

航芯 ACM32G103 用户手册

Version 1.3



上海航芯电子科技有限公司

<http://www.aisinochip.com>

条款协议

本文档的所有部分，其著作权归上海航芯电子科技有限公司（以下简称航芯科技）所有，未经航芯科技授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，航芯公司及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

AISINOCHEIP

版本修订

版本	日期	作者	描述
V1.0	2022-11-07	Aisinochip	初始化版本
V1.1	2022-12-07	Aisinochip	修改文字描述错误，增加图表说明。
V1.2	2022-04-20	Aisinochip	修改模块寄存器错误。
V1.3	2022-08-07	Aisinochip	COMP3 INP2 改为 PC3。COMP4 INP2 改为 PE9。

目录

版本修订	III
目录	1
图目录	21
表目录	25
1. 文档约定	27
1.1. 基本信息	27
1.2. 寄存器属性缩写表	27
1.3. 术语	27
2. 存储器及系统架构	28
2.1. 内核处理器	28
2.2. 系统架构	30
2.3. 存储器映射	31
2.4. 片上 SRAM	35
2.5. 片上 Flash	35
2.6. BOOT 配置	35
2.7. 芯片序列号	36
2.8. 系统配置寄存器 (SYSCFG)	36
2.8.1. 系统控制寄存器(SYSR) (偏移: 00h)	37
2.8.2. 工作模式寄存器(WMR) (偏移: 04h)	38
2.8.3. 版本寄存器(VER) (偏移: 08h)	39
2.8.4. USBPHY 配置寄存器(PHYCFG) (偏移: 10h)	39
3. 电源管理及低功耗 (PMU)	41
3.1. 供电电源	41
3.1.1. 主电源区	42
3.1.2. 待机电源区	43
3.1.3. 数字模块供电状态	43
3.2. 电源监控	44
3.2.1. POR 和 PDR	44
3.2.2. BOR 复位	44
3.2.3. LVD 模块	44
3.3. 低功耗模式	44
3.3.1. 低功耗模式描述	45
3.3.2. 低功耗模式下各模块工作状态	47
3.4. PMU 区引脚控制说明	48
3.5. PMU 寄存器	49
3.5.1. PMU 控制寄存器 0(PMU_CTRL0)(偏移: 00h)	49
3.5.2. PMU 控制寄存器 1(PMU_CTRL1)(偏移: 04h)	51
3.5.3. 电源控制寄存器 2(PMU_CTRL2) (偏移: 08h)	52
3.5.4. 电源控制寄存器 3 (PMU_CTRL3) (偏移: 0Ch)	54
3.5.5. PMU 状态寄存器(PMU_SR) (偏移: 10h)	54
3.5.6. PMU 状态清除寄存器(PMU_STCLR) (偏移: 14h)	55
3.5.7. STANDBY 域 IO 复用寄存器(PMU_IOSEL)(偏移: 18h)	56

4. 复位和时钟单元 (RCC)	59
4.1. 复位源	59
4.1.1. 上电复位	60
4.1.2. BOR 复位	60
4.1.3. POR12 复位	60
4.1.4. 外部引脚复位	60
4.1.5. EFC 复位	61
4.1.6. LVD 复位	61
4.1.7. LOCKUP 复位	61
4.1.8. Soft Reset	61
4.1.9. SYSREQRST	61
4.1.10. IWDG 复位	61
4.1.11. WDT 复位	61
4.2. 时钟结构图	62
4.3. 系统时钟选择	62
4.4. 时钟输出	63
4.5. 寄存器描述	64
4.5.1. 复位控制寄存器 RCC_RCR (偏移: 00h)	64
4.5.2. 复位源状态寄存器 RCC_RSR (偏移: 04h)	66
4.5.3. APB1 外设模块复位控制寄存器 RCC_APB1RSTR (偏移: 08h)	67
4.5.4. APB2 外设模块复位控制寄存器 RCC_APB2RSTR (偏移: 0Ch)	70
4.5.5. AHB 外设模块复位控制寄存器 RCC_AHBSTR (偏移: 10h)	72
4.5.6. 时钟控制寄存器 1(RCC_CCR1) (偏移: 14h)	74
4.5.7. 时钟控制寄存器 2 RCC_CCR2 (偏移: 18h)	74
4.5.8. 时钟中断寄存器 RCC_CIR (偏移: 1Ch)	77
4.5.9. APB1 外设模块时钟使能寄存器 RCC_APB1ENR (偏移: 20h)	79
4.5.10. APB2 外设模块时钟使能寄存器 RCC_APB2ENR (偏移: 24h)	81
4.5.11. AHB 外设模块时钟使能寄存器 RCC_AHBENR (偏移: 28h)	83
4.5.12. RCM64 模块控制寄存器 RCC_RC64MCR (偏移: 2Ch)	85
4.5.13. XTHCR 模块控制寄存器 RCC_XTHCR (偏移: 30h)	86
4.5.14. PLL 模块控制寄存器 RCC_PLLCR (偏移: 34h)	87
4.5.15. 时钟输出控制寄存器 RCC_CLKOCCR (偏移: 38h)	89
4.5.16. STANDBY 电源域控制寄存器 RCC_STDBYCTRL (偏移: 3Ch)	90
5. 嵌套矢量中断控制器 (NVIC)	93
5.1. 概述	93
5.2. 主要特性	93
5.3. 中断源	93
6. 外设内部互连	96
6.1. 概述	96
6.2. 定时器互连关系	96
6.2.1. 定时器内部触发源 (ITR) 互连关系	96
6.2.2. 定时器外部触发源 (ETR) 互连关系	97
6.2.3. OCxREF 清除输入信号互连关系	97
6.2.4. 定时器输入信号互连关系	98

6.2.5. 定时器刹车输入互连关系	99
7. 外部中断/事件控制器 (EXTI)	100
7.1. 概述	100
7.2. 主要特性	100
7.3. 外部中断及事件 (EXTI) 框图	100
7.4. 外部中断及事件 (EXTI) 触发源	101
7.5. 唤醒管理	102
7.6. 功能说明	102
7.6.1. EXTI 中断	102
7.6.2. EXTI 中断/事件唤醒	103
7.7. EXTI 寄存器	103
7.7.1. 中断使能寄存器 EXTI_IENR (偏移: 00h)	104
7.7.2. 事件使能寄存器 EXTI_EENR (偏移: 04h)	104
7.7.3. 上升沿触发使能寄存器 EXTI_RTENR (偏移: 08h)	104
7.7.4. 下降沿触发使能寄存器 EXTI_FTENR (偏移: 0Ch)	105
7.7.5. 软件中断事件寄存器 EXTI_SWIER (偏移: 10h)	105
7.7.6. 中断挂起寄存器 EXTI_PDR (偏移: 14h)	105
7.7.7. 外部中断配置寄存器 1 EXTI_CR1 (偏移: 18h)	106
7.7.8. 外部中断配置寄存器 2 EXTI_CR2 (偏移: 1Ch)	108
8. Flash 控制器 (EFC)	110
8.1. 概述	110
8.2. 主要特性	110
8.3. 功能描述	110
8.3.1. 主区闪存结构	110
8.3.2. 储存地址映射	111
8.4. 寄存器描述	111
8.4.1. 控制寄存器 EFC_CTRL (偏移: 0000h)	112
8.4.2. 写擦安全寄存器 EFC_SEC (偏移: 0004h)	114
8.4.3. 擦除超时寄存器 (EFC_ERTO) (偏移: 000ch)	114
8.4.4. 写超时寄存器 (EFC_WRTO) (偏移: 0010h)	114
8.4.5. 状态寄存器 EFC_STATUS (偏移: 0014h)	114
8.4.6. 中断状态寄存器 EFC_INTSTATUS (偏移: 0018h)	116
8.4.7. 中断使能寄存器 EFC_INEN (偏移: 001Ch)	116
8.5. 使用说明	117
8.5.1. Read 操作	117
8.5.2. Program 操作	117
8.5.3. Erase 操作	118
9. DMA 控制器 (DMA)	119
9.1. 概述	119
9.2. 主要特性	119
9.3. 结构框图	120
9.4. 使用说明	120
9.4.1. 外设请求选择	120
9.4.2. 传输模式	121

9.4.3. DMA 优先级	123
9.4.4. 软件注意事项	123
9.4.5. 启动 DMA 传输流程	124
9.4.6. 关闭 DMA 通道	124
9.4.7. DMA 链表使用流程	124
9.5. 寄存器描述	126
9.5.1. 中断状态寄存器 DMA_INTSTATUS (偏移: 00h)	127
9.5.2. 传输完成中断状态寄存器 DMA_INTTCSTATUS (偏移: 04h)	127
9.5.3. 传输完成中断清除寄存器 DMA_INTTCCLR (偏移: 08h)	128
9.5.4. 传输错误中断寄存器 DMA_INTERRSTATUS (偏移: 0Ch)	128
9.5.5. 传输错误中断清除寄存器 DMA_INTERRCLR (偏移: 10h)	128
9.5.6. 传输完成原始中断寄存器 DMA_RAWINTTCSTATUS (偏移: 14h)	129
9.5.7. 传输错误原始中断寄存器 DMA_RAWINTERRSTATUS (偏移: 18h)	129
9.5.8. 通道使能状态寄存器 DMA_ENCHSTATUS (偏移: 1Ch)	129
9.5.9. DMA 配置寄存器 DMA_CONFIG (偏移: 30h)	130
9.5.10. 通道 x 源地址寄存器 DMA_CHSRCADDR (偏移: 100h, 120h, 140h, 160h, 180h, 1A0h, 1C0h, 1E0h)	130
9.5.11. 通道 x 目标地址寄存器 DMA_CHDESTADDR (偏移: 104h, 124h, 144h, 164h, 184h, 1A4h, 1C4h, 1E4h)	130
9.5.12. 通道 x 链接表寄存器 DMA_CHLLI (偏移: 108h, 128h, 148h, 168h, 188h, 1A8h, 1C8h, 1E8h)	131
9.5.13. 通道 x 控制寄存器 DMA_CHCTRL (偏移: 10Ch, 12Ch, 14Ch, 16Ch, 18Ch, 1ACh, 1CCh, 1ECh)	131
9.5.14. 通道 x 配置寄存器 DMA_CHCONFIG (偏移: 110h, 130h, 150h, 170h, 190h, 1B0h, 1D0h, 1F0h)	133
10. 高级定时器 (TIM1/TIM8)	135
10.1. 概述	135
10.2. 主要特性	135
10.3. 功能描述	136
10.3.1. 定时器基本单元	136
10.3.2. 定时器计数模式	137
10.3.3. 预分频器	138
10.3.4. 重复计数器	139
10.3.5. 时钟源选择	140
10.3.6. 捕获比较通道	141
10.3.7. 强制输出模式	145
10.3.8. PWM 模式	145
10.3.9. 互补输出和死区插入	150
10.3.10. 刹车功能	151
10.3.11. 六步 PWM 输出	155
10.3.12. 正交编码器模式	155
10.3.13. 与霍尔传感器的接口	157
10.3.14. 单脉冲模式	159
10.3.15. 定时器互连	161

10.3.16. DMA 功能	164
10.3.17. 外部事件清除 OCxREF 信号	164
10.4. 寄存器描述	165
10.4.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)	166
10.4.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)	169
10.4.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)	173
10.4.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)	177
10.4.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)	179
10.4.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)	182
10.4.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18)	184
10.4.8. TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2 偏移: 0x1C)	189
10.4.9. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20)	191
10.4.10. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	195
10.4.11. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	195
10.4.12. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)	196
10.4.13. TIMx 重复计数寄存器 (TIMx_RCR 偏移: 0x30)	196
10.4.14. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)	197
10.4.15. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)	197
10.4.16. TIMx 捕获/比较寄存器 3 (TIMx_CCR3 偏移: 0x3C)	198
10.4.17. TIMx 捕获/比较寄存器 4 (TIMx_CCR4 偏移: 0x40)	198
10.4.18. TIMx 刹车和死区寄存器 (TIMx_BDTR 偏移: 0x44)	199
10.4.19. TIMx DMA 控制寄存器 (TIMx_DCR 偏移: 0x48)	202
10.4.20. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)	203
10.4.21. TIMx 捕获/比较模式寄存器 3 (TIMx_CCMR3 偏移: 0x54)	204
10.4.22. TIMx 捕获/比较寄存器 5 (TIMx_CCR5 偏移: 0x58)	206
10.4.23. TIMx 捕获/比较寄存器 6 (TIMx_CCR6 偏移: 0x5C)	206
10.4.24. TIMx 复用功能选择寄存器 1 (TIMx_AF1 偏移: 0x60)	207
10.4.25. TIMx 复用功能选择寄存器 2 (TIMx_AF2 偏移: 0x64)	208
10.4.26. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)	209
10.4.27. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)	210
11. 通用定时器 (TIM2)	212
11.1. 概述	212
11.2. 主要特性	212
11.3. 功能描述	213
11.3.1. 计数单元	213
11.3.2. 预分频器	213
11.3.3. 时钟源选择	213
11.3.4. 捕获比较通道	213
11.3.5. 强制输出模式	214
11.3.6. PWM 模式	214
11.3.7. 正交编码器模式	214
11.3.8. 与霍尔传感器的接口	214
11.3.9. 单脉冲模式	214
11.3.10. 定时器互连	214

11.3.11. DMA 功能	214
11.3.12. 外部事件清除 OCxREF 信号	214
11.4. 寄存器描述	215
11.4.1. TIM2 控制寄存器 1 (TIM2_CR1 偏移: 0x00)	216
11.4.2. TIM2 控制寄存器 2 (TIM2_CR2 偏移: 0x04)	218
11.4.3. TIM2 从模式控制寄存器 (TIM2_SMCR 偏移: 0x08)	220
11.4.4. TIM2 DMA/中断使能寄存器 (TIM2_DIER 偏移: 0x0C)	224
11.4.5. TIM2 状态寄存器 (TIM2_SR 偏移: 0x10)	226
11.4.6. TIM2 事件产生寄存器 (TIM2_EGR 偏移: 0x14)	228
11.4.7. TIM2 捕获/比较模式寄存器 1 (TIM2_CCMR1 偏移: 0x18)	230
11.4.8. TIM2 捕获/比较模式寄存器 2 (TIM2_CCMR2 偏移: 0x1C)	235
11.4.9. TIM2 捕获/比较使能寄存器 (TIM2_CCER 偏移: 0x20)	237
11.4.10. TIM2 计数器 (TIM2_CNT 偏移: 0x24)	241
11.4.11. TIM2 预分频器 (TIM2_PSC 偏移: 0x28)	241
11.4.12. TIM2 自动加载寄存器 (TIM2_ARR 偏移: 0x2c)	241
11.4.13. TIM2 捕获/比较寄存器 1 (TIM2_CCR1 偏移: 0x34)	242
11.4.14. TIM2 捕获/比较寄存器 2 (TIM2_CCR2 偏移: 0x38)	242
11.4.15. TIM2 捕获/比较寄存器 3 (TIM2_CCR3 偏移: 0x3C)	243
11.4.16. TIM2 捕获/比较寄存器 4 (TIM2_CCR4 偏移: 0x40)	243
11.4.17. TIM2 DMA 控制寄存器 (TIM2_DCR 偏移: 0x48)	244
11.4.18. TIM2 连续模式的 DMA 地址 (TIM2_DMAR 偏移: 0x4C)	245
11.4.19. TIM2 复用功能选择寄存器 1 (TIM2_AF1 偏移: 0x60)	245
11.4.20. TIM2 复用功能选择寄存器 2 (TIM2_AF2 偏移: 0x64)	246
11.4.21. TIM2 输入选择寄存器 (TIM2_TISEL 偏移: 0x68)	247
11.4.22. TIM2 DMA 请求类型选择寄存器 (TIM2_DBER 偏移: 0x6C)	248
12. 通用定时器 (TIM3/TIM4)	250
12.1. 概述	250
12.2. 主要特性	250
12.3. 结构框图	251
12.4. 功能描述	251
12.4.1. 计数单元	251
12.4.2. 预分频器	251
12.4.3. 时钟源选择	251
12.4.4. 捕获比较通道	252
12.4.5. 强制输出模式	252
12.4.6. PWM 模式	252
12.4.7. 正交编码器模式	252
12.4.8. 与霍尔传感器的接口	252
12.4.9. 单脉冲模式	252
12.4.10. 定时器互连	252
12.4.11. DMA 功能	252
12.4.12. 外部事件清除 OCxREF 信号	252
12.5. 寄存器描述	253
12.5.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)	254

12.5.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)	256
12.5.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)	258
12.5.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)	262
12.5.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)	264
12.5.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)	266
12.5.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18)	268
12.5.8. TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2 偏移: 0x1C)	273
12.5.9. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20)	275
12.5.10. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	278
12.5.11. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	278
12.5.12. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)	278
12.5.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)	278
12.5.14. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)	279
12.5.15. TIMx 捕获/比较寄存器 3 (TIMx_CCR3 偏移: 0x3C)	279
12.5.16. TIMx 捕获/比较寄存器 4 (TIMx_CCR4 偏移: 0x40)	280
12.5.17. TIMx DMA 控制寄存器 (TIMx_DCR 偏移: 0x48)	280
12.5.18. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)	282
12.5.19. TIMx 复用功能选择寄存器 1 (TIMx_AF1 偏移: 0x60)	282
12.5.20. TIMx 复用功能选择寄存器 2 (TIMx_AF2 偏移: 0x64)	283
12.5.21. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)	283
12.5.22. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)	285
13. 基本定时器 (TIM6/TIM7)	286
13.1. 概述	286
13.2. 主要特性	286
13.3. 结构框图	286
13.4. 功能描述	286
13.4.1. 计数单元	286
13.4.2. 预分频器	287
13.4.3. 时钟源选择	287
13.4.4. 定时器互连	287
13.4.5. DMA 功能	287
13.5. 寄存器描述	287
13.5.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)	288
13.5.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)	289
13.5.3. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)	290
13.5.4. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)	290
13.5.5. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)	291
13.5.6. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	291
13.5.7. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	292
13.5.8. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)	292
13.5.9.	292
14. 通用定时器 (TIM15/TIM16/TIM17)	293
14.1. 概述	293
14.2. 主要特性	293

14.3. 结构框图	294
14.4. 功能描述	295
14.4.1. 计数单元	295
14.4.2. 预分频器	295
14.4.3. 时钟源选择	295
14.4.4. 捕获比较通道	295
14.4.5. 强制输出模式	295
14.4.6. PWM 模式	295
14.4.7. 互补输出和死区插入	295
14.4.8. 刹车功能	295
14.4.9. 六步 PWM 输出	295
14.4.10. 单脉冲模式	296
14.4.11. 定时器互连	296
14.4.12. DMA 功能	296
14.4.13. 外部事件清除 OCxREF 信号	296
14.5. 寄存器描述	297
14.5.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)	298
14.5.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)	301
14.5.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)	303
14.5.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)	305
14.5.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)	306
14.5.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)	309
14.5.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18)	310
14.5.8. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20)	315
14.5.9. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	317
14.5.10. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	318
14.5.11. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)	318
14.5.12. TIMx 重复计数寄存器 (TIMx_RCR 偏移: 0x30)	318
14.5.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)	319
14.5.14. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)	319
14.5.15. TIMx 刹车和死区寄存器 (TIMx_BDTR 偏移: 0x44)	320
14.5.16. TIMx DMA 控制寄存器 (TIMx_DCR 偏移: 0x48)	323
14.5.17. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)	324
14.5.18. TIMx 复用功能选择寄存器 1 (TIMx_AF1 偏移: 0x60)	325
14.5.19. TIMx 复用功能选择寄存器 2 (TIMx_AF2 偏移: 0x64)	326
14.5.20. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)	327
14.5.21. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)	327
14.5.22.	328
15. 低功耗定时器 (LPTIM)	329
15.1. 概述	329
15.2. 主要特性	329
15.3. 结构框图	330
15.4. 功能说明	330
15.4.1. 触发映射	330

15.4.2. 复位和时钟	331
15.4.3. 干扰滤波器	332
15.4.4. 预分频器	332
15.4.5. 触发多路复用器	333
15.4.6. 工作模式	334
15.4.7. 超时功能	336
15.4.8. 生成波形	336
15.4.9. 寄存器更新	338
15.4.10. 计数器模式	339
15.4.11. 定时器使能	339
15.4.12. 定时器计数器复位	340
15.4.13. 编码器模式	340
15.4.14. 重复计数器	342
15.4.15. 调试模式	343
15.5. LPTIM 低功耗模式	343
15.6. LPTIM 中断	344
15.7. LPTIM 寄存器	345
15.7.1. 中断和状态寄存器 LPTIM_ISR (偏移: 00h)	345
15.7.2. 中断清零寄存器 LPTIM_ICR (偏移: 04h)	348
15.7.3. 中断使能寄存器 LPTIM_IER (偏移: 08h)	349
15.7.4. 配置寄存器 1 LPTIM_CFGR1 (偏移: 0Ch)	350
15.7.5. 控制寄存器 LPTIM_CR (偏移: 10h)	355
15.7.6. 比较寄存器 LPTIM_CMP (偏移: 14h)	357
15.7.7. 自动重载寄存器 LPTIM_ARR (偏移: 18h)	358
15.7.8. 计数器寄存器 LPTIM_CNT (偏移: 1Ch)	358
15.7.9. 配置寄存器 2 LPTIM_CFGR2 (偏移: 24h)	358
15.7.10. 重复寄存器 LPTIM_RCR (偏移: 28h)	359
16. 看门狗 (WDT)	361
16.1. 概述	361
16.2. 主要特性	361
16.3. 结构框图	361
16.4. 寄存器描述	362
16.4.1. 加载寄存器 WDT_LOAD (偏移: 00h)	362
16.4.2. 当前计数寄存器 WDT_COUNT (偏移: 04h)	362
16.4.3. 控制寄存器 WDT_CTRL (偏移: 08h)	363
16.4.4. 喂狗寄存器 WDT_FEED (偏移: 0Ch)	363
16.4.5. 中断清除时限寄存器 WDT_INTCLRTIME (偏移: 10h)	364
16.4.6. 原始中断状态寄存器 WDT_RIS (偏移: 14h)	364
16.5. 使用流程	364
16.5.1. 定时器溢出产生中断	364
16.5.2. 定时器溢出产生复位	365
17. 独立看门狗 (IWDG)	366
17.1. 概述	366
17.2. 主要特性	366

17.3. 功能描述	366
17.3.1. 寄存器访问保护	367
17.3.2. 窗口选项	367
17.3.3. 唤醒功能	367
17.4. 寄存器描述	368
17.4.1. 命令寄存器 IWDT_CMDR (偏移: 00h)	368
17.4.2. 预分频寄存器 IWDT_PR (偏移: 04h)	369
17.4.3. 重装载寄存器 IWDT_RLR (偏移: 08h)	369
17.4.4. 状态寄存器 IWDT_SR (偏移: 0Ch)	370
17.4.5. 窗口寄存器 IWDT_WINR (偏移: 10h)	370
17.4.6. 唤醒寄存器 IWDT_WUTR (偏移: 14h)	371
17.5. 使用流程	371
18. 实时时钟 (RTC)	373
18.1. 概述	373
18.2. 主要特性	373
18.3. RTC 功能描述	373
18.3.1. RTC 时间设置	374
18.3.2. RTC 时间读取	375
18.3.3. 闹钟功能	375
18.3.4. 时钟误差补偿	376
18.3.5. FSEL_OUT	377
18.3.6. 周期中断唤醒	377
18.3.7. 唤醒定时器	377
18.3.8. 侵入检测和备份寄存器	378
18.4. 寄存器描述	379
18.4.1. 写保护寄存器 RTC_WP (偏移: 00h)	380
18.4.2. 中断使能寄存器 RTC_IE (偏移: 04h)	380
18.4.3. 中断标志寄存器 RTC_SR (偏移: 08h)	382
18.4.4. 秒计数寄存器 RTC_SEC (偏移: 0Ch)	384
18.4.5. 分钟计数寄存器 RTC_MIN (偏移: 10h)	385
18.4.6. 小时计数寄存器 RTC_HOUR (偏移: 14h)	385
18.4.7. 日计数寄存器 RTC_DAY (偏移: 18h)	385
18.4.8. 周计数寄存器 RTC_WEEK (偏移: 1Ch)	385
18.4.9. 周计数寄存器 RTC_MONTH (偏移: 20h)	386
18.4.10. 年计数寄存器 RTC_YEAR (偏移: 24h)	386
18.4.11. 闹钟寄存器 RTC_ALM (偏移: 28h)	386
18.4.12. 控制寄存器 RTC_CR (偏移: 2Ch)	387
18.4.13. 时钟误差补偿寄存器 RTC_ADJUST (偏移: 30h)	390
18.4.14. 时间戳 1 RTC_CLKSTAMP1 (偏移: 44h)	390
18.4.15. 日历戳 1 RTC_CALSTAMP1 (偏移: 48h)	391
18.4.16. 时间戳 2 RTC_CLKSTAMP2 (偏移: 4Ch)	391
18.4.17. 日历戳 2 RTC_CALSTAMP2 (偏移: 50h)	391
18.4.18. 唤醒定时器寄存器 RTC_WUTR (偏移: 54h)	392
18.4.19. 备份寄存器 0~15 RTC_BAKUP0~15 (偏移: 70~ACh)	392

18.5. 使用流程	392
19. 通用输入输出接口 (GPIO)	393
19.1. 概述	393
19.2. 主要特性	393
19.3. 功能描述	393
19.3.1. 结构框图	394
19.3.2. 数字输入功能	394
19.3.3. 数字输出功能	394
19.3.4. 模拟功能	395
19.3.5. 复用功能	395
19.3.6. 外部中断/事件线	395
19.3.7. GPIO 锁定功能	395
19.4. 寄存器描述	395
19.4.1. 模式寄存器 GPIOx_MD (x=A to F) (偏移: 00h)	396
19.4.2. GPIO 输出类型寄存器 GPIOx_OTYP (x=A to F) (偏移: 04h)	397
19.4.3. GPIO 上下拉寄存器 GPIOx_PUPD (x=A to F) (偏移: 08h)	397
19.4.4. 输入引脚映射寄存器 GPIOx_IDATA (x=A to F) (偏移: 0Ch)	397
19.4.5. 输出引脚映射寄存器 GPIOx_ODATA (x=A to F) (偏移: 10h)	398
19.4.6. 输出置位/清零寄存器 GPIOx_BSC (x=A to F) (偏移: 14h)	398
19.4.7. 复用功能配置寄存器 0 GPIOx_AF0 (x=A to F) (偏移: 18h)	398
19.4.8. 复用功能配置寄存器 1 GPIOx_AF1 (x=A to F) (偏移: 1Ch)	399
19.4.9. 驱动能力配置寄存器 0 GPIOx_DS0 (x=A to F) (偏移: 20h)	399
19.4.10. 驱动能力配置寄存器 1 GPIOx_DS1 (x=A to F) (偏移: 24h)	400
19.4.11. 施密特使能寄存器 GPIOx_SMIT (x=A to F) (偏移: 28h)	401
19.4.12. 配置锁定寄存器 GPIOx_LOCK (x=A to F) (偏移: 2Ch)	401
19.4.13. 模拟开关配置寄存器 GPIOx_AIEN (x=A to F) (偏移: 30h)	402
20. 串行外设接口 (SPI)	404
20.1. 概述	404
20.2. 主要特性	404
20.3. 结构框图	404
20.4. 功能描述	405
20.4.1. 时序图	405
20.4.2. 时钟波特率设置	409
20.4.3. DMA 请求	409
20.4.4. 存储器读取模式	409
20.5. 寄存器描述	410
20.5.1. SPI 发送数据寄存器 SPI_TX_DAT(偏移 00h)	411
20.5.2. SPI 接收数据寄存器 SPI_RX_DAT(偏移 00h)	411
20.5.3. SPI 波特率设置寄存器 SPI_BAUD(偏移: 04h)	411
20.5.4. SPI 控制寄存器 SPI_CTL(偏移: 08h)	411
20.5.5. SPI 发送控制寄存器 SPI_TX_CTL (偏移: 0Ch)	413
20.5.6. SPI 接收控制寄存器 SPI_RX_CTL (偏移: 10h)	414
20.5.7. SPI 中断控制寄存器 SPI_IE (偏移: 14h)	415
20.5.8. SPI 状态寄存器 SPI_STATUS (偏移: 18h)	417

20.5.9. SPI 发送等待寄存器 SPI_TXDelay(偏移: 1Ch).....	420
20.5.10. SPI 批量传输数据个数寄存器 SPI_BATCH (偏移: 20h).....	420
20.5.11. SPI 从设备选择寄存器 SPI_CS(偏移: 24h).....	420
20.5.12. SPI 管脚输出方向 SPI_OUT_EN(偏移: 28h).....	420
20.5.13. SPIA 取值控制寄存器 SPI_MEMO_ACC(偏移: 2Ch).....	421
20.5.14. SPIA 取值命令寄存器 SPI_CMD(偏移: 30h).....	423
20.5.15. SPIA 取值参数寄存器 SPI_PARA(偏移: 34h).....	423
20.6. 使用流程.....	423
20.6.1. SPI 主模式发送.....	423
20.6.2. SPI 主模式接收.....	424
20.6.3. SPI 主模式接收时 Dummy 控制位.....	424
20.6.4. SPI 从模式发送时 Dummy 控制位.....	425
20.6.5. SPI 从模式接收时 Dummy 控制位.....	425
20.6.6. SPI 从模式 NCS 功能.....	425
20.6.7. SPI 存储器读取.....	425
20.6.8. SPI 存储器读取 (连读)	426
20.6.9. SPI SRAM 写入.....	426
21. 通用异步收发器 (UART)	428
21.1. 概述.....	428
21.2. 主要特性.....	428
21.3. 结构框图.....	429
21.4. 功能描述.....	429
21.4.1. CTS 和 RTS 流控功能.....	429
21.4.2. DMA 请求.....	430
21.4.3. LIN 总线功能.....	430
21.4.4. IrDA SIR 功能.....	431
21.4.5. 单线模式.....	432
21.4.6. 智能卡主模式.....	432
21.4.7. 多机通信.....	433
21.4.8. 波特率自适应.....	434
21.4.9. RS485 的控制器支持.....	435
21.4.10. 总线空闲检测.....	435
21.5. 寄存器描述.....	436
21.5.1. 数据寄存器 UART_DR(偏移: 00h).....	437
21.5.2. 标志位寄存器 UART_FR(偏移: 04h).....	437
21.5.3. 波特率寄存器 UART_BRR(偏移: 08h).....	439
21.5.4. 中断使能寄存器 UART_IE (偏移: 0Ch).....	439
21.5.5. 中断和状态寄存器 UART_ISR (偏移: 10h).....	441
21.5.6. 控制寄存器 1 UART_CR1 (偏移: 14h).....	443
21.5.7. 控制寄存器 2 UART_CR2 (偏移: 18h).....	445
21.5.8. 控制寄存器 3 UART_CR3 (偏移: 1Ch).....	446
21.5.9. 保护时间和预分频寄存器 UART_GTPR(偏移: 20h).....	448
21.5.10. 比特计时寄存器 UART_BCNT (偏移: 24h).....	449
21.6. 使用流程.....	450

21.6.1. 串口设置	450
21.6.2. 串口的发送和接收	450
21.6.3. LIN 硬件功能支持	450
21.6.4. IrDA SIR 功能使用流程	451
21.6.5. 单线模式功能使用流程	451
21.6.6. 智能卡功能使用流程	451
21.6.7. 多机通信功能使用流程	452
21.6.8. 波特率自适应功能使用流程	452
21.6.9. RS485 的 DE 控制功能使用流程	453
22. 内部集成电路总线接口 (I2C)	454
22.1. 概述	454
22.2. 主要特性	454
22.3. 结构框图	455
22.4. 功能描述	456
22.4.1. 模式选择	456
22.4.2. I2C 从模式	456
22.4.3. I2C 主模式	458
22.4.4. SCL 总线滤波算法	459
22.4.5. SCL 为低时检测 SDA 的跳变	460
22.4.6. TXE 状态	460
22.4.7. SMBus 超时配置	461
22.5. 寄存器描述	462
22.5.1. I2C 设备地址寄存器 1 I2C_SLAVE_ADDR1(偏移: 0x0h)	463
22.5.2. I2C 时钟分频寄存器 I2C_CLK_DIV(偏移: 0x04h)	463
22.5.3. 控制寄存器 I2C_CR(偏移: 0x08h)	463
22.5.4. 状态寄存器 I2C_SR(偏移: 0x0Ch)	465
22.5.5. 数据寄存器 I2C_DR(偏移: 0x10h)	468
22.5.6. I2C 设备地址寄存器 2/3 I2C_SLAVE_ADDR2/3 (偏移: 0x14h)	468
22.5.7. SDA 跳变阈值寄存器 I2C_DET(偏移: 0x18)	468
22.5.8. I2C 滤波寄存器 I2C_FILTER(偏移: 0x1C)	468
22.5.9. I2C 超时配置寄存器 I2C_TIMEOUT(偏移: 0x24h)	469
22.6. 软件操作流程	470
22.6.1. 作为主发送器	470
22.6.2. 作为主接收器	472
22.6.3. 作为从发送器	474
22.6.4. 作为从接收器	475
23. 音频接口 (I2S)	476
23.1. 概述	476
23.2. 主要特性	476
23.3. 结构框图	477
23.4. 寄存器描述	477
23.4.1. 数据寄存器 I2S_DR (偏移: 00h)	477
23.4.2. 控制寄存器 I2S_CR (偏移: 04h)	478
23.4.3. 时钟预分频寄存器 I2S_PR (偏移: 08h)	479

23.4.4. 中断使能寄存器 I2S_DIER (偏移: 0Ch)	480
23.4.5. I2S 状态寄存器 I2S_SR (偏移: 10h)	481
23.5. I2S 音频标准	483
23.5.1. I2S 飞利浦标准	483
23.5.2. MSB 对齐标准	485
23.5.3. LSB 对齐标准	486
23.5.4. PCM 对齐标准	487
23.6. 时钟	490
23.7. 使用流程	491
23.7.1. 主机发送	491
23.7.2. 主机接收	492
23.7.3. 从机发送	492
23.7.4. 从机接收	493
24. 低功耗串口 (LPUART)	494
24.1. 概述	494
24.2. 主要特性	494
24.3. 功能描述	494
24.3.1. 时钟选择	495
24.3.2. 波特率设置	495
24.3.3. STOP 模式唤醒	496
24.3.4. DMA 请求	497
24.4. 寄存器描述	497
24.4.1. 接收数据寄存器/LPUART_RXDR (偏移: 00h)	497
24.4.2. 发送数据寄存器/LPUART_TXDR (偏移: 04h)	498
24.4.3. 线控寄存器/LPUART_LCR (偏移: 08h)	498
24.4.4. 控制寄存器/LPUART_CR (偏移: 0Ch)	499
24.4.5. 波特率整数部分/LPUART_IBAUD (偏移: 10h)	500
24.4.6. 波特率小数部分/LPUART_FBAUD (偏移: 14h)	500
24.4.7. 中断使能寄存器/LPUART_IE (偏移: 18h)	500
24.4.8. 状态寄存器 LPUART_SR (偏移: 1Ch)	502
24.4.9. 地址寄存器 LPUART_ADDR (偏移: 20h)	503
24.5. 使用流程	503
24.5.1. 串口收发	503
24.5.2. 配置 STOP 唤醒	503
25. 模数转换器 (ADC)	505
25.1. 概述	505
25.2. 主要特性	505
25.3. 结构框图	507
25.4. 功能描述	507
25.4.1. 通道选择	507
25.4.2. 单次转换模式	509
25.4.3. 连续转换模式	510
25.4.4. 间断模式	510
25.4.5. 注入通道管理	511

25.4.6. 停止控制	512
25.4.7. 时序图	512
25.4.8. 模拟看门狗	512
25.4.9. 数据对齐	513
25.4.10. 偏移补偿	515
25.4.11. 可编程的通道采样时间	516
25.4.12. 外部触发转换	516
25.4.13. DMA 请求	517
25.4.14. 双 ADC 模式	518
25.4.15. 温度传感器	524
25.4.16. 差分信号转换	524
25.4.17. 溢出控制	524
25.4.18. 差分模式和有符号数	525
25.4.19. 过采样	525
25.4.20. ADC 中断	526
25.5. 寄存器描述	527
25.5.1. ADC 状态寄存器(ADC_SR 偏移 00h)	528
25.5.2. ADC 中断使能寄存器(ADC_IE 偏移 04h)	529
25.5.3. ADC 控制寄存器 1 (ADC_CR1 偏移 08h)	531
25.5.4. ADC 控制寄存器 2 (ADC_CR2 偏移 0ch)	534
25.5.5. ADC 采样时间寄存器 1 (ADC_SMPR1 偏移 10h)	537
25.5.6. ADC 采样时间寄存器 2 (ADC_SMPR2 偏移 14h)	538
25.5.7. ADC 采样时间寄存器 3 (ADC_SMPR3 偏移 18h)	538
25.5.8. ADC 看门狗高阈值寄存器(ADC_HTR 偏移 1Ch)	539
25.5.9. ADC 看门狗低阈值寄存器(ADC_LTR 偏移 20h)	539
25.5.10. ADC 规则序列寄存器 1(ADC_SQR1 偏移 24h)	540
25.5.11. ADC 规则序列寄存器 2(ADC_SQR2 偏移 28h)	541
25.5.12. ADC 规则序列寄存器 3(ADC_SQR3 偏移 2Ch)	541
25.5.13. ADC 注入通道寄存器(ADC_JSQR 偏移 30h)	541
25.5.14. ADC 注入数据寄存器 x(x 为 1~4)(ADC_JDRx 偏移 34~40h)	542
25.5.15. ADC 规则数据寄存器(ADC_DR 偏移 48h)	543
25.5.16. ADC 单端/差分选择寄存器 (ADC_DIFF 偏移 4Ch)	543
25.5.17. ADC 符号数选择寄存器 (ADC_SIGN 偏移 50h)	545
25.5.18. ADC 偏移寄存器 x (x 为 1~4) (ADC_OFRx 偏移 60~6Ch)	546
25.5.19. ADC 模拟配置寄存器 (ADC_ANACFG 偏移 80h)	547
25.5.20. ADC 共用状态寄存器 (ADC_CSR 偏移 300h)	547
25.5.21. ADC 共用控制寄存器 (ADC_CCR 偏移 304h)	549
25.5.22. ADC 共用规则数据寄存器 (ADC_CDR 偏移 308h)	550
25.5.23. ADC 共用温度传感器/REF 寄存器 (ADC_CTSREF 偏移 30Ch)	550
25.6. 使用流程	551
25.6.1. ADC 操作流程	551
25.6.2. 温度传感器操作	552
26. 运算放大器 (OPA)	553
26.1. 概述	553

26.2. 主要特性	553
26.3. 功能描述	553
26.3.1. 功能示意图	553
26.3.2. 操作模式和校准	554
26.3.3. 工作模式和负端输入	554
26.3.4. 负端输入管脚	554
26.3.5. 运放外置模式（SA 模式）	554
26.3.6. 运放单位增益模式（UG 模式）	555
26.3.7. 可编程增益放大模式（PGA 模式）	556
26.4. 寄存器描述	557
26.4.1. OPA1 控制寄存器(OPA_CSR1 偏移 00h)	557
26.4.2. OPA2 控制寄存器(OPA_CSR2 偏移 04h)	559
26.4.3. OPA3 控制寄存器(OPA_CSR3 偏移 08h)	561
26.5. 使用流程	564
26.5.1. 普通使用流程	564
26.5.2. 使用校准功能流程	564
27. 模拟比较器（COMP）	565
27.1. 概述	565
27.2. 主要特性	565
27.3. 结构框图	565
27.4. 功能描述	567
27.4.1. 负端输入	567
27.4.2. 滤波	567
27.4.3. 迟滞比较	567
27.4.4. 切断	567
27.4.5. STOP 模式唤醒	568
27.4.6. 窗口模式级联	568
27.5. 寄存器描述	569
27.5.1. COMP1 控制寄存器(COMP_CR1 偏移 00h)	569
27.5.2. COMP2 控制寄存器(COMP_CR2 偏移 04h)	571
27.5.3. COMP3 控制寄存器(COMP_CR3 偏移 08h)	574
27.5.4. COMP4 控制寄存器(COMP_CR4 偏移 0Ch)	576
27.5.5. COMP 状态寄存器(COMP_SR 偏移 10h)	578
27.6. 使用流程	580
28. 控制器区域网络（CAN）	581
28.1. 概述	581
28.2. 主要特性	581
28.3. 功能描述	582
28.3.1. 功能框图	582
28.3.2. 操作模式	583
28.3.3. 发送	583
28.3.4. 接收	586
28.3.5. 自我接收	589
28.3.6. 接收过滤	589

28.3.7. 波特率	591
28.3.8. 仲裁器	593
28.3.9. 错误处理	593
28.3.10. 睡眠模式	593
28.3.11. 测试模式	594
28.4. 寄存器描述	595
28.4.1. 模式寄存器 CAN_MOD(偏移: 00h)	596
28.4.2. 命令寄存器 CAN_CMR(偏移: 04h)	598
28.4.3. 状态寄存器 CAN_SR(偏移: 08h)	599
28.4.4. 中断寄存器 CAN_IR(偏移: 0Ch)	600
28.4.5. 中断使能寄存器 CAN_IER(偏移: 10h)	601
28.4.6. 时序寄存器 CAN_BTR(偏移: 14h)	602
28.4.7. 输出寄存器 CAN_OCR(偏移: 18h)	603
28.4.8. 错误代码获取寄存器 CAN_ECCR(偏移: 1Ch)	604
28.4.9. 错误计数寄存器 CAN_ERRCNT(偏移: 20h)	605
28.4.10. 接收报文状态寄存器 CAN_RSR(偏移: 0x24)	605
28.4.11. 接收过滤寄存器 CAN_ACR0(偏移: 0x28)	606
28.4.12. 接收过滤寄存器 CAN_ACR1(偏移: 0x2C)	606
28.4.13. 接收过滤寄存器 CAN_ACR2(偏移: 0x30)	606
28.4.14. 接收过滤寄存器 CAN_ACR3(偏移: 0x34)	606
28.4.15. 接收过滤寄存器 CAN_ACR4(偏移: 0x38)	606
28.4.16. 接收过滤寄存器 CAN_ACR5(偏移: 0x3C)	607
28.4.17. 接收过滤寄存器 CAN_ACR6(偏移: 0x40)	607
28.4.18. 接收屏蔽寄存器 CAN_AMR0(偏移: 0x44)	607
28.4.19. 接收屏蔽寄存器 CAN_AMR1(偏移: 0x48)	607
28.4.20. 接收屏蔽寄存器 CAN_AMR2(偏移: 0x4C)	607
28.4.21. 接收屏蔽寄存器 CAN_AMR3(偏移: 0x50)	607
28.4.22. 接收屏蔽寄存器 CAN_AMR4(偏移: 0x54)	608
28.4.23. 接收屏蔽寄存器 CAN_AMR5(偏移: 0x58)	608
28.4.24. 接收屏蔽寄存器 CAN_AMR6(偏移: 0x5C)	608
28.4.25. 发送缓存写寄存器 CAN_TXBUFFx(写偏移: 0x60~0x90)	608
28.4.26. 接收缓存读寄存器 CAN_RXBUFFx(读偏移: 0x60~0x90)	608
28.4.27. 接收 FIFO 访问寄存器 CAN_RXFIFO(偏移: 0xA0~0x19C)	609
28.4.28. 发送 FIFO 访问寄存器 CAN_TXFIFO(偏移: 0x1A0~0x1D0)	609
28.5. 使用流程	609
28.5.1. CAN 发送和接收	609
29. 数模转换器 (DAC)	611
29.1. 概述	611
29.2. 主要特性	611
29.3. 结构框图	612
29.4. 功能描述	612
29.4.1. DAC 通道使能	612
29.4.2. DAC 数据结构	613
29.4.3. DAC 转换和输出电压	615

29.4.4. DAC 触发选择	615
29.4.5. DAC 噪声叠加	616
29.4.6. DAC 锯齿波生成	619
29.4.7. DMA 请求	620
29.4.8. DAC 并发转换	621
29.4.9. 采样保持模式	622
29.5. 寄存器描述	624
29.5.1. DAC 控制寄存器(DAC_CR 偏移 00h)	625
29.5.2. DAC 软件触发寄存器(DAC_SWTRIGR 偏移 04h)	628
29.5.3. DAC 通道 1 12 位右对齐数据保持寄存器(DAC_DHR12R1 偏移 08h)	629
29.5.4. DAC 通道 1 12 位左对齐数据保持寄存器(DAC_DHR12L1 偏移 0Ch)	630
29.5.5. DAC 通道 1 8 位右对齐数据保持寄存器(DAC_DHR8R1 偏移 10h)	630
29.5.6. DAC 通道 2 12 位右对齐数据保持寄存器(DAC_DHR12R2 偏移 14h)	630
29.5.7. DAC 通道 2 12 位左对齐数据保持寄存器(DAC_DHR12L2 偏移 18h)	631
29.5.8. DAC 通道 2 8 位右对齐数据保持寄存器(DAC_DHR8R2 偏移 1Ch)	631
29.5.9. 双 DAC 12 位右对齐数据保持寄存器(DAC_DHR12RD 偏移 20h)	631
29.5.10. 双 DAC 12 位左对齐数据保持寄存器(DAC_DHR12LD 偏移 24h)	632
29.5.11. 双 DAC 8 位右对齐数据保持寄存器(DAC_DHR8RD 偏移 28h)	632
29.5.12. DAC 通道 1 数据输出寄存器(DAC_DOR1 偏移 2Ch)	632
29.5.13. DAC 通道 2 数据输出寄存器(DAC_DOR2 偏移 30h)	633
29.5.14. DAC 状态寄存器(DAC_SR 偏移 34h)	633
29.5.15. DAC 校准控制寄存器(DAC_CCR 偏移 38h)	635
29.5.16. DAC 模式控制寄存器(DAC_MCR 偏移 3Ch)	635
29.5.17. DAC 通道 1 采样时间寄存器(DAC_SHSR1 偏移 40h)	636
29.5.18. DAC 通道 2 采样时间寄存器(DAC_SHSR2 偏移 44h)	636
29.5.19. DAC 保持时间寄存器(DAC_SHHR 偏移 48h)	636
29.5.20. DAC 刷新时间寄存器寄存器(DAC_SHRR 偏移 4Ch)	637
29.5.21. DAC 通道 1 锯齿波寄存器(DAC_STR1 偏移 58h)	637
29.5.22. DAC 通道 2 锯齿波寄存器(DAC_STR2 偏移 5Ch)	637
29.5.23. DAC 锯齿波模式寄存器(DAC_STMODR 偏移 60h)	638
29.6. 使用流程	639
29.6.1. 单个 DAC 操作流程 (两个 DAC 独立工作)	639
29.6.2. 双 DAC 同步模式流程	640
30. 通用串行总线 (USB)	641
30.1. 概述	641
30.2. 主要特性	641
30.3. 功能描述	642
30.3.1. 中断状态和控制寄存器	642
30.3.2. 地址设置 Set Address	642
30.3.3. 远程唤醒	642
30.3.4. 令牌包与数据包 CRC 出错选择性回复 NAK	642
30.3.5. 数据包长度超过 64 Byte	643
30.3.6. 包丢失 EOP	643
30.3.7. IN 操作 ACK 超时下次 IN 操作回复 NAK	643

30.3.8. FIFO 访问与 Memory 访问	644
30.4. 寄存器描述	644
30.4.1. 工作模式寄存器 WORKING_MODE (偏移: 0x0h)	645
30.4.2. Ep0 传输控制寄存器 EP0CSR (偏移: 0x4h)	648
30.4.3. Ep1 传输控制寄存器 EP1CSR (偏移: 0x8h)	651
30.4.4. Ep2 传输控制寄存器 EP2CSR (偏移: 0xch)	653
30.4.5. Ep3 传输控制寄存器 EP3CSR (偏移: 0x10h)	656
30.4.6. Ep4 传输控制寄存器 EP4CSR (偏移: 0x14h)	659
30.4.7. Usb 地址寄存器 USB_ADDR (偏移: 0x18h)	661
30.4.8. SETUP 数据包寄存器 SETUP_0_3_DATA (偏移: 0x1ch)	662
30.4.9. SETUP 数据包寄存器 SETUP_4_7_DATA (偏移: 0x20h)	662
30.4.10. End Point 地址配置寄存器 EP_ADDR (偏移: 0x24h)	662
30.4.11. 总线包 PID 寄存器 CURRENT_PID (偏移: 0x28h)	662
30.4.12. Frame Number 寄存器 CURRENT_FRAME_NUMBER (偏移: 0x2ch)	663
30.4.13. CRC 错误 Counter 寄存器 CRC_ERROR_CNT (偏移: 0x30h)	663
30.4.14. 探测时间寄存器 USB_STATUS_DETECT_CNT (偏移: 0x34h)	663
30.4.15. EP0 发送数据数目寄存器 EPOSENDBN (偏移: 0x40h)	663
30.4.16. EP1 发送数据数目寄存器 EP1SENDBN (偏移: 0x44h)	663
30.4.17. EP2 发送数据数目寄存器 EP2SENDBN (偏移: 0x48h)	664
30.4.18. EP3 发送数据数目寄存器 EP3SENDBN (偏移: 0x4ch)	664
30.4.19. EP4 发送数据数目寄存器 EP4SENDBN (偏移: 0x50h)	664
30.4.20. Ep0 FIFO 访问入口 EP0FIFO (偏移: 0x100h)	664
30.4.21. Ep1 FIFO 访问入口 EP1FIFO (偏移: 0x104h)	664
30.4.22. Ep2 FIFO 访问入口 EP2FIFO (偏移: 0x108h)	665
30.4.23. Ep3 FIFO 访问入口 EP3FIFO (偏移: 0x10ch)	665
30.4.24. Ep4 FIFO 访问入口 EP4FIFO (偏移: 0x110h)	665
30.4.25. 状态寄存器 INT_STAT_RAW (偏移: 0xffe4h)	665
30.4.26. 中断使能寄存器 INT_EN (偏移: 0xffe8h)	667
30.4.27. 中断清除寄存器 INT_CLR (偏移: 0xffffh)	669
30.5. 软件流程	673
30.5.1. USB 连接	673
30.5.2. SETUP 数据和 EP0 控制传输数据	674
30.5.3. Endpoint In 传输	674
30.5.4. Endpoint Out 传输	675
31. 外部存储器控制器 (EXMC)	676
31.1. 概述	676
31.2. 主要特性	676
31.3. 结构框图	677
31.4. 功能描述	678
31.4.1. EXMC 访问基本规范	678
31.4.2. NOR/PSRAM 控制器	679
31.5. 寄存器描述	680
31.5.1. SRAM/ NOR Flash 控制寄存器 (EXMC_SNCTLx) (x 为 0~3) (偏移: 00/08/10/18h)	681
31.5.2. SRAM/ NOR Flash 时序配置寄存器 (EXMC_SNTCFGx) (x 为 0~3) (偏移:	

04/0C/14/1Ch).....	683
31.5.3. SRAM/ NOR Flash 写时序寄存器 (EXMC_SNWTCFGx) (x 为 0~3) (偏移: 104/10C/114/11Ch).....	685
31.6. 使用流程.....	686
32. CRC 计算单元.....	687
32.1. 概述.....	687
32.2. 主要特性.....	687
32.3. 寄存器描述.....	687
32.3.1. 数据寄存器/ CRC_DATA (偏移: 00h).....	687
32.3.2. 控制寄存器/ CRC_CTRL (偏移: 04h).....	687
32.3.3. 初始值寄存器/ CRC_INIT (偏移: 08h).....	688
32.3.4. 结果异或值寄存器/ CRC_OUTXOR (偏移: 10h).....	689
32.3.5. 多项式寄存器/ CRC_POLY (偏移: 14h).....	689
32.3.6. 独立数据寄存器/CRC_FDATA (偏移: 18h).....	689
32.4. 使用流程.....	689
33. 算法库.....	690
33.1. 数据类型.....	690
33.2. HRNG.....	690
33.2.1. 主要特性.....	690
33.2.2. 库文件说明.....	690
33.2.3. 函数说明.....	690
33.3. AES.....	691
33.3.1. 主要特性.....	691
33.3.2. 库文件说明.....	691
33.3.3. 函数说明.....	691
33.3.4. 注意事项.....	692
33.4. HASH.....	693
33.4.1. 主要特性.....	693
33.4.2. 库文件说明.....	693
33.4.3. 函数说明.....	693
33.4.4. 注意事项.....	694
33.5. FAU.....	694
33.5.1. 主要特性.....	694
33.5.2. 库文件说明.....	694
33.5.3. 函数说明.....	694
33.5.4. 注意事项.....	696

图目录

图 2- 1	处理器结构框图	29
图 2- 2	处理器寄存器组	29
图 2- 3	芯片系统架构图	30
图 2- 4	处理器 Memory Map	32
图 2- 5	芯片启动模式选择	36
图 3- 1	电源管理结构框图	42
图 4- 1	复位源关系图	60
图 4- 2	时钟结构图	62
图 7- 1	外部中断及事件框图	100
图 8- 1	eFlash 地址映射	111
图 8- 2	Program 操作流程	117
图 8- 3	Erase 操作流程	118
图 9- 1	结构框图	120
图 9- 2	外设到存储器	122
图 9- 3	存储器到外设	122
图 9- 4	存储器到存储器	123
图 9- 5	DMA 链表	125
图 10- 1	高级定时器结构框	136
图 10- 2	计数单元的结构	137
图 10- 3	当预分频器的参数从 1 变到 2 时，计数器的时序图	139
图 10- 4	当预分频器的参数从 1 变到 4 时，计数器的时序图	139
图 10- 5	外部时钟模式 1	140
图 10- 6	外部时钟模式 2	140
图 10- 7	通道 1 输入捕获 TI1 信号	141
图 10- 8	PWM 输入模式时序	143
图 10- 9	输出比较模式，翻转 OC1	145
图 10- 10	边沿对齐的 PWM 波形 (ARR=8)	146
图 10- 11	中央对齐的 PWM 波形 (ARR=8)	148
图 10- 12	带死区插入的互补输出	150
图 10- 13	死区波形延迟大于负脉冲	151
图 10- 14	死区波形延迟大于正脉冲	151
图 10- 15	响应刹车的输出	154
图 10- 16	刹车控制	154
图 10- 17	产生六步 PWM，使用 COM 的例子(OSSR=1)	155
图 10- 18	编码器模式下计数器操作实例	157
图 10- 19	霍尔传感器接口的实例	158
图 10- 20	单脉冲模式的例子	159
图 10- 21	复位模式下的控制电路	161
图 10- 22	门控模式下的控制电路	162
图 10- 23	触发模式下的控制电路	163
图 10- 24	外部时钟模式 2+触发模式控制电路	163
图 10- 25	ocref_clr 输入源选择	165

图 11- 1 通用定时器 TIM2 结构框图	213
图 12- 1 通用定时器 TIM3/TIM4 框图	251
图 13- 1 基本定时器结构框图	286
图 14- 1 通用定时器 TIM15 结构框图	294
图 14- 2 通用定时器 TIM16/TIM17 结构框图	294
图 15- 1 LPTIM 结构框图	330
图 15- 2 干扰滤波器时序图	332
图 15- 3 LPTIM 输出波形, 单触发模式	334
图 15- 4 LPTIM 输出波形, 单触发模式, 当重复寄存器不为 0 (PRELOAD=1)	335
图 15- 5 LPTIM 输出波形, 单触发模式, 一次激活模式 (WAVE=1)	335
图 15- 6 LPTIM 输出波形, 重复触发模式	336
图 15- 7 生成波形	338
图 15- 8 编码器模式计数序列	342
图 15- 9 重复计数模式 LPTIM_RCR 不为 0 (PRELOAD=1)	342
图 16- 1 WDT 结构图	361
图 17- 1 IWDG 结构框图	367
图 18- 1 RTC 内部结构框图	374
图 19- 1 GPIO 管脚结构框图	394
图 20- 1 SPI 结构框图	404
图 20- 2 SPI 一线模式时序图 (MSB)	405
图 20- 3 SPI 一线模式时序图 (LSB)	405
图 20- 4 SPI 二线模式时序图 (MSB)	406
图 20- 5 SPI 二线模式时序图 (LSB)	407
图 20- 6 SPI 四线模式时序图 (MSB)	407
图 20- 7 SPI 四线模式时序图 (LSB)	408
图 21- 1 UART 结构框图	429
图 21- 2 两个 UART 之间的硬件流控连接示意图	429
图 21- 3 LIN 帧格式	430
图 21- 4 LIN 间隔场格式	431
图 21- 5 LIN 字符格式	431
图 21- 6 IrDA SIR 数据调制解调	432
图 21- 7 UART 单线半双工模式框图	432
图 22- 1 I2C 结构框图	455
图 22- 2 I2C 总线传输	456
图 22- 3 7 位从发送器的传送图	457
图 22- 4 7 位从接收器的传送图	457
图 22- 5 从机发送模式时钟延长时序 (虚线部分 SCL 为低, 主机无法发送 SCL)	458
图 22- 6 7 位主发送器的传送图	458
图 22- 7 7 位主接收器的传送图	459
图 22- 8 滤波算法示意图	459
图 22- 9 SDA 跳变检测图	460
图 22- 10 TXE 功能说明图	460
图 22- 11 主机发送流程图 (虚线步骤可跳过, 下同)	470
图 22- 12 主机接收流程图	472

图 22- 13	从机发送接收流程图	474
图 23- 1	结构框图	477
图 23- 2	飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)	483
图 23- 3	飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)	483
图 23- 4	飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)	484
图 23- 5	飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)	484
图 23- 6	飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	484
图 23- 7	飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	484
图 23- 8	飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	484
图 23- 9	飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	485
图 23- 10	MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)	485
图 23- 11	MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)	485
图 23- 12	MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)	485
图 23- 13	MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)	485
图 23- 14	MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	486
图 23- 15	MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	486
图 23- 16	MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	486
图 23- 17	MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	486
图 23- 18	对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	487
图 23- 19	LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	487
图 23- 20	LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	487
图 23- 21	LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	487
图 23- 22	PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)	488
图 23- 23	PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)	488
图 23- 24	PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)	488
图 23- 25	PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)	488
图 23- 26	PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)	488
图 23- 27	PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)	488
图 23- 28	PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)	489
图 23- 29	PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)	489
图 23- 30	PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)	489
图 23- 31	PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)	489
图 23- 32	PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)	489
图 23- 33	PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)	489
图 23- 34	PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)	490
图 23- 35	PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)	490
图 23- 36	PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)	490
图 23- 37	PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)	490
图 24- 1	LPUART 结构框图	495
图 24- 2	LPUART BIT 分布图	496
图 25- 1	ADC 框图	507
图 25- 2	ADC 转换时序图	512
图 25- 3	AWD 阈值图	513
图 25- 4	数据右对齐	514

图 25- 5	数据左对齐	515
图 25- 6	4 个通道的注入同步模式	519
图 25- 7	16 个通道的规则同步模式	519
图 25- 8	连续转换模式下 1 通道的规则交叉模式	520
图 25- 9	交替触发：各个 ADC 的注入组	521
图 25- 10	交替触发：间断模式下的各个 ADC 的注入组	522
图 25- 11	规则同步+交替触发	523
图 25- 12	规则同步+交替触发：在注入转换期间出现新的注入触发事件	523
图 25- 13	混合的注入同步+规则交叉模式	523
图 26- 1	运放外置模式配置等效图	555
图 26- 2	运放单位增益模式配置等效图	555
图 26- 3	运放 PGA 同相模式配置等效图	556
图 26- 4	运放 PGA 反相模式配置等效图	556
图 27- 1	COMP 结构框图	566
图 27- 2	比较器窗口模式级联	568
图 28- 1	内部功能框图	582
图 28- 2	正常模式示意图	583
图 28- 3	CAN 数据帧结构	584
图 28- 4	CAN 接收示意图	587
图 28- 5	单过滤模式过滤器示意图	590
图 28- 6	双过滤模式过滤器示意图	591
图 28- 7	CAN 位时序图	592
图 28- 8	静默模式示意图	594
图 28- 9	环回模式示意图	594
图 28- 10	环回静默模式示意图	594
图 29- 1	DAC 结构框图	612
图 29- 2	DAC 通道模式的数据寄存器	613
图 29- 3	DAC 单通道模式的数据寄存器	614
图 29- 4	LFSR 噪声功能框图	617
图 29- 5	带 LFSR 噪声模的 DAC 转换波形	617
图 29- 6	三角波噪声功能框图	618
图 29- 7	带三角波噪声的 DAC 转换波形	618
图 29- 8	递减锯齿波 (STDIR=0)	619
图 29- 9	递增锯齿波 (STDIR=1)	620
图 29- 10	采样保持模式时序图	623
图 30- 1	USB 结构框架图	641
图 30- 2	控制传输过程	674
图 30- 3	Endpoint In 传输	675
图 30- 4	Endpoint Out 传输	675
图 31- 1	EXMC 结构框图	678
图 31- 2	EXMC Bank 划分	679

表目录

表 2- 1	ACM32G103 存储空间映射表	32
表 3- 1	数字模块供电状态表	43
表 3- 2	低功耗模式进入和退出	45
表 3- 3	各个模块在不同功耗模式下工作状态	47
表 4- 1	复位源	59
表 4- 2	系统时钟源	63
表 5- 1	NVIC 中断源表	93
表 6- 1	定时器 ITR 互连关系表	96
表 6- 2	定时器 ETR 互连关系表	97
表 6- 3	定时器 ocref_clr 互连关系表	97
表 6- 4	定时器输入 TI1 互连关系表	98
表 6- 5	定时器输入 TI2 互连关系表	98
表 6- 6	定时器输入 TI3 互连关系表	98
表 6- 7	定时器输入 TI4 互连关系表	98
表 6- 8	定时器刹车输入源互连关系	99
表 7- 1	EXTI 源	101
表 9- 1	目标外设和源外设请求号	120
表 10- 1	计数器方向和编码器信号的关系	156
表 10- 2	互补输出通道 OCx 和 OCxN 的控制	194
表 11- 1	互补输出通道 OCx 和 OCxN 的控制	239
表 15- 1	LPTIM1 输入 1 连接	330
表 15- 2	LPTIM1 输入 2 连接	330
表 15- 3	外部触发连接	331
表 15- 4	预分频系数	333
表 15- 5	编码器计数方案	341
表 15- 6	LPTIM 低功耗模式	343
表 15- 7	中断事件	344
表 23- 1	I2S 比特率计算公式	490
表 23- 2	Fs 采样频率计算公式	491
表 25- 1	ADC 模拟通道选择	508
表 25- 2	AWD 控制位表	513
表 25- 3	ADC 规则/注入转换触发源选择表	516
表 25- 4	中断使能和标志	526
表 28- 1	CAN_TXBUFFx 数据结构	584
表 28- 2	CAN_RXBUFFx 数据结构	587
表 28- 3	SegCode 错误码表	604
表 29- 1	Data format (12 位模式)	614
表 29- 2	DAC 触发源选择表	615
表 29- 3	递增触发选择	616
表 33- 1	HRNG 驱动函数说明	690
表 33- 2	库文件说明	691
表 33- 3	AES 驱动函数说明	691

表 33- 4	HASH 驱动函数说明	693
表 33- 5	FAU 驱动函数说明	694

ALISINOCHIP

1. 文档约定

1.1. 基本信息

ACM32G103 芯片是基于 ARMv8-M 架构内核的通用处理器芯片。芯片具体信息以数据手册为准。

1.2. 寄存器属性缩写表

read/write (RW): 可读写

read-only (RO): 只读

write-only (WO): 只写

read/clear write0 (RC_W0): 只读, 写 0 清 0, 写 1 无效

read/clear write1 (RC_W1): 只读, 写 1 清 0, 写 0 无效

read/clear by read (RC_R): 只读, 读该位后硬件自动清零该位, 写无效

RSV: 保留

1.3. 术语

AHB: advanced high-performance bus.

APB: advanced peripheral bus.

Word: data of 32-bit length.

Half-word: data of 16-bit length.

Byte: data of 8-bit length.

NVR: Non-Volatile Region

eFlash: embedded Flash

2. 存储器及系统架构

ACM32G103 芯片的内核基于 ARMv8-M 架构，支持 Cortex-M33 和 Cortex-M4F 指令集，系统最高频率达 120 MHz。内核支持一整套 DSP 指令用于数字信号处理，支持单精度 FPU 处理浮点数据，同时还支持 Memory Protection Unit（MPU）用于提升应用的安全性。

2.1. 内核处理器

内核处理器基于 ARMv8-M 架构。处理器包括两个总线接口分别称为 C-AHB 总线、S-AHB 总线：

C-AHB 总线：用于访问 ARMv8-M 存储架构下代码区的指令或数据。

S-AHB 总线：用于访问 ARMv8-M 存储架构下 SRAM 区、外部 RAM 区、外设区或厂商自定义系统区的指令或数据。

处理器功能强大，系统频率最高可达 120MHz，支持硬件加速，支持单精度浮点处理单元，支持数字信号处理指令，在安全方面，内核也集成了存储保护单元 MPU。

处理器结构框图如图 2-1 所示，寄存器组见图 2-2 所示。

图 2-1 处理器结构框图

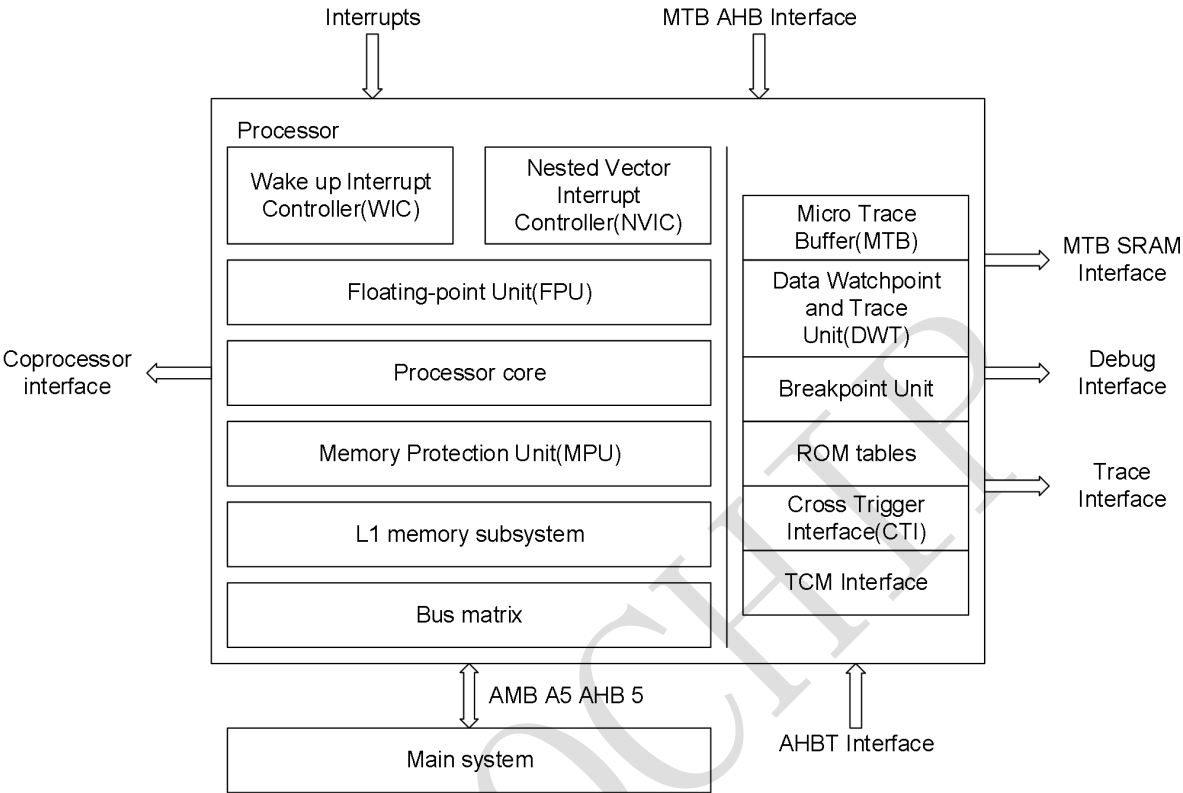
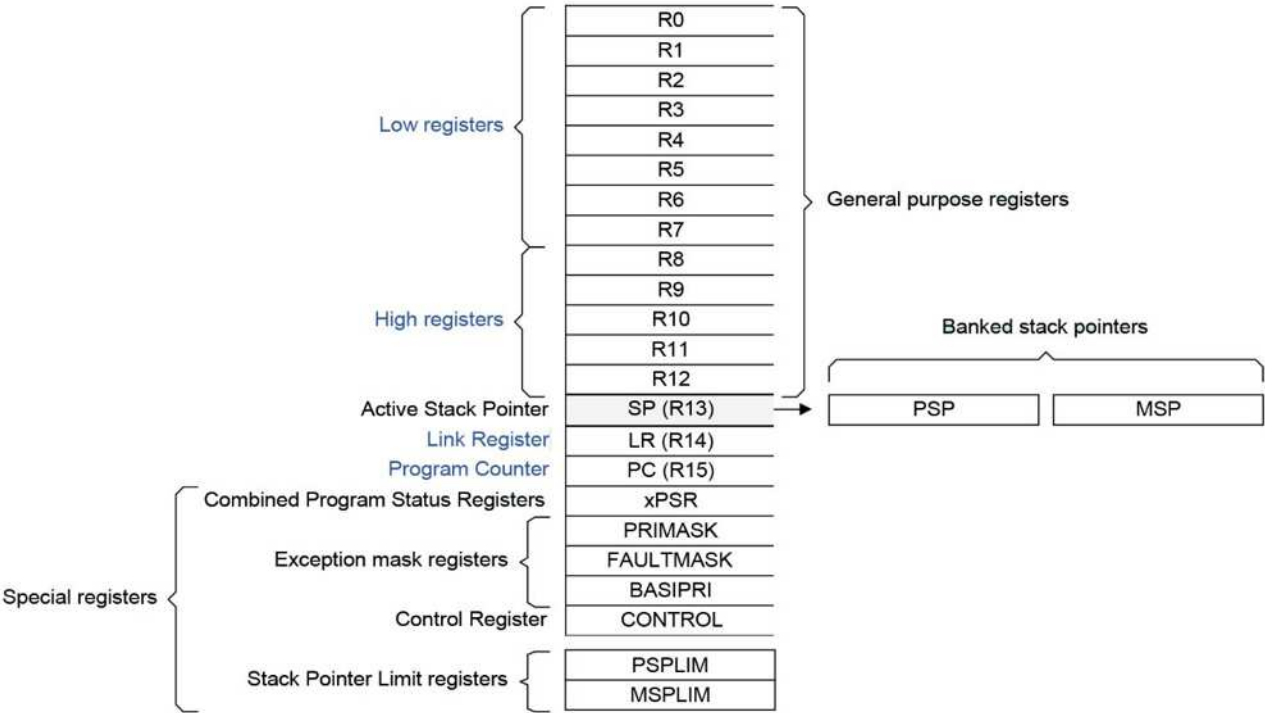


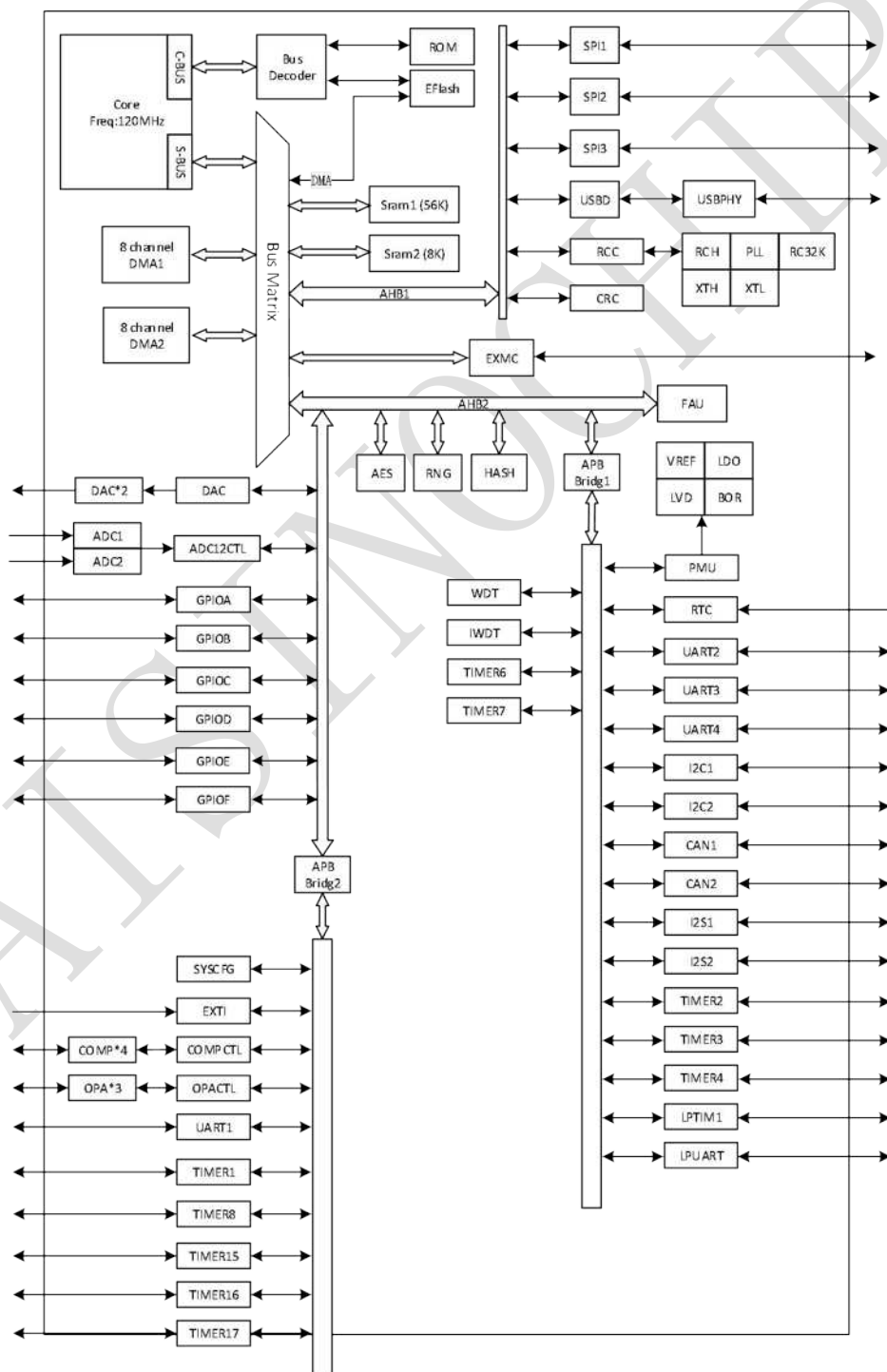
图 2-2 处理器寄存器组



2.2. 系统架构

系统架构图如图 2-3 所示。AHB1 和 AHB2 时钟频率与系统频率一致，最高可达 120MHz。APB1 和 APB2 的时钟可分别通过系统时钟分频而来（当不分频时，APB1 和 APB2 的时钟频率最高也可达 120MHz）。

图 2-3 芯片系统架构图



2.3. 存储器映射

内核处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和 I/O 端口都在同一个 4 GB 的地址空间之内。存储器中字节数据以小端方式排列。

处理器地址映射如图 2-4 所示。

ACM32G103 芯片的存储空间分为 ROM 启动和 eFlash 启动两种模式，具体如下表所示。

ROM 上电时，ROM 的地址为 0x0000_0000 和 0x1200_0000 同时有效，0x1200_0000 为 shadow 区；eFlash 地址映射为 0x1000_0000。系统支持存储地址空间重映射(remapping)机制，即将 320KB Flash 空间映射到 0x0 起始地址，以屏蔽 ROM 启动，直接从 eFlash 启动，便于用户控制所有系统资源，此时 eFlash 的地址为 0x0000_0000；ROM 空间起始地址为 0x1200_0000。

图 2-4 处理器 Memory Map

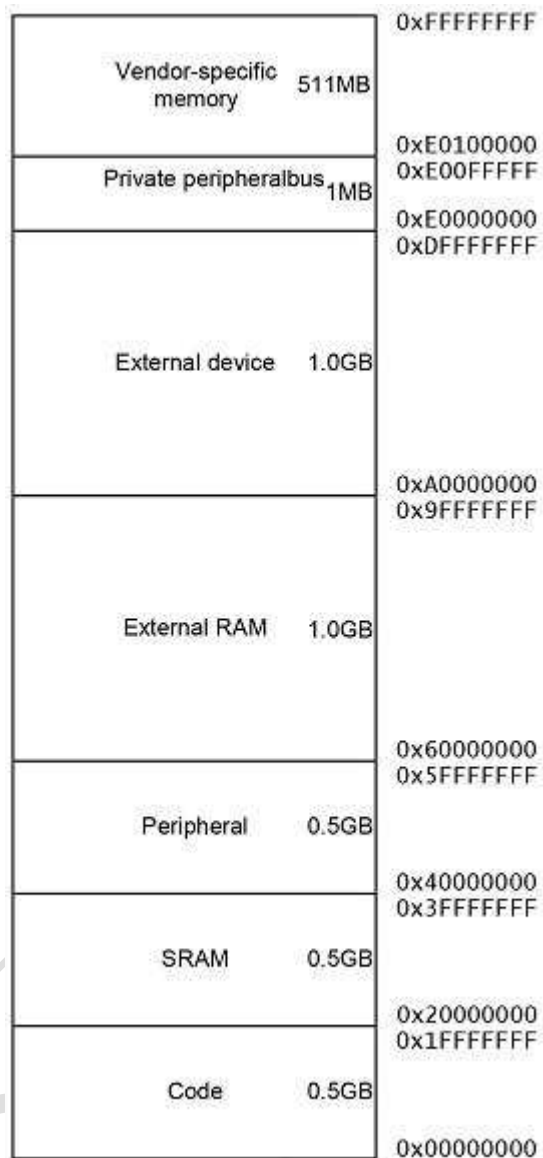


表 2-1 ACM32G103 存储空间映射表

模块名	ROM 启动	EFlash 启动
ROM	0x0000_0000——0x0000_2FFF 0x1200_0000——0x1200_2FFF	0x1200_0000——0x1200_2FFF
eFlash	0x1000_0000——0x1004_FFFF	0x0000_0000——0x0004_FFFF
eFlash NVR	0x1008_0000——0x1008_0FFF	0x0008_0000——0x0008_0FFF
SRAM		
SRAM	0x2000_0000——0x2000_FFFF	

APB1 Slave	
TIM2	0x4000_0000——0x4000_03FF
TIM3	0x4000_0400——0x4000_07FF
TIM4	0x4000_0800——0x4000_0BFF
TIM6	0x4000_1000——0x4000_13FF
TIM7	0x4000_1400——0x4000_17FF
RTC	0x4000_2800——0x4000_2BFF
WDT	0x4000_2C00——0x4000_2FFF
IWDT	0x4000_3000——0x4000_33FF
I2S1	0x4000_3800——0x4000_3BFF
I2S2	0x4000_3C00——0x4000_3FFF
UART2	0x4000_4400——0x4000_47FF
UART3	0x4000_4800——0x4000_4BFF
UART4	0x4000_4C00——0x4000_4FFF
I2C1	0x4000_5400——0x4000_57FF
I2C2	0x4000_5800——0x4000_5BFF
CAN1	0x4000_6400——0x4000_67FF
CAN2	0x4000_6800——0x4000_6BFF
PMU	0x4000_7000——0x4000_73FF
LPTIM1	0x4000_7C00——0x4000_7FFF
LPUART1	0x4000_8000——0x4000_83FF
APB2 Slave	
SYSCFG	0x4001_0000——0x4001_002F
COMP	0x4001_0200——0x4001_02FF
OPA	0x4001_0300——0x4001_03FF
EXTI	0x4001_0400——0x4001_07FF
TIM1	0x4001_2C00——0x4001_2FFF
TIM8	0x4001_3400——0x4001_37FF
UART1	0x4001_3800——0x4001_3BFF

TIM15	0x4001_4000——0x4001_43FF
TIM16	0x4001_4400——0x4001_47FF
TIM17	0x4001_4800——0x4001_4BFF
AHB1 Slave	
RCC	0x4002_1000——0x4002_13FF
EFC	0x4002_2000——0x4002_23FF
CRC	0x4002_3000——0x4002_33FF
SPI1	0x4003_0000——0x4003_03FF
SPI2	0x4003_0400——0x4003_07FF
SPI3	0x4003_0800——0x4003_0BFF
DMA1	0x4003_1000——0x4003_1FFF
DMA2	0x4003_2000——0x4003_2FFF
USB2.0	0x4005_0000——0x4005_FFFF
AHB2 Slave	
GPIOA	0x4800_0000——0x4800_03FF
GPIOB	0x4800_0400——0x4800_07FF
GPIOC	0x4800_0800——0x4800_0BFF
GIOD	0x4800_0C00——0x4800_0FFF
GPIOE	0x4800_1000——0x4800_13FF
GPIOF	0x4800_1400——0x4800_17FF
ADC1/ADC2	0x5000_0000——0x5000_03FF
DAC1	0x5000_0800——0x5000_0BFF
AES	0x5006_0000——0x5006_03FF
FAU	0x5006_0400——0x5006_07FF
HRNG	0x5006_0800——0x5006_0BFF
HASH	0x5006_0C00——0x5006_0FFF
QSPI memory	
SPI3	0x9000_0000——0x9FFF_FFFF
External memory	

EXMC_MEM	0x6000_0000——0x6FFF_FFFF
EXMC_REG	0xA000_0000——0xA000_03FF

2.4. 片上 SRAM

芯片集成了 64KB 的 SRAM，支持字节（8 位）、半字（16 位）以及字（32 位）访问。SRAM 最后的 8KB 支持在 STOP2 模式时 retention 功能，能保持数据。处理器可以以 120MHz 的系统频率无等待地访问 SRAM。SRAM 支持奇偶校验，当校验出错时，能产生错误标志。

2.5. 片上 Flash

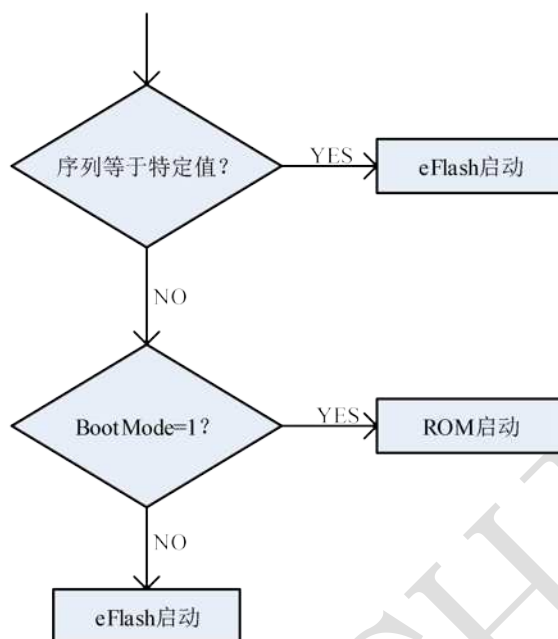
芯片集成了 320KB 的 Flash 以及 8 页的 NVR，支持字节（8 位）、半字（16 位）以及字（32 位）访问。NVR 区的配置字可用于芯片配置，重新上电后生效。片上 Flash 的具体信息请参考 EFC 章节。

2.6. BOOT 配置

芯片有两种启动模式：ROM 启动和 eFlash 启动。

系统上电时，芯片会读取安全序列字段和系统寄存器 WMR 的 BootMode 标志位，决定是将 eFlash 还是将 ROM 映射到 0x0 地址。BootMode 标志位由上电时 BOOT 引脚的高低电平决定。图 2-4 描述了芯片启动模式选择过程。

图 2-5 芯片启动模式选择



2.7. 芯片序列号

每颗芯片都包含唯一的 128 位（16 字节）的序列号。

基地址：0x0008_0208

2.8. 系统配置寄存器（SYSCFG）

系统配置寄存器是 APB2 从设备。寄存器映射如下表所示：

SYSCFG 寄存器基地址：0x4001_0000

地址	名称	描述
0x00	SYSCR	系统控制寄存器
0x04	WMR	工作模式寄存器
0x08	VER	版本寄存器
0x0C	保留	
0x10	PHYCFG	USBPHY 模块控制寄存器
0x14	Dummy	Dummy 寄存器，默认为 0，可读写

2.8.1. 系统控制寄存器(SYSR)（偏移：00h）

比特	名称	属性	默认值	功能描述
31:19	RSV	-	-	保留
18:17	IR_MODE	RW	00	IR_OUT 一端选择 TIM16_CH1 还是 UART1/2_TXD 00: IR_OUT= $\sim$ (TIM17_CH1& TIM16_CH1) 01: IR_OUT= $\sim$ (TIM17_CH1&UART1_TXD) 10: IR_OUT= $\sim$ (TIM17_CH1&UART2_TXD) 11: 保留
16	IR_POL	RW	0	IR_OUT 输出极性控制 0: 原极性 1: 取反
15:9	RSV	-	-	保留
8	SRAM_PEF	RC_W1	0	SRAM 校验错误标志，写 1 清除 0: 无 SRAM 校验错误 1: 发生 SRAM 校验错误
7:4	RSV	-	-	保留
3	Flash_PARITY_LOCK	RW	0	FLASH 奇偶校验错误锁定 0: FLASH 校验错误输出从 TIM1/8/15/16/17 的 break 输入端断开 1: FLASH 校验错误输出与 TIM1/8/15/16/17 的 break 输入端连接
2	LVD_LOCK	RW	0	LVD 检测错误锁定 0: LVD 检测错误输出从 TIM1/8/15/16/17 的 break 输入端断开 1: LVD 检测错误输出与 TIM1/8/15/16/17 的 break 输入端连接
1	SRAM_PARITY_LOCK	RW	0	SRAM 奇偶校验错误锁定

				0: SRAM 校验错误输出从 TIM1/8/15/16/17 的 break 输入端断开 1: SRAM 校验错误输出与 TIM1/8/15/16/17 的 break 输入端连接
0	LOCKUP_LOCK	RW	0	Core LOCKUP 输出锁定 0: LOCKUP 输出从 TIM1/8/15/16/17 的 break 输入端断开 1: LOCKUP 输出与 TIM1/8/15/16/17 的 break 输入端连接

2.8.2. 工作模式寄存器(WMR)（偏移：04h）

比特	名称	属性	默认值	功能描述
31:7	RSV	-	-	保留
6	RTC_Ready	RO	1	STANDBY 唤醒后，RTC 域是否准备完成 0: RTC 域未 ready，不能访问 1: RTC 域已 ready，可以访问
5	RSV	-	-	保留
4	REMAP_Flag	RO	X	REMAP 标志，表示现在是 ROM 工作模式还是 eFlash 工作模式： 0: ROM 工作模式 1: eFlash 工作模式 该寄存器的值会反映在 REMAP 管脚上。
3	BootMode	RO	X	BootMode 状态，用于指示芯片从 ROM 启动还是从 eFlash 启动，优先级低于 NVR Remapping 。 0:指示系统从 eFlash 启动； 1:指示系统从 ROM 启动，优先级低于 NVR remapping bit
2:0	RSV	-	-	保留

2.8.3. 版本寄存器(VER)（偏移：08h）

比特	名称	属性	默认值	功能描述
31: 0	Version	RO	0xFEDF0120	芯片版本 15~8: 0x01 代表 MCU 产品 7~0: 芯片版本, 0x20 表示 V1 版本 31~16 位为 15~0 位的取反值

2.8.4. USBPHY 配置寄存器(PHYCFG)（偏移：10h）

比特	名称	属性	默认值	功能描述
31:20	RSV	-	-	保留
19	CLKSEL_END	RO	X	USB PHY 外部时钟源查询标志 (OSC_MODE[1]=0 有效) 0: 查询中 1: 查询完成 查到此位为 1 后表示时钟查询完成(查询时间约 20ms), 可以查询 CLKSEL_LRC_HXO 来确定时钟源选择结果
18	ADJ_END	RO	X	RC 校准完成 (内部 RC 模式) 0: 校准未完成 1: 校准完成
17	PLLLOCK	RO	X	PLL 锁定标志 (外部晶振模式) 0: PLL 未锁定 1: PLL 锁定
16	CLKSEL_LRC_HXO	RO	X	时钟源选择标志 0: 选择内部 RC

				1: 选择外部晶振
15:8	RSV	-	-	保留
7	DUPD	RW	0	USBPHY DP 下拉电阻选择, 高有效
6	DMPD	RW	0	USBPHY DM 下拉电阻选择, 高有效
5	RSV	-	-	保留
4:3	OSC_MODE[1:0]	RW	00	PHY 内 RC 工作模式 00: 自动检测外部晶振 (复位释放后 20ms 检测) 01: 快速自动检测外部晶振 (1us) 10: 强制选择内部 RC 11: 强制选择外部晶振
2	PD_PLL	RW	1	PLL/RC Power Down 控制。从 PD 状态恢复, 时钟保持进入 PowerDown 前状态 0: 正常工作 1: PowerDown 模式
1	FPLL_H60M_L48M	RW	0	PHY 输出时钟 48M 和 60M 选择 0: 48M 1: 60M
0	PHY_RSTN	RW	1	USB_PHY 软复位, 写 0 复位, 写 1 撤销

3. 电源管理及低功耗（PMU）

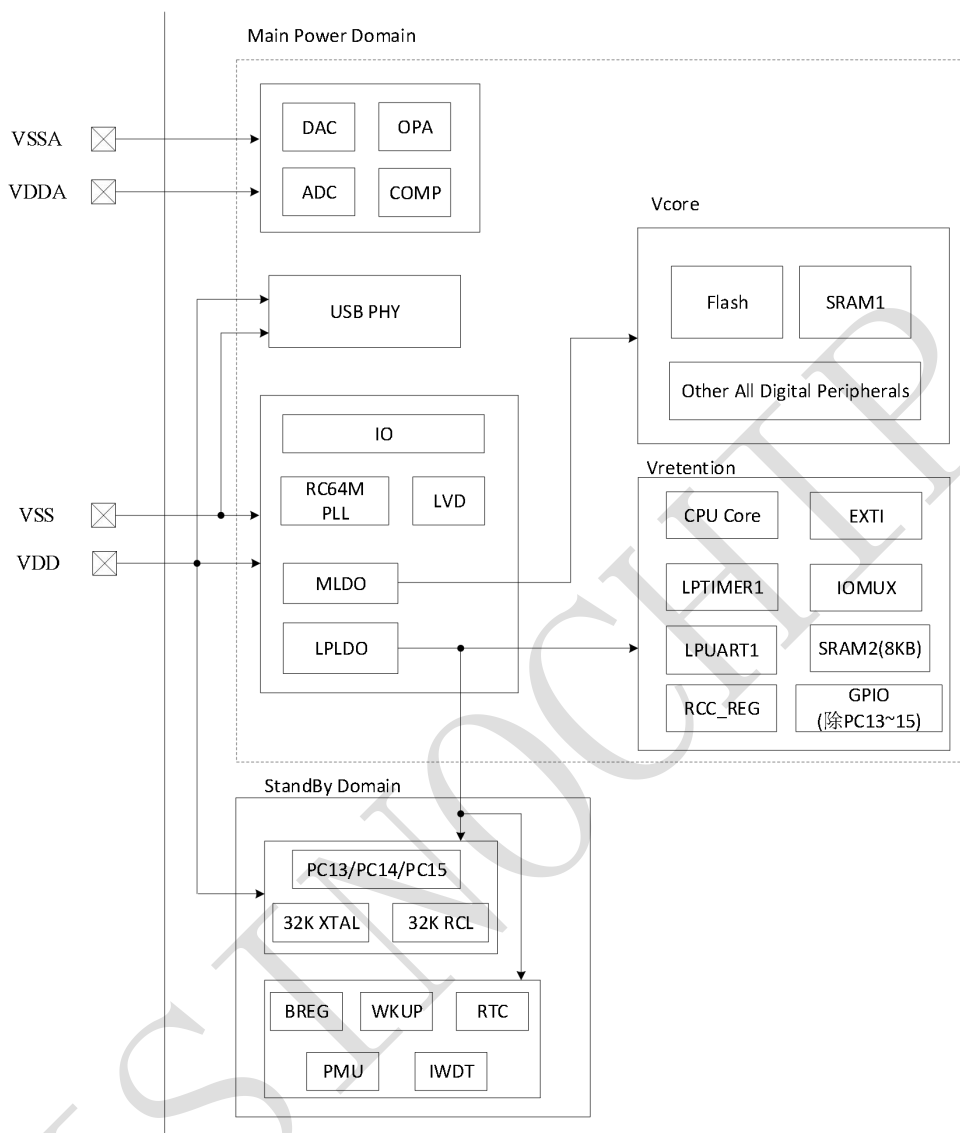
3.1. 供电电源

ACM32G103 系列产品的工作电压（VDD）1.70~3.60V。如下图所示，整个电源分为 2 个区域：主区和待机区。

主区为芯片正常运行（RUN 模式）的主供电区，通过 VDDA/VDD 分别提供模拟电源和数字电源。主电源区内嵌 2 个电压调节器（主调节器 MLDO 和低功耗调节器 LPLDO），用来为 1.2V 电压域供电。

待机区是芯片进入低功耗待机（STANDBY 模式）的供电区域，由低功耗调节器 LPLDO 供电。

图 3-1 电源管理结构框图



3.1.1. 主电源区

VDD 电源域

VDD 为数字区域电源，电压范围为 1.7V~3.6V，主要给 IO、内嵌电压调节器 LDO、BOR、LVD、USB PHY、PLL、内部 RC 时钟和 Flash 供电；其中 Flash 电源由 LDO17 提供。

VDDA 电源域

VDDA 为模拟区域电源，电压范围为 1.7V~3.6V，主要给大部分的模拟外设供电，主要包括 CMP、OPA、ADC、DAC 等模拟模块。

1.2V 电源域

芯片集成两个 1.2V 的 LDO：MLDO12 和 LPLDO12，在主电源区分别给 Vcore 和 Vretention 数字部分供电。

LPLDO12 和 MLDO12 的输出电压范围可分别通过寄存器来调节。

3.1.2. 待机电源区

待机区域中，32K XTAL、32K OSC 和 PC13~15 由 VDD 直接供电，待机区的数字部分由 LPLDO 供电。待机区的数字部分主要包括 IWDT、RTC、BREG（备份寄存器）、PMU 和唤醒控制模块。

3.1.3. 数字模块供电状态

数字模块由 MLDO 和 LPLDO 供电，在不同运行模式下，MLDO 和 LPLDO 会有不同开关状态；数字模块在不同运行模式下的供电状态如下表所示

表 3-1 数字模块供电状态表

供电区域	运行模式						
	RUN	Sleep	STOP0	STOP1	STOP2	StandBy	PowerOff
VcoreDomain	MLDO	MLDO	MLDO	LPLDO	OFF	OFF	OFF
VretentionDomain	MLDO	MLDO	LPLDO	LPLDO	LPLDO	OFF	OFF
StandBy Domain	MLDO	MLDO	LPLDO	LPLDO	LPLDO	LPLDO	OFF

注：STOP0 模式下，MLDO 可以调节电压输出

3.2. 电源监控

3.2.1. POR 和 PDR

芯片集成了 POR（上电复位）/PDR（掉电复位）模块，用于监控 VDD 的电压。POR 电压为 1.5V，在上电时当 VDD 由 0 变高未超过 1.5V 时，POR 保持复位状态，不需要外接复位电路，超过阈值后复位释放。PDR 电压为 1.45V，当 VDD 下降到低于此阈值时，PDR 产生复位。

POR/PDR 始终有效，不能由软件控制。

3.2.2. BOR 复位

BOR 用于检测 VDD 供电电压是否低于阈值，在 STANDBY 模式也能工作。上电默认不工作，通过配置 PMU_CTRL2 寄存器相关位使能，有四挡电压选择（2.0~2.8）。

3.2.3. LVD 模块

LVD 用于检测 VDD 供电电压是否低于低电压检测阈值，在 STANDBY 模式无效。

LVD 模块可以由软件使能和选择阈值（1.7~2.9）。LVD 模块的结果可以从 PMU_CTRL1 寄存器直接读出。

为了能够 LVD 事件从 STOP1 和 STOP2 模式唤醒，LVD 也连接到了外部中断源的 EXTI，通过外部中断也能唤醒系统和产生中断。

3.3. 低功耗模式

在芯片上电复位或者系统复位后，芯片处于运行模式（RUN）。芯片可以通过降低系统时钟频率，或者关闭不需要使用的模块的时钟来降低功耗。此外芯片还提供了多种不同的低功耗模式来实现更低的功耗。用户可根据最低功耗需求、最快唤醒时间、唤醒方式以及待机时的工作需求等选择合适的低功耗模式。主要包括如下低功耗模式：

- 睡眠模式（SLEEP）：内核停止工作，外设保持工作

- 停止模式 0 (STOP0)：PLL、RC64M、XTH 时钟关闭；RC32K/XTL 时钟下的外设可以工作；MLDO12 的输出电压范围可通过寄存器来调节
- 停止模式 1 (STOP1)：PLL、RC64M、XTH 时钟关闭；RC32K/XTL 时钟下的外设可以工作；Vcore 区域由 LPLDO 供电，MLDO 关闭
- 停止模式 2 (STOP2)：Vcore 区域断电，其中 CPU 和部分外设寄存器保持掉电前的状态，后 8KB 的 SRAM 可以保持数据
- 待机模式 (STANDBY)：主区断电，RTC 和 IWDG 待机区工作
- 断电模式 (POWERDOWN)：主区和待机区断电

注：在 STOP0/STOP1 模式下，PLL、XTH 时钟由软件控制开启或关闭；通过配置寄存器 PMU_CTRL0 的 bit4，控制 RC64M 时钟在进入 STOP0/STOP1 模式后，是否自动关闭。

3.3.1. 低功耗模式描述

低功耗模式详细描述如下表：

表 3-2 低功耗模式进入和退出

模式	模式描述	进入条件	退出条件
SLEEP	CPU 休眠；功能模块不休眠；软件可关闭各模块时钟。	1. 根据需要，关闭各外设模块时钟，仅留下需要监测中断事件的模块 2. 清除 Core 的 SLEEPDEEP 位，并执行 WFI/WFE 指令	1. 任何来自 EXTI/外设的中断或事件； 2. 如果是 WFI 唤醒，则进入中断服务程序清中断并返回； 3. 唤醒后，可继续执行后续指令
STOP0	PLL、RC64M、XTH 时钟关闭；RC32K/XTL 时钟下的外设可以工作；MLDO12 的输出电压范围可通过寄存器来调节。	1. 关闭全局中断和 SysTick 中断，并清除相应的中断标志； 2. 设置 PMU_CTRL0 的 LPMS 位选择 STOP0 模式； 3. 设置 Core 的 SLEEPDEEP 位并执行 WFI 或者 WFE。	1. 任何来自 EXTI/外设的中断或事件；RSTN 管脚复位/BOR 复位/IWDG 复位 2. 唤醒后，可继续执行后续指令 3. RSTN 管脚复位/BOR 复位/IWDG 复位可退出 STOP0，再进入 STOP0 前确保相应标志位

			被清除
STOP1	在 STOP0 的基础上，关闭 MLDO，Vcore 区域由 LPLDO 供电。	1. 关闭全局中断和 SysTick 中断，并清除相应的中断标志； 2. 设置系统时钟为 RC64M； 3. 设置 PMU_CTRL0 的 LPMS 位选择 STOP1 模式； 4. 设置 Core 的 SLEEPDEEP 位并执行 WFI 或者 WFE。	1. 任何来自 EXTI/外设的中断或事件；RSTN 管脚复位/BOR 复位/IWDT 复位 2. 唤醒后，可继续执行后续指令 3. RSTN 管脚复位/BOR 复位 /IWDT 复位可退出 STOP1，再进入 STOP1 前确保相应标志位被清除
STOP2	Vcore 区域断电，其中 CPU 寄存器保持掉电前的状态	1. 关闭全局中断和 SysTick 中断，并清除相应的中断标志； 2. 设置系统时钟为 RC32K； 3. 设置 PMU_CTRL0 的 LPMS 位选择 STOP2 模式； 4. 设置 Core 的 SLEEPDEEP 位并执行 WFI 或者 WFE。	1. 除 LVD、USB wakeup、GPIOA11、GPIOA12、COMP 的其他任何来自 EXTI 的中断或者事件 2. 唤醒后，可继续执行后续指令 3. RSTN 管脚复位/BOR 复位 /IWDT 复位可退出 STOP2，唤醒后，可继续执行后续指令
STANDBY	关闭主区电源，待机区工作	1. 关闭全局中断和 SysTick 中断，并清除相应的中断标志； 2. 设置 PMU_CTRL0 的 LPMS 位选择 STANDBY 模式； 3. 设置 Core 的 SLEEPDEEP 位，同时将系统时钟切换到	1. WAKEUP 引脚的有效电平 /RTC 中断/RSTN 管脚复位 /IWDT 复位/BOR 复位 2. 唤醒后，CPU 将从 0x00000000 地址开始执行指令代码

		RC32K 并执行 WFI 或者 WFE。	
POWER DOWN	关闭主区和待机区电源	1. 设置 PMU_CTRL0 的 LPMS 位选择 POWERDOWN 模式； 2. 设置 Core 的 SLEEPDEEP 位并执行 WFI 或者 WFE。	1.WAKEUP 引脚（GPIOA0 和 GPIOC13）的有效电平/RSTN 管脚复位 2.唤醒后，重新走上电过程

注：在 STOP0/STOP1 模式下，PLL、XTH 时钟由软件控制开启或关闭；通过配置寄存器 PMU_CTRL0 的 bit4，控制 RC64M 时钟在进入 STOP0/STOP1 模式后，是否自动关闭。

3.3.2. 低功耗模式下各模块工作状态

下表为各个模块在不同功耗模式的汇总

表 3-3 各个模块在不同功耗模式下工作状态

模块	RUN	SLEEP	STOP0	STOP1	STOP2	STANDBY	POW DOWN
CPU	Y	-	-	-	-	-	-
Flash	Y	Y	O	O	-	-	-
SRAM1	Y	Y	Y	Y	-	-	-
SRAM2	Y	Y	Y	Y	Y	-	-
备份寄存器	Y	Y	Y	Y	Y	Y	-
UART/I2C/SPI/I2S	O	O	-	-	-	-	-
LPUART	O	O	O	O	O	-	-
Timer/WDT	O	O	-	-	-	-	-
LPTIM	O	O	O	O	O	-	-
UAC（算法）	O	O	-	-	-	-	-

IWDT	O	O	O	O	O	O	-
USB	O	O	O	O	-	-	-
PLL	O	O	O	O	-	-	-
RC64M	O	O	O	O	-	-	-
RC32K	Y	Y	Y	Y	Y	Y	
XTH	O	O	O	O	-	-	-
XTL	O	O	O	O	O	O	-
ADC	O	O	-	-	-	-	-
DAC	O	O	-	-	-	-	-
COMP	O	O	O	O	-	-	-
LVD	O	O	O	O	-	-	-
MRLDO12	Y	Y	Y	-	-	-	-
LPLDO12	Y	Y	Y	Y	Y	Y	-
RTC	O	O	O	O	O	O	-
GPIO	Y	Y	Y	Y	Y	Y ^{注2}	Y ^{注3}

注 1：Y：使能工作；O：可选择使能或禁止工作；-：停止工作或掉电

注 2：5 个 Wakeup Pin

注 3：2 个 Wakeup Pin

3.4. PMU 区引脚控制说明

PC13、PC14、PC15 引脚属于 PMU 特殊控制引脚，在 Standby 模式下可以保持状态，PMU_IOSEL 设置优先级高于 GPIO 寄存器。

若 PMU_IOSEL 寄存器配置中不做主区 GPIO 功能，则以 PMU_IOSEL 寄存器为最

高优先级，此时 GPIO 寄存器对 PC13、PC14、PC15 的配置将会忽略。

3.5. PMU 寄存器

PMU 寄存器基地址：0x4000_7000

偏置	名称	描述
0x00	PMU_CTRL0	PMU 控制寄存器 0
0x04	PMU_CTRL1	PMU 控制寄存器 1
0x08	PMU_CTRL2	PMU 控制寄存器 2
0x0C	PMU_CTRL3	PMU 控制寄存器 3
0x10	PMU_SR	PMU 状态寄存器
0x14	PMU_STCLR	PMU 状态清除寄存器
0x18	PMU_IOSEL	STANDBY 域 IO 复用寄存器

3.5.1. PMU 控制寄存器 0(PMU_CTRL0)(偏移：00h)

比特	名称	属性	默认值	功能描述
31:28	RSV	-	-	保留
27:16	STOP_WPT	RW	0x3C0	STOP0/STOP1/STOP2 唤醒时间设置 从 STOP 模式唤醒所需的时钟周期数 注： 1、STOP0/STOP1 模式唤醒以 RC64M 时钟计数或以 RC64M 的 16 分频时钟计数（当 RC64MDIVEN 使能时） 2、STOP2 模式唤醒以 RC32K 时钟计数
15	RSV	-	-	保留
14:12	LPLDO12_LV	RW	100	STOP1/STOP2 模式下，LPLDO12 调整选择 100：调整为 1.1V（不变）

				其他：保留
11	RSV	-	-	保留
10	MLDO12_LOWP	RW	0	STOP0 模式下，MLDO 是否进入低功耗模式。 0：STOP0 下 MLDO12 不降低自身功耗 1：STOP0 下 MLDO12 降低自身功耗
9:8	MLDO12_LV	RW	00	STOP/STANDBY 模式下，MLDO12 电压是否自动调整 00：保持为 1.2V 01：调整为 1.0V 10：调整为 0.9V 11：调整为 0.8V
7	RTC_WE	RW	0	RTC 模块写使能 0: RTC 模块写禁止 1: RTC 模块写使能
6	RSV	-	-	保留
5	RC64MDIVEN	RW	0	STOP0/STOP1/STOP2 模式下，RC64M 选择 16 分频输出 0：选择原始不分频输出 1：选择 16 分频输出
4	RC64MPDEN	RW	0	STOP0/STOP1 模式下，RC64M 是否自动关闭 0：RC64M 不自动关闭 1：RC64M 自动关闭 注：PLL、RC32K、XTH、XTL 不会自动关闭，需软件控制
3	RSV	-	-	保留
2:0	LPMS	RW	000	低功耗模式选择位（内核进 DeepSleep 时有效）： 000：STOP0 模式 001：STOP1 模式 010：STOP2 模式

				011: 保留 10x: STANDBY 模式 11x: POWER DOWN 模式

3.5.2. PMU 控制寄存器 1(PMU_CTRL1)(偏移: 04h)

比特	名称	属性	默认值	功能描述
31:16	RSV	-	-	保留
15	LVD_Value	RO	0	LVD 原始状态
14	LVD_Filter	RO	0	LVD Filter 后的状态
13:12	RSV	-	-	保留
11:9	FLT_TIME	RW	000	LVD 滤波时间配置, 滤波时间按 FLTCLK 计算, FLTCLK 的时钟源由 RCC 的 CCR2 的 bit18 来选择。 000: 1 个周期 001: 2 个周期 010: 4 个周期 011: 16 个周期 100: 64 个周期 101: 256 个周期 110: 1024 个周期 111: 4095 个周期
8	LVD_FLTEN	RW	0	数字滤波使能配置 0: 禁止数字滤波 1: 使能数字滤波 如果在 STOP 模式用 LVD 唤醒, 需要禁止数字滤波 或者使用 RC32K 时钟滤波
7:4	RSV	-	-	保留
3:1	LVD_SEL	RW	0000	LVD 阈值电压选择, 下列电压值为触发报警的典型 值, 解除报警电压值为所列电压值+100mV。

				000: 1.71 001: 2.01 010: 2.23 011 : 2.43 100: 2.51 101: 2.73 110 : 2.80 111 : 2.90 其它: 预留
0	LVDEN	RW	0	LVD 使能控制 0: 禁止 LVD 1: 使能 LVD 注: 中断功能在 EXTI 中设置, 复用功能在 RCR 中设置。滤波后为高就产生复位。 使能后, LVD 模拟需要 2us 的稳定时间, 在使能 2us 内可能误报警。

注: 读取 LVD_FLTEN 和 FLT_TIME 的值时, LVD_FLTEN 对应 bit9, FLT_TIME 对应 bit12~10。

3.5.3. 电源控制寄存器 2(PMU_CTRL2) (偏移: 08h)

比特	名称	属性	默认值	功能描述
31:28	RSV	-	-	保留
27:26	BOR_CFG	RW	00	BOR 电压选择 00: 2.1V/2V 01: 2.3V/2.2V 10: 2.61V/2.49V 11: 2.90V/2.77V
25	RSV	-	-	保留
24	BOR_EN	RW	0	BOR 使能

				0: 禁止 BOR 1: 使能 BOR 注: BOR 使能后, 1us 后输出稳定, 可以设置 BOR 复位使能 (BORSTN_EN)。关闭 BOR 前, 请先关闭 BORRST_EN。
23:21	RSV	-	-	保留
20	BORRST_EN	RW	0	BOR 复位使能 0: BOR 复位不使能 1: BOR 复位使能
19	RSV	-	-	保留
18:16	STDBY_WPT	RW	010	STANDBY 模式唤醒等待寄存器 000: 等待 1 个 RC32K 001: 等待 2 个 RC32K 010: 等待 3 个 RC32K ... 110: 等待 7 个 RC32K 111: 无效, 不等待
15:13	RSV	-	-	保留
12:8	WUxFILEN	RW	0	WKUPx(x=5~1)管脚滤波使能, 使用 RTCCLK 滤波, 滤波 1 拍毛刺 0: WKUPx 管脚滤波不使能 1: WKUPx 管脚滤波使能
7:5	RSV	-	-	保留
4:0	EWUPx	RW	0	WKUPx(x=5~1)管脚唤醒功能使能 0: WKUPx 管脚唤醒功能不使能 1: WKUPx 管脚唤醒功能使能, 滤波功能由 WUxFIL 位选择。

3.5.4. 电源控制寄存器 3 (PMU_CTRL3) (偏移: 0Ch)

比特	名称	属性	默认值	功能描述
31:5	RSV	-	-	保留
4:0	WUPOLx	RW	0	WKUPx(x=5~1)管脚唤醒极性选择 0: WKUPx 管脚高电平唤醒 1: WKUPx 管脚低电平唤醒

3.5.5. PMU 状态寄存器(PMU_SR) (偏移: 10h)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	BORN	RO	1	BORN 的原始状态
15	BORWUF	RO	0	BOR 唤醒标志。通过写 CBORWUF 位 (PMU_STCLR 寄存器) 清除 0: 没有 BOR 唤醒事件 1: BOR 发生唤醒事件 BORWUF 不清除, 会引起 STANDBY 模式立刻唤醒
14	IWDTWUF	RO	0	IWDT 唤醒标志。通过写 CIWDTWUF 位 (PMU_STCLR 寄存器) 清除 0: 没有 IWDT 唤醒事件 1: IWDT 发生唤醒事件 IWDTWUF 不清除, 会引起 STANDBY 模式立刻唤醒
13	RSTWUF	RO	0	RSTN 唤醒标志。通过写 CRSTWUF 位 (PMU_STCLR 寄存器) 清除 0: 没有 RSTN 唤醒事件 1: RSTN 发生唤醒事件

				RSTWUF 不清除，会引起 STANDBY 模式立刻唤醒
12	RTCWUF	RO	0	RTC 唤醒标志。通过写 CRTCWUF 位（PMU_STCLR 寄存器）清除 0: 没有 RTC 唤醒事件 1: RTC 发生唤醒事件 RTCWUF 即使不清除，也不会引起 STANDBY 模式唤醒
11:9	RSV	-	-	保留
8	SBF	RO	0	STANDBY 标志。当芯片进入 STANDBY 模式时，硬件自动设置，只能被 POR 或者写 CSBF 位（PMU_STCLR 寄存器）清除 0: 芯片未进入过 STANDBY 模式 1: 芯片进入过 STANDBY 模式
7:5	RSV	-	-	保留
4:0	WUPFx	RO	0	WKUPx(x=5~1)唤醒标志。WKUPx 管脚检测到一个唤醒事件，可以通过写 CWUFx 位（PMU_STCLR 寄存器）清除 0: 没有 WKUPx 唤醒事件 1: WKUPx 管脚发生唤醒事件 注: WUPFx 标志如果为高，进入 STANDBY 后会立刻唤醒

3.5.6. PMU 状态清除寄存器(PMU_STCLR) (偏移: 14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	CBORWUF	WO	0	清除唤醒标志 BORWUF 标志。读始终为 0，写 1 清除唤醒标志

				0: 没有作用 1: 清除唤醒标志
14	CIWDTWUF	WO	0	清除唤醒标志 IWDTWUF 标志。读始终为 0, 写 1 清除唤醒标志 0: 没有作用 1: 清除唤醒标志
13	CRSTWUF	WO	0	清除唤醒标志 RSTWUF 标志。读始终为 0, 写 1 清除唤醒标志 0: 没有作用 1: 清除唤醒标志
12	CRTCWUF	WO	0	清除唤醒标志 RTCWUF 标志。读始终为 0, 写 1 清除唤醒标志 0: 没有作用 1: 清除唤醒标志
11:9	RSV	-	-	保留
8	CSBF	WO	0	清除 STANDBY 标志。读始终为 0, 写 1 清除 STANDBY 标志 0: 没有作用 1: 清除 STANDBY 标志
7:5	RSV	-	-	保留
4:0	CWUF _x	WO	0	清除唤醒标志 WUPF _x (x=5~1)。读始终为 0, 写 1 清除唤醒标志 0: 没有作用 1: 清除唤醒标志

3.5.7. STANDBY 域 IO 复用寄存器(PMU_IOSEL)(偏移: 18h)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留

10	PC15_Value	RW	0	PC15_Value, 具体见 PC15_SEL, 注: 读取的值为之前写入的值。
9	PC14_Value	RW	0	PC14_Value, 具体见 PC14_SEL, 注: 读取的值为之前写入的值。
8	PC13_Value	RW	0	PC13_Value, 具体见 PC13_SEL, 注: 读取的值为之前写入的值。
7	RSV	-	-	保留
6:5	PC15_SEL	RW	0	PC15 管脚功能选择 00: 主区 GPIO 功能, STANDBY 模式下为输入状态 01: 输出 PC15_Value, 以 push-pull 模式 其它: 输出关闭 注: 需要选择 XTL_OUT, 请配置为模拟端口
4:3	PC14_SEL	RW	0	PC14 管脚功能选择 00: 主区 GPIO 功能, STANDBY 模式下为输入状态 01: 输出 PC14_Value, 以 push-pull 模式 其它: 输出关闭 注: 需要选择 XTL_IN, 请配置为模拟端口
2	RSV	-	-	保留
1:0	PC13_SEL	RW	0	PC13 管脚功能选择 00: 主区 GPIO 功能, STANDBY 模式下为输入状态 01: RTC Fout, 选择 RTC 的 FSEL_OUT 输出。此时 PC13_Value 作为 OD 输出 (0) 或者 push-pull 输出 (1)。 10: RTC 输入, 作为时间戳功能 (Tamper) 11: 输出 PC13_Value, 以 push-pull 模式

ALISINOCHIP

4. 复位和时钟单元（RCC）

4.1. 复位源

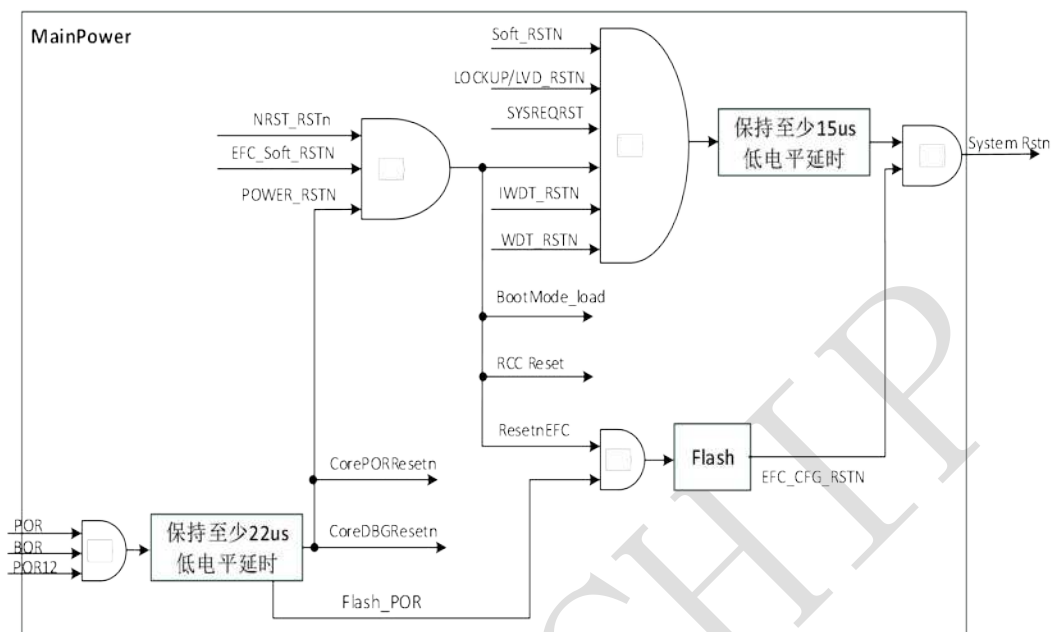
芯片有多个复位源，包括 POR 复位，检测异常复位，看门狗复位，软复位等。具体复位源如下表：

表 4-1 复位源

RESET SOURCE	FUNCTION
芯片上电 POR 复位	复位所有：芯片冷启动时产生该复位。会产生 NVR 预加载。
BOR 复位	复位主区和 IWDT 模块。会产生 NVR 预加载
POR12 复位	复位主区：当芯片从 STANDBY 模式唤醒或芯片冷启动时，产生该复位。会产生 NVR 预加载。
外部 NRST 管脚复位	复位主区：复位信号来自于外部复位引脚，低电平有效。会产生 NVR 预加载。
EFCRST	复位主区：会产生 NVR 预加载，所有的 RCC 寄存器被复位，EFC 被复位，模拟 IP 也被复位
LVDRSTN	复位主区，不影响模拟 IP、EFC 控制器、RCC 寄存器、系统配置寄存器。
LOCKUPRSTN	
SRST	
SYSREQRST	
IWDTRSTN	
WDTRSTN	
STANDBY 电源域软复位	复位 STANDBY 待机区域

各复位源之间的关系如下图所示。

图 4-1 复位源关系图



4.1.1. 上电复位

上电复位 POR：无条件复位整个芯片，包括 RTC 待机区和主区。POR 电压见数据手册。

4.1.2. BOR 复位

欠压复位，复位主区和 IWDN 模块。BOR 电压见数据手册。

4.1.3. POR12 复位

当芯片初次上电或者从 STANDBY 模式唤醒，LDO12 会重新上电，产生 POR12 复位。只对主区有效。

4.1.4. 外部引脚复位

在芯片引脚 RSTN 接低电平，芯片主区复位，RSTN 变高后芯片复位释放。只对主区有效。

4.1.5. EFC 复位

EFC 寄存器复位，继而引起芯片复位。

4.1.6. LVD 复位

当 LVD 模块检测到芯片 VDD 电压低于设定的 LVD 复位电压时，就会产生复位。只对主区有效。LVD 复位电压见数据手册。

4.1.7. LOCKUP 复位

当处理器遇到严重的错误（如在 HardFault 处理函数里再次遇到 HardFault 异常）时，如果 LOCKUP 复位使能（见系统寄存器 RCR），那么就会产生 LOCKUP 复位。只对主区有效。

4.1.8. Soft Reset

在复位控制寄存器 RCC_RCR 的 SRSTNOMAP 位（bit30）写 0，产生系统复位。只对主区有效。

4.1.9. SYSREQRST

调 CMSIS 中的 NVIC SYSREQRST 接口复位内核。只对主区有效。

4.1.10. IWDG 复位

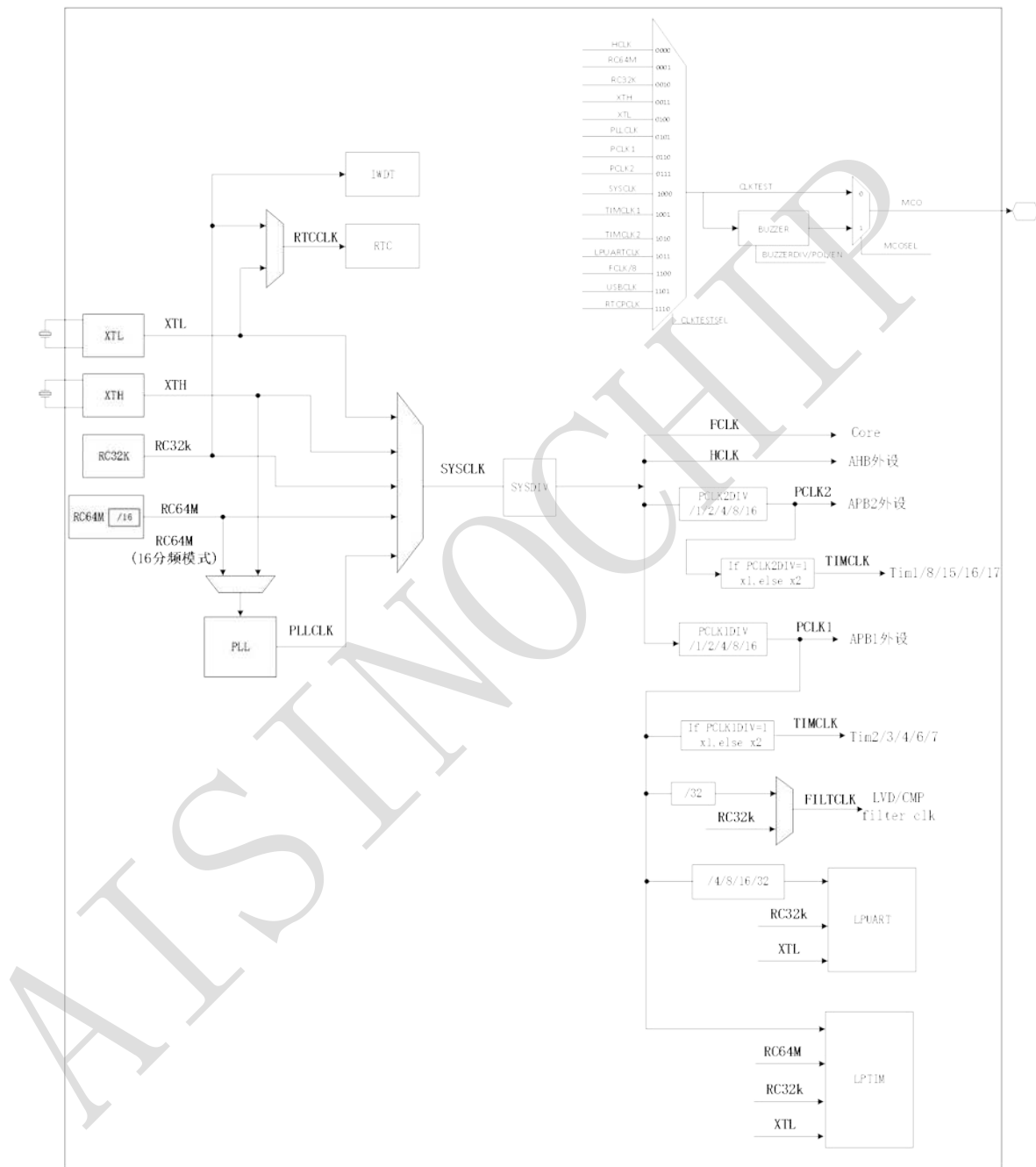
IWDG 计数器溢出产生系统复位。只对主区有效。

4.1.11. WDT 复位

WDT 计数器溢出产生系统复位。只对主区有效。

4.2. 时钟结构图

图 4-2 时钟结构图



4.3. 系统时钟选择

ACM32G103 共有 5 个系统时钟源：

- 内部高速 64MHz RC 振荡器时钟 RC64M
- 内部低速 32KHz RC 振荡器时钟 RC32K
- 外部高速 4~32MHz 晶体振荡器时钟 XTH
- 外部低速 32.768KHz 晶体振荡器时钟 XTL
- 内部锁相环时钟 PLL

根据工作模式不同，采用不同时钟方案，通过配置时钟控制寄存器 CCR 来选择系统时钟的来源。切换时钟时候，请保证目标时钟源已经使能，并且切换完成后才能关闭原时钟源。关系如下表所示：

表 4-2 系统时钟源

SYSCLKSEL	系统时钟源
000	RC64M
001	RC32K
010	XTH
011	XTL
100	PLLCLK

系统上电后，默认使用 RC64M 作为系统时钟 (64MHz，正负偏差 1%以内)。当系统从 STOP 模式唤醒后，自动启用 RC64M 时钟作为系统时钟，后续程序可以配置并使能 PLLCLK，待稳定后切换到更高频率的 PLLCLK。

PLL 时钟的输入可以来自于外部晶振 XTH，也可以来自于 RC64M/16(将 RC64MDIV 设为 1)。PLL 的输入时钟范围为 3MHz-48MHz。

4.4. 时钟输出

ACM32G103 芯片的时钟输出可以通过 MCO1 引脚或 MCO2 引脚。如图 4-2 所示，输出的时钟信号可以通过 CLKTESTSEL 选择，可以来自于系统时钟 SYSCLK，也可以来自于其他时钟源如 XTL（低速晶振），然后可以直接输出到 MCO1/MCO2，也可以通过一个 Buzzer 控制电路再输出到 MCO1/MCO2。Buzzer 控制电路主要影响了输出信号的频率，极性，以及输出使能，具体见 CLKOCR 系统寄存器。BUZZER1 对应 MCO1，

BUZZER2 对应 MCO2。

4.5. 寄存器描述

RCC 寄存器基地址：0x40021000

偏置	名称	描述
0x00	RCC_RCR	复位控制寄存器
0x04	RCC_RSR	复位源状态寄存器
0x08	RCC_APB1_IPRSTR	APB1 外设模块复位寄存器
0x0C	RCC_APB2_IPRSTR	APB2 外设模块复位寄存器
0x10	RCC_AHB_IPRSTR	AHB 外设模块复位寄存器
0x14	RCC_CCR1	时钟控制寄存器 1
0x18	RCC_CCR2	时钟控制寄存器 2
0x1C	RCC_CIR	时钟中断寄存器
0x20	RCC_APB1_IPCKENR	APB1 外设模块使能寄存器
0x24	RCC_APB2_IPCKENR	APB2 外设模块使能寄存器
0x28	RCC_AHB_IPCKENR	AHB 外设模块使能寄存器
0x2C	RCC_RC64MCR	RC64M 控制寄存器
0x30	RCC_XTHCR	XTH 控制寄存器
0x34	RCC_PLLCR	PLL 控制寄存器
0x38	RCC_CLKOCCR	时钟输出控制寄存器
0x3C	RCC_STDBY_CTRL	STANDBY 电源域控制寄存器

