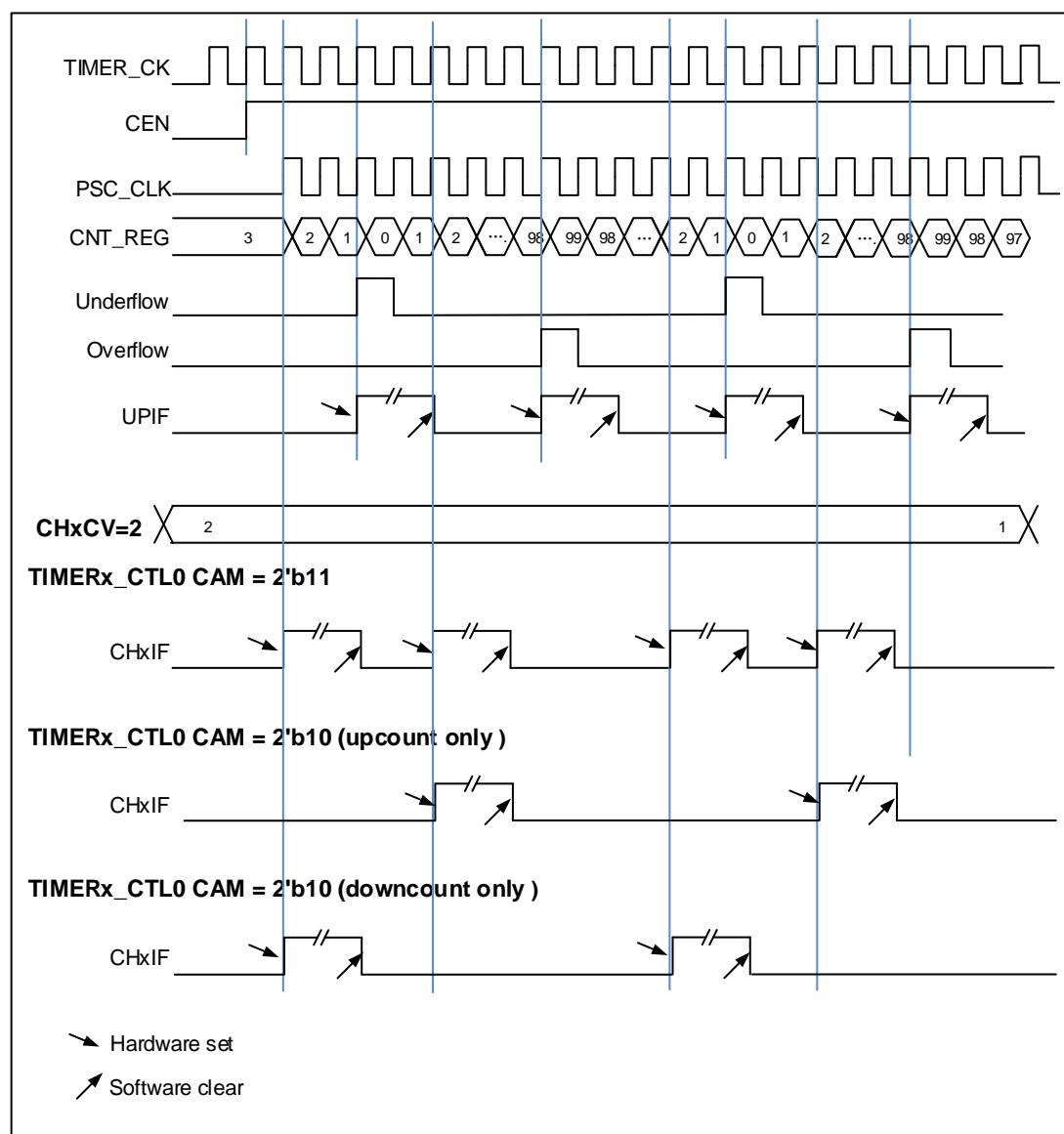
上溢或者下溢时，TIMERx\_INTF 寄存器中的 UPIF 位都会被置 1。但是 CHxIF 位是否置 1 与 TIMERx\_CTL0 寄存器中 CAM 的值有关。具体细节参考 [图 14-38. 中央计数模式计数器时序图](#)。

如果 TIMERx\_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

[图 14-38. 中央计数模式计数器时序图](#)给出了一些例子，当 TIMERx\_CAR=0x99，TIMERx\_PSC=0x0 时，计数器的行为。

图 14-38. 中央计数模式计数器时序图



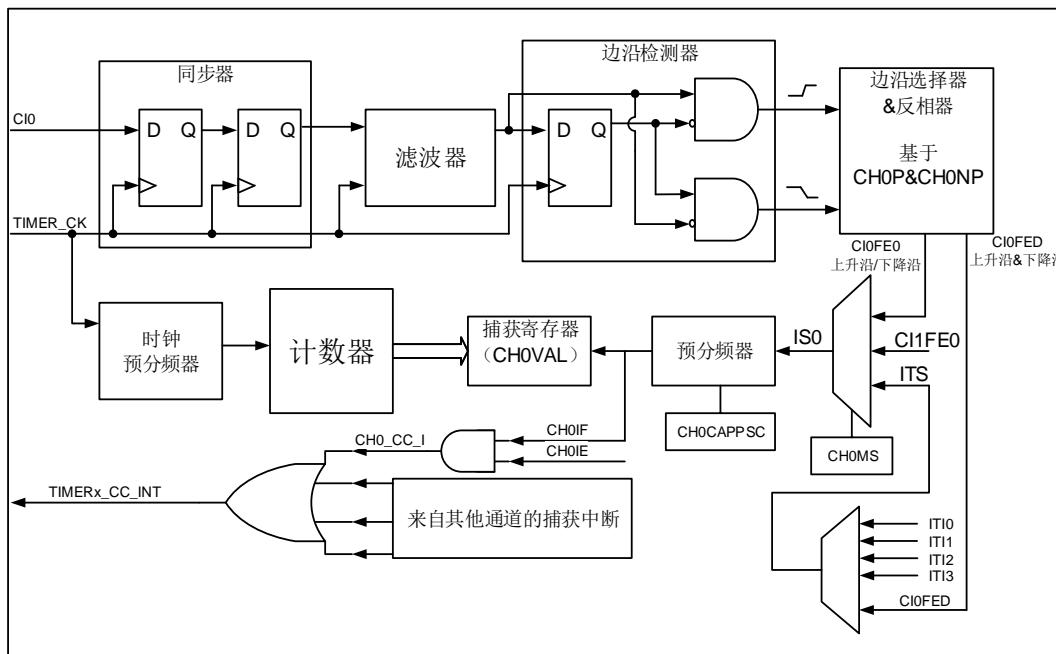
### 输入捕获和输出比较通道

通用定时器L0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。

#### ■ 通道输入捕获功能

输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。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。如果在输入引脚上出现被选择的边沿，**TIMERx\_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，如果**CHxIE** = 1则产生通道中断。

图 14-39. 通道输入捕获原理



通道输入信号  $Cl_x$  有两种选择，一种是  $TIMERx\_CHx$  信号，另一种是  $TIMERx\_CH0, TIMERx\_CH1$  和  $TIMERx\_CH2$  异或之后的信号。通道输入信号  $Cl_x$  先被  $TIMER\_CK$  信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置  $CHxP$  选择使用上升沿或者下降沿。配置  $CHxMS$ ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $CHxVAL$  存储计数器的值。

配置步骤如下：

**第一步：滤波器配置 (TIMERx\_CHCTL0 寄存器中 CHxCAPFLT):**

根据输入信号和请求信号的质量，配置相应的  $CHxCAPFLT$ 。

**第二步：边沿选择 (TIMERx\_CHCTL2 寄存器中 CHxP):**

配置  $CHxP$  选择上升沿或者下降沿。

**第三步：捕获源选择 (TIMERx\_CHCTL0 寄存器中 CHxMS):**

一旦通过配置  $CHxMS$  选择输入捕获源，必须确保通道配置在输入模式 ( $CHxMS \neq 0x0$ )，而且  $TIMERx\_CHxCV$  寄存器不能再被写。

**第四步：中断使能 (TIMERx\_DMAINTEN 寄存器中 CHxIE 和 CHxDEN):**

使能相应中断，可以获得中断和 DMA 请求。

**第五步：捕获使能 (TIMERx\_CHCTL2 寄存器中 CHxEN)。**

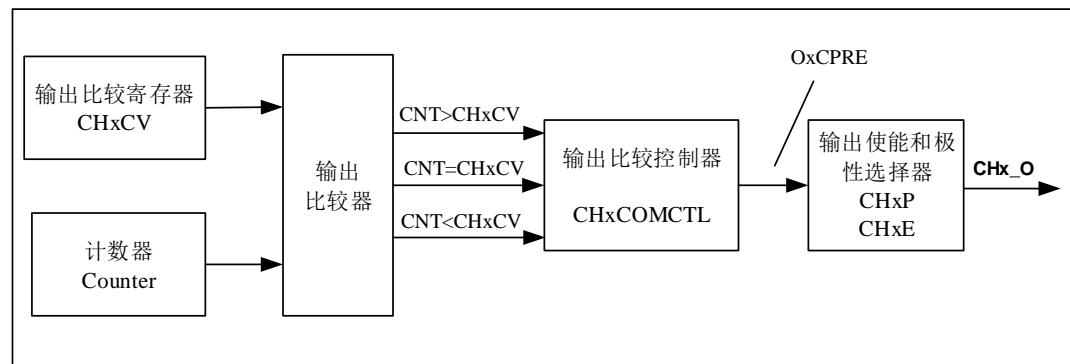
**结果：**当期望的输入信号发生时， $TIMERx\_CHxCV$  被设置成当前计数器的值， $CHxIF$  为置 1。如果  $CHxIF$  位已经为 1，则  $CHxOF$  位置 1。根据  $TIMERx\_DMAINTEN$  寄存器中  $CHxIE$  和  $CHxDEN$  的配置，相应的中断和 DMA 请求会被提出。

**直接产生：**软件设置  $CHxG$  位，会直接产生中断和 DMA 请求。

输入捕获模式也可用来测量TIMERx\_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx\_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx\_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX\_CH0CV寄存器测量PWM的周期值，TIMERx\_CH1CV寄存器测量PWM占空比值。

### ■ 通道输出比较功能

图 14-40. 通道输出比较原理 ( $x=0,1,2,3$ )



**图14-40. 通道输出比较原理 ( $x=0,1,2,3$ )** 给出了输出比较的逻辑电路。通道输出信号CHx\_O与OxCPR信号（详情请见[通道输出准备信号](#)）的关系描述如下：OxCPR信号高电平有效，CHx\_O的输出情况与OxCPR信号，CHxP位和CHxE位有关（具体情况请见TIMERx\_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx\_O高电平有效，与OxCPR输出极性相同）、CHxE=1（CHx\_O输出使能）时：

- 若OxCPR输出有效（高）电平，则CHx\_O输出有效（高）电平；
- 若OxCPR输出无效（低）电平，则CHx\_O输出无效（低）电平。

在通道输出比较功能，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxODE=1则会产生DMA请求。

配置步骤如下：

#### 第一步：时钟配置：

配置定时器时钟源，预分频器等。

#### 第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP位来选择有效电平的极性；

设置CHxEN使能输出。

#### 第三步：通过CHxIE/CxDDE位配置中断/DMA请求使能。

#### 第四步：通过TIMERx\_CAR寄存器和TIMERx\_CHxCV寄存器配置输出比较时基：

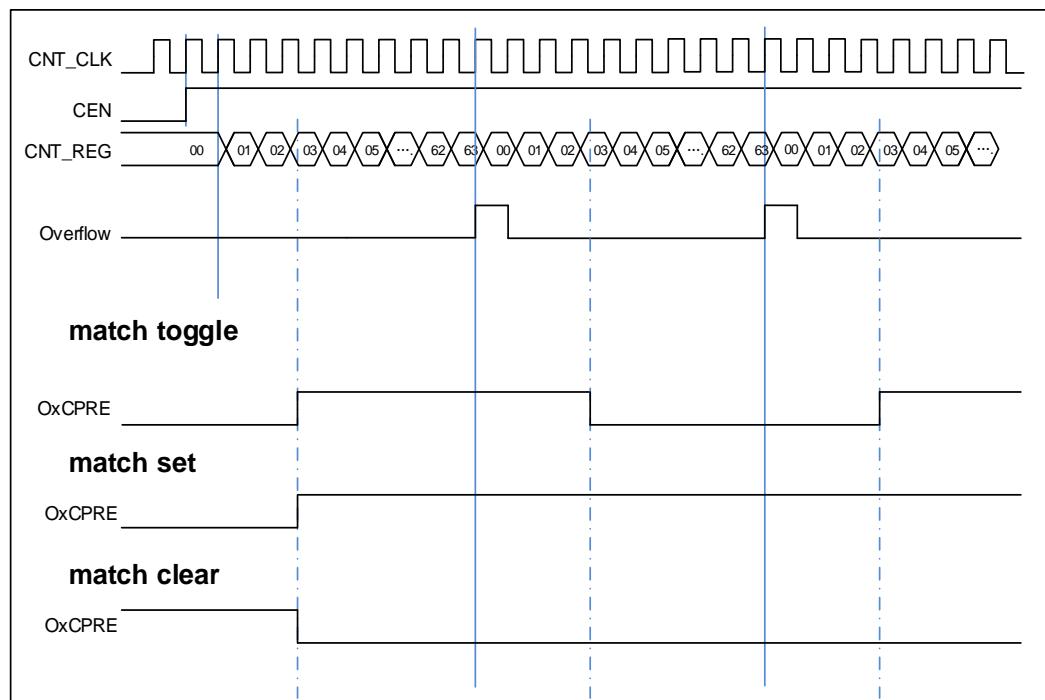
CHxVAL可以在运行时根据你所期望的波形而改变。

#### 第五步：设置CEN位使能定时器。

**图14-41. 三种输出比较模式**显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，

CHxVAL=0x3。

图 14-41. 三种输出比较模式



### 输出 PWM 功能

在 PWM 输出模式下 (PWM 模式 0 是配置 CHxCOMCTL 为 3'b110, PWM 模式 1 是配置 CHxCOMCTL 为 3'b111), 通道根据 TIMERx\_CAR 寄存器和 TIMERx\_CHxCV 寄存器的值, 输出 PWM 波形。

根据计数模式, 我们可以分为两种 PWM 波: EAPWM(边沿对齐 PWM)和CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx\_CAR 寄存器值决定, 占空比由 TIMERx\_CHxCV 寄存器值决定。图 14-42. EAPWM 时序图 显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由 (2\*TIMERx\_CAR 寄存器值) 决定, 占空比由 (2\*TIMERx\_CHxCV 寄存器值) 决定。图 14-43. CAPWM 时序图 显示了 CAPWM 的输出波形和中断。

在向上计数模式中, PWM 模式 0 下(CHxCOMCTL=3'b110), 如果 TIMERx\_CHxCV 寄存器的值大于 TIMERx\_CAR 寄存器的值, 通道输出一直为无效电平; PWM 模式 1 下(CHxCOMCTL=3'b111), 如果 TIMERx\_CHxCV 寄存器的值大于 TIMERx\_CAR 寄存器的值, 通道输出一直为有效电平。

图 14-42. EAPWM 时序图

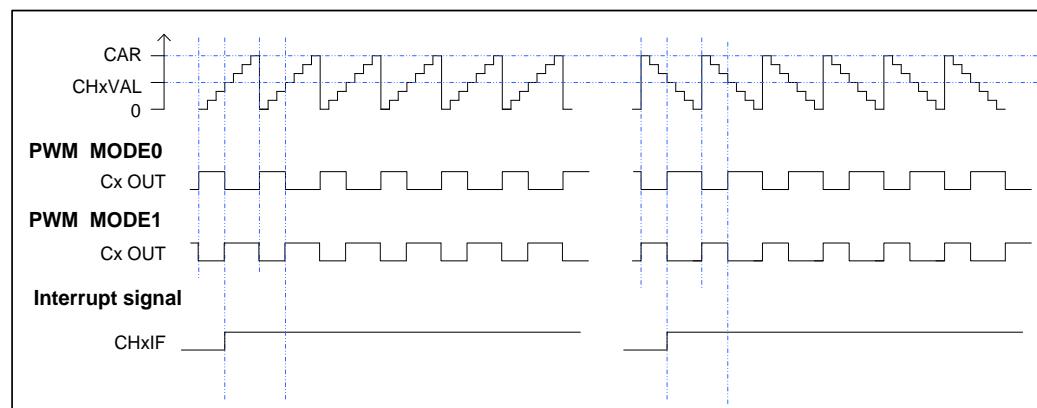
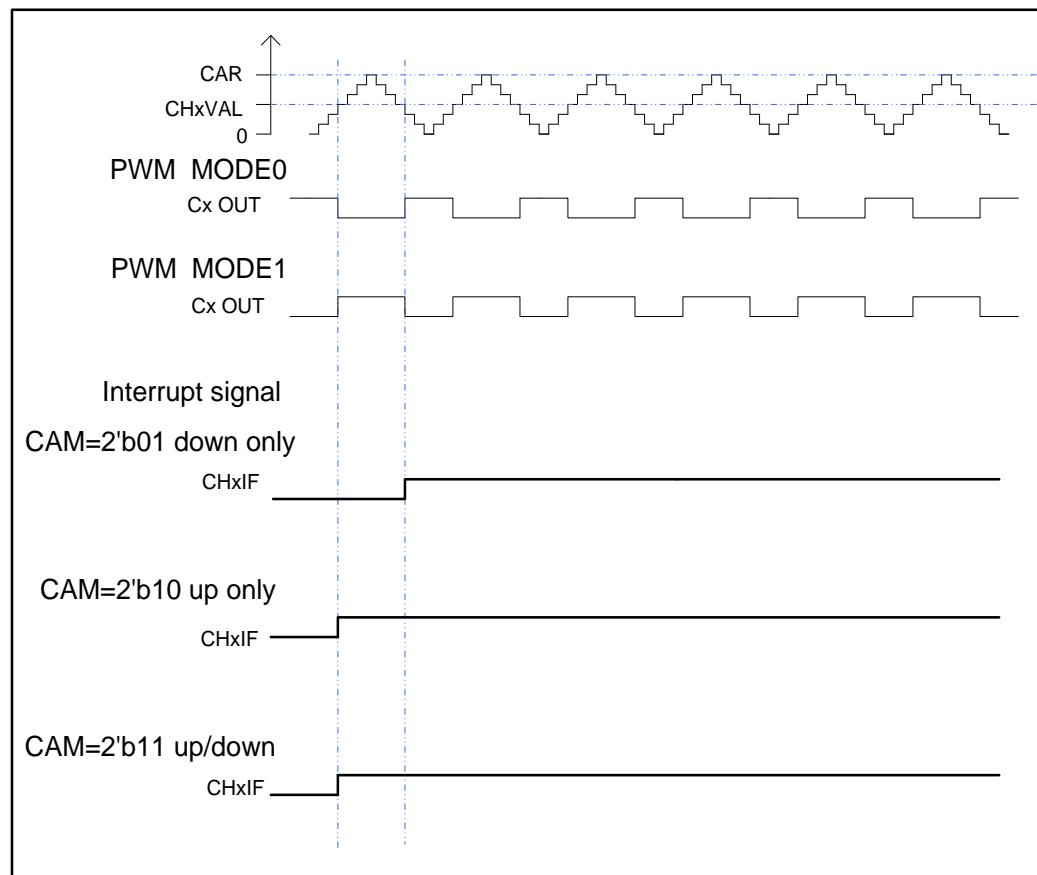


图 14-43. CAPWM 时序图



### 通道输出准备信号

根据 [图14-40. 通道输出比较原理 \(x=0,1,2,3\)](#) 所示，当TIMERx用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号（通道x输出准备信号）。设置 CHxCOMCTL 位可以定义 OxCPRE 信号类型。当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道x输出准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可

以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx\_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx\_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL=0x04或0x05可以实现OxCPRE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TIMERx\_CHxCV的值和计数器值之间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE信号才会回到有效电平状态。

### 正交译码器

参考[正交译码器](#)。

### 霍尔传感器接口功能

参考[霍尔传感器接口功能](#)。

### 主-从管理

参考[主-从管理](#)。

### 单脉冲模式

参考[单脉冲模式](#)。

### 定时器互连

参考[定时器互连](#)。

### 定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMERx\_DMACFG和TIMERx\_DMATB。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P（传输方向为从内存到外设）模式，PADDR（外设基地址）为TIMERx\_DMATB寄存器地址，DMA就会访问TIMERx\_DMATB寄存器。实际上，TIMERx\_DMATB寄存器只是一个缓冲，定时器会将TIMERx\_DMATB映射到一个内部寄存器，这个内部寄存器由TIMERx\_DMACFG寄存器中的DMATA来指定。如果TIMERx\_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器发送1个DMA请求就可以完成。如果TIMERx\_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMERx\_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4，DMATA+0x8，DMATA+0xC寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(DMATC+1)

次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

### 定时器调试模式

当Cortex®-M23内核停止，DBG\_CTL0寄存器中的TIMERx\_HOLD配置位被置1，定时器计数器停止。

### 14.2.5. TIMERx 寄存器 (x=2)

TIMER2基地址: 0x4000 0400

#### 控制寄存器 0 (TIMERx\_CTL0)

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				CKDIV[1:0]		ARSE		CAM[1:0]		DIR		SPM		UPS		UPDIS		CEN	
				rw		rw		rw		rw		rw		rw		rw			

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。</p> <p>00: <math>f_{DTS}=f_{CK\_TIMER}</math></p> <p>01: <math>f_{DTS}=f_{CK\_TIMER}/2</math></p> <p>10: <math>f_{DTS}=f_{CK\_TIMER}/4</math></p> <p>11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能TIMERx_CAR寄存器的影子寄存器</p> <p>1: 使能 TIMERx_CAR 寄存器的影子寄存器</p>
6:5	CAM[1:0]	<p>计数器对齐模式选择</p> <p>00: 无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向</p> <p>01: 中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)， 只有在向下计数时， CHxF位置1</p> <p>10: 中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)， 只有在向上计数时， CHxF位置1</p> <p>11: 中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)， 在向上和向下计数时， CHxF位都会置1</p> <p>当计数器使能以后，该位不能从 0x00 切换到非 0x00</p>
4	DIR	<p>方向</p> <p>0: 向上计数</p> <p>1: 向下计数</p>

当计数器配置为中央对齐计数模式或编码器模式时，该位只读。

3	<b>SPM</b>	单脉冲模式 0: 单脉冲模式禁能。更新事件发生后，计数器继续计数 1: 单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	<b>UPS</b>	更新请求源 软件配置该位，选择更新事件源。 0: 以下事件均会产生更新中断或DMA请求： <b>UPG位被置1</b> 计数器溢出/下溢 复位模式产生的更新 1: 下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	<b>UPDIS</b>	禁止更新。 该位用来使能或禁能更新事件的产生 0: 更新事件使能。更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： <b>UPG位被置1</b> 计数器溢出/下溢 复位模式产生的更新 1: 更新事件禁能。 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	<b>CEN</b>	计数器使能 0: 计数器禁能 1: 计数器使能 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。

### 控制寄存器 1 (TIMERx\_CTL1)

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								TIO5	MMC[2:0]		DMAS	保留.			

位/位域	名称	描述
------	----	----

31:8	保留	必须保持复位值
7	TIOS	通道0触发输入选择 0: 选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1: 选择 TIMERx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能 000: 当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001: 当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010: 当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和 UPS位决定 011: 当通道0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: 当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101: 当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110: 当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111: 当产生一次比较事件时，输出一个TRGO 信号，比较事件源来自 O3CPRE
3	DMAS	DMA 请求源选择 0: 当通道捕获/比较事件发生时，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
2:0	保留	必须保持复位值

### 从模式配置寄存器 (TIMERx\_SMCFG)

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
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]			MSM	TRGS[2:0]		OCRC	SMC[2:0]				
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	ETP	外部触发极性 该位指定ETI信号的极性

0: ETI高电平或上升沿有效

1: ETI低电平或下降沿有效

14

SMC1

SMC的一部分使能外部时钟模式1

在外部时钟模式1, 计数器由ETIFP信号上的任意有效边沿驱动

0: 外部时钟模式1禁能

1: 外部时钟模式1使能

当从模式配置为复位模式, 暂停模式和事件模式时, 定时器仍然可以工作在外部时钟模式1。但是TRGS必须不能为3'b111。

如果外部时钟模式0和外部时钟模式1同时被被配置, 外部时钟的输入是ETIF

注意: 外部时钟模式0使能在寄存器的SMC[2:0]位域。

13:12

ETPSC[1:0]

外部触发预分频

外部触发信号ETIFP的频率不能超过TIMER\_CK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ETIFP的频率。

00: 预分频禁能

01: 2分频

10: 4分频

11: 8分频

11:8

ETFC[3:0]

外部触发滤波控制

外部触发信号可以通过数字滤波器进行滤波, 该位域定义了数字滤波器的滤波能力。

数字滤波器的基本原理是: 以 $f_{SAMP}$ 频率连续采样外部触发信号, 同时记录采样相同电平的次数。当该次数达到配置的滤波能力时, 则认为是一个有效的电平信号。

EXTFC[3:0]	次数	$f_{SAMP}$
4'b0000		滤波器禁能.
4'b0001	2	$f_{TIMER\_CK}$
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS\_CK}/2$
4'b0101	8	
4'b0110	6	$f_{DTS\_CK}/4$
4'b0111	8	
4'b1000	6	$f_{DTS\_CK}/8$
4'b1001	8	
4'b1010	5	$f_{DTS\_CK}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS\_CK}/32$
4'b1110	6	
4'b1111	8	

7

MSM

主-从模式

该位用来同步被选择的定时器同时开始计数。TRGI用做启动事件, 通过TRGO, 定时器被连接在一起。

0: 主从模式禁能

**1: 主从模式使能**

6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步定时器的触发输入源 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP 从模式被使能后这些位不能改。
3	OCRC	OCPRE清除源选择 0: OCPRE_CLR_INT连接到OCPRE_CLR输入 1: OCPRE_CLR_INT连接到ETIF
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0。根据CI0FE0的电平，计数器在CI1FE1的边沿向上/下计数。 010: 编码器模式1。根据CI1FE1的电平，计数器在CI0FE0的边沿向上/下计数。 011: 编码器模式2。根据另一个信号的输入电平，计数器在CI0FE0和CI1FE1的边沿向上/下计数。 100: 复位模式。在选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。 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，则计数器时钟停止。 110: 事件模式。计数器在触发输入的上升沿启动。 111: 外部时钟模式0。选中的触发输入的上升沿驱动计数器。

**DMA 和中断使能寄存器 (TIMERx\_DMAINTEN)**

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	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CHOIE	UPIE
	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	TRGDEN	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	CH3DEN	通道3比较/捕获 DMA请求使能 0: 禁止通道3比较/捕获DMA请求 1: 使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获 DMA请求使能 0: 禁止通道2比较/捕获DMA请求 1: 使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获 DMA请求使能 0: 禁止通道1比较/捕获DMA请求 1: 使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获 DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	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	CH3IE	通道3比较/捕获中断使能 0: 禁止通道3中断 1: 使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0: 禁止通道2中断 1: 使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0: 禁止通道1中断 1: 使能通道1中断
1	CH0IE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能

0: 禁止更新中断

1: 使能更新中断

### 中断标志寄存器 (TIMERx\_INTF)

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		CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	CH3IF	CH2IF	CH1IF	CH0IF	UPIF	

rc\_w0      rc\_w0      rc\_w0      rc\_w0      .      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0      rc\_w0

位/位域	名称	描述
31:13	保留	必须保持复位值.
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时，在CH0IF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: 无捕获溢出中断发生 1: 发生了捕获溢出中断
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，此标志会置1，此位由软件清0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，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	CH3IF	通道3比较/捕获中断标志 参见CH0IF描述

---

3	CH2IF	通道2比较/捕获中断标志 参见 <b>CH0IF</b> 描述
2	CH1IF	通道1比较/捕获中断标志 参见 <b>CH0IF</b> 描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0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0中断发生 1：通道0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

### 软件事件产生寄存器 (**TIMERx\_SWEVG**)

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								TRGG	保留.	CH3G	CH2G	CH1G	CH0G	UPG	
w								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值.
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， <b>TIMERx_INTF</b> 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	CH3G	通道3捕获或比较事件发生 参见 <b>CH0G</b> 描述
3	CH2G	通道2捕获或比较事件发生

参见CH0G描述

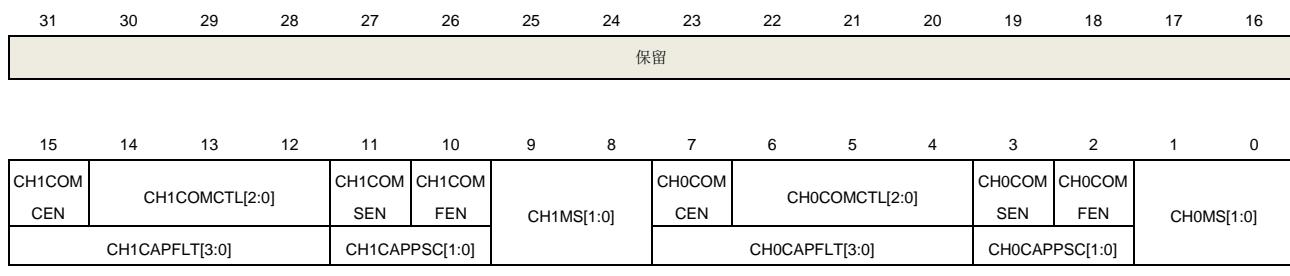
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)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

### 通道控制寄存器 0 (TIMERx\_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述

10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 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_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00: 通道 1 配置为输出 01: 通道 1 配置为输入, IS1 映射在 CI1FE1 上 10: 通道 1 配置为输入, IS1 映射在 CI0FE1 上 11: 通道 1 配置为输入, IS1 映射在 ITS 上 注意: 当 CH1MS[1:0]=11 时, 需要通过 TRGS 位 (位于 TIMERx_SMCFG 寄存器) 选择内部触发输入。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1, 当检测到 ETIFP 信号输入高电平时, O0CPRE 参考信号被清 0 0: 禁止通道 0 输出比较清零 1: 使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式, 而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外, O0CPRE 高电平有效, 而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000: 时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 翻转。 100: 强制为低。强制 O0CPRE 为低电平 101: 强制为高。强制 O0CPRE 为高电平 110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH0CV 时, O0CPRE 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_CH0CV 时, O0CPRE 为低电平, 否则为高电平。 111: PWM 模式 1。在向上计数时, 一旦计数器值小于 TIMERx_CH0CV 时, O0CPRE 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_CH0CV 时, O0CPRE 为高电平, 否则为低电平。 如果配置在 PWM 模式下, 只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时, O0CPRE 电平才改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1, TIMERx_CH0CV 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 0 输出/比较影子寄存器 1: 使能通道 0 输出/比较影子寄存器

仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式

2	CH0COMPEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较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，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速。 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，ISO 映射在 CI0FE0 上 10：通道 0 配置为输入，ISO 映射在 CI1FE0 上 11：通道 0 配置为输入，ISO 映射在 ITS 上 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

#### 输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 fSAMP 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。

滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	fSAMP
4'b0000		无滤波器
4'b0001	2	fCK_TIMER
4'b0010	4	
4'b0011	8	
4'b0100	6	fDTS/2
4'b0101	8	
4'b0110	6	fDTS/4
4'b0111	8	

4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	$f_{DTS}/32$
4'b1101	5	
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 **TIMERx\_CHCTL2** 寄存器中的 **CH0EN =0** 时，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 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

### 通道控制寄存器 1 (**TIMERx\_CHCTL1**)

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
CH3COM CEN	CH3COMCTL[2:0]	CH3COM SEN	CH3COM FEN	CH3MS[1:0]	CH2COM CEN	CH2COMCTL[2:0]	CH2COM SEN	CH2COM FEN	CH2MS[1:0]	CH2CAPFLT[3:0]	CH2CAPPSC[1:0]	CH2CAPFLT[3:0]	CH2CAPPSC[1:0]	CH2MS[1:0]	CH2MS[1:0]

rw rw rw rw rw rw rw

#### 输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述
11	CH3COMSEN	通道 3 输出比较影子寄存器使能

		参见 CH0COMSEN 描述
10	CH3COMFEN	通道 3 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 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_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 00: 通道 3 配置为输出 01: 通道 3 配置为输入, IS3 映射在 CI3FE3 上 10: 通道 3 配置为输入, IS3 映射在 CI2FE3 上 11: 通道 3 配置为输入, IS3 映射在 ITS 上 注意: 当 CH3MS[1:0]=11 时, 需要通过 TRGS 位 (位于 TIMERx_SMCFG 寄存器) 选择内部触发输入。
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1, 当检测到 ETIFFP 输入高电平时, O2CPRE 参考信号被清 0 0: 使能通道 2 输出比较清零 1: 禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式, 而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外, O2CPRE 高电平有效, 而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000: 时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时, 强制 O2CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时, 强制 O2CPRE 为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时, 强制 O2CPRE 翻转。 100: 强制为低。强制 O2CPRE 为低电平 101: 强制为高。强制 O2CPRE 为高电平 110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV 时, O2CPRE 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_CH2CV 时, O2CPRE 为低电平, 否则为高电平。 111: PWM 模式 1。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV 时, O2CPRE 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_CH2CV 时, O2CPRE 为高电平, 否则为低电平。 如果配置在 PWM 模式下, 只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时, O2CPRE 电平才改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1, TIMERx_CH2CV 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 2 输出/比较影子寄存器

1: 使能通道 2 输出/比较影子寄存器

仅在单脉冲模式下(**SPM =1**), 可以在未确认影子寄存器情况下使用 PWM 模式

2	<b>CH2COMFEN</b>	通道 2 输出比较快速使能 当该位为 1 时, 如果通道配置为 PWM0 模式或者 PWM1 模式, 会加快捕获/比较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, CH2_O 被设置为比较电平而与比较结果无关。 0: 禁止通道 2 输出比较快速. 1: 使能通道 2 输出比较快速。
1:0	<b>CH2MS[1:0]</b>	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的 CH2EN 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, IS2 映射在 CI2FE2 上 10: 通道 2 配置为输入, IS2 映射在 CI3FE2 上 11: 通道 2 配置为输入, IS2 映射在 ITS 上. 注意: 当 CH2MS[1:0]=11 时, 需要通过 TRGS 位 (位于 TIMERx_SMCFG 寄存器) 选择内部触发输入。

#### 输入捕获模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	<b>CH3CAPFLT[3:0]</b>	通道 3 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	<b>CH3CAPPSC[1:0]</b>	通道 3 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	<b>CH3MS[1:0]</b>	通道 3 模式选择 与输出模式相同
7:4	<b>CH2CAPFLT[3:0]</b>	通道 2 输入捕获滤波控制 CI2 输入信号可以通过数字滤波器进行滤波, 该位域配置滤波参数。 数字滤波器的基本原理: 根据 fSAMP 对 CI2 输入信号进行连续采样, 并记录信号相同电平的次数。达到该位配置的滤波参数后, 认为是有效电平。

滤波器参数配置如下:

<b>CH2CAPFLT [3:0]</b>	<b>采样次数</b>	<b>fSAMP</b>
4'b0000	无滤波器	
4'b0001	2	$f_{CK\_TIMER}$
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	

4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	$f_{DTS}/32$
4'b1101	5	
4'b1110	6	$f_{DTS}/32$
4'b1111	8	

3:2            CH2CAPPSC[1:0]            通道 2 输入捕获预分频器

这 2 位定义了通道 2 输入的预分频系数。当 **TIMERx\_CHCTL2** 寄存器中的 **CH2EN =0** 时，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            CH2MS[1:0]            通道 2 模式选择

与输出比较模式相同

### 通道控制寄存器 2 (**TIMERx\_CHCTL2**)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH3NP	通道 3 互补输出极性 参考 <b>CH0NP</b> 描述
14	保留	必须保持复位值
13	CH3P	通道 3 极性 参考 <b>CH0P</b> 描述

12	CH3EN	通道 3 使能 参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
10	保留	必须保持复位值
9	CH2P	通道 2 极性 参考 CH0P 描述
8	CH2EN	通道 2 使能 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位保持 0。 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道 0 高电平为有效电平 1：通道 0 低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]: 把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]: 把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]: 保留。 [CH0NP==1, CH0P==1]: 把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为

输入模式时，将此位置 1 使能通道 0 上的捕获事件。

- 0: 禁止通道 0
- 1: 使能通道 0

### **计数器寄存器 (TIMERx\_CNT)**

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
保留															
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	保留	必须保持复位值
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。

### **预分频寄存器 (TIMERx\_PSC)**

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
PSC[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (**TIMERx\_CAR**)

地址偏移: 0x2C

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	<p>计数器自动重载值</p> <p>这些位定义了计数器的自动重载值。</p> <p><b>注意:</b> 在定时器被配置为输入捕获模式时, 该寄存器需要被配置成一个大于用户期望值的非0值(例如0xFFFF)。</p>

### 通道 0 捕获/比较值寄存器 (**TIMERx\_CH0CV**)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	<p>通道 0 的捕获或比较值</p> <p>当通道 0 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 0 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>

### 通道 1 捕获/比较值寄存器 (**TIMERx\_CH1CV**)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[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。

### 通道 2 捕获/比较值寄存器 (**TIMERx\_CH2CV**)

地址偏移: 0x3C

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
CH2VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH2VAL[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。

### 通道 3 捕获/比较值寄存器 (TIMERx\_CH3CV)

地址偏移: 0x40

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
CH3VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[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。

### DMA 配置寄存器 (TIMERx\_DMACFG)

地址偏移: 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
保留				DMATC[4:0]				保留				DMATA [4:0]			
rw														rw	

位/位域	名称	描述
31:13	保留	必须保持复位值。
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMATB寄存器的数量n, n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址

该位域定义了 DMA 访问 TIMERx\_DMATB 寄存器的第一个地址。当通过 TIMERx\_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx\_DMATB 时，将访问起始地址+0x4。

### DMA 发送缓冲区寄存器 (TIMERx\_DMATB)

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

### 配置寄存器 (TIMERx\_CFG)

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															

rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	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: 无影响

0

保留

必须保持复位值

## 14.3. 通用定时器 L2 (TIMERx, x=13)

### 14.3.1. 简介

通用定时器L2 (TIMERx, x=13)是单通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。通用定时器L2含有一个16位无符号计数器。

通用定时器L2是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

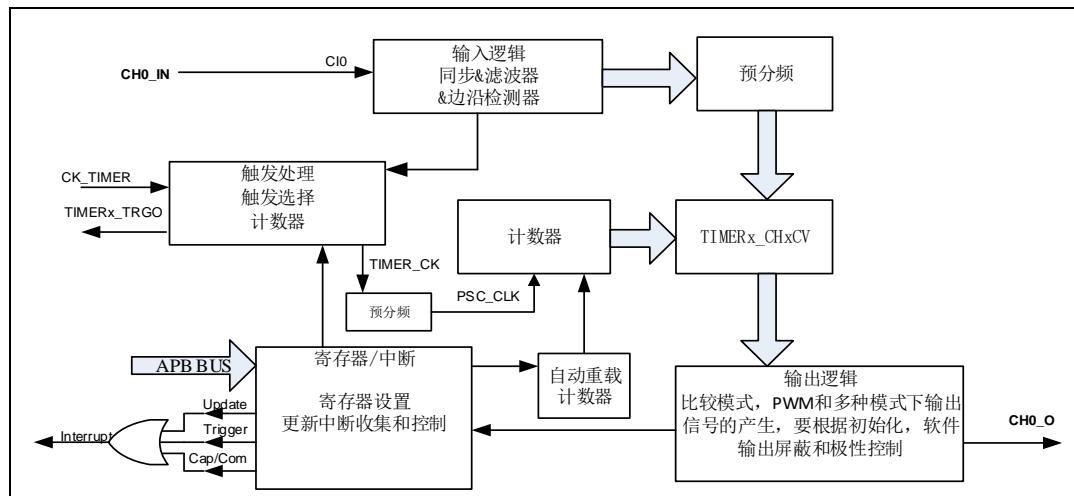
### 14.3.2. 主要特征

- 总通道数：1；
- 计数器宽度：16位；
- 时钟源：内部时钟；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式；
- 自动重装载功能；
- 中断输出：更新事件，比较/捕获事件。

### 14.3.3. 结构框图

[图14-44. 通用定时器L2结构框图](#)提供了通用定时器L2的内部配置细节。

图 14-44. 通用定时器 L2 结构框图



### 14.3.4. 功能说明

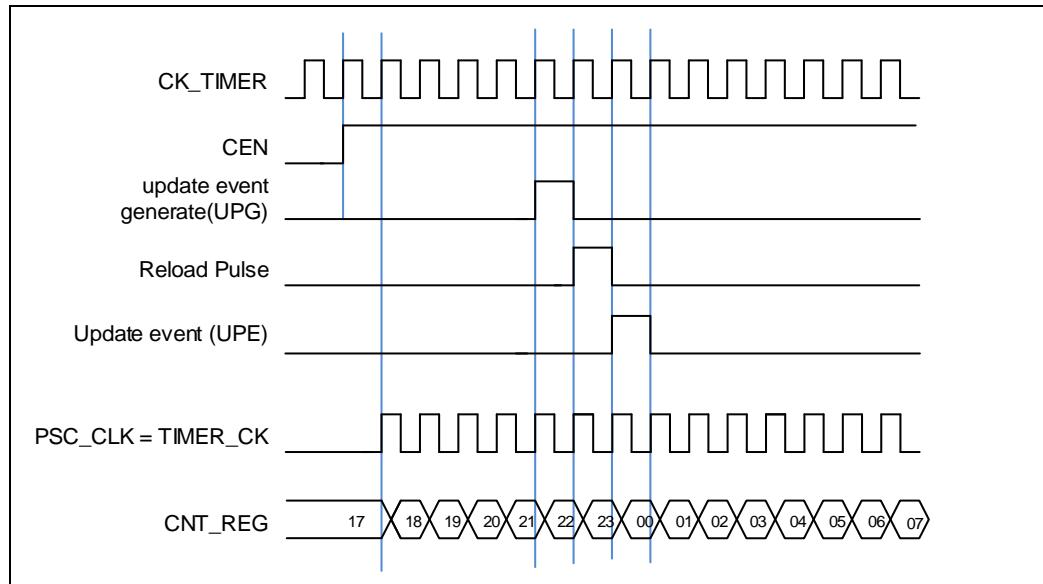
#### 时钟源配置

通用定时器L2可以是内部时钟源CK\_TIMER。

- 定时器时钟TIMER\_CK连接到RCU模块的CK\_TIMER。

通用定时器L2仅有一个时钟源CK\_TIMER，用来驱动计数器预分频器。当CEN置位，CK\_TIMER经过预分频器（预分频值由TIMERx\_PSC寄存器确定）产生PSC\_CLK。

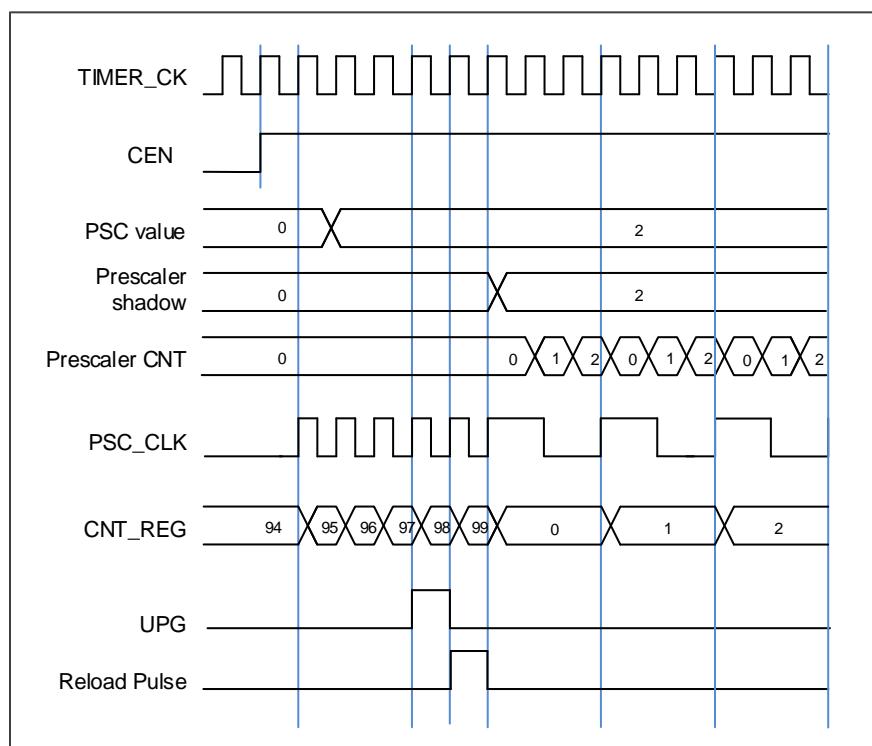
图 14-45. 内部时钟分频为 1 时，计数器的时序图



### 时钟预分频器

预分频器可以将定时器的时钟 (TIMER\_CK) 频率按1到65536之间的任意值分频，分频后的时钟PSC\_CLK驱动计数器计数。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 14-46. 当 PSC 数值从 0 变到 2 时，计数器的时序图



### 向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在`TIMERx_CAR`寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，`TIMERx_CTL0`寄存器中的计数方向控制位`DIR`应该被设置成0。

当通过`TIMERx_SWEVG`寄存器的`UPG`位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果`TIMERx_CTL0`寄存器的`UPDIS`置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

[图 14-47. 向上计数时序图，PSC=0/2](#) 和 [图 14-48. 向上计数时序图，在运行时改变 TIMERx CAR 寄存器的值](#)给出了一些例子，当 `TIMERx_CAR=0x99` 时，计数器在不同预分频因子下的行为。

图 14-47. 向上计数时序图, PSC=0/2

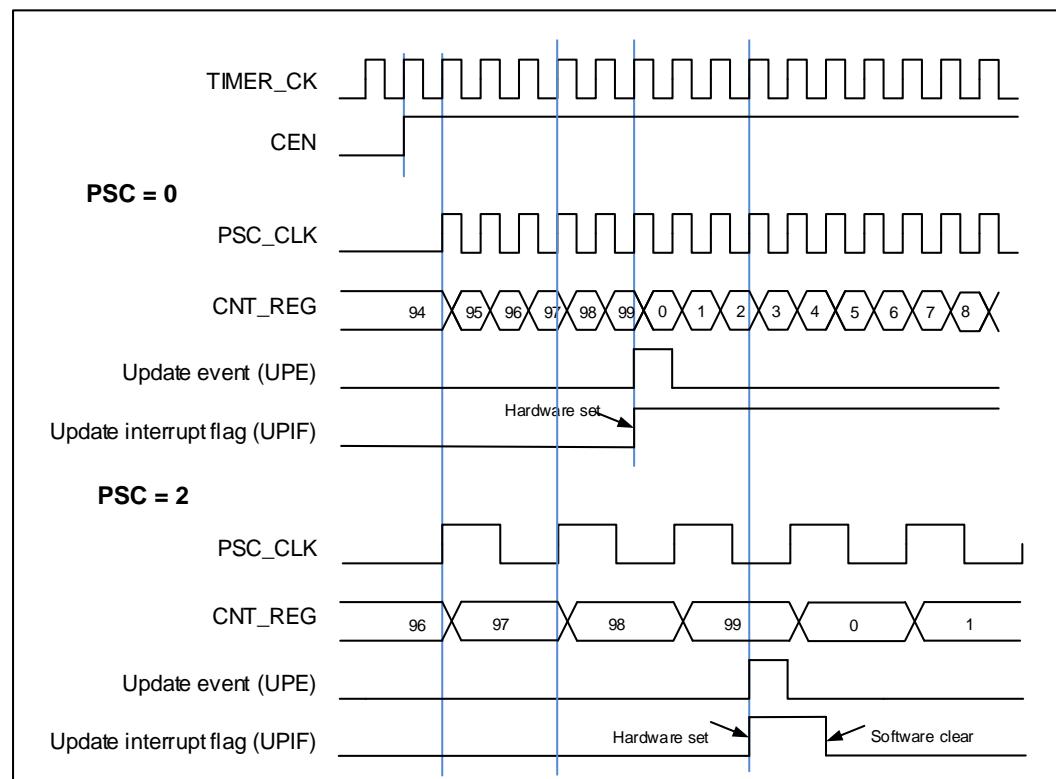
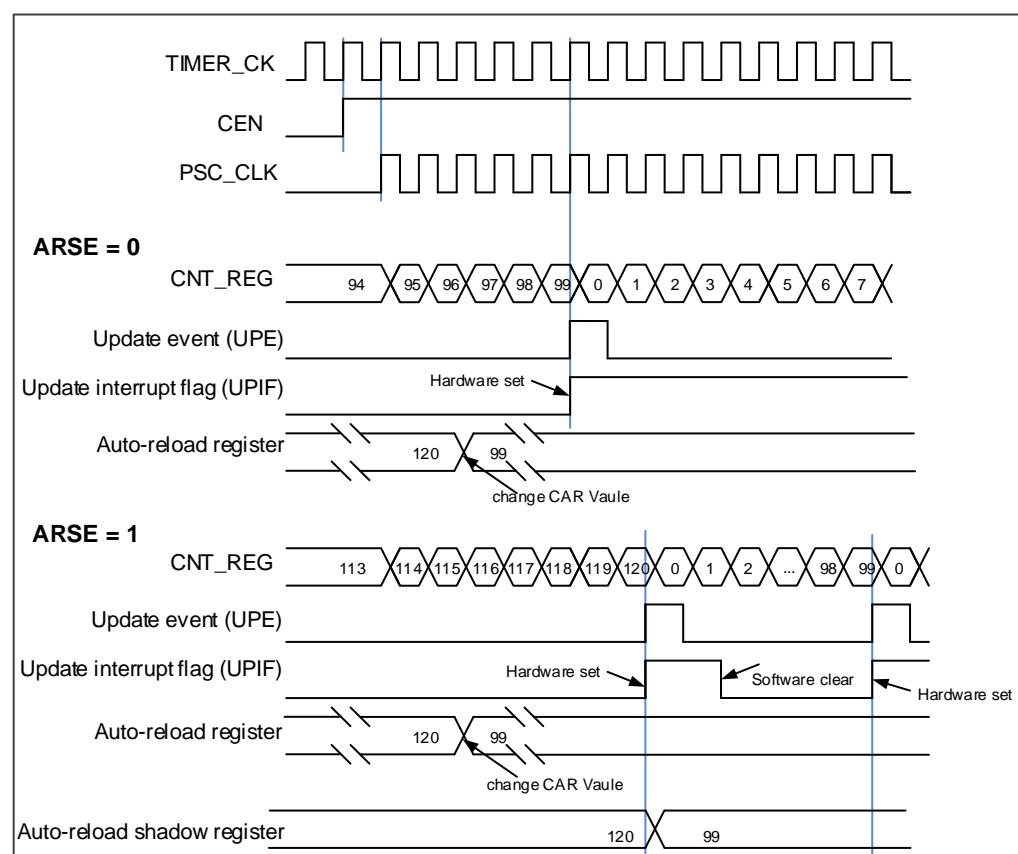


