

Nation  国民技术

N32G05x 系列

基于 32 位 ARM Cortex[®]-M0 微控制器

用户手册 V1.1.0

目录

1 文中的缩写	1
1.1 寄存器描述表中使用的缩写列表	1
1.2 可用的外设	1
2 存储器和总线架构	2
2.1 系统架构	2
2.1.1 总线架构	2
2.1.2 总线地址映射	4
2.1.3 启动管理	6
2.2 存储系统 (Memory system)	8
2.2.1 FLASH 规格	8
2.2.2 SRAM	18
2.2.3 FLASH 寄存器描述	19
3 电源控制 (PWR)	29
3.1 通用描述	29
3.1.1 电源	29
3.1.2 电压监控	30
3.2 低功耗模式	32
3.2.1 SLEEP 模式	34
3.2.2 STOP 模式	34
3.2.3 PD 模式	35
3.3 Debug 模式	35
3.3.1 低功耗模式调试支持	35
3.3.2 外设调试支持	36
3.4 PWR 寄存器	36
3.4.1 寄存器总览	36
3.4.2 电源控制寄存器 (PWR_CTRL)	38
3.4.3 电源控制状态寄存器 (PWR_CTRLSTS)	39
3.4.4 电源控制寄存器 2 (PWR_CTRL2)	41
3.4.5 电源控制寄存器 3 (PWR_CTRL3)	42
3.4.6 电源控制寄存器 4 (PWR_CTRL4)	43
3.4.7 电源控制寄存器 5 (PWR_CTRL5)	44
3.4.8 电源控制寄存器 6 (PWR_CTRL6)	44
3.4.9 调试控制寄存器 (DBG_CTRL)	45
4 复位和时钟控制(RCC)	48
4.1 复位控制单元	48
4.1.1 电源复位	48
4.1.2 系统复位	48
4.2 时钟控制单元	50
4.2.1 时钟树	52
4.2.2 HSE 时钟	52
4.2.3 HSI 时钟	53
4.2.4 PLL 时钟	54
4.2.5 LSI 时钟	54
4.2.6 系统时钟(SYSCLK)选择	55
4.2.7 时钟安全系统(CLKSS)	55

4.2.8 RTC 时钟	55
4.2.9 看门狗时钟	55
4.2.10 时钟输出(MCO)	55
4.3 RCC 寄存器	57
4.3.1 寄存器总览	57
4.3.2 时钟控制寄存器(RCC_CTRL)	59
4.3.3 时钟配置寄存器(RCC_CFG)	60
4.3.4 时钟中断寄存器 (RCC_CLKINT)	63
4.3.5 APB2 外设复位寄存器 (RCC_APB2PRST)	65
4.3.6 APB1 外设复位寄存器 (RCC_APB1PRST)	67
4.3.7 AHB 外设时钟使能寄存器 (RCC_AHBPCLEN)	69
4.3.8 APB2 外设时钟使能寄存器 (RCC_APB2PCLEN)	70
4.3.9 APB1 外设时钟使能寄存器 (RCC_APB1PCLEN)	72
4.3.10 低速时钟控制寄存器 (RCC_LSCTRL)	74
4.3.11 控制/状态寄存器 (RCC_CTRLSTS)	75
4.3.12 AHB 外设复位寄存器 (RCC_AHBPRST)	77
4.3.13 时钟配置寄存器 2 (RCC_CFG2)	77
4.3.14 时钟配置寄存器 3 (RCC_CFG3)	79
4.3.15 EMC 控制寄存器 (RCC_EMCTRL)	80

5 GPIO 和 AFIO 83

5.1 概述	83
5.2 功能描述	84
5.2.1 IO 模式配置	84
5.2.2 复位后状态	89
5.2.3 单独的位设置和位清除	89
5.2.4 外部中断/唤醒线	90
5.2.5 复用功能	90
5.2.6 外设的 IO 配置	101
5.2.7 GPIO 锁定机制	103
5.3 GPIO 寄存器	104
5.3.1 GPIO 寄存器总览	104
5.3.2 GPIO 端口模式寄存器 (GPIOx_PMODE)	105
5.3.3 GPIO 端口输出类型寄存器 (GPIOx_POTYPE)	106
5.3.4 GPIO 翻转率配置寄存器 (GPIOx_SR)	106
5.3.5 GPIO 端口上下拉寄存器 (GPIOx_PUPD)	107
5.3.6 GPIO 端口输入数据寄存器 (GPIOx_PID)	108
5.3.7 GPIO 端口输出数据寄存器 (GPIOx_POD)	108
5.3.8 GPIO 端口位设置/清除寄存器 (GPIOx_PBSC)	108
5.3.9 GPIO 端口锁定置寄存器 (GPIOx_PLOCK)	109
5.3.10 GPIO 复用功能低配置寄存器 (GPIOx_AFL)	110
5.3.11 GPIO 复用功能高配置寄存器 (GPIOx_AFH)	111
5.3.12 GPIO 端口位清除寄存器 (GPIOx_PBC)	111
5.3.13 GPIO 驱动能力配置寄存器 (GPIOx_DS)	112
5.4 AFIO 寄存器	112
5.4.1 AFIO 寄存器总览	112
5.4.2 AFIO 配置寄存器 (AFIO_CFG)	113
5.4.3 AFIO 外部中断配置寄存器 0 (AFIO_EXTI_CFG0)	114
5.4.4 AFIO 外部中断配置寄存器 1 (AFIO_EXTI_CFG1)	115
5.4.5 AFIO 外部中断配置寄存器 2 (AFIO_EXTI_CFG2)	116
5.4.6 AFIO 外部中断配置寄存器 3 (AFIO_EXTI_CFG3)	117

5.4.7 数字毛刺滤波器配置寄存器 1 (AFIO_DIGEFT_CFG1)	117
5.4.8 数字毛刺滤波器配置寄存器 2 (AFIO_DIGEFT_CFG2)	118
6 中断和事件	119
6.1 嵌套向量中断寄存器	119
6.1.1 SysTick 校准值寄存器	119
6.1.2 中断和异常向量	119
6.2 外部中断/事件控制器 (EXTI)	120
6.2.1 简介	120
6.2.2 主要特性	120
6.2.3 功能描述	121
6.2.4 EXTI 线路映射	123
6.3 EXTI 寄存器	124
6.3.1 EXTI 寄存器总览	124
6.3.2 EXTI 事件屏蔽寄存器 (EXTI_EMASK)	124
6.3.3 EXTI 中断屏蔽寄存器 (EXTI_IMASK)	125
6.3.4 EXTI 下降沿触发配置寄存器 (EXTI_FT_CFG)	125
6.3.5 EXTI 上升沿触发配置寄存器 (EXTI_RT_CFG)	126
6.3.6 EXTI 挂起寄存器 (EXTI_PEND)	126
6.3.7 EXTI 软件中断事件寄存器 (EXTI_SWIE)	127
6.3.8 EXTI 时间戳触发源选择寄存器 (EXTI_TS_SEL)	127
7 DMA 控制器	128
7.1 简介	128
7.2 主要特性	128
7.3 功能框图	129
7.4 功能描述	129
7.4.1 DMA 操作	129
7.4.2 通道优先级和仲裁器	130
7.4.3 DMA 通道和传输数量	130
7.4.4 可编程的数据位宽	130
7.4.5 外设/内存地址递增	132
7.4.6 通道配置流程	132
7.4.7 流量控制	132
7.4.8 循环模式	133
7.4.9 错误管理	133
7.4.10 中断	133
7.4.11 DMA 请求映射	134
7.5 DMA 寄存器	135
7.5.1 DMA 寄存器总览	135
7.5.2 DMA 中断状态寄存器 (DMA_INTSTS)	137
7.5.3 DMA 中断标志清除寄存器 (DMA_INTCLR)	137
7.5.4 DMA 通道 x 配置寄存器 (DMA_CHCFGx)	138
7.5.5 DMA 通道 x 传输数量寄存器 (DMA_TXNUMx)	140
7.5.6 DMA 通道 x 外设基地址寄存器 (DMA_PADDRx)	140
7.5.7 DMA 通道 x 存储器基地址寄存器 (DMA_MADDRx)	141
7.5.8 DMA 通道 x 请求选择寄存器 (DMA_CHSELx)	141
8 CRC 计算单元	143
8.1 简介	143
8.2 主要特性	143

8.3 CRC 功能描述	143
8.4 CRC 软件计算方式	144
8.5 CRC 寄存器	145
8.5.1 CRC 寄存器总览	145
8.5.2 CRC16 控制寄存器 (CRC_CRC16CTRL)	145
8.5.3 CRC16 待校验寄存器 (CRC_CRC16DAT)	146
8.5.4 CRC 循环冗余校验码寄存器 (CRC_CRC16D)	146
8.5.5 LRC 校验值寄存器 (CRC_LRC)	146

9 高级控制定时器 (TIM1) 148

9.1 TIM1 简介	148
9.2 TIM1 主要特性	148
9.3 TIM1 功能描述	149
9.3.1 时基单元	149
9.3.2 计数器模式	150
9.3.3 重复计数器	159
9.3.4 时钟选择	161
9.3.5 捕获/比较通道	165
9.3.6 输入捕获模式	167
9.3.7 PWM 输入模式	168
9.3.8 强制输出模式	169
9.3.9 输出比较模式	169
9.3.10 PWM 模式	170
9.3.11 单脉冲模式	172
9.3.12 在外部事件上清除 OCxREF 信号	174
9.3.13 互补输出和死区插入	174
9.3.14 刹车功能	176
9.3.15 调试模式	178
9.3.16 TIMx 定时器和外部触发的同步	178
9.3.17 定时器同步	181
9.3.18 产生六步 PWM 输出	181
9.3.19 编码器接口模式	182
9.3.20 与霍尔传感器的接口	184
9.4 TIM1 寄存器描述	186
9.4.1 寄存器总览	186
9.4.2 控制寄存器 1 (TIMx_CTRL1)	188
9.4.3 控制寄存器 2 (TIMx_CTRL2)	190
9.4.4 状态寄存器 (TIMx_STS)	192
9.4.5 事件产生寄存器 (TIMx_EVTGEN)	194
9.4.6 从模式控制寄存器 (TIMx_SMCTRL)	195
9.4.7 DMA/中断使能寄存器 (TIMx_DINTEN)	197
9.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1)	199
9.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2)	202
9.4.10 捕获/比较模式寄存器 3 (TIMx_CCMOD3)	203
9.4.11 捕获/比较使能寄存器 (TIMx_CCEN)	204
9.4.12 捕获/比较寄存器 1 (TIMx_CCDAT1)	207
9.4.13 捕获/比较寄存器 2 (TIMx_CCDAT2)	208
9.4.14 捕获/比较寄存器 3 (TIMx_CCDAT3)	209
9.4.15 捕获/比较寄存器 4 (TIMx_CCDAT4)	210
9.4.16 捕获/比较寄存器 5 (TIMx_CCDAT5)	210
9.4.17 捕获/比较寄存器 6 (TIMx_CCDAT6)	211

9.4.18 预分频器 (TIMx_PSC)	211
9.4.19 自动重载寄存器 (TIMx_AR)	212
9.4.20 计数器 (TIMx_CNT)	212
9.4.21 重复计数寄存器 (TIMx_REPCNT)	213
9.4.22 刹车和死区寄存器 (TIMx_BKDT)	213
9.4.23 捕获/比较寄存器 7 (TIMx_CCDA7)	215
9.4.24 捕获/比较寄存器 8 (TIMx_CCDA8)	215
9.4.25 捕获/比较寄存器 9 (TIMx_CCDA9)	216
9.4.26 刹车滤波寄存器 (TIMx_BKFR)	216
9.4.27 DMA 控制寄存器 (TIMx_DCTRL)	217
9.4.28 连续模式的 DMA 地址 (TIMx_DADDR)	218
10 通用定时器 (TIM2/TIM3/TIM4/TIM5)	220
10.1 TIM2/TIM3/TIM4/TIM5 简介	220
10.2 TIM2/TIM3/TIM4/TIM5 主要特性	220
10.3 TIM3 功能描述	221
10.3.1 时基单元	221
10.3.2 计数器模式	222
10.3.3 时钟选择	227
10.3.4 捕获/比较通道	231
10.3.5 输入捕获模式	234
10.3.6 PWM 输入模式	235
10.3.7 强制输出模式	236
10.3.8 输出比较模式	236
10.3.9 PWM 模式	238
10.3.10 单脉冲模式	240
10.3.11 在外部事件上清除 OCxREF 信号	242
10.3.12 调试模式	242
10.3.13 TIMx 定时器和外部触发的同步	242
10.3.14 定时器同步	243
10.3.15 编码器接口模式	247
10.3.16 与霍尔传感器的接口	249
10.4 TIMx 寄存器描述 (x=2/3/4/5)	249
10.4.1 寄存器总览	249
10.4.2 控制寄存器 1 (TIMx_CTRL1)	252
10.4.3 控制寄存器 2 (TIMx_CTRL2)	254
10.4.4 状态寄存器 (TIMx_STS)	255
10.4.5 事件产生寄存器 (TIMx_EVTGEN)	256
10.4.6 从模式控制寄存器 (TIMx_SMCTRL)	257
10.4.7 DMA/中断使能寄存器 (TIMx_DINTEN)	259
10.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1)	261
10.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2)	264
10.4.10 捕获/比较使能寄存器 (TIMx_CCEN)	266
10.4.11 捕获/比较寄存器 1 (TIMx_CCDA1)	267
10.4.12 捕获/比较寄存器 2 (TIMx_CCDA2)	267
10.4.13 捕获/比较寄存器 3 (TIMx_CCDA3)	268
10.4.14 捕获/比较寄存器 4 (TIMx_CCDA4)	269
10.4.15 预分频器 (TIMx_PSC)	269
10.4.16 自动重载寄存器 (TIMx_AR)	270
10.4.17 计数器 (TIMx_CNT)	270
10.4.18 通道 1 滤波寄存器 (TIMx_C1FILT)	271

10.4.19 通道2滤波寄存器 (TIMx_C2FILT)	271
10.4.20 通道3滤波寄存器 (TIMx_C3FILT)	272
10.4.21 通道4滤波寄存器 (TIMx_C4FILT)	273
10.4.22 输入通道滤波输出寄存器 (TIMx_FILTO)	274
10.4.23 输入通道选择寄存器 (TIMx_INSEL)	275
10.4.24 DMA 控制寄存器 (TIMx_DCTRL)	277
10.4.25 连续模式的DMA 地址 (TIMx_DADDR)	278
11 基本定时器 (TIM6).....	279
11.1 基本定时器简介	279
11.2 基本定时器主要特性	279
11.3 基础定时器描述	279
11.3.1 时基单元.....	279
11.3.2 计数模式.....	280
11.3.3 时钟选择.....	283
11.3.4 调试模式.....	283
11.4 TIM6 寄存器描述	283
11.4.1 寄存器总览	283
11.4.2 控制寄存器 1 (TIMx_CTRL1)	284
11.4.3 控制寄存器 2 (TIMx_CTRL2)	285
11.4.4 状态寄存器 (TIMx_STS)	286
11.4.5 事件产生寄存器 (TIMx_EVTGEN)	286
11.4.6 DMA/中断使能寄存器 (TIMx_DINTEN)	287
11.4.7 预分频器 (TIMx_PSC).....	287
11.4.8 自动重装载寄存器 (TIMx_AR)	288
11.4.9 计数器 (TIMx_CNT).....	288
12 独立看门狗 (IWDG)	290
12.1 简介	290
12.2 主要特性	290
12.3 功能描述	291
12.3.1 寄存器访问保护	291
12.3.2 调试模式	291
12.3.3 IWDG 冻结	292
12.4 用户界面	292
12.4.1 操作流程	292
12.4.2 IWDG 配置流程	293
12.5 IWDG 寄存器	293
12.5.1 IWDG 寄存器总览	293
12.5.2 IWDG 密钥寄存器 (IWDG_KEY)	293
12.5.3 IWDG 预分频寄存器 (IWDG_PREDIV)	294
12.5.4 IWDG 重装载寄存器 (IWDG_RELV)	295
12.5.5 IWDG 状态寄存器 (IWDG_STS)	295
12.5.6 IWDG 冻结寄存器 (IWDG_FREEZE)	296
13 窗口看门狗 (WWDG)	297
13.1 简介	297
13.2 主要特征	297
13.3 功能描述	297
13.4 刷新看门狗和中断产生的时许	298
13.5 调试模式	299

13.6 用户界面	299
13.6.1 WWDG 配置流程	299
13.7 WWDG 寄存器	300
13.7.1 WWDG 寄存器总览	300
13.7.2 WWDG 控制寄存器 (WWDG_CTRL)	300
13.7.3 WWDG 配置寄存器 (WWDG_CFG)	301
13.7.4 WWDG 状态寄存器 (WWDG_STS)	301
14 模拟数字转换 (ADC)	302
14.1 简述	302
14.2 ADC 主要特征	302
14.3 ADC 功能描述	302
14.3.1 ADC 时钟	304
14.3.2 ADC 开关控制	304
14.3.3 通道选择	304
14.3.4 内部通道	304
14.3.5 单次转换模式	304
14.3.6 连续转换模式	305
14.3.7 时序图	305
14.3.8 模拟看门狗	305
14.3.9 扫描模式	306
14.4 数据对齐	306
14.5 可编程的通道采样时间	306
14.6 外部触发转换	306
14.7 DMA 请求	307
14.8 温度传感器	307
14.8.1 测量温度值	308
14.9 ADC 中断	309
14.10 ADC 寄存器	310
14.10.1 ADC 寄存器总览	310
14.10.2 ADC 状态寄存器(ADC_STS)	310
14.10.3 ADC 控制寄存器 1(ADC_CTRL1)	311
14.10.4 ADC 控制寄存器 2(ADC_CTRL2)	312
14.10.5 ADC 控制寄存器 3 (ADC_CTRL3)	314
14.10.6 ADC 采样时间寄存器(ADC_SAMPT)	315
14.10.7 ADC 看门狗高阈值寄存器(ADC_WDGHIGH)	316
14.10.8 ADC 看门狗低阈值寄存器(ADC_WDGLOW)	316
14.10.9 ADC 规则数据寄存器 x(ADC_DATx)(x = 0..4)	317
15 数字模拟转换 (DAC)	318
15.1 DAC 介绍	318
15.2 DAC 主要特性	318
15.3 DAC 功能描述与操作说明	320
15.3.1 DAC 开启	320
15.3.2 DAC 输出缓存	320
15.3.3 DAC 数据格式	320
15.3.4 DAC 触发	320
15.3.5 DAC 转换	321
15.3.6 DAC 输出电压	322
15.3.7 DMA 请求	322
15.3.8 噪声产生	322

15.3.9 三角波产生	323
15.4 DAC 寄存器	325
15.4.1 DAC 寄存器总览	325
15.4.2 DAC 控制寄存器 (DAC_CTRL)	325
15.4.3 DAC 软件触发寄存器 (DAC_SOTTR)	327
15.4.4 DAC 数据输出寄存器 (DAC_DATO)	327
15.4.5 DAC 的 8 位右对齐数据保持寄存器 (DAC_DR8CH)	328
15.4.6 DAC 的 12 位左对齐数据保持寄存器 (DAC_DL12CH)	328
15.4.7 DAC 的 12 位右对齐数据保持寄存器 (DAC_DR12CH)	329
16 比较器 (COMP)	330
16.1 COMP 系统连接框图	330
16.2 COMP 特性	331
16.3 COMP 配置流程	332
16.4 COMP 工作模式	332
16.4.1 窗口比较器	332
16.4.2 独立比较器	332
16.5 比较器互联关系	333
16.6 比较器输出	334
16.7 中断	334
16.8 COMP 寄存器	334
16.8.1 COMP 寄存器总览	334
16.8.2 COMP1 控制寄存器 (COMP1_CTRL)	336
16.8.3 COMP1 滤波控制寄存器 (COMP1_FILC)	338
16.8.4 COMP1 滤波时钟寄存器 (COMP1_FILP)	338
16.8.5 COMP2 控制寄存器 (COMP2_CTRL)	339
16.8.6 COMP2 滤波控制寄存器 (COMP2_FILC)	340
16.8.7 COMP2 滤波时钟寄存器 (COMP2_FILP)	341
16.8.8 COMP3 控制寄存器 (COMP3_CTRL)	341
16.8.9 COMP3 滤波控制寄存器 (COMP3_FILC)	343
16.8.10 COMP3 滤波时钟寄存器 (COMP3_FILP)	343
16.8.11 COMP4 控制寄存器 (COMP4_CTRL)	344
16.8.12 COMP4 滤波控制寄存器 (COMP4_FILC)	345
16.8.13 COMP4 滤波时钟寄存器 (COMP4_FILP)	346
16.8.14 COMP 输出选择寄存器 (COMP_OSEL)	346
16.8.15 COMP 锁寄存器 (COMP_LOCK)	347
16.8.16 COMP 中断使能寄存器 (COMP_INTEN)	348
16.8.17 COMP 中断状态寄存器 (COMP_INTSTS)	348
16.8.18 COMP 参考输入比较电压寄存器 (COMP_INVREF)	349
16.8.19 COMP 输出到定时器使能寄存器 (COMP_OTIMEN)	349
17 液晶显示控制器 (LCD)	351
17.1 简介	351
17.2 主要特性	351
17.3 功能框图	352
17.4 功能描述	353
17.4.1 频率发生器	353
17.4.2 公共端驱动器	354
17.4.3 区段驱动器	355
17.4.4 电压发生器和对比度控制	360
17.4.5 二级缓冲显示	362

17.4.6 COM 和 SEG 多路复用.....	362
17.5 工作流程.....	364
17.6 中断请求.....	364
17.7 LCD 控制器寄存器.....	364
17.7.1 LCD 控制器寄存器总览.....	365
17.7.2 LCD 控制寄存器 (LCD_CTRL).....	366
17.7.3 LCD 帧控制寄存器 (LCD_FCTRL).....	367
17.7.4 LCD 状态寄存器 (LCD_STS).....	370
17.7.5 LCD 清除寄存器 (LCD_CLR).....	371
17.7.6 LCD 显示内存寄存器 (LCD_RAM1_COMx x = 0...7).....	372
17.7.7 LCD 显示内存寄存器 (LCD_RAM2_COMx x = 0...7).....	372
18 LED 驱动器.....	373
18.1 LED 介绍.....	373
18.2 LED 主要特性.....	373
18.3 LED 功能描述与操作说明.....	373
18.3.1 控制接口.....	373
18.3.2 时钟控制.....	374
18.3.3 通信协议格式描述.....	374
18.3.4 LED 工作时序.....	383
18.3.5 控制 LED 流程.....	384
18.3.6 回读 LED 寄存器流程描述.....	386
18.3.7 复用按键扫描.....	387
19 内部集成电路总线(I²C).....	388
19.1 简介.....	388
19.2 主要特性.....	388
19.3 功能描述.....	388
19.3.1 SDA/SCL 控制.....	388
19.3.2 软件通讯流程.....	389
19.3.3 错误条件.....	398
19.3.4 DMA 应用.....	399
19.3.5 包错误校验 (PEC).....	400
19.3.6 超时错误.....	401
19.3.7 SMBus.....	401
19.3.8 噪声滤波.....	403
19.4 调试模式.....	403
19.5 中断请求.....	403
19.6 I ² C 寄存器描述.....	404
19.6.1 I ² C 寄存器总览.....	404
19.6.2 I ² C 控制寄存器 1 (I2C_CTRL1).....	406
19.6.3 I ² C 控制寄存器 2 (I2C_CTRL2).....	408
19.6.4 I ² C 自身地址寄存器 1 (I2C_OADDR1).....	410
19.6.5 I ² C 自身地址寄存器 2 (I2C_OADDR2).....	410
19.6.6 I ² C 数据寄存器 (I2C_DAT).....	411
19.6.7 I ² C 状态寄存器 1 (I2C_STS1).....	411
19.6.8 I ² C 状态寄存器 2 (I2C_STS2).....	414
19.6.9 I ² C 时钟控制寄存器 (I2C_CLKCTRL).....	416
19.6.10 I ² C 上升时间寄存器 (I2C_TMRIS).....	417
19.6.11 I ² C 主机接收字节寄存器 (I2C_BYTENUM).....	417
19.6.12 I ² C 滤波控制寄存器 (I2C_GFLTRCTRL).....	417

20 通用异步收发器(UART)	419
20.1 简介	419
20.2 主要特性	419
20.3 功能框图	420
20.4 功能描述	420
20.4.1 UART 帧格式	421
20.4.2 发送器	421
20.4.3 接收器	424
20.4.4 分数波特率计算	426
20.4.5 UART 接收器容忍时钟的变化	429
20.4.6 校验控制	429
20.4.7 DMA 通信	430
20.4.8 多处理器通信	431
20.4.9 单线半双工模式	433
20.4.10 串行 IrDA 红外编解码模式	433
20.4.11 LIN 模式	434
20.5 中断请求	437
20.6 模式配置	438
20.7 UART 寄存器	439
20.7.1 UART 寄存器总览	439
20.7.2 UART 控制寄存器 1(UART_CTRL1)	440
20.7.3 UART 控制寄存器 2(UART_CTRL2)	441
20.7.4 UART 控制寄存器 3(UART_CTRL3)	442
20.7.5 UART 状态寄存器 (UART_STS)	443
20.7.6 UART 数据寄存器(UART_DAT)	445
20.7.7 UART 波特率配置寄存器 (UART_BRCF)	446
20.7.8 UART 保护时间和预分频寄存器(UART_GTP)	446
21 控制器局域网(CAN)	448
21.1 CAN 简介	448
21.2 CAN 的主要特性	448
21.3 CAN 整体介绍	448
21.3.1 CAN 模块	449
21.3.2 CAN 工作模式	449
21.3.3 发送邮箱	451
21.3.4 接收过滤器	451
21.3.5 接收 FIFO	451
21.3.6 CAN 测试模式	452
21.3.7 CAN 调试模式	454
21.4 CAN 功能描述	454
21.4.1 发送处理	454
21.4.2 时间触发通信模式	455
21.4.3 非自动重传模式	455
21.4.4 接收管理	456
21.4.5 标识符过滤	457
21.4.6 消息存储	460
21.4.7 位时序	461
21.5 CAN 中断	464
21.5.1 错误管理	465
21.5.2 总线关闭恢复	465
21.6 CAN 配置流程	465

21.7 CAN 寄存器	466
21.7.1 寄存器描述	467
21.7.2 CAN 寄存器总览	467
21.7.3 CAN 控制和状态寄存器	470
21.7.4 CAN 邮箱寄存器	480
21.7.5 CAN 过滤器寄存器	486
22 串行外设接口 (SPI)	490
22.1 SPI 简介	490
22.2 SPI 主要特性	490
22.3 SPI 功能描述	491
22.3.1 通用描述	491
22.3.2 SPI 工作模式	494
22.3.3 状态标志	500
22.3.4 关闭 SPI	501
22.3.5 使用 DMA 进行 SPI 通讯	501
22.3.6 CRC 计算	502
22.3.7 错误标志位	503
22.3.8 SPI 中断	504
22.4 SPI 寄存器描述	505
22.4.1 SPI 寄存器总览	505
22.4.2 SPI 控制寄存器 1 (SPI_CTRL1)	506
22.4.3 SPI 控制寄存器 2 (SPI_CTRL2)	507
22.4.4 SPI 状态寄存器 (SPI_STS)	509
22.4.5 SPI 数据寄存器 (SPI_DAT)	510
22.4.6 SPI Tx CRC 寄存器 (SPI_CRCTDAT)	510
22.4.7 SPI Rx CRC 寄存器 (SPI_CRCRDAT)	511
22.4.8 SPI CRC 多项式寄存器 (SPI_CRCPOLY)	511
22.4.9 SPI 时钟采样延迟寄存器 (SPI_CTRL3)	512
23 实时时钟(RTC)	514
23.1 简介	514
23.1.1 主要特性	515
23.2 RTC 功能描述	516
23.2.1 RTC 框图	516
23.2.2 RTC 控制的 GPIO	517
23.2.3 RTC 寄存器写保护	517
23.2.4 RTC 时钟和预分频	517
23.2.5 RTC 日历	518
23.2.6 日历初始化和配置	518
23.2.7 日历读取	518
23.2.8 校准时钟输出	519
23.2.9 可编程闹钟	519
23.2.10 闹钟配置	520
23.2.11 闹钟输出	520
23.2.12 周期性自动唤醒	520
23.2.13 唤醒定时器配置	520
23.2.14 时间戳功能	521
23.2.15 入侵检测	521
23.2.16 夏令时功能配置	522
23.2.17 RTC 亚秒寄存器位移操作	522

23.2.18 RTC 数字时钟精密校准.....	522
23.2.19 RTC 低功耗模式.....	523
23.3 RTC 寄存器.....	524
23.3.1 RTC 寄存器总览.....	524
23.3.2 RTC 初始状态寄存器 (RTC_INITSTS).....	525
23.3.3 RTC 控制寄存器(RTC_CTRL)	527
23.3.4 RTC 日历时间寄存器 (RTC_TSH)	529
23.3.5 RTC 日历日期寄存器 (RTC_DATE)	530
23.3.6 RTC 写保护寄存器(RTC_WRP).....	530
23.3.7 RTC 平移控制寄存器(RTC_SCTRL)	531
23.3.8 RTC 亚秒寄存器(RTC_SUBS)	532
23.3.9 RTC 时间戳时间寄存器 (RTC_TST).....	532
23.3.10 RTC 闹钟 A 寄存器(RTC_ALARM_A)	533
23.3.11 RTC 预分频寄存器(RTC_PRE)	534
23.3.12 RTC 闹钟 B 寄存器 (RTC_ALARM_B).....	534
23.3.13 RTC 唤醒定时器寄存器(RTC_WKUP_T)	535
23.3.14 RTC 入侵配置寄存器 (RTC_TMPCFG)	536
23.3.15 RTC 闹钟 A 亚秒寄存器(RTC_ALRMASS)	538
23.3.16 RTC 选项寄存器 (RTC_OPT).....	539
23.3.17 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSS).....	540
23.3.18 RTC 校准寄存器(RTC_CALIB).....	540
23.3.19 RTC 时间戳亚秒寄存器(RTC_TSSS)	541
23.3.20 RTC 时间戳日期寄存器 (RTC_TSD).....	542
24 蜂鸣器 (Beeper)	543
24.1 简介	543
24.2 功能描述	543
24.3 Beeper 寄存器.....	543
24.3.1 Beeper 寄存器总览.....	543
24.3.2 Beeper 控制寄存器 (BEEPER_CTRL)	543
25 调试支持 (DBG)	545
25.1 简介	545
25.2 SWD 功能	546
25.2.1 引脚分配	546
26 唯一设备序列号 (UID)	547
26.1 简介	547
26.2 UID 寄存器	547
26.3 UCID 寄存器	547
26.4 DBGMCU_ID 寄存器	547
27 版本历史	548
28 声明	549

表目录

表 2-1 外设寄存器地址列表	4
表 2-2 启动模式列表	7
表 2-3 存储总线地址列表	8
表 2-4 选项字节列表	12
表 2-5 读保护配置列表	14
表 2-6 存储区读写擦 ⁽¹⁾ 权限控制表	15
表 2-7 FLASH 寄存器总览	19
表 3-1 电源模式	32
表 3-2 外设运行状态	33
表 3-3 PWR 寄存器地址映像和复位值	36
表 4-1 RCC 寄存器总览	57
表 5-1 IO 模式和配置关系	84
表 5-2 IO 不同配置的输入输出特性	85
表 5-3 SWD 复用功能 I/O 重映射	90
表 5-4 TIM1 复用功能 I/O 重映射	90
表 5-5 TIM2 复用功能 I/O 重映射	91
表 5-6 TIM3 复用功能 I/O 重映射	92
表 5-7 TIM4 复用功能 I/O 重映射	92
表 5-8 TIM5 复用功能 I/O 重映射	93
表 5-9 UART1 复用功能 I/O 重映射	93
表 5-10 UART2 复用功能 I/O 重映射	93
表 5-11 UART3 复用功能 I/O 重映射	94
表 5-12 UART4 复用功能 I/O 重映射	94
表 5-13 UART5 复用功能 I/O 重映射	94
表 5-14 I2C1 复用功能 I/O 重映射	95
表 5-15 I2C2 复用功能 I/O 重映射	95
表 5-16 SPI1 管脚重映射	97
表 5-17 SPI2 管脚重映射	97
表 5-18 SPI3 管脚重映射	98
表 5-19 CAN 复用功能重映射	98
表 5-20 COMPx 复用功能 I/O 重映射	99
表 5-21 BEEPER 复用功能 I/O 重映射	99
表 5-22 RTC 复用功能 I/O 重映射	99

表 5-23 LCD 复用功能 I/O 重映射.....	99
表 5-24 EVENTOUT 复用功能 I/O 重映射.....	101
表 5-25 RCC 复用功能 I/O 重映射.....	101
表 5-26 ADC	101
表 5-27 LED	102
表 5-28 LCD.....	102
表 5-29 TIM1.....	102
表 5-30 TIM2、3、4、5.....	102
表 5-31 UART	102
表 5-32 I2C.....	102
表 5-33 SPI.....	102
表 5-34 CAN	103
表 5-35 COMP.....	103
表 5-36 RTC	103
表 5-37 BEEPER.....	103
表 5-38 其他.....	103
表 5-39 GPIO 寄存器总览.....	104
表 5-40 AFIO 寄存器总览.....	113
表 6-1 向量表.....	119
表 6-2 EXTI 寄存器总览.....	124
表 7-1 可编程的数据宽度和大小端操作(当 PINC = MINC = 1).....	130
表 7-2 流量控制表.....	133
表 7-3 DMA 中断请求.....	134
表 7-4 DMA 请求映射.....	135
表 7-5 DMA 寄存器总览.....	135
表 8-1 CRC 寄存器总览.....	145
表 9-1 计数方向与编码器信号的关系	183
表 9-2 TIM1 寄存器总览.....	186
表 9-3 TIMx 内部触发连接.....	197
表 9-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位	206
表 10-1 计数方向与编码器信号的关系	248
表 10-2 TIM1 寄存器总览.....	249
表 10-3 TIMx 内部触发连接.....	259
表 10-4 标准 OCx 的输出控制位.....	267

表 11-1 寄存器总览	283
表 12-1 IWDG 计数最大和最小复位时间	292
表 12-2 IWDG 寄存器总览	293
表 13-1 WWDG 的最大和最小计数时间	299
表 13-2 WWDG 寄存器总览	300
表 14-1 ADC 引脚	303
表 14-2 模拟看门狗通道选择	306
表 14-3 数据右对齐	306
表 14-4 数据左对齐	306
表 14-5 ADC 用于规则通道的外部触发	307
表 14-6 ADC 中断	309
表 14-7 ADC 寄存器总览	310
表 15-1 DAC 引脚	319
表 15-2 DAC 外部触发	321
表 15-3 DAC 寄存器总览	325
表 16-1 COMP 寄存器总览	334
表 17-1 帧频计算示例	353
表 17-2 闪烁频率配置示例	360
表 17-3 COM 与 SEG 管脚映射表	362
表 17-4 LCD 控制寄存器总览	365
表 19-1 SMBus 与 I ² C 的比较	402
表 19-2 I ² C 中断请求	403
表 19-3 I ² C 寄存器总览	404
表 20-1 停止位配置	422
表 20-2 噪声检测的数据采样	426
表 20-3 设置波特率时的误差计算	427
表 20-4 当 DIV_Decimal =0 时, UART 接收器的容忍度	429
表 20-5 当 DIV_Decimal !=0 时, UART 接收器的容忍度	429
表 20-6 帧格式	429
表 20-7 UART 中断请求	437
表 20-8 UART 模式设置 ⁽¹⁾	438
表 20-9 UART 寄存器总览	439
表 21-1 过滤器编号示例	459
表 21-2 发送邮箱寄存器列表	460

表 21-3 接收邮箱寄存器列表 461

表 21-4 CAN 寄存器总览..... 467

表 22-1 SPI 中断请求 504

表 22-2 SPI 寄存器总览 505

表 23-1 RTC 功能支持 515

表 23-2 RTC 寄存器总览 524

表 24-1 Beeper 寄存器总览..... 543

表 25-1 调试端口引脚..... 546

表 26-1 DBGMCU_ID 位描述 547

图目录

图 2-1 总线架构图	3
图 2-2 总线地址映射图	4
图 3-1 电源框图	30
图 3-2 上电复位和掉电复位的波形图	31
图 3-3 PVD 阈值图	31
图 3-4 LVR 阈值图	32
图 4-1 复位电路	50
图 4-2 时钟树	52
图 4-3 HSE 时钟源	53
图 4-4 PLL 时钟树	54
图 5-1 I/O 端口的基本结构（不支持 Fail-safe）	84
图 5-2 输入浮空/上拉/下拉模式（不支持 Fail-safe）	86
图 5-3 输出模式（不支持 Fail-safe）	87
图 5-4 复用功能模式（不支持 Fail-safe）	88
图 5-5 高阻抗的模拟功能模式（不支持 Fail-safe）	89
图 6-1 外部中断/事件控制器框图	121
图 6-2 外部中断通用 I/O 映射	123
图 7-1 DMA 框图	129
图 8-1 CRC 计算单元框图	143
图 9-1 TIM1 框图	149
图 9-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图	151
图 9-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图	152
图 9-5 内部时钟分频因子 = 2/N 时，向下计数时序图	153
图 9-6 内部时钟分频因子 = 2/N，中央对齐时序图	154
图 9-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)	155
图 9-8 非对称模式对应的输出波形	156
图 9-9 CC DAT _x (x=4,7,8,9),当 DIR = 0 时触发 ADC	157
图 9-10 CC DAT _x (x=4,7,8,9), 当 DIR = 1 时触发 ADC	158
图 9-11 CC DAT _x (x=4,7,8,9), 当 DIR = 1 或 DIR = 0 时触发 ADC	159
图 9-12 向下计数模式下的重复计数时序图	160
图 9-13 向上计数模式下的重复计数时序图	160
图 9-14 中央对齐模式下的重复计数时序图	161
图 9-15 正常模式下的控制电路，内部时钟除以 1	162

图 9-16 TI2 外部时钟连接示例	162
图 9-17 外部时钟模式 1 的控制电路	163
图 9-18 外部触发输入框图	164
图 9-19 外部时钟模式 2 的控制电路	164
图 9-20 捕获/比较通道（例如：通道 1 输入级）	165
图 9-21 捕获/比较通道 1 主电路	166
图 9-24 输出比较模式，开启 OC1	170
图 9-26 边沿对齐 PWM 波形 (AR=8)	172
图 9-27 单脉冲模式示例	173
图 9-28 清除 TIMx 的 OCxREF	174
图 9-29 带死区插入的互补输出	175
图 9-30 响应刹车的输出行为	177
图 9-31 滑动滤波	178
图 9-32 复位模式下的控制电路	179
图 9-33 触发器模式下的控制电路	179
图 9-34 门控模式下的控制电路	180
图 9-35 外部时钟模式 2+触发模式下的控制电路	181
图 9-36 产生六步 PWM，使用 COM 的例子 (OSSR=1)	182
图 9-37 编码器模式下的计数器操作实例	183
图 9-38 IC1FP1 反相的编码器接口模式实例	184
图 9-39 霍尔传感器接口的实例	185
图 10-1 TIMx 框图	221
图 10-2 当预分频的参数从 1 到 4，计数器的时序图	222
图 10-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图	223
图 10-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图	224
图 10-5 内部时钟分频因子 = 2/N 时，向下计数时序图	225
图 10-6 内部时钟分频因子 = 2/N，中央对齐时序图	226
图 10-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)	227
图 10-8 正常模式下的控制电路，内部时钟除以 1	228
图 10-9 TI2 外部时钟连接示例	229
图 10-10 外部时钟模式 1 的控制电路	230
图 10-11 外部触发输入框图	230
图 10-12 外部时钟模式 2 的控制电路	231
图 10-13 捕获/比较通道（例如：通道 1 输入级）	232

图 10-14 捕获/比较通道 1 主电路.....	233
图 10-15 通道 x 的输出部分 (x = 1/2/3/4, 以通道 4 为例子)	234
图 10-16 滑动滤波.....	235
图 10-17 PWM 输入模式时序	236
图 10-18 输出比较模式, 开启 OC1.....	237
图 10-20 边沿对齐 PWM 波形 (AR=8).....	240
图 10-21 单脉冲模式示例.....	241
图 10-22 清除 TIMx 的 OCxREF.....	242
图 10-23 主/从定时器的例子.....	243
图 10-24 定时器 2 由定时器 1 的 OC1REF 门控.....	244
图 10-25 定时器 2 由定时器 1 的使能门控.....	245
图 10-26 使用定时器 1 的更新触发定时器 2.....	246
图 10-27 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2.....	247
图 10-28 编码器模式下的计数器操作实例.....	248
图 10-29 IC1FP1 反相的编码器接口模式实例.....	249
图 11-1 TIMx 的框图 (x = 6)	279
图 11-2 预分频器分频从 1 到 4 的计数器时序图	280
图 11-3 向上计数时序图, 内部时钟分频因子 = 2/N.....	281
图 11-4 ARPEN=0/1 时向上计数、更新事件的时序图.....	282
图 11-5 正常模式下的控制电路, 内部时钟分频系数为 1	283
图 12-1 独立看门狗模块的功能框图.....	291
图 13-1 窗口看门狗功能框图.....	297
图 13-2 WWDG 的刷新窗口和中断时序.....	298
图 14-1 ADC 框图	303
图 14-2 时序图.....	305
图 14-3 温度传感器通道框图.....	308
图 15-1 DAC 通道的框图.....	319
图 15-2 单 DAC 通道模式的数据寄存器.....	320
图 15-3 触发禁能时转换时序图.....	322
图 15-4 DAC LFSR 算法.....	323
图 15-5 带 LFSR 波形生成的 DAC 转换 (使能软件触发)	323
图 15-6 DAC 三角波生成.....	324
图 15-7 带三角生成的 DAC 转换 (使能软件触发)	324
图 16-1 比较器 1 和比较器 2 系统连接图.....	330

图 16-2 比较器 3 和比较器 4 系统连接图	331
图 17-1 LCD 控制器框图	352
图 17-2 奇偶帧示例图（1/4 占空比，1/3 偏置）	354
图 17-3 静态占空比示例	355
图 17-4 1/2 占空比，1/2 偏置	356
图 17-5 1/3 占空比，1/3 偏置	357
图 17-6 1/4 占空比，1/3 偏置	358
图 17-7 1/8 占空比，1/4 偏置	359
图 17-8 LCD 驱动电压控制	361
图 17-9 死区时间	362
图 18-1 GCLK 时序图	374
图 18-2 GCLK 与 COM 对应时序图	383
图 18-3 GCLK 与 COM-SEG 控制时序图	384
图 19-1 I ² C 功能框图	390
图 19-2 I ² C 总线协议	390
图 19-3 从发送器传送序列	392
图 19-4 从接收器传送序列	394
图 19-5 主发送器传送序列	396
图 19-6 主接收器传送序列	398
图 20-1 UART 框图	420
图 20-2 字长=8 设置	421
图 20-3 字长=9 设置	421
图 20-4 停止位配置	422
图 20-5 发送时 TXC/TXDE 的变化情况	424
图 20-6 起始位检测	425
图 20-7 DMA 发送	430
图 20-8 DMA 接收	431
图 20-9 静默模式下的空闲总线检测	432
图 20-10 静默模式下的地址标识检测	433
图 20-11 IrDA SIR ENDEC-框图	434
图 20-12 IrDA 数据调制(3/16)-正常模式	434
图 20-13 LIN 模式下的断开检测（11 位断开帧长度-设置了 LINBDL 位）	436
图 20-14 LIN 模式下的断开检测与帧错误的检测	437
图 21-1 CAN 网络拓扑	449

图 21-2 CAN 工作模式.....	451
图 21-3 单 CAN 框图	452
图 21-4 回环模式.....	453
图 21-5 静默模式.....	453
图 21-6 回环静默模式.....	454
图 21-7 发送邮箱状态.....	456
图 21-8 接收邮箱状态.....	456
图 21-9 过滤器位宽设置-寄存器组织.....	458
图 21-10 过滤机制示例.....	460
图 21-11 位时序	462
图 21-12 各类帧格式.....	463
图 21-13 事件标志和中断产生.....	464
图 21-14 CAN 错误状态框图.....	465
图 22-1 SPI 框图	491
图 22-2 硬件/软件的从选择管理	492
图 22-3 单主和单从应用	492
图 22-4 数据时钟时序图.....	494
图 22-5 主机全双工模式下连续传输时，SPI_STS.TE/RNE/BUSY 的变化示意图	495
图 22-6 主机单向只发送模式下连续传输时，SPI_STS.TE/BUSY 变化示意图	496
图 22-7 只接收模式（BIDIRMODE=0 并且 RONLY=1）下连续传输时，RNE 变化示意图.....	497
图 22-8 从机全双工模式下连续传输时，SPI_STS.TE/RNE/BUSY 的变化示意图	498
图 22-9 从机单向只发送模式下连续传输时，SPI_STS.TE/BUSY 变化示意图	498
图 22-10 BIDIRMODE = 0，RONLY = 0 非连续传输发送时，SPI_STS.TE/BUSY 变化示意图.....	500
图 22-11 使用 DMA 发送.....	502
图 22-12 使用 DMA 接收.....	502
图 23-1 RTC 功能框图	516
图 25-1 N32G05X 级别和 Cortex®-M0 级别的调试框图.....	545

1 文中的缩写

1.1 寄存器描述表中使用的缩写列表

在对寄存器的描述中使用了下列缩写：

read/write(rw)	软件能读写此位。
read-only(r)	软件只能读此位。
write-only(w)	软件只能写此位，读此位将返回复位值。
read/clear(rc_w1)	软件可以读此位，也可以通过写‘1’清除此位，写‘0’对此位无影响。
read/clear(rc_w0)	软件可以读此位，也可以通过写‘0’清除此位，写‘1’对此位无影响。
read/clear by read(rc_r)	软件可以读此位，读此位将自动地清除它为‘0’，写‘0’对此位无影响。
read/set(rs)	软件可以读也可以设置此位，写‘0’对此位无影响。
read-only write trigger(rt_w)	软件可以读此位，写‘0’或‘1’触发一个事件但对此位数值没有影响。
toggle(t)	软件只能通过写‘1’来翻转此位，写‘0’对此位无影响。
Reserved(Res.)	保留位，必须保持默认值不变。

1.2 可用的外设

有关 N32G05X 系列全部型号，某外设存在与否及其数目，请查阅相应型号的数据手册。

2 存储器和总线架构

2.1 系统架构

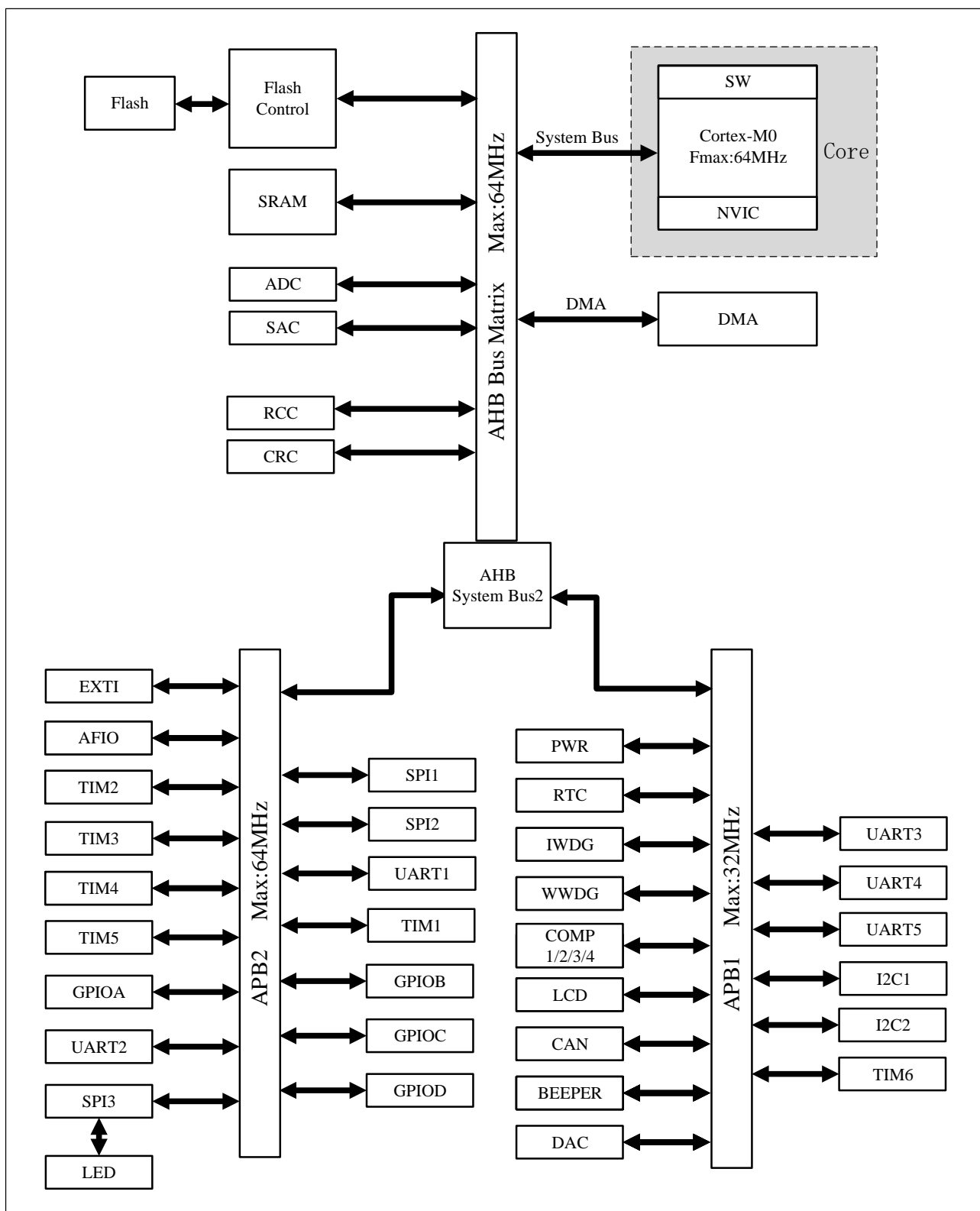
2.1.1 总线架构

主系统由以下部分构成：

- 两个主驱动单元：
 - ◆ Cortex®-M0 内核系统总线
 - ◆ 通用 DMA
- 六个被动单元
 - ◆ 内部 SRAM
 - ◆ 内部闪存存储器
 - ◆ ADC
 - ◆ AHB 到 AHB 的桥，它连接一些 AHB 设备
 - ◆ AHB 到 APB 的桥(AHB2APB_x, x=1,2)，它连接所有的 APB 设备

这些都是通过一个多级的 AHB 总线构架相互连接的，如图 2-1 所示：

图 2-1 总线架构图



- CPU 系统总线：连接 Cortex®-M0 内核的 Sbus 总线到总线矩阵，用来指令预取，数据加载（常量加载和调试访问）及 AHB/APB 外设访问。
- DMA 总线：DMA 的 AHB 主控接口连接到总线矩阵，总线矩阵协调着内核和 DMA 到 SRAM、闪存

和外设的访问。

- 总线矩阵协调内核系统总线和 DMA 主控总线之间的访问仲裁，仲裁利用轮算法。总线矩阵包含 2 个驱动部件(CPU 的系统总线、DMA 总线)和 7 个从部件(闪存存储器接口、SRAM、ADC、RCC、CRC、SAC 和 AHB 系统总线 2)。系统总线 2 连接 2 个 AHB2APB 桥。
- 系统包含 2 个 AHB2APB 桥，即 AHB2APB1 和 AHB2APB2。其中 APB1 包含 15 个 APB 外设，PCLK1 的最高速度为 32MHz；APB2 包含 16 个 APB 外设，PCLK2 最高速度等于 64MHz。

2.1.2 总线地址映射

总线地址映射包括所有 AHB 和 APB 外设: AHB 外设、APB1 外设、APB2 外设、Flash、SRAM、SystemMemory 等。具体映射如下

图 2-2 总线地址映射图

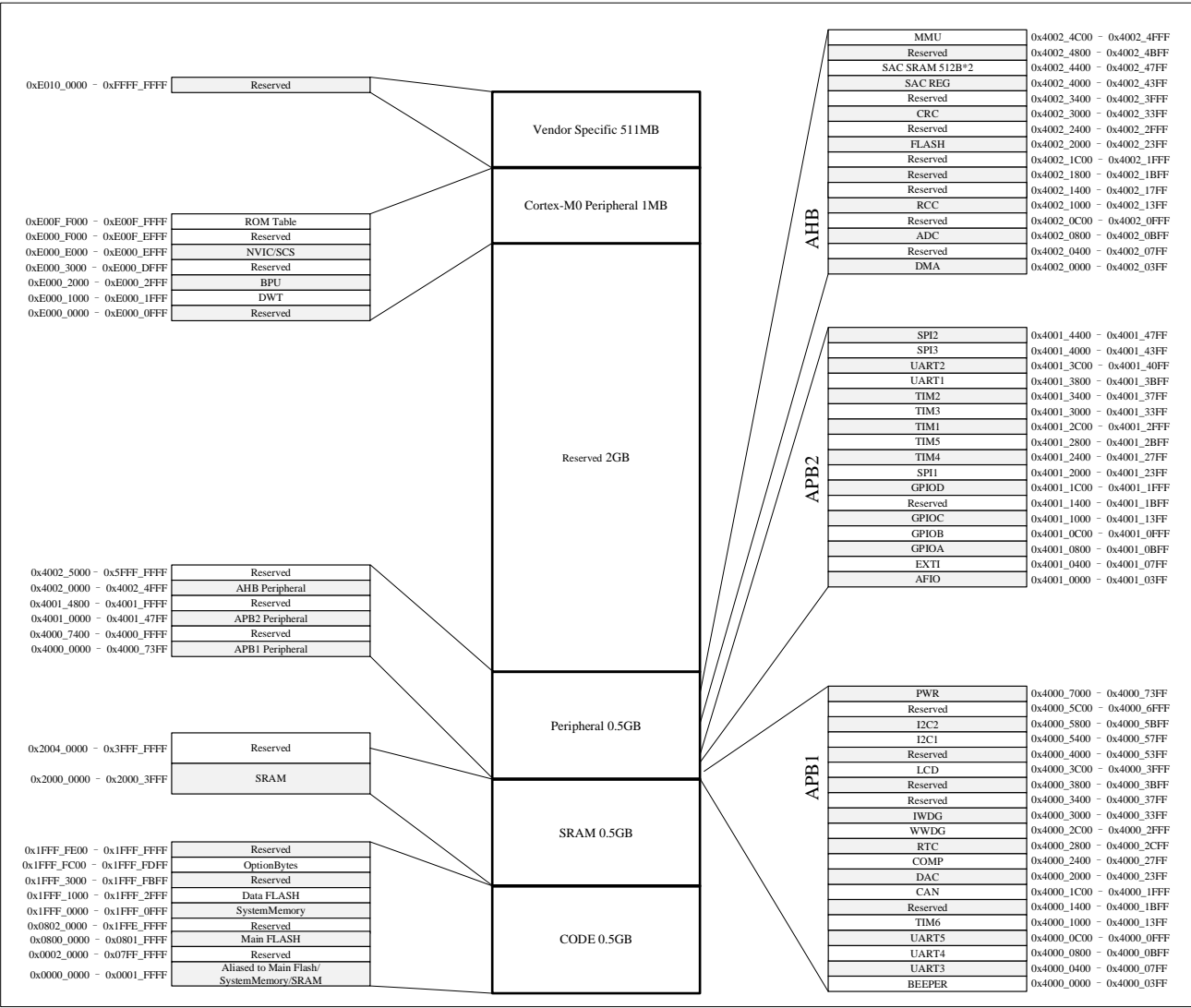


表 2-1 外设寄存器地址列表

地址范围	外设	总线
0x4002_4800 – 0x5FFF_FFFF	Reserved	AHB

地址范围	外设	总线
0x4002_4400 – 0x4002_47FF	SAC SRAM 512B * 2	
0x4002_4000 – 0x4002_43FF	SAC REG	
0x4002_3400 – 0x4002_3FFF	Reserved	
0x4002_3000 – 0x4002_33FF	CRC	
0x4002_2400 – 0x4002_2FFF	Reserved	
0x4002_2000 – 0x4002_23FF	FLASH	
0x4002_1400 – 0x4002_1FFF	Reserved	
0x4002_1000 – 0x4002_13FF	RCC	
0x4002_0C00 – 0x4002_0FFF	Reserved	
0x4002_0800 – 0x4002_0BFF	ADC	
0x4002_0400 – 0x4002_07FF	Reserved	
0x4002_0000 – 0x4002_03FF	DMA	
0x4001_4800 – 0x4001_FFFF	Reserved	APB2
0x4001_4400 – 0x4001_47FF	SPI2	
0x4001_4000 – 0x4001_43FF	SPI3	
0x4001_3C00 – 0x4001_3FFF	UART2	
0x4001_3800 – 0x4001_3BFF	UART1	
0x4001_3400 – 0x4001_37FF	TIM2	
0x4001_3000 – 0x4001_33FF	TIM3	
0x4001_2C00 – 0x4001_2FFF	TIM1	
0x4001_2800 – 0x4001_2BFF	TIM5	
0x4001_2400 – 0x4001_27FF	TIM4	
0x4001_2000 – 0x4001_23FF	SPI1	
0x4001_1C00 – 0x4001_1FFF	GPIOD	
0x4001_1400 – 0x4001_1BFF	Reserved	
0x4001_1000 – 0x4001_13FF	GPIOC	
0x4001_0C00 – 0x4001_0FFF	GPIOB	
0x4001_0800 – 0x4001_0BFF	GPIOA	
0x4001_0400 – 0x4001_07FF	EXTI	
0x4001_0000 – 0x4001_03FF	AFIO	
0x4000_7400 – 0x4000_FFFF	Reserved	APB1
0x4000_7000 – 0x4000_73FF	PWR	
0x4000_5C00 – 0x4000_6FFF	Reserved	
0x4000_5800 – 0x4000_5BFF	I2C2	
0x4000_5400 – 0x4000_57FF	I2C1	
0x4000_4000 – 0x4000_53FF	Reserved	
0x4000_3C00 – 0x4000_3FFF	LCD	
0x4000_3800 – 0x4000_3BFF	Reserved	
0x4000_3400 – 0x4000_37FF	Reserved	

地址范围	外设	总线
0x4000_3000 – 0x4000_33FF	IWDG	
0x4000_2C00 – 0x4000_2FFF	WWDG	
0x4000_2800 – 0x4000_2BFF	RTC	
0x4000_2400 – 0x4000_27FF	COMP	
0x4000_2000 – 0x4000_23FF	DAC	
0x4000_1C00 – 0x4000_1FFF	CAN	
0x4000_1400 – 0x4000_1BFF	Reserved	
0x4000_1000 – 0x4000_13FF	TIM6	
0x4000_0C00 – 0x4000_0FFF	UART5	
0x4000_0800 – 0x4000_0BFF	UART4	
0x4000_0400 – 0x4000_07FF	UART3	
0x4000_0000 – 0x4000_03FF	BEEPER	

2.1.3 启动管理

2.1.3.1 启动地址

在系统启动时，可以通过 BOOT0 引脚和用户选项字节 BOOT 配置，来选择在复位后的启动模式，在系统复位后或从掉电模式退出时，BOOT 引脚的值将被重新锁存。经过启动延迟之后，CPU 从地址 0x0000_0000 获取堆栈顶的地址，并从地址 0x0000_0004 指示的复位向量地址开始执行代码。由于 Cortex®-M0 始终从地址 0x0000_0000 和 0x0000_0004 获取堆栈顶指针和复位向量，所以启动仅适合于从 CODE 代码区开始，设计上需要对启动空间进行地址重映射。有三种启动模式可选：

- 从主闪存存储器(Main Flash)启动：
 - ◆ 主闪存存储器被映射到启动空间（0x0000_0000）；
 - ◆ 主闪存存储器可在两个地址区域访问，0x0000_0000 或 0x0800_0000；
- 从系统存储器(System Memory)启动：
 - ◆ 系统存储器被映射到启动空间（0x0000_0000）；
 - ◆ 系统存储器可在两个地址区域访问，0x0000_0000 或 0x1FFF_0000；
- 从内置 SRAM 启动：
 - ◆ 内置 SRAM 被映射到启动空间（0x0000_0000）；
 - ◆ 内置 SRAM 可在两个地址区域访问，0x0000_0000 或 0x2000_0000；

2.1.3.2 启动配置

可以通过 BOOT0 引脚和用户选项字节 BOOT 配置选择三种不同启动模式

表 2-2 启动模式列表

启动模式选择引脚					启动模式	对应启动模式下，访问内存空间的起始地址		
nBOOT1	nBOOT0	BOOT0 引脚	nSWBOOT0	BOOT0 _CFG		Main Flash	System Memory	SRAM
X	X	0	1	1	Main Flash 启动	0x0000_0000	0x1FFF_0000	0x2000_0000
X	1	X	0			0x0800_0000		
1	X	1	1		System Memory 启动	0x08000000	0x0000_0000 0x1FFF_0000	0x2000_0000
1	0	X	0					
0	X	1	1		SRAM 启 动	0x08000000	0x1FFF_0000	0x0000_0000 0x2000_0000
0	0	X	0					
X	X	1	1	0	Main Flash 启动	0x0000_0000	0x1FFF_0000	0x2000_0000
X	1	X	0			0x0800_0000		
1	X	0	1		System Memory 启动	0x08000000	0x0000_0000 0x1FFF_0000	0x2000_0000
1	0	X	0					
0	X	0	1		SRAM 启 动	0x08000000	0x1FFF_0000	0x0000_0000 0x2000_0000
0	0	X	0					

注：其中BOOT0和GPIO复用，上电默认状态受控于BOOT0_CFG值。

2.1.3.3 内嵌启动程序

内嵌的自举程序存放在系统存储器 System Memory 内，用于通过 UART1 对闪存存储器进行重新编程。而 UART1 接口除了可以依靠外部时钟（HSE）外，还可以依靠内部 8MHz 振荡器（HSI）运行。进一步的细节请查询自举程序手册。

2.1.3.4 BOOT SWAP 功能

此功能可以用于更新二级 boot，避免直接对二级 boot 进行更新时因为临时断电或其他原因导致二级 boot 区代码更新失败，且原二级 boot 代码丢失导致程序无法正常工作，boot swap 功能可以有效的避免这个问题。

应用实现举例：

假设旧的二级 boot 和新的二级 boot 都是 4K 以下大小

1. 芯片上电后，默认从 0x8000000 启动
2. 收到上位机指令，要对二级 boot 进行编程
3. 旧的二级 boot 将 0x8001000-0x8001FFF 区域擦除
4. 旧的二级 boot 接收上位机下载程序，将新的二级 boot 编程到 0x8001000-0x8001FFF 区域
5. 新的二级 boot 编程完成后，旧的二级 boot 将选项字节的 USER5[7:0]改为 0xF7，并进行软件复位
6. 软件复位后，芯片从 0x8001000 启动，即新的二级 boot 区域启动
7. 此时新的二级 boot 检查到 USER5[7:0]非 0xFF，会将自身复制到 0x8000000，在复制完成后将 USER5[7:0]改成 0xFF，并复位

8. 芯片复位后，就又从 0x8000000 启动，并且启动程序变成新的二级 boot 了

2.2 存储系统（Memory system）

程序存储器、数据存储器、寄存器和输入输出端口被组织在同一个 4GB 的线性地址空间内。数据字节以小端格式存放在存储器中，一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。对程序存储器和数据存储器的规格说明如下。

2.2.1 FLASH 规格

Flash 由主存储区、信息区组成，以下分别进行说明：（以下说明中的容量值不含 ECC）

- 主存储区最大为 128KB，也称作主闪存存储器，包含 256 个 Page，用于用户程序的存放和运行，以及数据存储。
- 信息区为 15.5KB，包含 31 个 Page，由 Data FLASH 区（8KB）、系统存储区（4KB）、系统配置区（3KB）、选项字节区（0.5KB）组成：
 - Data FLASH 区为 8KB，包含 16 个 Page，仅用于存储用户数据，不可用于代码执行，可以读写擦。
 - ◆ 系统存储区为 4KB，包含 8 个 Page，也称作 System Memory，用于引导程序（BOOT）的存放和运行。
 - ◆ 系统配置区为 3KB，包含 6 个 Page。
 - ◆ 选项字节区为 0.5KB，包含 1 个 Page，也称作 OptionByte，有效空间为 112B，BOOT 程序、用户程序均可以读写擦。

2.2.1.1 存储地址

主存储区、信息区都分配了总线地址空间。

表 2-3 存储总线地址列表

存储区	页名称	地址范围	大小
主存储区	页 0	0x0800_0000 – 0x0800_01FF	0.5KB
	页 1	0x0800_0200 – 0x0800_03FF	0.5KB
	页 2	0x0800_0400 – 0x0800_05FF	0.5KB
	⋮	⋮	⋮
	页 255	0x0801_FE00 – 0x0801_FFFF	0.5KB
信息区	系统存储区	0x1FFF_0000 – 0x1FFF_0FFF	4KB
	Data FLASH 区	0x1FFF_1000 – 0x1FFF_2FFF	8KB
	系统配置区	0x1FFF_F000 – 0x1FFF_FBF	3KB
	选项字节区	0x1FFF_FC00 – 0x1FFF_FC6F	112B
存储区接口 寄存器	FLASH_AC	0x4002_2000 – 0x4002_2003	4B
	FLASH_KEY	0x4002_2004 – 0x4002_2007	4B
	FLASH_OPTKEY	0x4002_2008 – 0x4002_200B	4B
	FLASH_STS	0x4002_200C – 0x4002_200F	4B
	FLASH_CTRL	0x4002_2010 – 0x4002_2013	4B

存储区	页名称	地址范围	大小
	FLASH_ADD	0x4002_2014 – 0x4002_2017	4B
	保留	0x4002_2018 – 0x4002_201B	4B
	FLASH_OB	0x4002_201C – 0x4002_201F	4B
	FLASH_WRP	0x4002_2020 – 0x4002_2023	4B
	FLASH_ECC	0x4002_2024 – 0x4002_2027	4B
	保留	0x4002_2028 – 0x4002_2047	32B
	FLASH_USER	0x4002_2048 – 0x4002_204B	4B
	FLASH_START_ADD	0x4002_204C – 0x4002_204F	4B
	FLASH_VTOR	0x4002_2050 – 0x4002_2053	4B

闪存存储器被组织成 64 位宽的存储器单元，可以存放代码和数据常数。

信息区分为三个部分：

- 系统存储区是用于存放在系统存储器自举模式下的启动程序，启动程序使用 UART1 接口实现对闪存存储器的编程。
- 系统配置区，包含芯片基本信息。
- 选项字节区。

对主存储器和信息块的写入由内嵌的闪存编程/擦除控制器管理。

主闪存存储器（main flash）有两种保护方式防止非法的访问（读、写、擦除）：

- 页写入保护（WRP）
- 读出保护（RDP）

DATA flash 区仅支持读出保护。

在执行 FLASH 写操作时，任何对 FLASH 的读操作都会锁住总线，在写操作完成后读操作才能正确地进行；即在写或擦除操作时，不能进行代码或数据的读取操作。

进行闪存编程操作时（写或擦除），必须打开内部的 RC 振荡器（HSI）。

注：在低功耗模式下，所有闪存存储器的操作都被中止。

2.2.1.2 读写操作

Flash 写操作仅支持 64 位操作，写操作之前先擦除 Flash，擦除最小块大小是一个页 0.5KB。写操作分为擦除和编程阶段。

读 Flash 时，读的等待周期数可以通过寄存器配置。使用时，需要结合 SYSCLK 时钟频率进行计算。比如：当 $\text{SYSCLK} \leq 24\text{MHz}$ 时，等待周期数最小为 0；当 $24\text{MHz} < \text{SYSCLK} \leq 48\text{MHz}$ 时，等待周期数最小为 1；当 $48\text{MHz} < \text{SYSCLK} \leq 64\text{MHz}$ 时，等待周期数最小为 2。

注意：无论等待周期数是否不为零，启用预取缓冲功能都可以提高整体读代码的效率。

2.2.1.3 Flash 解锁操作

复位后，Flash 模块是被保护的，不能写入 FLASH_CTRL 寄存器，以防因电气干扰等原因产生对 Flash 的意外操作。通过写入特定的键值序列到 FLASH_KEY 寄存器，可以开启对 FLASH_CTRL 寄存器的操作权限，这个特定的序列是：第一次在 Flash 密钥寄存器（FLASH_KEY）中写入 $\text{KEY1} = 0x45670123$ ，第二次

则在 Flash 密钥寄存器（FLASH_KEY）中写入 KEY2 = 0xCDEF89AB。

如果顺序出现错误或键值出现错误，将返回总线错误并锁定 FLASH_CTRL 寄存器，直到下一次复位，软件可以通过查看 FLASH_CTRL.LOCK 位来确认 Flash 是否已解锁。若需要进行正常的锁定设置，可以通过软件将 FLASH_CTRL.LOCK 位置 1 来实现，此后可以通过在 FLASH_KEY 中写入正确的键值系列来对 Flash 解锁。

2.2.1.4 擦除和编程

2.2.1.4.1 主存储区和 DATA flash 区擦除

主存储区可以按页擦除或者整片擦除，DATA flash 区仅支持按页擦除

页擦除

页擦除流程：

- 通过检查 FLASH_STS.BUSY 位来确保没有正在进行闪存操作；
- 设置 FLASH_CTRL.PER 为'1'；
- 将要擦除的页起始地址写入 FLASH_ADD 寄存器；
- 设置 FLASH_CTRL.START 为'1'；
- 等待 FLASH_STS.BUSY 变为'0'；
- 读出被擦除页的内容检查是否被擦除。

片擦除

片擦除流程：

- 通过检查 FLASH_STS.BUSY 位来确保没有正在进行闪存操作；
- 设置 FLASH_CTRL.MER 为'1'；
- 设置 FLASH_CTRL.START 为'1'；
- 等待 FLASH_STS.BUSY 位变为'0'；
- 读出所有被擦除页的内容检查是否被擦除。

2.2.1.4.2 主存储区和 DATA flash 编程

对主存储区和 DATA flash 编程每次可以写入 64 位。当 FLASH_CTRL.PG 为'1'时，在一个闪存地址写入两个字将启动一次编程；写入任何字节、半字的数据，都会产生总线错误；写入任何单字的数据都会产生编程错误；在编程过程中(FLASH_STS.BUSY 为'1')，任何读写闪存的操作都会使 CPU 暂停，直到此次闪存编程结束。

主存储区和 DATA flash 编程流程：

- 通过检查 FLASH_STS.BUSY 位来确保没有正在进行闪存操作；
- 设置 FLASH_CTRL.PG 为'1'；
- 在指定的地址写入要编程的双字；
- 等待 FLASH_STS.BUSY 变为'0'；
- 读出写入地址的数据检查是否正确。

注意：当 `FLASH_STS.BUSY` 为'1'时，不能对任何 Flash 寄存器执行写操作。

2.2.1.4.3 选项字节区擦除和编程

对选项字节区的编程与主存储区不同。选项字节的数目只有 14 个字节(4 个字节作为写保护，2 个字节作为读保护，6 个字节为配置选项，2 个字节存储用户数据)。对 Flash 解锁后，必须分别写入 `KEY1` 和 `KEY2`(见 2.2.1.3)到 `FLASH_OPTKEY` 寄存器，再设置 `FLASH_CTRL.OPTWE` 为'1'，此时可以对选项字节区进行编程：设置 `FLASH_CTRL.OTPG` 为'1'后写入双字到指定的地址。

对选项字节区字编程时，使用半字中的低字节并自动地计算出高字节(高字节为低字节的补码)，并开始编程操作，这将保证选项字节和它的补码始终是正确的。

选项字节区擦除过程：

- 通过检查 `FLASH_STS.BUSY` 位来确保没有正在进行闪存操作；
- 解锁 `FLASH_CTRL.OPTWE`；
- 设置 `FLASH_CTRL.OPTER` 为'1'；
- 设置 `FLASH_CTRL.START` 为'1'；
- 等待 `FLASH_STS.BUSY` 变为'0'；
- 读出被擦除选项字节的内容检查是否被擦除。

选项字节区编程流程：

- 通过检查 `FLASH_STS.BUSY` 位来确保没有正在进行闪存操作；
- 解锁 `FLASH_CTRL.OPTWE`；
- 设置 `FLASH_CTRL.OTPG` 为'1'；
- 在指定的地址写入要编程的双字；
- 等待 `FLASH_STS.BUSY` 变为'0'；
- 读出写入地址的数据检查是否正确。

2.2.1.5 ECC 功能

Flash 模块支持 ECC 功能，实现 2-bit 检错和 1-bit 纠错。ECC 编码、解码（纠错、检错）由硬件自动执行，如果检测到错误，置错误位并产生中断。

2.2.1.6 指令预取

Flash 模块的指令预取功能，支持 16B 的预取 Buffer。通过指令预取操作，可提高 CPU 的指令执行效率。指令预取功能可以通过寄存器配置为使能或除能，默认使能。

2.2.1.7 选项字节

选项字节块主要用于配置读写保护、BOOT 模式配置、软件/硬件看门狗以及系统处于 power-down 或 stop 模式下的复位选项，并分配了总线地址空间，可以进行读写访问。它们由有 14 个选项字节组成：4 个字节作为写保护，2 个字节作为读保护，6 个字节作为配置选项，2 个字节由用户定义，这 14 个字节需要通过总线写入。选项字节块同时还包含与这 14 个选项字节相对应的补码，这些补码需要在总线写入选项字节时，由硬件自动计算出来，一起写入 Flash，并用于选项字节读取时的验证。

默认状态下，选项字节块始终是可以读且被写保护。要想对选项字节块进行写操作（编程/擦除），首先要

解锁 Flash，然后解锁选项字节：在 FLASH_OPTKEY 中写入正确的键值序列（KEY1 = 0x45670123，KEY2 = 0xCDEF89AB），随后对选项字节块的写操作将被允许。如果顺序出现错误或键值出现错误，将返回总线错误并锁定选项字节，直到下一次复位。若需要正常进行锁定设置，可以通过软件将 FLASH_CTRL.OPTWE 位写 0 来实现，此后可以通过在 FLASH_OPTKEY 中写入正确的键值系列来对选项字节解锁。

每次系统复位后，从 Flash 的选项字节块中读出选项字节数据，并保存在具有只读属性的选项字节寄存器（FLASH_OB/ FLASH_USER/FLASH_WRP）中；同时一起读出来的选项字节补码数据，将用于验证选项字节数据是否正确，如果不匹配，将产生一个选项字节错误标志（FLASH_OB.OBERR）。当发生选项字节错误时，对应的选项字节被强置为 0xFF。当选项字节和它的补码均为 0xFF 时（擦除后的状态），则略过上述验证步骤，无需进行验证。

表 2-4 选项字节列表

地址	[63:16] Reserved	[15:8] 相应反码	[7:0] 选项字节
0x1FFF_FC00	-	nRDP1	RDP1
0x1FFF_FC08	-	nUSER1	USER1
0x1FFF_FC10	-	nUSER2	USER2
0x1FFF_FC18	-	nUSER3	USER3
0x1FFF_FC20	-	nUSER4	USER4
0x1FFF_FC28	-	nUSER5	USER5
0x1FFF_FC30	-	nUSER6	USER6
0x1FFF_FC38	-	nData0	Data0
0x1FFF_FC40	-	nData1	Data1
0x1FFF_FC48	-	nWRP0	WRP0
0x1FFF_FC50	-	nWRP1	WRP1
0x1FFF_FC58	-	nWRP2	WRP2
0x1FFF_FC60	-	nWRP3	WRP3
0x1FFF_FC68	-	nRDP2	RDP2

■ 读保护 L1 等级：RDP1

- ◆ 保护存储在闪存中的代码；
- ◆ 当写入正确的是数值时，将禁止读出闪存存储器；
- ◆ RDP1 是否开启的结果，可通过 FLASH_OB[1]查询；

■ 用户配置选项 1: USER1

- ◆ USER1[7]: 保留
- ◆ USER1[6]: BOOT0_CFG 配置选项，可通过 FLASH_OB[8]查询
- ◆ USER1[5]: nSWBOOT0_SEL 配置选项，可通过 FLASH_OB[7]查询
- ◆ USER1[4]: nBOOT1 配置选项，可通过 FLASH_OB[6]查询
- ◆ USER1[3]: nBOOT0 配置选项，可通过 FLASH_OB[5]查询
- ◆ USER1[2]: nRST_PD 配置选项，可通过 FLASH_OB[4]查询

0: 当进入 PD 模式时产生复位

1: 进入 PD 模式时不产生复位

◆ USER1[1]: nRST_STOP 配置选项, 可通过 FLASH_OB[3]查询

0: 当进入停机 (STOP) 模式时产生复位

1: 进入停机 (STOP) 模式时不产生复位

◆ USER1[0]: IWDG_SW 配置选项, 可通过 FLASH_OB[2]查询

0: 硬件看门狗

1: 软件看门狗

■ 用户配置选项字节: USERx

◆ USER2[3:0]: LVR 上电电压控制, 可通过 FLASH_USER[3:0]查询

◆ USER2[4]: LVR 使能, 可通过 FLASH_USER[4]查询

◆ USER2[5]: LVR 复位使能, 可通过 FLASH_USER[5]查询

◆ USER2[6]: LVR 过滤器使能, 可通过 FLASH_USER[6]查询

◆ USER2[7]: 保留

◆ USER3[7:0]: LVR 过滤器计数器控制, 可通过 FLASH_USER[15:8]查询

◆ USER4[7:0]: 上电延时复位控制, 可通过 FLASH_USER[23:16]查询

◆ USER5[7:0]: FLASH 启动起始地址, 可通过 FLASH_START_ADD[7:0]查询

◆ USER6[1:0]: 用于选择 boot 的串口引脚, 可通过 FLASH_OB [27:26]查询

◆ USER6[7:2]: 保留

■ 2 字节用户数据: Datax

◆ Data1 (FLASH_OB[25:18])

◆ Data0 (FLASH_OB [17:10])

■ 写保护选项字节: WRP0 ~ 3, 可通过寄存器 FLASH_WRP[31:0]查询

◆ WRP0: 第 0 ~63 页的写保护, bit[0]对应 Page (0~7),, bit[7]对应 Page (56~63)

◆ WRP1: 第 64~127 页的写保护, bit[0]对应 Page (64~71),, bit[7]对应 Page (120~127)

◆ WRP2: 第 128~191 页的写保护, bit[0]对应 Page (128~135),, bit[7]对应 Page (184~191)

◆ WRP3: 第 192~255 页的写保护, bit[0]对应 Page (192~199),, bit[7]对应 Page (248~255)

■ 读保护 L2 等级: RDP2

◆ 在 L1 的基础上增加保护功能, 具体见 2.2.1.9 读保护的详细描述;

◆ RDP2 是否开启的结果, 可通过 FLASH_OB[31]查询;

2.2.1.8 写保护

可以对 Flash 主存储区 (最大 128KB) 的所有 Page 配置写保护, 以防在程序跑飞或电气干扰等原因导致的

意外写操作，写保护的基本单位是：对于 Page0~255，每 8 页为一个基本保护单元。写保护可以通过设置选项字节块中的 WRP0~3 来进行配置；每次进行配置后，需要进行一次系统复位，配置的值才能生效。如果对一个受保护的页面进行编程或擦除操作，FLASH_STS 中将会返回一个保护错误标志。

系统信息区中的 data flash 区（8KB），存放用户数据，可更改。

系统信息区中的系统存储块（4KB），存放了 BOOT 程序，不可更改。

系统信息区中的系统配置块（3KB），存放了芯片基本信息，不可更改。

系统信息区中的选项字节块（0.5KB），存放了用户可配置选项字节信息，将 FLASH_CTRL.OPTWE 写 0 使能选项字节块的写保护，之后通过在 FLASH_OPTKEY 中写入正确的键值序列，来对选项字节解除写保护。

2.2.1.9 读保护

Flash 中的用户代码可以通过设置读保护来防止被非法读取。读保护通过配置选项字节块中的 RDP 字节进行设置，可以配置 3 种不同的读保护级别，如下列表

表 2-5 读保护配置列表

读保护等级	RDP1	nRDP1	nRDP2	RDP2
L0 level	0xA5	0x5A	RDP2! = 0xCC nRDP2! = 0x33	
L2 level	0xFF	0xFF	0x33	0xCC
L1 level	非以上两种配置			

■ L0 等级：

- ◆ 处于未保护状态，(RDP1 == 0xA5 & nRDP1 == 0x5A) && (RDP2! = 0xCC | nRDP2! = 0x33)
- ◆ 主存储区、DATA Flash 区和选项字节可以被任意读取
- ◆ 主存储区、DATA Flash 区和选项字节可以进行编程和擦除，可配置读写保护

■ L1 等级：

- ◆ ~ ((RDP1 == 0xA5 & nRDP1 == 0x5A) && (RDP2! = 0xCC | nRDP2! = 0x33)) | (RDP2 == 0xCC & nRDP2 == 0x33))
- ◆ 只允许从用户代码中对主存储区、DATA Flash 区的读操作，即以非调试方式从主闪存存储器启动程序的情况才允许对主存储区的读操作
- ◆ 全部主存储区页、DATA Flash 区可以通过在主闪存存储器中执行的代码进行编程（实现 IAP 或数据存储等功能）
- ◆ 全部主存储区页、DATA Flash 区不允许在调试模式下或从内部 SRAM 启动后执行写或擦除操作（整片擦除除外）
- ◆ 所有通过 JTAG/SWD 向内置 SRAM 装载代码并执行代码的功能依然有效，亦可以通过 JTAG/SWD 从内置 SRAM 启动，这个功能可以用来解除读保护；
- ◆ 当读保护的选项字节被改写为未保护的 L0 级别时，将会自动擦除全部主存储区、DATA Flash 区，执行的过程如下：（擦除选项字节块不会导致自动的擦除操作，因为擦除的结果是 0xFF，相当于仍然处于 L1 级别的保护状态）

- 在 FLASH_OPTKEY 中写入正确的键值序列解锁选项字节区；

- 总线发起命令擦除整个选项字节区（Page 擦）；
- 总线写入读保护选项字节 0xA5；
- 内部自动擦除全部主存储区；
- 内部自动写入 0xA5 到读保护选项字节；
- 进行系统复位（如软件复位等），选项字节块（包括新的 RDP 值 0xA5）将被重新加载到系统中，读保护被解除；

- L2 等级：除了 SRAM 启动被禁止、调试模式被禁止、选项字节写/页擦被禁止、保护级别不可修改（不可逆）之外，其余特性同 L1 级别。L2 级别通过配置另一个选项字节 RDP2 来实现，不管 RDP1 为何值，只要满足（RDP2=0xCC & nRDP2=0x33）即为 L2 级别

2.2.1.10 权限保护

- Flash 主存储区、DATA Flash 区权限：
 - ◆ L0 级别下：主存储区、DATA Flash 区都可以被读取；主存储区可以配置各 Page 的写保护属性；
 - ◆ L1/2 级别下：
 - 在 SWD 调试模式下或从内部 SRAM 启动后执行时，所有 Page 不允许（W/R/PE）操作；
 - 所有 Page 可以通过在 Flash 主存储器区中执行的代码进行编程（实现 IAP 或数据存储等功能）；
 - 主存储区所有 Page 可以配置各 Page 的写保护属性；
 - 当 L1 级别改写为 L0 级别时，将会自动擦除全部 Flash 主存储区、DATA Flash 区；
- Flash 选项字节区权限：
 - ◆ L0/L1 级别下，都被允许访问（W/R/PE）；
 - ◆ L2 级别下：除了调试模式被禁止外，都允许只读访问 Flash 选项字节区；
- Flash 系统存储区权限：
 - ◆ 只有 system memory 区执行的代码，才允许访问（W/R/PE）；Flash 和 sram 执行代码或者通过调试接口都不允许访问；
 - ◆ 在 L1/L2 级别下，sram 启动跳转到 system memory 执行代码不允许访问（W/R/PE）。其他情况允许访问（W/R/PE）；
- Flash 系统配置区
 - ◆ 用户信息：仅 system memory 启动后执行或跳转到 system memory 执行才可以读操作；
 - ◆ ID 信息：L0 级别下可读，L1/L2 级别下 SWD 调试模式下访问禁止，L1 级别下 sram 启动后执行下访问禁止，详细见表 2-6；

表 2-6 存储区读写擦⁽¹⁾权限控制表

保护 级别	启动模式	Main Flash				修改保护级别
	执行用户	SWD	Main Flash	System Memory	SRAM	

	访问区域					
L0级别	Flash主存储区	读写擦	读写擦	读写擦	读写擦	允许改为L1或L2
	DATA FLASH区	读写擦	读写擦	读写擦	读写擦	
	Flash主存储区片擦 ⁽²⁾	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	SRAM（All）	读写	读写	读写	读写	
L1级别	Flash主存储区	禁止	读写擦	读写擦	禁止	允许改为 L0 或 L2。 改为L0时，主存储区、DATA Flash区将被自动擦除。
	DATA FLASH区	禁止	读写擦	读写擦	禁止	
	Flash主存储区片擦 ⁽²⁾	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	SRAM（All）	读写	读写	读写	读写	
L2级别	Flash主存储区	SWD 接口被禁止	读写擦	读写擦	禁止	不允许修改。
	DATA FLASH区		读写擦	读写擦	禁止	
	Flash主存储区片擦 ⁽²⁾		允许	允许	允许	
	Flash选项字节区		只读	只读	只读	
	Flash系统存储区		禁止	读写擦	禁止	
	SRAM（All）		读写	读写	读写	
保护级别	启动模式	SRAM				修改保护级别
	执行用户	SWD	Main Flash	System Memory	SRAM	
	访问区域					
L0级别	Flash主存储区	读写擦	读写擦	读写擦	读写擦	允许改为L1或L2

	DATA FLASH区	读写擦	读写擦	读写擦	读写擦	
	Flash主存储区片擦 ⁽²⁾	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	SRAM（All）	读写	读写	读写	读写	
L1级别	Flash主存储区	禁止	读写擦	禁止	禁止	允许改为 L0 或 L2。 改为L0时，主存储区、DATA Flash区将被自动擦除。
	DATA FLASH区	禁止	读写擦	禁止	禁止	
	Flash主存储区片擦 ⁽²⁾	允许	允许	禁止	允许	
	Flash选项字节区	读写擦	读写擦	禁止	读写擦	
	Flash系统存储区	禁止	禁止	禁止	禁止	
	SRAM（All）	读写	读写	禁止	读写	
L2级别	Flash主存储区	L2保护级别，无法从SRAM启动				不允许修改。 SWD 被禁止。
	DATA FLASH区					
	Flash主存储区片擦 ⁽²⁾					
	Flash选项字节区					
	Flash系统存储区					
	SRAM（All）					
保护级别	启动模式	System Memory				修改保护级别
	<div>执行用户 访问区域</div>	SWD	Main Flash	System Memory	SRAM	
L0级别	Flash主存储区	读写擦	读写擦	读写擦	读写擦	允许改为L1或L2
	DATA FLASH区	读写擦	读写擦	读写擦	读写擦	
	Flash主存储区片擦 ⁽²⁾	允许	允许	允许	允许	

	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	SRAM (All)	读写	读写	读写	读写	
L1级别	Flash主存储区	禁止	读写擦	读写擦	禁止	允许改为 L0 或 L2。 改为L0时，主存储区、DATA Flash区将被自动擦除。
	DATA FLASH区	禁止	读写擦	读写擦	禁止	
	Flash主存储区片擦 ⁽²⁾	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	SRAM (All)	读写	读写	读写	读写	
L2级别	Flash主存储区	SWD 接口被禁止	读写擦	读写擦	禁止	不允许修改
	DATA FLASH区		读写擦	读写擦	禁止	
	Flash主存储区片擦 ⁽²⁾		允许	允许	允许	
	Flash选项字节区		只读	只读	只读	
	Flash系统存储区		禁止	读写擦	禁止	
	SRAM (All)		读写	读写	读写	

注：1.这里的擦是指Flash页擦除；

2.Flash主存储区片擦除是指mass erase，仅在L2级别调试模式下禁止；。

2.2.2 SRAM

SRAM 主要用于代码运行，存放程序执行过程中的变量和数据或堆栈，容量最大为 16KB。

SRAM 支持字节、半字、字的读写访问。

SRAM 支持代码运行，可以在 SRAM 全速运行程序。SRAM 的最大地址范围是 0x2000 0000~0x2000 3FFF。

SRAM 在 Power Down 模式下数据不能保持；其他工作模式（Run /Sleep/Stop）数据可以正常保持。

主要特性如下：

- 容量最大总共为 16KB
- 支持字节/半字/字读写
- CPU/DMA 均可访问

■ CPU BUS 可以 Remap 到 SRAM 全速运行程序

2.2.3 FLASH 寄存器描述

必须以字（32 位）的方式操作寄存器。

2.2.3.1 FLASH 寄存器总览

表 2-7 FLASH 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000h	FLASH_AC	Reserved																								PRFTBFS	PRFTBFE	Reserved	LATENCY						
	Reset Value																									1	1		0	0	0				
004h	FLASH_KEY	FKEY																																	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
008h	FLASH_OPTKEY	OPTKEY																																	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
00Ch	FLASH_STS	Reserved																								ECCERR	Reserved	EOP	WRPERR	Reserved	PGERR	Reserved	BUSY		
	Reset Value																									0		0	0		0		0	0	
010h	FLASH_CTRL	Reserved																ECCERRITE	EOPITE	Reserved	ERRITE	OPTWE	Reserved	LOCK	START	OPTER	OPTPG	Reserved	MER	PER	PG				
	Reset Value																	0	0			0		0		1	0		0	0		0	0	0	
014h	FLASH_ADD	FADD																																	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
018h	Reserved																																		
01Ch	FLASH_OB	RDPRT2	Reserved				BOOT_SEL		Data1								Data0								Not Used		BOOT0_CFG	nSWBOOT0	nBOOT1	nBOOT0	nRST_PD	nRST_STOP	WDG_SW	RDPRT1	OBERR
	Reset Value	0					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0		
020h	FLASH_WRP	WRPT																																	
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
024h	FLASH_ECC	Reserved																								ECCLW									
	Reset Value																									0	0	0	0	0	0	0	0	0	0
028h~047h	Reserved																																		
048h	FLASH_USER	Reserved								POR_DELAY				LVRCNT								Reserved	LVRFILEN	LVRIRST	LVRLEN	LVRLS									
	Reset Value									1	1	1	1	1	1	1	1	1	0	0	0		0	0	0	0	0	0	0	0	0	0	0		

04Ch	FLASH_START	Reserved																										START_ADD																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
	_ADD																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												

2.2.3.2 FLASH 控制和状态寄存器

有关寄存器说明中的缩写，请见 1.1 节

2.2.3.2.1 FLASH 访问控制寄存器 (FLASH_AC)

偏移地址：0x00

复位值：0x0000 0030

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

位域	名称	描述
31:6	Reserved	保留，必须保持复位值
5	PRFTBFS	预取缓冲区状态 该位指示预取缓冲区的状态 0：预取缓冲区关闭 1：预取缓冲区开启
4	PRFTBFE	预取缓冲区使能 0：关闭预取缓冲区 1：启用预取缓冲区
3:2	Reserved	保留，必须保持复位值
1:0	LATENCY	时延 这些位表示 SYSCLK（系统时钟）周期与闪存访问时间的比例 00：零周期时延，当 $0 < \text{SYSCLK} \leq 24\text{MHz}$ 01：一个周期时延，当 $24\text{MHz} < \text{SYSCLK} \leq 48\text{MHz}$ 10：两个周期时延，当 $48\text{MHz} < \text{SYSCLK} \leq 64\text{MHz}$ 11：保留

2.2.3.2.2 FLASH 键寄存器 (FLASH_KEY)

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEY[31:16]															
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
FKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:0	FKEY	用于解锁 FLASH_CTRL.LOCK 位

2.2.3.2.3 FLASH OPTKEY 寄存器 (FLASH_OPTKEY)

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY [31:16]															
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
OPTKEY [15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:0	OPTKEY	用于解锁 FLASH_CTRL.OPTWE 位

2.2.3.2.4 FLASH 状态寄存器 (FLASH_STS)

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ECCERR	Reserved	EOP	WRPERR	Reserved	PGERR	Reserved	BUSY
								rc_w1		rc_w1		rc_w1		rc_w1	
														r	

位域	名称	描述
31:6	Reserved	保留，必须保持复位值
7	ECCERR	ECC 错误 读 FLASH 时报错，硬件设置这位为'1'，写入'1'可以清除这位状态。
6	Reserved	保留，必须保持复位值
5	EOP	操作结束 当闪存操作（编程/擦除）完成时，硬件设置这位为'1'，写入'1'可以清除这位状态。 <i>注：每次成功的编程或擦除都会设置 EOP 状态。</i>
4	WRPERR	写保护错误 试图对写保护的闪存地址编程时，硬件设置这位为'1'，写入'1'可以清除这位状态。
3	Reserved	保留，必须保持复位值
2	PGERR	编程错误 试图对内容不是'0xFFFF_FFFF'的地址编程时，硬件设置这位为'1'，写入'1'可以清除这位状态。 <i>注：进行编程操作之前，必须先清除 FLASH_CTRL.START 位。</i>
1	Reserved	保留，必须保持复位值
0	BUSY	忙 该位指示闪存操作正在进行。在闪存操作开始时，该位被设置为'1'；在操作结束或发生错误时该位被清除为'0'。

2.2.3.2.5 FLASH 控制寄存器 (FLASH_CTRL)

偏移地址：0x10

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ECCERR ITE	EOPITE	Reserved	ERRITE	OPTWE	Reserved	LOCK	START	OPTER	OPTPG	Reserved	MER	PER	PG	
	rw	rw		rw	rw		rw	rw	rw	rw		rw	rw	rw	

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13	ECCERRITE	ECC 错误中断 该位允许在 FLASH_STS 寄存器中的 ECCERR 位变为'1'时产生中断。 0：禁止产生中断； 1：允许产生中断。
12	EOPITE	允许操作完成中断 该位允许在 FLASH_STS.EOP 位变为'1'时产生中断。 0：禁止产生中断

位域	名称	描述
		1: 允许产生中断
11	Reserved	保留, 必须保持复位值
10	ERRITE	允许错误状态中断 该位允许在发生 Flash 错误时产生中断 (当 FLASH_STS.PGERR/WRPERR 置为'1'时)。 0: 禁止产生中断 1: 允许产生中断
9	OPTWE	允许写选项字节 当该位为'1'时, 允许对选项字节进行编程操作。当在 FLASH_OPTKEY 寄存器写入正确的键序列后, 该位被置为'1'。 软件可清除此位。
8	Reserved	保留, 必须保持复位值
7	LOCK	锁定 只能写'1'。当该位为'1'时表示 Flash 和 FLASH_CTRL 被锁住。在检测到正确的解锁序列后, 硬件清除此位为'0'。 在一次不成功的解锁操作后, 下次系统复位前, 该位不能再被改变。
6	START	开始 当该位为'1'时将触发一次擦除操作。该位只可由软件置为'1'并在 FLASH_STS.BUSY 变为'1'时清除为'0'。
5	OPTER	擦除选项字节 0: 不开启选项字节擦除模式 1: 开启选项字节擦除模式
4	OPTPG	编程选项字节 0: 不开启选项字节编程模式 1: 开启选项字节编程模式
3	Reserved	保留, 必须保持复位值
2	MER	片擦除 0: 不开启片擦除模式 1: 开启片擦除模式
1	PER	页擦除 0: 不开启页擦除模式 1: 开启页擦除模式
0	PG	编程 0: 不开启编程模式 1: 开启编程模式

注: 关于编程及擦除请参考 2.2.1.4 节。

2.2.3.2.6 FLASH 地址寄存器 (FLASH_ADD)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FADD[31:16]															
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
FADD[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:0	FADD	闪存地址 当进行编程时选择要编程的地址，当进行页擦除时选择要擦除的页。 <i>注意：当FLASH_STS.BUSY 位为'1'时，不能写这个寄存器。</i>

2.2.3.2.7 FLASH 选项字节寄存器 (FLASH_OB)

偏移地址：0x1C

复位值：0x0FFF FFFC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDPRT2	Reserved				BOOT_SEL		Data1							Data0	
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0						Reserved		nSW_BOOT0	nBOOT1	nBOOT0	nRST_PD	nRST_STOP	IWDG_SW	RDPRT1	OBERR
r	r	r	r	r	r			r	r	r	r	r	r	r	r

位域	名称	描述
31	RDPRT2	读保护 L2 级别 0: 读保护 L2 级别未使能 1: 读保护 L2 级别使能 <i>注：只读位。</i>
30:28	Reserved	保留，必须保持复位值
27:26	BOOT_SEL	BOOT 串口通信引脚选择 11b:PA9,PA10, 默认 00b:PB10,PB11 01b:PD10,PD11 10b:PA2,PA3
25:18	Data1[7:0]	Data1 <i>注：只读位。</i>
17:10	Data0[7:0]	Data0 <i>注：只读位。</i>
9	Reserved	未使用，必须保持复位值

位域	名称	描述
8	BOOT0_CFG	BOOT0 上电默认状态 0: 默认高电平, 拉低有效 1: 默认低电平, 拉高有效
7	nSWBOOT0	使用规则见 2.1.3.2 启动配置章节。
6	nBOOT1	使用规则见 2.1.3.2 启动配置章节。
5	nBOOT0	使用规则见 2.1.3.2 启动配置章节。
4	nRST_PD	进入 Power Down 模式复位配置 0: 进入 Power Down 模式后立即产生复位, 即使执行了进入 Power Down 模式的过程, 系统将被复位而不是进入 Power Down 模式; 1: 进入 Power Down 模式后不产生复位。 <i>注: 该位为只读。</i>
3	nRST_STOP	进入 STOP 模式复位配置 0: 进入 STOP 模式后立即产生复位, 即使执行了进入停机模式的过程, 系统将被复位而不是进入停机模式; 1: 进入 STOP 模式后不产生复位。 <i>注: 该位为只读。</i>
2	IWDG_SW	看门狗设置 0: 硬件看门狗 1: 软件看门狗 <i>注: 只读位。</i>
1	RDPRT1	读保护 L1 级别 0: 读保护 L1 级别未使能 1: 读保护 L1 级别使能 <i>注: 只读位。</i>
0	OBERR	选项字节错误 当该位为'1'时表示选项字节和它的补码不匹配 <i>注: 只读位。</i>

2.2.3.2.8 FLASH 写保护寄存器 (FLASH_WRP)

偏移地址: 0x20

复位值: 0xFFFF FFFF

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

位域	名称	描述
31:0	WRPT	写保护

位域	名称	描述
		该寄存器包含由选项字节区加载的写保护选项字节。 0：写保护生效； 1：写保护失效。 注：只读位。

2.2.3.2.9 FLASH ECC 寄存器（FLASH_ECC）

偏移地址：0x24

复位值：0x0000 0000

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

位域	名称	描述
31:8	Reserved	硬件强制为 0。
7:0	ECCLW	32 位 Flash 地址对应的低 8-bit ECC 值。

2.2.3.2.10 FLASH USER 寄存器（FLASH_USER）

偏移地址：0x48

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								POR_DELAY[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVRcnt[7:0]								Reserved	LVR FILEN	LVRRST	LVREN	LVRLS[3:0]			
r	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	POR_DELAY	触发 POR 后，CPU 复位的延迟时间。 在系统初始上电完成后，Cortex®-M0 的系统解复位延时长可通过该位配置， 用来控制内核的复位延迟时间。 0x00：最大延迟时间

位域	名称	描述																																		
		0xFF: 无延迟 延迟时间 = (1/f _{LSI}) ×(0xFF - POR_DELAY)。																																		
15:8	LVRCNT	LVR 滤波控制计数值 0x00: 最大过滤宽度 0xFF: 不过滤 过滤宽度 = (1/f _{LSI}) ×(0xFF - LVRCNT)。																																		
7	Reserved	保留																																		
6	LVRFILEN	LVR 过滤器使能。 0: 已使能 LVR 过滤器 1: 未使能 LVR 过滤器																																		
5	LVRRST	LVR 复位使能控制。 0: 已使能 LVR 复位 1: 未使能 LVR 复位																																		
4	LVREN	LVR 使能。 0: 已使能 LVR 1: 未使能 LVR																																		
3:0	LVRLS	LVR 电压档位选择。 Value = 0x0F - LVRLS <table><tr><th>Value</th><th>Voltage</th></tr><tr><td>0000</td><td>保留</td></tr><tr><td>0001</td><td>2.0v</td></tr><tr><td>0010</td><td>2.2v</td></tr><tr><td>0011</td><td>2.4v</td></tr><tr><td>0100</td><td>2.6v</td></tr><tr><td>0101</td><td>2.8v</td></tr><tr><td>0110</td><td>3.0v</td></tr><tr><td>0111</td><td>3.2v</td></tr><tr><td>1000</td><td>3.4v</td></tr><tr><td>1001</td><td>3.6v</td></tr><tr><td>1010</td><td>3.8v</td></tr><tr><td>1011</td><td>4.0v</td></tr><tr><td>1100</td><td>4.2v</td></tr><tr><td>1101</td><td>4.4v</td></tr><tr><td>1110</td><td>4.6v</td></tr><tr><td>1111</td><td>4.8v</td></tr></table>	Value	Voltage	0000	保留	0001	2.0v	0010	2.2v	0011	2.4v	0100	2.6v	0101	2.8v	0110	3.0v	0111	3.2v	1000	3.4v	1001	3.6v	1010	3.8v	1011	4.0v	1100	4.2v	1101	4.4v	1110	4.6v	1111	4.8v
Value	Voltage																																			
0000	保留																																			
0001	2.0v																																			
0010	2.2v																																			
0011	2.4v																																			
0100	2.6v																																			
0101	2.8v																																			
0110	3.0v																																			
0111	3.2v																																			
1000	3.4v																																			
1001	3.6v																																			
1010	3.8v																																			
1011	4.0v																																			
1100	4.2v																																			
1101	4.4v																																			
1110	4.6v																																			
1111	4.8v																																			

2.2.3.2.11 FLASH 启动起始地址寄存器 (FLASH_START_ADD)

偏移地址: 0x4C

复位值: 0x0000 00FF

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

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

位域	名称	描述
31:8	Reserved	保留，必须保持复位值。
7:0	START_ADD	FLASH 启动起始地址：(值对应 1 页，递减；) 0xFF: 0x08000000,默认值 0xFE:0x08000200 0xFD:0x08000400 ... 0x01:0x801FC00 0x00:0x801FE00

2.2.3.2.12 FLASH VTOR 寄存器 (FLASH_VTOR)

偏移地址：0x50

复位值：0x0000 0000

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

位域	名称	描述
31	VTOR_EN	中断向量重映射使能 1: 使能 0: 禁用
30:0	VTOR_VALUE	用于中断向量重映射，存放中断向量表首地址 这些位在 VTOR_EN = 1 时有效。 中断地址 = VTOR_VALUE + 偏移地址。 注：偏移地址小于 0x100 时，该功能才有效。

3 电源控制（PWR）

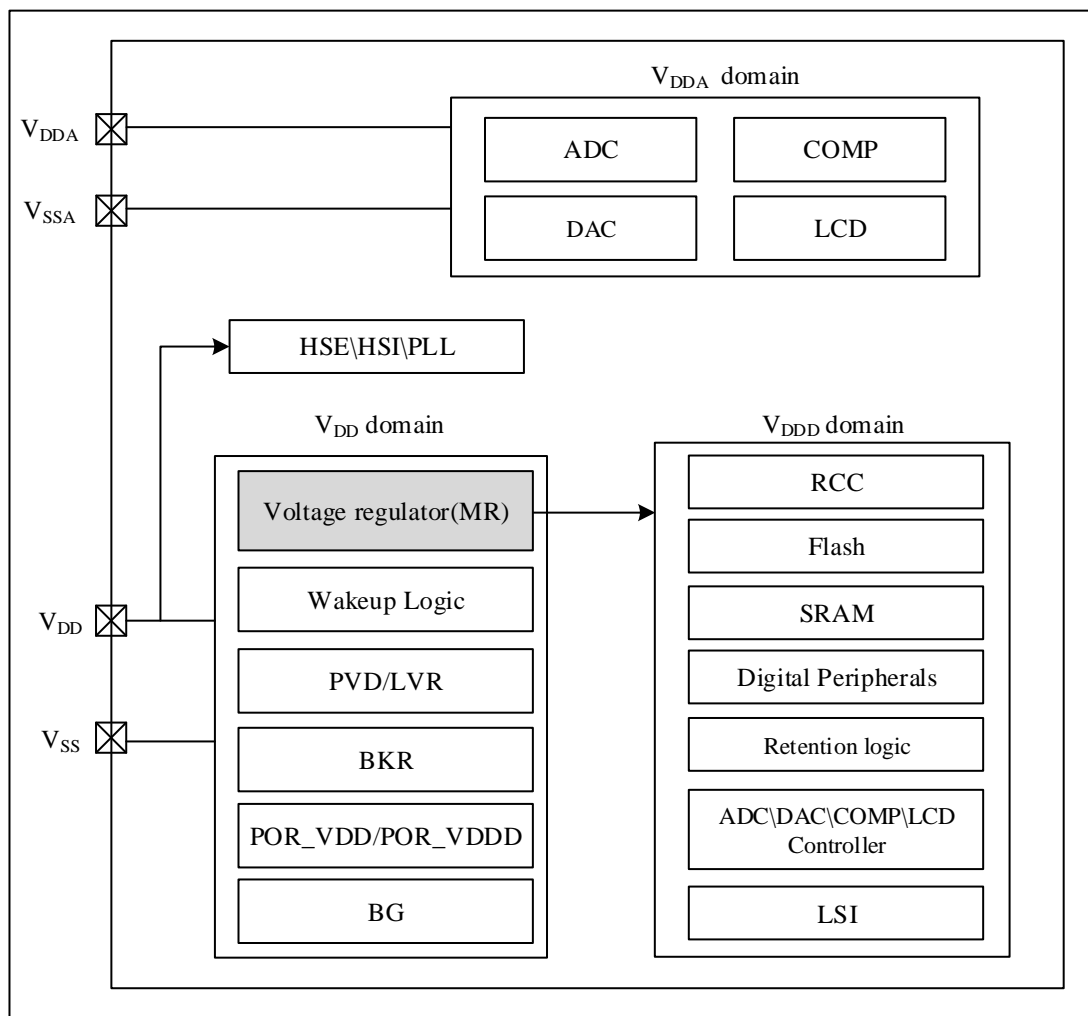
3.1 通用描述

PWR 是电源管理单元，它主要有两个电源域：模拟电源域及数字电源域。用于控制 MCU 进入不同的电源模式，并在事件或中断发生时唤醒。MCU 支持 RUN、SLEEP、STOP 和 PD 模式。PWR 控制不同电源模式下的电压调节器、时钟源、复位和 Flash/SRAM/GPIO 状态。

3.1.1 电源

- ◇ MCU 由外部 VDD 供电。嵌入式稳压器用于为内部 1.5V 数字电源供电。稳压器有两种模式，正常模式和低功耗模式。
 - VDD 区域：2.0V~5.5V，主要为 MR、IO 及时钟复位系统提供电源输入。
 - VDDA 区域：2.0V~5.5V，为大部分模拟外设供电，详细信息请参阅相关数据手册电气特性部分。
 - VDDA 和 VSSA 必须分别连接到 VDD 和 VSS。
- ◇ 电压调节器根据应用有几种不同的工作模式：
 - RUN 模式：电压调节器以正常电源模式供电。
 - SLEEP 模式：电压调节器以正常电源模式供电。
 - STOP 模式：电压调节器在低功耗模式下供电，输出电压可通过软件配置为 1.5V 或 1.2V。
 - PD 模式：电压调节器关闭。

图 3-1 电源框图

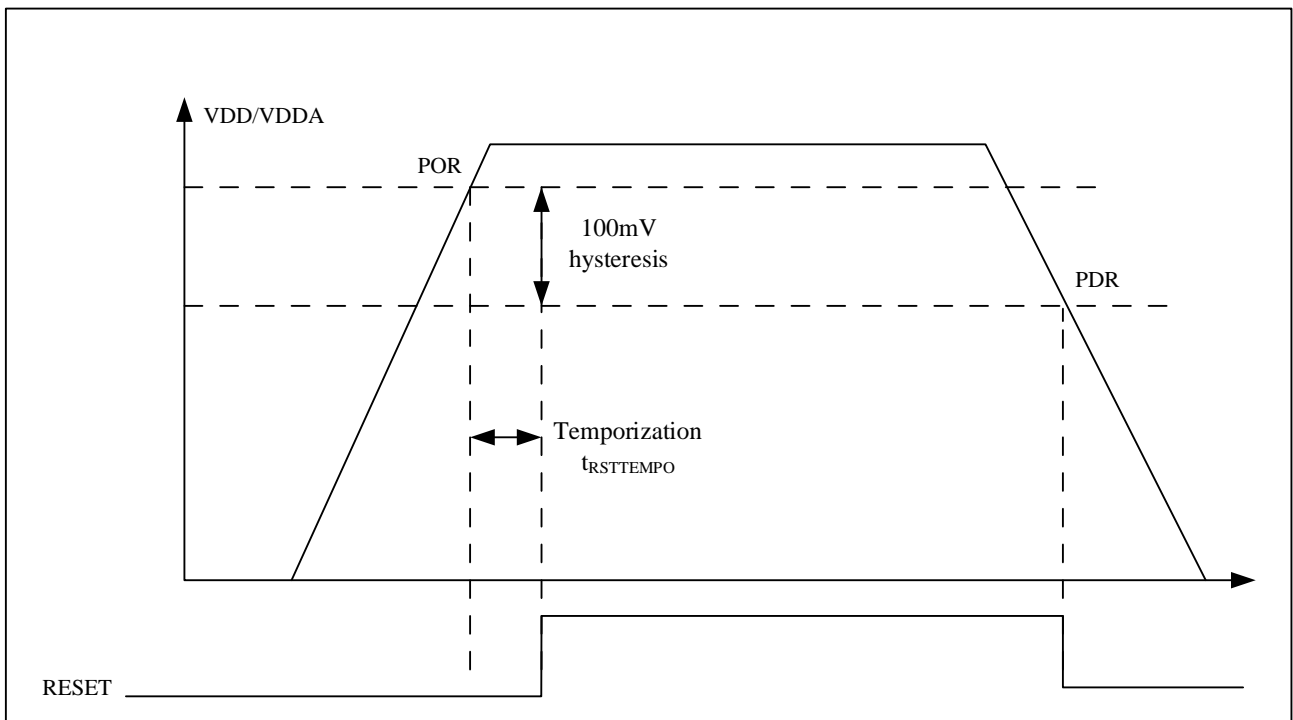


3.1.2 电压监控

3.1.2.1 上电复位（POR）和下电复位（PDR）

上电复位（POR）和下电复位（PDR）电路集成在芯片内部。当 VDD/VDDA 低于规定的限制电压 VPOR/VPDR 时，系统保持在复位状态，无需外部复位电路。有关上电和断电复位的详细信息，请参阅数据手册的电气特性部分。

图 3-2 上电复位和掉电复位的波形图

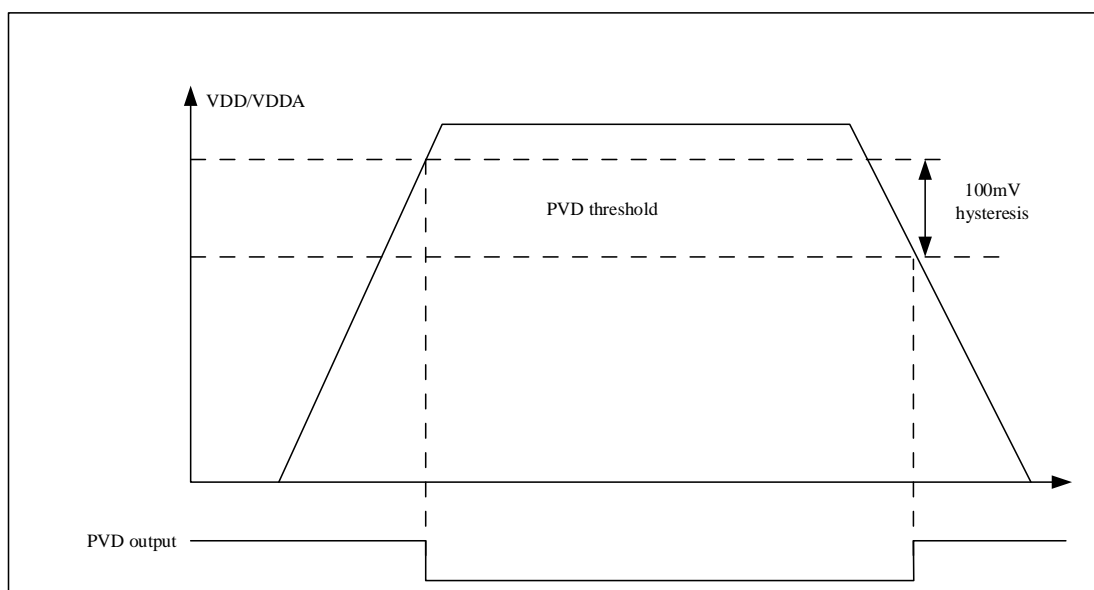


3.1.2.2 可编程电压监测器 (PVD)

PVD 通过比较 VDD 电压与 PWR_CTRL 寄存器中的相关位来产生中断。通过 PWR_CTRL.PLS[3:0] 选择监控电压的阈值，通过设置 PWR_CTRL.PVDEN 启用 PVD。

PWR_CTRLSTS.PVDO 标志用于指示 VDD 是否高于 PVD 电压阈值。该事件在内部连接到 EXTI 线 16，如果在外部中断寄存器中启用了中断，则会产生中断。根据 EXTI 线 16 的上升/下降沿触发设置，当 VDD 下降到 PVD 阈值以下和/或 VDD 上升到 PVD 阈值以上时，会发生 PVD 中断。此功能可用于执行紧急关断任务。

图 3-3 PVD 阈值图



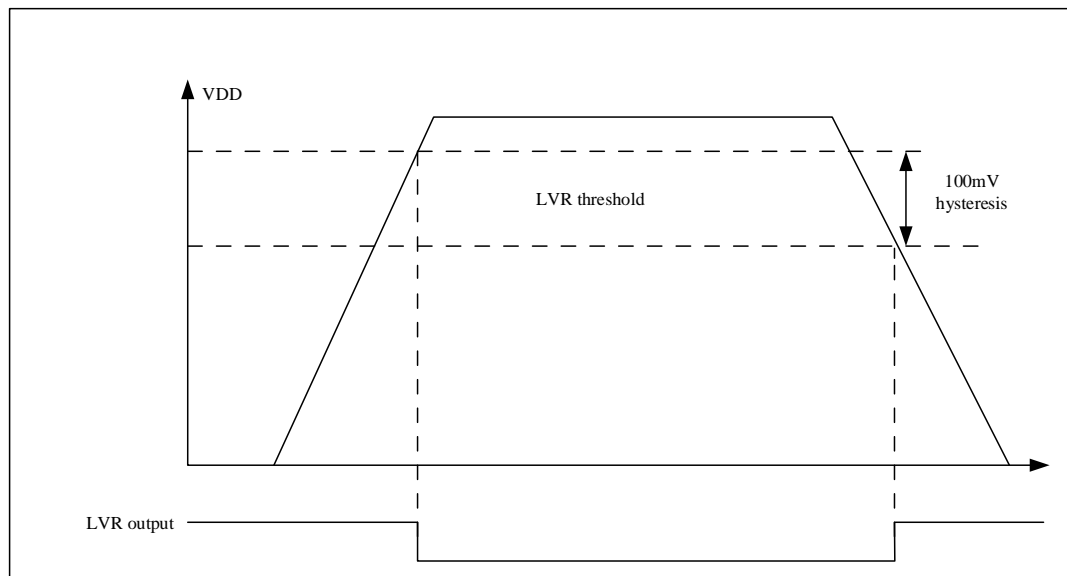
3.1.2.3 低电压复位（LVR）

LVR 通过比较 VDD 电压与 PWR_CTRL2 寄存器中的相关位来产生复位（PWR_CTRL2.LVRRSTEN 使能的情况下）。通过 PWR_CTRL2.LVRLS[3:0]选择复位电压的阈值，通过设置 PWR_CTRL2.LVREN 启用 LVR。

PWR_CTRL2.LVRO 位指示 VDD 是否高于或低于 LVR 阈值。

注：PWR_CTRL2.LVRKEY 需要写 A5，PWR_CTRL2 寄存器位[15:0]才能正常读写，否则默认为选项字节配置值。。

图 3-4 LVR 阈值图



3.2 低功耗模式

MCU 整体有四种电源模式：RUN、SLEEP、STOP 和 PD，不同的模式有不同的性能和功耗。MCU 功耗模式总结如下所示。

表 3-1 电源模式

模式	状态	进入	退出
RUN	CPU 运行。 所有外设运行可配置。	上电、系统复位或从其他低功耗模式唤醒	进入其他低功耗模式
SLEEP	CPU 处于睡眠模式。 所有的外设可配置，电压调节器处于正常模式。	WFI/WFE	任何中断唤醒事件。
STOP	CPU 处于深度睡眠模式。 外设时钟关闭，电压调节器处于低功耗模式。 HSE/HSI/PLL 关闭，LSI 可配置。Flash 进入深度待机模式。SRAM、寄存器保持，GPIO 状态保持。 唤醒后，HSI 自动开启，代码从停止的位置继续执行。	WFI/WFE: 1) SCB_SCR.SLEEPDEEP = 1, 没有挂起的中断/事件。 2) PWR_CTRL.PDSTP = 0	任何通过 EXTI(包含 GPIO、PVD、RTC、TIM6)、NRST、IWDG 的中断唤醒事件。
PD	CPU 处于深度睡眠模式。	WFI/WFE:	WKUP0/1 上升沿或下降沿，

模式	状态	进入	退出
	电压调节器关闭，所有时钟关闭。 大多数 IO 处于高阻状态。	1) SCB_SCR.SLEEPDEEP = 1，没有挂起的中断/事件。 2) PWR_CTRL.PDSTP = 1	NRST 复位

注意:

1. STOP 模式，在唤醒后，代码可以从停止位置继续运行。

不同模块在不同功耗模式下的运行使能情况如下表所示：

表 3-2 外设运行状态

Main Blocks	Run/Active	Sleep	Stop mode		Power down mode	
			Status	Wakeup capability	Status	Wakeup capability
Cortex-M0	Y	-	-	-	-	-
FLASH	O	O	O	-	-	-
SRAM	Y	Y	Y	-	-	-
POR/PDR	Y	Y	Y	Y	-	-
LVR	O	O	O	O	-	-
PVD	O	O	O	O	-	-
DMA	O	O	-	-	-	-
UART	O	O	-	-	-	-
I2C	O	O	-	-	-	-
SPI	O	O	-	-	-	-
LED	O	O	-	-	-	-
AD Timer	O	O	-	-	-	-
GP Timer	O	O	-	-	-	-
BS Timer	O	O	O	O	-	-
HSE	O	O	-	-	-	-
HSI	O	O	-	-	-	-
LSI	O	O	O	-	-	-
PLL	O	O	-	-	-	-
IWDG	O	O	O	O	-	-
WWDG	O	O	-	-	-	-
RTC	O	O	O	O	-	-
CAN	O	O	-	-	-	-
SAC	O	O	-	-	-	-
ADC	O	O	-	-	-	-
Temperature Sensor	O	O	-	-	-	-
DAC	O	O	-	-	-	-
COMP	O	O	-	-	-	-
LCD	O	O	-	-	-	-
SysTick	O	O	-	-	-	-

CRC	O	O	-	-	-	-
GPIOs	O	O	O	O	-	2 pins

注意:

1. Y: 是 (启用), O: 可选 (默认禁用, 软件启用), -: 无效。
2. 可以从 PD 唤醒的引脚有 PA0 (WKUP0)、PA2 (WKUP1)、NRST。

3.2.1 SLEEP 模式

CPU 停止, 包括 Cortex®-M0 内核周围的外设 (如 NVIC、SysTick 等) 在内的所有外设都可以运行并在发生中断或事件时唤醒 CPU。

3.2.1.1 进入 SLEEP 模式

通过在 SCB_SCR.SLEEPDEEP=0 时执行 WFI (等待中断) 或 WFE (等待事件) 指令进入 SLEEP 模式。根据 SCB_SCR.SLEEPONEXIT, 进入 SLEEP 模式有两个选项:

- SLEEP-NOW: SCB_SCR.SLEEPONEXIT=0, 立即执行 WFI 或 WFE 指令, 系统立即进入 SLEEP 模式。
- SLEEP-ON-EXIT: SCB_SCR.SLEEPONEXIT=1, 系统从最低优先级 ISR 退出后再进入 SLEEP 模式。

3.2.1.2 退出 SLEEP 模式

如果使用 WFI 指令进入 SLEEP 模式, 任何 NVIC 中断都可以将设备从 SLEEP 模式唤醒。

如果使用 WFE 指令进入 SLEEP 模式, MCU 将在事件发生时退出 SLEEP 模式。唤醒事件可以通过以下方式生成:

- 在外设控制寄存器中启用中断而不是 NVIC, 并启用 SCB_SCR.SEVONPEND。当 MCU 被 WFE 唤醒时, 外设中断挂起位和外设的 NVIC 中断通道挂起位 (在 NVIC 中断清除挂起寄存器中) 必须清零。
- 配置外部或内部 EXTI 事件模式。当 MCU 唤醒时, 不需要清除外设中断挂起位和外设的 NVIC 中断通道挂起位 (在 NVIC 中断清除挂起寄存器中), 因为没有设置事件线对应的挂起位。该模式提供最短的唤醒时间, 因为没有花时间在中断进入或退出上。

3.2.2 STOP 模式

STOP 模式基于 Cortex®-M0 深度睡眠模式与外设时钟控制相结合。电压调节器在低功耗模式下运行。输出电压可配置为 1.5V 或 1.2V。HSE/HSI/PLL 被禁用, LSI 可配置。所有 GPIO 状态、SRAM 和寄存器内容保持。FLASH 处于深度待机模式。唤醒后, 代码从挂起的地方继续执行。

3.2.2.1 进入 STOP 模式

进入 STOP 模式前, 用户需要设置 SCB_SCR.SLEEPDEEP=1 和 PWR_CTRL.PDSTP=0。

也可配置电压调节器低功耗模式输出电压, 以获取更低的电流消耗。STOP 模式下电压调节器输出电压默认为 1.5V, PDR 复位触发电压为 1.2V。当 PWR_CTRL5.STPMRSEL[1:0] = '01', 调节器输出电压为 1.5V, PWR_CTRL.PDRS[1:0] 必须配置为 '11' (PDR 触发电压 1.2V); 当 PWR_CTRL5.STPMRSEL[1:0] = '11', 调节器输出电压 1.2V, PWR_CTRL.PDRS[1:0] 必须配置为 '10' (PDR 触发电压 1.0V)。

如果正在进行 FLASH 操作, 则进入 STOP 模式的时间将延迟到存储器访问完成。

如果正在访问 APB 区域, 则进入 STOP 模式的时间将延迟到 APB 访问完成。

在 STOP 模式下，可以使用以下外设：

- 独立看门狗（IWDG）：一旦启动将一直工作，直到产生一个复位。
- RTC 可选：可以通过 RCC_LSCTRL.RTCEN 启用。
- TIM6/PVD 外设可配置唤醒。
- 内部 RC 振荡器（LSI RC）可选：可以通过 RCC_LSCTRL.LSIEN 开启。

进入 STOP 模式时应禁用 ADC，以避免不必要的功耗。

注意：如果应用程序需要在进入停止模式之前禁用外部时钟，则必须首先将系统时钟切换到 HSI，然后禁用 RCC_CTRL.HSEEN 位。否则，如果在进入停止模式时，RCC_CTRL.HSEEN 位保持使能，并且去掉外部时钟（外部振荡器），则必须启用时钟安全系统（CSS）功能，以检测任何外部振荡器故障，并避免进入停止模式时出现故障行为。

3.2.2.2 退出 STOP 模式

当中断或事件唤醒 STOP 模式时，选择 HSI RC 振荡器作为系统时钟，代码从挂起位置恢复。由于稳压器在 STOP 模式下处于低功耗模式，因此会消耗更多的启动时间。另外，用户可以在进入 STOP 前配置 PWR_CTRL4.FLASHWKUP = 1，以缩短 FLASH 的唤醒时间。

3.2.3 PD 模式

PD（Power Down）模式基于 Cortex®-M0 深度睡眠模式，可以实现更低的功耗。在此模式下，CPU、所有外设、电压调节器、HSE/HSI/PLL/LSI 时钟源和所有数字电源都关闭。除 NRST/PA0/PA2 外，大部分 IO 口输出高阻态。

3.2.3.1 进入 PD 模式

进入 PD 模式前，设置 SCB_SCR.SLEEPDEEP = 1 和 PWR_CTRL.PDSTP = 1。

如果正在对 FLAH 进行操作时，则进入 PD 模式的时间将被延迟，直到完成内存访问。

如果对 APB 区域的访问正在进行，则进入 PD 模式的时间将被延迟，直到 APB 访问完成。

3.2.3.2 退出 PD 模式

当外部复位（NRST 引脚）、WKUP 引脚上升沿或下降沿事件发生时，MCU 退出 PD 模式。所有寄存器在从 PD 状态唤醒后都将复位。

从 PD 模式中唤醒后，代码执行等同于复位后的执行（读取 Boot 引脚、读取复位向量等）。

3.3 Debug 模式

默认情况下，如果应用程序在使用调试特性时将 MCU 置于 SLEEP、STOP 或 PD 模式，则会丢失调试连接。这是由于 Cortex®-M0 内核失去了时钟。

但是，通过在 DBG_CTRL 寄存器中设置一些配置位，即使在使用 STOP 和 PD 模式时，也可以对软件进行调试。如果配置了这些寄存器位，电压调节器和 HSI 将不会被禁用或关闭。

3.3.1 低功耗模式调试支持

在低功耗模式下调试时，确保内核的 FCLK 开启，为内核调试提供必要的时钟。用户可以根据具体操作在

低功耗模式下调试 MCU（保证低功耗模式下 FCLK 的输出）。具体操作和功能请参考 3.4.9 章节对 DBG_CTRL.PD 和 DBG_CTRL.STOP 的描述。

3.3.2 外设调试支持

除了支持低功耗模式调试外，还支持部分外设调试状态下停止工作（TIM1、TIM2、TIM3、TIM4、TIM5、TIM6、I2C1、I2C2、CAN、IWDG、WWDG 等）。具体操作和特性请参考 3.4.9 章节对 DBG_CTRL 寄存器其他位域的描述。

3.4 PWR 寄存器

3.4.1 寄存器总览

表 3-3 PWR 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000h	PWR_CTRL	Reserved								PVDITEN	PVDIFILE	PVDTCNT[7:0]						Reserved				NRSTPOL	PDRS[1:0]		PLS[3:0]			PVDEN	CLRDBGPDF	CLRWDKUPF	PDSTP	IWDGRSTEN			
	Reset Value									0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	1	1	0	0	0	1
004h	PWR_CTRLSTS	Reserved																				WKUPPOL	Reserved	WKUPIEN	WKUPDEN	Reserved				PVDO	DBGPDF	WKUPF			
	Reset Value																					1		0						0	0	0	0	0	
008h	PWR_CTRL2	LVRKEY								Reserved						LVR0	Reserved	LVR1S	LVR1EN	LVR1STEN	LVR1FILE	LVR1CNT													
	Reset Value	0	0	0	0	0	0	0	0							0		0	0	0	1									0	0	0	0	0	0
014h	PWR_CTRL3	Reserved						NRSTFILEN	NRSTCNT										Reserved						PDRSELEN	Reserved	LSIEN	Reserved							
	Reset Value							0	0	0	0	0	0	0	0	0	0	0							0		0								
020h	PWR_CTRL4	Reserved																										RUNF	STBFLH	FLHWKUP					
	Reset Value																											0	0	0					
024h	PWR_CTRL5	Reserved																										STPMRSEL[1:0]		Reserved					
	Reset Value																											0	1						
028h	PWR_CTRL6	Reserved																										STPMREN[1:0]		Reserved					
	Reset Value																											0	0						

030h	DBG_CTRL	Reserved	TM2STP	TM4STP	TM6STP	TM5STP	I2C2TIMOUT	I2C1TIMOUT	CANSTP	Reserved	TM3STP	Reserved	TM1STP	WWDGSTP	IWDGSTP	Reserved	PD	STOP	SLEEP
	Reset Value		0	0	0	0	0	0	0		0		0	0	0		0	0	0

3.4.2 电源控制寄存器（PWR_CTRL）

偏移地址：0x00

复位值：0x0000 0621

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PVDITEN	PVDFIL EN	PVDCNT[7:0]							
						rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NRSTPOL	PDRS[1:0]		PLS[3:0]			PVDEN	CLRDBG PDF	CLRWKU PF	PDSTP	IWDGRST EN	
				rw	rw		rw			rw	w	w	rw	rw	

位域	名称	描述
31:26	Reserved	保留，必须保持复位值
25	PVDITEN	PVD 中断使能。 0：禁止 PVD 中断。 1：使能 PVD 中断。
24	PVDFILEN	PVD 过滤器使能。 0：禁止 PVD 过滤器。 1：使能 PVD 过滤器。
23:16	PVDCNT[7:0]	PVD 滤波控制计数值。 0x00：不过滤 0xFF：最大过滤宽度 过滤宽度 = $(1/f_{LSI}) \times PVDCNT$ 。
16:12	Reserved	保留，必须保持复位值。
11	NRSTPOL	NRST 极性选择。 0：NRST 引脚的下降沿触发复位。 1：NRST 引脚的上升沿触发复位。
10:9	PDRS[1:0]	STOP 模式下 V _{DDD} PDR 复位触发电平选择。 STOP 模式下电压小于设置的触发电平时，PDR 复位将会被触发。 配置该寄存器位前，软件必须先配置PWR_CTRL3.PDRSELEN = '1'。 00：保留。 01：保留。 10：V _{DDD} PDR 触发电平为 1.0V。 11：V _{DDD} PDR 触发电平为 1.2V。 只有 POR/PDR 可以复位该位。

位域	名称	描述																																		
8:5	PLS[3:0]	<p>PVD 等级选择。</p> <p>PVD 阈值控制如下：</p> <table><tr><th>PWR_CTRL.PLS</th><th>Voltage</th></tr><tr><td>0000</td><td>保留</td></tr><tr><td>0001</td><td>2.0V</td></tr><tr><td>0010</td><td>2.2V</td></tr><tr><td>0011</td><td>2.4V</td></tr><tr><td>0100</td><td>2.6V</td></tr><tr><td>0101</td><td>2.8V</td></tr><tr><td>0110</td><td>3.0V</td></tr><tr><td>0111</td><td>3.2V</td></tr><tr><td>1000</td><td>3.4V</td></tr><tr><td>1001</td><td>3.6V</td></tr><tr><td>1010</td><td>3.8V</td></tr><tr><td>1011</td><td>4.0V</td></tr><tr><td>1100</td><td>4.2V</td></tr><tr><td>1101</td><td>4.4V</td></tr><tr><td>1110</td><td>4.6V</td></tr><tr><td>1111</td><td>4.8V</td></tr></table>	PWR_CTRL.PLS	Voltage	0000	保留	0001	2.0V	0010	2.2V	0011	2.4V	0100	2.6V	0101	2.8V	0110	3.0V	0111	3.2V	1000	3.4V	1001	3.6V	1010	3.8V	1011	4.0V	1100	4.2V	1101	4.4V	1110	4.6V	1111	4.8V
PWR_CTRL.PLS	Voltage																																			
0000	保留																																			
0001	2.0V																																			
0010	2.2V																																			
0011	2.4V																																			
0100	2.6V																																			
0101	2.8V																																			
0110	3.0V																																			
0111	3.2V																																			
1000	3.4V																																			
1001	3.6V																																			
1010	3.8V																																			
1011	4.0V																																			
1100	4.2V																																			
1101	4.4V																																			
1110	4.6V																																			
1111	4.8V																																			
4	PVDEN	<p>PVD 使能控制.软件控制</p> <p>0: 禁止 PVD</p> <p>1: 使能 PVD</p>																																		
3	CLRDBGPDF	<p>清除 DBGPD 模式标志。</p> <p>始终读为 0.</p> <p>0: 无效</p> <p>1: 清除 PWR_CTRLSTS.DBGPDF 标志位.（写）</p>																																		
2	CLRWKUPF	<p>清除唤醒标志。</p> <p>始终读为 0.</p> <p>0: 无效</p> <p>1: 两个系统时钟周期后清除 PWR_CTRLSTS.WKUPF 标志位.（写）</p>																																		
1	PDSTP	<p>STOP/PD 模式选择。</p> <p>0: 当 CPU 输出 DEEPSLEEP 为'1'，芯片进入 STOP 模式；</p> <p>1: 当 CPU 输出 DEEPSLEEP 为'1'，芯片进入 PD 模式；</p>																																		
0	IWDGRSTEN	<p>IWDG 复位使能。</p> <p>0: IWDG 复位请求不会产生系统复位；</p> <p>1: IWDG 复位请求会产生系统复位。</p>																																		

3.4.3 电源控制状态寄存器 (PWR_CTRLSTS)

偏移地址: 0x04

复位值: 0x0000 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WKUP POL	Reserved	WKUP1 EN	WKUP0 EN	Reserved				PVDO	DBGPDF	WKUPF	
				rw		rw	rw					r	r	r	

位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
11	WKUPPOL	PA0/PA2 的唤醒极性。 使用上升沿或下降沿唤醒 PD 模式。确保在更改极性值之前禁用唤醒启用。 0: 下降沿 1: 上升沿
10	Reserved	保留，必须保持复位值。
9	WKUP1EN	WKUP 引脚 PA2 使能。 软件可以设置和清除该位。 0: WKUP 引脚用于通用 I/O。WKUP 引脚上的事件不会将器件从 PD 模式唤醒。 1: WKUP 引脚用于从 PD 模式唤醒。 <i>注：该位仅由 VDDD POR/PDR 复位来复位。</i>
8	WKUP0EN	WKUP 引脚 PA0 使能。 软件可以设置和清除该位。 0: WKUP 引脚用于通用 I/O。WKUP 引脚上的事件不会将器件从 PD 模式唤醒。 1: WKUP 引脚用于从 PD 模式唤醒。 <i>注：该位仅由 VDDD POR/PDR 复位来复位。</i>
7:3	Reserved	保留，必须保持复位值。
2	PVDO	PVD 输出。 硬件将设置和清除该位。仅当 PWR_CTRL.PVDEN = 1 时才有效。 0: VDD/VDDA 高于使用 PWR_CTRL.PLS[3:0]选择的 PVD 阈值 1: VDD/VDDA 低于使用 PWR_CTRL.PLS[3:0]选择的 PVD 阈值
1	DBGPDF	DBGPD 模式状态位。 进入 DBGPD 模式时，硬件将该位设为'1'; 软件往 PWR_CTRL.CLRDBGPDF 写'1'时，硬件将该位清零； 只有 VDDD POR/PDR 可以复位该位。 0: 芯片不曾进入 DBGPD 模式 1: 芯片曾进入 DBGPD 模式

位域	名称	描述
0	WKUPF	DBGPD 模式唤醒状态位。 该位在 WKUP 引脚唤醒 DBGPD 模式后由硬件设置; 软件往 PWR_CTRL.CLRWKUPF 写'1'时, 硬件将该位清零; 只有 VDDD POR/PDR 可以复位该位。 0: WKUP 管脚不曾发生唤醒事件。 1: WKUP 管脚发生了唤醒事件。

3.4.4 电源控制寄存器 2 (PWR_CTRL2)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LVRKEY								Reserved							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVRO	LVRLS				LVREN	LVRREST EN	LVRFILE EN	LVRCNT							
r	rw				rw	rw	rw	rw							

位域	名称	描述																						
31:24	LVRKEY	LVR key。 需要写 A5，CTRL2 寄存器位[15:0]才能正常读写																						
23:16	Reserved	保留，必须保持复位值。																						
15	LVRO	LVR 输出。 仅当 PWR_CTRL2.LVREN = 1 时才有效。 0：VDD 高于使用 PWR_CTRL2.LVRLS[3:0]选择的 LVR 阈值。 1：VDD 低于使用 PWR_CTRL2.LVRLS[3:0]选择的 LVR 阈值。																						
14:11	LVRLS	LVR 等级选择。 LVR 阈值控制如下： <table><tr><th>PWR_CTRL2.LVRLS</th><th>Voltage</th></tr><tr><td>0000</td><td>保留</td></tr><tr><td>0001</td><td>2.0V</td></tr><tr><td>0010</td><td>2.2V</td></tr><tr><td>0011</td><td>2.4V</td></tr><tr><td>0100</td><td>2.6V</td></tr><tr><td>0101</td><td>2.8V</td></tr><tr><td>0110</td><td>3.0V</td></tr><tr><td>0111</td><td>3.2V</td></tr><tr><td>1000</td><td>3.4V</td></tr><tr><td>1001</td><td>3.6V</td></tr></table>	PWR_CTRL2.LVRLS	Voltage	0000	保留	0001	2.0V	0010	2.2V	0011	2.4V	0100	2.6V	0101	2.8V	0110	3.0V	0111	3.2V	1000	3.4V	1001	3.6V
PWR_CTRL2.LVRLS	Voltage																							
0000	保留																							
0001	2.0V																							
0010	2.2V																							
0011	2.4V																							
0100	2.6V																							
0101	2.8V																							
0110	3.0V																							
0111	3.2V																							
1000	3.4V																							
1001	3.6V																							

位域	名称	描述	
		1010	3.8V
		1011	4.0V
		1100	4.2V
		1101	4.4V
		1110	4.6V
		1111	4.8V
10	LVREN	LVR 使能。 0：禁止 LVR。 1：使能 LVR。	
9	LVRSTEN	LVR 复位使能。 0：禁止 LVR 复位。 1：使能 LVR 复位。	
8	LVRFILEN	LVR 过滤器使能。 0：禁止 LVR 过滤器。 1：使能 LVR 过滤器。	
7:0	LVRCNT	LVR 滤波控制计数值。 0x00：不过滤 0xFF：最大过滤宽度 过滤宽度 = （1/f _{LSI} ）×LVRCNT。	

3.4.5 电源控制寄存器 3 (PWR_CTRL3)

偏移地址: 0x14

复位值: 0x0000 037F

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

位域	名称	描述
31:28	Reserved	保留, 必须保持复位值。
27	NRSTFILEN	NRST 滤波器使能 0: 不使能 1: 使能 <i>注意: STOP 模式下时钟关闭, 若开启 NRST 滤波会导致引脚复位失效, 因此需要在进 STOP 前关闭 NRST 滤波, 退出 STOP 后再开启 NRST 滤波。</i>
26:16	NRSTCNT	NRST 滤波控制计数值

位域	名称	描述
		0x000: 不过滤 0x7FF: 最大过滤宽度 过滤宽度 = $(1/f_{SYSCLK}) \times NRSTCNT$ 。
15:10	Reserved	保留, 必须保持复位值。
9	PDRSELEN	STOP 模式下 PDR 复位触发电平选择使能 0: PDR 复位触发电平固定为 1.2V。 1: PDR 复位触发电平可配, 配置位为 PWR_CTRL.PDRS[1:0]。
8	Reserved	保留, 必须保持复位值。
7	LSIEN	控制 PWR 使能 LSI。 0: 进入 STOP 模式后, 硬件自动关闭 LSI 时钟。 1: 进入 STOP 模式后, 硬件保持 LSI 时钟状态。
6:0	Reserved	保留, 必须保持复位值。

3.4.6 电源控制寄存器 4 (PWR_CTRL4)

偏移地址: 0x20

复位值: 0x0000 0000

该寄存器有写保护。软件每次对该寄存器进行写操作前, 必须先往该寄存器写入密钥 0x0175_3603 解锁。

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

位域	名称	描述
31:3	Reserved	保留, 必须保持复位值。
2	RUNF	RUN 模式标志。 0: 芯片不处于 RUN 模式 0: 芯片处于 RUN 模式
1	STBFLH	FLASH 深度待机模式使能 在 RUN 模式下, 该位由软件置位和清零, 也可以在 STOP 和 PD 模式下由硬件清零。 0: FLASH 保持正常模式 1: FLASH 进入深度待机模式
0	FLHWKUP	使能 Flash 快速唤醒 0: 芯片从 STOP 模式退出时, 使用 Flash 正常唤醒; 1: 芯片从 STOP 模式退出时, 使用 Flash 快速唤醒。

位域	名称	描述
		注意：唤醒时间请参考数据手册。

3.4.7 电源控制寄存器 5 (PWR_CTRL5)

偏移地址：0x24

复位值：0x0000 0004

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

rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3:2	STPMRSEL[1:0]	STOP 模式下电压调节器输出电压选择。 配置该寄存器位前，软件必须先配置PWR_CTRL6.STPMREN = '11'。 00：保留。 01：调节器输出电压为1.5V。 10：保留。 11：调节器输出电压为1.2V。 只有 POR/PDR 可以复位该位。
1:0	Reserved	保留，必须保持复位值。

3.4.8 电源控制寄存器 6 (PWR_CTRL6)

偏移地址：0x28

复位值：0x0000 0004

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

rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3:2	STPMREN[1:0]	STOP 模式下电压调节器输出电压选择使能。 00：调节器输出电压固定为1.5V。 01：保留。 10：保留。 11：调节器输出电压可配，配置位为PWR_CTRL5.SLPMRSEL。 只有 VDDD POR/PDR 可以复位该位。
1:0	Reserved	保留，必须保持复位值。

3.4.9 调试控制寄存器（DBG_CTRL）

偏移地址：0x30

复位值：0x0000 0000

只有 POR/PDR 可以复位该寄存器。只有连上仿真器后，软件才可以写访问该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TIM2STP	TIM4STP	TIM6STP	TIM5STP	I2C2 TIMOUT
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1 TIMOUT	CANSTP	Reserved	TIM3STP	Reserved	TIM1STP	WWDG STP	IWDG STP	Reserved					PD	STOP	SLEEP
rw	rw		rw		rw	rw	rw						rw	rw	rw

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20	TIM2STP	当内核进入调试状态时 TIM2 停止工作。 软件置 1 或清零。 0：选中定时器的计数器仍然正常工作； 1：选中定时器的计数器停止工作。
19	TIM4STP	当内核进入调试状态时 TIM4 停止工作。 软件置 1 或清零。 0：选中定时器的计数器仍然正常工作； 1：选中定时器的计数器停止工作。
18	TIM6STP	当内核进入调试状态时 TIM6 停止工作。 软件置 1 或清零。 0：选中定时器的计数器仍然正常工作； 1：选中定时器的计数器停止工作。
17	TIM5STP	当内核进入调试状态时 TIM5 停止工作。 软件置 1 或清零。 0：选中定时器的计数器仍然正常工作； 1：选中定时器的计数器停止工作。

位域	名称	描述
16	I2C2TIMOUT	当内核进入调试状态时 I2C2 SMBUS 超时定时器停止工作。 软件置 1 或清零。 0: SMBUS 超时定时器仍正常工作; 1: SMBUS 超时定时器停止工作。
15	I2C1TIMOUT	当内核进入调试状态时 I2C1 SMBUS 超时定时器停止工作。 软件置 1 或清零。 0: SMBUS 超时定时器仍正常工作; 1: SMBUS 超时定时器停止工作。
14	CANSTP	当内核进入调试状态时 CAN 停止工作。 软件置 1 或清零。 0: CAN 仍然正常工作; 1: CAN 停止工作。
13	Reserved	保留, 必须保持复位值。
12	TIM3STP	当内核进入调试状态时 TIM3 停止工作。 软件置 1 或清零。 0: 选中定时器的计数器仍然正常工作; 1: 选中定时器的计数器停止工作。
11	Reserved	保留, 必须保持复位值。
10	TIM1STP	当内核进入调试状态时 TIM1 停止工作。 软件置 1 或清零。 0: 选中定时器的计数器仍然正常工作; 1: 选中定时器的计数器停止工作。
9	WWDGSTP	当内核进入调试状态时调试窗口看门狗停止工作。 软件置 1 或清零。 0: 窗口看门狗计数器仍然正常工作; 1: 窗口看门狗计数器停止工作。
8	IWDGSTP	当内核进入调试状态时看门狗停止工作。 软件置 1 或清零。 0: 看门狗计数器仍然正常工作; 1: 看门狗计数器停止工作。
7:3	Reserved	保留, 必须保持复位值。
2	PD	调试 PD 模式控制。 软件置 1 或清零。 0: (FCLK 关, HCLK 关) 系统进入 PD 模式, 整个数字电路部分都断电。从软件的观点看, 退出 PD 模式与上电复位是一样的。 1: (FCLK 开, HCLK 开) 系统进入 PD 模式, 数字电路部分不下电, FCLK 时钟由内部 RC 振荡器提供时钟。另外, 微控制器通过产生系统复位来退出 DBGPD 模式, 和系统复位是一样的。

位域	名称	描述
1	STOP	<p>调试 STOP 模式。</p> <p>软件置 1 或清零。</p> <p>0: (FCLK 关, HCLK 关) 系统进入 STOP 模式, 时钟控制器禁止一切时钟 (包括 HCLK 和 FCLK)。当从 STOP 模式退出时, 时钟的配置和复位之后的配置一样 (微控制器由 8MHz 的内部 RC 振荡器 (HSI) 提供时钟)。因此, 软件必需重新配置时钟控制系统启动 PLL, 外部晶振等。</p> <p>1: (FCLK 开, HCLK 开) 系统进入 DBGSTOP 模式, FCLK 时钟由内部 RC 振荡器提供。当退出 DBGSTOP 模式时, 软件必需重新配置时钟系统启动 PLL, 外部晶振等 (与配置此比特位为 0 时的操作一样)。</p>
0	SLEEP	<p>调试 SLEEP 模式。</p> <p>软件置 1 或清零。</p> <p>0: (FCLK 开, HCLK 关) 在 SLEEP 模式时, FCLK 由原先已配置好的系统时钟提供, HCLK 则关闭。由于 SLEEP 模式不会复位已配置好的时钟系统, 因此从 SLEEP 模式退出时, 软件不需要重新配置时钟系统。</p> <p>1: (FCLK 开, HCLK 开) 在 DBG_LEEP 模式时, FCLK 和 HCLK 时钟都由原先配置好的系统时钟提供。</p>

4 复位和时钟控制(RCC)

4.1 复位控制单元

N32G05x 支持以下两种复位方式：

- 电源复位
- 系统复位

4.1.1 电源复位

当以下事件中之一发生时，产生电源复位：

- 上电/掉电复位（POR/PDR 复位）
- 从 PD 掉电模式中返回

电源复位将复位所有寄存器。（见图 3-1 电源框图）

复位源将最终作用于 NRST 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000_0004。更多细节，参阅表 6-1 向量表。

4.1.2 系统复位

除以下寄存器外，系统复位将复位所有寄存器至它们的复位状态。

- RCC_CTRL.HSEBP
- RCC_CTRL.HSITRIM
- RCC_LSCTRL.LSITRIM
- RCC_EMCCTRL
- RCC_CTRLSTS
- PWR_CTRL3.NRSTFILEN
- PWR_CTRL3.NRSTCNT
- PWR_CTRL.PDRS
- PWR_CTRL.NRSTPOL
- PWR_CTRLSTS.WKUPPOL
- PWR_CTRLSTS.WKUP1EN
- PWR_CTRLSTS.WKUP0EN
- PWR_CTRLSTS.DBGPDF
- PWR_CTRLSTS.WKUPF
- PWR_CTRL5.STBMRSEL

- PWR_CTRL6.STBMREN
- DBG_CTRL
- RTC_DATE
- RTC_TSH
- RTC_SUBS
- RTC_INITSTS

发生以下事件之一时会产生系统复位：

- NRST 引脚上的低电平（外部复位）
- 窗口看门狗计数终止（WWDG 复位）
- 独立看门狗计数终止（IWDG 复位）
- 软件复位（SW 复位）
- 低功耗管理复位
- MMU 保护复位
- LVR 复位
- RAM 奇偶校验出错复位
- EMC 复位
- M0 内核锁定复位

可以通过检查控制/状态寄存器(RCC_CTRLSTS)中的复位标志来识别复位源。

4.1.2.1 软件复位

可以通过设置 Cortex®-M0 应用中断和复位控制寄存器中的 SYSRESETREQ 位来产生软件复位。有关详细信息，请参阅 Cortex®-M0 技术参考手册。

4.1.2.2 低功耗管理复位

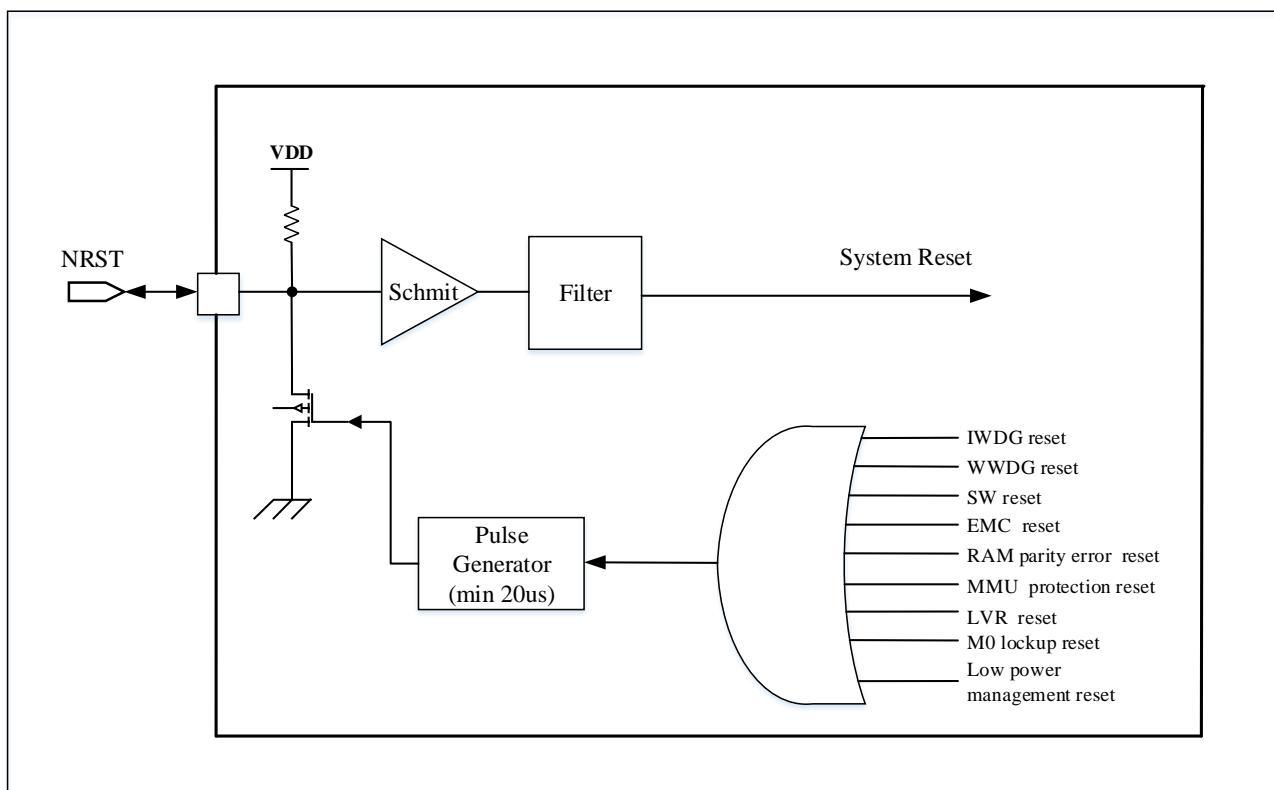
可以通过以下方式产生低功耗管理复位：

- 在进入 PD 模式时产生低功耗管理复位：通过将用户选择字节中的 nRST_PD 位置 1 将使能该复位。这时，即使执行了进入 PD 模式的过程，系统将被复位而不是进入 PD 模式。
- 在进入 STOP 模式时产生低功耗管理复位：通过将用户选择字节中的 nRST_STOP 位置 1 将使能该复位。这时，即使执行了进入 STOP 模式的过程，系统将被复位而不是进入 STOP 模式。

提供给芯片的系统复位信号会在 NRST 引脚上输出。脉冲发生器保证每个复位源（外部或内部）的复位脉冲至少持续时间 20μs。对于外部复位，当 NRST 引脚置为低电平时会产生复位脉冲。

下图展示了复位电路：

图 4-1 复位电路



4.2 时钟控制单元

可以使用四种不同的时钟源来驱动系统时钟(SYSCLK):

- HSI 振荡器时钟
- HSE 振荡器时钟
- PLL 时钟
- LSI 振荡器时钟

注意: 如无特殊说明, 文中提及的 HSI 均为 8M 高速内部时钟, HSI24M 时钟仅用于给 ADC 提供时钟源。

每个时钟源可以在不被使用时独立打开或关闭, 以此优化系统功耗。

多个预分频器可用于配置 AHB、高速 APB(APB2)和低速 APB(APB1)的频率。AHB 最大频率为 64MHz, APB2 最大频率为 64MHz, APB1 最大频率为 32MHz。

除去以下情况, 所有外设时钟都源于外设时钟(AHB、APB2、APB1):

- 通过配置 RCC_CFG2.ADCCLKSEL, 可选择下述情况之一作为 ADC 工作时钟源:
 - ◆ HSI 24M 时钟
 - ◆ PLL 时钟
 - ◆ AHB 时钟 (HCLK)
- 通过配置 RCC_CFG2.ADC1MSEL, 可选择下述情况之一作为 ADC1M 时钟源:

- ◆ HSI 时钟
- ◆ HSE 时钟
- 通过配置 RCC_CFG2.LCDCLKSEL，可选择下述情况之一作为 LCD 时钟源：
 - ◆ HSI 时钟 8 分频
 - ◆ HSE 时钟 16 分频
- 通过配置 RCC_CFG3.GCLKSEL，可选择下述情况之一作为 LED GCLK 时钟源：
 - ◆ HSI 时钟
 - ◆ HSE 时钟
- 通过配置 RCC_CFG2.TIM1CLKSEL，可选择下述情况之一作为 TIM1 工作时钟源：
 - ◆ APB2 时钟（PCLK2）
 - ◆ sysclk 时钟
- 通过配置 RCC_CFG2.TIM6CLKSEL，可选择下述情况之一作为 TIM6 工作时钟源：
 - ◆ APB1 时钟（PCLK1）
 - ◆ LSI 时钟

注意：当定时器时钟源为 PCLK 时，定时器时钟频率由硬件按以下 2 种情况自动设置：

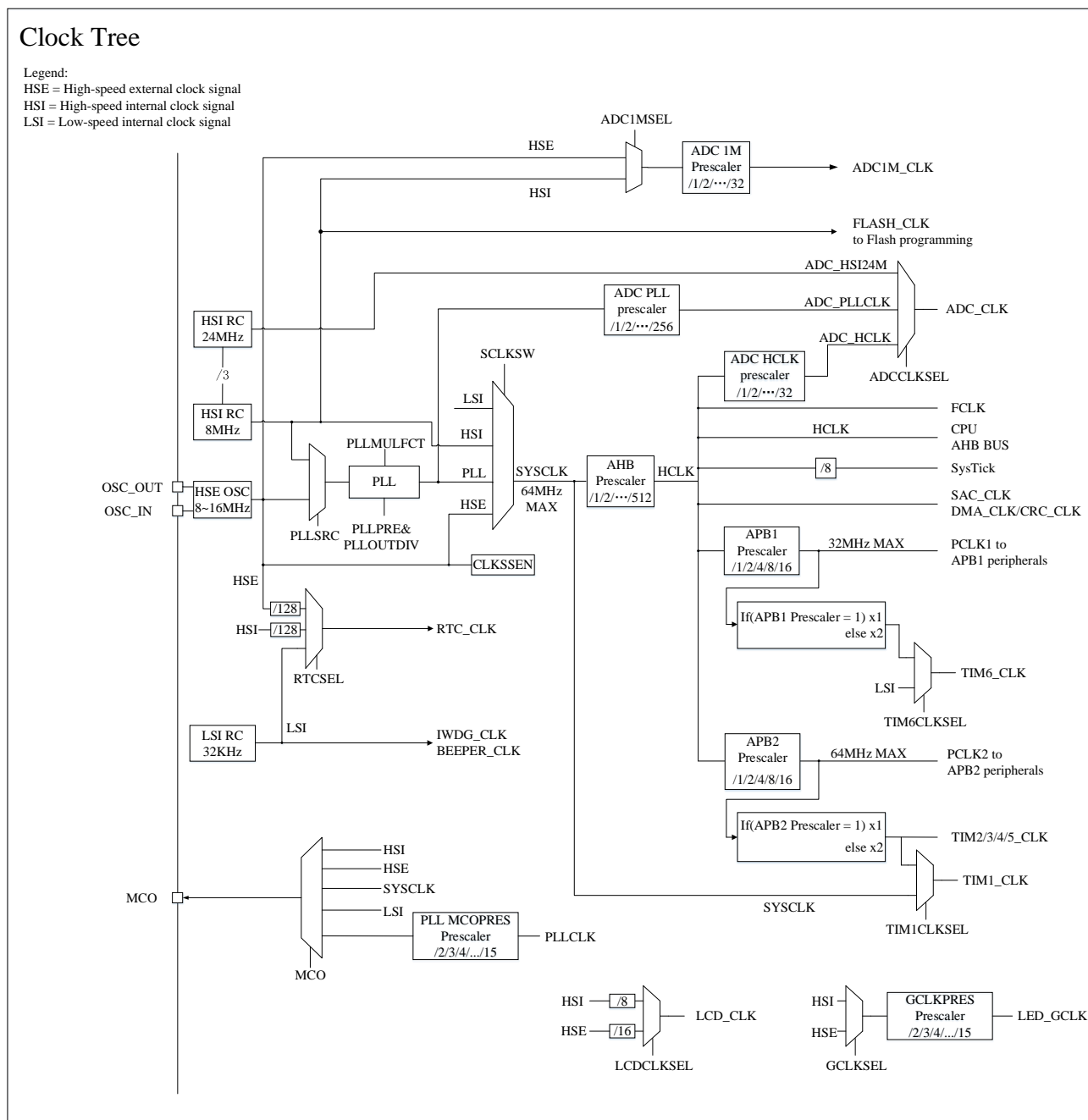
- ✧ 如果相应的 APB 时钟预分频系数是 1，定时器的时钟频率与 APB 总线频率相同
- ✧ 如果相应的 APB 时钟预分频系数不是 1，定时器的时钟频率是 APB 总线频率的 2 倍

- IWDG 和 beeper 的时钟源为 LSI 振荡器
- Flash 存储器编程接口时钟始终是 HSI 时钟
- 通过配置 SysTick 控制与状态寄存器，可选择下述两种情况之一作为 SysTick 时钟源：
 - ◆ AHB 时钟（HCLK）八分频
 - ◆ AHB 时钟（HCLK）

FCLK 是 Cortex®-M0 的自由运行时钟。详情见 ARM 的 Cortex®-M0 技术参考手册。。

4.2.1 时钟树

图 4-2 时钟树



1. 系统时钟的最大频率为 64MHz。
2. 有关内部和外部时钟源特性的详细信息，请参阅产品数据手册中的“电气特性”部分。

4.2.2 HSE 时钟

高速外部时钟信号（HSE）可以由以下两个时钟源产生：

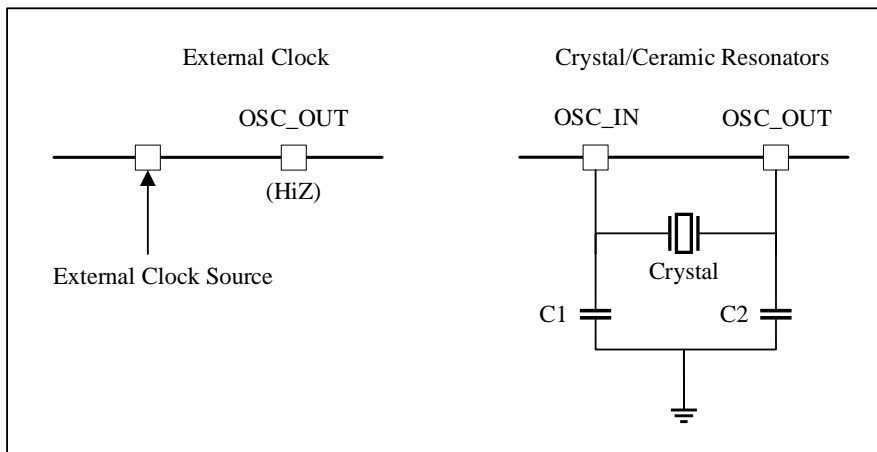
- HSE外部晶体/陶瓷谐振器

● HSE用户外部时钟(通过PD14管脚输入)

在 HSE 旁路模式和晶体模式下, `RCC_CTRL.HSEEN` 需要设置为 1, 如果 `RCC_CTRL.HSEEN=0` 会关闭 HSE。

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

图 4-3 HSE 时钟源



4.2.2.1 外部时钟源(HSE 旁路模式)

在这种模式下, 用户必须提供外部时钟源。它的频率最高可达 16MHz。用户可以通过设置 `RCC_CTRL.HSEBP` 和 `RCC_CTRL.HSEEN` 位来选择该模式。当 PD14 用作外部时钟信号(方波、正弦波或占空比为 50%的三角波)时, 必须连接到 `OSC_IN` 引脚, 而 `OSC_OUT` 引脚必须悬空(Hi-Z)。见图 4-3。

4.2.2.2 晶体/陶瓷谐振器(HSE 晶体模式)

8~16MHz 外部振荡器具有为系统产生更准确的主时钟的优势。相关的硬件配置如图 4-3 所示。更多详细信息, 请参阅数据手册的电气特性部分。

`RCC_CTRL.HSERDF` 位指示高速外部振荡器是否稳定。在启动时, 直到该位被硬件设置, 时钟才会被释放。如果在时钟中断寄存器(`RCC_CLKINT`)中使能对应位, 则可以产生中断。

通过设置 `RCC_CTRL.HSEEN` 位可以打开和关闭 HSE 时钟。

如用户需要在运行过程中改变 `RCC_CTRL.HSEBP` 的配置, 则应在使能 `RCC_CTRL.HSEEN` 之前配置。

4.2.3 HSI 时钟

HSI (高速内部) 时钟信号由内部 8MHz RC 振荡器产生, 可直接作为系统时钟或 PLL 输入。HSI RC 振荡器无需任何外部设备即可提供时钟源。它启动时间比 HSE 晶体振荡器更短。然而, 即使经过校准它的频率精度仍较差。

制造工艺决定了不同芯片的 RC 振荡器频率会不同, 这就是为什么每个芯片的 HSI 时钟频率在出厂前已经被校准到 1% (25°C) 的原因。

由于用户的应用场景会受到电压或温度变化的影响, 这也会影响 RC 振荡器的频率精度。用户可以使用 `RCC_CTRL.HSITRIM[4:0]` 位调整 HSI 频率。

`RCC_CTRL.HSIRDF` 位指示 HSI RC 振荡器是否稳定。在启动时, 直到该位被硬件设置, HSI RC 输出时钟

才会被释放。可以通过设置 `RCC_CTRL.HSIEN` 位打开和关闭 HSI 时钟。

如果 HSE 晶振出现故障，HSI 时钟可以作为备用源。请参阅 4.2.7 时钟安全系统(CLKSS)。

4.2.4 PLL 时钟

内部 PLL 可用于倍频 HSI RC 输出时钟或 HSE 晶体输出时钟。参考图 4-4。PLL 的设置（选择 HSI 或 HSE 作为 PLL 的输入时钟，选择乘法器，选择预分频器和输出分频器）必须在激活之前完成。一旦 PLL 被激活，这些参数就无法更改。可以使用 `RCC_CTRL` 和 `RCC_CFG` 寄存器中的控制位来配置 PLL。

在切换 PLL 的输入时钟源时（通过时钟配置寄存器位 `RCC_CFG.PLLSRC`），必须新的时钟源就绪后再关闭原来的时钟源。

如果在时钟中断寄存器中使能了 PLL 中断，则可以在 PLL 就绪时产生中断请求。

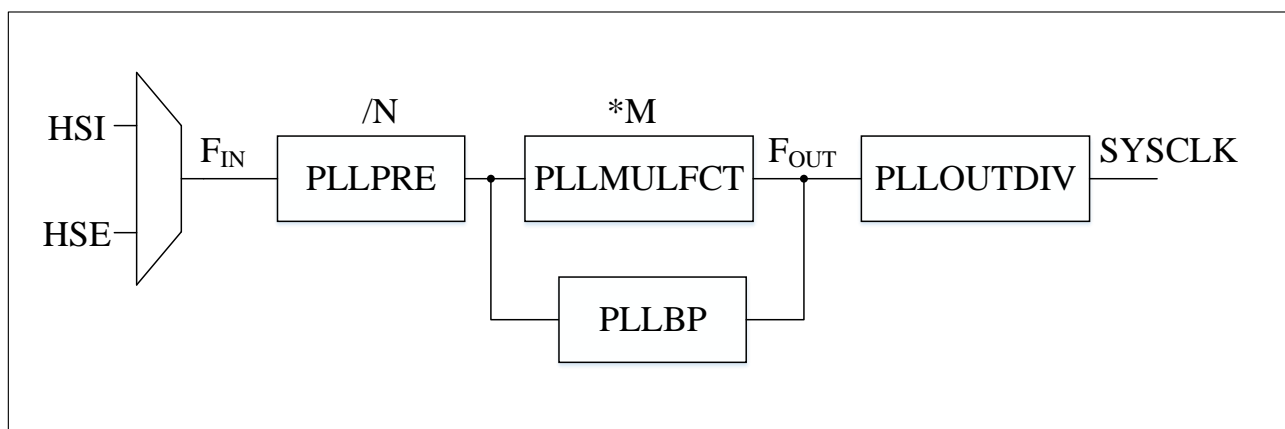
输入频率为 F_{IN} ， F_{IN} 要求在 4MHz~16MHz 范围内， F_{IN}/N 要求在 4MHz~16MHz 范围内。

不使能 PLL 旁路时：输出频率为 $F_{OUT} = F_{IN} * M/N$ ， F_{OUT} 要求在 48MHz~72MHz 范围内；使能 PLL 旁路时： $F_{OUT} = F_{IN}/N$ ， F_{OUT} 要求在 4MHz~16MHz 范围内。

系统频率由 F_{OUT} 除以分频因子后得出，需要正确配置软件以避免 `SYSCCLK` 超过 64MHz。

此外，用户还可以配置 `RCC_CTRL.PLLBP` 来绕过 PLL 的倍频功能。

图 4-4 PLL 时钟树



4.2.5 LSI 时钟

LSIRC 可以在 STOP 模式下为 IWDG 和 AWU 提供时钟。LSI 时钟频率约为 32KHz。有关详细信息，请参阅数据表的电气特性部分。

可以使用 `RCC_CTRLSTS.LSIEN` 位打开或关闭 LSI 时钟，`RCC_CTRLSTS.LSIRD` 位标志指示 LSI 时钟是否稳定。在启动时，时钟不会被释放，直到该位被硬件设置。如果在时钟中断寄存器 (`RCC_CLKINT`) 中使能，则可以产生中断。

4.2.5.1 LSI 校准

可以通过校准补偿内部低速振荡器 LSI 频率误差，以获得更高精度的 RTC（由 LSI 提供时钟时）、Beeper 和 IWDG 时基。

LSI 校准步骤如下：

1. 打开 TIM2，设置通道 3 为输入捕捉模式；
2. 设置 TIM2_CTRL1.C3SEL 位为 1，将 LSI 内部连接到 TIM2 的通道 3；
3. 通过 TIM2 捕获/比较 3 事件或中断测量 LSI 时钟频率；
4. 根据计算出的 LSI 实际频率与 32kHz 的偏差，调节 RCC_LSICTRL.LSITRIM[4:0]（默认值 16，调节步长 720Hz）

4.2.6 系统时钟(SYSCLK)选择

系统复位后，选择 HSI 振荡器作为系统时钟。当时钟源直接或通过 PLL 间接用作系统时钟时，HSI 无法被停止。

仅当目标时钟源准备好（在启动延迟或 PLL 锁定之后）时，才能从一个时钟源切换到另一个时钟源。当所选时钟源未准备好时，不会发生系统时钟的切换。

RCC_CFG.SCLKSW[1:0]用于选择系统时钟源。RCC_CTRL 和 RCC_LSICTRL 中的状态位指示哪个时钟已准备就绪，RCC_CFG 指示当前使用哪个时钟作为系统时钟。

4.2.7 时钟安全系统(CLKSS)

时钟安全系统可以通过软件通过设置 RCC_CTRL.CLKSSEN 位来激活。一旦被激活，时钟检测器在 HSE 振荡器的启动延时后被启用，并在 HSE 时钟关闭时被禁用。

如果 HSE 时钟出现故障，HSE 振荡器将自动关闭，时钟失效事件将发送到高级定时器（TIM1）的刹车输入，并产生时钟安全系统中断 CLKSSIF，允许软件执行营救措施。CLKSSIF 中断连接到 Cortex®-M0 的 NMI（不可屏蔽）中断。

一旦 CSS 被激活并且 HSE 时钟出现故障，就会产生 CSS 中断并自动产生 NMI。NMI 将连续执行，直到 CSS 中断挂起位被清除。因此，需要通过在 NMI 处理程序中设置 RCC_CLKINT.CLKSSICLR 位来清除 CSS 中断。

如果 HSE 振荡器直接或间接用作系统时钟（间接的意思是：HSE 用作 PLL 输入时钟，PLL 时钟用作系统时钟），时钟失效会导致系统时钟切换到 HSI 振荡器，并且外部 HSE 振荡器被禁用。如果选择 HSE 时钟（分频或不分频）作为 PLL 输入时钟，那么当 HSE 时钟故障时，PLL 将被关闭。

4.2.8 RTC 时钟

通过设置低速时钟控制寄存器（RCC_LSICTRL）里的 RTCSEL[1:0]位，RTCCLK 时钟可以由 HSE/128、HSI/128 或 LSI 时钟提供。

4.2.9 看门狗时钟

如果 IWDG 由硬件选项或软件启动，LSI 振荡器将被强制开启并且不能被禁用。LSI 振荡器稳定后，时钟被提供给 IWDG。

4.2.10 时钟输出(MCO)

微控制器时钟输出(MCO)功能允许将时钟信号输出到外部 MCO 引脚。

对应的 GPIO 口寄存器必须配置为对应的功能。可以选择以下 5 个时钟信号作为 MCO 时钟：

- SYSCLK
- HSI
- HSE
- LSI
- PLL 时钟分频

时钟选择由 RCC_CFG.MCO[2:0]位控制。

4.3 RCC 寄存器

RCC 寄存器可通过 AHB 总线访问，寄存器说明如下。

4.3.1 寄存器总览

表 4-1 RCC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000h	RCC_CTRL	Reserved						LKUPRSTEN	PLLRDF	PLLEN	PLLOUTEN	PLLBP	HSERDTM[1:0]		CLKSEN	HSEBP	HSERDF	HSEEN	Reserved						HSITRIM[3:0]				Reserved	HSI24MRDF	HSIRDF	HSIEN			
	Reset Value							0	0	0	1	0	0	0	0	0	0	0												0	0	0	0	1	0
004h	RCC_CFG	MCOPRES[3:0]				MCO[2:0]				PLLSRC	PLLOUTDIV[1:0]		PLLPRE[1:0]		PLLMULFCT[3:0]			SCLKSTS2[1:0]		APB2PRES[2:0]			APB1PRES[2:0]			AHBPRES[3:0]				SCLKSTS	SCLKSW[2:0]				
	Reset Value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
008h	RCC_CLKINT	Reserved									CLKSSICLR	Reserved	RAMPERRCLR	PLLRDICLR	HSERDICLR	HSIRDICLR	Reserved	LSIRDICLR	Reserved	RAMPERRRSTEN	RAMPERRIEN	PLLRDIEN	HSERDIEN	HSIRDIEN	Reserved	LSIRDIEN	CLKSSIF	Reserved	RAMPERRIF	PLLRDIF	HSERDIF	HSIRDIF	Reserved	LSIRDIF	
	Reset Value										0		0	0	0	0		0		0	0	0	0	0		0	0		0	0	0	0		0	0
00Ch	RCC_APB2PRST	Reserved														TIM5RST	TIM4RST	TIM3RST	UART2RST	UART1RST	TIM2RST	TIM1RST	SPI3RST	SPI2RST	SPI1RST	Reserved				IOPDRST	IOPCRST	IOPBRST	IOPARST	Reserved	AFORST
	Reset Value															0	0	0	0	0	0	0	0	0	0					0	0	0	0		0
010h	RCC_APB1PRST	Reserved			PWRRST	DACRST	CANRST	Reserved				I2C2RST	I2C1RST	UART5RST	UART4RST	UART3RST	Reserved						WWDGRST	LCDRST	Reserved	COMPRST	Reserved		BEEPRST	TIM6RST	Reserved				
	Reset Value				0	0	0					0	0	0	0	0							0	0		0			0	0				0	0
014h	RCC_AHBCLKEN	Reserved														ADCEN		SACEN	Reserved				CRCE		Reserved	FLITEN	Reserved	SRAMEN	Reserved	DMAEN					
	Reset Value															0	0	0					1	1		0									
018h	RCC_APB2PCLKEN	Reserved														TIM5EN	TIM4EN	TIM3EN	UART2EN	UART1EN	TIM2EN	TIM1EN	SPI3EN	SPI2EN	SPI1EN	Reserved				IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFOEN
	Reset Value															0	0	0	0	0	0	0	0	0	0					0	0	0	0		0
01Ch	RCC_APB1PCLKEN	Reserved			PWREN	DACEN	CANEN	Reserved				I2C2EN	I2C1EN	UART5EN	UART4EN	UART3EN	Reserved						WWDGEN	LCDEN	COMPFLTEN	COMPEN	Reserved		BEEPEN	TIM6EN	Reserved				
	Reset Value				1	0	0					0	0	0	0	0							0	0	0	0			0	0				0	0
020h	RCC_LSCTRL	Reserved														RTCST	RTCEN	RTCSEL[1:0]		Reserved				LSITRIM[4:0]				LSIRDIF	LSIEN						
	Reset Value															0	0												0	0	1	0	0	0	0
024h	RCC_CTRLSTS	Reserved														LKUPRSTF	EMCGBRSTF	EMCGBNRSTF	LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINKSTF	MMURSTF	RAMPERRSTF	RMRSTF								

	Reset Value															0	0	0	0	0	0	0	1	1	0	0	0				
028h	RCC_AHBPRST	Reserved												ADCRST	SACRST	Reserved															
	Reset Value													0	0																
02Ch	RCC_CFG2	TIM1CLKSEL	TIM6CLKSEL	Reserved	LCDCCLKSEL	ADCCLKSEL[1:0]		Reserved						ADC1MPRE[4:0]				ADC1MSEL	Reserved	ADCPLLPRES[4:0]					ADCHPRES [3:0]						
	Reset Value	0	0		0	0	0							0	0	0	1	1		1	0	0	0	0	0	0	0	0	0	0	
030h	RCC_CFG3	GCLKNNUM[7:0]							GCLKMNUM[7:0]							Reserved				GCLKPRES[3:0]				Reserved				GCLKDMAEN	GCLKDONE	GCLKSEL	GCLKEN
	Reset Value	0	1	1	1	1	0	0	0	0	0	0	0	1	0					1	0	0	0					0	0	0	0
034h	RCC_EMCCTRL	Reserved						GBRST3	GBRST2	GBRST1	GBRST0	GBNRST3	GBNRST2	GBNRST1	GBNRST0	Reserved				GBDET3	GBDET2	GBDET1	GBDET0	GBNDET3	GBNDET2	GBNDET1	GBNDET0	Reserved			
	Reset Value							0	0	0	0	0	0	0	0					0	0	0	0	0	0						

4.3.2 时钟控制寄存器(RCC_CTRL)

偏移地址：0x00

复位值：0x0080 0087

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					LKUP RSTEN	PLLRDF	PLLEN	PLLOUT EN	PLLBP	HSERDTM[1:0]		CLKSS EN	HSEBP	HSERDF	HSEEN
					rw	r	rw	rw	rw	rw		rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HSITRIM[3:0]				Reserved	HSI24M RDF	HSIRDF	HSIEN
								rw					r	r	rw

位域	名称	描述
31:27	Reserved	保留，必须保持复位值
26	LKUPRSTEN	M0 内核锁定复位使能 0: M0 内核锁定时不产生复位 1: M0 内核锁定时产生复位
25	PLLRDF	PLL 时钟就绪标志位 PLL 时钟就绪后由硬件置位。 0:PLL 未就绪 1:PLL 已就绪
24	PLLEN	PLL 使能位 由软件置位和清零。进入 STOP 或 PD 模式时，由硬件清零。当 PLL 用作系统时钟时，该位不能清零。 0: 禁用 PLL 1: 使能 PLL
23	PLLOUTEN	PLL 时钟输出使能位。 0:锁相环时钟输出被禁用 1:锁相环时钟输出已启用
22	PLLBP	PLL 旁路模式 0: $F_{out} = F_{in} * M / N$ 1: $F_{out} = F_{in} / N$
21:20	HSERDTM[1:0]	外部高速时钟就绪延时时间 00: HSE 就绪延时 0.5ms 01: HSE 就绪延时 1.0ms 10: HSE 就绪延时 1.5ms 11: HSE 就绪延时 2.5ms
19	CLKSSEN	时钟安全系统使能位 由软件置位和清零。 0: 禁用时钟检测器 1: 如果 HSE 振荡器就绪，则使能时钟检测器