4.5.1. 复位控制寄存器 RCC_RCR（偏移：00h）

比特	名称	属性	复位值	描述
31	SRSTMAP	W	1	SOFT RST Map: Soft reset 强制芯片从 Flash 中启动

				复位除了模拟 IP、EFC、SENSOR 和系统寄存器模块的整个系统，下降沿触发。写 0 产生系统复位，下一个时钟自动回 1。
30	SRSTNOMAP	W	1	SOFT RST NO MAP: 复位能力同 SRSTMAP, 但不更改 REMAP 状态。
29	EFCRST	W	1	eFlash 复位，芯片重新从 ROM 启动。
28:17	RSV	-	-	保留
16	REMAPIMP	RW	0	REMAP 立刻使能 0: 无效 1: 写 1 立刻使能 REMAP, 下次 REMAP 先写 0 再写 1。 (Remap 机制请参见存储架构章节)
15:4	RSV	-	-	保留
3	LOCKRSTEN	RW	0	CPU 发生 LOCKUP 后，是否复位系统 0: 不允许其复位系统逻辑 1: 允许 LOCKUP 复位系统逻辑
2	IWDTRSTEN	RW	0	IWDT reset 使能, 使能后允许独立看门狗信号复位系统 0: 不允许其复位系统逻辑 1: 允许看门狗 reset 复位系统逻辑
1	WDTRSTEN	RW	0	WDT reset 使能, 使能后允许看门狗信号复位系统 0: 不允许其复位系统逻辑 1: 允许看门狗 reset 复位系统逻辑
0	LVDRSTEN	RW	0	低压 LVD reset 使能, 使能后允许低电压检测电路产生的信号复位系统 0: 不允许其复位系统逻辑 1: 允许 LVD RESET 复位系统逻辑

4.5.2. 复位源状态寄存器 RCC_RSR （偏移：04h）

复位状态寄存器，记录引起系统复位的原因。需要通过 RSTFLAGCLR 清除，或者上电复位清除。

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	RSTFLAGCLR	WO	0	写 1 清除所有复位标志
15:9	RSV	-	-	保留
10	PWRRSTF	RO	1	PWR 复位标志： 0:上一次复位不是由 POR/BOR 复位设置引起 1:上一次复位由 POR/BOR 复位设置引起 注：此标志在 STANDBY 区域，清除会有延迟。
9	POR12RSTF	RO	1	POR12 复位标志： 0:上一次复位不是由 POR12 复位设置引起 1:上一次复位由 POR12 复位设置引起
8	SRSTF	RO	0	SOFT reset status: 0:上一次复位不是由系统软复位设置引起 1:上一次复位由系统软复位设置引起
7	SRSTNMF	RO	0	SOFT reset NO Map status: 0:上一次复位不是由 No Map 系统软复位设置引起 1:上一次复位由 No Map 系统软复位设置引起
6	EFCRSTF	RO	0	EFC Reset 标志： 0:上一次复位不是由 EFC 复位引起 1:上一次复位由 EFC 复位引起
5	RSTNF	RO	0	RSTN Reset 标志： 0:上一次复位不是由复位管脚引起 1:上一次复位由复位管脚引起
4	SYSREQRSTF	RO	0	CPU SYSREQ Reset 标志：

				0:上一次复位不是由 SYSREQ 引起 1:上一次复位由 SYSREQ 引起
3	LOCKUPRSTF	RO	0	LOCKUP Reset 标志: 0:上一次复位不是由 LOCKUP 引起 1:上一次复位由 LOCKUP 引起
2	IWDTRSTF	RO	0	IWDT Reset 标志: 0:上一次复位不是由独立看门狗模块引起 1:上一次复位由独立看门狗模块引起 注: 此标志在 STANDBY 区域, 清除会有延迟。
1	WDTRSTF	RO	0	WDT Reset 标志: 0:上一次复位不是由看门狗模块引起 1:上一次复位由看门狗模块引起
0	LVDRSTF	RO	0	LVD Reset 标志: 0: 上一次复位不是由低压检测模块引起 1: 上一次复位由低电压检测模块引起

4.5.3. APB1 外设模块复位控制寄存器 RCC_APB1RSTR （偏移：08h）

比特	名称	属性	复位值	描述
31	LPUART1RST	RW	1	LPUART1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
30	LPTIM1RST	RW	1	LPTIM1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
29:28	RSV	-	-	保留

27	PMURST	RW	1	PMU 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
26	RSV	-	-	保留
25	CAN2RST	RW	1	CAN2 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
24	CAN1RST	RW	1	CAN1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
23	RSV	-	-	保留
22	I2C2RST	RW	1	I2C2 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
21	I2C1RST	RW	1	I2C1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
20	RSV	-	-	保留
19	UART4RST	RW	1	UART4 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
18	UART3RST	RW	1	UART3 模块复位 0: 复位

				1: 不复位 写 0 复位, 写 1 退出复位。
17	UART2RST	RW	1	UART2 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
16	RSV	-	-	保留
15	I2S2RST	RW	1	I2S2 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
14	I2S1RST	RW	1	I2S1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
13:12	RSV	-	-	保留
11	WDTRST	RW	1	WDT 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
10:6	RSV	-	-	保留
5	TIM7RST	RW	1	TIM7 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
4	TIM6RST	RW	1	TIM6 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。

3	RSV	-	-	保留
2	TIM4RST	RW	1	TIM4 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
1	TIM3RST	RW	1	TIM3 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
0	TIM2RST	RW	1	TIM2 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。

4.5.4. APB2 外设模块复位控制寄存器 RCC_APB2RSTR (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留
13	TIM17RST	RW	1	TIM17 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
12	TIM16RST	RW	1	TIM16 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
11	TIM15RST	RW	1	TIM15 模块复位 0: 复位

				1: 不复位 写 0 复位, 写 1 退出复位。
10	RSV	-	-	保留
9	UART1RST	RW	1	UART1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
8	TIM8RST	RW	1	TIM8 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
7	RSV	-	-	保留
6	TIM1RST	RW	1	TIM1 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
5	RSV	-	-	保留
4	EXTIRST	RW	1	EXTI 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
3	OPARST	RW	1	OPA 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。
2	COMPRST	RW	1	COMP 模块复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。

1	RSV	-	-	保留
0	SYSCFGRST	RW	1	SYSCFG 复位 0: 复位 1: 不复位 写 0 复位, 写 1 退出复位。

4.5.5. AHB 外设模块复位控制寄存器 RCC_AHBRSTR (偏移: 10h)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29	EXMCRST	RW	1	EXMC 模块复位 0: 复位 1: 不复位
28	RSV	-	-	保留
27	UACRST	RW	1	UAC 模块复位 0: 复位 1: 不复位
26:21	RSV	-	-	保留
20	DAC1RST	RW	1	DAC1 模块复位 0: 复位 1: 不复位
19	RSV	-	-	保留
18	ADC12RST	RW	1	ADC1/ADC2 模块复位 0: 复位 1: 不复位
17:16	RSV	-	-	保留
15	GPIOFRST	RW	1	GPIOF 模块复位 0: 复位

				1: 不复位
14	GPIOERST	RW	1	GPIOE 模块复位 0: 复位 1: 不复位
13	GPIODRST	RW	1	GPIOD 模块复位 0: 复位 1: 不复位
12	GPIOCRST	RW	1	GPIOC 模块复位 0: 复位 1: 不复位
11	GPIOBRST	RW	1	GPIOB 模块复位 0: 复位 1: 不复位
10	GPIOARST	RW	1	GPIOA 模块复位 0: 复位 1: 不复位
9	RSV	-	-	保留
8	SPI3RST	RW	1	SPI3 模块复位 0: 复位 1: 不复位
7	SPI2RST	RW	1	SPI2 模块复位 0: 复位 1: 不复位
6	SPI1RST	RW	1	SPI1 模块复位 0: 复位 1: 不复位
5	CRCRST	RW	1	CRC 模块复位 0: 复位 1: 不复位

4	USBRST	RW	1	USB 控制器复位 0: 复位 1: 不复位
3:1	RSV	-	-	保留
0	DMARST	RW	1	DMA1/DMA2 模块复位 0: 复位 1: 不复位

4.5.6. 时钟控制寄存器 1(RCC_CCR1) (偏移: 14h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	SYSCLKSEL	RW	000	系统时钟 SYSCLK 选择: 000: 来自 RC64M 001: 来自 RC32K 010: 来自 XTH 011: 来自 XTL 1xx: 来自 PLLCLK 注: 在进入 STOP2 模式前, 需要将系统时钟切换到 RC32K 时钟

4.5.7. 时钟控制寄存器 2 RCC_CCR2 (偏移: 18h)

比特	名称	属性	复位值	描述
31	DIVDONE	RO	1	SYSDIV 分频设置计数 256 个时钟完成, 可以更新分频值 0: 计数 256 个时钟未完成 1: 计数 256 个时钟完成

30:22	RSV	-	-	保留
21:20	LPTIM1CLKSEL	RW	00	LPTIM1 时钟选择 00: PCLK 01: RC32K 10: RC64M 11: XTL
19	RSV	-	-	保留
18	FILTCLKSEL	RW	0	FLTCLK 选择, LVD 和 COMP 的滤波 时钟选择 0: PCLK 的 32 分频 1: RC32K
17:16	LPUART1CLKSEL	RW	00	LPUART1 时钟选择 00: RC32K 01: XTL 1x: PCLK 分频, 由 LPUARTDIV 确定
15:14	LPUARTDIV	RW	11	LPUART1 使用 PCLK 分频选择 00:4 分频 01:8 分频 10:16 分频 11:32 分频
13:11	PCLK2DIV	RW	000	PCLK2 分频设置 0xx: 不分频 100: 2 分频 101: 4 分频 110: 8 分频 111: 16 分频
10:8	PCLK1DIV	RW	000	PCLK1 分频设置 0xx: 不分频 100: 2 分频

				101: 4 分频 110: 8 分频 111: 16 分频
7:4	SYSDIV1	RW	0001	系统时钟第二级分频选择, 产生 HCLK 0000:不分频 0001:2 分频 0010:3 分频 0011:4 分频 ... 1100:13 分频 1101:14 分频 1110:15 分频 1111:16 分频
3:0	SYSDIV0	RW	0011	系统时钟第一级分频选择, 产生 SYSCLKDIV0 0000:不分频 0001:2 分频 0010:3 分频 0011:4 分频 ... 1100:13 分频 1101:14 分频 1110:15 分频 1111:16 分频 注: 为了防止频率突变导致电源不稳定, 更改系统时钟和算法时钟分频设置需要逐级变化, 每次更改变化后保证 256 个时钟后 (读取 DIVDONE 为 1) 再更新分频值。

4.5.8. 时钟中断寄存器 RCC_CIR（偏移：1Ch）

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20	PLLLOCKIC	WO	0	PLL LOCK 中断清除。 软件写 1 清除中断 PLLLOCKIF 0：无影响 1：清除 PLL LOCK 中断标志
19	XTHRDYIC	WO	0	XTH 时钟稳定中断清除。 软件写 1 清除中断 XTHRDYIF 0：无影响 1：清除 XTH 时钟稳定中断标志
18	RC64MRDYIC	WO	0	RC64M 时钟稳定中断清除。 软件写 1 清除中断 RC64MRDYIF 0：无影响 1：清除 RC64M 时钟稳定中断标志
17	XTLRDYIC	WO	0	XTL 时钟稳定中断清除。 软件写 1 清除中断 XTLRDYIF 0：无影响 1：清除 XTL 时钟稳定中断标志
16	RC32KRDYIC	WO	0	RC32K 时钟稳定中断清除。 软件写 1 清除中断 RC32KRDYIF 0：无影响 1：清除 RCK 时钟稳定中断标志
15:13	RSV	-	-	保留
12	PLLLOCKIE	RW	0	PLL LOCK 中断使能 0：禁止 PLL LOCK 中断 1：使能 PLL LOCK 中断
11	XTHRDYIE	RW	0	XTH 时钟稳定中断使能

				0: 禁止 XTH 时钟稳定中断 1: 使能 XTH 时钟稳定中断
10	RC64MRDYIE	RW	0	RC64M 时钟稳定中断使能 0: 禁止 RC64M 时钟稳定中断 1: 使能 RC64M 时钟稳定中断
9	XTLRDYIE	RW	0	XTL 时钟稳定中断使能 0: 禁止 XTL 时钟稳定中断 1: 使能 XTL 时钟稳定中断
8	RC32KRDYIE	RW	0	RC32K 时钟稳定中断使能 0: 禁止 RCK 时钟稳定中断 1: 使能 RCK 时钟稳定中断
7.5	RSV	-	-	保留
4	PLLLOCKIF	RO	0	PLL LOCK 中断标志 当 PLL 模块 LOCK, PLLLOCK(或 PLLFREERUN)标志有效后硬件置位。软件设置 PLLLOCKIC 清除 0: 无 PLL LOCK 中断 1: 产生 PLL LOCK 中断
3	XTHRDYIF	RO	0	XTH 时钟稳定中断标志 当 XTH 时钟稳定, XTHRDY 标志有效后硬件置 位。软件设置 XTHRDYIC 清除 0: 无 XTH 时钟稳定中断 1: 产生 XTH 时钟稳定中断
2	RC64MRDYIF	RO	1	RC64M 时钟稳定中断标志 当 RC64M 时钟稳定, RC64MRDY 标志有效后 硬件置位。软件设置 RC64MRDYIC 清除 0: 无 RC64M 时钟稳定中断 1: 产生 RC64M 时钟稳定中断
1	XTLRDYIF	RO	0	XTL 时钟稳定中断标志

				当 XTL 时钟稳定, XTLRDY 标志有效后硬件置位。软件设置 XTLRDYIC 清除 0: 无 XTL 时钟稳定中断 1: 产生 XTL 时钟稳定中断
0	RC32KRDYIF	RO	1	RC32K 时钟稳定中断标志 当 RC32K 时钟稳定, RC32KRDY 标志有效后硬件置位。软件设置 RC32KRDYIC 清除 0: 无 RC32K 时钟稳定中断 1: 产生 RC32K 时钟稳定中断

4.5.9. APB1 外设模块时钟使能寄存器 RCC_APB1ENR (偏移: 20h)

比特	名称	属性	复位值	描述
31	LPUART1CKEN	RW	0	LPUART1 模块时钟使能 0: 禁止 1: 使能
30	LPTIM1CKEN	RW	0	LPTIM1 模块时钟使能 0: 禁止 1: 使能
29:28	RSV	-	-	保留
27	PMUCKEN	RW	0	PMU 模块时钟使能 0: 禁止 1: 使能
26	RSV	-	-	保留
25	CAN2CKEN	RW	0	CAN2 模块时钟使能 0: 禁止 1: 使能
24	CAN1CKEN	RW	0	CNA1 模块时钟使能

				0: 禁止 1: 使能
23	RSV	-	-	保留
22	I2C2CKEN	RW	0	I2C2 模块时钟使能 0: 禁止 1: 使能
21	I2C1CKEN	RW	0	I2C1 模块时钟使能 0: 禁止 1: 使能
20	RSV	-	-	保留
19	UART4CKEN	RW	0	UART4 模块时钟使能 0: 禁止 1: 使能
18	UART3CKEN	RW	0	UART3 模块时钟使能 0: 禁止 1: 使能
17	UART2CKEN	RW	0	UART2 模块时钟使能 0: 禁止 1: 使能
16	RSV	-	-	保留
15	I2S2CKEN	RW	0	I2S2 模块时钟使能 0: 禁止 1: 使能
14	I2S1CKEN	RW	0	I2S1 模块时钟使能 0: 禁止 1: 使能
13:12	RSV	-	-	保留
11	WDTCKEN	RW	0	WDT 模块时钟使能 0: 禁止

				1: 使能
10	RTCCKEN	RW	0	RTC 模块总线时钟使能 0: 禁止 1: 使能
9:6	RSV	-	-	保留
5	TIM7CKEN	RW	0	TIM7 模块时钟使能 0: 禁止 1: 使能
4	TIM6CKEN	RW	0	TIM6 模块时钟使能 0: 禁止 1: 使能
3	RSV	-	-	保留
2	TIM4CKEN	RW	0	TIM4 模块时钟使能 0: 禁止 1: 使能
1	TIM3CKEN	RW	0	TIM3 模块时钟使能 0: 禁止 1: 使能
0	TIM2CKEN	RW	0	TIM2 模块时钟使能 0: 禁止 1: 使能

4.5.10. APB2 外设模块时钟使能寄存器 RCC_APB2ENR (偏移: 24h)

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留
13	TIM17CKEN	RW	0	TIM17 模块时钟使能 0: 禁止

				1: 使能
12	TIM16CKEN	RW	0	TIM16 模块时钟使能 0: 禁止 1: 使能
11	TIM15CKEN	RW	0	TIM15 模块时钟使能 0: 禁止 1: 使能
10	RSV	-	-	保留
9	UART1CKEN	RW	0	UART1 模块时钟使能 0: 禁止 1: 使能
8	TIM8CKEN	RW	0	TIM8 模块时钟使能 0: 禁止 1: 使能
7	RSV	-	-	保留
6	TIM1CKEN	RW	0	TIM1 模块时钟使能 0: 禁止 1: 使能
5	RSV	-	-	保留
4	EXTICKEN	RW	0	EXTI 模块时钟使能 0: 禁止 1: 使能
3	OPACKEN	RW	0	OPA 模块时钟使能 0: 禁止 1: 使能
2	COMPCKEN	RW	0	COMP 模块时钟使能 0: 禁止 1: 使能
1	RSV	-	-	保留

0	SYSCFGCKEN	RW	1	SYSCFG 时钟使能 0: 禁止 1: 使能

4.5.11. AHB 外设模块时钟使能寄存器 RCC_AHBENR (偏移: 28h)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29	EXMCKEN	RW	0	EXMC 模块时钟使能 0: 禁止 1: 使能
28	ROMCKEN	RW	1	ROM 模块时钟使能 0: 禁止 1: 使能
27	FAUCKEN	RW	0	FAU 模块时钟使能 0: 禁止 1: 使能
26	HRNGCKEN	RW	0	HRENG 模块时钟使能 0: 禁止 1: 使能
25	AESCKEN	RW	0	AES 模块时钟使能 0: 禁止 1: 使能
24:21	RSV	-	-	保留
20	DAC1CKEN	RW	0	DAC1 模块时钟使能 0: 禁止 1: 使能
19	RSV	-	-	保留

18	ADC12CKEN	RW	0	ADC1/ADC2 模块时钟使能 0: 禁止 1: 使能
17:16	RSV	-	-	保留
15	GPIOFCKEN	RW	0	GPIOF 模块时钟使能 0: 禁止 1: 使能
14	GPIOECKEN	RW	0	GPIOE 模块时钟使能 0: 禁止 1: 使能
13	GPIODCKEN	RW	0	GPIOD 模块时钟使能 0: 禁止 1: 使能
12	GPIOCKEN	RW	0	GPIOC 模块时钟使能 0: 禁止 1: 使能
11	GPIOBCKEN	RW	0	GPIOB 模块时钟使能 0: 禁止 1: 使能
10	GPIOACKEN	RW	0	GPIOA 模块时钟使能 0: 禁止 1: 使能
9	RSV	-	-	保留
8	SPI3CKEN	RW	0	SPI3 模块时钟使能 0: 禁止 1: 使能
7	SPI2CKEN	RW	0	SPI2 模块时钟使能 0: 禁止 1: 使能

6	SPI1CKEN	RW	0	SPI1 模块时钟使能 0: 禁止 1: 使能
5	CRCKEN	RW	0	CRC 模块时钟使能 0: 禁止 1: 使能
4	USBCKEN	RW	0	USB 控制器时钟使能 0: 禁止 1: 使能
3	EFCKEN	RW	1	eFlash 模块在 SLEEP 模式下时钟使能 0: 禁止 1: 使能
2	SRAMCKEN	RW	1	SRAM 在 SLEEP 模式下时钟使能 0: 禁止 1: 使能
1	DMA2CKEN	RW	0	DMA2 模块时钟使能 0: 禁止 1: 使能
0	DMA1CKEN	RW	0	DMA1 模块时钟使能 0: 禁止 1: 使能

4.5.12. RCM64 模块控制寄存器 RCC_RC64MCR（偏移：2Ch）

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16:8	RC64MTRIM	RW	0x000	RC64M 时钟 TRIM 值 TRIM 值越高频率越高
7:5	RSV	-	-	保留

4	RC64MRDY	RO	1	内部 RC64M 时钟稳定标志 0: RC64M 时钟未稳定 1: RC64M 时钟稳定，时钟有效
3	RC64MDIV	RW	0	选择 16 分频后的输出 0: 不分频输出 1: 选择 16 分频后输出。如果系统时钟为 RC64M，此位置 1 后，系统时钟会变为原来的 1/16。
2:1	RSV	-	-	保留
0	RC64MEN	RW	1	RC64M 模块使能 0: 禁止 1: 使能

4.5.13. XTHCR 模块控制寄存器 RCC_XTHCR（偏移：30h）

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4	XTHRDY	RO	0	XTH 振荡器 ready。XTH 振荡器时钟稳定标志，输出时钟有效。 0: XTH 时钟未稳定 1: XTH 时钟稳定，时钟有效
3:2	XTHRDYTIME	RW	11	XTH 稳定时间设置，以 XTH 时钟计数 00: 1024 个时钟 01: 4096 个时钟 10: 16384 个时钟 11: 32768 个时钟
1	XTHBYP	RW	0	XTH 振荡器旁路使能 0: 禁止 XTL 振荡器旁路模式 1: 使能 XTL 振荡器旁路模式

0	XTHEN	RW	0	XTH 模块使能 0: 禁止 1: 使能

4.5.14. PLL 模块控制寄存器 RCC_PLLCR（偏移：34h）

比特	名称	属性	复位值	描述
31	PLLLOCKSEL	RW	0	PLL LOCK 信号选择，用于产生 PLL LOCK 中断 0: 选择 PLLLOCK（模拟输出） 1: 选择 PLLFREERUN（数字计数）
30	PLLFREERUN	RO	0	PLL 工作状态寄存器。 该位即时地反映 PLL 的工作状态，当 PLL 进入 SLEEP 模式或重新配置频率后，硬件自动清除该位；当 PLL 退出 SLEEP 模式且配置生效后，等待 PLLRUNDLY 设置时间后，硬件自动置位。 1: PLL 正常工作。 0: PLL 正常不正常。
29	PLLLOCK	RO	0	PLL 锁定状态位。 1: PLL 已经锁定。 0: PLL 未锁定。
28:23	PLLRUNDLY	RW	0x10	PLL 锁定等待时钟周期数。 PLL 重新重新配置频率后，至少需要 100us 的时间才能锁定。该控制位设定的锁定时间计算如下： $PLLLOCKDLY * 512 * (1/PCLK)$ PCLK 为 32M 时，1 代表等待约 16us。
22	PLLUPDATEEN	RW	0	PLL 配置更新控制位，下一个周期自动回 0。重新改写 PLLM、PLLN 和 PLLF、PLLSRCSEL

				后，需要该该比特置位新的配置才能生效。复位时间 60* (1/PCLK) 1: PLL 配置更新。 0: PLL 配置不更新。
21	PLLSLEEP	RW	1	PLL 休眠控制位 (PLL 处于复位状态) 1: PLL 进入休眠。 0: PLL 不进入休眠。 注：先使能 PLL，再退出休眠。
20:18	RSV	-	-	保留
17:16	PLLM	RW	01	输出分频控制字段。PLLM 含有输出分频控制字段。对应分频值为 2 的 PLLM 次方。
15	RSV	-	-	保留
14:12	PLLN	RW	001	降频因子分频器字段。PLLN 含有 PLL 输入时钟的分频因子，该值的实际作用是参考频率的分频因子。对应分频值为 (PLLN+1)
11:8	RSV	-	-	保留
7:3	PLLF	RW	0x01	增频因子分频器字段。PLLF 含有 PLL 反馈回路中分频器的分频因子，该值的实际作用是参考频率的倍频因子。对应倍频值为(PLLF+15)。 PLL 输出公式为: $F_{pll} = (F_{in} * (PLLF + 15) / (PLLN + 1)) / 2^{PLLM}$ PLL 输入时钟 F_{in} 频率范围 3~48MHz $F_{in} / (PLLN + 1)$ 频率范围 3~6Mhz $F_{vco} = F_{in} * (PLLF + 15) / (PLLN + 1)$，频率范围在 75~150MHz。
2:1	PLLSRCSEL	RW	00	PLL 时钟源选择 0x: 选择片内 RC64M (需设置 16 分频模式) 1x: 选择片外 XTH
0	PLLEN	RW	0	PLL 模块使能

				0: 不使能, PLL 模块处于 power down 状态 1: 使能, PLL 模块使能

4.5.15. 时钟输出控制寄存器 **RCC_CLKOCR** (偏移: 38h)

比特	名称	属性	复位值	描述
31	BUZZER2EN	RW	0	蜂鸣器 2 时钟输出使能, 对应 MCO2 输出; 蜂鸣器 2 的时钟源见 CLKTESTSEL。 0: 不使能, 信号为 BUZZER2POL 1: 使能
30	BUZZER2POL	RW	0	蜂鸣器 2 时钟极性选择 0: 原极性 (停止的时候为 0) 1: 反极性 (停止的时候为 1)
29:24	BUZZER2DIV	RW	0x3F	蜂鸣器 2 时钟分频值: 分频数为寄存器值+1 默认为 64 分频
23	BUZZER1EN	RW	1	蜂鸣器 1 时钟输出使能, 对应 MCO1 输出。蜂鸣器 1 的时钟源见 CLKTESTSEL。 0: 不使能, 信号为 BUZZER1POL 1: 使能
22	BUZZER1POL	RW	0	蜂鸣器 1 时钟极性选择 0: 原极性 (停止的时候为 0) 1: 反极性 (停止的时候为 1)
21	RSV	-	-	保留
20:5	BUZZER1DIV	RW	0x004F	蜂鸣器 1 时钟分频值: 分频数为寄存器值+1
4	MCOSEL	RW	1	Main Clock Output 信号选择 0: CLKTEST 信号(详见 CLKTESTSEL) 1: BUZZER1/2 信号
3:0	CLKTESTSEL	RW	0000	CLKTEST/BUZZER1/2 选择 0000: HCLK

				0001: RC64M 0010: RC32K 0011: XTH 0100: XTL 0101: PLLCLK 0110: PCLK1 0111: PCLK2 1000: SYSCLK 1001: TIMCLK1 (APB1 总线分频时, 为 PCLK1 的 2 倍; 否则, 与 PCLK1 同频) 1010: TIMCLK2 (APB2 总线分频时, 为 PCLK2 的 2 倍; 否则, 与 PCLK2 同频) 1011: LPUART1CLK 1100: FCLK/8 1101: USBPHY_PLL48M 1110: RTC_PCLK (与 PCLK1 同频) 其他: 无时钟

4.5.16. STANDBY 电源域控制寄存器 RCC_STDBYCTRL (偏移: 3Ch)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23	STDBYRST	RW	1	STANDBY 电源域软复位, 写 0 后下一拍自动变 1 0: 复位 STANDBY 域 1: 不复位 STANDBY 域 注: BOR_EN/BOR_CFG/BORRST_EN 不会被复位
22	RTCEN	RW	1	RTCCLK 使能 0: 不使能

				1: 使能
21:20	RTCSEL	RW	00	RTCCLK 选择 00: RC32K 时钟 01: XTL 时钟 1x: 没有时钟
19:16	RSV	-	-	保留
15:10	RC32KTRIM	RW	0x20	RC32K 模块的 TRIM 值 TRIM 值增大, RC32K 时钟增加。
9	RC32KRDY	RO	1	RC32K 时钟 ready。RC32K 时钟稳定标志。在 RC32KEN 位清除后, RC32RDY 在额外 1 个 RC32K CLK 后变低 0: RC32K 时钟未稳定 1: RC32K 时钟稳定, 时钟有效
8	RC32EN	RW	1	Standby 模式下 RC32K 时钟使能位 0: Standby 模式下 RC32K 不使能 1: Standby 模式下 RC32K 使能 注: 芯片在非 Standby 模式下, RC32K 时钟硬件固定使能, 软件无法禁止
7:6	RSV	-	-	保留
5:3	XTLDRV	RW	111	XTL 振荡器驱动能力选择 XTLDRV[2]决定功耗模式: 0: 正常功耗 1: 低功耗 XTLDRV[1:0]决定功耗: 00: XTL 驱动 0 (最低驱动) 01: XTL 驱动 1 10: XTL 驱动 2 11: XTL 驱动 3 (最高驱动)
2	XTLBYP	RW	0	XTL 振荡器旁路使能 0: 禁止 XTL 振荡器旁路模式

				1: 使能 XTL 振荡器旁路模式
1	XTLRDY	RO	0	XTL 振荡器 ready。XTL 振荡器时钟稳定标志，输出时钟有效。在 XTLEN 位清除后，XTLRDY 在额外 6 个 XTLCLK 后变低 0: XTL 时钟未稳定 1: XTL 时钟稳定，时钟有效
0	XTLEN	RW	0	XTL 振荡器使能 0: XTL 不使能 1: XTL 使能 注：进行写操作后，需要等待至少 1 个 AHB 周期后才能读取

5. 嵌套矢量中断控制器（NVIC）

5.1. 概述

嵌套向量中断控制器(NVIC) 是内核处理器的一个重要组成部分。它与 CPU 处理器内核紧密耦合, 实现低中断延迟以及对新到中断的有效处理, 外部中断信号连接到 NVIC, NVIC 将对这些中断进行优先级排序。

所有的 NVIC 寄存器只能采用字传输。任何试图读/写半字或字节的结果都是不可预知的。

NVIC 寄存器都是小端格式。访问处理器要正确处理处理器的大小端配置。

(关于 NVIC 更详细的内容可查看 ARM China STAR Processor User Guild 官方文档)

5.2. 主要特性

- 支持 56 路可屏蔽向量中断
- 16 个可编程中断优先级
- 可嵌套中断支持
- 中断可屏蔽
- 电平触发和边沿触发

5.3. 中断源

表 5-1 NVIC 中断源表

中断号	中断源	备注
Int0	WDT	
Int1	LVD	连到 EXTI 线 16 的电源电压检测（LVD）中断
Int2	RTC	RTC 全局中断
Int3	保留	
Int4	EFC_INT	
Int5	SRAM_PARITY	SRAM 读校验出错中断

Int6	CLKRDY	时钟 Ready 中断，中断使能见系统寄存器 CIR
Int7	EXTI0	EXTI 线 0 中断
Int8	EXTI1	EXTI 线 1 中断
Int9	EXTI2	EXTI 线 2 中断
Int10	EXTI3	EXTI 线 3 中断
Int11	EXTI4	EXTI 线 4 中断
Int12	DMA1	
Int13	DMA2	
Int14	ADC1_2	ADC1 和 ADC2 全局中断
Int15	保留	
Int16	DAC	
Int17	COMP1_2	连到 EXTI 线 20/21 的 COMP1/COMP2 中断
Int18	USB	USB 全局中断
Int19	CAN1	
Int20	CAN2	
Int21	EXTI9_5	EXTI 线 9~5 中断
Int22	TIM1_BRK_UP_TRG_COM	
Int23	TIM1_CC	捕获/比较中断
Int24	TIM2	
Int25	TIM3	
Int26	TIM6	
Int27	TIM7	
Int28	TIM8_BRK_UP_TRG_COM	
Int29	TIM8_CC	捕获/比较中断
Int30	TIM15	
Int31	TIM16	
Int32	TIM17	
Int33	I2C1	
Int34	I2C2	

Int35	SPI1	
Int36	SPI2	
Int37	SPI3	
Int38	I2S1	
Int39	I2S2	
Int40	UART1	
Int41	UART2	
Int42	UART3	
Int43	UART4	
Int44	EXTI15_10	EXTI 线 15~10 中断
Int45	USB_WAKEUP	连到 EXTI 线 23 的 USB 唤醒中断
Int46	LPUART1	
Int47	LPTIM1	LPTIM1 全局中断
Int48	保留	
Int49	AES	
Int50	FPU	
Int51	保留	
Int52	保留	
Int53	TIM4	
Int54	COMP3_4	连到 EXTI 线 25/26 的 COMP3/COMP4 中断
Int55	IWDT_WAKEUP	连到 EXTI 线 19 的 IWDT 唤醒中断

6. 外设内部互连

6.1. 概述

处理器芯片的多个外围设备之间有着一些内部相互连接关系。这允许外围设备之间在芯片内部通信或同步，从而节省 CPU 资源和电源消耗。此外，这些硬件连接消除了软件延迟和外部电路的不确定性。

6.2. 定时器互连关系

6.2.1. 定时器内部触发源（ITR）互连关系

表 6-1 定时器 ITR 互连关系表

ITR 源	定时器（TIMx）					
	TIM1	TIM2	TIM3	TIM4	TIM8	TIM15
ITR0	-	tim1_trgo	tim1_trgo	tim1_trgo	tim1_trgo	tim1_trgo
ITR1	tim2_trgo	-	tim2_trgo	tim2_trgo	tim2_trgo	tim2_trgo
ITR2	tim3_trgo	tim3_trgo	-	tim3_trgo	tim3_trgo	tim3_trgo
ITR3	tim4_trgo	tim4_trgo	tim4_trgo	-	tim4_trgo	tim4_trgo
ITR4	-	-	-	-	-	-
ITR5	tim8_trgo	tim8_trgo	tim8_trgo	tim8_trgo	-	tim8_trgo
ITR6	tim15_trgo	tim15_trgo	tim15_trgo	tim15_trgo	tim15_trgo	-
ITR7	tim16_oc1	tim16_oc1	tim16_oc1	tim16_oc1	tim16_oc1	tim16_oc1
ITR8	tim17_oc1	tim17_oc1	tim17_oc1	tim17_oc1	tim17_oc1	tim17_oc1

注：

- 1、ITR 即 internal trigger，适用于定时器间互连的内部触发，使用的是内部信号，没有对应的管脚。
- 2、定时器的互连由主定时器的输出（如 TRGO\OC 信号）和从定时器的 ITR 组成，从定时器必须配置成从模式，并通过 SMCR 寄存器的 TS 位选择相应的触发信号，其中就包括 ITR 信号。
- 3、主定时器的 TRGO 即 trigger out 信号，由定时器 CR2 的 MMS 决定。OC 信号即输出比较信号，见定时器结构框图右边的 OCx 信号。

6.2.2. 定时器外部触发源（ETR）互连关系

表 6-2 定时器 ETR 互连关系表

ETR 源	定时器（TIMx）				
	TIM1	TIM2	TIM3	TIM4	TIM8
ETR0	tim1_etr(IO)	tim2_etr(IO)	tim3_etr(IO)	tim4_etr(IO)	tim8_etr(IO)
ETR1	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out
ETR2	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out
ETR3	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out
ETR4	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out
ETR5	adc1_awd	tim3_etr(IO)	tim2_etr(IO)	tim3_etr(IO)	adc2_awd
ETR6	adc2_awd	tim4_etr(IO)	tim4_etr(IO)	-	-
ETR7	-	-	adc2_awd	-	-

注：

- 1、ETR 即 external trigger，触发信号来自于外部引脚或外部非定时器信号。
- 2、当 ETR 来自于外部引脚时，该引脚必须配置成定时器的 ETR 复用功能。
- 3、timx_etr(IO)表示外部触发信号来自于 IO。

6.2.3. OCxREF 清除输入信号互连关系

表 6-3 定时器 ocref_clr 互连关系表

ocref_clr 源	定时器（TIMx）							
	TIM1	TIM2	TIM3	TIM4	TIM8	TIM15	TIM16	TIM17
ocref_clr0	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out
ocref_clr1	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out
ocref_clr2	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out
ocref_clr3	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out

注：

- 1、ocref：输出比较参考信号，即定时器结构框图中捕获/比较寄存器的下一级信号。
- 2、ocref_clr 信号，当定时器的 CCMR.OCxCE 信号为 1 时，ocref 可以被 OCREF_CLR_INT 清除掉（输出低），直到更新事件到来。OCREF_CLR_INT 可以选择来源于 ETRF（OCCS 置 1），也可以通过定时器的 AF2 来选择上述表中的 ocref_clr 信号。

6.2.4. 定时器输入信号互连关系

表 6-4 定时器输入 TI1 互连关系表

TI1 源	定时器 (TIMx)							
	TIM1	TIM2	TIM3	TIM4	TIM8	TIM15	TIM16	TIM17
ti1_in0	tim1_ch1	tim2_ch1	tim3_ch1	tim4_ch1	tim8_ch1	tim15_ch1	tim16_ch1	tim17_ch1
ti1_in1	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out	MCO	MCO
ti1_in2	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out	XTH_DIV32	USB_SOF
ti1_in3	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out	MCO	RTC_wakeup	RTC_wakeup
ti1_in4	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out	UART1_RX	RC32K	RC32K

注：

- 1、该表指的是各个定时器的 TI1 除了来自于引脚上的信号外，其他的可以连接到 TI1 的信号。

表 6-5 定时器输入 TI2 互连关系表

TI2 源	定时器 (TIMx)					
	TIM1	TIM2	TIM3	TIM4	TIM8	TIM15
ti2_in0	tim1_ch2	tim2_ch2	tim3_ch2	tim4_ch2	tim8_ch2	tim15_ch2
ti2_in1	-	comp1_out	comp1_out	comp1_out	-	comp2_out
ti2_in2	-	comp2_out	comp2_out	comp2_out	-	comp3_out
ti2_in3	-	comp3_out	comp3_out	comp3_out	-	MCO
ti2_in4	-	comp4_out	comp4_out	comp4_out	-	UART2_RX

注：

- 1、该表指的是各个定时器的 TI2 除了来自于引脚上的信号外，其他的可以连接到 TI2 的信号。

表 6-6 定时器输入 TI3 互连关系表

TI3 源	定时器 (TIMx)				
	TIM1	TIM2	TIM3	TIM4	TIM8
ti3_in0	tim1_ch3	tim2_ch3	tim3_ch3	tim4_ch3	tim8_ch3
ti3_in1	-	comp4_out	comp3_out	-	-

注：

- 1、该表指的是各个定时器的 TI3 除了来自于引脚上的信号外，其他的可以连接到 TI3 的信号。

表 6-7 定时器输入 TI4 互连关系表

TI4 源	定时器 (TIMx)				
	TIM1	TIM2	TIM3	TIM4	TIM8

ti4_in0	tim1_ch4	tim2_ch4	tim3_ch4	tim4_ch4	tim8_ch4
ti4_in1	-	comp1_out	-	-	-
ti4_in2	-	comp2_out	-	-	-

注：

1、该表指的是各个定时器的 TI4 除了来自于引脚上的信号外，其他的可以连接到 TI4 的信号。

6.2.5. 定时器刹车输入互连关系

表 6-8 定时器刹车输入源互联关系

刹车输入源	定时器 (TIMx)				
	TIM1	TIM8	TIM15	TIM16	TIM17
IO	tim1_bkin	tim8_bkin	tim15_bkin	tim16_bkin	tim17_bkin
comp1	comp1_out	comp1_out	comp1_out	comp1_out	comp1_out
comp2	comp2_out	comp2_out	comp2_out	comp2_out	comp2_out
comp3	comp3_out	comp3_out	comp3_out	comp3_out	comp3_out
comp4	comp4_out	comp4_out	comp4_out	comp4_out	comp4_out
System Break	LOCKUP	LOCKUP	LOCKUP	LOCKUP	LOCKUP
	SRAM_PARITY	SRAM_PARITY	SRAM_PARITY	SRAM_PARITY	SRAM_PARITY
	LVD	LVD	LVD	LVD	LVD
	Flash_PARITY	Flash_PARITY	Flash_PARITY	Flash_PARITY	Flash_PARITY

注：

1、来自于 IO 的 break 信号可以配置滤波，如果定时器内部时钟没有关闭的情况下。

2、LOCKUP 是指 CPU 因多次发生错误产生 LOCKUP 信号，LVD 是指芯片 LVD 检测到电源电压低于 LVD 设定的阈值，SRAM_PARITY 和 Flash_PARITY 是指读访问时产生校验错误。

7. 外部中断/事件控制器（EXTI）

7.1. 概述

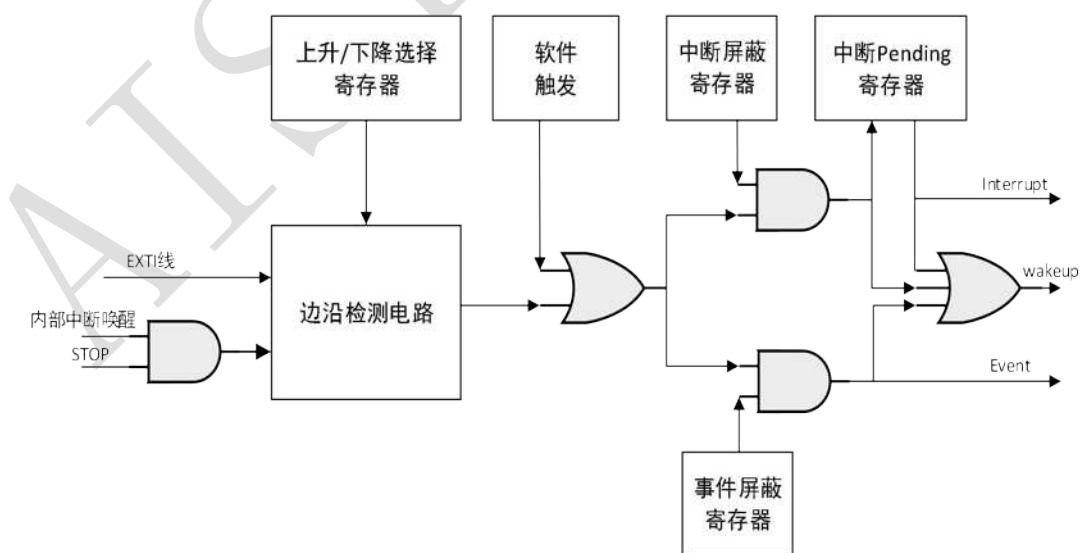
EXTI 包含 26 个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI 提供 3 种触发类型，其中请求源 0~15 为 GPIO 管脚可支持上升沿触发，下降沿触发和任意沿触发，其他请求源默认使用上升沿触发。EXTI 中每个边沿检测电路都可以分别配置或屏蔽。挂起寄存器保持着状态线的中断请求。

7.2. 主要特性

- EXTI 支持多达 26 个软件的中断/事件请求
- 每个中断/事件都有独立的触发和屏蔽
- 每个中断线都有专用的状态位

7.3. 外部中断及事件（EXTI）框图

图 7-1 外部中断及事件框图



7.4. 外部中断及事件（EXTI）触发源

EXTI 触发源包括来自 I/O 管脚的 16 根线以及来自内部模块的 10 根线，包括 LVD、RTC、LPUART1、IWDT（唤醒）、COMP1、COMP2、LPTIM1 和 USB_Wakeup、COMP3、COMP4。通过配置 EXTICR 寄存器，所有的 GPIO 管脚都可以被选作 EXTI 的触发源

表 7- 1 EXTI 源

EXTI 序号	对应源
0	PA0/PB0/PC0/PD0/PE0/PF0
1	PA1/PB1/PC1/PD1/PE1/PF1
2	PA2/PB2/PC2/PD2/PE2/PF2
3	PA3/PB3/PC3/PD3/PE3/PF3
4	PA4/PB4/PC4/PD4/PE4/PF4
5	PA5/PB5/PC5/PD5/PE5
6	PA6/PB6/PC6/PD6/PE6
7	PA7/PB7/PC7/PD7/PE7
8	PA8/PB8/PC8/PD8/PE8
9	PA9/PB9/PC9/PD9/PE9
10	PA10/PB10/PC10/PD10/PE10
11	PA11/PB11/PC11/PD11/PE11
12	PA12/PB12/PC12/PD12/PE12
13	PA13/PB13/PC13/PD13/PE13
14	PA14/PB14/PC14/PD14/PE14
15	PA15/PB15/PC15/PD15/PE15
16	LVD
17	RTC(中断)
18	LPUART1_Wakeup（中断）
19	IWDT
20	COMP1 输出信号
21	COMP2 输出信号

22	Reserved
23	USB_Wakeup (USB resume 信号)
24	LPTIM1_Wakeup (中断)
25	COMP3 输出信号
26	COMP4 输出信号
27~31	Reserved

7.5. 唤醒管理

唤醒事件可以通过下述配置产生：

- 在 SLEEP 模式下，在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时在 ARM 内核系统控制寄存器 (SCB_SCR) 中使能 SEVONPEND (bit4)。当 CPU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位(在 NVIC 中断清除挂起寄存器中)。
- 在 SLEEP 模式下，在外设的控制寄存器使能一个中断，同时在 NVIC 中使能。当 CPU 从 SLEEP 恢复后，通过中断处理函数处理相应中断。
- 在 STOP 模式下，配置一个外部或内部 EXTI 线为事件模式(EENR)，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。
- 在 STOP 模式下，使能 EXTI 的 NVIC 中断(INT5)，根据外部 EXTI 线的边沿检测需求设置 2 个触发寄存器(RTENR/FTENR)，同时在中断屏蔽寄存器的相应位写‘1’允许中断请求(IENR)。当外部中断线上发生了期待的边沿时，将产生一个 EXTI 中断请求，对应的挂起位(PDR)也随之被置‘1’。

7.6. 功能说明

7.6.1. EXTI 中断

EXTI 具有 26 个触发源，每个触发源都可以触发 EXTI 中断。

EXTI 中断配置流程：

- 配置 EXTICR1、EXTICR2，选择 GPIO 引脚作为触发源。
- 配置 IENR，使能所需的中断线。
- 配置 RTENR、FTENR 选择上升沿触发、下降沿触发或任意沿触发。
- 当线上发生了需要的边沿时，PDR 寄存器相应的位会置位。进入 EXTI 中断。

注：通过在寄存器 EXTI_SWIER 对应位写‘1’，也可以通过软件产生中断

7.6.2. EXTI 中断/事件唤醒

EXTI 可以通过中断或者事件唤醒 MCU。

硬件中断/事件源选择

通过下面的过程来配置 26 个线路做为中断/事件源：

- 配置 EXTICR1、EXTICR2，选择 GPIO 引脚作为触发源。
- 配置 26 个中断线的使能位 IENR 或配置 26 个事件线的使能位 EENR。
- 配置 RTENR、FTENR 选择上升沿触发、下降沿触发。
- 如果使用中断唤醒，需要配置 NVIC 控制寄存器中相应的使能位和屏蔽位，使得 26 个中断线中的请求可以被正确地响应。

7.7. EXTI 寄存器

EXTI 寄存器基地址：0x4001_0400

偏置	名称	描述
0x0000	EXTI_IENR	中断使能寄存器
0x0004	EXTI_EENR	事件使能寄存器
0x0008	EXTI_RTENR	上升沿触发使能寄存器
0x000C	EXTI_FTENR	下降沿触发使能寄存器
0x0010	EXTI_SWIER	软件中断事件寄存器
0x0014	EXTI_PDR	中断挂起寄存器
0x0018	EXTI_CR1	外部 I/O 选择寄存器 1
0x001C	EXTI_CR2	外部 I/O 选择寄存器 2

7.7.1. 中断使能寄存器 EXTI_IENR (偏移: 00h)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:0	INTENx	RW	0x1060000	EXTIx 上的中断使能 0: 禁止来自线 x 上的中断请求; 1: 使能来自线 x 上的中断请求。

7.7.2. 事件使能寄存器 EXTI_EENR (偏移: 04h)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:0	EVENx	RW	0x0000000	EXTIx 上的事件使能 0: 禁止来自线 x 上的事件请求; 1: 使能来自线 x 上的事件请求。

7.7.3. 上升沿触发使能寄存器 EXTI_RTENR (偏移: 08h)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:0	RTENx	RW	0x1060000	EXTIx 上的上升沿触发使能 0: 禁止来自线 x 上的上升沿触发; 1: 使能来自线 x 上的上升沿触发。 注: 外部唤醒线是边沿触发的, 这些线上不能出现毛刺信号。

7.7.4. 下降沿触发使能寄存器 EXTI_FTEMR (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:0	FTEMRx	RW	0x000000	EXTIx 上的下降沿触发使能 0: 禁止来自线 x 上的下降沿触发; 1: 使能来自线 x 上的下降沿触发。 注: 外部唤醒线是边沿触发的, 这些线上不能出现毛刺信号。

7.7.5. 软件中断事件寄存器 EXTI_SWIER (偏移: 10h)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:0	SWIERx	WO	0x000000	EXTIx 上的软件中断 当该位为 '0' 时, 写 '1' 将设置 PDR 中相应的挂起位。如果在 IENR 中允许产生该中断, 则此时将产生一个中断。

7.7.6. 中断挂起寄存器 EXTI_PDR (偏移: 14h)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:0	PDRx	RC_W1	0x000000	EXTIx 上的挂起位 0: 没有发生触发请求 1: 发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件, 该位被置 '1'。在该位中写 '1' 可以清除它。

7.7.7. 外部中断配置寄存器 1 EXTI_CR1 (偏移: 18h)

比特	名称	属性	复位值	描述
31:28	EXTI 7	RW	0000	EXTI 7 源选择。 0000: PA7 0001: PB7 0010: PC7 0011: PD7 0100: PE7
27:24	EXTI 6	RW	0000	EXTI 6 源选择。 0000: PA6 0001: PB6 0010: PC6 0011: PD6 0100: PE6
23:20	EXTI 5	RW	0000	EXTI 5 源选择。 0000: PA5 0001: PB5 0010: PC5 0011: PD5 0100: PE5
19:16	EXTI 4	RW	0000	EXTI 4 源选择。 0000: PA4 0001: PB4 0010: PC4 0011: PD4 0100: PE4 0101: PF4

15:12	EXTI 3	RW	0000	EXTI 3 源选择。 0000: PA3 0001: PB3 0010: PC3 0011: PD3 0100: PE3 0101: PF3
11:8	EXTI 2	RW	0000	EXTI 2 源选择。 0000: PA2 0001: PB2 0010: PC2 0011: PD2 0100: PE2 0101: PF2
7:4	EXTI 1	RW	0000	EXTI 1 源选择。 0000: PA1 0001: PB1 0010: PC1 0011: PD1 0100: PE1 0101: PF1
3:0	EXTI 0	RW	0000	EXTI 0 源选择。 0000: PA0 0001: PB0 0010: PC0 0011: PD0 0100: PE0 0101: PF0

7.7.8. 外部中断配置寄存器 2 EXTI_CR2 (偏移: 1Ch)

比特	名称	属性	复位值	描述
31:28	EXTI 15	RW	0000	EXTI 15 源选择。 0000: PA15 0001: PB15 0010: PC15 0011: PD15 0100: PE15
27:24	EXTI 14	RW	0000	EXTI 14 源选择。 0000: PA14 0001: PB14 0010: PC14 0011: PD14 0100: PE14
23:20	EXTI 13	RW	0000	EXTI 13 源选择。 0000: PA13 0001: PB13 0010: PC13 0011: PD13 0100: PE13
19:16	EXTI 12	RW	0000	EXTI 12 源选择。 0000: PA12 0001: PB12 0010: PC12 0011: PD12 0100: PE12
15:12	EXTI 11	RW	0000	EXTI 11 源选择。 0000: PA11

				0001: PB11 0010: PC11 0011: PD11 0100: PE11
11:8	EXTI 10	RW	0000	EXTI 10 源选择。 0000: PA10 0001: PB10 0010: PC10 0011: PD10 0100: PE10
7:4	EXTI 9	RW	0000	EXTI 9 源选择。 0000: PA9 0001: PB9 0010: PC9 0011: PD9 0100: PE9
3:0	EXTI 8	RW	0000	EXTI 8 源选择。 0000: PA8 0001: PB8 0010: PC8 0011: PD8 0100: PE8

8. Flash 控制器（EFC）

8.1. 概述

芯片上集成了 320KB 的 eFlash 存储器，用于保存芯片所有的关键脱机信息和数据。EFC 为 eFlash 控制器，在 CPU 的配置下，完成 eFlash Read、Program、Erase 等操作。

8.2. 主要特性

- 支持 eFlash 的读（8/16/32bit）、写（32bit）、擦除等操作流程。
- Read 等待时间由软件配置。
- 主区多达 640 个 page，每个 page 512 字节。
- NVR 区有 8 个 page，每个 page 512 字节。
- chip 擦除时间 10ms，page 擦除时间为 2ms，word 写 30us（max），读时间 50ns（max）。

8.3. 功能描述

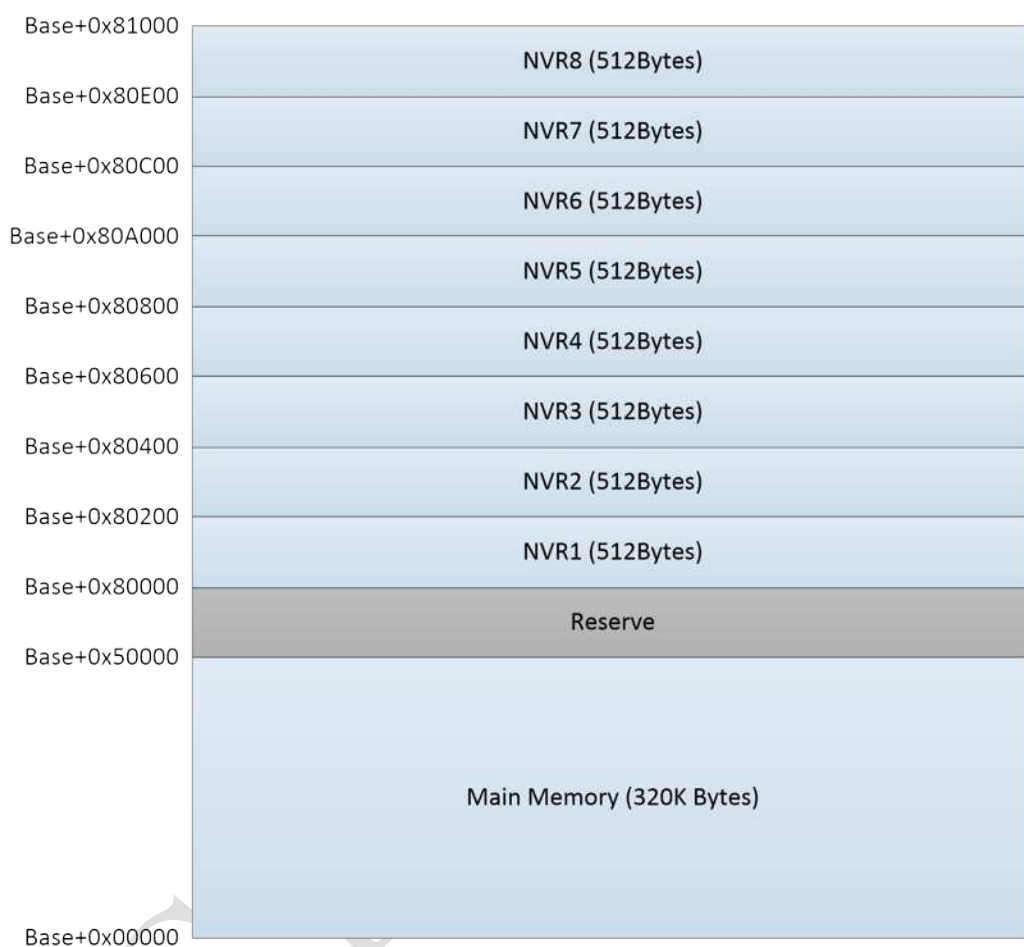
8.3.1. 主区闪存结构

主区闪存包括 320K byte eFlash，每 Page 大小为 512 字节。

闪存块	名称	地址范围	大小
主区闪存	Page 0	0x00000000~0x000001FF	512Byte
	Page 1	0x00000200~0x000003FF	512Byte
	Page 2	0x00000400~0x000005FF	512Byte
	...	...	...
	...	...	...
	Page 639	0x0004FE00~0x0004FFFF	512Byte

8.3.2. 储存地址映射

图 8-1 eFlash 地址映射



注：Base: ROM 启动为 0x1000_0000；eFlash 启动为 0x0000_0000。

8.4. 寄存器描述

寄存器基地址：0x40022000。

偏置	名称	描述
0x00	EFC_CTRL	控制寄存器。
0x04	EFC_SEC	写擦除操作安全寄存器。
0x08	EFC_ADCT	写擦除地址校验寄存器。
0x0c	EFC_ERTO	擦除超时寄存器。

0x10	EFC_WRT0	写超时寄存器。
0x14	EFC_STATUS	状态寄存器。
0x18	EFC_INTSTATUS	中断状态寄存器。
0x1c	EFC_INEN	中断使能寄存器。

8.4.1. 控制寄存器 EFC_CTRL (偏移: 0000h)

比特	名称	属性	默认值	功能描述
31:15	RSV	-	-	保留
14	EWCntEn	RW	0	Flash 擦/写完成标志选择位。 1: Flash 在擦写操作时, 将不再依靠 TBIT 信号(TBIT 为 1 表示 Flash 处于擦/写状态)判断擦写是否完成, 而是由 EFC_ERTO 和 EFC_WRT0 来判断擦写是否完成。此位为 1 时, Fto_En 位也需要置 1, 否则控制器将不会判断擦写是否完成, 控制器将无法退出擦写操作。 0: 使用 TBIT 信号作为擦写完成的判断。当此位为 0 时, 如果 Fto_En 位为 1, 则当 EFC_ERTO 和 EFC_WRT0 定时超时或者 TBIT 信号有一个条件成立时, 系统即退出擦写操作。 注: TBIT 是 eFlash 存储器输出的一个信号, 反映了 EFlash 的擦写是否完成。
13	Resetb	RW	1	eFlash 复位控制位。 0: eFlash 复位, 任何操作发起无效; 1: eFlash 复位释放, 可以发起操作; 任何操作需要在 Resetb 写 1 后延迟 10us 后才能发起。
12	SLEEP	RW	0	eFlash Sleep 模式启动位。

				0: eFlash 退出 SLEEP 模式; 1: eFlash 进入 SLEEP 模式; 退出 SLEEP 模式时, 要在 SLEEP 写 0 时延迟 10us, 再去操作 eFlash。进入 SLEEP 模式的最短时间为 20us。
11:7	Rd_Wait	RW	5	读等待时间设置位。读操作时 eFlash 端口等待的时钟周期数为 Rd_wait+1.0。 eFlash 要求读等待时间至少为 50ns, 即 $(Rd\_Wait+1.0)*\text{系统频率周期} \geq 50\text{ns}$ (Recall read 模式下为 95ns)。 0: 等待 1.0 个系统时钟周期; 1: 等待 2.0 个系统时钟周期; MCU 取址时间为: Rd_wait + 3
6	Fto_En	RW	0	写/擦除操作超时使能; 1: 写/擦除超时功能使能; 0: 写/擦除超时功能禁止;
5:3	RSV	-	-	保留
2	Chip_Erase_Mode	RW	0	Chip Erase Mode 模式设置位。 1: Chip Erase Mode 模式使能; 0: Chip Erase Mode 模式禁止。
1	Page_Erase_Mode	RW	0	Page Erase Mode 模式设置位。 1: Page Erase Mode 模式使能; 0: Page Erase Mode 模式禁止。
0	Write_Mode	RW	0	Write 模式设置位。 1: 写操作模式使能; 0: 写操作模式禁止。

8.4.2. 写擦安全寄存器 EFC_SEC (偏移: 0004h)

比特	名称	属性	默认值	功能描述
31:0	Write_Lock_Ser	W	0	向 eFlash 写数据/擦除前, 须向此寄存器内写 0x55aaaa55 值, 否则控制其忽略此次写操作。

8.4.3. 擦除超时寄存器 (EFC_ERTO) (偏移: 000ch)

比特	名称	属性	默认值	功能描述
31:15	RSV	-	-	保留
14:0	ErSd	RW	0x7fff	擦除 TimeOut 设置位。每一个单位代表 256 个系统时钟周期。当 eFlash 擦除操作在此时间断内未结束时, 控制器将认为此操作有误, 产生 TimeOut 状态信息, 并复位内部状态机。

8.4.4. 写超时寄存器 (EFC_WRTO) (偏移: 0010h)

比特	名称	属性	默认值	功能描述
31:8	RSV	-	-	保留
7:0	WrCycle	RW	0xff	写操作 TimeOut 设置位。每一个单位代表 256 个系统时钟周期。当 eFlash 写操作在此时间断内未结束时, 控制器将认为此操作有误, 产生 TimeOut 状态信息, 并复位内部状态机。

8.4.5. 状态寄存器 EFC_STATUS (偏移: 0014h)

比特	名称	属性	默认值	功能描述
31:18	RSV	-	-	保留
17	TBIT	RO	0	此位反映 eFlash TBIT 信号真实状态 0: Flash 未处于擦/写状态

				1: Flash 处于擦/写状态
16: 14	RSV	-	-	保留
13	RDP1_EN	RO	0	RDP Level1 使能 0: RDP level1 未使能 1: RDP level1 使能
12: 9	RSV	-	-	保留
8	flash_Crypt	RO	0	Flash 加解密状态: 0: Flash 加密功能禁止 1: Flash 加密功能使能 芯片出厂是加密使能的, 不能设置为加密禁止。
7	JTAG_disable	RO	0	JTAG 禁止状态: 0: JTAG 接口未禁止 1: JTAG 接口禁止
6:5	RSV	-	-	保留
4	Nvr4_Lock	RO	0/1	Nvr4 区是否锁住。 1: Nvr4 区已经锁住; 0: Nvr4 区没有锁住。
3	Nvr3_Lock	RO	0/1	Nvr3 区是否锁住。 1: Nvr3 区已经锁住; 0: Nvr3 区没有锁住。
2	Nvr2_Lock	RO	1	Nvr2 区是否锁住。 1: Nvr2 区已经锁住; 0: Nvr2 区没有锁住。
1	Nvr1_Lock	RO	0/1	Nvr1 区是否锁住。 1: Nvr1 区已经锁住; 0: Nvr1 区没有锁住。
0	EFlash_Ready	RO	1	eFlash 状态指示位。该反映 eFlash 工作的状态。 1: eFlash 状态空闲; 0: eFlash 状态忙。

8.4.6. 中断状态寄存器 EFC_INTSTATUS (偏移: 0018h)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9	ErWr_To_Err	RWC	0	擦除/写操作发生 TimeOut 中断状态位。 1: 发生擦除/写操作 TimeOut; 0: 未发生擦除/写操作 TimeOut。 写 1 清 0。
8:6	RSV	-	-	保留
5	Nvr_Err	RW	0	1: Nvr 区发生操作错误。对 Lock 住的 Nvr 区域进行读之外的任何操作, 或者对任一 Nvr 区域进行 Chip Erase 操作时 (不论有没有 Lock 住), 此位均会置 1; 0: 正常状态。 写 1 清 0。
4:1	RSV	-	-	保留
0	ErWr_done	RW	0	写/擦除完成中断状态位。 1: 写/擦除完成标志。写 1 清除该位。如果中断允许, 则产生中断; 0: 写/擦除未完成。

8.4.7. 中断使能寄存器 EFC_INEN (偏移: 001Ch)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9	ErWr_To_IE	RW	0	擦除/写操作发生 TimeOut 中断使能。 1: 擦除/写操作 TimeOut 中断使能; 0: 擦除/写操作 TimeOut 中断不使能。
8:6	RSV	-	-	保留
5	Nvr_Err_IE	RW	0	Nvr_Err 中断使能。

				1: Nvr_Err 中断使能; 0: Nvr_Err 中断不使能。
4:1	RSV	-	-	保留
0	Er_done_IE	RW	0	擦除完成中断使能。 1: 写/擦除中断使能; 0: 写/擦除中断不使能;

8.5. 使用说明

8.5.1. Read 操作

eFlash 上电稳定后可以执行读操作。读操作注意配置读等待时间 RD_WAIT，最小值为 50ns。配置太小会导致读取 eFlash 数据出错。

8.5.2. Program 操作

eFlash 上电稳定后可以执行写操作。写操作之前需要向 EFC_SEC 寄存器内写 0x55aaaa55，否则 EFC 忽略此次写操作。

每次只能写一个 Word。

图 8-2 Program 操作流程



8.5.3. Erase 操作

Page Erase 时，每次只能擦除一个 Page。

图 8-3 Erase 操作流程



9. DMA 控制器（DMA）

9.1. 概述

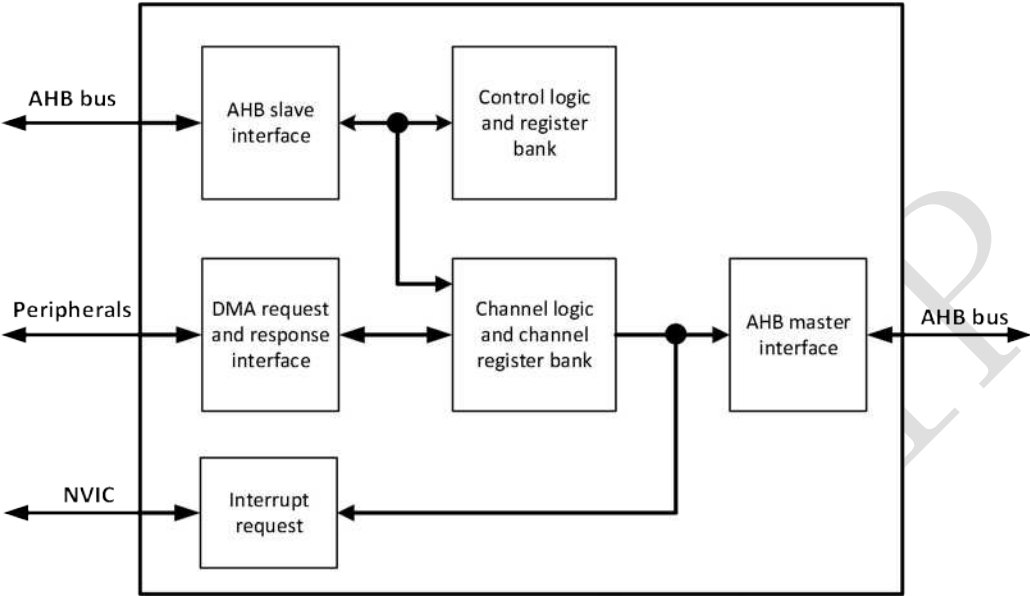
DMA 即 DMA 控制器，提供了一种硬件的数据传输方式，无需 CPU 的介入，可以处理外设和存储器之间或者存储器和存储器之间的传输数据。因无 CPU 介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 8 通道，每个通道都可以处理一个或多个外设的存储器访问请求。DMA 控制器内部包含了仲裁器，用来仲裁多个 DMA 请求的优先级。

9.2. 主要特性

- 两个 DMA 控制器，各支持 8 个 DMA 通道，每个通道都可独立配置
- 支持外设到存储器、存储器到外设、存储器到存储器的数据传输
- 每个通道连接固定的硬件 DMA 请求
- 支持硬件优先级（通道号越低优先级越高）
- 支持源地址/目标地址递增或不变
- 每个通道有 16 字节的内部 FIFO
- 支持 8/16/32 位宽的传输
- 支持 Burst 功能
- 支持链表功能
- 支持 Big-endian 和 Little-endian 的字节存储次序配置
- 支持 DMA 中断功能

9.3. 结构框图

图 9-1 结构框图



9.4. 使用说明

9.4.1. 外设请求选择

DMA1 和 DMA2 各自有 64 个外设请求,每个 DMA 内部的通道共享此 64 个外设请求。

表 9- 1 目标外设和源外设请求号

请求号	DMA1 请求源	DMA2 请求源	请求号	DMA1 请求源	DMA2 请求源
REQ 0	ADC1	ADC1	REQ 32	TIM15_CH1	TIM16_CH1
REQ 1	SPI1_TX	SPI1_TX	REQ 33	TIM15_CH2	TIM16_UP
REQ 2	SPI1_RX	SPI1_RX	REQ 34	TIM15_UP	TIM17_CH1
REQ 3	SPI2_TX	SPI2_TX	REQ 35	TIM15_TRIG	TIM17_UP
REQ 4	SPI2_RX	SPI2_RX	REQ 36	TIM15_COM	TIM6_UP
REQ 5	UART1_TX	UART1_TX	REQ 37	I2S1_TX	I2S1_TX
REQ 6	UART1_RX	UART1_RX	REQ 38	I2S1_RX	I2S1_RX
REQ 7	UART2_TX	UART2_TX	REQ 39	DAC1_CH1	DAC1_CH1
REQ 8	UART2_RX	UART2_RX	REQ 40	DAC1_CH2	DAC1_CH2

REQ 9	I2C1_TX	I2C1_TX	REQ 41	I2S2_TX	I2S2_TX
REQ10	I2C1_RX	I2C1_RX	REQ 42	I2S2_RX	I2S2_RX
REQ11	I2C2_TX	I2C2_TX	REQ 43		
REQ12	I2C2_RX	I2C2_RX	REQ 44		
REQ13	TIM1_CH1	TIM8_CH1	REQ 45	UART4_TX	UART4_TX
REQ14	TIM1_CH2	TIM8_CH2	REQ 46	UART4_RX	UART4_RX
REQ15	TIM1_CH3	TIM8_CH3	REQ 47	SPI3_TX	SPI3_TX
REQ16	TIM1_CH4	TIM8_CH4	REQ 48	SPI3_RX	SPI3_RX
REQ17	TIM1_UP	TIM8_UP	REQ 49	TIM4_CH1	TIM7_UP
REQ18	TIM1_TRIG	TIM8_TRIG	REQ 50	TIM4_CH2	
REQ19	TIM1_COM	TIM8_COM	REQ 51	TIM4_CH3	
REQ20	TIM2_CH1	TIM3_CH1	REQ 52	TIM4_CH4	
REQ21	TIM2_CH2	TIM3_CH2	REQ 53	TIM4_UP	
REQ22	TIM2_CH3	TIM3_CH3	REQ 54	TIM4_TRIG	
REQ23	TIM2_CH4	TIM3_CH4	REQ 55		
REQ24	TIM2_UP	TIM3_UP	REQ 56		
REQ25	TIM2_TRIG	TIM3_TRIG	REQ 57		
REQ26	ADC2	ADC2	REQ 58		
REQ27			REQ 59		
REQ28	UART3_TX	UART3_TX	REQ 60		
REQ29	UART3_RX	UART3_RX	REQ 61		
REQ30	LPUART_TX	LPUART_TX	REQ 62		
REQ31	LPUART_RX	LPUART_RX	REQ 63		

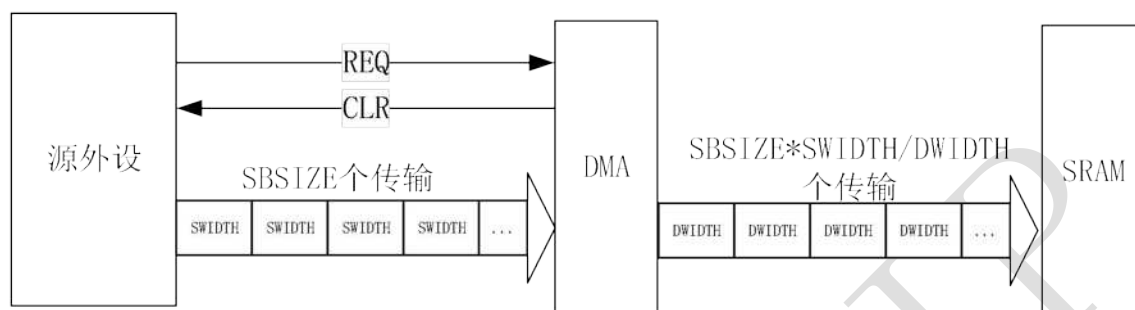
9.4.2. 传输模式

● 外设到存储器（P2M）

外设到存储器模式中，源外设准备好数据以后，会向 DMA 发送一个高电平的 REQ 请求信号。DMA 在 AHB 总线空闲并且此通道具有最高优先级的请求时，向该外设发送 CLR 信号，通知源外设准备数据传输。DMA 首先从源外设中取出 SBSIZE 个 SWIDTH 位宽的数据，然后将取出的数据拆分为 $(SBSIZE * SWIDTH / DWIDTH)$ 个 DWIDTH 位

宽的数据写入到 SRAM 中。总源传输大小 SrcSize 减少 SBSIZE, 总目标传输大小 DestSize 减少 $(SBSIZE * SWIDTH / DWIDTH)$, 其中 SrcSize 与 DestSize 的关系是固定的, $DestSize = (SWIDTH / DWIDTH) * SrcSize$ 。

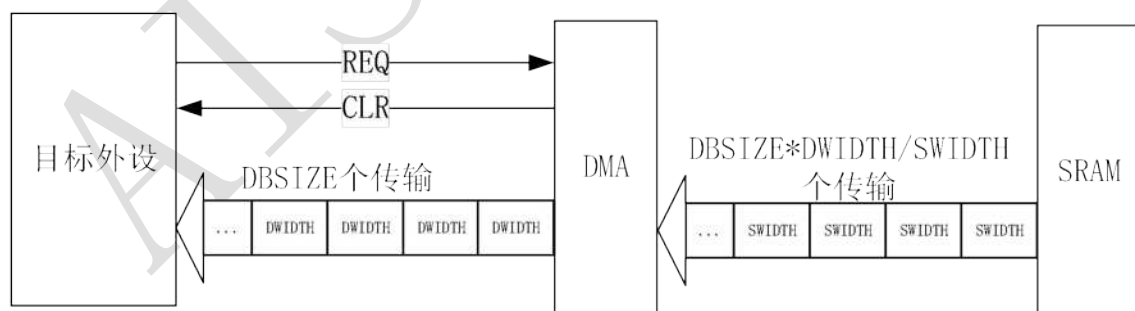
图 9-2 外设到存储器



● 存储器到外设 (M2P)

存储器到外设模式中, 目标外设准备好接受数据以后, 会向 DMA 发送一个高电平的 REQ 请求信号。DMA 在 AHB 总线空闲并且此通道具有最高优先级的请求时, 向该外设发送 CLR 信号, 通知目标外设准备数据传输。DMA 首先从 SRAM 中取出 $(DBSIZE * DWIDTH / SWIDTH)$ 个 SWIDTH 位宽的数据, 然后将取出的数据拆分为 DBSIZE 个 DWIDTH 位宽的数据写入到目标外设中。总源传输大小 SrcSize 减少 SBSIZE, 总目标传输大小 DestSize 减少 $(SBSIZE * SWIDTH / DWIDTH)$, 其中 SrcSize 与 DestSize 的关系是固定的, $DestSize = (SWIDTH / DWIDTH) * SrcSize$ 。

图 9-3 存储器到外设

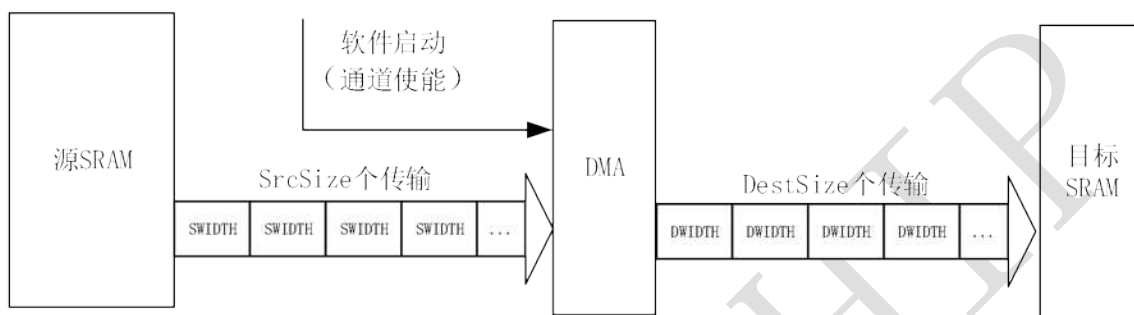


● 存储器到存储器 (M2M)

存储器到存储器模式中, 一旦软件使能通道以后, DMA 在 AHB 总线空闲并且此通道具有最高优先级的请求时, 开始搬运数据。DMA 首先从源 SRAM 中取出 SBSIZE 个 SWIDTH 位宽的数据, 然后将取出的数据拆分 DBSIZE 个 DWIDTH 位宽的数据写入到

目标外设中。软件启动一次以后，由硬件一直重复上述搬运过程，每重复一次上述搬运过程，DestSize 减少 DBSIZE，SrcSize 减少 SBSIZE，直到 DestSize 减少到 0 为止（传输完成），其中 SrcSize 与 DestSize 的关系是固定的， $\text{DestSize} = (\text{SWIDTH}/\text{DWIDTH}) * \text{SrcSize}$ 。

图 9-4 存储器到存储器



9.4.3. DMA 优先级

DMA 优先级是固定的，通道 0 为最高优先级，通道 7 为最低优先级。如果 DMA 正在进行一个低优先级通道的传输数据，此时一个高优先级通道的请求发生，DMA 将会等待低优先级通道的数据传输完毕，之后再处理高优先级通道的请求。

9.4.4. 软件注意事项

1. 通道使能后，不能对通道寄存器进行写操作。配置通道，必须在通道关闭的状态下进行。
2. 源/目标 burst size 必须是自身传输 size 的整数倍。
3. 源位宽乘以传输 Size 必须是目标位宽的整数倍。
4. 存储器位宽与外设位宽必须保持一致。
5. 如果通过设置 DMA_CHCONFIG 寄存器的 EN 位 (bit 0) 关闭通道，必须查询通道状态寄存器对应通道状态位为 0 后才能再次使能通道，因为禁止通道的过程有延时。
6. 如果 DMA 是流控制器，而且传输 Size 为 0，那么通道使能后不会产生任何数据传输。

9.4.5. 启动 DMA 传输流程

配置 DMA 通道 x 时应遵守下面的顺序：

1. 使能 DMA。设置 DMA_CONFIG 的 EN 位，使能 DMA。
2. 关闭通道 x。
3. 配置通道 x 传输参数，包括：传输类型、源/目标外设 ID、源/目标传输位宽、源/目标 burst size、源/目标地址递增使能/禁止、原始中断使能/禁止。
4. 配置通道 x 中断使能/禁止，包括半传输中断、传输完成中断、传输错误中断。
5. 配置通道 x 源/目标地址。
6. 配置通道 x 传输长度。
7. 使能通道，启动 DMA 传输。

9.4.6. 关闭 DMA 通道

关闭 DMA 通道有两种方法：

1. 直接清零 DMA_CHCONFIG 的 EN 位（bit0）。这种方法会丢失 FIFO 中的数据。
2. 使能 DMA_CHCONFIG 的 HALT 位（bit8），等待 DMA_CHCONFIG 的 ACTIVE 位（bit7）清 0，DMA_CHCONFIG 的 EN 位（bit0）清 0。这种方法不会丢失 FIFO 中的数据。

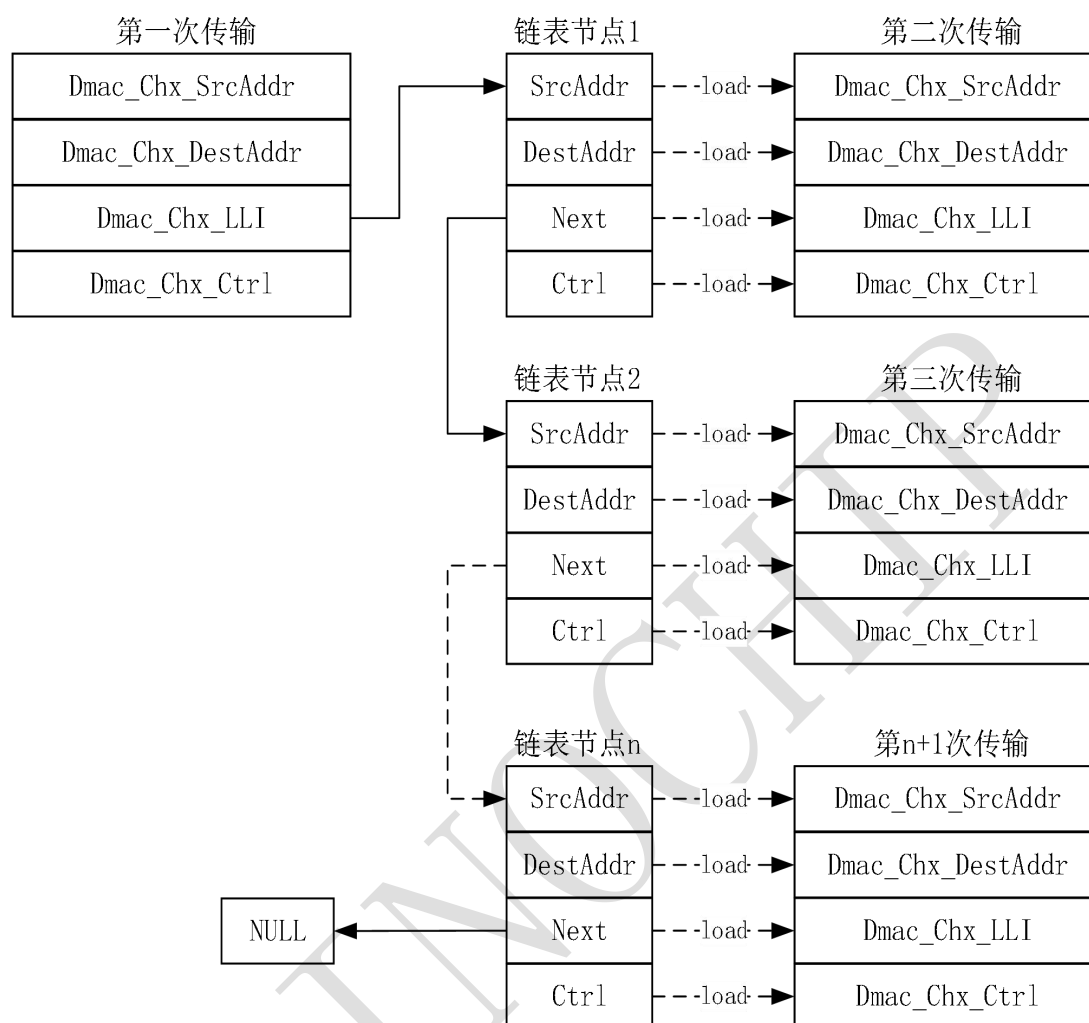
9.4.7. DMA 链表使用流程

一个 DMA 链表由一个或多个链表节点组成。每个链表节点都是一个链表结构体。一个链表节点（链表结构体）由四个数据组成，这些数据按照以下顺序来排列：

1. SrcAddr
2. DestAddr
3. Next
4. Ctrl

这四个数据分别对应 DMA 寄存器 DMA_CHSRCADDR、DMA_CHDESTADDR、DMA_CHLLI、DMA_CHCTRL，每个 bit 的意义都一样。

图 9-5 DMA 链表



每个链表节点的 Next 指向下一个链表节点的地址，最后一个链表节点的 Next 指向 NULL。如果最后一个链表节点的 Next 指向第一个链表节点，则构成一个单向循环链表。当只有一个链表结构体，且 Next 指向自身，则构成一个自循环链表。普通 DMA 传输因其 DMA_CHLLI 指向 NULL，其实就是一个单节点链表的传输。

实现步骤：

1. 每个链表节点设置参数，包括原始中断使能、源/目标地址、源/目标传输位宽、源/目标 burst size、源/目标地址递增使能、传输长度。
2. 将所有链表节点通过 Next 链接，最后一个链表节点的 Next 为 NULL。
3. 将链表首节点的四个数据复制给通道 x 的相应寄存器。
4. 使能通道 x，启动链表模式 DMA 传输。
5. 当 DMA 传输完成后，硬件自动将 DMA_CHLLI 指向节点的四个数据复制到

DMA 寄存中进行传输。一直到 DMA_CHLLI 指向 NULL 时 DMA 传输才停止。

设计技巧：

用户使能每个链表节点的原始中断，同时使能 DMA 传输完成中断，则每个链表节点传输完成后都会产生一个传输完成中断。

用户使能最后一个链表节点的原始中断，禁止其他链表节点的原始中断。同时使能 DMA 传输完成中断，则只有当所有链表节点的传输全部完成后，才会产生一个传输完成中断。

任何时候，只要使能传输错误中断，当传输错误发生时，都会立刻产生传输错误中断。

9.5. 寄存器描述

DMA 寄存器基地址：

DMA1: 0x4003_1000

DMA2: 0x4003_2000

偏置	名称	描述
0x00	DMA_INTSTATUS	中断状态寄存器
0x04	DMA_INTTCSTATUS	传输完成中断寄存器
0x08	DMA_INTTCCLR	传输完成中断清除寄存器
0x0C	DMA_INTERRSTATUS	传输错误中断寄存器
0x10	DMA_INTERRCLR	传输错误中断清除寄存器
0x14	DMA_RAWINTTCSTATUS	传输完成原始中断寄存器
0x18	DMA_RAWINTERRSTATUS	传输错误原始中断寄存器
0x1C	DMA_ENCHSTATUS	通道使能状态寄存器
0x30	DMA_CONFIG	DMA 配置寄存器
0x100, 0x120, 0x140, 0x160, 0x180, 0x1A0, 0x1C0, 0x1E0	DMA_CHSRCADDR	通道 x 源地址寄存器
0x104, 0x124, 0x144, 0x164, 0x184, 0x1A4, 0x1C4, 0x1E4	DMA_CHDESTADDR	通道 x 目标地址寄存器
0x108, 0x128, 0x148, 0x168,	DMA_CHLLI	通道 x 链接表寄存器

0x188,0x1A8,0x1C8,0x1E8		
0x10C, 0x12C, 0x14C, 0x16C, 0x18C,0x1AC,0x1CC,0x1EC	DMA_CHCTRL	通道 x 控制寄存器
0x110, 0x130h, 0x150, 0x170, 0x190 ,0x1B0,0x1D0,0x1F0	DMA_CHCONFIG	通道 x 配置寄存器

9.5.1. 中断状态寄存器 DMA_INTSTATUS（偏移：00h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	INTSTATUS	RO	0x00	通道 x 中断汇总。 0: 通道 x 上无中断 1: 通道 x 上发生中断 该位指示以下位的逻辑或运算结果： 半传输中断 传输完成中断 传输错误中断

9.5.2. 传输完成中断状态寄存器 DMA_INTTCSTATUS（偏移：04h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	HFTC	RO	0x00	通道 x 半传输中断状态 0: 通道 x 上无半传输中断 1: 通道 x 上发生半传输中断 注：如果禁止了该中断，可以查询半传输原始中断状态。
7:0	TC	RO	0x00	通道 x 传输完成中断状态

				0: 通道 x 上无传输完成中断 1: 通道 x 上发生传输完成中断 注: 如果禁止了该中断, 可以查询传输完成原始中断状态。

9.5.3. 传输完成中断清除寄存器 DMA_INTTCCLR (偏移: 08h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	HFTC	WO	0x00	通道 x 半传输中断清除 0: 无效操作 1: 清除通道 x 半传输中断标志位
7:0	TC	WO	0x00	通道 x 传输完成中断清除 0: 无效操作 1: 清除通道 x 传输完成中断标志位

9.5.4. 传输错误中断寄存器 DMA_INTERRSTATUS (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	ERR	RO	0x00	通道 x 传输错误中断状态 0: 通道 x 上无传输错误中断 1: 通道 x 上发生传输错误中断 注: 如果禁止了该中断, 可以查询传输错误原始中断状态。

9.5.5. 传输错误中断清除寄存器 DMA_INTERRCLR (偏移: 10h)

比特	名称	属性	复位值	描述

31:8	RSV	-	-	保留
7:0	ERR	WO	0x00	通道 x 传输错误中断状态清除。 0: 无效操作 1: 清除通道 x 传输错误状态标志位

9.5.6. 传输完成原始中断寄存器 DMA_RAWINTTCSTATUS (偏移: 14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	HFTC	RO	0x00	通道 x 半传输原始中断状态。 0: 通道 x 上无半传输原始中断 1: 通道 x 上发生半传输原始中断
7:0	TC	RO	0x00	通道 x 传输完成原始中断状态。 0: 通道 x 上无传输完成原始中断 1: 通道 x 上发生传输完成原始中断

9.5.7. 传输错误原始中断寄存器 DMA_RAWINTERRSTATUS (偏移: 18h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	ERR	RO	0x00	通道 x 传输错误原始中断状态。 0: 通道 x 上无传输错误原始中断 1: 通道 x 上发生传输错误原始中断

9.5.8. 通道使能状态寄存器 DMA_ENCHSTATUS (偏移: 1Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	ENCHSTAT	RO	0x00	通道 x 使能状态

				0: 通道 x 未使能 1: 通道 x 已使能

9.5.9. DMA 配置寄存器 DMA_CONFIG（偏移：30h）

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1	ENDIAN	RW	0	字节存储次序配置 1: Big-Endian 模式 0: Little-Endian 模式
0	EN	RW	0	DMA 使能 0: 禁止 DMA 1: 使能 DMA

9.5.10. 通道 x 源地址寄存器 DMA_CHSRCADDR（偏移：100h, 120h, 140h, 160h, 180h, 1A0h, 1C0h, 1E0h）

比特	名称	属性	复位值	描述
31:0	SRCADDR	RW	0x00000000	通道 x 源地址

注：通道使能时，禁止更改该寄存器的值。

9.5.11. 通道 x 目标地址寄存器 DMA_CHDESTADDR（偏移：104h, 124h, 144h, 164h, 184h, 1A4h, 1C4h, 1E4h）

比特	名称	属性	复位值	描述
31:0	DESTADDR	RW	0x00000000	通道 x 目标地址

注：通道使能时，禁止更改该寄存器的值。

9.5.12. 通道 x 链接表寄存器 DMA_CHLLI（偏移：108h, 128h, 148h, 168h, 188h, 1A8h, 1C8h, 1E8h）

比特	名称	属性	复位值	描述
31:2	LLI	RW	0x00000000	通道 x 链接表，表示下一个 32 位的链接表地址 Addr[31:2]，其中 Addr[1:0]为 0。
1:0	RSV	-	-	保留

9.5.13. 通道 x 控制寄存器 DMA_CHCTRL（偏移：10Ch, 12Ch, 14Ch, 16Ch, 18Ch, 1ACh, 1CCh, 1ECh）

比特	名称	属性	复位值	描述
31	ITC	RW	0	传输原始中断使能 0：禁止传输原始中断 1：使能传输原始中断 注：原始中断不会产生真正的中断，而是传输完成中断、半传输中断的前提。
30	RSV	-	-	保留
29	DI	RW	0	目标地址递增使能 0：禁止目标地址递增 1：使能目标地址递增
28	SI	RW	0	源地址递增使能 0：禁止源地址递增 1：使能源地址递增
27:25	DWIDTH	RW	000	目标传输位宽 000：字节（8bit） 001：半字（16bit） 010：字（32bit）

				其它：保留 注：源传输位宽和目标传输位宽可以不同，硬件会自动根据各自的传输位宽对数据打包、解包。
24:22	SWIDTH	RW	000	源传输位宽 000：字节（8bit） 001：半字（16bit） 010：字（32bit） 其它：保留 注：源传输位宽和目标传输位宽可以不同，硬件会自动根据各自的传输位宽对数据打包、解包。
21:19	DBSIZE	RW	000	目标 burst size 000：1 001：4 010：8 011：16 100：32 101：64 110：128 111：256 一个 burst 由 DBSIZE 个 DWIDTH 组成。
18:16	SBSIZE	RW	000	源 burst size 000：1 001：4 010：8 011：16 100：32 101：64 110：128 111：256

				一个 burst 由 SBSIZE 个 SWIDTH 组成。
15:0	SIZE	RW	0x0000	传输 Size 写 SIZE 表示一次 DMA 传输数据的长度，单位是源位宽。 读 SIZE 得到的是 DMA 未传输的数据长度，单位是目标位宽。一次 DMA 传输完成后，这些位由硬件自动清零。

注：通道控制寄存器的设置必须在通道使能关闭时进行。通道一旦使能，设置立即生效。通道使能时，禁止更改通道 x 控制寄存器。

9.5.14. 通道 x 配置寄存器 DMA_CHCONFIG（偏移：110h, 130h, 150h, 170h, 190h, 1B0h, 1D0h, 1F0h）

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:24	SRCPERIPH	RW	0x00	源外设 ID 当源外设为存储器时，该值无效。
23:22	RSV	-	-	保留
21:16	DESTPERIPH	RW	0x00	目标外设 ID。 当目标外设为存储器时，该值无效。
15:9	RSV	-	-	保留
8	HALT	RW	0	中止当前 DMA 传输 0：正常 DMA 传输 1：中止 DMA 传输。DMA 传输中止后，通道 FIFO 中的内容自动清除。
7	ACTIVE	RO	0	通道 FIFO 中数据状态 0：通道 FIFO 中没有数据 1：通道 FIFO 中有数据

6	IHFTC	RW	0	半传输中断使能 0: 禁止半传输中断 1: 使能半传输中断
5	ITC	RW	0	传输完成中断使能 0: 禁止传输完成中断 1: 使能传输完成中断
4	IE	RW	0	传输错误中断使能 0: 禁止传输错误中断 1: 使能传输错误中断
3:1	FLOWCTRL	RW	000	流控制器和传输类型 000: 存储器到存储器 001: 存储器到外设 010: 外设到存储器
0	EN	RW	0	通道使能 0: 禁止通道。通道使能的情况下，该位清 0，禁止通道。但禁止通道有一定延迟，查询通道使能状态直到通道真正关闭。 1: 使能通道。该位置 1 后，传输立即生效，传输完成后，该位由硬件自动清零。 通道使能状态，参考通道使能状态寄存器。

10. 高级定时器（TIM1/TIM8）

10.1. 概述

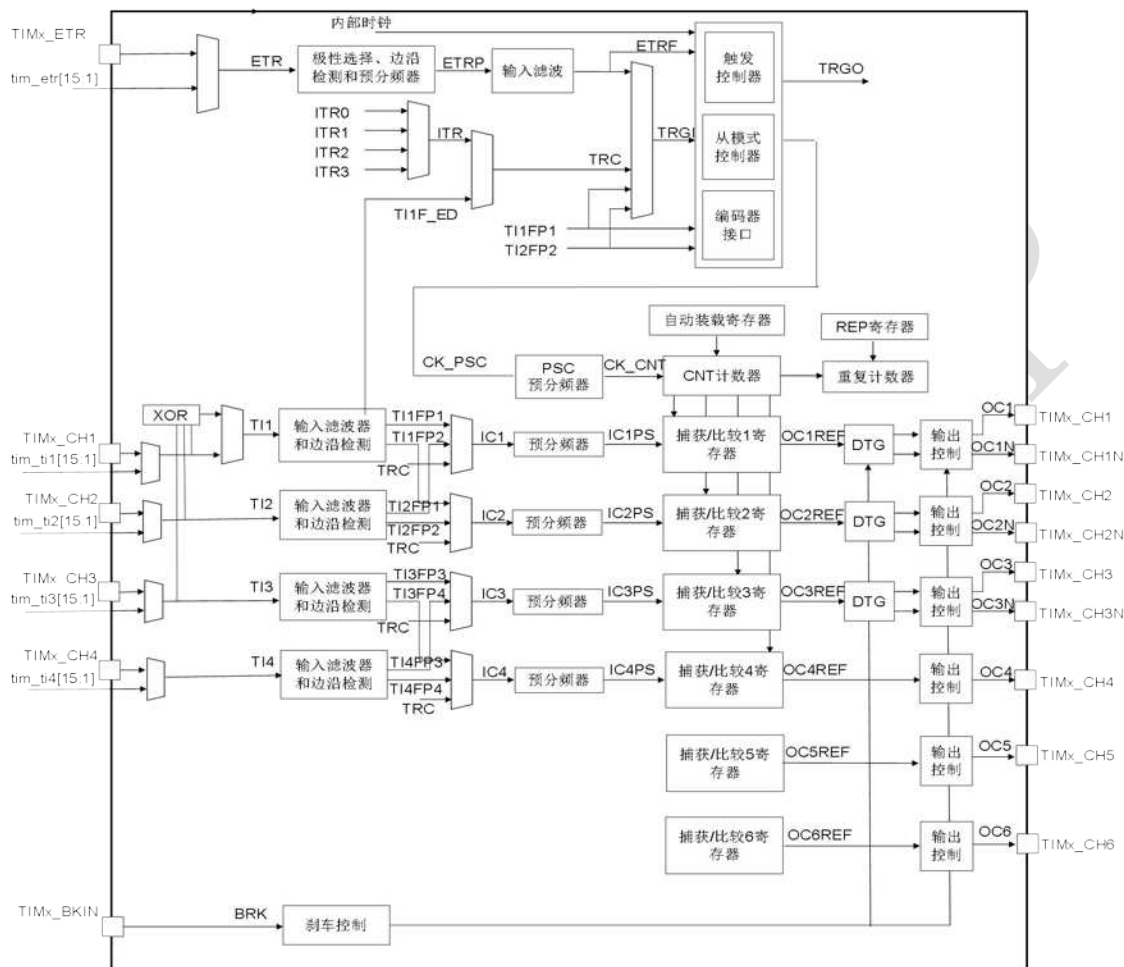
高级控制定时器由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。使用定时器预分频器和系统时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。高级控制定时器和通用定时器是完全独立的，它们不共享任何资源，但它们可以同步操作。

10.2. 主要特性

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值
- 多达 4 个独立通道
 - 输入捕获
 - 输出比较
 - PWM 生成（边沿或中间对齐模式）
 - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA：
 - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
 - 触发事件：(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
 - 刹车信号输入

- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

图 10-1 高级定时器结构框



10.3. 功能描述

10.3.1. 定时器基本单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行。定时器基本单元包含：

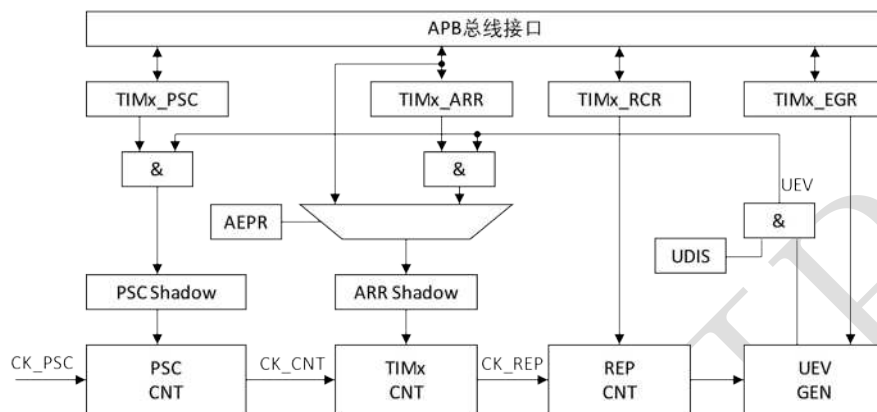
计数器寄存器(TIMx_CNT)

自动装载寄存器 (TIMx_ARR)

预分频器寄存器 (TIMx_PSC)

重复计数寄存器 (TIMx_RCR)

图 10-2 计数单元的结构



如图所示，自动重载寄存器（ARR）是可预装载的。当在 TIMx_CR1 寄存器中的 ARR 预装载使能位未使能时 (ARPE=0)，写入 ARR 的内容被立即传送到 ARR 影子寄存器（ARR Shadow），而当 ARPE=1 时，写入 ARR 的内容在每次的更新事件 UEV 时才被传送到影子寄存器。当计数器达到溢出条件并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。

注意，在设置了 TIMx_CR1 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

10.3.2. 定时器计数模式

定时器支持三种计数模式：

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (TIMx_ARR 计数器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件，如果 UDIS=0，就会产生一次更新事件。更新事件也可以由 UG bit 置位产生（通过 EGR 寄存器软件置位或通过硬件从模式方式）。当更新事件产生后，如果 TIMx_CR1 的 URS=0，并且更新中断或 DMA 已使能，就会产生更新中断或更新 DMA 请求；如果 URS=1，则只有溢出事件产生的更新事件才产生更新中断和更新 DMA 请求。

向下计数模式

在向下模式中，计数器从自动装入的值(TIMx_ARR 计数器的值)开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件，如果 UDIS=0，就会产生一次更新事件。更新事件也可以由 UG bit 置位产生（通过 EGR 寄存器软件置位或通过硬件从模式方式）。当更新事件产生后，如果 TIMx_CR1 的 URS=0，并且更新中断或 DMA 已使能，就会产生更新中断或更新 DMA 请求；如果 URS=1，则只有溢出事件产生的更新事件才产生更新中断和更新 DMA 请求。

中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数时，定时器模块在计数器计数到自动加载值减 1 产生一个上溢事件；向下计数时，定时器模块在计数器计数到 1 时产生一个下溢事件。无论上溢还是下溢，当 UDIS=0 时，都会产生一次更新事件。当更新事件产生后，如果 TIMx_CR1 的 URS=0，并且更新中断或 DMA 已使能，就会产生更新中断或更新 DMA 请求；如果 URS=1，则只有溢出事件产生的更新事件才产生更新中断和更新 DMA 请求。

在中央计数模式中，TIMx_CR1 寄存器中的计数方向控制位 DIR 为只读，指示计数方向。计数方向被硬件自动更新。

10.3.3. 预分频器

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个(在 TIMx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

如下两图给出了在预分频器运行时，更改计数器参数的例子。

图 10-3 当预分频器的参数从 1 变到 2 时，计数器的时序图

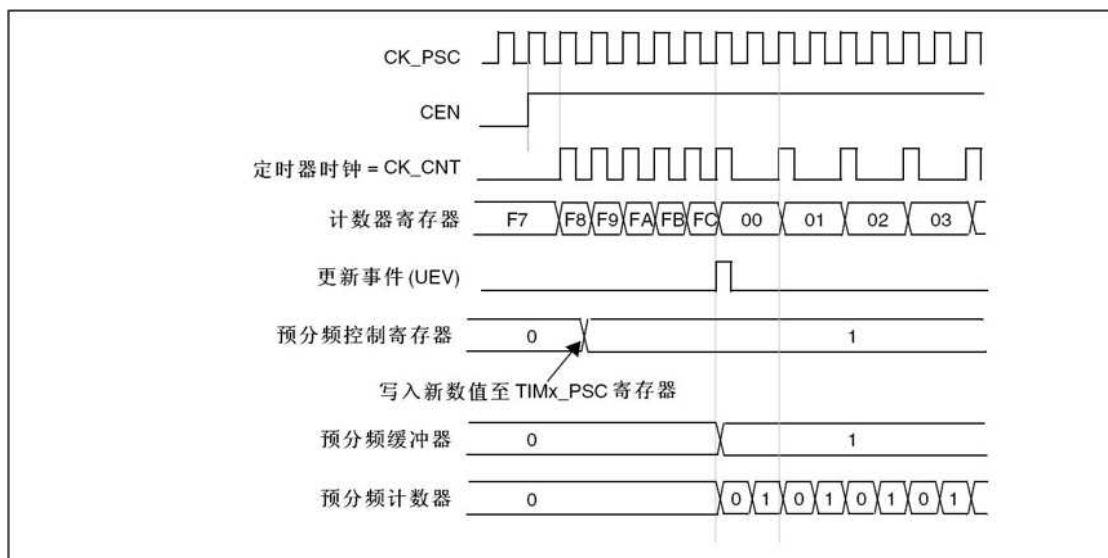
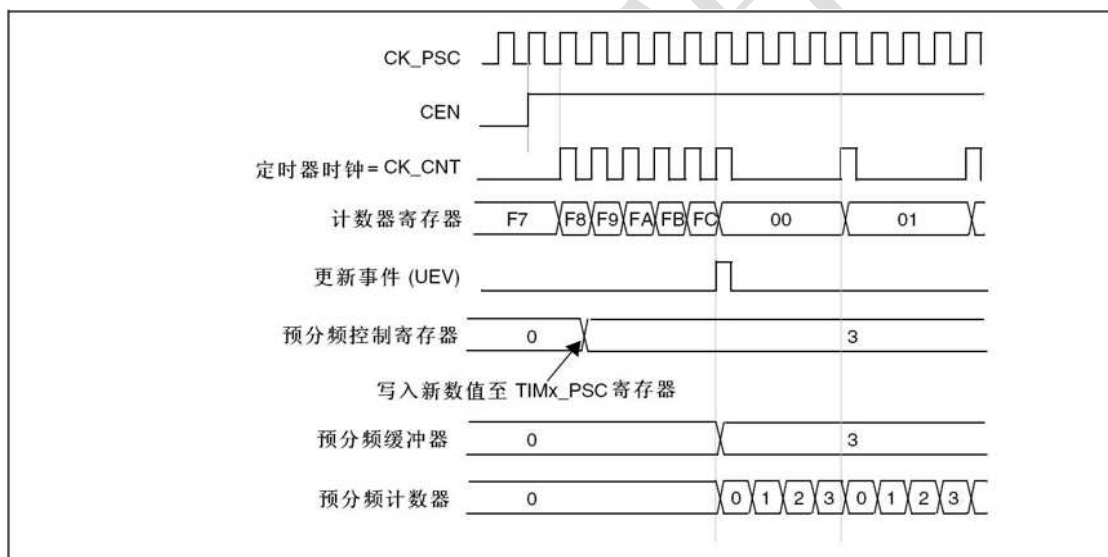


图 10-4 当预分频器的参数从 1 变到 4 时，计数器的时序图



10.3.4. 重复计数器

重复计数器是一个 8 位的递减计数器，更新事件 UEV 只能在重复计数器计数达到 0 时产生。重复计数器在下述任一条件成立时递减：

向上计数模式下每次计数器溢出时，

向下计数模式下每次计数器下溢时，

中央对齐模式下每次上溢和每次下溢时。

重复计数器是自动加载的，重复速率是由 TIMx_RCR 寄存器的值。当更新事件由软件产生(通过设置 TIMx_EGR 中的 UG 位)或者通过硬件的从模式控制器产生，则无论

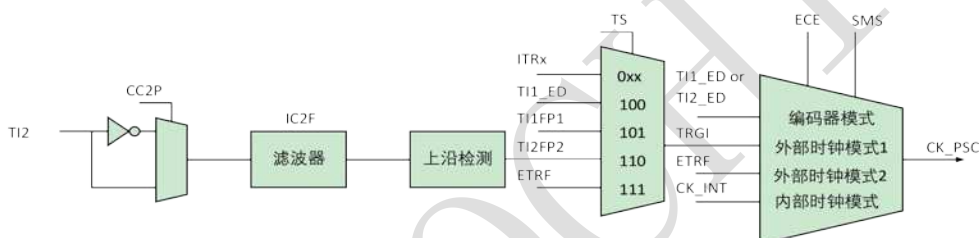
重复计数器的值是多少，TIMx_RCR 寄存器中的内容被重载入到重复计数器。

10.3.5. 时钟源选择

计数器时钟可由下列时钟源提供：

- 内部时钟(CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

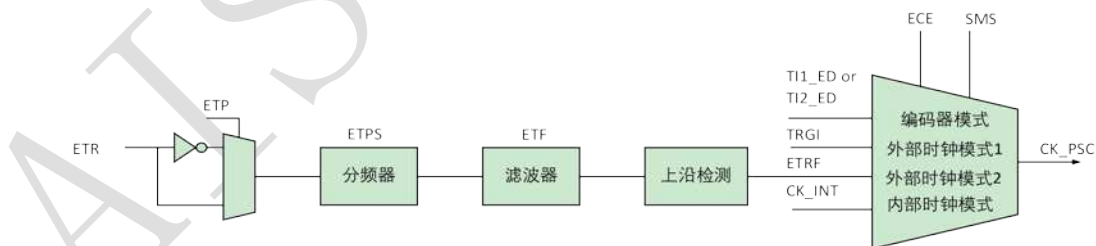
图 10-5 外部时钟模式 1



注：

- 1、TI1_ED：来自于 TI1 的双边沿有效信号，TI2_ED：来自于 TI2 的双边沿有效信号。
- 2、CC2P：控制有效信号的极性；IC2F：输入信号滤波。

图 10-6 外部时钟模式 2



注：

- 1、ETP：控制有效信号的极性；ETPS：对输入信号进行分频；ETF：对输入信号进行滤波。

内部时钟(CK_INT)：当禁止从模式（SMS=000，ECE=0）时，计数时钟就来自于 CK_INT，即来自于 PCLK。只要 CEN 位被写成 ‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

外部时钟模式 1: SMS=111, ECE=0, CK_PSC 由 TRGI 产生, TRGI 有多个信号源可选, 并由 TIMx_SMCR.TS 寄存器选择, 详情见寄存器说明。

外部时钟模式 2: ECE=1 或 SMS=111、TS=111, CK_PSC 来自 ETRF, 如上图所示。

编码器模式

编码器模式 1: SMS=001, ECE=0, CK_PSC 来自 TI1FP1 的上下沿

编码器模式 2: SMS=010, ECE=0, CK_PSC 来自 TI2FP2 的上下沿

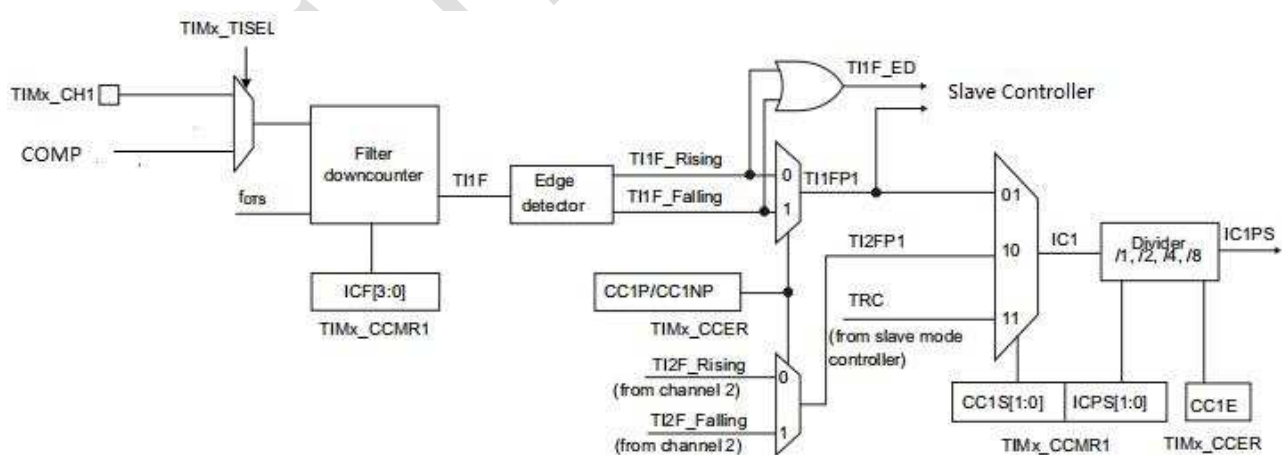
编码器模式 3: SMS=011, ECE=0, CK_PSC 来自 TI1FP1 和 TI2FP2 的上下沿

10.3.6. 捕获比较通道

输入捕获模式:

在输入捕获模式下, 当检测到 ICx 信号上相应的边沿后, 计数器的当前值被锁存到捕获/比较寄存器(TIMx_CCRx)中。当发生捕获事件时, 相应的 CCxIF 标志(TIMx_SR 寄存器)被置 1, 如果使能了中断或者 DMA 操作, 则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高, 那么重复捕获标志 CCxOF(TIMx_SR 寄存器)被置 1。写 CCxIF=0 可清除 CCxIF, 或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

图 10-7 通道 1 输入捕获 TI1 信号



以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中, 步骤如下:

- 1、由于是从 TI1 引脚输入, 因此需要将 TISEL bit0 写 0。
- 2、因为 TI1 捕获的值要写入到 TIMx_CCR1, 因此写入 TIMx_CCR1 寄存器中的

CC1S=01。因为通道被配置为输入，TIMx_CCR1 寄存器变为只读。

3、根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期，此例中我们可以以 F_{dts} 频率连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。

4、选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0(上升沿)。

5、配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIMx_CCMR1 寄存器的 IC1PS=00)。

6、设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。

如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位使能捕获中断请求，通过设置 TIMx_DIER 寄存器中的 CC1DE 位使能捕获 DMA 请求。

当发生一个输入捕获时：

产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。

CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。

如设置了 CC1IE 位，则会产生一个中断。

如设置了 CC1DE 位，则还会产生一个 DMA 请求。为了处理捕获溢出，建议在读捕获溢出标志之前读取数据，这是为了避免丢失在读捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

PWM 输入模式：

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

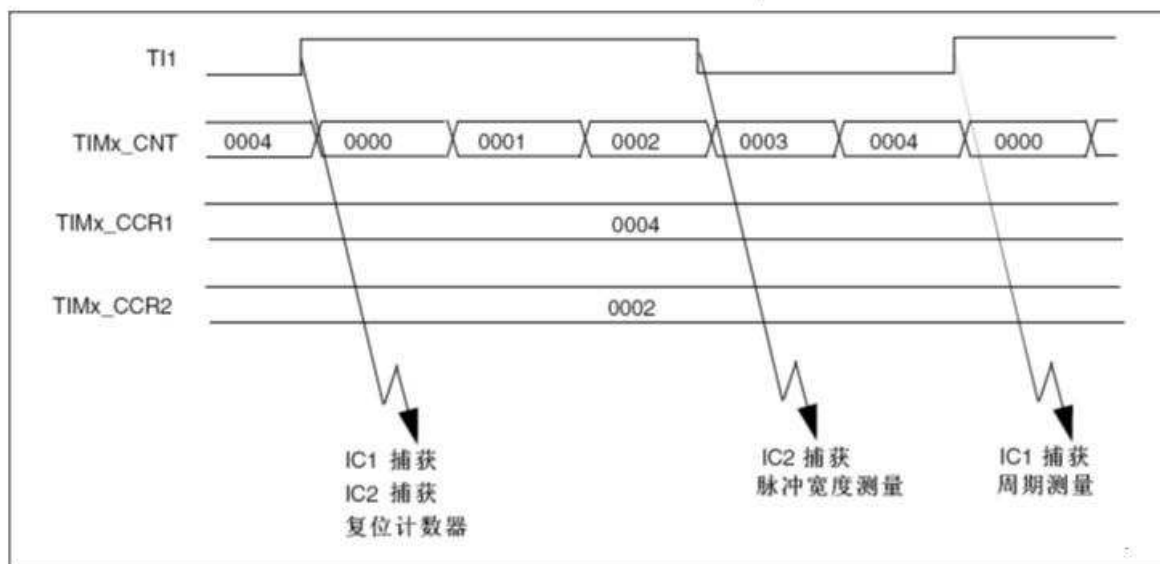
两个 ICx 信号被映射至同一个 TIx 输入。

这 2 个 ICx 信号为边沿有效，但是极性相反。

其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。例如，测量输入到 TI1 上的 PWM 信号的周期(TIMx_CCR1 寄存器)和占空比(TIMx_CCR2 寄存器)，具体步骤如下：

- 1、由于是从 TI1 引脚输入，因此需要将 TISEL bit0 写 0。
- 2、选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 3、选择 TI1FP1 的有效极性(用来捕获数据到 TIMx_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- 4、选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 5、选择 TI1FP2 的有效极性(捕获数据到 TIMx_CCR2)：置 CC2P=1(下降沿有效)。
- 6、选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 7、配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 8、使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

图 10-8 PWM 输入模式时序



输出比较模式：

此项功能是用来控制一个输出波形如 PWM，或者指示一段给定的时间已经到时。当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式(TIMx_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的 CCxIF 位)。

- 若设置了相应的中断屏蔽(TIMx_DIER 寄存器中的 CCxIE 位),则产生一个中断。
- 若设置了相应的使能位(TIMx_DIER 寄存器中的 CCxDE 位, TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能), 则产生一个 DMA 请求。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

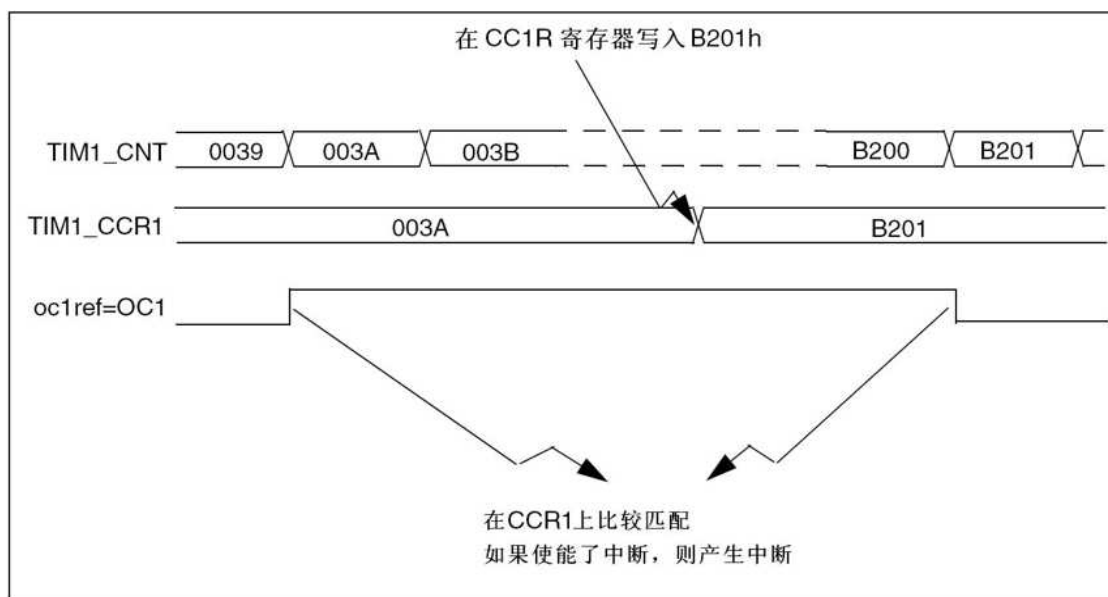
同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤:

- 1、选择计数器时钟(内部, 外部, 预分频器)。
- 2、将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
- 3、如果要产生一个中断请求, 设置 CCxIE 位。
- 4、选择输出模式, 例如:
 - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚, 设置 OCxM=011
 - 置 OCxPE = 0 禁用预装载寄存器
 - 置 CCxP = 0 选择极性为高电平有效
 - 置 CCxE = 1 使能输出。
- 5、设置 TIMx_CR1 寄存器的 CEN 位启动计数器。

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形, 条件是未使用预装载寄存器(OCxPE='0', 否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图是输出比较模式的一个例子。

图 10-9 输出比较模式，翻转 OC1



10.3.7. 强制输出模式

在输出模式(TIMx_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。配置 TIMx_CCMRx 寄存器中的 OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

置 TIMx_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。

10.3.8. PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入'110'(PWM 模式 1)或'111'(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE

位, (在向上计数或中心对称模式中)使能自动重载的预装载寄存器。

仅当发生一个更新事件的时候, 预装载寄存器才能被传送到影子寄存器, 因此在计数器开始计数之前, 必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置, 它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIMx_CCER 和 TIMx_BDTR 寄存器)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMx_CCER 和 BDTR 寄存器的描述。

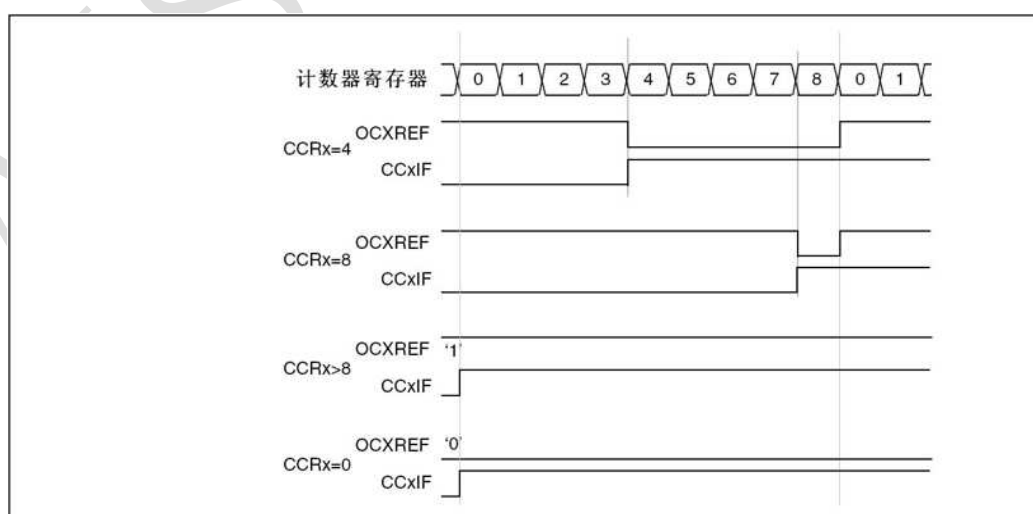
在 PWM 模式(模式 1 或模式 2)下, TIMx_CNT 和 TIMx_CCRx 始终在进行比较, (依据计数器的计数方向)以确定是否符合 $TIMx\_CCRx \leq TIMx\_CNT$ 或者 $TIMx\_CNT \leq TIMx\_CCRx$ 。根据 TIMx_CR1 寄存器中 CMS 位的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

PWM 边沿对齐模式

向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。下图是一个 PWM 模式 1 的例子。当 $TIMx\_CNT < TIMx\_CCRx$ 时, PWM 参考信号 OCxREF 为高, 否则为低。如果 TIMx_CCRx 中的比较值大于自动重载值(TIMx_ARR), 则 OCxREF 保持为'1'。如果比较值为 0, 则 OCxREF 保持为'0'。

图 10-10 边沿对齐的 PWM 波形 (ARR=8)



向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1, 当

TIMx_CNT>TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCxREF 保持为'1'。该模式下不能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为'00'时为中央对齐模式(所有保留的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。

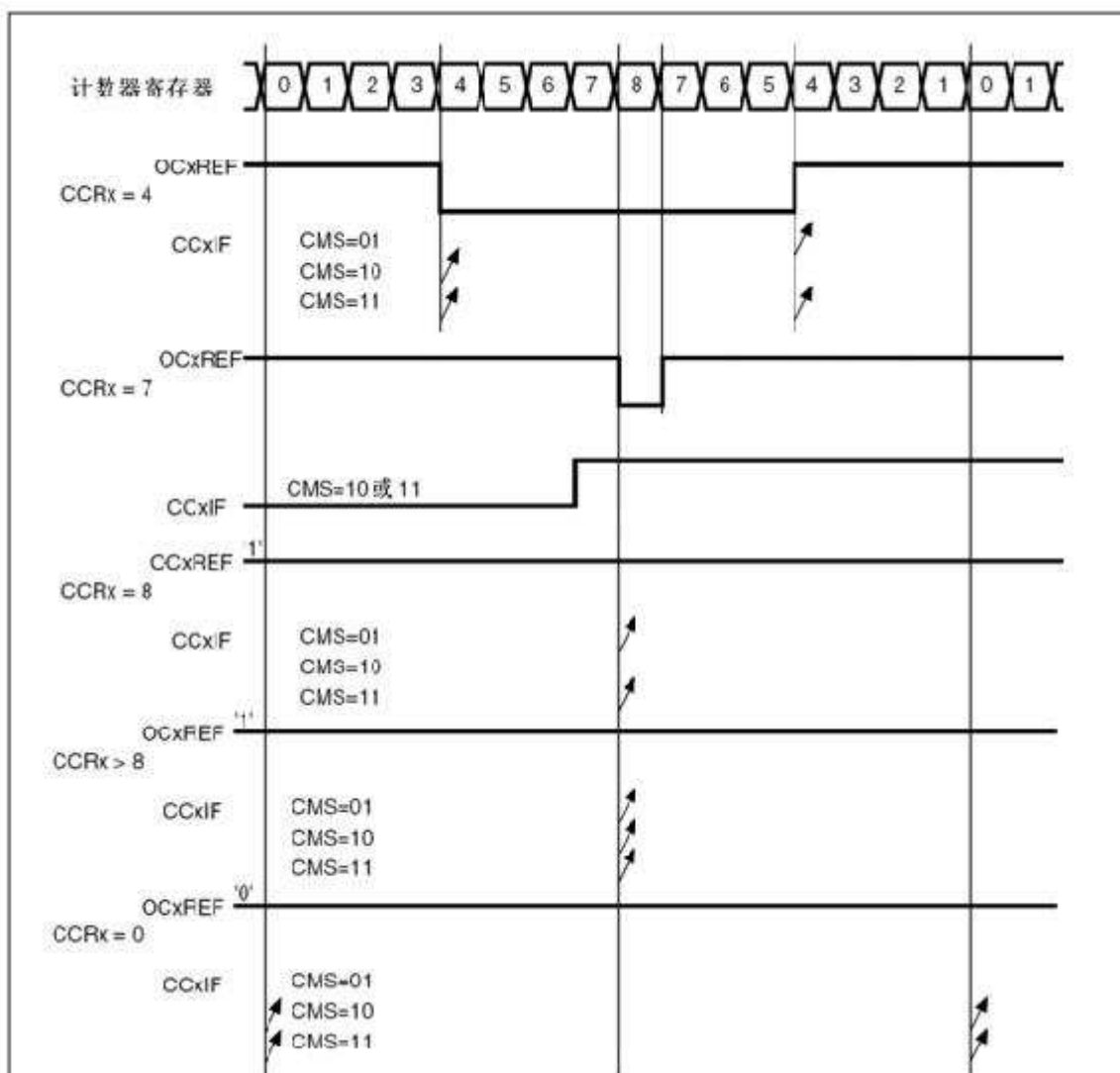
下图给出了一些中央对齐的 PWM 波形的例子

TIMx_ARR=8

PWM 模式 1

TIMx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

图 10-11 中央对齐的 PWM 波形 (ARR=8)



使用中央对齐模式的注意事项：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
 - 如果写入计数器的值大于自动重加载的值(TIMx_CNT>TIMx_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
 - 如果将 0 或者 TIMx_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设