图 14-48. 向上计数时序图, 在运行时改变 TIMERx\_CAR 寄存器的值



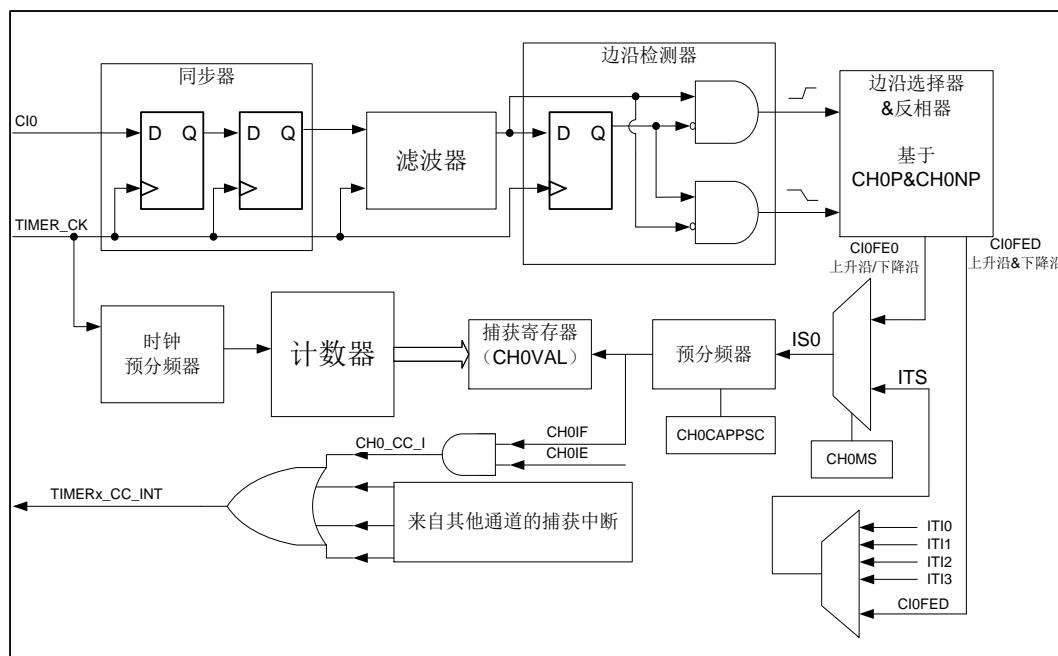
## 输入捕获和输出比较通道

通用定时器L2只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道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。

### ■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期等。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。如果在输入引脚上出现被选择的边沿，**TIMERx\_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，如果**CHxIE = 1**则产生通道中断。

**图 14-49. 通道输入捕获原理**



通道输入信号**CIx**先被**TIMER\_CK**信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置**CHxP**选择使用上升沿或者下降沿。配置**CHxMS**，可以选择其他通道的输入信号，内部触发信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，**TIMERx\_CHxCV**存储计数器的值。

配置步骤如下：

**第一步：滤波器配置 (TIMERx\_CHCTL0寄存器中CHxCAPFLT):**

根据输入信号和请求信号的质量，配置相应的**CHxCAPFLT**。

**第二步：边沿选择 (TIMERx\_CHCTL2寄存器中CHxP/CHxNP):**

配置**CHxP/CHxNP**选择上升沿，下降沿或者双边沿（上升沿和下降沿）。

**第三步：捕获源选择 (TIMERx\_CHCTL0寄存器中CHxMS):**

一旦通过配置**CHxMS**选择输入捕获源，必须确保通道配置在输入模式 (**CHxMS!=0x0**)，而且**TIMERx\_CHxCV**寄存器不能再被写。

**第四步：中断使能 (TIMERx\_DMAINTEN寄存器中CHxIE):**

使能相应中断，可以获得中断。

**第五步：捕获使能 (TIMERx\_CHCTL2寄存器中CHxEN)。**

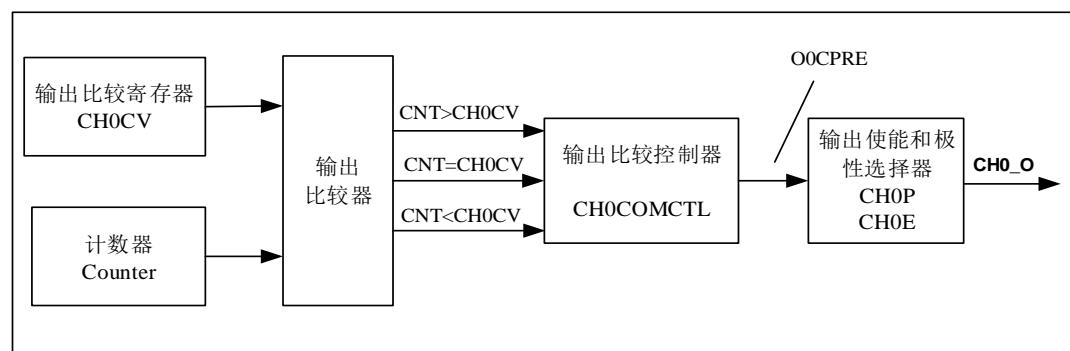
**结果：**当期望的输入信号发生时，TIMERx\_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx\_DMAINTEN寄存器中CHxIE的配置，相应的中断会被提出。

**直接产生：**软件设置CHxG位，会直接产生中断请求。

输入捕获模式也可用来测量TIMERx\_CHx引脚上信号的脉冲周期。例如，一个PWM波连接到CI0。配置TIMERx\_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0，同时设置上升沿捕获。。TIMERX\_CH0CV寄存器测量PWM的周期值。

#### ■ 通道输出比较功能

**图 14-50. 通道输出比较原理**



**图14-50. 通道输出比较原理**给出了输出比较的原理电路。通道输出信号CHx\_O与OxCPRE信号（详情请见[通道输出准备信号](#)）的关系描述：OxCPRE信号高电平有效，CHx\_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关（具体情况请见TIMERx\_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx\_O高电平有效，与OxCPRE输出极性相同）、CHxE=1（CHx\_O输出使能）时：

若OxCPRE输出有效（高）电平，则CHx\_O输出有效（高）电平；

若OxCPRE输出无效（低）电平，则CHx\_O输出无效（低）电平。

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的TIMERx\_CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与TIMERx\_CHxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxODE=1则会产生DMA请求。

配置步骤如下：

**第一步：时钟配置：**

配置定时器时钟源，预分频器等。

**第二步：比较模式配置：**

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

**第三步：**通过CHxIE位配置中断使能。

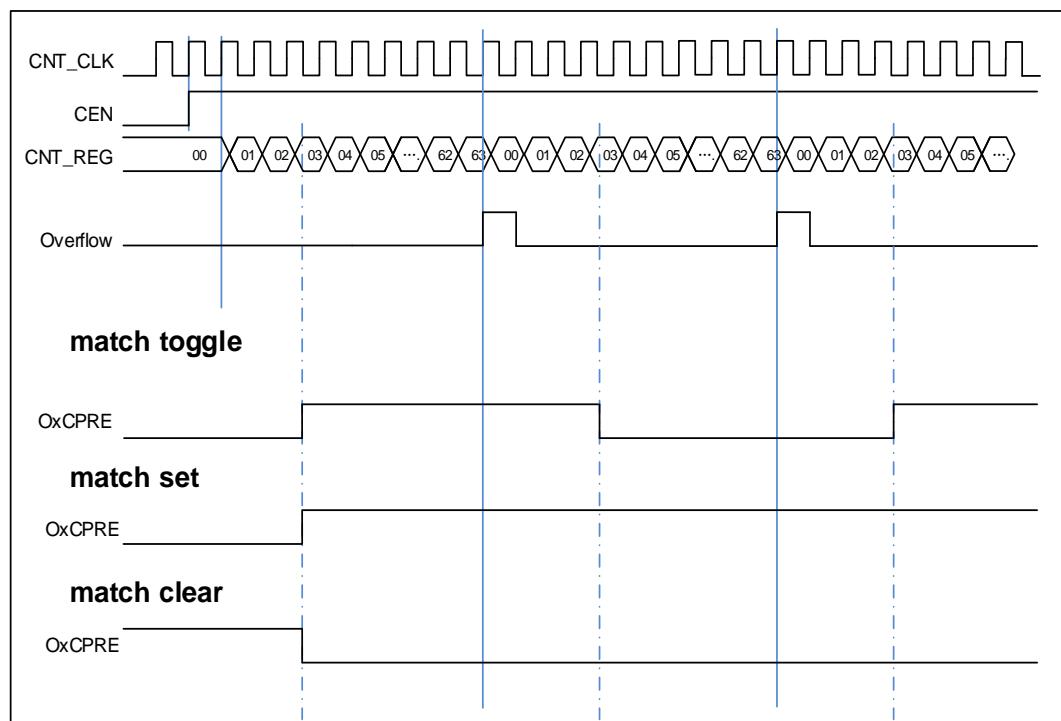
**第四步：**通过TIMERx\_CAR寄存器和TIMERx\_CHxCV寄存器配置输出比较时基：

TIMERx\_CHxCV可以在运行时根据你所期望的波形而改变。

**第五步：**设置CEN位使能定时器。

[图14-51. 三种输出比较模式](#)显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 14-51. 三种输出比较模式



### 输出 PWM 功能

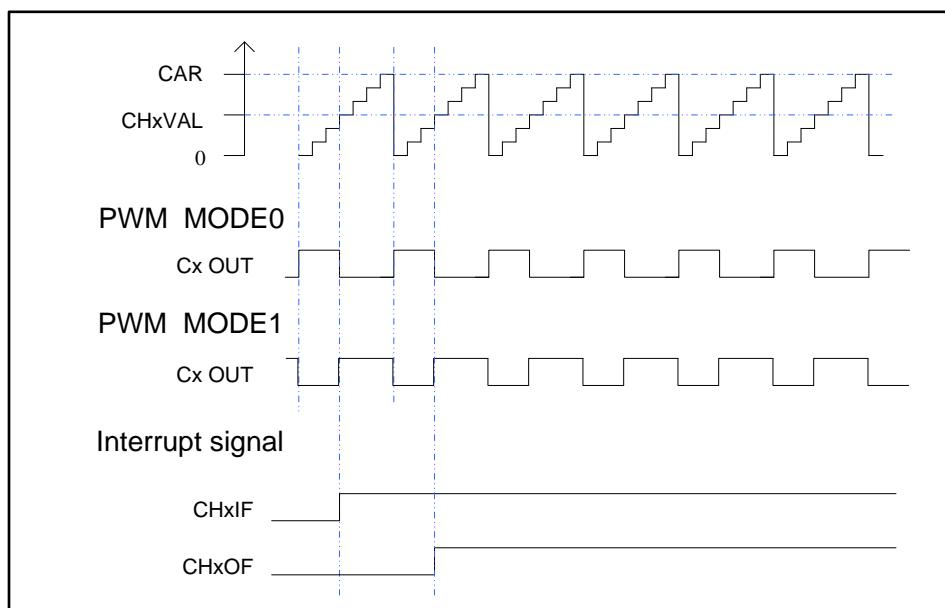
在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx\_CAR 寄存器和 TIMERx\_CHxCV 寄存器的值，输出 PWM 波形。

PWM 的周期由 TIMERx\_CAR 寄存器值决定，占空比由 TIMERx\_CHxCV 寄存器值决定。[图 14-52. PWM 时序图](#)显示了 EAPWM 的输出波形和中断。

在 PWM0 模式下（CHxCOMCTL==3'b110），如果 TIMERx\_CHxCV 寄存器的值大于 TIMERx\_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下（CHxCOMCTL==3'b110），如果 TIMERx\_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 14-52. PWM 时序图



### 通道输出准备信号

根据[图14-50. 通道输出比较原理](#)所示，当TIMERx用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号OxCPRE信号(通道x输出准备信号)。设置CHxCOMCTL位可以定义OxCPRE信号类型。当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x输出准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx\_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx\_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL=0x04或0x05可以实现OxCPRE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TIMERx\_CHxCV的值和计数器值之间的比较结果。

### 定时器调试模式

当Cortex®-M23内核停止，DBG\_CTL0寄存器中的TIMERx\_HOLD配置位被置1，定时器计数器停止。

### 14.3.5. TIMERx 寄存器(x=13)

TIMER13基地址: 0x4000 2000

#### 控制寄存器 0 (TIMERx\_CTL0)

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					CKDIV[1:0]	ARSE		保留		UPS	UPDIS	CEN			

rw                    rw                    rw                    rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。</p> <p>00: <math>f_{DTS}=f_{CK\_TIMER}</math></p> <p>01: <math>f_{DTS}=f_{CK\_TIMER} /2</math></p> <p>10: <math>f_{DTS}=f_{CK\_TIMER} /4</math></p> <p>11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能 TIMERx_CAR 寄存器的影子寄存器。</p> <p>1: 使能TIMERx_CAR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>软件配置该位，选择更新事件源.</p> <p>0: 以下事件均会产生更新中断或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>复位模式产生的更新</li> </ul> <p>1: 下列事件会产生更新中断或DMA请求:</p> <ul style="list-style-type: none"> <li>计数器溢出/下溢</li> </ul>
1	UPDIS	<p>禁止更新.</p> <p>该位用来使能或禁能更新事件的产生</p> <p>0: 更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均</p>

会产生更新事件:

UPG位被置1

计数器溢出/下溢

复位模式产生的更新

1: 更新事件禁能。

注意: 当该位被置1时, UPG位被置1或者复位模式不会产生更新事件, 但是计数器和预分频器被重新初始化

0	CEN	计数器使能
0:	计数器禁能	
1:	计数器使能	

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。

### DMA 和中断使能寄存器 (TIMERx\_DMAINTEN)

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
														CH0IE	UPIE

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CH0IE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

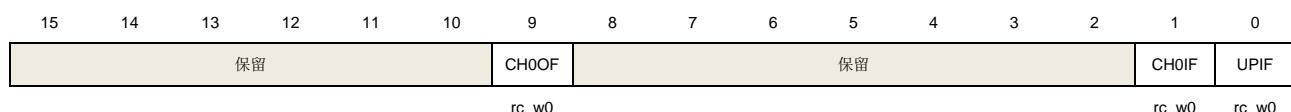
### 中断标志寄存器 (TIMERx\_INTF)

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															



位/位域	名称	描述
31:10	保留	必须保持复位值
9	CH0OF	<p>通道0捕获溢出标志</p> <p>当通道0被配置为输入模式时，在CH0IF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。</p> <p>0：无捕获溢出中断发生 1：发生了捕获溢出中断</p>
8:2	保留	必须保持复位值
1	CH0IF	<p>通道0比较/捕获中断标志</p> <p>此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。</p> <p>0：无通道0中断发生 1：通道0中断发生</p>
0	UPIF	<p>更新中断标志</p> <p>此位在任何更新事件发生时由硬件置1，软件清0。</p> <p>0：无更新中断发生 1：发生更新中断</p>

### 软件事件产生寄存器 (TIMERx\_SWEVG)

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
31:2	保留	必须保持复位值。
1	CH0G	<p>通道0捕获或比较事件发生</p> <p>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF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。</p>

此外，如果通道0配置为输入模式，计数器的当前值被TIMERx\_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。

- 0: 不产生通道0捕获或比较事件
- 1: 发生通道0捕获或比较事件

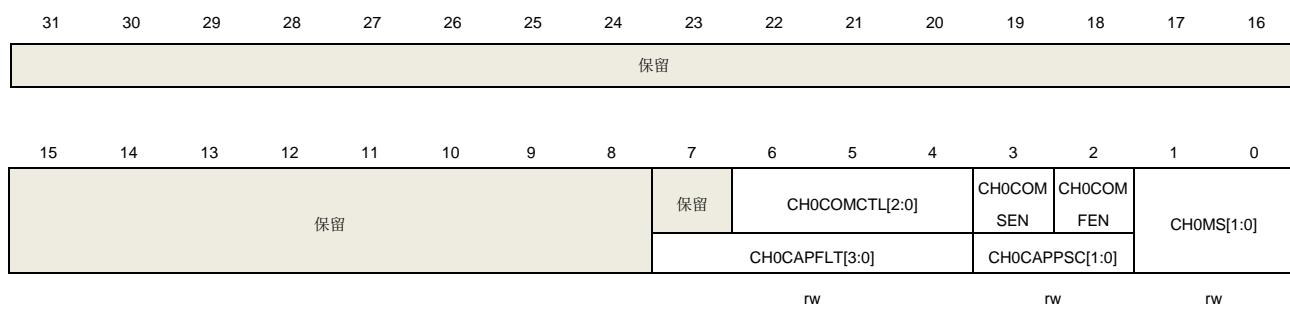
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。 0: 无更新事件产生 1: 产生更新事件
---	-----	--

### 通道控制寄存器 0 (TIMERx\_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



#### 输出比较模式:

位/位域	名称	描述
31:7	保留	必须保持复位值.
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000: 时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100: 强制为低。强制 O0CPRE 为低电平 101: 强制为高。强制 O0CPRE 为高电平
3:0	CH0MS[1:0]	

**110:** PWM 模式 0。在向上计数时，一旦计数器值小于 **TIMERx\_CH0CV** 时，O0CPRE 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\_CH0CV** 时，O0CPRE 为低电平，否则为高电平。

**111:** PWM 模式 1。在向上计数时，一旦计数器值小于 **TIMERx\_CH0CV** 时，O0CPRE 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\_CH0CV** 时，O0CPRE 为高电平，否则为低电平。

如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。

3	<b>CH0COMSEN</b>	<p>通道 0 输出比较影子寄存器使能</p> <p>当此位被置 1，<b>TIMERx_CH0CV</b> 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 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器</p> <p>仅在单脉冲模式下(<b>SPM =1</b>)，可以在未确认影子寄存器的情况下使用 PWM 模式</p>
2	<b>CH0COMFEN</b>	<p>通道 0 输出比较快速使能</p> <p>当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较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，<b>CH0_O</b> 被设置为比较电平而与比较结果无关。</p> <p>0：禁止通道 0 输出比较快速。 1：使能通道 0 输出比较快速。</p>
1:0	<b>CH0MS[1:0]</b>	<p>通道 0 I/O 模式选择</p> <p>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (<b>TIMERx_CHCTL2</b> 寄存器的 <b>CH0EN</b> 位被清 0) 时这些位才可写。</p> <p>00：通道 0 配置为输出 01：通道 0 配置为输入，<b>ISO</b> 映射在 <b>CIOFE0</b> 上 10：保留 11：通道 0 配置为输入，<b>ISO</b> 映射在 <b>ITS</b> 上。</p> <p>注意：当 <b>CH0MS[1:0]=11</b> 时，需要通过 <b>TRGS</b> 位（位于 <b>TIMERx_SMCFG</b> 寄存器）选择内部触发输入。</p>

### 输入捕获模式：

位/位域	名称	描述									
31:8	保留	必须保持复位值。									
7:4	<b>CH0CAPFLT[3:0]</b>	<p>通道 0 输入捕获滤波控制</p> <p><b>CIO</b> 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。</p> <p>数字滤波器的基本原理：根据 <b>fSAMP</b> 对 <b>CIO</b> 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。</p> <p>滤波器参数配置如下：</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; width: fit-content;"> <thead> <tr style="background-color: #e0e0e0;"> <th style="text-align: center;"><b>CH0CAPFLT [3:0]</b></th><th style="text-align: center;"><b>采样次数</b></th><th style="text-align: center;"><b>fSAMP</b></th></tr> </thead> <tbody> <tr> <td style="text-align: center;">4'b0000</td><td style="text-align: center;">无滤波器</td><td></td></tr> <tr> <td style="text-align: center;">4'b0001</td><td style="text-align: center;">2</td><td style="text-align: center;"><b>fCK_TIMER</b></td></tr> </tbody> </table>	<b>CH0CAPFLT [3:0]</b>	<b>采样次数</b>	<b>fSAMP</b>	4'b0000	无滤波器		4'b0001	2	<b>fCK_TIMER</b>
<b>CH0CAPFLT [3:0]</b>	<b>采样次数</b>	<b>fSAMP</b>									
4'b0000	无滤波器										
4'b0001	2	<b>fCK_TIMER</b>									

4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0]

通道0输入捕获预分频器

这2位定义了通道0输入的预分频系数。当TIMERx\_CHCTL2寄存器中的CH0EN =0时，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 CH0MS[1:0]

通道0模式选择

与输出比较模式相同

## 通道控制寄存器 2 (TIMERx\_CHCTL2)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留..										CH0NP	保留	CH0P	CH0EN		
rw rw rw															

位/位域	名称	描述
31:4	保留	必须保持复位值.
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0: 通道0互补输出高电平为有效电平

**1:** 通道0互补输出低电平为有效电平

当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。

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

**2** 保留 必须保持复位值.

**1** CH0P 通道0极性  
当通道0配置为输出模式时，此位定义了输出信号极性。

**0:** 通道0高电平为有效电平

**1:** 通道0低电平为有效电平

当通道0配置为输入模式时，此位定义了CI0信号极性

[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性

[CH0NP==0, CH0P==0]: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

[CH0NP==0, CH0P==1]: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。

[CH0NP==1, CH0P==0]: 保留。

[CH0NP==1, CH0P==1]: 把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

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

**0** CH0EN 通道0捕获/比较使能

当通道0配置为输出模式时，将此位置1使能CH0\_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。

**0:** 禁止通道0

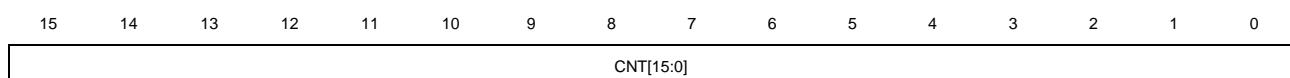
**1:** 使能通道0

### 计数器寄存器 (TIMERx\_CNT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



rw

位/位域	名称	描述
31:16	保留	必须保持复位值
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。

### 预分频寄存器 (TIMERx\_PSC)

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
PSC[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (TIMERx\_CAR)

地址偏移: 0x2C

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
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。 <b>注意:</b> 在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

### 通道 0 捕获/比较值寄存器 (TIMERx\_CH0CV)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 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 0 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。

### 通道输入重映射寄存器(TIMERx\_IRMP)

地址偏移: 0x50

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
CI0_RMP[1:0]															

rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CI0_RMP[1:0]	通道 0 输入重映射 00:通道 0 输入连接到 GPIO(TIMER13_CH0) 01: 通道 0 输入连接到 RTCCLK 10: 通道 0 输入连接到 HXTAL/32 clock 11: 通道 0 输入连接到 CKOUTSEL

### 配置寄存器 (TIMERx\_CFG)

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															
RW															

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	<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: 无影响</p>
0	保留	必须保持复位值

## 14.4. 通用定时器 L3 (TIMERx,x=14)

### 14.4.1. 简介

通用定时器L3 (TIMER14) 是两通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。通用定时器L3含有一个16位无符号计数器。

通用定时器L3是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

通用定时器L3包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

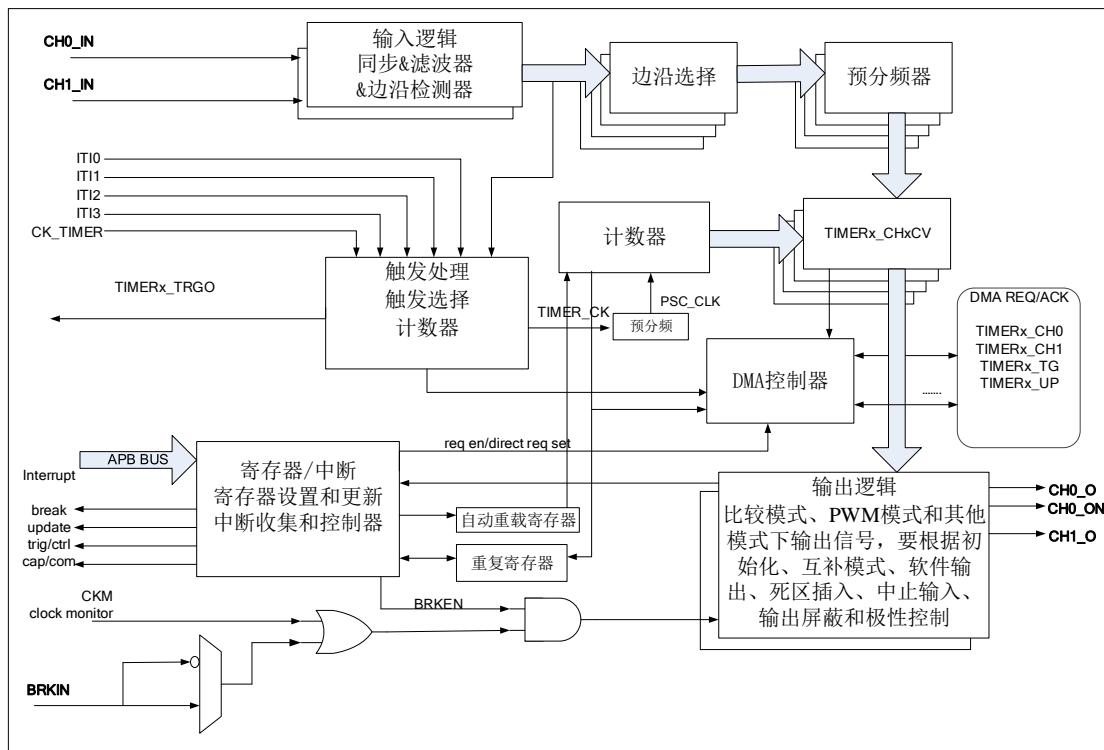
### 14.4.2. 主要特性

- 总通道数: 2;
- 计数器宽度: 16位;
- 时钟源可选: 内部时钟, 内部触发, 外部输入;
- 计数模式: 向上计数;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式, 单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和DMA请求: 更新事件, 比较/捕获事件, 换相事件和中止事件;
- 多个定时器的菊花链使得一个定时器可以同时启动多个定时器;
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数;
- 定时器主-从管理。

### 14.4.3. 结构框图

[图14-53. 通用定时器L3结构框图](#)提供了通用定时器L3的内部配置细节。

图 14-53. 通用定时器 L3 结构框图



#### 14.4.4. 功能描述

##### 时钟源配置

通用定时器L0可以是内部时钟源CK\_TIMER，或者是由SMC(TIMERx\_SMCFG寄存器位[2:0])位确定的时钟源。

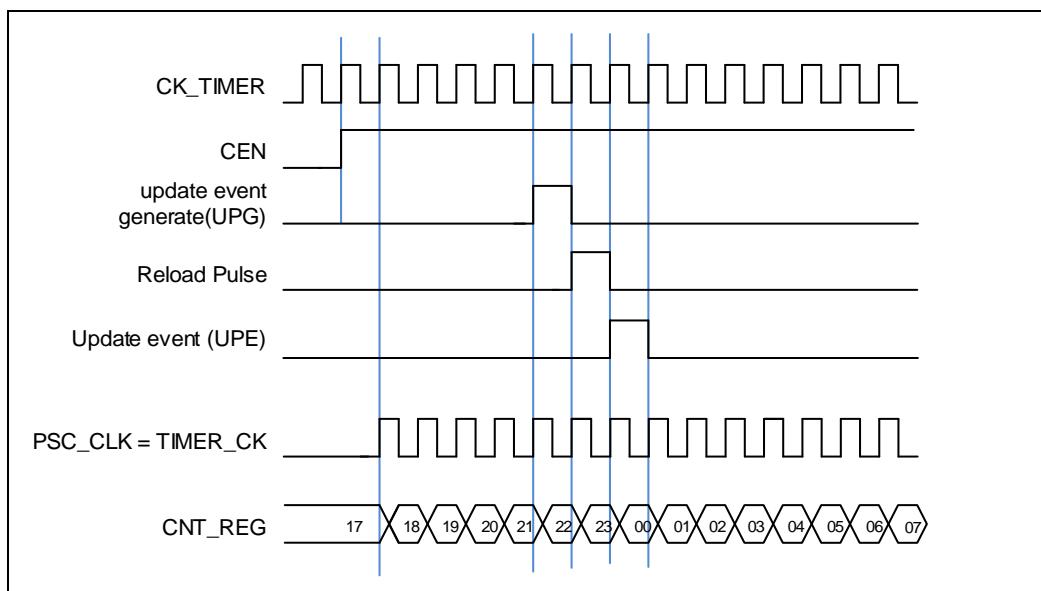
- SMC[2:0]=3'b000, 定时器选择内部时钟源（连接到RCU模块的CK\_TIMER）

如果SMC[2:0]=3'b000, 默认用来驱动计数器预分频器的是内部时钟源CK\_TIMER。当CEN置位, CK\_TIMER经过预分频器（预分频值由TIMERx\_PSC寄存器确定）产生PSC\_CLK。

这种模式下, 驱动预分频器计数的TIMER\_CK等于来自于RCU模块的CK\_TIMER。

如果将TIMERx\_SMCFG寄存器的SMC[2:0]设置为0x1、0x2、0x3和0x7, 预分频器被其他时钟源(由TIMERx\_SMCFG寄存器的TRGS[2:0]区域选择)驱动, 在下文说明。当SMC位被设置为0x4、0x5和0x6, 计数器预分频器时钟源由内部时钟CK\_TIMER驱动。

图 14-54. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]=3'b111 (外部时钟模式0)，定时器选择外部输入引脚作为时钟源。

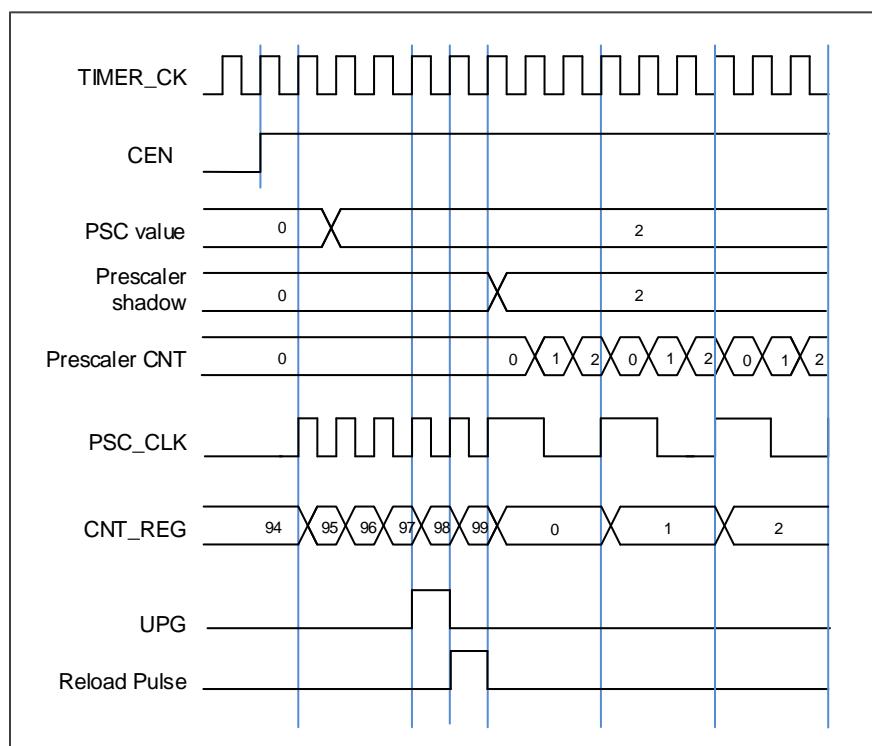
计数器预分频器可以在TIMERx\_CH0/TIMERx\_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。Clx是TIMERx\_Cl通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号ITIO/1/2/3的上升沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x0, 0x1, 0x2 或者0x3。

### 时钟预分频器

预分频器可以将定时器的时钟(TIMER\_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC\_CLK驱动计数器计数。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 14-55. 当 PSC 数值从 0 变到 2 时，计数器的时序图



### 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到计数器自动加载值(定义在TIMERx\_CAR寄存器中)，一旦计数器计数到自动加载值，会重新从0开始向上计数，并且产生上溢事件。另外，在(TIMERx\_CREP+1)次上溢后将会产生更新事件。在向上计数模式中，TIMERx\_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx\_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx\_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数寄存器，自动重载寄存器，预分频寄存器)都将被更新。

[图 14-56. 向上计数时序图, PSC=0/2](#) 和 [图 14-57. 向上计数时序图, 在运行时改变 TIMERx\\_CAR 寄存器的值](#)给出了一些例子，当 TIMERx\_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 14-56. 向上计数时序图, PSC=0/2

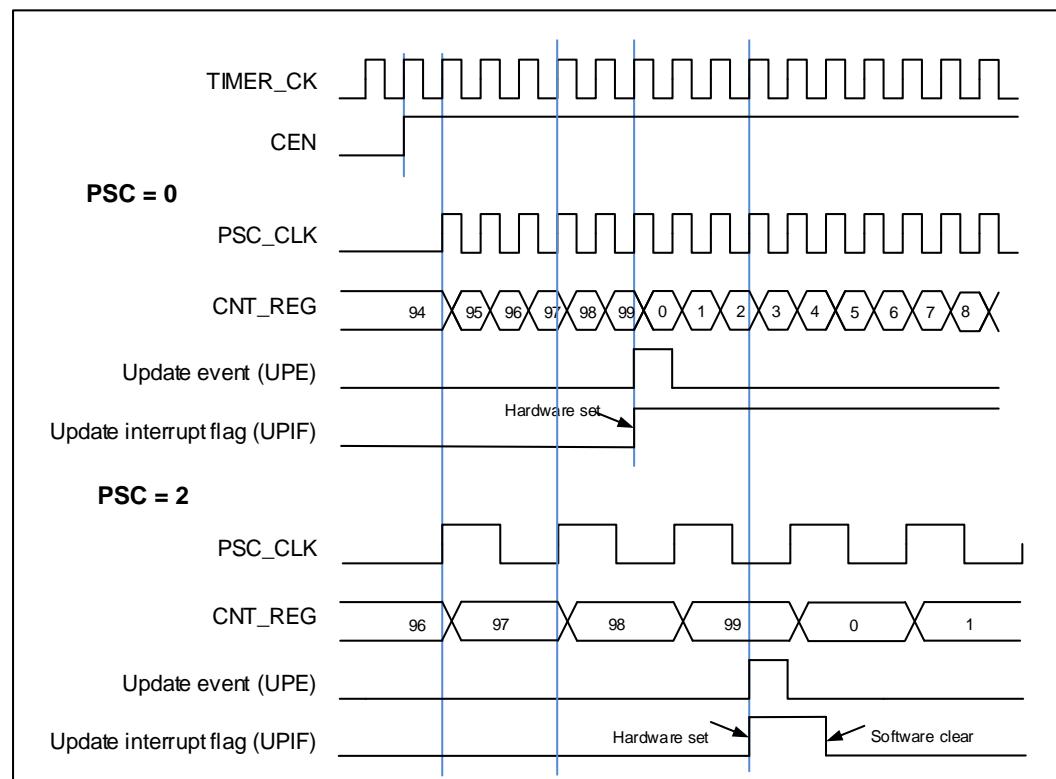
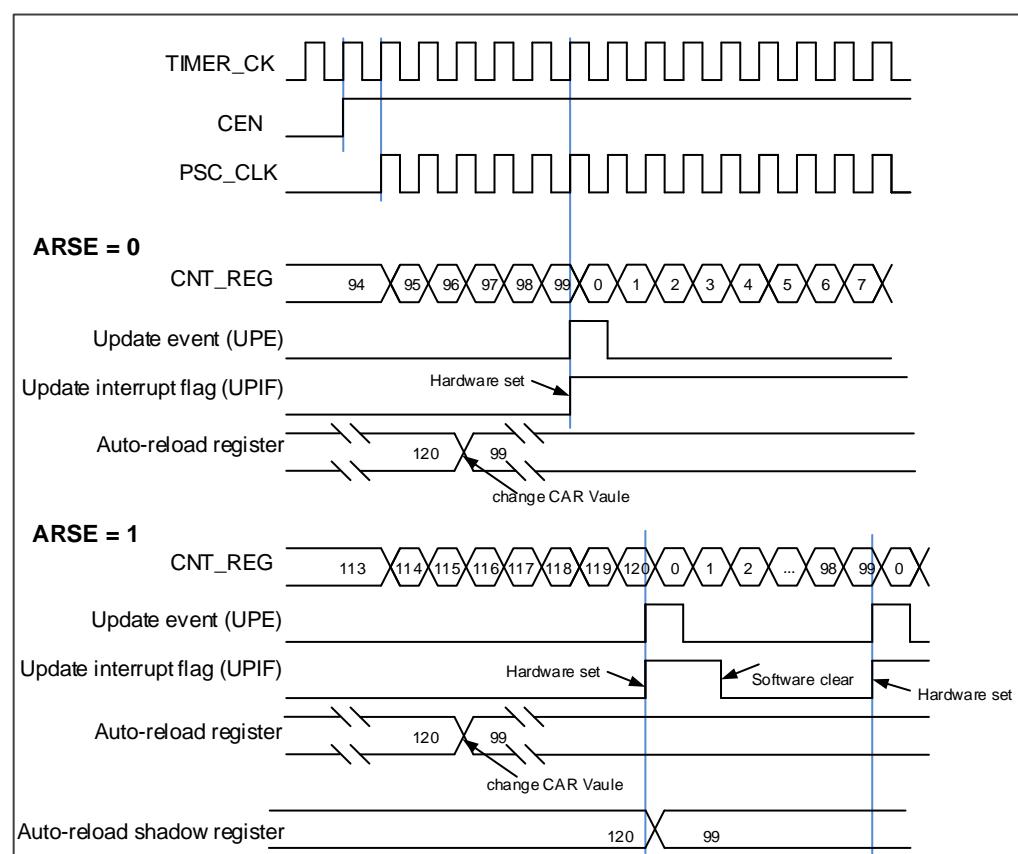


图 14-57. 向上计数时序图, 在运行时改变 TIMERx\_CAR 寄存器的值

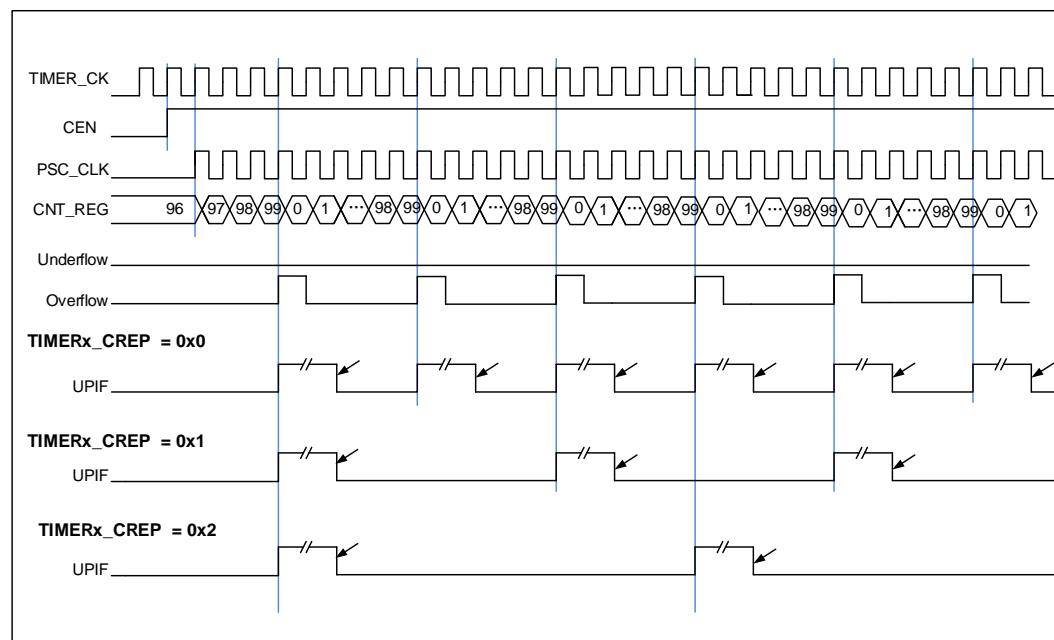


### 更新事件（来自上溢/下溢）频率配置

重复计数器是用来在  $N+1$  个计数周期之后产生更新事件，更新定时器的寄存器， $N$  为 **TIMERx\_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。

将 **TIMERx\_SWEVG** 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 14-58. 在向上计数模式下计数器重复时序图**



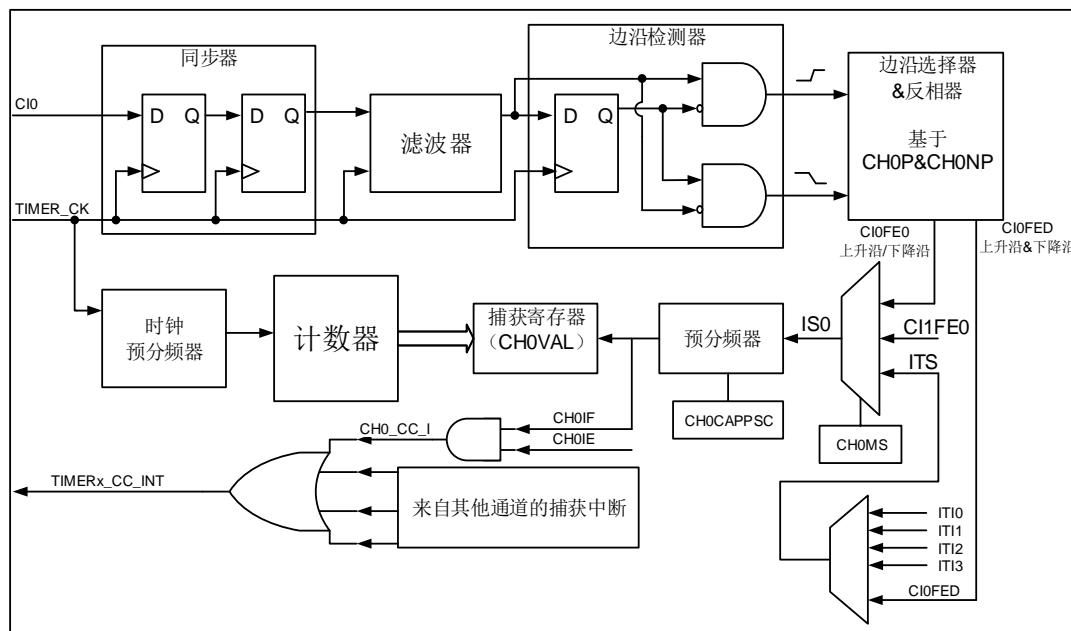
### 输入捕获和输出比较通道

通用定时器L3拥有两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。

#### ■ 通道输入捕获功能

输入捕获功能允许通道测量一个波形的时序，频率，周期和占空比等。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。如果在输入引脚上出现被选择的边沿，**TIMERx\_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，若**CHxIE=1**则产生通道中断。

图 14-59. 通道输入捕获原理



通道输入信号CIx来源于TIMERx\_CHx信号。通道输入信号CIx先被TIMER\_CK信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置CHxP选择使用上升沿或者下降沿。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，**TIMERx\_CHxCV**存储计数器的值。

配置步骤如下：

**第一步：滤波器配置 (TIMERx\_CHCTL0寄存器中CHxCAPFLT):**

根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

**第二步：边沿选择 (TIMERx\_CHCTL2寄存器中CHxP/CHxNP):**

配置CHxP/CHxNP选择上升沿或者下降沿。

**第三步：捕获源选择 (TIMERx\_CHCTL0寄存器中CHxMS):**

一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且**TIMERx\_CHxCV**寄存器不能再被写。

**第四步：中断使能 (TIMERx\_DMAINTEN寄存器中CHxIE和CHxDEN):**

使能相应中断，可以获得中断和DMA请求。

**第五步：捕获使能 (TIMERx\_CHCTL2寄存器中CHxEN)。**

**结果：**当期望的输入信号发生时，**TIMERx\_CHxCV**被设置成当前计数器的值，CHxIF为置1。

如果CHxIF位已经为1，则CHxOF位置1。根据**TIMERx\_DMAINTEN**寄存器中CHxIE和CHxDEN的配置，判断相应的中断和DMA请求是否被提出。

**直接产生：**软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量**TIMERx\_CHx**引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置**TIMERx\_CHCTL0**寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0，

同时设置上升沿捕获。配置TIMERx\_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0，同时设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX\_CH0CV寄存器测量PWM的周期值，TIMERx\_CH1CV寄存器测量PWM占空比值。

### ■ 通道输出比较功能

图 14-60. 通道输出比较原理（带有互补输出的通道， $x=0$ ）

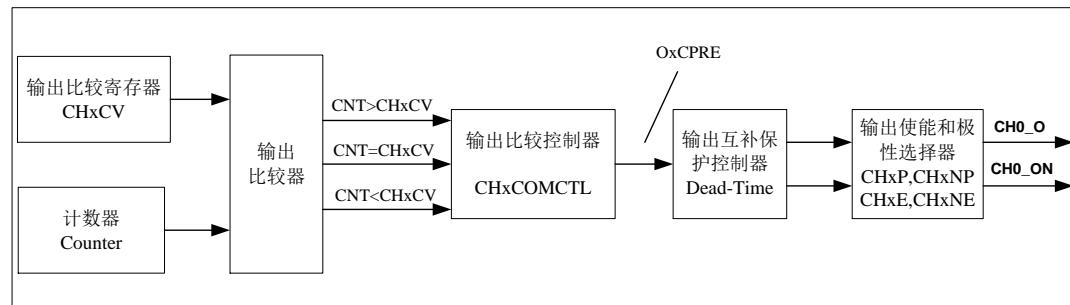
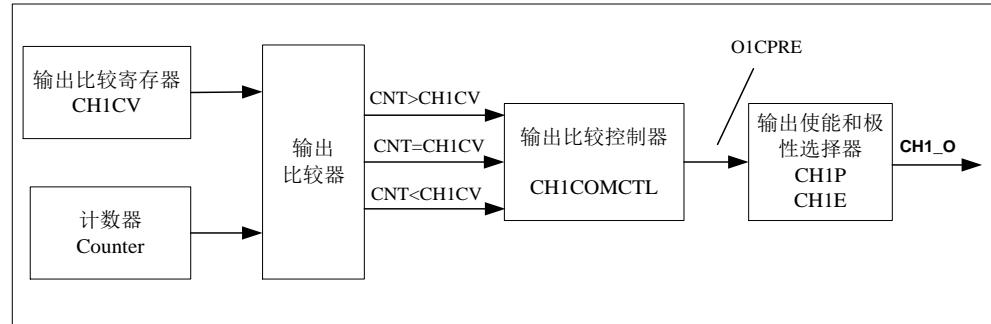


图 14-61. 通道输出比较原理



[图14-60. 通道输出比较原理（带有互补输出的通道， \$x=0\$ ）](#) 和 [图14-61. 通道输出比较原理](#) 分别给出了输出比较的原理电路。通道输出信号CHx\_O/CHx\_ON与OxCPR信号（详情请见[通道输出准备信号](#)）的关系描述如下：OxCPR信号高电平有效，CHx\_O/CHx\_ON的输出情况与OxCPR信号，CHxP/CHxNP位和CHxE/CHxNE位有关（具体情况请见TIMERx\_CHCTL2寄存器中的描述）。例如：

1) 当设置CHxP=0 (CHx\_O高电平有效，与OxCPR输出极性相同)、CHxE=1 (CHx\_O输出使能) 时：

- 若OxCPR输出有效（高）电平，则CHx\_O输出有效（高）电平；
- 若OxCPR输出无效（低）电平，则CHx\_O输出无效（低）电平。

2) 当设置CHxNP=1 (CHx\_ON低电平有效，与OxCPR输出极性相反)、CHxNE=1 (CHx\_ON输出使能) 时：

- 若OxCPR输出有效（高）电平，则CHx\_ON输出有效（低）电平；
- 若OxCPR输出无效（低）电平，则CHx\_ON输出无效（高）电平。

当CH0\_O和CH0\_ON同时输出时，CH0\_O和CH0\_ON的具体输出情况还与TIMERx\_CCHP寄存器中的相关位（ROS、IOS、POE和DTCFG等位）有关。详情请见[通道输出互补PWM](#)。

在通道输出比较功能，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的TIMERx\_CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与

TIMERx\_CHxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxCDE=1则会产生DMA请求。

配置步骤如下：

**第一步：时钟配置：**

配置定时器时钟源，预分频器等。

**第二步：比较模式配置：**

- 设置CHxCOMSEN位来配置输出比较影子寄存器；
- 设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
- 设置CHxP/CHxNP位来选择有效电平的极性；
- 设置CHxEN使能输出。

