

# 航芯 ACM32F403 用户手册

---

Version 1.7



上海爱信诺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，其著作权归上海爱信诺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。

## 版本修订

版本	日期	作者	描述
V1.0	2020-10-09	Aisinochip	初始化版本
V1.1	2020-12-17	Aisinochip	增加一些模块的功能说明
V1.2	2021-01-22	Aisinochip	更新模块框图；对一些模块增加了补充说明
V1.3	2021-02-19	Aisinochip	更新了LVDCFG寄存器滤波时间的描述； 修改8.3.9章节的引用错误，正确的引用为表格8-4； 更新了25.4.2章节DAC数据对齐描述错误的地方； 更新TIM2 ARR寄存器有效位数； 增加ADC和DAC章节中关于BUFFER的注释
V1.4	2021-02-25	Aisinochip	增加OPAMP章节寄存器说明中驱动模式HSM的说明； 修改比较器滤波部分功能描述和HYS数值 22.3.3章节增加时钟框图
V1.5	2021-04-19	Aisinochip	SPI修改MSB模式下的时序图，增加LSB模式下的时序图
V1.6	2021-09-09	Aisinochip	增加I2C滤波算法的描述；增加UART IFLS的TX 1/16的注意事项
V1.7	2022-03-04	Aisinochip	修改ADC寄存器表错误； 完善EFC寄存器rd_wait的描述； 增加定时器TIM1/TIM15/TIM16/TIM17 CR1寄存器关于刹车滤波参数的描述；增加定时器框图中中TRGO的描述；更新定时器刹车电路；更新I2C从机接收数据描述；更新I2C分频因子描述。

## 目录

版本修订 .....	3
目录 .....	I
图目录 .....	XIX
表目录 .....	XXIII
1. 文档约定 .....	0
1.1. 基本信息 .....	0
1.2. 寄存器属性缩写表 .....	0
1.3. 术语 .....	0
2. 存储器及系统架构 .....	1
2.1. 内核处理器 .....	1
2.2. 系统架构 .....	3
2.3. 存储器映射 .....	3
2.4. 片上 SRAM .....	7
2.5. 片上 Flash .....	7
2.6. BOOT 配置 .....	7
2.7. 芯片序列号 .....	8
2.8. 电源控制 .....	8
2.9. 低功耗模式 .....	8
2.10. 系统寄存器 .....	10
2.10.1. 复位控制寄存器/RCR (偏移: 00h) .....	12
2.10.2. 复位源状态寄存器/RSR (偏移: 04h) .....	13
2.10.3. 模块复位控制寄存器 2/IPRSTR2(偏移: 08h) .....	14
2.10.4. 模块复位控制寄存器 1/IPRSTR1(偏移: 0Ch) .....	15
2.10.5. 时钟控制寄存器 1/CCR1(偏移: 10h) .....	17
2.10.6. 时钟控制寄存器 2/CCR2 (偏移: 14h) .....	17
2.10.7. 时钟中断寄存器/CIR (偏移: 18h) .....	18
2.10.8. 模块时钟使能寄存器/IPCKENR (偏移: 1Ch) .....	20
2.10.9. 模块时钟使能寄存器 2/IPCKENR2 (偏移: 20h) .....	22
2.10.10. RCH 模块控制寄存器/RCHCR (偏移: 24h) .....	23
2.10.11. XTHCR 模块控制寄存器/XTHCR (偏移: 28h) .....	24
2.10.12. PLL 模块控制寄存器/PLLCR (偏移: 2Ch) .....	24
2.10.13. 模拟 LDO 控制寄存器 LDOCR (偏移: 30h) .....	26
2.10.14. 工作模式寄存器/WMR(偏移: 38h) .....	26
2.10.15. 时钟输出控制寄存器/CLKOCR(偏移: 3Ch) .....	27
2.10.16. 版本寄存器/VER (偏移: 40h) .....	28
2.10.17. 系统配置寄存器 1/SYSCFG1(偏移: 44h) .....	29
2.10.18. LVD 配置寄存器/LVDCFG (偏移: 48h) .....	30
2.10.19. STOP 模式配置寄存器/STOPCFG(偏移: 4Ch) .....	31
2.10.20. USBPHY 配置寄存器/PHYCFG (偏移: 54h) .....	32
2.10.21. Memory 配置寄存器/MEMCFG (偏移: 58h) .....	34
2.10.22. PA 管脚复用寄存器 1/PASEL1(偏移: 60h) .....	34
2.10.23. PA 管脚复用寄存器 2/PASEL2(偏移: 64h) .....	36

2.10.24. PB 管脚复用寄存器 1/PBSEL1(偏移: 68h).....	38
2.10.25. PB 管脚控制寄存器 2/PBSEL2(偏移: 6Ch) .....	39
2.10.26. PA/B 管脚上拉寄存器/PA/BPUR (偏移: 70h).....	41
2.10.27. PA/B 管脚下拉寄存器/PA/BPDR (偏移: 74h).....	41
2.10.28. PA 管脚驱动选择寄存器 PASTR (偏移: 78h) .....	42
2.10.29. PB 管脚驱动选择寄存器 PBSTR (偏移: 7ch) .....	43
2.10.30. PA/B 管脚施密特使能寄存器 PA/BSMTR (偏移: 80h) .....	43
2.10.31. PA/B 管脚 OD 使能寄存器 PA/BODR (偏移: 84h) .....	44
2.10.32. PA/B 管脚数模选择寄存器 PA/BADS (偏移: 88h) .....	44
2.10.33. PC 管脚复用寄存器 1/PCSEL1(偏移: 90h) .....	44
2.10.34. PC 管脚复用寄存器 2/PCSEL2(偏移: 94h) .....	46
2.10.35. PD 管脚复用寄存器 1/PDSEL1(偏移: 98h).....	48
2.10.36. PD 管脚复用寄存器 2/PDSEL2(偏移: 9Ch) .....	50
2.10.37. PC/D 管脚上拉寄存器 PC/DPUR (偏移: a0h).....	51
2.10.38. PC/D 管脚下拉寄存器/PC/DPDR (偏移: A4h) .....	52
2.10.39. PC 管脚驱动选择寄存器 PCSTR (偏移: A8h) .....	52
2.10.40. PD 管脚驱动选择寄存器 PDSTR (偏移: ACh) .....	53
2.10.41. PC/D 管脚施密特使能寄存器 PC/DSMTR (偏移: B0h) .....	53
2.10.42. PC/D 管脚 OD 使能寄存器 PC/DODR (偏移: B4h) .....	54
2.10.43. PC/D 管脚数模选择寄存器 PC/DADS (偏移: B8h) .....	54
2.10.44. PE 管用寄存器 1/PESEL1(偏移: C0h).....	54
2.10.45. PE 管脚复用寄存器 2/PESEL2(偏移: C4h).....	56
2.10.46. PF 管脚复用寄存器 1/PFSEL1(偏移: C8h).....	57
2.10.47. PE/F 管脚上拉寄存器 PE/FPUR (偏移: D0h).....	59
2.10.48. PE/F 管脚下拉寄存器/PE/FPDR (偏移: D4h).....	59
2.10.49. PE 管脚驱动选择寄存器 PESTR (偏移: D8h) .....	59
2.10.50. PF 管脚驱动选择寄存器 PFSTR (偏移: DCh) .....	60
2.10.51. PE/F 管脚施密特使能寄存器 PE/FSMTR (偏移: E0h) .....	60
2.10.52. PE/F 管脚 OD 使能寄存器 PE/FODR (偏移: E4h) .....	60
2.10.53. PE/F 管脚数模选择寄存器 PE/FADS (偏移: E8h) .....	61
3. 复位和时钟单元.....	62
3.1. 复位源.....	62
3.1.1. 上电复位: .....	63
3.1.2. BOR(Brown Out Reset)复位: .....	63
3.1.3. POR12 复位: .....	63
3.1.4. 外部引脚复位: .....	63
3.1.5. EFC 复位: .....	64
3.1.6. LVD 复位: .....	64
3.1.7. LOCKUP 复位: .....	64
3.1.8. Soft Reset: .....	64
3.1.9. SYSREQRST: .....	64
3.1.10. IWDT 复位: .....	64
3.1.11. WDT 复位: .....	64
3.1.12. RTC 域复位: .....	64

3.2. 时钟结构图 .....	65
3.3. 系统时钟选择 .....	65
3.4. 时钟输出 .....	66
4. 嵌套矢量中断控制器 (NVIC) .....	67
4.1. 主要特性 .....	67
4.2. 中断源 .....	67
5. 外部中断/事件控制器 (EXTI) .....	70
5.1. 系统框图 .....	70
5.2. 主要特性 .....	70
5.3. 触发源 .....	70
5.4. 唤醒管理 .....	72
5.5. 功能说明 .....	72
5.5.1. EXTI 中断 .....	72
5.5.2. EXTI 中断/事件唤醒 .....	73
5.6. EXTI 寄存器 .....	73
5.6.1. 中断使能寄存器/IENR (偏移: 00h) .....	73
5.6.2. 事件使能寄存器/EENR (偏移: 04h) .....	74
5.6.3. 上升沿触发使能寄存器/RTENR (偏移: 08h) .....	74
5.6.4. 下降沿触发使能寄存器/FTENR (偏移: 0Ch) .....	74
5.6.5. 软件中断事件寄存器/SWIER (偏移: 10h) .....	75
5.6.6. 中断挂起寄存器/PDR (偏移: 14h) .....	75
5.6.7. 外部中断配置寄存器 1/EXTICR1 (偏移: 18h) .....	75
5.6.8. 外部中断配置寄存器 2/EXTICR2 (偏移: 1Ch) .....	77
6. 片上 Flash 控制器 (EFC) .....	79
6.1. 概述 .....	79
6.2. 主要特性 .....	79
6.3. 功能描述 .....	79
6.3.1. 主区闪存结构 .....	79
6.3.2. 储存地址映射 .....	80
6.3.3. 读操作 .....	80
6.3.4. 闪存编程操作 .....	80
6.3.5. 页擦除 .....	82
6.4. 寄存器描述 .....	82
6.4.1. 控制寄存器 EFC_CTRL(偏移: 00h) .....	83
6.4.2. 写擦安全寄存器 EFC_SEC (偏移: 04h) .....	84
6.4.3. tERASE 等待周期寄存器 (EFC_TERASE) (偏移: 0ch) .....	84
6.4.4. tPROG 等待周期寄存器 (EFC_TPROG) (偏移: 10h) .....	84
6.4.5. 状态寄存器 EFC_STATUS (偏移: 14h) .....	85
6.4.6. 中断状态寄存器 EFC_INTSTATUS (偏移: 18h) .....	85
6.4.7. 中断使能寄存器 EFC_INTEN (偏移: 1ch) .....	86
7. DMA 控制器 (DMAC) .....	87
7.1. 概述 .....	87
7.2. 主要特性 .....	87
7.3. 结构框图 .....	88

7.4. 功能描述 .....	88
7.4.1. DMA 操作 .....	88
7.4.2. 外设握手 .....	88
7.4.3. 仲裁 .....	89
7.4.4. 地址生成 .....	89
7.4.5. 链表功能 .....	89
7.4.6. 存储器到存储器模式 .....	89
7.4.7. 突发模式 .....	90
7.4.8. 中断 .....	90
7.4.9. 目标外设和源外设请求号 .....	91
7.5. 寄存器描述 .....	93
7.5.1. 中断状态寄存器 DMACIntStatus (偏移: 00h) .....	93
7.5.2. 传输完成中断寄存器 DMACIntTCStatus (偏移: 04h) .....	94
7.5.3. 传输完成中断清除寄存器 DMACIntTCClr (偏移: 08h) .....	94
7.5.4. 传输错误中断寄存器 DMACIntErrStatus (偏移: 0Ch) .....	94
7.5.5. 传输错误中断清除寄存器 DMACIntErrClr (偏移: 10h) .....	95
7.5.6. 传输完成原始中断寄存器 DMACRawIntTCStatus (偏移: 14h) .....	95
7.5.7. 传输错误原始中断寄存器 DMACRawIntErrStatus (偏移: 18h) .....	95
7.5.8. 通道使能状态寄存器 DMACEnChStatus (偏移: 1Ch) .....	95
7.5.9. DMAC 配置寄存器 DMACConfig (偏移: 30h) .....	96
7.5.10. 同步寄存器 DMACSyncLo (偏移: 34h) .....	96
7.5.11. 同步寄存器 DMACSyncHi (偏移: 38h) .....	96
7.5.12. 源通道地址寄存器 DMACCxSrcAddr (偏移: 100h, 120h, 140h, 160h, 180h, 1A0h, 1C0h, 1E0h) .....	97
7.5.13. 目标通道地址寄存器 DMACCxDestAddr (偏移: 104h, 124h, 144h, 164h, 184h, 1A4h, 1C4h, 1E4h) .....	97
7.5.14. 通道链接表寄存器 DMACCxLLI (偏移: 108h, 128h, 148h, 168h, 188h, 1A8h, 1C8h, 1E8h) .....	97
7.5.15. 通道控制寄存器 DMACCxCtrl (偏移: 10Ch, 12Ch, 14Ch, 16Ch, 18Ch, 1ACh, 1CCh, 1ECh) .....	98
7.5.16. 通道配置寄存器 DMACCxConfig (偏移: 110h, 130h, 150h, 170h, 190h, 1B0h, 1D0h, 1F0h) .....	99
7.6. 使用说明 .....	100
7.6.1. DMA 优先级 .....	100
7.6.2. 软件注意事项 .....	100
7.6.3. DMAC 使用流程 .....	101
7.6.4. DMAC 链表使用流程 .....	101
8. 高级定时器 (TIM1) .....	102
8.1. 概述 .....	102
8.2. 主要特性 .....	102
8.3. 功能描述 .....	103
8.3.1. 定时器基本单元 .....	103
8.3.2. 定时器计数模式 .....	104
8.3.3. 预分频器 .....	105
8.3.4. 重复计数器 .....	106
8.3.5. 时钟源选择 .....	107

8.3.6. 捕获比较通道 .....	108
8.3.7. 强制输出模式 .....	112
8.3.8. PWM 模式 .....	112
8.3.9. 互补输出和死区插入 .....	115
8.3.10. 刹车功能 .....	116
8.3.11. 六步 PWM 输出 .....	119
8.3.12. 正交编码器模式 .....	120
8.3.13. 与霍尔传感器的接口 .....	121
8.3.14. 单脉冲模式 .....	123
8.3.15. 定时器互连 .....	125
8.3.16. DMA 功能 .....	128
8.4. 寄存器描述 .....	129
1.1.1. TIM1 控制寄存器 1 (TIM1_CR1 偏移: 0x00) .....	130
1.1.2. TIM1 控制寄存器 2 (TIM1_CR2 偏移: 0x04) .....	132
1.1.3. TIM1 从模式控制寄存器 (TIM1_SMCR 偏移: 0x08) .....	134
1.1.4. TIM1 DMA/中断使能寄存器 (TIM1_DIER 偏移: 0x0C) .....	137
1.1.5. TIM1 状态寄存器 (TIM1_SR 偏移: 0x10) .....	139
1.1.6. TIM1 事件产生寄存器 (TIM1_EGR 偏移: 0x14) .....	141
1.1.7. TIM1 捕获/比较模式寄存器 1 (TIM1_CCMR1 偏移: 0x18) .....	142
1.1.8. TIM1 捕获/比较模式寄存器 2 (TIM1_CCMR2 偏移: 0x1C) .....	146
1.1.9. TIM1 捕获/比较使能寄存器 (TIM1_CCER 偏移: 0x20) .....	148
1.1.10. TIM1 计数器 (TIM1_CNT 偏移: 0x24) .....	152
1.1.11. TIM1 预分频器 (TIM1_PSC 偏移: 0x28) .....	152
1.1.12. TIM1 自动加载寄存器 (TIM1_ARR 偏移: 0x2c) .....	152
1.1.13. TIM1 重复计数寄存器 (TIM1_RCR 偏移: 0x30) .....	153
1.1.14. TIM1 捕获/比较寄存器 1 (TIM1_CCR1 偏移: 0x34) .....	153
1.1.15. TIM1 捕获/比较寄存器 2 (TIM1_CCR2 偏移: 0x38) .....	154
1.1.16. TIM1 捕获/比较寄存器 3 (TIM1_CCR3 偏移: 0x3C) .....	154
1.1.17. TIM1 捕获/比较寄存器 4 (TIM1_CCR4 偏移: 0x40) .....	155
1.1.18. TIM1 刹车和死区寄存器 (TIM1_BDTR 偏移: 0x44) .....	155
1.1.19. TIM1 DMA 控制寄存器 (TIM1_DCR 偏移: 0x48) .....	158
1.1.20. TIM1 连续模式的 DMA 地址 (TIM1_DMAR 偏移: 0x4C) .....	159
1.1.21. TIM1 复用功能选择寄存器 (TIM1_AF1 偏移: 0x60) .....	159
1.1.22. TIM1 输入选择寄存器 (TIM1_TISEL 偏移: 0x68) .....	160
1.1.23. TIM1 DMA 请求类型选择寄存器 (TIM1_DBER 偏移: 0x6C) .....	160
9. 通用定时器 (TIM2) .....	162
9.1. 概述 .....	162
9.2. 主要特性 .....	162
9.3. 功能描述 .....	163
9.3.1. 计数单元 .....	163
9.3.2. 预分频器 .....	163
9.3.3. 时钟源选择 .....	163
9.3.4. 捕获比较通道 .....	164
9.3.5. 强制输出模式 .....	164

9.3.6. PWM 模式.....	164
9.3.7. 正交编码器模式.....	164
9.3.8. 与霍尔传感器的接口 .....	164
9.3.9. 单脉冲模式.....	164
9.3.10. 定时器互连 .....	164
9.3.11. DMA 功能 .....	164
9.4. 寄存器描述 .....	164
1.1.24. TIM2 控制寄存器 1 (TIM2_CR1 偏移: 0x00) .....	165
1.1.25. TIM2 控制寄存器 2 (TIM2_CR2 偏移: 0x04) .....	167
1.1.26. TIM2 从模式控制寄存器 (TIM2_SMCR 偏移: 0x08) .....	169
1.1.27. TIM2 DMA/中断使能寄存器 (TIM2_DIER 偏移: 0x0C) .....	172
1.1.28. TIM2 状态寄存器 (TIM2_SR 偏移: 0x10) .....	173
1.1.29. TIM2 事件产生寄存器 (TIM2_EGR 偏移: 0x14) .....	175
1.1.30. TIM2 捕获/比较模式寄存器 1 (TIM2_CCMR1 偏移: 0x18) .....	176
1.1.31. TIM2 捕获/比较模式寄存器 2 (TIM2_CCMR2 偏移: 0x1C) .....	180
1.1.32. TIM2 捕获/比较使能寄存器 (TIM2_CCER 偏移: 0x20) .....	182
1.1.33. TIM2 计数器 (TIM2_CNT 偏移: 0x24) .....	184
1.1.34. TIM2 预分频器 (TIM2_PSC 偏移: 0x28) .....	184
1.1.35. TIM2 自动重装载寄存器 (TIM2_ARR 偏移: 0x2C) .....	184
1.1.36. TIM2 捕获/比较寄存器 1 (TIM2_CCR1 偏移: 0x34) .....	185
1.1.37. TIM2 捕获/比较寄存器 2 (TIM2_CCR2 偏移: 0x38) .....	185
1.1.38. TIM2 捕获/比较寄存器 3 (TIM2_CCR3 偏移: 0x3C) .....	186
1.1.39. TIM2 捕获/比较寄存器 4 (TIM2_CCR4 偏移: 0x40) .....	186
1.1.40. TIM2 DMA 控制寄存器 (TIM2_DCR 偏移: 0x48) .....	186
1.1.41. TIM2 连续模式的 DMA 地址 (TIM2_DMAR 偏移: 0x4C) .....	188
1.1.42. TIM2 复用功能选择寄存器 (TIM2_AF1 偏移: 0x60) .....	188
1.1.43. TIM2 输入选择寄存器 (TIM2_TISEL 偏移: 0x68) .....	188
1.1.44. TIM2 DMA 请求类型选择寄存器 (TIM2_DBER 偏移: 0x6C) .....	189
10. 通用定时器 (TIM3/TIM4) .....	190
10.1. 概述.....	190
10.2. 主要特性.....	190
10.3. 功能描述.....	191
10.3.1. 计数单元.....	191
10.3.2. 预分频器.....	191
10.3.3. 时钟源选择 .....	191
10.3.4. 捕获比较通道.....	191
10.3.5. 强制输出模式.....	192
10.3.6. PWM 模式 .....	192
10.3.7. 正交编码器模式.....	192
10.3.8. 与霍尔传感器的接口 .....	192
10.3.9. 单脉冲模式 .....	192
10.3.10. 定时器互连.....	192
10.3.11. DMA 功能.....	192
10.4. 寄存器描述.....	192

10.4.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)	194
10.4.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)	196
10.4.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08)	198
10.4.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)	201
10.4.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)	202
10.4.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)	204
10.4.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18)	206
10.4.8. TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2 偏移: 0x1C)	210
10.4.9. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20)	212
10.4.10. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	214
10.4.11. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	214
10.4.12. TIMx 自动重装载寄存器 (TIMx_ARR 偏移: 0x2C)	214
10.4.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)	214
10.4.14. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)	215
10.4.15. TIMx 捕获/比较寄存器 3 (TIMx_CCR3 偏移: 0x3C)	215
TIMx 捕获/比较寄存器 4 (TIMx_CCR4 偏移: 0x40)	216
10.4.16. TIMx DMA 控制寄存器 (TIMx_DCR 偏移: 0x48)	216
10.4.17. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)	218
10.4.18. TIMx 复用功能选择寄存器 (TIMx_AF1 偏移: 0x60)	218
10.4.19. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)	218
10.4.20. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)	219
11. 基本定时器 (TIM6/TIM7)	220
11.1. 概述	220
11.2. 主要特性	220
11.3. 功能描述	220
11.3.1. 计数单元	220
11.3.2. 预分频器	221
11.3.3. 时钟源选择	221
11.3.4. 定时器互连	221
11.3.5. DMA 功能	221
11.4. 寄存器描述	221
11.4.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00)	222
11.4.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04)	224
11.4.3. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C)	225
11.4.4. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10)	226
11.4.5. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14)	227
11.4.6. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	227
11.4.7. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	228
11.4.8. TIMx 自动重装载寄存器 (TIMx_ARR 偏移: 0x2C)	228
12. 通用定时器 (TIM14)	229
12.1. 概述	229
12.2. 主要特性	229
12.3. 功能描述	230
12.3.1. 计数单元	230

12.3.2. 预分频器 .....	230
12.3.3. 时钟源选择 .....	230
12.3.4. 捕获比较通道 .....	230
12.3.5. 强制输出模式 .....	230
12.3.6. PWM 模式 .....	230
12.3.7. 单脉冲模式 .....	231
12.3.8. 定时器互连 .....	231
12.3.9. DMA 功能 .....	231
12.4. 寄存器描述 .....	231
12.4.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00) .....	232
12.4.2. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C) .....	234
12.4.3. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10) .....	234
12.4.4. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14) .....	236
12.4.5. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18) .....	237
12.4.6. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20) .....	240
12.4.7. TIMx 计数器 (TIMx_CNT 偏移: 0x24) .....	241
12.4.8. TIMx 预分频器 (TIMx_PSC 偏移: 0x28) .....	242
12.4.9. TIMx 自动重载寄存器 (TIMx_ARR 偏移: 0x2C) .....	242
12.4.10. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34) .....	242
13. 通用定时器 (TIM15/TIM16/TIM17) .....	243
13.1. 概述 .....	243
13.2. 主要特性 .....	243
13.3. 功能描述 .....	244
13.3.1. 计数单元 .....	244
13.3.2. 预分频器 .....	245
13.3.3. 时钟源选择 .....	245
13.3.4. 捕获比较通道 .....	245
13.3.5. 强制输出模式 .....	245
13.3.6. PWM 模式 .....	245
13.3.7. 互补输出和死区插入 .....	245
13.3.8. 刹车功能 .....	245
13.3.9. 六步 PWM 输出 .....	245
13.3.10. 单脉冲模式 .....	245
13.3.11. 定时器互连 .....	246
13.3.12. DMA 功能 .....	246
13.4. 寄存器描述 .....	246
13.4.1. TIMx 控制寄存器 1 (TIMx_CR1 偏移: 0x00) .....	248
13.4.2. TIMx 控制寄存器 2 (TIMx_CR2 偏移: 0x04) .....	250
13.4.3. TIMx 从模式控制寄存器 (TIMx_SMCR 偏移: 0x08) .....	251
13.4.4. TIMx DMA/中断使能寄存器 (TIMx_DIER 偏移: 0x0C) .....	253
13.4.5. TIMx 状态寄存器 (TIMx_SR 偏移: 0x10) .....	254
13.4.6. TIMx 事件产生寄存器 (TIMx_EGR 偏移: 0x14) .....	257
13.4.7. TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1 偏移: 0x18) .....	258
13.4.8. TIMx 捕获/比较使能寄存器 (TIMx_CCER 偏移: 0x20) .....	262

13.4.9. TIMx 计数器 (TIMx_CNT 偏移: 0x24)	265
13.4.10. TIMx 预分频器 (TIMx_PSC 偏移: 0x28)	267
13.4.11. TIMx 自动重装载寄存器 (TIMx_ARR 偏移: 0x2C)	267
13.4.12. TIMx 重复计数寄存器 (TIMx_RCR 偏移: 0x30)	268
13.4.13. TIMx 捕获/比较寄存器 1 (TIMx_CCR1 偏移: 0x34)	269
13.4.14. TIMx 捕获/比较寄存器 2 (TIMx_CCR2 偏移: 0x38)	270
13.4.15. TIMx 刹车和死区寄存器 (TIMx_BDTR 偏移: 0x44)	271
13.4.16. TIMx DMA 控制寄存器 (TIMx_DCR 偏移: 0x48)	273
13.4.17. TIMx 连续模式的 DMA 地址 (TIMx_DMAR 偏移: 0x4C)	275
13.4.18. TIMx 复用功能选择寄存器 (TIMx_AF1 偏移: 0x60)	276
13.4.19. TIMx 输入选择寄存器 (TIMx_TISEL 偏移: 0x68)	276
13.4.20. TIMx DMA 请求类型选择寄存器 (TIMx_DBER 偏移: 0x6C)	277
14. 看门狗 (WDT)	278
14.1. 概述	278
14.2. 主要特性	278
14.3. 功能描述	279
14.3.1. 功能框图	279
14.3.2. 看门狗模式	279
14.3.3. 中断清除时限	279
14.4. 寄存器描述	280
14.4.1. WDTLOAD 加载寄存器(偏移: 00h)	280
14.4.2. WDTCOUNT 当前计数寄存器(偏移: 04h)	280
14.4.3. WDTCTRL 控制寄存器(偏移: 08h)	280
14.4.4. WDTFEED 喂狗寄存器(偏移: 0Ch)	281
14.4.5. WDTINTCLRTIME 中断清除时限寄存器 (偏移: 10h)	281
14.4.6. WDTRIS 原始中断状态寄存器(偏移: 14h)	281
14.5. 使用流程	282
14.5.1. 定时器溢出产生中断	282
14.5.2. 定时器溢出产生复位	282
15. 独立看门狗 (IWDT)	283
15.1. 概述	283
15.2. 主要特性	283
15.3. 功能描述	283
15.3.1. 功能框图	283
15.3.2. 寄存器访问保护	284
15.3.3. 窗口选项	284
15.3.4. 唤醒功能	284
15.4. 寄存器描述	285
15.4.1. IWDT_CMDR 命令寄存器(偏移: 00h)	285
15.4.2. IWDT_PR 预分频寄存器(偏移: 04h)	286
15.4.3. IWDT_RLR 重装载寄存器(偏移: 08h)	286
15.4.4. IWDT_SR 状态寄存器(偏移: 0Ch)	287
15.4.5. IWDT_WINR 窗口寄存器(偏移: 10h)	288
15.4.6. IWDT_WUTR 窗口寄存器(偏移: 14h)	288