位域	名称	描述
18	HSEBP	外部高速时钟旁路使能位 由软件置位和清零。该位只能在 HSE 振荡器被禁止时写入。 0: 禁止 HSE 振荡器的旁路功能 1: 使能 HSE 振荡器的旁路功能 <i>注: 使能此位需和 RCC_CTRL.HSEEN 位一起使能</i>
17	HSERDF	外部高速时钟就绪标志位 HSE 就绪后由硬件置位。该位在 HSEEN 位清零后 6 个 HSE 时钟周期来清零。 0: HSE 未就绪 1: HSE 就绪
16	HSEEN	外部高速时钟使能位 由软件置位和清零。进入 STOP 或 PD 模式时, 由硬件清零。当 HSE 直接或间接用作系统时钟时, 该位不能被清零。 0: 禁止 HSE 振荡器 1: 使能 HSE 振荡器
15:8	Reserved	保留, 必须保持复位值
7:4	HSITRIM[3:0]	内部高速时钟校准值 由软件写入, 用以校准内部 HSI RC 振荡器的频率。 默认值为 8, 调节步长为 24KHz。 设置值(取整) = 当前值 - (HSI 实测频率 - 目标频率 8000K) / (步长 24K);
3	Reserved	保留, 必须保持复位值
2	HSI24MRDF	内部 24M 高速时钟就绪标志位 24MHSI 就绪后由硬件置位, 仅用于给 ADC 提供时钟。HSIEN 位清零后, 该位需要 6 个内部 24MHz 振荡器时钟周期才能清零。 0:HSI 未就绪 1:HSI 就绪
1	HSIRDF	内部 8M 高速时钟就绪标志位 8MHSI 就绪后由硬件置位。HSIEN 位清零后, 该位需要 6 个内部 8MHz 振荡器时钟周期才能清零。 0:HSI 未就绪 1:HSI 就绪
0	HSIEN	内部高速时钟使能 由软件置位和清零。当从 STOP 或 PD 模式返回或发生 HSE 故障时, 由硬件置位以启用 HSI 振荡器。如果 HSI 直接或间接用作系统时钟, 则该位不能复位。 0: 禁止 HSI 振荡器 1: 使能 HSI 振荡器

4.3.3 时钟配置寄存器(RCC_CFG)

偏移地址: 0x04

复位值: 0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCOPRES[3:0]				MCO[2:0]			PLLSRC	PLLOUTDIV[1:0]	PLLPRE[1:0]	PLLMULFCT[3:0]					
rw				rw			rw	rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLKSTS2[1:0]			APB2PRES[2:0]			APB1PRES[2:0]			AHBPRES[3:0]			SCLKSTS	SCLKSW[2:0]		
r			rw			rw			rw			r	rw		

位域	名称	描述
31:28	MCOPRES[3:0]	<p>MCO 预分频。</p> <p>软件设置或清零。</p> <p>0010: 由 PLL 时钟 2 分频作为 MCO 时钟</p> <p>0011: 由 PLL 时钟 3 分频作为 MCO 时钟</p> <p>0100: 由 PLL 时钟 4 分频作为 MCO 时钟</p> <p>0101: 由 PLL 时钟 5 分频作为 MCO 时钟</p> <p>0110: 由 PLL 时钟 6 分频作为 MCO 时钟</p> <p>0111: 由 PLL 时钟 7 分频作为 MCO 时钟</p> <p>1000: 由 PLL 时钟 8 分频作为 MCO 时钟</p> <p>1001: 由 PLL 时钟 9 分频作为 MCO 时钟</p> <p>1010: 由 PLL 时钟 10 分频作为 MCO 时钟</p> <p>1011: 由 PLL 时钟 11 分频作为 MCO 时钟</p> <p>1100: 由 PLL 时钟 12 分频作为 MCO 时钟</p> <p>1101: 由 PLL 时钟 13 分频作为 MCO 时钟</p> <p>1110: 由 PLL 时钟 14 分频作为 MCO 时钟</p> <p>1111: 由 PLL 时钟 15 分频作为 MCO 时钟</p> <p>其它值: 不允许设置</p>
27:25	MCO[2:0]	<p>MCU 时钟输出</p> <p>通过软件设置和清除。</p> <p>000:没有时钟</p> <p>001:LSI 时钟</p> <p>010:系统时钟</p> <p>011: HSI 时钟</p> <p>100: HSE 时钟</p> <p>101: PLL 分频后时钟</p> <p>其他值: 不允许设置</p> <p>注意:</p> <p>该时钟输出在启动和切换 MCO 时钟源时可能会被截断。</p> <p>在系统时钟作为输出至 MCO 引脚时, 应保证输出时钟频率不超过 I/O 口最高频率 30MHz。</p>
24	PLLSRC	<p>PLL 时钟源。</p> <p>软件设置或清零, 仅在 PLL 关闭时才能写入此位。</p> <p>0: HSI 时钟作为 PLL 输入时钟</p> <p>1: HSE 时钟作为 PLL 输入时钟</p>

位域	名称	描述
23:22	PLLOUTDIV[1:0]	PLL 输出时钟分频值 通过软件设置和清除。 00:不分频 01:除以 2 10:除以 3 11:除以 4
21:20	PLLPRE[1:0]	PLL 预分频器 仅在 PLL 关闭时才能写入此位。 00:PLL 输入时钟除以 1 01:PLL 输入时钟除以 2 10:PLL 输入时钟除以 3 11:PLL 输入时钟除以 4
19:16	PLLMULFCT[3:0]	PLL 倍频系数 仅在 PLL 关闭时才能写入此位。 0000: 乘以 3 0001: 乘以 4 0010: 乘以 5 0011: 乘以 6 1111: 乘以 18
15:14	SCLKSTS2[1:0]	配合 SCLKSTS 位使用
13:11	APB2PRES[2:0]	高速 APB (APB2) 预分频。 软件设置或清零, 配置 APB2 时钟 PCLK2 的预分频系数。必须保证 APB2 的时钟频率不超过 64MHz。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频
10:8	APB1PRES[2:0]	低速 APB (APB1) 预分频。 软件设置或清零, 配置 APB1 时钟 PCLK1 的预分频系数。必须保证 APB1 的时钟频率不超过 32MHz。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频
7:4	AHBPRES[3:0]	AHB 预分频。 软件设置或清零, 配置 AHB 时钟 HCLK 的预分频系数。 0xxx: SYSCLK 不分频 1000: SYSCLK 2 分频 1001: SYSCLK 4 分频 1010: SYSCLK 8 分频

位域	名称	描述
		1011: SYSCLK 16 分频 1100: SYSCLK 64 分频 1101: SYSCLK 128 分频 1110: SYSCLK 256 分频 1111: SYSCLK 512 分频
3	SCLKSTS	系统时钟切换状态，结合 SCLKSTS2 位一起使用 由硬件设置和清除，以指示使用哪个时钟源作为系统时钟。 000: HSI 振荡器用作系统时钟 001: HSE 振荡器用作系统时钟 010: PLL 用作系统时钟 011: LSI 用作系统时钟 其他值: 保留
2:0	SCLKSW[2:0]	系统时钟开关 设置并清除软件，选择 SYSCLK 源。 当退出 STOP 或 PD 模式时，或 HSE 振荡器发生故障且 CLKSSSEN 使能时，由硬件设置以强制选择 HSI。 000: HSI 被选为系统时钟 001: HSE 被选为系统时钟 010: PLL 被选为系统时钟 011: LSI 被选为系统时钟 其他值: 保留

4.3.4 时钟中断寄存器 (RCC_CLKINT)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CLKSS ICLR	Reserved	RAMP ERRCLR	PLLRD ICLR	HSERD ICLR	HSIRD ICLR	Reserved	LSIRD ICLR
r								w		w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RAMP RRSTEN	RAMP ERRIEN	PLLRD IEN	HSERD IEN	HSIRD IEN	Reserved	LSIRD IEN	CLKSSIF	Reserved	RAMP ERRIF	PLLRDIF	HSERDIF	HSIRDIF	Reserved	LSIRDIF
	rw	rw	rw	rw	rw		rw	r		r	r	r	r		r

位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23	CLKSSICLR	时钟安全系统中断清除位。 软件置 1 来清除 CLKSSIF 标志位。 0: 无作用 1: 清除 CLKSSIF 中断标志位
22	Reserved	保留，必须保持复位值。

位域	名称	描述
21	RAMPERRCLR	RAM 奇偶校验错误中断清除位。 软件置 1 来清除 RAMPERRIF 标志位。 0:不影响。 1:清除 RAMPERRIF
20	PLLRDICLR	PLL 就绪中断清除位。 软件置 1 来清除 PLLRDIF 标志位。 0: 无作用 1: 清除 PLLRDIF 中断标志位
19	HSERDICLR	HSE 就绪中断清除位。 软件置 1 来清除 HSERDIF 标志位。 0: 无使用 1: 清除 HSERDIF 中断标志位
18	HSIRDICLR	HSI 就绪中断清除位。 软件置 1 来清除 HSIRDIF 标志位。 0: 无使用 1: 清除 HSIRDIF 中断标志位
17	Reserved	保留, 必须保持复位值。
16	LSIRDICLR	LSI 就绪中断清除位。 软件置 1 来清除 LSIRDIF 标志位。 0: 无使用 1: 清除 LSIRDIF 中断标志位
15	Reserved	保留, 必须保持复位值。
14	RAMPERRRSTEN	RAM 奇偶校验错误复位使能。 0: 发生 RAM 奇偶校验错误时不产生复位 1: 发生 RAM 奇偶校验错误时产生复位
13	RAMPERRIEN	RAM 奇偶校验错误中断使能位。 软件置 1 或清零, 以使能或关闭 RAM 奇偶校验错误中断。 0:关闭 RAM 奇偶校验错误中断 1:使能 RAM 奇偶校验错误中断
12	PLLRDIEN	PLL 就绪中断使能位。 软件置 1 或清零, 以使能或关闭 PLL 就绪中断。 0: 关闭 PLL 就绪中断 1: 使能 PLL 就绪中断
11	HSERDIEN	HSE 就绪中断使能位。 软件置 1 或清零, 以使能或关闭 HSE 就绪中断。 0: 关闭 HSE 就绪中断 1: 使能 HSE 就绪中断
10	HSIRDIEN	HSI 就绪中断使能位。 软件置 1 或清零, 以使能或关闭 HSI 就绪中断。 0: 关闭 HSI 就绪中断 1: 使能 HSI 就绪中断
9	Reserved	保留, 必须保持复位值。
8	LSIRDIEN	LSI 就绪中断使能位。

位域	名称	描述
		软件置 1 或清零，以使能或关闭 LSI 就绪中断。 0: 关闭 LSI 就绪中断 1: 使能 LSI 就绪中断
7	CLKSSIF	时钟安全系统中断标志位。 在外部 HSE 振荡器出现问题时，硬件将置 1。 0: 没有 HSE 时钟错误引起的时钟安全系统中断 1: HSE 时钟错误引起了时钟安全系统中断
6	Reserved	保留，必须保持复位值。
5	RAMPERRIF	RAM 奇偶校验错误中断标志位。 当 RAMPERRIEN 被置 1 且发生 RAM 奇偶校验错误时，硬件会将此位置 1。 此位由软件对 RAMPERRCLR 位置 1 来清零。 1:发生 RAMC 奇偶校验错误 0:没有发生 RAMC 奇偶校验错误
4	PLLRDIF	PLL 就绪中断标志位。 当 PLLRDIEN 被置 1 且 PLL 时钟就绪时，硬件会将此位置 1。 此位由软件对 PLLRDICLR 位置 1 来清零。 0: 无 PLL 上锁产生的时钟就绪中断 1: PLL 上锁产生了时钟就绪中断
3	HSERDIF	HSE 就绪中断标志位。 当 HSERDIEN 被置 1 且外部高速时钟就绪时，硬件会将此位置 1。 此位由软件对 HSERDICLR 位置 1 来清零。 0: 无 HSE 振荡器产生的时钟就绪中断 1: HSE 振荡器产生了时钟就绪中断
2	HSIRDIF	HSI 就绪中断标志位。 当 HSIRDIEN 被置 1 且内部高速时钟就绪时，硬件会将此位置 1。 此位由软件对 HSERDICLR 位置 1 来清零。 0: 无 HSI 振荡器产生的时钟就绪中断 1: HSI 振荡器产生了时钟就绪中断
1	Reserved	保留，必须保持复位值。
0	LSIRDIF	LSI 就绪中断标志位。 当 LSIRDIEN 被置 1 且内部低速时钟就绪时，硬件会将此位置 1。 此位由软件对 LSIRDICLR 位置 1 来清零。 0: 无 LSI 振荡器产生的时钟就绪中断 1: LSI 振荡器产生了时钟就绪中断

4.3.5 APB2 外设复位寄存器（RCC_APB2PRST）

偏移地址：0x0c

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM5RST	TIM4RST	TIM3RST
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART2RST	UART1RST	TIM2RST	TIM1RST	SPI3RST	SPI2RST	SPI1RST	Reserved			IOPDRST	IOPCRST	IOPBRST	IOPARST	Reserved	AFIORST
rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw		rw

位域	名称	描述
31:19	Reserved	保留，必须保持复位值。
18	TIM5RST	TIM5 定时器复位。 软件置 1 或清零。 0: 清除复位 1: 复位 TIM5
17	TIM4RST	TIM4 定时器复位。 软件置 1 或清零。 0: 清除复位 1: 复位 TIM4
16	TIM3RST	TIM3 定时器复位。 软件置 1 或清零。 0: 清除复位 1: 复位 TIM3
15	UART2RST	UART2 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 UART2
14	UART1RST	UART1 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 UART1
13	TIM2RST	TIM2 定时器复位。 软件置 1 或清零。 0: 清除复位 1: 复位 TIM2
12	TIM1RST	TIM1 定时器复位。 软件置 1 或清零。 0: 清除复位 1: 复位 TIM1
11	SPI3RST	SPI3 复位 通过软件置 1 和清除。 0:清除复位 1:复位 SPI3

位域	名称	描述
10	SPI2RST	SPI2 复位 通过软件置 1 和清除。 0:清除复位 1:复位 SPI2
9	SPI1RST	SPI1 复位 通过软件置 1 和清除。 0:清除复位 1:复位 SPI1
8:6	Reserved	保留，必须保持复位值。
5	IOPDRST	GPIO 端口 D 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 GPIO 端口 D
4	IOPCRST	GPIO 端口 C 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 GPIO 端口 C
3	IOPBRST	GPIO 端口 B 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 GPIO 端口 B
2	IOPARST	GPIO 端口 A 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 GPIO 端口 A
1	Reserved	保留，必须保持复位值。
0	AFIORST	复用功能 IO 复位。 软件置 1 或清零。 0: 清除复位 1: 复位复用功能 IO

4.3.6 APB1 外设复位寄存器（RCC_APB1PRST）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWRRST	DACRST	CANRST	Reserved			I2C2RST	I2C1RST	UART5RST	UART4RST	UART3RST	Reserved	
			rw	rw	rw				rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WWDGRST	LCDRST	Reserved	COMP RST	Reserved		BEEPRST	TIM6RST	Reserved			
				rw	rw		rw			rw	rw				

位域	名称	描述
31:29	Reserved	保留，必须保持复位值。
28	PWRRST	PWR 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 PWR
27	DACRST	DAC 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 DAC
26	CANRST	CAN 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 CAN
25:23	Reserved	保留，必须保持复位值。
22	I2C2RST	I2C2 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 I2C2
21	I2C1RST	I2C1 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 I2C1
20	UART5RST	UART5 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 UART5
19	UART4RST	UART4 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 UART4
18	UART3RST	UART3 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 UART3
17:12	Reserved	保留，必须保持复位值。
11	WWDGRST	窗口看门狗复位。 软件置 1 或清零。 0: 清除复位 1: 复位窗口看门狗
10	LCDRST	LCD 复位。 软件置 1 或清零。

位域	名称	描述
		0: 清除复位 1: 复位 LCD
9	Reserved	保留, 必须保持复位值。
8	COMPRST	COMP 复位。 软件置 1 或清零。 0: 清除复位 1: 复位 COMP
7:6	Reserved	保留, 必须保持复位值。
5	BEEPRST	蜂鸣器复位。 软件置 1 或清零。 0: 清除复位 1: 复位蜂鸣器
4	TIM6RST	TIM6 定时器复位。 软件置 1 或清零。 0: 清除复位 1: 复位 TIM6 定时器
3:0	Reserved	保留, 必须保持复位值。

4.3.7 AHB 外设时钟使能寄存器 (RCC_AHBPCLOCKEN)

偏移地址: 0x14

复位值: 0x0000 0014

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

位域	名称	描述
31:13	Reserved	保留, 必须保持复位值。
12	ADCEN	ADC 时钟使能。 软件置 1 或清零。 0: 关闭 ADC 时钟 1: 使能 ADC 时钟
11	SACEN	SAC 时钟使能。 软件置 1 或清零。 0: 关闭 SAC 时钟 1: 使能 SAC 时钟
10:7	Reserved	保留, 必须保持复位值。

位域	名称	描述
6	CRCEN	CRC 时钟使能。 软件置 1 或清零。 0: 关闭 CRC 时钟 1: 使能 CRC 时钟
5	Reserved	保留, 必须保持复位值。
4	FLITFEN	SLEEP 模式 Flash 接口时钟使能。 软件置 1 或清零。 0: 关闭睡眠模式时 Flash 接口时钟 1: 使能睡眠模式时 Flash 接口时钟
3	Reserved	保留, 必须保持复位值。
2	SRAMEN	SLEEP 模式 SRAM 接口时钟使能。 软件置 1 或清零。 0: 关闭睡眠模式时 SRAM 接口时钟 1: 使能睡眠模式时 SRAM 接口时钟
1	Reserved	保留, 必须保持复位值。
0	DMAEN	DMA 时钟使能。 软件置 1 或清零。 0: 关闭 DMA 时钟 1: 使能 DMA 时钟

4.3.8 APB2 外设时钟使能寄存器 (RCC_APB2PCLKEN)

偏移地址: 0x18

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM5EN	TIM4EN	TIM3EN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART2EN	UART1EN	TIM2EN	TIM1EN	SPI3EN	SPI2EN	SPI1EN	Reserved			IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFIOEN
rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw		rw

位域	名称	描述
31:19	Reserved	保留, 必须保持复位值。
18	TIM5EN	TIM5 定时器时钟使能。 软件置 1 或清零。 0: 关闭 TIM5 定时器时钟 1: 使能 TIM5 定时器时钟
17	TIM4EN	TIM4 定时器时钟使能。 软件置 1 或清零。 0: 关闭 TIM4 定时器时钟

位域	名称	描述
		1: 使能 TIM4 定时器时钟
16	TIM3EN	TIM3 定时器时钟使能。 软件置 1 或清零。 0: 关闭 TIM3 定时器时钟 1: 使能 TIM3 定时器时钟
15	UART2EN	UART2 时钟使能。 软件置 1 或清零。 0: 关闭 UART2 时钟 1: 使能 UART2 时钟
14	UART1EN	UART1 时钟使能。 软件置 1 或清零。 0: 关闭 UART1 时钟 1: 使能 UART1 时钟
13	TIM2EN	TIM2 定时器时钟使能。 软件置 1 或清零。 0: 关闭 TIM2 定时器时钟 1: 使能 TIM2 定时器时钟
12	TIM1EN	TIM1 定时器时钟使能。 软件置 1 或清零。 0: 关闭 TIM1 定时器时钟 1: 使能 TIM1 定时器时钟
11	SPI3EN	SPI3 时钟使能。 软件置 1 或清零。 0: 关闭 SPI3 时钟 1: 使能 SPI3 时钟
10	SPI2EN	SPI2 时钟使能。 软件置 1 或清零。 0: 关闭 SPI2 时钟 1: 使能 SPI2 时钟
9	SPI1EN	SPI1 时钟使能。 软件置 1 或清零。 0: 关闭 SPI1 时钟 1: 使能 SPI1 时钟
8:6	Reserved	保留, 必须保持复位值。
5	IOPDEN	GPIO 端口 D 时钟使能。 软件置 1 或清零。 0: 关闭 GPIO 端口 D 的时钟 1: 使能 GPIO 端口 D 的时钟
4	IOPCEN	GPIO 端口 C 时钟使能。 软件置 1 或清零。 0: 关闭 GPIO 端口 C 的时钟 1: 使能 GPIO 端口 C 的时钟
3	IOPBEN	GPIO 端口 B 时钟使能。

位域	名称	描述
		软件置 1 或清零。 0: 关闭 GPIO 端口 B 的时钟 1: 使能 GPIO 端口 B 的时钟
2	IOPAEN	GPIO 端口 A 时钟使能。 软件置 1 或清零。 0: 关闭 GPIO 端口 A 的时钟 1: 使能 GPIO 端口 A 的时钟
1	Reserved	保留, 必须保持复位值。
0	AFIOEN	复用功能 IO 时钟使能。 软件置 1 或清零。 0: 关闭复用功能 IO 时钟 1: 使能复用功能 IO 时钟

4.3.9 APB1 外设时钟使能寄存器 (RCC_APB1PCLKEN)

偏移地址: 0x1c

复位值: 0x1000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	PWREN	DACEN	CANEN	Reserved	I2C2EN	I2C1EN	UART5 EN	UART4 EN	UART3 EN	Reserved					
	rw	rw	rw		rw	rw	rw	rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	WWDG EN	LCDEN	COMP FILTEN	COMPEN	Reserved	BEEPEN	TIM6EN	Reserved							
	rw	rw	rw	rw		rw	rw								

位域	名称	描述
31:29	Reserved	保留, 必须保持复位值。
28	PWREN	电源接口时钟使能。 软件置 1 或清零。 0: 关闭电源接口时钟 1: 使能电源接口时钟
27	DACEN	DAC 时钟使能。 软件置 1 或清零。 0: 关闭 DAC 时钟 1: 使能 DAC 时钟
26	CANEN	I2C2 时钟使能。 软件置 1 或清零。 0: 关闭 CAN 时钟 1: 使能 CAN 时钟
25:23	Reserved	保留, 必须保持复位值。
22	I2C2EN	I2C2 时钟使能。

位域	名称	描述
		软件置 1 或清零。 0: 关闭 I2C2 时钟 1: 使能 I2C2 时钟
21	I2C1EN	I2C1 时钟使能。 软件置 1 或清零。 0: 关闭 I2C1 时钟 1: 使能 I2C1 时钟
20	UART5EN	UART5 时钟使能。 软件置 1 或清零。 0: 关闭 UART5 时钟 1: 使能 UART5 时钟
19	UART4EN	UART4 时钟使能。 软件置 1 或清零。 0: 关闭 UART4 时钟 1: 使能 UART4 时钟
18	UART3EN	UART3 时钟使能。 软件置 1 或清零。 0: 关闭 UART3 时钟 1: 使能 UART3 时钟
17:12	Reserved	保留, 必须保持复位值。
11	WWDGEN	窗口看门狗时钟使能。 软件置 1 或清零。 0: 关闭窗口看门狗时钟 1: 使能窗口看门狗时钟
10	LCDEN	LCD 时钟使能。 软件置 1 或清零。 0: 关闭 LCD 时钟 1: 使能 LCD 时钟
9	COMPFILTEN	比较器过滤器时钟使能。 0: 关闭比较器过滤器时钟 1: 使能比较器过滤器时钟
8	COMPEN	比较器时钟使能。 0: 关闭比较器时钟 1: 使能比较器时钟
7:6	Reserved	保留, 必须保持复位值。
5	BEEPEN	蜂鸣器时钟使能。 0: 关闭蜂鸣器时钟 1: 使能蜂鸣器时钟
4	TIM6EN	TIM6 定时器时钟使能。 软件置 1 或清零。 0: 关闭 TIM6 定时器时钟 1: 使能 TIM6 定时器时钟
3:0	Reserved	保留, 必须保持复位值。

4.3.10 低速时钟控制寄存器（RCC_LSCTRL）

偏移地址：0x20

复位值：0x0000 0043

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

位域	名称	描述
31:15	Reserved	保留，必须保持复位值。
14	RTCRST	RTC 软件复位 软件置 1 或清零。 0:清除复位 1:复位 RTC
13	RTCEN	RTC 时钟使能 通过软件置 1 和清除。 0: 关闭 RTC 时钟 1: 使能 RTC 时钟
12:11	RTCSEL[1:0]	RTC 时钟源选择。 软件设置这些位以选择 RTC 时钟源。 00: 无时钟 01: 选择 LSI 振荡器作为 RTC 时钟 10: 选择 HSE 振荡器 128 分频后作为 RTC 时钟 11: 选择 HSI 振荡器 128 分频后作为 RTC 时钟
10:7	Reserved	保留，必须保持复位值。
6:2	LSITRIM[4:0]	内部低速时钟校准值 由软件写入，用以校准内部 LSI RC 振荡器的频率。 默认值为 16，调节步长为 720Hz。 设置值(取整) = 当前值 - (LSI 实测频率-目标频率 32000Hz) / (步长 720Hz)。
1	LSIRDF	内部低速时钟振荡器就绪。 硬件置 1 或清零，以指示 LSI 振荡器是否就绪。在 LSIEN 被清零后，该位需要 3 个内部低速振荡器的周期才能被清零。 0: LSI 未就绪 1: LSI 已就绪
0	LSIEN	内部低速时钟振荡器使能位。 软件置 1 或清零。 0: 关闭 LSI 振荡器

位域	名称	描述
		1: 开启 LSI 振荡器

4.3.11 控制/状态寄存器 (RCC_CTRLSTS)

偏移地址: 0x24

复位值: 0x00000018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		LVRSTF	Reserved												
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LKUP RSTF	EMCGB RSTF	EMCGBN RSTF	LPWR RSTF	WWDG RSTF	IWDG RSTF	SFTRSTF	PORRSTF	PINRSTF	MMU RSTF	RAMP ERRRSTF	RMRSTF
				r	r	r	r	r	r	r	r	r	r	r	rw

位域	名称	描述
31:30	Reserved	保留, 必须保持复位值。
29	LVRSTF	低电压复位标志 由硬件在低电压复位时置 1。 通过写入 RMRSTF 位清除。 0: 没有低电压复位发生 1: 低电压复位出现
29:12	Reserved	保留, 必须保持复位值。
11	LKUPRSTF	M0 内核锁定复位标志 由硬件在 M0 内核锁定复位时置 1。 通过写入 RMRSTF 位清除。 0: 没有 M0 内核锁定复位发生 1: M0 内核锁定复位出现
10	EMCGBRSTF	EMCGB 复位标志 由硬件在 EMCGB 复位时置 1。 通过写入 RMRSTF 位清除。 0: 没有 EMCGB 复位发生 1: EMCGB 复位出现
9	EMCGBNRSTF	EMCGBN 复位标志 由硬件在 EMCGBN 复位时置 1。 通过写入 RMRSTF 位清除。 0: 没有 EMCGBN 复位发生 1: EMCGBN 复位出现
8	LPWRRSTF	低功耗复位标志 由硬件在低功耗复位时置 1。 通过写入 RMRSTF 位清除。 0: 没有 Low-power 复位发生

位域	名称	描述
		1: Low-power 复位出现
7	WWDGRSTF	窗口看门狗复位标志 由硬件在窗口看门狗复位时置 1。 通过写入 RMRSTF 位清除。 0:没有窗口看门狗复位发生 1:窗口看门狗复位出现
6	IWDGRSTF	独立看门狗复位标志 由硬件在独立看门狗复位时置 1。 通过写入 RMRSTF 位清除。 0:没有独立看门狗复位发生 1:独立看门狗复位出现
5	SFTRSTF	软件复位标志 由硬件在软件复位时置 1。 通过写入 RMRSTF 位清除。 0:没有软件复位发生 1:软件复位出现
4	PORRSTF	POR/PDR 复位标志 由硬件在 POR/PDR 复位时置 1。 通过写入 RMRSTF 位清除 0:没有 POR/PDR 复位发生 1: POR/PDR 复位出现
3	PINRSTF	PIN 复位标志 由硬件在 NRST 引脚复位时置 1。 通过写入 RMRSTF 位清除。 0:没有 NRST 引脚复位发生 1: NRST 引脚复位出现
2	MMURSTF	MMU 复位标志 由硬件在 MMU 复位时置 1。 通过写入 RMRSTF 位清除。 0:没有 MMU 复位发生 1: MMU 复位出现
1	RAMPERRRSTF	RAM 奇偶校验错误复位标志 由硬件在 RAM 奇偶校验错误复位时置 1。 通过写入 RMRSTF 位清除。 0:没有 RAM 奇偶校验错误复位发生 1: RAM 奇偶校验错误复位出现
0	RMRSTF	清除复位标志 由软件设置此位清除所有复位标志。 0:无作用 1:清除所有复位标志

4.3.12 AHB 外设复位寄存器（RCC_AHBPRST）

偏移地址：0x28

复位值：0x0000 0000

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

位域	名称	描述
31:13	Reserved	保留，必须保持复位值。
12	ADCRST	ADC 接口复位 通过软件置 1 和清除。 0:清除复位 1:复位 ADC 接口
11	SACRST	SAC 接口复位 通过软件置 1 和清除。 0:清除复位 1:复位 SAC 接口
10:0	Reserved	保留，必须保持复位值。

4.3.13 时钟配置寄存器 2（RCC_CFG2）

偏移地址：0x2c

复位值：0x0000 3800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIM1CLK SEL	TIM6CLK SEL	Reserved		LCDCLK SEL	ADCCLKSEL[1:0]		Reserved								
rw	rw			rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC1MPRE[4:0]					ADC1M SEL	Reserved	ADCPLLPRE[4:0]					ADCHPRE[3:0]			
rw					rw		rw					rw			

位域	名称	描述
31	TIM1CLKSEL	TIM1 时钟源选择 该位由软件进行设置和清除。 0: 如果 APB2 预分频为 1, 则 TIM1 时钟源为 PCLK2; 否则, TIM1 时钟源为 PCLK2×2。 1: TIM1 时钟源为 SYSCLK
30	TIM6CLKSEL	TIM6 时钟源选择 该位由软件进行设置和清除。 0: 如果 APB1 预分频为 1, 则 TIM1 时钟源为 PCLK1; 否则, TIM6 时钟源为 PCLK1×2。 1: TIM1 时钟源为 LSI
29:28	Reserved	保留, 必须保持复位值。
27	LCDCLKSEL	LCD 时钟源选择位 该位由软件进行设置和清除 0: 选择 HSI 时钟 8 分频 1: 选择 HSE 时钟 16 分频
26:25	ADCCLKSEL[1:0]	ADC 工作时钟源选择位 该位由软件进行设置和清除 00: 选择 24M HSI 时钟 01: 选择 PLL 时钟 10: 选择 HCLK 时钟 11: 保留
24:16	Reserved	保留, 必须保持复位值。
15:11	ADC1MPRE[4:0]	ADC 1M 时钟预分频 软件设置或清除这些位来配置 ADC 1M 时钟源的预分频系数。 00000: ADC 1M 时钟源不分频 00001: ADC 1M 时钟源 2 分频 00010: ADC 1M 时钟源 3 分频 ... 11110: ADC 1M 时钟源 31 分频 11111: ADC 1M 时钟源 32 分频
10	ADC1MSEL	ADC 1M 时钟源选择 软件置 1 或清零。 0: 选择 HSI 振荡器时钟作为 ADC 1M 的输入时钟 1: 选择 HSE 振荡器时钟作为 ADC 1M 的输入时钟 <i>注意: 切换 ADC 1M 时钟源时, 需确保 HSI 打开</i>
9	Reserved	保留, 必须保持复位值。
8:4	ADCPLLPRE[4:0]	ADC PLL 预分频 软件设置或清除这些位以配置 PLL 时钟到 ADC 的分频系数。 0xxxx: ADC PLL 时钟被关闭 10000: PLL 时钟不分频 10001: PLL 时钟 2 分频 10010: PLL 时钟 4 分频 10011: PLL 时钟 6 分频

位域	名称	描述
		10100: PLL 时钟 8 分频 10101: PLL 时钟 10 分频 10110: PLL 时钟 12 分频 10111: PLL 时钟 16 分频 11000: PLL 时钟 32 分频 11001: PLL 时钟 64 分频 11010: PLL 时钟 128 分频 11011: PLL 时钟 256 分频 其它值: PLL 时钟 256 分频
3:0	ADCHPRE[3:0]	ADC HCLK 预分频 软件设置或清除这些位以配置 HCLK 时钟到 ADC 的分频系数。 0000: HCLK 时钟 1 分频 0001: HCLK 时钟 2 分频 0010: HCLK 时钟 4 分频 0011: HCLK 时钟 6 分频 0100: HCLK 时钟 8 分频 0101: HCLK 时钟 10 分频 0110: HCLK 时钟 12 分频 0111: HCLK 时钟 16 分频 1000: HCLK 时钟 32 分频 其它值: HCLK 时钟 32 分频

4.3.14 时钟配置寄存器 3 (RCC_CFG3)

偏移地址: 0x30

复位值: 0x7805 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GCLKNUM[7:0]								GCLKMNUM[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				GCLKPRES[3:0]				Reserved				GCLKDMAEN	GCLKDONE	GCLKSEL	GCLKEN
rw				rw				rw				rw	rc_w1	rw	rw

位域	名称	描述
31:24	GCLKNUM[7:0]	一个周期内的时钟数 由软件来设置或清除。详细描述见章节 18.3.2。 0x00: N = 0 0x01: N = 1 0x02: N = 2

位域	名称	描述
		0xFF: N = 255
23:16	GCLKMNUM[7:0]	一个周期内的低电平时钟数 由软件来设置或清除。详细描述见章节 18.3.2。 0x00: N = 0 0x01: N = 1 0x02: N = 2 0xFF: N = 255
15:12	Reserved	保留，必须保持复位值。
11:8	GCLKPRES[3:0]	LED GCLK 时钟预分频 软件设置或清除这些位来配置 LED GCLK 时钟源的预分频系数。 0xxxx: GCLK 输出关闭 1000: GCLK 时钟源 32 分频 1001: GCLK 时钟源 40 分频 1010: GCLK 时钟源 50 分频 1011: GCLK 时钟源 64 分频 1100: GCLK 时钟源 80 分频 1101: GCLK 时钟源 160 分频 其他值: GCLK 时钟源 64 分频
7:4	Reserved	保留，必须保持复位值。
3	GCLKDMAEN	LED GCLK 的 DMA 请求使能 0: 禁止 LED GCLK 的 DMA 请求 1: 使能 LED GCLK 的 DMA 请求
2	GCLKDONE	LED GCLK 时钟输出脉冲个数完成标志 LED GCLK 时钟输出脉冲数量达到配置值 M+N 后由硬件置位，对该位写 1 清零。 0: 输出脉冲个数已完成 1: 输出脉冲个数未完成
1	GCLKSEL	LED GCLK 时钟源选择 软件置 1 或清零。 0: 选择 HSI 时钟 1: 选择 HSE 时钟
0	GCLKEN	LED GCLK 时钟使能位 由软件置 1 和清零。 0: 关闭 LED GCLK 1: 使能 LED GCLK

4.3.15 EMC 控制寄存器 (RCC_EMCCTRL)

偏移地址: 0x34

复位值: 0x0000 0000

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

Reserved								GBRST3	GBRST2	GBRST1	GBRST0	GBNRST3	GBNRST2	GBNRST1	GBNRST0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				GBDET3	GBDET2	GBDET1	GBDET0	GBNDET3	GBNDET2	GBNDET1	GBNDET0	Reserved			
				rw	rw	rw	rw	rw	rw	rw	rw				

位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23	GBRST3	GB3 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
22	GBRST2	GB2 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
21	GBRST1	GB1 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
20	GBRST0	GB0 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
19	GBNRST3	GBN3 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
18	GBNRST2	GBN2 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
17	GBNRST1	GBN1 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
16	GBNRST0	GBN0 复位使能 这个位由软件置 1 和清除。 0:禁止复位请求 1:开启复位请求
15:12	Reserved	保留，必须保持复位值。
11	GBDET3	GB3 检测使能

位域	名称	描述
		这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
10	GBDET2	GB2 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
9	GBDET1	GB1 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
8	GBDET0	GB0 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
7	GBNDET3	GBN3 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
6	GBNDET2	GBN2 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
5	GBNDET1	GBN1 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
4	GBNDET0	GBN0 检测使能 这个位由软件置 1 和清除。 0:禁止检测 1:开启检测
3:0	Reserved	保留，必须保持复位值。

5 GPIO 和 AFIO

5.1 概述

GPIO (General purpose input/output) 即通用型 I/O, AFIO (Alternate-function input/output) 即复用功能 I/O。芯片最多支持 61 个 GPIO, 共被分为 4 组 (GPIOA/GPIOB/GPIOC/GPIOD), A/B/C/每组 16 个端口,D 组 13 个 IO。GPIO 端口和其他的复用外设共用引脚, 用户可以根据需求灵活配置。每个 GPIO 引脚都可以独立配置成输出、输入或复用的外设功能端口。除了模拟输入引脚外, 其他的 GPIO 引脚都有大电流通过能力。

GPIO 端口具有以下特性:

■ GPIO 端口可由软件分别配置成以下模式:

- ◆ 输入浮空
- ◆ 输入上拉
- ◆ 输入下拉
- ◆ 模拟功能
- ◆ 开漏输出及上/下拉可配
- ◆ 推挽输出及上/下拉可配
- ◆ 推挽复用功能及上/下拉可配
- ◆ 开漏复用功能及上/下拉可配

■ 单独的位设置或位清除功能

■ 所有 I/O 支持外部中断功能

■ 所有 I/O 支持低功耗模式唤醒, 上升或下降沿可配置

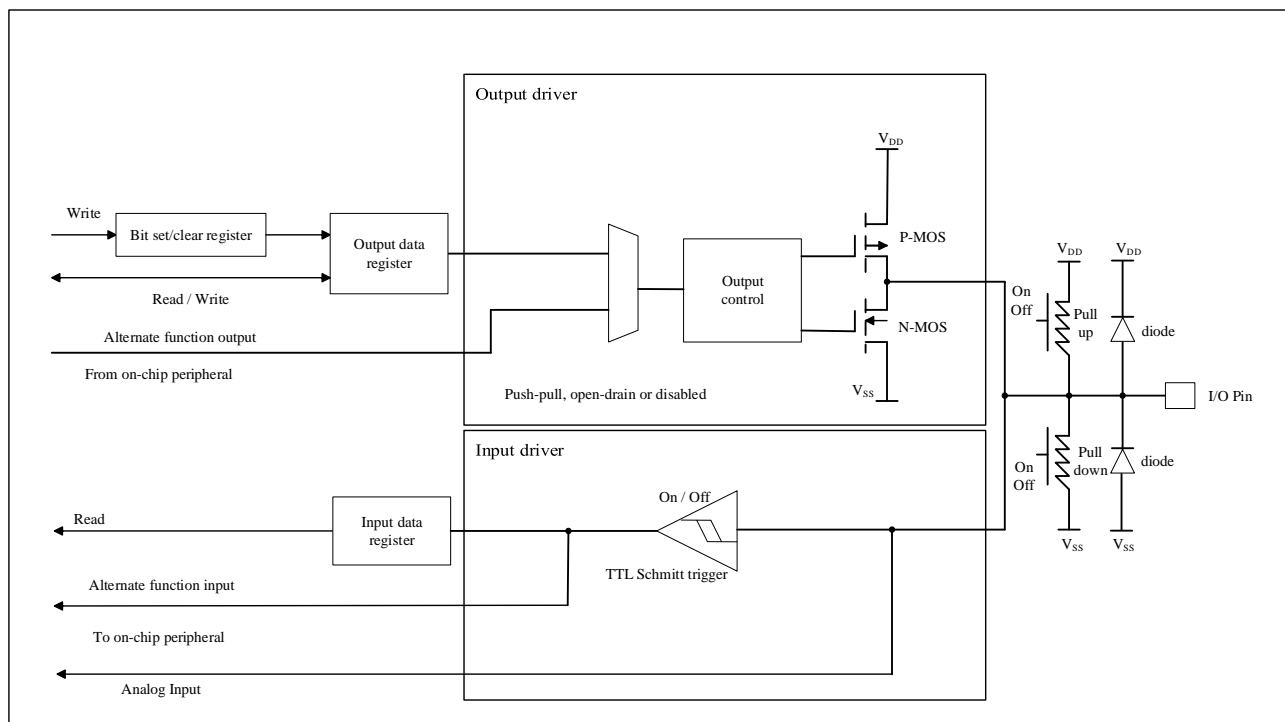
- ◆ 16 个 EXTI 可用于 SLEEP 或 STOP 模式唤醒, 所有 I/O 可复用为 EXTI
- ◆ NRST/PA0/PA2 三个唤醒 I/O 可用于 PD 模式唤醒, I/O 滤波时间最大 1us

■ 支持软件重新映射 I/O 复用功能

■ 支持 GPIO 锁定机制, 复位方式清除锁定状态

每个 I/O 端口位可以任意编程, 但必须按照 32 位字访问 I/O 端口寄存器 (不允许 16 位半字或 8 位字节访问)。下图给出了一个 I/O 端口的基本结构。

图 5-1 I/O 端口的基本结构（不支持 Fail-safe）



5.2 功能描述

5.2.1 IO 模式配置

IO 的模式控制由配置寄存器 GPIOx_PMODE, GPIOx_POTYPE 和 GPIOx_PUPD (x=A,B,C,D) 来设置, 不同的操作模式下的配置如下表所示:

表 5-1 IO 模式和配置关系

PMODE[1:0]	POTYPE	PUPD[1:0]		I/O 配置
01	0	0	0	通用输出推挽 (Push-Pull)
	0	0	1	通用输出推挽 (Push-Pull) + 上拉
	0	1	0	通用输出推挽 (Push-Pull) + 下拉
	0	1	1	保留
	1	0	0	通用输出开漏 (Open-Drain)
	1	0	1	通用输出开漏 (Open-Drain) + 上拉
	1	1	0	通用输出开漏 (Open-Drain) + 下拉
	1	1	1	保留
10	0	0	0	复用功能+推挽 (Push-Pull)
	0	0	1	复用功能+推挽 (Push-Pull) + 上拉
	0	1	0	复用功能+推挽 (Push-Pull) + 下拉
	0	1	1	保留
	1	0	0	复用功能+开漏 (Open-Drain)
	1	0	1	复用功能+开漏 (Open-Drain) + 上拉

PMODE[1:0]	POTYPE	PUPD[1:0]		I/O 配置
	1	1	0	复用功能+开漏（Open-Drain）+下拉
	1	1	1	保留
00	x	0	0	浮空输入
	x	0	1	上拉输入
	x	1	0	下拉输入
	x	1	1	保留
11	x	0	0	模拟模式
	x	0	1	保留
	x	1	0	
	x	1	1	

另外 GPIOx_DS.DS_y 位可用来配置高/低驱动强度，GPIOx_SR.SR_y 位可用来高/低翻转速率的配置。

IO 在不同的配置下的输入输出特性如下表所示：

表 5-2 IO 不同配置的输入输出特性

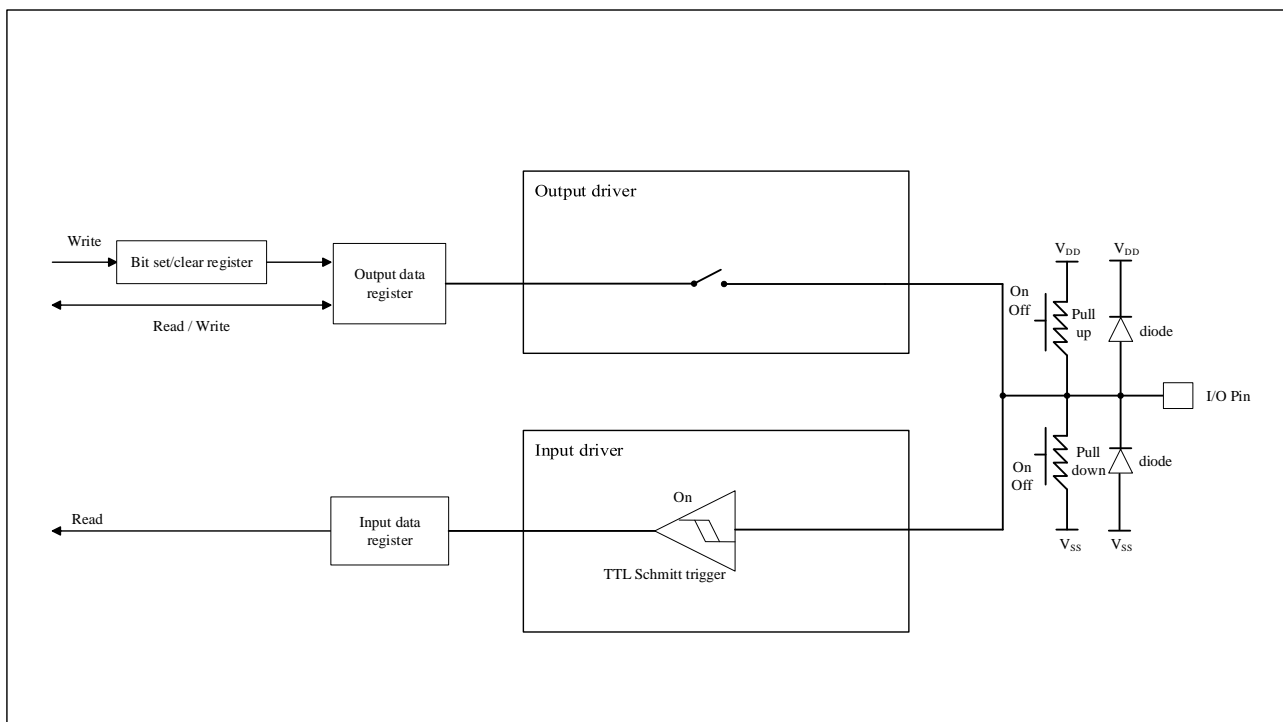
特性	GPIO 输入	GPIO 输出	模拟	外设复用
输出缓冲器	禁能	使能	禁能	根据外设功能配置
施密特触发器	使能	使能	禁能 输出值被强制为 0	使能
上下拉/浮空	可配	可配	禁能	根据外设功能配置
开漏模式	禁能	可配， 输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 高 阻	禁能	可配，输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 高阻
推挽模式	禁能	可配， 输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 输 出 1	禁能	可配，输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 输出 1
输入数据寄存器（IO 状态）	可读	可读	读出为 0	可读
输出数据寄存器（输出值）	无效	可读写	无效	可读

5.2.1.1 输入模式

当 I/O 端口配置为输入模式时：

- 输出缓冲器被禁止
- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx_PUPD 寄存器的配置
- 出现在 I/O 脚上的数据在每个 APB2 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问得到 I/O 状态

图 5-2 输入浮空/上拉/下拉模式（不支持 Fail-safe）

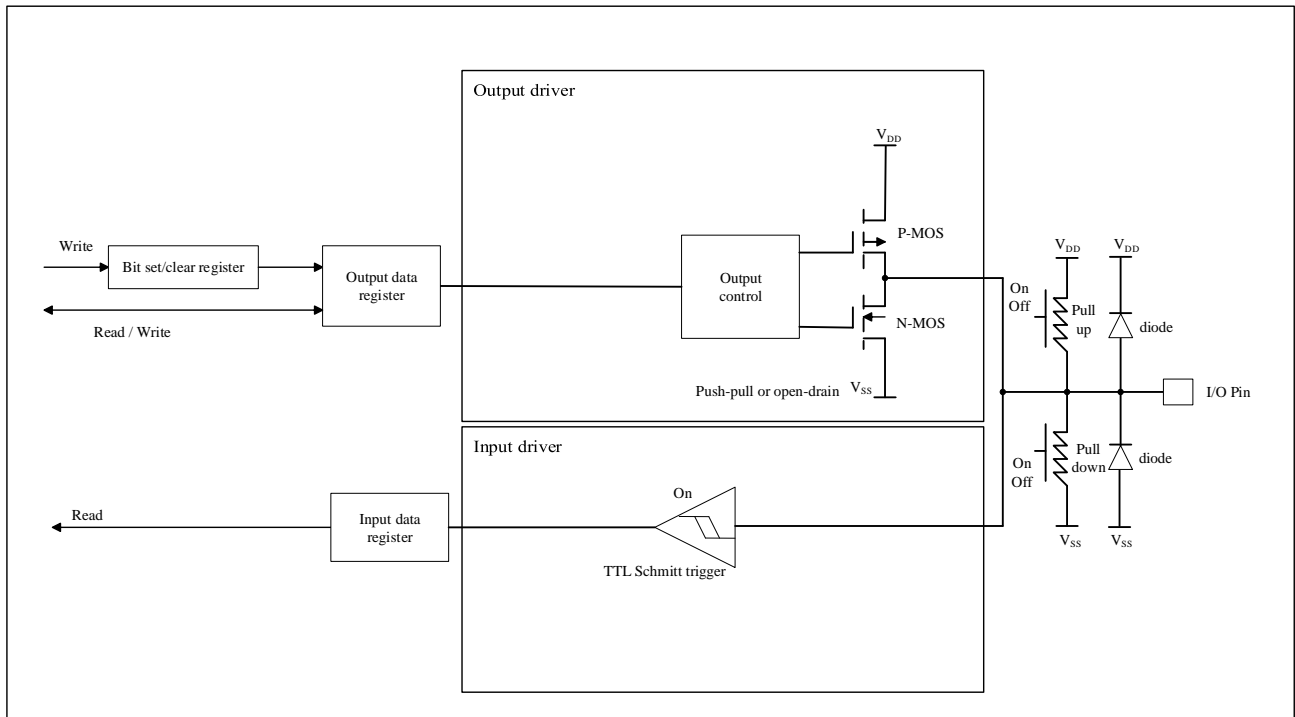


5.2.1.2 输出模式

当 I/O 端口配置为输出模式时：

- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx_PUPD 寄存器的配置
- 输出缓冲器被激活
 - ◆ 开漏模式： 输出数据寄存器上的'0'激活 N-MOS，引脚输出低电平
输出数据寄存器上的'1'使端口置于高阻状态（P-MOS 从不被激活）
 - ◆ 推挽模式： 输出数据寄存器上的'0'激活 N-MOS，引脚输出低电平
输出数据寄存器上的'1'激活 P-MOS，引脚输出高电平
- 出现在 I/O 脚上的数据在每个 APB2 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后一次写入的值

图 5-3 输出模式（不支持 Fail-safe）

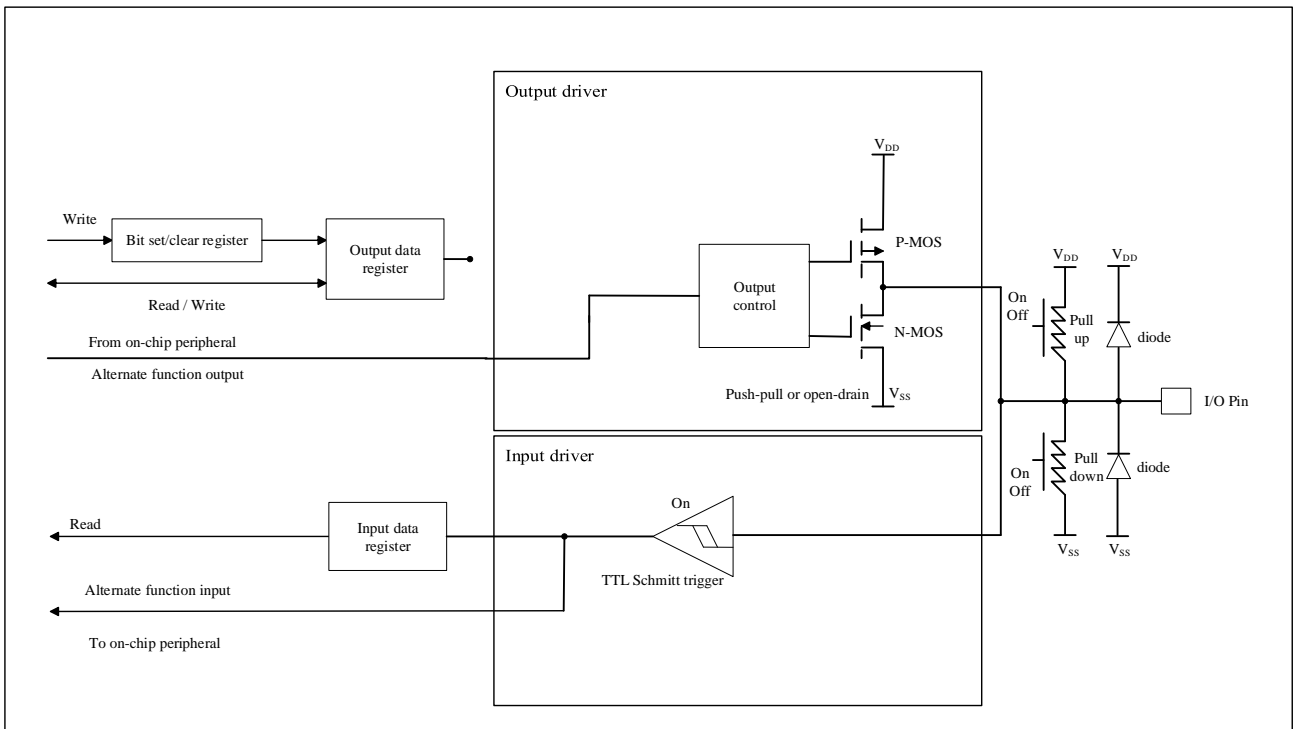


5.2.1.3 复用功能模式

当 I/O 端口配置为复用功能模式时：

- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx_PUPD 寄存器的配置
- 在开漏或推挽式配置中，输出缓冲器由外设控制
- 内置外设的信号驱动输出缓冲器
- 在每个 APB2 时钟周期，出现在 I/O 脚上的数据被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后一次写入的值

图 5-4 复用功能模式（不支持 Fail-safe）

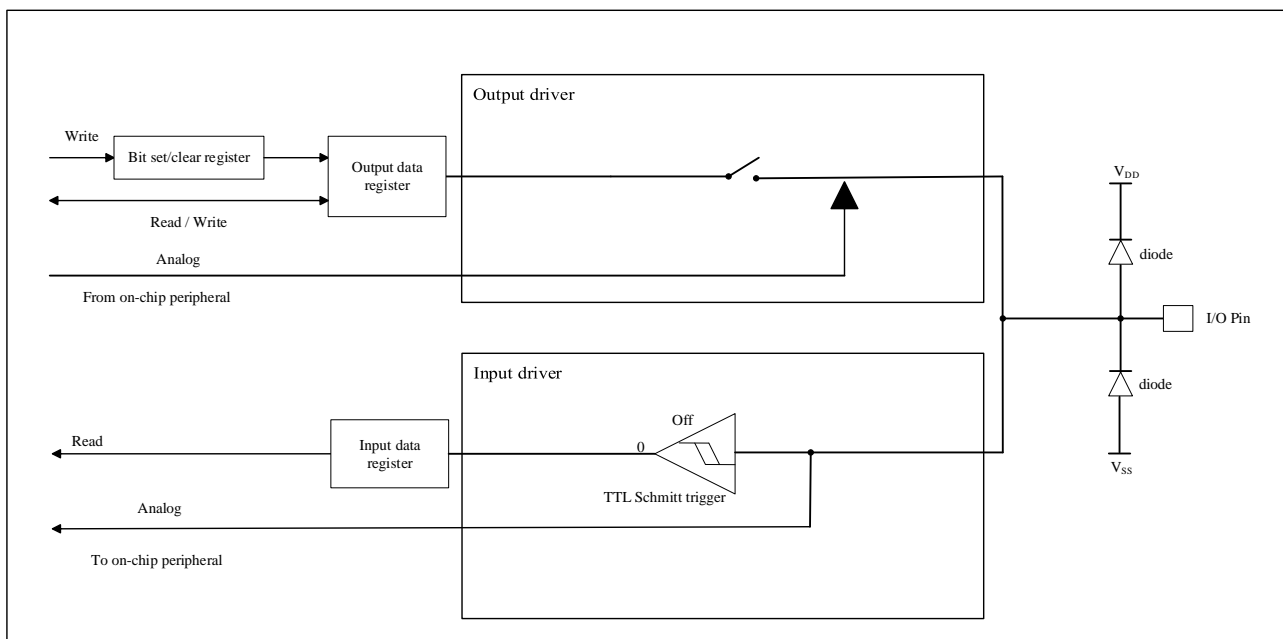


5.2.1.4 模拟功能模式

当 I/O 端口被配置为模拟功能模式时：

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

图 5-5 高阻抗的模拟功能模式（不支持 Fail-safe）



5.2.2 复位后状态

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成模拟功能模式（GPIOx_PMODE.PMODEx[1:0]=2'b11）。但有以下几个例外的信号：

- NRST 默认无 GPIO 功能：
 - ◆ NRST 上拉输入
- 复位后，调试系统相关的引脚默认配置为 SWD 接口 I/O 配置：
 - ◆ PA14: SWCLK 置于输入下拉模式
 - ◆ PA13: SWDIO 置于输入上拉模式
- PA0 和 PD14：
 - ◆ PA0 和 PD14 默认浮空输入模式
 - ◆ PD14 复用到 OSC_IN
- PD0/BOOT0：
 - ◆ 芯片启动过程中，PD0 作为 BOOT0 功能，默认下拉输入模式，芯片初始化完成后，用作通用 GPIO，被配置成模拟功能模式

5.2.3 单独的位设置和位清除

通过对“位设置/清除寄存器（GPIOx_PBSC）和位清除寄存器（GPIOx_PBC）”中想要更改的位写‘1’来实现对数据寄存器（GPIOx_POD）的个别位操作，可以一个或多个位。写‘1’的位被相应的置位或清除，没被写‘1’的位将不被更改。软件不需要禁止中断，在单次 APB2 写操作里完成。

5.2.4 外部中断/唤醒线

所有端口都有外部中断能力，可以在 EXTI 模块中配置：

- 端口必须配置成输入模式
- 所有端口可配置用于 SLEEP/STOP 模式唤醒，支持上升或下降沿可配
- NRST/PA0/PA2，可用于 PD 模式唤醒，PA0/PA2 有独立的唤醒使能，三个引脚均支持上升沿/下降沿可配，需要在进 PD 模式前配置
- 通用 I/O 端口以图 6-2 的方式连接到 16 个外部中断/事件线上，由寄存器 AFIO_EXTI_CFGx 配置

5.2.5 复用功能

当 I/O 端口被配置为复用功能模式时，使用前必须对端口位配置寄存器（GPIOx_AFL/ GPIOx_AFH，GPIOx_PMODE，GPIOx_POTYPE 和 GPIOx_PUPD），复用输入或输出由外设确定。

5.2.5.1 软件重新映射 I/O 复用功能

为拓展不同器件封装下的复用外设功能灵活性，可以把一些外设复用功能重新映射到其他引脚上。每个 IO 有多达 16 个可复用的功能 (AF0~ AF15)。复位后，除 PA13 和 PA14 外，AFSELY 默认选择为 AF15。可以通过软件配置相应的寄存器（GPIOx_AFL/AFH）来重新映射 IO 复用功能。

这时，复用功能就不再映射到它们的原始引脚上(对于外设的 IO 重映射功能，若重映射到不同的引脚，则输入为重映射多选一，输出则接到重映射后的位置，原位置断开)。

5.2.5.1.1 SWD 复用功能 I/O 重映射

表 5-3 SWD 复用功能 I/O 重映射

复用功能	I/O	重映射
SWDIO	PA13	AF0
SWCLK	PA14	AF0

5.2.5.1.2 TIMx 复用功能 I/O 重映射

5.2.5.1.2.1 TIM1 复用功能 I/O 重映射

表 5-4 TIM1 复用功能 I/O 重映射

复用功能	IO	重映射
TIM1_ETR	PA12	AF2
	PA14	AF2
	PA15	AF2
TIM1_BKIN	PA1	AF2
	PA5	AF3
	PA11	AF2
TIM1_CH1	PA4	AF2
	PA9	AF2
	PC12	AF2
	PD4	AF2

TIM1_CH2	PA6	AF2
	PA9	AF1
	PC12	AF1
	PD6	AF2
TIM1_CH3	PA11	AF5
	PD4	AF1
	PD8	AF2
TIM1_CH4	PA10	AF2
	PA13	AF2
	PD4	AF3
TIM1_CH1N	PA5	AF2
	PA7	AF2
	PB13	AF2
	PD5	AF2
TIM1_CH2N	PA6	AF1
	PA7	AF1
	PB14	AF2
	PD7	AF2
TIM1_CH3N	PA4	AF3
	PA9	AF3
	PB15	AF2
	PD9	AF2
TIM1_CH4N	PA11	AF4
	PA14	AF4
	PB12	AF7
	PD12	AF3

5.2.5.1.2.2 TIM2 复用功能 I/O 重映射

表 5-5 TIM2 复用功能 I/O 重映射

复用功能	IO	重映射
TIM2_ETR	PA5	AF1
	PB9	AF2
TIM2_CH1	PA10	AF1
	PA14	AF1
	PB0	AF2
TIM2_CH2	PA11	AF1
	PB14	AF1
	PD0	AF2
TIM2_CH3	PA12	AF1
	PB12	AF2
	PC14	AF2
TIM2_CH4	PA13	AF1
	PB6	AF4

复用功能	IO	重映射
	PD14	AF2

5.2.5.1.2.3 TIM3 复用功能 I/O 重映射

表 5-6 TIM3 复用功能 I/O 重映射

复用功能	IO	重映射
TIM3_ETR	PC5	AF2
	PC13	AF2
TIM3_CH1	PA8	AF1
	PC6	AF2
	PC11	AF2
	PD10	AF2
TIM3_CH2	PA8	AF4
	PC7	AF2
	PC11	AF1
TIM3_CH3	PC8	AF2
	PD5	AF1
	PD12	AF1
TIM3_CH4	PC9	AF2
	PD5	AF0
	PD12	AF2

5.2.5.1.2.4 TIM4 复用功能 I/O 重映射

表 5-7 TIM4 复用功能 I/O 重映射

复用功能	IO	重映射
TIM4_ETR	PC4	AF2
	PC12	AF3
TIM4_CH1	PA10	AF3
	PB6	AF1
	PC1	AF2
	PD11	AF2
TIM4_CH2	PA10	AF4
	PB7	AF2
	PC1	AF1
TIM4_CH3	PB7	AF1
	PB8	AF2
	PD13	AF2
TIM4_CH4	PB8	AF1
	PB9	AF1
	PD14	AF1

5.2.5.1.2.5 TIM5 复用功能 I/O 重映射

表 5-8 TIM5 复用功能 I/O 重映射

复用功能	IO	重映射
TIM5_ETR	PA15	AF1
	PC8	AF1
TIM5_CH1	PA0	AF2
	PA12	AF4
	PA15	AF4
	PB5	AF2
TIM5_CH2	PA1	AF1
	PA12	AF5
	PA15	AF5
	PB5	AF1
TIM5_CH3	PA2	AF2
	PB4	AF2
	PC8	AF3
	PD13	AF1
TIM5_CH4	PA3	AF2
	PB4	AF1
	PC8	AF4
	PD13	AF3

5.2.5.1.3 UARTx 复用功能 I/O 重映射

5.2.5.1.3.1 UART1 复用功能 I/O 重映射

表 5-9 UART1 复用功能 I/O 重映射

复用功能	IO	重映射
UART1_TX	PA2	AF5
	PA9	AF5
	PB10	AF2
	PD10	AF4
UART1_RX	PA3	AF5
	PA10	AF5
	PB11	AF2
	PD11	AF4

5.2.5.1.3.2 UART2 复用功能 I/O 重映射

表 5-10 UART2 复用功能 I/O 重映射

复用功能	IO	重映射
UART2_TX	PA6	AF4
	PB4	AF6
	PB12	AF6

复用功能	IO	重映射
	PC2	AF6
UART2_RX	PA7	AF4
	PB5	AF4
	PB11	AF3
	PC3	AF2

5.2.5.1.3.3 UART3 复用功能 I/O 重映射

表 5-11 UART3 复用功能 I/O 重映射

复用功能	IO	重映射
UART3_TX	PA9	AF6
	PB8	AF5
	PC10	AF5
	PD13	AF5
UART3_RX	PA8	AF5
	PB9	AF4
	PC9	AF5
	PD13	AF4

5.2.5.1.3.4 UART4 复用功能 I/O 重映射

表 5-12 UART4 复用功能 I/O 重映射

复用功能	IO	重映射
UART4_TX	PA9	AF7
	PA14	AF5
	PD5	AF4
	PD10	AF5
UART4_RX	PA10	AF8
	PA13	AF5
	PD4	AF5
	PD11	AF5

5.2.5.1.3.5 UART5 复用功能 I/O 重映射

表 5-13 UART5 复用功能 I/O 重映射

复用功能	IO	重映射
UART5_TX	PB8	AF4
	PB14	AF4
	PC12	AF5
	PD8	AF5
UART5_RX	PB9	AF8
	PB15	AF4
	PD5	AF6
	PD9	AF6

5.2.5.1.4 I2Cx 复用功能 I/O 重映射

5.2.5.1.4.1 I2C1 复用功能 I/O 重映射

表 5-14 I2C1 复用功能 I/O 重映射

复用功能	IO	重映射
I2C1_SCL	PA4	AF6
	PA8	AF7
	PA12	AF6
	PB8	AF6
	PB10	AF6
	PC0	AF6
	PC5	AF6
	PC6	AF6
	PC13	AF6
	PD12	AF6
	PD15	AF6
I2C1_SDA	PA3	AF6
	PA8	AF8
	PA10	AF6
	PB9	AF6
	PB11	AF6
	PC3	AF6
	PC6	AF7
	PC7	AF6
	PD11	AF6
	PD13	AF6
	PD14	AF6
I2C1_SMBA	PA2	AF1
	PB7	AF6
	PC4	AF1
	PD4	AF6

5.2.5.1.4.2 I2C2 复用功能 I/O 重映射

表 5-15 I2C2 复用功能 I/O 重映射

复用功能	IO	重映射
I2C2_SCL	PA8	AF3
	PA12	AF7
	PB8	AF7
	PB10	AF7
	PC0	AF7
	PC5	AF7
	PC6	AF5
	PC10	AF6

复用功能	IO	重映射
	PC13	AF7
	PD12	AF7
I2C2_SDA	PA8	AF2
	PA10	AF7
	PB9	AF7
	PB11	AF7
	PC11	AF7
	PC3	AF7
	PC6	AF0
	PC7	AF7
	PD11	AF7
	PD13	AF7
I2C2_SMBA	PA13	AF4
	PB13	AF0
	PC8	AF6
	PC9	AF6

5.2.5.1.5 SPIx 复用功能 I/O 重映射

5.2.5.1.5.1 SPI1 复用功能 I/O 重映射

表 5-16 SPI1 管脚重映射

复用功能	IO	重映射
SPI1_NSS	PB9	AF0
	PD0	AF0
	PD7	AF0
SPI1_SCK	PA12	AF0
	PB10	AF0
	PB12	AF0
	PC0	AF0
	PC5	AF0
	PD6	AF0
SPI1_MISO	PA9	AF0
	PB11	AF0
	PB12	AF5
	PC2	AF0
	PC7	AF0
	PD5	AF5
SPI1_MOSI	PA10	AF0
	PB10	AF5
	PB11	AF5
	PC3	AF0
	PC6	AF1
	PD4	AF0

5.2.5.1.5.2 SPI2 复用功能 I/O 重映射

表 5-17 SPI2 管脚重映射

复用功能	IO	重映射
SPI2_NSS	PB9	AF5
	PC1	AF0
	PC4	AF0
SPI2_SCK	PB10	AF1
	PC0	AF1
	PC5	AF1
	PC13	AF0
SPI2_MISO	PB12	AF1
	PC2	AF1
	PC3	AF1
	PC7	AF1
	PD10	AF1

SPI2_MOSI	PB11	AF1
	PC2	AF5
	PC3	AF5
	PC6	AF4
	PD11	AF0

5.2.5.1.5.3 SPI3 复用功能 I/O 重映射

表 5-18 SPI3 管脚重映射

复用功能	IO	重映射
SPI3_NSS	PA7	AF0
	PC1	AF5
	PD2	AF0
	PD12	AF0
SPI3_SCK	PA8	AF6
	PB10	AF4
	PC0	AF4
	PC5	AF4
SPI3_MISO	PA1	AF0
	PA7	AF8
	PB12	AF4
	PC2	AF4
	PC7	AF4
	PD13	AF0
SPI3_MOSI	PA0	AF4
	PA9	AF4
	PB11	AF4
	PC3	AF4
	PC6	AF3

5.2.5.1.6 CAN 复用功能重映射

表 5-19 CAN 复用功能重映射

复用功能	管脚	重映射
CAN_RX	PA3	AF7
	PA12	AF8
	PD8	AF7
CAN_TX	PA4	AF7
	PA11	AF8
	PB3	AF4
	PD9	AF7

5.2.5.1.7 COMPx 复用功能 I/O 重映射

表 5-20 COMPx 复用功能 I/O 重映射

复用功能	IO	重映射
COMP1_OUT	PB6	AF5
COMP1_OUT	PB8	AF8
COMP2_OUT	PA7	AF6
COMP2_OUT	PB9	AF9
COMP3_OUT	PA10	AF9
COMP3_OUT	PB4	AF5
COMP4_OUT	PC5	AF5
COMP4_OUT	PC10	AF4

5.2.5.1.8 BEEPER 复用功能 I/O 重映射

表 5-21 BEEPER 复用功能 I/O 重映射

复用功能	IO	重映射
BEEPER_OUT1	PA13	AF7
BEEPER_OUT2	PB14	AF7
BEEPER_OUT3	PC11	AF6

5.2.5.1.9 RTC 复用功能 I/O 重映射

表 5-22 RTC 复用功能 I/O 重映射

复用功能	IO	重映射
TAMPER1	PC13	AF3
TAMPER2	PA0	AF7
TAMPER3	PB8	AF3
RTC_REFCLK	PA1/PB15	AF7

注：TAMPER 引脚需外接上/下拉电阻。

5.2.5.1.10 LCD 复用功能 I/O 重映射

表 5-23 LCD 复用功能 I/O 重映射

复用功能	IO	重映射
LCD_COM0	PD4	AF9
LCD_COM1	PD5	AF9
LCD_COM2	PD6	AF9
LCD_COM3/ LCD_SEG36	PD7	AF8
LCD_COM4/ LCD_SEG35	PD8	AF8
LCD_COM5/ LCD_SEG34	PD9	AF8
LCD_COM6/ LCD_SEG33	PD10	AF8
LCD_COM7/ LCD_SEG32	PD11	AF8
LCD_SEG0	PB0	AF7
LCD_SEG1	PB1	AF7

LCD_SEG2	PB2	AF7
LCD_SEG3	PB10	AF9
LCD_SEG4	PB11	AF9
LCD_SEG5	PB12	AF9
LCD_SEG6	PD0	AF7
LCD_SEG7	PA14	AF8
LCD_SEG8	PA13	AF8
LCD_SEG9	PB14	AF9
LCD_SEG10	PB15	AF9
LCD_SEG11	PC8	AF8
LCD_SEG12	PA15	AF8
LCD_SEG13	PB4	AF9
LCD_SEG14	PB5	AF9
LCD_SEG15	PB6	AF9
LCD_SEG16	PB8	AF9
LCD_SEG17	PB9	AF10
LCD_SEG18	PC4	AF8
LCD_SEG19	PB7	AF9
LCD_SEG20	PC11	AF8
LCD_SEG21	PC10	AF8
LCD_SEG22	PC9	AF8
LCD_SEG23	PC5	AF8
LCD_SEG24	PC6	AF8
LCD_SEG25	PC7	AF8
LCD_SEG26	PC2	AF8
LCD_SEG27	PC3	AF8
LCD_SEG28	PC0	AF8
LCD_SEG29	PC1	AF8
LCD_SEG30	PC12	AF8
LCD_SEG31	PC13	AF8

5.2.5.1.11 EVENT_OUT 复用功能 I/O 重映射

表 5-24 EVENTOUT 复用功能 I/O 重映射

复用功能	IO	重映射
EVENT_OUT	PA1	AF3
	PA6	AF3
	PA7	AF3
	PA11	AF3
	PA12	AF3
	PA15	AF3
	PB0	AF3
	PB3	AF3
	PB4	AF3
	PB9	AF3
	PB10	AF3
	PB12	AF3
	PC0	AF3
	PC1	AF3
	PC2	AF3
	PC3	AF3
	PC4	AF3
	PD4	AF4
	PD5	AF3

5.2.5.1.12 MCO 复用功能 I/O 重映射

5.2.5.1.12.1 MCO 复用功能 I/O 重映射

表 5-25 RCC 复用功能 I/O 重映射

复用功能	IO	重映射
MCO	PA13	AF6
	PB3	AF6
	PB14	AF6

5.2.5.1.13 ADC 外部触发复用功能重映射

ADC 规则转换的外部触发源支持 PX0~PX15(X=A、B、C、D)。

5.2.6 外设的 IO 配置

表 5-26 ADC

ADC	GPIO配置
ADC	模拟模式

表 5-27 LED

LED引脚	配置	GPIO配置
LED_COMx (x=0~9)	LED公共端输出	模拟模式
LED_SEGx (x=0~15)	LED区段输出	模拟模式
LED_GCLK	LED时钟输入	推挽复用

表 5-28 LCD

LCD引脚	配置	GPIO配置模式
LCD_COMx (x=0~7)	LCD公共端输出	模拟模式+复用功能
LCD_SEGx (x=0~36)	LCD区段输出	模拟模式+复用功能

表 5-29 TIM1

TIM1引脚	配置	GPIO配置模式
TIM1_CHx	输入捕获通道x	推挽复用
	输出比较通道x	推挽复用
TIM1_CHxN	互补输出通道x	推挽复用
TIM1_BKIN	刹车输入	推挽复用
TIM1_ETR	外部触发时钟输入	推挽复用

表 5-30 TIM2、3、4、5

TIM2、3、4、5引脚	配置	GPIO配置模式
TIMx_CHx	输入捕获通道x	推挽复用
	输出比较通道x	推挽复用
TIMx_ETR	外部触发时钟输入	推挽复用

表 5-31 UART

UART引脚	配置	GPIO配置
UARTx_TX	全双工模式	推挽复用
	半双工同步模式	推挽复用+上拉
UARTx_RX	全双工模式	推挽复用
	半双工同步模式	未用，可作为通用I/O

表 5-32 I2C

I2C引脚	配置	GPIO配置
I2Cx_SCL	I2C时钟	开漏复用
I2Cx_SDA	I2C数据	开漏复用

表 5-33 SPI

SPI引脚	配置	GPIO配置
SPIx_SCK	主模式	推挽复用
	从模式	推挽复用
SPIx_MOSI	全双工模式/主模式	推挽复用
	全双工模式/从模式	推挽复用
	单线双向数据线/主模式	推挽复用

SPI引脚	配置	GPIO配置
	单线双向数据线/从模式	未用，可作为通用I/O
SPIx_MISO	全双工模式/主模式	推挽复用
	全双工模式/从模式	推挽复用
	单线双向数据线/主模式	未用，可作为通用I/O
	单线双向数据线/从模式	推挽复用
SPIx_NSS	硬件主/从模式	推挽复用
	软件模式	未用，可作为通用I/O

表 5-34 CAN

CAN引脚	配置	GPIO配置
CAN_TX	CAN发送	推挽复用
CAN_RX	CAN接收	推挽复用

表 5-35 COMP

COMPx引脚	配置	GPIO配置
COMPx_OUT	COMP输出	推挽复用
COMPx_INM	COMP负端输入	模拟模式
COMPx_INP	COMP正端输入	模拟模式

表 5-36 RTC

RTC引脚	IO	重映射
RTC_REFCLK	参考时钟输入	推挽复用

表 5-37 BEEPER

BEEPER引脚	配置	GPIO配置
BEEPER_OUTx	BEEPER输出	推挽复用

表 5-38 其他

引脚	复用功能	GPIO配置
MCO	时钟输出	推挽复用
EXTI输入线	外部中断输入	浮空输入+上/下拉

5.2.7 GPIO 锁定机制

锁定机制用于冻结 IO 配置以防止被意外更改。当在一个端口位上执行了锁定（LOCK）程序，在下一次复位之前，不能再更改端口的配置，参考端口配置锁定寄存器 GPIOx_PLOCK。

- 只有在对 GPIOx_PLOCK.PLOCKK 按照正确的序列 w1->w0->w1->r0（此处 r0 必须有）操作之后，才会变为 1；之后只有在进行系统复位才会变为 0。
- 只有在 GPIOx_PLOCK.PLOCKK = 0 也就是未锁定的时候才能进行 GPIOx_PLOCK.PLOCK[15:0]修改。
- GPIOx_PLOCK.PLOCK 只有在和非 0 的 GPIOx_PLOCK.PLOCK[15:0]同时写入的情况下，序列 w1->w0->w1->r0 才会有效；序列写入的过程中，GPIOx_PLOCK.PLOCK[15:0]必须不能改变；
- 只要 GPIOx_PLOCK.PLOCKK = 0，GPIOx_PMODE / GPIOx_POTYPE / GPIOx_PUPD / GPIOx_AFL /

GPIOx_AFH 的位都能修改，不受 GPIOx_PLOCK.PLOCK[15:0]配置的影响。

- GPIOx_PLOCK.PLOCKK= 1,GPIOx_PMODE/GPIOx_POTYPE/GPIOx_PUPD/GPIOx_AFL/GPIOx_AFH 受 GPIOx_PLOCK.PLOCK[15:0]的控制。对应 GPIOx_PLOCK.PLOCKy (y = 0...15) =1 时，为锁定配置，不可修改；PLOCKy = 0，可以修改。
- 假如序列操作错误，必须重新进行 w1->w0->w1->r0 才能再次发起锁定操作。

5.3 GPIO 寄存器

必须以 32 位字的方式操作这些外设寄存器。

5.3.1 GPIO 寄存器总览

GPIOA 基地址：0x40010800

GPIOB 基地址：0x40010C00

GPIOC 基地址：0x40011000

GIOD 基地址：0x40011C00

表 5-39 GPIO 寄存器总览

Offset	Register		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
000h	GPIOx_PMODE	x=A,B, C,D	PMODE15[1:0]		PMODE14[1:0]		PMODE13[1:0]		PMODE12[1:0]		PMODE11[1:0]		PMODE10[1:0]		PMODE9[1:0]		PMODE8[1:0]		PMODE7[1:0]		PMODE6[1:0]		PMODE5[1:0]		PMODE4[1:0]		PMODE3[1:0]		PMODE2[1:0]		PMODE1[1:0]		PMODE0[1:0]				
	Reset	x=A	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
	Value	x=B,C, D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
004h	GPIOx_POTYPE	x=A,B, C,D	Reserved																	POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0		
	Reset	x=A,B, C,D	Reserved																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Value	C,D	Reserved																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
080h	GPIOx_SR	x=A,B, C,D	Reserved																	SR15	SR16	SR17	SR18	SR19	SR20	SR21	SR22	SR23	SR24	SR25	SR26	SR27	SR28	SR29	SR30		
	Reset	x=A,B, C,D	Reserved																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Value	C,D	Reserved																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
00Ch	GPIOx_PUPD	x=A,B, C,D	PUPD15		PUPD14		PUPD13		PUPD12		PUPD11		PUPD10		PUPD9		PUPD8		PUPD7		PUPD6		PUPD5		PUPD4		PUPD3		PUPD2		PUPD1		PUPD0				
	Reset	x=A	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	Value	x=B,C, D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
010h	GPIOx_PID	x=A,B, C,D	Reserved																	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0		
	Reset	x=A,B, C,D	Reserved																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Value	C,D	Reserved																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路 109 号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationstech.com 邮编：518057

位域	名称	描述
31:30	PMODEy [1:0]	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的模式: 00: 输入模式 01: 通用输出模式 10: 复用功能模式 11: 模拟功能模式
29:28		
27:26		
25:24		
23:22		
21:20		
19:18		
17:16		
15:14		
13:12		
11:10		
9:8		
7:6		
5:4		
3:2		
1:0		

5.3.3 GPIO 端口输出类型寄存器 (GPIOx_POTYPE)

偏移地址: 0x04

复位值: 0x0000 0000 (x=A,B,C,D)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	POTy	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的输出类型: 0: 输出推挽模式 (复位后的状态) 1: 输出开漏模式

5.3.4 GPIO 翻转率配置寄存器 (GPIOx_SR)

偏移地址: 0x08

复位值: 0x0000 FFFF (x=A,B,C,D)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	SRy	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的翻转率配置位 0: 快速翻转 1: 慢速翻转

5.3.5 GPIO 端口上下拉寄存器 (GPIOx_PUPD)

偏移地址: 0x0C

复位值: 0x2400 0000 (x = A) ; 0x0000 0000 (x = B,C,D)

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

位域	名称	描述
31:30	PUPDy[1:0]	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的上拉下拉模式: 00: 无上/下拉 01: 上拉 10: 下拉 11: 保留
29:28		
27:26		
25:24		
23:22		
21:20		
19:18		
17:16		
15:14		
13:12		
11:10		
9:8		
7:6		
5:4		
3:2		

位域	名称	描述
1:0		

5.3.6 GPIO 端口输入数据寄存器 (GPIOx_PID)

偏移地址: 0x10

复位值: 0x0000 0000 (x= A,B,C,D)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	PIDy	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的输入数据 这些位为只读, 读出的值为对应 I/O 口的状态。

5.3.7 GPIO 端口输出数据寄存器 (GPIOx_POD)

偏移地址: 0x14

复位值: 0x0000 0000 (x= A,B,C,D)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	PODy	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的输出数据 这些位可由软件读出或写入, 可对相应的 POD 位进行独立的设置/清除。

5.3.8 GPIO 端口位设置/清除寄存器 (GPIOx_PBSC)

偏移地址: 0x18

复位值：0x0000 0000 (x= A,B,C,D)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
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
PBS15	PBS14	PBS13	PBS12	PBS11	PBS10	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:16	PBCy	清除端口 GPIOx (x = A,B,C,D) 的位 y 这些位只能写入。 0: 对相应的 PODy 位不产生影响 1: 清除对应的 PODy 位为 0 <i>注: 如果同时设置了 PBCy 和 PBCy 的对应位, PBCy 位起作用。</i>
15:0	PBSy	设置端口 GPIOx (x = A,B,C,D) 的位 y 这些位只能写入。 0: 对相应的 PODy 位不产生影响 1: 设置对应的 PODy 位为 1

5.3.9 GPIO 端口锁定置寄存器 (GPIOx_PLOCK)

偏移地址：0x1C

复位值：0x0000 0000 (x = A,B,C,D)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															PLOCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLOCK15	PLOCK14	PLOCK13	PLOCK12	PLOCK11	PLOCK10	PLOCK9	PLOCK8	PLOCK7	PLOCK6	PLOCK5	PLOCK4	PLOCK3	PLOCK2	PLOCK1	PLOCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:17	Reserved	保留, 必须保持复位值。
16	PLOCKK	锁键 该位可随时读出, 它只可通过锁键写入序列修改。 0: 端口配置锁键位未激活 1: 端口配置锁键位被激活, 下次系统复位前 GPIOx_PLOCK 寄存器被锁住。 锁键写入序列: 写 1 -> 写 0 -> 写 1 -> 读 0 -> (读 1)

位域	名称	描述
		最后一个读 1 可省略，但可以用来确认锁键已被激活。 <i>注：在操作锁键写入序列时，不能改变 PLOCK[15:0] 的值。操作锁键写入序列中的任何错误将不能激活锁键。</i>
15:0	PLOCKy	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的配置锁定位 这些位可读可写但只能在 PLOCKK 位为 0 时写入。 0: 不锁定端口的配置 1: 锁定端口的配置

5.3.10 GPIO 复用功能低配置寄存器 (GPIOx_AFL)

偏移地址: 0x20

复位值: 0xFFFF FFFF (x = A,B,C,D)

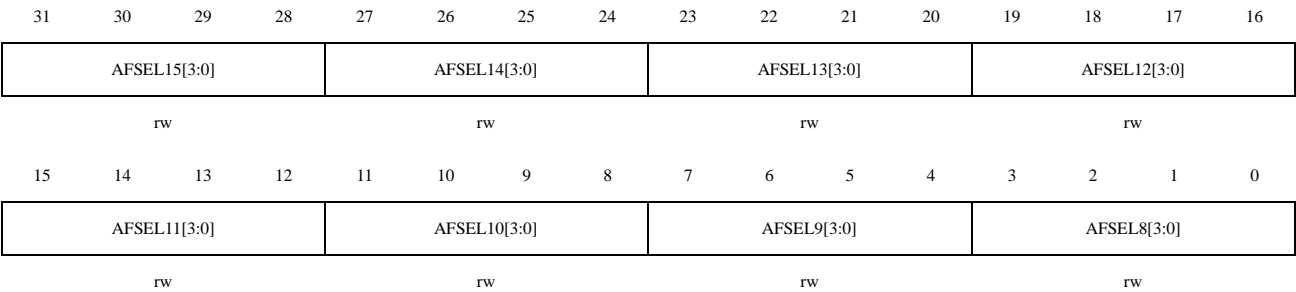
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:28 27:24 23:20 19:16 15:12 11:8 7:4 3:0	AFSELy[3:0]	端口 GPIOx (x = A,B,C,D) 引脚 PINy (y = 0...7) 的复用功能配置位 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 <i>注意: AF15 为 GPIO 无复用。</i>

5.3.11 GPIO 复用功能高配置寄存器（GPIOx_AFH）

偏移地址：0x24

复位值：0xF00F FFFF（x = A）；0xFFFF FFFF（x = B,C,D）

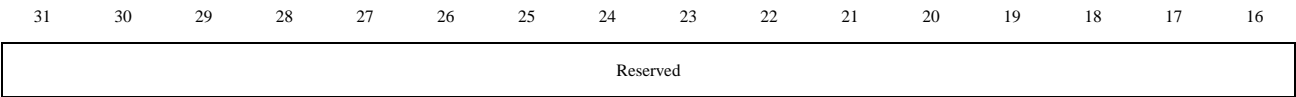


位域	名称	描述
31:28 27:24 23:20 19:16 15:12 11:8 7:4 3:0	AFSELy[3:0]	端口 GPIOx（x = A,B,C,D）引脚 PINy（y = 8...15）的复用功能配置位 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15 <i>注意：AF15 为 GPIO 无复用。</i>

5.3.12 GPIO 端口位清除寄存器（GPIOx_PBC）

偏移地址：0x28

复位值：0x0000 0000（x = A,B,C,D）



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	PBCy	清除端口 GPIOx (x = A,B,C,D) 的位 y 这些位只能写入。 0: 对相应的 PODy 位不产生影响 1: 清除对应的 PODy 位为 0

5.3.13 GPIO 驱动能力配置寄存器 (GPIOx_DS)

偏移地址: 0x2C

复位值: 0x0000 0000 (x = A,B,C,D)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS15	DS14	DS13	DS12	DS11	DS10	DS9	DS8	DS7	DS6	DS5	DS4	DS3	DS2	DS1	DS0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	DSy	端口 GPIOx (x = A,B,C,D) 引脚 PINy 的驱动能力配置位 0: 高驱动能力 (16mA(5V)/8mA(3.3V)/4mA(2.0V)) 1: 低驱动能力 (8mA(5V)/4mA(3.3V)/2mA(2.0V))

5.4 AFIO 寄存器

5.4.1 AFIO 寄存器总览

AFIO 基地址: 0x40010000

表 5-40 AFIO 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	AFIO_CFG	Reserved			TIM3CH2_RMP	Reserved				SPI3_NSS	SPI2_NSS	SPI1_NSS	Reserved				EXTI_ETRR[3:0]				Reserved					IOFLTCFG[4:0]							
	Reset Value																0	0	0	0											0	0	0
008h	AFIO_EXTI_CF G0	Reserved												EXTI3_CFG[3:0]				EXTI2_CFG[3:0]				EXTI1_CFG[3:0]				EXTI0_CFG[3:0]							
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	AFIO_EXTI_CF G1	Reserved												EXTI7_CFG[3:0]				EXTI6_CFG[3:0]				EXTI5_CFG[3:0]				EXTI4_CFG[3:0]							
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	AFIO_EXTI_CF G2	Reserved												EXTI11_CFG[3:0]				EXTI10_CFG[3:0]				EXTI9_CFG[3:0]				EXTI8_CFG[3:0]							
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	AFIO_EXTI_CF G3	Reserved												EXTI15_CFG[3:0]				EXTI14_CFG[3:0]				EXTI13_CFG[3:0]				EXTI12_CFG[3:0]							
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	AFIO_DIGEFG T_CFG1	PB15DIGEFEN	PB14DIGEFEN	PB13DIGEFEN	PB12DIGEFEN	PB11DIGEFEN	PB10DIGEFEN	PB9DIGEFEN	PB8DIGEFEN	PB7DIGEFEN	PB6DIGEFEN	PB5DIGEFEN	PB4DIGEFEN	PB3DIGEFEN	PB2DIGEFEN	PB1DIGEFEN	PB0DIGEFEN	PA15DIGEFEN	PA14DIGEFEN	PA13DIGEFEN	PA12DIGEFEN	PA11DIGEFEN	PA10DIGEFEN	PA9DIGEFEN	PA8DIGEFEN	PA7DIGEFEN	PA6DIGEFEN	PA5DIGEFEN	PA4DIGEFEN	PA3DIGEFEN	PA2DIGEFEN	PA1DIGEFEN	PA0DIGEFEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	AFIO_DIGEFG T_CFG2	PD15DIGEFEN	PD14DIGEFEN	PD13DIGEFEN	PD12DIGEFEN	PD11DIGEFEN	PD10DIGEFEN	PD9DIGEFEN	PD8DIGEFEN	PD7DIGEFEN	PD6DIGEFEN	PD5DIGEFEN	PD4DIGEFEN	Reserved	PD2DIGEFEN	PD1DIGEFEN	PD0DIGEFEN	PC15DIGEFEN	PC14DIGEFEN	PC13DIGEFEN	PC12DIGEFEN	PC11DIGEFEN	PC10DIGEFEN	PC9DIGEFEN	PC8DIGEFEN	PC7DIGEFEN	PC6DIGEFEN	PC5DIGEFEN	PC4DIGEFEN	PC3DIGEFEN	PC2DIGEFEN	PC1DIGEFEN	PC0DIGEFEN
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.2 AFIO 配置寄存器（AFIO_CFG）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TIM3CH2_RMP	Reserved				SPI3_NSS	SPI2_NSS	SPI1_NSS	Reserved				
rw				rw				rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EXTI_ETRR[3:0]				Reserved						IOFLITCFG[4:0]				
rw				rw											
位域			名称				描述								
31:24			Reserved				保留，必须保持复位值。								

位域	名称	描述
28	TIM3CH2_RMP	TIM3 通道 2 内部重映射 0: TIM3_CH2 与 IO 相连 1: TIM3_CH2 与 LSI 相连
27:24	Reserved	保留, 必须保持复位值。
23	SPI3_NSS	SPI3 的 NSS 模式选择位 (NSS 配置为 AFIO 推挽模式)。 0: NSS 空闲时为高阻态 1: NSS 空闲时为高电平
22	SPI2_NSS	SPI2 的 NSS 模式选择位 (NSS 配置为 AFIO 推挽模式)。 0: NSS 空闲时为高阻态 1: NSS 空闲时为高电平
21	SPI1_NSS	SPI1 的 NSS 模式选择位 (NSS 配置为 AFIO 推挽模式)。 0: NSS 空闲时为高阻态 1: NSS 空闲时为高电平
20:15	Reserved	保留, 必须保持复位值。
14:11	EXTI_ETRR[3:0]	选择中断线规则转换外部触发重映射 0000: 选择 EXTI0 规则转换外部触发 0001: 选择 EXTI1 规则转换外部触发 1111: 选择 EXTI15 规则转换外部触发
10:5	Reserved	保留, 必须保持复位值。
4:0	IOFLITCFG[4:0]	IO 滤波控制 00000: 旁路滤波器 00001: IO 滤波时间为 1 个 APB2 时钟周期 00010: IO 滤波时间为 2 个 APB2 时钟周期 11111: IO 滤波时间为 31 个 APB2 时钟周期

5.4.3 AFIO 外部中断配置寄存器 0 (AFIO_EXTI_CFG0)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_CFG[3:0]				EXTI2_CFG[3:0]				EXTI1_CFG[3:0]				EXTI0_CFG[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	EXTIx_CFG[3:0]	<p>EXTIx 配置 (x = 0 ... 3)</p> <p>这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。</p> <p>EXTI0 配置</p> <p>0001: PA0 引脚 0010: PB0 引脚 0100: PC0 引脚</p> <p>1000: PD0 引脚</p> <p>EXTI1 配置</p> <p>0001: PA1 引脚 0010: PB1 引脚 0100: PC1 引脚</p> <p>1000: 保留</p> <p>EXTI2 配置</p> <p>0001: PA2 引脚 0010: PB2 引脚 0100: PC2 引脚</p> <p>1000: 保留</p> <p>EXTI3 配置</p> <p>0001: PA3 引脚 0010: PB3 引脚 0100: PC3 引脚</p> <p>1000: 保留</p> <p>注意：不可组合使用，例如：PA0,PB0,PC0,PD0 引脚只能使用其中一个触发 EXTI0 中断</p>

5.4.4 AFIO 外部中断配置寄存器 1 (AFIO_EXTI_CFG1)

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7_CFG[3:0]				EXTI6_CFG[3:0]				EXTI5_CFG[3:0]				EXTI4_CFG[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	EXTIx_CFG[3:0]	<p>EXTIx 配置 (x = 4 ... 7)</p> <p>这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。</p> <p>EXTI4 配置</p> <p>0001: PA4 引脚 0010: PB4 引脚 0100: PC4 引脚</p> <p>1000: PD4 引脚</p> <p>EXTI5 配置</p> <p>0001: PA5 引脚 0010: PB5 引脚 0100: PC5 引脚</p> <p>1000: PD5 引脚</p> <p>EXTI6 配置</p>

位域	名称	描述
		0001: PA6 引脚 0010: PB6 引脚 0100: PC6 引脚 1000: PD6 引脚 EXTI7 配置 0001: PA7 引脚 0010: PB7 引脚 0100: PC7 引脚 1000: PD7 引脚 注意: 不可组合使用, 例如: PA4,PB4,PC4,PD4 引脚只能使用其中一个触发 EXTI4 中断

5.4.5 AFIO 外部中断配置寄存器 2 (AFIO_EXTI_CFG2)

偏移地址: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_CFG[3:0]				EXTI10_CFG[3:0]				EXTI9_CFG[3:0]				EXTI8_CFG[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	EXTIx_CFG[3:0]	EXTIx 配置 (x = 8... 11) 这些位可由软件读写, 用于选择 EXTIx 外部中断的输入源。 EXTI8 配置 0001: PA8 引脚 0010: PB8 引脚 0100: PC8 引脚 1000: PD8 引脚 EXTI9 配置 0001: PA9 引脚 0010: PB9 引脚 0100: PC9 引脚 1000: PD9 引脚 EXTI10 配置 0001: PA10 引脚 0010: PB10 引脚 0100: PC10 引脚 1000: PD10 引脚 EXTI11 配置 0001: PA11 引脚 0010: PB11 引脚 0100: PC11 引脚 1000: PD11 引脚 注意: 不可组合使用, 例如: PA8,PB8,PC8,PD8 引脚只能使用其中一个触发 EXTI8 中断

5.4.6 AFIO 外部中断配置寄存器 3 (AFIO_EXTI_CFG3)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_CFG[3:0]				EXTI14_CFG[3:0]				EXTI13_CFG[3:0]				EXTI12_CFG[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	EXTIx_CFG[3:0]	<p>EXTIx 配置 (x = 12 ... 15)</p> <p>这些位可由软件读写, 用于选择 EXTIx 外部中断的输入源。</p> <p>这些位可由软件读写, 用于选择 EXTIx 外部中断的输入源。</p> <p>EXTI12 配置</p> <p>0001: PA12 引脚 0010: PB12 引脚 0100: PC12 引脚</p> <p>1000: PD12 引脚</p> <p>EXTI13 配置</p> <p>0001: PA13 引脚 0010: PB13 引脚 0100: PC13 引脚</p> <p>1000: PD13 引脚</p> <p>EXTI14 配置</p> <p>0001: PA14 引脚 0010: PB14 引脚 0100: PC14 引脚</p> <p>1000: PD14 引脚</p> <p>EXTI15 配置</p> <p>0001: PA15 引脚 0010: PB15 引脚 0100: PC15 引脚</p> <p>1000: PD15 引脚</p> <p>注意: 不可组合使用, 例如: PA12,PB12,PC12,PD12 引脚只能使用其中一个触发 EXTI12 中断</p>

5.4.7 数字毛刺滤波器配置寄存器 1 (AFIO_DIGEFT_CFG1)

偏移地址: 0x18

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PB15DIG EFTEN	PB14DIG EFTEN	PB13DIG EFTEN	PB12DIG EFTEN	PB11DIG EFTEN	PB10DIG EFTEN	PB9DIG EFTEN	PB8DIG EFTEN	PB7DIG EFTEN	PB6DIG EFTEN	PB5DIG EFTEN	PB4DIG EFTEN	PB3DIG EFTEN	PB2DIG EFTEN	PB1DIG EFTEN	PB0DIG EFTEN
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

PA15DIG EFTEN	PA14DIG EFTEN	PA13DIG EFTEN	PA12DIG EFTEN	PA11DIG EFTEN	PA10DIG EFTEN	PA9DIG EFTEN	PA8DIG EFTEN	PA7DIG EFTEN	PA6DIG EFTEN	PA5DIG EFTEN	PA4DIG EFTEN	PA3DIG EFTEN	PA2DIG EFTEN	PA1DIG EFTEN	PA0DIG EFTEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	PByDIGEFTEN	引脚 PBy 数字滤波使能位(y=0...15) 0: 失能 EFT 1: 使能 EFT
15:0	PAyDIGEFTEN	引脚 PAy 数字滤波使能位(y=0...15) 0: 失能 EFT 1: 使能 EFT

5.4.8 数字毛刺滤波器配置寄存器 2 (AFIO_DIGEFT_CFG2)

偏移地址: 0x1C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD15DIG EFTEN	PD14DIG EFTEN	PD13DIG EFTEN	PD12DIG EFTEN	PD11DIG EFTEN	PD10DIG EFTEN	PD9DIG EFTEN	PD8DIG EFTEN	PD7DIG EFTEN	PD6DIG EFTEN	PD5DIG EFTEN	PD4DIG EFTEN	Reserved	PD2DIG EFTEN	PD1DIG EFTEN	PD0DIG EFTEN
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
PC15DIG EFTEN	PC14DIG EFTEN	PC13DIG EFTEN	PC12DIG EFTEN	PC11DIG EFTEN	PC10DIG EFTEN	PC9DIG EFTEN	PC8DIG EFTEN	PC7DIG EFTEN	PC6DIG EFTEN	PC5DIG EFTEN	PC4DIG EFTEN	PC3DIG EFTEN	PC2DIG EFTEN	PC1DIG EFTEN	PC0DIG EFTEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	PDyDIGEFTEN	引脚 PDy 数字滤波使能位(y=0,4,5...15) 0: 失能 EFT 1: 使能 EFT
15:0	PCyDIGEFTEN	引脚 PCy 数字滤波使能位(y=0...15) 0: 失能 EFT 1: 使能 EFT

6 中断和事件

6.1 嵌套向量中断寄存器

特性

- 32 个可屏蔽中断通道（不包含 16 个 Cortex®-M0 的中断线）。
- 4 个可编程的优先等级（使用了 2 位中断优先级）；
- 低延迟的异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现；

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

6.1.1 SysTick 校准值寄存器

系统嘀嗒校准值固定为 8000，当系统嘀嗒时钟设定为 8MHz（HCLK/8 的最大值），产生 1ms 时间基准。

6.1.2 中断和异常向量

表 6-1 向量表

位置	优先级	优先级类型	名称	说明	地址
-	-	-	-	保留(Reserved)	0x0000 0000
-	-3	固定	Reset	复位(Reset)	0x0000 0004
-	-2	固定	NMI	不可屏蔽中断。RCC时钟安全系统(CSS) 连接到NMI向量。	0x0000 0008
-	-1	固定	HardFault	所有类型的错误(fault)	0x0000 000C
-	3	可设置	SVCall	通过SWI指令调用的系统服务	0x0000 002C
-	5	可设置	PendSV	可挂起的系统服务请求	0x0000 0038
-	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	WWDG	窗口看门狗中断	0x0000 0040
1	8	可设置	PVD	PVD中断（联接EXTI线16）	0x0000 0044
2	9	可设置	RTC	RTC中断（联接EXTI线17、18、19）	0x0000 0048
3	10	可设置	MMU_RAMC_ERR	MMU/RAMC_ERR全局中断	0x0000 004C
4	11	可设置	FLASH	Flash全局中断	0x0000 0050
5	12	可设置	RCC	RCC全局中断	0x0000 0054
6	13	可设置	EXTI0_1	EXTI线[1:0] 中断	0x0000 0058
7	14	可设置	EXTI2_3	EXTI 线[3:2] 中断	0x0000 005C
8	15	可设置	EXTI4_15	EXTI线[15:4] 中断	0x0000 0060

位置	优先级	优先级类型	名称	说明	地址
9	16	可设置	SAC	SAC中断	0x0000 0064
10	17	可设置	DMA_CH1_2	DMA通道1/2中断	0x0000 0068
11	18	可设置	DMA_CH3_4_5	DMA通道3/4/5中断	0x0000 006C
12	19	可设置	TIM4	TIM4全局中断	0x0000 0070
13	20	可设置	TIM1_BRK_UP_TRG_COM	TIM1刹车、更新、触发和通信中断	0x0000 0074
14	21	可设置	TIM1_CC	TIM1捕获比较中断	0x0000 0078
15	22	可设置	CAN	CAN中断	0x0000 007C
16	23	可设置	TIM3	TIM3全局中断	0x0000 0080
17	24	可设置	UART3_4	UART3/4全局中断	0x0000 0084
18	25	可设置	TIM5	TIM5全局中断	0x0000 0088
19	26	可设置	TIM6	TIM6全局中断，连接EXTI20	0x0000 008C
20	27	可设置	TIM2	TIM2全局中断	0x0000 0090
21	28	可设置	ADC	ADC全局中断	0x0000 0094
22	29	可设置	SPI2	SPI2全局中断	0x0000 0098
23	30	可设置	I2C1	I2C1全局中断	0x0000 009C
24	31	可设置	I2C2	I2C2全局中断	0x0000 00A0
25	32	可设置	SPI1	SPI1全局中断	0x0000 00A4
26	33	可设置	UART1	UART1全局中断	0x0000 00A8
27	34	可设置	SPI3	SPI3全局中断	0x0000 00AC
28	35	可设置	UART5	UART5全局中断	0x0000 00B0
29	36	可设置	LCD	LCD全局中断	0x0000 00B4
30	37	可设置	UART2	UART2全局中断	0x0000 00B8
31	38	可设置	COMP1_2_3_4	COMP1/2/3/4全局中断	0x0000 00BC

6.2 外部中断/事件控制器（EXTI）

6.2.1 简介

外部中断/事件控制器包含 21 个产生中断/事件触发的边沿检测电路，每条输入线可以独立地配置脉冲或挂起输入类型，以及上升沿、下降沿或者双边沿 3 种触发事件类型，也可以独立地被屏蔽。挂起寄存器保持着状态线的中断请求，可通过在挂起寄存器的对应位写‘1’操作，清除中断请求。

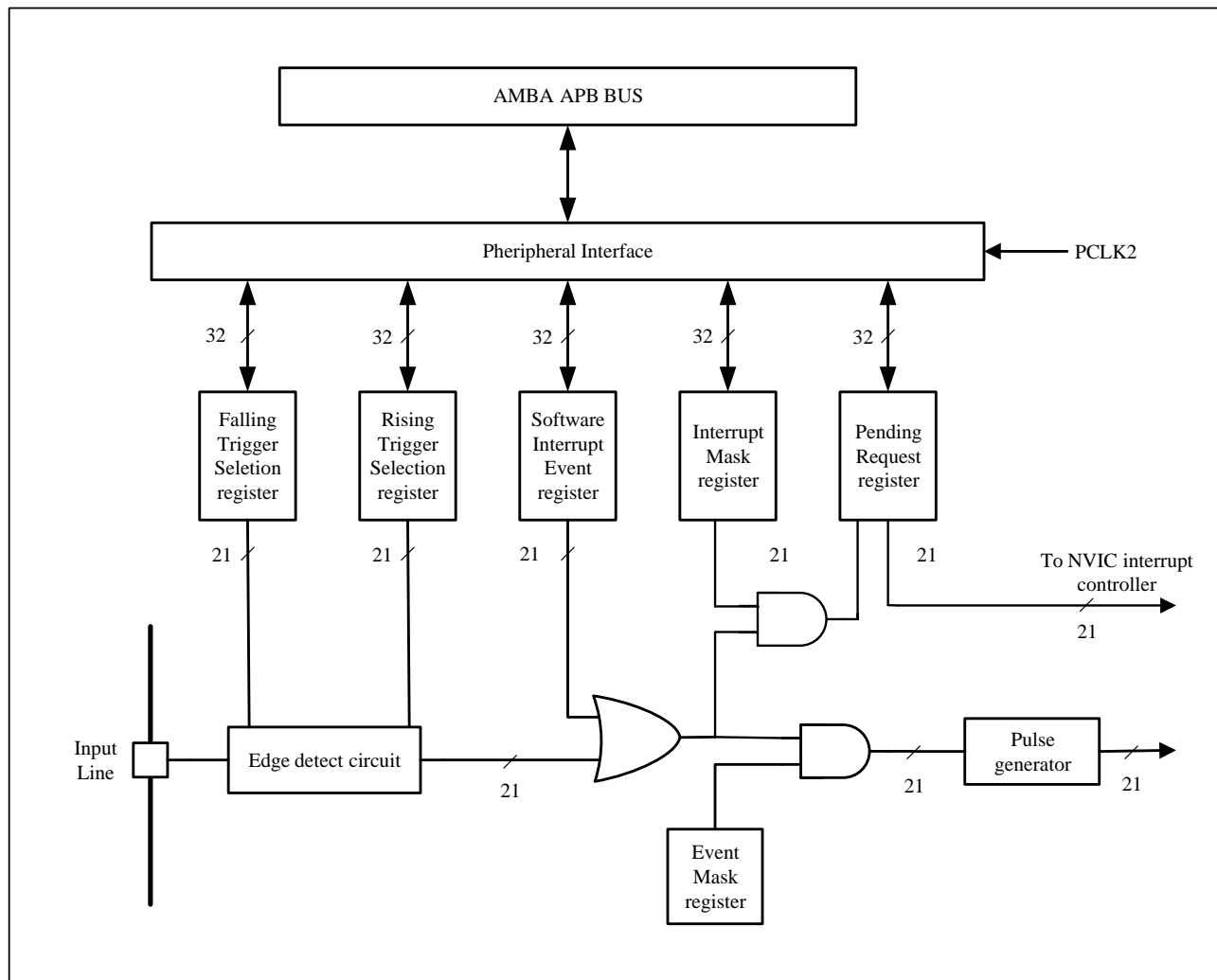
6.2.2 主要特性

EXTI 控制器的主要特性如下：

- 支持 21 个软件中断/事件请求
- 每条输入线对应的中断/事件都能独立配置触发或屏蔽
- 每条中断线都有独立的状态位
- 支持脉冲或挂起输入类型
- 支持上升沿、下降沿或双边沿 3 种触发事件类型

■ 可唤醒退出低功耗模式

图 6-1 外部中断/事件控制器框图



6.2.3 功能描述

EXTI 包含 21 条中断线，其中 16 条来自 I/O 管脚，另 5 条来自内部模块。要产生中断，必须配置外部中断控制器的 NVIC 中断通道使能相应的中断线。通过沿触发配置寄存器 `EXTI_RT_CFG` 和 `EXTI_FT_CFG` 选择上升沿、下降沿或双边沿触发事件类型，并将中断屏蔽寄存器 `EXTI_IMASK` 的相应位写‘1’开放允许中断请求。当外部中断线上检测到预设的边沿触发极性，将产生一个中断请求，对应的挂起位也随之被置‘1’。在挂起寄存器的对应位写‘1’，将清除该中断请求。

要产生事件，必须配置并使能对应的事件线。根据需要的边沿检测极性，设置上升/下降沿触发配置寄存器，同时在事件屏蔽寄存器的相应位写‘1’允许中断请求。当事件线上发生预设的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置‘1’。

另外，通过在软件中断/事件寄存器写‘1’，也可以通过软件产生中断/事件请求。

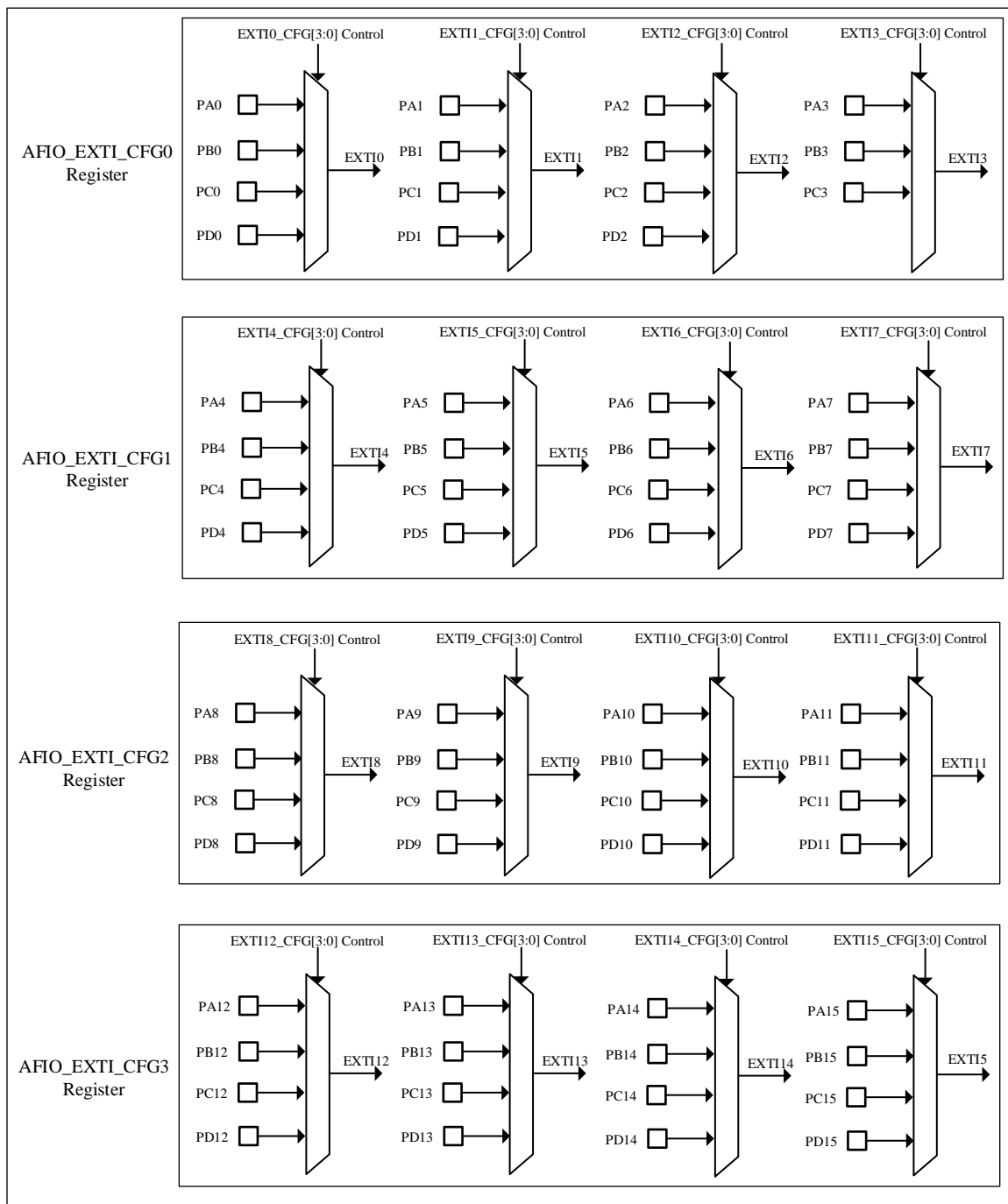
■ 硬件中断配置，根据需要选择配置 21 条线路作为中断源：

- ◆ 配置 21 条中断线的屏蔽位（`EXTI_IMASK`）；
- ◆ 配置所选中断线的触发配置位（`EXTI_RT_CFG` 和 `EXTI_FT_CFG`）；

- ◆ 配置对应到外部中断控制器的 NVIC 中断通道的使能和屏蔽位，使 21 条中断线中的请求可以被正确地响应。
- 硬件事件配置，根据需要选择配置 21 条线路作为事件源：
 - ◆ 配置 21 条事件线的屏蔽位（EXTI_EMASK）；
 - ◆ 配置所选事件线的触发配置位（EXTI_RT_CFG 和 EXTI_FT_CFG）。
- 软件中断/事件配置，根据需要选择配置 21 条线路作为软件中断/事件线：
 - ◆ 配置 21 条中断/事件线屏蔽位（EXTI_IMASK,EXTI_EMASK）；
 - ◆ 配置软件中断事件寄存器的请求位（EXTI_SWIE）。

6.2.4 EXTI 线路映射

图 6-2 外部中断通用 I/O 映射



通过 AFIO_EXTI_CFG1~4 配置 GPIO 线上的外部中断/事件，必须先使能 AFIO 时钟。通用 I/O 端口以上图的方式连接到 16 条外部中断/事件线上。另外 5 条 EXTI 线的连接方式如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 18 连接到 RTC 入侵检测或 RTC 时间戳唤醒事件

- ### 6.3 EXTI 寄存器

6.3.1 EXTI 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000h	EXTL_EMASK	Reserved												EMASK20	EMASK19	EMASK18	EMASK17	EMASK16	EMASK15	EMASK14	EMASK13	EMASK12	EMASK11	EMASK10	EMASK9	EMASK8	EMASK7	EMASK6	EMASK5	EMASK4	EMASK3	EMASK2	EMASK1	EMASK0
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	EXTL_IMASK	Reserved												IMASK20	IMASK19	IMASK18	IMASK17	IMASK16	IMASK15	IMASK14	IMASK13	IMASK12	IMASK11	IMASK10	IMASK9	IMASK8	IMASK7	IMASK6	IMASK5	IMASK4	IMASK3	IMASK2	IMASK1	IMASK0
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	EXTL_FT_CFG	Reserved												FT_CFG20	FT_CFG19	FT_CFG18	FT_CFG17	FT_CFG16	FT_CFG15	FT_CFG14	FT_CFG13	FT_CFG12	FT_CFG11	FT_CFG10	FT_CFG9	FT_CFG8	FT_CFG7	FT_CFG6	FT_CFG5	FT_CFG4	FT_CFG3	FT_CFG2	FT_CFG1	FT_CFG0
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	EXTL_RT_CFG	Reserved												RT_CFG20	RT_CFG19	RT_CFG18	RT_CFG17	RT_CFG16	RT_CFG15	RT_CFG14	RT_CFG13	RT_CFG12	RT_CFG11	RT_CFG10	RT_CFG9	RT_CFG8	RT_CFG7	RT_CFG6	RT_CFG5	RT_CFG4	RT_CFG3	RT_CFG2	RT_CFG1	RT_CFG0
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	EXTL_PEND	Reserved												PEND20	PEND19	PEND18	PEND17	PEND16	PEND15	PEND14	PEND13	PEND12	PEND11	PEND10	PEND9	PEND8	PEND7	PEND6	PEND5	PEND4	PEND3	PEND2	PEND1	PEND0
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	EXTL_SWIE	Reserved												SWIE20	SWIE19	SWIE18	SWIE17	SWIE16	SWIE15	SWIE14	SWIE13	SWIE12	SWIE11	SWIE10	SWIE9	SWIE8	SWIE7	SWIE6	SWIE5	SWIE4	SWIE3	SWIE2	SWIE1	SWIE0
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	EXTL_TS_SEL	Reserved																								TSSEL[3:0]								
	0																									0	0	0						

6.3.2 EXTI 事件屏蔽寄存器 (EXTI EMASK)

复位值: 0x0000 0000

[illegible]

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:0	EMASKx	线 x 上的事件屏蔽（x 表示 0, 1, 2...19, 20） 0: 屏蔽来自线 x 上的事件请求； 1: 开放来自线 x 上的事件请求。

6.3.3 EXTI 中断屏蔽寄存器（EXTI_IMASK）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											IMASK20	IMASK19	IMASK18	IMASK17	IMASK16
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMASK15	IMASK14	IMASK13	IMASK12	IMASK11	IMASK10	IMASK9	IMASK8	IMASK7	IMASK6	IMASK5	IMASK4	IMASK3	IMASK2	IMASK1	IMASK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:0	IMASKx	线 x 上的中断屏蔽（x 表示 0, 1, 2...19, 20） 0: 屏蔽来自线 x 上的中断请求； 1: 开放来自线 x 上的中断请求。

6.3.4 EXTI 下降沿触发配置寄存器（EXTI_FT_CFG）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											FT_CFG20	FT_CFG19	FT_CFG18	FT_CFG17	FT_CFG16
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT_CFG15	FT_CFG14	FT_CFG13	FT_CFG12	FT_CFG11	FT_CFG10	FT_CFG9	FT_CFG8	FT_CFG7	FT_CFG6	FT_CFG5	FT_CFG4	FT_CFG3	FT_CFG2	FT_CFG1	FT_CFG0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:0	FT_CFGx	线 x 上的下降沿触发配置位（x 表示 0, 1, 2...19, 20） 0: 禁止输入线 x 上的下降沿触发（中断和事件）

位域	名称	描述
		1: 允许输入线 x 上的下降沿触发（中断和事件）

6.3.5 EXTI 上升沿触发配置寄存器（EXTI_RT_CFG）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											RT_CFG20	RT_CFG19	RT_CFG18	RT_CFG17	RT_CFG16
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT_CFG15	RT_CFG14	RT_CFG13	RT_CFG12	RT_CFG11	RT_CFG10	RT_CFG9	RT_CFG8	RT_CFG7	RT_CFG6	RT_CFG5	RT_CFG4	RT_CFG3	RT_CFG2	RT_CFG1	RT_CFG0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:0	RT_CFGx	线 x 上的上升沿触发配置位（x 表示 0, 1, 2...19, 20） 0: 禁止输入线 x 上的上升沿触发（中断和事件） 1: 允许输入线 x 上的上升沿触发（中断和事件）

6.3.6 EXTI 挂起寄存器（EXTI_PEND）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											PEND20	PEND19	PEND18	PEND17	PEND16
											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
PEND15	PEND14	PEND13	PEND12	PEND11	PEND10	PEND9	PEND8	PEND7	PEND6	PEND5	PEND4	PEND3	PEND2	PEND1	PEND0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:0	PENDx	线 x 上的挂起位（x 表示 0, 1, 2...19, 20） 0: 没有发生挂起请求 1: 发生了挂起触发请求 当外部中断线上发生了选择的边沿触发事件，该位被置'1'。在该位中写入'1'可以清除它，也可以通过改变边沿检测的极性清除此位。

6.3.7 EXTI 软件中断事件寄存器 (EXTI_SWIE)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											SWIE20	SWIE19	SWIE18	SWIE17	SWIE16
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIE15	SWIE14	SWIE13	SWIE12	SWIE11	SWIE10	SWIE9	SWIE8	SWIE7	SWIE6	SWIE5	SWIE4	SWIE3	SWIE2	SWIE1	SWIE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:21	Reserved	保留, 必须保持复位值。
20:0	SWIE _x	<p>线 x 上的软件中断 (x 表示 0, 1, 2...19, 20)</p> <p>当该位为'0'时, 写'1'将设置 EXTI_PEND 中相应的挂起位。如果在 EXTI_IMASK 和 EXTI_EMASK 中允许产生该中断, 此时将产生一个中断。</p> <p>注: 通过写入'1'清除 EXTI_PEND 的对应位, 可以清除该位为'0'。</p>

6.3.8 EXTI 时间戳触发源选择寄存器 (EXTI_TS_SEL)

偏移地址: 0x18

复位值: 0x0000 0000

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

位域	名称	描述
31:4	Reserved	保留, 必须保持复位值。
3:0	TSSEL[3:0]	<p>选择外部中断输入作为时间戳事件的触发源</p> <p>0000: 选择 EXTI0 作为时间戳事件的触发源;</p> <p>0001: 选择 EXTI1 作为时间戳事件的触发源;</p> <p>.....</p> <p>1111: 选择 EXTI15 作为时间戳事件的触发源。</p>

7 DMA 控制器

7.1 简介

DMA 控制器总共可以访问 5 个 AHB 从机：Flash、SRAM、ADC、ABP1 和 APB2。DMA 控制器由 CPU 控制以执行从源到目的的快速数据移动。配置完成后，无需 CPU 干预即可传输数据。因此，可以释放 CPU 用于其他计算/控制任务或节省整体系统功耗。

MCU 的主要架构是具有循环仲裁方案的多层 AHB-Lite 总线结构。DMA 和 CPU 内核可以并行访问不同的从机，也可以顺序访问相同的从机。

DMA 控制器有 5 个逻辑通道。每个逻辑通道用于服务来自单个或多个外设的内存访问请求。内部仲裁器控制不同 DMA 通道的优先级。

支持外设请求通道用户可配置。

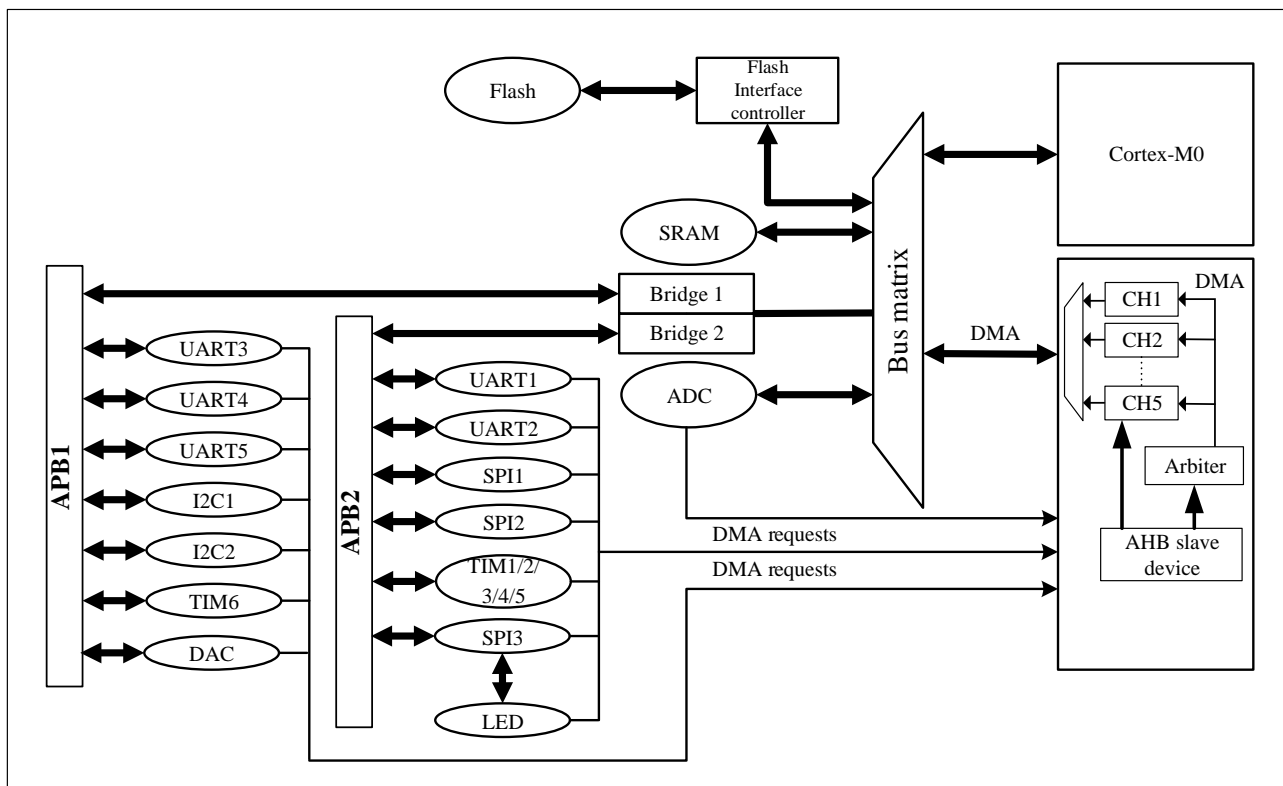
7.2 主要特性

DMA 主要特性：

- 5 个可独立配置的 DMA 通道。
- 每个 DMA 通道支持硬件请求和软件触发来启动传输，并由软件配置。
- 每个 DMA 通道都有专用的软件优先级（DMA_CHCFGx.PRIOLVL[1:0]位，对应 4 个优先级），可以单独配置。具有相同软件优先级的通道将进一步比较硬件索引（通道号）以确定最终优先级（索引号越低的通道优先级越高）。
- 可配置的源和目标大小。地址设置应与数据大小相对应。
- 每个通道可配置循环传输模式。
- 每个通道有 3 个独立的事件标志和中断（传输完成、半传输、传输错误）和 1 个全局中断标志（由 3 个事件的逻辑或设置）。
- 支持内存到内存、内存到外设和外设到内存三种传输类型。
注：也可以支持外设 SPI 到外设 DAC 的搬运（模式配置为外设到内存模式）。
- 共访问 5 个 AHB 从机：Flash、SRAM、ADC、APB1 和 APB2。
- 可配置数据传输数（0~65535）。

7.3 功能框图

图 7-1 DMA 框图



7.4 功能描述

DMA 控制器和 Cortex™-M0 内核共享相同的系统数据总线。当 CPU 和 DMA 同时访问同一个目标（RAM 或外设）时，DMA 请求会暂停 CPU 访问系统总线几个周期，由总线仲裁器进行循环调度。这允许 CPU 获得至少一半的系统总线（内存或外围设备）带宽。

7.4.1 DMA 操作

DMA 请求可以由硬件外设或软件触发，DMA 控制器根据通道的优先级处理请求。根据配置的传输地址和位宽从源地址读取数据，然后将读取的数据存储在目的地址空间中。一次操作后，控制器计算剩余传输次数，并更新下一次传输的源地址和目的地址。

每个 DMA 数据传输包括三个操作：

- 数据访问：根据传输方向确定源地址（DMA_PADDR_x 或 DMA_MADDR_x），从源地址读取数据。
- 数据存储：根据传输方向确定目的地址（DMA_PADDR_x 或 DMA_MADDR_x），将读取的数据存储到目的地址空间。
- 计算未完成操作的数量，对 DMA_TXNUM_x 寄存器进行减量操作，更新下一个操作的源地址和目的地址。

7.4.2 通道优先级和仲裁器

DMA 使用仲裁策略来处理来自不同通道的多个请求。每个通道的优先级可在通道控制寄存器 (DMA_CHCFGx) 中进行编程。

4 个优先级：

- ◆ 非常高优先级
- ◆ 高优先级
- ◆ 中优先级
- ◆ 低优先级

默认情况下，如果编程的优先级相同，则索引较低的通道具有较高的优先级。

对于内存到内存的传输，在 4 次传输操作后进行重新仲裁。

对于与外设相关的传输，每次传输操作后都会进行重新仲裁。

7.4.3 DMA 通道和传输数量

每个通道都可以在指定地址的外设寄存器和内存地址之间进行 DMA 传输。DMA 传输的数据数量是可编程的，最大支持值为 65535。DMA_TXNUM 寄存器在每次传输后递减。

7.4.4 可编程的数据位宽

外设和内存传输数据位宽支持字节、半字和字，可以通过 DMA_CHCFGx.PSIZE 和 DMA_CHCFGx.MSIZE 进行编程。

当 DMA_CHCFGx.PSIZE 和 DMA_CHCFGx.MSIZE 不同时，DMA 模块根据表 7-1 对齐数据。

表 7-1 可编程的数据宽度和大小端操作(当 PINC = MINC = 1)

Source width (bit)	Destination width (bit)	Number of transfer (Byte)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
8	8	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W B0 [7:0] @0x0 2: R B1 [7:0] @0x1, W B1 [7:0] @0x1 3: R B2 [7:0] @0x2, W B2 [7:0] @0x2 4: R B3 [7:0] @0x3, W B3 [7:0] @0x3	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3
8	16	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W 00B0 [15:0] @0x0 2: R B1 [7:0] @0x1, W 00B1 [15:0] @0x2 3: R B2 [7:0] @0x2, W 00B2 [15:0] @0x4 4: R B3 [7:0] @0x3, W 00B3 [15:0] @0x6	0x0 / 00B0 0x2 / 00B1 0x4 / 00B2 0x6 / 00B3
8	32	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W 000000B0 [31:0] @0x0 2: R B1 [7:0] @0x1, W 000000B1 [31:0] @0x4 3: R B2 [7:0] @0x2, W 000000B2 [31:0] @0x8 4: R B3 [7:0] @0x3, W 000000B3 [31:0] @0xC	0x0 / 000000B0 0x4 / 000000B1 0x8 / 000000B2 0xC / 000000B3

Source width (bit)	Destination width (bit)	Number of transfer (Byte)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
16	8	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W B0 [7:0] @0x0 2: R B3B2 [15:0] @0x2, W B2 [7:0] @0x1 3: R B5B4 [15:0] @0x4, W B4 [7:0] @0x2 4: R B7B6 [15:0] @0x6, W B6 [7:0] @0x3	0x0 / B0 0x1 / B2 0x2 / B4 0x3 / B6
16	16	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W B1B0 [15:0] @0x0 2: R B3B2 [15:0] @0x2, W B3B2 [15:0] @0x2 3: R B5B4 [15:0] @0x4, W B5B4 [15:0] @0x4 4: R B7B6 [15:0] @0x6, W B7B6 [15:0] @0x6	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6
16	32	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W 0000B1B0 [31:0] @0x0 2: R B3B2 [15:0] @0x2, W 0000B3B2 [31:0] @0x4 3: R B5B4 [15:0] @0x4, W 0000B5B4 [31:0] @0x8 4: R B7B6 [15:0] @0x6, W 0000B7B6 [31:0] @0xC	0x0 / 0000B1B0 0x4 / 0000B3B2 0x8 / 0000B5B4 0xC / 0000B7B6
32	8	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B0 [7:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B4 [7:0] @0x1 3: R BBBAB9B8 [31:0] @0x8, W B8 [7:0] @0x2 4: R BFBEBDBC [31:0] @0xC, W BC [7:0] @0x3	0x0 / B0 0x1 / B4 0x2 / B8 0x3 / BC
32	16	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B1B0 [15:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B5B4 [15:0] @0x2 3: R BBBAB9B8 [31:0] @0x8, W B9B8 [15:0] @0x4 4: R BFBEBDBC [31:0] @0xC, W BDBC [15:0] @0x6	0x0 / B1B0 0x2 / B5B4 0x4 / B9B8 0x6 / BDBC
32	32	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B3B2B1B0 [31:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B7B6B5B4 [31:0] @0x4 3: R BBBAB9B8 [31:0] @0x8, W BBBAB9B8 [31:0] @0x8 4: R BFBEBDBC [31:0] @0xC, W BFBEBDBC [31:0] @0xC	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC

注意:

DMA 总是向 HWDATA[31:0]提供完整的 32 位数据, 无论它是什么目标大小 (HSIZE 仍然遵循设备支持字节/半字操作的目标大小设置)。它提供的 HWDATA[31:0]遵循以下规则:

- 当源大小小于目标大小时, DMA 用 0 填充 MSB, 直到它们的大小匹配并将其复制为 32 位。例如, 源是 8 位数据 0x55, 目标大小是 16 位。DMA 用 0 填充源数据使其成为 16 位 0x0055, 然后将其复制为 32 位数据 0x0055_0055 并提供给 HWDATA[31:0]; (如果目标大小为 32 位, 则 DMA 只会用 0 填充源数据)。
- 当源大小大于或等于目标大小且小于 32 位时, DMA 将源数据复制到 32 位数据。例如, 源数据为 8 位数据 0x1F, HWDATA[31:0] = 0x1F1F_1F1F。如果源数据是 16 位数据 0x2345, 则 HWDATA[31:0] = 0x2345_2345。

这保证了仅支持字操作的外设不会产生总线错误, 并且所需的数据仍然可以通过额外的位 (即 0 填充) 移动到我们想要的位置。如果用户想要配置一个 8 位寄存器但与 32 位地址边界对齐, 则源大小应设置为 8 位, 目标大小应设置为 32 位, 因此额外的位将用 0 填充。

7.4.5 外设/内存地址递增

DMA_CHCFGx.PINC 和 DMA_CHCFGx.MINC 分别控制外设地址和内存地址是否使能自动递增模式。软件在传输过程中不能写（可以读）地址寄存器。

- 在自动递增模式下，下一个要传输的地址在每次传输后根据数据位宽（1、2 或 4）自动增加。第一次传输的地址存储在 DMA_PADDRx 或 DMA_MADDRx 寄存器中。
- 在固定模式下，地址始终固定为初始地址。

在传输结束时（即传输计数变为 0），将根据当前是否工作于循环模式进行不同的处理。

- 在非循环模式下，DMA 在传输完成后停止。要开始新的 DMA 传输，需要在禁用 DMA 通道的情况下重写 DMA_TXNUMx 寄存器中的传输数量。
- 在循环模式下，在传输结束时，DMA_TXNUMx 寄存器的内容会自动重新加载其初始值，并且当前内部外设或内存地址也会重新加载 DMA_PADDRx 或 DMA_MADDRx 寄存器设置的初始基地址。

7.4.6 通道配置流程

详细配置流程如下：

1. 配置中断屏蔽位，1：启用中断，0：禁用中断。
2. 配置通道外设地址和内存地址以及传输方向。
3. 配置通道优先级，0：最低，3：最高。
4. 配置外设和内存地址增量。
5. 配置通道传输数据的宽度和数量。
6. 如有必要，配置循环模式。
7. 如果是存储器到存储器，配置 MEM2MEM 模式（注：要配置 DMA 工作在 M2M 模式，用户需要将相应的通道选择值设置为保留值，例如 47）。
8. 在通道 1~5 上重复第 1~8 步。
9. 最后使能相应通道。

如果使用软件提供中断服务，则软件必须查询中断状态寄存器以检查发生了哪个中断（软件需要向中断标志清除位写 1 来清除相应的中断）。在使能通道之前，应清除该通道对应的所有中断。

如果中断是传输完成中断，软件可以配置下一次传输，或者向用户报告该通道传输完成。

7.4.7 流量控制

支持三种主要的流量控制：

- 存储器到存储器
- 存储器到外设
- 外设到存储器

流控制由每个 DMA 通道配置寄存器中的两个寄存器位控制。流控制用于控制 DMA 通道的源/目标和方向。

表 7-2 流量控制表

DMA_CHCFGx.MEM2MEM	DMA_CHCFGx.DIR	Source	Destination	Transfer
1	x	Memory	Memory	AHB read to AHB write, can do back2back transfer
0	1	Memory	AHB Peripheral	AHB read to AHB write, single transfer
			APB Peripheral	AHB read to APB write, single transfer
0	0	AHB Peripheral	Memory	AHB read to AHB write, single transfer
		APB Peripheral		APB read to AHB write, single transfer

7.4.8 循环模式

循环模式用于处理循环缓冲区和连续数据传输（如 ADC 扫描模式）。DMA_CHCFGx.CIRC 用于启用此功能。激活循环模式时，如果要传输的数据数变为 0，则在配置通道时会自动恢复到初始值，继续进行 DMA 操作。

如果用户想关闭循环模式，用户需要向 DMA_CHCFGx.CHEN 写入 0 以禁用 DMA 通道，然后向 DMA_CHCFGx.CIRC 写入 0（当 DMA_CHCFGx.CHEN 为 1 时，DMA_CHCFGx 寄存器中的其他位不能被改写）。

7.4.9 错误管理

对保留地址区域的 DMA 访问会导致 DMA 传输错误。发生错误时，设置传输错误标志，硬件自动清除当前 DMA 通道使能位（DMA_CHCFGx.CHEN），通道操作停止。如果在 DMA_CHCFGx 寄存器中设置了传输错误中断使能位，则会产生中断。

7.4.10 中断

● 传输完成中断：

通道数据传输完成时会产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

● 半传输中断：

当传输了一半的通道数据时会产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

● 传输错误中断：

总线返回错误时产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

表 7-3 DMA 中断请求

中断事件	事件标志位	使能控制位
半传输	HTXF	HTXIE
传输完成	TXCF	TXCIE
传输错误	ERRF	ERRIE

7.4.11 DMA 请求映射

总共有来自所有外设的 55 个 DMA 请求。为了获得更好的支持和完全的灵活性，可以使用寄存器位来选择将哪个 DMA 请求映射到哪个 DMA 通道。下表显示了外设的 DMA 请求到 DMA 控制器的 DMA 通道的映射方案。

表 7-4 DMA 请求映射

DMA request source select	Peripheral DMA request	DMA request source select	Peripheral DMA request
sel = 0	ADC	sel = 28	TIM2_CH1
sel = 1	UART1_TX	sel = 29	TIM2_CH2
sel = 2	UART1_RX	sel = 30	TIM2_CH3
sel = 3	UART2_TX	sel = 31	TIM2_CH4
sel = 4	UART2_RX	sel = 32	TIM2_UP
sel = 5	UART3_TX	sel = 33	TIM2_TRIG
sel = 6	UART3_RX	sel = 34	TIM3_CH1
sel = 7	UART4_TX	sel = 35	TIM3_CH2
sel = 8	UART4_RX	sel = 36	TIM3_CH3
sel = 9	UART5_TX	sel = 37	TIM3_CH4
sel = 10	UART5_RX	sel = 38	TIM3_UP
sel = 11	SPI1_TX	sel = 39	TIM3_TRIG
sel = 12	SPI1_RX	sel = 40	TIM4_CH1
sel = 13	SPI2_TX	sel = 41	TIM4_CH2
sel = 14	SPI2_RX	sel = 42	TIM4_CH3
sel = 15	SPI3_TX	sel = 43	TIM4_CH4
sel = 16	SPI3_RX	sel = 44	TIM4_UP
sel = 17	I2C1_TX	sel = 45	TIM4_TRIG
sel = 18	I2C1_RX	sel = 46	TIM5_CH1
sel = 19	I2C2_TX	sel = 47	TIM5_CH2
sel = 20	I2C2_RX	sel = 48	TIM5_CH3
sel = 21	TIM1_CH1	sel = 49	TIM5_CH4
sel = 22	TIM1_CH2	sel = 50	TIM5_UP
sel = 23	TIM1_CH3	sel = 51	TIM5_TRIG
sel = 24	TIM1_CH4	sel = 52	TIM6_UP
sel = 25	TIM1_COM	sel = 53	DAC
sel = 26	TIM1_UP	sel = 54	GCLK_TRIG
sel = 27	TIM1_TRIG		

注意: DMA 不同通道不能使用同一请求源, 否则在多个通道都使能的情况下, 只有优先级较高的通道会被触发。

7.5 DMA 寄存器

7.5.1 DMA 寄存器总览

表 7-5 DMA 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000h	DMA_INTSTS	Reserved													ERRF5	HTXF5	TXCF5	GLBF5	ERRF4	HTXF4	TXCF4	GLBF4	ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
004h	DMA_INTCLR	Reserved												CERRF5	CHTXF5	CTXCF5	CGLBF5	CERRF4	CHTXF4	CTXCF4	CGLBF4	CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1	0	0	0	0	0																										
	Reset Value	0												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																										
008h	DMA_CHCFG1	Reserved														MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	0	0	0	0	0	0	0	0																										
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
00Ch	DMA_TXNUM1	Reserved														NDTX[15:0]														0					0	0	0	0	0	0	0	0	0	0	0	0	0	0																
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																						
010h	DMA_PADDR1	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																													
014h	DMA_MADDR1	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																												
018h	DMA_CHSEL1	Reserved																								CH_SEL[5:0]					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
	Reset Value	0																								0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
01Ch	DMA_CHCFG2	Reserved														MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	0	0	0	0	0	0	0	0	0																									
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
020h	DMA_TXNUM2	Reserved														NDTX[15:0]														0					0	0	0	0	0	0	0	0	0	0	0	0	0	0																
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
024h	DMA_PADDR2	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																												
028h	DMA_MADDR2	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
02Ch	DMA_CHSEL2	Reserved																								CH_SEL[5:0]					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
	Reset Value	0																								0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
030h	DMA_CHCFG3	Reserved														MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	0	0	0	0	0	0	0	0	0																									
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
034h	DMA_TXNUM3	Reserved														NDTX[15:0]														0					0	0	0	0	0	0	0	0	0	0	0	0	0	0																
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
038h	DMA_PADDR3	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
03Ch	DMA_MADDR3	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
040h	DMA_CHSEL3	Reserved																								CH_SEL[5:0]					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
	Reset Value	0																								0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
044h	DMA_CHCFG4	Reserved														MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	0	0	0	0	0	0	0	0	0																									
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
048h	DMA_TXNUM4	Reserved														NDTX[15:0]														0					0	0	0	0	0	0	0	0	0	0	0	0	0	0																
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
04Ch	DMA_PADDR4	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
050h	DMA_MADDR4	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
054h	DMA_CHSEL4	Reserved																								CH_SEL[5:0]					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
	Reset Value	0																								0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
058h	DMA_CHCFG5	Reserved														MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	0	0	0	0	0	0	0	0	0																									
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
05Ch	DMA_TXNUM5	Reserved														NDTX[15:0]														0					0	0	0	0	0	0	0	0	0	0	0	0	0	0																
	Reset Value	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
060h	DMA_PADDR5	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
064h	DMA_MADDR5	ADDR[31:0]																																	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																											
068h	DMA_CHSEL5	Reserved																								CH_SEL[5:0]					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
	Reset Value	0																								0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												

7.5.2 DMA 中断状态寄存器 (DMA_INTSTS)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ERRF5	HTXF5	TXCF5	GLBF5
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRF4	HTXF4	TXCF4	GLBF4	ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	名称	描述
31:20	Reserved	保留，必须保持复位值。
19/15/11/7/3	ERRFx	通道 x(x = 1...5)的传输错误标志。 发生传输错误时，硬件设置该位。该位由软件通过向 DMA_INTCLR.CERRFx 位写 1 清零。 0: 通道 x 上没有发生传输错误。 1: 通道 x 上发生传输错误。
18/14/10/6/2	HTXFx	通道 x(x = 1...5)的半传输标志。 当半传输完成时，硬件设置该位。该位由软件通过向 DMA_INTCLR.CHTXFx 位写 1 清零。 0: 通道 x 上的半传输尚未完成。 1: 通道 x 上的半传输已完成。
17/13/9/5/1	TXCFx	通道 x(x = 1...5)的传输完成标志。 当传输完成时，硬件设置该位。该位由软件通过向 DMA_INTCLR.CTXCFx 位写 1 清零。 0: 通道 x 上的传输尚未完成。 1: 通道 x 上的传输已完成。
16/12/8/4/0	GLBFx	通道 x(x = 1...5)的全局标志。 当该通道中发生任何中断事件时，硬件设置该位。该位由软件通过向 DMA_INTCLR.CGLBFx 位写 1 清零。 0: 通道 x 上没有发生传输错误、半传输或传输完成事件。 1: 通道 x 上发生传输错误、半传输或传输完成事件之一。

7.5.3 DMA 中断标志清除寄存器 (DMA_INTCLR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CERRF5	CHTXF5	CTXCF5	CGLBF5
												w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CERRF4	CHTXF4	CTXCF4	CGLBF4	CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
19/15/11/7/3	CERRFx	清除通道 x(x = 1...5)的传输错误标志。 软件可以设置该位来清除相应通道的 ERRF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.ERRF 位。
18/14/10/6/2	CHTXFx	清除通道 x(x = 1...5)的半传输标志。 软件可以设置该位来清除相应通道的 HTXF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.HTXF 位。
17/13/9/5/1	CTXCFx	清除通道 x(x = 1...5)的传输完成标志。 软件可以设置该位来清除相应通道的 TXCF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.TXCF 位。
16/12/8/4/0	CGLBFx	清除通道 x(x = 1...5)的全局事件标志。 软件可以设置该位来清除相应通道的 GLBF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.GLBF 位。

7.5.4 DMA 通道 x 配置寄存器 (DMA_CHCFGx)

注: x 为通道号, x = 1...5

偏移地址: 0x08+20 * (x-1)

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2 MEM	PRIOLVL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:15	Reserved	保留，必须保持复位值。
14	MEM2MEM	存储器到存储器模式。 当通道尚未使能时，软件可以将此通道配置为存储器到存储器传输。 0：存储器和外设之间的通道传输。 1：通道设置为存储器到存储器间的传输。
13:12	PRIOLVL[1:0]	通道优先级。 当通道未使能时，软件可以编程通道优先级。 00：低 01：中 10：高 11：非常高
11:10	MSIZE[1:0]	存储器数据大小。 软件可以配置从/向存储器地址读取/写入的数据大小。 00：8 位 01：16 位 10：32 位 11：保留
9:8	PSIZE[1:0]	外设数据大小。 软件可以配置从/向外设地址读取/写入的数据大小。 00：8 位 01：16 位 10：32 位 11：保留
7	MINC	存储器地址递增模式。 软件可以使能/禁能存储器地址递增模式。 0：内存地址不会随着每次传输而递增。 1：内存地址随着每次传输而递增。
6	PINC	外设地址增量模式。 软件可以使能/禁能外设地址递增模式。 0：外设地址不会随着每次传输而递增。 1：外设地址随每次传输而递增。
5	CIRC	循环模式。 软件可以设置/清除该位。 0：经过一轮传输后通道停止。 1：通道配置为循环模式。
4	DIR	数据传输方向 软件可以设置/清除该位。 0：从外设到存储器的数据传输 1：从存储器到外设的数据传输。
3	ERRIE	传输错误中断使能。 软件可以使能/禁能传输错误中断。 0：禁止通道 x 的传输错误中断。 1：使能通道 x 的传输错误中断。

位域	名称	描述
2	HTXIE	半传输中断使能。 软件可以使能/禁能半传输中断。 0: 禁止通道 x 的半传输中断。 1: 使能通道 x 的半传输中断。
1	TXCIE	传输完成中断使能。 软件可以使能/禁能传输完成中断。 0: 禁止通道 x 的传输完成中断。 1: 使能通道 x 的传输完成中断。
0	CHEN	通道使能。 软件可以设置/复位该位。 0: 禁用通道。 1: 使能通道。

7.5.5 DMA 通道 x 传输数量寄存器 (DMA_TXNUMx)

注: x 为通道号, x = 1...5

偏移地址: 0x0c+20 * (x-1)

复位值: 0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	NDTX	数据传输数量。 要传输的数据数量 (0~65535)。软件可以在通道禁用时写入/读出传输数量, 并且通道使能后该位为只读。相应的 DMA 通道每次成功传输后, 该寄存器就会减 1。如果使能循环模式, 它会在达到零时自动重新加载预设值。否则它将保持为零并复位通道使能位。

7.5.6 DMA 通道 x 外设基地址寄存器 (DMA_PADDRx)

注: x 为通道号, x = 1...5

偏移地址: 0x10+20 * (x-1)

复位值: 0x0000 0000

只有在禁用通道(DMA_CHCFGx.CHEN = 0)时才能写该寄存器。

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

位域	名称	描述
31:0	ADDR	<p>外设基地址。</p> <p>DMA 读取/写入的外设起始地址。</p> <p>地址的递增由 DMA_CHCFGx.PSIZE 决定。DMA_CHCFGx.PSIZE 等于‘01’，DMA 忽略 PADDR 的第 0 位，如果 DMA_CHCFGx.PSIZE 等于‘10’，DMA 将忽略 PADDR 的第 0 位和第 1 位。</p>

7.5.7 DMA 通道 x 存储器基地址寄存器（DMA_MADDRx）

注：x 为通道号，x = 1...5

偏移地址：0x14+20 * (x-1)

复位值：0x0000 0000

只有在禁用通道(DMA_CHCFGx.CHEN = 0)时才能写该寄存器。

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

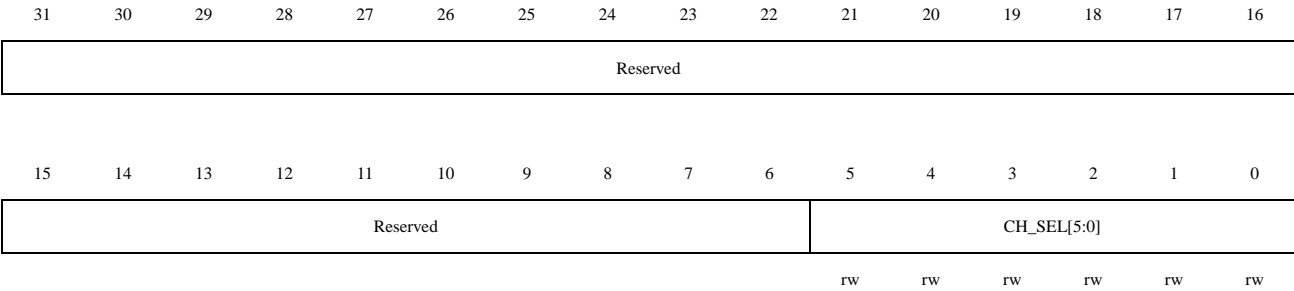
位域	名称	描述
31:0	ADDR	<p>存储器基地址。</p> <p>DMA 读取/写入的存储器起始地址。</p> <p>地址的递增由 DMA_CHCFGx.MSIZE 决定。DMA_CHCFGx.MSIZE 等于‘01’，DMA 忽略 MADDR 的第 0 位，如果 DMA_CHCFGx.MSIZE 等于‘10’，DMA 将忽略 MADDR 的第 0 位和第 1 位。</p>

7.5.8 DMA 通道 x 请求选择寄存器（DMA_CHSELx）

注：x 为通道号，x = 1...5

偏移地址：0x18+20 * (x-1)

复位值：0x0000 0000



位域	名称	描述
31:6	Reserved	保留，必须保持复位值。
5:0	CH_SEL[5:0]	DMA 通道请求源选择 0x00: adc_dma 0x36: GCLK_TRIG 外设 DMA 请求映射到 DMA 输入请求通道号请参考表 7-4 DMA 请求映射。

8 CRC 计算单元

8.1 简介

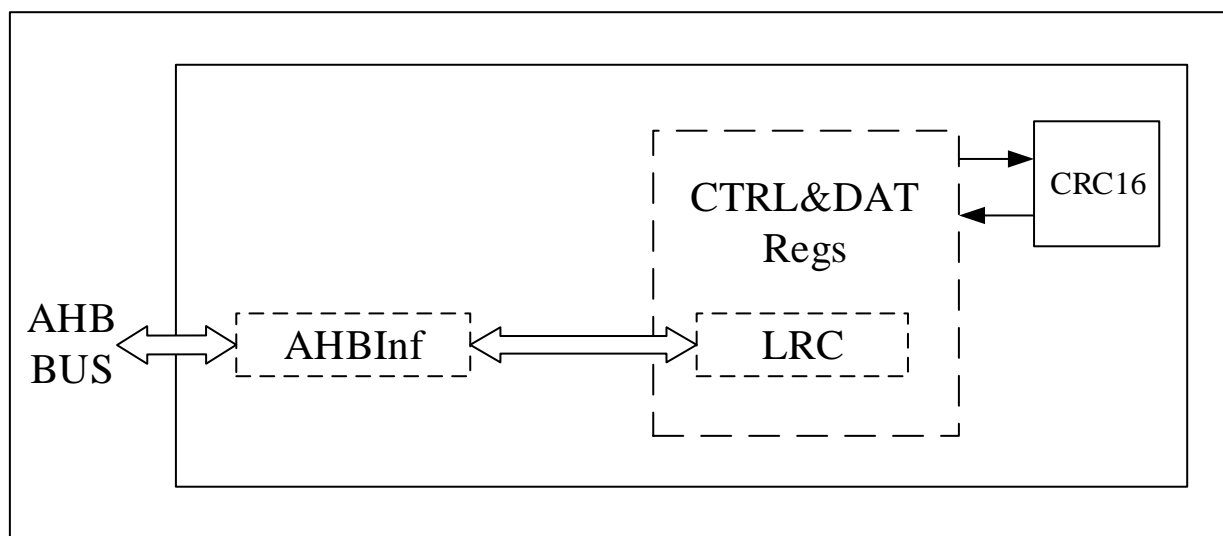
该模块集成了 CRC16 的功能，循环冗余校验（CRC）计算单元根据固定的生成多项式得到任意 CRC 计算结果。在其他应用中，CRC 技术主要用于验证数据传输或数据存储的正确性和完整性。CRC 计算单元可以在程序运行时计算出软件的标识符，然后与编译链接时产生的参考标识符进行比较，然后存储在指定的内存空间中。

8.2 主要特性

- CRC16($X^{16}+X^{15}+X^2+1$)。
- 8 位待校验数据和 16 位输出校验码。
- CRC 计算时间：1 个 AHB 时钟周期（HCLK）。
- 可配置校验初始值，可配置待校验数据的大小端。
- 支持 8bit LRC 校验值生成。

下图为 CRC 计算单元框图：

图 8-1 CRC 计算单元框图



8.3 CRC 功能描述

通过 CRC_CRC16CTRL.ENDHL 位来控制校验数据的小端或大端。

要清除最后一次 CRC 操作的结果，请将 CRC_CRC16CTRL.CLR 设置为 1 或 CRC_CRC16D 设置为 0。

CRC 计算的初始值可以通过写 CRC_CRC16D 寄存器来配置。默认情况下，初始值是上一次计算的结果。

LRC 计算与 CRC 计算相同。两者同时进行。可根据需要读出 CRC 或 LRC。如果需要设置初始值，首先要配置 LRC 寄存器。

8.4 CRC 软件计算方式

在实际应用中，如果需要通过软件计算 CRC16 来匹配硬件计算结果，则需要正确配置如下内容：

- 宽度 WIDTH: 16
- 多项式 POLY: 0x8005
- 初始值 INIT: 0x0000
- 结果异或值 XOROUT: 0x0000
- 输入数据反转 REFIN: 否
- 输出数据反转 REFOUT: 否

8.5 CRC 寄存器

8.5.1 CRC 寄存器总览

下表列出了 CRC 的寄存器映射和复位值

表 8-1 CRC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00Ch	CRC16CTRL	Reserved																										CLR	ENDHL	Reserved			
	Reset Value																											0	0				
010h	CRC16DAT	Reserved																	CRC16DAT[7:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0					
014h	CRC16D	Reserved												CRC16D[15:0]																			
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
018h	LRC	Reserved																	LRCDAT[7:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0					

8.5.2 CRC16 控制寄存器（CRC_CRC16CTRL）

Address offset: 0x0C

Reset value: 0x0000 0000

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

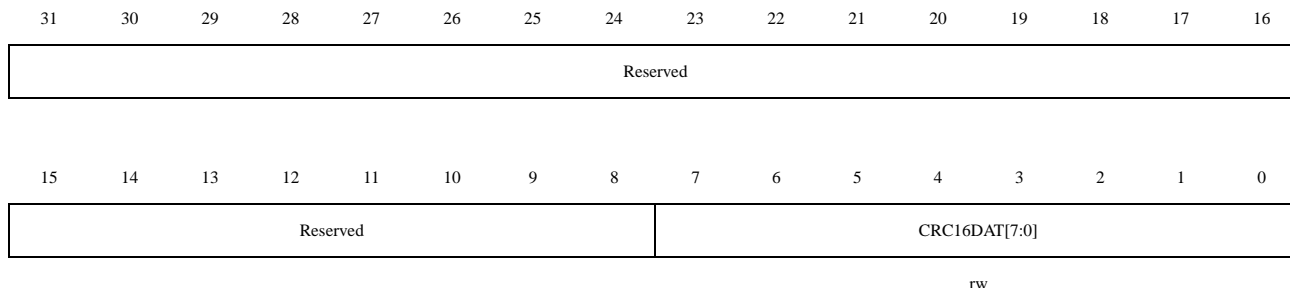
位域	名称	描述
31:3	Reserved	保留，必须保持复位值。
2	CLR	清除 CRC16 结果。 0: 不清除 1: 清除为默认值 0x0000。将此位设置为 1 只会维持 1 时钟周期，硬件会自动清零。 （软件读取始终为 0）。
1	ENDHL	要验证的数据从 MSB 或 LSB 开始计算。 0: 从 MSB 到 LSB 1: 从 LSB 到 MSB 该位仅用于要验证的数据。
0	Reserved	保留，必须保持复位值。

注：支持 8 位、16 位、32 位操作

8.5.3 CRC16 待校验寄存器（CRC_CRC16DAT）

偏移地址：0x10

复位值：0x0000 0000



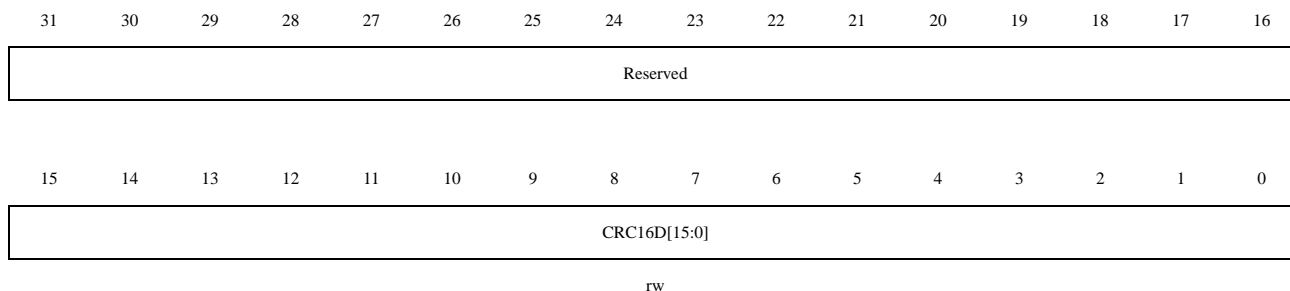
位域	名称	描述
31:8	Reserved	保留，必须保持复位值。
7:0	CRC16DAT[7:0]	待校验的数据。

注：支持 8 位、16 位、32 位操作

8.5.4 CRC 循环冗余校验码寄存器（CRC_CRC16D）

偏移地址：0x14

复位值：0x0000 0000



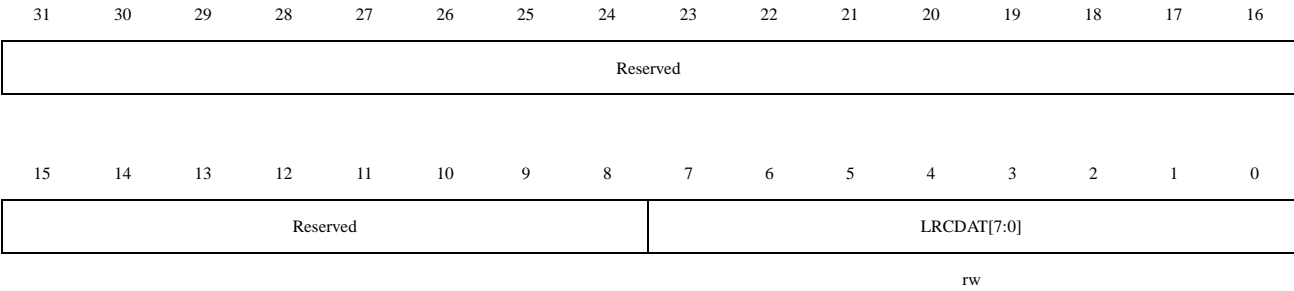
位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRC16D[15:0]	16 位循环冗余结果值。 每次软件写入 CRC16DAT 寄存器时，来自 CRC16 的 16 位计算数据都会在该寄存器中更新。

注：支持 8 位、16 位和 32 位操作（8 位操作必须连续执行两次才能保证 16 位初始值配置正确）

8.5.5 LRC 校验值寄存器（CRC_LRC）

偏移地址：0x18

复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，必须保持复位值。
7:0	LRCDAT[7:0]	LRC 校验值。 软件使用前需要写入初始值。然后每次写入 CRC_CRC16DAT 的数据都会与 CRC_LCR 寄存器的值进行“异或”。结果将存储在 CRC_LCR 中。软件读取结果，下次使用前应清除。

9 高级控制定时器（TIM1）

9.1 TIM1 简介

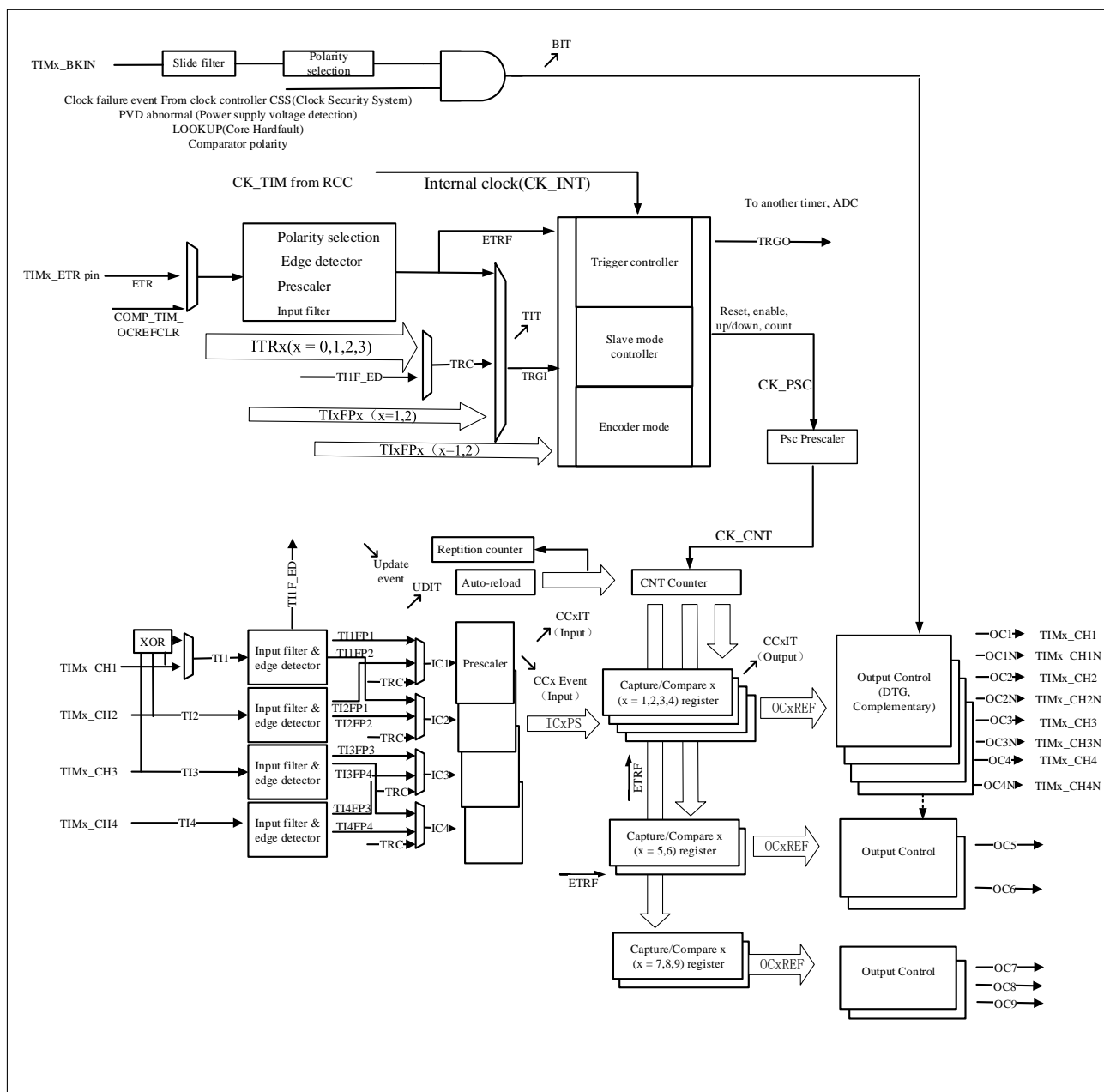
高级控制定时器（TIM1）主要用于以下场合：对输入信号进行计数、测量输入信号的脉冲宽度和产生输出波形等。

高级定时器具有互补输出功能、死区插入和刹车功能。适用于电机控制。

9.2 TIM1 主要特性

- 16 位自动装载计数器。（可实现向上计数、向下计数、向上/下计数）
- 16 位可编程预分频器。（分频系数可配置为 1 到 65536 之间的任意值）
- 可编程重复计数器
- TIM1 最多 9 个通道
- 4 个捕获/比较通道，工作模式为：PWM 输出、输出比较、单脉冲模式输出、输入捕获
- 如下事件发生时产生中断/DMA：
 - ◆ 更新事件
 - ◆ 触发事件
 - ◆ 输入捕获
 - ◆ 输出比较
 - ◆ 刹车信号输入
- 死区时间可编程的互补输出
 - 对于 TIM1，通道 1、2、3、4 支持此功能
- 可通过外部信号控制定时器
- 多个定时器内部连接在一起，以实现定时器的同步或链接
- TIM1_CC5 用于比较器消隐
- TIM1 的通道 4/7/8/9 可输出的脉冲信号，可通过配置上升沿和下降沿来触发 ADC
- 增量（正交）编码器接口：用于追踪运行轨迹和解析旋转方位
- 霍尔传感器接口：用于三相电机控制

图 9-1 TIM1 框图



9.3 TIM1 功能描述

9.3.1 时基单元

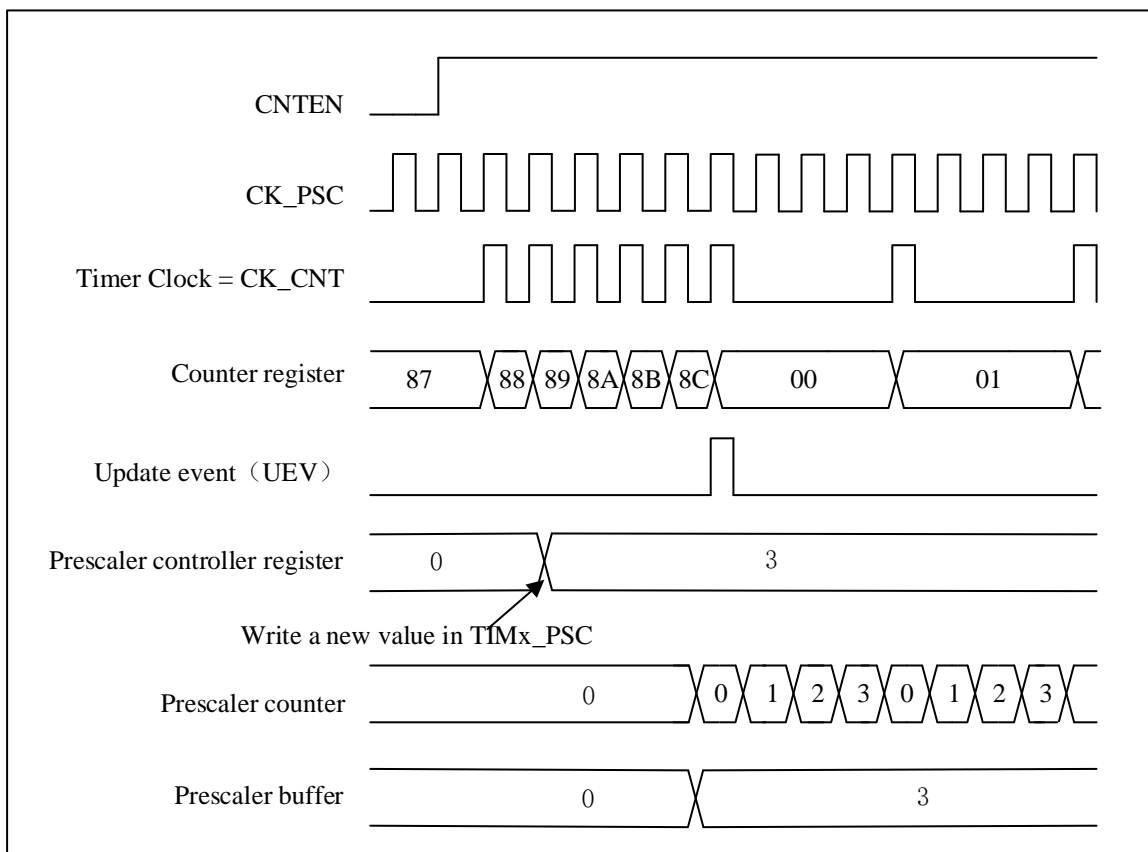
高级控制器的时基单元主要包括：预分频器、计数器、自动重载寄存器和重复计数器。当时基单元工作时，软件可以随时读取和写入相应的寄存器（TIMx_PSC、TIMx_CNT、TIMx_AR 和 TIMx_REPCNT）。

根据自动重载预装载使能位（TIMx_CTRL1.ARPEN）的设置，预装载寄存器的值会立即或在每次更新事件 UEV 时传输到影子寄存器。TIMx_CTRL1.UPDIS=0 时，计数器上溢/下溢或软件设置 TIMx_EVTGEN.UDGN 将生成更新事件。计数器 CK_CNT 仅在 TIMx_CTRL1.CNTEN 位被设置时有效。计数器在 TIMx_CTRL.CNTEN 位被设置后一个时钟周期之后开始计数。

9.3.1.1 预分频器描述

TIMx_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

图 9-2 当预分频的参数从 1 到 4，计数器的时序图



9.3.2 计数器模式

9.3.2.1 向上计数模式

使用向上计数模式，计数器将从 0 计数到寄存器 TIMx_AR 的值，然后重置为 0。并产生一个计数器溢出事件。

如果设置 TIMx_CTRL1.UPRS 位（选择更新请求）和 TIMx_EVTGEN.UDGN 位，将产生一个更新事件（UEV）。但是 TIMx_STS.UDITF 不会被硬件置起，因此不会产生更新中断或 DMA 更新请求。这是为了避免清除计数器时产生更新中断。

取决于 TIMx_CTRL1.UPRS 的配置，当发生更新事件时，TIMx_STS.UDITF 被设置，所有寄存器都会更新：

- 重复计数器被重新加载为 TIMx_REPCNT 的内容
- 当 TIMx_CTRL1.ARPEN = 1，预装载寄存器(TIMx_AR)的值被更新到自动装载影子寄存器
- 预加载值 (TIMx_PSC) 被重新加载到预分频器影子寄存器中

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以通过设置 TIMx_CTRL1.UPDIS=1 来禁止更新事件。

当产生一个更新事件时，计数器仍将被清除，预分频器计数器也将被设置为 0 (但预分频器值将保持不变)。

下图给出一些示例，展示了向上计数模式计数器在不同分频因子下的动作。

图 9-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图

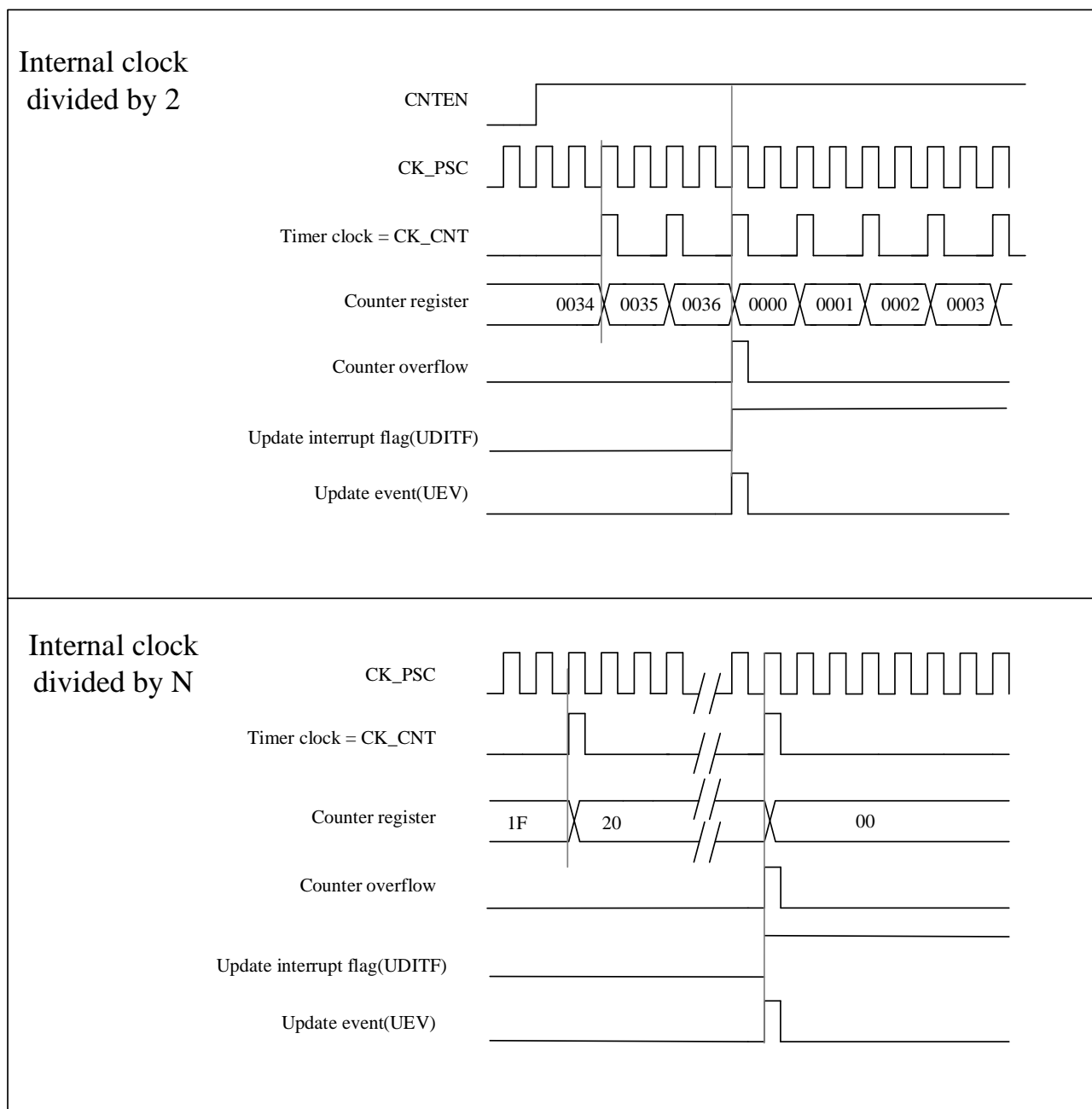
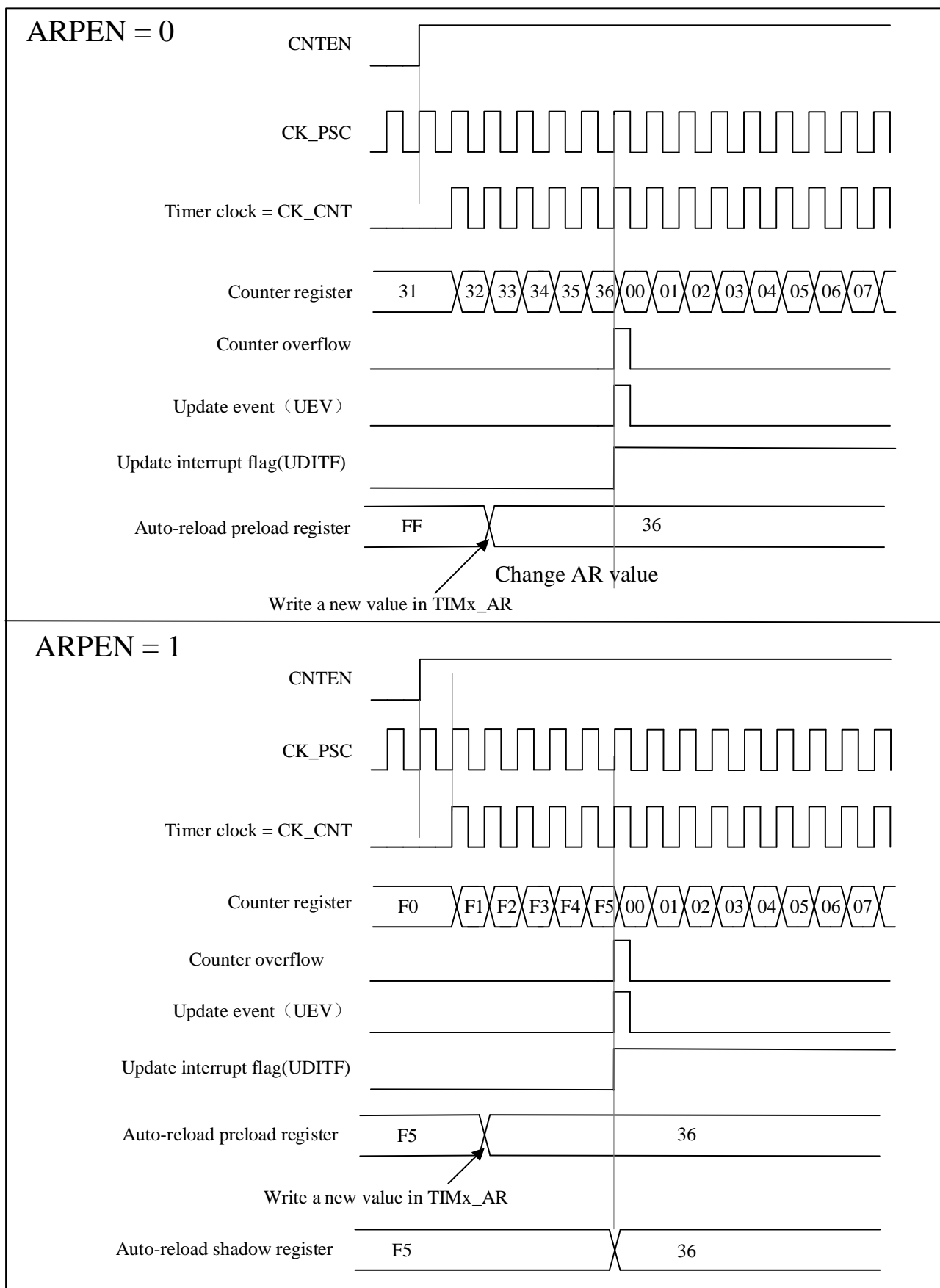