**第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。**

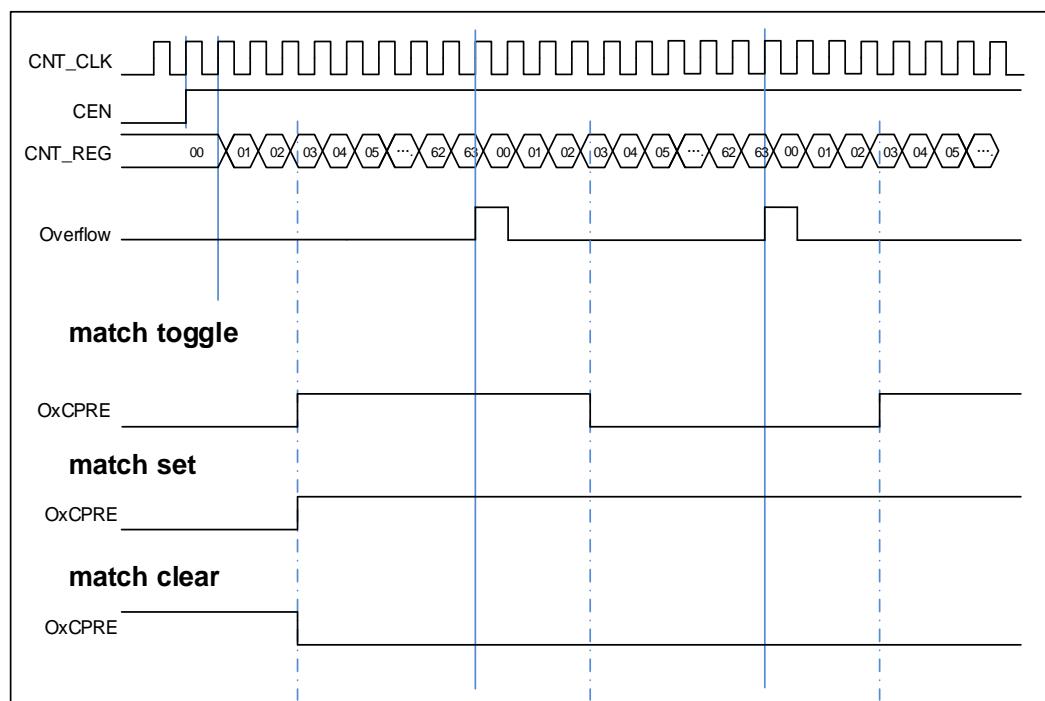
**第四步：通过TIMERx\_CAR寄存器和TIMERx\_CHxCV寄存器配置输出比较时基：**

TIMERx\_CHxCV可以在运行时根据你所期望的波形而改变。

**第五步：设置CEN位使能定时器。**

**图14-62. 三种输出比较模式**显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 14-62. 三种输出比较模式



### 输出 PWM 功能

在PWM输出模式下（PWM模式0是配置CHxCOMCTL为3'b110，PWM模式1是配置CHxCOMCTL为3'b111），通道根据TIMERx\_CAR寄存器和TIMERx\_CHxCV寄存器的值，输出

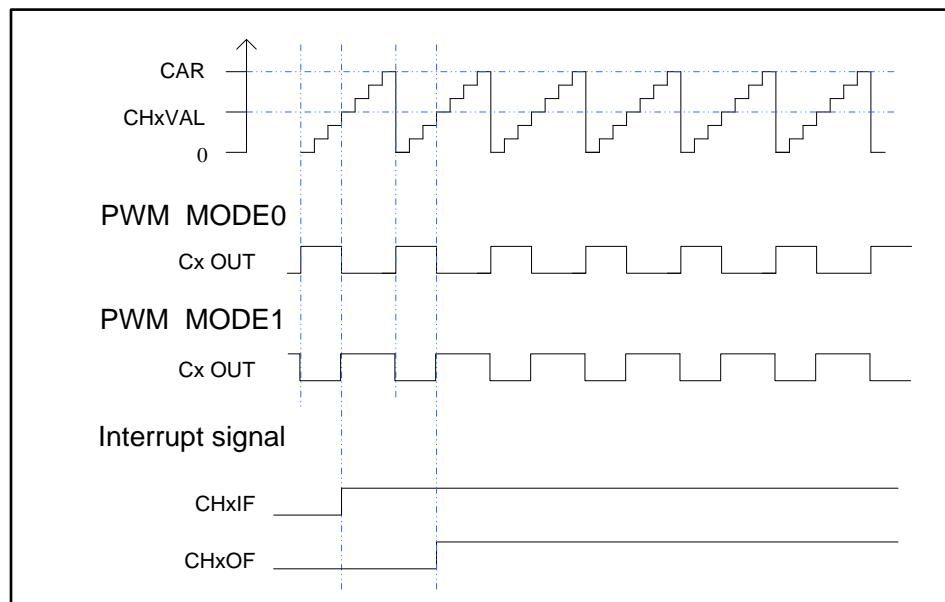
PWM波形。

PWM的周期由TIMERx\_CAR寄存器值决定，占空比由TIMERx\_CHxCV寄存器值决定。[图14-63. PWM时序图](#)显示了PWM的输出波形和中断。

在PWM0模式下(CHxCOMCTL==3'b110)，如果TIMERx\_CHxCV寄存器的值大于TIMERx\_CAR寄存器的值，通道输出一直为有效电平。

在PWM0模式下(CHxCOMCTL==3'b110)，如果TIMERx\_CHxCV寄存器的值等于0，通道输出一直为无效电平。

**图 14-63. PWM 时序图**



### 通道输出准备信号

根据[图14-60. 通道输出比较原理（带有互补输出的通道, x=0）](#)所示，当TIMERx用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号OxCPRE信号(通道x输出准备信号)。设置CHxCOMCTL位可以定义OxCPRE信号类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx\_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx\_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL=0x04或0x05可以实现OxCPRE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TIMERx\_CHxCV的值和计数器值之间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE信号才会回到有效电平状态。

### 通道输出互补 PWM

CH<sub>x</sub>\_O和CH<sub>x</sub>\_ON是一对互补输出通道，这两个信号不能同时有效。TIMER<sub>x</sub>有两路通道，只有第一路有互补输出通道。互补信号CH<sub>x</sub>\_O和CH<sub>x</sub>\_ON是由一组参数来决定：TIMER<sub>x</sub>\_CHCTL2寄存器中的CHxEN和CHxNEN位，TIMER<sub>x</sub>\_CCHP寄存器中和TIMER<sub>x</sub>\_CTL1寄存器中的POEN, ROS, IOS, ISO<sub>x</sub>和ISO<sub>x</sub>N位。输出极性由TIMER<sub>x</sub>\_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 14-5. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出禁用.	
			1	0		
				1	如果时钟使能: CHx_O = ISO <sub>x</sub> CHx_ON = ISO <sub>x</sub> N  CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
		1	0	0		
				1		
			1	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出使能.  如果时钟使能: CHx_O = ISO <sub>x</sub> CHx_ON = ISO <sub>x</sub> N	
				1		
1	0/1	0	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON 输出禁用.	
				1	CHx_O = LOW CHx_O 输出禁用.	CHx_ON=OxCPRE⊕CHxNP CHx_ON 输出使能
			1	0	CHx_O=OxCPRE CHxP CHx_O输出使能	⊕ CHx_ON = LOW CHx_ON输出禁用.
				1	CHx_O=OxCPRE CHxP CHx_O输出使能	⊕ CHx_ON=(!OxCPRE)⊕ CHxNP CHx_ON输出使能
		1	0	0	CHx_O = CHxP CHx_O输出禁用.	CHx_ON = CHxNP CHx_ON输出禁用.
				1	CHx_O = CHxP	CHx_ON=OxCPRE⊕CHxNP

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
					CHx_O输出使能	CHx_ON输出使能
					CHx_O=OxCPRE CHxP CHx_O输出使能	$\oplus$ CHx_ON = CHxNP CHx_ON输出使能
				1	CHx_O=OxCPRE CHxP CHx_O输出使能	$\oplus$ CHx_ON=(!OxCPRE) $\oplus$ CHxNP CHx_ON输出使能

### 互补 PWM 插入死区时间

设置CHxEN和CHxNEN为1'b1同时设置POEN，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对通道0有效。死区时间的细节，请参考TIMERx\_CCHP寄存器。

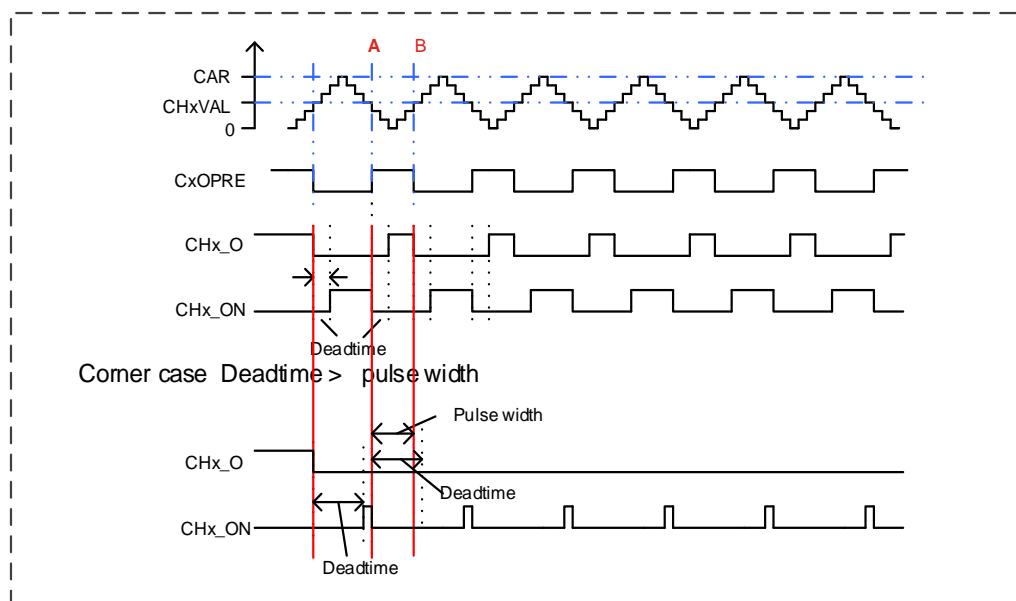
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在PWM0模式，当通道x匹配发生时（TIMERx计数器= CHxVAL），OxCPR反转。[在图14-64. 带死区时间的互补输出](#)中的A点，CHx\_O信号在死区时间内为低电平，直到死区时间过后才变为高电平，而CHx\_ON信号立刻变为低电平。同样，在B点，计数器再次匹配（TIMERx计数器= CHxVAL），OxCPR信号被清0，CHx\_O信号被立即清零，CHx\_ON信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于CHx\_O信号的占空比，CHx\_O信号一直为无效值（如[图14-64. 带死区时间的互补输出](#)）。
- 如果死区延时大于或者等于CHx\_ON信号的占空比，CHx\_ON信号一直为无效值。

图 14-64. 带死区时间的互补输出



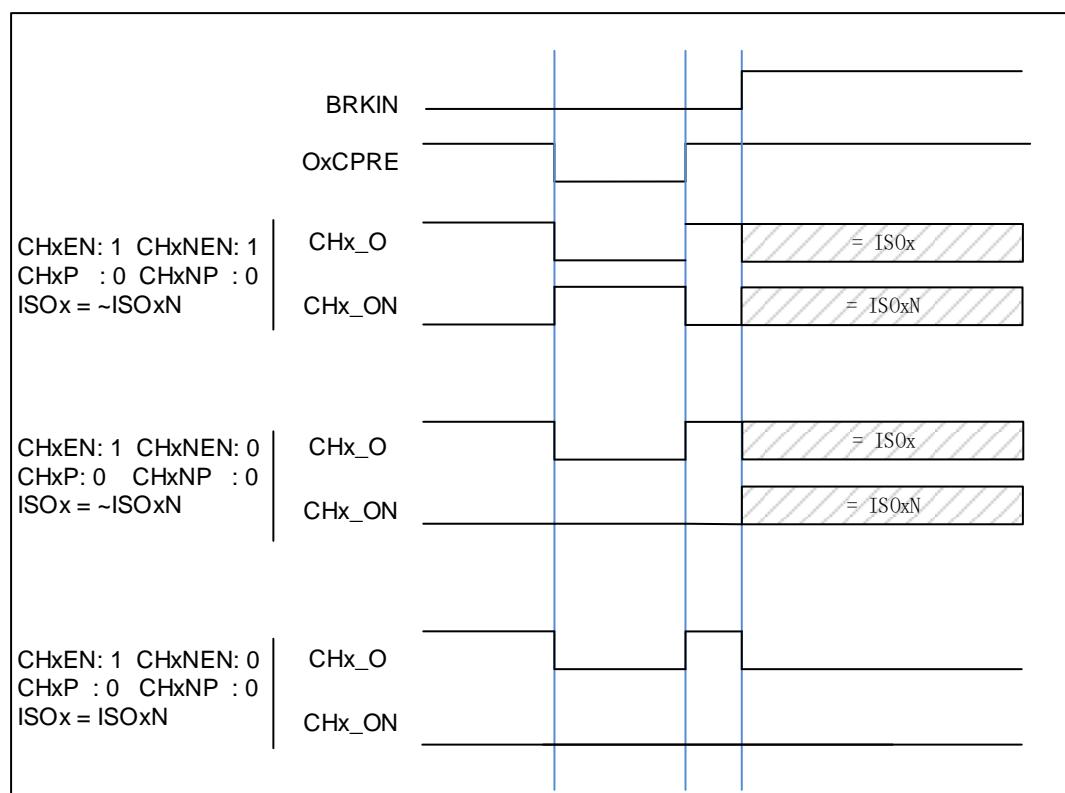
### 中止模式

使用中止模式时，输出CH<sub>x</sub>\_O和CH<sub>x</sub>\_ON信号电平被以下位控制，TIMER<sub>x</sub>\_CCHP寄存器的POEN, IOS和ROS位，TIMER<sub>x</sub>\_CTL1寄存器的ISO<sub>x</sub>和ISO<sub>x</sub>N位。当中止事件发生时，CH<sub>x</sub>\_O和CH<sub>x</sub>\_ON信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择HXTAL时钟失效事件。时钟失效事件由RCU中的时钟监视器(CKM)产生。将TIMER<sub>x</sub>\_CCHP寄存器的BRKEN位置1可以使能中止功能。TIMER<sub>x</sub>\_CCHP寄存器的BRKP位决定了中止输入极性。

发生中止时，POEN位被异步清除，一旦POEN位为0，CH<sub>x</sub>\_O和CH<sub>x</sub>\_ON被TIMER<sub>x</sub>\_CTL1寄存器中的ISO<sub>x</sub>位和ISO<sub>x</sub>N驱动。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ISO<sub>x</sub>和ISO<sub>x</sub>N位配置。

发生中止时，TIMER<sub>x</sub>\_INTF寄存器的BRKIF位被置1。如果BRKIE=1，中断产生。

图 14-65. 通道响应中止输入（高电平有效）时，输出信号的行为

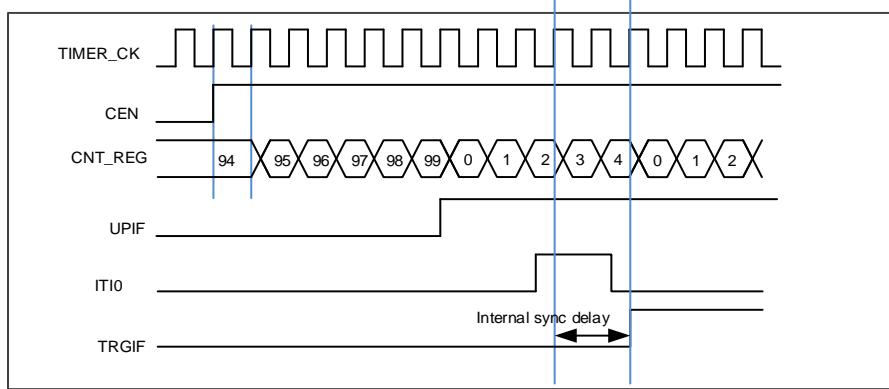
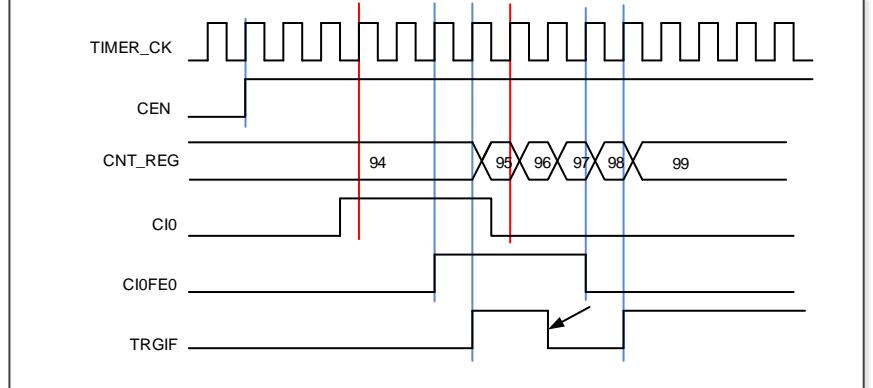


### 主-从管理

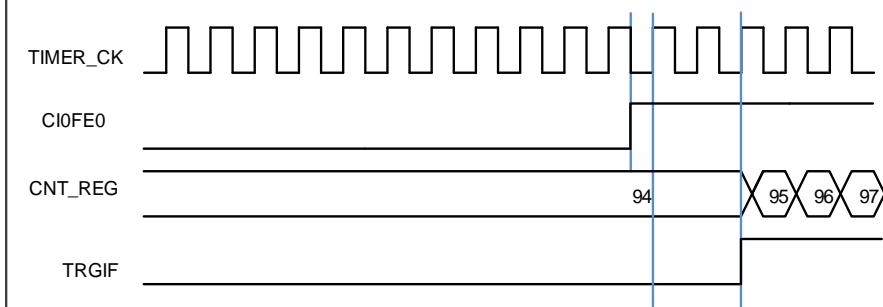
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx\_SMCFG寄存器中的SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx\_SMCFG寄存器中的TRGS [2:0]来选择。

表 14-6. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: 保留	如果触发源是 CI0FE0 或者 CI1FE1，配置 CHxP 和 CHxNP 来选择极性和反相	触发源ITIx, 滤波和预分频不可用 触发源CIx, 配置 CHxCAPFLT 设置滤波, 分频不可用
例1	复位模式 当触发输入上升沿, 计数器清零	TRGIS[2:0]=3'b000 选择ITI0为触发源	触发源是 ITI0, 极性选择不可用	触发源是 ITI0, 滤波和预分频不可用

	模式选择	触发源选择	极性选择	滤波和预分频	
	重启				
<b>图 14-66. 复位模式下的控制电路</b>					
				 <p>The timing diagram illustrates the control signals for the timer in reset mode. It shows the following waveforms over 10 clock cycles:</p> <ul style="list-style-type: none"> <li><b>TIMER_CK:</b> A square wave clock signal.</li> <li><b>CEN:</b> A low-to-high transition at the start of each cycle.</li> <li><b>CNT_REG:</b> A counter register showing values from 94 to 99, followed by 0, 1, 2, 3, 4, 0, 1, 2.</li> <li><b>UPIF:</b> An interrupt flag that goes high during the 4th cycle and remains high until the end of the sequence.</li> <li><b>ITIO:</b> An output signal that goes high during the 4th cycle and remains high until the end of the sequence.</li> <li><b>TRGIF:</b> A trigger flag that goes high during the 4th cycle and remains high until the end of the sequence. A double-headed arrow labeled "Internal sync delay" indicates a fixed delay between the UPIF and TRGIF rising edges.</li> </ul>	
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3'b101 选择 CI0FE0 为触发源	TI0S=0. (非异或) $[CH0NP==0, CH0P==0]$ 不反相。 在上升沿捕获	在这个例子中滤波被旁路	
	<b>图 14-67. 暂停模式下的控制电路</b>				
				 <p>The timing diagram illustrates the control signals for the timer in pause mode. It shows the following waveforms over 10 clock cycles:</p> <ul style="list-style-type: none"> <li><b>TIMER_CK:</b> A square wave clock signal.</li> <li><b>CEN:</b> A low-to-high transition at the start of each cycle.</li> <li><b>CNT_REG:</b> A counter register showing values from 94 to 99, followed by 0, 1, 2, 3, 4, 99.</li> <li><b>CI0:</b> An input signal that goes high during the 4th cycle and remains high until the end of the sequence.</li> <li><b>CI0FE0:</b> An output signal that goes high during the 4th cycle and remains high until the end of the sequence.</li> <li><b>TRGIF:</b> A trigger flag that goes high during the 4th cycle and remains high until the end of the sequence.</li> </ul>	
例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3'b101 选择 CI0FE0 为触发源。	TI0S=0. (非异或) $[CH0NP==0, CH0P==0]$ 不反相	在这个例子中滤波被旁路	

	模式选择	触发源选择	极性选择	滤波和预分频
	图 14-68. 事件模式下的控制电路			



### 单脉冲模式

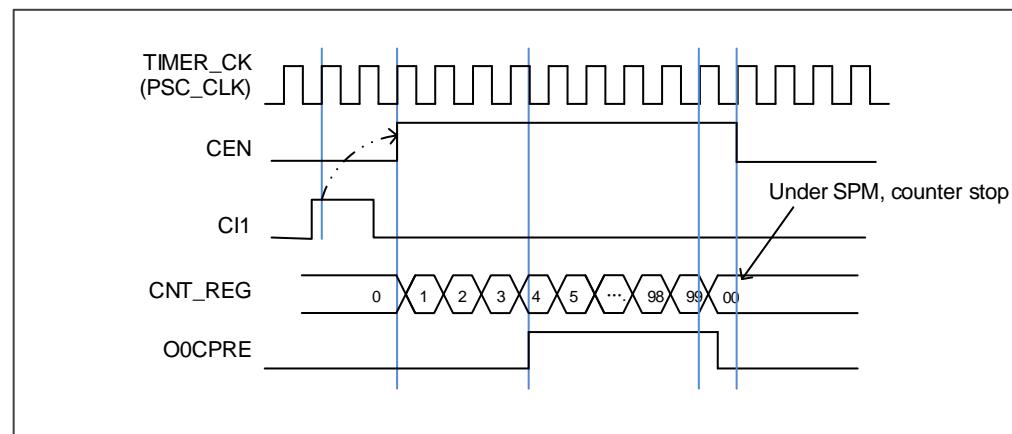
单脉冲模式与重复模式是相反的，设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1，则使能单脉冲模式。当 `SPM` 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 `CHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 `TIMERx_CTL0` 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。

在单脉冲模式下，有效的外部触发边沿会将 `CEN` 位置 1，使能计数器。然而，执行计数值和 `TIMERx_CHxCV` 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_CHCTL0` 寄存器的 `CHxCOMFEN` 位置 1。单脉冲模式下，触发上升沿产生之后，`OxCPRE` 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 `PWM0` 或 `PWM1` 输出运行模式下时 `CHxCOMFEN` 位才可用，触发源来源于触发信号。

[图 14-69. 单脉冲模式, `TIMERx\_CHxCV = 4` `TIMERx\_CAR=99`](#) 展示了一个例子。

图 14-69. 单脉冲模式, `TIMERx_CHxCV = 4` `TIMERx_CAR=99`



## 定时器互连

参考[定时器互连](#)

## 定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：**TIMERx\_DMACFG**和**TIMERx\_DMATB**。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，**TIMERx**会给DMA发送请求。DMA配置成M2P（传输方向为从内存到外设）模式，**PADDR**（外设基地址）为**TIMERx\_DMATB**寄存器地址，DMA就会访问**TIMERx\_DMATB**寄存器。实际上，**TIMERx\_DMATB**寄存器只是一个缓冲，定时器会将**TIMERx\_DMATB**映射到一个内部寄存器，这个内部寄存器由**TIMERx\_DMACFG**寄存器中的**DMATA**来指定。如果**TIMERx\_DMACFG**寄存器的**DMATC**位域值为0，表示1次传输，定时器发送1个DMA请求就可以完成。如果**TIMERx\_DMACFG**寄存器的**DMATC**位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对**TIMERx\_DMATB**寄存器的访问会映射到访问定时器的**DMATA+0x4**，**DMATA+0x8**，**DMATA+0xC**寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(**DMATC+1**)次请求。

如果再来1次DMA请求事件，**TIMERx**将会重复上面的过程。

## 定时器调试模式

当Cortex®-M23内核停止，**DBG\_CTL1**寄存器中的**TIMERx\_HOLD**配置位被置1，定时器计数器停止。

#### 14.4.5. TIMERx 寄存器(x=14)

TIMER13基地址: 0x4000 2000

##### 控制寄存器 0 (TIMERx\_CTL0)

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				CKDIV[1:0]		ARSE		保留				SPM		UPS		UPDIS		CEN	
				rw		rw						rw		rw		rw		rw	

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV, 规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。</p> <p>00: <math>f_{DTS}=f_{CK\_TIMER}</math></p> <p>01: <math>f_{DTS}=f_{CK\_TIMER}/2</math></p> <p>10: <math>f_{DTS}=f_{CK\_TIMER}/4</math></p> <p>11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能 TIMERx_CAR 寄存器的影子寄存器。</p> <p>1: 使能TIMERx_CAR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: 单脉冲模式禁能。更新事件发生后, 计数器继续计数</p> <p>1: 单脉冲模式使能。在下一次更新事件发生时, 计数器停止计数</p>
2	UPS	<p>更新请求源</p> <p>软件配置该位, 选择更新事件源.</p> <p>0: 以下事件均会产生更新中断或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>复位模式产生的更新</li> </ul> <p>1: 下列事件会产生更新中断或DMA请求:</p> <ul style="list-style-type: none"> <li>计数器溢出/下溢</li> </ul>

1	UPDIS	<p>禁止更新.</p> <p>该位用来使能或禁能更新事件的产生</p> <p>0: 更新事件使能. 更新事件发生时, 相应的影子寄存器被装入预装载值, 以下事件均会产生更新事件:</p> <ul style="list-style-type: none"> <li>UPG位被置1</li> <li>计数器溢出/下溢</li> <li>复位模式产生的更新</li> </ul> <p>1: 更新事件禁能.</p> <p>注意: 当该位被置1时, UPG位被置1或者复位模式不会产生更新事件, 但是计数器和预分频器被重新初始化</p>
0	CEN	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</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。</p>

### 控制寄存器 1 (TIMERx\_CTL1)

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					ISO1	ISOON	ISO0	保留	MMC[2:0]	DMAS	CCUC	保留	CCSE		

位/位域	名称	描述
31:11	保留	必须保持复位值
10	ISO1	通道1的空闲状态输出 参考ISO0位
9	ISOON	通道0的互补通道空闲状态输出 0: 当POEN复位, CH0_ON设置低电平. 1: 当POEN复位, CH0_ON设置高电平 此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.
8	ISO0	通道0的空闲状态输出 0: 当POEN复位, CH0_O设置低电平 1: 当POEN复位, CH0_O设置高电平  如果CH0_ON生效, 一个死区时间后CH0_O输出改变. 此位只有在TIMERx_CCHP寄存器的PROT [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]	<p>主模式控制I</p> <p>这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能</p> <p>000: 当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1</p> <p>001: 当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1</p> <p>010: 当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定</p> <p>011: 当通道0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: 当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE</p> <p>101: 当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE</p> <p>110: 保留</p> <p>111: 保留</p>
3	DMAS	<p>DMA请求源选择</p> <p>0: 当通道捕获/比较事件发生时，发送通道x的DMA请求 .</p> <p>1: 当更新事件发生，发送通道x的DMA请求</p>
2	CCUC	<p>换相控制影子寄存器更新控制</p> <p>当换相控制影子寄存器（CHxEN, CHxNEN和CHxCOMCTL位）使能(CCSE=1)，这些影子寄存器更新控制如下：</p> <p>0: CMTG位被置1时更新影子寄存器</p> <p>1: 当CMTG位被置1或检测到TRIGI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。</p>
1	保留	必须保持复位值.
0	CCSE	<p>换相控制影子使能</p> <p>0: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位禁能.</p> <p>1: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位使能.</p> <p>如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效</p>

### 从模式配置寄存器 (TIMERx\_SMCFG)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	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	MSM	TRGS[2:0]	保留	SMC[2:0]
		rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	MSM	<p>主-从模式</p> <p>该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。</p> <p>0: 主从模式禁能。</p> <p>1: 主从模式使能。</p>
6:4	TRGS[2:0]	<p>触发选择</p> <p>该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源。</p> <ul style="list-style-type: none"> <li>000: ITIO</li> <li>001: ITI1</li> <li>010: ITI2</li> <li>011: ITI3</li> <li>100: CI0F_ED</li> <li>101: CI0FE0</li> <li>110: CI1FE1</li> <li>111: 保留</li> </ul> <p>从模式被使能后这些位不能改。</p>
3	保留	必须保持复位值
2:0	SMC[2:0]	<p>从模式控制</p> <ul style="list-style-type: none"> <li>000: 关闭从模式。如果 CEN=1，则预分频器直接由内部时钟驱动。</li> <li>001: 保留。</li> <li>010: 保留。</li> <li>011: 保留。</li> <li>100: 复位模式。选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。</li> <li>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，则计数器时钟停止。</li> <li>110: 事件模式。计数器在触发输入的上升沿启动。</li> <li>111: 外部时钟模式 0。选中的触发输入的上升沿驱动计数器。</li> </ul> <p>由于 CI0F_ED 是一个脉冲波形，而暂停模式是检测触发信号的电平，所以，当 CI0F_ED 用作触发输入时，暂停模式必须禁能。</p>

### DMA 和中断使能寄存器 (TIMERx\_DMAINTEN)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	保留	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	保留	CH1IE	CHOIE	UPIE		

RW      RW

位/位域	名称	描述
31:15	保留	必须保持复位值.
14	TRGDEN	触发 DMA 请求使能 0: 禁止触发 DMA 请求 1: 使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0: 禁止换相 DMA 更新请求 1: 使能换相 DMA 更新请求
12:11	保留	必须保持复位值.
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0: 禁止通道 1 比较/捕获 DMA 请求 1: 使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁止通道 0 比较/捕获 DMA 请求 1: 使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7	BRKIE	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	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4:3	保留	必须保持复位值.
2	CH1IE	通道 1 比较/捕获中断使能 0: 禁止通道 1 中断 1: 使能通道 1 中断

1	CH0IE	通道 0 比较/捕获中断使能 0: 禁止通道 0 中断 1: 使能通道 0 中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### 中断标志寄存器 (**TIMERx\_INTF**)

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				CH1OF	CH0OF	保留.	BRKIF	TRGIF	CMTIF	保留.	CH1IF	CH0IF	UPIF		
				rc_w0	rc_w0		rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0		

位/位域	名称	描述
31:11	保留	必须保持复位值.
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时, 在CH0IF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: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 当中止输入有效时, 由硬件对该位置‘1’。 当中止输入无效时, 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, 此标志会置 1, 此位由软件清 0。当暂停模式使能时, 触发输入的任意边沿都可以产生触发事件。否则, 其它模式时, 仅在触发输入端检测到有效边沿, 产生触发事件。 0: 无触发事件产生

		1: 触发中断产生
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置1，此位由软件清0。 0: 无通道换相更新中断发生 1: 通道换相更新中断发生
4:3	保留	必须保持复位值.
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0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0中断发生 1: 通道0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

### 软件事件产生寄存器 (**TIMERx\_SWEVG**)

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								BRKG	TRGG	CMTG	保留		CH1G	CH0G	UPG
w								w	w	w	w		w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值.
7	BRKG	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，POEN位被清0且BRKIF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\_INTF**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	CMTG	通道换相更新事件发生 此位由软件置1，由硬件自动清0。当此位被置1，通道捕获/比较控制寄存器（CHxEN, CHxNEN 和CHxCOMCTL）的互补输出被更新（根据 <b>TIMERx_CTL1</b> 中CCSE值）。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4:3	保留	必须保持复位值。
2	CH1G	通道1捕获或比较事件发生 参见 <b>CH0G</b> 描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1， <b>CH0IF</b> 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0配置为输入模式，计数器的当前值被 <b>TIMERx_CH0CV</b> 寄存器捕获，如果 <b>CH0IF</b> 标志位已经为1，则 <b>CH0OF</b> 标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 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

### 通道控制寄存器 0 (**TIMERx\_CHCTL0**)

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	CH1COMCTL[2:0]	CH1COMSEN	CH1COMFEN	CH1MS[1:0]	保留	CH0COMCTL[2:0]	CH0COMSEN	CH0COMFEN	CH0MS[1:0]						
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]			CH0CAPFLT[3:0]	CH0CAPPSC[1:0]										

rw

rw

rw

rw

rw

rw

**输出比较模式：**

位/位域	名称	描述
31:15	保留	必须保持复位值。
14:12	CH1COMCTL[2:0]	通道1输出比较模式 参见CH0COMCTL描述
11	CH1COMSEN	通道1输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH1COMFEN	通道1输出比较快速使能 参见CH0COMFEN描述
9:8	CH1MS[1:0]	通道 1 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_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00: 通道 1 配置为输出 01: 通道 1 配置为输入, IS1 映射在 CI1FE1 上 10: 通道 1 配置为输入, IS1 映射在 CI0FE1 上 11: 通道 1 配置为输入, IS1 映射在 ITS 上 注意: 当CH1MS[1:0]=11时, 需要通过TRGS位(位于TIMERx_SMCFG寄存器)选择内部触发输入。
7	保留	必须保持复位值。
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式, 而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外, O0CPRE 高电平有效, 而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000: 时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时, 强制 O0CPRE 翻转。 100: 强制为低。强制 O0CPRE 为低电平 101: 强制为高。强制 O0CPRE 为高电平 110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH0CV 时, O0CPRE 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_CH0CV 时, O0CPRE 为低电平, 否则为高电平。 111: PWM 模式 1。在向上计数时, 一旦计数器值小于 TIMERx_CH0CV 时, O0CPRE 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_CH0CV 时, O0CPRE 为高电平, 否则为低电平。 如果配置在 PWM 模式下, 只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时, O0CPRE 电平才改变。 当TIMERx_CCHP寄存器的PROT[1:0]=11且CH0MS =00 (比较模式) 时此位不能被

改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1, TIMERx_CH0CV 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 0 输出/比较影子寄存器 1: 使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1), 可以在未确认影子寄存器的情况下使用 PWM 模式 当TIMERx_CCHP 寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时, 如果通道配置为 PWM0 模式或者 PWM1 模式, 会加快捕获/比较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, CH0_O 被设置为比较电平而与比较结果无关。 0: 禁止通道 0 输出比较快速. 1: 使能通道0输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00: 通道 0 配置为输出 01: 通道 0 配置为输入, ISO 映射在 CI0FE0 上 10: 通道 0 配置为输入, ISO 映射在 CI1FE0 上 11: 通道 0 配置为输入, ISO 映射在 ITS 上. 注意: 当CH0MS[1:0]=11时, 需要通过TRGS位(位于TIMERx_SMCFG 寄存器)选择内部触发输入。

#### 输入捕获模式:

位/位域	名称	描述
31:15	保留	必须保持复位值.
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波, 该位域配置滤波参数。 数字滤波器的基本原理: 根据 fSAMP 对 CI0 输入信号进行连续采样, 并记录信号相同电平的次数。达到该位配置的滤波参数后, 认为是有效电平。 滤波器参数配置如下:

CH0CAPFLT [3:0]	采样次数	fSAMP
4'b0000	无滤波器	
4'b0001	2	fCK_TIMER
4'b0010	4	
4'b0011	8	
4'b0100	6	
4'b0101	8	fDTS/2
4'b0110	6	
4'b0111	8	fDTS/4
4'b1000	6	
4'b1001	8	fDTS/8
4'b1010	5	
4'b1011	6	fDTS/16
4'b1100	8	
4'b1101	5	fDTS/32
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx\_CHCTL2 寄存器中的 CH0EN =0 时，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 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

## 通道控制寄存器 2 (TIMERx\_CHCTL2)

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.						CH1NP	保留	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN		

位/位域	名称	描述
31:8	保留	必须保持复位值
7	CH1NP	通道1互补输出极性

## 参考CH0NP描述

6	保留	必须保持复位值
5	CH1P	通道1极性 参考CH0P 描述
4	CH1EN	通道1使能 参考CH0EN 描述
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。 当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。
2	CH0NEN	通道0互补输出使能 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。 0：禁止通道0互补输出 1：使能通道0互补输出
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道0配置为输入模式时，此位定义了CI0信号极性 [CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。 [CH0NP==0, CH0P==1]: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。 [CH0NP==1, CH0P==0]: 保留。 [CH0NP==1, CH0P==1]: 把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。 当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0：禁止通道0 1：使能通道0

**计数器寄存器 (TIMERx\_CNT)**

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
保留															
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	保留	必须保持复位值.
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。

### 预分频寄存器 (TIMERx\_PSC)

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
PSC[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (TIMERx\_CAR)

地址偏移: 0x2C

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
CAR[15:0]															

CARL[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	CARL[15:0]	<p>计数器自动重载值</p> <p>这些位定义了计数器的自动重载值。</p> <p><b>注意:</b> 在定时器被配置为输入捕获模式时, 该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。</p>

### 重复计数寄存器 (TIMERx\_CREP)

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								CREP[7:0]							
rw															

位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP[7:0]	<p>重复计数器的值</p> <p>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)。</p>

### 通道 0 捕获/比较寄存器 (TIMERx\_CH0CV)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]								rw							
rw															

位/位域	名称	描述
------	----	----

31:16	保留	必须保持复位值.
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 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 0 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。

### 通道 1 捕获/比较寄存器 (**TIMERx\_CH1CV**)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	CH1VAL[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。

### 互补通道保护寄存器 (**TIMERx\_CCHP**)

地址偏移: 0x44

复位值: 0x0000 0000

该寄存器通过字访问 (32位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN OAEN BRKP BRKEN ROS IOS PROT[1:0] DTCFG[7:0]															
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	POEN	<p>所有的通道输出使能</p> <p>该位通过以下方式置 1:</p> <ul style="list-style-type: none"> <li>-写 1 置位</li> <li>-如果 OAEN=1，则在下一次更新事件发生时置 1.</li> </ul> <p>该位通过以下方式清 0:</p> <ul style="list-style-type: none"> <li>-写 0 清 0</li> <li>-有效的中止输入（异步）</li> </ul> <p>如果一个通道配置为输出模式，如果设置了相应的使能位（<b>TIMERx_CHCTL2</b> 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。</p> <p>0: 禁止通道输出或强制为空闲状态。</p> <p>1: 使能通道输出。</p>
14	OAEN	<p>自动输出使能</p> <p>0: POEN 位只能使用软件方式置 1。</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1。</p> <p>此位只有在 <b>TIMERx_CCHP</b> 寄存器的 PROT [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 BKIN 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	BRKEN	<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: 禁能中止输入。</p> <p>1: 使能中止输入。</p> <p>此位只有在 <b>TIMERx_CCHP</b> 寄存器的 PROT [1:0] =00 时才可修改。</p>
11	ROS	<p>运行模式下“关闭状态”配置</p> <p>当 POEN 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。</p> <p>0: 当 POEN 位被置 1，通道输出信号（CHx_O/ CHx_ON）被禁止。</p> <p>1: 当 POEN 位被置 1，通道输出信号（CHx_O / CHx_ON）被使能，和 <b>TIMER0_CHCTL2</b> 寄存器 CHxEN/CHxNEN 位有关。</p> <p>此位在 <b>TIMERx_CCHP</b> 寄存器的 PROT [1:0]=10 或 11 时不能被更改。</p>
10	IOS	<p>空闲模式下“关闭状态”配置</p> <p>当 POEN 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。</p> <p>0: 当 POEN 位被清 0，通道输出信号（CHx_O/ CHx_ON）被禁止。</p> <p>1: 当 POEN 位被清 0，通道输出信号（CHx_O/ CHx_ON）被使能，和 <b>TIMERx_CHCTL2</b> 寄存器 CHxEN/CHxNEN 位有关。</p> <p>此位在 <b>TIMERx_CCHP</b> 寄存器的 PROT [1:0]=10 或 11 时不能被更改。</p>
9:8	PROT[1:0]	<p>互补寄存器保护控制</p> <p>这两位定义了寄存器的写保护特性。</p> <p>00: 禁能保护模式。无写保护。</p> <p>01: PROT 模式 0。<b>TIMERx_CTL1</b> 寄存器中 ISOx/ISOxN 位，<b>TIMERx_CCHP</b> 寄</p>

存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。

10: PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx\_CHCTL2 寄存器中 CHxP/ChNP 位（如果相应通道配置为输出模式），TIMERx\_CCHP 寄存器中 ROS/IOS 位。

11: PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx\_CHCTL0 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。

系统复位后这两位只能被写一次，一旦 TIMERx\_CCHP 寄存器被写入，这两位被写保护。

7:0	DTCFG[7:0]	死区时间控制
这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下：		
DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x tDT, tDT=tDTS.		
DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0]) xtDT, tDT =tDTS*2.		
DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0]) xtDT, tDT=tDTS*8.		
DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0]) xtDT, tDT =tDTS*16.		
此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。		

### DMA 配置寄存器 (TIMERx\_DMACFG)

地址偏移: 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
保留				DMATC[4:0]				保留			DMATA [4:0]				

rw

rw

位/位域	名称	描述
31:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMATB寄存器的数量n, n = (DMATC [4:0]+1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当通过TIMERx_DMATB第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问起始地址+0x4。

### DMA 发送缓冲区寄存器 (TIMERx\_DMATB)

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	DMATB [15:0]	<p>DMA 发送缓冲</p> <p>对这个寄存器的读或写, (起始地址+传输次数*4) 地址范围内的寄存器会被访问</p> <p>传输次数由硬件计算, 范围为 0 到 DMATC。</p>

### 配置寄存器 (TIMERx\_CFG)

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															

rw      rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	<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: 无影响</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POEN位与IOS位均为0, 则输出无效</p>

---

0: 无影响

## 14.5. 通用定时器 L4 (TIMERx,x=15,16)

### 14.5.1. 简介

通用定时器L4 (TIMER15/16) 是单通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。通用定时器L4含有一个16位无符号计数器。

通用定时器L4是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

通用定时器L4包含了一个死区时间插入模块，非常适合电机控制。

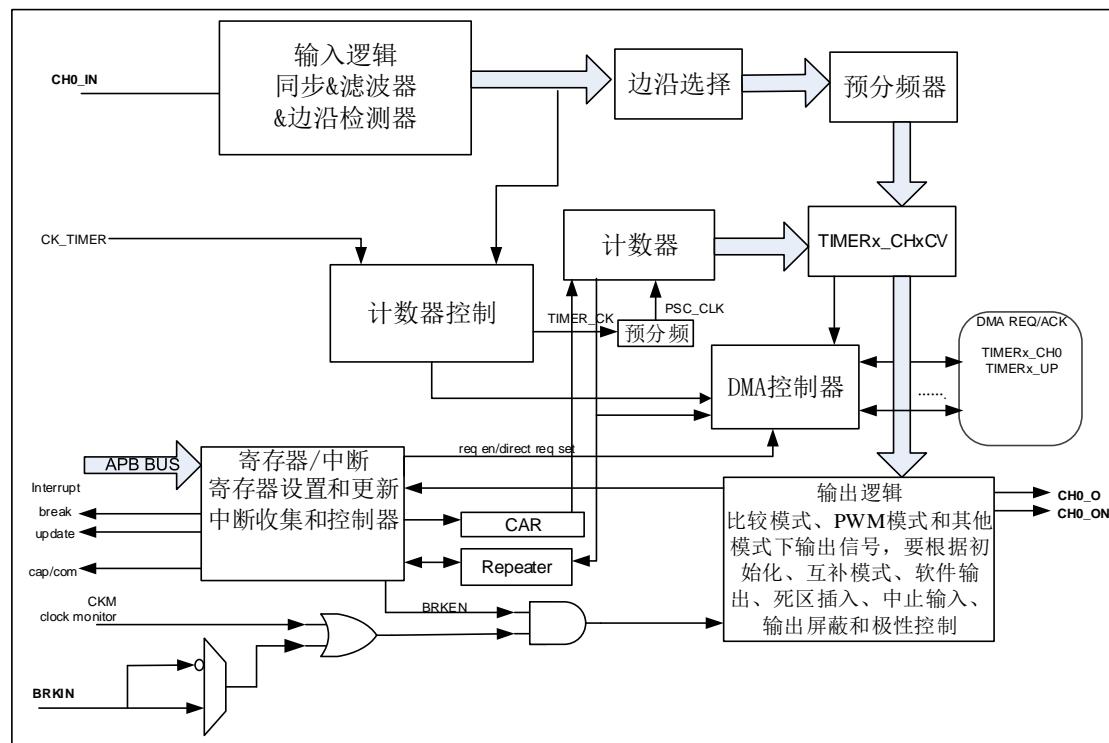
### 14.5.2. 主要特性

- 总通道数: 1;
- 计数器宽度: 16位;
- 时钟源可选: 内部时钟;
- 计数模式: 向上计数;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式, 单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和DMA请求: 更新事件, 比较/捕获事件, 换相事件和中止事件;

### 14.5.3. 结构框图

[图14-70. 通用定时器L4结构框图](#)提供了通用定时器L4的内部配置细节。

图 14-70. 通用定时器 L4 结构框图



#### 14.5.4. 功能描述

##### 时钟源配置

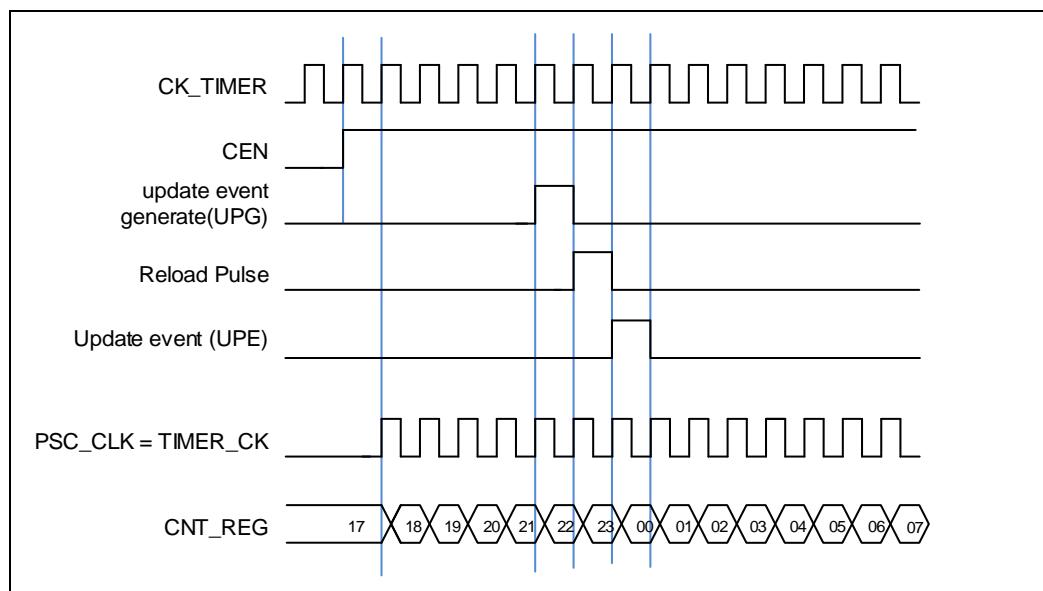
通用定时器L4由内部时钟源CK\_TIMER驱动。

- 定时器选择内部时钟源（连接到RCU模块的CK\_TIMER）

通用定时器L4只有一个时钟源：内部时钟源。用来驱动计数器预分频器的是内部时钟源CK\_TIMER。当CEN置位，CK\_TIMER经过预分频器（预分频值由TIMERx\_PSC寄存器确定）产生PSC\_CLK。

驱动预分频器计数的TIMER\_CK等于来自于RCU模块的CK\_TIMER

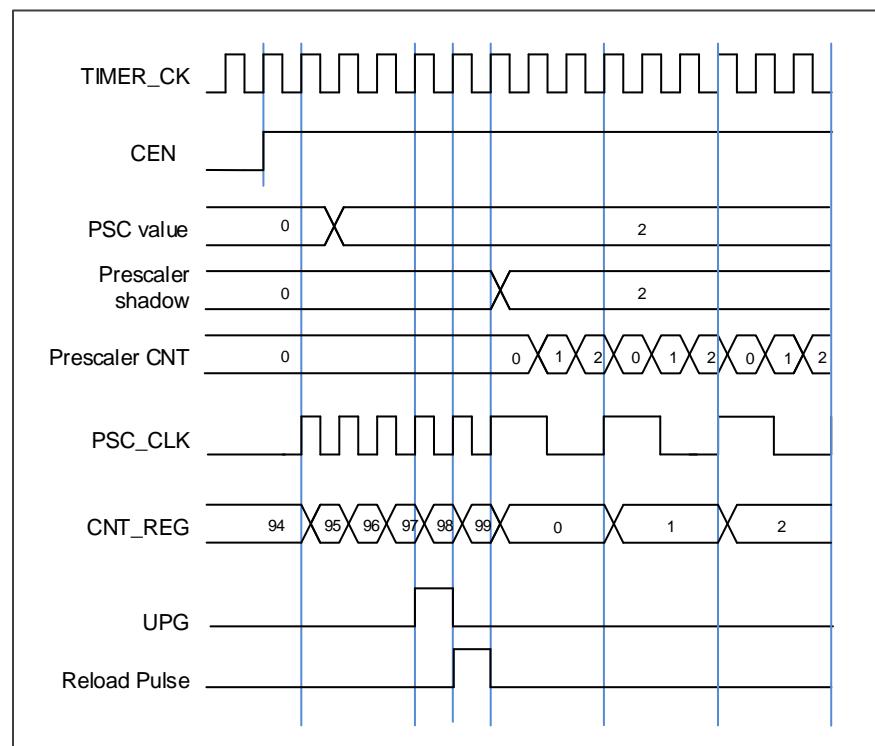
图 14-71. 内部时钟分频为 1 时, 计数器的时序图



### 时钟预分频器

预分频器可以将定时器的时钟 (TIMER\_CK) 频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC\_CLK 驱动计数器计数。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 14-72. 当 PSC 数值从 0 变到 2 时, 计数器的时序图