15.5. 使用流程.....	289
15.5.1. 不带窗口.....	289
15.5.2. 带窗口.....	289
16. 实时时钟 (RTC) .....	290
16.1. 概述.....	290
16.2. 主要特性.....	290
16.3. RTC 功能描述.....	290
16.3.1. RTC 时间设置.....	291
16.3.2. RTC 时间读取.....	292
16.3.3. 闹钟功能.....	292
16.3.4. 时钟误差补偿.....	293
16.3.5. FSEL_OUT.....	294
16.3.6. 周期中断唤醒.....	294
16.3.7. 侵入检测和备份寄存器 .....	294
16.4. 寄存器描述.....	295
16.4.1. RTC_WP 写保护寄存器(偏移: 00h).....	296
16.4.2. RTC_IE 中断使能寄存器(偏移: 04h).....	296
16.4.3. RTC_SR 中断标志寄存器(偏移: 08h).....	297
16.4.4. RTC_SEC 秒计数寄存器(偏移: 0Ch).....	299
16.4.5. RTC_MIN 时计数寄存器(偏移: 10h).....	299
16.4.6. RTC_HOUR 分计数寄存器(偏移: 14h).....	300
16.4.7. RTC_DAY 日计数寄存器(偏移: 18h).....	300
16.4.8. RTC_WEEK 周计数寄存器(偏移: 1Ch).....	300
16.4.9. RTC_MONTH 周计数寄存器(偏移: 20h).....	300
16.4.10. RTC_YEAR 年计数寄存器(偏移: 24h).....	301
16.4.11. RTC_ALM 闹钟寄存器(偏移: 28h).....	301
16.4.12. RTC_CR 控制寄存器(偏移: 2Ch).....	301
16.4.13. RTC_ADJUST 时钟误差补偿寄存器(偏移: 30h).....	304
16.4.14. RTC_CLKSTAMP1 时间戳 1(偏移: 44h).....	304
16.4.15. RTC_CALSTAMP1 日历戳 1(偏移: 48h).....	304
16.4.16. RTC_CLKSTAMP2 时间戳 2(偏移: 4Ch).....	305
16.4.17. RTC_CALSTAMP2 日历戳 2(偏移: 50h).....	305
16.4.18. RTC_BAKUP0~4 备份寄存器 0~4(偏移: 70~80h).....	305
17. 待机区电源管理 (RPMU) .....	307
17.1. 概述.....	307
17.2. 功能描述.....	307
17.2.1. 唤醒引脚配置.....	307
17.2.2. RTC Clock 管理.....	308
17.2.3. RPMU 区引脚控制.....	308
17.3. 寄存器描述.....	308
17.3.1. RPMU_CR/RTC 域控制寄存器(偏移: 00h).....	308
17.3.2. RPMU_SR/RTC 域状态寄存器(偏移: 04h).....	310
17.3.3. RPMU_IOSEL/RTC 域 IO 复用寄存器(偏移: 08h).....	311
17.3.4. RPMU_IOCRR/RTC 域 IO 控制寄存器(偏移: Ch).....	312

17.3.5. RPMU_ANACR/ RTC_PMU 模拟控制寄存器 (偏移: 10h).....	314
17.3.6. RPMU_CR2/RTC 域控制寄存器 2(偏移: 14h) .....	315
18. 通用输入输出接口 (GPIO) .....	317
18.1. 概述.....	317
18.2. 主要特性.....	317
18.3. 功能描述.....	317
18.3.1. 结构框图.....	318
18.3.1. 输入功能.....	318
18.3.2. 输出功能.....	318
18.3.3. 复用功能 (AF) .....	319
18.3.4. 开漏、驱动能力、上下拉、管脚复用、数字模拟配置 .....	319
18.3.5. 附加功能.....	319
18.4. 寄存器描述.....	319
18.4.1. 数据方向寄存器 GPIO_DIR(偏移: 00h).....	320
18.4.2. 输出置位寄存器 GPIO_SET(偏移: 08h).....	320
18.4.3. 输出清零寄存器 GPIO_CLR(偏移: 0Ch).....	320
18.4.4. GPIO 输出引脚映射寄存器 GPIO_ODATA(偏移: 10h) .....	321
18.4.5. GPIO 输入引脚映射寄存器 GPIO_IDATA(偏移: 14h).....	321
18.4.6. GPIO 中断使能寄存器 GPIO_IEN(偏移: 18h).....	321
18.4.7. GPIO 中断触发模式寄存器 GPIO_IS(偏移: 1Ch) .....	321
18.4.8. GPIO 中断触发模式寄存器 GPIO_IBE(偏移: 20h).....	322
18.4.9. GPIO 中断触发模式寄存器 GPIO_IEV(偏移: 24h).....	322
18.4.10. GPIO 中断状态清除寄存器 GPIO_IC(偏移: 28h).....	322
18.4.11. GPIO 原始中断状态寄存器 GPIO_RIS(偏移: 2Ch).....	322
18.4.12. GPIO 屏蔽后中断状态寄存器 GPIO_MIS(偏移: 30h).....	323
18.5. 使用流程.....	323
18.5.1. 输入输出 IO .....	323
18.5.2. 中断触发模式.....	323
18.5.3. 清除中断.....	323
19. 串行外设接口 (SPI) .....	324
19.1. 概述.....	324
19.2. 主要特性.....	324
19.3. 结构框图.....	324
19.4. 功能描述.....	325
19.4.1. 时序图.....	325
19.4.2. 时钟波特率设置 .....	329
19.4.3. DMA 请求.....	329
19.4.4. 存储器读取模式 .....	329
19.5. 寄存器描述.....	330
19.5.1. SPI 发送数据寄存器 SPI_TX_DAT(偏移: 00h).....	330
19.5.2. SPI 接收数据寄存器 SPI_RX_DAT(偏移: 00h).....	331
19.5.3. SPI 波特率设置寄存器 SPI_BAUD(偏移: 04h).....	331
19.5.4. SPI 控制寄存器 SPI_CTL(偏移: 08h).....	331
19.5.5. SPI 发送控制寄存器 SPI_TX_CTL(偏移: 0Ch).....	333

19.5.6. SPI 接收控制寄存器 SPI_RX_CTL(偏移: 10h)	333
19.5.7. SPI 中断控制寄存器 SPI_IE(偏移: 14h)	334
19.5.8. SPI 状态寄存器 SPI_STATUS(偏移: 18h)	335
19.5.9. SPI 发送等待寄存器 SPI_TXDelay(偏移: 1Ch)	338
19.5.10. SPI 批量传输数据个数寄存器 SPI_BATCH (偏移: 20h)	338
19.5.11. SPI 从设备选择寄存器 SPI_CS(偏移: 24h)	338
19.5.12. SPI 管脚输出方向 SPI_OUT_EN(偏移: 28h)	339
19.5.13. SPI3 取值控制寄存器 SPI_MEMO_ACC(偏移: 2Ch)	339
19.5.14. SPI3 取值命令寄存器 SPI_CMD(偏移: 30h)	340
19.5.15. SPI3 取值参数寄存器 SPI_PARA(偏移: 34h)	341
19.6. 使用流程	341
19.6.1. SPI 主模式发送	341
19.6.2. SPI 主模式接收	342
19.6.3. SPI 主模式接收时 Dummy 控制位	342
19.6.4. SPI 从模式发送时 Dummy 控制位	342
19.6.5. SPI 从模式接收时 Dummy 控制位	343
19.6.6. SPI 存储器读取	343
19.6.7. SPI 存储器读取 (连读)	344
19.6.8. SPI SRAM 写入	344
20. 通用异步收发器 (UART)	345
20.1. 概述	345
20.2. 主要特性	345
20.3. 结构框图	346
20.4. 功能描述	346
20.4.1. 串口设置	346
20.4.2. CTS 和 RTS 流控功能	347
20.4.3. DMA 请求	348
20.4.4. LIN 总线功能	348
20.4.5. IrDA SIR 功能	349
20.4.6. 单线模式	350
20.5. 寄存器描述	350
20.5.1. 数据寄存器 UART_DR(偏移: 00h)	351
20.5.2. 接收状态寄存器 UART_RSR(偏移: 04h)	352
20.5.3. 标志位寄存器 UART_FR(偏移: 18h)	353
20.5.4. IrDA 低功耗分频因子寄存器 UART_ILPR(偏移: 20h)	354
20.5.5. 整数分频因子寄存器 UART_IBRD(偏移: 24h)	354
20.5.6. 小数分频因子寄存器 UART_FBRD (偏移: 28h)	354
20.5.7. 线控制器寄存器 UART_LCRH (偏移: 2Ch)	354
20.5.8. 控制寄存器 UART_CR (偏移: 30h)	355
20.5.9. FIFO 中断触发寄存器 UART_IFLS(偏移: 34h)	356
20.5.10. 中断使能寄存器 UART_IMSC (偏移: 38h)	357
20.5.11. 原始中断状态寄存器 UART_RIS (偏移: 3Ch)	358
20.5.12. MASK 后的中断状态寄存器 UART_MIS (偏移: 40h)	360
20.5.13. 中断状态清除 UART_ICR (偏移: 44h)	361

20.5.14. DMA 控制寄存器 UART_DMACR (偏移: 48h).....	362
20.5.15. 控制寄存器 2UART_CR2 (偏移: 54h).....	362
20.5.16. 比特计时寄存器 UART_BCNT (偏移: 58h).....	363
20.6. 使用流程.....	363
20.6.1. 串口的发送和接收.....	363
20.6.2. CTS 和 RTS 控制流功能.....	363
20.6.3. LIN 硬件功能支持.....	364
20.6.4. IrDA SIR 功能使用流程.....	364
20.6.5. 单线模式功能使用流程.....	364
21. 内部集成电路总线接口 (I2C).....	365
21.1. 概述.....	365
21.2. 主要特性.....	365
21.3. 结构框图.....	366
21.4. 功能描述.....	366
21.4.1. 模式选择.....	366
21.4.2. I2C 从模式.....	367
21.4.3. I2C 主模式.....	368
21.4.4. TXE 状态.....	370
21.5. 寄存器描述.....	370
21.5.1. I2C 设备地址寄存器 1 I2C_SLAVE_ADDR1(偏移: 0x0h).....	371
21.5.2. I2C 时钟分频寄存器 I2C_CLK_DIV(偏移: 0x04h).....	371
21.5.3. 控制寄存器 I2C_CR(偏移: 0x08h).....	371
21.5.4. 状态寄存器 I2C_SR(偏移: 0x0Ch).....	374
21.5.5. 数据寄存器 I2C_DR(偏移: 0x10h).....	376
21.5.6. I2C 设备地址寄存器 2/3 I2C_SLAVE_ADDR2/3 (偏移: 0x14h).....	377
21.5.7. I2C 滤波寄存器 I2C_FILTER (偏移: 0x1Ch).....	377
21.5.8. I2C 超时配置寄存器 I2C_TIMEOUT(偏移: 0x24h).....	377
21.6. 软件操作流程.....	379
21.6.1. 作为主发送器.....	379
21.6.2. 作为主接收器.....	381
21.6.3. 作为从发送器.....	383
21.6.4. 作为从接收器.....	384
21.6.5. 滤波算法.....	384
22. 音频接口 (I2S).....	385
22.1. 概述.....	385
22.2. 主要特性.....	385
22.3. 功能描述.....	386
22.3.1. I2S 结构框图.....	386
22.3.2. I2S 音频标准.....	386
22.3.3. I2S 时钟.....	395
22.4. 寄存器描述.....	396
22.4.1. I2S 数据寄存器 I2S_DR(偏移 00h).....	397
22.4.2. I2S 控制寄存器 I2S_CR(偏移: 04h).....	397
22.4.3. I2S 时钟预分频寄存器 I2S_PR (偏移: 08h).....	398

22.4.4. I2S DMA/中断使能寄存器 I2S_DIER (偏移: 0Ch)	399
22.4.5. I2S 状态寄存器 I2S_SR (偏移: 10h)	399
22.5. 使用流程	400
22.5.1. 主机发送	401
22.5.2. 主机接收	401
22.5.3. 从机发送	402
22.5.4. 从机接收	402
23. 低功耗串口 (LPUART)	403
23.1. 概述	403
23.2. 主要特性	403
23.3. 功能描述	403
23.3.1. 时钟选择	404
23.3.2. 波特率设置	404
23.3.3. STOP 模式唤醒	405
23.3.4. DMA 请求	406
23.4. 寄存器描述	406
23.4.1. 接收数据寄存器/LPUART_RXDR (偏移: 00h)	406
23.4.2. 发送数据寄存器/LPUART_TXDR (偏移: 04h)	406
23.4.3. 线控寄存器/LPUART_LCR (偏移: 08h)	407
23.4.4. 控制寄存器/LPUART_CR (偏移: 0Ch)	408
23.4.5. 波特率整数部分/LPUART_IBAUD (偏移: 10h)	408
23.4.6. 波特率小数部分/LPUART_FBAUD (偏移: 14h)	409
23.4.7. 中断使能寄存器/LPUART_IE (偏移: 18h)	409
23.4.8. 状态寄存器 LPUART_SR (偏移: 1Ch)	410
23.4.9. 地址寄存器/LPUART_ADDR (偏移: 20h)	411
23.5. 使用流程	411
23.5.1. 串口收发	411
23.5.2. 配置 STOP 唤醒	411
24. 模数转换器 (ADC)	413
24.1. 概述	413
24.2. 主要特性	413
24.3. 结构框图	414
24.4. 功能描述	415
24.4.1. 通道选择	415
24.4.2. 单次转换模式	415
24.4.3. 连续转换模式	416
24.4.4. 间断模式	416
24.4.5. 停止控制	417
24.4.6. 时序图	417
24.4.7. 模拟看门狗	418
24.4.8. 注入通道	418
24.4.9. 可编程的通道采样时间	419
24.4.10. 外部触发转换	419
24.4.11. DMA 请求	420

24.4.12. 温度传感器 .....	420
24.4.13. 差分信号转换 .....	420
24.4.14. 溢出控制 .....	421
24.4.15. 差分模式和有符号数 .....	421
24.4.16. 过采样 .....	421
24.4.17. ADC 中断 .....	422
24.5. 寄存器描述 .....	423
24.5.1. ADC 状态寄存器(ADC_SR 偏移 00h) .....	423
24.5.2. ADC 中断使能寄存器(ADC_IE 偏移 04h) .....	424
24.5.3. ADC 控制寄存器 1 (ADC_CR1 偏移 08h) .....	425
24.5.4. ADC 控制寄存器 2 (ADC_CR2 偏移 0ch) .....	428
24.5.5. ADC 采样时间寄存器 1 (ADC_SMPR1 偏移 10h) .....	430
24.5.6. ADC 采样时间寄存器 2 (ADC_SMPR2 偏移 14h) .....	431
24.5.7. ADC 看门狗高阈值寄存器(ADC_HTR 偏移 18h) .....	432
24.5.8. ADC 看门狗低阈值寄存器(ADC_LTR 偏移 1ch) .....	432
24.5.9. ADC 规则序列寄存器 1(ADC_SQR1 偏移 20h) .....	432
24.5.10. ADC 规则序列寄存器 2(ADC_SQR2 偏移 24h) .....	433
24.5.11. ADC 规则序列寄存器 3(ADC_SQR3 偏移 28h) .....	434
24.5.12. ADC 注入通道寄存器(ADC_JSQR 偏移 2ch) .....	434
24.5.13. ADC 注入数据寄存器(ADC_JDR 偏移 30h) .....	434
24.5.14. ADC 规则数据寄存器(ADC_DR 偏移 34h) .....	434
24.5.15. ADC 单端/差分选择寄存器 (ADC_DIFF 偏移 38h) .....	435
24.5.16. ADC 符号数选择寄存器 (ADC_SIGN 偏移 3ch) .....	436
24.5.17. ADC 温度传感器/REF 寄存器 (ADC_TSREF 偏移 40h) .....	437
24.5.18. ADC 采样时间寄存器 3 (ADC_SMPR3 偏移 44h) .....	438
24.6. 使用流程 .....	438
24.6.1. ADC 操作流程 .....	438
24.6.2. 温度传感器操作 .....	439
25. 数模转换器 (DAC) .....	440
25.1. 概述 .....	440
25.2. 主要特性 .....	440
25.3. 结构框图 .....	440
25.4. 功能描述 .....	442
25.4.1. DAC 通道使能 .....	442
25.4.2. DAC 数据结构 .....	442
25.4.3. DAC 转换和输出电压 .....	443
25.4.4. DAC 触发选择 .....	444
25.4.5. DAC 噪声叠加 .....	444
25.4.6. DMA 请求 .....	446
25.4.7. DAC 并发转换 .....	446
25.4.8. 采样保持模式 .....	446
25.5. 寄存器描述 .....	448
25.5.1. DAC 控制寄存器(DAC_CR 偏移 00h) .....	448
25.5.2. DAC 软件触发寄存器(DAC_SWTRIGR 偏移 04h) .....	451

25.5.3. DAC 通道 1 12 位右对齐数据保持寄存器(DAC_DHR12R1 偏移 08h)	452
25.5.4. DAC 通道 1 12 位左对齐数据保持寄存器(DAC_DHR12L1 偏移 0ch)	452
25.5.5. DAC 通道 1 8 位右对齐数据保持寄存器(DAC_DHR8R1 偏移 10h)	452
25.5.6. DAC 通道 2 12 位右对齐数据保持寄存器(DAC_DHR12R2 偏移 14h)	452
25.5.7. DAC 通道 2 12 位左对齐数据保持寄存器(DAC_DHR12L2 偏移 18ch)	453
25.5.8. DAC 通道 2 8 位右对齐数据保持寄存器(DAC_DHR8R2 偏移 1ch)	453
25.5.9. 双 DAC 12 位右对齐数据保持寄存器(DAC_DHR12RD 偏移 20h)	453
25.5.10. 双 DAC 12 位左对齐数据保持寄存器(DAC_DHR12LD 偏移 24h)	454
25.5.11. 双 DAC 8 位右对齐数据保持寄存器(DAC_DHR8RD 偏移 28h)	454
25.5.12. DAC 通道 1 数据输出寄存器(DAC_DOR1 偏移 2ch)	454
25.5.13. DAC 通道 2 数据输出寄存器(DAC_DOR2 偏移 30h)	454
25.5.14. DAC 状态寄存器(DAC_SR 偏移 34h)	455
25.5.15. DAC 校准控制寄存器(DAC_CCR 偏移 38h)	456
25.5.16. DAC 模式控制寄存器(DAC_MCR 偏移 3Ch)	456
25.5.17. DAC 通道 1 采样时间寄存器(DAC_SHSR1 偏移 40h)	456
25.5.18. DAC 通道 2 采样时间寄存器(DAC_SHSR2 偏移 44h)	457
25.5.19. DAC 保持时间寄存器(DAC_SHHR 偏移 48h)	457
25.5.20. DAC 刷新时间寄存器寄存器(DAC_SHRR 偏移 4ch)	457
25.6. 使用流程	458
25.6.1. 单个 DAC 操作流程	458
25.6.2. 双 DAC 并发模式流程	458
26. 运算放大器 (OPAMP)	459
26.1. 概述	459
26.2. 主要特性	459
26.3. 结构框图	459
26.4. 功能描述	460
26.4.1. 操作模式和校准	460
26.4.2. 工作模式和负端输入	460
26.4.3. 运放外置模式 (SA 模式)	460
26.4.4. 运放单位增益模式 (UG 模式)	461
26.4.5. 可编程增益放大模式 (PGA 模式)	461
26.5. 寄存器描述	463
26.5.1. OPAMP1 控制寄存器(OPAMP1_CSR 偏移 00h)	463
26.5.2. OPAMP2 控制寄存器(OPAMP2_CSR 偏移 04h)	465
26.5.3. OPAMP3 控制寄存器(OPAMP3_CSR 偏移 08h)	467
26.6. 使用流程	469
26.6.1. 普通使用流程	469
26.6.2. 使用修调功能流程	469
27. 模拟比较器 (COMP)	470
27.1. 概述	470
27.2. 主要特性	470
27.3. 结构框图	470
27.4. 功能描述	471
27.4.1. 负端输入	471

27.4.2. 滤波 .....	471
27.4.3. 迟滞比较 .....	471
27.4.4. 切断 .....	471
27.4.5. STOP 模式唤醒 .....	471
27.4.6. 窗口模式级联 .....	472
27.5. 寄存器描述 .....	473
27.5.1. COMP1 控制寄存器(COMP_CR1 偏移 00h) .....	473
27.5.2. COMP2 控制寄存器(COMP_CR2 偏移 04h) .....	475
27.5.3. COMP 状态寄存器(COMP_SR 偏移 08h) .....	477
27.6. 使用流程 .....	478
28. 控制器区域网络 (CAN) .....	479
28.1. 概述 .....	479
28.2. 主要特性 .....	479
28.3. 功能描述 .....	480
28.3.1. 功能框图 .....	480
28.3.2. 操作模式 .....	480
28.3.3. 发送 .....	482
28.3.4. 接收 .....	484
28.3.5. 自我接收 .....	485
28.3.6. 接收过滤 .....	485
28.3.7. 波特率 .....	487
28.3.8. 仲裁器 .....	489
28.3.9. 错误处理 .....	489
28.3.10. 睡眠模式 .....	489
28.4. 寄存器描述 .....	489
28.4.1. 模式寄存器 CAN_MOD(偏移: 00h) .....	490
28.4.2. 命令寄存器 CAN_CMR(偏移: 04h) .....	491
28.4.3. 状态寄存器 CAN_SR(偏移: 08h) .....	492
28.4.4. 中断寄存器 CAN_IR(偏移: 0Ch) .....	493
28.4.5. 中断使能寄存器 CAN_IER(偏移: 10h) .....	494
28.4.6. 时序寄存器 CAN_BTR0(偏移: 18h) .....	495
28.4.7. 时序寄存器 CAN_BTR1(偏移: 1Ch) .....	495
28.4.8. 输出寄存器 CAN_OCR(偏移: 20h) .....	496
28.4.9. 仲裁失败位置获取寄存器 CAN_ALC(偏移: 2Ch) .....	496
28.4.10. 错误代码获取寄存器 CAN_ECC(偏移: 30h) .....	496
28.4.11. 错误报警寄存器 CAN_EWLR(偏移: 34h) .....	497
28.4.12. 接收错误寄存器 CAN_RXERR(偏移: 38h) .....	498
28.4.13. 发送错误寄存器 CAN_TXERR(偏移: 3Ch) .....	498
28.4.14. 发送缓存写寄存器 CAN_TXBUFFx(写偏移: 0x40~0x70) .....	498
28.4.15. 接收缓存读寄存器 CAN_RXBUFFx(读偏移: 0x40~0x70) .....	498
28.4.16. 接收过滤寄存器 CAN_ACRx(偏移: 0x40~0x4C) .....	498
28.4.17. 接收过滤屏蔽寄存器 CAN_AMRx(偏移: 0x50~0x5C) .....	499
28.4.18. 接收报文计数寄存器 CAN_RMC(偏移: 0x74) .....	499
28.4.19. 接收缓存起始地址寄存器 CAN_RBSA(偏移: 0x78) .....	499

28.4.20. 时钟输出分频寄存器 CAN_CDR(偏移: 0x7C) .....	499
28.4.21. 接收 FIFO 访问寄存器 CAN_RXFIFO(偏移: 0x80~0x17C).....	500
28.4.22. 发送 FIFO 访问寄存器 CAN_TXFIFO(偏移: 0x180~0x1B0).....	500
28.5. 使用流程.....	500
28.5.1. CAN 发送和接收 .....	500
29. 通用串行总线 USB.....	502
29.1. 概述.....	502
29.2. 主要特性.....	502
29.3. 功能描述.....	503
29.3.1. 中断状态和控制寄存器 .....	503
29.3.2. 地址设置 Set Address .....	503
29.3.3. 远程唤醒.....	503
29.3.4. 令牌包与数据包 CRC 出错选择性回复 NAK.....	504
29.3.5. 数据包长度超过 64 Byte .....	504
29.3.6. 包丢失 EOP .....	504
29.3.7. IN 操作 ACK 超时下次 IN 操作回复 NAK.....	504
29.3.8. FIFO 访问与 Memory 访问 .....	505
29.4. 寄存器描述.....	505
29.4.1. 工作模式寄存器 WORKING_MODE (偏移: 0x0h) .....	506
29.4.2. EP0 传输控制寄存器 EP0CSR (偏移: 0x4h) .....	509
29.4.3. EP1 传输控制寄存器 EP1CSR (偏移: 0x8h) .....	511
29.4.4. EP2 传输控制寄存器 EP2CSR (偏移: 0xch) .....	514
29.4.5. EP3 传输控制寄存器 EP3CSR (偏移: 0x10h) .....	517
29.4.6. EP4 传输控制寄存器 EP4CSR (偏移: 0x14h) .....	519
29.4.7. USB 地址寄存器 USB_ADDR (偏移: 0x18h) .....	522
29.4.8. SETUP 数据包寄存器 SETUP_0_3_DATA (偏移: 0x1ch) .....	522
29.4.9. SETUP 数据包寄存器 SETUP_4_7_DATA (偏移: 0x20h) .....	522
29.4.10. End Point 地址配置寄存器 EP_ADDR (偏移: 0x24h) .....	522
29.4.11. 总线包 PID 寄存器 CURRENT_PID (偏移: 0x28h) .....	523
29.4.12. Frame Number 寄存器 CURRENT_FRAME_NUMBER (偏移: 0x2ch) .....	523
29.4.13. CRC 错误 Counter 寄存器 CRC_ERROR_CNT (偏移: 0x30h) .....	523
29.4.14. 探测时间寄存器 USB_STATUS_DETECT_CNT (偏移: 0x34h) .....	523
29.4.15. EP0 发送数据数目寄存器 EP0SENDBN (偏移: 0x40h) .....	523
29.4.16. EP1 发送数据数目寄存器 EP1SENDBN (偏移: 0x44h) .....	524
29.4.17. EP2 发送数据数目寄存器 EP2SENDBN (偏移: 0x48h) .....	524
29.4.18. EP3 发送数据数目寄存器 EP3SENDBN (偏移: 0x4ch) .....	524
29.4.19. EP4 发送数据数目寄存器 EP4SENDBN (偏移: 0x50h) .....	524
29.4.20. EP0 FIFO 访问入口 EP0FIFO (偏移: 0x100h) .....	524
29.4.21. EP1 FIFO 访问入口 EP1FIFO (偏移: 0x104h) .....	525
29.4.22. EP2 FIFO 访问入口 EP2FIFO (偏移: 0x108h) .....	525
29.4.23. EP3 FIFO 访问入口 EP3FIFO (偏移: 0x10ch) .....	525
29.4.24. EP4 FIFO 访问入口 EP4FIFO (偏移: 0x110h) .....	525
29.4.25. 状态寄存器 INT_STAT_RAW (偏移: 0xFFE4h) .....	525
29.4.26. 中断使能寄存器 INT_EN (偏移: 0xFFE8h) .....	527