置 TIMx_EGR 位中的 UG 位), 并且不要在计数进行过程中修改计数器的值。

AISINOCHIP

10.3.9. 互补输出和死区插入

高级控制定时器能够输出两路互补信号,并且能够管理输出的瞬时关断和接通。这段时间通常被称为死区,用户应该根据连接的输出器件和它们的特性(电平转换的延时、电源开关的延时等)来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位,可以为每一个输出独立地选择极性(主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制: TIMx_CCER 寄存器的 CCxE 和 CCxNE 位, TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位,带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是,在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区,如果存在刹车电路,则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效:

- OCx 输出信号与参考信号相同,只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反,只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度(OCx 或者 OCxN),则不会产生相应的脉冲。下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

图 10-12 带死区插入的互补输出

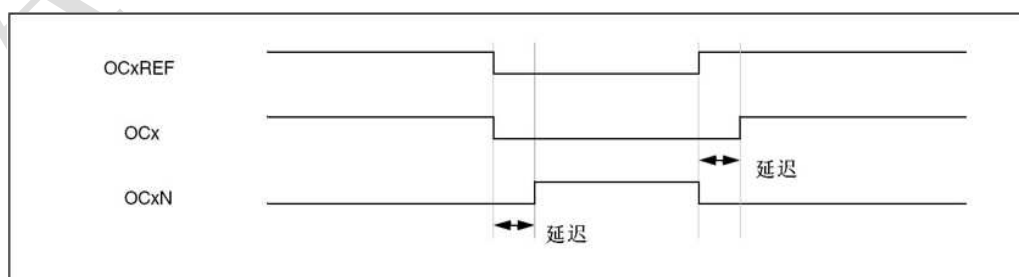


图 10-13 死区波形延迟大于负脉冲

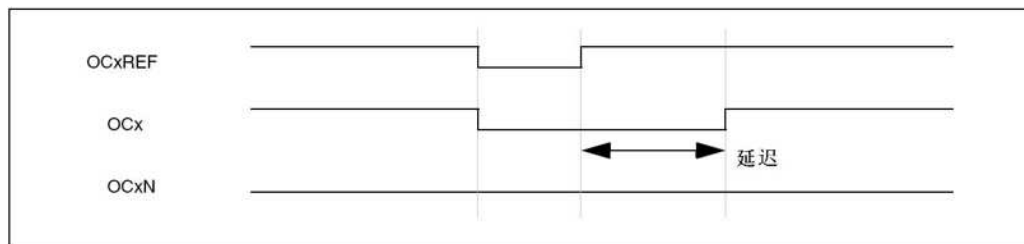
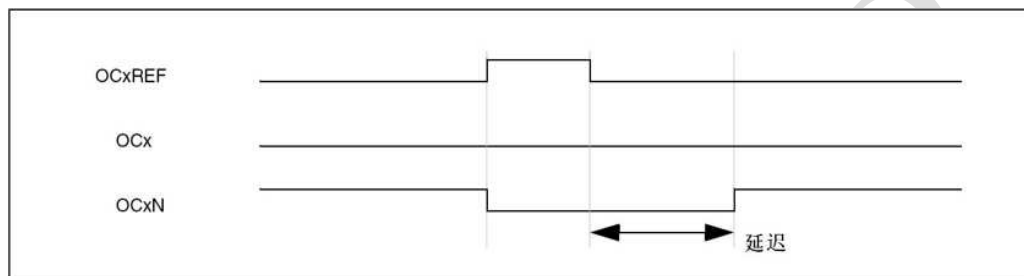


图 10-14 死区波形延迟大于正脉冲



每一个通道的死区延时都是相同的，是由 TIMx_BDTR 寄存器中的 DTG 位编程配置。

重定向 OCxREF 到 OCx 或 OCxN:

在输出模式下(强置、输出比较或 PWM)，通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形(例如 PWM 或者静态有效电平)。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注：当只使能 OCxN(CCxE=0, CCxNE=1)时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时(CCxE=CCxNE=1)，当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

10.3.10. 刹车功能

当使用刹车功能时，依据相应的控制位(TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OISx 和 OISxN 位)，输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。

刹车源既可以是刹车输入引脚又可以是一个系统异常事件 (System Break)。刹车

源的详情见定时器刹车输入互连关系。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置 TIMx_BDTR 寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 PCLK 时钟周期的延迟，因此需要等待一个 PCLK 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号(作用在输出端)和同步控制位(在 TIMx_BDTR 寄存器中)之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当 MOE=0 低时写 MOE=1，则读出它之前必须先插入一个延时(空指令)才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

1、MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSSR 位选择)。如果未开启对 break 输入信号的滤波，这个特性在 MCU 的振荡器关闭时依然有效。

2、一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSR=0，则定时器输出禁止，否则输出使能。

当使用互补输出时：

输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。

如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些(大约 2 个 CK_INT 的时钟周期)。

如果 OSSR=0，定时器输出禁止，否则保持使能输出；或一旦 CCxE 与 CCxNE 二者之一变高时，使能输出。

详情请参考互补输出通道 OCx 和 OCxN 的控制。

如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志(TIMx_SR 寄存器中的 BIF 位)为'1'时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 BDE 位，则

产生一个 DMA 请求。

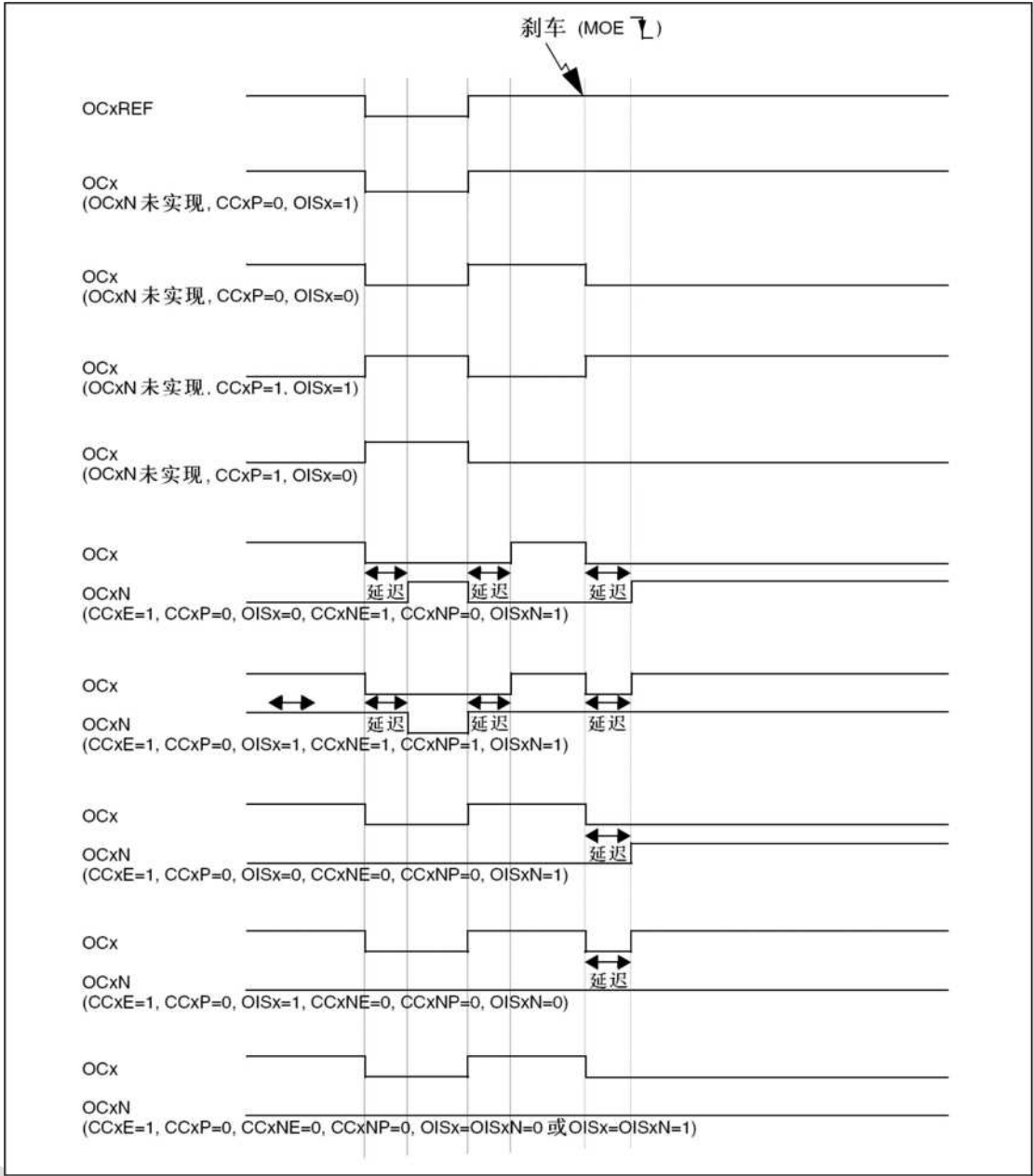
如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位。否则，MOE 始终保持低直到被再次置'1'。

注： 刹车输入为电平有效。所以，当刹车输入有效时，不能同时(自动地或者通过软件)设置 MOE。状态标志 BIF 不能被软件清除，只能等刹车输入无效后自动被硬件清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMx_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。用户可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看刹车和死区寄存器(TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

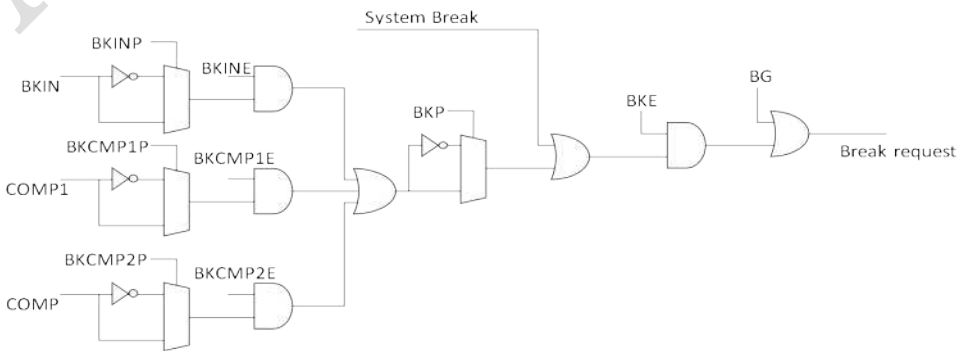
下图显示响应刹车的输出实例。

图 10-15 响应刹车的输出



下图显示了刹车输入信号以及极性选择。

图 10-16 刹车控制



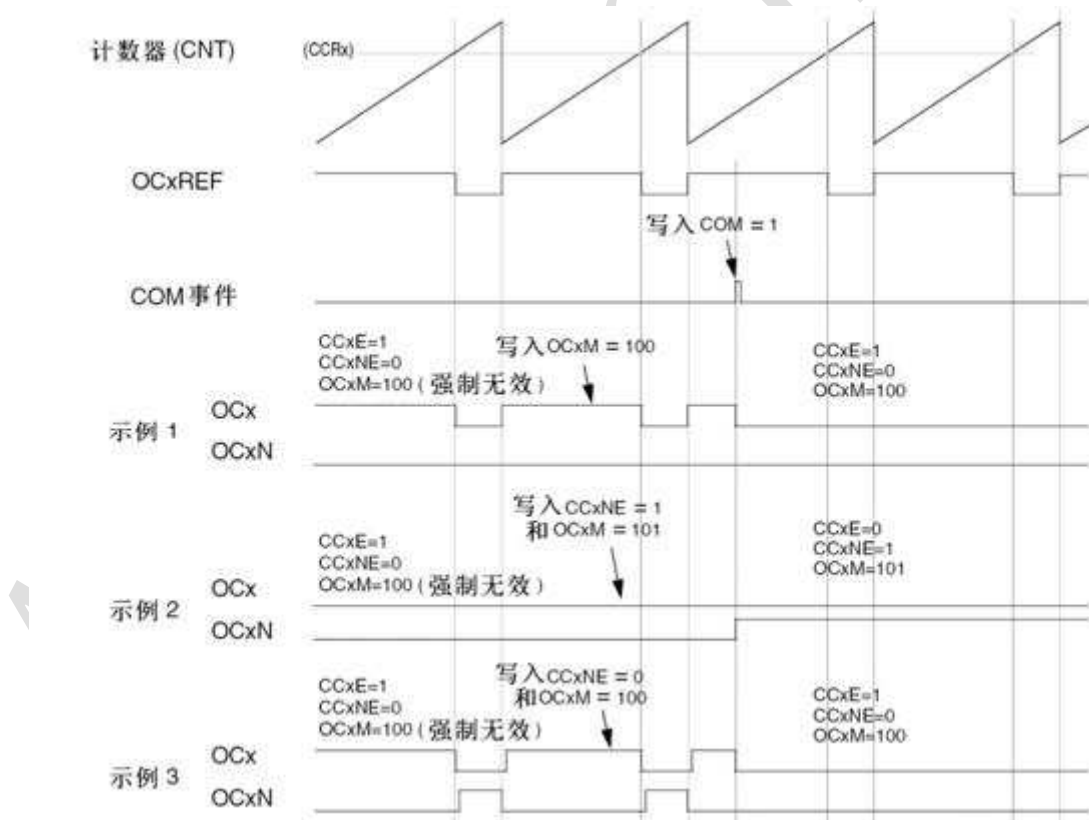
10.3.11. 六步 PWM 输出

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样你就可以预先设置好下一步骤配置，并在同一个时刻同时修更改所有通道的配置。

COM 可以通过设置 TIMx_EGR 寄存器的 COM 位由软件产生，或在 TRGI 上升沿由硬件产生(取决于 CR2 寄存器中的 CCUS 位)。当发生 COM 事件时会设置一个标志位(TIMx_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIMx_DIER 寄存器的 COMIE 位,则产生一个中断;如果已设置了 TIMx_DIER 寄存器的 COMDE 位,则产生一个 DMA 请求。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

图 10-17 产生六步 PWM，使用 COM 的例子(OSSR=1)



10.3.12. 正交编码器模式

从上节可知，编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容

始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

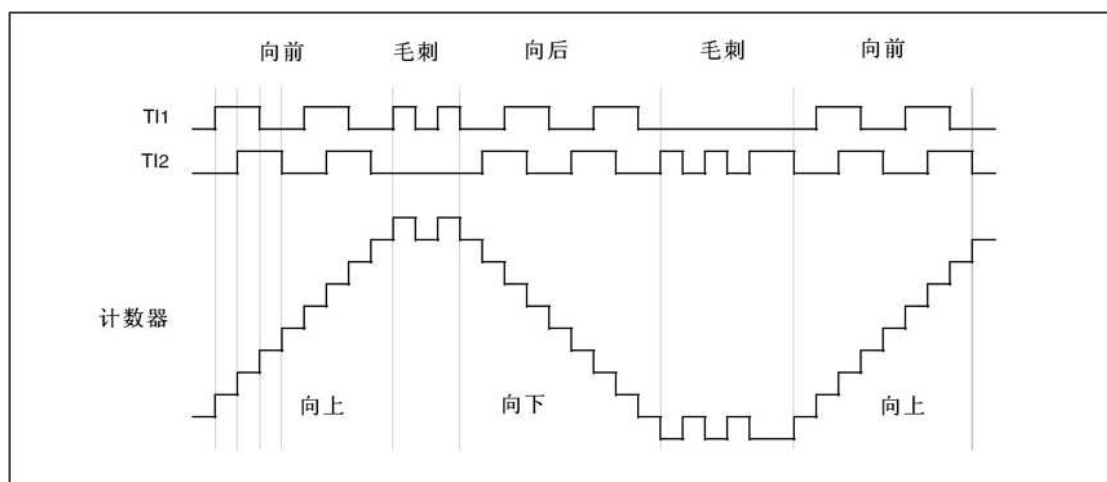
表 10-1 计数器方向和编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1 对应 TI2, TI2FP2 对应 TI1)	TI1FP1		TI2FP2	
		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
TI1 和 TI2 都计数	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

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。假定计数器已经启动 (TIMx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。

图 10-18 编码器模式下计数器操作实例



10.3.13. 与霍尔传感器的接口

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。下面给出了此特性用于连接霍尔传感器的例子。

使用高级控制定时器产生 PWM 信号驱动马达时，可以用另一个通用定时器作为“接口定时器”来连接霍尔传感器，见下图，3 个定时器输入脚通过一个异或门连接到 TI1 输入通道(通过设置 TIMx_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置于复位模式，从输入是 TI1F_ED。每当 3 个输入之一变化时，计数器重新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

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

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级定时器各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器。举例：霍尔输入连接到 TIMx 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIMx 的 PWM 配置。

置 TIMx_CR2 寄存器的 TI1S 位为‘1’，配置三个定时器输入逻辑或到 TI1 输入，

时基编程：置 TIMx_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。

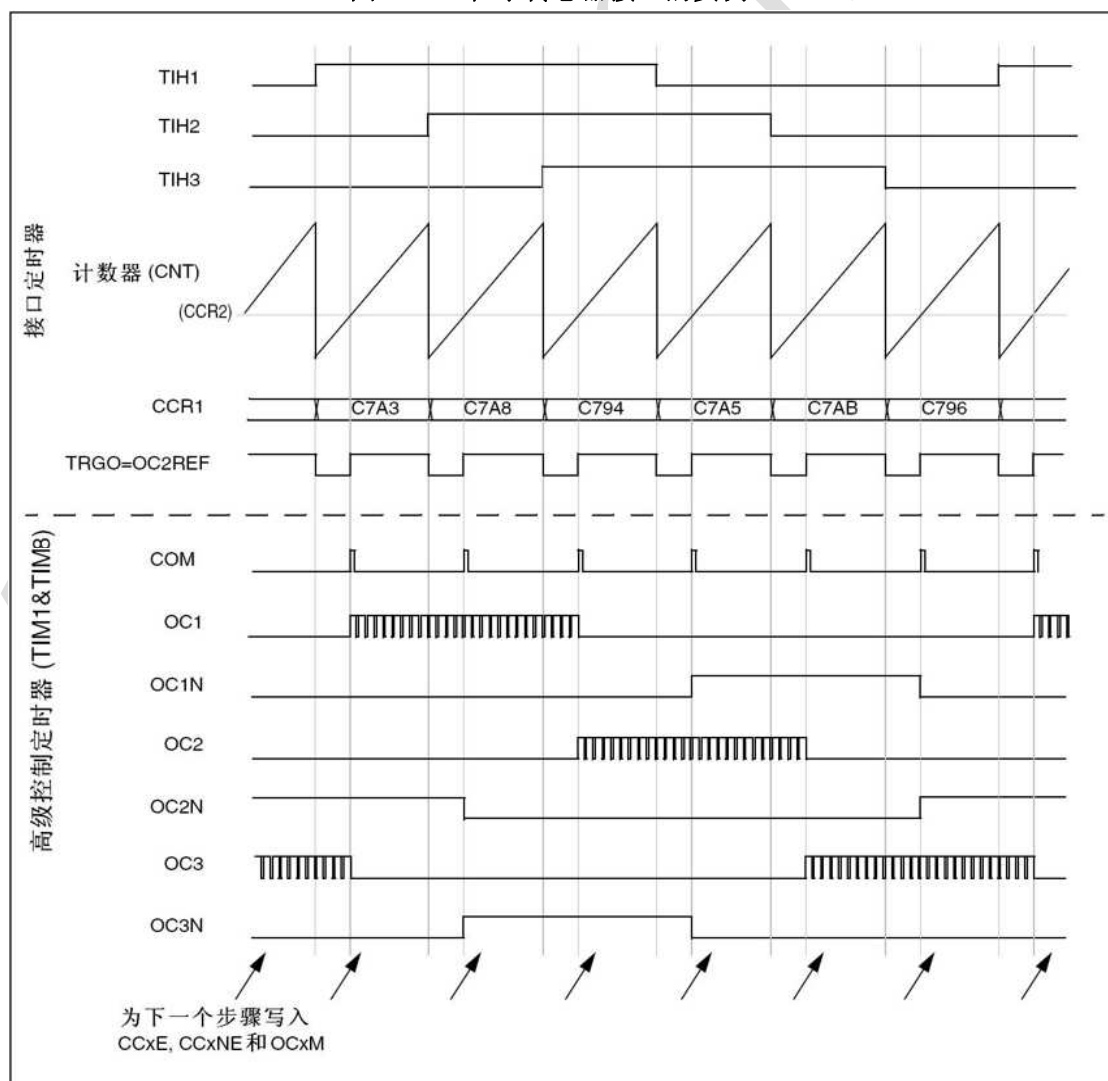
设置通道 1 为捕获模式(选中 TRC)：置 TIMx_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。

设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIMx_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。

选择 OC2REF 作为 TRGO 上的触发输出：置 TIMx_CR2 寄存器中的 MMS=101。

在高级控制寄存器中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIMx_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件(TIMx_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一步的 PWM 控制位(CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

图 10-19 霍尔传感器接口的实例



10.3.14. 单脉冲模式

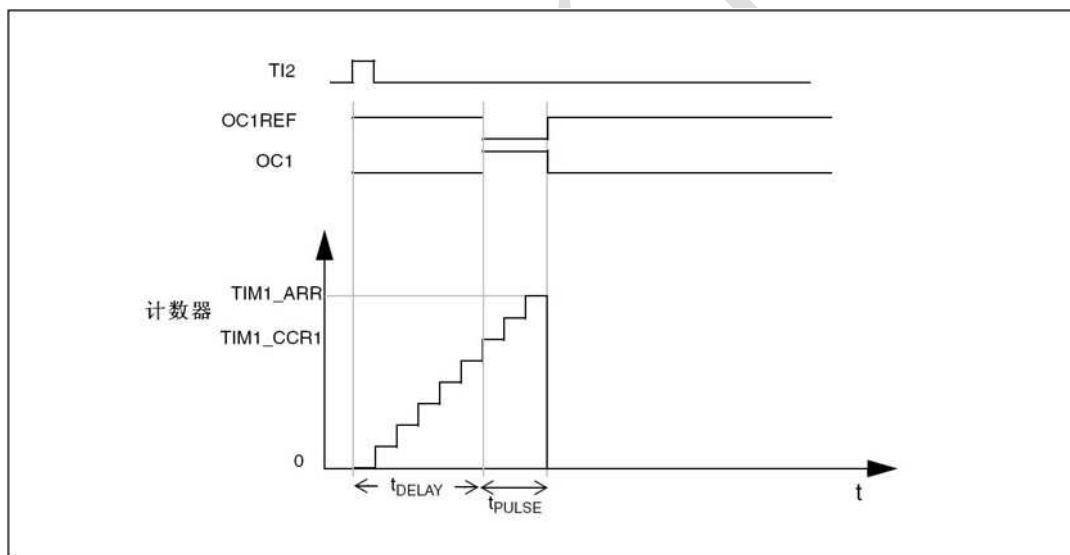
单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励,并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。可以通过从模式控制器启动计数器,在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器中的 OPM 位将选择单脉冲模式,这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时,才能产生一个脉冲。启动之前(当定时器正在等待触发),必须如下配置:

向上计数方式: 计数器 $CNT < CCRx \leq ARR$ (特别地, $0 < CCRx$),

向下计数方式: 计数器 $CNT > CCRx$ 。

图 10-20 单脉冲模式的例子



例如,你需要在从 TI2 输入脚上检测到一个上升沿开始,延迟 t_{DELAY} 之后,在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。假定 TI2FP2 作为触发 1:

置 TIMx_CCMR1 寄存器中的 CC2S=01, 把 TI2FP2 映像到 TI2。

置 TIMx_CCER 寄存器中的 CC2P=0, 使 TI2FP2 能够检测上升沿。

置 TIMx_SMCR 寄存器中的 TS=110, TI2FP2 作为从模式控制器的触发(TRGI)。

置 TIMx_SMCR 寄存器中的 SMS=110(触发模式), TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)

t_{DELAY} 由 TIMx_CCR1 寄存器中的值定义。

t_{PULSE} 由自动装载值和比较值之间的差值定义($TIMx\_ARR - TIMx\_CCR1$)。

假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形；首先要置 TIMx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。因为只需要一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=1，在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。

10.3.15. 定时器互连

TIMx 定时器能够在从模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

复位模式：

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIMx_ARR, TIMx_CCRx)都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

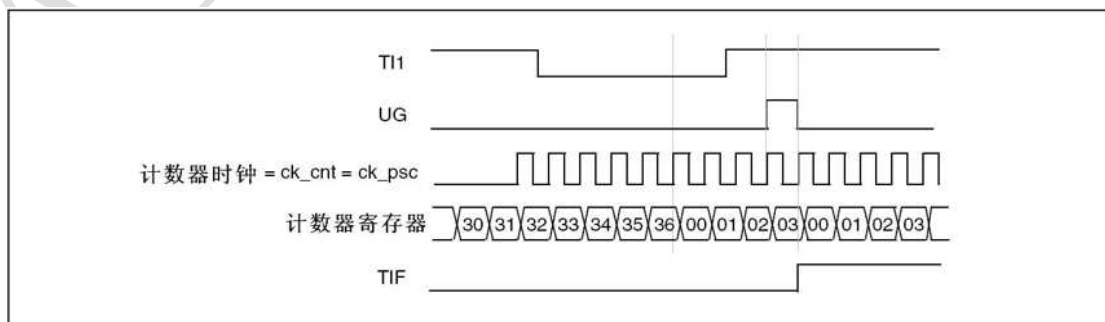
配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。

置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIMx_SR 寄存器中的 TIF 位)被设置，根据 TIMx_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

图 10-21 复位模式下的控制电路



门控模式：

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中, 不需要滤波, 所以保持 IC1F=0000)。触发操作中不使用捕获预分频器, 所以不需要配置。CC1S 位用于选择输入捕获源, 置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。

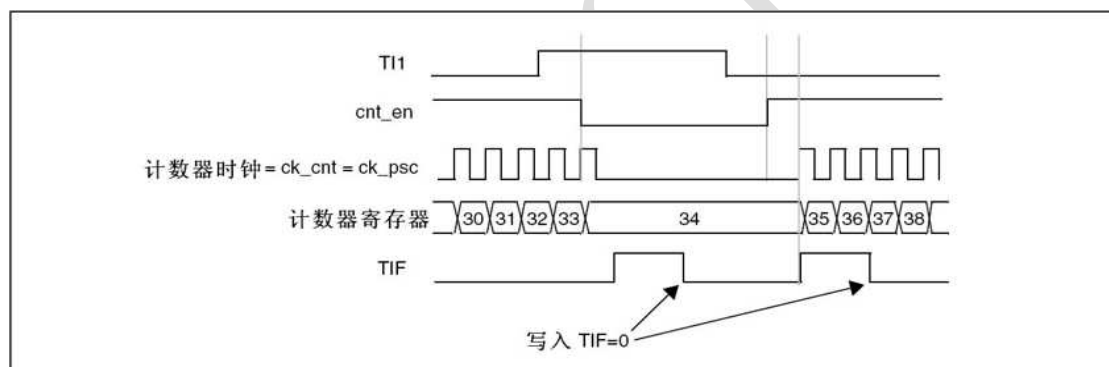
置 TIMx_SMCR 寄存器中 SMS=101, 配置定时器为门控模式; 置 TIMx_SMCR 寄存器中 TS=101, 选择 TI1 作为输入源。

置 TIMx_CR1 寄存器中 CEN=1, 启动计数器。在门控模式下, 如果 CEN=0, 则计数器不能启动, 不论触发输入电平如何。

只要 TI1 为低, 计数器开始依据内部时钟计数, 一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

图 10-22 门控模式下的控制电路



触发模式:

输入端上选中的事件使能计数器。

在下面的例子中, 计数器在 TI2 输入的上升沿开始向上计数:

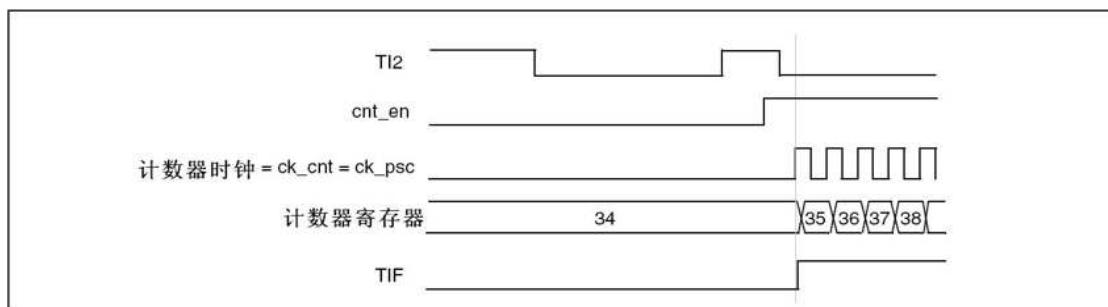
配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中, 不需要任何滤波器, 保持 IC2F=0000)。触发操作中不使用捕获预分频器, 不需要配置。CC2S 位只用于选择输入捕获源, 置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。

置 TIMx_SMCR 寄存器中 SMS=110, 配置定时器为触发模式; 置 TIMx_SMCR 寄存器中 TS=110, 选择 TI2 作为输入源。

当 TI2 出现一个上升沿时, 计数器开始在内部时钟驱动下计数, 同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时, 取决于 TI2 输入端的重同步电路。

图 10-23 触发模式下的控制电路

**外部时钟模式 2+触发模式：**

外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次：

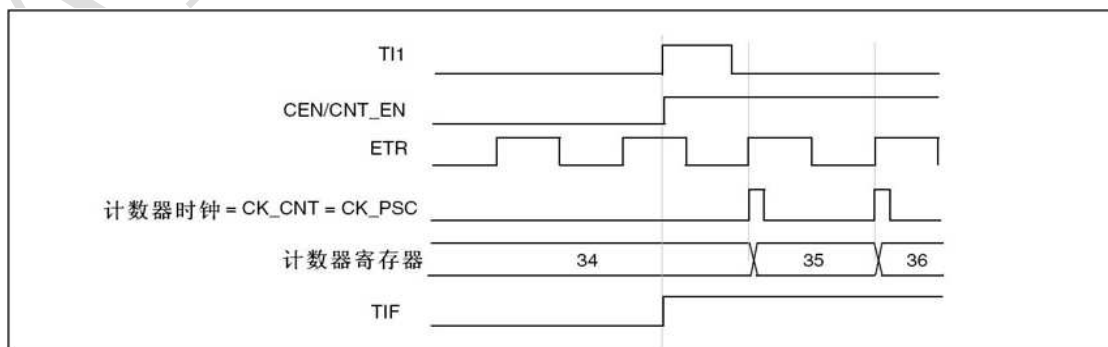
通过 TIMx_SMCR 寄存器配置外部时钟模式 2：ETF=0000：没有滤波，ETPS=00：不用预分频器，ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。

按如下配置通道 1，检测 TI 的上升沿：IC1F=0000：没有滤波；触发操作中不使用捕获预分频器，因此不需要配置；TIMx_CCMR1 寄存器中 CC1S=01，选择输入捕获源；TIMx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。

置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。ETR 信号的上升沿和计数器实际计数间的延时，取决于 ETRP 输入端的重同步电路。

图 10-24 外部时钟模式 2+触发模式控制电路



10.3.16. DMA 功能

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器，一些内部事件可以产生 DMA 请求，如 UPDATE 事件，比较事件等，需要配置 DIER 寄存器来使能相应的 DMA 请求。

定时器的 DMA 可以分为 single 模式的 DMA 请求和 burst 模式的 DMA 请求，由 DBER 寄存器来设置。

对于 single 模式的 DMA 请求来说，DMA 配置成 M2P 模式，PADDR 是目标寄存器地址，一旦产生 DMA 请求，DMA 就会访问该寄存器，如果目标地址配置了地址递增模式请按字对齐，那么下一个 DMA 请求将访问下一个地址的寄存器。

对于 burst 模式的 DMA 请求来说，有两个寄存器与之相关：TIMx_DCR 和 TIMx_DMAR。DMA 配置成 M2P 模式，PADDR 是 TIMx_DMAR 寄存器地址，DMA 就会访问 TIMx_DMAR 寄存器。实际上，TIMx_DMAR 寄存器只是一个缓冲，定时器会将 TIMx_DMAR 映射到一个内部寄存器，映射的初始寄存器由 TIMx_DCR 的 DBA 决定，后续每进行一次 DMA 传输，TIMx_DMAR 更新一次地址，从初始值往下加 4 即到下一个寄存器的地址。如果 TIMx_DCR 寄存器的 DBL 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMx_DCR 寄存器的 DBL 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求，在这 3 次请求下，DMA 对 TIMx_DMAR 寄存器的访问会映射到访问定时器的 DBA+0x4, DBA+0x8, DBA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送 (DBL+1) 次请求。

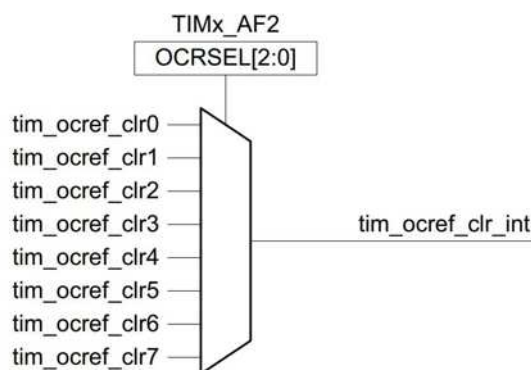
如果再来 1 次 DMA 请求事件，TIMx 将会重复上面的过程。

10.3.17. 外部事件清除 OCxREF 信号

当 OCREF_CLR_INT 信号为高电平时，指定通道的 OCxREF 信号会被清除 (TIMx_CCMRx 寄存器的 OCxCE 位需设置为 1)。在下次更新事件(UEV)前, OCxREF 信号保持为低。该功能仅用于比较输出和 PWM 模式。

OCREF_CLR_INT 信号可以通过 TIMx_AF2 寄存器中的 OCRSEL 位域配置。如图所示：

图 10-25 ocref_clr 输入源选择



10.4. 寄存器描述

TIM1 寄存器基地址：0x40012c00

TIM8 寄存器基地址：0x40013400

偏置	名称	描述
0x00	TIMx_CR1	TIMx 控制寄存器 1
0x04	TIMx_CR2	TIMx 控制寄存器 2
0x08	TIMx_SMCR	TIMx 从模式控制寄存器
0x0C	TIMx_DIER	TIMx DMA/中断使能寄存器
0x10	TIMx_SR	TIMx 状态寄存器
0x14	TIMx_EGR	TIMx 事件产生寄存器
0x18	TIMx_CCMR1	TIMx 捕获/比较模式寄存器 1
0x1C	TIMx_CCMR2	TIMx 捕获/比较模式寄存器 2
0x20	TIMx_CCER	TIMx 捕获/比较使能寄存器
0x24	TIMx_CNT	TIMx 计数器
0x28	TIMx_PSC	TIMx 预分频器
0x2C	TIMx_ARR	TIMx 自动装载寄存器
0x30	TIMx_RCR	TIMx 重复计数寄存器
0x34	TIMx_CCR1	TIMx 捕获比较寄存器 1

0x38	TIMx_CCR2	TIMx 捕获比较寄存器 2
0x3C	TIMx_CCR3	TIMx 捕获比较寄存器 3
0x40	TIMx_CCR4	TIMx 捕获比较寄存器 4
0x44	TIMx_BDTR	TIMx 刹车和死区控制寄存器
0x48	TIMx_DCR	TIMx DMA 控制寄存器
0x4C	TIMx_DMAR	TIMx 连续模式的 DMA 地址
0x54	TIMx_CCMR3	TIMx 捕获/比较模式寄存器 3
0x58	TIMx_CCR5	TIMx 捕获比较寄存器 5
0x5C	TIMx_CCR6	TIMx 捕获比较寄存器 6
0x60	TIMx_AF1	TIMx 复用功能选择寄存器 1
0x64	TIMx_AF2	TIMx 复用功能选择寄存器 2
0x68	TIMx_TISEL	TIMx 输入选择寄存器
0x6C	TIMx_DBER	TIMx DMA 请求类型选择寄存器

10.4.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15: 14	-	-	-	保留, 始终读为 0。
13:10	BKF	RW	0x0	刹车滤波器 (Break filter) 这几位定义了用于刹车输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器, 以 fDTS 采样 1000: 采样频率 fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 1001: 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010: 采样频率 fSAMPLING=fDTS/16, N=5

				0011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=8$ 1011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=6$ 0100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=6$ 1100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$ 0101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=8$ 1101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$ 0110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$ 1110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 0111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$ 1111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$
9:8	CKD	RW	0x0	时钟分频因子 死区发生器和数字滤波器所用的采样时钟(T_{dts})与 定时器时钟 (CK_INT) 的分频比例。 $F_{\text{dts}}=1/ T_{\text{dts}}$ 。 00: $t_{\text{DTS}}=t_{\text{CK\_INT}}$ 01: $t_{\text{DTS}}=2 \times t_{\text{CK\_INT}}$ 10: $t_{\text{DTS}}=4 \times t_{\text{CK\_INT}}$ 11:保留
7	ARPE	RW	0x0	自动重装载预装载允许位 0:TIMx_ARR 寄存器没有缓冲 1:TIMx_ARR 寄存器被装入缓冲器
6:5	CMS	RW	0x0	计数模式 00:边沿对齐模式, 计数器根据方向位 (DIR) 向上 或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计 数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向下 计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计 数。配置为输出的通道(TIMx_CCMRx 寄存器中

				CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	RW	0x0	方向控制位 0: 计数器向上计数； 1: 计数器向下计数。 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	RW	0x0	单脉冲模式 0: 在发生更新事件时，计数器不停止； 1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止。
2	URS	RW	0x0	更新请求源 软件通过该位选择 UEV 事件的源 0: 如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0x0	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生：

				– 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。
0	CEN	RW	0x0	使能计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

10.4.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
23:20	MMS2	RW	0x0	主模式选择 2 (Master mode selection 2) 这些位选择 TRG02, 可用于触发 ADC (TRG02), 具体使用请参考 ADC 外部触发部分。这些位的组合如下: 0000: 复位——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRG02)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRG02 上的信号相比实际复位会有延迟。 0001: 使能——计数器使能信号用作触发输出 (TRG02)。该触发输出可用于同时

			启动多个定时器，或者控制在一段时间内使能从定时器。 注意： 1、在门控模式下，计数器使能信号是 CEN 控制位和触发输入信号的逻辑与产生。 2、当计数器使能信号由触发输入控制时，TRG02 上会存在延迟。 0010： 更新——选择更新事件作为触发输出（TRG02）。例如，主定时器可用作从定时器的预分频器。 0011： 比较脉冲——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出（TRG02）都会发送一个正脉冲。 0100： 比较——OC1REF 信号用作触发输出 0101： 比较——OC2REF 信号用作触发输出 0110： 比较——OC3REF 信号用作触发输出 0111： 比较——OC4REF 信号用作触发输出 1000： 比较——OC5REF 信号用作触发输出 1001： 比较——OC6REF 信号用作触发输出 1010： 比较脉冲——OC4REF 上升沿或下降沿时，TRG02 上生成脉冲 1011： 比较脉冲——OC6REF 上升沿或下降沿时，TRG02 上生成脉冲 1100： 比较脉冲——OC4REF 或 OC6REF 上升沿时，TRG02 上生成脉冲 1101： 比较脉冲——OC4REF 上升沿或 OC6REF 下降沿时，TRG02 上生成脉冲 1110： 比较脉冲——OC5REF 或 OC6REF 上升沿时，TRG02 上生成脉冲

				1111: 比较脉冲——OC5REF 上升沿或 OC6REF 下降沿时, TRG02 上生成脉冲 注: 必须先使能从定时器或 ADC 的时钟, 才能从主定时器接收事件; 并且从主定时器接收触发信号时, 不得实时更改从定时器或 ADC 的时钟。
19	-	-	-	保留, 始终读为 0。
18	OIS6	RW	0x0	输出空闲状态 6(OC6 输出)。参见 OIS1 位。
17	-	-	-	保留, 始终读为 0。
16	OIS5	RW	0x0	输出空闲状态 5(OC5 输出)。参见 OIS1 位。
15	OIS4N	RW	0x0	输出空闲状态 4(OC4N 输出)。参见 OIS1N 位。
14	OIS4	RW	0x0	输出空闲状态 4(OC4 输出)。参见 OIS1 位。
13	OIS3N	RW	0x0	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。
12	OIS3	RW	0x0	输出空闲状态 3(OC3 输出)。参见 OIS1 位。
11	OIS2N	RW	0x0	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。
10	OIS2	RW	0x0	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	RW	0x0	输出空闲状态 1(OC1N 输出) (Output Idle state 1) 0: 当 MOE=0 时, 死区后 OC1N=0; 1: 当 MOE=0 时, 死区后 OC1N=1。 注: 已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
8	OIS1	RW	0x0	输出空闲状态 1(OC1 输出) (Output Idle state 1) 0: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=0; 1: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=1。 注: 已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
7	TIIS	RW	0x0	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。
6:4	MMS	RW	0x0	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: 000: 复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式),则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入 (TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。 100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。

				111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	CCDS	RW	0x0	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。
2	CCUS	RW	0x0	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1), 只能通过设置 COM 位更新它们; 1: 如果捕获/比较控制位是预装载的(CCPC=1), 可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注: 该位只对具有互补输出的通道起作用。
1	-	-	-	保留, 始终读为 0。
0	CCPC	RW	0x0	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位不是预装载的; 1: CCxE, CCxNE 和 OCxM 位是预装载的; 设置该位后, 它们只在设置了 COM 位后被更新。

10.4.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)

比特	名称	属性	复位值	描述
21:20	TS[4:3]	RW	0x0	和 TS[2:0]组成 5 位的 TS 信号。
19:16	-	-	-	保留, 始终读为 0。

15	ETP	RW	0x0	外部触发极性 (External trigger polarity) 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相, 高电平或上升沿有效; 1: ETR 被反相, 低电平或下降沿有效。
14	ECE	RW	0x0	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2; 1: 使能外部时钟模式 2。 计数器由 ETRF 信号上的任意有效边沿驱动。 注 1: 设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2: 下述从模式可以与外部时钟模式 2 同时使用: 复位模式, 门控模式和触发模式; 但是, 这时 TRGI 不能连到 ETRF(TS 位不能是'111')。 注 3: 外部时钟模式 1 和外部时钟模式 2 同时被使能时, 外部时钟的输入是 ETRF。
13:12	ETPS	RW	0x0	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 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。
11:8	ETF	RW	0x0	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产

				生一个输出的跳变。0000: 无滤波器, 以 fDTS 采样 1000: 采样频率 fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 1001: 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010: 采样频率 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
7	MSM	RW	0x0	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。
6:4	TS[2:0]	RW	0x0	触发选择 (Trigger selection) , 和 TS[4:3]组成 5 位的 TS 信号。 这 5 位选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) 00100: TI1 的边沿检测器(TI1F_ED) 00001: 内部触发 1(ITR1)

				00101: 滤波后的定时器输入 1(TI1FP1) 00010: 内部触发 2(ITR2) 00110: 滤波后的定时器输入 2(TI2FP2) 00011: 内部触发 3(ITR3) 00111: 外部触发输入(ETRF) 01000: 内部触发 4(ITR4) 01001: 内部触发 5(ITR5) 01010: 内部触发 6(ITR6) 01011: 内部触发 7(ITR7) 01100: 内部触发 8(ITR8) 01101: 内部触发 9(ITR9) 01110: 内部触发 10(ITR10) 其它: 保留 注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。 内部触发源 ITRx 详情请参考<u>定时器内部触发源 (ITR) 互连关系</u>。
3	OCCS	RW	0x0	OCREF 清零选择 (OCREF clear selection) 该位用于选择 OCREF 清零源。 0: OCREF_CLR_INT 连接到 COMPx 输出, 具体取决于 TIMx_AF2. OCRSEL 1: OCREF_CLR_INT 连接到 ETRF
2:0	SMS	RW	0x0	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 CEN=1, 则预分频器直接由内部时钟驱动。 001: 编码器模式 1 – 根据 TI1FP1 的电平, 计

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

10.4.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)

比特	名称	属性	复位值	描述
15	-	-	-	保留, 始终读为 0。
14	TDE	RW	0x0	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。
13	COMDE	RW	0x0	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。
12	CC4DE	RW	0x0	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。
11	CC3DE	RW	0x0	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。
10	CC2DE	RW	0x0	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。
9	CC1DE	RW	0x0	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。
8	UDE	RW	0x0	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。
7	BIE	RW	0x0	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。
6	TIE	RW	0x0	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。

5	COMIE	RW	0x0	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断; 1: 允许 COM 中断。
4	CC4IE	RW	0x0	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。
3	CC3IE	RW	0x0	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。
2	CC2IE	RW	0x0	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。
1	CC1IE	RW	0x0	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。
0	UIE	RW	0x0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

10.4.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
17	CC6IF	W0C	0x0	捕获/比较 6 中断标记 (Capture/Compare 6 interrupt flag) 参考 CC1IF 描述。
16	CC5IF	W0C	0x0	捕获/比较 5 中断标记 (Capture/Compare 5 interrupt

				flag) 参考 CC1IF 描述。
15:13	-	-	-	保留，始终读为 0。
12	CC4OF	W0C	0x0	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	W0C	0x0	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	W0C	0x0	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	W0C	0x0	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_CCR1 寄存器时，CC1IF 的状态已经为'1'。
8	-	-	-	位 8 保留，始终读为 0。
7	BIF	W0C	0x0	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。
6	TIF	W0C	0x0	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。
5	COMIF	W0C	0x0	COM 中断标记 (COM interrupt flag)

				一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置'1'。它由软件清'0'。 0: 无 COM 事件产生; 1: COM 中断等待响应。
4	CC4IF	W0C	0x0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	W0C	0x0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	W0C	0x0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	W0C	0x0	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: 当计数器值与比较值匹配时该位由硬件置 1, 但在中心对称模式下除外(参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生; 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时, 在向上或向上/下计数模式时计数器溢出, 或向下计数模式时的计数器下溢条件下, CC1IF 位变高 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置'1', 它由软件清'0'或通过读 TIMx_CCR1 清'0'。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	W0C	0x0	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件

				清'0'。 0: 无更新事件产生； 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1'： – 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。

10.4.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)

比特	名称	属性	复位值	描述
15:8	-	-	-	位 15:8 保留，始终读为 0。
7	BG	WO	0x0	产生刹车事件 (Break generation) 该位由软件置'1'，用于产生一个刹车事件，由硬件自动清'0'。 0: 无动作； 1: 产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
6	TG	WO	0x0	产生触发事件 (Trigger generation) 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0: 无动作； 1: TIMx_SR 寄存器的 TIF=1，若开启对应的中断

				和 DMA，则产生相应的中断和 DMA。
5	COMG	WO	0x0	捕获/比较事件，产生控制更新 (Capture/Compare control update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。 注：该位只对拥有互补输出的通道有效。
4	CC4G	WO	0x0	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	WO	0x0	产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	WO	0x0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0x0	产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0: 无动作； 1: 在通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	WO	0x0	产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。

				注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'; 若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。

10.4.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18)

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 CCxS 位定义输出比较模式:

比特	名称	属性	复位值	描述
15	OC2CE	RW	0x0	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	0x0	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0x0	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	RW	0x0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7	OC1CE	RW	0x0	输出比较 1 清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受 ETRF 输入的影响;

				1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。
6:4	OC1M	RW	0x0	输出比较 1 模式 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用; 001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时, 强制 OC1REF 为低。 011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器

				中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。
3	OC1PE	RW	0x0	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能, 可随时写入 TIMx_CCR1 寄存器, 并且新写入的数值立即起作用。 1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMx_CR1 寄存器的 OPM=1), 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定。
2	OC1FE	RW	0x0	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成

				PWM1 或 PWM2 模式时起作用。
1:0	CC1S	RW	0x0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC2F	RW	0x0	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	0x0	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	RW	0x0	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到

				N 个事件后会产生一个输出的跳变： 0000：无滤波器，以 fDTS 采样 1000：采样频率 fSAMPLING=fDTS/8，N=6 0001：采样频率 fSAMPLING=fCK_INT，N=2 1001：采样频率 fSAMPLING=fDTS/8，N=8 0010：采样频率 fSAMPLING=fCK_INT，N=4 1010：采样频率 fSAMPLING=fDTS/16，N=5 0011：采样频率 fSAMPLING=fCK_INT，N=8 1011：采样频率 fSAMPLING=fDTS/16，N=6 0100：采样频率 fSAMPLING=fDTS/2，N=6 1100：采样频率 fSAMPLING=fDTS/16，N=8 0101：采样频率 fSAMPLING=fDTS/2，N=8 1101：采样频率 fSAMPLING=fDTS/32，N=5 0110：采样频率 fSAMPLING=fDTS/4，N=6 1110：采样频率 fSAMPLING=fDTS/32，N=6 0111：采样频率 fSAMPLING=fDTS/4，N=8 1111：采样频率 fSAMPLING=fDTS/32，N=8
3:2	IC1PSC	RW	0x0	输入/捕获 1 预分频器 (Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMx_CCER 寄存器中)，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每 2 个事件触发一次捕获； 10：每 4 个事件触发一次捕获； 11：每 8 个事件触发一次捕获。
1:0	CC1S	RW	0x0	捕获/比较 1 选择 (Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出； 01：CC1 通道被配置为输入，IC1 映射在 TI1 上；

				10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。

10.4.8. TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2 偏移: 0x1C)

输出比较模式:

比特	名称	属性	复位值	描述
15	OC4CE	RW	0x0	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M	RW	0x0	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	RW	0x0	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	RW	0x0	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S	RW	0x0	捕获/比较 4 选择 (Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	RW	0x0	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M	RW	0x0	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	RW	0x0	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	RW	0x0	输出比较 3 快速使能 (Output compare 3 fast enable)

1:0	CC3S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC4F	RW	0x0	输入捕获 4 滤波器 (Input capture 4 filter)
11:10	IC4PSC	RW	0x0	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	RW	0x0	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	0x0	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。

10.4.9. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移： 0x20)

比特	名称	属性	复位值	描述
21	CC6P	RW	0x0	输入/捕获 6 输出极性 (Capture/Compare 6 output polarity) 参考 CC5P 的描述。
20	CC6E	RW	0x0	输入/捕获 6 输出使能 (Capture/Compare 6 output enable) 参考 CC5E 的描述。
19:18	-	-	-	保留，始终读为 0。
17	CC5P	RW	0x0	输入/捕获 5 输出极性 (Capture/Compare 5 output polarity) 参考 CC5P 的描述。
16	CC5E	RW	0x0	输入/捕获 5 输出使能 (Capture/Compare 5 output enable) 参考 CC5E 的描述。
15	CC4NP	RW	0x0	输入/捕获 4 互补输出极性 (Capture/Compare 4 complementary output polarity) 参考 CC1NP 的描述。
14	-	-	-	保留，始终读为 0。
13	CC4P	RW	0x0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0x0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0x0	输入/捕获 3 互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。

				述。
10	CC3NE	RW	0x0	输入/捕获 3 互补输出使能 (Capture/Compare 3 complementary output enable) 参考 CC1NE 的描述。
9	CC3P	RW	0x0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0x0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0x0	输入/捕获 2 互补输出极性 (Capture/Compare 2 complementary output polarity) 参考 CC1NP 的描述。
6	CC2NE	RW	0x0	输入/捕获 2 互补输出使能 (Capture/Compare 2 complementary output enable) 参考 CC1NE 的描述。
5	CC2P	RW	0x0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0x0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0x0	输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N 高电平有效; 1: OC1N 低电平有效。 CC1 通道配置为输入: 该位与 CC1P 结合使用以定义 TI1FP1 和 TI2FP1 的极性。参考 CC1P 的描述。 注: 一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 3 或 2 且 CC1S=00 (通道配置为输出) 则该位不能被修改。
2	CC1NE	RW	0x0	输入/捕获 1 互补输出使能 (Capture/Compare 1 complementary output enable) 0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E

				位的值。 1: 开启— OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	RW	0x0	输入/捕获 1 输出 极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出： 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入： CC1NP/CC1P 位选择 TI1FP1 和 TI2FP1 的有效极性，用于触发或捕获操作。 00: 不反相/上升沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿，在门控模式或编码器模式下触发操作，TIxFP1 不反相。 01: 反向/下降沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的下降沿，在门控模式或编码器模式下触发操作，TIxFP1 反相。 10: 保留，不使用此配置。 11: 不反相/双边沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿和下降沿，在门控模式下触发操作，TIxFP1 不反相（此配置不得在编码器模式下使用） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	RW	0x0	输入/捕获 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 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE

				位的值。 1: 开启— OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入：该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 0: 捕获禁止 1: 捕获使能

表 10-2 互补输出通道 OCx 和 OCxN 的控制

控制位					输出状态	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止（与定时器断开） OCx=0, OCx_EN=0	输出禁止（与定时器断开） OCxN=0, OCxN_EN=0
		0	0	1	输出禁止（与定时器断开） OCx=0, OCx_EN=0	OCxREF + 极性， OCxN=OCxREF $\oplus$ CCxNP OCxN_EN=1
		0	1	0	OCxREF + 极性， OCx=OCxREF $\oplus$ CCxP OCx_EN=1	输出禁止（与定时器断开） OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相 + 极性 + 死区， OCxN_EN=1
		1	0	0	输出禁止（与定时器断开） OCx=CCxP， OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP, OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP， OCx_EN=1	OCxREF + 极性， OCxN=OCxREF $\oplus$ CCxNP， OCxN_EN=1
		1	1	0	OCxREF + 极性， OCx=OCxREF $\oplus$ CCxP， OCxN_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP, OCx_EN=1
		1	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相 + 极性 + 死区， OCxN_EN=1

0	0	X	0	0	输出禁止（与定时器断开） OCx=CCxP， OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP，OCxN_EN=0
	0		0	1	输出禁止（与定时器断开） 异步地：OCx=CCxP，OCx_EN=0，OCxN=CCxNP， OCxN_EN=0， 若时钟存在：经过一个死区时间后，OCx=OISx， OCxN=OISx， 假设 OISx 和 OISxN 并不都对应 OCx 和 OCxN 的有效电平	
	0		1	0		
	0		1	1	输出禁止（与定时器断开） OCx=CCxP， OCx_EN=0	
	1		0	0		
	1		0	1	关闭状态（输出使能且为无效电平） 异步地：OCx=CCxP，OCx_EN=1，OCxN=CCxNP， OCxN_EN=1， 若时钟存在：经过一个死区时间后 OCx=OISx， OCxN=OISxN， 假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。	
	1		1	0		
	1		1	1		

10.4.10. TIMx 计数器（TIMx_CNT 偏移：0x24）

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0	计数器的值 (Counter value)

10.4.11. TIMx 预分频器（TIMx_PSC 偏移：0x28）

比特	名称	属性	复位值	描述
15:0	PSC	RW	0x0	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR

				的UG位清'0'或被工作在复位模式的从控制器清'0'

10.4.12. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0	自动重载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重载寄存器的值。当自动重载的值为空时, 计数器不工作。

10.4.13. TIMx 重复计数寄存器 (TIMx_RCR 偏移: 0x30)

比特	名称	属性	复位值	描述
15:8	-	-	-	位 15:8 保留, 始终读为 0。
7:0	REP	RW	0x0	重复计数器的值 (Repetition counter value) 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。每次向下计数器 REP_CNT 达到 0, 会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值, 因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。这意味着在 PWM 模式中, (REP+1)对应着: - 在边沿对齐模式下, PWM 周期的数目; - 在中心对称模式下, PWM 半周期的数目;

10.4.14. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)

比特	名称	属性	复位值	描述
15:0	CCR1	RW	0x0	捕获/比较通道 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出: CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。

注意: 当 CCMRx 寄存器的 OCxM 位为 PWM 模式时, 设置 CCRx=0 时占空比不为 0 (有一个时钟周期的高电平)。将 OCxM 设为强制无效电平, 可解决此问题。其他定时器的情况同 TIM1

10.4.15. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)

比特	名称	属性	复位值	描述
15:0	CCR2	RW	0x0	捕获/比较通道 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出: CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。如果在 TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 2 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入: CCR2

				包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

10.4.16. TIMx 捕获/比较寄存器 3 (TIMx_CCR3 偏移: 0x3C)

比特	名称	属性	复位值	描述
15:0	CCR3	RW	0x0	捕获/比较通道 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出: CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。如果在 TIMx_CCMR3 寄存器(OC3PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 3 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入: CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。

10.4.17. TIMx 捕获/比较寄存器 4 (TIMx_CCR4 偏移: 0x40)

比特	名称	属性	复位值	描述
15:0	CCR4	RW	0x0	捕获/比较通道 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出: CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。如果在 TIMx_CCMR4 寄存器(OC4PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 4 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC4 端口上

				产生输出信号。若 CC4 通道配置为输入：CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。

10.4.18. TIMx 刹车和死区寄存器 (TIMx_BDTR 偏移：0x44)

比特	名称	属性	复位值	描述
15	MOE	RW	0x0	主输出使能 (Main output enable) 一旦刹车输入有效，该位被硬件异步清'0'。根据 AOE 位的设置值，该位可以由软件清'0'或被自动置 1。它仅对配置为输出的通道有效。 0: 禁止 OC 和 OCN 输出或强制为空闲状态； 1: 如果设置了相应的使能位(TIMx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。
14	AOE	RW	0x0	自动输出使能 (Automatic output enable) 0: MOE 只能被软件置'1'； 1: MOE 能被软件置'1'或在下一个更新事件被自动置'1'(如果刹车输入无效)。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。
13	BKP	RW	0x0	刹车输入极性 (Break polarity) 0: 刹车输入低电平有效； 1: 刹车输入高电平有效。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
12	BKE	RW	0x0	刹车功能使能 (Break enable) 0: 禁止刹车输入(BRK 及 CCS 时钟失效事件)；

				1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。 注: 当设置了 LOCK 级别 1 时(TIMx_BDTR 寄存器中的 LOCK 位), 该位不能被修改。 注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
11	OSSR	RW	0x0	运行模式下“关闭状态”选择 (Off-state selection for Run mode) 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明。 0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0); 1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, 首先开启 OC/OCN 并输出无效电平, 然后置 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。
10	OSSI	RW	0x0	空闲模式下“关闭状态”选择 (Off-state selection for Idle mode) 该位用于当 MOE=0 且通道设为输出时。参考 OC/OCN 使能的详细说明。 0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0); 1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, OC/OCN 首先输出其空闲电平, 然后 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。
9:8	LOCK	RW	0x0	锁定设置 (Lock configuration) 该位为防止软件错

				误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位; 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出, CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位; 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出, CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位); 注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIMx_BDTR 寄存器, 则其内容冻结直至复位。
7:0	UTG	RW	0x0	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间: $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times Tdtg$, $Tdtg = TDTs$; $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times Tdtg$, $Tdtg = 2 \times TDTs$; $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg$, $Tdtg = 8 \times TDTs$; $DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg$, $Tdtg = 16 \times TDTs$; 例: 若 $TDTs = 125ns(8MHZ)$, 可能的死区时间为: 0 到 15875ns, 若步长时间为 125ns; 16us 到 31750ns, 若步长时间为 250ns; 32us 到 63us, 若步长时间为 1us;

				64us 到 126us，若步长时间为 2us； 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3，则不能修改这些位。

10.4.19. TIMx DMA 控制寄存器 (TIMx_DCR 偏移: 0x48)

比特	名称	属性	复位值	描述
15:13	-	-	-	位 15:13 保留，始终读为 0。
12:8	DBL	RW	0x0	DMA 连续传送长度 (DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，即：定义传输的次数，传输可以是半字(双字节)或字节： 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 10001: 18 次传输 例：我们考虑这样的传输：DBL=7, DBA=TIMx_CR1 - 如果 DBL=7, DBA=TIMx_CR1 表示待传输数据的地址，那么传输的地址由下式给出：(TIMx_CR1 的地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL 其中(TIMx_CR1 的地址) + DBA 再加上 7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址(TIMx_CR1 的地址) + DBA 开始的 7 个寄存器。 根据 DMA 数据长度的设置，可能发生以下情况： - 如果设置数据为半字(16 位)，那么数据就会传输给全部 7 个寄存器。

				- 如果设置数据为字节，数据仍然会传输给全部 7 个寄存器：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，以此类推。因此对于定时器，用户必须指定由 DMA 传输的数据宽度。
7:5	-	-	-	位 7:5 保留，始终读为 0。
4:0	DBA	RW	0x0	这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时)，DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量： 00000: TIMx_CR1, 00001: TIMx_CR2, 00010: TIMx_SMCR, ...

10.4.20. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)

比特	名称	属性	复位值	描述
31:0	DMAB	RW	0x0	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作：TIMx_CR1 地址 + DBA + DMA 索引，其中：“TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1)所在的地址；“DBA”是 TIMx_DCR 寄存器中定义的基地址；“DMA 索引”是由 DMA 自动控制的偏移量，它的最大值取决于 TIMx_DCR 寄存器中定义的 DBL。

10.4.21. TIMx 捕获/比较模式寄存器 3 (TIMx_CCMR3 偏移: 0x54)

通道 5 和通道 6 只能配置为输出。

比特	名称	属性	复位值	描述
24:16	RSV	-	-	保留
15	OC6CE	RW	0x0	输出比较 6 清 0 使能 (Output Compare 6 clear enable)
14:12	OC6M	RW	0x0	输出比较 6 模式 (Output Compare 6 mode), 参考 OC5M 的说明
11	OC6PE	RW	0x0	输出比较 6 预装载使能 (Output Compare 6 preload enable)
10	OC6FE	RW	0x0	输出比较 6 快速使能 (Output Compare 6 fast enable)
9:8	-	-	-	保留, 始终读为 0。
7	OC5CE	RW	0x0	输出比较 5 清'0'使能 (Output Compare 1 clear enable) 0: OC5REF 不受 ETRF 输入的影响; 1: 一旦检测到 ETRF 输入高电平, 清除 OC5REF=0。
6:4	OC5M	RW	0x0	输出比较 5 模式 (Output Compare 5 mode) 该 3 位定义了输出参考信号 OC5REF 的动作。 000: 冻结。输出比较寄存器 TIMx_CCR5 与计数器 TIMx_CNT 间的比较对 OC5REF 不起作用; 001: 匹配时设置通道 5 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 5 (TIMx_CCR5)相同时, 强制 OC5REF 为高。 010: 匹配时设置通道 5 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器

				5(TIMx_CCR5)相同时, 强制 OC5REF 为低。 011: 翻转。当 TIMx_CCR5=TIMx_CNT 时, 翻转 OC5REF 的电平。 100: 强制为无效电平。强制 OC5REF 为低。 101: 强制为有效电平。强制 OC5REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR5 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR5 时通道 1 为无效电平 (OC5REF=0), 否则为有效电平(OC5REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR5 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR5 时通道 1 为有效电平, 否则为无效电平。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC5REF 电平才改变。
3	OC5PE	RW	0x0	输出比较 5 预装载使能 (Output Compare 5 preload enable)
2	OC5FE	RW	0x0	输出比较 5 快速使能 (Output Compare 5 fast enable)
1:0	-	-	-	保留, 始终读为 0。

10.4.22. TIMx 捕获/比较寄存器 5 (TIMx_CCR5 偏移: 0x58)

比特	名称	属性	复位值	描述
15:0	CCR5	RW	0x0	捕获/比较通道 5 的值 (Capture/Compare 1 value) 若 CC5 通道配置为输出: CCR5 包含了装入当前捕获/比较 5 寄存器的值(预装载值)。如果在 TIMx_CCMR3 寄存器(OC5PE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 5 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC5 端口上产生输出信号。若 CC5 通道配置为输入: CCR5 包含了由上一次输入捕获 5 事件(IC5)传输的计数器值。

10.4.23. TIMx 捕获/比较寄存器 6 (TIMx_CCR6 偏移: 0x5C)

比特	名称	属性	复位值	描述
15:0	CCR6	RW	0x0	捕获/比较通道 6 的值 (Capture/Compare 6 value) 若 CC6 通道配置为输出: CCR6 包含了装入当前捕获/比较 6 寄存器的值(预装载值)。如果在 TIMx_CCMR3 寄存器(OC6PE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 6 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC6 端口上产生输出信号。若 CC6

				通道配置为输入：CCR6 包含了由上一次输入捕获 6 事件(IC6)传输的计数器值。

10.4.24. TIMx 复用功能选择寄存器 1 (TIMx_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
17:14	ETRSEL	RW	0000	ETR 输入源选择 0000: ETR0 0001: ETR1 0010: ETR2 0011: ETR3 0100: ETR4 0101: ETR5 0110: ETR6 0111: ETR7 其他: 保留 详情请参考定时器外部触发源 (ETR) 互连关系
13	BKCMP4P	RW	0x0	比较器 4 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
12	BKCMP3P	RW	0x0	比较器 3 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
11	BKCMP2P	RW	0x0	比较器 2 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
10	BKCMP1P	RW	0x0	比较器 1 输入极性控制 0: 输入高电平有效

				1: 输入低电平有效
9	BKINP	RW	0x0	刹车输入极性控制 0: 输入高电平有效 1: 输入低电平有效
8:5	-	-	-	保留, 始终为 0。
4	BKCMP4E	RW	0x0	比较器 4 输入使能控制 0: 禁止 1: 使能
3	BKCMP3E	RW	0x0	比较器 3 输入使能控制 0: 禁止 1: 使能
2	BKCMP2E	RW	0x0	比较器 2 输入使能控制 0: 禁止 1: 使能
1	BKCMP1E	RW	0x0	比较器 1 输入使能控制 0: 禁止 1: 使能
0	BKINE	RW	0x0	刹车输入使能控制 0: 禁止 1: 使能

刹车输入详情请参考定时器刹车输入互连关系。

10.4.25. TIMx 复用功能选择寄存器 2 (TIMx_AF2 偏移: 0x64)

比特	名称	属性	复位值	描述
18:16	OCRSEL[2:0]	RW	0x0	ocref_clr 源选择 这些位选择 ocref_clr 输入源。 000: tim_ocref_clr0 001: tim_ocref_clr1 010: tim_ocref_clr2

				011: tim_ocref_clr3 100: tim_ocref_clr4 101: tim_ocref_clr5 110: tim_ocref_clr6 111: tim_ocref_clr7 详情请参考 OCxREF 清除输入信号互连关系
15:0	-	-	-	保留, 始终为 0。

10.4.26. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
31:28	-	-	-	保留, 始终为 0。
27:24	TI4SEL	RW	0x0	TI4 输入选择 0000: tim_ti4_in0 0001: tim_ti4_in1 0010: tim_ti4_in2 0011: tim_ti4_in3 ... 1111: tim_ti4_in15 详情请参考 定时器输入信号互连关系
23:20	-	-	-	保留, 始终为 0。
19:16	TI3SEL	RW	0x0	TI3 输入选择 0000: tim_ti3_in0 0001: tim_ti3_in1 0010: tim_ti3_in2 0011: tim_ti3_in3 ... 1111: tim_ti3_in15 详情请参考 定时器输入信号互连关系

15:12	-	-	-	保留，始终为 0。
11:8	TI2SEL	RW	0x0	TI2 输入选择 0000: tim_ti2_in0 0001: tim_ti2_in1 0010: tim_ti2_in2 0011: tim_ti2_in3 ... 1111: tim_ti2_in15 详情请参考定时器输入信号互连关系
7:4	-	-	-	保留，始终为 0。
3:0	TI1SEL	RW	0x0	TI1 输入选择 0000: tim_ti1_in0 0001: tim_ti1_in1 0010: tim_ti1_in2 0011: tim_ti1_in3 ... 1111: tim_ti1_in15 详情请参考定时器输入信号互连关系

10.4.27. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留，始终读为 0。
6	TBE	RW	0x0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0x0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;

4	CC4BE	RW	0x0	捕获/比较 4 事件的 DMA 请求类型 0: Single; 1: Burst;
3	CC3BE	RW	0x0	捕获/比较 3 事件的 DMA 请求类型 0: Single; 1: Burst;
2	CC2BE	RW	0x0	捕获/比较 2 事件的 DMA 请求类型 0: Single; 1: Burst;
1	CC1BE	RW	0x0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0x0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

注：当选择 burst 模式时，DMA 的 TIMER 外设地址只能选择 DMAB 寄存器地址，此时还需配置 DCR 寄存器以决定起始地址和 burst size。如果选择 single 模式时，DMA 的 TIMER 外设地址为除了 DMAB 寄存器以外寄存器的地址。

11. 通用定时器（TIM2）

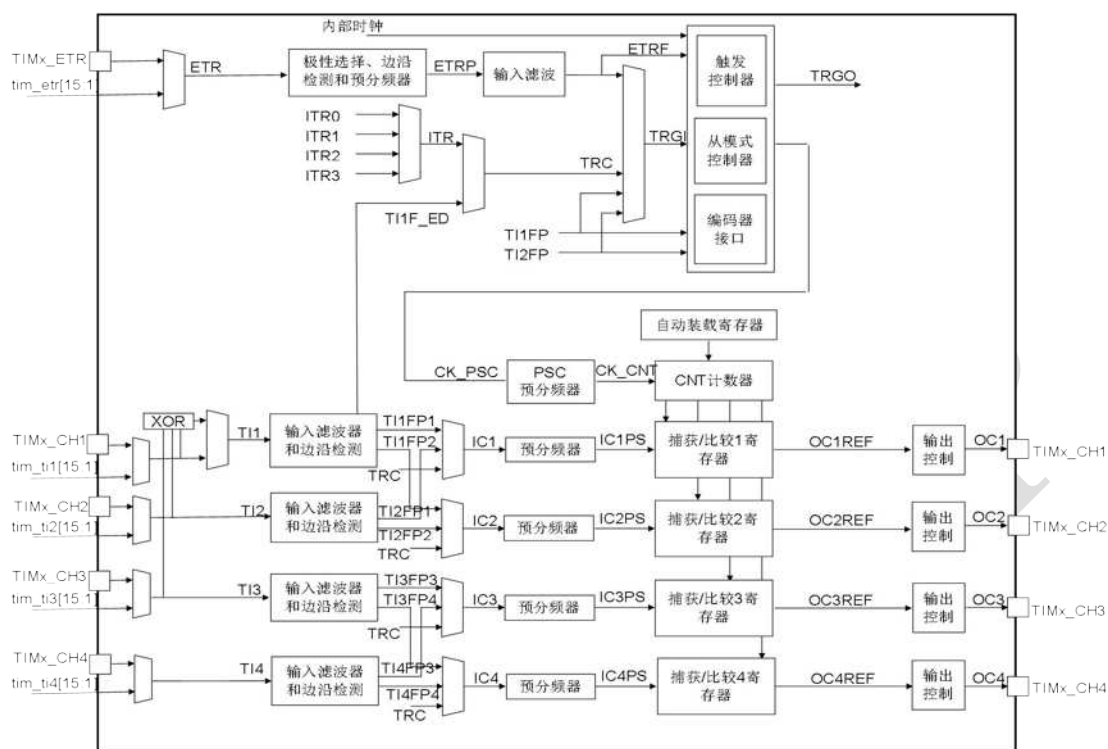
11.1. 概述

通用定时器 TIM2 由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM 等)。高级控制定时器和通用定时器是完全独立的，它们不共享任何资源，但它们可以同步操作。

11.2. 主要特性

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值
- 多达 4 个独立通道：
 - 输入捕获
 - 输出比较
 - PWM 生成(边缘或中间对齐模式)
 - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 如下事件发生时产生中断/DMA：
 - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
 - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

图 11-1 通用定时器 TIM2 结构框图



11.3. 功能描述

11.3.1. 计数单元

请参考 TIM1 相应章节。

11.3.2. 预分频器

请参考 TIM1 相应章节。

11.3.3. 时钟源选择

请参考 TIM1 相应章节。

11.3.4. 捕获比较通道

请参考 TIM1 相应章节。

11.3.5. 强制输出模式

请参考 TIM1 相应章节。

11.3.6. PWM 模式

请参考 TIM1 相应章节。

11.3.7. 正交编码器模式

请参考 TIM1 相应章节。

11.3.8. 与霍尔传感器的接口

请参考 TIM1 相应章节。

11.3.9. 单脉冲模式

请参考 TIM1 相应章节。

11.3.10. 定时器互连

11.3.11. DMA 功能

请参考 TIM1 相应章节。

11.3.12. 外部事件清除 OCxREF 信号

请参考 TIM1 相应章节。

11.4. 寄存器描述

TIM2 寄存器基地址：0x4000 0000

偏置	名称	描述
0x00	TIM2_CR1	TIM2 控制寄存器 1
0x04	TIM2_CR2	TIM2 控制寄存器 2
0x08	TIM2_SMCR	TIM2 从模式控制寄存器
0x0C	TIM2_DIER	TIM2 DMA/中断使能寄存器
0x10	TIM2_SR	TIM2 状态寄存器
0x14	TIM2_EGR	TIM2 事件产生寄存器
0x18	TIM2_CCMR1	TIM2 捕获/比较模式寄存器 1
0x1C	TIM2_CCMR2	TIM2 捕获/比较模式寄存器 2
0x20	TIM2_CCER	TIM2 捕获/比较使能寄存器
0x24	TIM2_CNT	TIM2 计数器
0x28	TIM2_PSC	TIM2 预分频器
0x2C	TIM2_ARR	TIM2 自动装载寄存器
0x34	TIM2_CCR1	TIM2 捕获比较寄存器 1
0x38	TIM2_CCR2	TIM2 捕获比较寄存器 2
0x3C	TIM2_CCR3	TIM2 捕获比较寄存器 3
0x40	TIM2_CCR4	TIM2 捕获比较寄存器 4
0x48	TIM2_DCR	TIM2 DMA 控制寄存器
0x4C	TIM2_DMAR	TIM2 连续模式的 DMA 地址
0x60	TIM2_AF1	TIM2 复用功能选择寄存器 1
0x64	TIM2_AF2	TIM2 复用功能选择寄存器 2
0x68	TIM2_TISEL	TIM2 输入选择寄存器
0x6C	TIM2_DBER	TIM2 DMA 请求类型选择寄存器

11.4.1. TIM2 控制寄存器 1 (TIM2_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15: 10	-	-	-	保留, 始终读为 0。
9:8	CKD	RW	0x0	时钟分频因子 死区发生器和数字滤波器所用的采样时钟(Tdts)与定时器时钟 (CK_INT) 的分频比例。 00: tDTS=tCK_INT 01: tDTS=2 x tCK_INT 10: tDTS=4 x tCK_INT 11:保留
7	ARPE	RW	0x0	自动重装载预装载允许位 0:TIM2_ARR 寄存器没有缓冲 1:TIM2_ARR 寄存器被装入缓冲器
6:5	CMS	RW	0x0	计数模式 00:边沿对齐模式, 计数器根据方向位 (DIR) 向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIM2_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向下计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIM2_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIM2_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被设置。

				注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	RW	0x0	方向控制位 0：计数器向上计数； 1：计数器向下计数。 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	RW	0x0	单脉冲模式 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止。
2	URS	RW	0x0	更新请求源 软件通过该位选择 UEV 事件的源 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0x0	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新(UEV)事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器) 1：禁止 UEV。不产生更新事件，影子寄存器(ARR、

				PSC、CCR _x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	RW	0x0	使能计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

11.4.2. TIM2 控制寄存器 2 (TIM2_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
23:8	-	-	-	保留。
7	TI1S	RW	0x0	TI1 选择 (TI1 selection) 0: TIM2_CH1 引脚连到 TI1 输入; 1: TIM2_CH1、TIM2_CH2 和 TIM2_CH3 引脚经异或后连到 TI1 输入。
6:4	MMS	RW	0x0	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: 000: 复位 – TIM2_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控

				模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIM2_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	CCDS	RW	0x0	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。
2	CCUS	RW	0x0	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们； 1: 如果捕获/比较控制位是预装载的(CCPC=1)，

				可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。注：该位只对具有互补输出的通道起作用。
1	-	-	-	保留，始终读为 0。
0	CCPC	RW	0x0	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位不是预装载的； 1: CCxE, CCxNE 和 OCxM 位是预装载的； 设置该位后，它们只在设置了 COM 位后被更新。

11.4.3. TIM2 从模式控制寄存器 (TIM2_SMCR 偏移: 0x08)

比特	名称	属性	复位值	描述
21:20	TS[4:3]	RW	0x0	触发选择信号
19:16	-	-	-	保留，始终读为 0。
15	ETP	RW	0x0	外部触发极性 (External trigger polarity) 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相，高电平或上升沿有效； 1: ETR 被反相，低电平或下降沿有效。
14	ECE	RW	0x0	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2； 1: 使能外部时钟模式 2。 计数器由 ETRF 信号上的任意有效边沿驱动。 注 1：设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2：下述从模式可以与外部时钟模式 2 同时

				使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是'111')。 注 3：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。
13:12	ETPS	RW	0x0	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 TIM2CLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00：关闭预分频； 01：ETRP 频率除以 2； 10：ETRP 频率除以 4； 11：ETRP 频率除以 8。
11:8	ETF	RW	0x0	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。0000：无滤波器，以 fDTS 采样 1000：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$，N=6 0001：采样频率 $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$，N=2 1001：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$，N=8 0010：采样频率 $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$，N=4 1010：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$，N=5 0011：采样频率 $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$，N=8 1011：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$，N=6 0100：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$，N=6 1100：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$，N=8 0101：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$，N=8 1101：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$，N=5

				0110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$ 1110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 0111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$ 1111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$
7	MSM	RW	0x0	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。
6:4	TS[2:0]	RW	0x0	触发选择 (Trigger selection) , 和 TS[4:3]组成 5 位的 TS 信号。 这 5 位选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) 00100: TI1 的边沿检测器(TI1F_ED) 00001: 内部触发 1(ITR1) 00101: 滤波后的定时器输入 1(TI1FP1) 00010: 内部触发 2(ITR2) 00110: 滤波后的定时器输入 2(TI2FP2) 00011: 内部触发 3(ITR3) 00111: 外部触发输入(ETRF) 01000: 内部触发 4(ITR4) 01001: 内部触发 5(ITR5) 01010: 内部触发 6(ITR6) 01011: 内部触发 7(ITR7) 01100: 内部触发 8(ITR8) 01101: 内部触发 9(ITR9) 01110: 内部触发 10(ITR10) 其它: 保留

				注：这些位只能在未用到(如 SMS=000)时被改变，以避免在改变时产生错误的边沿检测。 内部触发源 ITRx 详情请参考<u>定时器内部触发源（ITR）互连关系</u>。
3	OCCS	RW	0x0	OCREF 清零选择（OCREF clear selection） 该位用于选择 OCREF 清零源。 0：OCREF_CLR_INT 连接到 COMPx 输出，具体取决于 TIM2_AF2. OCRSEL 1： OCREF_CLR_INT 连接到 ETRF
2:0	SMS	RW	0x0	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 CEN=1，则预分频器直接由内部时钟驱动。 001：编码器模式 1 – 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。 010：编码器模式 2 – 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。 011：编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 100：复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101：门控模式 – 当触发输入 (TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110：触发模式 – 计数器在触发输入 TRGI 的

				上升沿启动(但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式 1 – 选中的触发输入(TRGI) 的上升沿驱动计数器。 注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。

11.4.4. TIM2 DMA/中断使能寄存器 (TIM2_DIER 偏移: 0x0C)

比特	名称	属性	复位值	描述
15	-	-	-	保留, 始终读为 0。
14	TDE	RW	0x0	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。
13	COMDE	RW	0x0	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。
12	CC4DE	RW	0x0	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。
11	CC3DE	RW	0x0	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。
10	CC2DE	RW	0x0	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。
9	CC1DE	RW	0x0	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。
8	UDE	RW	0x0	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。
7	-	-	-	保留, 始终读为 0。
6	TIE	RW	0x0	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
5	COMIE	RW	0x0	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断; 1: 允许 COM 中断。
4	CC4IE	RW	0x0	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。
3	CC3IE	RW	0x0	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。
2	CC2IE	RW	0x0	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。
1	CC1IE	RW	0x0	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt

				enable) 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。
0	UIE	RW	0x0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

11.4.5. TIM2 状态寄存器 (TIM2_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
17:13	-	-	-	保留, 始终读为 0。
12	CC4OF	W0C	0x0	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	W0C	0x0	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	W0C	0x0	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	W0C	0x0	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 TIM2_CCR1 寄存器时, CC1IF 的状态已经为'1'。
8:7	-	-	-	保留, 始终读为 0。
6	TIF	W0C	0x0	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。
5	COMIF	W0C	0x0	COM 中断标记 (COM interrupt flag) 一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置'1'。它由软件清'0'。 0: 无 COM 事件产生； 1: COM 中断等待响应。
4	CC4IF	W0C	0x0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	W0C	0x0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	W0C	0x0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	W0C	0x0	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: 当计数器值与比较值匹配时该位由硬件置 1, 但在中心对称模式下除外(参考 TIM2_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生； 1: TIM2_CNT 的值与 TIM2_CCR1 的值匹配。 当 TIM2_CCR1 的内容大于 TIM2_APR 的内容时, 在向上或向上/下计数模式时计数器溢出, 或向下计数模式时的计数器下溢条件下, CC1IF 位变高 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置'1', 它由软件清'0'或通过读 TIM2_CCR1 清'0'。 0: 无输入捕获产生；

				1: 计数器值已被捕获(拷贝)至 TIM2_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	W0C	0x0	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1': – 若 TIM2_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIM2_CR1 寄存器的 URS=0、UDIS=0, 当设置 TIM2_EGR 寄存器的 UG=1 时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 – 若 TIM2_CR1 寄存器的 URS=0、UDIS=0, 当计数器 CNT 被触发事件重新初始化时。

11.4.6. TIM2 事件产生寄存器 (TIM2_EGR 偏移: 0x14)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留, 始终读为 0。
6	TG	WO	0x0	产生触发事件 (Trigger generation) 该位由软件置'1', 用于产生一个触发事件, 由硬件自动清'0'。 0: 无动作; 1: TIM2_SR 寄存器的 TIF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。
5	COMG	WO	0x0	捕获/比较事件, 产生控制更新 (Capture/Compare control update generation)

				该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。 注：该位只对拥有互补输出的通道有效。
4	CC4G	WO	0x0	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	WO	0x0	产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	WO	0x0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0x0	产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0: 无动作； 1: 在通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIM2_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	WO	0x0	产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。 注意预分频器的计数器也被清'0' (但是预分频系数不变)。若在中心对称模式下或 DIR=0 (向上计数) 则计数器被清'0'；若 DIR=1 (向下计数) 则计数器取

				TIM2_ARR 的值。

11.4.7. TIM2 捕获/比较模式寄存器 1 (TIM2_CCMR1 偏移: 0x18)

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 CCxS 位定义输出比较模式:

比特	名称	属性	复位值	描述
15	OC2CE	RW	0x0	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	0x0	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0x0	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	RW	0x0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E=0)才是可写的。
7	OC1CE	RW	0x0	输出比较 1 清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受 ETRF 输入的影响; 1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。
6:4	OC1M	RW	0x0	输出比较 1 模式 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作, 而

				OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效,而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIM2_CCR1 与计数器 TIM2_CNT 间的比较对 OC1REF 不起作用; 001: 匹配时设置通道 1 为有效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1 (TIM2_CCR1)相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1 (TIM2_CCR1)相同时, 强制 OC1REF 为低。 011: 翻转。当 TIM2_CCR1=TIM2_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平。 注 1: 一旦 LOCK 级别设为 3(TIM2_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比

				较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	RW	0x0	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0: 禁止 TIM2_CCR1 寄存器的预装载功能，可随时写入 TIM2_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 TIM2_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM2_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIM2_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIM2_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	RW	0x0	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S	RW	0x0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选

				择： 00： CC1 通道被配置为输出； 01： CC1 通道被配置为输入，IC1 映射在 TI1 上； 10： CC1 通道被配置为输入，IC1 映射在 TI2 上； 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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注： CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式：

比特	名称	属性	复位值	描述
15:12	IC2F	RW	0x0	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	0x0	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注： CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	RW	0x0	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000： 无滤波器，以 fDTS 采样 1000： 采样频率 fSAMPLING=fDTS/8， N=6

				0001: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=2$ 1001: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=8$ 0010: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=4$ 1010: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=5$ 0011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=8$ 1011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=6$ 0100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=6$ 1100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$ 0101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=8$ 1101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$ 0110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$ 1110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 0111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$ 1111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$
3:2	IC1PSC	RW	0x0	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 $\text{CC1E}=0$ (TIM2_CCER 寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每 2 个事件触发一次捕获; 10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获。
1:0	CC1S	RW	0x0	捕获/比较 1 选择 (Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 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器输入被选中时(由

				TIM2_SMCR 寄存器的 TS 位选择)。

11.4.8. TIM2 捕获/比较模式寄存器 2 (TIM2_CCMR2 偏移: 0x1C)

输出比较模式:

比特	名称	属性	复位值	描述
15	OC4CE	RW	0x0	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M	RW	0x0	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	RW	0x0	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	RW	0x0	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S	RW	0x0	捕获/比较 4 选择 (Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	RW	0x0	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M	RW	0x0	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	RW	0x0	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	RW	0x0	输出比较 3 快速使能 (Output compare 3 fast enable)
1:0	CC3S	RW	0x0	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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM2_CCER 寄存器的 CC3E=0)才是可写的。

输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC4F	RW	0x0	输入捕获 4 滤波器 (Input capture 4 filter)
11:10	IC4PSC	RW	0x0	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	0x0	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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	RW	0x0	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	0x0	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	0x0	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器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。

11.4.9. TIM2 捕获/比较使能寄存器 (TIM2_CCER 偏移: 0x20)

比特	名称	属性	复位值	描述
21:16	-	-	-	保留, 始终读为 0。
15	CC4NP	RW	0x0	输入/捕获 4 互补输出极性 (Capture/Compare 4 complementary output polarity) 参考 CC1NP 的描述。
14	-	-	-	保留, 始终读为 0。
13	CC4P	RW	0x0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0x0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0x0	输入/捕获 3 互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。
10	CC3NE	RW	0x0	输入/捕获 3 互补输出使能 (Capture/Compare 3 complementary output enable) 参考 CC1NE 的描述。
9	CC3P	RW	0x0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0x0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0x0	输入/捕获 2 互补输出极性 (Capture/Compare 2 complementary output polarity) 参考 CC1NP 的描述。
6	CC2NE	RW	0x0	输入/捕获 2 互补输出使能 (Capture/Compare 2

				complementary output enable) 参考 CC1NE 的描述。
5	CC2P	RW	0x0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0x0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0x0	输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N 高电平有效; 1: OC1N 低电平有效。 CC1 通道配置为输入: 该位与 CC1P 结合使用以定义 TI1FP1 和 TI2FP1 的极性。参考 CC1P 的描述。 注: 一旦 LOCK 级别 (TIM2_BDTR 寄存器中的 LOCK 位) 设为 3 或 2 且 CC1S=00 (通道配置为输出) 则该位不能被修改。
2	CC1NE	RW	0x0	输入/捕获 1 互补输出使能 (Capture/Compare 1 complementary output enable) 0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	RW	0x0	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: CC1NP/CC1P 位选择 TI1FP1 和 TI2FP1 的有效极性, 用于触发或捕获操

				作。 00: 不反相/上升沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿，在门控模式或编码器模式下触发操作，TIxFP1 不反相。 01: 反向/下降沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的下降沿，在门控模式或编码器模式下触发操作，TIxFP1 反相。 10: 保留，不使用此配置。 11: 不反相/双边沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿和下降沿，在门控模式下触发操作，TIxFP1 不反相（此配置不得在编码器模式下使用） 注：一旦 LOCK 级别(TIM2_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	RW	0x0	输入/捕获 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 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启— OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIM2_CCR1 寄存器。 0: 捕获禁止 1: 捕获使能

表 11-1 互补输出通道 OCx 和 OCxN 的控制

控制位					输出状态	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态

1	X	0	0	0	输出禁止（与定时器断开） OCx=0, OCx_EN=0	输出禁止（与定时器断开） OCxN=0, OCxN_EN=0
		0	0	1	输出禁止（与定时器断开） OCx=0, OCx_EN=0	OCxREF + 极性, OCxN=OCxREF $\oplus$ CCxNP OCxN_EN=1
		0	1	0	OCxREF + 极性, OCx=OCxREF $\oplus$ CCxP OCx_EN=1	输出禁止（与定时器断开） OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
		1	0	0	输出禁止（与定时器断开） OCx=CCxP, OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP, OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP, OCx_EN=1	OCxREF + 极性, OCxN=OCxREF $\oplus$ CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性, OCx=OCxREF $\oplus$ CCxP, OCxN_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP, OCx_EN=1
		1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
0	0	0	0	0	输出禁止（与定时器断开） OCx=CCxP, OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP, OCxN_EN=0
	0	0	0	1	输出禁止（与定时器断开）	异步地：OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0, 若时钟存在：经过一个死区时间后，OCx=OISx, OCxN=OISx, 假设 OISx 和 OISxN 并不都对应 OCx 和 OCxN 的有效电平
	0	0	1	0		
	0	X	1	1		
	1	0	0	0	输出禁止（与定时器断开） OCx=CCxP, OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP, OCxN_EN=0
	1	0	1	0	关闭状态（输出使能且为无效电平）	异步地：OCx=CCxP, OCx_EN=1, OCxN=CCxNP,
	1	1	1	0		

	1		1	1	OCxN_EN=1, 若时钟存在：经过一个死区时间后 OCx=OISx , OCxN=OISxN, 假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。

11.4.10. TIM2 计数器 (TIM2_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x00000000	计数器的值 (Counter value)

11.4.11. TIM2 预分频器 (TIM2_PSC 偏移: 0x28)

比特	名称	属性	复位值	描述
15:0	PSC	RW	0x0	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ 。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清'0'或被工作在复位模式的从控制器清'0'。

11.4.12. TIM2 自动加载寄存器 (TIM2_ARR 偏移: 0x2c)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0	自动重装载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重装载寄存器的值。当自动重装载的值为空时，计数器不工作。

11.4.13. TIM2 捕获/比较寄存器 1 (TIM2_CCR1 偏移: 0x34)

比特	名称	属性	复位值	描述
15:0	CCR1	RW	0x0	捕获/比较通道 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出: CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。如果在 TIM2_CCMR1 寄存器(OC1PE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1 寄存器中。当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。

11.4.14. TIM2 捕获/比较寄存器 2 (TIM2_CCR2 偏移: 0x38)

比特	名称	属性	复位值	描述
15:0	CCR2	RW	0x0	捕获/比较通道 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出: CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。如果在 TIM2_CCMR2 寄存器(OC2PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 2 寄存器中。当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入: CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

11.4.15. TIM2 捕获/比较寄存器 3 (TIM2_CCR3 偏移: 0x3C)

比特	名称	属性	复位值	描述
15:0	CCR3	RW	0x0	捕获/比较通道 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出: CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。如果在 TIM2_CCMR3 寄存器(OC3PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 3 寄存器中。当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入: CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。

11.4.16. TIM2 捕获/比较寄存器 4 (TIM2_CCR4 偏移: 0x40)

比特	名称	属性	复位值	描述
15:0	CCR4	RW	0x0	捕获/比较通道 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出: CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。如果在 TIM2_CCMR4 寄存器(OC4PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 4 寄存器中。当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC4 端口上产生输出信号。若 CC4 通道配置为输入: CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。

11.4.17. TIM2 DMA 控制寄存器 (TIM2_DCR 偏移: 0x48)

比特	名称	属性	复位值	描述
15:13	-	-	-	位 15:13 保留, 始终读为 0。
12:8	DBL	RW	0x0	DMA 连续传送长度 (DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIM2_DMAR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的次数, 传输可以是半字(双字节)或字节: 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 10001: 18 次传输 例: 我们考虑这样的传输: DBL=7, DBA=TIM2_CR1 - 如果 DBL=7, DBA=TIM2_CR1 表示待传输数据的地址, 那么传输的地址由下式给出: (TIM2_CR1 的地址) + DBA + (DMA 索引), 其中 DMA 索引 = DBL 其中(TIM2_CR1 的地址) + DBA 再加上 7, 给出了将要写入或者读出数据的地址, 这样数据的传输将发生在从地址(TIM2_CR1 的地址) + DBA 开始的 7 个寄存器。 根据 DMA 数据长度的设置, 可能发生以下情况: - 如果设置数据为半字(16 位), 那么数据就会传输给全部 7 个寄存器。 - 如果设置数据为字节, 数据仍然会传输给全部 7 个寄存器: 第一个寄存器包含第一个 MSB 字节, 第二个寄存器包含第一个 LSB 字节,

				以此类推。因此对于定时器，用户必须指定由 DMA 传输的数据宽度。
7:5	-	-	-	位 7:5 保留，始终读为 0。
4:0	DBA	RW	0x0	这些位定义了 DMA 在连续模式下的基地址(当对 TIM2_DMAR 寄存器进行读或写时)，DBA 定义为从 TIM2_CR1 寄存器所在地址开始的偏移量： 00000: TIM2_CR1， 00001: TIM2_CR2， 00010: TIM2_SMCR， ...

11.4.18. TIM2 连续模式的 DMA 地址 (TIM2_DMAR 偏移: 0x4C)

比特	名称	属性	复位值	描述
31:0	DMAB	RW	0x0	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIM2_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作： TIM2_CR1 地址 + DBA + DMA 索引，其中：“TIM2_CR1 地址”是控制寄存器 1(TIM2_CR1)所在的地址；“DBA”是 TIM2_DCR 寄存器中定义的基地址；“DMA 索引”是由 DMA 自动控制的偏移量，它的最大值取决于 TIM2_DCR 寄存器中定义的 DBL。

11.4.19. TIM2 复用功能选择寄存器 1 (TIM2_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
17:14	ETRSEL	RW	0000	ETR 输入源选择 0000: ETR0

				0001: ETR1 0010: ETR2 0011: ETR3 0100: ETR4 0101: ETR5 0110: ETR6 0111: ETR7 其他: 保留 详情请参考 定时器外部触发源 (ETR) 互连关系
13:0	-	-	-	保留, 始终为 0。

11.4.20. TIM2 复用功能选择寄存器 2 (TIM2_AF2 偏移: 0x64)

比特	名称	属性	复位值	描述
18:16	OCRSEL[2:0]	RW	0x0	ocref_clr 源选择 这些位选择 ocref_clr 输入源。 000: tim_ocref_clr0 001: tim_ocref_clr1 010: tim_ocref_clr2 011: tim_ocref_clr3 100: tim_ocref_clr4 101: tim_ocref_clr5 110: tim_ocref_clr6 111: tim_ocref_clr7 详情请参考 OCxREF 清除输入信号互连关系
15:0	-	-	-	保留, 始终为 0。

11.4.21. TIM2 输入选择寄存器 (TIM2_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
31:28	-	-	-	保留, 始终为 0。
27:24	TI4SEL	RW	0x0	TI4 输入选择 0000: tim_ti4_in0 0001: tim_ti4_in1 0010: tim_ti4_in2 0011: tim_ti4_in3 ... 1111: tim_ti4_in15 详情请参考定时器输入信号互连关系
23:20	-	-	-	保留, 始终为 0。
19:16	TI3SEL	RW	0x0	TI3 输入选择 0000: tim_ti3_in0 0001: tim_ti3_in1 0010: tim_ti3_in2 0011: tim_ti3_in3 ... 1111: tim_ti3_in15 详情请参考定时器输入信号互连关系
15:12	-	-	-	保留, 始终为 0。
11:8	TI2SEL	RW	0x0	TI2 输入选择 0000: tim_ti2_in0 0001: tim_ti2_in1 0010: tim_ti2_in2 0011: tim_ti2_in3 ... 1111: tim_ti2_in15

				详情请参考定时器输入信号互连关系
7:4	-	-	-	保留，始终为 0。
3:0	TI1SEL	RW	0x0	TI1 输入选择 0000: tim_til_in0 0001: tim_til_in1 0010: tim_til_in2 0011: tim_til_in3 ... 1111: tim_til_in15 详情请参考定时器输入信号互连关系

11.4.22. TIM2 DMA 请求类型选择寄存器 (TIM2_DBER 偏移: 0x6C)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留，始终读为 0。
6	TBE	RW	0x0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0x0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;
4	CC4BE	RW	0x0	捕获/比较 4 事件的 DMA 请求类型 0: Single; 1: Burst;
3	CC3BE	RW	0x0	捕获/比较 3 事件的 DMA 请求类型 0: Single; 1: Burst;
2	CC2BE	RW	0x0	捕获/比较 2 事件的 DMA 请求类型 0: Single;

				1: Burst;
1	CC1BE	RW	0x0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0x0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

12. 通用定时器（TIM3/TIM4）

12.1. 概述

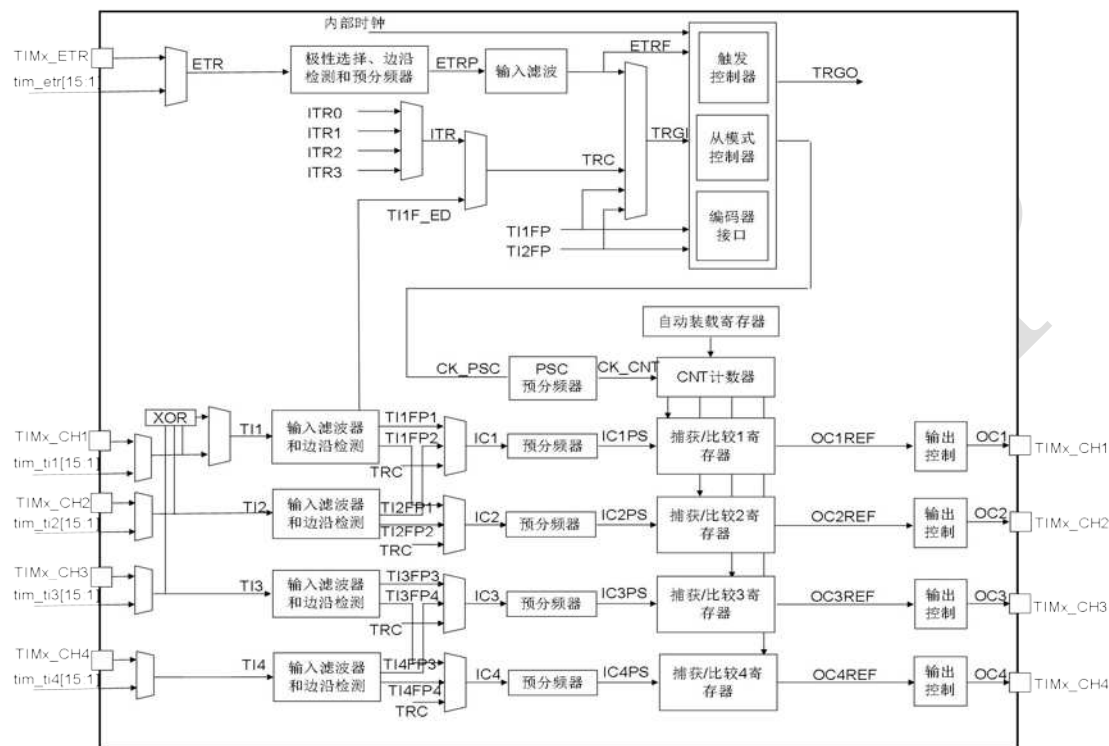
通用定时器 TIM3/TIM4 由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM 等)。

12.2. 主要特性

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值
- 多达 4 个独立通道：
 - 输入捕获
 - 输出比较
 - PWM 生成(边沿或中间对齐模式)
 - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 如下事件发生时产生中断/DMA：
 - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
 - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

12.3. 结构框图

图 12-1 通用定时器 TIM3/TIM4 框图



12.4. 功能描述

12.4.1. 计数单元

请参考 TIM1 相应章节。

12.4.2. 预分频器

请参考 TIM1 相应章节。

12.4.3. 时钟源选择

请参考 TIM1 相应章节。

12.4.4. 捕获比较通道

请参考 TIM1 相应章节。

12.4.5. 强制输出模式

请参考 TIM1 相应章节。

12.4.6. PWM 模式

请参考 TIM1 相应章节。

12.4.7. 正交编码器模式

请参考 TIM1 相应章节。

12.4.8. 与霍尔传感器的接口

请参考 TIM1 相应章节。

12.4.9. 单脉冲模式

请参考 TIM1 相应章节。

12.4.10. 定时器互连

请参考 TIM1 相应章节。

12.4.11. DMA 功能

请参考 TIM1 相应章节。

12.4.12. 外部事件清除 OCxREF 信号

请参考 TIM1 相应章节。

12.5. 寄存器描述

寄存器基地址：

TIM3: 0x4000_0400

TIM4: 0x4000_0800

偏置	名称	描述
0x00	TIMx_CR1	TIMx 控制寄存器 1
0x04	TIMx_CR2	TIMx 控制寄存器 2
0x08	TIMx_SMCR	TIMx 从模式控制寄存器
0x0C	TIMx_DIER	TIMx DMA/中断使能寄存器
0x10	TIMx_SR	TIMx 状态寄存器
0x14	TIMx_EGR	TIMx 事件产生寄存器
0x18	TIMx_CCMR1	TIMx 捕获/比较模式寄存器 1
0x1C	TIMx_CCMR2	TIMx 捕获/比较模式寄存器 2
0x20	TIMx_CCER	TIMx 捕获/比较使能寄存器
0x24	TIMx_CNT	TIMx 计数器
0x28	TIMx_PSC	TIMx 预分频器
0x2C	TIMx_ARR	TIMx 自动装载寄存器
0x34	TIMx_CCR1	TIMx 捕获比较寄存器 1
0x38	TIMx_CCR2	TIMx 捕获比较寄存器 2
0x3C	TIMx_CCR3	TIMx 捕获比较寄存器 3
0x40	TIMx_CCR4	TIMx 捕获比较寄存器 4
0x48	TIMx_DCR	TIMx DMA 控制寄存器
0x4C	TIMx_DMAR	TIMx 连续模式的 DMA 地址
0x60	TIMx_AF1	TIMx 复用功能选择寄存器 1
0x64	TIMx_AF2	TIMx 复用功能选择寄存器 2
0x68	TIMx_TISEL	TIMx 输入选择寄存器
0x6C	TIMx_DBER	TIMx DMA 请求类型选择寄存器

12.5.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15: 10	-	-	-	保留，始终读为 0。
9:8	CKD	RW	0x0	时钟分频因子 死区发生器和数字滤波器所用的采样时钟(Tdts)与定时器时钟 (CK_INT) 的分频比例。 00: tDTS=tCK_INT 01: tDTS=2 x tCK_INT 10: tDTS=4 x tCK_INT 11:保留
7	ARPE	RW	0x0	自动重装载预装载允许位 0:TIMx_ARR 寄存器没有缓冲 1:TIMx_ARR 寄存器被装入缓冲器
6:5	CMS	RW	0x0	计数模式 00:边沿对齐模式，计数器根据方向位 (DIR) 向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。

				注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	RW	0x0	方向控制位 0：计数器向上计数； 1：计数器向下计数。 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	RW	0x0	单脉冲模式 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止。
2	URS	RW	0x0	更新请求源 软件通过该位选择 UEV 事件的源 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0x0	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新(UEV)事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器) 1：禁止 UEV。不产生更新事件，影子寄存器(ARR、

				PSC、CCR _x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	RW	0x0	使能计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

12.5.2. TIM_x 控制寄存器 2 (TIM_x_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
23:8	-	-	-	保留。
7	TIIS	RW	0x0	TI1 选择 (TI1 selection) 0: TIM_x_CH1 引脚连到 TI1 输入; 1: TIM_x_CH1、TIM_x_CH2 和 TIM_x_CH3 引脚经异或后连到 TI1 输入。
6:4	MMS	RW	0x0	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: 000: 复位 – TIM_x_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控

				模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	CCDS	RW	0x0	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。
2	CCUS	RW	0x0	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们； 1: 如果捕获/比较控制位是预装载的(CCPC=1)，

				可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。注：该位只对具有互补输出的通道起作用。
1	-	-	-	保留，始终读为 0。
0	CCPC	RW	0x0	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位不是预装载的； 1: CCxE, CCxNE 和 OCxM 位是预装载的； 设置该位后，它们只在设置了 COM 位后被更新。

12.5.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)

比特	名称	属性	复位值	描述
21:20	TS[4:3]	RW	0x0	触发选择信号
19:16	-	-	-	保留，始终读为 0。
15	ETP	RW	0x0	外部触发极性 (External trigger polarity) 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相，高电平或上升沿有效； 1: ETR 被反相，低电平或下降沿有效。
14	ECE	RW	0x0	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2； 1: 使能外部时钟模式 2。 计数器由 ETRF 信号上的任意有效边沿驱动。 注 1：设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2：下述从模式可以与外部时钟模式 2 同时

				使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是'111')。 注 3：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。
13:12	ETPS	RW	0x0	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 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。
11:8	ETF	RW	0x0	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。0000：无滤波器，以 fDTS 采样 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，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 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$ 1110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 0111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$ 1111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$
7	MSM	RW	0x0	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。
6:4	TS[2:0]	RW	0x0	触发选择 (Trigger selection) , 和 TS[4:3]组成 5 位的 TS 信号。 这 5 位选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) 00100: TI1 的边沿检测器(TI1F_ED) 00001: 内部触发 1(ITR1) 00101: 滤波后的定时器输入 1(TI1FP1) 00010: 内部触发 2(ITR2) 00110: 滤波后的定时器输入 2(TI2FP2) 00011: 内部触发 3(ITR3) 00111: 外部触发输入(ETRF) 01000: 内部触发 4(ITR4) 01001: 内部触发 5(ITR5) 01010: 内部触发 6(ITR6) 01011: 内部触发 7(ITR7) 01100: 内部触发 8(ITR8) 01101: 内部触发 9(ITR9) 01110: 内部触发 10(ITR10) 其它: 保留

				注：这些位只能在未用到(如 SMS=000)时被改变，以避免在改变时产生错误的边沿检测。 内部触发源 ITRx 详情请参考<u>定时器内部触发源（ITR）互连关系</u>。
3	OCCS	RW	0x0	OCREF 清零选择（OCREF clear selection） 该位用于选择 OCREF 清零源。 0: OCREF_CLR_INT 连接到 COMPx 输出，具体取决于 TIMx_AF2. OCRSEL 1: OCREF_CLR_INT 连接到 ETRF
2:0	SMS	RW	0x0	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 CEN=1，则预分频器直接由内部时钟驱动。 001: 编码器模式 1 – 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。 010: 编码器模式 2 – 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。 011: 编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101: 门控模式 – 当触发输入 (TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 – 计数器在触发输入 TRGI 的

				上升沿启动(但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式 1 – 选中的触发输入(TRGI) 的上升沿驱动计数器。 注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。

12.5.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)

比特	名称	属性	复位值	描述
15	-	-	-	保留, 始终读为 0。
14	TDE	RW	0x0	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。
13	COMDE	RW	0x0	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。
12	CC4DE	RW	0x0	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。
11	CC3DE	RW	0x0	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。
10	CC2DE	RW	0x0	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。
9	CC1DE	RW	0x0	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。
8	UDE	RW	0x0	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。
7	-	-	-	保留, 始终读为 0。
6	TIE	RW	0x0	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
5	COMIE	RW	0x0	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断; 1: 允许 COM 中断。
4	CC4IE	RW	0x0	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。
3	CC3IE	RW	0x0	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。
2	CC2IE	RW	0x0	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。
1	CC1IE	RW	0x0	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt

				enable) 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。
0	UIE	RW	0x0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

12.5.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
17:13	-	-	-	保留, 始终读为 0。
12	CC4OF	W0C	0x0	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	W0C	0x0	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	W0C	0x0	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	W0C	0x0	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_CCR1 寄存器时, CC1IF 的状态已经为'1'。
8:7	-	-	-	保留, 始终读为 0。
6	TIF	W0C	0x0	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。
5	COMIF	W0C	0x0	COM 中断标记 (COM interrupt flag) 一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置'1'。它由软件清'0'。 0: 无 COM 事件产生； 1: COM 中断等待响应。
4	CC4IF	W0C	0x0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	W0C	0x0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	W0C	0x0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	W0C	0x0	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: 当计数器值与比较值匹配时该位由硬件置 1, 但在中心对称模式下除外(参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生； 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时, 在向上或向上/下计数模式时计数器溢出, 或向下计数模式时的计数器下溢条件下, CC1IF 位变高 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置'1', 它由软件清'0'或通过读 TIMx_CCR1 清'0'。 0: 无输入捕获产生；

				1: 计数器值已被捕获(拷贝)至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	W0C	0x0	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1': – 若 TIMx_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当计数器 CNT 被触发事件重新初始化时。

12.5.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留, 始终读为 0。
6	TG	WO	0x0	产生触发事件 (Trigger generation) 该位由软件置'1', 用于产生一个触发事件, 由硬件自动清'0'。 0: 无动作; 1: TIMx_SR 寄存器的 TIF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。
5	COMG	WO	0x0	捕获/比较事件, 产生控制更新 (Capture/Compare control update generation)

				该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。 注：该位只对拥有互补输出的通道有效。
4	CC4G	WO	0x0	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	WO	0x0	产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	WO	0x0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0x0	产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0: 无动作； 1: 在通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	WO	0x0	产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。 注意预分频器的计数器也被清'0' (但是预分频系数不变)。若在中心对称模式下或 DIR=0 (向上计数) 则计数器被清'0'；若 DIR=1 (向下计数) 则计数器取

				TIMx_ARR 的值。

12.5.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18)

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 CCxS 位定义输出比较模式:

比特	名称	属性	复位值	描述
15	OC2CE	RW	0x0	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	0x0	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0x0	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	RW	0x0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7	OC1CE	RW	0x0	输出比较 1 清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受 ETRF 输入的影响; 1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。
6:4	OC1M	RW	0x0	输出比较 1 模式 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作, 而

				OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效,而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用; 001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1)相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1)相同时, 强制 OC1REF 为低。 011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比

				较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	RW	0x0	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMx_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	RW	0x0	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S	RW	0x0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选

				择： 00： CC1 通道被配置为输出； 01： CC1 通道被配置为输入，IC1 映射在 TI1 上； 10： CC1 通道被配置为输入，IC1 映射在 TI2 上； 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注： CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式：

比特	名称	属性	复位值	描述
15:12	IC2F	RW	0x0	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	0x0	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注： CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	RW	0x0	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000： 无滤波器，以 fDTS 采样 1000： 采样频率 fSAMPLING=fDTS/8， N=6

				0001: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=2$ 1001: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=8$ 0010: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=4$ 1010: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=5$ 0011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK\_INT}}$, $N=8$ 1011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=6$ 0100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=6$ 1100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$ 0101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=8$ 1101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$ 0110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$ 1110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 0111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$ 1111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$
3:2	IC1PSC	RW	0x0	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 $\text{CC1E}=0$ (TIMx_CCER 寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每 2 个事件触发一次捕获; 10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获。
1:0	CC1S	RW	0x0	捕获/比较 1 选择 (Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 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器输入被选中时(由

				TIMx_SMCR 寄存器的 TS 位选择)。

12.5.8. TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2 偏移: 0x1C)

输出比较模式:

比特	名称	属性	复位值	描述
15	OC4CE	RW	0x0	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M	RW	0x0	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	RW	0x0	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	RW	0x0	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S	RW	0x0	捕获/比较 4 选择 (Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	RW	0x0	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M	RW	0x0	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	RW	0x0	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	RW	0x0	输出比较 3 快速使能 (Output compare 3 fast enable)
1:0	CC3S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC4F	RW	0x0	输入捕获 4 滤波器 (Input capture 4 filter)
11:10	IC4PSC	RW	0x0	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	RW	0x0	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	0x0	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。

12.5.9. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20)

比特	名称	属性	复位值	描述
21:16	-	-	-	保留, 始终读为 0。
15	CC4NP	RW	0x0	输入/捕获 4 互补输出极性 (Capture/Compare 4 complementary output polarity) 参考 CC1NP 的描述。
14	-	-	-	保留, 始终读为 0。
13	CC4P	RW	0x0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0x0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0x0	输入/捕获 3 互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。
10	CC3NE	RW	0x0	输入/捕获 3 互补输出使能 (Capture/Compare 3 complementary output enable) 参考 CC1NE 的描述。
9	CC3P	RW	0x0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0x0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0x0	输入/捕获 2 互补输出极性 (Capture/Compare 2 complementary output polarity) 参考 CC1NP 的描述。
6	CC2NE	RW	0x0	输入/捕获 2 互补输出使能 (Capture/Compare 2

				complementary output enable) 参考 CC1NE 的描述。
5	CC2P	RW	0x0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0x0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0x0	输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N 高电平有效; 1: OC1N 低电平有效。 CC1 通道配置为输入: 该位与 CC1P 结合使用以定义 TI1FP1 和 TI2FP1 的极性。参考 CC1P 的描述。 注: 一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 3 或 2 且 CC1S=00 (通道配置为输出) 则该位不能被修改。
2	CC1NE	RW	0x0	输入/捕获 1 互补输出使能 (Capture/Compare 1 complementary output enable) 0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	RW	0x0	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: CC1NP/CC1P 位选择 TI1FP1 和 TI2FP1 的有效极性, 用于触发或捕获操

				作。 00: 不反相/上升沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿，在门控模式或编码器模式下触发操作，TIxFP1 不反相。 01: 反向/下降沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的下降沿，在门控模式或编码器模式下触发操作，TIxFP1 反相。 10: 保留，不使用此配置。 11: 不反相/双边沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿和下降沿，在门控模式下触发操作，TIxFP1 不反相（此配置不得在编码器模式下使用） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	RW	0x0	输入/捕获 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 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启— OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 0: 捕获禁止 1: 捕获使能

12.5.10. TIMx 计数器 (TIMx_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0	计数器的值 (Counter value)

12.5.11. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)

比特	名称	属性	复位值	描述
15:0	PSC	RW	0x0	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$。 PSC 包含了每次当更新事件产生时, 装入当前预分频器寄存器的值; 更新事件包括计数器被 TIM_EGR 的 UG 位清'0'或被工作在复位模式的从控制器清'0'。

12.5.12. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0	自动重装载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重装载寄存器的值。 当自动重装载的值为空时, 计数器不工作。

12.5.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)

比特	名称	属性	复位值	描述
15:0	CCR1	RW	0x0	捕获/比较通道 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出: CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载

				功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入：CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。

12.5.14. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移：0x38)

比特	名称	属性	复位值	描述
15:0	CCR2	RW	0x0	捕获/比较通道 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出：CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。如果在 TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入：CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

12.5.15. TIMx 捕获/比较寄存器 3 (TIMx_CCR3 偏移：0x3C)

比特	名称	属性	复位值	描述
15:0	CCR3	RW	0x0	捕获/比较通道 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出：CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。如果在 TIMx_CCMR3 寄存器(OC3PE 位)中未选择预装载

				特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入：CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。

12.5.16. TIMx 捕获/比较寄存器 4 (TIMx_CCR4 偏移：0x40)

比特	名称	属性	复位值	描述
15:0	CCR4	RW	0x0	捕获/比较通道 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出：CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。如果在 TIMx_CCMR4 寄存器(OC4PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。若 CC4 通道配置为输入：CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。

12.5.17. TIMx DMA 控制寄存器 (TIMx_DCR 偏移：0x48)

比特	名称	属性	复位值	描述
15:13	-	-	-	位 15:13 保留，始终读为 0。
12:8	DBL	RW	0x0	DMA 连续传送长度 (DMA burst length)

				这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，即：定义传输的次数，传输可以是半字(双字节)或字节： 00000：1 次传输 00001：2 次传输 00010：3 次传输 10001：18 次传输 例：我们考虑这样的传输：DBL=7, DBA=TIMx_CR1 - 如果 DBL=7, DBA=TIMx_CR1 表示待传输数据的地址，那么传输的地址由下式给出：(TIMx_CR1 的地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL 其中(TIMx_CR1 的地址) + DBA 再加上 7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址(TIMx_CR1 的地址) + DBA 开始的 7 个寄存器。 根据 DMA 数据长度的设置，可能发生以下情况： - 如果设置数据为半字(16 位)，那么数据就会传输给全部 7 个寄存器。 - 如果设置数据为字节，数据仍然会传输给全部 7 个寄存器：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，以此类推。因此对于定时器，用户必须指定由 DMA 传输的数据宽度。
7:5	-	-	-	位 7:5 保留，始终读为 0。
4:0	DBA	RW	0x0	这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时)，DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量：

				00000: TIMx_CR1, 00001: TIMx_CR2, 00010: TIMx_SMCR, ...