## 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到计数器自动加载值(定义在TIMERx\_CAR寄存器中)，一旦计数器计数到自动加载值，会重新从0开始向上计数，并且产生上溢事件。另外，在(TIMERx\_CREP+1)次上溢后将会产生更新事件。在向上计数模式中，TIMERx\_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx\_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx\_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

**图 14-73. 向上计数时序图，PSC=0/2** 给出了一些例子，当 TIMERx\_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 14-73. 向上计数时序图，PSC=0/2

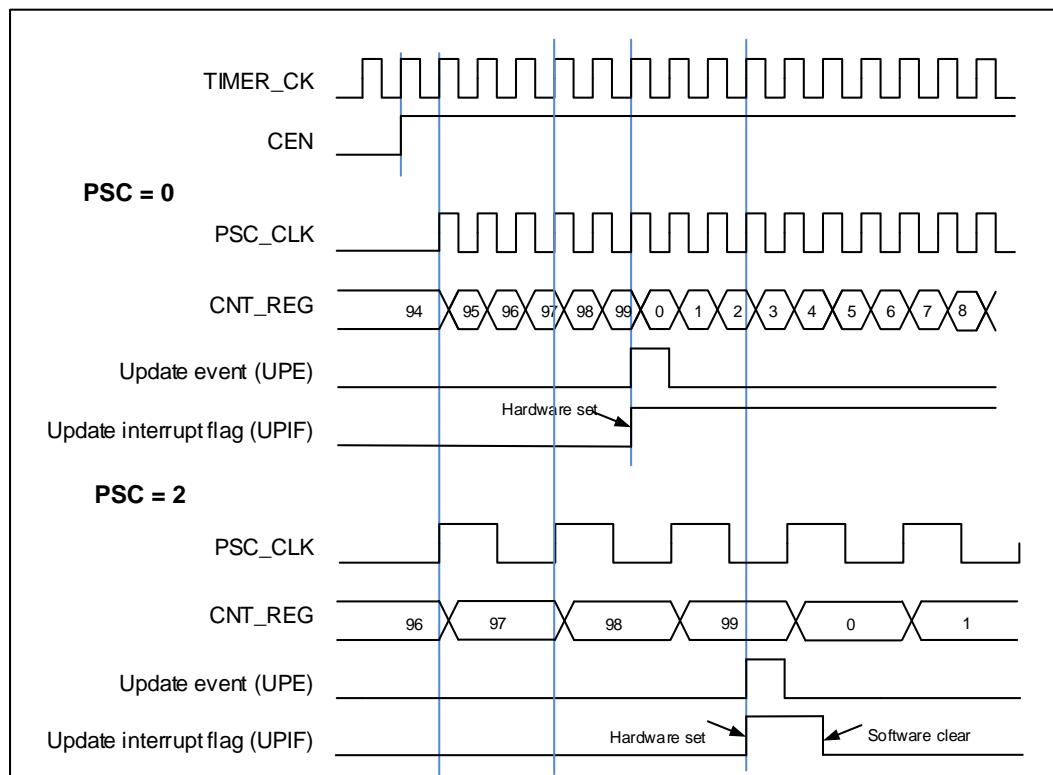
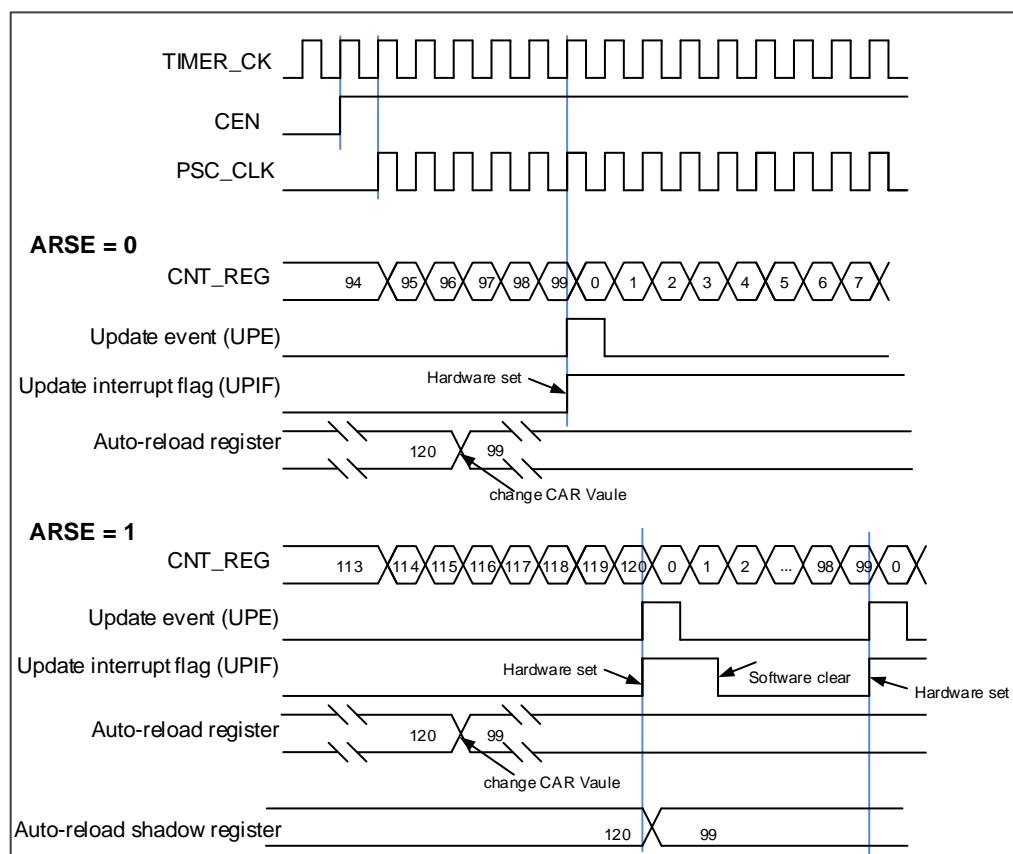


图 14-74. 向上计数时序图，在运行时改变 TIMERx\_CAR 寄存器的值

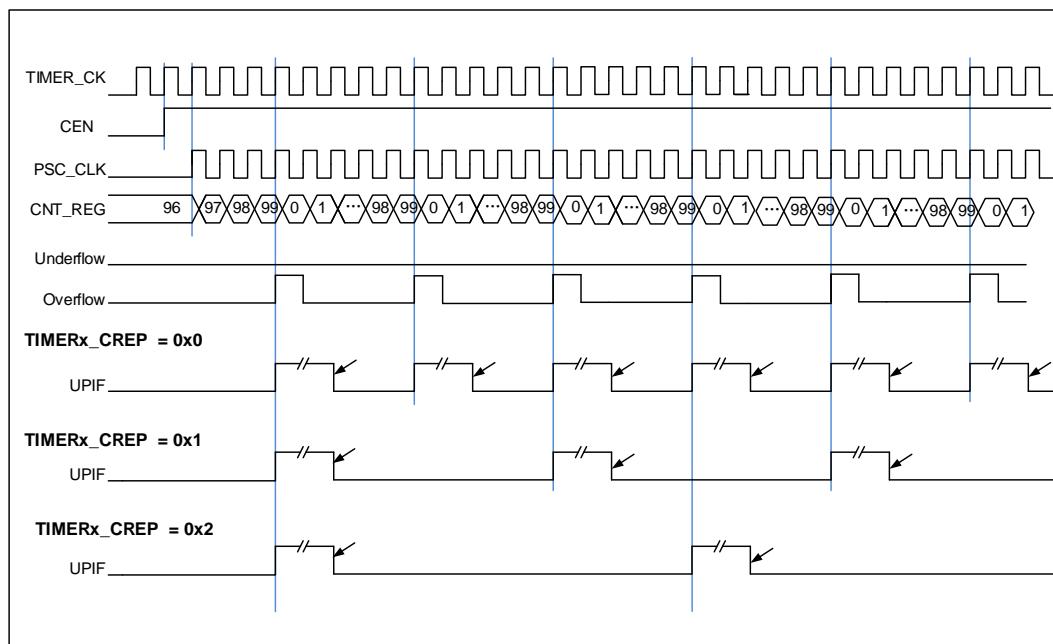


#### 更新事件（来自上溢/下溢）频率配置

重复计数器是用来在  $N+1$  个计数周期之后产生更新事件，更新定时器的寄存器， $N$  为 TIMERx\_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。

将 TIMERx\_SWEVG 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 14-75. 在向上计数模式下计数器重复时序图



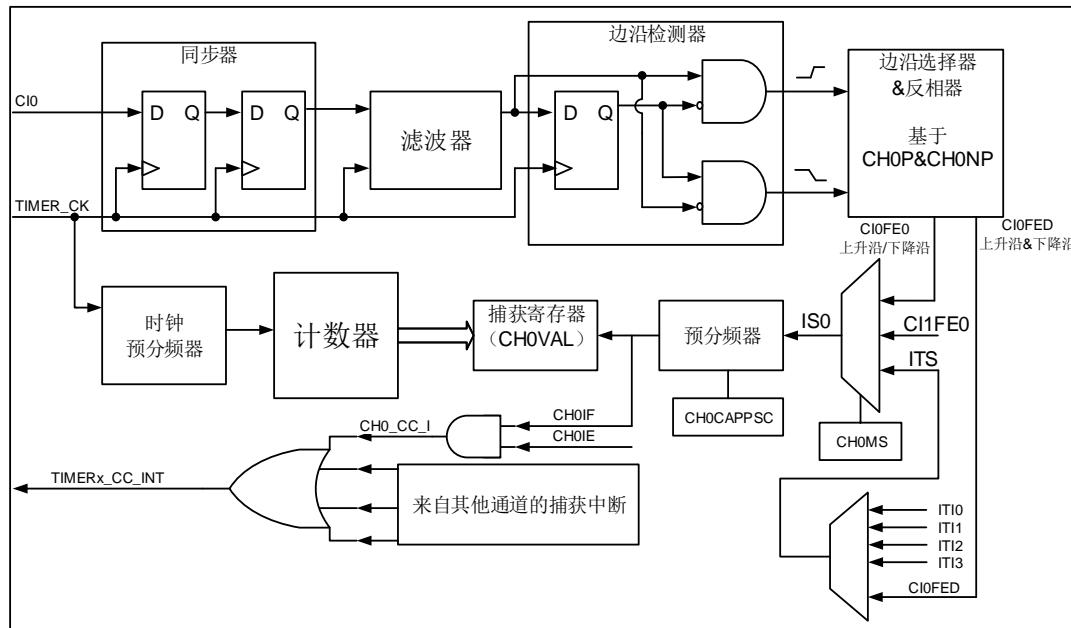
### 输入捕获和输出比较通道

通用定时器L4拥有一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。

#### ■ 通道输入捕获功能

输入捕获功能允许通道测量一个波形的时序，频率，周期等。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。如果在输入引脚上出现被选择的边沿，**TIMERx\_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，若**CHxIE=1**则产生通道中断。

图 14-76. 通道输入捕获原理



通道输入信号 $\text{CI}_x$ 来源于 $\text{TIMER}_{\text{x}}\text{-CH}_x$ 信号。通道输入信号 $\text{CI}_x$ 先被 $\text{TIMER\_CK}$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 $\text{CH}_x\text{P}$ 选择使用上升沿或者下降沿。通过配置 $\text{CH}_x\text{MS}$ ，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $\text{TIMER}_{\text{x}}\text{-CH}_x\text{CV}$ 存储计数器的值。

配置步骤如下：

**第一步：滤波器配置（ $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 $\text{CH}_x\text{CAPFLT}$ ）：**

根据输入信号和请求信号的质量，配置相应的 $\text{CH}_x\text{CAPFLT}$ 。

**第二步：边沿选择（ $\text{TIMER}_{\text{x}}\text{-CHCTL2}$ 寄存器中 $\text{CH}_x\text{P}/\text{CH}_x\text{NP}$ ）：**

配置 $\text{CH}_x\text{P}/\text{CH}_x\text{NP}$ 选择上升沿或者下降沿。

**第三步：捕获源选择（ $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 $\text{CH}_x\text{MS}$ ）：**

一旦通过配置 $\text{CH}_x\text{MS}$ 选择输入捕获源，必须确保通道配置在输入模式（ $\text{CH}_x\text{MS} \neq 0x0$ ），而且 $\text{TIMER}_{\text{x}}\text{-CH}_x\text{CV}$ 寄存器不能再被写。

**第四步：中断使能（ $\text{TIMER}_{\text{x}}\text{-DMAINTEN}$ 寄存器中 $\text{CH}_x\text{IE}$ 和 $\text{CH}_x\text{DEN}$ ）：**

使能相应中断，可以获得中断和DMA请求。

**第五步：捕获使能（ $\text{TIMER}_{\text{x}}\text{-CHCTL2}$ 寄存器中 $\text{CH}_x\text{EN}$ ）。**

**结果：**当期望的输入信号发生时， $\text{TIMER}_{\text{x}}\text{-CH}_x\text{CV}$ 被设置成当前计数器的值， $\text{CH}_x\text{IF}$ 为置1。

如果 $\text{CH}_x\text{IF}$ 位已经为1，则 $\text{CH}_x\text{OF}$ 位置1。根据 $\text{TIMER}_{\text{x}}\text{-DMAINTEN}$ 寄存器中 $\text{CH}_x\text{IE}$ 和 $\text{CH}_x\text{DEN}$ 的配置，判断相应的中断和DMA请求是否被提出。

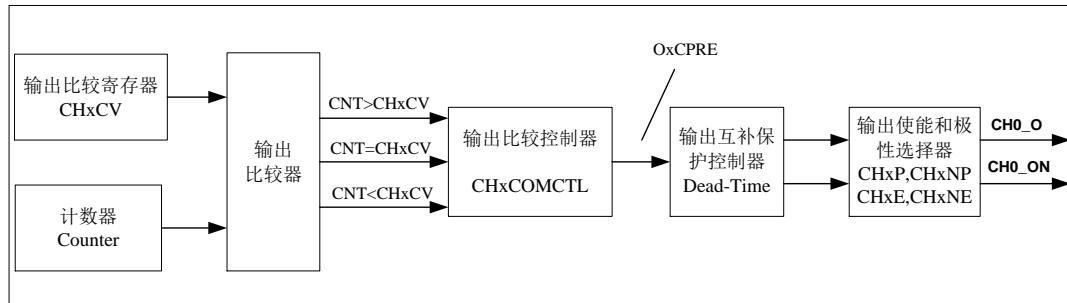
**直接产生：**软件设置 $\text{CH}_x\text{G}$ 位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量 $\text{TIMER}_{\text{x}}\text{-CH}_x$ 引脚上信号的脉冲波周期。例如，一个PWM波连接到 $\text{CI}_0$ 。配置 $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 $\text{CH}_0\text{MS}$ 为2'b01，选择通道0的捕获信号为 $\text{CI}_0$ ，同时

设置上升沿捕获。TIMERX\_CH0CV寄存器测量PWM的周期值。

### ■ 通道输出比较功能

图 14-77. 通道输出比较原理（带有互补输出的通道， $x=0$ ）



[图14-77. 通道输出比较原理（带有互补输出的通道， \$x=0\$ ）](#) 给出了输出比较的原理电路。通道输出信号CHx\_O/CHx\_ON与OxCPRE信号（详情请见[通道输出准备信号](#)）的关系描述如下：

OxCPRE信号高电平有效，CHx\_O/CHx\_ON的输出情况与OxCPRE信号，CHxP/CHxNP位和CHxE/CHxNE位有关（具体情况请见TIMERx\_CHCTL2寄存器中的描述）。例如：

1) 当设置CHxP=0 (CHx\_O高电平有效，与OxCPRE输出极性相同)、CHxE=1 (CHx\_O输出使能) 时：

- 若OxCPRE输出有效（高）电平，则CHx\_O输出有效（高）电平；
- 若OxCPRE输出无效（低）电平，则CHx\_O输出无效（低）电平。

2) 当设置CHxNP=1 (CHx\_ON低电平有效，与OxCPRE输出极性相反)、CHxNE=1 (CHx\_ON输出使能) 时：

- 若OxCPRE输出有效（高）电平，则CHx\_ON输出有效（低）电平；
- 若OxCPRE输出无效（低）电平，则CHx\_ON输出无效（高）电平。

当CH0\_O和CH0\_ON同时输出时，CH0\_O和CH0\_ON的具体输出情况还与TIMERx\_CCHP寄存器中的相关位（ROS、IOS、POE和DTCFG等位）有关。详情请见[通道输出互补PWM](#)。

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的TIMERx\_CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与TIMERx\_CHxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxODE=1则会产生DMA请求。

配置步骤如下：

#### 第一步：时钟配置：

配置定时器时钟源，预分频器等。

#### 第二步：比较模式配置：

- 设置CHxCOMSEN位来配置输出比较影子寄存器；
- 设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
- 设置CHxP/CHxNP位来选择有效电平的极性；
- 设置CHxEN使能输出。

#### 第三步：通过CHxIE/CxODE位配置中断/DMA请求使能。

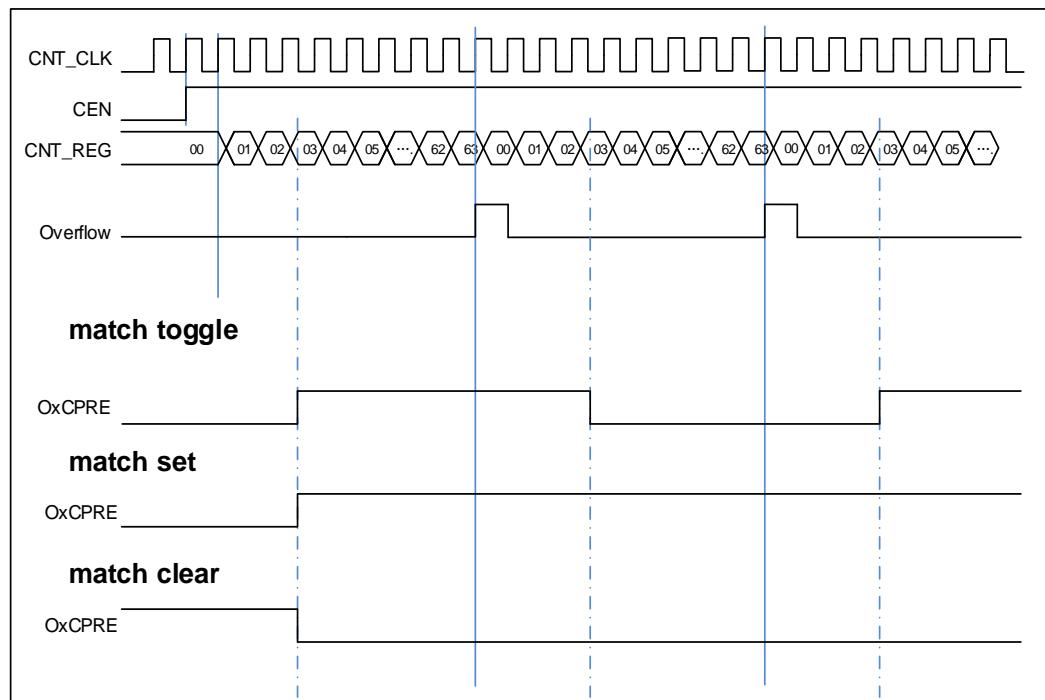
第四步：通过TIMERx\_CAR寄存器和TIMERx\_CHxCV寄存器配置输出比较时基：

TIMERx\_CHxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

[图14-78. 三种输出比较模式](#)显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 14-78. 三种输出比较模式



### 输出 PWM 功能

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110， PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx\_CAR 寄存器和 TIMERx\_CHxCV 寄存器的值，输出 PWM 波形。

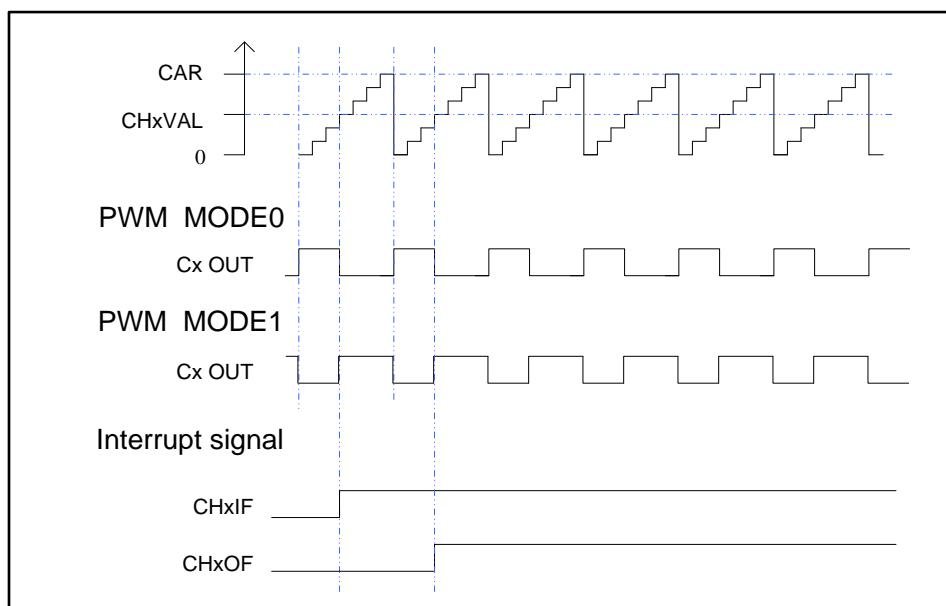
根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx\_CAR 寄存器值决定，占空比由 TIMERx\_CHxCV 寄存器值决定。[图14-79. PWM时序图](#)显示了 PWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx\_CHxCV 寄存器的值大于 TIMERx\_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx\_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 14-79. PWM 时序图



### 通道输出准备信号

根据 [图14-77. 通道输出比较原理（带有互补输出的通道， \$x=0\$ ）](#) 所示，当  $\text{TIMER}_x$  用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号  $\text{OxCPRE}$  信号(通道  $x$  输出准备信号)。设置  $\text{CH}_x\text{COMCTL}$  位可以定义  $\text{OxCPRE}$  信号类型。 $\text{OxCPRE}$  信号有若干类型的输出功能，包括，设置  $\text{CH}_x\text{COMCTL}=0x00$  可以保持原始电平；设置  $\text{CH}_x\text{COMCTL}=0x01$  可以将  $\text{OxCPRE}$  信号设置为高电平；设置  $\text{CH}_x\text{COMCTL}=0x02$  可以将  $\text{OxCPRE}$  信号设置为低电平；设置  $\text{CH}_x\text{COMCTL}=0x03$ ，在计数器值和  $\text{TIMER}_x\_\text{CH}_x\text{CV}$  寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是  $\text{OxCPRE}$  的另一种输出类型，设置  $\text{CH}_x\text{COMCTL}$  位域为  $0x06$  或  $0x07$  可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和  $\text{TIMER}_x\_\text{CH}_x\text{CV}$  寄存器值的关系以及计数方向， $\text{OxCPRE}$  信号改变其电平。具体细节描述，请参考相应的位。

设置  $\text{CH}_x\text{COMCTL}=0x04$  或  $0x05$  可以实现  $\text{OxCPRE}$  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  $\text{TIMER}_x\_\text{CH}_x\text{CV}$  的值和计数器值之间的比较结果。

设置  $\text{CH}_x\text{COMCEN}=1$ ，当由外部 ETI 引脚信号产生的 ETIFP 信号为高电平时， $\text{OxCPRE}$  被强制为低电平。在下一次更新事件到来时， $\text{OxCPRE}$  信号才会回到有效电平状态。

### 通道输出互补 PWM

$\text{CH}_x\text{O}$  和  $\text{CH}_x\text{ON}$  是一对互补输出通道，这两个信号不能同时有效。 $\text{TIMER}_x$  只有一路有互补输出通道。互补信号  $\text{CH}_x\text{O}$  和  $\text{CH}_x\text{ON}$  是由一组参数来决定： $\text{TIMER}_x\_\text{CHCTL2}$  寄存器中的  $\text{CH}_x\text{EN}$  和  $\text{CH}_x\text{NEN}$  位， $\text{TIMER}_x\_\text{CCHP}$  寄存器中和  $\text{TIMER}_x\_\text{CTL1}$  寄存器中的  $\text{POEN}$ ,  $\text{ROS}$ ,  $\text{IOS}$ ,  $\text{ISOx}$  和  $\text{ISOxN}$  位。输出极性由  $\text{TIMER}_x\_\text{CHCTL2}$  寄存器中的  $\text{CH}_x\text{P}$  和  $\text{CH}_x\text{NP}$  位来决定。

表 14-7. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出禁用.	
			1	0	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出使能. 如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
		1	0	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出使能. 如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
			1	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON 输出禁用.	
				1	CHx_O = LOW CHx_O 输出禁用.	CHx_ON=OxCPRE⊕CHxNP CHx_ON 输出使能
1	0/1	0	0	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁用.
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON=(!OxCPRE)⊕ CHxNP CHx_ON输出使能
			1	0	CHx_O = CHxP CHx_O输出禁用.	CHx_ON = CHxNP CHx_ON输出禁用.
				1	CHx_O = CHxP CHx_O输出使能	CHx_ON=OxCPRE⊕CHxNP CHx_ON输出使能
		1	0	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能
				1	CHx_O=OxCPRE ⊕ CHxP	CHx_ON=(!OxCPRE)⊕ CHxNP

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
					CHx_O输出使能	CHx_ON输出使能

### 互补 PWM 插入死区时间

设置CHxEN和CHxNEN为1'b1同时设置POEN，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对通道0有效。死区时间的细节，请参考TIMERx\_CCHP寄存器。

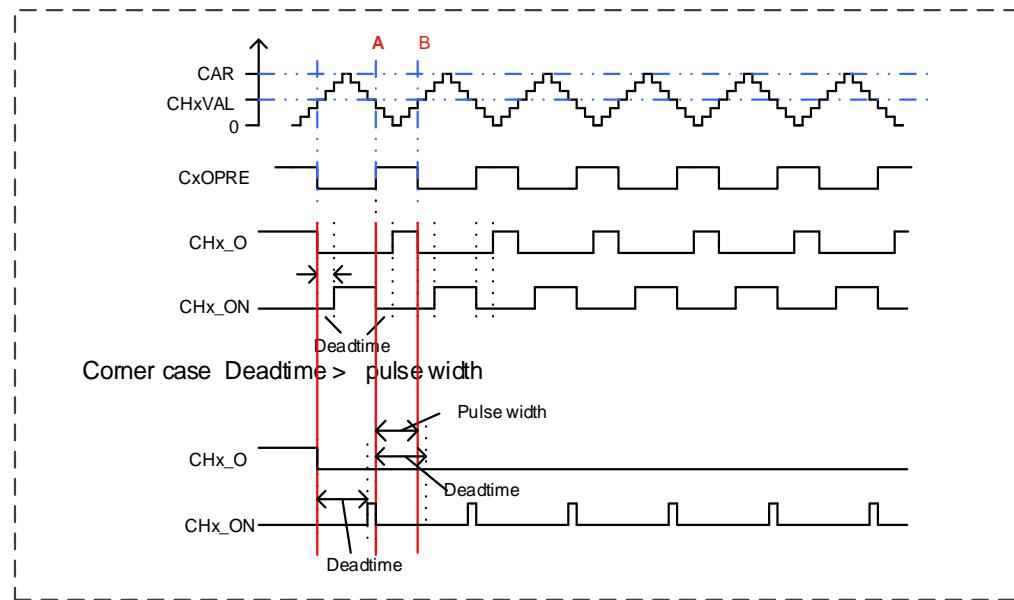
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在PWM0模式，当通道x匹配发生时（TIMERx计数器= CHxVAL），OxCOPRE反转。在[图14-80. 带死区时间的互补输出](#)中的A点，CHx\_O信号在死区时间内为低电平，直到死区时间过后才变为高电平，而CHx\_ON信号立刻变为低电平。同样，在B点，计数器再次匹配（TIMERx计数器= CHxVAL），OxCOPRE信号被清0，CHx\_O信号被立即清零，CHx\_ON信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于CHx\_O信号的占空比，CHx\_O信号一直为无效值（如[图14-80. 带死区时间的互补输出](#)）。
- 如果死区延时大于或者等于CHx\_ON信号的占空比，CHx\_ON信号一直为无效值。

图 14-80. 带死区时间的互补输出



### 中止模式

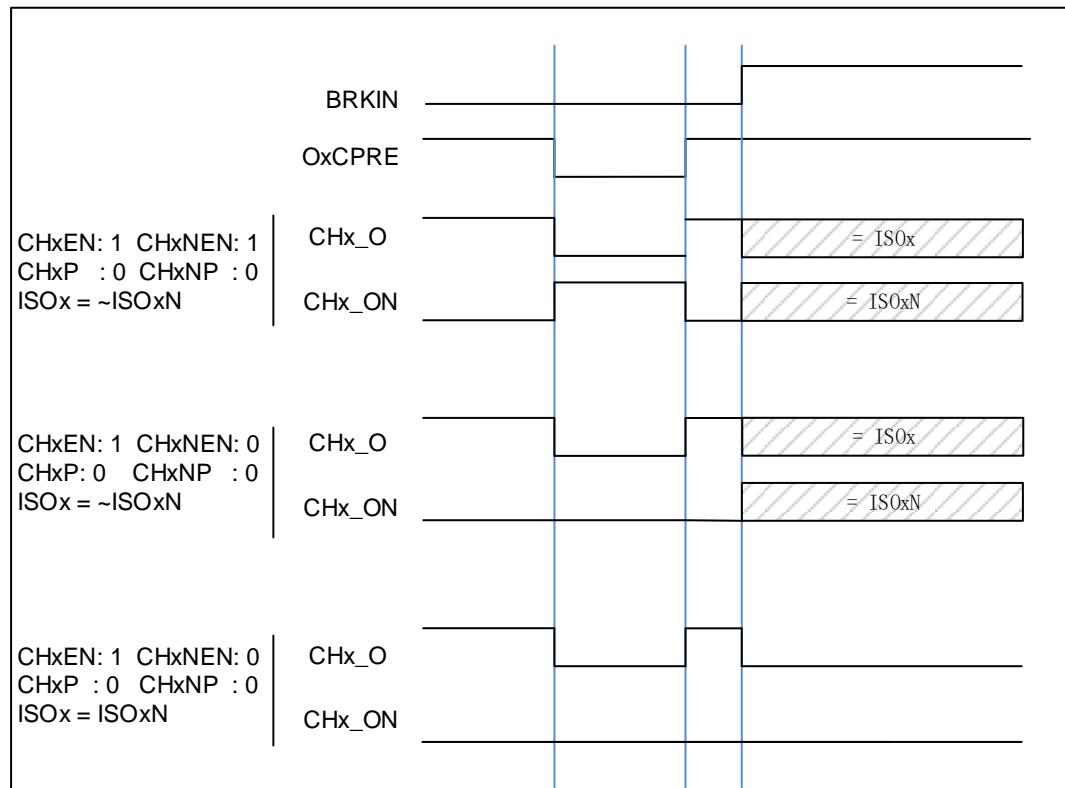
使用中止模式时，输出CHx\_O和CHx\_ON信号电平被以下位控制，TIMERx\_CCHP寄存器的POEN, IOS和ROS位，TIMERx\_CTL1寄存器的ISOx 和ISOxN位。当中止事件发生时，CHx\_O和CHx\_ON信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可

以选择 HXTAL 时钟失效事件。时钟失效事件由 RCU 中的时钟监视器 (CKM) 产生。将 TIMERx\_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx\_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx\_O 和 CHx\_ON 被 TIMERx\_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。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\_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 14-81. 通道响应中止输入（高电平有效）时，输出信号的行为



### 单脉冲模式

单脉冲模式与重复模式是相反的，设置 TIMERx\_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

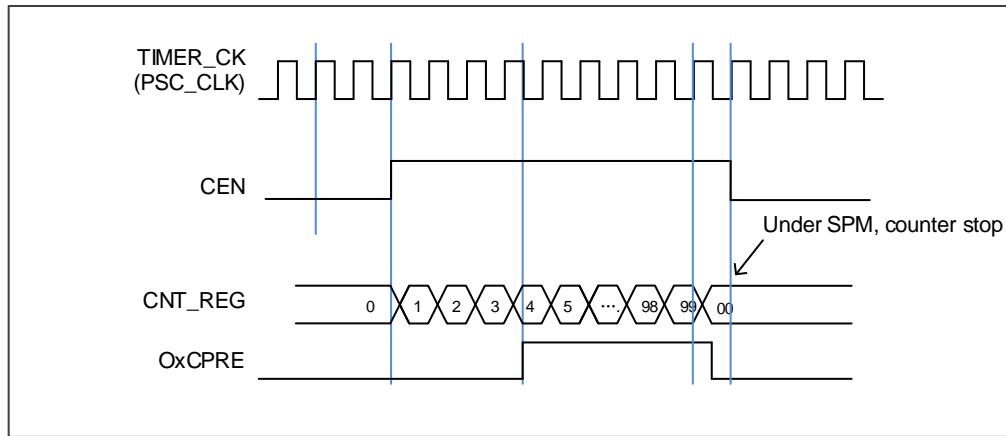
一旦设置定时器运行在单脉冲模式下，需要设置 TIMERx\_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx\_CHxCV 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\_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后，OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较

结果。只有输出通道配置为 PWM0 或 PWM1 输出运行模式下时 CH<sub>x</sub>COMFEN 位才可用，触发源来源于触发信号。

[图 14-82. 单脉冲模式, TIMER<sub>x</sub>\\_CH<sub>x</sub>CV = 4 TIMER<sub>x</sub>\\_CAR=99](#) 展示了一个例子。

图 14-82. 单脉冲模式, TIMER<sub>x</sub>\_CH<sub>x</sub>CV = 4 TIMER<sub>x</sub>\_CAR=99



## 定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMER<sub>x</sub>\_DMACFG和TIMER<sub>x</sub>\_DMATB。当然，必须要使能DMA请求，一些内部中断事件可以产生DMA请求。当中断事件发生，TIMER<sub>x</sub>会给DMA发送请求。DMA配置成M2P模式，PADDR是TIMER<sub>x</sub>\_DMATB寄存器地址，DMA就会访问TIMER<sub>x</sub>\_DMATB寄存器。实际上，TIMER<sub>x</sub>\_DMATB寄存器只是一个缓冲，定时器会将TIMER<sub>x</sub>\_DMATB映射到一个内部寄存器，这个内部寄存器由TIMER<sub>x</sub>\_DMACFG寄存器中的DMATA来指定。如果TIMER<sub>x</sub>\_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器的发送1个DMA请求就可以完成。如果TIMER<sub>x</sub>\_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMER<sub>x</sub>\_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4, DMATA+0x8, DMATA+0xc寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件，TIMER<sub>x</sub>将会重复上面的过程。

## 定时器调试模式

当Cortex®-M23内核停止，DBG\_CTL1寄存器中的TIMER<sub>x</sub>\_HOLD配置位被置1，定时器计数器停止。

### 14.5.5. TIMERx 寄存器(x=15,16)

TIMER15基地址: 0x4001 4400

TIMER16基地址: 0x4001 4800

#### 控制寄存器 0 (TIMERx\_CTL0)

地址偏移: 0x00

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
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。</p> <p>00: <math>f_{DTS}=f_{CK\_TIMER}</math></p> <p>01: <math>f_{DTS}= f_{CK\_TIMER} /2</math></p> <p>10: <math>f_{DTS}= f_{CK\_TIMER} /4</math></p> <p>11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能 TIMERx_CAR 寄存器的影子寄存器。</p> <p>1: 使能TIMERx_CAR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: 单脉冲模式禁能。更新事件发生后，计数器继续计数</p> <p>1: 单脉冲模式使能。在下一次更新事件发生时，计数器停止计数</p>
2	UPS	<p>更新请求源</p> <p>软件配置该位，选择更新事件源。</p> <p>0: 以下事件均会产生更新中断或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>复位模式产生的更新</li> </ul> <p>1: 下列事件会产生更新中断或DMA请求：</p>

计数器溢出/下溢

1	<b>UPDIS</b>	<p>禁止更新.</p> <p>该位用来使能或禁能更新事件的产生</p> <p>0: 更新事件使能. 更新事件发生时, 相应的影子寄存器被装入预装载值, 以下事件均会产生更新事件:</p> <ul style="list-style-type: none"> <li><b>UPG位被置1</b></li> <li>计数器溢出/下溢</li> <li>复位模式产生的更新</li> </ul> <p>1: 更新事件禁能.</p> <p>注意: 当该位被置1时, UPG位被置1或者复位模式不会产生更新事件, 但是计数器和预分频器被重新初始化</p>
0	<b>CEN</b>	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</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。</p>

### **控制寄存器 1 (TIMERx\_CTL1)**

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
31:10	保留	必须保持复位值
9	<b>ISOON</b>	<p>通道0的互补通道空闲状态输出</p> <p>0: 当POEN复位, CH0_ON设置低电平.</p> <p>1: 当POEN复位, CH0_ON设置高电平</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.</p>
8	<b>ISOO</b>	<p>通道0的空闲状态输出</p> <p>0: 当POEN复位, CH0_O设置低电平</p> <p>1: 当POEN复位, CH0_O设置高电平</p> <p>如果CH0_ON生效, 一个死区时间后CH0_O输出改变. 此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.</p>
7:4	保留	必须保持复位值

3	DMAS	DMA请求源选择 0: 当通道捕获/比较事件发生时, 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	换相控制影子寄存器更新控制 当换相控制影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 使能(CCSE=1), 这些影子寄存器更新控制如下: 0: CMTG位被置1时更新影子寄存器 1: 当CMTG位被置1或检测到TRIGI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	换相控制影子使能 0: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位禁能. 1: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位使能. 如果这些位已经被写入了, 换相事件到来时这些位才被更新 当通道没有互补输出时, 此位无效

### DMA 和中断使能寄存器 (**TIMERx\_DMAINTEN**)

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					CH0DEN	UPDEN	BRKIE	保留		CMTIE	保留			CHOIE	UPIE
rw								rw				rw			

位/位域	名称	描述
31:10	保留	必须保持复位值.
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁止通道 0 比较/捕获 DMA 请求 1: 使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7	BRKIE	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	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4:2	保留	必须保持复位值.
1	CHOIE	通道 0 比较/捕获中断使能 0: 禁止通道 0 中断 1: 使能通道 0 中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

### 中断标志寄存器 (TIMERx\_INTF)

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
31:10	保留	必须保持复位值.
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时, 在 CHOIF 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: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 当中止输入有效时, 由硬件对该位置 '1'。 当中止输入无效时, 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
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1, 此位由软件清 0。

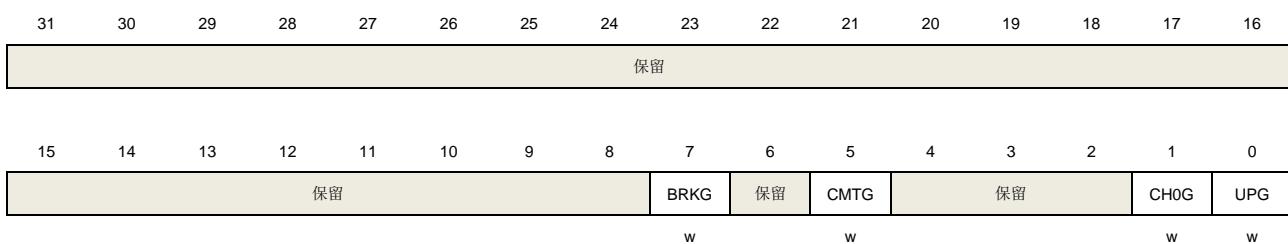
		0: 无通道换相更新中断发生 1: 通道换相更新中断发生
4:2	保留	必须保持复位值.
1	CH0IF	通道 0 比较/捕获中断标志  此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。  0: 无通道 0 中断发生 1: 通道 0 中断发生
0	UPIF	更新中断标志  此位在任何更新事件发生时由硬件置 1，软件清 0。  0: 无更新中断发生 1: 发生更新中断

### 软件事件产生寄存器 (**TIMERx\_SWEVG**)

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
31:8	保留	必须保持复位值.
7	BRKG	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，POEN位被清0且BRKIF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
5	CMTG	通道换相更新事件发生  此位由软件置1，由硬件自动清0. 当此位被置1，通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和CHxCOMCTL) 的互补输出被更新（根据TIMERx_CTL1中CCSE值）。  0: 不产生通道控制更新事件 1: 产生通道控制更新事件
4:2	保留	必须保持复位值.

1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF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0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，在向上计数模式中，计数器被清0，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

### 通道控制寄存器 0 (TIMERx\_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



#### 输出比较模式:

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。
3:0	CH0CAPPSC[1:0]	

100: 强制为低。强制 O0CPRE 为低电平

101: 强制为高。强制 O0CPRE 为高电平

110: PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx\_CH0CV 时，O0CPRE 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\_CH0CV 时，O0CPRE 为低电平，否则为高电平。

111: PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx\_CH0CV 时，O0CPRE 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\_CH0CV 时，O0CPRE 为高电平，否则为低电平。

如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。

当 TIMERx\_CCHP 寄存器的 PROT[1:0]=11 且 CH0MS=00（比较模式）时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 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 0 输出/比较影子寄存器 1: 使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM=1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较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，CH0_O 被设置为比较电平而与比较结果无关。 0: 禁止通道 0 输出比较快速。 1: 使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0) 时这些位才可写。 00: 通道 0 配置为输出 01: 通道 0 配置为输入，ISO 映射在 CI0FE0 上 10: 通道 0 配置为输入，ISO 映射在 CI1FE0 上 11: 通道 0 配置为输入，ISO 映射在 ITS 上。 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

#### 输入捕获模式：

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 fSAMP 对 CI0 输入信号进行连续采样，并记录信号相同

电平的次数。达到该位配置的滤波参数后，认为是有效电平。

滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	fSAMP
4'b0000	无滤波器	
4'b0001	2	fCK_TIMER
4'b0010	4	
4'b0011	8	
4'b0100	6	fDTS/2
4'b0101	8	
4'b0110	6	fDTS/4
4'b0111	8	
4'b1000	6	fDTS/8
4'b1001	8	
4'b1010	5	fDTS/16
4'b1011	6	
4'b1100	8	
4'b1101	5	fDTS/32
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] 通道0输入捕获预分频器

这2位定义了通道0输入的预分频系数。当TIMERx\_CHCTL2寄存器中的CH0EN =0时，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 CH0MS[1:0] 通道0模式选择

与输出比较模式相同

## 通道控制寄存器 2 (TIMERx\_CHCTL2)

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															
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
------	----	----

31:4	保留	必须保持复位值
3	CH0NP	<p>通道0互补输出极性</p> <p>当通道0配置为输出模式，此位定义了互补输出信号的极性。</p> <p>0: 通道0互补输出高电平为有效电平</p> <p>1: 通道0互补输出低电平为有效电平</p> <p>当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。</p>
2	CH0NEN	<p>通道0互补输出使能</p> <p>当通道0配置为输出模式时，将此位置1使能通道0的互补输出。</p> <p>0: 禁止通道0互补输出</p> <p>1: 使能通道0互补输出</p>
1	CH0P	<p>通道0极性</p> <p>当通道0配置为输出模式时，此位定义了输出信号极性。</p> <p>0: 通道0高电平为有效电平</p> <p>1: 通道0低电平为有效电平</p> <p>当通道0配置为输入模式时，此位定义了CI0信号极性</p> <p>[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性</p> <p>[CH0NP==0, CH0P==0]: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。</p> <p>[CH0NP==0, CH0P==1]: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。</p> <p>[CH0NP==1, CH0P==0]: 保留。</p> <p>[CH0NP==1, CH0P==1]: 把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。</p>
0	CH0EN	<p>通道0捕获/比较使能</p> <p>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。</p> <p>0: 禁止通道0</p> <p>1: 使能通道0</p>

### 计数器寄存器 (**TIMERx\_CNT**)

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
保留															

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	保留	必须保持复位值.
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。

### 预分频寄存器 (TIMERx\_PSC)

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															
PSC[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (TIMERx\_CAR)

地址偏移: 0x2C

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															
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
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。

**注意:** 在定时器被配置为输入捕获模式时, 该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

### 重复计数寄存器 (**TIMERx\_CREP**)

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								CREP[7:0]							

rw

位/位域	名称	描述
31:8	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)。

### 通道 0 捕获/比较寄存器 (**TIMERx\_CH0CV**)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 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 0 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。

### 互补通道保护寄存器 (TIMERx\_CCHP)

地址偏移: 0x44

复位值: 0x0000 0000

该寄存器通过字访问（32位）。

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

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	POEN	<p>所有的通道输出使能</p> <p>该位通过以下方式置 1:</p> <ul style="list-style-type: none"> <li>-写 1 置位</li> <li>-如果 OAEN=1，则在下一次更新事件发生时置 1.</li> </ul> <p>该位通过以下方式清 0:</p> <ul style="list-style-type: none"> <li>-写 0 清 0</li> <li>-有效的中止输入（异步）</li> </ul> <p>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_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。</p> <p>0: 禁止通道输出</p> <p>1: 使能通道输出</p> <p>注意：仅当 CHxMS[1:0]=2'b00 时该位有效</p>
14	OAEN	<p>自动输出使能</p> <p>0: POEN 位只能使用软件方式置 1。</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1。</p> <p>此位只有在 TIMERx_CCHP 寄存器的 PROT [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 BKIN 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	BRKEN	<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: 禁能中止输入。</p> <p>1: 使能中止输入。</p> <p>此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。</p>

11	ROS	运行模式下“关闭状态”配置 当 POEN 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 POEN 位被置 1，通道输出信号（CHx_O/ CHx_ON）被禁止。 1: 当 POEN 位被置 1，通道输出信号（CHx_O / CHx_ON）被使能，和 TIMER0_CHCTL2 寄存器 CHxEN/CHxNEN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [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 POEN 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 POEN 位被清 0，通道输出信号（CHx_O/ CHx_ON）被禁止。 1: 当 POEN 位被清 0，通道输出信号（CHx_O / CHx_ON）被使能，和 TIMERx_CHCTL2 寄存器 CHxEN/CHxNEN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00: 禁能保护模式。无写保护。 01: PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。 10: PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位。 11: PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。 系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。
7:0	DTCFG[7:0]	死区时间控制 DTCFG 值和死区时间的关系如下：

DTCFG[7:5]	The duration of dead-time
3'b0xx	DTCFG[7:0] * tDTS_CK
3'b10x	(64+ DTCFG[5:0]) * tDTS_CK *2
3'b110	(32+ DTCFG[4:0]) * tDTS_CK *8
3'b111	(32+ DTCFG[4:0]) * tDTS_CK *16

注意：

1. tDTS\_CK 是 DTS\_CK 的周期，由 TIMERx\_CTL0 中的 CKDIC[1:0] 定义。
2. 此位只有在 TIMERx\_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

### DMA 配置寄存器 (TIMERx\_DMACFG)

地址偏移：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
保留				DMATC[4:0]				保留				DMATA [4:0]			
rw															rw

位/位域	名称	描述
31:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMATB寄存器的数量n, n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当通过TIMERx_DMA第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问起始地址+0x4。

### **DMA 发送缓冲区寄存器 (TIMERx\_DMATB)**

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	DMATB [15:0]	DMA发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为0到DMATC。

### **配置寄存器 (TIMERx\_CFG)**

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
														CHVSEL	OUTSEL

rw rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	<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: 无影响</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POEN位与IOS位均为0，则输出无效 0: 无影响</p>

## 14.6. 基本定时器 (TIMERx, x=5)

### 14.6.1. 简介

基本定时器(TIMER5)包含一个无符号16位计数器。基本定时器可以配置产生DMA请求。

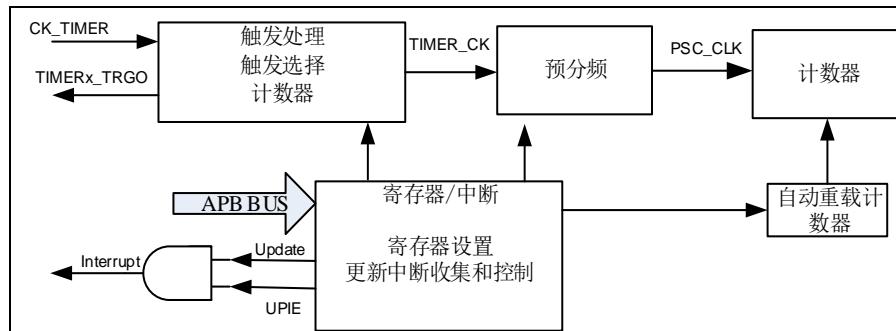
### 14.6.2. 主要特征

- 计数器宽度: 16位;
- 定时器时钟源只有内部时钟;
- 计数模式: 向上计数;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 自动重装载功能;
- 中断输出和DMA请求: 更新事件。