29.4.27. 中断清除寄存器 INT_CLR (偏移: 0xFFF0h) .....	529
29.5. 软件流程.....	533
29.5.1. USB 连接 .....	533
29.5.2. SETUP 数据和 EPO 控制传输数据.....	533
29.5.3. Endpoint In 传输 .....	534
29.5.4. Endpoint Out 传输 .....	534
30. CRC 计算单元.....	536
30.1. 概述.....	536
30.2. 主要特性.....	536
30.3. 寄存器描述.....	536
30.3.1. 数据寄存器/ CRC_DATA (偏移: 00h).....	536
30.3.2. 控制寄存器/ CRC_CTRL (偏移: 04h).....	536
30.3.3. 初始值寄存器/ CRC_INIT (偏移: 08h).....	537
30.3.4. 结果异或值寄存器/ CRC_OUTXOR (偏移: 10h).....	537
30.3.5. 多项式寄存器/ CRC_POLY (偏移: 14h).....	538
30.3.6. 独立数据寄存器/CRC_FDATA (偏移: 18h).....	538
30.4. 使用流程.....	538
31. 算法库.....	539
31.1. 数据类型.....	539
31.2. HRNG.....	539
31.2.1. 主要特性 .....	539
31.2.2. 库文件说明 .....	539
31.2.3. 函数说明.....	539
31.2.4. 注意事项.....	540
31.3. AES .....	540
31.3.1. 主要特性 .....	540
31.3.2. 库文件说明 .....	540
31.3.3. 函数说明.....	540
31.3.4. 注意事项.....	542
31.4. HASH .....	543
31.4.1. 主要特性 .....	543
31.4.2. 库文件说明 .....	543
31.4.3. 函数说明.....	543
31.4.4. 注意事项: .....	543
31.5. FAU.....	544
31.5.1. 主要特性 .....	544
31.5.2. 库文件说明 .....	544
31.5.3. 函数说明.....	544
31.5.4. 注意事项.....	545

## 图目录

图 2-1 处理器结构框图 .....	2
图 2-2 处理器寄存器组 .....	2

图 2-3 芯片系统架构图 .....	3
图 2-4 处理器 Memory Map .....	4
图 2-5 芯片启动模式选择 .....	7
图 2-6 芯片电源域 .....	8
图 3-1 复位关系图 .....	63
图 3-2 时钟结构图 .....	65
图 5-1 EXTI 系统框图 .....	70
图 6-1 eFlash 地址映射 .....	80
图 6-2 闪存编程操作流程 .....	81
图 6-3 页擦除操作流程 .....	82
图 7-1 DMA 结构框图 .....	88
图 8-1 高级定时器结构框图 .....	103
图 8-2 计数单元的结构 .....	104
图 8-3 当预分频器的参数从 1 变到 2 时, 计数器的时序图 .....	106
图 8-4 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	106
图 8-5 外部时钟模式 1 .....	107
图 8-6 外部时钟模式 2 .....	107
图 8-7 通道 1 输入捕获 TI1 信号 .....	108
图 8-8 PWM 输入模式时序 .....	110
图 8-9 输出比较模式, 翻转 OC1 .....	111
图 8-10 边沿对齐的 PWM 波形 (ARR=8) .....	113
图 8-11 中央对齐的 PWM 波形 (ARR=8) .....	114
图 8-12 带死区插入的互补输出 .....	115
图 8-13 死区波形延迟大于负脉冲 .....	116
图 8-14 死区波形延迟大于正脉冲 .....	116
图 8-15 响应刹车的输出 .....	118
图 8-16 刹车控制 .....	119
图 8-17 产生六步 PWM, 使用 COM 的例子(OSSR=1) .....	120
图 8-18 编码器模式下计数器操作实例 .....	121
图 8-19 霍尔传感器接口的实例 .....	123
图 8-20 单脉冲模式的例子 .....	124
图 8-21 复位模式下的控制电路 .....	125
图 8-22 门控模式下的控制电路 .....	126
图 8-23 触发模式下的控制电路 .....	127
图 8-24 外部时钟模式 2+触发模式控制电路 .....	127
图 9-1 通用定时器 TIM2 结构框图 .....	163
图 10-1 通用定时器 TIM3/TIM4 框图 .....	191
图 11-1 基本定时器 TIM6/TIM7 结构框图 .....	220
图 12-1 通用定时器 TIM14 结构框图 .....	230
图 13-1 通用定时器 TIM15 结构框图 .....	244
图 13-2 通用定时器 TIM16/TIM17 结构框图 .....	244
图 14-1 WDT 功能框图 .....	279
图 15-1 IWDI 框图 .....	284
图 16-1 RTC 内部结构框图 .....	291

图 18-1 IO 结构框图.....	318
图 19-1 SPI 结构框图 .....	324
图 19-2 SPI 一线模式时序图 (MSB) .....	325
图 19-3 SPI 一线模式时序图 (LSB) .....	325
图 19-4 SPI 二线模式时序图 (MSB) .....	326
图 19-5 SPI 二线模式时序图 (LSB) .....	327
图 19-6 SPI 四线模式时序图 (MSB) .....	327
图 19-7 SPI 四线模式时序图 (LSB) .....	328
图 20-1 UART 结构框图 .....	346
图 20-2 两个 UART 之间的硬件流控连接示意图 .....	347
图 20-3 硬件流控时序图.....	348
图 20-4 LIN 帧格式 .....	348
图 20-5 LIN 间隔场格式 .....	349
图 20-6 LIN 字符格式.....	349
图 20-7 IrDA SIR 数据调制解调.....	350
图 20-8 UART 单线半双工模式框图 .....	350
图 21-1 I2C 结构框图 .....	366
图 21-2 I2C 总线传输 .....	367
图 21-3 7 位从发送器的传送图 .....	367
图 21-4 7 位从接收器的传送图 .....	368
图 21-5 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) .....	368
图 21-6 7 位主发送器的传送图 .....	369
图 21-7 7 位主接收器的传送图 .....	369
图 21-8 TXE 功能说明图 .....	370
图 21-9 主机发送流程图 (虚线步骤可跳过, 下同) .....	379
图 21-10 主机接收流程图 .....	381
图 21-11 从机发送接收流程图 .....	383
图 22-1 I2S 结构框图 .....	386
图 22-2 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=0) .....	387
图 22-3 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=1) .....	387
图 22-4 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=0) .....	387
图 22-5 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=1) .....	387
图 22-6 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0) .....	388
图 22-7 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1) .....	388
图 22-8 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0) .....	388
图 22-9 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1) .....	388
图 22-10 MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0) .....	389
图 22-11 MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1) .....	389
图 22-12 MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0) .....	389
图 22-13 MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1) .....	389
图 22-14 MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0) .....	390
图 22-15 MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1) .....	390
图 22-16 MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0) .....	390
图 22-17 MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1) .....	390

图 22-18 LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	391
图 22-19 LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	391
图 22-20 LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	391
图 22-21 LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	391
图 22-22 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)	392
图 22-23 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)	392
图 22-24 PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)	392
图 22-25 PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)	392
图 22-26 PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)	393
图 22-27 PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)	393
图 22-28 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)	393
图 22-29 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)	393
图 22-30 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)	394
图 22-31 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)	394
图 22-32 PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)	394
图 22-33 PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)	394
图 22-34 PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)	394
图 22-35 PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)	395
图 22-36 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)	395
图 22-37 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)	395
图 22-38 I2S 时钟生成结构框图	395
图 23-1 LPURT 整体框图	404
图 23-2 LPURT BIT 分布	405
图 24-1 ADC 框图	414
图 24-2 ADC 转换时序图	417
图 24-3 AWD 阈值图	418
图 25-1 DAC 结构框图	441
图 25-2 DAC 通道模式的数据寄存器	442
图 25-3 双 DAC 通道模式的数据寄存器	443
图 25-4 LFSR 噪声功能框图	445
图 25-5 三角波噪声功能框图	445
图 25-6 采样保持模式时序图	447
图 26-1 OPAMP 结构框图	459
图 26-2 运放外置模式配置等效图	461
图 26-3 运放单位增益模式配置等效图	461
图 26-4 运放 PGA 同相模式配置等效图	462
图 26-5 运放 PGA 反相模式配置等效图	462
图 27-1 COMP 结构框图	470
图 27-2 比较器窗口模式级联	472
图 28-1 内部功能框图	480
图 28-2 正常模式示意图	481
图 28-3 监听模式示意图	481
图 28-4 自测试模式示意图	482
图 28-5 CAN 数据帧结构	482

图 28-6 TxBuff 数据结构 .....	483
图 28-7 RxBuff 数据结构 .....	484
图 28-8 CAN 接收示意图 .....	485
图 28-9 单过滤模式过滤器示意图 .....	486
图 28-10 双过滤模式过滤器示意图 .....	487
图 28-11 CAN 位时序图 .....	488
图 29-1 USB 结构框架图 .....	503
图 29-2 控制传输过程 .....	533
图 29-3 Endpoint In 传输 .....	534
图 29-4 Endpoint Out 传输 .....	535

## 表目录

表格 2-1 低功耗模式 .....	9
表格 3-1 系统复位源 .....	62
表格 3-2 系统时钟选择 .....	66
表格 7-1 DBSIZE/SBSIZE 对应的 burst size .....	90
表格 7-2 目标外设和源外设请求号 .....	91
表格 8-1 计数器方向和编码器信号的关系 .....	120
表格 8-2 TIM1 从模式互连-ITRx .....	137
表格 8-3 TIM1 从模式互连-ETRx .....	137
表格 8-4 互补输出通道 OCx 和 OCxN 的控制 .....	150
表格 9-1 TIM2 从模式互联: ITRx .....	171
表格 9-2 TIM2 从模式互联: ETRx .....	171
表格 10-1 TIM3 从模式互联: ITRx .....	200
表格 10-2 TIM3 从模式互联: ETRx .....	200
表格 10-3 TIM4 从模式互联: ITRx .....	200
表格 10-4 TIM4 从模式互联: ETRx .....	201
表格 10-5 标准 OCx 通道的输出控制位 .....	213
表格 12-1 标准 OCx 通道的输出控制位 .....	241
表格 13-1 TIM15 从模式互联: ITRx .....	253
表格 22-1 比特率计算公式 .....	396
表格 22-2 音频采样率计算公式 .....	396
表格 24-1 ADC 通道说明 .....	415
表格 24-2 AWD 控制位表 .....	418
表格 24-3 ADC 触发源选择表 .....	419
表格 24-4 ADC 中断控制表 .....	422
表格 26-1 运放工作模式控制和输出关系 .....	460
表格 26-2 工作模式和负端输入 .....	460
表格 26-3 负端输入管脚 .....	460
表格 31-1 库文件说明 .....	539
表格 31-2 HRNG 驱动函数说明 .....	539
表格 31-3 库文件说明 .....	540

表格 31-4 AES 驱动函数说明.....	540
表格 31-5 库文件说明.....	543
表格 31-6 驱动函数说明.....	543
表格 31-7 库文件说明.....	544
表格 31-8 驱动函数说明.....	544

# 1. 文档约定

## 1.1. 基本信息

ACM32F403 芯片的内核基于 ARMv8-M 架构，支持 Cortex-M33 和 Cortex-M4F 指令集，芯片最高系统频率为 180 MHz。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
read/clear write1(RC_W1):	只读，写 1 清 0
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
X:	视具体情况而定
寄存器默认值格式:	带 0x 前缀的为 16 进制，其它的为 2 进制

## 1.3. 术语

AHB: advanced high-performance bus.

APB: advanced peripheral bus.

NVR: Non-Volatile Region

FPU: Floating-point Unit

MPU: Memory Protect Unit

DSP: Digital Signal Process

eFlash: Embedded Flash

LVD: Low Voltage Detector

## 2. 存储器及系统架构

ACM32F403 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 180 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 180MHz，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

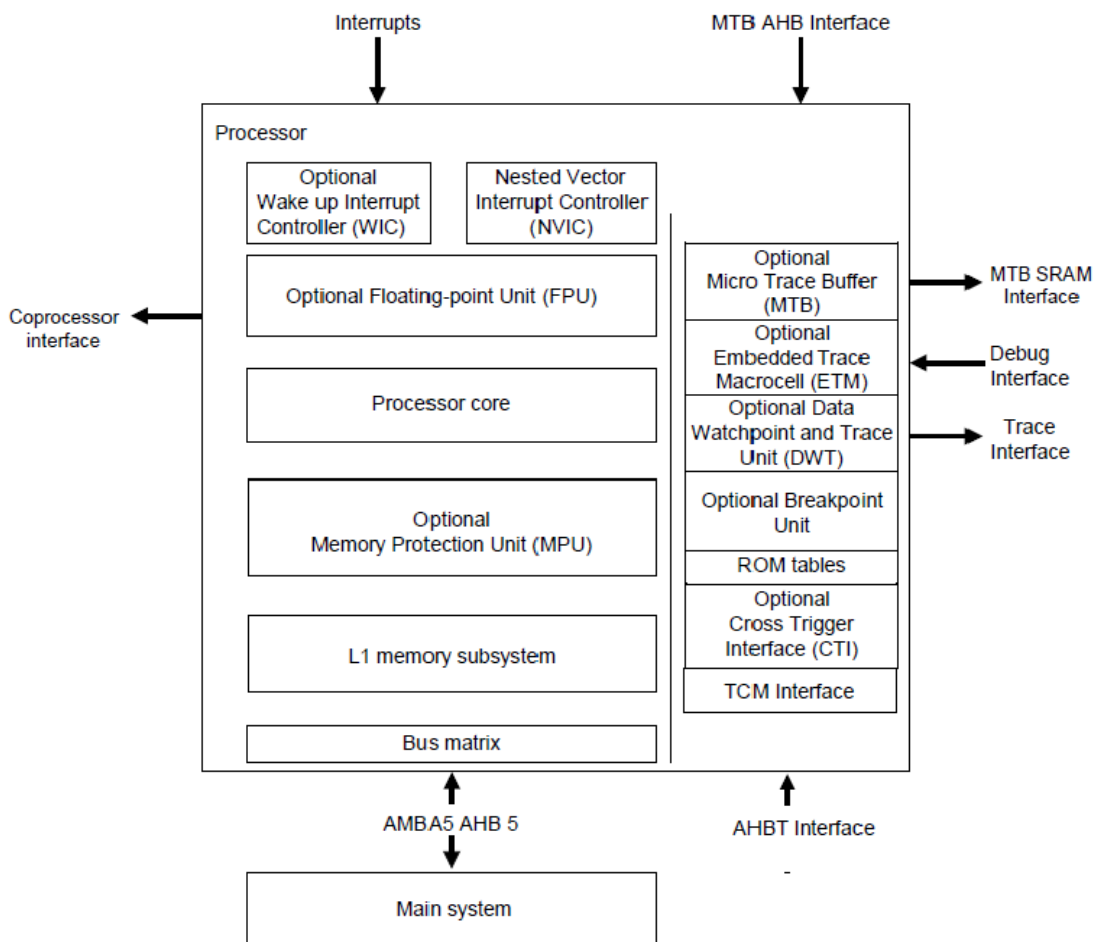
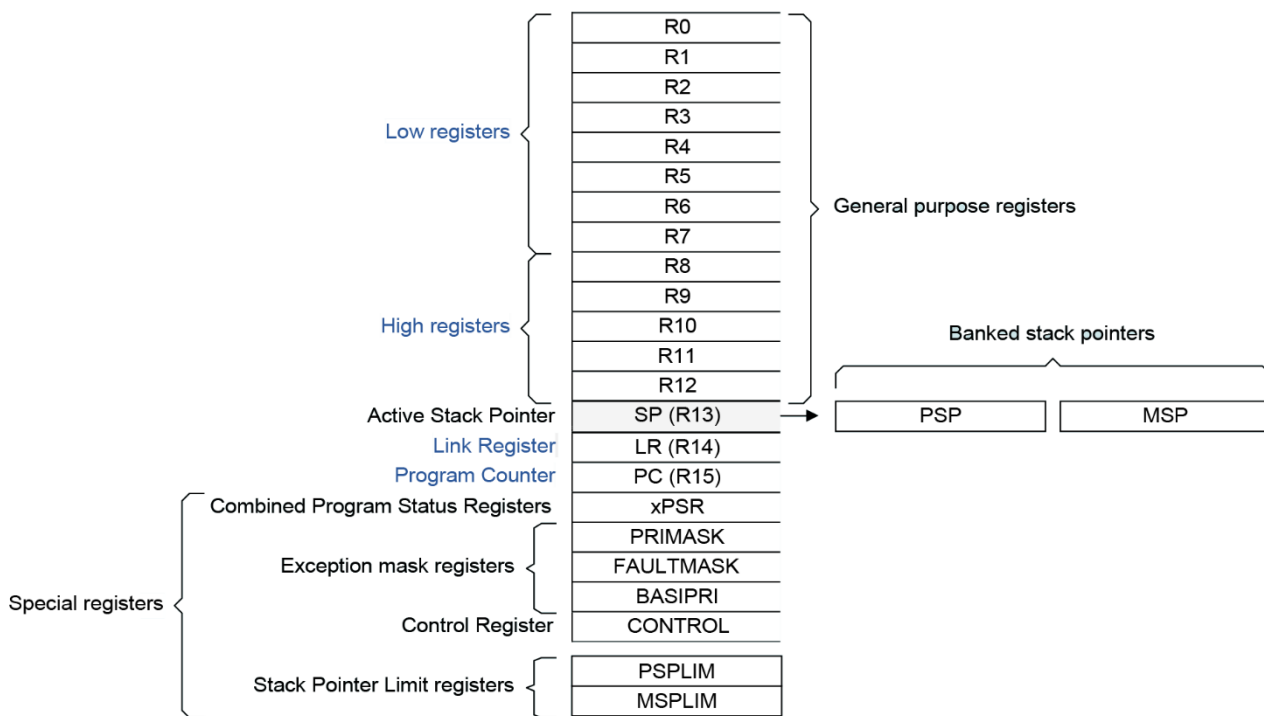


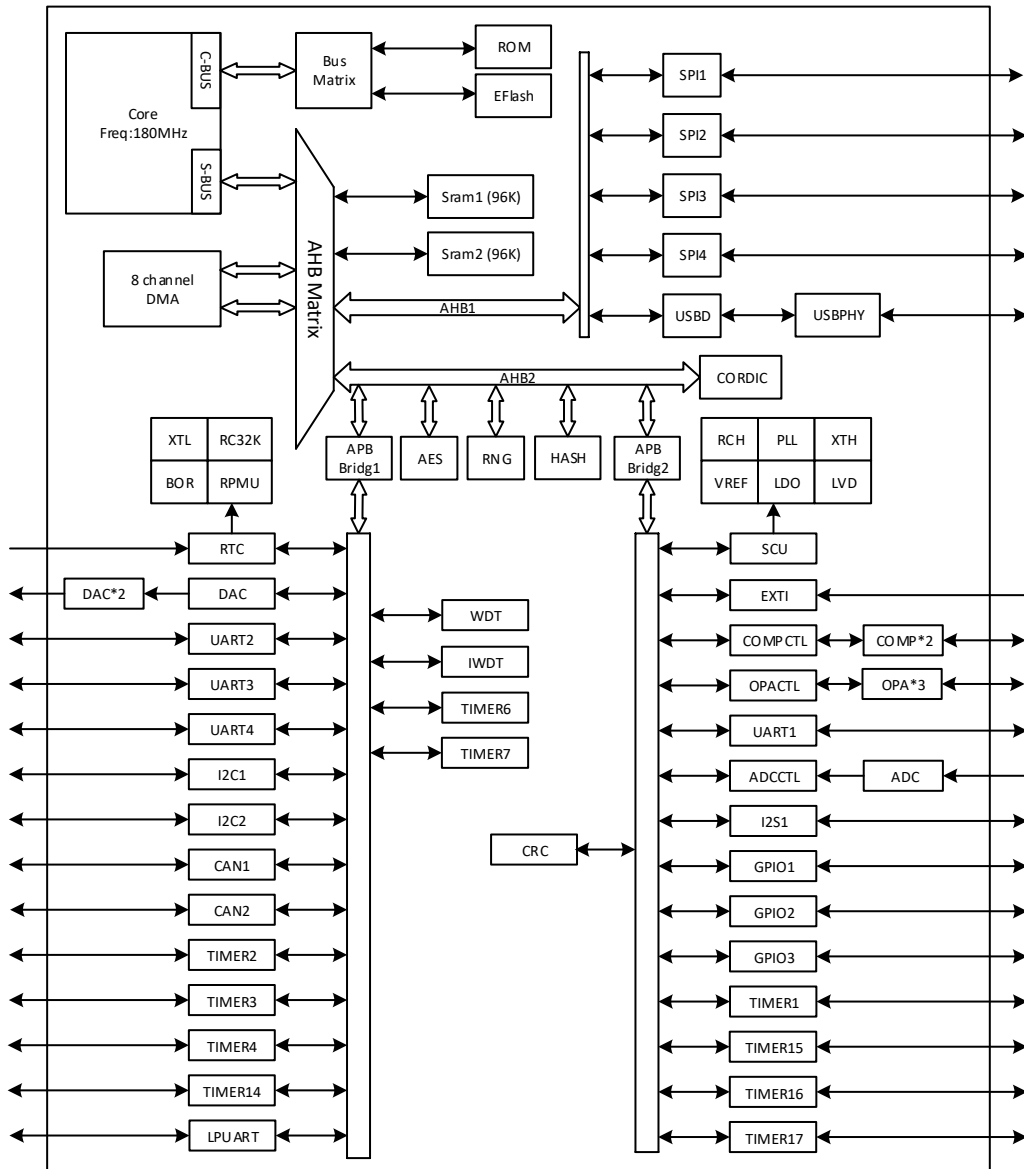
图 2-2 处理器寄存器组



## 2.2. 系统架构

系统架构图如图 2-3 所示。AHB1 和 AHB2 的时钟频率一致且等于系统频率，APB1 和 APB2 的时钟频率一致且可由系统频率分频而来。

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。

ACM32F403 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 512KB 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

Vendor-specific memory	511MB	0xFFFFFFFF
Private peripheralbus	1MB	0xE0100000 0xE00FFFFFF
External device	1.0GB	0xE0000000 0xDFFFFFFF
External RAM	1.0GB	0xA0000000 0x9FFFFFFF
Peripheral	0.5GB	0x60000000 0x5FFFFFFF
SRAM	0.5GB	0x40000000 0x3FFFFFFF
Code	0.5GB	0x20000000 0x1FFFFFFF
		0x00000000

表 2-1 ACM32F403 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——0x12000_2fff
eFlash	0x1000_0000——0x1007_ffff	0x0000_0000——0x0007_ffff
eFlash Ctrl	0x1010_0000——0x1010_ffff	0x0010_0000——0x0010_ffff
eFlash NVR	0x1008_0400——0x1008_07ff	0x0008_0400——0x0008_07ff
<b>SRAM</b>		
SRAM	0x2000_0000——0x2002_ffff	
<b>APB1 Slave</b>		
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	
TIM14	0x4000_2000——0x4000_23ff	
RPMU	0x4000_2400——0x4000_27ff	
RTC	0x4000_2800——0x4000_2Bff	
WDT	0x4000_2C00——0x4000_2fff	
IWDT	0x4000_3000——0x4000_33ff	
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	
DAC	0x4000_7400——0x4000_77ff	

LPUART	0x4000_8000——0x4000_83ff
<b>APB2 Slave</b>	
COMP	0x4001_0200——0x4001_02ff
OPA	0x4001_0300——0x4001_03ff
EXTI	0x4001_0400——0x4001_07ff
SCU	0x4001_0800——0x4001_0Bff
CRC	0x4001_0C00——0x4001_0fff
ADC	0x4001_2400——0x4001_27ff
TIM1	0x4001_2c00——0x4001_2fff
I2S1	0x4001_3000——0x4001_33ff
UART1	0x4001_3800——0x4001_3bff
TIM15	0x4001_4000——0x4001_43ff
TIM16	0x4001_4400——0x4001_47ff
TIM17	0x4001_4800——0x4001_4bff
GPIO1(31-0)	0x4001_F000——0x4001_F3ff
GPIO2(63-32)	0x4001_f400——0x4001_f7ff
GPIO3(95-64)	0x4001_f800——0x4001_fbff
<b>AHB1 Slave</b>	
SPI1	0x4002_0000——0x4002_03ff
SPI2	0x4002_0400——0x4002_07ff
SPI3	0x4002_0800——0x4002_0bff
SPI4	0x4002_0C00——0x4002_0fff
DMA Ctrl	0x4002_1000——0x4002_1fff
AES	0x4003_0000——0x4003_03ff
FAU	0x4003_0400——0x4003_07ff
HRNG	0x4003_0800——0x4003_0bff
HASH	0x4003_0c00——0x4003_0fff
USB2.0	0x4004_0000——0x4004_ffff
<b>QSPI memory</b>	

SPI3	0x9000_0000——0x9fff_ffff
------	--------------------------

## 2.4. 片上 SRAM

芯片集成了多达 192KB 的 SRAM，支持字节、半字（16 位）以及字（32 位）访问。处理器可以以 180MHz 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