12.5.18. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)

比特	名称	属性	复位值	描述
31:0	DMAB	RW	0x0	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIMx_CR1 地址 + DBA + DMA 索引, 其中: “TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1)所在的地址; “DBA”是 TIMx_DCR 寄存器中定义的基地址; “DMA 索引”是由 DMA 自动控制的偏移量, 它的最大值取决于 TIMx_DCR 寄存器中定义的 DBL。

12.5.19. TIMx 复用功能选择寄存器 1 (TIMx_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
17:14	ETRSEL	RW	0000	ETR 输入源选择 0000: ETR0 0001: ETR1 0010: ETR2 0011: ETR3 0100: ETR4 0101: ETR5 0110: ETR6

				0111: ETR7 其他: 保留 详情请参考 定时器外部触发源 (ETR) 互连关系
13:0	-	-	-	保留, 始终为 0。

12.5.20. TIMx 复用功能选择寄存器 2 (TIMx_AF2 偏移: 0x64)

比特	名称	属性	复位值	描述
18:16	OCRSEL[2:0]	RW	0x0	ocref_clr 源选择 这些位选择 ocref_clr 输入源。 000: tim_ocref_clr0 001: tim_ocref_clr1 010: tim_ocref_clr2 011: tim_ocref_clr3 100: tim_ocref_clr4 101: tim_ocref_clr5 110: tim_ocref_clr6 111: tim_ocref_clr7 详情请参考 OCxREF 清除输入信号互连关系
15:0	-	-	-	保留, 始终为 0。

12.5.21. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
31:28	-	-	-	保留, 始终为 0。
27:24	TI4SEL	RW	0x0	TI4 输入选择 0000: tim_ti4_in0 0001: tim_ti4_in1

				0010: tim_ti4_in2 0011: tim_ti4_in3 ... 1111: tim_ti4_in15 详情请参考 <u>定时器输入信号互连关系</u>
23:20	-	-	-	保留，始终为 0。
19:16	TI3SEL	RW	0x0	TI3 输入选择 0000: tim_ti3_in0 0001: tim_ti3_in1 0010: tim_ti3_in2 0011: tim_ti3_in3 ... 1111: tim_ti3_in15 详情请参考 <u>定时器输入信号互连关系</u>
15:12	-	-	-	保留，始终为 0。
11:8	TI2SEL	RW	0x0	TI2 输入选择 0000: tim_ti2_in0 0001: tim_ti2_in1 0010: tim_ti2_in2 0011: tim_ti2_in3 ... 1111: tim_ti2_in15 详情请参考 <u>定时器输入信号互连关系</u>
7:4	-	-	-	保留，始终为 0。
3:0	TI1SEL	RW	0x0	TI1 输入选择 0000: tim_ti1_in0 0001: tim_ti1_in1 0010: tim_ti1_in2 0011: tim_ti1_in3

				...
				1111: tim_til_in15
				详情请参考定时器输入信号互连关系

12.5.22. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留，始终读为 0。
6	TBE	RW	0x0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0x0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;
4	CC4BE	RW	0x0	捕获/比较 4 事件的 DMA 请求类型 0: Single; 1: Burst;
3	CC3BE	RW	0x0	捕获/比较 3 事件的 DMA 请求类型 0: Single; 1: Burst;
2	CC2BE	RW	0x0	捕获/比较 2 事件的 DMA 请求类型 0: Single; 1: Burst;
1	CC1BE	RW	0x0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0x0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

13. 基本定时器（TIM6/TIM7）

13.1. 概述

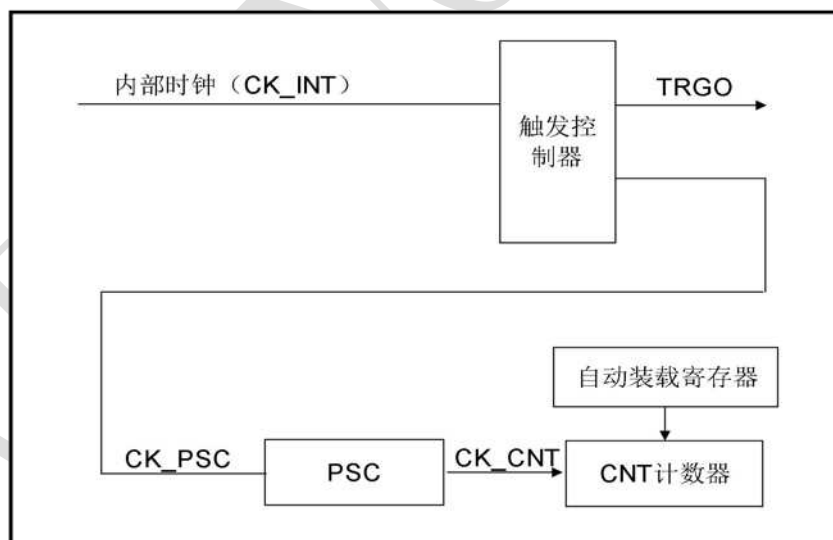
基本定时器包含一个 16 位自动装载计数器，由各自的可编程预分频器驱动。

13.2. 主要特性

- 16 位自动重装载累加计数器
- 16 位可编程(可实时修改)预分频器，用于对输入的时钟按系数为 1~65536 之间的任意数值分频
- 在更新事件(计数器溢出)时产生中断/DMA 请求

13.3. 结构框图

图 13-1 基本定时器结构框图



13.4. 功能描述

13.4.1. 计数单元

请参考 TIM1 相应章节。

13.4.2. 预分频器

请参考 TIM1 相应章节。

13.4.3. 时钟源选择

计数器的时钟由内部时钟(CK_INT)提供。具体请参考 TIM1 相应章节。

13.4.4. 定时器互连

请参考 TIM1 相应章节。

13.4.5. DMA 功能

请参考 TIM1 相应章节。

13.5. 寄存器描述

寄存器基地址：

TIM6: 0x4000_1000

TIM7: 0x4000_1400

偏置	名称	描述
0x00	TIMx_CR1	TIMx 控制寄存器 1
0x04	TIMx_CR2	TIMx 控制寄存器 2
0x0C	TIMx_DIER	TIMx DMA/中断使能寄存器
0x10	TIMx_SR	TIMx 状态寄存器
0x14	TIMx_EGR	TIMx 事件产生寄存器
0x24	TIMx_CNT	TIMx 计数器
0x28	TIMx_PSC	TIMx 预分频器
0x2C	TIMx_ARR	TIMx 自动装载寄存器

13.5.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15:8	-	-	-	保留, 始终读为 0。
7	ARPE	RW	0x0	自动重装载预装载允许位 0:TIMx_ARR 寄存器没有缓冲 1:TIMx_ARR 寄存器被装入缓冲器
6:4	-	-	-	保留, 始终读为 0。
3	OPM	RW	0x0	单脉冲模式 0: 在发生更新事件时, 计数器不停止; 1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。
2	URS	RW	0x0	更新请求源 软件通过该位选择 UEV 事件的源 0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0x0	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器)

				1: 禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、CCR _x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	RW	0x0	使能计数器 0: 禁止计数器； 1: 使能计数器。 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

13.5.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
23:7	-	-	-	保留。
6:4	MMS	RW	0x0	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000: 复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见

				TIMx_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 其他: 保留。
3:0	-	-	-	保留，始终读为 0。

13.5.3. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)

比特	名称	属性	复位值	描述
15:9	-	-	-	保留，始终读为 0。
8	UDE	RW	0x0	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。
7:1	-	-	-	保留，始终读为 0。
0	UIE	RW	0x0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

13.5.4. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
17:1	-	-	-	保留，始终读为 0。
0	UIF	W0C	0x0	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1'。

				– 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。

13.5.5. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)

比特	名称	属性	复位值	描述
15:1	-	-	-	保留，始终读为 0。
0	UG	WO	0x0	产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。 注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'；若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。

13.5.6. TIMx 计数器 (TIMx_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0	计数器的值 (Counter value)

13.5.7. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)

比特	名称	属性	复位值	描述
15:0	PSC	RW	0x0	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK\_PSC}/(PSC[15:0]+1)$ 。 PSC 包含了每次当更新事件产生时, 装入当前预分频器寄存器的值; 更新事件包括计数器被 TIM_EGR 的 UG 位清'0'或被工作在复位模式的从控制器清'0'。

13.5.8. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0	自动重载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重载寄存器的值。 当自动重载的值为空时, 计数器不工作。

14. 通用定时器（TIM15/TIM16/TIM17）

14.1. 概述

通用定时器由 TIM15 一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。使用定时器预分频器和系统时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

14.2. 主要特性

- 16 位向上自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值
- 1 个/2 个独立通道（TIM15 两个通道，TIM16 和 TIM17 各一个通道）：
 - 输入捕获
 - 输出比较
 - PWM 生成
 - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA：
 - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
 - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
 - 刹车信号输入
- 触发输入作为外部时钟

14.3. 结构框图

图 14-1 通用定时器 TIM15 结构框图

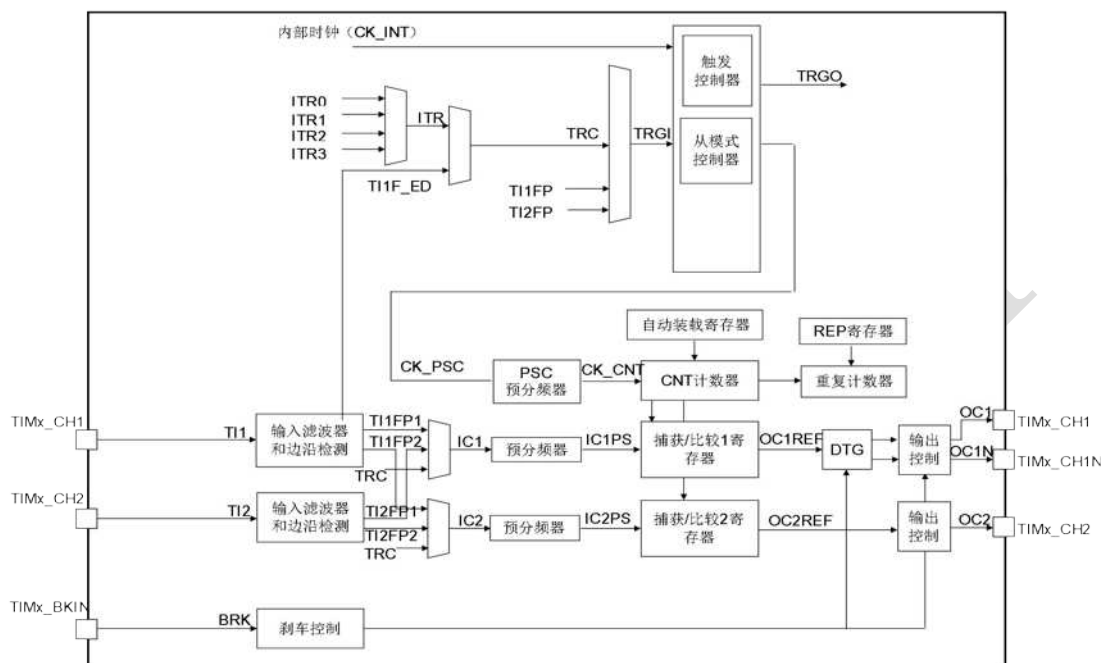
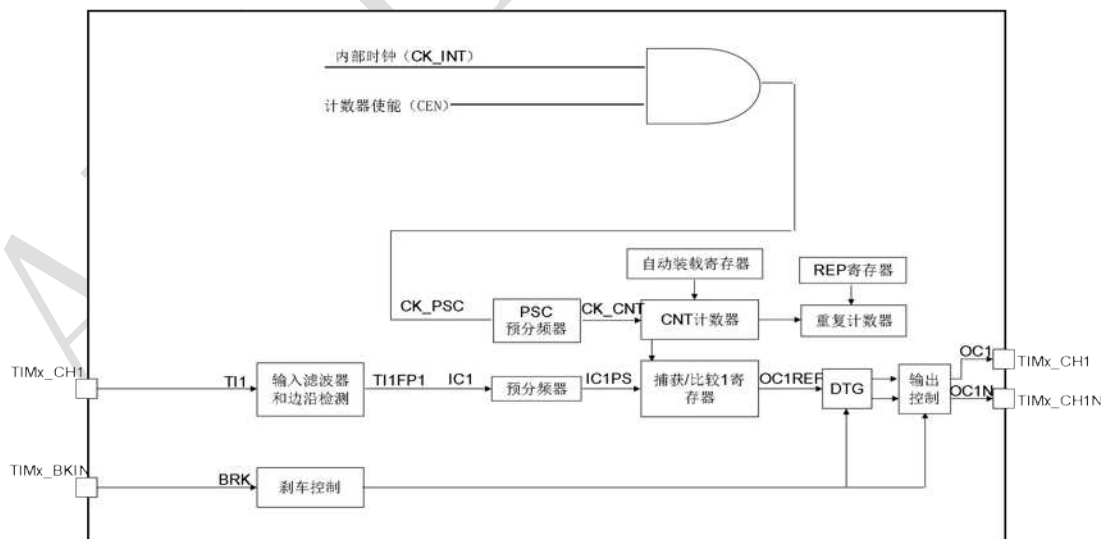


图 14-2 通用定时器 TIM16/TIM17 结构框图



14.4. 功能描述

14.4.1. 计数单元

请参考 TIM1 相应章节。

14.4.2. 预分频器

请参考 TIM1 相应章节。

14.4.3. 时钟源选择

请参考 TIM1 相应章节。

14.4.4. 捕获比较通道

请参考 TIM1 相应章节。

14.4.5. 强制输出模式

请参考 TIM1 相应章节。

14.4.6. PWM 模式

请参考 TIM1 相应章节。

14.4.7. 互补输出和死区插入

请参考 TIM1 相应章节。

14.4.8. 刹车功能

请参考 TIM1 相应章节。

14.4.9. 六步 PWM 输出

请参考 TIM1 相应章节。

14.4.10. 单脉冲模式

请参考 TIM1 相应章节。

14.4.11. 定时器互连

请参考 TIM1 相应章节。

14.4.12. DMA 功能

请参考 TIM1 相应章节。

14.4.13. 外部事件清除 OCxREF 信号

请参考 TIM1 相应章节。

14.5. 寄存器描述

寄存器基地址：

TIM15: 0x4001_4000

TIM16: 0x4001_4400

TIM17: 0x4001_4800

偏置	名称	描述
0x00	TIMx_CR1	TIMx 控制寄存器 1
0x04	TIMx_CR2	TIMx 控制寄存器 2
0x08	TIMx_SMCR	TIMx 从模式控制寄存器
0x0C	TIMx_DIER	TIMx DMA/中断使能寄存器
0x10	TIMx_SR	TIMx 状态寄存器
0x14	TIMx_EGR	TIMx 事件产生寄存器
0x18	TIMx_CCMR1	TIMx 捕获/比较模式寄存器 1
0x20	TIMx_CCER	TIMx 捕获/比较使能寄存器
0x24	TIMx_CNT	TIMx 计数器
0x28	TIMx_PSC	TIMx 预分频器
0x2C	TIMx_ARR	TIMx 自动装载寄存器
0x30	TIMx_RCR	TIMx 重复计数寄存器
0x34	TIMx_CCR1	TIMx 捕获比较寄存器 1
0x38	TIMx_CCR2	TIMx 捕获比较寄存器 2
0x44	TIMx_BDTR	TIMx 刹车和死区控制寄存器
0x48	TIMx_DCR	TIMx DMA 控制寄存器
0x4C	TIMx_DMAR	TIMx 连续模式的 DMA 地址
0x60	TIMx_AF1	TIMx 复用功能选择寄存器 1
0x64	TIMx_AF2	TIMx 复用功能选择寄存器 2
0x68	TIMx_TISEL	TIMx 输入选择寄存器
0x6C	TIMx_DBER	TIMx DMA 请求类型选择寄存器

14.5.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15: 14	-	-	-	保留, 始终读为 0。
13:10	BKF	RW	0x0	刹车滤波器 (Break filter) 这几位定义了用于刹车输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器, 以 fDTS 采样 1000: 采样频率 fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 1001: 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010: 采样频率 fSAMPLING=fDTS/16, N=5 0011: 采样频率 fSAMPLING=fCK_INT, N=8 1011: 采样频率 fSAMPLING=fDTS/16, N=6 0100: 采样频率 fSAMPLING=fDTS/2, N=6 1100: 采样频率 fSAMPLING=fDTS/16, N=8 0101: 采样频率 fSAMPLING=fDTS/2, N=8 1101: 采样频率 fSAMPLING=fDTS/32, N=5 0110: 采样频率 fSAMPLING=fDTS/4, N=6 1110: 采样频率 fSAMPLING=fDTS/32, N=6 0111: 采样频率 fSAMPLING=fDTS/4, N=8 1111: 采样频率 fSAMPLING=fDTS/32, N=8
9:8	CKD	RW	0x0	时钟分频因子 死区发生器和数字滤波器所用的采样时钟(Tdts)与定时器时钟 (CK_INT) 的分频比例。 00: tDTS=tCK_INT 01: tDTS=2 x tCK_INT

				10: tDTS=4 x tCK_INT 11:保留
7	ARPE	RW	0x0	自动重装载预装载允许位 0:TIMx_ARR 寄存器没有缓冲 1:TIMx_ARR 寄存器被装入缓冲器
6:5	CMS	RW	0x0	计数模式 00:边沿对齐模式，计数器根据方向位（DIR）向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	RW	0x0	方向控制位 0: 计数器向上计数； 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	RW	0x0	单脉冲模式 0: 在发生更新事件时，计数器不停止；

				1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。
2	URS	RW	0x0	更新请求源 软件通过该位选择 UEV 事件的源 0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	RW	0x0	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、CCR_x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。
0	CEN	RW	0x0	使能计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

14.5.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
23:11	-	-	-	保留，始终读为 0。
10	OIS2	RW	0x0	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	RW	0x0	输出空闲状态 1(OC1N 输出) (Output Idle state 1) 0: 当 MOE=0 时，死区后 OC1N=0; 1: 当 MOE=0 时，死区后 OC1N=1。 注：已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
8	OIS1	RW	0x0	输出空闲状态 1(OC1 输出) (Output Idle state 1) 0: 当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=0; 1: 当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=1。 注：已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
7	-	-	-	保留，始终读为 0。
6:4	MMS	RW	0x0	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000: 复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时

				器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 其他: 保留。
3	CCDS	RW	0x0	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。
2	CCUS	RW	0x0	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们； 1: 如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注：该位只对具有互补输出的通道

				起作用。
1	-	-	-	保留，始终读为 0。
0	CCPC	RW	0x0	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位不是预装载的； 1: CCxE, CCxNE 和 OCxM 位是预装载的； 设置该位后，它们只在设置了 COM 位后被更新。

14.5.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)

比特	名称	属性	复位值	描述
21:20	TS[4:3]	RW	0x0	触发选择信号
19:8	-	-	-	保留，始终读为 0。
7	MSM	RW	0x0	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。
6:4	TS[2:0]	RW	0x0	触发选择 (Trigger selection) 这 5 位选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) 00100: TI1 的边沿检测器(TI1F_ED) 00001: 内部触发 1(ITR1) 00101: 滤波后的定时器输入 1(TI1FP1) 00010: 内部触发 2(ITR2) 00110: 滤波后的定时器输入 2(TI2FP2) 00011: 内部触发 3(ITR3)

				01000: 内部触发 4(ITR4) 01001: 内部触发 5(ITR5) 01010: 内部触发 6(ITR6) 01011: 内部触发 7(ITR7) 01100: 内部触发 8(ITR8) 01101: 内部触发 9(ITR9) 01110: 内部触发 10(ITR10) 其它: 保留 注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。 内部触发源 ITR_x 详情请参考<u>定时器内部触发源 (ITR) 互连关系</u>。
3	OCCS	RW	0x0	OCREF 清零选择 (OCREF clear selection) 该位用于选择 OCREF 清零源。 0: OCREF_CLR_INT 连接到 COMP_x 输出, 具体取决于 TIM_x_AF2. OCRSEL 1: 保留。
2:0	SMS	RW	0x0	从模式选择 (Slave mode selection), 和 TS[4:3] 组成 5 位的 TS 信号。 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 CEN=1, 则预分频器直接由内部时钟驱动。 100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。 101: 门控模式 – 当触发输入 (TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数

				器的启动和停止都是受控的。 110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。 其他: 保留。

14.5.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)

比特	名称	属性	复位值	描述
15	-	-	-	保留, 始终读为 0。
14	TDE	RW	0x0	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。
13:11	-	-	-	保留, 始终读为 0。
10	CC2DE	RW	0x0	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。
9	CC1DE	RW	0x0	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。
8	UDE	RW	0x0	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。
7	BIE	RW	0x0	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。
6	TIE	RW	0x0	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
5	COMIE	RW	0x0	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断; 1: 允许 COM 中断。
4:3	-	-	-	保留, 始终读为 0。
2	CC2IE	RW	0x0	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。
1	CC1IE	RW	0x0	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。
0	UIE	RW	0x0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

14.5.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
17:11	-	-	-	保留, 始终读为 0。
10	CC2OF	W0C	0x0	捕获/比较 2 重复捕获标记 (Capture/Compare 2

				overcapture flag) 参见 CC10F 描述。
9	CC10F	W0C	0x0	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_CCR1 寄存器时，CC1IF 的状态已经为'1'。
8	-	-	-	位 8 保留，始终读为 0。
7	BIF	W0C	0x0	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。
6	TIF	W0C	0x0	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。
5	COMIF	W0C	0x0	COM 中断标记 (COM interrupt flag) 一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置'1'。它由软件清'0'。 0: 无 COM 事件产生； 1: COM 中断等待响应。
4:3	-	-	-	保留，始终读为 0。
2	CC2IF	W0C	0x0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt

				flag) 参考 CC1IF 描述。
1	CC1IF	W0C	0x0	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： 当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外(参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清'0'。 0：无匹配发生； 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1IF 位变高 如果通道 CC1 配置为输入模式： 当捕获事件发生时该位由硬件置'1'，它由软件清'0'或通过读 TIMx_CCR1 清'0'。 0：无输入捕获产生； 1：计数器值已被捕获(拷贝)至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	W0C	0x0	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置'1'： – 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。

				– 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。

14.5.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)

比特	名称	属性	复位值	描述
15:8	-	-	-	位 15:8 保留，始终读为 0。
7	BG	WO	0x0	产生刹车事件 (Break generation) 该位由软件置'1'，用于产生一个刹车事件，由硬件自动清'0'。 0: 无动作; 1: 产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
6	TG	WO	0x0	产生触发事件 (Trigger generation) 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0: 无动作; 1: TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
5	COMG	WO	0x0	捕获/比较事件，产生控制更新 (Capture/Compare control update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作; 1: 当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。 注：该位只对拥有互补输出的通道有效。
4:3	-	-	-	保留，始终读为 0。
2	CC2G	WO	0x0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。

1	CC1G	WO	0x0	产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0：无动作； 1：在通道 CC1 上产生一个捕获/比较事件：若通道 CC1 配置为输出：设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若通道 CC1 配置为输入：当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	WO	0x0	产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。 0：无动作； 1：重新初始化计数器，并产生一个更新事件。 注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'；若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。

14.5.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移：0x18)

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 CCxS 位定义输出比较模式：

比特	名称	属性	复位值	描述
15	-	-	-	保留，始终读为 0。
14:12	OC2M	RW	0x0	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0x0	输出比较 2 预装载使能 (Output Compare 2 preload)

				enable)
10	OC2FE	RW	0x0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7	-	-	-	保留, 始终读为 0。
6:4	OC1M	RW	0x0	输出比较 1 模式 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用; 001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1)相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1)相同时, 强制 OC1REF 为低。 011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。

				101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 $TIMx\_CNT < TIMx\_CCR1$ 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 $TIMx\_CNT > TIMx\_CCR1$ 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 $TIMx\_CNT < TIMx\_CCR1$ 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 $TIMx\_CNT > TIMx\_CCR1$ 时通道 1 为有效电平, 否则为无效电平。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。
3	OC1PE	RW	0x0	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能, 可随时写入 TIMx_CCR1 寄存器, 并且新写入的数值立即起作用。 1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMx_CR1 寄存器的

				OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	RW	0x0	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S	RW	0x0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC2F	RW	0x0	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	0x0	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	0x0	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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	RW	0x0	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器, 以 fDTS 采样 1000: 采样频率 fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 1001: 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010: 采样频率 fSAMPLING=fDTS/16, N=5 0011: 采样频率 fSAMPLING=fCK_INT, N=8 1011: 采样频率 fSAMPLING=fDTS/16, N=6 0100: 采样频率 fSAMPLING=fDTS/2, N=6 1100: 采样频率 fSAMPLING=fDTS/16, N=8 0101: 采样频率 fSAMPLING=fDTS/2, N=8 1101: 采样频率 fSAMPLING=fDTS/32, N=5 0110: 采样频率 fSAMPLING=fDTS/4, N=6 1110: 采样频率 fSAMPLING=fDTS/32, N=6 0111: 采样频率 fSAMPLING=fDTS/4, N=8 1111: 采样频率 fSAMPLING=fDTS/32, N=8

3:2	IC1PSC	RW	0x0	输入/捕获 1 预分频器 (Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMx_CCER 寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每 2 个事件触发一次捕获; 10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获。
1:0	CC1S	RW	0x0	捕获/比较 1 选择 (Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 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器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。

14.5.8. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20)

比特	名称	属性	复位值	描述
21:8	-	-	-	保留, 始终读为 0。
7	CC2NP	RW	0x0	输入/捕获 2 互补输出极性 (Capture/Compare 2 complementary output polarity) 参考 CC1NP 的描述。
6	-	-	-	保留, 始终读为 0。
5	CC2P	RW	0x0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0x0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。

3	CC1NP	RW	0x0	输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N 高电平有效; 1: OC1N 低电平有效。 CC1 通道配置为输入: 该位与 CC1P 结合使用以定义 TI1FP1 和 TI2FP1 的极性。参考 CC1P 的描述。 注: 一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 3 或 2 且 CC1S=00 (通道配置为输出) 则该位不能被修改。
2	CC1NE	RW	0x0	输入/捕获 1 互补输出使能 (Capture/Compare 1 complementary output enable) 0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	RW	0x0	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: CC1NP/CC1P 位选择 TI1FP1 和 TI2FP1 的有效极性, 用于触发或捕获操作。 00: 不反相/上升沿。在复位、外部时钟或触发模式下, 捕获或触发发生在 TIxFP1 的上升沿, 在门控模式或编码器模式下触发操作, TIxFP1 不反相。 01: 反向/下降沿。在复位、外部时钟或触发模式下,

				捕获或触发发生在 TIxFP1 的下降沿，在门控模式或编码器模式下触发操作，TIxFP1 反相。 10: 保留，不使用此配置。 11: 不反相/双边沿。在复位、外部时钟或触发模式下，捕获或触发发生在 TIxFP1 的上升沿和下降沿，在门控模式下触发操作，TIxFP1 不反相（此配置不得在编码器模式下使用） 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	RW	0x0	输入/捕获 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 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启— OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 0: 捕获禁止 1: 捕获使能

14.5.9. TIMx 计数器 (TIMx_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0	计数器的值 (Counter value)

14.5.10. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)

比特	名称	属性	复位值	描述
15:0	PSC	RW	0x0	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK\_PSC}/(PSC[15:0]+1)$。 PSC 包含了每次当更新事件产生时, 装入当前预分频器寄存器的值; 更新事件包括计数器被 TIM_EGR 的 UG 位清'0'或被工作在复位模式的从控制器清'0'。

14.5.11. TIMx 自动加载寄存器 (TIMx_ARR 偏移: 0x2c)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0	自动重载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重载寄存器的值。 当自动重载的值为空时, 计数器不工作。

14.5.12. TIMx 重复计数寄存器 (TIMx_RCR 偏移: 0x30)

比特	名称	属性	复位值	描述
15:8	-	-	-	位 15:8 保留, 始终读为 0。
7:0	REP	RW	0x0	重复计数器的值 (Repetition counter value) 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。 每次向下计数器 REP_CNT 达到 0, 会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值,

				因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。这意味着在 PWM 模式中, (REP+1)对应着: — 在边沿对齐模式下, PWM 周期的数目; — 在中心对称模式下, PWM 半周期的数目;

14.5.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)

比特	名称	属性	复位值	描述
15:0	CCR1	RW	0x0	捕获/比较通道 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出: CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较, 并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。

注意: 当 CCMRx 寄存器的 OCxM 位为 PWM 模式时, 设置 CCRx=0 时占空比不为 0 (有一个时钟周期的高电平)。将 OCxM 设为强制无效电平, 可解决此问题。其他定时器的情况同 TIMx

14.5.14. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)

比特	名称	属性	复位值	描述
15:0	CCR2	RW	0x0	捕获/比较通道 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出: CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。如果在

				TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入：CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

14.5.15. TIMx 刹车和死区寄存器 (TIMx_BDTR 偏移：0x44)

比特	名称	属性	复位值	描述
15	MOE	RW	0x0	主输出使能 (Main output enable) 一旦刹车输入有效，该位被硬件异步清'0'。根据 AOE 位的设置值，该位可以由软件清'0'或被自动置 1。它仅对配置为输出的通道有效。 0: 禁止 OC 和 OCN 输出或强制为空闲状态； 1: 如果设置了相应的使能位(TIMx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。
14	AOE	RW	0x0	自动输出使能 (Automatic output enable) 0: MOE 只能被软件置'1'； 1: MOE 能被软件置'1'或在下一个更新事件被自动置'1'(如果刹车输入无效)。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。
13	BKP	RW	0x0	刹车输入极性 (Break polarity) 0: 刹车输入低电平有效； 1: 刹车输入高电平有效。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的

				LOCK 位)设为'1', 则该位不能被修改。 注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
12	BKE	RW	0x0	刹车功能使能 (Break enable) 0: 禁止刹车输入(BRK 及 CCS 时钟失效事件); 1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。 注: 当设置了 LOCK 级别 1 时(TIMx_BDTR 寄存器中的 LOCK 位), 该位不能被修改。 注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
11	OSSR	RW	0x0	运行模式下“关闭状态”选择 (Off-state selection for Run mode) 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明。 0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0); 1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, 首先开启 OC/OCN 并输出无效电平, 然后置 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。
10	OSSI	RW	0x0	空闲模式下“关闭状态”选择 (Off-state selection for Idle mode) 该位用于当 MOE=0 且通道设为输出时。参考 OC/OCN 使能的详细说明。 0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0); 1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1,

				OC/OCN 首先输出其空闲电平, 然后 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。
9:8	LOCK	RW	0x0	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位; 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出, CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位; 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出, CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位); 注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIMx_BDTR 寄存器, 则其内容冻结直至复位。
7:0	UTG	RW	0x0	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间: $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times Tdtg, Tdtg = TDTs;$ $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times Tdtg, Tdtg = 2 \times TDTs;$ $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg, Tdtg = 8 \times TDTs;$ $DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg, Tdtg$

				$= 16 \times \text{TDTs}$; 例：若 $\text{TDTs} = 125\text{ns}(8\text{MHz})$，可能的死区时间为： 0 到 15875ns，若步长时间为 125ns; $16\mu\text{s}$ 到 31750ns，若步长时间为 250ns; $32\mu\text{s}$ 到 $63\mu\text{s}$，若步长时间为 $1\mu\text{s}$; $64\mu\text{s}$ 到 $126\mu\text{s}$，若步长时间为 $2\mu\text{s}$; 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3，则不能修改这些位。

14.5.16. TIMx DMA 控制寄存器 (TIMx_DCR 偏移：0x48)

比特	名称	属性	复位值	描述
15:13	-	-	-	位 15:13 保留，始终读为 0。
12:8	DBL	RW	0x0	DMA 连续传送长度 (DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，即：定义传输的次数，传输可以是半字(双字节)或字节： 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 10001: 18 次传输 例：我们考虑这样的传输：$\text{DBL}=7, \text{DBA}=\text{TIMx\_CR1}$ - 如果 $\text{DBL}=7, \text{DBA}=\text{TIMx\_CR1}$ 表示待传输数据的地址，那么传输的地址由下式给出：(TIMx_CR1 的地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL 其中(TIMx_CR1 的地址) + DBA 再加上 7，给出了将要写入或者读出数据的地址，这样数据的传

				输将发生在从地址(TIMx_CR1 的地址) + DBA 开始的 7 个寄存器。 根据 DMA 数据长度的设置, 可能发生以下情况: - 如果设置数据为半字(16 位), 那么数据就会传输给全部 7 个寄存器。 - 如果设置数据为字节, 数据仍然会传输给全部 7 个寄存器: 第一个寄存器包含第一个 MSB 字节, 第二个寄存器包含第一个 LSB 字节, 以此类推。因此对于定时器, 用户必须指定由 DMA 传输的数据宽度。
7:5	-	-	-	位 7:5 保留, 始终读为 0。
4:0	DBA	RW	0x0	这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时), DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量: 00000: TIMx_CR1, 00001: TIMx_CR2, 00010: TIMx_SMCR, ...

14.5.17. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)

比特	名称	属性	复位值	描述
31:0	DMAB	RW	0x0	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIMx_CR1 地址 + DBA + DMA 索引, 其中: “TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1)所在的地址; “DBA”是 TIMx_DCR 寄存器中定义的基地址; “DMA 索引”

				是由 DMA 自动控制的偏移量，它的最大值取决于 TIMx_DCR 寄存器中定义的 DBL。

14.5.18. TIMx 复用功能选择寄存器 1 (TIMx_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
17:14	-	-	-	保留，始终读为 0。
13	BKCMP4P	RW	0x0	比较器 4 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
12	BKCMP3P	RW	0x0	比较器 3 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
11	BKCMP2P	RW	0x0	比较器 2 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
10	BKCMP1P	RW	0x0	比较器 1 输入极性控制 0: 输入高电平有效 1: 输入低电平有效
9	BKINP	RW	0x0	刹车输入极性控制 0: 输入高电平有效 1: 输入低电平有效
8:5	-	-	-	保留，始终为 0。
4	BKCMP4E	RW	0x0	比较器 4 输入使能控制 0: 禁止 1: 使能
3	BKCMP3E	RW	0x0	比较器 3 输入使能控制 0: 禁止 1: 使能

2	BKCMP2E	RW	0x0	比较器 2 输入使能控制 0: 禁止 1: 使能
1	BKCMP1E	RW	0x0	比较器 1 输入使能控制 0: 禁止 1: 使能
0	BKINE	RW	0x0	刹车输入使能控制 0: 禁止 1: 使能

刹车输入详情请参考[定时器刹车输入互连关系](#)。

14.5.19. TIMx 复用功能选择寄存器 2 (TIMx_AF2 偏移: 0x64)

比特	名称	属性	复位值	描述
18:16	OCRSEL[2:0]	RW	0x0	ocref_clr 源选择 这些位选择 ocref_clr 输入源。 000: tim_ocref_clr0 001: tim_ocref_clr1 010: tim_ocref_clr2 011: tim_ocref_clr3 100: tim_ocref_clr4 101: tim_ocref_clr5 110: tim_ocref_clr6 111: tim_ocref_clr7 详情请参考 OCxREF 清除输入信号互连关系
15:0	-	-	-	保留, 始终为 0。

14.5.20. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
31:12	-	-	-	保留, 始终为 0。
11:8	TI2SEL	RW	0x0	TI2 输入选择 0000: tim_ti2_in0 0001: tim_ti2_in1 0010: tim_ti2_in2 0011: tim_ti2_in3 ... 1111: tim_ti2_in15 详情请参考定时器输入信号互连关系
7:4	-	-	-	保留, 始终为 0。
3:0	TI1SEL	RW	0x0	TI1 输入选择 0000: tim_ti1_in0 0001: tim_ti1_in1 0010: tim_ti1_in2 0011: tim_ti1_in3 ... 1111: tim_ti1_in15 详情请参考定时器输入信号互连关系

14.5.21. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留, 始终读为 0。
6	TBE	RW	0x0	触发事件的 DMA 请求类型 0: Single; 1: Burst;

5	COMBE	RW	0x0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;
4:3	-	-	-	保留，始终读为 0。
2	CC2BE	RW	0x0	捕获/比较 2 事件的 DMA 请求类型 0: Single; 1: Burst;
1	CC1BE	RW	0x0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0x0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

15. 低功耗定时器（LPTIM）

15.1. 概述

LPTIM 是一个 16 位定时器。由于 LPTIM 的时钟源具有多样性，因此 LPTIM 能够在所有电源模式（Standby 和 Powerdown 模式下除外）下保持运行状态。即使没有内部时钟源，LPTIM 也能运行，鉴于这一点，可将其用作“脉冲计数器”，这种脉冲计数器在某些应用中十分有用。此外，LPTIM 还能将系统从低功耗模式唤醒，因此非常适合实现“超时功能”，而且功耗极低。

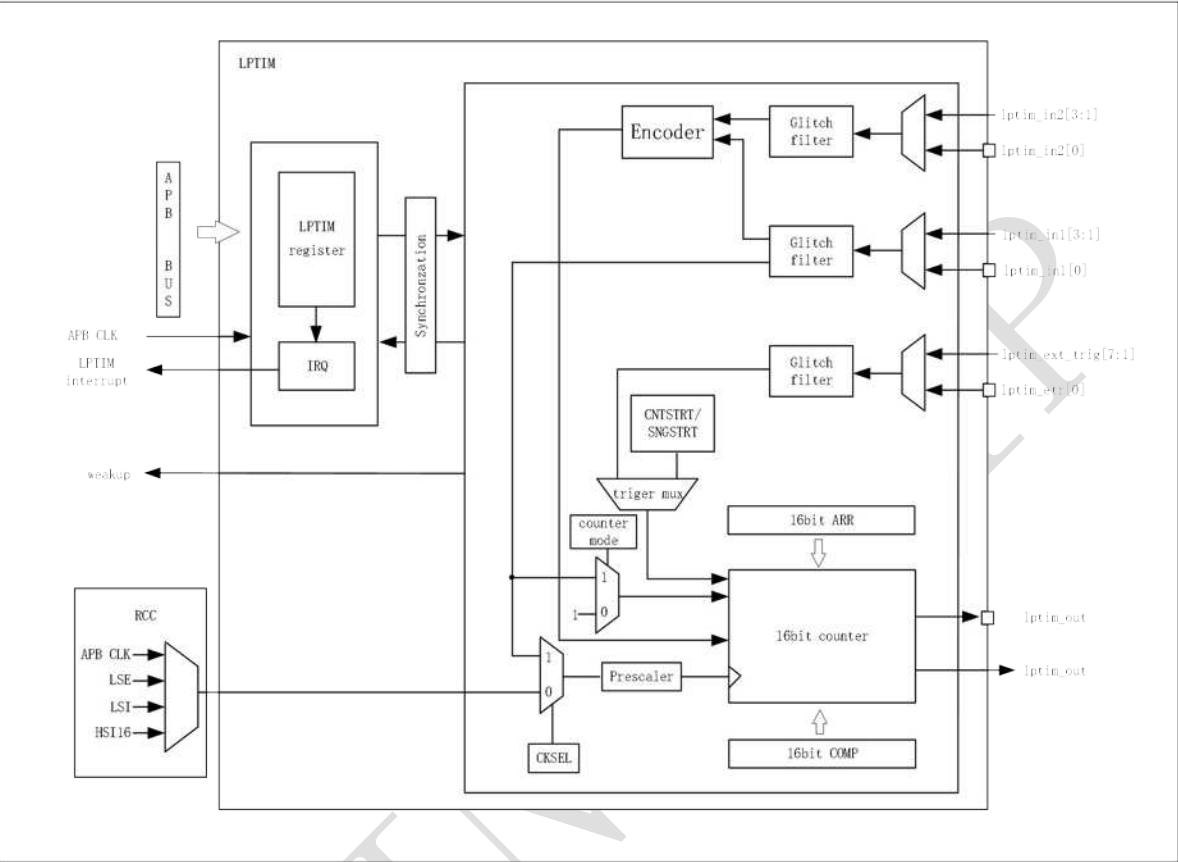
LPTIM 引入了一个灵活的时钟方案，该方案能够提供所需的功能和性能，同时还能最大程度地降低功耗。

15.2. 主要特性

- 16 位递增计数器
- 3 位预分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可选时钟
 - 内部时钟源：PCLK、RC32K、RC64M、XTL 时钟
 - LPTIM 输入的外部时钟源（在没有低功耗振荡器运行的情况下工作，由脉冲计数器作为外部时钟源）
- 16 位 ARR 自动重载寄存器
- 16 位比较寄存器
- 连续/单次模式
- 可选软件/硬件输入触发
- 可编程数字干扰滤波器
- 可配置输出：脉冲和 PWM
- 可配置 I/O 极性
- 编码器模式
- 重复计数器

15.3. 结构框图

图 15- 1 LPTIM 结构框图



lptim_out是内部LPTIM输出信号，可以连接到内部外设。

15.4. 功能说明

15.4.1. 触发映射

表 15- 1 LPTIM1 输入 1 连接

LPTIM_IN1_MUX	LPTIM1 输入 1 连接位置
LPTIM_IN1_MUX0	用作 LPTIM1_IN1 复用功能的 GPIO 引脚
LPTIM_IN1_MUX1	COMP1_OUT
LPTIM_IN1_MUX2	COMP3_OUT
LPTIM_IN1_MUX3	未连接

表 15- 2 LPTIM1 输入 2 连接

LPTIM_IN2_MUX	LPTIM1 输入 1 连接位置

LPTIM_IN2_MUX0	用作 LPTIM1_IN2 复用功能的 GPIO 引脚
LPTIM_IN2_MUX1	COMP2_OUT
LPTIM_IN2_MUX2	COMP4_OUT
LPTIM_IN2_MUX3	未连接

表 15- 3 外部触发连接

TRIGSEL	External trigger
LPTIM_EXT_TRIG0	GPIO
LPTIM_EXT_TRIG1	RTC alarm
LPTIM_EXT_TRIG2	RTC_TAMP1 input detection
LPTIM_EXT_TRIG3	RTC_TAMP2 input detection
LPTIM_EXT_TRIG4	COMP1_OUT
LPTIM_EXT_TRIG5	COMP2_OUT
LPTIM_EXT_TRIG6	COMP3_OUT
LPTIM_EXT_TRIG7	COMP4_OUT

15.4.2. 复位和时钟

LPTIM 可通过多个时钟源提供时钟。它可以由内部时钟信号提供时钟，内部时钟信号可通过复位和时钟控制器(RCC)在 PCLK、RC32K、RC64M 和 XTL 时钟源中进行选择。此外 LPTIM 还可通过注入到其外部 Input1 上的外部时钟信号提供时钟。当通过外部时钟源提供时钟时，LPTIM 可以在下述两种可能配置中的其中一种配置下运行：

- 第一种配置是，LPTIM 通过外部信号提供时钟，但是同时，内部时钟信号从 PCLK 或任何其他嵌入式振荡器（包括 RC32K、RC64M、XTL）提供给 LPTIM。
- 第二种配置是，LPTIM 仅由外部时钟源通过外部 Input1 提供时钟。此配置可在进入低功耗模式后所有内置振荡器关闭时，用于实现超时功能或脉冲计数器功能。

对 CKSEL 和 COUNTMODE 位进行编程，可控制 LPTIM 使用外部时钟源还是内部时钟源。

当使用外部时钟源时，可使用 CKPOL 位选择外部时钟信号的有效边沿。如果上升

沿和下降沿均为有效边沿，则还应提供内部时钟信号（第一种配置）。在这种情况下，内部时钟信号频率应至少为外部时钟信号频率的 4 倍。

15.4.3. 干扰滤波器

LPTIM 输入、外部（映射到 GPIO）或内部（在芯片级上映射到其他嵌入式外设，例如嵌入式比较器）由数字滤波器保护，避免任何毛刺和噪声干扰在 LPTIM 内部传播，从而防止产生意外计数或触发。在激活数字滤波器之前，首先应向 LPTIM 提供内部时钟源，这是保证滤波器正常工作的必要条件。

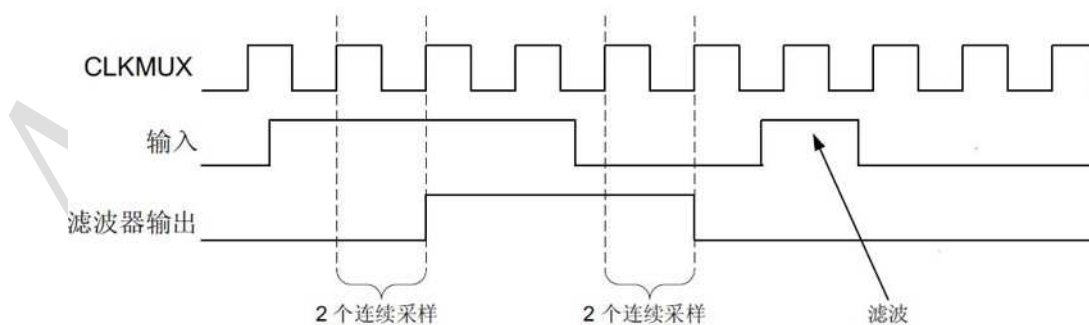
数字滤波器分为两组：

- 第一组数字滤波器保护 LPTIM 外部输入。数字滤波器的敏感性由 CKFLT 位控制。
- 第二组数字滤波器保护 LPTIM 内部触发输入。数字滤波器的敏感性由 TRGFLT 位控制。

注：数字滤波器的敏感性以组为单位进行控制。无法单独配置同一组内各个数字滤波器的敏感性。

滤波器的敏感性会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续采样时，才能将某信号电平变化视为有效切换。图 193 给出了编程 2 个连续采样时，干扰滤波器行为的示例。

图 15-2 干扰滤波器时序图



注：不提供内部时钟信号时，必须通过将 CKFLT 和 TRGFLT 位设为 0 来停用数字滤波器。在这种情况下，可使用外部模拟滤波器来防止 LPTIM 外部输入产生干扰。

15.4.4. 预分频器

LPTIM16 位计数器前面有一个可配置的 2 次幂预分频器。预分频器的分频比由

PRESC[2: 0]3 位字段控制。下表列出了所有可能的分频比：

表 15- 4 预分频系数

PRESC	分频系数
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

使能匹配中断情况下，不建议使用预分频功能。否则当计数时钟较低时，中断程序退出后会因为中断标志位再次产生从而立即又进入中断程序，直到计数时钟加 1 后才能退出中断。

15.4.5. 触发多路复用器

LPTIM 计数器可通过软件启动，也可以在 8 个触发输入之一上检测到有效边沿后启动。

TRIGEN[1:0]用于确定 LPTIM 触发源：

- TRIGEN[1:0]等于“00”时，LPTIM 计数器会在通过软件将 CNTSTRT 位或 SNGSTRT 位其中之一置 1 后立即启动。TRIGEN[1:0]的其余三个可能的值用于配置触发输入使用的有效边沿。LPTIM 计数器会在检测到有效边沿后立即启动。
- TRIGEN[1:0]不等于“00”时，TRIGSEL[2:0]用于选择使用 8 个触发输入中的哪一个来启动计数器。

外部触发信号视为 LPTIM 的异步信号。因此，检测到触发信号后，由于同步问题，需要延迟两个计数器时钟周期，定时器才能开始运行。

如果在定时器已启动时发生新的触发事件，则此事件将被忽略（除非已使能超时功能）。

注：必须使能定时器，才能将 SNGSTRT/CNTSTRT 位置 1。当定时器禁止时，对这些位执行的任何写操作都将被硬件丢弃。

15.4.6. 工作模式

LPTIM 支持以下两种工作模式：

- 连续模式：定时器自由运行，由触发事件启动并且直到被禁止才会停止
- 单触发模式：定时器由触发事件启动，并在 LPTIM 产生更新事件时（或在达到无重复计数器的产品的 ARR 值时）停止。

单触发模式

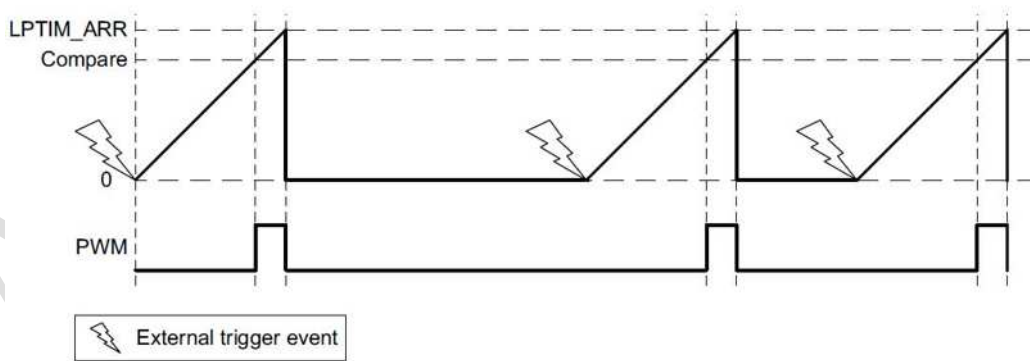
要启用单触发模式，必须将 SNGSTRT 位置 1。

- 重复计数器功能不可用

新的触发事件将重新启动计时器。在计数器启动之后且计数器达到 ARR 之前发生的任何触发事件都将被丢弃。

选择外部触发时，在 SNGSTRT 位置 1 后以及计数器寄存器停止后（包含零值）到达的每个外部触发事件都将为计数器启动新的单触发计数周期，如图所示。

图 15-3 LPTIM 输出波形，单触发模式



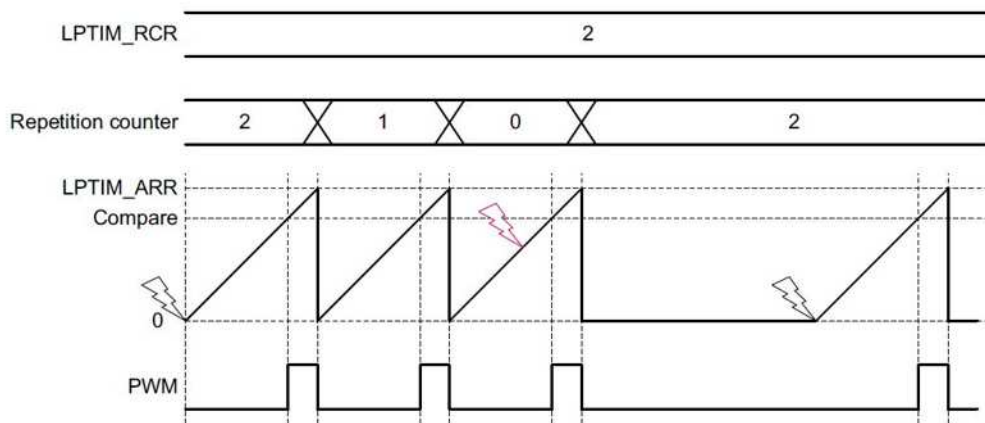
- 重复计数器功能可用

新的触发事件将重新启动计时器。在计数器启动之后和下一个 LPTIM 更新事件之前发生的任何触发事件都将被丢弃。

如果选择了外部触发，则每个外部触发事件在 SNGSTRT 位置 1 并且重复计数器停止之后（在更新事件之后）到达，并且如果重复寄存器的内容不为零，则重新加载重复

计数器使用重复寄存器已经包含的值，并开始一个新的单次计数周期，如图所示。

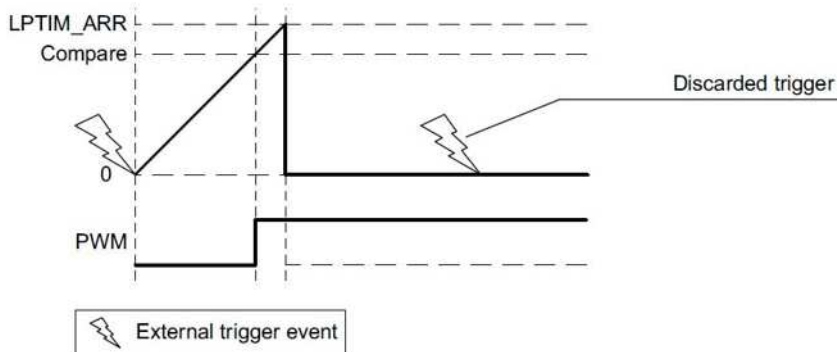
图 15-4 LPTIM 输出波形，单触发模式，当重复寄存器不为 0（PRELOAD=1）



已激活一次设置模式：

LPTIMx_CFGR 寄存器中的 WAVE 位域置 1 时，将激活置 1 一次模式。在这种情况下，计数器仅会在第一个触发事件后启动一次，任何后续触发事件都将被丢弃，如图 194 所示。

图 15-5 LPTIM 输出波形，单触发模式，一次激活模式（WAVE=1）



若通过软件启动（TRIGEN[1:0]="00"），将 SNGSTRT 置 1 会使计数器进行单触发计数。要启用连续计数，必须将 CNTSTRT 位置 1。

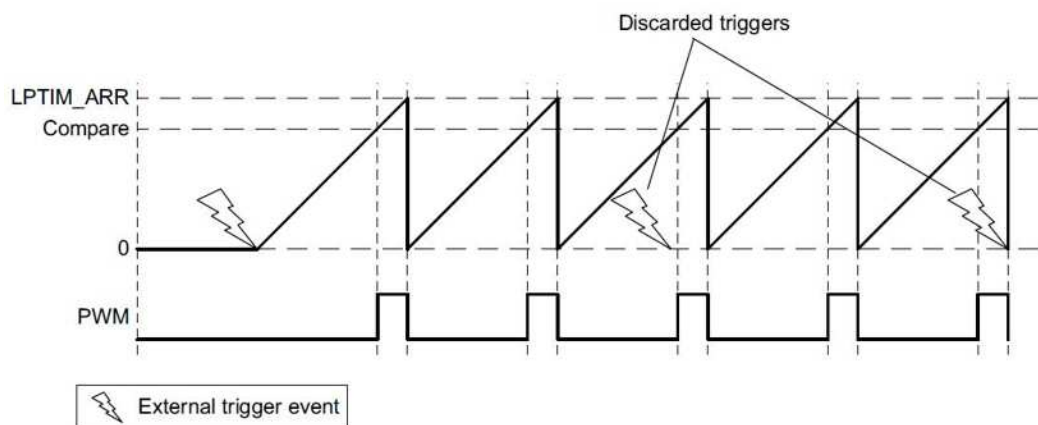
连续模式

要启用连续计数，必须将 CNTSTRT 位置 1。

如果选择了外部触发，则在设置 CNTSTRT 之后到达的外部触发事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，如图 341 所示。

若通过软件启动（TRIGEN[1:0]="00"），将 CNTSTRT 置 1 会使计数器开始连续计数。

图 15-6 LPTIM 输出波形，重复触发模式



SNGSTRT 和 CNTSTRT 位只能在定时器使能时（ENABLE 位置 1）置 1。可以“实时的”从单次模式切换为连续模式。

如果先前选择了连续模式，则设置 SNGSTRT 会将 LPTIM 切换为单发模式。一旦生成 LPTIM 更新事件（或对于没有重复计数器的产品，当达到 ARR 时），计数器（如果处于活动状态）将停止。

如果之前选择的是单触发模式，则将 CNTSTRT 置 1 会使 LPTIM 切换为连续模式。计数器（激活时）将在达到 ARR 后立即重新启动。

15.4.7. 超时功能

若在一个选定的触发输入上检测到有效边沿，则可用于复位 LPTIM 计数器。该功能通过 TIMOUT 位进行控制。

第一个触发事件将启动计时器，任何连续的触发事件将复位 LPTIM 计数器和重复计数器，并且计时器将重新启动。

可实现低功耗超时功能。超时值对应于比较值；如果在预期的时间帧内未发生触发，MCU 将由比较匹配事件唤醒。

15.4.8. 生成波形

两个 16 位寄存器，LPTIMx_ARR（自动重载寄存器）和 LPTIMx_CMP（比较寄存器）用于在 LPTIM 输出上生成多个不同的波形

定时器可生成以下波形：

- PWM 模式

若 LPTIMx_CMP 寄存器与 LPTIMx_CNT 寄存器匹配（相等），则会立即将 LPTIM 输出置 1。若 LPTIMx_ARR 寄存器与 LPTIMx_CNT 寄存器匹配（相等），则会立即将 LPTIM 输出复位。

立即更新寄存器模式下，若动态更改的 LPTIMx_CMP 值小于当前 LPTIMx_CNT 值，则当前周期内将无法匹配，LPTIM 输出保持不变，直到 LPTIMx_CNT 值与 LPTIMx_ARR 值匹配，LPTIM 输出复位。

立即更新寄存器模式下，若动态更改的 LPTIMx_ARR 值小于当前 LPTIMx_CNT 值，则当前周期内将无法匹配，LPTIM 输出保持不变，直到 LPTIM_CNT 计数到 0xFFFF 溢出回到 0，下个周期 LPTIMx_CMP 值与 LPTIMx_CNT 值匹配时，LPTIM 输出置 1。

- 单脉冲模式

对于第一个脉冲，输出波形与 PWM 模式输出波形类似，随后输出将永久复位。

- 置 1 一次模式

除输出保持最后一个信号电平外（取决于配置的输出极性），输出波形与单脉冲模式输出波形类似。

上述模式要求 LPTIMx_ARR 寄存器的值严格大于 LPTIMx_CMP 寄存器的值。

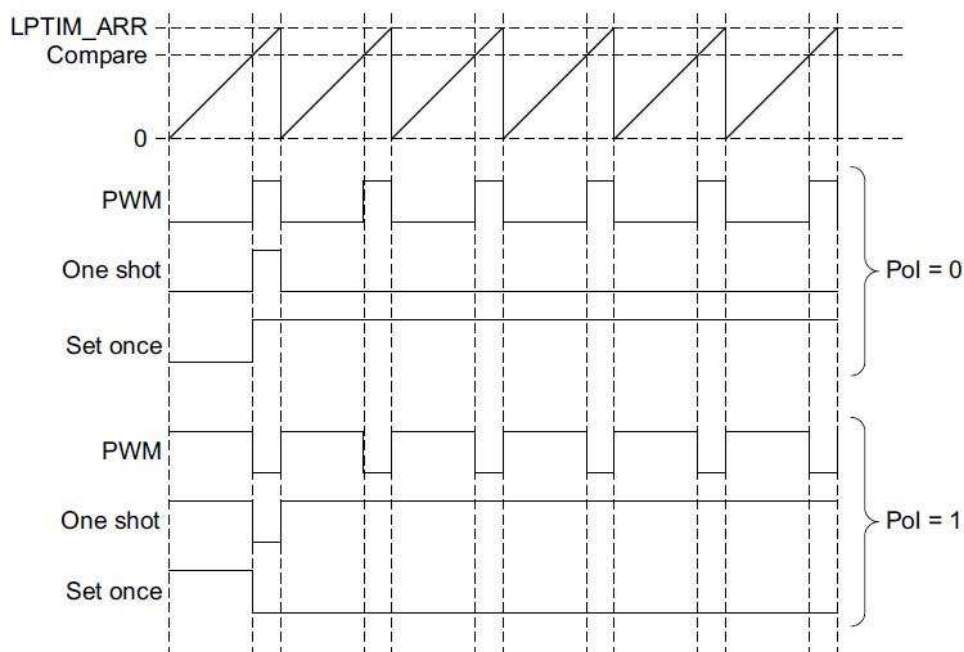
LPTIM 输出波形可通过 WAVE 位配置，具体如下：

- ◆ 若将 WAVE 位复位为 0，则会强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于将哪个位（CNTSTRT 或 SNGSTRT）置 1。
- ◆ 若将 WAVE 位置 1，则会强制 LPTIM 生成置 1 一次模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此输出默认值将在极性重新配置后立即更改，甚至会在定时器使能前进行更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。图 197 给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 15-7 生成波形



15.4.9. 寄存器更新

LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器在 APB 总线写操作后会立即更新，或如果计时器已启动则与下一个 LPTIM 更新事件同步（如果没有重复计数器的产品已经启动计时器，则在当前时间段结束时）。

PRELOAD 位控制 LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器的更新方式：

- ◆ 当 PRELOAD 位复位为 0 时，LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器会在写访问后立即更新。
- ◆ 当 PRELOAD 位置 1 时，若定时器已启动，LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器会在下一个 LPTIM 更新事件中更新（或在无重复计数器产品的当前阶段结束时）。

APB 总线和 LPTIM 使用的时钟不同，因此在 APB 写操作后，需要经过一定的延迟，写入值才能用于计数器比较器。在此延迟期间，必须避免向这些寄存器执行其他写操作。

LPTIMx_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示 LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器的写操作已完成。

向 LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器执行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志或 CMPOK 标志置 1 前执行连续的写操作将造成无法预知的结果。

15.4.10. 计数器模式

LPTIM 计数器可用于对 LPTIM Input1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。CKSEL 位和 COUNTMODE 位用于控制将使用哪些源更新计数器。

若使用 LPTIM 对 Input1 上的外部事件进行计数，计数器可在上升沿、下降沿或两种边沿进行更新，具体取决于写入 CKPOL[1:0]位的值。

根据 CKSEL 和 COUNTMODE 值，可选择以下计数模式：

- CKSEL = 0: LPTIM 由内部时钟源提供时钟

- COUNTMODE = 0

LPTIM 配置为由内部时钟源提供时钟，而 LPTIM 计数器配置为在每个内部时钟脉冲之后进行更新。

- COUNTMODE = 1

LPTIM 外部 Input1 通过提供给 LPTIM 的内部时钟采样。因此，为了不丢失任何事件，外部 Input1 信号变化的频率决不应超过提供给 LPTIM 的内部时钟的频率。此外，提供给 LPTIM 的内部时钟一定不能预先分频（PRESC[2: 0] = 000）。

- CKSEL = 1: LPTIM 由外部时钟源提供时钟

- COUNTMODE 值不相关。

在这种配置下，LPTIM 无需内部时钟源（已使能干扰滤波器时除外）。注入到 LPTIM 外部 Input1 的信号用作 LPTIM 的系统时钟。此配置适合未使能任何内置振荡器的工作模式。对于这种配置，LPTIM 计数器可以在 input1 时钟信号的上升沿或下降沿进行更新，但不可在上升沿和下降沿均更新。

由于注入到 LPTIM 外部 Input1 的信号也可用于 LPTIM 的时钟，计数器递增计数前存在一些初始延时（使能 LPTIM 后）。更确切地说，LPTIM 外部 Input1 的前五个有效边沿将丢失（使能 LPTIM 后）。

15.4.11. 定时器使能

LPTIMx_CR 寄存器的 ENABLE 位用于使能/禁止 LPTIM。将 ENABLE 位置 1 后，需要延迟两个计数器时钟周期，才能真正使能 LPTIM。

LPTIMx_CFGR 和 LPTIMx_IER 寄存器必须在禁止 LPTIM 后才能修改。

15.4.12. 定时器计数器复位

为了将 LPTIM_CNT 寄存器的内容重置为零，实现了两种重置机制：

同步复位机制：同步复位由 LPTIM_CR 寄存器中的 COUNTRST 位控制。将 COUNTRST 位字段设置为“1”后，复位信号将在 LPTIM 内核时钟域中传播。因此，重要的是要注意，在考虑复位之前，将经过 LPTIM 内核逻辑的几个时钟脉冲。这将使 LPTIM 计数器在触发复位到生效之间产生一些额外的计数。由于 COUNTRST 位位于 APB 时钟域中，而 LPTIM 计数器位于 LPTIM 内核时钟域中，向 COUNTRST 位写入 1 时，内核时钟需要 3 个时钟周期的延迟才能同步 APB 时钟域发出的复位信号。

异步复位机制：异步复位由位于 LPTIM_CR 寄存器中的 RSTARE 位控制。当该位置 1 时，对 LPTIM_CNT 寄存器的任何读访问都将其内容复位为零。异步复位应在没有提供 LPTIM 内核时钟的时间范围内触发。例如，当 LPTIM Input1 用作外部时钟源时，仅当有足够的保险保证在 LPTIM Input1 上不会发生翻转时，才应应用异步复位。

应该注意的是，为了可靠地读取 LPTIM_CNT 寄存器的内容，必须执行两个连续的读取访问并进行比较。当两个读取访问的值相等时，可以认为读取访问是可靠的。不幸的是，启用异步复位后，将无法读取两次 LPTIM_CNT 寄存器。

注：LPTIM 内部没有机制可以防止开发人员同时使用两种复位机制。因此，开发人员应确保应用程序上具备不能同时使用这两种复位机制的方法，如互斥机制。

15.4.13. 编码器模式

此模式用于处理来自正交编码器的信号，此正交编码器用于检测旋转元件的角度位置。编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 LPTIMx_ARR 寄存器中编程的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 LPTIMx_ARR。通过两个外部输入信号 Input1 和 Input2 生成时钟信号作为 LPTIM 计数器时钟。这两个信号间的相位确定计数方向。

仅当 LPTIM 由内部时钟源提供时钟时才可使用编码器模式。Input1 和 Input2 输入上的信号频率不得超过 LPTIM 内部时钟频率 4 分频。必须满足此条件才能确保 LPTIM 正常工作。

方向更改通过 LPTIM_ISR 寄存器中的两个 Down 和 Up 标志发出信号。使能 DOWNIE 位、UPIE 位，两个方向的改变事件都能产生中断。

要激活编码器模式，必须将 ENC 位置 1。LPTIM 必须首先配置为连续模式。

激活编码器模式后，将根据增量编码器的速度和方向自动修改 LPTIM 计数器。因此，其内容始终代表编码器的位置。计数方向由递增和递减标志指示，对应于所连传感器的旋转方向。

根据使用 CKPOL[1:0]位配置的边沿敏感性，可得几种不同的计数方案。下表汇总了可能的组合（假设 Input1 和 Input2 不同时切换）。

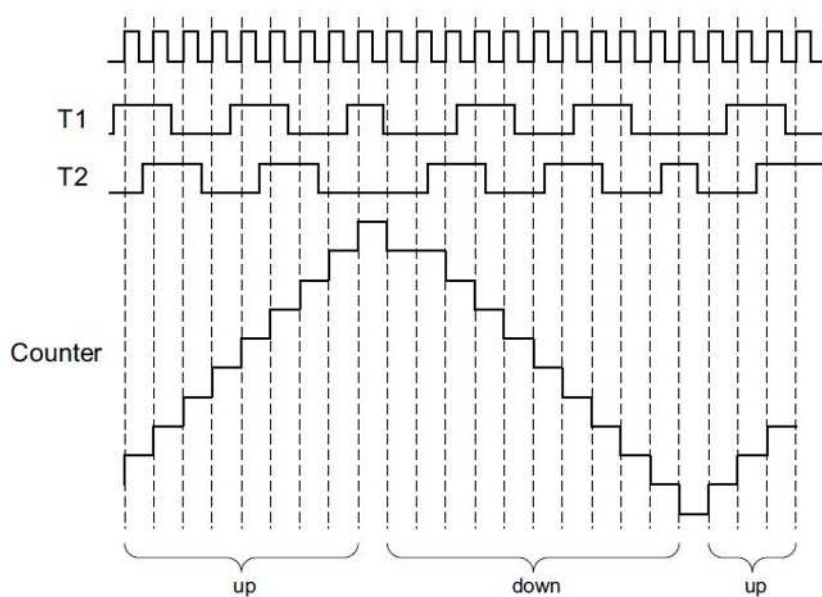
表 15- 5 编码器计数方案

有效边沿	相反信号的电平 input1 对应 input2 input2 对应 input1	Input1 信号		Input2 信号	
		上升	下降	上升	下降
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
上升沿	高	递减	递增	递增	递减
下降沿	低	递增	递减	递减	递增

下图所示为编码器模式下配置了两种边沿敏感性的计数序列。

注意：在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0。另外，预分频器分频比必须等于其复位值 1（PRESC[2:0]位必须为“000”）。

图 15-8 编码器模式计数序列



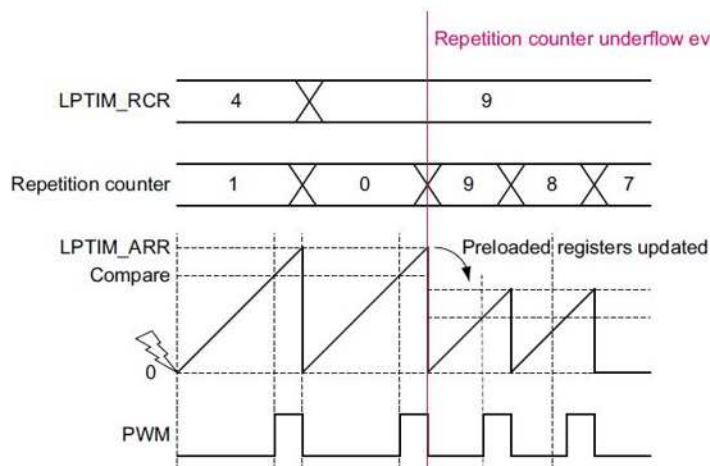
15.4.14. 重复计数器

LPTIM 具有一个重复计数器，每次 LPTIM 计数器溢出事件发生时，该计数器就会递减 1。当重复计数器到达零并且 LPTIM 计数器溢出时，将生成重复计数器下溢事件。在每个重复计数器下溢事件之后，向重复计数器加载属于重复寄存器 LPTIM_RCR 的 REP[7: 0]位字段的内容。

当重复计数为 0 时，每个 LPTIM 计数器溢出都会产生重复下溢事件。

写入 REP[7: 0]位域的值并不能立即加载到重复计数器中，仅当下一次重复下溢事件发生，写入的 REP[7: 0]数值才会加载到重复计数器中。此行为在下图进行了描述。

图 15-9 重复计数模式 LPTIM_RCR 不为 0 (PRELOAD=1)



重复计数器下溢事件与 LPTIM 预加载的寄存器更新有系统地关联（有关更多信息，请参见“寄存器更新”部分）。

重复计数器下溢事件通过映射到 LPTIM_ISR 寄存器中的更新事件（REPUE）标志发送给软件。置位时，如果设置了对应的映射到 LPTIM_IER 寄存器的更新事件中断使能（REPUEIE）控制位，则 REPUE 标志可以触发 LPTIM 中断。

重复寄存器 LPTIM_RCR 位于 APB 总线接口时钟域中，重复计数器本身位于 LPTIM 内核时钟域中。每次将新值写入 LPTIM_RCR 寄存器时，该新内容就会从 APB 总线接口时钟域传播到 LPTIM 内核时钟域，以便在重复计数器下溢事件发生后立即将新写入的值加载到重复计数器。新写入内容的同步延迟是四个 APB 时钟周期加上三个 LPTIM 内核时钟周期，并且经过 LPTIM_ISR 寄存器中的 REPOK 标志来发出信号。当 LPTIM 内核时钟周期相对较慢时例如，当 XTL 时钟源为 LPTIM 内核提供时钟时，可能需要很长时间才能通过软件对 REPOK 标志进行轮询，以检测 LPTIM_RCR 寄存器内容的同步已完成。因此，如果设置了 REPOK 标志，则 LPTIM_IER 寄存器中的 REPOKIE 控制位将被置位，从而产生中断。

注：对 LPTIM_RCR 寄存器进行写操作后，只有在完成前一个写操作后，才能对同一个寄存器执行新的写操作。REPOK 标志置 1 之前的任何连续写操作都将导致不可预测的结果。

注意：使用重复计数器时，必须将 LPTIM_CFGR 寄存器中的 PRELOAD 位置 1，否则可能会发生不可预测的行为。

15.4.15. 调试模式

当微控制器进入调试模式（内核暂停）时，LPTIM 计数器将继续正常工作或停止，具体取决于 DBG 模块中的 DBG_LPTIM_STOP 配置位。

15.5. LPTIM 低功耗模式

表 15- 6 LPTIM 低功耗模式

模式	描述
Sleep	无影响。LPTIM 中断可退出 Sleep 模式。
Low-power run	无影响。

Low-power sleep	无影响。LPTIM 中断可退出 Low-power sleep 模式。
Stop 0/1	无影响。LPTIM 时钟源选择 XTL、RC32K 时，LPTIM 中断可退出 Stop0/1 模式。
Stop2	无影响。LPTIM 时钟源选择 XTL、RC32K 时，LPTIM 中断可退出 Stop2 模式。
Standby	LPTIM 断电，退出 Standby、Powerdown 必须重新初始化 LPTIM。
Powerdown	

15.6. LPTIM 中断

若以下事件通过 LPTIMx_IER 寄存器使能，则这些事件会生成中断/唤醒事件：

- 与设定的数值比较后匹配
- 自动重载匹配（编码器模式下无论哪种方向）
- 外部触发事件
- 自动重载寄存器写操作完成
- 比较寄存器写操作完成
- 方向变化（编码器模式），可编程（递增/递减/同时递增和递减）。
- 重复计数器下溢事件
- 重复寄存器更新完成

注：只要 LPTIMx_IER 寄存器（中断使能寄存器）中的位在 LPTIMx_ISR 寄存器（状态寄存器）中相应标志置 1 后置 1，就不会触发中断。

表 15- 7 中断事件

中断事件	说明
比较匹配	当计数器寄存器 (LPTIM_CNT) 的内容与比较寄存器 (LPTIM_CMP) 的内容匹配时，生成中断标志。
自动重载匹配	当计数器寄存器 (LPTIM_CNT) 的内容与自动重新加载寄存器 (LPTIM_ARR) 的内容匹配时，生成中断标志。
外部触发事件	当检测到外部触发事件时，会生成中断标志。

自动重载寄存器写操作完成	当对 LPTIM_ARR 寄存器的写操作完成时，生成中断标志。
比较寄存器写操作完成	当对 LPTIM_CMP 寄存器的写操作完成时，生成中断标志。
方向更改	用于编码器模式。嵌入两个中断标志以发出方向更改的信号： - UP 标志发出递增计数方向更改的信号 - DOWN 标志发出递减计数方向更改的信号
重复计数器下溢事件	重复计数器下溢事件发生时，生成中断标志。
重复寄存器更新完成	当对 LPTIM_RCR 中 REP[7: 0]的写操作完成时，生成中断标志。

15.7. LPTIM 寄存器

LPTIM 寄存器基地址：

LPTIM1: 0x4000_7C00

偏置	名称	描述
0x00	LPTIM_ISR	中断和状态寄存器
0x04	LPTIM_ICR	中断清零寄存器
0x08	LPTIM_IER	中断使能寄存器
0x0C	LPTIM_CFGR1	配置寄存器 1
0x10	LPTIM_CR	控制寄存器
0x14	LPTIM_CMP	比较寄存器
0x18	LPTIM_ARR	自动重载寄存器
0x1C	LPTIM_CNT	计数器寄存器
0x24	LPTIM_CFGR2	配置寄存器 2
0x28	LPTIM_RCR	重复寄存器

15.7.1. 中断和状态寄存器 LPTIM_ISR（偏移：00h）

比特	名称	属性	复位值	描述

31:9	RSV	-	-	保留
8	REPOK	R	0	重复寄存器更新成功 0: 重复寄存器未更新。 1: 重复寄存器更新成功。 当 LPTIM_RCR 寄存器 REP 的 APB 总线写操作已成功完成，此位由硬件置位。 注：如果 LPTIM 不支持重复计数器功能，该位保留。
7	REPUE	R	0	重复计数器下溢事件 0: 重新计数器未发生下溢事件。 1: 重复计数器发生下溢事件。 当重复计数器下溢事件发生时，此位由硬件置 1。 注：如果 LPTIM 不支持重复计数器功能，该位保留。
6	DOWN	R	0	计数方向从递增变为递减 (Counter direction change up to down) 0: 计数方向未从递增变为递减 1: 计数方向从递增变为递减 在编码器模式下，计数方向从递增变为递减时，此位由硬件置 1。
5	UP	R	0	计数方向从递减变为递增 (Counter direction change down to up) 0: 计数方向未从递减变为递增 1: 计数方向从递减变为递增 在编码器模式下，计数方向从递减变为递增时，此位由硬件置 1。
4	ARROK	R	0	自动重载寄存器更新成功 (Autoreload register update OK)

				0: 自动重载寄存器未更新 1: 自动重载寄存器更新成功 当对 LPTIM_ARR 寄存器的 APB 总线写操作完成时, 此位由硬件置 1。
3	CMPOK	R	0	比较寄存器更新成功 (Compare register update OK) 0: 比较寄存器未更新 1: 比较寄存器更新成功 当对 LPTIM_CMP 寄存器的 APB 总线写操作完成时, 此位由硬件置 1。
2	EXTTRIG	R	0	外部触发边沿事件 (External trigger edge event) 0: 未发生外部触发边沿事件 1: 发生外部触发边沿事件 外部触发输入上产生有效边沿时, 此位由硬件置 1。如果此时定时器已启动, 而忽略触发事件, 则不会将此标志置 1。
1	ARRM	R	0	自动重载匹配 (Autoreload match) 0: LPTIM_CNT 值与 LPTIM_ARR 值不匹配 1: LPTIM_CNT 值与 LPTIM_ARR 值匹配 当 LPTIM_CNT 值与 LPTIM_ARR 值相等时, 此位由硬件置 1。
0	CMPM	R	0	比较匹配 (Compare match) 0: LPTIM_CNT 值与 LPTIM_CMP 值不匹配 1: LPTIM_CNT 值与 LPTIM_CMP 值匹配 当 LPTIM_CNT 值与 LPTIM_CMP 值相等时, 此位由硬件置 1。

15.7.2. 中断清零寄存器 LPTIM_ICR (偏移: 04h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8	REPOKCF	W	0	REPOK 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 REPOK 标志 注: 如果 LPTIM 不支持重复计数器功能, 该位保留。
7	REPUECF	W	0	REPUE 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 REPUE 标志 注: 如果 LPTIM 不支持重复计数器功能, 该位保留。
6	DOWNCF	W	0	DOWN 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 DOWN 标志
5	UPCF	W	0	UP 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 UP 标志
4	ARROKCF	W	0	ARROK 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 ARROK 标志
3	CMPOKCF	W	0	CMPOK 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 CMPOK 标志
2	EXTTRIGCF	W	0	EXTTRIG 标志清零 0: 无效操作

				1: 清除 LPTIM_ISR 中 EXTTRIG 标志
1	ARRMCF	W	0	ARRM 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 ARRM 标志
0	CMPMCF	W	0	CMPM 标志清零 0: 无效操作 1: 清除 LPTIM_ISR 中 CMPM 标志

15.7.3. 中断使能寄存器 LPTIM_IER (偏移: 08h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8	REPOKIE	RW	0	重复寄存器更新成功中断使能 0: 禁止 1: 使能 注: 如果 LPTIM 不支持重复计数器功能, 该位无意义。
7	REPUEIE	RW	0	重复计数器下溢事件中断使能 0: 禁止 1: 使能 注: 如果 LPTIM 不支持重复计数器功能, 该位无意义。
6	DOWNIE	RW	0	方向变为递减中断使能 (Direction change to down Interrupt Enable) 0: 禁止 1: 使能
5	UPIE	RW	0	方向变为递增中断使能 (Direction change to UP Interrupt Enable) 0: 禁止 1: 使能

4	ARROKIE	RW	0	自动重载寄存器更新成功中断使能 (Autoreload register update OK Interrupt Enable) 0: 禁止 1: 使能
3	CMPOKIE	RW	0	比较寄存器更新成功中断使能 (Compare register update OK Interrupt Enable) 0: 禁止 1: 使能
2	EXTTRIGIE	RW	0	外部触发有效边沿中断使能 (External trigger valid edge Interrupt Enable) 0: 禁止 1: 使能
1	ARRMIE	RW	0	自动重载匹配中断使能 (Autoreload match Interrupt Enable) 0: 禁止 1: 使能
0	CMPMIE	RW	0	比较匹配中断使能 (Compare match Interrupt Enable) 0: 禁止 1: 使能

注：
LPTIM_IER 寄存器必须在 LPTIM_CR 的 ENABLE 位为 0) 时才能修改。

15.7.4. 配置寄存器 1 LPTIM_CFGR1 (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
24	ENC	RW	0	编码器模式使能 (Encoder mode enable) ENC 位控制编码器模式 0: 禁止编码器模式 1: 使能编码器模式

23	COUNTMODE	RW	0	计数器模式使能 (counter mode enabled) COUNTMODE 位用于选择 LPTIM 使用哪个时钟源来为计数器提供时钟： 0：计数器在每个内部时钟脉冲后递增 1：计数器在 LPTIM 外部 Input1 上的每个有效时钟脉冲后递增
22	PRELOAD	RW	0	寄存器更新模式 (Registers update mode) PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式 0：寄存器在每次 APB 总线写访问后更新 1：寄存器在当前 LPTIM 周期结束时更新
21	WAVPOL	RW	0	波形极性 (Waveform shape polarity) WAVEPOL 位控制输出极性 0： LPTIM_CNT 小于 LPTIM_CMP 时，输出低电平， LPTIM_CNT 等于 LPTIM_CMP 时，输出高电平。 1： LPTIM_CNT 小于 LPTIM_CMP 时，输出高电平， LPTIM_CNT 等于 LPTIM_CMP 时，输出低电平。
20	WAVE	RW	0	波形 (Waveform shape) WAVE 位控制输出波形 0： 停用置 1。一次模式和 PWM/单脉冲波形 1： 激活置 1。一次模式
19	TIMOUT	RW	0	超时使能 (Timeout enable) TIMOUT 位控制超时功能 0：定时器已启动时到达的触发事件将被忽略 1：定时器已启动时到达的触发事件将复位

				并重新启动计数器
18:17	TRIGEN[1:0]	RW	00	触发使能和极性 (Trigger enable and polarity) TRIGEN 位控制 LPTIM 计数器是否由外部触发信号启动。如果已选择由外部触发信号启动，触发有效边沿的配置有以下三种： 00：软件触发（由软件启动计数） 01：上升沿为有效边沿 10：下降沿为有效边沿 11：上升沿和下降沿均为有效边沿
16	RSV	-	-	保留
15:13	TRIGSEL[2:0]	RW	000	触发源选择器 (Trigger selection) TRIGSEL 位用于选择作为 LPTIM 触发事件的触发源，可用触发源包括以下 8 种： 000: lptim_ext_trig0 001: lptim_ext_trig1 010: lptim_ext_trig2 011: lptim_ext_trig3 100: lptim_ext_trig4 101: lptim_ext_trig5 110: lptim_ext_trig6 111: lptim_ext_trig7
12	RSV	-	-	保留
11:9	PRESC[2:0]	RW	000	时钟预分频器 (Clock prescaler) PRESC 位配置预分频器的分频系数。分频系数可从以下分频系数中选择： 000: 1

				001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
8	RSV	-	-	保留
7:6	TRGFLT[1:0]	RW	00	触发信号的可配置数字滤波器 (Configurable digital filter for trigger) TRGFLT 值用于设置连续相同采样的数量，若在内部触发信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能。 00: 任何触发信号有效电平变化均视为有效触发。 01: 触发信号有效电平变化必须至少稳定 2 个时钟周期，才能将其视为有效触发。 10: 触发信号有效电平变化必须至少稳定 4 个时钟周期，才能将其视为有效触发。 11: 触发信号有效电平变化必须至少稳定 8 个时钟周期，才能将其视为有效触发。
5	RSV	-	-	保留
4:3	CKFLT[1:0]	RW	00	外部时钟的可配置数字滤波器 (Configurable digital filter for external clock) CKFLT 值用于设置连续相同采样的数量，若在外时钟信号电平发生变化时检测到

				此类连续采样， 才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能。 00：任何外部时钟信号电平变化均视为有效切换。 01：外部时钟信号电平变化必须至少稳定 2 个时钟周期，才能将其视为有效切换。 10：外部时钟信号电平变化必须至少稳定 4 个时钟周期，才能将其视为有效切换。 11：外部时钟信号电平变化必须至少稳定 8 个时钟周期，才能将其视为有效切换。
2:1	CKPOL[1:0]	RW	00	时钟极性 (Clock Polarity) 如果 LPTIM 由外部时钟源提供时钟： 当 LPTIM 由外部时钟源提供时钟时，CKPOL 位用于配置计数所使用的有效边沿： 00：上升沿为用于计数的有效边沿。 01：下降沿为用于计数的有效边沿。 10：上升沿和下降沿均为有效边沿。当外部时钟信号的上升沿和下降沿均视为有效边沿时，LPTIM 还必须由内部时钟源提供时钟，且内部时钟源频率至少等于外部时钟频率的四倍。 11：不允许 如果将 LPTIM 配置为编码器模式（ENC 位置 1）： 00：编码器子模式 1 激活。 01：编码器子模式 2 激活。 10：编码器子模式 3 激活。

				有关编码器模式子模式的更多详细信息，请参见第 38.4.14 节：编码器模式。
0	CKSEL	RW	0	时钟选择器 (Clock selector) CKSEL 位选择 LPTIM 将使用的时钟源： 0： LPTIM 由内部时钟源（APB 时钟或任意内置振荡器）提供时钟。 1： LPTIM 由外部时钟源通过 LPTIM 外部 Input1 提供时钟。

注：LPTIM_CFGR1 寄存器必须在 LPTIM 禁止（LPTIM_CR 的 ENABLE 位为 0）时才能修改。

15.7.5. 控制寄存器 LPTIM_CR（偏移：10h）

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4	RSTARE	W	0	读操作后复位使能 (Reset after read enable) 此位由软件置 1 和清零。当 RSTARE 置 1 时，对 LPTIM_CNT 寄存器的任何读取访问都将异步复位 LPTIM_CNT 寄存器的内容。 注意： 只能对该位域执行写操作。这意味着该位不能被读回以验证已写入的值。例如，如果该位设置为 1，即使使能了“读操作后复位”功能（由于该位域以前已写入 1），尝试读取它也将返回 0。要关闭“读操作后复位”或确保已经关闭，应（通过将其编程为 0）将该位复位，即使其已经为 0。
3	COUNTRST	RW	0	计数器复位 (Counter reset)

				此位通过软件置 1，通过硬件清零。当设置为 1 时，该位将触发 LPTIM_CNT 计数器寄存器的同步复位。由于该复位的同步性质，它仅在 3 个 LPTimer 内核时钟周期（LPTimer 内核时钟可能不同于 APB 时钟）的同步延迟之后发生。 注意：COUNTRST 在已由硬件清零之前，不得由软件置“1”。因此，软件应在尝试将 COUNTRST 位置“1”之前检查其是否已清零。
2	CNTSTRT	RW	0	定时器以连续模式启动 (Timer start in Continuous mode) 此位通过软件置 1，通过硬件清零。 若通过软件启动 (TRIGEN[1:0] = “00”)，将此位置 1 会使 LPTIM 以连续模式启动。 如果禁止软件启动 (TRIGEN[1:0] 不等于 “00”)，将此位置 1 会使定时器在检测到外部触发信号后立即以连续模式启动。 如果在进行单脉冲模式计数时将此位置 1，则在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时定时器不会停止，LPTIM 计数器将继续以连续模式计数。 只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。
1	SNGSTRT	RW	0	LPTIM 以单脉冲模式启动 (LPTIM start in Single mode) 此位通过软件置 1，通过硬件清零。

				若通过软件启动（TRIGEN[1:0] = “00”），将此位置 1 会使 LPTIM 以单脉冲模式启动。 如果禁止软件启动（TRIGEN[1:0] 不等于 “00”），将此位置 1 会使 LPTIM 在检测到外部触发信号后立即以单脉冲模式启动。 如果在 LPTIM 处于连续计数模式时将此位置 1，LPTIM 将在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时停止。 只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。
0	ENABLE	RW	0	LPTIM 使能 (LPTIM enable) ENABLE 位由软件置 1 和清零。 0: 禁止 LPTIM 1: 使能 LPTIM

15.7.6. 比较寄存器 LPTIM_CMP（偏移：14h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CMP[15:0]	RW	0x0000	比较值 (Compare value) CMP 为 LPTIM 所使用的比较值。 注意： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_CMP 寄存器。

15.7.7. 自动重载寄存器 LPTIM_ARR（偏移：18h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR[15:0]	RW	0x0001	自动重载值 (Auto reload value) ARR 为 LPTIM 的自动重载值。 此值必须严格大于 CMP[15:0] 的值。 注意： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_ARR 寄存器。

15.7.8. 计数器寄存器 LPTIM_CNT（偏移：1Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT[15:0]	R	0x0000	计数器值 (Counter value) 当 LPTIM 通过异步时钟运行时，读取 LPTIM_CNT 寄存器会返回不可靠的值。因此在这种情况下，必须连续执行读访问两次，并验证两次返回的值是否相同。当两次连续读访问的值相同时，可认为读访问可靠。

15.7.9. 配置寄存器 2 LPTIM_CFGR2（偏移：24h）

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:4	IN2SEL[1:0]	RW	00	LPTIM 输入 2 选择 (LPTIM input 2 selection)

				IN2SEL 位控制 LPTIM 输入 2 多路复用器, 它将 LPTIM 输入 2 连接到可用输入之一。 00: lptim_in2_mux0 01: lptim_in2_mux1 10: lptim_in2_mux2 11: lptim_in2_mux3
3:2	RSV	-	-	保留
1:0	IN1SEL[1:0]	RW	00	LPTIM 输入 1 选择 (LPTIM input 1 selection) IN2SEL 位控制 LPTIM 输入 1 多路复用器, 它将 LPTIM 输入 1 连接到可用输入之一。 00: lptim_in1_mux0 01: lptim_in1_mux1 10: lptim_in1_mux2 11: lptim_in1_mux3

注: LPTIM_CFGR2 寄存器必须在 LPTIM 使能 (LPTIM_CR 的 ENABLE 位为 1) 时才能修改。

15.7.10. 重复寄存器 LPTIM_RCR (偏移: 28h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
7:0	REP[7:0]	RW	0x00	重复寄存器值 REP 是 LPTIM 的重复值。在下一个重复下溢事件发生之前, 写入 REP[7:0] 对重复计数器的内容没有影响。 注意: LPTIM_RCR 寄存器只能在 LPTIM 使能 (ENABLE 位设置为 1) 时修改。

ALISINOCHIP

16. 看门狗（WDT）

16.1. 概述

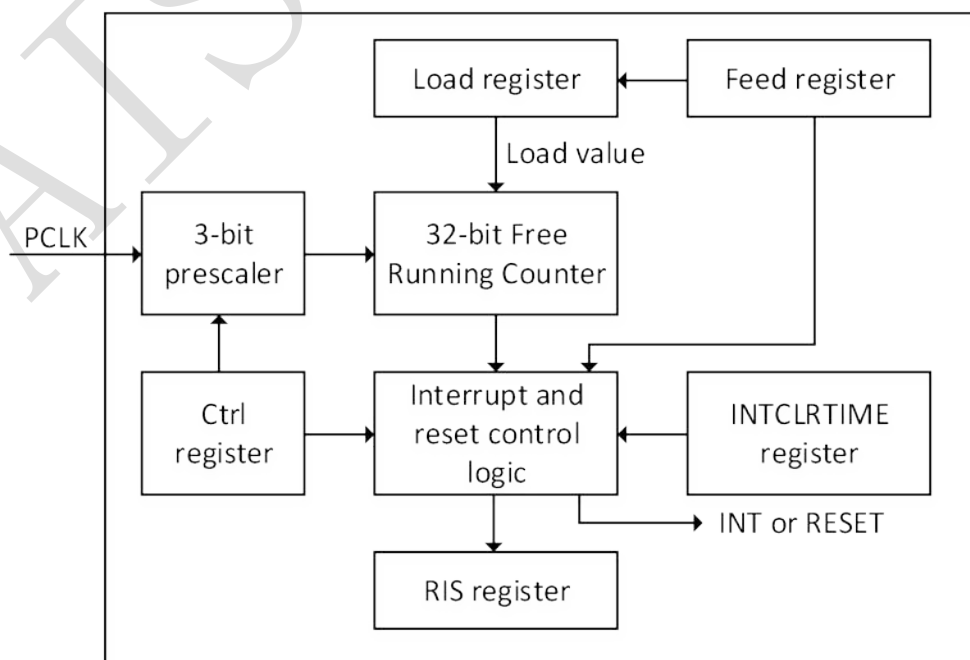
看门狗模块（WatchDog）采用 32 位的递减计数器，可从一个可编程的加载值减到零。当计数器计数减为 0，如果看门狗动作设为复位，则看门狗模块输出复位信号，复位系统；如果看门狗动作设为中断，则触发看门狗中断，如果在设定的清除时间限定内软件仍未清除看门狗中断，则产生复位信号复位系统。用户可以通过设置定时器使能位来停止/启动计数器。

16.2. 主要特性

- 32 位的递减计数器
- 可编程预分频因子
- 可编程装载值
- 可编程中断清除时限

16.3. 结构框图

图 16-1 WDT 结构图



加载寄存器（Load register）写入需要向下计数的加载值（Load value），计数的时钟来自于 3 位预分频器（3-bit prescaler）。当看门狗启动时，会从加载值减到 0。看门狗可以产生复位或者中断，可以通过控制寄存器（Ctrl register）来选择。在产生中断时，如果在清除时限寄存器（INTCLRTIME register）内设定的时间内未清除看门狗中断，则产生复位信号复位系统。喂狗寄存器（Feed register）中写入特征值 0xAA55A55A 使计数器从加载寄存器加载装载值，加载动作会清除中断。原始中断寄存器（RIS register）保存着看门狗的原始中断状态位。

16.4. 寄存器描述

WDT 寄存器基地址：0x4000_2C00

偏置	名称	描述
0x00	WDT_LOAD	加载寄存器
0x04	WDT_COUNT	当前计数寄存器
0x08	WDT_CTRL	控制寄存器
0x0C	WDT_FEED	喂狗寄存器
0x10	WDT_INTCLRTIME	中断清除时限寄存器
0x14	WDT_RIS	原始中断状态寄存器

16.4.1. 加载寄存器 WDT_LOAD （偏移：00h）

比特	名称	属性	复位值	描述
31:0	LOAD	R/W	0xFFFFFFFF	加载值

16.4.2. 当前计数寄存器 WDT_COUNT （偏移：04h）

比特	名称	属性	复位值	描述
31:0	COUNT	RO	0xFFFFFFFF	当前计数值。

16.4.3. 控制寄存器 WDT_CTRL（偏移：08h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	EN	R/W	0	看门狗使能 0: 禁止看门狗 1: 使能看门狗
6	MODE	R/W	0	模式 0: 复位 1: 中断
5	RSV	-	-	保留
4	INTEN	R/W	0	中断使能 0: 禁止中断 1: 使能中断
3	RSV	-	-	保留
2:0	DIVISOR	R/W	110	看门狗时钟预分频： 000: 不分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频

16.4.4. 喂狗寄存器 WDT_FEED（偏移：0Ch）

比特	名称	属性	复位值	描述
31:0	FEED	WO	0x00000000	喂狗

				写入 0xAA55A55A，计数器从装载寄存器加载装载值，加载动作会清除 WDT 中断。

16.4.5. 中断清除时限寄存器 WDT_INTCLRTIME（偏移：10h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	INTCLRT	R/W	0x1000	中断清除时限 中断产生后（即使中断未使能），在这个时限内没有去喂狗，WDT 会产生复位信号。时限的单位是看门狗工作时钟。

16.4.6. 原始中断状态寄存器 WDT_RIS（偏移：14h）

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	WDTRIS	RO	0	原始中断状态标志位。

16.5. 使用流程

16.5.1. 定时器溢出产生中断

- 配置加载寄存器；
- 配置中断清除时限寄存器；
- 配置定时器时钟预分频值；
- 配置看门狗工作模式；
- 配置中断使能位；
- 使能看门狗定时器。

注意：如果没有使能中断，则在计数到 0 溢出，经过中断清除时限寄存器值个计数

后，将产生一个系统复位。

16.5.2. 定时器溢出产生复位

- 配置加载寄存器；
- 配置中断清除时限寄存器；
- 配置定时器时钟预分频值；
- 配置看门狗工作模式；
- 配置中断使能位；
- 使能看门狗定时器。

注意：计数器计数到 0 溢出，将产生一个系统复位。

17. 独立看门狗（IWDT）

17.1. 概述

独立看门狗定时器(IWDT)有独立的时钟源(RC32K)。因此即使主时钟失效了，它仍然能保持工作状态，这非常适合于需要独立环境且对计时精度要求不高的场合。当内部向下计数器的计数值达到 0，独立看门狗会产生一个复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

17.2. 主要特性

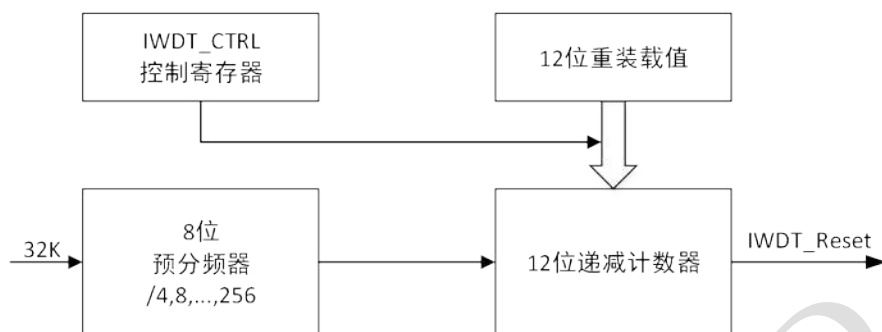
- 自由运行的 12 位向下计数器
- 计数器计数至 0x000 时产生复位
- 当递减计数器在窗口外被重新装载，则产生复位
- 可编程预分频因子和可编程装载值
- 时钟由独立的 32K 时钟提供
- 可作为 STOP 模式唤醒源

17.3. 功能描述

在命令寄存器(IWDT_CMDR)中写入 0xCCCC，开始启用独立看门狗；此时计数器开始从其复位值 0xFFFF 递减计数。当计数器计数到末尾 0x000 时，会产生一个复位信号(IWDT_RESET)。无论何时，只要在控制寄存器 IWDT_CMD 中写入 0xAAAA，IWDT_RLR 中的值就会被重新加载到计数器，从而避免产生看门狗复位。

结构框图如下：

图 17-1 IWDT 结构框图



17.3.1. 寄存器访问保护

IWDT_PR、IWDT_RLR、IWDT_WINR 和 IWDT_WUTR 寄存器具有写保护功能。要修改这几个寄存器的值，必须先向 IWDT_CMDR 寄存器中写入 0x5555 打开写保护。写保护打开后可以对多个受保护寄存器进行修改。直到再次往该寄存器写入一个不同的值来中断操作流程，此时寄存器将重新被保护。重装载操作(即写入 0xAAAA)也会启动写保护功能。状态寄存器指示 IWDT_PR、IWDT_RLR、IWDT_WINR 和 IWDT_WUTR 寄存器是否正在被更新。

17.3.2. 窗口选项

独立看门狗定时器通过配置窗口寄存器 IWDT_WINR 中的 WIN[11:0]位用来设定窗口值。当计数值大于 WIN[11:0]中值，重装载操作(IWDT_CMDR 写入 0xAAAA)会引起复位；只有当计数器的值小于窗口值，重装载向下计数器可以避免复位。

IWDT_WINR 的默认值为 0xFFF，与重装载值 IWDT_RLR 一样，因此默认不开启窗口功能。只有当 IWDT_WINR 设置为小于 IWDT_RLR 时，窗口功能才会开启。

配置 IWDT_WINR 会自动发起计数器重载为重载值 IWDT_RLR。

17.3.3. 唤醒功能

独立看门狗定时器有独立唤醒功能，开启后可以从 STOP 模式唤醒，防止产生复位。IWDT_WUTR 寄存器可以设置唤醒值，当计数到预设唤醒值后，会产生一个预分频周期的 WAKEUP 信号，通过 PMU 模块唤醒系统。IWDT_WUTR 设置需要小于 IWDT_RLR，

否则唤醒功能无法工作。此功能默认不开启，对 IWDT_CMD 写入 0x6666 开启；开启后对 IWDT_CMD 写入 0x9999 关闭。

17.4. 寄存器描述

IWDT 寄存器基地址：0x4000_3000

偏置	名称	描述
0x00	IWDT_CMDR	命令寄存器
0x04	IWDT_PR	预分频寄存器
0x08	IWDT_RLR	重装载寄存器
0x0C	IWDT_SR	状态寄存器
0x10	IWDT_WINR	窗口寄存器
0x14	IWDT_WUTR	唤醒值设置

17.4.1. 命令寄存器 IWDT_CMDR（偏移：00h）

比特	名称	属性	复位值	描述
31:0	CMD	WO	0x00000000	命令 0xAAAA：喂狗，重装载计数器。 0x5555：禁止写保护。IWDT_PR、IWDT_RLR 和 IWDT_WINR 三个寄存器具有写保护，在写寄存器之前需取消写保护。 0xCCCC：使能 IWDT 0xEF01ABCD：禁止 IWDT。 0x6666：使能唤醒功能。 0x9999：禁止唤醒功能。

注：请在启动独立看门狗后再配置寄存器。

17.4.2. 预分频寄存器 IWDT_PR (偏移: 04h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	PR	R/W	000	预分频因子 000: 4 分频 001: 8 分频 010: 16 分频 011: 32 分频 100: 64 分频 101: 128 分频 110: 256 分频 111: 256 分频 IWDT_SR 寄存器的 PVU 位为 0 时, 才允许对 PR 进行读写操作。

注: IWDT_PR 具有写保护功能, 配置 IWDT_PR 前, 需禁止写保护 (向 IWDT_CMDR 寄存器中写入 0x5555)。

17.4.3. 重装载寄存器 IWDT_RLR (偏移: 08h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	PL	R/W	0xFFFF	重装载值 当向 IWDT_CMDR 寄存器写入 0xAAAA 时, 重装载值会被传送到计数器中。随后计数器从这个值开始递减计数。独立看门狗超时周期可通过此重装载值和时钟预分频值来计算。 IWDT_SR 寄存器的 RVU 位为 0 时, 才允

				许对 PL 进行读写操作。

注: IWDT_RLR 具有写保护功能, 配置 IWDT_RL 前, 需禁止写保护 (向 IWDT_CMDR 寄存器中写入 0x5555)。

17.4.4. 状态寄存器 IWDT_SR (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
4	RLF	RO	0	喂狗完成状态 0: 喂狗完成 1: 喂狗进行中 进入 STOP 模式前, 需要等待喂狗完成。
3	WTU	RO	0	唤醒值更新状态 0: 唤醒值更新完成 1: 唤醒值更新进行中
2	WVU	RO	0	窗口值更新状态 0: 窗口值更新完成 1: 窗口值更新进行中
1	RVU	RO	0	重装载值更新状态 0: 重装载值更新完成 1: 重装载值更新进行中
0	PVU	RO	0	预分频值更新状态 0: 预分频值更新完成 1: 预分频值更新进行中

17.4.5. 窗口寄存器 IWDT_WINR (偏移: 10h)

比特	名称	属性	复位值	描述

31:12	RSV	-	-	保留
11:0	WIN	R/W	0xFFFF	窗口值 定义窗口值的上限值，用于与向下递减计数器进行比较。当计数值大于 WIN 值，重装载操作会引起复位。当计数值小于等于 WIN 值，重装载操作不会引起复位。 IWDT_SR 寄存器的 WVU 位为 0 时，才允许对 WIN 进行读写操作。

注：IWDT_WINR 具有写保护功能，配置 IWDT_WINR 前，需禁止写保护（向 IWDT_CMDR 寄存器中写入 0x5555）。

17.4.6. 唤醒寄存器 IWDT_WUTR（偏移：14h）

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	WUT	R/W	0xFFFF	唤醒值 定义唤醒值，用于定义唤醒信号产生的计数值，产生一个预分频周期的 WakeUp 信号。 IWDT_SR 寄存器的 WTU 位为 0 时，才允许对 WUT 进行读写操作。

注：IWDT_WUTR 具有写保护功能，配置 IWDT_WUTR 前，需禁止写保护（向 IWDT_CMDR 寄存器中写入 0x5555）。

17.5. 使用流程

1. 使能 IWDT

IWDT_CMDR 写入 0xCCCC。

2. 禁止写保护

IWDT_CMDR 写入 0x5555。

3. 配置预分频值

IWDT_PR 写入预分频值。

4. 配置重装载值

等待 IWDT_SR 的 PVU 为 0，IWDT_RLR 写入重装载值。

5. 配置唤醒值

如果启用唤醒功能：

等待 IWDT_SR 的 RVU 为 0，IWDT_WUTR 写入唤醒值，配置唤醒模式（EXTI）。

如果未启用唤醒功能：

等待 IWDT_SR 的 RVU 为 0，IWDT_WUTR 写入 0xFFF。

6. 配置默认窗口值

如果未启用窗口功能：

等待 IWDT_SR 的 WVU 为 0，IWDT_WINR 写入 0xFFF。

7. 等待所有配置完成

等待 IWDT_SR 为 0。

8. 使能唤醒功能

如果启用唤醒功能：

IWDT_CMDR 写入 0x6666。

9. 配置窗口值

如果启用窗口功能

IWDT_WINR 写入窗口值。

10. 重装载

如果未启用窗口功能：

IWDT_CMDR 写入 0xAAAA。

11. 等待所有配置完成

等待 IWDT_SR 为 0。

18. 实时时钟（RTC）

18.1. 概述

RTC 提供秒、分、时、日、周、月、年的信息，每月的天数和闰年的天数可自动调整。时间和日期都以 BCD 码的形式显示。RTC 可以进行夏令时补偿。RTC 可以工作在省电模式下，并通过软件配置来智能唤醒。RTC 支持外接更高精度的低频时钟，以达到更高的日历精度

18.2. 主要特性

- 可计算 00~99 年之间的秒、分、时、日、周、月、年
- BCD 时间显示
- 自动闰年调整
- 数字校准功能：通过调整最小时间单位（最大可调精度 0.95ppm）来进行日历校准，调校后理论精度 $\pm 0.477\text{ppm}$
- 周期唤醒中断
- 闹钟功能
- 可从管脚输出 XTALF 时钟信号供用户校准
- RTC 计时器部分不复位
- 2 路输入上下沿(侵入)时间戳功能
- 16 个 32 位 (共 64 字节) 通用备份寄存器，能够在省电模式下保存数据。当有外部侵入事件发生时，备份寄存器可复位（可配置）。

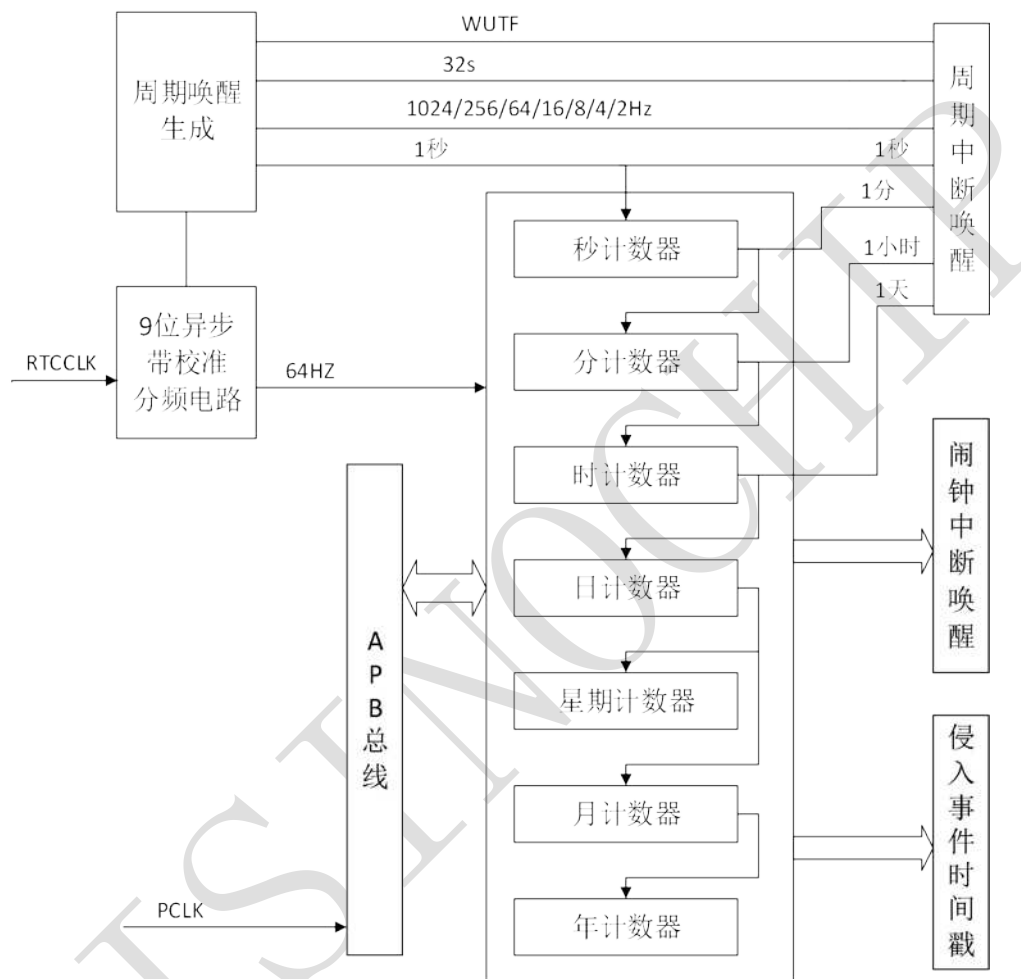
18.3. RTC 功能描述

实时时钟的时钟源可配置为外部低速晶振，外部高速晶振，内部低速 RC；默认使用外部低速晶振。RTC 中断使能寄存器 IE、闹钟寄存器 ALARM、侵入时间戳使能只受上电复位控制，其他复位源不能复位这三个控制寄存器。其他数据寄存器上电状态不定，

上电后需要初始化，不受任何复位影响。所有软件写入和读取的日期时间值都为 BCD 码，无须十六进制转换为十进制。

整体框图如下：

图 18-1 RTC 内部结构框图



18.3.1. RTC 时间设置

RTC 模块提供时间写保护功能，必须先对写保护寄存器写入 0xCA53CA53，才能改写时间寄存器；软件写入除 0xCA53CA53 外的任意值来禁止时间寄存器的写入，恢复写保护。

软件应在秒中断发生后再去置时，计时更加准确。当软件写入秒时间时，硬件自动清除秒内计数器。

推荐的 RTC 时间设置流程如下：

- 等待秒中断时间发生
- 打开写保护
- 连续写入年月日时分秒寄存器
- 读出时间寄存器进行校验
- 使能写保护

注：秒、分、时、天、周、月、年不会被复位

18.3.2. RTC 时间读取

有两种读取方式

方式 1：任意时刻读取方式

1. 读出分，时，周，日，月，年计数寄存器值；
2. 读出秒计数寄存器值；
3. 再次读出秒计数寄存器值；
4. 判断两次秒的读出值是否相同，不同重新从第一步开始，相同读取结束。

方式 2：中断读取方式

在 RTC 周期中断服务中读取秒，分，时，周，日，月，年计数寄存器值。因为中断发生后到下次数据改变至少 1s 的时间。

18.3.3. 闹钟功能

闹钟功能可以设定一个星期内，任意几天（0 到 7 天）定时唤醒，可以精确到秒级别。软件先设置好闹钟秒寄存器，闹钟分寄存器和闹钟小时寄存器的值。当 RTC 日历中的秒计数器，分计数器和小时计数器的值分别与闹钟寄存器的值相等时，产生闹钟中断。

1. 设定 CR.ALM_EN=0，闹钟关闭
2. 设定 ALM_IE =1，闹钟中断禁止；
3. 周闹钟 ALM WEEK 设定，设置每周几闹钟工作；秒闹钟 ALM SEC，分闹钟 ALM MIN，时闹钟 ALM HOUR，

4. 设定 CR.ALM_EN=1，闹钟许可；
5. 等待发生中断；
6. 由于闹钟中断和周期中断共用中断请求信号，当 SR.ALM_IF=1 时，进入闹钟中断处理。

18.3.4. 时钟误差补偿

由于计数器采用 32.768KHz 的时钟计数，如果需要对每秒精度进行补偿时，只能按照 32.768KHz 的整数周期补偿，则每秒补偿的最小单位为 $(1/32768) * 10^6 = 30.5\text{ppm}$ ，无法满足高精度的要求。

那么要在 32.768KHz 的计数时钟下实现精度较高的时钟补偿时，需要在算法上做调整，将最大补偿周期扩大 32 倍。则在只能补偿的最小单位为 30.5ppm 的情况下，平均到每秒的补偿单位变为 $30.5\text{ppm}/32 = 0.96\text{ppm}$ ，最大调校范围为 $\pm(511 * 30.517\text{us}/32\text{s}) = \pm 487\text{ppm}$ 。满足了精度较高的时钟补偿要求。而且补偿发生在每 32 秒内比较均匀的范围。

调校值由 10bit 寄存器组成，其中最高位为符号位，表示计数值增减，其余 9bit 表示增减的绝对值。为了提高每秒的平均精度，避免较大的秒间跃变，采取将 32s 调校值平均分配到每秒内的做法，其实现方法如下：

除了最高符号位，其余 9bit 可分为高 4bit 的公共值和 5bit 私有值，其中公共值表示 32s 内每秒都要调整的值，私有值表示 32s 内部分秒需要加减 1。

Bit9	Bit[8:5]	Bit[4:0]
Sign	Common Value (C)	Differential Value (D)

调校值公式可表示为：Correction = (C*32 + D)*30.517/32000000

假设只使时钟增加 0.952ppm，即 32s 周期只增加一个 30.5us，调校值写为 0_0000_00001，所以公共值为 0，私有值为 1，只需要对 32s 内的一个秒周期加 1 即可；假设增加 487ppm，即 32s 周期内增加 511 个 30.5us，调校值写为 0_1111_11111，公共值为 15，私有值为 31，表示 32s 中每秒都要加 15，同时其中还有 31s 需要额外加 1。

调校值举例：

ppm	ADJUST	Common	Differential	Expression
0.953	0_0000_00001	0	1	$1*30.517/32000000$
-125.88	1_0100_00100	4	4	$(4*32+4)*30.517/32000000$
32.42	0_0001_00010	1	2	$(1*32+2)*30.517/32000000$
487	0_1111_11111	15	31	$(15*32+31)*30.517/32000000$

通过以上方式，可以得到平滑调整的秒周期，秒和秒之间最大只差 30.5ppm，可以避免集中式调整引入的秒周期抖动。

18.3.5. FSEL_OUT

RTC 模块可以从 FSEL_OUT 输出内部的频率信号或者闹钟信号到管脚上用于调试和校准。具体输出信号由 RTC_CR 寄存器的 FSEL 选择。

18.3.6. 周期中断唤醒

RTC 模块提供多种周期唤醒，除了包括 1 秒、1 分、1 小时、1 天日历唤醒外，还提供 1024Hz、256Hz、64Hz、16Hz、8Hz、4Hz、2Hz 和 32s 等多种唤醒方式。

18.3.7. 唤醒定时器

RTC 模块提供一个可任意设置的定时唤醒定时器（WUTIMER），由 16 位可编程自动重载递减定时器组成。可通过 RTC_CR 寄存器中的 WUTE 位来使能此唤醒定时器功能。

唤醒定时器的时钟输入可以是：

- 2、4、8 或 16 分频的 RTC 时钟 (RTCCLK)。当 RTCCLK 为 XTL (32.768 kHz) 时，可配置的唤醒中断周期介于 122 μ s 和 32 s 之间，且分辨率低至 61 μ s。
- 0.5Hz、1Hz、2Hz 信号，可配置的唤醒中断周期为 0.5s 和 36 小时之间。

完成初始化后，定时器开始从 RTC_WUTR 递减计数。在低功耗模式下使能唤醒功能时，递减计数保持有效。此外，当计数器计数到 0 时，RTC_ISR 寄存器的 WUTF 标志会置 1，并且唤醒寄存器会使用其重载值（RTC_WUTR 寄存器值）自动重载。之后必须用软件清零 WUTF 标志。

通过将 RTC_IE 寄存器中的 WUTIE 位置 1 来使能周期性唤醒中断时，它会使器件退出低功耗模式。

要配置或更改唤醒定时器的自动重载值（RTC_WUTR 中的 WUT[15:0]），需要按照以下顺序操作：

1. 清零 RTC_CR 中的 WUTE 以禁止唤醒定时器。
2. 轮询 RTC_ISR 中的 WUTWF，直到该位置 1，以确保可以访问唤醒自动重载定时器和 WUCKSEL[2:0] 位。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 编程唤醒自动重载值 WUT[15:0]，并选择唤醒时钟（RTC_CR 中的 WUCKSEL[2:0] 位）。将 RTC_CR 寄存器中的 WUTE 位置 1 以再次使能定时器。唤醒定时器重新开始递减数。由于时钟同步的缘故，在 WUTE 置 1 后，WUTWF 位也会清零，但需要花费 2 个 RTCCLK 时钟周期。

如果已通过 RTC_CR 寄存器的 FSEL 位选择 WUT 周期性唤醒连接到 FSEL_OUT 输出。

18.3.8. 侵入检测和备份寄存器

备份寄存器处于备份域中。待机模式唤醒或系统复位操作都不会影响这些寄存器。只有当被检测到有侵入事件和备份域复位时，这些寄存器才会复位。

RTC 支持两个外部 IO 侵入事件检测，侵入检查通过 TAMPxEN 使能对应 IO 检测功能。两个侵入 IO 各有一组独立 STAMP 寄存器组（CLKSTAMPx 和 CALSTAMPx），通过设置 TSxEDGE 选择记录上升沿或者下降沿时间戳，当侵入 IO 出现选择的上升沿或下降沿时，RTC 会自动记录当前时间到 STAMP 寄存器组中。侵入 IO 有上升沿相应的标志，可用于产生中断或者供软件查询。为了确保输入检测的可靠性，建议使能外部 IO 的输入数字滤波，通过 TAMPxFLTEN 使能某个通道滤波功能，设置 TAMPxFLT 选择滤波周期。滤波时钟可以选择 RTC 时钟或者 64HZ 时钟，滤波时钟两个 IO 使用同一

种配置。备份寄存器是否同时清除可以由 TAMPxFCLR（下降沿）TAMPxRCLR（上升沿）控制。

侵入检查时间戳在相应标志寄存器为 0 的情况下记录事件发生时间，如果对应标志已经为 1，则忽略相应事件。因此如果有多次事件发生，时间戳仅记录第一次事件发生的时间，除非软件在事件发生后清除了标志寄存器。

18.4. 寄存器描述

RTC 寄存器基地址：0x4000_2800

偏置	名称	描述
0x00	RTC_WP	写保护寄存器
0x04	RTC_IE	中断使能寄存器
0x08	RTC_SR	中断标志寄存器
0x0C	RTC_SEC	秒计数寄存器
0x10	RTC_MIN	分计数寄存器
0x14	RTC_HOUR	时计数寄存器
0x18	RTC_DAY	日计数寄存器
0x1C	RTC_WEEK	周计数寄存器
0x20	RTC_MONTH	月计数寄存器
0x24	RTC_YEAR	年计数寄存器
0x28	RTC_ALM	闹钟寄存器
0x2C	RTC_CR	控制寄存器
0x30	RTC_ADJUST	时钟误差补偿寄存器
0x44	RTC_CLKSTAMP1	时间戳 1
0x48	RTC_CALSTAMP1	日历戳 1
0x4C	RTC_CLKSTAMP2	时间戳 2
0x50	RTC_CALSTAMP2	日历戳 2
0x54	RTC_WUTR	唤醒定时器寄存器
0x70~0xAC	RTC_BAKUP0~15	备份寄存器 0~15

18.4.1. 写保护寄存器 RTC_WP (偏移: 00h)

比特	名称	属性	复位值	描述
31:0	WE	RW	0	RTC 写使能命令, 当 CPU 向 RTC_WP 写入 0x CA53CA53 时, 允许 CPU 向 RTC 的时间/日期寄存器写入初值,这时 WE 读为 1;当 CPU 向 RTC_WP 写入不为 0x CA53CA53 的任意值时恢复写保护,这时读 WE 为 0。

18.4.2. 中断使能寄存器 RTC_IE (偏移: 04h)

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	WUTIE	RW	0	唤醒定时器中断使能 0: 禁止唤醒定时器中断 1: 使能唤醒定时器中断
16	STP2RIE	RW	0	RTC STAMP2 上升沿事件中断使能 0: 禁止中断 1: 使能中断
15	STP2FIE	RW	0	RTC STAMP2 下降沿事件中断使能 0: 禁止中断 1: 使能中断
14	STP1RIE	RW	0	RTC STAMP1 上升沿事件中断使能 0: 禁止中断 1: 使能中断
13	STP1FIE	RW	0	RTC STAMP1 下降沿事件中断使能 0: 禁止中断 1: 使能中断

12	ADJ32_IE	RW	0	32 秒中断使能。 1: 中断使能打开 0: 中断使能禁止
11	ALM_IE	RW	0	闹钟中断使能。 1: 中断使能打开 0: 中断使能禁止
10	1KHZ_IE	RW	0	1khz 中断使能。 1: 中断使能打开 0: 中断使能禁止
9	256HZ_IE	RW	0	256hz 中断使能。 1: 中断使能打开 0: 中断使能禁止
8	64HZ_IE	RW	0	64hz 中断使能。 1: 中断使能打开 0: 中断使能禁止
7	16HZ_IE	RW	0	16hz 中断使能。 1: 中断使能打开 0: 中断使能禁止
6	8HZ_IE	RW	0	8hz 中断使能。 1: 中断使能打开 0: 中断使能禁止
5	4HZ_IE	RW	0	4hz 中断使能。 1: 中断使能打开 0: 中断使能禁止
4	2HZ_IE	RW	0	2hz 中断使能。 1: 中断使能打开 0: 中断使能禁止
3	SEC_IE	RW	0	秒中断使能。 1: 中断使能打开

				0: 中断使能禁止
2	MIN_IE	RW	0	分中断使能。 1: 中断使能打开 0: 中断使能禁止
1	HOUR_IE	RW	0	小时中断使能。 1: 中断使能打开 0: 中断使能禁止
0	DATE_IE	RW	0	天中断使能。 1: 中断使能打开 0: 中断使能禁止

18.4.3. 中断标志寄存器 RTC_SR (偏移: 08h)

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
24	WUTWF	RO	1	唤醒定时器写标志 此位在 RTC_CR 中的 WUTE 位清零后, 并在 2 个 RTCCLK 周期后由硬件置 1, 在 WUTE 位置 1 后并在 2 个 RTCCLK 周期后清零。当 WUTE 位清零且 WUTWF 位置 1 时, 可更改唤醒定时器的值。 0: 不允许更新唤醒定时器配置 1: 允许更新唤醒定时器配置
23:18	RSV	-	-	保留
17	WUTF	RC_W1	0	唤醒定时器标志 (Wakeup timer flag) 当唤醒自动重载计数器计数到 0 时, 由硬件将此标志置 1。写 1 清零 1: 中断置位 0: 无中断产生
16	STP2RIF	RC_W1	0	RTC TAMP2 上升沿事件中断标志。写 1 清零

				1: 中断置位 0: 无中断产生 此寄存器为 1 的情况下时间戳不再记录新的上升沿事件
15	STP2FIF	RC_W1	0	RTC TAMP2 下降沿事件中断标志。写 1 清零 1: 中断置位 0: 无中断产生 此寄存器为 1 的情况下时间戳不再记录新的下降沿事件
14	STP1RIF	RC_W1	0	RTC TAMP1 上升沿事件中断标志。写 1 清零 1: 中断置位 0: 无中断产生 此寄存器为 1 的情况下时间戳不再记录新的上升沿事件
13	STP1FIF	RC_W1	0	RTC TAMP1 下降沿事件中断标志。写 1 清零 1: 中断置位 0: 无中断产生 此寄存器为 1 的情况下时间戳不再记录新的下降沿事件
12	ADJ32_IF	RC_W1	0	32S 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
11	ALM_IF	RC_W1	0	闹钟中断标志。写 1 清零 1: 中断置位 0: 无中断产生
10	1KHZ_IF	RC_W1	0	1khz 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
9	256HZ_IF	RC_W1	0	256hz 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
8	64HZ_IF	RC_W1	0	64hz 中断标志。写 1 清零 1: 中断置位

				0: 无中断产生
7	16HZ_IF	RC_W1	0	16hz 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
6	8HZ_IF	RC_W1	0	8hz 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
5	4HZ_IF	RC_W1	0	4hz 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
4	2HZ_IF	RC_W1	0	2hz 中断标志。写 1 清零 1: 中断置位 0: 无中断产生
3	SEC_IF	RC_W1	0	秒中断标志。写 1 清零 1: 中断置位 0: 无中断产生
2	MIN_IF	RC_W1	0	分中断标志。写 1 清零 1: 中断置位 0: 无中断产生
1	HOUR_IF	RC_W1	0	小时中断标志。写 1 清零 1: 中断置位 0: 无中断产生
0	DATE_IF	RC_W1	0	天中断标志。写 1 清零 1: 中断置位 0: 无中断产生

18.4.4. 秒计数寄存器 RTC_SEC (偏移: 0Ch)

比特	名称	属性	复位值	描述

31:7	RSV	-	-	保留
6:0	BCDSEC	RW	X	秒时间数值，BCD 格式。

18.4.5. 分钟计数寄存器 RTC_MIN (偏移：10h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6:0	BCDMIN	RW	X	分钟时间数值，BCD 格式。

18.4.6. 小时计数寄存器 RTC_HOUR (偏移：14h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:0	BCDHOUR	RW	X	小时数值，BCD 格式。

18.4.7. 日计数寄存器 RTC_DAY (偏移：18h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:0	BCDDATE	RW	X	天数值，BCD 格式。

18.4.8. 周计数寄存器 RTC_WEEK (偏移：1Ch)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	BCDWEEK	RW	X	周数值，BCD 格式。

18.4.9. 周计数寄存器 RTC_MONTH (偏移: 20h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:0	BCDMONTH	RW	X	月数值, BCD 格式。

18.4.10. 年计数寄存器 RTC_YEAR (偏移: 24h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	BCDYEAR	RW	X	年数值, BCD 格式。

18.4.11. 闹钟寄存器 RTC_ALM (偏移: 28h)

比特	名称	属性	复位值	描述
31	ALM_WDS	RW	0	闹钟星期/天选择 0: 选择闹钟星期模式 1: 选择闹钟日模式
30:24	ALM WEEK/ ALM DAY	RW	00	闹钟的星期数值/日数值 当为星期模式时: b24:b30 分别对应周日:周六, 对应为置 '1' 时, 代表每周该日闹钟有效。 如, b24=1, b30=1 代表周日和周六闹钟设定有效 当为日模式时: [29:24]: 闹钟的日 BCD 格式
23:22	RSV	-	-	保留
21:16	ALMHOUR	RW	00	闹钟的小时 BCD 格式。
15	RSV	-	-	保留
14:8	ALMMIN	RW	00	闹钟的分 BCD 格式。

7	RSV	-	-	保留
6:0	ALMSEC	RW	00	闹钟的秒 BCD 格式

18.4.12. 控制寄存器 RTC_CR (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:27	RSV	-	-	保留
26:24	WUCKSEL[2:0]	RW	0	唤醒时钟选择 000: 选择 RTCCLK/16(RTCCLK 16 分频) 001: 选择 RTCCLK/8(RTCCLK 8 分频) 010: 选择 RTCCLK/4(RTCCLK 4 分频) 011: 选择 RTCCLK/2(RTCCLK 2 分频) 100: 选择 1Hz 101: 选择 0.5Hz 其它: 选择 2Hz
23	WUTE	RW	0	唤醒定时器使能 0: 禁止唤醒定时器 1: 使能唤醒定时器
22	TAMPFLTCLK	RW	0	侵入信号滤波时钟选择 0: RTCCLK 1: 512 个 RTCCLK (64Hz)
21	TS2EDGE	RW	0	TAMP2 记录 STAMP2 边沿选择, 只影响 STAMP2 时间戳记录的边沿选择, 不影响 TAMP2 的 STP2FIF 和 STP2RIF 标志产生。 0: STAMP2 选择记录 TAMP2 上升沿 1: STAMP2 选择记录 TAMP2 下降沿
20:19	TAMP2FLT	RW	00	侵入信号 2 滤波周期 00: 1 个 RTCCLK 或 64Hz 01: 2 个 RTCCLK 或 64Hz

				10: 4 个 RTCCLK 或 64Hz 11: 8 个 RTCCLK 或 64Hz
18	TAMP2FLTEN	RW	0	侵入信号 2 滤波使能 0: 滤波不使能 1: 滤波使能
17	TAMP2FCLR	RW	0	侵入 2 下降沿清除备份寄存器 0: 不清除备份寄存器 1: 清除备份寄存器
16	TAMP2RCLR	RW	0	侵入 2 上升沿清除备份寄存器 0: 不清除备份寄存器 1: 清除备份寄存器
15	TS1EDGE	RW	0	TAMP1 记录 STAMP1 边沿选择, 只影响 STAMP1 时间戳记录的边沿选择, 不影响 TAMP1 的 STP1FIF 和 STP1RIF 标志产生。 0: STAMP1 选择记录 TAMP1 上升沿 1: STAMP1 选择记录 TAMP1 下降沿
14:13	TAMP1FLT	RW	00	侵入信号 1 滤波周期 00: 1 个 RTCCLK 或 64Hz 01: 2 个 RTCCLK 或 64Hz 10: 4 个 RTCCLK 或 64Hz 11: 8 个 RTCCLK 或 64Hz
12	TAMP1FLTEN	RW	0	侵入信号 1 滤波使能 0: 滤波不使能 1: 滤波使能
11	ALM_MSKD	RW	0	闹钟星期/天数值功能屏蔽位 0: 闹钟不屏蔽星期/天数值比较 1: 闹钟屏蔽星期/天数值比较
10	ALM_MSKH	RW	0	闹钟时数值屏蔽位 0: 闹钟不屏蔽时数值比较

				1: 闹钟屏蔽时数值比较
9	ALM_MSKM	RW	0	闹钟分数字屏蔽位 0: 闹钟不屏蔽分数值比较 1: 闹钟屏蔽分数值比较
8	TAMP1FCLR	RW	0	侵入 1 下降沿清除备份寄存器 0: 不清除备份寄存器 1: 清除备份寄存器
7	TAMP1RCLR	RW	0	侵入 1 上升沿清除备份寄存器 0: 不清除备份寄存器 1: 清除备份寄存器
6	TAMP2EN	RW	0	侵入 2 时间戳功能使能位。 1: 打开时间戳 0: 关闭时间戳
5	TAMP1EN	RW	0	侵入 1 时间戳功能使能位。 1: 打开时间戳 0: 关闭时间戳
4	ALM_EN	-	-	闹钟功能使能 0: 不使能闹钟 1: 使能闹钟
3:0	FSEL	RW	0000	频率输出选择信号: 4'b0000: 保留 4'b0001: 保留 4'b0010: 输出秒计数器进位信号, 高电平宽度 1s 4'b0011: 输出分计数器进位信号, 高电平宽度 1s 4'b0100: 输出小时计数器进位信号, 高电平宽度 1s 4'b0101: 输出天计数器进位信号, 高电平宽度 1s 4'b0110: 输出闹钟匹配信号 4'b0111: 输出 32 秒方波信号 4'b1000: 输出 WUT 唤醒信号

				4'b1001: 反向输出秒计数器进位信号 4'b1010: 反向输出分计数器进位信号 4'b1011: 反向输出小时计数器进位信号 4'b1100: 反向输出天计数器进位信号 4'b1101: 反向输出闹钟匹配信号 4'b1110: 反向输出 WUT 唤醒信号 4'b1111: 输出 RTC 内部秒时标方波

18.4.13. 时钟误差补偿寄存器 RTC_ADJUST (偏移: 30h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	ADJSIGN	RW	X	补偿调整方向 0: 增加 1: 减少
8:0	ADJVALUE	RW	XX	补偿调整数值

18.4.14. 时间戳 1 RTC_CLKSTAMP1 (偏移: 44h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:16	HRSTP1	RO	0	检测到 TAMP1 后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP1	RO	0	检测到 TAMP1 后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP1	RO	0	检测到 TAMP1 后存储 BCD 秒寄存器的值

18.4.15. 日历戳 1 RTC_CALSTAMP1 (偏移: 48h)

比特	名称	属性	复位值	描述
31:24	YEARSTP1	RO	0	检测到 TAMP1 后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP1	RO	0	检测到 TAMP1 后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP1	RO	0	检测到 TAMP1 后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP1	RO	0	检测到 TAMP1 后存储 BCD 日寄存器的值

18.4.16. 时间戳 2 RTC_CLKSTAMP2 (偏移: 4Ch)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
21:16	HRSTP2	RO	0	检测到 TAMP2 后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP2	RO	0	检测到 TAMP2 后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP2	RO	0	检测到 TAMP2 后存储 BCD 秒寄存器的值

18.4.17. 日历戳 2 RTC_CALSTAMP2 (偏移: 50h)

比特	名称	属性	复位值	描述
31:24	YEARSTP2	RO	0	检测到 TAMP1 后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP2	RO	0	检测到 TAMP2 后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP2	RO	0	检测到 TAMP2 后存储 BCD 周寄存器的值。

7:6	RSV	-	-	保留
5:0	DAYSTP2	RO	0	检测到 TAMP2 后存储 BCD 日寄存器的值

18.4.18. 唤醒定时器寄存器 RTC_WUTR (偏移：54h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	WUT	RW	0xFFFF	唤醒自动重载值位 (Wakeup auto-reload) 当使能唤醒定时器时 (WUTE 置 1)，每 (WUT[15:0] + 1) 个 ck_wut 周期将 WUTF 标志置 1 一次。 ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0] 位进行选择

18.4.19. 备份寄存器 0~15 RTC_BAKUP0~15 (偏移：70~ACh)

比特	名称	属性	复位值	描述
31:0	BAKUP	RW	00	备份寄存器

18.5. 使用流程

1. 使能 PMU 模块时钟 RCC.APB1ENR.PMUCKEN=1 和 RTC 模块总线时钟 RCC.APB1ENR.RTCCKEN=1;
2. RTC 模块写使能 PMU.CTL0.RTC_WE=1;
3. 选择 RTCCLK 时钟源 RCC.STDBYCTL.RTCSEL，当时钟源选择 RC32K 时，需要使能 RC32K 时钟 RCC.STDBYCTL.RC32EN=1；当选择外部低速晶振 XTL 时，需要使能 XTL 振荡器 RCC.STDBYCTL.XTLEN=1;
4. RTC 写使能 RTC.WP.WE=0xCA53CA53;
5. 写入年、月、日、时、分、秒。

19. 通用输入输出接口（GPIO）

19.1. 概述

每组 GPIO 包含 16 个通用数据输入输出接口，这些管脚可以与其他功能管脚共享，这取决于芯片的配置。通过这些数据接口，可以配置任意数目的管脚作为 EXTI 中断信号输入。

复用功能（AF）的备用引脚，极大提高了端口利用的灵活性。GPIO 引脚通过配置相关的寄存器可以用作复用功能输入/输出引脚。

每个 GPIO 引脚可以独立配置为输出（推挽或开漏）、输入、外设复用功能或模拟模式。每个 GPIO 引脚可以独立配置为上拉、下拉或浮空。

19.2. 主要特性

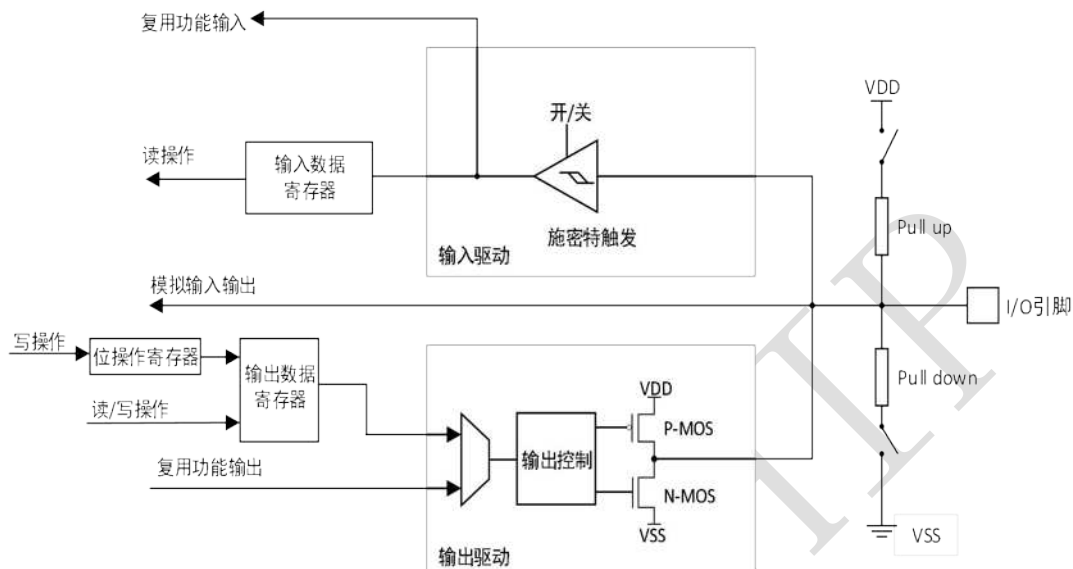
- 所有输入/输出引脚方向都可以通过软件进行配置；
- 支持施密特触发器输入
- 每个引脚具有弱上/下拉功能
- 支持推挽/开漏输出
- 支持置位/清零输出功能，可按位操作
- 支持模拟输入/输出配置
- 所有 GPIO 引脚可作为 EXTI 输入，且触发边沿可配置
- 支持复用功能输入/输出配置
- 支持端口配置锁定