### 14.6.3. 结构框图

[图14-83. 基本定时器结构框图](#)提供了基本定时器内部配置的细节。

图 14-83. 基本定时器结构框图



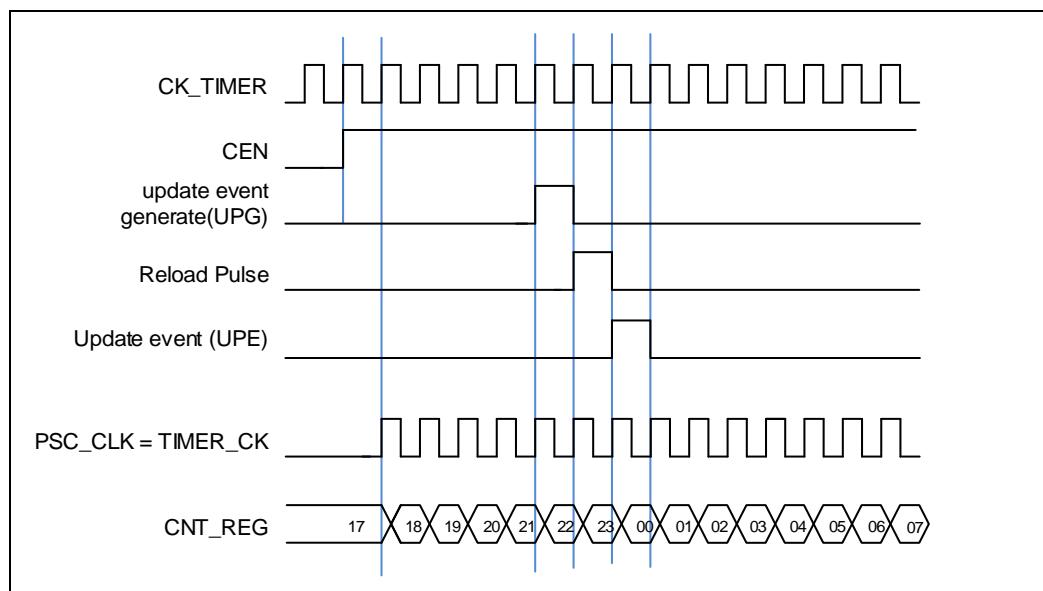
### 14.6.4. 功能说明

#### 时钟源配置

基本定时器可以是内部时钟源CK\_TIMER驱动。

基本定时器仅有一个时钟源CK\_TIMER, 用来驱动计数器预分频器。当CEN置位, CK\_TIMER 经过预分频器 (预分频值由TIMERx\_PSC寄存器确定) 产生PSC\_CLK。

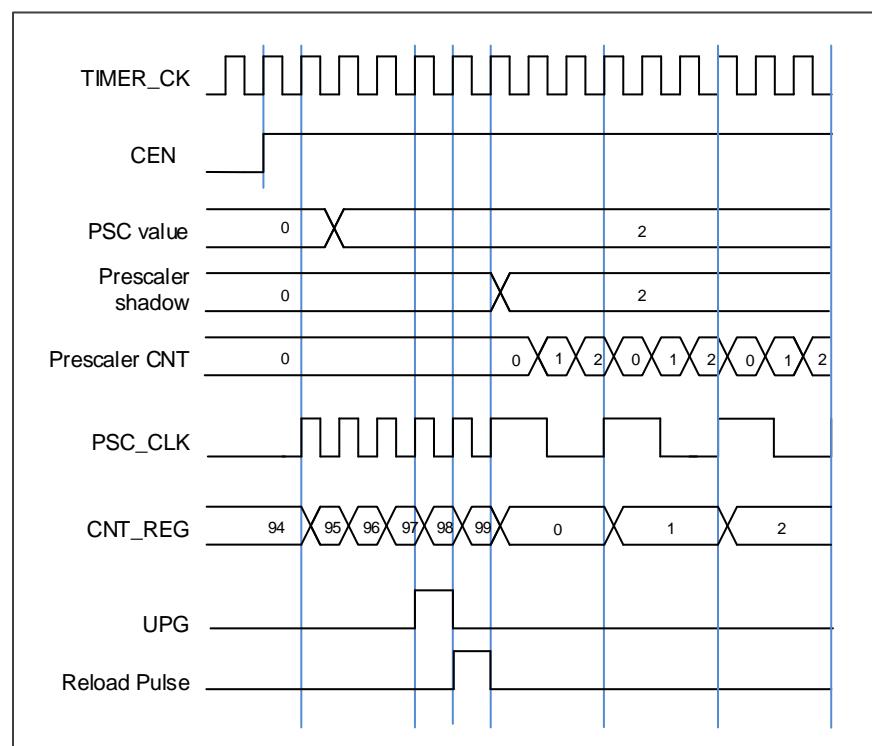
图 14-84. 内部时钟分频为 1 时, 计数器的时序图



### 时钟预分频器

预分频器可以将定时器的时钟 (TIMER\_CK) 频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC\_CLK 驱动计数器计数。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 14-85. 当 PSC 数值从 0 变到 2 时, 计数器的时序图



## 计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx\_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx\_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx\_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx\_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

[图 14-86. 向上计数时序图，PSC=0/2](#) 和 [图 14-87. 向上计数时序图，在运行时改变 TIMERx CAR 寄存器的值](#)给出了一些例子，当 TIMERx\_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 14-86. 向上计数时序图，PSC=0/2

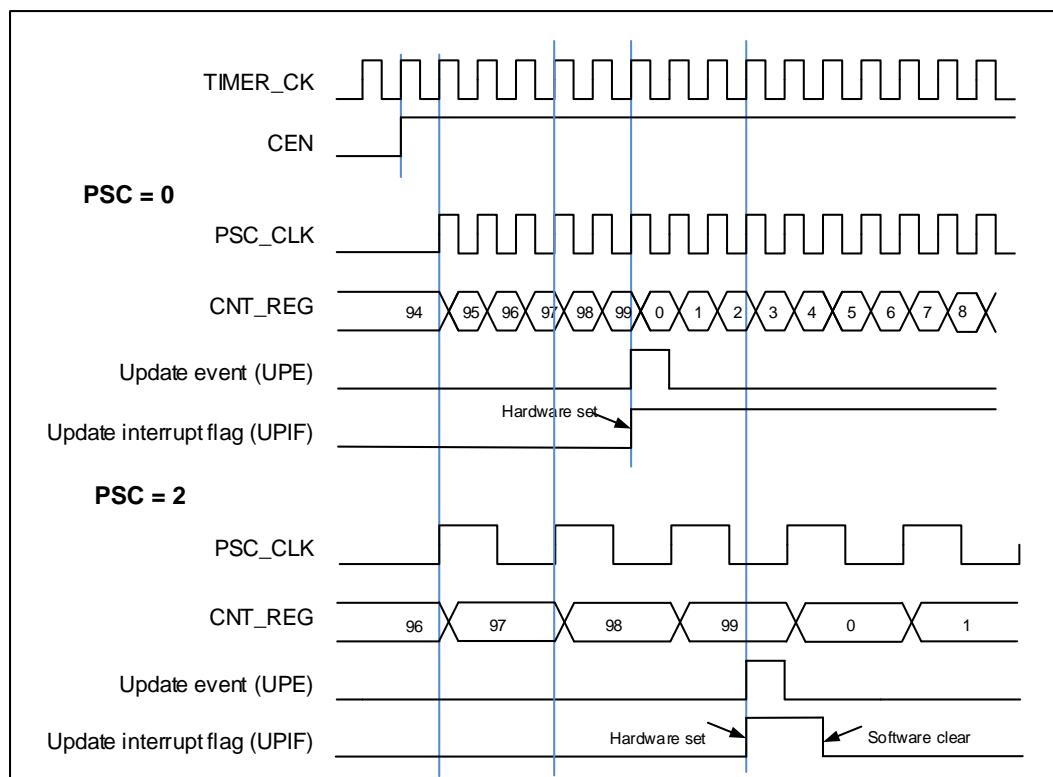
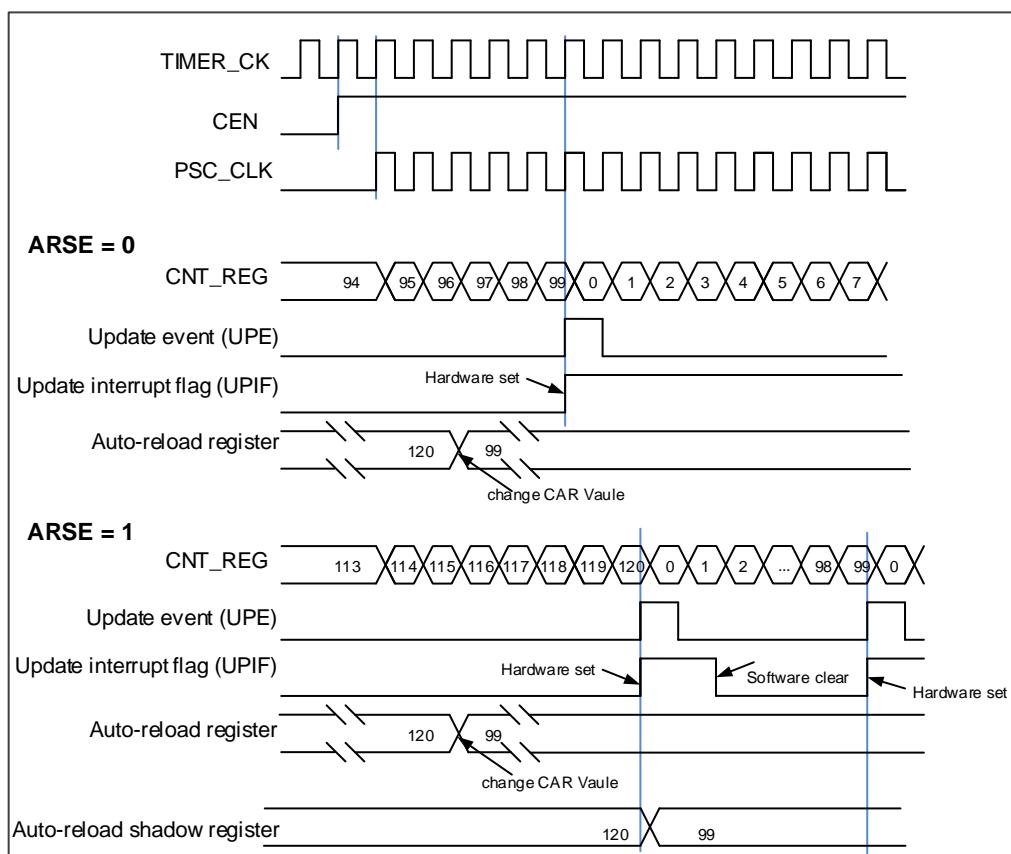


图 14-87. 向上计数时序图，在运行时改变 TIMERx\_CAR 寄存器的值



### 单脉冲模式

单脉冲模式与重复模式是相反的，设置TIMERx\_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。

一旦设置定时器运行在单脉冲模式下，需要设置TIMERx\_CTL0寄存器的定时器使能位CEN=1来使能计数器，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。

### 定时器调试模式

当Cortex®-M23内核停止，DBG\_CTL0寄存器中的TIMERx\_HOLD配置位被置1，定时器计数器停止。

### 14.6.5. TIMERx 寄存器(x=5)

TIMER5基地址: 0x4000 1000

#### 控制寄存器 0 (TIMERx\_CTL0)

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								ARSE	保留				SPM	UPS	UPDIS	CEN

rw                                rw                                rw                                rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7	ARSE	自动重载影子使能 0: 禁能TIMERx_CAR寄存器的影子寄存器 1: 使能TIMERx_CAR寄存器的影子寄存器
6:4	保留	必须保持复位值
3	SPM	单脉冲模式 0: 单脉冲模式禁能。更新事件发生后，计数器继续计数 1: 单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源。 0: 以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1: 下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新。 该位用来使能或禁能更新事件的产生 0: 更新事件使能。更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新

1: 更新事件禁能。

注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

0	CEN	计数器使能
		0: 计数器禁能
		1: 计数器使能

### 控制寄存器 1 (TIMERx\_CTL1)

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								MMC[2:0]				保留			

rw

位/位域	名称	描述
31:7	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，TRGO信号由主定时器发给从定时器用于同步功能 000: 当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001: 当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010: 当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和 UPS位决定
3:0	保留	必须保持复位值

### DMA 和中断使能寄存器 (TIMERx\_DMAINTEN)

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								UPDEN	保留								UPIE
								rw									rw

位/位域	名称	描述
31:9	保留	必须保持复位值
8	UPDEN	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

### 中断标志寄存器 (TIMERx\_INTF)

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															

位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPIF	更新中断标志 此位在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

### 软件事件产生寄存器 (TIMERx\_SWEVG)

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

UPG

w

位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPG	<p>更新事件产生</p> <p>此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，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>

### 计数器寄存器 (TIMERx\_CNT)

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

保留

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	保留	必须保持复位值.
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。

### 预分频寄存器 (TIMERx\_PSC)

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

PSC[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 <b>TIMER_CK</b> 时钟除以(PSC+1), 每次当更新事件产生时, PSC 的值被装入到对应的影子寄存器。

### 计数器自动重载寄存器 (**TIMERx\_CAR**)

地址偏移: 0x2C

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值.
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。

## 15. 红外线接口 (IFRP)

### 15.1. 简介

红外线接口 (IFRP) 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TIMER15定时器和TIMER16定时器控制。通过设置GPIO引脚为高速模式，可以提高模块的输出电流能力。

### 15.2. 主要特性

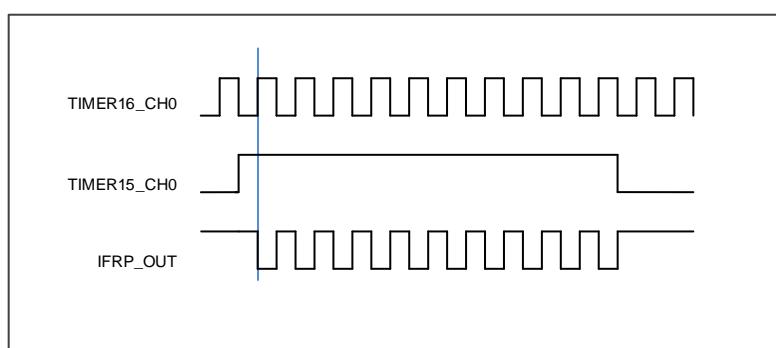
- IFRP输出信号由TIMER15定时器的通道0和TIMER16定时器的通道0决定
- 为了获取正确的红外信号，TIMER15定时器应该产生低频调制包络信号，TIMER16应该产生高频载波信号
- 通过设置SYSCFG\_CFG0中的PB9\_HCCE，红外线接口输出（PB9）能够提供高电流输出驱动LED接口

### 15.3. 功能描述

IFRP 模块整合了 TIMER15 定时器和 TIMER16 定时器的输出来产生红外信号。

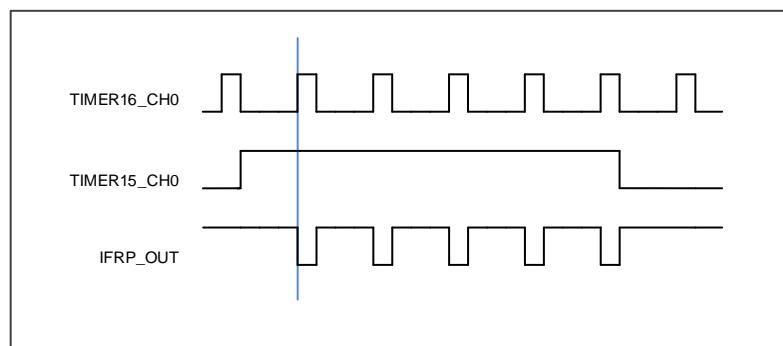
1. 通过对 TIMER15 定时器的通道 0 编程输出低频 PWM 信号来产生调制包络信号，对 TIMER16 定时器的通道 0 编程输出高频 PWM 信号来产生载波信号。产生信号之前需要开启这些通道。
2. 配置 GPIO 口为复用，并使能这些引脚。
3. 如果你想获得更高的驱动电流输出，需要将 IFRP\_OUT 映射到 PB9 口上，并且在 SYS\_CFG 模式下通过相应寄存器设置 PB9 口为高速模式。

图 15-1. IFRP 输出时序图 1



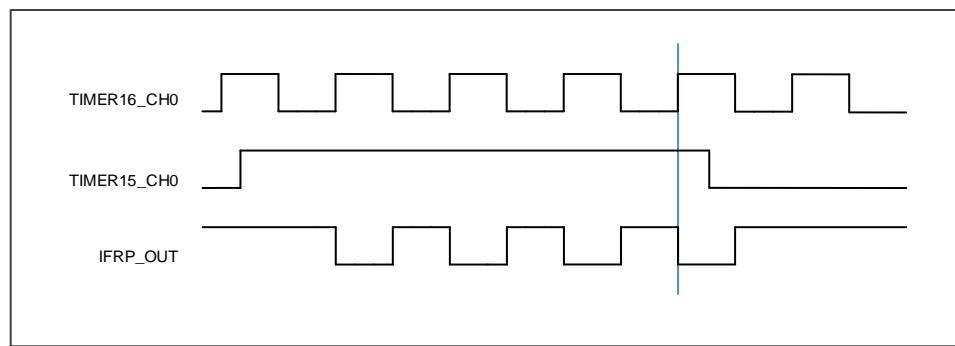
注：IFRP\_OUT比TIMER16\_CH0时钟通道1输出信号有一个APB时钟延迟。

图 15-2. IFRP 输出时序图 2



注：载波（TIMER15\_CH0）的占空比可以被改变，在TIMER15定时器的通道0输出电平为高时，IFRP\_OUT和TIMER16定时器的通道0输出信号反相。

图 15-3. IFRP 输出时序图 3



注：IFRP\_OUT将保持TIMER16定时器通道0输出信号的完整，即使TIMER15定时器的包络信号无效。

## 16. 通用同步异步收发器（USART）

### 16.1. 简介

通用同步异步收发器（USART）提供了一个灵活方便的串行数据交换接口。数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART提供了可编程的波特率发生器，能对能对UCLK（PCLK1, PCLK2以及仅USART0可用的CK\_USART0）进行分频产生USART发送和接收所需的特定频率。

USART不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，半双工以及同步模式。它还支持多处理器通信和Modem流控操作（CTS/RTS）。数据帧支持从LSB或者MSB开始传输。数据位的极性和TX/RX引脚都可以灵活配置。

所有USART都支持DMA功能，以实现高速率的数据通信。

### 16.2. 主要特性

- NRZ标准格式
- 全双工异步通信
- 半双工单线通信
- 接收FIFO功能
- 双时钟域：
  - 互为异步关系的APB时钟和USART时钟
  - 不依赖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，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，无校验位的生成或检测
  - 产生0.5, 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
- 支持硬件Modem流控操作（CTS/RTS）和RS485驱动使能
- 可配置的多级缓存通信DMA访问数据缓冲区
- 发送器和接收器可分别使能
- 奇偶校验位控制：
  - 发送奇偶校验位
  - 检测接收的数据字节的奇偶校验位
- LIN断开帧的产生和检测
- 支持红外数据协议（IrDA）
- 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的智能卡接口：

- 字节模式 ( $T=0$ )
- 块模式 ( $T=1$ )
- 直接和反向转换
- 多处理器通信:
  - 如果地址不匹配，则进入静默模式
  - 通过线路空闲检测或者地址匹配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:
  - 通过标准的RBNE中断
  - 通过WUF中断
- 多种状态标志:
  - 传输检测标志：接收缓冲区不为空 (RBNE)，接收FIFO满 (RFF)，发送缓冲区为空 (TBE)，传输完成 (TC)
  - 错误检测标志：过载错误 (ORERR)，噪声错误 (NERR)，帧格式错误 (FERR)，奇偶校验错误 (PERR)
  - 硬件流控操作标志：CTS变化 (CTSF)
  - LIN模式标志：LIN断开检测 (LBDF)
  - 多处理器通信模式标志：IDLE帧检测 (IDLEF)
  - ModBus通信标志：地址/字符匹配 (AMF)，接收超时 (RTF)
  - 智能卡模式标志：块结束 (EBF) 和接收超时 (RTF)
  - 从深度睡眠模式唤醒标志
  - 若相应的中断使能，这些事件发生将会触发中断

USART0完全实现上述功能，但是USART1只实现了上面所介绍的部分功能，下面这些功能在USART1中没有实现：

- 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

### 16.3. 功能描述

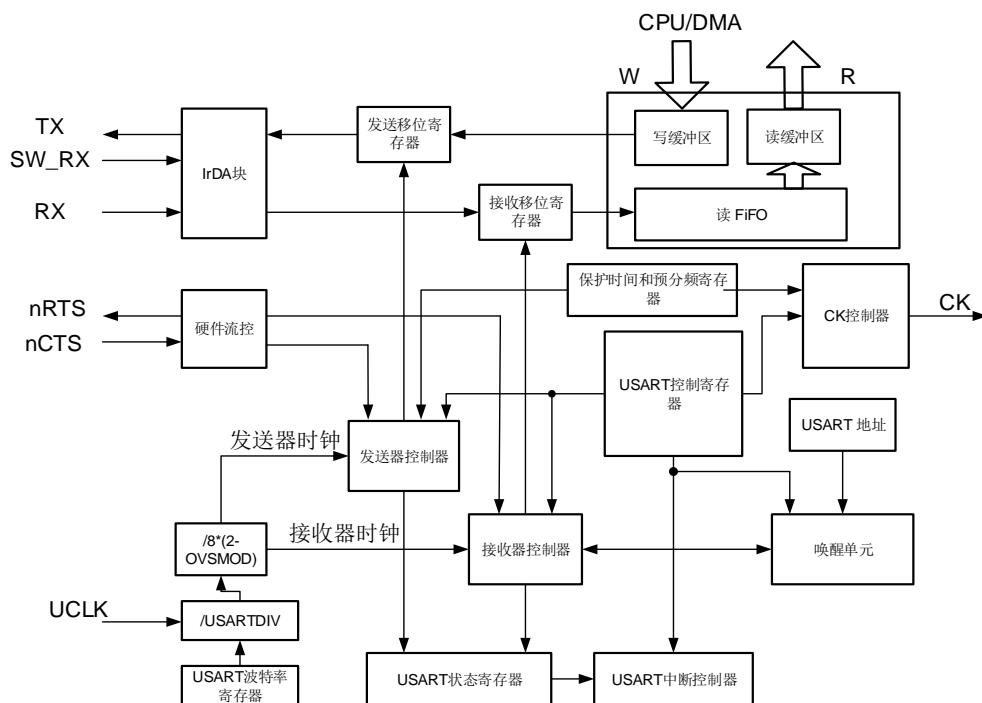
USART 接口通过[表 16-1. USART 重要引脚描述](#)中主要引脚从外部连接到其他设备。

**表 16-1. USART 重要引脚描述**

引脚	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

引脚	类型	描述
CK	输出	用于同步通信的串行时钟信号
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

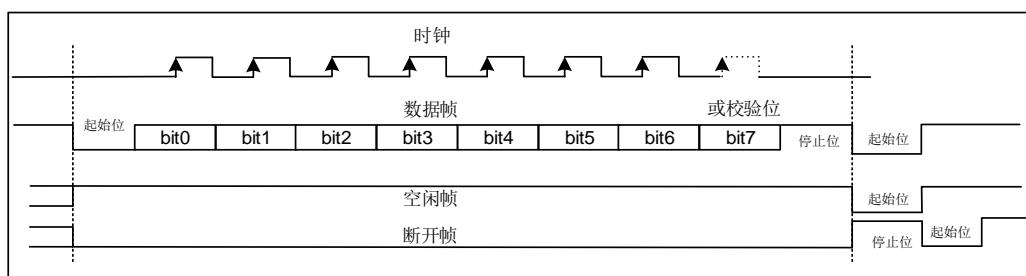
图 16-1. USART 模块内部框图



### 16.3.1. USART 帧格式

USART数据帧开始于起始位，结束于停止位。USART\_CTL0寄存器中WL位可以设置数据长度。将USART\_CTL0寄存器中PCEN置位，最后一个数据位可以用作校验位。若WL位为0，第七位为校验位。若WL位置1，第八位为校验位。USART\_CTL0寄存器中PM位用于选择校验位的计算方法。

图16-2. USART字符帧（8数据位和1停止位）



在发送和接收中，停止位可以在USART\_CTL1寄存器中STB[1:0]位域中配置。

表 16-2. 停止位配置

STB[1:0]	停止位长度 (位)	功能描述
00	1	默认值
01	0.5	智能卡模式接收
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

在一个空闲帧中，所有位都为1。数据帧长度与正常USART数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART数据帧的传输速度由UCLK时钟频率，波特率发生器的配置，以及过采样模式共同决定。

### 16.3.2. 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部分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UCLK具有如下关系：

如果过采样率是16，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (\text{式16-1})$$

如果过采样是8，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (\text{式16-2})$$

例如，当过采样是16：

- 由USART\_BAUD寄存器的值得到USARTDIV：

假设USART\_BAUD=0x21D，则INTDIV=33（0x21），FRADIV=13（0xD）。

UASRTDIV=33+13/16=33.81。

- 由USARTDIV得到USART\_BAUD寄存器的值：

假设要求UASRTDIV=30.37，INTDIV=30（0x1E）

16\*0.37=5.92，接近整数6，所以FRADIV=6（0x6）

USART\_BAUD=0x1E6。

**注意：**若取整后FRADIV=16（溢出），则进位必须加到整数部分。

### 16.3.3. USART 发送器

如果USART\_CTL0寄存器的发送使能位（TEN）被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过USART\_CTL1寄存器中TINV位来配置。时钟脉冲通过CK引脚输出。

TEN置位后发送器会发出一个空闲帧。TEN位在数据发送过程中是不可以被复位的。

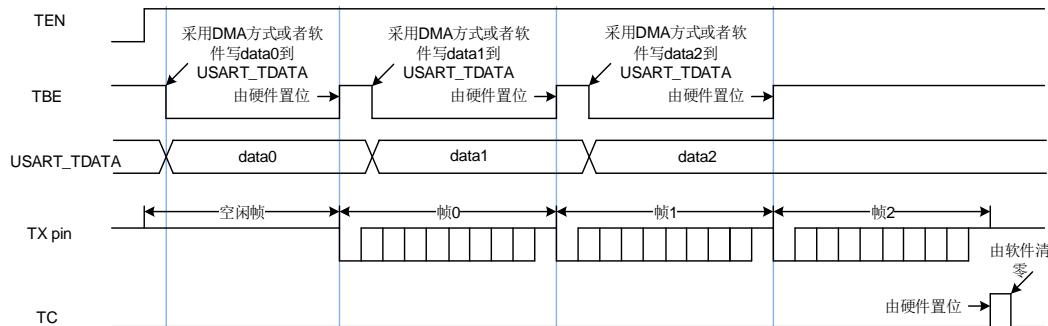
系统上电后，TBE默认为高电平。在USART\_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART\_TDATA寄存器。当数据写入USART\_TDATA寄存器，TBE位将被清0。在数据由USART\_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART\_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART\_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

假如一帧数据已经被发送出去，并且TBE位已被置位，那么USART\_STAT寄存器中TC位将被置1。如果USART\_CTL0寄存器中的中断使能位（TCIE）为1，将会产生中断。

**图 16-3. USART 发送步骤**给出了 USART 发送步骤。软件操作按以下流程进行：

1. 通过USART\_CTL0寄存器的WL设置字长；
2. 在USART\_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART\_CTL2寄存器中使能DMA（DENT位）；
4. 在USART\_BAUD寄存器中设置波特率；
5. 在USART\_CTL0寄存器中置位UEN位，使能USART；
6. 在USART\_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向USART\_TDATA寄存器写数据；
9. 若DMA未使能，每发送一个字节都需重复步骤7-8；
10. 等待TC=1，发送完成。

**图 16-3. USART 发送步骤**



在禁用USART或进入低功耗状态之前，必须等待TC置位。通过向USART\_INTC寄存器的TCC位写1可将TC位清零。

当SBKCMD置位时，会发送一个断开帧，发送完成后，SBKCMD将被清0。

#### 16.3.4. USART 接收器

上电后，按以下步骤使能USART接收器：

1. 写USART\_CTL0寄存器的WL位去设置字长；
2. 在USART\_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART\_CTL2寄存器中使能DMA（DENR位）；
4. 在USART\_BAUD寄存器中设置波特率；
5. 在USART\_CTL0寄存器中置位UEN位，使能USART；

## 6. 在USART\_CTL0中设置REN位。

接收器在使能后若检测到一个有效的起始脉冲便开始接收码流。在接收一个数据帧的过程中会检测噪声错误，奇偶校验错误，帧错误和过载错误。

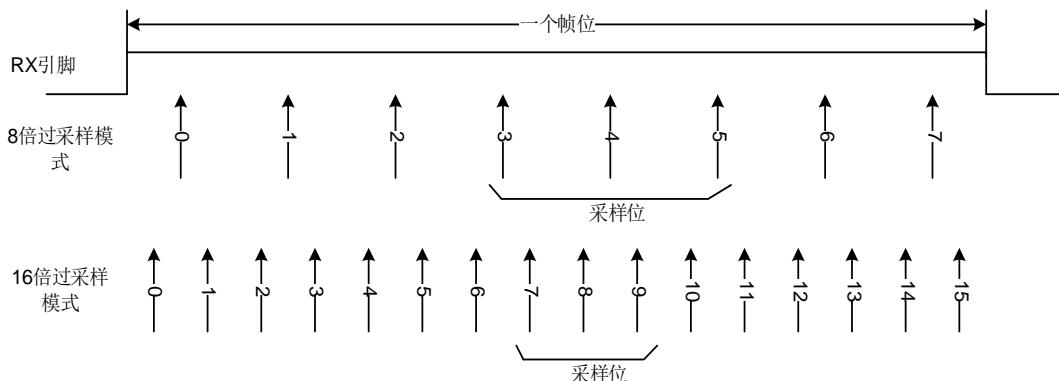
当接收到一个数据帧，USART\_STAT寄存器中的RBNE置位，如果设置了USART\_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART\_STAT寄存器中可以观察接收状态标志。

软件可以通过读USART\_RDATA寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对USART\_RDATA寄存器的一个读操作都可以清除RBNE位。

在接收过程中，需使能REN位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果是8倍过采样模式，选择第3、4、5个采样点；如果是16倍过采样模式，选择第7、8、9个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是起始位，数据位，奇偶校验位或者停止位，都将产生噪声错误（NERR）。如果使能DMA，并置位USART\_CTL2寄存器中ERRIE，将会产生中断。如果在USART\_CTL2中置位OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

**图 16-4. 过采样方式接收一个数据位（OSB=0）**



通过置位USART\_CTL0寄存器中的PCEN位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART\_STAT寄存器中PERR被置位。如果设置了USART\_CTL0寄存器中的PERRIE位，将产生中断。

如果在停止位传输过程中RX引脚为0，将产生帧错误，USART\_STAT寄存器中FERR置位。如果使能DMA并置位USART\_CTL2寄存器中ERRIE位，将产生中断。

当接收到一帧数据，而RBNE位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。USART\_STAT寄存器中的溢出错误标志位ORERR将置位。如果使能DMA并置位USART\_CTL2寄存器中ERRIE位或者置位RBNEIE，将产生中断。

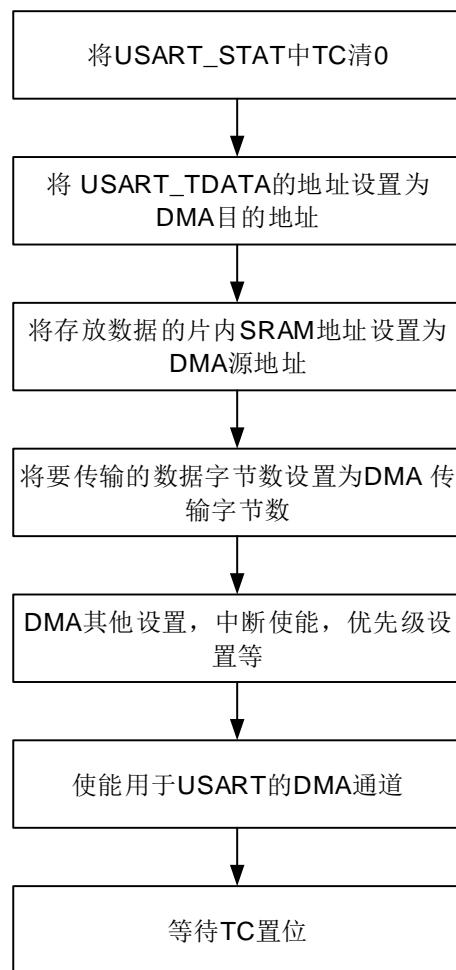
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR位将和RBNE位同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

### 16.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位USART\_CTL2寄存器中DENT位可以使能DMA发送，置位USART\_CTL2寄存器中DENR位可以使能DMA接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配置步骤如[图 16-5. 采用 DMA 方式实现 USART 数据发送配置步骤](#)所示。

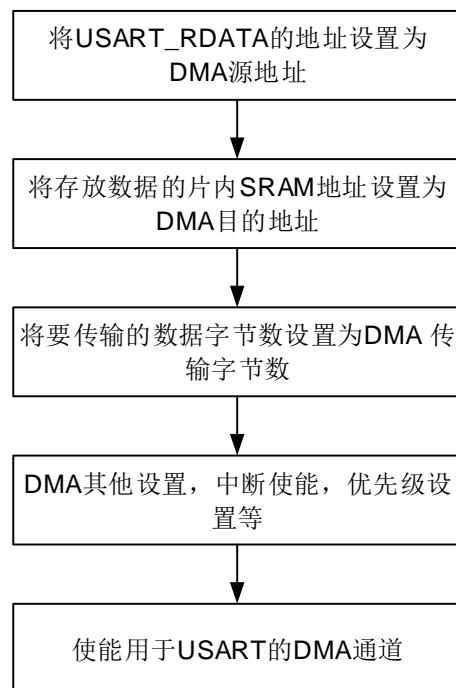
**图 16-5. 采用 DMA 方式实现 USART 数据发送配置步骤**



所有数据帧都传输完成后，USART\_STAT寄存器中TC位置1。如果USART\_CTL0寄存器中TCIE置位，将产生中断。

当 DMA 用于 USART 接收时，DMA 将数据从接收缓冲区传送到片内 SRAM。配置步骤如[图 16-6. 采用 DMA 方式实现 USART 数据接收配置步骤](#)所示。如果将 USART\_CTL2 寄存器中ERRIE 位置 1，USART\_STAT 寄存器中的错误标志位（FERR、ORERR 和 NERR）置位时将产生中断。

图 16-6. 采用 DMA 方式实现 USART 数据接收配置步骤

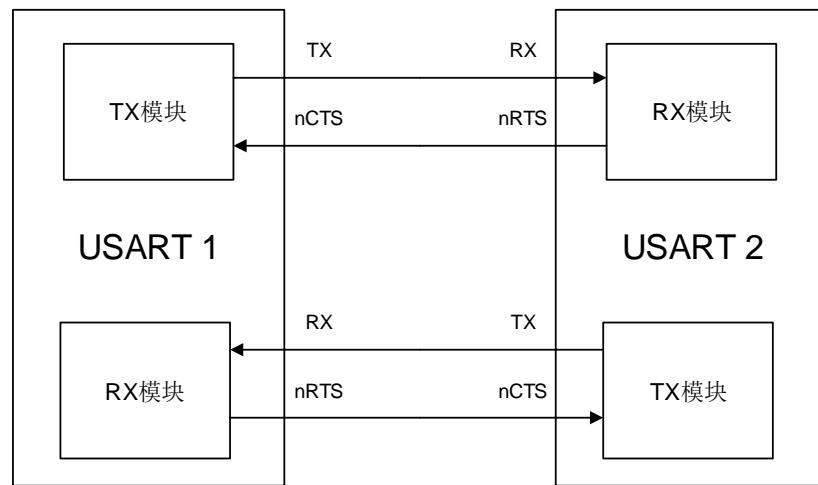


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

### 16.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将USART\_CTL2寄存器中RTSEN位置1来使能RTS流控，将USART\_CTL2寄存器中CTSEN位置1来使能CTS流控。

图 16-7. 两个 USART 之间的硬件流控制



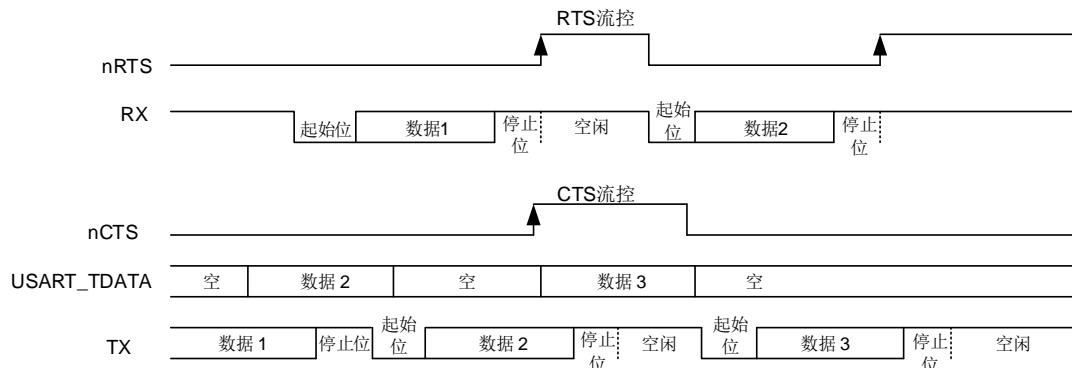
#### RTS 流控

USART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平。

## CTS 流控

USART发送器监视nCTS输入引脚来决定数据帧是否可以发送。如果USART\_STAT寄存器中TBE位是0且nCTS为低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

**图 16-8. 硬件流控制**



## RS485 驱动使能

驱动使能功能通过设置USART\_CTL2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\_CTL0控制器的DEA[4:0]位域中进行设置。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\_CTL0控制寄存器的DED[4:0]位域中进行设置。DE信号的极性可以通过USART\_CTL2控制寄存器的DEP位进行设置。

### 16.3.7. 多处理器通信

在多处理器通信中，多个USART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将USART\_CMD寄存器中MMCMD位置1使USART进入静默模式。

如果USART处于静默模式，所有的接收状态标志位将不会被置位。此外，USART可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址匹配检测。

设备默认使用空闲总线检测方法唤醒USART。当在RX引脚检测到空闲帧时，硬件会将RWU清零，从而退出静默模式，但USART\_STAT寄存器中IDLEF位不会被置1。

当USART\_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位是0，该字节被认为是数据字节。如果地址字节的低4位或低7位与USART\_CTL1寄存器中的ADDR位相同，硬件会将RWU清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。状态标志可以从USART\_STAT寄存器中获取。如果地址字节的低4位或低7位与USART\_CTL1寄存器中的ADDR位不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会被置位。

如果USART\_CTL0寄存器中PCEN位被置位，地址字节最高位被视为校验位，其余位被视为地址位。如果ADDM位被置位，且接收帧为7位的数据，其中最低的6位将与ADDR[5:0]比较。如

果ADDM位被置位，且接收帧为9位的数据，其中最低8位将与ADDR[7:0]进行比较。

### 16.3.8. LIN 模式

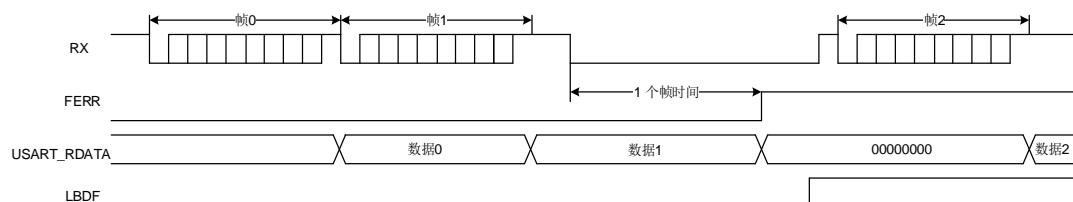
将USART\_CTL1寄存器的LMEN置位即可使能本地互联网络模式。在LIN模式下，USART\_CTL1寄存器中CKEN, STB[1:0]和USART\_CTL2的SCEN, HDEN, IREN位都应该被清0。

在发送一个普通数据帧时，LIN发送过程与普通发送过程相同。数据位的长度只能为8。一个停止位后连续13个0为断开帧。

断开检测功能完全独立于普通USART接收器。因此，断开检测可以是在空闲状态下，也可以在数据传输过程中。USART\_CTL1寄存器中LBLEN位可以选择断开帧的长度。如果在RX引脚检测到大于或等于与预期的断开帧长度的0（LBLEN=0时，10个0；LBLEN=1时，11个0），USART\_STAT寄存器中LBDF置位。如果USART\_CTL1寄存器中LBDIE被置位，将产生中断。

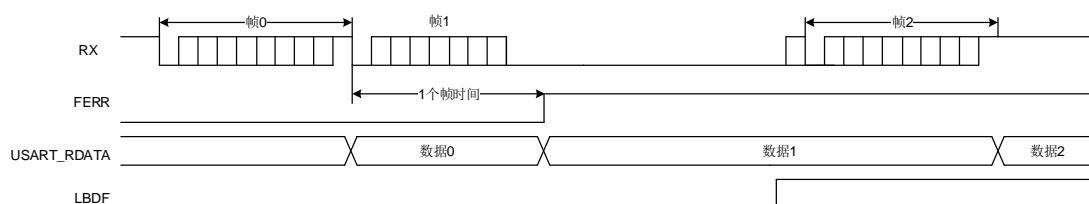
如[图 16-9. 空闲状态下检测断开帧](#)所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

**图 16-9. 空闲状态下检测断开帧**



如[图 16-10. 数据传输过程中检测断开帧](#)所示，如果断开帧发生在数据传输过程中，当前传输帧发生错误，FERR置位。

**图 16-10. 数据传输过程中检测断开帧**



### 16.3.9. 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\_CTL1的CKEN位来使能。在同步模式下，USART\_CTL1的LMEN和USART\_CTL2的SCEN, HDEN, IREN位应被清0。CK引脚作为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引脚输出时钟脉冲。USART\_CTL1的CLEN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的发送过程中，也不会有时钟信号产生。USART\_CTL1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\_CTL1的CPL位用来决定在USART同步模式空闲状态下，时钟引脚的电平。

CK引脚输出波形由USART\_CTL1寄存器中CPL, CPH, CLEN位决定。软件仅在USART禁用(UEN=0)时才可以改变它们的值。

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，并无任何过采样。

图 16-11. 同步模式下的 USART 示例

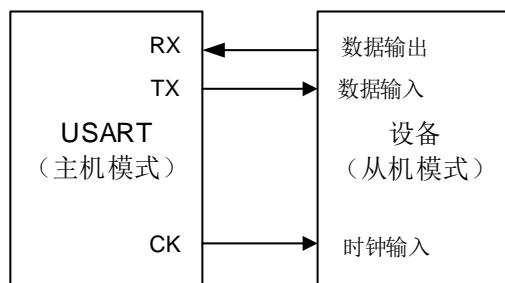
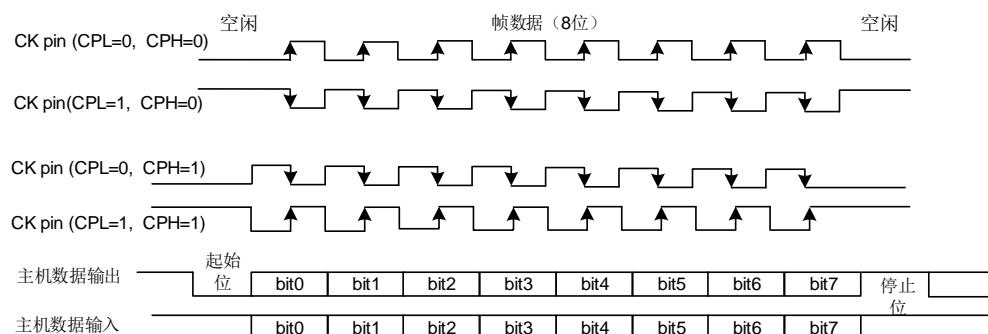


图 16-12. 8-bit 格式的 USART 同步通信波形 (CLEN=1)

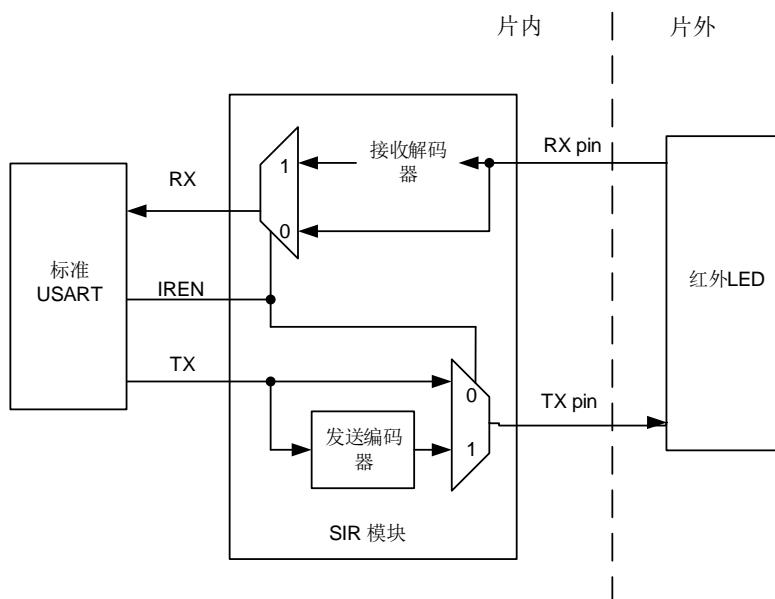


### 16.3.10. 串行红外 (IrDA SIR) 编解码功能模块

串行红外编解码功能通过置位USART\_CTL2寄存器中IREN使能。在IrDA模式下，USART\_CTL1寄存器的LMEN, STB[1:0], CKEN位和USART\_CTL2寄存器的HDEN, SCEN位将被清0。

在IrDA模式下，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，经解调后将数据发送至USART接收器。对于编码器而言，波特率应小于115200。

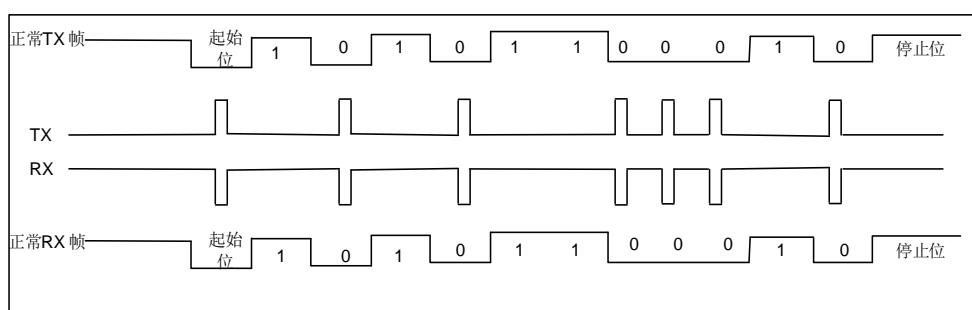
图 16-13. IrDA SIR ENDEC 模块



在IrDA模式下，TX引脚与RX引脚电平不同。TX引脚通常为低电平，RX引脚通常为高电平。IrDA引脚电平保持稳定代表逻辑‘1’，红外光源脉冲（RTZ信号）代表逻辑‘0’。其脉冲宽度通常占一个位时间的3/16。IrDA无法检测到宽度小于1个PSC时钟的脉冲。如果脉冲宽度大于1但是小于2倍PSC时钟，IrDA则无法可靠地检测到。

由于IrDA是一种半双工协议，因此在IrDA SIR ENDEC模块中，发送和接收不得同时进行。

图 16-14. IrDA 数据调制



将USART\_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART\_GP寄存器中PSC[7:0]位配置。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收解码器工作模式与正常IrDA模式相同。

### 16.3.11. 半双工通信模式

通过设置USART\_CTL2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\_CTL1寄存器的LMEN，CKEN位和USART\_CTL2寄存器的SCEN，IREN位清零。

半双工模式下仅用单线通信。TX引脚和RX引脚从内部连接到一起，TX引脚应被配置为IO管脚。通信冲突应由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

### 16.3.12. 智能卡 (ISO7816-3) 模式

智能卡模式是一种异步通信模式，支持ISO7816-3协议。支持字节模式( $T=0$ )和块模式( $T=1$ )。将USART\_CTL2寄存器的SCEN位置1，即可使能智能卡模式。在智能卡模式下，USART\_CTL1寄存器的LMEN位和USART\_CTL2的HDEN, IREN位应该清0。

如果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个奇偶校验位）+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引脚须被设置成开漏模式，这个引脚将会与智能卡驱动同一条双向连线。

**图 16-15. ISO7816-3 数据帧格式**



#### 字节模式 ( $T=0$ )

相较于正常操作模式下的时序，从发送移位寄存器到TX引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USART\_GP寄存器的GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2.5位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，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。将USART\_CTL2寄存器中的NKEN置位可以使能NACK信号。

空闲帧和断开帧在智能卡模式下不适用。

#### 块模式 ( $T=1$ )

在 $T=1$ （块模式）下，USART\_CTL2寄存器的NKEN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USART\_RT寄存器设置成BWT（块等待）-11的值并将RBNEIE置位。如果到了这个时间，还没有从智能卡收到应答，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完后去使能DMA。

在接收到第一个字节之后（RBNE中断）必须将USART\_RT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统计收到的字节数，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\_RT寄存器的BL[7:0]。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。软件可以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。

如果选择直接转换，从数据帧的最低位开始传输，TX引脚高电平代表逻辑‘1’，偶校验。在这种情况下，MSBF位和DINV位都设置为0（默认值）。

如果选择反向转换，从数据帧的最高位开始传输，TX引脚低电平代表逻辑‘1’，偶校验。在这种情况下，MSBF位和DINV位都设置为1。

#### 16.3.13. ModBus 通信

通过实现块尾检测功能，USART提供实现ModBus/RTU和ModBus/ASCII协议的基本支持。

在ModBus/RTU模式下，通过一个超过2个字符长度的空闲状态来识别块尾。这个功能是通过一个可设置的超时检测功能来实现的。

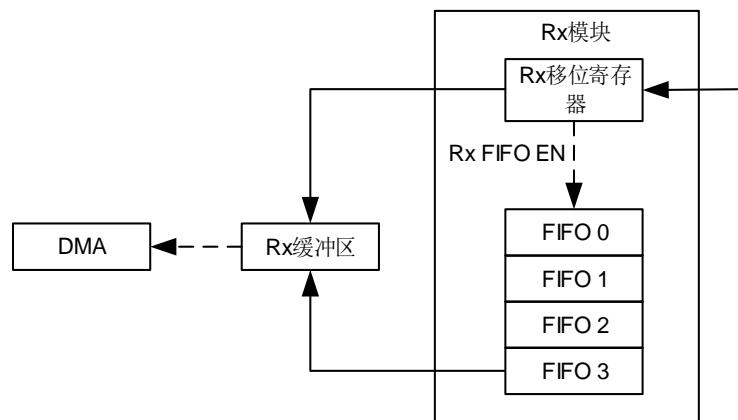
为了检测空闲状态，必须置位USART\_CTL1寄存器的RTEN位和USART\_CTL0寄存器的RTIE位。USART\_RT寄存器必须被设置成与2个字符超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