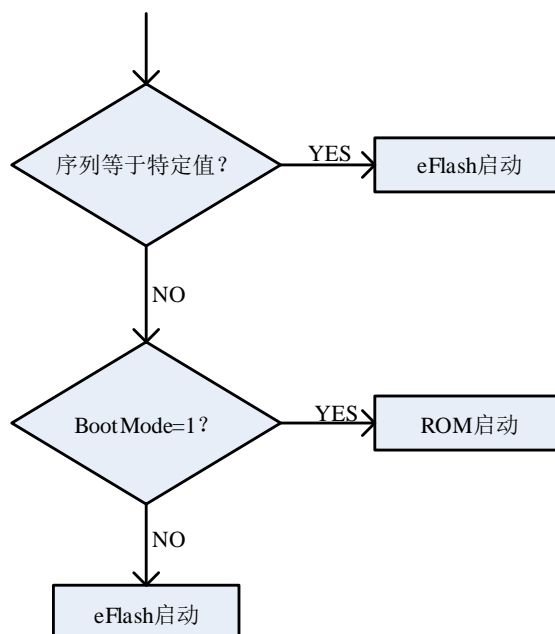
芯片集成了多达 512KB 的 Flash 以及 2 页的 NVR，支持字节、半字（16 位）以及字（32 位）访问。NVR 区的配置字可用于芯片配置，重新上电后生效。片上 Flash 的具体信息请参考 EFC 章节。

## 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 起始逻辑地址。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

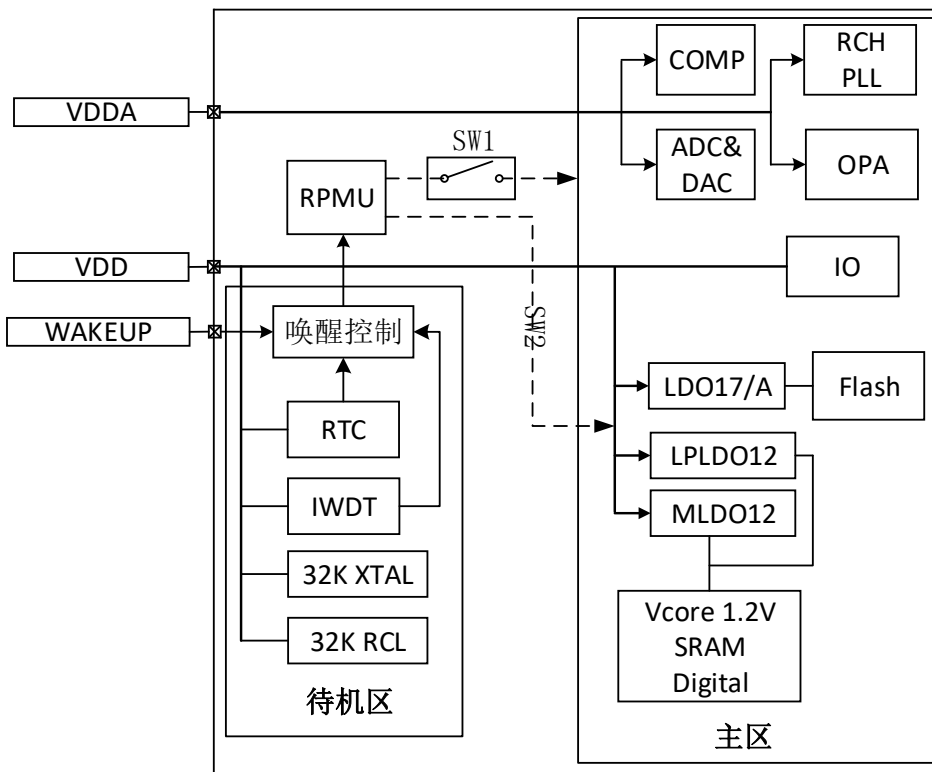
每颗芯片都包含唯一的 128 位（16 字节）的序列号。

基地址：0x0008\_0208

## 2.8. 电源控制

芯片的供电电源为 VCC，支持 1.70~3.60V 的电压范围。如下图所示，整个电源分为主区和待机区域两个部分。主区为芯片正常运行的主供电区，通过 VDDA/VDD 分别提供模拟电源和数字电源。待机区是芯片进入低功耗待机（STANDBY 模式）的供电区域。RPMU 模块见 RPMU 章节的详细介绍。

图 2-6 芯片电源域



## 2.9. 低功耗模式

芯片除正常工作模式外，为了降低芯片的功耗，芯片提供三种低功耗模式：睡眠（SLEEP）

模式、停止（STOP）模式和待机（STANDBY）模式。

- 睡眠模式（SLEEP）：内核停止工作，但所有外设的时钟仍然工作
- 停止模式（STOP）：所有时钟停止，IWDT、RTC 和 LPUART 可以工作在低速时钟
- 待机模式（STANDBY）：主区断电，待机区工作，IWDT 和 RTC 能继续工作

详细的描述如下表：

表格 2-1 低功耗模式

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.执行 WFI/WFE 指令	1.CPU 检测到中断或事件发生； 2.如果是 WFI 唤醒，则进入中断服务程序清中断并返回； 3.继续执行后续指令
STOP	大部分数字逻辑关闭时钟 (LPUART/RTC/IWDT 数字时钟保留，因为可工作在 RC32K 或 XTAL 时钟下，可作为唤醒源)；Disable 所有 analog 模块（LDO 除外），eFlash 可选择是否进入 SLEEP 模式。	1.关闭全局中断 2.确保系统时钟切换到 RCH；配置唤醒源，将 STOPCFG 中的 PDDS 写为 0，关闭不需要的模拟模块（如 PLL，比较器，运放，RC32K, LVD 等） 4.如需更低功耗，可将系统寄存器 STOPCFG 中的 RC64MPDEN 位和 LPSTOP 位置 1，在进入 STOP 模式后，LDO12 会自动进入低功耗模式，RC64M 会自动关闭。 5.调用 SRAM 中的 STOP 接口函数进入 STOP 模式。 注：如果配置的唤醒源在调用 SRAM 中 STOP 接口函数之前就是有效的，则不会进入 STOP 模式。	1.唤醒信号（EXTI 信号，见 EXTI 唤醒管理）到来。 2.MCU 唤醒，运行 SRAM 中 STOP 接口函数的后续程序，为回到正常模式提供运行环境。 3.程序从 SRAM 中返回，回到了正常模式，使能全局中断，执行后续指令。
STANDBY	关闭主区电源，保留 RTC 待机区工作	1.配置 RPMU 中 CR 寄存器来使能唤醒管脚的唤醒功能，或者使能 RTC 模块以及 RTC 中断。	WAKEUP 引脚的有效边沿/RTC 中断/RSTN 管脚复位/IWDT 复位

		2. 将 STOPCFG 中的 PDDS 写为 1。 3, 通过 WFI 或 WFE 进入 STANDBY 模式。	
--	--	--	--

## 2.10. 系统寄存器

系统控制器是 APB2 从设备。寄存器映射如下表所示：

SCU 寄存器基地址：0x4001\_0800

地址	名称	描述
0x0000	RCR	复位控制寄存器
0x0004	RSR	复位源状态寄存器
0x0008	IPRSTR2	模块复位寄存器 2
0x000C	IPRSTR1	模块复位寄存器 1
0x0010	CCR1	时钟控制寄存器 1
0x0014	CCR2	时钟控制寄存器 2
0x0018	CIR	时钟中断寄存器
0x001C	IPCKENR	模块使能寄存器
0x0020	IPCKENR2	模块使能寄存器 2
0x0024	RCHCR	RCH 控制寄存器
0x0028	XTHCR	XTH 控制寄存器
0x002C	PLLCR	PLL 控制寄存器
0x0030	LDOCR	模拟 LDO 控制寄存器
0x0034	REV	备用
0x0038	WMR	工作模式寄存器
0x003C	CLKOCR	时钟输出控制寄存器
0x0040	VER	版本寄存器
0x0044	SYSCFG1	系统配置寄存器 1
0x0048	LVDCFG	LVD 配置寄存器

0x004C	STOPCFG	STOP 模式配置寄存器
0x0050	RSV	备用
0x0054	PHYCR	USBPHY 模块控制寄存器
0x0058	MEMCFG	Memory 配置寄存器
0x005C	Dummy	Dummy 寄存器，默认为 0，可读写
0x0060	PASEL1	PA 管脚复用寄存器 1
0x0064	PASEL2	PA 管脚复用寄存器 2
0x0068	PBSEL1	PB 管脚复用寄存器 1
0x006C	PBSEL2	PB 管脚复用寄存器 2
0x0070	PA/BPUR	PA/PB 管脚上拉寄存器
0x0074	PA/BPDR	PA/PB 管脚下拉寄存器
0x0078	PASTR	PA 管脚输出驱动选择寄存器
0x007C	PBSTR	PB 管脚输出驱动选择寄存器
0x0080	PA/BSMTR	PA/PB 管脚 SMT 使能寄存器
0x0084	PA/BODR	PA/PB 管脚 OD 使能寄存器
0x0088	PA/BADS	PA/PB 管脚数模选择寄存器
0x008C	RSV	备用
0x0090	PCSEL1	PC 管脚复用寄存器 1
0x0094	PCSEL2	PC 管脚复用寄存器 2
0x0098	PDSEL1	PPD 管脚复用寄存器 1
0x009C	PDSEL2	PPD 管脚复用寄存器 2
0x00A0	PC/DPUR	PC/PD 管脚上拉寄存器
0x00A4	PC/DPDR	PC/PD 管脚下拉寄存器
0x00A8	PCSTR	PC 管脚驱动选择寄存器
0x00AC	PDSTR	PD 管脚驱动选择寄存器
0x00B0	PC/DSMTR	PC/PD 管脚 SMT 使能寄存器
0x00B4	PC/DODR	PC/PD 管脚 OD 使能寄存器
0x00B8	PC/DAD	PC/PD 管脚数模选择寄存器
0x00BC	RSV	备用

0x00C0	PESEL1	PE 管脚复用寄存器 1
0x00C4	PESEL2	PE 管脚复用寄存器 2
0x00C8	PFSEL1	PF 管脚复用寄存器 1
0x00CC	REV	备用
0x00D0	PE/FPUR	PE/PF 管脚上拉寄存器
0x00D4	PE/FPDR	PE/PF 管脚下拉寄存器
0x00D8	PESTR	PE 管脚驱动选择寄存器
0x00DC	PFSTR	PF 管脚驱动选择寄存器
0x00E0	PE/FSMTR	PE/PF 管脚 SMT 使能寄存器
0x00E4	PE/FODR	PE/PF 管脚 OD 使能寄存器
0x00E8	PE/FAD	PE/PF 管脚数模选择寄存器

### 2.10.1. 复位控制寄存器/RCR (偏移: 00h)

比特	名称	属性	默认值	功能描述
31	RSV	-	-	保留
30	SRST	WO	1	Soft Reset: 芯片软复位, 写 0 产生系统复位, 下一个时钟自动回 1
29	EFC_RST	WO	1	eFlash 复位: 芯片复位并发起 NVR 重新加载, 写 0 产生系统复位。复位层级和 NRST 复位相当, 详情见时钟与复位章节。
28:17	RSV	-	-	保留
16	REMAP_IMP	RW	0	REMAP 立刻使能, eFlash 起始地址被映射到 0 地址 0: 无效 1: 写 1 立刻使能 REMAP, 下次 REMAP 先写 0 再写 1
15:4	RSV	-	-	保留
3	LOCKUPRST_EN	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	IWDTRST_EN	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	WDTRST_EN	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	LVDRST_EN	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

### 2.10.2. 复位源状态寄存器/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 RSTFlag\_Clr 清除，或者上电复位清除。

比特	名称	属性	默认值	功能描述
31:17	RSV	-	-	保留
16	RSTFlag_Clr	WO	0	写 1 清除所有复位标志
15:9	RSV	-	-	保留
10	PWRRST_F	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	POR12RST_F	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	RSV	-	-	保留
7	SRST_F	RO	0	Soft Reset 标志: 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
6	EFCRST_F	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	RSTN_F	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	SYSREQRST_F	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	LOCKUPRST_F	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	IWDRST_F	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	WDTRST_F	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	LVDRST_F	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

### 2.10.3. 模块复位控制寄存器 2/IPRSTR2(偏移: 08h)

比特	名称	属性	默认值	功能描述
31:7	RSV	-	-	保留
6	SPI4RST	RW	1	SPI4 模块复位 0: 复位; 1: 不复位
5	GPIO3RST	RW	1	GPIO3 模块复位, 对应 PE、PF Port 所有的 GPIO 0: 复位; 1: 不复位
4	SPI3RST	RW	1	SPI3 模块复位 0: 复位; 1: 不复位
3	UART4RST	RW	1	UART4 模块复位 0: 复位; 1: 不复位
2	TIM4RST	RW	1	TIM4 模块复位 0: 复位; 1: 不复位
1	CAN2RST	RW	1	CAN2 模块复位 0: 复位; 1: 不复位
0	I2S1RST	RW	1	I2S1 模块复位 0: 复位; 1: 不复位

## 2.10.4. 模块复位控制寄存器 1/IPRSTR1(偏移: 0Ch)

比特	名称	属性	默认值	功能描述
31	CAN1	RW	1	CAN1 模块复位 0: 复位; 1: 不复位
30	TIM2RST	RW	1	TIM2 模块复位 0: 复位; 1: 不复位
29	USBCTLRST	RW	1	USB 控制器复位 0: 复位; 1: 不复位
28	UACRST	RW	1	UAC 模块复位 0: 复位; 1: 不复位
27	TIM17RST	RW	1	TIM17 模块复位 0: 复位; 1: 不复位
26	TIM16RST	RW	1	TIM16 模块复位 0: 复位; 1: 不复位
25	TIM15RST	RW	1	TIM15 模块复位 0: 复位; 1: 不复位
24	TIM14RST	RW	1	TIM14 模块复位 0: 复位; 1: 不复位
23	TIM7RST	RW	1	TIM7 模块复位 0: 复位; 1: 不复位
22	TIM6RST	RW	1	TIM6 模块复位 0: 复位; 1: 不复位
21	TIM3RST	RW	1	TIM3 模块复位 0: 复位; 1: 不复位
20	TIM1RST	RW	1	TIM1 模块复位 0: 复位; 1: 不复位
19	UART3RST	RW	1	UART3 模块复位 0: 复位; 1: 不复位
18	EXTIRST	RW	1	EXTI 模块复位 0: 复位; 1: 不复位
17	OPARST	RW	1	OPA 模块复位 0: 复位; 1: 不复位

16	COMPRST	RW	1	COMP 模块复位 0: 复位; 1: 不复位
15	RSV	-	-	保留
14	ADCRST	RW	1	ADC 模块复位 0: 复位; 1: 不复位
13	DACRST	RW	1	DAC 模块复位 0: 复位; 1: 不复位
12	DMACRST	RW	1	DMA 模块复位 0: 复位; 1: 不复位
11	CRCRST	RW	1	CRC 模块复位 0: 复位; 1: 不复位
10	RSV	-	-	保留
9	WDTRST	RW	1	WDT 模块复位 0: 复位; 1: 不复位
8	LPUARTRST	RW	1	LPUART 模块复位 0: 复位; 1: 不复位
7	I2C2RST	RW	1	I2C2 模块复位 0: 复位; 1: 不复位
6	I2C1RST	RW	1	I2C1 模块复位 0: 复位; 1: 不复位
5	SPI2RST	RW	1	SPI2 模块复位 0: 复位; 1: 不复位
4	SPI1RST	RW	1	SPI1 模块复位 0: 复位; 1: 不复位
3	UART2RST	RW	1	UART2 模块复位 0: 复位; 1: 不复位
2	UART1RST	RW	1	UART1 模块复位 0: 复位; 1: 不复位
1	GPIO2RST	RW	1	GPIO2 模块复位, 对应 PC、PD Port 所有的 GPIO 0: 复位; 1: 不复位
0	GPIO1RST	RW	1	GPIO1 模块复位, 对应 PA、PB Port 所有的 GPIO 0: 复位; 1: 不复位

## 2.10.5. 时钟控制寄存器 1/CCR1(偏移: 10h)

比特	名称	属性	默认值	功能描述
31:3	RSV	-	-	保留
2:0	SYS_CLK_SEL	RW	000	系统时钟 SYS_CLK 选择: 000: 来自 RCH 001: 来自 RC32K 010: 来自 XTH 011: 来自 XTL 1xx: 来自 PLLCLK

## 2.10.6. 时钟控制寄存器 2/CCR2 (偏移: 14h)

比特	名称	属性	默认值	功能描述
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
30:16	RSV	-	-	保留
15	FLTCLK_SEL	RW	0	FLTCLK 选择, LVD 和 COMP 的滤波时钟选择 0: PCLK 的 32 分频 1: RC32K
14:13	LPUCLK_SEL	RW	00	LPUART 时钟选择 00: RC32K 01: XTL 10: PCLK 分频, 由 LPUARTDIV 确定
12:11	LPUARTDIV	RW	11	LPUART 使用 PCLK 分频选择 00:4 分频 01:8 分频 10:16 分频 11:32 分频
10:8	PCLKDIV	RW	000	PCLK 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 SYS_CLK_DIV0 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

注意: 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。

### 2.10.7. 时钟中断寄存器/CIR (偏移: 18h)

比特	名称	属性	默认值	功能描述
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	XTHRDIYIC	WO	0	XTH 时钟稳定中断清除。

				软件写 1 清除中断 XTHRDIYF 0: 无影响 1: 清除 XTH 时钟稳定中断标志
18	RCHRDIYIC	WO	0	RCH 时钟稳定中断清除。 软件写 1 清除中断 RCHRDIYF 0: 无影响 1: 清除 RCH 时钟稳定中断标志
17	XTLRDIYIC	WO	0	XTL 时钟稳定中断清除。 软件写 1 清除中断 XTLRDIYF 0: 无影响 1: 清除 XTL 时钟稳定中断标志
16	RC32KRDIYIC	WO	0	RC32K 时钟稳定中断清除。 软件写 1 清除中断 RC32KRDIYF 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	XTHRDIYIE	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	RCHRDIYIE	RW	0	RCH 时钟稳定中断使能 0: 禁止 RCH 时钟稳定中断 1: 使能 RCH 时钟稳定中断
9	XTLRDIYIE	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	RC32KRDIYIE	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, PLL_LOCK(或 PLL_FREE_RUN) 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	XTHRDIYIF	RO	0	XTH 时钟稳定中断标志 当 XTH 时钟稳定, XTHRDIY 标志有效后硬件置位。软件设置 XTHRDIYIC 清除 0: 无 XTH 时钟稳定中断 1: 产生 XTH 时钟稳定中断
2	RCHRDIYIF	RO	1	RCH 时钟稳定中断标志 当 RCH 时钟稳定, RCHRDIY 标志有效后硬件置位。软件设置 RCHRDIYIC 清除 0: 无 RCH 时钟稳定中断 1: 产生 RCH 时钟稳定中断
1	XTLRDIYIF	RO	0	XTL 时钟稳定中断标志 当 XTL 时钟稳定, XTLRDIY (RTC 阈) 标志有效后硬件置位。软件设置 XTLRDIYIC 清除 0: 无 XTL 时钟稳定中断 1: 产生 XTL 时钟稳定中断
0	RC32KRDIYIF	RO	1	RC32K 时钟稳定中断标志 当 RC32K 时钟稳定, RC32KRDIY (RTC 阈) 标志有效后硬件置位。软件设置 RC32KRDIYIC 清除 0: 无 RC32K 时钟稳定中断 1: 产生 RC32K 时钟稳定中断

### 2.10.8. 模块时钟使能寄存器/IPCKENR (偏移: 1Ch)

比特	名称	属性	默认值	功能描述
31	CAN1 CLKEN	RW	0	CAN1 模块时钟使能 0: 不使能; 1: 使能
30	ROM CLKEN	RW	1	ROM 模块时钟使能 0: 不使能; 1: 使能
29	HRNGCLKEN	RW	0	HRNG 模块时钟使能 0: 不使能; 1: 使能
28	AES CLKEN	RW	0	AES 模块时钟使能 0: 不使能; 1: 使能

27	TIM17 CLKEN	RW	0	TIM17 模块时钟使能 0: 不使能; 1: 使能
26	TIM16 CLKEN	RW	0	TIM16 模块时钟使能 0: 不使能; 1: 使能
25	TIM15 CLKEN	RW	0	TIM15 模块时钟使能 0: 不使能; 1: 使能
24	TIM14 CLKEN	RW	0	TIM14 模块时钟使能 0: 不使能; 1: 使能
23	TIM7 CLKEN	RW	0	TIM7 模块时钟使能 0: 不使能; 1: 使能
22	TIM6 CLKEN	RW	1	TIM6 模块时钟使能 0: 不使能; 1: 使能
21	TIM3 CLKEN	RW	0	TIM3 模块时钟使能 0: 不使能; 1: 使能
20	TIM1 CLKEN	RW	0	TIM1 模块时钟使能 0: 不使能; 1: 使能
19	UART3 CLKEN	RW	1	UART3 模块时钟使能 0: 不使能; 1: 使能
18	TIM2 CLKEN	RW	1	TIM2 模块时钟使能 0: 不使能; 1: 使能
17	OPA CLKEN	RW	0	OPA 模块时钟使能 0: 不使能; 1: 使能
16	COMP CLKEN	RW	0	COMP 模块时钟使能 0: 不使能; 1: 使能
15	RSV	-	-	保留
14	ADC CLKEN	RW	0	ADC 模块时钟使能 0: 不使能; 1: 使能
13	RTC PCLKEN	RW	1	RTC 模块总线时钟 PCLK 使能 0: 不使能; 1: 使能
12	DMAC CLKEN	RW	0	DMA 模块时钟使能 0: 不使能; 1: 使能
11	CRC CLKEN	RW	0	CRC 模块时钟使能 0: 不使能; 1: 使能
10	DAC CLKEN	RW	0	DAC 模块时钟使能 0: 不使能; 1: 使能

9	USBCTL CLKEN	RW	0	USB 控制器时钟使能 0: 不使能; 1: 使能
8	LPUART CLKEN	RW	0	LPUART 模块时钟使能 0: 不使能; 1: 使能
7	I2C2 CLKEN	RW	0	I2C2 模块时钟使能 0: 不使能; 1: 使能
6	I2C1 CLKEN	RW	0	I2C1 模块时钟使能 0: 不使能; 1: 使能
5	SPI2 CLKEN	RW	0	SPI2 模块时钟使能 0: 不使能; 1: 使能
4	SPI1 CLKEN	RW	0	SPI1 模块时钟使能 0: 不使能; 1: 使能
3	UART2 CLKEN	RW	0	UART2 模块时钟使能 0: 不使能; 1: 使能
2	UART1 CLKEN	RW	1	UART1 模块时钟使能 0: 不使能; 1: 使能
1	CAN2 CLKEN	RW	0	CAN2 模块时钟使能 0: 不使能; 1: 使能
0	I2S1 CLKEN	RW	0	I2S1 模块时钟使能 0: 不使能; 1: 使能

### 2.10.9. 模块时钟使能寄存器 2/IPCKENR2 (偏移: 20h)

比特	名称	属性	默认值	功能描述
31:14	RSV	-	-	保留
13	SPI4 CLKEN	RW	0	SPI4 模块时钟使能 0: 不使能; 1: 使能
12	GPIO3 CLKEN	RW	0	GPIO3 模块时钟使能, 对应 PE 和 PF Port 所有的 GPIO 0: 不使能; 1: 使能
11	SPI3 CLKEN	RW	0	SPI3 模块时钟使能 0: 不使能; 1: 使能
10	UART4 CLKEN	RW	0	UART4 模块时钟使能 0: 不使能; 1: 使能

9	TIM4 CLKEN	RW	0	TIM4 模块时钟使能 0: 不使能; 1: 使能
8	FAU CLKEN	RW	0	FAU 模块时钟使能 0: 不使能; 1: 使能
7	RSV	-	-	保留
6	EFC CLKEN	RW	1	Eflash 模块在 SLEEP 模式下时钟使能 0: SLEEP 模式下不使能 1: SLEEP 模式下使能
5	SRAM CLKEN	RW	1	SRAM 在 SLEEP 模式下时钟使能 0: SLEEP 模式下不使能 1: SLEEP 模式下使能
4	EXTI CLKEN	RW	1	EXTI 模块时钟使能 0: 不使能; 1: 使能
3	RSV	-	-	保留
2	WDT CLKEN	RW	0	WDT 模块时钟使能 0: 不使能; 1: 使能
1	GPIO2 CLKEN	RW	0	GPIO2 模块时钟使能, 对应 PC 和 PD Port 所有的 GPIO 0: 不使能; 1: 使能
0	GPIO1 CLKEN	RW	1	GPIO1 模块时钟使能, 对应 PA 和 PB Port 所有的 GPIO 0: 不使能; 1: 使能

### 2.10.10. RCH 模块控制寄存器/RCHCR (偏移: 24h)

该寄存器用于模拟 RCH

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9	RCHRDY	RO	1	内部 RCH 时钟稳定标志 0: RCH 时钟未稳定 1: RCH 时钟稳定, 时钟有效
8	RCH_DIV	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 RCH, 此位

				置 1 后，系统时钟会变为原来的 1/16。
7:1	RCH_TRIM[6:0]	RW	0x0	RCH 时钟 TRIM 值 TRIM 值越高频率越高
0	RCH_EN	RW	1	RCH 模块使能

### 2.10.11. XTHCR 模块控制寄存器/XTHCR (偏移: 28h)

该寄存器用于模拟 XTH

比特	名称	属性	默认值	功能描述
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	XTH_RDYTIME	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	XTH_BYP	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	XTH_EN	RW	0	XTH 模块使能

### 2.10.12. PLL 模块控制寄存器/PLLCR (偏移: 2Ch)

该寄存器用于模拟 PLL

比特	名称	属性	默认值	功能描述
31	PLL_LOCK_SEL	RW	0	PLL LOCK 信号选择，用于产生 PLL LOCK 中断 0: 选择 PLL_LOCK (模拟输出) 1: 选择 PLL_FREE_RUN (数字计数)
30	PLL_FREE_RUN	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