图 9-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图



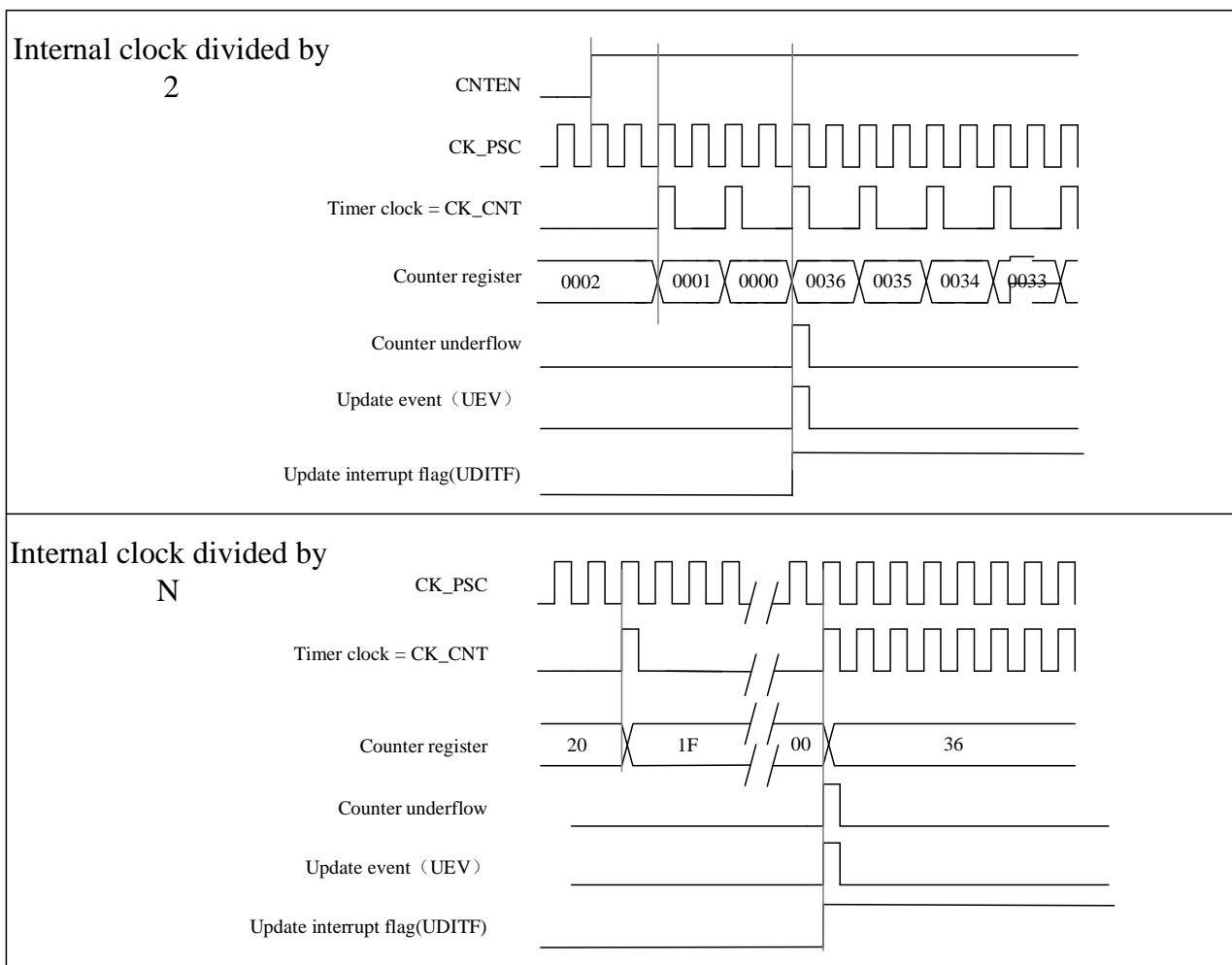
9.3.2.2 向下计数模式

向下计数模式，计数器将从寄存器 TIMx_AR 的值减至 0，然后从自动重载值重新开始，并产生计数器向下溢出事件

向下计数模式和向上计数模式配置更新事件和更新寄存器的过程相同，请查阅 9.3.2.1 章节。

下图给出一些示例，展示了向下计数模式计数器在不同分频因子下的动作。

图 9-5 内部时钟分频因子 = 2/N 时，向下计数时序图



9.3.2.3 中央对齐模式

9.3.2.3.1 中央对称模式

在中央对齐模式下，计数器从 0 增加到值 (TIMx_AR) - 1，产生计数器溢出事件。然后，它从自动重载值 (TIMx_AR) 向下计数到 1，并生成一个计数器向下溢出事件。然后计数器重置为 0 并再次开始计数。

在这种模式下，TIMx_CTRL1.DIR 方向位无效，由硬件更新和指定当前计数方向。当 TIMx_CTRL1.CAMSEL 位不等于“00”时，中央对齐模式有效。

每次计数上溢和计数下溢时都会生成更新事件。或者，也可以通过设置 TIMx_EVTGEN.UDGN 位（通过软件或使用从模式控制器）来生成更新事件。在这种情况下，计数器从 0 重新开始计数，预分频器的计数器也从 0 重新开始计数。

注：如果因为计数器溢出而产生更新，自动重载将在计数器重新载入之前被更新。

图 9-6 内部时钟分频因子 = 2/N，中央对齐时序图

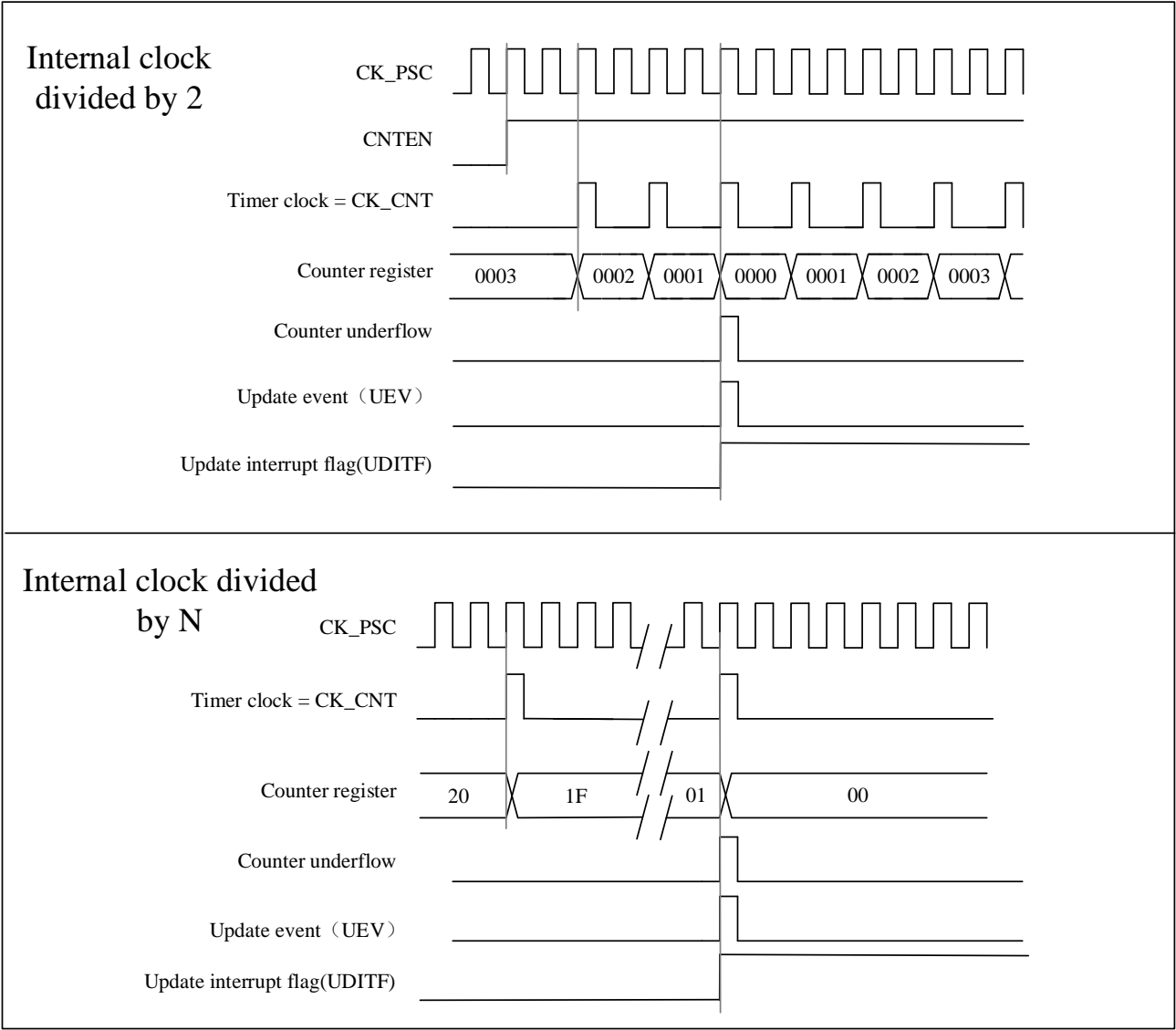
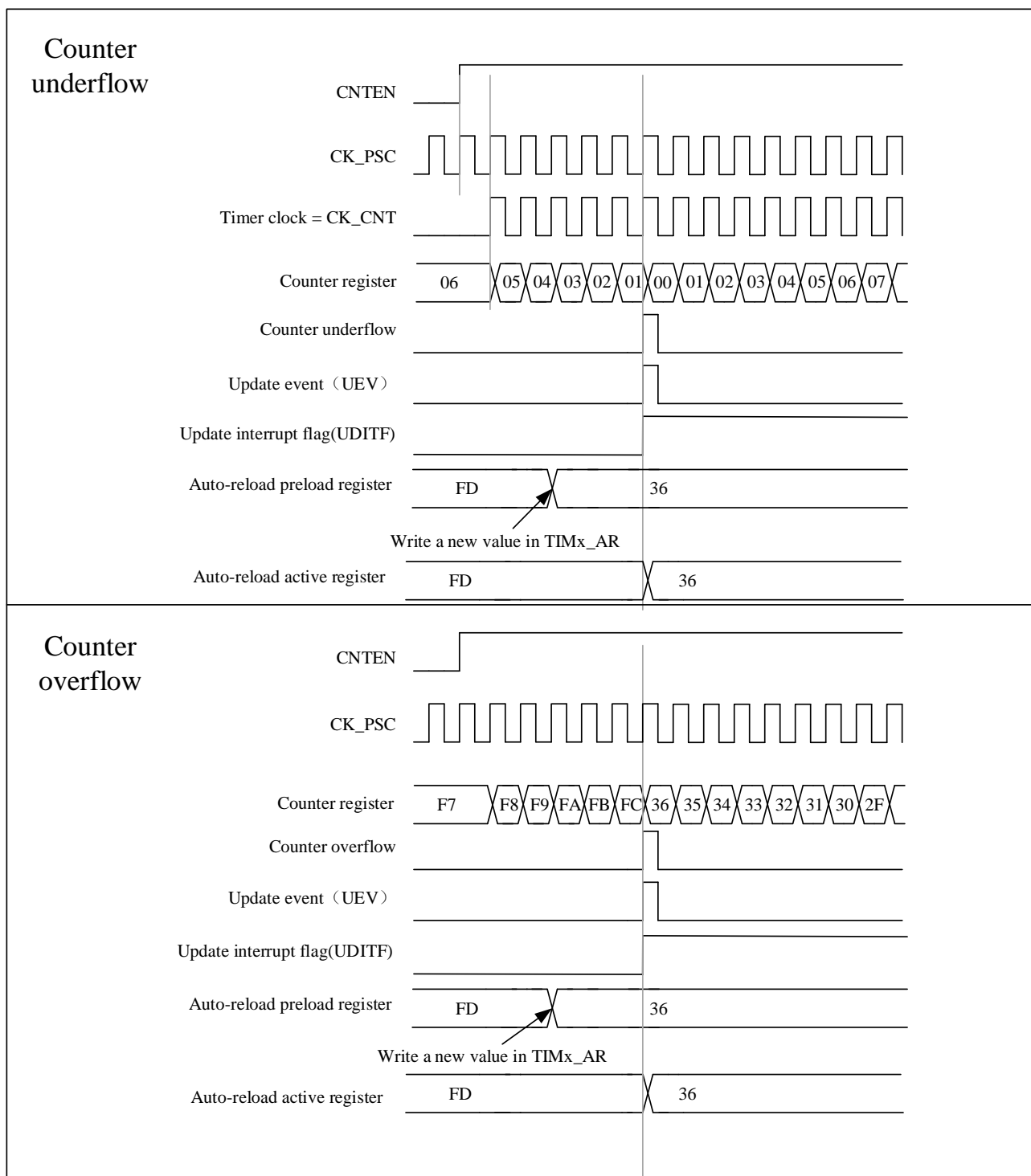


图 9-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)



9.3.2.3.2 中央对齐非对称模式

在中央对齐非对称模式下（TIMx_CTRL1.ASYMMETRIC 为 1，TIMx_CTRL1.CAMSEL[1:0]为非零），计数器从 0 计数到自动重载值（TIMx_AR）-1，并产生计数器溢出事件，然后从自动重载值计数到 1，并产生计数器向下溢出事件，然后从 0 重新开始计数。

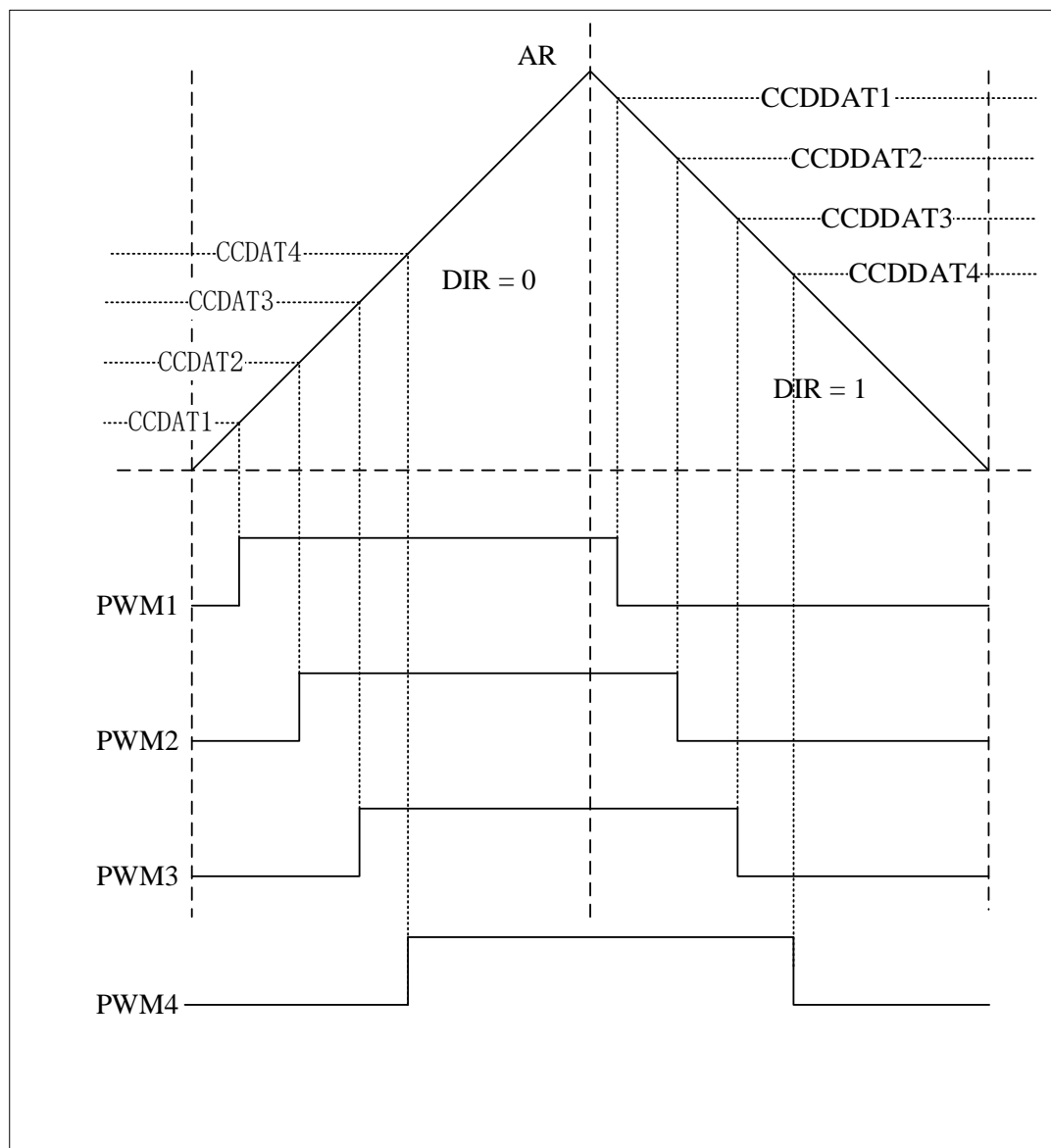
TIMx_CTRL1.DIR 值不能在此模式下写入，它由硬件更新和指定当前计数方向。

当通道不是 1,2,3,4 时, 比较值是 CC_{DATA}_x。当死区时间发生器打开时, 请注意, 当 DIR=0 时, 死区时间插入点是计数器值等于 CC_{DATA}_x (x=1,2,3,4), 当 DIR=1 时, 死区时间插入点是计数器值等于 CC_{DATA}_x (x=1,2,3,4)。

每次计数上溢和计数下溢时都会产生更新事件。或者, 也可以通过设置 TIM_x_EVTGEN.UDGN 位(通过软件或使用从模式控制器)产生更新事件。在这种情况下, 在这种情况下, 计数器从 0 重新开始计数, 预分频器的计数器也从 0 重新开始计数。

注: 如果因为计数器溢出而产生更新, 自动重载将在计数器重新载入之前被更新。

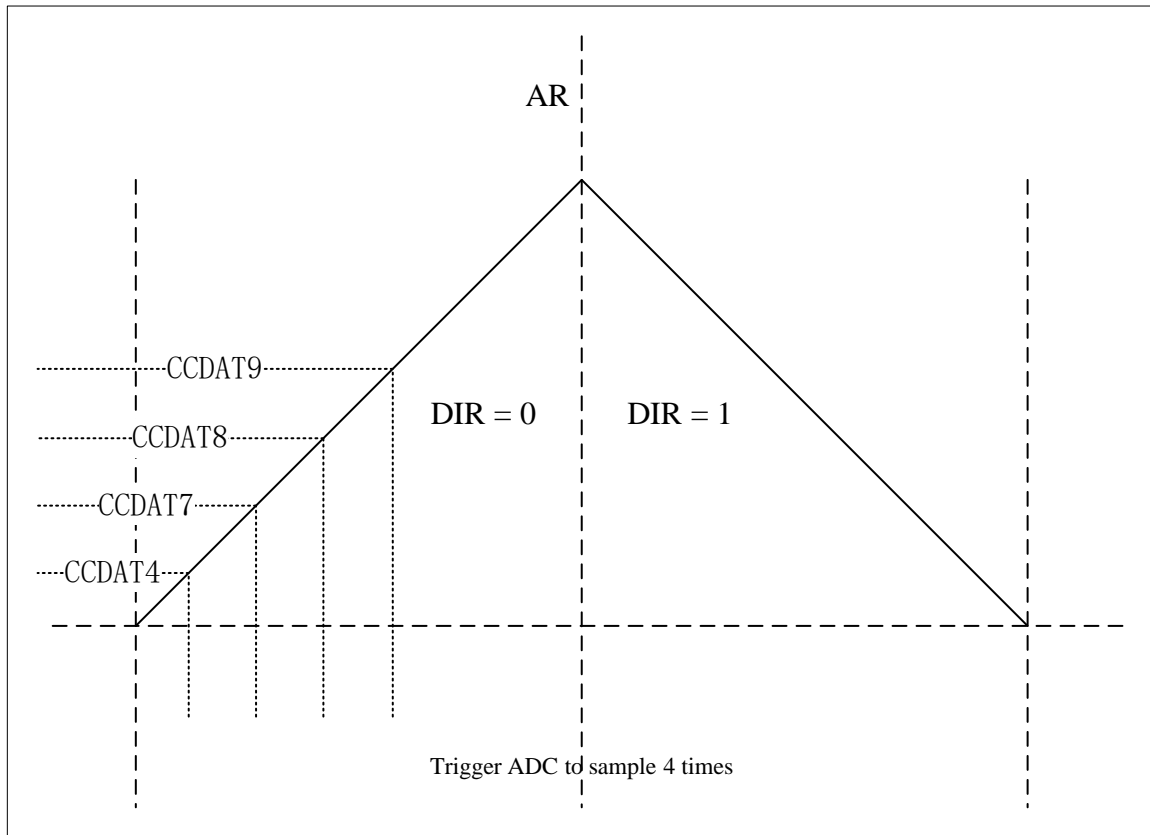
图 9-8 非对称模式对应的输出波形



由于增加了 CC7/CC8/CC9 三个通道的触发功能, 并对 CC4 的触发功能进行了修改, 现在描述 CC4/CC7/CC8/CC9 通道对 ADC 触发的描述。

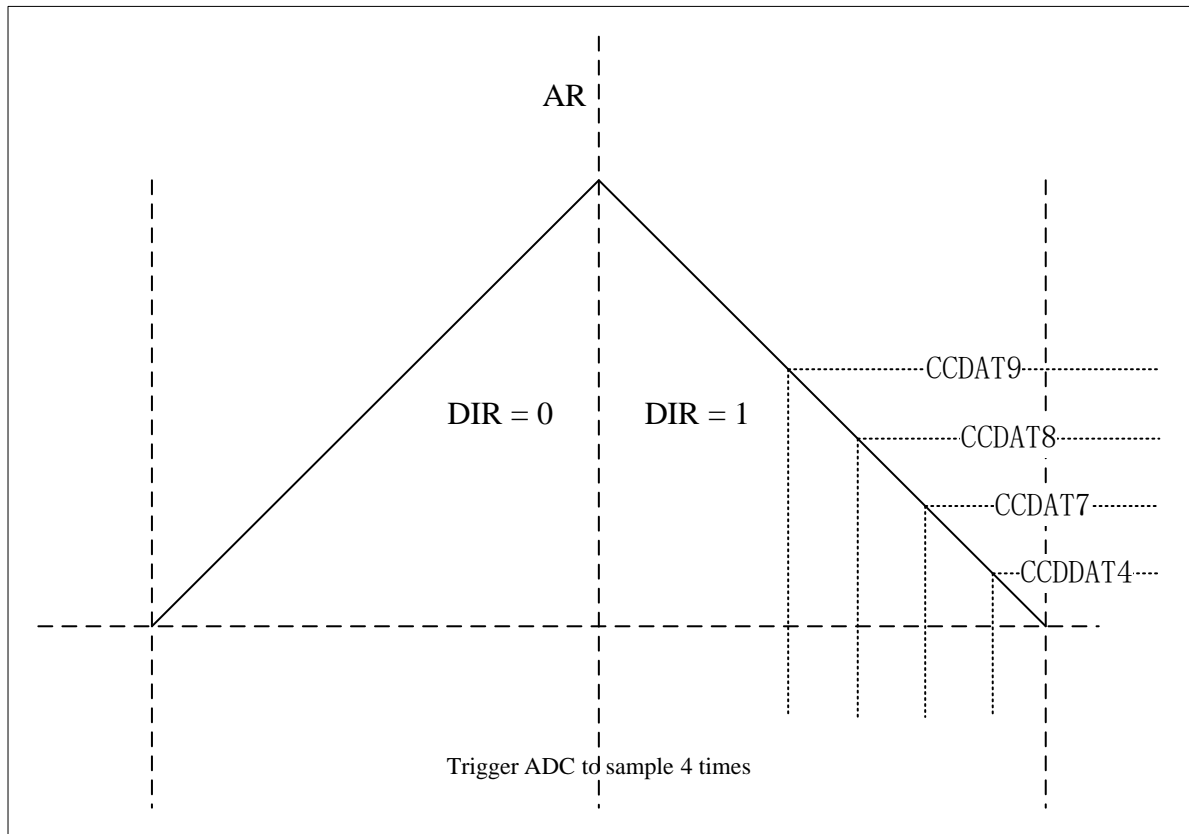
当定时器在中央对齐非对称模式下工作时, 每个通道 (CC4/CC7/CC8/CC9) 都可以单独触发 ADC。如果 TIM_x_CTRL1.CMODE[1:0]=00, 在 CC_{DATA}_x (x=4,7,8,9) 中, CC_{DATA}_x 的 CC_{DATA} 值仅在 DIR=0 时触发 ADC。

图 9-9 CCDATx(x=4,7,8,9),当 DIR = 0 时触发 ADC



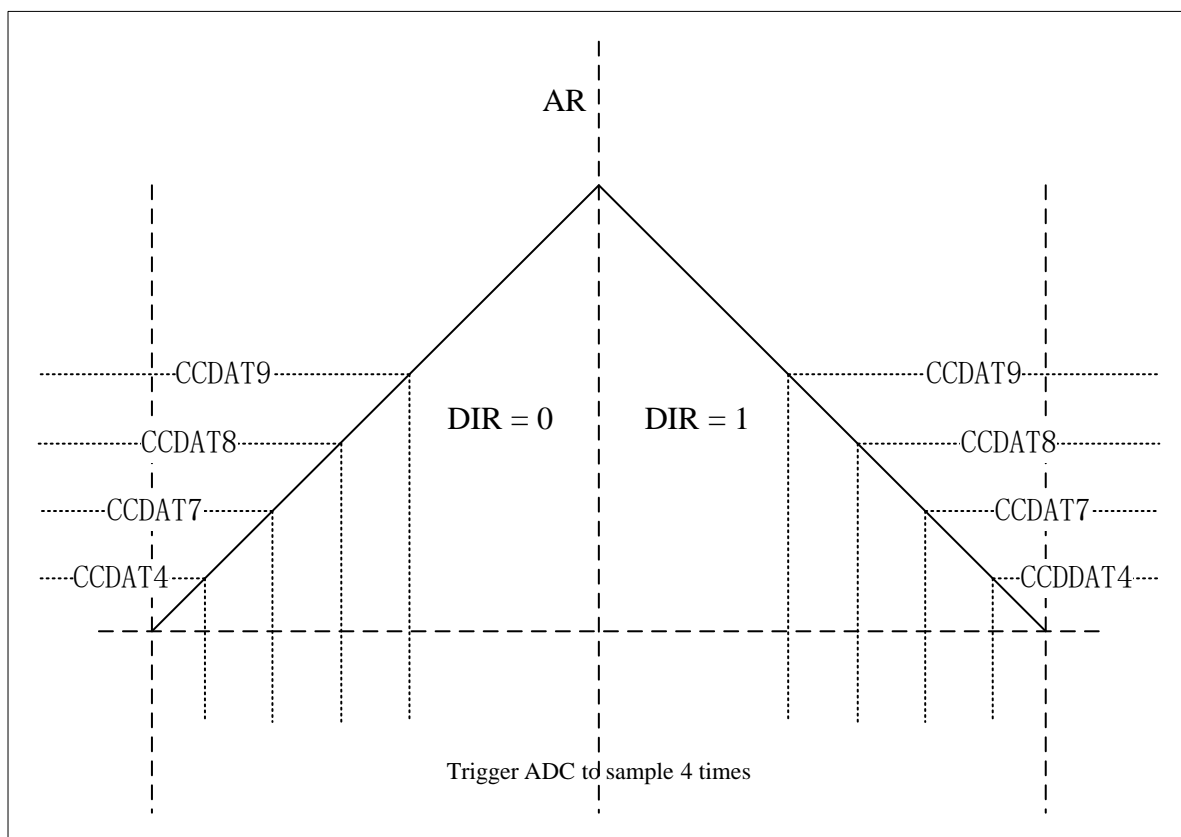
如果 TIMx_CTRL1.CMODE [1:0]=01，在 CCDATx (x=4,7,8,9) 中，CCDATx 的 CCDAT/CCDDAT 值仅在 DIR=1 时触发 ADC。

图 9-10 CCDAT_x(x=4,7,8,9), 当 DIR = 1 时触发 ADC



如果 $TIM_x_CTRL1.CMODE[1:0]=1x$, 在 $CCDAT_x$ ($x=4,7,8,9$) 中, 当 $DIR=0$ 或 $DIR=1$ 时, $CCDAT_x$ 的 $CCDAT/CCDDAT$ 值将触发 ADC.

图 9-11 CC DAT_x(x=4,7,8,9), 当 DIR = 1 或 DIR = 0 时触发 ADC



在上图中，通道 4 向上计数至 CC DAT4 或向下计数至 CC DDAT4，通道 7/8/9 向上计数或向下计数至 CC DAT7/8/9，触发有效。

9.3.3 重复计数器

第 9.3.1 章节的基本单元描述了生成更新事件（UEV）的条件。更新事件（UEV）实际上仅在重复计数器达到零时生成，这对于生成 PWM 信号非常有用。

这意味着每 N+1 计数器溢出或下溢一次，数据就会从预加载寄存器传输到影子寄存器，其中 N 是 TIM_x_REPCNT 中的值。

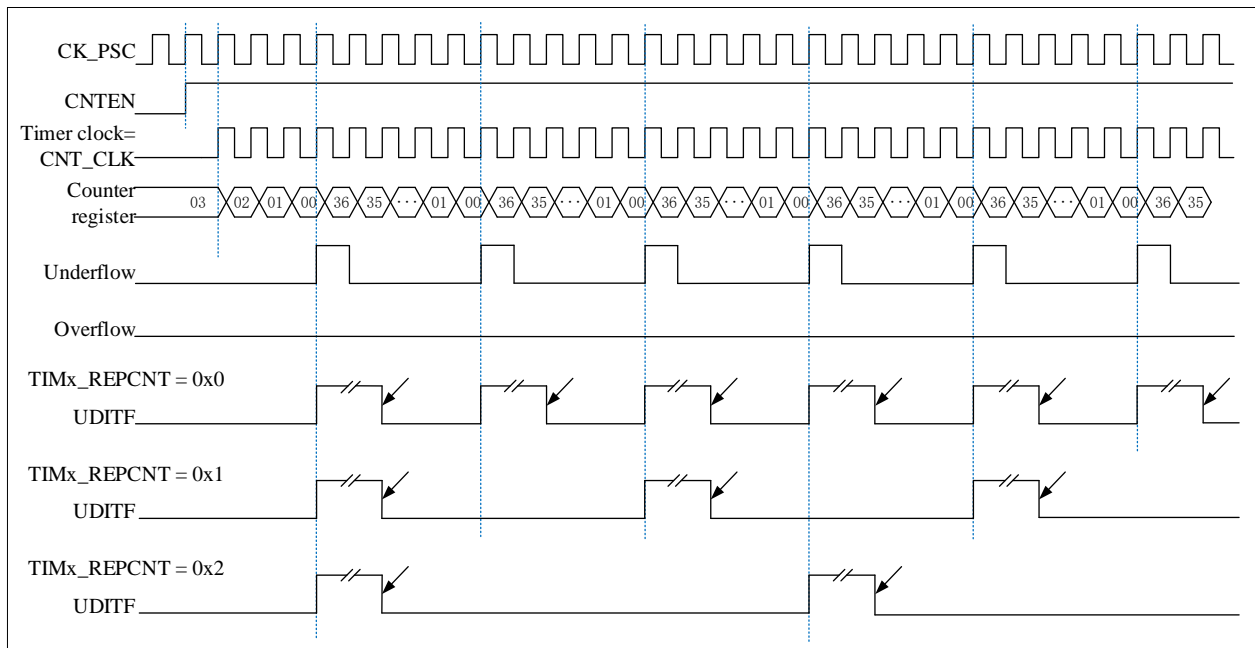
重复计数器递减：

- 在向上计数模式下，每次计数器达到最大值时，都会发生溢出
- 在向下计数模式下，每次计数器减至最小值时，都会发生下溢
- 在中央对齐模式下，每次计数上溢或下溢时

其重复率由 TIM_x_REPCNT 寄存器的值定。

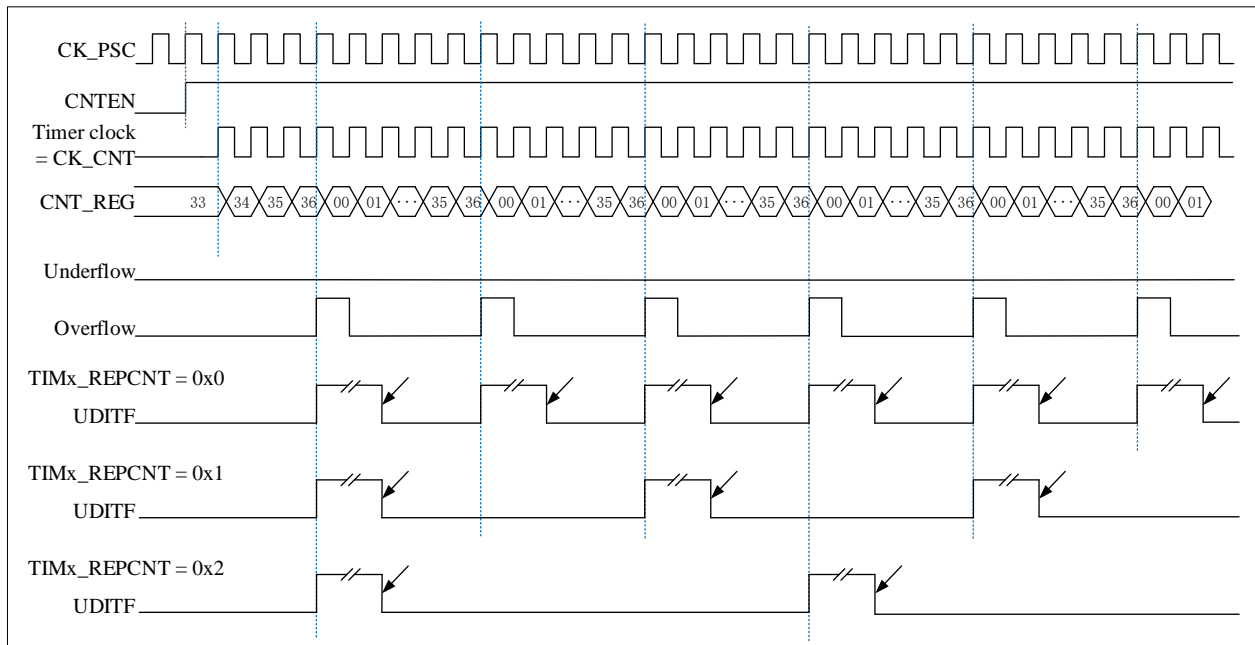
重复计数器具有自动重新加载功能。无论重复计数器的值如何，更新事件（通过从模式控制器设置 TIM_x_EVTGEN.UDGN 或硬件生成）都会立即发生。

图 9-12 向下计数模式下的重复计数时序图



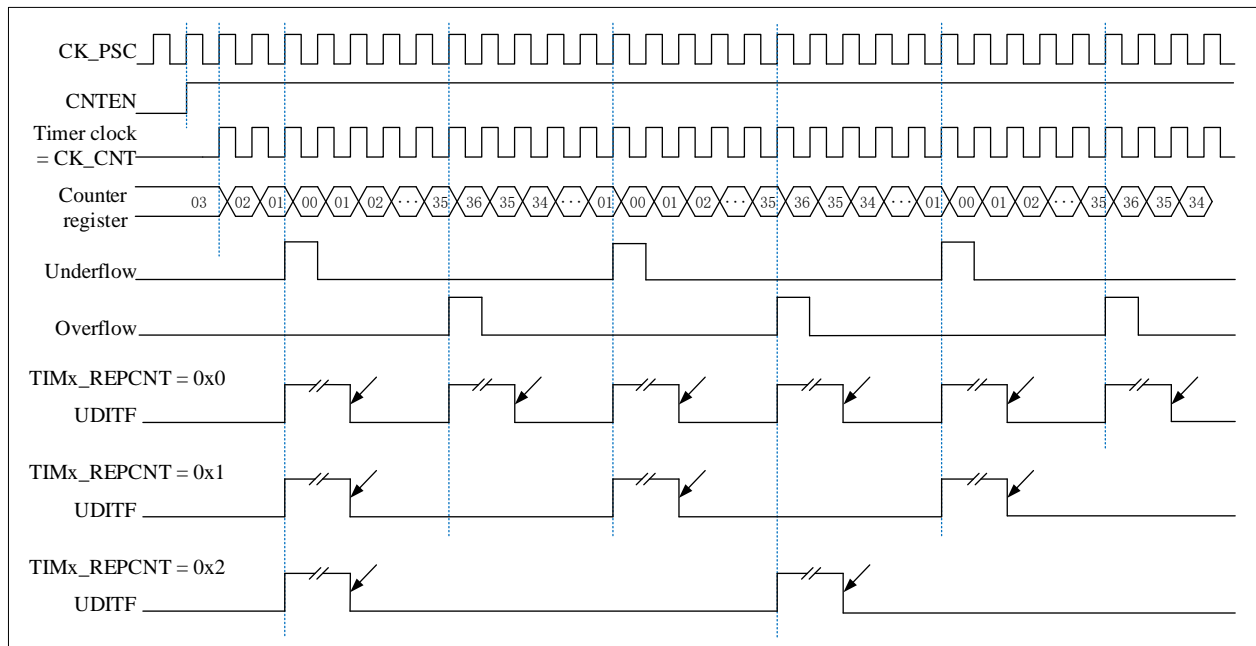
软件清除

图 9-13 向上计数模式下的重复计数时序图



软件清除

图 9-14 中央对齐模式下的重复计数时序图



软件清除

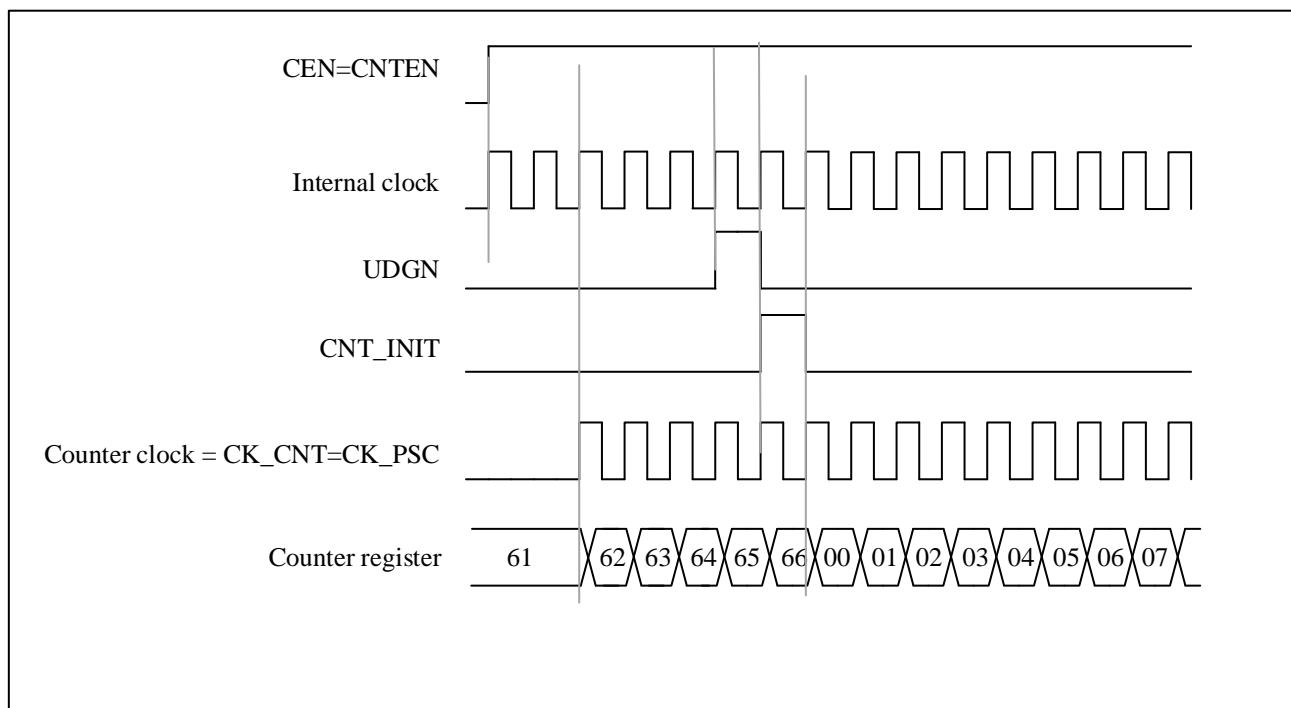
9.3.4 时钟选择

- CK_INT 高级控制定时器的内部时钟：CK_INT：
- 两种外部时钟模式：
 - 外部输入引脚
 - 外部触发输入 ETR
- 内部触发输入（ITRx）：一个定时器用作另一个定时器的预分频器

9.3.4.1 内部时钟源(CK_INT)

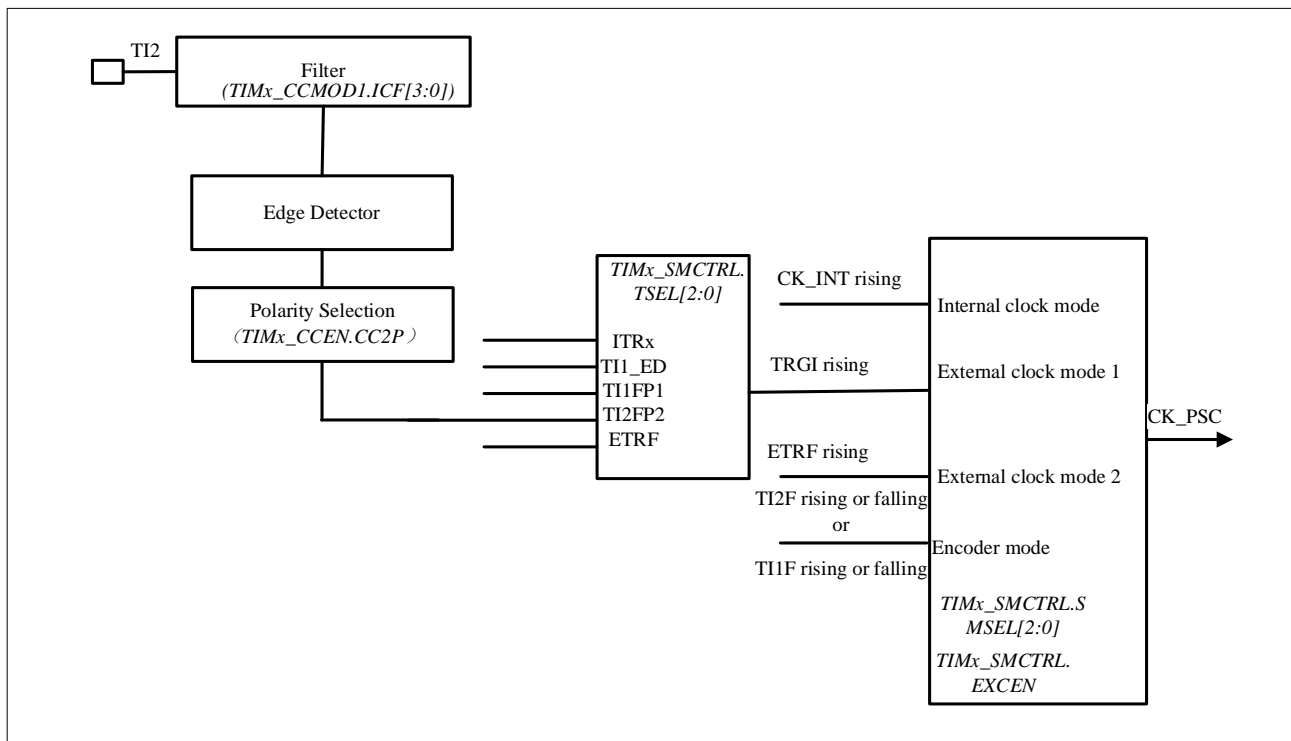
当 TIMx_SMCTRL.SMSEL 等于“000”时，从模式控制器被禁用。这三个控制位（TIMx_CTRL1.CNTEN、TIMx_CTRL1.DIR、TIMx_EVTGEN.UDGN）只能由软件改变（TIMx_EVTGEN.UDGN 除外，它保持自动清零）。前提是 TIMx_CTRL1.CNTEN 位被软写为‘1’，预分频器的时钟源由内部时钟 CK_INT 提供。

图 9-15 正常模式下的控制电路，内部时钟除以 1



9.3.4.2 外部时钟源模式 1

图 9-16 TI2 外部时钟连接示例



通过配置 TIMx_SMCTRL.SMSEL=111 选择该模式。计数器可以配置为在所选输入的时钟上升沿或下降沿进行计数。

例如，配置向上计数模式在 TI2 输入的时钟上升沿计数，配置步骤如下：

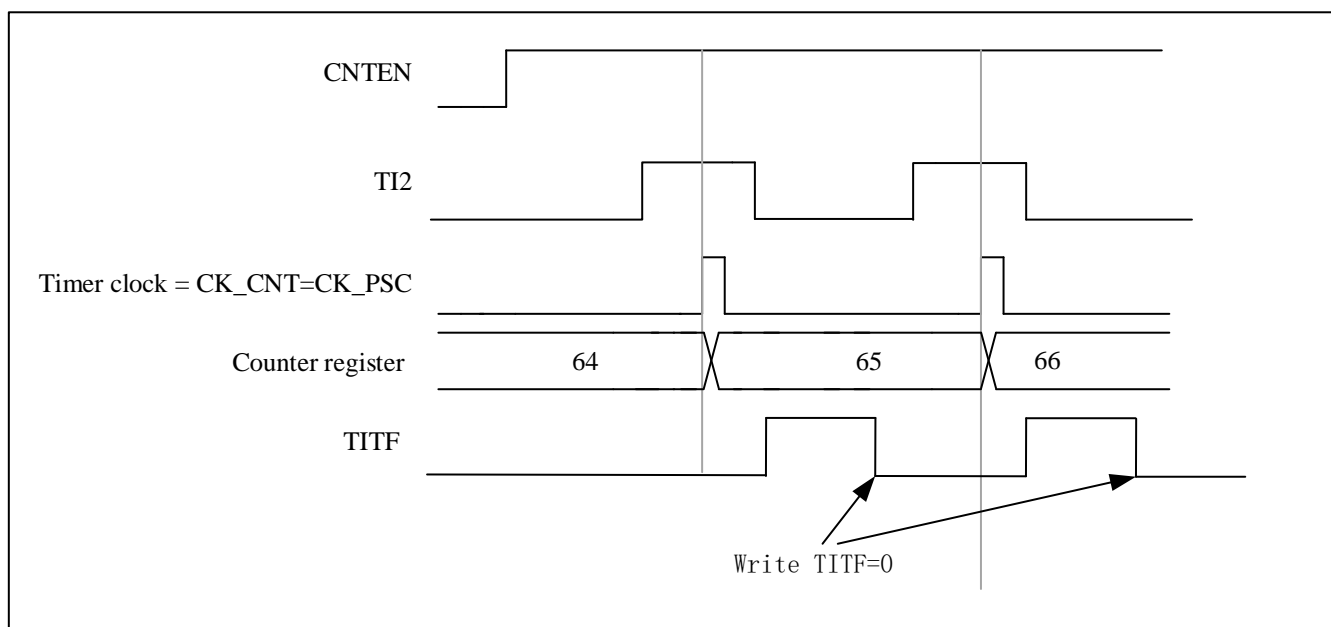
- 配置 TIMx_CCMOD1.CC2SEL 等于‘01’，CC2 通道配置为输入，IC2 映射到 TI2
- 配置 TIMx_CCEN.CC2P 等于‘0’，选择时钟上升沿极性
- 通过配置 TIMx_CCMOD1.IC2F[3:0] 选择输入滤波器带宽（如果不需要滤波器，保持 IC2F 位为‘0000’）
- 配置 TIMx_SMCTRL.SMSEL 等于‘111’，选择定时器外部时钟模式 1
- 配置 TIMx_SMCTRL.TSEL 等于‘110’，选择 TI2 作为触发输入源
- 配置 TIMx_CTRL1.CNTEN 等于‘1’以启动计数器

注意：捕获预分频器不用于触发，所以不需要配置

当定时器时钟的上升沿出现在 TI2=1 时，计数器计数一次并且 TIMx_STS.TITF 标志被拉高。

TI2 的上升沿与计数器实际时钟之间的延迟取决于 TI2 输入端的再同步电路。

图 9-17 外部时钟模式 1 的控制电路

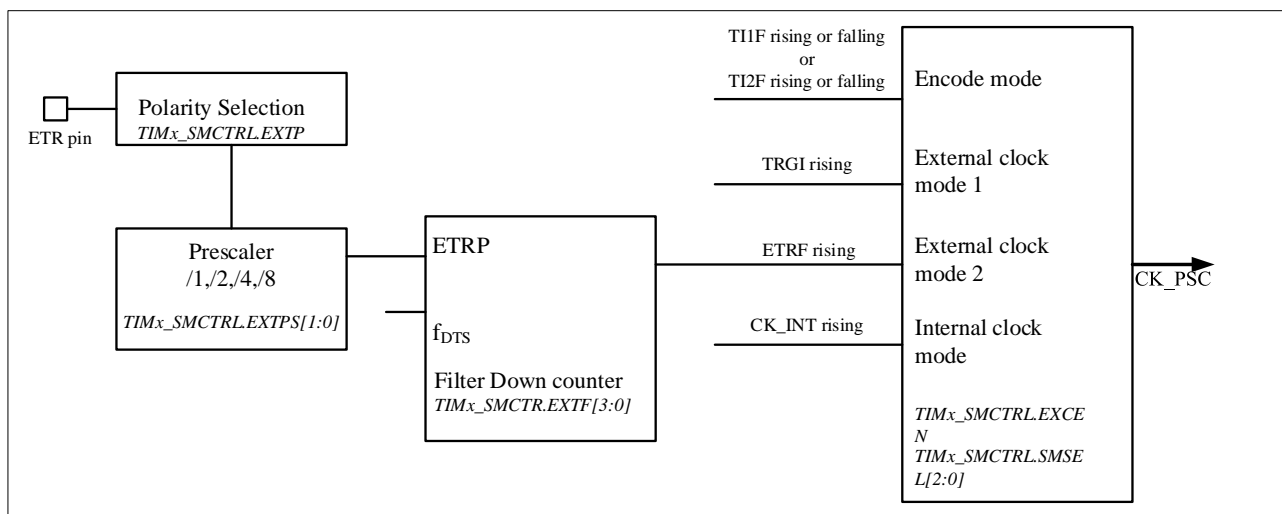


9.3.4.3 外部时钟源模式 2

此模式由 TIMx_SMCTRL.EXCEN 选择等于 1。计数器可以在外部触发输入 ETR 的每个上升沿或下降沿计数。

下图为外部时钟源模式 2 的外部触发输入模块示意图。

图 9-18 外部触发输入框图

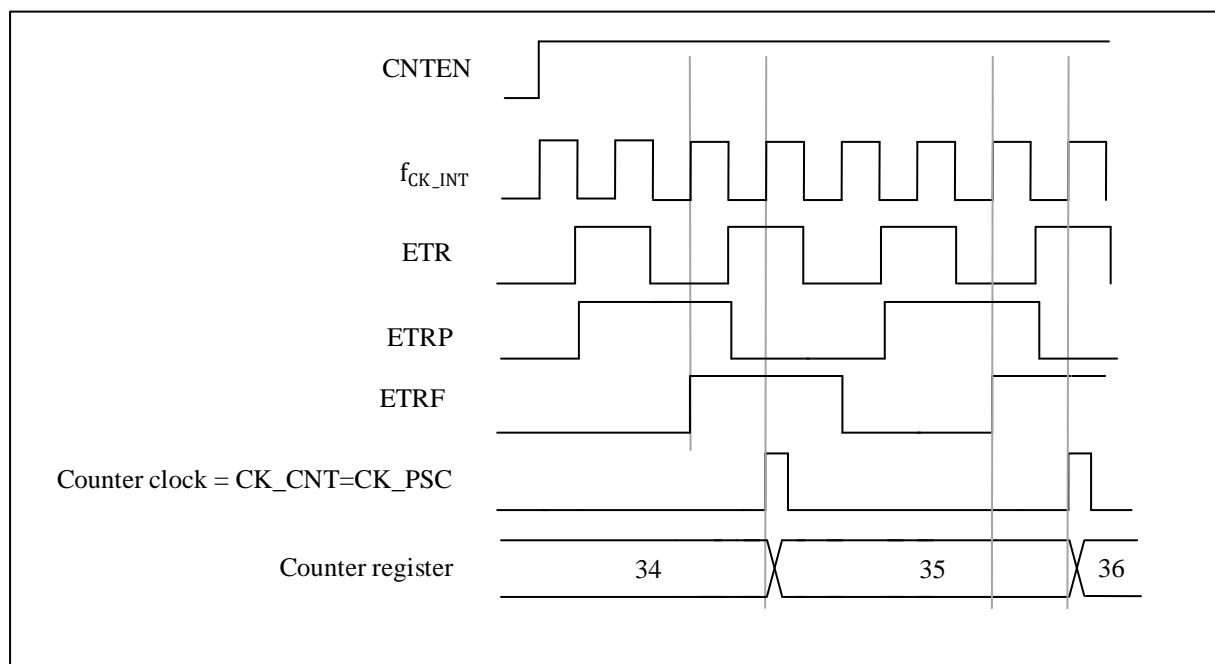


例如，使用以下配置步骤使向上计数器在 ETR 上每 2 个上升沿计数一次。

- 由于在这种情况下不需要过滤器，因此使 TIMx_SMCTRL.EXTF[3:0] 等于‘0000’
- 通过使 TIMx_SMCTRL.EXTPS[1:0] 等于 ‘01’ 来配置预分频器
- 通过设置 TIMx_SMCTRL.EXTP 等于‘0’来选择 ETR 引脚的极性，ETR 的上升沿有效
- 外部时钟模式 2 通过设置 TIMx_SMCTRL.EXCEN 等于‘1’来选择
- 通过设置 TIMx_CTRL1.CNTEN 等于“1”启动计数器。

计数器每 2 个 ETR 上升沿计数一次。ETR 的上升沿与计数器的实际时钟之间的延迟是由于 ETRP 信号上的再同步电路造成的。

图 9-19 外部时钟模式 2 的控制电路

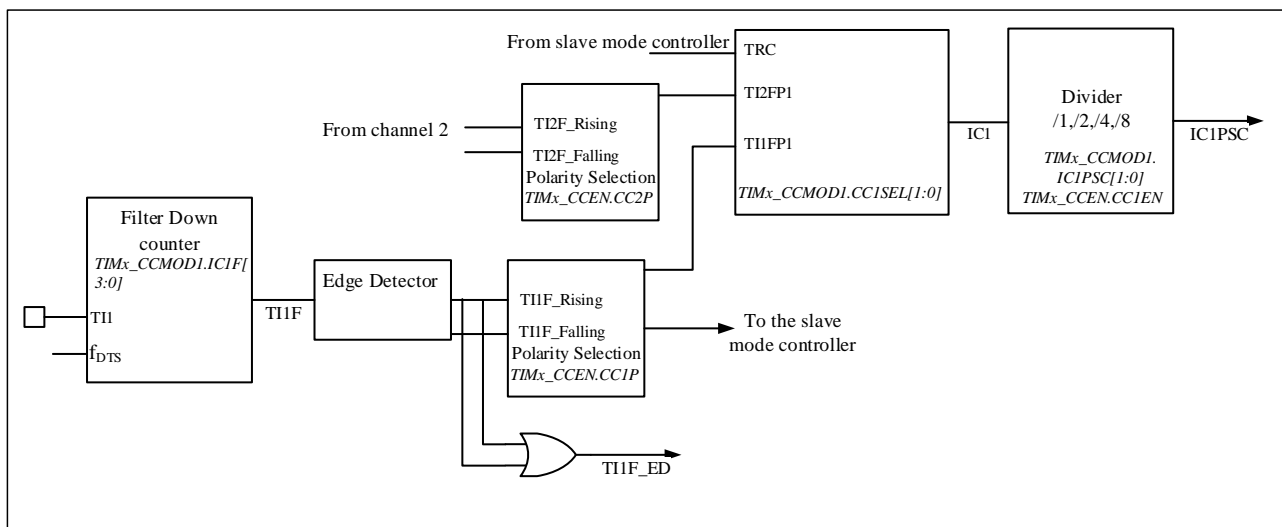


9.3.5 捕获/比较通道

捕获/比较通道包括捕获/比较寄存器和影子寄存器。输入部分由数字滤波器、多路复用器和预分频器组成。输出部分包括比较器和输出控制。

输入信号 TIx 被采样和滤波以产生信号 $TIxF$ 。然后由极性选择功能的边沿检测器生成信号 ($TIxF_rising$ 或 $TIxF_falling$)，其极性由 $TIMx_CCEN.CCxP$ 位选择。该信号可用作从模式控制器的触发输入。同时，信号 ICx 经过分频后送入捕获寄存器。下图显示了捕获/比较通道的框图。

图 9-20 捕获/比较通道（例如：通道 1 输入级）



输出部分生成一个中间波形 $OCxRef$ （高电平有效）作为参考。极性作用在链的末端。

图 9-21 捕获/比较通道 1 主电路

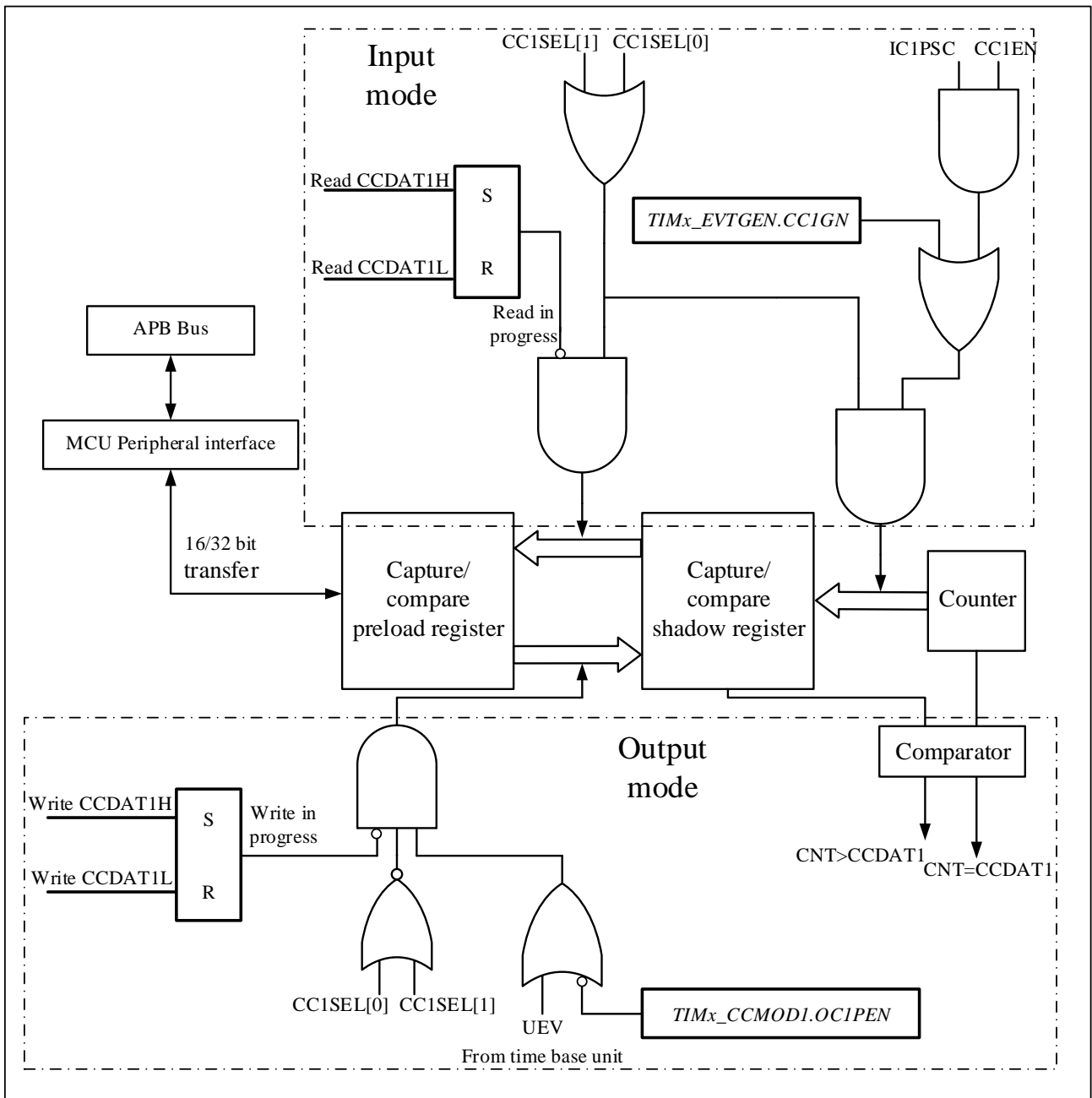
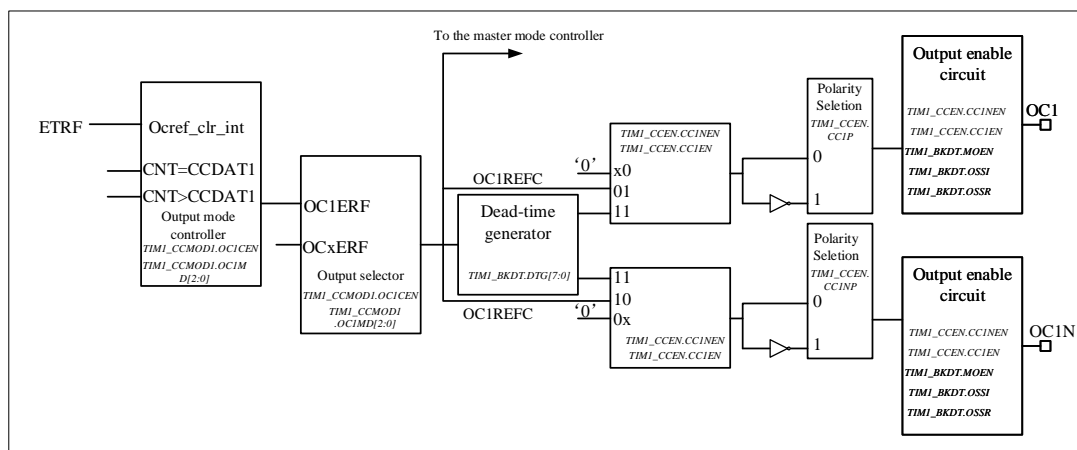


图 9-22 通道 x 的输出部分 (x= 1,2,3,4; 以通道 1 为例子)



在捕获/比较时，读取和写入始终访问预加载的寄存器。两个具体工作流程如下：

在捕获模式下，捕获实际上是在影子寄存器中完成的，然后将影子寄存器中的值复制到预加载寄存器中。

在比较模式下，与捕获模式相反，预加载寄存器的值被复制到影子寄存器中，并与计数器进行比较。

9.3.6 输入捕获模式

在捕获模式下，TIMx_CCxDATx 寄存器用于在检测到 ICx 信号后锁存计数器值。

有一个捕获中断标志 TIMx_STS.CCxITF，如果相应的中断使能被拉高，它可以发出中断或 DMA 请求。

TIMx_STS.CCxITF 位在发生捕获事件时由硬件设置，并由软件或读取 TIMx_CCxDATx 寄存器清零。

当 TIMx_CCxDATx 寄存器中的计数器值被捕获并且 TIMx_STS.CCxITF 被拉高时，重复捕获标志 TIMx_STS.CCxOCF 设置为 1。与前者不同，TIMx_STS.CCxOCF 通过向其写入 0 来清除。

为实现 TI1 输入的上升沿将计数器值捕获到 TIMx_CCxDAT1 寄存器中，配置流程如下：

■ 选择有效输入：

将 TIMx_CCMOD1.CC1SEL 配置为“01”。此时输入为 CC1 通道，IC1 映射到 TI1。

■ 编程所需的输入滤波器持续时间：

通过配置 TIMx_CCMODx.ICxIF 位来定义 TI1 输入的采样频率和数字滤波器的长度。示例：如果输入信号抖动多达 5 个内部时钟周期，我们必须选择比这 5 个时钟周期更长的滤波器持续时间。当检测到具有新电平的 8 个连续样本（以 f_{DTS} 频率采样）时，我们可以验证 TI1 上的转换。然后配置 TIMx_CCMOD1.IC1F 到“0011”

■ 通过配置 TIMx_CCEN.CC1P=0，选择上升沿作为 TI1 通道的有效跳变极性

■ 配置输入预分频器。在本例中，配置 TIMx_CCMOD1.IC1PSC=‘00’ 以禁用预分频器，因为我们想要捕获每个有效转换

■ 通过配置 TIMx_CCEN.CC1EN=‘1’ 启用捕获。

如果要使能 DMA 请求，可以配置 TIMx_DINTEN.CC1DEN=1。如果要使能相关中断请求，可以配置 TIMx_DINTEN.CC1IEN=1。

9.3.7 PWM 输入模式

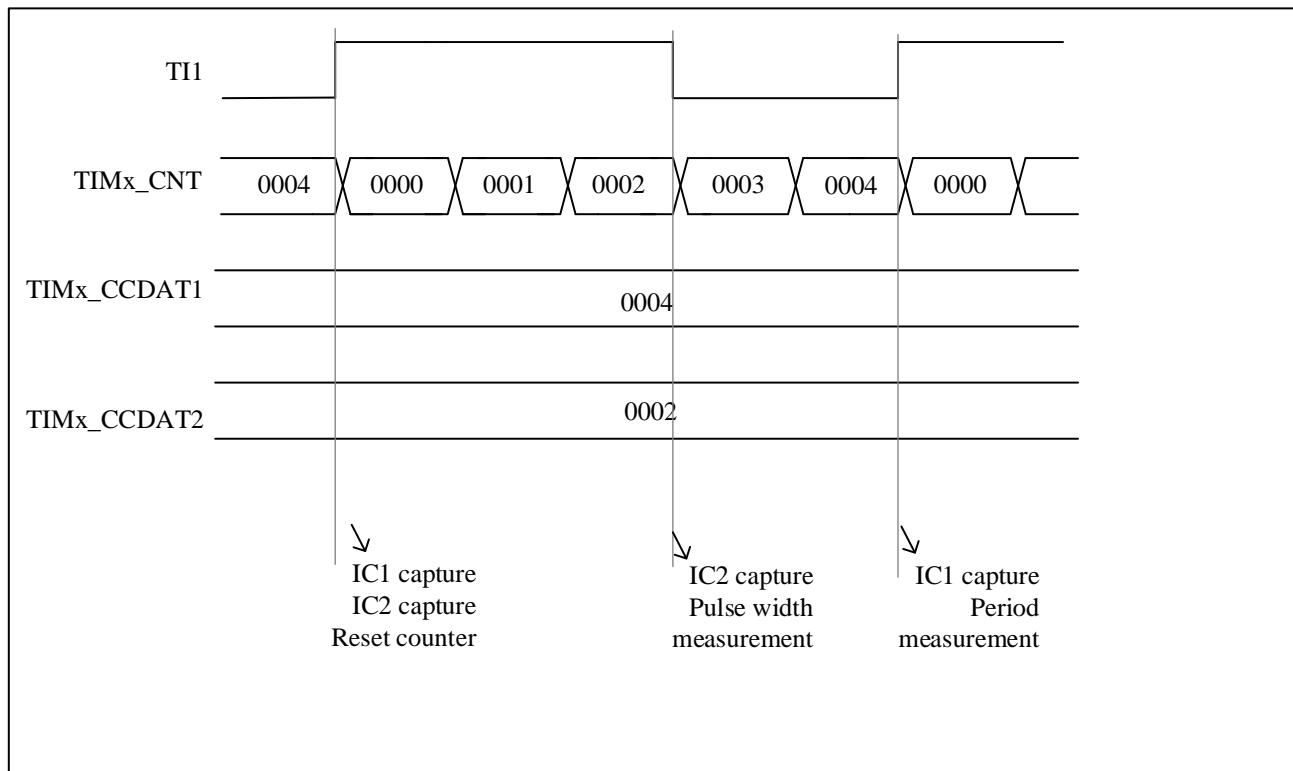
PWM 输入模式和普通输入捕获模式有一些区别，包括：

- 两个 ICx 信号映射到同一个 TIx 输入
- 两个 ICx 信号在极性相反的边沿有效
- 选择两个 TIxFP 信号之一作为触发输入
- 从机模式控制器配置为复位模式

例如，下面的配置流程可以用来知道 TI1 上 PWM 信号的周期和占空比（这取决于 CK_INT 的频率和预分频器的值）。

- 配置 TIMx_CCMOD1.CC1SEL 等于 ‘01’ 以选择 TI1 作为 TIMx_CCDAT1 的有效输入
- 配置 TIMx_CCEN.CC1P 等于 ‘0’ 选择滤波定时器输入 1(TI1FP1) 的有效极性，在上升沿有效
- 配置 TIMx_CCMOD1.CC2SEL 等于 ‘10’ 选择 TI1 作为 TIMx_CCDAT2 的有效输入
- 配置 TIMx_CCEN.CC2P 等于 1 选择滤波定时器输入 2(TI1FP2)的有效极性，下降沿有效
- 配置 TIMx_SMCTRL.TSEL=101 选择 Filtered timer input 1 (TI1FP1) 作为有效触发输入
- 配置 TIMx_SMCTRL.SMSEL=100 配置从模式控制器为复位模式
- 配置 TIMx_CCEN.CC1EN=1 和 TIMx_CCEN.CC2EN=1 以启用捕获

图 9-23 PWM 输入模式时序



由于只有滤波器定时器输入 1 (TI1FP1) 和滤波器定时器输入 2 (TI2FP2) 连接到从模式控制器，因此 PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号一起使用。

9.3.8 强制输出模式

在输出模式（TIMx_CCMODx.CCxSEL=00）下，软件可以直接将输出比较信号强制为有效或无效电平。

用户可以设置 TIMx_CCMODx.OCxMD=101 强制输出比较信号为有效电平。OCxREF 将被强制为高电平，OCx 得到与 CCxP 极性位相反的值。另一方面，用户可以设置 TIMx_CCMODx.OCxMD=100 强制输出比较信号为无效电平，即 OCxREF 被强制为低电平。

在此模式下，TIMx_CCDATx 影子寄存器和计数器的值仍然相互比较。

输出比较寄存器 TIMx_CCDATx 和计数器 TIMx_CNT 之间的比较对 OCxREF 没有影响。并且仍然可以设置标志。因此，仍然可以发送中断和 DMA 请求。

9.3.9 输出比较模式

用户可以使用此模式来控制输出波形，或指示一段时间已过。

当捕获/比较寄存器和计数器的值相同时，输出比较函数的操作如下：

- TIMx_CCMODx.OCxMD 为输出比较模式，TIMx_CCEN.CCxP 为输出极性。当比较匹配时，如果设置 TIMx_CCMODx.OCxMD=000，则输出管脚将保持其电平；如果设置 TIMx_CCMODx.OCxMD=001，则设置输出管脚有效；如果设置 TIMx_CCMODx.OCxMD=010，则输出管脚将为 设置为无效；如果设置 TIMx_CCMODx.OCxMD=011，则输出引脚将设置为翻转。
- 设置 TIMx_STS.CCxITF
- 如果用户设置了 TIMx_DINTEN.CCxIEN，将产生相应的中断
- 如果用户设置 TIMx_DINTEN.CCxDEN 并设置 TIMx_CTRL2.CCDSEL 选择 DMA 请求，将发送 DMA 请求

用户可以设置 TIMx_CCMODx.OCxPEN 来选择是否使用捕获/比较预加载寄存器（TIMx_CCDATx）来选择捕获/比较影子寄存器。

时间分辨率是计数器的一个计数周期。

在单脉冲模式下，输出比较模式也可用于输出单脉冲。

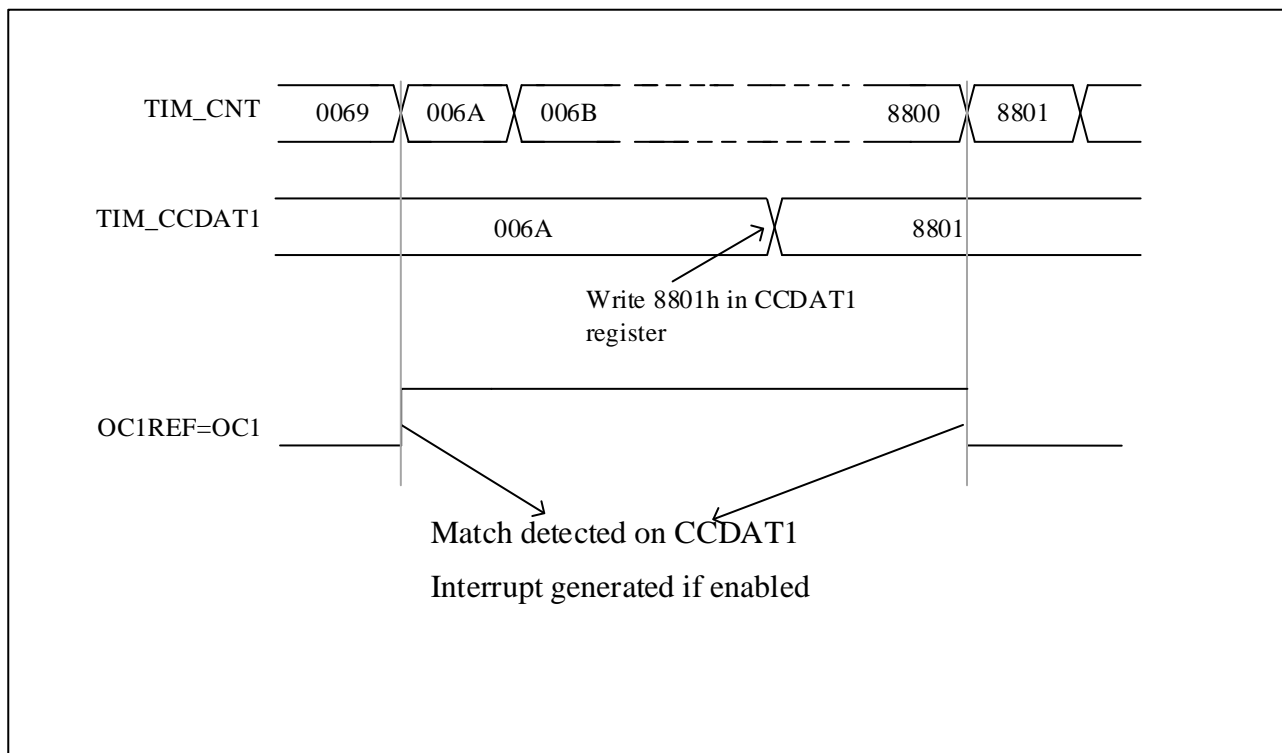
以下是输出比较模式的配置步骤：

- 首先，用户应该选择计数器时钟
- 其次，用所需数据设置 TIMx_AR 和 TIMx_CCDATx
- 如果用户需要产生中断，设置 TIMx_DINTEN.CCxIEN
- 然后通过设置 TIMx_CCEN.CCxP、TIMx_CCMODx.OCxMD、TIMx_CCEN.CCxEN 等选择输出模式
- 最后，设置 TIMx_CTRL1.CNTEN 启用计数器

用户可以随时通过设置 TIMx_CCDATx 来更新输出波形，只要不启用预加载寄存器。否则，TIMx_CCDATx 影子寄存器将在下一次更新事件中更新。

例如：

图 9-24 输出比较模式，开启 OC1



9.3.10 PWM 模式

用户可以使用 PWM 模式产生一个信号，其占空比由 TIMx_CCDATx 寄存器的值决定，其频率由 TIMx_AR 寄存器的值决定。并且取决于 TIMx_CTRL1.CAMSEL 的值，TIM 可以在边沿对齐模式或中央对齐模式下产生 PWM 信号。

用户可以通过设置 TIMx_CCMODx.OCxMD=110 或设置 TIMx_CCMODx.OCxMD=111 来设置 PWM 模式 1 或 PWM 模式 2。要能使预加载寄存器，用户必须设置相应的 TIMx_CCMODx.OCxPEN。然后设置 TIMx_CTRL1.ARPEN 自动重载预加载寄存器。

用户可以通过设置 TIMx_CCEN.CCxP 来设置 OCx 的极性。另一方面，要能使 OCx 的输出，用户需要在 TIMx_CCEN 和 TIMx_BKDT 中设置 CCxEN、CCxNEN、MOEN、OSSI 和 OSSR 的值的组合。

当 TIM 处于 PWM 模式时，TIMx_CNT 和 TIMx_CCDATx 的值总是相互比较。

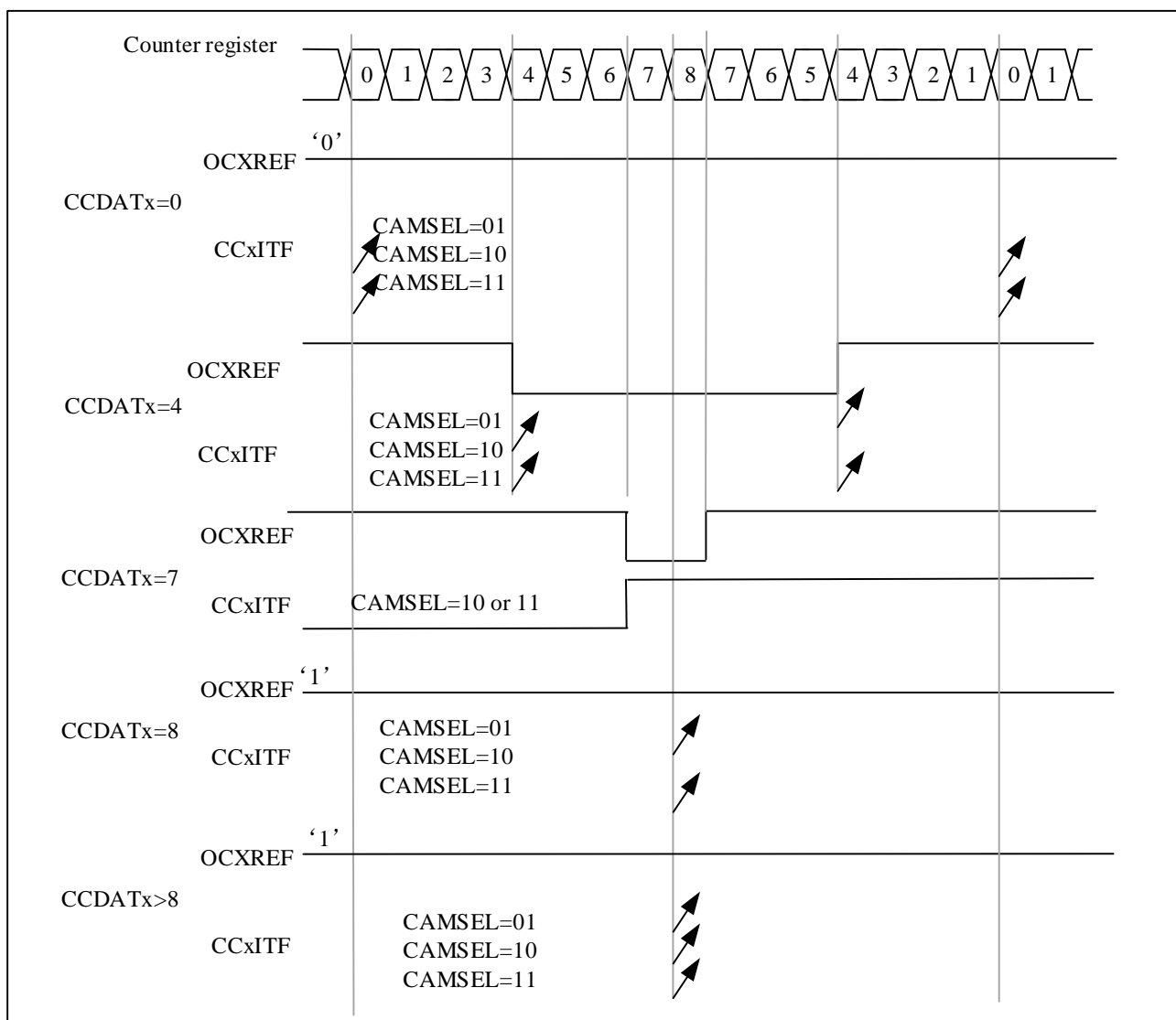
只有当更新事件发生时，预加载寄存器才会转移到影子寄存器。因此，用户必须在计数器开始计数之前通过设置 TIMx_EVTGEN.UDGN 来复位所有寄存器。

9.3.10.1 PWM 中央对齐模式

如果用户设置 TIMx_CTRL1.CAMSEL 等于 01、10 或 11，PWM 中央对齐模式将被激活。比较标志的设置取决于 TIMx_CTRL1.CAMSEL 的值。设置比较标志的情况有 3 种，仅当计数器向上计数时，仅当计数器向下计数时，或当计数器向上计数和向下计数时。用户不应通过软件修改 TIMx_CTRL1.DIR，它是由硬件更新的。

中央对齐 PWM 波形示例如下，波形设置为：TIMx_AR=8，PWM 模式 1，当计数器向下计数对应 TIMx_CTRL1.CAMSEL=01 时设置比较标志。

图 9-25 中央对齐的 PWM 波形 (AR=8)



使用中央对齐模式时用户应注意的事项如下：

- 计数器向上或向下计数取决于 `TIMx_CTRL1.DIR` 的值。 注意不要同时更改 `DIR` 和 `CAMSEL` 位
- 用户在中央对齐模式下不要写计数器，否则会导致意想不到的结果。 例如：
 - ◆ 如果写入计数器的值为 0 或者是 `TIMx_AR` 的值，则方向会被更新，但不会产生更新事件
 - ◆ 如果写入计数器的值大于自动重载的值，则方向不会更新
- 为了安全起见，建议用户在启动计数器之前设置 `TIMx_EVTGEN.UDGN` 以通过软件生成更新，并且在计数器运行时不要写入计数器

9.3.10.2 PWM 中央对齐非对称模式

关于 PWM 中央对齐非对称模式请查阅9.3.2.3.2。

9.3.10.3 PWM 边沿对齐模式

边沿对齐模式有两种配置，向上计数和向下计数。

- 向上计数

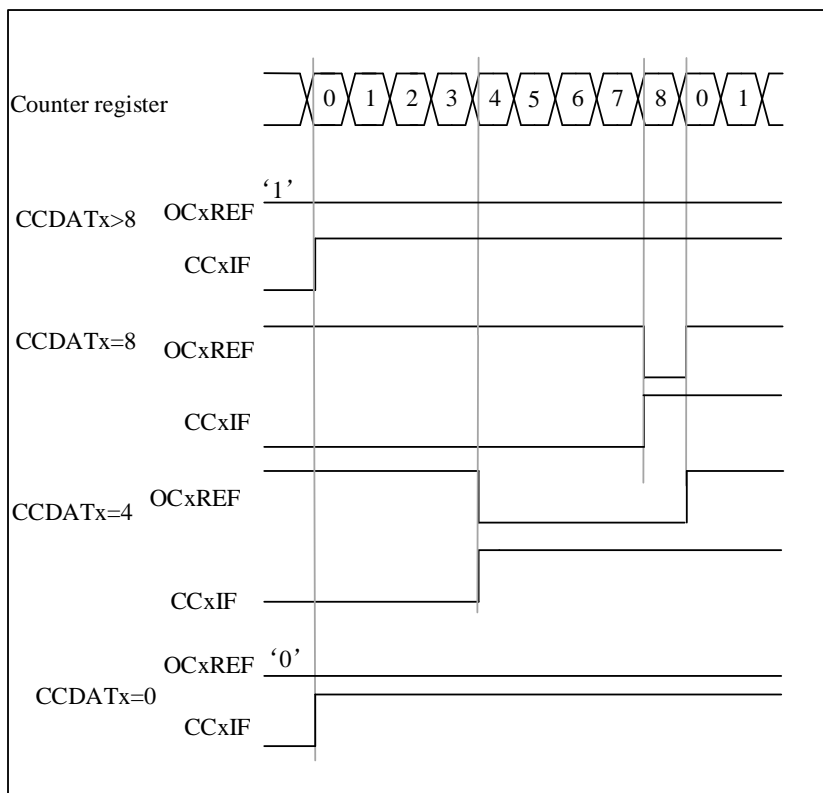
用户可以设置 $TIMx_CTRL1.DIR=0$ 使计数器向上计数。

PWM 模式 1 的示例：

当 $TIMx_CNT < TIMx_CCDATx$ 时， $OCxREF$ 为高电平，否则为低电平。如果 $TIMx_CCDATx$ 中的比较值大于自动重载值，则 $OCxREF$ 将保持为 1。相反，如果比较值为 0，则 $OCxREF$ 将保持为 0。

当 $TIMx_AR=8$ 时，PWM 波形如下：

图 9-26 边沿对齐 PWM 波形 (AR=8)



● 向下计数

用户可以设置 $TIMx_CTRL1.DIR=1$ 使计数器向下计数。

PWM 模式 1 的示例：

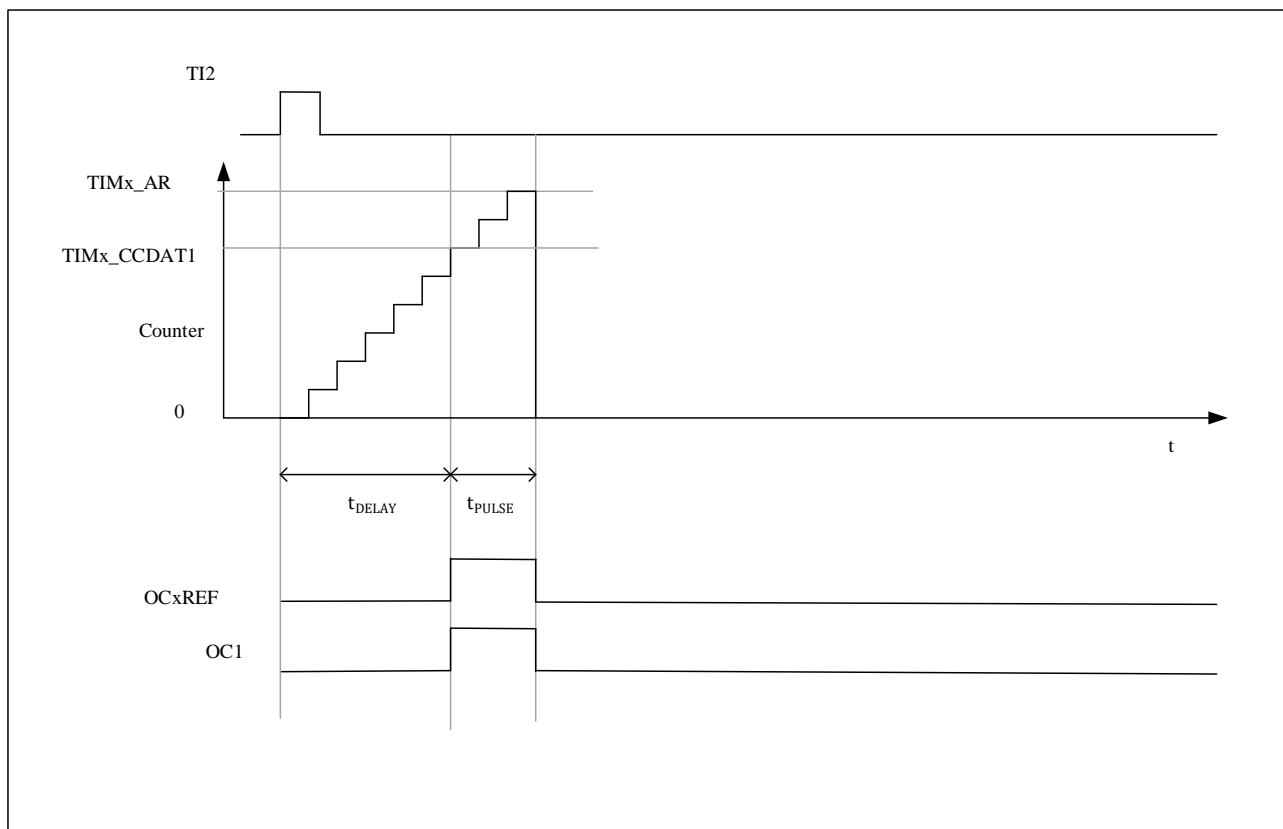
当 $TIMx_CNT > TIMx_CCDATx$ 时， $OCxREF$ 为低电平，否则为高电平。如果 $TIMx_CCDATx$ 中的比较值大于自动重载值，则 $OCxREF$ 将保持为 1。

注：若第 n 个 PWM 周期 $CCDATx$ 影子寄存器 $\geq AR$ 值，第 $n+1$ 个 PWM 周期 $CCDATx$ 的影子寄存器值是 0。在第 $n+1$ 个 PWM 周期的计数器为 0 的时刻，虽然计数器 = $CCDATx$ 影子寄存器的值 = 0， $OCxREF = '0'$ ，但不会产生比较事件。

9.3.11 单脉冲模式

在单脉冲模式(ONEPM)中，接收到触发信号，经过可控延迟 t_{DELAY} 后产生脉宽可控的脉冲 t_{PULSE} 。输出模式需要配置为输出比较模式或 PWM 模式。选择单脉冲模式后，计数器会在更新事件 UEV 产生后停止计数。

图 9-27 单脉冲模式示例



以下是单脉冲模式的示例：

从 TI2 输入检测到上升沿触发，延迟 t_{DELAY} 后在 OC1 上产生宽度为 t_{PULSE} 的脉冲。

1. 计数器配置：向上计数，计数器 $\text{TIMx_CNT} < \text{TIMx_CCDAT1} \leq \text{TIMx_AR}$ ；
2. TI2FP2 映射到 TI2, $\text{TIMx_CCMOD1.CC2SEL} = '01'$ ；TI2FP2 配置为上升沿检测, $\text{TIMx_CCEN.CC2P} = '0'$ ；
3. TI2FP2 充当从模式控制器的触发器（TRGI）并启动计数器， $\text{TIMx_SMCTRL.TSEL} = '110'$ ， $\text{TIMx_SMCTRL.SMSEL} = '110'$ （触发模式）；
4. TIMx_CCDAT1 写入要延迟的计数值（ t_{DELAY} ）， $\text{TIMx_AR} - \text{TIMx_CCDAT1}$ 为脉宽 t_{PULSE} 的计数值；
5. 配置 $\text{TIMx_CTRL1.ONEPM} = 1$ 使能单脉冲模式，配置 $\text{TIMx_CCMOD1.OC1MD} = '111'$ 选择 PWM2 模式；
6. 等待 TI2 有外部触发事件，OC1 输出一个单脉冲波形；

9.3.11.1 特殊情况：OCx 快速使能：

在单脉冲模式下，通过 TIx 输入检测到一个边沿，并触发计数器开始计数到比较值，然后输出一个脉冲。这些操作限制了可以达到的最小延迟 t_{DELAY} 。

您可以设置 $\text{TIMx_CCMODx.OCxFEN} = 1$ 开启 OCx 快速使能，在触发上升沿后，OCxREF 信号将被强制转换为与比较匹配立即发生的电平相同的电平，而不管比较结果如何。OCxFEN 快速使能仅在通道模式配置为 PWM1 和 PWM2 模式时生效。

9.3.12 在外部事件上清除 OCxREF 信号

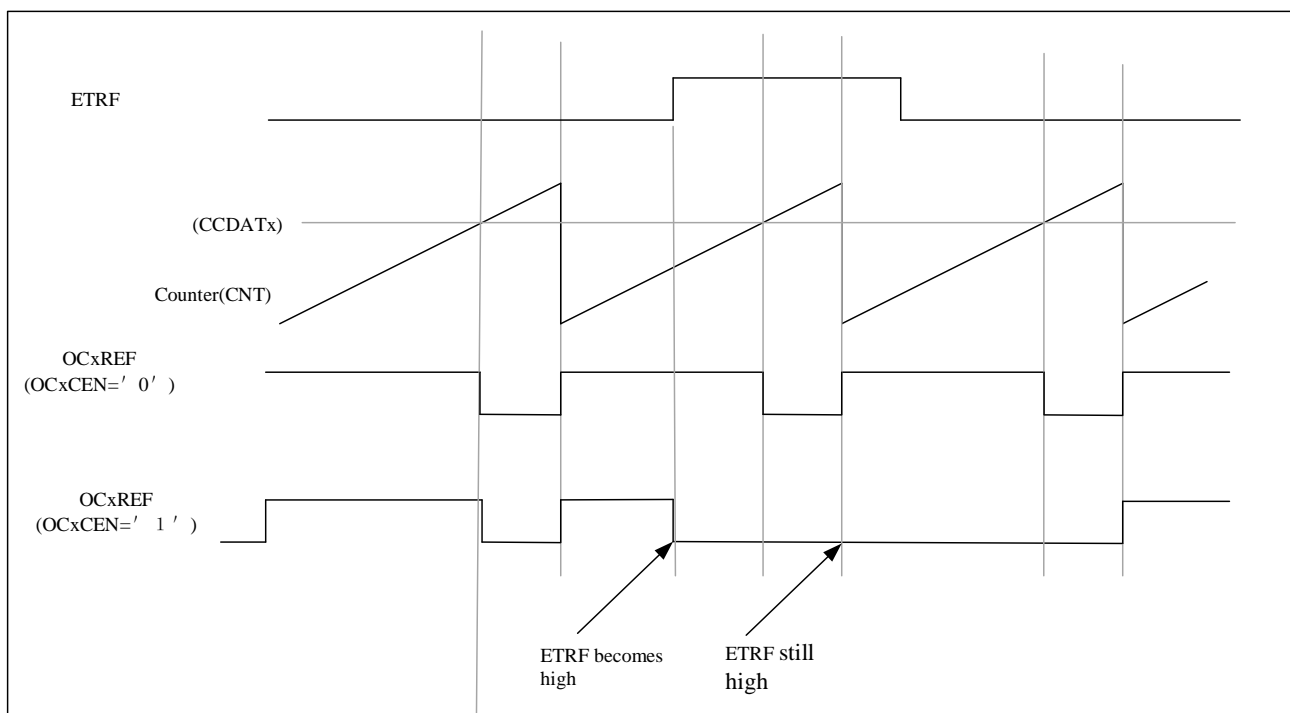
如果用户设置 $TIMx_CCMODx.OCxCEN=1$ ，ETRF 输入的高电平可用于驱动 OCxREF 信号为低电平，OCxREF 信号将保持低电平，直到下一次 UEV 发生。只有输出比较和 PWM 模式可以使用该功能。在强制模式下不能使用。

例如：为了控制电流，用户可以将 ETR 信号连接到比较器的输出端，ETR 的操作如下：

- 设置 $TIMx_SMCTRL.EXTPS=00$ 禁用外部触发预分频器。
- 设置 $TIMx_SMCTRL.EXCEN=0$ 禁用外部时钟模式 2。
- 设置 $TIMx_SMCTRL.EXTP$ 和 $TIMx_SMCTRL.EXTF$ ，根据需要配置外触发极性和外触发滤波器。

例：当 ETRF 输入变高时，OCxREF 信号对于不同的 OCxCEN 值的行为。在这种情况下，定时器设置为 PWM 模式。

图 9-28 清除 TIMx 的 OCxREF



9.3.13 互补输出和死区插入

高级控制定时器可以输出两个互补信号，并管理输出的关闭和打开。这称为死区时间。用户应根据连接到输出的设备及其特性调整死区时间。

用户可以通过设置 $TIMx_CCEN.CCxP$ 和 $TIMx_CCEN.CCxNP$ 来选择输出的极性。并且此选择对于每个输出都是独立的。

用户可以通过设置几个控制位的组合来控制互补信号 OCx 和 OCxN，它们分别是 $TIMx_CCEN.CCxEN$ 、 $TIMx_CCEN.CCxNEN$ 、 $TIMx_BKDT.MOEN$ 、 $TIMx_CTRL2.OIx$ 、 $TIMx_CTRL2.OIxN$ 、 $TIMx_BKDT.OSSI$ 和 $TIMx_BKDT.OSSR$ 。当切换到空闲状态时，死区时间将被激活。

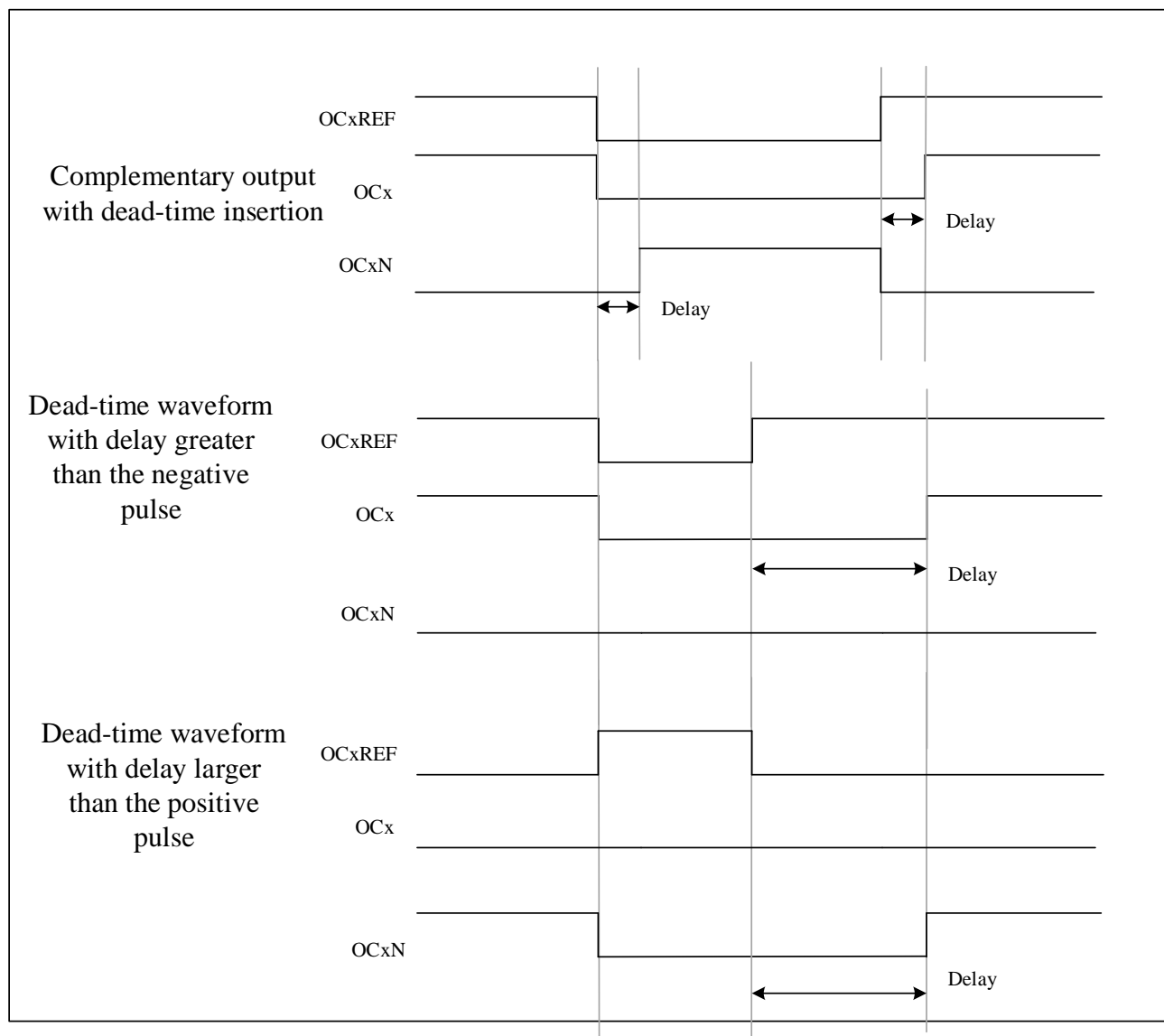
如果用户同时设置 $TIMx_CCEN.CCxEN$ 和 $TIMx_CCEN.CCxNEN$ ，则会插入死区时间。如果有刹车，还要设置 $TIMx_BKDT.MOEN$ 。每个通道都有 10 位死区时间发生器。

参考波形 $OCxREF$ 可以生成 2 个输出 OCx 和 $OCxN$ 。如果 OCx 和 $OCxN$ 为高电平有效，则 OCx 输出信号与参考信号相同，而 $OCxN$ 输出信号与参考信号相反。但是， OCx 输出信号将相对于参考上升沿延迟，而 $OCxN$ 输出信号将相对于参考下降沿延迟。如果延迟大于有效 OCx 或 $OCxN$ 输出的宽度，则不会产生相应的脉冲。

死区时间发生器的输出信号与参考信号 $OCxREF$ 之间的关系如下。

假设 $TIMx_CCEN.CCxP=0$ ， $TIMx_CCEN.CCxNP=0$ ， $TIMx_BKDT.MOEN=1$ ， $TIMx_CCEN.CCxEN=1$ ， $TIMx_CCEN.CCxNEN=1$ 。

图 9-29 带死区插入的互补输出



用户可以设置 $TIMx_BKDT.DTGN$ 来编程每个通道的死区时间延迟。

9.3.13.1 重定向 $OCxREF$ 到 OCx 或 $OCxN$

在输出模式下，用户可以设置 $TIMx_CCEN.CCxEN$ 和 $TIMx_CCEN.CCxNEN$ 以将 $OCxREF$ 重定向到

OCx 输出或 OCxN 输出。

这里有两种使用这个功能的方法。当互补保持在其无效电平时，用户可以使用此功能发送特定波形，例如 PWM 或静态有效电平。用户还可以使用此功能将两个输出设置为无效电平，或将两个输出都设置为有效，两者互补且带死区。

如果用户设置 TIMx_CCEN.CCxEN=0 和 TIMx_CCEN.CCxNEN=1，两者不互补，当 OCxREF 为高电平时 OCxN 将变为有效。另一方面，如果用户设置 TIMx_CCEN.CCxEN=1 和 TIMx_CCEN.CCxNEN=1，当 OCxREF 为高电平时，OCx 将变为有效。相反，当 OCxREF 为低电平时，OCxN 将变为有效。

9.3.14 刹车功能

使用刹车功能时，设置相应的控制位时会修改输出使能信号和无效电平。但是，无论何时，OCx 和 OCxN 的输出都不能同时处于有效电平，即需要满足 $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$ 。

当启用多个刹车信号时，每个刹车信号构成一个 OR 逻辑。这里有一些信号可能是刹车的来源。

- 刹车输入引脚
- 时钟失效事件，由时 RCC 中的时钟安全系统（CSS）生成。
- PVD 事件。
- 内核 Hardfault 事件。
- 比较器的输出信号（在比较器模块中配置，高电平刹车）。
- 软件设置 TIMx_EVTGEN.BGN。

复位后刹车电路将被禁用。MOEN 位将为低电平。用户可以设置 TIMx_BKDT.BKEN 来启用刹车功能。通过设置 TIMx_BKDT.BKP 可以选择刹车输入信号的极性。用户可以同时修改 TIMx_BKDT.BKEN 和 TIMx_BKDT.BKP。用户设置 TIMx_BKDT.BKEN 和 TIMx_BKDT.BKP 后，生效前有 1 个 APB 时钟周期延迟。因此，用户需要等待 1 个 APB 时钟周期才能读回写入位的值。

MOEN 的下降沿可以是异步的，所以在实际信号和同步控制位之间设置了一个再同步电路。该电路将导致异步和同步信号之间的延迟。当用户设置 TIMx_BKDT.MOEN 为低电平时，用户需要在读取该值之前插入一个延迟。因为写入了异步信号，但用户读取了同步信号。

刹车发生后的行为如下：

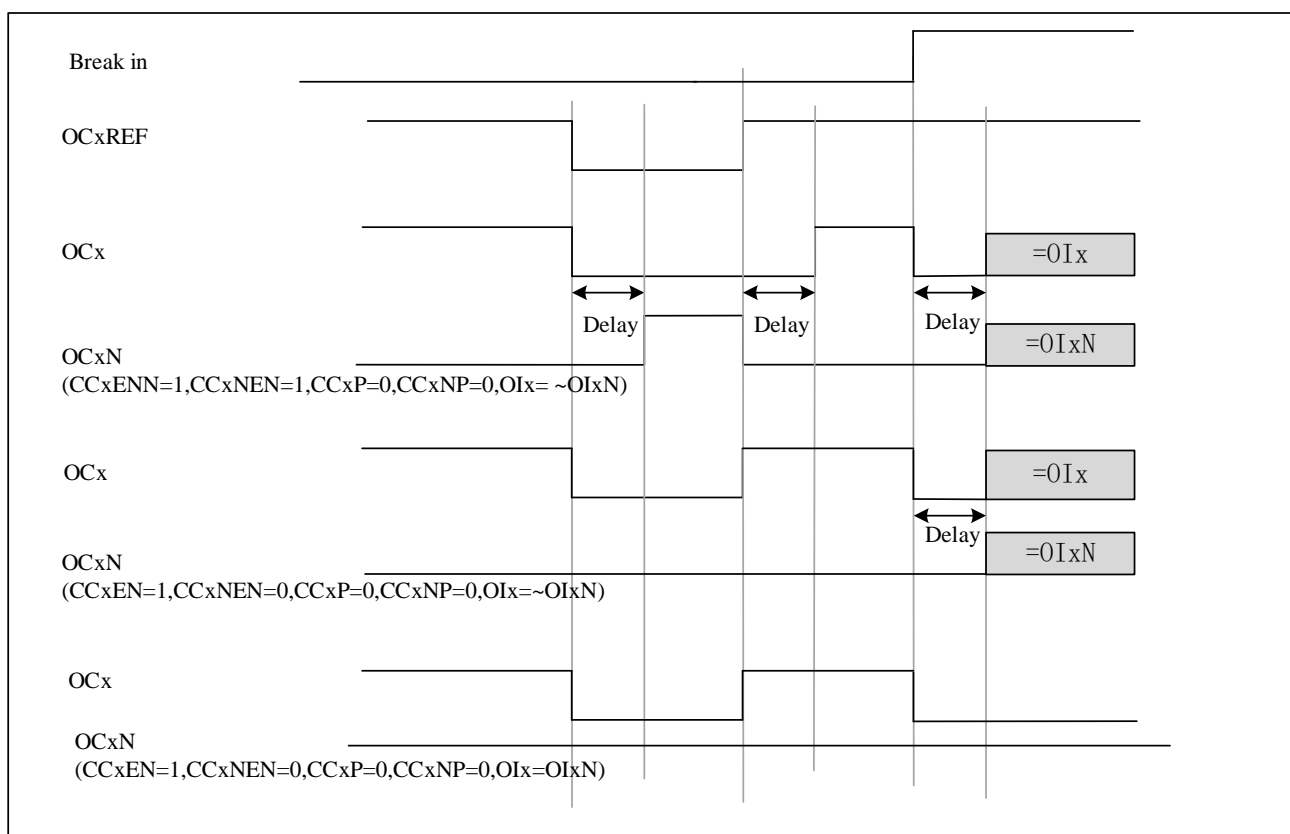
- TIMx_BKDT.MOEN 将被异步清除，然后输出将进入无效状态、空闲状态或复位状态。通过设置 TIMx_BKDT.OSSI 选择输出状态。即使 MCU 振荡器关闭，这也会生效。
- 一旦 TIMx_BKDT.MOEN=0，每个输出通道的输出将使用 TIMx_CTRL2.OIx 中编程的电平驱动。如果 TIMx_BKDT.OSSI=0，定时器将释放使能输出（由 GPIO 控制器接管），否则将保持高电平。
- 如果用户选择使用互补输出，TIM 的行为如下
 - 取决于极性，输出将首先设置为复位状态。它是一个异步选项，因此即使没有为计时器提供时钟，它仍然可以工作。
 - 如果仍然提供定时器时钟，死区发生器将重新激活，当 $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$ ，即 OCx 和 OCxN 仍然不能同时被驱动到有效电平，在死区时间后根据 TIMx_CTRL2.OIx 和 TIMx_CTRL2.OIxN 的值驱动输出。请注意，由于 MOEN 上的重新同步（大概 2 个 ck_tim 周期），死区时间将比平时长。

- 如果 $TIMx_BKDT.OSSI=0$ ，定时器将释放输出控制。否则，如果使能输出为高电平，它将保持为高电平。如果为低电平，则在 $TIMx_CCEN.CCxEN$ 或 $TIMx_CCEN.CCxNEN$ 为高电平时变为高电平。
- 如果 $TIMx_DINTEN.BIEN=1$ ，当 $TIMx_STS.BITF=1$ 时，会产生中断。
- 如果用户设置了 $TIMx_BKDT.AOEN$ ， $TIMx_BKDT.MOEN$ 将在下一次 UEV 发生时自动设置。用户可以使用它来调节。如果用户未设置 $TIMx_BKDT.AOEN$ ，则 $TIMx_BKDT.MOEN$ 将保持低电平，直到再次设置为 1。在这种情况下，用户可以使用它来保证安全。用户可以将刹车输入连接到热传感器、电源驱动器警报或其他安全组件。
- 刹车输入有效时， $TIMx_BKDT.MOEN$ 不能自动置位或软件同时置位， $TIMx_STS.BITF$ 也不能清零。因为刹车输入在电平上处于有效状态。

为保证应用安全，刹车电路具有写保护功能，并有刹车输入输出管理。它允许用户冻结一些参数，例如死区持续时间、 $OCx/OCxN$ 极性和禁用时的状态、 $OCxMD$ 配置、刹车启用和极性。用户可以通过设置 $TIMx_BKDT.LCKCFG$ 选择使用 3 种保护级别之一。但是， $TIMx_BKDT.LCKCFG$ 只能在 MCU 复位后写入一次。

响应刹车的输出行为示例如下

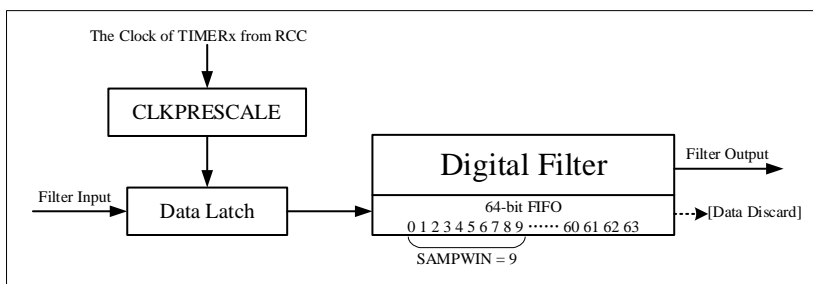
图 9-30 响应刹车的输出行为



9.3.14.1 刹车滤波

寄存器 $TIM1_BKFR$ 描述如下：

图 9-31 滑动滤波



- 数字滤波器通过 RCC 的 TIM1 时钟采样刹车信号，在 64 位 FIFO 中累积采样。仅在 TIM1_BKFR.WSIZE [5:0] 中定义的窗口大小内采样数据，最大大小为 64。
- 过滤器输出采样窗口内的多数值，该值由 TIM1_BKFR.THRESH [5:0] 中的阈值定义，最大阈值为 63。此值应等于或大于窗口大小的一半。如果采样窗口内的逻辑 1 和逻辑 0 计数均不大于阈值，则数字滤波器保持先前的输出值。
- TIM1_BKFR.PSC 寄存器决定相应数字滤波器的采样率。过滤器 FIFO 在每个采样时钟从输入中捕获一个采样值。
- 如果数字滤波器关闭，滤波器输入将像电线一样绕过输出。

9.3.15 调试模式

当微控制器处于调试模式（Cortex-M0 内核停止）时，根据 DBG_CTRL.TIMx_STOP 配置，TIMx 计数器可以继续正常工作或停止。详见 3.4.9。

9.3.16 TIMx 定时器和外部触发的同步

TIMx 定时器可以通过从模式（复位、触发和门控）中的触发器进行同步。

9.3.16.1 从模式：复位模式

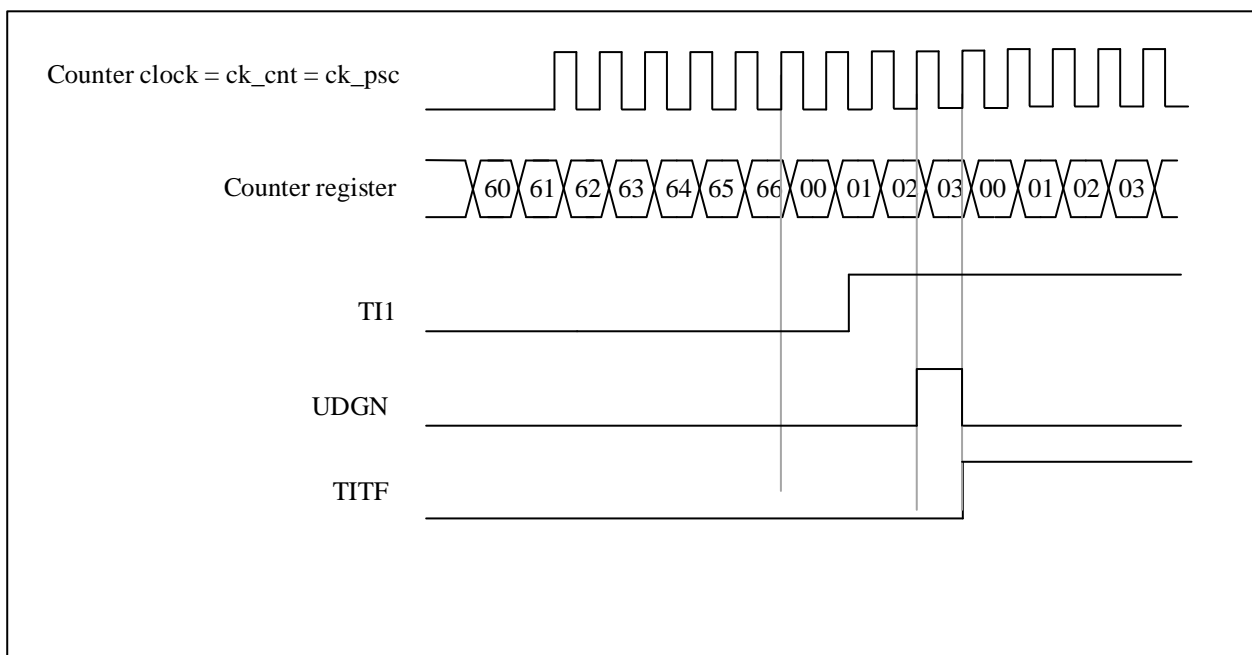
在复位模式下，触发事件可以复位计数器和预分频器。更新预加载寄存器 TIMx_AR、TIMx_CCDATx，并产生更新事件 UEV（TIMx_CTRL1.UPRS=0）。

以下是复位模式的示例：

1. 通道 1 配置为输入检测 TI1 的上升沿（TIMx_CCMOD1.CC1SEL=01，TIMx_CCEN.CC1P=0）；
2. 从模式选择为复位模式（TIMx_SMCTRL.SMSEL=100），触发输入选择为 TI1（TIMx_SMCTRL.TSEL=101）；
3. 启动计数器（TIMx_CTRL1.CNTEN = 1）

启动定时器后，当 TI1 检测到上升沿时，计数器复位并重新开始计数，并设置触发标志（TIMx_STS.TITF=1）；TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重同步电路引起的。

图 9-32 复位模式下的控制电路



9.3.16.2 从模式：触发模式

在触发模式下，输入端口的触发事件（上升沿/下降沿）可以触发计数器开始计数。

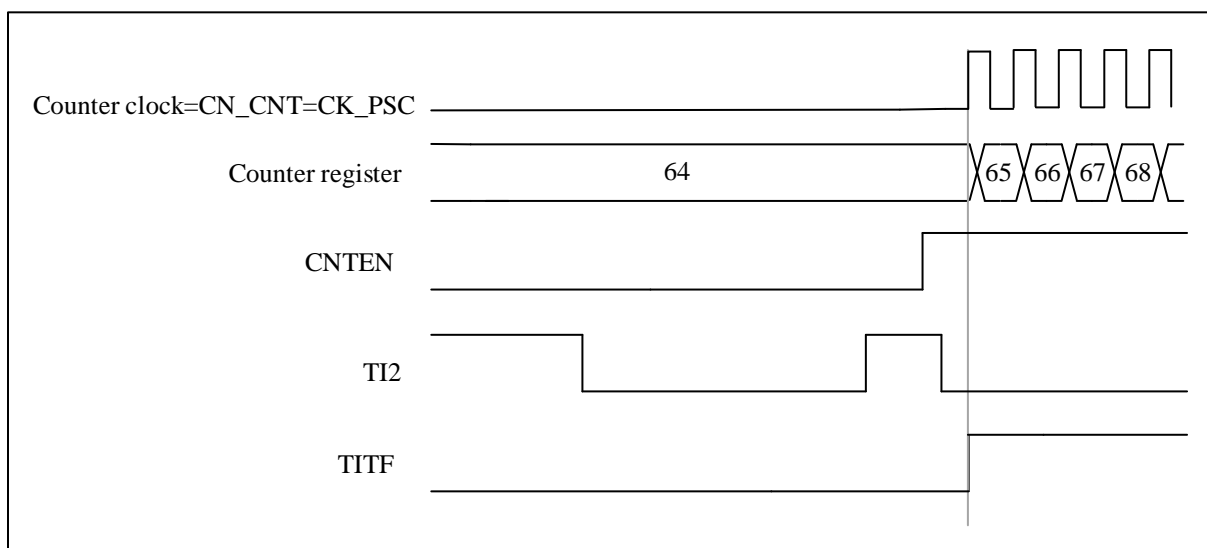
以下是触发模式的示例：

1. 通道 2 配置为输入，检测 TI2 的上升沿（TIMx_CCMOD1.CC2SEL=01，TIMx_CCEN.CC2P=0）；
2. 选择从模式为触发模式（TIMx_SMCTRL.SMSEL=110），触发输入选择 TI2（TIMx_SMCTRL.TSEL=110）；

当 TI2 检测到上升沿时，计数器开始计数，触发标志置位（TIMx_STS.TITF=1）；

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 9-33 触发器模式下的控制电路



9.3.16.3 从模式：门控模式

在门控模式下，输入端口的电平极性可以控制计数器是否计数。

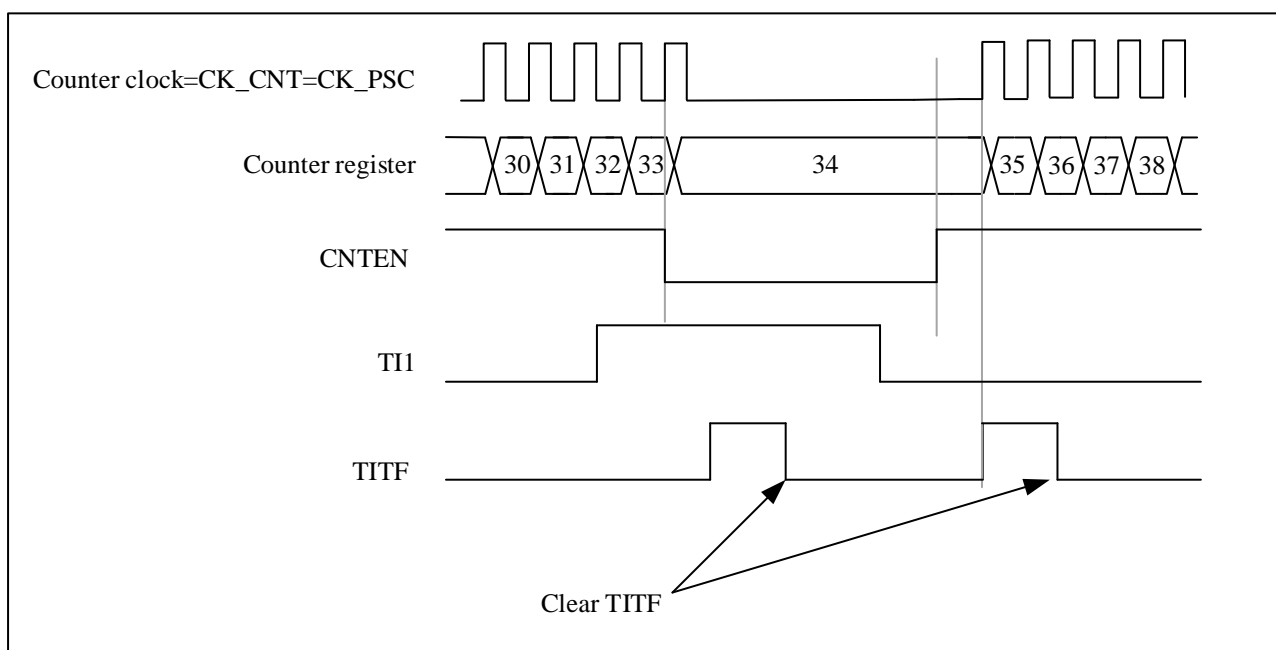
以下是门控模式的示例：

1. 通道 1 配置为 TI1 上的输入检测低电平有效 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=1);
2. 选择从模式为门控模式 (TIMx_SMCTRL.SMSEL=101)，选择 TI1 作为 TRGI (TIMx_SMCTRL.TSEL=101)；
3. 启动计数器 (TIMx_CTRL1.CNTEN = 1)；

当 TI1 检测到电平由低变高时，计数器停止计数，当 TI1 检测到电平由高变低时，计数器开始计数，开始或停止计数时触发标志置位 (TIMx_STS.TITF=1)。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 9-34 门控模式下的控制电路



9.3.16.4 从模式：触发模式 + 外部时钟模式 2

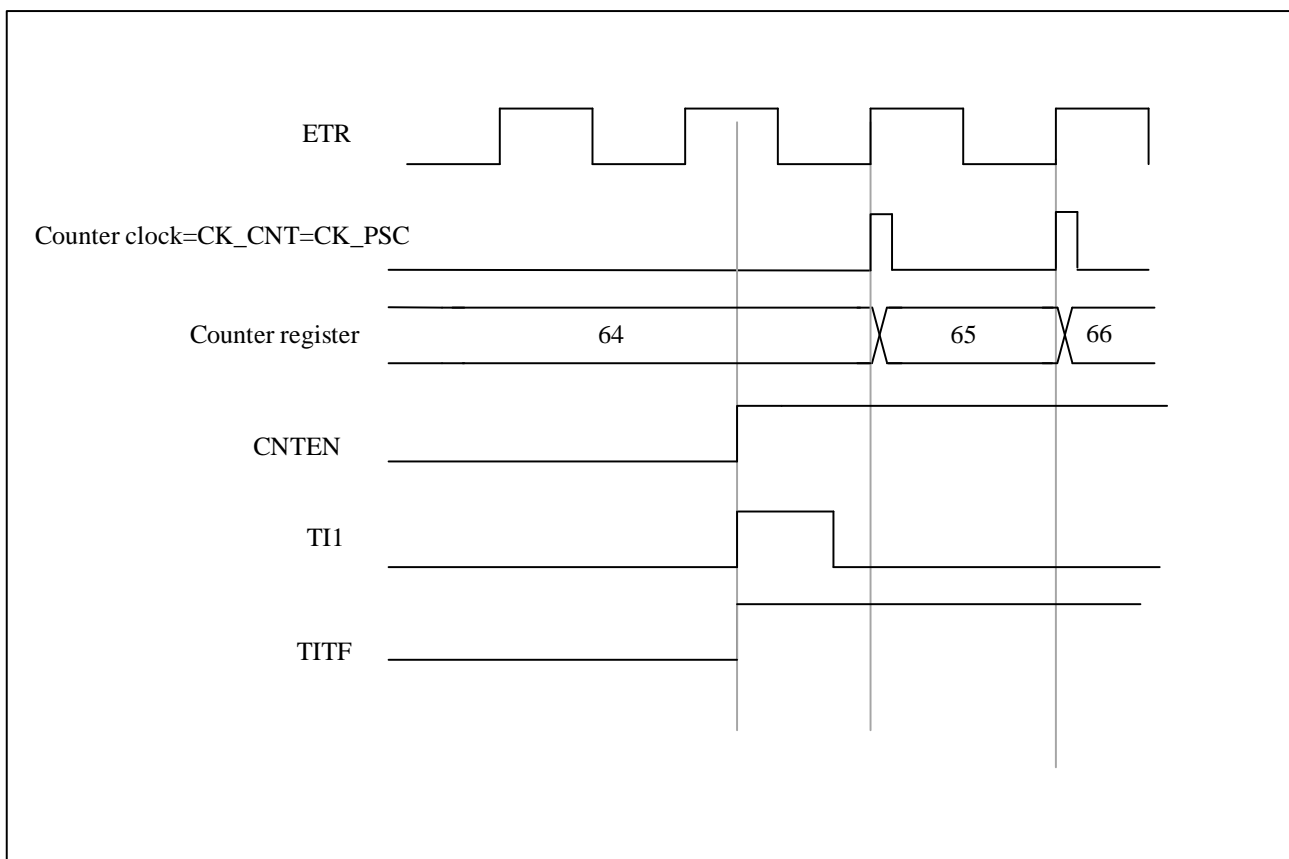
在复位模式、触发模式和门控模式下，计数器时钟可选择为外部时钟模式 2，ETR 信号作为外部时钟源输入。这时候触发选择需要选择非 ETRF (TIMx_SMCTRL.TSEL=111)。

这是一个例子：

1. 通道 1 配置为输入检测 TI1 的上升沿 (TIMx_CCMOD1.CC1SEL=01, TIMx_CCEN.CC1P=0)；
2. 使能外部时钟模式 2 (TIMx_SMCTRL.EXCEN=1)，外部触发极性选择上升沿 (TIMx_SMCTRL.EXTTP=0)，触发模式作为从模式 (TIMx_SMCTRL.SMSEL=110)，TRGI 选择 TI1 (TIMx_SMCTRL.TSEL=101)；

当 TI1 检测到上升沿时，计数器在 ETR 的上升沿开始计数，并设置触发标志 (TIMx_STS.TITF=1)；

图 9-35 外部时钟模式 2+触发模式下的控制电路



9.3.17 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。详见 10.3.14 章节。

9.3.18 产生六步 PWM 输出

为了同时修改所有通道的配置，可以提前设置下一步的配置（预加载位为 OCxMD、CCxEN 和 CCxNEN）。当发生 COM 换相事件时，OCxMD、CCxEN 和 CCxNEN 预加载位被传送到影子寄存器位。

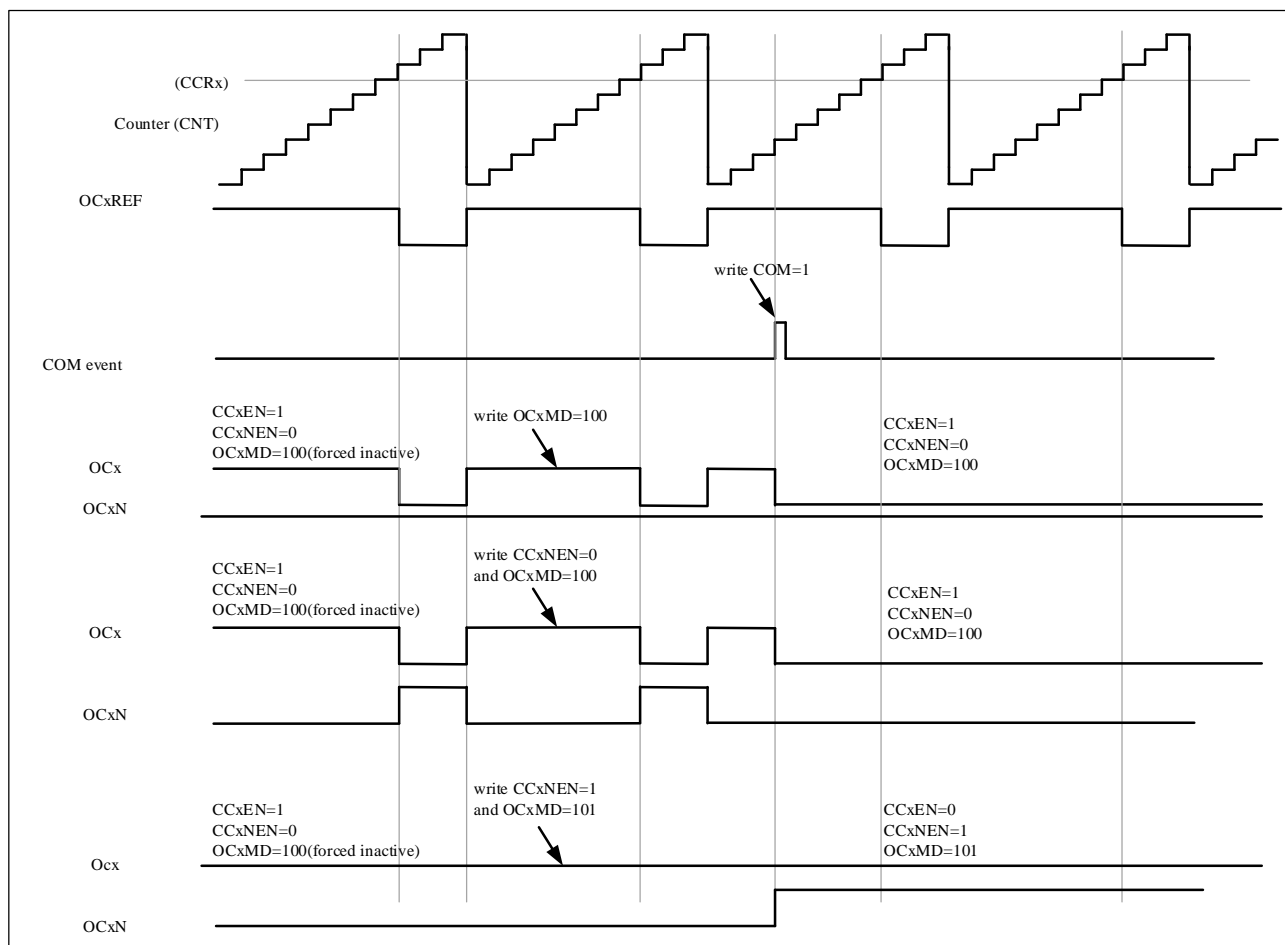
COM 换相事件生成方法：

1. 软件设置 TIMx_EVTGEN.CCUDGN；
2. 在 TRGI 的上升沿由硬件产生；

当 COM 换相事件发生时，TIMx_STS.COMITF 标志将被设置，启用中断 (TIMx_DINTEN.COMIEN) 将产生中断，启用 DMA 请求 (TIMx_DINTEN.COMDEN) 将产生 DMA 请求。

下图显示了三种不同配置下发生 COM 换向事件时 OCx 和 OCxN 的输出时序图：

图 9-36 产生六步 PWM，使用 COM 的例子（OSSR=1）



9.3.19 编码器接口模式

编码器使用两个输入 TI1 和 TI2 作为接口，计数器对 TI1FP1 或 TI2FP2 上的每个边沿变化进行计数。计数方向由硬件 TIM_x_CTRL1.DIR 自动控制。编码器计数模式共有三种：

1. 计数器只在 TI1 的边沿计数，TIM_x_SMCTRL.SMSEL = '001'；
2. 计数器只在 TI2 的边沿计数，TIM_x_SMCTRL.SMSEL = '010'；
3. 计数器同时在 TI1 和 TI2 的边沿计数，TIM_x_SMCTRL.SMSEL = '011'；

编码器接口相当于使用带方向选择的外部时钟，计数器只在 0 和自动重载值(TIM_x_AR.AR[15:0])之间连续计数。因此，需要提前配置自动重载寄存器 TIM_x_AR。

注意：编码器模式和外部时钟模式 2 不兼容，不能同时选择。

计数方向与编码器信号的关系如下表：

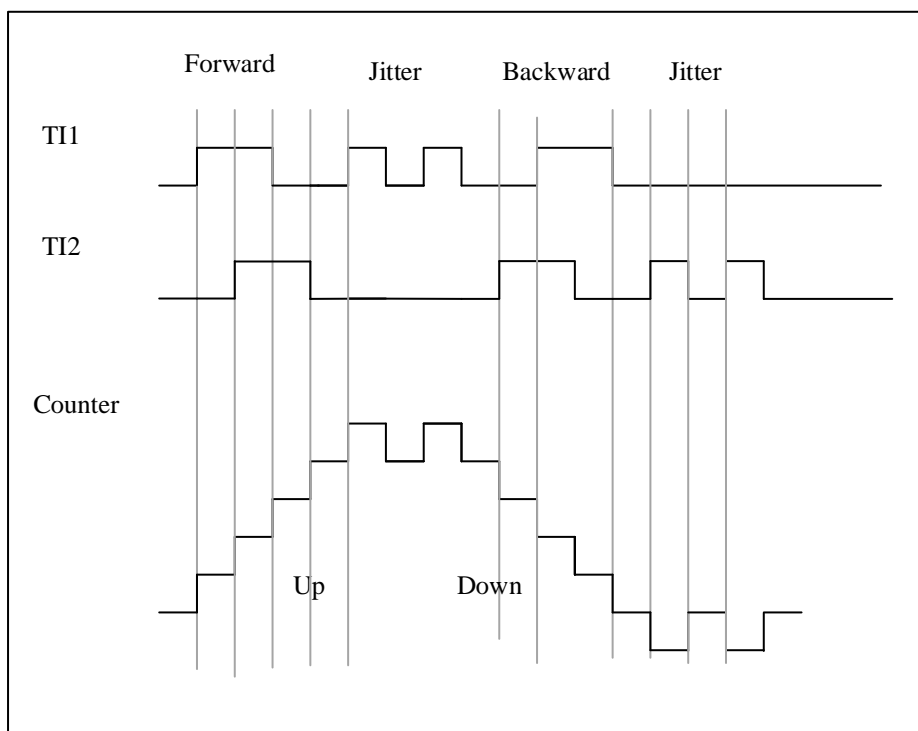
表 9-1 计数方向与编码器信号的关系

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

以下是选择了双边沿触发以抑制输入抖动的编码器示例：

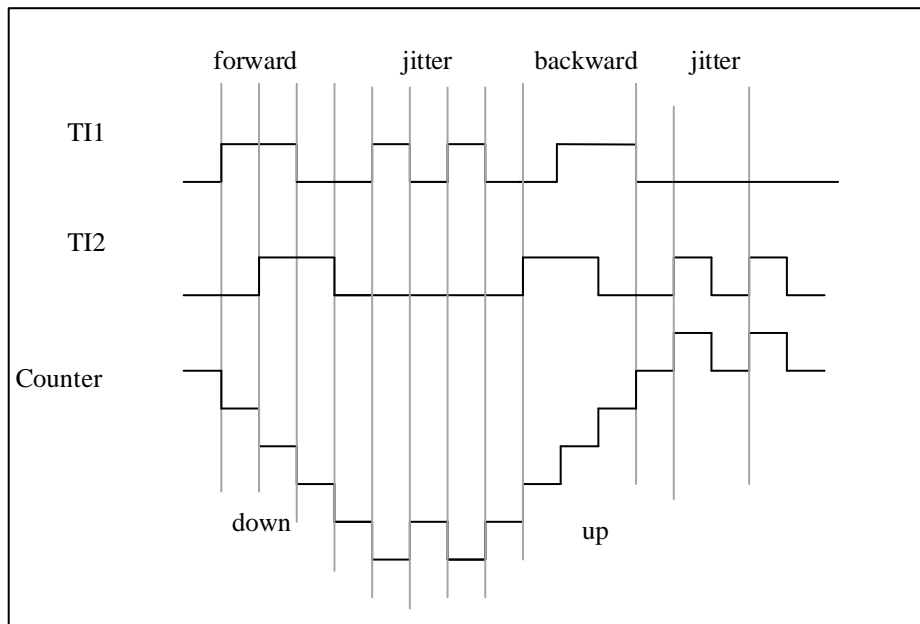
1. IC1FP1 映射到 TI1 (TIMx_CCMOD1.CC1SEL = '01')，IC1FP1 不反相 (TIMx_CCEN.CC1P = '0')；
2. IC1FP2 映射到 TI2 (TIMx_CCMOD2.CC2SEL = '01')，IC2FP2 不反相 (TIMx_CCEN.CC2P = '0')；
3. 输入在上升沿和下降沿均有效 (TIMx_SMCTRL.SMSEL = '011')；
4. 启用计数器 TIMx_CTRL1.CNTEN = '1'；

图 9-37 编码器模式下的计数器操作实例



下图为 IC1FP1 极性反转时的计数器行为示例（CC1P = '1'，其他配置同上）

图 9-38 IC1FP1 反相的编码器接口模式实例



9.3.20 与霍尔传感器的接口

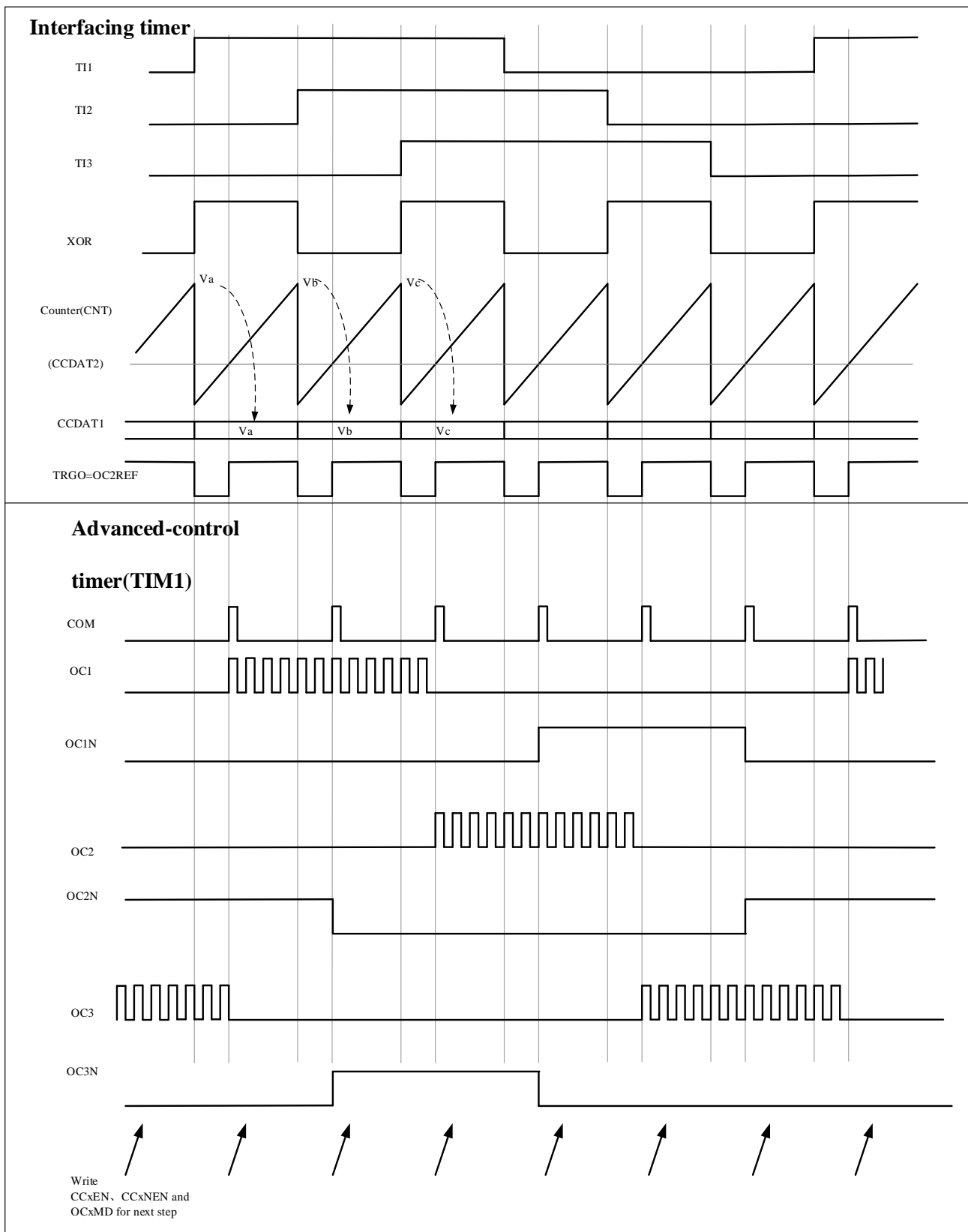
将霍尔传感器连接到定时器的三个输入引脚（CC1、CC2 和 CC3），然后选择异或功能将 TIMx_CH1、TIMx_CH2 和 TIMx_CH3 的输入通过异或门作为 TI1 的输出到通道 1 进行捕捉信号。

定时器需要配置为从模式下的复位模式（TIMx_SMCTRL.SMSEL='100'）；触发选择 TI1 的边沿触发 TI1F_ED (TIMx_SMCTRL.TSEL='100')，霍尔 3 输入的任何变化都会触发计数器重新计数，因此用作时间参考；捕获/比较通道 1 配置为捕获模式下的 TRC 信号（TIMx_CCMOD1.CC1SEL='11'），用于计算两个输入时间间隔，从而反映电机速度。

选择定时器通道 2 向高级定时器输出脉冲，触发高级定时器的 COM 事件，更新输出 PWM 的控制位。高级定时器的触发选择需要选择对应的内部触发信号（TIMx_SMCTRL.TSEL="ITRx"），捕获/比较预加载控制位需要配置为支持预加载（TIMx_CTRL2.CCPCTL=1）并支持上升沿 TRGI 边沿触发更新（TIMx_CTRL2.CCUSEL=1）。

此示例如下图所示。

图 9-39 霍尔传感器接口的实例



9.4 TIM1 寄存器描述

关于在寄存器描述里面所用到的缩写，详见 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

9.4.1 寄存器总览

表 9-2 TIM1 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000h	TIMx_CTRL1	Reserved								ASYMMETRIC	Reserved	CMODE[1:0]			Reserved			CISEL	COMPBKPEN	Reserved	CLSEL	IOBKPEN	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN	
	0									0								0	0		0	0	0	0	0	0			0	0			0	0	0
004h	TIMx_CTRL2	Reserved								TRIG9	TRIG8	TRIG7	TRIG4	TI1SEL	CCPCTL	CCDSEL	CCUSEL	MMSEL[3:0]				Reserved	O16	Reserved	O15	O14N	O14	O13N	O13	O12N	O12	O11N	O11		
	0									0	0	0	0	0	0	0	0						0		0	0	0	0	0	0	0	0	0	0	0
008h	TIMx_STS	Reserved												BITF	TI1F	COM1TF	UD1TF	Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved		CC6ITF	CC5ITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF		
	0													0	0	0	0					0	0	0	0			0	0	0	0	0	0	0	0
00Ch	TIMx_EVTGEN	Reserved																		BGN	TGN	CCUDGN	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN				
	0																			0	0	0	0					0	0	0	0	0	0	0	0
010h	TIMx_SMCTRL	Reserved																MSMD	EXTF[3:0]				EXTP	EXCEN	EXTPS[1:0]		Reserved	SMSEL[2:0]		Reserved	TSEL[2:0]				
	0																	0					0	0									0	0	0
014h	TIMx_DINTEN	Reserved								COM1EN	TDEN	COMDEN	UDEN	BIEN	TIEN	UIEN	Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN			
	0									0	0	0	0	0	0	0					0	0	0	0					0	0	0	0	0	0	0
018h	TIMx_CCMOD1	Reserved																OC2MD[2:0]		OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]		OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]					
	0																				0	0						0	0			0	0	0	0
	Reset Value	Reserved																IC2F[3:0]		IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]		IC1PSC[1:0]		CC1SEL[1:0]							
	0																													0	0	0	0	0	0
01Ch	TIMx_CCMOD2	Reserved																OC4MD[2:0]		OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]		OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]					
	0																				0	0						0	0			0	0	0	0
	Reset Value	Reserved																IC4F[3:0]		IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]		IC3PSC[1:0]		CC3SEL[1:0]							
	0																													0	0	0	0	0	0
020h	TIMx_CCMOD3	Reserved								OC9PEN	Reserved			OC8PEN	Reserved			OC7PEN	OC6MD[2:0]		OC6CEN	OC6FEN	OC6PEN	Reserved		OC5MD[2:0]		OC5CEN	OC5FEN	OC5PEN	Reserved				
	0									0				0				0			0	0	0					0	0	0			0	0	0
024h	TIMx_CCEN	Reserved								CC6P	CC6EN	Reserved			CC5P	CC5EN	Reserved			CC4P	CC4EN	CC4NP	CC4NEN	CC3P	CC3EN	CC3NP	CC3NEN	CC2P	CC2EN	CC2NP	CC2NEN	CC1P	CC1EN	CC1NP	CC1NEN
	0									0	0				0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	TIMx_CCDAT1	CCDDAT1[15:0]																CCDAT1[15:0]																	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

02Ch	TIMx_CCDA2	CCDDAT2[15:0]															CCDAT2[15:0]														
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
030h	TIMx_CCDA3	CCDDAT3[15:0]															CCDAT3[15:0]														
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
034h	TIMx_CCDA4	CCDDAT4[15:0]															CCDAT4[15:0]														
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
038h	TIMx_CCDA5	Reserved															CCDAT5[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	TIMx_CCDA6	Reserved															CCDAT6[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	TIMx_PSC	Reserved															PSC[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044h	TIMx_AR	Reserved															AR[15:0]														
	Reset Value																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
048h	TIMx_CNT	Reserved															CNT[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04Ch	TIMx_REPCNT	Reserved															REPCNT[7:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
050h	TIMx_BKDT	Reserved															LCKCFG[1:0]	OSSR	OSSI	BKEN	BKP	AOEN	MOEN	DTGN[7:0]							
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
054h	TIMx_CCDA7	Reserved															CCDAT7[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
058h	TIMx_CCDA8	Reserved															CCDAT8[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05Ch	TIMx_CCDA9	Reserved															CCDAT9[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
060h	TIMx_BKFR	Reserved	THRESH[5:0]					Reserved	WSIZE[5:0]					FILTEN	PSC[15:0]																
	Reset Value		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
094h	TIMx_DCTRL	Reserved															DBADDR[4:0]				Reserved	DBLEN[4:0]									
	Reset Value																0	0	0	0		0	0	0	0	0	0	0	0	0	0
098Ch	TIMx_DADDR	Reserved															BURST[15:0]														
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.2 控制寄存器 1 (TIMx_CTRL1)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ASYMMETRIC	Reserved	CMODE[1:0]		Reserved			C1SEL
rw										rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPBKPEN	Reserved	CLRSEL	IOMBKPEN	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN
rw		rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw

位域	名称	描述
31:24	Reserved	保留，必须保持复位值
23	ASYMMETRIC	中央对齐非对称模式使能 0: 禁用 1: 使能（当TIMx_CTRL1.CAMSEL[1:0]非零时有效，每个通道向上计数时与CCDATx比较，向下计数时与CCDDATx比较） 注意：此功能仅适用于 TIM1
22	Reserved	保留，必须保持复位值
21:20	CMODE	中心对齐非对称模式下，通道4/7/8/9触发的方式，只有当CTRL2的MMSEL为'1xxx'时候才会通过trgo信号输出生效 00: 向上计数达到CCDAT4/7/8/9的值，触发有效 01: 向下计数，通道4达到CCDDAT4，通道7/8/9达到CCDAT7/8/9的值，触发有效 1X: 通道4向上计数达到CCDAT4，向下计数达到CCDDAT4；通道7/8/9向上和向下计数达到CCDAT7/8/9的值，触发均有效。 中心对齐对称模式下，通道4/7/8/9触发的方式，只有当CTRL2的MMSEL为'1xxx'时候才会通过trgo信号输出生效 00: 向上计数达到CCDAT4/7/8/9的值，触发有效 01: 向下计数达到CCDAT4/7/8/9的值，触发有效 1X: 向上和向下计数达到CCDAT4/7/8/9的值，触发均有效
19:17	Reserved	保留，必须保持复位值
16	C1SEL	通道1选择（Channel 1 selection） 0: 选择外部CH1（来自IOM）信号 1: 选择内部CH1（来自COMP）信号
15	COMPBKPEN	COMP作为BRK使能（COMP as brk Enable） 0: 使能 1: 禁止
14	Reserved	保留，必须保持复位值
13	CLRSEL	OCxRef选择（OCxRef selection） 0: 选择外部Ocxclr（ETR）信号 1: 选择内部Ocxclr（来自COMP）信号

位域	名称	描述
12	IOMBKPEN	IOM作为BRK使能 (IOM as brk Enable) 0: 使能。选择外部刹车信号 (来自IOM) 1: 禁止。选择内部刹车信号 (来自COMP)
11	PBKPEN	PVD作为BRK启用 (PVD as brk Enable) 0: 禁止 1: 使能
10	LBKPEN	锁存作为BRK使能 (LockUp as brk Enable) (Core Hardfault) 0: 禁止 1: 使能
9	ARPEN	自动重载预装载允许位 (Auto-reload preload enable) 0: TIMx_AR 寄存器的影子寄存器禁用 1: TIMx_AR 寄存器的影子寄存器使能
8	ONEPM	单脉冲模式 (One pulse mode) 0: 禁用单脉冲模式, 发生更新事件时不影响计数器计数。 1: 使能单脉冲模式, 下次更新事件发生时计数器停止计数
7:6	CLKD[1:0]	时钟分频因子 (Clock division) CLKD[1:0] 表示 CK_INT (定时器时钟) 和 DTS (用于死区时间发生器和数字滤波器 (ETR、TIx) 的时钟) 之间的分频比。 00: $t_{DTS} = t_{CK_INT}$ 01: $t_{DTS} = 2 \times t_{CK_INT}$ 10: $t_{DTS} = 4 \times t_{CK_INT}$ 11: 保留, 不要使用这个配置
5	UPDIS	更新禁用 (Update disable) 该位用于启用/禁用软件生成的更新事件 (UEV) 事件。 0: 启用。 如果满足以下条件之一, 将生成 UEV: – 计数器上溢/下溢 – TIMx_EVTGEN.UDGN 位被设置 – 从模式控制器的更新生成 影子寄存器将使用预加载值进行更新。 1: UEV 禁用。 不生成更新事件, 影子寄存器 (AR、PSC 和 CCDATx) 保持它们的值。 如果 TIMx_EVTGEN.UDGN 位置位或从模式控制器发出硬件复位, 则重新初始化计数器和预分频器。
4	UPRS	更新请求源 (Update request source) 该位用于通过软件选择 UEV 事件源。 0: 如果更新中断或 DMA 请求使能, 以下任何事件都会产生更新中断或 DMA 请求: – 计数器上溢/下溢 – TIMx_EVTGEN.UDGN 位被设置 – 从模式控制器的更新生成 1: 如果更新中断或 DMA 请求使能, 只有计数器上溢/下溢会产生更新中断或 DMA 请求。

位域	名称	描述
3:2	CAMSEL[1:0]	选择中央对齐模式（Center-aligned mode selection） 00：边缘对齐模式。TIMx_CTRL1.DIR 指定向上计数或向下计数。 01：中央对齐模式1。计数器在中央对齐模式下计数，向下计数时输出比较中断标志位设置为 1。 10：中央对齐模式2。计数器在中央对齐模式下计数，向上计数时输出比较中断标志位设置为1。 11：中央对齐模式3。计数器在中央对齐模式下计数，向上计数或向下计数时输出比较中断标志位设置为 1。 <i>注意：当计数器仍然启用时（TIMx_CTRL1.CNTEN = 1），不允许从边缘对齐模式切换到中央对齐模式。</i>
1	DIR	方向（Direction） 0：计数器向上计数； 1：计数器向下计数。 <i>注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。</i>
0	CNTEN	使能计数器（Counter enable） 0：禁止计数器； 1：使能计数器。 <i>注：在软件设置了CNTEN位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。</i>

9.4.3 控制寄存器 2（TIMx_CTRL2）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TRIG9	TRIG8	TRIG7	TRIG4	TI1SEL	CCPCTL	CCDSEL	CCUSEL
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSEL[3:0]				Reserved	OI6	Reserved	OI5	OI4N	OI4	OI3N	OI3	OI2N	OI2	OI1N	OI1
rw					rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:24	Reserved	保留，必须保持复位值
23	TRIG9	OC9REF 触发 ADC 使能 0：触发禁用 1：触发使能
22	TRIG8	OC8REF 触发 ADC 使能 0：触发禁用 1：触发使能
21	TRIG7	OC7REF 触发 ADC 使能 0：触发禁用

位域	名称	描述
		1: 触发使能
20	TRIG4	OC4REF 触发 ADC 使能 0: 触发禁用 1: 触发使能
19	TI1SEL	TI1选择 (TI1 selection) 0: TIMx_CH1 引脚连到TI1输入; 1: TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
18	CCPCTL	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxEN, CCxNEN和OCxMD位不是预装载的; 1: CCxEN, CCxNEN和OCxMD位是预装载的; 设置该位后, 它们只在设置了CCUDGN位后被更新。 <i>注: 该位只对具有互补输出的通道起作用。</i>
17	CCDSEL	捕获/比较的DMA选择 (Capture/compare DMA selection) 0: 当发生CCx事件时, 送出CCx的DMA请求; 1: 当发生更新事件时, 送出CCx的DMA请求。
16	CCUSEL	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的 (CCPCTL =1), 只能通过设置CCUDGN位更新它们; 1: 如果捕获/比较控制位是预装载的 (CCPCTL =1), 可以通过设置CCUDGN位或TRGI上的一个上升沿更新它们。 <i>注: 该位只对具有互补输出的通道起作用。</i>
15:12	MMSEL[3:0]	主模式选择 这 4 位用于选择在主模式下发送到从定时器的同步信息 (TRGO)。可能的组合如下: 000: 复位 - 当 TIMx_EVTGEN.UDGN 置位或从模式控制器产生复位时, 将出现 TRGO 脉冲。在后一种情况下, TRGO 上的信号与实际复位相比有所延迟。 0001: 使能 - TIMx_CTRL1.CNTEN 位用作触发输出 (TRGO)。有时需要同时启动多个定时器或者在一段时间内开启从定时器。 当 TIMx_CTRL1.CNTEN 位置位或门控模式下的触发输入为高电平时, 计数器使能信号置位。 当计数器使能信号由触发输入控制时, TRGO 上有一个延迟, 除非选择了主/从模式 (参见 TIMx_SMCTRL.MSMD 位的说明)。 0010: 更新 - 选择更新事件作为触发输出 (TRGO)。例如, 主定时器时钟可用作从定时器预分频器。 0011: 比较脉冲 - 当 TIMx_STS.CC1ITF 被设置时 (即使它已经是高电平), 即捕获或比较成功时, 触发输出发送一个正脉冲 (TRGO)。 0100: 比较 - OC1REF 信号用作触发输出 (TRGO)。 0101: 比较 - OC2REF 信号用作触发输出 (TRGO)。 0110: 比较 - OC3REF 信号用作触发输出 (TRGO)。 0111: 比较 - OC4REF 信号用作触发输出 (TRGO)。 1xxx: 比较-如果计数器为中央对齐模式: OC4REF/OC7REF/OC8REF/OC9REF的相应边沿信号作为触发输出 (TRGO), 向上/向下计数可配置, 具体参考TIMx_CTRL1.CMODE 中。 如果计数器为边沿对齐模式: OC4REF 信号用作触发输出 (TRGO)。

位域	名称	描述
11	Reserved	保留，必须保持复位值
10	OI6	输出空闲状态6（OC6输出）。参见OI1位。
9	Reserved	保留，必须保持复位值
8	OI5	输出空闲状态5（OC5输出）。参见OI1位。
7	OI4N	输出空闲状态4（OC4N输出）。参见OI1N位。
6	OI4	输出空闲状态4（OC4输出）。参见OI1位。
5	OI3N	输出空闲状态3（OC3N输出）。参见OI1N位。
4	OI3	输出空闲状态3（OC3输出）。参见OI1位。
3	OI2N	输出空闲状态2（OC2N输出）。参见OI1N位。
2	OI2	输出空闲状态2（OC2输出）。参见OI1位。
1	OI1N	输出空闲状态1（OC1N输出）（Output Idle state 1N） 0：当MOEN=0时，死区后OC1N=0； 1：当MOEN=0时，死区后OC1N=1。 <i>注：已经设置了LCKCFG（TIMx_BKR寄存器）级别1、2或3后，该位不能被修改。</i>
0	OI1	输出空闲状态1（OC1输出）（Output Idle state 1） 0：当MOEN=0时，如果实现了OC1N，则死区后OC1=0； 1：当MOEN=0时，如果实现了OC1N，则死区后OC1=1。 <i>注：已经设置了LCKCFG（TIMx_BKR寄存器）级别1、2或3后，该位不能被修改。</i>

9.4.4 状态寄存器（TIMx_STS）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved												BITF	TITF	COMITF	UDITF	
												rc_w0	rc_w0	rc_w0	rc_w0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved			CC6ITF	CC5ITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF
				rc_w0	rc_w0	rc_w0	rc_w0				rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位域	名称	描述
31:20	Reserved	保留，必须保持复位值
19	BITF	刹车中断标记（Break interrupt flag） 一旦刹车输入有效，由硬件对该位置'1'。如果刹车输入无效，则该位可由软件清'0'。 0：无刹车事件产生； 1：刹车输入上检测到有效电平。
18	TITF	触发器中断标记（Trigger interrupt flag） 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在TRGI输入端检测

位域	名称	描述
		到有效边沿，或门控模式下的任一边沿）时由硬件对该位置'1'。它由软件清'0'。 0：无触发器事件产生； 1：触发中断等待响应。
17	COMITF	COM中断标记（COM interrupt flag） 一旦产生COM事件（当捕获/比较控制位：CCxEN、CCxNEN、OCxMD已被更新）该位由硬件置'1'。它由软件清'0'。 0：无COM事件产生； 1：COM中断等待响应。
16	UDITF	更新中断标志（Update interrupt flag） 当在以下条件下发生更新事件时，该位由硬件设置： – 当 TIMx_CTRL1.UPDIS = 0 时，并且重复计数器值上溢或下溢（当重复计数器等于 0 时生成更新事件UEV）。 – 当 TIMx_CTRL1.UPRS = 0 时，TIMx_CTRL1.UPDIS = 0，并通过软件设置 TIMx_EVTGEN.UDGN 位以重新初始化 CNT。 – 当 TIMx_CTRL1.UPRS = 0 时，TIMx_CTRL1.UPDIS = 0，并且计数器 CNT 由触发事件重新初始化。（参见 TIMx_SMCTRL 寄存器说明） 该位由软件清零。 0：未发生更新事件 1：发生更新中断
15:12	Reserved	保留，必须保持复位值
11	CC4OCF	捕获/比较4重复捕获标记（Capture/Compare 4 overcapture flag） 参见CC1OCF描述。
10	CC3OCF	捕获/比较3重复捕获标记（Capture/Compare 3 overcapture flag） 参见CC1OCF描述。
9	CC2OCF	捕获/比较2重复捕获标记（Capture/Compare 2 overcapture flag） 参见CC1OCF描述。
8	CC1OCF	捕获/比较1重复捕获标记（Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置1。写0可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到TIMx_CCDA1寄存器时，CC1ITF的状态已经为'1'。
7:6	Reserved	保留，必须保持复位值
5	CC6ITF	捕获/比较6中断标记（Capture/Compare 6 interrupt flag） 参考CC1ITF描述。
4	CC5ITF	捕获/比较5中断标记（Capture/Compare 5 interrupt flag） 参考CC1ITF描述。
3	CC4ITF	捕获/比较4中断标记（Capture/Compare 4 interrupt flag） 参考CC1ITF描述。
2	CC3ITF	捕获/比较3中断标记（Capture/Compare 3 interrupt flag） 参考CC1ITF描述。
1	CC2ITF	捕获/比较2中断标记（Capture/Compare 2 interrupt flag） 参考CC1ITF描述。
0	CC1ITF	捕获/比较1中断标记（Capture/Compare 1 interrupt flag） 如果通道CC1配置为输出模式：

位域	名称	描述
		<p>除中央对齐模式外，当计数器值与比较值相同时，该位由硬件设置（参见TIMx_CTRL1.CAMSEL 位描述）。该位由软件清零。</p> <p>0：未发生匹配。</p> <p>1：TIMx_CNT 的值与 TIMx_CCDA1 的值相同。</p> <p>当 TIMx_CCDA1 的值大于 TIMx_AR 的值时，如果计数器溢出（在向上计数和向上/向下计数模式下）和向下计数模式下溢，则 TIMx_STS.CC1ITF 位将变为高电平。</p> <p>如果通道CC1配置为输入模式：</p> <p>当捕捉事件发生时，该位由硬件设置。该位由软件或读取 TIMx_CCDA1 清零。</p> <p>0：未发生输入捕捉。</p> <p>1：发生输入捕捉。计数器值已在 TIMx_CCDA1 中捕获。在 IC1 上检测到与所选极性相同的边沿。</p>

9.4.5 事件产生寄存器 (TIMx_EVTGEN)

偏移地址:0x0C

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BGN	TGN	CCUDGN	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN
				w	w	w	w					w	w	w	w

位域	名称	描述
31:12	Reserved	保留，必须保持复位值
11	BGN	<p>产生刹车事件（Break generation）</p> <p>当由软件设置时，该位可以产生一个刹车事件。而此时TIMx_BKDT.MOEN = 0，TIMx_STS.BITF = 1，如果相应的中断和DMA被使能，就会产生相应的中断和DMA。该位由硬件自动清零。</p> <p>0：无动作</p> <p>1：产生刹车事件</p>
10	TGN	<p>产生触发事件（Trigger generation）</p> <p>当由软件置位时，该位可以产生一个触发事件。而此时TIMx_STS.TITF = 1，如果相应的中断和DMA被使能，就会产生相应的中断和DMA。该位由硬件自动清零。</p> <p>0：无动作</p> <p>1：产生触发事件</p>
9	CCUDGN	<p>捕获/比较事件，产生控制更新（Capture/Compare control update generation）</p> <p>该位由软件设置。如果此时 TIMx_CTRL2.CCPCTL = 1，则允许更新 CCxEN、CCxNEN 和 OCxMD 位。该位由硬件自动清零。</p> <p>0：无动作</p> <p>1：产生一个COM事件</p>

位域	名称	描述
		注意：该位仅对具有互补输出的通道有效。
8	UDGN	产生更新事件（Update generation） 该位由软件置'1'，由硬件自动清'0'。 当由软件设置时，该位可以生成更新事件。而此时计数器会重新初始化，预分频计数器会被清零，计数器在中央对齐或向上计数模式下会被清零，但在向下计数模式下取TIMx_AR寄存器的值。 该位由硬件自动清零。 0：无动作 1：生成更新事件
7:4	Reserved	保留，必须保持复位值
3	CC4GN	产生捕获/比较4事件（Capture/Compare 4 generation） 参考CC1GN描述。
2	CC3GN	产生捕获/比较3事件（Capture/Compare 3 generation） 参考CC1GN描述。
1	CC2GN	产生捕获/比较2事件（Capture/Compare 2 generation） 参考CC1GN描述。
0	CC1GN	产生捕获/比较1事件（Capture/Compare 1 generation） 当由软件设置时，该位可以产生一个捕获/比较事件。 该位由硬件自动清零。 CC1对应通道为输出模式时： TIMx_STS.CC1ITF 标志将被拉高，如果相应的中断和 DMA 被使能，就会产生相应的中断和 DMA。 CC1对应通道为输入模式时： TIMx_CC1ITF 将捕获当前计数器值，并将 TIMx_STS.CC1ITF 标志拉高，如果相应的中断和 DMA 被使能，则会产生相应的中断和 DMA。 如果 TIMx_STS.CC1ITF 已经拉高，则拉高 TIMx_STS.CC1OCF。 0：无动作 1：生成 CC1 捕获/比较事件

9.4.6 从模式控制寄存器（TIMx_SMCTRL）

偏移地址：0x10

复位值：0x0000 0000

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

位域	名称	描述
31:17	Reserved	保留，必须保持复位值
16	MSMD	主/从模式（Master/slave mode） 0：无作用； 1：触发输入（TRGI）上的事件被延迟了，以允许在当前定时器（通过TRGO）与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
15:12	EXTF[3:0]	外部触发滤波（External trigger filter） 这些位用于定义 ETRP 信号的采样频率和 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，在记录连续 N 个事件后生成验证输出。 0000：无滤波器，以fDTS采样 1000：采样频率fSAMPLING=fDTS/8，N=6 0001：采样频率fSAMPLING=fCK_INT，N=2 1001：采样频率fSAMPLING=fDTS/8，N=8 0010：采样频率fSAMPLING=fCK_INT，N=4 1010：采样频率fSAMPLING=fDTS/16，N=5 0011：采样频率fSAMPLING=fCK_INT，N=8 1011：采样频率fSAMPLING=fDTS/16，N=6 0100：采样频率fSAMPLING=fDTS/2，N=6 1100：采样频率fSAMPLING=fDTS/16，N=8 0101：采样频率fSAMPLING=fDTS/2，N=8 1101：采样频率fSAMPLING=fDTS/32，N=5 0110：采样频率fSAMPLING=fDTS/4，N=6 1110：采样频率fSAMPLING=fDTS/32，N=6 0111：采样频率fSAMPLING=fDTS/4，N=8 1111：采样频率fSAMPLING=fDTS/32，N=8
11	EXTP	外部触发极性（External trigger polarity） 该位选择是用ETR还是ETR的反相来作为触发操作 0：ETR高电平或上升沿有效； 1：ETR低电平或下降沿有效。
10	EXCEN	外部时钟使能位（External clock enable） 该位启用外部时钟模式2。启用后，计数器由ETRF信号上的任意有效边沿驱动。 0：禁止外部时钟模式2； 1：使能外部时钟模式2。 <i>注 1：当同时使能外部时钟模式 1 和外部时钟模式 2 时，外部时钟的输入为 ETRF。</i> <i>注2：以下从机模式可以与外部时钟模式2同时使用：复位模式、门控模式和触发模式；但是，TRGI 无法连接到 ETRF (TIMx_SMCTRL.TSEL ≠ '111')。</i> <i>注 3：设置 TIMx_SMCTRL.EXCEN 位与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (TIMx_SMCTRL.SMSEL = 111 和 TIMx_SMCTRL.TSEL = 111) 的效果相同</i>
9:8	EXTPS[1:0]	外部触发预分频（External trigger prescaler） 外部触发信号 ETRP 的频率必须最多为 TIMxCLK 频率的 1/4。当输入更快的外部时钟时，可以使用预分频器来降低 ETRP 的频率。 00：关闭预分频； 01：ETRP频率除以2； 10：ETRP频率除以4； 11：ETRP频率除以8。
7	Reserved	保留，必须保持复位值
6:4	SMSSEL[2:0]	从模式选择（Slave mode selection） 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明） 000：关闭从模式 – 如果CNTEN=1，则预分频器直接由内部时钟驱动。 001：编码器模式1 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。 010：编码器模式2 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。011：编码器模式3

		<p>– 根据另一个信号的输入电平，计数器在TI1FP1和TI2FP2的边沿向上/下计数。</p> <p>100：复位模式 – 在选定触发输入（TRGI）的上升沿，计数器重新初始化并更新影子寄存器。</p> <p>101：门控模式 – 当触发输入（TRGI）为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止（但不复位）。计数器的启动和停止都是受控的。</p> <p>110：触发模式 – 计数器在触发输入TRGI的上升沿启动（但不复位），只有计数器的启动是受控的。</p> <p>111：外部时钟模式1 – 选中的触发输入（TRGI）的上升沿驱动计数器。</p> <p><i>注：如果TI1F_ED被选为触发输入（TSEL=100）时，不要使用门控模式。这是因为，TI1F_ED在每次TI1F变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。</i></p>
3	Reserved	保留，必须保持复位值
2:0	TSEL[2:0]	<p>触发选择（Trigger selection）</p> <p>这3位选择用于同步计数器的触发输入。</p> <p>000：内部触发0（ITR0） 100：TI1的边沿检测器（TI1F_ED）</p> <p>001：内部触发1（ITR1） 101：滤波后的定时器输入1（TI1FP1）</p> <p>010：内部触发2（ITR2） 110：滤波后的定时器输入2（TI2FP2）</p> <p>011：内部触发3（ITR3） 111：外部触发输入（ETRF）</p> <p>更多有关ITRx的细节，参见表9-3。</p> <p><i>注：这些位只能在未用到（如SMSEL=000）时被改变，以避免在改变时产生错误的边沿检测。</i></p>

表 9-3 TIMx 内部触发连接

Slave timer	ITR0 (TSEL = 000)	ITR1 (TSEL = 001)	ITR2 (TSEL = 010)	ITR3 (TSEL = 011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM2	TIM1	NA	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	NA
TIM5	TIM2	TIM3	TIM4	NA

9.4.7 DMA/中断使能寄存器（TIMx_DINTEN）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									COMIEN	TDEN	COMDEN	UDEN	BIEN	TIEN	UIEN
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw					rw	rw	rw	rw

位域	名称	描述
31:23	Reserved	保留，必须保持复位值

位域	名称	描述
22	COMIEN	允许COM中断 (COM interrupt enable) 0: 禁止COM中断; 1: 允许COM中断。
21	TDEN	允许触发DMA请求 (Trigger DMA request enable) 0: 禁止触发DMA请求; 1: 允许触发DMA请求。
20	COMDEN	允许COM的DMA请求 (COM DMA request enable) 0: 禁止COM的DMA请求; 1: 允许COM的DMA请求。
19	UDEN	允许更新的DMA请求 (Update DMA request enable) 0: 禁止更新的DMA请求; 1: 允许更新的DMA请求。
18	BIEN	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。
17	TIEN	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
16	UIEN	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。
15:12	Reserved	保留, 必须保持复位值
11	CC4DEN	允许捕获/比较4的DMA请求 (Capture/Compare 4 DMA request enable) 0: 禁止捕获/比较4的DMA请求; 1: 允许捕获/比较4的DMA请求。
10	CC3DEN	允许捕获/比较3的DMA请求 (Capture/Compare 3 DMA request enable) 0: 禁止捕获/比较3的DMA请求; 1: 允许捕获/比较3的DMA请求。
9	CC2DEN	允许捕获/比较2的DMA请求 (Capture/Compare 2 DMA request enable) 0: 禁止捕获/比较2的DMA请求; 1: 允许捕获/比较2的DMA请求。
8	CC1DEN	允许捕获/比较1的DMA请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较1的DMA请求; 1: 允许捕获/比较1的DMA请求。
7:4	Reserved	保留, 必须保持复位值
3	CC4IEN	允许捕获/比较4中断 (Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。
2	CC3IEN	允许捕获/比较3中断 (Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。

位域	名称	描述
1	CC2IEN	允许捕获/比较2中断（Capture/Compare 2 interrupt enable） 0：禁止捕获/比较2中断； 1：允许捕获/比较2中断。
0	CC1IEN	允许捕获/比较1中断（Capture/Compare 1 interrupt enable） 0：禁止捕获/比较1中断； 1：允许捕获/比较1中断。

9.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1)

偏移地址：0x18

复位值：0x0000 0000

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

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:13	OC2MD[2:0]	输出比较2模式（Output Compare 2 mode）
12	OC2CEN	输出比较2清0使能（Output Compare 2 clear enable）
11	OC2FEN	输出比较2快速使能（Output Compare 2 fast enable）
10	OC2PEN	输出比较2预装载使能（Output Compare 2 preload enable）
9:8	CC2SEL[1:0]	捕获/比较2选择。（Capture/Compare 2 selection） 该位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:5	OC1MD[2:0]	输出比较1模式（Output Compare 1 mode） 这些位用于管理输出参考信号 OC1REF，它决定了 OC1 和 OC1N 的值，在高电平有效，而 OC1 和 OC1N 的有效电平取决于 TIMx_CCEN.CC1P 和 TIMx_CCEN.CC1NP 位。 000：冻结。TIMx_CC DAT1 寄存器和计数器 TIMx_CNT 之间的比较对 OC1REF

位域	名称	描述
		<p>信号没有影响。</p> <p>001: 将通道 1 设置为匹配时的有效电平。当 $TIMx_CCDAT1 = TIMx_CNT$ 时, OC1REF 信号将被强制为高电平。</p> <p>010: 将通道 1 设置为匹配时的无效电平。当 $TIMx_CCDAT1 = TIMx_CNT$ 时, OC1REF 信号将被强制为低电平。</p> <p>011: 翻转。当 $TIMx_CCDAT1 = TIMx_CNT$ 时, OC1REF 信号将被翻转。</p> <p>100: 强制无效电平。OC1REF 信号被强制为低电平。</p> <p>101: 强制有效电平。OC1REF 信号被强制为高电平。</p> <p>110: PWM 模式 1 - 在向上计数模式下, 如果 $TIMx_CNT < TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。在向下计数模式下, 如果 $TIMx_CNT > TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。</p> <p>111: PWM 模式 2 - 在向上计数模式下, 如果 $TIMx_CNT < TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。在向下计数模式下, 如果 $TIMx_CNT > TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。</p> <p><i>注 1: 在 PWM 模式 1 或 PWM 模式 2 中, OC1REF 电平仅在比较结果改变或输出比较模式从冻结模式切换到 PWM 模式时才会改变。</i></p>
4	OC1CEN	<p>输出比较1清'0'使能 (Output Compare 1 clear enable)</p> <p>0: OC1REF 不受ETRF输入的影响;</p> <p>1: 一旦检测到ETRF输入高电平, 清除OC1REF=0。</p>
3	OC1FEN	<p>输出比较1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快CC输出对触发输入事件的响应。</p> <p>0: 根据计数器与CCDAT1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就像发生了一次比较匹配。因此, OC1被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。</p> <p>OCxFEN只在通道被配置成PWM1或PWM2模式时起作用。</p>
2	OC1PEN	<p>输出比较 1 预加载使能 (Output Compare 1 preload enable)</p> <p>0: 禁用 $TIMx_CCDAT1$ 寄存器的预加载功能。支持随时对$TIMx_CCDAT1$寄存器进行写操作, 写入的值立即生效。</p> <p>1: 使能 $TIMx_CCDAT1$ 寄存器的预加载功能。仅对预加载寄存器进行读写操作。当更新事件发生时, $TIMx_CCDAT1$ 的值被加载到影子寄存器中。</p> <p><i>注 1: 只有当 $TIMx_CTRL1.ONEPM = 1$ (在单脉冲模式下) 时, 才能使用 PWM 模式而不验证预加载寄存器, 否则无法预测其他行为。</i></p>
1:0	CC1SEL[1:0]	<p>捕获/比较1 选择。(Capture/Compare 1 selection)</p> <p>这2位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(由$TIMx_SMCTRL$寄存器的TSEL位选择)。</p>

位域	名称	描述
		注：CC1SEL仅在通道关闭时（TIMx_CCEN寄存器的CC1EN=0）才是可写的。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw		rw		rw				rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC2F[3:0]	输入捕获2滤波器（Input capture 2 filter）
11:10	IC2PSC[1:0]	输入/捕获2预分频器（Input capture 2 prescaler）
9:8	CC2SEL[1:0]	捕获/比较2选择（Capture/Compare 2 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。
7:4	IC1F[3:0]	输入捕获1滤波器（Input capture 1 filter） 这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以f _{DTS} 采样 1000：采样频率f _{SAMPLING} =f _{DTS} /8，N=6 0001：采样频率f _{SAMPLING} =f _{CK_INT} ，N=2 1001：采样频率f _{SAMPLING} =f _{DTS} /8，N=8 0010：采样频率f _{SAMPLING} =f _{CK_INT} ，N=4 1010：采样频率f _{SAMPLING} =f _{DTS} /16，N=5 0011：采样频率f _{SAMPLING} =f _{CK_INT} ，N=8 1011：采样频率f _{SAMPLING} =f _{DTS} /16，N=6 0100：采样频率f _{SAMPLING} =f _{DTS} /2，N=6 1100：采样频率f _{SAMPLING} =f _{DTS} /16，N=8 0101：采样频率f _{SAMPLING} =f _{DTS} /2，N=8 1101：采样频率f _{SAMPLING} =f _{DTS} /32，N=5 0110：采样频率f _{SAMPLING} =f _{DTS} /4，N=6 1110：采样频率f _{SAMPLING} =f _{DTS} /32，N=6 0111：采样频率f _{SAMPLING} =f _{DTS} /4，N=8 1111：采样频率f _{SAMPLING} =f _{DTS} /32，N=8
3:2	IC1PSC[1:0]	输入/捕获1预分频器（Input capture 1 prescaler） 这2位定义了CC1输入（IC1）的预分频系数。 一旦TIMx_CCEN.CC1EN=0，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	CC1SEL[1:0]	<p>捕获/比较1选择 (Capture/Compare 1 Selection)</p> <p>这2位定义通道的方向 (输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。</p> <p><i>注: CC1SEL仅在通道关闭时 (TIMx_CCEN寄存器的CC1EN=0) 才是可写的。</i></p>
-----	-------------	--

9.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2)

偏移地址: 0x1C

复位值: 0x0000 0000

参看以上 CCMOD1 寄存器的描述

输出比较模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]			OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]			OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:13	OC4MD[2:0]	输出比较4模式 (Output compare 4 mode)
12	OC4CEN	输出比较4清0使能 (Output compare 4 clear enable)
11	OC4FEN	输出比较4快速使能 (Output compare 4 fast enable)
10	OC4PEN	输出比较4预装载使能 (Output compare 4 preload enable)
9:8	CC4SEL[1:0]	<p>捕获/比较4选择 (Capture/Compare 4 selection)</p> <p>该2位定义通道的方向 (输入/输出), 及输入脚的选择:</p> <p>00: CC4通道被配置为输出;</p> <p>01: CC4通道被配置为输入, IC4映射在TI4上;</p> <p>10: CC4通道被配置为输入, IC4映射在TI3上;</p> <p>11: CC4通道被配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。</p> <p><i>注: CC4SEL仅在通道关闭时 (TIMx_CCEN寄存器的CC4EN=0) 才是可写的。</i></p>
7:5	OC3MD[2:0]	输出比较3模式 (Output compare 3 mode)
4	OC3CEN	输出比较3清0使能 (Output compare 3 clear enable)
3	OC3FEN	输出比较3快速使能 (Output compare 3 fast enable)
2	OC3PEN	输出比较3预装载使能 (Output compare 3 preload enable)
1:0	CC3SEL[1:0]	<p>捕获/比较3选择 (Capture/Compare 3 selection)</p> <p>这2位定义通道的方向 (输入/输出), 及输入脚的选择:</p>

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
		00: CC3通道被配置为输出； 01: CC3通道被配置为输入，IC3映射在TI3上； 10: CC3通道被配置为输入，IC3映射在TI4上； 11: CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC3SEL仅在通道关闭时（TIMx_CCEN寄存器的CC3EN=0）才是可写的。</i>

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw		rw		rw				rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC4F[3:0]	输入捕获4滤波器 (Input capture 4 filter)
11:10	IC4PSC[1:0]	输入/捕获4预分频器 (Input capture 4 prescaler)
9:8	CC4SEL[1:0]	捕获/比较4选择 (Capture/Compare 4 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4通道被配置为输出； 01: CC4通道被配置为输入，IC4映射在TI4上； 10: CC4通道被配置为输入，IC4映射在TI3上； 11: CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注：CC4SEL仅在通道关闭时(TIMx_CCEN寄存器的CC4EN=0)才是可写的。</i>
7:4	IC3F[3:0]	输入捕获3滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获3预分频器 (Input capture 3 prescaler)
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3通道被配置为输出； 01: CC3通道被配置为输入，IC3映射在TI3上； 10: CC3通道被配置为输入，IC3映射在TI4上； 11: CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注：CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。</i>

9.4.10 捕获/比较模式寄存器 3 (TIMx_CCMOD3)