在ModBus/ASCII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。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。

### 16.3.14. 接收 FIFO

通过将USART\_RFCS寄存器的RFEN置位使能接收FIFO，可以避免当CPU无法迅速响应RBNE中断时，发生过载错误。接收FIFO和接收缓存区可储存多至5帧的数据。若接收FIFO满，RFFINT位将被置位。如果RFFIE被置位，将产生中断。

**图 16-16. USART 接收 FIFO 结构**



如果软件在响应RBNE中断时读数据接收缓冲区，在响应开始时，RBNEIE位应清0。当所有接收的数据被读出后，RBNEIE位应置位。在读出接收的数据前，PERR, NERR, FERR, EBF都应被清0。

### 16.3.15. 从 DeepSleep 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IRC8M或LXTAL（请参考RCU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是否正在传送数据。这可以通过USART\_STAT寄存器中的BSY标志来判断。REA位必须被检测以确保USART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标志位通过硬件被置1，并且在WUIE被置位的情况下，触发一个唤醒中断。

### 16.3.16. USART 中断

USART 中断事件和标志如[表 16-3. USART 中断请求](#)所示：

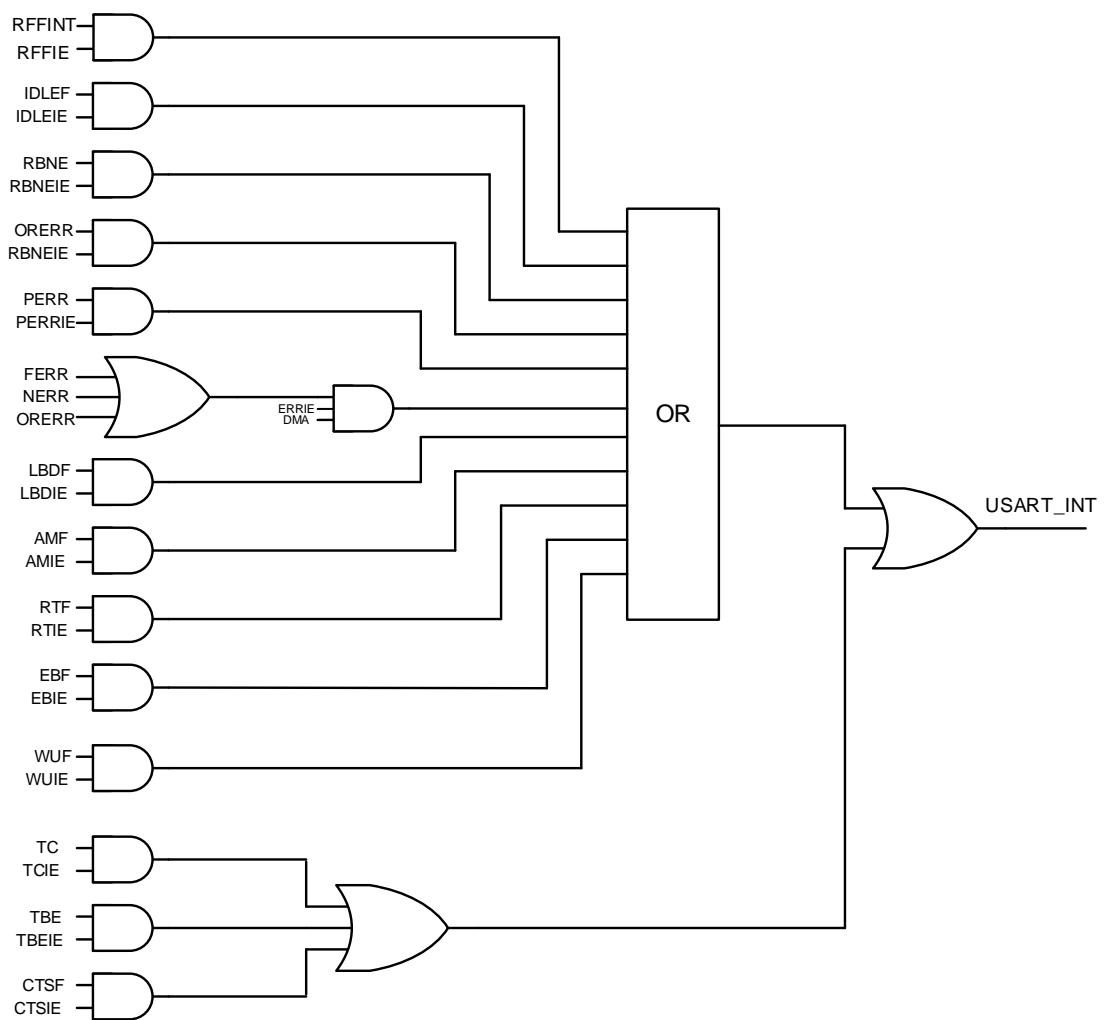
**表 16-3. USART 中断请求**

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE

中断事件	事件标志	使能控制位
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	ORERR	
接收FIFO满	RFFINT	RFFIE
检测到线路空闲	IDLEF	IDLEIE
奇偶校验错误	PERR	PERRIE
LIN模式下，检测到断开标志	LBDF	LBDIE
接收错误（噪声错误、溢出错误、帧错误）当DMA接收使能时	NERR或ORERR或FERR	ERRIE
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 16-17. USART 中断映射框图



## 16.4. USART 寄存器

USART0基地址: 0x4001 3800

USART1基地址: 0x4000 4400

### 16.4.1. USART 控制寄存器 0 (USART\_CTL0)

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	rw			rw					rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	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: 禁能中断 1: 使能中断 在USART1, 该位保留。
26	RTIE	接收超时中断使能 0: 禁能中断 1: 使能中断 在USART1, 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OVSMOD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OVSMOD位来配置。 当USART被使能（UEN=1）时，该位域不能被改写。
15	OVSMOD	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，该位保持清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: 禁能ADDR字符匹配中断 1: 使能ADDR字符匹配中断
13	MEN	静默模式使能 0: 禁能静默模式 1: 使能静默模式
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: 地址匹配 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: 禁能校验控制 1: 使能校验控制被 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	PERRIE	校验错误中断使能 0: 禁能校验错误中断 1: 使能校验错误中断，当USART_STAT寄存器的PERR位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能 0: 禁能发送寄存器空中断 1: 使能发送寄存器空中断，当USART_STAT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: 禁能发送完成中断 1: 使能发送完成中断，当USART_STAT寄存器中TC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: 禁能读数据缓冲区非空中断和过载错误中断 1: 使能读数据缓冲区非空中断和过载错误中断，当USART_STAT寄存器的ORERR或RBNE位置位时，将触发中断。
4	IDLEIE	IDLE线检测中断使能 0: 禁能IDLE线检测中断

1: 使能IDLE线检测中断, 当USART\_STAT寄存器的IDLEF位置位时, 将触发中断。

3	TEN	发送器使能 0: 禁能发送器 1: 使能发送器
2	REN	接收器使能 0: 禁能接收器 1: 使能接收器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IRC8M或LXTAL。 在USART1, 该位保留。
0	UEN	USART使能 0: 禁能USART预分频器和输出 1: 使能USART预分频器和输出

### 16.4.2. USART 控制寄存器 1 (USART\_CTL1)

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
ADDR[7:0]								RTEN	保留		MSBF	DINV	TINV	RINV	
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	CPL	CLEN	保留	LBDIE	LBLEN	ADDM	保留			
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	保留			

位/位域	名称	描述
31:24	ADDR[7:0]	USART的节点地址 这些位给出USART的节点地址。 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匹配的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM位被清零时, 仅仅ADDR[3:0]被用来比较。 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位) 与ADDR[7:0]的值进行比较, 如果匹配, AMF标志将被置位。 当接收器 (REN=1) 和USART (UEN=1) 被使能时, 该位域不能被改写。
23	RTEN	接收器超时使能 0: 禁能接收器超时功能 1: 使能接收器超时功能

在USART1，该位保留。

22:20	保留	必须保持复位值。
19	MSBF	高位在前 0: 数据发送/接收，采用低位在前 1: 数据发送/接收，采用高位在前 USART被使能（UEN=1）时，该位域不能被改写。
18	DINV	数据位反转 0: 数据位信号值没有反转 1: 数据位信号值被反转 USART被使能（UEN=1）时，该位域不能被改写。
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管脚功能不被交换 1: TX和RX管脚功能被交换 当USART被使能（UEN=1）时，该位域不能改写。
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USART1，该位保留。
13:12	STB[1:0]	STOP位长 00: 1停止位 01: 0.5停止位 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: 禁能CK管脚 1: 使能CK管脚 USART被使能（UEN=1）时，该位域不能被改写。
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	CLEN	CK长度	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	LIN断开信号检测中断使能	0: 禁能断开信号检测中断 1: 使能断开信号检测中断, 当USART_STAT的LBDF位置位, 将产生中断。 在USART1, 该位保留。
5	LBDL	LIN断开帧长度	0: 检测10位断开帧 1: 检测11位断开帧 USART被使能 (UEN=1) 时, 该位域不能被改写。 在USART1, 该位保留
4	ADDM	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 (ADDR[5:0], ADDR[6:0] 和ADDR[7:0]) 执行。 USART被使能 (UEN=1) 时, 该位域不能被改写。
3:0	保留	必须保持复位值。	

### 16.4.3. USART 控制寄存器 2 (USART\_CTL2)

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	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
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
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	从深度睡眠模式唤醒中断使能 0: 禁能从深度睡眠模式唤醒中断 1: 使能从深度睡眠模式唤醒中断 在USART1, 该位保留。
21:20	WUM[1:0]	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_STAT寄存器中的WUF（从深度睡眠唤醒标志）标志。 00: WUF在地址匹配的时候置位。如何实现地址匹配在ADDR和ADDM中定义。 01: 保留 10: WUF在检测到起始位时置位。 11: WUF在检测到RBNE时置位。 USART被使能（UEN=1）时，该位域不能被改写。 在USART1, 该位保留。
19:17	SCRTNUM[2:0]	智能卡自动重试数目 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（FERR位置位）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PERR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，该位域被清零，并停止重发。 在USART1, 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: 禁能DE功能 1: 使能DE功能 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，但是相应错误标志位会被置位。这种模式可用于智能卡模式。

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置位。软件在清除错误标志前，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: 使能溢出功能。当接收到的数据在新数据到达前没有被读走，ORERR错误标志位将被置位，并且新数据将会丢失。
		1: 禁能溢出功能。当接收到的数据在新数据到达前没有被读走，ORERR错误标志位将不会被置位，新数据会将USART_RDATA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	OSB	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: 禁能CTS中断
		1: 使能CTS中断，当USART_STAT的CTS位置位时，会产生中断。
9	CTSEN	CTS使能
		0: 禁能CTS硬件流控
		1: 使能CTS硬件流控
		USART被使能（UEN=1）时，该位域不能被改写。
8	RTSEN	RTS使能
		0: 禁能RTS硬件流控禁用
		1: 使能RTS硬件流控，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: 禁能智能卡模式
		1: 使能智能卡模式
		USART被使能（UEN=1）时，该位域不能被改写。
		在USART1中，该位保留。
4	NKEN	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USART1中，该位保留。

3	HDEN	半双工使能 0: 禁能半双工模式 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: 禁能IrDA 1: 使能IrDA USART被使能（UEN=1）时，该位域不能被改写。 在USART1中，该位保留。
0	ERRIE	多级缓存通信模式的错误中断使能 0: 禁能错误中断 1: 在多级缓存通信时，当USART_STAT寄存器的FERR位，ORERR位或NERR位被置位时，会产生中断。

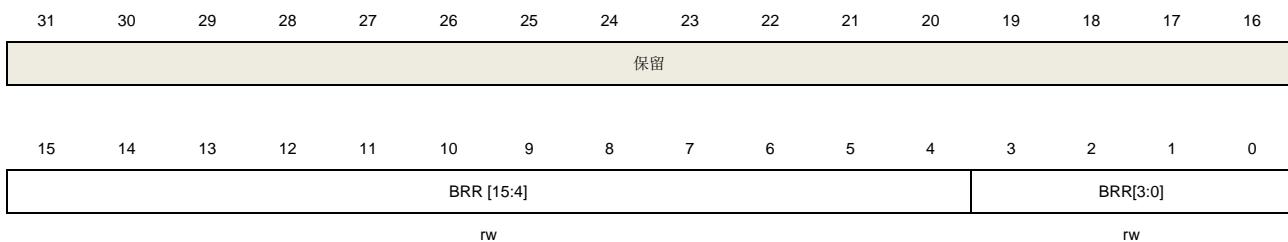
#### 16.4.4. USART 波特率寄存器 (USART\_BAUD)

地址偏移: 0x0C

复位值: 0x0000 0000

当USART（UEN=1）被使能时，该寄存器不能被改写。

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	BRR[15:4]	波特率分频系数的整数部分 $\text{INTDIV}[11:0] = \text{BRR}[15:4]$
3:0	BRR [3:0]	波特率分频系数的小数部分 如果OVSMOD = 0, FRADIV [3:0] = BRR [3:0];

如果OVSMOD = 1, FRADIV [3:1] = BRR [2:0], BRR [3]必须被置0。

### 16.4.5. USART 保护时间和预分频器寄存器 (USART\_GP)

地址偏移: 0x10

复位值: 0x0000 0000

USART被使能 (UEN=1) 时, 该寄存器不能被改写。

在USART1中, 该寄存器保留。

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	GUAT[7:0]	在智能卡模式下的保护时间值 USART被使能 (UEN=1) 时, 该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下, 对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数。 00000000: 保留 – 不设置这个值 00000001: 1分频 00000010: 2分频 ... 在IrDA正常模式下的分频值 00000001: 仅能设为这个值 在智能卡模式下, 对系统时钟进行分频的值存于PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 00000: 保留 -不设置这个值 00001: 2分频 00010: 4分频 00011: 6分频 ... USART被使能 (UEN=1) 时, 该位域不能被改写。

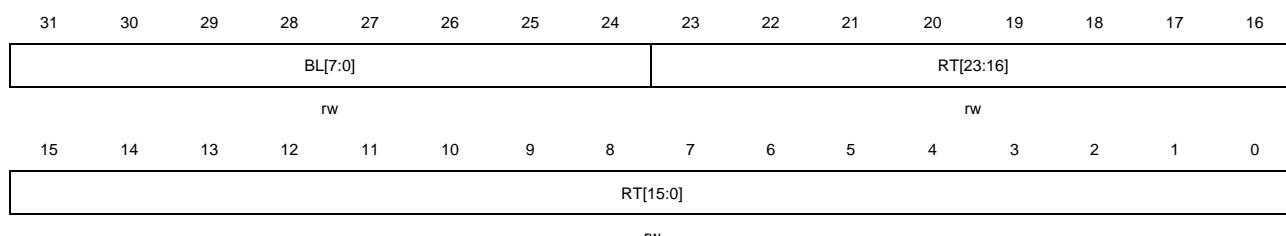
### 16.4.6. USART 接收超时寄存器 (USART\_RT)

地址偏移: 0x14

复位值: 0x0000 0000

在USART1中，该寄存器保留。

该寄存器只能按字 (32位) 访问。



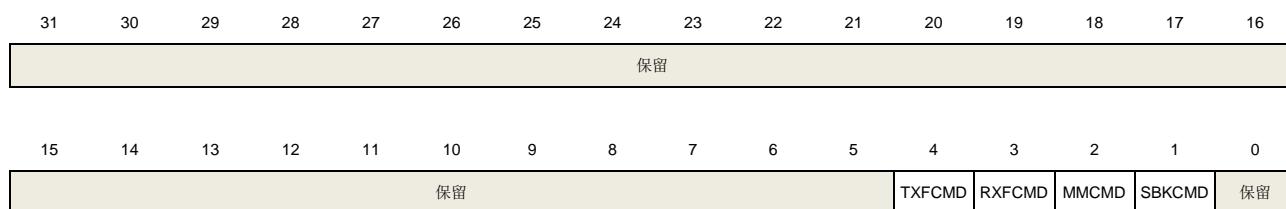
位/位域	名称	描述
31:24	BL[7:0]	<p>块长度</p> <p>这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度 (1-LEC/2-CRC) - 1。</p> <p>这个值可以在块接收开始时设置（用于需要从块的序言提取块的长度的情形），这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。</p> <p>在其他模式下，当REN=0（禁用接收器）并且/或者当EBC位被写1时块的长度计数器被清0。</p>
23:0	RT[23:0]	<p>接收器超时门限</p> <p>该位域指定接收超时值，单位是波特时钟的时长</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。</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>

### 16.4.7. 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。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFCMD	发送数据清空请求 向该位写1去置位 <b>TBE</b> 标志位，以取消发送数据。 在 <b>USART1</b> 中，该位保留。
3	RXFCMD	接收数据清空请求 向该位写1来清除 <b>RBNE</b> 标志位，以丢弃未读的接收数据。
2	MMCMD	静默模式请求 向该位写1使 <b>USART</b> 进入静默模式并且置位 <b>RWU</b> 标志位。
1	SBKCMD	发送断开帧请求 向该位写1置位 <b>SBKF</b> 标志并使 <b>USART</b> 在空闲时发送一个断开帧。
0	保留	必须保持复位值。

#### 16.4.8. USART 状态寄存器 (**USART\_STAT**)

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	
									REA	TEA	WUF	RWU	SBF	AMF	BSY	
									r	r	r	r	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		保留		EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			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 <b>USART</b> 核心逻辑的接收使能状态，该位可以通过硬件设置。 0: <b>USART</b> 核心接收逻辑禁用 1: <b>USART</b> 核心接收逻辑被使能
21	TEA	发送使能通知标志 该位反映了 <b>USART</b> 核心逻辑的发送使能状态，该位可以通过硬件设置。 0: <b>USART</b> 核心发送逻辑禁用 1: <b>USART</b> 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_CTL2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， 该位通过硬件置位， 这个事件在WUM位域被定义。 向USART_INTC寄存器中的WUC写1， 该位被清0。 当UESM被清0时， 该位清0。 在USART1中， 该位保留。
19	RWU	接收器从静默模式唤醒 该位表示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）是用通过USART_CTL0寄存器的WAKE位选择。 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来将该位置位。
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。 在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_CTL0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_INTC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
15:13	保留	必须保持复位值。
12	EBF	块结束标志 0: 块没有结束 1: 块结束已到（足够的字节数）， 如果USART_CTL1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_INTC寄存器的EBC写1清0。 在USART1中， 该位保留。
11	RTF	接收超时标志 0: 尚未超时

1: 已经超时，如果USART\_CTL1寄存器的RTIE被置位，将会引发中断。  
 如果空闲的时间已经超过了在USART\_RT寄存器中设定的RT值，通过硬件置1。  
 通过向USART\_INTC寄存器的RTC位写1清0。  
 在智能卡模式，这个超时相当于CWT或BWT计时。  
 在USART1中，该位保留。

10	CTS	CTS电平 这个值等于nCTS输入引脚电平的反向拷贝。 0: nCTS输入引脚高电平 1: nCTS输入引脚低电平
9	CTSF	CTS变化标志 0: nCTS状态线没有变化 1: nCTS状态线发生变化 如果USART_CTL2寄存器的CTSIE位置位，将引发中断。 当nCTS输入变化时，由硬件置位。 通过向USART_INTC寄存器的CTSC位写1，清零该位。
8	LBDF	LIN断开检测标志 0: 没有检测到LIN断开字符 1: 检测到LIN断开字符。当USART_CTL1寄存器的LBDIE位被置位时，将会有中断产生。 当LIN断开帧被检测到的时候，硬件置位。 通过向USART_INTC寄存器的LBDC位写1，清零该位。 在USART1中，该位保留。
7	TBE	发送数据寄存器空 0: 数据没有发送到移位寄存器 1: 数据发送到移位寄存器。如果USART_CTL0寄存器的TBEIE位置位，将会有中断产生。 当USART_TDATA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。 通过向USART_TDATA寄存器中写数据来清0。
6	TC	发送完成 0: 发送没有完成 1: 发送完成。如果USART_CTL0寄存器的TCIE被置位，将会有中断产生。 如果一个包含数据的帧的发送完成且TBE被置位，该位由硬件置位。 通过向USART_INTC寄存器的TCC位写1清0。
5	RBNE	读数据缓冲区非空 0: 没有接收到数据 1: 已接收到数据并且可以读取。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。 当接收移位寄存器的内容已经被转移到寄存器USART_RDATA，由硬件置位。 通过读USART_RDATA寄存器或向USART_CMD寄存器的RXFCMD位写1清0。

4	IDLEF	空闲线检测标志 0: 没检测到空闲线 1: 检测到空闲线。如果USART_CTL0寄存器的IDLEIE位置1，将会有中断产生。 当检测到空闲线时，通过硬件置位。直到RBNE位置位，否则它不会被再次置位。 向USART_INTC寄存器的IDLEC位写1清0。
3	ORERR	溢出错误 0: 未检测到溢出错误 1: 检测到溢出错误。在多级缓存通信中，如果寄存器USART_CTL0的RBNEIE位置位，将会引发中断。如果寄存器USART_CTL2的ERRIE位置位也会引发中断。 在RBNE置位的情况下，如果接收移位寄存器的数据传递给USART_RDATA寄存器，将会由硬件置位。 向USART_INTC寄存器的OREC位写1清0。
2	NERR	噪声错误标志 0: 未检测到噪声错误 1: 检测到噪声错误。在多级缓存通信中，如果寄存器USART_CTL2的ERRIE位置位，将会有中断产生。 在接收帧的时候检测到噪声错误，将会由硬件置位。 向寄存器USART_INTC的NEC位写1清0。
1	FERR	帧错误 0: 未检测到帧错误 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_CTL2的ERRIE位置位，将会有中断产生。 当一个不同步，强噪声或者断开字符被检测到时，硬件置位。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。 向USART_INTC寄存器的FEC位写1清0。
0	PERR	校验错误 0: 未检测到校验错误 1: 检测到校验错误 在多级缓存通信中，如果寄存器USART_CTL0的PERRIE位置位，将会有中断产生。 当在接收模式的时候检测到校验错误，将会由硬件置位。 向USART_INTC寄存器的PEC位写1清0。

#### 16.4.9. USART 中断标志清除寄存器 (USART\_INTC)

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

w

w

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 向该位写1清除USART_STAT寄存器的WUF位。 在USART1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 向该位写1清除USART_STAT寄存器的AMF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清除USART_STAT寄存器的EBF位。 在USART1中，该位保留。
11	RTC	接收超时标志清除 向该位写1清除USART_STAT寄存器的RTF标志。 在USART1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清除USART_STAT寄存器的CTSF位。
8	LBDC	LIN断开字符检测标志清除 向该位写1清除USART_STAT寄存器的LBDF标志位。 在USART1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清除USART_STAT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清除USART_STAT寄存器的IDLEF位。
3	OREC	溢出标志清除 向该位写1清除USART_STAT寄存器的ORERR位。
2	NEC	噪声检测清除 向该位写1清除USART_STAT寄存器的NERR位。

---

1	FEC	帧格式错误标志清除 向该位写1清除USART_STAT寄存器的FERR位。
0	PEC	校验错误标志清除 向该位写1清除USART_STAT寄存器的PERR位。

#### 16.4.10. USART 数据接收寄存器 (USART\_RDATA)

地址偏移: 0x24

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

r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	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_CTL0寄存器的PCEN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。

#### 16.4.11. USART 数据发送寄存器 (USART\_TDATA)

地址偏移: 0x28

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

rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值 包含发送的数据字节

如果发送到的数据打开了奇偶校验位（USART\_CTL0寄存器的PCEN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\_STAT寄存器的TBE位被置位时，这个寄存器才可以改写。

#### 16.4.12. USART 兼容性控制寄存器（USART\_CHC）

地址偏移: 0xC0

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								EPERR	保留							

w0c

rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 该标志将在RBNE置位前，校验位被检测到时置位。 软件写0可以清除该位。 0: 没有检测到校验错误 1: 检测到校验错误
7:1	保留	必须保持复位值。
0	HCM	硬件流控制兼容性模式 0: nRTS信号等于RBNE状态寄存器 1: 当最后一个数据位（PCE置位时的奇偶位）被采样时，nRTS信号置位

#### 16.4.13. USART 接收 FIFO 控制和状态寄存器（USART\_RFCS）

地址偏移: 0xD0

复位值: 0x0000 0400

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
RFFINT	RFCNT[2:0]		RFF	RFE	RFFIE	RFEN	保留								ELNACK

r\_w0

r

r

r

rw

rw

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RFFINT	接收FIFO满中断标志
14:12	RFCNT[2:0]	接收FIFO计数值
11	RFF	接收FIFO满标志 0: 接收FIFO不为满 1: 接收FIFO满
10	RFE	接收FIFO空标志 0: 接收FIFO不为空 1: 接收FIFO空
9	RFFIE	接收FIFO满中断使能 0: 禁止接收FIFO满中断 1: 使能接收FIFO满中断
8	RFEN	接收FIFO使能 当UESM=1, 该位置位。 0: 禁止使用接收FIFO 1: 使能接收FIFO
7:1	保留	必须保持复位值。
0	ELNACK	若选择了智能卡模式, 提前NACK 如果检测到校验位错误, NACK脉冲提前1/16位的时间。 0: 若选择了智能卡模式, 禁止提前NACK。 1: 若选择了智能卡模式, 使能提前NACK。 在USART1中, 该位保留。

## 17. 内部集成电路总线接口 (I2C)

### 17.1. 简介

I2C（内部集成电路总线）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，快速模式以及快速+ 模式，具备 CRC 计算和校验功能、支持 SMBus（系统管理总线）、PMBus（电源管理总线）和 SAM\_V（验证安全控制模块）模式，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。

### 17.2. 主要特征

- 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），快速（最高 400 kHz）和快速+ 模式（最高 1MHz）。
- 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：字节成功传输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。
- 支持 SAM\_V 模式。

### 17.3. 功能说明

I2C 接口的内部结构如 [图 17-1. I2C 模块框图](#) 所示。

图 17-1. I2C 模块框图

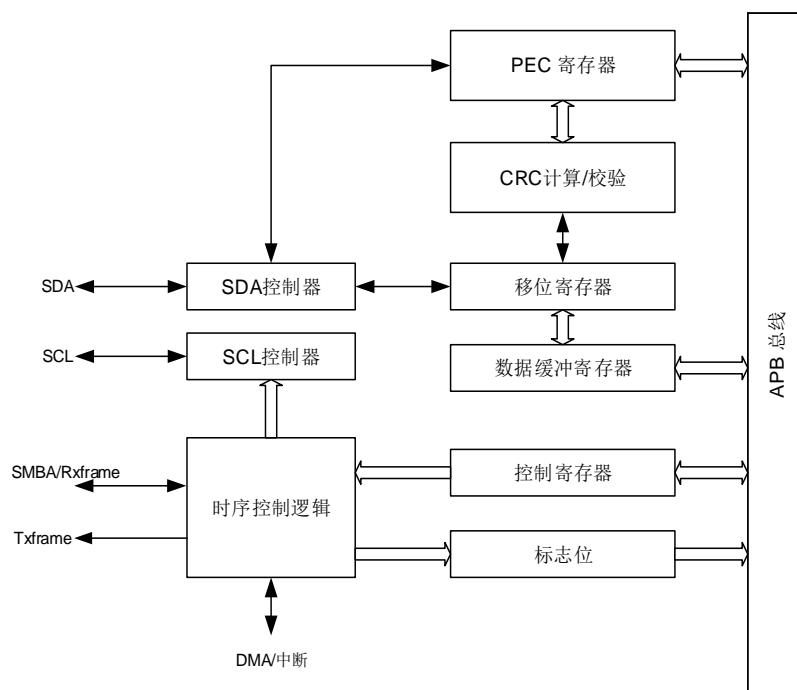


表 17-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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	多个主机可以尝试在不破坏信息的前提下同时控制总线
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

### 17.3.1. SDA 线和 SCL 线

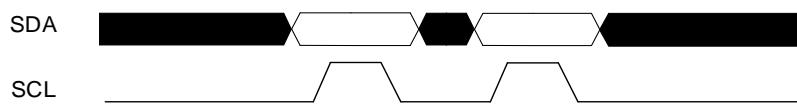
I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须是开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 Kbit/s，在快速模式下可以达到 400 Kbit/s，当 I2C\_FMPCFG 中 FMPEN 被置位时，在快速+ 模式下可达 1 Mbit/s。由于 I2C 总线上可能会连接不同工艺的设备（CMOS, NMOS, 双极性器件），逻辑‘0’和逻辑‘1’的电平并不是固定的，取决于 V<sub>DD</sub> 的实际电平。

### 17.3.2. 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 17-2. 数据有效性](#))。每个数据比特传输需要一个时钟脉冲。

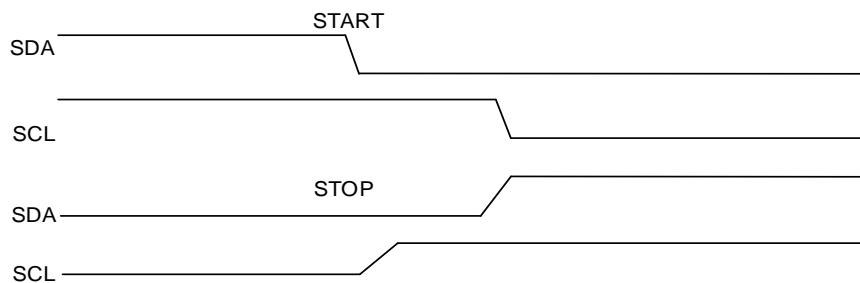
**图 17-2. 数据有效性**



### 17.3.3. 开始和停止信号

所有的数据传输起始于一个 START 结束于一个 STOP (参见 [图 17-3. 起始和停止信号](#))。START 信号定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 信号定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

**图 17-3. 起始和停止信号**

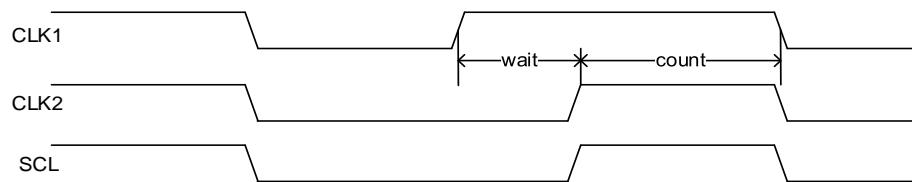


### 17.3.4. 时钟同步

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并完成传输，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计数它们的低电平周期，而且当主机的时钟变低电平时，它会使 SCL 线保持这种状态直到到达时钟的高电平 (参见 [图 17-4. 时钟同步](#))。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 17-4. 时钟同步**



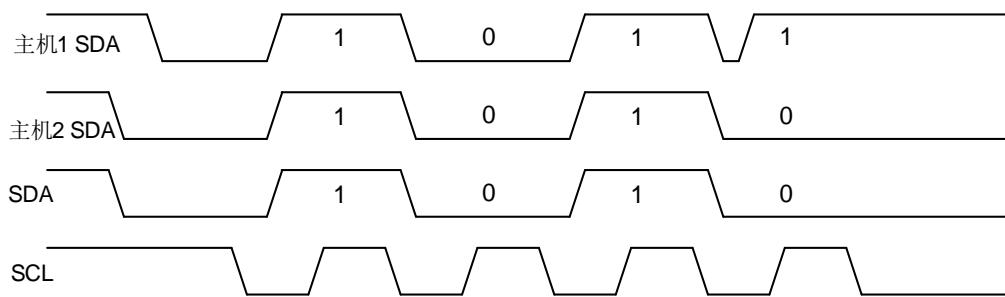
### 17.3.5. 仲裁

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信号的最短保持时间内在总线上产生一个有效的START信号，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，那么它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 17-5. SDA 线仲裁**



### 17.3.6. 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。

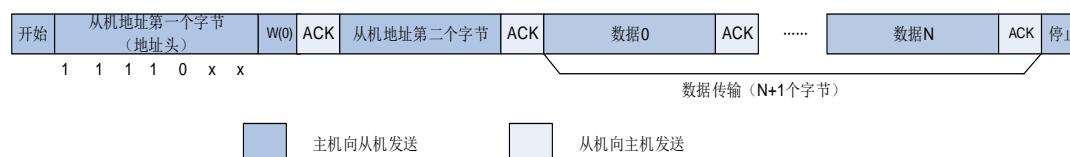
I2C从机检测到I2C总线上的START信号之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址（通过软件编程）进行比较，当两个地址相同时，I2C从机将发送一个确认应答（ACK），并响应总线的后续命令：发送或接收所需数据。此外，如果软件开启了广播呼叫，则I2C从机始终对一个广播地址（0x00）发送确认应答。I2C模块始终支持7位和10位的地址。

I2C主机负责产生START信号和STOP信号来开始和结束一次传输，并且负责产生SCL时钟。

**图 17-6. 7 位地址的 I2C 通讯流程**



**图 17-7. 10 位地址的 I2C 通讯流程（主机发送）**



**图 17-8. 10 位地址的 I2C 通讯流程（主机接收）**



### 17.3.7. 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信号之后，I2C变为主机模式，软件配置在I2C总线上发送STOP信号后，I2C又变回从机模式。

#### 从机发送模式下的软件流程

如[图17-9. 从机发送模式（10位地址模式）](#)所示，在从机模式下要发送数据，软件应该按照以下步骤来运行操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C\_CTL1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I2C总线上的START信号和地址。
2. 当接收到一个START信号及随后的地址后，地址可以是7位格式也可以是10位格式，I2C硬件将I2C\_STAT0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\_STAT0寄存器然后读I2C\_STAT1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并发送一个地址头到I2C总线。从机在检测到START和紧接着的地址头之后会继续将ADDSEND位置1。软件可以通过读I2C\_STAT0寄存器和接着读I2C\_STAT1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\_DATA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\_DATA寄存器，但是TBE位并没有被清0，因为写入I2C\_DATA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\_DATA，此时TBE位被清0，因为I2C\_DATA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\_DATA，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\_DATA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\_DATA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信号时被清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AERR（应答错误）会置起以通知软件发送结束。软件写0到AERR位可以清除此位。

**图 17-9. 从机发送模式（10 位地址模式）**



### 从机接收模式下的软件流程

如[图 17-10. 从机接收模式（10 位地址模式）](#)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\_CTL1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信号以及地址。
2. 在接收到START起始信号和匹配的7位或10地址之后，I2C硬件将I2C状态寄存器0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\_STAT0寄存器然后读I2C\_STAT1寄存器来清除ADDSEND位。当ADDSEND位被清0时，I2C就开始接收来自I2C总线的数据。
3. 当接收到第一个字节时，RBNE位被硬件置1，软件可以读取I2C\_DATA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\_DATA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信号，STPDET位被置1，软件通过先读I2C\_STAT0寄存器再写I2C\_CTL0寄存器来清除STPDET位。

**图 17-10. 从机接收模式（10 位地址模式）**



### 主机发送模式下的软件流程

如[图17-11. 主机发送模式（10位地址模式）](#)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\_CTL1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信号，随后等待I2C总线寻址。
2. 软件将START位置1，在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C\_STAT0的SBSEND位置1然后进入主机模式。现在软件应该读I2C\_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C\_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，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位地址的地址头，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\_STAT0寄存器然后写10位低地址到I2C\_DATA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，软件通过读I2C\_STAT0

寄存器然后读I2C\_STAT1寄存器清除ADDSEND位。

5. I2C进入数据发送状态，因为移位寄存器和数据寄存器I2C\_DATA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\_DATA寄存器，但是TBE位此时不会被清零，因为写入I2C\_DATA寄存器的字节会被立即移入内部移位寄存器。当移位寄存器非空时，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\_DATA，此时TBE会被清零，因为I2C\_DATA寄存器和移位寄存器都不为空。
7. 任意时刻TBE被置1，软件都可以向I2C\_DATA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\_DATA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信号时被清零。
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\_DATA寄存器此时都为空。软件此时应该配置STOP来发送一个STOP信号，此后TBE和BTC状态位都将被清0。

**图 17-11. 主机发送模式（10 位地址模式）**



## 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信号。因此，需要特别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和B。方案A需要保证软件能对I2C事件进行快速响应，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\_CTL1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信号，随后等待I2C总线寻址。
2. 软件将START位置1，从而在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C\_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C\_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C\_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，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位地址的地址头，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\_STAT0寄存器然后写10位低地址到I2C\_DATA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，软件应该通过读I2C\_STAT0寄存器然后读I2C\_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该再次将START位置1来重新产生一个START。在START产生后，SBSEND位会被置1。软件应该通过先读I2C\_STAT0然后写地址头到I2C\_DATA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\_STAT0然后读I2C\_STAT1来清除ADDSEND位。
5. 当接收到第一个字节时，硬件会将RBNE位置1。此时软件可以从I2C\_DATA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\_DATA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STOP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信号。

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 17-12. 主机接收使用方案 A 模式（10 位地址模式）**


### 方案 B

1. 首先，软件应该使能I2C外设时钟，配置I2C\_CTL1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信号和地址。
2. 软件将START位置1，从而在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C\_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C\_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C\_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，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位地址的地址头，硬件在发送地址头之后会将ADD10SEND位置1，软件应该通过读I2C\_STAT0寄存器然后写10位低地址到I2C\_DATA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，软件应该通过读I2C\_STAT0寄存器然后读I2C\_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该接着将START位再次置1来产生一个开始信号，START被发送出去以后SBSEND位被再次置1。软件应该通过先读I2C\_STAT0然后写地址头到I2C\_DATA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\_STAT0然后读I2C\_STAT1来清除ADDSEND位。
5. 当第一个字节被接收时，RBNE位会被硬件置1。此时软件可从I2C\_DATA寄存器读取出第一个字节，同时RBNE位被清0。
6. 此后任何时刻，只要RBNE位被置1，软件就可以从I2C\_DATA寄存器读取一个字节的数据，直到主机接收了N-3个字节。

如[图 17-13. 主机接收使用方案 B 模式 \(10 位地址模式\)](#)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。

7. 软件从I2C\_DATA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\_DATA，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。
8. 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STOP位置1，主机发出一个STOP信号。
9. 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\_DATA。
10. 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ACKEN位清0，在清除ADDSEND位之后将STOP位置1。当N=1时步骤5是最后一步。

#### N=2

在第2步，软件应该在START置1之前将POAP置1。在第4步，软件应该在清除ADDSEND位之前将ACKEN位清0。在第5步，软件应该一直等到BTC位被置1然后将STOP位置1且读取I2C\_DATA两次。

**图 17-13. 主机接收使用方案 B 模式 (10 位地址模式)**


### 17.3.8. SCL 线控制

SCL 线拉低功能是为了避免在接收时发生上溢错误以及在发送时发生下溢错误。如在软件编程模型中所示，在发送模式，当 TBE 和 BTC 被置位，发送器保持 SCL 线为低电平直到下一个发送数据写入传输缓冲区寄存器。在接收模式，当 RBNE 和 BTC 被置位，发送器保持 SCL 线为低电平直到传输缓冲区寄存器里的数据被读出。

当工作在从模式的时候，可以通过置位 I2C\_CTL0 寄存器的 SS 位禁止 SCL 线拉低功能。如果该位置位，软件要能足够快的处理 TBE，RBNE 和 BTC 状态，否则上溢或下溢的情况可能会发生。

### 17.3.9. DMA 模式下数据传输

按照前面的软件流程，每当 TBE 位或 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。

DMA 请求通过 I2C\_CTL1 寄存器的 DMAON 位使能。该位应该在清除 ADDSEND 状态位之后被置位。如果一个从机的 SCL 线延长功能被禁止，DMAON 位应该在 ADDSEND 事件前被置位。

参考 DMA 控制器的关于 DMA 的配置方法说明。DMA 必须在 I2C 传输开始之前配置和使能。当指定个数的字节已经传输完成，DMA 会发送一个传输结束（EOT）信号给 I2C 接口，并产生一个 DMA 传输完成中断。

当主机接收两个或两个以上字节时，需将 I2C\_CTL1 寄存器的 DMALST 位置位。在接收到最后一个字节之后，I2C 主机发送 NACK。在 DMA 传输完成中断 ISR 中，通过置位 STOP 位，产生一个停止信号。

当主机仅接收一个字节时，清除 ADDSEND 状态前 ACKEN 位必须被清除。在清除 ADDSEND 状态后或在 DMA 传输完成中断 ISR 中，通过置位 STOP 位，产生一个停止信号。

### 17.3.10. 报文错误校验

I2C 模块中有一个 PEC (包错误检查) 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  $x^8 + x^2 + x + 1$ ，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）。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 位和 PECTRANS 位被置 1，I2C 将自动发送或者检查 PEC 值。

### 17.3.11. 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 Battery Data）。

#### 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**是基于I2C硬件实现的，它使用了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接口。在此协议中，系统中的host与设备之间有一个重要的区别，即host具有分配地址的功能。

## 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之后，将可以继续I2C通信。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，如果某个通信耗时太久，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。这样就并不允许从设备将时钟拉得太长时间。

## 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，每次通信最后都将传输PEC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 $x^8+x^2+x+1$ （CRC-8-ATM HEC算法，初始化为0）。

## SMBus 警报

**SMBus**还有一个额外的中断信号，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”，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。为了使用**SMBus**模式，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，需要将I2C\_CTL0中SMBEN位置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（SMBSEL=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SMBALT标志位，并实现相应的功能。

### 17.3.12. SAM\_V 支持

为了支持 SAM\_V 标准，I2C 模块增加两个附加的引脚：txframe 和 rxframe。Txframe 是一个输出引脚，在主机模式下，当该引脚输出电平为高电平时，表示 I2C 是忙的。Rxframe 是一个输入引脚，应该与 SMBALERT 信号一起多路复用。

SAM\_V 模式通过置位 I2C\_SAMCS 寄存器的 SAMEN 位使能。txframe 和 rxframe 引脚的状态可以通过 I2C\_SAMCS 寄存器的 RFR, RFF, TFR, TFF, RXF 和 TXF 标志反映。如果对应的中断使能位置位，将产生 I2C 中断。

### 17.3.13. 状态、错误和中断

I2C有一些状态、错误标志位，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 17-2. 事件状态标志位**

事件标志位名称	说明
SBSEND	主机发送 START 信号
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 信号
BTC	字节发送结束
TBE	发送时 I2C_DATA 为空
RBNE	接收时 I2C_DATA 非空
RFR	SAM_V 模式时检测到 rxframe 上升沿
RFF	SAM_V 模式时检测到 rxframe 下降沿
TFR	SAM_V 模式时检测到 txframe 上升沿
TFF	SAM_V 模式时检测到 txframe 下降沿

**表 17-3. 错误标志位**

错误名称	说明
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	当禁用 SCL 拉低后，发生了上溢或下溢
AERR	没有接收到应答
PECERR	CRC 值不相同
SMBTO	SMBus 模式下总线超时
SMBALT	SMBus 警报

## 17.4. I2C 寄存器

I2C0基地址: 0x4000 5400

I2C1基地址: 0x4000 5800

### 17.4.1. 控制寄存器 0 (I2C\_CTL0)

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
SRESET	保留	SALT	PECTRA NS	POAP	ACKEN	STOP	START	SS	GCEN	PECEN	ARPEN	SMBSEL	保留	SMBEN	I2CEN

位/位域	名称	描述
31:16	保留	必须保持复位值。
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	SMBus 警报 通过 SMBA 引脚发出警报。 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，或监测到 START / STOP 信号，或 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 位决定对目前正在接收的字节是否发送 ACK，PECTRANS 位表明正在接收的字节是否为 PEC。 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	<b>ACKEN</b>	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	<b>STOP</b>	I2C 总线上产生一个 STOP 信号 软件置 1 和清 0, SMBus 超时时, 硬件置 1, 监测到 STOP 信号时, 硬件清 0。 0: 不发送 STOP 1: 发送 STOP
8	<b>START</b>	I2C 总线上产生一个 START 信号 软件置 1 和清 0, 当监测到 START 信号或 I2CEN=0 时由硬件清 0。 0: 不发送 START 1: 发送 START
7	<b>SS</b>	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	<b>GCEN</b>	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	<b>PECEN</b>	PEC 计算使能 0: PEC 计算禁能 1: PEC 计算使能
4	<b>ARPEN</b>	SMBus 下 ARP 协议使能 0: ARP 禁能 1: ARP 使能
3	<b>SMBSEL</b>	SMBus 类型选择 0: 从机 1: 主机
2	保留	必须保持复位值。
1	<b>SMBEN</b>	SMBus / I2C 模式开关 0: I2C 模式 1: SMBus 模式
0	<b>I2CEN</b>	I2C 外设使能 0: 禁能 I2C 1: 使能 I2C

### 17.4.2. 控制寄存器 1 (I2C\_CTL1)

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	DMALST	DMAON	BUFIE	EVIE	ERRIE	保留						I2CCLK[6:0]				

rw      rw      rw      rw      rw      rw      rw

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	DMALST	DMA 最后传输配置 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	BUFIE	缓冲区中断使能 0: 禁用缓冲区中断, 如果 EVIE=1, 当 TBE = 1 或 RBNE = 1 时不产生中断。 1: 使能缓冲区中断, 如果 EVIE=1, 当 TBE = 1 或 RBNE = 1 时产生中断。
9	EVIE	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 BUFIE=1 时 TBE=1 或 RBNE=1 时产生中断。
8	ERRIE	错误中断使能 0: 禁用错误中断 1: 使能错误中断, 意味着当 BERR、LOSTARB、AERR、OUERR、PECERR、SMBTO 或 SMBALT 标志位生效时产生中断。
7	保留	必须保持复位值。
6:0	I2CCLK[6:0]	I2C 外设时钟频率 I2CCLK[6:0]应该是输入 APB1 时钟频率, 最低 2MHz。 0000000 - 0000001: 无时钟 0000010 - 1001000: 2MHz~72MHz 1001001 - 1111111: 由于 APB1 时钟限制, 无时钟 <b>注意:</b> 在标准模式下, APB1 时钟频率需大于或者等于 2MHz。在快速模式下, APB1 时钟

频率需大于或者等于 8MHz。在快速模式+ 下，APB1 时钟频率需大于或者等于 24MHz。

### 17.4.3. 从机地址寄存器 0 (I2C\_SADDR0)

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。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDFOR	保留				ADDRESS[9:8]	ADDRESS[7:1]				ADDRESS[7:1]				ADDRES	S0
MAT															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDFORMAT	I2C 从机地址格式 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

### 17.4.4. 从机地址寄存器 1 (I2C\_SADDR1)

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								ADDRESS2[7:1]							

位/位域	名称	描述
31:8	保留	必须保持复位值。

7:1 ADDRESS2[7:1] 从机在双重地址模式下第二个 I<sub>C</sub> 地址

0 DUADEN 双重地址模式使能  
0: 禁用双重地址模式  
1: 使能双重地址模式

#### 17.4.5. 传输缓冲区寄存器 (I<sub>C</sub>\_DATA)

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															

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
31:8	保留	必须保持复位值。
7:0	TRB[7:0]	数据发送接收缓冲区

#### 17.4.6. 传输状态寄存器 0 (I<sub>C</sub>\_STAT0)

地址偏移: 0x14

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
SMBALT	SMBTO	保留	PECERR	OUERR	AERR	LOSTAR B	BERR	TBE	RBNE	保留	STPDET	ADD10S END	BTC	ADDSEN D	SBSEND

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	SMBALT	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 引脚被拉低且接收到警报地址 (从机模式) 或监测到警报 (主机模式)

14	SMBTO	<p><b>SMBus 模式下超时信号</b></p> <p>硬件置 1，软件写 0 清 0。</p> <p>0: 无超时错误</p> <p>1: 超时事件发生（SCL 被拉低达 25ms）</p>
13	保留	必须保持复位值。
12	PECERR	<p><b>接收数据时 PEC 错误</b></p> <p>硬件置 1，软件写 0 清 0。</p> <p>0: 接收到 PEC 且校验正确</p> <p>1: 接收到 PEC 但检验错误，此时不管 ACKEN 位的值，I2C 将发送 NACK</p>
11	OUERR	<p><b>当禁用 SCL 拉低功能后，在从机模式下发生了上溢或下溢事件。在从机接收模式下，假如 I2C_DATA 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生上溢错误。在从机发送模式下，假如当前字节已经发送完成，而 I2C_DATA 仍然为空，就会发生下溢错误。</b></p> <p>硬件置 1，软件写 0 清 0。</p> <p>0: 无上溢或下溢错误发生</p> <p>1: 发生上溢或下溢错误</p>
10	AERR	<p><b>应答错误</b></p> <p>硬件置 1，软件写 0 清 0。</p> <p>0: 未发生应答错误</p> <p>1: 发生了应答错误</p>
9	LOSTARB	<p><b>主机模式下仲裁丢失</b></p> <p>硬件置 1，软件写 0 清 0。</p> <p>0: 无仲裁丢失</p> <p>1: 发生仲裁丢失，I2C 模块返回从机模式。</p>
8	BERR	<p><b>总线错误，表示 I2C 总线上发生了预料之外的 START 信号或 STOP 信号。</b></p> <p>硬件置 1，软件写 0 清 0。</p> <p>0: 无总线错误</p> <p>1: 发生了总线错误</p>
7	TBE	<p><b>发送期间 I2C_DATA 为空</b></p> <p>硬件从 I2C_DATA 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_DATA 寄存器清除该位。如果移位寄存器和 I2C_DATA 寄存器都是空的，写 I2C_DATA 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）</p> <p>0: I2C_DATA 非空</p> <p>1: I2C_DATA 空，软件可以写</p>
6	RBNE	<p><b>接收期间 I2C_DATA 非空</b></p> <p>硬件从移位寄存器移动一个字节到 I2C_DATA 寄存器之后将此位置 1，读 I2C_DATA 可以清除此位。如果 BTC 和 RBNE 都被置 1，读 I2C_DATA 将不会清除 RBNE，因为移位寄存器的字节将被立即移到 I2C_DATA。</p> <p>0: I2C_DATA 为空</p>