				<p>自动清除该位；当 PLL 退出 SLEEP 模式且配置生效后，等待 PLL_RUN_DLY 设置时间后，硬件自动置位。</p> <p>1: PLL 正常工作。</p> <p>0: PLL 正常不正常。</p>
29	PLL_LOCK	RO	0	<p>PLL 锁定状态位。检测模拟 PLL 锁定信号的上升沿。</p> <p>1: PLL 已经锁定。</p> <p>0: PLL 未锁定。</p>
28:23	PLL_RUN_DLY	RW	0x10=010000	<p>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：</p> $\text{PLLLOCKDLY} * 512 * (1/\text{PCLK})$ <p>PCLK 为 32M 时，1 代表等待约 16us。</p>
22	PLL_UPDATE_EN	RW	0	<p>PLL 配置更新控制位，下一个周期自动回 0。重新改写 PLL_M、PLL_N 和 PLL_F、PLL_SRC_SEL 后，需要该该比特置位新的配置才能生效。复位时间 60* (1/PCLK)</p> <p>1: PLL 配置更新。</p> <p>0: PLL 配置不更新。</p>
21	PLL_SLEEP	RW	1	<p>PLL 休眠控制位（PLL 处于复位状态）</p> <p>1: PLL 进入休眠。</p> <p>0: PLL 不进入休眠。</p> <p>注：先使能 PLL，再退出休眠。</p>
20	RSV	-	-	保留
19:16	PLL_M	RW	0001	<p>输出分频控制字段。PLL_M 含有输出分频控制字段。对应分频值为(PLL_M+1)</p>
14:12	PLL_N	RW	001	<p>降频因子分频器字段。PLL_N 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 (PLL_N+1)</p>
11:9	RSV	-	-	保留
8:3	PLL_F	RW	0x01	<p>增频因子分频器字段。PLL_F 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(PLL_F+12)</p> <p>PLL 输出公式为: <math>F_{pll} = (\text{Fin} * (\text{PLL\_F} + 12) / (\text{PLL\_N} + 1)) / (\text{PLL\_M} + 1)</math></p>

				PLL 输入时钟 Fin 频率范围 3~48MHz Fin/(PLL_N+1)频率范围 3~6Mhz Fvco= Fin* (PLL_F+12)/(PLL_N+1), 频率范围在 96~216MHz。
2:1	PLL_SRC_SEL	RW	00	PLL 时钟源选择 00: 选择片内 RCH (需设置 16 分频模式) 1x: 选择片外 XTH
0	PLL_EN	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

### 2.10.13. 模拟 LDO 控制寄存器 LDOCR (偏移: 30h)

比特	名称	属性	默认值	功能描述
31:29	RSV	-	-	保留
28:26	LPLDO12_TRIM	RW	100	正常模式下, LPLDO12 选择 100: 1.1V 其他值: 保留
25:15	RSV	-	-	保留
14	LDO17A_EN	RW	1	LDO17A 使能信号, 默认为高
13:10	RSV	-	-	保留
9	LDO17_EN	RW	1	LDO17 使能信号, 默认为高
8	LDOCTL_SEL	RW	0	LDO 控制选择 0: MLDO12_LV、MLDO12_LOWP、LPSTP、LPLDO_EN 受 STOPCFG 相关位控制 1: 测试模式
7:4	LDO12_TRIM	RW	1000	LDO12 电压 TRIM
3	RSV	-	-	保留
2:0	VREF_TRIM	RW	100	基准电压 TRIM

### 2.10.14. 工作模式寄存器/WMR(偏移: 38h)

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

### 2.10.15. 时钟输出控制寄存器/CLKOCR(偏移: 3Ch)

比特	名称	属性	默认值	功能描述
31	BUZZER2_EN	RW	0	蜂鸣器 2 时钟输出使能, 对应 MCO2 输出 0: 不使能, 信号为 BUZZER2_POL 1: 使能
30	BUZZER2_POL	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	BUZZER2_DIV	RW	0x3F=111111	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	BUZZER_EN	RW	1	蜂鸣器时钟输出使能, 对应 MCO 输出 0: 不使能, 信号为 BUZZER_POL 1: 使能
22	BUZZER_POL	RW	0	蜂鸣器时钟极性选择 0: 原极性 (停止的时候为 0) 1: 反极性 (停止的时候为 1)

21:5	BUZZER_DIV	RW	0x4F	蜂鸣器时钟分频值：分频数为寄存器值+1 默认为 80 分频
4	MCO_SEL	RW	1	Main Clock Output 信号选择 0: CLK_TEST 信号(详见 CLKTEST_SEL) 1: BUZZER 信号 (详见 BUZZER_DIV)
3:0	CLKTEST_SEL	RW	0000	CLK_TEST 选择 0000: HCLK 0001: RCH 0010: RC32K 0011: XTH 0100: XTL 0101: PLLCLK 0110: PCLK 0111: SYS_CLK 1000: TIMCLK 1001: LPUARTCLK 1010: SysTick 1011: USBPHY_PLL48M 1100: RTC_PCLK 其他: 无时钟

### 2.10.16. 版本寄存器/VER (偏移: 40h)

比特	名称	属性	复位值	描述
31: 0	Version	RO	0xFFCF0030	芯片版本 0-7 位为芯片大版本, 0x30 表示 V1 版本 8-15 位为芯片小版本 0x00:a;版本 0x01:b 版本 0x02:c 版本 16-31 位为 0-15 位的取反值

## 2.10.17. 系统配置寄存器 1/SYSCFG1(偏移: 44h)

比特	名称	属性	默认值	功能描述
31:26	RSV	-	-	保留
25:24	TIM15_CH2_SEL	RW	00	TIM15_CH2 输入选择 00: 来自 IO 输入 01: 来自 MCO, 用于时钟校准 10: 来自 UART1_RX, 用于 UART 波特率校验 11: 来自 UART2_RX, 用于 UART 波特率校验
23: 22	TIM15_CH1_SEL	RW	00	TIM15_CH1 输入选择 00: 来自 IO 输入 01: 来自 MCO, 用于时钟校准 10: 来自 UART1_RX, 用于 UART 波特率校验 11: 来自 UART2_RX, 用于 UART 波特率校验
21	RSV	-	-	保留
20	TIM14_CH1_SEL	RW	0	TIM14_CH1 输入选择 0: 来自 IO 输入 1: 来自 MCO, 用于时钟校准
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
7:3	RSV	-	-	保留
2	LVD_LOCK	RW	0	LVD 检测错误锁定 0: LVD 检测错误输出从 TIM1/15/16/17 的 break 输

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

### 2.10.18. LVD 配置寄存器/LVDCFG (偏移: 48h)

比特	名称	属性	默认值	功能描述
31:16	RSV	-	-	保留
15	LVD_Value	RW	0	LVD 原始状态
14	LVD_Filter	RW	0	LVD Filter 后的状态
11: 9	FLT_TIME	RW	000	LVD 滤波时间配置, 滤波时间按 FiltClk 计算, FiltClk 由系统寄存器 CCR2 的 BIT15 进行控制。 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
7:6	RSV	-	-	保留
4:1	LVD_TRIM	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。 0000: 1.71 0001: 2.01 0010: 2.23 0011: 2.43 0100: 2.51 0101: 2.73 0110: 2.80 0111: 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，复位功能在 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。

### 2.10.19. STOP 模式配置寄存器/STOPCFG(偏移: 4Ch)

比特	名称	属性	默认值	功能描述
31:28	RSV	-	-	保留
27:16	WakeTime	RW	0x3C0	STOP 唤醒时间设置 从 STOP 模式唤醒所需的时钟周期数(以系统时钟计数)
15:13	LPLDO12_LV	RW	100	停止 (STOP) 模式下，低功耗模式(LPSTP)的 LPLDO12 调整选择 100: 调整为 1.1V (不变) 其他: 保留
12	RSV	-	-	保留
11	PDDS	RW	0	内核进 DeepSleep 时芯片低功耗模式的选择 0: 内核进 DeepSleep 时芯片进入 STOP 模式。LDO 工

				作状态受 LPSTOP 控制 1: 内核进 DeepSleep 时芯片进入 STANDBY 模式。 STANDBY 模式下主区断电
10	RSV	-	-	保留
9	LPSTOP	RW	0	停止 (STOP) 模式下, 是否进入 LPSTOP 模式。 LDO12 使用 LPLDO12, LDO17 关闭, MLDO12 关闭。 0: STOP 下不进入 LPSTP 模式 1: STOP 下自动进入 LPSTP 模式
8	RCH_DIV_EN	RW	0	停止 (STOP) 模式下, RCH 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
7	RSV	-	-	保留
6	MLDO12_LOWP	RW	0	停止 (STOP) 模式下, MLDO 是否进入低功耗模式(非 LPSTOP 模式有效)。 0: STOP 下 MLDO12 不降低自身功耗 1: STOP 下 MLDO12 降低自身功耗
5:4	MLDO12_LV	RW	00	停止 (STOP) 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
3	RC64MPDEN	RW	0	停止 (STOP) 模式下, RCH 即 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
2:1	RSV	-	-	保留
0	RTC_WE	RW	0	RTC 域写使能 0: RTC 域写禁止 1: RTC 域写使能

### 2.10.20. USBPHY 配置寄存器/PHYCFG (偏移: 54h)

该寄存器用于 USBPHY 的整体控制

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	PLL_LOCK	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

## 2.10.21. Memory 配置寄存器/MEMCFG (偏移: 58h)

比特	名称	属性	默认值	功能描述
31:13	REV	-	-	保留
12:1	SRAM_RET[11:0]	RW	0x0000	SRAM 进入 Retention 模式，高有效 Retention 模式下 SRAM 不能操作，但每块会降低约 3.5uA 的功耗（每一块 16KB） 以 16KB 为单位设置 SRAM 进入 Retention 模式，对应关系如下： SRAM_RET[x]: 16K*x~16K*(x+1)。 例如： SRAM_RET[0]设置 0~16KB SRAM_RET[5]设置 80KB~96KB
0	ROM_SLP	RW	0	ROM 进入 SLEEP 模式，高有效。 SLEEP 不能进行读操作，会降低 1uA 功耗

## 2.10.22. PA 管脚复用寄存器 1/PASEL1(偏移: 60h)

注：MCU 复位时 IO 配置也会复位，同 PB/PC/PD/PE/PF，具体功能见数据手册

比特	名称	属性	默认值	功能描述
31:28	PA7_SEL	RW	0000	PA7 复用功能选择 0000: GPIO7 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PA6_SEL	RW	0000	PA6 复用功能选择 0000: GPIO6 0001: 复用功能 1 0010: 复用功能 2 ...

				1001: 复用功能 9 其余: 保留
23:20	PA5_SEL	RW	0000	PA5 复用功能选择 0000: GPIO5 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
19:16	PA4_SEL	RW	0000	PA4 复用功能选择 0000: GPIO4 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PA3_SEL	RW	0000	PA3 复用功能选择 0000: GPIO3 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PA2_SEL	RW	0000	PA2 复用功能选择 0000: GPIO2 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
7:4	PA1_SEL	RW	0000	PA1 复用功能选择 0000: GPIO1 0001: 复用功能 1 0010: 复用功能 2 ...

				1001: 复用功能 9 其余: 保留
3:0	PA0_SEL	RW	0000	PA0 复用功能选择 0000: GPIO0 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.23. PA 管脚复用寄存器 2/PASEL2(偏移: 64h)

比特	名称	属性	默认值	功能描述
31:28	PA15_SEL	RW	0001	PA15 复用功能选择 (默认 JTDI) 0000: GPIO15 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PA14_SEL	RW	0001	PA14 复用功能选择 (默认 SWCLK) 0000: GPIO14 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
23:20	PA13_SEL	RW	0001	PA13 复用功能选择 (默认 SWDIO) 0000: GPIO13 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

19:16	PA12_SEL	RW	0000	PA12 复用功能选择 0000: GPIO12 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PA11_SEL	RW	0000	PA11 复用功能选择 0000: GPIO11 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PA10_SEL	RW	0000	PA10 复用功能选择 0000: GPIO10 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
7:4	PA9_SEL	RW	0000	PA9 复用功能选择 0000: GPIO9 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
3:0	PA8_SEL	RW	0001	PA8 复用功能选择 0000: GPIO8 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

## 2.10.24. PB 管脚复用寄存器 1/PBSEL1(偏移: 68h)

比特	名称	属性	默认值	功能描述
31:28	PB7_SEL	RW	0000	PB7 复用功能选择 0000: GPIO23 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PB6_SEL	RW	0000	PB6 复用功能选择 0000: GPIO22 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
23:20	PB5_SEL	RW	0000	PB5 复用功能选择 0000: GPIO21 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
19:16	PB4_SEL	RW	0001	PB4 复用功能选择 (默认 JTRST) 0000: GPIO20 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PB3_SEL	RW	0001	PB3 复用功能选择 (默认 JTDO) 0000: GPIO19

				0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PB2_SEL	RW	0000	PB2 复用功能选择 0000: GPIO18 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
7:4	PB1_SEL	RW	0000	PB1 复用功能选择 0000: GPIO17 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
3:0	PB0_SEL	RW	0000	PB0 复用功能选择 0000: GPIO16 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.25. PB 管脚控制寄存器 2/PBSEL2(偏移: 6Ch)

比特	名称	属性	默认值	功能描述
31:28	PB15_SEL	RW	0000	PB15 复用功能选择 0000: GPIO31 0001: 复用功能 1 0010: 复用功能 2

				... 1001: 复用功能 9 其余: 保留
27:24	PB14_SEL	RW	0000	PB14 复用功能选择 0000: GPIO30 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
23:20	PB13_SEL	RW	0000	PB13 复用功能选择 0000: GPIO29 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
19:16	PB12_SEL	RW	0000	PB12 复用功能选择 0000: GPIO28 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PB11_SEL	RW	0000	PB11 复用功能选择 0000: GPIO27 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PB10_SEL	RW	0000	PB10 复用功能选择 0000: GPIO26 0001: 复用功能 1 0010: 复用功能 2

				... 1001: 复用功能 9 其余: 保留
7:4	PB9_SEL	RW	0000	PB9 复用功能选择 0000: GPIO25 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
3:0	PB8_SEL	RW	0000	PB8 复用功能选择 0000: GPIO24 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.26. PA/B 管脚上拉寄存器/PA/BPUR (偏移: 70h)

该位控制管脚是否上拉，对应的管脚在复用时此控制位仍然有效。

比特	名称	属性	默认值	功能描述
31:16	PBPU	RW	0x0010	端口 PB 上拉使能配置寄存器 (对应 PB15-PB00) 1: 使能 0: 禁止
15:0	PAPU	RW	0xA000	端口 PA 上拉使能配置寄存器 (对应 PA15-PA00) 1: 使能 0: 禁止

### 2.10.27. PA/B 管脚下拉寄存器/PA/BPDR (偏移: 74h)

该位控制管脚是否下拉，对应的管脚在复用时此控制位仍然有效。

比特	名称	属性	默认值	功能描述
----	----	----	-----	------

31:16	PBPD	RW	0x0008	端口 PB 下拉使能配置寄存器 (对应 PB15-PB00) 1: 使能 0: 禁止
15:0	PAPD	RW	0x4000	端口 PA 下拉使能配置寄存器 (对应 PA15-PA00) 1: 使能 0: 禁止

### 2.10.28. PA 管脚驱动选择寄存器 PASTR (偏移: 78h)

比特	名称	属性	默认值	功能描述
31:30	PA15_STH	RW	00	PA15 输出驱动设置
29:28	PA14_STH	RW	00	PA14 输出驱动设置
27:26	PA13_STH	RW	00	PA13 输出驱动设置
25:24	PA12_STH	RW	00	PA12 输出驱动设置
23:22	PA11_STH	RW	00	PA11 输出驱动设置
21:20	PA10_STH	RW	00	PA10 输出驱动设置
19:18	PA9_STH	RW	00	PA9 输出驱动设置
17:16	PA8_STH	RW	00	PA8 输出驱动设置
15:14	PA7_STH	RW	00	PA7 输出驱动设置
13:12	PA6_STH	RW	00	PA6 输出驱动设置
11:10	PA5_STH	RW	00	PA5 输出驱动设置
9:8	PA4_STH	RW	00	PA4 输出驱动设置
7:6	PA3_STH	RW	00	PA3 输出驱动设置
5:4	PA2_STH	RW	00	PA2 输出驱动设置
3:2	PA1_STH	RW	00	PA1 输出驱动设置
1:0	PA0_STH	RW	00	PA0 输出驱动设置 IO 电压 3.3V 时, 输出驱动如下 00: 4.5mA 01: 9mA 10: 13.5mA 11: 18mA IO 电压 2.5V 时, 输出驱动如下 00: 3.5mA 01: 7mA

				10: 10.5mA 11: 14mA O 电压 1.8V 时，输出驱动如下 00: 2.1mA 01: 4.2mA 10: 6.3mA 11: 8.4mA
--	--	--	--	--

### 2.10.29. PB 管脚驱动选择寄存器 PBSTR (偏移: 7ch)

比特	名称	属性	默认值	功能描述
31:30	PB15_STH	RW	00	PB15 输出驱动设置
29:28	PB14_STH	RW	00	PB14 输出驱动设置
27:26	PB13_STH	RW	00	PB13 输出驱动设置
25:24	PB12_STH	RW	00	PB12 输出驱动设置
23:22	PB11_STH	RW	00	PB11 输出驱动设置
21:20	PB10_STH	RW	00	PB10 输出驱动设置
19:18	PB9_STH	RW	00	PB9 输出驱动设置
17:16	PB8_STH	RW	00	PB8 输出驱动设置
15:14	PB7_STH	RW	00	PB7 输出驱动设置
13:12	PB6_STH	RW	00	PB6 输出驱动设置
11:10	PB5_STH	RW	00	PB5 输出驱动设置
9:8	PB4_STH	RW	00	PB4 输出驱动设置
7:6	PB3_STH	RW	00	PB3 输出驱动设置
5:4	PB2_STH	RW	00	PB2 输出驱动设置
3:2	PB1_STH	RW	00	PB1 输出驱动设置
1:0	PB0_STH	RW	00	PB0 输出驱动设置，同 PA0

### 2.10.30. PA/B 管脚施密特使能寄存器 PA/BSMTR (偏移: 80h)

比特	名称	属性	默认值	功能描述
31:16	PB_SMTEN	RW	0xFFFF	端口 PB 施密特使能配置寄存器 (对应 PB15-PB00) 1: 使能

				0: 禁止
15:0	PA_SMTEN	RW	0xFFFF	端口 PA 施密特使能配置寄存器 (对应 PA15-PA00) 1: 使能 0: 禁止

### 2.10.31. PA/B 管脚 OD 使能寄存器 PA/BODR (偏移: 84h)

比特	名称	属性	默认值	功能描述
31:16	PB_ODEN	RW	0x0000	端口 PB open-drain 使能配置寄存器 (对应 PB15-PB00) 1: 使能 0: 禁止
15:0	PA_ODEN	RW	0x0000	端口 PA open-drain 使能配置寄存器 (对应 PA15-PA00) 1: 使能 0: 禁止

### 2.10.32. PA/B 管脚数模选择寄存器 PA/BADS (偏移: 88h)

比特	名称	属性	默认值	功能描述
31:16	PB_ADEN	RW	0xFFE7	端口 PB 数模选择寄存器 (对应 PB15-PB00) 1: 配置为模拟端口 0: 配置为数字端口
15:0	PA_ADEN	RW	0x1EFF	端口 PA 数模选择寄存器 (对应 PA15-PA00) 1: 配置为模拟端口 0: 配置为数字端口

### 2.10.33. PC 管脚复用寄存器 1/PCSEL1(偏移: 90h)

比特	名称	属性	默认值	功能描述
31:28	PC7_SEL	RW	0000	PC7 复用功能选择 0000: GPIO39 0001: 复用功能 1

				0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PC6_SEL	RW	0000	PA6 复用功能选择 0000: GPIO38 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
23:20	PC5_SEL	RW	0000	PC5 复用功能选择 0000: GPIO37 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
19:16	PC4_SEL	RW	0000	PC4 复用功能选择 0000: GPIO36 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PC3_SEL	RW	0000	PC3 复用功能选择 0000: GPIO35 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PC2_SEL	RW	0000	PC2 复用功能选择 0000: GPIO34 0001: 复用功能 1

				0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
7:4	PC1_SEL	RW	0000	PC1 复用功能选择 0000: GPIO33 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
3:0	PC0_SEL	RW	0000	PC0 复用功能选择 0000: GPIO32 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.34. PC 管脚复用寄存器 2/PCSEL2(偏移: 94h)

比特	名称	属性	默认值	功能描述
31:28	PC15_SEL	RW	0000	PC15 复用功能选择(最终复用由 RTC 域决定) 0000: GPIO47 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PC14_SEL	RW	0000	PC14 复用功能选择(最终复用由 RTC 域决定) 0000: GPIO46 0001: 复用功能 1 0010: 复用功能 2 ...

				1001: 复用功能 9 其余: 保留
23:20	PC13_SEL	RW	0000	PC13 复用功能选择(最终复用由 RTC 域决定) 0000: GPIO45 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
19:16	PC12_SEL	RW	0000	PC12 复用功能选择 0000: GPIO44 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PC11_SEL	RW	0000	PC11 复用功能选择 0000: GPIO43 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PC10_SEL	RW	0000	PC10 复用功能选择 0000: GPIO42 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
7:4	PC9_SEL	RW	0000	PC9 复用功能选择 0000: GPIO41 0001: 复用功能 1 0010: 复用功能 2 ...

				1001: 复用功能 9 其余: 保留
3:0	PC8_SEL	RW	0000	PC8 复用功能选择 0000: GPIO40 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.35. PD 管脚复用寄存器 1/PDSEL1(偏移: 98h)

比特	名称	属性	默认值	功能描述
31:28	PD7_SEL	RW	0000	PD7 复用功能选择 0000: GPIO55 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PD6_SEL	RW	0000	PD6 复用功能选择 0000: GPIO54 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
23:20	PD5_SEL	RW	0001	PD5 复用功能选择 (默认 REMAP) 0000: GPIO53 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

19:16	PD4_SEL	RW	0001	<p>PD4 复用功能选择（默认 RSTO）</p> <p>0000: GPIO52</p> <p>0001: 复用功能 1</p> <p>0010: 复用功能 2</p> <p>...</p> <p>1001: 复用功能 9</p> <p>其余: 保留</p>
15:12	PD3_SEL	RW	0000	<p>PD3 复用功能选择</p> <p>0000: GPIO51</p> <p>0001: 复用功能 1</p> <p>0010: 复用功能 2</p> <p>...</p> <p>1001: 复用功能 9</p> <p>其余: 保留</p>
11:8	PD2_SEL	RW	0000	<p>PD2 复用功能选择</p> <p>0000: GPIO50</p> <p>0001: 复用功能 1</p> <p>0010: 复用功能 2</p> <p>...</p> <p>1001: 复用功能 9</p> <p>其余: 保留</p>
7:4	PD1_SEL	RW	0000	<p>PD1 复用功能选择</p> <p>0000: GPIO49</p> <p>0001: 复用功能 1</p> <p>0010: 复用功能 2</p> <p>...</p> <p>1001: 复用功能 9</p> <p>其余: 保留</p>
3:0	PD0_SEL	RW	0000	<p>PD0 复用功能选择</p> <p>0000: GPIO48</p> <p>0001: 复用功能 1</p> <p>0010: 复用功能 2</p> <p>...</p> <p>1001: 复用功能 9</p> <p>其余: 保留</p>

## 2.10.36. PD 管脚复用寄存器 2/PDSEL2(偏移: 9Ch)

控制 IO 复用状态

比特	名称	属性	默认值	功能描述
31:28	PD15_SEL	RW	0000	PD15 复用功能选择 0000: GPIO63 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
27:24	PD14_SEL	RW	0000	PD14 复用功能选择 0000: GPIO62 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
23:20	PD13_SEL	RW	0000	PD13 复用功能选择 0000: GPIO61 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
19:16	PD12_SEL	RW	0000	PD12 复用功能选择 0000: GPIO60 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PD11_SEL	RW	0000	PD11 复用功能选择 0000: GPIO59

				0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PD10_SEL	RW	0000	PD10 复用功能选择 0000: GPIO58 0001: 复用功能 1 0010: 复用功能 2 ... 0111: 复用功能 7 其余: 保留
7:4	PD9_SEL	RW	0000	PD9 复用功能选择 0000: GPIO57 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
3:0	PD8_SEL	RW	0000	PD8 复用功能选择 0000: GPIO56 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.37. PC/D 管脚上拉寄存器 PC/DPUR (偏移: a0h)

该位控制管脚是否上拉，对应的管脚在复用时此控制位仍然有效。

比特	名称	属性	默认值	功能描述
31:16	PDPUR	RW	0x0000	端口 PD 上拉使能配置寄存器 (对应 PD15-PD00) 1: 使能 0: 禁止
15:13	RSV	-	-	保留

12:0	PCPU	RW	0x0000	端口 PC 上拉使能配置寄存器 (对应 PC12-PC00) 1: 使能 0: 禁止
------	------	----	--------	--

注：PC15~13 由 RTC 域的 RPMU 寄存器决定

### 2.10.38. PC/D 管脚下拉寄存器/PC/DPDR (偏移：A4h)

该位控制管脚是否下拉，对应的管脚在复用时此控制位仍然有效。

比特	名称	属性	默认值	功能描述
31:16	PDPD	RW	0x0000	端口 PC 下拉使能配置寄存器 (对应 PD15-PC00) 1: 使能 0: 禁止
15:13	RSV	-	-	保留
12:0	PCPD	RW	0x0000	端口 PC 下拉使能配置寄存器 (对应 PC12-PC00) 1: 使能 0: 禁止

注：PC15~13 由 RTC 域的 RPMU 寄存器决定

### 2.10.39. PC 管脚驱动选择寄存器 PCSTR (偏移：A8h)

比特	名称	属性	默认值	功能描述
25:24	PC12_STH	RW	00	PC12 输出驱动设置
23:22	PC11_STH	RW	00	PC11 输出驱动设置
21:20	PC10_STH	RW	00	PC10 输出驱动设置
19:18	PC9_STH	RW	00	PC9 输出驱动设置
17:16	PC8_STH	RW	00	PC8 输出驱动设置
15:14	PC7_STH	RW	00	PC7 输出驱动设置
13:12	PC6_STH	RW	00	PC6 输出驱动设置
11:10	PC5_STH	RW	00	PC5 输出驱动设置
9:8	PC4_STH	RW	00	PC4 输出驱动设置
7:6	PC3_STH	RW	00	PC3 输出驱动设置
5:4	PC2_STH	RW	00	PC2 输出驱动设置
3:2	PC1_STH	RW	00	PC1 输出驱动设置

1:0	PC0_STH	RW	00	PC0 输出驱动设置 具体驱动能力参见 PA0_STH
-----	---------	----	----	--------------------------------

注：PC15~13 由 RTC 域的 RPMU 寄存器决定

#### 2.10.40. PD 管脚驱动选择寄存器 PDSTR（偏移：ACh）

比特	名称	属性	默认值	功能描述
31:30	PD15_STH	RW	00	PD15 输出驱动设置
29:28	PD14_STH	RW	00	PD14 输出驱动设置
27:26	PD13_STH	RW	00	PD13 输出驱动设置
25:24	PD12_STH	RW	00	PD12 输出驱动设置
...	...	...	...	...
7:6	PD3_STH	RW	00	PD3 输出驱动设置
5:4	PD2_STH	RW	00	PD2 输出驱动设置
3:2	PD1_STH	RW	00	PD1 输出驱动设置
1:0	PD0_STH	RW	00	PD0 输出驱动设置 具体驱动能力参见 PA0_STH

#### 2.10.41. PC/D 管脚施密特使能寄存器 PC/DSMTR（偏移：B0h）

比特	名称	属性	默认值	功能描述
31:16	PD_SMTEN	RW	0xFFFF	端口 PD 施密特使能配置寄存器 (对应 PD15-PD00) 1: 使能 0: 禁止
15:13	RSV	-	-	保留
12:0	PC_SMTEN	RW	0x1FFF	端口 PC 施密特使能配置寄存器 (对应 PC12-PC00) 1: 使能 0: 禁止

注：PC15~13 由 RTC 域的 RPMU 寄存器决定

## 2.10.42. PC/D 管脚 OD 使能寄存器 PC/DODR (偏移: B4h)

比特	名称	属性	默认值	功能描述
31:16	PD_ODEN	RW	0x0000	端口 PD open-drain 使能配置寄存器 (对应 PD15-PD00) 1: 使能 0: 禁止
15:13	RSV	-	-	保留
12:0	PC_ODEN	RW	0x0000	端口 PC open-drain 使能配置寄存器 (对应 PC12-PC00) 1: 使能 0: 禁止

注: PC15~13 由 RTC 域的 RPMU 寄存器决定

## 2.10.43. PC/D 管脚数模选择寄存器 PC/DADS (偏移: B8h)

比特	名称	属性	默认值	功能描述
31:16	PD_ADEN	RW	0xFFCF	端口 PD 数模选择寄存器 (对应 PD15-PD00) 1: 配置为模拟端口 0: 配置为数字端口
15:13	RSV	-	-	保留
12:0	PC_ADEN	RW	0x1FFF	端口 PC 数模选择寄存器 (对应 PC12-PC00) 1: 配置为模拟端口 0: 配置为数字端口

注: PC15~13 由 RTC 域的 RPMU 寄存器决定

## 2.10.44. PE 管用寄存器 1/PESEL1(偏移: C0h)

控制 IO 复用状态

比特	名称	属性	默认值	功能描述
31:28	PE7_SEL	RW	0000	PE7 复用功能选择 0000: GPIO71 0001: 复用功能 1 ... 1001: 复用功能 9

				其余：保留
27:24	PE6_SEL	RW	0000	PE6 复用功能选择 0000: GPIO70 0001: 复用功能 1 ... 1001: 复用功能 9 其余：保留
23:20	PE5_SEL	RW	0000	PC5 复用功能选择 0000: GPIO69 0001: 复用功能 1 ... 1001: 复用功能 9 其余：保留
19:16	PE4_SEL	RW	0000	PC4 复用功能选择 0000: GPIO68 0001: 复用功能 1 ... 1001: 复用功能 9 其余：保留
15:12	PE3_SEL	RW	0000	PC3 复用功能选择 0000: GPIO67 0001: 复用功能 1 ... 1001: 复用功能 9 其余：保留
11:8	PE2_SEL	RW	0000	PC2 复用功能选择 0000: GPIO66 0001: 复用功能 1 ... 1001: 复用功能 9 其余：保留
7:4	PE1_SEL	RW	0000	PC1 复用功能选择 0000: GPIO65 0001: 复用功能 1 ...