19.3. 功能描述

在复位后，除 Wakeup 唤醒引脚、Debug 调试引脚、MCO 功能引脚为数字模式外，大部分 GPIO 配置为模拟模式（数字功能失效，上下拉电阻失效）。

19.3.1. 结构框图

图 19-1 GPIO 管脚结构框图



19.3.2. 数字输入功能

当 GPIO 配置为输入功能时：

- 施密特触发输入使能。
- 可选择弱上拉或弱下拉。
- I/O 引脚的状态会不断采集并保存在 GPIOx_IDATA 寄存器中。

19.3.3. 数字输出功能

- 可选择弱上拉或弱下拉。
- 可选择推挽输出或开漏输出。
- 可配置输出驱动能力
- 可通过配置 GPIO_BSC(SET、CLR)控制 I/O 引脚状态或者通过 GPIOx_ODATA 控制 I/O 引脚状态。

19.3.4. 模拟功能

- 弱上拉和下拉禁止
- 输出缓存器被禁止
- 施密特触发输入被禁止
- 读取输入数据寄存器时数值为 0

19.3.5. 复用功能

当端口配置为 AFIO 时, GPIO 引脚用作外设复用功能; 使用复用功能前, 先对 AFIO 配置寄存器和输出类型寄存器进行操作, 复用输入或输出由外设决定。

19.3.6. 外部中断/事件线

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。

19.3.7. GPIO 锁定功能

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序列 (详见 GPIOx_LOCK 寄存器描述), 在下一次复位之前, 不能再更改端口的配置。

被锁定的寄存器包括: GPIOx_MD、GPIOx_OTYP、GPIOx_PUPD、GPIOx_AF0、GPIOx_AF1、GPIOx_DS0、GPIOx_DS1、GPIOx_SMIT、GPIOx_AIEN。

19.4. 寄存器描述

GPIO 寄存器基地址:

GPIOA: 0x4800_0000

GPIOB: 0x4800_0400

GPIOC: 0x4800_0800

GPIOD: 0x4800_0C00

GPIOE: 0x4800_1000

GPIOF: 0x4800_1400

偏置	名称	描述
0x00	GPIOx_MD	模式寄存器
0x04	GPIOx_OTYP	输出类型寄存器
0x08	GPIOx_PUPD	上下拉寄存器
0x0C	GPIOx_IDATA	输入引脚映射寄存器
0x10	GPIOx_ODATA	输出引脚映射寄存器
0x14	GPIOx_BSC	输出置位/清零寄存器
0x18	GPIOx_AF0	复用功能配置寄存器 0
0x1C	GPIOx_AF1	复用功能配置寄存器 1
0x20	GPIOx_DS0	驱动能力配置寄存器 0
0x24	GPIOx_DS1	驱动能力配置寄存器 1
0x28	GPIOx_SMIT	施密特使能寄存器
0x2C	GPIOx_LOCK	配置锁定寄存器
0x30	GPIOx_AIEN	模拟开关配置寄存器

19.4.1. 模式寄存器 GPIOx_MD (x=A to F) (偏移: 00h)

比特	名称	属性	复位值	描述
31:0	MD[15:0][1:0]	R/W	GPIOA: 0xABFEFFFF GPIOB: 0xFFFFFEBF GPIOC: 0xFFFFFFFF GPIOD: 0xFFFFFAFF GPIOE: 0xFFFFFFFF GPIOF: 0x000003BF	GPIO 的 PINy(y=15 to 0)模式配置位: 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

19.4.2. GPIO 输出类型寄存器 GPIOx_OTYP (x=A to F) (偏移: 04h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	OTYP	R/W	0x0000	GPIOx 的 PINy(y=15 to 0)输出类型配置位: 0: 输出推挽模式 1: 输出开漏模式

19.4.3. GPIO 上下拉寄存器 GPIOx_PUPD (x=A to F) (偏移: 08h)

比特	名称	属性	复位值	描述
31:0	PUPD[15:0][1:0]	R/W	GPIOA: 0x64000000 GPIOB: 0x00000180 GPIOC~E: 0x00000000 GPIOF: 0x00000040	GPIOx 的 PINy(y=15 to 0)上拉/下拉配置位: 00: 无上/下拉 01: 上拉 10: 下拉 11: 保留

19.4.4. 输入引脚映射寄存器 GPIOx_IDATA (x=A to F) (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	IDATA	RO	0x0000	GPIOx 的 PINy(y=15 to 0)输入引脚映射寄存器: 当 GPIO 方向为输入有效, 读获得外部引脚值; 此寄存器为只读寄存器。

19.4.5. 输出引脚映射寄存器 GPIOx_ODATA (x=A to F) (偏移: 10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ODATA	R/W	0x0000	GPIOx 的 PINy(y=15 to 0)输出引脚映射寄存器: 当 GPIO 方向为输出有效, 写直接写至外部引脚, 读获得外部引脚值。

19.4.6. 输出置位/清零寄存器 GPIOx_BSC (x=A to F) (偏移: 14h)

比特	名称	属性	复位值	描述
31:16	CLR	WO	0x0000	GPIOx 的 PINy(y=15 to 0)输出清零寄存器: 0: 无效操作; 1: 当 IO 为输出时, IO 清零。 注: 如果同时设置了 GPIO_SET 和 GPIO_CLR 的对应位, GPIO_SET 位起作用。
15:0	SET	WO	0x0000	GPIOx 的 PINy(y=15 to 0)输出置位寄存器: 0: 无效操作; 1: 当 IO 为输出时, IO 置位。

19.4.7. 复用功能配置寄存器 0 GPIOx_AF0 (x=A to F) (偏移: 18h)

比特	名称	属性	复位值	描述
31:0	AFSEL[7:0] [3:0]	R/W	0x00000000	GPIOx 的 PINy(y=7 to 0)复用功能配

				置位 0000~1111: AF0~AF15

19.4.8. 复用功能配置寄存器 1 GPIOx_AF1 (x=A to F) (偏移: 1Ch)

比特	名称	属性	复位值	描述
31:0	AFSEL[15:8] [3:0]	R/W	0x00000000	GPIOx 的 PINy(y=15 to 8)复用功能配置位 0000~1111: AF0~AF15

19.4.9. 驱动能力配置寄存器 0 GPIOx_DS0 (x=A to F) (偏移: 20h)

比特	名称	属性	复位值	描述
31:0	DS[7:0] [3:0]	R/W	0x00000000	GPIOx 的 PINy(y=7 to 0) 驱动能力 5V 耐压 IO 的驱动能力配置: 0000: 2mA 0001: 4mA 0010: 6mA 0011: 8mA 0100: 10mA 0101: 12mA 0110: 14mA 0111: 16mA 1xxx: 2mA 非 5V 耐压 IO 的驱动能力配置: 0000: 3mA 0001: 6mA

				0010: 9mA 0011: 12mA 0100: 15A 0101: 18mA 0110: 21mA 0111: 24mA 1xxx: 3mA 以上为 VDD=3.3V 时驱动能力。其他供电电压的驱动能力请查阅相应芯片的 datasheet。

19.4.10. 驱动能力配置寄存器 1 GPIOx_DS1 (x=A to F) (偏移: 24h)

比特	名称	属性	复位值	描述
31:0	DS[15:8] [3:0]	R/W	0x00000000	GPIOx 的 PINy(y=15 to 8) 驱动能力 5V 耐压 IO 的驱动能力配置: 0000: 2mA 0001: 4mA 0010: 6mA 0011: 8mA 0100: 10mA 0101: 12mA 0110: 14mA 0111: 16mA 1xxx: 2mA 非 5V 耐压 IO 的驱动能力配置: 0000: 3mA

				0001: 6mA 0010: 9mA 0011: 12mA 0100: 15A 0101: 18mA 0110: 21mA 0111: 24mA 1xxx: 3mA 以上为 VDD=3.3V 时驱动能力。 其他供电电压的驱动能力请查阅 相应芯片的 datasheet。

19.4.11. 施密特使能寄存器 GPIOx_SMIT (x=A to F) (偏移: 28h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	SMTEN	R/W	0x0000	GPIOx 的 PINy(y=15 to 0)施密特 输入使能位: 0: 禁止 1: 使能

19.4.12. 配置锁定寄存器 GPIOx_LOCK (x=A to F) (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	LOCKKEY	R/W	0	锁定键。该位可随时读出, 它只可 通过锁定键写入序列修改。 0: 端口配置锁定键位无效

				1：端口配置锁定键位有效； GPIOx_LOCK 寄存器被锁住，直到下次系统复位或 GPIOx 模块外设复位。 锁定键的写入序列（写 LOCKKEY）： 写 1 -> 写 0 -> 写 1 -> 读 0 -> 读 1 最后一个读可省略，但可以用来确认锁定键已被激活。 注：在进行锁定键的写入序列时，不能改变 LOCKEN[15:0]的值；锁定键写入序列中的任何错误都将不能激活锁定键。
15:0	LOCKEN	R/W	0x0000	GPIOx 的 PINy(y=15 to 0)配置锁定使能位： 0：禁止 1：使能 注：当 LOCKKEY=0 时该位可写入

19.4.13. 模拟开关配置寄存器 GPIOx_AIEN (x=A to F) (偏移：30h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	AIEN	R/W	0x0000	GPIOx 的 PINy(y=15 to 0)模拟开关配置位（当 GPIOx 为模拟功能模式时有效）： 0：开关打开

				1: 开关闭合

ALISINOCHIP

20. 串行外设接口（SPI）

20.1. 概述

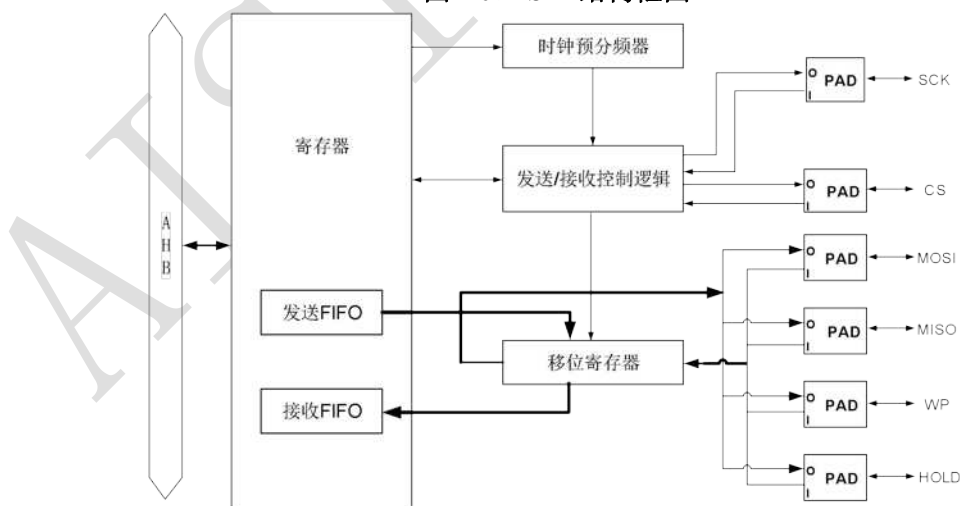
SPI 接口模块用于微控制器（MCU）与满足 SPI 外设之间进行全双工、全同步、串行通讯；SPI 接口 IP 可以工作在查询或中断方式下。

20.2. 主要特性

- 可选择主/从模式；
- 可通过两级分频因子来配置宽范围波特率；
- 支持 Mode0/1/2 /3 四种传输协议；
- 支持 SPI 一线、二线、四线传输；
- SPI 从模式支持软件 CS 功能
- SPI3 支持 XIP（直接从外部存储器取指执行程序）模式；

20.3. 结构框图

图 20-1 SPI 结构框图



20.4. 功能描述

20.4.1. 时序图

SPI 模块支持一线模式和多线传输模式（二线和四线模式）下的 IO 定义为：IO0 (MOSI)，IO1 (MISO)，IO2 (WP)，IO3 (HOLD)。

时序图如下：

- 一线模式时序

图 20-2 SPI 一线模式时序图（MSB）

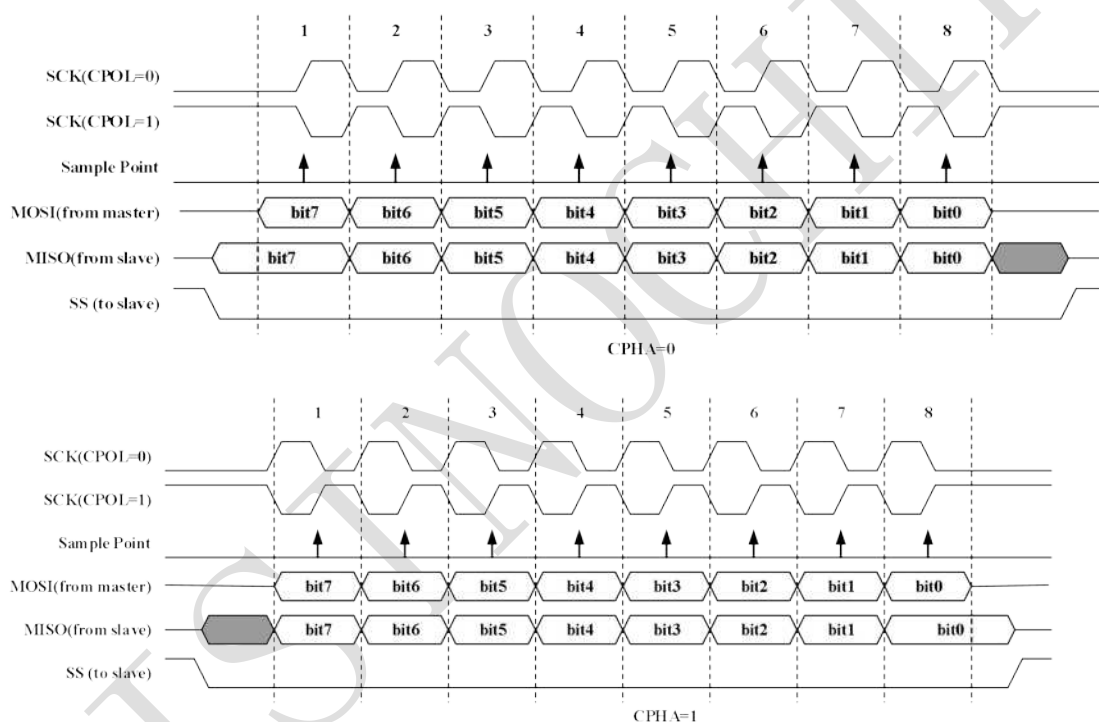
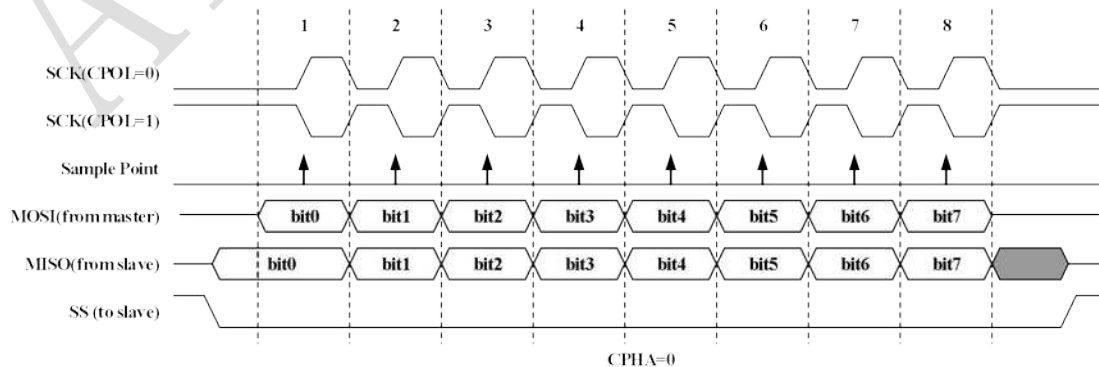
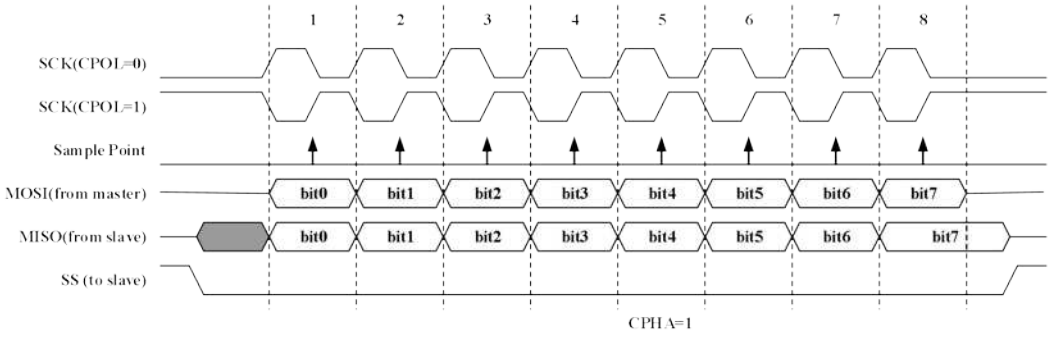


图 20-3 SPI 一线模式时序图（LSB）





● 二线模式时序

图 20-4 SPI 二线模式时序图 (MSB)

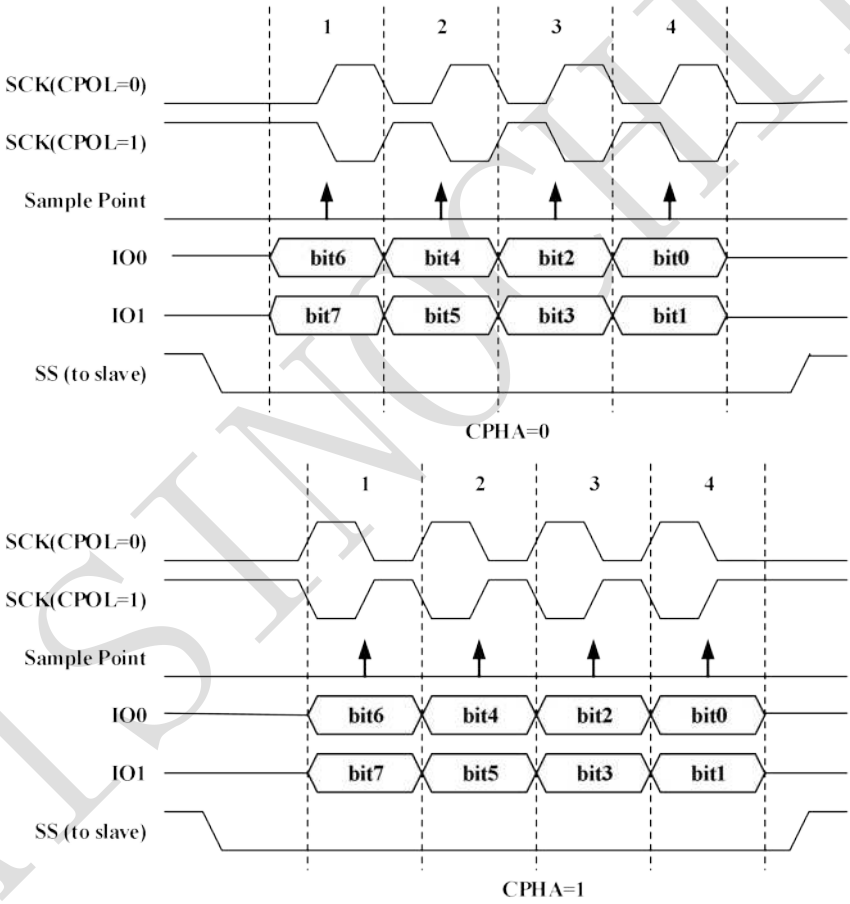
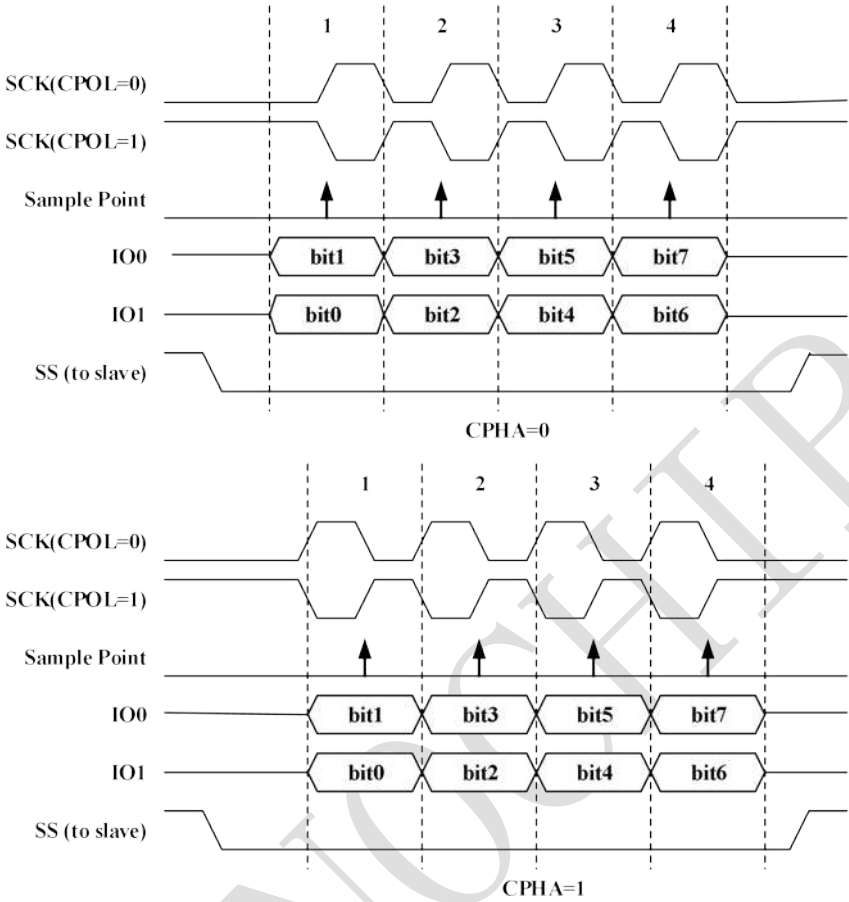
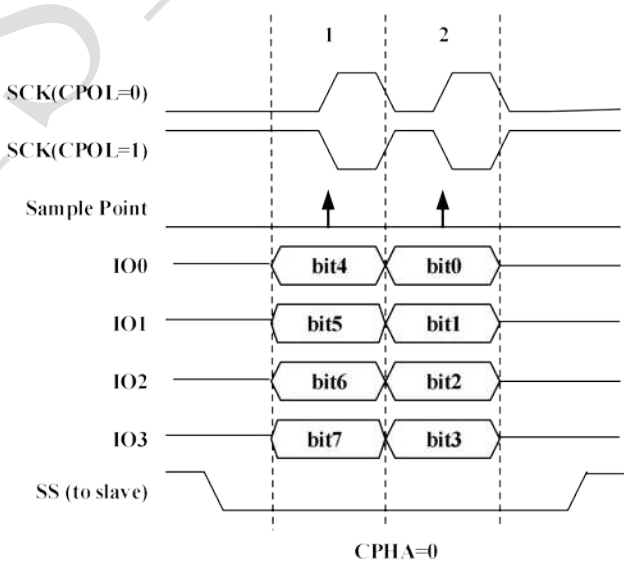


图 20-5 SPI 二线模式时序图 (LSB)



● 四线模式时序

图 20-6 SPI 四线模式时序图 (MSB)



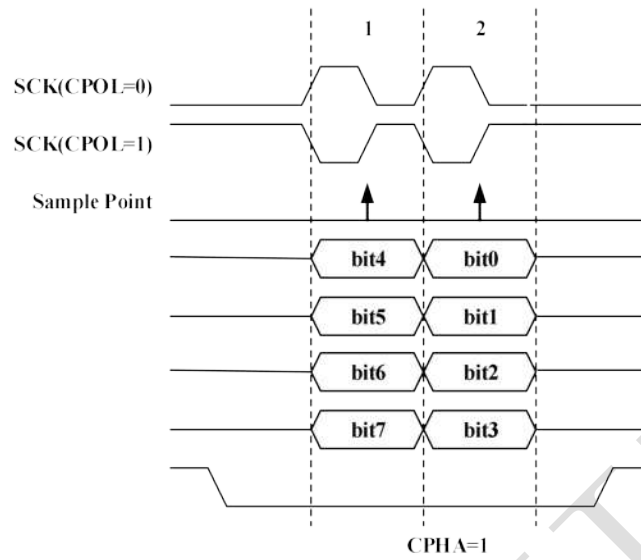
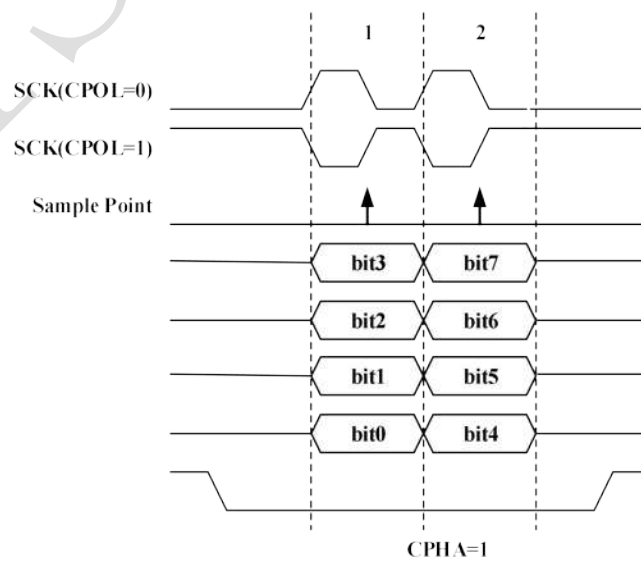
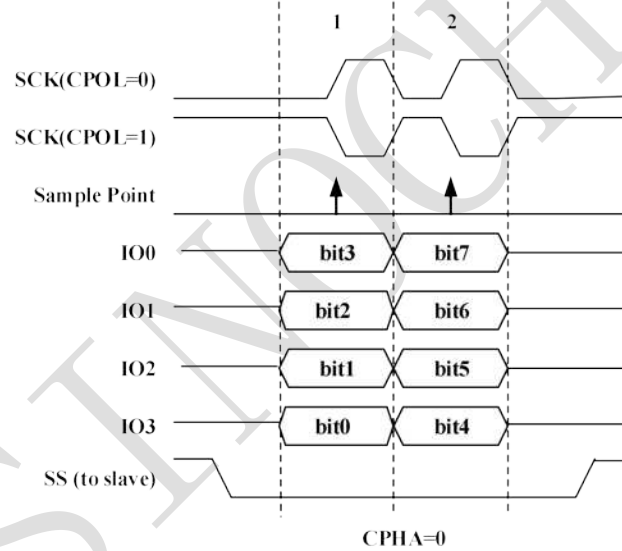


图 20-7 SPI 四线模式时序图 (LSB)



20.4.2. 时钟波特率设置

SPI 的时钟波特率设置，通过设置 SPI_BAUD 寄存器来完成。

根据 HCLK 频率和需要设定的 SPI 时钟频率，计算并设置 SPI_BAUD 的 DIV2 和 DIV1。

注：分频因子 DIV1 必须是 2 到 254 之间的偶数（包括 2 和 254）。

SPI 串行时钟计算公式： $SPI\_CLK = F_{HCLK} / (DIV1 * (DIV2+1))$ 。

例如：在 F_{HCLK} 为 64MHz 时，如需设置 SPI 的时钟频率为 8MHz，可设置 DIV1 为 8，DIV2 为 0。

20.4.3. DMA 请求

每个 SPI 接口的 TX 和 RX 都支持 DMA 功能，都有其对应的 DMA 请求号。

设置 SPI_TX_CTL.TX_DMA_REQ_EN 位使能 SPI 的 DMA 发送，设置 SPI_RX_CTL.RX_DMA_REQ_EN 位使能 SPI 的 DMA 接收。

同时还可以用 SPI 的 FIFO 功能来发起 DMA 请求。

设置 SPI_TX_CTL.TX_DMA_Level 位域，值为 TX DMA 请求 level。当 TX FIFO 中的数据小于等于此值时，TX DMA 请求有效。设置 SPI_RX_CTL.RX_DMA_Level 位域，值为 RX DMA 请求 level。当 RX FIFO 中的数据大于等于此值时，RX DMA 请求有效。

20.4.4. 存储器读取模式

其中一个 SPI 控制器接口模块（SPI3）支持对满足 SPI 协议的存储器的快速读取功能，当该功能开启时 MCU 可以如读取普通存储器一样通过内存地址直接访问 SPI 协议的 NOR Flash。

基本原理是在设定 SPI 的基本参数如时钟波特率、工作模式、线宽模式之后，在 SPI_CMD.Rd_Cmd 位中写入需要发送的读命令，SPI_PARA.Para1 位设置需要发送的参数 1(如果需要)；再通过 SPI_MEMO_ACC 寄存器 Addr_width / PARA_NO2/ PARA_NO1/ Para_Ord2/ Para_Ord1 设置地址宽度、参数及发送次序；最后使能 SPI_MEMO_ACC.SPI_Acc_EN 位，开启 SPI_Acc_EN 位后，硬件自动控制 SPI 接口的输入与输出，即完成读取 SPI 存储设备的准备设置。

AHB 总线可以直接对存储器地址进行读操作，硬件会自动开启对 SPI 存储器的通信，包括发送命令和地址和接收存储器发回的数据，在通信完成后 SPI 接口将把读取的数值通过 AHB 总线返回 MCU。

例如：在开启存储器读取模式后，SPI flash 中的数据将被映射到 0x9000_0000 开始的地址中，可以从该地址直接读取数据。

20.5. 寄存器描述

SPI1 寄存器基地址：0x4003_0000

SPI2 寄存器基地址：0x4003_0400

SPI3 寄存器基地址：0x4003_0800

偏置	名称	描述
0x00	SPI_TX_DAT	发送数据寄存器
0x00	SPI_RX_DAT	接收数据寄存器
0x04	SPI_BAUD	波特率设置寄存器
0x08	SPI_CTL	控制寄存器
0x0C	SPI_TX_CTL	发送控制寄存器
0x10	SPI_RX_CTL	接收控制寄存器
0x14	SPI_IE	中断控制寄存器
0x18	SPI_STATUS	状态寄存器
0x1C	SPI_TX_DELAY	发送等待寄存器
0x20	SPI_BATCH	批量数据个数寄存器
0x24	SPI_CS	从设备选择寄存器
0x28	SPI_OUT_EN	管脚输出使能
0x2C	SPI_MEMO_ACC	取值控制寄存器
0x30	SPI_CMD	取值命令寄存器
0x34	SPI_PARA	取值参数寄存器

20.5.1. SPI 发送数据寄存器 SPI_TX_DAT(偏移 00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	TX_DAT	WO	0x0	发送数据寄存器 该寄存器只写，读操作返回 SPI_RX_DAT 的值。

20.5.2. SPI 接收数据寄存器 SPI_RX_DAT(偏移 00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	RX_DAT	RO	0x0	接收数据寄存器 该寄存器只读，写操作改变 SPI_TX_DAT 的值。

20.5.3. SPI 波特率设置寄存器 SPI_BAUD(偏移：04h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	DIV2	RW	0x0	SPI 串行时钟二级分频因子。
7:0	DIV1	RW	0x2	SPI 串行时钟一级分频因子。该分频因子必须是 2 到 254 之间的偶数(包括 2 和 254)。Bit[0]返回值总是为 0。

SPI 串行时钟: $SPI\_CLK = SYS\_CLK / (DIV1 * (DIV2+1))$

20.5.4. SPI 控制寄存器 SPI_CTL(偏移：08h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留

21	SWCS_EN	RW	0x0	从机 CS 软件模式选择 1: CS 软件模式, CS 电平取决于 SWCS 位 0: CS 硬件模式, CS 电平取决于 CS 引脚
20	SWCS	RW	0x0	CS 软件模式下 CS 引脚选择 该位只在 SWCS_EN 位为 ‘1’ 时有意义。 它决定了 CS 上的电平, 在 CS 引脚上的 I/O 操作无效
19	CRYPT_EN	RW	0x0	加解密使能位 1: 使能 0: 不使能
18: 11	CS_TIME	RW	0x05	存储映射模式下 CS 高电平持续周期 (CS 高电平持续周期为寄存器值加 1, 为 0 时 CS 持续一个系统时钟周期)
10	CS_FILTER	RW	0x0	从机 CS 滤毛刺选择位 1: 开启从机 CS 滤毛刺功能; 0: 不开启从机 CS 滤毛刺功能;
9	CS_RST	RW	0x0	从机 CS 复位选择位 1: CS 无效时不复位从机内部比特计数; 0: CS 无效时复位从机内部比特计数; 注: 此寄存器不向客户开放
8	SLAVE_EN	RW	0x0	从机收发逻辑使能位 1: 使能 SPI 从机功能; 0: 不使能 SPI 从机功能 ; 注: 默认 SPI 从模式时不使能 SPI 从机逻辑 不影响 FIFO 读写, SPI 从模式初始化时, 该位需要置 0, 待从机 SPI 初始化完成后使能该位。
7	IO_MODE	RW	0x0	IO 方向模式选择位 1: 硬件自动切换;

				0: 软件切换;
6: 5	X_Mode	RW	0x0	多线模式控制位 00: 1X 模式; 01: 2X 模式; 10: 4X 模式; 11: 保留;
4	LSB_first	RW	0x0	MSB/LSB 在前选择位 1: SPI 总线传输中 LSB 在前; 0: SPI 总线传输中 MSB 在前;
3	CPOL	RW	0x0	时钟极性控制位 1: SCLK 低电平有效。空闲状态下为高; 0: SCLK 高电平有效。空闲状态下为低;
2	CPHA	RW	0x0	时钟相位控制位 1: 在时钟 SCLK 的偶边沿采样数据。 0: 在时钟 SCLK 的奇边沿采样数据。
1	SFILTER	RW	0x0	从机时钟滤毛刺选择位 1: 开启从机时钟滤毛刺功能; 0: 不开启从机时钟滤毛刺功能; 注: 开启从机时钟滤毛刺功能后, 48M 主频下从机 SPI 最高频率为 4M
0	Mst_mode	RW	0x0	主从模式选择位 1: SPI 工作在主模式下; 0: SPI 工作在从模式下;

20.5.5. SPI 发送控制寄存器 SPI_TX_CTL (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
15:8	Dummy	RW	0x0	无效字节寄存器

7: 4	TX_DMA_Level	RW	0x	TX DMA 请求 level。当 TX FIFO 中的数据小于等于此值时，TX DMA 请求有效。
3	TX_DMA_REQ_EN	RW	0x0	TX DMA 请求使能。 其值为 1，且 FIFO 中的数据小于等于 TX_DMA_Level 时，发出 DMA 请求。
2	TX_MODE	RW	0x0	从机发送数据选择位 1: 发送 FIFO 中数据，不发送 Dummy; 0: 先发送 Dummy，然后发送 FIFO 中数据 置 1 不发送 Dummy 时，必须保证 SPI 作为从机发送数据时 FIFO 有数据，该位值必须在 CS 无效或 SLAVE_EN 为 0 时修改。
1	TX_FIFO_Reset	RW	0x0	TX_FIFO 复位控制位 1: 写 1 复位发送 FIFO 指针; 0: 无影响; 写 1 复位有效，直到写 0 复位才会撤销。
0	TX_EN	RW	0x0	发送使能位 1: TX 方向使能; 0: TX 方向禁止;

20.5.6. SPI 接收控制寄存器 SPI_RX_CTL (偏移: 10h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
9: 8	SSHIFT	RW	0x0	主机采样移位控制位 00:不发生移位 01:主机推迟 1 个 hclk 周期开始采集数据 10:主机推迟 2 个 hclk 周期开始采集数据 11:主机推迟 3 个 hclk 周期开始采集数据

				注： spi 波特率设置为二分频时，在 FIFO 模式下，不能使用采样移位功能。 取指模式下不支持采样移位功能。
7: 4	RX_DMA_Level	RW	0x0	RX DMA 请求 level。当 RX FIFO 中的数据大于等于此值时，RX DMA 请求有效。
3	RX_DMA_REQ_EN	RW	0x0	RX DMA 请求使能。 其值为 1，且当 FIFO 中的数据大于等于 RX_DMA_Level 时，发出 DMA 请求。
2	RSV	-	-	保留
1	RX_FIFO_Reset	RW	0x0	RX_FIFO 复位控制位 1: 写 1 复位接收 FIFO 指针； 0: 无影响； 写 1 复位有效，直到写 0 复位才会撤销
0	RX_EN	RW	0x0	接收使能位 1: RX 方向使能； 0: RX 方向禁止；

20.5.7. SPI 中断控制寄存器 SPI_IE（偏移：14h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	RX_BATCH_DONE_EN	RW	0x0	接收批量传输完成中断使能位 1: 中断使能； 0: 中断禁止；
14	TX_BATCH_DONE_EN	RW	0x0	发送批量传输完成中断使能位 1: 中断使能； 0: 中断禁止；
13	RX_FIFO_FULL_OVERFLOW_EN	RW	0x0	从机接收 FIFO 写溢出中断使能位 1: 中断使能；

				0: 中断禁止;
12	RX_FIFO_EMPTY_OVERFLOW_EN	RW	0x0	从机接收 FIFO 读溢出中断使能位 1: 中断使能; 0: 中断禁止;
11	RX_FIFO_NOT_EMPTY_EN	RW	0x0	接收 FIFO 非空中断使能位 1: 中断使能; 0: 中断禁止;
10	CS_POS_EN	RW	0x0	CS 管脚电平上升沿事件中断使能位 1: 中断使能; 0: 中断禁止;
9	RX_FIFO_HALF_FULL_EN	RW	0x0	接收 FIFO 半满中断使能位 1: 中断使能; 0: 中断禁止;
8	RX_FIFO_HALF_EMPTY_EN	RW	0x0	接收 FIFO 半空中断使能位 1: 中断使能; 0: 中断禁止;
7	TX_FIFO_HALF_FULL_EN	RW	0x0	发送 FIFO 半满中断使能位 1: 中断使能; 0: 中断禁止;
6	TX_FIFO_HALF_EMPTY_EN	RW	0x0	发送 FIFO 半空中断使能位 1: 中断使能; 0: 中断禁止;
5	RX_FIFO_FULL_EN	RW	0x0	接收 FIFO 满中断使能位 1: 中断使能; 0: 中断禁止;
4	RX_FIFO_EMPTY_EN	RW	0x0	接收 FIFO 空中断使能位 1: 中断使能; 0: 中断禁止;
3	TX_FIFO_FULL_EN	RW	0x0	发送 FIFO 满中断使能位

				1: 中断使能; 0: 中断禁止;
2	TX_FIFO_EMPTY_EN	RW	0x0	发送 FIFO 空中断使能位 1: 中断使能; 0: 中断禁止;
1	BATCH_DONE_EN	RW	0x0	批量完成中断使能位 1: 中断使能; 0: 中断禁止;
0	RSV	-	-	保留

20.5.8. SPI 状态寄存器 SPI_STATUS (偏移: 18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	RX_BATCH_DONE	RO	0x0	接收模式下批量传输完成标志位 写第 1 位 BATCH_DONE 清除该位 注: 全双工模式下与 TX_BATCH_DONE 同时产生
14	TX_BATCH_DONE	RO	0x0	发送模式下批量传输完成标志位 写第 1 位 BATCH_DONE 清除该位
13	RX_FIFO_FULL_OVERFLOW	RO	0x0	从机接收 FIFO 写入溢出标志位 1: 发生从机接收 FIFO 写入溢出; 0: 未发生从机接收 FIFO 写入溢出;
12	RX_FIFO_EMPTY_OVERFLOW	RO	0x0	从机接收 FIFO 读出溢出标志位 1: 发生从机接收 FIFO 读出溢出; 0: 未发生从机接收 FIFO 读出溢出;
11	RX_FIFO_NOT_EMPTY	RO	0x0	接收 FIFO 非空标志位
10	CS_POS_Flg	RO	0x0	CS 管脚电平上升沿事件标志位

				1:发生管脚电平上升沿事件; 0: 未发生事件; 写 1 清除该标志位。
9	RX_FIFO_HALF_FULL	RO	0x0	接收 FIFO 半满标志位 1: 接收 FIFO 中的字节数大于等于 8; 0: 接收 FIFO 中的字节数小于 8;
8	RX_FIFO_HALF_EMPTY	RO	0x1	接收 FIFO 半空标志位 1: 接收 FIFO 的剩余空间大于等于 8; 0: 接收 FIFO 中的字节数小于等于 8;
7	TX_FIFO_HALF_FULL	RO	0x0	发送 FIFO 半满标志位 1: 发送 FIFO 中的字节数大于等于 8; 0: 发送 FIFO 中的字节数小于 8;
6	TX_FIFO_HALF_EMPTY	RO	0x1	发送 FIFO 半空标志位 1: 发送 FIFO 的剩余空间大于等于 8; 0: 发送 FIFO 中的字节数小于等于 8;
5	RX_FIFO_FULL	RO	0x0	接收 FIFO 满标志位 1: 接收 FIFO 中满; 0: 接收 FIFO 未滿; 注: 从机接收数据时, 为防止数据接收溢出, 可通过查询接收 FIFO 非空和接收 FIFO 半满来读数据。
4	RX_FIFO_EMPTY	RO	0x1	接收 FIFO 空标志位 1: 接收 FIFO 空; 0: 接收 FIFO 非空;
3	TX_FIFO_FULL	RO	0x0	发送 FIFO 满标志位 1: 发送 FIFO 中满; 0: 发送 FIFO 未滿;
2	TX_FIFO_EMPTY	RO	0x1	发送 FIFO 空标志位 1: 发送 FIFO 空;

				0: 发送 FIFO 非空; 注:从机发送数据时, 为从机连续发送有效数据, 可通过查询发送 FIFO 半空或发送 FIFO 满标志来往 FIFO 中写数据。
1	BATCH_DONE	RW	0x0	批量传输完成标志位。写 1 则清除该标志位 1: 传输完成; 0: 传输未完成; 该状态发送模式和接收模式均会产生
0	TX_BUSY	RO	0	SPI 忙于发送标志位 1: 作从机时, SPI 正在发送数据或发送状态下发送 FIFO 不为空。作主机时, SPI 主机正在发送数据; 0: SPI 空闲; 硬件清 0 和置 1 从机模式下, FIFO 非空, SPI 正在发送数据, TX_BUSY 将置 1

注明: SPI 模块作为主机时, 批量传输完成后, BATCH_DONE 置位, SPI 模块不会再发送/接收数据。

SPI 模块作为从机发送模式时, 批量传送完成后, BATCH_DONE 置位, 如主机继续读取数据, 从机重新开始一次新的计数, SPI 模块会继续发送数据, 优先发送 FIFO 中的数据, FIFO 中数据发送为空后, 发送 dummy byte。

SPI 模块作为从机接收模式时, 批量传送完成后, BATCH_DONE 置位, 如主机继续发送数据, 从机重新开始一次新的计数, 并将数据写入 FIFO 中。

在全双工模式下时, BATCH_DONE 表示批量发送和接收完成。从机双工模式下, 批量发送和接收完成后, 如果主机继续传输, 从机从新开始一次新的计数。从机双工模式下, 批量传输完成后主机停止发送和接收。

TX_BUSY 标志由硬件置 1 和清 0 (对此操作写没有任何作用), TX_BUSY 用于 SPI 通信状态。TX_BUSY 置 1 时, 表示 SPI 正在发送数据。如果软件要关闭 SPI 或 SPI 其他操作, 可以使用 TX_BUSY 标志检测传输是否结束以避免破坏最后一个字节传输。

20.5.9. SPI 发送等待寄存器 SPI_TXDelay(偏移: 1Ch)

比特	名称	属性	复位值	描述
31:0	SPI_TDY	RW	0x0	SPI 每发送一个字节需要等待的 SPI 引擎。 该控制位只在主模式下有效。

在发送等待的过程中, CLK 将按照 CPOL 设定值停止。

20.5.10. SPI 批量传输数据个数寄存器 SPI_BATCH (偏移: 20h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19:0	Batch_Number	RW	0x0	该寄存器用来存储 SPI 总线上即将传输的数据字节个数。

20.5.11. SPI 从设备选择寄存器 SPI_CS(偏移: 24h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1:0	SPI_CS	WO	0x00	置 1 将使设备选择信号 SPI_CS[1:0]变低。主模式下此位应该在配置的最后一步写入, 写此位后数据传输立即开始。如果主模式在传输的不同阶段需要改变其它寄存器的配置, 即么需要重新写此位 (不管此位变或不变) 来触发下一次传输。 从模式下, 写此位无效。 读此位 0, 反映 CS[0]管脚的状态。

注: 从机 SPI_CLK 极性必须在 CS 有效之前的一个 SPI_CLK 周期初始化完成。

20.5.12. SPI 管脚输出方向 SPI_OUT_EN(偏移: 28h)

比特	名称	属性	复位值	描述

31:4	RSV	-	-	保留
3	SPI_HOLD_EN	RW	0x0	管脚输出使能位 1: 输出模式; 0: 输入模式; 在 IO 自动切换的时候, 该位无效
2	SPI_WP_EN	RW	0x0	管脚输出使能位 1: 输出模式; 0: 输入模式; 在 IO 自动切换的时候, 该位无效
1	SPI_MISO_EN	RW	0x0	管脚输出使能位 1: 输出模式; 0: 输入模式; 在 IO 自动切换的时候, 该位无效
0	SPI_MOSI_EN	RW	0x0	管脚输出使能位 1: 输出模式; 0: 输入模式; 在 IO 自动切换的时候, 该位无效

20.5.13. SPIA 取值控制寄存器 SPI_MEMO_ACC(偏移: 2Ch)

比特	名称	属性	复位值	描述
31:19	RSV	-	-	保留位。
18: 14	Addr_width	RW	0x10	发送地址的位数等于该寄存器值。 地址从 AHB 地址总线 LSB 向上取值。 最大值为 24, 超过将会按 24 进行后续计算 0x08 : 8 bit 0x10 : 16bit 0x18 : 24bit 其他非 0: 24bit
13: 9	PARA_NO2	RW	0x0	发送参数 1 的位数等于该寄存器值。

				参数从 PARA2 的 LSB 向上取值。 0 表示不使用参数 2，最大值为 16，超过将会按 16 进行后续计算。 0x08 : 8 bit 0x10 : 16bit 0x00: 不使能 其他非 0: 16bit
8: 5	PARA_NO1	RW	0x0	发送参数 2 的位数等于该寄存器值。 参数从 PARA1 的 LSB 向上取值。 0 表示不使用参数 1，最大值为 8，超过将会按 8 进行后续计算。 0x08: 8 bit 0x00: 不使能 其他非 0: 8bit。
4	RVS	-	-	保留
3	Con_Rd_EN	RW	0x0	连读使能位。 1:使能连读。 0:不使能连读。
2	Para_Ord2	RW	0x0	决定在发送地址前后加入参数 2。 1:在地址后发送。 0:在地址前发送。
1	Para_Ord1	RW	0x0	决定在发送地址前后加入参数 1。 1:在地址后发送。 0:在地址前发送。
0	SPI_Acc_EN	RW	0x0	SPI 快速访问存储器功能块使能。 1:使能。 0:不使能，作为普通 spi 接口。

20.5.14. SPIA 取值命令寄存器 SPI_CMD(偏移: 30h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留位。
15:8	Wr_Cmd	RW	0x0	存放写 SPI 存储器的指令。
7:0	Rd_Cmd	RW	0x0	存放读 SPI 存储器的指令。

20.5.15. SPIA 取值参数寄存器 SPI_PARA(偏移: 34h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留位。
23:8	Para2	RW	0x0	存放可能使用的地址 Dummy 或命令参数。
7:0	Para1	RW	0x0	存放可能使用的地址 Dummy 或命令参数。

PS: 取值模式多线模式下, 命令、地址、参数, 仍然以一线模式传输。

参数 1 和参数 2 如果需要同时发送, 先发送参数 1, 后发送参数 2。

20.6. 使用流程

20.6.1. SPI 主模式发送

1、初始阶段

- 1) 配置 SPI 控制寄存器 SPI_CTL 中的 X_Mode、LSB_first、CPOL、CPHA、Mst_mode 比特位。
- 2) 配置波特率寄存器 SPI_BAUD。
- 3) 配置发送等待寄存器 SPI_TXDelay(可按需求配置)。
- 4) 配置 SPI_OUT_EN 寄存器切换管脚或配置 SPI_CTL 的 IO_MODE 位为 1 硬件自动切换管脚。

2、发送数据

- 1) 设置 SPI 发送控制寄存器 SPI_TX_CTL 中的 TX_EN 比特位。
- 2) 设置 SPI_BATCH 寄存器为要发送的数据的个数。

- 3) 通过 SPI 从设备选择寄存器 SPI_CS[bit0], 将 SPI 从设备配置成选择状态。
- 4) 当发送 FIFO 非满时, 写入待发送的数据, 直到 BATCH 个数据全部发完。
- 5) 等待 SPI 状态寄存器 BATCH_DONE 状态位置位。
- 6) 再次发送数据重复 2-5, 直至 SPI 数据全部发送完成。
- 7) 清除 SPI_TX_CTL 中的 TX_EN 位。
- 8) 清除 SPI 从设备选择寄存器 SPI_CS[bit0], 结束发送。

20.6.2. SPI 主模式接收

- 1、初始阶段
 - 1) 配置 SPI 控制寄存器。X_Mode、LSB_first、CPOL、CPHA、Mst_mode 比特位。
 - 2) 配置波特率寄存器。
 - 3) 配置发送等待寄存器。
 - 4) 配置管脚输出使能寄存器。
 - 5) 配置 SPI_OUT_EN 寄存器切换管脚或配置 SPI_CTL.IO_MODE 硬件自动切换管脚
- 2、接收数据
 - 1) 设置 SPI 接收控制寄存器 SPI_RX_CTRL 中的 RX_EN 比特位。
 - 2) 设置 SPI_BATCH 寄存器。
 - 3) 通过 SPI 从设备选择寄存器, 将 SPI 配置成选择状态。
 - 4) 当接收 FIFO 非空时, 读取接收 FIFO 中的数据, 直到 BATCH 个数据全部收完。
 - 5) 等待 SPI 状态寄存器 BATCH_DONE 状态位置位。
 - 6) 再次接收数据重复 2-5, 直至 SPI 数据全部接收完成。
 - 7) 清除 SPI_RX_CTRL 中的 RX_EN 位。
 - 8) 清除 SPI 从设备选择寄存器, 结束接收。

20.6.3. SPI 主模式接收时 Dummy 控制位

SPI 工作在主模式, 当仅处于接收模式下, MOSI 数据线的状态由 Dummy 控制位决

定。当 SPI 引擎会将 Dummy 数据按比特移出。

20.6.4. SPI 从模式发送时 Dummy 控制位

当 TX_MODE 为 0，SPI 发送 FIFO 为空时，FIFO 指针需要从 CPU 时钟域同步到 SPI 时钟域，1 线和 2 线模式需要一个字节进行同步，4 线模式则需要 2 个字节进行同步，当主设备发起传输时，主设备需要多接收 1 到 2 个 Dummy 字节。

当 TX_MODE 为 1，SPI 发送 FIFO 为空时，FIFO 指针不需要从 CPU 时钟域同步到 SPI 时钟域，当主设备发起传输时，从机中的 FIFO 有效数据可以立即发送出去，主设备不会多接收 Dummy 字节。

20.6.5. SPI 从模式接收时 Dummy 控制位

SPI 工作在从模式，当仅处于接收模式下，MISO 数据线的状态由 Dummy 控制位决定。SPI 引擎会将 Dummy 数据按比特移出。

20.6.6. SPI 从模式 NCS 功能

NCS 功能主要用于在 GPIO 资源紧张时，可以为 SPI 设备节省一根信号线（即 CS）。NCS 模式是指从机的片选信号不受主机的 CS 管脚信号影响，可以通过软件的方式让从机处于片选状态。NCS 模式的配置流程为设置 SPI_CTL 寄存器的 SWCS_EN 位为 1，再设置 SPI_CTL 的 SWCS 位为 1。则进入了 NCS 模式。使用 NCS 模式时需要注意，应在主机的 SPI 模块初始化配置完成之后，从机再开启 NCS 功能。这是因为，如果从机先进入 NCS 模式后，从机就处于了工作状态，此时主机再配置时钟极性时，可能会在 SCK 线上产生一个高低电平的翻转，而从机误认为是一个 CLOCK 有效边沿，从而造成从机的时序错乱。

20.6.7. SPI 存储器读取

SPI3 接口模块支持对满足 SPI 协议的存储器的快速读取功能，当该功能开启时 MCU 可以如读取普通存储器一般快速读取支持 SPI 协议的 sram 或者 nor_flash。其使用方法及设置步骤如下：

对于不同的 spi 存储器件，如有需要需按照器件的各自要求设置其状态寄存器。关于 spi 存储器的设置要求，请参照其各自 spec。

进行读操作前，需要进行一些基本设定，通过 SPI_BAUD 寄存器设置时钟波特率；通过 SPI_CTL 寄存器的 CPOL 位和 CPHA 位设置 SPI 工作模式；通过设置 SPI_CTL 寄存器的 X_mode 位设置线宽模式；之后在在 SPI_CMD 寄存器的 Rd_Cmd 位中写入需要发送的读命令，SPI_PARA 寄存器的 Para1 位设置需要发送的参数 1（如果需要）；再通过 SPI_MEMO_ACC 寄存器 Addr_width / PARA_NO2/ PARA_NO1/ Para_Ord2/ Para_Ord1 设置地址宽度、参数及发送次序；最后使能 SPI_ACC_EN 位，开启 SPI_ACC_EN 位后，硬件自动控制 SPI 接口的输入与输出，即完成读取 SPI 存储设备的准备设置。

AHB 总线可以直接对存储器地址进行读操作，硬件会自动开启对 SPI 存储器的通信，包括发送命令和地址和接收存储器发回的数据，在通信完成后 SPI 接口将把读取的数值通过 AHB 总线返回 MCU。

20.6.8. SPI 存储器读取（连读）

启动连读使能位即 SPI_MEMO_ACC 寄存器的 Con_Rd_EN 位之后，当条件满足时，硬件会自动开启连读功能停止发送命令以及地址以减少读操作用时。连读的条件为：

- 1) 读取地址为连续值，即没有地址跳变。
- 2) 没有进行过写操作。

除了设置寄存器外，连读模式对用户操作没有影响，当连读模式启动时，用户操作等同于读取普通读取 spi 存储器。

20.6.9. SPI SRAM 写入

同时 SPI 接口模块支持对满足 SPI 协议的 sram 的写功能，其使用方法如下：

首先需要设置 SPI_CTL 寄存器的 Mst_mode 位，将 SPI 接口置于主模式；然后通过 SPI_BAUD 设置时钟波特率，通过 SPI_CTL 的 CPOL 和 CPHA 设置 SPI 工作模式；然后设置 SPI_CTL 的 X_mode 位和 SPI_OUT_EN 寄存器设定通信 IO。以上为 SPI 主模块发送准备设置。

之后通过 SPI_CMD 寄存器的 Wr_Cmd 位设置需要发送的写命令、SPI_PARA 寄存器的 Para1 (2) 位设置需要发送的参数 1 (2) (如果需要)；再通过 SPI_MEMO_ACC 寄存器 Addr_width / PARA_NO2/ PARA_NO1/ Para_Ord2/ Para_Ord1 设置地址宽度、参数及发送次序，最后使能 SPI_ACC_EN 位，开启 SPI_ACC_EN 位后，硬件自动控制 SPI 接口的输入与输出，即完成写 SPI sram 的准备设置。

完成上述配置后，软件就可以直接通过 memory 地址访问的方式来读写 SPI sram 空间了。

AISINOCHIP

21. 通用异步收发器（UART）

21.1. 概述

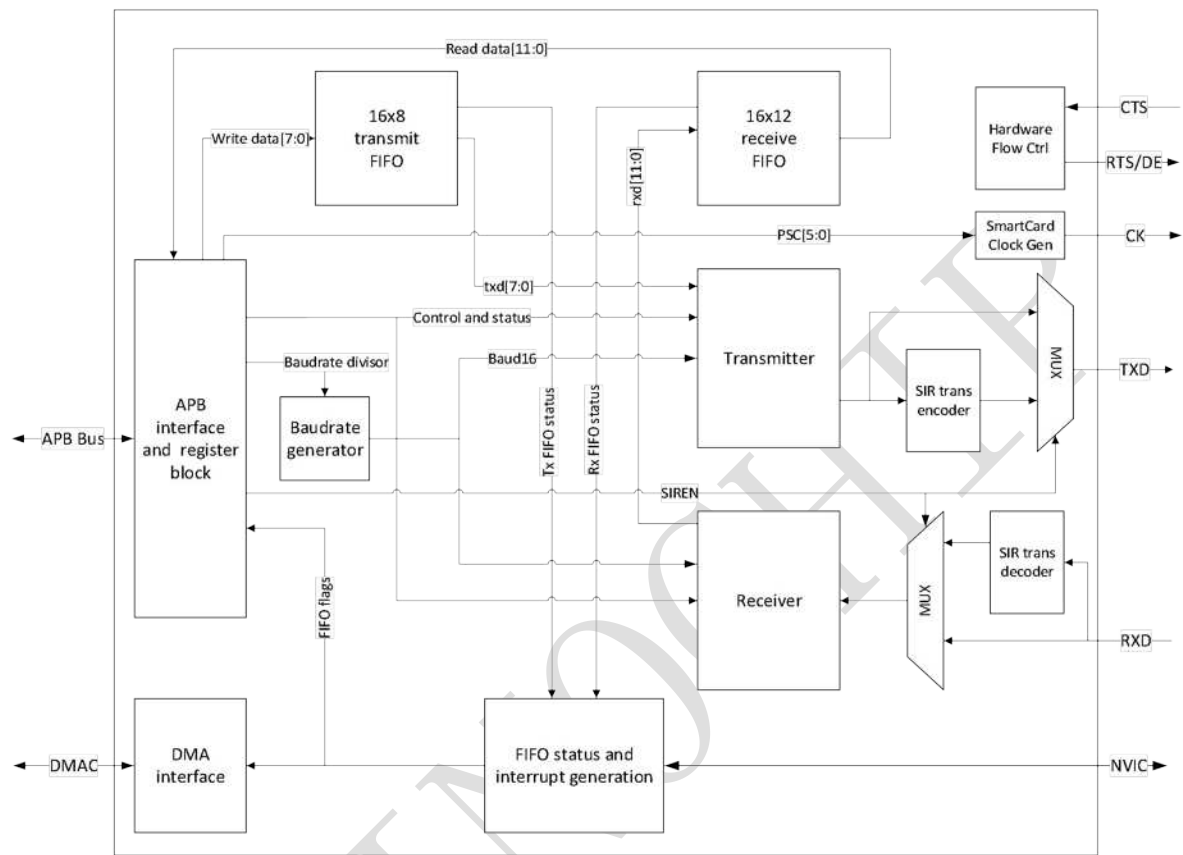
芯片上集成了四个 UART 串口模块。UART 部分除了支持常用的 UART 功能外，还可以支持 LIN 总线、IrDA SIR、智能卡（主模式）、单线半双工模式、多机通信、RS485 等功能。

21.2. 主要特性

- 支持 CTS，RTS 流控制
- 16 字节的硬件收发 FIFO
- 波特率支持整数和小数分频
- 总线空闲检测
- 支持多机通信功能
- 支持 RS485 通信
- 支持 7816 主机模式
- 支持单线模式
- 支持波特率自适应功能
- 支持波特率计数功能
- 可编程位宽（5~9bit），奇偶或 0/1 校验，停止位个数

21.3. 结构框图

图 21-1 UART 结构框图

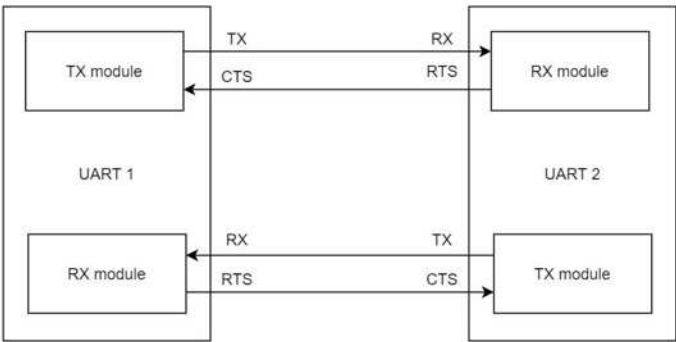


21.4. 功能描述

21.4.1. CTS 和 RTS 流控功能

硬件流控功能的通过 CTS 和 RTS 引脚来实现。通过配置控制寄存器 UART_CR1 的 CTSEn 和 RTSEn 位来使能硬件流控功能。

图 21-2 两个 UART 之间的硬件流控连接示意图



- 控制流功能不影响 UART 的正常使用，没用使用需求时请禁止使用该功能。
- CTS 为 UART 输入端口，低电平有效，表示 UART 可以发送数据。如果 CTS 输入状态为 1，用在当前传输结束时阻止数据发送。写 UART_DR 寄存器时，数据只会保存在发送 FIFO 中不会被发出，为 0 时开始发送。
- RTS 为 UART 输出端口，低电平有效，表示 UART 已经准备好可以接收数据，当接收 FIFO 中数据个数大于 RXIFLSEL 寄存器所设的中断触发点个数时，RTS 输出状态会被置位，表示不能再接收更多数据。

21.4.2. DMA 请求

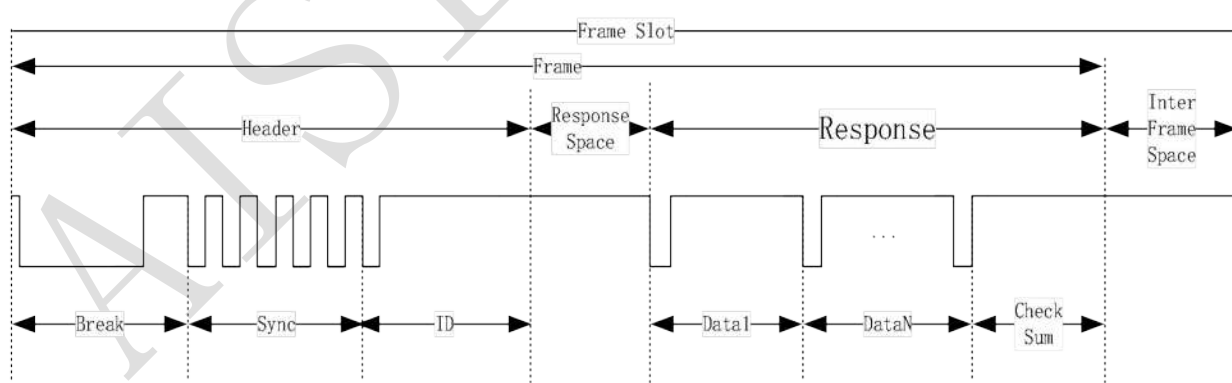
每个 UART 接口的 TX 和 RX 都支持 DMA 功能，都有其对应的 DMA 请求号（见错误!未找到引用源。）。

设置 UART_CR1.TXDMAE 位使能 DMA 发送，设置 RXDMAE 位使能 DMA 接收。

另外可以通过设置 UART_CR1.DMAONERR 来选择在 UART 发生接收错误（PE、FE、BE、OE）时是否产生 DMA 请求。

21.4.3. LIN 总线功能

图 21-3 LIN 帧格式

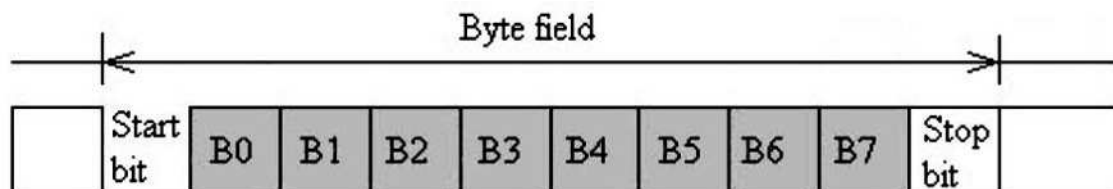


上图是 LIN 总线的基本数据帧格式图，其中包含间隔场、同步场、标识符场、数据场和校验场。其中除了间隔场外，其他场的数据格式都和普通带 1 个起始位和 1 个停止位的 UART 数据格式一样。间隔场包含一个连续不少于 13 个 bit 的低电平信号。见下图所示。

图 21-4 LIN 间隔场格式



图 21-5 LIN 字符格式



设置 UART_BCNT.BCNT_VALUE 位域可以改变 LIN 总线模式间隔场的 Break 信号长度，使能其中的 BCNT_START 位开始计时，然后使能 UART_CR3.BRK 位发送间隔场。

在发送同步场时，可以发送一个 0x55，发送过程与普通 UART 发送过程相同。

在发送标识符场、数据场和校验场时，LIN 发送过程与普通 UART 发送过程相同。

从模式接收时，需使用间隔场 Break 信号的检测功能，可以通过直接轮询访问 UART_ISR.LBDI 位,或通过使能 UART_IE.LBDI 中断，在中断服务函数中检测 Break 信号来实现。

同时 LIN 总线还支持在从机进入 STOP 模式时，通过 EXTI 模块唤醒，实现低功耗应用。

21.4.4. IrDA SIR 功能

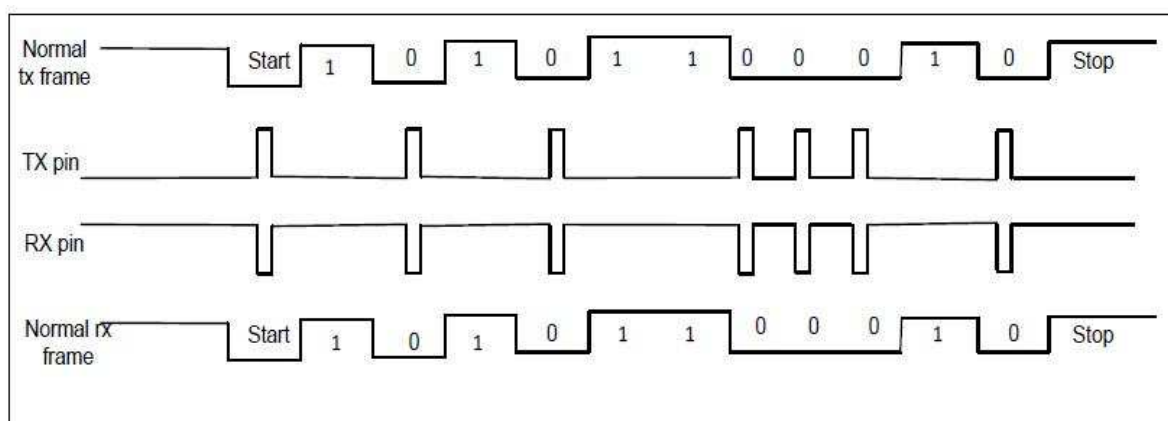
使能 UART_CR1.SIREN 位，即可打开 IrDA SIR 红外功能。

如若使用 IrDA SIR Low Power 模式，需要使能 UART_CR1.SIRLP 位。

在 IrDA 模式下，UART 数据帧由 SIR 发送编码器进行调制，调制后的信号经由红外 LED 进行发送，经解调后将数据发送至 UART 接收器。对于编码器而言，波特率应小于 115200。在 IrDA 模式下，TX 引脚电平与 RX 引脚不同。TX 引脚通常为低电平，RX 引脚通常为高电平。IrDA 引脚电平保持稳定代表逻辑‘1’，红外光源脉冲(RTZ 信号)代表逻辑‘0’。其脉冲宽度通常占一个位时间的 3/16。

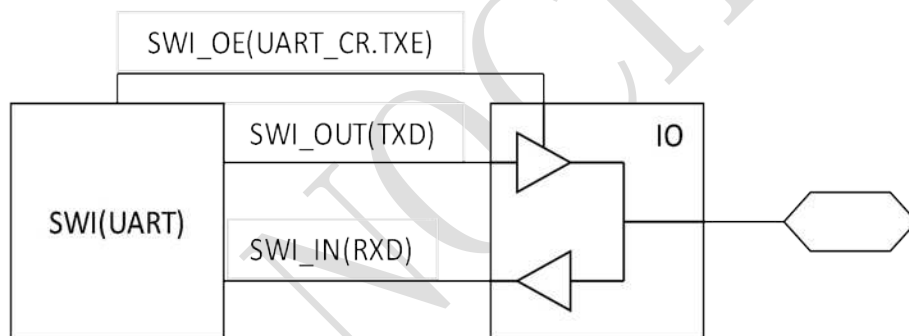
启用 Low Power 模式时，高电平脉冲宽度为为最高波特率 115200 时的一个位宽的 3/16。

图 21-6 IrDA SIR 数据调制解调



21.4.5. 单线模式

图 21-7 UART 单线半双工模式框图



UART 可以工作在单线半双工模式，通过设置使能 `UART_CR2.HDSEL` 位开启。开启单线半双工模式时，发送和接收都是通过 TX 引脚完成。

发送时，设置 `UART_CR1.TXE` 为 1，开启 TX 引脚发送，此时需禁用 RXE。

接收时，设置 `UART_CR1.TXE` 为 0，开启 TX 引脚接收，此时需启用 RXE。

21.4.6. 智能卡主模式

UART 支持智能卡模式，置位 `UART_CR2.SCEN`，打开 UART 的智能卡模式。在智能卡模式中，UART 固定采用 1.5 位停止位，此时 `UART_CR3.STP2` 配置无效。

- 智能卡模式中，需要如下配置。
 - 需要置位 `UART_CR2.HDSEL`，因为接收发送都是通过 TX 管脚完成。
 - 需要置位 `UART_CR3.PEN`，使能校验位。
 - 需要置位 `UART_CR2.CLKEN`。可以从 CK 管脚上输出时钟，由 UART 时钟

根据 UART_GTPR.PSC 分频得到，这个功能比较独立，在非智能卡模式也可以使用。

- 如果支持重发机制，则最好关闭发送 FIFO，在发送时检测到 FE 错误时，需要软件配合重新发送上一个字节。
- 如果发送时需要额外保护时间，需要配置 UART_GTPR.GT。
- 如果需要实现接收超时功能（BWT 和 CWT），可以配置 UART_BCNT.AUTO_START_EN 位，并配置合适大小的 UART_BCNT.BCNT_VALUE。

- 发送模式

- 配置 UART_GTPR.GT 以增加额外保护时间。发送一个字符需要 $12 + \text{UART\_GTPR.GT}$ 个 ETU 时间，然后才会将发送完成标志 TXI 置位。
- 在开始位、数据位和校验位期间，发送的输出使能被置位，可以驱动总线。在停止位期间，UART 释放总线，并在校验位结束后延迟一个 ETU 时间，采样总线状态，如果检测到低电平，将帧错误标志 FE 在 11 个 ETU 处置位。注意 FE 标志无需等待额外保护时间，先于发送完成标志 TXI 至少一个 ETU 时间。

- 接收模式

- 接收时，如果检测到校验错误，会根据 UART_CR2.NACK 位来决定是否在停止位期间拉低总线，以告诉发送器发生了校验错误。
- 如果检测到校验错误，会在 9.5 个 ETU 处置位 PE。如果置位了 UART_CR2.NACK 位，则会在从第 10.5 个 ETU 处拉低总线，并维持 1 个 ETU 时间。

21.4.7. 多机通信

多机通信网络支持一主多从模式，某个 UART 设备可以为主，它的 TX 输出和其他 UART 从设备的 RX 输入相连接；UART 从设备各自的 TX 输出逻辑与在一起，并且和主设备的 RX 输入相连接。

- 未被寻址的设备可启动静默模式，在静默模式中：
 - 任何接收状态位都不会被置位，包括 PE、FE、BE、LBDI。

- 所有接收中断被禁止。
- UART_CR2.RWU 位被置 1。
- 根据 UART_CR2.WAKE 位状态, UART 可以用两种方法进入或退出静默模式。
 - 如果 WAKE = 0: 进行空闲总线检测。
 - 当 UART_CR2.RWU 被软件写 1 时, UART 进入静默模式。当检测到空闲帧时, 它被唤醒。然后 UART_CR2.RWU 被硬件清零, 但是 UART_ISR.IDLEI 不置位。
 - RWU 可以在任何时候被软件写 0, 并退出静默模式。
 - 空闲帧 (IDLEI) 是从接收字符的停止位开始计数, 等待一个完整字符时间, 如果没有新的数据在总线上传输, 则置位。
 - 如果 WAKE = 1: 进行地址标记检测。
 - 如果接收到的字节与它的编程地址不匹配时, 硬件置位 UART_CR2.RWU, UART 进入静默模式。接收的地址字节不会写入到接收 FIFO 中, 也不会产生中断, 因为此时已经进入了静默模式。
 - 当接收到的字节与接收器内的编程地址匹配时, 硬件清零 UART_CR2.RWU, UART 退出静默模式。接收的地址字节被正常接收, 因为 RWU 已被清零。
 - 在静默模式时, 当接收缓冲器不包含数据时, UART_CR2.RWU 位可以被软件写 0 或 1。否则该次软件写操作被忽略。
- 地址标记检测
 - 在这个模式下, 如果 MSB 是 1, 该字节被认为是地址, 否则被认为是数据。在一个地址字节中, 目标地址被放在低位。
 - 如果字宽 WLEN 等于 8 比特或者 9 比特, 则支持 7 位地址模式。其他字宽 (5、6、7 比特) 不支持 7 位地址模式。所有字宽都支持 4 位地址模式。UART_CR2.ADDM7 决定 7 位地址模式还是 4 位地址模式。
 - 接收到的地址同 UART_CR2.ADDR 做比较。

21.4.8. 波特率自适应

- UART 可根据接收一个字符检测并自动设置波特率寄存器。自动波特率检测在以下两种情况下非常有用:

- 事先不知道系统的通信速度。
- 系统正在使用精度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。
- 波特率自适应功能需发送 0x7F 字符, UART 会测量下降沿到下降沿的时间 (8 个比特位时间)。
- 可以被检测的波特率范围为 UART 模块工作时钟频率的 16~65536 分频。
- 置位 UART_CR2.ABREN 位使能波特率自适应功能。之后 UART 将等待 RX 线路上的传递的字符。通过 UART_ISR.ABRI 位来指示自动波特率操作完成。
- 在波特率自适应完成后, 需要软件将 UART_CR2.ABREN 位清零, 否则会一直处于波特率自适应模式。
- 在波特率自适应模式中, UART 接收到的数据将会被丢弃。

21.4.9. RS485 的控制器支持

RS485 属于半双工总线, 所以 RS485 收发器需要进行发送和接收的方向转换。一种方式是可以使用 MCU 的某个通用 GPIO 口连接到 RS485 收发器的方向控制脚 (以下简称 DE Pin), 由软件来控制 RS485 收发器的收发方向。如果 MCU 自带 RS485 功能, 则可以使用 MCU 的 RS485 专用 Pin 来自动控制收发器的收发方向。ACM32G103 自带 RS485 收发器的方向控制 DE Pin, 即 UARTx_RTS_DE Pin (详见 ACM32G103 数据手册)。

【注意, UART RTS 和 DE 共用一个管脚, DE 功能优先。】

根据用户的硬件电路, DE 信号可能需要高有效, 或者低有效, 这可以通过 UART_BCNT.DEP 来选择 DE 信号的极性。

例如, RS485 收发器的 DE 脚高电平代表发送方向, 低电平代表接收方向。当选择 DE Pin 高有效时, 在 UART 发送第一个字符的起始位之前的一段时间 (通过 UART_BCNT.BCNT_VALUE.DEAT 配置), 会先将 DE Pin 拉高, 在发送最后一个字符的停止位之后的一段时间 (通过 UART_BCNT.BCNT_VALUE.DEDT 配置), 再将 DE Pin 拉低, 让 RS485 收发器处于接收状态。

21.4.10. 总线空闲检测

- 总线空闲检测有两种方式实现, 一种是通过空闲帧 IDLEI 标志实现, 一种是通

过比特计时功能实现。

- 空闲帧 IDLE

- 接收数据时，检测到上一个字符的停止位时，内部定时器开始工作，如果定时器的时长大于一个字符的时长，就会置位 IDLEI，并可以进入中断。
- 字符长度包括起始位、数据位、校验位和停止位长度。
- 这种方式时长固定，且检测时长较短。

- 比特计时功能

- 使能 UART_BCNT.AUTO_START_EN 后，接收数据时，检测到上一个字符的停止位时，内部定时器开始工作，如果定时器的时长大于 UART_BCNT.BCNT_VALUE，就会置位 BCNTI，并可以进入中断。
- 这种方式时长可配置。

21.5. 寄存器描述

UART1 寄存器基地址：0x4001_3800

UART2 寄存器基地址：0x4000_4400

UART3 寄存器基地址：0x4000_4800

UART4 寄存器基地址：0x4000_4C00

偏置	名称	描述
0x00	UART_DR	数据寄存器
0x04	UART_FR	标志寄存器
0x08	UART_BRR	波特率寄存器
0x0C	UART_IE	中断使能寄存器
0x10	UART_ISR	中断和状态寄存器
0x14	UART_CR1	控制寄存器 1
0x18	UART_CR2	控制寄存器 2
0x1C	UART_CR3	控制寄存器 3
0x20	UART_GTPR	保护时间和预分频寄存器
0x24	UART_BCNT	比特计时寄存器

21.5.1. 数据寄存器 UART_DR(偏移: 00h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	OE	RO	0	Overrun 错误
11	BE	RO	0	Break 错误
10	PE	RO	0	奇偶校验错误
9	FE	RO	0	帧格式错误
8:0	DATA	R/W	0	发送或接收的数据 注: 1. 使能 FIFO 功能时, 若 FIFO 已满, 有新的数据进入, FIFO 中原有数据不会被覆盖, 新的数据会被直接丢失。 2. 禁止 FIFO 功能时, 若 UART_DR 寄存器中有数据还未发送, 写 UART_DR 寄存器, 新的发送数据会被直接丢失, 不会覆盖 UART_DR 寄存器中原有未发送数据。

21.5.2. 标志位寄存器 UART_FR(偏移: 04h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	BUSY	RO	0	发送忙标志 0: 发送 FIFO 为空并且所有位都从移位寄存器中移出 1: 发送 FIFO 有数据
8	CTS	RO	0	CTS 输入管脚状态 0: CTS 输入高电平 1: CTS 输入低电平
7	TXFE	RO	1	发送 FIFO/UART_DR 寄存器空状态位:

				0: 如果使能 FIFO 表示发送 FIFO 非空; 如果禁止 FIFO 表示 UART_DR 寄存器有数据。 1: 如果使能 FIFO 表示发送 FIFO 为空; 如果禁止 FIFO 表示 UART_DR 寄存器无数据。
6	RXFF	RO	0	接收 FIFO/UART_DR 寄存器满状态位: 0: 如果使能 FIFO 表示接收 FIFO 非满; 如果禁止 FIFO 表示 UART_DR 寄存器非满。 1: 如果使能 FIFO 表示接收 FIFO 为满; 如果禁止 FIFO 表示 UART_DR 寄存器为满。
5	TXFF	RO	0	发送 FIFO/UART_DR 寄存器满状态位: 0: 如果使能 FIFO 表示发送 FIFO 非满; 如果禁止 FIFO 表示 UART_DR 寄存器非满。 1: 如果使能 FIFO 表示发送 FIFO 为满; 如果禁止 FIFO 表示 UART_DR 寄存器为满。
4	RXFE	RO	1	接收 FIFO/UART_DR 寄存器空状态位: 0: 如果使能 FIFO 表示接收 FIFO 非空; 如果禁止 FIFO 表示 UART_DR 寄存器有数据。 1: 如果使能 FIFO 表示接收 FIFO 为空; 如果禁止 FIFO 表示 UART_DR 寄存器无数据。
3	OE	RO	0	Overrun 错误 0: 无错误 1: 有错误 注: OE 只在接收时有效 1. 开启 FIFO 功能时, 当接收 FIFO 已满后再接收到数据时被置位。 2. 关闭 FIFO 功能时, 当 UART_DR 寄存器中有数据后再接收到数据时被置位 写 1 清 0
2	BE	RO	0	Break 错误

				0: 无错误 1: 有错误 当接收数据持续为低超过传输 1 个 word 的时间时被置位
1	PE	RO	0	奇偶或 0/1 校验位错误 0: 无错误 1: 有错误
0	FE	RO	0	帧格式错误 0: 无错误 1: 有错误 当停止位错误时被置位

21.5.3. 波特率寄存器 UART_BRR(偏移: 08h)

比特	名称	属性	复位值	描述
31:22	RSV	RO	0x00	保留
21:6	UART_IBAUD	R/W	0x0000	波特率分频整数因子 $UART\_IBAUD = (integer(F_{sys}/(16*BAUD)))$, integer 为取整操作
5:0	UART_FBAUD	R/W	0x00	波特率分频分数因子。 $UART\_FBAUD = (integer(badf*64 + 0.5))$, integer 为取整操作, badf 为 $F_{sys}/(16*BAUD)$ 小数部分 注: 若 FBAUD 计算结果大于等于 64, 将本寄存器写 0, 将原本写入 UART_IBRD 的值加 1;

21.5.4. 中断使能寄存器 UART_IE (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:15	RSV	RO	0	保留

14	ABRI	R/W	0	波特率自适应完成中断使能位 0: 禁止 1: 使能
13	IDLEI	R/W	0	IDLE 中断使能位 0: 禁止 1: 使能
12	BCNTI	R/W	0	Bit Count Timeout 中断使能位 0: 禁止 1: 使能
11	LBDI	R/W	0	LIN Break Detection 中断使能位。 0: 禁止 1: 使能
10	OEI	R/W	0	overrun 中断使能位 0: 禁止 1: 使能
9	BEI	R/W	0	break error 中断使能位 0: 禁止 1: 使能
8	PEI	R/W	0	奇偶校验错误中断使能位 0: 禁止 1: 使能
7	FEI	R/W	0	帧格式错误中断使能位 0: 禁止 1: 使能
6	RSV	RO	0	保留
5	TXI	R/W	0	发送中断使能位 0: 禁止 1: 使能
4	RXI	R/W	0	接收中断使能位

				0: 禁止 1: 使能
3:0	RSV	RO	0	保留

21.5.5. 中断和状态寄存器 UART_ISR (偏移: 10h)

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
14	ABRI	RO	0	波特率自适应功能完成原始中断 0: 未完成 1: 完成 写 1 清 0
13	IDLEI	RO	0	监测总线空闲原始中断 0: 没有检测到空闲总线 1: 检测到空闲总线 写 1 清 0
12	BCNTI	RO	0	Bit Count Timeout 原始中断 0: 无中断 1: 有中断 写 1 清 0
11	LBDI	RO	0	LIN Break Detection 原始中断 0: 无中断 1: 有中断 写 1 清 0
10	OEI	RO	0	overrun 原始中断 0: 无中断 1: 有中断 当 OE 标志产生时此位被置位 写 1 清 0

9	BEI	RO	0	break error 原始中断 0: 无中断 1: 有中断 当 BE 标志产生时此位被置位 写 1 清 0
8	PEI	RO	0	奇偶校验错误原始中断 0: 无中断 1: 有中断 当 PE 标志产生时此位被置位 写 1 清 0
7	FEI	RO	0	帧格式错误原始中断 0: 无中断 1: 有中断 当 FE 标志产生时此位被置位 写 1 清 0
6	RSV	RO	0	保留
5	TXI	RO	0	发送原始中断 0: 无中断 1: 有中断 注: 1. 使能 FIFO 功能时, 当发送时发送 FIFO 中的数据个数到达 TXIFLSEL 寄存器所设的中断触发点时此位被置位。可以通过以下 3 种方式清除该中断: a) 将 UART_ICR[5]寄存器写 1 b) 向 FIFO 中填入数据使 FIFO 中的数据数量大于 TXIFLSEL 寄存器所设的中断触发个数 c) 重新设置 TXIFLSEL 寄存器使中断触发个数由大变小 (如由 8 个触发变为 2 个触发)。 2. 禁止 FIFO 功能时, 当数据从 UART_DR 寄存器进

				入发送移位寄存器时此位被置位。可以通过以下 2 种方式清楚该中断： a) 写 UART_ICR[5]寄存器 b) 发送未完成时，写新的数据到 UART_DR 寄存器。 写 1 清 0
4	RXI	RO	0	接收原始中断 0：无中断 1：有中断 注： 1. 使能 FIFO 功能时，接收时当接收 FIFO 中的数据个数到达 RXIFLSEL 寄存器所设的中断触发点时此位被置位。可以通过以下 3 种方式清除该中断： a) 写 UART_ICR[4]寄存器 b) 将接收 FIFO 中的数据读走使 FIFO 中的数据数量小于 RXIFLSEL 寄存器所设的中断触发点个数 c) 重新设置 RXIFLSEL 寄存器使中断触发个数由小变大（如由 2 个触发变为 8 个触发）。 2. 禁止 FIFO 功能时，当 UART_DR 寄存器有新的数据进入时此位被置位。可以通过以下 2 种方式清除该中断： 1) 写 UART_ICR[4]寄存器 2) 读 UART_DR 寄存器 写 1 清 0
3:0	RSV	-	-	保留

21.5.6. 控制寄存器 1 UART_CR1 (偏移：14h)

比特	名称	属性	复位值	描述

15	CTSEn	R/W	0	CTS 流控制使能位 0: 禁止 1: 使能
14	RTSEn	R/W	0	RTS 流控制使能位 0: 禁止 1: 使能
13:12	RSV	RO	0	保留
11	RTS	R/W	0	RTS 输出管脚状态 0: RTS 输出高电平 1: RTS 输出低电平
10	RSV	R/W	0	保留
9	RXE	R/W	1	接收使能位 0: 禁止 1: 使能
8	TXE	R/W	1	发送使能位: 0: 禁止 1: 使能
7:6	RSV	RO	0	保留
5	DMAONERR	R/W	0	发生接收错误 (PE、FE、BE、OE) 时, DMA 接收请求不置位 0: 不使能 1: 使能
4	TXDMAE	R/W	0	发送 DMA 使能 0: 不使能 1: 使能
3	RXDMAE	R/W	0	接收 DMA 使能 0: 不使能 1: 使能
2	SIRLP	R/W	1'b0	IrDA SIR 低功耗模式使能位:

				0: 禁止 1: 使能
1	SIREN	R/W	1'b0	IrDA SIR ENDEC 模块使能位: 0: 禁止 1: 使能
0	UARTEN	R/W	1b'0	UART 使能位 0: 禁止 1: 使能

21.5.7. 控制寄存器 2UART_CR2 (偏移: 18h)

比特	名称	属性	复位值	描述
31:19	RSV	-	-	保留
18	ADDM7	R/W	0	是否选择 7 位地址模式 0: 4 位地址模式 1: 7 位地址模式
17:11	ADDR	R/W	0	本设备的 UART 节点地址 该位域给出本设备 UART 节点的地址。
10:8	RSV	RO	0	保留
7	CLKEN	R/W	0	CK 时钟使能 0: 禁止 CK 引脚 1: 使能 CK 引脚
6	NACK	R/W	0	智能卡 NACK 使能 0: 校验错误出现时, 不发送 NACK 1: 校验错误出现时, 发送 NACK
5	SCEN	R/W	0	智能卡模式使能 0: 禁止智能卡模式 1: 使能智能卡模式

4	ABREN	R/W	0	使能波特率自适应功能 0: 禁止波特率自适应功能 1: 使能波特率自适应功能
3	WAKE	R/W	0	静默模式唤醒方法选择 0: 被空闲总线唤醒 1: 被地址标记唤醒
2	RWU	R/W	0	接收唤醒 该位用来决定是否把 UART 进入静默模式，由软件设置或清除。当唤醒序列到来时，硬件也会将其清零。 0: 接收器处于正常工作模式 1: 接收器处于静默模式
1	RSV	RO	0	保留
0	HDSEL	R/W	0	单线半双工模式选择 0: 全双工模式 1: 单线半双工模式 单线半双工模式: RX 不再使用 TX 输出在发送时输出，其他处于接收模式。

21.5.8. 控制寄存器 3 UART_CR3 (偏移: 1Ch)

比特	名称	属性	复位值	描述
15:13	RXIFLSEL	R/W	010	接收中断的触发点选择位: 000: 1/8 (接收 FIFO 收到 2 个数据)。 001: 1/4 (接收 FIFO 收到 4 个数据)。 010: 1/2 (接收 FIFO 收到 8 个数据)。 011: 3/4 (接收 FIFO 收到 12 个数据)。 100: 7/8 (接收 FIFO 收到 14 个数据)。 101: 接收 FIFO 变非空 (接收 FIFO 收到 1 个数据)

				据)。
12:10	TXIFLSEL	R/W	010	发送中断的触发点选择位: 000: 1/8 (发送到 FIFO 中剩余 2 个数据)。 001: 1/4 (发送到 FIFO 中剩余 4 个数据)。 010: 1/2 (发送到 FIFO 中剩余 8 个数据)。 011: 3/4 (发送到 FIFO 中剩余 12 个数据)。 100: 7/8 (发送到 FIFO 中剩余 14 个数据)。 101: 发送 FIFO 变空(发送 FIFO 为空时不会产生中断,从仅剩 1 个数据到完全为空变化时才会产生中断;产生中断不代表最后一个数据发送完成,仅代表最后一个数据从 FIFO 中移除并开始发送)。 注:中断产生不是单纯由 FIFO 中的数据数量决定,触发仅产生于特定操作行为时数据数量到达中断触发点的瞬间。对于接收 FIFO,仅产生于接收数据时 FIFO 中数据数量到达中断触发点的瞬间,读取时到达触发点不会产生中断;对于发送 FIFO,中断仅产生于发送时 FIFO 中数据数量到达中断触发点的瞬间,往发送 FIFO 中写入数据达到触发点时也不会产生中断。
9	SPS	R/W	0	校验模式选择位 0: 奇/偶校验 1: 0/1 校验
8:6	WLEN	R/W	0	字宽选择位 000: 5bits 001: 6bits 010: 7bits 011: 8bits 1xx: 9bits
5	FEN	R/W	0	FIFO 使能位

				0: 禁止 FIFO 1: 使能 FIFO
4	RSV	RO	0	保留
3	STP2	R/W	0	停止位数选择位: 0: 1 位停止位 1: 2 位停止位
2	EPS	R/W	0	0/1 校验或者奇/偶校验选择位（取决于 SPS） 0: 奇/偶校验选择奇校验，或 0/1 校验选择校验位强制为 1 1: 奇/偶校验选择偶校验，或 0/1 校验选择校验位强制为 0
1	PEN	R/W	0	校验使能位: 0: 禁止奇/偶校验或 0/1 校验 1: 使能奇/偶校验或 0/1 校验
0	BRK	R/W	0	BREAK 发送使能位 0: 禁止 1: 使能

21.5.9. 保护时间和预分频寄存器 UART_GTPR(偏移: 20h)

比特	名称	属性	复位值	描述
31:12	RSV	RO	0	保留
11:8	GT	R/W	0x00	智能卡模式下使用。保护时间，以 etu 为单位
7:0	PSC	R/W	0x00	该位段在 IrDA 或智能卡模式下使用。 IrDA 模式下: $PSC = \text{integer}(F_{sys}/(16 * ILPBAUD))$, integer 为取整操作, ILPBAUD 是低功耗模式的目标波特率, 正常为 115200, 范围要求在 88750~132500 之间。 智能卡模式下（低 5 位有效）：对源时钟进行分

				频。 0: 源时钟进行 2 分频 1: 源时钟进行 4 分频 2: 源时钟进行 6 分频 31: 源时钟进行 64 分频

21.5.10. 比特计时寄存器 UART_BCNT (偏移: 24h)

比特	名称	属性	复位值	描述
31:28	RSV	-	0	保留
27	DEM	RW	0	使能 RS485 的 DE 功能 0: 禁止 1: 使能
26	DEP	RW	0	DE 信号极性选择 0: 高电平有效 1: 低电平有效
25	AUTO_START_EN	RW	0	硬件自动计时控制 0: 禁止 1: 使能 在接收数据, 如果使能此功能, 在接收到 STOP 位时自动启动一次比特计时。
24	BCNT_START	WO	0	比特计时开始。
23:0	BCNT_VALUE	R/W	13	DEM 不使能时: 比特计时初值。 DEM 使能时: [7:4]是 DEAT [3:0]是 DEDT

DE 和 RTS 共用一个管脚, DE 优先。

21.6. 使用流程

21.6.1. 串口设置

串口设置中，首先配置波特率，通过设置分频因子寄存器来完成。波特率寄存器 UART_BRR，其中 UART_IBRD 位域设置波特率计算值的整数部分， $UART\_IBAUD = (\text{integer}(F_{PCLK}/(16*BAUD)))$ ，integer 为取整操作；UART_FBRD 位域设置波特率计算值的小数部分， $UART\_FBAUD = (\text{integer}(\text{badf}*64 + 0.5))$ ，integer 为取整操作，badf 为 $F_{PCLK}/(16*BAUD)$ 小数部分。

在寄存器 UART_CR3 中：

- 1、SPS 位选择校验模式（选择奇偶校验还是 0/1 校验）；
- 2、WLEN 位域选择字宽（支持 5~9bit）；
- 3、FEN 位配置 FIFO(是否使用 FIFO)；
- 4、STP2 位配置停止位的个数；
- 5、EPS 位配置具体的校验方式（选择奇检验还是偶校验、选择强制 0 校验还是强制 1 校验，和 SPS 位有关）；
- 6、PEN 位配置校验使能；

21.6.2. 串口的发送和接收

- 配置波特率
- 配置 FIFO(是否使用 FIFO)
- 配置控制寄存器(奇偶校验位等)
- 配置中断使能寄存器(是否使用中断)
- 使能 UART

21.6.3. LIN 硬件功能支持

- LIN 作为从机，检测 Break 同步间隔帧有两种方式：
 - 直接轮询访问 UART_ISR.LBDI 位；
 - 使能 UART_IE.LBDI 位，然后触发中断，在中断查看 UART_ISR.LBDI 位。

- LIN 检测 Break 同步间隔帧的最小长度由一个当前设置的标准数据帧决定，LBDI 信号会在 RXD 数据变为高时被置为有效。
 - 1 位起始位
 - N 位数据位（由 UART_CR3.WLEN 决定，建议 8 位。）
 - 校验位（0 位或 1 位校验位，由 UART_CR3.PEN 决定）
 - 停止位（1 位或 2 位停止位，由 UART_CR3.STP2 决定）
- LIN 作为主机发送 Break 同步间隔帧流程：
 - 配置 UART_BCNT 寄存器的 BCNT_VALUE 为 13。
 - 同步使能 UART_CR3.BRK 位和 UART_BCNT.BCNT_START 位。
 - 轮询访问 UARTISR.BCNTI 位，等待置 1；或者使能中断，等待 BCNTI 中断。
 - 清零 UART_CR3.BRK 位。

21.6.4. IrDA SIR 功能使用流程

- 配置 UART_GTPR.PSC，设置 IrDA 低功耗波特率为 115200。
- 使能 UART_CR1.SIREN 位，即可打开 IrDA SIR 红外功能。
- 如若使用 IrDA SIR Low Power 模式发送数据，需要使能 UART_CR1.SIRLP 位。
- UART 模块在 IrDA SIR 模式时，总是采用低功耗波特率来采样接收的数据。

21.6.5. 单线模式功能使用流程

- 使能 UART_CR2.HDSEL 位。
- 单线模式中发送数据和接收数据都使用 TX 管脚。
- 单线模式在发送时需要开启 TXE 同时禁止 RXE；接收时开启 RXE

21.6.6. 智能卡功能使用流程

- 置位 UART_CR2.SCEN。
- 置位 UART_CR2.HDSEL，设置 TX 引脚单线半双工。
- 置位 UART_CR3.PEN，使能校验位。
- 通过 UART_GTPR.PSC 设置 CK 分频。

- 置位 UART_CR2.CLKEN，从 CK 管脚上输出时钟。
- 如果需要在支持重发机制，则关闭发送 FIFO，在发送时检测到 FE 错误时，需要软件配合重新发送上一个字节。
- 如果发送时需要额外保护时间，需要配置 UART_GTPR.GT。
- 如果需要在实现接收超时功能（BWT 和 CWT），可以配置 UART_BCNT.AUTO_START_EN 位，并配置合适大小的 UART_BCNT.BCNT_VALUE。
- 发送时，配置 UART_GTPR.GT 以增加额外保护时间。
- 接收时，使能 UART_CR2.NACK 位以检测校验错误。

21.6.7. 多机通信功能使用流程

- 多机通信网络主设备的 TX 输出和其他 UART 从设备的 RX 输入相连接；从设备各自的 TX 输出逻辑与在一起后，和主设备的 RX 输入相连接。
- 设备可通过设置 UART_CR2.RWU 进入静默模式。
- 根据需求设置 UART_CR2.WAKE 位，UART 可以用两种方法进入或退出静默模式。
 - 如果 WAKE = 0：进行空闲总线检测。
 - 如果 WAKE = 1：进行地址标记检测。
- 如果使用地址标记检测
 - 设置 UART_CR2.ADDM7 决定 7 位地址模式还是 4 位地址模式。
 - 设置 UART_CR2.ADDR 节点地址。

21.6.8. 波特率自适应功能使用流程

- 置位 UART_CR2.ABREN 位使能波特率自适应功能。
- UART 等待 RX 线路上的主机传递的 0x7F 字符。通过 UART_ISR.ABRI 位来指示自动波特率操作完成。也可 UART_IE.ABRI 产生相应的中断。
- 在波特率自适应完成后，UART_BRR 波特率寄存器值将被自动更新，需要软件将 UART_CR2.ABREN 位清零，否则会一直处于波特率自适应模式。
- 在波特率自适应模式中，UART 接收到的数据将会被丢弃。

21.6.9. RS485 的 DE 控制功能使用流程

- 将 UART_BCNT.DEM 位置 1，使能 DE 信号控制功能。
- 设置 UART_BCNT.DEAT 位域值，配置 DE 信号与 START 位之间的时间。
- 设置 UART_BCNT.DEDT 位域值，配置发送的消息中最后一个字符的停止位与 DE 信号之间的时间。
- 通过 UART_BCNT.DEP 位配置 DE 的极性。

ALSI NOCHIP

22. 内部集成电路总线接口（I2C）

22.1. 概述

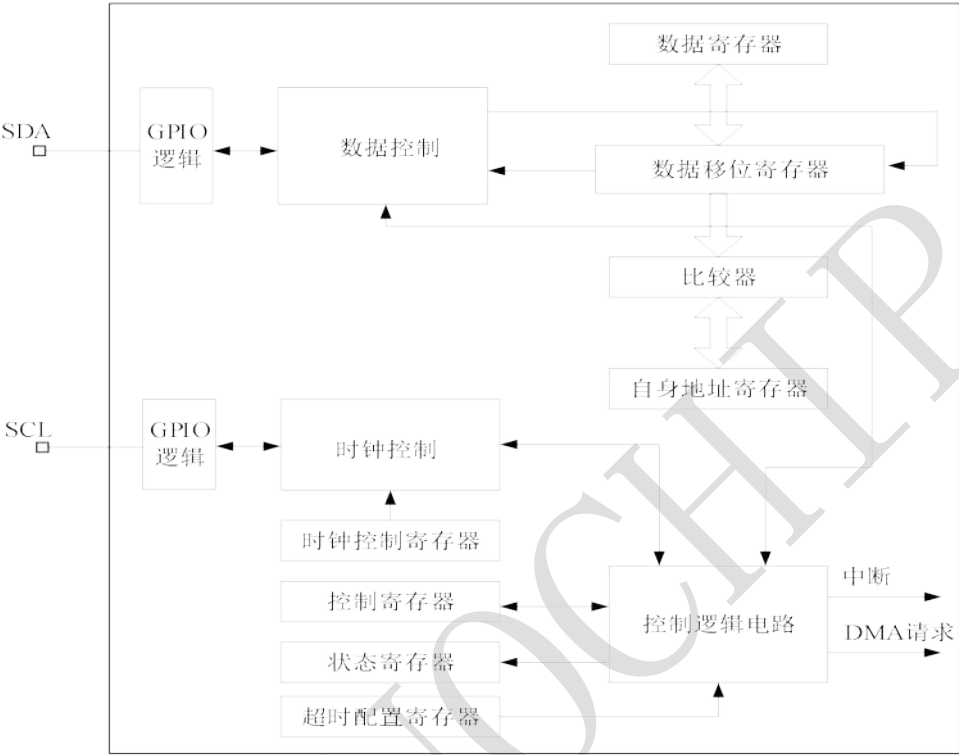
I2C 总线接口连接微控制器和串行 I2C 总线。I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。I2C 模块通过数据引脚 SDA 和时钟引脚 SCL 连接到 I2C 总线，控制所有 I2C 总线规定的时序。本模块支持主模式和从模式。

22.2. 主要特性

- I2C 主设备功能；
- I2C 从设备功能：
 - 可编程的 I2C 从设备地址；
 - 可编程的 NACK/ACK 回复；
- 速率支持 100K/400K/1M；
- 支持 7bit 设备地址，支持多个从设备地址；
- 支持从机拉时钟功能；
- 支持 DMA 收发数据；
- 支持主机仲裁模式；

22.3. 结构框图

图 22-1 I2C 结构框图

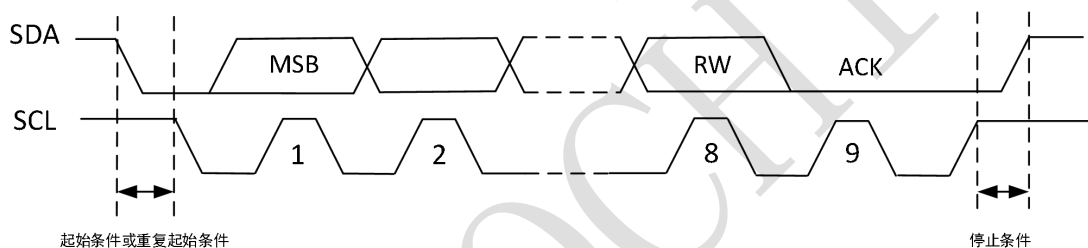


22.4. 功能描述

22.4.1. 模式选择

该模块默认为从模式，当软件写 I2C_CR.MASTER 为 1 时设备变成主模式。主模式时，I2C 接口启动数据传输并产生时钟信号，并可以发出停止条件信号停止传输。从模式时，I2C 接口能识别设置的设备地址（7 位）。数据和地址按 8 位/字节进行传输，高位在前。跟在起始条件后的是地址。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。参考下图：

图 22-2 I2C 总线传输



22.4.2. I2C 从模式

一旦检测到起始条件，在 SDA 线上接收到的地址被送到移位寄存器。然后与芯片自己的设备地址相比较，如果地址不匹配 I2C 将其忽略并等待另一个起始条件。如果地址匹配，则此控制器会检测当前操作是发送还是接收（I2C_SR.SRW 位），I2C 接口进行如下操作：

- 从发送器：
 - 发送器将字节从数据寄存器加载到内部移位寄存器发送到 SDA 线上，并将状态寄存器 TXE 置 1 表示数据寄存器数据已被取走，软件需要更新数据寄存器来清除 TXE 标志。
 - 当收到应答脉冲后，如果在下一个数据发送结束之前新数据仍然没有被写进数据寄存器，即 TXE 仍然为 1，则字节等待标志位（OVR）被置 1。此时如果 NOSTRETCH 为 1，则从机不会延长 SCL 时钟，当主机发起新的时序时，数据寄存器的数据将再次发送给主机。此时如果 NOSTRETCH 为 0 且收到 ACK 应答，则 I2C 接口保持 SCL 为低以等待新的数据被写进数据寄存器。

图 22-3 7 位从发送器的传送图



说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

:主机到从机

:从机到主机

● 从接收器：

在接收到数据后，从接收器将通过内部移位寄存器从 SDA 线接收到的字节存储到数据寄存器，并产生数据寄存器非空标志 RXNE，软件需要读出数据寄存器的值来清除 RXNE 标志。

- I2C 接口在接收到每个字节后都产生一个应答脉冲。
- 如果在接收到新数据前数据寄存器的值未被读出，即 RXNE 仍然为 1，则字节等待标志位（OVR）被置 1。此时如果 NOSTRETCH 为 0，这时 I2C 接口保持 SCL 为低以等待数据寄存器的值被读出；否则主机将继续传输数据，新收到的数据将被写到数据寄存器中，原来的数据丢失。

图 22-4 7 位从接收器的传送图



说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

● 关闭从通信

在传输完最后一个数据字节后，主设备发出一个停止操作，I2C 接口检测到这一条件时释放 SCL 和 SDA 线。

● 时钟延长

- 发送模式：当数据寄存器里的数据没有被更新时且主机应答为 ACK，把 SCL 拉低以等待新的数据写入。当主机应答为 NACK 时，SCL 不会被拉低
- 接收模式：当数据寄存器里的数据没有被读走时，把 SCL 拉低以等待旧的数据被读走。
- SCL 拉低功能可以通过 I2C_CR 的 NOSTRETCH 禁止。

图 22-5 从机发送模式时钟延长时序（虚线部分 SCL 为低，主机无法发送 SCL）



22.4.3. I2C 主模式

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。当写控制寄存器的 START 位为 1 在总线上发起一次起始条件，设备就进入了主模式传输。

以下是主模式的操作顺序：

- 使能设备，配置时钟控制寄存器和主模式。
- 配置控制寄存器的 START 位为 1，产生起始条件。
- 写入数据寄存器，开始发送数据。

● 主发送器

发送了地址后，主设备通过内部移位寄存器将字节从数据寄存器加载到 SDA 线上，并将 I2C_SR.TXE 置 1 表示数据寄存器数据已被取走，软件需要更新数据寄存器来清除 I2C_SR.TXE 标志。

收到应答脉冲后确认新的数据已经发送到数据寄存器。如果在下一个数据发送结束之前新数据仍然没有被写进数据寄存器，即 I2C_SR.TXE 仍然为 1，这时 I2C 接口保持 SCL 为低以等待新的数据被写进数据寄存器。

主设备发出 STOP 信号产生停止条件。

图 22-6 7 位主发送器的传送图



说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

● 主接收器

发送了地址后，I2C 接口从 SDA 线接收数据字节，并通过内部移位寄存器存储到数据寄存器，产生数据寄存器非空标志 I2C_SR.RXNE，软件需要读出数据寄存器的值来清除 I2C_SR.RXNE 标志。

若当前字节传输完成，不去读取数据清除 I2C_SR.RXNE，这时 I2C 接口保持 SCL 为低以等待数据寄存器的值被读出。

读出数据后，寄存器非空标志 I2C_SR.RXNE 清 0，主机开始一次新的传输。

主设备在从设备接发送最后一个字节后发送一个 NACK 应答。收到 NACK 应答后，从设备释放 SCL 线和 SDA 线的控制。主设备就可以发送一个停止/重新开始条件。

图 22-7 7 位主接收器的传送图



说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

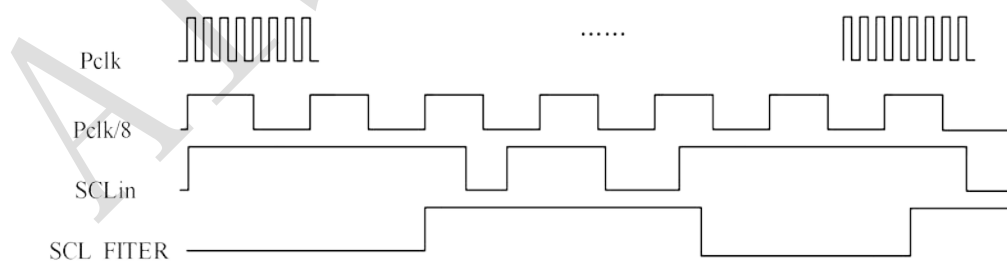
22.4.4. SCL 总线滤波算法

当 I2C_FILTER 寄存器的值为 0 时，表示 SCL 没有滤波功能。不为 0 时滤波时间为 $T_{cntc} * I2C\_FILTER$ 。其中 T_{cntc} 为 PCLK 的 8 分频时钟周期。

例：

I2C_FILTER=2 时，SCL 必须采样到连续两个 T_{cntc} 宽度的高电平才能输出高电平，宽度小于两个 T_{cntc} 的脉冲被认为是干扰毛刺而被过滤掉。

图 22-8 滤波算法示意图



注：

1、在使用过程中，设置 I2C_FILTER.SDA_IN_DELAY 与 I2C_FILTER.SCL_FILTER 的值相同，SDA 信号与 SCL 信号在经过滤波后，将保持输入时的相位。

2、在通信过程中，SCL 滤波功能只滤低于 $T_{cntc} * I2C\_FILTER$ 时间的毛刺。非通信过程中，SCL 滤波功能只滤低于 $T_{cntc} * I2C\_FILTER$ 时间的低电平，高电平不滤波；SCL

滤波功能的电平会根据 START 位自动切换。

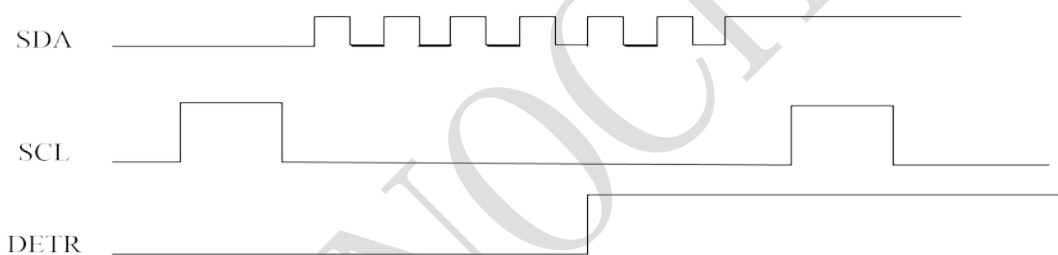
22.4.5. SCL 为低时检测 SDA 的跳变

当 I2C_DET 寄存器的值为 0 时，表示不使能检测功能，为其他值时表示使能检测功能，并且当 SCL 为低电平时检测到 SDA 的上升沿跳变次数大于 I2C_DET 寄存器的值时，把 I2C_SR.DETR 位置 1。

例：

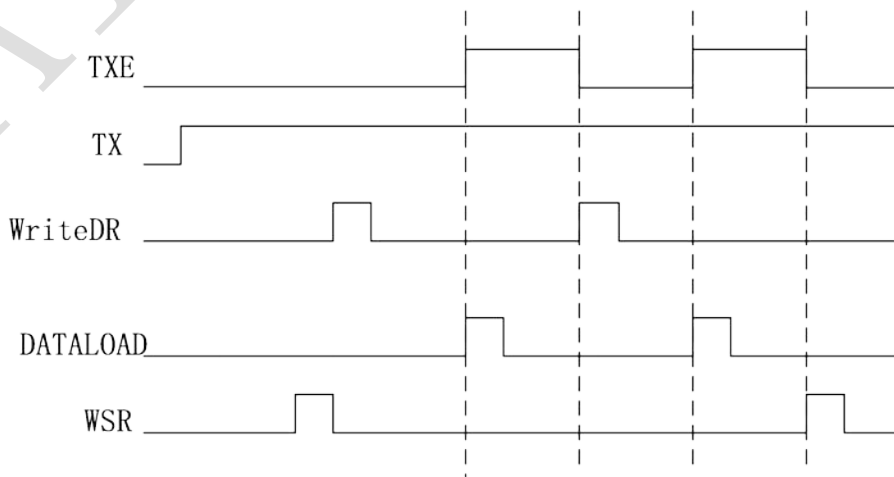
当 I2C_DET=5 时，在 SCL 低电平时，内部计数器开始工作，当检测到 SDA 上升沿跳变累积大于等于 5 时，I2C_SR.DETR=1；当 SCL 变成高电平时，内部计数器清零，等待 SCL 下个低电平到来时再开始计数。如下图所示：

图 22-9 SDA 跳变检测图



22.4.6. TXE 状态

图 22-10 TXE 功能说明图



I2C_SR.TXE 反应了 I2C_DR 数据是否被取走。

在接收模式下，TXE 为 0。

在发送模式下，写 I2C_DR 寄存器，TXE 清 0，发送器取走 I2C_DR 寄存器中的数据，TXE 置 1，TXE 位写 1 清 0；

备注：TX 1 为发送，0 为接收，DATALOAD 加载数据到数据移位寄存器，WriteDR 写 I2C_DR 寄存器，WSR 写状态寄存器。

22.4.7. SMBus 超时配置

系统管理总线（SMBus）是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。系统可使用 SMBus 与设备进行消息传递，而无需切换各个控制线。

系统管理总线规范涉及三类器件。从器件，用于接收或响应命令。主器件，用于发出命令、生成时钟和终止传输。主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主-从设备功能，并且必须支持 SMBus 主机通知协议，系统中只允许存在一个主机。

■ SMBus 与 I2C 的相似之处

- 双线制总线协议（1 个时钟总线，1 个数据总线）+ 可选 SMBus 报警线
- 主从通信，主器件提供时钟
- 多主器件功能
- SMBus 数据格式与 I2C 7 位地址格式相似

■ SMBus 与 I2C 之间的差异

SMBus	I2C
最大速度 100KHz	最大速度 400KHz
最小速度 10KHz	无最小时钟速度
35ms 时钟低电平超时	无超时
逻辑电平固定	逻辑电平取决于 VDD
地址类型不同（保留、动态）	7 位、10 位和广播从模式地址类型
总线协议不同	无总线协议

■ SMBus 应用用途

通过系统管理总线，器件可以提供制造商信息、告诉系统它的型号/部件号、保存暂停事件的状态、报告不同的错误类型、接受控制参数并返回其状态。SMBus 可针对系统和电源管理相关的任务提供控制总线。

■ 器件标识

系统管理总线中作为从器件的任何器件均具有一个唯一地址，被称为从地址。有关保留的从地址列表的信息，请参见 SMBus 规范版本 2.0 (<http://smbus.org/specs/>)。

■ 总线协议

SMBus 规范支持多达 9 类总线协议。有关这些协议和 SMBus 地址类型的详细信息，请参见 SMBus 规范版本 2.0 (<http://smbus.org/specs/>)。这些协议应通过用户软件实施。

■ 超时错误

SMBus 和 I2C 之间存在一些定时规范方面的差异。SMBus 定义了一个时钟低电平超时，TIMEOUT 为 35ms。另外，SMBus 还指定 TLOW: SEXT 作为从器件的累积时钟低电平延长时间。SMBus 指定 TLOW: MEXT 作为主器件的累积时钟低电平延长时间。有关这些超时的详细信息，请参见 SMBus 规范版本 2.0(<http://smbus.org/specs/>)。

I2C 超时配置寄存器可以配置 TIMEOUT 的时长，以 PCLK 的 2048 倍为计时长度。当时钟低电平超过 TIMEOUTA 时，TIMEOUTAF 置 1。

I2C 超时配置寄存器可以配置 TLOW: SEXT 或 TLOW: MEXT 的时长，当从器件的累积时钟低电平延长时间或主器件的累积时钟低电平延长时间超过 TIMEOUTB 时，TIMEOUTBF 置 1。

22.5. 寄存器描述

I2C1 寄存器基地址：0x4000_5400

I2C2 寄存器基地址：0x4000_5800

偏置	名称	描述
0x00	I2C_SLAVE_ADDR1	I2C 设备地址寄存器 1
0x04	I2C_CLK_DIV	I2C 时钟分频寄存器
0x08	I2C_CR	控制寄存器
0x0C	I2C_SR	状态寄存器

0x10	I2C_DR	数据寄存器
0x14	I2C_SLAVE_ADDR2/3	I2C 设备地址寄存器 2/3
0x18	I2C_DET	SDA 跳变阈值寄存器
0x1C	I2C_FILTER	I2C 滤波寄存器
0x24	I2C_Timeout	I2C 超时配置寄存器

22.5.1. I2C 设备地址寄存器 1 I2C_SLAVE_ADDR1(偏移: 0x0h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	ADDR1[7:1]	RW	0x0	地址的7~1位
0	RSV	-	-	保留

22.5.2. I2C 时钟分频寄存器 I2C_CLK_DIV(偏移: 0x04h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	I2C_CLK_DIV	RW	0x0	I2C 时钟分频值，只在主模式时设置。 $F_{scl} = (F_{pclk}) / (4 * (I2C\_CLK\_DIV + 1))$ 注： 1、Fpclk 为 APB 时钟频率，和系统时钟频率一致 2、I2C_CLK_DIV 的值必须大于 2

22.5.3. 控制寄存器 I2C_CR(偏移: 0x08h)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20	STOPF_INTEN	RW	0x0	STOPF中断使能 0: STOPF=1中断不使能 1: STOPF=1中断使能
19	RX_ADDR3_INTEN	RW	0x0	ADDR3地址匹配中断使能: 0: RX_ADDR3=1中断不使能

				1: RX_ADDR3=1中断使能
18	DMA_EN	RW	0x0	DMA功能使能 0: 不使能 1: 使能, 使用TXE和RXNE分别产生发送和接收请求。此时不能打开TXE和RXNE中断, 不能用软件清除两个状态
17	TXE_SEL	RW	0x1	从模式发送地址匹配后, TXE是否变高 0: 不变高, 兼容以前产品 1: 变高, DMA模式一定要选择
16	MARLO_INTEN	RW	0x0	主仲裁丢失中断使能: 0: MARLO=1中断不使能 1: MARLO=1中断使能
15	TX_AUTO_EN	RW	0x1	SDA数据线方向自动切换。此位设置为1, 根据地址字节的RW位自动切换SDA数据线的传输方向。 0: 不使能自动切换功能; 1: 使能自动切换功能;
14	RSV	-	-	保留
13	DETR_INT_EN	RW	0x0	SDA跳变检测中断使能: 0: DETR=1中断不使能 1: DETR=1中断使能
12	RX_ADDR2_INT_EN	RW	0x0	ADDR2地址匹配中断使能: 0: RX_ADDR2=1中断不使能 1: RX_ADDR2=1中断使能
11	OVR_INT_EN	RW	0x0	从机上溢/下溢中断使能: 0: OVR=1中断不使能 1: OVR=1中断使能
10	RXNE_INT_EN	RW	0x0	接收数据中断使能: 0: RXNE=1中断不使能 1: RXNE=1中断使能
9	TXE_INT_EN	RW	0x0	发送数据中断使能: 0: TXE=1中断不使能 1: TXE=1中断使能
8	RX_ADDR1_INT_EN	RW	0x0	ADDR1地址匹配中断使能: 0: RX_ADDR1=1中断不使能 1: RX_ADDR1=1中断使能
7	MTF_INT_EN	RW	0x0	字节传输完成中断使能: 0: MTF=1中断不使能 1: MTF=1中断使能
6	TACK	RW	0x0	传输应答位 0: 接收一字节后, 在应答周期产生ACK; 1: 接收一字节后, 在应答周期产生NACK; 注:TACK必须在应答周期前写入。
5	STOP	RW	0x0	结束条件产生位

				0: 发送完当前字节不产生结束条件; 1: 主设备在发送完当前字节后, 将产生结束条件。产生结束条件后, 硬件自动清0
4	START	RW	0x0	起始条件产生位 0: 主模式下不产生起始条件 1: 主模式下产生起始条件 注: 空闲时刻和NACK/ACK应答后才可产生起始条件, 起始条件产生后, 硬件自动清0, I2C_SR的STARTF位置1。
3	TX	RW	0x0	发送接收选择位 0: 设备作为接收器 1: 设备作为发送器 当作为从设备时, 处理器应该查询I2C_SR的SRW位, 判断是作为发送器还是接收器, 然后设置与之匹配的TX位 TX_AUTO_EN位使能后TX位设置无效
2	MASTER	RW	0x0	主从设备选择位 0: 从模式 1: 主模式。
1	NOSTRETCH	RW	0x0	从模式禁止时钟延长 0: 使能时钟延长 1: 禁止时钟延长
0	MEN	RW	0x0	设备使能位 0: 设备不使能 1: 设备使能

注: TX_AUTO_EN 自动使能后 I2C 模块 SDA 方向由硬件自动切换, 无需软件配置, 主机模式下默认为发送器, 根据第一个发送的地址的 RW 位自动切换发送器或接收器, 从机模式下默认为接收器, 根据第一个接收的地址的 RW 位自动切换发送器或接收器。

22.5.4. 状态寄存器 I2C_SR(偏移: 0x0Ch)

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留位
16	TIMEOUTBF	RO	0x0	TIMEOUTB超时标志, 表示SMBus EXT超时 写1清0
15	TIMEOUTAF	RO	0x0	TIMEOUTA超时标志, 表示SMBus SCL Timeout超时 写1清0
14	RX_ADDR3	RO	0x0	从设备地址3匹配状态位 0: 设备地址3和接收到的地址不相等 1: 设备地址3和接收到的地址相等 写1清0

				注：当地址匹配时，SRW位表示了地址字节的RW位。
13	DETR	RO	0x0	SDA跳变检测状态位 0：当SCL为低时SDA上升沿跳变次数没有超过I2C_DET所设置的检测阈值 1：当SCL为低时SDA上升沿跳变次数超过I2C_DET所设置的检测阈值 写1清0
12	RX_ADDR2	RO	0x0	从设备地址2匹配状态位 0：设备地址2和接收到的地址不相等 1：设备地址2和接收到的地址相等 写1清0 注：当地址匹配时，SRW位表示了地址字节的RW位。
11	OVR	RO	0x0	从机上溢/下溢状态位 0：未发生上溢/下溢 1：发生上溢/下溢 从机收到的字节尚未读取，并收到新的字节时OVR置1。 从机发送过程中从机需要发送一个新的字节但尚未向DR寄存器写入数据时OVR置1。
10	RXNE	RO	0x0	接收数据时数据寄存器状态位 0：接收时数据寄存器空 1：接收时数据寄存器非空 硬件置位，通过读数据寄存器I2C_DR可以清除该位，写1清0。 注：主机接收模式下收到数据后必须读取DR寄存器数据否则主机不会产生新的读时序
9	TXE	RO	0x0	发送数据时数据寄存器状态位 0：I2C_DR数据未被发送器取走 1：I2C_DR数据被发送器取走 硬件置位，写1清0，写DR清0 注：主机发送模式下，写DR寄存器后主机才会发送数据
8	RX_ADDR1	RO	0x0	从设备地址1匹配状态位 0：设备地址1和接收到的地址不相等 1：设备地址1和接收到的地址相等 写1清0 注：当地址匹配时，SRW位表示了地址字节的RW位。
7	MTF	RO	0x0	字节传输完成状态位 0：字节传输未完成 1：字节传输完成 当一个字节数据（包括地址）正在传输时，该位为0； 在一个字节传输完后，在第9个SCL时钟下降沿（应答周期）MTF被置为1。写1清0
6	MARLO	RO	0x0	主模式仲裁丢失

				0: 没有检测到仲裁丢失 1: 检测到仲裁丢失 主发送 SDA 为高, 但接收到 SDA 为低时认为丢失仲裁, 需要软件如理
5	TX_RX_FLAG	RO	0x0	主机发送转接收状态位 0: 主机未由发送状态转成接收状态 1: 主机由发送状态转为接收状态。 硬件置1, 写1清0; 注: 当TX_RX_FLAG状态置1后, 硬件停止I2C时钟, 需软件清该状态。从模式下该状态无效。
4	BUS_BUSY	RO	0x0	总线忙碌状态位 0: 总线上无数据通信(检测到总线上的结束条件, 此位置0) 1: 总线上正在进行数据通信(检测到总线上的起始条件, 此位置1)
3	SRW	RO	0x0	从机读状态指示位 0: 作为从设备接收器 1: 作为从设备发送器 当地址匹配后, SRW指示地址字节中的RW位, 该位仅在如下条件有效: 一个完整的传输已经发生, 没有其他传输被初始化; 并且I2C被配置为从模式, 且从地址匹配。当接收到停止条件或一个新的起始条件, 该位自动清除
2	STOPF	RO	0x0	STOP条件位检测位, 主从都会产生 1: 检测到停止位 0: 未检测到停止位 只能写1清0
1	STARTF	RO	0x0	主机起始条件发送状态位 1: 起始条件已发送 0: 起始条件未发送 写1清0 注: I2C_DR寄存器有数据将自动启动发送时序并清除该位, 主机模式下, SDA方向不由硬件切换并处于接收状态时STARTF也将被清除。
0	RACK	RO	0x1	应答接收状态位 0: 最近的发送应答周期接收到ACK 1: 最近的发送应答周期接收到NACK 只有START条件将清除RACK位 注: 主机发送模式下收到NACK停止发送, 等待软件处理事件

22.5.5. 数据寄存器 I2C_DR(偏移: 0x10h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	I2CDR	RW	0x0	I2C 数据寄存器

22.5.6. I2C 设备地址寄存器 2/3 I2C_SLAVE_ADDR2/3 (偏移: 0x14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:9	ADDR3[7:1]	RW	0x0	地址的 7~1 位
8	ADDR3_EN	RW	0	0: SLAVE_ADDR3 地址匹配不使能 1: SLAVE_ADDR3 地址匹配使能
7:1	ADDR2[7:1]	RW	0x0	地址的 7~1 位
0	ADDR2_EN	RW	0	0: SLAVE_ADDR2 地址匹配不使能 1: SLAVE_ADDR2 地址匹配使能

22.5.7. SDA 跳变阈值寄存器 I2C_DET(偏移: 0x18)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	DETCNT	RW	0x0	当SCL线为低时，SDA线上跳变次数阈值。 0: 没有检测功能 其他值: SDA的上升沿跳变次数阈值

22.5.8. I2C 滤波寄存器 I2C_FILTER(偏移: 0x1C)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12:8	SDA_IN_DELAY	RW	0x0	SDA 输入延时设置位，SCL 滤波功能使能后，SDA_IN_DELAY 与SCL_FILTER的值设置相同，SDA信号与SCL信号在经过滤波后，将保持输入时的相位。
7:5	RSV	-	-	保留

4:0	SCL_FILTER	RW	0x0	滤波0算法SCL滤波值设置位。 滤的毛刺的最大宽度为 $T_{fclk} * 8 * SCL\_FILTER$ 。Tfclk为系统时钟周期