		1: I2C_DATA 非空, 软件可以读
5	保留	必须保持复位值。
4	STPDET	<p>从机模式下监测到 STOP 信号</p> <p>此位被硬件置 1, 先读 I2C_STAT0 然后写 I2C_CTL0 可以清除此位。</p> <p>0: 从机模式下未监测到 STOP 信号</p> <p>1: 从机模式下监测到 STOP 信号</p>
3	ADD10SEND	<p>主机模式下 10 位地址的地址头被发送</p> <p>该位由硬件置 1, 软件读 I2C_STAT0 和写 I2C_DATA 清除此位。</p> <p>0: 主机模式下未发送 10 位地址的地址头</p> <p>1: 主机模式下发送 10 位地址的地址头</p>
2	BTC	<p>字节发送结束</p> <p>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_DATA 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_DATA 寄存器仍然是空的, 如果使能了 SCL 拉低功能, 位硬件就会置起 BTC 标志。</p> <p>此位由硬件置 1。</p> <p>可由以下三种方式清除:</p> <ol style="list-style-type: none"> <li>1、软件清除: 读 I2C_STAT0, 然后读或写 I2C_DATA 寄存器清除此位</li> <li>2、硬件清除: 发送一个 STOP 或 START 信号</li> <li>3、寄存器 I2C_CTL0 中 I2CEN=0</li> </ol> <p>0: 未发生 BTC</p> <p>1: 发生了 BTC</p>
1	ADDSEND	<p>主机模式下: 成功发送了地址并收到 ACK</p> <p>从机模式下: 接收到的地址与自身的地址匹配</p> <p>此位由硬件置 1, 软件读 I2C_STAT0 寄存器和读 I2C_STAT1 清 0。</p> <p>0: 从机模式下, 未收到地址或者收到的地址不匹配; 主机模式下, 无地址被发送或地址已发送但未收到从机回复的 ACK</p> <p>1: 从机模式下, 接收到的地址与自身的地址匹配; 主机模式下, 地址已发送并收到 ACK</p>
0	SBSEND	<p>主机模式下发送 START 信号</p> <p>此位由硬件置 1, 软件读 I2C_STAT0 和写 I2C_DATA 清 0。</p> <p>0: 未发送 START 信号</p> <p>1: START 信号被发送</p>

#### 17.4.7. 传输状态寄存器 1 (I2C\_STAT1)

地址偏移: 0x18

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
					PECV[7:0]			DUMODF	HSTSMB	DEFSMB	RXGC	保留	TR	I2CBSY	MASTER

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。
15:8	PECV[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 信号产生后或 I2CEN=0 时此位由硬件清 0。 0: 地址和 I2C_SADDR0 匹配 1: 地址和 I2C_SADDR1 匹配
6	HSTSMB	从机模式下监测到 SMBus 主机地址头 STOP 或 START 信号产生后或 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	SMBus 设备缺省地址 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0: SMBus 设备没有接收到缺省地址 1: SMBus 设备接收到缺省地址
4	RXGC	是否接收到广播地址 (0x00) STOP 或 START 信号产生后或 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
3	保留	必须保持复位值。
2	TR	发送端或接收端 该位表明 I2C 作为发送端还是接收端。STOP 或 START 信号产生后或 I2CEN 或 LOSTARB=1 时此位由硬件清 0。 0: 接收端 1: 发送端
1	I2CBSY	忙标志 STOP 信号后硬件清 0。 0: 无 I2C 通讯 1: I2C 正在通讯
0	MASTER	主机模式 表明 I2C 时钟在主机模式还是从机模式的标志位。 该位在 START 信号产生后由硬件置 1。 该位在 STOP 信号产生后或 I2CEN=0 或 LOSTARB=1 时此位由硬件清 0。 0: 从机模式

1: 主机模式

#### 17.4.8. 时钟配置寄存器 (I2C\_CKCFG)

地址偏移: 0x1C

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
FAST	DTCY	保留										CLKC[11:0]			

rw            rw            rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FAST	主机模式下 I2C 速度选择 0: 标准速度 1: 快速
14	DTCY	快速模式或快速+ 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: $T_{high}=T_{low}=CLKC \cdot T_{PCLK1}$ 如果 DTCY=0, 快速模式或快速+ 模式下: $T_{high}=CLKC \cdot T_{PCLK1}$ , $T_{low}=2 \cdot CLKC \cdot T_{PCLK1}$ 如果 DTCY=1, 快速模式或快速+ 模式下: $T_{high}=9 \cdot CLKC \cdot T_{PCLK1}$ , $T_{low}=16 \cdot CLKC \cdot T_{PCLK1}$ <b>注意:</b> 如果 DTCY=0, 当 PCLK1 为 3 的整数倍时, 波特率会比较准确。如果 DTCY=1, 当 PCLK1 为 25 的整数倍时, 波特率会比较准确。

#### 17.4.9. 上升时间寄存器 (I2C\_RT)

地址偏移: 0x20

复位值: 0x0000 0002

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								RISETIME[6:0]							
rw															

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:0	RISETIME[6:0]	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

#### 17.4.10. SAM 控制状态寄存器 (I2C\_SAMCS)

地址偏移: 0x80

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
RFR	RFF	TFR	TFF	保留	RXF	TXF	RFRIE	RFFIE	TFRIE	TFFIE	保留	STOEN	SAMEN		
rc_w0	rc_w0	rc_w0	rc_w0		r	r	rw	rw	rw	rw	保留	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RFR	接收帧上升沿标志, 由软件写 0 清 0
14	RFF	接收帧下降沿标志, 由软件写 0 清 0
13	TFR	发送帧上升沿标志, 由软件写 0 清 0
12	TFF	发送帧下降沿标志, 由软件写 0 清 0
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: 接收帧上升沿中断失能 1: 接收帧上升沿中断使能
6	RFFIE	接收帧下降沿中断使能 0: 接收帧下降沿中断失能 1: 接收帧下降沿中断使能
5	TFRIE	发送帧上升沿中断使能

		0: 发送帧上升沿中断失能 1: 发送帧上升沿中断使能
4	TFFIE	发送帧下降沿中断使能 0: 发送帧下降沿中断失能 1: 发送帧下降沿中断使能
3:2	保留	必须保持复位值。
1	STOEN	SAM_V 接口超时检测使能 0: SAM_V 接口超时检测失能 1: SAM_V 接口超时检测使能
0	SAMEN	SAM_V 接口使能 0: SAM_V 接口失能 1: SAM_V 接口使能

#### 17.4.11. 快速+ 模式配置寄存器 (**I2C\_FMPCFG**)

地址偏移: 0x90

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	描述
31:1	保留	必须保持复位值。
0	FMPEN	快速+ 模式使能 当该位被置 1 时, I2C 设备支持高达 1MHz

## 18. 串行外设接口/I片上音频接口（SPI/I2S）

### 18.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。SPI1还支持SPI四线主机模式。

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。

### 18.2. 主要特性

#### 18.2.1. SPI 主要特性

- 具有全双工、半双工和单工模式的主从操作。
- 16位宽度，独立的发送和接收缓冲区（只有SPI0）。
- 32位宽度，独立的发送和接收FIFO（只有SPI1）。
- 8位或16位数据帧格式（只有SPI0）。
- 4位到16位的数据帧格式（只有SPI1）。
- 低位在前或高位在前的数据位顺序。
- 软件和硬件NSS管理。
- 硬件CRC计算、发送和校验。
- 发送和接收支持DMA模式。
- 支持SPI TI模式。
- 支持SPI NSS脉冲模式。
- 支持SPI四线功能的主机模式（只有SPI1）。

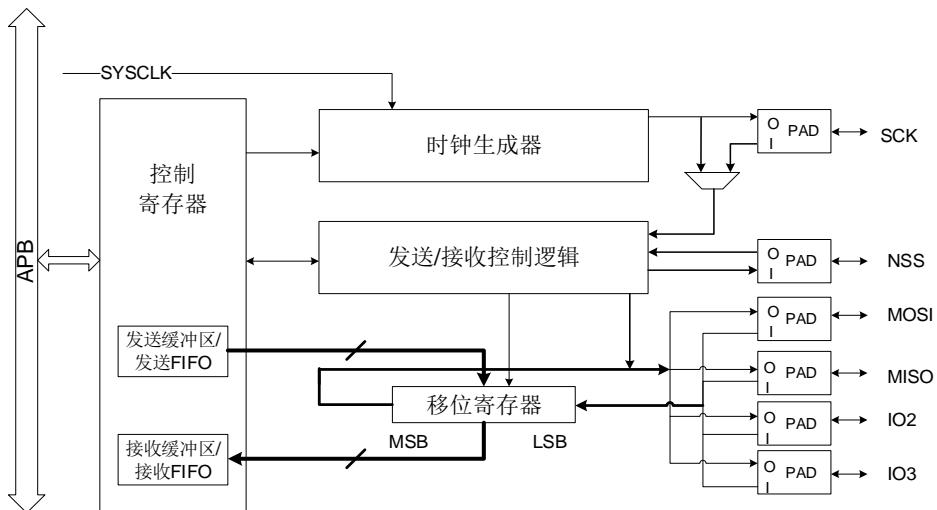
#### 18.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作。
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。
- 数据长度可以为16位，24位和32位。
- 通道长度为16位或32位。
- 16位缓冲区用于发送和接收。
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率。
- 可编程空闲状态时钟极性。
- 可以输出主时钟（MCK）。
- 发送和接收支持DMA功能。

## 18.3. SPI 功能说明

### 18.3.1. SPI 结构框图

图 18-1. SPI 结构框图



### 18.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 18-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机: SPI 时钟输出 从机: SPI 时钟输入
MISO	I/O	主机: 数据接收线 从机: 数据发送线 主机双向线模式: 不使用 从机双向线模式: 数据发送和接收线
MOSI	I/O	主机: 数据发送线 从机: 数据接收线 主机双向线模式: 数据发送和接收线 从机双向线模式: 不使用
NSS	I/O	软件 NSS 模式: 不使用 主机硬件 NSS 模式: NSSDRV=1 时, 为 NSS 输出, 适用于单主机模式; NSSDRV=0 时, 为 NSS 输入, 适用于多主机模式。 从机硬件 NSS 模式: 为 NSS 输入, 作为从机的片选信号。

### SPI 四线配置

SPI默认配置为单路模式，当SPI\_QCTL中的QMOD位置1时，配置为SPI四线模式（只适用于SPI1）。SPI四线模式只能工作在主机模式。

通过配置SPI\_QCTL中的IO23\_DRV位，在常规非四线SPI模式下，软件可以驱动IO2引脚和IO3引脚为高电平。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

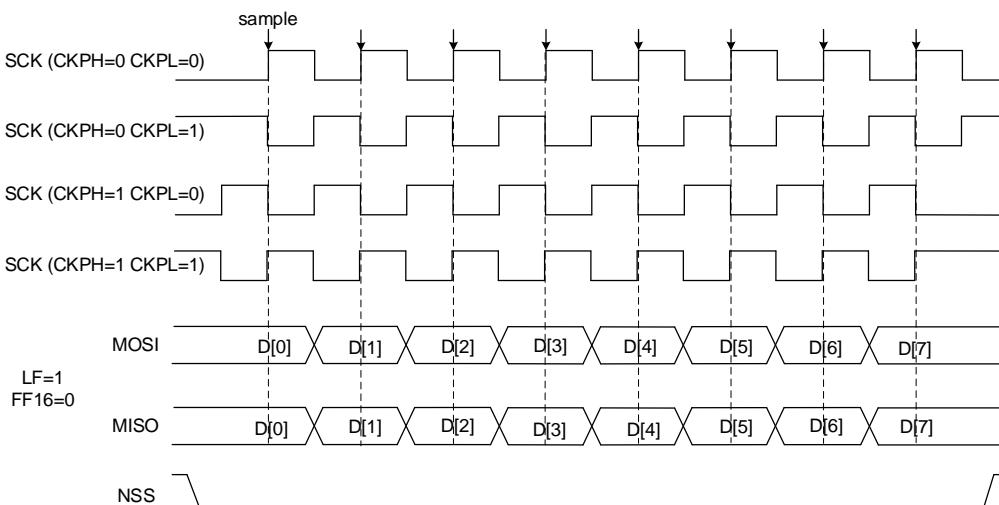
**表 18-2. SPI 四线信号描述**

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0
MISO	I/O	发送或接收数据 1
IO2	I/O	发送或接收数据 2
IO3	I/O	发送或接收数据 3
NSS	O	NSS 输出

### 18.3.3. SPI 时序和数据帧格式

SPI\_CTL0寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。在TI模式下，这两位没有意义。

**图 18-2. SPI0 常规模式下的时序图**



在SPI0常规模式中，通过SPI\_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。

通过设置SPI\_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI0先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

图 18-3. SPI1 常规模式下的时序图

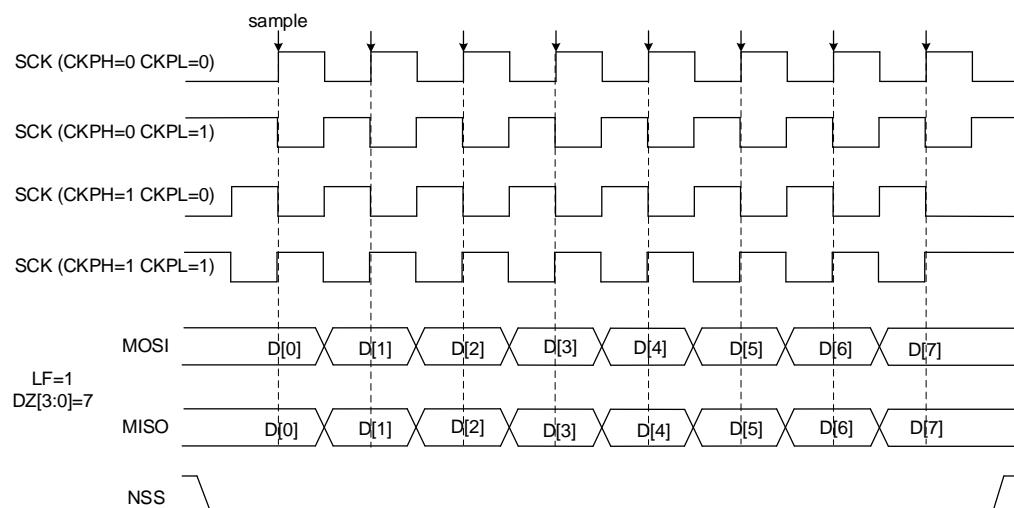
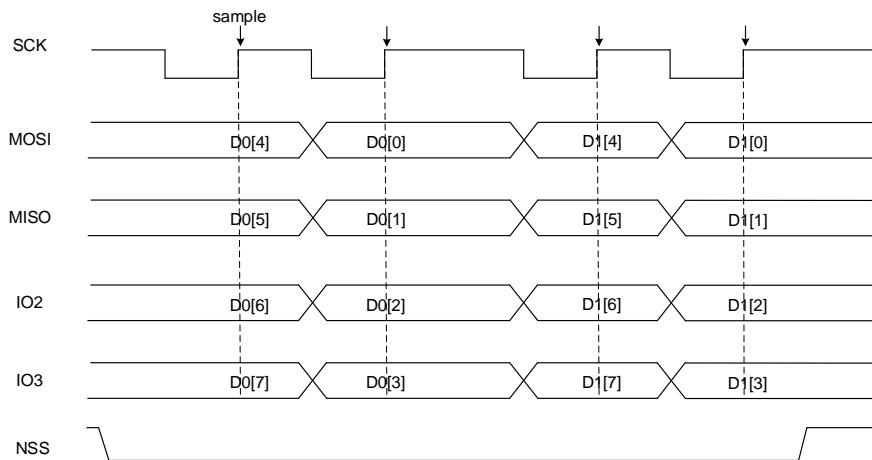


图 18-4. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0）



在SPI1常规模式中，通过SPI\_CTL1中的DZ[3:0]位域配置数据长度，可以设置为4位至16位。该设置不仅适用于数据的发送也适用于数据的接收。不论设置的数据长度是多少，对FIFO的读访问必须与SPI\_CTL1寄存器中的BYTEN位设置的对齐。在SPI四线模式下，数据长度固定为8位。

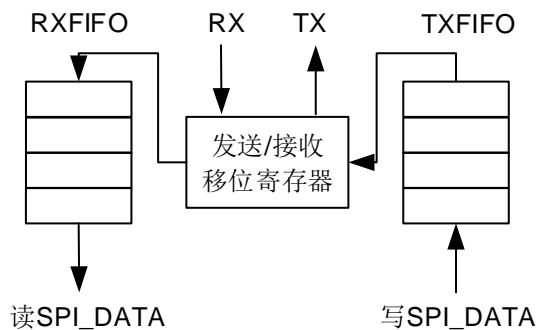
同样，通过设置SPI\_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI1先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

当访问SPI\_DATA寄存器时，数据帧总是右对齐成一个字节（如果数据长度小于或等于一个字节）或一个半字。通讯时，只有数据长度内的位会随时钟输出。

#### 18.3.4. 独立发送和接收缓冲区

独立的32位的接收缓冲区（RXFIFO）和发送缓冲区（TXFIFO）分别用于SPI数据传输的不同方向，它们使得SPI可以连续工作（只适用于SPI1）。

图 18-5. 发送/接收缓冲区



当当前TXFIFO的存储量小于或等于整体存储能力的一半时，TXFIFO被视为空<sup>(1)</sup>并且此时TBE被硬件置1。当TBE位置位时，向SPI\_DATA寄存器写数据，会把数据存入发送FIFO的末尾。当当RXFIFO被视为非空<sup>(2)</sup>时硬件将RBNE位置1。当RBNE位置位时，从SPI\_DATA寄存器读数据，将从接收FIFO获得最早数据。

**注意：**

(1) 对于SPI1，TXFIFO空意味着TXFIFO当前的存储量小于或等于TXFIFO整体存储能力的一半。TXFIFO满的意义与之相反。所以，当数据长度不大于8位时，TXFIFO最多能存储3个数据帧。如果下文出现TXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

(2) 对于SPI1，RXFIFO空的意义分为以下两种情况：如果SPI\_CTL1中BYTEN位为1时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的四分之一。此时，当数据长度不大于8位时，RXFIFO最多可以存储4个数据帧。如果SPI\_CTL1中BYTEN位为0时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的一半。RXFIFO满的意义与之相反。如果下文出现RXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

### 数据合并（仅适用于 SPI1）

在SPI\_CTL1寄存器中DZ[3:0]配置传输数据位宽为8位或者小于8位的情况下，通过配置SPI\_CTL1寄存器中BYTEN位为0，开启数据合并传输模式功能。在配置SPI\_CTL1寄存器中DZ[3:0]配置传输数据位宽为小于等于8位时，该功能可以实现当对SPI\_DATA寄存器进行16位写访问时，两个数据帧的发送是并行方式而不是串行方式。同样的，在接收端接收器通过对SPI\_DATA的一次16位读访问，获取这两个数据帧，并且这两帧数据在接收时，仅会产生一个RBNE事件。

**注意：**当被传输的数据为奇数个字节时，在发送端，需要用8位访问SPI\_DATA，发出最后一个数据帧。在接收端，为了产生最后一个字节的RBNE事件，接收器必须在接收最后一个数据帧时，改变BYTEN位。

### 18.3.5. NSS 功能

#### 从机模式

当配置为从机模式（MSTMOD=0）时，在硬件NSS模式（SWNSSEN = 0）下，SPI从NSS引

脚获取NSS电平，在软件NSS（SWNSSEN = 1）下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

**表 18-3. 从机模式 NSS 功能**

模式	寄存器配置	描述
从机硬件 NSS 模式	MSTMOD = 0 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0: NSS 电平为低 SWNSS = 1: NSS 电平为高

### 主机模式

在主机模式（MSTMOD=1）下，如果应用程序使用多主机连接方式，NSS可以配置为硬件输入模式（SWNSSEN=0，NSSDRV=0）或者软件模式（SWNSSEN=1）。一旦NSS引脚（在硬件NSS模式下）或SWNSS位（在软件NSS模式下）被拉低，SPI将自动进入从机模式，并且产生主机配置错误，CONFERR位置1。

如果应用程序希望使用 NSS 引脚控制 SPI 从设备，NSS 应该配置为硬件输出模式（SWNSSEN=0，NSSDRV=1）。使能SPI之后，NSS变为低电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

**表 18-4. 主机模式 NSS 功能**

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	适用于单主机模式，主机使用 NSS 引脚控制 SPI 从设备，此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。
主机硬件 NSS 输入模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	适用于多主机模式，此时 NSS 配置为硬件输入模式，一旦 NSS 引脚被拉低，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
主机软件 NSS 模式	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV: 不要求	适用于多主机模式，一旦 SWNSS = 0，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV: 不要求	从机可以使用硬件或软件 NSS 模式

### 18.3.6. SPI 运行模式

**表 18-5. SPI 运行模式**

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: 不要求	MOSI: 不使用 MISO: 接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: 发送 MISO: 不使用
MRB	双向线连接主机接收模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: 接收 MISO: 不使用
SFD	全双工从机模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 接收 MISO: 发送
STU	单向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 不使用 MISO: 发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: 不要求	MOSI: 接收 MISO: 不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI: 不使用 MISO: 发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1	MOSI: 不使用 MISO: 接收

模式	描述	寄存器配置	数据引脚用法
		BDOEN = 0	

图 18-6. 典型的全双工模式连接

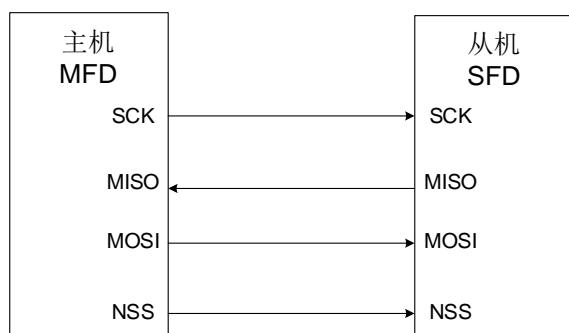


图 18-7. 典型的单工模式连接（主机：接收，从机：发送）

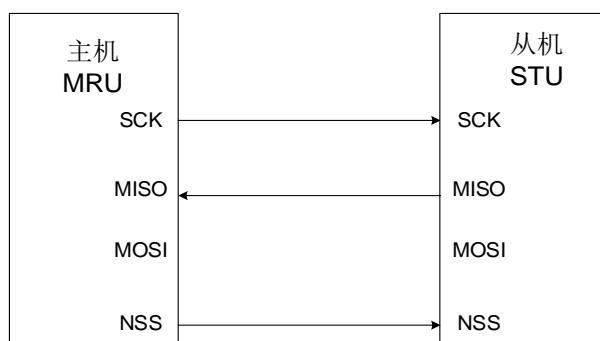


图 18-8. 典型的单工模式连接（主机：只发送，从机：接收）

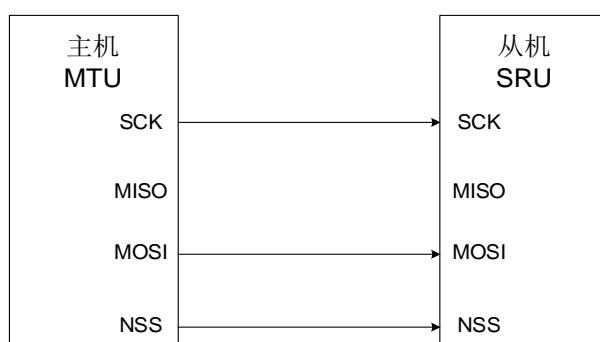
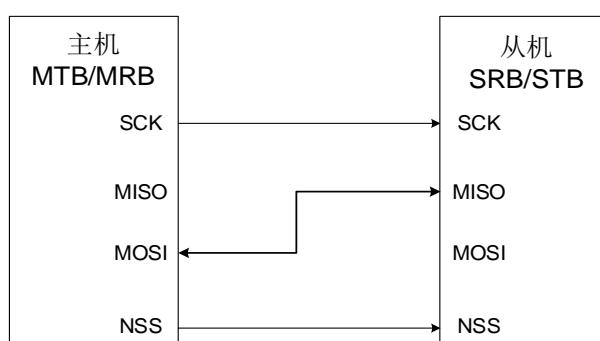


图 18-9. 典型的双向线连接



## SPI 初始化流程

### SPI0:

在发送或接收数据之前，应用程序应遵循如下的SPI初始化流程：

1. 如果工作在主机模式或从机TI模式，配置SPI\_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置数据格式（SPI\_CTL0中的FF16位）。
3. 配置时钟时序（SPI\_CTL0中的CKPL位和CKPH位）。
4. 配置帧格式（SPI\_CTL0中的LF位）。
5. 按照上文[NSS功能](#)的描述，根据应用程序的需求，配置NSS模式（SPI\_CTL0中的SWNSSEN位和NSSDRV位）。
6. 如果工作在TI模式，需要将SPI\_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI\_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。
8. 根据[表18-5. SPI运行模式](#)，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 使能SPI（将SPIEN位置1）。

**注意：**在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF位。

### SPI1:

1. 如果工作在主机模式或从机TI模式，配置SPI\_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置时钟时序（SPI\_CTL0中的CKPL位和CKPH位）。
3. 配置帧格式（SPI\_CTL0中的LF位）。
4. 配置数据格式（SPI\_CTL1的DZ[3:0]位域）和SPI\_DATA的访问方式（SPI\_CTL1的BYTEN位）。
5. 按照上文[NSS功能](#)的描述，根据应用程序的需求，配置NSS模式（SPI\_CTL0中的SWNSSEN位和NSSDRV位）。
6. 如果工作在TI模式，需要将SPI\_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI\_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。
8. 根据[表18-5. SPI运行模式](#)，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 根据应用程序的需求，配置TXDMA\_ODD和RXDMA\_ODD位。
10. 如果工作在SPI四线模式，需要将SPI\_QCTL中的QM0D位置1，如果不是，则忽略此步骤。
11. 使能SPI（将SPIEN位置1）。

**注意：**在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF、DZ[3:0]位。

## SPI 基本发送和接收流程

### 发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区/发送FIFO时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区/发送FIFO中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区/发送FIFO加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE（发送缓冲区空）位置1。TBE标志位置1，说明发送缓冲区/发送FIFO为空，此时如果需要发送更多数据，软件应该继续写SPI\_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI\_DATA寄存器中。

### 接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区/接收FIFO，且RBNE（接收缓冲区/接收FIFO非空）位置1。软件通过读SPI\_DATA寄存器获得接收的数据，此操作会自动清除RBNE标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式（MFD）中，仅当发送缓冲区/发送FIFO非空时，硬件才接收下一个数据帧。

### SPI 不同模式下的操作流程（非 SPI 四线模式，TI 模式或 NSSP 模式）

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式（MTU, MTB, STU或STB）与全双工模式中的发送流程类似，不同的是需要忽略RBNE位和RXORERR位。

相比于发送模式的情况，主机接收模式（MRU或MRB）与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区/接收FIFO内的数据，否则，将会产生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式（SRU或SRB）与全双工模式类似。

### SPI TI 模式

SPI TI模式将NSS作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式（MFD, MTU, MRU, MTB, MRB, SFD, STU, SRU, STB和SRB）都支持TI模式。但是，在TI模式中，SPI\_CTL0中的CKPL位和CKPH位是没有意义的，SCK信号的采样边沿为下降沿。

图 18-10. 主机 TI 模式在不连续发送时的时序图

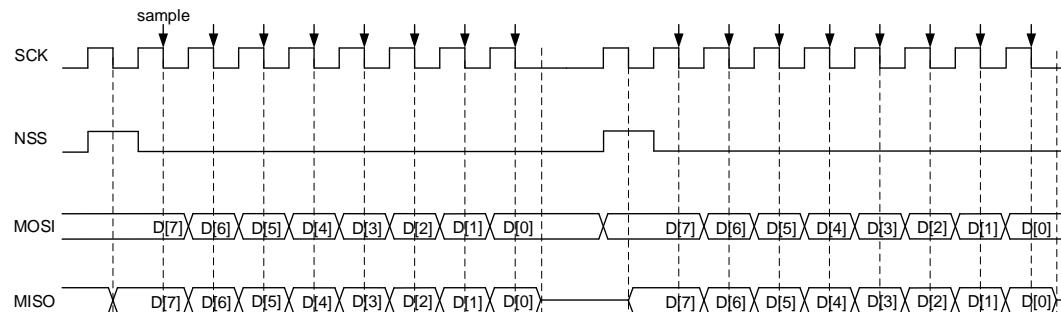
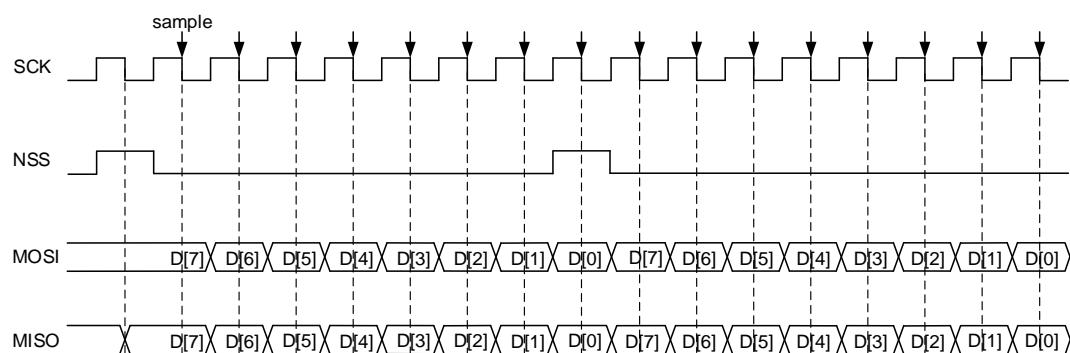
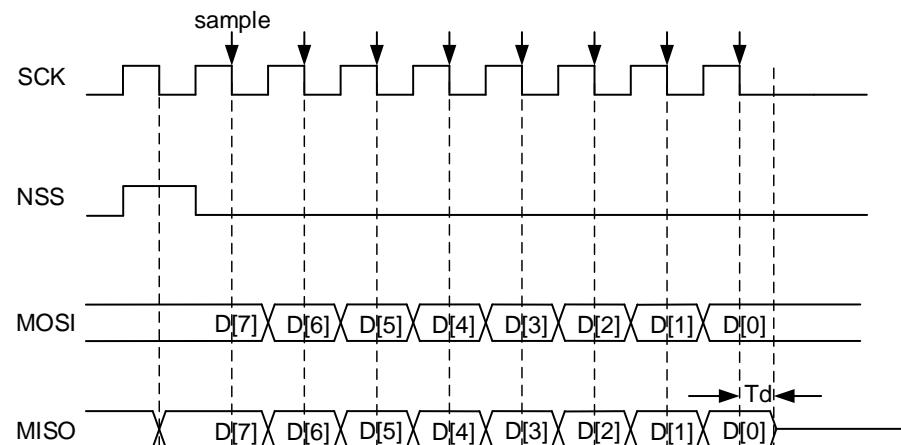


图 18-11. 主机 TI 模式在连续发送时的时序图



在主机TI模式下，SPI模块可实现连续传输或者不连续传输。如果主机写SPI\_DATA的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。在连续传输中，额外的时钟周期只存在于第一个字节之前，随后字节的起始时钟周期被前一个字节的最后一一位的时钟周期覆盖。

图 18-12. 从机 TI 模式时序图



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为 $T_d$ ， $T_d$ 通过SPI\_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (17-1)$$

例如，如果PSC[2:0] = 010，那么 $T_d$ 数值为 $9 * T_{pclk}$ 。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

### NSS 脉冲模式操作流程

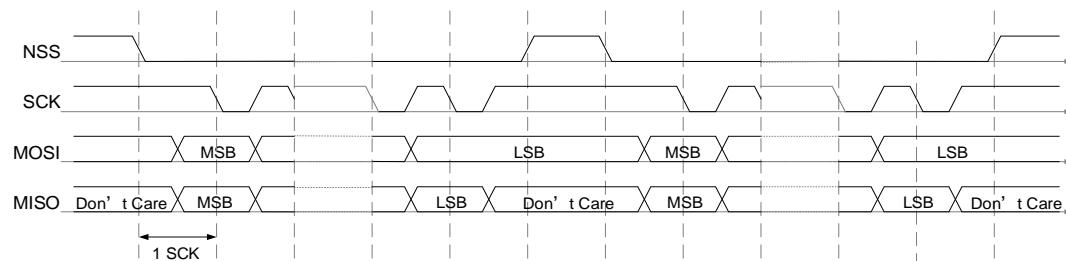
配置SPI\_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1， NSSP = 1， CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区/发送FIFO的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区/发送FIFO保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了NSS脉冲模式在主机连续发送时的时序图。

**图 18-13. NSS 脉冲模式时序图（主机连续发送）**



### SPI 四线模式操作流程

SPI四线模式用于控制四线SPI Flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI\_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI\_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、CRCNT位、RO位和LF位保持清零，DZ[3:0]位域配置数据长度为8位，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI\_QCTL寄存器中的QRD位进行配置。

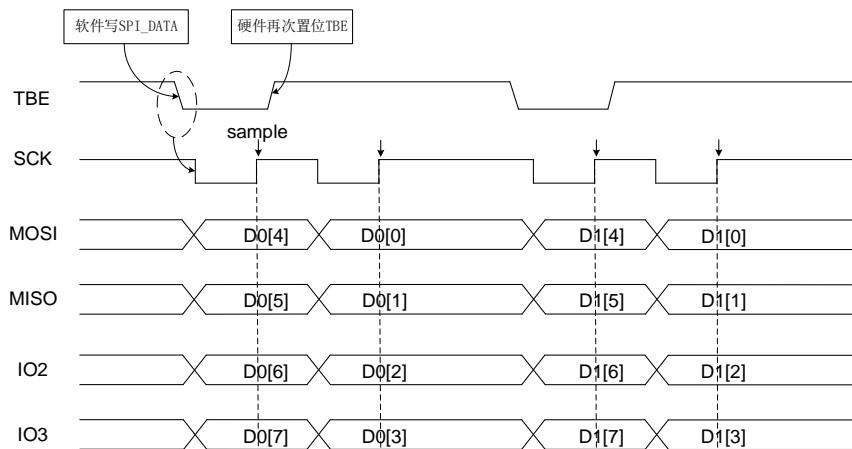
#### 四线写模式

当SPI\_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI\_DATA寄存器（TBE位清零）且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI\_CTL0和SPI\_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI\_QCTL中的QMOD位置1，然后将SPI\_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI\_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 18-14. SPI 四线模式写操作时序图



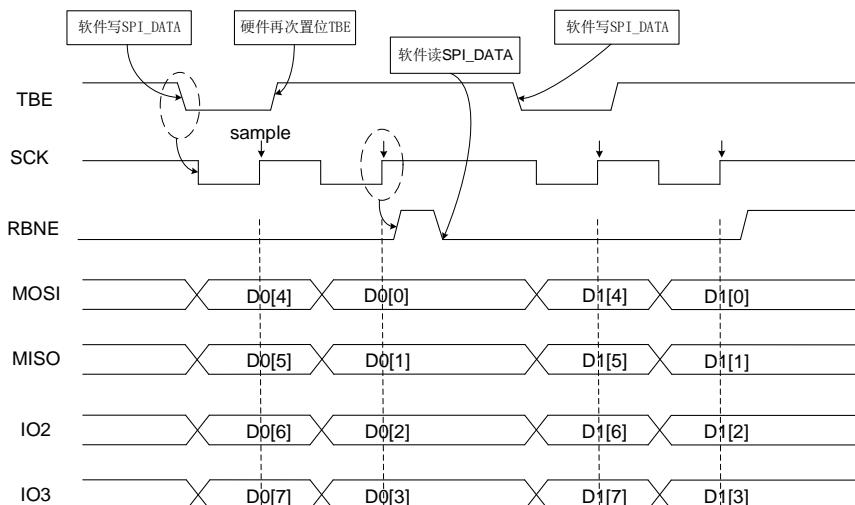
#### 四线读模式

当SPI\_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI\_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI\_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI\_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI\_CTL0和SPI\_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI\_QCTL中的QMOD位和QRD位置1，然后将SPI\_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据（例如0xFF）到SPI\_DATA寄存器；
4. 等待RBNE位置1，然后读SPI\_DATA寄存器来获取接收的数据；
5. 写任意数据（例如0xFF）到SPI\_DATA寄存器，以接收下一个字节数据。

图 18-15. SPI 四线模式读操作时序图



## SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

### MFD SFD

SPI0:

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

SPI1:

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。最后，读取数据直到RXLVL[1:0]=00。

### MTU MTB STU STB

SPI0:

将最后一个数据写入SPI\_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

SPI1:

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。

### MRU MRB

SPI0:

等待倒数第二个RBNE位置1，从SPI\_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI\_DATA读数据。

SPI1:

等待倒数第二个RBNE位置1，从SPI\_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，读取数据直到RXLVL[1:0]=00。

### SRU SRB

SPI0:

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

SPI1:

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

### TI模式

TI模式的停止流程与上面描述过程相同。

### NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

#### SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI\_QCTL中的QMOD位和SPI\_CTL0中的SPIEN位清零。

### 18.3.7. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI\_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI\_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI\_DATA寄存器读取数据。

#### DMA 数据合并传输（仅适用于 SPI1）

采用DMA进行数据传输，当BYTEN设置为0且DZ[3:0]配置的数据长度小于或等于8位且数据合并模式使能时，DMA将会以16位方式访问SPI\_DATA寄存器，自动完成数据的发送。

在数据合并模式使能且传输数据帧的帧数不是偶数倍的情况下，为了避免最后一次DMA传输多一帧数据的问题，需要将SPI\_CTL1寄存器中TXDMA\_ODD/RXDMA\_ODD位为设置为1。

### 18.3.8. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI\_CRCPOLY寄存器中定义的多项式。

通过配置SPI\_CTL0中的CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI\_TCRC寄存器和SPI\_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI\_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB，MRU，SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

对于SPI0，如果是8位数据长度，CRC计算基于CRC8标准进行。如果是16位数据长度，CRC计算基于CRC16标准进行。如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

对于SPI1，只有数据长度为8位或者16位时，SPI提供CRC计算，且独立于数据长度，可以固定设置为8位或16位CRC计算。对于其他所有的数据长度，CRC无效。CRC数据交换，通常需

要在数据序列结束后，再占用一个或多个数据通信的时间。例如，当设置为8位的数据长度并做16位CRC检查时，发送完整的CRC数据就要两帧。如果使能了DMA功能，硬件将会自动处理CRC传输和校验，但SPI需设置DMA发送通道和接收通道的计数器值。发送DMA计数器值为不包括CRC帧的数据帧的数量。接收DMA计数器值的配置如下：

1.全双工模式：假设SPI接收的数据量为L，当CRCL = 0且DZ = 8时，，则DMA接收通道的计数值等于L + 1，否则DMA接收通道的计数值等于L + 2。

2.只接收模式：DMA接收通道计数值只等于接收的数据量。接收数据完成后，通过软件读取SPI\_RCRC寄存器的方式获取CRC值。

**注意：**当SPI处于从机模式且CRC功能使能时，无论SPI是否使能，CRC计算器都对输入SCK时钟敏感。只有当时钟稳定时，软件才能启用CRC，以避免错误的CRC计算。当SPI作为从机工作时，在数据阶段和CRC阶段之间，内部NSS信号需要保持低电平。

### 18.3.9. SPI 中断

#### 状态标志位

##### ■ 发送缓冲区空标志位 (TBE)

当发送缓冲区为空或当前发送FIFO的存储量小于或等于总存储量的一半时，TBE置位。软件可以通过写SPI\_DATA寄存器将下一个待发送数据写入发送缓冲区/发送FIFO。

##### ■ 接收缓冲区非空标志位 (RBNE)

对于SPI0，当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI\_DATA寄存器来读取此数据。

对于SPI1，该位根据SPI\_CTL1中的BYTEN位设置：如果BYTEN=0，则当前接收FIFO的存储量大于或等于总存储量的1/2时，RBNE置位。如果BYTEN=1，则当前接收FIFO的存储量大于或等于总存储量的1/4时，RBNE置位。表示此时接收到数据，并已存入接收FIFO中，软件可以通过读SPI\_DATA寄存器来读取此数据。

##### ■ SPI通信进行中标志位 (TRANS)

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

#### 错误标志

##### ■ 配置错误标志 (CONFERR)

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的

多主冲突。

#### ■ 接收过载错误 (RXORERR)

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。对于SPI0，这说明，上一帧数据还未被读出而新的数据已经接收了。对于SPI1，这说明，接收FIFO没有足够的空间来存储接收到的数据了。接收缓冲区/接收FIFO的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

#### ■ 帧错误 (FERR)

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

#### ■ CRC错误 (CRCERR)

当CRCEN位置1时，SPI\_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据后接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

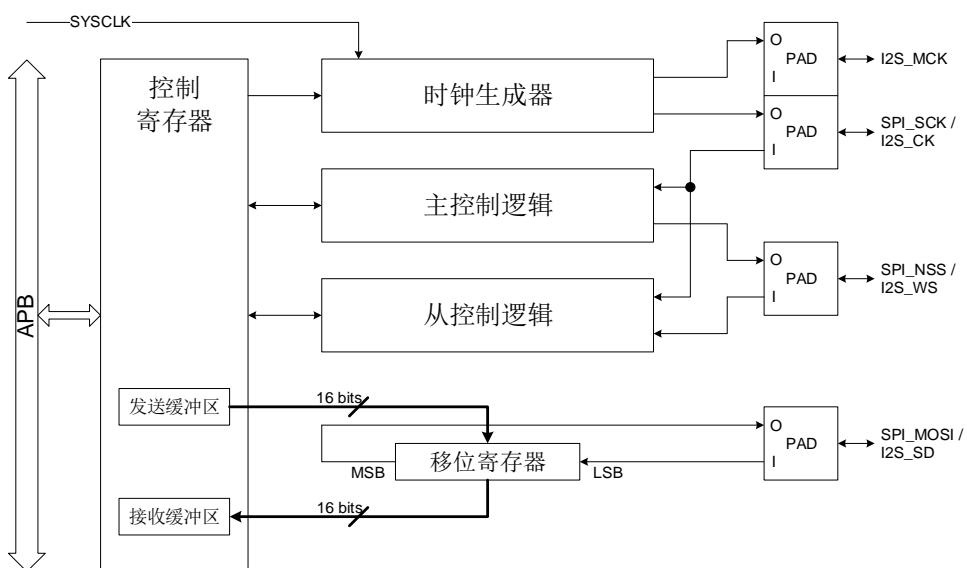
**表 18-6. SPI 中断请求**

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区/发送FIFO空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区/接收FIFO非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器	ERRIE
RXORERR	接收过载错误	读SPI_DATA寄存器，然后读 SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	
FERR	TI模式帧错误	写0到FERR位	

## 18.4. I2S 功能说明

### 18.4.1. I2S 结构框图

图 18-16. I2S 结构框图



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。时钟生成器用来在主机模式下生成I2S通信时钟。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I2S\_CK和I2S\_WS信号来控制从机模式的通信。移位寄存器控制I2S\_SD上的串行数据发送和接收。

### 18.4.2. I2S 信号线描述

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共享引脚。I2S\_WS是数据帧控制信号，与SPI\_NSS共享引脚。I2S\_SD是串行数据信号，与SPI\_MOSI共享引脚。I2S\_MCK是主时钟信号，它最大可提供一个256倍于Fs的时钟频率，其中Fs是音频采样率。

### 18.4.3. I2S 音频标准

I2S音频标准是通过设置SPI\_I2SCTL寄存器中的I2SSTD位来选择的，可以选择四种音频标准：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，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寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。用于发送和接收的数据缓冲区都是16位宽度。所以，要完成数

据长度为24位或32位的数据帧传输，SPI\_DATA寄存器需要被访问2次；而要完成数据长度为16位的数据帧传输，SPI\_DATA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将16位数据扩展为32位格式。

对于所有标准和数据包类型来说，数据的最高有效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，总是先发送左通道，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，I2S\_WS在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 18-17. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

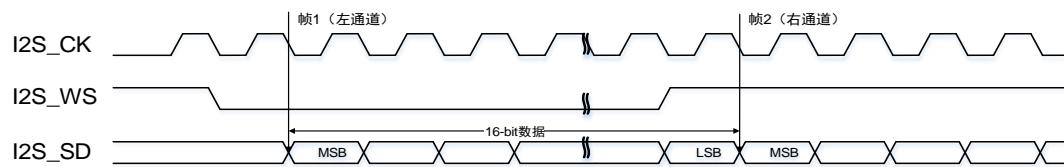
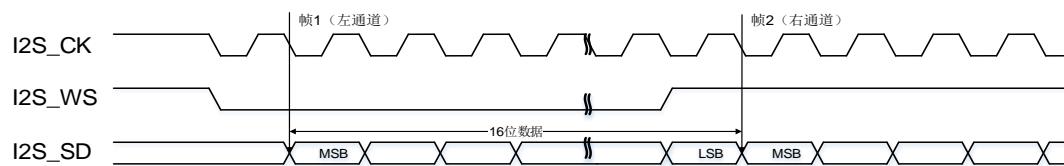


图 18-18. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI\_DATA 寄存器一次。

图 18-19. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

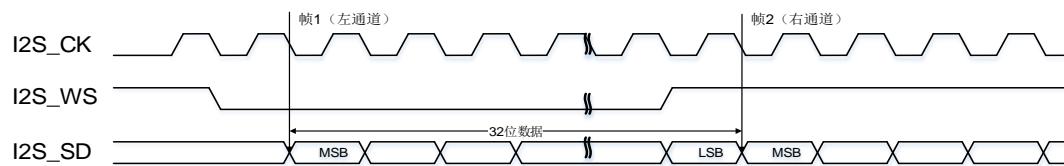
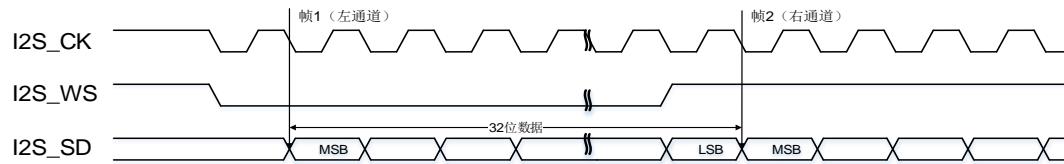


图 18-20. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



当32位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI\_DATA寄存器2次。在发送模式下，如果要发送一个32位数据，第一个写入SPI\_DATA寄存器的数据应该是高16位数据，第二个数据应该是低16位数据。在接收模式下，如果要接收一个32位数据，第一个从SPI\_DATA寄存器读到的数据应该是高16位数据，第二个数据应该是低16位数据。

图 18-21. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

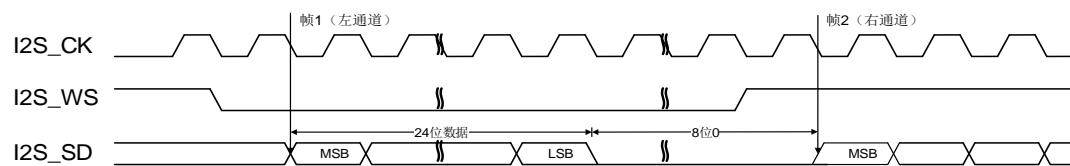
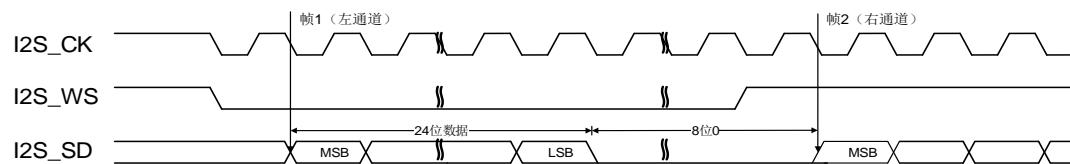


图 18-22. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI\_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI\_DATA寄存器的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据可以是任意值。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI\_DATA寄存器读到的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据全是0。

图 18-23. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

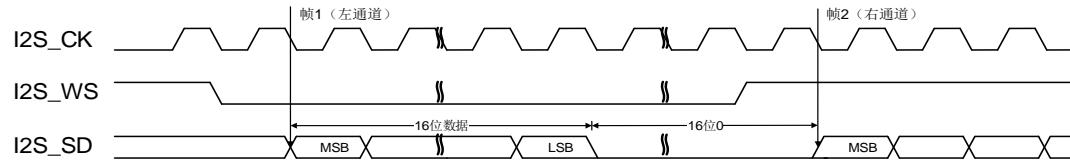
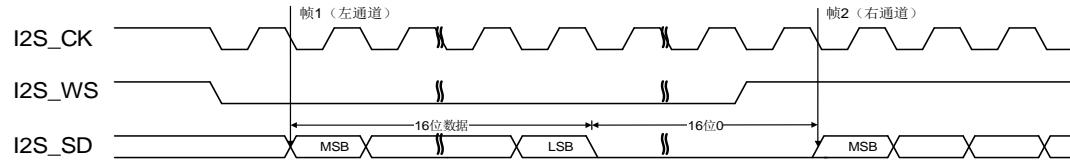


图 18-24. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI\_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为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\_DATA寄存器的处理方式与I2S飞利浦标准完全相同。各个配置情况的时序图如下所示。

图 18-25. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)

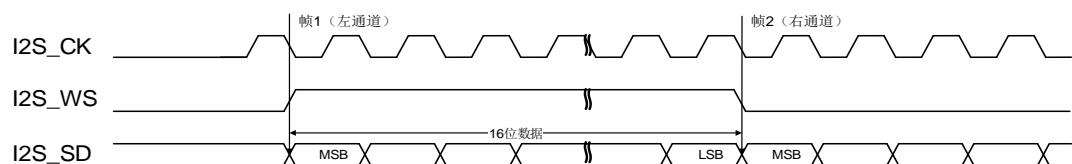


图 18-26. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)