				1001: 复用功能 9 其余: 保留
3:0	PE0_SEL	RW	0000	PC0 复用功能选择 0000: GPIO64 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留

#### 2.10.45. PE 管脚复用寄存器 2/PESEL2(偏移: C4h)

比特	名称	属性	默认值	功能描述
31:28	PE15_SEL	RW	0000	PE15 复用功能选择 0000: GPIO79 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留
27:24	PE14_SEL	RW	0000	PE14 复用功能选择 0000: GPIO78 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留
23:20	PE13_SEL	RW	0000	PE13 复用功能选择 0000: GPIO77 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留
19:16	PE12_SEL	RW	0000	PE12 复用功能选择 0000: GPIO76 0001: 复用功能 1 ...

				1001: 复用功能 9 其余: 保留
15:12	PE11_SEL	RW	0000	PE11 复用功能选择 0000: GPIO75 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留
11:8	PE10_SEL	RW	0000	PE10 复用功能选择 0000: GPIO74 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留
7:4	PE9_SEL	RW	0000	PE9 复用功能选择 0000: GPIO73 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留
3:0	PE8_SEL	RW	0000	PE8 复用功能选择 0000: GPIO72 0001: 复用功能 1 ... 1001: 复用功能 9 其余: 保留

#### 2.10.46. PF 管脚复用寄存器 1/PFSEL1(偏移: C8h)

控制 IO 复用状态

比特	名称	属性	默认值	功能描述
19:16	PF4_SEL	RW	0000	PF4 复用功能选择 0000: GPIO84 0001: 复用功能 1

				0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
15:12	PF3_SEL	RW	0000	PF3 复用功能选择(默认 BOOT) 0000: GPIO83 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
11:8	PF2_SEL	RW	0000	PF2 复用功能选择 0000: GPIO82 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
7:4	PF1_SEL	RW	0000	PF1 复用功能选择 0000: GPIO81 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留
3:0	PF0_SEL	RW	0000	PF0 复用功能选择 0000: GPIO80 0001: 复用功能 1 0010: 复用功能 2 ... 1001: 复用功能 9 其余: 保留

### 2.10.47. PE/F 管脚上拉寄存器 PE/FPUR (偏移: D0h)

该位控制管脚是否上拉，对应的管脚在复用时此控制位仍然有效。

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20:16	PFPUR	RW	0x08	端口 PF 上拉使能配置寄存器 (对应 PF4-PF00) 1: 使能 0: 禁止
15:0	PEPUR	RW	0x0000	端口 PE 上拉使能配置寄存器 (对应 PE15-PE00) 1: 使能 0: 禁止

### 2.10.48. PE/F 管脚下拉寄存器/PE/FPDR (偏移: D4h)

该位控制管脚是否下拉，对应的管脚在复用时此控制位仍然有效。

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20:16	PFPDR	RW	0x00	端口 PF 下拉使能配置寄存器 (对应 PF4-PF00) 1: 使能 0: 禁止
15:0	PEPDR	RW	0x0000	端口 PE 下拉使能配置寄存器 (对应 PE15-PE00) 1: 使能 0: 禁止

### 2.10.49. PE 管脚驱动选择寄存器 PESTR (偏移: D8h)

比特	名称	属性	默认值	功能描述
31:30	PE15_STH	RW	00	PE15 输出驱动设置
29:28	PE14_STH	RW	00	PE14 输出驱动设置
...	...	...	...	...
5:4	PE2_STH	RW	00	PE2 输出驱动设置
3:2	PE1_STH	RW	00	PE1 输出驱动设置

1:0	PE0_STH	RW	00	PE0 输出驱动设置 具体驱动能力参见 PA0_STH
-----	---------	----	----	--------------------------------

### 2.10.50. PF 管脚驱动选择寄存器 PFSTR (偏移: DCh)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9:8	PF4_STH	RW	00	PF4 输出驱动设置
7:6	PF3_STH	RW	00	PF3 输出驱动设置
5:4	PF2_STH	RW	00	PF2 输出驱动设置
3:2	PF1_STH	RW	00	PF1 输出驱动设置
1:0	PF0_STH	RW	00	PF0 输出驱动设置

### 2.10.51. PE/F 管脚施密特使能寄存器 PE/FSMTR (偏移: E0h)

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20:16	PF_SMTEN	RW	0x1F	端口 PF 施密特使能配置寄存器 (对应 PF04-PF00) 1: 使能 0: 禁止
15:0	PE_SMTEN	RW	0xFFFF	端口 PE 施密特使能配置寄存器 (对应 PE15-PE00) 1: 使能 0: 禁止

### 2.10.52. PE/F 管脚 OD 使能寄存器 PE/FODR (偏移: E4h)

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20:16	PF_ODEN	RW	0x00	端口 PF open-drain 使能配置寄存器 (对应 PF4-PF00) 1: 使能 0: 禁止
15:0	PE_ODEN	RW	0x0000	端口 PE open-drain 使能配置寄存器 (对应 PE15-PC00)

				1: 使能 0: 禁止
--	--	--	--	----------------

### 2.10.53. PE/F 管脚数模选择寄存器 PE/FADS (偏移: E8h)

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20:16	PF_ADEN	RW	0x17	端口 PF 数模选择寄存器 (对应 PF04-PF00) 1: 配置为模拟端口 0: 配置为数字端口
15:0	PE_ADEN	RW	0xFFFF	端口 PE 数模选择寄存器 (对应 PE15-PE00) 1: 配置为模拟端口 0: 配置为数字端口

## 3. 复位和时钟单元

### 3.1. 复位源

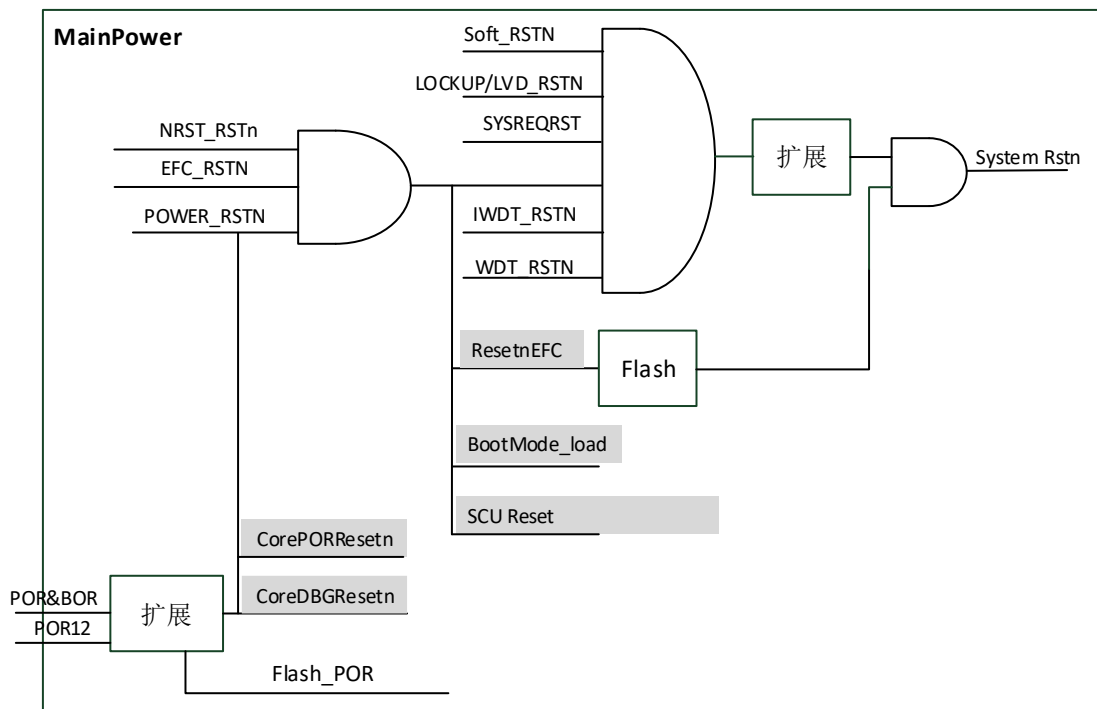
芯片有多个复位源，包括 POR 复位，检测异常复位，看门狗复位，软复位等。具体复位源如下表：

表格 3-1 系统复位源

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。
EFC_RST	复位主区：会产生 NVR 预加载，所有的 SCU 寄存器被复位，EFC 被复位，模拟 IP 也被复位
LVD_resetrn	复位主区，不影响模拟 IP、EFC 控制器、系统寄存器（偏移量 0x60-0xE8 除外）。
LOCKUP_resetrn	
SRST	
SYSREQRST	
IWDT_resetrn	
WDT_resetrn	
RTC 软复位	复位 RTC 待机区域

各复位源之间的关系如图 3-1 所示。

图 3-1 复位关系图



### 3.1.1. 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。

### 3.1.2. BOR(Brown Out Reset)复位:

欠压复位, 复位主区和 IWDT 模块。BOR 电压见数据手册。

### 3.1.3. 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。

### 3.1.4. 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。

### 3.1.5. EFC 复位:

EFC 寄存器复位，继而引起芯片复位。

### 3.1.6. 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。

### 3.1.7. LOCKUP 复位:

当处理器遇到严重的错误时，如在 HardFault 处理函数里再次遇到 HardFault 异常，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。

### 3.1.8. Soft Reset:

在系统寄存器 RCR 的 BIT30 写 0，产生系统复位。只对主区有效。

### 3.1.9. SYSREQRST:

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

### 3.1.10. IWDT 复位:

IWDT 计数器溢出产生系统复位。只对主区有效。

### 3.1.11. WDT 复位:

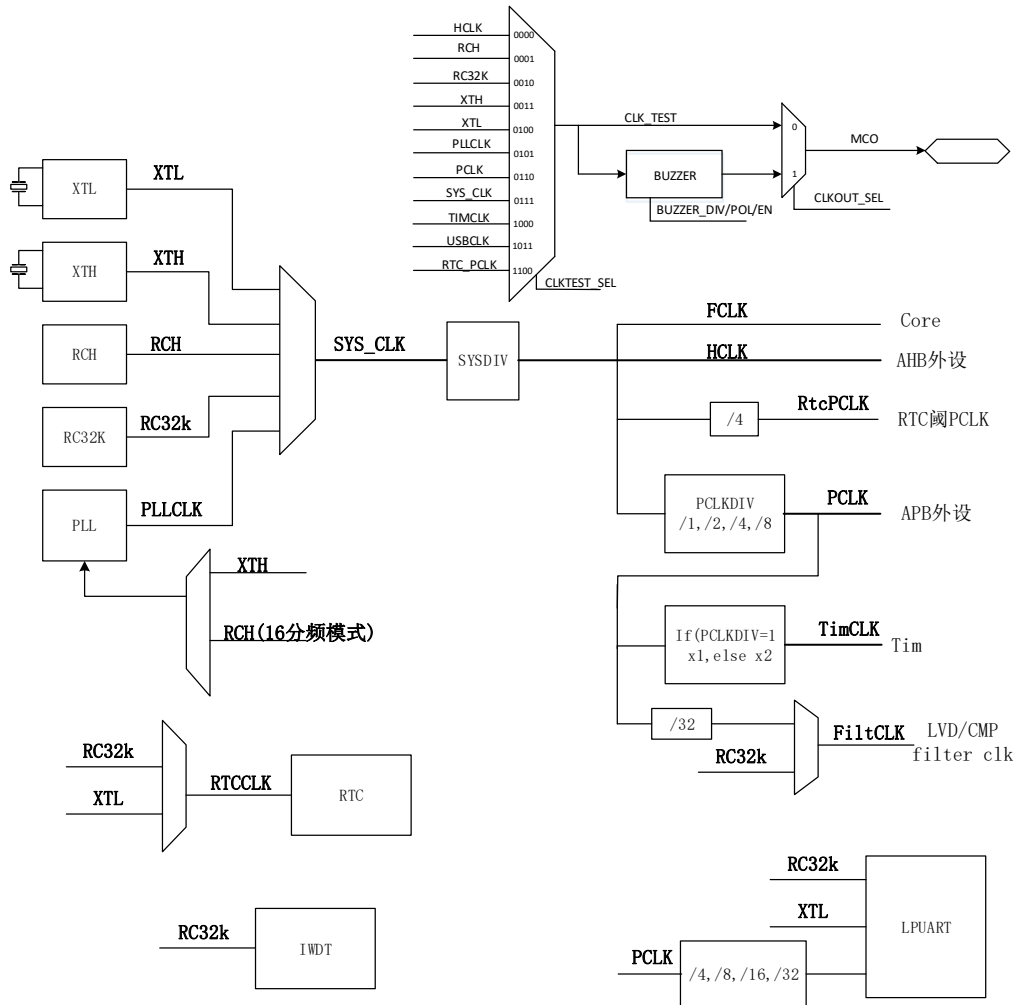
WDT 计数器溢出产生系统复位。只对主区有效。

### 3.1.12. RTC 域复位:

RPMU 模块的 CR 寄存器 BIT6 写 0 产生 RTC 域复位。只对 RTC 域有效。

## 3.2. 时钟结构图

图 3-2 时钟结构图



## 3.3. 系统时钟选择

ACM32F403 共有 5 个系统时钟源：

- 内部高速 RC 产生的 RCH 时钟，最高 64M
- 内部低速 32K 产生的 RC32K 时钟，作为低功耗时钟
- 外部高速晶振时钟 XTH
- 外部低速晶振时钟 XTL
- 从 PLL 产生的最高 180M 时钟

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：

表格 3-2 系统时钟选择

SYS_CLK_SEL	系统时钟源
000	RCH
001	RC32K
010	XTH
011	XTL
100	PLLCLK

系统上电后，默认使用 RCH 作为系统时钟，RCH 时钟的频率约为 64MHz(正负偏差 1%以内)。当系统从 STOP 模式唤醒后，自动启用 RCH 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 RCH/16(将 RCH\_DIV 设为 1)。PLL 的输入时钟范围为 3MHz-48MHz。

### 3.4. 时钟输出

ACM32F403 芯片的时钟输出可以通过 MCO 引脚或 MCO2 引脚。如图 3-1 所示，输出的时钟信号可以通过 CLKTEST\_SEL 选择，可以来自于系统时钟 SYS\_CLK，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 MCO/MCO2，也可以通过一个 Buzzer 控制电路再输出到 MCO/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。

## 4. 嵌套矢量中断控制器（NVIC）

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)

### 4.1. 主要特性

- 支持 40 路可屏蔽向量中断。
- 8 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。

### 4.2. 中断源

中断号	中断源	备注
Int0	Watchdog	
Int1	RTC	
Int2	EFC_INT	
Int3	GPIO1	包括 PA/PB 的所有引脚
Int4	GPIO2	包括 PC/PD 的所有引脚
Int5	EXTI	
Int6	SRAM Parity	
Int7	CLKRDY_INT	时钟 Ready 中断，中断使能可见系统寄存器 CIR

Int8	UART4	
Int9	DMA	
Int10	UART3	
Int11	RSV	保留
Int12	ADC	
Int13	TIM1_BRK_UP_TRG_COM	
Int14	TIM1_CC	捕获/比较中断
Int15	TIM2	
Int16	TIM3	
Int17	TIM6	
Int18	TIM7	
Int19	TIM14	
Int20	TIM15	
Int21	TIM16	
Int22	TIM17	
Int23	I2C1	
Int24	I2C2	
Int25	SPI1	
Int26	SPI2	
Int27	UART1	
Int28	UART2	
Int29	LPUART	
Int30	SPI3	
Int31	AES	
Int32	USB2.0	
Int33	DAC	
Int34	I2S1	
Int35	GPIO3	包括 PE/PF 的所有引脚
Int36	CAN1	

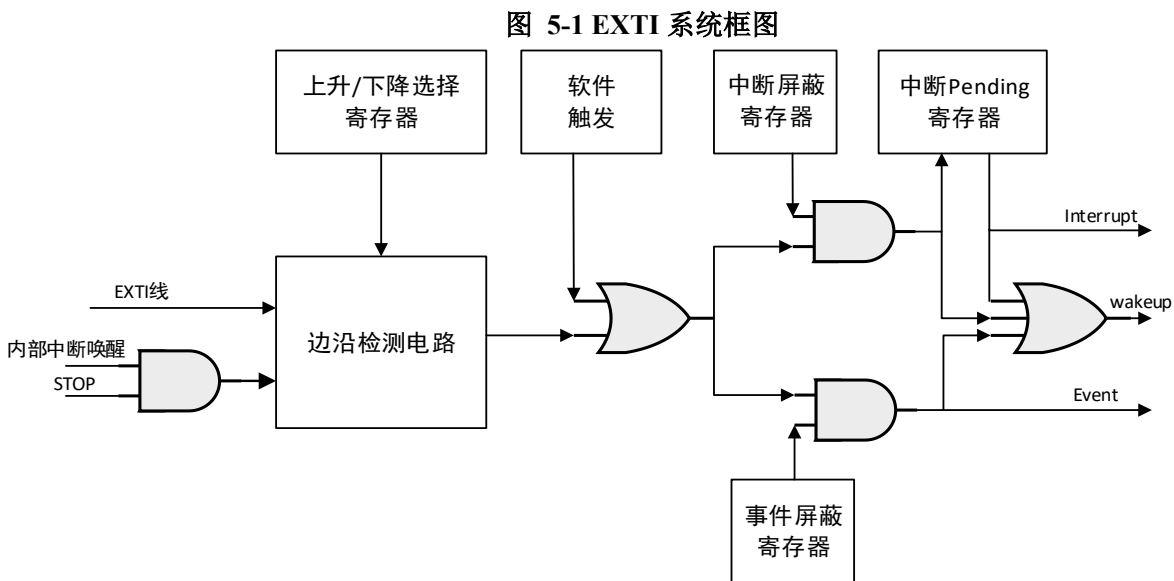
---

Int37	CAN2	
Int38	FPU_INT	
Int39	TIM4	
Int40	SPI4	

## 5. 外部中断/事件控制器（EXTI）

EXTI 包含 23 个相互独立的边沿检测电路并且可以向处理器产生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。

### 5.1. 系统框图



### 5.2. 主要特性

EXTI 控制器的主要特性如下：

- 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
- 支持多达 23 个软件的中断/事件请求

### 5.3. 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 7 根线，包括 LVD、RTC、LPUART、IWDG（唤醒）、COMP1、COMP2 和 USB\_Wakeup。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。

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	LPUART (中断)
19	IWDT
20	COMP1
21	COMP2
22	Reserved
23	USB_Wakeup (中断)
24~31	Reserved

## 5.4. 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 Core 的系统控制寄存器中使能 SEVONPEND 位。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’。

## 5.5. 功能说明

### 5.5.1. EXTI 中断

EXTI 具有 23 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

### 5.5.2. 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 23 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 23 个中断线的使能位 IENR 或配置 23 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 23 个中断线中的请求可以被正确地响应。

## 5.6. EXTI 寄存器

寄存器映射如下表

EXTI 寄存器基地址：0x4001\_0400

地址	名称	描述
0x0000	IENR	中断使能寄存器
0x0004	EENR	事件使能寄存器
0x0008	RTENR	上升沿触发使能寄存器
0x000C	FTENR	下降沿触发使能寄存器
0x0010	SWIER	软件中断事件寄存器
0x0014	PDR	中断挂起寄存器
0x0018	EXTICR1	外部 I/O 选择寄存器 1
0x001C	EXTICR2	外部 I/O 选择寄存器 2

#### 5.6.1. 中断使能寄存器/IENR (偏移：00h)

比特	名称	属性	默认值	功能描述
31:24	RSV	-	-	保留

23:0	INTENx	RW	0x0	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。
------	--------	----	-----	--

### 5.6.2. 事件使能寄存器/EENR (偏移: 04h)

比特	名称	属性	默认值	功能描述
31:24	RSV	-	-	保留
23:0	EVENx	RW	0x0	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。

### 5.6.3. 上升沿触发使能寄存器/RTENR (偏移: 08h)

比特	名称	属性	默认值	功能描述
31:23	RSV	-	-	保留
23:0	RTENx	RW	0x0	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。

### 5.6.4. 下降沿触发使能寄存器/FTENR (偏移: 0Ch)

比特	名称	属性	默认值	功能描述
31:23	RSV	-	-	保留
23:0	FTENx	RW	0x0	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。

## 5.6.5. 软件中断事件寄存器/SWIER (偏移: 10h)

比特	名称	属性	默认值	功能描述
31:23	RSV	-	-	保留
23:0	SWIE <sub>x</sub>	WO	0x0	EXTI <sub>x</sub> 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。

## 5.6.6. 中断挂起寄存器/PDR (偏移: 14h)

比特	名称	属性	默认值	功能描述
31:23	RSV	-	-	保留
23:0	PD <sub>x</sub>	RC_W1	0x0	EXTI <sub>x</sub> 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。

## 5.6.7. 外部中断配置寄存器 1/EXTICR1 (偏移: 18h)

比特	名称	属性	默认值	功能描述
28:31	EXTI 7	RW	0000	EXTI 7 源选择。 0000: PA7 0001: PB7 0010: PC7 0011: PD7 0100: PE7
24:27	EXTI 6	RW	0000	EXTI 6 源选择。 0000: PA6 0001: PB6 0010: PC6 0011: PD6

				0100: PE6
20:23	EXTI 5	RW	0000	EXTI 5 源选择。 0000: PA5 0001: PB5 0010: PC5 0011: PD5 0100: PE5
16:19	EXTI 4	RW	0000	EXTI 4 源选择。 0000: PA4 0001: PB4 0010: PC4 0011: PD4 0100: PE4 0100: PF4
12:15	EXTI 3	RW	0000	EXTI 3 源选择。 0000: PA3 0001: PB3 0010: PC3 0011: PD3 0100: PE3 0100: PF3
8:11	EXTI 2	RW	0000	EXTI 2 源选择。 0000: PA2 0001: PB2 0010: PC2 0011: PD2 0100: PE2 0100: PF2
4:7	EXTI 1	RW	0000	EXTI 1 源选择。 0000: PA1 0001: PB1 0010: PC1 0011: PD1 0100: PE1 0100: PF1

0:3	EXTI 0	RW	0000	EXTI 0 源选择。 0000: PA0 0001: PB0 0010: PC0 0011: PD0 0100: PE0 0100: PF0
-----	--------	----	------	---

### 5.6.8. 外部中断配置寄存器 2/EXTICR2 (偏移: 1Ch)

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

				0100: PE12
12:15	EXTI 11	RW	0000	EXTI 11 源选择。 0000: PA11 0001: PB11 0010: PC11 0011: PD11 0100: PE11
8:11	EXTI 10	RW	0000	EXTI 10 源选择。 0000: PA10 0001: PB10 0010: PC10 0011: PD10 0100: PE10
4:7	EXTI 9	RW	0000	EXTI 9 源选择。 0000: PA9 0001: PB9 0010: PC9 0011: PD9 0100: PE9
0:3	EXTI 8	RW	0000	EXTI 8 源选择。 0000: PA8 0001: PB8 0010: PC8 0011: PD8 0100: PE8

## 6. 片上 Flash 控制器（EFC）

### 6.1. 概述

芯片上集成了 512KB 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。

### 6.2. 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。
- Program/Erase 等待时间由软件配置。
- 主区多达 1024 个 Page，每个 page 512 字节。
- NVR 区有 2 个 Page，每个 Page 512 字节。
- 支持对 NVR 区域 Program/Erase 保护功能（OTP）。
- Page Erase 时间为 4ms，Word Program 写 40us（max），Read 时间 47ns（max）。