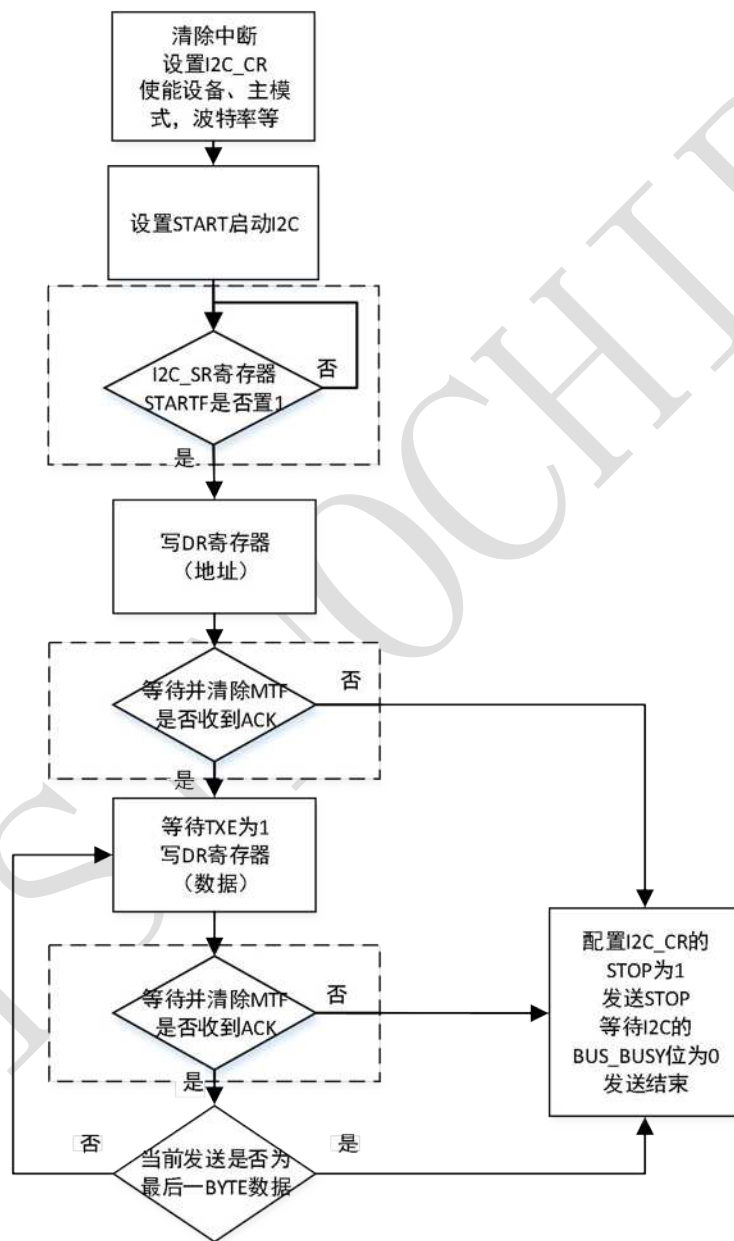
22.5.9. I2C 超时配置寄存器 I2C_TIMEOUT(偏移: 0x24h)

比特	名称	属性	复位值	描述
31	EXTEN	RW	0x0	TimeoutB定时使能, 超出后产生TIMEOUTB标志
30	TOUTB_INTEN	RW	0x0	TIMEOUTBF标志中断使能
29	EXT_MODE	RW	0x0	EXT计数(TimeoutB)模式选择 0: 计数SEXT模式 1: 计数MEXT模式
28	RSV	-	-	保留
27:16	TIMEOUTB	RW	0x0	SMBus EXT时间设置, 以PCLK的2048倍为计时长度
15	TIMEOUTEN	RW	0x0	TimeoutA定时使能, 超出后产生TIMEOUTA标志
14	TOUTA_INTEN	RW	0x0	TIMEOUTAF标志中断使能
13:12	RSV	-	-	保留
11:0	TIMEOUTA	RW	0x0	SMBus Timeout时间设置, 以PCLK的2048倍为计时长度

22.6. 软件操作流程

22.6.1. 作为主发送器

图 22-11 主机发送流程图（虚线步骤可跳过，下同）



1. 清除中断标志。
2. 写 I2C_CLK_DIV 寄存器的值确定 I2C 传输频率。
3. 写 I2C_CR 寄存器的 TX、MEN、MASTER 和 START 为 1，发起 START 条件。
4. 等待 I2C_SR.STARTF 标志为 1(也可不等待)，把 I2C 要访问的 SLAVE 的 7 位

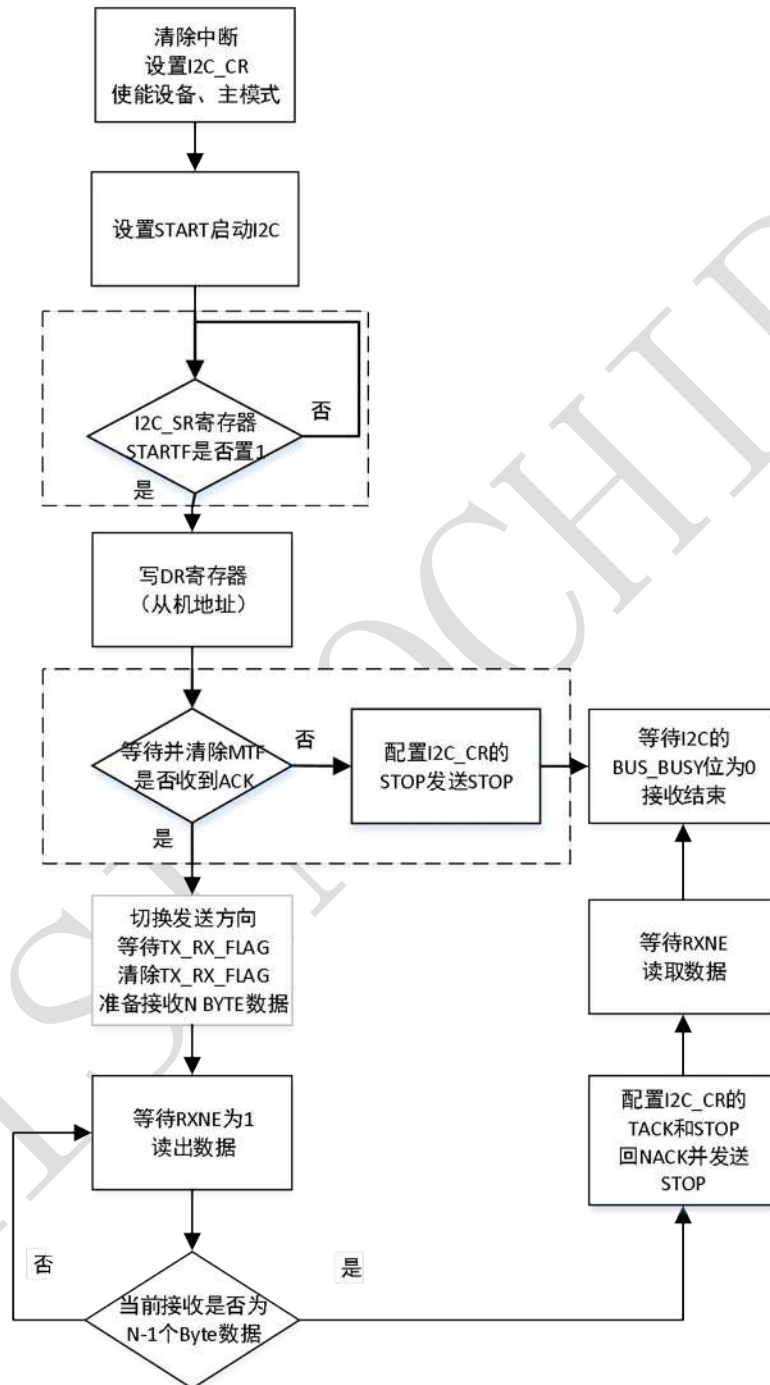
地址写入 I2C_DR 寄存器中。(如若直接写数据或 start 之前数据已写入, STARTF 将被清 0)

5. 等待 I2C_SR.MTF 标志并清除 MTF, 判断是否收到 ACK 后, 表示从机正确。如果收到 NACK, 软件写 I2C_CR.STOP 位为 1, 结束发送并释放总线, 软件等待 I2C_SR.BUS_BUSY 为 0 后退出。
6. 等待 I2C_SR.TXE 为 1, 往 I2C_DR 寄存器写入要发送的字节, 同时硬件会清除 TXE 位。若当前数据发送完成后, 新的数据没有写入, 主机不会产生新的发送时序。
7. 等待 I2C_SR.MTF 标志并清除 MTF, 判断是否收到 ACK, 如果收到 ACK 表示从机正确。如果收到 NACK, 软件写 I2C_CR.STOP 位为 1, 结束发送并释放总线, 软件等待 I2C_SR.BUS_BUSY 为 0 后退出。
8. 重复 6—7 操作。在 7 操作中如确定回 ACK, 可跳过。
9. 向 I2C_DR 写完最后一个字节后, 等到倒数第二个字节发送完成 (MTF==1)。最后一个字节写 I2C_CR.STOP 为 1 结束发送。STOP 也可以在最后字节发送完成后写入。

I2C_CR.TX_AUTO_EN 使能后, I2C_CR.TX 位可以不配置, 主默认作为发送器, 地址字节最后 1bit, 为 1 则为接收器, 为 0 则为发送器。

22.6.2. 作为主接收器

图 22-12 主机接收流程图

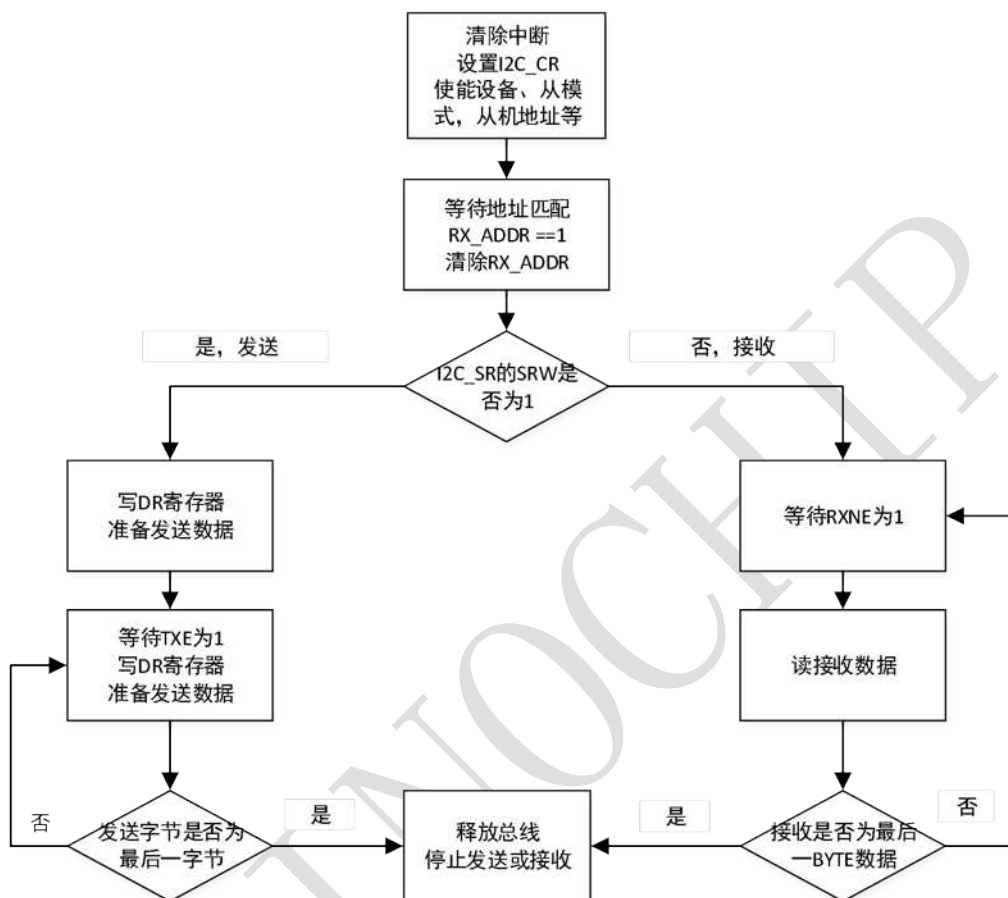


1. 清除中断标志
2. 写 I2C_CLK_DIV 寄存器的值确定 I2C 传输频率
3. 写 I2C_CR 寄存器的 MEN、MASTER 和 START 为 1，发起 START 条件。
4. 等待 I2C_SR.STARTF 标志为 1(也可不等待)，把 I2C 要访问的 SLAVE 的 7 位

- 地址和 1bit 的 1 写入 I2C_DR 寄存器中，表示作为接收器。
5. 等待 I2C_SR.MTF 标志并清除 MTF，判断是否收到 ACK 后，如果收到 ACK 表示从机正确。如果收到 NACK，软件写 I2C_CR 的 STOP 位，结束发送并释放总线，软件等待 I2C_SR.BUS_BUSY 位为 0 后退出。或者收到 NACK 后，软件写 I2C_CR.START 位为 1，重新发起一次新的操作。（写 STOP 位后，需要重新写 DR 寄存器）。
 6. 切换发送为接收后，软件需要查看 I2C_SR.TX_RX_FLAG，若为 1，清除该状态。若该状态不清除。硬件会停止接收时序。
 7. 读到 I2C_SR.RXNE 为 1 时，处理器需读取 I2C_DR 寄存器中接收到的字节，同时硬件会清除 RXNE 位。等待下一个字节的接收结束。数据接收完成后若软件不读取数据，硬件将不会产生 SCL 时钟接收新的数据。
 8. 重复 7 操作。
 9. 当倒数第二个字节接收完成，且发送完 ACK 信号后(即倒数第二个 RXNE 标志)，主机写 I2C_CR 的 TACK 和 STOP 为 1，表示下一个要接收的字节为最后一个字节。
 10. 最后一个字节接收完成后硬件发出 NACK 信号并产生 STOP 条件，软件等待 I2C_SR.BUS_BUSY 为 0 后退出。

22.6.3. 作为从发送器

图 22-13 从机发送接收流程图



1. 清除中断标志
2. 向 I2C_SLAVE_ADDR1 寄存器或 I2C_SLAVE_ADDR2/3 寄存器写入 7 位地址作为自己在从机状态下被寻址的地址。
3. 写 I2C_CR.MEN 为 1，使能 I2C 模块。
4. 等待 RX_ADDR1、RX_ADDR2 (ADDR2_EN=1) 或 RX_ADDR3 (ADDR3_EN=1) 标志是否有效。地址匹配无效则重复 4。
5. 地址匹配有效，判断 SRW 位是否为 1。为 0 表示从接收，为 1 表示从发送。
6. 写 I2C_CR.TX 为 1，切换到发送器，写第一个要发送的数据给 I2C_DR。
7. 等待 I2C_SR.TXE 为 1 时，向 I2C_DR 寄存器中写入即将要发送的数据，同时硬件会清除 TXE 位。
8. 重复 7，当收到主机发来的 STOP 后，I2C 模块释放总线。软件等待 I2C_SR 的 BUS_BUSY 为 0 后退出。

注：I2C_CR.TX_AUTO_EN 使能后，I2C_CR 位的 TX 位可以不配置，从默认作为接收器，到地址字节最后 1bit，如果是 1 则为发送器，为 0 则为接收器。

22.6.4. 作为从接收器

1. 清除中断标志
2. 向 I2C_SLAVE_ADDR1 寄存器或 I2C_SLAVE_ADDR2/3 寄存器写入 7 位地址作为自己在从机状态下被寻址的地址。
3. 写 I2C_CR 寄存器的 MEN 为 1，使能 I2C 模块。
4. 等待 RX_ADDR1、RX_ADDR2 (ADDR2_EN=1) 或 RX_ADDR3 (ADDR3_EN=1) 标志是否有效。地址匹配无效则重复 4。
5. 地址匹配有效，判断 I2C_SR.SRW 位是否为 1。为 0 表示从接收，为 1 表示从发送
6. 等待 I2C_SR.RXNE 为 1 时，读取 I2C_DR 寄存器中接收到的字节，同时硬件会清除 RXNE 位。
7. 重复 6，当收到主机发来的 STOP 后，I2C 模块释放总线。软件等待 I2C_SR.BUS_BUSY 为 0 后退出。

23. 音频接口（I2S）

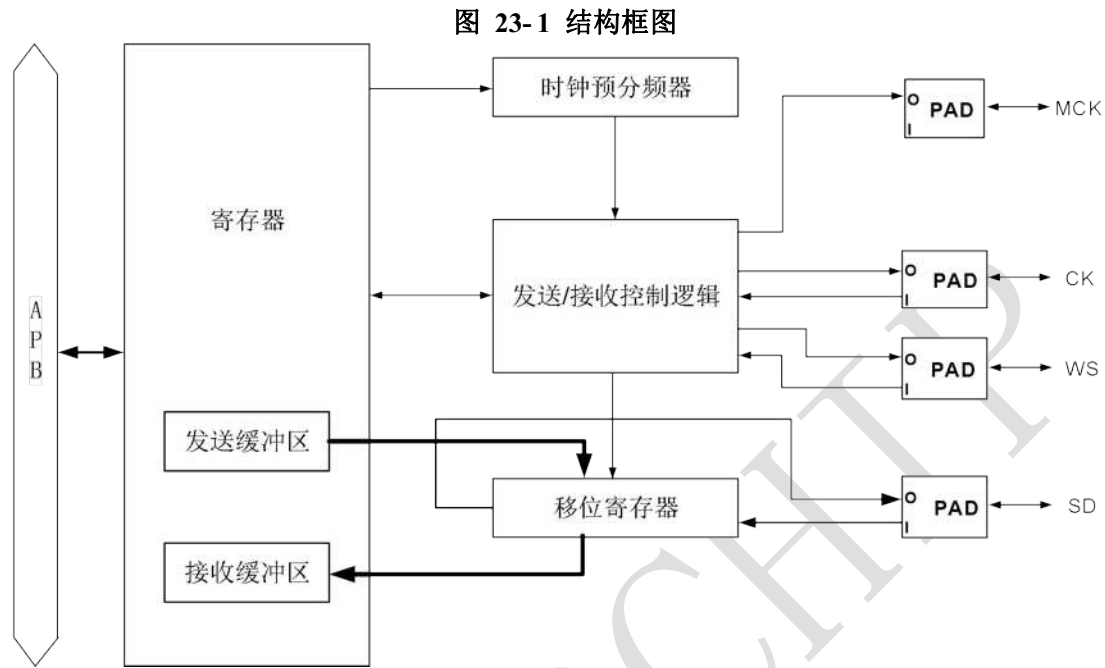
23.1. 概述

片上音频接口（Inter-IC Sound，缩写为 I2S）模块可以通过 I2S 音频协议与外部设备进行通信。I2S 接口支持四种音频标准，分别是 I2S 飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

23.2. 主要特性

- 具有发送和接收功能的主从操作；
- 支持四种 I2S 音频标准：飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准；
- 数据长度可以为 16 位，24 位和 32 位；
- 通道长度为 16 位或 32 位；
- 32 位缓冲区用于发送和接收；
- 通过 I2S 时钟分频器，可以得到 8 kHz 到 192 kHz 的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟（MCK）；
- 发送和接收支持 DMA 功能。

23.3. 结构框图



23.4. 寄存器描述

I2S 寄存器基地址:

I2S1: 0x4000_3800

I2S2: 0x4000_3C00

偏置	名称	描述
0x00	I2S_DR	数据寄存器
0x04	I2S_CR	控制寄存器
0x08	I2S_PR	时钟预分频寄存器
0x0C	I2S_DIER	中断使能寄存器
0x10	I2S_SR	状态寄存器

23.4.1. 数据寄存器 I2S_DR（偏移：00h）

比特	名称	属性	复位值	描述
31 : 0	DR[31:0]	RW	0x00000000	数据寄存器

				硬件有两个缓冲区：发送缓冲区和接收缓冲区。向 I2S_DR 写数据将会把数据存入发送缓冲区，从 I2S_DR 读数据，将从接收缓冲区获得数据。

23.4.2. 控制寄存器 I2S_CR（偏移：04h）

比特	名称	属性	复位值	描述
31: 11	RSV	-	-	保留
10	I2SE	RW	0	I2S 使能 0: I2S 禁止; 1: I2S 使能;
9: 8	I2SOPMOD	RW	00	I2S 运行模式 00: 从机发送模式 01: 从机接收模式 10: 主机发送模式 11: 主机接收模式 当 I2S 模式关闭时配置该位。
7	PCMSMOD	RW	0	PCM 帧同步模式 0: 短帧同步 1: 长帧同步 只有在 PCM 标准下，该位才有意义。 当 I2S 模式关闭时配置该位。
6	RSV	-	-	保留
5: 4	I2SSTD	RW	00	I2S 标准选择 00: I2S 飞利浦标准 01: MSB 对齐标准 10: LSB 对齐标准 11: PCM 标准 当 I2S 模式关闭时配置该位。

3	CKPL	RW	0	空闲状态时钟极性 0: I2S_CK 空闲状态为低电平 1: I2S_CK 空闲状态为高电平 当 I2S 模式关闭时配置该位。
2: 1	DTLEN	RW	00	数据长度 00: 16 位 01: 24 位 10: 32 位 11: 保留 当 I2S 模式关闭时配置该位。
0	CHLEN	RW	0	通道长度 0: 16 位 1: 32 位 通道长度必须大于或等于数据长度。 当 I2S 模式关闭时配置该位。

23.4.3. 时钟预分频寄存器 I2S_PR (偏移: 08h)

比特	名称	属性	复位值	描述
31: 11	RSV	-	-	保留
10	MCKOE	RW	0	I2S_MCK 输出使能 0: I2S_MCK 输出禁止 1: I2S_MCK 输出使能 当 I2S 模式关闭时配置该位。
9	OF	RW	0	预分频器的奇系数 0: 实际分频系数为 $DIV * 2$ 1: 实际分频系数为 $DIV * 2 + 1$ 当 I2S 模式关闭时配置该位。
8: 0	DIV	RW	0x000	预分频器的分频系数

				实际分频系数是 $DIV * 2 + OF$。 实际分频系数不能为 0，最低为 1（不分频），$DIV=0$，$OF=1$。 当 I2S 模式关闭时配置该位。

23.4.4. 中断使能寄存器 I2S_DIER（偏移：0Ch）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	TBEIE	RW	0	发送缓冲区空中断使能 0: TBE 中断禁止 1: TBE 中断使能。当 TBE 置位时，产生中断。
6	RBNEIE	RW	0	接收缓冲区非空中断使能 0: RBNE 中断禁止。 1: RBNE 中断使能。当 RBNE 置位时，产生中断。
5	ERRIE	RW	0	错误中断使能 0: 错误中断禁止 1: 错误中断使能。当 CONFERR 位，RXORERR 位或者 TXURERR 位置 1 时，产生中断。
4:2	RSV	-	-	保留
1	TXDMATEN	RW	0	发送缓冲区 DMA 使能 0: 发送缓冲区 DMA 禁止 1: 发送缓冲区 DMA 使能。当 I2S_SR 中的 TBE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。
0	RXDMAREN	RW	0	接收缓冲区 DMA 使能

				0: 接收缓冲区 DMA 禁止 1: 接收缓冲区 DMA 使能。当 I2S_SR 中的 RBNE 置位时, 将会在相应的 DMA 通道上产生一个 DMA 请求。

23.4.5. I2S 状态寄存器 I2S_SR (偏移: 10h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8	FERR	RC W0	0	帧错误 I2S 模式: 0: 没有 I2S 帧错误发生 1: I2S 帧错误发生 WS 的帧头与对应的 bit 位发生错位时, 判定为帧错误。 该位由硬件置位, 可以通过写 0 清除或读状态寄存器清除。
7	TRANS	RO	0	通信进行中标志 0: I2S 空闲 1: I2S 当前正在发送或接收数据 该位由硬件置位和清除。
6	RXORERR	RO	0	接收过载错误标志 0: 没有接收过载错误发生 1: 接收过载错误发生 从机: 接受数据时, 接受缓冲区已满, 主机仍提供时钟时, 则发生接受过载。 主机: 接受数据时, 接受缓冲区已满, 但从机还在发送数据, 则发生接受过

				载。 该位由硬件置位，软件序列清零。软件序列为：先读 I2S_DR 寄存器，然后读 I2S_SR 寄存器。
5: 4	RSV	-	-	保留
3	TXURERR	RO	0	发送欠载错误标志 0：无发送欠载错误发生 1：发送欠载错误发生 从机：发送数据时，发生缓冲区已经空，但主机仍提供时钟，则发生发送欠载。 该位由硬件置位，通过写 I2S_SR 寄存器清除。
2	I2SCH	RO	0	I2S 通道标志 0：下一个将要发送或接收的数据属于左通道 1：下一个要发送或接收的数据属于右通道 该位由硬件置位和清除。 I2S PCM 模式下该位没有意义。
1	TXBE	RO	1	发送缓冲区空 0：发送缓冲区非空 1：发送缓冲区空
0	RXBNE	RO	0	接收缓冲区非空 0：接收缓冲区空 1：接收缓冲区非空

23.5. I2S 音频标准

I2S 音频标准是通过设置 I2S_CR 寄存器中的 I2SSTD 位来选择的, 可以选择四种音频标准: I2S 飞利浦标准, MSB 对齐标准, LSB 对齐标准和 PCM 标准。除 PCM 之外的所有标准都是两个通道(左通道和右通道)的音频数据分时复用 I2S 接口的, 并通过 I2S_WS 信号来区分当前数据属于哪个通道。对于 PCM 标准, I2S_WS 信号表示帧同步信息。

数据长度和通道长度可以通过 I2SCR 寄存器中的 DTLEN 位和 CHLEN 位来设置。由于通道长度必须大于或等于数据长度, 所以有四种数据包类型可供选择。它们分别是: 16 位数据打包成 16 位数据帧格式, 16 位数据打包成 32 位数据帧格式(剩余数据补零, 同下), 24 位数据打包成 32 位数据帧格式, 32 位数据打包成 32 位数据帧格式。用于发送和接收的数据缓冲区都是 32 位宽度。

对于所有标准和数据包类型来说, 数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说, 总是先发送左通道, 然后是右通道。

23.5.1. I2S 飞利浦标准

对于 I2S 飞利浦标准, I2S_WS 和 I2S_SD 在 I2S_CK 的下降沿变化, 各种配置情况的时序图如下所示。

图 23-2 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

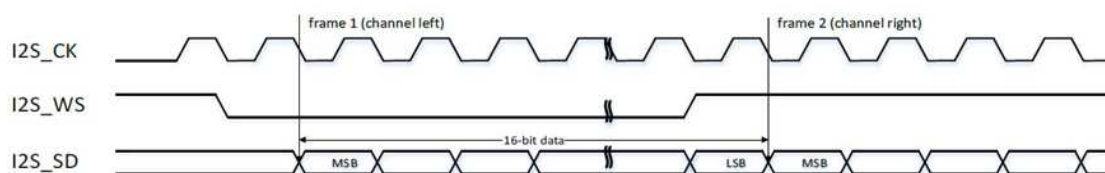


图 23-3 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

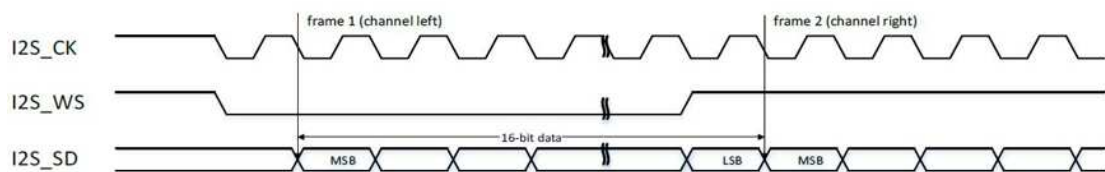


图 23-4 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

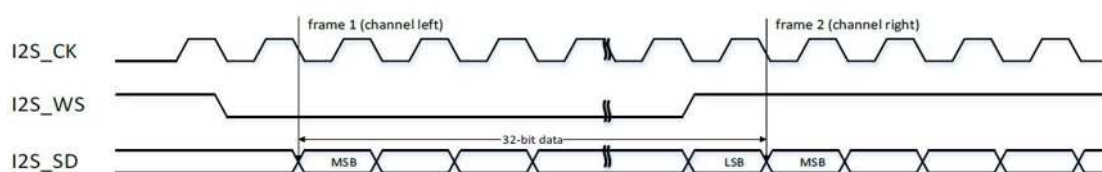
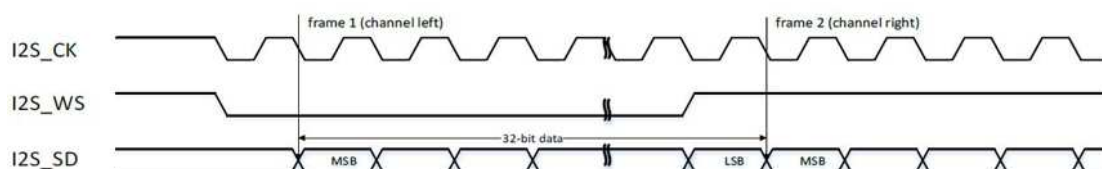


图 23-5 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)



当 24 位数据打包成 32 位数据帧的帧格式时,为了将该 24 位数据扩展成 32 位数据,剩下的 8 位被硬件强制填充为 0x00。

图 23-6 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

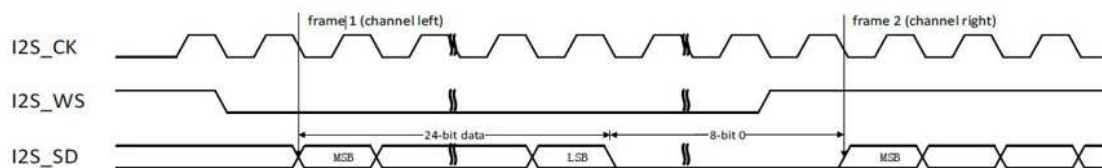
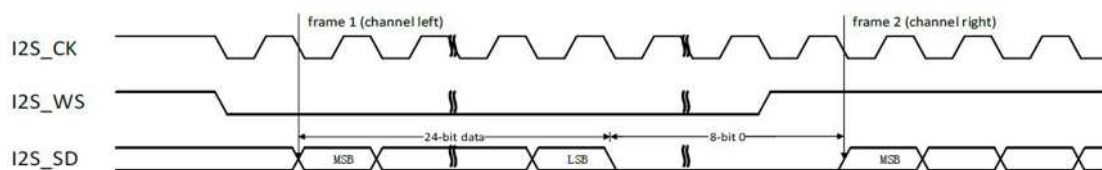


图 23-7 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



当 16 位数据打包成 32 位数据帧时,为了将该 16 位数据扩展成 32 位数据,剩下的 16 位被硬件强制填充为 0x0000。

图 23-8 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

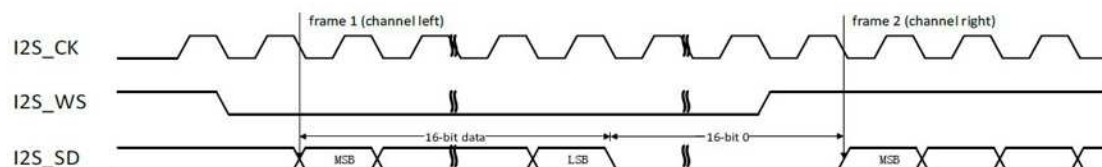
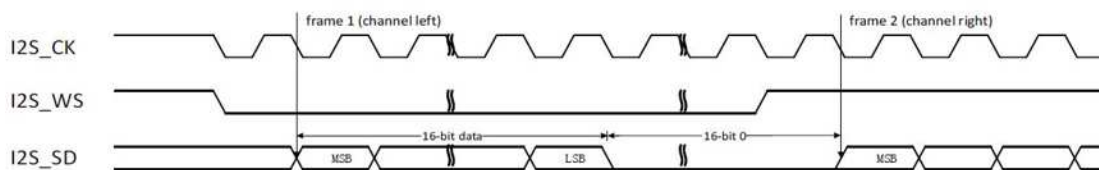


图 23-9 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



23.5.2. MSB 对齐标准

对于 MSB 对齐标准, I2S_WS 和 I2S_SD 在 I2S_CK 的下降沿变化。SPI_DATA 寄存器的处理方式与 I2S 飞利浦标准完全相同。各个配置情况的时序图如下所示。

图 23-10 MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

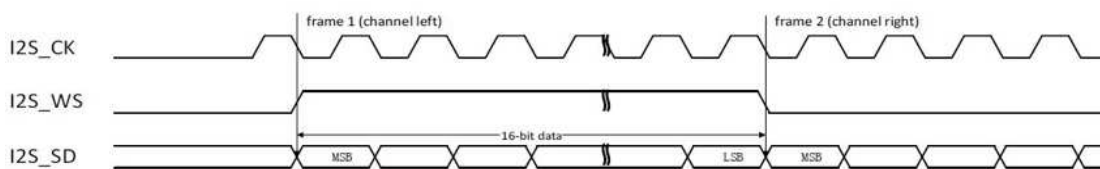


图 23-11 MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

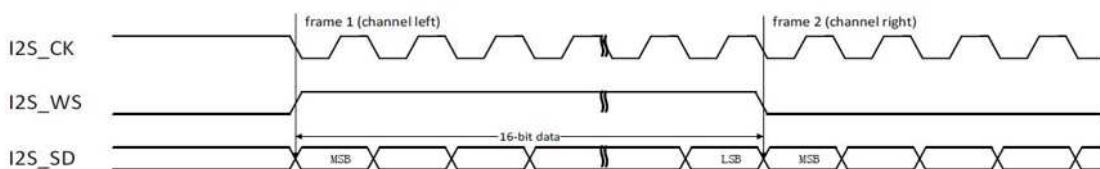


图 23-12 MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

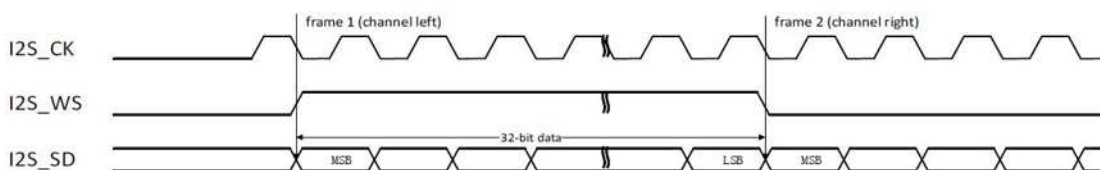


图 23-13 MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)

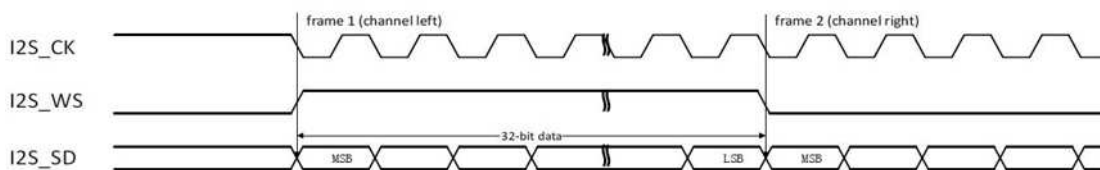


图 23-14 MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

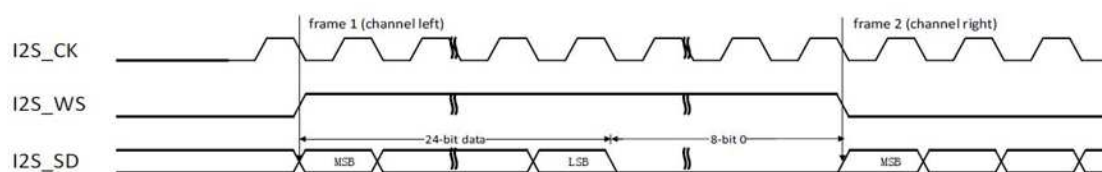


图 23-15 MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)

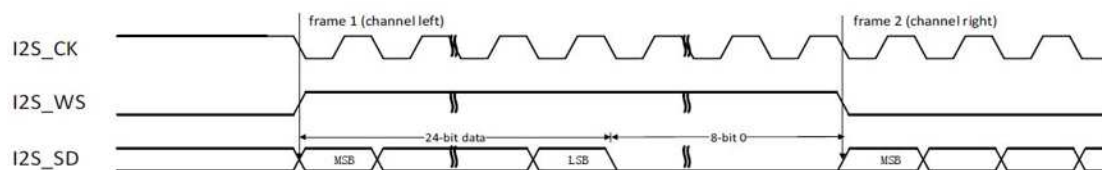


图 23-16 MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

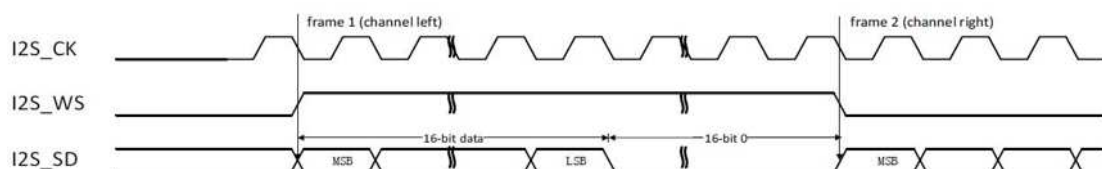
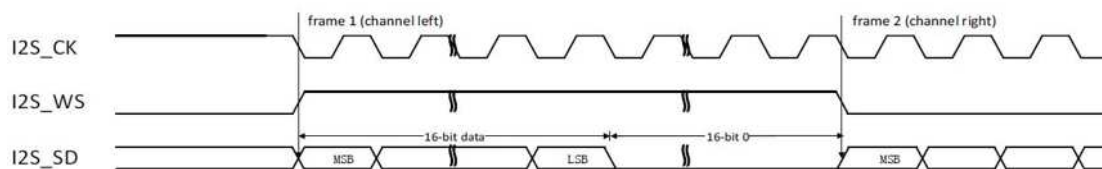


图 23-17 MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



23.5.3. LSB 对齐标准

对于 LSB 对齐标准, I2S_WS 和 I2S_SD 在 I2S_CK 的下降沿变化。在通道长度与数据长度相同的情况下, LSB 对齐标准和 MSB 对齐标准是完全相同的。对于通道长度大于数据长度的情况, LSB 对齐标准的有效数据与最低位对齐, 而 MSB 对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。当 24 位数据打包成 32 位数据帧时, 为了将该 24 位数据扩展成 32 位数据, 剩下的 8 位被硬件强制填充为 0x00。

图 23-18 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

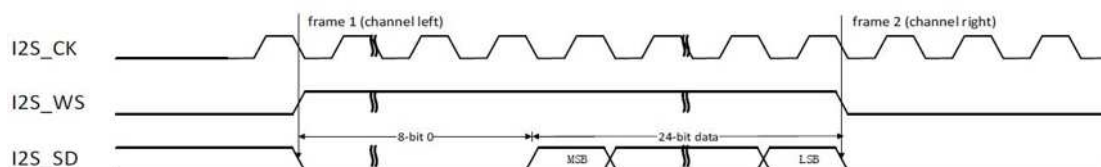
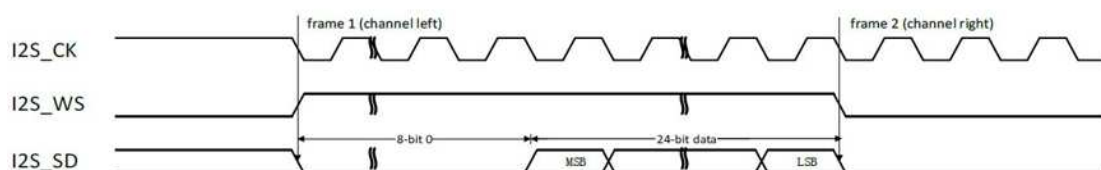


图 23-19 LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



当 16 位数据打包成 32 位数据帧时，为了将该 16 位数据扩展成 32 位数据，剩下的 16 位被硬件强制填充为 0x0000。

图 23-20 LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

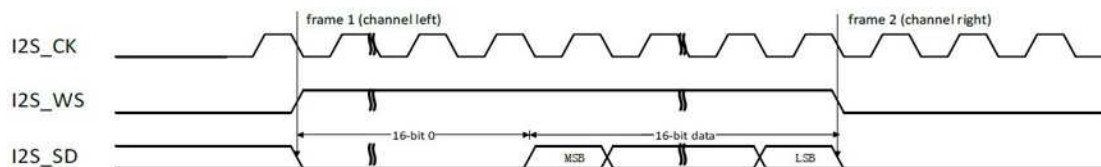
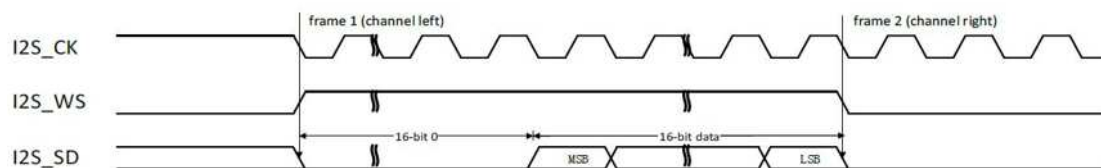


图 23-21 LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



23.5.4. PCM 对齐标准

对于 PCM 标准，I2S_WS 和 I2S_SD 在 I2S_CK 的上升沿变化，I2S_WS 信号表示帧同步信息。可以通过 SPI_I2SCTL 寄存器的 PCMSMOD 位来选择短帧同步模式和长帧同步模式，短帧指 I2S_WS 只保持一位数据长度的时间，而长帧保持 13 位数据长度的时间，帧头与数据的关系请参看时序图。SPI_DATA 寄存器的处理方式与 I2S 飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 23-22 PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

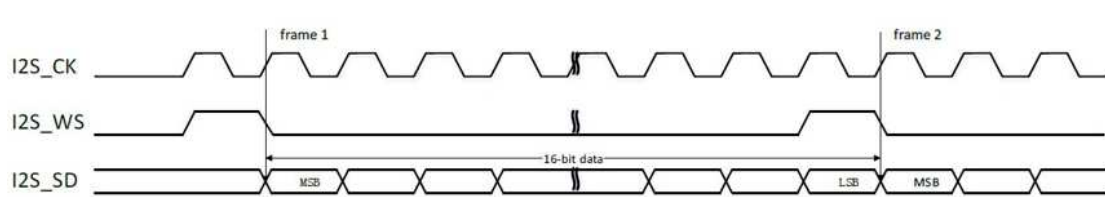


图 23-23 PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

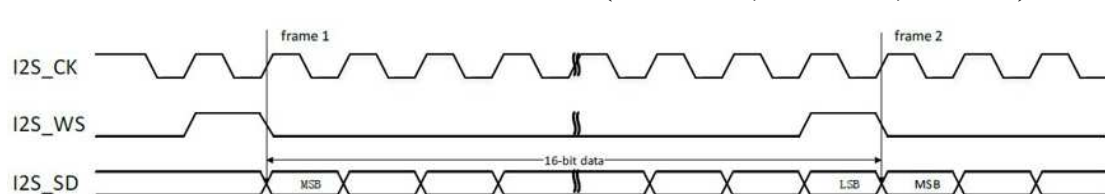


图 23-24 PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

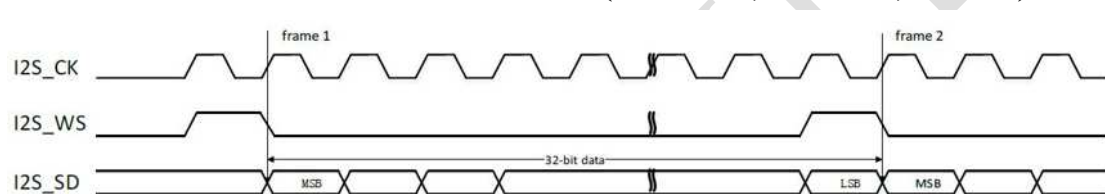


图 23-25 PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

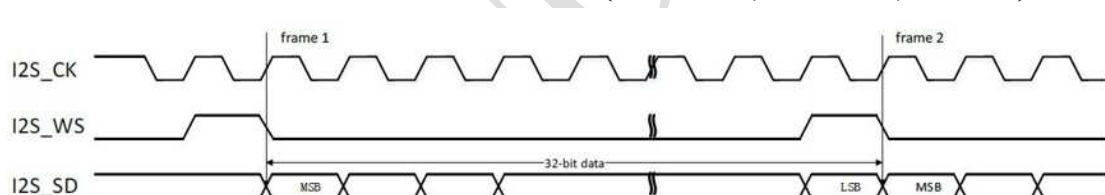


图 23-26 PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

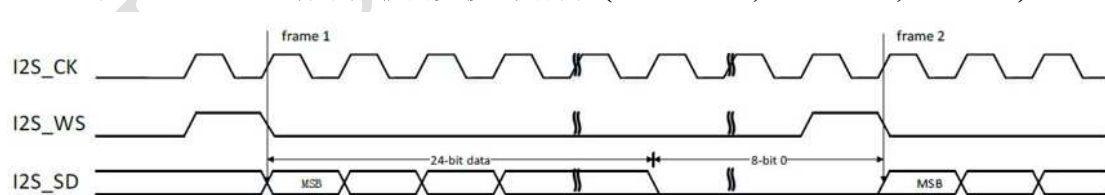


图 23-27 PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

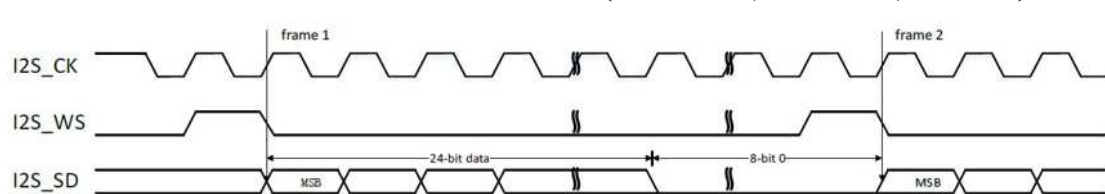


图 23-28 PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

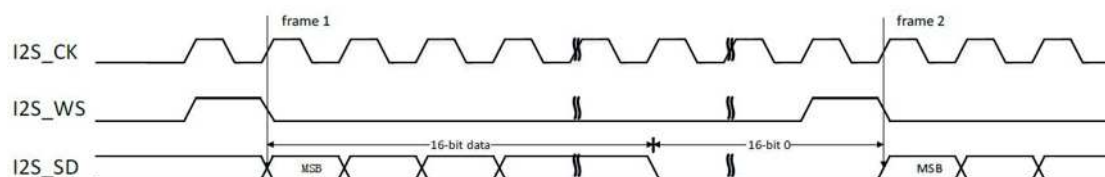
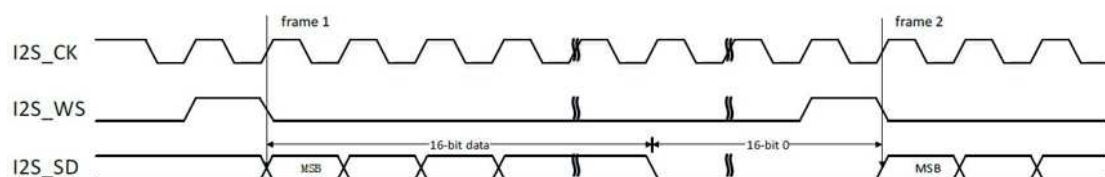


图 23-29 PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 23-30 PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

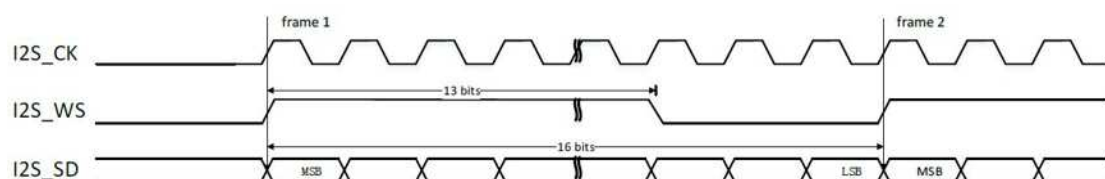


图 23-31 PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

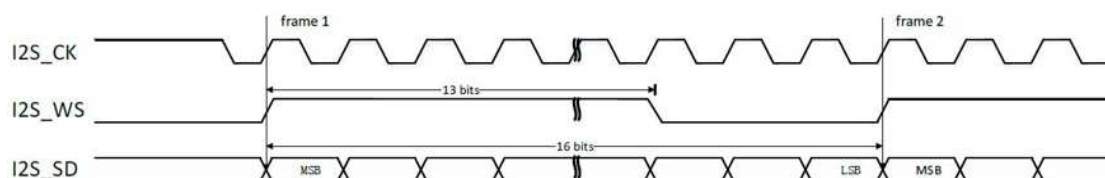


图 23-32 PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

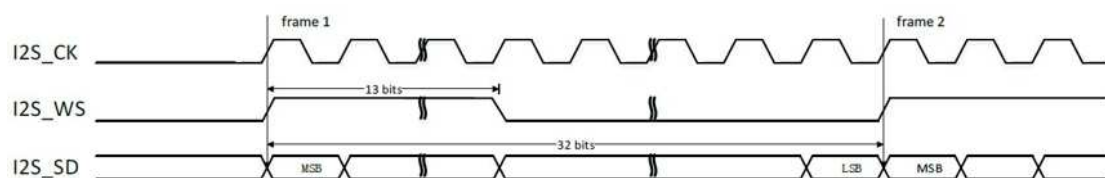


图 23-33 PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

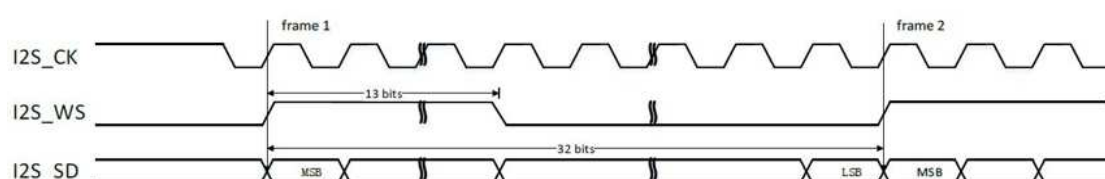


图 23-34 PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

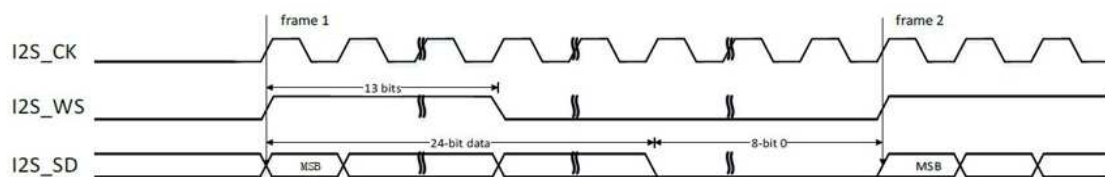


图 23-35 PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

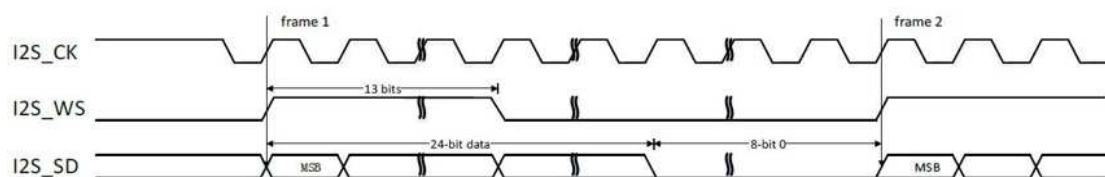


图 23-36 PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

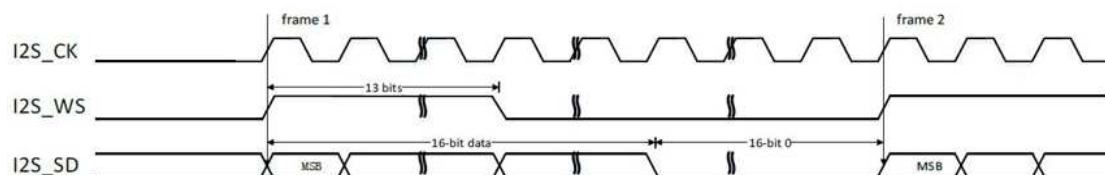
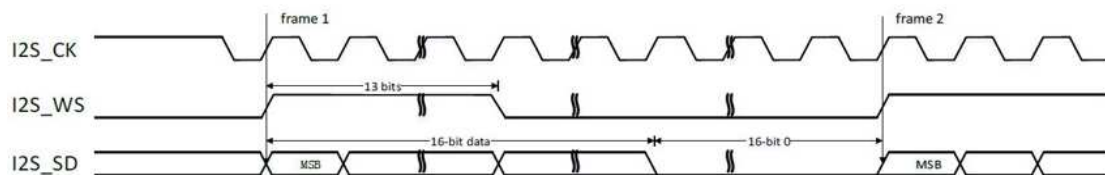


图 23-37 PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



23.6. 时钟

I2S 接口时钟是通过 I2S_PR 寄存器的 DIV 位, OF 位和 MCKOE 位以及 I2S_CR 寄存器的 CHLEN 位来配置的。I2S 的时钟源是系统时钟 SYSCLK。I2S 比特率可以通过下表 I2S 比特率计算公式所示的公式计算。

表 23- 1 I2S 比特率计算公式

MCKOE	CHLEN	公式
0	0	$I2SCLK / (DIV * 2 + OF)$
0	1	$I2SCLK / (DIV * 2 + OF)$
1	0	$I2SCLK / (8 * (DIV * 2 + OF))$

1	1	$I2SCLK / (4 * (DIV * 2 + OF))$

音频采样率(F_s)和 I2S 比特率的关系由如下公式定义:

$$F_s = \text{I2S 比特率} / (\text{通道长度} * \text{通道数})$$

所以, 为了得到期望的音频采样率, 时钟生成器需要按下表音频采样频率计算公式所列的公式进行配置。

表 23- 2 F_s 采样频率计算公式

MCKOE	CHLEN	公式
0	0	$I2SCLK / (32 * (DIV * 2 + OF))$
0	1	$I2SCLK / (64 * (DIV * 2 + OF))$
1	0	$I2SCLK / (256 * (DIV * 2 + OF))$
1	1	$I2SCLK / (256 * (DIV * 2 + OF))$

注: I2SCLK 为系统时钟 SYSCLK。

23.7. 使用流程

I2S 初始化过程包括以下五个步骤。如果要初始化 I2S 工作在主机模式, 五个步骤都要执行, 如果要初始化 I2S 工作在从机模式, 只需要执行步骤 2、3、4、5。

- 步骤 1: 配置 I2S_PR 寄存器的 DIV[7:0]位, OF 位和 MCKOE 位, 定义 I2S 的比特率和选择是否需要提供 I2S_MCK 信号。
- 步骤 2: 配置 I2S_CR 寄存器的 CKPL 位, 定义空闲状态的时钟极性。
- 步骤 3: 配置 I2S_CR 寄存器的 I2SSTD[1:0] 位, PCMSMOD 位, I2SOPMOD[1:0] 位, DTLEN[1:0]位和 CHLEN 位, 定义 I2S 的特性。
- 步骤 4: 配置 I2S_DIER 寄存器的 TBEIE 位, RBNEIE 位, ERRIE 位, TXDMATEN 位和 RXRXDMAREN 位, 选择中断源和 DMA 功能。此步骤可选。
- 步骤 5: 将 I2S_CR 寄存器的 I2SE 位置 1, 来启动 I2S。

23.7.1. 主机发送

TBE 标志位被用来控制发送流程。如前文所述, TBE 标志位表示发送缓冲区空, 此时, 如果 I2S_IE 寄存器的 TBEIE 位为 1, 将产生中断。首先, 发送缓冲区为空(TBE 为

1), 且移位寄存器中没有发送序列。当 32 位数据被写入 I2S_DR 寄存器时(TBE 变为 0), 数据立即从发送缓冲区装载到移位寄存器中(TBE 变为 1)。此时, 发送序列开始。

数据是并行地装载到 32 位移位寄存器中的, 然后串行地从 I2S_SD 引脚发出(高位先发)。下一个数据应该在 TBE 为 1 时写入 I2S_DR 寄存器。数据写入 I2S_DR 寄存器之后, TBE 变为 0。当前发送序列结束时, 发送缓冲区的数据会自动装载到移位寄存器中, 然后 TBE 标志变回 1。为了保证连续的音频数据发送, 下一个将要发送的数据必须在当前发送序列结束之前写入 I2S_DR 寄存器。

对于除 PCM 标准外的所有标准, I2SCH 标志用来区别当前传输数据所属的通道。I2SCH 标志在每次 TBE 标志由 0 变 1 的时候更新。刚开始 I2SCH 标志为 0, 表示左通道的数据应该被写入 I2S_DR 寄存器。

为了关闭 I2S, I2SE 位必须在 TBE 标志为 1 且 TRANS 标志为 0 之后清零。

23.7.2. 主机接收

RBNE 标志被用来控制接收序列。如前文所述, RBNE 标志表示接收缓冲区非空, 如果 I2S_IE 寄存器的 RBNEIE 位为 1, 将产生中断。当 I2S_CR 寄存器的 I2SE 位被置 1 时, 接收流程立即开始。首先, 接收缓冲区为空(RBNE 为 0)。当一个接收流程结束时, 接收到的数据将从移位寄存器装载到接收缓冲区(RBNE 变为 1)。当 RBNE 为 1 时, 用户应该将数据从 I2S_DR 寄存器中读走。读操作完成后, RBNE 变为 0。必须在下一次接收结束之前读走 I2S_DR 寄存器中的数据。

对于除 PCM 之外的所有标准来说, I2SCH 标志用来区分当前传输数据所属的通道。I2SCH 标志在每次 RBNE 标志由 0 变 1 时更新。

23.7.3. 从机发送

从机发送流程和主机发送流程相似, 不同之处如下:

在从机模式下, 从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且 I2S_WS 信号请求传输数据时, 发送流程开始。数据需要在外部主机发起通讯之前写入 I2S_DR 寄存器。为了确保音频数据的连续传输, 必须在当前发送序列结束之前将下一个待发送的数据写入 I2S_DR 寄存器, 否则会产生发送欠载错误。此时 TXURERR 标志会置 1, 如果 I2S_IE 寄存器的 ERRIE 位为 1, 将会产生中断。这种情况下, 必须先

关闭 I2S 再打开 I2S 来恢复通讯。从机模式下, I2SCH 标志是根据外部主机发送的 I2S_WS 信号而变化的。

为关闭 I2S, 必须在 TBE 标志变为 1 且 TRANS 标志变为 0 之后, 才能清除 I2SE 位。

23.7.4. 从机接收

从机接收流程与主机接收流程类似, 不同之处如下。

在从机模式下, 从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且 I2S_WS 信号指示数据开始时, 接收流程开始。从机模式下, I2SCH 标志是根据外部主机发送的 I2S_WS 信号而变化的。

为了关闭 I2S, 必须在收到最后一个 RBNE 之后立即清除 I2SE 位。

24. 低功耗串口（LPUART）

24.1. 概述

低功耗 UART(LPUART)是一个低功耗的 UART 模块，通讯可以使用独立的时钟源。只需要使用 32.768KHz 的时钟就可以使用 9600 波特率通讯。更高的波特率可以通过外部选择更快的时钟源实现。

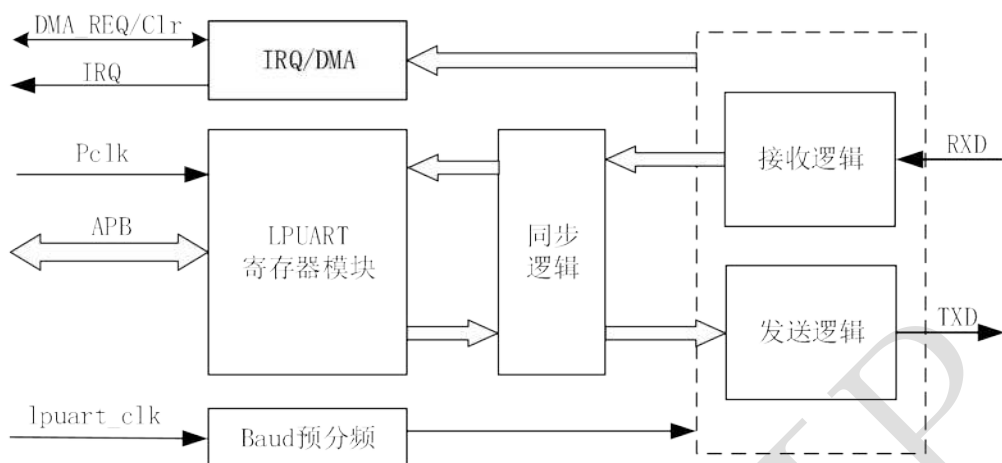
24.2. 主要特性

- 寄存器和通讯时钟独立
- 32.768KHz 时钟下，最大支持 9600 波特率
- 可使用 PCLK 的分频时钟作为工作时钟
- 7-8 位数据位可配置
- 奇/偶校验、0/1 校验或者无校验，可配置
- 1-2 停止位可配置
- STOP 模式下唤醒系统：起始位、收到 1 字节或者收到字节匹配
- 支持 DMA 模式

24.3. 功能描述

LPUART 主要分为寄存器接口、接收逻辑、发送逻辑、中断/唤醒/DMA 逻辑等组成。结构框图如下：

图 24-1 LPUART 结构框图



24.3.1. 时钟选择

LPUART 一共支持三种时钟，分别为：

- RC32K
- XTL
- PCLK 分频，分频系数由 LPUARTDIV 确定

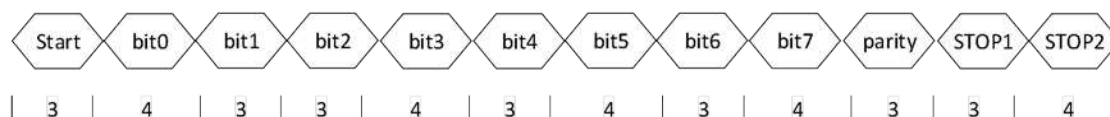
通过 RCC_CCR2 寄存器进行设置，当选择 RC32K 或 XTL 为时钟源时，最大波特率支持 9600，LPUART 提供低功耗支持。当选择 PCLK 分频作为时钟源，支持更高波特率，LPUART 提供高性能支持。

24.3.2. 波特率设置

波特率设置分为整数部分和小数部分，其中整数部分至少为 3 分频（设置值为 2），设置整数分频；小数部分按照比特设置从 Start 到 STOP 分析每个比特是否增加 1。由于 LPUART 工作在 32.768K 频率下时，通讯时钟不是 9600 的整数倍，故必须采用这种小数分频的方式，如采用 3-4 分频循环的方式实现。

下图为 32.768K 频率下，使用 9600 波特率，8 位数据，校验位，和 2 个停止位的示意图。波特率整数部分配置为 3 分频（每个比特 3-4 个时钟，设置值为 2），接收采样设为第 2 个时钟（设置值为 1）；小数寄存器配置位 12'b1001_0101_0010。其中接收采样在每个 bit 的中间时钟。

图 24-2 LPUART BIT 分布图



波特率小数部分的加权平均加上整数部分，应该和工作时钟除以波特率相等；并且从一个字节看小数的 0-1 分布应该均匀，比如要设置为 4'b0101,而不是 4'b0011。上面的例子中，工作时钟除以波特率为 $32768/9600=3.413$ ，波特率加权平均值为 3.416。

如果外部选择更高的时钟，则 LPUART 可以实现更快的波特率。

- 波特率计算公式为
 - 波特率整数部分=（工作频率/需要波特率）的整数部分
 - 小数部分相求加权平均=工作频率/需要波特率-波特率整数部分
- 如要设为 115200，系统时钟为 64M，经过 32 分频产生 LPUART 工作时钟，则：
 - 波特率应为 $2000000/115200=17.361$
 - 整数部分可以设置为 17 分频（寄存器值 16）
 - 小数加权平均为 0.361,如采用 Start+8 位数据+校验+1 位 STOP 共 11 位，则小数部分在 11 比特中需要 $11*0.361\approx 4$ 个 1，可以设置为 12'b100100100100。

24.3.3. STOP 模式唤醒

在 STOP 模式系统时钟（PCLK）停止情况下，LPUART 收发逻辑可以独立工作在 32.768K 时钟下，此时可以选择接收唤醒方式，通过设置 RXWKS[6:5]选择三种唤醒中的一种：

- START 位检测唤醒：当接收到 START 位时唤醒；
- 1byte 数据接收完成：当接收到 1byte 的字节时唤醒；
- 接收数据匹配成功：若选择此唤醒模式，则还需要设置地址寄存器 LPUART_ADDR，写入匹配地址，当接收到 1byte 字节数据且数据等于 LPUART_ADDR 中的匹配地址时唤醒，即使采用地址匹配方式，收到的数据也会放入数据寄存器。

另外，通过设置 WKCK 位，可以选择接收完 1 字节，是否检查校验位和 STOP 位，再触发唤醒/中断

24.3.4. DMA 请求

LPUART 可以利用 DMA 连续通信,通过设置线控制寄存器 LPUART_CR 的 DMAEN 位,使能 DMA 功能,利用 DMA 可以实现 LPUART 的发送/接收。

当传输完成 DMA 控制器指定的数据量时, DMA 控制器会产生完成中断。

24.4. 寄存器描述

LPUART 寄存器基地址:

LPUART1: 0x4000_8000

偏置	名称	描述
0x00	LPUART_RXDR	接收数据寄存器
0x04	LPUART_TXDR	发送数据寄存器
0x08	LPUART_LCR	线控制寄存器
0x0C	LPUART_CR	控制寄存器
0x10	LPUART_IBAUD	波特率整数部分
0x14	LPUART_FBAUD	波特率整数部分
0x18	LPUART_IE	中断使能寄存器
0x1C	LPUART_SR	状态寄存器
0x20	LPUART_ADDR	地址寄存器

24.4.1. 接收数据寄存器/LPUART_RXDR （偏移：00h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	RXDATA	RO	0x00	接收的数据

24.4.2. 发送数据寄存器/LPUART_TXDR （偏移：04h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	TXDATA	WO	0x00	发送的数据

24.4.3. 线控寄存器/LPUART_LCR （偏移：08h）

比特	名称	属性	复位值	描述
15:10	RSV	-	-	保留
9	TXPOL	RW	0	发送数据极性是否取反 0: 不取反 1: 取反
8	RXPOL	RW	0	接收数据极性是否取反 0: 不取反 1: 取反
7	WKCK	RW	1	接收唤醒校验选择 0: 接收完 1 字节，不检查校验位和 STOP 位，直接触发唤醒/中断 1: 接收完 1 字节，检查校验位和 STOP 位都正确，才触发唤醒/中断
6:5	RXWKS	RW	00	接收唤醒选择，STOP 模式用于唤醒选择 00: START 位检测唤醒 01: 1byte 数据接收完成 10: 接收数据匹配成功 11: 无唤醒 STOP 模式下用于唤醒
4	WLEN	RW	0	字宽选择位 0: 8bits

				1: 7bits
3	STP2	RW	0	停止位数选择位: 0: 1 位停止位 1: 2 位停止位
2	EPS	RW	0	0/1 校验或者奇/偶校验选择位（取决于 SPS） 0: 奇/偶校验选择奇校验，或 0/1 校验选择校验位强制为 1 1: 奇/偶校验选择偶校验，或 0/1 校验选择校验位强制为 0
1	SPS	RW	0	校验模式选择位 0: 奇/偶校验 1: 0/1 校验
0	PEN	RW	0	校验使能位: 0: 禁止奇/偶校验或 0/1 校验 1: 使能奇/偶校验或 0/1 校验

24.4.4. 控制寄存器/LPUART_CR（偏移：0Ch）

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
2	DMAEN	RW	0	使能 DMA 功能
1	TXE	RW	0	发送使能位 0: 禁止 1: 使能
0	RXE	RW	0	接收使能位: 0: 禁止 1: 使能

24.4.5. 波特率整数部分/LPUART_IBAUD（偏移：10h）

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
15:8	RXSAM	RW	0x01	接收采样点设置，一般设为 IBAUD>>1，可以略微调整。
7:0	IBAUD	RW	0x02	波特率分频整数因子-1，设置范围为 2~254 $\text{IBAUD} = (\text{integer}(\text{F}_{\text{PCLK}}/\text{BAUD})) - 1$ integer 为取整操作 最小值为 2，即 3 分频；最大值为 254，即 256 分频。 结合小数部分，波特率最大为 3~4 分频之间，最小值为 255~256 之间。

24.4.6. 波特率小数部分/LPUART_FBAUD（偏移：14h）

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	FBAUD	RW	0x000	波特率小数部分，设置 1 字节内的每一个 bit 是否调整。从低位到高位对应为 Start 到 STOP。 0：不调整 1：增加一个时钟。 这个比较麻烦，需要增加一张常用波特率表 小数部分的加权平均为 $(\text{F}_{\text{PCLK}}/\text{BAUD}) - 1 - \text{IBAUD}$

24.4.7. 中断使能寄存器/LPUART_IE（偏移：18h）

比特	名称	属性	复位值	描述

15:11	RSV	-	-	保留
9	STARTIE	RW	0	起始位检测中断使能位 0: 禁止 1: 使能
8	MATCHIE	RW	0	地址匹配中断使能位 0: 禁止 1: 使能
7:6	RSV	-	-	保留
5	RXOVIE	RW	0	接收 overrun 中断使能位 0: 禁止 1: 使能
4	FEIE	RW	0	接收帧格式错误(STOP 位)中断使能位 0: 禁止 1: 使能
3	PEIE	RW	0	接收校验错误中断使能位 0: 禁止 1: 使能
2	TXEIE	RW	0	发送 buffer 空中断使能 0: 禁止 1: 使能
1	TCIE	RW	0	数据发送完成中断使能位 0: 禁止 1: 使能
0	RXIE	RW	0	字节接收完成中断使能位 0: 禁止 1: 使能

24.4.8. 状态寄存器 LPUART_SR (偏移: 1Ch)

比特	名称	属性	复位值	描述
15:11	RSV	-	-	保留
10	TXE	RO	1	发送缓冲空标志
9	STARTIF	RC_W1	0	起始位检测中断标志, 写 1 清零
8	MATCHIF	RC_W1	0	地址匹配中断标志, 表示接收缓冲区内的数据与地址寄存器相同, 写 1 清零 可以设置 WKCK 位选择是否检查校验和 STOP 位
7	TXOVF	RC_W1	0	TXDR 溢出错误, 软件写 1 清零 当 TXDR 满时, 软件又向 TXDR 写入新数据, 该位置 1
6	RXF	RO	0	接收缓冲满, 读 RXDR 清零, 不产生中断 注: 不管校验是否正确
5	RXOVIF	RC_W1	0	接收 overrun 中断标志, 写 1 清零 0: 无中断 1: 有中断 注: 不管校验是否正确
4	FEIF	RC_W1	0	帧格式错误 (STOP) 中断标志, 写 1 清零 0: 无中断 1: 有中断
3	PEIF	RC_W1	0	校验错误中断标志, 写 1 清零 0: 无中断 1: 有中断
2	TXEIF	RO	1	发送 buffer 空中断标志, 写入 TXDR 后清零 0: 无中断产生 1: 发送 buffer 空
1	TCIF	RC_W1	0	数据发送完成中断标志, TXDR 为空并且当前字

				节发送完成。写 1 清 0: 无中断 1: 有中断
0	RXIF	RC_W1	0	字节接收完成中断标志，写 1 或读取 RXDR 时 清零 1: 接收完一帧数据后中断产生 0: 无中断产生

24.4.9. 地址寄存器 LPUART_ADDR （偏移：20h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	ADDR	RW	0x00	匹配地址设置

24.5. 使用流程

24.5.1. 串口收发

1. 配置波特率寄存器（FBAUD/IBAUD）
2. 配置线控制寄存器(LCR),选择校验、位宽和 STOP 位数
3. 配置中断使能寄存器（IE），使能收发中断等
4. 设置控制寄存器（CR），使能 RXE/TXE
5. 等待中断或者查询标志

24.5.2. 配置 STOP 唤醒

1. 配置波特率寄存器（FBAUD/IBAUD）
2. 配置线控制寄存器(LCR),选择校验、位宽和 STOP 位数
3. 配置线控寄存器 RXWKS，选择唤醒方式 START、接收 1 字节或者接收 1 字节并且地址匹配

4. 配置线控寄存器 WKCK，选择接收数据是否检查校验位和 STOP
5. 设置控制寄存器（CR），使能 RXE

ALSTINOCCHIP

25. 模数转换器（ADC）

25.1. 概述

本部分包含 2 个 12 位逐次逼近型 ADC。

ADC1 与 ADC2 紧密耦合，可在双重模式下运行（ADC1 为主器件，ADC2 为从器件）。ADC1 支持 20 个通道，其中 17 个为外部通道，IN17-IN19 三个通道为内部通道。ADC2 支持 20 个通道，其中 18 个为外部通道，IN16-IN17 两个通道为内部通道。

ADC 支持最大 16 次可设通道的规则转换，以及 4 次通道可设的注入转换。每次 A/D 转换可以按照单次、连续或间断模式执行。

ADC 具有模拟来看门狗特性，允许应用程序检测输入电压是否超出用户定义的高/低阈值。

ADC 映射到 AHB 总线，从而可实现快速数据处理。

内置硬件过采样器，能减轻 CPU 进行相关计算的负担。

转换逻辑 ADC_CLK 由内部分频产生，ADC_CLK 不得超过 48MHz

25.2. 主要特性

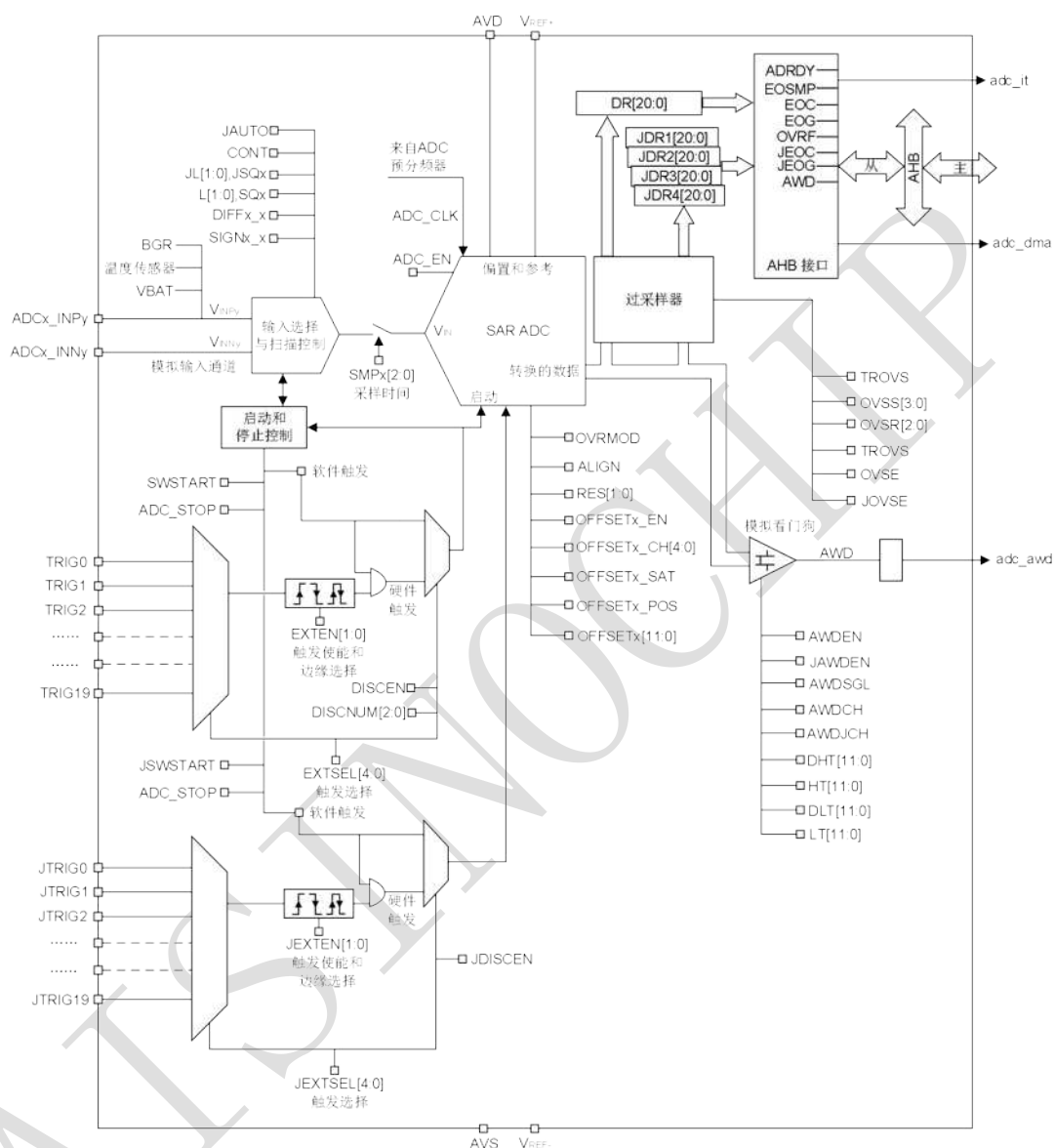
- 12 位分辨率，可配置 10 位、8 位或 6 位分辨率
- 转换速率最高可达 3Msps
- 20 个可用通道，包括外部信号源和内部信号源
 - 内建 BGR 连接到 ADC1
 - 温度传感器连接到 ADC1
 - VBAT 连接到 ADC1
 - The OPA1/2 内部输出连接到 ADC1
 - The OPA2/3 内部输出连接到 ADC2
- 支持单端信号转换和差分信号转换
- 规则转换结束、注入转换结束和发生模拟看门狗事件时产生中断
- 支持单次、连续转换模式
- 支持间断转换模式

- 支持双重模式（两个 ADC 设备）
- 双 ADC 支持交替触发模式
- 最多支持 16 个规则通道组和 4 个注入通道组
- 采样时间可以按通道分别编程
- 规则转换和注入转换均有外部触发选项
- 规则通道转换期间有 DMA 请求产生
- AHB 总线便于系统集成，同时实现高速的读写操作
- 支持过采样：16 位数据寄存器，过采样率支持 2~256 倍
- 数据寄存器可配置数据对齐方式

ALISINOCHIP

25.3. 结构框图

图 25-1 ADC 框图



25.4. 功能描述

25.4.1. 通道选择

一共有 20 个 ADC 通道(通道 0-通道 19), 使用时可以把转换组织成两组: 规则组和注入组。在任意多个通道上以任意顺序进行的一系列转换构成成组转换。例如, 可以如下顺序完成转换: 通道 3、通道 8、通道 2、通道 2、通道 0、通道 2、通道 2、通道 15。

注：通道 0 对应模拟输入 VIN0（ADC_IN0），通道 15 对应模拟输入 VIN15（ADC_IN15）。

- 规则组由多达 16 个转换组成。规则通道和它们的转换顺序在 ADC_SQRx 寄存器中选择。规则组中转换的总数应写入 ADC_SQR1 寄存器的 L[3:0]位中。
- 注入组最多支持 4 个转换组成。注入通道在 ADC_JSQR 寄存器中选择，注入组的转换总数写入 ADC_JSQR 的 JL 位。

如果 ADC_SQRx 或 ADC_JSQR 寄存器在转换期间被更改，当前的转换继续完成，随后序列以新组继续进行转换。

表 25-1 ADC 模拟通道选择

	ADC1	ADC2
IN0	PA0	PA0
IN1	PA1	PA1
IN2	PA2/OPA1_OUT	PA2/OPA1_OUT
IN3	PA3	PA3
IN4	PA4/DAC_OUT1	PA4/DAC_OUT1
IN5	PA5/DAC_OUT2	PA5/DAC_OUT2
IN6	PA6/OPA2_OUT	PA6/OPA2_OUT
IN7	PA7	PA7
IN8	PB0	PB0
IN9	PB1/OPA3_OUT	PB1/OPA3_OUT
IN10	PC0	PC0
IN11	PC1	PC1
IN12	PC2	PC2
IN13	PC3	PC3

IN14	PC4	PC4
IN15	PC5	PC5
IN16	TempSensor/VBAT(PB12)	OPA2_OUT_INT
IN17	VBG1P2(ANA_TEST)	OPA3_OUT_INT
IN18	OPA2_OUT_INT	PE7
IN19	OPA1_OUT_INT	PE8

25.4.2. 单次转换模式

在规则组内，单次转换模式下，ADC 模块只执行一组转换。该模式既可通过设置软件触发也可通过外部触发启动，此时 CONT 位为 0。

每次转换结束：

- 每次规则通道转换结束：
 - 转换数据被储存在 16 位 ADC_DR 寄存器中
 - EOC(转换结束)标志被设置
 - 如果设置了 EOCIE，则产生 EOC 中断。
 - 如果所有规则通道（由 ADC_SQR1.L 决定的转换长度）转换结束，EOG(规则组转换结束)标志被置位
 - 如果设置了 EOGIE，则产生 EOG 中断，ADC 停止。
- 每次注入通道转换结束：
 - 转换数据被存储在 16 位 ADC_JDRx 寄存器
 - JOEC(转换结束)标志被设置
 - 如果设置了 JEOCIE，则产生 JEOP 中断
 - 如果所有注入通道（由 ADC_JQR.JL 决定的转换长度）转换结束，JEOG(注入组转换结束)标志被置位
 - 如果设置了 JEOGIE，则产生 JEOG 中断

如果要只转换一个通道，规则组或注入组的长度设置为 1。

25.4.3. 连续转换模式

连续转换只对规则组有效，连续转换模式中，当前面 ADC 转换一结束马上就启动另一次转换。此模式可通过外部触发启动或软件触发位启动，此时 CONT 位是 1。

- 每次转换结束：
 - 转换数据被储存在 16 位的 ADC_DR 寄存器中
 - EOC(转换结束)标志被设置
 - 如果设置了 EOCIE，则产生中断
 - 如果所有规则通道（由 ADC_SQR1.L 决定的转换长度）转换结束，EOG(规则组转换结束)标志被置位
 - 如果设置了 EOGIE，则产生 EOG 中断。

注入通道组不支持连续转换模式。只有在设置自动注入模式（JAUTO=1），在规则组完成后，注入通道自动开始。

25.4.4. 间断模式

对于规则组，此模式通过设置 ADC_CR1 的 DISCEN 位激活。它可以用来在规则组的转换中执行一个短序列的 $n(n \leq 8)$ 次转换，此转换是 ADC_SQRx 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CR1 寄存器的 DISCNUM[2:0] 位给出。

一个触发信号可以启动 ADC_SQRx 寄存器中描述的下轮 n 次转换，直到此序列所有的转换完成为止。总的序列长度由 ADC_SQR1 寄存器的 L[3:0] 定义。

举例：

$n=3$ ，被转换的通道=0,1,2,3,6,7,9,10

第一次触发：转换的序列为 0,1,2

第二次触发：转换的序列为 3,6,7

第三次触发：转换的序列为 9,10

第四次触发：转换的序列为 0,1,2

注意：当以间断模式转换一个规则组时，转换序列结束后不自动从头开始。当所有子组被转换完成，下一次触发启动第一个子组的转换。在上述示例中，第 4 次触发重新转换了第 1 个子组中的 0,1,2，而不是在第 3 次触发中转换通道 0。

每次转换结束均产生 EOC 事件，一个规则组转换结束后产生 EOG 事件。当 ADC_CR1 寄存器的 DISCEN 位和 CONT 位同时使能时，即间断模式和连续转换模式同时使能，规则通道以连续模式进行转换。

对于注入组，可将 ADC_CR1 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部触发事件之后，可使用该模式逐通道转换在 ADC_JSQR 寄存器中选择的序列，每次触发只进行一次转换。相当于规则组间断模式中的 $n=1$ 。

出现外部触发时，将启动在 ADC_JSQR 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADC_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

$n = 1$ ，要转换的通道 = 1、2、3

第 1 次触发：转换通道 1

第 2 次触发：转换通道 2

第 3 次触发：转换通道 3

第 4 次触发：转换通道 1

注意：转换完所有注入通道后，下一个触发信号将启动第一个注入通道的转换。在上述示例中，第 4 次触发重新转换了第 1 个注入通道。在每次转换结束均产生 JEOP 事件，一个规则组转换结束后产生 JOEG 事件。

不能同时使用自动注入和不连续采样模式。

25.4.5. 注入通道管理

● 触发注入

要使用触发注入，必须将 ADC_CR1 寄存器中的 JAUTO 位清零。如果在规则通道组转换期间出现外部注入触发或者 JSWSTART 位置 1，则当前的转换会复位，并且注入通道序列会切换为单次扫描模式。然后，规则通道组的规则转换会从上上次中断的规则转换处恢复。如果在注入转换期间出现规则事件，注入转换不会中断，但在注入序列结束时执行规则序列。

● 自动注入

如果将 JAUTO 位置 1，则注入组中的通道会在规则组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADC_SQRx 和 ADC_JSQR 寄存

器中编程。在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则在转换规则通道之后会继续转换注入通道。

在 JAUTO 为 1 时，需要禁止外部注入触发，并保证 JSWSTART 为 0。

不能同时使用自动注入和间断采样模式。

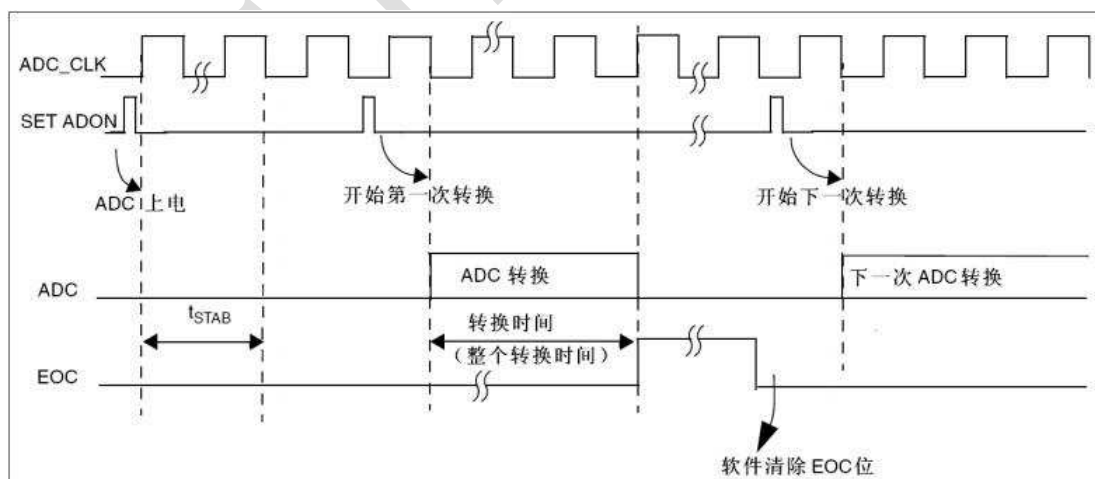
25.4.6. 停止控制

当 ADC_CR2 寄存器的 ADC_STP 置位后，ADC 控制器将在当前通道转换结束后停止 ADC 转换，并且停止后硬件会自动清除 ADC_STP 位，此时可以等待新的触发事件。当新的触发事件发生时，规则通道的转换序列将从 SQ1 开始启动，如果使能了过采样功能，过采样次数也将从 0 开始计数。

25.4.7. 时序图

如下图所示，ADC 转换完成后，EOC 标志被设置，12 位 ADC 数据寄存器包含转换的结果。

图 25-2 ADC 转换时序图

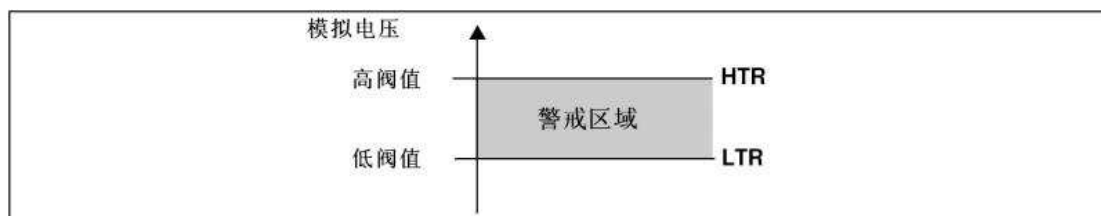


25.4.8. 模拟看门狗

如果被 ADC 转换的模拟电压低于低阈值或高于高阈值，AWD 模拟看门狗状态位被设置。单端输入通道的高低阈值分别由 ADC_HTR 寄存器的 HT 位和 ADC_LTR 寄存器

的 LT 位决定，用无符号数表示；差分输入通道的高低阈值分别由 ADC_HTR 寄存器的高阈值和 ADC_LTR 寄存器的 DLT 位决定，比较时都当做无符号数。通过设置 ADC_IE 寄存器的 AWDIE 位以允许产生相应中断。

图 25-3 AWD 阈值图



通过配置 ADC_CR1 寄存器的 AWDEN 位或 JAWDEN 位，使能模拟看门狗功能，AWDSGL 位控制模拟看门狗作用于匹配 AWDCH/AWDJCH 的规则/注入通道或所有通道，如表所示。

表 25-2 AWD 控制位表

模拟看门狗警戒的通道	ADC_CR1 寄存器控制位		
	AWDSGL 位	AWDEN 位	JAWDEN 位
无	x	0	0
所有规则通道	0	1	0
所有注入通道	0	0	1
所有注入和规则通道	0	1	1
匹配 AWDCH 的规则通道	1	1	0
匹配 AWDJCH 的注入通道	1	0	1
匹配 AWDJCH 的注入或 匹配 AWDCH 的规则通道	1	1	1

对于模拟看门狗使用原始转换结果进行，不使用偏移计算后的数据。

如果转换分辨率小于 12 位（RES 位设置），由于始终对完整的 12 位原始数据比较（左对齐），因此编程阈值的 LSB 必须保持清零。

25.4.9. 数据对齐

ADC_CR2 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。可选择左对齐和右对齐两种方式，如图所示。采用左对齐时，数据基于半字对齐，分辨率设置为 6 位时除外。当分辨率设置为 6 位时，数据基于字节对齐。

注： 过采样模式下不支持左对齐。当 ROVSE 和/或 JOVSE 位置 1 时，忽略 ALIGN 位的值，并且 ADC 仅提供右对齐数据。

如果通道使能 offset 功能，转换数据将减去 ADC_OFRx 寄存器中写入的用户自定义偏移量 OFFSETx[11:0]，因此结果可以是一个负值，使用有符号数。SIGN 位表示扩展的符号值。如果不使能 offset，则转换结果为无符号数。

过采样模式下不支持偏移校准。如果 ROVSE 和/或 JOVSE 位置 1，ADCx_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。

图 25-4 数据右对齐



图 25-5 数据左对齐



25.4.10. 偏移补偿

支持可选的通道数据偏移补偿功能，通过 ADC_OFRx(x 为 1~4)寄存器设置。共有四组偏移补偿可以选择，每组有独立的使能位、补偿值、补偿方式、通道号和补偿结果格式。当对应寄存器 OFFSETx_EN 为 1 时，通过 OFFSETx_CH[4:0]选择需要进行偏移补偿的通道。使能后，此通道的转换结果减去或者加上 OFFSETx[11:0]定义的偏移量（OFFSETx_POS 选择）。结果有可能是一个负数，因此使用有符号数表示，或者强制使用无符号表示（OFFSETx_SAT 选择）。

对于有符号结果(OFFSET_x_SAT=0),数据为 16 位的补码形式;如果 OFFSET_x_POS 为 1,以 12 位数据右对齐为例,则 BIT12 位代表进位,非符号位。

对于无符号结果(OFFSET_x_SAT=1),如果 OFFSET_x_POS 为 1,加上偏移量计算,最大值为 0xffff;减去偏移计算,最小值为 0x000。

如果多个偏移 (OFFSET_x) 指向同一通道,相减时只会考虑使用 x 值最小的偏移。在硬件过采样模式下,偏移补偿功能忽略。

如果 ADC 模拟通道使用有符号形式,则在进行 OFFSET 计算时,仍然认为原始转换结果为无符号数。

对于模拟看门狗使用原始转换结果进行,不使用偏移计算后的数据。

25.4.11. 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,每个通道的采样时间 TSAMP 可以通过 ADC_SMPR_x(x=1, 2, 3)的 SMP_x[3:0]位进行设置。

ADC 转换时间 TCONV 的计算如下: 其中 bit 为 12/10/8/6, 由 RES 设置

$$T_{\text{conv}} = T_{\text{SAMP}} + (\text{bit} + 3 - \text{FASTMOD})T_{\text{ADC\_CLK}}$$

例如: ADC_CLK = 40MHz, FASTMOD = 1, T_{SAMP} = 3T_{ADC_CLK}, T_{CONV} = (3+12+3-1)T_{ADC_CLK} = 17T_{ADC_CLK} = 425ns。

注意: 在 ADC 转换期间,新发生的触发事件将被忽略,直到转换结束,才启动新的转换序列,EOC 事件还是在转换结束后立即产生。

25.4.12. 外部触发转换

转换可以由外部事件或者软件触发(例如定时器捕获,EXTI 线)。EXTSEL[4:0]和 JEXTSEL[4:0]控制位允许应用程序选择可能的事件中的某一个,可以触发规则和注入组的采样。

注意: 当外部触发信号被选为 ADC 规则或注入转换时,外部触发的极性可以通过 EXTEN[1:0]和 JEXTEN[1:0]设置。

通道的触发选择,具体选择信号见下表:

表 25-3 ADC 规则/注入转换触发源选择表

TSEL[4:0]	规则转换触发源	注入转换触发源

00000	TIM1_CC1	TIM1_TRGO
00001	TIM1_CC2	TIM1_CC4
00010	TIM1_CC3	TIM2_TRGO
00011	TIM2_CC2	TIM2_CC1
00100	TIM3_TRGO	TIM3_CC4
00101	TIM4_CC4	TIM4_TRGO
00110	EXTI11	EXTI11
00111	TIM8_TRGO	TIM8_CC4
01000	TIM8_TRGO2	TIM1_TRGO2
01001	TIM1_TRGO	TIM8_TRGO
01010	TIM1_TRGO2	TIM8_TRGO2
01011	TIM2_TRGO	TIM3_CC3
01100	TIM4_TRGO	TIM3_TRGO
01001	TIM6_TRGO	TIM3_CC1
01110	TIM15_TRGO	TIM6_TRGO
01111	TIM3_CC4	TIM15_TRGO
10000	TIM1_CC4	TIM16_CC1
10001	LPTIM_OUT	LPTIM_OUT
10010	TIM7_TRGO	TIM7_TRGO
10011	TIM8_CC4	TIM17_CC1

注：当 EXTEN[1:0]和 JEXTEN[1:0]为 0 时，为软件触发，非 0 时为外部触发

25.4.13. DMA 请求

因为规则通道转换的值储存在一个仅有的数据寄存器中，所以当转换多个规则通道时需要使用 DMA，这可以避免丢失已经存储在 ADC_DR 寄存器中的数据。

只有在规则通道的转换结束时才产生 DMA 请求，并将转换的数据从 ADC_DR 寄存器传输到用户指定的目的地址。

对于双 ADC 模式，主 ADC 的 DMA 请求可以在主从两个 ADC 转换都完成才产生，并结果从 ADC_CDR 读取（DMADUAL=1X）；也可以主从 ADC 的 DMA 独立工作。

对于规则交叉模式，如果同时需要使用注入功能，只能才 ADC_CDR 读取结果，DMA 也只能使用主从合并模式。

25.4.14. 双 ADC 模式

本产品支持双 ADC 模式（ADC1 为主，ADC2 为从）。在双 ADC 模式里，根据 ADC1_CCR 寄存器中 DUALMOD[2:0]位所选的模式，转换的启动可以是 ADC1 主和 ADC2 从的交替触发或同步触发。

在双 ADC 模式下，配置外部事件触发转换时，必须设置为仅主 ADC 触发而禁止从 ADC 触发，以防止出现意外触发而启动不需要的从转换。

可实现以下四种模式：

- 注入同步模式
- 规则同步模式
- 规则交叉模式
- 交替触发模式

也可按以下方式组合使用上述模式：

- 注入同步模式 + 规则同步模式
- 规则同步模式 + 交替触发模式
- 规则交叉模式 + 注入同步模式

注意：在双重 ADC 模式下，可在主 ADC 共用规则数据寄存器 (ADC1_CDR) 中读取转换的数据。

● 注入同步模式

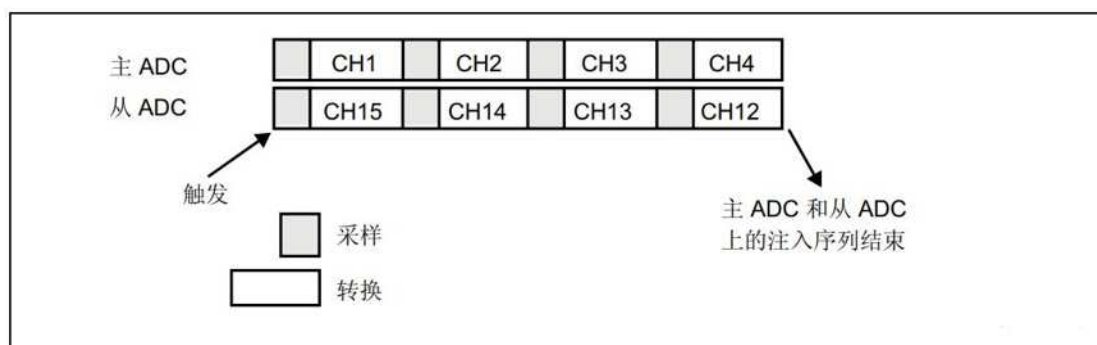
此模式转换一个注入通道组。外部触发来自 ADC1 的注入组多路选择(由 ADC1_JSQR 寄存器的 JEXTSEL[4:0]选择)，它同时给 ADC2 提供同步触发。

注意：不要在 2 个 ADC 上转换相同的通道(两个 ADC 在同一个通道上的采样时间不能重叠)。

在 ADC1 或 ADC2 的转换结束时：

- 转换的数据存储在每个 ADC 接口的 ADC_JDRx 寄存器中。
- 当 ADC1/ADC2 的注入通道全部完成转换后，会生成一个 JEOG 中断（如果已在两个 ADC 接口中的一个接口上使能）。

图 25-6 4 个通道的注入同步模式



在间断模式，每次触发只进行一次转换。

● 规则同步模式

此模式转换一个规则通道组。外部触发来自 ADC1 的规则组多路选择(由 ADC1_CR1 寄存器的 EXTSEL[4:0]选择)，它同时给 ADC2 提供同步触发。

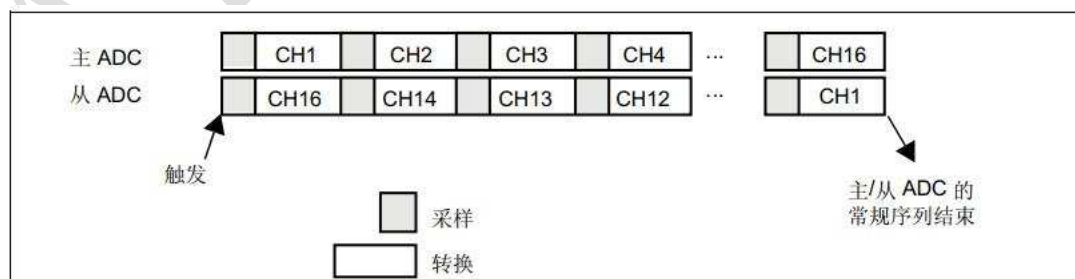
注意：不要在 2 个 ADC 上转换相同的通道(两个 ADC 在同一个通道上的采样时间不能重叠)。

在 ADC1 或 ADC2 的转换结束时：

- 产生一个 32 位 DMA 传输请求(如果设置了 DMA 位)，32 位的 ADC_CDR 寄存器内容传输到 SRAM 中，它上半个字包含 ADC2 的转换数据，低半个字包含 ADC1 的转换数据。
- 当所有 ADC1/ADC2 规则通道都被转换完时，产生 EOG 中断(若任一 ADC 接口开放了中断)。

对于规则同步模式，注入通道处理和单 ADC 模式相同，立刻结束当前规则转换，插入注入通道转换，等待注入通道完成后重新开始被打断的规则通道。

图 25-7 16 个通道的规则同步模式



● 规则交叉模式

此模式只能用于规则组（通常为一个通道）。外部触发源来自 ADC1 的规则组多路选择(由 ADC1_CR1 寄存器的 EXTSEL[4:0]选择)。

出现外部触发之后：

- ADC1 立即启动
- 经过几个 ADC 时钟周期延迟后 ADC2 启动

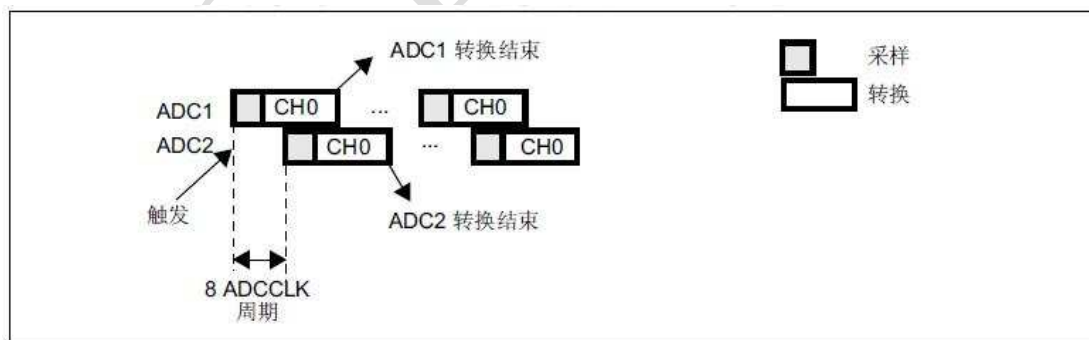
规则交叉模式下 2 个转换之间的最小延迟通过 ADC_CCR 寄存器中的 DELAY 位进行配置。但是，如果某个 ADC 的互补 ADC 仍在对其输入进行采样，则该 ADC 无法启动转换（在给定时间内，只有一个 ADC 能够对输入信号采样）。在这种情况下，延迟时间为采样时间+ 2 个 ADC 时钟周期。例如，如果两个 ADC 的 DELAY = 5 个时钟周期，且采样时间为 15 个时钟周期，则 ADC1 和 ADC2 之间的转换延迟为 17 个时钟周期。

如果 ADC1 和 ADC2 上的 CONT 位均置 1，则这两个 ADC 所选规则通道会连续进行转换。

ADC2 产生一个 EOC 中断后(需使能 EOC 中断 ADC.IE.EOCIE=1)，产生一个 32 位的 DMA 传输请求(如果设置了 DMA 位)，ADC_CDR 寄存器的 32 位数据被传输到 SRAM，上半个字包含 ADC2 的转换数据，低半个字包含 ADC1 的转换数据。

对于规则交叉模式，注入通道处理和单 ADC 模式相同，立刻结束当前规则转换，插入注入通道转换，等待注入通道完成后重新开始被打断的规则通道。在注入功能使能时，规则通道只支持一个通道的连续模式，规则转换从 ADC_CDR 寄存器读取转换结果。

图 25-8 连续转换模式下 1 通道的规则交叉模式



● 交替触发模式

此模式用于注入通道组。外部触发来自 ADC1 的注入组多路选择(由 ADC1_JSQR 寄存器的 JEXTSEL[4:0]选择)。

- 发生第一次触发时，将转换 ADC1 中注入组的所有通道
- 发生第二次触发时，将转换 ADC2 中注入组的所有通道
- 如此循环

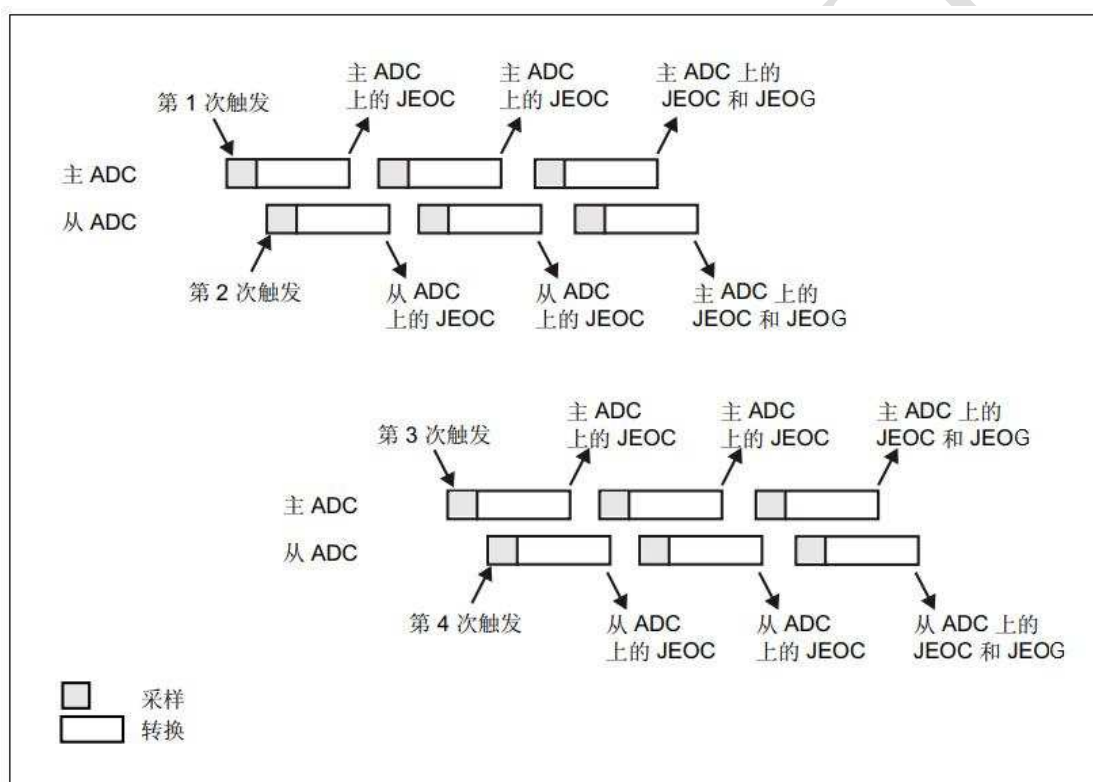
当组中的所有注入 ADC1 通道都转换完成后，会生成一个 JEOG 中断（如果已使能）。

当组中的所有注入 ADC2 通道都转换完成后，会生成一个 JEOG 中断（如果已使能）。

如果使能了 JEOC 中断，每次转换完成都产生一个 JEOC 中断。

如果在组中的所有注入通道都完成转换后出现另一个外部触发，则可通过转换组中的注入 ADC1 通道来重新启动交替触发过程。

图 25-9 交替触发：各个 ADC 的注入组



如果使能 ADC1 和 ADC2 的注入不连续采样模式：

- 发生第一次触发时，将转换第一个注入 ADC1 通道
- 发生第二次触发时，将转换第一个注入 ADC2 通道
- 如此循环

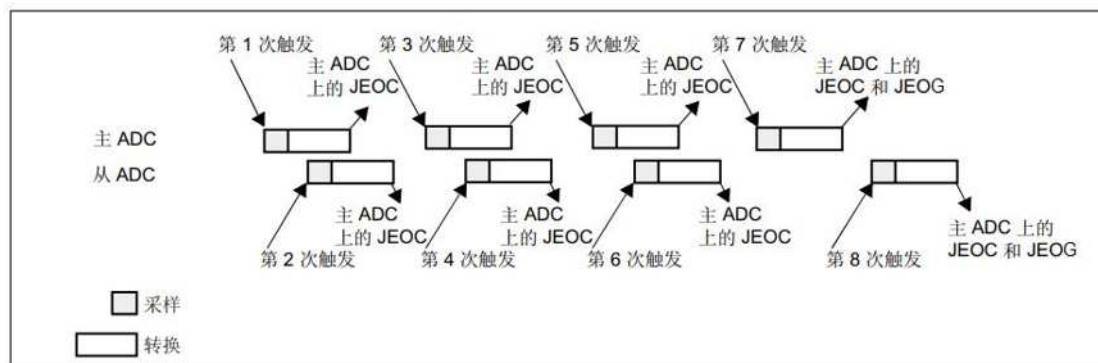
当组中的所有注入 ADC1 通道都转换完成后，会生成一个 JEOG 中断（如果已使能）。

当组中的所有注入 ADC2 通道都转换完成后，会生成一个 JEOG 中断（如果已使能）。

如果使能了 JEOP 中断，每次转换完成都产生一个 JEOP 中断。

如果注入组中的所有通道都完成转换后出现另一个外部触发，则会重新启动交替触发过程。

图 25-10 交替触发：间断模式下的各个 ADC 的注入组



- 独立模式

此模式里，双 ADC 同步不工作，每个 ADC 接口独立工作。

- 混合的规则+注入同步模式

规则组同步转换可以被中断，以启动注入组的同步转换。

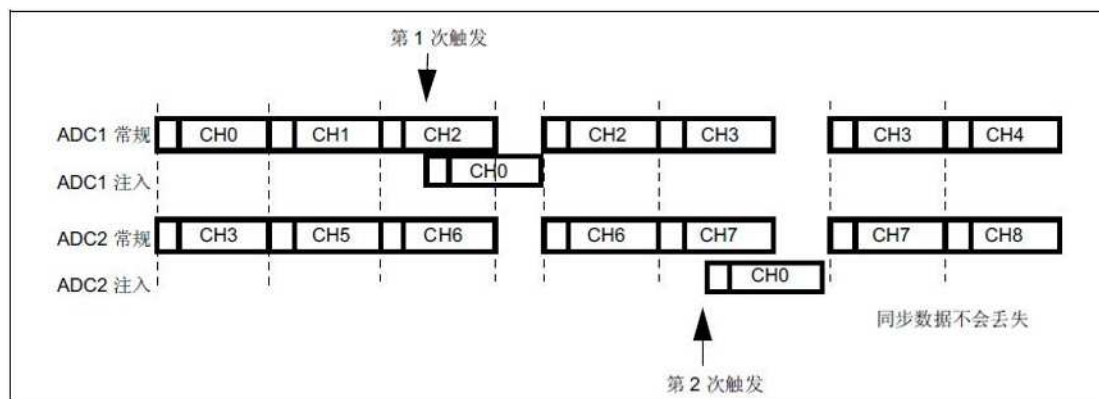
注：在混合的规则/注入同步模式中，必须转换具有相同时间长度的序列，或保证触发的间隔比 2 个序列中较长的序列长，否则当较长序列的转换还未完成时，具有较短序列的 ADC 转换可能会被重启。

- 混合的规则同步+交替触发模式

规则组同步转换可以被中断，以启动注入组交替触发转换。下图显示了一个规则同步转换被交替触发所中断。注入交替转换在注入事件到达后立即启动。如果规则转换已经在运行，为了在注入转换后确保同步，所有的 ADC(主和从)的规则转换被停止，并在注入转换结束时同步恢复。

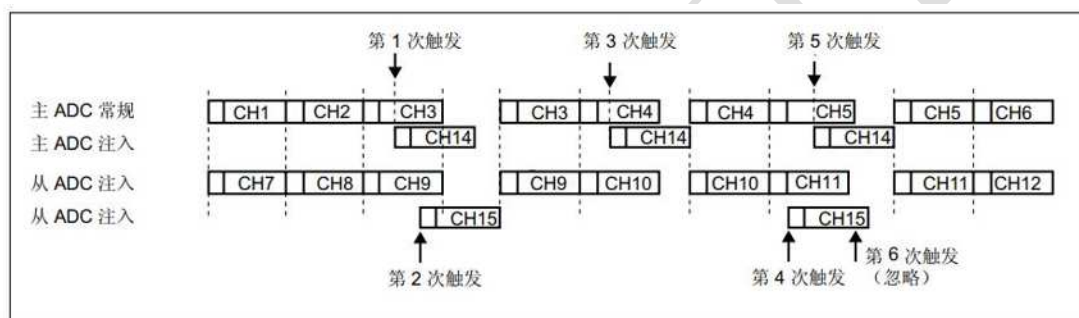
注：在混合的规则同步+交替触发模式中，必须转换具有相同时间长度的序列，或保证触发的间隔比 2 个序列中较长的序列长，否则当较长序列的转换还未完成时，具有较短序列的 ADC 转换可能会被重启。

图 25-11 规则同步+交替触发



在已导致规则转换中断的注入转换期间出现的触发将被忽略。下图说明了这种情况下的行为（第2次触发被忽略）。

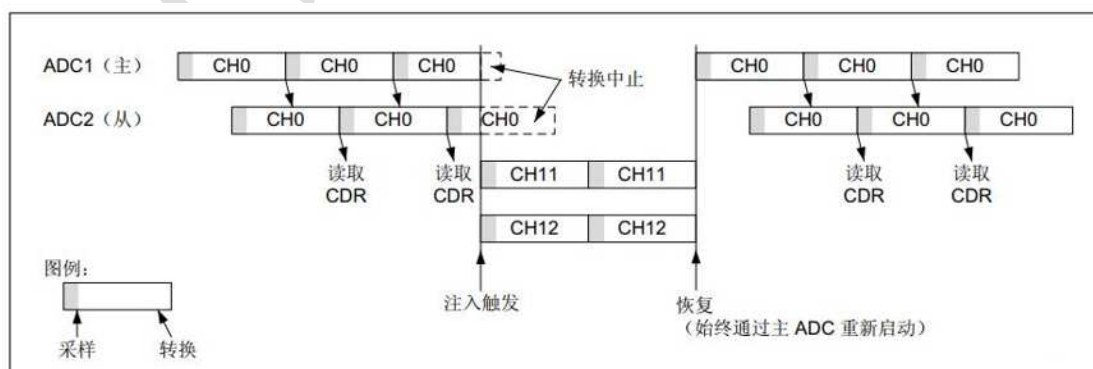
图 25-12 规则同步+交替触发：在注入转换期间出现新的注入触发事件



● 混合的注入同步+ 规则交叉模式

一个注入事件可以中断一个交叉转换。这种情况下，交叉转换被中断，注入转换被启动，在注入序列转换结束时，交叉转换被恢复，并且从主 ADC（ADC1）开始新的转换。下图是这种情况的一个例子。

图 25-13 混合的注入同步+规则交叉模式



25.4.15. 温度传感器

温度传感器可以用来测量器件周围的温度(TA)。温度传感器在内部和 ADC 模拟输入通道 16 相连接，此通道把传感器输出的电压转换成数字值。

温度传感器输出电压随温度线性变化，由于生产过程中的变化，温度变化曲线的偏移在不同芯片内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

25.4.16. 差分信号转换

ADC 比较电路支持差分信号转换，选择差分模式 (DIFFx_x=1) 后，输入差分信号成对比较后再进行 AD 转换。差分信号分为 10 对：VIN0/VIN10、VIN1/VIN11、VIN2/VIN12、VIN3/VIN13、VIN4/VIN14、VIN5/VIN15、VIN6/VIN16、VIN7/VIN17、VIN8/VIN18、VIN9/VIN19。在 ADC_SQRx 或 ADC_JSQR 输入某一个通道号，按照通道号选择单端或差分模式。

对于差分模式，通道号只能选择正级 (VIN0~VIN9) 输入信号。差分模式支持把结果变为有符号数，通过设置 SIGNx_x=1 使能。

注：ADC1 的 VIN17-VIN19 作为差分模式的负极通道已经接了固定的信号(VBAT、VBGA、TS)，不可选择。ADC1 的 IN16 为 Buffer 通道，作为差分信号的输入端，需要输入端的信号是变化缓慢的信号，因此通常不用作差分信号转换。

25.4.17. 溢出控制

当 ADC_DR 寄存器的数据没有及时被软件或 DMA 读取时，将产生溢出事件，ADC_SR 寄存器的 OVERF 被置位。当发生溢出事件时，最新采样数据的处理方式有两种，由 ADC_CR2 寄存器的 OVRMOD 为控制，当 OVRMOD=0 时，最新采样数据被丢弃，ADC_DR 寄存器保留上次的转换数据；当 OVRMOD=1 时，最新采样数据保存到 ADC_DR 寄存器。当 OVERF 状态为高时，停止发送 EOC 事件的 DMA 请求，以保证通过 DMA 方式存放到 RAM 中的转换数据是有效的。

25.4.18. 差分模式和有符号数

对应通道 VIN0~VIN19 支持单端和差分输入两种模式，其他通道只支持单端模式；同时与差分相对应，VIN0~VIN19 支持有符号和无符号两种输出。

单端模式下(DIFFx_x=0)，输入 VINx 为无符号数，无符号输出(SIGNx_x=0)为 DR[11:0]=VINx；有符号数最高位取反即为无符号数结果。例如 VIN0=0x004，配置 DIFF0_10=0/ SIGN0_10=0，则 DR[11:0]=0x004；配置 DIFF0_10=0/ SIGN0_10=1，则 DR[11:0]=0x804。

差分模式下(DIFFx_x=1)，输入 VINx_P 和 VINx_N 为无符号数，当 VINx_P>VINx_N，对应有符号输出(SIGNx_x=1)为 DR[11:0]=(VINx_P-VINx_N)/2，有符号数最高位取反即为无符号数结果。当 VINx_P<VINx_N，对应有符号输出(SIGNx_x=1)DR[11:0]=(~(VINx_N-VINx_P)/2)+1，有符号数最高位取反即为无符号数结果。一个差分对中序号小的为 VINx_P，序号大的为 VINx_N。例如当 VIN0=0x008, VIN10=0x000，配置 DIFF0_10=1/ SIGN0_10=0，则 DR[11:0]=(VIN0-VIN10)/2=0x004；在配置 DIFF0_10=1/ SIGN0_10=1 时，把最高位取反 DR=0x804。当 VIN0=0x000, VIN10=0x008，配置 DIFF0_10=1/ SIGN0_10=1，则 DR[11:0]=(~(VIN10-VIN0)/2)+1=0xffc；在配置 DIFF0_10=1/ SIGN0_10=0 时，把最高位取反 DR[11:0]=0x7fc。

25.4.19. 过采样

如果使能了 ADC 的过采样功能，那么每个通道将进行 N 次转换后产生一次 EOC 事件，并对每次的转换结果进行累加，累加器的长度为 20-bit（256 x 12-bit），然后，右移实现除数为 M 的平均，并对右移舍弃的数据做四舍五入处理，截去数据位的高 4 比特，最终保留 16-bit 的有效数据；将结果保存到 ADC_DR 寄存器中。累加平均的表达式如下所示。

$$result = \frac{1}{M} \times \sum_{n=0}^{N-1} Conversion(t_n)$$

过采样率 N 由 ADC_CR2 寄存器的 OVSR[2:0]位决定，表示范围从 2 倍到 256 倍。平均系数 M 由 ADC_CR2 寄存器的 OVSS[3:0]决定，最大右移范围 8-bit。

规则组通道的过采样通过 OVSE 位使能，注入组通过 JOVSE 使能。规则组过采样

的触发模式由 TROVS 控制。当 TROVS=1 时，一次触发进行一次 ADC 转换，此时忽略 DISCEN 位的内容，并将该位视为 1。当 TROVS=0 时，一次触发进行 N 次 ADC 转换。处理过采样数据时，不可使用对齐模式。ALIGN 位会被忽略，且数据始终采用右对齐格式。过采样模式下不支持偏移校准。如果 ROVSE 和/或 JOVSE 位置 1，ADCx_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。

对于分辨率不为 12 位的情况，原始数据格式为 12 位左对齐，LSB 为 0。例如分辨率为 6，则原始数据为 DATA[11:6],000000。过采样时，累加运算低位也始终为 0。

25.4.20. ADC 中断

规则组和注入组转换结束时能产生中断，当模拟看门狗状态位被设置时也能产生中断。它们都有独立的中断使能位。

表 25-4 中断使能和标志

中断事件	事件标志	使能控制位
规则通道转换结束	EOC	EOCIE
规则组转换结束	EOG	EOGIE
注入转换结束	JEOC	JEOCIE
注入组转换结束	JEOG	JEOGIE
模拟看门狗状态	AWD	AWDIE
规则转换数据溢出	OVERF	OVERFIE
规则通道采样完成	EOSMP	EOSMPIE

25.5. 寄存器描述

ADC 寄存器基地址：0x5000_0000

ADC 模块地址划分

偏置	名称
0x000~0x0FC	Master ADC1
0x100~0x1FC	Slave ADC2
0x300~0x3FC	Common Register

偏置	名称	描述
0x00	ADC_SR	ADC 状态寄存器
0x04	ADC_IE	ADC 中断使能寄存器
0x08	ADC_CR1	ADC 控制寄存器 1
0x0C	ADC_CR2	ADC 控制寄存器 2
0x10	ADC_SMPR1	ADC 转换时间寄存器 1
0x14	ADC_SMPR2	ADC 转换时间寄存器 2
0x18	ADC_SMPR3	ADC 转换时间寄存器 3
0x1C	ADC_HTR	ADC 看门狗高阈值寄存器
0x20	ADC_LTR	ADC 看门狗低阈值寄存器
0x24	ADC_SQR1	ADC 规则序列寄存器 1
0x28	ADC_SQR2	ADC 规则序列寄存器 2
0x2C	ADC_SQR3	ADC 规则序列寄存器 3
0x30	ADC_JSQR	ADC 注入序列寄存器
0x34~0x40	ADC_JDRx	ADC 注入数据寄存器
0x48	ADC_DR	ADC 规则数据寄存器
0x4C	ADC_DIFF	ADC 单端/差分选择寄存器
0x50	ADC_SIGN	ADC 符号数选择寄存器
0x60~0x6C	ADC_OFRx	ADC 偏移寄存器

0x80	ADC_ANACFG	ADC 模拟配置寄存器
0x300	ADC_CSR	ADC 共用状态寄存器
0x304	ADC_CCR	ADC 共用控制寄存器
0x308	ADC_CDR	ADC 共用规则数据寄存器
0x30C	ASC_CTSREF	ADC 共用温度传感器/REF 寄存器

25.5.1. ADC 状态寄存器(ADC_SR 偏移 00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	AWD	RC_W1	0	模拟看门狗标志位 该位由硬件在转换的电压值超出了 ADC_LTR 和 ADC_HTR 寄存器定义的范围时设置，由软件写 1 清除 0：没有发生模拟看门狗事件； 1：发生模拟看门狗事件。
6	JEOG	RC_W1	0	注入通道组转换结束位 该位由硬件在一个注入组序列转换结束时设置，由软件写 1 清除 0：一组转换未完成； 1：一组转换完成。
5	JEOC	RC_W1	0	注入通道转换结束位 该位由硬件在注入通道转换结束时设置，由软件写 1 清除或由读取 ADC_JDRx 时清除 0：转换未完成； 1：转换完成
4	OVERF	RC_W1	0	规则通道数据溢出位 该位在规则转换数据溢出时置位，由软件写 1 清除 0：规则转换数据未溢出

				1: 规则转换数据溢出
3	EOG	RC_W1	0	规则通道组转换结束位 该位由硬件在一个规则组序列转换结束时设置，由软件写 1 清除 0: 一组转换未完成； 1: 一组转换完成。
2	EOC	RC_W1	0	规则通道转换结束位 该位由硬件在规则通道（或者一个规则组序列）转换结束时设置，由软件写 1 清除或由读取 ADC_DR 时清除 0: 转换未完成； 1: 转换完成。
1	EOSMP	RC_W1	0	规则通道采样完成标志 该位由硬件在规则通道采样结束时设置，由软件写 1 清除 0: 采样未完成； 1: 采样完成。
0	ADRDY	RO	0	ADC 就绪状态 该位由硬件置位。 0: 复位状态 1: 就绪状态

25.5.2. ADC 中断使能寄存器(ADC_IE 偏移 04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	AWDIE	RW	0	允许产生模拟看门狗中断 该位由软件设置和清除，用于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范围

				的数值时，设置了该位时扫描也不会中止。 0：禁止模拟看门狗中断； 1：允许模拟看门狗中断。
6	JEOGIE	RW	0	允许产生 JEOG 中断 该位由软件设置和清除，用于禁止或允许注入通道组转换结束后产生中断。 0：禁止 JEOG 中断； 1：允许 JEOG 中断。当硬件设置 JEOG 位时产生中断
5	JEOCIE	RW	0	允许产生注入通道转换结束中断 该位由软件设置和清除，用于禁止或允许所有注入通道转换结束后产生中断。 0：禁止 JEOC 中断； 1：允许 JEOC 中断。当硬件设置 JEOC 位时产生中断
4	OVERFIE	RW	0	允许产生规则通道数据溢出中断 该位由软件设置和清除，用于禁止或允许规则通道数据溢出产生中断。 0：溢出中断禁止 1：允许产生溢出中断，规则序列上次结果未取走，又有新的转换结果时，产生中断
3	EOGIE	RW	0	允许产生 EOG 中断 该位由软件设置和清除，用于禁止或允许规则组转换结束后产生中断。 0：禁止 EOG 中断； 1：允许 EOG 中断。当硬件设置 EOG 位时产生中断
2	EOCIE	RW	0	允许产生 EOC 中断 该位由软件设置和清除，用于禁止或允许转换结束后产生中断。 0：禁止 EOC 中断； 1：允许 EOC 中断。当硬件设置 EOC 位时产生中断

1	EOSMPIE	RW	0	允许产生 EOSMP 中断 该位由软件设置和清除，用于禁止或允许 EOSMP 中断。 0：禁止 EOSMP 中断； 1：允许 EOSMP 中断。当硬件设置 EOSMP 位时产生中断
0	RSV	-	-	保留

25.5.3. ADC 控制寄存器 1 (ADC_CR1 偏移 08h)

比特	名称	属性	复位值	描述
31	RSV	-	-	保留
30:28	DISCNUM[2:0]	RW	000	规则通道间断模式通道计数 软件通过这些定义在间断模式下，收到外部触发后转换规则通道的数目。 000：1 个通道 001：2 个通道 ... 111：8 个通道 注：需要小于或者等于 ADC_SQR1 中的 L 长度
27	JDISCEN	RW	0	注入通道上的间断模式使能控制 0：规则通道组禁用间断模式 1：规则通道组使用间断模式
26	DISCEN	RW	0	规则通道上的间断模式使能控制 0：规则通道组禁用间断模式 1：规则通道组使用间断模式
25	JAUTO	RW	0	注入组自动转换 该位由软件设置和清除。如果设置了此位，在规则组转换后使能注入组自动转换。