偏移地址：0x20

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC9PEN	Reserved			OC8PEN	Reserved			OC7PEN
rw							rw			rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6MD[2:0]			OC6CEN	OC6FEN	OC6PEN	Reserved		OC5MD[2:0]			OC5CEN	OC5FEN	OC5PEN	Reserved	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:25	Reserved	保留，必须保持复位值
24	OC9PEN	输出比较9预装载使能（Output compare 9 preload enable）
23:21	Reserved	保留，必须保持复位值
20	OC8PEN	输出比较8预装载使能（Output compare 8 preload enable）
19:17	Reserved	保留，必须保持复位值
16	OC7PEN	输出比较7预装载使能（Output compare 7 preload enable）
15:13	OC6MD[2:0]	输出比较6模式（Output compare 6 mode）
12	OC6CEN	输出比较6清除使能（Output compare 6 clear enable）
11	OC6FEN	输出比较6快速使能（Output compare 6 fast enable）
10	OC6PEN	输出比较6预装载使能（Output compare 6 preload enable）
9:8	Reserved	保留，必须保持复位值
7:5	OC5MD[2:0]	输出比较5模式（Output compare 5 mode）
4	OC5CEN	输出比较5清除使能（Output compare 5 clear enable）
3	OC5FEN	输出比较5快速使能（Output compare 5 fast enable）
2	OC5PEN	输出比较5预装载使能（Output compare 5 preload enable）
1:0	Reserved	保留，必须保持复位值

9.4.11 捕获/比较使能寄存器（TIMx_CCEN）

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							CC6P	CC6EN	Reserved			CC5P	CC5EN	Reserved	
rw							rw	rw	rw			rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4P	CC4EN	CC4NP	CC4NEN	CC3P	CC3EN	CC3NP	CC3NEN	CC2P	CC2EN	CC2NP	CC2NEN	CC1P	CC1EN	CC1NP	CC1NEN

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位域	名称	描述															
31:24	Reserved	保留，必须保持复位值															
23	CC6P	捕获/比较6输出极性（Capture/Compare 6 output polarity） 参考TIMx_CCEN.CC1P的描述。															
22	CC6EN	捕获/比较6输出使能（Capture/Compare 6 output enable） 参考TIMx_CCEN.CC1EN 的描述															
21:20	Reserved	保留，必须保持复位值															
19	CC5P	捕获/比较5输出极性（Capture/Compare 5 output polarity） 参考TIMx_CCEN.CC1P的描述。															
18	CC5EN	捕获/比较5输出使能（Capture/Compare 5 output enable） 参考TIMx_CCEN.CC1EN 的描述															
17: 16	Reserved	保留，必须保持复位值															
15	CC4P	捕获/比较4输出极性（Capture/Compare 4 output polarity） 参考TIMx_CCEN.CC1P的描述。															
14	CC4EN	捕获/比较4输出使能（Capture/Compare 4 output enable） 参考TIMx_CCEN.CC1EN 的描述。															
13	CC4NP	捕获/比较4互补输出极性（Capture/Compare 4 complementary output polarity） 参考TIMx_CCEN.CC1NP的描述。															
12	CC4NEN	捕获/比较4互补输出使能（Capture/Compare 4 complementary output enable） 参考TIMx_CCEN.CC1NEN的描述。															
11	CC3P	捕获/比较3输出极性（Capture/Compare 3 output polarity） 参考TIMx_CCEN.CC1P的描述。															
10	CC3EN	捕获/比较3输出使能（Capture/Compare 3 output enable） 参考TIMx_CCEN.CC1E 的描述。															
9	CC3NP	捕获/比较3互补输出极性（Capture/Compare 3 complementary output polarity） 参考TIMx_CCEN.CC1NP的描述。															
8	CC3NEN	捕获/比较3互补输出使能（Capture/Compare 3 complementary output enable） 参考TIMx_CCEN.CC1NEN的描述。															
7	CC2P	捕获/比较2输出极性（Capture/Compare 2 output polarity） 参考TIMx_CCEN.CC1P的描述。															
6	CC2EN	捕获/比较2输出使能（Capture/Compare 2 output enable） 参考TIMx_CCEN.CC1EN的描述。															
5	CC2NP	捕获/比较2互补输出极性（Capture/Compare 2 complementary output polarity） 参考TIMx_CCEN.CC1NP的描述。															
4	CC2NEN	捕获/比较2互补输出使能（Capture/Compare 2 complementary output enable） 参考TIMx_CCEN.CC1NEN的描述。															
3	CC1P	捕获/比较1输出极性（Capture/Compare 1 output polarity） CC1对应通道为输出模式时： 0: OC1 高电平有效 1: OC1 低电平有效 CC1对应通道为输入模式时： 此时，该位用于选择是使用IC1还是IC1的反相信号作为触发信号或捕捉信号。															

位域	名称	描述
		0: 非反相: 当 IC1 产生上升沿时发生捕获动作。 当用作外部触发时, IC1 是非反相的。 1: 反相: 当 IC1 产生下降沿时发生捕获动作。 当用作外部触发时, IC1 被反相。
2	CC1EN	捕获/比较1输出使能 (Capture/Compare 1 output enable) CC1通道配置为输出: 0: 关闭 - OC1禁止输出, 因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1EN位的值。 1: 开启 - OC1信号输出到对应的输出引脚, 其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1EN位的值。 CC1通道配置为输入: 该位决定了计数器的值是否能捕获入TIMx_CCDA1寄存器。 0: 捕获禁止; 1: 捕获使能。
1	CC1NP	捕获/比较1互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N高电平有效; 1: OC1N低电平有效。
0	CC1EN	捕获/比较1互补输出使能 (Capture/Compare 1 complementary output enable) 0: 禁用 - 禁用输出 OC1N 信号。 OC1N 的电平取决于TIMx_BKDT.MOEN、TIMx_BKDT.OSSI、TIMx_BKDT.OSSR、TIMx_CTRL2.OI1、TIMx_CTRL2.OI1N 和 TIMx_CCEN.CC1EN 的值。 1: 使能 - 使能输出 OC1N 信号。 OC1N 的电平取决于TIMx_BKDT.MOEN、TIMx_BKDT.OSSI、TIMx_BKDT.OSSR、TIMx_CTRL2.OI1、TIMx_CTRL2.OI1N 和 TIMx_CCEN.CC1EN 的值。

表 9-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态 ⁽¹⁾	
MOEN	OSSI	OSSR	CCxEN	CCxNEN	OCx 输出状态	OCxN 输出状态
1X		0	0	0	输出禁止 (与定时器断开)	输出禁止 (与定时器断开)
					OCx=0, OCx_EN=0	OCxN=0, OCxN_EN=0
		0	0	1	输出禁止 (与定时器断开)	OCxREF + 极性,
					OCx=0, OCx_EN=0	OCxN= OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + 极性,	输出禁止 (与定时器断开)
					OCx= OCxREF xor CCxP, OCx_EN=1	OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
					输出禁止 (与定时器断开)	输出禁止 (与定时器断开)
		1	0	0	OCx=CCxP, OCx_EN=0	OCxN=CCxNP, OCxN_EN=0

		1	0	1	关闭状态（输出使能且为无效电平）	OCxREF + 极性，
					OCx=CCxP, OCx_EN=1	OCxN= OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性，	关闭状态（输出使能且为无效电平）
					OCx= OCxREF xor CCxP, OCx_EN=1	OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区，	OCxREF 反相 + 极性 + 死区，
					OCx_EN=1	OCxN_EN=1
0	X	0	0	0	输出禁止（与定时器断开）	
		0	0	1	异步： OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0;	
		0	1	0	若时钟存在：假设 $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$,	
		0	1	1	经过一个死区时间后 OCx=OIx, OCxN=OIxN	
		1	0	0	关闭状态（输出使能且为无效电平）	
		1	0	1	异步： OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1;	
		1	1	0	若时钟存在：假设 $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$,	
		1	1	1	经过一个死区时间后 OCx=OIx, OCxN=OIxN,	

1. 如果一个通道的 2 个输出都没有使用（CCxEN = CCxNEN = 0），那么 OIx, OIxN, CCxP 和 CCxNP 都必须清零。

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

9.4.12 捕获/比较寄存器 1 (TIMx_CC DAT1)

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT1[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT1[15:0]															
rw															

位域	名称	描述
31:16	CCDDAT1[15:0]	<p>捕获/比较通道1向下计数值(Capture/Compare 1 down-counting value), 专用于中央对齐非对称模式。</p> <p>■ CC1 通道只能配置为输出:</p> <p>CCDDAT1 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC1 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p>
15:0	CCDAT1[15:0]	<p>捕获/比较通道1的值 (Capture/Compare 1 value)</p> <p>■ CC1 通道配置为输出:</p> <p>CCDAT1 包含要与计数器 TIMx_CNT 比较的值, 在 OC1 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p> <p>■ CC1 通道配置为输入:</p> <p>CCDAT1 包含由最后一个输入捕获 1 事件 (IC1) 传输的计数值。</p> <p>当配置为输入模式时, 寄存器 CCDAT1 和 CCDDAT1 只能读取。</p> <p>当配置为输出模式时, 寄存器 CCDAT1 和 CCDDAT1 是可读写的。</p>

9.4.13 捕获/比较寄存器 2 (TIMx_CCDAT2)

偏移地址: 0x2C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT2[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT2[15:0]															
rw															

位域	名称	描述
31:16	CCDDAT2[15:0]	<p>捕获/比较通道2向下计数值(Capture/Compare 2 down-counting value), 专用于中央对齐非对称模式。</p> <p>■ CC2 通道只能配置为输出:</p> <p>CCDDAT2 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC2 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p>
15:0	CCDAT2[15:0]	<p>捕获/比较通道2的值 (Capture/Compare 2 value)</p> <p>■ CC2 通道配置为输出:</p> <p>CCDAT2 包含要与计数器 TIMx_CNT 比较的值, 在 OC2 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能, 则写入的值会立即传输到有效</p>

位域	名称	描述
31:16	CCDDAT2[15:0]	<p>捕获/比较通道2向下计数值(Capture/Compare 2 down-counting value), 专用于中央对齐非对称模式。</p> <p>■ CC2 通道只能配置为输出: CCDDAT2 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC2 输出上发出信号。 如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p>
		<p>寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p> <p>■ CC2 通道配置为输入: CCDAT2 包含由最后一个输入捕获 2 事件 (IC2) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT2 和 CCDDAT2 只能读取。 当配置为输出模式时, 寄存器 CCDAT2 和 CCDDAT2 是可读写的。</p>

9.4.14 捕获/比较寄存器 3 (TIMx_CCDAT3)

偏移地址: 0x30

复位值: 0x0000 0000

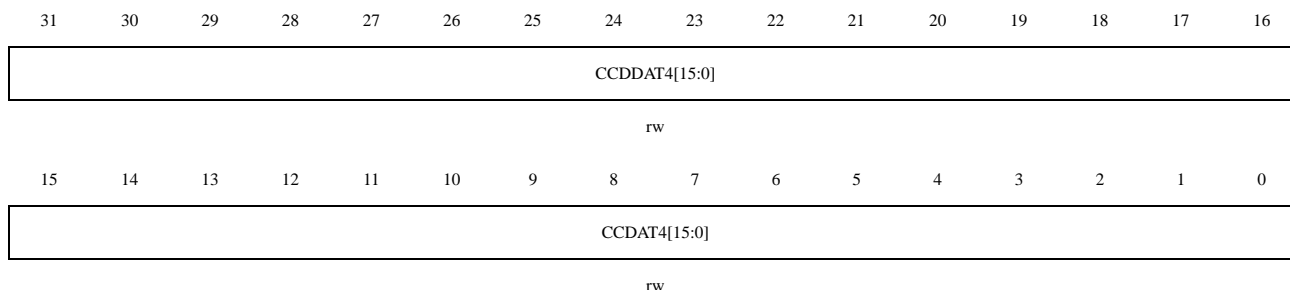
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT3[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT3[15:0]															
rw															

位域	名称	描述
31:16	CCDDAT3[15:0]	<p>捕获/比较通道3向下计数值(Capture/Compare 3 down-counting value), 专用于中央对齐非对称模式。</p> <p>■ CC3 通道只能配置为输出: CCDDAT3 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p>
15:0	CCDAT3[15:0]	<p>捕获/比较通道3的值 (Capture/Compare 3 value)</p> <p>■ CC3 通道配置为输出: CCDAT3 包含要与计数器 TIMx_CNT 比较的值, 在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p> <p>■ CC3 通道配置为输入: CCDAT3 包含由最后一个输入捕获 3 事件 (IC3) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT3 和 CCDDAT3 只能读取。 当配置为输出模式时, 寄存器 CCDAT3 和 CCDDAT3 是可读写的。</p>

9.4.15 捕获/比较寄存器 4 (TIMx_CC DAT4)

偏移地址: 0x34

复位值: 0x0000 0000

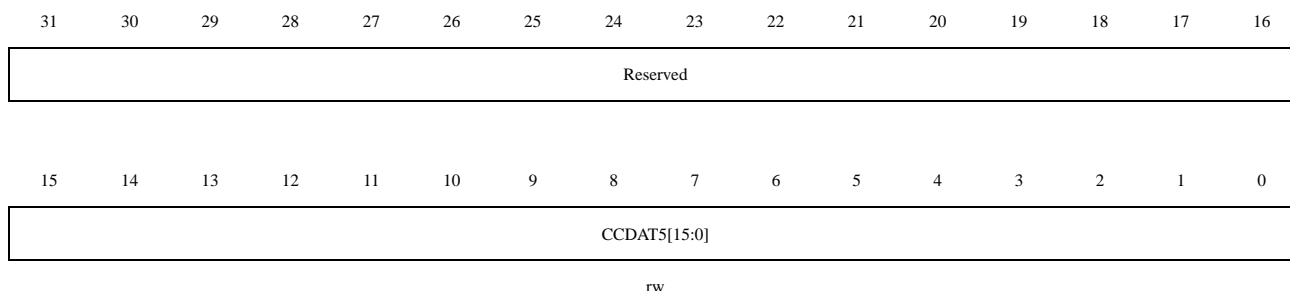


位域	名称	描述
31:16	CCDDAT4[15:0]	<p>捕获/比较通道4向下计数值(Capture/Compare 4 down-counting value), 专用于中央对齐非对称模式。</p> <p>■ CC4 通道只能配置为输出:</p> <p>CCDDAT4 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC4 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD2.OC4PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p>
15:0	CCDAT4[15:0]	<p>捕获/比较通道4的值 (Capture/Compare 4 value)</p> <p>■ CC4 通道配置为输出:</p> <p>CCDAT4 包含要与计数器 TIMx_CNT 比较的值, 在 OC4 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD2.OC4PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。</p> <p>■ CC4 通道配置为输入:</p> <p>CCDAT4 包含由最后一个输入捕获 4 事件 (IC4) 传输的计数器值。</p> <p>当配置为输入模式时, 寄存器 CCDAT4 和 CCDDAT4 只能读取。</p> <p>当配置为输出模式时, 寄存器 CCDAT4 和 CCDDAT4 是可读写的。</p>

9.4.16 捕获/比较寄存器 5 (TIMx_CC DAT5)

偏移地址: 0x38

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT5[15:0]	<p>捕获/比较通道5的值（Capture/Compare 5 value）</p> <p>■ CC5 通道只能配置为输出：</p> <p>CCDAT5 包含要与计数器 TIMx_CNT 比较的值，在 OC5 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD3.OC5PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</p> <p>CC5 用于比较器消隐。</p>

9.4.17 捕获/比较寄存器 6 (TIMx_CCDAT6)

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT6[15:0]															

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT6[15:0]	<p>捕获/比较通道6的值（Capture/Compare 6 value）</p> <p>■ CC6 通道只能配置为输出：</p> <p>CCDAT6 包含要与计数器 TIMx_CNT 比较的值，在 OC6 输出上发出信号。</p> <p>如果未在 TIMx_CCMOD3.OC6PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</p>

9.4.18 预分频器 (TIMx_PSC)

偏移地址：0x40

复位值：0x0000 0000

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

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	PSC[15:0]	<p>预分频器的值（Prescaler value）</p> <p>计数器时钟 $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$。</p> <p>每次发生更新事件时，PSC 值都会加载到预分频器的影子寄存器中。</p>

9.4.19 自动重装载寄存器（TIMx_AR）

偏移地址:0x44

复位值: 0x0000 FFFF

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	AR[15:0]	<p>自动重装载的值（Auto-reload value）</p> <p>AR包含了将要装载入实际的自动重装载寄存器的值。详细参考9.3.1节：有关AR的更新和动作。</p> <p>当自动重装载的值为空时，计数器不工作。</p>

9.4.20 计数器（TIMx_CNT）

偏移地址: 0x48

复位值: 0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CNT[15:0]	计数器的值（Counter value）

9.4.21 重复计数寄存器 (TIMx_REPCNT)

偏移地址: 0x4C

复位值: 0x0000 0000

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

rw

位域	名称	描述
31:8	Reserved	保留, 必须保持复位值
7:0	REPCNT[7:0]	重复计数器的值 (Repetition counter value) 重复计数器仅在给定数量 (N+1) 个计数器周期后用于生成更新事件或更新定时器寄存器, 其中 N 是 TIMx_REPCNT.REPCNT 的值。在向上计数模式下, 每次计数器溢出, 向下计数模式下每次计数器下溢或中央对齐模式下每次计数器溢出和每次计数器下溢时, 重复计数器都会递减。设置 TIMx_EVTGEN.UDGN 位将重新加载 TIMx_REPCNT.REPCNT 的内容并生成更新事件。

9.4.22 刹车和死区寄存器 (TIMx_BKDT)

偏移地址: 0x50

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKCFG[1:0]	OSSR	OSSI	BKEN	BKP	AOEN	MOEN	DTGN[7:0]								

rw

rw

rw

rw

rw

rw

rw

rw

注释: 根据锁定设置, AOEN、BKP、BKEN、OSSI、OSSR 和 DTGN[7:0]位均可被写保护, 有必要在第一次写入 TIMx_BKDT 寄存器时对它们进行配置。

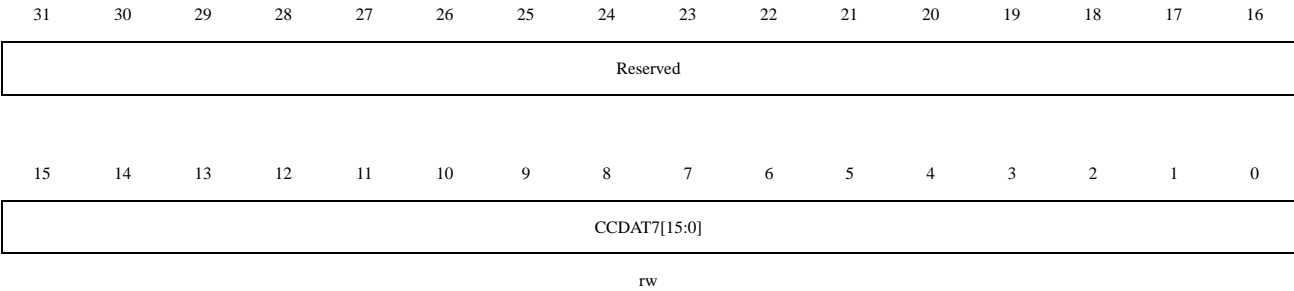
位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:14	LCKCFG[1:0]	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 这些位提供针对软件错误的写保护。 00: – 没有写保护。 01:

		<p>– 锁定级别 1</p> <p>TIMx_BKDT.DTGN、TIMx_BKDT.BKEN、TIMx_BKDT.BKP、TIMx_BKDT.AOEN、TIMx_CTRL2.OIx、TIMx_CTRL2.OIxN 位启用写保护。</p> <p>10:</p> <p>– 锁定 2 级</p> <p>除了 LOCK Level 1 模式下的寄存器写保护外，TIMx_CCEN.CCxP 和 TIMx_CCEN.CCxNP（如果相应通道配置为输出模式），TIMx_BKDT.OSSR 和 TIMx_BKDT.OSSI 位也使能写保护。</p> <p>11:</p> <p>– 锁定 3 级</p> <p>除了 LOCK Level 2 中的寄存器写保护外，TIMx_CCMODx.OCxMD 和 TIMx_CCMODx.OCxPEN 位（如果相应通道配置为输出模式）也启用写保护。</p> <p>注意：定时器复位后，LCKCFG 位只能写一次。一旦写入 TIMx_BKDT 寄存器，LCKCFG 将受到保护，直到下一次复位。</p>
13	OSSR	<p>当 TIMx_BKDT.MOEN=1 且通道为互补输出时使用该位。</p> <p>没有互补输出的定时器中不存在 OSSR 位。</p> <p>0：当定时器不工作时，禁止OC/OCN输出（OC/OCN使能输出信号=0）；</p> <p>1：当定时器不工作时，一旦CCxEN=1或CCxNEN=1，首先开启OC/OCN并输出无效电平，然后置OC/OCN使能输出信号=1。</p> <p>有关更多详细信息，请参见第9.4.11节，捕获/比较使能寄存器（TIMx_CCEN）。</p>
12	OSSI	<p>空闲模式下“关闭状态”选择（Off-state selection for Idle mode）</p> <p>当 TIMx_BKDT.MOEN=0 且通道配置为输出时使用该位。</p> <p>0：当定时器不工作时，禁止OC/OCN输出（OC/OCN使能输出信号=0）；</p> <p>1：当定时器不工作时，一旦CCxEN=1 或CCxNEN=1，OC/OCN首先输出其空闲电平，然后OC/OCN使能输出信号=1。</p> <p>有关更多详细信息，请参见第9.4.11节，捕获/比较启用寄存器（TIMx_CCEN）。</p>
11	BKEN	<p>刹车功能使能（Break enable）</p> <p>0：禁止刹车输入（BRK及CCS时钟失效事件）；</p> <p>1：开启刹车输入（BRK及CCS时钟失效事件）。</p> <p>注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。</p>
10	BKP	<p>刹车输入极性（Break polarity）</p> <p>0：刹车输入低电平有效；</p> <p>1：刹车输入高电平有效。</p> <p>注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。</p>
9	AOEN	<p>自动输出使能（Automatic output enable）</p> <p>0：只有软件可以设置TIMx_BKDT.MOEN；</p> <p>1：软件设置TIMx_BKDT.MOEN；或者如果刹车输入未激活，则在下一次更新事件发生时，硬件自动设置 TIMx_BKDT.MOEN。</p>
8	MOEN	<p>主输出使能（Main output enable）</p> <p>该位可由软件或硬件根据 TIMx_BKDT.AOEN 位设置，一旦刹车输入有效，该位由硬件异步清零。它仅对配置为输出的通道有效。</p> <p>0：OC 和 OCN 输出被禁用或强制进入空闲状态。</p> <p>1：如果设置了 TIMx_CCEN.CCxEN 或 TIMx_CCEN.CCxNEN 位，则使能 OC 和 OCN 输出。有关更多详细信息，请参见第 9.4.11节捕获/比较使能寄存器（TIMx_CCEN）。</p>

7:0	DTGN[7:0]	死区发生器设置（Dead-time generator setup） 这些位定义插入的互补输出之间的死区持续时间。DTGN值与死区时间的关系如下： $DTGN[7:5]=0xx \Rightarrow DT=DTGN[7:0] \times T_{dtgn}, T_{dtgn} = T_{DTS};$ $DTGN[7:5]=10x \Rightarrow DT=(64+DTGN[5:0]) \times T_{dtgn}, T_{dtgn} = 2 \times T_{DTS};$ $DTGN[7:5]=110 \Rightarrow DT=(32+DTGN[4:0]) \times T_{dtgn}, T_{dtgn} = 8 \times T_{DTS};$ $DTGN[7:5]=111 \Rightarrow DT=(32+DTGN[4:0]) \times T_{dtgn}, T_{dtgn} = 16 \times T_{DTS};$
-----	-----------	---

9.4.23 捕获/比较寄存器 7（TIMx_CCDA7）

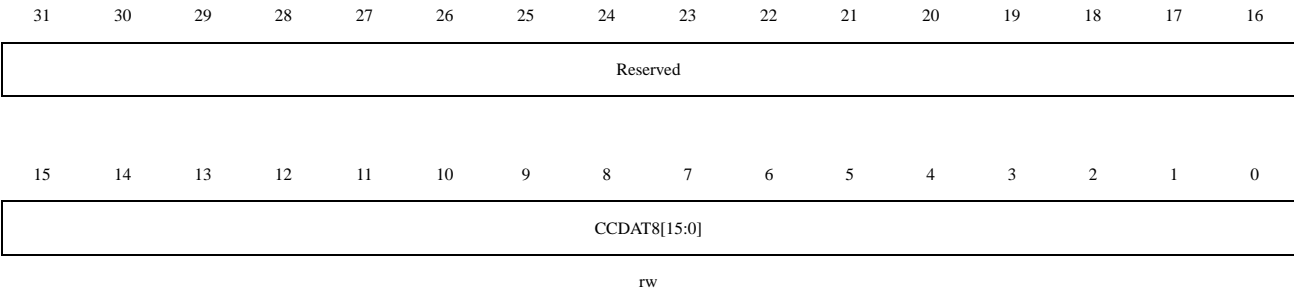
偏移地址：0x54
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT7[15:0]	捕获/比较通道7的值（Capture/Compare 7 value） ■ CC7 通道只能配置为输出： CCDAT7 包含要与计数器 TIMx_CNT 比较的值，在 OC7 输出上发出信号。 如果未在 TIMx_CCMOD3.OC7PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。

9.4.24 捕获/比较寄存器 8（TIMx_CCDA8）

偏移地址：0x58
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT8[15:0]	捕获/比较通道8的值（Capture/Compare 8 value） <div> <div>■ CC8 通道只能配置为输出：</div> <div>CCDAT8 包含要与计数器 TIMx_CNT 比较的值，在 OC8 输出上发出信号。</div> <div>如果未在 TIMx_CCMOD3.OC8PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</div> </div>

9.4.25 捕获/比较寄存器 9（TIMx_CCDAT9）

偏移地址：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT9[15:0]															
rw															

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT9[15:0]	捕获/比较通道9的值（Capture/Compare 9 value） <div> <div>■ CC9 通道只能配置为输出：</div> <div>CCDAT9 包含要与计数器 TIMx_CNT 比较的值，在 OC9 输出上发出信号。</div> <div>如果未在 TIMx_CCMOD3.OC9PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</div> </div>

9.4.26 刹车滤波寄存器（TIMx_BKFR）

偏移地址：0x60

复位值：0x0000 0000

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

位域	名称	描述
----	----	----

31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： 最小值： 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ 最大值： 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 3.2*（预分频时钟周期），则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:0	PSC[15:0]	用于配置滤波器采样时钟的预分频寄存器值（Prescaler）： 对于此过滤器，它支持 65535 分频（16 位）。 时钟预分频器将系统时钟缩放到采样时钟。采样时钟决定两个采样点之间的距离。只有采样点的值有效才会考虑有效的逻辑电平计算。

9.4.27 DMA 控制寄存器（TIMx_DCTRL）

偏移地址：0x94

复位值：0x0000 0000

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

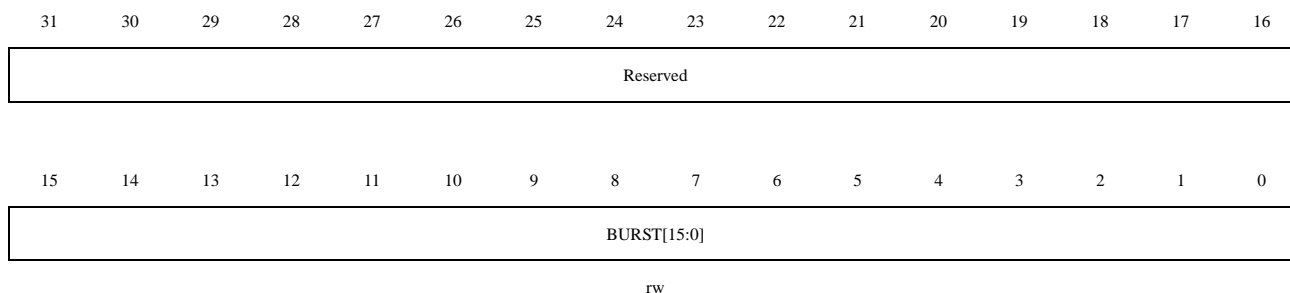
位域	名称	描述
31:13	Reserved	保留，必须保持复位值
12:8	DBADDR[4:0]	DMA基地址（DMA base address） 该位字段定义 DMA 访问 TIMx_DADDR 寄存器的第一个地址。 当第一次通过 TIMx_DADDR 完成访问时，该位域指定您刚刚访问的地址。然后第二次访问TIMx_DADDR，会访问到“DMA Base Address + 4”的地址 00000: TIMx_CTRL1， 00001: TIMx_CTRL2，

位域	名称	描述
		00010: TIMx_SMCTRL, 10001: TIMx_BKDT 10010: TIMx_DCTRL
7:5	Reserved	保留, 必须保持复位值
4:0	DBLEN[4:0]	DMA连续传送长度 (DMA burst length) 该位字段定义 DMA 将访问 (写入/读取) TIMx_DADDR 寄存器的次数。 00000: 1次传输 00001: 2次传输 00010: 3次传输 ... 10001: 18次传输

9.4.28 连续模式的DMA地址 (TIMx_DADDR)

偏移地址: 0x98

复位值: 0x0000



位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:0	BURST[15:0]	DMA 访问缓冲区。 当对该寄存器分配读或写操作时, 将访问位于地址范围 (DMA base address + DMA burst length × 4) 的寄存器。 DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL.DBADDR * 4; DMA burst len = TIMx_DCTRL.DBLEN + 1. 例子: 如果 TIMx_DCTRL.DBLEN = 0x3 (4 次传输), TIMx_DCTRL.DBADDR = 0xD (TIMx_CCDA1), DMA 数据长度 = 半字, DMA 存储器地址 = SRAM 中的缓冲区地址, DMA 外设地址 = TIMx_DADDR 地址。 当事件发生时, TIMx 将向 DMA 发送请求, 并传输 4 次数据。 第一次, 对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CCDA1 寄存器; 第二次, 对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CCDA2 寄存器;

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
	 第四次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT4 寄存器；

10 通用定时器（TIM2/TIM3/TIM4/TIM5）

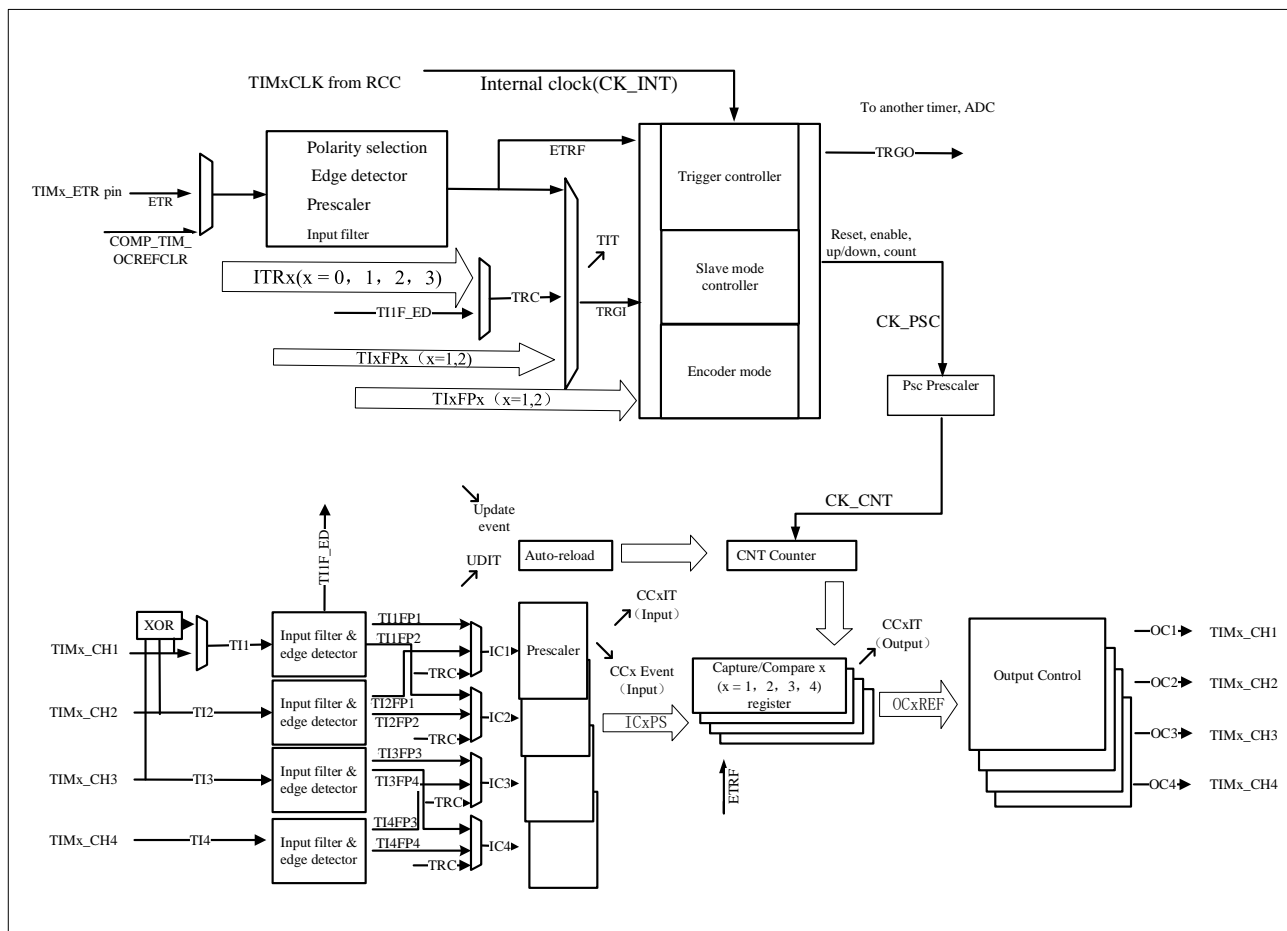
10.1 TIM2/TIM3/TIM4/TIM5 简介

通用定时器（TIM2、TIM3、TIM4 和 TIM5）主要用于以下场合：对输入信号进行计数、测量输入信号的脉冲宽度和产生输出波形等。

10.2 TIM2/TIM3/TIM4/TIM5 主要特性

- 16 位自动装载计数器。（可实现向上计数、向下计数、向上/下计数）。
- 16 位可编程预分频器。（分频系数可配置为 1 到 65536 之间的任意值）
- TIM2、TIM3、TIM4 和 TIM5 最多支持 4 个通道
- 通道工作模式：PWM 输出、输出比较、单脉冲模式输出、输入捕获
- 如下事件发生时产生中断/DMA：
 - ◆ 更新事件
 - ◆ 触发事件
 - ◆ 输入捕获
 - ◆ 输出比较
- 可通过外部信号控制定时器
- 多个定时器内部连接在一起，以实现定时器的同步或链接
- 增量（正交）编码器接口：用于追踪运行轨迹和解析旋转方位
- 霍尔传感器接口：用于三相电机控制
- 支持捕获内部比较器输出信号。

图 10-1 TIMx 框图



↓ The event ↑ Interrupt and DMA output

1. For TIMx (x = 2,3,4,5), the ETR input is from IOM or COMP output, for TIM5, the ETR input is not support.
2. For TIMx (x = 2,3,4,5), The capture channel 1/2/3/4 input can come from IOM or comparator output.
3. For TIM2, capture channel 1 comes from IOM/COMP or HSI, capture channel 3 comes from IOM/COMP or LSI, capture channel 4 comes from IOM/COMP or HSE/128.
4. For each GPTIM, there is a sliding filter in front of TIMx_CH1, TIMx_CH2, TIMx_CH3 and TIMx_CH4. Sliding filter is only for IOM input, not for HSE/128, LSI, HSI input.

10.3 TIM3 功能描述

10.3.1 时基单元

通用器的时基单元主要包括：预分频器、计数器和自动重装载寄存器。当时基单元工作时，软件可以随时读取和写入相应的寄存器（TIMx_PSC、TIMx_CNT 和 TIMx_AR）。

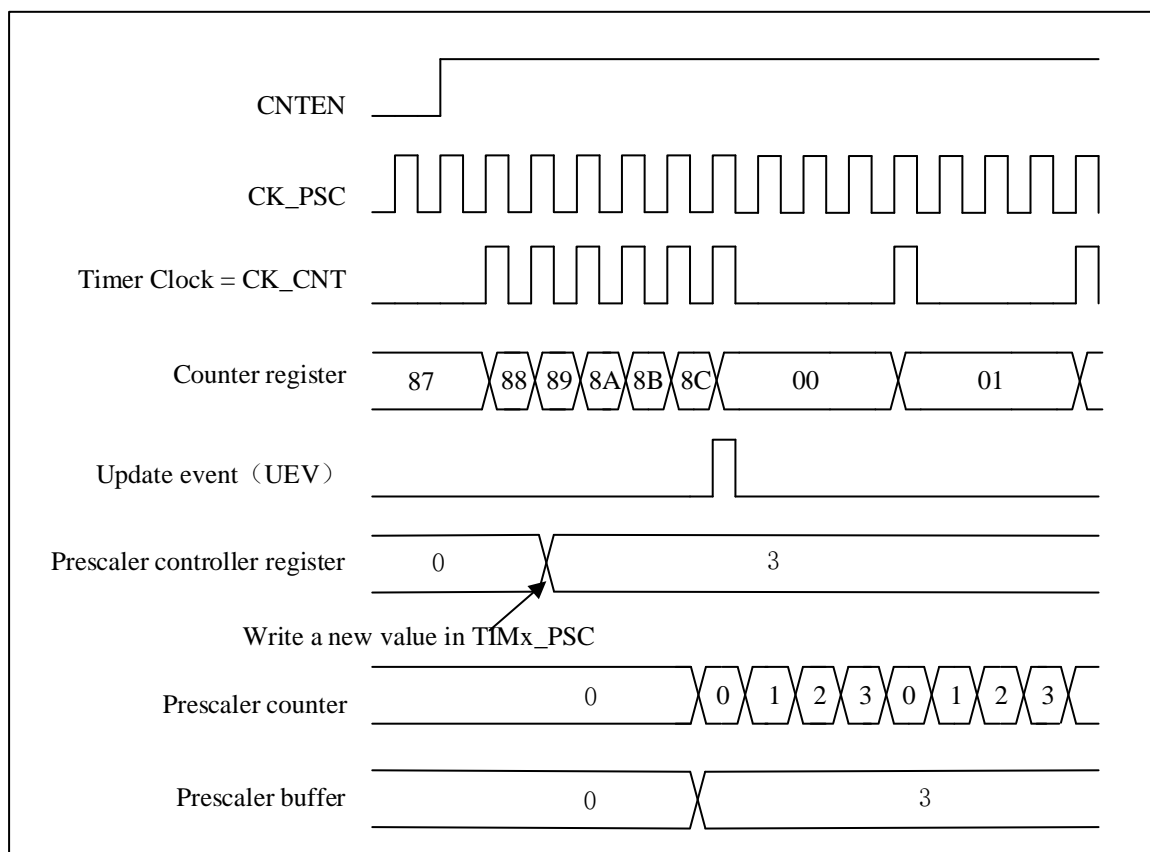
根据自动重装载预装载使能位（TIMx_CTRL1.ARPEN）的设置，预装载寄存器的值会立即或在每次更新事

件 UEV 时传输到影子寄存器。TIMx_CTRL1.UPDIS=0 时，当计数器上溢/下溢或软件设置 TIMx_EVTGEN.UDGN，将生成更新事件。计数器 CK_CNT 仅在 TIMx_CTRL1.CNTEN 位被设置时有效。计数器在 TIMx_CTRL.CNTEN 位被设置后一个时钟周期之后开始计数。

10.3.1.1 预分频器描述

TIMx_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

图 10-2 当预分频的参数从 1 到 4，计数器的时序图



10.3.2 计数器模式

10.3.2.1 向上计数模式

使用向上计数模式，计数器将从 0 计数到寄存器 TIMx_AR 的值，然后重置为 0。并产生一个计数器溢出事件。

如果设置 TIMx_CTRL1.UPRS 位（选择更新请求）和 TIMx_EVTGEN.UDGN 位，将产生一个更新事件（UEV）。但是 TIMx_STS.UDITF 不会被硬件置起，因此不会产生更新中断或 DMA 更新请求。这是为了避免清除计数器时产生更新中断。

取决于 TIMx_CTRL1.UPRS 的配置，当发生更新事件时，TIMx_STS.UDITF 被设置，所有寄存器都会更新：

- 当 TIMx_CTRL1.ARPEN = 1，预装载寄存器(TIMx_AR)的值被更新到自动装载影子寄存器
- 预加载值 (TIMx_PSC) 被重新加载到预分频器影子寄存器中

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以通过设置 `TIMx_CTRL1.UPDIS=1` 来禁止更新事件。

当产生一个更新事件时，计数器仍将被清除，预分频器计数器也将被设置为 0（但预分频器值将保持不变）。

下图给出一些示例，展示了向上计数模式计数器在不同分频因子下的动作。

图 10-3 当内部时钟分频因子 = $2/N$ 时，向上计数的时序图

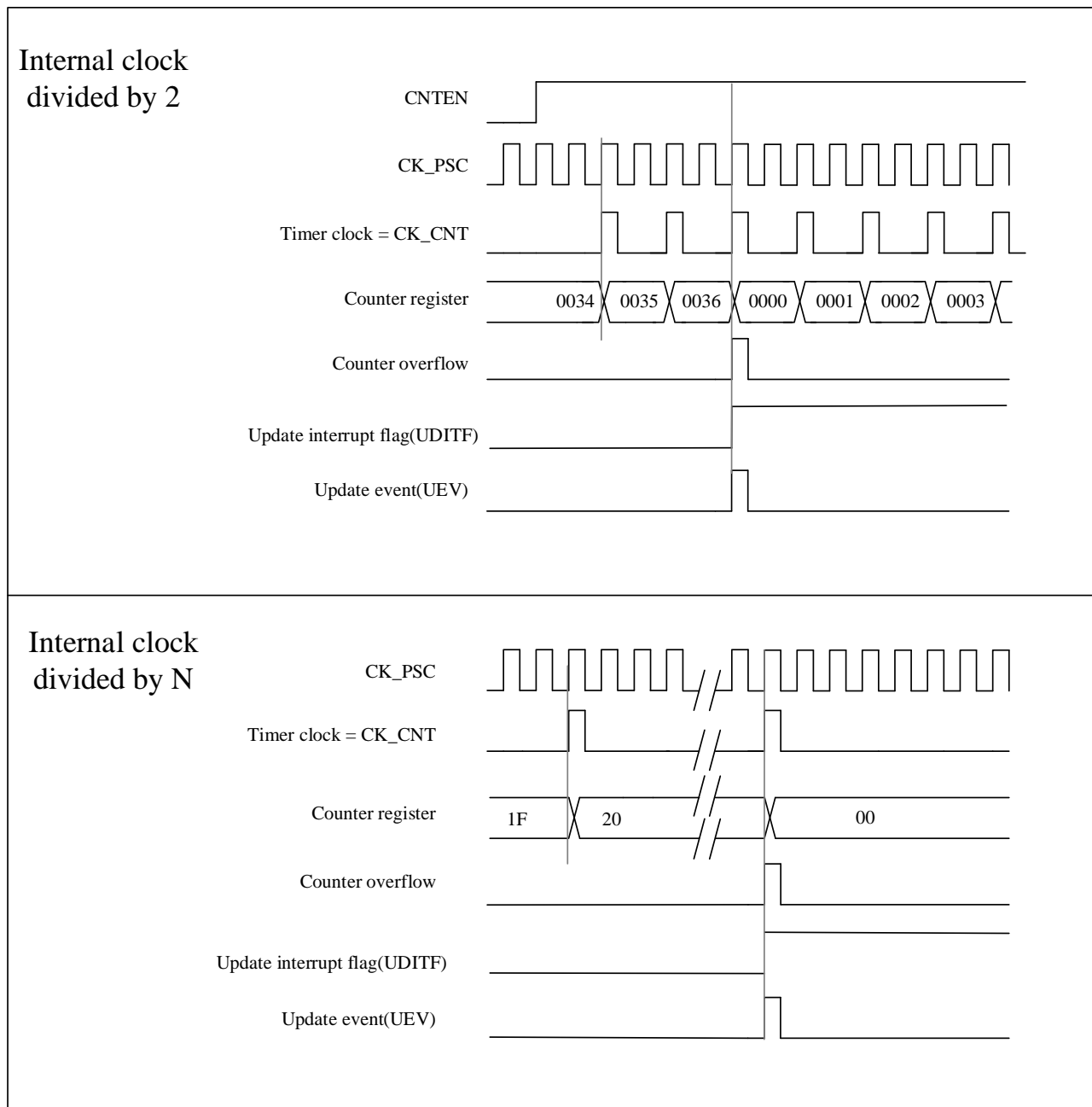
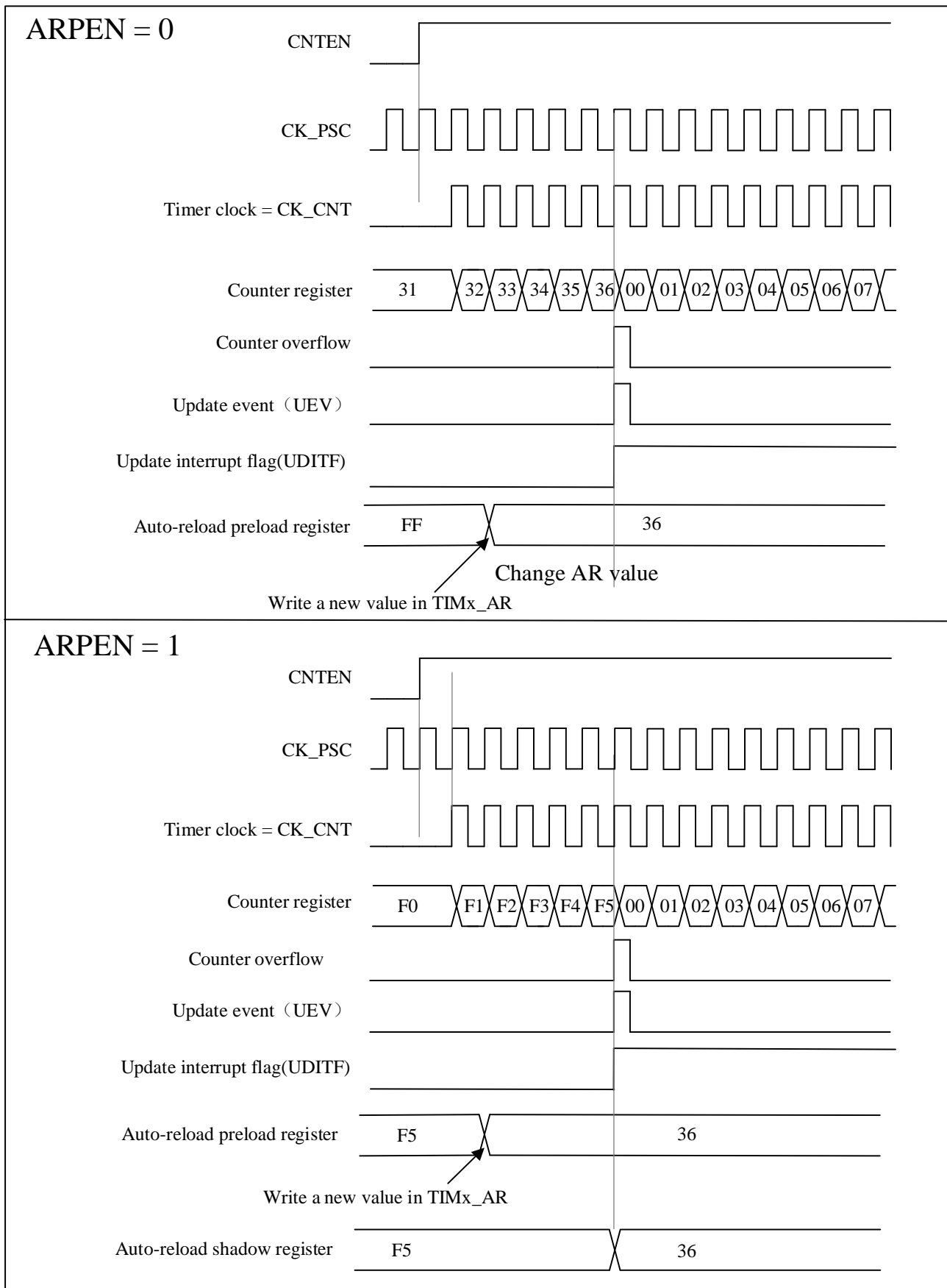


图 10-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图



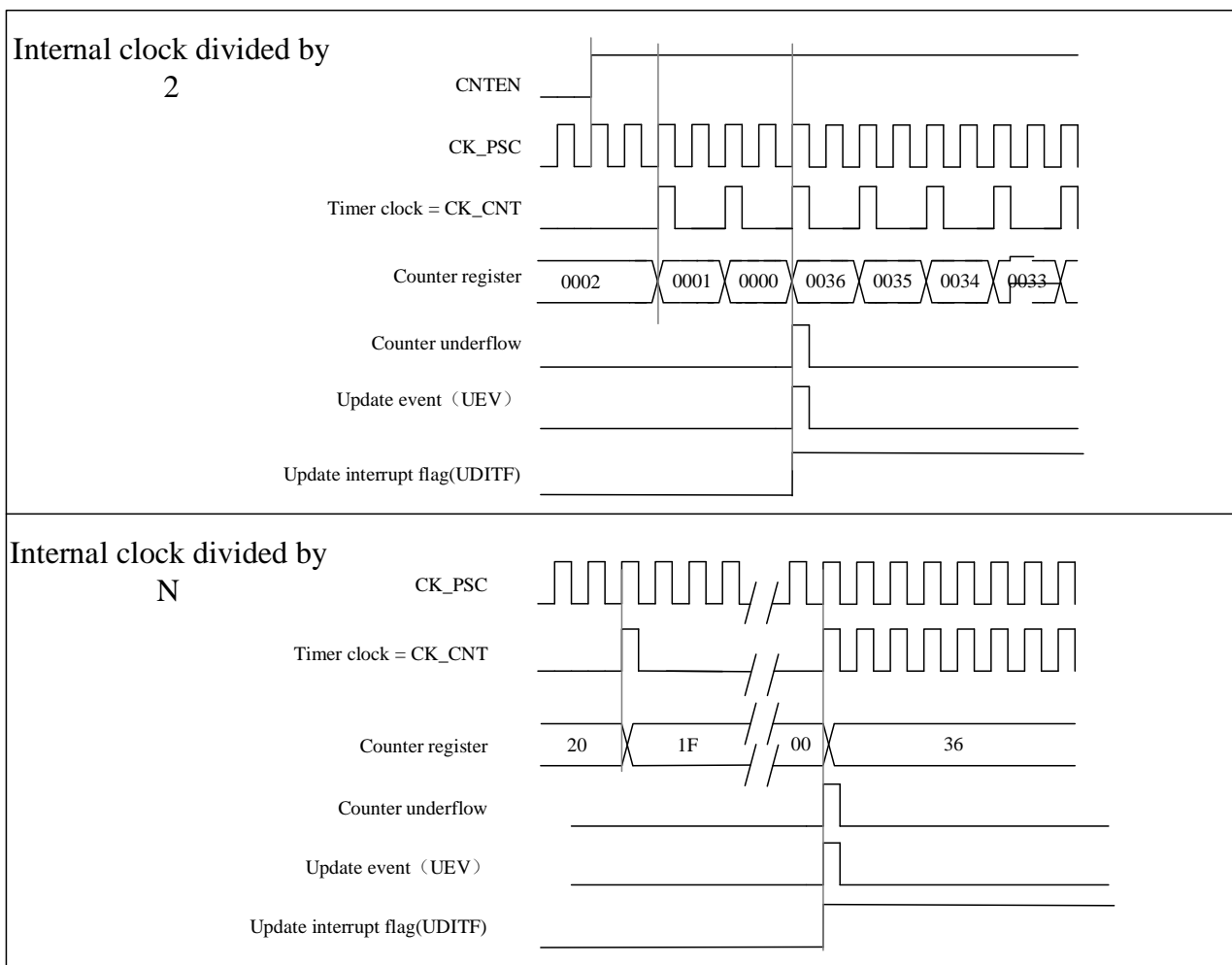
10.3.2.2 向下计数模式

向下计数模式，计数器将从寄存器 TIMx_AR 的值减至 0，然后从自动重载值重新开始，并产生计数器向下溢出事件

向下计数模式和向上计数模式配置更新事件和更新寄存器的过程相同，请查阅 10.3.2.1 章节。

下图给出一些示例，展示了向下计数模式计数器在不同分频因子下的动作。

图 10-5 内部时钟分频因子 = 2/N 时，向下计数时序图



10.3.2.3 中央对齐模式

在中央对齐模式下，计数器从 0 增加到值 (TIMx_AR) - 1，产生计数器溢出事件。然后，它从自动重载值 (TIMx_AR) 向下计数到 1，并生成一个计数器向下溢出事件。然后计数器重置为 0 并再次开始计数。

在这种模式下，TIMx_CTRL1.DIR 方向位无效，由硬件更新和指定当前计数方向。当 TIMx_CTRL1.CAMSEL 位不等于“00”时，中央对齐模式有效。

每次计数上溢和计数下溢时都会生成更新事件。或者，也可以通过设置 TIMx_EVTGEN.UDGN 位（通过软件或使用从模式控制器）来生成更新事件。在这种情况下，计数器从 0 重新开始计数，预分频器的计数器也从 0 重新开始计数。

注：如果因为计数器溢出而产生更新，自动重载将在计数器重新载入之前被更新。

图 10-6 内部时钟分频因子 = 2/N，中央对齐时序图

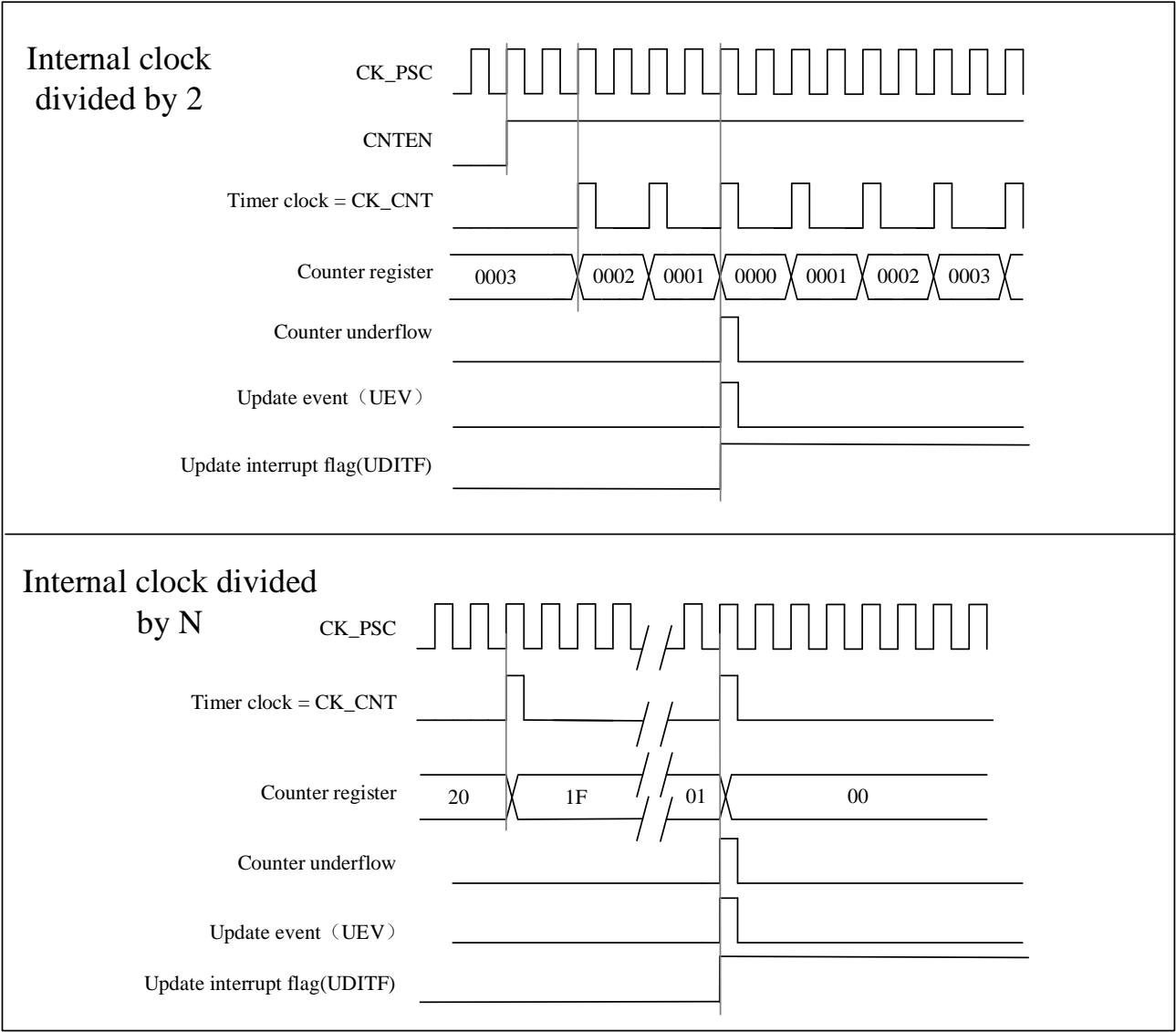
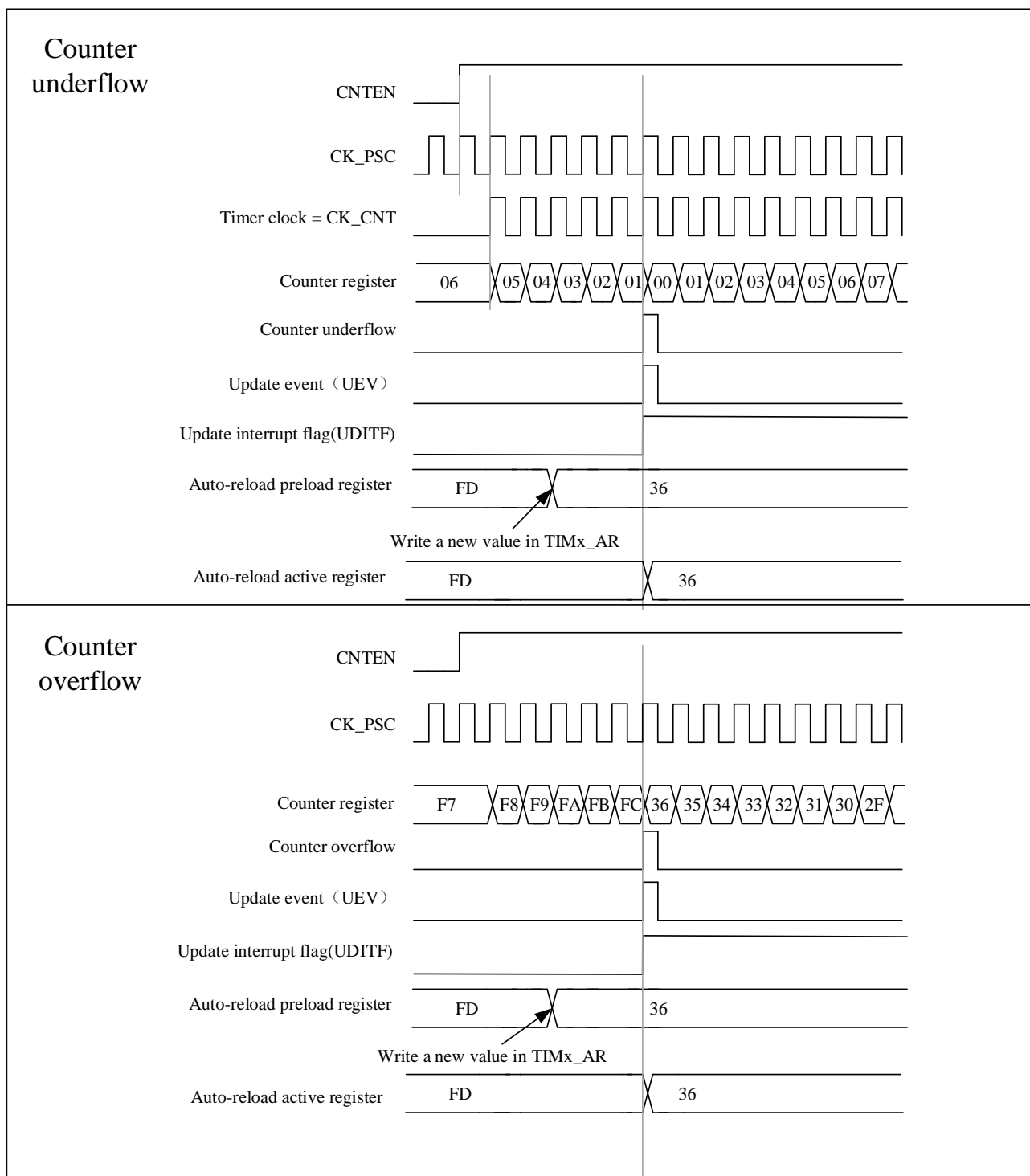


图 10-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)



10.3.3 时钟选择

- 通用定时器的内部时钟: CK_INT:
- 两种外部时钟模式:
 - 外部输入引脚

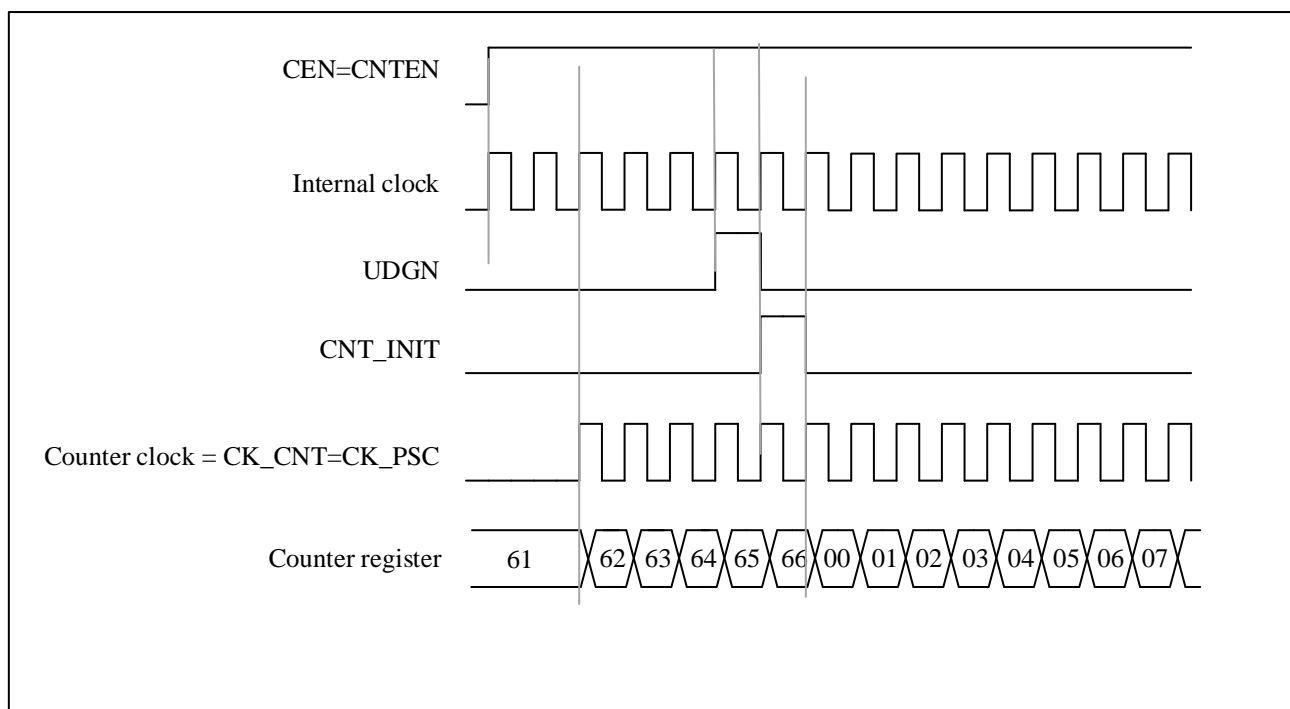
■ 外部触发输入 ETR

■ 内部触发输入 (ITR_x)：一个定时器用作另一个定时器的预分频器

10.3.3.1 内部时钟源(CK_INT)

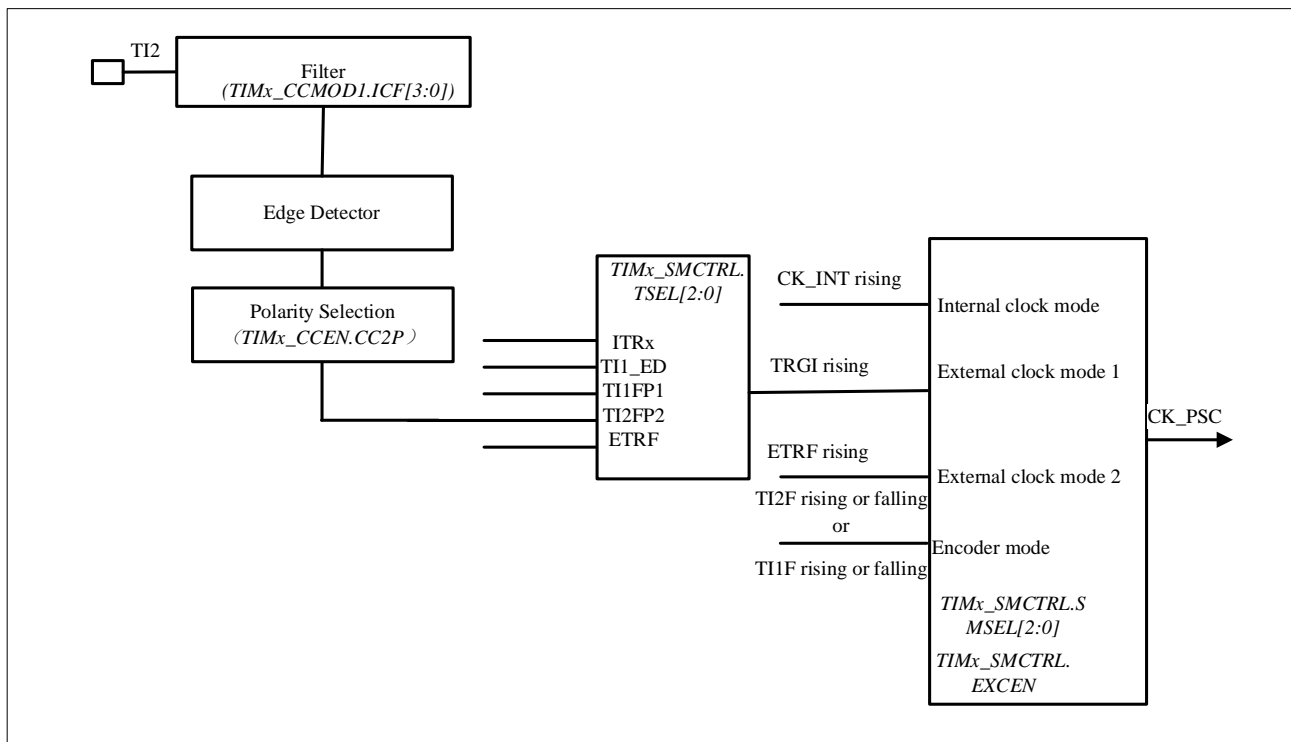
当 TIM_x_SMCTRL.SMSEL 等于“000”时，从模式控制器被禁用。这三个控制位 (TIM_x_CTRL1.CNTEN、TIM_x_CTRL1.DIR、TIM_x_EVTGEN.UDGN) 只能由软件改变 (TIM_x_EVTGEN.UDGN 除外，它保持自动清零)。前提是 TIM_x_CTRL1.CNTEN 位被软写为'1'，预分频器的时钟源由内部时钟 CK_INT 提供。

图 10-8 正常模式下的控制电路，内部时钟除以 1



10.3.3.2 外部时钟源模式 1

图 10-9 TI2 外部时钟连接示例



通过配置 `TIMx_SMCTRL.SMSEL=111` 选择该模式。计数器可以配置为在所选输入的时钟上升沿或下降沿进行计数。

例如，配置向上计数模式在 `TI2` 输入的时钟上升沿计数，配置步骤如下：

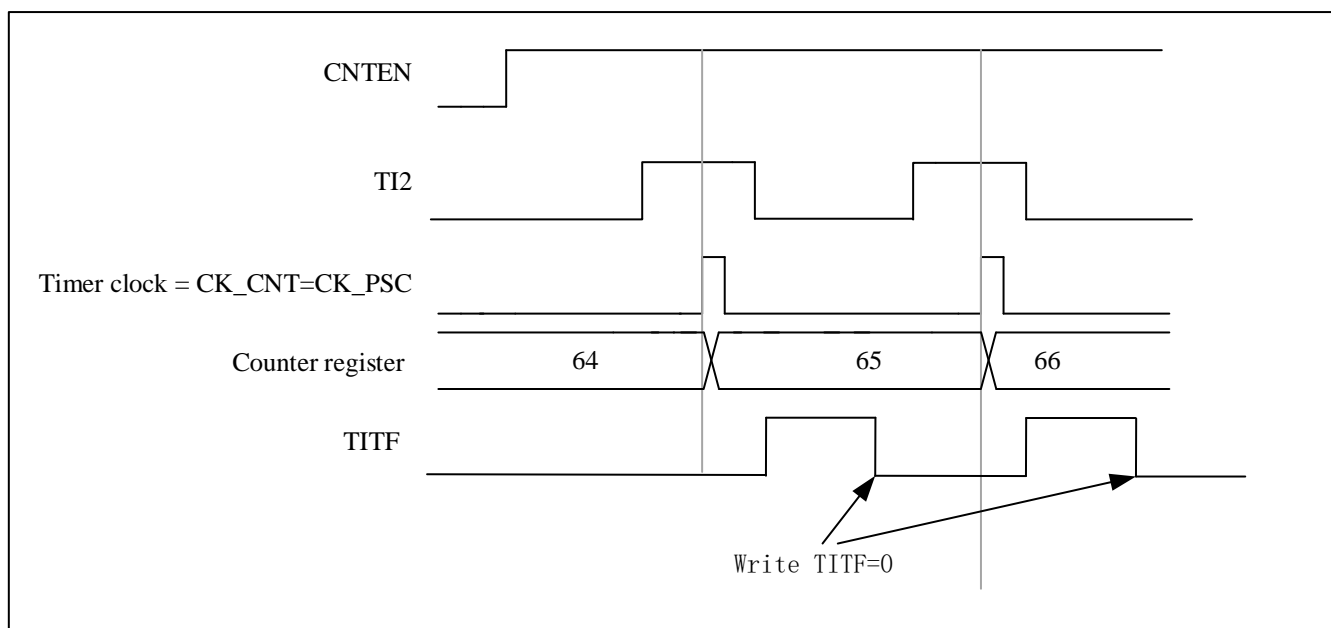
- 配置 `TIMx_CCMOD1.CC2SEL` 等于‘01’，`CC2` 通道配置为输入，`IC2` 映射到 `TI2`
- 配置 `TIMx_CCEN.CC2P` 等于‘0’，选择时钟上升沿极性
- 通过配置 `TIMx_CCMOD1.IC2F[3:0]` 选择输入滤波器带宽（如果不需要滤波器，保持 `IC2F` 位为‘0000’）
- 配置 `TIMx_SMCTRL.SMSEL` 等于‘111’，选择定时器外部时钟模式 1
- 配置 `TIMx_SMCTRL.TSEL` 等于‘110’，选择 `TI2` 作为触发输入源
- 配置 `TIMx_CTRL1.CNTEN` 等于‘1’以启动计数器

注意：捕获预分频器不用于触发，所以不需要配置

当定时器时钟的上升沿出现在 `TI2=1` 时，计数器计数一次并且 `TIMx_STS.TITF` 标志被拉高。

`TI2` 的上升沿与计数器实际时钟之间的延迟取决于 `TI2` 输入端的再同步电路。

图 10-10 外部时钟模式 1 的控制电路

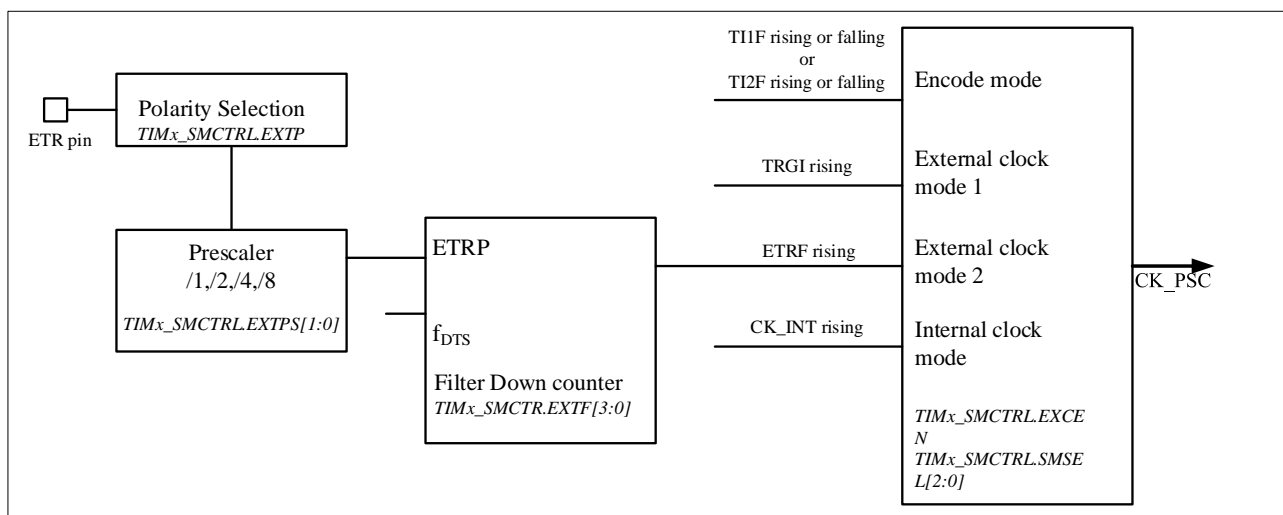


10.3.3.3 外部时钟源模式 2

此模式由 TIMx_SMCTRL.EXCEN 选择等于 1。计数器可以在外部触发输入 ETR 的每个上升沿或下降沿计数。

下图为外部时钟源模式 2 的外部触发输入模块示意图。

图 10-11 外部触发输入框图



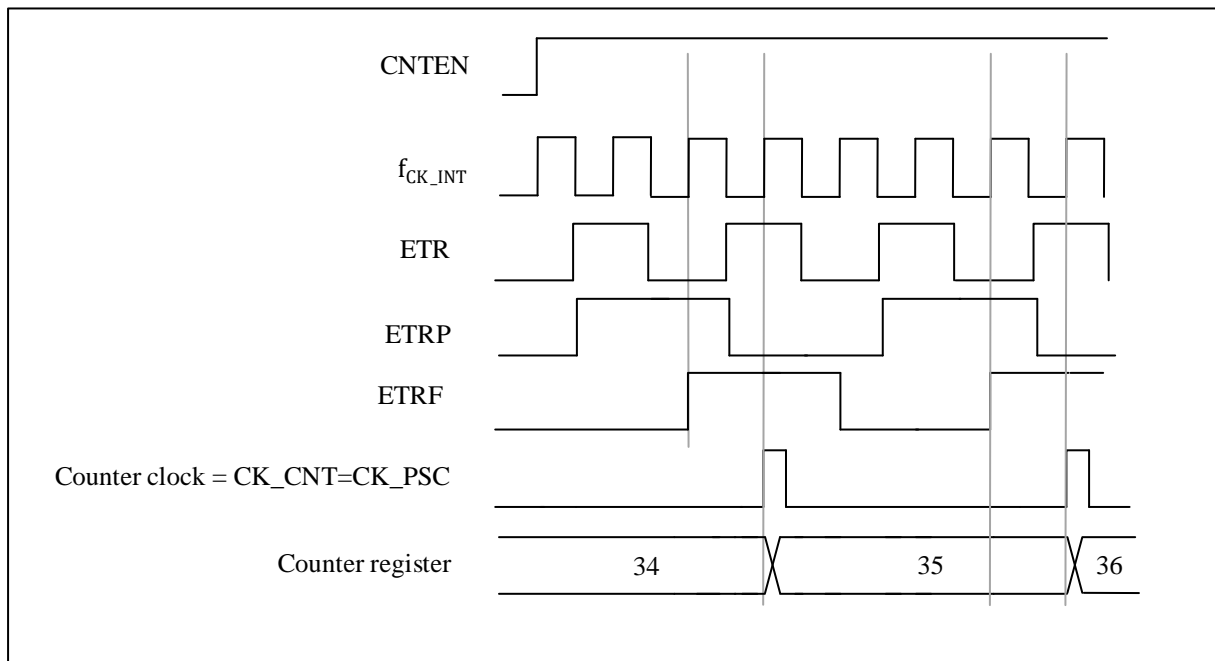
例如，使用以下配置步骤使向上计数器在 ETR 上每 2 个上升沿计数一次。

- 由于在这种情况下不需要过滤器，因此使 TIMx_SMCTRL.EXTF[3:0] 等于‘0000’
- 通过使 TIMx_SMCTRL.EXTPS[1:0] 等于 ‘01’ 来配置预分频器
- 通过设置 TIMx_SMCTRL.EXTP 等于‘0’来选择 ETR 引脚的极性，ETR 的上升沿有效

- 外部时钟模式 2 通过设置 TIMx_SMCTRL.EXCEN 等于‘1’来选择
- 通过设置 TIMx_CTRL1.CNTEN 等于“1”启动计数器。

计数器每 2 个 ETR 上升沿计数一次。ETR 的上升沿与计数器的实际时钟之间的延迟是由于 ETRP 信号上的再同步电路造成的。

图 10-12 外部时钟模式 2 的控制电路

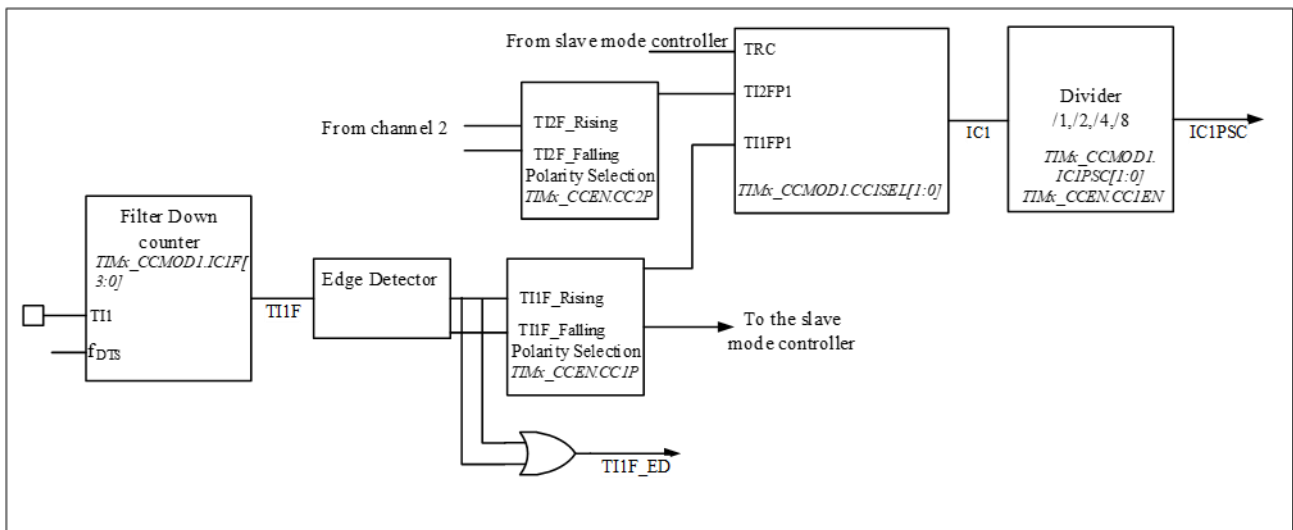


10.3.4 捕获/比较通道

捕获/比较通道包括捕获/比较寄存器和影子寄存器。输入部分由数字滤波器、多路复用器和预分频器组成。输出部分包括比较器和输出控制。

输入信号 TIx 被采样和滤波以产生信号 TIxF。然后由极性选择功能的边沿检测器生成信号 (TIxF_rising 或 TIxF_falling), 其极性由 TIMx_CCEN.CCxP 位选择。该信号可用作从模式控制器的触发输入。同时, 信号 ICx 经过分频后送入捕获寄存器。下图显示了捕获/比较通道的框图。

图 10-13 捕获/比较通道（例如：通道 1 输入级）



输出部分生成一个中间波形 OCxRef（高电平有效）作为参考。极性作用在链的末端。

图 10-14 捕获/比较通道 1 主电路

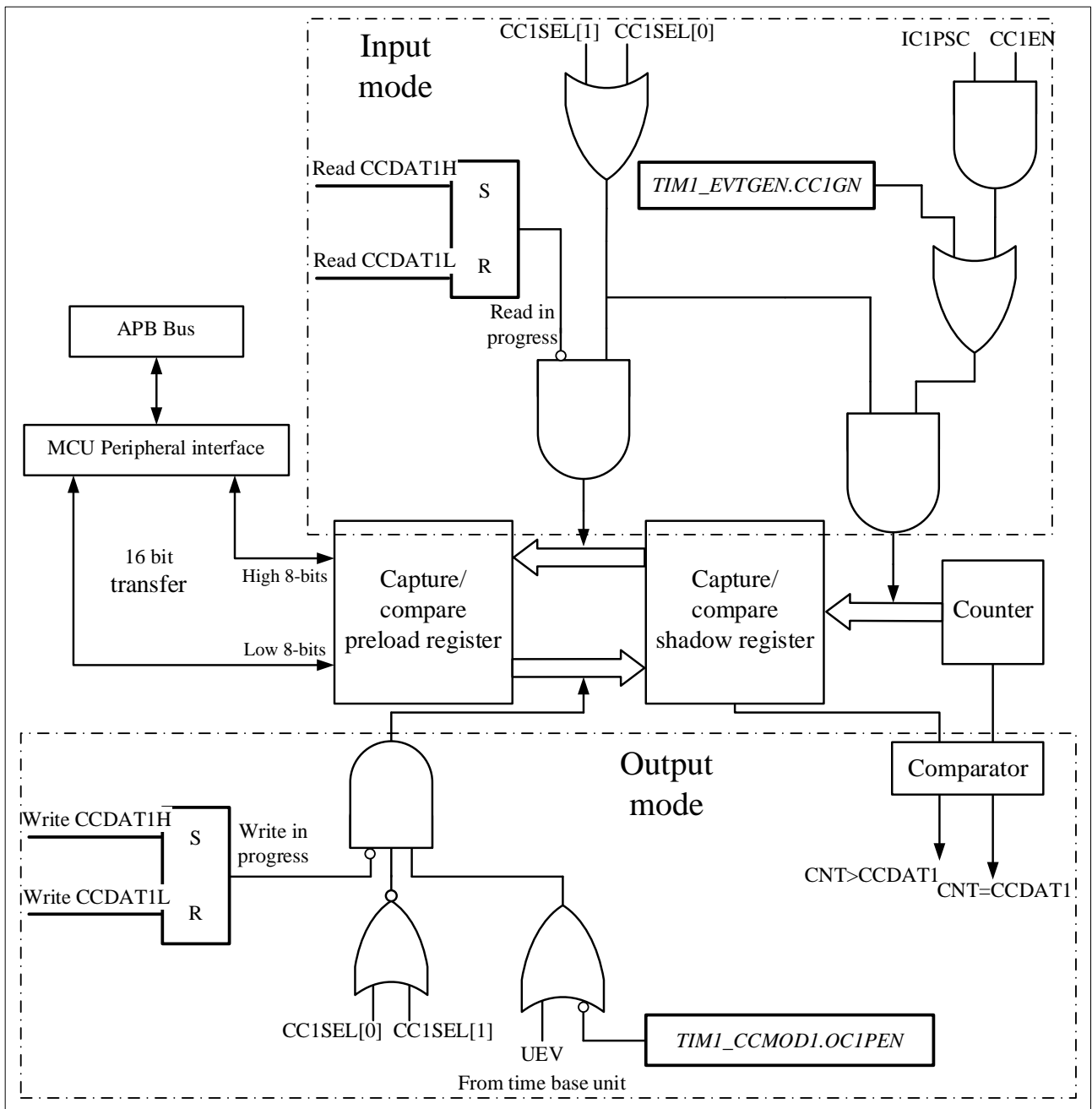
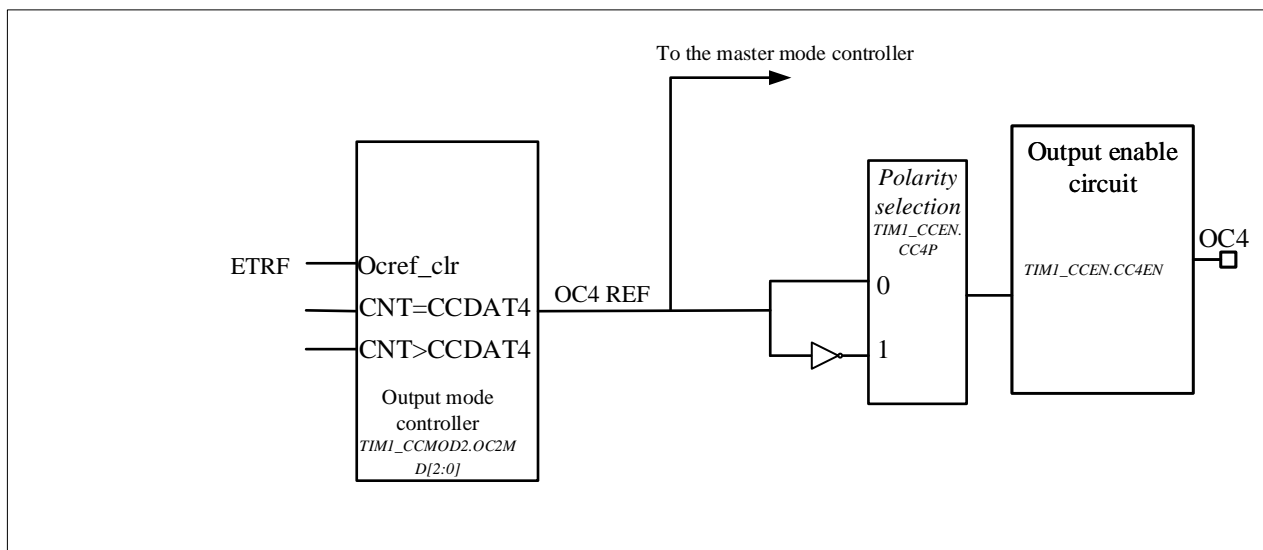


图 10-15 通道 x 的输出部分 (x = 1/2/3/4, 以通道 4 为例子)



在捕获/比较时，读取和写入始终访问预加载的寄存器。两个具体工作流程如下：

在捕获模式下，捕获实际上是在影子寄存器中完成的，然后将影子寄存器中的值复制到预加载寄存器中。

在比较模式下，与捕获模式相反，预加载寄存器的值被复制到影子寄存器中，并与计数器进行比较。

10.3.5 输入捕获模式

在捕获模式下，TIMx_CCDATx 寄存器用于在检测到 ICx 信号后锁存计数器值。

有一个捕获中断标志 TIMx_STS.CCxITF，如果相应的中断使能被拉高，它可以发出中断或 DMA 请求。

TIMx_STS.CCxITF 位在发生捕获事件时由硬件设置，并由软件或读取 TIMx_CCDATx 寄存器清零。

当 TIMx_CCDATx 寄存器中的计数器值被捕获并且 TIMx_STS.CCxITF 已经被拉高时，重复捕获标志 TIMx_STS.CCxOCF 设置为 1。与前者不同，TIMx_STS.CCxOCF 通过向其写入 0 来清除。

为实现 TI1 输入的上升沿将计数器值捕获到 TIMx_CCDAT1 寄存器中，配置流程如下：

■ 选择有效输入：

将 TIMx_CCMOD1.CC1SEL 配置为“01”。此时输入为 CC1 通道，IC1 映射到 TI1。

■ 编程所需的输入滤波器持续时间：

通过配置 TIMx_CCMODx.ICxPSC 来定义 TI1 输入的采样频率和数字滤波器的长度。示例：如果输入信号抖动多达 5 个内部时钟周期，我们必须选择比这 5 个时钟周期更长的滤波器持续时间。当检测到具有新电平的 8 个连续样本（以 f_{DTS} 频率采样）时，我们可以验证 TI1 上的转换。然后配置 TIMx_CCMOD1.IC1F 到“0011”

■ 通过配置 TIMx_CCEN.CC1P=0，选择上升沿作为 TI1 通道的有效跳变极性

■ 配置输入预分频器。在本例中，配置 TIMx_CCMOD1.IC1PSC=‘00’ 以禁用预分频器，因为我们想要捕获每个有效转换

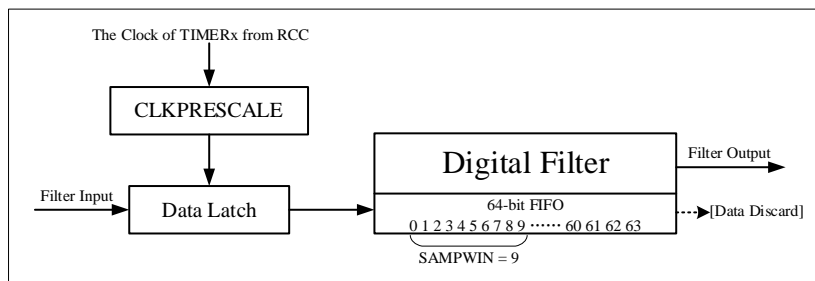
■ 通过配置 TIMx_CCEN.CC1EN=‘1’ 启用捕获。

如果要使能 DMA 请求，可以配置 TIMx_DINTEN.CC1DEN=1。如果要使能相关中断请求，可以配置 TIMx_DINTEN.CC1IEN=1。

10.3.5.1 通道输入滤波

寄存器 TIMx_CxFLT(x = 1, 2, 3, 4) 描述如下：

图 10-16 滑动滤波



- 数字滤波器通过 RCC 的 TIMx 时钟采样通道输入信号，在 64 位 FIFO 中累积采样。仅在 TIMx_CxFLT.WSIZE [5:0] 中定义的窗口大小内采样数据，最大大小为 64。
- 过滤器输出采样窗口内的多数值，该值由 TIMx_CxFLT.THRESH [5:0] 中的阈值定义，最大阈值为 63。此值应等于或大于窗口大小的一半。如果采样窗口内的逻辑 1 和逻辑 0 计数均不大于阈值，则数字滤波器保持先前的输出值。
- TIMx_CxFLT.PSC 寄存器决定相应数字滤波器的采样率。过滤器 FIFO 在每个采样时钟从输入中捕获一个采样值。
- 如果数字滤波器关闭，滤波器输入将像电线一样绕过输出。

10.3.6 PWM 输入模式

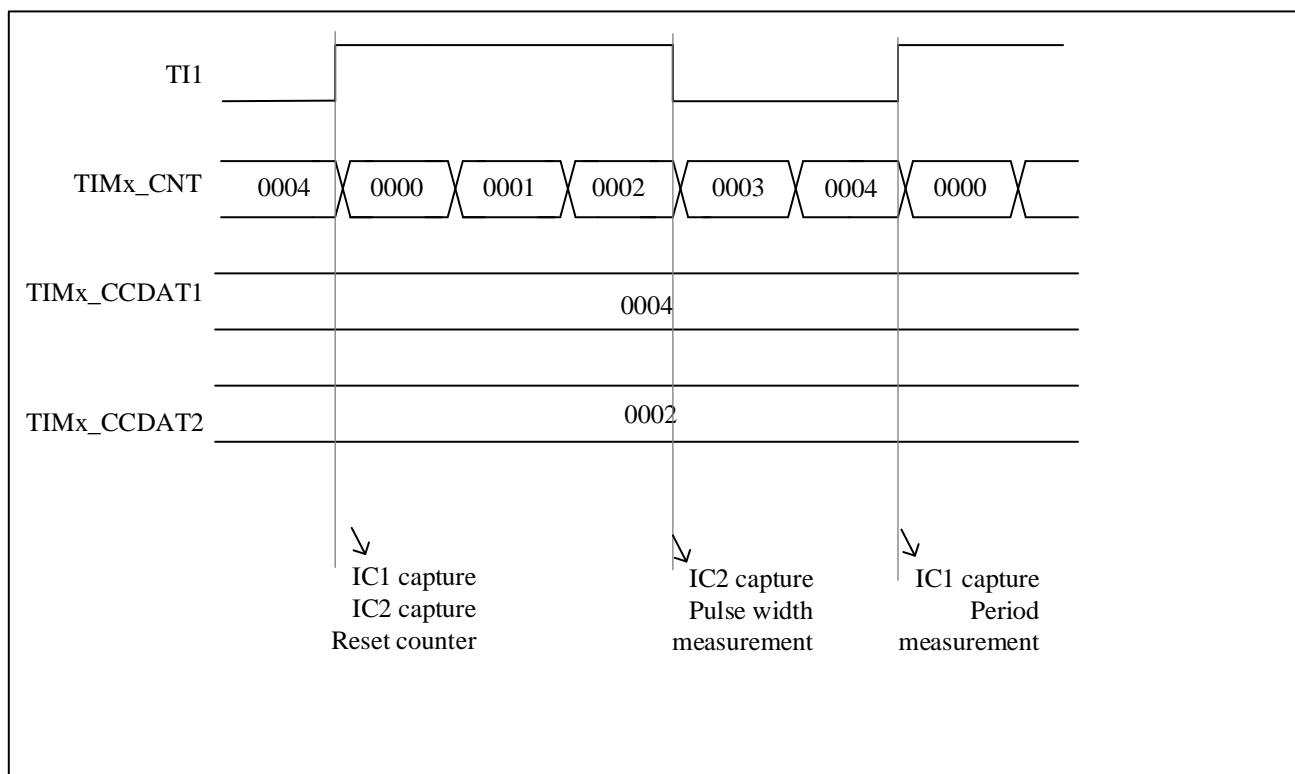
PWM 输入模式和普通输入捕获模式有一些区别，包括：

- 两个 ICx 信号映射到同一个 TIx 输入
- 两个 ICx 信号在极性相反的边沿有效
- 选择两个 TIxFP 信号之一作为触发输入
- 从机模式控制器配置为复位模式

例如，下面的配置流程可以用来知道 TI1 上 PWM 信号的周期和占空比（这取决于 CK_INT 的频率和预分频器的值）。

- 配置 TIMx_CCMOD1.CC1SEL 等于 ‘01’ 以选择 TI1 作为 TIMx_CCDAT1 的有效输入
- 配置 TIMx_CCEN.CC1P 等于 ‘0’ 选择滤波定时器输入 1(TI1FP1) 的有效极性，在上升沿有效
- 配置 TIMx_CCMOD1.CC2SEL 等于 ‘10’ 选择 TI1 作为 TIMx_CCDAT2 的有效输入
- 配置 TIMx_CCEN.CC2P 等于 1 选择滤波定时器输入 2(TI1FP2) 的有效极性，下降沿有效
- 配置 TIMx_SMCTRL.TSEL=101 选择 Filtered timer input 1 (TI1FP1) 作为有效触发输入
- 配置 TIMx_SMCTRL.SMSEL=100 配置从模式控制器为复位模式
- 配置 TIMx_CCEN.CC1EN=1 和 TIMx_CCEN.CC2EN=1 以启用捕获

图 10-17 PWM 输入模式时序



由于只有滤波器定时器输入 1 (TI1FP1) 和滤波器定时器输入 2 (TI2FP2) 连接到从模式控制器，因此 PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号一起使用。

10.3.7 强制输出模式

在输出模式 (TIMx_CCMODx.CCxSEL=00) 下，软件可以直接将输出比较信号强制为有效或无效电平。

用户可以设置 TIMx_CCMODx.OCxMD=101 强制输出比较信号为有效电平。OCxREF 将被强制为高电平，OCx 得到与 CCxP 极性位相反的值。另一方面，用户可以设置 TIMx_CCMODx.OCxMD=100 强制输出比较信号为无效电平，即 OCxREF 被强制为低电平。

在此模式下，TIMx_CC DATx 影子寄存器和计数器的值仍然相互比较。

输出比较寄存器 TIMx_CC DATx 和计数器 TIMx_CNT 之间的比较对 OCxREF 没有影响。并且仍然可以设置标志。因此，仍然可以发送中断和 DMA 请求。

10.3.8 输出比较模式

用户可以使用此模式来控制输出波形，或指示一段时间已过。

当捕获/比较寄存器和计数器的值相同时，输出比较函数的操作如下：

- TIMx_CCMODx.OCxMD 为输出比较模式，TIMx_CCEN.CCxP 为输出极性。当比较匹配时，如果设置 TIMx_CCMODx.OCxMD=000，则输出管脚将保持其电平；如果设置 TIMx_CCMODx.OCxMD=001，则设置输出管脚有效；如果设置 TIMx_CCMODx.OCxMD=010，则输出管脚将为 设置为无效；如果设置 TIMx_CCMODx.OCxMD=011，则输出引脚将设置为翻转。
- 设置 TIMx_STS.CCxITF

- 如果用户设置了 TIMx_DINTEN.CCxIEN，将产生相应的中断
- 如果用户设置 TIMx_DINTEN.CCxDEN 并设置 TIMx_CTRL2.CCDSEL 选择 DMA 请求，将发送 DMA 请求

用户可以设置 TIMx_CCMODx.OCxPEN 来选择是否使用捕获/比较预加载寄存器 (TIMx_CCxATx) 来选择捕获/比较影子寄存器。

时间分辨率是计数器的一个计数周期。

在单脉冲模式下，输出比较模式也可用于输出单脉冲。

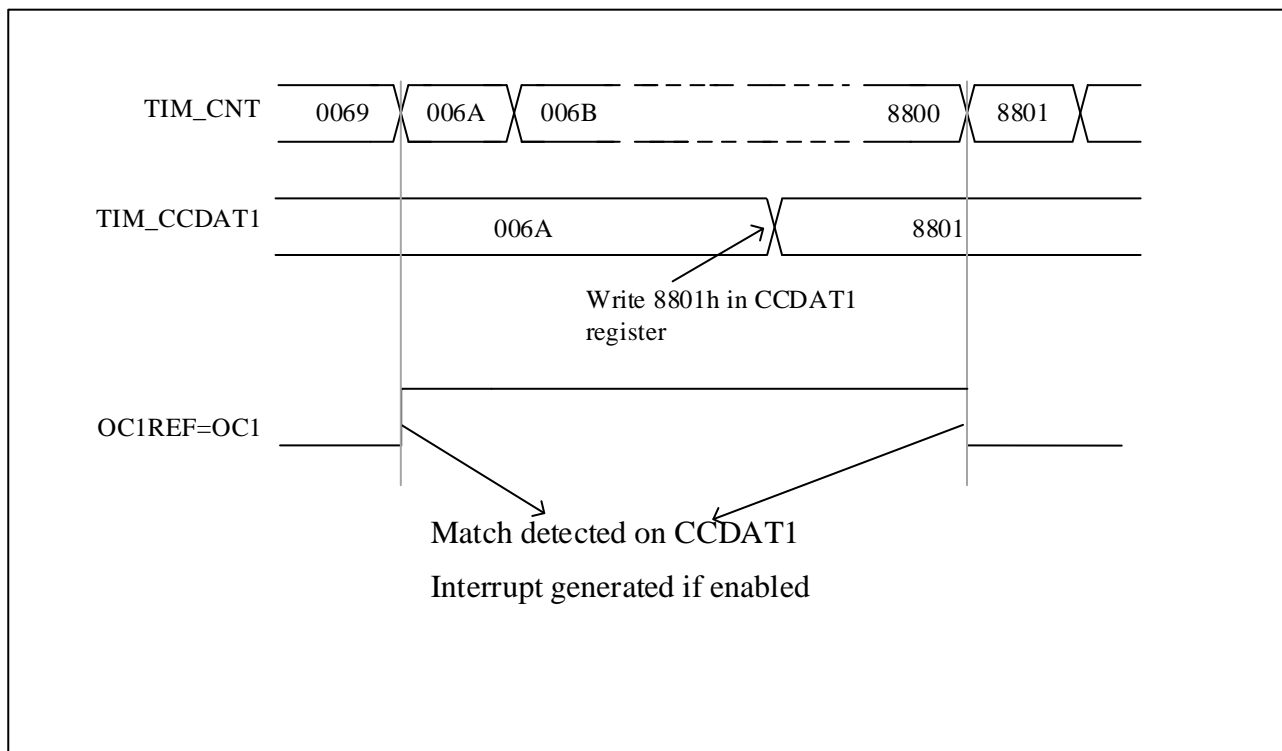
以下是输出比较模式的配置步骤：

- 首先，用户应该选择计数器时钟
- 其次，用所需数据设置 TIMx_AR 和 TIMx_CCxATx
- 如果用户需要产生中断，设置 TIMx_DINTEN.CCxIEN
- 然后通过设置 TIMx_CCEN.CCxP、TIMx_CCMODx.OCxMD、TIMx_CCEN.CCxEN 等选择输出模式
- 最后，设置 TIMx_CTRL1.CNTEN 启用计数器

用户可以随时通过设置 TIMx_CCxATx 来更新输出波形，只要不启用预加载寄存器。否则，TIMx_CCxATx 影子寄存器将在下一次更新事件中更新。

例如：

图 10-18 输出比较模式，开启 OC1



10.3.9 PWM 模式

用户可以使用 PWM 模式产生一个信号，其占空比由 TIMx_CCxDATx 寄存器的值决定，其频率由 TIMx_AR 寄存器的值决定。并且取决于 TIMx_CTRL1.CAMSEL 的值，TIM 可以在边沿对齐模式或中央对齐模式下产生 PWM 信号。

用户可以通过设置 TIMx_CCMODx.OCxMD=110 或设置 TIMx_CCMODx.OCxMD=111 来设置 PWM 模式 1 或 PWM 模式 2。要使用预加载寄存器，用户必须设置相应的 TIMx_CCMODx.OCxPEN。然后设置 TIMx_CTRL1.ARPEN 自动重载预加载寄存器。

用户可以通过设置 TIMx_CCEN.CCxP 来设置 OCx 的极性。

当 TIM 处于 PWM 模式时，TIMx_CNT 和 TIMx_CCxDATx 的值总是相互比较。

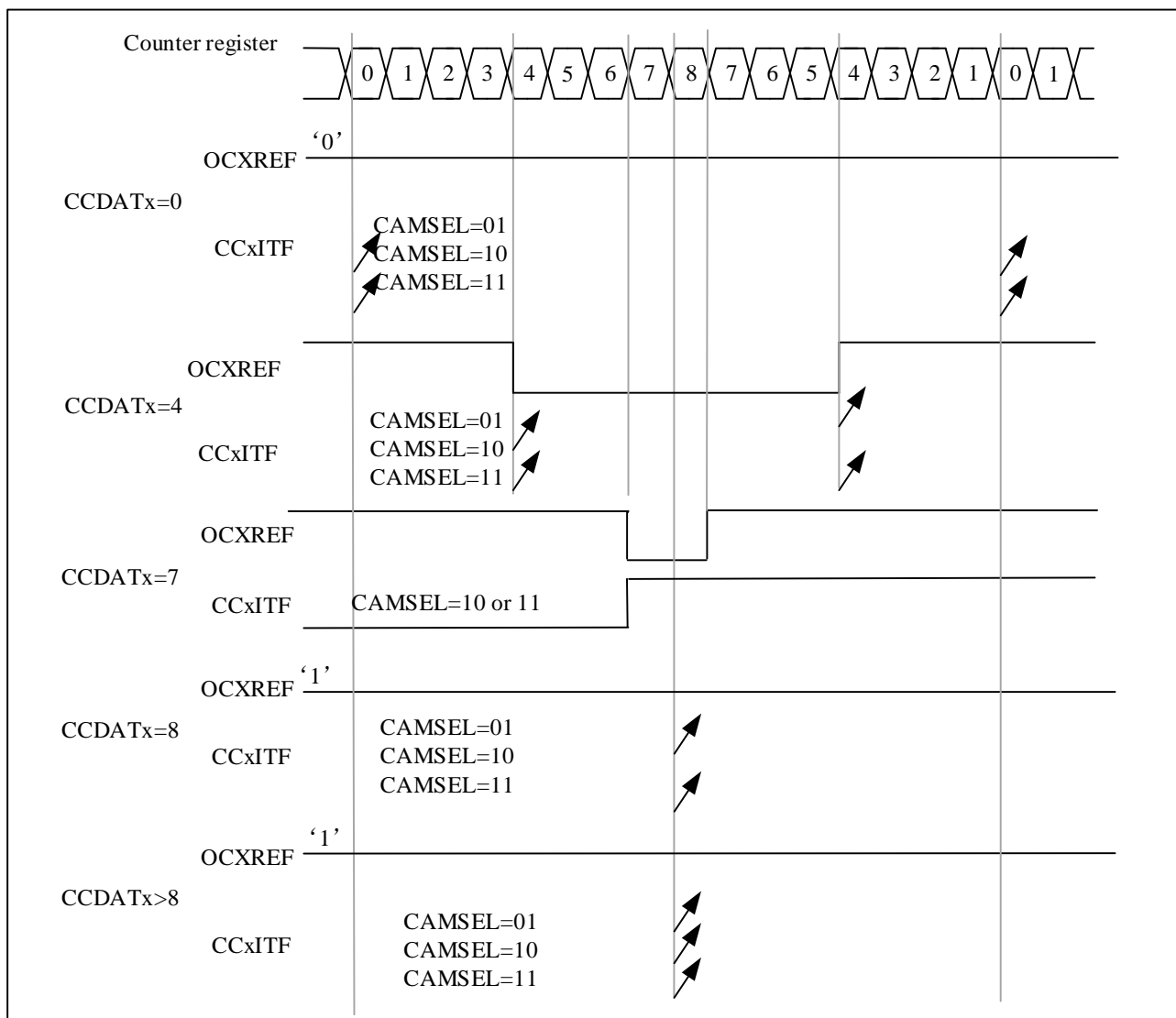
只有当更新事件发生时，预加载寄存器才会转移到影子寄存器。因此，用户必须在计数器开始计数之前通过设置 TIMx_EVTGEN.UDGN 来复位所有寄存器。

10.3.9.1 PWM 中央对齐模式

如果用户设置 TIMx_CTRL1.CAMSEL 等于 01、10 或 11，PWM 中央对齐模式将被激活。比较标志的设置取决于 TIMx_CTRL1.CAMSEL 的值。设置比较标志的情况有 3 种，仅当计数器向上计数时，仅当计数器向下计数时，或当计数器向上计数和向下计数时。用户不应通过软件修改 TIMx_CTRL1.DIR，它是由硬件更新的。

中央对齐 PWM 波形示例如下，波形设置为：TIMx_AR=8，PWM 模式 1，当计数器向下计数对应 TIMx_CTRL1.CAMSEL=01 时设置比较标志。

图 10-19 中央对齐的 PWM 波形 (AR=8)



使用中央对齐模式时用户应注意的事项如下：

- 计数器向上或向下计数取决于 `TIMx_CTRL1.DIR` 的值。 注意不要同时更改 `DIR` 和 `CAMSEL` 位
- 用户在中央对齐模式下不要写计数器，否则会导致意想不到的结果。 例如：
 - ◆ 如果写入计数器的值为 0 或者是 `TIMx_AR` 的值，则方向会被更新，但不会产生更新事件
 - ◆ 如果写入计数器的值大于自动重载的值，则方向不会更新
- 为了安全起见，建议用户在启动计数器之前设置 `TIMx_EVTGEN.UDGN` 以通过软件生成更新，并且在计数器运行时不要写入计数器

10.3.9.2 PWM 边沿对齐模式

边沿对齐模式有两种配置，向上计数和向下计数。

● 向上计数

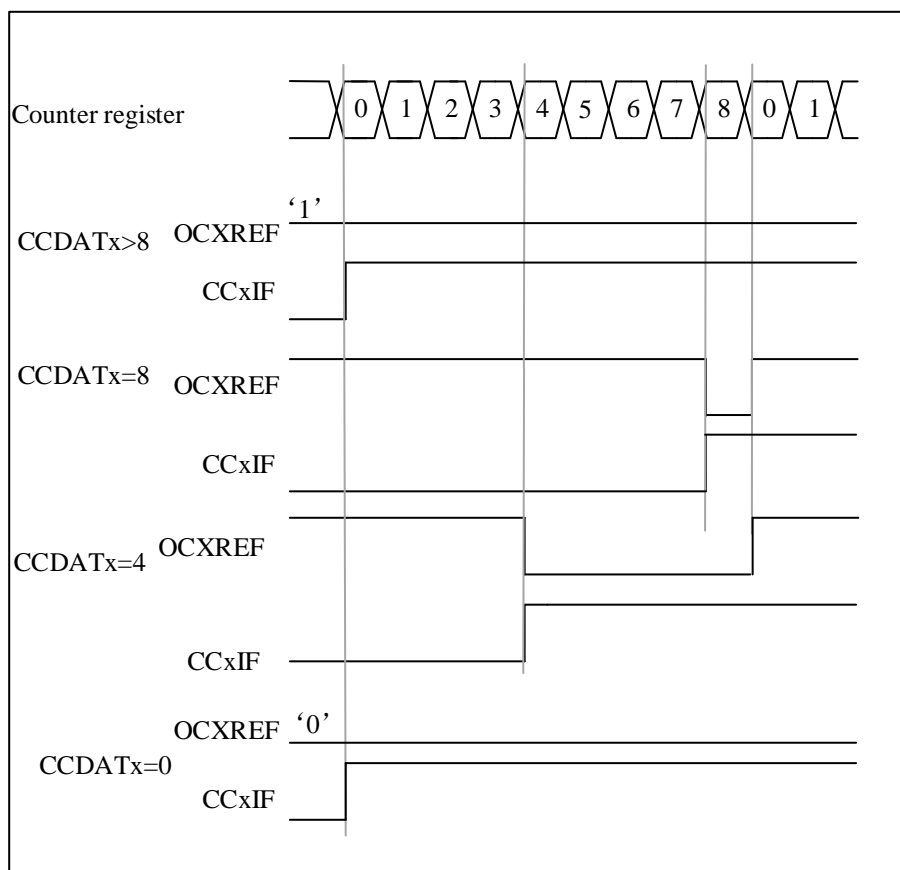
用户可以设置 `TIMx_CTRL1.DIR=0` 使计数器向上计数。

PWM 模式 1 的示例：

当 $TIMx_CNT < TIMx_CCDATx$ 时， $OCxREF$ 为高电平，否则为低电平。如果 $TIMx_CCDATx$ 中的比较值大于自动重载值，则 $OCxREF$ 将保持为 1。相反，如果比较值为 0，则 $OCxREF$ 将保持为 0。

当 $TIMx_AR=8$ 时，PWM 波形如下：

图 10-20 边沿对齐 PWM 波形 (AR=8)



● 向下计数

用户可以设置 $TIMx_CTRL1.DIR=1$ 使计数器向下计数。

PWM 模式 1 的示例：

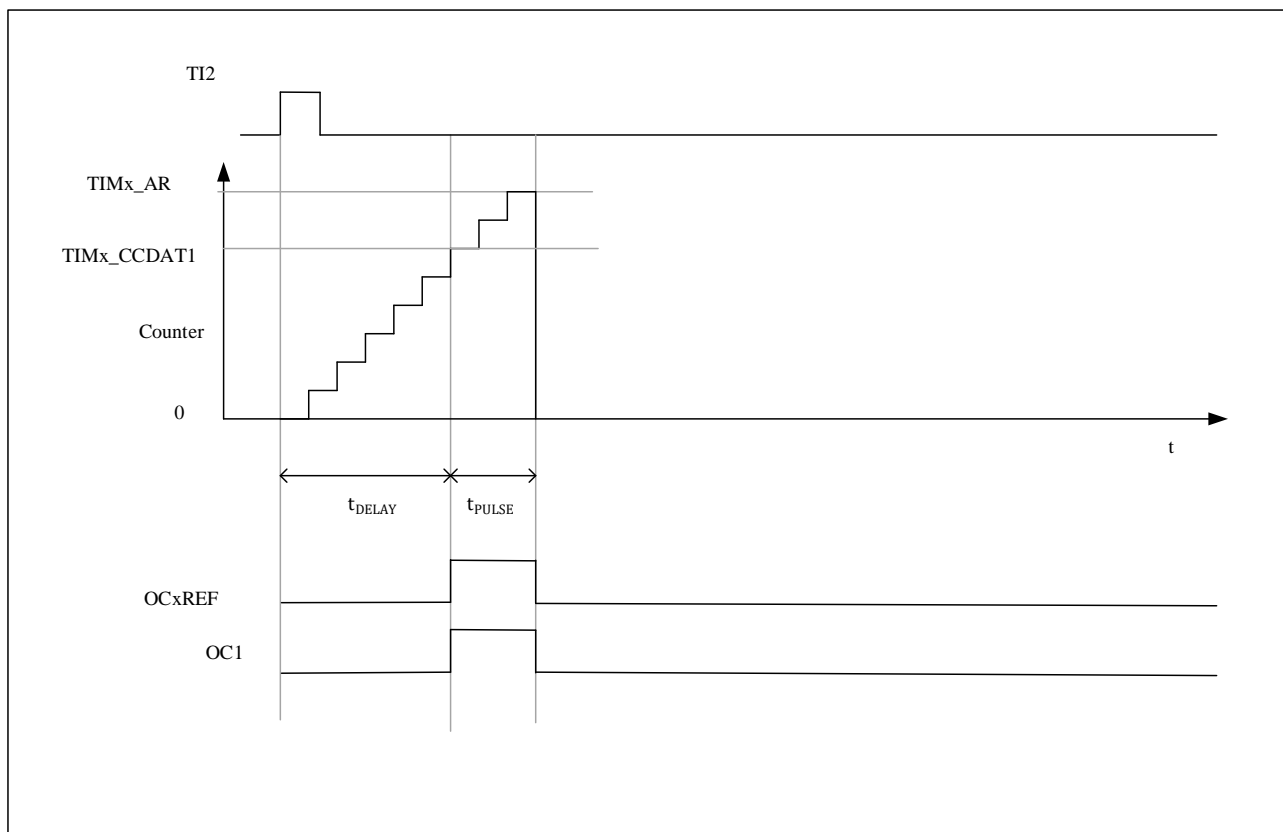
当 $TIMx_CNT > TIMx_CCDATx$ 时， $OCxREF$ 为低电平，否则为高电平。如果 $TIMx_CCDATx$ 中的比较值大于自动重载值，则 $OCxREF$ 将保持为 1。

注：若第 n 个 PWM 周期 $CCDATx$ 影子寄存器 $\geq AR$ 值，第 $n+1$ 个 PWM 周期 $CCDATx$ 的影子寄存器值是 0。在第 $n+1$ 个 PWM 周期的计数器为 0 的时刻，虽然计数器 = $CCDATx$ 影子寄存器的值 = 0， $OCxREF = '0'$ ，但不会产生比较事件。

10.3.10 单脉冲模式

在单脉冲模式(ONEPM)中，接收到触发信号，经过可控延迟 t_{DELAY} 后产生脉宽可控的脉冲 t_{PULSE} 。输出模式需要配置为输出比较模式或 PWM 模式。选择单脉冲模式后，计数器会在更新事件 UEV 产生后停止计数。

图 10-21 单脉冲模式示例



以下是单脉冲模式的示例：

从 TI2 输入检测到上升沿触发，延迟 t_{DELAY} 后在 OC1 上产生宽度为 t_{PULSE} 的脉冲。

1. 计数器配置：向上计数，计数器 $\text{TIMx_CNT} < \text{TIMx_CCDAT1} \leq \text{TIMx_AR}$ ；
2. TI2FP2 映射到 TI2, $\text{TIMx_CCMOD1.CC2SEL} = '01'$ ；TI2FP2 配置为上升沿检测, $\text{TIMx_CCEN.CC2P} = '0'$ ；
3. TI2FP2 充当从模式控制器的触发器（TRGI）并启动计数器， $\text{TIMx_SMCTRL.TSEL} = '110'$ ， $\text{TIMx_SMCTRL.SMSEL} = '110'$ （触发模式）；
4. TIMx_CCDAT1 写入要延迟的计数值（ t_{DELAY} ）， $\text{TIMx_AR} - \text{TIMx_CCDAT1}$ 为脉宽 t_{PULSE} 的计数值；
5. 配置 $\text{TIMx_CTRL1.ONEPM} = 1$ 使能单脉冲模式，配置 $\text{TIMx_CCMOD1.OC1MD} = '111'$ 选择 PWM2 模式；
6. 等待 TI2 有外部触发事件，OC1 输出一个单脉冲波形；

10.3.10.1 特殊情况：OCx 快速使能：

在单脉冲模式下，通过 TIx 输入检测到一个边沿，并触发计数器开始计数到比较值，然后输出一个脉冲。这些操作限制了可以达到的最小延迟 t_{DELAY} 。

您可以设置 $\text{TIMx_CCMODx.OCxFEN} = 1$ 开启 OCx 快速使能，在触发上升沿后，OCxREF 信号将被强制转换为与比较匹配立即发生的电平相同的电平，而不管比较结果如何。OCxFEN 快速使能仅在通道模式配置为 PWM1 和 PWM2 模式时生效。

10.3.11 在外部事件上清除 OCxREF 信号

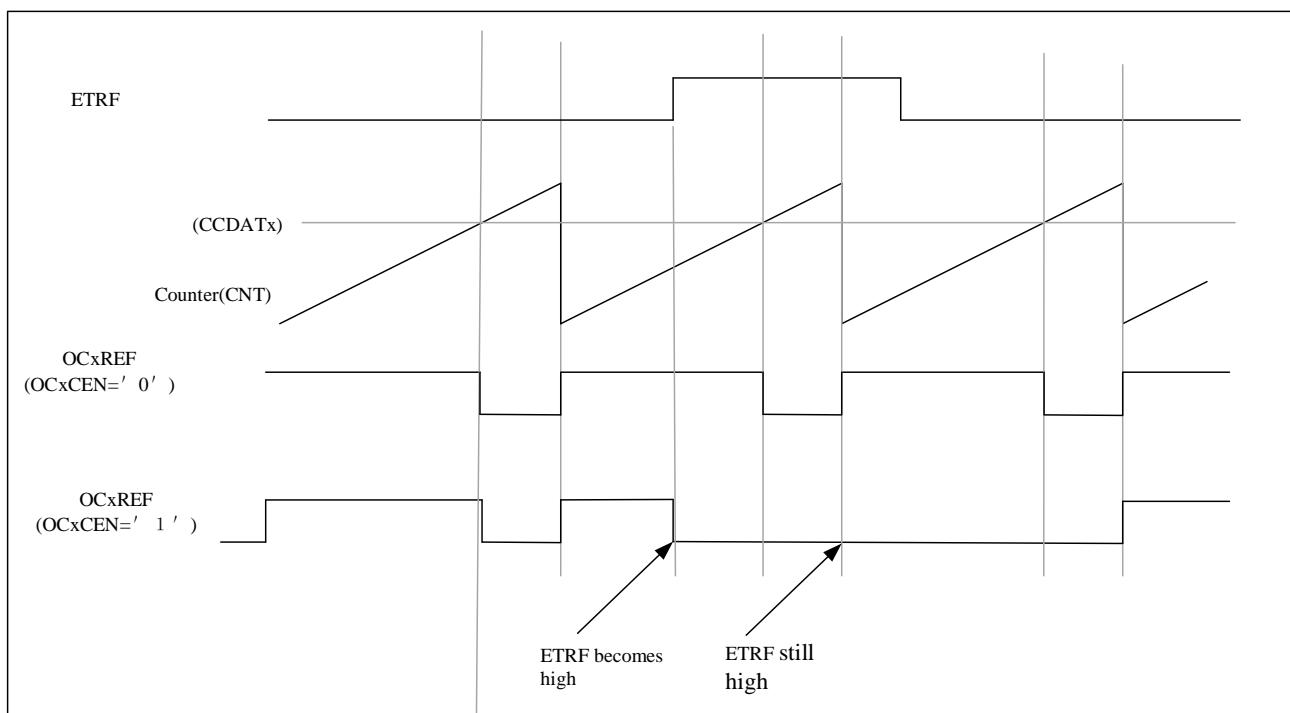
如果用户设置 $TIMx_CCMODx.OCxCEN=1$ ，ETRF 输入的高电平可用于驱动 OCxREF 信号为低电平，OCxREF 信号将保持低电平，直到下一次 UEV 发生。只有输出比较和 PWM 模式可以使用该功能。在强制模式下不能使用。

例如：为了控制电流，用户可以将 ETR 信号连接到比较器的输出端，ETR 的操作如下：

- 设置 $TIMx_SMCTRL.EXTPS=00$ 禁用外部触发预分频器。
- 设置 $TIMx_SMCTRL.EXCEN=0$ 禁用外部时钟模式 2。
- 设置 $TIMx_SMCTRL.EXTP$ 和 $TIMx_SMCTRL.EXTF$ ，根据需要配置外触发极性和外触发滤波器。

例：当 ETRF 输入变高时，OCxREF 信号对于不同的 OCxCEN 值的行为。在这种情况下，定时器设置为 PWM 模式。

图 10-22 清除 TIMx 的 OCxREF



10.3.12 调试模式

当微控制器处于调试模式（Cortex-M0 内核停止）时，根据 $DBG_CTRL.TIMx_STOP$ 配置，TIMx 计数器可以继续正常工作或停止。详见 3.4.9 章节。

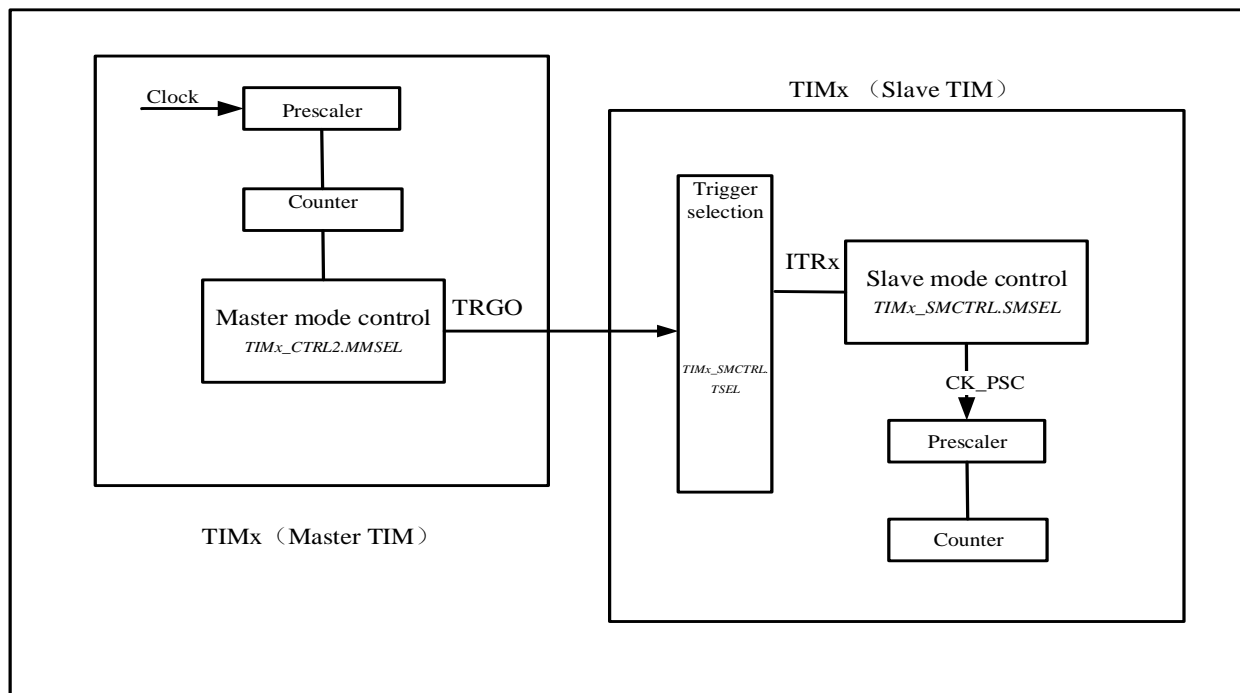
10.3.13 TIMx 定时器和外部触发的同步

与高级定时器相同，见 9.3.16。

10.3.14 定时器同步

所有 TIMx 定时器都在内部相互连接。该实现允许任何主定时器提供触发以复位、启动、停止或为其他从定时器提供时钟。主时钟用于内部计数器，可以预分频。下图为定时器互连框图。同步功能不支持连接的动态变化。用户应在启用主定时器的触发器或时钟之前配置并启用从定时器。

图 10-23 主/从定时器的例子



10.3.14.1 主定时器作为另一个定时器的预分频器

定时器 1 作为定时器 2 的预分频器。TIM1 是主，TIM2 是从。

用户需要为此配置执行以下步骤。

- 设置 TIM1_CTRL2.MMSEL='010' 以使用 TIM1 的更新事件作为触发输出。
- 配置 TIM2_SMCTRL.TSEL='000'，将 TIM1 的 TRGO 连接到 TIM2。
- 配置 TIM2_SMCTRL.SMSEL='111'，从模式控制器将配置为外部时钟模式 1。
- 通过设置 TIM2_CTRL1.CNTEN='1'，启动 TIM2。
- 通过设置 TIM1_CTRL1.CNTEN='1'，启动 TIM1。

注：如果用户通过配置 MMSEL='1xx' 选择 OCx 作为 TIM1 的触发输出，则 OCx 上升沿将用于驱动 TIM2。

10.3.14.2 主定时器使能另一个定时器

在本例中，TIM2 通过 TIM1 的输出比较使能。TIM1 的 OC1REF 输出为高电平后，TIM2 计数器将开始计数。两个计数器的时钟均基于 CK_INT，通过预分频器除以 3 ($f_{CK_CNT} = f_{CK_INT}/3$)。

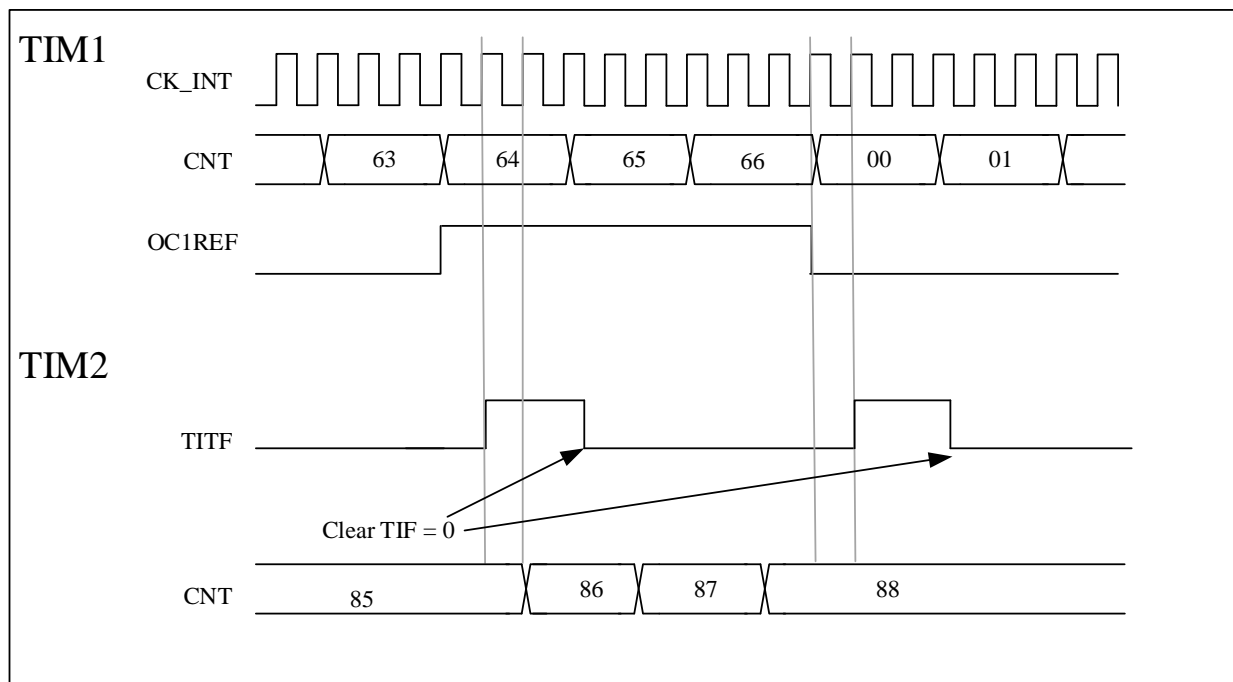
配置步骤如下所示。

- 设置 TIM1_CTRL2.MMSEL='100' 以使用 TIM1 的 OC1REF 作为触发输出。

- 配置 TIM1_CCMOD1 寄存器来配置 OC1REF 输出波形。
- 设置 TIM2_SMCTRL.TSEL = '000' 将 TIM1 触发输出连接到 TIM2。
- 设置 TIM2_SMCTRL.SMSEL = '101' 将 TIM2 设置为门控模式。
- 设置 TIM2_CTRL1.CNTEN = '1' 来启动 TIM2。
- 设置 TIM1_CTRL1.CNTEN = '1' 以启动 TIM1。

注：TIM2 时钟与 TIM1 时钟不同步，该模式仅影响 TIM2 计数器使能信号。

图 10-24 定时器 2 由定时器 1 的 OC1REF 门控



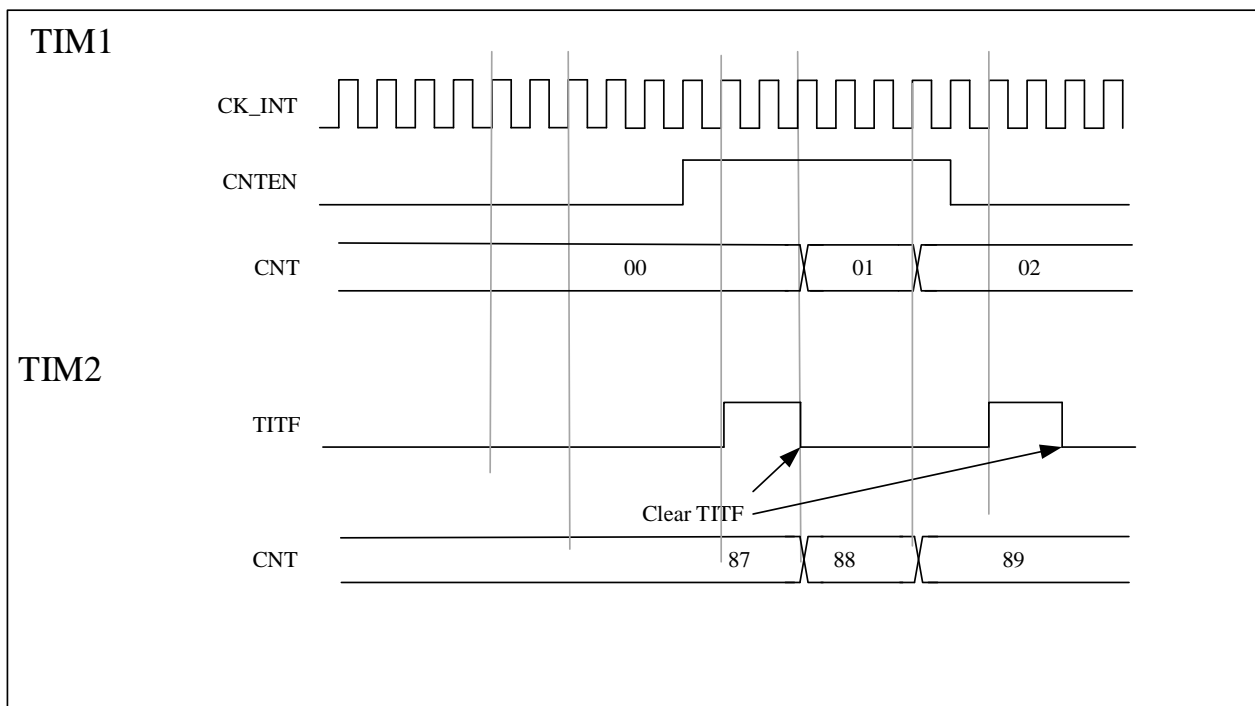
在下一个示例中，用 TIM1 的使能信号门控 TIM2，设置 TIM1_CTRL1.CNTEN = '0' 以停止 TIM1。

仅当 TIM1 使能时，TIM2 才基于分频的内部时钟计数。两个计数器的时钟均基于 CK_INT，通过预分频器除以 3 ($f_{CK_CNT} = f_{CK_INT}/3$)。

配置步骤如下所示

- 设置 TIM1_CTRL2.MMSEL = '001' 使用 TIM1 的使能信号作为触发输出
- 设置 TIM2_SMCTRL.TSEL = '000' 配置 TIM2 从 TIM1 获取触发输入
- 设置 TIM2_SMCTRL.SMSEL = '101' 将 TIM2 配置为门控模式。
- 设置 TIM2_CTRL1.CNTEN = '1' 来启动 TIM2。
- 设置 TIM1_CTRL1.CNTEN = '1' 以启动 TIM1。
- 设置 TIM1_CTRL1.CNTEN = '0' 以停止 TIM1。

图 10-25 定时器 2 由定时器 1 的使能门控



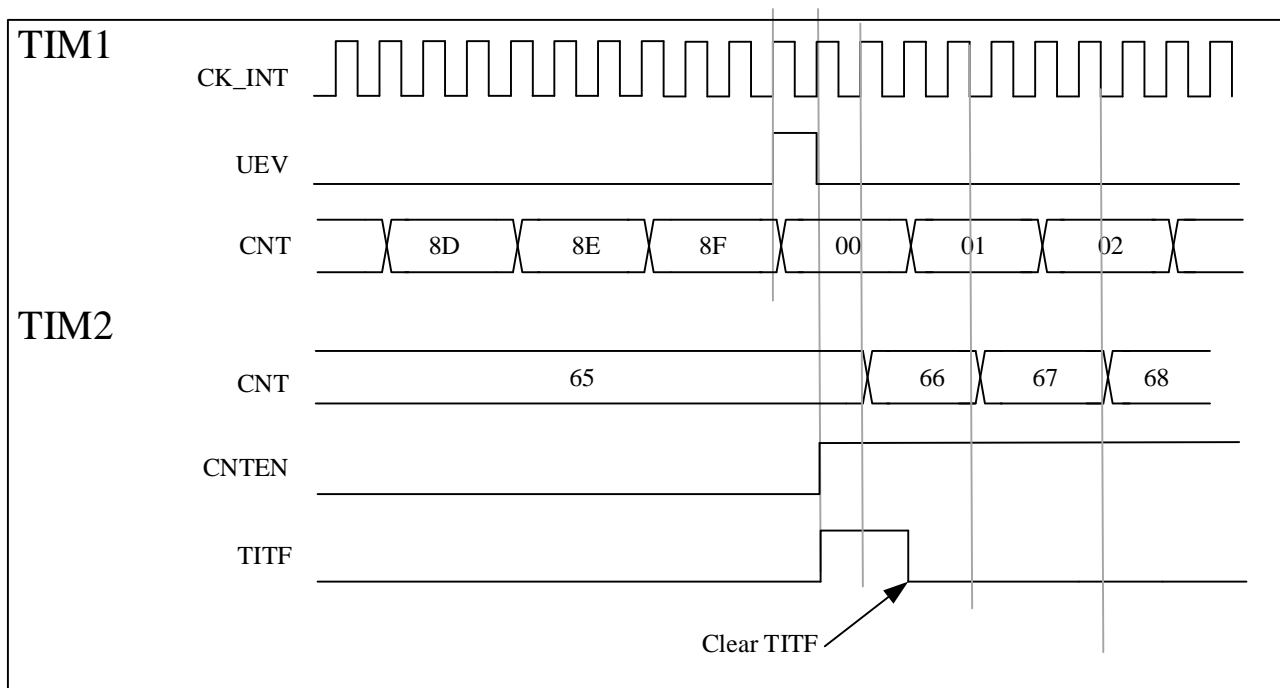
10.3.14.3 主定时器启动另一个定时器

在这个例子中，我们可以使用更新事件作为触发源。TIM1 是主，TIM2 是从。

配置步骤如下图所示：

- 设置 TIM1_CTRL2.MMSEL='010' 使用 TIM1 的更新事件作为触发输出
- 配置 TIM1_AR 寄存器设置输出周期。
- 设置 TIM2_SMCTRL.TSEL='000' 将 TIM1 触发输出连接到 TIM2。
- 设置 TIM2_SMCTRL.SMSEL='110' 将 TIM2 设置为触发模式。
- 设置 TIM1_CTRL1.CNTEN=1 启动 TIM1。

图 10-26 使用定时器 1 的更新触发定时器 2



10.3.14.4 使用一个外部触发同步地启动 2 个定时器

在本例中，TIM1 的 TI1 输入上升时使能 TIM1，使能 TIM1 时使能 TIM2。为确保计数器对齐，TIM1 必须配置为主/从模式。对于 TI1，TIM1 是从；对于 TIM3，TIM1 是主。

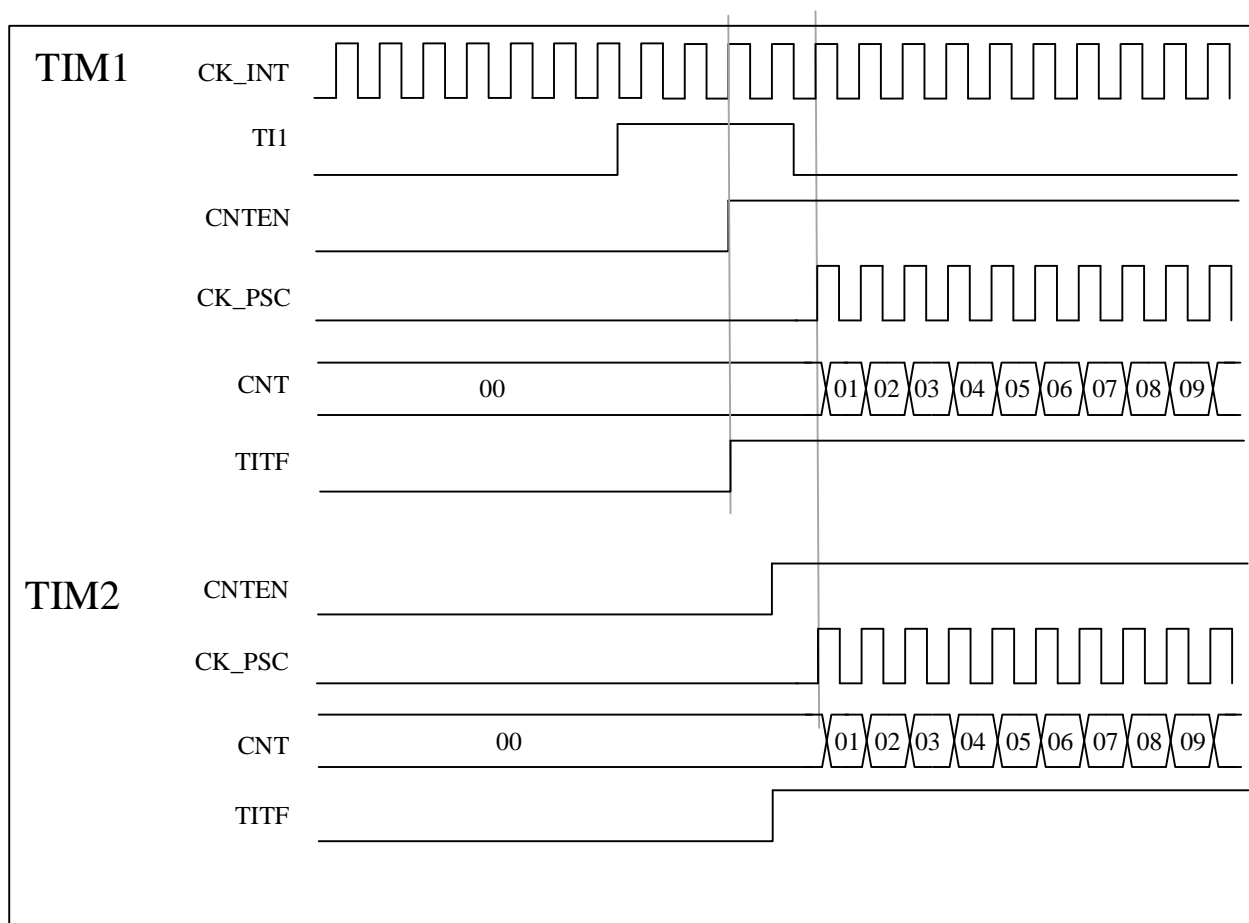
配置步骤如下图所示：

- 设置 TIM1_MMSEL = '001' 使用使能信号作为触发输出
- 设置 TIM1_SMCTRL.TSEL = '100' 将 TIM1 配置为从模式并接收 TI1 的触发输入。
- 设置 TIM1_SMCTRL.SMSEL = '110' 将 TIM1 配置为触发模式。
- 设置 TIM1_SMCTRL.MSMD = '1' 将 TIM1 配置为主/从模式。
- 设置 TIM2_SMCTRL.TSEL = '000' 将 TIM1 触发输出连接到 TIM2。
- 设置 TIM2_SMCTRL.SMSEL = '110' 将 TIM2 配置为触发模式。

当 TI1 上升沿到来时，两个定时器开始根据内部时钟同步计数，两个 TITF 标志同时置位。

注：下图显示了在主/从模式下 CNTEN 和 TIM1 的 CK_PSC 之间的延迟。

图 10-27 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2



10.3.15 编码器接口模式

编码器使用两个输入 TI1 和 TI2 作为接口，计数器对 TI1FP1 或 TI2FP2 上的每个边沿变化进行计数。计数方向由硬件 TIMx_CTRL1.DIR 自动控制。编码器计数模式共有三种：

4. 计数器只在 TI1 的边沿计数，TIMx_SMCTRL.SMSEL = '001'；
5. 计数器只在 TI2 的边沿计数，TIMx_SMCTRL.SMSEL = '010'；
6. 计数器同时在 TI1 和 TI2 的边沿计数，TIMx_SMCTRL.SMSEL = '011'；

编码器接口相当于使用带方向选择的外部时钟，计数器只在 0 和自动重载值 (TIMx_AR.AR [15:0]) 之间连续计数。因此，需要提前配置自动重载寄存器 TIMx_AR。

注意：编码器模式和外部时钟模式 2 不兼容，不能同时选择。

计数方向与编码器信号的关系如下表：

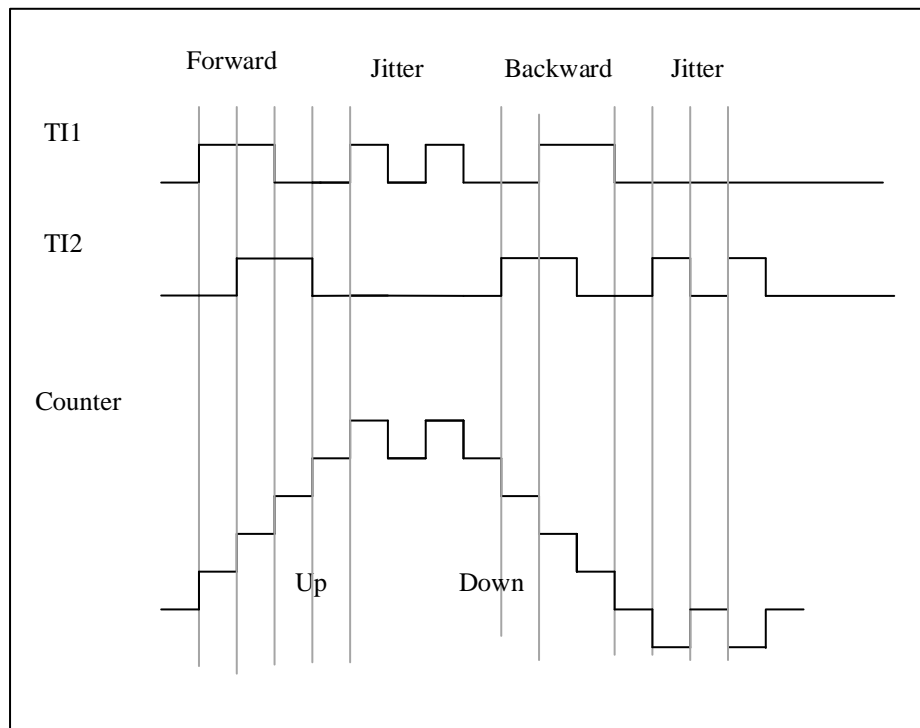
表 10-1 计数方向与编码器信号的关系

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

以下是选择了双边沿触发以抑制输入抖动的编码器示例：

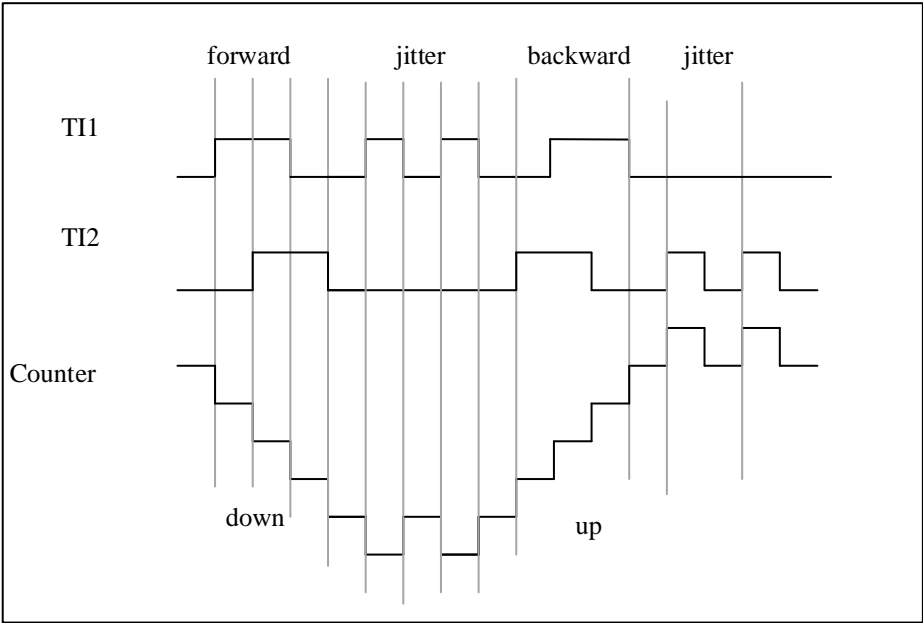
1. IC1FP1 映射到 TI1 (TIMx_CCMOD1.CC1SEL= '01')，IC1FP1 不反相 (TIMx_CCEN.CC1P= '0')；
2. IC1FP2 映射到 TI2 (TIMx_CCMOD2.CC2SEL= '01')，IC2FP2 不反相 (TIMx_CCEN.CC2P= '0')；
3. 输入在上升沿和下降沿均有效 (TIMx_SMCTRL.SMSEL= '011')；
4. 启用计数器 TIMx_CTRL1.CNTEN= '1'；

图 10-28 编码器模式下的计数器操作实例



下图为 IC1FP1 极性反转时的计数器行为示例（CC1P='1'，其他配置同上）

图 10-29 IC1FP1 反相的编码器接口模式实例



10.3.16 与霍尔传感器的接口

请查阅9.3.20节

10.4 TIMx 寄存器描述（x=2/3/4/5）

关于在寄存器描述里面所用到的缩写，详见 1.1 节。
可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

10.4.1 寄存器总览

表 10-2 TIM1 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
000h	TIMx_CTRL1	Reserved												C4SEL	C3SEL	Reserved	C1SEL	Reserved		CLRSEL	IOBKP	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN			
	Reset Value													0	0	Reserved	0	Reserved		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	TIMx_CTRL2	Reserved												TI1SEL	Reserved	CCDSEL	Reserved	MMSEL[2:0]				Reserved														
	Reset Value													0	Reserved	0	Reserved	0				0	0	0	0											
008h	TIMx_STS	Reserved												TITF		UDITF	Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved				CC4ITF	CC3ITF	CC2ITF	CC1ITF				
	Reset Value													0		0	Reserved				0	0	0	0	Reserved				0	0	0	0				
00Ch	TIMx_EVTGEN	Reserved												Reserved								TGN	Reserved	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN				
	Reset Value																					0	Reserved	0	Reserved				0	0	0	0				

010h	TIMx_SMCTRL	Reserved							MSMD	EXTF[3:0]					EXTP	EXCEN	EXTPS[1:0]				Reserved	SMSEL[2:0]				Reserved	TSEL[2:0]								
	Reset Value									0	0	0	0	0	0	0	0	0	0	0		0	0	0	0		0								
014h	TIMx_DINTEN	Reserved					TDEN	Reserved	UDEN	Reserved	TIEN	UEN	Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN							
	0							0		0	0	0					0	0	0	0					0	0	0	0	0						
018h	TIMx_CCMOD1	Reserved							OC2MD[2:0]				OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]				OC1MD[2:0]				OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]								
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
	Reset Value	Reserved							IC2F[3:0]				IC2PSC[1:0]				CC2SEL[1:0]				IC1F[3:0]				IC1PSC[1:0]				CC1SEL[1:0]						
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
01Ch	TIMx_CCMOD2	Reserved							OC4MD[2:0]				OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]				OC3MD[2:0]				OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]								
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
	Reset Value	Reserved							IC4F[3:0]				IC4PSC[1:0]				CC4SEL[1:0]				IC3F[3:0]				IC3PSC[1:0]				CC3SEL[1:0]						
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
020h	TIMx_CCMOD3	Reserved			OC9PEN	Reserved			OC8PEN	Reserved			OC7PEN	OC6MD[2:0]				OC6CEN	OC6FEN	OC6PEN	Reserved				OC5MD[2:0]				OC5CEN	OC5FEN	OC5PEN	Reserved			
	0				0				0				0	0	0	0	0	0	0	0					0	0	0	0	0	0	0				
024h	TIMx_CCEN	Reserved							CC4P	CC4EN	Reserved			CC3P	CC3EN	Reserved				CC2P	CC2EN	Reserved			CC1P	CC1EN	Reserved								
	0								0	0				0	0					0	0				0	0					0	0	0	0	0
028h	TIMx_CCDAT1	Reserved							CCDAT1[15:0]																										
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
02Ch	TIMx_CCDAT2	Reserved							CCDAT2[15:0]																										
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
030h	TIMx_CCDAT3	Reserved							CCDAT3[15:0]																										
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
034h	TIMx_CCDAT4	Reserved							CCDAT4[15:0]																										
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
040h	TIMx_PSC	Reserved							PSC[15:0]																										
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
044h	TIMx_AR	Reserved							AR[15:0]																										
	1								1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
048h	TIMx_CNT	Reserved							CNT[15:0]																										
	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
064h	TIMx_C1FILT	Reserved	THRESH[5:0]					Reserved	WSIZE[5:0]					FILTEN	PSC[15:0]																				
	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0								
068h	TIMx_C2FILT	Reserved	THRESH[5:0]					Reserved	WSIZE[5:0]					FILTEN	PSC[15:0]																				
	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0								
06Ch	TIMx_C3FILT	Reserved	THRESH[5:0]					Reserved	WSIZE[5:0]					FILTEN	PSC[15:0]																				
	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0								
070h	TIMx_C4FILT	Reserved	THRESH[5:0]					Reserved	WSIZE[5:0]					FILTEN	PSC[15:0]																				
	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0								

074h	TIMx_FILTO	Reserved												C4FILTO	C3FILTO	C2FILTO	C1FILTO
	Reset Value													0	0	0	0
078h	TIMx_INSEL	Reserved				C4SEL	Reserved		C3SEL	Reserved		C2SEL	Reserved			C1SEL	
	Reset Value					0			0			0				0	
094h	TIMx_DCTRL	Reserved				DBADDR[4:0]					Reserved		DBLEN[4:0]				
	Reset Value					0	0	0	0	0			0	0	0	0	0
098Ch	TIMx_DADDR	Reserved				BURST[15:0]											
	Reset Value					0	0	0	0	0	0	0	0	0	0	0	0

10.4.2 控制寄存器 1 (TIMx_CTRL1)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												C4SEL	C3SEL	Reserved	C1SEL
												rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CLRSEL	Reserved				ARPEN	ONEPM	CLKD[1:0]	UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN	
	rw					rw	rw	rw	rw	rw	rw		rw	rw	

位域	名称	描述
31:24	Reserved	保留, 必须保持复位值
19	C4SEL	通道1选择 (Channel 4 selection) 0: 选择外部CH4 (来自IOM或者COMP, 具体选择见TIMx_INSEL.CH4SEL) 信号 1: 选择内部CH4 (来自HSE/128) 信号 注: 仅TIM2有效
18	C3SEL	通道1选择 (Channel 3 selection) 0: 选择外部CH3 (来自IOM或者COMP, 具体选择见TIMx_INSEL.CH3SEL) 信号 1: 选择内部CH3 (来自LSI) 信号 注: 仅TIM2有效
17	Reserved	保留, 必须保持复位值
16	C1SEL	通道1选择 (Channel 1 selection) 0: 选择外部CH1 (来自IOM或者COMP, 具体选择见TIMx_INSEL.CH1SEL) 信号 1: 选择内部CH1 (来自HSI) 信号 注: 仅TIM2有效
15:14	Reserved	保留, 必须保持复位值
13	CLRSEL	OCxRef选择 (OCxRef selection) 0: 选择外部Ocxclr (ETR) 信号 1: 选择内部Ocxclr (来自COMP) 信号
12:10	Reserved	保留, 必须保持复位值
9	ARPEN	自动重载预装载允许位 (Auto-reload preload enable) 0: TIMx_AR 寄存器的影子寄存器禁用 1: TIMx_AR 寄存器的影子寄存器使能
8	ONEPM	单脉冲模式 (One pulse mode) 0: 禁用单脉冲模式, 发生更新事件时不影响计数器计数。 1: 使能单脉冲模式, 下次更新事件发生时计数器停止计数
7:6	CLKD[1:0]	时钟分频因子 (Clock division) CLKD[1:0] 表示 CK_INT (定时器时钟) 和 DTS (用于死区时间发生器和数字滤波器 (ETR、TIx) 的时钟) 之间的分频比。 00: $t_{DTS} = t_{CK_INT}$

位域	名称	描述
		<p>01: $t_{DTS} = 2 \times t_{CK_INT}$</p> <p>10: $t_{DTS} = 4 \times t_{CK_INT}$</p> <p>11: 保留, 不要使用这个配置</p>
5	UPDIS	<p>更新禁用 (Update disable)</p> <p>该位用于启用/禁用软件生成的更新事件 (UEV) 事件。</p> <p>0: 启用。 如果满足以下条件之一, 将生成 UEV:</p> <ul style="list-style-type: none"> – 计数器上溢/下溢 – TIMx_EVTGEN.UDGN 位被设置 – 从模式控制器的更新生成 <p>影子寄存器将使用预加载值进行更新。</p> <p>1: UEV 禁用。 不生成更新事件, 影子寄存器 (AR、PSC 和 CCDA Tx) 保持它们的值。 如果 TIMx_EVTGEN.UDGN 位置位或从模式控制器发出硬件复位, 则重新初始化计数器和预分频器。</p>
4	UPRS	<p>更新请求源 (Update request source)</p> <p>该位用于通过软件选择 UEV 事件源。</p> <p>0: 如果更新中断或 DMA 请求使能, 以下任何事件都会产生更新中断或 DMA 请求:</p> <ul style="list-style-type: none"> – 计数器上溢/下溢 – TIMx_EVTGEN.UDGN 位被设置 – 从模式控制器的更新生成 <p>1: 如果更新中断或 DMA 请求使能, 只有计数器上溢/下溢会产生更新中断或 DMA 请求。</p>
3:2	CAMSEL[1:0]	<p>选择中央对齐模式 (Center-aligned mode selection)</p> <p>00: 边缘对齐模式。 TIMx_CTRL1.DIR 指定向上计数或向下计数。</p> <p>01: 中央对齐模式1。 计数器在中央对齐模式下计数, 向下计数时输出比较中断标志位设置为 1。</p> <p>10: 中央对齐模式2。 计数器在中央对齐模式下计数, 向上计数时输出比较中断标志位设置为1。</p> <p>11: 中央对齐模式3。 计数器在中央对齐模式下计数, 向上计数或向下计数时输出比较中断标志位设置为 1。</p> <p><i>注意: 当计数器仍然启用时 (TIMx_CTRL1.CNTEN = 1), 不允许从边缘对齐模式切换到中央对齐模式。</i></p>
1	DIR	<p>方向 (Direction)</p> <p>0: 计数器向上计数;</p> <p>1: 计数器向下计数。</p> <p><i>注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。</i></p>
0	CNTEN	<p>使能计数器 (Counter enable)</p> <p>0: 禁止计数器;</p> <p>1: 使能计数器。</p> <p><i>注: 在软件设置了CNTEN位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。</i></p>

10.4.3 控制寄存器 2 (TIMx_CTRL2)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TI1SEL	Reserved	CCDSEL	Reserved
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMSEL[3:0]				Reserved											
rw															

位域	名称	描述
31:20	Reserved	保留, 必须保持复位值
19	TI1SEL	TI1选择 (TI1 selection) 0: TIMx_CH1 引脚连到TI1输入; 1: TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
18	Reserved	保留, 必须保持复位值
17	CCDSEL	捕获/比较的DMA选择 (Capture/compare DMA selection) 0: 当发生CCx事件时, 送出CCx的DMA请求; 1: 当发生更新事件时, 送出CCx的DMA请求。
16	Reserved	保留, 必须保持复位值
15:12	MMSEL[3:0]	主模式选择 这 4 位用于选择在主模式下发送到从定时器的同步信息 (TRGO)。可能的组合如下: X000: 复位 - 当 TIMx_EVTGEN.UDGN 置位或从模式控制器产生复位时, 将出现 TRGO 脉冲。在后一种情况下, TRGO 上的信号与实际复位相比有所延迟。 X001: 使能 - TIMx_CTRL1.CNTEN 位用作触发输出 (TRGO)。有时需要同时启动多个定时器或者在一段时间内开启从定时器。 当 TIMx_CTRL1.CNTEN 位置位或门控模式下的触发输入为高电平时, 计数器使能信号置位。 当计数器使能信号由触发输入控制时, TRGO 上有一个延迟, 除非选择了主/从模式 (参见 TIMx_SMCTRL.MSMD 位的说明)。 X010: 更新 - 选择更新事件作为触发输出 (TRGO)。例如, 主定时器时钟可用作从定时器预分频器。 X011: 比较脉冲 - 当 TIMx_STS.CC1ITF 被设置时 (即使它已经是高电平), 即捕获或比较成功时, 触发输出发送一个正脉冲 (TRGO)。 X100: 比较 - OC1REF 信号用作触发输出 (TRGO)。 X101: 比较 - OC2REF 信号用作触发输出 (TRGO)。 X110: 比较 - OC3REF 信号用作触发输出 (TRGO)。 X111: 比较 - OC4REF 信号用作触发输出 (TRGO)。
11:0	Reserved	保留, 必须保持复位值

10.4.4 状态寄存器 (TIMx_STS)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TITF	Reserved	UDITF
													rc_w0		rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved				CC4ITF	CC3ITF	CC2ITF	CC1ITF
				rc_w0	rc_w0	rc_w0	rc_w0					rc_w0	rc_w0	rc_w0	rc_w0

位域	名称	描述
31:19	Reserved	保留, 必须保持复位值
18	TITF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在TRGI输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置'1'。它由软件清'0'。 0: 无触发器事件产生; 1: 触发中断等待响应。
17	Reserved	保留, 必须保持复位值
16	UDITF	更新中断标志 (Update interrupt flag) 当在以下条件下发生更新事件时, 该位由硬件设置: - 当 TIMx_CTRL1.UPDIS = 0 时, 并且重复计数器值上溢或下溢 (当重复计数器等于 0 时生成更新事件UEV)。 - 当 TIMx_CTRL1.UPRS = 0 时, TIMx_CTRL1.UPDIS = 0, 并通过软件设置 TIMx_EVTGEN.UDGN 位以重新初始化 CNT。 - 当 TIMx_CTRL1.UPRS = 0 时, TIMx_CTRL1.UPDIS = 0, 并且计数器 CNT 由触发事件重新初始化。 (参见 TIMx_SMCTRL 寄存器说明) 该位由软件清零。 0: 未发生更新事件 1: 发生更新中断
15:12	Reserved	保留, 必须保持复位值
11	CC4OCF	捕获/比较4重复捕获标记 (Capture/Compare 4 overcapture flag) 参见CC1OCF描述。
10	CC3OCF	捕获/比较3重复捕获标记 (Capture/Compare 3 overcapture flag) 参见CC1OCF描述。
9	CC2OCF	捕获/比较2重复捕获标记 (Capture/Compare 2 overcapture flag) 参见CC1OCF描述。
8	CC1OCF	捕获/比较1重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置1。写0可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到TIMx_CCDAT1寄存器时, CC1ITF的状态已经为'1'。

位域	名称	描述
7:4	Reserved	保留，必须保持复位值
3	CC4ITF	捕获/比较4中断标记（Capture/Compare 4 interrupt flag） 参考CC1ITF描述。
2	CC3ITF	捕获/比较3中断标记（Capture/Compare 3 interrupt flag） 参考CC1ITF描述。
1	CC2ITF	捕获/比较2中断标记（Capture/Compare 2 interrupt flag） 参考CC1ITF描述。
0	CC1ITF	捕获/比较1中断标记（Capture/Compare 1 interrupt flag） 如果通道CC1配置为输出模式： 除中央对齐模式外，当计数器值与比较值相同时，该位由硬件设置（参见TIMx_CTRL1.CAMSEL 位描述）。该位由软件清零。 0：未发生匹配。 1：TIMx_CNT 的值与 TIMx_CC1 的值相同。 当 TIMx_CC1 的值大于 TIMx_ARR 的值时，如果计数器溢出（在向上计数和向上/向下计数模式下）和向下计数模式下溢，则 TIMx_STS.CC1ITF 位将变为高电平。 如果通道CC1配置为输入模式： 当捕捉事件发生时，该位由硬件设置。该位由软件或读取 TIMx_CC1 清零。 0：未发生输入捕捉。 1：发生输入捕捉。计数器值已在 TIMx_CC1 中捕获。在 IC1 上检测到与所选极性相同的边沿。

10.4.5 事件产生寄存器（TIMx_EVTGEN）

偏移地址:0x0C

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TGN	Reserved	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN
w					w				w				w	w	w

位域	名称	描述
31:11	Reserved	保留，必须保持复位值
10	TGN	产生触发事件（Trigger generation） 当由软件置位时，该位可以产生一个触发事件。而此时TIMx_STS.TITF = 1，如果相应的中断和DMA被使能，就会产生相应的中断和DMA。该位由硬件自动清零。 0：无动作 1：产生触发事件
9	Reserved	保留，必须保持复位值

位域	名称	描述
8	UDGN	产生更新事件（Update generation） 该位由软件置'1'，由硬件自动清'0'。 当由软件设置时，该位可以生成更新事件。而此时计数器会重新初始化，预分频计数器会被清零，计数器在中央对齐或向上计数模式下会被清零，但在向下计数模式下取TIMx_AR寄存器的值。 该位由硬件自动清零。 0：无动作 1：生成更新事件
7:4	Reserved	保留，必须保持复位值
3	CC4GN	产生捕获/比较4事件（Capture/Compare 4 generation） 参考CC1GN描述。
2	CC3GN	产生捕获/比较3事件（Capture/Compare 3 generation） 参考CC1GN描述。
1	CC2GN	产生捕获/比较2事件（Capture/Compare 2 generation） 参考CC1GN描述。
0	CC1GN	产生捕获/比较1事件（Capture/Compare 1 generation） 当由软件设置时，该位可以产生一个捕获/比较事件。 该位由硬件自动清零。 CC1对应通道为输出模式时： TIMx_STS.CC1ITF 标志将被拉高，如果相应的中断和 DMA 被使能，就会产生相应的中断和 DMA。 CC1对应通道为输入模式时： TIMx_CC1ITF 将捕获当前计数器值，并将 TIMx_STS.CC1ITF 标志拉高，如果相应的中断和 DMA 被使能，则会产生相应的中断和 DMA。 如果 TIMx_STS.CC1ITF 已经拉高，则拉高 TIMx_STS.CC1OCF。 0：无动作 1：生成 CC1 捕获/比较事件

10.4.6 从模式控制寄存器（TIMx_SMCTRL）

偏移地址：0x10

复位值：0x0000 0000

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

位域	名称	描述
31:17	Reserved	保留，必须保持复位值
16	MSMD	主/从模式（Master/slave mode） 0：无作用； 1：触发输入（TRGI）上的事件被延迟了，以允许在当前定时器（通过TRGO）与它的从定时器间

		的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
15:12	EXTF[3:0]	<p>外部触发滤波 (External trigger filter)</p> <p>这些位用于定义 ETRP 信号的采样频率和 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 在记录连续 N 个事件后生成验证输出。</p> <p>0000: 无滤波器, 以fDTS采样 1000: 采样频率fSAMPLING=fDTS/8, N=6</p> <p>0001: 采样频率fSAMPLING=fCK_INT, N=2 1001: 采样频率fSAMPLING=fDTS/8, N=8</p> <p>0010: 采样频率fSAMPLING=fCK_INT, N=4 1010: 采样频率fSAMPLING=fDTS/16, N=5</p> <p>0011: 采样频率fSAMPLING=fCK_INT, N=8 1011: 采样频率fSAMPLING=fDTS/16, N=6</p> <p>0100: 采样频率fSAMPLING=fDTS/2, N=6 1100: 采样频率fSAMPLING=fDTS/16, N=8</p> <p>0101: 采样频率fSAMPLING=fDTS/2, N=8 1101: 采样频率fSAMPLING=fDTS/32, N=5</p> <p>0110: 采样频率fSAMPLING=fDTS/4, N=6 1110: 采样频率fSAMPLING=fDTS/32, N=6</p> <p>0111: 采样频率fSAMPLING=fDTS/4, N=8 1111: 采样频率fSAMPLING=fDTS/32, N=8</p>
11	EXTP	<p>外部触发极性 (External trigger polarity)</p> <p>该位选择是用ETR还是ETR的反相来作为触发操作</p> <p>0: ETR高电平或上升沿有效;</p> <p>1: ETR低电平或下降沿有效。</p>
10	EXCEN	<p>外部时钟使能位 (External clock enable) 该位启用外部时钟模式2。启用后, 计数器由ETRF信号上的任意有效边沿驱动。</p> <p>0: 禁止外部时钟模式2;</p> <p>1: 使能外部时钟模式2。</p> <p>注 1: 当同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入为 ETRF。</p> <p>注2: 以下从机模式可以与外部时钟模式2同时使用: 复位模式、门控模式和触发模式; 但是, TRGI 无法连接到 ETRF (TIMx_SMCTRL.TSEL ≠ '111')。</p> <p>注 3: 设置 TIMx_SMCTRL.EXCEN 位与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (TIMx_SMCTRL.SMSEL = 111 和 TIMx_SMCTRL.TSEL = 111) 的效果相同</p>
9:8	EXTPS[1:0]	<p>外部触发预分频 (External trigger prescaler)</p> <p>外部触发信号 ETRP 的频率必须最多为 TIMxCLK 频率的 1/4。当输入更快的外部时钟时, 可以使用预分频器来降低 ETRP 的频率。</p> <p>00: 关闭预分频;</p> <p>01: ETRP频率除以2;</p> <p>10: ETRP频率除以4;</p> <p>11: ETRP频率除以8。</p>
7	Reserved	保留, 必须保持复位值
6:4	SMSEL[2:0]	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明)</p> <p>000: 关闭从模式 – 如果CNTEN=1, 则预分频器直接由内部时钟驱动。</p> <p>001: 编码器模式1 – 根据TI2FP2的电平, 计数器在TI1FP1的边沿向上/下计数。</p> <p>010: 编码器模式2 – 根据TI1FP1的电平, 计数器在TI2FP2的边沿向上/下计数。011: 编码器模式3 – 根据另一个信号的输入电平, 计数器在TI1FP1和TI2FP2的边沿向上/下计数。</p> <p>100: 复位模式 – 在选定触发输入 (TRGI) 的上升沿, 计数器重新初始化并更新影子寄存器。</p> <p>101: 门控模式 – 当触发输入 (TRGI) 为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止 (但不复位)。计数器的启动和停止都是受控的。</p> <p>110: 触发模式 – 计数器在触发输入TRGI的上升沿启动 (但不复位), 只有计数器的启动是受控</p>

		的。 111：外部时钟模式1 – 选中的触发输入（TRGI）的上升沿驱动计数器。 <i>注：如果TI1F_ED被选为触发输入（TSEL=100）时，不要使用门控模式。这是因为，TI1F_ED在每次TI1F变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。</i>
3	Reserved	保留，必须保持复位值
2:0	TSEL[2:0]	触发选择（Trigger selection） 这3位选择用于同步计数器的触发输入。 000：内部触发0（ITR0） 100：TI1的边沿检测器（TI1F_ED） 001：内部触发1（ITR1） 101：滤波后的定时器输入1（TI1FP1） 010：内部触发2（ITR2） 110：滤波后的定时器输入2（TI2FP2） 011：内部触发3（ITR3） 111：外部触发输入（ETRF） 更多有关ITRx的细节，参见表9-3。 <i>注：这些位只能在未用到（如SMSEL=000）时被改变，以避免在改变时产生错误的边沿检测。</i>

表 10-3 TIMx 内部触发连接

Slave timer	ITR0 (TSEL = 000)	ITR1 (TSEL = 001)	ITR2 (TSEL = 010)	ITR3 (TSEL = 011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM2	TIM1	NA	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	NA
TIM5	TIM2	TIM3	TIM4	NA

10.4.7 DMA/中断使能寄存器（TIMx_DINTEN）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TDEN	Reserved	UDEN	Reserved	TIEN	UIEN
										rw		rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw					rw	rw	rw	rw

位域	名称	描述
31:22	Reserved	保留，必须保持复位值
21	TDEN	允许触发DMA请求（Trigger DMA request enable） 0：禁止触发DMA请求； 1：允许触发DMA请求。
20	Reserved	保留，必须保持复位值

位域	名称	描述
19	UDEN	允许更新的DMA请求 (Update DMA request enable) 0: 禁止更新的DMA请求; 1: 允许更新的DMA请求。
18	Reserved	保留, 必须保持复位值
17	TIEN	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
16	UIEN	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。
15:12	Reserved	保留, 必须保持复位值
11	CC4DEN	允许捕获/比较4的DMA请求 (Capture/Compare 4 DMA request enable) 0: 禁止捕获/比较4的DMA请求; 1: 允许捕获/比较4的DMA请求。
10	CC3DEN	允许捕获/比较3的DMA请求 (Capture/Compare 3 DMA request enable) 0: 禁止捕获/比较3的DMA请求; 1: 允许捕获/比较3的DMA请求。
9	CC2DEN	允许捕获/比较2的DMA请求 (Capture/Compare 2 DMA request enable) 0: 禁止捕获/比较2的DMA请求; 1: 允许捕获/比较2的DMA请求。
8	CC1DEN	允许捕获/比较1的DMA请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较1的DMA请求; 1: 允许捕获/比较1的DMA请求。
7:4	Reserved	保留, 必须保持复位值
3	CC4IEN	允许捕获/比较4中断 (Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。
2	CC3IEN	允许捕获/比较3中断 (Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。
1	CC2IEN	允许捕获/比较2中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。
0	CC1IEN	允许捕获/比较1中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。

10.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1)

偏移地址：0x18

复位值：0x0000 0000

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

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:13	OC2MD[2:0]	输出比较2模式（Output Compare 2 mode）
12	OC2CEN	输出比较2清0使能（Output Compare 2 clear enable）
11	OC2FEN	输出比较2快速使能（Output Compare 2 fast enable）
10	OC2PEN	输出比较2预装载使能（Output Compare 2 preload enable）
9:8	CC2SEL[1:0]	捕获/比较2选择。（Capture/Compare 2 selection） 该位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:5	OC1MD[2:0]	输出比较1模式（Output Compare 1 mode） 这些位用于管理输出参考信号 OC1REF，它决定了 OC1 和 OC1N 的值，在高电平有效，而 OC1 和 OC1N 的有效电平取决于 TIMx_CCEN.CC1P 和 TIMx_CCEN.CC1NP 位。 000：冻结。TIMx_CCDAT1 寄存器和计数器 TIMx_CNT 之间的比较对 OC1REF 信号没有影响。 001：将通道 1 设置为匹配时的有效电平。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被强制为高电平。 010：将通道 1 设置为匹配时的无效电平。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被强制为低电平。 011：翻转。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被翻转。 100：强制无效电平。OC1REF 信号被强制为低电平。

位域	名称	描述
		<p>101: 强制有效电平。OC1REF 信号被强制为高电平。</p> <p>110: PWM 模式 1 - 在向上计数模式下, 如果 $TIMx_CNT < TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。在向下计数模式下, 如果 $TIMx_CNT > TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。</p> <p>111: PWM 模式 2 - 在向上计数模式下, 如果 $TIMx_CNT < TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。在向下计数模式下, 如果 $TIMx_CNT > TIMx_CCDAT1$, 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。</p> <p><i>注 1: 在 PWM 模式 1 或 PWM 模式 2 中, OC1REF 电平仅在比较结果改变或输出比较模式从冻结模式切换到 PWM 模式时才会改变。</i></p>
4	OC1CEN	<p>输出比较1清'0'使能 (Output Compare 1 clear enable)</p> <p>0: OC1REF 不受ETRF输入的影响;</p> <p>1: 一旦检测到ETRF输入高电平, 清除OC1REF=0。</p>
3	OC1FEN	<p>输出比较1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快CC输出对触发输入事件的响应。</p> <p>0: 根据计数器与CCDAT1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就像发生了一次比较匹配。因此, OC1被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。</p> <p>OCxFEN只在通道被配置成PWM1或PWM2模式时起作用。</p>
2	OC1PEN	<p>输出比较 1 预加载使能 (Output Compare 1 preload enable)</p> <p>0: 禁用 $TIMx_CCDAT1$ 寄存器的预加载功能。支持随时对 $TIMx_CCDAT1$ 寄存器进行写操作, 写入的值立即生效。</p> <p>1: 使能 $TIMx_CCDAT1$ 寄存器的预加载功能。仅对预加载寄存器进行读写操作。当更新事件发生时, $TIMx_CCDAT1$ 的值被加载到影子寄存器中。</p> <p><i>注 1: 只有当 $TIMx_CTRL1.ONEPM = 1$ (在单脉冲模式下) 时, 才能使用 PWM 模式而不验证预加载寄存器, 否则无法预测其他行为。</i></p>
1:0	CC1SEL[1:0]	<p>捕获/比较1 选择。(Capture/Compare 1 selection)</p> <p>这2位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(由 $TIMx_SMCTRL$ 寄存器的TSEL位选择)。</p> <p><i>注: CC1SEL仅在通道关闭时($TIMx_CCEN$寄存器的CC1EN=0)才是可写的。</i></p>

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw		rw		rw				rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC2F[3:0]	输入捕获2滤波器 （Input capture 2 filter）
11:10	IC2PSC[1:0]	输入/捕获2预分频器 （Input capture 2 prescaler）
9:8	CC2SEL[1:0]	捕获/比较2选择 （Capture/Compare 2 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:4	IC1F[3:0]	输入捕获1滤波器 （Input capture 1 filter） 这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以f _{DTS} 采样 1000：采样频率f _{SAMPLING} =f _{DTS} /8，N=6 0001：采样频率f _{SAMPLING} =f _{CK_INT} ，N=2 1001：采样频率f _{SAMPLING} =f _{DTS} /8，N=8 0010：采样频率f _{SAMPLING} =f _{CK_INT} ，N=4 1010：采样频率f _{SAMPLING} =f _{DTS} /16，N=5 0011：采样频率f _{SAMPLING} =f _{CK_INT} ，N=8 1011：采样频率f _{SAMPLING} =f _{DTS} /16，N=6 0100：采样频率f _{SAMPLING} =f _{DTS} /2，N=6 1100：采样频率f _{SAMPLING} =f _{DTS} /16，N=8 0101：采样频率f _{SAMPLING} =f _{DTS} /2，N=8 1101：采样频率f _{SAMPLING} =f _{DTS} /32，N=5 0110：采样频率f _{SAMPLING} =f _{DTS} /4，N=6 1110：采样频率f _{SAMPLING} =f _{DTS} /32，N=6 0111：采样频率f _{SAMPLING} =f _{DTS} /4，N=8 1111：采样频率f _{SAMPLING} =f _{DTS} /32，N=8
3:2	IC1PSC[1:0]	输入/捕获1预分频器 （Input capture 1 prescaler） 这2位定义了CC1输入（IC1）的预分频系数。 一旦TIMx_CCEN.CC1EN=0，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	CC1SEL[1:0]	<p>捕获/比较1选择 (Capture/Compare 1 Selection)</p> <p>这2位定义通道的方向 (输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。</p> <p><i>注: CC1SEL仅在通道关闭时 (TIMx_CCEN寄存器的CC1EN=0) 才是可写的。</i></p>
-----	-------------	--

10.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2)

偏移地址: 0x1C

复位值: 0x0000 0000

参看以上 CCMOD1 寄存器的描述

输出比较模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]		OC4CEN		OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]			OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]	
rw		rw		rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:13	OC4MD[2:0]	输出比较4模式 (Output compare 4 mode)
12	OC4CEN	输出比较4清0使能 (Output compare 4 clear enable)
11	OC4FEN	输出比较4快速使能 (Output compare 4 fast enable)
10	OC4PEN	输出比较4预装载使能 (Output compare 4 preload enable)
9:8	CC4SEL[1:0]	<p>捕获/比较4选择 (Capture/Compare 4 selection)</p> <p>该2位定义通道的方向 (输入/输出), 及输入脚的选择:</p> <p>00: CC4通道被配置为输出;</p> <p>01: CC4通道被配置为输入, IC4映射在TI4上;</p> <p>10: CC4通道被配置为输入, IC4映射在TI3上;</p> <p>11: CC4通道被配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。</p> <p><i>注: CC4SEL仅在通道关闭时 (TIMx_CCEN寄存器的CC4EN=0) 才是可写的。</i></p>
7:5	OC3MD[2:0]	输出比较3模式 (Output compare 3 mode)
4	OC3CEN	输出比较3清0使能 (Output compare 3 clear enable)
3	OC3FEN	输出比较3快速使能 (Output compare 3 fast enable)
2	OC3PEN	输出比较3预装载使能 (Output compare 3 preload enable)
1:0	CC3SEL[1:0]	<p>捕获/比较3选择 (Capture/Compare 3 selection)</p> <p>这2位定义通道的方向 (输入/输出), 及输入脚的选择:</p>