### 6.3. 功能描述

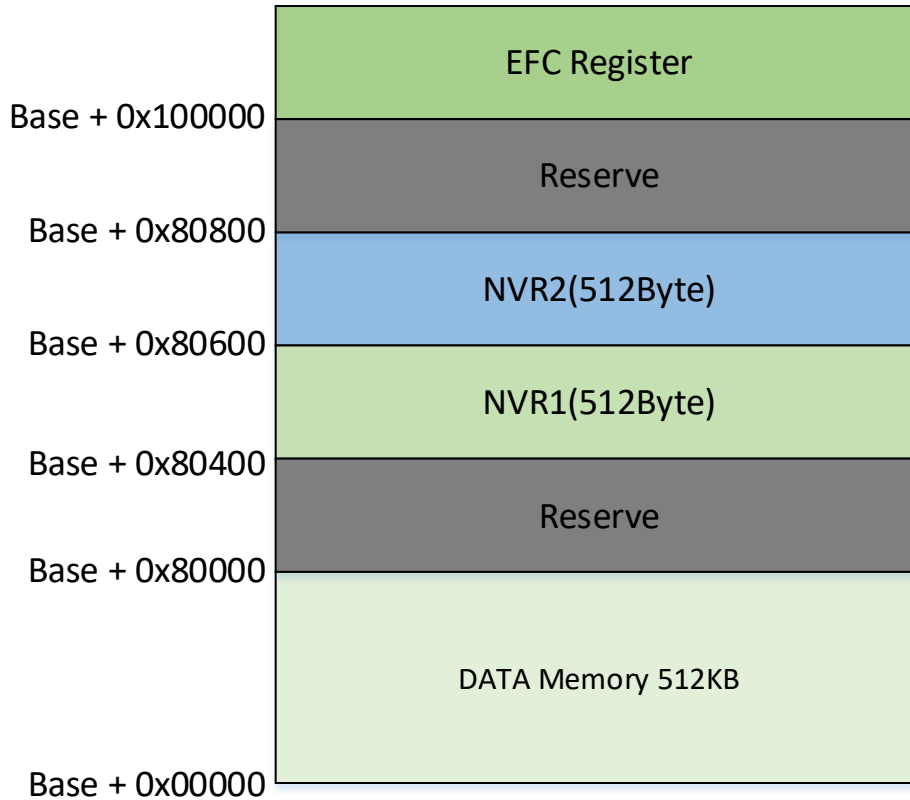
#### 6.3.1. 主区闪存结构

主区闪存包括 512K 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 1023	0x0007FE00~0x0007FFFF	512Byte

### 6.3.2. 储存地址映射

图 6-1 eFlash 地址映射



注：Base：ROM 启动为 0x1000\_0000；eFlash 启动为 0x0000\_0000。

### 6.3.3. 读操作

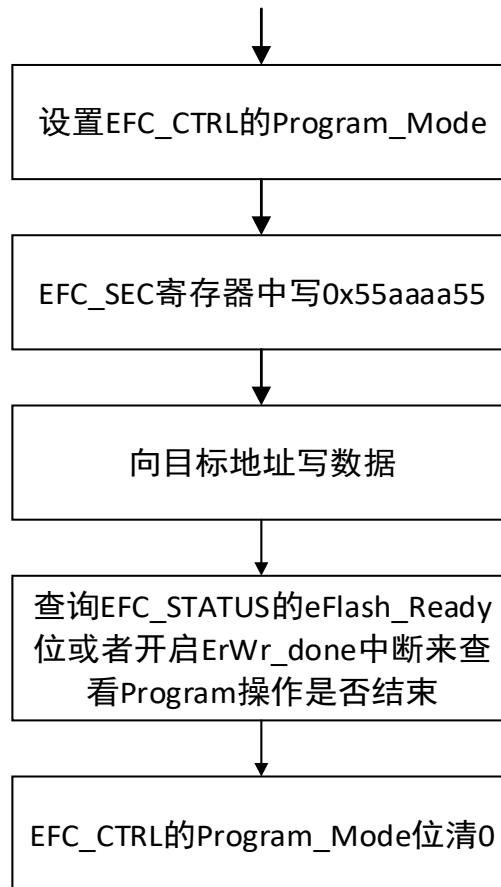
闪存可以像普通存储空间一样直接寻址访问。在 eFlash 上电稳定后就可以执行读操作。读操作注意配置读等待时间 RD\_WAIT，最小值为 47ns。

### 6.3.4. 闪存编程操作

EFC 提供了 32 位整字编程功能，用来修改主闪存存储器的内容。

eFlash 上电稳定后可以执行 Program 操作。Program 操作之前需要向 EFC\_SEC 寄存器内写 0x55AAAA55，否则 EFC 忽略此次 Program 操作。

图 6-2 闪存编程操作流程

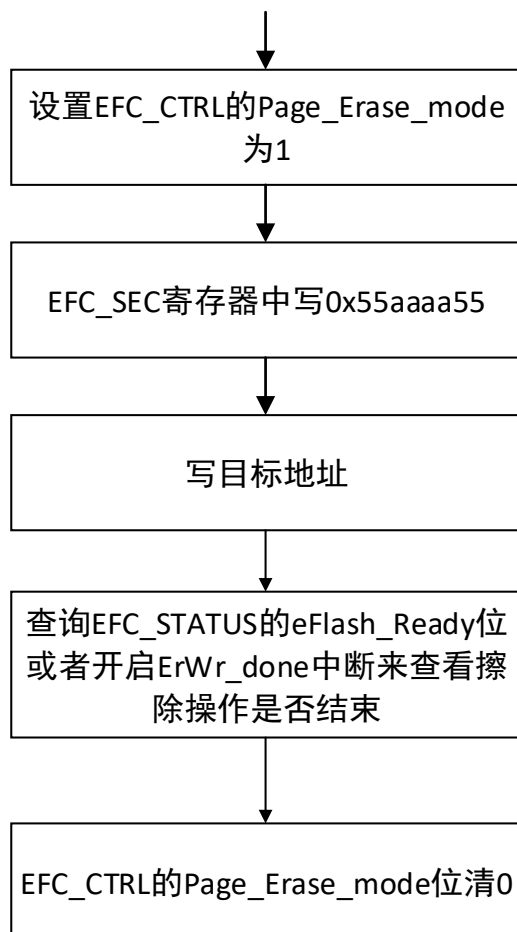


注：进行 program 操作时时钟频率需要大于 0.5M

### 6.3.5. 页擦除

EFC 的页擦除功能将一个主闪存存储页的内容初始化为高电平。每一页都可以被独立擦除，不影响其他页的内容。

图 6-3 页擦除操作流程



## 6.4. 寄存器描述

寄存器基地址：ROM 启动为 0x10100000 ， eFlash 启动为 0x00100000。

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。
0x0C	EFC_tERASE	擦除等待周期寄存器。
0x10	EFC_tPROG	写等待周期寄存器。

0x14	EFC_STATUS	状态寄存器。
0x18	EFC_INTSTATUS	中断状态寄存器。
0x1C	EFC_INTEN	中断使能寄存器。

#### 6.4.1. 控制寄存器 EFC\_CTRL(偏移: 00h)

比特	名称	属性	默认值	功能描述
31:14	RSV	-	-	保留
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。 注: 复位持续时间至少为 1us。复位释放后需要延迟 10us 后才能对 eFlash 发起 Read/Program/Erase 操作。
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。 注: 退出 SLEEP 模式后, 要延时至少 10us 再去对 eFlash 进行 Read/Program/Erase 操作。
11:7	Rd_Wait	RW	0xC	读等待时间设置位。 Rd_Wait 典型值: 系统时钟<=32M时, Rd_Wait=0; 系统时钟=45M时, Rd_Wait=1; 系统时钟=64M时, Rd_Wait=2; 系统时钟=90M时, Rd_Wait=3; 系统时钟=120M时, Rd_Wait=5; 系统时钟=150M时, Rd_Wait=6; 系统时钟=180M时, Rd_Wait=8 当系统频率较高时, 开启硬件加速功能能显著提高运行效率。开启硬件加速功能请参考固件库中的 __ACCELERATE_PRESENT和 __ACCELERATE_EH_PRESENT宏开关。
6:2	RSV	--	--	保留
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	Program_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。

注：Program\_Mode，Page\_Erase\_Mode 不能同时写 1，否则会造成擦写失败并且 EFC\_INTSTATUS 的 Wrong\_Program 位会置 1。

注：Program/Erase 发起后在 EFC\_STATUS 中的 eFlash\_Ready 位不为 1 时不可以改变控制寄存器的值，否则也会造成 Program/Erase 失败并且 Wrong\_program 位会置 1。

#### 6.4.2. 写擦安全寄存器 EFC\_SEC (偏移：04h)

比特	名称	属性	默认值	功能描述
31:0	Write_Lock_Ser	WO	0x0	eFlash 在执行 Program/Erase 操作前，须向此寄存器内写 0x55AAAA55 值，否则控制器忽略此次操作。

#### 6.4.3. tERASE 等待周期寄存器 (EFC\_TERASE) (偏移：0ch)

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20:0	ErCycle	RW	0x1FFFFFF	tERASE 等待周期设置位。每一个单位代表 1 个系统时钟周期。Pageerase 操作设置时间要求范围：3.2~4ms。 注：TERASE 等待周期必须在范围内。

#### 6.4.4. tPROG 等待周期寄存器 (EFC\_TPROG) (偏移：10h)

比特	名称	属性	默认值	功能描述
31:11	RSV	-	-	保留

10:0	ProgCycle	RW	0x7FF	tPROG 等待周期设置位，每一个单位代表 1 个系统时钟周期。设置时间要求范围为 8~10us。 注：进行 Program 操作时时钟频率需要大于 0.5M。
------	-----------	----	-------	---

#### 6.4.5. 状态寄存器 EFC\_STATUS (偏移：14h)

比特	名称	属性	默认值	功能描述
31:5	RSV	-	-	保留
4	NVR2_Lock	RO	0/1	NVR2 区是否锁住。 1: NVR2 区已经锁住；0: NVR2 区没有锁住。
3	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。
2: 1	RSV	-	-	保留
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。

#### 6.4.6. 中断状态寄存器 EFC\_INTSTATUS (偏移：18h)

比特	名称	属性	默认值	功能描述
31:7	RSV	--	--	保留
6	Wrong_program	RO_W1	0	1: Program/Erase 操作有误。当 EFC_CTRL 寄存器的 Program_mode, Page_erase_mode 位都为 1 时并进行 Program&Erase 操作，或在 Program/Erase 操作还没完成（EFC_STATUS 中的 eFlash_Ready 为 0）但 EFC_CTRL 相应操作位被写 0 时，此位均会置 1。 0: 正常状态。 写 1 清 0。
5	Nvr_Err	RO_W1	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，此位会置 1。

				0: 正常状态。 写 1 清 0。
4: 1	RSV	--	--	保留
0	ErWr_done	RO_W1	0	Program/Erase 完成中断状态位, 如果中断允许, 则产生中断 1: Program/Erase 完成。 0: Program/Erase 未完成。 写 1 清 0。 注: Program/Erase 被 OTP 锁住的 eNVR 区时 EFC 不会发起任何操作, 此位不会置起。查 eNVR_Err 状态位可以确定是否发生过错误操作。

#### 6.4.7. 中断使能寄存器 EFC\_INTEN (偏移: 1ch)

比特	名称	属性	默认值	功能描述
31:7	RSV	-	-	保留
6	Wrong_program_IE	RW	0	Wrong_program 中断使能。 1: Wrong_program 中断使能; 0: Wrong_program 中断不使能。
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;

## 7. DMA 控制器（DMAC）

### 7.1. 概述

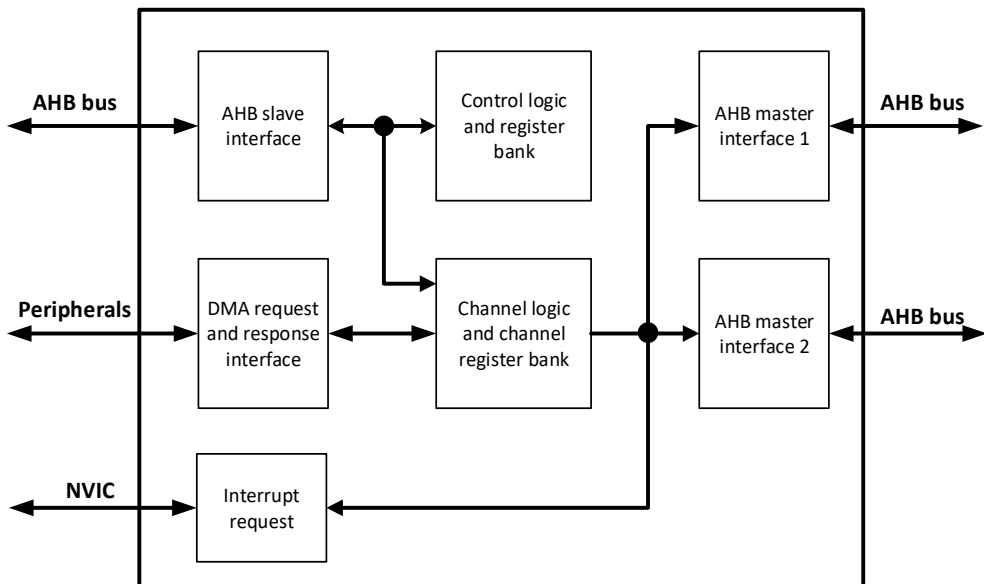
DMAC 即 DMA 控制器，提供了一种硬件的数据传输方式，无需 CPU 的介入，可以处理外设和存储器之间或者存储器和存储器之间的传输数据。因无 CPU 介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 8 通道，每个通道都可以处理一个或多个外设的存储器访问请求。DMA 控制器内部包含了仲裁器，用来仲裁多个 DMA 请求的优先级。

### 7.2. 主要特性

- 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 bytes 的内部 FIFO
- 支持 8/16/ 32 位宽的传输
- 支持 Big-endian 和 Little-endian 的字节存储次序配置
- 支持 DMA 中断功能
- 支持链表功能

## 7.3. 结构框图

图 7-1 DMA 结构框图



## 7.4. 功能描述

### 7.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，然后将读取的数据存储到目标地址。DMA 控制器基于 `DMACCxSrcAddr`、`DMACCxDestAddr`、`DMACCxCtrl` 寄存器的值计算下一次操作的源/目的地址。`DMACCxCtrl` 寄存器 `TransferSize` 位域用于控制传输的次数，`SWidth` 和 `DWidth` 位域决定每次读取和存储的字节数（字节/半字/字）。

`DMACCxCtrl` 寄存器 `TransferSize` 位域需要在 `DMACCxConfig` 的 `EN` 被使能之前填入，读 `TransferSize` 可以得到当前目标总线上剩余还未完成的数据量。当一次 DMA 传输完成后，`TransferSize` 自动清零。

### 7.4.2. 外设握手

DMA 控制器中引入了外设和存储器的握手机制，包括请求信号和应答信号：

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；

- 应答信号: 由 DMA 控制器响应, 表明 DMA 控制器已经发送 AHB 命令去访问外设。

### 7.4.3. 仲裁

当 DMA 控制器在同一时间接收到多个请求时, 仲裁器将根据硬件优先级来决定响应哪一个外设请求。

硬件优先级: 编号低的通道优先级高。例: 通道 0 和通道 1 配置为相同的软件优先级时, 通道 0 的优先级高于通道 1。

### 7.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法: 固定模式和递增模式。寄存器 DMACCxCtrl 的 DI 和 SI 位用来设置存储器和外设的地址生成算法。

在固定模式中, 地址一直固定为初始化的基地址 (DMACCxSrcAddr、DMACCxDestAddr)。

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

### 7.4.5. 链表功能

通道链表寄存器 DMACCxLLI 中, 可以写入下一个 32 位链表的地址。链表项必须由几个通道寄存器项组成 (DMACCxSrcAddr, DMACCxDestAddr, DMACCxLLI, DMACCxCtrl), 通过链表功能, DMA 控制器会自动载入下一次通道寄存器配置参数, 可以实现更复杂的 DMA 传输方式, 例如: 循环模式。

### 7.4.6. 存储器到存储器模式

将 DMACCxConfig 寄存器的 FlowCtrl 位域设置为 0 时可以使能存储器到存储器模式。在此模式下, DMA 通道传输数据时不依赖外设的请求信号。一旦 DMACCxConfig 的 EN 位被置 1, DMA 通道就立即开始传输数据, 直到 DMACCxCtrl 寄存器 TransferSize 达到 0, DMA 通道才会停止。

### 7.4.7. 突发模式

DMA 可以通过设置 DMACCxCtrl 的位域 DBSize 和 SBSize 来实现突发模式。DBSize 和 SBSize 的设置值必须一样（DBSize、SBSize 和 Burst size 的对应关系见下表）。设置为突发模式时，每次 DMA 请求将产生 Burst size 次数据传输（每次传输的长度为 DWidth 或 SWidth 设置值）。普通的 DMA 模式相当于 Burst size 为 1 的模式。

设置为突发模式时，DMACCxCtrl 的 TransferSize 依然为总的传输长度，但必须是 Burst size 的整数倍。

例如：如需使用 Burst size 为 8 的突发模式传输 128 个字（32 位）的数据，需设置 DWidth 和 SWidth 为 0b010（字 32 位），DBSIZE 和 SBSIZE 为 0b010（Burst size 为 8），TransferSize 为 128（刚好是 Burst size 为 8 的 16 倍）；传输时一共会产生 16 次 DMA 请求，每次 DMA 请求会产生 8 次传输，每次传输 1 个字的数据。

表格 7-1 DBSIZE/SBSIZE 对应的 burst size

DBSIZE/SBSIZE	Burst size
0b000	1
0b001	4
0b010	8
0b011	16
0b100	32
0b101	64
0b110	128
0b111	256

### 7.4.8. 中断

每个 DMA 通道都有一个专用的中断。中断事件有两种类型：传输完成和传输错误。每一个中断事件在状态寄存器和清除寄存器中有专用的标志位，其中 DMACIntTCStatus 指示传输完成中断的状态，DMACIntTCClr 寄存器负责清除传输完成中断状态，DMACIntErrStatus 指示传输错误中断的状态，DMACIntErrClr 寄存器负责清除传输错误中断状态，DMACRawIntTCStatus 指示传输完成原始中断的状态，DMACRawIntErrStatus 寄存器指示传输错误原始中断的状态。

## 7.4.9. 目标外设和源外设请求号

表格 7-2 目标外设和源外设请求号

请求号	请求源	备注
REQ 0	ADC 转换完成	
REQ 1	SPI1 发送请求。	
REQ 2	SPI1 接收请求。	
REQ 3	SPI2 发送请求。	
REQ 4	SPI2 接收请求。	
REQ 5	UART1 发送请求。	
REQ 6	UART1 接收请求。	
REQ 7	UART2 发送请求。	
REQ 8	UART2 接收请求。	
REQ 9	I2C1 发送请求。	
REQ10	I2C1 接收请求。	
REQ11	I2C2 发送请求。	
REQ12	I2C2 接收请求。	
REQ13	TIM1_CH1	
REQ14	TIM1_CH2	
REQ15	TIM1_CH3	
REQ16	TIM1_CH4	
REQ17	TIM1_UP	更新事件请求
REQ18	TIM1_TRIG_COM	触发或 COM 事件请求
REQ19	TIM3_CH3	
REQ20	TIM3_CH4 或 TIM3_UP	
REQ21	TIM3_CH1 或 TIM3_TRIG	
REQ22	TIM3_CH2	
REQ23	TIM6_UP	

REQ24	TIM15_CH1 或 TIM15_UP 或 TIM15_TRIG 或 TIM15_COM	
REQ25	TIM15_CH2	
REQ26	TIM16_CH1 或 TIM16_UP	
REQ27	UART3 发送请求	
REQ28	TIM17_CH1 或 TIM17_UP	
REQ29	UART3 接收请求	
REQ30	LPUART 发送请求	
REQ31	LPUART 接收请求	
REQ32	TIM2_CH3	
REQ33	TIM2_CH4 或 TIM2_UP	
REQ34	TIM2_CH1 或 TIM2_TRIG	
REQ35	TIM2_CH2	
REQ36	TIM7_UP	
REQ37	I2S1 发送请求	
REQ38	I2S1 接收请求	
REQ39	DAC1_CH1	
REQ40	DAC1_CH2	
REQ41	TIM4_CH3	
REQ42	TIM4_CH4 或 TIM4_UP	
REQ43	TIM4_CH4 或 TIM4_TRIG	
REQ44	TIM4_CH2	
REQ45	UART4 发送请求	
REQ46	UART4 接收请求	
REQ47	SPI3 发送请求	
REQ48	SPI3 接收请求	
REQ49	SPI4 发送请求	
REQ50	SPI4 接收请求	
REQ51~REQ63	保留	

## 7.5. 寄存器描述

DMAC 寄存器基地址：0x4002\_1000

偏置	名称	描述
0x00	DMACIntStatus	中断状态寄存器
0x04	DMACIntTCStatus	传输完成中断寄存器
0x08	DMACIntTCClr	传输完成中断清除寄存器
0x0C	DMACIntErrStatus	传输错误中断寄存器
0x10	DMACIntErrClr	传输错误中断清除寄存器
0x14	DMACRawIntTCStatus	传输完成原始中断寄存器
0x18	DMACRawIntErrStatus	传输错误原始中断寄存器
0x1C	DMACEnChStatus	通道使能状态寄存器
0x30	DMACConfig	DMAC 配置寄存器
0x34	DMACSyncLo	低同步寄存器
0x38	DMACSyncHi	高同步寄存器
0x100, 0x120, 0x140, 0x160, 0x180, 0x1A0, 0x1C0, 0x1E0	DMACCxSrcAddr	源通道 0/1/2/3/4/5/6/7 地址寄存器
0x104, 0x124, 0x144, 0x164, 0x184, 0x1A4, 0x1C4, 0x1E4	DMACCxDestAddr	目标通道 0/1/2/3/4/5/6/7 地址寄存器
0x108, 0x128, 0x148, 0x168, 0x188, 0x1A8, 0x1C8, 0x1E8	DMACCxLLI	通道 0/1/2/3/4/5/6/7 链接表寄存器
0x10C, 0x12C, 0x14C, 0x16C, 0x18C, 0x1AC, 0x1CC, 0x1EC	DMACCxCtrl	通道 0/1/2/3/4/5/6/7 控制寄存器
0x110, 0x130h, 0x150, 0x170, 0x190, 0x1B0, 0x1D0, 0x1F0	DMACCxConfig	通道 0/1/2/3/4/5/6/7 配置寄存器

### 7.5.1. 中断状态寄存器 DMACIntStatus（偏移：00h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	IntStatus	RO	0x0	中断状态寄存器。

				1: 表示对应的通道产生中断, 当 DMA 控制器传输完成 (DMACIntTCStatus) 或传输错误 (DMACIntErrStatus) 时触发该中断。 0: 没有产生中断
--	--	--	--	---

### 7.5.2. 传输完成中断寄存器 DMACIntTCStatus (偏移: 04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	IntTCStatus	RO	0x0	传输完成中断状态 1: 对应的通道传输完成中断 0: 没有传输完成中断 如果禁止了该中断, 可以查询 DMACRawIntTCStatus 寄存器。

### 7.5.3. 传输完成中断清除寄存器 DMACIntTCClr (偏移: 08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	IntTCClr	WO	0x0	传输完成中断状态清除 1: 写 1 将清除相应通道的传输完成状态。 0: 无动作

### 7.5.4. 传输错误中断寄存器 DMACIntErrStatus (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	IntErrStatus	RO	0x0	传输错误中断状态 1: 对应的通道传输错误中断。 0: 没有传输错误中断。 如果禁止了该中断, 可以查询

				DMACRawIntErrStatus 寄存器。
--	--	--	--	--------------------------

### 7.5.5. 传输错误中断清除寄存器 DMACIntErrClr (偏移: 10h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	IntErrClr	WO	0x0	传输错误中断状态清除。 1: 写 1 将清除相应通道的传输错误状态 0: 无动作

### 7.5.6. 传输完成原始中断寄存器 DMACRawIntTCStatus (偏移: 14h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	RawIntTCStatus	RO	0x0	传输完成原始中断状态。 1: 对应的通道传输完成 0: 没有传输完成

### 7.5.7. 传输错误原始中断寄存器 DMACRawIntErrStatus (偏移: 18h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	RawIntErrStatus	RO	0x0	传输错误中断状态。 1: 对应的通道传输错误 0: 没有错误

### 7.5.8. 通道使能状态寄存器 DMACEnChStatus (偏移: 1Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留

7:0	EnChStat	RO	0x0	1: 对应的通道已使能 0: 对应的通道未使能
-----	----------	----	-----	----------------------------

### 7.5.9. DMAC 配置寄存器 DMACConfig (偏移: 30h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	M2ENDIAN	RW	0	Master 2 字节存储次序配置 1: big-endian 模式 0: little-endian 模式
1	M1ENDIAN	RW	0	Master 1 字节存储次序配置 1: big-endian 模式 0: little-endian 模式
0	EN	RW	0	Dmac 使能 1: 打开 dmac 0: 关闭 dmac

### 7.5.10. 同步寄存器 DMACSyncLo (偏移: 34h)

比特	名称	属性	复位值	描述
31:0	SyncLo	RW	0xFFFFFFFF	DMAC 低位 (0~31) 同步逻辑控制 1: 对应通道同步逻辑不打开 0: 对应通道同步逻辑打开

注意: 当外设产生请求信号的时钟和 DMAC 的时钟不是同一个时钟时, 必须打开同步逻辑。

### 7.5.11. 同步寄存器 DMACSyncHi (偏移: 38h)

比特	名称	属性	复位值	描述
31:0	SyncHi	RW	0xFFFFFFFF	DMAC 高位 (32~63) 同步逻辑控制 1: 对应通道同步逻辑不打开 0: 对应通道同步逻辑打开

注意: 当外设产生请求信号的时钟和 DMAC 的时钟不是同一个时钟时, 必须打开同步逻辑。

### 7.5.12. 源通道地址寄存器 DMACCxSrcAddr (偏移: 100h, 120h, 140h, 160h, 180h, 1A0h, 1C0h, 1E0h)

比特	名称	属性	复位值	描述
31:0	DMACCxSrcAddr	RW	0x0	源通道地址寄存器

注意: 当通道使能打开时, 不允许改变该寄存器的值。

### 7.5.13. 目标通道地址寄存器 DMACCxDestAddr (偏移: 104h, 124h, 144h, 164h, 184h, 1A4h, 1C4h, 1E4h)

比特	名称	属性	复位值	描述
31:0	DMACCxDestAddr	RW	0x0	目标通道地址寄存器

注意: 当通道使能打开时, 不允许改变该寄存器的值。

### 7.5.14. 通道链接表寄存器 DMACCxLLI (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	0x0	通道链接表, 表示下一个 32 位的链接表地址 Addr[31:2], 其中 Addr[1:0]为 0.
1	RSV	-	-	保留
0	LM	RW	0	下一个 LLI 的 master 号 1: master 2 0: master 1