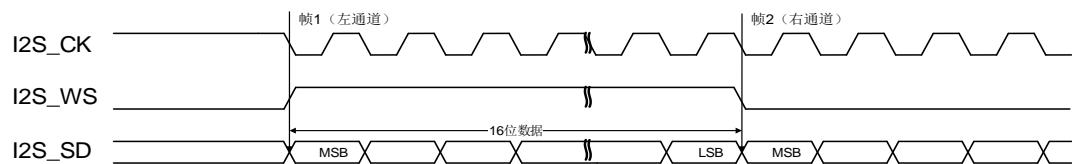


图 18-27. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)

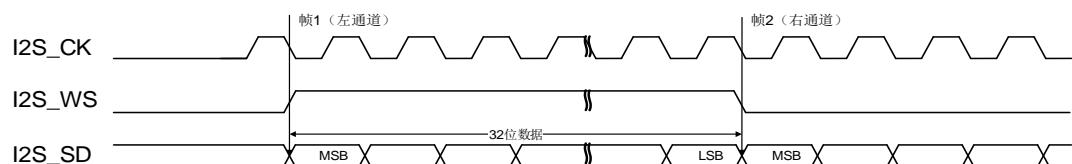


图 18-28. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)

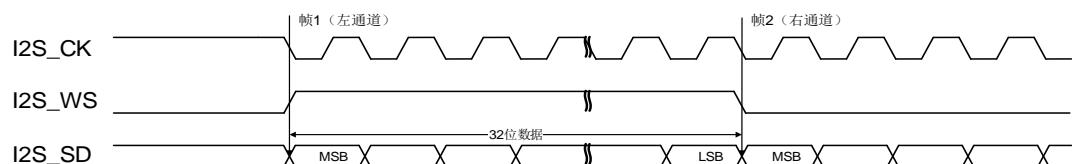


图 18-29. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

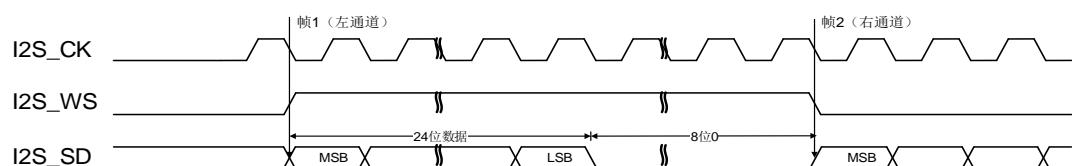


图 18-30. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)

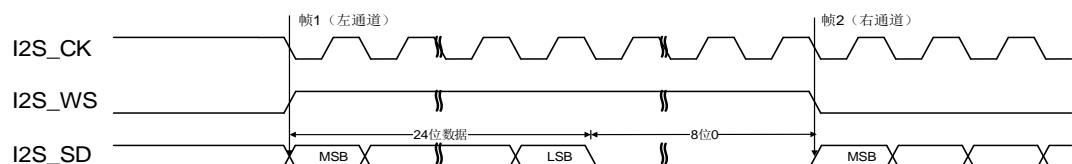


图 18-31. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

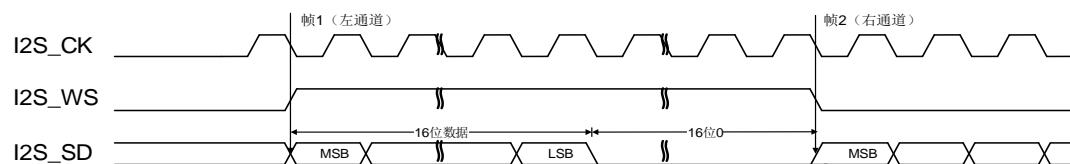
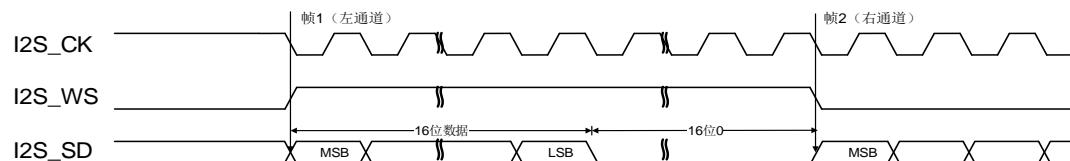


图 18-32. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



### 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 18-33. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

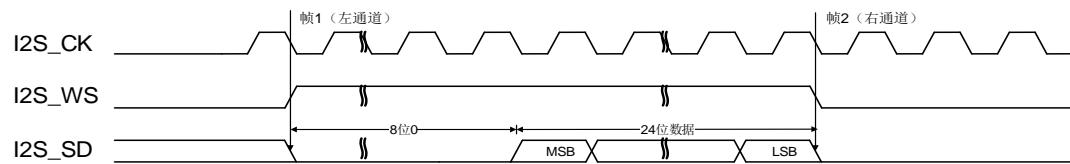
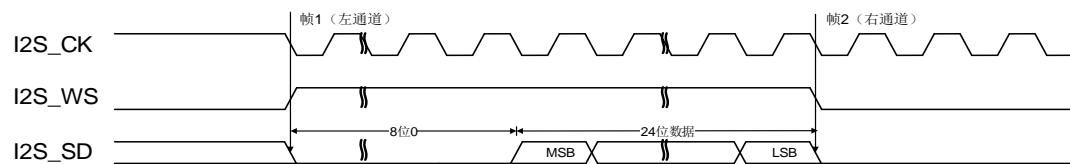


图 18-34. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI\_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI\_DATA寄存器的数据应该是一个16位数据，该16位数据的高8位可以是任意值，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI\_DATA寄存器读到的数据应该是一个16位数据，该16位数据的高8位是0，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。

图 18-35. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

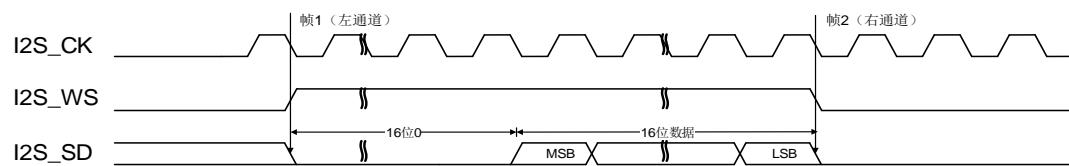
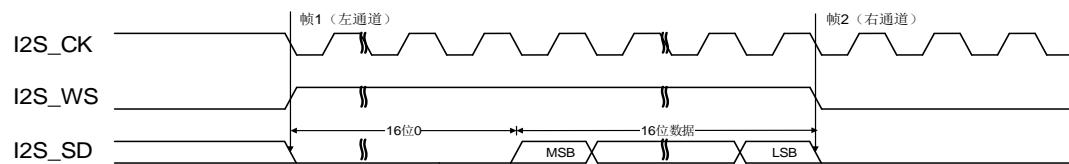


图 18-36. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问**SPI\_DATA**寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为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。可以通过**SPI\_I2SCTL**寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。**SPI\_DATA**寄存器的处理方式与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 18-37. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

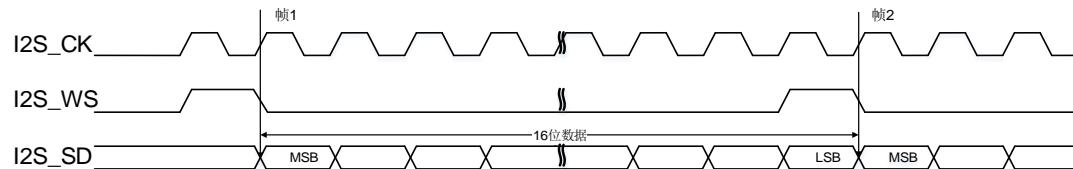


图 18-38. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

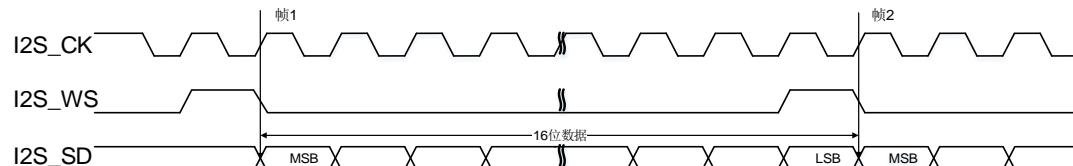


图 18-39. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

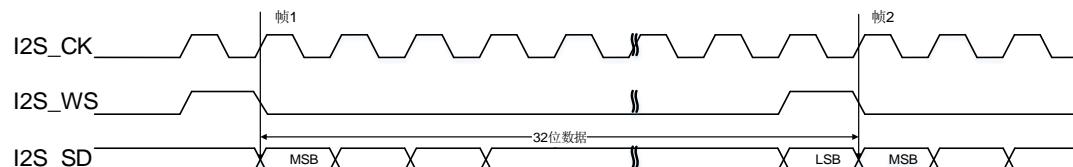


图 18-40. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

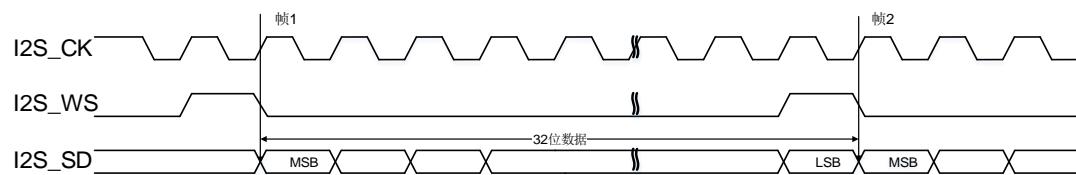


图 18-41. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

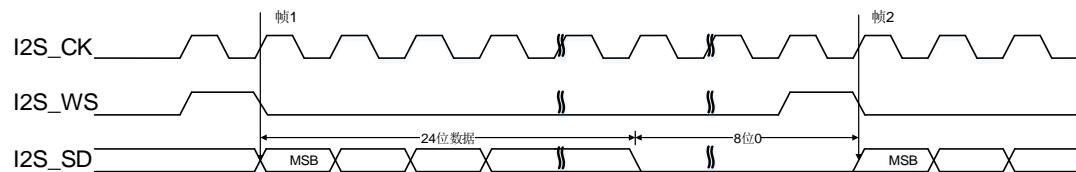


图 18-42. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

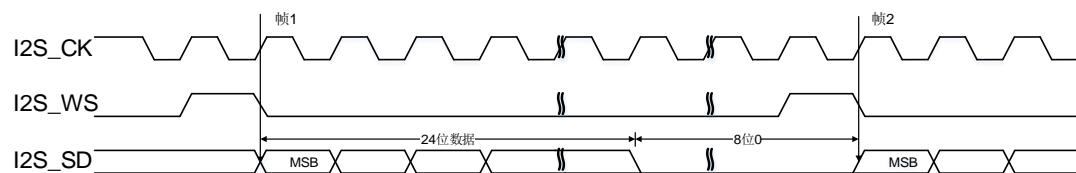


图 18-43. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

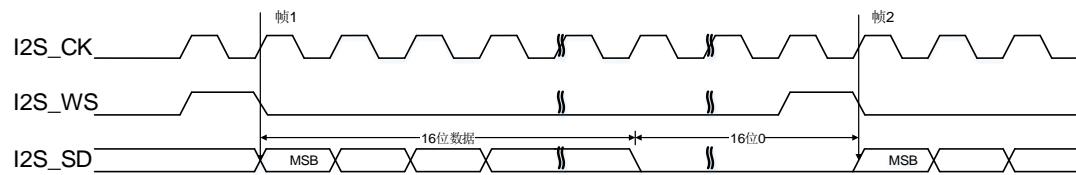
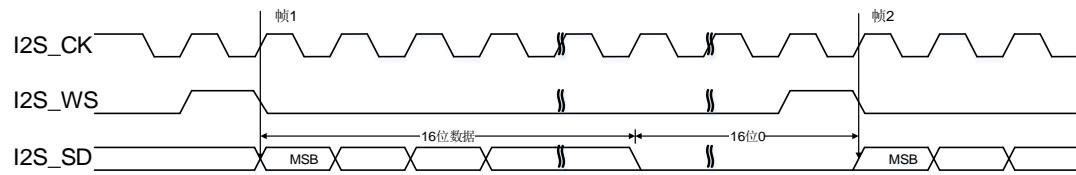


图 18-44. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 18-45. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

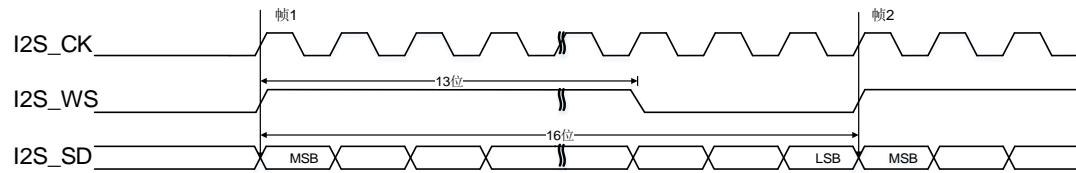


图 18-46. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

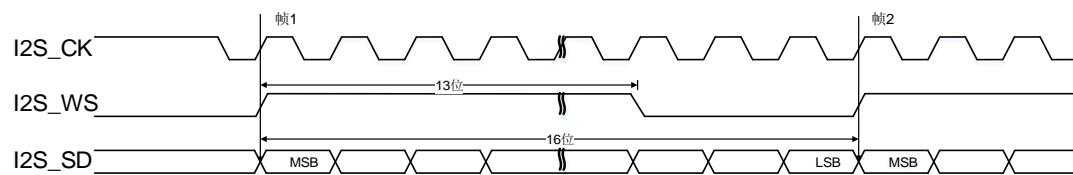


图 18-47. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

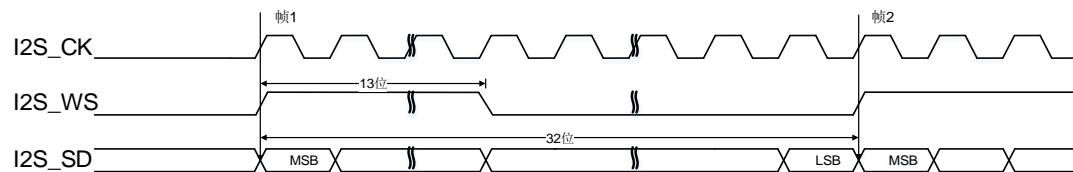


图 18-48. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

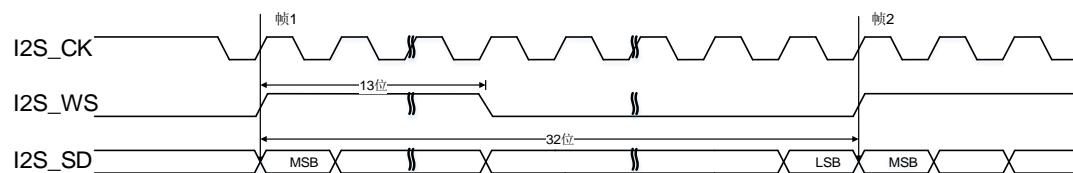


图 18-49. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

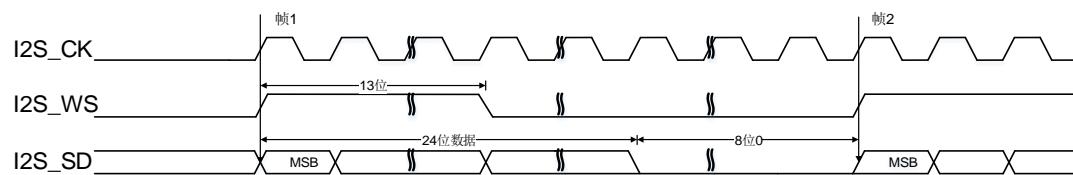


图 18-50. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

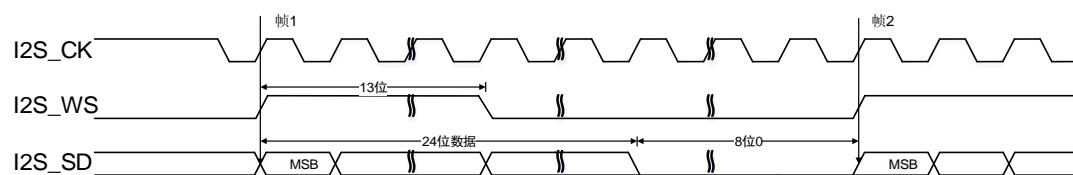


图 18-51. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

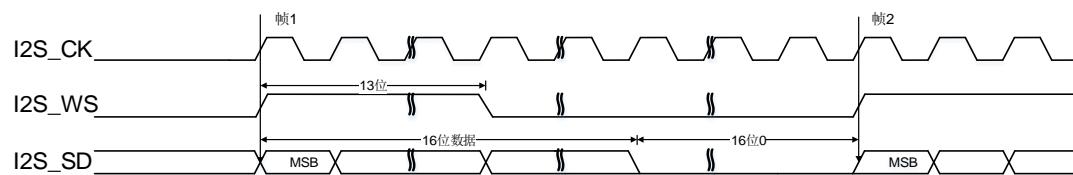
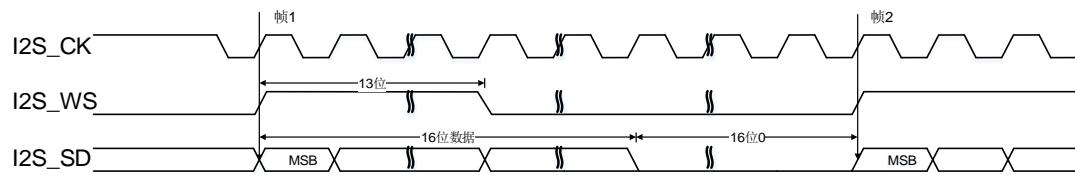
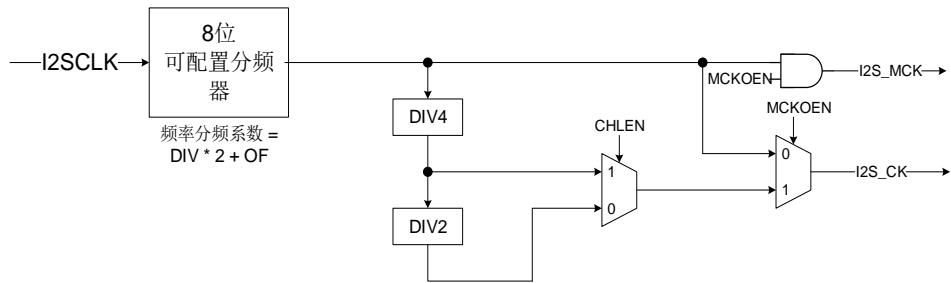


图 18-52. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



#### 18.4.4. I2S 时钟

图 18-53. I2S 时钟生成结构框图



I2S 时钟生成器框图如 [图 18-53. I2S 时钟生成结构框图](#) 所示。I2S 接口时钟是通过 SPI\_I2SPSC 寄存器的 DIV 位, OF 位和 MCKOEN 位以及 SPI\_I2SCTL 寄存器的 CHLEN 位来配置的。时钟源是系统时钟 (CK\_SYS)。I2S 比特率可以通过 [表 18-7. I2S 比特率计算公式](#) 所示的公式计算。

表 18-7. I2S 比特率计算公式

MCKOEN	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 ( $F_s$ ) 和 I2S 比特率的关系由如下公式定义：

$$F_s = \text{I2S 比特率} / (\text{通道长度} * \text{通道数})$$

所以，为了得到期望的音频采样率，时钟生成器需要按 [表 18-8. 音频采样频率计算公式](#) 所列的公式进行配置。

表 18-8. 音频采样频率计算公式

MCKOEN	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	$I2SCLK / (256 * (DIV * 2 + OF))$

### 18.4.5. 运行

#### 运行模式

运行模式是通过SPI\_I2SCTL寄存器的I2SOPMOD[1:0]位域来选择的。共有四种运行模式可供选择：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。各种运行模式下I2S接口信号的方向如[表18-9. 各种运行模式下I2S接口信号的方向](#)所示。

**表 18-9. 各种运行模式下 I2S 接口信号的方向**

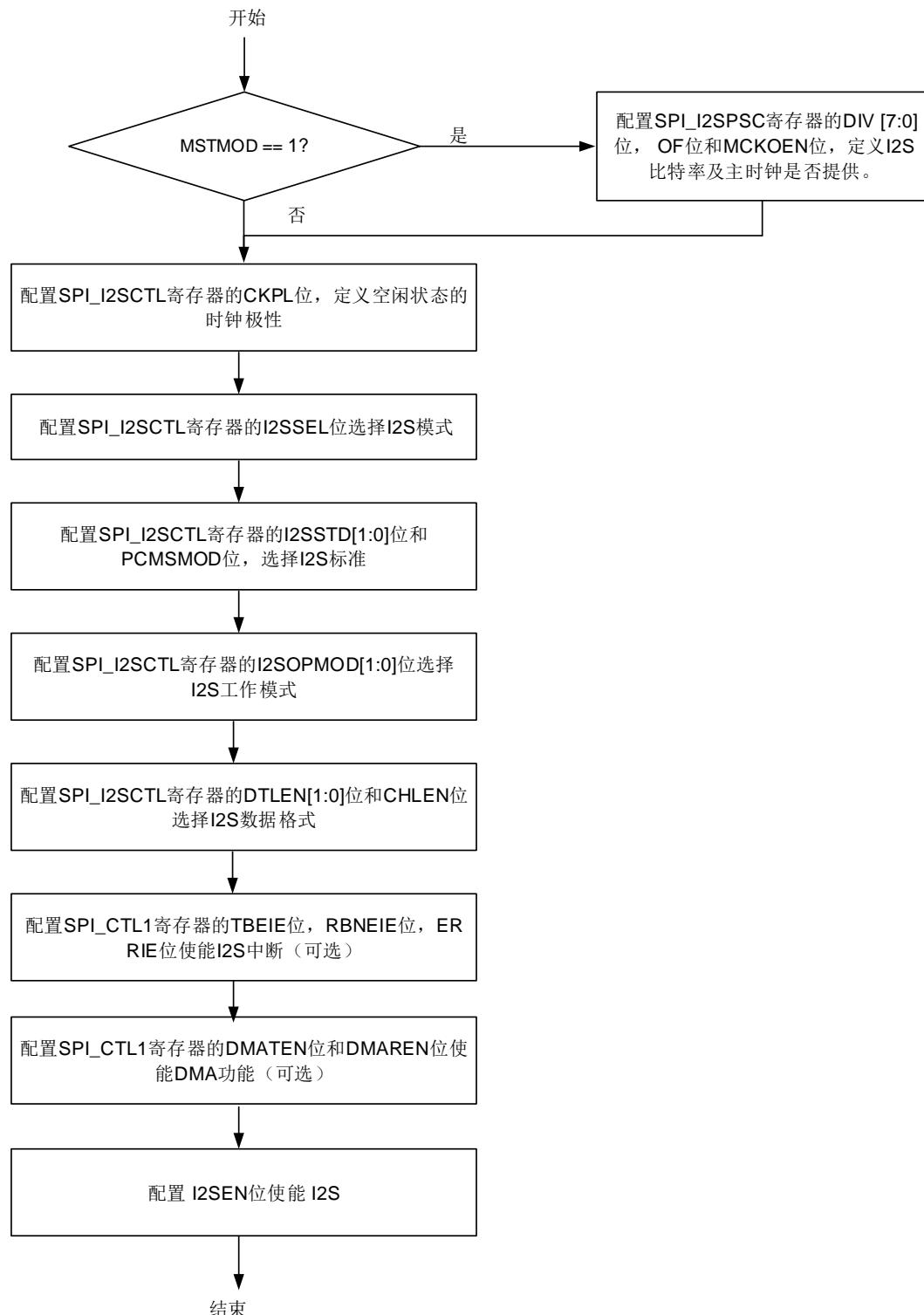
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU <sup>(1)</sup>	输出	输出	输出
主机接收	输出或 NU <sup>(1)</sup>	输出	输出	输入
从机发送	输出或 NU <sup>(1)</sup>	输入	输入	输出
从机接收	输出或 NU <sup>(1)</sup>	输入	输入	输入

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

#### I2S 初始化流程

I2S初始化过程如[图18-54. I2S初始化流程](#)所示。

图 18-54. I2S 初始化流程



### I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位置位表示发送缓冲区空，此时，如果SPI\_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空（TBE为1），且移位

寄存器中没有发送序列。当16位数据被写入SPI\_DATA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）。此时，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时写入SPI\_DATA寄存器。数据写入SPI\_DATA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。为了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\_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI\_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

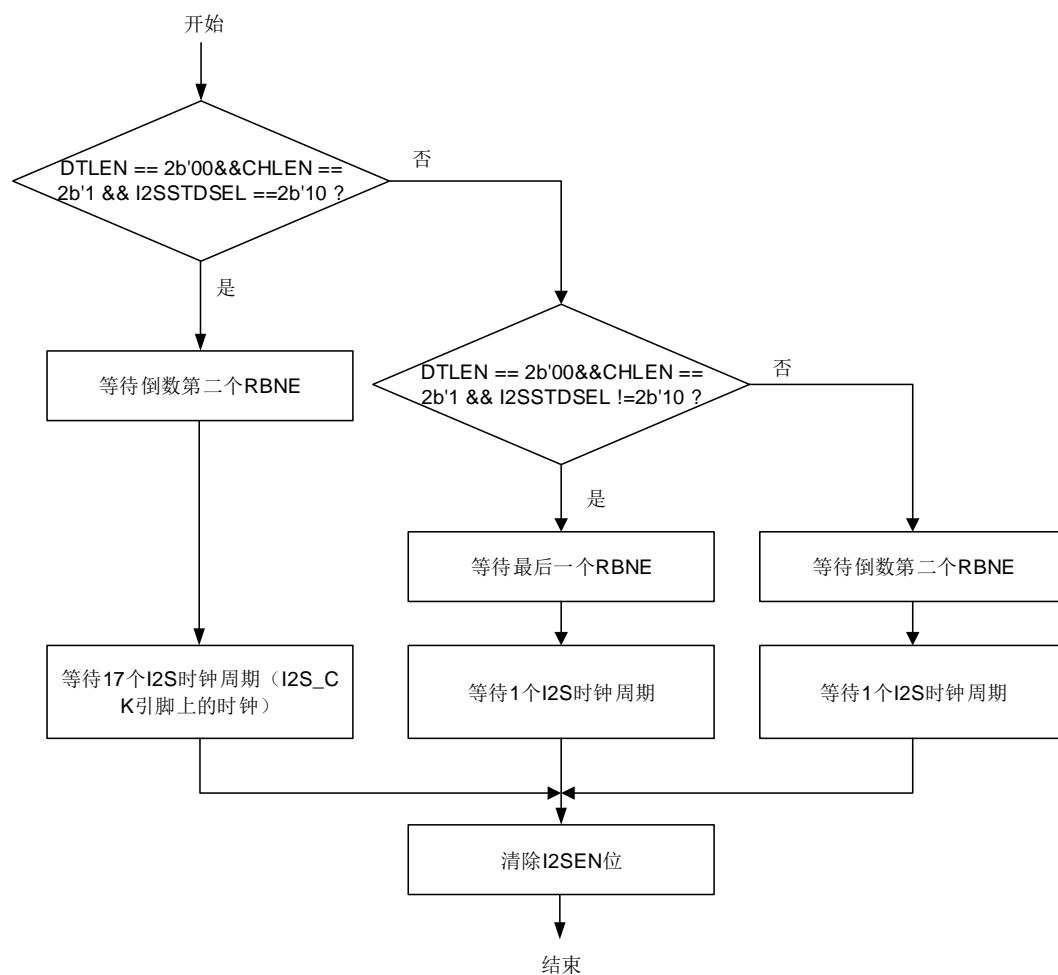
### I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志置位表示接收缓冲区非空，如果SPI\_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI\_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，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\_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI\_DATA寄存器中的数据，否则将发生接收过载错误。此时RXORERR标志位会被置1，如果SPI\_CTL1寄存器的ERRIE位为1，将会产生中断。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标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作如[图18-55. I2S 主机接收禁能流程](#)所示。

图 18-55. I2S 主机接收禁能流程



## I2S 从机发送流程

从机发送流程和主机发送流程相似，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使能。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\_DATA 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\_DATA 寄存器，否则会产生发送欠载错误。此时 TXURERR 标志会置 1，如果 SPI\_CTL1 寄存器的 ERRIE 位为 1，将会产生中断。这种情况下，必须先关闭 I2S 再打开 I2S 来恢复通讯。从机模式下，I2SCH 标志是根据外部主机发送的 I2S\_WS 信号而变化的。

为了关闭 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。

## I2S 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使能。当外部主机开始发送时钟信号且 I2S\_WS 信号指示数据开始时，接收流程开始。从机模式下，I2SCH 标志是根据外部主机发送

的I2S\_WS信号而变化的。

为了关闭I2S，必须在收到最后一个RBNE之后立即清除I2SEN位。

#### 18.4.6. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

#### 18.4.7. I2S 中断

##### 状态标志位

SPI\_STAT寄存器中有4个可用的标志位，分别是TBE、RBNE、TRANS和I2SCH，用户通过这些标志位可以全面监视I2S总线的状态。

###### ■ 发生缓冲区空标志（TBE）

当发送缓冲区为空时，TBE置位。软件可以通过写SPI\_DATA寄存器将下一个数据写入发送缓冲区。

###### ■ 接收缓冲区非空标志（RBNE）

接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读SPI\_DATA寄存器来读取此数据。

###### ■ I2S通信进行中标志（TRANS）

TRANS是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，无法进行软件操作。该标志位不会产生任何中断。

###### ■ I2S通道标志（I2SCH）

I2SCH用来表明当前传输数据的通道信息，对PCM音频标准来说没有意义。在发送模式下，I2SCH标志在每次TBE由0变1时更新，在接收模式下，I2SCH标志在每次RBNE由0变1时更新。该标志位不会产生任何中断。

##### 错误标志

有三个错误标志：

###### ■ 发送欠载错误标志（TXURERR）

在从发送模式下，有效的SCK信号开始发送，当发送缓冲区为空时，发送欠载错误标志TXURERR置位。

###### ■ 接收过载错误标志（RXORERR）

当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志RXORERR置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。

###### ■ 帧格式错误（FERR）

在从I2S模式下，I2S模块监视I2S\_WS信号，如果I2S\_WS信号在一个错误的位置发生翻转，将会置位FERR帧错误标志位。

表18-10. I2S中断总结了I2S中断事件和相应的使能位。

表 18-10. I2S 中断

中断标志	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器, 然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

## 18.5. SPI/I2S 寄存器

SPI0/I2S0基地址: 0x4001 3000

SPI1基地址: 0x4000 3800

### 18.5.1. 控制寄存器 0 (SPI\_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

该寄存器在I2S模式下没有意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16 CRCL	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC[2:0]	MSTMOD	CKPL	CKPH		
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	BDEN	双向数据模式使能 0: 2线单向传输模式。 1: 1线双向传输模式。数据在主机的MOSI引脚和从机的MISO引脚之间传输。
14	BDOEN	双向传输输出使能 当BDEN置位时，该位决定了数据的传输方向。 0: 工作在只接收模式。 1: 工作在只发送模式。
13	CRCEN	CRC计算使能 0: 禁能CRC计算。 1: 使能CRC计算。
12	CRCNT	下一次传输CRC 0: 下一次传输值为数据。 1: 下一次传输值为CRC值。 当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入SPI_DATA寄存器后应将该位置1。 在只接收模式下，在接收完倒数第二个数据后应将该位置1。
11	FF16	数据帧格式（只有SPI0） 0: 8位数据帧格式

		1: 16位数据帧格式
	CRCL	CRC长度（只有SPI1） 0: 8位CRC长度 1: 16位CRC长度
10	RO	只接收模式 当BDEN清零时，该位决定了数据的传输方向。 0: 全双工模式 1: 只接收模式
9	SWNSSEN	NSS软件模式使能 0: NSS硬件模式，NSS电平取决于NSS引脚。 1: NSS软件模式，NSS电平取决于SWNSS位。 该位在SPI TI模式下没有意义。
8	SWNSS	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置位时，该位有效。 该位在SPI TI模式下没有意义。
7	LF	最低有效位先发模式 0: 先发送最高有效位。 1: 先发送最低有效位。 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0: 禁能SP 1: 使能SPI
5:3	PSC[2:0]	主时钟预分频选择 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: PCLK/256 当使用SPI0时，PCLK=PCLK2。当使用SPI1时，PCLK=PCLK1。
2	MSTMOD	主从模式使能 0: 从机模式 1: 主机模式
1	CKPL	时钟极性选择 0: SPI为空闲状态时，CLK引脚拉低。

1: SPI为空闲状态时，CLK引脚拉高。

0	CKPH	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。
---	------	--

### 18.5.2. 控制寄存器 1 (SPI\_CTL1)

地址偏移: 0x04

复位值: SPI0: 0x0000 0000

SPI1: 0x0000 0700

该寄存器可以按字节（8位）或半字（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	TXDMA_ODD	RXDMA_ODD	BYTEN	DZ[3:0]	TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN			
	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	TXDMA_ODD	DMA发送通道奇数字节（只有SPI1）  在数据合并传输模式中，当通过DMA发送的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时（数据长度小于等于8位且对SPI_DATA写入访问是16位宽）有效。  必须在SPI禁止时写入。  0: 通过DMA发送的数据总量为偶数个。 1: 通过DMA发送的数据总量为奇数个。
13	RXDMA_ODD	DMA接收通道奇数字节（只有SPI1）  在数据合并传输模式中，当通过DMA接收的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时（数据长度小于等于8位且对SPI_DATA写入访问是16位宽）有效。  必须在SPI禁止时写入。  0: 通过DMA接收的数据总量为偶数个。 1: 通过DMA接收的数据总量为奇数个。
12	BYTEN	字节访问使能（只有SPI1）  该位用于指示对FIFO的访问宽度，并设置产生RBNE的RXFIFO的阈值。  0: 半字访问，且当RXLVL>=2时，RBNE置位。 1: 字节访问，且当RXLVL>=1时，RBNE置位。
11:8	DZ[3:0]	数据位宽（只有SPI1）

这些位配置SPI传输数据的位宽。

0000: 强制为“0111”

0001: 强制为“0111”

0010: 强制为“0111”

0011: 4位

0100: 5位

.....

1111: 16位

7	TBEIE	发送缓冲区/发送FIFO空中断使能 0: 禁能TBE中断 1: 使能TBE中断。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区/接收FIFO非空中断使能 0: 禁能RBNE中断 1: 使能RBNE中断。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0: 禁能错误中断 1: 使能错误中断。当CRCERR位，CONFERR位，FERR位，RXORERR位或者TXURERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0: 禁能SPI TI模式 1: 使能SPI TI模式
3	NSSP	SPI NSS脉冲模式使能 0: 禁能SPI NSS脉冲模式 1: 使能SPI NSS脉冲模式
2	NSSDRV	NSS输出使能 0: 禁能主机NSS输出 1: 使能主机NSS输出
1	DMATEN	发送缓冲区/发送FIFO DMA使能 0: 禁能发送缓冲区/发送FIFO DMA 1: 使能发送缓冲区/发送FIFO DMA。当SPI_STAT中的TBE置位时，将会在相应的DMA通道上产生一个DMA请求。
0	DMAREN	接收缓冲区/接收FIFO DMA使能 0: 禁能接收缓冲区/接收FIFO DMA 1: 使能接收缓冲区/接收FIFO DMA。当SPI_STAT中的RBNE置位时，将会在相应的DMA通道上产生一个DMA请求。

### 18.5.3. 状态寄存器 (SPI\_STAT)

地址偏移: 0x08

复位值: 0x0000 0002

该寄存器可以按字节（8位）或半字（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	TXLVL[1:0]	RXLVL[1:0]	FERR	TRANS	RXORER	CONFER	CRCERR	TXURER	I2SCH	TBE	RBNE				

r r rc\_w0 r r r rc\_w0 r r r r r r r

位/位域	名称	描述
31:13	保留	必须保持复位值。
12:11	TXLVL[1:0]	<p>发送FIFO状态（只有SPI1）</p> <p>00: 空</p> <p>01: 1/4满</p> <p>10: 1/2满</p> <p>11: 满</p> <p><b>注意:</b> 这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。</p>
10:9	RXLVL[1:0]	<p>接收FIFO状态（只有SPI1）</p> <p>00: 空</p> <p>01: 1/4满</p> <p>10: 1/2满</p> <p>11: 满</p> <p>这些位在打开了CRC计算功能时的SPI只接收模式下，不适用。</p> <p><b>注意:</b> 这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。</p>
8	FERR	<p>帧错误</p> <p><b>SPI TI 模式:</b></p> <p>0: 没有 TI 模式帧错误发生</p> <p>1: TI 模式帧错误发生</p> <p><b>I2S 模式:</b></p> <p>0: 没有 I2S 帧错误发生</p> <p>1: I2S 帧错误发生</p> <p>该位由硬件置位，可以通过写0清除。</p>
7	TRANS	<p>通信进行中标志</p> <p>0: SPI或I2S空闲</p> <p>1: SPI或I2S目前正在发送且/或接收数据。</p> <p>该位由硬件置位和清除。</p>
6	RXORERR	接收过载错误标志

		0: 没有接收过载错误发生 1: 接收过载错误发生 该位由硬件置位，软件序列清零。软件序列为：先读SPI_DATA寄存器，然后读SPI_STAT寄存器。
5	CONFERR	SPI配置错误 0: 无配置错误发生 1: 配置错误发生（主机模式下，在硬件NSS模式时NSS引脚被拉低，或者软件NSS模式时SWNSS位为0，都会产生CONFERR错误。） 该位由硬件置位，软件序列清零。软件序列为：先写SPI_STAT寄存器，然后读或写SPI_CTL0寄存器。 I2S模式下不使用该位。
4	CRCERR	SPI CRC错误标志 0: SPI_RCRC值等于最后接收到的CRC值。 1: SPI_RCRC值不等于最后接收到的CRC值该位由硬件置位，可以通过写0清除。 I2S模式下不使用该位。
3	TXURERR	发送欠载错误标志 0: 无发送欠载错误发生 1: 发送欠载错误发生 该位由硬件置位，通过写SPI_STAT寄存器清除。 SPI模式下不使用该位。
2	I2SCH	I2S通道标志 0: 下一个将要发送或接收的数据属于左通道 。 1: 下一个要发送或接收的数据属于右通道 。 该位由硬件置位和清除。 SPI模式下不使用该位， I2S PCM模式下该位没有意义。
1	TBE	发送缓冲区/发送FIFO空 0: 发送缓冲区/发送FIFO非空 1: 发送缓冲区/发送FIFO空
0	RBNE	接收缓冲区/接收FIFO非空 0: 接收缓冲区/接收FIFO空 1: 接收缓冲区/接收FIFO非空

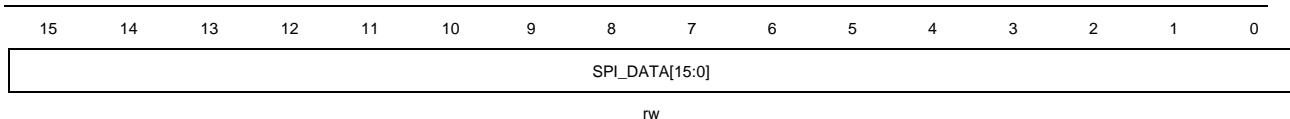
#### 18.5.4. 数据寄存器 (**SPI\_DATA**)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器可以按字节（8位）或半字（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															



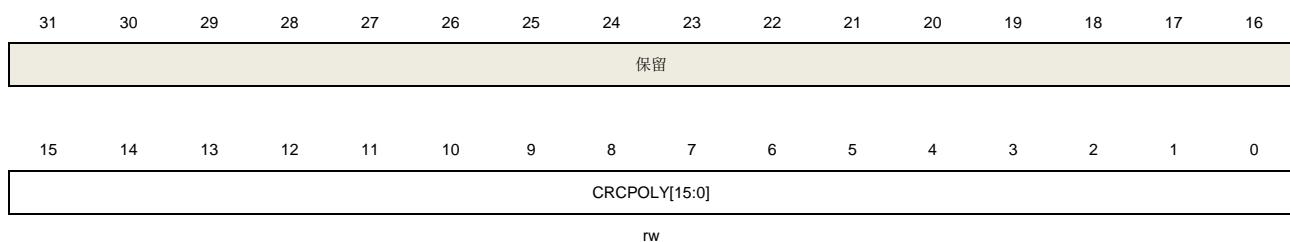
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	<p>数据传输寄存器值</p> <p>对于SPI0，硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。</p> <p>对于SPI1，硬件有两个FIFO：TXFIFO和RXFIFO。向SPI_DATA写数据将会把数据存入发送FIFO，从SPI_DATA读数据，将从接收FIFO获得数据。</p> <p><b>注意：</b>对于SPI1，实际上硬件只根据配置好的BYTEN这一位来判断每一次访问SPI_DATA的位宽，与软件当前操作所使用的位宽无关。</p>

### 18.5.5. CRC 多项式寄存器 (SPI\_CRCPOLY)

地址偏移: 0x10

复位值: 0x0000 0007

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	<p>CRC多项式寄存器值</p> <p>该值包含了CRC多项式，用于CRC计算，默认值为0007h。</p>

### 18.5.6. 接收 CRC 寄存器 (SPI\_RCRC)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。



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

r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	<p>接收CRC寄存器值</p> <p>当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCRC寄存器中。对于SPI0，如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。对于SPI1，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在RCRC[7:0]中，否则CRC计算基于CRC16标准进行，并将值保存在RCRC[15:0]中。硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。</p> <p>当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。</p>

### 18.5.7. 发送 CRC 寄存器 (SPI\_TCRC)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字节（8位）或半字（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															

r

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	<p>发送CRC寄存器值</p> <p>当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。对于SPI0，如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCRC[15:0]。对于SPI1，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在TCRC[7:0]中，否则CRC计算基于CRC16标准进行，并将值保存在TCRC[15:0]中。硬件在发送出每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。不同的数据帧格式(SPI_CTL0中的LF位决定)将会得到不同的CRC值。</p>

当SPI\_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

### 18.5.8. I2S 控制寄存器 (SPI\_I2SCTL)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器可以按字节（8位）或半字（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	I2SEL	I2SEN	I2SOPMOD[1:0]	PCMS MOD	保留	I2SSTD[1:0]	CKPL	DTLEN[1:0]	CHLEN						

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	I2SEL	I2S模式选择 0: SPI模式 1: I2S模式 当SPI模式或I2S模式关闭时配置该位。
10	I2SEN	I2S使能 0: 禁能I2S 1: 使能I2S SPI模式不使用该位。
9:8	I2SOPMOD[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 当I2S关闭时配置该位。SPI模式不使用该位。
7	PCMSMOD	PCM帧同步模式 0: 短帧同步 1: 长帧同步 只有在PCM标准下，该位才有意义。 当I2S关闭时配置该位。SPI模式不使用该位。
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  
 当I2S关闭时配置该位。**SPI**模式不使用该位。

3	<b>CKPL</b>	空闲状态时钟极性 0: I2S_CK空闲状态为低电平 1: I2S_CK空闲状态为高电平 当I2S关闭时配置该位。 <b>SPI</b> 模式不使用该位。
2:1	<b>DTLEN[1:0]</b>	数据长度 00: 16位 01: 24位 10: 32位 11: 保留 当I2S关闭时配置该位。 <b>SPI</b> 模式不使用该位。
0	<b>CHLEN</b>	通道长度 0: 16位 1: 32位 通道长度必须大于或等于数据长度。 当I2S关闭时配置该位。 <b>SPI</b> 模式不使用该位。

### 18.5.9. I2S 时钟预分频寄存器 (**SPI\_I2SPSC**)

地址偏移: 0x20

复位值: 0x0000 0002

该寄存器可以按字节(8位)或半字(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					MCKOEN	OF	DIV[7:0]								
					rw	rw	rw								

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	<b>MCKOEN</b>	I2S_MCK输出使能 0: 禁能I2S_MCK输出 1: 使能I2S_MCK输出 当I2S关闭时配置该位。 <b>SPI</b> 模式不使用该位。
8	<b>OF</b>	预分频器的奇系数

0: 实际分频系数为DIV \* 2  
 1: 实际分频系数为DIV \* 2 + 1  
 当I2S关闭时配置该位。SPI模式下不使用该位。

7:0            DIV[7:0]            预分频器的分频系数  
 实际分频系数是DIV \* 2 + OF。  
 DIV不能为0。  
 当I2S关闭时配置该位。SPI模式下不使用该位。

### 18.5.10. SPI1 四线 SPI 控制寄存器 (SPI\_QCTL)

地址偏移: 0x80

复位值: 0x0000 0000

该寄存器可以按字节（8位）或半字（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												<input type="checkbox"/> IO23_DRV V	<input type="checkbox"/> QRD	<input type="checkbox"/> QMOD	
rw          rw          rw															

位/位域	名称	描述
31:3	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 该位仅适用于SPI1。
1	QRD	四线SPI模式读选择 0: SPI四线模式写操作 1: SPI四线模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI1。
0	QMOD	四线SPI模式使能 0: SPI工作在单路模式 1: SPI工作在四线模式 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI1。

## 19. 运算放大器 (OPA)

本章内容适用于 GD32E231xx 产品

### 19.1. 简介

两个运算放大器均为低噪声、低电压、低功耗运算放大器，高增益带宽积为 6 MHz，转换速率为 5 V/μs，最大输入偏移电压为 3.5mV，输入共模范围超出电源轨。

### 19.2. 主要特性

- 与 ADC 组合工作
- 低偏移电压：1mV（典型）
- 高增益：95dB（典型）
- 高增益带宽：6MHz
- 轨到轨的输入输出
- 低输入失调电压：+2.7 V 至 +3.6 V
- 低功耗模式：3.3 V 工作电压时 600μA（每个放大器）

### 19.3. 功能描述

#### 19.3.1. 使能运放

有两个 OPA，由 ENAB 引脚启用。如果 ENAB 为 0，则禁用 OPA。PA6 / PB1 / PA14 / PA13 用作一般 GPIO 功能。如果 ENAB 引脚连接到 1 或浮空（ENAB 引脚上有内部弱上拉），则 OPA 使能。PA6 / PB1 / PA14 / PA13 需要配置为模拟或输入浮动模式。

#### 19.3.2. 使能运放

OPA 输出 PA6 / PB1 可由内部 ADC 采样。PA6 / PB1 必须配置为模拟模式。

#### 19.3.3. 使能运放

PA13 / PA14 与 OPA 输入复用。如果要使用 SW 功能，必须配置 PA13 / PA14 至 AF0 (SW 功能)。如果要使用 OPA 功能，必须将 PA13 / PA14 配置为模拟或输入浮动模式。

## 20. 附录

### 20.1. 寄存器表中使用的缩写列表

表 20-1. 寄存器功能位访问属性

功能位访问属性	描述
读/写(rw)	软件可以对这个位进行读写。
只读(r)	软件只能对这个位进行读。
只写(w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零(rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零(rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转(t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发 (rt_w1)	软件可以读该位，写入 1 触发事件，但对位值没有影响。

### 20.2. 术语表

表 20-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据.
字节	8 位长度数据
IAP(应用内编程)	IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。
ICP(在线编程)	ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。
选项字节	存储在闪存中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

### 20.3. 可用外设

对于各个 MCU 系列的外设及其数量，请参考相应型号的数据手册。

## 21. 版本历史

表 21-1. 版本历史

版本号	描述	日期
1.0	初稿发布	2019 年 3 月 8 日
1.1	1. 修改 TIMER 寄存器的访问方式及复位值。 2. 删除图 1-2 中的 I-BUS 和 D-BUS 描述。 3. 添加 TIMER 模块寄存器中[31:16]位域描述。 4. 修改 I2C_FMPCMG 寄存器复位值。	2019 年 10 月 8 日
1.2	1. 修改 FMC_WS 寄存器复位值为 0x00000030。 2. 修改 TMIER 章节的 EAPWM 时序图 CAPWM 时序图。	2020 年 3 月 21 日
1.3	1. 修改 RCU 章节的时钟树。 2. 修改 I2C 章节的图 17-1。 3. 将 I2C 模块中 17.3.11 章节的 SMBTYPE 替换为 SMBSEL。 4. 在 WDGT 模块的 12.1.3 章节中，添加关于喂完狗后要立刻进 deepsleep/standby 模式的注意事项。 5. 在 ADC 模块的 10.4.3 章节中，添加关于 ADC 使能后延时的说明。	2020 年 7 月 2 日
1.4	1. 修改 USART 章节中 USART <u>控制寄存器 1 (USART CTL1)</u> CKEN 位域描述。 2. PMU 章节 <u>VDD</u> 域中，增加 VDDA 监视器相关描述。 3. 更新 <u>表 5-1. Cortex-M23 中的 NVIC 异常类型</u> 内容。 4. 修改 I2C 章节中 <u>传输状态寄存器 0 (I2C_STAT0)</u> bit15 描述。	2021 年 6 月 7 日
1.5	1. 更新 <u>电源管理单元 (PMU)</u> 章节。 2. 更新 <u>串行外设接口/片上音频接口 (SPI/I2S)</u> 章节。 3. 更新 <u>内部集成电路总线接口 (I2C)</u> 章节。 4. 更新 <u>通用输入输出接口 (GPIO)</u> 章节。 5. 更新 <u>复位和时钟单元 (RCU)</u> 章节。 6. 更新 <u>模拟数字转换器 (ADC)</u> 章节。 7. 更新 <u>比较器 (CMP)</u> 章节。 8. 更新 <u>定时器 (TIMER)</u> 章节。 9. 更新 <u>通用同步异步收发器 (USART)</u> 章节。 10. 修改 <u>表 12.1. 独立看门狗定时器在 40kHz (IRC40K) 时的最小/最大超时周期</u> 中的超时时间。	2022 年 7 月 18 日

## Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company under the intellectual property laws and treaties of the People's Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as its suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.