				0: 禁止注入组自动转换 1: 使能注入组自动转换
24	CONT	RW	0	连续转换 该位由软件设置和清除。如果设置了此位，则转换将连续进行直到该位被清除。 0: 单次转换模式； 1: 连续转换模式。
23	SWSTART	RW	0	开始转换规则通道 由软件设置该位用于启动一组规则通道的转换，在单次转换模式下，当 EOG 标志置位时，或者间断模式下，当 EOC 标志置位时，由硬件清除该位。连续转换模式下，由 ADC_CR2 的 ADC_STP 位，清除该位。 0: 复位状态； 1: 开始转换规则通道。
22	JSWSTART	RW	0	开始转换注入通道 由软件设置该位用于启动一组注入通道的转换，当 JEOG 置位时，由硬件清除该位。在间断模式下，当 JEOC 标志置位时，由硬件清除该位。 0: 复位状态； 1: 开始转换注入通道。
21	RSV	-	-	保留
20:16	EXTSEL[4:0]	RW	00000	选择启动规则通道组转换的外部触发 选择用于启动规则通道组转换的外部触发 00000: TRIG0 00001: TRIG1 00010: TRIG2 ... 10010: TRIG18 10011: TRIG19

				其他： TRIG19
15:14	EXTEN	RW	00	规则通道的外部触发使能 选择外部触发极性和使能规则组的触发 00：禁止触发检测 01：上升沿上的触发检测 10：下降沿上的触发检测 11：上升沿和下降沿上的触发检测
13	DMA	RW	0	直接存储器访问模式 该位由软件设置和清除。详见 DMA 控制器章节。 0：不使用 DMA 模式； 1：使用 DMA 模式。
12	AWDEN	RW	0	在规则通道上开启模拟看门狗 该位由软件设置和清除。 0：在规则通道上禁用模拟看门狗； 1：在规则通道上使用模拟看门狗。
11	JAWDEN	RW	0	在注入通道上开启模拟看门狗 该位由软件设置和清除。 0：在注入通道上禁用模拟看门狗； 1：在注入通道上使用模拟看门狗。
10	AWDSGL	RW	0	在一个单一的通道上使用看门狗 该位由软件设置和清除，用于开启或关闭由 AWDCH/AWDJCH 位指定的通道上的模拟看门狗功能 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。
9:5	AWDJCH[4:0]	RW	00000	注入通道模拟看门狗通道选择位 这些位由软件设置和清除，用于选择模拟看门狗保护的注入输入通道。 定义参见 AWDCH

4:0	AWDCH[4:0]	RW	00000	模拟看门狗规则通道选择位 这些位由软件设置和清除，用于选择模拟看门狗保护的规则输入通道。 00000: ADC 模拟输入通道 0 00001: ADC 模拟输入通道 1 01111: ADC 模拟输入通道 15 10000: ADC 模拟输入通道 16 10001: ADC 模拟输入通道 17 10010: ADC 模拟输入通道 18 10011: ADC 模拟输入通道 19

25.5.4. ADC 控制寄存器 2 (ADC_CR2 偏移 0ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	FASTMOD	RW	0	ADC 快速转换模式(转换时间最短) 注：正常转换模式需要 16 个 ADC_CLK，快速转换模式只需 15 个 ADC_CLK。 0: 禁止 1: 使能 注：模拟温度传感器和 BUFFER 通道不支持快速转换模式
26	AFE_RSTN	RW	0	ADCAFE 复位信号,复位 AFE 内部数字部分。在 ADC_EN 使能后，等待 5us 释放（设为高）。 在强制退出某次转换，需要立刻启动一次新的转换时，可以拉低 ADC_RSTN 复位 AFE 数字部分，然后启动新的转换。 注：ADC 模块由模拟模块+数字控制模块构成，其中模拟模块称作模拟前端 Analog Front End(AFE)，AFE

				内部也包含寄存器、状态机等数字逻辑。
25	JOVSE	RW	0	注入组过采样功能使能 0: 禁止 1: 使能
24	TROVS	RW	0	规则组过采样触发模式 0: 一次触发进行 N 次 ADC 转换, N 是由 OVSR 决定的过采样率。 1: 一次触发进行 1 次 ADC 转换
23:20	OVSS[3:0]	RW	0000	过采样移位系数 0000: 不移位 0001: 右移 1 位 0010: 右移 2 位 0011: 右移 3 位 0100: 右移 4 位 0101: 右移 5 位 0110: 右移 6 位 0111: 右移 7 位 1000: 右移 8 位 其他: 保留
19:17	OVSR[2:0]	RW	000	过采样率 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x
16	OVSE	RW	0	规则组过采样功能使能

				0: 禁止 1: 使能
15:6	RSV	-	-	保留
5:4	RES	RW	00	数据分辨率 (Data resolution) 通过软件写入这些位可选择转换的分辨率。 00: 12 位 01: 10 位 10: 8 位 11: 6 位
3	ALIGN	RW	0	数据对齐 软件设置, 选择数据对齐模式 0: 右对齐 1: 左对齐
2	ADC_STP	RW	0	ADC 停止控制 该位由软件设置为 1 时, 由硬件等待当前 ADC 转换结束后, 该位由硬件清除。 0: 不执行 ADC 停止转换; 1: 写 1 用来停止 ADC, 读为 1 表明 ADC 转换仍在进行;
1	OVRMOD	RW	0	溢出模式 0: 发生溢出时 ADC_DR 保留上次采样数据; 1: 发生溢出时 ADC_DR 保存最新采样数据;
0	ADC_EN	RW	0	开/关 A/D 转换器 该位由软件设置和清除。当该位为 '0' 时, 写入 '1' 将把 ADC 从断电模式下唤醒。启动时间需要 2us 0: 关闭 ADC 转换, 并进入断电模式; 1: 开启 ADC 并启动转换。

25.5.5. ADC 采样时间寄存器 1 (ADC_SMPR1 偏移 10h)

比特	名称	属性	复位值	描述
31:28	SMP7[3:0]	RW	0000	通道 7 采样时间 具体定义见 SMP0
27:24	SMP6[3:0]	RW	0000	通道 6 采样时间 具体定义见 SMP0
23:20	SMP5[3:0]	RW	0000	通道 5 采样时间 具体定义见 SMP0
19:16	SMP4[3:0]	RW	0000	通道 4 采样时间 具体定义见 SMP0
15:12	SMP3[3:0]	RW	0000	通道 3 采样时间 具体定义见 SMP0
11:8	SMP2[3:0]	RW	0000	通道 2 采样时间 具体定义见 SMP0
7:4	SMP1[3:0]	RW	0000	通道 1 采样时间 具体定义见 SMP0
3:0	SMP0[3:0]	RW	0000	通道 0 采样时间 0000: 1 周期 0001: 3 周期 0010: 5 周期 0011: 7 周期 0100: 10 周期 0101: 13 周期 0110: 16 周期 0111: 20 周期 1000: 30 周期 1001: 60 周期 1010: 80 周期 1011: 100 周期 1100: 120 周期 1101: 160 周期 1110: 320 周期 1111: 480 周期

25.5.6. ADC 采样时间寄存器 2 (ADC_SMPR2 偏移 14h)

比特	名称	属性	复位值	描述
31:28	SMP15[3:0]	RW	0000	通道 15 采样时间 具体定义见 SMP0
27:24	SMP14[3:0]	RW	0000	通道 14 采样时间 具体定义见 SMP0
23:20	SMP13[3:0]	RW	0000	通道 13 采样时间 具体定义见 SMP0
19:16	SMP12[3:0]	RW	0000	通道 12 采样时间 具体定义见 SMP0
15:12	SMP11[3:0]	RW	0000	通道 11 采样时间 具体定义见 SMP0
11:8	SMP10[3:0]	RW	0000	通道 10 采样时间 具体定义见 SMP0
7:4	SMP9[3:0]	RW	0000	通道 9 采样时间 具体定义见 SMP0
3:0	SMP8[3:0]	RW	0000	通道 8 采样时间 具体定义见 SMP0

25.5.7. ADC 采样时间寄存器 3 (ADC_SMPR3 偏移 18h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19:16	SMP20[3:0]	RW	0000	通道 20 采样时间 具体定义见 SMP0
15:12	SMP19[3:0]	RW	0000	通道 19 采样时间 具体定义见 SMP0
11:8	SMP18[3:0]	RW	0000	通道 18 采样时间

				具体定义见 SMP0
7:4	SMP17[3:0]	RW	0000	通道 17 采样时间 具体定义见 SMP0
3:0	SMP16[3:0]	RW	0000	通道 16 采样时间 具体定义见 SMP0

25.5.8. ADC 看门狗高阈值寄存器(ADC_HTR 偏移 1Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	DHT[11:0]	RW	0x0	差分通道的模拟看门狗高阈值 这些位定义了模拟看门狗的阈值高限，用无符号数表示。
15:12	RSV	-	-	保留
11:0	HT[11:0]	RW	0x0	单端通道的模拟看门狗高阈值 这些位定义了模拟看门狗的阈值高限，用无符号数表示。

25.5.9. ADC 看门狗低阈值寄存器(ADC_LTR 偏移 20h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	DLT[11:0]	RW	0x0	差分通道的模拟看门狗低阈值 这些位定义了模拟看门狗的阈值低限，用有符号数表示。
15:12	RSV	-	-	保留
11:0	LT[11:0]	RW	0x0	单端通道的模拟看门狗低阈值 这些位定义了模拟看门狗的阈值低限，用无符号数表示。

25.5.10. ADC 规则序列寄存器 1(ADC_SQR1 偏移 24h)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:25	SQ5[4:0]	RW	00000	规则序列中的第 5 个转换
24:20	SQ4[4:0]	RW	00000	规则序列中的第 4 个转换
19:15	SQ3[4:0]	RW	00000	规则序列中的第 3 个转换
14:10	SQ2[4:0]	RW	00000	规则序列中的第 2 个转换
9:5	SQ1[4:0]	RW	00000	规则序列中的第 1 个转换 这些位由软件定义转换序列中的第 1 个转换通道的编号(0~18)。 00000: ADC 模拟输入通道 0 00001: ADC 模拟输入通道 1 01111: ADC 模拟输入通道 15 10000: ADC 模拟输入通道 16 10001: ADC 模拟输入通道 17 10010: ADC 模拟输入通道 18 10011: ADC 模拟输入通道 19
4	RSV	-	-	保留
3:0	L[3:0]	RW	0000	规则通道序列长度 这些位由软件定义在规则通道转换序列中的通道数目。 0000: 1 个转换 0001: 2 个转换 1111: 16 个转换

25.5.11. ADC 规则序列寄存器 2(ADC_SQR2 偏移 28h)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:25	SQ11[4:0]	RW	00000	规则序列中的第 11 个转换
24:20	SQ10[4:0]	RW	00000	规则序列中的第 10 个转换
19:15	SQ9[4:0]	RW	00000	规则序列中的第 9 个转换
14:10	SQ8[4:0]	RW	00000	规则序列中的第 8 个转换
9:5	SQ7[4:0]	RW	00000	规则序列中的第 7 个转换
4:0	SQ6[4:0]	RW	00000	规则序列中的第 6 个转换

25.5.12. ADC 规则序列寄存器 3(ADC_SQR3 偏移 2Ch)

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
24:20	SQ16[4:0]	RW	00000	规则序列中的第 16 个转换
19:15	SQ15[4:0]	RW	00000	规则序列中的第 15 个转换
14:10	SQ14[4:0]	RW	00000	规则序列中的第 14 个转换
9:5	SQ13[4:0]	RW	00000	规则序列中的第 13 个转换
4:0	SQ12[4:0]	RW	00000	规则序列中的第 12 个转换

25.5.13. ADC 注入通道寄存器(ADC_JSQR 偏移 30h)

比特	名称	属性	复位值	描述
31:27	JEXTSEL[4:0]	RW	00000	选择启动注入通道组转换的外部事件 选择用于启动注入通道组转换的外部触发 00000: JTRIG0 00001: JTRIG1 00010: JTRIG2

				... 10010: JTRIG18 10011: JTRIG19 其他: JTRIG31
26:25	JEXTEN	RW	00	注入通道的外部触发使能 选择外部触发极性和使能规则组的触发。 00: 禁止触发检测 01: 上升沿上的触发检测 10: 下降沿上的触发检测 11: 上升沿和下降沿上的触发检测
24:20	JSQ4[4:0]	RW	00000	注入序列中的第 4 个转换
19:15	JSQ3[4:0]	RW	00000	注入序列中的第 3 个转换
14:10	JSQ2[4:0]	RW	00000	注入序列中的第 2 个转换
9:5	JSQ1[4:0]	RW	00000	注入序列中的第 1 个转换 这些位由软件定义注入转换通道的编号(0~18)。
4:2	RSV	-	-	保留
1:0	JL[1:0]	RW	00	注入序列长度 注入通道转换序列中的转换总数,从 JSQ1 开始。 00: 1 次转换 (JSQ1) 01: 2 次转换 10: 3 次转换 11: 4 次转换 (JSQ1-JSQ2-JSQ3-JSQ4)

25.5.14. ADC 注入数据寄存器 x(x 为 1~4)(ADC_JDRx 偏移 34-40h)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20:16	JCH[4:0]	RO	00000	注入转换结果通道号 这些位为只读, 包含了注入通道的转换结果对应通道号

15:0	JDATA[15:0]	RO	0x0	注入转换的结果数据 这些位为只读，包含了注入通道的转换结果。数据可以以为有符号或者无符号

25.5.15. ADC 规则数据寄存器(ADC_DR 偏移 48h)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20:16	CH[4:0]	RO	00000	规则转换结果通道号 这些位为只读，包含了规则通道的转换结果对应通道号
15:0	DATA[15:0]	RO	0x0	规则转换的结果数据 这些位为只读，包含了规则通道的转换结果。数据可以以为有符号或者无符号

25.5.16. ADC 单端/差分选择寄存器 (ADC_DIFF 偏移 4Ch)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	DIFF9_19	RW	0	通道 9/19 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
8	DIFF8_18	RW	0	通道 9/18 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
7	DIFF7_17	RW	0	通道 7/17 单端/差分转换模式选择 该位由软件设置和清除。

				0: 单端模式 1: 差分模式
6	DIFF6_16	RW	0	通道 6/16 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
5	DIFF5_15	RW	0	通道 5/15 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
4	DIFF4_12	RW	0	通道 4/14 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
3	DIFF3_13	RW	0	通道 3/13 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
2	DIFF2_12	RW	0	通道 2/12 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
1	DIFF1_11	RW	0	通道 1/11 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式 1: 差分模式
0	DIFF0_10	RW	0	通道 0/10 单端/差分转换模式选择 该位由软件设置和清除。 0: 单端模式

				1: 差分模式

注：只对外部输入通道有效，内部通道恒定为无符号数。

25.5.17. ADC 符号数选择寄存器 (ADC_SIGN 偏移 50h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	SIGN9_19			通道 9/19 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
8	SIGN8_18			通道 8/18 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
7	SIGN7_17	RW	0	通道 7/17 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
6	SIGN6_16	RW	0	通道 6/16 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
5	SIGN5_15	RW	0	通道 5/15 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
4	SIGN4_14	RW	0	通道 4/14 转换结果符号数选择 该位由软件设置和清除。

				0: 结果为无符号数 1: 结果为有符号数
3	SIGN3_13	RW	0	通道 3/13 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
2	SIGN2_12	RW	0	通道 2/12 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
1	SIGN1_11	RW	0	通道 1/11 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数
0	SIGN0_10	RW	0	通道 0/10 转换结果符号数选择 该位由软件设置和清除。 0: 结果为无符号数 1: 结果为有符号数

注：只对外部输入 0~15 通道有效，内部通道和 BUF 通道恒定为无符号数。

25.5.18. ADC 偏移寄存器 x (x 为 1~4) (ADC_OFRx 偏移 60~6Ch)

比特	名称	属性	复位值	描述
31	OFFSETx_EN	RW	0	偏移x使能 软件配置是否使能偏移x功能 0: 偏移x功能禁止 1: 偏移 x 功能使能
30:26	OFFSETx_CH[4:0]	RW	0	偏移x通道选择 软件定义偏移 x 功能对应的通道号
25	OFFSETx_SAT	RW	0	偏移 x 结果格式选择 0: 计算结果为有符号数 1: 计算结果为无符号数, 最小为 0x000, 最大为 0xFFFF。

24	OFFSETx_POS	RW	0	偏移 x 的计算方式 0: 转换结果减去 OFFSETx[11:0] 1: 转换结果加上 OFFSETx[11:0]
23:12	RSV	-	-	保留
11:0	OFFSETx[11:0]	RW	0	偏移量 x OFFSETx_CH[4:0]选择的通道偏移补偿量，对于规则通道和注入通道都有效。转换结果与 OFFSETx 计算后，放入 ADC_DR 或者 ADC_JDRx 中。

25.5.19. ADC 模拟配置寄存器 (ADC_ANACFG 偏移 80h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:3	CLK_DLY_SEL	RW	00	数据采样时钟延迟选择 00: 默认延迟延迟时间 01: 默认延迟延迟时间*1.2 10: 默认延迟延迟时间*1.4 11: 时钟下降沿采样
2:0	BIAS_C	RW	101	SAR比较器BIAS功耗控制选择 000: 25% bias电流 001: 50% bias电流 010: 75% bias电流 011: 100% bias电流 (1Msps default) 100: 125% bias电流 101: 150% bias电流 (2Msps default) 110: 175% bias电流 111: 200% bias 电流 (3Msps default)

25.5.20. ADC 共用状态寄存器 (ADC_CSR 偏移 300h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23	AWD_SLV	RO	0	从 ADC 模拟看门狗标志位

				该位是从 ADC_SR 寄存器中 AWD 的副本
22	JEOG_SLV	RO	0	从 ADC 注入通道组转换结束位 该位是从 ADC_SR 寄存器中 JEOG 的副本
21	JEOC_SLV	RO	0	从 ADC 注入通道转换结束位 该位是从 ADC_SR 寄存器中 JEOC 的副本
20	OVERF_SLV	RO	0	从 ADC 规则通道数据溢出位 该位是从 ADC_SR 寄存器中 OVERF 的副本
19	EOG_SLV	RO	0	从 ADC 规则通道组转换结束位 该位是从 ADC_SR 寄存器中 EOG 的副本
18	EOC_SLV	RO	0	从 ADC 规则通道转换结束位 该位是从 ADC_SR 寄存器中 EOC 的副本
17	EOSMP_SLV	RO	0	从 ADC 规则通道采样完成标志 该位是从 ADC_SR 寄存器中 EOSMP 的副本
16	ADRDY_SLV	RO	0	从 ADC 就绪状态 该位是从 ADC_SR 寄存器中 ADRDY 的副本
15:8	RSV	-	-	保留
7	AWD_MST	RO	0	主 ADC 模拟看门狗标志位 该位是主 ADC_SR 寄存器中 AWD 的副本
6	JEOG_MST	RO	0	主 ADC 注入通道组转换结束位 该位是主 ADC_SR 寄存器中 JEOG 的副本
5	JEOC_MST	RO	0	主 ADC 注入通道转换结束位 该位是主 ADC_SR 寄存器中 JEOC 的副本
4	OVERF_MST	RO	0	主 ADC 规则通道数据溢出位 该位是主 ADC_SR 寄存器中 OVERF 的副本
3	EOG_MST	RO	0	主 ADC 规则通道组转换结束位 该位是主 ADC_SR 寄存器中 EOG 的副本
2	EOC_MST	RO	0	主 ADC 规则通道转换结束位 该位是主 ADC_SR 寄存器中 EOC 的副本
1	EOSMP_MST	RO	0	主 ADC 规则通道采样完成标志 该位是主 ADC_SR 寄存器中 EOSMP 的副本
0	ADRDY_MST	RO	0	主 ADC 就绪状态 该位是主 ADC_SR 寄存器中 ADRDY 的副本

25.5.21. ADC 共用控制寄存器 (ADC_CCR 偏移 304h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19:16	ADCDIV [3:0]	RW	0001	ADC_CLK 分频选择 该位由软件设置和清除，决定 ADC_CLK 相对于 HCLK 的分频数 注：ADC_CLK 最大不超过 48MHZ 0000：不分频 0001：2 分频 0010：3 分频 ... 1110：15 分频 1111：16 分频
15:14	DMADUAL[1:0]	RW	0	双 ADC 下 DMA 功能选择 00：主从 ADC 的各自 DMA 通道独立。 01：Reserved 10：主 ADC1 产生 DMA 请求,ADC_CDR 寄存读取数据，分辨率为支持 12 和 10 比特，ADC_CDR 上半个字包含 ADC2 的转换数据，低半个字包含 ADC1 的转换数据 11：主 ADC1 产生 DMA 请求,ADC_CDR 寄存读取数据，分辨率为支持 8 和 6 比特，ADC_CDR 的 15:8 包含 ADC2 的转换数据(SLV_DR[7:0])，7:0 包含 ADC1 的转换数据(MST_DR[7:0])
13:12	RSV	-	-	保留
11:8	DELAY[3:0]	RW	0	2 个采样阶段之间的延迟 这些位在双重交错模式下使用。 0000：5 * TADCCLK 0001：6 * TADCCLK 0010：7 * TADCCLK ... 1111：20 * TADCCLK
7:5	RSV	-	-	保留
4:0	DUALMOD[4:0]	RW	0	双 ADC 模式选择 — 所有 ADC 均独立： 00000：独立模式 — 00001 到 01001：双重模式，ADC1 和 ADC2 一

				起工作 00001: 规则同步 + 注入同步混合模式 00010: 规则同步 + 交替触发混合模式 00011: 规则交叉 + 注入同步混合模式 00101: 仅注入同步模式 00110: 仅规则同步模式 00111: 仅规则交叉模式 01001: 仅交替触发模式

25.5.22. ADC 共用规则数据寄存器 (ADC_CDR 偏移 308h)

比特	名称	属性	复位值	描述
31:16	DATA2[15:0]	RO	0	多重 ADC 规则转换的第二组结果数据。在双重模式下，这些位包含 ADC2 的规则数据。 这些位为只读，包含了规则通道的转换结果。数据可以为有符号或者无符号
15:0	DATA1[15:0]	RO	0	多重 ADC 规则转换的第一组结果数据。在双重模式下，这些位包含 ADC1 的规则数据。 这些位为只读，包含了规则通道的转换结果。数据可以为有符号或者无符号。

25.5.23. ADC 共用温度传感器/REF 寄存器 (ADC_CTSREF 偏移 30Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	HIZ_EN	RW	1	当内嵌 VREF 关闭后，VREFBI 信号高阻使能。 高有效。VREFBI 信号高阻使能时，可以减少漏电，同时可以外部输入 VREFP。
26:25	VREFBI_SEL	RW	00	ADC 内嵌 VREF 电压选择 注：使用内嵌 VREF 时，VREFP 需断开不供电 00: 1.5V 01: 2.0V

				1x: 2.5V
24	VREFBI_EN	RW	0	ADC 内嵌 VREF 使能，不受 ADC_EN 控制。最大需要 200us 启动时间 注：使用内嵌 VREF 时，VREFP 需断开不供电
23:19	VTRIM	RW	01110	内建 REF 电压输出 Trim 值
18:15	TTRIM	RW	0111	内建 REF 电压温度系数控制
14:12	RSV	-	-	保留
11	VBAT_EN	RW	0	VBAT 通道使能 注：内嵌 1/4 分压电路，使能后可以实时测量电池电压，测量范围 1.0~5.5V，无需外部电阻分压。 0: VBAT 关闭 1: VBAT 使能
10:9	BUF_ADDR[1:0]	RW	00	BUF 通道功能选择 00: 温度传感器 01: VBAT/4 xx: 其它
8:5	ADJ_TD_OS[3:0]	RW	1000	温度传感器失调系数 TRIM 值
4:1	ADJ_TD_GA[3:0]	RW	1000	温度传感器增益系数 TRIM 值
0	EN_TS	RW	0	温度传感器使能信号 该位由软件设置和清除。 0: 温度传感器断电 1: 温度传感器正常工作

25.6. 使用流程

25.6.1. ADC 操作流程

1. 设置 ADC 输入源：

根据需要设置 ADC 输入源，输入 GPIO 设为 GPIO_ANALOG，并打开 GPIO 模

拟开关

2. 设置 ADC_CR1/2 寄存器，配置工作模式，包括：
 - a) 连续模式 (CONT)
 - b) 注入转换是否使能 (JEN)
 - c) ADC_CLK 的分频 (DIV)
 - d) 规则/注入转换触发源 (EXTSEL/JEXTSEL)
 - e) 选择模拟看门模拟功能(AWDEN/J AWDEN/ AWDSGL/ AWDCH)
 - f) 是否支持 DMA 功能 (DMA)
 - g) 中断使能 (EOCIE/ JEOCIE/AWDIE)
 - h) 结果是否为有符号位 (ADC_SIGN)
3. 设置 ADC_SQR1/2/3，选择规则序列的长度和通道号
4. 设置 ADC_DIFF，选择每个通道差分/单端模式
5. 设置 ADC_JSQR，选择注入转换的通道号 (可选)
6. 设置 ADC_SMPR1/2/3，配置每个通道的采样时间
7. 设置 ADC_HTR/LTR，选择模拟看门狗的高低阈值 (可选)
8. 设置 DMA 通道相关设置 (可选)
9. 使能 ADC 转换电路(ADC_EN)
10. 等待外部 TRIG 或者触发软件 TRIG (SWSTART/ JSWSTART)，使能转换序列
11. 等待相应的中断信号，处理后重复 7-8 步骤。

25.6.2. 温度传感器操作

温度传感器的 VTEMP 与 ADC 转换电路的通道 16 相连，为了使温度传感器工作正常，需要额外的设置，流程如下

1. 设置 ADC_TSREF 寄存器 EN_TS 为 1，使能温度传感器
2. 从 NVR 读取 TRIM 值，设置 ADC_TSREF 寄存器 ADJ_TD_OS/ ADJ_TD_GA 位，配置温度传感器
3. 设置 ADC_TSREF 的 ADDRESS 位，选择温度传感器

26. 运算放大器（OPA）

26.1. 概述

OPA 模块包含三个独立配置的运算放大器。每个运算放大器有两个输入和一个输出，三个 I/O 可以连接到外部引脚。运算放大器可以在内部配置为一个跟随器，或者是一个可编程增益的放大器。

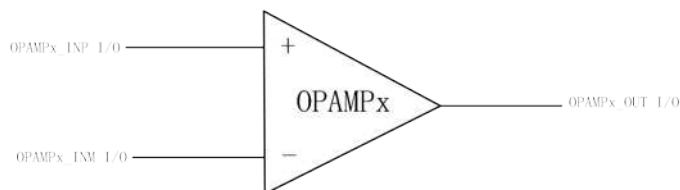
26.2. 主要特性

- 轨到轨输入输出；
- 可配置成运放外置模式；
- 可配置成 PGA 模式，增益可编程；
- 可配置成跟随器模式，实现单位增益；
- 运算放大器输出到引脚；
- 运算放大器输出可作为 ADC 的输入；
- 运算放大器正端输入来自引脚或者是 DAC 的输出；
- 运算放大器负端输入来自引脚、OPA 或者是 PGA 的反馈电阻网络输出。

26.3. 功能描述

26.3.1. 功能示意图

下图为运放信号通路。



26.3.2. 操作模式和校准

运放可工作在正常模式和校准模式。正常模式下软件可对运放的失调电压进行校准。

模式	控制位			输出	
	EN	CAL_NEN	CAL_PEN	Vout	CALout flag
正常模式	1	0	0	analog	0
		1	1	analog	0
掉电模式	0	x	x	z	0
N 端校准模式	1	1	0	analog	x
P 端校准模式	1	0	1	analog	x

注：当正常模式，EN=1，CAL_NEN=1，CAL_PEN=1，校准值生效。当正常模式，EN=1，CAL_NEN=0，CAL_PEN=0，校准值无效，此时采用默认值。

26.3.3. 工作模式和负端输入

工作模式	MODE_SEL	VINM0_EN	VINM1	VINM0
SA(运放外置)	00	0	输入	-
UG(单位增益)	01	0	-	-
PGA(倍数可调)	10	0	反相输入	-
SA(运放外置)	11	1	-	输入

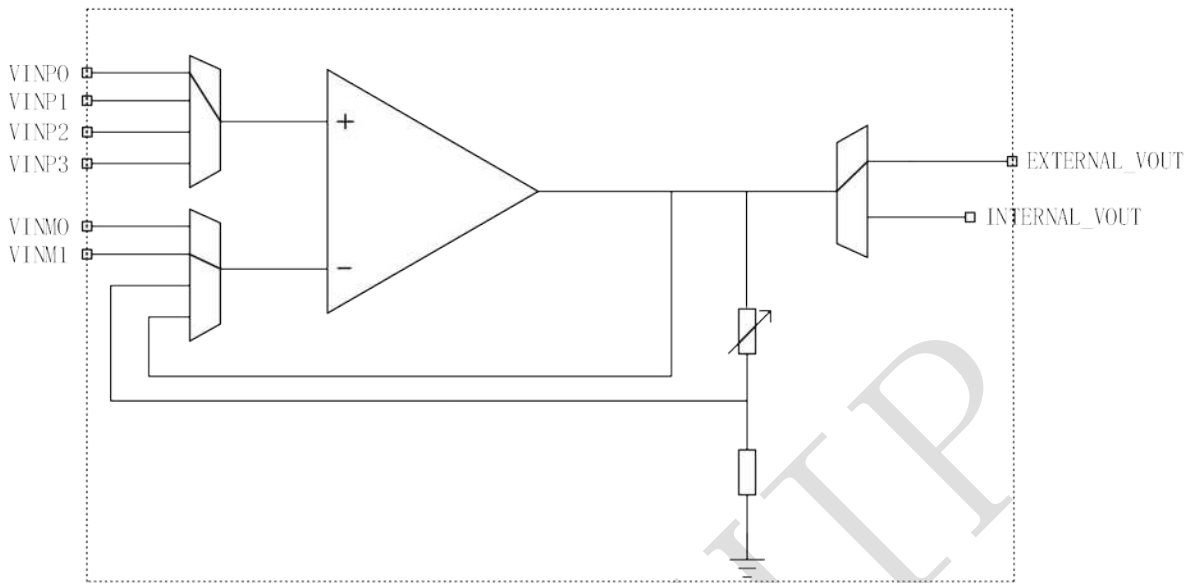
26.3.4. 负端输入管脚

OPA 编号	VINM1	VINM0
OPA1	PC5	PA3
OPA2	PC5	PA5
OPA3	PB10	PB2

26.3.5. 运放外置模式（SA 模式）

配置寄存器中 MODE_SEL 配置成 0'b00 或 0'b11 时可以将运放配置成外置模式，此时内部电阻被切断，需通过外部电路来调整运放功能。等效的电路如下图所示：

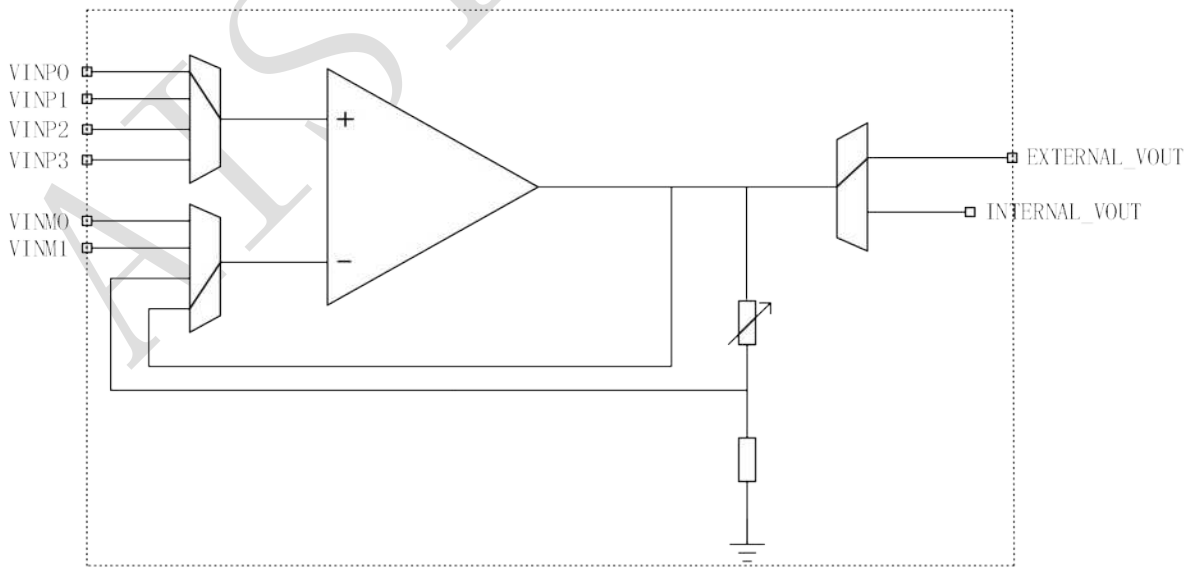
图 26-1 运放外置模式配置等效图



26.3.6. 运放单位增益模式（UG 模式）

配置寄存器中 MODE_SEL 配置成 0'b01 时可以将运放配置成单位增益模式，此时内部电阻被切断，负端输入直接连接到输出端，此时运放增益为 1。等效的电路如下图所示：

图 26-2 运放单位增益模式配置等效图



26.3.7. 可编程增益放大模式（PGA 模式）

配置寄存器中 `MODE_SEL` 配置成 `0'b10` 时，工作模式选择为 PGA。可以通过寄存器配置运放的放大倍数，从而可省略外部电阻。配置 `OPAx_CSR` 中的 `POL_SEL` 位域可以配置运放为同相或反相模式，配置 `GAIN_SEL` 位域可以编程增益倍数，同相和反相的增益有区别。

配置为 PGA 同相模式时，负端输入端无效；配置为 PGA 反相模式时，正端输入和负端输入均有效。

图 26-3 运放 PGA 同相模式配置等效图

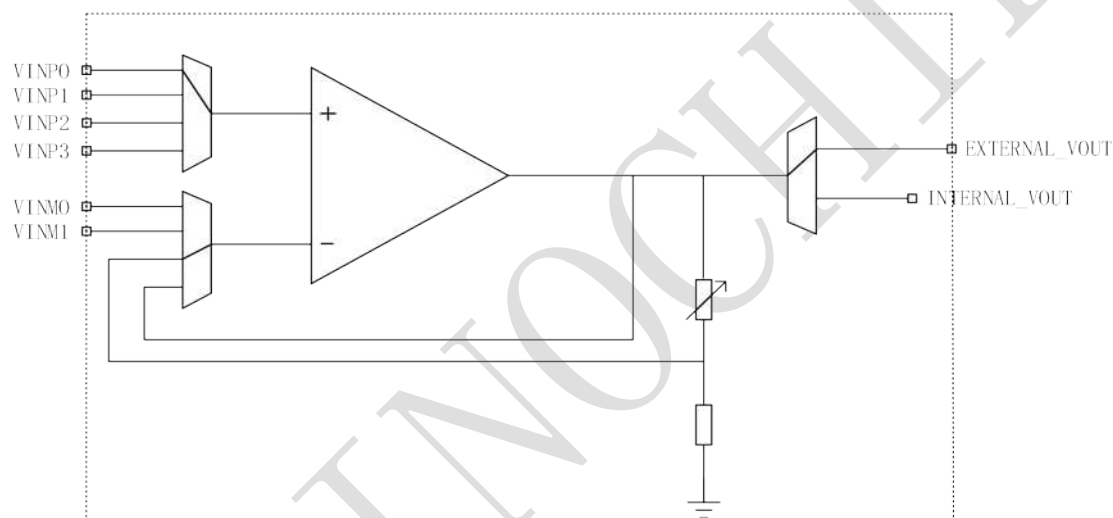
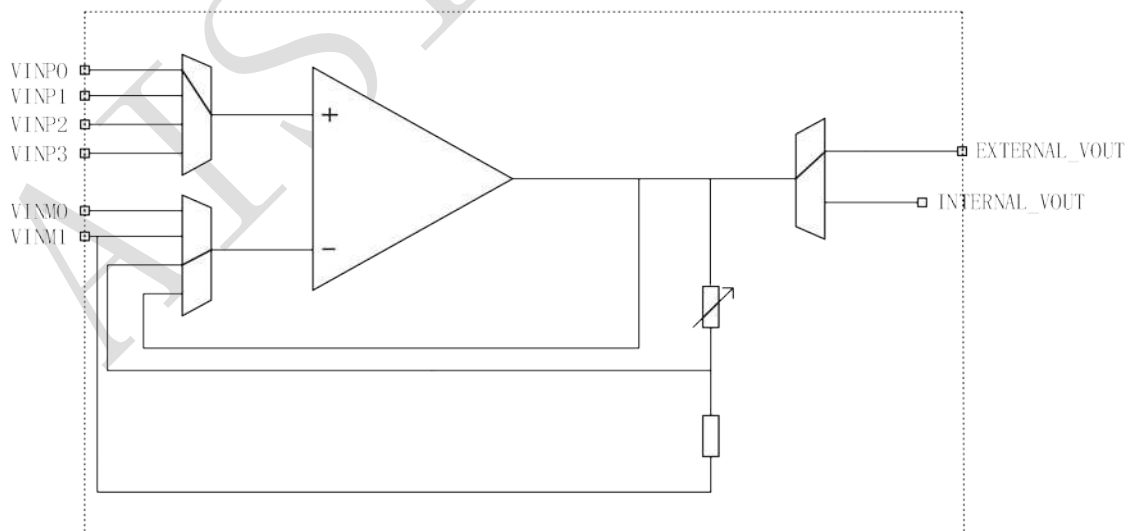


图 26-4 运放 PGA 反相模式配置等效图



26.4. 寄存器描述

OPA 寄存器基地址：0x40010300

偏置	名称	属性	描述
0x00	OPA_CSR1	R/W	OPA1 控制/状态寄存器
0x04	OPA_CSR2	R/W	OPA2 控制/状态寄存器
0x08	OPA_CSR3	R/W	OPA3 控制/状态寄存器

26.4.1. OPA1 控制寄存器(OPA_CSR1 偏移 00h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	OPA_CSR1 寄存器写保护控制 该位由软件设置，通过 RCC 对模块进行复位清除。 0：允许软件写入 OPA_CSR1 寄存器 1：禁止软件写入 OPA_CSR1 寄存器
30:29	RSV	RO	0	保留位。
28	HSM	RW	0	驱动模式选择 0：低驱动 1：高驱动
27:23	GAIN_SEL	RW	10000	PGA1 增益倍数选择 同相： 00000：64 00001：32 00010：16 00100：8 01000：4 10000：2 反相： 00000：63 00001：31 00010：15

				00100: 7 01000: 3 10000: 1
22:21	POL_SEL	RW	0	极性选择（仅对 PGA 模式有效） 00: 同相 01: 反相 其他: 保留
20	VINM0_EN	RW	0	负端输入通道 0 使能 0: 禁止 1: 使能
19:18	MODE_SEL	RW	0	工作模式选择 00: SA（运放外置模式） 01: UG（单位增益） 10: PGA（倍数可调） 11: SA（运放外置模式）
17:16	VINP_SEL	RW	0	正端信号选择 00: PA1 01: PA3 10: PA7 11: DAC1 输出
15:14	OUT_SEL	RW	0	输出通道信号选择 00: 输出到 GPIO。 01: 输出到内部（ADC 模块），并且断开外部输出。 其他: 保留
13:9	TRIM_OSN	RW	0	OPA1 的 N 差分对管失调电压的修调值。
8:4	TRIM_OSP	RW	0	OPA1 的 P 差分对管失调电压的修调值。
3	CAL_OUT	RO	0	OPA1 的差分对管失调电压修调完成标志 0: 未完成 1: 完成
2	CAL_NEN	RW	0	OPA1 的 N 差分对管失调电压修调使能 0: 禁止 1: 使能

1	CAL_PEN	RW	0	OPA1 的 P 差分对管失调电压修调使能 0: 禁止 1: 使能
0	EN	RW	0	OPA1 使能位 0: 禁止 1: 使能

26.4.2. OPA2 控制寄存器(OPA_CSR2 偏移 04h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	OPA_CSR2 寄存器写保护控制 该位由软件设置，通过系统控制单元 SCU 复位清除。 0: 允许软件写入 OPA_CSR2 寄存器 1: 禁止软件写入 OPA_CSR2 寄存器
30:29	RSV	RO	0	保留位。
28	HSM	RW	0	驱动模式选择 0: 低驱动 1: 高驱动
27:23	GAIN_SEL	RW	10000	PGA2 增益倍数选择 同相: 00000: 64 00001: 32 00010: 16 00100: 8 01000: 4 10000: 2 反相: 00000: 63 00001: 31 00010: 15 00100: 7 01000: 3

				10000: 1
22:21	POL_SEL	RW	0	极性选择（仅对 PGA 模式有效） 00: 同相 01: 反相 其他: 保留
20	VINM0_EN	RW	0	负端输入通道 0 使能 0: 禁止 1: 使能
19:18	MODE_SEL	RW	0	工作模式选择 00: SA（运放外置模式） 01: UG（单位增益） 10: PGA（倍数可调） 11: SA（运放外置模式）
17:16	VINP_SEL	RW	0	正端信号选择 00: PA7 01: PB0 10: OPA1 输出 11: DAC2 输出
15:14	OUT_SEL	RW	0	输出通道信号选择 00: 输出到 GPIO。 01: 输出到内部（ADC 模块），并且断开外部输出。 其他: 保留
13:9	TRIM_OSN	RW	0	OPA2 的 N 差分对管失调电压的修调值。
8:4	TRIM_OSP	RW	0	OPA2 的 P 差分对管失调电压的修调值。
3	CAL_OUT	RO	0	OPA2 的差分对管失调电压修调完成标志 0: 未完成 1: 完成
2	CAL_NEN	RW	0	OPA2 的 N 差分对管失调电压修调使能 0: 禁止 1: 使能
1	CAL_PEN	RW	0	OPA2 的 P 差分对管失调电压修调使能 0: 禁止

				1: 使能
0	EN	RW	0	OPA2 使能位 0: 禁止 1: 使能

26.4.3. OPA3 控制寄存器(OPA_CSR3 偏移 08h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	OPA_CSR3 寄存器写保护控制 该位由软件设置，通过系统控制单元 SCU 复位清除。 0: 允许软件写入 OPA_CSR3 寄存器 1: 禁止软件写入 OPA_CSR3 寄存器
30:29	REV	RO	0	保留位。
28	HSM	RW	0	驱动模式选择 0: 低驱动 1: 高驱动
27:23	GAIN_SEL	RW	10000	PGA3 增益倍数选择 同相: 00000: 64 00001: 32 00010: 16 00100: 8 01000: 4 10000: 2 反相: 00000: 63 00001: 31 00010: 15 00100: 7 01000: 3 10000: 1
22:21	POL_SEL	RW	0	极性选择（仅对 PGA 模式有效） 00: 同相

				01: 反相 其他: 保留
20	VINM0_EN	RW	0	负端输入通道 0 使能 0: 禁止 1: 使能
19:18	MODE_SEL	RW	0	工作模式选择 00: SA (运放外置模式) 01: UG (单位增益) 10: PGA (倍数可调) 11: SA (运放外置模式)
17:16	VINP_SEL	RW	0	正端信号选择 00: PB0 01: PA1 10: PB11 11: DAC1 输出
15:14	OUT_SEL	RW	0	输出通道信号选择 00: 输出到 GPIO。 01: 输出到内部 (ADC 模块), 并且断开外部输出。 其他: 保留
13:9	TRIM_OSN	RW	0	OPA3 的 N 差分对管失调电压的修调值。
8:4	TRIM_OSP	RW	0	OPA3 的 P 差分对管失调电压的修调值。
3	CAL_OUT	RO	0	OPA3 的差分对管失调电压修调完成标志 0: 未完成 1: 完成
2	CAL_NEN	RW	0	OPA3 的 N 差分对管失调电压修调使能 0: 禁止 1: 使能
1	CAL_PEN	RW	0	OPA3 的 P 差分对管失调电压修调使能 0: 禁止 1: 使能
0	EN	RW	0	OPA3 使能位 0: 禁止

				1: 使能

ALISINOCHIP

26.5. 使用流程

26.5.1. 普通使用流程

1. 将运放对应的 GPIO 口配置成模拟端口。
2. 通过 MODE_SEL 配置运放的工作模式。
3. 通过 HSM 配置运放的驱动模式。
4. 配置运放的正端信号选择 (VINP_SEL)。
5. 将负端信号选择为 VINM0 (VINM0_EN 置 1) (仅模式 3 (SA))。
6. 通过 OUT_SEL 配置运放的输出。
7. 配置 PGA 选项 (PGA 模式)。
8. 从 NVR 中读取 TRIM 值, 配置 TRIM_OSN/TRIM_OSP。
9. 配置 CAL_NEN /CAL_PEN 为 1 使 TRIM 值生效。
10. 使能运放。

26.5.2. 使用校准功能流程

1. 前序步骤同普通使用流程步骤 1~7。
2. 使能运放。
3. 配置 CAL_NEN 为 1、CAL_PEN 为 0。
4. 从 0~31 填写 TRIM_OSN, 直到 CAL_OUT 标志为 1 为止。
5. 配置 CAL_NEN 为 0、CAL_PEN 为 1。
6. 从 0~31 填写 TRIM_OSP, 直到 CAL_OUT 标志为 1 为止。
7. 配置 CAL_NEN /CAL_PEN 为 1 使 TRIM 值生效。

27. 模拟比较器（COMP）

27.1. 概述

模拟电压比较器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。当比较器正端输入电压高于负端输入电压时，电压比较器输出高电平；当比较器正端输入电压低于负端输入电压时，电压比较器输出低电平。

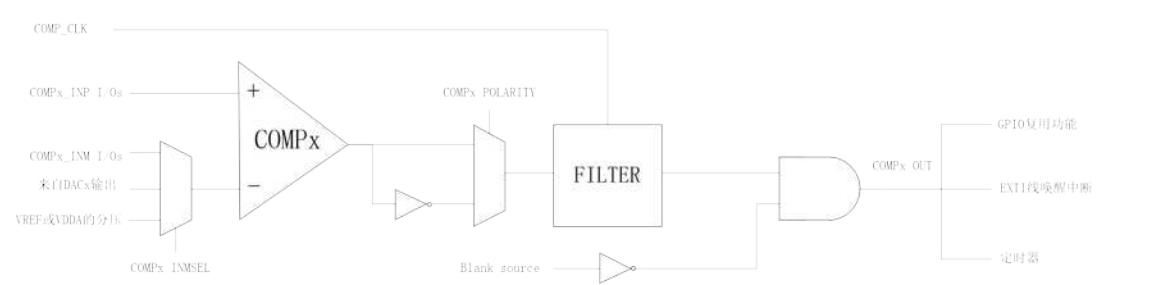
27.2. 主要特性

- 支持电压比较功能；
- 比较器负端输入可配置：
 - 芯片管脚；
 - DAC 的输出；
 - 内部基准电压 VREF 或 VDDA 的分压；
- 可编程的迟滞窗口；
- 比较器输出到管脚；
- 比较器输出可作为定时器的刹车输入或捕获输入；
- 比较器输出可通过定时器切断；
- 比较器输出可作为 EXTI 控制器输入，支持 Sleep 和 Stop 模式下的唤醒功能；
- 两个比较器级联可以组合构成窗口比较器；
- 提供软件可配置的滤波时间以增强芯片的抗干扰能力。

27.3. 结构框图

下图为比较器结构框图。

图 27-1 COMP 结构框图



AISINOCHIP

27.4. 功能描述

27.4.1. 负端输入

负端输入可以选择来自芯片管脚、DAC 输出、或者来自内部基准电压 VREF 或 VDDA 的分压。通过配置 COMP_CR 的 INMSEL 位域选择。

选择内部基准电压分压时，可以通过配置 CRV_SEL 位域选择基准分压来源为 VDDA 或 VREF，分压系数通过配置 CRV_CFG 来实现内部输入不同负端电压。

27.4.2. 滤波

滤波功能可以滤除来自比较器输入端的尖峰毛刺，防止应用电路的误触发。比较器滤波电路可以滤除因毛刺输入产生的窄脉冲输出，可以通过配置控制寄存器的 FLTEN 位使能滤波功能，配置 FLTIME 位域可以更改滤除毛刺的最大宽度。

比较器滤波时钟可通过设置 RCC 模块中的 CCR2 寄存器，通过设置其中的 FLTCLKSEL 位来选择，设置 0 时滤波时钟为 PCLK 的 32 分频，设置 1 时为 RC32K。

注：在比较器用于唤醒 STOP 模式 MCU，且需要使用比较器滤波功能时，滤波时钟只能选择 RC32K。

27.4.3. 迟滞比较

迟滞比较功能可以防止在比较电压附近时，输出产生振荡。通过配置控制寄存器的 HYS 位域开启或更改迟滞窗口的电压范围值。

27.4.4. 切断

切断功能可以通过其他输入来切断比较器输出。通过配置控制寄存器的 BLANKSEL 位域开启或更改切断源，配置 BLANKTIME 位域更改切断窗口时间宽度。

27.4.5. STOP 模式唤醒

比较器输出支持将 MCU 从 STOP 模式唤醒。比较器输出在内部连接到 EXTI 模块的触发输入，详情见 EXTI 模块的“触发源”章节。

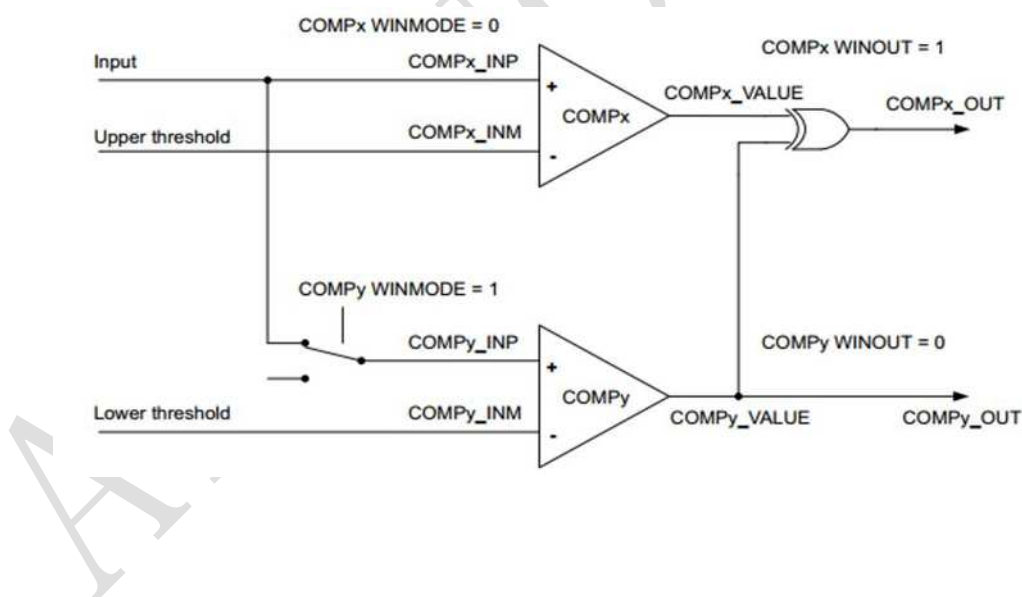
27.4.6. 窗口模式级联

可以将两个比较器连接成窗口模式，设置两个负端为不同输入电压，其中一个为上门限电压，一个为下门限电压，这样输入在门限电压之间时产生比较输出。

设置为窗口模式级联时，如下图所示：

- 需将 COMPx 的输出设置 WINOUT 设为 1, 此时输出为 COMPx_VALUE 异或 COMPy_VALUE 的结果；
- 需将 COMPy 的正端输入设置 WINMODE 设为 1, 此时 COMPy 的正端输入连接到 COMPx 的正端输入。

图 27-2 比较器窗口模式级联



27.5. 寄存器描述

COMP 寄存器基地址：0x4001_0200

偏置	名称	属性	描述
0x00	COMP_CR1	RW	COMP1 控制寄存器
0x04	COMP_CR2	RW	COMP2 控制寄存器
0x08	COMP_CR3	RW	COMP3 控制寄存器
0x0C	COMP_CR4	RW	COMP4 控制寄存器
0x10	COMP_SR	RW	COMP 状态寄存器

27.5.1. COMP1 控制寄存器(COMP_CR1 偏移 00h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	COMP_CR1 寄存器写保护控制 该位由软件设置，通过系统控制单元 SCU 复位清除。 0：允许软件写入 COMP_CR1 寄存器 1：禁止软件写入 COMP_CR1 寄存器
30:29	RSV	-	-	保留
28:25	CRV_CFG	RW	0000	基准电阻分压配置 分压为：(CRV_CFG+1)/20。
24	CRV_SEL	RW	0	基准分压来源选择。 0：选择 VDDA 1：选择 VREF
23	CRV_EN	RW	0	基准分压使能信号。 0：禁止 1：使能
22	WINMODE	RW	0	比较器 1 窗口模式正端输入选择

				0: 比较器 1 的 INPSEL 决定。 1: 比较器 2 的正端
21	WINOUT	RW	0	比较器 1 输出模式控制 0: 输出 VCOUT1 1: 输出 VCOUT1 XOR VCOUT2
20	POLARITY	RW	0	比较器 1 极性选择
19	FLTEN	RW	0	比较器 1 滤波使能 0: 禁止 1: 使能
18:16	FLTTIME	RW	000	比较器滤波时间配置，滤波时间按 FILT_CLK 计算，FILT_CLK 由系统控制单元 SCU 进行配置。 000: 1 个周期 001: 2 个周期 010: 4 个周期 011: 16 个周期 100: 64 个周期 101: 256 个周期 110: 1024 个周期 111: 4095 个周期
15	RSV	-	-	保留
14:12	BLANKSEL	RW	000	比较器 1 切断源选择 000: 不切断 001: TIM1_OC5 010: TIM2_OC3 011: TIM3_OC3 100: TIM8_OC5 101: TIM1_OC4 110: TIM15_OC1

				111: TIM4_OC3
11:8	INPSEL	RW	0000	比较器 1 正端信号选择 0000: PA1 0001: PB1 0010: PB10 其它: 保留
7:4	INMSEL	RW	0000	比较器 1 负端信号选择 0000: PA4 0001: PA0 0010: DAC1 输出 0011: VREF 或 VDDA 的分压 其它: 保留
3:1	HYS	RO	000	比较器 1 迟滞窗口选择 0xx: 禁止迟滞功能 100: 12mV 101: 24mV 110: 36mV 111: 48mV
0	EN	RW	0	比较器 1 使能位 0: 禁止 1: 使能

27.5.2. COMP2 控制寄存器(COMP_CR2 偏移 04h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	COMP_CR2 寄存器写保护控制 该位由软件设置, 通过系统控制单元 SCU 复位清除。 0: 允许软件写入 COMP_CR2 寄存器

				1: 禁止软件写入 COMP_CR2 寄存器
30:29	RSV	-	-	保留
28:25	CRV_CFG	RW	0000	基准电阻分压配置 分压为: $(CRV\_CFG+1)/20$ 。
24	CRV_SEL	RW	0	基准分压来源选择。 0: 选择 VDDA 1: 选择 VREF
23	CRV_EN	RW	0	基准分压使能信号。 0: 禁止 1: 使能
22	WINMODE	RW	0	比较器 2 窗口模式正端输入选择 0: 比较器 2 的 INPSEL 决定。 1: 比较器 1 的正端
21	WINOUT	RW	0	比较器 2 输出模式控制 0: 输出 VCOUT2 1: 输出 VCOUT1 XOR VCOUT2
20	POLARITY	RW	0	比较器 2 极性选择
19	FLTEN	RW	0	比较器 2 滤波使能 0: 禁止 1: 使能
18:16	FLTTIME	RW	000	比较器滤波时间配置, 滤波时间按 FILT_CLK 计算, FILT_CLK 由系统控制单元 SCU 进行配置。 000: 1 个周期 001: 2 个周期 010: 4 个周期 011: 16 个周期 100: 64 个周期 101: 256 个周期

				110: 1024 个周期 111: 4095 个周期
15	RSV	-	-	保留
14:12	BLANKSEL	RW	000	比较器 2 切断源选择 000: 不切断 001: TIM1_OC5 010: TIM2_OC3 011: TIM3_OC3 100: TIM8_OC5 101: TIM1_OC4 110: TIM15_OC1 111: TIM4_OC3
11:8	INPSEL	RW	0000	比较器 2 正端信号选择 0000: PA7 0001: PA3 0010: PB11 其它: 保留
7:4	INMSEL	RW	0000	比较器 2 负端信号选择 0000: PA5 0001: PA2 0010: DAC2 输出 0011: VREF 或 VDDA 的分压 其它: 保留
3:1	HYS	RO	000	比较器 2 迟滞窗口选择 0xx: 禁止迟滞功能 100: 12mV 101: 24mV 110: 36mV 111: 48mV

0	EN	RW	0	比较器 2 使能位 0: 禁止 1: 使能

27.5.3. COMP3 控制寄存器(COMP_CR3 偏移 08h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	COMP_CR3 寄存器写保护控制 该位由软件设置, 通过系统控制单元 SCU 复位清除。 0: 允许软件写入 COMP_CR3 寄存器 1: 禁止软件写入 COMP_CR3 寄存器
30:29	RSV	-	-	保留
28:25	CRV_CFG	RW	0000	基准电阻分压配置 分压为: $(CRV\_CFG+1)/20$ 。
24	CRV_SEL	RW	0	基准分压来源选择。 0: 选择 VDDA 1: 选择 VREF
23	CRV_EN	RW	0	基准分压使能信号。 0: 禁止 1: 使能
22	WINMODE	RW	0	比较器 3 窗口模式正端输入选择 0: 比较器 3 的 INPSEL 决定。 1: 比较器 4 的正端
21	WINOUT	RW	0	比较器 3 输出模式控制 0: 输出 VCOUT3 1: 输出 VCOUT3 XOR VCOUT4
20	POLARITY	RW	0	比较器 3 极性选择
19	FLTEN	RW	0	比较器 3 滤波使能

				0: 禁止 1: 使能
18:16	FLTTIME	RW	000	比较器滤波时间配置，滤波时间按 FILT_CLK 计算，FILT_CLK 由系统控制单元 SCU 进行配置。 000: 1 个周期 001: 2 个周期 010: 4 个周期 011: 16 个周期 100: 64 个周期 101: 256 个周期 110: 1024 个周期 111: 4095 个周期
15	RSV	-	-	保留
14:12	BLANKSEL	RW	000	比较器 3 切断源选择 000: 不切断 001: TIM1_OC5 010: TIM3_OC3 011: TIM2_OC4 100: TIM8_OC5 101: TIM8_OC4 110: TIM15_OC1 111: TIM4_OC3
11:8	INPSEL	RW	0000	比较器 3 正端信号选择 0000: PA0 0001: PC1 0010: PC3 其它: 保留
7:4	INMSEL	RW	0000	比较器 3 负端信号选择

				0000: PF1 0001: PC0 0010: DAC1 输出 0011: VREF 或 VDDA 的分压 其它: 保留
3:1	HYS	RO	000	比较器 3 迟滞窗口选择 0xx: 禁止迟滞功能 100: 12mV 101: 24mV 110: 36mV 111: 48mV
0	EN	RW	0	比较器 3 使能位 0: 禁止 1: 使能

27.5.4. COMP4 控制寄存器(COMP_CR4 偏移 0Ch)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	COMP_CR4 寄存器写保护控制 该位由软件设置, 通过系统控制单元 SCU 复位清除。 0: 允许软件写入 COMP_CR4 寄存器 1: 禁止软件写入 COMP_CR4 寄存器
30:29	RSV	-	-	保留
28:25	CRV_CFG	RW	0000	基准电阻分压配置 分压为: $(CRV\_CFG+1)/20$ 。
24	CRV_SEL	RW	0	基准分压来源选择。 0: 选择 VDDA 1: 选择 VREF

23	CRV_EN	RW	0	基准分压使能信号。 0: 禁止 1: 使能
22	WINMODE	RW	0	比较器 4 窗口模式正端输入选择 0: 比较器 4 的 INPSEL 决定。 1: 比较器 3 的正端
21	WINOUT	RW	0	比较器 4 输出模式控制 0: 输出 VCOUT4 1: 输出 VCOUT3 XOR VCOUT4
20	POLARITY	RW	0	比较器 4 极性选择
19	FLTEN	RW	0	比较器 4 滤波使能 0: 禁止 1: 使能
18:16	FLTTIME	RW	000	比较器滤波时间配置，滤波时间按 FILT_CLK 计算，FILT_CLK 由系统控制单元 SCU 进行配置。 000: 1 个周期 001: 2 个周期 010: 4 个周期 011: 16 个周期 100: 64 个周期 101: 256 个周期 110: 1024 个周期 111: 4095 个周期
15	RSV	-	-	保留
14:12	BLANKSEL	RW	000	比较器 4 切断源选择 000: 不切断 001: TIM3_OC4 010: TIM8_OC5

				011: TIM15_OC1 100: TIM1_OC5 101: TIM8_OC4 110: TIM15_OC1 111: TIM4_OC3
11:8	INPSEL	RW	0000	比较器 4 正端信号选择 0000: PB0 0001: PE7 0010: PE9 其它: 保留
7:4	INMSEL	RW	0000	比较器 4 负端信号选择 0000: PE8 0001: PB2 0010: DAC1 输出 0011: VREF 或 VDDA 的分压 其它: 保留
3:1	HYS	RO	000	比较器 4 迟滞窗口选择 0xx: 禁止迟滞功能 100: 12mV 101: 24mV 110: 36mV 111: 48mV
0	EN	RW	0	比较器 4 使能位 0: 禁止 1: 使能

27.5.5. COMP 状态寄存器(COMP_SR 偏移 10h)

比特	名称	属性	复位值	描述

31:8	RSV	RO	0	保留位。
7	VCOUT4_ORG	RO	0	比较器 4 原始输出状态
6	VCOUT3_ORG	RO	0	比较器 3 原始输出状态
5	VCOUT2_ORG	RO	0	比较器 2 原始输出状态
4	VCOUT1_ORG	RO	0	比较器 1 原始输出状态
3	VCOUT4	RO	0	比较器 4 滤波输出状态
2	VCOUT3	RO	0	比较器 3 滤波输出状态
1	VCOUT2	RO	0	比较器 2 滤波输出状态
0	VCOUT1	RO	0	比较器 1 滤波输出状态

27.6. 使用流程

1. 将比较器对应的 GPIO 口配置成模拟端口。
2. 配置比较器的正端信号选择 (INPSEL) 和负端信号选择 (INMSEL)。
3. 设置窗口模式正端输入选择 WINMODE (窗口模式可选)。
4. 设置输出模式 WINOUT (窗口模式可选)。
5. 设置输出极性 POLARITY。
6. 设置 CRV_SEL 选择基准分压源 (内部基准源可选)。
7. 设置 CRV_EN 使能基准分压 (基准分压可选)。
8. 设置分压系数 CRV_CFG (基准分压可选)。
9. 设置 FLTEN 使能滤波功能 (滤波可选)。
10. 设置滤波时间配置 (FLTTIME) (滤波可选)。
11. 设置 HYS 配置迟滞比较窗口值 (迟滞比较可选)。
12. 设置 BLANKSEL 选择切断源 (切断可选)。
13. 设置和切断源对应的系统定时器实现触发功能 (切断可选)。
14. 设置 EN 使能比较器。
15. 设置控制寄存器 COMP_CR 的 LOCK 位锁定寄存器 (可选)。
16. 在 VOUT 端测量或者在 COMP_SR 状态寄存器中读取输出信号。
17. 如需将输出用作定时器刹车输入功能, 请在定时器刹车输入相关寄存器中配置 (可选)。
18. 如需将输出用作 EXTI 功能, 请配置 EXTI 的触发源为对应比较器 (可选)。

28. 控制器区域网络（CAN）

28.1. 概述

CAN 是控制器局域网络(Controller Area Network)的简称,是一种异步的半双工通讯。ACM32G103 芯片中集成了 2 路 CAN。本芯片中的 CAN 模块已通过 BOSCH CAN2.0 测试。

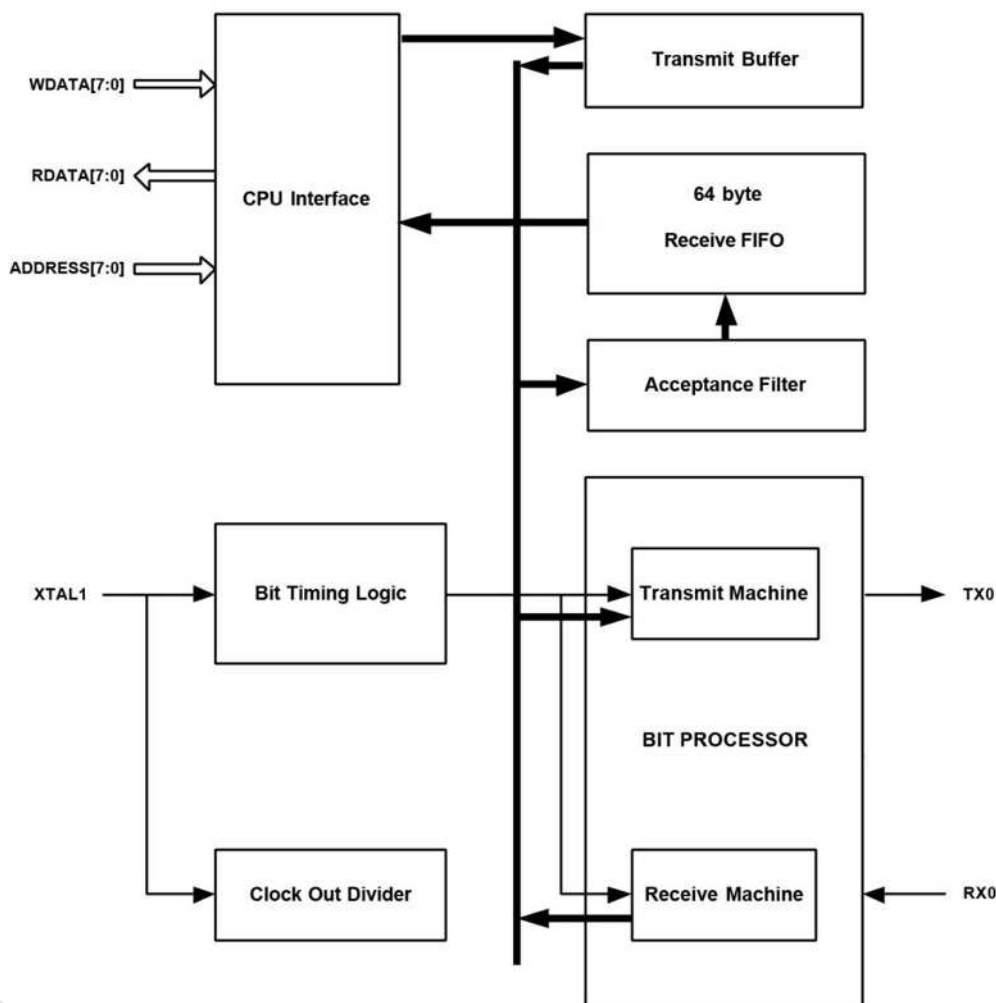
28.2. 主要特性

- 支持 CAN2.0, 包括 CAN2.0A 和 CAN2.0B。
- 支持 11 比特和 29 比特的识别符。
- 支持最低 125KB 波特率和 1MB 波特率。
- 64 字节的接收 FIFO。
- 支持热拔插。
- 支持 7 组接收器滤波。
- Single-Shot 传输选项。
- 支持只监听模式。
- 可以接收自己的信息。
- 支持自测模式。
- 支持 CAN 总线错误的中断。
- 记录仲裁失败后的 bit 位置。
- 读写错误计数器。
- 可编程的错误上限警告。

28.3. 功能描述

28.3.1. 功能框图

图 28-1 内部功能框图



注：CPU 通过独立的地址线、输入数据线、输出数据线来访问 CAN 模块。发送的数据被放到 CAN_TXBUFF 中去，并由发送器发送出去。接收的数据先通过接收滤波器过滤以后再放到 CAN_RXFIFO 中。CPU 通过一个 13bytes 宽度的窗口访问 CAN_RXFIFO。CAN_RXFIFO 总共有 64 个字节的缓存，可以一次性存储 5 组扩展帧。Bit-Timing-Logic 模块负责用来产生波特率。TX0 用于发送，RX0 用于接收。

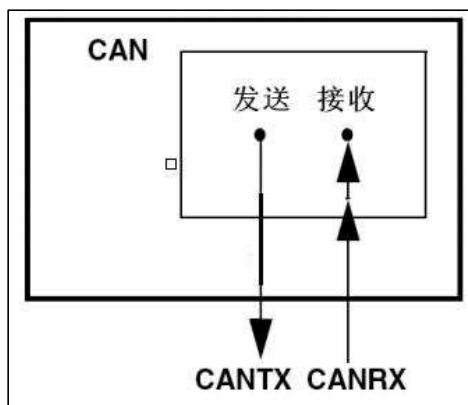
28.3.2. 操作模式

CAN 有两种工作模式：

- **Operating Mode:** 可以正常发送接收 CAN 总线数据

进入 Operating Mode 的方式是将 CAN_MOD.RM 清零。

图 28-2 正常模式示意图



- **Reset Mode:** 可以修改时间参数和报文过滤参数。

进入 Reset Mode 的方式有两种：第一种就是执行一次 CAN 模块复位，第二种方式是将 CAN_MOD.RM 置 1。

CAN 可以支持监听模式和自测模式，在 Reset Mode 和 Operating Mode 两种模式下都可使用。

开启监听模式，会使节点强制进入“Error Passive”模式，并且只能用于接收数据，不能发送。CAN 在成功接收到一帧数据后也不会回复 ACK。在这种模式下，可以通过软件的方式来实现总线上的波特率检测，使 CAN 支持 hot plug-in 模式。

在自测试模式下，发送数据时不需要从远程 CAN 节点获取 ACK，从而可以用来接收自己发送的数据，实现自发自收。

CAN 也提供了一种时钟输出模式，但是只能在 Reset Mode 时选择，TX0 用来输出发送数据时的波特率时钟信号而不是数据（参考 CAN_OCR.TX0_SEL）。

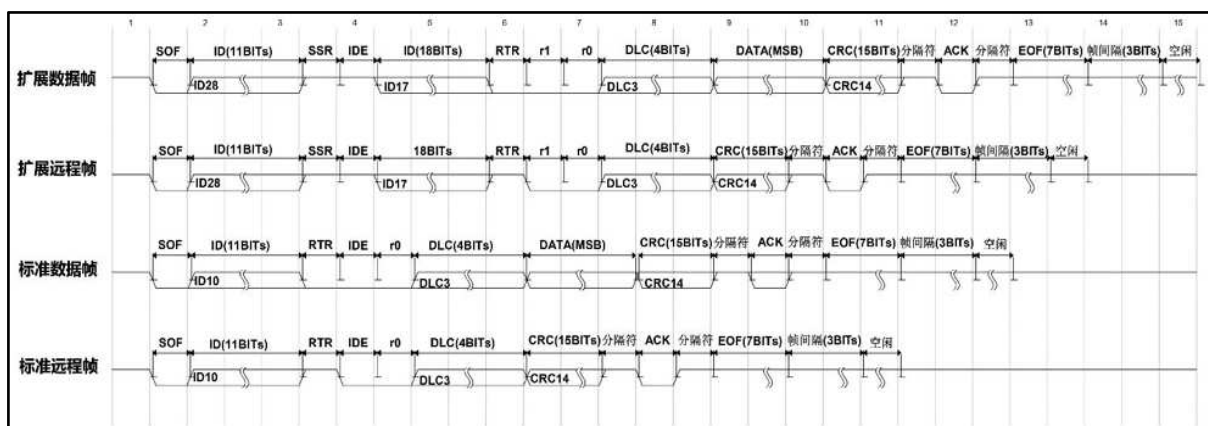
28.3.3. 发送

根据 CAN 数据帧的结构，如图 28-3 所示。将需要发送的帧类型、帧长度、帧数据写入到 CAN_TXBUFF 中，不管数据帧格式是标准帧还是扩展帧，CAN_TXBUFF 由 13 个字节的数据组成，这样可以保证能写入一个数据长度为 8 字节的数据帧。需要注意

的是，在向 CAN_TXBUFF 写数据之前需要先检查一下 CAN_SR.TBS 状态，确保 CAN_TXBUFF 被释放了，否则写入的数据就丢失。

写入到 CAN_TXBUFF（只写）中的帧，可以通过 CAN_TXFIFO（只读）来读取。

图 28-3 CAN 数据帧结构



为了将存放在 CAN_TXBUFF 里面的数据发送出去，可以通过设置 CAN_CMR.TR 发送请求位或者设置 CAN_CMR.SRR 自接收请求位。开始发送时，CAN_SR.TS 被置为 1 并且发送请求（TR 或 SRR）被清除。

发送的比特流是通过 TX0 发送出去的，如果遇到仲裁失败或者发送错误，CAN 能够自动重发。

在每个数据帧后都会自动发送一个 15bits 的 CRC 校验值。CRC 是根据 SOF、仲裁域、控制域和数据域产生。

如果发送还没开始的话，可以通过将 CAN_CMR.AT 置 1 来中断一帧数据的发送，但是一旦开始了，就不能中断发送过程了。

CAN_TXBUFF 的数据结构如图所示，由 13 个字节组成（地址空间是基于 CAN 模块的偏移 0x60~0x90），发送时只需按照 CAN_TXBUFF 结构填入相应的 RTR（远程帧 1/数据帧 0）、DLC（数据长度）、FF（Frame Format）（标准帧 0/扩展帧 1）、ID 以及数据即可。

表 28-1 CAN_TXBUFFx 数据结构

Standard Frame Format (SFF)		Extended Frame Format (EFF)	
CAN Address	Field	CAN Address	Field
60h	TX Frame Information	60h	TX Frame Information

64h	TX Identifier 1	64h	TX Identifier 1
68h	TX Identifier 2	68h	TX Identifier 2
6Ch	TX Data Byte 1	6Ch	TX Identifier 3
70h	TX Data Byte 2	70h	TX Identifier 4
74h	TX Data Byte 3	74h	TX Data Byte 1
78h	TX Data Byte 4	78h	TX Data Byte 2
7Ch	TX Data Byte 5	7Ch	TX Data Byte 3
80h	TX Data Byte 6	80h	TX Data Byte 4
84h	TX Data Byte 7	84h	TX Data Byte 5
88h	TX Data Byte 8	88h	TX Data Byte 6
8Ch	(Unused)	8Ch	TX Data Byte 7
90h	(Unused)	90h	TX Data Byte 8

Transmit Frame (SFF)								
CAN Address	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
60h	FF	RTR	X ₍₁₎	X ₍₁₎	DLC.3	DLC.2	DLC.1	DLC.0
64h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
68h	ID.20	ID.19	ID.18	X ₍₂₎	X ₍₁₎	X ₍₁₎	X ₍₁₎	X ₍₁₎

Transmit Frame (SFF)								
CAN	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

Address								
60h	FF	RTR	X ₍₁₎	X ₍₁₎	DLC.3	DLC.2	DLC.1	DLC.0
64h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
68h	ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13
6Ch	ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5
70h	ID.4	ID.3	ID.2	ID.1	ID.0	X ₍₂₎	X ₍₁₎	X ₍₁₎

注：X₍₁₎ 表示不关心，但建议为 0

X₍₂₎ 表示不关心，但建议和 CAN_RXBUFFx 中的 RTR 位匹配

28.3.4. 接收

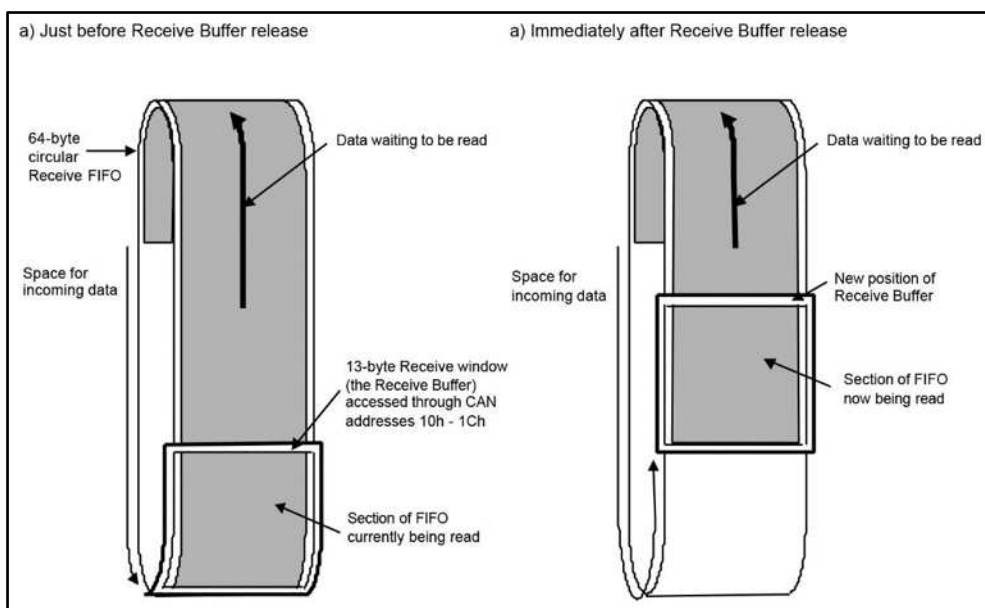
CAN 接收的数据首先通过接收滤波器才能写入到 CAN_RXFIFO。接收滤波器只会通过那些标识符合条件的报文。

CAN 模块正在接收总线上数据时，CAN_SR.RS 位会被置位。一旦正确接收到一帧数据并且通过接收过滤器放入 CAN_RXFIFO 后，CAN_SR.RBS 位就会被置位，同时产生一个接收中断。

CAN_RXFIFO 是 64 字节深度，最多允许存放 5 组 EFF 数据。如果 CAN_RXFIFO 中没有空间接收新的数据，那么新的数据进来的话就会触发一次 Over Run，CAN_SR.DOS (bit1) 位会被置位，接收的数据也会被丢弃，也产生一个中断。

放在 CAN_RXFIFO 中的数据通过一个 13 个字节的窗口 CAN_RXBUFF（地址空间是基于 CAN 模块的偏移 0x60~0x90）来读取。由于 CAN_RXFIFO 中可能存放多组报文，因此为了读取所有的报文，CPU 需要移动窗口，这是通过设置 CAN_CMR.RRB 位来实现的。如果还有数据需要被读取，那么这个数据将会被移动到窗口，如果没有数据被读取了，那么接收标志就会自动清零，接收过程如图所示。

图 28-4 CAN 接收示意图



CAN_RXBUFF 的数据结构如图所示，由 13 个字节组成，接收时只需按照 CAN_RXBUFF 结构读取相应的 RTR（远程帧 1/数据帧 0）、DLC（数据长度）、FF（标准帧 0/扩展帧 1）、ID 以及数据即可。

表 28-2 CAN_RXBUFFx 数据结构

Standard Frame Format (SFF)		Extended Frame Format (EFF)	
CAN Address	Field	CAN Address	Field
60h	RX Frame Information	60h	RX Frame Information
64h	RX Identifier 1	64h	RX Identifier 1
68h	RX Identifier 2	68h	RX Identifier 2
6Ch	RX Data Byte 1	6Ch	RX Identifier 3
70h	RX Data Byte 2	70h	RX Identifier 4
74h	RX Data Byte 3	74h	RX Data Byte 1
78h	RX Data Byte 4	78h	RX Data Byte 2
7Ch	RX Data Byte 5	7Ch	RX Data Byte 3
80h	RX Data Byte 6	80h	RX Data Byte 4

84h	RX Data Byte 7	84h	RX Data Byte 5
88h	RX Data Byte 8	88h	RX Data Byte 6
8Ch	(Unused)	8Ch	RX Data Byte 7
90h	(Unused)	90h	RX Data Byte 8

Transmit Frame (SFF)								
CAN Address	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
60h	FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0
64h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
68h	ID.20	ID.19	ID.18	RTR	0	0	0	0

Transmit Frame (SFF)								
CAN Address	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
60h	FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0
64h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
68h	ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13
6Ch	ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5
70h	ID.4	ID.3	ID.2	ID.1	ID.0	RTR	0	0

28.3.5. 自我接收

CAN 可以接收自己发送给其他节点的数据。通过 CAN_CMRR.SRR 位来使能此功能。CAN 自动产生发送和接收中断。

此功能的作用在于让 CAN 总线可以直接同时接收和发送，而不需要其他节点配合，方便测试。

28.3.6. 接收过滤

在 CAN 总线中，所有的节点接收总线上的所有报文。为了让节点忽略与它无关的报文信息，CAN 允许对接收的报文进行过滤，这是通过 7 组 32 位的接收过滤器来实现的。只有报文的标识符与过滤器匹配时，才会被写入到 CAN_RXFIFO。

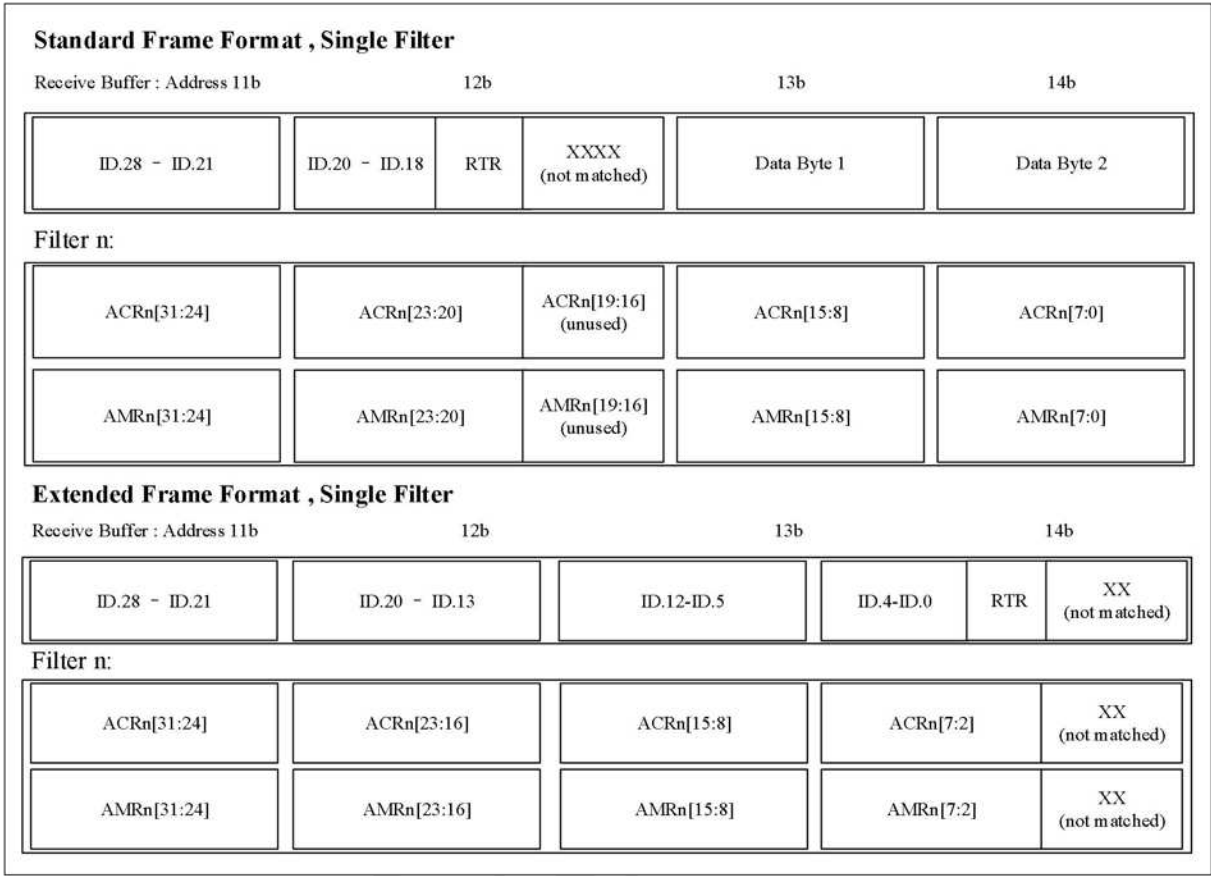
过滤器的实现是通过 CAN_ACRx 和 CAN_AMRx (x 为 0~6) 共同完成的，CAN_ACR0 和 CAN_AMR0 是第一组接收过滤器，CAN_ACR6 和 CAN_AMR6 是第七组接收过滤器。

CAN_ACRx 用于存放需要被接收的总线上的帧 ID 号，而 CAN_AMRx 是一个屏蔽寄存器。可以通过设置 CAN_AMRx 里相应的位来决定是否过滤 CAN_ACRx 中对应的位，当 CAN_AMRx 中的某个位为 1 时，表示“Don't Care”，不对 CAN_ACRx 中相对应的位进行过滤；当 CAN_AMRx 中的位为 0 时，表示对 CAN_ACRx 中相对应的位进行过滤，只有当接收到的 CAN 报文 ID 对应的位与 CAN_ACRx 中对应的位一致，报文才能被接收进来写入到 CAN_RXFIFO。

CAN 一共有两种过滤模式：

- 单过滤模式：CAN_ACRx 和 CAN_AMRx 配合作为一个 32 位的过滤器，过滤一组 ID 和数据，根据标准帧和扩展帧 ID 长度不同，单过滤模式过滤器结构也不同，结构如图所示。

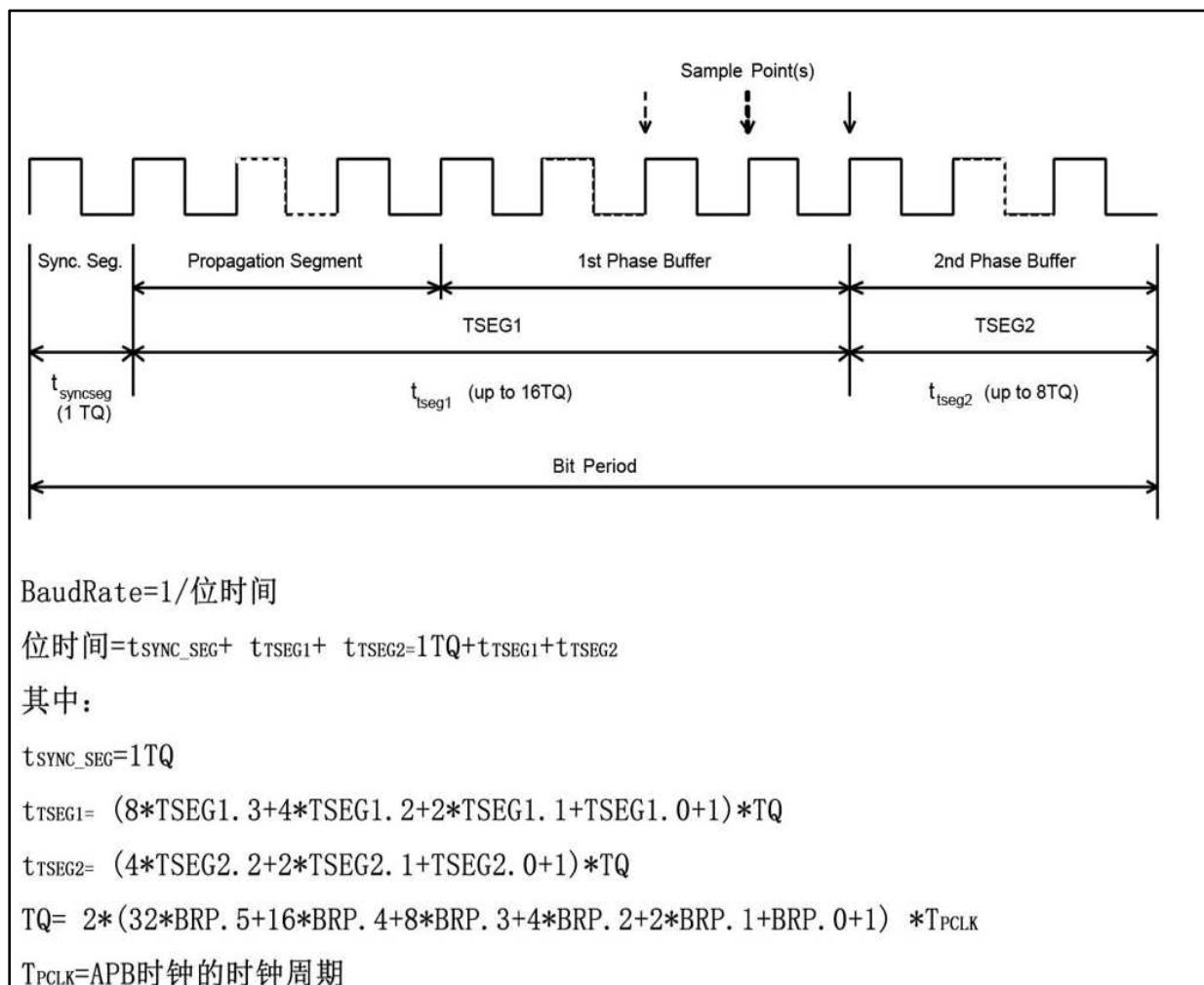
图 28-5 单过滤模式过滤器示意图



- 双过滤模式:CAN_ACRx 和 CAN_AMRx 配合后分别作为两个 16bits 的过滤器，过滤两组不同格式的 ID 和数据。如果使能 2 个过滤器，那么接收的报文只要符合其中一个条件，就可以被接收进来，双过滤模式过滤器结构如图所示。

CAN 模块定义的位时序如图所示，共有三部分组成：

图 28-7 CAN 位时序图



- 同步段(SYNC_SEG)：通常期望位的变化发生在该时间段内，其值固定为 1TQ。
- 时间段 1(TSEG1)：定义采样点的位置。它包含 CAN 标准里的 PROP_SEG 和 PHASE_SEG1。其值可以编程为 1 到 16 个 TQ。
- 时间段 2(TSEG2)：定义发送点的位置。它代表 CAN 标准里的 PHASE_SEG2。其值可以编程为 1 到 8 个 TQ。

重新同步跳跃宽度(SJW)定义了在该位中可以延长或缩短多少个时间单元的上限，其值可以为 1 到 4 个 TQ。

如设置波特率为 500K，系统时钟 64M,APB 时钟为 64M，则 BRP=7，
 $TQ=2*(7+1)*(1/64M)=16*(1/64M)=1/4M$ ，设置 TSEG1=2，TSEG2=3，则数据位时间
 $=1+(2+1)+(3+1)=8TQ$ ，则 $baud=1/8TQ=1/(8*1/4M)=1/(2M)=500Kbps$ 。

28.3.8. 仲裁器

CAN 总线是由小的识别符节点控制的，失去仲裁的节点必须在总线空闲之前放弃控制总线。如果 CAN 仲裁失败，那么一个仲裁失败的中断将会起来，并且会记录失去仲裁的位置。

28.3.9. 错误处理

CAN_ERRCNTR 中包括两个错误计数器，接收错误计数器 RXERR 和发送错误接收器 TXERR。并且错误的类型和错误的位置都可以在 CAN_ECCR（Error-Code-Capture-Register）中看到。CAN_ERRCNTR 中的 EWL 域，其值表示接收或者发送的错误值达到多少时产生一个警告。默认的 EWL 值为 96，不管接收错误还是发送错误达到这个值都会产生一个错误警告中断。

如果错误计数超过 127，那么 CAN 就会进入“Error Passive”状态。如果发送错误计数器超过 255，那么 CAN_SR.BS 就会被置 1（Bus Off）。CAN 就会进入 Reset Mode 并且会产生一个 EI 中断。在重现进入 Bus On 状态前，CAN 必须等待 128 个 Bus-Free-Sequence。

28.3.10. 睡眠模式

如果 Bus 是空闲的，并且没有中断被挂起，那么可以将 CAN 模块进入 Sleep 模式。进入 Sleep 模式的方法是置位 CAN_MOD.SM。TX0 在 Sleep 模式时为高。

可以通过以下方式唤醒：

1. 将 CAN_MOD.SM 置为 0。
2. 在 RX0 上检查的数据。
3. NINT_IN 有一个低电平，NINT_IN 为 CAN 中断信号线，即有任意 CAN 中断触发，将会唤醒 Sleep 模式。

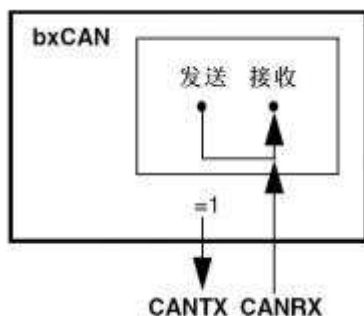
唤醒后，CAN 会产生一个 Wake-Up 中断 WUI。

28.3.11. 测试模式

CAN 模块为了方便自测，增加了静默模式和环回模式，以及两者的组合模式。

- 静默模式

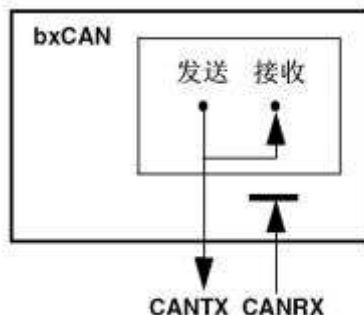
图 28-8 静默模式示意图



在该模式下，CAN 模块不能往总线上发送数据，但是可以接收自己发送的数据（经 CAN 模块内部），也可以接收总线上其他节点发送的数据。

- 环回模式

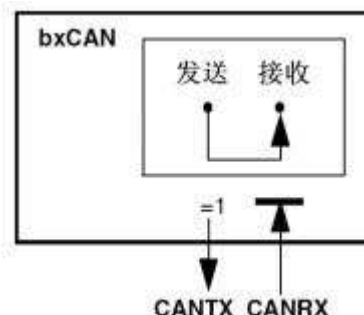
图 28-9 环回模式示意图



在该模式下，CAN 模块可以往总线上发送数据，只能接收自己发送的数据（经 CAN 模块内部），不能接收总线上其他节点发送的数据。

- 环回静默模式

图 28-10 环回静默模式示意图



在该模式下，CAN 模块既不能往总线上发送数据，也不能接收总线上其他节点发送的数据。只能接收自己发送的数据（经 CAN 模块内部）

28.4. 寄存器描述

CAN1 寄存器基地址：0x4000_6400

CAN2 寄存器基地址：0x4000_6800

偏置	名称	正常模式	复位模式	描述
0x00	CAN_MOD	R/W	R/W	模式寄存器
0x04	CAN_CMRR	WO	WO	命令寄存器
0x08	CAN_SR	RO	RO	状态寄存器
0x0C	CAN_IR	RO	RO	中断寄存器
0x10	CAN_IER	R/W	R/W	中断使能寄存器
0x14	CAN_BTR	RO	R/W	总线时序控制寄存器
0x18	CAN_OCR	RO	R/W	输出控制寄存器
0x1C	CAN_ECCR	RO	RO	错误抓取寄存器
0x20	CAN_ERRCNTR	RO	R/W	错误计数寄存器
0x24	CAN_RSR	RO	R/W	接受报文状态寄存器
0x28	CAN_ACR0	RO	R/W	接收过滤寄存器 0
0x2C	CAN_ACR1	RO	R/W	接收过滤寄存器 1
0x30	CAN_ACR2	RO	R/W	接收过滤寄存器 2
0x34	CAN_ACR3	RO	R/W	接收过滤寄存器 3
0x38	CAN_ACR4	RO	R/W	接收过滤寄存器 4
0x3C	CAN_ACR5	RO	R/W	接收过滤寄存器 5
0x40	CAN_ACR6	RO	R/W	接收过滤寄存器 6
0x44	CAN_AMR0	RO	R/W	接收屏蔽寄存器 0
0x48	CAN_AMR1	RO	R/W	接收屏蔽寄存器 1
0x4C	CAN_AMR2	RO	R/W	接收屏蔽寄存器 2
0x50	CAN_AMR3	RO	R/W	接收屏蔽寄存器 3
0x54	CAN_AMR4	RO	R/W	接收屏蔽寄存器 4
0x58	CAN_AMR5	RO	R/W	接收屏蔽寄存器 5
0x5C	CAN_AMR6	RO	R/W	接收屏蔽寄存器 6

0x60~0x90	CAN_TXBUFF	WO	--	发送缓存寄存器（写操作）
0x60~0x90	CAN_RXBUFF	RO	RO	接收缓存寄存器（读操作）
0xA0~0x19C	CAN_RX_FIFO	RO	R/W	接收 FIFO
0x1A0~0x1D0	CAN_TX_FIFO	RO	RO	发送 FIFO

28.4.1. 模式寄存器 CAN_MOD(偏移: 00h)

比特	名称	属性	复位值	描述
31:23	RSV	-	-	保留
22	FACT6	R/W	0	过滤器 6 使能位 0: 禁止过滤器 1: 使能过滤器
21	FACT5	R/W	0	过滤器 5 使能位 0: 禁止过滤器 1: 使能过滤器
20	FACT4	R/W	0	过滤器 4 使能位 0: 禁止过滤器 1: 使能过滤器
19	FACT3	R/W	0	过滤器 3 使能位 0: 禁止过滤器 1: 使能过滤器
18	FACT2	R/W	0	过滤器 2 使能位 0: 禁止过滤器 1: 使能过滤器
17	FACT1	R/W	0	过滤器 1 使能位 0: 禁止过滤器 1: 使能过滤器
16	FACT0	R/W	0	过滤器 0 使能位

				0: 禁止过滤器 1: 使能过滤器
15	RSV	-	-	保留
14	AFM6	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
13	AFM5	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
12	AFM4	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
11	AFM3	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
10	AFM2	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
9	AFM1	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
8	AFM0	R/W	0	接收过滤器模式 0: 双过滤模式，两个 16 位的过滤器。 1: 单过滤模式，单个 32 位的过滤器。
7	RSV	-	-	保留
6	LBKM	R/W	0	环回模式
5	SILM	R/W	0	静默模式
4	SM	R/W	0	睡眠模式 0: 正常状态。

				1: 进入睡眠模式。（如果有中断挂起或者总线在传输数据，则立即会被唤醒） 注：仅能在 Operating 模式下进入。
3	RSV	-	-	保留
2	STM	R/W	0	发送时不检查 ACK 0: 检测 ACK。 1: 无需检测 ACK。
1	LOM	R/W	0	监听模式使能。 0: 正常模式，错误计数器停止工作。 1: 监听模式，会使节点强制进入 Error-passive 模式。CAN 接收到正确的数据帧也不会回发 ACK。
0	RM	R/W	1	进入复位模式 0: 正常模式，此位“1->0”可以进入正常模式。 1: 进入复位模式，并中断正在发送或接收的数据。

28.4.2. 命令寄存器 CAN_CMRR(偏移: 04h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4	SRR	WO	0	自我接收请求 0: 无效。 1: 同步接收自己即将发送的报文。（包含发送功能） 注意：不能和 TR 同时使能，如果 SRR 和 TR 同时使能，SRR 将被忽略。
3	CDO	WO	0	清除过载状态 0: 无效。

				1: 清除过载状态 CAN_SR.DOS。
2	RRB	WO	0	释放接收缓存 0: 无效。 1: 释放接收缓存。【见本章 3.4】
1	AT	WO	0	终止发送 0: 无效 1: 如果数据还未在总线上传输，则取消该发送请求。如果数据已在传输则无法再终止发送。
0	TR	WO	0	发送请求 0: 无效。 1: 发送报文。

28.4.3. 状态寄存器 CAN_SR(偏移: 08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	BS	RO	0	总线状态 0: 总线处于激活状态。 1: 总线处于“Bus Off”状态。
6	ES	RO	0	错误状态 0: 接收和发送错误计数器低于报警值。 1: 接收或者发送错误计数器大于等于报警值。
5	TS	RO	1(Reset) /0(Normal)	发送状态 0: 没有报文正在被发送 1: 正在发送报文
4	RS	RO	1(Reset) /0(Normal)	接收状态 0: 没有报文正在被接收。

				1: 正在接收报文。
3	TCS	RO	1	发送完成状态 0: 最近一次的发送请求还未完成。 1: 最近一次的发送请求已经完成。
2	TBS	RO	1	发送缓存状态 0: 发送缓存锁定状态, 报文正在被发送或者等待被发送, CPU 不能访问发送缓存。 1: 发送缓存释放。CPU 可以访问发送缓存。
1	DOS	RO	0	数据过载状态位。 0: 无数据过载。 1: 数据过载, 因为接收缓存没有空间导致丢失报文。
0	RBS	RO	0	接收缓存状态位 0: 接收缓存为空。 1: 接收缓存不为空。

28.4.4. 中断寄存器 CAN_IR(偏移: 0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	BEI	RO	0	总线错误中断 0: 无中断。 1: 有中断。 注: 读清 0
6	ALI	RO	0	仲裁失败中断 0: 无中断。 1: 有中断。 注: 读清 0
5	EPI	RO	0	Error-Passive 中断

				0: 无中断 1: 有中断 注: 读清 0
4	WUI	RO	0	唤醒中断 0: 无中断。 1: 有中断。 注: 读清 0
3	DOI	RO	0	数据过载中断 0: 无中断。 1: 有中断。 注: 读清 0
2	EI	RO	0	错误报警中断 0: 无中断。 1: 有中断。 注: 读清 0
1	TI	RO	0	发送中断。 0: 无中断。 1: 有中断。 注: 读清 0
0	RI	RO	0	接收中断 0: 无中断。 1: 有中断。 注: 读清 0

28.4.5. 中断使能寄存器 CAN_IER(偏移: 10h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留

7	BEIE	R/W	0	总线错误中断使能 0: 禁止。 1: 使能。
6	ALIE	R/W	0	仲裁失败中断使能 0: 禁止。 1: 使能。
5	EPIE	R/W	0	Error-Passive 中断使能 0: 禁止 1: 使能
4	WUIE	R/W	0	唤醒中断使能 0: 禁止。 1: 使能。
3	DOIE	R/W	0	数据过载中断使能 0: 禁止。 1: 使能。
2	EIE	R/W	0	错误报警中断使能 0: 禁止。 1: 使能。
1	TIE	R/W	0	发送中断使能。 0: 禁止。 1: 使能。
0	RIE	R/W	0	接收中断使能 0: 禁止。 1: 使能。

28.4.6. 时序寄存器 CAN_BTR(偏移: 14h)

比特	名称	正常模式	复位模式	复位值	描述

31:16	RSV	-		-	保留
15:14	SJW	RO	R/W	0	同步偏移宽度。
13:8	BRP	RO	R/W	0	波特率因子 TQ
7	SAM	RO	R/W	0	采样选择 0: 采样一次 1: 采样三次
6:4	TSEG2	RO	R/W	0	采样点尾部 = TSEG2+1
3:0	TSEG1	RO	R/W	0	采样点前部 = TSEG1+1

28.4.7. 输出寄存器 CAN_OCR(偏移: 18h)

比特	名称	正常模式	复位模式	复位值	描述
31:12	RSV	-		-	保留
11	Clock OFF	R/W	R/W	0	控制 TX0_SEL 选择 Clock out 时的开启和关闭 0: 使能 Clock Out 1: 禁止 Clock Out
10:8	CD	R/W	R/W	0	Clock Division。用于 TX0_SEL 选择 Clock out 3'b000: $f_{osc}/2$ 3'b001: $f_{osc}/4$ 3'b010: $f_{osc}/6$ 3'b011: $f_{osc}/8$ 3'b100: $f_{osc}/10$ 3'b101: $f_{osc}/12$ 3'b110: $f_{osc}/14$ 3'b111: 保留
7:6	RSV	-	-	-	保留
5	CRC_ERR_COD	RO	RW	0	CRC 错误发生时产生的 ECC 值

4:2	TX0_SEL	RO	RW	3'b000	3'b000:TX0 3'b001:~TX0 3'b010:Simple Time 3'b011:Tx_clock 3'b100:Clock out 3'b101:0 3'b110:1 3'b111:1
1:0	RSV	-	-	-	保留

28.4.8. 错误代码获取寄存器 CAN_ECCR(偏移: 1Ch)

比特	名称	正常模式	复位模式	复位值	描述
31:13	RSV	-	-	-	保留
12:8	ALC	RO	RO	0	仲裁失败位置值=ALC + 1
7:6	ErrCode	RO	RO	0	0b00: 比特错误 0b01: 形式错误 0b10: 填充错误 0b11: 其他错误
5	Direction	RO	RO	0	出错方向: 0: 发送时出错 1: 接收时出错
4:0	SegCode	RO	RO	0	出错的段代码, 见下表

表 28-3 SegCode 错误码表

SegCode[4:0]	描述	SegCode[4:0]	描述
00011	SOF	01010	数据域
00010	ID.28~ID.21	01000	CRC 域

00110	ID.20~ID.18	11000	CRC 分隔符
00100	SRTR 位	11001	ACK
00101	IDE 位	11011	ACK 分隔符
00111	ID.17~ID.13	11010	EOF
01111	ID.12~ID.5	10010	间隙
01110	ID.4~ID.0	10001	Active Error 标志
01100	RTR 位	10110	Passive Error 标志
01101	保留	10011	容忍显性位
01001	保留	10111	错误分隔符
01011	DLC	11100	过载标志

28.4.9. 错误计数寄存器 CAN_ERRCNTR(偏移: 20h)

比特	名称	正常模式	复位模式	复位值	描述
31:24	RSV	-		-	保留
23:16	EWL	RO	R/W	96	错误报警值
15:8	TXERR	RO	R/W	0	发送错误计数寄存器
7:0	RXERR	RO	R/W	0	接收错误计数寄存器

28.4.10. 接收报文状态寄存器 CAN_RSR(偏移: 0x24)

比特	名称	正常模式	复位模式	复位值	描述
31:14	RSV	--	--	--	保留
13:8	RBSA	RO	R/W	0	RXBUFFx 当前对应于 RXFIFO 的起始地址

7:5	RSV	--	--	--	保留
4:0	RMC	RO	RO	0	接收 FIFO 中已接收的报文个数

28.4.11. 接收过滤寄存器 CAN_ACR0(偏移: 0x28)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR0	--	R/W	0	接收过滤

28.4.12. 接收过滤寄存器 CAN_ACR1(偏移: 0x2C)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR1	--	R/W	0	接收过滤

28.4.13. 接收过滤寄存器 CAN_ACR2(偏移: 0x30)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR2	--	R/W	0	接收过滤

28.4.14. 接收过滤寄存器 CAN_ACR3(偏移: 0x34)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR3	--	R/W	0	接收过滤

28.4.15. 接收过滤寄存器 CAN_ACR4(偏移: 0x38)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR4	--	R/W	0	接收过滤

28.4.16. 接收过滤寄存器 CAN_ACR5(偏移: 0x3C)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR5	--	R/W	0	接收过滤

28.4.17. 接收过滤寄存器 CAN_ACR6(偏移: 0x40)

比特	名称	正常模式	复位模式	复位值	描述
31:0	ACR6	--	R/W	0	接收过滤

28.4.18. 接收屏蔽寄存器 CAN_AMR0(偏移: 0x44)

比特	名称	正常模式	复位模式	复位值	描述
31:0	AMR0	--	R/W	0	接收过滤屏蔽

28.4.19. 接收屏蔽寄存器 CAN_AMR1(偏移: 0x48)

比特	名称	正常模式	复位模式	复位值	描述
31:0	AMR1	--	R/W	0	接收过滤屏蔽

28.4.20. 接收屏蔽寄存器 CAN_AMR2(偏移: 0x4C)

比特	名称	正常模式	复位模式	复位值	描述
31:0	AMR2	--	R/W	0	接收过滤屏蔽

28.4.21. 接收屏蔽寄存器 CAN_AMR3(偏移: 0x50)

比特	名称	正常模式	复位模式	复位值	描述

31:0	AMR3	--	R/W	0	接收过滤屏蔽

28.4.22. 接收屏蔽寄存器 CAN_AMR4(偏移: 0x54)

比特	名称	正常模式	复位模式	复位值	描述
31:0	AMR4	--	R/W	0	接收过滤屏蔽

28.4.23. 接收屏蔽寄存器 CAN_AMR5(偏移: 0x58)

比特	名称	正常模式	复位模式	复位值	描述
31:0	AMR5	--	R/W	0	接收过滤屏蔽

28.4.24. 接收屏蔽寄存器 CAN_AMR6(偏移: 0x5C)

比特	名称	正常模式	复位模式	复位值	描述
31:0	AMR6	--	R/W	0	接收过滤屏蔽

28.4.25. 发送缓存写寄存器 CAN_TXBUFFx(写偏移: 0x60~0x90)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:0	TXBUFFx	WO	--	0	发送缓存的字节 x

28.4.26. 接收缓存读寄存器 CAN_RXBUFFx(读偏移: 0x60~0x90)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留

7:0	RXBUFx	RO	--	0	接收缓存的字节 x

28.4.27. 接收 FIFO 访问寄存器 CAN_RXFIFO(偏移: 0xA0~0x19C)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	--	--	--	保留
7:0	RXFIFOx	RO	R/W	0	接收 FIFO 字节 x (0~63)

28.4.28. 发送 FIFO 访问寄存器 CAN_TXFIFO(偏移: 0x1A0~0x1D0)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	--	--	-	保留
7:0	TXFIFOx	RO	RO	0	发送 FIFO 字节 x (0~12)

28.5. 使用流程

28.5.1. CAN 发送和接收

1、初始阶段

- 初始化 CAN 引脚，使能 CAN 模块时钟，复位 CAN 模块；
- 配置 CAN_MOD 寄存器，进入复位模式；
- 配置 CAN 通讯速率，通过时序寄存器 CAN_BTR0 和 CAN_BTR1 设置；
- 配置接收过滤模式和规则，通过 CAN_ACRx 和 CAN_AMRx 设置过滤规则，通过 CAN_MOD 设置过滤模式；
- 配置使能/禁止相应中断；
- 配置 CAN_MOD 寄存器，进入正常模式；

2、发送阶段

- 检查状态寄存器 CAN_SR 中发送状态；

- 将要发送的数据（包括长度、ID、Data）写入到发送缓存写寄存器 CAN_TXBUFF 中；
- 使能命令寄存器 CAN_CMRR 中的发送请求，开始发送数据；
- 等待状态寄存器 CAN_SR 中的发送完成置位，发送完成；

3、接收阶段

- 检查状态寄存器 CAN_SR 中接收缓存状态是否非空；
- 读取接收缓存读寄存器 CAN_RXBUFF 中的数据（包括长度、ID、Data），接收完成；
- 配置命令寄存器 CAN_CMRR，释放接收缓存，等待下一次接收；

29. 数模转换器（DAC）

29.1. 概述

数字/模拟转换器（DAC）可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用 8 位或 12 位模式,左对齐(12 位)或右对齐模式。当使能了外部触发,DMA 可被用于更新输入端数字数据。DAC 模块有 2 个输出通道,每个通道都有单独的转换器。在双 DAC 模式下,2 个通道可以独立地进行转换,也可以同时进行转换并同步地更新 2 个通道的输出。

DAC 输出到 PAD 可以断开,内部连接到芯片内其它模拟外设。在输出电压时,可以利用 DAC 输出 BUFFER 来获得更高的驱动能力。DAC 输出支持在低功耗模式下的采样保持模式。

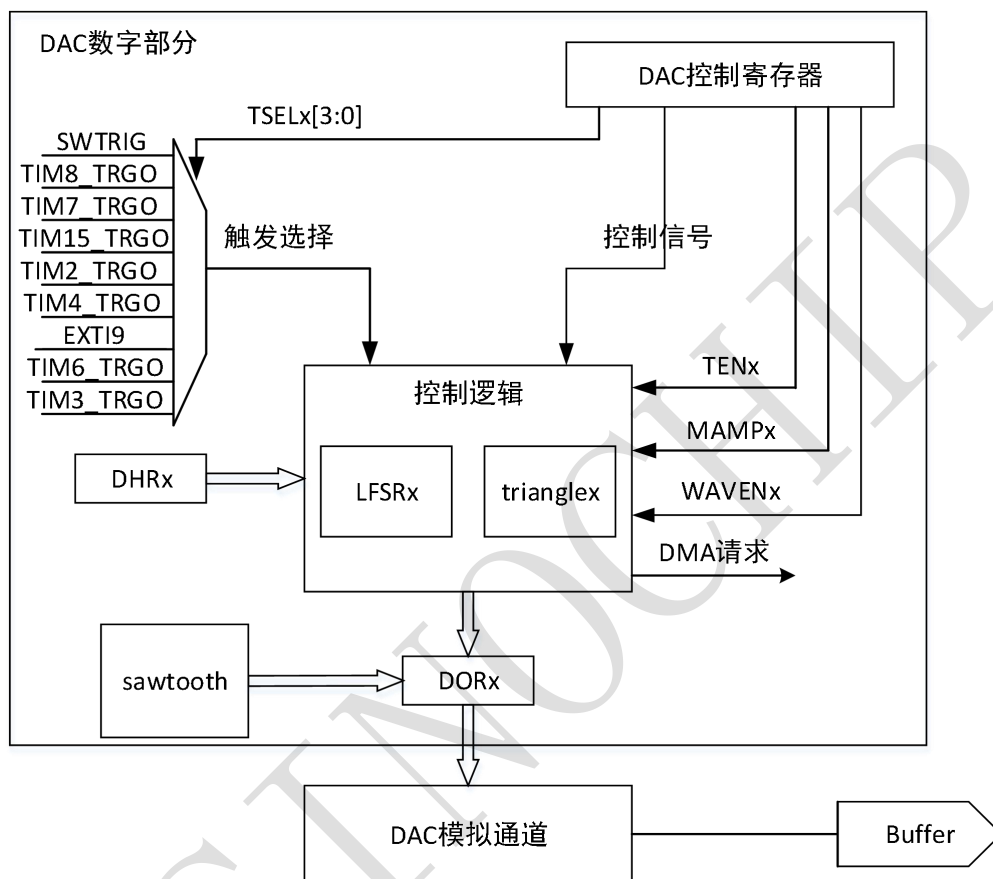
29.2. 主要特性

- 12 位模式下数据左对齐或右对齐
- 每个转换器对应 1 个输出通道
- 8 位或者 12 位分辨率
- 支持有符号数输入
- 噪声波形生成
- 三角波形生成
- 锯齿波形生成
- 双 DAC 通道同时或者分别转换
- DMA 双数据模式降低总线开销
- 每个通道都有 DMA 功能
- 外部触发转换
- 输出 BUFFER 可选, BUFFER 偏差可校准
- 每个通道输出到 PAD 可以断开
- STOP 模式支持采样保存功能
- 输入参考电压 VREFP

29.3. 结构框图

下图为 DAC 控制模块的框图

图 29-1 DAC 结构框图



29.4. 功能描述

29.4.1. DAC 通道使能

将 DAC_CR 寄存器的 ENx 位置'1'即可打开对 DAC 模拟通道 x 的供电。经过一段启动时间 tWAKEUP，DAC 通道 x 即被使能。注意：ENx 位只会使能 DAC 通道 x 的模拟部分，即便该位被置'0'，DAC 通道 x 的数字部分仍然工作。

为了降低输出阻抗，并在没有外部运算放大器的情况下驱动外部负载，每个 DAC 模拟通道内部各自集成了一个输出 BUFFER。可以通过设置 DAC 的寄存器 DAC_MCR 开启或者关闭输出 BUFFER，默认为开启状态。

29.4.2. DAC 数据结构

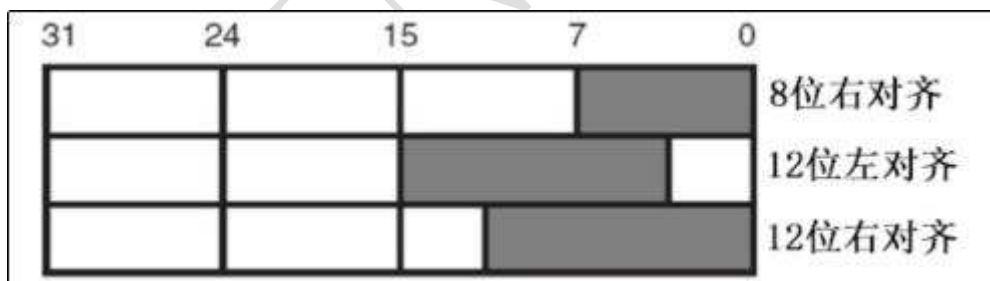
根据选择的配置模式，数据按照下文所述写入指定的寄存器：

单 DAC 通道 x，有 3 种情况：

- 8 位数据右对齐：用户须将数据写入寄存器 DAC_DHR8Rx[7:0]位(实际是存入寄存器 DHRx[11:4]位，低位强制为 0)
- 12 位数据左对齐：用户须将数据写入寄存器 DAC_DHR12Lx[15:4]位(实际是存入寄存器 DHRx[11:0]位)
- 12 位数据右对齐：用户须将数据写入寄存器 DAC_DHR12Rx[11:0]位(实际是存入寄存器 DHRx[11:0]位)

根据对 DAC_DHRyyyx 寄存器的操作，经过相应的移位后，写入的数据被转存到 DHRx 寄存器中(DHRx 是内部的数据保存寄存器 x)。随后 DHRx 寄存器的内容或被自动地传送到 DORx 寄存器，或通过软件触发或外部事件触发被传送到 DORx 寄存器。

图 29-2 DAC 通道模式的数据寄存器



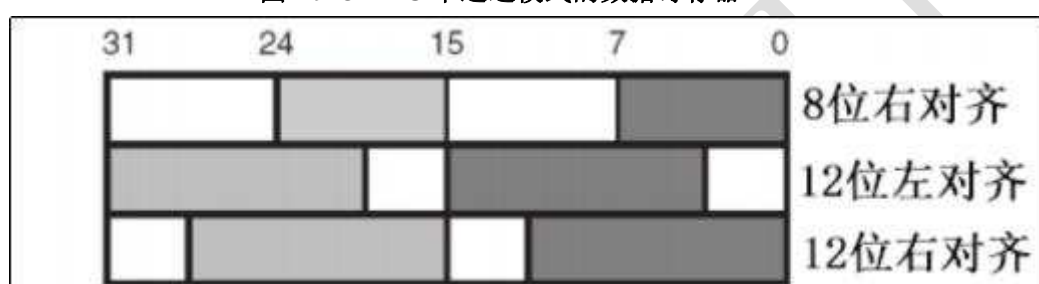
双 DAC 通道，有 3 种情况：

- 8 位数据右对齐：用户须将 DAC 通道 1 数据写入寄存器 DAC_DHR8RD[7:0]位(实际是存入寄存器 DHR1[11:4]位，低位强制为 0)，将 DAC 通道 2 数据写入寄存器 DAC_DHR8RD[23:16]位(实际是存入寄存器 DHR2[11:4]位，低位强制为 0)
- 12 位数据左对齐：用户须将 DAC 通道 1 数据写入寄存 DAC_DHR12LD[15:4]位(实际是存入寄存器 DHR1[11:0]位)，将 DAC 通道 2 数据写入寄存器 DAC_DHR12LD[31:20]位(实际是存入寄存器 DHR2[11:0]位)

- 12 位数据右对齐：用户须将 DAC 通道 1 数据写入寄存 DAC_DHR12RD[11:0] 位(实际是存入寄存器 DHR1[11:0]位)，将 DAC 通道 2 数据写入寄存器 DAC_DHR12RD[27:16]位(实际是存入寄存器 DHR2[11:0]位)

根据对 DAC_DHRyyyD 寄存器的操作，经过相应的移位后，写入的数据被转存到 DHR1 和 DHR2 寄存器中(DHR1 和 DHR2 是内部的数据保存寄存器 x)。随后，DHR1 和 DHR2 的内容或被自动地传送到 DORx 寄存器，或通过软件触发或外部事件触发被传送到 DORx 寄存器。

图 29-3 DAC 单通道模式的数据寄存器



- 有符号/无符号数

DAC 输入支持有符号数或无符号数格式。当输入为无符号数 12bit 模式时，0x000 表示电压最小值，0xFFF 代表电压最大值。

DAC 输入支持 2's 补码形式的有符号数,通过设置 DAC_MCR 寄存器 SINFORMATx 位选择有符号数。

在有符号数模式，写入 DHRx 寄存器的数据传送到 DAC_DORx 寄存器时，最高位（MSB）会取反，DAC 的输入数据支持 Q1.15、Q1.11 或者 Q1.7 格式的有符号数。DAC_DHR12Lx 寄存器可以用来存放 16 位有符号数。16 位中的最高 12 位用于 DAC 输出数据（DAC_DORx），其中最高位会被取反。16 位中的最低 4 位会被忽略。

表 29-1 Data format (12 位模式)

SINFORMATx bit	写入 DHRx 寄存器数据	传送到 DORx 寄存器数据
0	0x000	0x000
0	0xFFF	0xFFF
1	0x7FF	0xFFF
1	0x000	0x800
1	0xFFF	0x7FF

1	0x800	0x000

29.4.3. DAC 转换和输出电压

不能直接对寄存器 DAC_DORx 写入数据，任何输出到 DAC 通道 x 的数据都必须写入 DAC_DHRx 寄存器（数据实际写入 DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12LD、或者 DAC_DHR12RD 寄存器）。

如果没有选中硬件触发（寄存器 DAC_CR1 的 TENx 位置 '0'），存入寄存器 DAC_DHRx 的数据会自动传至寄存器 DAC_DORx。如果选中硬件触发（寄存器 DAC_CR1 的 TENx 位置 '1'），在触发发生以后 DAC 保持数据（DAC_DHRx）会被转移到 DAC 数据输出寄存器（DAC_DORx）。

一旦数据从 DAC_DHRx 寄存器装入 DAC_DORx 寄存器，在经过时间 tSETTLING 之后，输出即有效，这段时间的长短依电源电压和模拟输出负载的不同会有所变化。

数字输入经过 DAC 被线性地转换为模拟电压输出，其范围为 0 到 VREF+。任一 DAC 通道引脚上的输出电压满足下面的关系：

$$\text{DAC 输出} = \text{VREF} * (\text{DOR} / 4095)。$$

29.4.4. DAC 触发选择

如果 TENx 位被置 1，DAC 转换可以由某外部事件触发（定时器计数器、外部管脚信号）。配置控制位 TSELx[3:0] 可以选择 16 个触发事件之一触发 DAC 转换。

表 29-2 DAC 触发源选择表

触发源	类型	TSELx[3:0]
SWTRIG	软件触发位	0000
TIM8_TRGO	内部定时器的触发信号	0001
TIM7_TRGO		0010
TIM15_TRGO		0011
TIM2_TRGO		0100
TIM4_TRGO		0101
EXTI9	外部管脚	0110

TIM6_TRGO	内部定时器的触发信号	0111
TIM3_TRGO		1000

当 DAC 通道检测到来自选中的定时器 TRGO 或者 EXTI9 触发信号，存放在寄存器 DAC_DHRx 中的数据会被传送到寄存器 DAC_DORx 中。

如果选择软件触发，一旦 SWTRIG 位置'1'，转换即开始。在数据从 DAC_DHRx 寄存器传送到 DAC_DORx 寄存器后， SWTRIG 位由硬件自动清'0'。

锯齿波生成的复位触发选择和递增触发选择分别通过 STRSTTRIGSELx 和 STINCTRICSELx 实现。其中复位触发选择 STRSTTRIGSELx 与 TSELx 一致，见 DAC 转换触发选择。递增触发选择 STRINCTRIGSELx 见下表。

表 29-3 递增触发选择

触发源	类型	STINCTRIGSELx[3:0]
SWTRIGB	软件触发位	0000
TIM8_TRGO	内部定时器的触发信号	0001
TIM7_TRGO		0010
TIM15_TRGO		0011
TIM2_TRGO		0100
TIM4_TRGO		0101
EXTI10	外部管脚	0110
TIM6_TRGO	内部定时器的触发信号	0111
TIM3_TRGO		1000

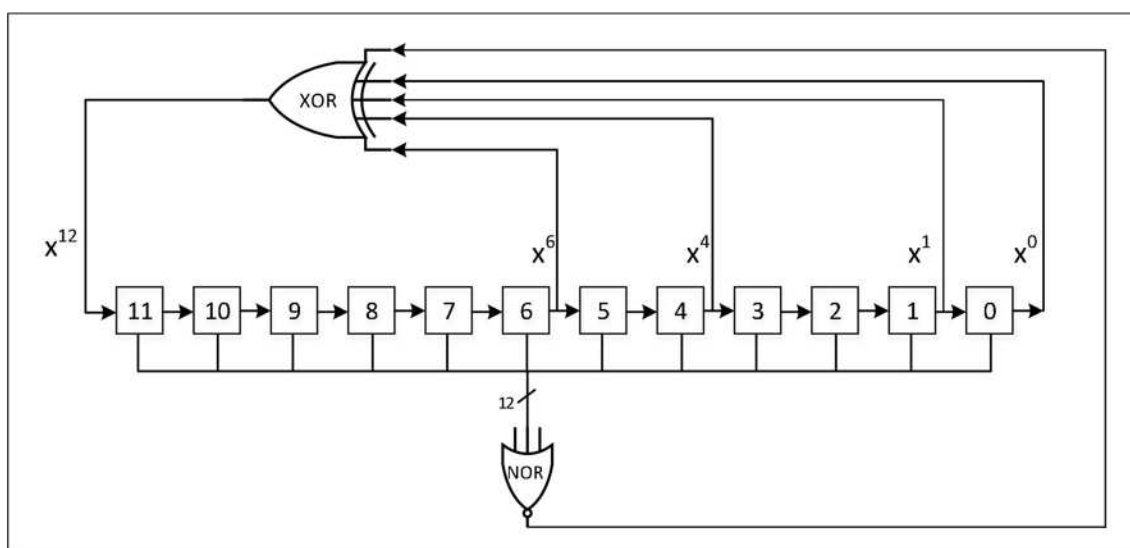
29.4.5. DAC 噪声叠加

DAC 通道有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通过 DAC_CR 寄存器的 WAVEx 位来进行选择。噪声的幅值可以通过配置 DAC_CR 寄存器的 DAC 噪声波位宽（MAMPx）位来进行设置。

- LFSR 噪声模式：设置 WAVE[1:0]位为'01'选择 LFSR 噪声生成功能。在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。寄存器 LFSR 的预装入值为