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
		00: CC3通道被配置为输出； 01: CC3通道被配置为输入，IC3映射在TI3上； 10: CC3通道被配置为输入，IC3映射在TI4上； 11: CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC3SEL仅在通道关闭时（TIMx_CCEN寄存器的CC3EN=0）才是可写的。</i>

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw		rw		rw				rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC4F[3:0]	输入捕获4滤波器 (Input capture 4 filter)
11:10	IC4PSC[1:0]	输入/捕获4预分频器 (Input capture 4 prescaler)
9:8	CC4SEL[1:0]	捕获/比较4选择 (Capture/Compare 4 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4通道被配置为输出； 01: CC4通道被配置为输入，IC4映射在TI4上； 10: CC4通道被配置为输入，IC4映射在TI3上； 11: CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注：CC4SEL仅在通道关闭时(TIMx_CCEN寄存器的CC4EN=0)才是可写的。</i>
7:4	IC3F[3:0]	输入捕获3滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获3预分频器 (Input capture 3 prescaler)
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3通道被配置为输出； 01: CC3通道被配置为输入，IC3映射在TI3上； 10: CC3通道被配置为输入，IC3映射在TI4上； 11: CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注：CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。</i>

10.4.10 捕获/比较使能寄存器 (TIMx_CCEN)

偏移地址: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4P	CC4EN	Reserved		CC3P	CC3EN	Reserved		CC2P	CC2EN	Reserved		CC1P	CC1EN	Reserved	
rw		rw		rw		rw		rw		rw		rw		rw	
位域	名称		描述												
31:16	Reserved		保留，必须保持复位值												
15	CC4P		捕获/比较4输出极性 （Capture/Compare 4 output polarity） 参考TIMx_CCEN.CC1P的描述。												
14	CC4EN		捕获/比较4输出使能 （Capture/Compare 4 output enable） 参考TIMx_CCEN.CC1EN 的描述。												
13:12	Reserved		保留，必须保持复位值												
11	CC3P		捕获/比较3输出极性 （Capture/Compare 3 output polarity） 参考TIMx_CCEN.CC1P的描述。												
10	CC3EN		捕获/比较3输出使能 （Capture/Compare 3 output enable） 参考TIMx_CCEN.CC1E 的描述。												
9:8	Reserved		保留，必须保持复位值												
7	CC2P		捕获/比较2输出极性 （Capture/Compare 2 output polarity） 参考TIMx_CCEN.CC1P的描述。												
6	CC2EN		捕获/比较2输出使能 （Capture/Compare 2 output enable） 参考TIMx_CCEN.CC1EN的描述。												
5:4	Reserved		保留，必须保持复位值												
3	CC1P		捕获/比较1输出极性 （Capture/Compare 1 output polarity） CC1对应通道为输出模式时： 0： OC1 高电平有效 1： OC1 低电平有效 CC1对应通道为输入模式时： 此时，该位用于选择是使用IC1还是IC1的反相信号作为触发信号或捕捉信号。 0： 非反相：当 IC1 产生上升沿时发生捕获动作。 当用作外部触发时， IC1 是非反相的。 1： 反相：当 IC1 产生下降沿时发生捕获动作。 当用作外部触发时， IC1 被反相。												
2	CC1EN		捕获/比较1输出使能 （Capture/Compare 1 output enable） CC1通道配置为输出： 0： 关闭— OC1禁止输出，因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 1： 开启— OC1信号输出到对应的输出引脚，其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 CC1通道配置为输入：												

位域	名称	描述
		该位决定了计数器的值是否能捕获入TIMx_CC DAT1寄存器。 0: 捕获禁止; 1: 捕获使能。
1:0	Reserved	保留, 必须保持复位值

表 10-4 标准 OCx 的输出控制位

CCxEN	OCx output status
0	Disable output (OCx=0)
1	OCx = OCxREF + polarity

注: 连接到标准 OCx 通道的外部 I/O 引脚的状态取决于 OCx 通道状态以及 GPIO 和 AFIO 寄存器。

10.4.11 捕获/比较寄存器 1 (TIMx_CC DAT1)

偏移地址: 0x28

复位值: 0x0000 0000

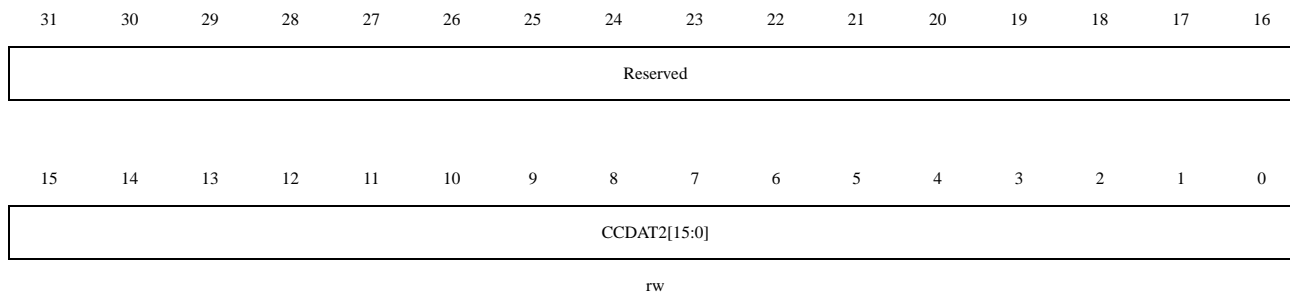
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC DAT1[15:0]															
rw															

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:0	CC DAT1[15:0]	捕获/比较通道1的值 (Capture/Compare 1 value) ■ CC1 通道配置为输出: CC DAT1 包含要与计数器 TIMx_CNT 比较的值, 在 OC1 输出上发出信号。 如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。 ■ CC1 通道配置为输入: CC DAT1 包含由最后一个输入捕获 1 事件 (IC1) 传输的计数器值。 当配置为输入模式时, 寄存器 CC DAT1 和 CC DDAT1 只能读取。 当配置为输出模式时, 寄存器 CC DAT1 和 CC DDAT1 是可读写的。

10.4.12 捕获/比较寄存器 2 (TIMx_CC DAT2)

偏移地址: 0x2C

复位值: 0x0000 0000

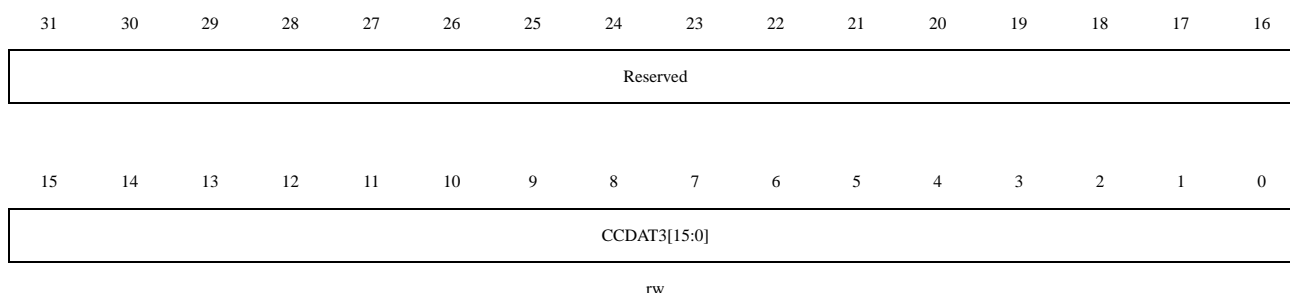


位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT2[15:0]	<p>捕获/比较通道2的值（Capture/Compare 2 value）</p> <p>■ CC2 通道配置为输出： CCDAT2 包含要与计数器 TIMx_CNT 比较的值，在 OC2 输出上发出信号。 如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</p> <p>■ CC2 通道配置为输入： CCDAT2 包含由最后一个输入捕获 2 事件 (IC2) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT2 和 CCDDAT2 只能读取。 当配置为输出模式时，寄存器 CCDAT2 和 CCDDAT2 是可读写的。</p>

10.4.13 捕获/比较寄存器 3 (TIMx_CCDAT3)

偏移地址：0x30

复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT3[15:0]	<p>捕获/比较通道3的值（Capture/Compare 3 value）</p> <p>■ CC3 通道配置为输出： CCDAT3 包含要与计数器 TIMx_CNT 比较的值，在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</p> <p>■ CC3 通道配置为输入：</p>

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
		CCDAT3 包含由最后一个输入捕获 3 事件 (IC3) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT3 和 CCDDAT3 只能读取。 当配置为输出模式时，寄存器 CCDAT3 和 CCDDAT3 是可读写的。

10.4.14 捕获/比较寄存器 4 (TIMx_CCDAT4)

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT4[15:0]															
rw															

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT4[15:0]	捕获/比较通道4的值 (Capture/Compare 4 value) <ul style="list-style-type: none"> CC4 通道配置为输出： CCDAT4 包含要与计数器 TIMx_CNT 比较的值，在 OC4 输出上发出信号。 如果未在 TIMx_CCMOD2.OC4PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。 CC4 通道配置为输入： CCDAT4 包含由最后一个输入捕获 4 事件 (IC4) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT4 和 CCDDAT4 只能读取。 当配置为输出模式时，寄存器 CCDAT4 和 CCDDAT4 是可读写的。

10.4.15 预分频器 (TIMx_PSC)

偏移地址：0x40

复位值：0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	PSC[15:0]	<p>预分频器的值（Prescaler value）</p> <p>计数器时钟 $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$。</p> <p>每次发生更新事件时，PSC 值都会加载到预分频器的影子寄存器中。</p>

10.4.16 自动重装载寄存器（TIMx_AR）

偏移地址:0x44

复位值: 0x0000 FFFF

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	AR[15:0]	<p>自动重装载的值（Auto-reload value）</p> <p>AR包含了将要装载入实际的自动重装载寄存器的值。详细参考9.3.1节：有关AR的更新和动作。</p> <p>当自动重装载的值为空时，计数器不工作。</p>

10.4.17 计数器（TIMx_CNT）

偏移地址: 0x48

复位值: 0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CNT[15:0]	计数器的值（Counter value）

10.4.18 通道1 滤波寄存器 (TIMx_C1FILT)

偏移地址: 0x64

复位值: 0x0000 0000

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

位域	名称	描述
31:30	Reserved	保留, 必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold), 最大 63: 有效逻辑电平的阈值。在采样窗口内, 如果逻辑高的数量大于或等于阈值, 则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值, 则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为: 最小值: 比最大毛刺大小的上限 (预分频时钟周期) 多 1 个预分频时钟周期, 并且需要大于窗口大小的一半。 例如, 如果毛刺大小为 $3.2 * (\text{预分频时钟周期})$, 则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ 最大值: 有效信号最小尺寸的底值 (在预分频时钟周期内), 需要小于窗口尺寸。 例如, 如果最小信号大小为 $3.2 * (\text{预分频时钟周期})$, 则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留, 必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值 (Window size), 最大 63: 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位, 最大索引为 63, 只能将窗口大小设置为 63。
16	FILTEN	滤波器使能 (Filter enable): 0: 滤波器禁能 1: 滤波器使能
15:8	Reserved	保留, 必须保持复位值
7:0	PSC[15:0]	用于配置滤波器采样时钟的预分频寄存器值 (Prescaler): 对于此过滤器, 它支持 128 分频 (8 位)。 时钟预分频器将系统时钟缩放到采样时钟。采样时钟决定两个采样点之间的距离。只有采样点的值有效才会考虑有效的逻辑电平计算。

10.4.19 通道2 滤波寄存器 (TIMx_C2FILT)

偏移地址: 0x68

复位值: 0x0000 0000

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

位域	名称	描述
31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： 最小值： 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ 最大值： 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 3.2*（预分频时钟周期），则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:8	Reserved	保留，必须保持复位值
7:0	PSC[15:0]	用于配置滤波器采样时钟的预分频寄存器值（Prescaler）： 对于此过滤器，它支持 128分频（8 位）。 时钟预分频器将系统时钟缩放到采样时钟。采样时钟决定两个采样点之间的距离。只有采样点的值有效才会考虑有效的逻辑电平计算。

10.4.20 通道3 滤波寄存器（TIMx_C3FILT）

偏移地址：0x6C

复位值：0x0000 0000

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

Reserved	PSC
rw	

位域	名称	描述
31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： 最小值： 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 $3.2 * (\text{预分频时钟周期})$ ，则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ 最大值： 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 $3.2 * (\text{预分频时钟周期})$ ，则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:8	Reserved	保留，必须保持复位值
7:0	PSC[15:0]	用于配置滤波器采样时钟的预分频寄存器值（Prescaler）： 对于此过滤器，它支持 128 分频（8 位）。 时钟预分频器将系统时钟缩放到采样时钟。采样时钟决定两个采样点之间的距离。只有采样点的值有效才会考虑有效的逻辑电平计算。

10.4.21 通道 4 滤波寄存器（TIMx_C4FILT）

偏移地址：0x70

复位值：0x0000 0000

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

位域	名称	描述
----	----	----

31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： 最小值： 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ 最大值： 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 3.2*（预分频时钟周期），则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:8	Reserved	保留，必须保持复位值
7:0	PSC[15:0]	用于配置滤波器采样时钟的预分频寄存器值（Prescaler）： 对于此过滤器，它支持 128分频（8 位）。 时钟预分频器将系统时钟缩放到采样时钟。采样时钟决定两个采样点之间的距离。只有采样点的值有效才会考虑有效的逻辑电平计算。

10.4.22 输入通道滤波输出寄存器（TIMx_FILTO）

偏移地址：0x74

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												C4FILTO	C3FILTO	C2FILTO	C1FILTO
												r	r	r	r

位域	名称	描述
31:4	Reserved	保留，必须保持复位值
3	C4FILTO	通道4滤波输出电平状态 0: 输出低电平； 1: 输出高电平；

2	C3FILTO	通道3滤波输出状态 0: 输出低电平; 1: 输出高电平;
1	C2FILTO	通道2滤波输出状态 0: 输出低电平; 1: 输出高电平;
0	C1FILTO	通道1滤波输出状态 0: 输出低电平; 1: 输出高电平;

10.4.23 输入通道选择寄存器 (TIMx_INSEL)

偏移地址: 0x78

复位值: 0x0000 0000

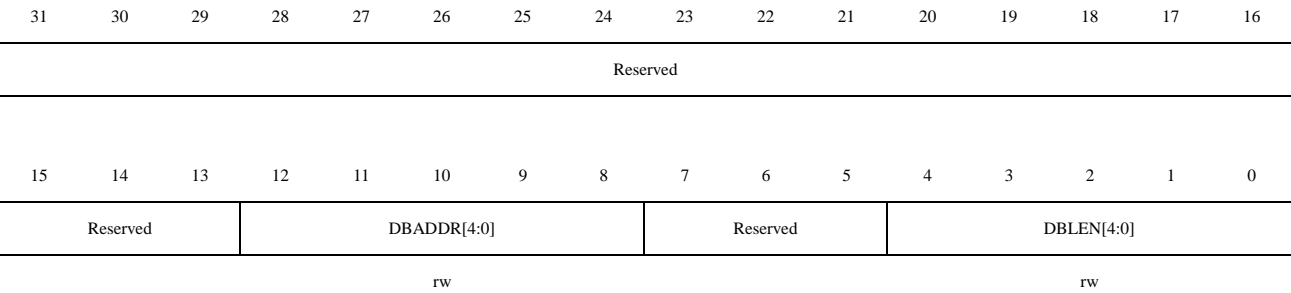
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				C4SEL	Reserved			C3SEL	Reserved			C2SEL	Reserved		C1SEL
rw					rw				rw				rw		

位域	名称	描述
31:13	Reserved	保留, 必须保持复位值
12	C4SEL	选择输入通道4的输入信号 0: 选择来自IOM的信号 1: 选择来自COMP的信号 注: TIM2/TIM3/TIM4/TIM5有效
11:9	Reserved	保留, 必须保持复位值
8	C3SEL	选择输入通道3的输入信号 0: 选择来自IOM的信号 1: 选择来自COMP的信号 注: TIM2/TIM3/TIM4/TIM5有效
7:5	Reserved	保留, 必须保持复位值
4	C2SEL	选择输入通道2的输入信号 0: 选择来自IOM的信号 1: 选择来自COMP的信号 注: TIM2/TIM3/TIM4/TIM5有效
3:1	Reserved	保留, 必须保持复位值
0	C1SEL	选择输入通道1的输入信号 0: 选择来自IOM的信号 1: 选择来自COMP的信号 注: TIM2/TIM3/TIM4/TIM5有效

10.4.24 DMA 控制寄存器（TIMx_DCTRL）

偏移地址：0x94

复位值：0x0000 0000

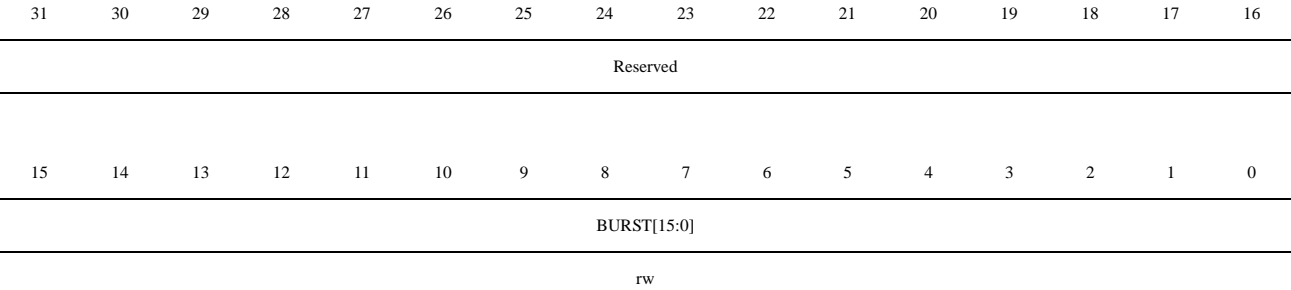


位域	名称	描述
31:13	Reserved	保留，必须保持复位值
12:8	DBADDR[4:0]	DMA基地址 （DMA base address） 该位字段定义 DMA 访问 TIMx_DADDR 寄存器的第一个地址。 当第一次通过 TIMx_DADDR 完成访问时，该位域指定您刚刚访问的地址。 然后第二次访问TIMx_DADDR，会访问到“DMA Base Address + 4”的地址 00000: TIMx_CTRL1, 00001: TIMx_CTRL2, 00010: TIMx_SMCTRL, 10001: TIMx_BKDT 10010: TIMx_DCTRL
7:5	Reserved	保留，必须保持复位值
4:0	DBLEN[4:0]	DMA连续传送长度 （DMA burst length） 该位字段定义 DMA 将访问（写入/读取）TIMx_DADDR 寄存器的次数。 00000: 1次传输 00001: 2次传输 00010: 3次传输 ... 10001: 18次传输

10.4.25 连续模式的DMA 地址 (TIMx_DADDR)

偏移地址: 0x98

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	BURST[15:0]	<p>DMA 访问缓冲区。</p> <p>当对该寄存器分配读或写操作时，将访问位于地址范围（DMA base address + DMA burst length × 4）的寄存器。</p> <p>DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL.DBADDR * 4;</p> <p>DMA burst len = TIMx_DCTRL.DBLEN + 1.</p> <p>例子：</p> <p>如果 TIMx_DCTRL.DBLEN = 0x3（4 次传输），TIMx_DCTRL.DBADDR = 0xD (TIMx_CCDAT1)，DMA 数据长度 = 半字，DMA 存储器地址 = SRAM 中的缓冲区地址，DMA 外设地址 = TIMx_DADDR 地址。</p> <p>当事件发生时，TIMx 将向 DMA 发送请求，并传输 4 次数据。</p> <p>第一次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CCDAT1 寄存器；</p> <p>第二次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CCDAT2 寄存器；</p> <p>.....</p> <p>第四次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CCDAT4 寄存器；</p>

11 基本定时器 (TIM6)

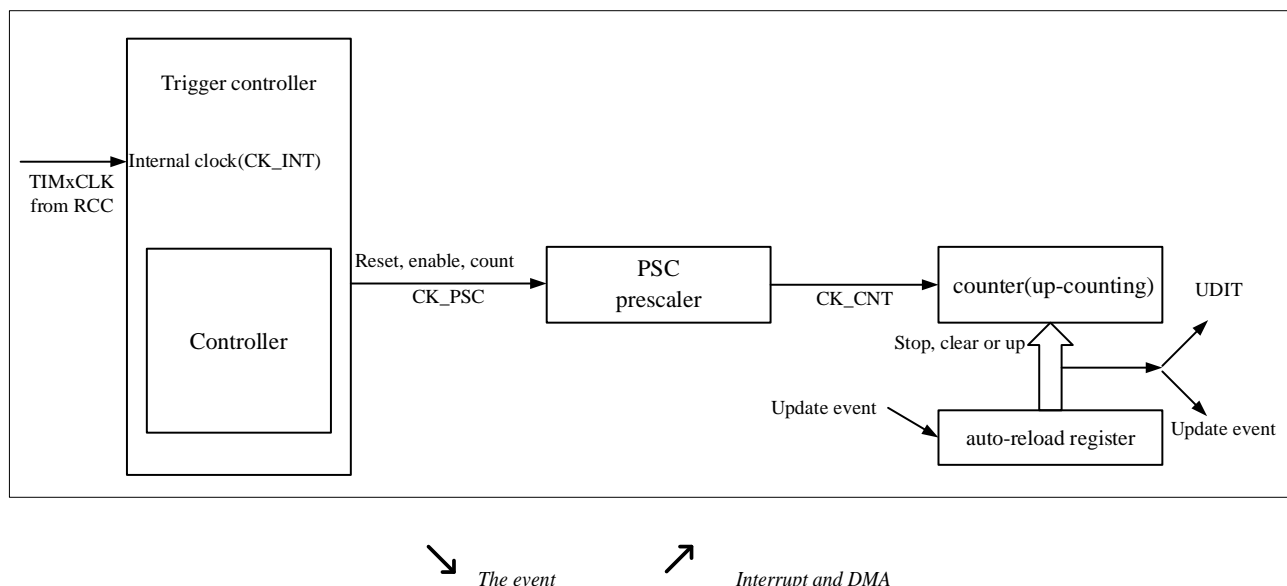
11.1 基本定时器简介

基本定时器 TIM6 包含一个 16 位自动装载计数器。

11.2 基本定时器主要特性

- 16 位自动重载向上计数计数器。
- 16 位可编程预分频器。（分频系数可配置为 1 到 65536 之间的任意值）
- 产生中断/DMA 的事件如下：
 - ◆ 更新事件
- 支持 STOP 模式唤醒：时钟源配置为 LSI 时，可通过更新中断（联接到 EXTI 20）唤醒 STOP 模式

图 11-1 TIMx 的框图 (x = 6)



11.3 基础定时器描述

11.3.1 时基单元

时基单元主要包括：预分频器、计数器、自动重载和重复计数器。当时基单元工作时，软件可以随时读写相应的寄存器（TIMx_PSC、TIMx_CNT 和 TIMx_AR）。

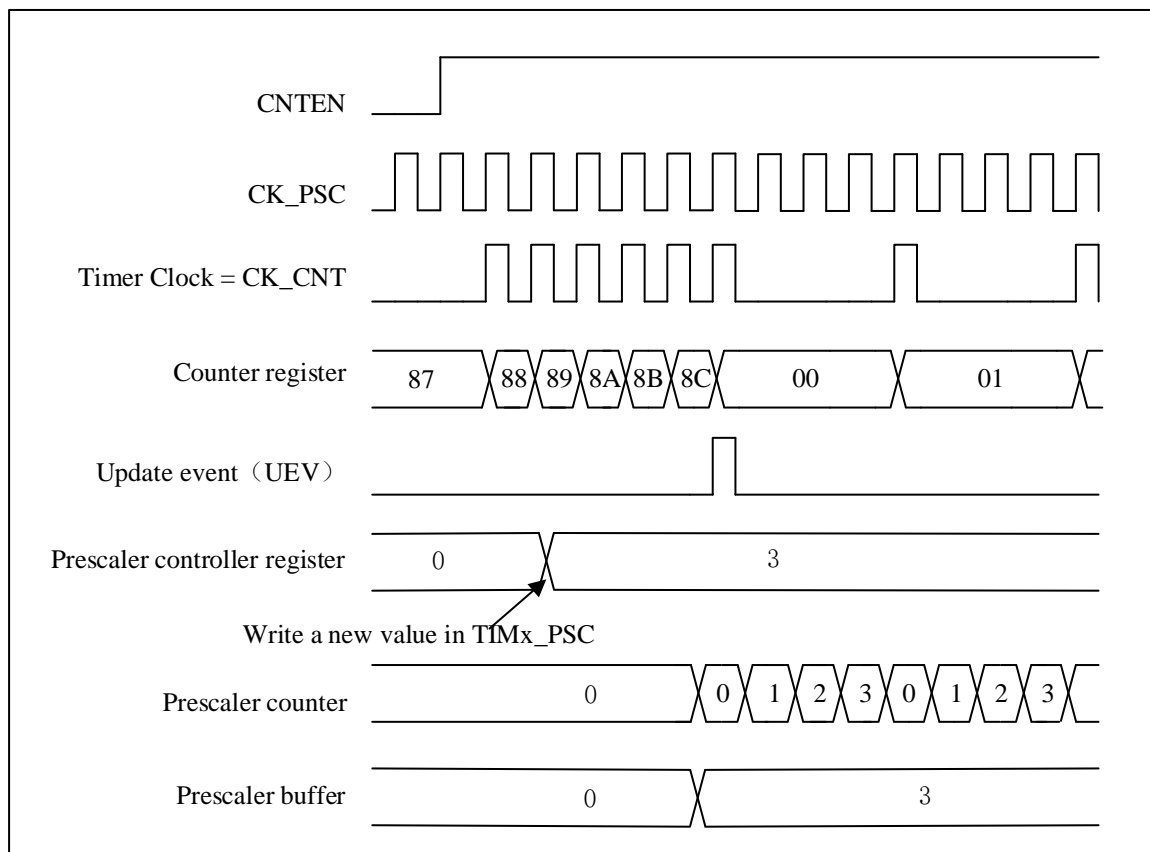
注意：时钟源配置为 LSI 时，TIMx_CNT 不支持写入。

根据自动重载预加载使能位(TIMx_CTRL1.ARPEN)的设置，预加载寄存器的值会立即或在每次更新事件 UEV 时传输到影子寄存器。TIMx_CTRL1.UPDIS=0 时，当计数器达到上溢条件或软件设置 TIMx_EVTGEN.UDGN 位，将生成更新事件。仅当 TIMx_CTRL1.CNTEN 位置位时，计数器 CK_CNT 才有效。计数器在 TIMx_CTRL1.CNTEN 位置位后一个时钟周期开始计数。

11.3.1.1 预分频器描述

TIMx_PSC 寄存器包含一个 16 位计数器，可用于将计数器时钟频率除以 1 到 65536 之间的任何因子。它可以在缓冲时动态更改。仅在下一次更新事件时才考虑预分频器值。

图 11-2 预分频器分频从 1 到 4 的计数器时序图



11.3.2 计数模式

11.3.2.1 向上计数模式

在向上计数模式下，计数器会从 0 计数到寄存器 TIMx_AR 的值，然后复位为 0。并产生计数器溢出事件。

如果设置了 TIMx_CTRL1.UPRS 位(选择更新请求)和 TIMx_EVTGEN.UDGN 位，则会生成更新事件(UEV)，并且不会由硬件设置 TIMx_STS.UDITF。因此，不会产生更新中断或更新 DMA 请求。此设置用于您想要清除计数器但不想产生更新中断的场景。

取决于 TIMx_CTRL1.UPRS 的配置，当更新事件发生时，TIMx_STS.UDITF 被设置，所有寄存器都被更新：

- 当 TIMx_CTRL1.ARPEN =1 时，使用预加载值(TIMx_AR)更新自动重载影子寄存器。
- 预分频器影子寄存器重新加载预加载值(TIMx_PSC)。

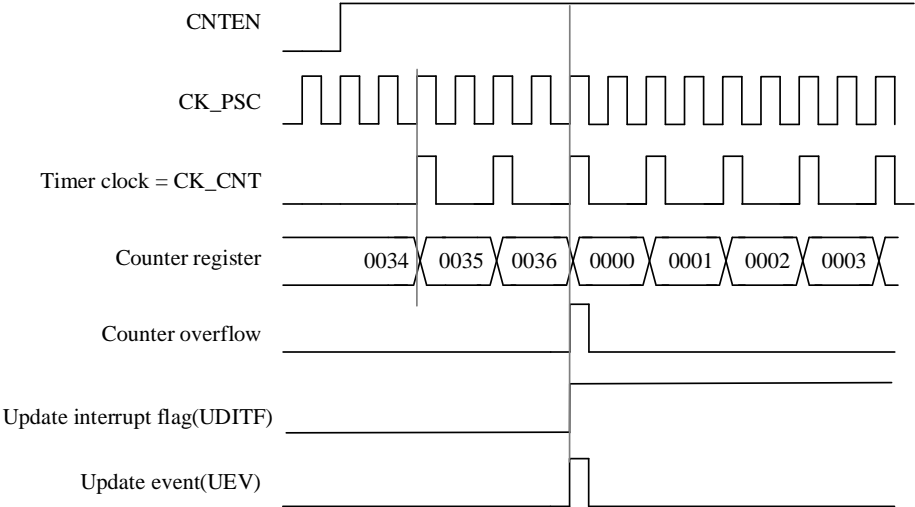
为避免在将新值写入预加载寄存器时更新影子寄存器，您可以通过设置 TIMx_CTRL1.UPDIS=1 来禁用更新。

当更新事件发生时，计数器仍将被清零，预分频器计数器也将设置为 0（但预分频器值将保持不变）。

下图显示了向上计数模式下不同除法因子的计数器行为和更新标志的一些示例。

图 11-3 向上计数时序图，内部时钟分频因子 = 2/N

Internal clock
divided by 2



Internal clock
divided by N

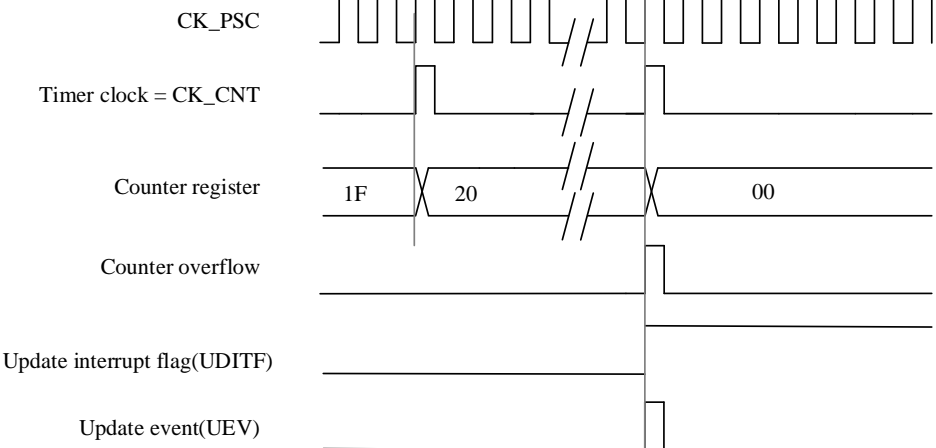
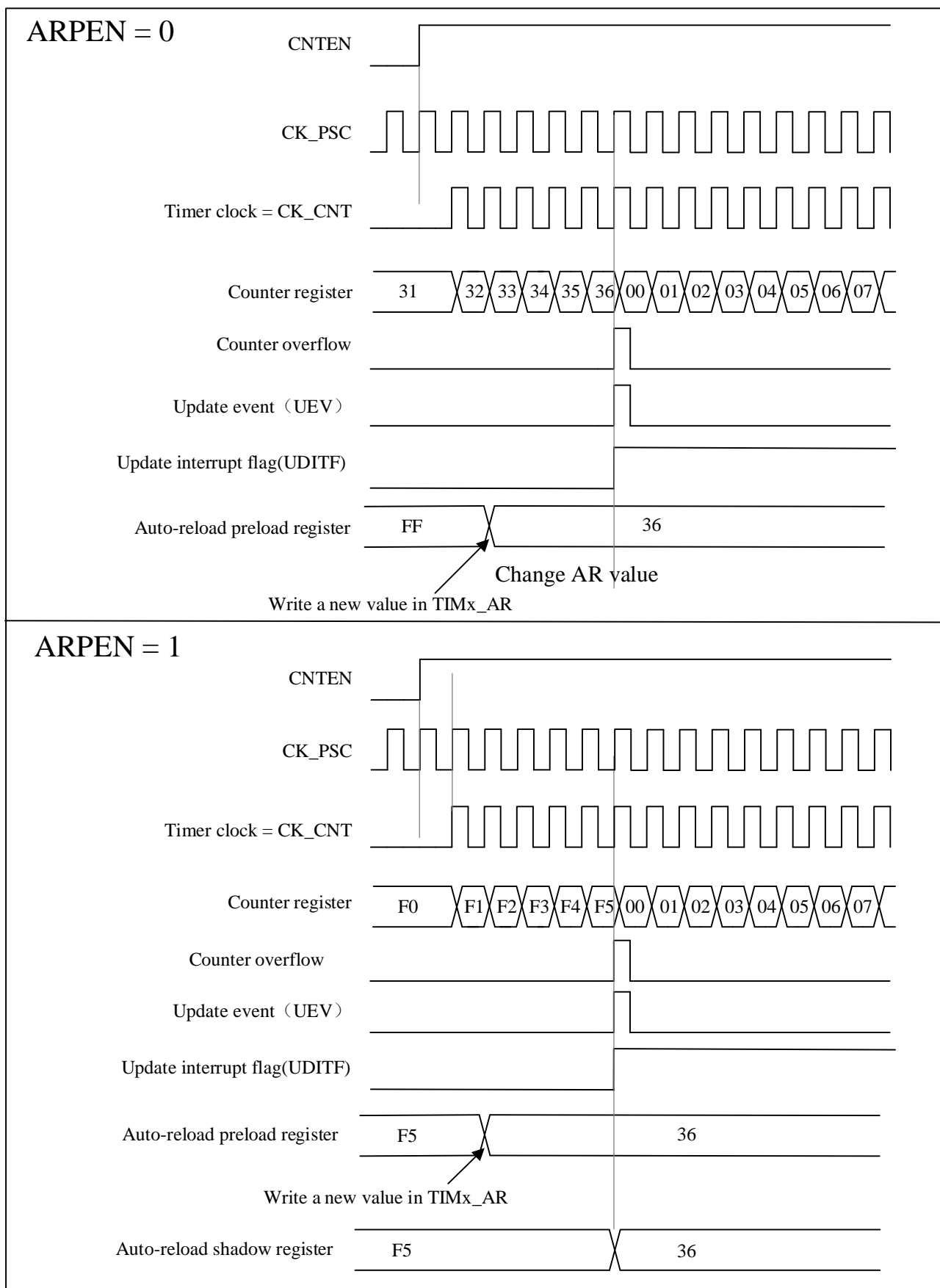


图 11-4 ARPEN=0/1 时向上计数、更新事件的时序图



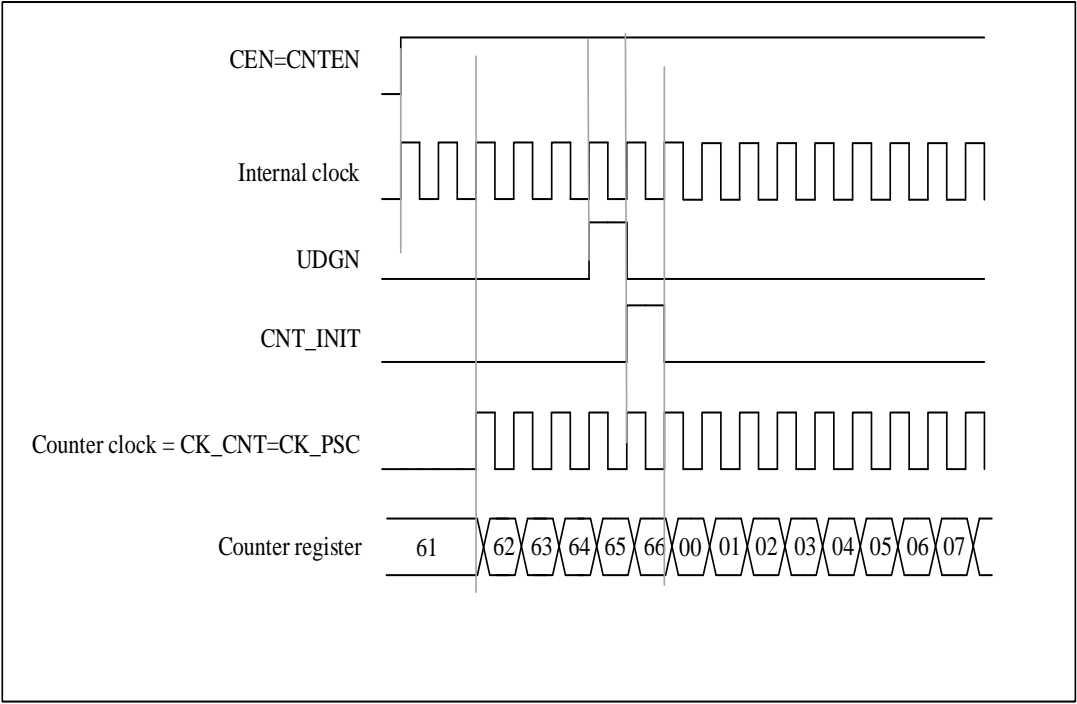
11.3.3 时钟选择

■ 定时器内部时钟：CK_INT

11.3.3.1 内部时钟源 (CK_INT)

前提是 TIMx_CTRL1.CNTEN 位由软件写为'1'，预分频器的时钟源由内部时钟 CK_INT 提供。

图 11-5 正常模式下的控制电路，内部时钟分频系数为 1



11.3.4 调试模式

当微控制器处于调试模式（Cortex-M0 内核停止）时，根据 DBG_CTRL.TIM6STP 位配置，TIM6 计数器可以继续正常工作或停止。有关详细信息，请参阅 3.4.9 节。

11.4 TIM6 寄存器描述

有关寄存器中使用的缩写，请参阅第 1.1 节

这些外设寄存器可以作为半字（16 位）或一个字（32 位）操作。

11.4.1 寄存器总览

表 11-1 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIMx_CTRL1	Reserved																						ARPEM	ONEPM	Reserved	UPDIS	UPRS	Reserved			CNTEN	
	Reset Value																							0	0								0
004h	TIMx_CTRL2	Reserved														MMSEL[3:0]						Reserved											

	Reset Value		0	0	0	0	
008h	TIMx_STS	Reserved	UDTIF				Reserved
	Reset Value		0				
00Ch	TIMx_EVTGEN	Reserved	UDGN				Reserved
	Reset Value		0				
014h	TIMx_DINTEN	Reserved	UDEN	Reserved	UIEN		Reserved
	Reset Value		0		0		
040h	TIMx_PSC	Reserved				PSC[15:0]	
	Reset Value					0	0
044h	TIMx_AR	Reserved				AR[15:0]	
	Reset Value					1	1
048h	TIMx_CNT	Reserved				CNT[15:0]	
	Reset Value					0	0

11.4.2 控制寄存器 1 (TIMx_CTRL1)

地址偏移: 0x00

复位值: 0x0000 0000

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

位域	名称	描述
31:10	Reserved	保留，必须保持复位值
9	ARPEN	自动重载预装载允许位（Auto-reload preload enable） 0: TIMx_AR 寄存器的影子寄存器禁用 1: TIMx_AR 寄存器的影子寄存器使能
8	ONEPM	单脉冲模式（One pulse mode） 0: 禁用单脉冲模式，发生更新事件时不影响计数器计数。 1: 使能单脉冲模式，计数器在下次更新事件发生时停止计数（清 TIMx_CTRL1.CNTEN 位）。
7:6	Reserved	保留，必须保持复位值
5	UPDIS	禁止更新（Update disable） 该位用于启用/禁用软件生成的更新事件（UEV）事件。 0: 启用UEV。如果满足以下条件之一，将生成UEV： – 计数器溢出 – TIMx_EVTGEN.UDGN 位被设置

		影子寄存器将使用预加载值进行更新。 1: UEV禁用。不生成更新事件，影子寄存器（AR、PSC）保持其值。如果设置了TIMx_EVTGEN.UDGN位，则重新初始化计数器和预分频器。
4	UPRS	更新请求源（Update request source） 该位用于通过软件选择 UEV 事件源。 0: 如果更新中断或 DMA 请求使能，以下任何事件都会产生更新中断或 DMA 请求： – 计数器溢出 – TIMx_EVTGEN.UDGN 位被设置 1: 如果更新中断或 DMA 请求使能，只有计数器溢出会产生更新中断或 DMA 请求
3:1	Reserved	保留，必须保持复位值
0	CNTEN	使能计数器（Counter enable） 0: 禁止计数器； 1: 使能计数器。

11.4.3 控制寄存器 2（TIMx_CTRL2）

偏移地址：0x04

复位值：0x0000 0000

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

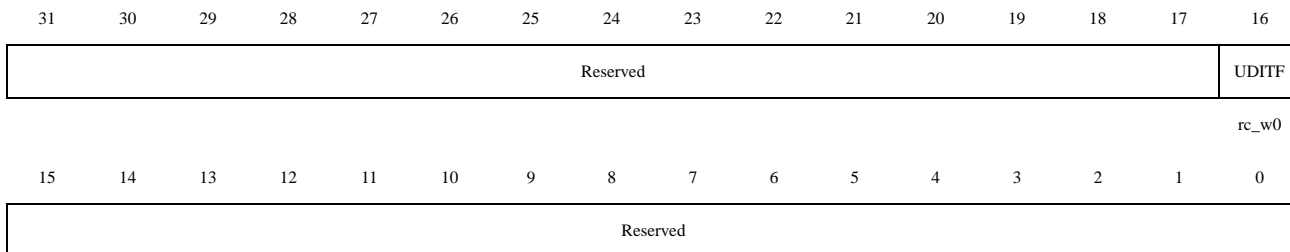
rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	MMSEL[3:0]	主模式选择 这 4 位用于选择在主模式下发送到从定时器的同步信息（TRGO）。可能的组合如下： X000: 复位 - 当 TIMx_EVTGEN.UDGN 置位或从模式控制器产生复位时，将出现 TRGO 脉冲。在后一种情况下，TRGO 上的信号与实际复位相比有所延迟。 X001: 使能 - TIMx_CTRL1.CNTEN 位用作触发输出（TRGO）。有时需要同时启动多个定时器或者在一段时间内开启从定时器。 当 TIMx_CTRL1.CNTEN 位置位或门控模式下的触发输入为高电平时，计数器使能信号置位。 当计数器使能信号由触发输入控制时，TRGO 上有一个延迟，除非选择了主/从模式（参见 TIMx_SMCTRL.MSMD 位的说明）。 X010: 更新 - 选择更新事件作为触发输出（TRGO）。例如，主定时器时钟可用作从定时器预分频器。
11:0	Reserved	保留，必须保持复位值

11.4.4 状态寄存器 (TIMx_STS)

地址偏移: 0x08

复位值: 0x0000 0000

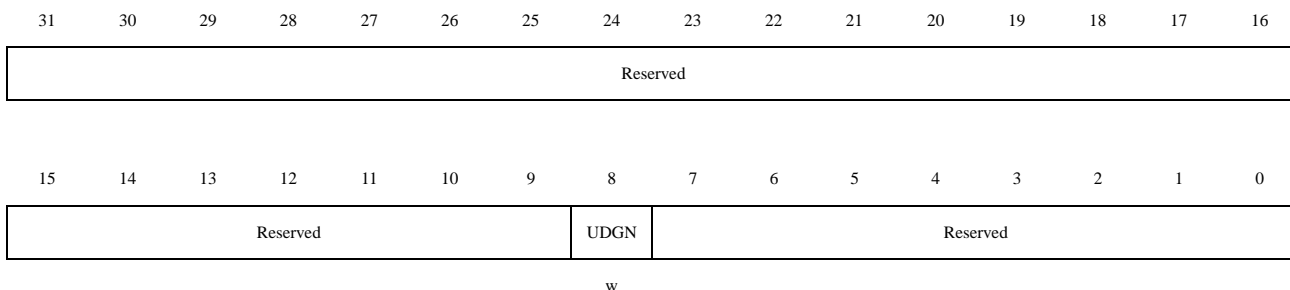


位域	名称	描述
31:17	Reserved	保留，必须保持复位值
16	UDITF	更新中断标志（Update interrupt flag） 当在以下条件下发生更新事件时，该位由硬件设置： <ul style="list-style-type: none"> – 当 TIMx_CTRL1.UPDIS = 0 且计数器值溢出时。 – 当 TIMx_CTRL1.UPRS = 0 时，TIMx_CTRL1.UPDIS = 0，并通过软件设置 TIMx_EVTGEN.UDGN 位以重新初始化 CNT。 该位由软件清零。 0：未发生更新事件 1：发生更新中断
15:0	Reserved	保留，必须保持复位值

11.4.5 事件产生寄存器 (TIMx_EVTGEN)

地址偏移: 0x0C

复位值: 0 x0000 0000



w

位域	名称	描述
31:9	Reserved	保留，必须保持复位值
8	UDGN	产生更新事件（Update generation）

位域	名称	描述
		软件可以设置该位来更新配置寄存器的值，硬件会自动清除它。 0：无效果。 1：定时器计数器将重新启动，所有影子寄存器将被更新。 它也将重新启动预分频器计数器。
7:0	Reserved	保留，必须保持复位值

11.4.6 DMA/中断使能寄存器 (TIMx_DINTEN)

地址偏移: 0x14

复位值: 0x0000 0000

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

位域	名称	描述
31:20	Reserved	保留，必须保持复位值
19	UDEN	更新DMA请求使能（Update DMA request enable） 0：禁止更新DMA请求 1：使能更新DMA请求
18:17	Reserved	保留，必须保持复位值
16	UIEN	更新中断使能（Update interrupt enable） 0：禁止更新中断 1：使能更新中断
15:0	Reserved	保留，必须保持复位值

11.4.7 预分频器 (TIMx_PSC)

地址偏移: 0x40

复位值: 0x0000 0000

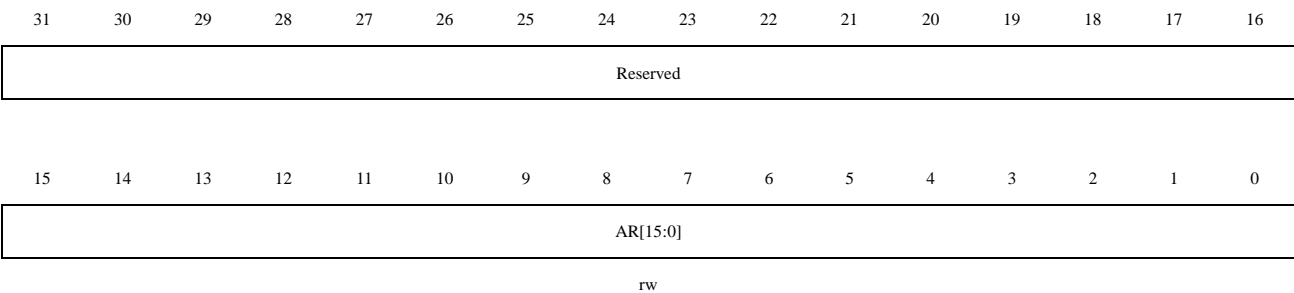
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	PSC[15:0]	预分频器数值（Prescaler value） PSC寄存器值将在更新事件时更新到预分频器寄存器。计数器时钟频率是输入时钟分频PSC+1。

11.4.8 自动重载寄存器 (TIMx_AR)

地址偏移: 0x44

复位值: 0x0000 FFFF

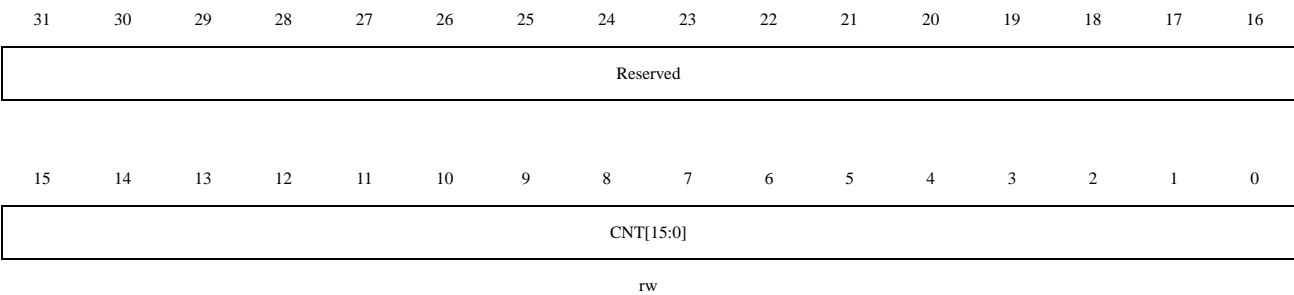


位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	AR[15:0]	自动重载数值（Auto-reload value） 这些位定义将加载到实际自动重载寄存器中的值。 有关详细信息，请参阅 11.3.1。 当TIMx_AR.AR [15:0]值为空时，计数器不工作。

11.4.9 计数器 (TIMx_CNT)

地址偏移: 0x48

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CNT[15:0]	计数器数值（Counter value）

12 独立看门狗（IWDG）

12.1 简介

N32G05x 内置独立看门狗（IWDG）和窗口看门狗（WWDG）定时器，解决软件错误导致的问题。看门狗定时器使用非常灵活，提高了系统的安全性和定时控制的准确性。

独立看门狗（IWDG）由运行在 32KHz 的低速内部时钟（LSI 时钟）驱动，在死循环事件或 MCU 卡死发生时，它仍然可以运行。这可以提供更高的安全级别、定时精度和看门狗的灵活性。它可以通过重置来解决由于软件故障引起的系统故障。IWDG 最适合需要看门狗在主应用程序之外作为完全独立进程运行但时序精度限制较低的应用程序。

当电源控制寄存器 PWR_CTRL.IWDGRSTEN 位置‘1’，IWDG 计数器达到 0 时，会产生系统复位（若该位置‘0’，IWDG 会计数但不产生复位）。IWDG 复位也可用于低功耗唤醒。

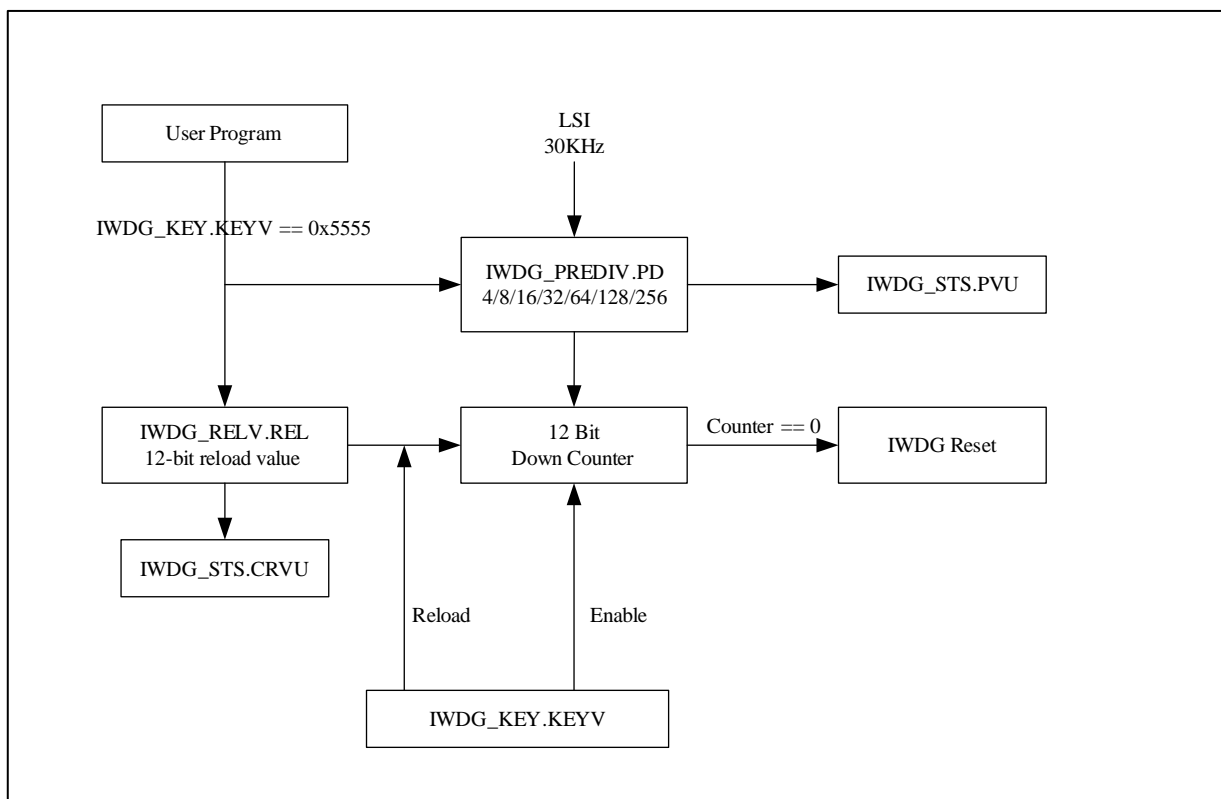
注：本章基于系统默认值 IWDGRSTEN=1 讨论。

12.2 主要特性

- 独立运行的 12 位递减计数器
- RC 振荡器提供独立时钟源，在 SLEEP 和 STOP 模式下仍能正常工作
- 可以匹配复位和低功耗唤醒
- 当递减计数器达到 0x000 时，系统复位（如果激活了看门狗）
- 支持计数器冻结

12.3 功能描述

图 12-1 独立看门狗模块的功能框图



要启用 IWDG，我们需要将 0xCCCC 写入 IWDG_KEY.KEYV[15:0]位，计数器开始递减计数。当计数器计数到 0x000 时，它会向 MCU 产生一个复位信号（IWDG_RESET）。除此之外，只要在复位前将 0xAAAA（重载请求）写入 IWDG_KEY.KEYV[15:0]位，计数器值就会设置为 IWDG_RELV.REL[11:0]位中的重载值并防止看门狗复位整个设备。

如果通过选项字节使能“硬件看门狗定时器”功能，则看门狗将在系统上电后自动开始运行并产生系统复位，除非软件在计数器到达‘0’之前重新加载计数器。

12.3.1 寄存器访问保护

IWDG_PREDIV 和 IWDG_RELV 寄存器具有写保护功能。在修改这两个寄存器数据之前，必须先配置 IWDG_KEY 寄存器为 0x5555。配置成其他任何数据，都将再次启动寄存器写保护。IWDG_STS.PVU 指示预分频器值更新是否正在进行。IWDG_STS.CRVU 指示 IWDG 是否正在更新重载值。当预分频器值和/或重载值更新时，硬件设置 IWDG_STS.PVU 位和/或 IWDG_STS.CRVU 位。预分频器值和/或重载值更新完成后，硬件清除 IWDG_STS.PVU 位和/或 IWDG_STS.CRVU 位。

重载操作（IWDG_KEY 配置 0xAAAA）也会启动写保护功能。

12.3.2 调试模式

在调试模式下（Cortex-M0 内核停止），IWDG 计数器将继续正常工作或停止，具体取决于调试模块中的 DBG_CTRL.IWDG_STOP 位。如果该位设置为“1”，则计数器停止。该位为“0”时，计数器正常工作。。详

见 3.3.2 外设调试支持章节。

12.3.3 IWDG 冻结

一旦 IWDG 使能（无论是硬件还是软件），除非生成系统复位或通过将 IWDG 冻结，否则 IWDG 不会停止计数。

在 RUN 模式下将 0x4567 写入 IWDG_KEY.KEYV[15:0]位来配置运行时冻结，将 0x89AB 写入 IWDG_KEY.KEYV[15:0]位可以配置运行时解冻。

用户还可以在 SLEEP 或者 STOP 模式下配置 IWDG 冻结，IWDG 在 RUN 模式下运行时，将 0xDDDD 写入 IWDG_FREEZE.FREEZE[15:0]寄存器，当 MCU 进入 SLEEP 或者 STOP 模式时，IWDG 会自动冻结，当再次将 0xDDDD 写入 IWDG_FREEZE.FREEZE[15:0]寄存器，当进入 SLEEP 或者 STOP 模式时，IWDG 自动解冻，计数器恢复工作。

当 IWDG 开启时，它将强制 LSI 时钟开启。

12.4 用户界面

IWDG 模块用户界面包含 4 个寄存器：密钥寄存器（IWDG_KEY）、预分频寄存器（IWDG_PREDIV）、重装载寄存器（IWDG_RELV）和状态寄存器（IWDG_STS）。

12.4.1 操作流程

当 IWDG 从软件（将 0xAAAA 写入 IWDG_KEY.KEYV[15:0]位）或硬件（清零 FLASH_OB.WDG_SW 位）复位启用时。它从 0xFFF 开始递减计数。向下计数间隙由预分频 LSI 时钟确定。重新加载计数器后，新一轮递减计数器的值将从 IWDG_RELV.REL[11:0]中的值开始，而不是 0xFFF。

程序正常运行时，软件需要在计数器到达 0 前喂狗，开始新一轮的递减计数。当计数器达到 0 时，表示程序故障。IWDG 在这种情况下产生复位信号。

如果用户想要配置 IWDG 预分频和重装载值寄存器，需要先将 0x5555 写入 IWDG_KEY.KEYV[15:0]。然后确认 IWDG_STS.CRVU 位和 IWDG_STS.PVU 位。IWDG_STS.CRVU 位指示重装载值更新正在进行，IWDG_STS.PVU 表示预分频值更新正在进行。只有当这两位为 0 时，用户才能更新相应的值。当更新正在进行时，硬件将相应位设置为 1。此时，读取 IWDG_PREDIV.PD[2:0]或 IWDG_RELV.REL[11:0]无效，因为数据需要同步到 LSI 时钟域。从 IWDG_PREDIV.PD[2:0]或 IWDG_RELV.REL[11:0]读取的值将在硬件清除 IWDG_STS.PVU 位或 IWDG_STS.CRVU 位后才有效。

如果应用程序使用多个重装载值或预分频值，则必须等到 IWDG_STS.CRVU 位复位后才能更改重装载值，IWDG_STS.PVU 位复位后才能更改预分频值。但是，在更新预分频值和重装载值后，或只更新预分频值后，或只更新重装载值后，无需等到 IWDG_STS.CRVU 位或 IWDG_STS.PVU 位复位后才能继续执行代码（即使在进低功耗模式的情况下，写入操作也会被考虑并完成）。

预分频寄存器和重装载寄存器控制产生复位的时间，如表 12-1。

表 12-1 IWDG 计数最大和最小复位时间

预分频系数	PD [2:0]	最短时间（ms） REL [11:0]=0x000	最长时间（ms） REL [11:0]=0xFFFF
/4	000	0.125	512

/8	001	0.25	1024
/16	010	0.5	2048
/32	011	1.0	4096
/64	100	2	8192
/128	101	4	16384
/256	11x	8	32768

12.4.2 IWDG 配置流程

软件配置流程：

1. 将 0x5555 写入 IWDG_KEY.KEYV[15:0]位以启用对 IWDG_PREDIV 和 IWDG_RELV 寄存器的写访问；
2. 检查 IWDG_STS.PVU 位或 IWDG_STS.CRVU 位，如果为 0，则继续下一步；
3. 配置 IWDG_PREDIV.PD[2:0]位以选择预分频值；
4. 配置 IWDG_RELV.REL[11:0]位重装载值；
5. 将 0xAAAA 写入 IWDG_KEY.KEYV[15:0]位，用重装载值更新计数器；
6. 通过软件或硬件将 0xCCCC 写入 IWDG_KEY.KEYV[15:0]位来启用看门狗。

如果用户想改变预分频值和重装载值，重复步骤 1~5。如果没有，只需按照第 5 步喂狗。

12.5 IWDG 寄存器

12.5.1 IWDG 寄存器总览

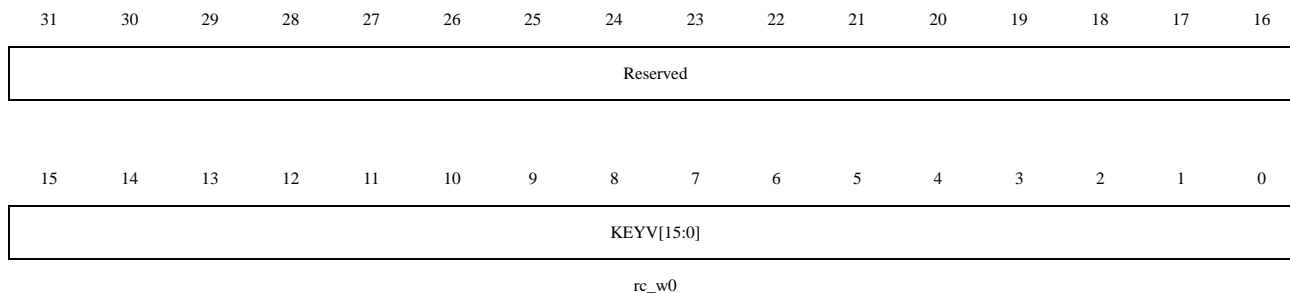
表 12-2 IWDG 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	IWDG_KEY	Reserved																KEYV[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	IWDG_PREDIV	Reserved																												PD[2:0]			
	Reset Value																													0	0	0	
008h	IWDG_RELV	Reserved																REL[11:0]															
	Reset Value																	1	1	1	1	1	1	1	1	1	1	1	1	1			
00Ch	IWDG_STS	Reserved																												LPFRE	RUNF	CRVU	PVU
	Reset Value																													0	0	0	0
010h	IWDG_FREEZE	Reserved																LPFRV[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.5.2 IWDG 密钥寄存器（IWDG_KEY）

偏移地址：0x00

复位值：0x00000000

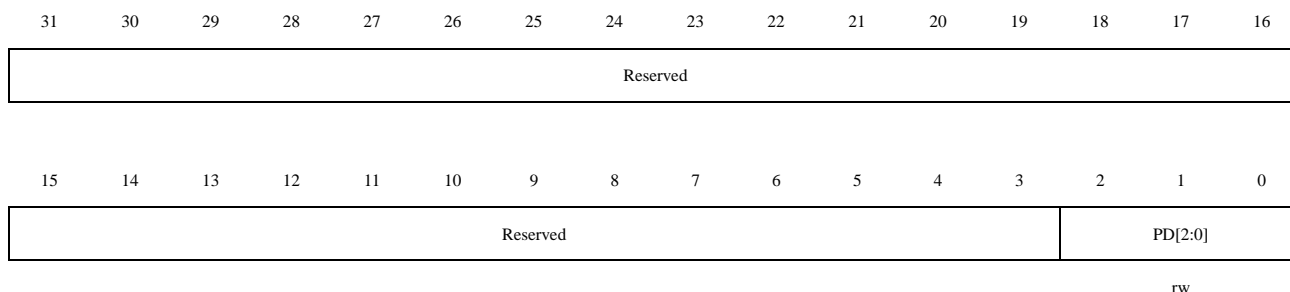


位域	名称	描述
31:16	保留	保留，必须保持复位值。
15:0	KEYV[15:0]	密钥寄存器：只有特定的值才能发挥特定的作用 0xCCCC：启动看门狗计数器，如果硬件看门狗使能则无效，（如果选择了硬件看门狗，则不受该命令字限制） 0xAAAA：用 IWDG_RELV 寄存器中的 REL 值重新加载计数器以防止复位 0x5555：禁用 IWDG_PREDIV 和 IWDG_RELV 寄存器的写保护 0x4567：IWDG 冻结，RUN 模式下停止 IWDG 计数器 0x89AB：IWDG 解冻，计数器恢复计数

12.5.3 IWDG 预分频寄存器（IWDG_PREDIV）

偏移地址：0x04

复位值：0x00000000



位域	名称	描述
31:3	保留	保留，必须保持复位值。
2:0	PD[2:0]	预分频因子 当 IWDG_KEY.KEYV[15:0]不是 0x5555 时具有写访问保护。IWDG_STS.PVU 位必须为 0，否则 PD[2:0]值无法更改。分频系数如下： 000：预分频因子=4 001：预分频因子=8 010：预分频因子=16 011：预分频因子=32 100：预分频因子=64

位域	名称	描述
		101: 预分频因子=128 其他: 预分频因子=256 <i>注意: 读取该寄存器将返回来自 VDD 电压域的预分频值。如果正在进行写操作, 则回读值可能无效。因此, 读取值仅在 IWDG_STS.PVU 位为 0 时有效。</i>

12.5.4 IWDG 重装载寄存器 (IWDG_RELV)

偏移地址: 0x08

复位值: 0x00000FFF

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

rw

位域	名称	描述
31:12	保留	保留, 必须保持复位值。
11:0	REL[11:0]	看门狗计数器重装载值。 带写保护。定义看门狗计数器的重装载值, 每次将 0xAAAA 写入 IWDG_KEY.KEYV[15:0]位时将其加载到计数器。然后计数器从该值开始倒计时。看门狗超时周期可以根据这个重装载值和时钟预分频值计算, 参考表 12-1。 该寄存器只能在 IWDG_STS.CRVU 位为 0 时修改。 <i>注意: 读取该寄存器将返回来自 VDD 电压域的重装载值。如果正在进行写操作, 则回读值可能无效。因此, 读取值仅在 IWDG_STS.CRVU 位为 0 时有效。</i>

12.5.5 IWDG 状态寄存器 (IWDG_STS)

偏移地址: 0x0C

复位值: 0x00000000

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

r r r r

位域	名称	描述
31:4	保留	保留，必须保持复位值。
3	LPFREF	低功耗模式下 IWDG 冻结标志
2	RUNFREF	RUN 模式下 IWDG 冻结标志
1	CRVU	看门狗重装载值更新 重装载值更新：该位表示正在更新重装载值。硬件置位，硬件清零。软件只能在 IWDG_KEY.KEYV[15:0]位的值为 0x5555 且该位为 0 时尝试更改 IWDG_RELV.REL[11:0]的值。
0	PVU	看门狗预分频值更新 预分频值更新：该位表示正在更新预分频值。硬件置位，硬件清零。软件只能在 IWDG_KEY.KEYV[15:0]位的值为 0x5555 且该位为 0 时尝试更改 IWDG_PREDIV.PD[2:0]的值。

12.5.6 IWDG 冻结寄存器（IWDG_FREEZE）

偏移地址：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPRERV[15:0]															
rc_w0															

位域	名称	描述
31:16	保留	保留，必须保持复位值。
15:0	LPRERV	低功耗模式下 IWDG 冻结/解冻 0xDDDD：当 MCU 进入 SLEEP 或者 STOP 模式时使能 IWDG 冻结； 再次写 0xDDDD，当 MCU 进入 SLEEP 或者 STOP 模式时失能 IWDG 冻结

13 窗口看门狗（WWDG）

13.1 简介

窗口看门狗（WWDG）的时钟是 APB1 时钟频率除以 4096 得到的，通过时间窗口的配置来检测程序运行是否异常。因此，WWDG 适用于精确定时，常用于监控因外部干扰或无法预见的逻辑条件导致应用程序偏离其正常操作顺序的软件故障。当 WWDG 递减计数器在达到窗口寄存器值之前或 WWDG_CTRL.T6 位变为 0 之后刷新时，发生系统复位。

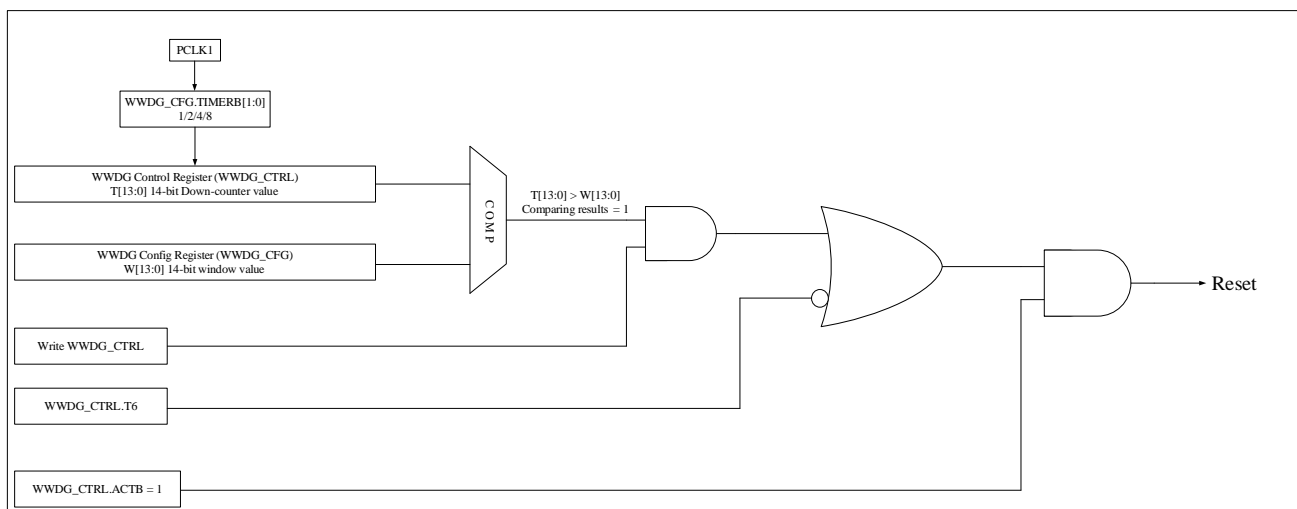
13.2 主要特征

- 14 位独立递减计数器可编程
- WWDG 启用后，在以下情况下会发生复位
 - ◆ 递减计数器的值小于 0x40
 - ◆ 当递减后的计数器值大于窗口寄存器的值时，重新加载
- 提前唤醒中断：如果看门狗启动并且中断使能，当计数值达到 0x40 时会产生唤醒中断（WWDG_CFG.EWINT）

13.3 功能描述

如果看门狗被激活（WWDG_CTRL.ACTB），当 14 位（WWDG_CTRL.T[13:0]）递减计数器到达 0x3F（WWDG_CTRL.T6 位清零），或者软件重新加载计数器时计数器值大于窗口寄存器的值，将产生系统复位。为了避免系统复位，软件在正常运行时必须定期刷新窗口中的计数器值。

图 13-1 窗口看门狗功能框图



设置 WWDG_CTRL.ACTB 位以启用看门狗，此后，WWDG 将保持打开状态，直到发生复位。14 位递减计数器独立运行，无论 WWDG 是否使能，计数器都会持续递减计数。因此，需要将高 8 位 MSB(WWDG_CTRL.T[13:6]) 中的某一位设置为 1，以防止在启用后立即复位。由时钟 APB1 和 WWDG_CFG.TIMERB[1:0] 位设置的预分频器值决定了计数器的递减速度。WWDG_CFG.W[13:0] 位设置窗

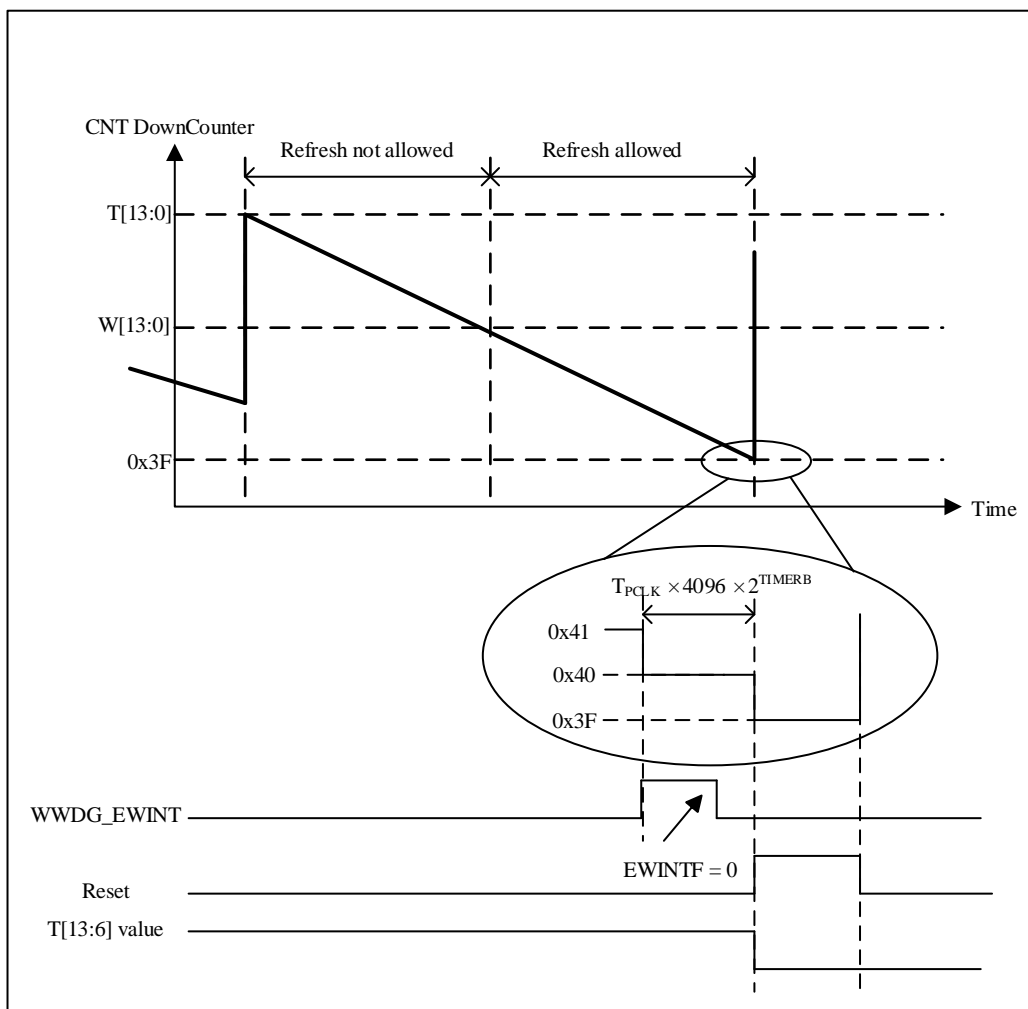
口的上限。

当递减计数器在达到窗口寄存器值之前或 WWDG_CTRL.T6 位变为 0 之后刷新，将产生系统复位。图 13-2 描述了窗口寄存器的工作过程。

设置 WWDG_CFG.EWINT 位以启用提前唤醒中断。当倒数计数器到达 0x40 时，将产生中断。您可以分析软件故障的原因或将重要数据保存在相应的中断服务程序（ISR）中，并重新加载计数器以防止 WWDG 复位。将“0”写入 WWDG_STS.EWINTF 位以清除中断。

13.4 刷新看门狗和中断产生的时序

图 13-2 WWDG 的刷新窗口和中断时序



看门狗刷新窗口在 WWDG_CFG.W[13:0] 值（最大值 0x3FFF）和 0x3F 之间，在此窗口外刷新将向 MCU 生成复位请求。计数器使用分频后的 APB 时钟从 0x3FFF 向下计数到 0x3F，最大计数时间和最小计数时间如图 13-1 所示（假设 APB 时钟为 32MHz），计算公式为：

$$T_{WWDG} = T_{PCLK1} \times 4096 \times 2^{TIMERB} \times (T[13:0] - 0X3F + 1)$$

其中：

T_{WWDG}:WWDG 超时

T_{PCLK1}:APB1 时钟间隔，单位为：ms

PCLK1=32MHz 时的最小-最大超时时长

表 13-1 WWDG 的最大和最小计数时间

TIMERB	最大超时 (ms)	最小超时 (ms)
0	2089	0.128
1	4178	0.256
2	8356	0.512
3	16712	1.024

13.5 调试模式

在调试模式下（Cortex-M0 内核停止），WWDG 计数器将继续正常工作或停止，具体取决于调试模块中的 DBG_CTRL.WWDG_STOP 位。如果该位设置为“1”，则计数器停止。该位为“0”时，计数器正常工作。详见 3.3.2 外设调试支持章节。

13.6 用户界面

13.6.1 WWDG 配置流程

1. 配置 RCC_APB1PCLKEN.WWDGEN[11]位使能 WWDG 模块的时钟
2. 软件设置 WWDG_CFG.TIMERB[15:14]位来配置 WWDG 的预分频因子
3. 软件配置 WWDG_CTRL.T[13:0]位，设置计数器的起始值。需要将高 8MSB 位(WWDG_CTRL.T[13:6])中的某一位设置为 1，以防止在启用后立即复位
4. 配置 WWDG_CFG.W[13:0]位配置上边界窗口值
5. 设置 WWDG_CTRL.ACTB[14]位使能 WWDG
6. 软件操作 WWDG_STS.EWINTF[0]位清除唤醒中断标志
7. 配置 WWDG_CFG.EWINT[16]位使能提前唤醒中断

13.7 WWDG 寄存器

13.7.1 WWDG 寄存器总览

表 13-2 WWDG 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	WWDG_CTRL	Reserved																	ACTB	T[13:0]														
	Reset value																			0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	WWDG_CFG	Reserved															EWINT	TIMERB[1:0]	W[13:0]															
	Reset value																		0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	WWDG_STS	Reserved																															EWINTF	
	Reset value																																	0

13.7.2 WWDG 控制寄存器（WWDG_CTRL）

偏移地址：0x00

复位值：0x00003FFF

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

位域	名称	描述
31:15	保留	保留，必须保持复位值。
14	ACTB	激活位 当 ACTB=1 时，看门狗可以产生复位。该位由软件置位，仅在复位后由硬件清零。当 ACTB=1 时，看门狗可以产生复位。 0：禁用看门狗 1：启用看门狗
13:0	T[13:0]	这些位包含看门狗计数器的值。它每(4096x2 ^{TIMERRB})个 PCLK1 周期递减。当它从 0x40 翻转到 0x3F（T6 清零）时，会产生一个复位。

13.7.3 WWDG 配置寄存器（WWDG_CFG）

偏移地址：0x04

复位值：0x00003FFF

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

位域	名称	描述
31:17	保留	保留，必须保持复位值。
16	EWINT	提前唤醒中断 设置后，只要计数器达到值 0x40，就会发生中断。此中断仅在复位后由硬件清除。
15:14	TIMERB[1:0]	时基 预分频器的时基可以修改如下： 00：CK 计数器时钟（PCLK1 除以 4096）除以 1 01：CK 计数器时钟（PCLK1 除以 4096）除以 2 10：CK 计数器时钟（PCLK1 除以 4096）除以 4 11：CK 计数器时钟（PCLK1 除以 4096）除以 8
13:0	W[13:0]	14 位窗口值 这些位包含要与递减计数器比较的窗口值。

13.7.4 WWDG 状态寄存器（WWDG_STS）

偏移地址：0x08

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EWINTF
rc_w0															

位域	名称	描述
31:1	保留	保留，必须保持复位值。

位域	名称	描述
0	EWINTF	提前唤醒中断标志 当计数器达到值 0x40 时，该位由硬件设置。它必须由软件通过写入“0”来清零。写入“1”无效。如果中断未使能，该位也会置位。

14 模拟数字转换（ADC）

14.1 简述

12 位 ADC 是使用逐次逼近的高速模数转换器。共有 17 个通道，可测 15 个外部和 2 个内部信号源。各个通道的 A/D 转换通道可以在单次、连续、扫描模式下执行。ADC 转换值存储（左对齐/右对齐）在 16 位数据寄存器中。可以通过模拟看门狗检测输入电压是否在用户定义的高/低阈值内，并且 ADC 的输入时钟的最大频率为 24MHz。

14.2 ADC 主要特征

- 支持 1 个 ADC，支持单端输入，最多可测量 15 个外部和 2 个内部源
- 支持 12 位分辨率，最高采样速率 1MSPS
- ADC 时钟源分为工作时钟源和计时时钟源
 - ◆ HIS、PLL、AHB 作为 ADC_CLK 工作时钟源，最高到 24M
 - ◆ HIS 或 HSE 作为 ADC_1MCLK 计时时钟源，用于内部计时功能，频率必须配置成 1MHz
- 支持定时器触发采样
- 当转换完成或者模拟看门狗事件可触发中断
- 支持 2 种转换模式
 - ◆ 单次转换
 - ◆ 连续转换
- 扫描模式最大支持任意 5 个通道，每个通道有一个独立的结果数据寄存器 buffer
- 所有通道采样间隔可以统一编程
- 可以外部触发规则转换。
- ADC 的工作电压在 2.4V 到 5.5V 之间。
- ADC 支持转换的电压在 0 和 V_{DD} 之间。

14.3 ADC 功能描述

下图为一个 ADC 模块的框图，表 14-1 为 ADC 引脚的说明。

图 14-1 ADC 框图

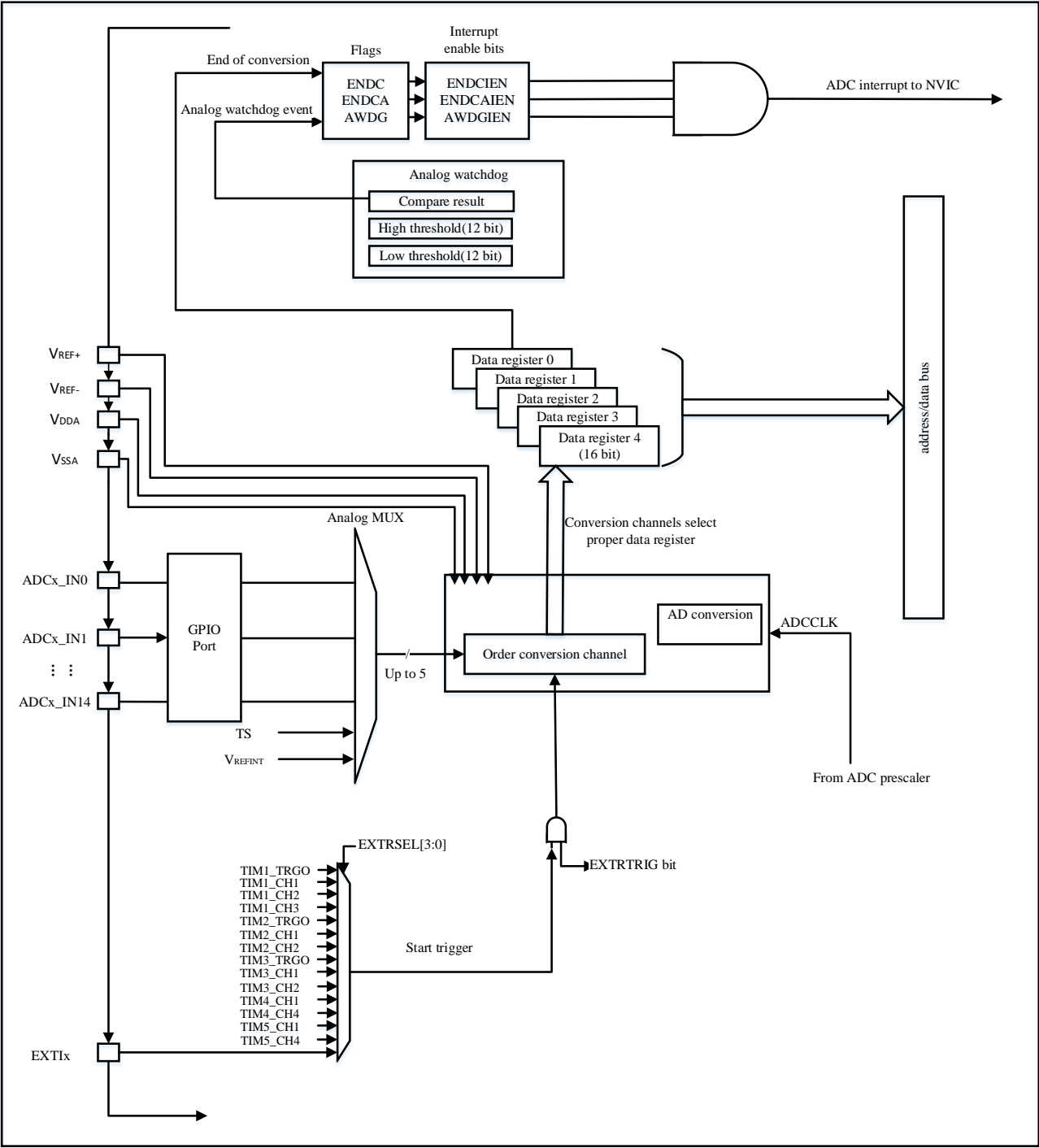


表 14-1 ADC 引脚

名称	信号类型	注解
V _{REF+}	ADC使用的正参考电压， 2.4V ≤ V _{REF+} ≤ V _{DDA} (5.5V)	输入，正模拟参考电压
V _{DDA}	输入，模拟电源	等效于V _{DD} 的模拟电源且：2.4V ≤ V _{DD} ≤ 5.5V
V _{SSA}	输入，模拟电源地	等效于V _{SS} 的模拟电源地
ADCx_IN[14:0]	模拟输入信号	15个模拟外部输入通道

14.3.1 ADC 时钟

ADC 需要三个时钟，ADC_CLK，HCLK, ADC_1MCLK。

- HCLK 用于寄存器的访问时钟。
- ADC_CLK 为 ADC 的工作时钟。
- ADC_1MCLK 用于内部计时功能，在 RCC 中配置，频率大小必须配置成 1MHz

注意：

1. 配置 ADC_CLK 分频作为工作时钟最高可到 24M，支持分频 1,2,3,4,6,8,10,12,16,24,32,64,128,256。

14.3.2 ADC 开关控制

只有在上电过程完成后，您才能进行下一步。您可以通过轮询 ADC_CTRL3.RDY 来检查上电是否完成。

您可以设置 ADC_CTRL2.ON 来打开 ADC。第一次设置 ADC_CTRL2.ON 时，它将 ADC 从断电状态唤醒。在 ADC 的上电延迟(tSTAB)之后，当 ADC_CTRL2.ON 再次置位时，转换开始。

可以通过清除 ADC_CTRL2.ON 将 ADC 置于断电模式来停止转换。在这种模式下，ADC 几乎不消耗功率（仅几 μA ）。可以通过轮询 ADC_CTRL3.PDRDY 来检查掉电情况。

在 ADC 禁用的时候，默认都是 power-down 模式。

14.3.3 通道选择

每个通道可以配置为规则组。在任意多个通道上以任意顺序进行的一系列转换。例如可以如下顺序完成转换：通道 3、通道 8、通道 2、通道 1、通道 0。

规则组由多次转换组成，最多 5 个，规则通道和他们的转换顺序在 ADC_DATx.SEQx[4:0]寄存器指定规则通道。ADC_CTRL2.LEN[2:0]位指定规则通道序列长度。

注意：在转换期间，禁止更改 ADC_DATx.SEQx[4:0] 寄存器；ADC_DATx.SEQx[4:0] 寄存器只能在 ADC 空闲时更改。

14.3.4 内部通道

- 温度传感器连接到 ADC_IN15
- V_{REFINT} 连接到 ADC_IN16

ADC 内部通道可以按规则通道的方式进行转换。

14.3.5 单次转换模式

ADC 可以通过配置 ADC_CTRL2.CTU 为 0 进入单次转换模式。在该模式下，外部触发（只适用于规则通道）或设置 ADC_CTRL2.ON=1（仅适用于规则通道）可以启动 ADC 启动转换，ADC 只进行一次转换。

转换开始后，当一个规则通道转换完成时，规则通道转换结束标志（ADC_STS.ENDC）将被置 1。如果规则通道转换结束中断使能（ADC_CTRL1.ENDCIEN）位被置 1，则一个中断将生成，转换后的数据将存储在 ADC_DATx.DAT[15:0]寄存器中。

单次转换后，ADC 会停止。

14.3.6 连续转换模式

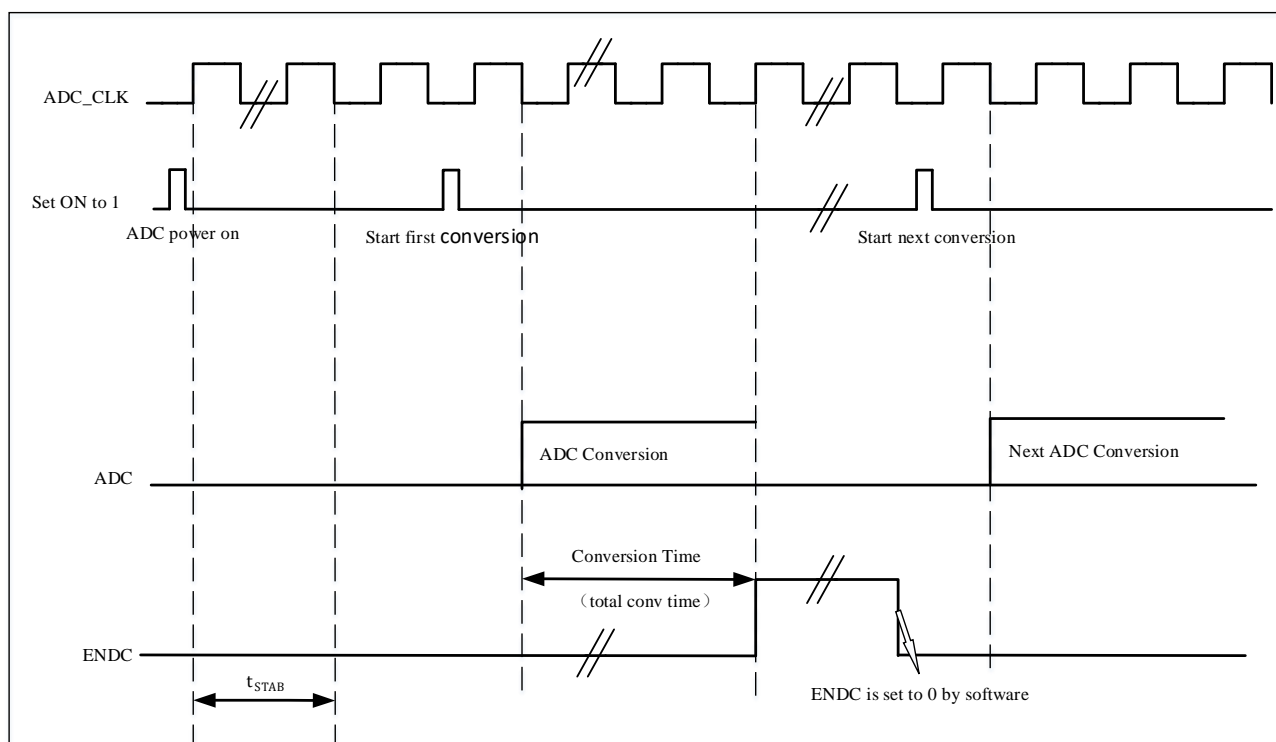
ADC 可以通过配置 ADC_CTRL2.CTU 为 1 进入连续转换模式。在该模式下，外部触发或设置 ADC_CTRL2.ON 为 1 可以启动 ADC 开始转换，ADC 会持续转换选择的通道。

转换开始后，当规则通道转换完成时，规则通道转换结束标志位（ADC_STS.ENDC）将设置为 1。如果规则通道转换结束中断使能位（ADC_CTRL1.ENDCIEN）设置为 1，将产生一个中断。转换后的数据将存储在 ADC_DATx.DAT[15:0]寄存器中。

14.3.7 时序图

ADC_CTRL2.ON 首次设置为 1 时，ADC 上电。ADC 上电后，ADC 需要一定时间来保证其稳定性。ADC 稳定后，ADC_CTRL2.ON 被设置为 1，同时，再次通过软件对 ADC_CTRL2.ON 写 1，ADC 开始转换，24 个时钟周期后，转换结束标志位将在转换完成后设置为 1。

图 14-2 时序图



14.3.8 模拟看门狗

可以通过设置 ADC_CTRL1.AWDGERCH 为 1 在规则通道上打开模拟看门狗。可以通过配置 ADC_WDGHIGH.HTH[11:0] 设置模拟看门狗的高阈值，模拟看门狗的 low 阈值可以通过 ADC_WDGLow.LTH[11:0] 来设置。模拟看门狗的阈值与数据对齐的方式无关，因为 ADC 的转换值与阈值的比较是在对齐之前完成。当 ADC 转换的值高于模拟看门狗的高阈值或低于模拟看门狗的 low 阈值时，则模拟看门狗标志(ADC_STS.AWDG)将被置为 1，如果 ADC_CTRL1.AWDGIEN 已配置为 1，此时会产生中断。通过配置 ADC_CTRL1.AWDGSGLEN 和 ADC_CTRL1.AWDGCH[4:0]，可以控制模拟看门狗作用于一个或

多个通道，如表 14-2 所示。

表 14-2 模拟看门狗通道选择

模拟看门狗警戒的通道	ADC_CTRL1 寄存器控制位	
	AWDGSGLEN位	AWDGERCH位
所有规则通道	0	1
单一的规则通道	1	1

14.3.9 扫描模式

通过配置 ADC_CTRL2.SCAMD 为 1 可以开启扫描模式，通过配置寄存器 ADC_DATx.SEQx[4:0]，可以选择转换通道序列，ADC 会对所有选择的通道进行扫描转换，每个序列可选择任意通道进行转换。转换开始后，通道将顺序转换，最多支持 5 个转换序列。如果此时 ADC_CTRL2.CTU 为 1，则在所有选中的规则通道的转换完成后，将从转换序列的第一个通道重新开始转换。

14.4 数据对齐

转换后的数据有两种对齐方式：左对齐和右对齐。对齐可以通过 ADC_CTRL2.ALIG 位设置。ADC_CTRL2.ALIG=0 为右对齐，ADC_CTRL2.ALIG=1 为左对齐，如下表所示。

表 14-3 数据右对齐

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	-----	-----	----	----	----	----	----	----	----	----	----	----

表 14-4 数据左对齐

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
-----	-----	----	----	----	----	----	----	----	----	----	----	---	---	---	---

14.5 可编程的通道采样时间

ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT.SAMP[4:0]更改。对不同的通道，采样间隔可以统一的编程。总转换时间如下计算：

$$T_{CONV} = \text{采样时间} + 12 \text{ 个周期}$$

注意：连续转换模式的第一次转换和单次转换模式每次转换需要额外增加 2 个周期。

例如：

当 ADCCLK=24MHz，采样时间为 12 个周期且分辨率为 12bit，总转换时间就是“12+12”，ADC_CLK 周期，就是：

$$T_{CONV} = 12 + 12 = 24 \text{ 周期} = 1\mu\text{s}$$

注：所有 ADC 通道共用一个通道采样时间配置。

14.6 外部触发转换

对于规则序列，软件将 ADC_CTRL2.EXTRTRIG 位设置为 1，则规则通道可以使用外部事件的上升沿触发

启动转换，然后软件通过配置 ADC_CTRL2.EXTRSEL[3:0]来选择规则序列的外部触发源。外部触发源选择如下表所示。如果选择 SWSTRCH 作为外部触发源，则可以通过将 ADC_CTRL2.SWSTRCH 设置为 1 来启动规则通道转换。

表 14-5 ADC 用于规则通道的外部触发

EXTRSEL[3:0]	触发源	类型
0000	TIM1_TRGO事件	来自片上定时器的内部信号
0001	TIM1_CC1事件	
0010	TIM1_CC2事件	
0011	TIM1_CC3事件	
0100	TIM2_TRGO事件	
0101	TIM2_CC1事件	
0110	TIM2_CC2事件	
0111	TIM3_TRGO事件	
1000	TIM3_CC1事件	
1001	TIM3_CC2事件	
1010	TIM4_CC1事件	
1011	TIM4_CC4事件	
1100	TIM5_CC1事件	
1101	TIM5_CC4事件	
1110	EXTI 线0~15事件	外部引脚/来自片上定时器的内部信号
1111	SWSTRCH	软件控制位

14.7 DMA 请求

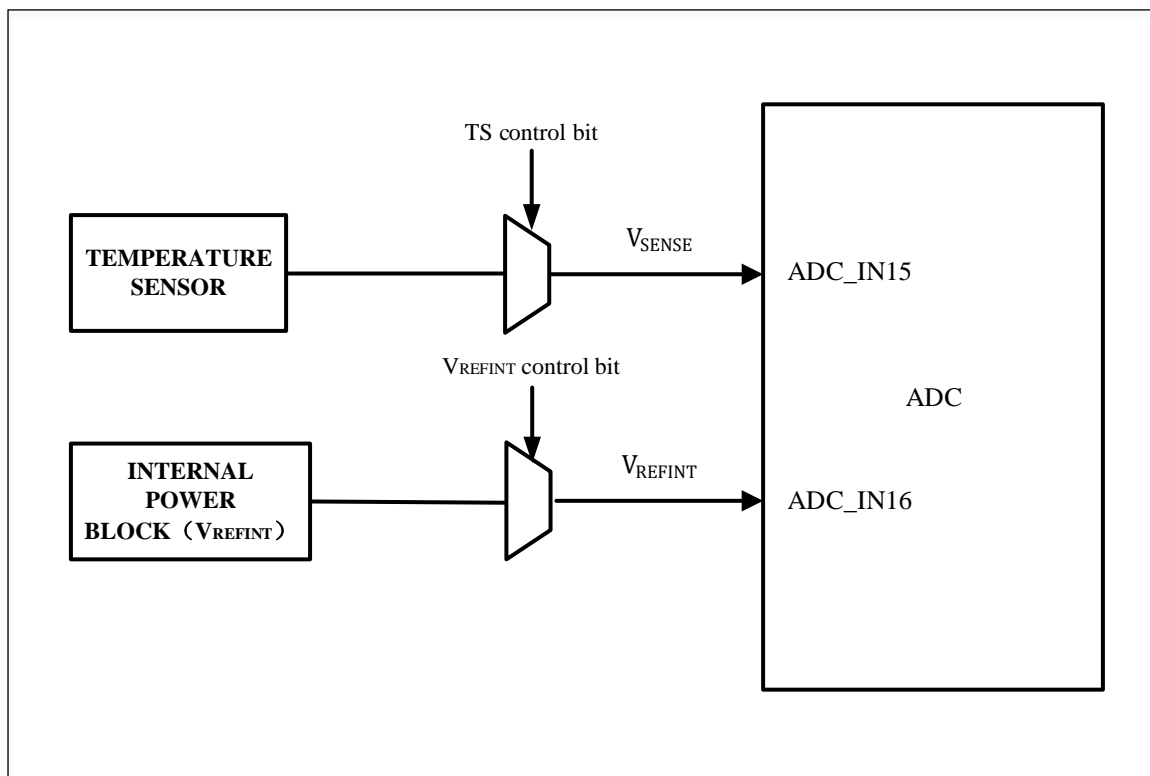
避免多个规则通道转换时数据过多，导致 ADC_DATx 寄存器中保存的规则通道转换结果丢失，可以将 ADC_CTRL2.ENDDMA 位设置为 1，以此使用 DMA。当 ADC 规则通道转换结束时，会产生一个 DMA 请求。DMA 收到请求后，会将转换后的数据从 ADC_DATx 寄存器传送到用户指定的目标地址。

14.8 温度传感器

设置 ADC_CTRL2.TEMPEN 位为 1，使能温度传感器，设备工作时使用温度传感器检测环境温度。温度传感器采样的输出电压通过 ADC_IN15 通道转换为数字值。温度传感器工作时，理想的采样时间为 17.1us；当温度传感器不工作时，ADC_CTRL2.TEMPEN 位可通过软件清零以降低功耗。图 14-3 是温度传感器的框图。

温度传感器的输出电压随温度线性变化。不同的芯片由于生产工艺的不同，在温度曲线上会有不同的偏移量。通过测试，发现最大偏移为 3° C。这一特性使得内部温度传感器更适合检测温度变化。不适合测量绝对温度。当需要精确的温度测量时，应使用外部温度传感器。

图 14-3 温度传感器通道框图



14.8.1 测量温度值

1. 配置通道（ADC_IN15）和通道的采样时间为 17.1us
2. 将 ADC_CTRL2.TEMPEN 位设置为 1 以启用温度传感器
3. 设置 ADC_CTRL2.ON 位为 1 以启动 ADC 转换（或通过外部触发）
4. 读取 ADC 数据寄存器中的温度数据，通过以下公式计算温度值：

$$\text{温度 } (^{\circ}\text{C}) = \{ (V_{\text{SENSE}} - V_{\text{Temperature}}) / \text{Avg_Slope} \} + \text{Temperature} - \text{Toffset}$$

其中：

$V_{\text{Temperature}}$ = Temperature 对应的 V_{SENSE}

Toffset = 0 $^{\circ}$ C，表示温度误差补偿经验值（单位为 $^{\circ}$ C）

Temperature 为校准温度

Avg_Slope = 温度和 V_{SENSE} 曲线的平均斜率（mV/ $^{\circ}$ C 或 μ V/ $^{\circ}$ C）

参考数据手册的电气特性章节中 Avg_Slope 的实际值。

注意：在传感器从断电模式唤醒到正确输出 V_{SENSE} 之前，有一个建立时间；ADC 上电后还有一个建立时间，所以为了缩短延迟，ADC_CTRL2.TEMPEN 和 ADC_CTRL2.ON 位应该同时置位。

14.9 ADC 中断

ADC 中断可以来自规则序列转换的结束、输入电压不在模拟看门狗设置的范围、任何规则通道转换的结束。这些中断具有独立的中断使能位。

ADC_STS 寄存器中有 1 个状态标志：规则序列通道转换启动(STR)。但是 ADC 中没有与这个标志相关的中断。

表 14-6 ADC 中断

中断事件	事件标志	使能控制位
规则序列转换结束	ENDC	ENDCIEN
超出模拟看门狗阈值	AWDG	AWDGIEN
任何通道转换结束	ENDCA	ENDCAIEN

14.10 ADC 寄存器

14.10.1 ADC 寄存器总览

表 14-7 ADC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
000h	ADC_STS	Reserved																												ENDCA	STR	ENDC	AWDG									
	Reset Value																													0	0	0	0									
004h	ADC_CTRL1	Reserved																				REFSEL		AWDGEN	AWDGSGLN	AWDGIEN	ENDCIEN	AWDGCCH[4:0]														
	Reset Value																					0		0	0	0	0	0	0	0	0	0										
008h	ADC_CTRL2	Reserved																TEMPEN	ENDMA	COV_MODE	LEN[2:0]	SWSTART	EXTRTRIG	EXTRSEL[3:0]			ALIG	CONT	ON													
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0													
00Ch	ADC_CTRL3	Reserved																								ENDCAIEN	PDRDY	RDY	BUF_READY	BUF_EN	Reserved											
	Reset Value																									0	1	0	0	0												
010h	ADC_SAMP	Reserved																												SAMP[4:0]												
	Reset Value																													0	0	0	0	0	0	0	0	0	0	0	0	0
014h	ADC_WDGHIGH	Reserved																HTH[11:0]																								
	Reset Value																	1	1	1	1	1	1	1	1	1	1	1	1													
018h	ADC_WDGLOW	Reserved																LTH[11:0]																								
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
01Ch	ADC_DAT0	Reserved										SEQ0[20:16]				DAT0[15:0]																										
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
020h	ADC_DAT1	Reserved										SEQ1[20:16]				DAT1[15:0]																										
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
024h	ADC_DAT2	Reserved										SEQ2[20:16]				DAT2[15:0]																										
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
028h	ADC_DAT3	Reserved										SEQ3[20:16]				DAT3[15:0]																										
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
02Ch	ADC_DAT4	Reserved										SEQ4[20:16]				DAT4[15:0]																										
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										

14.10.2 ADC 状态寄存器(ADC_STS)

地址偏移: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												ENDCA	STR	ENDC	AWDG
												rc_w0	rc_w0	rc_w0	rc_w0

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3	ENDCA	任意通道转换结束位（Any End of conversion flag） 任意规则通道转换结束时被硬件设置为1。 0：转换没有完成； 1：转换完成。
2	STR	规则通道开始位（Regular channel Start flag） 规则通道转换开始时该位被硬件设置为1，由软件清除。 0：规则通道转换未开始； 1：规则通道转换已开始。
1	ENDC	转换结束位（End of conversion） 规则通道序列转换结束时被硬件设置为1。 0：转换未完成； 1：转换完成。
0	AWDG	模拟看门狗标志位（Analog watchdog flag） 转换的电压值超出了ADC_WDGHIGH.HTH和ADC_WDGLow.LTH寄存器定义的范围时被硬件设置为1，由软件清除 0：没有发生模拟看门狗事件； 1：发生模拟看门狗事件。

14.10.3 ADC 控制寄存器 1(ADC_CTRL1)

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				REFSEL	Reserved	AWDG ERCH	AWDG SGLEN	AWD GIEN	END CIEN	AWDGCH[3:0]					
				rw		rw	rw	rw	rw	rw					

位域	名称	描述
31:11	Reserved	保留，必须保持复位值。
10	REFSEL	ADC参考源选择(ADC reference source selset) 0：参考源为外部的VDDA 1：参考源为内部的VREF+
9	Reserved	保留，必须保持复位值。
8	AWDGERCH	在规则通道上开启模拟看门狗（Analog watchdog enable on regular channels） 该位由软件设置和清除。

位域	名称	描述
		0: 在规则通道上关闭模拟看门狗; 1: 在规则通道上开启模拟看门狗。
7	AWDGSLEN	扫描模式中在一个单一的通道上使用看门狗 (Enable the watchdog on a single channel in scan mode) 该位由软件设置和清除, 用于AWDCH[4:0]位指定的通道上的模拟看门狗功能开启或所有通道上模拟看门狗功能开启 0: 在所有的通道上使用模拟看门狗; 1: 在单一通道上使用模拟看门狗。
6	AWDGIEN	模拟看门狗中断使能 (Analog watchdog interrupt enable) 该位由软件设置和清除以禁止或允许模拟看门狗产生中断。在扫描模式下, 如果看门狗检测到超出范围的值, 则仅当该位置位时才会中止扫描。 0: 禁止模拟看门狗中断; 1: 使能模拟看门狗中断。
5	ENDCIEN	ENDC的中断使能 (Interrupt enable for ENDC) 该位由软件设置和清除, 以禁止或允许在规则转换序列转换结束后发生中断。 0: 禁止ENDC中断; 1: 使能ENDC中断。
4:0	AWDGCH[4:0]	模拟看门狗通道选择位 (Analog watchdog channel select bits) 这些位由软件设置和清除以选择模拟看门狗保护的输入通道。 00000: ADC模拟输入通道0; 00001: ADC模拟输入通道1; 10001: ADC模拟输入通道16; 其他值保留

14.10.4 ADC 控制寄存器 2(ADC_CTRL2)

地址偏移: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TEMPEN	ENDMA	SCANMD	LEN[2:0]			SWSTR RCH	EXT RTRIG	EXTRSEL[3:0]				ALIG	CTU	ON
	rw	rw	rw	rw			rw	rw	rw				rw	rw	rw

位域	名称	描述
31:15	Reserved	保留, 必须保持复位值。
14	TEMPEN	温度传感器使能 (Temperature sensor enable) 该位由软件设置和清除, 用于开启或禁止温度传感器通道

位域	名称	描述
		0: 禁止温度传感器测量 1: 使能温度传感器测量
13	ENDMA	直接存储器访问模式 (Direct memory access mode) 该位由软件设置和清除。有关详细信息, 请参见DMA控制器章节。 0: 不使用DMA模式。 1: 使用DMA模式。
12	SCANMD	扫描模式 (Scan mode) 该位由软件设置和清除以启用或禁用扫描模式。在扫描模式下, 转换由ADC_DATx.ADC_SEQx[4:0]寄存器选定的通道。 0: 禁用扫描模式; 1: 启用扫描模式。 <i>注意: 如果单独设置ADC_CTRL1.ENDCIEN 位, ADC_STS.ENDC中断仅在最后一个通道转换后发生。</i>
11:9	LEN[2:0]	规则通道序列长度 (Regular channel sequence length) 这些位由软件定义规则序列通道转换中的通道数。 000: 1个转换 001: 2个转换 010: 3个转换 011: 4个转换 100: 5个转换
8	SWSTRRCH	开始转换规则通道 (Start conversion of regular channels) 该位由软件设置以启动转换, 并在转换开始后由硬件清零。如果在ADC_CTRL2.EXTRSEL[3:0] 位中选择 SWSTRRCH 作为触发事件, 该位用于启动一组规则通道的转换 0: 复位状态; 1: 开始转换规则通道
7	EXTRTRIG	规则通道的外触发转换模式 (External trigger conversion mode for regular channels) 该位由软件设置和清零, 以启用或禁用可以启动规则序列转换的外部触发事件。 0: 不使用外部事件启动转换; 1: 使用外部事件启动转换。
6:3	EXTRSEL[3:0]	选择启动规则通道序列转换的外部事件 (External event select for regular group) 这些位选择外部事件以启动规则序列转换。 0000: 定时器1的TRGO事件 0001: 定时器1的CC1事件 0010: 定时器1的CC2事件 0011: 定时器1的CC3事件 0100: 定时器2的TRGO事件 0101: 定时器2的CC1事件 0110: 定时器2的CC2事件 0111: 定时器3的TRGO事件 1000: 定时器3的CC1事件 1001: 定时器3的CC2事件 1010: 定时器4的CC1事件

位域	名称	描述
		1011: 定时器4的CC4事件 1100: 定时器5的CC1事件 1101: 定时器5的CC4事件 1110: EXTI 线 1111: SWSTRRCH
2	ALIG	数据对齐 (Data alignment) 该位由软件设置和清除。 0: 右对齐; 1: 左对齐。
1	CTU	连续转换 (Continuous conversion) 该位由软件设置和清除。如果该位置位, 则转换将继续, 直到该位被清除。 0: 单次转换模式; 1: 连续转换模式。
0	ON	A/D 转换器开/关 (A/D converter ON/OFF) 该位由软件设置和清除。当该位为“0”时, 写入“1”会将ADC从断电模式中唤醒。 当该位为“1”时, 写入“1”开始转换。应注意, 转换器上电与转换开始之间存在延迟 tSTAB, 请参见图14-2 0: 关闭ADC转换/校准并进入掉电模式; 1: 启动ADC并开始转换。 注意: 如果该寄存器中的其他位随着ON发生变化, 则不会触发转换。这是为了防止触发错误的转换。

14.10.5 ADC 控制寄存器 3 (ADC_CTRL3)

地址偏移: 0x0C

复位值: 0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										ENDC AIEN	PDRDY	RDY	VREF RDY	VREF EN	Reserved
										rw	r	r	r	rw	

位域	名称	描述
31:6	Reserved	保留, 必须保持复位值。
5	ENDCAIEN	任何通道中断使能 (Interrupt enable for any regular channels) 该位由软件设置和清除以启用/禁用任意通道转换结束中断。 0: ADC_STS.ENDCA 禁用中断; 1: ADC_STS.ENDCA 启用中断。

位域	名称	描述
4	PDRDY	ADC掉电准备（Power down ready） 0: 没有准备好； 1: 准备好。
3	RDY	ADC准备(Ready) 0: 没有准备好； 1: 准备好。
2	VREFRDY	VREFINT准备好（VREFINT_READY） ADC内部输入buffer准备状态，在测量VREFINT之前软件必须检测该状态位 0: VREFINT 没有准备好； 1: VREFINT准备好。
1	VREFEN	VREFINT使能（VREFINT_EN） ADC内部输入buffer使能，在测量VREFINT之前软件必须使能该位 0: 禁止VREFINT测量 1: 使能 VREFINT测量
0	Reserved	保留，必须保持复位值。

14.10.6 ADC 采样时间寄存器(ADC_SAMPT)

地址偏移：0x10

复位值：0x0000 0000

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

位域	名称	描述
31:5	Reserved	保留，必须保持复位值。
4:0	SAMP[4:0]	通道采样时间选择（Channel Sample time selection） 这些位用于独立选择每个通道的采样时间。通道选择位在采样期间必须保持不变。 00000: 6 周期 （只在24M的工作频率下才设置，其中HSI为24M） 00001: 8 周期 00010: 12 周期 00011: 14 周期 00100: 20 周期 00101: 26 周期 00110: 30 周期 00111: 42 周期 01000: 56 周期

位域	名称	描述
		01001: 72 周期 01010: 88 周期 01011: 120 周期 01100: 182 周期 01101: 240 周期 01110: 380 周期 01111: 760 周期 10000: 1520 周期 10001: 3040 周期

14.10.7 ADC 看门狗高阈值寄存器(ADC_WDGHIGH)

地址偏移: 0x14

复位值: 0x0000 0FFF

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

rw

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值。
11:0	HTH[11:0]	模拟看门狗高阈值 (Analog watchdog high threshold) 这些位定义模拟看门狗的高阈值。

14.10.8 ADC 看门狗低阈值寄存器(ADC_WDGLOW)

地址偏移: 0x18

复位值: 0x0000 0000

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

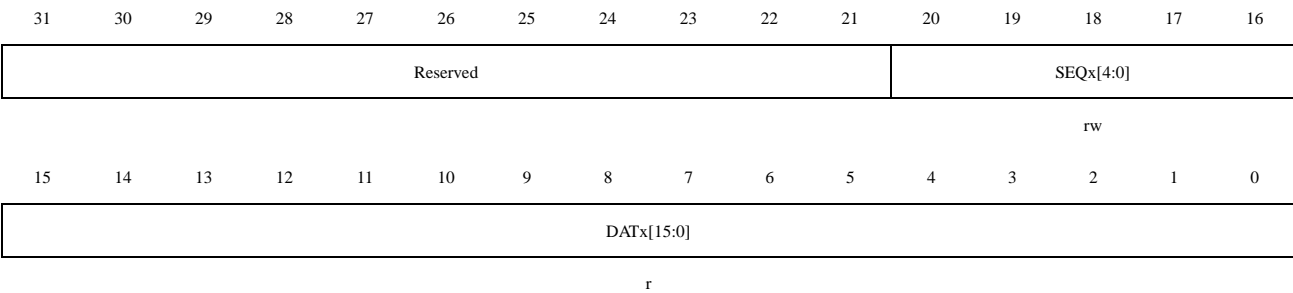
rw

位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
11:0	LTH[11:0]	模拟看门狗低阈值（Analog watchdog low threshold） 这些位定义模拟看门狗的低阈值。

14.10.9 ADC 规则数据寄存器 x(ADC_DATx)(x = 0..4)

地址偏移：0x1C – 0x2C

复位值：0x0000 0000



位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:16	SEQx[4:0]	选择结果数据寄存器储存的通道号 00000：通道0 00001：通道1 ... 10000：通道16 其他值：保留
15:0	DATx[15:0]	规则转换的数据（Regular data） 这些位是只读的，包含规则通道的转换结果。数据左对齐或右对齐。

15 数字模拟转换（DAC）

15.1 DAC 介绍

DAC 是数字/模拟转换器，主要是数字输入，电压输出。DAC 数据有 8 位或 12 位两种模式，支持 DMA 功能。当 DAC 配置为 12bit 模式时，DAC 数据可以左对齐或者右对齐；当 DAC 配置为 8bit 模式时，DAC 数据可以右对齐。DAC 输出通道有 1 个，有独立的转换器。 V_{REF+} 通过引脚输入作为 DAC 参考电压，使 DAC 的转换数据精确度更高。

15.2 DAC 主要特性

- 1 个独立的 DAC 转换器，对应 1 个输出通道
- 单调输出
- 支持 8 位或 12 位输出，数据在 12 位模式下分右对齐和左对齐两种模式
- 同步更新
- 支持 DMA 功能
- 噪声波、三角波形生成
- 输入参考电压 V_{REF+}
- 外部事件触发转换

DAC 结构框图如下图，表 15-1 为引脚的说明。

图 15-1 DAC 通道的框图

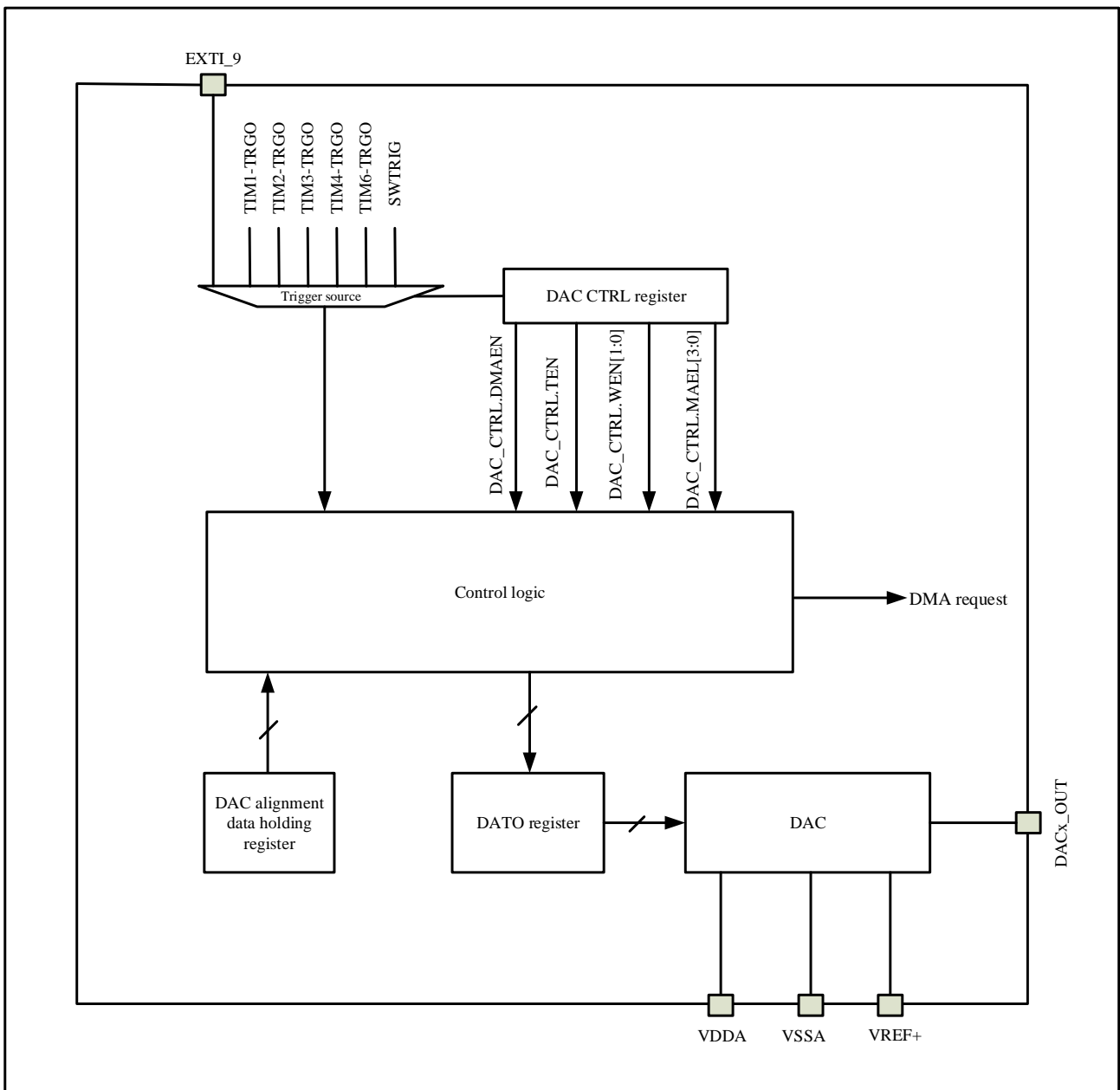


表 15-1 DAC 引脚

名称	描述	类型
V_{REF+}	DAC使用的正参考电压， $2.97V \leq V_{REF+} \leq V_{DDA}$	输入，正模拟参考电压
V_{DDA}	模拟电源	输入，模拟电源
V_{SSA}	模拟电源的地	输入，模拟电源地
DAC_OUT	DAC模拟输出	模拟输出信号

注意：使能 DAC 时，PB13 需要配置为模拟输入模式，PB13 会自动连接到 DAC 的输出。

15.3 DAC 功能描述与操作说明

15.3.1 DAC 开启

给 DAC 上电可通过配置 `DAC_CTRL.CHEN = 1` 完成，DAC 需要一段时间 `tWAKEUP` 打开。

15.3.2 DAC 输出缓存

通过配置 `DAC_CTRL.BEN` 开启或关闭 DAC 的输出缓存，输出缓存开启，输出阻抗降低，驱动能力增强，可以在没有外部运放的条件下驱动外部负载。

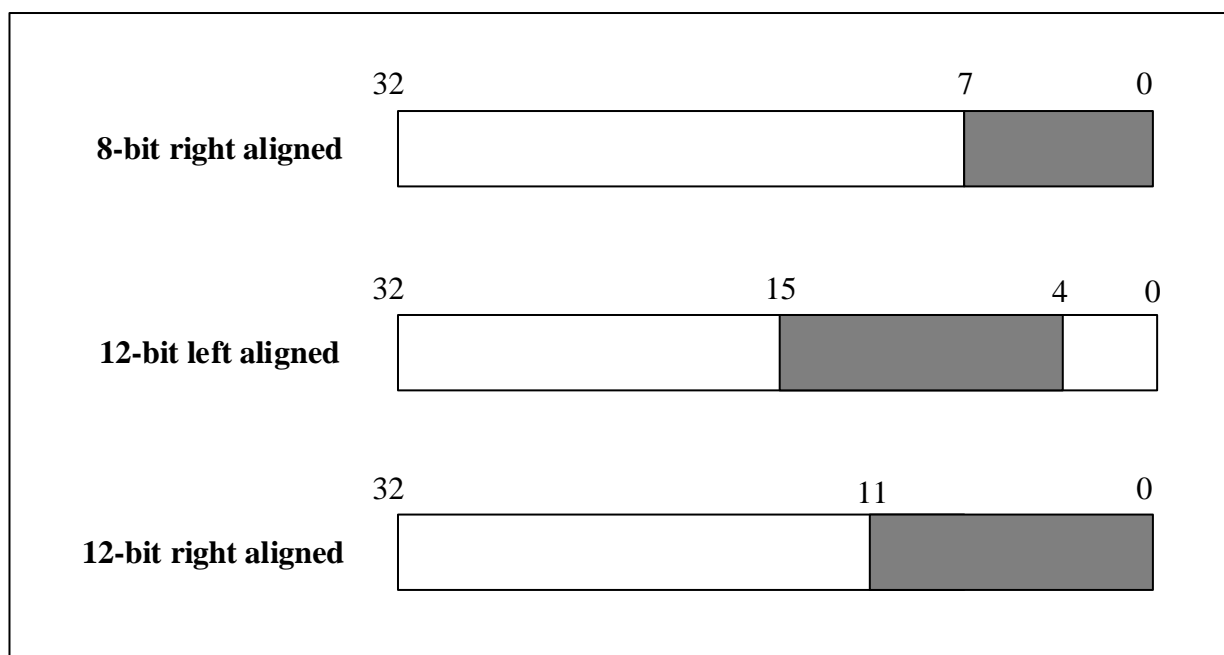
15.3.3 DAC 数据格式

配置数据写入 `DAC_DR12CH` 寄存器时，数据写入 `DAC_DR12CH[11:0]`, 12 位数据右对齐。（实际是存入寄存器 `DACCHD[11:0]` 位，`DACCHD` 是内部的数据保存寄存器）

配置数据写入 `DAC_DL12CH` 寄存器时，数据写入 `DAC_DL12CH[15:4]`, 12 位数据左对齐。（实际是存入寄存器 `DACCHD[11:0]` 位，`DACCHD` 是内部的数据保存寄存器）

配置数据写入 `DAC_DR8CH` 寄存器时，数据写入 `DAC_DR8CH[7:0]`, 8 位数据右对齐。（实际是存入寄存器 `DACCHD[11:4]` 位，`DACCHD` 是内部的数据保存寄存器）

图 15-2 单 DAC 通道模式的数据寄存器



15.3.4 DAC 触发

配置 `DAC_CTRL.TEN = 1` 可以使能 DAC 的外部触发，通过配置 `DAC_CTRL.TSEL[2:0]` 来选择一个外部触发事件作为 DAC 的外部触发源。

表 15-2 DAC 外部触发

触发源	类型	TSEL[2:0]
定时器 1 TRGO 事件	来自片上定时器的内部信号	000
定时器 2 TRGO 事件		001
定时器 3 TRGO 事件		010
定时器 4 TRGO 事件		011
定时器 6 TRGO 事件		100
EXTI line 9	外部引脚	101
SWTRIG（软件触发）	软件控制位	110

当 DAC 的触发源为定时器输出或者 EXTI line 9 的上升沿时，当触发产生，对齐数据保持寄存器的数据会被传送到 DAC_DATO 寄存器中，这个数据传输过程需要 3 个 APB1 时钟周期的时间。

配置 DAC_SOTTR.TREN = 1 可以使能 DAC 软件触发，当 DAC 由软件触发时，对齐数据保持寄存器的数据会被传送到 DAC_DATO 寄存器中。

注意：

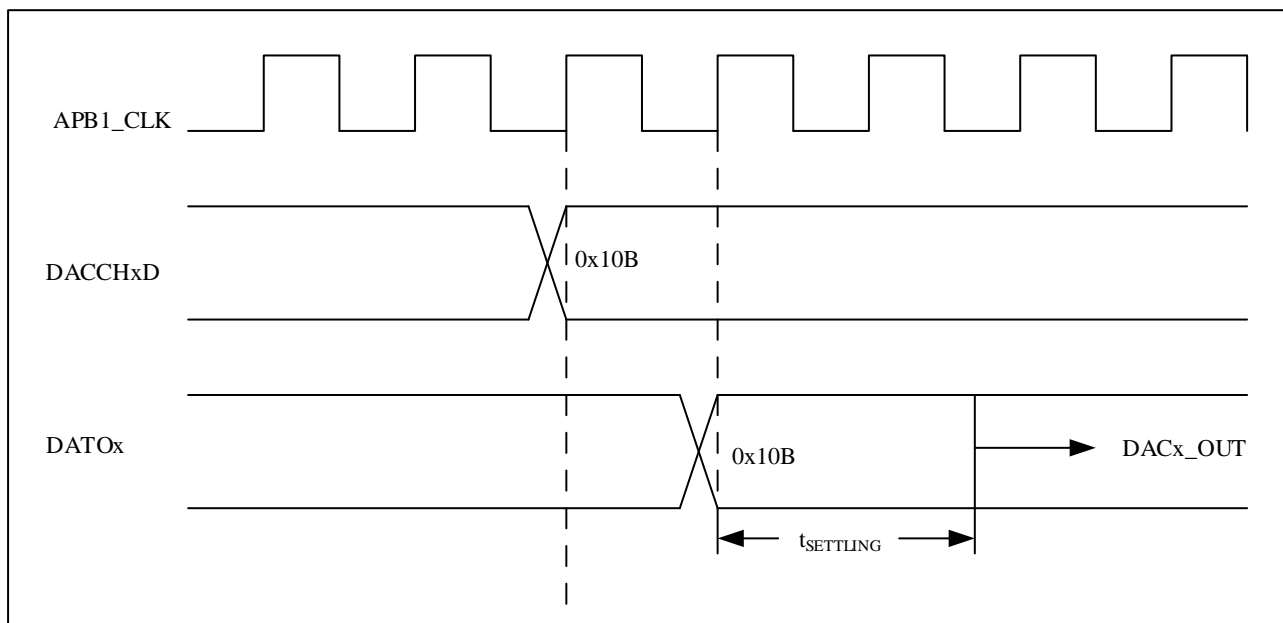
1. DAC 使能状态下禁止改变 DAC_CTRL.TSEL[2:0] 位。
2. 软件触发时对齐数据保持寄存器的数据被传送到 DAC_DATO 寄存器中需要 1 个 APB1 时钟周期的时间。

15.3.5 DAC 转换

如果 DAC 触发开启，那么根据选择的触发事件，硬件触发发生时，DAC 对齐数据保持寄存器的数据会在三个 APB1 周期后将数据传送到 DAC_DATO 寄存器，软件触发发生时，DAC 对齐数据保持寄存器的数据会在一个 APB1 周期后将数据传送到 DAC_DATO 寄存器。如果触发未使能，DAC 对齐数据保持寄存器的数据会在一个 APB1 周期后将数据自动传送到 DAC_DATO 寄存器。

当 DAC 对其数据保持寄存器将数据传送到 DAC_DATO 寄存器后，经过时间 $t_{SETTLING}$ 输出才有效，这个时间与电源电压及模拟输出负载相关。

图 15-3 触发禁能时转换时序图



15.3.6 DAC 输出电压

数字输入通过 DAC 模块转换为模拟电压输出，它们之间呈线性关系，输出范围为 0 到 VREF+。以下是 DAC 的输出电压计算公式：

DAC 输出 = $VREF+ * (DATO / 4095)$ 。

15.3.7 DMA 请求

配置 DAC_CTRL.DMAEN=1 来开启 DMA 功能，有外部触发发生时（不是软件触发），生成一个 DMA 请求，随后对齐数据保持寄存器的数据被传输到 DAC_DATO 寄存器。

注意：DAC 的 DMA 请求没有累计功能，当第 2 个外部触发发生在响应第 1 个外部触发之前，则不能处理第 2 个 DMA 请求，也没有报告错误机制。

15.3.8 噪声产生

DAC 可以生成噪声，通过配置 DAC_CTRL.WEN[1:0] 2b'01 开启噪声功能，通过配置 DAC_CTRL.MASEL[3:0] 来选择屏蔽线性反馈移位寄存器（LFSR）的哪些位，LFSR 寄存器的值与 DAC 对齐数据保持寄存器的值相加后写入到 DAC_DATO 寄存器中（溢出位被舍弃）。LFSR 的初始值为 0xAAA，LFSR 的值更新于触发事件发生后的 3 个 APB1 周期之后。

图 15-4 DAC LFSR 算法

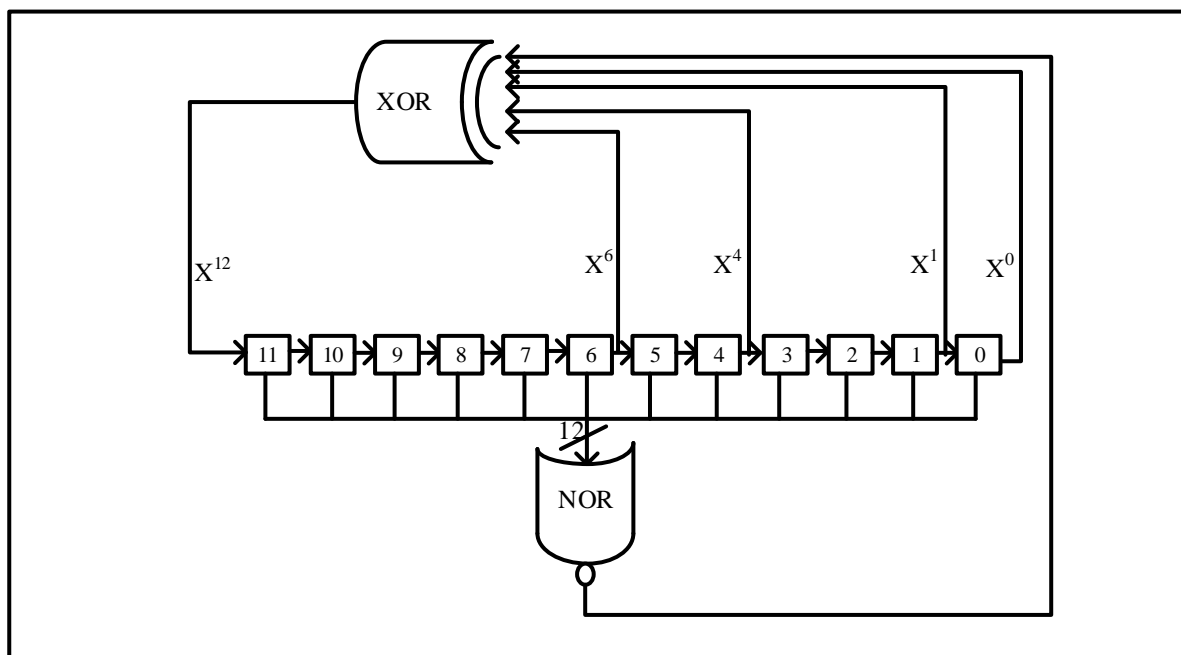
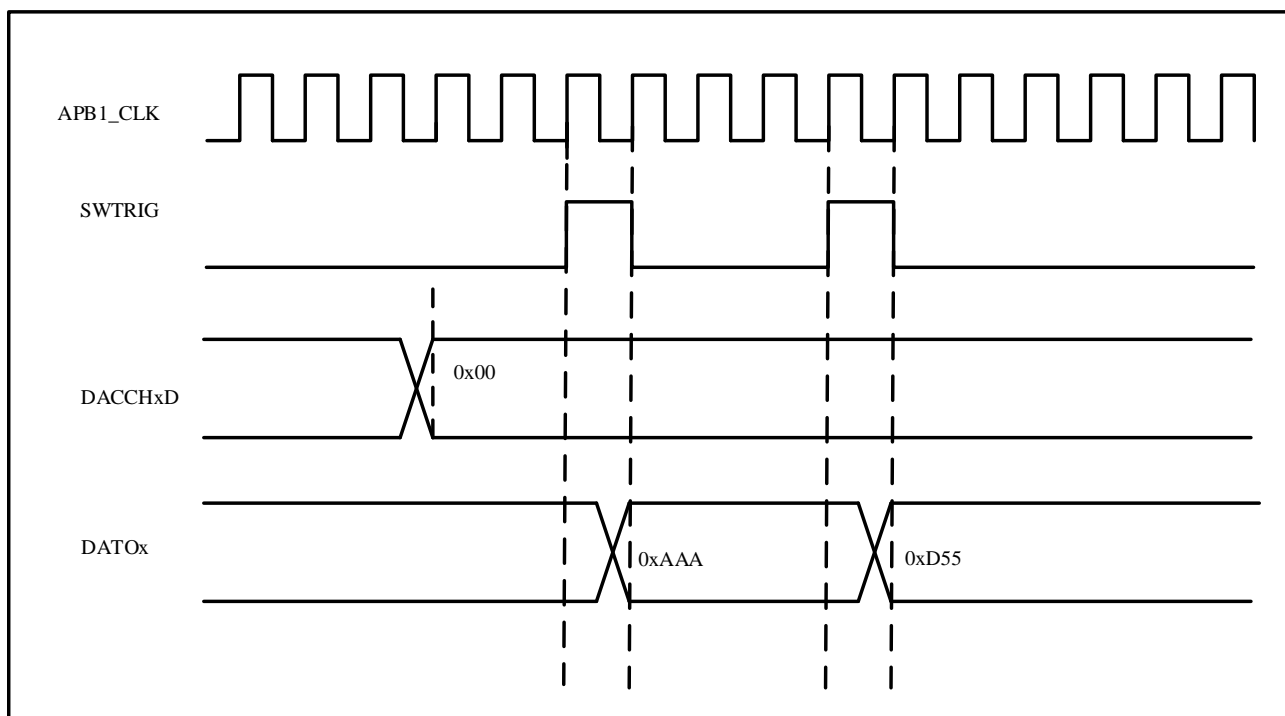


图 15-5 带 LFSR 波形生成的 DAC 转换（使能软件触发）



注意：DAC 配置为触发才能产生噪声。

15.3.9 三角波产生

DAC 可以生成三角波，通过配置 `DAC_CTRL.WEN[1:0]` 为 `2b'10` 开启三角波功能，通过配置 `DAC_CTRL.MASEL[3:0]` 来选择三角波的幅值，内部的三角波计数器值与 DAC 对齐数据保持寄存器的值相加后写入到 `DAC_DATO` 寄存器中（溢出位被舍弃）。三角波计数器的值更新于触发事件发生后 3 个 APB1

周期，三角波计数器会累加到设置的最大幅值，然后递减到 0，依此循环。

图 15-6 DAC 三角波生成

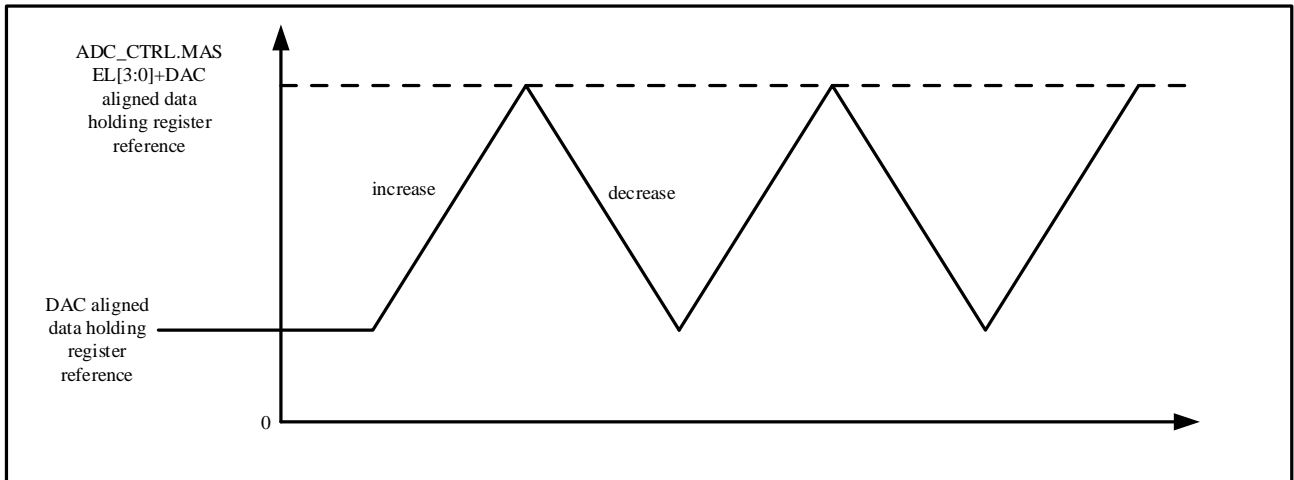
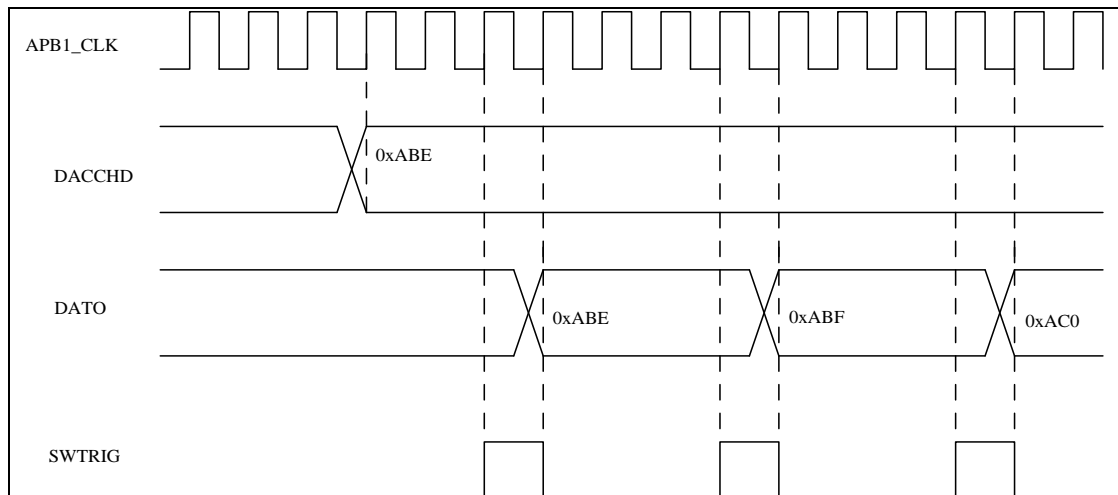


图 15-7 带三角生成的 DAC 转换（使能软件触发）



- 注意：
1. DAC 配置为触发才能产生三角波
 2. 不允许在 DAC 使能后设置 DAC_CTRL.MASEL[3:0]

15.4 DAC 寄存器

15.4.1 DAC 寄存器总览

表 15-3 DAC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
000h	DAC_CTRL	Reserved																TSEL[2:0]			WEN[1:0]		Reserved	BDASEL	MASEL[3:0]			TEN			BEN	DMAEN	CHEN	0		
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	DAC_SOTTR	Reserved																														TREN				
	Reset Value																															0				
008h	DAC_DATO	Reserved																		DACCHDO[11:0]																
	Reset Value																																			
00Ch	DAC_DR8CH	Reserved																						DACCHD[7:0]												
	Reset Value																							0	0	0	0	0	0	0	0	0	0	0	0	0
010h	DAC_DL12CH	Reserved														DACCHD [11:0]										Reserved										
	Reset Value															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
014h	DAC_DR12CH	Reserved																		DACCHD [11:0]																
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.2 DAC 控制寄存器（DAC_CTRL）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
Reserved		TSEL[2:0]			WEN[1:0]			Reserved	BDASEL	MASEL[3:0]			TEN	BEN	DMAEN	CHEN																
		rw			rw				rw	rw			rw	rw	rw	rw																

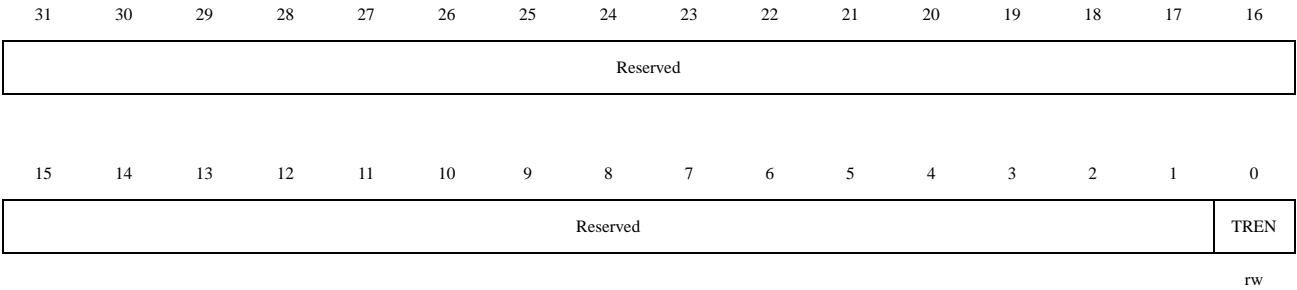
位域	名称	描述
31:15	Reserved	保留，必须保持复位值。
14:12	TSEL[2:0]	DAC 触发选择。 该位用于 DAC 外部触发的选择。 000: TIM1 TRGO 事件; 001: TIM2 TRGO 事件; 010: TIM3 TRGO 事件; 011: TIM4 TRGO 事件; 100: TIM6 TRGO 事件; 101: 外部中断线 9;

位域	名称	描述
		110: 软件触发;
11:10	WEN[1:0]	DAC 噪声/三角波功能选择。 这些位由软件置 1 和清零。 00: 禁用噪声和三角波 01: 开启噪声功能 1x: 开启三角波功能
9	Reserved	保留, 必须保持复位值。
8	BDASEL	DAC 通道输出 Buffer 驱动能力选择。 这些位由软件置 1 和清零。 0: DAC buffer 正常驱动能力 1: DAC buffer 高驱动能力
7:4	MASEL[3:0]	DAC 屏蔽/幅值选择器。 这些位由软件配置, 可以设置噪声功能的 LFSR 屏蔽位和三角波的幅值。 0000: 不屏蔽 LFSR 位 0 / 三角波幅值等于 1 0001: 不屏蔽 LFSR 位[1:0] / 三角波幅值等于 3 0010: 不屏蔽 LFSR 位[2:0] / 三角波幅值等于 7 0011: 不屏蔽 LFSR 位[3:0] / 三角波幅值等于 15 0100: 不屏蔽 LFSR 位[4:0] / 三角波幅值等于 31 0101: 不屏蔽 LFSR 位[5:0] / 三角波幅值等于 63 0110: 不屏蔽 LFSR 位[6:0] / 三角波幅值等于 127 0111: 不屏蔽 LFSR 位[7:0] / 三角波幅值等于 255 1000: 不屏蔽 LFSR 位[8:0] / 三角波幅值等于 511 1001: 不屏蔽 LFSR 位[9:0] / 三角波幅值等于 1023 1010: 不屏蔽 LFSR 位[10:0] / 三角波幅值等于 2047 ≥1011: 不屏蔽 LFSR 位[11:0] / 三角波幅值等于 4095
3	TEN	DAC 触发开启 该位由软件置 1 和清零, 用来开启/禁用 DAC 的触发。 0: 禁用 DAC 触发; 1: 开启 DAC 触发。
2	BEN	开启 DAC 输出缓存。 该位由软件置 1 和清零, 用来开启/禁用 DAC 的输出缓存。 0: 禁用 DAC 通道输出缓存; 1: 开启 DAC 通道输出缓存。
1	DMAEN	DAC 的 DMA 功能开启 该位由软件置 1 和清零。 0: 禁用 DAC 的 DMA 功能 1: 开启 DAC 的 DMA 功能
0	CHEN	DAC 开启。 该位由软件置 1 和清零, 用来开启/禁用 DAC。 0: 禁用 DAC; 1: 开启 DAC。

15.4.3 DAC 软件触发寄存器（DAC_SOTTR）

偏移地址：0x04

复位值：0x0000 0000

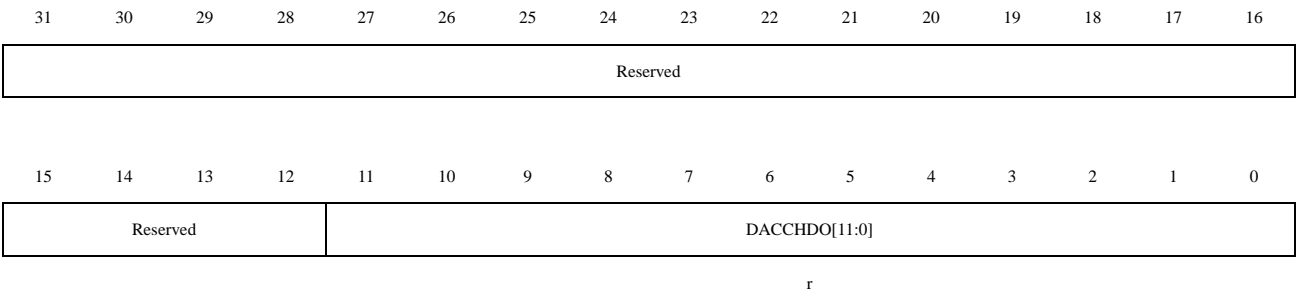


位域	名称	描述
31:1	Reserved	保留，必须保持复位值。
0	TREN	DAC 软件触发开启。 该位由软件置 1，用来开启/禁用软件触发。 0：禁用 DAC 软件触发； 1：开启 DAC 软件触发。 <i>注意：对齐数据保持寄存器将数据传送至 DAC_DATO 寄存器之后，一个 APBI 时钟之后该位会由硬件清 0。</i>

15.4.4 DAC 数据输出寄存器（DAC_DATO）

偏移地址：0x08

复位值：0x0000 0000

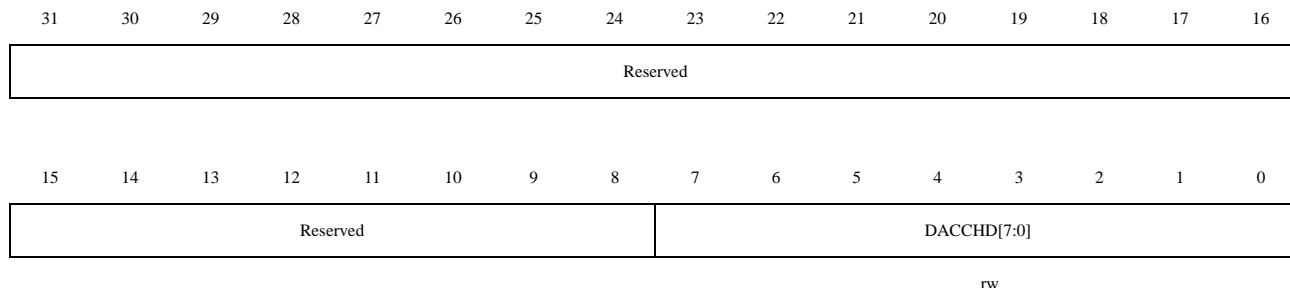


位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
11:0	DACCHDO[11:0]	DAC 数据输出。 这些位为只读，表示 DAC 通道的输出数据

15.4.5 DAC 的 8 位右对齐数据保持寄存器 (DAC_DR8CH)

偏移地址: 0x0C

复位值: 0x0000 0000

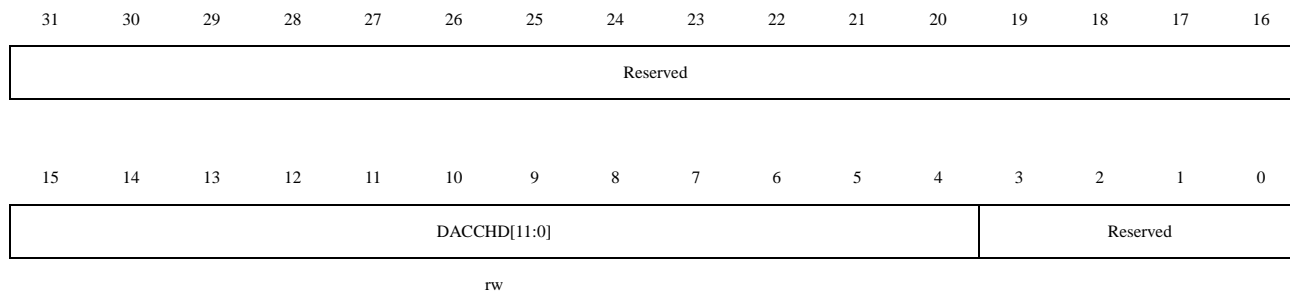


位域	名称	描述
31:8	Reserved	保留, 必须保持复位值。
7:0	DACCHD[7:0]	DAC8 位右对齐数据 这些位由软件配置, DAC 转换这些数据。

15.4.6 DAC 的 12 位左对齐数据保持寄存器 (DAC_DL12CH)

偏移地址: 0x10

复位值: 0x0000 0000

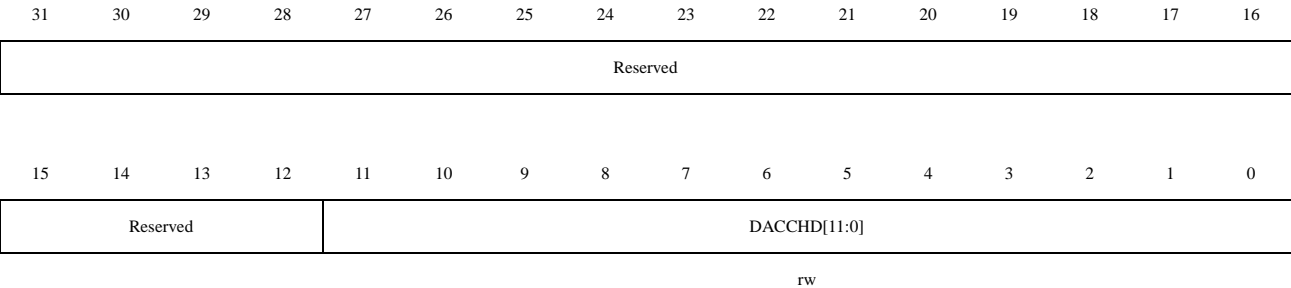


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:4	DACCHD[11:0]	DAC12 位左对齐数据 这些位由软件配置, DAC 转换这些数据。
3:0	Reserved	保留, 必须保持复位值。

15.4.7 DAC 的 12 位右对齐数据保持寄存器（DAC_DR12CH）

偏移地址：0x14

复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
11:0	DACCHD[11:0]	DAC12 位右对齐数据 这些位由软件配置，DAC 转换这些数据。

16 比较器（COMP）

COMP 模块用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“INP”输入端电压高于“INM”输入端电压时，比较器输出为高电平，当“INP”输入端电压低于“INM”输入端电压时，比较器输出为低电平。

16.1 COMP 系统连接框图

COMP 模块支持 1 个独立比较器，挂接在 APB1 总线上。

图 16-1 比较器 1 和比较器 2 系统连接图

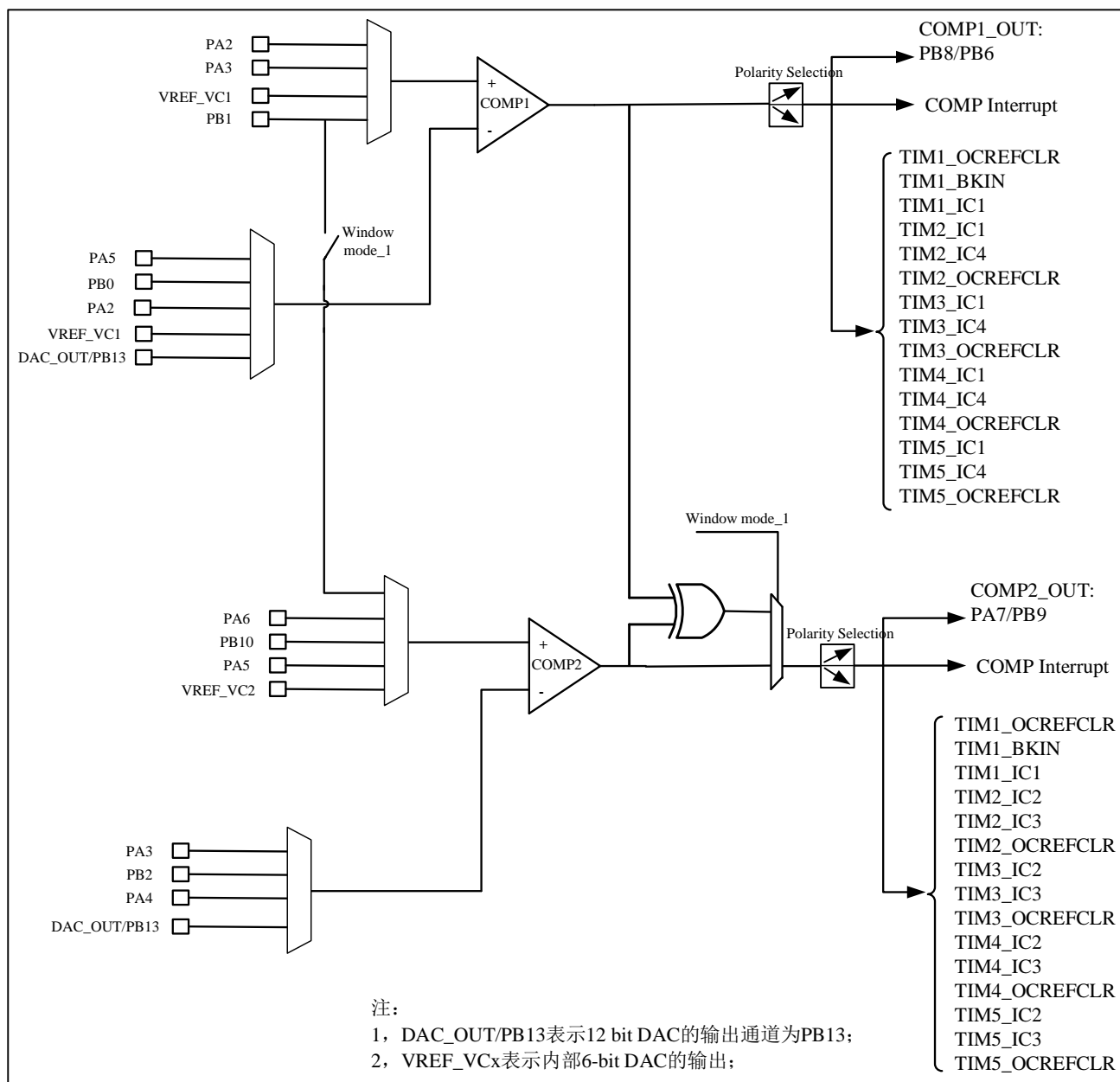
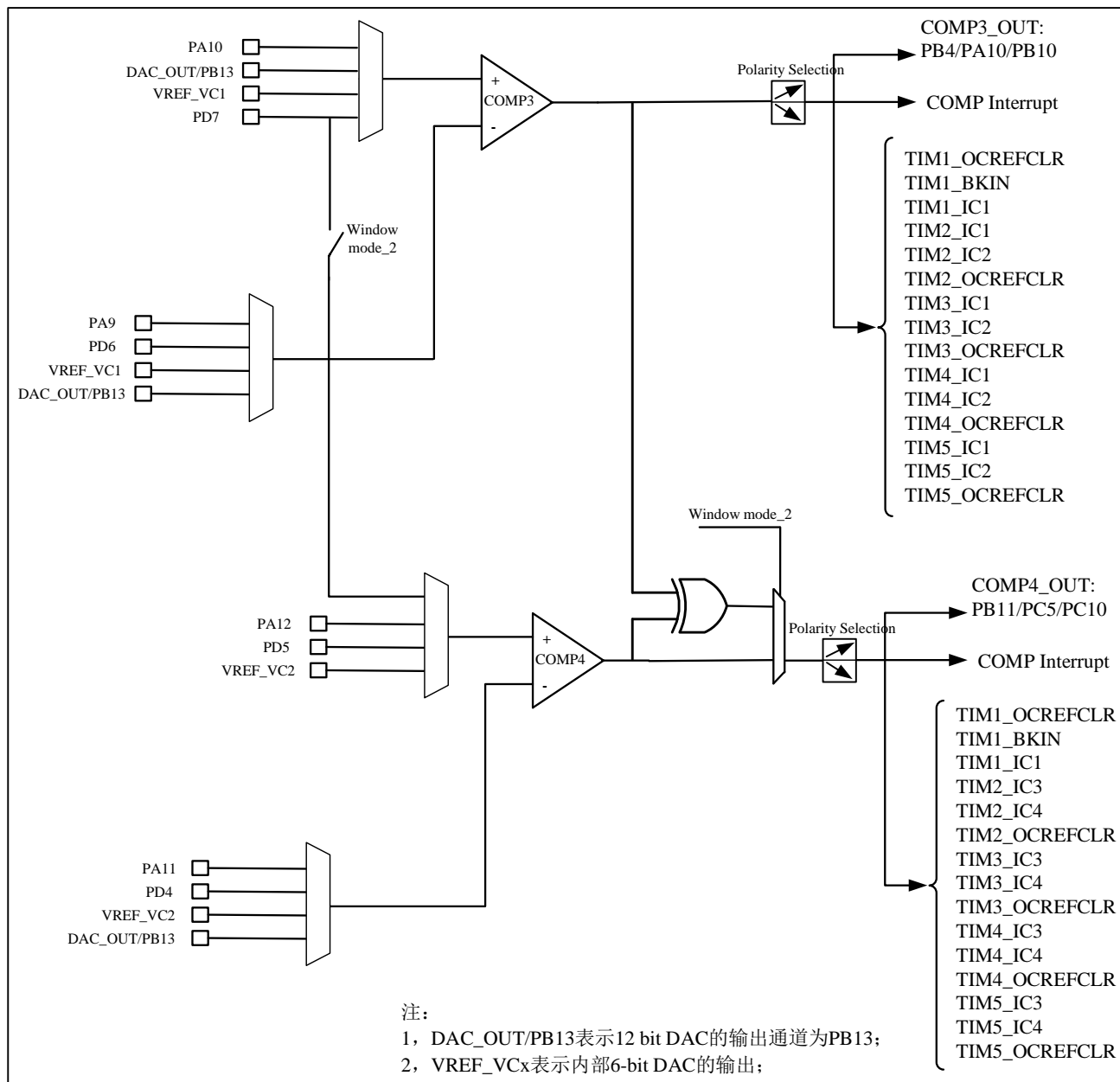


图 16-2 比较器 3 和比较器 4 系统连接图



16.2 COMP 特性

- 1 个独立的比较器
- 共享两个独立的 6bit DAC 的内部参考输入
- 支持滤波时钟，滤波复位
- 输出极性可配置高、低
- 迟滞配置可配置无、低、中、高
- 比较结果可输出到 I/O 端口或触发定时器，用于捕获事件、OCREF_CLR 事件、刹车事件、产生中断

- 输入通道可复选 I/O 端口、专用的 6bit DAC、通用的 12bit DAC 的通道输出
- 可配只读或读写，在锁定的情况下需要复位才能解锁
- 支持消隐（Blanking），可配置产生 Blanking 的消隐源
- COMP1/COMP2、COMP3/COMP4 可以组成窗口比较器
- 可通过产生中断的方式将系统从 Sleep 模式唤醒
- 可配置滤波窗口大小
- 可配置滤波阈值大小
- 可配置用于滤波的采样频率

16.3 COMP 配置流程

完整的配置项包括如下所示，某些项目如果采用系统默认配置，跳过相应的配置项。

- 可配置的迟滞等级 COMP_x_CTRL.HYST[1:0]。
- 配置输出极性 COMP_x_CTRL.POL。
- 配置输入选择，比较器正极 COMP_x_CTRL.INPSEL[3:0]，负极 COMP_x_CTRL.INMSEL[2:0]。
- 配置输出选择 COMP_x_CTRL.OUTSEL[3:0]。
- 配置消隐源 COMP_x_CTRL.BLKING[2:0]。
- 配置滤波器采样窗口 COMP_x_FILC.SAMPW[4:0]。
- 配置阈值 COMP_x_FILC.THRESH[4:0]（阈值应当大于 COMP_x_FILC.SAMPW[4:0]/2）。
- 配置滤波器采样频率（对于计时器应用，采样频率应当大于 5MHz。）
- 打开滤波器使能 COMP_x_FILC.FILEN。
- 打开比较器使能 COMP_x_CTRL.EN。

注: 对于以上步骤，需先打开滤波器使能，再打开比较器使能，比较器使能需要在滤波（若启用）配置、使能完成后启用，此外在比较器控制寄存器锁定 LOCK 的情况下，只有通过复位才能取消锁定。

16.4 COMP 工作模式

16.4.1 窗口比较器

比较器可以组合窗口比较器，比较器 1 和比较器 2 共享 PB1 组成窗口比较器，比较器 3 和比较器 4 共享 PD7 组成窗口比较器。

16.4.2 独立比较器

1 个比较器可独立配置，完成比较器功能。比较器的输出可以输出到 IO 端口，每一个比较器都有不同的重映射端口，通过配置可以选择比较器的输出，连接到相应的端口。

比较器输出，支持触发事件，比如可以配置成定时器 1 的刹车功能。

注：具体配置参考比较器互联关系

16.5 比较器互联关系

比较器输出端口的互联，可以参考 GPIO 的复用功能章节，定义了比较器 OUT 重映射的值。

比较器 OUT 引脚如下：

COMP1	COMP2	COMP3	COMP4
PB6	PA7	PB4	PB11
PB8	PB9	PA10	PC10
-	-	PB10	PC5

比较器 INP 引脚有如下配置：

INPSEL	COMP1	COMP2	COMP3	COMP4
000	PA2	PB1	PA10	PD7
001	PA3	PA6	DAC_OUT(PB13)	PA12
010	VREF_VC1	PB10	VREF_VC1	PD5
011	PB1	PA5	PD7	VREF_VC2
100	-	VREF_VC2	-	-

比较器 INM 引脚有如下配置：

INMSEL	COMP1	COMP2	COMP3	COMP4
000	PA5	PA3	PA9	PA11
001	PB0	PB2	PD6	PD4
010	PA2	PA4	VREF_VC1	VREF_VC2
011	VREF_VC1	DAC_OUT(PB13)	DAC_OUT(PB13)	DAC_OUT(PB13)
100	DAC_OUT(PB13)	-	-	-

比较器输出 TRIG 的信号有如下互联关系

TRIG	COMP1	COMP2	COMP3	COMP4
0000	TIM1_OCrefclear	TIM1_OCrefclear	TIM1_OCrefclear	TIM1_OCrefclear
0001	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN
0010	TIM1_IC1	TIM1_IC1	TIM1_IC1	TIM1_IC1
0011	TIM2_IC1	TIM2_IC2	TIM2_IC1	TIM2_IC3
0100	TIM2_IC4	TIM2_IC3	TIM2_IC2	TIM2_IC4
0101	TIM2_OCrefclear	TIM2_OCrefclear	TIM2_OCrefclear	TIM2_OCrefclear
0110	TIM3_IC1	TIM3_IC2	TIM3_IC1	TIM3_IC3
0111	TIM3_IC4	TIM3_IC3	TIM3_IC2	TIM3_IC4
1000	TIM3_OCrefclear	TIM3_OCrefclear	TIM3_OCrefclear	TIM3_OCrefclear
1001	TIM4_IC1	TIM4_IC2	TIM4_IC1	TIM4_IC3
1010	TIM4_IC4	TIM4_IC3	TIM4_IC2	TIM4_IC4
1011	TIM4_OCrefclear	TIM4_OCrefclear	TIM4_OCrefclear	TIM4_OCrefclear
1100	TIM5_IC1	TIM5_IC2	TIM5_IC1	TIM5_IC3

1101	TIM5_IC4	TIM5_IC3	TIM5_IC2	TIM5_IC4
1110	TIM5_OCrefclear	TIM5_OCrefclear	TIM5_OCrefclear	TIM5_OCrefclear
Other	-	-	-	-

16.6 比较器输出

COMP 输出支持 COMP_x_CTRL.OUT 输出滤波前的值与 COMP_x_CTRL.FLTOUT 滤波后的值。

- 滤波关闭时，COMP_x_CTRL.OUT 输出结果有效，COMP_x_CTRL.FLTOUT 的输出结果无效；
- 滤波开启时，COMP_x_CTRL.OUT、COMP_x_CTRL.FLTOUT 分别指示滤波前和滤波后的比较器输出结果。

16.7 中断

COMP 支持中断响应。中断产生有如下 2 种情况。

- COMP_x_CTRL.POL 极性不反转，中断使能，当 INPSEL>INMSEL 时，COMP_x_CTRL.OUT 由硬件置为 1 时即产生比较器中断。
- COMP_x_CTRL.POL 极性反转，中断使能，当 INPSEL<INMSEL 时，COMP_x_CTRL.OUT 由硬件置为 1 时即产生比较器中断。

16.8 COMP 寄存器

16.8.1 COMP 寄存器总览

表 16-1 COMP 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
010h	COMP1_CTRL	Reserved												FLT_OUT	OUT	BLKING[3:0]			HYST		POL	OUTSEL[3:0]			INPSEL[2:0]		INMSEL[3:0]			EN			
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	COMP1_FILC	Reserved																						SAMPW [4:0]			THRESH [4:0]			EN			
	Reset Value																							0	0	0	0	0	0	0	0	0	0
018h	COMP1_FILP	Reserved												CLKPSC[15:0]																			
	Reset Value																													0	0	0	0
01Ch	Reserved																																

020h	COMP2_CTRL	Reserved	FLT_OUT	OUT	BLKING[3:0]			HYST		POL	OUTSEL[3:0]			INPSEL[2:0]		INMSEL[3:0]			EN	
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
024h	COMP2_FILC	Reserved									SAMPW [4:0]			THRESH [4:0]			FILEN			
	Reset Value										0	0	0	0	0	0	0	0	0	
028h	COMP2_FILP	Reserved									CLKPSC[15:0]									
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
02Ch	Reserved																			
030h	COMP3_CTRL	Reserved	FLT_OUT	OUT	BLKING[3:0]			HYST		POL	OUTSEL[3:0]			INPSEL[2:0]		INMSEL[3:0]			EN	
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
034h	COMP3_FILC	Reserved									SAMPW [4:0]			THRESH [4:0]			FILEN			
	Reset Value										0	0	0	0	0	0	0	0	0	
038h	COMP3_FILP	Reserved									CLKPSC[15:0]									
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
03Ch	Reserved																			
040h	COMP4_CTRL	Reserved	FLT_OUT	OUT	BLKING[3:0]			HYST		POL	OUTSEL[3:0]			INPSEL[2:0]		INMSEL[3:0]			EN	
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
044h	COMP4_FILC	Reserved									SAMPW [4:0]			THRESH [4:0]			FILEN			
	Reset Value										0	0	0	0	0	0	0	0	0	
048h	COMP4_FILP	Reserved									CLKPSC[15:0]									
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
04Ch~06Ch	Reserved																			

070h	COMP_OSEL	Reserved														CMP4XO	CMP2XO		
	Reset Value															0	0		
074h~080h	Reserved																		
084h	COMP_LOCK	Reserved														CMP4LK	CMP3LK	CMP2LK	CMP1LK
	Reset Value															0	0	0	0
088h	Reserved																		
08Ch	COMP_INTEN	Reserved														CMP4IEN	CMP3IEN	CMP2IEN	CMP1IEN
	Reset Value															0	0	0	0
090h	COMP_INTSTS	Reserved														CMP4IS	CMP3IS	CMP2IS	CMP1IS
	Reset Value															0	0	0	0
094h	COMP_INVREF	Reserved						VV2TRM[5:0]				VV2EN	VV1TRM[5:0]				VV1EN		
	Reset Value							0	0	0	0	0	0	0	0	0	0	0	0
098h	COMP_OTIMEN	Reserved														CMP4OEN	CMP3OEN	CMP2OEN	CMP1OEN
	Reset Value															0	0	0	0

16.8.2 COMP1 控制寄存器（COMP1_CTRL）

偏移地址:0x10

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FLTOUT	OUT	BLKING[1:0]	
r												r	r	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKING[0]	HYST[1:0]		POL	OUTSEL[3:0]			INPSEL[2:0]			INMSEL[3:0]				EN	
rw	rw		rw	rw			rw			rw				rw	

位域	名称	描述
31:20	Reserved	保留，必须保持复位值。
19	FLTOUT	指示比较器 1 滤波后的输出状态 0: 输出低 1: 输出高
18	OUT	指示比较器 1 输出的状态 0: 输出低 1: 输出高
17:15	BLKING[2:0]	这些位选择哪个定时器输出控制比较器 1 输出消隐。 000: 不消隐 001: 选择 Tim1 OC5 作为消隐源 其他配置:保留
14:13	HYST[1:0]	这些位选择比较器 1 的迟滞等级。 00: 无迟滞; 01: 低迟滞; 10: 中等迟滞; 11: 高迟滞。
12	POL	该位用于反转比较器 1 的输出 0:输出未反转; 1:输出反转。
11:8	OUTSEL[3:0]	0000: TIM1_OCrefclear 0001: TIM1_BKIN 0010: TIM1_IC1 0011: TIM2_IC1 0100: TIM2_IC4 0101: TIM2_OCrefclear 0110: TIM3_IC1 0111: TIM3_IC4 1000: TIM3_OCrefclear 1001: TIM4_IC1 1010: TIM4_IC4 1011: TIM4_OCrefclear 1100: TIM5_IC1 1101: TIM5_IC4 1110: TIM5_OCrefclear 1111: Reserved
7:5	INPSEL[2:0]	比较器 1 正端选择位 000: PA2 001: PA3 010: VREF_VC1 011: PB1
4:1	INMSEL[3:0]	比较器 1 负端输入选择位 0000: PA5 0001: PB0

位域	名称	描述
		0010: PA2 0011: VREF_VC1 0100: DAC_OUT/PB13
0	EN	该位打开/关闭 COMP1 0: 比较器 1 已禁用; 1: 比较器 1 已启用。

16.8.3 COMP1 滤波控制寄存器 (COMP1_FILC)

偏移地址:0x14

复位值:0x0000 0000

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

位域	名称	描述
31:11	Reserved	保留，必须保持复位值。
10:6	SAMPW[4:0]	低通滤波器采样窗口大小，采样窗口 = SAMPW + 1。
5:1	THRESH[4:0]	低通滤波器门限置，样本窗口中至少出现相反状态的采样阈值，才能改变输出状态，此值要求大于 SAMPW / 2。
0	FILEN	滤波器使能位 0: 关闭; 1: 使能。

16.8.4 COMP1 滤波时钟寄存器 (COMP1_FILP)

偏移地址:0x18

复位值:0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CLKPSC[15:0]	低通滤波采样时钟预分频，系统时钟分频数 = CLKPSC + 1。 0：每 1 个时钟； 1：每 2 个时钟； 2：每 3 个时钟； ...

16.8.5 COMP2 控制寄存器（COMP2_CTRL）

偏移地址:0x20

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FLTOUT	OUT	BLKING[1:0]	
r												r	r	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKING[0]	HYST[1:0]		POL	OUTSEL[3:0]			INPSEL[2:0]			INMSEL[3:0]			EN		
rw	rw		rw	rw			rw			rw			rw		

位域	名称	描述
31:20	Reserved	保留，必须保持复位值。
19	FLTOUT	指示比较器 2 滤波后的输出状态 0：输出低 1：输出高
18	OUT	指示比较器 2 输出的状态 0：输出低 1：输出高
17:15	BLKING[2:0]	这些位选择哪个定时器输出控制比较器 2 输出消隐。。 000：不消隐 001：选择 Tim1 OC4 作为消隐源 其他配置:保留
14:13	HYST[1:0]	这些位选择比较器 2 的迟滞等级。 00：无迟滞； 01：低迟滞； 10：中等迟滞； 11：高迟滞。
12	POL	该位用于反转比较器 2 的输出 0:输出未反转； 1:输出反转。

位域	名称	描述
11:8	OUTSEL [3:0]	0000: TIM1_OCrefclear 0001: TIM1_BKIN 0010: TIM1_IC1 0011: TIM2_IC2 0100: TIM2_IC3 0101: TIM2_OCrefclear 0110: TIM3_IC2 0111: TIM3_IC3 1000: TIM3_OCrefclear 1001: TIM4_IC2 1010: TIM4_IC3 1011: TIM4_OCrefclear 1100: TIM5_IC2 1101: TIM5_IC3 1110: TIM5_OCrefclear 1111: Reserved
7:5	INPSEL[2:0]	比较器 2 正端选择位 000: PB1 001: PA6 010: PB10 011: PA5 100: VREF_VC2
4:1	INMSEL[3:0]	比较器 2 负端输入选择位 0000: PA3 0001: PB2 0010: PA4 0011: DAC_OUT/PB13
0	EN	该位打开/关闭 COMP2 0: 比较器 2 已禁用; 1: 比较器 2 已启用。

16.8.6 COMP2 滤波控制寄存器 (COMP2_FILC)

偏移地址:0x24

复位值:0x0000 0000

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

位域	名称	描述
31:11	Reserved	保留，必须保持复位值。
10:6	SAMPW[4:0]	低通滤波器采样窗口大小，采样窗口 = SAMPW + 1。
5:1	THRESH[4:0]	低通滤波器门限置，样本窗口中至少出现相反状态的采样阈值，才能改变输出状态，此值要求大于 SAMPW / 2。
0	FILEN	滤波器使能位 0：关闭； 1：使能。

16.8.7 COMP2 滤波时钟寄存器（COMP2_FILP）

偏移地址:0x28

复位值:0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15: 0	CLKPSC[15:0]	低通滤波采样时钟预分频，系统时钟分频数 = CLKPSC + 1。 0：每 1 个时钟； 1：每 2 个时钟； 2：每 3 个时钟； ...

16.8.8 COMP3 控制寄存器（COMP3_CTRL）

偏移地址:0x30

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FLTOUT	OUT	BLKING[1:0]	
r												r	r	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKING[0]	HYST[1:0]		POL	OUTSEL[3:0]			INPSEL[2:0]		INMSEL[3:0]			EN			

rw

rw

rw

rw

rw

rw

rw

位域	名称	描述
31:20	Reserved	保留，必须保持复位值。
19	FLTOUT	指示比较器 3 滤波后的输出状态 0: 输出低 1: 输出高
18	OUT	指示比较器 3 输出的状态 0: 输出低 1: 输出高
17:15	BLKING[2:0]	这些位选择哪个定时器输出控制比较器 3 输出消隐。 000: 不消隐 001: 选择 Tim1 OC4 作为消隐源 其他配置:保留
14:13	HYST[1:0]	这些位选择比较器 3 的迟滞等级。 00: 无迟滞; 01: 低迟滞; 10: 中等迟滞; 11: 高迟滞。
12	POL	该位用于反转比较器 3 的输出 0:输出未反转; 1:输出反转。
11:8	OUTSEL [3:0]	0000: TIM1_OCrefclear 0001: TIM1_BKIN 0010: TIM1_IC1 0011: TIM2_IC1 0100: TIM2_IC2 0101: TIM2_OCrefclear 0110: TIM3_IC1 0111: TIM3_IC2 1000: TIM3_OCrefclear 1001: TIM4_IC1 1010: TIM4_IC2 1011: TIM4_OCrefclear 1100: TIM5_IC1 1101: TIM5_IC2 1110: TIM5_OCrefclear 1111: Reserved
7:5	INPSEL[2:0]	比较器 3 正端选择位 000: PA10 001: DAC_OUT/PB13 010: VREF_VC1 011: PD7

位域	名称	描述
4:1	INMSEL[3:0]	比较器 3 负端输入选择位 0000: PA9 0001: PD6 0010: VREF_VC1 0011: DAC_OUT/PB13
0	EN	该位打开/关闭 COMP3 0: 比较器 3 已禁用; 1: 比较器 3 已启用。

16.8.9 COMP3 滤波控制寄存器 (COMP3_FILC)

偏移地址:0x34

复位值:0x0000 0000

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

位域	名称	描述
31:11	Reserved	保留, 必须保持复位值。
10:6	SAMPW[4:0]	低通滤波器采样窗口大小, 采样窗口 = SAMPW + 1。
5:1	THRESH[4:0]	低通滤波器门限置, 样本窗口中至少出现相反状态的采样阈值, 才能改变输出状态, 此值要求大于 SAMPW / 2。
0	FILEN	滤波器使能位 0: 关闭; 1: 使能。

16.8.10 COMP3 滤波时钟寄存器 (COMP3_FILP)

偏移地址:0x38

复位值:0x0000 0000

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

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15: 0	CLKPSC[15:0]	低通滤波采样时钟预分频，系统时钟分频数 = CLKPSC + 1。 0: 每 1 个时钟; 1: 每 2 个时钟; 2: 每 3 个时钟; ...