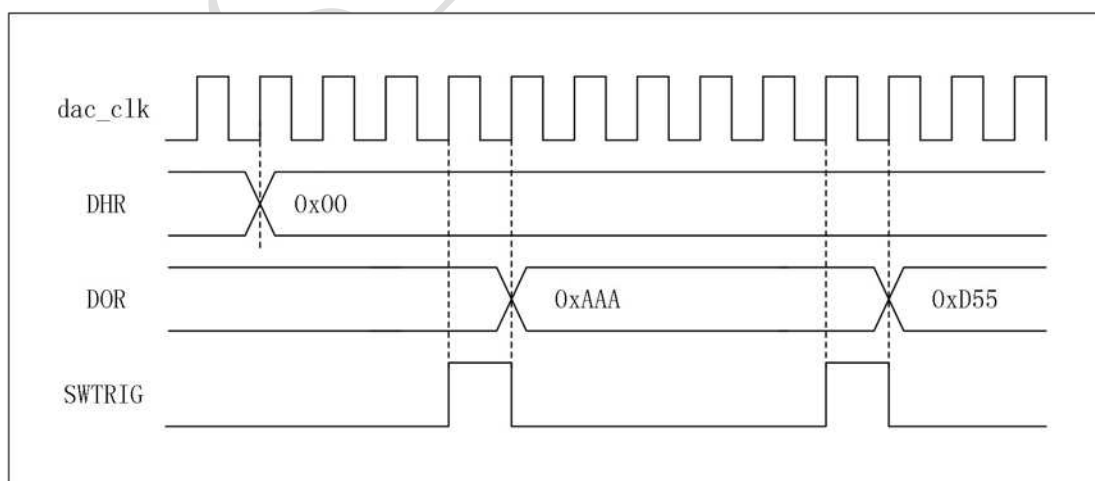
0xAAA。按照特定算法，在每次触发事件后更新该寄存器的值。

图 29-4 LFSR 噪声功能框图



设置 DAC_CR 寄存器的 MAMPx[3:0] 位可以屏蔽部分或者全部 LFSR 的数据，这样得到的 LFSR 值与 DAC_DHRx 的数值相加，去掉溢出位之后即被写入 DAC_DORx 寄存器。将 WAVEx[1:0] 位置 '0' 可以复位 LFSR 波形的生成算法。为了产生噪声，必须使能 DAC 触发，即设 DAC_CR 寄存器的 TENx 位为 '1'。

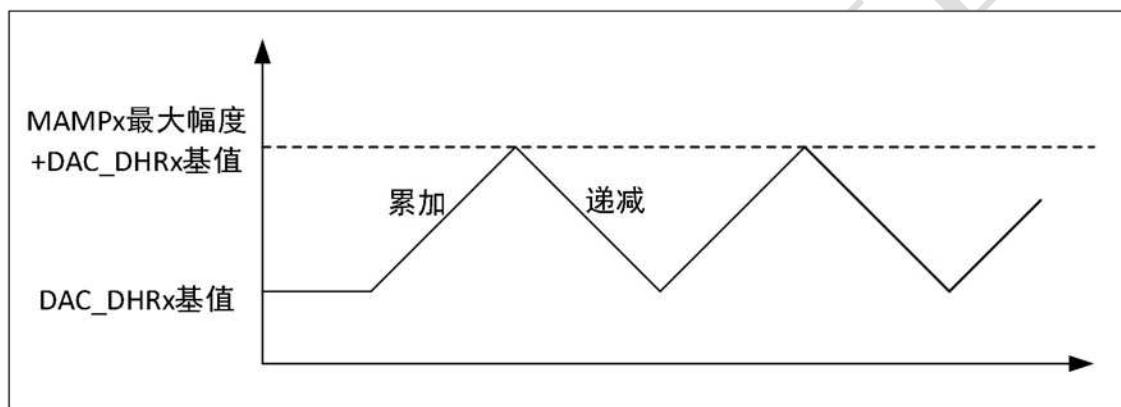
图 29-5 带 LFSR 噪声模的 DAC 转换波形



- 三角噪声模式：可以在 DC 或者缓慢变化的信号上加一个小幅度的三角波噪声。设置 WAVEx[1:0] 位为 '10' 选择 DAC 的三角波生成功能。设置 DAC_CR 寄

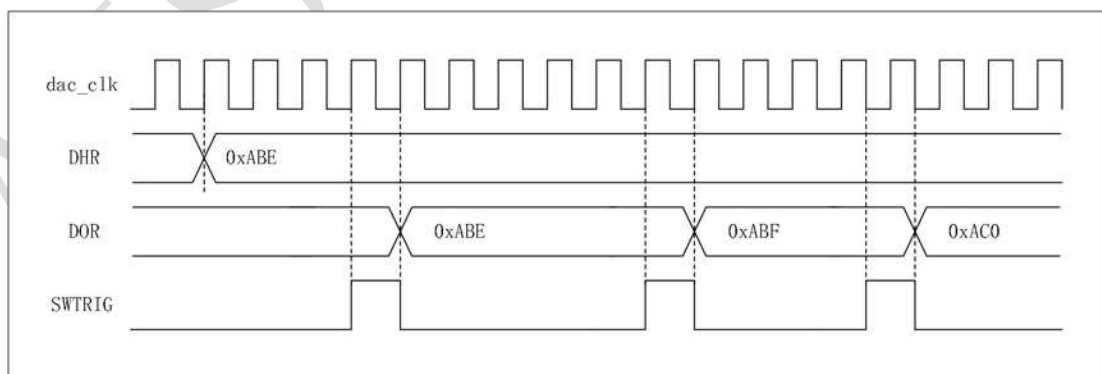
寄存器的 MAMPx[3:0]位来选择三角波的幅度。内部的三角波计数器每次触发事件之后累加 1。计数器的值与 DAC_DHRx 寄存器的数值相加并丢弃溢出位后写入 DAC_DORx 寄存器。在传入 DAC_DORx 寄存器的数值小于 MAMP[3:0]位定义的最大幅度时，三角波计数器逐步累加。一旦达到设置的最大幅度，则计数器开始递减，达到 0 后再开始累加，周而复始。

图 29-6 三角波噪声功能框图



将 WAVEx[1:0]位置'0'可以复位三角波的生成。为了产生噪声，必须使能 DAC 触发，即设 DAC_CR 寄存器的 TENx 位为'1'。

图 29-7 带三角波噪声的 DAC 转换波形



29.4.6. DAC 锯齿波生成

DAC 可以产生锯齿波，通过设置锯齿波的初始值、递增值和方向可以通过相关寄存器设置：

- 设置 WAVE[1:0]位为'11'使能锯齿波功能
- 锯齿波计数器（16 位）的初始值（复位值）通过 DAC_STRx 寄存器的 STRSTDATAx[11:0]设置
- 锯齿波计数器的递增值通过 DAC_STRx 寄存器的 STINCDATAx[15:0]设置
- 锯齿波计数器的方向通过 DAC_STRx 寄存器的 STDIRx 位设置
- 锯齿波计数器从 STRSTDATAx[11:0]开始计数（初始值计数器的高 12 位），当递增触发信号有效，计数器按照 STINCDATAx[15:0]的值递增或递减（12.4 位，低 4 位为小数部分）。DAC 转换使用锯齿波计数器的高 12 位。当计数器计数到 0x0000 或者 0xFFFF，计数值不再变化。当复位触发信号有效，计数器回到初始值 STRSTDATAx[11:0]（低 4 位为 0）。

图 29-8 递减锯齿波（STDIR=0）

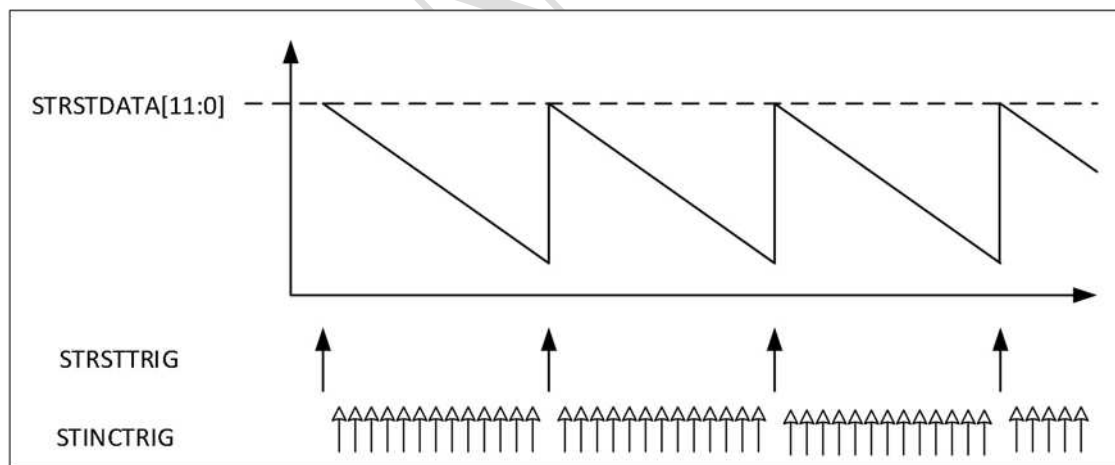
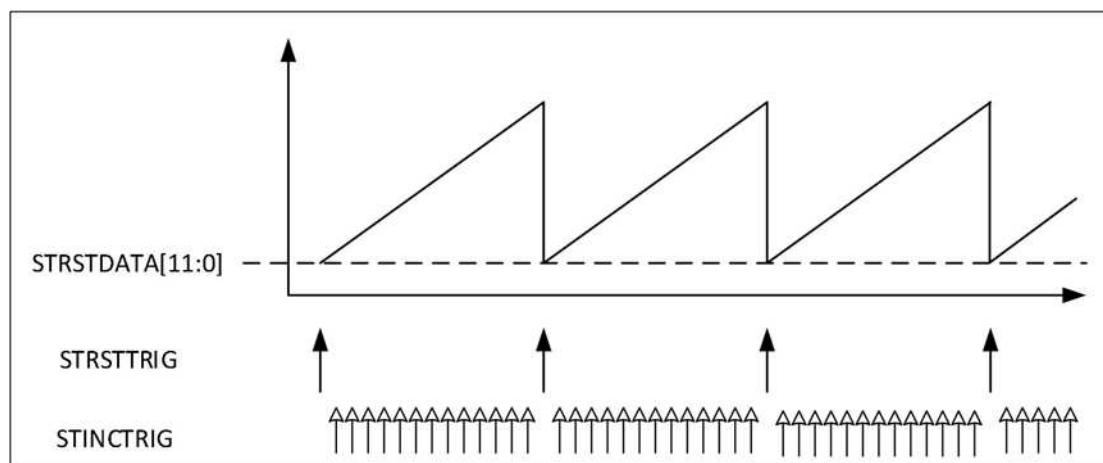


图 29-9 递增锯齿波 (STDIR=1)



递增触发信号和复位触发信号通过 $STINCTRIGSELx[3:0]$ 和 $STRSTTRIGSELx[3:0]$ 选择。参见 DAC 触发选择章节。STRSTTRIG 比 STINCTRIG 有更高的优先级。锯齿波模式触发不受 $TENx$ 控制

29.4.7. DMA 请求

每个 DAC 通道都具有 DMA 功能。2 个 DMA 通道可分别用于 2 个 DAC 通道的 DMA 请求。如果 $DMAENx$ 位置 '1'，一旦有外部触发(而不是软件触发)发生，则产生一个 DMA 请求，然后 DAC_DHRx 寄存器的数据被传送到 DAC_DORx 寄存器，随后产生一个 DMA 请求。

在双 DAC 模式下，如果 2 个通道的 $DMAENx$ 位都为 '1'，则会产生 2 个 DMA 请求。如果实际只需要一个 DMA 传输，则应只选择其中一个 $DMAENx$ 位置 '1'。这样，程序可以在只使用一个 DMA 请求，一个 DMA 通道的情况下，处理工作在双 DAC 模式的 2 个 DAC 通道。

因为数据从 DAC_DHRx 转移到 DAC_DORx 之后才产生 DMA 请求，因此第一笔数据必须在第一次触发信号有效前写入 DAC_DHRx 寄存器。

DAC 的 DMA 请求不会累计，因此如果第 2 个外部触发发生在响应第 1 个外部触发的 DMA 请求之前，则不能产生第 2 个 DMA 请求，但会产生 DMA underrun 标志 $DMAUDRx$ (DAC_SR 寄存器)。

- DMA 双数据模式

在普通的模式下，一个 DMA 请求只传输 12 位（或 8 位）有效数据。AHB 总线位宽是 32 位的，理论上一次传输可以同时写入两个 12 位数据。设置 DAC_MCR 寄存器的 DMADOUBLEx 位可以使能一次 DMA 请求传输两个 12 位数据模式，即 DMA 双数据模式。

当使能 DMA 双数据模式，每两次外部触发产生一次 DMA 请求。

1. 当检测到第一次外部触发，DAC_DHRx 和 DAC_DHRBx 的数据会传送到 DAC_DORx 和 DAC_DORBx 寄存器。当前有效的 DAC 数据存放在 DAC_DORx 寄存器中。随后，产生一次 DMA 请求。DMA 写入新的数据到 DAC_DHRx 和 DAC_DHRBx 中。

2. 当检测到第二次外部触发，当前有效的 DAC 数据存放在 DAC_DORBx 中（通过第一次触发从 DAC_DHRBx 传送），不会产生新的 DMA 请求。状态寄存器中的 DORSTATx 位表明当前 DOR 中哪个数据作用于模拟 DAC 转换。第二次触发会把 DORBx 数据搬运到 DORx 寄存器输出；如果设置了噪声叠加，数据从 DORBx 搬运到 DORx 时，会叠加上噪声。

DMA underrun 功能在 DMA 双数据模式同样有效。

DMA 双数据模式，DMA 请求只能用于单 DAC 通道。如果两个通道都需要使能双数据模式，每个 DMA 通道需要单独设置。

在 DMA 单数据模式和双数据模式切换，必须关闭 DAC 模块和 DMA 功能（ENx=0 和 DMAENx=0）。

29.4.8. DAC 并发转换

当两个 DAC 同时工作时，为了在特定应用中最大限度利用总线带宽，两个 DAC 可以被配置为并发模式。在并发模式中，两个 DAC 的 DAC_DORx 的值将同时被更新。

有 3 个寄存器可被用于加载 DAC_DHRx 的值，分别是：DHR8RD、DHR12RD 和 DHR12LD，只需要访问一个寄存器即可完成同时驱动 2 个 DAC 通道的操作。

当使能了外部触发时，两个 DAC 通道的 TENx 位都应被置位。和 TSEL1 位应被配置为相同的值。当需要使能 DMA 功能时，某一个 DAC 通道的 DMAENx 位被置位即可。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

除了并发工作外，每个 DAC 通道可以设置成独立的工作模式，互不干扰，使用自己的通道寄存器。

29.4.9. 采样保持模式

在采样保持模式，DAC 按照触发完成一次转换，并把转换电压保持在通道输出的电容上。当不在转换时，DAC 的模拟和 BUFFER 会关闭，会降低整体功耗。在每次转换过程，有一个采样稳定时间（sample）需要等待。在保持阶段电容会慢慢漏电，电压会下降。因此经过一定的保持时间（hold），需要进行刷新充电操作（Refresh）。

在此模式下，sample、hold 和 Refresh 的计数都由 RC32K 完成。

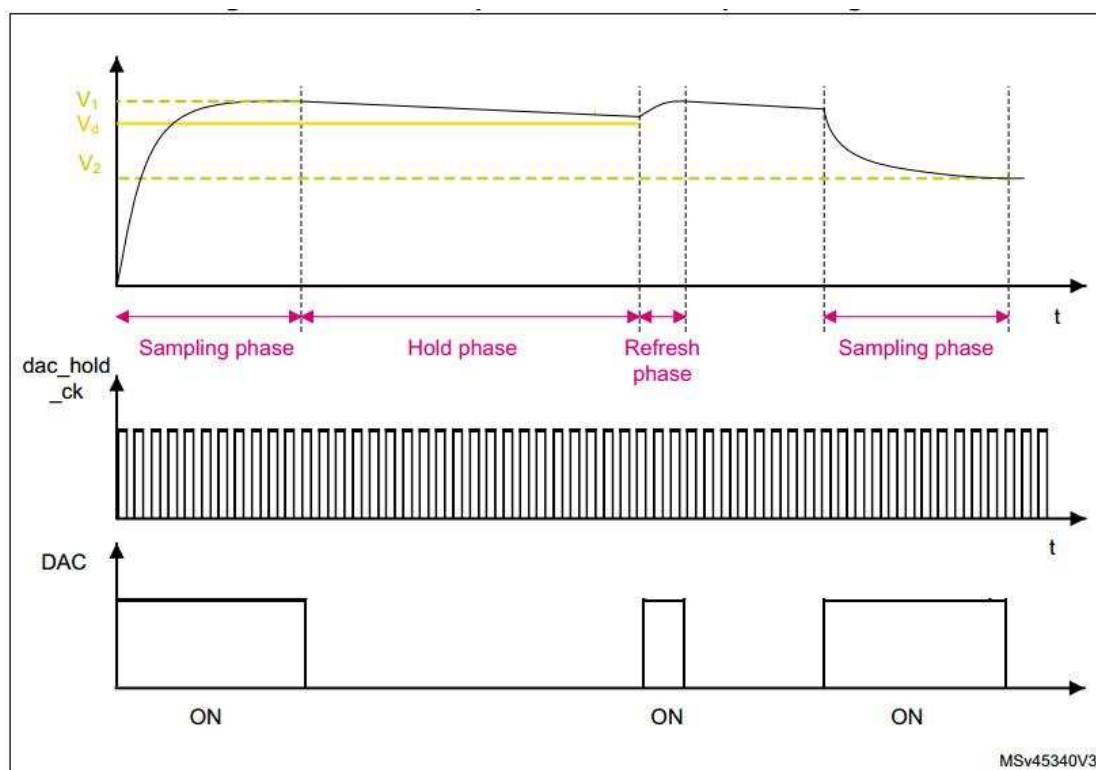
采样保持模式总共分为三个阶段：

- 1.采样阶段。在转换数据变化后，需要把 DAC 输出充电到目标电压，具体时间取决于电容大小。采样（sample）时间可以通过 DAC_SHSRx 寄存器的 TSAMPLEx[9:0]设置。
- 2.保持阶段。DAC 输出处于高阻状态，DAC 模拟部分和 BUFFER 处于关闭状态，用于降低功耗。保持（hold）时间通过 DAC_SHHR 寄存器的 THOLDx[9:0]位设置。
- 3. 刷新（refresh）阶段。对 DAC 输出通道再次充电。刷新（refresh）时间通过 DAC_SHRR 寄存器的 TREFRESHx[7:0]设置。

上述三个阶段时间采用 RC32K 计数。例如 sample 时间为 350us，hold 时间为 2ms，refresh 时间为 100us。则 sample 需要 12 个周期，TSAMPLEx[9:0] = 11；hold 需要 62 个周期，THOLDx[9:0] = 62；refresh 需要 4 个周期，TREFRESHx[7:0] = 4。

这个过程如下图：

图 29-10 采样保持模式时序图



29.5. 寄存器描述

DAC 寄存器基地址：0x5000_0800

偏置	名称	描述
0x00	DAC_CR	DAC 控制寄存器
0x04	DAC_SWTRIGR	DAC 软件触发寄存器
0x08	DAC_DHR12R1	DAC 通道 1 12 位右对齐数据保持寄存器
0x0C	DAC_DHR12L1	DAC 通道 1 12 位左对齐数据保持寄存器
0x10	DAC_DHR8R1	DAC 通道 1 8 位右对齐数据保持寄存器
0x14	DAC_DHR12R2	DAC 通道 2 12 位右对齐数据保持寄存器
0x18	DAC_DHR12L2	DAC 通道 2 12 位左对齐数据保持寄存器
0x1C	DAC_DHR8R2	DAC 通道 2 8 位右对齐数据保持寄存器
0x20	DAC_DHR12RD	双 DAC 12 位右对齐数据保持寄存器
0x24	DAC_DHR12LD	双 DAC 12 位左对齐数据保持寄存器
0x28	DAC_DHR8RD	双 DAC 8 位右对齐数据保持寄存器
0x2C	DAC_DOR1	DAC 通道 1 数据输出寄存器
0x30	DAC_DOR2	DAC 通道 2 数据输出寄存器
0x34	DAC_SR	DAC 状态寄存器
0x38	DAC_CCR	DAC 校准控制寄存器
0x3C	DAC_MCR	DAC 模式控制寄存器
0x40	DAC_SHSR1	DAC 通道 1 采样时间寄存器
0x44	DAC_SHSR2	DAC 通道 2 采样时间寄存器
0x48	DAC_SHHR	DAC 保持时间寄存器
0x4C	DAC_SHRR	DAC 刷新时间寄存器
0x58	DAC_STR1	DAC 通道 1 锯齿波寄存器
0x5C	DAC_STR2	DAC 通道 2 锯齿波寄存器
0x60	DAC_STMODR	DAC 锯齿波模式寄存器

29.5.1. DAC 控制寄存器(DAC_CR 偏移 00h)

比特	名称	属性	复位值	描述
31	RSV	-	-	保留
30	CEN2	RW	0	DAC 通道 2 calibration 使能
29	DMAUDIE2	RW	0	DAC 通道 2 Underrun 错误中断使能 0: 关闭 DAC 通道 2 Underrun 错误中断 1: 使能 DAC 通道 2 Underrun 错误中断
28	DMAEN2	RW	0	DAC 通道 2 DMA 使能 0: 关闭 DAC 通道 2 DMA 模式; 1: 使能 DAC 通道 2 DMA 模式
27:24	MAMP2[3:0]	RW	0	DAC 通道 2 屏蔽/幅值选择器, 用来在噪声生成模式下选择屏蔽位, 在三角波生成模式下选择波形的幅值。 0000: 不屏蔽 LSFR 位[0] / 三角波幅值等于 1; 0001: 不屏蔽 LSFR 位[1:0] / 三角波幅值等于 3; 0010: 不屏蔽 LSFR 位[2:0] / 三角波幅值等于 7; 0011: 不屏蔽 LSFR 位[3:0] / 三角波幅值等于 15; 0100: 不屏蔽 LSFR 位[4:0] / 三角波幅值等于 31; 0101: 不屏蔽 LSFR 位[5:0] / 三角波幅值等于 63; 0110: 不屏蔽 LSFR 位[6:0] / 三角波幅值等于 127; 0111: 不屏蔽 LSFR 位[7:0] / 三角波幅值等于 255; 1000: 不屏蔽 LSFR 位[8:0] / 三角波幅值等于 511; 1001: 不屏蔽 LSFR 位[9:0] / 三角波幅值等于 1023; 1010: 不屏蔽 LSFR 位[10:0] / 三角波幅值等于 2047; ≥1011: 不屏蔽 LSFR 位[11:0] / 三角波幅值等于 4095
23:22	WAVE2[1:0]	RW	0	DAC 通道 2 LFSR 噪声/三角波生成使能 00: 关闭波形生成; 01: 使能 LFSR 噪声波形发生器; 10: 使能三角波噪声发生器

				11: 使能锯齿波发生器
21:18	TSEL2[3:0]	RW	0	DAC 通道 2 触发选择, 该位用于选择 DAC 通道 2 的外部触发事件。 0000: SWTRIG2; 0001: TIM8 TRGO 事件 0010: TIM7 TRGO 事件 0011: TIM15 TRGO 事件 0100: TIM2 TRGO 事件 0101: TIM4 TRGO 事件 0110: EXTI9 0111: TIM6 TRGO 事件 1000: TIM3 TRGO 事件 其它: 备用 注意: 该位只能在 TEN2= 1(DAC 通道 2 触发使能)时有效
17	TEN2	RW	0	DAC 通道 2 触发使能 0: 关闭 DAC 通道 2 触发, 写入寄存器 DAC_DHRx 的数据在 1 个时钟周期后传入寄存器 DAC_DOR2; 1: 使能 DAC 通道 2 触发, 写入寄存器 DAC_DHRx 的数据在 3 个时钟周期后传入寄存器 DAC_DOR2。 注意: 如果选择软件触发, 写入寄存器 DAC_DHRx 的数据只需要 1 个时钟周期就可以传入寄存器 DAC_DOR2。 注: 锯齿波模式触发不受 TENx 控制
16	EN2	RW	0	DAC 模拟通道 2 使能 0: 关闭 DAC 模拟通道 2; 1: 使能 DAC 模拟通道 2。 注: 需要设置 MODEx 后使能
15	RSV	-	-	保留

14	CEN1	RW	0	DAC 通道 1 calibration 使能
13	DMAUDIE1	RW	0	DAC 通道 1 Underrun 错误中断使能 0: 关闭 DAC 通道 1 Underrun 错误中断 1: 使能 DAC 通道 1 Underrun 错误中断
12	DMAEN1	RW	0	DAC 通道 1 DMA 使能 0: 关闭 DAC 通道 1 DMA 模式; 1: 使能 DAC 通道 1 DMA 模式
11:8	MAMP1[3:0]	RW	0	DAC 通道 1 屏蔽/幅值选择器, 用来在噪声生成模式下选择屏蔽位, 在三角波生成模式下选择波形的幅值。 0000: 不屏蔽 LSFR 位[0] / 三角波幅值等于 1; 0001: 不屏蔽 LSFR 位[1:0] / 三角波幅值等于 3; 0010: 不屏蔽 LSFR 位[2:0] / 三角波幅值等于 7; 0011: 不屏蔽 LSFR 位[3:0] / 三角波幅值等于 15; 0100: 不屏蔽 LSFR 位[4:0] / 三角波幅值等于 31; 0101: 不屏蔽 LSFR 位[5:0] / 三角波幅值等于 63; 0110: 不屏蔽 LSFR 位[6:0] / 三角波幅值等于 127; 0111: 不屏蔽 LSFR 位[7:0] / 三角波幅值等于 255; 1000: 不屏蔽 LSFR 位[8:0] / 三角波幅值等于 511; 1001: 不屏蔽 LSFR 位[9:0] / 三角波幅值等于 1023; 1010: 不屏蔽 LSFR 位[10:0] / 三角波幅值等于 2047; ≥1011: 不屏蔽 LSFR 位[11:0] / 三角波幅值等于 4095
7:6	WAVE1[1:0]	RW	0	DAC 通道 1 LFSR 噪声/三角波生成使能 00: 关闭波形生成; 01: 使能 LFSR 噪声波形发生器; 10: 使能三角波发生器。 11: 使能锯齿波发生器
5:2	TSEL1[3:0]	RW	0	DAC 通道 1 触发选择, 该位用于选择 DAC 通道 1 的外部触发事件。 0000: SWTRIG1;

				0001: TIM8 TRGO 事件 0010: TIM7 TRGO 事件 0011: TIM15 TRGO 事件 0100: TIM2 TRGO 事件 0101: TIM4 TRGO 事件 0110: EXTI9 0111: TIM6 TRGO 事件 1000: TIM3 TRGO 事件 其它: 备用 注意: 该位只能在 TEN1= 1(DAC 通道 1 触发使能)时有效
1	TEN1	RW	0	DAC 通道 1 触发使能 0: 关闭 DAC 通道 1 触发, 写入寄存器 DAC_DHRx 的数据在 1 个时钟周期后传入寄存器 DAC_DOR1; 1: 使能 DAC 通道 1 触发, 写入寄存器 DAC_DHRx 的数据在 3 个时钟周期后传入寄存器 DAC_DOR1。 注意: 如果选择软件触发, 写入寄存器 DAC_DHRx 的数据只需要 1 个时钟周期就可以传入寄存器 DAC_DOR1。 注: 锯齿波模式触发不受 TENx 控制
0	EN1	RW	0	DAC 模拟通道 1 使能 0: 关闭 DAC 模拟通道 1; 1: 使能 DAC 模拟通道 1。 注: 需要设置 MODEx 后使能

29.5.2. DAC 软件触发寄存器(DAC_SWTRIGR 偏移 04h)

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留

17	SWTRIGB2	RW	0	DAC 通道 2 软件触发 B 0: 无触发; 1: DAC 通道 2 锯齿波递增软件触发。 注意: 软件写 1 触发通道 2 锯齿波递增操作, 该位由硬件置' 0' 。
16	SWTRIGB1	RW	0	DAC 通道 1 软件触发 B 0: 无触发; 1: DAC 通道 1 锯齿波递增软件触发。 注意: 软件写 1 触发通道 1 锯齿波递增操作, 该位由硬件置' 0' 。
15:2	RSV	-	-	保留
1	SWTRIG2	RW	0	DAC 通道 2 软件触发 0: 关闭 DAC 通道 2 软件触发; 1: 使能 DAC 通道 2 软件触发。 注意: 一旦寄存器 DAC_DHR2 的数据传入寄存器 DAC_DOR2, (1 个时钟周期后)该位由硬件置' 0' 。
0	SWTRIG1	RW	0	DAC 通道 1 软件触发 0: 关闭 DAC 通道 1 软件触发; 1: 使能 DAC 通道 1 软件触发。 注意: 一旦寄存器 DAC_DHR1 的数据传入寄存器 DAC_DOR1, (1 个时钟周期后)该位由硬件置' 0' 。

29.5.3. DAC 通道 1 12 位右对齐数据保持寄存器(DAC_DHR12R1 偏移 08h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	DACC1DHRB[11:0]	RW	0	DAC 通道 1 的 12 位右对齐数据 B, 表示 DAC 通道 1 DMA 双数据模式的 12 位数据 B。

15:12	RSV	-	-	保留
11:0	DACC1DHR[11:0]	RW	0	DAC 通道 1 的 12 位右对齐数据，表示 DAC 通道 1 的 12 位数据。

29.5.4. DAC 通道 1 12 位左对齐数据保持寄存器(DAC_DHR12L1 偏移 0Ch)

比特	名称	属性	复位值	描述
31:20	DACC1DHRB[11:0]	RW	0	DAC 通道 1 的 12 位左对齐数据 B，表示 DAC 通道 1 DMA 双数据模式的 12 位数据 B。
19:16	RSV	-	-	保留
15:4	DACC1DHR[11:0]	RW	0	DAC 通道 1 的 12 位左对齐数据，表示 DAC 通道 1 的 12 位数据。
3:0	RSV	-	-	保留

29.5.5. DAC 通道 1 8 位右对齐数据保持寄存器(DAC_DHR8R1 偏移 10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	DACC1DHRB[11:4]	RW	0	DAC 通道 1 的 8 位右对齐数据 B，表示 DAC 通道 1 DMA 双数据模式的高 8 位数据 B。
7:0	DACC1DHR[11:4]	RW	0	DAC 通道 1 的 8 位右对齐数据，表示 DAC 通道 1 的高 8 位数据。

29.5.6. DAC 通道 2 12 位右对齐数据保持寄存器(DAC_DHR12R2 偏移 14h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	DACC2DHRB[11:0]	RW	0	DAC 通道 2 的 12 位右对齐数据 B，表示 DAC 通

				道 2 DMA 双数据模式的 12 位数据 B。
15:12	RSV	-	-	保留
11:0	DACC2DHR[11:0]	RW	0	DAC 通道 2 的 12 位右对齐数据, 表示 DAC 通道 2 的 12 位数据。

29.5.7. DAC 通道 2 12 位左对齐数据保持寄存器(DAC_DHR12L2 偏移 18h)

比特	名称	属性	复位值	描述
31:20	DACC2DHRB[11:0]	RW	0	DAC 通道 2 的 12 位左对齐数据 B, 表示 DAC 通道 2 DMA 双数据模式的 12 位数据 B。
19:16	RSV	-	-	保留
15:4	DACC2DHR[11:0]	RW	0	DAC 通道 2 的 12 位左对齐数据, 表示 DAC 通道 2 的 12 位数据。
3:0	RSV	-	-	保留

29.5.8. DAC 通道 2 8 位右对齐数据保持寄存器(DAC_DHR8R2 偏移 1Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	DACC2DHRB[11:4]	RW	0	DAC 通道 2 的 8 位右对齐数据 B, 表示 DAC 通道 2 DMA 双数据模式的高 8 位数据 B。
7:0	DACC2DHR[11:4]	RW	0	DAC 通道 2 的 8 位右对齐数据, 表示 DAC 通道 2 的高 8 位数据。

29.5.9. 双 DAC 12 位右对齐数据保持寄存器(DAC_DHR12RD 偏移 20h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	DACC2DHR[11:0]	RW	0	DAC 通道 2 的 12 位右对齐数据, 表示 DAC 通道 2

				的 12 位数据。
15:12	RSV	-	-	保留
11:0	DACC1DHR[11:0]	RW	0	DAC 通道 1 的 12 位右对齐数据, 表示 DAC 通道 1 的 12 位数据。

29.5.10. 双 DAC 12 位左对齐数据保持寄存器(DAC_DHR12LD 偏移 24h)

比特	名称	属性	复位值	描述
31:20	DACC2DHR[11:0]	RW	0	DAC 通道 2 的 12 位左对齐数据, 表示 DAC 通道 2 的 12 位数据。
19:16	RSV	-	-	保留
15:4	DACC1DHR[11:0]	RW	0	DAC 通道 1 的 12 位左对齐数据, 表示 DAC 通道 1 的 12 位数据。
3:0	RSV	-	-	保留

29.5.11. 双 DAC 8 位右对齐数据保持寄存器(DAC_DHR8RD 偏移 28h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:16	DACC2DHR[11:4]	RW	0	DAC 通道 2 的 8 位右对齐数据, 表示 DAC 通道 2 的高 8 位数据。
15:8	RSV	-	-	保留
7:0	DACC1DHR[11:4]	RW	0	DAC 通道 1 的 8 位右对齐数据, 表示 DAC 通道 1 的高 8 位数据。

29.5.12. DAC 通道 1 数据输出寄存器(DAC_DOR1 偏移 2Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留

27:16	DACC1DORB[11:0]	RO	0	DAC 通道 1 输出数据 B,这些位为只读类型, 存储由 DAC 通道 1 转换的数据 B。
15:12	RSV	-	-	保留
11:0	DACC1DOR[11:0]	RO	0	DAC 通道 1 输出数据,这些位为只读类型, 存储由 DAC 通道 1 转换的数据。

29.5.13. DAC 通道 2 数据输出寄存器(DAC_DOR2 偏移 30h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	DACC2DORB[11:0]	RO	0	DAC 通道 2 输出数据 B,这些位为只读类型, 存储由 DAC 通道 2 转换的数据 B。
15:12	RSV	-	-	保留
11:0	DACC2DOR[11:0]	RO	0	DAC 通道 2 输出数据,这些位为只读类型, 存储由 DAC 通道 2 转换的数据。

29.5.14. DAC 状态寄存器(DAC_SR 偏移 34h)

比特	名称	属性	复位值	描述
31	RSV	-	-	保留
30	CAL_FLAG2	RO	0	DAC 通道 2 校准标志 0: 校准值低于 offset value, 校准未完成 1: 校准值等于或高于 offset value, 校准完成
29	DMAUDR2	RO		DAC 通道 2 DMA underrun 标志 硬件置位, 软件写 1 清零 0: DAC 通道 2 未发生 DMA underrun 错误; 1: DAC 通道 2 发生 DMA underrun 错误。
28	DORSTAT2	RO	0	DAC 通道 2 输出数据转换使用标志, 在 DMA 双数据模式有效。

				0: DOR[11:0]用于当前 DAC 输出转换 1: DORB[11:0]用于当前 DAC 输出转换
27:25	RSV	-	-	保留
24	SAMOV2	RO	0	DAC 通道 2 采样事件同步完成，可以关闭 PCLK（进入 STOP；一次 DAC 转换信号是否同步到 RC32K 标志。 0: 同步完成 1: 正在同步中，不能关闭 PCLK。
23:15	RSV	-	-	保留
14	CAL_FLAG1	RO	0	DAC 通道 1 校准标志 0: 校准值低于 offset value，校准未完成 1: 校准值等于或高于 offset value，校准完成
13	DMAUDR1	RO	0	DAC 通道 1 DMA underrun 标志 硬件置位，软件写 1 清零 0: DAC 通道 1 未发生 DMA underrun 错误； 1: DAC 通道 1 发生 DMA underrun 错误。
12	DORSTAT1	RO	0	DAC 通道 1 输出数据转换使用标志，在 DMA 双数据模式有效。 0: DOR[11:0]用于当前 DAC 输出转换 1: DORB[11:0]用于当前 DAC 输出转换
11:9	RSV	-	-	保留
8	SAMOV1	RO	0	DAC 通道 1 采样事件同步完成，可以关闭 PCLK（进入 STOP；一次 DAC 转换信号是否同步到 RC32K 标志。 0: 同步完成 1: 正在同步中，不能关闭 PCLK。
7:0	RSV	-	-	保留

29.5.15. DAC 校准控制寄存器(DAC_CCR 偏移 38h)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20:16	OTRIM2[4:0]	RW	0	DAC 通道 2 TRIM 值
15:5	RSV	-	-	保留
4:0	OTRIM1[4:0]	RW	0	DAC 通道 1 TRIM 值

29.5.16. DAC 模式控制寄存器(DAC_MCR 偏移 3Ch)

比特	名称	属性	复位值	描述
31:26	RSV	-	-	保留
25	SINFORMAT2	RW	0	DAC 通道 2 有符号数选择： 0: 输入数据为无符号数模式 1: 输入数据为有符号数模式
24	DMADOUBLE2	RW	0	DAC 通道 2 DMA 双数据模式 0: DMA 普通模式 1: DMA 双数据模式
23:19	RSV	-	-	保留
18:16	MODE2[2:0]	RW	0	DAC 通道 2 MODE 设置，在 DAC_ENx 使能前设置 具体定义参见 MODE1
15:10	RSV	-	-	保留
9	SINFORMAT1	RW	0	DAC 通道 1 有符号数选择： 0: 输入数据为无符号数模式 1: 输入数据为有符号数模式
8	DMADOUBLE1	RW	0	DAC 通道 1 DMA 双数据模式 0: DMA 普通模式 1: DMA 双数据模式
7:3	RSV	-	-	保留

2:0	MODE1[2:0]	RW	0	DAC 通道 1 MODE 设置, 在 DAC_ENx 使能前设置正常模式: 000: DAC Buffer 使能, 输出到 PAD(GPIO) 001: DAC Buffer 使能, 输出到 PAD(GPIO)和内部 OPA 010: DAC Buffer 禁止, 输出到 PAD(GPIO) 011: DAC Buffer 禁止, 输出到内部 OPA 采样保持模式: 100: DAC buffer 使能, 输出到 PAD(GPIO) 101: DAC Buffer 使能, 输出到 PAD(GPIO)和内部 OPA 110: DAC Buffer 禁止, 输出到 PAD(GPIO)和内部 OPA 111: DAC Buffer 禁止, 输出到内部 OPA

29.5.17. DAC 通道 1 采样时间寄存器(DAC_SHSR1 偏移 40h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9:0	TSAMPLE1[9:0]	RW	0	DAC 通道 1 采样时间周期设置 周期数为 TSAMPLE1+1

29.5.18. DAC 通道 2 采样时间寄存器(DAC_SHSR2 偏移 44h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9:0	TSAMPLE2[9:0]	RW	0	DAC 通道 2 采样时间周期设置 周期数为 TSAMPLE2+1

29.5.19. DAC 保持时间寄存器(DAC_SHHR 偏移 48h)

比特	名称	属性	复位值	描述

31:26	RSV	-	-	保留
25:16	THOLD2[9:0]	RW	0	DAC 通道 2 保持时间周期设置 周期数为 THOLD2
15:10	RSV	-	-	保留
9:0	THOLD1[9:0]	RW	0	DAC 通道 1 保持时间周期设置 周期数为 THOLD1

29.5.20. DAC 刷新时间寄存器寄存器(DAC_SHRR 偏移 4Ch)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:16	TREFRESH2[7:0]	RW	0	DAC 通道 2 刷新时间周期设置 周期数为 TREFRESH2
15:8	RSV	-	-	保留
7:0	TREFRESH1[7:0]	RW	0	DAC 通道 1 刷新时间周期设置 周期数为 TREFRESH1

29.5.21. DAC 通道 1 锯齿波寄存器(DAC_STR1 偏移 58h)

比特	名称	属性	复位值	描述
31:16	STINCDATA1[15:0]	RW	0	DAC 通道 1 锯齿波递增值
15:13	RSV	-	-	保留
12	STDIR1	RW	0	DAC 通道 1 锯齿波方向设置 0: 递减 1: 递增
11:0	STRSTDATA1[11:0]	RW	0	DAC 通道 1 锯齿波复位值

29.5.22. DAC 通道 2 锯齿波寄存器(DAC_STR2 偏移 5Ch)

比特	名称	属性	复位值	描述

31:16	STINCDATA2[15:0]	RW	0	DAC 通道 2 锯齿波递增值
15:13	RSV	-	-	保留
12	STDIR2	RW	0	DAC 通道 2 锯齿波方向设置 0: 递减 1: 递增
11:0	STRSTDATA2[11:0]	RW	0	DAC 通道 2 锯齿波复位值

29.5.23. DAC 锯齿波模式寄存器(DAC_STMODR 偏移 60h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:24	STINCTRIGSEL2[3:0]	RW	0	DAC 通道 2 锯齿波递增触发选择 0000: SWTRIGB2; 0001: TIM8 TRGO 事件 0010: TIM7 TRGO 事件 0011: TIM15 TRGO 事件 0100: TIM2 TRGO 事件 0101: TIM4 TRGO 事件 0110: EXTI10; 0111: TIM6 TRGO 事件 1000: TIM3 TRGO 事件 其它: 备用
23:20	RSV	-	-	保留
19:16	STRSTTRIGSEL2[3:0]	RW	0	DAC 通道 2 锯齿波复位触发选择 触发选择与 DAC_CR.TSEL2[3:0]一致
15:12	RSV	-	-	保留
11:8	STINCTRIGSEL1[3:0]	RW	0	DAC 通道 1 锯齿波递增触发选择 0000: SWTRIGB1; 0001: TIM8 TRGO 事件

				0010: TIM7 TRGO 事件 0011: TIM15 TRGO 事件 0100: TIM2 TRGO 事件 0101: TIM4 TRGO 事件 0110: EXTI-10; 0111: TIM6 TRGO 事件 1000: TIM3 TRGO 事件 其它: 备用
7:4	RSV	-	-	保留
3:0	STRSTTRIGSEL1[3:0]	RW	0	DAC 通道 1 锯齿波复位触发选择 触发选择与 DAC_CR.TSEL1[3:0]一致

29.6. 使用流程

29.6.1. 单个 DAC 操作流程（两个 DAC 独立工作）

1. 设置 DAC 通道的触发使能位 TEN_x 为'1'；
2. 通过设置 TSEL_x[3:0]和 STMODR_x，配置 DAC 通道的触发源；
3. 设置 SINFORMAT 选择有无符号数
4. 根据需要设置 DAC 通道的 WAVE_x[1:0]位选择无噪、LFSR 噪声或者三角波噪声或者锯齿波模式，并设 MAMP_x[3:0]为不同的 LFSR 屏蔽位或三角波幅值。
5. 设置 DAC_STR_x 选择锯齿波形（锯齿波模式）
6. 设置 DMADOUBLE_x 选择是否采用双数据模式
7. 通过设置 DMAEN_x 选择是否使能 DMA 模式
8. 设置 EN_x 使能对应 DAC 通道
9. 通过 DMA 或者软件将 DAC 通道转换数据装入所需的 DHR 寄存器(DHR12R_x、DHR12L_x 或 DHR8R_x)。

29.6.2. 双 DAC 同步模式流程

1. 分别设置 2 个 DAC 通道的触发使能位 TEN1 和 TEN2 为'1'；
2. 通过设置 TSELx[3:0]位为相同值，分别配置 2 个 DAC 通道使用相同触发源；
3. 根据需要设置每个 DAC 通道的 WAVEx[1:0]位选择无噪、LFSR 噪声、三角波噪声或者锯齿波模式，并设 MAMPx[3:0]为不同的 LFSR 屏蔽位或三角波幅值。
4. 设置 DAC_STRx 选择锯齿波型（锯齿波模式）
5. 设置 DMADOUBLEx 为 Disable 双数据模式
6. 通过设置 DMAENx 选择是否使能 DMA 模式。当需要使能 DMA 功能时，某一个 DAC 通道的 DMAENx 位被置位即可。
7. 使能两个 DAC 通道(EN1/EN2=1)
8. 通过 DMA 或者软件将双 DAC 通道转换数据装入所需的 DHR 寄存器 (DHR12RD、 DHR12LD 或 DHR8RD)。

30. 通用串行总线（USB）

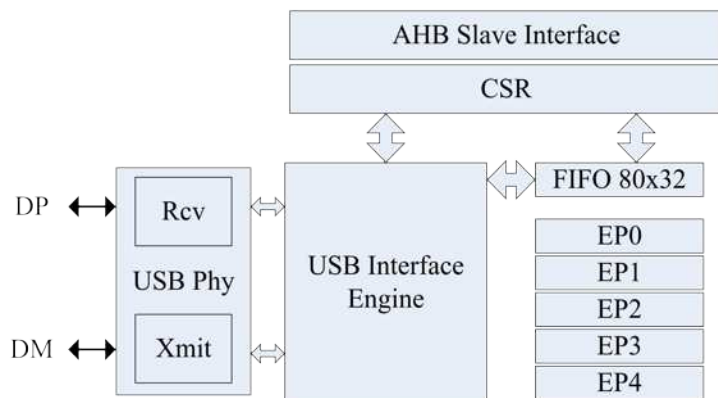
30.1. 概述

USB 设备控制器是一个兼容 USB2.0 全速协议设备接口,其与 USB PHY 配合使用,提供芯片与 USB HOST 通讯的功能。

30.2. 主要特性

- 兼容 USB1.1 和 USB2.0 全速协议;
- 含有 4 个通用双向传输 End Point (EP1、EP2、EP3、EP4) ;
- End Point 支持最大包长度 64Byte, 支持 Memory 和 Fifo 两种访问功能;
- FIFO 模式支持 32bit 方式访问;
- Memory 访问支持 8、16、32bit 三种访问方式;
- 支持挂起、唤醒和远程唤醒功能;
- 支持 Toggle 硬件比对与软件控制功能;
- 支持每一个 End Point 数据传输产生中断功能;
- 支持可选的 CRC 错误回复 NAK 功能;
- 支持数据包超过最大包长度(64byte)自动回复 NAK 功能;
- 支持 IN 操作主机未回 ACK, 接下来 IN 操作 USB 设备回复 NAK 功能;
- 支持令牌包与数据包 EOP 丢失检测,支持可选的丢失 EOP 自动回复 NAK 功能;

图 30-1 USB 结构框架图



30.3. 功能描述

30.3.1. 中断状态和控制寄存器

USB 设备对应每一个中断都有中断状态、中断使能、中断清除寄存器，分别是：

- INT_STAT_RAW：原始中断标志寄存器，无论中断有无使能都会显示中断发生的状态；
- INT_EN：中断时能寄存器，使能某个中断的发生；
- INT_CLR：中断清除寄存器，用户用这个寄存器来清除中断发生的标志。只需要写入 1 即可清除中断，该寄存器会自动清零。

30.3.2. 地址设置 Set Address

控制传输中 Set Address 命令可以全部由硬件完成，也可以由软件完成。软件如果需要知道 Set Address 命令已经发生，则可以使用中断位 SETADDR。

30.3.3. 远程唤醒

远程唤醒功能，由 WORKING_MODE 寄存器中的 USB_REMOTE_WAKEUP 位来实现。向 USB_REMOTE_WAKEUP 写 1，将会在 USB 端口上发出一个 K 状态，其持续的时间由软件控制。软件向 USB_REMOTE_WAKEUP 写 0，USB 端口停止发送 K 状态。

在发起远程唤醒前，需要确认当前 USB Phy 是否为使能状态，Phy 产生的 CLK48M 时钟是否开启。如果没有 CLK48M 时钟，控制器将无法发起远程唤醒操作。USB PHY 的相应配置由 SYSCFG 模块的 PHYCFG 决定。

30.3.4. 令牌包与数据包 CRC 出错选择性回复 NAK

令牌包和数据包发生 CRC 错误，控制器可以选择性的回复 NAK，通过 WORKING_MODE 寄存器中的 USB_CRCERR_NAK_En 来控制。当 USB_CRCERR_NAK_En 为 1 时，USB 设备在 CRC 出错时的握手时相回复 NAK 握手包，否则将不回复握手包。

30.3.5. 数据包长度超过 64 Byte

当 USB 控制器的端点接收到的数据包长度超过 64 Byte 时，状态寄存器 INT_STAT_RAW 中的 DATA_BYTE_MORETHAN_64 位将会置 1。如果 INT_EN 寄存器中的 DATA_BYTE_MORETHAN_64_EN 位为 1，将会产生中断。

30.3.6. 包丢失 EOP

当 WORKING_MODE 寄存器中的 USB_TOKEN_NOEOP_En 位与 USB_DATA_NOEOP_En 位为 1 时，令牌包或数据包在限定的包长度内未收到包长度中状态 (Eop)，状态寄存器 INT_STAT_RAW 中的 NOEOP_ERR_RAW 位将会置 1。如果 INT_EN 寄存器中的 NOEOP_ERR_RAW_EN 位为 1，将会产生中断。控制器对令牌包按长度 32 个 Full Speed 比特位来检测，当令牌包在超过 32 个 Full Speed 比特位时未收到 EOP，则会触发此错误，设备会复位内部状态机，从而不影响下一个包的接收。数据包按有效数据 64byte+8byte 检测，当数据包内有效数据长度超过此值时，也会产生此错误。

令牌包和数据包在发生此类错误时，可以选择是否向 USB 主机回复 NAK 握手包。分别通过设置 WORKING_MODE 寄存器中的 USB_TOKEN_NOEOP_NAK_En 位和 USB_DATA_NOEOP_NAK_En 位来设置。当 USB_TOKEN_NOEOP_NAK_En 位为 1 时，发生令牌包此类错误时，在令牌包的第 32+11 个 Full Speed 比特位后，USB 控制器将回复 NAK 握手包。当 USB_DATA_NOEOP_NAK_En 位为 1 时，发生数据包此类错误时，在数据包的第 64byte+8byte+11bit 个 Full Speed 比特位后，USB 控制器将回复 NAK 握手包。

30.3.7. IN 操作 ACK 超时下次 IN 操作回复 NAK

当主机发起 IN 操作时，如果在握手时相，主机需回复的 ACK 握手包超时，则根据不同的端点，状态寄存器 INT_STAT_RAW 中的 EPx_IN_HANDSHAKE_ERR_RAW 将会被置 1。当 EPx_IN_HANDSHAKE_ERR_RAW 为 1 且主机再次发起针对此端点的 IN 操作时，USB 控制器将会回复 NAK 握手包，直到软件清除 EPx_IN_HANDSHAKE_ERR_RAW 位。

30.3.8. FIFO 访问与 Memory 访问

此 USB 控制器支持 FIFO 访问与 Memory 访问两种方式。FIFO 只支持 32bit 访问，而 Memory 访问支持 8、16、32bit 访问。

30.4. 寄存器描述

USB 寄存器基地址：0x4005_0000

偏置	名称	描述
0x00	WORKING_MODE	工作模式寄存器
0x04	EP0CSR	EP0 传输控制寄存器
0x08	EP1CSR	EP1 传输控制寄存器
0x0C	EP2CSR	EP2 传输控制寄存器
0x10	EP3CSR	EP3 传输控制寄存器
0x14	EP4CSR	EP4 传输控制寄存器
0x18	USB_ADDR	USB 地址寄存器
0x1C	SETUP_0_3_DATA	SETUP 数据包寄存器 0
0x20	SETUP_4_7_DATA	SETUP 数据包寄存器 1
0x24	EP_ADDR	End Point 地址配置寄存器
0x28	CURRENT_PID	当前 USB 总线包 PID 寄存器
0x2C	CURRENT_FRAME_NUMBER	Frame Number 寄存器
0x30	CRC_ERROR_CNT	CRC 错误 Counter 寄存器
0x34	USB_STATUS_DETECT_CNT	Suspend/Resume/Reset 探测时间寄存器
0x40	EP0SENDBN	EP0 发送数据数目寄存器
0x44	EP1SENDBN	EP1 发送数据数目寄存器
0x48	EP2SENDBN	EP2 发送数据数目寄存器
0x4C	EP3SENDBN	EP3 发送数据数目寄存器
0x50	EP4SENDBN	EP4 发送数据数目寄存器
0x100	EP0FIFO	EP0 FIFO 访问入口
0x104	EP1FIFO	EP1 FIFO 访问入口

0x108	EP2FIFO	EP2 FIFO 访问入口
0x10C	EP3FIFO	EP3 FIFO 访问入口
0x110	EP4FIFO	EP4 FIFO 访问入口
0x200~0x23C	EP0MEM	EP0 Memory 访问入口
0x240~0x27C	EP1MEM	EP1 Memory 访问入口
0x280~0x2BC	EP2MEM	EP2 Memory 访问入口
0x2c0~0x2FC	EP3MEM	EP3 Memory 访问入口
0x300~0x33C	EP4MEM	EP4 Memory 访问入口
0xFFE4	INT_STAT_RAW	状态寄存器
0xFFE8	INT_EN	中断使能寄存器
0xFFFF0	INT_CLR	中断清除寄存器

30.4.1. 工作模式寄存器 WORKING_MODE (偏移: 0x0h)

比特	名称	属性	复位值	描述
31: 26	RSV	-	-	保留
25	USB_EP0_ZOD_INTR_En	RW	0x0	USB OUT操作中的0长度数据包是否产生中断选择: 1: 使能此中断; 0: 不使能此中断。
24	USB_DATA_NOEOP_En	RW	0x0	USB控制器在收到的数据包长度超过64+8Byte时, 会认为此包数据丢失了EOP, 控制器内部会复位状态机, 并产生错误状态。 1: 使能此功能; 0: 不使能此功能。
23	USB_TOKEN_NOEOP_En	RW	0x0	USB控制器在收到的令牌包长度超过规定长度时, 会认为此令牌包丢失了EOP, 控制器内部会复位状态机, 并产生错误状态。

				1: 使能此功能; 0: 不使能此功能。
22	USB_DATA_NOEOP_NAK_En	RW	0x0	USB控制器在收到的数据包长度超过64+8Byte时, 会认为此包数据丢失了EOP。此位为1时, 控制器会回复NAK握手包给主机。 1: 回复NAK; 0: 不回复NAK。 此功能只有在USB_DATA_NOEOP_En为1时有效。
21	USB_TOKEN_NOEOP_NAK_En	RW	0x0	USB控制器在收到的令牌包长度超过规定长度时, 会认为此令牌包丢失了EOP。此位为1时, 控制器会回复NAK握手包给主机。 1: 回复NAK; 0: 不回复NAK。 此功能只有在USB_TOKEN_NOEOP_En为1时有效。
20	USB_REMOTE_WAKEUP	RW	0x0	USB远程唤醒控制位。 1: 控制器发送K状态; 0: 控制器不发送K状态 唤醒结束后, 此位需要软件写0清除。
19: 13	RSV	-	-	保留
12	USB_MORETHAN64_NAK_En	RW	0x0	收到的数据包长度超过64Byte, 回复NAK握手包使能位。 1: 使能此功能; 0: 不使能此功能。
11	USB_IN_TIMEOUT_NAK_En	RW	0x0	控制器在IN操作数据包发送完毕后, 未收到主机的ACK握手包, 下次该端点的IN操作设备将回复NAK, 直到软件清除错误

				状态位。 1: 使能此功能; 0: 不使能此功能。
10	USB_CRCERR_NAK_En	RW	0x0	令牌包和数据包CRC错误回复NAK功能使能信号。 1: 使能此功能; 0: 不使能此功能。
9:8	USB_LINE_STATE	R/-	0x0	USB Dp/Dm信号状态位。 8bit: DM; 9bit: DP。
7	RSV	-	-	保留
6	USB_DPPU_LO	RW	0x0	Usb控制器Dp信号2.8k上拉电阻控制位。 1: 开启上拉; 0: 关闭上拉;
5	RSV	-	-	保留
4	USB_DPPU	RW	0x0	Usb控制器Dp信号2.1k上拉电阻控制位。 1: 开启上拉; 0: 关闭上拉;
3	USB_BUS_AUTO_RST_EN	RW	0x1	USB总线复位可以复位控制器地址、收发状态机、收发FIFO使能位; 1: 使能; 0: 不使能;
2	USB_FORCE_RST	RW	0x0	复位控制器地址、收发状态机、收发FIFO; 1: 复位; 0: 不复位; 每次复位操作后, 需软件写0, 清除该位。

1	USB_SUSPEND	RW	0x0	1: 设置UsbPhy为挂起状态; 0: 清除UsbPhy的挂起状态;
0	SPEED_MODE	RW	0x1	USB工作模式选择位。 1: 全速模式; 0: 低速模式; 此芯片仅支持全速模式。

30.4.2. Ep0 传输控制寄存器 EP0CSR (偏移: 0x4h)

比特	名称	属性	复位值	描述
31: 25	RSV	-	-	保留
27	EP0_SEND_HALT	WO	0x0	EP0发送数据停止。 1: 写1停止发送。下一周期自动清零 0: 写0无效。 EP0_SEND_HALT写1后, 如果需要再次启动, 需要写EP0_DATA_START。
26	EP0_RECEIVED_STALL	RO	0x0	EP0接收到Host的STALL握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP0_DATA_START写1可以清除该位。
25	EP0_RECEIVED_NAK	RO	0x0	EP0接收到Host的NAK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP0_DATA_START写1可以清除该位。
24	EP0_RECEIVED_ACK	RO	0x0	EP0接收到Host的ACK握手包标志信号。

				志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向 EP0_DATA_START 写 1 可以清除该位。
23:21	RSV	-	-	保留
20	EP0_OUT_VALID	RO	0x0	Ep0 Out有效标志, 当有效数据进入FIFO时, 此位会置1。写1清零。
19	EP0_OUT_TOGGLE_STATE	RO	0x0	EP0 Out/Setup状态错误标志。 1: Toggle收到的与预期不符; 0: Toggle正常;
18	EP0_OUT_TOGGLE_CTRL_En	WO	0x0	EP0_OUT_TOGGLE_WANT 中的值生效使能。 1: EP0_OUT_TOGGLE_WANT 值写入生效;
17	EP0_OUT_TOGGLE_WANT	RW	0x0	EP0 Out/Setup数据包Toggle的比对值。 1: Data1; 0: Data0;
16	EP0_OUT_TOGGLE_VALUE	RO	0x0	EP0收到的数据包的Toggle值。 1: Data1; 0: Data0;
15	EP0_IN_TOGGLE_CTRL_EN	WO	0x0	EP0_IN_TOGGLE_VALUE 中值生效使能。 1 : EP0_IN_TOGGLE_VALUE 值写入生效;
14	EP0_IN_TOGGLE_VALUE	RW	0x0	IN操作Toggle控制位。 1: Data1; 0: Data0; 写此位会更改比对寄存器的值, 读此位返回的是当前IN Toggle

				的值。
13	EP0_SEND_STALL_DONE	RW	0x0	EP0发送STALL完成标志位。 1: STALL发送完成; 0: STALL未发送完成; 写1清0。
12	EP0_SEND_STALL	WO	0x0	STALL发送控制位。 1: 发送STALL; 0: 不发送STALL; 一次写操作只发送一次STALL。
11	EP0_RECEIVED_DONE	WO	0x0	接收完成控制位。 1: 接收完成; 0: 接收未完成; 向此位写1, 会将FIFO置为Ready状态, 每次接收完数据并且读取完数据后都需要向此位写1, 否则下次OUT/SETUP操作, 设备将会回NAK。
10	EP0_DATA_START	WO	0x0	发送START。 1: 将FIFO中的数据发出; 0: 无操作;
9	EP0_FIFOCLEAR	WO	0x0	EP0 FIFO指针复位控制位。 1: 复位FIFO指针; 0: 不复位FIFO指针;
8	EP0_EN	RW	0x1	EP0端点使能位。 1: 使能; 0: 不使能;
7:0	EP0_RECEIVED_BYTE	RO	0x0	EP0接收到的数据Byte数目。此位需要小于最大包长度。

30.4.3. Ep1 传输控制寄存器 EP1CSR (偏移: 0x8h)

比特	名称	属性	复位值	描述
31: 25	RSV	-	-	保留
27	EP1_SEND_HALT	WO	0x0	EP1发送数据停止。 1: 写1停止发送。下一周期自动清零 0: 写0无效。 EP1_SEND_HALT写1后, 如果需要再次启动, 需要写EP1_DATA_START。
26	EP1_RECEIVED_STALL	RO	0x0	EP1接收到Host的STALL握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP1_DATA_START写1可以清除该位。
25	EP1_RECEIVED_NAK	RO	0x0	EP1接收到Host的NAK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP1_DATA_START写1可以清除该位。
24	EP1_RECEIVED_ACK	RO	0x0	EP1接收到Host的ACK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP1_DATA_START写1可以清除该位。
23:21	RSV	-	-	保留

20	EP1_OUT_VALID	RO	0x0	Ep1 Out有效标志, 当有效数据进入FIFO时, 此位会置1。写1清零。
19	EP1_OUT_TOGGLE_STATE	RO	0x0	EP1 Out/Setup状态错误标志。 1: Toggle收到的与预期不符; 0: Toggle正常;
18	EP1_OUT_TOGGLE_CTRL_En	WO	0x0	EP1_OUT_TOGGLE_WANT 中的值生效使能。 1: EP1_OUT_TOGGLE_WANT 值写入生效;
17	EP1_OUT_TOGGLE_WANT	RW	0x0	EP1 Out/Setup数据包Toggle的比对值。 1: Data1; 0: Data0;
16	EP1_OUT_TOGGLE_VALUE	RO	0x1	EP1收到的数据包Toggle值。 1: Data1; 0: Data0;
15	EP1_IN_TOGGLE_CTRL_EN	WO	0x0	EP1_IN_TOGGLE_VALUE中值生效使能。 1: EP1_IN_TOGGLE_VALUE 值写入生效;
14	EP1_IN_TOGGLE_VALUE	RW	0x0	IN操作Toggle控制位。 1: Data1; 0: Data0; 写此位会更改比对寄存器的值, 读此位返回的是当前IN Toggle 的值。
13	EP1_SEND_STALL_DONE	RO	0x0	EP1发送STALL完成标志位。 1: STALL发送完成; 0: STALL未发送完成; 写1清0。
12	EP1_SEND_STALL	RW	0x0	STALL发送控制位。 1: 发送STALL;

				0: 不发送STALL; 一次写操作只发送多次STALL, 直到清除该位。
11	EP1_RECEIVED_DONE	WO	0x0	接收完成控制位。 1: 接收完成; 0: 接收未完成; 向此位写1, 会将FIFO置为Ready状态, 每次接收完数据并且读取完数据后都需要向此位写1, 否则下次OUT/SETUP操作, 设备将会回NAK。
10	EP1_DATA_START	WO	0x0	发送START。 1: 将FIFO中的数据发出; 0: 无操作;
9	EP1_FIFOCLR	WO	0x0	EP1 FIFO指针复位控制位。 1: 复位FIFO指针; 0: 不复位FIFO指针;
8	EP1_EN	RW	0x1	EP1端点使能位。 1: 使能; 0: 不使能;
7:0	EP1_RECEIVED_BYTE	RO	0x0	EP1接收到的数据Byte数目。此位需要小于最大包长度。

30.4.4. Ep2 传输控制寄存器 EP2CSR (偏移: 0xch)

比特	名称	属性	复位值	描述
31: 25	RSV	-	-	保留
27	EP2_SEND_HALT	WO	0x0	EP2发送数据停止。 1: 写1停止发送。下一周期自动清零 0: 写0无效。 EP2_SEND_HALT写1后, 如果需要再次启动, 需要写

				EP2_DATA_START。
26	EP2_RECEIVED_STALL	RO	0x0	EP2接收到Host的STALL握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP2_DATA_START写1可以清除该位。
25	EP2_RECEIVED_NAK	RO	0x0	EP2接收到Host的NAK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP2_DATA_START写1可以清除该位。
24	EP2_RECEIVED_ACK	RO	0x0	EP2接收到Host的ACK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP2_DATA_START写1可以清除该位。
23:21	-	-	-	-
20	EP1_OUT_VALID	RO	0x0	Ep1 Out有效标志, 当有效数据进入FIFO时, 此位会置1。写1清零。
19	EP2_OUT_TOGGLE_STATE	RO	0x0	EP2 Out/Setup状态错误标志。 1: Toggle收到的与预期不符; 0: Toggle正常;
18	EP2_OUT_TOGGLE_CTRL_En	WO	0x0	EP2_OUT_TOGGLE_WANT 中的值生效使能。 1: EP2_OUT_TOGGLE_WANT 值写入生效;
17	EP2_OUT_TOGGLE_WANT	RW	0x0	EP2 Out/Setup数据包Toggle的

				比对值。 1: Data1; 0: Data0;
16	EP2_OUT_TOGGLE_VALUE	RO	0x1	EP2收到的数据包Toggle值。 1: Data1; 0: Data0;
15	EP2_IN_TOGGLE_CTRL_EN	WO	0x0	EP2_IN_TOGGLE_VALUE中值生效使能。 1: EP2_IN_TOGGLE_VALUE值写入生效;
14	EP2_IN_TOGGLE_VALUE	RW	0x0	IN操作Toggle控制位。 1: Data1; 0: Data0; 写此位会更改比对寄存器的值，读此位返回的是当前IN Toggle的值。
13	EP2_SEND_STALL_DONE	RO	0x0	EP2发送STALL完成标志位。 1: STALL发送完成; 0: STALL未发送完成; 写1清0。
12	EP2_SEND_STALL	RW	0x0	STALL发送控制位。 1: 发送STALL; 0: 不发送STALL; 一次写操作只发送多次STALL，直到清除该位。
11	EP2_RECEIVED_DONE	WO	0x0	接收完成控制位。 1: 接收完成; 0: 接收未完成; 向此位写1，会将FIFO置为Ready状态，每次接收完数据并且读取完数据后都需要向此位写1，否则下次OUT/SETUP操作，

				设备将会回NAK。
10	EP2_DATA_START	WO	0x0	发送START。 1: 将FIFO中的数据发出; 0: 无操作;
9	EP2_FIFOCLR	WO	0x0	EP2 FIFO指针复位控制位。 1: 复位FIFO指针; 0: 不复位FIFO指针;
8	EP2_EN	RW	0x1	EP2端点使能位。 1: 使能; 0: 不使能;
7:0	EP2_RECEIVED_BYTE	RO	0x0	EP2接收到的数据Byte数目。此位需要小于最大包长度。

30.4.5. Ep3 传输控制寄存器 EP3CSR (偏移: 0x10h)

比特	名称	属性	复位值	描述
31: 25	RSV	-	-	保留
27	EP3_SEND_HALT	WO	0x0	EP3发送数据停止。 1: 写1停止发送。下一周期自动清零 0: 写0无效。 EP3_SEND_HALT写1后, 如果需要再次启动, 需要写EP3_DATA_START。
26	EP3_RECEIVED_STALL	RO	0x0	EP3接收到Host的STALL握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP3_DATA_START写1可以清除该位。
25	EP3_RECEIVED_NAK	RO	0x0	EP3接收到Host的NAK握手包标

				志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP3_DATA_START写1可以清除该位。
24	EP3_RECEIVED_ACK	RO	0x0	EP3接收到Host的ACK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP3_DATA_START写1可以清除该位。
23:21	-	-	-	-
20	EP3_OUT_VALID	RO	0x0	Ep3 Out有效标志, 当有效数据进入FIFO时, 此位会置1。写1清零。
19	EP3_OUT_TOGGLE_STATE	RO	0x0	EP3 Out/Setup状态错误标志。 1: Toggle收到的与预期不符; 0: Toggle正常;
18	EP3_OUT_TOGGLE_CTRL_En	WO	0x0	EP3_OUT_TOGGLE_WANT 中的值生效使能。 1: EP3_OUT_TOGGLE_WANT 值写入生效;
17	EP3_OUT_TOGGLE_WANT	RW	0x0	EP3 Out/Setup数据包Toggle的比对值。 1: Data1; 0: Data0;
16	EP3_OUT_TOGGLE_VALUE	RO	0x1	EP3收到的数据包Toggle值。 1: Data1; 0: Data0;
15	EP3_IN_TOGGLE_CTRL_EN	WO	0x0	EP3_IN_TOGGLE_VALUE中值生效使能。 1: EP3_IN_TOGGLE_VALUE

				值写入生效；
14	EP3_IN_TOGGLE_VALUE	RW	0x0	IN操作Toggle控制位。 1: Data1; 0: Data0; 写此位会更改比对寄存器的值， 读此位返回的是当前IN Toggle 的值。
13	EP3_SEND_STALL_DONE	RO	0x0	EP3发送STALL完成标志位。 1: STALL发送完成； 0: STALL未发送完成； 写1清0。
12	EP3_SEND_STALL	RW	0x0	STALL发送控制位。 1: 发送STALL； 0: 不发送STALL； 一次写操作只发送多次STALL， 直到清除该位。
11	EP3_RECEIVED_DONE	WO	0x0	接收完成控制位。 1: 接收完成； 0: 接收未完成； 向此位写1，会将FIFO置为 Ready状态，每次接收完数据并 且读取完数据后都需要向此位写 1，否则下次OUT/SETUP操作， 设备将会回NAK。
10	EP3_DATA_START	WO	0x0	发送START。 1: 将FIFO中的数据发出； 0: 无操作；
9	EP3_FIFOCLR	WO	0x0	EP3 FIFO指针复位控制位。 1: 复位FIFO指针； 0: 不复位FIFO指针；
8	EP3_EN	RW	0x1	EP3端点使能位。

				1: 使能; 0: 不使能;
7:0	EP3_RECEIVED_BYTE	RO	0x0	EP3接收到的数据Byte数目。此位需要小于最大包长度。

30.4.6. Ep4 传输控制寄存器 EP4CSR (偏移: 0x14h)

比特	名称	属性	复位值	描述
31: 25	RSV	-	-	保留
27	EP4_SEND_HALT	WO	0x0	EP4发送数据停止。 1: 写1停止发送。下一周期自动清零 0: 写0无效。 EP4_SEND_HALT写1后, 如果需要再次启动, 需要写EP4_DATA_START。
26	EP4_RECEIVED_STALL	RO	0x0	EP4接收到Host的STALL握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP4_DATA_START写1可以清除该位。
25	EP4_RECEIVED_NAK	RO	0x0	EP4接收到Host的NAK握手包标志信号。 1: 接收到握手包; 0: 未接收到握手包; 此位为只读位, USB Reset或者向EP4_DATA_START写1可以清除该位。
24	EP4_RECEIVED_ACK	RO	0x0	EP4接收到Host的ACK握手包标志信号。 1: 接收到握手包;

				0: 未接收到握手包; 此位为只读位, USB Reset或者向EP4_DATA_START写1可以清除该位。
23:21	-	-	-	-
20	EP4_OUT_VALID	RO	0x0	Ep4 Out有效标志, 当有效数据进入FIFO时, 此位会置1。写1清零。
19	EP4_OUT_TOGGLE_STATE	RO	0x0	EP4 Out/Setup状态错误标志。 1: Toggle收到的与预期不符; 0: Toggle正常;
18	EP4_OUT_TOGGLE_CTRL_En	WO	0x0	EP4_OUT_TOGGLE_WANT 中的值生效使能。 1: EP4_OUT_TOGGLE_WANT 值写入生效;
17	EP4_OUT_TOGGLE_WANT	RW	0x0	EP4 Out/Setup数据包Toggle的比对值。 1: Data1; 0: Data0;
16	EP4_OUT_TOGGLE_VALUE	RO	0x1	EP4收到的数据包Toggle值。 1: Data1; 0: Data0;
15	EP4_IN_TOGGLE_CTRL_EN	WO	0x0	EP4_IN_TOGGLE_VALUE中值生效使能。 1 : EP4_IN_TOGGLE_VALUE 值写入生效;
14	EP4_IN_TOGGLE_VALUE	RW	0x0	IN操作Toggle控制位。 1: Data1; 0: Data0; 写此位会更改比对寄存器的值, 读此位返回的是当前IN Toggle的值。
13	EP4_SEND_STALL_DONE	RO	0x0	EP4发送STALL完成标志位。

				1: STALL发送完成; 0: STALL未发送完成; 写1清0。
12	EP4_SEND_STALL	RW	0x0	STALL发送控制位。 1: 发送STALL; 0: 不发送STALL; 一次写操作只发送多次STALL, 直到清除该位。
11	EP4_RECEIVED_DONE	WO	0x0	接收完成控制位。 1: 接收完成; 0: 接收未完成; 向此位写1, 会将FIFO置为 Ready 状态, 每次接收完数据并 且读取完数据后都需要向此位写 1, 否则下次OUT/SETUP操作, 设备将会回NAK。
10	EP4_DATA_START	WO	0x0	发送START。 1: 将FIFO中的数据发出; 0: 无操作;
9	EP4_FIFOCLR	WO	0x0	EP4 FIFO指针复位控制位。 1: 复位FIFO指针; 0: 不复位FIFO指针;
8	EP4_EN	RW	0x1	EP4端点使能位。 1: 使能; 0: 不使能;
7:0	EP4_RECEIVED_BYTE	RO	0x0	EP4接收到的数据Byte数目。此 位需要小于最大包长度。

30.4.7. Usb 地址寄存器 USB_ADDR (偏移: 0x18h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留

6:0	USB_ADDR	RO	0x0	USB地址寄存器。

30.4.8. SETUP 数据包寄存器 SETUP_0_3_DATA (偏移: 0x1ch)

比特	名称	属性	复位值	描述
31:0	SETUP_0_3_DATA	RO	0x0	SETUP Data包Byte0~Byte3寄存器。

30.4.9. SETUP 数据包寄存器 SETUP_4_7_DATA (偏移: 0x20h)

比特	名称	属性	复位值	描述
31:0	SETUP_4_7_DATA	RO	0x0	SETUP Data包Byte4~Byte7寄存器。

30.4.10. End Point 地址配置寄存器 EP_ADDR (偏移: 0x24h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	EP4_ADDR	RW	0x4	Ep4地址配置。
11:8	EP3_ADDR	RW	0x3	Ep3地址配置。
7:4	EP2_ADDR	RW	0x2	Ep2地址配置。
3:0	EP1_ADDR	RW	0x1	Ep1地址配置。

30.4.11. 总线包 PID 寄存器 CURRENT_PID (偏移: 0x28h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3:0	CURRENT_PID	RO	0x0	当前接收的USB包的PID值。

30.4.12. Frame Number 寄存器 CURRENT_FRAME_NUMBER（偏移：0x2ch）

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留
10:0	CURRENT_FRAME_NUMBER	RO	0x3f	当前帧序号。

30.4.13. CRC 错误 Counter 寄存器 CRC_ERROR_CNT（偏移：0x30h）

比特	名称	属性	复位值	描述
31: 8	RSV	-	-	保留
7:0	CRC_ERROR_CNT	RO	0x0	CRC错误包的个数，在USB Reset时复位。

30.4.14. 探测时间寄存器 USB_STATUS_DETECT_CNT（偏移：0x34h）

比特	名称	属性	复位值	描述
31: 9	RSV	-	-	保留
8:0	USB_STATUS_DETECT_CNT	RW	0xff	Reset/Resume/Suspend检测阈值设定。设置时间为 $USB\_STATUS\_DETECT\_CNT * 5.3\mu s + 2.5\mu s$

30.4.15. EP0 发送数据数目寄存器 EP0SENDBN（偏移：0x40h）

比特	名称	属性	复位值	描述
31: 8	RSV	-	-	保留
7:0	EP0_SEND_BYTE	RW	0x0	EP0发送数据Byte数量寄存器。

30.4.16. EP1 发送数据数目寄存器 EP1SENDBN（偏移：0x44h）

比特	名称	属性	复位值	描述

31: 8	RSV	-	-	保留
7:0	EP1_SEND_BYTE	RW	0x0	EP1发送数据Byte数量寄存器。

30.4.17. EP2 发送数据数目寄存器 EP2SENDBN（偏移：0x48h）

比特	名称	属性	复位值	描述
31: 8	RSV	-	-	保留
7:0	EP2_SEND_BYTE	RW	0x0	EP2发送数据Byte数量寄存器。

30.4.18. EP3 发送数据数目寄存器 EP3SENDBN（偏移：0x4ch）

比特	名称	属性	复位值	描述
31: 8	RSV	-	-	保留
7:0	EP3_SEND_BYTE	RW	0x0	EP3发送数据Byte数量寄存器。

30.4.19. EP4 发送数据数目寄存器 EP4SENDBN（偏移：0x50h）

比特	名称	属性	复位值	描述
31: 8	RSV	-	-	保留
7:0	EP4_SEND_BYTE	RW	0x0	EP4发送数据Byte数量寄存器。

30.4.20. Ep0 FIFO 访问入口 EP0FIFO（偏移：0x100h）

比特	名称	属性	复位值	描述
31:0	EP0FIFO	RW	0x0	EP0 FIFO入口地址，只支持32bit访问。

30.4.21. Ep1 FIFO 访问入口 EP1FIFO（偏移：0x104h）

比特	名称	属性	复位值	描述
31:0	EP1FIFO	RW	0x0	EP1 FIFO入口地址，只支持32bit访问。

30.4.22. Ep2 FIFO 访问入口 EP2FIFO（偏移：0x108h）

比特	名称	属性	复位值	描述
31:0	EP2FIFO	RW	0x0	EP2 FIFO入口地址，只支持32bit访问。

30.4.23. Ep3 FIFO 访问入口 EP3FIFO（偏移：0x10ch）

比特	名称	属性	复位值	描述
31:0	EP3FIFO	RW	0x0	EP3 FIFO入口地址，只支持32bit访问。

30.4.24. Ep4 FIFO 访问入口 EP4FIFO（偏移：0x110h）

比特	名称	属性	复位值	描述
31:0	EP4FIFO	RW	0x0	EP4 FIFO入口地址，只支持32bit访问。

30.4.25. 状态寄存器 INT_STAT_RAW（偏移：0xffe4h）

比特	名称	属性	复位值	描述
31	TOGGLE_STATE_ERR_RAW	RO	0x0	IN/OUT/Setup 操作发生Toggle错误时，此位置1。
30	NOEOP_ERR_RAW	RO	0x0	设备接收到的令牌包如果长度超过协议规定值，或者数据包数据超过64+8byte，此状态位会置1。
29	EP4_IN_HANDSHAKE_ERR_RAW	RO	0x0	EP4 IN操作，主机未成功返回ACK信号时，此位会置1。
28	EP3_IN_HANDSHAKE_ERR_RAW	RO	0x0	EP3 IN操作，主机未成功返回ACK信号时，此位会置1。
27	EP2_IN_HANDSHAKE_ERR_RAW	RO	0x0	EP2 IN操作，主机未成功返回ACK信号时，此位会置1。
26	EP1_IN_HANDSHAKE_ERR_RAW	RO	0x0	EP1 IN操作，主机未成功返回

	RAW			ACK信号时，此位会置1。
25	EP0_IN_HANDSHAKE_ERR_RAW	RO	0x0	EP0 IN操作，主机未成功返回ACK信号时，此位会置1。
24	DATA_BYTE_MORETHAN_64_RAW	RO	0x0	收到的DATA数据包长度超过64byte，此位会置1。
23	CRC_ERR_RAW	RO	0x0	收到令牌包或者数据包的CRC Error，此位会置1。
22	SETADDR_RAW	RO	0x0	当Host设置USB设备地址完成，此位会置1。
21	TURNAROUND_ERROR_RAW	RO	0x0	Host回复Ack包发生TimeOut中断。
20	EP4_ACK_RAW	RO	0x0	Ep4 Ack状态，发送或者接收Ack包，此位会置1。
19	EP4_OUT_RAW	RO	0x0	Ep4 Out中断，当有效数据进入FIFO时，此位会置1。
18	EP4_IN_RAW	RO	0x0	Ep4接收到IN令牌包时，此位会置1。
17	EP3_ACK_RAW	RO	0x0	Ep3 Ack状态，发送或者接收Ack包，此位会置1。
16	EP3_OUT_RAW	RO	0x0	Ep3 Out中断，当有效数据进入FIFO时，此位会置1。
15	EP3_IN_RAW	RO	0x0	Ep3接收到IN令牌包时，此位会置1。
14	EP2_ACK_RAW	RO	0x0	Ep2 Ack状态，发送或者接收Ack包，此位会置1。
13	EP2_OUT_RAW	RO	0x0	Ep2 Out中断，当有效数据进入FIFO时，此位会置1。
12	EP2_IN_RAW	RO	0x0	Ep2接收到IN令牌包时，此位会置1。
11	EP1_ACK_RAW	RO	0x0	Ep1 Ack状态，发送或者接收Ack包，此位会置1。
10	EP1_OUT_RAW	RO	0x0	Ep1 Out中断，当有效数据进入FIFO时，此位会置1。
9	EP1_IN_RAW	RO	0x0	Ep1接收到IN令牌包时，此位会置1。

8	EP0_ACK_RAW	RO	0x0	Ep0 Ack状态，发送或者接收Ack包，此位会置1。
7	EP0_OUT_RAW	RO	0x0	Ep0 Out中断，当有效数据进入FIFO时，此位会置1。
6	EP0_IN_RAW	RO	0x0	Ep0接收到IN令牌包时，此位会置1。
5	SUDAV_RAW	RO	0x0	接收到Setup数据包，此位会置1。
4	SETUPTOK_RAW	RO	0x0	接收到Setup令牌包，此位会置1。
3	SOF_RAW	RO	0x0	接收到Sof包，此位会置1。
2	RESUME_RAW	RO	0x0	Host Resume，此位会置1。
1	SUSPEND_RAW	RO	0x0	Host Suspend，此位会置1。
0	BUS_RESET_RAW	RO	0x0	Host Reset，此位会置1。

30.4.26. 中断使能寄存器 INT_EN（偏移：0xffe8h）

比特	名称	属性	复位值	描述
31	TOGGLE_STATE_ERR_EN	RW	0x0	TOGGLE_STATE_ERR中断使能。
30	NOEOP_ERR_EN	RW	0x0	NOEOP_ERR中断使能。
29	EP4_IN_HANDSHAKE_ERR_EN	RW	0x0	EP4_IN_HANDSHAKE_ERR中断使能。
28	EP3_IN_HANDSHAKE_ERR_EN	RW	0x0	EP3_IN_HANDSHAKE_ERR中断使能。
27	EP2_IN_HANDSHAKE_ERR_EN	RW	0x0	EP2_IN_HANDSHAKE_ERR中断使能。
26	EP1_IN_HANDSHAKE_ERR_EN	RW	0x0	EP1_IN_HANDSHAKE_ERR中断使能。
25	EP0_IN_HANDSHAKE_ERR_EN	RW	0x0	EP0_IN_HANDSHAKE_ERR中断使能。
24	DATA_BYTE_MORETHAN_64_EN	RW	0x0	收到的DATA数据包长度超过64byte中断使能。

23	CRC_ERR_EN	RW	0x0	CRC错误中断使能。
22	SETADDR_EN	RW	0x0	当Host设置USB设备地址完成中断使能。
21	TURNAROUND_ERROR_EN	RW	0x0	Host回复Ack包发生TimeOut中断使能。
20	EP4_ACK_EN	RW	0x0	Ep4 Ack状态，发送或者接收Ack包中断使能。
19	EP4_OUT_EN	RW	0x0	Ep4 Out中断，当有效数据进入FIFO时产生中断使能。
18	EP4_IN_EN	RW	0x0	Ep4接收到IN令牌包中断使能。
17	EP3_ACK_EN	RW	0x0	Ep3 Ack状态，发送或者接收Ack包中断使能。
16	EP3_OUT_EN	RW	0x0	Ep3 Out中断，当有效数据进入FIFO时产生中断使能。
15	EP3_IN_EN	RW	0x0	Ep3接收到IN令牌包中断使能。
14	EP2_ACK_EN	RW	0x0	Ep2 Ack状态，发送或者接收Ack包中断使能。
13	EP2_OUT_EN	RW	0x0	Ep2 Out中断，当有效数据进入FIFO时产生中断使能。
12	EP2_IN_EN	RW	0x0	Ep2接收到IN令牌包中断使能。
11	EP1_ACK_EN	RW	0x0	Ep1 Ack状态，发送或者接收Ack包中断使能。
10	EP1_OUT_EN	RW	0x0	Ep1 Out中断，当有效数据进入FIFO时产生中断使能。
9	EP1_IN_EN	RW	0x0	Ep1接收到IN令牌包中断使能。
8	EP0_ACK_EN	RW	0x0	Ep0 Ack状态，发送或者接收Ack包中断使能。
7	EP0_OUT_EN	RW	0x0	Ep0 Out中断，当有效数据进入FIFO时产生中断使能。
6	EP0_IN_EN	RW	0x0	Ep0接收到IN令牌包中断使能。
5	SUDAV_EN	RW	0x0	接收到Setup数据包中断使能。
4	SETUPTOK_EN	RW	0x0	接收到Setup令牌包中断使能。
3	SOF_EN	RW	0x0	接收到Sof包中断使能。

2	RESUME_EN	RW	0x0	Host Resume中断使能。。
1	SUSPEND_EN	RW	0x0	Host Suspend中断使能。
0	BUS_RESET_EN	RW	0x0	Host Reset中断使能。

30.4.27. 中断清除寄存器 INT_CLR（偏移：0xff0h）

比特	名称	属性	复位值	描述
31	TOGGLE_STATE_ERR_CLR	RW	0x0	TOGGLE_STATE_ERR_RAW中断寄存器清除位。 1: TOGGLE_STATE_ERR_RAW清0; 0: TOGGLE_STATE_ERR_RAW保持不变。
30	NOEOP_ERR_CLR	RW	0x0	NOEOP_ERR_RAW中断寄存器清除位。 1: NOEOP_ERR_RAW清0; 0: NOEOP_ERR_RAW保持不变。
29	EP4_IN_HANDSHAKE_ERR_CLR	RW	0x0	EP4_IN_HANDSHAKE_ERR_RAW中断寄存器清除位。 1: EP4_IN_HANDSHAKE_ERR_RAW清0; 0: EP4_IN_HANDSHAKE_ERR_RAW保持不变。
28	EP3_IN_HANDSHAKE_ERR_CLR	RW	0x0	EP3_IN_HANDSHAKE_ERR_RAW中断寄存器清除位。 1: EP3_IN_HANDSHAKE_ERR_RAW清0; 0: EP3_IN_HANDSHAKE_ERR_RAW保持不变。
27	EP2_IN_HANDSHAKE_ERR_CLR	RW	0x0	EP2_IN_HANDSHAKE_ERR_RAW中断寄存器清除位。

				1: EP2_IN_HANDSHAKE_ERR_RA W清0; 0: EP2_IN_HANDSHAKE_ERR_RA W保持不变。
26	EP1_IN_HANDSHAKE_ERR_CLR	RW	0x0	EP1_IN_HANDSHAKE_ERR_RA W中断寄存器清除位。 1: EP1_IN_HANDSHAKE_ERR_RA W清0; 0: EP1_IN_HANDSHAKE_ERR_RA W保持不变。
25	EP0_IN_HANDSHAKE_ERR_CLR	RW	0x0	EP0_IN_HANDSHAKE_ERR_RA W中断寄存器清除位。 1: EP0_IN_HANDSHAKE_ERR_RA W清0; 0: EP0_IN_HANDSHAKE_ERR_RA W保持不变。
24	DATA_BYTE_MORETHAN_64_CLR	RW	0x0	DATA_BYTE_MORETHAN_64_RA W中断寄存器清除位。 1: DATA_BYTE_MORETHAN_64_RA W清0; 0: DATA_BYTE_MORETHAN_64_RA W保持不变。
23	CRC_ERR_CLR	RW	0x0	CRC_ERR_RAW中断寄存器清除位。 1: CRC_ERR_RAW清0; 0: CRC_ERR_RAW保持不变。
22	SETADDR_CLR	RW	0x0	SETADDR_RAW中断寄存器清除位。

				1: SETADDR_RAW清0; 0: SETADDR_RAW保持不变。
21	TURNAROUND_ERROR_CLR	RW	0x0	TURNAROUND_ERROR_RAW 中断寄存器清除位。 1: TURNAROUND_ERROR_RAW 清0; 0 : TURNAROUND_ERROR_RAW 保持不变。
20	EP4_ACK_CLR	RW	0x0	EP4_ACK_RAW中断寄存器清除 位。 1: EP4_ACK_RAW清0; 0: EP4_ACK_RAW保持不变。
19	EP4_OUT_CLR	RW	0x0	EP4_OUT_RAW中断寄存器清除 位。 1: EP4_OUT_RAW清0; 0: EP4_OUT_RAW保持不变。
18	EP4_IN_CLR	RW	0x0	EP4_IN_RAW中断寄存器清除位。 1: EP4_IN_RAW清0; 0: EP4_IN_RAW保持不变。
17	EP3_ACK_CLR	RW	0x0	EP3_ACK_RAW中断寄存器清除 位。 1: EP3_ACK_RAW清0; 0: EP3_ACK_RAW保持不变。
16	EP3_OUT_CLR	RW	0x0	EP3_OUT_RAW中断寄存器清除 位。 1: EP3_OUT_RAW清0; 0: EP3_OUT_RAW保持不变。
15	EP3_IN_CLR	RW	0x0	EP3_IN_RAW中断寄存器清除位。 1: EP3_IN_RAW清0; 0: EP3_IN_RAW保持不变。

14	EP2_ACK_CLR	RW	0x0	EP2_ACK_RAW中断寄存器清除位。 1: EP2_ACK_RAW清0; 0: EP2_ACK_RAW保持不变。
13	EP2_OUT_CLR	RW	0x0	EP2_OUT_RAW中断寄存器清除位。 1: EP2_OUT_RAW清0; 0: EP2_OUT_RAW保持不变。
12	EP2_IN_CLR	RW	0x0	EP2_IN_RAW中断寄存器清除位。 1: EP2_IN_RAW清0; 0: EP2_IN_RAW保持不变。
11	EP1_ACK_CLR	RW	0x0	EP1_ACK_RAW中断寄存器清除位。 1: EP1_ACK_RAW清0; 0: EP1_ACK_RAW保持不变。
10	EP1_OUT_CLR	RW	0x0	EP1_OUT_RAW中断寄存器清除位。 1: EP1_OUT_RAW清0; 0: EP1_OUT_RAW保持不变。
9	EP1_IN_CLR	RW	0x0	EP1_IN_RAW中断寄存器清除位。 1: EP1_IN_RAW清0; 0: EP1_IN_RAW保持不变。
8	EP0_ACK_CLR	RW	0x0	EP0_ACK_RAW中断寄存器清除位。 1: EP0_ACK_RAW清0; 0: EP0_ACK_RAW保持不变。
7	EP0_OUT_CLR	RW	0x0	EP0_OUT_RAW中断寄存器清除位。 1: EP0_OUT_RAW清0; 0: EP0_OUT_RAW保持不变。
6	EP0_IN_CLR	RW	0x0	EP0_IN_RAW中断寄存器清除位。

				1: EP0_IN_RAW清0; 0: EP0_IN_RAW保持不变。
5	SUDAV_CLR	RW	0x0	接收到Setup数据包中断寄存器清除位。 1: SUDAV_RAW清0; 0: SUDAV_RAW保持不变。
4	SETUPTOK_CLR	RW	0x0	接收到Setup令牌包中断寄存器清除位。 1: SETUPTOK_RAW清0; 0: SETUPTOK_RAW保持不变。
3	SOF_CLR	RW	0x0	接收到Sof包中断寄存器清除位。 1: SOF_RAW清0; 0: SOF_RAW保持不变。
2	RESUME_CLR	RW	0x0	Host Resume中断寄存器清除位。 1: RESUME_RAW清0; 0: RESUME_RAW保持不变。
1	SUSPEND_CLR	RW	0x0	Host Suspend中断寄存器清除位。 1: SUSPEND_RAW清0; 0: SUSPEND_RAW保持不变。
0	BUS_RESET_CLR	RW	0x0	Host Reset中断寄存器清除位。 1: BUS_RESET_RAW清0; 0: BUS_RESET_RAW保持不变。

30.5. 软件流程

30.5.1. USB 连接

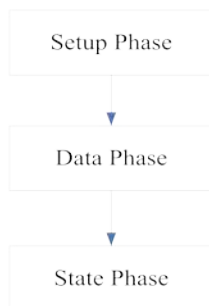
默认状态 USB 是不连接的，需要在完成初始化之后，将 WORKING_MODE 中的寄存器 USB_DPPU 和 USB_DPPU_LO 置位，反之可以断开 USB 连接。

30.5.2. SETUP 数据和 EP0 控制传输数据

控制传输中 set address 命令全部由硬件完成，软件如果需要知道 set address 命令已经发生，则可以通过对 SETADDR_EN 位置 1 来使能 SETADDR 中断。

USB 每次控制传输都要经过，SETUP phase，DATA phase(可选)，STATUS phase。

图 30-2 控制传输过程



STATUS phase 由长度为 0 的数据包来完成。

- Setup phase: 当控制传输 SETUP 数据接收完毕，USB 会产生 SUDAV_RAW 中断，这时 SETUP 数据会在寄存器 SETUP_0_3_DATA 和 SETUP_4_7_DATA;
- Data phase: 当数据 Data Phase 方向为 OUT，那么当数据传输完成后 EP0_OUT_RAW 将会置 1，接收到的数据保存在 EP0 的 FIFO 中，而数据发生的状态将保存在 EP0CSR 中。当数据 Data Phase 方向为 IN，那么当 Host 发来 EP0 的 In Token，这时 EP0IN_RAW 中断将被触发（如果中断已使能），同时会将 EP0 FIFO 中的数据按照 EP0CSR 中的指示发送给 Host。如果数据没有准备好，那么 USB 控制器将自动回复 NAK packet;
- Status phase: Status Phase 的 Status 可能由 Host 发给 Device，也可能由 Device 发回 Host，取决于 Data Phase 的方向。这需要软件维护这一状态。如果需要由 Device 发给 Host，用户需要采用 Data Phase 的方法，将 0 长度数据准备好。USB 控制器会在 EP0 In Token 后将数据发给 Host。

30.5.3. Endpoint In 传输

当 Host 发来 EPx 的 IN token，这时 EPxIN_INT 中断将被触发（如果中断已使能），同时会将 EPx FIFO 中的数据按照 EPxCSR 中的指示发送给 Host，包括发送的数据长度等。Endpoint In 传输的流程图如下图所示：

图 30-3 Endpoint In 传输



30.5.4. Endpoint Out 传输

当 EPxOUT_RAW 置 1 时，EPxCSR 寄存器的 EPx_RECEIVED_BYTE，字段记录了数据接收的长度，用户可以根据这一长度，从 EPxFIFO 或者 EPxMEM 中读取数据。当 EPxOUT 的数据读取完成之后，用户需要写一下 EPxCSR 寄存器的 EPx_RECEIVED_DONE 位，已表示可以允许接收下一笔数据。Endpoint Out 传输的流程图如下图所示：

图 30-4 Endpoint Out 传输



31. 外部存储器控制器（EXMC）

31.1. 概述

外部存储器控制器 EXMC，用来访问各种片外存储器。通过配置寄存器，EXMC 可以把 AMBA 协议转换为专用的片外存储器通信协议，包括 SRAM，PSRAM，ROM，NOR Flash 和 8080 TFT-LCD。用户还可以调整配置寄存器中的时间参数来提高通信效率。

注：100PIN 芯片支持此功能。

31.2. 主要特性

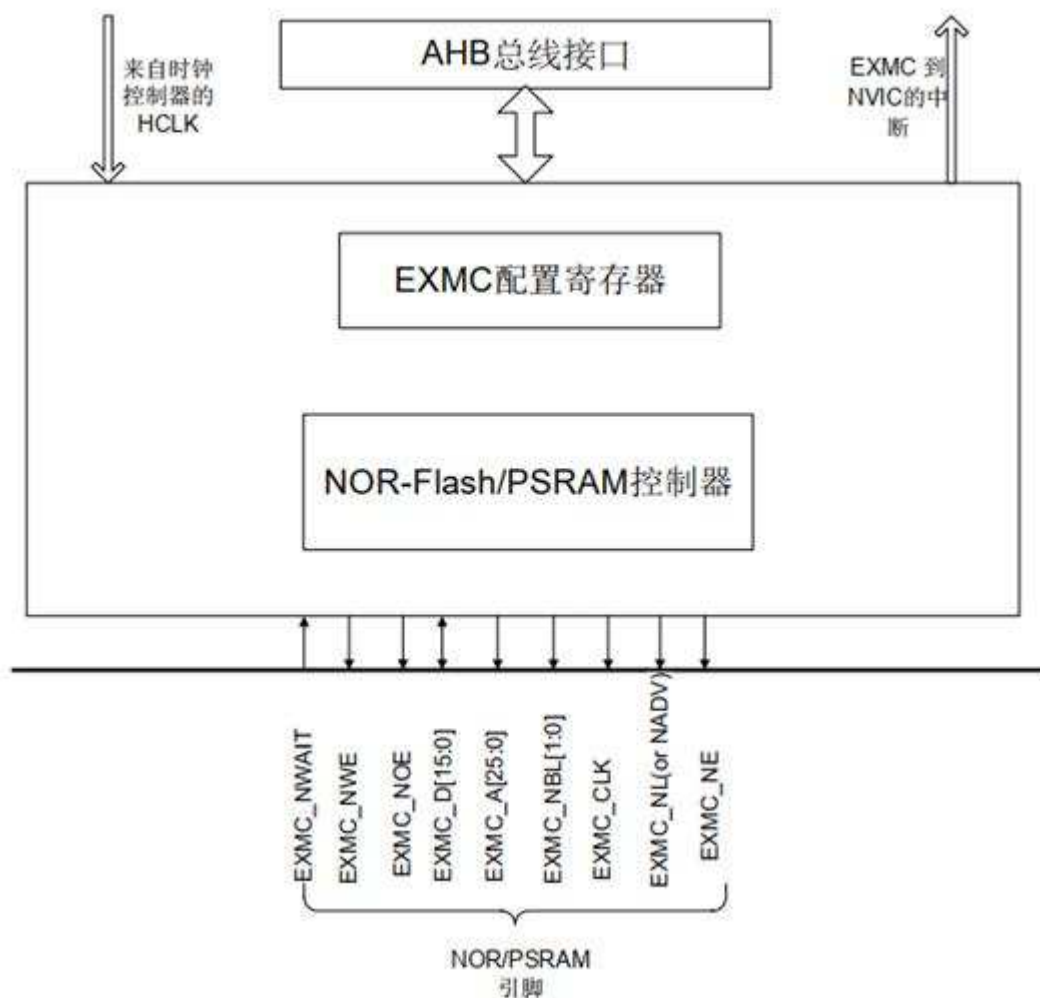
- 支持片外存储器类型

- SRAM
- PSRAM
- ROM
- NOR Flash
- 支持 8080 TFT-LCD
- AMBA 协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 对于部分存储器类型支持独立的读写时序；
- 支持 8 位，或 16 位总线带宽；
- NOR Flash 和 PSRAM 支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当 AMBA 总线宽度与外部存储器数据宽度不同时，会自动分割操作

31.3. 结构框图

EXMC 由 4 个模块组成：AHB 总线接口，EXMC 配置寄存器，NOR/PSRAM 控制器和外部设备接口。AHB 时钟（HCLK）是参考时钟。

图 31-1 EXMC 结构框图



31.4. 功能描述

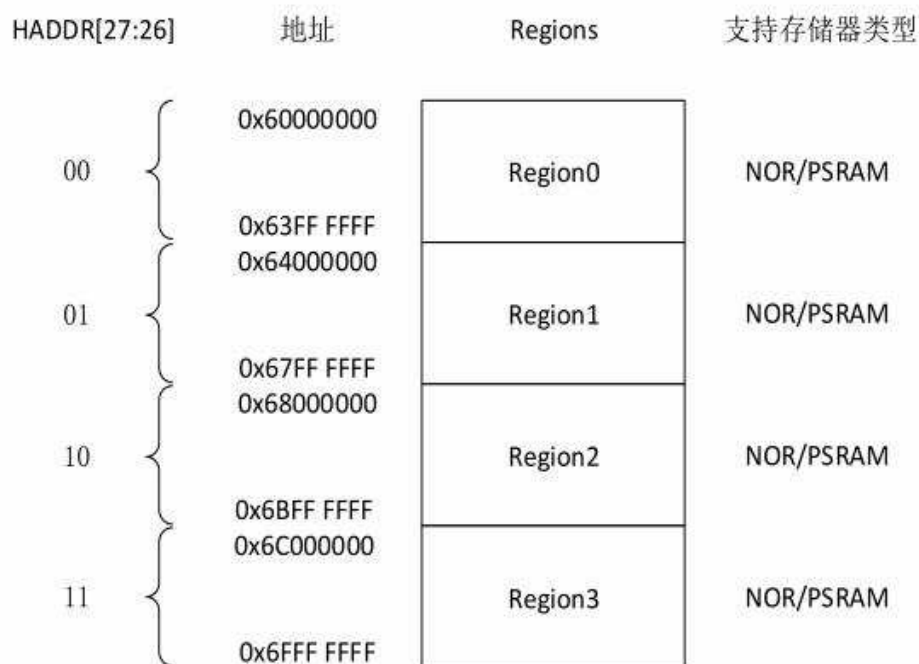
31.4.1. EXMC 访问基本规范

EXMC 是 AHB 总线至外部设备协议的转换接口。32 位的 AHB 读写操作可以转化为几个连续的 8 位或 16 位读写操作。在数据传输的过程中，AHB 数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，EXMC 读写访问需要遵从以下规范。

- AHB 访问宽度等于存储器宽度，则没有问题。
- AHB 访问宽度大于存储器宽度，则自动将 AHB 访问分割成几个连续的存储器数据宽度的传输。

■ AHB 访问宽度小于存储器宽度,如果外部存储设备具有字节选择功能,如 SRAM、ROM、PSRAM,则可通过它的字节通道 EXMC_NBL[1:0]来访问对应的字节。否则禁止写操作,只允许读操作。

图 31-2 EXMC Bank 划分



EXMC 访问区域为 Bank0, 占 64M*4 字节, 用于访问 NOR、PSRAM 设备。Bank0 分为四个 Region 的地址映射。AHB 地址线 HADDR[27:26]作为四个 Region 的片选信号。

由于 HADDR[25:0]是字节地址, 而外部存储器访问有可能不是按字节访问的, 所以会出现地址不一致的情况, 但 EXMC 能实现对 HADDR 的调整以适应外部存储器的数据宽度。具体规则如下:

■ 如果外部存储器的数据宽度是 8 位按字节对齐, HADDR[25:0]与 EXMC_A[25:0]相连, 然后用 EXMC_A[25:0]去连接外部存储器的地址线;

■ 如果外部存储器的数据宽度是 16 位按半字对齐, 就需要将 HADDR 的字节地址转化为半字地址之后再连接外存储器, 所以需要将 HADDR[25:1]与 EXMC_A[24:0]相连。然后用 EXMC_A[24:0]去连接外部存储器的地址线。

31.4.2. NOR/PSRAM 控制器

控制器控制 Bank0, 它可以支持 NOR Flash、PSRAM、SRAM、ROM 和蜂窝 RAM

外部存储器。

注意：

在异步模式下，所有控制器输出信号在内部 AHB 总线时钟（HCLK）的上升沿改变。

在同步模式下，所有控制器输出数据在外部存储器时钟（EXMC_CLK）的下降沿改变。

31.5. 寄存器描述

EXMC 寄存器基地址：0xA000_0000

偏置	名称	描述
0x00	EXMC_SNCTL0	SRAM/NOR Flash 控制寄存器 0
0x04	EXMC_SNTCFG0	SRAM/NOR Flash 时序配置寄存器 0
0x08	EXMC_SNCTL1	SRAM/NOR Flash 控制寄存器 1
0x0C	EXMC_SNTCFG1	SRAM/NOR Flash 时序配置寄存器 1
0x10	EXMC_SNCTL2	SRAM/NOR Flash 控制寄存器 2
0x14	EXMC_SNTCFG2	SRAM/NOR Flash 时序配置寄存器 2
0x18	EXMC_SNCTL3	SRAM/NOR Flash 控制寄存器 3
0x1C	EXMC_SNTCFG3	SRAM/NOR Flash 时序配置寄存器 3
0x104	EXMC_SNWTCFG0	SRAM/NOR Flash 写时序寄存器 0
0x10C	EXMC_SNWTCFG1	SRAM/NOR Flash 写时序寄存器 1
0x114	EXMC_SNWTCFG2	SRAM/NOR Flash 写时序寄存器 2
0x11C	EXMC_SNWTCFG3	SRAM/NOR Flash 写时序寄存器 3

31.5.1. SRAM/ NOR Flash 控制寄存器 (EXMC_SNCTLx) (x 为 0~3) (偏移: 00/08/10/18h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19	SYNCWR	RW	0	选择写操作模式 0: 异步写操作 1: 同步写操作
18:16	CPS[2:0]	RW	000	CRAM 页大小 000: 页边界自动突发分割 001: 128 字节 010: 256 字节 011: 512 字节 100: 1024 字节 其他: 保留
15	ASYNCWAIT	RW	0	异步等待功能使能位 0: 禁用异步等待功能 1: 使能异步等待功能
14	EXMODEN	RW	0	扩展模式使能 0: 禁用扩展模式 1: 使能扩展模式
13	NRWTEN	RW	1	NWAIT 信号使能 对于存储器的突发模式访问, 该位使能/禁用 在 NWAIT 信号中插入等待状态功能。 0: 禁用 NWAIT 信号 1: 使能 NWAIT 信号
12	WREN	RW	1	写操作使能 0: 禁止 EXMC 对外部存储器的写操作, 否则 产生一个 AHB 错误

				1: 允许 EXMC 对外部存储器的写操作（复位缺省值）
11	NRWTCFG	RW	0	NWAIT 信号配置， 只在同步模式有效 0: NWAIT 信号在等待状态前的一个数据周期有效 1: NWAIT 信号在等待状态期间有效
10	WRAPEN	RW	0	非对齐突发模式使能 0: 禁止非对齐突发操作 1: 允许非对齐突发操作
9	NRWTPOL	RW	0	NWAIT 信号极性 0: NWAIT 低电平有效 1: NWAIT 高电平有效
8	SBRSTEN	RW	0	同步突发模式使能 0: 禁止同步突发模式 1: 使能同步突发模式
7	RSV	RW	1	保留
6	NREN	RW	1	NOR 闪存访问使能 0: 禁止 NOR Flash 访问 1: 允许 NOR Flash 访问
5:4	NRW[1:0]	RW	01	存储器数据宽度 00: 8 位 01: 16 位（复位缺省值） 10/11: 保留
3:2	NRTP[1:0]	RW	10	存储器类型 00: SRAM 01: PSRAM（CRAM） 10: NOR Flash（复位之后的默认值） 11: 保留
1	NRMUX	RW	1	数据线/地址线复用

				0: 禁用地址/数据复用功能 1: 允许地址/数据复用功能
0	NRBKEN	RW	0	存储块使能 0: 禁用对应的存储器块 1: 使能对应的存储器块

31.5.2. SRAM/ NOR Flash 时序配置寄存器(EXMC_SNTCFGx)(x 为 0~3) (偏移: 04/0C/14/1Ch)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:28	ASYNCMOD[1:0]	RW	00	异步访问模式 该位域仅在当 EXMC_SNCTL 寄存器的 EXMODEN 位为 1 时有效。 00: 模式 A 01: 模式 B 10: 模式 C 11: 模式 D
27:24	DLAT[3:0]	RW	1111	NOR Flash 数据延时, 仅在同步模式有效 0000: 第一组突发传输时, 数据延迟时间为 2 个 EXMC_CLK 时钟周期 0001: 第一组突发传输时, 数据延迟时间为 3 个 EXMC_CLK 时钟周期 1111: 第一组突发传输时, 数据延迟时间为 17 个 EXMC_CLK 时钟周期
23:20	CKDIV[3:0]	RW	1111	同步模式时钟分频比, 仅在同步模式有效 0000: 保留 0001: EXMC_CLK 周期=2 个 HCLK 周期

				 1111: EXMC_CLK 周期=16 个 HCLK 周期
19:16	BUSLAT[3:0]	RW	1111	总线延迟时间 在复用读模式中使用，避免总线冲突，是总线恢复到高阻态的最小时间。 0000: 总线延迟=1 个 HCLK 周期 0001: 总线延迟=2 个 HCLK 周期 1111: 总线延迟=16 个 HCLK 周期
15:8	DSET[7:0]	RW	11111111	异步数据建立时间 该位域仅在异步模式有效。 00000000: 保留 00000001: 数据建立时间=2 个 HCLK 周期 11111111: 数据建立时间=256 个 HCLK 周期
7:4	AHLD[3:0]	RW	1111	异步地址保持时间 该位域设置地址保持时间，仅在模式 D 与复用模式有效。 0000: 保留 0001: 地址保持时间=2 个 HCLK 1111: 地址保持时间=16 个 HCLK
3:0	ASET[3:0]	RW	1111	异步地址建立时间 该位域设置地址建立时间。 注意：该位域仅在 SRAM，ROM，NOR Flash 的异步模式有效。 0000: 地址建立时间= 1 个 HCLK 1111: 地址建立时间= 16 个 HCLK

31.5.3. SRAM/ NOR Flash 写时序寄存器(EXMC_SNWTCFGx)(x 为 0~3) (偏移: 104/10C/114/11Ch)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:28	WASYNCMOD[1:0]	RW	00	异步访问模式 该位域仅在当 EXMC_SNCTL 寄存器的 EXMODEN 位为 1 时有效。 00: 模式 A 01: 模式 B 10: 模式 C 11: 模式 D
27:24	DLAT[3:0]	RW	1111	NOR Flash 数据延时, 仅在同步模式有效 0000: 第一组突发传输时, 数据延迟时间为 2 个 EXMC_CLK 时钟周期 0001: 第一组突发传输时, 数据延迟时间为 3 个 EXMC_CLK 时钟周期 1111: 第一组突发传输时, 数据延迟时间为 17 个 EXMC_CLK 时钟周期
23:20	RSV	RW	1111	保留
19:16	WBUSLAT[3:0]	RW	1111	总线延迟时间 在复用读模式中使用, 避免总线冲突, 是总线恢复到高阻态的最小时间。 0000: 总线延迟=1 个 HCLK 周期 0001: 总线延迟=2 个 HCLK 周期 1111: 总线延迟=16 个 HCLK 周期
15:8	WDSET[7:0]	RW	11111111	异步数据建立时间

				该位域仅在异步模式有效。 00000000: 保留 00000001: 数据建立时间=2 个 HCLK 周期 11111111: 数据建立时间=256 个 HCLK 周期
7:4	WAHLD[3:0]	RW	1111	异步地址保持时间 该位域设置地址保持时间，仅在模式 D 与复用模式有效。 0000: 保留 0001: 地址保持时间=2 个 HCLK 1111: 地址保持时间=16 个 HCLK
3:0	WASET[3:0]	RW	1111	异步地址建立时间 该位域设置地址建立时间。 注意：该位域仅在 SRAM，ROM，NOR Flash 的异步模式有效。 0000: 地址建立时间= 1 个 HCLK 1111: 地址建立时间= 16 个 HCLK

31.6. 使用流程

1. 使能 EXMC 时钟
2. 设置控制寄存器 EXMC_SNCTLx，配置总线宽度、外部存储器类型等
3. 设置时序配置寄存器 EXMC_SNTCFGx，依据外部存储器手册，配置地址建立时间、地址保持时间、数据建立时间等
4. 配置 GPIO 管脚，使能 GPIO 时钟

32. CRC 计算单元

32.1. 概述

循环冗余校验(Cyclic Redundancy Check CRC)是一种根据数据产生简短固定位数校验码的一种散列函数，主要利用除法及余数的原理来检测或校验数据传输或者保存后可能出现的错误。CRC 硬件电路支持用户设置 CRC 初值，结果异或值，输入输出数据正反向，并且支持配置 7/8/16/32 位多项式。

32.2. 主要特性

- 支持设置 7/8/16/32 位多项式及其正反向设置；
- 支持 8/16/32 位数据输入输出及其正反向设置；
- 支持设置 CRC 初值和结果异或值及其正反向设置；
- 对 32/16/8(7)位的数据长度，CRC 的计算分别要花费 4/2/1 个 AHB 时钟周期；

32.3. 寄存器描述

CRC 寄存器基地址：0x40010C00

32.3.1. 数据寄存器/ CRC_DATA (偏移：00h)

比特	名称	属性	复位值	描述
31:0	DATA	RW	0	写：写入需要进行 CRC 校验计算的数据，如需要校验的数据是多个字节数据只需按顺序逐次写入 读：读出 CRC 计算结果，写入的数据无法再次读出，读操作返回的是上一次 CRC 计算的结果

32.3.2. 控制寄存器/ CRC_CTRL (偏移：04h)

比特	名称	属性	复位值	描述

31:11	RSV	-	-	保留
10	PLOY_REV	RW	0	多项式是否进行高低位倒序 0: 不倒序 1: 倒序
9	OUTXOR_REV	RW	0	结果异或值是否进行高低位倒序 0: 不倒序 1: 倒序
8	INITIAL_REV	RW	0	CRC 初始值是否进行高低位倒序 0: 不倒序 1: 倒序
7	RSLT_REV	RW	0	CRC 计算结果是否进行高低位倒序 0: 不倒序 1: 倒序
6:5	DATA_REV	RW	0	CRC 计算数据是否进行高低位倒序 0: 输入数据不倒序 1: 输入数据按字节倒序 2: 输入数据按半字倒序 3: 输入数据按字倒序
4:3	PLOY_LEN	RW	0	多项式长度 0: 32 位 1: 16 位 2: 8 位 3: 7 位
2:1	DATA_LEN	RW	0	数据寄存器有效数据字节长度 0: 1 个字节 1: 2 个字节 2: 3 个字节 3: 4 个字节
0	RST	RW	0	写 1 复位 CRC_DATA 寄存器, CRC_DATA 寄存器将自动初始化为 CRC_INIT 寄存器中的值

32.3.3. 初始值寄存器/ CRC_INIT (偏移: 08h)

比特	名称	属性	复位值	描述
31:0	INIT	RW	0	写入 CRC 初始值

32.3.4. 结果异或值寄存器/ CRC_OUTXOR (偏移: 10h)

比特	名称	属性	复位值	描述
31:0	OUTXOR	RW	0	写入结果异或值

32.3.5. 多项式寄存器/ CRC_POLY (偏移: 14h)

比特	名称	属性	复位值	描述
31:0	POLY	RW	0x04C11DB7	写入多项式值， 同时需要配置 CTRL[4: 3]

32.3.6. 独立数据寄存器/CRC_FDATA (偏移: 18h)

比特	名称	属性	复位值	描述
31:0	DATA	RW	0	独立数据寄存器位，这些位与 CRC 计算无关，可以给任何其他外设用于其他目的。

32.4. 使用流程

1. 设置控制寄存器 CRC_CTRL，选择多项式长度和有效数据字节长度以及是否逆序；
2. 往多项式寄存器 CRC_POLY 写入多项式；
3. 往初始值寄存器 CRC_INIT 写入初始值，如果初始值为 0，此步可省略；
4. 往结果异或寄存器 CRC_OUTXOR 写入结果异或值，如果初始值为 0，此步可省略；
5. 向 CRC_DATA 中依次写入 8/16/32 位 CRC 计算数据；
6. 写完之后即可读 CRC_DATA，将一次返回 CRC 计算结果。

33. 算法库

33.1. 数据类型

算法库中的数据类型定义如下：

```
typedef unsigned char uint8_t;
```

```
typedef unsigned int uint32_t;
```

即 `uint8_t` 表示 8 位无符号数，`uint32_t` 表示 32 位无符号数。

Q31 格式中，数字由 1 个符号位和 31 个二进制小数位表示，因此数字范围是 $-1(0x80000000)$ 到 $1-2^{31}(0x7FFFFFFF)$ ；

例如 0.5 转换成 Q31 格式为： $0.5 * 2^{31} = 1073741824 = 0x40000000$ ，Q31 格式 $0x40000000$ 转换成浮点数为： $0x40000000 / 2^{31} = 0.5$

33.2. HRNG

33.2.1. 主要特性

- 内含可靠噪声振荡器；
- 符合国际 FIPS-140-2 和 NIST SP800-22 测试标准；
- 符合国密局《随机数检测规范》测试标准；

33.2.2. 库文件说明

HRNG 模块需要的库及头文件如下：

<code>hal_hrng.lib</code>	实现取随机数操作
<code>hal_hrng.h</code>	<code>hal_hrng.lib</code> 对应的头文件

33.2.3. 函数说明

表 33-1 HRNG 驱动函数说明

函数名	功能描述	参数	
HAL_HRNG_Init	初始化函数用于启动随机数模块	输入	无
		输出	无
		返回值	无
HAL_HRNG_DeInit	关闭随机数模块	输入	无
		输出	无

		返回值	无
HAL_HRNG_GetHrng	取多个字节随机数	输入	uint32_t byte_len: 取随机数的字节长度
		输出	uint8_t *hdata: 存放随机数的起始地址
		返回值	0: 表示取随机数成功; 1: 表示失败

33.3. AES

33.3.1. 主要特性

- 支持 AES 加密和解密运算;
- 支持 ECB 模式和 CBC 模式;
- 数据输入和输出支持 SWAP 模式, 即大小端可配置;
- 支持 128/192/256 bit 密钥长度;

33.3.2. 库文件说明

表 33-2 库文件说明

hal_aes.lib	实现 AES 加解密运算
hal_aes.h	源文件对应的头文件

33.3.3. 函数说明

表 33-3 AES 驱动函数说明

函数名	功能描述	参数	
HAL_AES_SetKey	配置 AES 密钥及 对调模式	输入	uint32_t *keyin: 存放密钥的起始地址
			uint8_t key_len: 密钥长度(128/192/256)
			uint8_t swap_en: 输入输出大小端对调使能 0: 输入输出数据大小端对调禁止 1: 输入输出数据大小端对调使能
		输出	无
		返回值	无
HAL_AES_Crypt	AES 加解密函数	输入	uint32_t *indata: 存放输入数据的起始地址
			uint32_t block_len: 加解密的块(128 bit)个数
			uint8_t operation: 加解密操作选择 0: 解密 1: 加密
			uint8_t mode: 加解密模式选择 0: ECB 模式 1: CBC 模式
			uint32_t *iv: 存放 CBC 模式初始向量的起始地址

			uint32_t security_mode: 安全模式选择 0x12345678: 普通模式 0: 安全模式
		输出	uint32_t *outdata: 存放输出结果的起始地址
		返回值	uint32_t 加解密运行结果 0: 运行失败 0xa59ada68: 运行成功
HAL_AES_SetKey_U8	配置 AES 密钥及 对调模式 (字节方式)	输入	uint8_t *keyin: 存放密钥的起始地址
			uint8_t key_len: 密钥长度(128/192/256)
			uint8_t swap_en: 输入输出大小端对调使能 0: 输入输出数据大小端对调禁止 1: 输入输出数据大小端对调使能
		输出	无
HAL_AES_Crypt_U8	AES 加解密函数 (字节方式)	输入	uint8_t *indata: 存放输入数据的起始地址
			uint32_t block_len: 加解密的块(128 bit)个数
			uint8_t operation: 加解密操作选择 0: 解密 1: 加密
			uint8_t mode: 加解密模式选择 0: ECB 模式 1: CBC 模式
			uint8_t *iv: 存放 CBC 模式初始向量的起始地址
			uint32_t security_mode: 安全模式选择 0x12345678: 普通模式 0: 安全模式
		输出	uint8_t *outdata: 存放输出结果的起始地址
		返回值	uint32_t 加解密运行结果 0: 运行失败 0xa59ada68: 运行成功

33.3.4. 注意事项

1) 如果输入 AES 模块进行加解密运算的数据为 32bit 数组, 高 32 位字数据存储在数组的低位元素中, 每个字以大端方式存放, 举例如下:

设待加密数据为: 0x112233445566778899aabbccddeeff00

则输入 AES 模块进行运算的数组为:

```
uint32_t plain_text[4] = {0x11223344, 0x55667788, 0x99aabbcc, 0xdddeeff0};
```

如果使能 SWAP 模式, 则输入 AES 模块进行运算的数组为:

```
uint32_t plain_text[4] = {0x44332211, 0x88776655, 0xccbb99, 0x00ffeedd};
```

输出数据格式与输入相同。

- 2) 如果输入 AES 模块进行加解密运算的数据为 8bit 数组，举例如下：

设待加密数据为：0x112233445566778899aabbccddeeff00

则输入 AES 模块进行运算的数组为：

```
uint8_t plain_text[16] =
{0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xaa,0xbb,
0xcc,0xdd,0xee,0xff,0x00};
```

如果使能 SWAP 模式，则输入 AES 模块进行运算的数组为：

```
uint8_t plain_text [16]={0x44,0x33,0x22,0x11,0x88,0x77,0x66,0x55,0xcc,0xbb,
0xaa,0x99,0x00,0xff,0xee,0xdd};
```

输出数据格式与输入相同。

- 3) SWAP 模式对密钥、初始向量、输入数据、输出数据同时有效。

33.4. HASH

33.4.1. 主要特性

- 支持 SHA1/SHA256 算法；

33.4.2. 库文件说明

HASH 模块需要的库及头文件如下：

hal_sha1.lib	实现 SHA1 运算
hal_sha1.h	hal_sha1.lib 对应的头文件
hal_sha256.lib	实现 SHA256 运算
hal_sha256.h	hal_sha256.lib 对应的头文件

33.4.3. 函数说明

表 33-4 HASH 驱动函数说明

函数名	功能描述	参数	
HAL_SHA1_Hash	SHA1 运算	输入	uint8_t *pDataIn: 待压缩数据的起始地址
			uint32_t DataLen: 待压缩数据的字节长度
		输出	uint8_t *pDigest: 摘要值的起始地址
		返回值	无
HAL_SHA256_Hash	SHA256 运算	输入	uint8_t *pDataIn: 待压缩数据的起始地址
			uint32_t DataLen: 待压缩数据的字节长度
		输出	uint8_t *pDigest: 摘要值的起始地址

		返回值	无

33.4.4. 注意事项

- 1) 涉及到 HASH 运算的数据格式都是 big endian;

若: $A = 0x11223344556677889900$;

则: $A[0] = 0x11$;

$A[1] = 0x22$;

$A[2] = 0x33$;

$A[3] = 0x44$;

$A[4] = 0x55$;

$A[5] = 0x66$;

$A[6] = 0x77$;

$A[7] = 0x88$;

$A[9] = 0x99$;

$A[10] = 0x00$;

33.5. FAU

33.5.1. 主要特性

- 24 位 CORDIC 旋转引擎;
- 支持 $\sin/\cos/\tan2/\sinh/\cosh/\tanh/\ln/\sqrt{}$ 等函数;

33.5.2. 库文件说明

FAU 模块需要的库及头文件如下:

hal_fau.lib	实现 FAU 运算
hal_fau.h	hal_fau.lib 对应的头文件

33.5.3. 函数说明

表 33-5 FAU 驱动函数说明

函数名	功能描述	参数	
HAL_CORDIC_CosSin_x	cos/sin 运算	输入	int angle_para: 以弧度为单位的角度值 (除以 π ,范围 $[-1,1]$,Q31 格式)

x 为计算精度， 范围(1~8)		输出	int* cos_data:输入角度的 cos 值 (范围[-1,1],Q31 格式)
			int* sin_data: 输入角度的 sin 值 (范围[-1,1],Q31 格式)
		返回值	无
HAL_CORDIC_ AtanSqrt_x x 为计算精度， 范围(1~8) atan 与 sqrt 值 在精度 1 时误差 非常大，不建议 使用	atan/sqrt 运算	输入	int x: x 方向坐标值(如果 x >1,那么必须对其进行 缩放以使其范围在[-1,1], Q31 格式)
			int y: y 方向坐标值(如果 y >1,那么必须对其进行 缩放以使其范围在[-1,1], Q31 格式)
		输出	int* sqrt_data: xy 代表向量的长度(Q31 格式) 需要除以(CORDIC_F_31>>4)得到 float 格式值
			int* atan_data: xy 代表向量的角度(Q31 格式) 需要乘以 PI 得到以弧度为单位的角度值
		返回值	无
HAL_CORDIC_ CoshSinh_x x 为计算精度， 范围(1~8)	cosh/sinh 运算	输入	int angle_para: 双曲线角度值 (范围[-0.559~0.559],Q31 格式)
		输出	int* cosh_data:输入角度的 cosh 值 (范围[0.5 0.846],Q31 格式)
			int* sinh_data: 输入角度的 sinh 值 (范围[-0.683 0.683],Q31 格式)
		返回值	无
HAL_CORDIC_ Atanh_x x 为计算精度， 范围(1~8)	atanh 运算	输入	int angle_para: 双曲线角度值 (范围[-0.403 0.403], Q31 格式)
		输出	int* atanh_value: 输入角度的 atanh 值，需要乘 2 得到正确的 atanh 值
		返回值	无
HAL_CORDIC_ Ln_x x 为计算精度， 范围(1~8)	ln 运算	输入	int arg:输入参数(范围[0.054 0.875], Q31 格式)
			int scale:缩放因子 需要根据原始数据 x 范围进行缩放: 原始数据范围 缩放因子 arg 范围 $0.107 \leq x < 1$ 1 $0.0535 \leq \text{arg} < 0.5$ $1 \leq x < 3$ 2 $0.25 \leq \text{arg} < 0.75$ $3 \leq x < 7$ 3 $0.375 \leq \text{arg} < 0.875$ $7 \leq x \leq 9.35$ 4 $0.4375 \leq \text{arg} < 0.584$
		输出	int* ln_value: 输入参数的 ln 值，需要乘以 4 得到 正确的 ln 值
		返回值	无
HAL_CORDIC_ Sqrt_x x 为计算精度， 范围(1~8)	sqrt 运算	输入	int arg: 输入参数(范围[0.027 0.875], Q31 格式)
			int scale:缩放因子 需要根据原始数据 x 范围进行缩放: 原始数据范围 缩放因子 arg 范围 $0.027 \leq x < 0.75$ 0 $0.027 \leq \text{arg} < 0.75$

			$0.75 \leq x < 1.75$ 1 $0.375 \leq \text{ARG1} < 0.875$ $1.75 \leq x \leq 2.341$ 2 $0.4375 \leq \text{ARG1} \leq 0.585$
		输出	int* sqrt_value: 输入参数的 ln 值，需要乘以 $2^{\text{scale}/\text{W\_INV\_Q31}}$ 得到正确的 ln 值
		返回值	无

33.5.4. 注意事项

- 1) Q31 格式中，数字由 1 个符号位和 31 个二进制小数位表示，因此数字范围是 $-1(0x80000000)$ 到 $1-2^{-31}(0x7FFFFFFF)$ ；

例如 0.5 转换成 Q31 格式为： $0.5 * 2^{31} = 1073741824 = 0x40000000$