16.8.11 COMP4 控制寄存器 (COMP4_CTRL)

偏移地址:0x40

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												FLTOUT	OUT	BLKING[1:0]	
r												r	r	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKING[0]	HYST[1:0]		POL	OUTSEL[3:0]			INPSEL[2:0]			INMSEL[3:0]			EN		
rw	rw		rw	rw			rw			rw			rw		

位域	名称	描述
31:20	Reserved	保留，必须保持复位值。
19	FLTOUT	指示比较器 4 滤波后的输出状态 0: 输出低 1: 输出高
18	OUT	指示比较器 4 输出的状态 0: 输出低 1: 输出高
17:15	BLKING[2:0]	这些位选择哪个定时器输出控制比较器 4 输出消隐。。 000: 不消隐 001: 选择 Tim1 OC5 作为消隐源 其他配置:保留
14:13	HYST[1:0]	这些位选择比较器 4 的迟滞等级。 00: 无迟滞; 01: 低迟滞; 10: 中等迟滞; 11: 高迟滞。
12	POL	该位用于反转比较器 4 的输出 0:输出未反转;

位域	名称	描述
		1:输出反转。
11:8	OUTSEL [3:0]	0000: TIM1_OCrefclear 0001: TIM1_BKIN 0010: TIM1_IC1 0011: TIM2_IC3 0100: TIM2_IC4 0101: TIM2_OCrefclear 0110: TIM3_IC3 0111: TIM3_IC4 1000: TIM3_OCrefclear 1001: TIM4_IC3 1010: TIM4_IC4 1011: TIM4_OCrefclear 1100: TIM5_IC3 1101: TIM5_IC4 1110: TIM5_OCrefclear 1111: Reserved
7:5	INPSEL[2:0]	比较器 4 正端选择位 000: PD7 001: PA12 010: PD5 011: VREF_VC2
4:1	INMSEL[3:0]	比较器 4 负端输入选择位 0000: PA11 0001: PD4 0010: VREF_VC2 0011: DAC_OUT/PB13
0	EN	该位打开/关闭 COMP4 0: 比较器 4 已禁用; 1: 比较器 4 已启用。

16.8.12 COMP4 滤波控制寄存器 (COMP4_FILC)

偏移地址:0x44

复位值:0x0000 0000

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

位域	名称	描述
31:11	Reserved	保留，必须保持复位值。
10:6	SAMPW[4:0]	低通滤波器采样窗口大小，采样窗口 = SAMPW + 1。
5:1	THRESH[4:0]	低通滤波器门限置，样本窗口中至少出现相反状态的采样阈值，才能改变输出状态，此值要求大于 SAMPW / 2。
0	FILEN	滤波器使能位 0：关闭； 1：使能。

16.8.13 COMP4 滤波时钟寄存器（COMP4_FILP）

偏移地址:0x48

复位值:0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CLKPSC[15:0]	低通滤波采样时钟预分频，系统时钟分频数 = CLKPSC + 1。 0：每 1 个时钟； 1：每 2 个时钟； 2：每 3 个时钟； ...

16.8.14 COMP 输出选择寄存器（COMP_OSEL）

偏移地址:0x70

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CMP4XO	CMP2XO	

rw rw

位域	名称	描述
31:2	Reserved	保留，必须保持复位值。
1	CMP4XO	比较器 4 比较结果和比较器 3 输出异或后输出 0: COMP4 输出; 1: COMP3 和 COMP4 结果异或输出。
0	CMP2XO	比较器 2 比较结果和比较器 1 输出异或后输出 0: COMP2 输出; 1: COMP1 和 COMP2 结果异或输出。

16.8.15 COMP 锁寄存器 (COMP_LOCK)

偏移地址:0x84

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CMP4LK	CMP3LK	CMP2LK	CMP1LK
												rw	rw	rw	rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3	CMP4LK	仅写一次，是由软件控制的，只能通过系统重置来清除 设置此位可以将 COMP4_CTRL 设置为只读 0: COMP4_CTRL 是可读写的 1: COMP4_CTRL 是只读的
2	CMP3LK	仅写一次，是由软件控制的，只能通过系统重置来清除 设置此位可以将 COMP3_CTRL 设置为只读 0: COMP3_CTRL 是可读写的 1: COMP3_CTRL 是只读的
1	CMP2LK	仅写一次，是由软件控制的，只能通过系统重置来清除 设置此位可以将 COMP2_CTRL 设置为只读 0: COMP2_CTRL 是可读写的 1: COMP2_CTRL 是只读的
0	CMP1LK	仅写一次，是由软件控制的，只能通过系统重置来清除 设置此位可以将 COMP1_CTRL 设置为只读 0: COMP1_CTRL 是可读写的 1: COMP1_CTRL 是只读的

16.8.16 COMP 中断使能寄存器 (COMP_INTEN)

偏移地址:0x8C

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CMP4IEN	CMP3IEN	CMP2IEN	CMP1IEN
												rw	rw	rw	rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3	CMP4IEN	该位控制 COMP4 的中断启用 0: 禁用 1: 使用
2	CMP3IEN	该位控制 COMP3 的中断启用 0: 禁用 1: 使用
1	CMP2IEN	该位控制 COMP2 的中断启用 0: 禁用 1: 使用
0	CMP1IEN	该位控制 COMP1 的中断启用 0: 禁用 1: 使用

16.8.17 COMP 中断状态寄存器 (COMP_INTSTS)

偏移地址:0x90

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CMP4IS	CMP3IS	CMP2IS	CMP1IS
												rw	rw	rw	rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3	CMP4IS	COMP4 中断状态位，写 0 清除
2	CMP3IS	COMP3 中断状态位，写 0 清除
1	CMP2IS	COMP2 中断状态位，写 0 清除
0	CMP1IS	COMP1 中断状态位，写 0 清除

16.8.18 COMP 参考输入比较电压寄存器（COMP_INVREF）

偏移地址:0x94

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		VV2TRM[5:0]						VV2EN	VV1TRM[5:0]						VV1EN
		rw						rw	rw						rw

位域	名称	描述
31:14	Reserved	保留，必须保持复位值。
13:8	VV2TRM[5:0]	内部比较器 2 参考输入比较电压 VREF 档位选择 0~0b'111111 对应输出电压范围 0~VDDA，共 64 个档位，例如 7 则代表 $(7+1) * VDDA / 32 = 1/4 VDDA$
7	VV2EN	内部比较器 2 参考输入比较电压使能: 0: 禁用; 1: 启用,
6:1	VV1TRM[5:0]	内部比较器 1 参考输入比较电压 VREF 档位选择 0~0b'111111 对应输出电压范围 0~VDDA，共 64 个档位，例如 7 则代表 $(7+1) * VDDA / 32 = 1/4 VDDA$
0	VV1EN	内部比较器 1 参考输入比较电压使能: 0: 禁用; 1: 启用,

16.8.19 COMP 输出到定时器使能寄存器（COMP_OTIMEN）

偏移地址:0x98

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CMP4 OEN	CMP3 OEN	CMP2 OEN	CMP1 OEN
												rw	rw	rw	rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3	CMP4OEN	该位控制 COMP4 输出到定时器功能启用 0：禁用 1：使用
2	CMP3OEN	该位控制 COMP3 输出到定时器功能启用 0：禁用 1：使用
1	CMP2OEN	该位控制 COMP2 输出到定时器功能启用 0：禁用 1：使用
0	CMP1OEN	该位控制 COMP1 输出到定时器功能启用 0：禁用 1：使用

17 液晶显示控制器（LCD）

17.1 简介

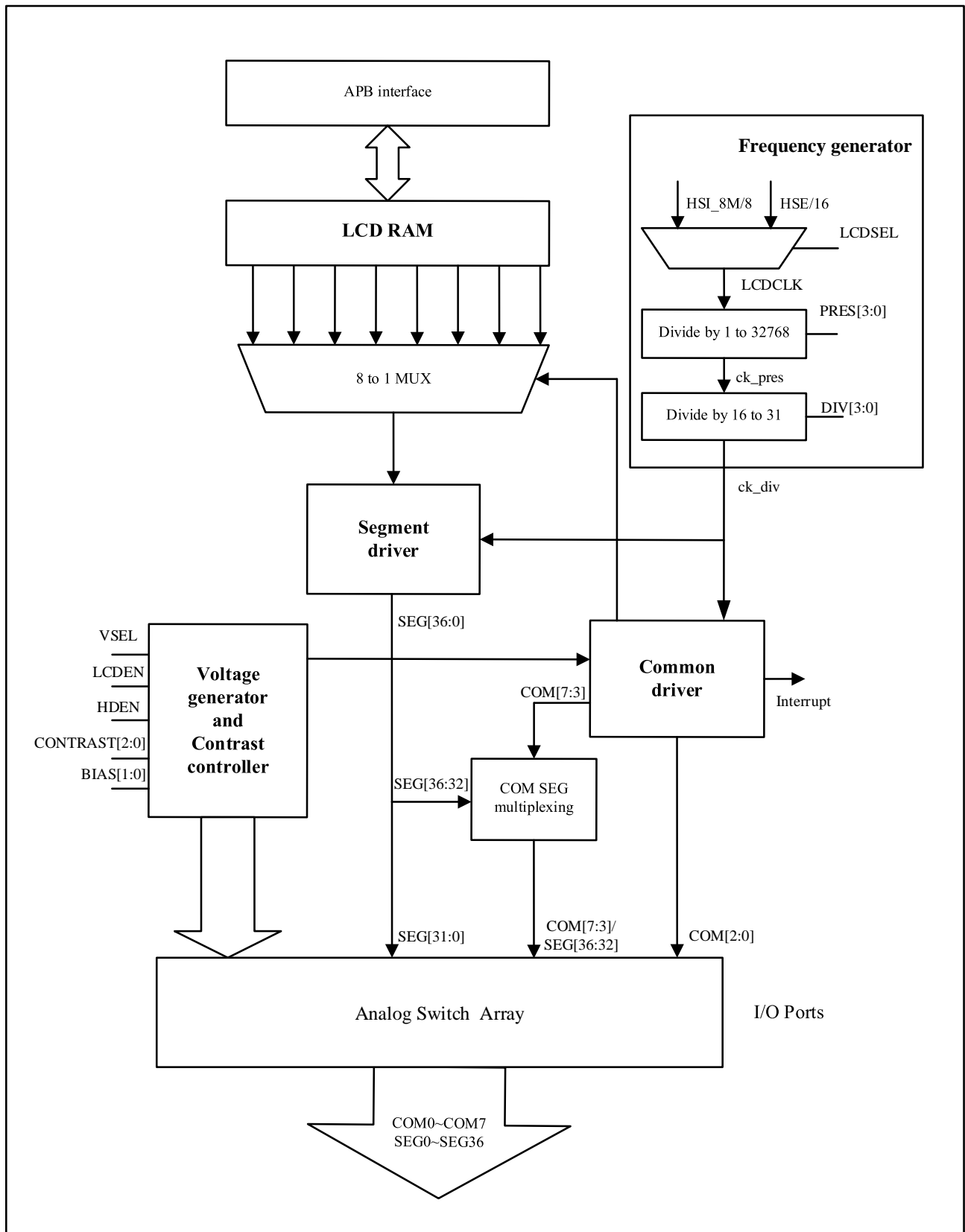
该 LCD 控制器适用于单色无源段式液晶显示器（Segment LCD），最多具有 8 个公共端子（COM）和 32 个区段端子（SEG），具体的端子数量取决于不同封装的引脚，具体可参考数据手册。Segment LCD 由若干区段组成，这些区段均可点亮或熄灭。每个区段都包含一层在两根电极之间对齐的液晶分子。当向液晶施加高于阈值电压的电压时，则对应的区段可见。为了避免液晶中出现电泳效应，区段电压必须为交流。

17.2 主要特性

- 帧速率可配置。
- 占空比可配置：支持静态、1/2、1/3、1/4 和 1/8 占空比。
- 电压偏置可配置：支持静态、1/2、1/3 和 1/4 偏置。
- 双缓冲机制允许用户随时更新显示内存寄存器中的数据（像素活动/非活动信息）。
- LCD 时钟源可选：HSE/16 或 HSI_8M/8
- 两种对比度控制方式：调节帧之间多达 7 个相位周期的死区时间；调节 V_{LCD} 在 $V_{LCDmin} \sim V_{LCDmax}$ 范围内变化（仅使用内部升压转换器时）。
- 内置电阻网络用来产生 LCD 中间电压，可通过软件配置，以匹配 LCD 面板上的电容性负载。
- 内置电压输出缓冲器
- 内置相位反转，可降低电磁干扰（EMI）和功耗。
- 支持闪烁功能：可配置 1、2、3、4、8 或所有像素在指定的频率（0.5Hz、1Hz、2Hz 或 4Hz）下闪烁
- 用作 SEG 和 COM 功能的管脚应配置为相应的 AFIO。

17.3 功能框图

图 17-1 LCD 控制器框图



17.4 功能描述

LCD 控制器提供了一个完全可配置的接口,以支持多种单色无源液晶显示器,帧频率可灵活配置。每个 COM 波形相同,但相位不同。COM 端口的数量取决于占空比配置,在一个帧中具有相同的波形,但每个相位只有一个 COM 处于活动状态。LCD 控制器支持多种偏置和占空比,以实现广泛的显示特性。

光学对比度是开段和关段透明度的差异,即对比度可以定义为开段和关段的 RMS 电压之间的差:

$$\text{光学对比度} = [V_{\text{on(rms)}} - V_{\text{off(rms)}}]$$

对比度还取决于开段电压 $V_{\text{on(rms)}}$ 和阈值电压 V_{th} 之间的差异。

其中, $V_{\text{on(rms)}}$ 和 $V_{\text{off(rms)}}$ 与用于驱动显示器的占空比有关。当驱动 LCD 所需的 COM 端数量增加时, $V_{\text{on(rms)}}$ 和 $V_{\text{off(rms)}}$ 之间的间隔增大,对比度降低。此外,对于更高的 V_{LCD} , 应使用更高的偏压水平,以更好地分离 $V_{\text{on(rms)}}$ 和 $V_{\text{off(rms)}}$, 从而获得更好的光学对比度。

17.4.1 频率发生器

频率发生器由一个预分频器和一个 16 至 31 时钟分频器组成。配置 LCD_FCTRL.PRES[3:0], 选择 LCDCLK 除以 $2^{\text{PRES}[3:0]}$ 。配置 LCD_FCTRL.DIV[3:0], 时钟进一步除以 16 到 31, 可以获得更好的分频。

频率发生器的输出时钟频率 $f_{\text{ck_div}}$ 是整个 LCD 控制器的时基。 $f_{\text{ck_div}}$ 是 LCD 相位频率,而不是帧频率(它们仅在静态占空比情况下相等)。帧频率(f_{frame})是通过 $f_{\text{ck_div}}$ 除以有效 COM 端的数量(或乘以占空比)获得的。

频率发生器的输入时钟频率 f_{LCDCLK} 与其输出时钟频率 $f_{\text{ck_div}}$ 之间的关系是:

$$f_{\text{ck_div}} = \frac{f_{\text{LCDCLK}}}{2^{\text{PRES}} \times (\text{DIV} + 16)}$$

输出时钟频率 $f_{\text{ck_div}}$ 与帧频率 f_{frame} 之间的关系是:

$$f_{\text{frame}} = f_{\text{ck_div}} \times \text{Duty}$$

通过控制 LCD_FCTRL.BLINKF[2:0](可配置为 0, 1, 2, ...7), 闪烁频率在 0.5Hz、1Hz、2Hz 或 4Hz 范围内。输出时钟频率 $f_{\text{ck_div}}$ 与闪烁频率 f_{BLINK} 之间的关系是:

$$f_{\text{BLINK}} = f_{\text{ck_div}} / 2^{(3+\text{BLINKF})}$$

帧频率计算示例如下表所示:

表 17-1 帧频计算示例

LCDCLK	PRES[3:0]	DIV[3:0]	Ratio	Duty	f_{frame}
1.00MHz	7	3	2432	1/8	51.4Hz
1.00MHz	8	3	4864	1/4	51.4Hz
1.00MHz	8	10	6656	1/3	50.1Hz
1.00MHz	9	3	9728	1/2	51.4Hz
1.00MHz	10	3	19456	static	51.4Hz

注意: 为了达到较低的功耗和较高的刷新率, 帧频率范围建议在 40Hz ~100Hz 内。

17.4.2 公共端驱动器

17.4.2.1 COM 信号偏置

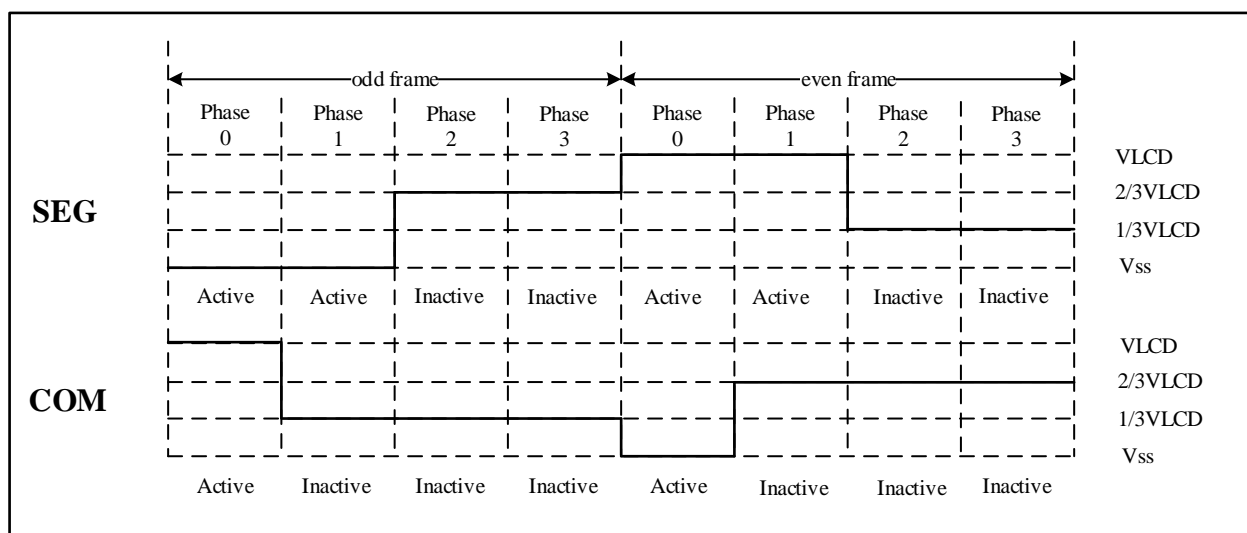
偏置为驱动 LCD 时使用的电压电平数。通过 LCD_CTRL.BIAS[1:0]选择 COM 信号偏置，可配置为 1/2 偏置、1/3 偏置、1/4 偏置。

COM[n]在 n 相位处于活动状态，COM pin 在奇数帧被驱动到 V_{LCD} ，在偶数帧被驱动到 VSS。COM[n]在其他相位不活动，因此在 1/3 或 1/4 偏置时：COM pin 在奇数帧被驱动到 1/3 或 1/4 V_{LCD} ，偶数帧被驱动到 2/3 或 3/4 V_{LCD} ；在 1/2 偏置时：无论是奇数帧还是偶数帧总是被驱动到 1/2 V_{LCD} 。

SEG 在 n 相位处于活动状态，SEG pin 在奇数帧被驱动到 VSS，在偶数帧被驱动到 V_{LCD} 。SEG 在不活动状态时，如在 1/3 偏置时：SEG pin 在奇数帧被驱动到 2/3 V_{LCD} ，偶数帧被驱动到 1/3 V_{LCD} ；在 1/4 偏置时：无论是奇数帧还是偶数帧总是被驱动到 1/2 V_{LCD} ；在 1/2 或静态偏置时：SEG pin 在奇数帧被驱动到 V_{LCD} ，偶数帧被驱动到 VSS。

当一个像素对应的 COM 和 SEG 在同一相位都处于活动状态时，COM 和 SEG 的电压差最大，像素被激活。如图 17-2 所示，通过相位反转减少电磁干扰，在每个奇数周期结束时，平均电压为 1/2 V_{LCD} 。

图 17-2 奇偶帧示例图（1/4 占空比，1/3 偏置）



注：此图为 COM0-SEG0, COM1-SEG0 导通，COM2-SEG0, COM3-SEG0 不导通状态。

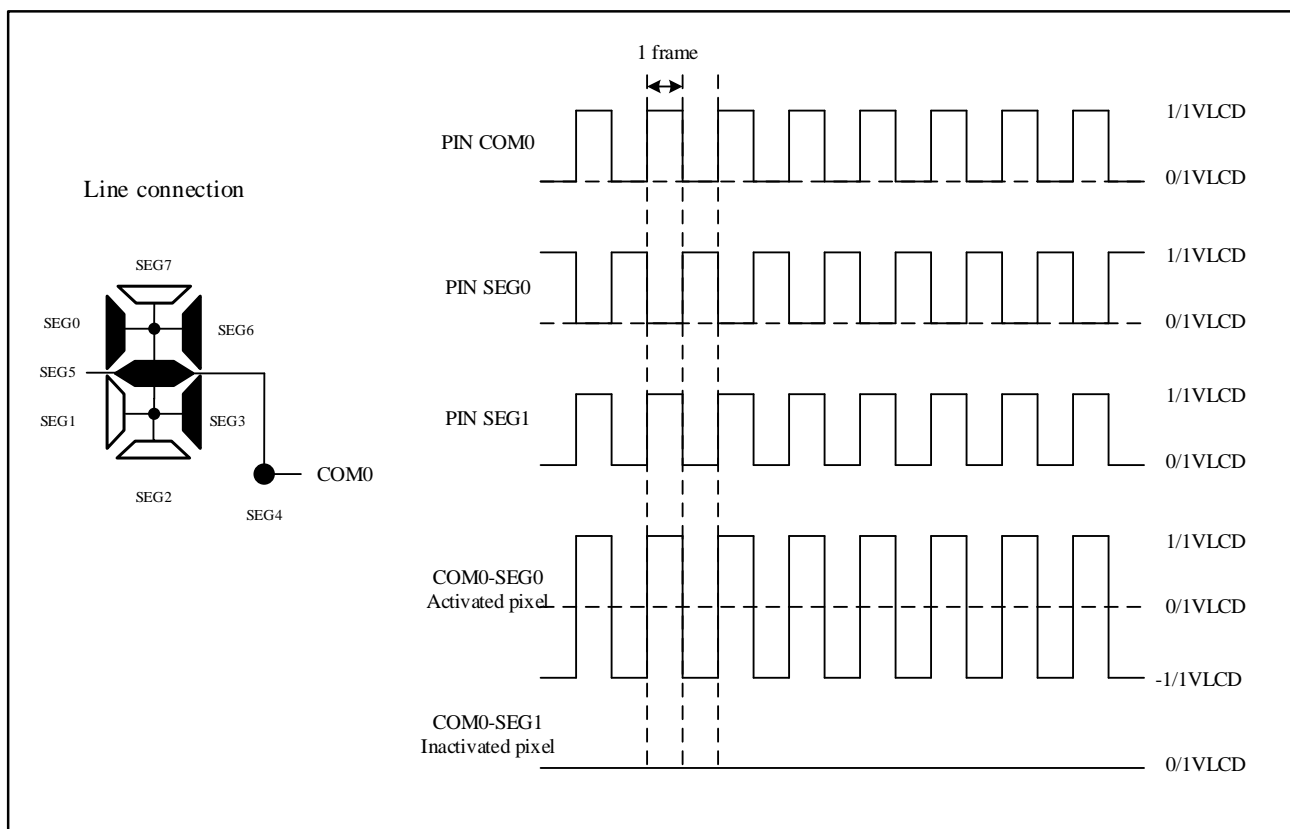
17.4.2.2 COM 信号占空比

占空比为 1/（给定 LCD 显示器上的公共端子数）。通过 LCD_CTRL.DUTY[2:0]选择 COM 信号占空比，可配置为静态占空比、1/2 占空比、1/3 占空比、1/4 占空比、1/8 占空比。举例如果选择 1/4 占空比，共 4 个 COM 可用，则一帧中有四个相位，其中 COM[0]在相位 0 期间处于活动状态，COM[1]在相位 1 期间处于活动状态，COM[2]在相位 2 期间处于活动状态，COM[3]在相位 3 期间处于活动状态。

当选择静态占空比时，COM[0]始终处于活动状态，而 COM[7:1]不使用并被驱动到 VSS，在每个帧中只有一个相位，因此 f_{frame} 等于 f_{LCD} 。此时 SEG 和 COM 只有 V_{LCD} 和 VSS 两个电压级别。如果对应的 SEG 端与 COM 端电压差为 0，则像素处于非活动状态，否则像素处于活动状态，此时液晶显示器的对比度最大。如图 17-3 所示，像素 0 处于活动状态，而像素 1 处于非活动状态。

当 LCD_CTRL.LCDEN 位被禁用时，所有 COM 被拉到 VSS，LCD_STS.ENSTS 标志变为 0。

图 17-3 静态占空比示例



17.4.2.3 八选一多路选择器

当 COM[0]激活时，COM 驱动模块驱动八选一多路选择器（参考图 17-1 LCD 控制器框图），选择前两个 RAM 寄存器的数据作为当前显示内容。当 COM[7]处于活动状态时，八选一多路选择器的输出是最后两个 RAM 寄存器的数据。

17.4.3 区段驱动器

如果像素 n 活动，则在奇数帧的 0 相位，SEG[n]管脚被驱动到 VSS；在偶数帧的 0 相位，SEG[n]管脚被驱动到 V_{LCD} 。

如果像素 n 不活动，SEG[n]管脚在奇数帧中被驱动到 $2/3$ ($2/4$) V_{LCD} ，偶数帧中被驱动到 $1/3$ ($2/4$) V_{LCD} ；如果偏置为 $1/2$ ，SEG[n]管脚在奇数帧被驱动到 V_{LCD} ，在偶数帧被驱动到 VSS。

当 LCD_CTRL.LCDEN 位被禁用时，所有 SEG 端口都被下拉到 VSS。

下面的图片展示了不同占空比和偏置情况下的波形

图 17-4 1/2 占空比, 1/2 偏置

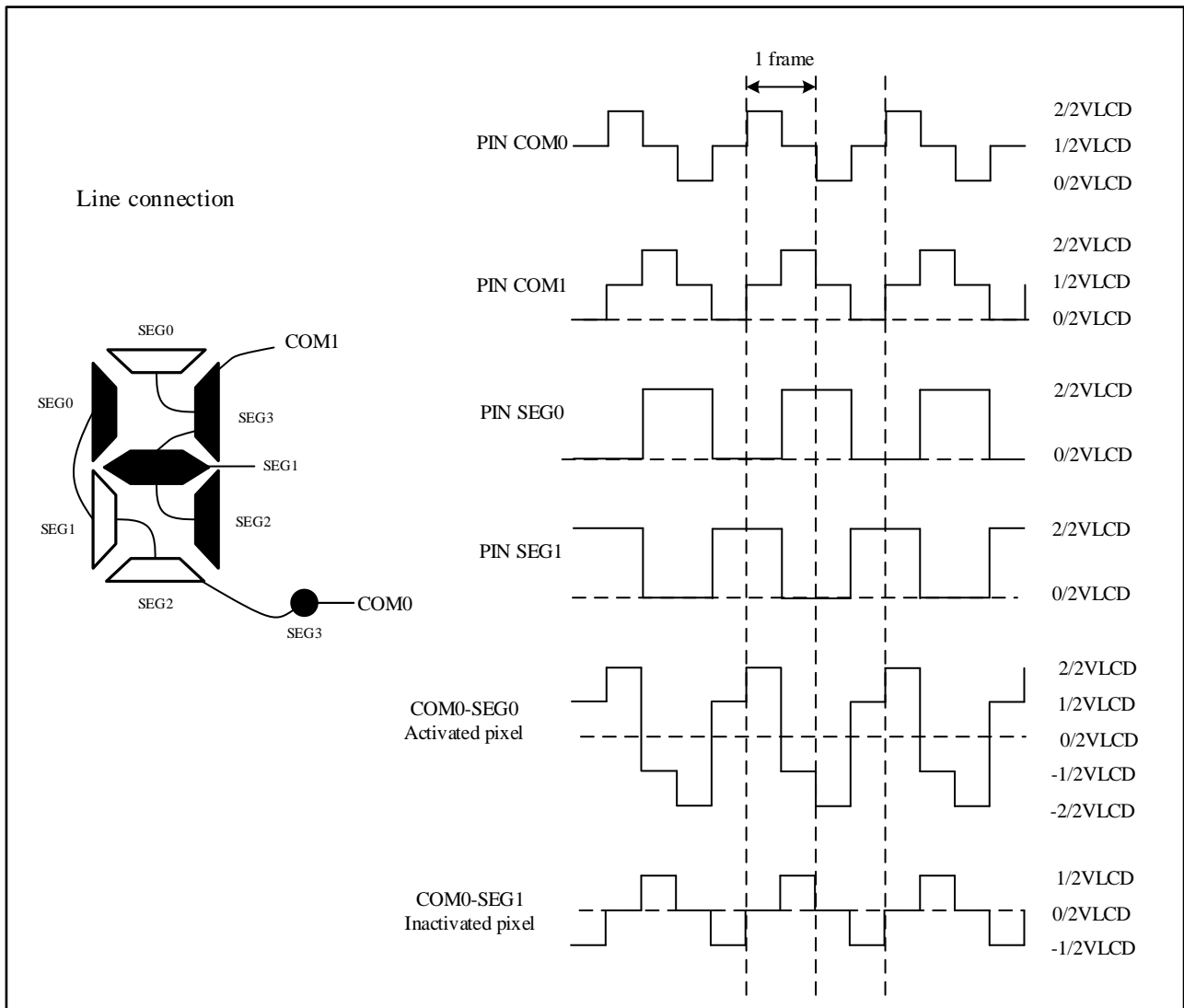
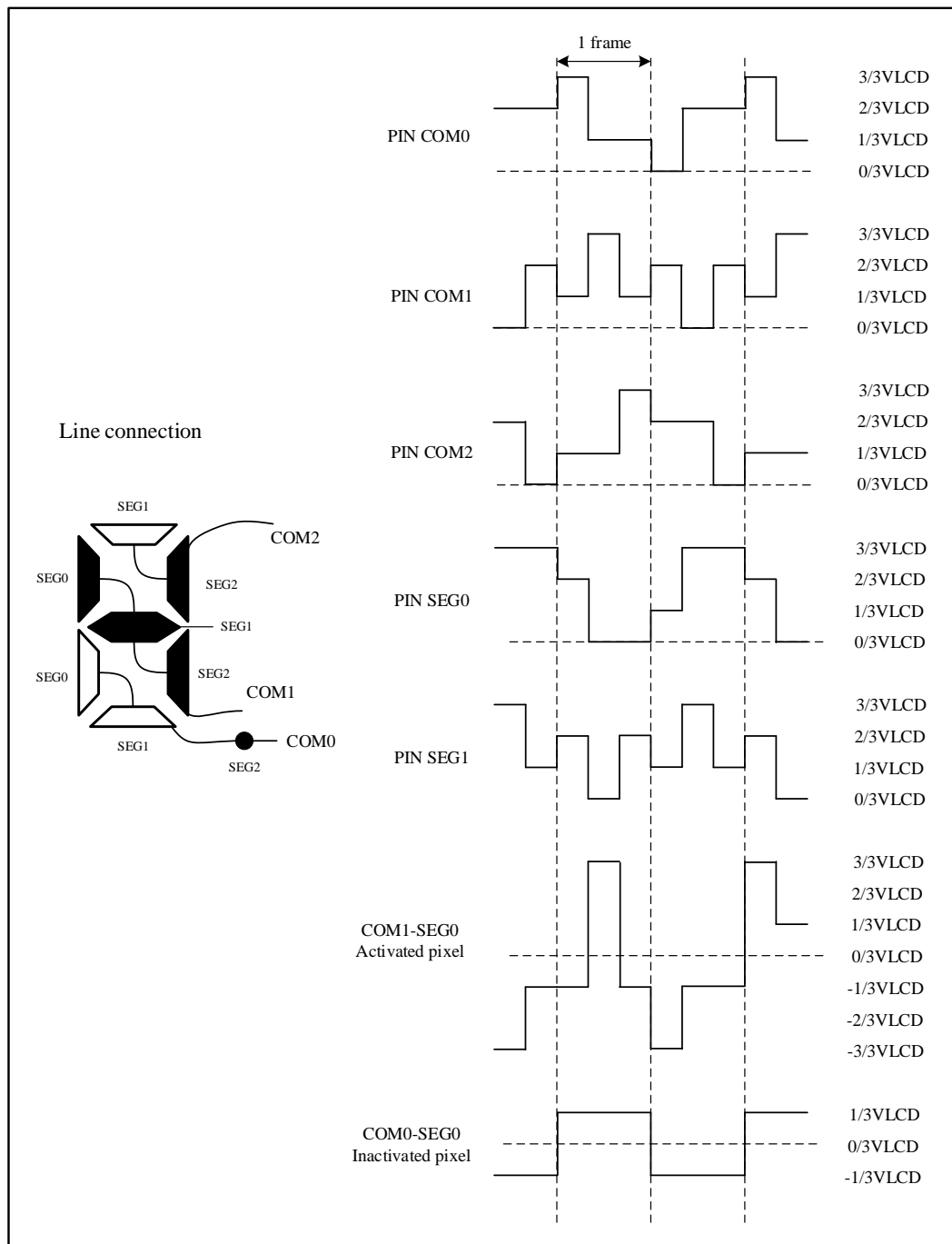


图 17-5 1/3 占空比, 1/3 偏置



注：图中未显示的COM2-SEG0 为导通状态

图 17-6 1/4 占空比, 1/3 偏置

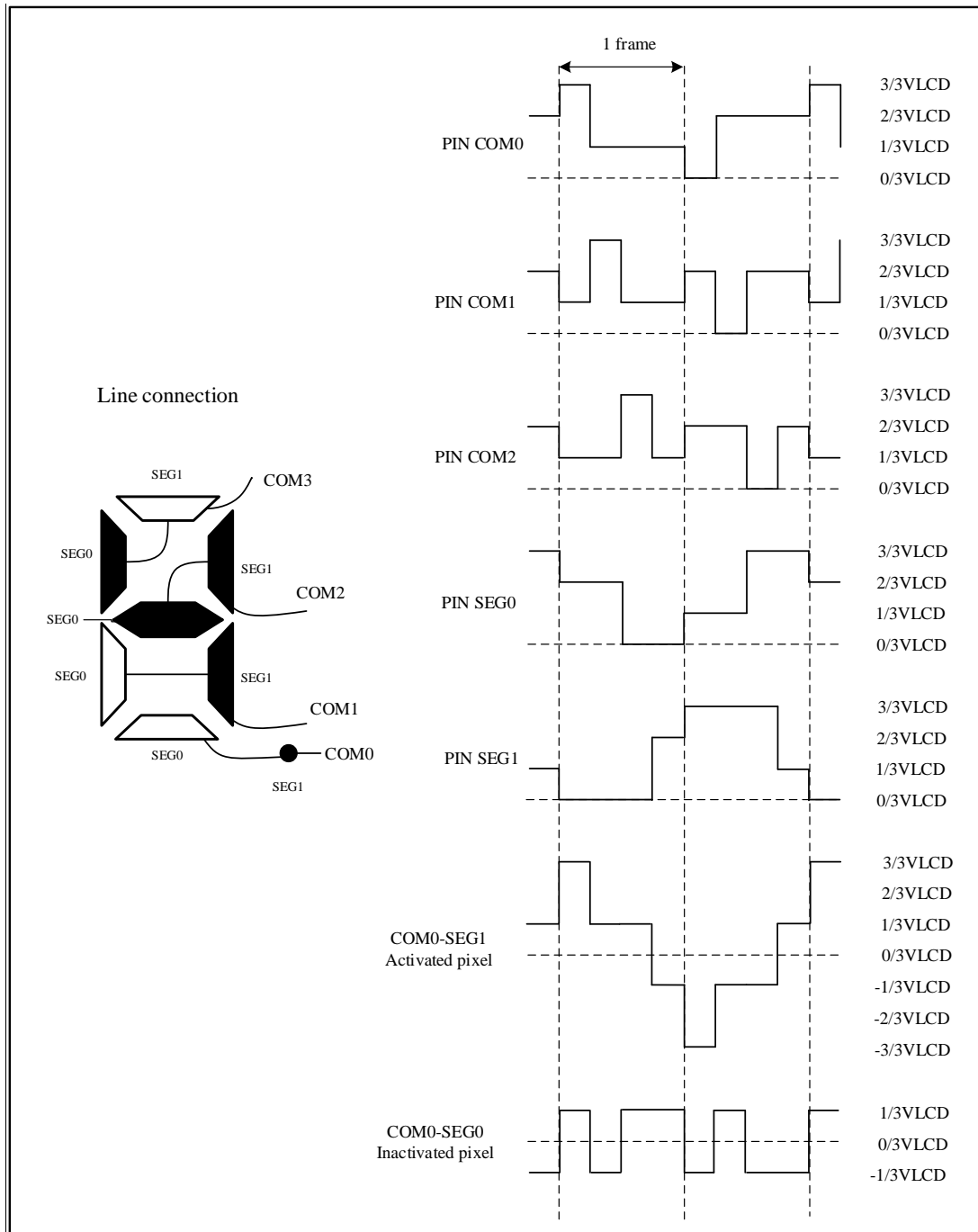
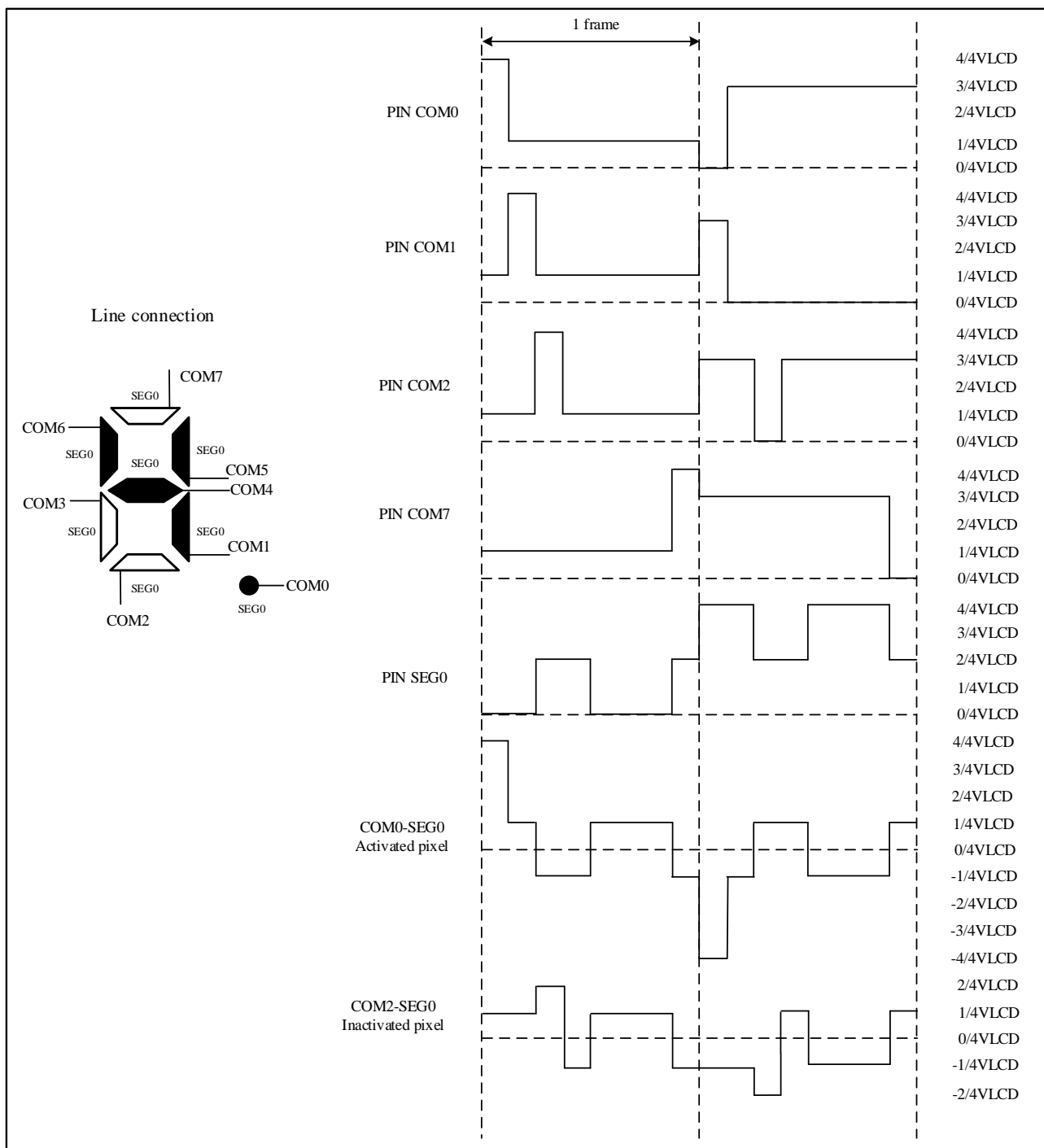


图 17-7 1/8 占空比, 1/4 偏置



17.4.3.1 闪烁功能

SEG 驱动器的闪烁功能可以使一些像素以特定的频率连续闪烁。通过 LCD_FCTRL.BLINK[1:0]位配置闪烁模式，使多达 1、2、3、4、8 或所有像素进行闪烁。通过 LCD_FCTRL.BLINKF[2:0]位配置闪烁频率，对 f_{ck_div} 进行分频。表 17-2 列出了不同闪烁频率的配置示例。

表 17-2 闪烁频率配置示例

BLINKF[2:0]			ck_div			
			32Hz	64Hz	128Hz	256Hz
0	0	0	4.0Hz	N/A	N/A	N/A
0	0	1	2Hz	4.0Hz	N/A	N/A
0	1	0	1Hz	2Hz	4.0Hz	N/A
0	1	1	0.5Hz	1Hz	2Hz	4.0Hz
1	0	0	0.25Hz	0.5Hz	1Hz	2Hz
1	0	1	N/A	0.25Hz	0.5Hz	1Hz
1	1	0	N/A	N/A	0.25Hz	0.5Hz
1	1	1	N/A	N/A	N/A	0.25Hz

17.4.4 电压发生器和对比度控制

17.4.4.1 电源选择

通过 LCD_CTRL.VSEL 配置选择 LCD 驱动对比度控制。当 VSEL=1 时，启用对比度控制 VLCD 可调节，此时可以通过 LCD_FCTRL.CONTRAST[3:0]位控制对比度在 V_{LCDmin} (2.6V) 到 V_{LCDmax} (5.0V) 范围， V_{LCD} 的新值在新的一帧开始时生效；当 VSEL=0 时，禁用对比度控制 $V_{LCD}=V_{DDA}$ ，时可通过调节帧间死区时间来控制对比度。

17.4.4.2 驱动选择

LCD 电压发生器通过内部电阻分压网络生成介于 V_{LCD} 与 VSS 之间的中间电压，如图 17-8 所示。

LCD 驱动内部有两个电阻网络，分别采用低值电阻 R_L 和高值电阻 R_H 来增加驱动电流或降低静态功耗。

对于 LCD_CTRL.LCDEN 位：

如果设置了 LCD_CTRL.LCDEN 位，则框图中的 LCDEN 开关闭合；当清除 LCD_CTRL.LCDEN 位时，框图中的 LCDEN 开关在偶数帧结束时断开，以避免在整个奇偶帧出现与 VSS 不同的中间电压。

对于 LCD_FCTRL.PULSEON[2:0]位：

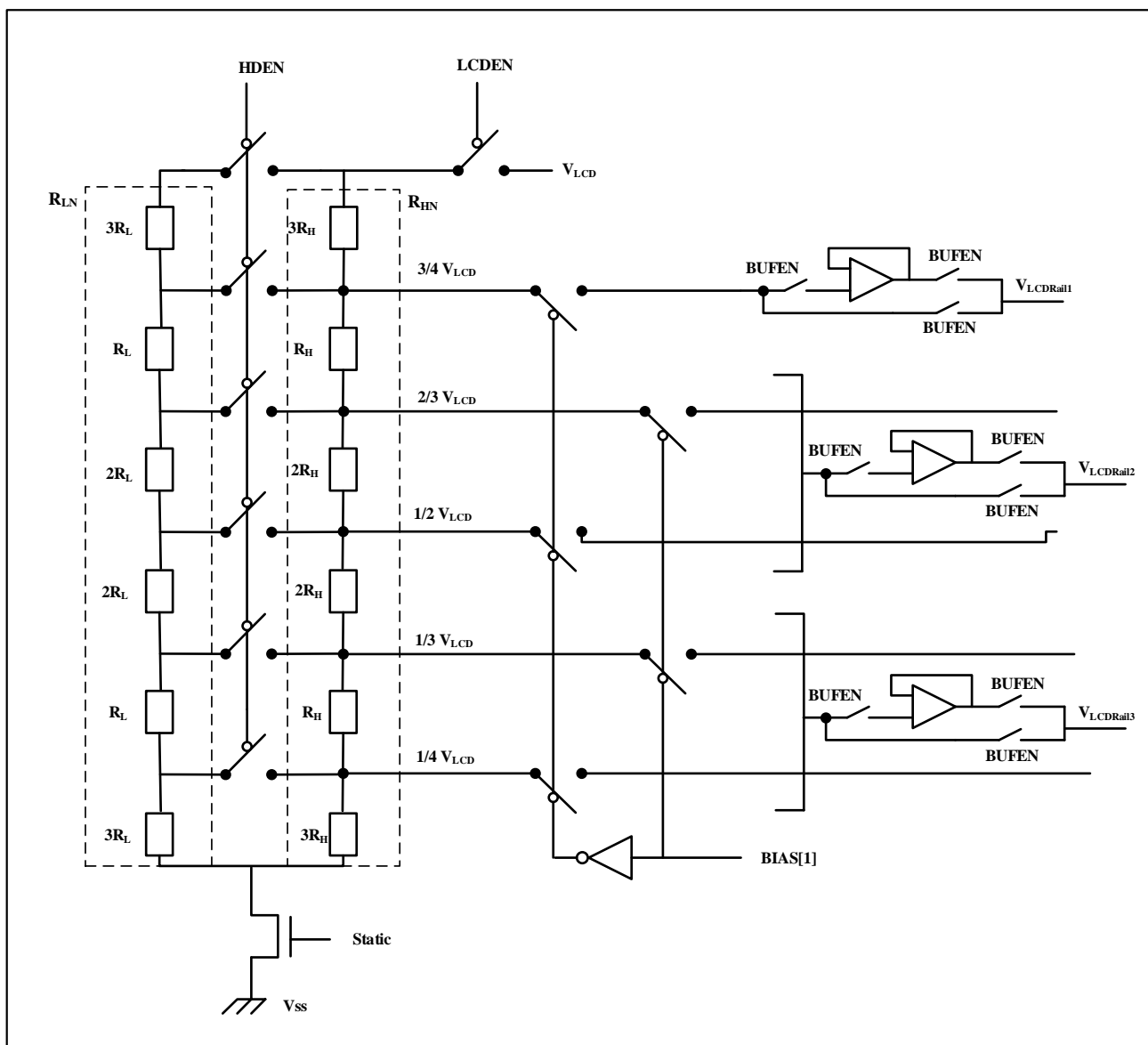
LCD_FCTRL.PULSEON[2:0]可配置在 COM 和 SEG 电平发生变化时，通过 HDEN 开关启用 R_L 的时间。驱动时间越短功耗越低，但具有高内阻的显示器需要较长的驱动时间才能获得好的对比度。

对于 LCD_FCTRL.HDEN 位：

如果 LCD_FCTRL.HDEN 位和 LCD_FCTRL.PULSEON[2:0]位被清零，则 HDEN 开关断开；如果 LCD_FCTRL.HDEN 位被清零，LCD_FCTRL.PULSEON[2:0]位不为 0，则 HDEN 开关在 LCD_FCTRL.PULSEON[2:0]位中定义的脉冲数期间闭合；如果 LCD_FCTRL 寄存器中的 HDEN 位为 1，则 HDEN 开关始终闭合。

R_{LN} 和 R_{HN} 分别表示低值电阻和高值电阻的电阻分压网络， R_{LN} 始终可以通过 LCD_FCTRL.HDEN 位打开。

图 17-8 LCD 驱动电压控制



17.4.4.3 电压缓冲模式

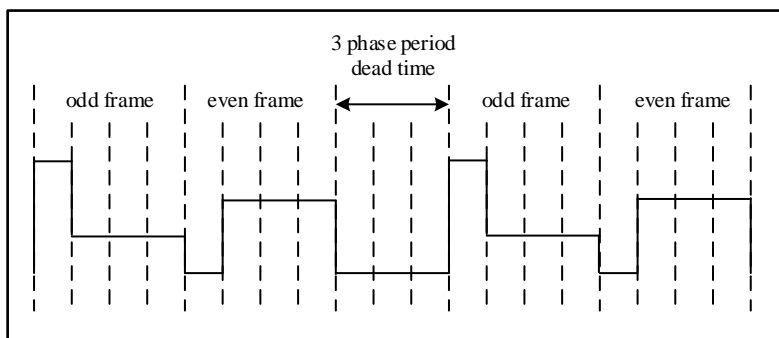
配置 LCD_CTRL.BUFEN 位（在 LCD_CTRL.LCDEN 禁能时配置）启用电压输出缓冲器时，高值电阻桥 R_{HN} 产生中间电压以降低功耗，低值电阻桥 R_{LN} 自动禁用（忽略 LCD_FCTRL.HDEN 位或 LCD_FCTRL.PULSEON 位配置情况）。LCD_CTRL.LCDEN 位置位后，LCD_STS.RDY 位在电压水平稳定后自动置位，LCD 控制器开始工作。

由于缓冲器阻止了 LCD 电容性负载直接加载至电阻桥网络并干扰其电压产生，LCD 驱动能力得到提高，中间电压更稳定，从而提高了施加到 LCD 像素上的均方根电压。

17.4.4.4 死区时间

通过设置 LCD_FCTRL.DEAD[2:0] 位，可以在每帧之间设置死区来控制对比度。在死区时间内，COM 和 SEG 端口被置为 V_{SS} 。

图 17-9 死区时间



17.4.5 二级缓冲显示

LCD 控制器内置双缓冲存储器，可确保显示信息的一致性，而不需要使用中断同步来控制 LCD_RAM 的修改。应用程序可以通过 APB 接口访问一级缓冲区 LCD_RAM，修改了 LCD_RAM 后，软件置位 LCD_STS.UDR 标志，硬件会将待更新的信息复制到二级缓冲区 LCD_DISPLAY。更新操作与帧同步完成（在下一帧开始时），直到更新完成，此过程中 LCD_RAM 写保护并且 LCD_STS.UDR 标志保持置位状态。更新完成后，LCD_STS.UDD 自动置位，如果 LCD_FCTRL.UDDIE 位使能，将产生中断。更新操作最多需要两帧的时间。在 LCD_CTRL.LCDEN=1 之前，不会进行更新操作（LCD_STS.UDR=1 和 LCD_STS.UDD=0）。

17.4.6 COM 和 SEG 多路复用

全部输出管脚包括：SEG[36:0]和 COM[2:0]。

通过配置 LCD_CTRL.DUTY[2:0]自动选择 SEG 管脚数量。在静态、1/2 和 1/3 占空比模式中，最多有 37 个 SEG 管脚，COM 管脚分别为 1、2 和 3 个；在 1/4 占空比模式，最多有 36 个 SEG 管脚，SEG[36]可作为 COM[3]使用；在 1/8 占空比模式，最多有 32 个 SEG 管脚，SEG[36:32]可作为 COM[3:7]使用。

COM 和 SEG 功能与占空比及管脚重映射配置的关系如表 17-3 所示。

表 17-3 COM 与 SEG 管脚映射表

配置位（DUTY）	SEG × COM	芯片输出引脚	LCD 模块功能
1/8	32×8	SEG[32]/COM[7]	COM[7]
		SEG[33]/COM[6]	COM[6]
		SEG[34]/COM[5]	COM[5]
		SEG[35]/COM[4]	COM[4]
		SEG[36]/COM[3]	COM[3]
		COM[2:0]	COM[2:0]
		SEG[31:0]	SEG[31:0]

1/4	36×4	SEG[32]/COM[7]	SEG[32]
		SEG[33]/COM[6]	SEG[33]
		SEG[34]/COM[5]	SEG[34]
		SEG[35]/COM[4]	SEG[35]
		SEG[36]/COM[3]	COM[3]
		COM[2:0]	COM[2:0]
		SEG[31:0]	SEG[31:0]
1/3	37×3	SEG[32]/COM[7]	SEG[32]
		SEG[33]/COM[6]	SEG[33]
		SEG[34]/COM[5]	SEG[34]
		SEG[35]/COM[4]	SEG[35]
		SEG[36]/COM[3]	SEG[36]
		COM[2:0]	COM[2:0]
		SEG[31:0]	SEG[31:0]
1/2	37×2	SEG[32]/COM[7]	SEG[32]
		SEG[33]/COM[6]	SEG[33]
		SEG[34]/COM[5]	SEG[34]
		SEG[35]/COM[4]	SEG[35]
		SEG[36]/COM[3]	SEG[36]
		COM[2]	未使用
		COM[1:0]	COM[1:0]
		SEG[31:0]	SEG[31:0]
STATIC	37×1	SEG[32]/COM[7]	SEG[32]
		SEG[33]/COM[6]	SEG[33]

		SEG[34]/COM[5]	SEG[34]
		SEG[35]/COM[4]	SEG[35]
		SEG[36]/COM[3]	SEG[36]
		COM[2:1]	未使用
		COM[0]	COM[0]
		SEG[31:0]	SEG[31:0]

17.5 工作流程

LCD 控制器工作流程如下：

1. 配置 LCD 模块参数，时钟源，COM/SEG 端口；
2. 将默认数据写入 LCD_RAM，并软件置位 LCD_STS.UDR；
3. 配置好帧频率及对比度后，置位 LCD_CTRL.LCDEN，使能 LCD 模块；
4. 若需要调节对比度，可以修改 LCD_FCTRL.PRES[3:0]、LCD_FCTRL.DIV[3:0]、LCD_FCTRL.CONTRAST[3:0]、LCD_FCTRL.PULSEON[2:0]、LCD_FCTRL.DEAD[2:0] 或 LCD_FCTRL.HDEN 位；
5. 若需要修改显示数据，需要先判断 LCD_STS.UDR，为 1 则需要等待；为 0 则可以更新数据到 LCD_RAM，然后软件置位 LCD_STS.UDR；每两次 LCD_STS.UDR 置位操作至少需要间隔数个 f_{ck_div} 周期，间隔周期数与使用 COM 数量有关，即 LCD_CTRL.DUTY[2:0] 配置有关，具体见下表；
6. 若需要修改闪烁像素及频率，可以修改 LCD_FCTRL.BLINK[1:0] 和 LCD_FCTRL.BLINKF[2:0]。

LCD_CTRL.DUTY[2:0] 配置	间隔周期数
000: 静态 LCD;	4
001: 1/2 duty;	2
010: 1/3 duty;	3
011: 1/4 duty;	4
100: 1/8 duty;	8

17.6 中断请求

LCD 中断请求如下表所示：

中断事件	标志	中断使能控制位	清除标志/中断方法
帧起始中断 (Start of frame)	LCD_STS.SOF	LCD_FCTRL.SOFIE	LCD_CLR.SOFCLR 写 1
显示更新完成中断 (Update display done)	LCD_STS.UDD	LCD_FCTRL.UDDIE	LCD_CLR.UDDCLR 写 1

17.7 LCD 控制器寄存器

LCD 相关寄存器必须按字（32 bits）进行读写操作

17.7.1 LCD 控制器寄存器总览

表 17-4 LCD 控制寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	LCD_CTRL	Reserved																							BUFEN	Reserved	BIAS[1:0]		DUTY[2:0]		VSEL	LCDEN	
	Reset Value																								0		0	0	0	0	0	0	
004h	LCD_FCTR	Reserved					PRES[3:0]				DIV[3:0]				BLINK [1:0]		BLINKF[2:0]			CONTRAST[3:0]			DEAD[2:0]		PULSEON[2:0]			UDDIE		Reserved	SOFIE	HDEN	
	L						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
008h	LCD_STS	Reserved																										FCRSF	RDY	UDD	UDR	SOF	ENSTS
	Reset Value																											0	0	0	0	0	0
00Ch	LCD_CLR	Reserved																											UDDCLR		Reserved	SOFCLR	Reserved
	Reset Value																																
014h	LCD_RAM1 _COM0	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
018h	LCD_RAM2 _COM0	Reserved																										S36	S35	S34	S33	S32	
	Reset Value																											0	0	0	0	0	
01Ch	LCD_RAM1 _COM1	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
020h	LCD_RAM2 _COM1	Reserved																										S36	S35	S34	S33	S32	
	Reset Value																											0	0	0	0	0	
024h	LCD_RAM1 _COM2	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
028h	LCD_RAM2 _COM2	Reserved																										S36	S35	S34	S33	S32	
	Reset Value																											0	0	0	0	0	
02Ch	LCD_RAM1 _COM3	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
030h	LCD_RAM2 _COM3	Reserved																										S36	S35	S34	S33	S32	
	Reset Value																											0	0	0	0	0	
034h	LCD_RAM1 _COM4	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

038h	LCD_RAM2 _COM4	Reserved																								S36	S35	S34	S33	S32			
	Reset Value																									0	0	0	0	0			
03Ch	LCD_RAM1 _COM5	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
040h	LCD_RAM2 _COM5	Reserved																								S36	S35	S34	S33	S32			
	Reset Value																									0	0	0	0	0			
044h	LCD_RAM1 _COM6	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
048h	LCD_RAM2 _COM6	Reserved																								S36	S35	S34	S33	S32			
	Reset Value																									0	0	0	0	0			
04Ch	LCD_RAM1 _COM7	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
050h	LCD_RAM2 _COM7	Reserved																								S36	S35	S34	S33	S32			
	Reset Value																									0	0	0	0	0			

17.7.2 LCD 控制寄存器 (LCD_CTRL)

偏移地址: 0x00

复位值: 0x0000 0000

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

位域	名称	描述
31:9	Reserved	保留, 必须保持复位值。
8	BUFEN	Voltage output buffer enable(Buffer 使能) 此位用于使能或禁用高驱动能力电压输出。 0: 禁用; 1: 使能。

位域	名称	描述
7	Reserved	保留，必须保持复位值。
6:5	BIAS[1:0]	Bias selector（偏压比选择） 00: 1/2 Bias; 01: 1/3 Bias; 10: 1/4 Bias; 11: 保留。
4:2	DUTY[2:0]	Duty selection（占空比选择） 000: 静态 LCD; 001: 1/2 duty; 010: 1/3 duty; 011: 1/4 duty; 100: 1/8 duty; 其他: 保留。
1	VSEL	LCD 驱动对比度控制 0: 禁用对比度控制 VLCD=VDDA 1: 启用对比度控制 VLCD 可调节
0	EN	LCD controller enable（LCD 控制器使能位） 该位由软件设置或清零来使能或禁用 LCD 控制器。软件清零会在下一帧开始前关闭 LCD，禁用后所有 COM 和 SEG 下拉到 VSS。 0: 禁用; 1: 使能。

注意: LCD_CTRL.VSEL、LCD_CTRL.DUTY、LCD_CTRL.BIAS 和 LCD_CTRL.BUFEN 位在 LCD_CTRL.LCDEN 使能后处于写保护状态，如果要修改，必须先禁用 LCD 控制器。

17.7.3 LCD 帧控制寄存器（LCD_FCTRL）

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PRES[3:0]				DIV[3:0]				BLINK[1:0]		BLINKF[2]
					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
BLINKF[1:0]		CONTRAST[3:0]				DEAD[2:0]			PULSEON[2:0]			UDDIE	Reserved	SOFIE	HDEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位域	名称	描述
31:27	Reserved	保留，必须保持复位值。
26:23	PRES[3:0]	16-bit prescaler（16 位分频器） $ck_pres = LCDCLK/2^n$

位域	名称	描述
		0000: $ck_pres = LCDCLK$; 0001: $ck_pres = LCDCLK/2$; 0010: $ck_pres = LCDCLK/4$; ... 1111: $ck_pres = LCDCLK/32768$ 。
22:19	DIV[3:0]	DIV clock divider (时钟分频器) $ck_div = ck_pres/(16+n)$ 0000: $ck_div = ck_pres/16$; 0001: $ck_div = ck_pres/17$; 0010: $ck_div = ck_pres/18$; ... 1111: $ck_div = ck_pres/31$ 。
18:17	BLINK[1:0]	Blink mode selection (闪烁模式选择) 00: 禁用闪烁; 01: 使能 SEG[0], COM[0] 闪烁 (1 个像素); 10: 使能 SEG[0], 所有 COM 闪烁 (最多 8 个像素, 取决于占空比); 11: 使能所有的 SEG 和 COM 闪烁 (所有像素)。
16:14	BLINKF[2:0]	Blink frequency selection (闪烁频率选择) 000: $ck_div / 8$; 001: $ck_div / 16$; 010: $ck_div / 32$; 011: $ck_div / 64$; 100: $ck_div / 128$; 101: $ck_div / 256$; 110: $ck_div / 512$; 111: $ck_div / 1024$ 。
13:10	CONTRAST[3:0]	Contrast Control (对比度控制) 这些位指定 VLCD 最大电压 (独立于 VDD), 范围在 2.60V~5.0V。 当 LCD_FCTRL.HDEN = 0 时: 0000: $VLCD = VDD * 1.00$; 0001: $VLCD = VDD * 0.97$; 0010: $VLCD = VDD * 0.94$; 0011: $VLCD = VDD * 0.90$; 0100: $VLCD = VDD * 0.87$; 0101: $VLCD = VDD * 0.84$; 0110: $VLCD = VDD * 0.80$; 0111: $VLCD = VDD * 0.77$; 1000: $VLCD = VDD * 0.74$; 1001: $VLCD = VDD * 0.70$; 1010: $VLCD = VDD * 0.67$; 1011: $VLCD = VDD * 0.64$; 1100: $VLCD = VDD * 0.61$; 1101: $VLCD = VDD * 0.58$; 1111: $VLCD = VDD * 0.55$ 。

位域	名称	描述
		<p>1110: VLCD = VDD*0.54; 1111: VLCD = VDD*0.50。 当 LCD_FCTRL.HDEN = 1 时: 0000: VLCD = VDD*1.00; 0001: VLCD = VDD*0.96; 0010: VLCD = VDD*0.92; 0011: VLCD = VDD*0.88; 0100: VLCD = VDD*0.84; 0101: VLCD = VDD*0.80; 0110: VLCD = VDD*0.76; 0111: VLCD = VDD*0.73; 1000: VLCD = VDD*0.70; 1001: VLCD = VDD*0.66; 1010: VLCD = VDD*0.63; 1011: VLCD = VDD*0.60; 1100: VLCD = VDD*0.57; 1101: VLCD = VDD*0.53; 1110: VLCD = VDD*0.50; 1111: VLCD = VDD*0.47。</p>
9:7	DEAD[2:0]	<p>Dead time duration (死区持续时间) 这些位通过软件配置帧之间的死区时间。在死区时间期间, COM 和 SEG 电压电平稳定为 0V, 使在不修改帧速率的情况下降低对比度。 000: 无死区时间; 001: 1 相周期死区时间; 010: 2 相周期死区时间; 011: 3 相周期死区时间; 100: 4 相周期死区时间; 101: 5 相周期死区时间; 110: 6 相周期死区时间; 111: 7 相周期死区时间。</p>
6:4	PULSEON[2:0]	<p>Pulse on duration (脉冲开启持续时间) 该位在 ck_pres 基础上配置脉冲持续时间。 000: 0; 001: 1/ck_pres; 010: 2/ck_pres; 011: 3/ck_pres; 100: 4/ck_pres; 101: 5/ck_pres; 110: 6/ck_pres; 111: 7/ck_pres。 注: 脉冲不要超过 LCD 时钟周期的一半。</p>
3	UDDIE	<p>Update display done interrupt enable (显示更新完成中断使能位) 该位由软件设置和清除。</p>

位域	名称	描述
		0: 禁用; 1: 使能。
2	Reserved	保留, 必须保持复位值。
1	SOFIE	Start of frame interrupt enable (帧开始中断使能位) 该位由软件设置和清除。 0: 禁用; 1: 使能。
0	HDEN	High drive enable (高驱动使能位) 该位可使能低电阻分压器。 0: 永久高驱动禁用; 1: 永久高驱动使能。 注: 当 HDEN 为 1 时, LCD_FCTRL.PULSEON 位必须配置为 001。

注意: LCD_FCTRL 寄存器可随时修改, 新的配置在下一帧起始时生效 (LCD_FCTRL.UDDIE 与 LCD_FCTRL.SOFIE 位除外, 修改后立即生效)。当 LCD_CTRL.BUFEN 位置 1 后, 低阻值电阻网络自动禁用, LCD_FCTRL.HDEN 和 LCD_FCTRL.PULSEON 的配置被忽略。

17.7.4 LCD 状态寄存器 (LCD_STS)

偏移地址: 0x08

复位值: 0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										FCRSF	RDY	UDD	UDR	SOF	ENSTS
										r	r	r	rs	r	r

位域	名称	描述
31:6	Reserved	保留, 必须保持复位值。
5	FCRSF	LCD Frame Control Register Synchronization flag (LCD 帧控制寄存器同步标志) 每次 LCD_FCTRL 寄存器在 LCDCLK 域中更新时, 该位由硬件置 1。当对 LCD_FCTRL 寄存器写操作时, 该位由硬件清零。 0: 未同步; 1: 已同步。
4	RDY	Ready Flag (电压转换器准备状态) 该位由硬件置位和清零。 0: 电压转换器未准备就绪; 1: 电压转换器已使能并能提供正确的电压。

位域	名称	描述
3	UDD	Update Display Done（更新显示完成） 该位由硬件置 1。如果 LCD_FCTRL.UDDIE 位置 1，将产生一个 UDD 中断。 对 LCD_CLR.UDDCLR 位写 1 时，该位清零。置位优先级高于清零。 0：无事件； 1：显示更新请求已完成。
2	UDR	Update Display Request（更新显示请求） 每次修改 LCD_RAM 后软件必须置位 UDR 位，通知硬件将更新后的数据传输到二级缓冲器。 0：无效； 1：更新显示请求。
1	SOF	Start of frame flag（帧起始标志） 该位在新帧开始时由硬件置位。如果 LCD_FCTRL.SOFIE 位置 1，将产生一个帧起始中断。对 LCD_CLR.SOFCLR 位写 1 时，该位清零。清零优先级高于置位。 0：无事件； 1：帧起始事件发生。
0	ENSTS	LCD 控制器状态。 该位由硬件置位和清零。 0：LCD 控制器未使能； 1：LCD 控制器使能。

17.7.5 LCD 清除寄存器（LCD_CLR）

偏移地址：0x0C

复位值：0x0000 0000

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

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3	UDDCLR	Update display done clear（更新显示完成标志清除） 0：不清除 LCD_STS.UDD 标志位； 1：清除 LCD_STS.UDD 标志位。 <i>注：此位置 1 后需要手动清 0，否则 LCD_STS.UDD 位会被一直清 0</i>
2	Reserved	保留，必须保持复位值。

位域	名称	描述
1	SOFCLR	Start of frame flag clear（帧起始标志清除） 0：无效； 1：清除 LCD_STS.SOF 标志位。
0	Reserved	保留，必须保持复位值。

17.7.6 LCD 显示内存寄存器（LCD_RAM1_COMx x = 0...7）

偏移地址：0x14 + x*8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16
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
S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:0	Sy	显示内存 LCD_RAM1_COMx（x = 0...7）像素位（y = 0...31），每一位对应一个像素，像素值为 1 代表活动，为 0 表示不活动。

17.7.7 LCD 显示内存寄存器（LCD_RAM2_COMx x = 0...7）

偏移地址：0x18 + x*8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											S36	S35	S34	S33	S32
											rw	rw	rw	rw	rw

位域	名称	描述
31:5	Reserved	保留，必须保持复位值。
4:0	Sy	显示内存 LCD_RAM2_COMx（x = 0...7）像素位（y = 0...4），每一位对应一个像素，像素值为 1 代表活动，为 0 表示不活动。

18 LED 驱动器

18.1 LED 介绍

LED 是恒流驱动器，最大支持 10 个 COM 输出引脚和 16 个 SEG 输出引脚。每个 COM 端口最大支持 8 个 SEG 同时导通，每个 SEG 最大支持 20mA 的恒流调节，电流调节占空比范围 1/1~1/64。

18.2 LED 主要特性

- 支持 10 x COM、16 x SEG
- 每个 COM 端口最大支持 8 个 SEG 同时导通，最大 160mA 电流输出
- 支持 0.5ms、1ms、2ms 切换 COM 扫描
- COM 电路具有消影功能
- 每个 SEG 端口支持可配置 5~20mA 恒流灌电流输入
- 每个 SEG 通道可选择是否工作
- 每个像素点端口可以支持 64 阶 PWM 电流可调
- 工作电压范围 4.5V ~ 5.5V
- 支持 SPI 四线通信，最大数据传输速率 3MHz

18.3 LED 功能描述与操作说明

18.3.1 控制接口

支持 4 线 SPI 从机控制，分别是 SPI_CS，SPI_CLK，SPI_MOSI，SPI_MISO，空闲状态下默认为低电平；SPI_CS，空闲状态下默认为高电平；

1. MCU 作为主机，“主机单向只发送模式”下下发命令，LED 作为从机，“从机单向只接收模式”下接收命令；
2. MCU 作为主机，“主机单向只接收模式”下回读状态，LED 作为从机，“从机单向只发送模式”下发送状态；
3. LED SPI 接口：
 - 1) 时钟极性：空闲状态时，SCLK 保持低电平；
 - 2) 时钟相位：数据采样从第一个时钟边沿开始；
 - 3) 片选 CS：空闲状态时，CS 保持高电平；当 CS 被拉低时，LED 识别 SPI_CLK；当 CS 被拉高时，LED 忽略 SPI_CLK；
 - 4) 当 MCU 处于接收状态下，在第一个边沿采集，即上升沿采样数据；
 - 5) 16 位数据帧格式，帧格式为 MSB；

6) COM 切换时，LED SPI 接口重置采样接收移位。

注意：当使用 LED 功能时，芯片内部默认使用 SPI3 接口。

18.3.2 时钟控制

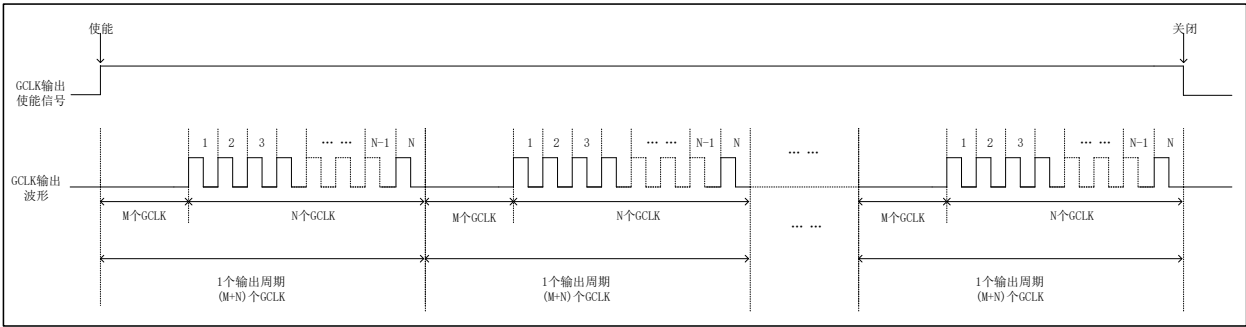
GCLK 是 LED 用于 SEG 占空比控制、COM 扫描切换的时钟。

GCLK 由 MCU 控制通过 PAD 输出，输出范围：50KHz~250KHz，典型 100KHz。

M（低电平时钟周期个数，8bit 配置）+ N（时钟周期 Clock 信号个数，8bit 配置，典型 130）；

扫描一个 COM 所需要的时间公式： $(M + N) / Ft = 1ms$ ；（ Ft 为 GCLK 输出频率，单位为 KHz）

图 18-1 GCLK 时序图



18.3.3 通信协议格式描述

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	一级命令			二级命令			DATA							

其中：

- 1. BIT15、BIT14 表示命令头，固定为 1b，0b ；
- 2. BIT13、BIT12 表示一级命令类型，共有四种一级命令类型，见下表；
- 3. BIT11~BIT8 表示二级命令类型，范围 0000b ~ 1111b，见下表；
- 4. BIT7~BIT0 表示下发命令的数据域；

18.3.3.1 一级命令类型

一级命令类型	命令域 BIT[13:12]	描述
基本命令	00b	其中包括以下命令集，“COM 强制关断”“COM 扫描复位”“COM 扫描使能”“读取 LED 寄存器”“芯片唤醒”
修调命令	01b	其中包括以下命令集，“SEG 修调寄存器写入”“RSET 修调寄存器写入”“BG 修调寄存器写入”

全局命令	10b	其中包括以下命令集,“设置 SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”“设置使用 COM 个数&COM0-1 的工作状态”“设置使用 COM2-9 的工作状态”“设置使用 SEG0~7 的工作状态”“设置使用 SEG8~15 的工作状态”
设置 SEG 占空比命令	11b	其中包括以下命令集,“设置 SEG0 占空比”“设置 SEG1 占空比” “设置 SEG15 占空比”

18.3.3.2 二级命令类型

二级命令类型	命令域 BIT[15:8]	命令数据域 BIT[7:0]说明
COM 强制关断	0x80	<p>数据位 BIT[7:0]=0x00（保留填 0）；</p> <p>功能说明：</p> <p>1，强制 COM 扫描显示停止且状态机恢复到初始化状态，COM-IO & SEG-IO 保持为三态，内部的模拟电源处于关闭状态；</p> <p>2，寄存器变化的有：</p> <p>“当前扫描 COM 寄存器”：BIT[3:0]变成 0x0F，BIT[7:4]：保留值为 0；</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”：BIT[7]变为 0，处于 Disable 状态；</p> <p>3，配置仍然保持不变：</p> <p>“修调相关寄存器”：8 个不变；</p> <p>“使用 COM 个数&COM0-1 的工作状态寄存器”：不变；</p> <p>“COM2-9 的工作状态寄存器”：不变；</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”：BIT[6: 0]不变；</p> <p>“SEG0~7 工作状态寄存器”：不变</p> <p>“SEG8~15 工作状态寄存器”：不变</p> <p>“SEG0~15 占空比”：不变</p> <p>4，当 COM 强制关断后，读取“当前扫描 COM 寄存器”时，返回值为 0x0F；</p>
COM 扫描复位	0x81	<p>数据位 BIT[7:0]=0x00（保留填 0）；</p> <p>功能说明：</p> <p>1，COM 扫描的状态机恢复到初始化状态；即恢复到 COM0 开始，并使 COM 扫描暂停，当继续发送 GCLK 可直接开启显示；</p> <p>2，寄存器变化的有：</p> <p>“当前扫描 COM 寄存器”：BIT[3:0]变成 0x0F，BIT[7:4]：保留值为 0；</p> <p>3，以下寄存器仍然保持 COM 扫描复位前的值：</p>

		<p>“修调相关寄存器”：8个不变；</p> <p>“使用 COM 个数&COM0-1 的工作状态寄存器”：不变；</p> <p>“COM2-9 的工作状态寄存器”：不变；</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”：不变；</p> <p>“SEG0~7 工作状态寄存器”：不变</p> <p>“SEG8~15 工作状态寄存器”：不变</p> <p>“SEG0~15 占空比”：不变；</p> <p>3，内部的模拟电源处于开启状态</p> <p>4，COM-IO &SEG-IO 保持为正常工作状态；</p> <p>5，COM 扫描复位完成后，读取“当前扫描 COM 寄存器”时，返回值为 0x0F，表示当前处于默认状态；</p>																	
COM 扫描使能	0x82	<p>数据位 BIT[7:0]=0x00（保留填 0）；</p> <p>功能说明：</p> <p>1，COM-IO &SEG-IO 根据参数设置自动配置为“LED 工作状态”或者“三态”；</p> <p>2，配置“当前扫描 COM 寄存器”：BIT[3:0]配置为 0，表示从 COM0 扫描，BIT[7:4]：保留值为 0；</p> <p>3，COM 扫描显示开启，从 COM0 开启扫描，等待接收到 GCLK 后开始显示；</p> <p>4，该命令会主动触发一次 LED 从二级缓存数据搬运到一级缓存数据中；</p>																	
读取 LED 寄存器	0x83	<p>MCU 回读 LED 寄存器状态需要两个步骤：</p> <p>1，SPI 发送“读取 LED 寄存器指令”命令到 LED；</p> <p>数据位 BIT[7:0] 表示 MCU 下发给 LED 读取 “LED 寄存器指令集”索引，见下表：</p> <table border="1"> <thead> <tr> <th>bit[7:4]</th><th>bit[3:0]</th><th>LED 寄存器指令</th></tr> </thead> <tbody> <tr> <td rowspan="3">xx01b</td><td>0000b~ 0101b</td><td>回读 SEG 修调寄存器数据</td></tr> <tr> <td>0110b</td><td>回读 RSET 修调寄存器数据</td></tr> <tr> <td>0111b</td><td>回读 BG 修调寄存器数据</td></tr> <tr> <td rowspan="3">xx10b</td><td>0000b</td><td>SEG 工作电流&SEG 显示模式&芯片 Enable/Disable</td></tr> <tr> <td>0001b</td><td>使用 COM 个数&COM0-1 的工作状态寄存器</td></tr> <tr> <td>0010b</td><td>COM2-9 的工作状态寄存器</td></tr> </tbody> </table>	bit[7:4]	bit[3:0]	LED 寄存器指令	xx01b	0000b~ 0101b	回读 SEG 修调寄存器数据	0110b	回读 RSET 修调寄存器数据	0111b	回读 BG 修调寄存器数据	xx10b	0000b	SEG 工作电流&SEG 显示模式&芯片 Enable/Disable	0001b	使用 COM 个数&COM0-1 的工作状态寄存器	0010b	COM2-9 的工作状态寄存器
bit[7:4]	bit[3:0]	LED 寄存器指令																	
xx01b	0000b~ 0101b	回读 SEG 修调寄存器数据																	
	0110b	回读 RSET 修调寄存器数据																	
	0111b	回读 BG 修调寄存器数据																	
xx10b	0000b	SEG 工作电流&SEG 显示模式&芯片 Enable/Disable																	
	0001b	使用 COM 个数&COM0-1 的工作状态寄存器																	
	0010b	COM2-9 的工作状态寄存器																	

			0011b	SEG0~7 工作状态寄存器
			0100b	SEG8~15 工作状态寄存器
			0111b	当前扫描 COM 寄存器
		xx11b	0000b	SEG0 占空比
			0001b	SEG1 占空比
		
			1111b	SEG15 占空比
		2, MCU 通过 SPI 发送给 LED 全 0 的 16bit 数据, 从而读取对应 LED 寄存器状态位, 返回状态位具体含义如下:		
		LED 寄存器指令	BIT[15:8]: 有效指令固定为 “0x83”; BIT[7:0]: 含义如下所述	
		回读 BG 修调寄存器数据	回读出来的值, 与写入的值保持一致	
		回读 RSET 修调寄存器数据	回读出来的值, 与写入的值保持一致	
		回读 SEG 修调寄存器数据	回读出来的值, 与写入的值保持一致	
		SEG 工作电流 & SEG 显示模式 & 芯片 Enable/Disable	BIT[7]: 表示芯片是否处于使能状态, 当 BIT[7]为 0 时, 表示芯片 Disable, 当 BIT[7]为 1 时, 表示芯片 Enable; BIT[6]: 表示 SEG 显示方式: 当 BIT[6]为 0 时, 表示 SEG 以前 0.5ms 完成 SEG0~7 的方式显示, 后 0.5ms 完成 SEG8~15 的方式显示。当 BIT[6]为 1 时, 表示 SEG 以 1ms 完成 SEG0~15 的方式显示; BIT[6]默认值为 0; BIT[5]: 保留值为 1; BIT[4:0]: 表示设置 SEG 共享电流值大小。BIT[4:0] = N, 表示 (0.5*N+5) mA (N=0~30,N 为正整数, 步进 0.5mA)。当 BIT[4:0] = 0, 表示 5mA; BIT[4:0] = 1, 表示 5.5mA; BIT[4:0] = 30, 表示 20mA。	

		使用 COM 个数 & COM0-1 的工作状态寄存器	<p>BIT[7:6]: 表示 COM1~0 的使用个数;</p> <p>1, 数据 BIT[3:0]表示 COM0~9 的使用个数, 例如 BIT[3:0] = 3: 表示使用 COM0~2; BIT[3:0] = 0x0, 表示未使用 COM; BIT[3:0] = 0x1, 表示使用 COM0, 共 1 个 COM; BIT[3:0] = 0xA, 表示使用 COM0~COM9, 共 10 个 COM; BIT[3:0] = 0xB~0xF 为无效保留; 使用的 COM-IO 为“LED 工作状态”, 未使用的 COM-IO 为“三态”;</p> <p>2, BIT[7:6]分别表示 COM1-0 的导通情况, 其中 BIT[7]对应 COM1, BIT[6]对应 COM0; (当数据位为 1 时, 代表对应 COM 处于被选中的“LED 工作状态”; 数据位为 0 时, 代表对应 COM 处于未被选中的“三态”)</p>
		COM2-9 的工作状态寄存器	<p>BIT[7:0]分别表示 COM9-2 的导通情况, 其中 BIT[0]对应 COM2, BIT[1]对应 COM3, BIT[2]对应 COM4, BIT[3]对应 COM5, BIT[4]对应 COM6, BIT[5]对应 COM7, BIT[6]对应 COM8, BIT[7]对应 COM9</p> <p>(当数据位为 1 时, 代表对应 COM 处于被选中的“LED 工作状态”; 数据位为 0 时, 代表对应 COM 处于未被选中的“三态”)</p>
		SEG0~7 工作状态寄存器	<p>BIT[7:0]: BIT7~BIT0 分别对应 SEG7~SEG0, 当数据位为 1 时, 代表对应 SEG 处于被选中的“工作状态”; 数据位为 0 时, 代表对应 SEG 处于未被选中的“三态”; 例如 BIT[7:0] = 0x11, 表示 SEG0 和 SEG4 处于被选中的工作状态, 其它 SEG 处于未被选中的“三态”。</p>
		SEG8~15 工作状态寄存器	<p>BIT[7:0]: BIT7~BIT0 分别对应 SEG15~SEG8, 当数据位为 1 时, 代表对应 SEG 处于被选中的“工作状态”; 数据位为 0 时, 代表对应 SEG 处于未被选中的“三态”; 例如 BIT[7:0] = 0x44, 表示 SEG9 和 SEG14 处于被选中的工作状态, 其它 SEG 处于未被选中的“三态”。</p>
		当前扫描 COM 寄存器	<p>BIT[7:4]: 保留值为 0;</p> <p>BIT[3:0]: 值对应当前扫描的 COM0~COM9, BIT[3:0] = 0000b~1001b 有效, 0x0F 表示当前处于关断状态, 其余值无效;</p> <p>BIT[3:0] = 0x0F: 表示 COM 处于关断状态或芯片 Disable;</p> <p>BIT[3:0] = N 表示当前正在扫描 COM (N), N=0~9 正整数;</p> <p>注意: 回读该数据之前, 必须先使能 COM 扫描, 即对应的“0x8200”命令;</p>
		SEG0 占空比	BIT[7:6]: 保留值为 0;

			<p>BIT[5:0]: 表示 SEG0 的占空比参数; BIT[5:0] = N (其中 N=0~63, N 为正整数), 表示 SEG0 的占空比 N/63, 其中 63=2⁶-1; 例如 BIT[5:0] = 24, 表示 SEG0 的占空比为 24/63; 例如 BIT[5:0] = 0, 表示 SEG0 的占空比为 0/63;</p> <p>注意: 回读该数据之前, 必须先使能 COM 扫描, 即对应的"0x8200"命令;</p>
		SEG1 占空比	同"SEG0 占空比"
	
		SEG15 占空比	同"SEG0 占空比"
		无效回读指令	BIT[7:0]: 保留值为 0;
读取关闭	0x84	<p>读取关闭 (MISO 常低);</p> <p>数据位 BIT[7:0]保留</p>	
芯片唤醒	0x87	<p>数据位 BIT[7:0]保留</p> <p>芯片唤醒, 所有 IO 从三态恢复到配置状态;</p> <p>功能说明:</p> <p>1, COM-IO & SEG-IO 恢复为正常工作状态, 内部的模拟电源处于打开状态;</p> <p>2, 寄存器变化的有:</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”: BIT[7]变为 1, 处于 Enable 状态;</p> <p>3, 配置仍然保持不变:</p> <p>“修调相关寄存器”: 8 个不变;</p> <p>“使用 COM 个数&COM0-1 的工作状态寄存器”: 不变;</p> <p>“当前扫描 COM 寄存器”: 不变;</p> <p>“COM2-9 的工作状态寄存器”: 不变;</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”: BIT[6: 0]不变;</p> <p>“SEG0~7 工作状态寄存器”: 不变</p> <p>“SEG8~15 工作状态寄存器”: 不变</p> <p>“SEG0~15 占空比”: 不变</p>	
无效控制命令	0x85、0x86、 0x88~0x90	<p>BIT[7:0]: 任意值;</p> <p>对 LED 不产生影响;</p>	
修调命令 (1~N,	0x90~0x95	Bit[7:0]为有效位	

N = 8)		SEG 修调寄存器写入指令(6)
	0x96	Bit[4:0]为有效位, Bit[7:5]默认为 0 RSET 修调寄存器写入指令
	0x97	Bit[4:0]为有效位, Bit[7:5]默认为 0 BG 修调寄存器写入指令
无效控制命令	0x99~0x9F	BIT[7:0]: 任意值; 对 LED 不产生影响;
设置 SEG 工作电流 & SEG 显示模式 & 芯片 Enable/Disable	0xA0	<p>设置 16 个 SEG 共享电流大小、SEG 显示模式以及芯片是否使能。</p> <p>1, 其中 BIT[4:0]代表设置 SEG 共享电流。BIT[4:0] = N, 表示 (0.5*N + 5) mA (N = 0~30, N 为正整数, 步进 0.5mA)。BIT[4:0] = 0, 表示 5mA; BIT[4:0] = 1, 表示 5.5mA; BIT[4:0] = 30, 表示 20mA;</p> <p>2, BIT[6]表示 SEG 显示方式:</p> <p>当 BIT[6]为 0 时, 表示 SEG 以前 0.5ms 完成 SEG0~7 的方式显示, 后 0.5ms 完成 SEG8~15 的方式显示。当 BIT[6]为 1 时, 表示 SEG 以 1ms 完成 SEG0~15 的方式显示; BIT[6]默认值为 0;</p> <p>3, BIT[7]表示芯片是否使能状态:</p> <p>当 BIT[7]为 0 时, 表示芯片 Disable; 当 BIT[7]为 1 时, 表示芯片 Enable;</p> <p>数据位 BIT[5]默认保留为 1;</p> <p>4, 芯片 Enable 后会使得 LED 内部恒流源以及数字相关的逻辑电路, 从而开启 LED 显示;</p> <p>1, 芯片 Disable 后, 会关闭 LED 内部恒流源, 从而关闭 LED, 与设置“COM 强制关断”命令其它寄存器状态表现相同, 其它寄存器状态如下:</p> <p>寄存器变化的有:</p> <p>“当前扫描 COM 寄存器”: BIT[3:0]变成 0x0F, BIT[7:4]: 保留值为 0;</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”: BIT[7]变为 0, 处于 Disable 状态;</p> <p>配置仍然保持不变:</p> <p>“修调相关寄存器”: 8 个不变;</p> <p>“使用 COM 个数&COM0-1 的工作状态寄存器”: 不变;</p> <p>“COM2-9 的工作状态寄存器”: 不变;</p> <p>“SEG 工作电流&SEG 显示模式&芯片 Enable/Disable”: BIT[6:0]不变;</p>

		<p>“SEG0~7 工作状态寄存器”：不变</p> <p>“SEG8~15 工作状态寄存器”：不变</p> <p>“SEG0~15 占空比”：不变</p>
设置使用 COM 个数 & COM0-1 的工作状态	0xA1	<p>1, 数据 BIT[3:0] 表示 COM0~9 的使用个数, 例如 BIT[3:0] = 3: 表示使用 COM0~2; BIT[3:0] = 0x0, 表示未使用 COM; BIT[3:0] = 0x1, 表示使用 COM0, 共 1 个 COM; BIT[3:0] = 0xA, 表示使用 COM0~COM9, 共 10 个 COM; BIT[3:0] = 0xB~0xF 为无效保留; 使用的 COM-IO 为“LED 工作状态”, 未使用的 COM-IO 为“三态”;</p> <p>2, BIT[7:6] 分别表示 COM1-0 的导通情况, 其中 BIT[7] 对应 COM1, BIT[6] 对应 COM0;</p> <p>3, COM 有两个状态, 分别是“LED 工作状态”或“三态”; 当数据位为 1 时, 代表对应 COM 处于“LED 工作状态”; 数据位为 0 时, 代表对应 COM 处于“三态”; 例如 BIT[7] = 1, 代表 COM1 为“LED 工作状态”。</p>
设置 COM2-9 的工作状态	0xA2	<p>1, 数据表示 COM2~9 的工作状态, COM 有两个状态, 分别是“LED 工作状态”或“三态”;</p> <p>2, BIT[7:0] 分别表示 COM9-2 的导通情况, 其中 BIT[0] 对应 COM2, BIT[1] 对应 COM3, BIT[2] 对应 COM4, BIT[3] 对应 COM5, BIT[4] 对应 COM6, BIT[5] 对应 COM7, BIT[6] 对应 COM8, BIT[7] 对应 COM9;</p> <p>3, 当数据位为 1 时, 代表对应 COM 处于“LED 工作状态”; 数据位为 0 时, 代表对应 COM 处于“三态”; 例如 BIT[5] = 1, 代表 COM7 为“LED 工作状态”。</p>
设置 SEG0~7 工作状态	0xA3	<p>BIT[7:0] 表示 SEG[7:0]-IO 状态处于“LED 工作状态”或“三态”, 0 表示三态, 1 表示 LED 工作状态, BIT0~BIT7 分别对应 SEG0~SEG7; 例如 BIT[7:0] = 0x17, 代表 SEG0、SEG1、SEG2、SEG4 为“LED 工作状态”, 其它为“三态”。</p>
设置 SEG8~15 工作状态	0xA4	<p>BIT[7:0] 表示 SEG[8:15]-IO 的处于“LED 工作状态”或“三态”, 0 表示“三态”, 1 表示“LED 工作状态”, BIT0~BIT7 分别对应 SEG8~SEG15;</p> <p>例如 BIT[7:0] = 0x33, 代表 SEG8、SEG9、SEG12、SEG13 为“LED 工作状态”, 其它为“三态”。</p>
无效命令	0xA5~0xAF	<p>BIT[7:0]: 任意值;</p> <p>对 LED 不产生影响;</p>
设置 SEG0 占空比	0xB0	<p>数据表示 SEG0 的占空比参数, BIT0~BIT5 有效, BIT6~BIT7 保留 (写无效, 读返回 2'b00); BIT[5:0] = N (N 为正整数, 范围 0~63), 表示 SEG0 的占空比 N/63, 其中 $63=2^6-1$; 例如 BIT[5:0] = 24, 表示 SEG0 的占空比为 24/63; 例如 BIT[5:0] = 0, 表示 SEG0 的占空比为 0/63;</p>
...

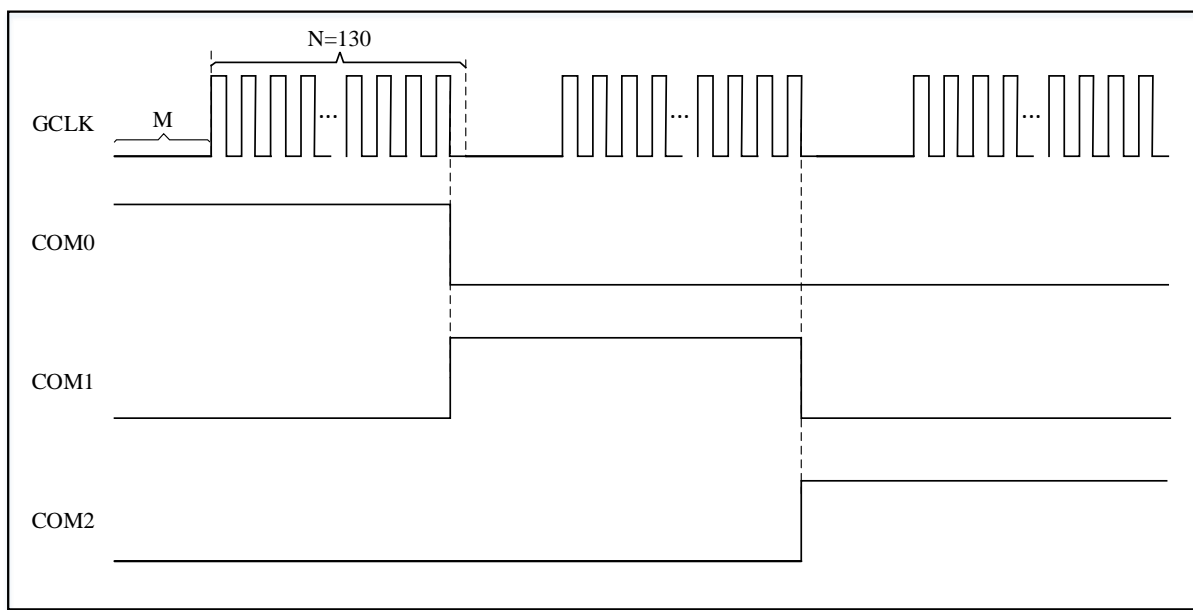
设置 SEG15 占空比	0xBF	数据表示 SEG15 的占空比参数，BIT0~BIT5 有效，BIT6~BIT7 保留；BIT[5:0] = N（N 为正整数，范围 0~63），表示 SEG0 的占空比 N/63，其中 63=2 ⁶ -1；
无效控制命令	0xC0~0xFF	BIT[7:0]：任意值； 对 LED 不产生影响；

18.3.4 LED 工作时序

18.3.4.1 GCLK 与 COM-SEG 之间的关系

1. GCLK 输出每完成输出 130 个脉冲，LED 会从当前 COM 导通的状态切换到下一个 COM 导通，最后一个 COM 的下一个 COM 是 COM0，如下图所示：

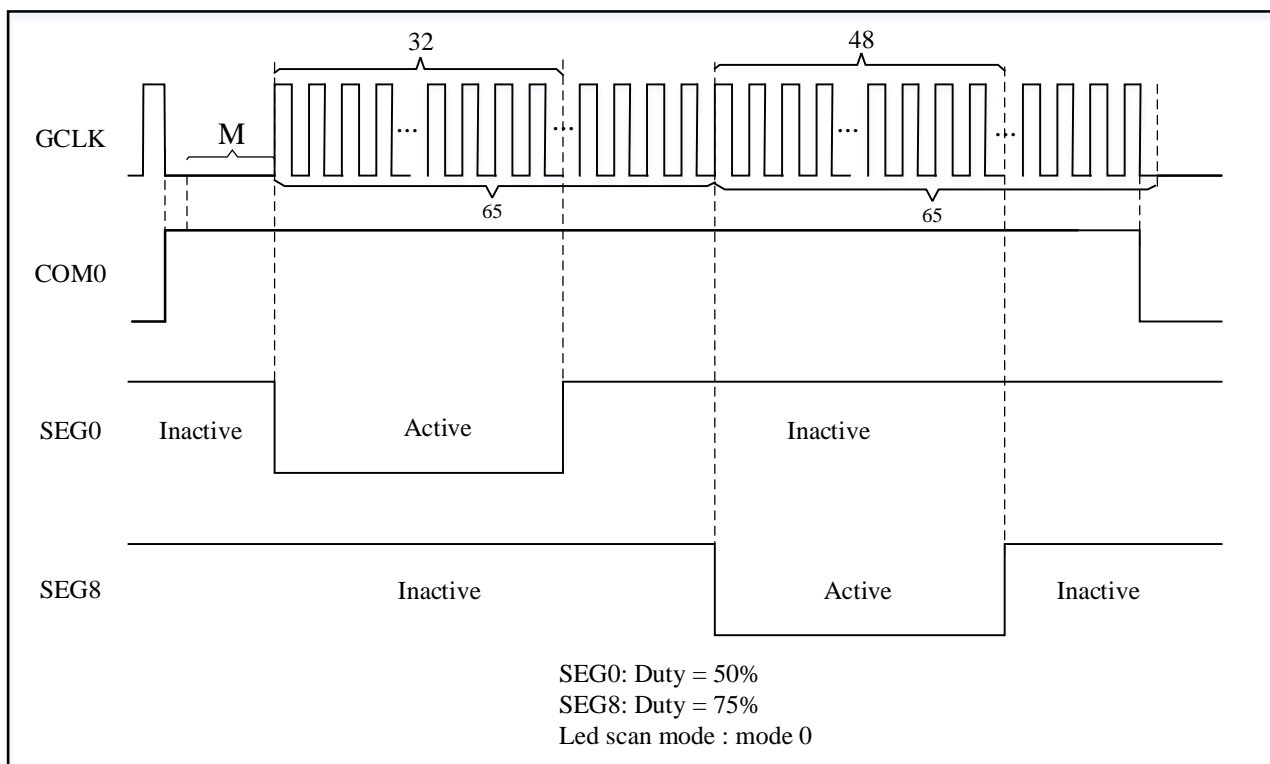
图 18-2 GCLK 与 COM 对应时序图



2. SEG 关闭(Inactive 状态) 的电压在 3.4V 左右，SEG 导通 (Active 状态) 的电压在 1V 左右。当 GCLK 脉冲个数为第 1 个 - 第 65 个时，SEG0-SEG7 会同时导通，SEG8-SEG15 此时处于关闭状态；GCLK 脉冲个数为第 66 个 - 第 130 个时，SEG0-SEG7 会处于关闭状态, SEG8-SEG15 此时处于导通状态；

SEG 占空比控制主要依靠 GCLK 当前的输出个数来判断某个 SEG 是否应当关闭，SEG 占空比控制如下图所示：

图 18-3 GCLK 与 COM-SEG 控制时序图



18.3.5 控制 LED 流程

18.3.5.1 上电初始化

1. SPI3 初始化：主机单向只发送模式，数据位宽 16bit，时钟极性：空闲状态时，SCLK 保持低电平，时钟相位：数据采样从第一个时钟边沿开始，使用硬件控制 NSS，帧格式数据为 MSB，通信速率小于 3MHz；
2. DMA 初始化：源地址-SRAM（COM2 对应的所有 SEG(0-15)占空比参数（bit[7:6]为保留位，bit[5:0]共 6bit 有效）数据的首地址），目标地址-SPIDAT 寄存器，方向：内存到外设，数据位宽 16bit，内存地址递增，外设地址不变，传输长度-最大 16，触发源-RCC 的 GCLK 输出完成信号，使能 DMA 传输完成中断；
3. GCLK 初始化：输出频率-130KHz，设置 M 个低电平时钟周期个数，设置 N 个时钟周期 Clock 信号个数；
4. LED 初始化：通过 SPI 接口下发初始化配置 LED，在按照顺序发送完 COM 强制关断，芯片唤醒命令之后、再发送全局命令（包含配置 COM 个数，SEG 工作电流&SEG 显示模式&芯片 Enable、SEG 的工作状态）、之后再发送修调相关的指令即可；

18.3.5.2 LED 显示工作流程

1. MCU 内部定义一个“当前扫描 COM 全局变量”（简称 Var），该变量初始值为 COM0；
2. 通过 SPI 发送 COM0 对应的 SEG0~15 占空比数据到 LED；
3. 通过 SPI 发送“使能 COM 扫描命令”；

4. 通过 SPI 发送 COM1 对应的 SEG0~15 占空比数据到 LED，并使能 GCLK 输出；
5. 等到 GCLK 输出完成时（MCU 产生 GCLK 输出完成标志），该信号会触发 LED 会从 COM0 切换到 COM1。同时 MCU 会产生 GCLK 输出完成信号来触发 DMA，DMA 开始将 SRAM 中 COM2 对应的 SEG0~15 占空比数据依次搬运到 SPI_DAT 寄存器，SPI 自动将该数据发送给 LED；

在 DMA 传输完成中断处理函数中依次进行下述软件处理：

- MCU 将 DMA 对应通道的源地址改为下一个 COM 对应的 SEG0~15 占空比数据的首地址，传输个数依然设置为 16；
 - 通过 Var 变量判断当前所有使能的 COM 是否扫描完成。如果扫描完成则配置 Var 设为 COM0；否则 Var 设为下一个 COM。
 - MCU 最后软件清除 DMA 完成中断标志后退出 DMA 完成中断处理函数；
6. 重复 5 的操作 LED 将当前具体 COM 切换到下一个 COM，MCU 将 DMA 对应通道的源地址依次改为下个 COM 对应的 SEG0~15 占空比数据的首地址（最后一个 COM 对应的下个 COM 为 COM0）；

18.3.5.3 带按键扫描的 LED 显示工作流程

1. MCU 内部定义一个“当前扫描 COM 全局变量”（简称 Var），该变量初始值为 COM0；
2. 通过 SPI 发送 COM0 对应的 SEG0~15 占空比数据到 LED；
3. 通过 SPI 发送“使能 COM 扫描命令”；
4. 通过 SPI 发送 COM1 对应的 SEG0~15 占空比数据到 LED，并使能 GCLK 输出；
5. 等到 GCLK 输出完成时（MCU 产生 GCLK 输出完成标志），该信号会触发 LED 会从 COM0 切换到 COM1。同时 MCU 会产生 GCLK 输出完成信号来触发 DMA，DMA 开始将 SRAM 中 COM2 对应的 SEG0~15 占空比数据依次搬运到 SPI_DAT 寄存器，SPI 自动将该数据发送给 LED；

在 DMA 传输完成中断处理函数中依次进行下述软件处理：

- 1) MCU 将 DMA 对应通道的源地址改为下一个 COM 对应的 SEG0~15 占空比数据的首地址，传输个数依然设置为 16；
- 2) 通过 Var 变量判断当前所有使能的 COM 是否扫描完成。如果扫描完成则配置 Var 设为 COM0；否则 Var 累加 1。
- 3) MCU 最后软件清除 DMA 完成中断标志后退出 DMA 完成中断处理函数；
6. 重复 5 的操作 LED 将当前具体 COM 切换到下一个 COM，MCU 将 DMA 对应通道的源地址依次改为下个 COM 对应的 SEG0~15 占空比数据的首地址（最后一个 COM 对应的下个 COM 为 COM0）。直到完成最后一个 COM 的显示（此时是完成 DMA 将 SRAM 中 COM1 对应的 SEG0~15 占空比数据依次搬运到 SPI_DAT 寄存器），则执行步骤 7；
7. 在最后一个 COM 扫描完成时，在 DMA 传输完成中断处理函数里依次进行下述软件处理：
 - 1) MCU 先关闭 GCLK 输出，再通过 SPI 下发“LED 强制关断命令”；
 - 2) MCU 初始化配置需要扫描的 GPIO（通用输入模式，下拉输入），完成按键扫描（读取扫描 GPIO 的电平状态），完成扫描后再将需要扫描的 GPIO 配置为高阻状态；
 - 3) MCU 通过软件控制 SPI 发送 COM0 对应的 SEG0~15 占空比数据到 LED；
 - 4) MCU 通过 SPI 先发送“芯片唤醒命令”，然后再发送“使能 COM 扫描命令”；
 - 5) MCU 通过 SPI 发送 COM1 对应的 SEG0~15 占空比数据到 LED；
 - 6) MCU 使能 GCLK 输出并将 Var 设为 COM0；
 - 7) MCU 最后软件清除 DMA 完成中断标志后退出 DMA 完成中断处理函数；
8. 重复步骤 6 的操作即可

18.3.6 回读 LED 寄存器流程描述

1. 初始化配置 SPI 为主机单向只发送模式，数据位宽 16bit，时钟极性：空闲状态时，SCLK 保持低电平，时钟相位：数据采样从第一个时钟边沿开始，使用硬件控制 NSS，帧格式数据为 MSB，通信速率小于 3MHz；
2. 通过 SPI 下发读取某个 LED 寄存器指令（指令的 DATA 内容代表某个 LED 的寄存器）；此时 LED 接收到 SPI 读取某个 LED 寄存器指令后等待 MCU 完成步骤 3、4 再将 LED 寄存器的内容通过 SPI 发送给 MCU；

3. 重新配置 SPI 为主机单向只接收模式，其它配置不变；
4. MCU 通过 SPI 执行只接收操作：使能 SPI，SPI 主动发送 SPI_CLK 给 LED，LED 接收到 SPI_CLK 后将 LED 寄存器的数据通过 SPI_MISO 串行发出，MCU 通过 SPI_MISO 引脚上采集接收到的数据并保存在 SPI_DAT 数据寄存器里，MCU 可以通过 SPI 接收标志判断数据是否接收成功，并将该数据读取后用于对比预期配置值；
5. 重复步骤 1-4，用于读取其它 LED 寄存器；

如果需要结束读取 LED 寄存器，先执行步骤 1，MCU 再通过 SPI 下发读取关闭指令给 LED 即可；

18.3.7 复用按键扫描

由于 LED 显示通常对刷新率要求不高，通常 1ms 扫描一个 COM。考虑到 LED 通常占用 I/O 资源较大，为了 I/O 的复用，用户可以考虑在全部 COM 扫描完成后，在 SEG 端口做按键扫描，完成 I/O 扫描之后，再继续 LED 显示的工作。

19 内部集成电路总线(I²C)

19.1 简介

I²C(inter-integrated circuit)总线是一种广泛应用的总线结构，它只有两根双向线，即数据总线 SDA 和时钟总线 SCL，通过这两根线，所有与 I²C 总线兼容的设备都可以通过 I²C 总线彼此直接通信。

I²C 接口连接微控制器和串行 I²C 总线，可用于 MCU 和外部 I²C 设备的通讯。I²C 接口模块实现了 I²C 协议的标速模式和快速模式，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)和 PMBus (电源管理总线)，此外它提供多主机功能，控制所有 I²C 总线特定的时序、协议、仲裁。I²C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

19.2 主要特性

- 同一接口既可实现主机功能又可实现从机功能
- 是并行总线到 I²C 总线协议的转换器
- 支持 7 位和 10 位的地址模式和广播寻址
- 作为 I²C 主设备可以产生时钟、起始信号和停止信号
- 作为 I²C 从设备具有可编程的 I²C 地址检测、停止位检测的功能
- 支持标速(最高 100kHz)、快速模式(最高 400kHz)、快速+模式(最高 1MHz)
- 支持中断向量，事件中断和错误中断共用一个中断向量
- 可选的时钟延展功能
- 支持 DMA 模式
- 可选择的 PEC（报文错误检测）生成和校验
- 兼容 SMBus 2.0 和 PMBus
- 可编程的模拟滤波和数字滤波

注：不是所有产品中都包含上述所有特性，请参考相关的数据手册，确认该产品支持的 I²C 功能。

19.3 功能描述

I²C 接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I²C 总线与外部设备进行通信，可以连接到标准（高达 100kHz）或快速（400kHz、1MHz）的 I²C 总线。I²C 模块接收时将数据从串行转换成并行，发送时将数据从并行转换成串行。支持中断模式，用户可以根据需要开启或禁止中断。

19.3.1 SDA/SCL 控制

I²C 模块有两条接口线：串行数据线（SDA）和串行时钟线（SCL）。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须带开漏或者开集，以提供线与功能。I²C 总线上的数据在标准模式下可以达到 100 kbit/s，在快速模式下可以达到 1000kbit/s。由于 I²C 总线上可能会连接不同工艺的设

备，逻辑‘0’和逻辑‘1’的电平并不是固定的，取决于 VDD 的实际电平。

如果允许时钟延长，即 SCL 线拉低，就可以避免在接收时发生过载错误以及在发送时发生欠载错误。

比如，在发送模式时，如果发送数据寄存器为空且字节发送结束位置起（I2C_STS1.TXDATE = 1, I2C_STS1.BSF = 1），I²C 接口在传输前保持时钟线为低，以等待软件读取 STS1 后把数据写进数据寄存器（缓冲器和移位寄存器都是空的）；在接收模式时，如果数据寄存器非空且字节发送结束位置起（I2C_STS1.RXDATNE = 1, I2C_STS1.BSF = 1），I²C 接口在接收到数据字节后保持时钟线为低，以等待软件读 STS1，然后读数据寄存器 DAT（缓冲器和移位寄存器都是满的）。

如果从模式中禁止时钟延长，在接收模式时，如果接收数据寄存器非空（I2C_STS1.RXDATNE = 1），在接收到下个字节前数据还没有被读出，则发生过载错误，最后一字节也将被丢弃。在发送模式时，如果发送数据寄存器空（I2C_STS1.TXDATE = 1），在必须发送下个字节之前还没有新数据写进数据寄存器，则发生欠载错误。相同的字节将被重复发出。这种情况下不控制重复写冲突。

19.3.2 软件通讯流程

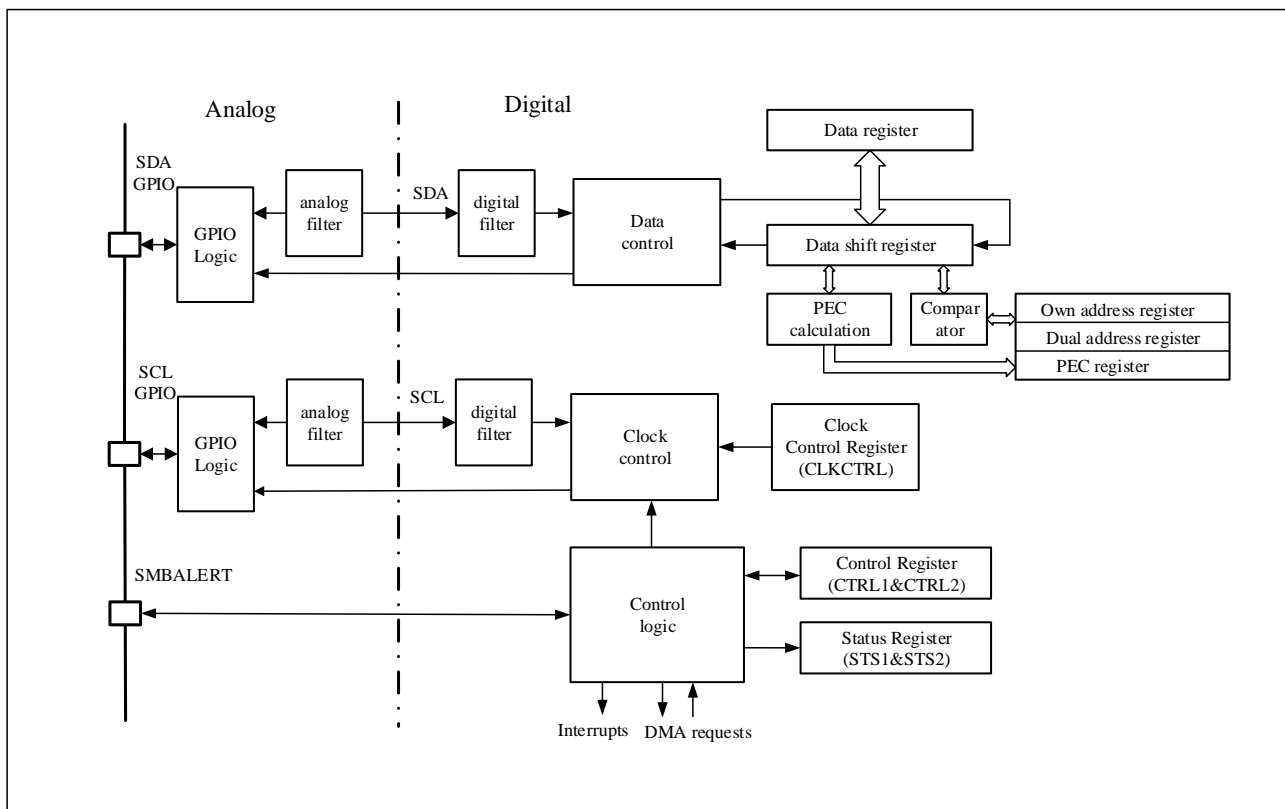
I²C 设备的数据传输分为主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。不管 I²C 设备是主机还是从机，都可以发送或接收数据。I²C 接口支持 4 种运行模式：

- 从机发送器模式
- 从机接收器模式
- 主机发送器模式
- 主机接收器模式

系统复位后，I²C 默认工作在从机模式下。通过软件配置 I²C 接口在总线上发送一个起始位，随后接口自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。

I²C 接口的功能框图如下：

图 19-1 I²C 功能框图

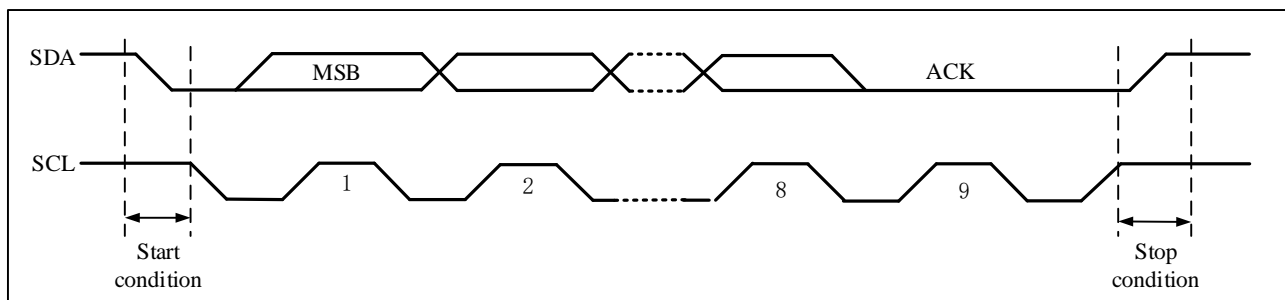


注：在 SMBus 模式下，SMBALERT 是可选信号。如果禁止了 SMBus，则不能使用该信号。

19.3.2.1 开始和停止条件

所有的数据传输总是以起始位开始并以停止位结束。起始条件和停止条件都是在主模式下由软件控制产生。起始位（START）是指：在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。停止位（STOP）是指在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。如下图所示：

图 19-2 I²C 总线协议



19.3.2.2 时钟同步与仲裁

I²C 接口支持多主机仲裁，即两个主机可以同时在空闲总线上开始传送数据，因此就必须通过一些机制来决定哪个主机获取总线的控制权，这一般是通过时钟同步和仲裁来完成的。

I²C 电路具有两个关键特点：

- SDA 和 SCL 为漏极开路结构，通过外部的上拉电阻实现了信号的“线与”逻辑；

- SDA 和 SCL 引脚在输出信号的同时还将引脚上的电平进行检测，检测是否与刚才输出一致。这为“时钟同步”和“总线仲裁”提供硬件基础。

I²C 设备对总线的操作是通过“把线路接地”来输出逻辑 0。基于 I²C 总线的特点，如果一设备发送逻辑 0，其他发送逻辑 1，那么线路看到的只有逻辑 0，所以线路上不可能出现电平冲突的现象。

总线的物理接法允许主设备往总线写数据的同时读取数据。这样两主设备争总线的时候发送逻辑 0 的并不知道竞争的发生，只有发送逻辑 1 的才会发现冲突（当写一个逻辑 1，却读到了 0）从而退出竞争。

时钟同步

多个主机可以同时空闲总线上产生时钟。SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平。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。

当所有有关的器件数完了它们的低电平周期后，时钟线被释放并且变成高电平，之后器件时钟和 SCL 线的状态没有差别，而且所有器件会开始数它们的高电平周期，首先完成高电平周期的器件会再次将 SCL 线拉低。

这样，产生的同步 SCL 时钟的低电平周期由低电平时钟周期最长的器件决定，而高电平周期由高电平时钟周期最短的器件决定。

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。当两个主机在总线空闲的时候都产生了一个有效的起始位，这种情况就需要决定由哪个主机来完成数据传输，这就是仲裁的过程。

各个主控制器没有对总线实施控制的优先级别，都是由仲裁决定的。总线控制随即而定且逐位进行，他们遵循“低电平优先”的原则，即谁先发送低电平谁就会掌握对总线的控制权。在每一位的仲裁期间，当 SCL 为高时，每个主机都检查自己的 SDA 电平是否和自己发送的相同。理论上讲，如果两个主机所传输的内容完全相同，那么他们能够成功传输而不出现错误。如果一个主机发送高电平但检测到 SDA 电平为低，则认为自己仲裁失败并关闭自己的 SDA 输出驱动，而另一个主机则继续完成自己的传输。

19.3.2.3 I²C 数据通信流程

每个 I²C 设备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 主机负责产生起始位和结束位来开始和结束一次传输，并且负责产生 SCL 时钟。

I²C 模块支持 7 位和 10 位的地址，用户可以通过软件配置 I²C 从机的地址。I²C 从机检测到 I²C 总线上的起始位之后，就开始从总线上接收地址，并把接收到的地址和自身的地址进行比较，一旦两个地址相同，I²C 从机将发送一个确认应答(ACK)，并响应总线的后续命令：发送或接受所要求的数据。此外，如果软件开启了广播呼叫，则 I²C 从机始终对一个广播地址(0x00)发送确认应答。

数据和地址按 8 位字节进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址（7 位模式为 1 个字节，10 位模式为 2 个字节）。地址只在主模式发送。在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位（ACK）给发送器。如图 19-2 I²C 总线协议所示。

软件可以开启或禁能应答（ACK），并可以设置 I²C 接口的地址（7 位、10 位地址或广播呼叫地址）。

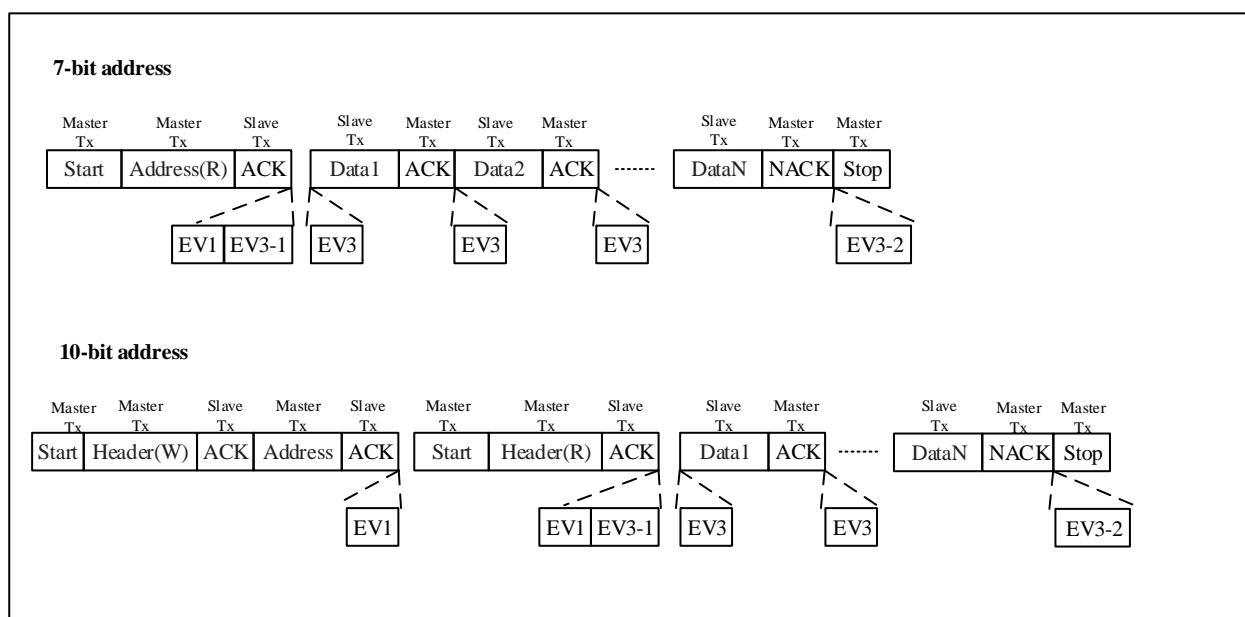
19.3.2.4 I²C 从机发送模式

在从机模式下发送接收标记位（I2C_STS2.TRF）指示当前是处于接收器模式还是发送器模式。当在发送器

模式下要发送数据到 I²C 总线，软件应该按照下面的步骤操作：

1. 首先，使能 I²C 外设时钟，配置 I2C_CTRL1 中相关寄存器，确保输出正确的 I²C 时序。当这两步都完成以后，I²C 运行在从机模式下，等待接收起始位和地址。
2. I²C 从机先收到一个起始位，随后收到匹配的 7 位或 10 地址，I²C 硬件将 I2C_STS1.ADDRF 位（接收到了地址并且和自身的地址匹配）置 1，软件应该定期查询此位或者中断监视此位，发现置位后，软件读 I2C_STS1 寄存器然后读 I2C_STS2 寄存器来清除 I2C_STS1.ADDRF 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 I2C_STS1.ADDRF 位置 1。软件继续通过读 I2C_STS1 寄存器和接着读 I2C_STS2 寄存器来第二次清除 I2C_STS1.ADDRF 位。
3. I²C 进入数据发送状态，现在移位寄存器和数据寄存器 I2C_DAT 都是空，所以硬件将 I2C_STS1.TXDATE（发送数据空）位置 1。此时软件可以写入第一个字节数据到 I2C_DAT 寄存器，但是，因为写入 I2C_DAT 寄存器的字节被立即移入内部移位寄存器了，所以 I2C_STS1.TXDATE 位并没有被清 0。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，软件写第二个字节到 I2C_DAT，此时 I2C_DAT 寄存器和移位寄存器都不是空。I2C_STS1.TXDATE 位被清 0。
5. 第一个字节发送完成之后，I2C_STS1.TXDATE 再次被置起，软件写第三个字节到 I2C_DAT，同时 I2C_STS1.TXDATE 位被清 0。在此之后，只要依然有数据待等待被发送且 I2C_STS1.TXDATE 被置 1，软件都可以写入一个字节到 I2C_DAT 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_DAT 寄存器来清除 I2C_STS1.TXDATE 标志位，之后就再也不用关心 I2C_STS1.TXDATE 的状态。I2C_STS1.TXDATE 位会在倒数第二个字节发送完成后置起，直到检测到 STOP 结束位时被清 0。
7. 根据 I²C 协议，I²C 主机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 ACKFAIL 位（应答出错）会置起以通知软件发送结束。软件写 0 到 I2C_STS1.ACKFAIL 位可以清除此位。

图 19-3 从发送器传送序列



说明:

1. EV1: I2C_STS1.ADDRF = 1, 读 STS1 然后读 STS2 将清除该事件。
2. EV3-1: I2C_STS1.TXDATE=1, 移位寄存器空,数据寄存器空, 写 DAT。
3. EV3: I2C_STS1.TXDATE=1, 移位寄存器非空, 数据寄存器空, 写 DAT 将清除该事件。
4. EV3-2: I2C_STS1.ACKFAIL=1, 在 STS1 的 ACKFAIL 位写“0”可清除该事件。

注:

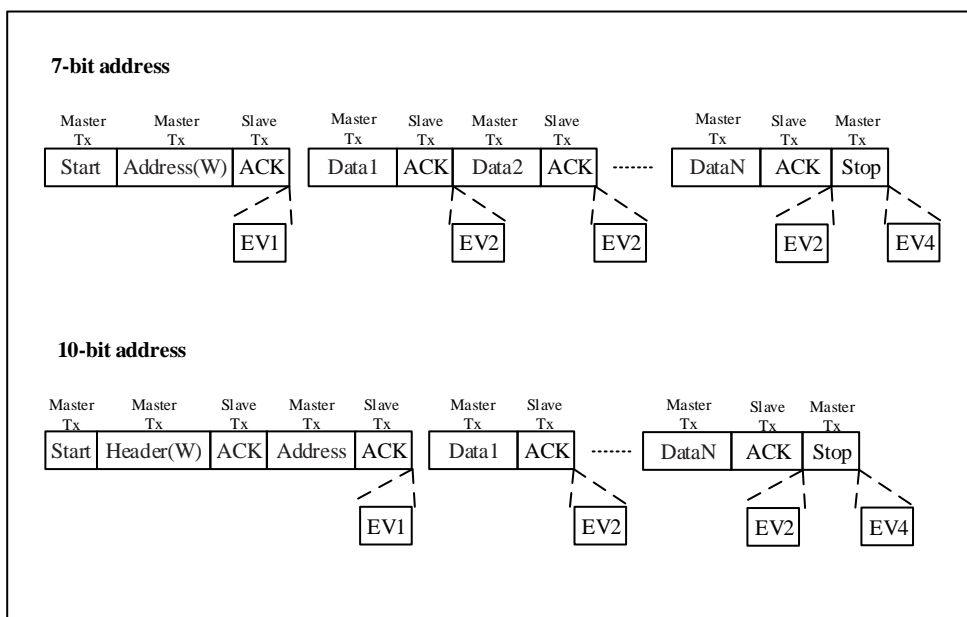
- a) EV1 和 EV3_1 事件拉长 SCL 低的时间, 直到对应的软件序列结束。
- b) EV3 的软件序列必须在当前字节传输结束之前完成。

19.3.2.5 I²C 从机接收模式

在从机模式下接收数据时, 软件应该按如下步骤操作:

1. 首先, 使能 I²C 外设时钟, 配置 I2C_CTRL1 中相关寄存器, 确保输出正确的 I²C 时序。当这两步都完成以后, I²C 运行在从机模式下, 等待接收起始位和地址。
2. 在接收到 START 起始条件和匹配的 7 位或 10 地址之后, I²C 硬件将 I2C_STS1.ADDRF 位 (接收到了地址并且和自身的地址匹配) 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_STS1 寄存器然后再读 I2C_STS2 寄存器来清除 I2C_STS1.ADDRF 位。一旦 I2C_STS1.ADDRF 位被清 0, I²C 从机就开始接收来自 I²C 总线的数据。
3. 当接收到第一个字节后, I2C_STS1.RXDATNE 位 (接收数据非空) 被硬件置 1, 如果设置了 I2C_CTRL2.EVTINTEN 和 I2C_CTRL2.BUFINTEN 位, 则产生一个中断。软件应该通过轮询或者中断来检测该位, 一旦发现置起后, 软件可以读取 I2C_DAT 寄存器的第一个字节, 此时 I2C_STS1.RXDATNE 位被清 0。注意, 如果设置了 I2C_CTRL1.ACKEN 位, 则在接收到一个字节后, 从机应该产生一个应答脉冲。
4. 任何时候, 只要 I2C_STS1.RXDATNE 位被置 1, 软件均可以从 I2C_DAT 寄存器读取一个字节。当接收到最后一个字节后, I2C_STS1.RXDATNE 被置 1, 软件读取最后一个字节
5. 当从机检测到 I²C 总线上的停止位 (STOP) 后, 将 I2C_STS1.STOPF 位置 1, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则产生一个中断。软件通过先读 I2C_STS1 寄存器再写 I2C_CTRL1 寄存器来清除 I2C_STS1.STOPF 位 ((见下图的 EV4))。

图 19-4 从接收器传送序列



说明:

1. EV1: I2C_STS1.ADDRF = 1, 读 STS1 然后读 STS2 将清除该事件。
2. EV2: I2C_STS1.RXDATNE = 1, 读 DAT 将清除该事件。
3. EV4: I2C_STS1.STOPF = 1, 读 STS1 然后写 CTRL1 寄存器将清除该事件。

注:

- a) EV1 事件拉长 SCL 低的时间, 直到对应的软件序列结束。
- b) EV2 的软件序列必须在当前字节传输结束之前完成。

19.3.2.6 I²C 主机发送模式

在主模式时, I²C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。当通过 START 位在总线上产生了起始条件, 设备就进入了主模式。

在主机模式下发送数据到 I²C 总线时, 软件应该按如下步骤操作:

1. 首先, 使能 I²C 外设时钟, 配置 I2C_CTRL1 中相关寄存器, 确保输出正确的 I²C 时序。当这两步都完成以后, I²C 默认运行在从机模式下, 等待接收起始位和地址。
2. 当 BUSY=0 时, 设置 I2C_CTRL1.STARTGEN 位为 1, I²C 接口将产生一个开始条件并切换至主模式 (I2C_STS2.MSMODE 位置位)。
3. 一旦发出开始条件, I²C 硬件将 I2C_STS1.STARTBF 位 (起始位标志) 置 1 然后进入主机模式, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则会产生一个中断。接着软件读 I2C_STS1 寄存器然后写一个 7 位地址位或带有地址头的 10 位地址位到 I2C_DAT 寄存器来清除 I2C_STS1.STARTBF 位。I2C_STS1.STARTBF 位被清 0 后 I²C 就开始发送地址或者地址头到 I²C 总线。

在 10 位地址模式时, 发送一个头序列会产生以下事件:

- ◆ I2C_STS1.ADDR10F 位被硬件置位, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则产生一个中断。然后主设备读 STS1 寄存器, 再将第二个地址字节写入 DAT 寄存器。

- ◆ I2C_STS1.ADDRF 位被硬件置位，如果设置了 I2C_CTRL2.EVTINTEN 位，则产生一个中断。随后主设备读 STS1 寄存器，跟着读 STS2 寄存器

注：在发送器模式，主设备先发送头字节（11110xx0），然后发送从地址的低 8 位。（这里 xx 代表 10 位地址中的最高 2 位）。

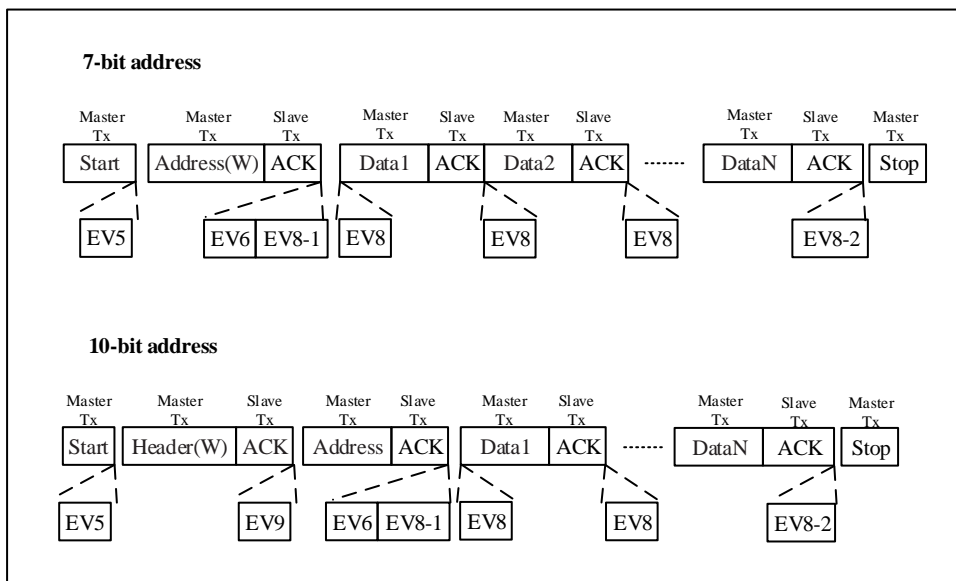
在 7 位地址模式时，只需送出一个地址字节，一旦该地址字节被送出：

- ◆ ADDRf 位被硬件置位，如果设置了 I2C_CTRL2.EVTINTEN 位，则产生一个中断。随后主设备等待一次读 STS1 寄存器，跟着读 STS2 寄存器。

注：在发送器模式，主设备发送从地址时置最低位为‘0’。

4. 7 位或 10 位的地址位发送完成后，I²C 硬件将 I2C_STS1.ADDRF 位（地址已被发送）置 1，如果设置了 I2C_CTRL2.EVTINTEN 位，则产生一个中断，软件通过读 I2C_STS1 寄存器然后读 I2C_STS2 寄存器来清除 I2C_STS1.ADDRF
5. I²C 进入数据发送状态，因为移位寄存器和数据寄存器（I2C_DAT）都是空的，所以硬件将 I2C_STS1.TXDATE 位（发送数据空）置 1，接着软件写第一个字节数据到 I2C_DAT 寄存器，但是，因为写入 I2C_DAT 寄存器的字节被立即移入内部移位寄存器，所以 I2C_STS1.TXDATE 位此时不会被清零。一旦移位寄存器非空，I²C 就开始发送数据到总线。
6. 在第一个字节的发送过程中，软件写第二个字节到 I2C_DAT，此时 I2C_STS1.TXDATE 被清零。任何时候，只要还有数据等待被发送，且 I2C_STS1.TXDATE 位被置 1，软件都可以向 I2C_DAT 寄存器写入一个字节。
7. 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_DAT 来清除 I2C_STS1.TXDATE 标志位，此后就不用关心 I2C_STS1.TXDATE 位的状态。I2C_STS1.TXDATE 位会在倒数第二个字节发送完成后被置起，直到发送停止位（STOP）时被清零。
8. 最后一个字节发送结束后，因为移位寄存器和 I2C_DAT 寄存器此时都为空，I²C 主机将 I2C_STS1.BSF 位（字节发送结束）置位，在清除 I2C_STS1.BSF 位之前 I²C 接口将保持 SCL 为低电平；读出 I2C_STS1 之后再写入 I2C_DAT 寄存器将清除 I2C_STS1.BSF 位。软件此时设置 I2C_CTRL1.STOPGEN 位产生一个停止条件，然后 I²C 接口将自动回到从模式（I2C_STS2.MSMODE 位清除）。

图 19-5 主发送器传送序列



说明:

1. EV5: I2C_STS1.STARTBF = 1, 读 STS1 然后将地址写如 DAT 寄存器将清除该事件。
2. EV6: I2C_STS1.ADDRF = 1, 读 STS1 然后读 STS2 将清除该事件。
3. EV8_1: I2C_STS1.TXDATE = 1, 移位寄存器空, 数据寄存器空, 写 DAT 寄存器。
4. EV8: I2C_STS1.TXDATE = 1, 移位寄存器非空, 数据寄存器空, 写 DAT 寄存器将清除该事件。
5. EV8_2: I2C_STS1.TXDATE = 1, I2C_STS1.BSF = 1, 请求设置停止位。这两个事件由硬件在产生停止条件时清除。

6. EV9: I2C_STS1.ADDR10F = 1, 读 STS1 然后写入 DAT 寄存器将清除该事件。

注:

- a) EV5、EV6、EV9、EV8_1 和 EV8_2 事件拉长 SCL 低的时间, 直到对应的软件序列结束。
- b) EV8 的软件序列必须在当前字节传输结束之前完成。
- c) 当 TXDATE 或 BSF 位置位时, 停止条件应安排在出现 EV8_2 事件时。

19.3.2.7 I²C 主机接收模式

支持 BYTENUM 字节控制模式, 在主机接收模式下, 配置好接收字节数之后, 由硬件自动结束通讯, 无需软件介入配置 NACK 及发送 START/STOP 条件, 当然也可以使用普通的主机接收模式。

在主机模式下从 I²C 总线接收数据软件应该按如下步骤操作:

1. 首先, 使能 I²C 外设时钟, 配置 I2C_CTRL1 中相关寄存器, 确保输出正确的 I²C 时序。使能和配置以后, I²C 默认运行在从机模式下, 等待接收起始位和地址。

注: 若需启用主机接收字节控制, 则需要在此步骤使能 I2C 外设时钟后额外使能 I2C_BYTENUM.BYTENUMEN, 并通过 I2C_BYTENUM.BYTENUM 配置接收字节数, I2C_BYTENUM.RXFSEL 选择接收完成后主机发送 START 或 STOP 条件。

2. 当 BUSY=0 时, 设置 I2C_CTRL1.STARTGEN 位为 1, I²C 接口将产生一个开始条件并切换至主模式 (I2C_STS2.MSMODE 位置位)。
3. 一旦发出开始条件, I²C 硬件将 I2C_STS1.STARTBF 位 (起始位标志) 置 1 然后进入主机模式, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则会产生一个中断。接着软件读 I2C_STS1 寄存器然后写一个 7 位地址位或带有地址头的 10 位地址位到 I2C_DAT 寄存器来清除 I2C_STS1.STARTBF 位。I2C_STS1.STARTBF 位被清 0 后 I²C 就开始发送地址或者地址头到 I²C 总线。

在 10 位地址模式时, 发送一个头序列会产生以下事件:

- I2C_STS1.ADDR10F 位被硬件置位, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则产生一个中断。然后主设备读 STS1 寄存器, 再将第二个地址字节写入 DAT 寄存器。
- I2C_STS1.ADDRF 位被硬件置位, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则产生一个中断。随后主设备读 STS1 寄存器, 跟着读 STS2 寄存器。

注: 在接收器模式, 主设备先发送头字节 (11110xx0), 然后发送从地址的低 8 位, 然后再重新发送一个开始条件, 后面跟着头字节 (11110xx1) (这里 xx 代表 10 位地址中的最高 2 位)。

在 7 位地址模式时, 只需送出一个地址字节, 一旦该地址字节被送出:

- I2C_STS1.ADDRF 位被硬件置位, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则产生一个中断。随后主设备等待一次读 STS1 寄存器, 跟着读 STS2 寄存器。

注: 在接收器模式, 主设备发送从地址时置最低位为 '1'。

4. 7 位或 10 位的地址位发送完成后, I²C 硬件将 I2C_STS1.ADDRF 位 (地址已被发送) 置 1, 如果设置了 I2C_CTRL2.EVTINTEN 位, 则产生一个中断, 软件通过读 I2C_STS1 寄存器然后读 I2C_STS2 寄存器来清除 I2C_STS1.ADDRF, 在发送地址和清除 I2C_STS1.ADDRF 之后, 如果地址是 10 位格式, 软件应该再次将 STARTGEN 位置 1 来重新产生一个 START(Sr)。在 START 产生后, I2C_STS1.STARTBF 位会被置 1。软件应该通过先读 I2C_STS1 然后写地址头到 I2C_DAT 来清除 I2C_STS1.STARTBF 位, 然后地址头被发到 I²C 总线, ADDR10F 再次被置 1。软件应该再次通过先读 I2C_STS1 然后读 I2C_STS2 来清除 I2C_STS1.ADDRF 位。
5. 在发送完地址和清除 I2C_STS1.ADDRF 之后, I²C 接口进入主机接收器模式。在此模式下, I²C 接口从 SDA 线接收数据字节, 并通过内部移位寄存器送至 DAT 寄存器。一旦接收到第一个字节, 硬件会将 I2C_STS1.RXDATNE 位 (接收数据非空标志位) 置 1, 如果 ACKEN 位被置位, 发出一个应答脉冲。此时软件可以从 I2C_DAT 寄存器读取第一个字节, 之后 I2C_STS1.RXDATNE 位被清 0。此后, 只要 I2C_STS1.RXDATNE 被置 1, 软件就可以从 I2C_DAT 寄存器读取一个字节。
6. 主设备在从设备接收到最后一个字节后发送一个 NACK。接收到 NACK 后, 从设备释放对 SCL 和 SDA 线的控制; 主设备就可以发送一个停止/重新起始条件。为了在收到最后一个字节后产生一个 NACK 脉冲, 接收完倒数第二个字节(N-1)数据之后, 软件应该立即清除 ACKEN 位。为了产生一个停止/重新起始条件, 软件必须在读倒数第二个数据字节之后将 I2C_CTRL1.STOPGEN 位或者 I2C_CTRL1.STARTGEN 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。

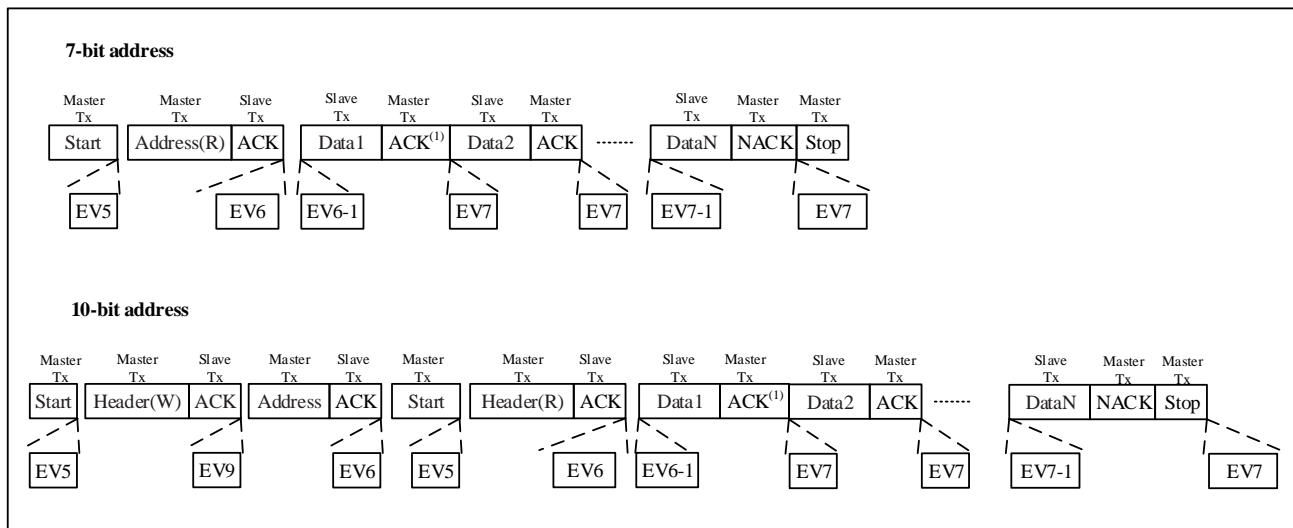
注: 若前面已启用主机接收字节控制, 则可以忽略步骤 6、7, 软件只需要根据接收标志读取字节即可, 硬件会自动在接收完成后自动发送 NACK 及 START/STOP 条件。

7. 最后一个字节接收完毕后, I2C_STS1.RXDATNE 位被置 1, 软件可以读取最后一个字节。由于 I2C_CTRL1.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 停止位。

注意：以上步骤要求字节数目 $N > 1$ ，如果 $N = 1$ ，步骤 6 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 19-6 主接收器传送序列



说明:

- EV5: I2C_STS1.STARTBF=1, 读 STS1 然后将地址写入 DAT 寄存器清除该事件。
- EV6: I2C_STS1.ADDRF=1, 读 STS1 然后读 STS2 将清除该事件。在 10 位主接收模式下，该事件后应设置 CTRL1 的 STARTGEN=1。
- EV6_1: 没有对应的事件标志，至适于接收 1 个字节的情况。恰好在 EV6 之后（即清除了 ADDRf 之后），要清除响应和停止条件的产生位。
- EV7: I2C_STS1.RXDATNE=1, 读 DAT 寄存器消除该事件。
- EV7_1: I2C_STS1.RXDATNE =1, 读 DAT 寄存器清除该事件。设置 I2C_CTRL1.ACKEN=0 和 I2C_CTRL1.STOPGEN=1。
- EV9: I2C_STS1.ADDR10F=1, 读 STS1 然后写入 DAT 寄存器将清除该事件。

注:

- 如果收到一个单独的字节，则是NA。
- EV5、EV6和EV9事件拉长SCL低电平，直到对应的软件序列结束。
- EV7的软件序列必须在当前字节传输结束前完成。
- EV6_1或EV7_1的软件序列必须在当前传输字节的ACK脉冲之前完成。

19.3.3 错误条件

I²C 的错误主要有总线错误、应答错误、仲裁丢失、过载/欠载错误。这些错误都可能造成通讯的失败。

19.3.3.1 应答错误 (ACKFAIL)

当接口检测到应答位与期望不符时，将产生应答错误。此时 I2C_STS1.ACKFAIL 被置位，如果设置了

I2C_CTRL2.ERRINTEN 位，则产生一个中断。当发送器接收到一个 NACK 时，必须复位通讯：如果处于从模式，硬件会释放总线。如果是处于主模式，软件必须生产一个停止条件。

19.3.3.2 总线错误 (BUSERR)

在一个地址或数据字节传输期间，当 I²C 接口检测到一个外部的停止或起始条件则产生总线错误，I2C_STS1.BUSERR 被置位。如果设置了 I2C_CTRL2.ERRINTEN 位为“1”，则产生一个中断。

在主模式情况下，硬件不释放总线，同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。

在从模式情况下，数据被丢弃，硬件释放总线。此时有两种情况：如果检测到错误的开始条件，从设备认为是一个重启动，并等待地址或停止条件。如果检测到错误的停止条件，从设备按正常的停止条件操作，同时硬件释放总线。

19.3.3.3 仲裁丢失 (ARLOST)

当 I²C 接口检测到仲裁丢失时产生仲裁丢失错误，硬件释放总线，I2C_STS1.ARLOST 位被置位。如果设置了 I2C_CTRL2.ERRINTEN 位为“1”，则产生一个中断。

I²C 接口自动回到从模式（I2C_STS2.MSMODE 位被清除）。当 I²C 接口丢失了仲裁，则它无法在同一次传输中作为从机响应它的从地址，但它可以在赢得了仲裁的主设备重新发送起始条件之后响应。

19.3.3.4 过载/欠载错误 (OVERRUN)

在从机接收模式下，如果禁止时钟延长，容易发生过载/欠载错误。

当它已经接收到一个字节（I2C_STS1.RXDATNE=1），但在 DAT 寄存器中前一个字节数据还没有被读出，则发生过载错误，数据寄存器最后接收的数据被丢弃；同时软件应清除 I2C_STS1.RXDATNE 位，发送器重新发送最后一次发送的字节。

在从机发送模式下，如果禁止时钟延长，在当前字节已经发送完成，而 DAT 仍然为空（I2C_STS1.TXDATE=1），则发生欠载错误。此时，在 DAT 寄存器中的前一个字节将被重复发出；用户应该确定在发生欠载错时，接收端应丢弃重复接收到的数据。发送端应按 I²C 总线标准在规定的时间内更新 I2C_DAT 寄存器。

在发送第一个字节时，必须在清除 I2C_STS1.ADDRF 之后并且第一个 SCL 上升沿之前写入 I2C_DAT 寄存器；如果不能做到这点，则接收方应该丢弃第一个数据。

19.3.4 DMA 应用

在传输时，当数据寄存器为空或满时，能够生成 DMA 请求。DMA 能够写数据到 I²C 数据寄存器，或从 I²C 数据寄存器读出数据以减少 CPU 的开销。

DMA 请求必须在当前字节传输结束之前被响应。当相应 DMA 通道设置的数据传输已经完成时，DMA 控制器发送传输结束信号 EOT 到 I²C 接口，并且在中断使能时产生一个中断。

在主机发送模式，在 EOT 中断服务程序中，需禁止 DMA 请求，然后在等到 I2C_STS1.BSF 事件后设置停止条件。

在主机接收模式，当要接收的数据数目大于或等于 2 时，DMA 控制器发送一个硬件信号 EOT_1，它对应 DMA 传输（字节数-1）。如果设置了 I2C_CTRL2.DMALAST 位，硬件在发送完 EOT_1 后的下一个字节，将自动发送 NACK。在中断允许的情况下，用户可以在 DMA 传输完成的中断服务程序中产生一个停止条件。

19.3.4.1 发送流程

DMA 模式通过设置 I2C_CTRL2.DMAEN 位使能。只要 I2C_STS1.TXDATE 位被置位，数据将由 DMA 从预置的存储区装载进 I2C_DAT 寄存器。设置 DMA 通道进行 I²C 发送，须执行以下步骤（x 是通道号）：

1. 在 DMA_PADDRx 寄存器中设置 I2C_DAT 寄存器地址。数据将在每个 I2C_STS1.TXDATE 事件后从存储区传送至这个地址。
2. 在 DMA_MADDRx 寄存器中设置存储器地址。数据在每个 I2C_STS1.TXDATE 事件后从这个存储区传送至 I2C_DAT。
3. 在 DMA_TXNUMx 寄存器中设置所需传输的字节数。在每个 I2C_STS1.TXDATE 事件后，此值递减，直到 0。
4. 利用 DMA_CHCFGx 寄存器中的 PRIOLVL[1:0]位配置通道优先级。
5. 设置 DMA_CHCFGx 寄存器中的 DIR 位，并根据应用要求可以配置在整个传输完成一半或全部完成时发出中断请求。
6. 通过设置 DMA_CHCFGx 寄存器上的 CHEN 位激活通道。
7. 当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I²C 接口发送一个传输结束的 EOT/EOT_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

注：使用 DMA 进行发送时，不要设置 I2C_CTRL2.BUFINTEN 位。

19.3.4.2 接收流程

DMA 模式通过设置 I2C_CTRL2.DMAEN 位使能。每次接收到数据字节时，将由 DMA 把 I2C_DAT 寄存器的数据传送到设置的存储区。设置 DMA 通道进行 I2C 接收，须执行以下步骤（x 是通道号）

1. 在 DMA_PADDRx 寄存器中设置 I2C_DAT 寄存器的地址。数据将在每次 I2C_STS1.RXDATNE 事件后从此地址传送到存储区。
2. 在 DMA_MADDRx 寄存器中设置存储区地址。数据将在每次 I2C_STS1.RXDATNE 事件后从 I2C_DAT 寄存器传送到此存储区。
3. 在 DMA_TXNUMx 寄存器中设置所需的传输字节数。在每个 I2C_STS1.RXDATNE 事件后，此值递减，直到 0。
4. 用 DMA_CHCFGx 寄存器中的 PRIOLVL[1:0]配置通道优先级。
5. 清除 DMA_CHCFGx 寄存器中的 DIR 位，根据应用要求可以设置在数据传输完成一半或全部完成时发出中断请求。
6. 设置 DMA_CHCFGx 寄存器中的 CHEN 位激活该通道。
7. 当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/EOT_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

注：使用 DMA 进行接收时，不要设置 I2C_CTRL2.BUFINTEN 位。

19.3.5 包错误校验 (PEC)

将 I2C_CTRL1.PECEN 位置 1 就可以使能 PEC 功能，PEC 使用 CRC-8 算法对包括地址和读/写位在内所有信息字节进行计算，从而提高通信的可靠性。包错误校验（PEC）计算器使用的 CRC-8 多项式为 $C(x) = x^8$

$+x^2+x+1$ 。

在发送模式时，软件可以在最后一个 I2C_STS1.TXDATE 事件时设置 I2C_CTRL1.PEC 传输位，PEC 将在最后一个字节后被发送。在接收模式时，软件在最后一个 I2C_STS1.RXDATNE 事件之后设置 I2C_CTRL1.PEC 位，然后接收 PEC 字节，并将接收到的 PEC 字节与内部计算的 PEC 值进行比较。如果不等于内部计算的 PEC，接收器发送一个 NACK。如果是主机接收器模式，不管校对的结果如何，PEC 后都将发送 NACK。需要注意，I2C_CTRL1.PEC 位必须在接收当前字节的 ACK 脉冲之前设置。

如果 DMA 和 PEC 计算器都被激活，I²C 将自动发送或者检查 PEC 值。

在发送模式时，当 I²C 接口从 DMA 控制器处接收到 EOT 信号时，它在最后一个字节后自动发送 PEC。在接收模式时，当 I²C 接口从 DMA 处接收到一个 EOT_1 信号时，它将自动把下一个字节作为 PEC，并且和内部计算的 PEC 进行比较。在接收到 PEC 后产生一个 DMA 请求。

为了允许中间 PEC 传输，I2C_CTRL2.DMALAST 位用于判别是否真是最后一个 DMA 传输。如果确实是最后一个主接收器的 DMA 请求，在接收到最后一个字节后自动发送 NACK。

当仲裁丢失的时候，PEC 计算失效。

19.3.6 超时错误

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。这就是为什么 SMBus 有最小传输速率的要求——为了防止超时后长时间锁死总线。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 会话。

N32G05X 系列芯片无论是 I2C 模式还是 SMBUS 模式均支持超时时间配置，包含 SCL 低超时判断、SCL 高超时判断、SDA 低超时判断；其中 SCL 低超时功能通过 I2C_CTRL1.LTOSEL 配置超时阈值，I2C_CTRL2.LTOEN 使能超时判断，SCL 为低的时间超过设置阈值时 I2C_STS1.SCLLTO 置 1，此时若 I2C_CTRL2.SCLLTOINTEN 为 1 则产生中断；SCL 高超时功能通过 I2C_CTRL1.HTOSEL 配置超时阈值，I2C_CTRL2.HTOEN 使能超时判断，SCL 为高的时间超过设置阈值时 I2C_STS1.SCLHTO 置 1，此时若 I2C_CTRL2.SCLHTOINTEN 为 1 则产生中断；SDA 低超时功能通过 I2C_CTRL1.LTOSEL 配置超时阈值，I2C_CTRL2.LTOEN 使能超时判断，SDA 为低的时间超过设置阈值时 I2C_STS1.SDALTO 置 1，此时若 I2C_CTRL2.SDALTOINTEN 为 1 则产生中断；如果某个会话耗时超过阈值，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状态（问题）。

19.3.7 SMBus

19.3.7.1 介绍

系统管理总线（System Management Bus，简称为 SMBus 或 SMB）是一种结构简单的单端双线制总线。通过它，各设备之间以及设备与系统的其他部分之间可以互相通信。SMBus 是 I2C 的一种衍生总线形式，为系统和电源管理相关的任务提供一条控制总线。SMBus 基于 I2C 通信标准，是一个与系统管理和电源管理相关的控制总线。想要了解更多信息，请参考 SMBus 规范 V2.0(<http://smbus.org/specs/>)。

SMBus 有三类设备标准：

- 主设备：发送命令、产生时钟和终止发送设备；
- 从设备：接收或响应命令设备；

- 主机：一个系统仅有一个主机，它提供与系统 CPU 的主接口。主机具有主-从机功能并必须支持 SMBus 提醒协议。

SMBus 与 I2C 的相似点：

- 总线协议都是两条线（一个时钟线 SCL 和一个数据线 SDA），再加一条可选的 SMBus 提醒线；
- 数据格式相似，SMBus 数据格式类似于 I2C 的 7 位地址格式（见图 19-2）；
- 都是主-从通信模式，且主设备提供时钟；
- 都支持多主机功能；

SMBus 和 I2C 的不同点：

表 19-1 SMBus 与 I²C 的比较

SMBus	I ² C
最大传输速度 100KHz	最大传输速度 1MHz
最小传输速度 10KHz	无最小传输速度
固定逻辑电平	逻辑电平由 VDD 决定
不同的地址类型(保留的, 动态的等)	7 位、10 位和广播呼叫从地址类型
不同的总线协议(快速命令, 处理呼叫等)	无总线协议

19.3.7.2 SMBus 用途

SMBus 利用系统管理总线，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，为系统和电源管理相关的任务提供控制总线。

19.3.7.3 设备标识

在 SMBus 中，任意一个设备作为从设备时都有一个地址，叫从地址。

为了给每一个设备分配地址，必须有一个唯一的设备标识（UDID）分配给设备。

19.3.7.4 总线协议

SMBus 技术规范包括 8 个总线协议。想要了解关于 SMBus 的详细信息和地址类型请参考 SMBus 技术规范 V2.0(<http://smbus.org/specs/>)。用户可以决定使用哪些协议。

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。

注：SMBus 不支持 Quick command 协议。

19.3.7.5 地址解析协议 (ARP)

通过 SMBus 协议动态分配新的唯一地址给每个从设备来解决地址冲突，这是地址解析协议（ARP）。地址解析协议具有一下特性：

任何一个 SMBus 主设备都可以遍历总线；

使用 SMBus 物理层仲裁机制分配地址。当设备维持供电期间，分配的地址保持不变，协议也允许再断电后保留其地址。

地址分配之后，没有额外的 SMBus 打包开销（访问分配地址的设备与访问固定地址的设备所用的时间是一样的）。

19.3.7.6 SMBus 提醒模式

SMBus 提供了一个可选的中断信号 SMBALERT（与 SCL 和 SDA 一样，是一种有线与信号），设备使用该信号来扩展其控制能力，但需要牺牲一个引脚。SMBALERT 通常和 SMBus 广播呼叫地址结合使用。关于 SMBus 的消息有 2 个字节。

仅具有从机功能的设备可以设置 I2C_CTRL1.SMBALERT 位以指示它要与主机通信。主机处理该中断并通过提醒响应地址 ARA（Alert Response Address，地址值为 0001100x）访问所有 SMBALERT 设备。只有那些将 SMBALERT 拉低的设备能应答 ARA。此状态是由 I2C_STS1.SMBALERT 来标识的。从发送设备提供的 7 位设备地址被放在字节的 7 个最高位上，第八个位可以是‘0’或‘1’。

当多个设备的 SMBALERT 为低时，在地址传输过程中，最高优先级（地址越小优先级越高）可以通过标准仲裁赢得总线通信。如果确认从机地址，则设备的 SMBALERT 不再为低。如果报文传输完毕，设备的 SMBALERT 仍为低，表示主机将再次读取 ARA。没有采用 SMBALERT 信号时主机可以定期访问 ARA。

19.3.7.7 SMBus 通信流程

SMBus 的通讯流程和标准 I²C 的流程相似。如果要使用 SMBus 模式，还需要在程序中配置 SMBus 特定寄存器、并响应 SMBus 特定标志位、实现在 SMBus 手册中介绍的上层协议。

1. 首先，设置 I2C_CTRL1.SMBMODE 位；并按照应用要求配置 I2C_CTRL1.SMBTYPE 和 I2C_CTRL1.ARPEN 位。如果 I2C_CTRL1.ARPEN=1 且 I2C_CTRL1.SMBTYPE=0，使用 SMB 设备默认地址；如果 I2C_CTRL1.ARPEN=1 且 I2C_CTRL1.SMBTYPE=1，使用 SMB 主设备头字段。
2. 为了支持 ARP 协议（I2C_CTRL1.ARPEN=1），在 SMBus 主机模式下（I2C_CTRL1.SMBTYPE=1），软件需要响应标志位 I2C_STS2.SMBHADDR（在 SMBus 从机模式下，响应 I2C_STS2.SMBDADDR 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 I2C_STS1.SMBALERT 标志位，并实现相应的功能。

19.3.8 噪声滤波

I²C 接口标准要求能够过滤在 SCL/SDA 线上的 50ns 的毛刺，因此设计了模拟滤波器和数字滤波器。模拟滤波器默认开启，也可以通过设置 I2C_GFLTRCTRL.SCLAFENN/SDAAFENN 位禁用此功能。模拟滤波器通过配置 I2C_GFLTRCTRL.SCLAFE/SDAAFW 设置过滤毛刺的宽度 5ns,15ns,25ns,35ns。数字滤波器通过配置 I2C_GFLTRCTRL.SCLDFW/SDADFW 为非 0 值去使能。其最大滤波宽度为（SCLDEW[3:0]或 SDADFW[3:0]）*T_{PCLK}。使能数字滤波器将增加 SDA 线的保持时间，增加值为(SDADFW[3:0]+1)*T_{PCLK}。

19.4 调试模式

当微控制器进入调试模式（Cortex®-M0 核心处于停止状态）时，根据 PWR 模块中的 DBG_CTRL.I2C1TIMOUT 和 DBG_CTRL.I2C2TIMOUT 配置位，SMBUS 超时控制或者继续正常工作或者可以停止。详见 3.3.2 节。

19.5 中断请求

下表列出了所有的 I²C 中断请求：

表 19-2 I²C 中断请求

中断函数	中断事件	事件标志	设置控制位
I2C 全局中断	起始位已发送（主）	STARTBF	EVTINTEN

中断函数	中断事件	事件标志	设置控制位
	地址已发送 (主) 或地址匹配 (从)	ADDRF	
	10 位头段地址已发送 (主)	ADDR10F	
	收到停止位 (从)	STOPF	
	数据字节传输完成	BSF	
	接收缓冲区非空	RXDATNE	EVTINTEN 和 BUFINTEN
	发送缓冲区为空	TXDATE	
	总线错误	BUSERR	ERRINTEN
	仲裁丢失 (主)	ARLOST	
	应答失败	ACKFAIL	
	过载/欠载	OVERRUN	
	PEC 错误	PECERR	
	SMBus 提醒	SMBALERT	
	SCL 低超时	SCLLTO	SCLLTOINTEN
	SCL 高超时	SCLHTO	SCLHTOINTEN
	SDA 低超时	SDALTO	SDALTOINTEN

19.6 I2C 寄存器描述

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

19.6.1 I²C 寄存器总览

表 19-3 I²C 寄存器总览

Offset	Register	32	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
000h	I2C_CTRL1	Reserved	LTOSEL		HTOSEL		Reserved										Reserved		SWRESET		SMBALERT	PEC	ACKPOS	ACKEN	STOPGEN	STARTGEN	NOEXTEND	GCEN	PECEN	ARPEN	SMBTYPE	SMBMODE	EN	0								
	Reset Value		0	0	0	0													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
004h	I2C_CTRL2	Reserved	LTOEN	HTOEN	SCLLTOINTEN	SCLHTOINTEN	SDALTOINTEN	Reserved										DMAEN	ERRINTEN	EVTINTEN	BUFINTEN	Reserved			DMALAST	Reserved	CLKFREQ[6:0]															
	Reset Value		0	0	0	0	0											0	0	0	0				0			0	0	0	0	0	0	0	0	0	0					
008h	I2C_OADDR1	Reserved																ADDRMODE	Reserved							ADDR [9:8]		ADDR[7:1]										ADDR0				
	Reset Value																	0								0	0											0				
00Ch	I2C_OADDR2	Reserved																Reserved										ADDR2[7:1]										DUALEN				
	Reset Value																																					0				
010h	I2C_DAT	Reserved																Reserved										DATA[7:0]														
	Reset Value																											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	I2C_STS1	Reserved						SCLLTO	SCLHTO	SDALTO	Reserved										Reserved	SMBALERT	Reserved	PECERR	OVERRUN	BUSERR	ARLOST	ACKFAIL	Reserved	ADDR10F	TXDATE	RXDATE	STOPF	BSF	ADDRF	STARTBTF						
	Reset Value							0	0	0												0		0	0	0	0	0	0		0	0	0	0	0	0	0					

018h	I2C_STS2	Reserved	PECVAL[7:0]								SMBHADDR	SMBDADDR	DUALFLAG	GCALLADD	Reserved	TRF	MSMODE	BUSY
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
01Ch	I2C_CLKCTRL	Reserved	DUTY	FSMODE	Reserved	CLKCTRL[11:0]												
	Reset Value		0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
020h	I2C_TMRISE	Reserved	Reserved											TMRISE[5:0]				
	Reset Value													0	0	0	0	0
024h	I2C_BYTENUM	Reserved	BYTENUMEN	RXFSEL	BYTENUM[13:0]													
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
028h	I2C_GFLTRCTRL	Reserved	SCLAFENN	Reserved	SCLAFW[1:0]		SDAAFENN	Reserved	SDAAFW[1:0]		SCLDFW[3:0]				SDADFW[3:0]			
	Reset Value		0		0	0	0		0	0	0	0	0	0	0	0		

19.6.2 I²C 控制寄存器 1 (I2C_CTRL1)

地址偏移：0x00

复位值：0x04400000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	LTOSEL	HTOSEL	Reserved												
rw		rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SW RESET	SMB ALERT	PEC	ACKPOS	ACKEN	STOP GEN	START GEN	NO EXTEND	GCEN	PECEN	ARPEN	SMB TYPE	SMB MODE	EN	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
30	Reserved	保留，必须保持复位值
30:29	LTOSEL	低超时阈值选择（low timeout threshold selection） 00: 25ms 01: 100ms 10: 1s 11: 4s
28:27	HTOSEL	高超时阈值选择（hightimeout threshold selection） 00: 256us 01: 512us 10: 1ms 11: 128ms
26:14	Reserved	保留，必须保持复位值
13	SWRESET	软件复位（Software reset） 在复位该位前确认 I ² C 的引脚被释放，总线是空的。 0: I ² C 模块不处于复位状态； 1: I ² C 模块处于复位状态。 <i>注：该位可以用于 I2C_STS2.BUSY 位为‘1’，在总线上又没有检测到停止条件时。</i>
12	SMBALERT	SMBus提醒（SMBus alert） 软件可以设置或清除该位；当I2C_CTRL1.EN=0时，由硬件清除。 0: 释放SMBAlert引脚使其变高。提醒响应地址头紧跟在NACK信号后面； 1: 驱动 SMBAlert 引脚使其变低。提醒响应地址头紧跟在 ACK 信号后面。
11	PEC	数据包出错检测（Packet error checking） 软件可以设置或清除该位；当传送完PEC后或检测到起始或停止条件时或当I2C_CTRL1.EN=0时，硬件均会将其清除。 0: 无 PEC 传输 1: PEC传输 <i>注：仲裁丢失时，PEC 的计算失效。</i>
10	ACKPOS	应答/PEC位置（用于数据接收）（Acknowledge/PEC Position（for data reception））

位域	名称	描述
		<p>软件可以设置或清除该位，或当I2C_CTRL1.EN=0时，由硬件清除。</p> <p>0: I2C_CTRL1.ACKEN 位决定是否向当前正在接收的字节发送ACK；I2C_CTRL1.PEC 位表示当前移位寄存器中的字节为 PEC。</p> <p>1: I2C_CTRL1.ACKEN 位决定是否向下一个接收到的字节发送ACK；I2C_CTRL1.PEC 位指示移位寄存器中接收到的下一个字节是 PEC。</p> <p>注：ACKPOS位只能用在2字节的接收配置中，必须在接收数据之前配置。</p> <p>为了NACK第2个字节，I2C_CTRL1.ACKEN 位必须在 I2C_STS1.ADDRF 位清零后清零。</p> <p>为了检测第 2 个字节的 PEC，必须在配置 ACKPOS 位之后，ADDR 延展事件时设置 I2C_CTRL1.PEC 位。</p>
9	ACKEN	<p>应答使能（Acknowledge enable）</p> <p>软件可以设置或清除该位，或当I2C_CTRL1.EN=0时，由硬件清除。</p> <p>0: 无应答返回；</p> <p>1: 在接收到一个字节后返回一个应答（匹配的地址或数据）。</p>
8	STOPGEN	<p>停止条件产生（Stop generation）</p> <p>软件可以设置或清除该位；或当检测到停止条件时，由硬件清除；当检测到SMBus超时错误时，硬件将其置位。</p> <p>在主模式下：</p> <p>0: 无停止条件产生；</p> <p>1: 在当前字节传输或在当前起始条件发出后产生停止条件。</p> <p>在从模式下：</p> <p>0: 无停止条件产生；</p> <p>1: 在当前字节传输后释放SCL和SDA线。</p> <p>注：当设置了STOPGEN、STARTGEN 或 PEC 位，在硬件清除这个位之前，软件不要执行任何对I2C_CTRL1 的写操作；否则有可能会第 2 次设置STOPGEN、STARTGEN 或 PEC 位。</p>
7	STARTGEN	<p>起始条件产生（Start generation）</p> <p>软件可以设置或清除该位，当起始条件发出后或I2C_CTRL1.EN=0时，由硬件清除。</p> <p>0: 无起始条件产生；</p> <p>1: 产生起始条件。</p>
6	NOEXTEND	<p>禁止时钟延长（从模式）（Clock stretching disable （Slave mode））</p> <p>该位决定在从机模式下数据未就绪（I2C_STS1.ADDRF 或 I2C_STS1.BSF 标志置位）时是否拉低 SCL。通过软件复位清零。</p> <p>0: 允许时钟延长；</p> <p>1: 禁止时钟延长。</p>
5	GCEN	<p>广播呼叫使能（General call enable）</p> <p>0: 禁止广播呼叫。不应答(NACK)地址 00h；</p> <p>1: 允许广播呼叫。以应答(ACK)地址 00h。</p>
4	PECEN	<p>PEC使能（PEC enable）</p> <p>0: 禁止PEC模式；</p> <p>1: 开启 PEC 模式。</p>
3	ARPEN	<p>ARP使能（ARP enable）</p> <p>0: 禁止ARP；</p> <p>1: 使能ARP。</p> <p>如果I2C_CTRL1.SMBTYPE=0，使用SMBus设备的默认地址。</p>

位域	名称	描述
		如果 I2C_CTRL1.SMBTYPE=1，使用 SMBus 的主地址。
2	SMBTYPE	SMBus类型（SMBus type） 0：SMBus设备； 1：SMBus 主机。
1	SMBMODE	SMBus模式（SMBus mode） 0：I2C模式； 1：SMBus 模式。
0	EN	I ² C模块使能（Peripheral enable） 0：禁用I ² C模块； 1：启用I ² C模块。 <i>注：如果清除该位时通讯正在进行，在当前通讯结束后，I2C 模块被禁用并返回空闲状态。由于在通讯结束后发生EN=0，所有的位被清除。 在主模式下，通讯结束之前，绝不能清除该位。</i>

19.6.3 I²C 控制寄存器 2 (I2C_CTRL2)

地址偏移：0x04

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	LTOEN	HTOEN	SCLLTO INTEN	SCLHTO INTEN	SDALTO INTEN	Reserved									
	rw	rw	rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN	ERRINT EN	EVTINT EN	BUFINT EN	Reserved			DMA LAST	Reserved	CLKFREQ[6:0]						
rw	rw	rw	rw				rw		rw						

位域	名称	描述
31	Reserved	保留，必须保持复位值
30	LTOEN	低超时功能使能（low timeout function enable） 0：禁用； 1：使能； <i>注：I2C 模式下不需要超时报错时可禁用此位。</i>
29	HTOEN	高超时功能使能（high timeout function enable） 0：禁用； 1：使能； <i>注：I2C 模式下不需要超时报错时可禁用此位。</i>
28	SCLLTOINTEN	SCL低超时错误中断使能（SCL low timeout error interrupt enable） 0：I2C_STS1.SCLLTO = 1时，不产生中断； 1：I2C_STS1.SCLLTO = 1 时，产生中断；
27	SCLHTOINTEN	SCL高超时错误中断使能（SCL high timeout error interrupt enable） 该位由软件写‘0’清除，或在I2C_CTRL1.EN=0=0时由硬件清除。

位域	名称	描述
		0: I2C_STS1.SCLHTO = 1时, 不产生中断; 1: I2C_STS1.SCLHTO = 1时, 产生中断;
26	SDALTOINTEN	SDA低超时错误中断使能 (SDA low timeout error interrupt enable) 该位由软件写‘0’清除, 或在I2C_CTRL1.EN=0=0时由硬件清除。 0: I2C_STS1.SDALTO = 1时, 不产生中断; 1: I2C_STS1.SDALTO = 1时, 产生中断;
25:16	Reserved	保留, 必须保持复位值
15	DMAEN	DMA请求使能 (DMA requests enable) 0: 禁止DMA请求; 1: 使能 DMA 请求。
14	ERRINTEN	出错中断使能 (Error interrupt enable) 0: 禁止出错中断; 1: 允许出错中断。 在下列条件下, 将产生该中断: I2C_STS1.BUSERR = 1 I2C_STS1.ARLOST = 1 I2C_STS1.ACKFAIL = 1 I2C_STS1.OVERRUN = 1 I2C_STS1.PECERR = 1 I2C_STS1.SMBALERT = 1
13	EVTINTEN	事件中断使能 (Event interrupt enable) 0: 禁止事件中断 1: 允许事件中断 在下列条件下, 将产生该中断: I2C_STS1.STARTBF = 1 (主模式) I2C_STS1.ADDRF = 1 (主/从模式) I2C_STS1.ADD10F = 1 (主模式) I2C_STS1.STOPF = 1 (从模式) I2C_STS1.BSF = 1, 但是没有I2C_STS1.TXDATE或I2C_STS1.RXDATNE事件 如果I2C_STS1.BUFINTEN = 1, I2C_STS1.TXDATE标志为1 如果 I2C_STS1.BUFINTEN = 1, I2C_STS1.RXDATNE 标志为 1
12	BUFINTEN	缓冲器中断使能 (Buffer interrupt enable) 0: 当 I2C_STS1.TXDATE=1 或 I2C_STS1.RXDATNE=1 时, 不产生任何中断; 1: 当 I2C_STS1.TXDATE=1 或 I2C_STS1.RXDATNE=1 时 (I2C_CTRL2.EVTINTEN=1), 产生事件中断 (不管 DMAEN 是何种状态)。
11:9	Reserved	保留, 必须保持复位值
8	DMALAST	DMA最后一次传输 (DMA last transfer) 0: 下一次DMA的EOT不是最后的传输; 1: 下一次DMA的EOT是最后的传输。 注: 该位在主接收模式使用, 使得在最后一次接收数据时可以产生一个NACK。
7	Reserved	保留, 必须保持复位值
6:0	CLKFREQ[6:0]	I2C模块时钟频率 (Peripheral clock frequency) CLKFREQ[6:0] 应该为APB1时钟频率以产生正确的时序:

位域	名称	描述
		0000000: 禁用 0000001: 禁用 0000010: 2MHz 0000011: 3MHz ... 0100000: 32MHz 0100001~1111101: 禁用

19.6.4 I²C 自身地址寄存器 1 (I2C_OADDR1)

地址偏移: 0x08

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR MODE	Reserved	Reserved					ADDR[9:8]	ADDR[7:1]							ADDR0
rw							rw	rw							rw

位域	名称	描述
15	ADDRMODE	寻址模式（从模式）（Addressing mode（slave mode）） 0: 7位从地址（不响应10位地址）； 1: 10 位从地址（不响应 7 位地址）。
14	Reserved	必须始终由软件保持为‘1’。
13:10	Reserved	保留，必须保持复位值
9:8	ADDR[9:8]	接口地址（Interface address） 地址的 9~8 位。 <i>注：7 位地址模式时不用关心</i>
7:1	ADDR[7:1]	接口地址（Interface address） 地址的 7~1 位。
0	ADDR0	接口地址（Interface address） 地址第 0 位。 <i>注：7 位地址模式时不用关心</i>

19.6.5 I²C 自身地址寄存器 2 (I2C_OADDR2)

地址偏移: 0x0C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADDR2[7:1]							DUALEN
								rw							rw

位域	名称	描述
15:8	Reserved	保留，必须保持复位值
7:1	ADDR2[7:1]	接口地址（Interface address） 在双地址模式下地址的 7~1 位。
0	DUALEN	双地址模式使能位（Dual addressing mode enable） 0：不使能双地址模式，只有OADDR1被识别； 1：使能双地址模式，OADDR1和OADDR2都被识别。 <i>注：仅 7 位地址模式有效</i>

19.6.6 I²C 数据寄存器 (I2C_DAT)

地址偏移：0x10

复位值：0x0000

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

位域	名称	描述
15:8	Reserved	保留，必须保持复位值
7:0	DATA[7:0]	8位数据寄存器（8-bit data register） 发送或接收数据 <i>注意：从模式下。地址不会被拷贝到数据寄存器</i> <i>注意：如果 I2C_STS1.TXDATE = 0，数据仍然会被写入数据寄存器</i> <i>注意：如果在处理 ACK 时，发生了 ARLOST 事件，接收到的字节不会被拷贝到数据寄存器，因此不能读到它。</i>

19.6.7 I²C 状态寄存器 1 (I2C_STS1)

地址偏移：0x14

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						SCLLTO	SCLHTO	SDALTO	Reserved						
rc_w0						rc_w0	rc_w0	rc_w0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SMB ALERT	Reserved	PECERR	OVER RUN	BUSERR	ARLOST	ACKFAIL	Reserved	ADDR10F	TXDATE	RXDAT NE	STOPF	BSF	ADDRF	STARTBF
rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		r	r	r	r	r	r	r

位域	名称	描述
31:26	Reserved	保留，必须保持复位值

位域	名称	描述
25	SCLLTO	<p>SCL低超时错误 (SCL low timeout error)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0=0时由硬件清除。</p> <p>0: 无超时错误；</p> <p>1: 超时错误发生；</p> <p>从模式超时：从设备复位通讯，硬件释放总线。</p> <p>主模式超时：硬件发出停止条件。</p>
24	SCLHTO	<p>SCL高超时错误 (SCL high timeout error)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0=0时由硬件清除。</p> <p>0: 无超时错误；</p> <p>1: 超时错误发生；</p> <p>从模式超时：从设备复位通讯，硬件释放总线。</p> <p>主模式超时：硬件发出停止条件。</p>
23	SDALTO	<p>SDA低超时错误 (SDA low timeout error)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0=0时由硬件清除。</p> <p>0: 无超时错误；</p> <p>1: 超时错误发生；</p> <p>从模式超时：从设备复位通讯，硬件释放总线。</p> <p>主模式超时：硬件发出停止条件。</p>
22:15	Reserved	保留，必须保持复位值
14	SMBALERT	<p>SMBus提醒 (SMBus alert)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0时由硬件清除。</p> <p>0: 无SMBus提醒（主模式）或没有SMBAlert响应地址头序列（从模式）；</p> <p>1: 在引脚上产生 SMBAlert 提醒事件（主模式）或收到 SMBAlert 响应地址（从模式）；</p>
13	Reserved	保留，必须保持复位值
12	PECERR	<p>在接收时发生PEC错误 (PEC Error in reception)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0时由硬件清除。</p> <p>0: 无 PEC 错误: 接收到 PEC 后接收器返回 ACK（如果 I2C_CTRL1.ACKEN=1）</p> <p>1: PEC 错误: 接收到 PEC 后接收器返回 NACK（不管 I2C_CTRL1.ACKEN 是否置位）</p>
11	OVERRUN	<p>过载/欠载 (Overrun/Underrun)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0时由硬件清除。</p> <p>0: 无过载/欠载；</p> <p>1: 出现过载/欠载。</p> <p>当I2C_CTRL1.NOEXTEND=1时，在从模式下该位被硬件置位。同时：</p> <p>在接收模式中当接收到一个新的字节时，如果数据寄存器里的内容还未被读出，则发生过载错误，新接收的字节将会丢失。</p> <p>在发送模式中要发送一个新的字节时，却没有新的数据写入数寄存器，将发生欠载错误，同样的数据将会被发送两次。</p>
10	BUSERR	<p>总线错误 (Bus error)</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0时由硬件清除。</p> <p>0: 无起始或停止条件出错；</p> <p>1: 起始或停止条件出错。</p>
9	ARLOST	<p>仲裁丢失（主模式）</p> <p>该位由软件写‘0’清除，或在I2C_CTRL1.EN=0时由硬件清除。</p>

位域	名称	描述
		<p>0: 没有仲裁丢失; 1: 仲裁丢失。</p> <p>当接口失去对总线的控制时, 硬件将置该位为‘1’。在ARLOST事件之后, I²C接口自动切换回从模式(I2C_STS2.MSMODE=0)。</p> <p><i>注: 在SMBUS 模式下, 在从模式下对数据的仲裁仅仅发生在数据阶段, 或应答传输区间(不包括地址的应答)。</i></p>
8	ACKFAIL	<p>应答失败 (Acknowledge failure)</p> <p>该位由软件写‘0’清除, 或在I2C_CTRL1.EN=0时由硬件清除。</p> <p>0: 没有应答失败; 1: 应答失败。</p>
7	Reserved	保留, 必须保持复位值
6	ADDR10F	<p>10位头序列已发送 (主模式)</p> <p>软件读取STS1寄存器后, 对CTRL1寄存器的写操作将清除该位, 或当I2C_CTRL1.EN=0时, 硬件清除该位。</p> <p>0: 没有ADD10F事件发生; 1: 主设备已经将第一个地址字节发送出去。</p> <p>在10位地址模式下, 当主设备已经将第一个地址字节发送出去时, 硬件将该位置‘1’。</p> <p><i>注: 收到一个NACK 后, I2C_STS1.ADD10F 位不被置位。</i></p>
5	TXDATE	<p>数据寄存器为空 (发送时)</p> <p>软件写数据到DAT寄存器可清除该位; 或在发生一个起始或停止条件后, 或当I2C_CTRL1.EN=0时由硬件自动清除。</p> <p>0: 数据寄存器非空; 1: 数据寄存器空。</p> <p>在发送数据时, 数据寄存器为空时该位被置‘1’, 在发送地址阶段不设置该位。</p> <p>如果收到一个NACK, 或下一个要发送的字节是PEC (I2C_CTRL1.PEC=1), 该位不被置位。</p> <p><i>注: 在写入第1 个要发送的数据后, 或设置了BSF 时写入数据, 都不能清除TXDATE 位, 这是因为数据寄存器仍然为空。</i></p>
4	RXDATNE	<p>数据寄存器非空 (接收时)</p> <p>软件对数据寄存器的读写操作清除该位, 或当I2C_CTRL1.EN=0时由硬件清除。</p> <p>0: 数据寄存器为空; 1: 数据寄存器非空。</p> <p>在接收时, 当数据寄存器不为空, 该位被置‘1’。在接收地址阶段, 该位不被置位。</p> <p>在发生ARLOST事件时, RXDATNE不被置位。</p> <p><i>注: 当设置了BSF 时, 读取数据不能清除RXDATNE 位, 因为数据寄存器仍然为满。</i></p>
3	STOPF	<p>停止条件检测位 (从模式)</p> <p>软件读取STS1寄存器后, 对I2C_CTRL1寄存器的写操作将清除该位, 或当I2C_CTRL1.EN=0时, 硬件清除该位。</p> <p>0: 没有检测到停止条件; 1: 检测到停止条件。</p> <p>在一个应答之后, 当从设备在总线上检测到停止条件时, 硬件将该位置‘1’。</p> <p><i>注: 在收到NACK 后, I2C_STS1.STOPF 位不被置位。</i></p>
2	BSF	字节传输结束 (Byte transfer finished)

位域	名称	描述
		<p>在软件读取STS1寄存器后，对数据寄存器的读或写操作将清除该位；或在传输中发送一个起始或停止条件后，或当I2C_CTRL1.EN =0时，由硬件清除该位。</p> <p>0：字节传输未完成；</p> <p>1：字节传输结束。</p> <p>当I2C_CTRL1.NOEXTEND =0时，在下列情况下硬件将该位置‘1’：</p> <p>在接收时，当收到一个新字节（包括ACK脉冲）且数据寄存器还未被读取（I2C_STS1.RXDATNE =1）。</p> <p>在发送时，当一个新数据将被发送且数据寄存器还未被写入新的数据（I2C_STS1.TXDATE =1）。</p> <p><i>注：在收到一个NACK后，BSF位不会被置位。</i></p> <p>如果下一个要传输的字节是 PEC（I2C_STS2.TRF 为‘1’，同时 I2C_CTRL1.PEC 为‘1’），BSF 位不会被置位。</p>
1	ADDRF	<p>地址已被发送（主模式）/地址匹配（从模式）</p> <p>在软件读取STS1寄存器后，对STS2寄存器的读操作将清除该位，或当I2C_CTRL1.EN=0时，由硬件清除该位。</p> <p>0：地址不匹配或没有收到地址（从模式），地址发送未完成（主模式）；</p> <p>1：收到的地址匹配（从模式），地址发送完成（主模式）。</p> <p>在主模式下：</p> <p>7位地址模式时，当收到地址的ACK后该位被置‘1’，10位地址模式时，当收到地址的第二个字节的ACK后该位被置‘1’。</p> <p>在从模式下：</p> <p>当收到的从地址与OADDR寄存器中的内容相匹配，或广播呼叫或SMBus设备默认地址或SMBus主机或SMBus提醒被识别时，硬件就将该位置‘1’（当对应的设置被使能时）。</p> <p><i>注：在收到NACK后，I2C_STS1.ADDRF 位不会被置位。</i></p>
0	STARTBF	<p>起始位（主模式）</p> <p>软件读取I2C_STS1寄存器后，写数据寄存器的操作将清除该位，或当I2C_CTRL1.EN=0时，硬件清除该位。</p> <p>0：未发送起始条件；</p> <p>1：起始条件已发送。</p> <p>当发送出起始条件时该位被置‘1’。</p>

19.6.8 I²C 状态寄存器 2 (I2C_STS2)

地址偏移：0x18

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PECVAL[7:0]								SMBH ADDR	SMBD ADDR	DUAL FLAG	GCALL ADDR	Reserved	TRF	MS MODE	BUSY
r								r	r	r	r		r	r	r

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:8	PECVAL[7:0]	数据包出错检测（Packet error checking register） 当 I2C_CTRL1.PECEN=1 时，PECVAL[7:0]存放内部的 PEC 值。
7	SMBHADDR	SMBus主机头地址（从模式） 在产生一个停止条件或一个重复的起始条件时，或I2C_CTRL1.EN =0时，硬件将该位清除。 0：未收到SMBus主机的地址； 1：当 I2C_CTRL1.SMBTYPE=1 且 I2C_CTRL1.ARPEN=1 时，收到 SMBus 主机地址。
6	SMBDADDR	SMBus 设备默认地址（从模式） 在产生一个停止条件或一个重复的起始条件时，或I2C_CTRL1.EN=0时，硬件将该位清除。 0：未收到SMBus设备的默认地址； 1：当 I2C_CTRL1.ARPEN=1 时，收到 SMBus 设备的默认地址。
5	DUALFLAG	双标志（从模式） 在产生一个停止条件或一个重复的起始条件时，或I2C_CTRL1.EN =0时，硬件将该位清除。 0：接收到的地址与OADDR1内的内容相匹配； 1：接收到的地址与 OADDR2 内的内容相匹配。
4	GCALLADDR	广播呼叫地址（从模式） 在产生一个停止条件或一个重复的起始条件时，或I2C_CTRL1.EN=0时，硬件将该位清除。 0：未收到广播呼叫地址； 1：当 I2C_CTRL1.GCEN =1 时，收到广播呼叫的地址。
3	Reserved	保留，必须保持复位值
2	TRF	发送/接收（Transmitter/receiver） 在检测到停止条件（I2C_STS1.STOPF =1）、重复的起始条件或总线仲裁丢失（I2C_STS1.ARLOST =1）后，或当I2C_CTRL1.EN=0时，硬件将其清除。 0：数据接收模式； 1：数据发送模式； 在整个地址传输阶段的结尾，该位根据地址字节的 R/W 位来设定。
1	MSMODE	主从模式（Master/slave） 当总线上检测到一个停止条件、仲裁丢失（I2C_STS1.ARLOST =1）时、或当 I2C_CTRL1.EN =0时，硬件清除该位。 0：从模式； 1：主模式。 当接口处于主模式（I2C_CTRL1.STARTBF=1）时，硬件将该位置位；
0	BUSY	总线忙（Bus busy） 当检测到一个停止条件时，硬件将该位清除。 0：在总线上无数据通讯； 1：在总线上正在进行数据通讯。 在检测到SDA或SCL为低电平时，硬件将该位置‘1’；

位域	名称	描述
		注：该位指示当前正在进行的总线通讯，当接口被禁用（I2C_CTRL1.EN = 0）时该信息仍然被更新。

19.6.9 I²C 时钟控制寄存器 (I2C_CLKCTRL)

地址偏移：0x1C

复位值：0x0000

注：CLKCTRL 寄存器只有在关闭 I²C 时（I2C_CTRL1.EN = 0）才能设置。

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

位域	名称	描述
15	DUTY	快速模式占空比 0: Tlow/Thigh= 2; 1: Tlow/Thigh= 16/9
14	FSMODE	I2C 主模式选择 0: I2C 标准模式(占空比默认 1/1); 1: I2C 快速模式(占空比可配置)
13:12	Reserved	保留，必须保持复位值
11:0	CLKCTRL[11:0]	快速/标准模式下的时钟控制分频系数（主模式） 该分频系数用于设置主模式下的 SCL 时钟。 ■ 如果 duty cycle = Tlow/Thigh = 1/1: $CLKCTRL = f_{PCLK1}(Hz)/100000/2$ $T_{low} = CLKCTRL \times T_{PCLK1}$ $T_{high} = CLKCTRL \times T_{PCLK1}$ ■ 如果 duty cycle = Tlow/Thigh = 2/1: $CLKCTRL = f_{PCLK1}(Hz)/100000/3$ $T_{low} = 2 \times CLKCTRL \times T_{PCLK1}$ $T_{high} = CLKCTRL \times T_{PCLK1}$ ■ 如果 duty cycle = Tlow/Thigh = 16/9: $CLKCTRL = f_{PCLK1}(Hz)/100000/25$ $T_{low} = 16 \times CLKCTRL \times T_{PCLK1}$ $T_{high} = 9 \times CLKCTRL \times T_{PCLK1}$ 例如，如果 $f_{PCLK1}(Hz) = 8MHz$, duty cycle = 1/1, $CLKCTRL = 8000000/100000/2 = 0x28$ 。 注意：1. 允许设定的最小值为 0x04，在快速 DUTY 模式下允许的最小值为 0x01。 2. $T_{high} = T_{f(SCL)} + T_{w(SCLH)}$ ，详见数据手册中对这些参数的定义。 3. $T_{low} = T_{f(SCL)} + T_{w(SCLL)}$ ，详见数据手册中对这些参数的定义。

19.6.10 I²C 上升时间寄存器 (I2C_TMRISE)

地址偏移: 0x20

复位值: 0x0002

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

位域	名称	描述
15:6	Reserved	保留, 必须保持复位值
5:0	TMRISE[5:0]	<p>在快速/标准模式下的最大上升时间 (主模式)</p> <p>这些位必须设置为I²C总线规范里给出的最大的SCL上升时间, 增长步幅为1。</p> <p>例如, 标准模式中最大允许SCL上升时间为1000ns。如果I2C_CTRL2.CLKFREQ[5:0]中的值等于0x08(8MHz)且T_{PCLK1}=125ns, 故TMRISE[5:0]中必须写入09h (1000ns/125 ns + 1)。</p> <p>如果结果不是一个整数, 则将整数部分写入 TMRISE[5:0]以确保 t_{HIGH} 参数。</p> <p>注: 只有当I2C 禁用 (I2C_CTRL1.EN=0) 时才能配置 TMRISE[5:0]</p>

19.6.11 I²C 主机接收字节寄存器 (I2C_BYTENUM)

地址偏移: 0x24

复位值: 0x0000

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

位域	名称	描述
15	BYTENUMEN	<p>主机接收字节控制功能使能</p> <p>0: 禁用</p> <p>1: 使能</p>
14	RXFSEL	<p>接收结束发送条件选择</p> <p>0: 主机接收完所有字节后发送 STOP 条件</p> <p>1: 主机接收完所有字节后发送 START 条件</p>
13:0	BYTENUM[13:0]	<p>主机接收字节数配置</p> <p>注意: 接收完所有字节后若需重新配置 BYTENUM, 需先等待 I2C_STS2.BUSY 为 0, 并重新使能 I2C_CTRL1.ACKEN</p>

19.6.12 I²C 滤波控制寄存器 (I2C_GFLTRCTRL)

地址偏移: 0x28

复位值：0x0000

注: *GFLTRCTRL* 寄存器只有在关闭 I^2C 时 ($I2C_CTRL1.EN=0$) 才能设置。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLAFENN	Reserved	SCLAFW[1:0]	SDAAFENN	Reserved	SDAAFW[1:0]	SCLDFW[3:0]			SDADFW[3:0]						
rw		rw	rw		rw	rw			rw						

位域	名称	描述
15	SCLAFENN	SCL 模拟滤波器使能 0: 使能 1: 禁用
14	Reserved	Reserved
13:12	SCLAFW[1:0]	SCL 模拟滤波器宽度选择 00: 5ns 01: 15ns 10: 25ns 11: 35ns
11	SDAAFENN	SDA 模拟滤波器使能 0: 使能 1: 禁用
10	Reserved	Reserved
9:8	SDAAFW[1:0]	SDA 模拟滤波器宽度选择: 00: 5ns 01: 15ns 10: 25ns 11: 35ns
7:4	SCLDFW[3:0]	SCL 数字滤波器宽度选择 0000: 禁用 SCL 数字滤波器 其他值: 滤波宽度为 $SCLDFW * T_{PCLK}$
3:0	SDADFW[3:0]	SDA 数字滤波器宽度选择 0000: 禁用 SDA 数字滤波器 其他值: 滤波宽度为 $SDADFW * T_{PCLK}$

20 通用异步收发器(UART)

20.1 简介

通用异步收发器（UART）是一种全双工串行数据交换接口，支持异步通信。可灵活配置，以便于与多种外部设备进行全双工数据交换。

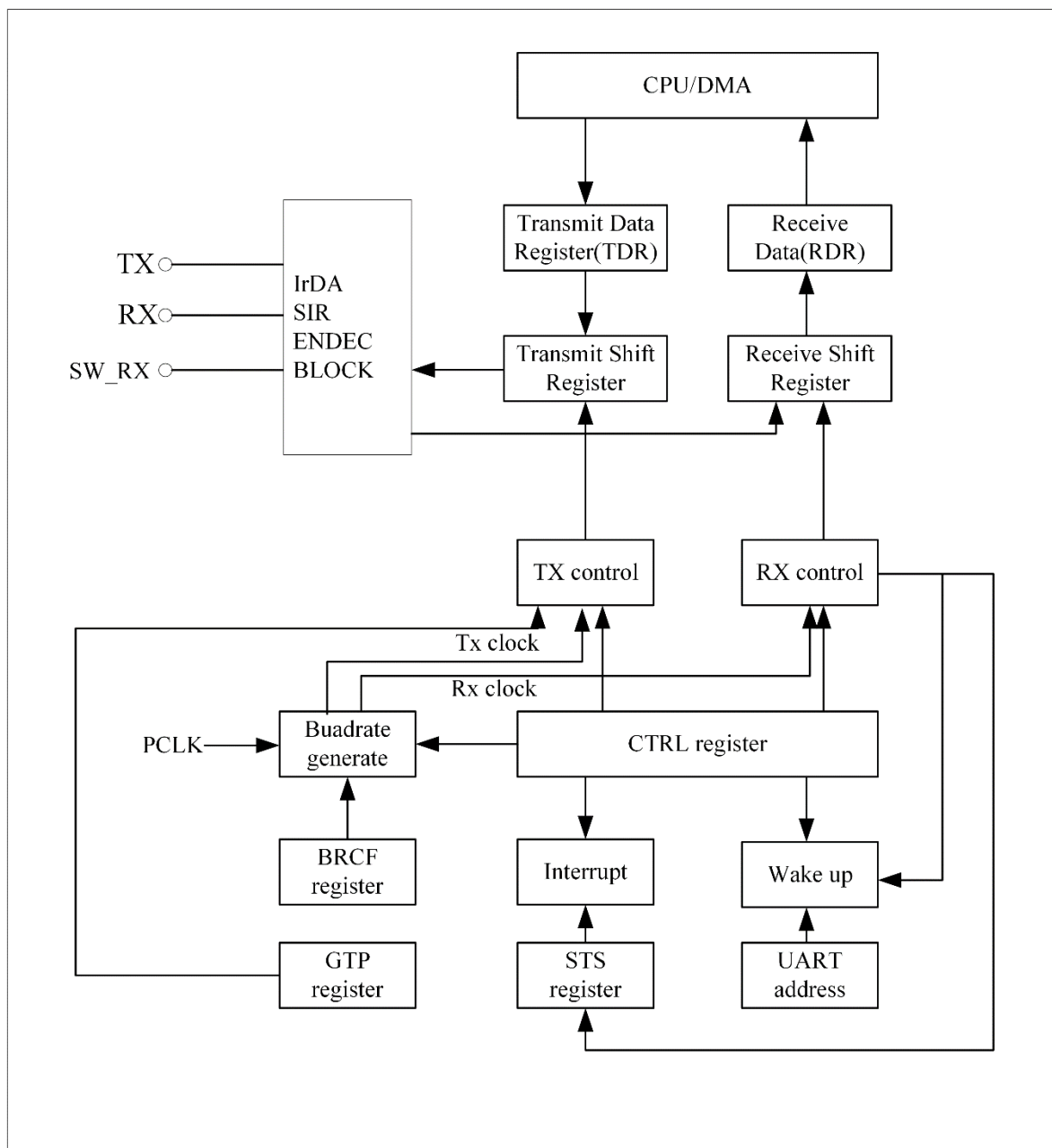
UART 接口发送与接收波特率可配置，也支持通过 DMA 进行连续通信。UART 还支持多处理器通信、LIN 模式、单线半双工通信、IrDA SIR ENDEC 功能。

20.2 主要特性

- 支持全双工通信
- 支持单线半双工通信
- 波特率可配置，最高波特率可达 4Mbit/s
- 支持 8bit 或 9bit 数据帧
- 支持 1bit 或 2bit 停止位
- 支持硬件生成校验位及校验位检查
- 支持 DMA 收发
- 支持多处理器通信：如果地址不匹配，则进入静默模式，可通过空闲总线检测或地址标识唤醒
- 支持串行红外协议（IrDA SIR）编码与解码，提供正常与低功耗两种运行模式
- 支持 LIN 模式
- 支持多钟错误检测：数据溢出错误、帧错误、噪声错误、检验错误
- 支持多个中断请求：发送数据寄存器为空、发送完成、数据已接收、数据溢出、总线空闲、检验错误、LIN 模式断开帧检测、以及多缓冲区通信中的噪声标志/溢出错误/帧错误

20.3 功能框图

图 20-1 UART 框图



20.4 功能描述

如 图 20-1 所示, UART 的双向通信都需要使用 RX 和 TX 引脚与外部器件连接。其中 TX 为数据发送引脚（输出），当发送功能使能但没有发送数据时，TX 引脚输出高电平，当发送功能被禁用时，TX 引脚为普通 IO 端口，状态由应 IO 配置决定。RX 为数据接收引脚（输入），接收数据时采用了过采样技术。

当设备作为发送端时，通过 TX 引脚发送数据，作为接收端时则通过 RX 引脚接收数据。当没有数据收发时，总线处于空闲状态。数据帧格式为：1 个起始位+ 8 或 9 位数据（最低有效位在前）+ 1 个检验位（可选）+ 0.5,1,1.5 或 2 个停止位。

使用分数波特率发生器来配置发送与接收波特率。

20.4.1 UART 帧格式

起始位：1 位，低电平有效

数据位：可通过 UART_CTRL1.WL 配置为 8 或 9 位，最低有效位在前。

停止位：高电平有效。

空闲帧：全部由‘1’组成的一个完整的数据帧，包括起始位。后跟包含数据的数据帧的起始位。

断开帧：全部由‘0’组成的一个完整的数据帧，包括停止位。在断开帧结束后，发送端再插入 1 或 2 个停止位来应答起始位。

图 20-2 字长=8 设置

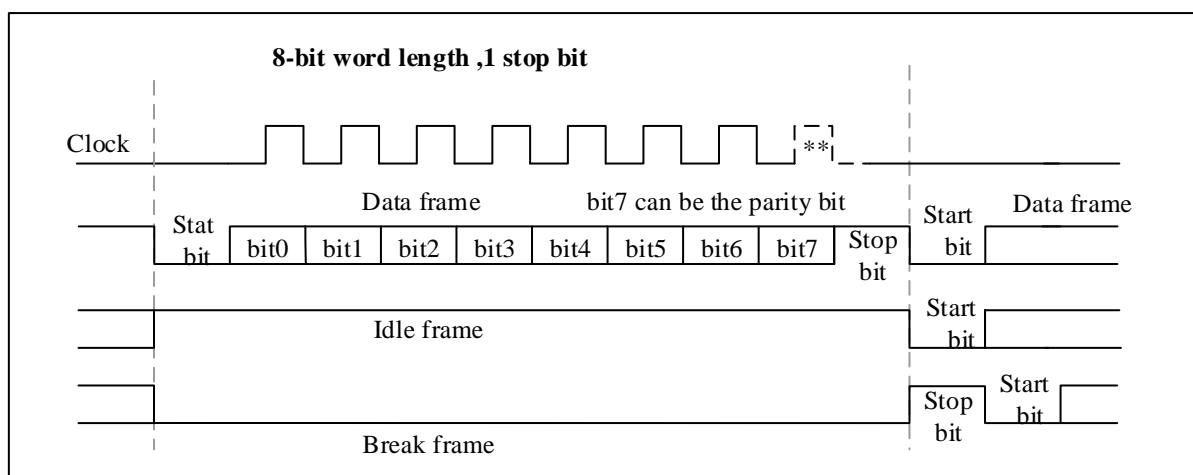
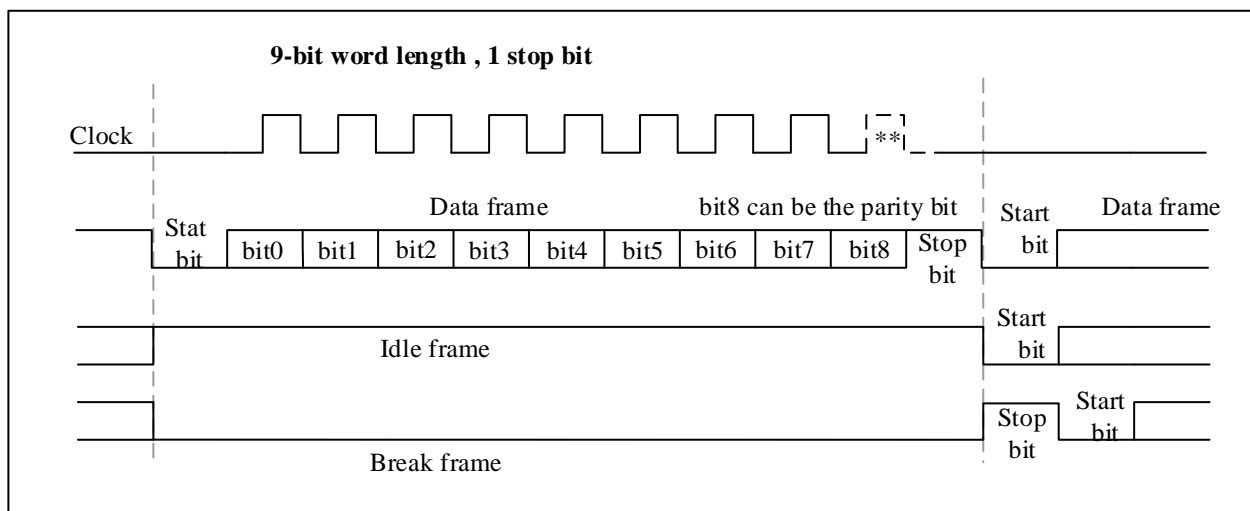


图 20-3 字长=9 设置



20.4.2 发送器

当发送功能使能后，进入移位寄存器中的数据通过 TX 引脚输出。

20.4.2.1 空闲帧

UART_CTRL1.TXEN 置 1 后，UART 会在发送数据之前发送一个空闲帧。

20.4.2.2 字符发送

在空闲帧结束后，字符可正常发送。在每个字符发送前，先发送一个起始位（低电平）。发送器根据数据长度配置发送 8 位或 9 位数据，其中最低有效位先发送。如果在数据传输时 UART_CTRL1.TXEN 被清零，将导致波特率计数器停止计数，从而破坏正在传输的数据。

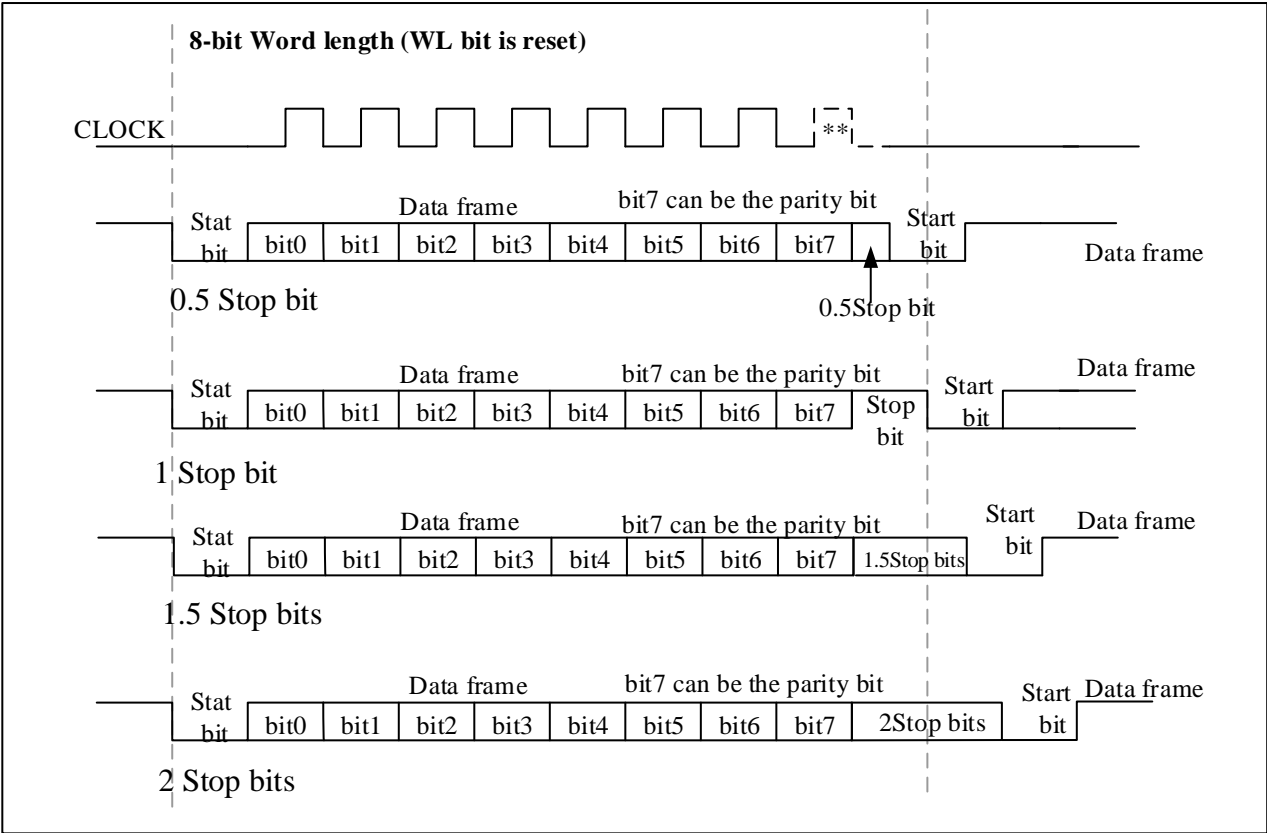
20.4.2.3 停止位

字符发送完成后，发送器自动发送停止位。停止位位数可通过 UART_CTRL2.STPB[1:0]配置。

表 20-1 停止位配置

UART_CTRL2.STPB[1:0]	停止位长度 (位)	功能描述
00	1	默认
01	0.5	
10	2	用于常规 UART 模式、单线模式以及调制解调器模式.
11	1.5	

图 20-4 停止位配置



20.4.2.4 断开帧

可通过置位 UART_CTRL1.SDBRK 来发送 1 个断开帧。当数据长度为 8 位时，断开帧由 10 位低电平组成，当数据长度为 9 位时，断开帧由 11 位低电平组成。断开帧结束后将插入一位停止位（高电平）。

断开帧发送完成后，UART_CTRL1.SDBRK 被硬件清零，同时自动发送停止位。因此，如果要连续发送断开帧，必须在前一个断开帧与停止位发送完成后再次置位 UART_CTRL1.SDBRK。

如果在断开帧开始发送前软件清零 UART_CTRL1.SDBRK，当前断开帧不会发送。

20.4.2.5 发送流程

1. 置位 UART_CTRL1.UEN 来使能 UART；
2. 配置波特率、数据长度、校验位、停止位长度、以及根据需要配置相关 DMA；
3. 使能发送功能 (UART_CTRL1.TXEN)；
4. 通过 CPU 或 DMA 将要发送的数据依次写入数据寄存器 UART_DAT，当数据写入数据寄存器时将清零 UART_STS.TXDE；
5. 当所有数据已写入到数据寄存器 UART_DAT 后，等待发送完成标志位 UART_STS.TXC 置 1，数据发送完成。

20.4.2.6 单字节通信

对数据寄存器 UART_DAT 的写操作将清零标志位 UART_STS.TXDE。

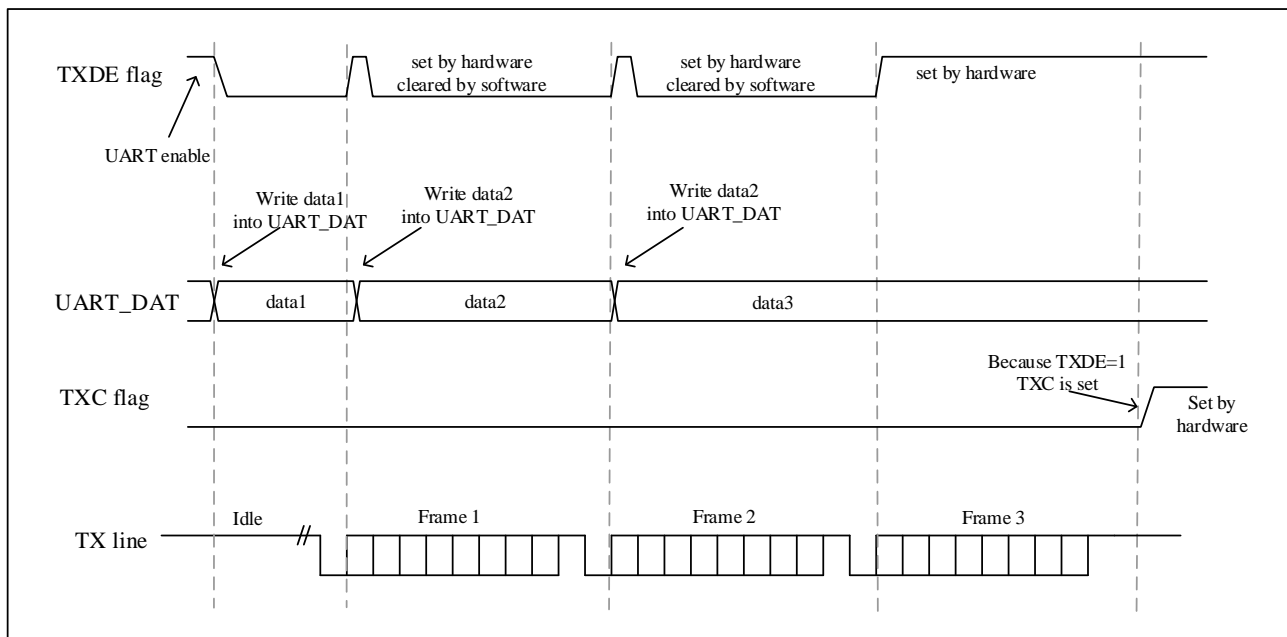
当数据已从发送数据寄存器送到移位寄存器时，UART_STS.TXDE 位由硬件置 1，表示数据开始发送。如果 UART_CTRL1.TXDEIEN 已置 1，将产生一个中断。此时，可将下一个数据写入数据寄存器 UART_DAT。

对数据寄存器 UART_DAT 进行写操作时：

- 如果移位寄存器空闲，数据将直接送到移位寄存器，同时 UART_STS.TXDE 硬件置 1
- 如果移位寄存器正在发送数据，数据保存在数据寄存器，待上一个数据发送完成后，再送到移位寄存器

当一帧数据发送完成后并且 UART_STS.TXDE 置 1，UART_STS.TXC 被硬件置 1。如果 UART_CTRL1.TXCIEN 已置 1，将产生一个中断。UART_STS.TXC 通过以下软件操作清零：先读一次 UART_STS 寄存器，再写一次 UART_DAT 寄存器。

图 20-5 发送时 TXC/TXDE 的变化情况



20.4.3 接收器

20.4.3.1 起始位检测

在 UART 中，如果识别到一个特殊的采样序列 1 1 1 0 X 0 X 0 X 0 0 0 0，就认为检测到一个起始位。

在第 3、5、7 位的采样，以及在第 8、9、10 位的采样都为'0'（也即 6 个'0'），则确认收到起始位，并将 UART_STS.RXDNE 置 1，但不会置位 NEF 噪声标志。如果 UART_CTRL1.RXDNEIEN 已置 1，则产生一个中断。

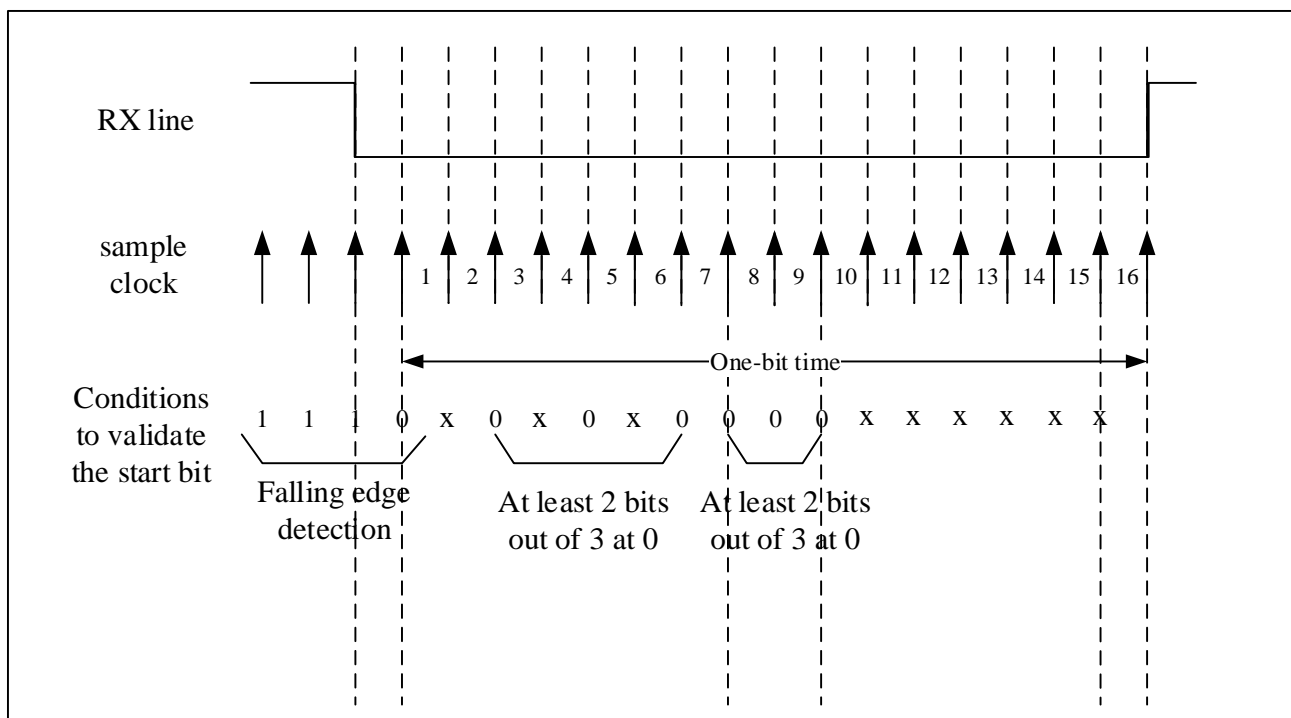
第 3、5、7 位的采样有两个'0'，与此同时，第 8、9、10 位的采样有两个'0'点，也确认收到起始位，但是会置位 NEF 噪声标志位。

第 3、5、7 位的采样有三个'0'，与此同时，第 8、9、10 位的采样有两个'0'点，也确认收到起始位，并置位 NEF 噪声标志位。

第 3、5、7 位的采样有两个'0'，与此同时，第 8、9、10 位的采样有三个'0'点，也确认收到起始位，并置位 NEF 噪声标志位。

如果在第 3、5、7、8、9、10 位的采样值满足不了上面四种要求，UART 接收器认为没有接受到正确的起始位，将退出起始位侦测并回到空闲状态等待下降沿。

图 20-6 起始位检测



20.4.3.2 停止位

停止位长度可通过 UART_CTRL2.STPB[1:0]配置。常规模式下，可配置为 1 位或 2 位。

- 0.5 位停止位：不对停止位进行采样。因此，此时不能检测帧错误和断开帧。
- 1 个停止位：默认情况下通过三个点对 1 个停止位的采样，选择第 8，第 9 和第 10 采样位上进行。
- 1.5 个停止位：对 1.5 个停止位的采样是在第 16，第 17 和第 18 采样点进行的，可分成两个部分：0.5 个数据位周期，接收器不做任何处理；然后是 1 个数据位周期，接收器对其进行采样。
- 2 个停止位：对 2 个停止位的采样是在第一停止位的第 8，第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误，帧错误标志将被置起。在第一个停止位结束时 UART_STS.RXDNE 标志将被设置。第二个停止位将不会检测帧错误。

20.4.3.3 接收流程

- 将 UART_CTRL1.UEN 置 1 来使能 UART；
- 配置波特率、数据长度、校验位、停止位长度、以及根据需要配置相关 DMA；
- 使能接收器 (UART_CTRL1.RXEN)，开始起始位检测；
- 接收 8 位或 9 位数据，通过 RX 引脚送往接收移位寄存器，最低有效位在前；
- 当数据由接收移位寄存器送到 RDR 寄存器，UART_STS.RXDNE 被置 1，表示数据可以被读出。如果 UART_CTRL2.RXNEIEN 已置 1，将产生一个中断；
- 当接收过程中检测到帧错误、噪音或溢出错误，这样错误标志将被置 1。如果在数据传输过程中 UART_CTRL1.RXEN 被清零，当前接收数据丢失；
- UART_STS.RXDNE 通过对 UART_DAT 寄存器进行读操作清零；

- 在多缓冲器通信模式, UART_STS.RXDNE 通过 DMA 对数据寄存器的读操作清零。
- 在单缓冲器通信模式, UART_STS.RXDNE 通过软件对数据寄存器的读操作清零。

20.4.3.4 空闲帧检测

当一空闲帧被检测到时, UART_STS.IDLEF 置 1。此时如果 UART_CTRL1.IDLEIEN 已置 1, 将产生一个中断。UART_STS.IDLEF 可通过以下软件操作清零: 先读 UART_STS 寄存器, 再读 UART_DAT 寄存器。

20.4.3.5 断开帧检测

当一断开帧被检测到时, 帧错误标志 UART_STS.FEF 被硬件置 1, 可通过以下软件操作清零: 先读 UART_STS 寄存器, 再读 UART_DAT 寄存器。

20.4.3.6 帧错误

如果在预期的时间内没有接收和识别到停止位, 产生一个帧错误, 标志位 UART_STS.FEF 置 1, 同时无效数据将从移位寄存器送到 UART_DAT 寄存器。在单字节通信时, 没有帧错误中断产生, 因为此时 UART_STS.RXDNE 位同时置 1, 后者将产生中断。在多缓冲器通信 (DMA) 情况下, 如果 UART_CTRL3.ERRIEN 已置 1, 将产生一个中断。

20.4.3.7 溢出错误

如果 UART_STS.RXDNE 已被置 1, 而接收移位寄存器又有数据需要送入数据寄存器, 则发生溢出错误, 同时标志位 UART_STS.OREF 硬件置 1。此时数据寄存器中的数据不会丢失, 但移位寄存器中的数据将被覆盖。UART_STS.OREF 可通过以下软件操作清零: 先读 UART_STS 寄存器, 再读 UART_DAT 寄存器。

当产生溢出错误时, 若 UART_STS.RXDNEIEN 已置 1, 将产生一个接收中断。多缓冲器通信模式(DMA)下, 如果 UART_CTRL3.ERRIEN 已置 1, 将产生一个错误中断。

20.4.3.8 噪声错误

当接收器检测到噪声错误时, UART_STS.NEF 被置 1, 可通过以下软件操作清零: 先读 UART_STS 寄存器, 再读 UART_DAT 寄存器。在单字节通信模式下不会产生噪声中断, 因为此时 UART_STS.RXDNE 也被置 1 并产生接收中断。在多缓冲器通信模式(DMA), 如果 UART_CTRL3.ERRIEN 已置 1, 将产生一个错误中断。

表 20-2 噪声检测的数据采样

采样值	NE 状态	接收的位	数据有效性
000	0	0	有效
001	1	0	无效
010	1	0	无效
011	1	1	无效
100	1	0	无效
101	1	1	无效
110	1	1	无效
111	0	1	有效

20.4.4 分数波特率计算

波特率通过 UART_BRCF 寄存器配置, 分频系数由整数部分和小数部分组成, 同时适用于发送器与接收器。在写入 UART_BRCF 之后, 波特率计数器会被波特率寄存器的新值替换。因此, 不要在通信过程中改变波

特率寄存器的数值。

$$\text{TX / RX 波特率} = f_{\text{PCLK}} / (16 * \text{UARTDIV})$$

其中 f_{PCLK} 为 UART 外设时钟：

- PCLK2 用于 UART1/UART2，最高 64MHz
- PCLK1 用于 UART3/UART4/UART5，最高 32MHz.

UARTDIV 为无符号分频系数

20.4.4.1 分频系数 UARTDIV 与 UART_BRCF 寄存器配置

例 1：

如果 DIV_Integer = 27, DIV_Decimal = 12 (UART_BRCF = 0x1BC)，于是

DIV_Integer (UARTDIV) = 27

DIV_Decimal (UARTDIV) = 12/16 = 0.75

所以 UARTDIV = 27.75

例 2：

要求 UARTDIV = 25.62, 就有：

DIV_Decimal = 16 * 0.62 = 9.92

最接近的整数是：10 = 0x0A

DIV_Integer = DIV_Integer (25.620) = 25 = 0x19

于是，UART_BRCF = 0x19A

例 3：

要求 UARTDIV = 50.99 就有：

DIV_Decimal = 16 * 0.99 = 15.84

最接近的整数是：16 = 0x10 => DIV_Decimal[3:0] 溢出 => 进位必须加到小数部分

DIV_Integer = DIV_Integer (0d50.990 + 进位) = 51 = 0x33

于是：UART_BRCF = 0x330

表 20-3 设置波特率时的误差计算

波特率		$f_{\text{pclk}}=32\text{M}$			$f_{\text{pclk}}=36\text{M}$		
序号	Kbps	实际	寄存器设置值	误差%	实际	寄存器设置值	误差%
1	2.4	2.400	833.3125	0%	2.400	937.5	0%
2	9.6	9.601	208.3215	0.01%	9.600	234.375	0%
3	19.2	19.196	104.1875	0.02%	19.200	117.1875	0%

4	57.6	57.554	34.750	0.08%	57.600	39.0625	0%
5	115.2	115.108	17.375	0.08%	115.016	19.5625	0.16%
6	230.4	230.216	8.6875	0.08%	230.769	9.75	0.16%
7	460.8	463.768	4.3125	0.64%	461.538	4.875	0.16%
8	921.6	914.286	2.1875	0.79%	923.077	2.4375	0.16%
9	2250	不可能	不可能	不可能	2250.000	1	0%
10	3000	不可能	不可能	不可能	不可能	不可能	不可能
11	4000	不可能	不可能	不可能	不可能	不可能	不可能
波特率		f _{clk} =48M			f _{clk} =64M		
序号	Kbps	实际	寄存器设置值	误差%	实际	寄存器设置值	误差%
1	2.4	2.400	1250	0%	2.400	1666.6875	0%
2	9.6	9.600	312.5	0%	9.600	416.6875	0%
3	19.2	19.200	156.25	0%	19.202	208.3125	0.01%
4	57.6	57.600	52.0625	0.04%	57.606	69.4375	0.01%
5	115.2	115.016	26.0625	0.08%	115.108	34.750	0.08%
6	230.4	230.769	13	0.16%	230.216	17.375	0.08%
7	460.8	461.538	6.5	0.16%	460.432	8.6875	0.08%
8	921.6	923.077	3.25	0.16%	927.536	4.3125	0.64%
9	2250	2285.714	1.3125	1.59%	2285.714	1.75	1.59%
10	3000	3000.000	1.000	0%	3047.619	1.3125	1.59%
11	4000	不可能	不可能	不可能	4000.000	1.000	0%

注意: CPU 的时钟频率越低, 则某一特定波特率的误差也越低。

20.4.5 UART 接收器容忍时钟的变化

应用中可能会出现发送误差(包括发射端时钟的变化)、接收端波特率误差及振荡器变化、传输线变化(通常由数据上升沿和下降沿时序不一致引起)。这些因素都会影响整个时钟系统的变化。只有当上述四个变化之和小于 UART 接收机的容差时, UART 异步接收机才能正常工作。

正常接收数据时, UART 接收器的容忍度为最大能容忍的变化, 取决于数据位长度的选择, 以及是否使用分数波特率分频系数。

表 20-4 当 DIV_Decimal =0 时, UART 接收器的容忍度

WL 位	认为 NEF 是错误	不认为 NEF 是错误
0	3.75%	4.375%
1	3.41%	3.97%

表 20-5 当 DIV_Decimal !=0 时, UART 接收器的容忍度

WL 位	认为 NEF 是错误	不认为 NEF 是错误
0	3.33%	4.0%
1	3.03%	3.63%

20.4.6 校验控制

通过设置 UART_CTRL1.PCEN 来使能奇偶校验功能。

使能后, 在发送数据时自动生成并发送校验位, 接收数据时对校验位进行检查。

表 20-6 帧格式

WL 位	PCEN 位	UART 帧
0	0	起始位 8 位数据 停止位
0	1	起始位 7 位数据 奇偶校验位 停止位
1	0	起始位 9 位数据 停止位
1	1	起始位 8 位数据 奇偶校验位 停止位

偶校验

UART_CTRL1.PSEL 设置为 0, 使能偶校验

偶校验表示一帧数据(包括校验位)中'1'的个数为偶数。例如: 数据=11000101, 有 4 个'1', 则发送端偶校验位为'0'(总共 4 个'1')。接收端对数据中'1'个数进行确认: 如果是偶数, 校验通过; 如果是奇数, 表示产生了校验错误, UART_STS.PEF 标志位置 1, 此时如果 UART_CTRL1.PE1EN 已置 1, 产生一个中断。

奇校验

UART_CTRL1.PSEL 设置为 1，使能奇校验

奇校验表示一帧数据（包括校验位）中‘1’的个数为奇数。例如：数据=11000101，有 4 个‘1’，则发送端奇校验位为‘1’（总共 5 个‘1’）。接收端对数据中‘1’个数进行确认：如果是奇数，校验通过；如果是偶数，表示产生了校验错误，UART_STS.PEF 标志位置 1，此时如果 UART_CTRL1.PEIE 已置 1，产生一个中断。

20.4.7 DMA 通信

UART 支持 DMA 通信，此时采用多缓冲模式可达到较高的通信效率。

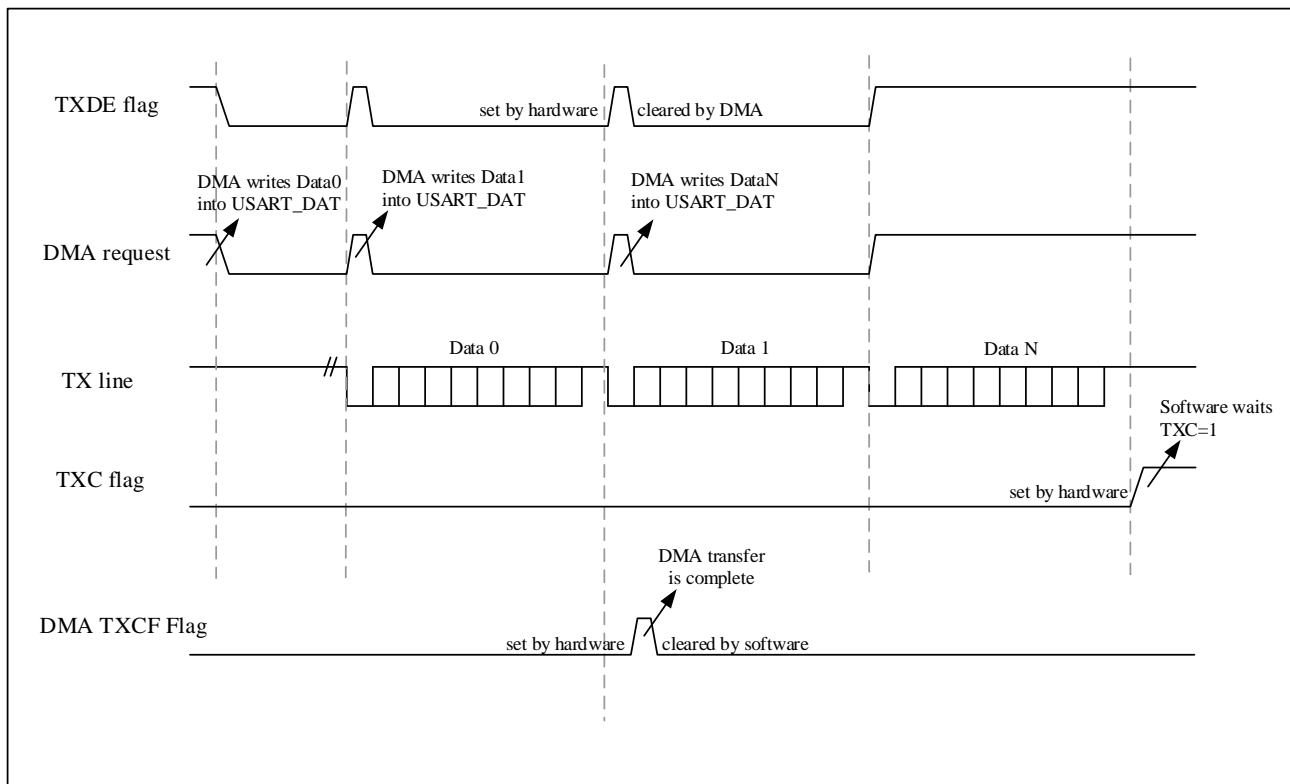
20.4.7.1 DMA 发送

发送器通过将 UART_CTRL3.DMATXEN 置 1 来使能 DMA 发送。当发送移位寄存器为空时 (UART_STS.TXDE=1)，DMA 将数据由 SRAM 送到数据寄存器 UART_DAT。

使用 DMA 发送功能时，按照以下流程对 DMA 进行配置：

1. 设置 DMA 传输的源地址，DMA 传输时从此地址读取要发送的数据
2. 设置 DMA 传输的目的地地址为 UART_DAT 寄存器地址
3. 设置要传输的总的字节数。
4. 设置 DMA 通道优先级、循环模式、地址增加模式、传输数据宽度、中断（传输完成一半还是全部完成时）
5. 激活当前 DMA 通道
6. 传输完成后，标志位 DMA_INTSTS.TXCFx 被置 1

图 20-7 DMA 发送



20.4.7.2 DMA 接收

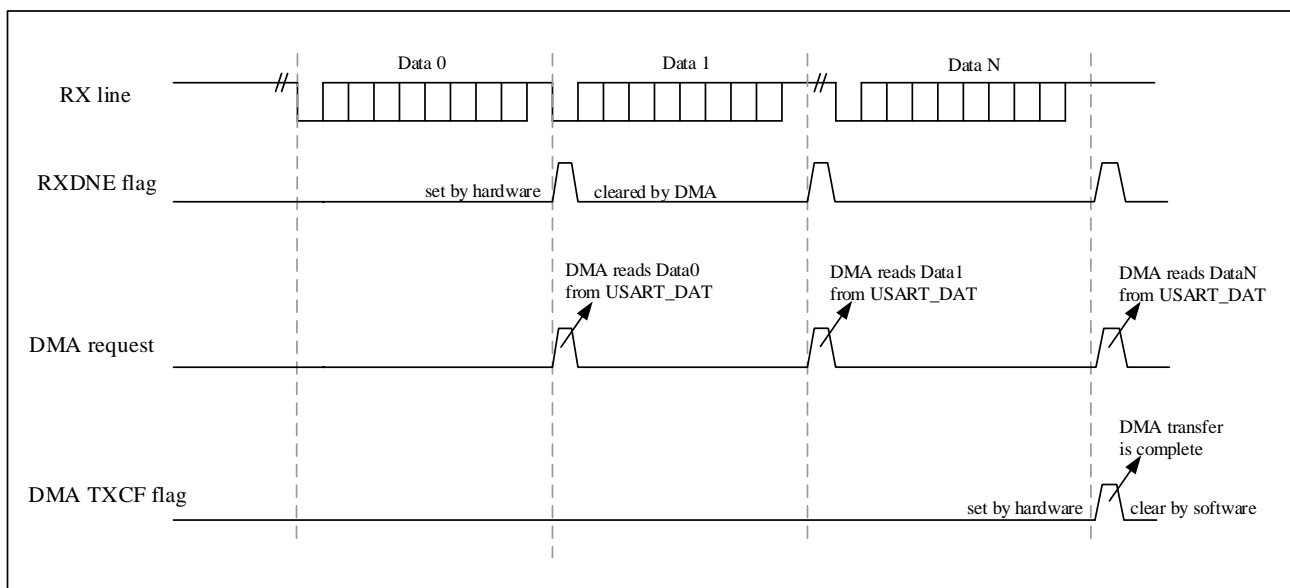
接收器通过将 UART_CTRL3.DMATXEN 置 1 来使能 DMA 接收。当收到 1 字节数据时

(UART_STS.RXDNE=1), DMA 将数据从数据寄存器 UART_DAT 读出数据, 送到 SRAM。

使用 DMA 接收功能时, 按照以下流程对 DMA 进行配置:

1. 设置 DMA 传输的源地址为 UART_DAT 寄存器地址, DMA 传输时从此地址读取要发送的数据
2. 设置 DMA 传输的目的地地址, DMA 传输时将数据送到此地址。
3. 设置要传输的总的字节数。
4. 设置 DMA 通道优先级、循环模式、地址增加模式、传输数据宽度、中断 (传输完成一半还是全部完成时)
5. 激活当前 DMA 通道

图 20-8 DMA 接收



在多缓冲器通信模式, 当检测到帧错误、溢出错误、噪声错误时, 相应标志位置 1。如果此时 UART_CTRL3.ERRIEN 已置 1, 产生一个错误中断。

20.4.8 多处理器通信

UART 支持多处理器通信: 多个设备同时连接到 UART 进行通信, 因此必须判定哪一个设备作为主设备, 其他设备自动做为从设备。主机的 TX 引脚直接连接到其他从设备的 RX 引脚, 所有从设备的 TX 引脚通过逻辑与的方式合并, 再连接到主设备的 RX 引脚。

在多处理器通信模式下, 从设备处于静默模式, 主设备在需要时通过通过指定方式唤醒某一个从设备, 从而从设备可以和主设备进行正常通信。

UART 可通过空闲总线检测或地址标识检测的方式从静默模式唤醒。

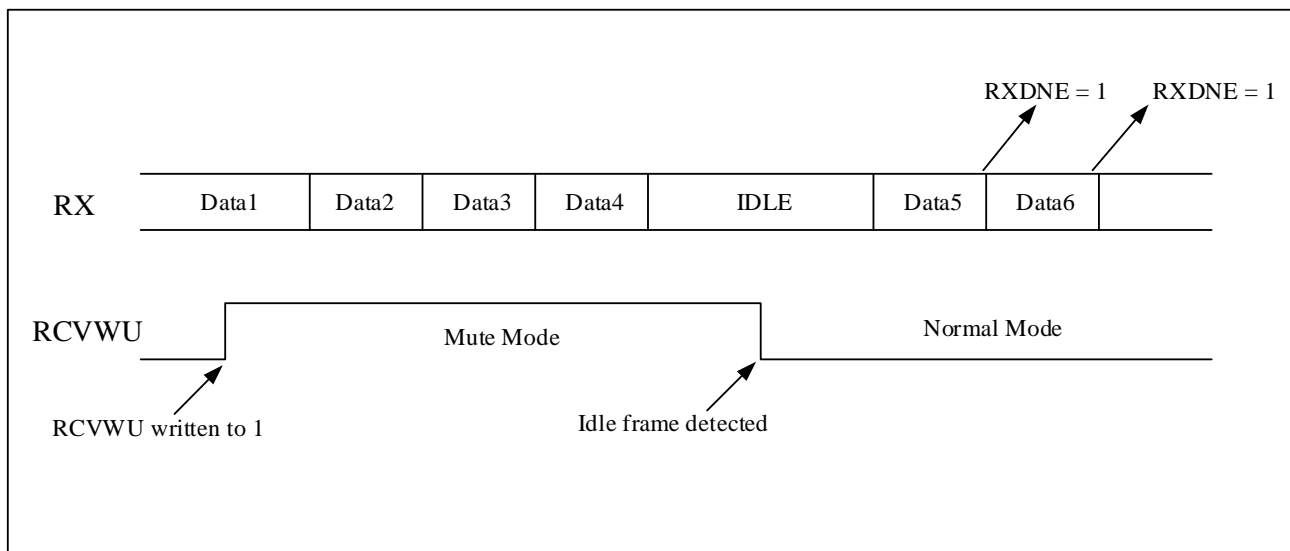
20.4.8.1 空闲总线检测

空闲总线检测流程如下:

1. 清零 UART_CTRL1.WUM 位, UART 启用空闲总线检测功能。
2. 当 UART_CTRL1.RCVWU 已置 1 (可通过硬件自动控制或由在特定条件下由软件配置), UART 进入静默模式, 此时接收状态标志位不会置位, 同时接收中断被禁用。

3. 如图 20-9 所示，当检测到空闲帧时，UART 被唤醒,同时 UART_CTRL1.RCVWU 被硬件清零,此时 UART_STS.IDLEF 标志位不会被置 1。

图 20-9 静默模式下的空闲总线检测



20.4.8.2 地址标识检测

当 UART_CTRL1.WUM 置 1 时，UART 启用地址标识检测功能。标识地址通过 UART_CTRL2.ADDR[3:0] 来配置。如果接收的数据最高有效位（MSB）为 1，当前数据为地址，低 4 位有效；如果 MSB=0，则当前数据为普通数据。

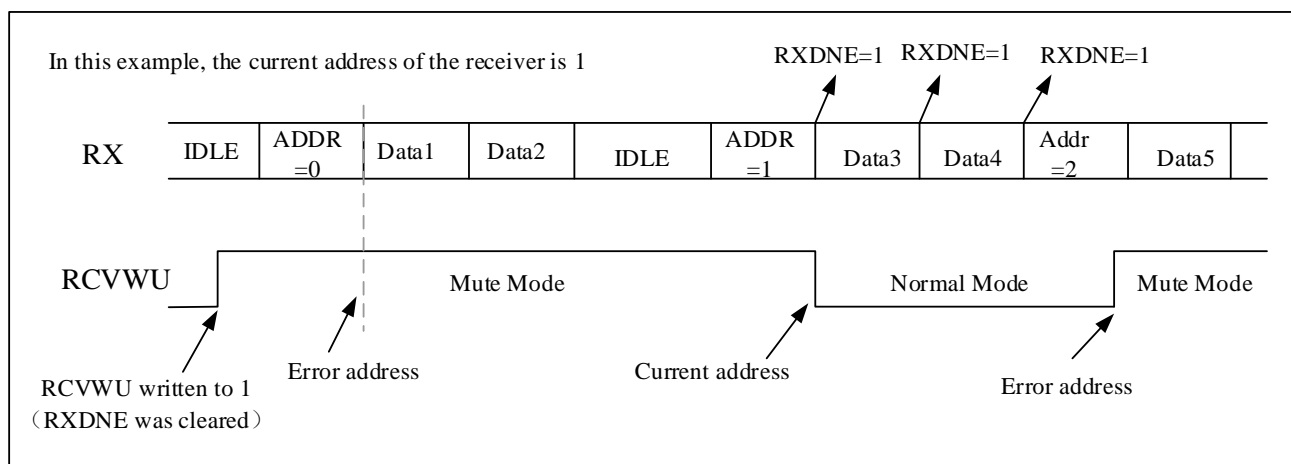
此模式下，UART 可通过以下方式进入静默模式：

- 当接收器没有数据处理时，可通过软件将 UART_CTRL1.RCVWU 置 1，使 UART 进入静默模式。
注意：当接收数据寄存器为空时，(UART_STS.RXNE=0)，UART_CTRL1.RCVWU 位可通过软件写 0 或写 1。否则，对 UART_CTRL1.RCVWU 的写操作被忽略。
- 当接收器收到的地址与预设的地址标识不匹配时，UART_CTRL1.RCVWU 由硬件置 1。

静默模式下，所有接收状态标志位不会置位，同时所有接收中断被禁用。

当接收器收到的地址与预设的地址标识相同时，UART 从静默模式唤醒，UART_CTRL1.RCVWU 被硬件清零，同时 UART_STS.RXDNE 位置 1，此时可进行正常的数据传输。

图 20-10 静默模式下的地址标识检测



20.4.9 单线半双工模式

UART 支持单线半双工通信模式，允许数据双向收发，但同一时间只能单向接收数据或发送数据，数据通信的冲突由软件控制。

通过设置 UART_CTRL3.HDMEN 位来选择单线半双工模式，此时以下控制位必须全部清零：UART_CTRL2.LINMEN、UART_CTRL3.IRDAMEN。

启用单线半双工通信模式后，TX 引脚与 RX 引脚在芯片内部相连，外部 RX 引脚不再使用。当没有数据发送时，TX 引脚被释放。因此，TX 引脚未被 UART 使用时，必须配置为浮空输入或开漏输出高电平。

20.4.10 串行 IrDA 红外编解码模式

UART 支持 IrDA (Infrared Data Association) SIR ENDEC 规范。

通过设置 UART_CTRL3.IRDAMEN 位来选择是否使用 IrDA 模式。当启用 IrDA 模式时，以下配置位必须全部清零：UART_CTRL2.STPB[1:0]、UART_CTRL2.LINMEN、UART_CTRL3.HDMEN。

通过设置 UART_CTRL3.IRDALP 位，可选择 IrDA 的正常工作模式或低功耗模式。

20.4.10.1 IrDA 正常模式

当 UART_CTRL3.IRDALP=0，IrDA 工作在正常模式。

IrDA 是一个半双工通信接口，因此在发送和接收之间最小要有 10ms 的延时。数据采用反相归零(RZI)调制，即采用红外光源脉冲表示逻辑 0。脉冲宽度规定为一个位周期的 3/16，如图 20-12 所示。最大波特率为 115200bps。

UART 将数据送到 SIR 编码器进行调制后输出。调制后的数据流输出给外部红外发送器进行发送。接收时，先通过外部红外接收器接收数据并解调后，发送到 SIR 解码器，解码后再将数据送给 UART。

发送编码器与解码器输入极性相反。空闲时，编码器输出为低电平，而解码器输入为高电平。编码器输出高脉冲表示逻辑 0，输出低电平作为逻辑 1。解码器输入则与之相反。

当 UART 正在发送数据给 IrDA 编码器时，解码器将忽略数据线上的所有数据。当 UART 正在从解码器接收数据时，发送到编码器的数据也被忽略，不进行编码操作。

脉冲宽度可软件配置。IrDA 规范要求脉冲宽度大于 1.41us。如果脉冲宽度小于 2 个周期，数据被过滤而丢失。PSCV 是在 UART_GTP 寄存器配置的预分频值。

20.4.10.2 IrDA 低功耗模式

当 UART_CTRL3.IRDALP=1，IrDA 工作在低功耗模式。

在低功耗模式下发送数据时，脉冲宽度为 3 倍 PSCV 周期。经 PSCV 分频后的时钟频率最小值为 1.42MHz，典型值为 1.8432MHz，(1.42 MHz < 时钟频率 < 2.12 MHz)。

接收数据时，有效低电平信号宽度必须大于 2 个 PSCV 周期。

图 20-11 IrDA SIR ENDEC-框图

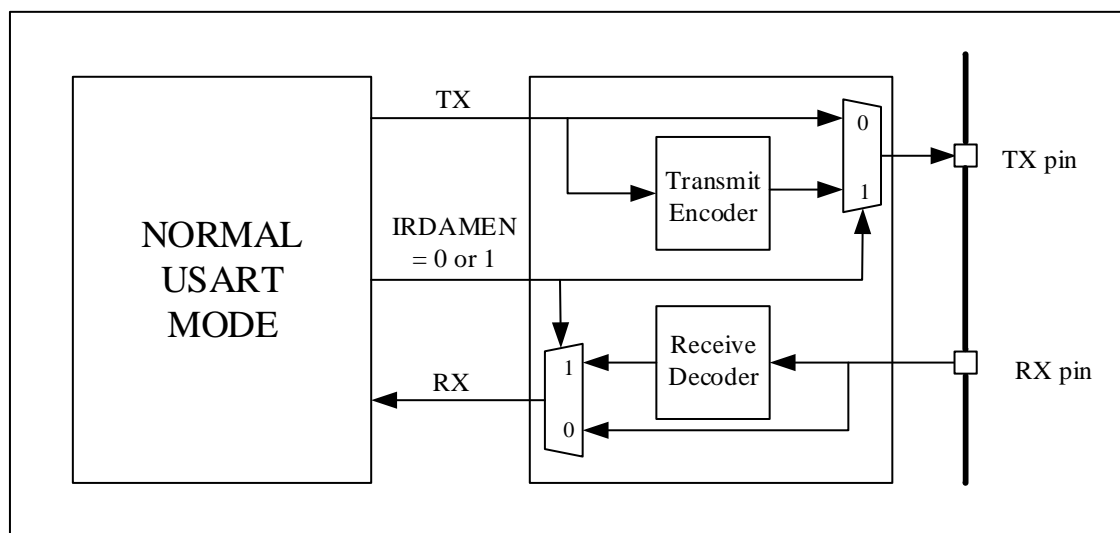
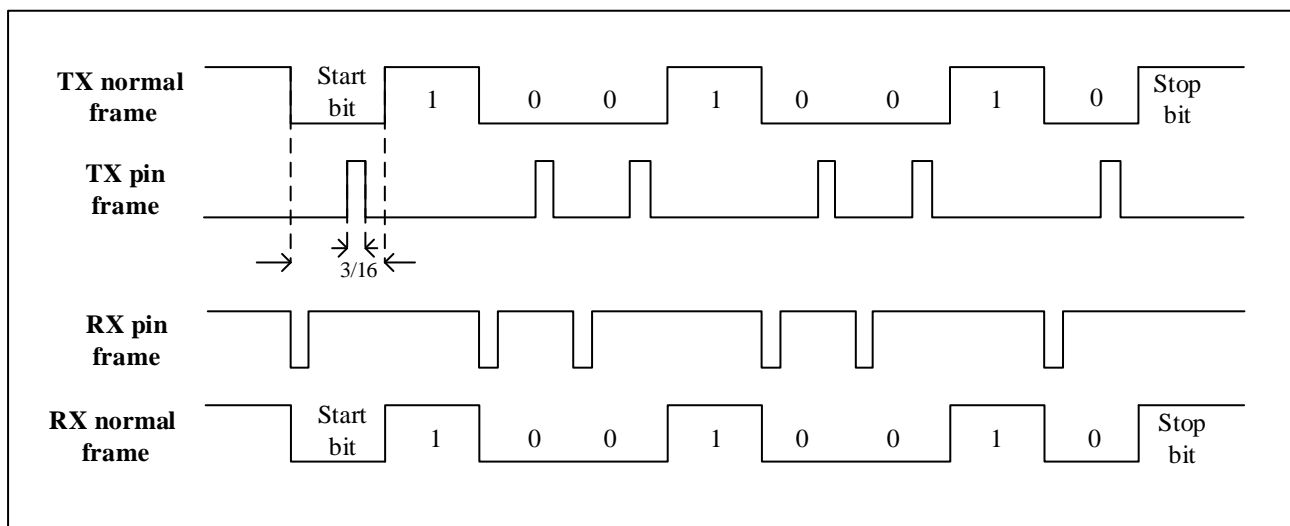


图 20-12 IrDA 数据调制(3/16)-正常模式



20.4.11 LIN 模式

UART 支持 LIN (Local interconnection Network)模式，支持作为主机时发送同步断开帧，也支持作为从机检测断开帧。通过设置 UART_CTRL2.LINMEN 位来使能 LIN 模式。。

注意 当使用 LIN 模式时, 以下配置位必须全部被清零: `UART_CTRL2.STPB[1:0]`、`UART_CTRL3.HDMEN`、`UART_CTRL3.IRDAMEN`。

20.4.11.1 LIN 发送

在 LIN 模式下发送数据时, 数据长度只能配置为 8 位。将 `UART_CTRL1.SDBRK` 置 1 将发送一个 13 位“0”断开帧, 并插入一个停止位。

20.4.11.2 LIN 接收

当总线空闲或数据传输过程中均可检测断开帧。断开帧检测机制独立于 UART 接收器。

通过配置 `UART_CTRL2.LINBDL` 位, 断开帧检测有效低电平位可选择 10 位或 11 位。

当接收器检测到一个起始位, 采样电路在每个位的第 8、9、10 个过采样时钟点进行过采样。如果 10 个或 11 个位都是‘0’, 并且又跟着一个定界符, 表示检测到一个断开帧, `UART_STS.LINBDF` 位置 1。在确认为断开帧前, 必须检测定界符, 意味着 RX 线已经回归空闲状态(高电平)。此时如果 `UART_CTRL2.LINBDIEN` 已置 1, 将产生一个中断。

如果在 10 个或 11 个位前收到了‘1’, 当前断开帧检测被取消, 并重新寻找起始位。

图 20-13 LIN 模式下的断开检测（11 位断开帧长度-设置了 LINBDL 位）

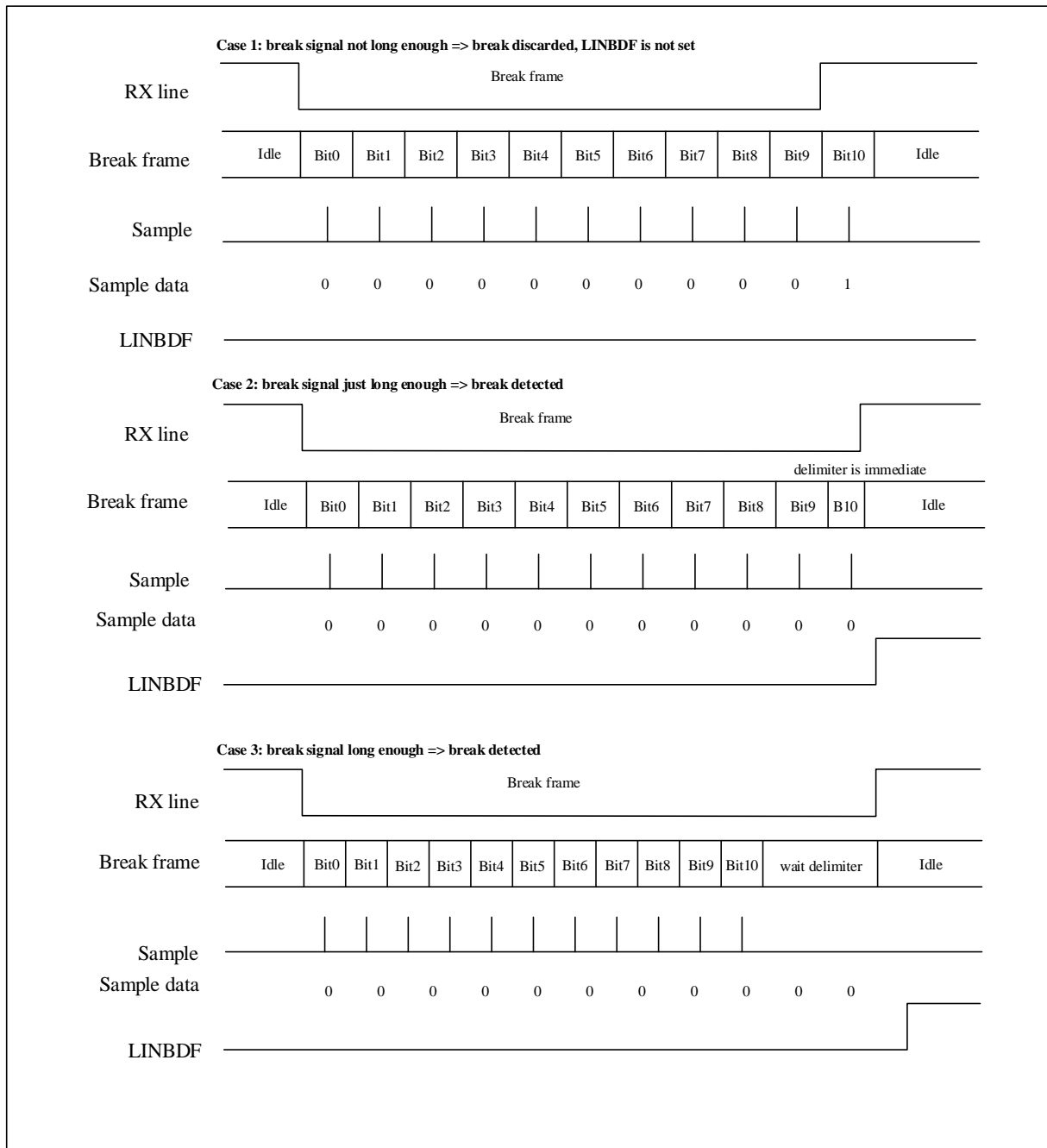
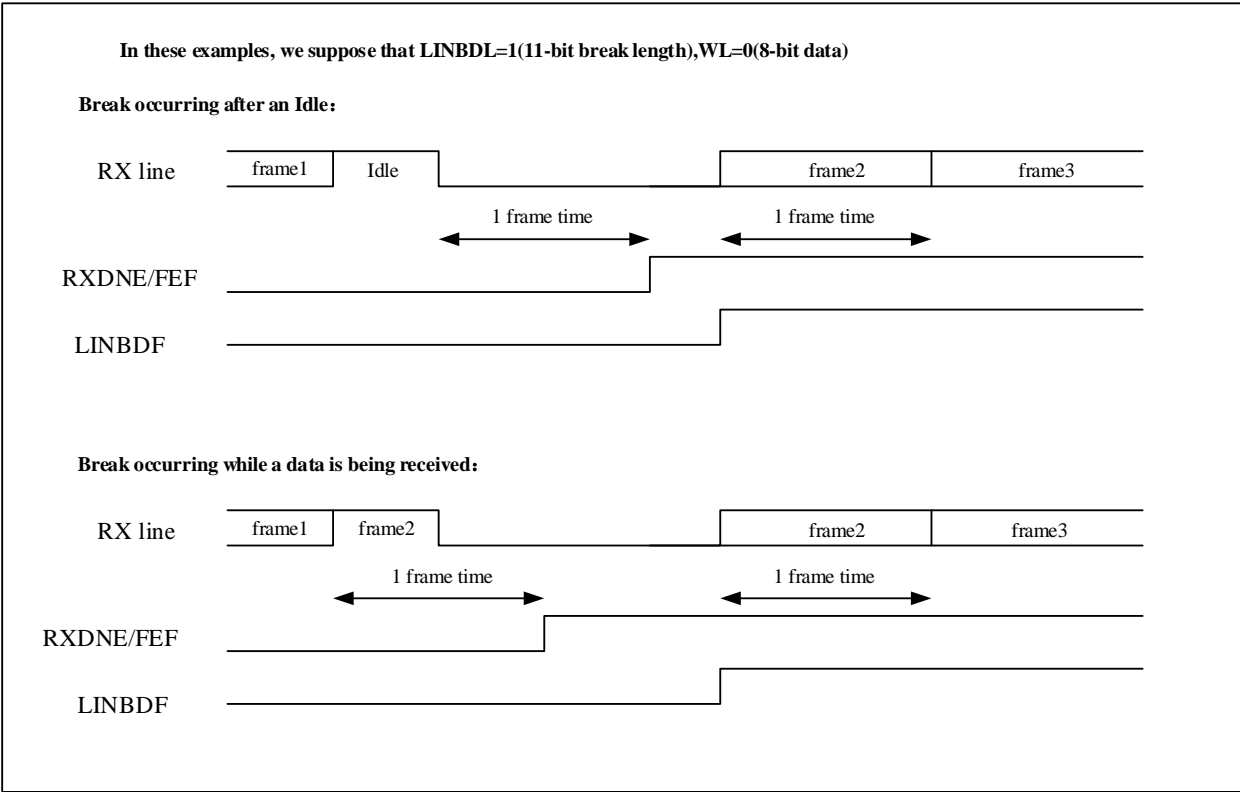


图 20-14 LIN 模式下的断开检测与帧错误的检测



20.5 中断请求

UART 的各种中断事件是逻辑或的关系。如果某个事件对应的中断使能位已置 1，将产生一个相应的中断。但同一个时间只产生一个中断请求。

表 20-7 UART 中断请求

中断函数	中断事件	事件标志	使能
UART 全局中断	发送数据寄存器空	TXDE	TXDEIEN
	发送完成	TXC	TXCIEN
	接收数据就绪可读	RXDNE	RXDNEIEN
	检测到数据溢出	ORERR	
	检测到空闲线路	IDLEF	IDLEIEN
	奇偶检验错	PEF	PEIEN
	断开标志	LINBDF	LINBDIEN
	噪声标志，多缓冲通信（DMA）中的溢出错误和帧错误 ⁽¹⁾	NEF/OREF/FEF	ERRIEN ⁽¹⁾

(1) 仅当使用 DMA 接收数据(UART_CTRL3.DMARXEN=1)时，才使用这个标志位。

20.6 模式配置

表 20-8 UART 模式设置⁽¹⁾

通信模式	UART1	UART2	UART3	UART4	UART5
异步模式	Y	Y	Y	Y	Y
DMA 通讯模式	Y	Y	Y	Y	Y
多处理器	Y	Y	Y	Y	Y
单线半双工模式	Y	Y	Y	Y	Y
IrDA 红外模式	Y	Y	Y	Y	Y
LIN	Y	Y	Y	Y	Y

(1) Y = 支持该模式，N = 不支持该模式

20.7 UART 寄存器

20.7.1 UART 寄存器总览

表 20-9 UART 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000h	UART_CTRL1	Reserved																		SDBRK	PEIEN	TXCIEN	TXDEIEN	RXDNEIEN	IDLEIEN	WUM	RCVWU	WL	PCEN	PSEL	TXEN	RXEN	UEN	
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	UART_CTRL2	Reserved																		LINBDL	LINBDIEN	LINMEN	Reserved					STPB	Reserved	ADDR[3:0]				
	Reset Value																			0	0	0						[1:0]						0
008h	UART_CTRL3	Reserved																								IRDALP	IRDAMEN	ERRIEN	DMARXEN	DMATXEN	HDMEN	Reserved		
	Reset Value																									0	0	0	0	0	0			
00Ch	UART_STS	Reserved																		FEF	NEF	OREF	PEF	LINBDF	Reserved	RXDNE	TXC	TXDE	IDLEF	Reserved				
	Reset Value																			0	0	0	0	0		0	1	1	0					
010h	UART_DAT	Reserved																								DATV[8:0]								
	Reset Value																									0	0	0	0	0	0	0	0	0
014h	UART_BRCF	Reserved																		DIV_Integer[11:0]										DIV_Decimal				
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	UART_GTP	Reserved																								PSCV[7:0]								
	Reset Value																									0	0	0	0	0	0	0	0	0

20.7.2 UART 控制寄存器 1(UART_CTRL1)

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SDBRK	PEIEN	TXCIEN	TXDEIEN	RXDNEIEN	IDLEIEN	WUM	RCVWU	WL	PCEN	PSEL	TXEN	RXEN	UEN	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:14	Reserved	保留，必需保持复位值。
13	SDBRK	发送断开帧（Send break）。 软件通过将该位置 1 发送断开帧。 断开帧传输结束由硬件清 0 该位。 0：没有发送断开帧。 1：发送断开帧。
12	PEIEN	校验错误中断使能（PE interrupt enable）。 如果该位置 1，UART_STS.PEF 被置位时产生中断。 0：校验错误中断禁用。 1：校验错误中断使能。
11	TXCIEN	发送完成中断使能（Transmission complete interrupt enable）。 如果该位置 1，UART_STS.TXC 被置位时产生中断。 0：发送完成中断禁用。 1：发送完成中断使能。
10	TXDEIEN	发送缓冲区空中断使能（TXDE interrupt enable）。 如果该位置 1，UART_STS.TXDE 被置位时产生中断。 0：发送缓冲区空中断禁止。 1：发送缓冲区空中断使能。
9	RXDNEIEN	读数据缓冲区非空中断和过载错误中断使能（RXDNE interrupt enable）。 如果该位置 1，UART_STS.RXDNE 或 UART_STS.OREF 被置位时产生中断。 0：读数据缓冲区非空中断和过载错误中断禁用。 1：读数据缓冲区非空中断和过载错误中断使能。
8	IDLEIEN	IDLE 线检测中断使能（IDLE interrupt enable）。 如果该位置 1，UART_STS.IDLEF 被置位时产生中断。 0：IDLE 线检测中断禁用。 1：IDLE 线检测中断使能。
7	WUM	从静默模式唤醒方法（Wake up mode）。 0：空闲帧唤醒。

位域	名称	描述
		1: 地址标识唤醒。
6	RCVWU	接收器从静默模式中唤醒 (Receiver wakeup) 软件可以通过将该位置 1 使得 UART 进入静默模式, 将该位清 0 唤醒 UART。 空闲帧唤醒模式下 (UART_CTRL1.WUM=0), 当检测到空闲帧时, 该位由硬件清 0。地址标识唤醒模式下 (UART_CTRL1.WUM=1), 当接收到一个地址匹配帧时, 该位由硬件清 0; 或接收到一个地址非匹配帧时, 由硬件置 1。 0: 接收器处于普通工作模式。 1: 接收器处于静默模式。
5	WL	字长 (Word length)。 0: 8 数据位。 1: 9 数据位。 <i>注意: 在数据传输过程中 (发送或者接收时), 不能修改这个位。</i>
4	PCEN	校验控制使能 (Parity control enable)。 0: 校验控制禁用。 1: 校验控制被使能。
3	PSEL	校验模式 (Parity selection)。 0: 偶校验。 1: 奇校验。
2	TXEN	发送器使能 (Transmitter enable)。 0: 发送器禁用。 1: 发送器使能。
1	RXEN	接收器使能 (Receiver enable)。 0: 接收器禁用。 1: 接收器使能。
0	UEN	UART 使能 (UART enable)。 当该位被清零, 在当前字节传输完成后 UART 的分频器和输出停止工作, 以减少功耗。该位由软件设置和清零。 0: UART 禁用。 1: UART 使能。

20.7.3 UART 控制寄存器 2(UART_CTRL2)

偏移地址: 0x04

复位值: 0x0000 0000

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

位域	名称	描述
31:15	Reserved	保留，必需保持复位值。
14	LINBDL	LIN 断开帧检测长度（LIN break detection length）。 该位用来设定在断开帧长度。 0: 10 位 1: 11 位 <i>注意：LINBDL 可用于 LIN 模式及其他模式下的断开帧的检测长度控制，且检测长度和 LIN 模式相同。</i>
13	LINBDIEN	LIN 断开帧检测中断使能（LIN break detection interrupt enable）。 如果该位置 1，当 UART_STS.LINBDF 被置位时将产生中断。 0: 断开信号检测中断禁用 1: 断开信号检测中断使能
12	LINMEN	LIN 模式使能（LIN mode enable） 0: LIN 模式禁用 1: LIN 模式使能
11:7	Reserved	保留，必需保持复位值。
6:5	STPB[1:0]	停止位长（STOP bits）。 00: 1 停止位。 01: 0.5 停止位。 10: 2 停止位。 11: 1.5 停止位。 <i>注：UART 推荐使用 1/2 停止位，0.5/1.5 停止位一般用于智能卡模式</i>
4	Reserved	保留，必需保持复位值。
3:0	ADDR[3:0]	UART 地址。 在多处理器通信下的静默模式中使用的，使用地址标识来唤醒某个 UART 设备。 地址标识唤醒模式下（UART_CTRL1.WUM=1），如果接收到的数据帧低四位与 ADDR[3:0]值不相等，UART 就会进入静默模式；如果接收到的数据帧低四位与 ADDR[3:0]值相等，UART 就会被唤醒。

20.7.4 UART 控制寄存器 3(UART_CTRL3)

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						IRDALP	IRDAMEN	ERRIEN	DMA RXEN	DMA TXEN	HDMEN	Reserved			
						rw	rw	rw	rw	rw	rw				

位域	名称	描述
31:9	Reserved	保留，必需保持复位值。
8	IRDALP	IrDA 低功耗模式（IrDA low-power）。 该位用于为 IrDA 模式选择低功耗模式。 0：正常模式。 1：低功耗模式。
7	IRDAMEN	IrDA 模式使能（IrDA mode enable）。 0：IrDA 禁用。 1：IrDA 使能。
6	ERRIEN	错误中断使能（Error interrupt enable）。 当 DMA 接收模式（UART_CTRL3.DMARXEN=1）使能时，如果该位被置 1， UART_STS.FEF、UART_STS.OREF、UART_STS.NEF 被置位将产生中断。 0：错误中断禁用。 1：错误中断使能。
5	DMARXEN	DMA 接收使能（DMA receiver enable）。 0：DMA 接收模式禁用。 1：DMA 接收模式使能。
4	DMATXEN	DMA 发送使能（DMA transmitter enable）。 0：DMA 发送模式禁用。 1：DMA 发送模式使能。
3	HDMEN	半双工模式使能（Half-duplex mode enable）。 该位用于使能半双工模式。 0：半双工模式禁用。 1：半双工模式使能。
2:0	Reserved	保留，必需保持复位值。

20.7.5 UART 状态寄存器 (UART_STS)

偏移地址：0x0C

复位值：0x0000 0180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEF	NEF	OREF	PEF	LINBDF	Reserved	RXDNE	TXC	TXDE	IDLEF	Reserved					
r	r	r	r	rc_w0		rc_w0	rc_w0	r	r						

位域	名称	描述
31:16	Reserved	保留，必需保持复位值。

位域	名称	描述
15	FEF	<p>帧错误（Framing error）。</p> <p>当检测到同步错位、过多的噪声或者检测到断开符（即没有检测到预期的停止位），该位被硬件置位。由软件序列将其清零（先读 UART_STS，再读 UART_DAT）。</p> <p>0：未检测到帧错误。</p> <p>1：检测到帧错误或者断开帧（break frame）。</p> <p><i>注意：该位不会产生中断，因为它和 UART_STS.RXDNE 一起出现，硬件会在设置 UART_STS.RXDNE 标志时产生中断。如果当前传输的数据既产生了帧错误，又产生了过载错误，硬件还是会继续该数据的传输，并且只设置 OREF 标志位。</i></p> <p><i>在多缓冲区通信模式（DMA）下，如果设置了 UART_CTRL3.ERRIEN 位，则设置 FEF 标志时会产生中断。</i></p>
14	NEF	<p>噪声错误标志（Noise error flag）。</p> <p>在接收到的帧检测到噪音时，由硬件对该位置位。由软件序列对其清零（先读 UART_STS，再读 UART_DAT）。</p> <p>0：没检测到噪声错误。</p> <p>1：检测到噪声错误。</p> <p><i>注意：该位不会产生中断，因为它和 UART_STS.RXDNE 一起出现，硬件会在设置 UART_STS.RXDNE 标志时产生中断。在多缓冲区通信模式(DMA)下，如果设置了 UART_CTRL3.ERRIEN 位，则设置 NEF 标志时会产生中断。</i></p>
13	OREF	<p>溢出错误（Overrun error）。</p> <p>在 RXDNE 置位的情况下，如果 UART_DAT 寄存器接收到来自移位寄存器的数据，该位置 1。当寄存器 UART_CTRL3.RXDNEIEN 位被置位，将会有中断产生。</p> <p>软件先读 UART_STS，再读 UART_DAT 可清除该位。</p> <p>0：没有检测到溢出错误。</p> <p>1：检测到溢出错误。</p> <p><i>注意：在多缓冲区通信模式(DMA)下，如果设置了 UART_CTRL3.ERRIEN 位，则设置 OREF 标志时会产生中断。</i></p>
12	PEF	<p>校验错误（Parity error）。</p> <p>当接收到的数据帧校验位与预期校验值不同时，该位置位。</p> <p>软件先读 UART_STS，再读 UART_DAT 可清除该位。</p> <p>0：没检测到校验错误。</p> <p>1：检测到校验错误。</p>
11	LINBDF	<p>LIN 断开检测标志（LIN break detection flag）。</p> <p>如果设置了 UART_CTRL2.LINMEN 位，当检测到 LIN 断开，该位由硬件置位。如果 UART_CTRL2.LINBDIEN 被置位时，将产生中断。</p> <p>该位由软件清 0。</p> <p>0：没有检测到 LIN 断开字符。</p> <p>1：检测到 LIN 断开字符。</p>
10	Reserved	保留，必需保持复位值。
9	RXDNE	<p>读数据缓冲区非空（Read data register not empty）。</p> <p>当读数据缓冲区接收到来自移位寄存器的数据时，该位置 1。当寄存器</p>

位域	名称	描述
		<p>UART_CTRL1.RXDNEIEN 位被置位，将会有中断产生。</p> <p>软件可以通过对该位写 0 或读 UART_DAT 寄存器来将该位清 0。</p> <p>0：读数据缓冲区为空。</p> <p>1：读数据缓冲区不为空。</p>
8	TXC	<p>发送完成（Transmission complete）。</p> <p>上电复位后，该位被置 1。如果 UART_STS.TXDE 置位，在当前数据发送完成时该位置 1。</p> <p>UART_CTRL1.TXCIEN 被置位将产生中断。</p> <p>该位由软件清 0。</p> <p>0：发送没有完成。</p> <p>1：发送完成。</p>
7	TXDE	<p>发送数据缓冲区空（Transmit data register empty）。</p> <p>上电复位或待发送数据已发送至移位寄存器后，该位置 1。</p> <p>UART_CTRL1.TXDEIEN 被置位将产生中断。</p> <p>该位在软件将待发送数据写入 UART_DAT 时被清 0。</p> <p>0：发送数据缓冲区不为空。</p> <p>1：发送数据缓冲区空。</p>
6	IDLEF	<p>空闲线检测标志（IDLE line detected）。</p> <p>在一个帧时间内，在 RX 引脚检测到空闲状态，该位置 1。当寄存器 UART_CTRL1.IDLEIEN 位被置位，将会有中断产生。</p> <p>软件先读 UART_STS，再读 UART_DAT 可清除该位。</p> <p>0：未检测到空闲帧。</p> <p>1：检测到空闲帧。</p> <p><i>注意：IDLEF 位不会再次被置高直到 RXDNE 位被置起（即又检测到一次空闲总线）。</i></p>
5:0	Reserved	保留，必需保持复位值。

20.7.6 UART 数据寄存器(UART_DAT)

偏移地址：0x10

复位值：未定义（不确定值）

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

位域	名称	描述
31:9	Reserved	保留，必需保持复位值。

位域	名称	描述
		Reserved, must be kept at reset value.
8:0	DATV[8:0]	数据值 (Data value) 包含了发送或接收的数据；软件可以通过写这些位来改变发送数据，或读这些位的值来获取接收数据。 如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位（第 7 位或第 8 位取决于 UART_CTRL1.WL 位）将被校验位取代。

20.7.7 UART 波特率配置寄存器 (UART_BRCF)

偏移地址： 0x14

复位值： 0x0000 0000

注意：UART_CTRL1.UEN=1 时，不能写该寄存器；如果 UART_CTRL1.TXNE 或 UART_CTRL1.RXNE 被分别禁止，波特计数器停止计数。

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

位域	名称	描述
31:16	Reserved	保留，必需保持复位值。
15:4	DIV_Integer [11:0]	波特率分频器的整数部分。
3:0	DIV_Decimal[3:0]	波特率分频器的小数部分。

20.7.8 UART 保护时间和预分频寄存器(UART_GTP)

偏移地址： 0x18

复位值： 0x0000 0000

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

rw

位域	名称	描述
31:8	Reserved	保留，必需保持复位值。 Reserved, must be kept at reset value.
7:0	PSCV[7:0]	预分频器值（Prescaler value）。 在 IrDA 低功耗模式下，这些位用来设定将外设时钟（PCLK1/PCLK2）分频产生低功耗频率的分频系数。 00000000：保留 – 不要写入该值 00000001：对源时钟 1 分频 ... 11111111：对源时钟 255 分频 在 IrDA 正常模式下，PSCV 只能设置成 00000001。 在智能卡模式下，PSCV[4:0]用于设定外设时钟（APB1/APB2）生成智能卡时钟的分频 系数。实际的分频系数为 PSCV[4:0]设定值的两倍。 00000：保留 – 不要写入该值 00001：对源时钟 2 分频 00010：对源时钟 4 分频 ... 11111：对源时钟 62 分频

21 控制器局域网(CAN)

21.1 CAN 简介

作为 CAN 网络接口，基本扩展 CAN 支持 CAN 协议 2.0A 和 2.0B。它可以高效地处理大量接收到的消息，大大降低 CPU 资源的消耗。报文发送的优先级特性可以通过软件进行配置，CAN 的硬件功能可以支持时间触发的通信方式，适用于一些对安全性要求较高的应用。

21.2 CAN 的主要特性

- 波特率最高支持 1Mbit/s
- 支持 CAN 协议 2.0A/B。
- 支持时间触发通讯功能
- 支持独立的中断控制。
- CAN Core 管理 CAN 和 512 字节 SRAM 存储器之间的通信（参见图 21-3）

时间触发通信模式

- 16 位自由运行定时器
- 禁止自动重传模式。
- 可配置最后 2 个数据字节为发送时间戳。

发送

- 有三个发送邮箱。
- 记录发送 SOF 时间的时间戳功能
- 软件可以配置发送消息的优先级特性。

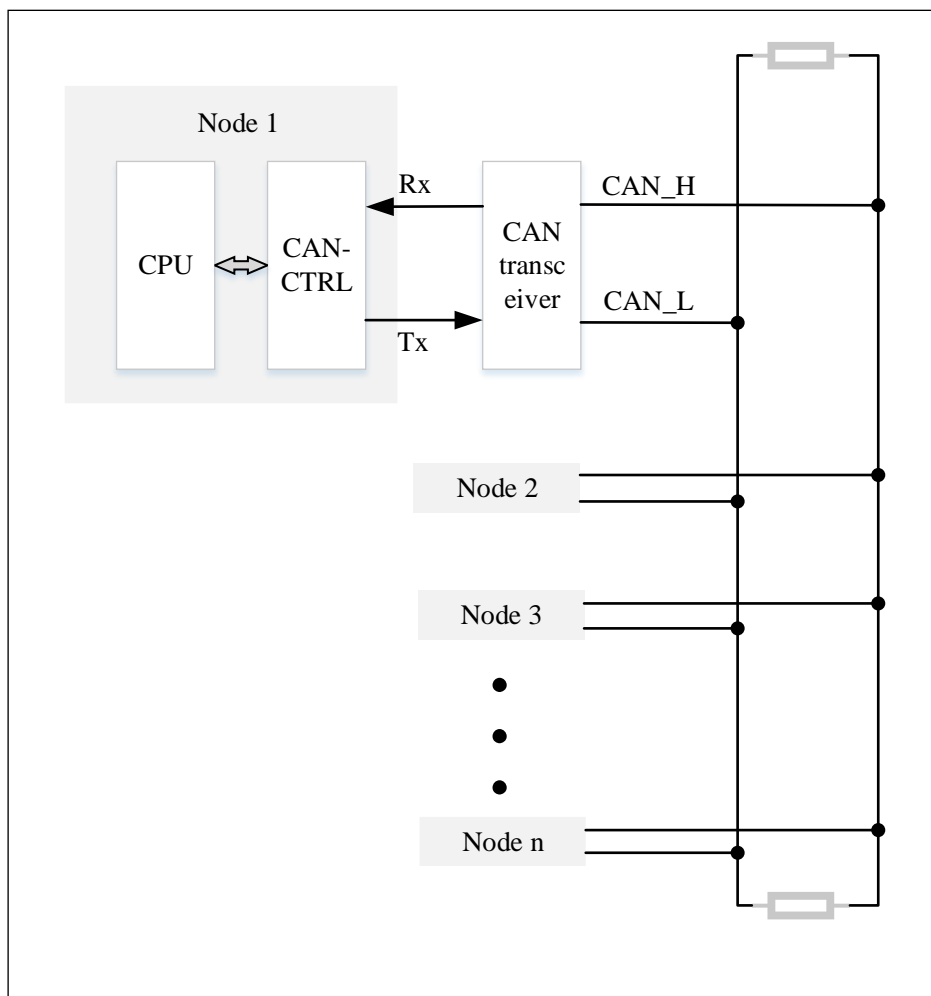
接收

- 过滤器组支持标识符列表模式
- 有两个接收 FIFO，每个深度为 3 级
- 共有 14 个过滤器组
- 可配置的 FIFO 溢出处理方法
- 记录接收 SOF 时间的时间戳功能

21.3 CAN 整体介绍

随着 CAN 的广泛应用，CAN 网络的节点数量迅速增长，多个 CAN 节点通过 CAN 网络连接。随着 CAN 节点数量的增加，CAN 网络中的报文也急剧增加，会占用大量的 CPU 资源。在这个 CAN 控制器中，增加了接收 FIFO 和过滤机制作为 CPU 消息处理的硬件支持，降低了 CAN 消息的实时响应要求。

图 21-1 CAN 网络拓扑



21.3.1 CAN 模块

CAN 模块可自动收发 CAN 报文，支持标准标识符（11 位）和扩展标识符（29 位）。

21.3.2 CAN 工作模式

初始化、正常和睡眠模式是 CAN 的三种主要工作模式。CANTX 引脚的内部上拉电阻在硬件复位后被激活，CAN 工作在睡眠模式以降低功耗。

软件可以设置 CAN_MCTRL.INIRQ 和 CAN_MCTRL.SLPRQ 位来配置 CAN 进入初始化或睡眠模式。软件读 CAN_MSTS.INIAK 或 CAN_MSTS.SLPAK 位的值来确认是否进入初始化或睡眠模式，此时 CANTX 引脚的内部上拉电阻被禁用。

当 CAN_MSTS.INIAK 和 CAN_MSTS.SLPAK 位都为“0”时，CAN 处于正常模式，它必须与 CAN 总线同步才能进入正常模式。当在 CANRX 引脚上监测到 11 个连续的隐性位时，CAN 总线处于空闲状态，同步完成。

21.3.2.1 正常模式

初始化完成后，软件配置 CAN 进入正常模式。清除 CAN_MCTRL.INIRQ 位并等待硬件清除 CAN_MSTS.INIRQ 位以确认 CAN 进入正常模式。只有与 CAN 总线同步完成后，CAN 才能正常收发报文。

过滤器组的位宽和模式的设置必须在过滤器处于初始化模式（CAN_FMC.FINITM 位为 1）时完成。过滤器初始值的设置必须在未激活时完成（对应的 CAN_FA1.FAC 位为 0）。

21.3.2.2 初始化模式

软件只有在 CAN 处于初始化模式时才能进行初始化配置。设置 CAN_MCTRL.INIRQ 位并清除 CAN_MCTRL.SLPRQ 位，并等待硬件设置 CAN_MSTS.INIAK 位以确认 CAN 进入初始化模式。进入初始化模式时，寄存器配置不受影响。CAN 处于初始化模式时，报文收发被禁止，CANTX 引脚输出一个隐性位（高电平）。要退出初始化模式，清除 CAN_MCTRL.INIRQ 位，并等待硬件清除 CAN_MSTS.INIAK 位以确认 CAN 退出初始化模式。

通过软件对 CAN 进行初始化配置，至少需要配置位时间特性寄存器（CAN_BTIM）和控制寄存器（CAN_MCTRL）。软件需要设置 CAN_FMC.FINITM 位来初始化配置 CAN 的过滤器组（模式、位宽、FIFO 关联、激活和过滤器值）。配置 CAN 的过滤器组不一定需要处于初始化模式。

需要注意的是当 CAN_FMC.FINITM=1 时，禁止接收报文。如果要修改对应过滤器的值，需要先清除过滤器激活位（在 CAN_FA1 中）。有必要保持未使用的过滤器组处于非活动状态（将其 CAN_FA1.FAC 位保持为“0”）。

21.3.2.3 睡眠模式（低功耗）

要进入睡眠模式，需要设置 CAN_MCTRL.SLPRQ 位，并等待硬件设置 CAN_MSTS.SLPAK 位以确认 CAN 进入睡眠模式。在未使用时 CAN 可以配置为睡眠模式以降低功耗。在睡眠模式下，CAN 的时钟停止工作，但软件仍然可以访问发送/接收邮箱寄存器。当 CAN 处于睡眠模式时，CAN_MCTRL.INIRQ 位必须置 1，并且 CAN_MCTRL.SLPRQ 位必须同时清零才能进入初始化模式。

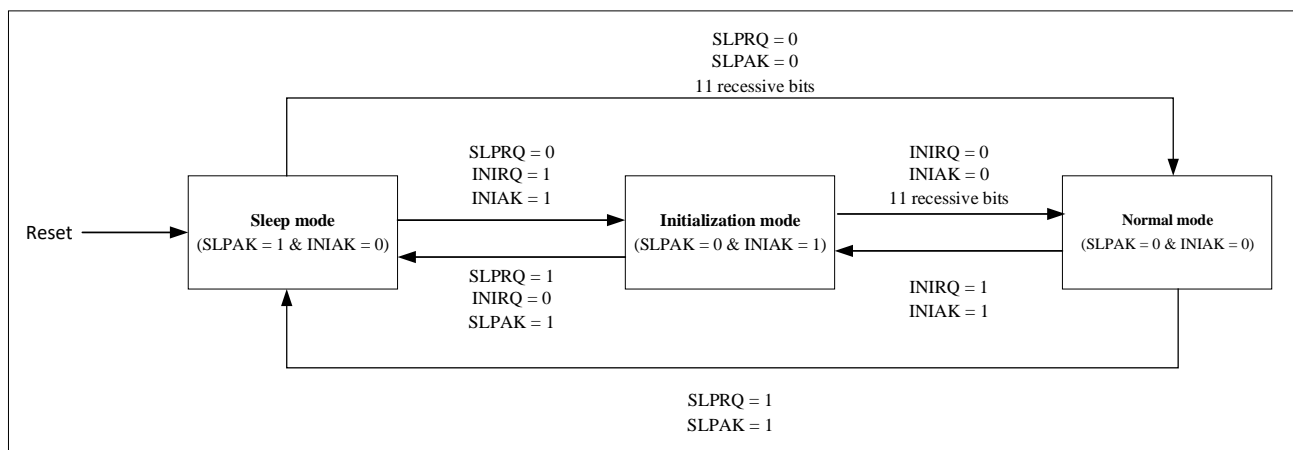
唤醒 CAN 有两种方式（CAN 退出睡眠模式）：

- 当 CAN_MCTRL.AWKUM 位被置位（使能硬件自动唤醒）时，一旦检测到 CAN 总线的活动，硬件将自动清除 CAN_MSTS.SLPRQ 位以唤醒 CAN。
- 当 CAN_MCTRL.AWKUM 位清零（使能软件唤醒），并且唤醒中断发生时，软件必须清零 CAN_MCTRL.SLPRQ 位才能退出睡眠状态。

如果唤醒中断（设置 CAN_INTE.WKUIE 位）使能，一旦检测到 CAN 总线活动，将产生唤醒中断，无论硬件是否使能自动唤醒 CAN。

在进入正常模式之前，CAN 必须与 CAN 总线同步等到 CAN_MSTS.SLPAK 位清零以确认已退出睡眠模式。请参考图 21-2。

图 21-2 CAN 工作模式



注：硬件设置 CAN_MSTS.INIAK 或 CAN_MSTS.SLPAK 位以响应睡眠或初始化请求的状态。

21.3.3 发送邮箱

应用程序可以通过三个发送邮箱发送消息，发送三个邮箱消息的顺序是由发送调度器根据消息的优先级决定的，优先级可以由消息的标识符决定，也可以由发送请求的顺序决定。

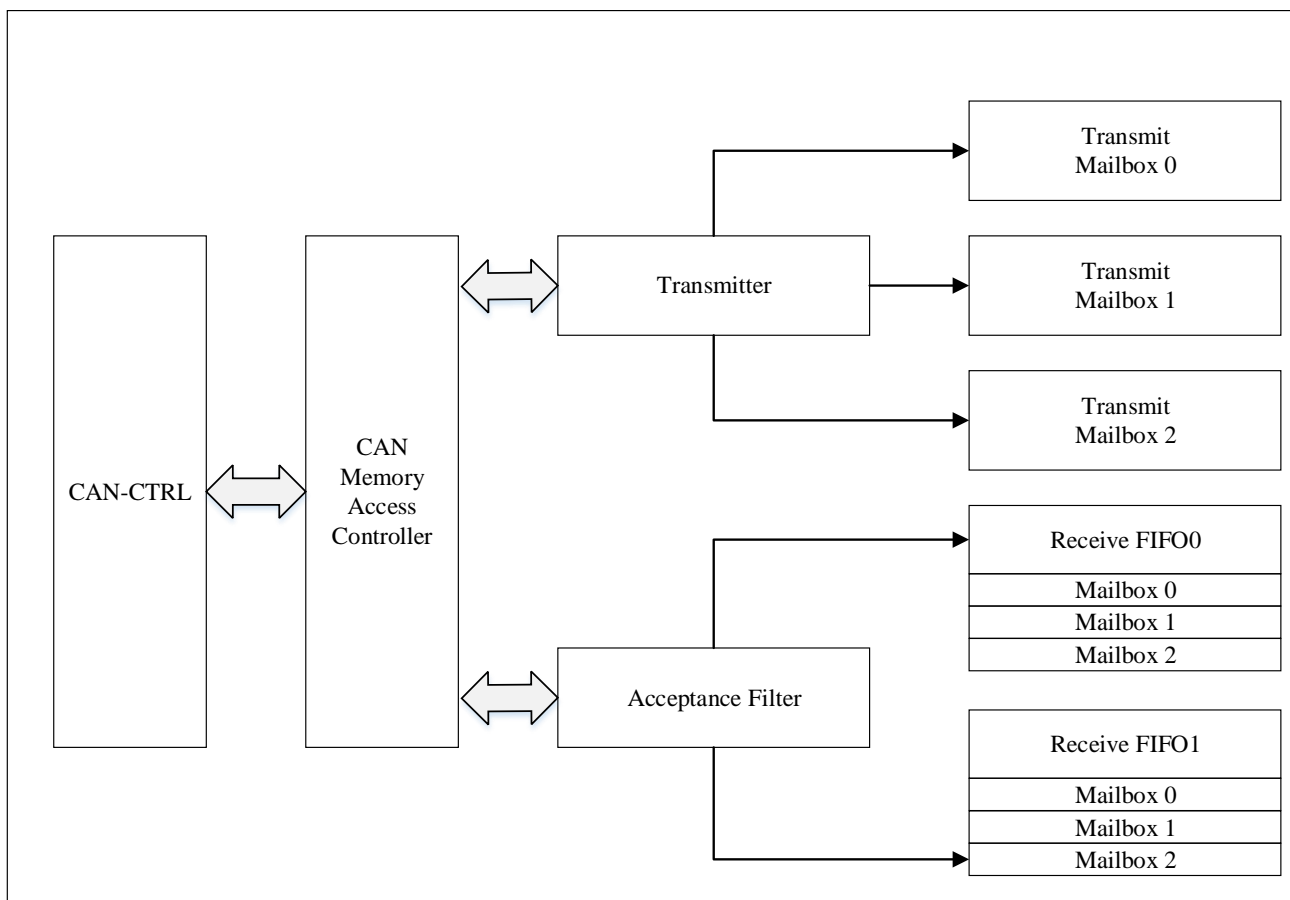
21.3.4 接收过滤器

CAN 有 14 个可配置的标识符过滤器组。应用配置标识符过滤组后，接收邮箱会自动接收需要的邮件，丢弃其他邮件。

21.3.5 接收 FIFO

CAN 有两个接收 FIFO，每个 FIFO 可以存储三个完整的报文。无需应用程序管理，由硬件管理。

图 21-3 单 CAN 框图



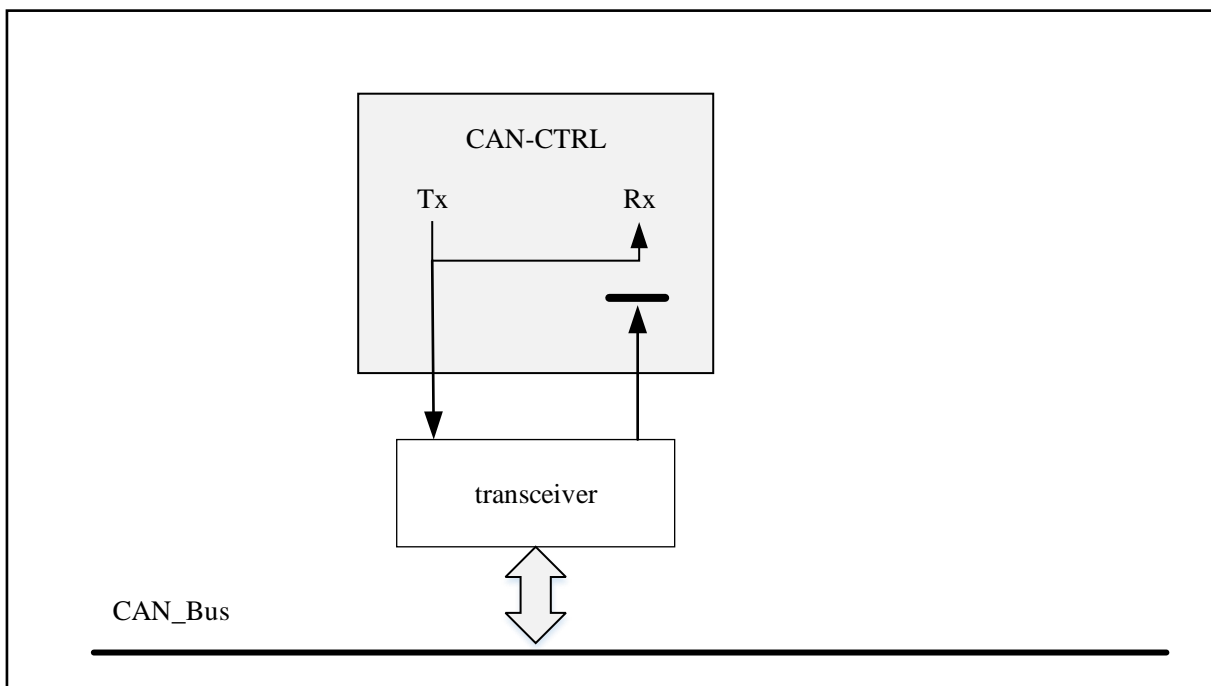
21.3.6 CAN 测试模式

在初始化模式下，必须通过组合 **CAN_BTIM.SLM** 位和 **CAN_BTIM.LBM** 位来选择测试模式。选择测试模式后，软件需要清除 **CAN_MCTRL.INIRQ** 位退出初始化模式，进入测试模式。

21.3.6.1 回环模式

回环模式可用于自检。在回环模式下，CAN 将发送的消息作为接收消息保存在接收邮箱中（如果可以通过接收过滤）。在回环模式下，CAN 在内部将 Tx 输出反馈到 Rx 输入，完全忽略 **CANRX** 引脚的实际状态。可以在 **CANTX** 引脚上检测到发送的消息。为了避免外部影响，CAN 内核忽略了确认错误（在数据/远程帧确认位的时刻，不检测是否有显性位）。要进入回环模式，应清除 **CAN_BTIM.SLM** 位并设置 **CAN_BTIM.LBM** 位。

图 21-4 回环模式

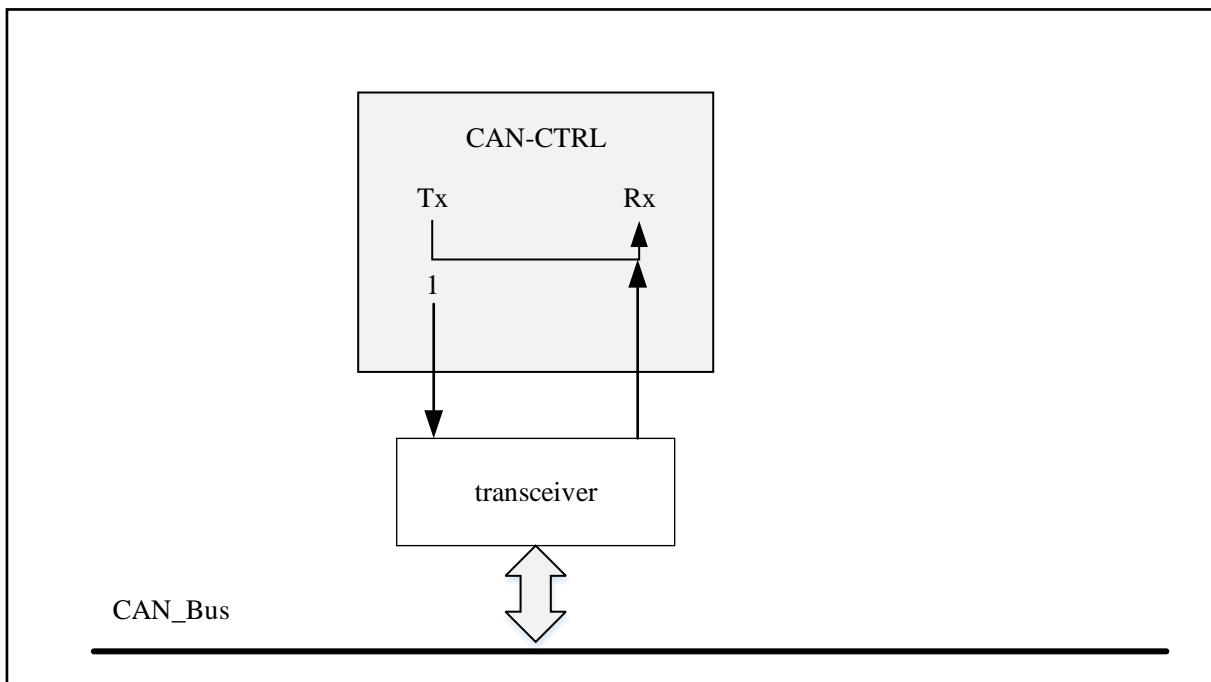


21.3.6.2 静默模式

在静默模式下，CAN 可以正常接收数据帧和远程帧，但不能真正发送消息。如果 CAN 需要发送过载标志、主动错误标志或 ACK 位（这些是显性位），这些显性位在内部连接回来以便被 CAN 内核检测到。同时，CAN 总线不会受到影响，仍然保持在隐性位状态。因此，静默模式通常用于分析 CAN 总线的活动，而不影响总线，因为显性位实际上并未发送到总线。

要进入静默模式，应设置 CAN_BTIM.SLM 位并清除 CAN_BTIM.LBM 位。

图 21-5 静默模式

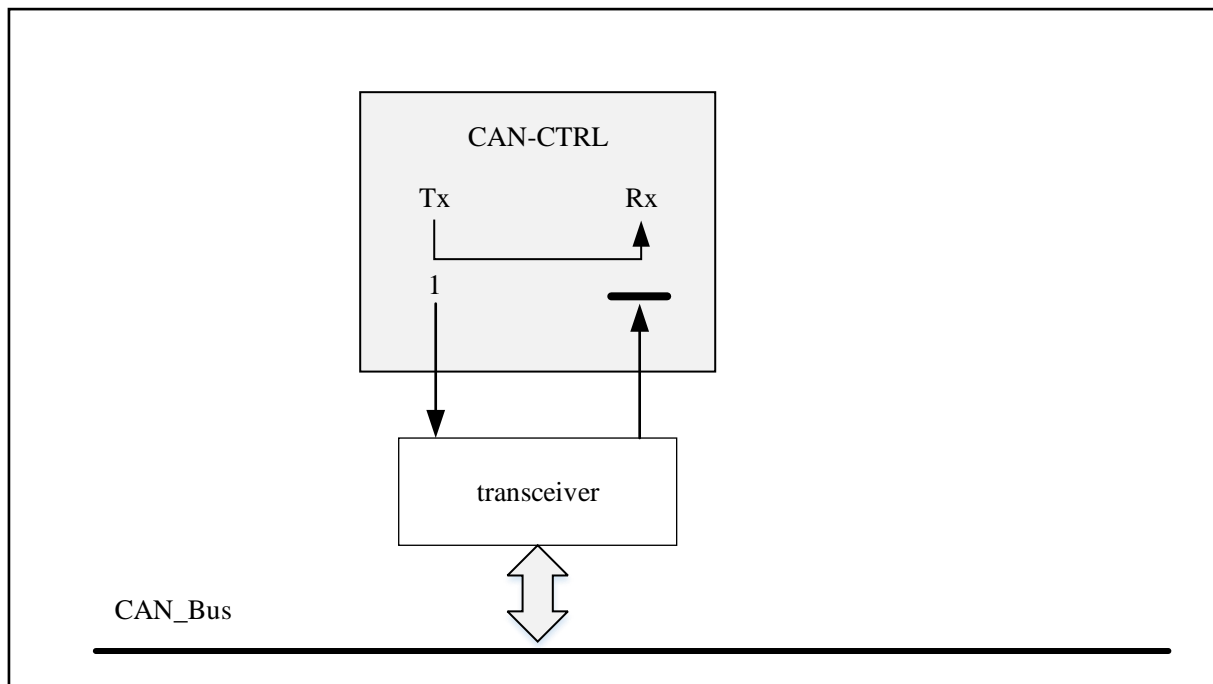


21.3.6.3 回环静默模式

在回环静默模式下，CANRX 引脚与 CAN 总线断开，而 CANTX 引脚被驱动为隐性位状态。它可以用于“热自检”，就像可以在回环模式下测试 CAN 一样，但不影响 CANTX 和 CANRX 连接的整个 CAN 系统。

要进入回环静默模式，应设置 CAN_BTIM.SLM 位和 CAN_BTIM.LBM 位。

图 21-6 回环静默模式



21.3.7 CAN 调试模式

CAN 可以根据以下配置位的状态继续正常工作或停止工作：

- 调试支持(DBG)模块中 CAN 的 DBG_CTRL.CAN_STOP 位。参见第 3.3.2 节：外设调试支持。
- CAN_MCTRL.DBGF 位见 21.7.3.1 节:CAN_MCTRL。

当微控制器处于调试模式时，Cortex-M0 内核处于挂起状态。

21.4 CAN 功能描述

21.4.1 发送处理

发送消息的流程如下：

- 应用程序选择一个空的发送邮箱；
- 将标识符、数据长度和要发送的数据写入发送邮箱寄存器；
- 设置 CAN_TMIx.TXRQ 位请求发送（设置 CAN_TMIx.TXRQ 后邮箱不再是空邮箱，软件无权写入邮箱寄存器）；
- 邮箱进入 pending 状态，等待成为最高优先级，见 21.4.1.1 发送优先级；

- 邮箱成为最高优先级邮箱后，转为就绪状态；
- CAN 总线一进入空闲状态就发送就绪邮箱中的消息，然后进入发送状态。
- 邮箱中的消息发送成功后变为空邮箱；
- 硬件设置 CAN_TSTS 寄存器中对应邮箱的 RQCPM 和 TXOKM 位，表示发送成功。

需要注意的是，如果发送失败，CAN_TSTS.ALSTM 位被设置表示失败是由仲裁引起的，或者 CAN_TSTS.TERRM 位被设置表示是传输错误引起的（具体错误请查看错误状态寄存器 CAN_ESTS 的 LEC[2:0]错误代码位）。

21.4.1.1 发送优先级

由发送请求的顺序决定：

设置 CAN_MCTRL.TXFP 位，发送邮箱可以配置为发送 FIFO。此时，发送的优先级由发送请求的顺序决定。这种模式对于分段传输很有用。

由标识符决定：

根据 CAN 协议，具有最低标识符的消息具有最高优先级。如果标识符的值相等，则先发送邮箱号小的消息。当注册多个发送邮箱时，发送顺序由邮箱中邮件的标识符决定。

21.4.1.2 取消发送

设置 CAN_TSTS.ABRQM 位可以中止发送请求。如果邮箱就绪或挂起，发送请求将立即中止。如果邮箱处于发送状态，请求中止可能会导致两种结果：

- 如果邮箱中的消息发送失败，邮箱变为就绪状态，则发送请求中止，邮箱变为空邮箱并清除 CAN_TSTS.TXOKM 位；
- 如果邮箱中的消息发送成功，邮箱变为空邮箱，CAN_TSTS.TXOKM 位将由硬件设置为 1。

因此，当发送邮箱处于发送状态时，无论发送结果如何，发送操作完成后邮箱都会变成空邮箱。

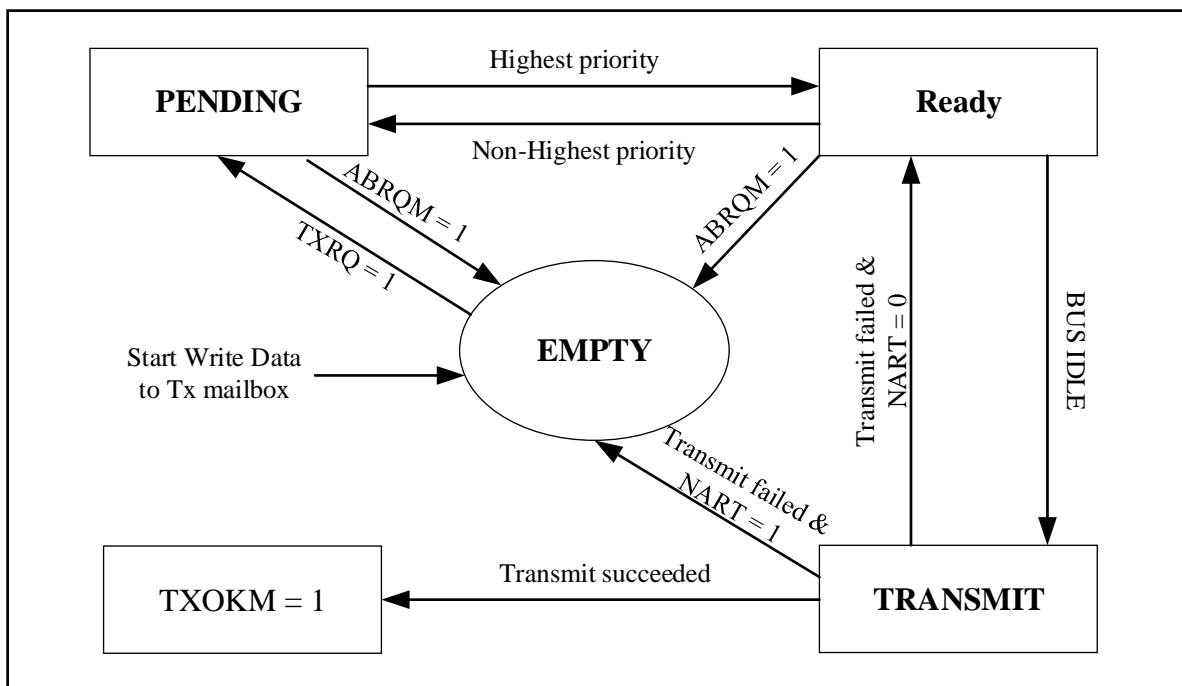
21.4.2 时间触发通信模式

CAN 的内部定时器在时间触发通信模式下被激活。它在每个 CAN 位时间递增（参见第 21.4.7 节）。CAN 在收发帧起始位的采样点位置对内部定时器的值进行采样，采样值即为收发邮箱的时间戳。内部定时器生成的时间戳将分别存储在 CAN_RMDTx/CAN_TMDTx 寄存器中。

21.4.3 非自动重传模式

要启用非自动重传模式，应设置 CAN_MCTRL.NART 位。该模式对应 CAN 标准中时间触发通信选项的功能。在非自动重传模式下，发送操作只会执行一次。如果发送操作失败，无论是仲裁丢失还是错误，硬件都不会自动再次发送报文。发送操作结束时，硬件判定发送请求已经完成，硬件设置 CAN_TSTS.RQCPM 位。同时，传输结果可以查询 CAN_TSTS.TXOKM、CAN_TSTS.ALSTM 和 CAN_TSTS.TERRM 位。

图 21-7 发送邮箱状态



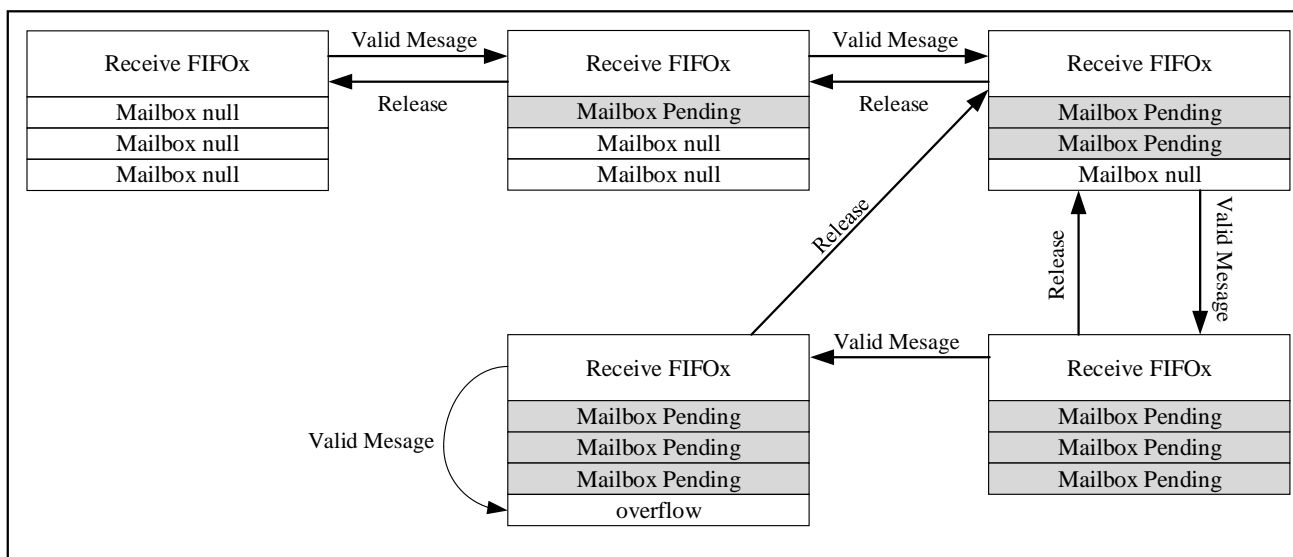
21.4.4 接收管理

具有 3 级深度的 FIFO 用于存储接收到的消息。当应用程序读取 FIFO 输出邮箱时，它会读取 FIFO 中第一个接收到的消息。FIFO 完全由硬件管理，可以简化应用程序，保证数据的一致性，减少 CPU 的处理时间。

21.4.4.1 有效报文

根据 CAN 协议，当消息被正确接收（直到 EOF 字段的最后一位没有错误发送）并通过标识符过滤时，则该消息被标记为有效消息。请参考 21.4.5 章节：标识符过滤。

图 21-8 接收邮箱状态



21.4.4.2 FIFO 接收

一个 FIFO 包含三级深度的邮箱，初始状态为空。接收到第一个有效消息后，其中一个邮箱将被挂起，硬件会将 CAN_RFF.FFMP[1:0]位设置为 1，表示收到有效消息。在释放邮箱之前再次收到有效消息。这时候会同时挂起两个邮箱，硬件会将 CAN_RFF.FFMP[1:0]位设置为 2，表示有两个有效的报文处于挂起状态。如上所述，第三条有效消息将暂停所有三个邮箱并将 CAN_RFF.FFMP[1:0]位设置为 3。

当 FIFO 的三级邮箱全部挂起时，再次接收到有效报文会导致邮箱溢出丢失报文，硬件会将 CAN_RFF.FFOVR 位设置为 1，表示事件发生。丢失消息的规则取决于 FIFO 的配置。如果禁用 FIFO 锁定功能（清除 CAN_MCTRL.RFLM 位），则 FIFO 中接收的最后一条消息将被新消息覆盖。这样最新接收到的报文不会被丢弃；如果启用 FIFO 锁定功能（设置 CAN_MCTRL.RFLM 位），那么新接收到的报文将被丢弃，软件可以读取 FIFO 中的前三个报文。

21.4.4.3 FIFO 释放

存储在 FIFO 中的消息将通过相应的接收邮箱读取。软件读取邮箱报文并通过设置 CAN_RFR.RFOM 位为 1 释放邮箱，CAN_RFF.FFMP[1:0]位减 1 直到为 0。

21.4.4.4 接收相关中断

当消息存储在接收 FIFO 中时，硬件将更新 CAN_RFF.FFMP[1:0]位。如果当前使能了 FIFO 报文挂起中断（CAN_INTE.FMPITE 位置位），则将产生挂起中断请求。

当存储第三条消息时，FIFO 变满，然后 CAN_RFF.FFULL 位将被置位，如果当前使能 FIFO 满中断（CAN_INTE.FFITE 位被置位），将产生一个 FIFO 满中断请求。

当 FIFO 溢出时，FFOVR 位将被设置。如果当前使能 FIFO 溢出中断（CAN_INTE.FOVITE 位置位），将产生 FIFO 溢出中断请求。

21.4.5 标识符过滤

在 CAN 网络中，当 CAN 处于发送器状态时，它通过向总线发送消息的方式向各个节点广播消息；当 CAN 处于接收者状态时，节点收到报文后根据报文的标识来判断是否需要该报文。如果报文有效，CAN 内核将报文复制到 SRAM（CAN 自身的 SRAM）；如果不需要，该消息将被丢弃。此过程不需要软件干预。与软件过滤相比，硬件过滤降低了 CPU 使用率。CAN 控制器为应用程序提供了 14 个位宽可变的可配置过滤器组（0~13）来满足这一需求，这些过滤器组用于接收仅软件所需的那些消息。每个过滤器组 x 包含两个 32 位寄存器，即 CAN_FxR1 和 CAN_FxR2。

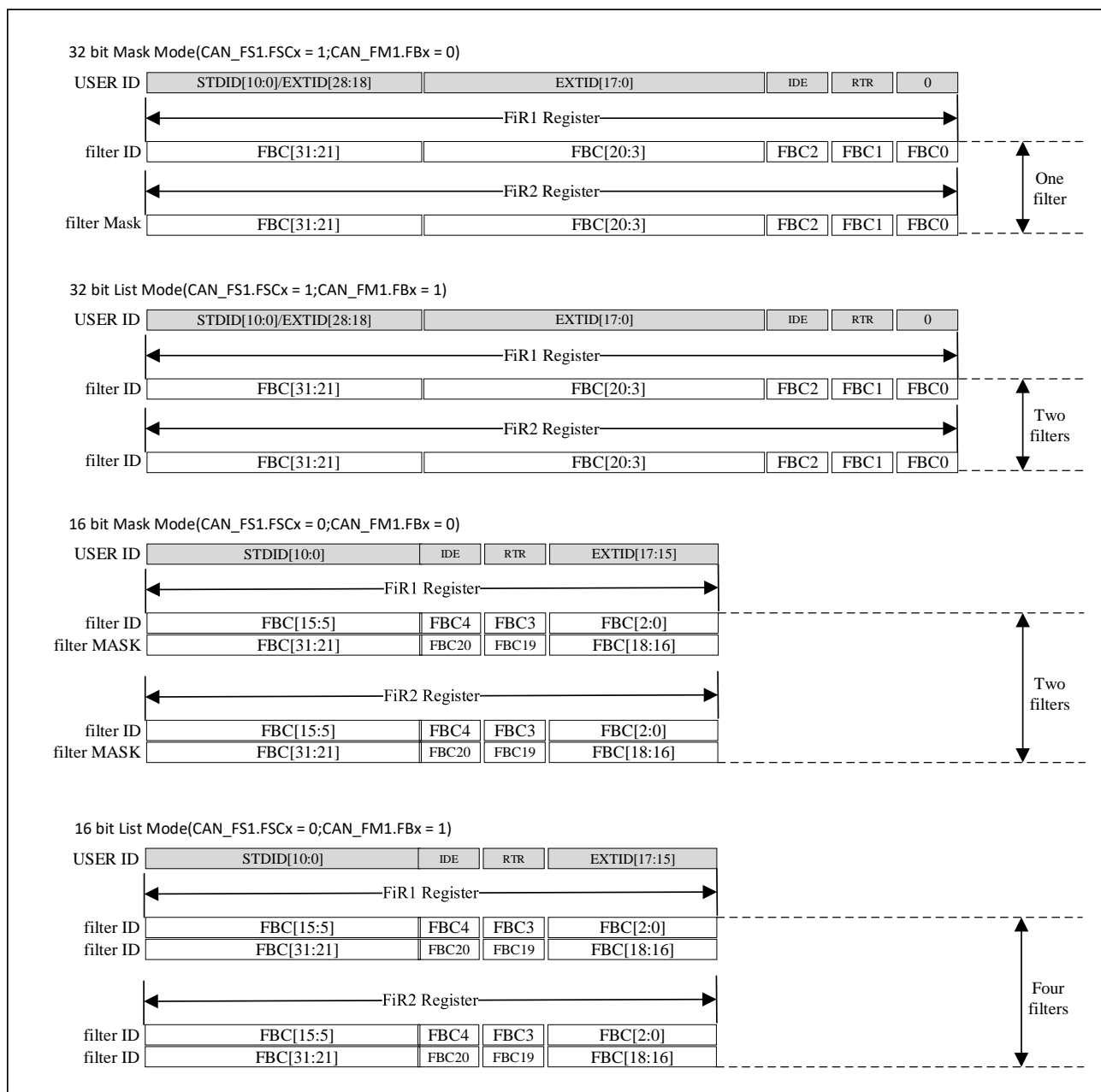
21.4.5.1 过滤器位宽和模式的设置

根据过滤器组的模式和位宽设置，对过滤器组中的每个过滤器进行编号（过滤器编号，从 0 到一个确定的最大值）。过滤器配置见下图。过滤器组可以通过对应的 CAN_FMC 寄存器进行配置。应用程序未使用的过滤器组应保持禁用状态。在配置过滤器组之前，必须通过清除 CAN_FA1.FAC 位将其设置为禁用状态。

通过设置相应的 CAN_FS1.FSCx 位，您可以配置过滤器组的位宽，参见图 21-9。

通过 CAN_FM1.FBx 位，可以将对应的掩码/标识符寄存器配置为标识符列表或标识符掩码模式。过滤器组设置为工作在掩码模式，可过滤出一组标识符。过滤器组设置为在标识符列表模式下工作，可过滤掉一个标识符。

图 21-9 过滤器位宽设置-寄存器组织



21.4.5.2 可变位宽

每个滤波器组的位宽可以独立配置。每个滤波器组可配置为一个 32 位滤波器：包括 STDID[10:0]、EXTID[17:0]、IDE 和 RTRQ 位；或两个 16 位过滤器，包括 STDID[10:0]、IDE、RTRQ 和 EXTID[17:15] 位，见图 21-9。此外，过滤器可以配置为两种不同的模式，即掩码模式和标识符列表模式。

掩码模式

过滤器 ID 用于存储标识符格式，过滤器掩码用于指示哪些位必须检查，哪些位可以忽略。

标识符列表模式

过滤器 ID 用于存储标识符格式。此时没有掩码进行比较，可以使用掩码位多存储一个过滤器 ID。但此时消息的标识需要与过滤器 ID 格式完全一致，否则无法通过过滤器。

21.4.5.3 过滤匹配序列号

CAN 内核收到有效消息后，会将消息 ID 与过滤器一一匹配，直到有一个过滤器通过或所有过滤器都失败。如果此消息未能通过任何启用的过滤器，则它将被丢弃。否则当 CAN 内核找到 ID 可以通过的第一个过滤器时，它会将过滤器索引与 CAN 报文打包并根据过滤器设置存储在 SRAM 中的接收 FIFO 中（CAN_FFA1 决定存储在哪个 FIFO 中）。用户可以在 CAN_RMDTx 寄存器的 FMI[7:0]位中找到滤波器索引。此过滤器匹配索引可以帮助识别它在此接收 FIFO 中的消息类型。

过滤器匹配序列号有两种使用方式。第一个是将过滤器匹配序列号与一系列预期值进行比较。另一种是使用过滤器匹配的序列号作为索引来访问目标地址。在对过滤器进行编号时，不考虑过滤器组是否处于活动状态。此外，每个 FIFO 为其关联的过滤器编号。请参考以下示例。

对于掩码模式的过滤器，软件只需要比较需要的掩码位（必须匹配的位）。对于标识符列表模式的过滤器（非筛选过滤器），软件无需直接与标识符进行比较。

表 21-1 过滤器编号示例

指向 FIFOx	过滤器组	过滤器模式	FIFO0 过滤器编号
FIFO	0	32 位掩码模式	0
	2	16 位掩码模式	1/2
	5	32 位列表模式	3/4
	7	16 位列表模式	5/6/7/8
	9	32 位列表模式	9/10
	11	16 位列表模式	11/12/13/14
	13	32 位掩码模式	15
指向 FIFOx	过滤器组	过滤器模式	FIFO1 过滤器编号
FIFO1	1	32 位列表模式	0/1
	3	16 位列表模式	2/3/4/5
	4	32 位掩码模式	6
	6	16 位掩码模式	7/8
	8	32 位掩码模式	9
	10	16 位掩码模式	10/11
	12	32 位列表模式	12/13

21.4.5.4 过滤器优先规则

根据过滤器的不同配置，一个消息标识符可以被多个过滤器过滤是可能的；在这种情况下，首先根据位宽确定接收邮箱中存储的过滤器匹配序列号，32 位宽的过滤器比 16 位宽的过滤器具有更高的优先级。对于具有相同位宽的过滤器，标识符列表模式优先于掩码模式。如果过滤器具有相同的位宽和模式，则优先级由过滤器组编号确定，编号较小的过滤器组优先级较高。在过滤器组内，过滤器编号越小，优先级越高。

图 21-10 过滤机制示例

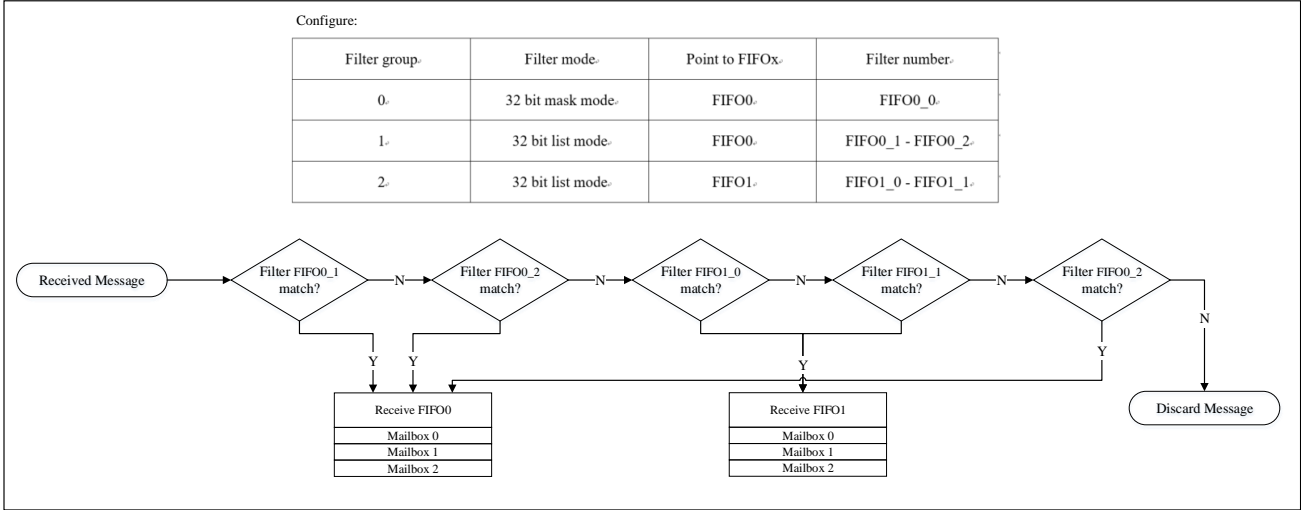


图 21-10 说明了 CAN 的过滤规则。收到消息时，首先将其标识符与标识符列表模式中配置的过滤器进行比较。如果匹配，则将消息存储在关联的 FIFO 中，并将匹配过滤器的序列号存储在过滤器匹配序列号中。如果不匹配，则将消息标识符与掩码模式中配置的过滤器进行比较。如果消息标识符与过滤器中的任何标识符不匹配，硬件将自动丢弃该消息而无需软件干预。

21.4.6 消息存储

邮箱包含与消息相关的所有信息：标识符、数据、控制、状态和时间戳信息。邮箱是软件和硬件之间传递消息的接口。

21.4.6.1 发送邮箱

在启用发送请求之前，应通过软件将消息写入一个空的发送邮箱。您可以通过 CAN_TSTS 寄存器查询发送状态。

表 21-2 发送邮箱寄存器列表

相对发送邮箱的基地址偏移	寄存器名称
0	CAN_TMIx
4	CAN_TMDTx
8	CAN_TMDLx
12	CAN_TMDHx

21.4.6.2 接收邮箱(FIFO)

CAN_RMDTx.FMI[7:0]字段可以存储过滤器匹配序列号，CAN_RMDTx.MTIM[15:0]字段可以存储 16 位时间戳。软件可以访问接收 FIFO 的输出邮箱读取接收到的报文。一旦软件对报文进行了处理，比如读出，软件应该设置 CAN_RFFx.RFFOM 位来释放相应的接收 FIFO，为以后的报文预留存储空间。

表 21-3 接收邮箱寄存器列表

相对接收邮箱的基地址偏移	寄存器名称
0	CAN_RMIx
4	CAN_RMDTx
8	CAN_RMDLx
12	CAN_RMDHx

21.4.7 位时序

位时间特性逻辑通过采样监控串行 CAN 总线，通过与帧起始位的边沿同步并与下一个边沿重新同步来调整其采样点。为避免软件编程错误，设置位时间特性寄存器（CAN_BTIM）只能在 CAN 初始化时进行。

其操作可以简单理解为将每个位时间分为三段：同步段（SYNC_SEG）、时间段 1（BS1）和时间段 2（BS2）。

通常，位变化将发生在 SYNC_SEG 中。其值固定为 1 个时间单位($1 \times t_{CAN}$)。

BS1 定义了采样点的位置。它包括 CAN 标准中的 PROP_SEG 和 PHASE_SEG1。其值可编成 1~16 个时间单位，但为了补偿网络中不同节点频率差异引起的相位正向漂移，也可自动扩展。

在 BS2 中，它定义了发送点的位置。它代表 CAN 标准中的 PHASE_SEG2。它的值可以编程为 1 到 8 个时间单位，但也可以自动缩短以补偿相位的负向漂移。

如果在 BS1 中检测到有效转换但在 SYNC_SEG 中没有检测到，则 BS1 的时间最多延长 RSJW 以延迟采样点。反之，如果在 BS2 中检测到有效转换但在 SYNC_SEG 中没有检测到，则 BS2 的时间最多缩短 RSJW 以提前采样点。

在上面的描述中，RSJW（重新同步跳转宽度）定义了每个比特可以延长或缩短多少时间单元的上限。它的值可以编程为 1 到 4 个时间单位。有效转换定义为 CAN 本身不发送隐性位时从显性位到隐性位的第一次转换。

注：1. CAN 位的时间特性和重同步机制详见 ISO11898 标准。

2. 为了提高 CAN 位时间精度，不推荐使用 HSI 作为时钟源。

图 21-11 位时序

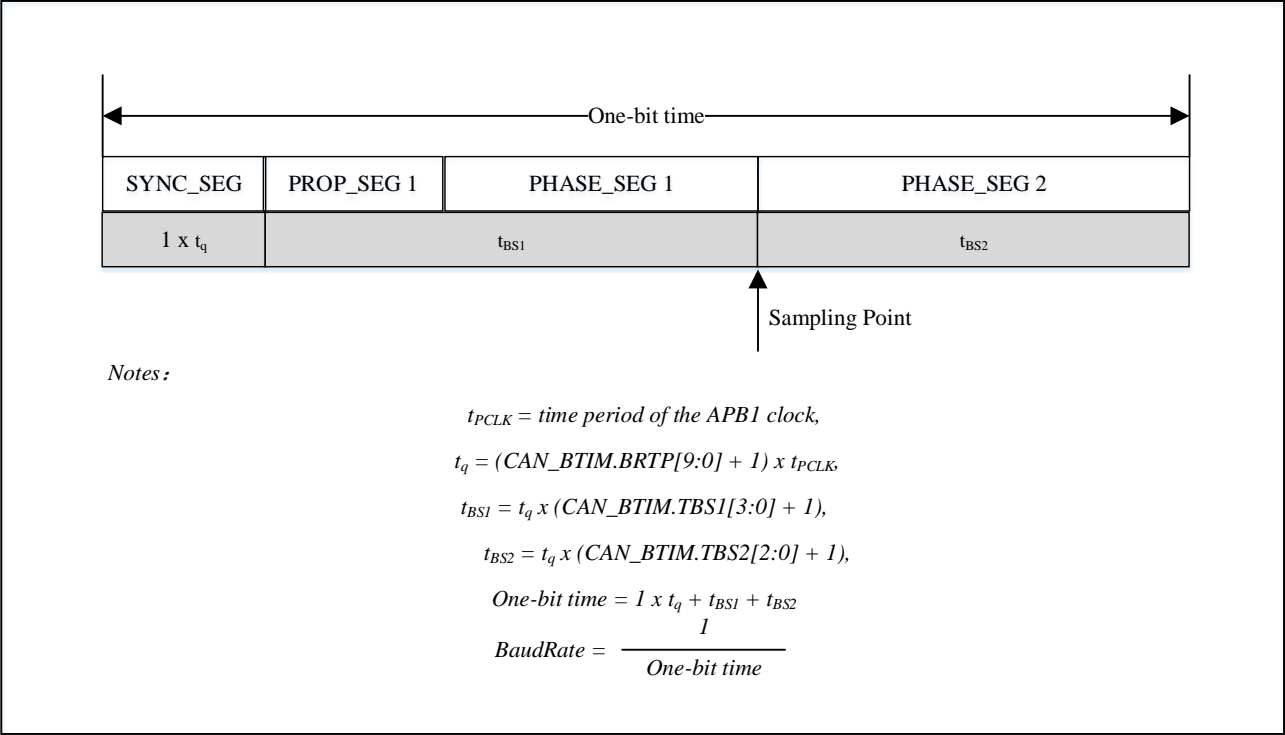
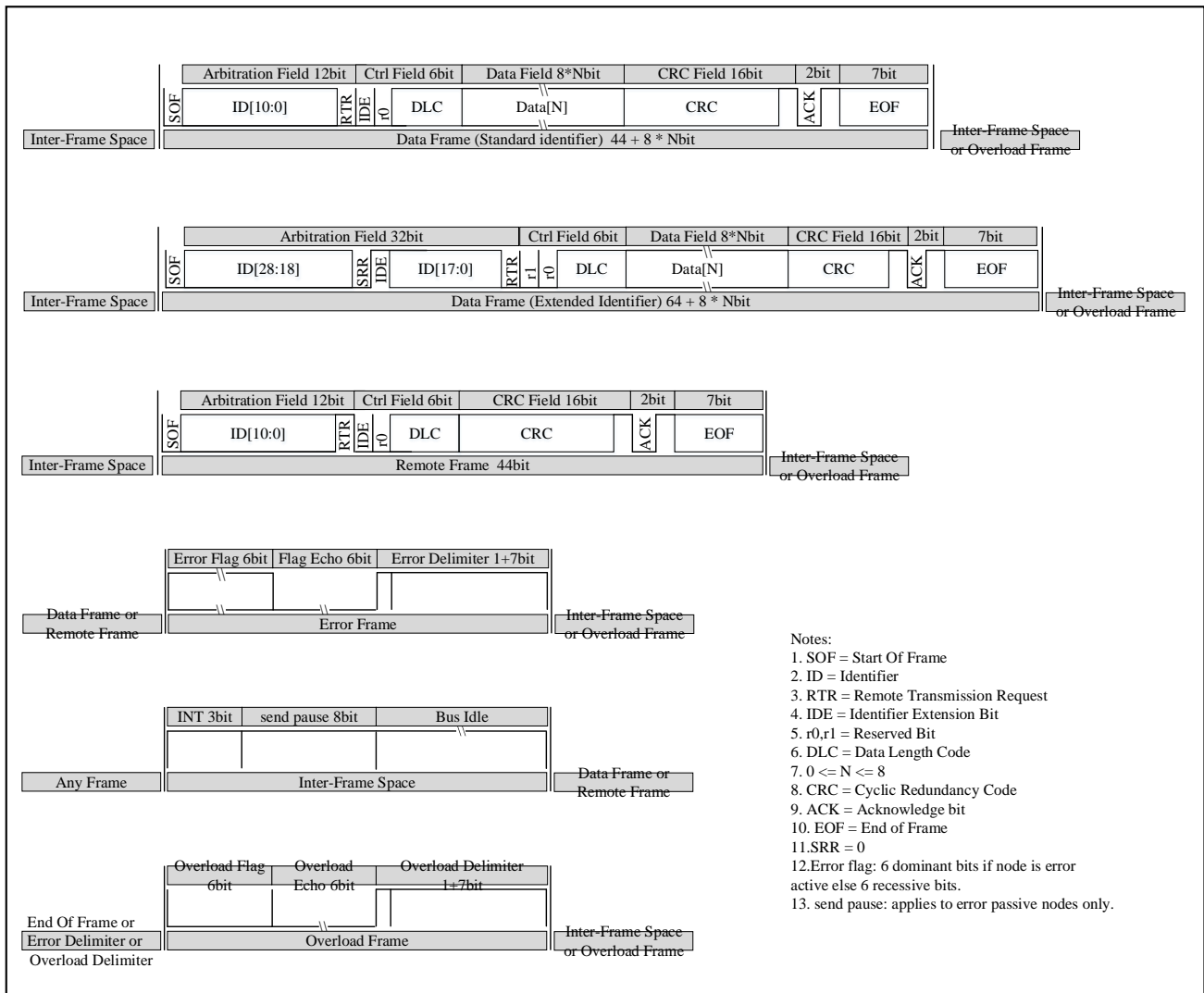
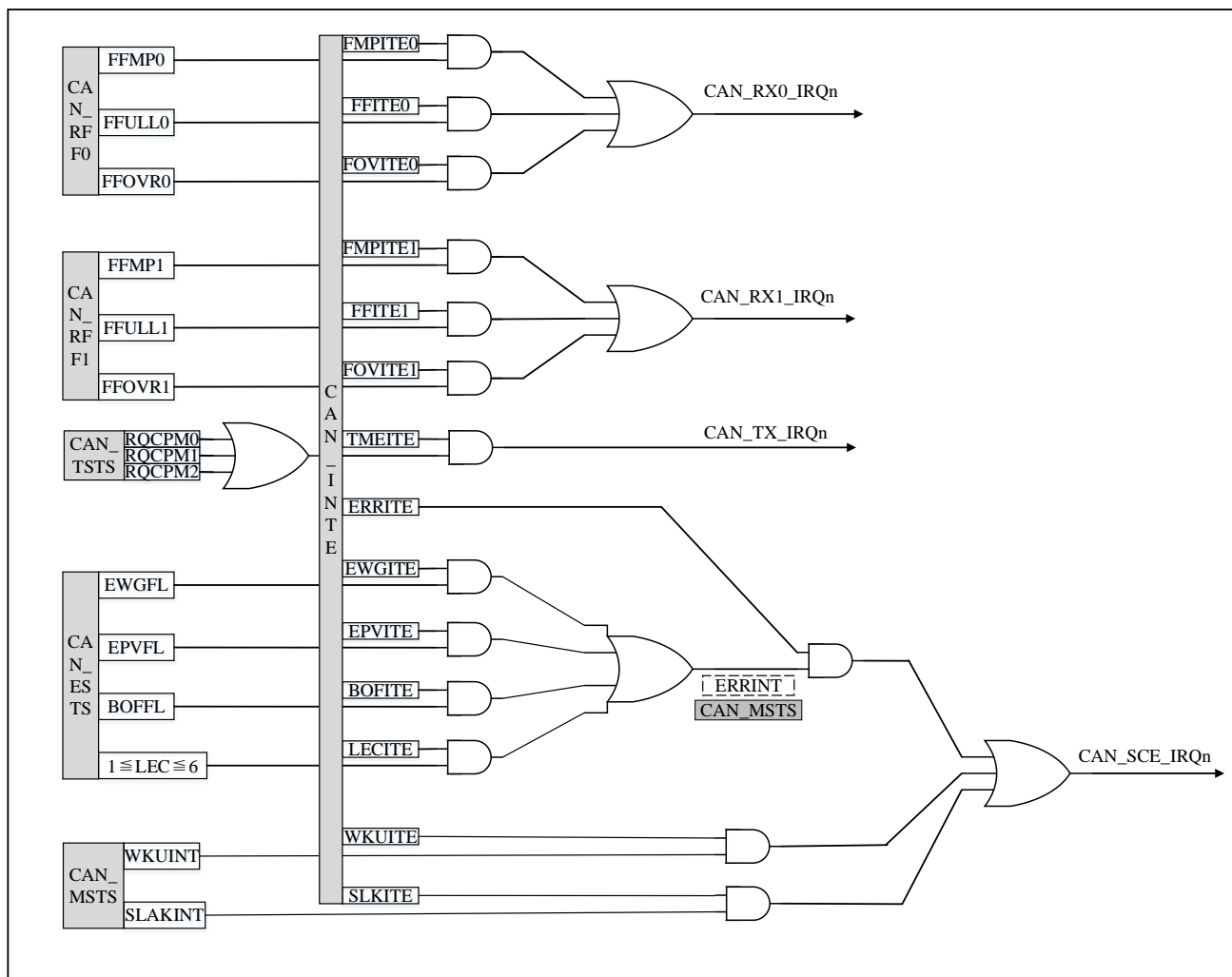


图 21-12 各类帧格式



21.5 CAN 中断

图 21-13 事件标志和中断产生



CAN 有四个中断向量。通过设置 CAN 中断启用寄存器（CAN_INTE），您可以单独启用或禁用每个中断源。以下是可以产生每个中断的事件。

■ FIFO0 中断(CAN_RX0_IRQn):

FIFO0 收到一条新消息，并且 CAN_RFF0.FFMP0 位不是'00'；

当 FIFO0 变满时，并且 CAN_RFF0.FFULL0 位被置位；

当 FIFO0 溢出，并且 CAN_RFF0.FFOVR0 位被置位。

■ FIFO1 中断(CAN_RX1_IRQn):

FIFO1 接收到一条新消息，并且 CAN_RFF1.FFMP1 位不是“00”。

当 FIFO1 变满时，并且 CAN_RFF1.FFULL1 位被置位。

当 FIFO1 溢出，并且 CAN_RFF1.FFOVR1 位被置位。

■ 发送中断(CAN_TX_IRQn):

发送邮箱 x 变空，对应的 CAN_TSTS.RQCPM_x 位被置位(x=1/2/3)。

■ 错误和状态转变中断(CAN_SCE_IRQn):

CAN 进入睡眠模式;

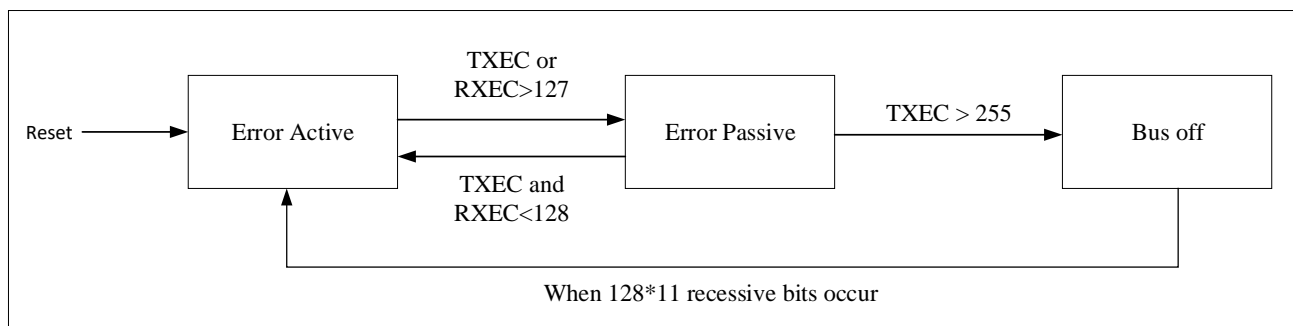
唤醒条件，在 CAN 接收引脚上监视到帧起始位(SOF)。

错误情况，请参考 CAN 错误状态寄存器(CAN_ESTS)了解错误详情。

21.5.1 错误管理

如 CAN 协议所述，错误管理完全由硬件通过发送错误计数器（CAN_ESTS.TXEC 字段）和接收错误计数器（CAN_ESTS.RXEC 字段）来实现。计数器值会根据错误情况增加或减少。如果您想了解有关 CAN_ESTS.TXEC 和 CAN_ESTS.RXEC 管理的更多详细信息，请参阅 CAN 标准。

图 21-14 CAN 错误状态框图



软件可以读取发送/接收错误计数器的值来判断 CAN 网络的稳定性，读取 CAN_ESTS.LEC[2:0]位可以得到当前错误状态的详细信息。更重要的是，通过设置 CAN_INTE 寄存器（如 CAN_INTE.LECITE 位），软件可以灵活控制检测到错误时中断的产生。

21.5.2 总线关闭恢复

当 TXEC 大于 255 时，CAN_ESTS.BOFFL 位置位，表示 CAN 总线关闭。此时，CAN 无法接收和发送消息。

在正常模式下，根据 CAN_MCTRL.ABOM 位，CAN 可以自动或根据软件的要求，从总线关闭状态恢复，并切换到主动错误状态。如果设置了 CAN_MCTRL.ABOM 位，恢复过程将在进入总线关闭状态后自动启动。否则，软件必须请求 CAN 进入然后退出初始化模式后，才会开始恢复过程。在这两种情况下，CAN 都必须等待 CAN 标准中描述的恢复过程，即在 CANRX 引脚上检测到 128*11 个连续的隐性位。

在初始化模式下，CAN 不会监控 CANRX 引脚的状态，因此无法完成恢复过程。

21.6 CAN 配置流程

本章将介绍 CAN 的常用配置过程，其他详细信息如各模式的功能和寄存器位将在本手册的其他部分进行介绍。CAN 配置流程可以分为多个阶段。只要满足先前的要求（例如，过滤器值），就可以随时更改某些配置。

■ 准备阶段:

1. 配置 RCC 使能 CAN 时钟

2. 配置 RCC 使能 AFIO 和 GPIO 时钟
3. 写入 GPIO 寄存器以将 CAN TX 和 CAN RX 信号映射到所需的 GPIO 引脚。

■ 基础配置阶段:

1. 复位后 CAN 设备以睡眠模式启动。
2. 通过清除 CAN_MCTRL.SLPRQ 位退出睡眠模式。
3. 通过设置 CAN_MCTRL.INIRQ 位进入初始化模式。
4. 等待 CAN_MSTS.INIAK 位变为 1（进入初始化模式）。
5. 通过将值写入 CAN_BTIM.BSJW、CAN_BTIM.TBS2、CAN_BTIM.TBS1 和 CAN_BTIM.BRTP 位来配置 CAN 的位时序。CAN 总线的波特率由以下公式定义：

$$\text{波特率} = \frac{1}{(1 + (TBS1 + 1) + (TBS2 + 1)) * (BRTP * t_{pclk})}$$

6. 通过写入寄存器中的 CAN_BTIM.SLM（静默）或 CAN_BTIM.LBM 来配置 CAN 的工作模式选项。
7. 通过 CAN_MCTRL 配置 CAN 行为（TTCM、ABOM、AWKUM、NART、RFLM、TXFP）。该寄存器中的大部分配置都可以即时更改，但建议不要这样做。否则在几个周期内，CAN 行为将变得不可预测。
8. 通过清除 CAN_MCTRL.INIRQ 位退出初始化模式。
9. 等待 CAN_MSTS.INIAK 位变为 0（退出初始化模式）。
10. 用户可以使用过滤器来过滤他们想要接收的消息。要配置滤波器，用户需要将“1”写入 CAN_FMC.FINITM 位以请求滤波器进入初始化模式。当滤波器处于初始化模式时，CAN 停止接收。
11. 为每个过滤器配置工作模式（CAN_FM1）、过滤器比例（CAN_FS1）和过滤器分配（CAN_FFA1）。用户还可以在此期间更改过滤器值（CAN_FiRx）。完成过滤器配置后，清除 CAN_FMC.FINITM 位以退出过滤器初始化。

■ 发送:

1. 使能必要的发送相关中断 CAN_INTE.TMEITE 位。
2. 检查 CAN_TSTS 中每个邮箱的状态位。如果任何一个 TMEIx(x=0~2)为‘1’的邮箱，用户可以将等待发送的消息写入对应的邮箱地址。CAN_TMIx.TXRQ(x=0~2)位必须在该邮箱被编程后写入“1”。
3. 一段时间后或等待发送中断后，返回检查 CAN_TSTS 中的发送状态。重复步骤 2~3 进行新消息传输。

■ 接收:

1. 当相应的过滤器被禁用时，用户也可以更改过滤器值（CAN_FiRx）。要禁用某个过滤器，用户需要将“0”写入 CAN_FA1 寄存器中的相应位。
2. 在 CAN_INTE 寄存器中配置接收相关的中断。
3. 一旦 CAN 接收到报文并将其存储在接收 FIFO 中，用户需要按时读取相应的 FIFO 并通过向寄存器 CAN_RFFx(x=0,1)中的 RFFOMx 写入“1”释放接收邮箱。

21.7 CAN 寄存器

这些外设寄存器必须以字的方式（32 位）操作。

21.7.1 寄存器描述

21.7.1.1 寄存器访问保护

当 CAN 节点正常工作时，不正确的访问/修改某些配置寄存器可能会导致节点出现硬件错误，暂时干扰整个 CAN 网络。因此，只有在 CAN 内核处于初始化模式时才允许修改 CAN_BTIM 寄存器。

只有当发送邮箱状态位 CAN_TSTS.TMEM=1 时用户才能修改发送邮箱的数据。

21.7.1.2 控制和状态寄存器

通过配置这些寄存器，用户可以：配置 CAN 参数，例如工作模式和波特率；开始消息发送；处理消息接收；中断设置；读取诊断信息。

21.7.1.3 邮箱寄存器描述

发送邮箱和接收邮箱基本相同，只是接收邮箱是只读的，包含 CAN_RMDTx.FMI 字段。发送邮箱为空时可写。

注意：设置了对应的 CAN_TSTS.TMEM 位，表示发送邮箱为空。

有 3 个发送邮箱和 2 个接收 FIFO。每个接收 FIFO 有 3 个邮箱，只能访问 FIFO 中第一个接收到的报文。

每个邮箱包含 4 个寄存器：

FIFO0 包含 CAN_RMI0、CAN_RMDT0、CAN_RMDL0、CAN_RMDH0；

FIFO1 包含 CAN_RMI1、CAN_RMDT1、CAN_RMDL1、CAN_RMDH1；

发送邮箱（x）包含 CAN_TMIx、CAN_TMDTx、CAN_TMDLx、CAN_TMDHx；x=0,1,2。

21.7.1.4 过滤寄存器描述

只有在关闭相应的过滤器组或设置了 CAN_FMC.FINITM 位时才能修改过滤器的值。另外，只有当整个滤波器设置为初始化模式（即 CAN_FMC.FINITM=1）时，才能修改滤波器的设置，即可以修改 CAN_FM1、CAN_FS1 和 CAN_FFA1 寄存器。

21.7.2 CAN 寄存器总览

表 21-4 CAN 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
000h	CAN_MCTRL	Reserved																DBGF	MRST	Reserved								TTCM	ABOM	AWKUM	NART	RFLM	TXFP	SLPRQ	INIRO	
	Reset Value																	1	0									0	0	0	0	0	0	0	0	0
004h	CAN_MSTS	Reserved																		RXS	LSMP	RXMD	TXMD	Reserved				SLAKINT	WKUINT	ERRINT	SLPAK	INIACK				
	Reset Value																			1	1	0	0					0	0	0	0	0	0	0	0	0
008h	CAN_TSTS	LOWM[2:0]		TMEM[2:0]			CODE[1:0]		ABRQM2		Reserved				TERRM2	ALSTM2	TXOKM2	RQCPM2	ABRQM1	Reserved				TERRM1	ALSTM1	TXOKM1	RQCPM1	ABRQM0	Reserved				TERRM0	ALSTM0	TXOKM0	RQCPM0
	Reset Value	0	0	0	1	1	1	0	0	0					0	0	0	0	0					0	0	0	0	0					0	0	0	0
00Ch	CAN_RFF0	Reserved																												RFFOM0	FFOVR0	FFULL0	Reserved		FFMP0[1:0]	
	Reset Value																													0	0	0			0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
010h	CAN_RFF1	Reserved																								RFFOM1	FFOVR1	FFULL1	Reserved	FFMP1[1:0]												
	Reset Value																									0	0	0		0	0											
014h	CAN_INTE	Reserved														SLKITE	WKUTE	ERRITE	Reserved				LECITE	BOFITE	EPVITE	EWGITE	Reserved	FOVITE1	FFITE1	FMPITE1	FOVITE0	FFITE0	FMPITE0	TMEITE								
	Reset Value															0	0	0	Reserved				0	0	0	0	Reserved		0	0	0	0	0	0	0							
018h	CAN_ESTS	RXEC[7:0]								TXEC[7:0]								Reserved								LEC[2:0]			Reserved		BOFFL	EPVFL	EWGFL									
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	Reserved		0	0	0	0								
01Ch	CAN_BTIM	SLM	LBM	Reserved				RSIW[1:0]		Reserved	TBS2[2:0]				TBS1[3:0]				Reserved				BRTP[9:0]																			
	Reset Value	0	0	Reserved				0	1	Reserved	0	1	0	0	0	1	1	Reserved				0	0	0	0	0	0	0	0	0	0	0	0	0								
020h - 17Fh	Reserved																																									
180h	CAN_TMI0	STDID[10:0]/EXTID[28:18]														EXTID[17:0]														IDE		RTRQ	TXRQ									
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0										
184h	CAN_TMDT0	MTIM[15:0]														Reserved						TGT	Reserved						DLC[3:0]													
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x							x	Reserved						x	x	x	x									
188h	CAN_TMDL0	DATA3[7:0]								DATA2[7:0]						DATA1[7:0]						DATA0[7:0]																				
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x										
18Ch	CAN_TMDH0	DATA7[7:0]								DATA6[7:0]						DATA5[7:0]						DATA4[7:0]																				
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x										
190h	CAN_TMI1	STDID[10:0]/EXTID[28:18]														EXTID[17:0]														IDE		RTRQ	TXRQ									
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0										
194h	CAN_TMDT1	MTIM[15:0]														Reserved						TGT	Reserved						DLC[3:0]													
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x							x	Reserved						x	x	x	x									
198h	CAN_TMDL1	DATA3[7:0]								DATA2[7:0]						DATA1[7:0]						DATA0[7:0]																				
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x										
19Ch	CAN_TMDH1	DATA7[7:0]								DATA6[7:0]						DATA5[7:0]						DATA4[7:0]																				
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x										
1A0h	CAN_TMI2	STDID[10:0]/EXTID[28:18]														EXTID[17:0]														IDE		RTRQ	TXRQ									
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0										
1A4h	CAN_TMDT2	MTIM[15:0]														Reserved						TGT	Reserved						DLC[3:0]													
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x							x	Reserved						x	x	x	x									

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1A8h	CAN_TMDL2	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1ACh	CAN_TMDH2	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1B0h	CAN_RMI0	STDID[10:0]/EXTID[28:18]												EXTID[17:0]																IDE	RTRQ	Reserved	
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
1B4h	CAN_RMDT0	MTIM[15:0]																FMI[7:0]								Reserved	DLC[3:0]						
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x		
1B8h	CAN_RMDL0	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1BCh	CAN_RMDH0	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1C0h	CAN_RMI1	STDID[10:0]/EXTID[28:18]												EXTID[17:0]																IDE	RTRQ	Reserved	
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
1C4h	CAN_RMDT1	MTIM[15:0]																FMI[7:0]								Reserved	DLC[3:0]						
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	
1C8h	CAN_RMDL1	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1CCh	CAN_RMDH1	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1D0h - 1FFh	Reserved																																
200h	CAN_FMC	Reserved																															FINITM
	Reset Value																																
204h	CAN_FMI	Reserved																FB[13:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
208h	Reserved																																
20Ch	CAN_FS1	Reserved																FSC[13:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
210h	Reserved																																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
214h	CAN_FFA1	Reserved																		FAF[13:0]													
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
218h	Reserved																																
21Ch	CAN_FA1	Reserved																		FAC[13:0]													
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
220h	Reserved																																
224h - 23Fh	Reserved																																
240h	CAN_F0B1	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
244h	CAN_F0B2	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
248h	CAN_F1B1	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
24Ch	CAN_F1B2	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
.	Reserved																															
2A8h	CAN_F13B1	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
2ACh	CAN_F13B2	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

21.7.3 CAN 控制和状态寄存器

寄存器描述中使用的缩写，请参考 1.1 部分。

21.7.3.1 CAN 主控制寄存器(CAN_MCTRL)

偏移地址:0x00

复位值:0x0001 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															DBGF
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRST	Reserved							TTCM	ABOM	AWKUM	NART	RFLM	TXFP	SLPRQ	INIRQ
rw								rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:17	Reserved	保留，必须保持复位值
16	DBGF	调试冻结（Debug freeze） 0：在调试时，CAN 照常工作 1：在调试时，冻结 CAN 的接收/发送。仍然可以正常地读写和控制接收 FIFO。。 注：CAN 冻结时必须设置 DBG_CTRL.CAN_STOP 位，请参考 21.3.7：调试模式。
15	MRST	CAN 软件复位（bxCAN software master reset） 0：该外设正常工作； 1：强制复位 CAN，之后 CAN 进入睡眠模式，CAN_RFFx.FFMP 位和 CAN_MCTRL 寄存器被初始化为其复位值。之后，硬件自动清除该位。
14:8	Reserved	保留，必须保持复位值
7	TTCM	时间触发通信模式（Time triggered communication mode） 0：禁用时间触发通信模式； 1：使能时间触发通信模式。 注：关于时间触发通信方式的更多信息，请参考 21.4.2：时间触发通信方式。
6	ABOM	自动离线（Bus-Off）管理（Automatic bus-off management） 该位确定 CAN 硬件可以退出总线关闭状态的条件。 0：退出总线关闭状态的过程是软件设置 CAN_MCTRL.INIRQ 位然后清零后，一旦硬件检测到 128 个连续的 11 位隐性位，就退出总线关闭状态； 1：一旦硬件检测到 128 个连续的 11 位隐性位，将自动退出总线关闭状态。 注：有关总线关闭状态的更多信息，请参阅 21.5.1：错误管理。
5	AWKUM	自动唤醒模式（Automatic wakeup mode） 该位决定 CAN 在睡眠模式下是被硬件唤醒还是软件唤醒。 0：软件通过清除 CAN_MCTRL.SLPRQ 位唤醒睡眠模式； 1：睡眠模式由硬件通过检测 CAN 报文自动唤醒。 在唤醒的同时，硬件自动清除 CAN_MSTS.SLPRQ 和 CAN_MSTS.SLPAK 位。
4	NART	禁止报文自动重传（No automatic retransmission） 0：根据 CAN 标准，当 CAN 硬件发送报文失败时，会自动重传，直到发送成功； 1：CAN 报文只发送一次，与发送结果无关（成功、错误或仲裁失败）。

位域	名称	描述
3	RFLM	接收 FIFO 锁定模式 (Receive FIFO locked mode) 0: 接收溢出时 FIFO 不锁定, 当接收 FIFO 的报文没有被读出时, 下一个接收到的报文会覆盖上一条报文; 1: 接收溢出时锁定 FIFO。当接收 FIFO 的报文没有被读出时, 下一个接收到的报文将被丢弃。
2	TXFP	发送 FIFO 优先级 (Transmit FIFO priority) 当有多个消息同时等待发送时, 该位决定这些消息的发送顺序。 0: 优先级由消息的标识符决定; 1: 优先级由发送请求的顺序决定。
1	SLPRQ	睡眠模式请求 (Sleep mode request) 软件可以通过设置该位请求 CAN 进入睡眠模式, 一旦当前 CAN 活动 (发送或接收报文) 结束, CAN 将进入睡眠模式。 软件清零使 CAN 退出睡眠模式。 当 CAN_MCTRL.AWKUM 位置位并且在 CAN Rx 信号中检测到 SOF 位时, 硬件会清除该位。 该位在复位后置位, 即 CAN 在复位后处于睡眠模式。
0	INIRQ	初始化请求 (Initialization request) 软件清除该位可以使 CAN 退出初始化模式: 当 CAN 离开初始化模式进入正常模式时, 需要在接收引脚上检测到 11 个连续的隐性位, CAN 将同步并准备好接收和发送数据, 此时硬件相应地清除 CAN_MSTS.INIAK 位。 通过软件设置该位使 CAN 从正常操作模式进入初始化模式: 一旦当前 CAN 活动 (发送或接收) 结束, 此时硬件设置 CAN_MSTS.INIAK 位, CAN 进入初始化模式。

21.7.3.2 CAN 主状态寄存器 (CAN_MSTS)

偏移地址: 0x04

复位值: 0x0000c02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved				RXS	LSMP	RXMD	TXMD	Reserved				SLAKINT	WKUINT	ERRINT	SLPAK	INIAK
				r	r	r	r					rc_wl	rc_wl	rc_wl	r	r

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值
11	RXS	CAN 接收电平 (CAN Rx signal) 该位反映 CAN 接收引脚 (CAN_RX) 的实际电平。
10	LSMP	上次采样值 (Last sample point) CAN 接收引脚的上次采样值 (对应于当前接收位的值)。

位域	名称	描述
9	RXMD	接收模式 (Receive mode) 该位为'1'表示 CAN 当前为接收器。
8	TXMD	发送模式 (Transmit mode) 该位为'1'表示 CAN 当前为发送器。
7:5	Reserved	保留, 必须保持复位值
4	SLAKINT	睡眠确认中断 (Sleep acknowledge interrupt) 当 CAN_INTE.SLKITE=1 时, 一旦 CAN 进入睡眠模式, 硬件会设置该位, 然后触发相应的中断。当该位置位时, 如果 CAN_INTE.SLKITE 位置位, 将产生状态改变中断。 软件可以清零该位, 当 CAN_MSTS.SLPAK 位清零时, 硬件也会清零该位。 <i>注: 当 CAN_INTE.SLKITE=0 时, 不应查询该位, 但应查询 CAN_MSTS.SLPAK 位以了解睡眠状态。</i>
3	WKUINT	唤醒中断 (Wakeup interrupt) 当 CAN 处于睡眠状态时, 一旦检测到帧起始位 (SOF), 硬件将设置该位; 如果设置了 CAN_INTE.WKUIE 位, 则会生成状态改变中断。 该位由软件清零。
2	ERRINT	错误中断 (Error interrupt) 当检测到错误时, 会设置 CAN_ESTS 寄存器的某个位, 如果 CAN_INTE 寄存器的相应中断使能位也被设置, 则硬件会设置该位; 如果设置了 CAN_INTE.ERRITE 位, 则会生成状态更改中断。该位由软件清零。
1	SLPAK	睡眠模式确认 (Sleep acknowledge) 该位由硬件置位, 表示软件 CAN 模块处于睡眠模式。该位用于确认软件请求进入睡眠模式 (设置 CAN_MCTRL.SLPRQ 位)。 当 CAN 退出睡眠模式 (CAN 离开睡眠模式并进入正常模式, 需要与 CAN 总线同步) 时, 硬件清零该位。这里与 CAN 总线同步意味着硬件需要检测 CAN 的 RX 引脚上的 11 个连续的隐性位。 <i>注: 通过软件或硬件清除 CAN_MCTRL.SLPRQ 位将启动退出睡眠模式的过程。有关清除 CAN_MCTRL.SLPRQ 位的详细信息, 请参见 CAN_MCTRL.AWKUM 位的说明。</i>
0	INIAK	初始化确认 (Initialization acknowledge) 该位由硬件置位, 表示软件 CAN 模块处于初始化模式。该位是软件请求进入初始化模式的确认 (CAN_MCTRL.INIRQ 位被设置)。 当 CAN 退出初始化模式 (CAN 离开初始化模式并进入正常模式, 需要与 CAN 总线同步) 时, 硬件清零该位。这里与 CAN 总线同步意味着硬件需要检测 CAN 的 RX 引脚上的 11 个连续的隐性位。

21.7.3.3 CAN 发送状态寄存器 (CAN_TSTS)

偏移地址: 0x08

复位值: 0x1C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOWM2	LOWM1	LOWM0	TMEM2	TMEM1	TMEM0	CODE[1:0]		ABRQM2	Reserved			TERRM2	ALSTM2	TXOKM2	RQCPM2
r	r	r	r	r	r	r	r	rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQM1	Reserved			TERRM1	ALSTM1	TXOKM1	RQCPM1	ABRQM0	Reserved			TERRM0	ALSTM0	TXOKM0	RQCPM0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

位域	名称	描述
31	LOWM2	邮箱 2 最低优先级标志 (Lowest priority flag for mailbox 2) 当多个邮箱在等待发送报文, 且邮箱 2 的优先级最低时, 硬件对该位置'1'。
30	LOWM1	邮箱 1 最低优先级标志 (Lowest priority flag for mailbox 1) 当多个邮箱在等待发送报文, 且邮箱 1 的优先级最低时, 硬件对该位置'1'。
29	LOWM0	邮箱 0 最低优先级标志 (Lowest priority flag for mailbox 0) 当多个邮箱在等待发送报文, 且邮箱 0 的优先级最低时, 硬件对该位置'1'。 <i>注: 如果只有 1 个邮箱在等待, 则 CAN_TSTS.LOWM[2:0] 被清'0'。</i>
28	TMEM2	发送邮箱 2 空 (Transmit mailbox 2 empty) 当邮箱 2 中没有等待发送的报文时, 硬件对该位置'1'。
27	TMEM1	发送邮箱 1 空 (Transmit mailbox 1 empty) 当邮箱 1 中没有等待发送的报文时, 硬件对该位置'1'。
26	TMEM0	发送邮箱 0 空 (Transmit mailbox 0 empty) 当邮箱 0 中没有等待发送的报文时, 硬件对该位置'1'。
25:24	CODE[1:0]	邮箱号 (Mailbox code) 当有至少 1 个发送邮箱为空时, 这 2 位表示下一个空的发送邮箱号。 当所有的发送邮箱都为空时, 这 2 位表示优先级最低的那个发送邮箱号。
23	ABRQM2	邮箱 2 中止发送 (Abort request for mailbox 2) 设置该位, 软件可以停止邮箱 2 的发送请求, 当邮箱 2 的发送消息空闲时, 硬件清零该位。 如果邮箱 2 中没有等待发送的消息, 则设置该位无效。
22:20	Reserved	保留, 必须保持复位值
19	TERRM2	邮箱 2 发送失败 (Transmission error of mailbox 2) 当邮箱 2 因为出错而导致发送失败时, 对该位置'1'。
18	ALSTM2	邮箱 2 仲裁丢失 (Arbitration lost for mailbox 2) 当邮箱 2 因为仲裁丢失而导致发送失败时, 对该位置'1'。
17	TXOKM2	邮箱 2 发送成功 (Transmission OK of mailbox 2) 每次在邮箱 2 进行发送尝试后, 硬件对该位进行更新: 0: 上次发送尝试尚未成功; 1: 上次发送尝试成功。 当邮箱 2 的发送请求被成功完成后, 硬件对该位置'1'。请参见图 21-7。
16	RQCPM2	邮箱 2 请求完成 (Request completed mailbox 2)

位域	名称	描述
		当上次对邮箱 2 的请求（发送或中止）完成后，硬件将设置该位。 软件向该位写'1'可以清零；当硬件接收到发送请求时，也可以清零该位（CAN_TMI2.TXRQ 位被置位）。 当该位清零时，邮箱 2 的其他发送状态位（CAN_TSTS.TXOKM2、CAN_TSTS.ALSTM2 和 CAN_TSTS.TERRM2 位）也被清零。
15	ABRQM1	邮箱 1 中止发送（Abort request for mailbox 1） 设置该位，软件可以停止邮箱 1 的发送请求，当邮箱 1 的发送报文空闲时，硬件清零该位。如果邮箱 1 中没有等待发送的消息，则设置该位无效。
14:12	Reserved	保留，必须保持复位值
11	TERRM1	邮箱 1 发送失败（Transmission error of mailbox 1） 当邮箱 1 因为出错而导致发送失败时，对该位置'1'。
10	ALSTM1	邮箱 1 仲裁丢失（Arbitration lost for mailbox 1） 当邮箱 1 因为仲裁丢失而导致发送失败时，对该位置'1'。
9	TXOKM1	邮箱 1 发送成功（Transmission OK of mailbox 1） 每次在邮箱 1 进行发送尝试后，硬件对该位进行更新： 0：上次发送尝试尚未成功； 1：上次发送尝试成功。 当邮箱 1 的发送请求被成功完成后，硬件对该位置'1'。请参见图 21-7。
8	RQCPM1	邮箱 1 请求完成（Request completed mailbox 1） 当邮箱 1 的上次请求（发送或中止）完成时，硬件设置该位。 软件向该位写'1'可以清零；当硬件接收到发送请求时，也可以清除该位（CAN_TMI1.TXRQ 位被置位）。 当该位清零时，邮箱 1 的其他发送状态位（CAN_TSTS.TXOKM1、CAN_TSTS.ALSTM1 和 CAN_TSTS.TERRM1 位）也被清零。
7	ABRQM0	邮箱 0 中止发送（Abort request for mailbox 0） 软件可以通过设置该位来停止邮箱 0 的发送请求，当邮箱 0 的发送消息空闲时硬件清零该位。如果邮箱 0 中没有等待发送的消息，则设置该位无效。
6:4	Reserved	保留，必须保持复位值
3	TERRM0	邮箱 0 发送失败（Transmission error of mailbox 0） 当邮箱 0 因为出错而导致发送失败时，对该位置'1'。
2	ALSTM0	邮箱 0 仲裁丢失（Arbitration lost for mailbox 0） 当邮箱 0 因为仲裁丢失而导致发送失败时，对该位置'1'。
1	TXOKM0	邮箱 0 发送成功（Transmission OK of mailbox 0） 每次尝试发送邮箱 0 后，硬件都会更新此位： 0：上次发送尝试尚未成功； 1：上次发送尝试成功。 当邮箱 0 的发送请求成功完成时，硬件设置该位。 参见图 21-7。
0	RQCPM0	邮箱 0 请求完成（Request completed mailbox 0） 当上次对邮箱 0 的请求（发送或中止）完成后，硬件设置该位。 软件向该位写'1'可以清零；当硬件接收到发送请求时，也可以清除该位（CAN_TMI0.TXRQ 位被置位）。 当该位清零时，邮箱 0 的其他发送状态位（CAN_TSTS.TXOKM0、CAN_TSTS.ALSTM0 和 CAN_TSTS.TERRM0 位）也被清零。

21.7.3.4 CAN 接收 FIFO0 寄存器 (CAN_RFF0)

偏移地址: 0x0c

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFFOM0	FFOVR0	FFULL0	Reserved	FFMP0[1:0]	
										rs	rc_wl	rc_wl		r	r

位域	名称	描述
31:6	Reserved	保留，必须保持复位值
5	RFFOM0	释放接收 FIFO 0 输出邮箱（Release FIFO 0 output mailbox） 软件通过设置该位来释放接收 FIFO 的输出邮箱。如果接收 FIFO 为空，则设置该位没有影响，即只有在 FIFO 中有报文时，设置该位才有意义。如果 FIFO 中有两条以上的消息，由于 FIFO 的特性，软件需要释放输出邮箱才能访问第二条消息。 当输出邮箱被释放时，硬件清除该位。
4	FFOVR0	FIFO 0 溢出（FIFO 0 overrun） 当 FIFO0 已满，又收到新的报文且报文符合过滤条件，硬件对该位置'1'。该位由软件清'0'。
3	FFULL0	FIFO 0 满（FIFO 0 full） 当 FIFO 0 中有 3 个报文时，硬件对该位置'1'。该位由软件清'0'。
2	Reserved	保留，必须保持复位值
1:0	FFMP0[1:0]	FIFO 0 报文数目（FIFO 0 message pending） FIFO 0 报文数目这两位反映了当前存储在接收 FIFO0 中的报文数量。每在接收 FIFO0 中存储一条新报文，硬件将 CAN_RFF0.FFMP0 加 1。 每次软件向 CAN_RFF0.RFFOM0 位写入“1”以释放输出邮箱时，CAN_RFF0.FFMP0 减 1，直到为 0。

21.7.3.5 CAN 接收 FIFO1 寄存器 (CAN_RFF1)

偏移地址: 0x10 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFFOM1	FFOVR1	FFULL1	Reserved	FFMP1[1:0]	
										rs	rc_wl	rc_wl		r	r

位域	名称	描述
31:6	Reserved	保留，必须保持复位值
5	RFFOM1	释放接收 FIFO 1 输出邮箱（Release FIFO 1 output mailbox） 软件通过对该位置‘1’来释放接收 FIFO 的输出邮箱。如果接收 FIFO 为空，那么对该位置‘1’没有任何效果，即只有当 FIFO 中有报文时对该位置‘1’才有意义。如果 FIFO 中有 2 个以上的报文，由于 FIFO 的特点，软件需要释放输出邮箱才能访问第 2 个报文。 当输出邮箱被释放时，硬件对该位清‘0’。
4	FFOVR1	FIFO 1 溢出（FIFO 1 overrun） 当 FIFO 1 已满，又收到新的报文且报文符合过滤条件，硬件对该位置‘1’。该位由软件清‘0’。
3	FFULL1	FIFO 1 满（FIFO 1 full） 当 FIFO 1 中有 3 个报文时，硬件对该位置‘1’。该位由软件清‘0’。
2	Reserved	保留，必须保持复位值
1:0	FFMP1[1:0]	FIFO 1 报文数目（FIFO 1 message pending） FIFO 1 中的报文数 这两位反映了当前接收 FIFO 1 中存储的报文数量。接收 FIFO 1 中每存储一条新报文，硬件将 CAN_RFF1.FFMP1 加 1。 每次软件通过向 CAN_RFF1.RFFOM1 位写入“1”来释放输出邮箱时，CAN_RFF1.FFMP1 减 1，直到为 0。

21.7.3.6 CAN 中断使能寄存器(CAN_INTE)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														SLKITE	WKUITE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRITE	Reserved			LECITE	BOFITE	EPVITE	EWGITE	Reserved	FOVITE1	FFITE1	FMPITE1	FOVITE0	FFITE0	FMPITE0	TMEITE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:18	Reserved	保留，必须保持复位值
17	SLKITE	睡眠中断使能（Sleep interrupt enable） 0：当 CAN_MSTS.SLAKINT 位置位时，不产生中断； 1：当 CAN_MSTS.SLAKINT 位置位时，产生中断。
16	WKUITE	唤醒中断使能（Wakeup interrupt enable） 0：当 CAN_MSTS.WKUINT 位置位时，不产生中断； 1：当 CAN_MSTS.WKUINT 位置位时，产生中断。
15	ERRITE	错误中断使能（Error interrupt enable）

位域	名称	描述
		0: 当 CAN_ESTS 寄存器有错误发生时, 不产生中断; 1: 当 CAN_ESTS 寄存器有错误发生时, 产生中断。
14:12	Reserved	保留, 必须保持复位值
11	LECITE	上次错误号中断使能 (Last error code interrupt enable) 0: 检测到错误时, 当硬件置位 CAN_ESTS.LEC[2:0]时, CAN_MSTS.ERRINT 位不置位; 1: 检测到错误时, 当硬件设置 CAN_ESTS.LEC[2:0]时, 设置 CAN_MSTS.ERRINT 位。
10	BOFITE	离线中断使能 (Bus-off interrupt enable) 0: 当设置 CAN_ESTS.BOFFL 位时, 不设置 CAN_MSTS.ERRINT 位; 1: 设置 CAN_ESTS.BOFFL 位时, 设置 CAN_MSTS.ERRINT 位。
9	EPVITE	错误被动中断使能 (Error Passive Interrupt Enable) 0: 当设置 CAN_ESTS.EPVFL 位时, 不设置 CAN_MSTS.ERRINT 位; 1: 当设置 CAN_ESTS.EPVFL 位时, 设置 CAN_MSTS.ERRINT 位。
8	EWGITE	错误警告中断使能 (Error warning interrupt enable) 0: 当设置 CAN_ESTS.EWGFL 位时, 不设置 CAN_MSTS.ERRINT 位; 1: 当设置 CAN_ESTS.EWGFL 位时, 设置 CAN_MSTS.ERRINT 位。
7	Reserved	保留, 必须保持复位值
6	FOVITE1	FIFO 1 溢出中断使能 (FIFO overrun interrupt enable) 0: 当 CAN_RFF1.FOR 位置位时, 不产生中断; 1: 当 CAN_RFF1.FO 位置位时, 产生中断。
5	FFITE1	FIFO 1 满中断使能 (FIFO full interrupt enable) 0: 当 CAN_RFF1.FULL 位置位时, 不产生中断; 1: 当 CAN_RFF1.FULL 位置位时, 产生中断。
4	FMPITE1	FIFO 1 消息挂号中断使能 (FIFO message pending interrupt enable) 0: 当 CAN_RFF1.FFMP[1:0]位为非 0 时, 不产生中断; 1: 当 CAN_RFF1.FFMP[1:0]位不为 0 时, 产生中断。
3	FOVITE0	FIFO 0 溢出中断使能 (FIFO overrun interrupt enable) 0: 当 CAN_RFF0.FFOVR 位置位时, 不产生中断; 1: 当 CAN_RFF0.FFOVR 位置位时, 产生中断。
2	FFITE0	FIFO 0 满中断使能 (FIFO full interrupt enable) 0: 当 CAN_RFF0.FFULL 位置位时, 不产生中断; 1: 当 CAN_RFF0.FFULL 位置位时, 产生中断。
1	FMPITE0	FIFO 0 消息挂号中断使能 (FIFO message pending interrupt enable) 0: 当 CAN_RFF0.FFMP[1:0]位为非 0 时, 不产生中断; 1: 当 CAN_RFF0.FFMP[1:0]位不为 0 时, 产生中断。
0	TMEITE	发送邮箱空中断使能 (Transmit mailbox empty interrupt enable) 0: 当 CAN_TSTS.RPMx 位置位时, 不产生中断; 1: 当 CAN_TSTS.RPMx 位置位时, 产生中断。 注: 请参考 21.5 节 CAN 中断。

21.7.3.7 CAN 错误状态寄存器(CAN_ESTS)

偏移地址: 0x18

复位值: 0x0000 0000

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

位域	名称	描述
31:24	RXEC[7:0]	接收错误计数器（Receive error counter） 这个计数器按照 CAN 协议的故障界定机制的接收部分实现。按照 CAN 的标准，当接收出错时，根据出错的条件，该计数器加 1 或加 8；而在每次接收成功后，该计数器减 1，或当该计数器的值大于 127 时，设置它的值为 120。当该计数器的值超过 127 时，CAN 进入错误被动状态。
23:16	TXEC[7:0]	9 位发送错误计数器的低 8 位（Least significant byte of the 9-bit transmit error counter） 与上面相似，这个计数器按照 CAN 协议的故障界定机制的发送部分实现。
15:7	Reserved	保留，必须保持复位值
6:4	LEC[2:0]	上次错误代码（Last error code） 当 CAN 总线上检测到错误时，根据错误情况设置硬件。当正确发送或接收消息时，硬件将其值清除为“0”。 硬件不使用错误代码 7，软件可以设置此值，以便检测到代码更新。 000：没有错误； 001：位填充错误； 010：错误的格式（Form）； 011：确认（ACK）错误； 100：隐性错位（CAN 传输隐性但在总线上检测显性）； 101：显性错位（CAN 传输显性但在总线上检测到隐性）； 110：CRC 错误； 111：软件设置。
3	Reserved	保留，必须保持复位值
2	BOFFL	离线标志（Bus-off flag） 当总线关闭时，硬件设置该位。当传输错误计数器 CAN_TSTS.TXEC 溢出，即大于 255 时，CAN 总线关闭。请参阅 21.5.1 部分。
1	EPVFL	错误被动标志（Error passive flag） 当出错次数达到错误被动的阈值时，硬件对该位置‘1’。 （接收错误计数器或发送错误计数器的值>127）。
0	EWGFL	错误警告标志（Error warning flag） 当出错次数达到警告的阈值时，硬件对该位置‘1’。 （接收错误计数器或发送错误计数器的值≥96）。

21.7.3.8 CAN 位时序寄存器 (CAN_BTIM)

偏移地址: 0x1C 复位值: 0x0123 0000

注: 当 CAN 处于初始化模式时, 该寄存器只能由软件访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLM	LBM	Reserved				RSJW[1:0]		Reserved	TBS2[2:0]			TBS1[3:0]			
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
Reserved						BRTP[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31	SLM	静默模式（用于调试）（Silent mode（debug）） 0: 正常状态; 1: 静默模式。
30	LBM	回环模式（用于调试）（Loop back mode（debug）） 0: 禁止回环模式; 1: 允许回环模式。
29:26	Reserved	保留, 必须保持复位值
25:24	RSJW[1:0]	重新同步跳跃宽度（Resynchronization jump width） 为了重新同步, 该位域定义了 CAN 硬件在每位中可以延长或缩短多少个时间单元的上限。 $t_{RJW} = t_{CAN} \times (RSJW[1:0] + 1)$ 。
23	Reserved	保留, 必须保持复位值
22:20	TBS2[2:0]	时间段 2（Time segment 2） 该位域定义了时间段 2 占用了多少个时间单元 $t_{BS2} = t_{CAN} \times (TBS2[2:0] + 1)$ 。
19:16	TBS1[3:0]	时间段 1（Time segment 1） 该位域定义了时间段 1 占用了多少个时间单元 $t_{BS1} = t_{CAN} \times (TBS1[3:0] + 1)$ 有关位时间特性的详细信息, 请参阅第 21.4.7 节: 位时序。
15:10	Reserved	保留, 必须保持复位值
9:0	BRTP[9:0]	波特率分频器（Baud rate prescaler） 该位域定义了时间单元（tq）的时间长度 $t_q = (BRTP[9:0] + 1) \times t_{PCLK}$

21.7.4 CAN 邮箱寄存器

本节介绍发送和接收邮箱寄存器。有关寄存器映射的更多信息, 请参阅 21.4.6 部分: 消息存储。

21.7.4.1 发送邮箱标识寄存器(CAN_TMIx)(x=0..2)

偏移地址: 0x180, 0x190, 0x1A0

复位值：0xFFFF XXXX，X=未定义位（除了第0位，复位时 TXRQ=0）

注：1 当其所属的邮箱处在等待发送的状态时，该寄存器是写保护的；

2 该寄存器实现了发送请求控制功能（第0位）—复位值为0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STDID[10:0]/EXTID[28:18]											EXTID[17:13]				
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
EXTID[12:0]													IDE	RTRQ	TXRQ
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:21	STDID[10:0]/EXTID[28:18]	标准标识符或扩展标识符（Standard identifier or extended identifier） 依据 IDE 位的内容，这些位或是标准标识符，或是扩展身份标识的高字节。
20:3	EXTID[17:0]	扩展标识符（Extended identifier） 扩展身份标识的低字节。
2	IDE	标识符选择（Identifier extension） 该位决定发送邮箱中报文使用的标识符类型 0：使用标准标识符； 1：使用扩展标识符。
1	RTRQ	远程发送请求（Remote transmission request） 0：数据帧； 1：远程帧。
0	TXRQ	发送数据请求（Transmit mailbox request） 由软件对其置'1'，来请求发送邮箱的数据。当数据发送完成，邮箱为空时，硬件对其清'0'。

21.7.4.2 发送邮箱数据长度和时间戳寄存器 (CAN_TMDTx)(x=0..2)

当邮箱不在空置状态时，该寄存器的所有位为写保护。

偏移地址：0x184, 0x194, 0x1A4

复位值：未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MTIM[15:0]															
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
Reserved								TGT	Reserved				DLC[3:0]		
								rw					rw	rw	rw

位域	名称	描述
31:16	MTIM[15:0]	报文时间戳 (Message time stamp) 该域包含了, 在发送该报文 SOF 的时刻, 16 位定时器的值。
15:9	Reserved	保留, 必须保持复位值
8	TGT	发送时间戳 (Transmit global time) 该位仅在 CAN 处于时间触发通信模式时有效, 即 CAN_MCTRL.TTCM 位置位。 0: 不发送时间戳 CAN_TMDTx.MTIM[15:0]; 1: 发送时间戳 CAN_TMDTx.MTIM[15:0]。在长度为 8 的消息中, 时间戳 CAN_TMDTx.MTIM[15:0]是发送的最后两个字节; CAN_TMDTx.MTIM[7:0] 是第七个字节, CAN_TMDTx.MTIM[15:8]是第八个字节。它们替换了写入 CAN_TMDHx[31:16]的数据 (CAN_TMDLx.DATA6[7:0]和 CAN_TMDLx.DATA7[7:0])。为了发送时间戳的 2 个字节, DLC 必须编程为 8。
7:4	Reserved	保留, 必须保持复位值
3:0	DLC[3:0]	发送数据长度 (Data length code) 该域指定了数据报文的数据长度或者远程帧请求的数据长度。1 个报文包含 0 到 8 个字节数据, 而这由 DLC 决定。

21.7.4.3 发送邮箱低字节数据寄存器(CAN_TMDLx) (x=0..2)

当邮箱不在空置状态时, 该寄存器的所有位为写保护。

偏移地址: 0x188, 0x198, 0x1A8

复位值: 未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
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
DATA1[7:0]								DATA0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:24	DATA3[7:0]	数据字节 3 (Data byte 3) 报文的数据字节 3。
23:16	DATA2[7:0]	数据字节 2 (Data byte 2) 报文的数据字节 2。
15:8	DATA1[7:0]	数据字节 1 (Data byte 1) 报文的数据字节 1。
7:0	DATA0[7:0]	数据字节 0 (Data byte 0) 报文的数据字节 0。

位域	名称	描述
		注：报文包含 0 到 8 个字节数据，且从字节 0 开始。

21.7.4.4 发送邮箱高字节数据寄存器(CAN_TMDHx) (x=0..2)

当邮箱不在空置状态时，该寄存器的所有位为写保护。

偏移地址: 0x18c, 0x19c, 0x1ac

复位值: 未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
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
DATA5[7:0]								DATA4[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:24	DATA7[7:0]	数据字节 7 (Data byte 7) 报文的数据字节 7。 注意：如果 CAN_MCTRL.TTCM 位为“1”且此邮箱的 CAN_TMDTx.TGT 位也为“1”，则 DATA7 和 DATA6 将被时间戳替换。
23:16	DATA6[7:0]	数据字节 6 (Data byte 6) 报文的数据字节 6。
15:8	DATA5[7:0]	数据字节 5 (Data byte 5) 报文的数据字节 5。
7:0	DATA4[7:0]	数据字节 4 (Data byte 4) 报文的数据字节 4。

21.7.4.5 接收 FIFO 邮箱标识符寄存器(CAN_RMIx) (x=0..1)

偏移地址: 0x1B0, 0x1C0 复位值: 未定义

注：所有接收邮箱寄存器都是只读的。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STDID[10:0]/EXTID[28:18]											EXTID[17:13]				
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
EXTID[12:0]													IDE	RTRQ	TXRQ
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:21	STDID[10:0]/EXTID[28:18]	标准标识符或扩展标识符（Standard identifier or extended identifier） 根据 CAN_RMIx.IDE 位的内容，这些位要么是标准标识符，要么是扩展标识的高字节。
20:3	EXTID[17:0]	扩展标识符（Extended identifier） 扩展身份标识的低字节。
2	IDE	标识符选择（Identifier extension） 该位决定发送邮箱中报文使用的标识符类型 0：使用标准标识符； 1：使用扩展标识符。
1	RTRQ	远程发送请求（Remote transmission request） 0：数据帧； 1：远程帧。
0	Reserved	保留，必需保持复位值。

21.7.4.6 接收 FIFO 邮箱数据长度和时间戳寄存器(CAN_RMDTx)(x=0..1)

偏移地址: 0x1B4, 0x1C4

复位值: 未定义

注：所有接收邮箱寄存器都是只读的。

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

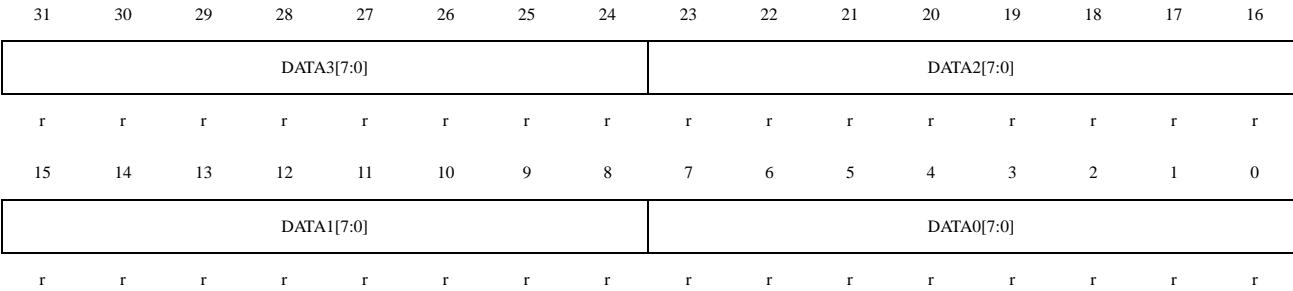
位域	名称	描述
31:16	MTIM[15:0]	报文时间戳（Message time stamp） 该域包含了，在捕获该报文 SOF 的时刻，16 位定时器的值。
15:8	FMI[7:0]	过滤器匹配序号（Filter match index） 这是存储在邮箱中的信息传输的过滤器序列号。有关标识符过滤的详细信息，请参阅21.4.5 部分：标识符过滤。
7:4	Reserved	保留，必须保持复位值
3:0	DLC[3:0]	接收数据长度（Data length code） 该域表明接收数据帧的数据长度（0~8）。对于远程帧请求，数据长度 DLC 恒为 0。

21.7.4.7 接收 FIFO 邮箱低字节数据寄存器(CAN_RMDLx)(x=0..1)

偏移地址: 0x1B8, 0x1C8

复位值: 未定义

注：所有接收邮箱寄存器都是只读的。



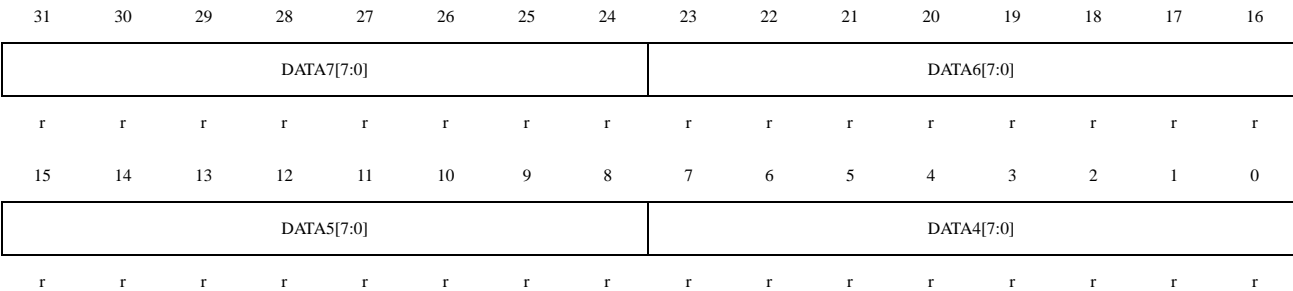
位域	名称	描述
31:24	DATA3[7:0]	数据字节 3（Data byte 3） 报文的数据字节 3。
23:16	DATA2[7:0]	数据字节 2（Data byte 2） 报文的数据字节 2。
15:8	DATA1[7:0]	数据字节 1（Data byte 1） 报文的数据字节 1。
7:0	DATA0[7:0]	数据字节 0（Data byte 0） 报文的数据字节 0。 注：报文包含 0 到 8 个字节数据，且从字节 0 开始。

21.7.4.8 接收 FIFO 邮箱低高字节数据寄存器(CAN_RMDHx) (x=0..1)

偏移地址: 0x1BC, 0x1CC

复位值: 未定义

注：所有接收邮箱寄存器都是只读的。



位域	名称	描述
31:24	DATA7[7:0]	数据字节 7（Data byte 7） 报文的数据字节 7。
23:16	DATA6[7:0]	数据字节 6（Data byte 6） 报文的数据字节 6。
15:8	DATA5[7:0]	数据字节 5（Data byte 5）

位域	名称	描述
		报文的数据字节 5。
7:0	DATA4[7:0]	数据字节 4 (Data byte 4) 报文的数据字节 4。

21.7.5 CAN 过滤器寄存器

21.7.5.1 CAN 过滤器主控制寄存器(CAN_FMC)

偏移地址: 0x200

复位值: 0x2A1C 0E01

注：该寄存器的非保留位完全由软件控制。

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

rw

位域	名称	描述
31:1	Reserved	保留，必须保持复位值
0	FINITM	过滤器初始化模式 (Filter init mode) 针对所有过滤器组的初始化模式设置。 0: 过滤器组工作在正常模式; 1: 过滤器组工作在初始化模式。

21.7.5.2 CAN 过滤器模式寄存器(CAN_FM1)

偏移地址: 0x204

复位值: 0x0000 0000

注：当设置 CAN_FMC.FINITM 位将滤波器置于初始化模式时，才能写入该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13:0	FBx	过滤器模式（Filter mode） 过滤器组 x 的工作模式。 0：CAN_FiRx 的两个 32 位寄存器工作在标识符屏蔽模式； 1：CAN_FiRx 的两个 32 位寄存器工作在标识符列表模式。

21.7.5.3 CAN 过滤器位宽寄存器(CAN_FS1)

偏移地址: 0x20C

复位值: 0x0000 0000

注：当设置 CAN_FMC.FINITM 位将滤波器置于初始化模式时，才能写入该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13:0	FSCx	过滤器位宽设置（Filter scale configuration） 过滤器组 x（13~0）的位宽。 0：过滤器位宽为 2 个 16 位； 1：过滤器位宽为单个 32 位。

21.7.5.4 CAN 过滤器 FIFO 关联寄存器 (CAN_FFA1)

偏移地址: 0x214

复位值: 0x0000 0000

注：当设置 CAN_FMC.FINITM 位将滤波器置于初始化模式时，才能写入该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13:0	FAFx	过滤器 x 的过滤器 FIFO 分配（Filter FIFO assignment for filter x） 报文在通过了某过滤器的过滤后，将被存放到其关联的 FIFO 中。 0：过滤器被关联到 FIFO0； 1：过滤器被关联到 FIFO1。

21.7.5.5 CAN 过滤器激活寄存器(CAN_FA1)

偏移地址: 0x21C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FAC13	FAC12	FAC11	FAC10	FAC9	FAC8	FAC7	FAC6	FAC5	FAC4	FAC3	FAC2	FAC1	FAC0	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13:0	FACx	过滤器激活（Filter active） 软件为某位设置“1”以激活相应的过滤器。只有在清除 CAN_FA1.FACx 位或设置 CAN_FMC.FINITM 位后，才能修改相应的滤波器寄存器 i(CAN_FiR[2:1])。 0：过滤器被禁用； 1：过滤器被激活。

21.7.5.6 CAN 过滤器 i 寄存器 x (CAN_FiRx) (i=0..13;x=1..2)

偏移地址: 0x240h, 0x31C

复位值: 未定义

注：14 组滤波器：i = 0 ... 13。每组滤波器由两个 32 位寄存器 CAN_FiR[2:1] 组成。

只有当相应的 CAN_FA1.FACx 位被清零或 CAN_FMC.FINIT 位被置位时，才能修改相应的过滤寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FBC31	FBC30	FBC29	FBC28	FBC27	FBC26	FBC25	FBC24	FBC23	FBC22	FBC21	FBC20	FBC19	FBC18	FBC17	FBC16
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
FBC15	FBC14	FBC13	FBC12	FBC11	FBC10	FBC9	FBC8	FBC7	FBC6	FBC5	FBC4	FBC3	FBC2	FBC1	FBC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:0	FBC[31:0]	过滤器位（Filter bits） 标识符模式 寄存器的每位对应于所期望的标识符的相应位的电平。 0：期望相应位为显性位； 1：期望相应位为隐性位。 屏蔽位模式 寄存器的每位指示是否对应的标识符寄存器位一定要与期望的标识符的相应位一致。 0：不关心，该位不用于比较； 1：必须匹配，到来的标识符位必须与滤波器对应的标识符寄存器位相一致。

注： 根据滤波器位宽和模式的不同设置，滤波器组中的两个寄存器的功能是不同的。关于过滤器的映射、功能描述和屏蔽寄存器的关联，请参见 21.4.5 章节：标识符过滤。

掩码模式中的掩码/标识符寄存器与**标识符列表模式**中的寄存器位定义相同。

滤波器组寄存器地址见表 21-4。

22 串行外设接口（SPI）

22.1 SPI 简介

SPI 可以工作在主模式或从模式，支持全双工和单工高速通讯模式，并且具有硬件 CRC 计算能力且可配置多主模式。

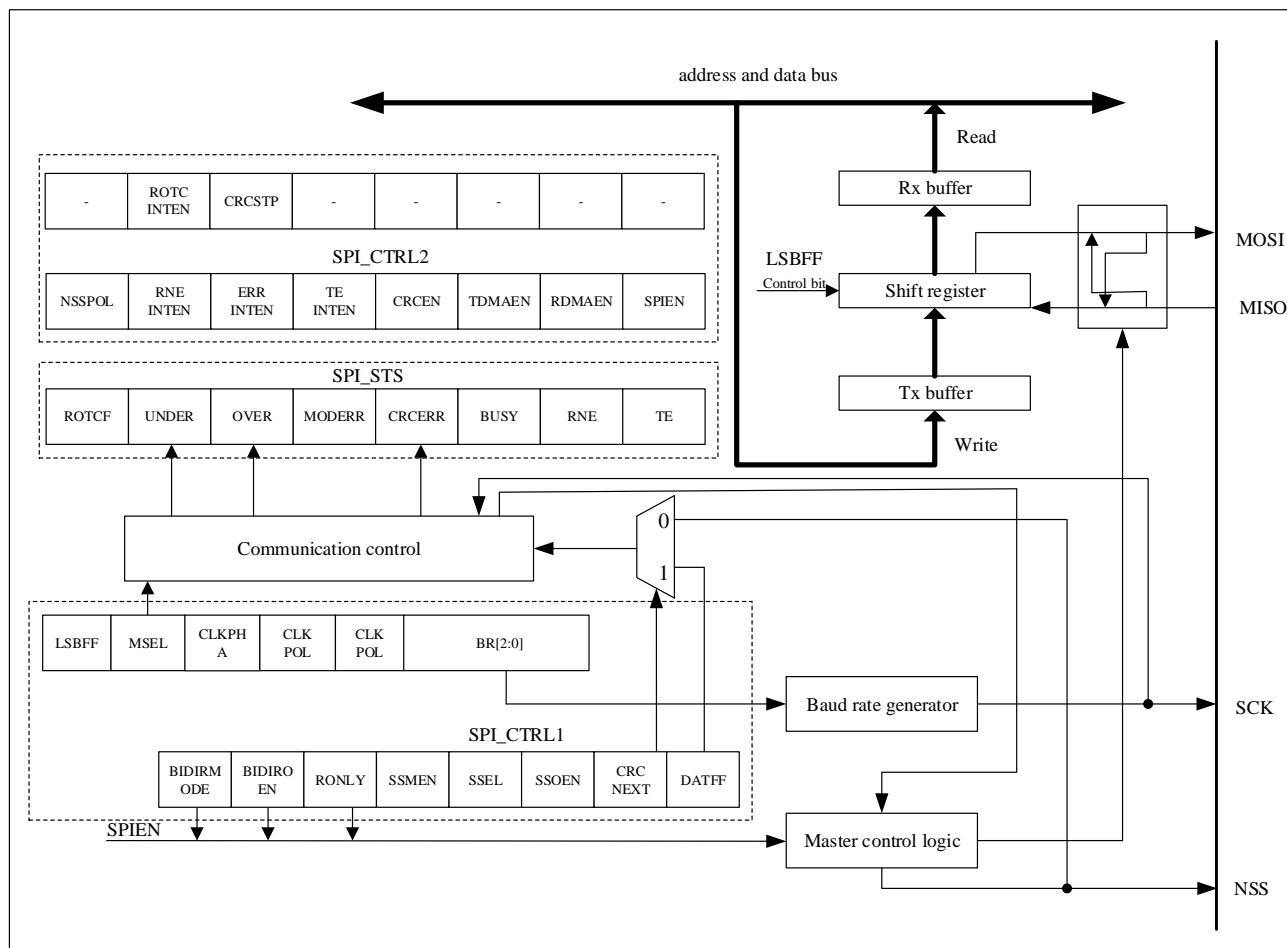
22.2 SPI 主要特性

- 全双工和单工同步模式
- 支持主模式、从模式和多主模式
- 支持 8bit 或 16bit 数据帧格式
- 数据位顺序可编程
- 硬件或软件片选管理
- 时钟极性和时钟相位可配置
- 发送和接收支持硬件 CRC 计算及校验
- 支持 DMA 传输功能

22.3 SPI 功能描述

22.3.1 通用描述

图 22-1 SPI 框图



为了连接外部设备，SPI 接口有 4 个引脚与外设器件连接，具体如下：

- SCLK: 串行时钟引脚，该信号从主设备 SCLK 引脚输出，由从设备 SCLK 引脚输入
- MISO: 主输入/从输出引脚，数据从主设备的 MISO 引脚输入，由从设备的 MISO 引脚输出
- MOSI: 主输出/从输入引脚，数据从主设备的 MOSI 引脚输出，由从设备的 MOSI 引脚输入
- NSS: 片选引脚，有两种 NSS 引脚类型，外部引脚和内部引脚。如果内部引脚检测到高电平，SPI 工作在主模式，相反，SPI 工作在从模式。用户可以使用主设备的一个标准 I/O 引脚控制从设备的 NSS 引脚

软件 NSS 模式

当 SPI_CTRL1.SSMEN = 1（图 22-2），软件从设备管理被使能。

软件 NSS 模式时，不需要使用 NSS 引脚。在这种模式下，通过写 SPI_CTRL1.SSEL 位（主模式 SPI_CTRL1.SSEL = 1，从模式 SPI_CTRL1.SSEL = 0），驱动内部 NSS 信号电平。

硬件 NSS 模式

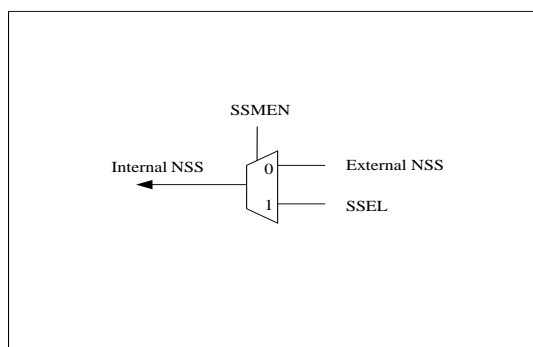
当 $SPI_CTRL1.SSMEN = 0$ （图 22-2），软件从设备管理被禁能。

NSS 输入模式：主设备的 NSS 输出被禁止（ $SPI_CTRL1.MSEL = 1, SPI_CTRL1.SSOEN = 0$ ），允许操作在多主模式下。在整个数据帧传输期间主机应该连接 NSS 到高电平，从机应该连接 NSS 到低电平。

NSS 输出模式：主设备的 NSS 输出被使能（ $SPI_CTRL1.MSEL = 1, SPI_CTRL1.SSOEN = 1$ ），主设备必须驱动 NSS 到低电平，所有与主设备连接并且设置为硬件 NSS 模式的设备将会检测到低电平，并自动进入从模式。当主设备的 NSS 没有被驱动到低电平，设备进入从模式，并产生主模式失效错误。

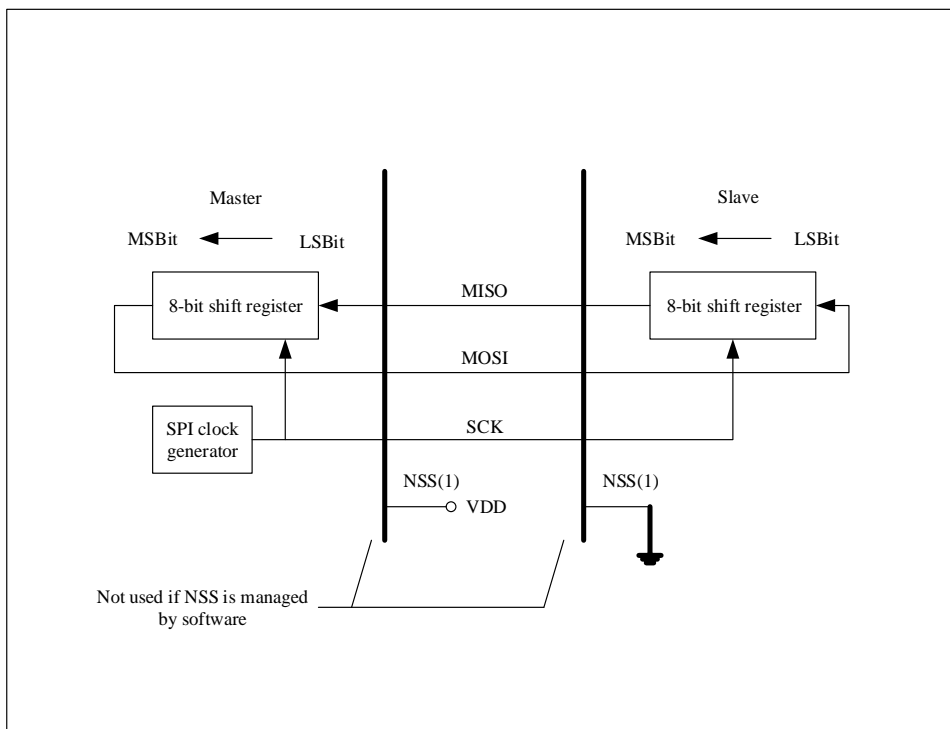
注：软件模式或硬件模式的选择，取决于通讯协议中是否需要 NSS 控制。如果不需要，可以选择软件模式，释放一个 GPIO 管脚另作他用。

图 22-2 硬件/软件的从选择管理



下图是一个单主和单从设备互连的例子。

图 22-3 单主和单从应用



注：NSS 引脚设置为输入

主设备通过 SCLK 引脚输出同步时钟信号，主设备的 MOSI 引脚连接到从设备的 MOSI 引脚，并且主设备

的 MISO 引脚连接到从设备的 MISO 引脚，以便数据可以在设备之间传输。主设备和从设备之间的连续数据传输，通过 MOSI 引脚发送数据到从设备，而从设备通过 MISO 引脚发送数据到主设备。

SPI 时序模式

通过设置 SPI_CTRL1.CLKPOL 位和 SPI_CTRL1.CLKPHA 位，用户可以选择数据捕获的时钟沿。

当 CLKPOL = 0, CLKPHA = 0，空闲时 SCLK 引脚将保持低电平，数据将在第一个时钟沿被采样，即上升沿。

当 CLKPOL = 0, CLKPHA = 1，空闲时 SCLK 引脚将保持低电平，数据将在第二个时钟沿被采样，即下降沿。

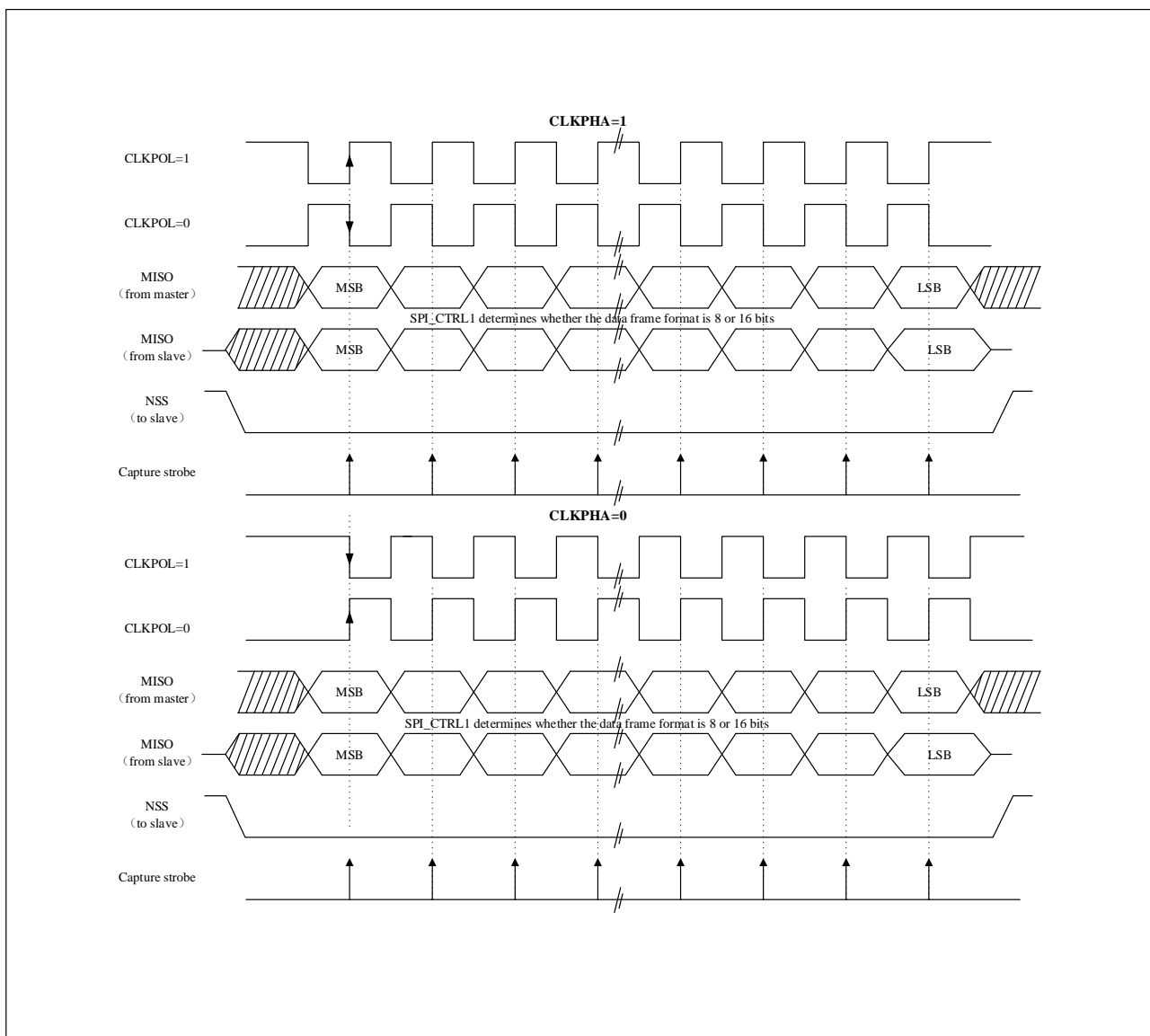
当 CLKPOL = 1, CLKPHA = 0，空闲时 SCLK 引脚将保持高电平，数据将在第一个时钟沿被采样，即下降沿。

当 CLKPOL = 1, CLKPHA = 1，空闲时 SCLK 引脚将保持高电平，数据将在第二个时钟沿被采样，即上升沿。

不管选择哪种时序模式，主设备和从设备的时序模式配置必须相同。

图 22-4 是当 SPI_CTRL1.LSBFF = 0 时，SPI 传输的 4 种 CLKPHA 和 CLKPOL 位组合时序。

图 22-4 数据时钟时序图



数据格式

通过设置 SPI_CTRL1.LSBFF 位，用户可以选择数据的位顺序，当 SPI_CTRL1.LSBFF=0，SPI 将先发送数据的高位（MSB），当 SPI_CTRL1.LSBFF=1，SPI 将先发送数据的低位（LSB）。

通过设置 SPI_CTRL1.DATFF 位，用户可以选择数据帧格式。

22.3.2 SPI 工作模式

■ 主机全双工模式（SPI_CTRL1.MSEL=1，SPI_CTRL1.BIDIROMODE=0，SPI_CTRL1.ONLY=0）

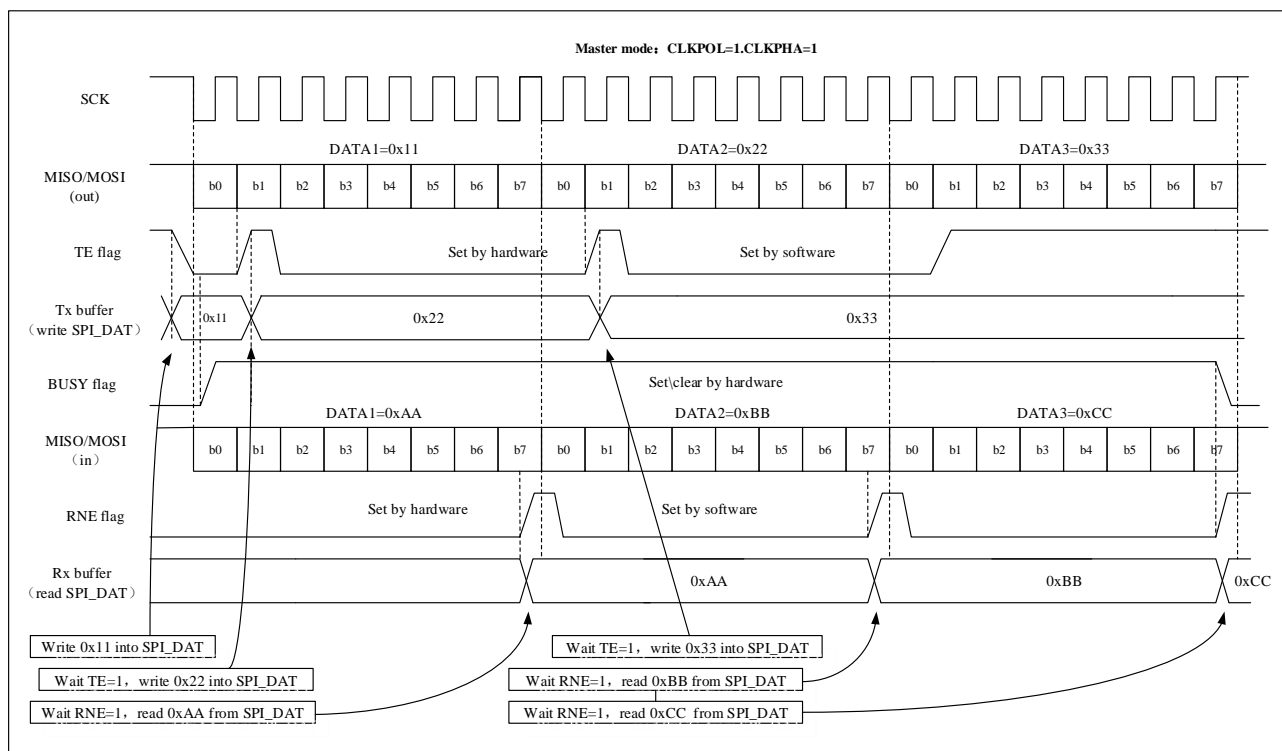
第一个数据被写到 SPI_DAT 寄存器后，将会开始传输，数据第一个位被发送时，数据字节并行从数据寄存器装载进入移位寄存器，然后数据位按照 SPI_CTRL1.LSBFF 位的配置，数据位按照 MSB 或 LSB 顺序被串行移位进入 MOSI 引脚。与此同时，在 MISO 引脚上接收到的数据，按照同样顺序被串行地移位进入移位寄存器，然后并行装载入 SPI_DAT 寄存器。

1. 设置 SPI_CTRL2.SPIEN 位为 1，使能 SPI 模块；

2. 写待发送的第一个数据到SPI_DAT（这个写操作会清除SPI_STS.TE标志位）；
3. 等待SPI_STS.TE标志位置1后，再写入第二个待发送的数据到SPI_DAT寄存器，等待SPI_STS.RNE标志位置1后，读取SPI_DAT寄存器获得第一个接收的数据，读取SPI_DAT寄存器，SPI_STS.RNE标志位会清0。重复上述操作，发送后续的数据，同时接收第n-1个数据；
4. 等待SPI_STS.RNE置1后，读取最后一个数据；
5. 等待SPI_STS.TE标志位置1，等待SPI_STS.BUSY标志位清除后再关闭SPI模块。

数据的发送和接收处理可以在 SPI_STS.RNE 标志位或 SPI_STS.TE 标志位的上升沿产生的中断处理程序中实现。

图 22-5 主机全双工模式下连续传输时，SPI_STS.TE/RNE/BUSY 的变化示意图



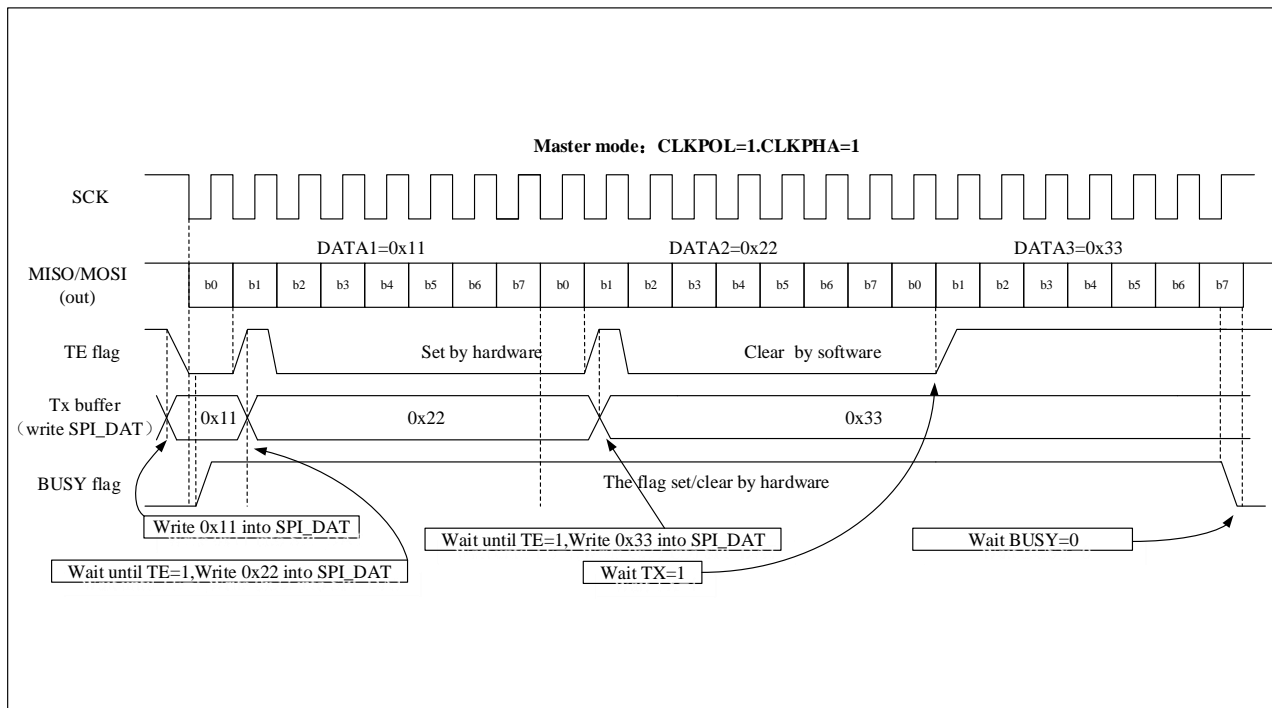
■ 主机双线单向仅发送模式 (SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.ONLY = 0)

双线单向仅发送模式与全双工模式硬件配置一致，但是在双线单向仅发送模式时，接收的数据将不会被读取，因此 SPI_STS.OVER 标志位将会置位，软件应该忽略这个位。软件操作流程如下（图 22-6 主机单向只发送模式下连续传输时，SPI_STS.TE/BUSY 变化示意图）：

1. 设置SPI_CTRL2.SPIEN位为1，使能SPI模块；
2. 写待发送的第一个数据到 SPI_DAT 寄存器（该操作会清除 SPI_STS.TE 标志位）；
3. 等待 SPI_STS.TE 标志位置 1，写待发送的第二个数据到 SPI_DAT 寄存器，重复这个操作发送后续的数据；
4. 写最后一个数据到 SPI_DAT 寄存器，等待 SPI_STS.TE 标志位置 1，然后等待 SPI_STS.BUSY 位清除，完成所有数据的发送。

数据发送可以在 SPI_STS.TE 标志位上升沿产生的中断处理程序里实现。

图 22-6 主机单向只发送模式下连续传输时，SPI_STS.TE/BUSY 变化示意图



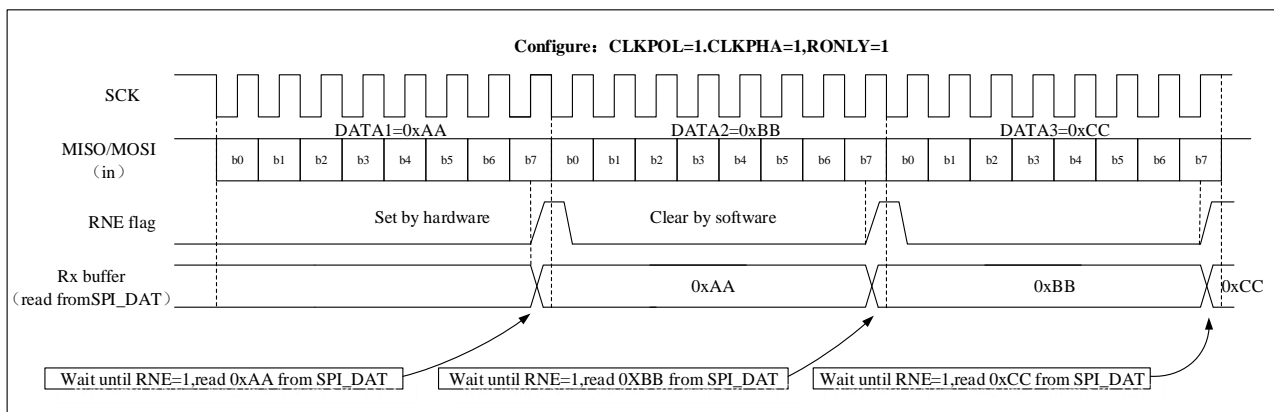
■ 主机双线单向仅接收模式（SPI_CTRL1.MSEL = 1，SPI_CTRL1.BIDIRMODE = 0，SPI_CTRL1.ROONLY = 1）

当 SPI_CTRL2.SPIEN = 1，开始接收过程。来自 MISO 引脚的数据位依次连续移位进入移位寄存器，然后并行传送数据到 SPI_DAT 寄存器。软件操作流程如下（见图 22-7）：

1. 设置 SPI_CTRL1.ROONLY = 1，使能仅接收模式；
2. 主机模式下，设置 SPIEN 位为 1，使能 SPI 模块，SCLK 信号会立即产生，在 SPI 关闭前（SPIEN=0），数据连续被接收。从机模式下，当主设备驱动 NSS 信号低电平并且产生 SCLK，数据持续被接收；
3. 等待 SPI_STS.RNE 位置 1，读取 SPI_DAT 寄存器获得接收的数据，当读取 SPI_DAT 寄存器，SPI_STS.RNE 位将会清除。重复这个操作接收所有数据。

数据处理可以在 SPI_STS.RNE 标志位产生的中断处理程序里实现。

图 22-7 只接收模式（BIDIRMODE=0 并且 RONLY=1）下连续传输时，RNE 变化示意图



■ 主机单线双向发送模式（SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 1, SPI_CTRL1.BIDIROEN = 1, SPI_CTRL1.RONLY = 0）

数据写进 SPI_DAT 寄存器后，传输过程开始。这个模式不接收数据。发送第一个数据位的同时，被发送的数据并行装载进移位寄存器，然后根据 LSBFF 位的配置，SPI 按照 MSB 或 LSB 顺序将数据位串行移位到 MOSI 引脚。

主机单线双向发送的软件操作流程和仅发送模式的流程相同。

■ 主机单线双向接收模式（SPI_CTRL1.MSEL = 1, SPI_CTRL1.BIDIRMODE = 1, SPI_CTRL1.BIDIROEN = 0, SPI_CTRL1.RONLY = 0）

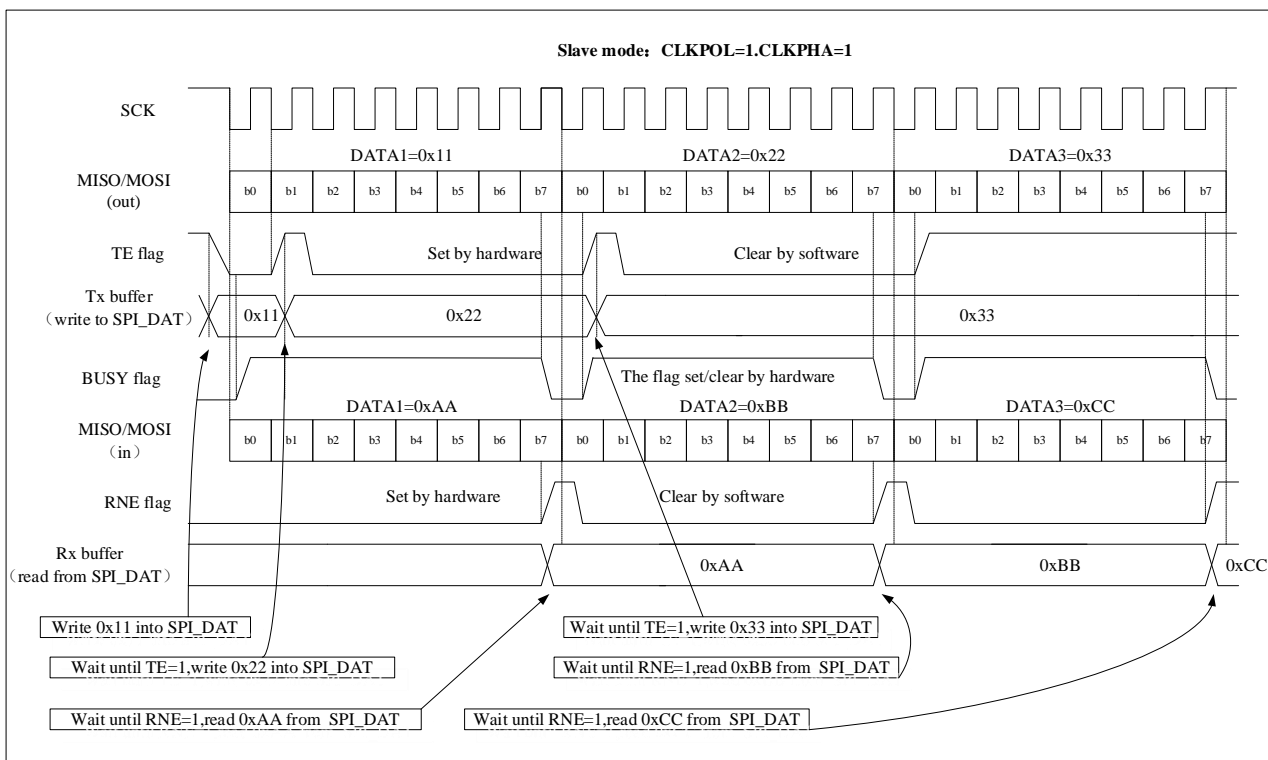
该模式下，当 SPI 使能（SPI_CTRL2.SPIEN = 1），接收过程开始。该模式下，没有数据输出，接收到的数据位顺序且连续移位进入移位寄存器，并行的传输进 SPI_DAT 寄存器（接收缓存）。

主机单线双向接收模式的软件操作流程和仅接收模式一样。

■ 从机全双工模式（SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 0）

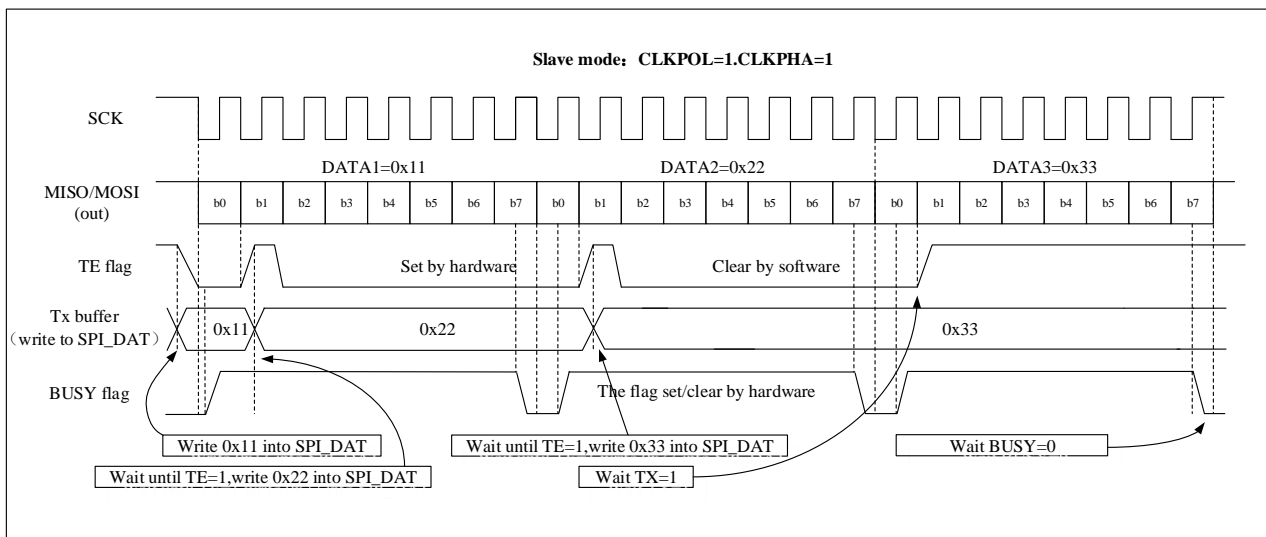
当从设备接收到第一个时钟沿，数据传输过程开始。主设备开始数据传输之前，软件必须确保待发送的数据写入 SPI_DAT 寄存器。

图 22-8 从机全双工模式下连续传输时，SPI_STS.TE/RNE/BUSY 的变化示意图



■ 从机双线单向仅发送模式 (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 0)

图 22-9 从机单向只发送模式下连续传输时，SPI_STS.TE/BUSY 变化示意图



■ 从机双线单向仅接收模式 (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 0, SPI_CTRL1.RONLY = 1)

当从设备接收到时钟信号和来自 MOSI 引脚的第一个数据位，数据接收过程开始。接收到的数据位顺序且连续地串行移位到移位寄存器，然后并行地装载到 SPI_DAT 寄存器（接收缓存）。

■ 从机单线双向发送模式 (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 1,

SPI_CTRL1.BIDIROEN = 1)

当从设备接收到第一个时钟沿，数据发送过程开始。该模式没有数据接收，SPI 主机开始数据传输前，软件必须确保待发送数据已经被写进 SPI_DAT 寄存器。

■ 从机单线双向接收模式 (SPI_CTRL1.MSEL = 0, SPI_CTRL1.BIDIRMODE = 1, SPI_CTRL1.BIDIROEN = 0)

当从设备接收到第一个时钟沿和来自 MOSI 引脚的数据位时，数据接收开始。该模式没有数据输出，接收到的数据位顺序且连续地串行移位到移位寄存器，然后并行地装载到 SPI_DAT 寄存器（接收缓存）。

注意：从机的软件操作流程参考主机的。

SPI 初始化流程

1. 通过设置 SPI_CTRL1.BR[2:0]位配置数据传输的波特率（如果工作在从模式，则忽略该步骤）；
2. 选择时钟极性（SPI_CTRL1.CLKPOL）和时钟相位（SPI_CTRL1.CLKPHA），定义数据传输和时钟的相位关系；
3. 设置 SPI_CTRL1.DATFF 位定义帧格式为 8bit 还是 16bit；
4. 配置 SPI_CTRL1.LSBFF 定义数据位发送的顺序是 LSB 还是 MSB；
5. 配置 NSS 模式；
6. 配置 SPI_CTRL1.MSEL、SPI_CTRL1.BIDIRMODE、SPI_CTRL1.BIDIROEN 和 SPI_CTRL1.ONLY 位；
7. 设置 SPI_CTRL2.SPIEN 位使能 SPI 模块。

SPI 协议基本的发送和接收处理

当 SPI 发送 1 个数据帧，首先，数据帧从数据缓存装载进移位寄存器，然后装载的数据被发送。当来自发送缓存的数据传输进移位寄存器，发送缓存器为空，SPI_STS.TE 标志位置 1，然后下一个数据可装载进入发送缓存。如果 SPI_CTRL2.TEINTEN 位置 1，中断将会产生。写 SPI_DAT 寄存器可以对 SPI_STS.TE 标志位清 0。

采样时钟的最后一个边沿，当数据从移位寄存器传输进接收缓存，SPI_STS.RNE 标志位置 1，数据准备就绪，可以从 SPI_DAT 寄存器读取。如果 SPI_CTRL2.RNEINTEN 标志位置 1，中断将会产生。读 SPI_DAT 寄存器可以对 SPI_STS.RNE 标志位清 0。

主模式下，当数据写进发送缓存，发送过程开始。当前数据帧发送完成前，如果下个数据写进 SPI_DAT 寄存器，连续发送可以实现。

从机模式下，NSS 引脚为低，当第一个时钟沿到来，发送过程开始。为了避免意外的数据传输，数据发送前（主机发送时钟前，建议先使能 SPI 模块）软件必须写数据到发送缓存。

在有些配置里，当发送最后数据时，SPI_STS.BUSY 标志位可以用于等待数据发送结束。

连续和非连续传输

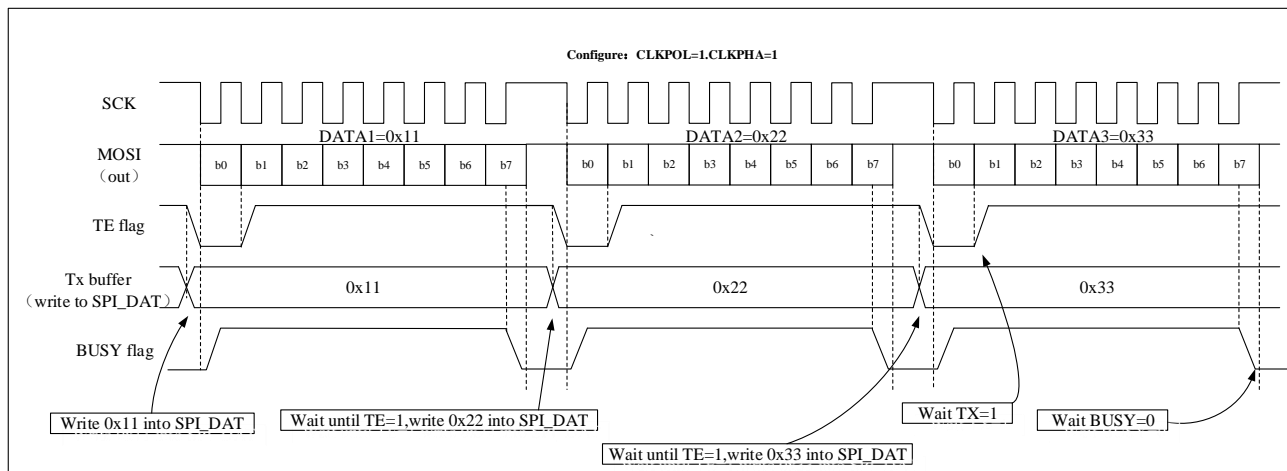
当主模式下发送数据，如果软件足够快，检测到每个 TE 上升沿（或 TE 中断），且正进行的传输结束前，立即将数据写入 SPI_DAT 寄存器，此时，每个数据项之间的 SPI 时钟保持连续，SPI_STS.BUSY 标志位将不会被清除，连续通讯可以实现。

如果软件不够快，将导致不连续的通讯，此时，每个数据传输之间 SPI_STS.BUSY 标志位会被清除（图 22-10 BIDIRMODE = 0, ONLY = 0 非连续传输发送时，SPI_STS.TE/BUSY 变化示意图）。

主机仅接收模式下（SPI_CTRL1.ONLY = 1），通讯总是连续的，并且 BUSY 标志位总是为高。

从模式下，通讯的连续性由主设备决定，任何情况下，即使通讯是连续的，每个数据项之间，BUSY 标志位将至少有 1 个 SPI 时钟周期为低（见图 22-9）。

图 22-10 BIDIRMODE = 0, ONLY = 0 非连续传输发送时，SPI_STS.TE/BUSY 变化示意图



22.3.3 状态标志

SPI_STS 寄存器中有以下 3 个标志位，用以监视 SPI 总线的状态。

发送缓存空标志位（TE）

当发送缓存空，TE 标志位置 1，意味着可以将新数据写进 SPI_DAT 寄存器。当发送缓存非空，该标志位将硬件清 0。

接收缓存非空标志位（RNE）

当接收缓存非空，RNE 标志位置 1，因此用户知道接收缓存有数据。读取 SPI_DAT 寄存器后，该标志位将硬件清 0。

忙标志位（BUSY）

当传输开始，BUSY 标志位置 1，传输结束后 BUSY 标志位硬件清 0。

仅当设备在主机单线双向接收模式，当通讯进行中，BUSY 标志位将会设置为 0。

下面情况，BUSY 标志位将会清 0：

- 传输结束（主模式下连续通信的情况除外）；
- 关闭 SPI 模块（SPI_CTRL2.SPIEN = 0）；
- 产生主模式失效（SPI_STS.MODERR = 1）。

当通讯是不连续的：每个数据项传输之间，BUSY 标志位清 0。

当通讯是连续的：在主机模式，整个传输过程，BUSY 标志位保持为高。在从机模式，每个数据项传输之间 BUSY 标志位会有 1 个 SPI 时钟周期为低。因此不要使用 BUSY 标志位处理每个数据项的发送和接收。

22.3.4 关闭 SPI

为了关闭 SPI 模块，不同的操作模式需要采用不同的操作步骤：

在主机或从机全双工模式

1. 等待 SPI_STS.RNE 标志位置 1，并且接收到最后一个字节；
2. 等待 SPI_STS.TE 标志位置 1；
3. 等待 SPI_STS.BUSY 标志位清 0；
4. 关闭 SPI 模块（SPI_CTRL2.SPIEN = 0）。

在主机或从机单向发送模式

1. 向 SPI_DAT 寄存器写完最后一个字节后，等待 SPI_STS.TE 标志位置 1；
2. 等待 SPI_STS.BUSY 标志位清 0；
3. 关闭 SPI 模块（SPI_CTRL2.SPIEN = 0）。

在主机单向只接收模式

1. 等待倒数第二个 SPI_STS.RNE 置 1；
2. 关闭 SPI 模块前（SPI_CTRL2.SPIEN = 0），等待 1 个 SPI 时钟周期（使用软件延时）；
3. 进入关机模式前（或关闭 SPI 模块时钟），等待最后一个 SPI_STS.RNE 置 1。

从机单向只接收模式

1. 可以在任意时间关闭 SPI 模块（SPI_CTRL2.SPIEN = 0），并且当前传输结束后，SPI 模块将被关闭；
2. 如果想进入关机模式，进入关机模式之前（或关闭 SPI 模块时钟），必须等待 SPI_STS.BUSY 标志位为 0。

22.3.5 使用 DMA 进行 SPI 通讯

用户可以选择 DMA 进行 SPI 数据传输，应用程序可以得到释放，系统效率可以大大提升。

当发送缓存 DMA 使能（SPI_CTRL2.TDMAEN 位置 1），每次 SPI_STS.TE 标志位置 1，会产生 DMA 请求，DMA 自动将数据写入 SPI_DAT 寄存器，这将会清除 TE 标志位。当接收缓存 DMA 使能（SPI_CTRL2.RDMAEN 位置 1），每次 SPI_STS.RNE 标志位置 1，会产生 DMA 请求，DMA 自动读取 SPI_DAT 寄存器，这将会清除 SPI_STS.RNE 标志位。

当 SPI 仅用于数据发送，仅需要使能 SPI 的发送 DMA 通道（SPI_CTRL2.TDMAEN 位置 1）。此时，由于接收到的数据还没有被读取，OVER 标志被设置为“1”（软件不需要注意这个标志）。

当 SPI 仅用于数据接收，仅需要使能 SPI 的接收 DMA 通道（SPI_CTRL2.RDMAEN 位置 1）。

在发送模式，DMA 已经发送完所有待发送的数据后（DMA_INTSTS.TXCF 标志位变为 1），SPI_STS.BUSY 标志位可以被监视确认 SPI 通讯结束，这样可以避免当 SPI 关闭或进入停机模式，破坏最后数据的发送。因此，软件需要等待 SPI_STS.TE 标志位置 1，并且等待 SPI_STS.BUSY 标志位为 0。

图 22-11 使用 DMA 发送

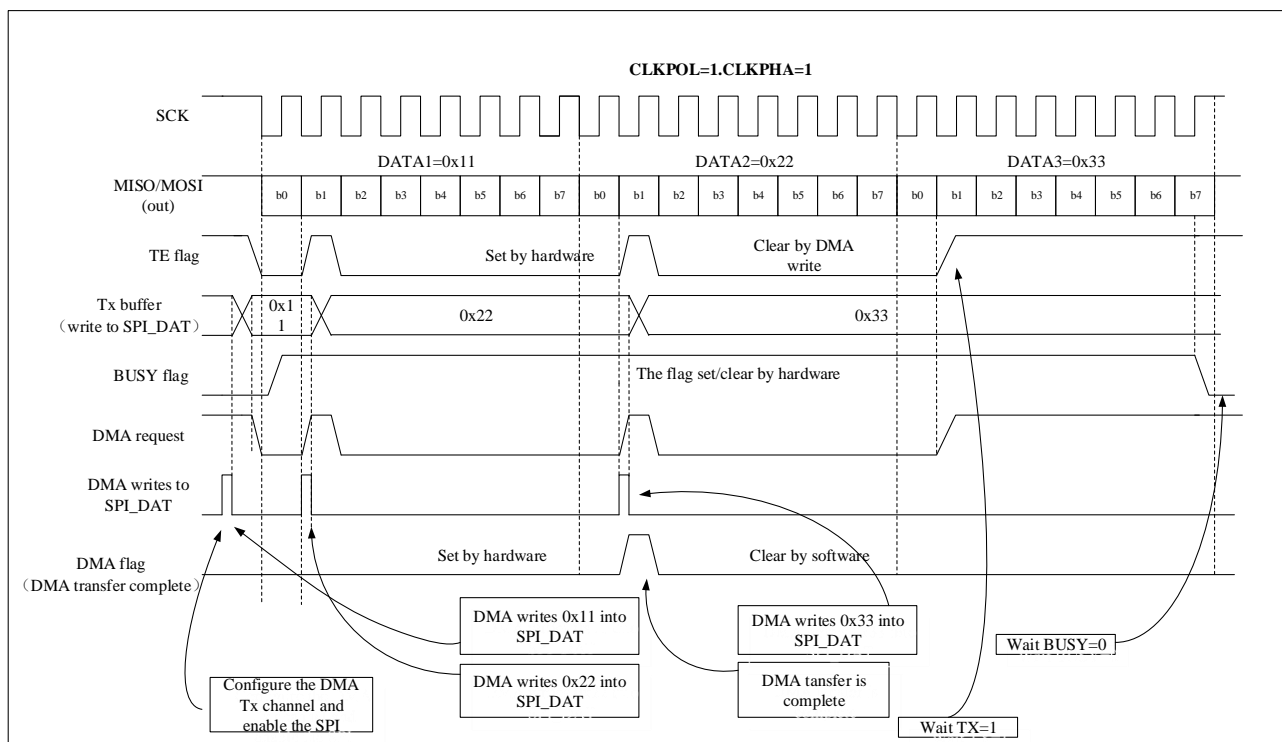
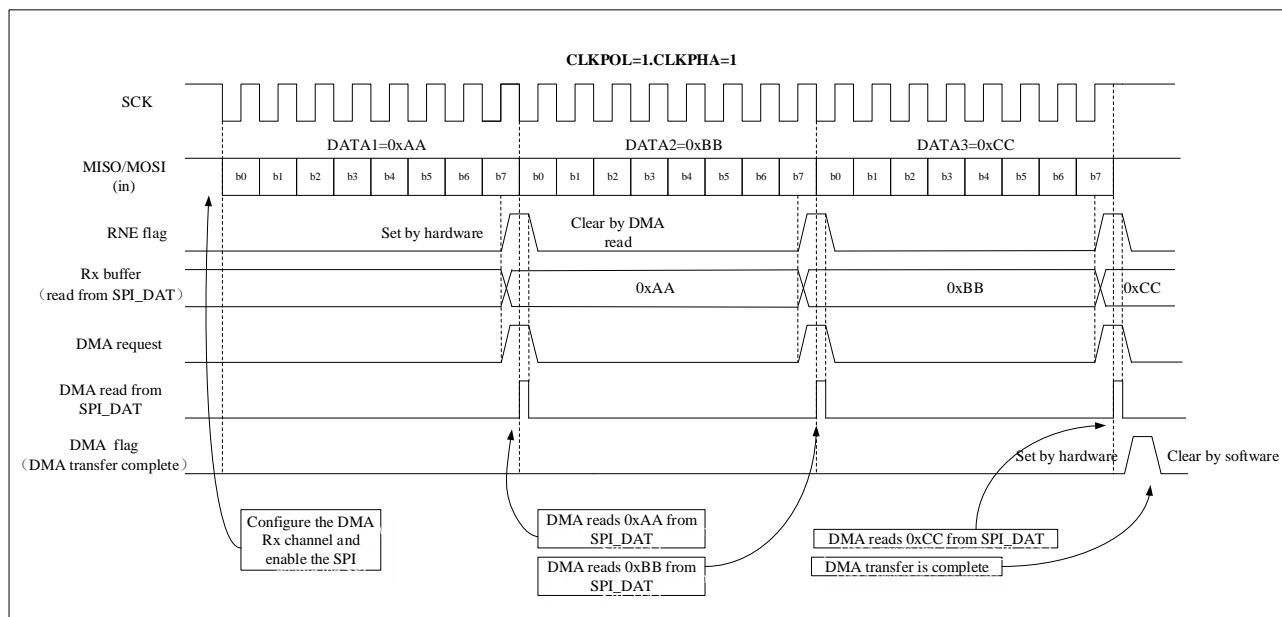


图 22-12 使用 DMA 接收



22.3.6 CRC 计算

SPI 包含两个独立的 CRC 计算器，用于数据发送和数据接收，以确保数据传输的正确性。根据发送和接收数据帧格式，CRC 采用不同的计算方法，8 位数据帧格式采用 CRC8，16 位数据帧格式采用 CRC16。SPI CRC 计算使用的多项式由 SPI_CRCPOLY 寄存器设置，用户设置 SPI_CTRL2.CRCEN 位使能 CRC 计算。

在发送模式，最后的数据写进发送缓存后，设置 SPI_CTRL1.CRCNEXT 位为 1，这指示发送完数据后硬件

将开始发送 CRC 值（SPI_CRCTDAT 值）。发送 CRC 时，CRC 计算将停止。

在接收模式，倒数第二个数据帧接收到后，设置 SPI_CTRL1.CRCNEXT 位为 1。接收到的 CRC 和 SPI_CRCRDAT 值进行比较，如果他们不同，SPI_STS.CRCERR 位置 1，当 SPI_CTRL2.ERRINTEN 位置 1，中断产生。

为了保持主设备-从设备下次的 CRC 计算结果的同步，用户应清除主设备-从设备的 CRC 值。SPI_CTRL2.CRCEN 位置 1 会复位 SPI_CRCRDAT 寄存器和 SPI_CRCTDAT 寄存器。按顺序采用下面步骤：SPI_CTRL2.SPIEN = 0；SPI_CTRL2.CRCEN = 0；SPI_CTRL2.CRCEN = 1；SPI_CTRL2.SPIEN = 1。

最重要的是，当 SPI 配置为从模式且 CRC 使能，只要 SCLK 引脚有时钟脉冲，即使 NSS 引脚为高，CRC 计算将仍然被执行。这种情况常见于当主设备和多个从设备交替通讯时，因此这需要避免 CRC 误操作。

当 SPI 硬件 CRC 检查使能（SPI_CTRL2.CRCEN = 1）且 DMA 使能，通讯结束时硬件自动完成 CRC 字节发送和接收。

22.3.7 错误标志位

主模式失效错误（MODERR）

以下两种情况下会发生主模式失效错误：

- NSS 引脚硬件管理模式，主设备 NSS 引脚被驱动低电平；
- NSS 引脚软件模式管理，SSEL 位被置 0。
- 当主模式失效错误发生，SPI_STS.MODERR 标志位置 1。如果用户允许相应的中断，则产生中断。SPI_CTRL2.SPIEN 位和 SPI_CTRL1.MSEL 将写保护，且硬件清除。SPI 关闭且强制进入从模式。
- 软件执行 SPI_STS 寄存器读或写操作，然后写 SPI_CTRL1 寄存器可以清除 SPI_STS.MODERR 位（在多主配置下，主机的 NSS 引脚必须先拉高）。
- 通常，从机的 SPI_STS.MODERR 位不能设置为 1。然而，在多主配置下，从设备的 SPI_STS.MODERR 位可能置位。这种情况下，SPI_STS.MODERR 位指示存在多主冲突。中断程序可以执行复位或返回默认状态从错误状态恢复。

上溢标志位（OVER）

当 SPI_STS.RNE 位置 1，但是仍然有数据发送进入接收缓存，上溢错误将发生，此时，上溢标志 SPI_STS.OVER 置 1。如果用户使能相应的中断，则产生中断。所有接收到的数据丢失，且 SPI_DAT 寄存器仅保留之前未读的数据。

依次读 SPI_DAT 寄存器和 SPI_STS 寄存器可以清除 SPI_STS.OVER 位。

CRC 错误（CRCERR）

CRC 错误标志用于检查接收数据的有效性。当接收到的 CRC 值和 SPI_CRCRDAT 值不匹配，CRC 错误发生。此时，SPI_STS.CRCERR 标志位被设置为“1”，如果用户使能相应的中断（SPI_CTRL2.ERRINTEN = 1），则产生中断。

22.3.8 SPI 中断

表 22-1 SPI 中断请求

中断事件	事件标志位	使能控制位
发送缓存空标志	TE	TEINTEN
接收缓存非空标志	RNE	RNEINTEN
主模式失效事件	MODERR	ERRINTEN
溢出错误	OVER	
CRC 错误标志	CRCERR	

22.4 SPI 寄存器描述

22.4.1 SPI 寄存器总览

表 22-2 SPI 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000h	SPI_CTRL1	Reserved																BIDIROEN	BIDIRMODE	RONLY	SSMEN	SSEL	SSOEN	CRCNEXT	DATFF	LSBFF	MSEL	CLKPHA	CLKPOL	Reserved	BR[2:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0
004h	SPI_CTRL2	Reserved																ROTCINTEN	CRCSTP	Reserved						NSSPOL	ERRINTEN	RNEINTEN	TEINTEN	CRCEN	TDMAEN	RDMAEN	SPIEN	
	Reset Value																	0	0							0	0	0	0	0	0	0	0	0
008h	SPI_STS	Reserved																ROTCF	Reserved						OVER	MODERR	CRCERR	BUSY	RNE	TE				
	Reset Value																		0							0	0	0	0	1	0			
00Ch	SPI_DAT	Reserved																DAT[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	SPI_CRCTDAT	Reserved																CRCTDAT[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	SPI_CRCRDAT	Reserved																CRCRDAT[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	SPI_CRCPOLY	Reserved																CRCPOLY[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
038h	SPI_CTRL3	Reserved																										DELAYTIME[3:0]						
	Reset Value																											0	0	0	0			

22.4.2 SPI 控制寄存器 1 (SPI_CTRL1)

地址偏移: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIR MODE	BIDIR OEN	RONLY	SSMEN	SSEL	SSOEN	CRC NEXT	DATFF	LSBFF	MSEL	CLKPHA	CLKPOL	Reserved	BR[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15	BIDIRMODE	双向数据模式使能 0: 选择“双线双向”模式; 1: 选择“单线双向”模式。
14	BIDIROEN	“单线双向”模式下输出使能 该位和 SPI_CTRL1.BIDIRMODE 位一起决定“单线双向”模式的传输方向。 0: 输出禁止（仅接收模式）。 1: 输出使能（仅发送模式）。 在主机模式下，“单线”数据线是 MOSI 引脚，在从机模式下，“单线”数据线是 MISO 引脚。
13	RONLY	“双线双向”模式下输出使能 该位和 SPI_CTRL1.BIDIRMODE 位一起决定“双线双向”模式的传输方向。在多个从设备的应用场景，该位仅被访问的从设备置 1，仅被访问的从设备可以输出，从而避免数据线冲突。 0: 全双工（发送和接收模式）。 1: 输出禁能（仅接收模式）。
12	SSMEN	软件从设备管理 当 SPI_CTRL1.SSMEN 被置位时，NSS 引脚上的电平由 SPI_CTRL1.SSEL 位的值决定。 0: 禁止软件从设备管理; 1: 启用软件从设备管理。
11	SSEL	内部从设备选择 该位仅在 SPI_CTRL1.SSMEN 位置 1 时有意义。它决定了 NSS 电平，且 NSS 引脚的 I/O 操作无效。
10	SSOEN	NSS 输出使能 0: 禁止在主模式下 NSS 输出，该设备可以工作在主设备模式; 1: 设备开启时，开启主模式下 NSS 输出，该设备不能工作在主设备模式。
9	CRCNEXT	下一个发送 CRC 0: 下一个发送的值来自发送缓存。

位域	名称	描述
		1: 下一个发送的值来自发送 CRC 寄存器。 <i>注意: 在 SPI_DAT 寄存器写入最后一个数据后应马上设置该位。</i>
8	DATFF	数据帧格式 0: 使用 8 位数据帧格式进行发送/接收; 1: 使用 16 位数据帧格式进行发送/接收。 <i>注意: 只有当 SPI 禁止 (SPI_CTRL2.SPIEN = 0) 时, 才能写该位, 否则出错。</i>
7	LSBFF	帧格式 0: 先发送 MSB。 1: 先发送 LSB。 <i>注意: 通讯过程中该位不能被改变。</i>
6	MSEL	主设备选择 0: 配置为从设备; 1: 配置为主设备。 <i>注意: 当通信正在进行的时候, 不能修改该位。</i>
5	CLKPHA	时钟相位 0: 第一个时钟沿采样数据 1: 第二个时钟沿采样数据。 <i>注意: 当通信正在进行的时候, 不能修改该位。</i>
4	CLKPOL	时钟极性 0: 空闲状态时, SCLK 保持低电平; 1: 空闲状态时, SCLK 保持高电平。 <i>注意: 当通信正在进行的时候, 不能修改该位。</i>
3	Reserved	保留, 必需保持复位值。
2:0	BR[2:0]	波特率控制 000: $f_{PCLK}/2$ 001: $f_{PCLK}/4$ 010: $f_{PCLK}/8$ 011: $f_{PCLK}/16$ 100: $f_{PCLK}/32$ 101: $f_{PCLK}/64$ 110: $f_{PCLK}/128$ 111: $f_{PCLK}/256$ <i>注意: 当通信正在进行的时候, 不能修改这些位。</i>

22.4.3 SPI 控制寄存器 2 (SPI_CTRL2)

地址偏移: 0x04

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ROTC INTEN	CRCSTP	Reserved					NSSPOL	ERR INTEN	RNE INTEN	TE INTEN	CRCEN	TDMAEN	RDMAEN	SPIEN
rw		rw						rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:15	Reserved	保留，必须保持复位值
14	ROTCINTEN	只收模式下，传输完成中断使能 0：传输完成不使能 1：在传输完成使能
13	CRCSTP	NSS 失效后，CRC 计算控制 0：停止计算 1：在时钟存在的情况下，仍然进行计算
12:8	Reserved	保留，必须保持复位值
7	NSSPOL	NSS 极性控制 0：NSS 低电平有效； 1：NSS 高电平有效。
6	ERRINTEN	错误中断使能 当错误（SPI_STS.CRCERR、SPI_STS.OVER、SPI_STS.MODERR）产生时，该位控制是否产生中断 0：禁止错误中断； 1：允许错误中断。
5	RNEINTEN	接收缓存非空中断使能 0：禁止 RNE 中断； 1：允许 RNE 中断，当 SPI_STS.RNE 标志置位时产生中断请求。
4	TEINTEN	发送缓存空中断使能 0：禁止 TE 中断； 1：允许 TE 中断，当 SPI_STS.TE 标志置位为‘1’时产生中断请求。
3	CRCEN	硬件 CRC 校验使能 0：禁止 CRC 计算； 1：启动 CRC 计算。 该位只能用于全双工模式。 <i>注意：只有在禁止 SPI 时（SPI_CTRL2.SPIEN = 0），才能写该位，否则出错。</i>
2	TDMAEN	发送缓存 DMA 使能 当该位被设置时，TE 标志一旦被置位就发出 DMA 请求 0：禁止发送缓存 DMA； 1：启动发送缓存 DMA。
1	RDMAEN	接收缓存 DMA 使能 当该位被设置时，RNE 标志一旦被置位就发出 DMA 请求 0：禁止接收缓存 DMA； 1：启动接收缓存 DMA。

位域	名称	描述
0	SPIEN	SPI 使能 0: 禁能 SPI 模块。 1: 使能 SPI 模块。 <i>注意: 当关闭 SPI 设备时, 请遵循 22.3.4 的流程操作。</i>

22.4.4 SPI 状态寄存器 (SPI_STS)

地址偏移: 0x08

复位值: 0x0000 0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ROTCF	Reserved					OVER	MODERR	CRCERR	BUSY	RNE	TE
				r						r	r	rc_w0	r	r	r

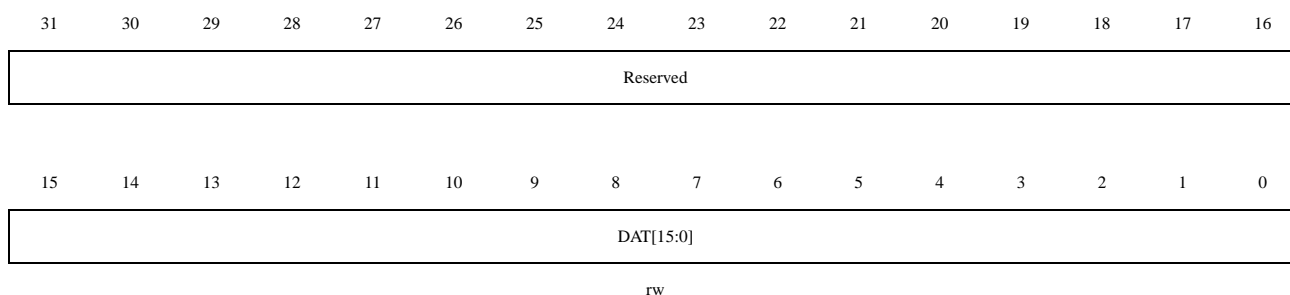
位域	名称	描述
31:13	Reserved	保留, 必须保持复位值。
12	ROTCF	只收模式下, 传输完成状态 0: 传输完成标志未拉起 1: 传输完成标志拉起
11:6	Reserved	保留, 必须保持复位值。
5	OVER	溢出标志 0: 没有出现溢出错误; 1: 出现溢出错误。 该位由硬件置位, 由软件序列复位。关于软件序列的详细信息, 参考 22.3.7 节。
4	MODERR	模式错误 0: 没有出现模式错误; 1: 出现模式错误。 该位由硬件置位, 由软件序列复位。关于软件序列的详细信息, 参考 22.3.7 节。
3	CRCERR	CRC 错误标志 0: 收到的 CRC 值和 SPI_CRCRDAT 寄存器中的值匹配; 1: 收到的 CRC 值和 SPI_CRCRDAT 寄存器中的值不匹配。 该位由硬件置位, 由软件写‘0’而复位。
2	BUSY	忙标志 0: SPI 不忙; 1: SPI 正忙于通信, 或者发送缓冲非空。 该位由硬件置位或者复位。 <i>注意: 使用这个标志时需要特别注意, 详见第 22.3.3 章节和第 22.3.4 章节。</i>

位域	名称	描述
1	RNE	接收缓冲非空 0: 接收缓冲为空; 1: 接收缓冲非空。
0	TE	发送缓冲为空 0: 发送缓冲非空; 1: 发送缓冲为空。

22.4.5 SPI 数据寄存器 (SPI_DAT)

地址偏移: 0x0C

复位值: 0x0000 0000

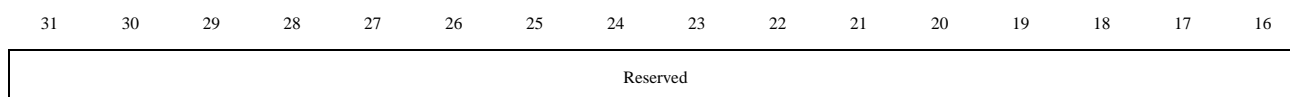


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	DAT[15:0]	数据寄存器 待发送或者已经收到的数据 数据寄存器对应两个缓存: 一个用于写 (发送缓存); 另外一个用于读 (接收缓存)。写操作将数据写到发送缓存; 读操作将返回接收缓存里的数据。 对 SPI 模式的注释: 根据 SPI_CTRL1.DATFF 位对数据帧格式的选择, 数据的发送和接收可以是 8 位或者 16 位的。为保证正确的操作, 需要在启用 SPI 之前就确定好数据帧格式。 对于 8 位的数据, 缓冲器是 8 位的, 发送和接收时只会用到 SPI_DAT[7:0]。在接收时, SPI_DAT[15:8]被强制为 0。 对于 16 位的数据, 缓冲器是 16 位的, 发送和接收时会用到整个数据寄存器, 即 SPI_DAT[15:0]。

22.4.6 SPI Tx CRC 寄存器 (SPI_CRCTDAT)

地址偏移: 0x10

复位值: 0x0000



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

CRCTDAT[15:0]

r

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRCTDAT	<p>发送 CRC 寄存器</p> <p>在启用 CRC 计算时，CRCTDAT[15:0]中包含了依据将要发送的字节计算的 CRC 数值。当在 SPI_CTRL2 中的 CRCEN 位写入‘1’时，该寄存器被复位。CRC 计算使用 SPI_CRCPOLY 中的多项式。</p> <p>当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 个位都参与计算，并且按照 CRC16 的标准。</p> <p><i>注意：当 BUSY 标志为‘1’时读该寄存器，将可能读到不正确的数值。</i></p>

22.4.7 SPI Rx CRC 寄存器（SPI_CRCRDAT）

地址偏移：0x14

复位值：0x0000

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

Reserved

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

CRCRDAT[15:0]

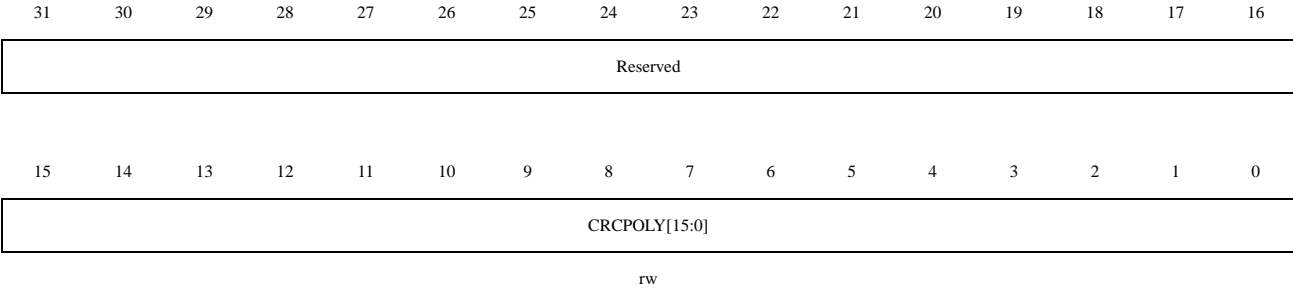
r

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRCRDAT	<p>接收 CRC 寄存器</p> <p>在启用 CRC 计算时，CRCRDAT[15:0]中包含了依据收到的字节计算的 CRC 数值。当在 SPI_CTRL2.CRCEN 位写入‘1’时，该寄存器被复位。CRC 计算使用 SPI_CRCPOLY 中的多项式。</p> <p>当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 位都参与计算，并且按照 CRC16 的标准。</p> <p><i>注意：当 BUSY 标志为‘1’时读该寄存器，将可能读到不正确的数值。</i></p>

22.4.8 SPI CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x18

复位值：0x0007

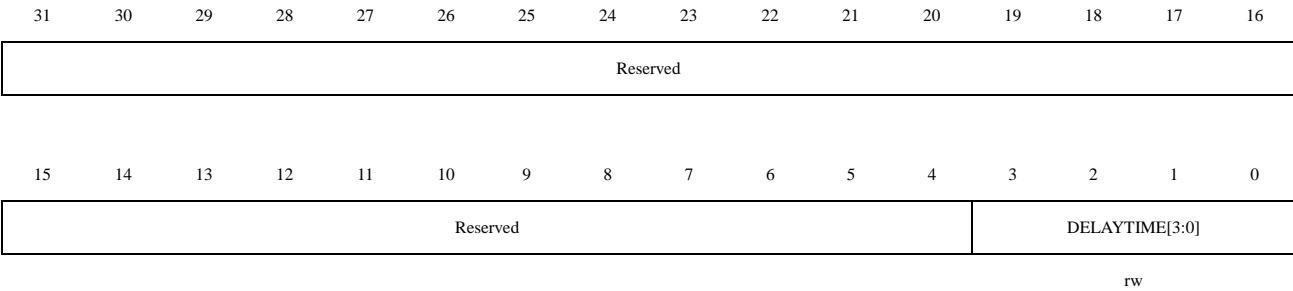


位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRCPOLY [15:0]	CRC 多项式寄存器 该寄存器包含了 CRC 计算时用到的多项式。 其复位值为 0x0007，根据应用可以设置其他数值。

22.4.9 SPI 时钟采样延迟寄存器（SPI_CTRL3）

地址偏移：0x38

复位值：0x0000



位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3:0	DELAYTIME [15:0]	SPI 主机时钟延迟时间配置 SPI 作为主机模式下，用于配置 MISO 与 SCK 之间的延时，用于规避总线上带来的接收延时 4'b0000: bypass，对主机生成的时钟不做延迟处理； 4'b0001: 用 APB2 时钟下降沿对 SPI SCLK 采 1 个 PCLK2，延迟为 1/2 个 PCLK2； 4'b0010: 用 APB2 时钟下降沿对 SPI SCLK 采 1 个 PCLK2，延迟为 2/2 个 PCLK2； 4'b0011: 用 APB2 时钟下降沿对 SPI SCLK 采 2 个 PCLK2，延迟为 3/2 个 PCLK2； 4'b0100: 用 APB2 时钟下降沿对 SPI SCLK 采 2 个 PCLK2，延迟为 4/2 个 PCLK2； 4'b0101: 用 APB2 时钟下降沿对 SPI SCLK 采 3 个 PCLK2，延迟为 5/2 个 PCLK2； 4'b0110: 用 APB2 时钟下降沿对 SPI SCLK 采 3 个 PCLK2，延迟为 6/2 个 PCLK2； 4'b0111: 用 APB2 时钟下降沿对 SPI SCLK 采 4 个 PCLK2，延迟为 7/2 个 PCLK2； 4'b1000: 用 APB2 时钟下降沿对 SPI SCLK 采 4 个 PCLK2，延迟为 8/2 个 PCLK2；

位域	名称	描述
		4'b1001: 用 APB2 时钟下降沿对 SPI SCLK 采 5 个 PCLK2, 延迟为 9/2 个 PCLK2; 4'b1010: 用 APB2 时钟下降沿对 SPI SCLK 采 5 个 PCLK2, 延迟为 10/2 个 PCLK2; 4'b1011: 用 APB2 时钟下降沿对 SPI SCLK 采 6 个 PCLK2, 延迟为 11/2 个 PCLK2; 4'b1100: 用 APB2 时钟下降沿对 SPI SCLK 采 6 个 PCLK2, 延迟为 12/2 个 PCLK2; 4'b1101: 用 APB2 时钟下降沿对 SPI SCLK 采 7 个 PCLK2, 延迟为 13/2 个 PCLK2; 4'b1110: 用 APB2 时钟下降沿对 SPI SCLK 采 7 个 PCLK2, 延迟为 14/2 个 PCLK2; 4'b1111: 用 APB2 时钟下降沿对 SPI SCLK 采 8 个 PCLK2, 延迟为 15/2 个 PCLK2; 注: 该寄存器只可在 SPI 作为主机模式下可配置, SPI 作为从机模式下配置该位无效

23 实时时钟(RTC)

23.1 简介

- 实时时钟（RTC）是一个独立的 BCD 定时器/计数器
- 软件支持夏令时补偿
- 可编程周期性自动唤醒定时器
- 两个 32 位寄存器包含时、分、秒、年、月、日（几号）、星期（星期几）
- 独立的 32 位寄存器包含亚秒
- 两个编程闹钟
- 两个 32 位寄存器包含编程闹钟时、分、秒、年、月、日（几号）、星期（星期几）
- 两个独立的 32 位寄存器包含编程闹钟亚秒
- 数字精密校准功能
- 参考时钟检测：一个更加精确的外部时钟源（50 或 60Hz）能够用于改进日历精度
- 三个可配置滤波和内部上拉的入侵检测事件
- 时间戳功能
- 多个中断/事件唤醒源，包括闹钟 A、闹钟 B、唤醒定时器、时间戳、入侵
- RCC 寄存器使能 RTC 模块且电压保持在工作范围内，RTC 在除 PD 模式外任何模式下都不会停止（包括 RUN 模式、SLEEP 模式、STOP 模式）
- RTC 提供多种唤醒源可以使 MCU 从所有的低功耗模式下唤醒（SLEEP 模式和 STOP 模式）

23.1.1 主要特性

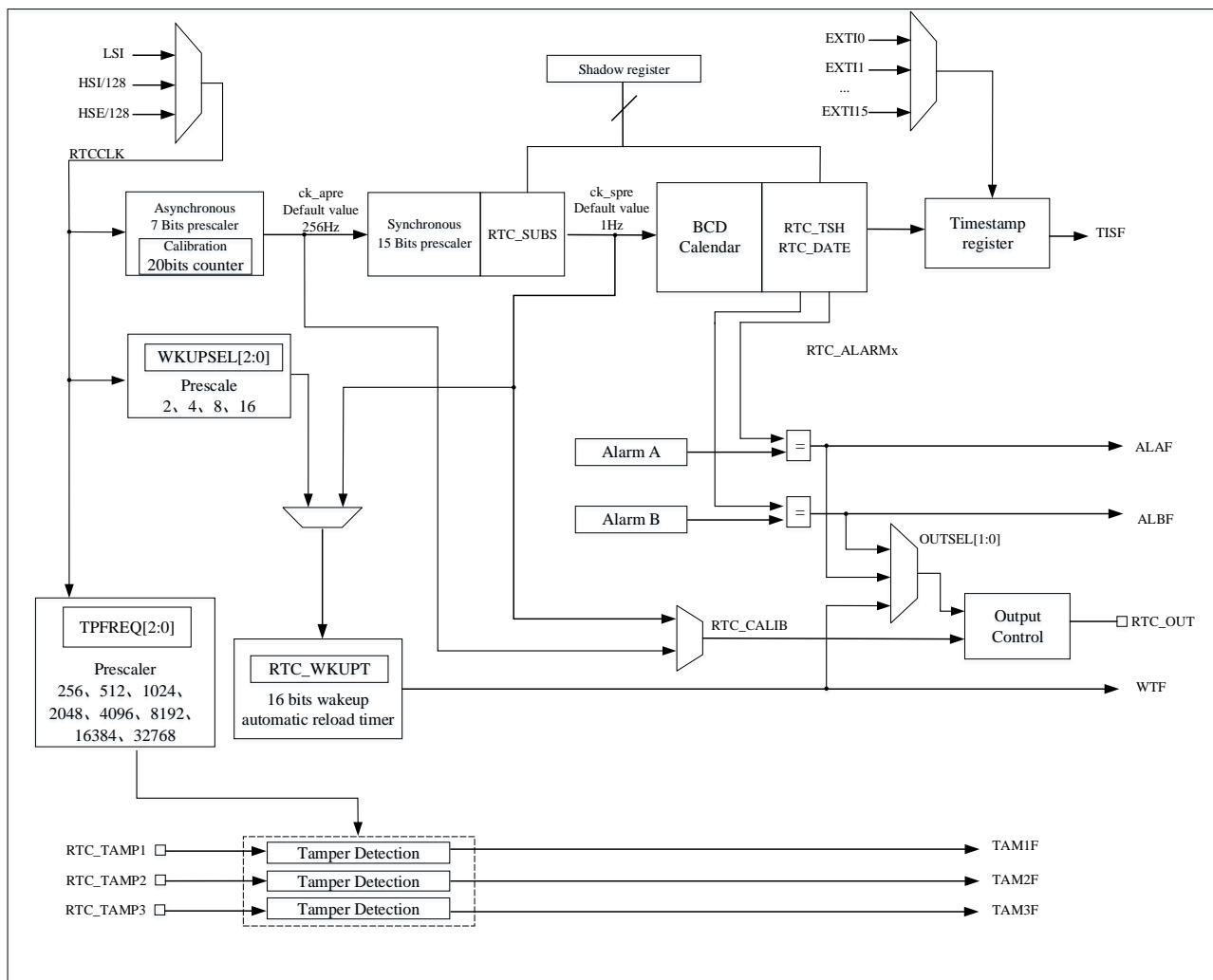
表 23-1 RTC 功能支持

主要功能	描述
时钟	RTC 时钟源可以选择 LSI、HSI/128 或 HSE/128
Reset	<p>APB 接口被系统复位， RTC 模块通过 APB 同步的一些寄存器会被复位</p> <p>下面寄存器当系统复位时会被清除</p> <ul style="list-style-type: none"> ● RTC_SUBS ● RTC_TSH ● RTC_DATE ● RTC_INITSTS(一些 bits) <p>RTC 内核可以通过备份域复位而复位</p> <p>复位 RTC，以及在低功耗模式下保留一些寄存器的内容，包括：</p> <ul style="list-style-type: none"> ● RTC_CTRL ● RTC_PRE ● RTC_CALIB ● RTC_SCTRL ● RTC_TSSS, RTC_TST and RTC_TSD ● RTC_TMPCFG ● RTC_WKUPT ● RTC_ALRMAS/RTC_ALRMA ● RTC_ALRMBSS/RTC_ALRMB ● RTC_OPT
Calendar	日历包含亚秒、秒、分、时（12 小时或 24 小时制）、星期、日、月、年，这些数据都存在 APB 模块的影子寄存器中。
Wakeup Timer	输出寄存器 RTC_OUT 可以配置为发送唤醒事件到 GPIO，同时我们可以选择中断/事件来唤醒 CPU 的 SLEEP、STOP 模式。
Alarm	RTC_OUT 配置输出到 GPIO，也可以唤醒 CPU 或触发 PWR 在匹配发生时从 SLEEP、STOP 模式中唤醒
Tamper	3 个入侵检测逻辑，入侵事件发生时可将 CPU 从 SLEEP、STOP 模式中唤醒。
Timestamp	<p>GPIO 事件可触发保存时间戳功能。它是从低功耗模式唤醒的一个来源。</p> <p>另外，入侵事件可以是时间戳事件的来源。</p>
Interrupts/events	<p>闹钟 A/闹钟 B 中断/事件</p> <p>唤醒中断/事件</p> <p>时间戳中断/事件</p> <p>入侵中断/事件</p>

23.2 RTC 功能描述

23.2.1 RTC 框图

图 23-1 RTC 功能框图



RTC 包括以下功能模块：

- Alarm A 和 Alarm B 事件/中断
- 时间戳事件/中断
- 入侵事件/中断
- RTC 输出功能：
 - ◆ 256 Hz 或者 1Hz 时钟输出(当 LSI 频率是 32 kHz)
 - ◆ 闹钟输出（极性可配置），闹钟 A 和闹钟 B 可选
 - ◆ 自动唤醒输出（极性可配置）
- RTC 输入功能：

- ◆ 时间戳事件检测
- ◆ 50 或者 60Hz 参考时钟输入
- ◆ 入侵事件检测
- 通过配置输出寄存器控制 PC13:
 - ◆ 设置 RTC_OPT.TYPE 位配置 PC13 开漏/推挽输出

23.2.2 RTC 控制的 GPIO

时间戳输入来自 IOM（映射到 PC13）或者 EXTI 模块，如果是 EXTI 模块，具体请参考时间戳触发源选择 (EXTI_TS_SEL)。

RTC_OUT（闹钟、唤醒事件或者校准输出（256Hz 或者 1Hz））映射到 PC13，不管 PC13 GPIO 是什么配置，PC13 的引脚配置由 RTC 控制为输出。

PC13 引脚被用作 TAMPER1 入侵检测引脚，PA0 引脚被用作 TAMPER2 入侵检测引脚，

PB8 引脚被用作 TAMPER3 入侵检测引脚。

PA1 或者 PB15 能够被用作 RTC_REFCLKIN 参考时钟输入引脚。

23.2.3 RTC 寄存器写保护

上电后，所有的 RTC 写保护寄存器都会写保护，所有的 RTC 写保护寄存器需要按如下步骤去解锁写保护：

- 将 0xCA 写入 RTC_WRP 寄存器
- 将 0x53 写入 RTC_WRP 寄存器

在解锁这些寄存器后，不能重新使能写保护除非 RTC 被软件复位或者重新上电。解锁机制只检查 RTC_WRP 寄存器的写操作。在解锁过程中、解锁前、解锁后，对其他寄存器的写操作不会影响解锁结果。

RTC 复位后，RTC 寄存器除 RTC_CTRL、RTC_WRP、RTC_TST、RTC_DATE、RTC_TSSS、RTC_TMPCFG、RTC_SUBS、RTC_OPT 和 RTC_INITSTS[13:8]外，其他寄存器都是写保护。

23.2.4 RTC 时钟和预分频

RTC 时钟源：

- LSI 时钟
- HSI/128 时钟
- HSE/128 时钟

为了降低功耗，将预分频器分为异步预分频器和同步预分频器。如果同时使用两个预分频器，建议异步预分频器的值尽可能大。

- 7 位异步预分频器由 RTC_PRE.DIVA[6:0] 位控制
- 15 位同步预分频器由 RTC_PRE.DIVS[14:0] 位控制

注意：预分频配置必须在日历初始化中实现。

f_{ck_apre} 和 f_{ck_spre} 公式如下:

$$f_{ck_apre} = \frac{f_{RTCCLK}}{RTC_PRE.DIVA[6:0]+1}$$

$$f_{ck_spre} = \frac{f_{RTCCLK}}{(RTC_PRE.DIVS[14:0]+1)*(RTC_PRE.DIVA[6:0]+1)}$$

ck_apre 时钟用于对 RTC_SUBS 亚秒递减计数器提供时钟。当到达 0 时, 用 RTC_PRE.DIVS[14:0]的值重新加载 RTC_SUBS。

23.2.5 RTC 日历

这里有三个影子寄存器, 分别是 RTC_DATE, RTC_TSH 和 RTC_SUBS。RTC 时间和日期寄存器可以通过影子寄存器访问。也可以直接访问, 以避免等待同步时间。这三个影子寄存器如下:

- RTC_DATE: 设置和读取日期
- RTC_TSH: 设置和读取时间
- RTC_SUBS: 读取亚秒

每隔两个 RTCCLK 周期之后, 将当前的日历值复制到影子寄存器中, 并将 RTC_INITSTS.RSYF 位置为 1。此过程在低功耗(停止和待机)模式下不执行。当退出这些模式时, 影子寄存器在 2 个 RTCCLK 周期后更新值。

默认情况下, 当用户尝试访问日历寄存器时, 它将访问影子寄存器的内容。用户可以通过设置 RTC_CTRL.BYPS 位直接访问日历寄存器。

当 RTC_CTRL.BYPS=0, 日历从影子寄存器获取值, 当读 RTC_SUBS、RTC_TSH 或 RTC_DATE 寄存器时, 有必要确保 APB1 时钟的频率(f_{APB1})至少 7 倍于 RTC 时钟频率(f_{RTCCLK}), 而且不允许出现 APB1 时钟频率低于 RTC 时钟频率的情况。系统复位将复位影子寄存器。

23.2.6 日历初始化和配置

预分频值和日历值可通过以下步骤进行初始化:

- 通过设置 RTC_INITSTS.INITM 位为 1 进入初始模式, 然后等待 RTC_INITSTS.INITF 位被置 1
- 设置 RTC_PRE.DIVS[14:0] 和 RTC_PRE.DIVA[6:0] 位
- 写入初始日历值, 包括时间和日期到影子寄存器 (RTC_TSH 和 RTC_DATE), 通过 RTC_CTRL.HFMT 位配置时间格式 (12 小时或 24 小时制)
- 通过清除 RTC_INITSTS.INITM 位退出初始化模式

日历计数器的值将在 4 个 RTCCLK 时钟周期后自动从影子寄存器加载, 然后重新启动日历计数器。

注意: RTC 进入初始化模式前, 需保证 RTC_SUBS.SS[15:0]的值不小于 2。

23.2.7 日历读取

1. 当 RTC_CTRL.BYPS=0 时读取日历

如果 RTC_CTRL.BYPS=0, 则从影子寄存器读取日历值。为了正确读取 RTC 日历寄存器(RTC_SUBS,

RTC_TSH 和 RTC_DATE), APB1 时钟频率必须设置为大于 RTC 时钟频率的 7 倍。在任何情况下, APB1 时钟频率都不能小于 RTC 时钟频率。

如果 APB1 时钟频率不大于或不等于 RTC 时钟频率的 7 倍, 请参考下面的步骤读取日历值:

- 读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 值两次
- 比较两次读到的数据, 如果相等, 则认为读到的数据是正确的, 如果不相等, 需要读第三次数据
- 第三次读到的数据可以认为是正确的

影子寄存器(RTC_SUBS, RTC_TSH 和 RTC_DATE)每两个 RTCCLK 周期更新一次。如果用户希望在短时间内(小于两个 RTCCLK 周期)读取日历值, 则第一次读取后必须软件清除 RTC_INITSTS.RSYF 位。

在一些情况下, 在读取日历之前需要等待 RTC_INITSTS.RSYF 位被置 1。

- 从低功耗模式(待机模式)唤醒后, 清除 RTC_INITSTS.RSYF 位, 然后等待 RTC_INITSTS.RSYF 位重新置 1。
- 系统复位。
- 日历完成初始化。
- 日历完成同步。

2. 当 RTC_CTRL.BYPS=1 时读取日历

如果 RTC_CTRL.BYPS=1, 直接从日历计数器中读取日历值。这种配置的优点是, 从低功耗模式唤醒后读取日历值没有延迟, 缺点是 RTC_SUBS、RTC_TSH 和 RTC_DATE 的这些数据可能不是同一时刻的。

为了保证读取的日历值的正确性, 需要分别读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 两次, 然后对两次读取的数据进行比较, 如果两者相等, 则认为读取的数据是正确的。

23.2.8 校准时钟输出

当 RTC_CTRL.COEN 位置 1, PC13 引脚将输出校准时钟。如果 RTC_CTRL.CALOSEL= 0 和 RTC_PRE.DIVA[6:0] = 0x7C, RTC_CALIB 频率结果为 $f_{RTCCLK} / (RTC_PRE.DIVA[6:0]+1)$ 。当 RTCCLK 频率为 32 kHz 时, 校准输出 256Hz。由于下降沿有轻微的抖动, 建议使用上升沿。

当 RTC_CTRL.CALOSEL=1, " RTC_PRE.DIVS[14:0]+1" 是 256 的非零整数倍, RTC_CALIB 频率由公式 $f_{RTCCLK} / (256 * (DIVA+1))$ 给出。当 RTCCLK 频率为 32 kHz 和 RTC_PRE.DIVA[6:0] = 0x7C 时, 校准输出 1Hz。

注意: 当选择 RTC_CALIB 或 RTC_ALARM 输出时, RTC_OUT 引脚(PC13)被自动配置为输出。

23.2.9 可编程闹钟

RTC 有 2 个可编程闹钟: 闹钟 A 和闹钟 B。

通过 RTC_CTRL.ALxEN 位可以使能或关闭 RTC 闹钟。如果 Alarm 值与日历值相匹配, 则 RTC_INITSTS.ALxF 标志被置 1。如果 RTC_CTRL.ALxIEN 使能, 可以选择任意日历字段来触发闹钟中断。

闹钟输出: 当 RTC_CTRL.OUTSEL[1:0]配置后, 闹钟 A 或闹钟 B 可以映射到 RTC_ALxRM 输出, 可以通过 RTC_CTR.OPOL 位配置输出极性。

注意: 当秒字段被选择(RTC_ALARMx.MASK1 位复位), RTC_PRE.DIVS[14:0]必须大于 3, 以保证正确操作。

23.2.10 闹钟配置

闹钟 A 和闹钟 B 配置步骤如下：

- 通过清除 RTC_CTRL.ALAEN/RTC_CTRL.ALBEN 位失能闹钟 A/闹钟 B
- 配置闹钟 x 寄存器 (RTC_ALRMxSS/RTC_ALARMx)
- 通过设置 RTC_CTRL.ALAIEEN/RTC_CTRL.ALBIEEN 位为 1 使能闹钟 A/闹钟 B 中断（这一步根据需要添加）
- 通过设置 RTC_CTRL.ALAEN/RTC_CTRL.ALBEN 位为 1 使能闹钟 A/闹钟 B

23.2.11 闹钟输出

当 RTC_CTRL.OUTSEL[1:0] != 0, RTC_ALARM 输出功能开启。根据 RTC_CTRL.OUTSEL[1:0] 的值选择闹钟 A 输出、闹钟 B 输出或者唤醒输出。

RTC_CTRL.OPOL 位控制闹钟 A、闹钟 B 或唤醒输出的极性。

RTC_OPT.TYPE 位控制 RTC_ALARM 引脚开漏或者推挽输出。

选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚（PC13）会自动配置为输出。

23.2.12 周期性自动唤醒

16 位可编程自动加载计数器可以在达到 0 时产生周期性唤醒标志。它也可以将唤醒定时器的范围扩展到 17 位。通过设置 RTC_CTRL.WTEN 可以启用周期性自动唤醒功能。

可以选择两种唤醒输入时钟源：

- 2、4、8 或 16 分频的 RTC 时钟（RTCCLK）。

假设 RTCCLK 来自 LSI (32KHz)，在分辨率到 62.5us 的情况下，可以配置唤醒中断周期为 125us ~ 32s。

- 内部时钟 ck_spre。

假设 ck_spre 频率为 1Hz，可用唤醒时间范围为 1s ~ 36h，分辨率为 1 秒

- ◆ 当 RTC_CTRL.WKUPSEL [2:0] = 10x，周期范围为 1s 到 18h
- ◆ 当 RTC_CTRL.WKUPSEL [2:0] = 11x，周期范围为 18h 到 36h。

当 RTC_CTRL.WTEN 位设置为 1 之后，向下计数器正在运行，当它达到 0 时，RTC_INITSTS.WTF 位会被置 1，通过设置 RTC_CTRL.WTIEEN 位为 1，当周期性唤醒中断被启用触发时，设备可以退出低功耗模式。

周期性唤醒输出：当 RTC_CTRL.OUTSEL[1:0] 选择周期性唤醒后可以映射到 RTC_ALxRM 输出，自动将 RTC_OUT 引脚(PC13)配置为输出，输出极性可由 RTC_CTRL.OPOL 位配置。

23.2.13 唤醒定时器配置

唤醒计时器自动重新加载值配置如下：

- 通过清除 RTC_CTRL.WTEN 关闭唤醒定时器，然后等待 RTC_INITSTS.WTWF 标志位被置 1
- 通过设置 RTC_CTRL.WKUPSEL[2:0] 选择唤醒定时器时钟

- 通过设置 RTC_WKUPT.WKUPT[15:0]配置唤醒自动重加载值
- 通过设置 RTC_CTRL.WTIEN 位使能唤醒中断 (此步可根据需要选择)
- 通过设置 RTC_CTRL.WTEN 位开启唤醒定时器

23.2.14 时间戳功能

时间戳可以通过将 RTC_CTRL.TSEN 位设置为 1 来启用。当在 RTC_TS 引脚上检测到时间戳事件时，该事件的日历值将存储在时间戳寄存器（RTC_TSSS、RTC_TST、RTC_TSD）中，并且 RTC_INITSTS.TISF 位被设置为 1。如果 RTC_CTRL.TSIEN 设置为 1，则时间戳事件可以产生中断。如果在 RTC_INITSTS.TISF 已经设置为 1 时检测到新的时间戳事件，则硬件将 RTC_INITSTS.TISOVF 标志设置为 1，并且时间戳寄存器（RTC_TST 和 RTC_TSD）将继续保存前一个事件的值，这意味着当 RTC_INITSTS.TISF=1 时，时间戳寄存器（RTC_TST 和 RTC_TSD）数据不会改变。

在同步过程引起的时间戳事件再次发生后，RTC_INITSTS.TISF 在 2 个 RTC_CLK 周期内设置为 1。RTC_INITSTS.TISOVF 的生成没有延迟。这意味着如果两个时间戳事件非常接近，这可能导致 RTC_INITSTS.TISOVF 为“1”而 RTC_INITSTS.TISF 为“0”。因此，在检测到 RTC_INITSTS.TISF 为“1”后，再检测 RTC_INITSTS.TISOVF 位。当 RTC_TMPCFG.TPTS 位设置为 1 时，入侵事件可以触发时间戳事件。

如果启用时间戳事件，时间戳将在时间戳寄存器中捕获读取的日历。当入侵事件和时间戳事件都启用时，入侵事件也会导致时间戳捕获。时间戳事件可以在 EXTI 选择的 16 个 GPIO 端口中的任何一个上生成。通过设置相应的 EXTI_TS_SEL.TSSEL[3:0] 位来选择任一端口的 GPIO 引脚。

23.2.15 入侵检测

共有三个入侵检测引脚，RTC_TAMP1 引脚为 PC13，RTC_TAMP2 引脚为 PA0，RTC_TAMP3 引脚为 PA8。RTC_TAMPx 引脚可用作入侵事件检测功能输入引脚。有两种检测模式，边缘检测模式和可配置滤波功能的电平检测模式。

入侵检测初始化

共有三个入侵检测引脚，每个引脚都可以独立配置。用户需要在设置 RTC_TMPCFG.TPxEN 位之前配置入侵检测。当入侵检测使能后检测到入侵事件时，如果 RTC_TMPCFG.TPxINTEN 位置 1，则入侵事件可以产生中断并且 RTC_INITSTS.TAMxF 位将被置 1。

当 RTC_INITSTS.TAMxF 位为 1 时，无法检测到同一引脚上的新入侵事件。

入侵事件的时间戳

当 RTC_TMPCFG.TPTS 位设置为 1 时，任何入侵事件都可以触发时间戳事件，并且 RTC_INITSTS.TISF 位和 RTC_INITSTS.TISOVF 位将被设置为正常的时间戳事件。

入侵输入的边缘检测

当 RTC_TMPCFG.TPFLT[1:0] 位设置为 0 时，入侵检测设置为边沿检测，上升沿或下降沿由 RTC_TMPCFG.TPxTRG 位控制。当检测到相应的边沿时，RTC_TAMPx 引脚将产生一个入侵检测事件。

RTC_TAMPx 输入的滤波电平检测

当 RTC_TMP_CFG.TPFLT[1:0] 位设置为 1/2/3 时, 入侵检测设置为电平检测。RTC_TMP_CFG.TPFLT[1:0] 的值决定了采样次数。

每次采样前可通过入侵引脚的内部上拉电阻进行预充电, 预充电时间由 RTC_TMP_CFG.TPPRCH[1:0] 位控制。当 RTC_TMP_CFG.TPPUDIS 设置为 1 时, 预充电将被禁用。

使用 RTC_TMP_CFG.TPFREQ[2:0] 确定电平检测的采样频率, 可以优化入侵检测延迟和上拉功耗之间的最佳平衡。

23.2.16 夏令时功能配置

夏令时功能可通过 RTC_CTRL.SUIH、RTC_CTRL.AD1H 和 RTC_CTRL.BAKP 位控制。设置 RTC_CTRL.SUIH 位为 1 时日历会减一小时, 设置 RTC_CTRL.AD1H 为 1 时会增加一小时。RTC_CTRL.BAKP 位可用于记住或不记住此调整。

23.2.17 RTC 亚秒寄存器位移操作

当日历的值与外部精密时钟相比有亚秒级的偏差时, 可以使用移位功能来提高日历的精度。

日历可以使用 RTC_SCTRL.AD1S 和 RTC_SCTRL.SUBF[14:0] 位来控制最大延迟或提前 1s。调整分辨率为 $1/(RTC_PRE.DIVS[14:0]+1)$, 表示 RTC_PRE.DIVS[14:0] 的值越大, 分辨率越高。为了使同步预分频器输出保持在 1Hz, RTC_PRE.DIVS[14:0] 越高意味着 RTC_PRE.DIVA[6:0] 越低, 则功耗越大。

注意: 在开始移位操作之前, 用户必须检查 RTC_SUBS.SS[15] 位是否为 0。

每当写入 RTC_SCTRL 寄存器时, 硬件都会设置 RTC_INITSTS.SHOPF 标志, 表明平移操作处于挂起状态。一旦平移操作完成, 该位由硬件清零。

23.2.18 RTC 数字时钟精密校准

数字精密校准是通过调整校准周期内的 RTC 时钟脉冲数来实现的。数字精度校准分辨率为 0.954 PPM, 范围为 -487.1 PPM 到 +488.5 PPM。

当输入频率为 32000 Hz 时, 校准周期可配置为 $2^{20}/2^{19}/2^{18}$ 个 RTCCLK 周期或 32/16/8 秒。精密校准寄存器 (RTC_CALIB) 表示将在指定周期内减少 RTC_CALIB.CM[8:0] 个 RTCCLK 时钟周期。

RTC_CALIB.CM[8:0] 的值表示在指定周期内要减少的 RTCCLK 脉冲数。RTC_CALIB.CP 可用于增加 488.5 PPM, 每 2^{11} 个 RTCCLK 周期将插入一个 RTCCLK 脉冲。

当 RTC_CALIB.CM[8:0] 和 RTC_CALIB.CP 组合使用时, 增加的周期范围为 -511 到 +512 个 RTCCLK 周期, 校准范围为 -487.1 ppm 到 +488.5 ppm, 分辨率约为 0.954 ppm。

有效校准频率 (f_{CAL}) 可使用以下公式计算:

$$f_{CAL} = f_{RTCCLK} * \left(1 + \frac{RTC_CALIB.CP * 512 - RTC_CALIB.CM[8:0]}{2^n + RTC_CALIB.CM[8:0] - RTC_CALIB.CP * 512}\right)$$

注意: $n=20/19/18$

当 RTC_PRE.DIVA[6:0] < 3 时校准

当异步预分频器值 (RTC_PRE.DIVA[6:0]) 小于 3 时, 不能将 RTC_CALIB.CP 设置为 1, 如果 RTC_CALIB.CP 值已设置为 1, 则将被忽略。

假设 RTCCLK 频率为 32000Hz，当 RTC_PRE.DIVA[6:0]<3 时，RTC_PRE.DIVS[14:0]的值应该减小：

- 当 RTC_PRE.DIVA[6:0]=2, RTC_PRE.DIVS[14:0]=8189.
- 当 RTC_PRE.DIVA[6:0]=1, RTC_PRE.DIVS[14:0]=16379.
- 当 RTC_PRE.DIVA[6:0]=0, RTC_PRE.DIVS[14:0]=32759.

有效校准频率 (f_{CAL}) 可使用以下公式计算：

$$f_{CAL} = f_{RTCCLK} * \left(1 + \frac{256 - RTC_CALIB.CM[8:0]}{2^n + RTC_CALIB.CM[8:0] - 265} \right)$$

注意：n=20/19/18

验证 RTC 校准

RTC 输出 1Hz 波形，用于测量和验证 RTC 精度。

在有限测量周期内测量 RTC 频率时，最多可能出现 2 个 RTCCLK 周期测量误差。如果测量周期与校准周期相同，则可以消除误差。

- 校准周期为 32 秒（默认）

使用精确的 32 秒周期测量 1Hz 校准输出可以确保测量误差在 0.447ppm 以内（32 秒内为 0.5 个 RTCCLK 周期）。

- 校准周期为 16 秒。

使用精确的 16 秒周期测量 1Hz 校准输出可以确保测量误差在 0.954ppm 以内（16 秒内为 0.5 个 RTCCLK 周期）。

- 校准周期为 8 秒。

使用精确的 8 秒周期测量 1Hz 校准输出可以确保测量误差在 1.907ppm 以内（8 秒内为 0.5 个 RTCCLK 周期）。

动态重新校准

当 RTC_INITSTS.INITF=0 时，RTC_CALIB 寄存器可以通过以下步骤更新：

- 等待 RTC_INITSTS.RECPF=0
- 一个新值被写入 RTC_CALIB，然后 RTC_INITSTS.RECPF 位自动置 1
- 新的校准设置将在数据写入 RTC_CALIB 后的 3 个 ck_apre 周期内生效

23.2.19 RTC 低功耗模式

RTC 在低功耗模式下的工作状态。

低功耗模式	RTC 工作状态	退出低功耗模式
SLEEP	正常工作	RTC 中断
STOP	RTC 时钟源为 LSI 时正常工作	闹钟 A、闹钟 B、定期唤醒、入侵事件和时间戳事件

注意：当使用 RTC 事件将 MCU 从 STANDBY 模式唤醒时，需要将 PWR_CTRLSTS.WKUPRTCEN 位使能。

23.3 RTC 寄存器

23.3.1 RTC 寄存器总览

表 23-2 RTC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
000h	RTC_INITSTS	Reserved															RECFP	TAM3F	TAM2F	TAM1F	TISOVF	TISF	WTF	ALBF	ALAF	INITM	INITF	RSYF	INITSF	SHOPF	WTWF	ALBWF	ALAWF																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
004h	RTC_CTRL	Reserved									COEN	OUTSEL[1:0]		OPOL	CALOSEL	BAKP	SUIH	ADIH	TSIEN	WTIEN	ALBIEN	ALAIEN	TSEN	WTEN	ALBEN	ALAIEN	Reserved	HFMT	BYP	REFCLKEN	TEDGE	WKUPSEL[2:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
	Reset Value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
008h	RTC_TSH	Reserved									APM	HOT	1:0		HOU[3:0]			Reserved	MIT[2:0]			MIU[3:0]			Reserved	SCT[2:0]		SCU[3:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
	Reset Value										0	0	0	0	0	0	0		0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
00Ch	RTC_DATE	Reserved									YRT[3:0]			YRU[3:0]			WDU[2:0]		MOT	MOU[3:0]			Reserved	DAT[1:0]		DAU[3:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
	Reset Value										0	0	0	0	0	0	0	0	1	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

034h	RTC_ALRM	Reserved	MASKSSA[3:0]				Reserved	SSV[14:0]													
	ASS		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0		
	Reset Value		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	
038h	RTC_OPT	Reserved																		TYPE	
	Reset Value																				0
03Ch	RTC_ALRM	Reserved	MASKSSB[3:0]				Reserved	SSV[14:0]													
	BSS		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	
	Reset Value		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	
040h	RTC_CALIB	Reserved										C	CW8	CW16	CM[8:0]						
	0														0	0	0	0	0	0	0
	Reset Value											0	0	0	0	0	0	0	0	0	0
044h	RTC_TSSS	Reserved						SSE[15:0]													
	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	RTC_TSD	Reserved						MOT	WDU[2:0]		MOU[3:0]			Reserved	DAT[1:0]		DAU[3:0]				
	0														0	0	0	0	0	0	0
	Reset Value							0	0	0	0	0	0	0	0	0	0	0	0		

23.3.2 RTC 初始状态寄存器 (RTC_INITSTS)

偏移地址: 0x00

复位值: 0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
Reserved																RECPF				
																r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
TAM3F	TAM2F	TAM1F	TISOVF	TISF	WTF	ALBF	ALAF	INTM	INTF	RSYF	INITSF	SHOPF	WTWF	ALBWF	ALAWF					
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r					

位域	名称	描述
31:17	Reserved	保留，必须保持复位值。
16	RECPF	重新校准挂起标志位。 当软件写入 RTC_CALIB 寄存器时，RECPF 状态标志自动设置为 1，表示 RTC_CALIB 寄存器被阻塞。当考虑到新的校准设置时，这个位将恢复为 0。
15	TAM3F	RTC_TAMP3 检测标志位。 当在 RTC_TAMP3 输入口上检测到侵入事件时，硬件将设置此标志。通过软件写 0 清除
14	TAM2F	RTC_TAMP2 检测标志位。 当在 RTC_TAMP2 输入口上检测到侵入事件时，硬件将设置此标志。通过软件写 0 清除
13	TAM1F	RTC_TAMP1 检测标志位。 当在 RTC_TAMP1 输入口上检测到侵入事件时，硬件将设置此标志。通过软件

位域	名称	描述
		写 0 清除
12	TISOVF	时间戳溢出标志位。 当时间戳事件发生的同时 TISF 位已经被置 1 时，硬件将此标志置 1。建议在清除 TISF 位之后再检查并清除 TISOVF 位。否则，如果时间戳事件恰好在清除 TISF 位之前刚刚发生，则溢出事件可能会被漏掉。
11	TISF	时间戳标志位。 当发生时间戳事件时，硬件将设置此标志。此标志通过写入 0 被软件清除。
10	WTF	唤醒定时器标志位。 当唤醒自动重载计数器达到 0，硬件将设置此标志。此标志通过写入 0 被软件清除。在 WTF 再次设置为 1 之前，此标志必须由软件至少在 1.5 RTCCLK 周期内清除。
9	ALBF	闹钟 B 标志位。 当时间/日期寄存器(RTC_TSH 和 RTC_DATE)与闹钟 B 寄存器(RTC_ALARMB)匹配时，硬件将设置此标志。此标志通过写入 0 被软件清除。
8	ALAF	闹钟 A 标志位。 当时间/日期寄存器(RTC_TSH 和 RTC_DATE)与闹钟 A 寄存器(RTC_ALARMA)匹配时，硬件将设置此标志。此标志通过写入 0 被软件清除。
7	INITM	进入初始化模式 0: 自由运行模式 1: 进入初始化模式，设置日历时间值、日期值、预分频值。
6	INITF	初始标志位。 当这个位设置为 1 时，RTC 处于初始化状态，可以更新时间、日期和预分频寄存器。 0: 日历寄存器更新禁止 1: 日历寄存器更新允许
5	RSYF	寄存器同步标志位。 当日历值被复制到影子寄存器中时，该标志由硬件设置为“1”。当处于初始化模式、移位操作挂起 (SHOPF=1) 或处于旁路影子寄存器模式 (RTC_CTRL.BYPS=1) 时，该位由硬件清零，该位也可以通过软件清零。 在初始化模式下，该位通过软件或硬件清除。 0: 日历影子寄存器尚未同步 1: 日历影子寄存器同步
4	INITSF	初始状态标志位。 当日历年字段不等于 0 (备份域复位状态)时，由硬件设置此位。 0: 日历没有被初始化 1: 日历已经被初始化
3	SHOPF	平移操作挂起标志位。 当向 RTC_SCTRL 寄存器写入一个平位操作时，硬件立即设置此标志。当执行相应的平移操作时，硬件将清除它。写入 SHOPF 位不起作用。 0: 没有位移操作挂起 1: 有位移操作挂起
2	WTWF	唤醒定时器写标志位。 0: 唤醒时间配置更新不允许

位域	名称	描述
		1: 唤醒时间配置更新允许
1	ALBWF	闹钟 B 写标志。 当 RTC_CTRL.ALBEN 位设置为 0, 同时闹钟 B 值可更改时, 硬件将该位置 1。它在初始化模式下被硬件清除。 0: 闹钟 B 更新不允许 1: 闹钟 B 更新允许
0	ALAWF	闹钟 A 写标志。 当 RTC_CTRL.ALAEN 位设置为 0, 同时闹钟 A 可更改时, 硬件将该位置 1。它在初始化模式下被硬件清除。 0: 闹钟 A 更新不允许 1: 闹钟 A 更新允许

23.3.3 RTC 控制寄存器(RTC_CTRL)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								COEN	OUTSEL[1:0]	OPOL	CALOSEL	BAKP	SU1H	AD1H	
								rw	rw	rw	rw	rw	w	w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIEN	WTIEN	ALBIEN	ALAIEN	TSEN	WTEN	ALBEN	ALAEN	Reserved	HFMT	BYPS	REF CLKEN	TEDGE	WKUPSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

位域	名称	描述
31:24	Reserved	保留, 必须保持复位值。
23	COEN	校准输出使能。 0: 校准输出禁止 1: 校准输出开启
22:21	OUTSEL[1:0]	输出选择位。 该位用来选择要连接到 RTC_ALARM 输出的标志。 00: 输出禁止 01: 闹钟 A 输出开启 10: 闹钟 B 输出开启 11: 唤醒输出开启
20	OPOL	输出极性位。 此位用于配置 RTC_ALARM 输出的极性。 0: 当 ALAF/ALBF/WTF 标志位置 1(取决于 OUTSEL [1:0]), 该引脚输出高电平 1: 当 ALAF/ALBF/WTF 标志位置 1(取决于 OUTSEL [1:0]), 该引脚输出低电平

位域	名称	描述
19	CALOSEL	校准输出选择位。 当 COEN=1 时，该位选择在 RTC_CALIB 上输出哪个信号。 在 RTCCLK 为 32 kHz 且预分频为默认值(RTC_PRE.DIVA[6:0]=124 和 RTC_PRE.DIVS[14:0]=255)的条件下，这些频率有效。 0: 校准输出 256Hz 1: 校准输出 1Hz
18	BAKP	这个位可以由用户写入，以记住是否执行了夏令时的更改。
17	SU1H	减去 1 小时位(冬季时间更改)。 当设置此位时，如果当前小时不是 0，则从日历时间中减去 1 小时。这个位总是被读取为 0。当当前小时为 0 时，设置此位无效。 0: 无使用 1: 用当前时间减去 1 小时。这可以用于冬季改变户外初始化模式
16	AD1H	加 1 小时位(夏季时间更改)。 设置此位后，将 1 小时添加到日历时间中。这个位总是被读为 0。 0: 无使用 1: 用当前时间加上 1 小时。这可以用于夏季改变户外初始化模式
15	TSIEN	时间戳中断使能位。 0: 时间戳中断禁止 1: 时间戳中断开启
14	WTIEN	唤醒定时器中断使能位。 0: 唤醒定时器中断禁止 1: 唤醒定时器中断开启
13	ALBIEN	闹钟 B 中断使能位。 0: 闹钟 B 中断禁止 1: 闹钟 B 中断开启
12	ALAIEN	闹钟 A 中断使能位。 0: 闹钟 A 中断禁止 1: 闹钟 A 中断开启
11	TSEN	时间戳使能位。 0: 时间戳禁止 1: 时间戳开启
10	WTEN	唤醒定时器使能位。 0: 唤醒定时器禁止 1: 唤醒定时器开启
9	ALBEN	闹钟 B 使能位。 0: 闹钟 B 禁止 1: 闹钟 B 开启
8	ALAEEN	闹钟 A 使能位。 0: 闹钟 A 禁止 1: 闹钟 A 开启
7	Reserved	保留，必须保持复位值。
6	HFMT	小时格式位。 0: 24 小时格式

位域	名称	描述
		1: AM/PM 格式
5	BYPS	旁路影子寄存器位。 0: 日历值(从 RTC_SUBS、RTC_TSH 和 RTC_DATE 读取时)取自影子寄存器, 影子寄存器每两个 RTCCLK 周期更新一次。 1: 日历值(从 RTC_SUBS、RTC_TSH 和 RTC_DATE 读取时)直接从日历计数器中获取。 <i>注意: 如果 APB1 时钟的频率小于 RTCCLK 的 7 倍, 则 BYPS 必须设置为 1</i>
4	REFCLKEN	RTC_REFIN 参考时钟检测位(50 或 60hz)。 0: RTC_REFIN 检测禁止 1: RTC_REFIN 检测开启 <i>注意: DIVS 必须为 0x00FF</i>
3	TEDGE	时间戳事件触发沿配置位。 0: RTC_TS 输入上升沿生成一个时间戳事件 1: RTC_TS 输入下降沿生成一个时间戳事件 <i>注意: 更改 TEDGE 时必须重置 RTC_CTRL.TSEN, 以避免 RTC_INITSTS.TISF 意外置 1。</i>
2:0	WKUPSEL[2:0]	唤醒时钟选择位。 000: 选择 RTC/16 时钟 001: 选择 RTC/8 时钟 010: 选择 RTC/4 时钟 011: 选择 RTC/2 时钟 10x: 选择 ck_spre(通常 1Hz)时钟 11x: 选择 ck_spre(通常 1Hz)时钟并且唤醒定时器计数器值配置成 2 ¹⁶

23.3.4 RTC 日历时间寄存器 (RTC_TSH)

偏移地址: 0x08

复位值: 0x0000 0000

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

位域	名称	描述
31:23	Reserved	保留, 必须保持复位值
22	APM	AM/PM 格式。 0: AM 格式或者 24 小时格式 1: PM 格式

位域	名称	描述
21:20	HOT[1:0]	小时的十位(BCD 格式)。
19:16	HOU[3:0]	小时的个位(BCD 格式)。
15	Reserved	保留，必需保持复位值。
14:12	MIT [2: 0]	分钟的十位(BCD 格式)。
11:8	MIU[3:0]	分钟的个位(BCD 格式)。
7	Reserved	保留，必需保持复位值。
6:4	SCT[2:0]	秒的十位(BCD 格式)。
3:0	SCU[3:0]	秒的个位(BCD 格式)。

23.3.5 RTC 日历日期寄存器 (RTC_DATE)

偏移地址：0x0C

复位值：0x0000 2101

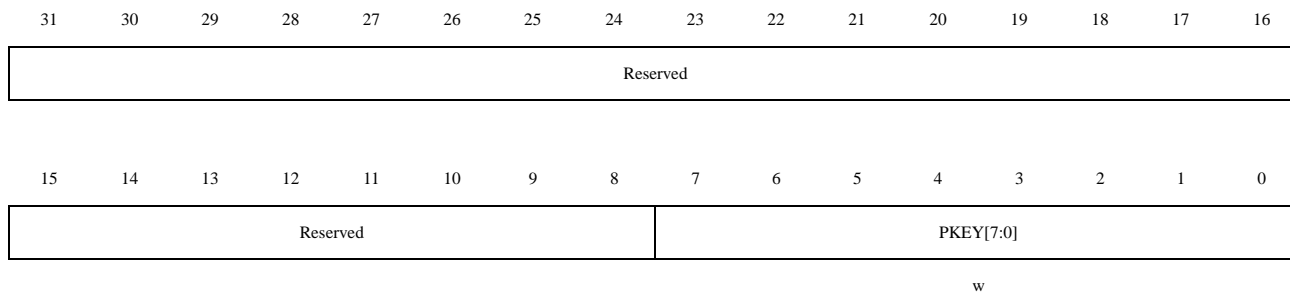
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								YRT[3:0]				YRU[3:0]			
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MOT	MOU[2:0]				Reserved		DAT[1:0]		DAU[3:0]			
rw			rw	rw				rw				rw			

位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:20	YRT[3:0]	年的十位(BCD 格式)。
19:16	YRU[3:0]	年的个位(BCD 格式)。
15:13	WDU[2:0]	星期几 000: 禁止 001: 星期一 ... 111: 星期天
12	MOT	月的十位(BCD 格式)。
11:8	MOU[3:0]	月的个位(BCD 格式)。
7:6	Reserved	保留，必须保持复位值。
5:4	DAT[1:0]	日期的十位(BCD 格式)。
3:0	DAU[3:0]	日期的个位(BCD 格式)。

23.3.6 RTC 写保护寄存器(RTC_WRP)

偏移地址：0x10

复位值：0x0000 0000

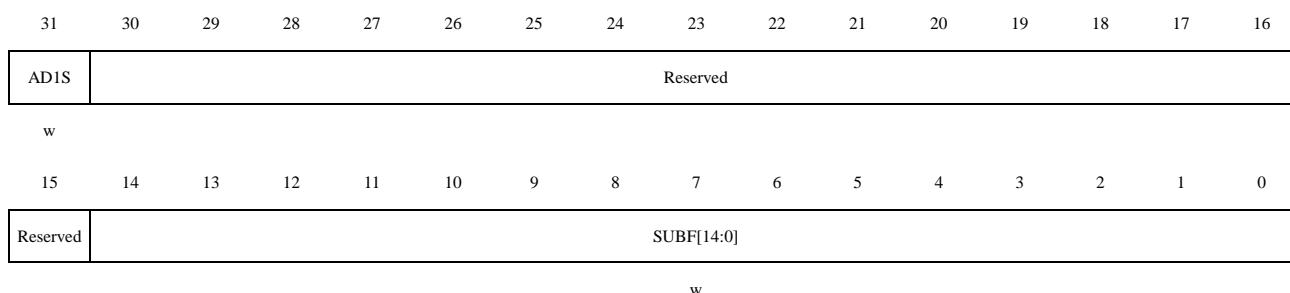


位域	名称	描述
31:8	Reserved	保留，必须保持复位值。
7:0	PKEY[7:0]	写保护密钥 读取该字节总是返回 0x00。 有关如何解锁 RTC 寄存器写保护的详细信息，请参阅 RTC 写保护寄存器章节。

23.3.7 RTC 平移控制寄存器(RTC_SCTRL)

偏移地址：0x14

复位值：0x0000 0000



位域	名称	描述
31	AD1S	加一秒 0：不加一秒。 1：时钟/日历增加一秒 该位只能写入且读取为零。当 RTC_INITSTS.SHOPF=1 时，写入该位没有影响。
30:15	Reserved	保留，必须保持复位值。
14:0	SUBF[14:0]	减去亚秒值位 这些位只能写入且读取为零。当 RTC_INITSTS.SHOPF=1 时，写入该位没有影响。写入 SUBF[14:0]的值被添加到同步预分频计数器，时钟将延迟： 延迟（秒）= (SUBF[14:0]+1) / (DIVS[14:0] + 1) AD1S 位可以与 SUBF[14:0] 位一起使用： 提前（秒）= (1 - ((SUBF[14:0]+1) / (DIVS[14:0] + 1)))。 注意：RTC_INITSTS.RSYF 位将在写入 SUBF[14:0] 时被清除。当

位域	名称	描述
		$RTC_INITSTS.RSYF=1$ 时, 影子寄存器已更新为平移后的时间。

23.3.8 RTC 亚秒寄存器(RTC_SUBS)

偏移地址: 0x18

复位值: 0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	SS[15:0]	亚秒值。 该值是同步预分频器计数器值。 此亚秒值由以下公式计算: 亚秒值 = $(RTC_PRE.DIVS[14:0]-SS)/(RTC_PRE.DIVS[14:0]+1)$ 注意: SS[15:0] 只有在移位操作完成后才能大于 $RTC_PRE.DIVS[14:0]$ 。 在这种情况下, 正确的时间/日期比 RTC_TSH/RTC_DATE 指示的时间/日期慢一秒。

23.3.9 RTC 时间戳时间寄存器 (RTC_TST)

偏移地址: 0x1C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rserved									APM	HOT[1:0]		HOU[3:0]			
									r	r		r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rserved	MIT[2:0]			MIU[3:0]				Rserved	SET[2:0]			SEU[3:0]			
r				r				r				r			

位域	名称	描述
31:23	Reserved	保留, 必须保持复位值。
22	APM	AM/PM 符号位。 0: AM 或 24 小时制

位域	名称	描述
		1: PM
21:20	HOT[1:0]	小时的十位(BCD 格式)。
19:16	HOU[3:0]	小时的个位(BCD 格式)。
15	Reserved	保留, 必须保持复位值。
14:12	MIT[2:0]	分钟的十位(BCD 格式)。
11:8	MIU[3:0]	分钟的个位(BCD 格式)。
7	Reserved	保留, 必须保持复位值。
6:4	SET[2:0]	秒的十位(BCD 格式)。
3:0	SEU[3:0]	秒的个位(BCD 格式)。

23.3.10 RTC 闹钟 A 寄存器(RTC_ALARM_A)

偏移地址: 0x20

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK4	WKDSEL	DTT[1:0]	DTU[3:0]	MASK3	APM	HOT[1:0]	HOU[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
MASK2	MIT[2:0]	MIU[3:0]	MASK1	SET[2:0]	SEU[3:0]										
rw	rw	rw	rw	rw	rw										

位域	名称	描述
31	MASK4	闹钟日期掩码位。 0: 日期/日匹配 1: 日期/日不匹配
30	WKDSEL	星期几选择位。 0: DTU[3:0]代表日期的个位 1: DTU[3:0]代表星期几。DTT[1:0]为无关位
29:28	DTT[1:0]	日期的十位(BCD 格式)。
27:24	DTU[3:0]	日期的个位(BCD 格式)
23	MASK3	闹钟小时掩码位。 0: 小时匹配 1: 小时不匹配
22	APM	AM/PM 符号位。 0: AM 或 24 小时制 1: PM
21:20	HOT[1:0]	小时的十位(BCD 格式)。
19:16	HOU[3:0]	小时的个位(BCD 格式)。
15	MASK2	闹钟分钟掩码位。

位域	名称	描述
		0: 分钟匹配 1: 分钟不匹配
14:12	MIT[2:0]	分钟的十位(BCD 格式)。
11:8	MIU[3:0]	分钟的个位(BCD 格式)。
7	MASK1	闹钟秒掩码位。 0: 秒匹配 1: 秒不匹配
6:4	SET[2:0]	秒的十位(BCD 格式)。
3:0	SEU[3:0]	秒的个位(BCD 格式)。

23.3.11 RTC 预分频寄存器(RTC_PRE)

偏移地址: 0x24

复位值: 0x007F 00FF

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

位域	名称	描述
31:23	Reserved	保留, 必须保持复位值。
22:16	DIVA[6:0]	异步分频参数位。 $f_{ck_apre} = RTCCLK/(DIVA[6:0]+1)$
15	Reserved	保留, 必须保持复位值。
14:0	DIVS[14:0]	同步分频位。 $f_{ck_spre} = f_{ck_apre}/(DIVS[14:0]+1)$

23.3.12 RTC 闹钟 B 寄存器 (RTC_ALARM B)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK4	WKDSEL	DTT[1:0]	DTU[3:0]			MASK3	APM	HOT[1:0]		HOU[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

MASK2	MIT[2:0]	MIU[3:0]	MASK1	SET[2:0]	SEU[3:0]
rw	rw	rw	rw	rw	rw

位域	名称	描述
31	MASK4	闹钟日期掩码位。 0: 日期/日匹配 1: 日期/日不匹配
30	WKDSEL	星期几选择位。 0: DTU[3:0]代表日期的个位 1: DTU[3:0]代表星期几。DTT[1:0]为无关位
29:28	DTT[1:0]	日期的十位(BCD 格式)。
27:24	DTU[3:0]	日期的个位(BCD 格式)
23	MASK3	闹钟小时掩码位。 0: 小时匹配 1: 小时不匹配
22	APM	AM/PM 符号位。 0: AM 或 24 小时制 1: PM
21:20	HOT[1:0]	小时的十位(BCD 格式)。
19:16	HOU[3:0]	小时的个位(BCD 格式)。
15	MASK2	闹钟分钟掩码位。 0: 分钟匹配 1: 分钟不匹配
14:12	MIT[2:0]	分钟的十位(BCD 格式)。
11:8	MIU[3:0]	分钟的个位(BCD 格式)。
7	MASK1	闹钟秒掩码位。 0: 秒匹配 1: 秒不匹配
6:4	SET[2:0]	秒的十位(BCD 格式)。
3:0	SEU[3:0]	秒的个位(BCD 格式)。

23.3.13 RTC 唤醒定时器寄存器(RTC_WKUPT)

偏移地址: 0x2C

复位值: 0x0000 FFFF

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

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

WKUPT[15:0]

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	WKUPT[15:0]	<p>唤醒自动重载值位</p> <p>当 RTC_CTRL.WTEN=1 时，每 (WKUPT[15:0] + 1) 个 ck_wut 周期设置 RTC_INITSTS.WTF 标志。当 RTC_CTRL.WKUPSEL[2]=1 时，唤醒定时器变为 17 位。该位不能配成 0。</p> <p>注意：</p> <p>这个寄存器的变化（如第二次设置或以后的设置）需要在唤醒中断中进行更改，否则更改后的设置不会立即生效，而是在下次唤醒后生效；特别是当 RTC_CTRL.WKUPSEL[2:0] 设置为 010 时，修改后的设置不会立即生效，而是在下一个周期唤醒后生效。</p>

23.3.14 RTC 入侵配置寄存器（RTC_TMPCFG）

偏移地址：0x30

复位值：0x0000 0000

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

Reserved	TP3MF	Reserved	TP3INTEN	TP2MF	Reserved	TP2INTEN	TP1MF	Reserved	TP1INTEN	TPPUDIS	TPPRCH[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

Reserved	TPFLT[1:0]	Reserved	TPFREQ[2:0]	TPTS	TP3TRG	TP3EN	TP2TRG	TP2EN	TPINTEN	TP1TRG	TP1EN
----------	------------	----------	-------------	------	--------	-------	--------	-------	---------	--------	-------

rw

rw

rw

rw

rw

rw

rw

rw

rw

rw

位域	名称	描述
31:28	Reserved	保留，必须保持复位值。
27	TP3MF	<p>入侵 3 掩码标志。</p> <p>0：不屏蔽入侵 3 事件。</p> <p>1：屏蔽入侵 3 事件。</p> <p>注意：当 TP3MF 置位时，不得使能 Tamper 3 中断。</p>
26	Reserved	保留，必须保持复位值。
25	TP3INTEN	<p>入侵 3 中断使能位。</p> <p>0：TPINTEN = 0 时禁止入侵 3 中断</p> <p>1：使能入侵 3 中断</p>
24	TP2MF	<p>入侵 2 掩码标志。</p> <p>0：不屏蔽入侵 2 事件。</p>

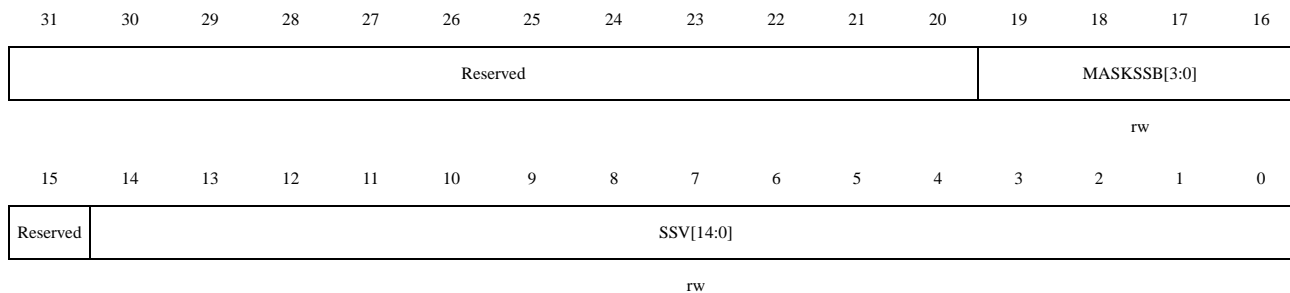
位域	名称	描述
		1: 屏蔽入侵 2 事件。 注意: 当 TP2MF 置位时, 不得使能 Tamper 2 中断。
23	Reserved	保留, 必须保持复位值。
22	TP2INTEN	入侵 2 中断使能位。 0: TPINTEN = 0 时禁止入侵 2 中断 1: 使能入侵 2 中断
21	TP1MF	入侵 1 掩码标志。 0: 不屏蔽入侵 1 事件。 1: 屏蔽入侵 1 事件。 注意: 当 TP1MF 置位时, 不得使能 Tamper 1 中断。
20	Reserved	保留, 必须保持复位值。
19	TP1INTEN	入侵 1 中断使能位。 0: TPINTEN = 0 时禁止入侵 1 中断 1: 使能入侵 1 中断
18	TPPUDIS	RTC_TAMPx 上拉禁用位。 0: 每次采样前启用预充电 RTC_TAMPx 引脚。 1: 禁用预充电 RTC_TAMPx 引脚
17:16	TPPRCH[1:0]	RTC_TAMPx 预充电持续时间。 这些位确定每次采样前的预充电时间。 0x0: 1 个 RTCCLK 周期 0x1: 2 个 RTCCLK 周期 0x2: 4 个 RTCCLK 周期 0x3: 8 个 RTCCLK 周期
15:14	Reserved	保留, 必须保持复位值。
13:12	TPFLT[1:0]	RTC_TAMPx 过滤器计数。 这些位决定在有效电平时的连续采样次数。 0x0: 在有效电平上 1 次采样后触发入侵事件 0x1: 在有效电平上连续 2 次采样后触发入侵事件 0x2: 在有效电平上连续 4 次采样后触发入侵事件 0x3: 在有效电平上连续 8 次采样后触发入侵事件
11	Reserved	保留, 必须保持复位值。
10:8	TPFREQ[2:0]	入侵采样频率。 该位决定对每个 RTC_TAMPx 输入进行采样时的频率。 0x0: 每 32768 个 RTCCLK 采样一次 0x1: 每 16384 个 RTCCLK 采样一次 0x2: 每 8192 个 RTCCLK 采样一次 0x3: 每 4096 个 RTCCLK 采样一次 0x4: 每 2048 个 RTCCLK 采样一次 0x5: 每 1024 个 RTCCLK 采样一次 0x6: 每 512 个 RTCCLK 采样一次 0x7: 每 256 个 RTCCLK 采样一次
7	TPTS	发生入侵检测事件时激活时间戳位。 0: 发生入侵检测事件时不保存时间戳

位域	名称	描述
		1: 发生入侵检测事件时保存时间戳 即便 RTC_CTRL.TSEN=0, TPTS 仍有效。
6	TP3TRG	入侵 3 事件触发模式。 如果 TPFLT[1:0] != 00, 入侵检测处于电平模式: 0: 低电平触发入侵检测事件。 1: 高电平触发入侵检测事件。 如果 TPFLT[1:0] = 00, 入侵检测处于边沿模式: 0: 上升沿触发入侵检测事件。 1: 下降沿触发入侵检测事件
5	TP3EN	RTC_TAMP3 检测使能位。 0: 禁止 RTC_TAMP3 输入检测 1: 开启 RTC_TAMP3 输入检测
4	TP2TRG	入侵 2 事件触发模式。 如果 TPFLT[1:0] != 00, 入侵检测处于电平模式: 0: 低电平触发入侵检测事件。 1: 高电平触发入侵检测事件。 如果 TPFLT[1:0] = 00, 入侵检测处于边沿模式: 0: 上升沿触发入侵检测事件。 1: 下降沿触发入侵检测事件
3	TP2EN	RTC_TAMP2 检测使能位。 0: 禁止 RTC_TAMP2 输入检测 1: 开启 RTC_TAMP2 输入检测
2	TPINTEN	入侵事件中断使能。 0: 禁止入侵中断 1: 使能入侵中断 注: 该位使能所有入侵引脚事件的中断, 与 TPxINTEN 电平无关。如果该位清零, 每个入侵事件中断可以通过设置 TPxINTEN 单独启用。
1	TP1TRG	入侵 1 事件触发模式。 如果 TPFLT[1:0] != 00, 入侵检测处于电平模式: 0: 低电平触发入侵检测事件。 1: 高电平触发入侵检测事件。 如果 TPFLT[1:0] = 00, 入侵检测处于边沿模式: 0: 上升沿触发入侵检测事件。 1: 下降沿触发入侵检测事件
0	TP1EN	RTC_TAMP1 检测使能位。 0: 禁止 RTC_TAMP1 输入检测 1: 开启 RTC_TAMP1 输入检测

23.3.15 RTC 闹钟 A 亚秒寄存器(RTC_ALRMAS)

偏移地址: 0x34

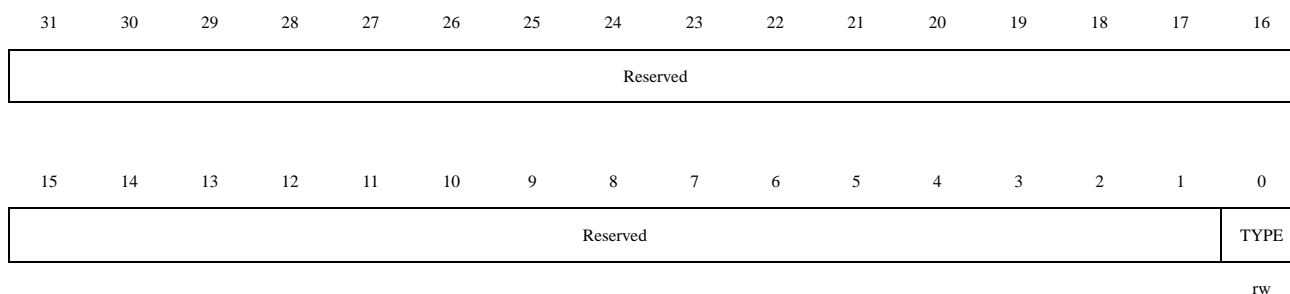
复位值: 0x0000 0000



位域	名称	描述
31:20	Reserved	保留，必须保持复位值。
19:16	MASKSSB[3:0]	屏蔽此位开始的最高有效位。 0x0: 闹钟的亚秒不比较。当秒单位增加时设置闹钟（假设其余字段匹配）。 0x1: 只比较 SS[0]，不比较其他位。 0x2: 仅比较 SS[1:0]，不比较其他位。 0x3: 仅比较 SS[2:0]，不比较其他位。 ... 0xC: 仅比较 SS[11:0]，不比较其他位。 0xD: 仅比较 SS[12:0]，不比较其他位。 0xE: 仅比较 SS[13:0]，不比较其他位。 0xF: 比较 SS[14:0] 从不比较同步计数器 RTC_SUBS.SS[15]位。
15	Reserved	保留，必须保持复位值。
14:0	SSV[14:0]	亚秒值。 该值与同步预分频计数器 RTC_SUBS.SS[14:0]进行比较，比较的位数由 MASKSSB[3:0]控制。

23.3.16 RTC 选项寄存器 (RTC_OPT)

偏移地址: 0x38 复位值: 0x0000 0000



位域	名称	描述
31:1	Reserved	保留，必须保持复位值。
0	TYPE	PC13 上的 RTC_ALARM 输出类型位。 0: 开漏输出

位域	名称	描述
		1: 推挽输出

23.3.17 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSS)

偏移地址: 0x3C

复位值: 0x0000 0000

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

位域	名称	描述
31:20	Reserved	保留, 必须保持复位值。
19:16	MASKSSB[3:0]	屏蔽此位开始的最高有效位。 0x0: 闹钟的亚秒不比较。当秒单位增加时设置闹钟 (假设其余字段匹配)。 0x1: 只比较 SS[0], 不比较其他位。 0x2: 仅比较 SS[1:0], 不比较其他位。 0x3: 仅比较 SS[2:0], 不比较其他位。 ... 0xC: 仅比较 SS[11:0], 不比较其他位。 0xD: 仅比较 SS[12:0], 不比较其他位。 0xE: 仅比较 SS[13:0], 不比较其他位。 0xF: 比较 SS[14:0]。 从不比较同步计数器 RTC_SUBS.SS[15]位。
15	Reserved	保留, 必须保持复位值。
14:0	SSV[14:0]	亚秒值 该值与同步预分频计数器 RTC_SUBS.SS[14:0]进行比较, 比较的位数由 MASKSSB[3:0] 控制。

23.3.18 RTC 校准寄存器(RTC_CALIB)

偏移地址: 0x40

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CP	CW8	CW16	CM[8:0]								
rw				rw	rw	rw	rw								

位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
11	CP	将 RTC 频率提高 488.5 ppm。 此功能与 CM[8:0]一起使用。当 RTCCLK 频率为 32768 Hz 时，在 32 秒窗口期间添加的 RTCCLK 脉冲数为 $((512 * CP) - CM[8:0])$ 。 0：不增加 RTCCLK 脉冲 1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲
10	CW8	使用 8 秒校准周期位。 0：不使用 1：选择 8 秒校准周期 注意：当 CW8 = 1 时，CM[1:0]将始终保持为‘00’
9	CW16	使用 16 秒校准周期位。 0：不使用 1：选择 16 秒校准周期，如果 CW8 = 1，则不能将该位置 1 注意：当 CW16 = 1 时，CM[0]将始终保持为‘0’
8:0	CM[8:0]	负校准位。 2^{20} 个 RTCCLK 脉冲中的屏蔽脉冲数。这有效地降低了分辨率为 0.9537 ppm 的日历频率。

23.3.19 RTC 时间戳亚秒寄存器(RTC_TSSS)

偏移地址：0x44

复位值：0x0000 0000

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

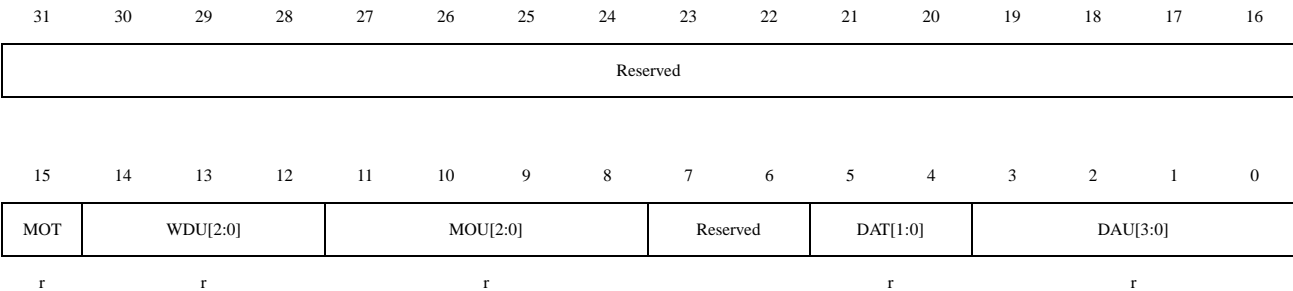
位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	SSE[15:0]	亚秒值 SSE[15:0] 是同步预分频计数器中的值。亚秒值由以下公式提供：

位域	名称	描述
		亚秒值 = (RTC_PRE.DIVS[14:0] – SSE[15:0]) / (RTC_PRE.DIVS[14:0] + 1) 注意: SSE[15:0] 只能在移位操作后大于 RTC_PRE.DIVS[14:0]。在这种情况下, 正确的时间/日期比 RTC_TSH/RTC_DATE 指示的时间少一秒。

23.3.20 RTC 时间戳日期寄存器 (RTC_TSD)

偏移地址: 0x48

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15	MOT	月份的十位(BCD 格式)。
14:12	WDU[2:0]	星期几 000: 禁止 001: 星期一 ... 111: 星期天
11:8	MOU[3:0]	月份的个位(BCD 格式)。
7:6	Reserved	保留, 必须保持复位值。
5:4	DAT[1:0]	日期的十位(BCD 格式)。
3:0	DAU[3:0]	日期的个位(BCD 格式)。



24 蜂鸣器（Beeper）

24.1 简介

Beeper 模块可以产生周期信号来驱动外部无源蜂鸣器。用于产生提示音或者报警发声。

24.2 功能描述

蜂鸣器 Beeper 作为一个独立的模块，挂载于 APB1 总线上，最高工作频率 32MHz。其中音色共分为 25 档，使用时通过配置相关寄存器实现不同音色可调。

24.3 Beeper 寄存器

必须以字（32 位）的方式操作这些外设寄存器。

24.3.1 Beeper 寄存器总览

表 24-1 Beeper 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	BEEPER_CTRL	Reserved																										FREQ_SEL[5:0]					BEEP_EN
	Reset Value																											0	0	0	0	0	0

24.3.2 Beeper 控制寄存器（BEEPER_CTRL）

地址偏移量：0x00

复位值：0x0000 0000

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

位域	名称	描述
31:7	Reserved	保留，必须保持复位值。
6:1	FREQ_SEL[5:0]	蜂鸣器输出频率选择：

位域	名称	描述
		<p>000000: 输出频率为 131Hz</p> <p>000001: 输出频率为 147Hz</p> <p>000010: 输出频率为 165Hz</p> <p>000011: 输出频率为 175Hz</p> <p>000100: 输出频率为 196Hz</p> <p>000101: 输出频率为 220Hz</p> <p>000110: 输出频率为 247Hz</p> <p>000111: 输出频率为 262Hz</p> <p>001000: 输出频率为 294Hz</p> <p>001001: 输出频率为 330Hz</p> <p>001010: 输出频率为 349Hz</p> <p>001011: 输出频率为 392Hz</p> <p>001100: 输出频率为 440Hz</p> <p>001101: 输出频率为 494Hz</p> <p>001110: 输出频率为 524Hz</p> <p>001111: 输出频率为 588Hz</p> <p>010000: 输出频率为 660Hz</p> <p>010001: 输出频率为 698Hz</p> <p>010010: 输出频率为 784Hz</p> <p>010011: 输出频率为 880Hz</p> <p>010100: 输出频率为 988Hz</p> <p>010101: 输出频率为 1KHz</p> <p>010110: 输出频率为 2KHz</p> <p>010111: 输出频率为 4KHz</p> <p>011000: 输出频率为 8KHz</p> <p>备注: 输出频率为理论值, 实际输出会受 LSI 频率产生偏差</p>
0	BEEP_EN	<p>蜂鸣器使能</p> <p>0: 蜂鸣器失能 (注意: <i>BEEPER_EN=0</i> 之后可通过禁能 <i>RCC_APBCLKEN.BEEPEREN</i> 来立即关掉 <i>Beeper</i> 输出)</p>

位域	名称	描述
		1: 蜂鸣器使能

25 调试支持（DBG）

25.1 简介

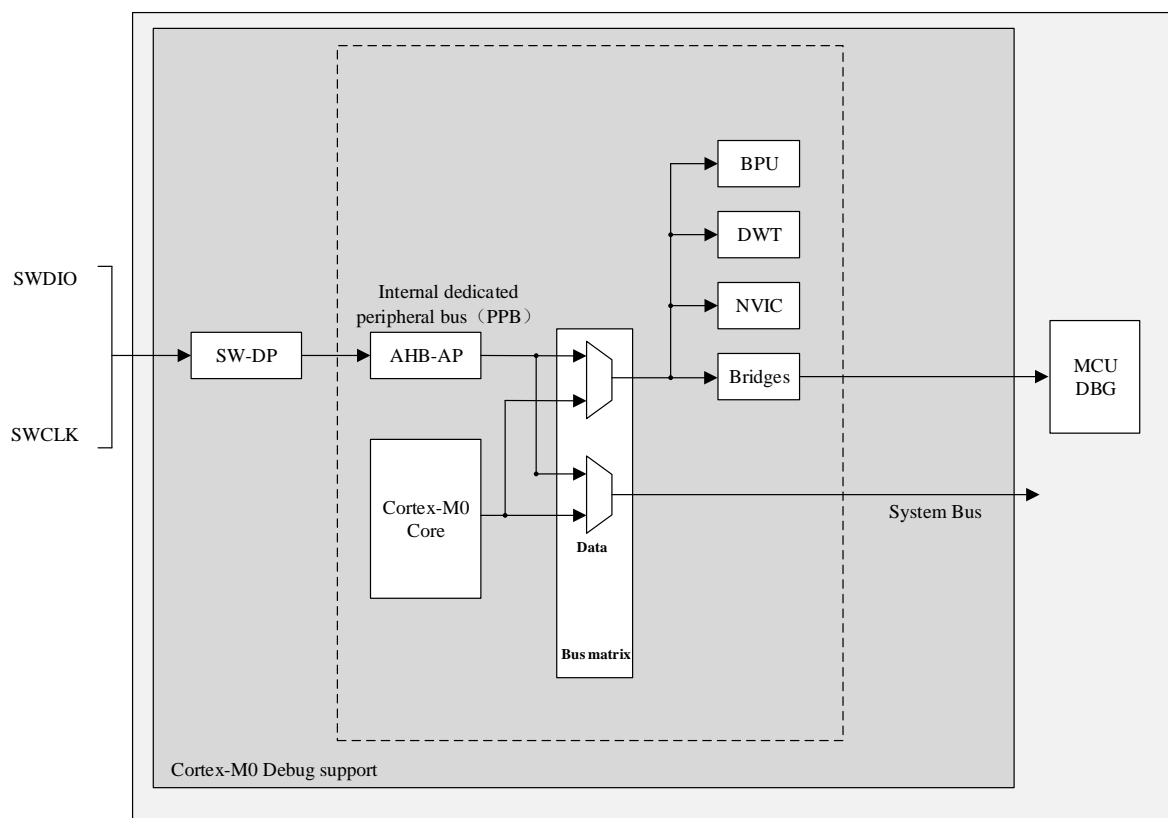
N32G05x 使用 Cortex®-M0 内核，内核集成硬件调试模块。支持指令断点（指令取值时停止）和数据断点（数据访问时停止）。当内核停止时，用户可以查看内核的内部状态和系统的外部状态。用户查询操作完成后，可以使内核和外设恢复，并继续执行相应程序。

N32G05X 内核的硬件调试模块在连接到调试器时即可被使用（在未被禁止情况下）。

N32G05X 支持以下调试接口：

■ 串行接口

图 25-1 N32G05X 级别和 Cortex®-M0 级别的调试框图



ARM Cortex®-M0 内核硬件调试模块可提供如下调试功能：

- SW-DP：串行调试端口
- AHP-AP：AHB 访问端口
- BPU：断点产生

■ DWT：数据触发

可参考：

- Cortex®-M0 技术参考手册（TRM）
- ARM 调试接口 V5 结构规范
- ARM CoreSight 开发工具集（r1p0 版）技术参考手册

25.2 SWD 功能

调试工具可以通过上述的 SWD 调试接口来调用调试功能。

25.2.1 引脚分配

SWD（串行调试）接口包含 2 个管脚：SWCLK（时钟管脚）和 SWDIO（数据输入输出管脚）。

SWD 调试接口管脚分配见下表：

表 25-1 调试端口引脚

调试端口	引脚分配
SWDIO	PA13
SWCLK	PA14

26 唯一设备序列号 (UID)

26.1 简介

MCU 系列产品内置两个不同长度的唯一设备序列号，分别为 96 位的 UID(Unique device ID)和 128 位的 UCID(Unique Customer ID)，这两个设备序列号存放在闪存存储器的系统配置块中，它们所包含的信息在出厂时编写，并保证对任意一个 MCU 微控制器在任何情况下都是唯一的，用户应用程序或外部设备可以通过 CPU 或 SWD 接口读取，不可被修改。

UID 为 96 位，通常用来作为序列号或作为密码，在编写闪存时，将此唯一标识与软件加解密算法相结合，进一步提高代码在闪存存储器内的安全性，也可用于激活带安全功能的自举程序(Secure Bootloader)。

UCID 为 128 位，遵守国民技术芯片序列号定义，它包含芯片生产及版本相关信息。

除以上两个设备序列号外，还有一个 32 位的 DBGMCU_ID，它包含了芯片版本号、芯片型号、Flash/SRAM 容量信息。

26.2 UID 寄存器

起始地址： 0x1FFF_F910 长度 96 位。

26.3 UCID 寄存器

起始地址： 0x1FFF_F8D0 长度 128 位。

26.4 DBGMCU_ID 寄存器

起始地址： 0x1FFFF920，长度 32 位。不同字节，低字节在前，高字节在后；同一字节，高位在前，低位在后。

表 26-1 DBGMCU_ID 位描述

描述	位数	备注
芯片版本号	4bit	芯片版本号低 4 位。
	4bit	芯片版本号高 4 位。
芯片型号	4bit	设备型号的高 4 位。 设备型号由高、中、低共 12 位组成，代表芯片的型号。
	4bit	设备型号的中 4 位。
	4bit	设备型号的低 4 位。
Flash 容量	4bit	FLASH 容量指示位。 16KB 为单位，FLASH 大小 = (N+1) * 16 KB
SRAM 容量	4bit	SRAM 容量指示位。 4KB 为单位，SRAM 大小 = (N+1) * 4KB
保留	4bit	保持为全 1。

27 版本历史

日期	版本号	修改点
2024/7/18	V1.0.0	1. 初始版本
2024/12/2	V1.1.0	<ol style="list-style-type: none"> GPIO 章节表号乱序问题修正 更新 COM 和 SEG 多路复用的描述 文中 USART 描述统一改为 UART 更新读保护配置列表 更新权限表 SRAM 启动 L1 级别下 system memory 区权限 删除 PWR_CTRL.PLS 和 PWR_CTRL2.LVRLS 寄存器位中 1.8V 档位描述 新增 SPI3_NSS 复用定义引脚 PD2 GPIOx_DS 寄存器描述更新 优化 DMA 章节的通道配置流程描述 更新 TIM 章节中的框图，删除 TIM1/8 字样，删除中文字样，删除全文中 TIM8 的字样 图 9-21 捕获/比较通道 1 主电路图更新 图 9-23 通道 x 的输出部分（x=4）删除 添加可编程的通道采样时间的注意事项 COMP 寄存器描述优化 LCD 闪烁功能描述优化 SPI 章节添加一些注意事项，删除 I2S 字样 RTC 章节删除 32.768KHz 相关信息 DBG 章节删除对 SWD 引脚的重复描述 GPIO 章节表号乱序问题修正 GPIO 与 EXTI 描述中没有 PD1/PD2 引脚的字样删除 删除 VLCD 可选电源描述

28 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。