### 7.5.15. 通道控制寄存器 DMACCxCtrl ( 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	当前传输是否使能中断产生逻辑。 1: 使能。 0: 禁止。
30:28	RSV	-	-	保留
27	DI	RW	0	目标地址递增使能位, 置 1 使能目标地址递增, 每个 transfer 后目标地址将加上一个递增量
26	SI	RW	0	源地址递增使能位, 置 1 使能源地址递增, 每个 transfer 后源地址将加上一个递增量
25	D	RW	0	目标 AHB master 选择 1: master 1 0: master 0
24	S	RW	0	源 AHB master 选择 1: master 1 0: master 0
23:21	DWidth	RW	000	目标传输位宽。源和目标位宽可以不同, 硬件会自动打包、解包数据。 DWidth 对应的位宽如下: 000: 字节 (8 位) 001: 半字 (16 位) 010: 字 (32 位) 其它: 保留
20:18	SWidth	RW	000	源传输位宽。源和目标位宽可以不同, 硬件会自动打包、解包数据。 SWidth 对应的位宽如下: 000: 字节 (8 位) 001: 半字 (16 位) 010: 字 (32 位) 其它: 保留
17:15	DBSize	RW	000	目标 burst size。一个 burst 由 DBSize 个 transfer 组成。用户必须把这个值设置成与目标外设的

				burst size 一致。Burst size 是目标外设的 DMACxBREQ 为高时传输的数据量。DBSize 对应的 Burst size 见下表
14:12	SBSize	RW	000	源 burst size。一个 burst 由 SBSize 个 transfer 组成。用户必须把这个值设置成与源外设的 burst size 一致。Burst size 是源外设的 DMACxBREQ 为高时传输的数据量。SBSize 对应的 Burst size 见功能描述“突发模式”章节
11:0	TransferSize	RW	0x0	Transfer Size(写操作的单位是源位宽，读操作的单位是目标位宽) 读该位可以得到当前目标总线上剩余还未完成的数据量。当一次 DMA 传输完成后，该位自动清零。

注意：当通道使能打开时，不允许改变该寄存器的值。在通道使能打开前将各个通道寄存器配置好，打开使能后，配置好的寄存器即生效。

### 7.5.16. 通道配置寄存器 DMACCxConfig (偏移：110h, 130h, 150h, 170h, 190h, 1B0h, 1D0h, 1F0h)

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
24:19	DestPeriph	RW	0x0	目标外设 ID。对应关系见功能描述“目标外设和源外设请求号”章节。当目标是存储器时，该值无效。
18	Halt	RW	0	1: 中止当前 DMA 传输，中止后通道 FIFO 中的内容会被清除。 0: 正常
17	Active	RO	0	1: 通道 FIFO 中有数据 0: 通道 FIFO 中没有数据
16	Lock	RW	0	LOCK (芯片无此功能，必须设为 0)
15	ITC	RW	0	传输完成中断使能 1: 使能该通道的传输完成中断 0: 屏蔽该通道的传输完成中断

14	IE	RW	0	传输错误中断使能 1: 使能该通道的传输错误中断 0: 屏蔽该通道的传输错误中断
13:11	FlowCtrl	RW	000	流控制器和传输类型。 FlowCtrl      传输方向      控制器 000            存储器到存储器      DMA 001            存储器到外设        DMA 010            外设到存储器        DMA
10:7	RSV	-	-	保留
6:1	SrcPeriph	RW	0x0	源外设 ID。对应关系见功能描述“目标外设和源外设请求号”章节。当源是存储器时，该值无效。
0	EN	RW	0	通道使能 1: 通道使能，传输完成后硬件自动清零。软件写 0 无效。 0: 通道关闭 可以读取 DMACEChStatus 寄存器知道通道使能的情况。

注：DMA 不能访问 eFlash、ROM 空间

## 7.6. 使用说明

### 7.6.1. DMA 优先级

DMA 优先级是固定的，通道 0 最高优先级，通道 7 最低优先级。如果 DMAC 正在为一个低优先级的通道传输数据时，一个高优先级的通道请求产生了，那么 DMAC 将会先把低优先级通道的数据传输完毕，然后再来处理高优先级的通道要求。

### 7.6.2. 软件注意事项

1. 在通道使能后，不能再对通道寄存器有任何写操作；如果一定要写通道寄存器，只能在把使能关闭后再写。
2. 源位宽乘以 TransferSize 必须是目标位宽的整数倍。

3. 如果软件通过设置 DMACCxConfig 寄存器的 bit 0 位来关闭通道，那么必须在读 EnChStat 对应通道 enable 为 0 后才能再次打开通道。因为关闭通道的过程是有延时的。
4. 如果 DMAC 是流控制器，而且 TransferSize 为 0，那么打开 enable 之后不会产生任何数据传输。

### 7.6.3. DMAC 使用流程

1. 打开 DMAC 总开关（DMAC 配置寄存器（config）第 0 位）。
2. 配置好 DMAC 的通道，传输方向，DMA master，源地址和目标地址，源位宽，目标位宽，源/目标的 burst size，传输数据的 transfer size 等传输参数。
3. 打开 DMAC 通道 enable（DMAC 通道配置寄存器（DMACCxConfig）第 0 位）。

### 7.6.4. DMAC 链表使用流程

一个链表项（LLI）由四个字组成，这些字按照以下顺序来排列：

1. REG\_DMACCSrcAddr(x)
2. REG\_DMACCDestAddr(x)
3. REG\_DMACCLinkList(x)
4. REG\_DMACCControl(x)

其中 REG\_DMACCLinkList(x)为下一个链表项的地址，最后一个 LLI 的链表项指针需设置为 0。

实现特点：

将第一个先前写入寄存器的链表项写入 DMA 控制器的相关通道。向通道配置寄存器写入通道配置信息，并置位通道使能位，接下来会启动 DMA 传输，每个数据块传输完毕后，会自动装载下个链表项每个数据块传输完成后都可以产生中断。

## 8. 高级定时器（TIM1）

### 8.1. 概述

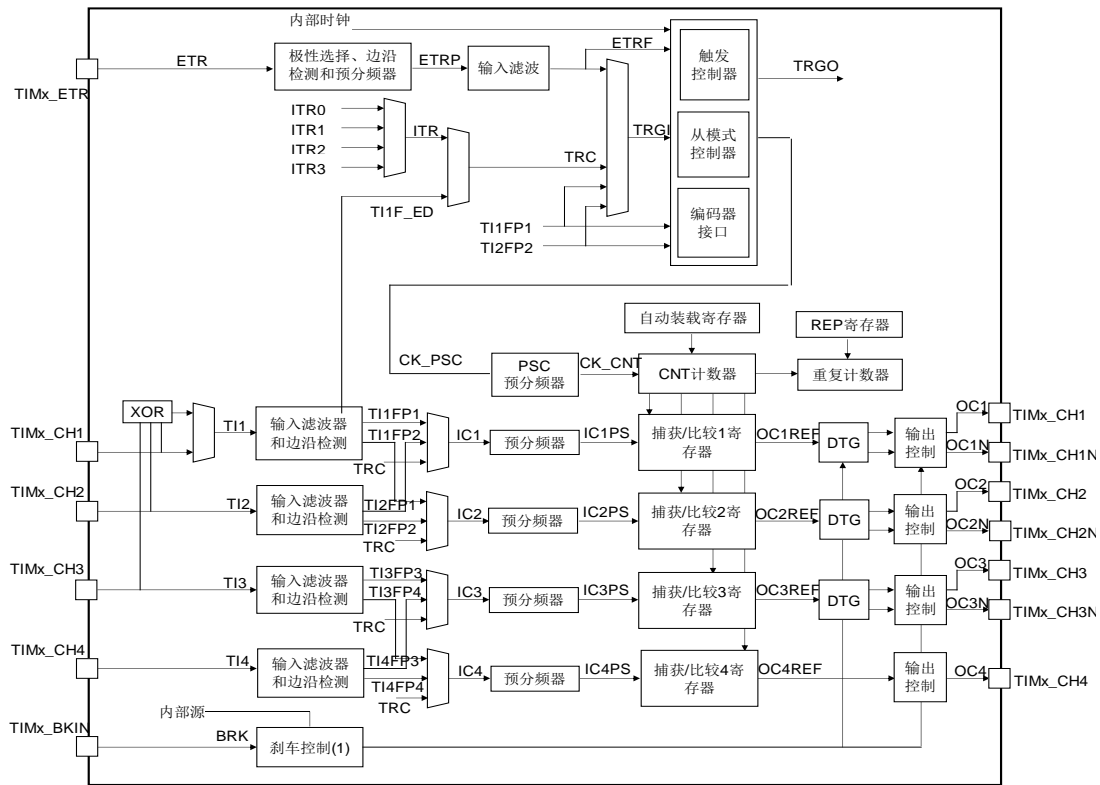
高级控制定时器 TIM1 由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。使用定时器预分频器和系统时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。高级控制定时器和通用定时器是完全独立的，它们不共享任何资源，但它们可以同步操作。

### 8.2. 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 8-1 高级定时器结构框图



注：TIM1 的 TRGO 可作为 DAC/ADC 的外部触发源，详见 ADC/DAC 章节。定时器之间的互联以及刹车输入信号源，详见航芯 ACM32F4 和 F3 芯片定时器差异说明.docx 的定时器互联章节以及本章的刹车功能章节。

## 8.3. 功能描述

### 8.3.1. 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：

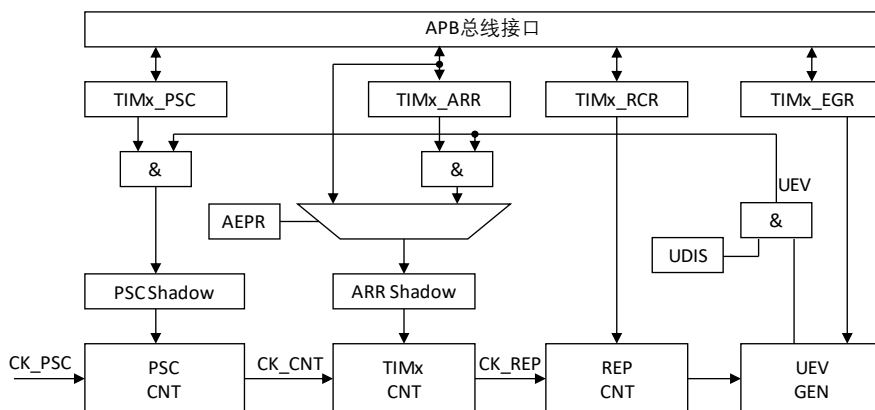
计数器寄存器(TIMx\_CNT)

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 8-2 计数单元的结构



如图 8-2 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。

## 8.3.2. 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。

### 8.3.3. 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。

图 8-3 和图 8-4 给出了在预分频器运行时，更改计数器参数的例子。

图 8-3 当预分频器的参数从 1 变到 2 时，计数器的时序图

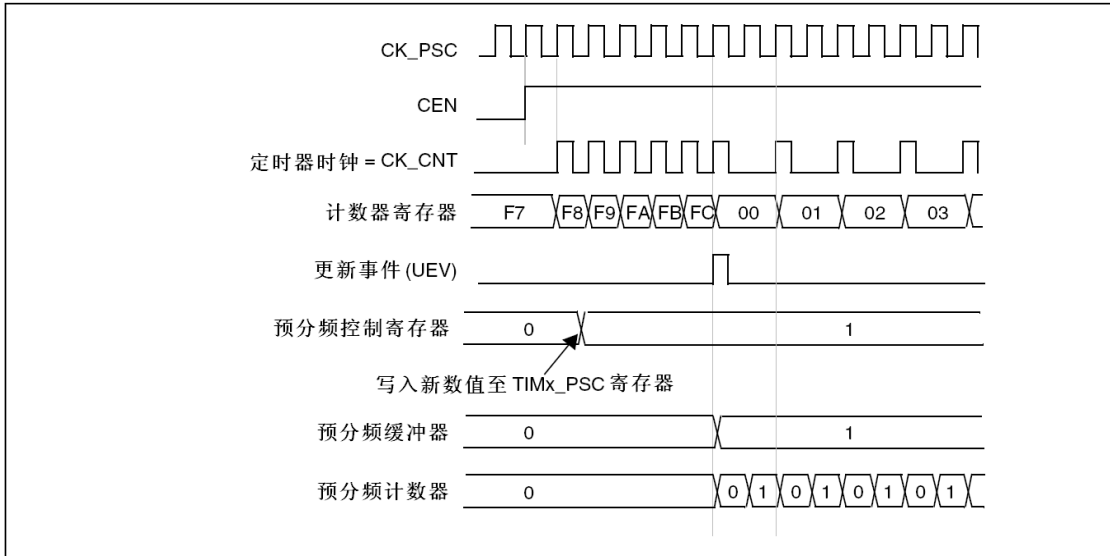
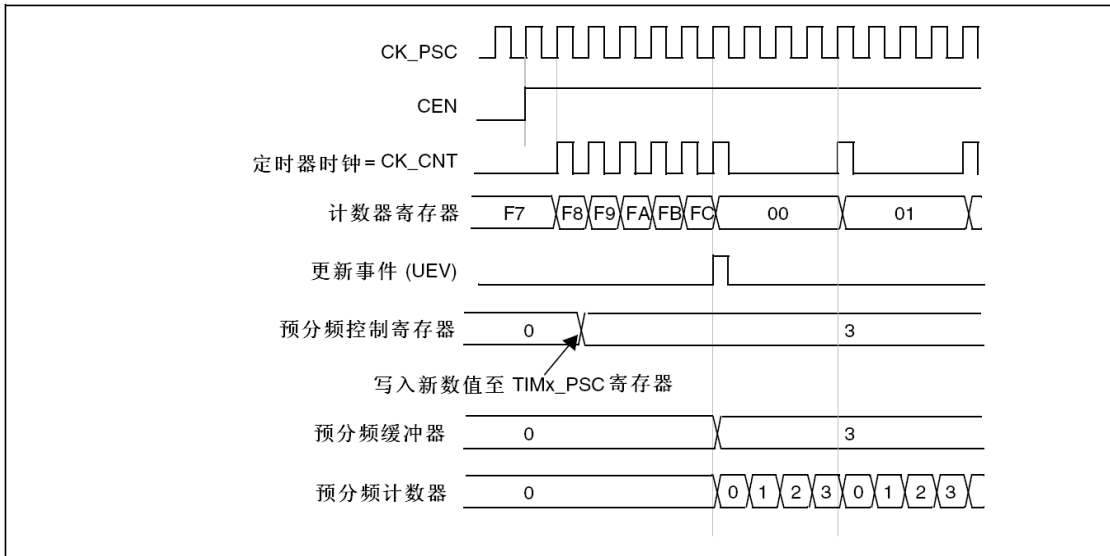


图 8-4 当预分频器的参数从 1 变到 4 时，计数器的时序图



### 8.3.4. 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。

### 8.3.5. 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 8-5 外部时钟模式 1

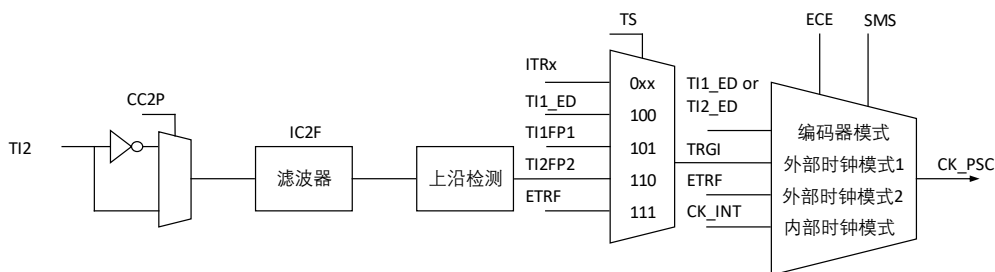
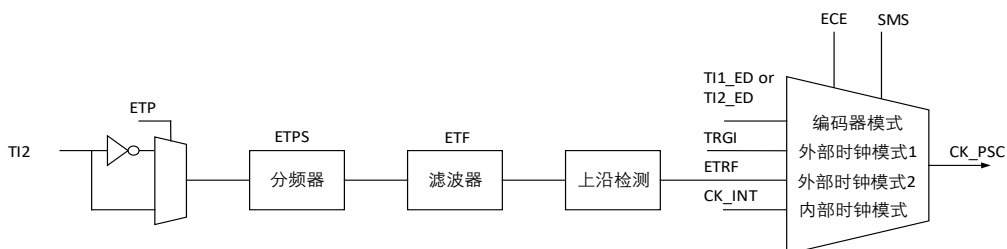


图 8-6 外部时钟模式 2



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 有七个信号源，并由 TIMx\_SMCR.TS 寄存器选择，如图 8-5 所示。

TS=000，内部触发 0 (ITR0)

TS=001，内部触发 1 (ITR1)

TS=010，内部触发 2 (ITR2)

TS=011，内部触发 3 (ITR3)

TS=100, TI1 的边沿检测器 (TI1F\_ED)

TS=101, 滤波后的定时器输入 1 (TI1FP1)

TS=110, 滤波后的定时器输入 2 (TI2FP2)

外部时钟模式 2: ECE=1 或 SMS=111、TS=111, CK\_PSC 来自 ETRF, 如图 8-6 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

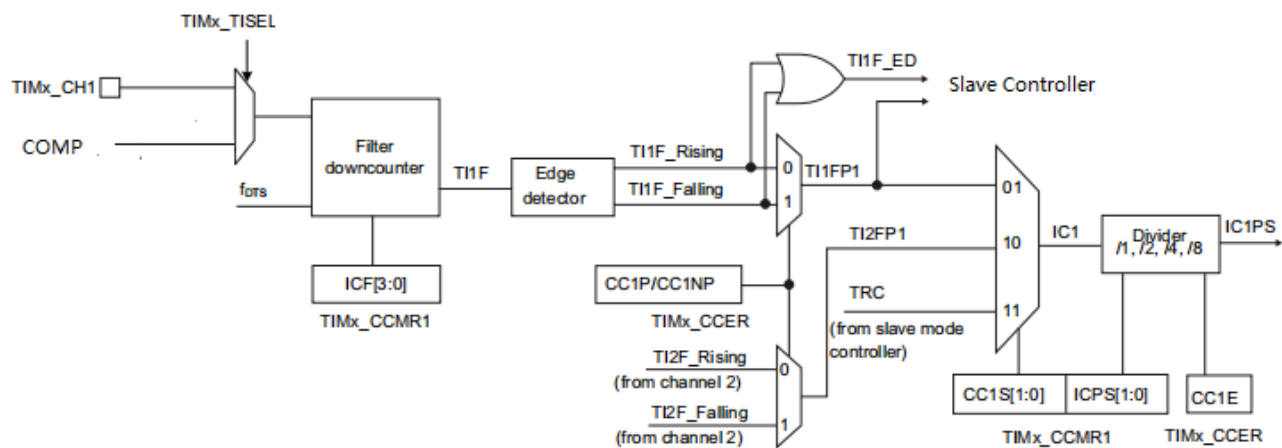
编码器模式 3: SMS=011, ECE=0, CK\_PSC 来自 TI1FP1 和 TI2FP2 的上下沿

### 8.3.6. 捕获比较通道

**输入捕获模式:**

在输入捕获模式下, 当检测到 IC<sub>x</sub> 信号上相应的边沿后, 计数器的当前值被锁存到捕获/比较寄存器(TIM<sub>x</sub>\_CCR<sub>x</sub>)中。当发生捕获事件时, 相应的 CC<sub>x</sub>IF 标志(TIM<sub>x</sub>\_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 CC<sub>x</sub>IF 标志已经为高, 那么重复捕获标志 CC<sub>x</sub>OF(TIM<sub>x</sub>\_SR 寄存器)被置 1。写 CC<sub>x</sub>IF=0 可清除 CC<sub>x</sub>IF, 或读取存储在 TIM<sub>x</sub>\_CCR<sub>x</sub> 寄存器中的捕获数据也可清除 CC<sub>x</sub>IF。写 CC<sub>x</sub>OF=0 可清除 CC<sub>x</sub>OF。

图 8-7 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 TIM<sub>x</sub>\_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 TIM<sub>x</sub>\_CCR1, 因此写入 TIM<sub>x</sub>\_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 fDTS 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。

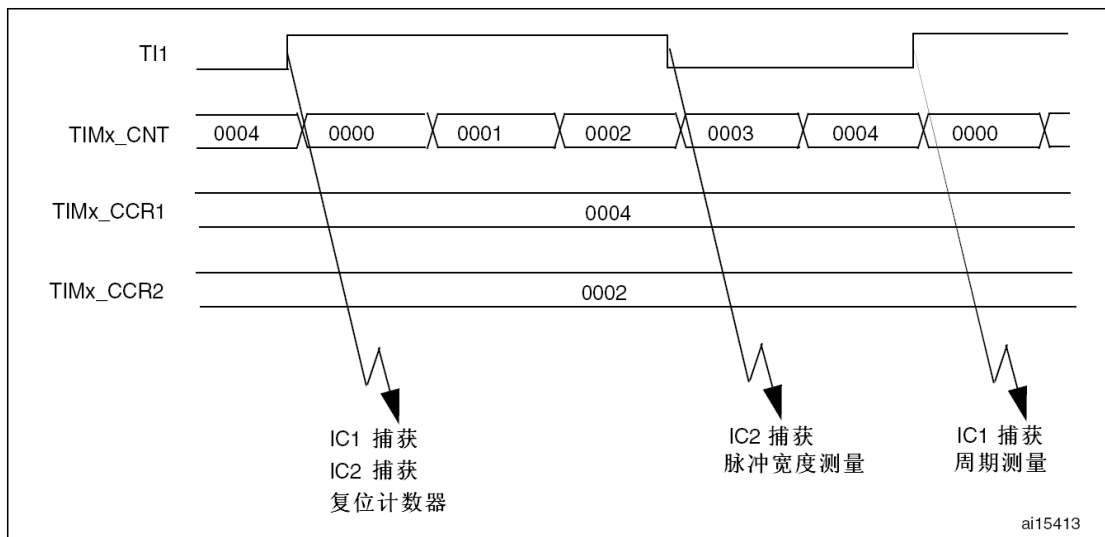
这 2 个 ICx 信号为边沿有效，但是极性相反。

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 8-8 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。

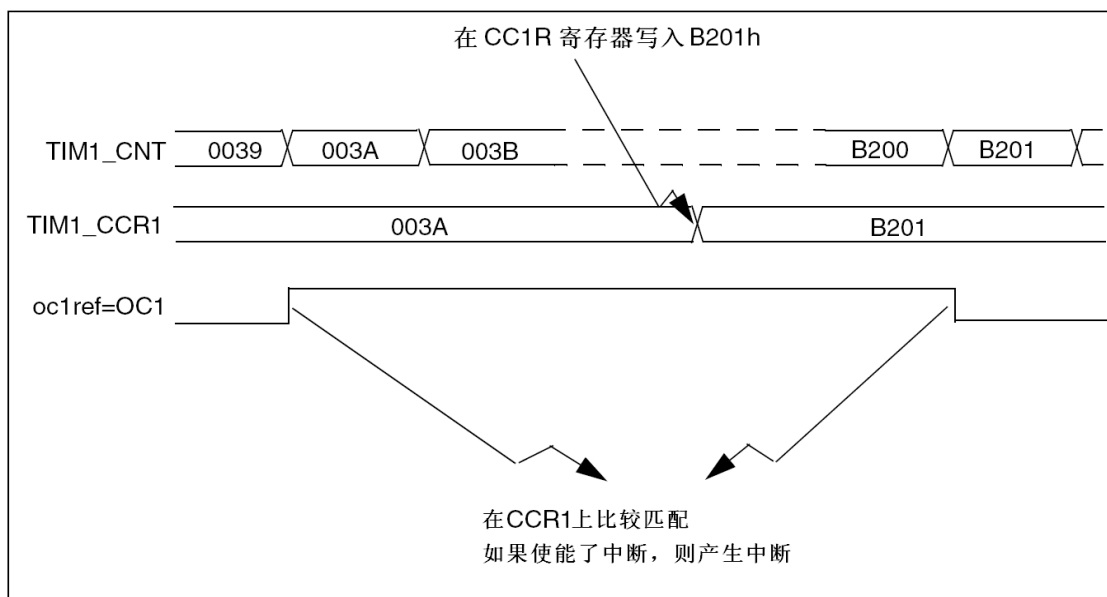
同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

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)。图 8-9 是输出比较模式的一个例子。

图 8-9 输出比较模式, 翻转 OC1



### 8.3.7. 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。

### 8.3.8. 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。

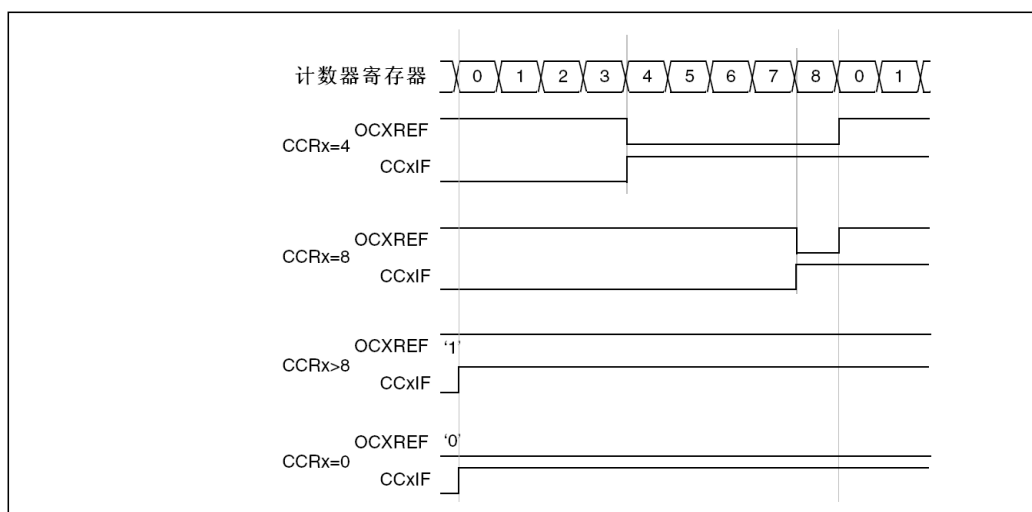
在 PWM 模式(模式 1 或模式 2)下, TIMx\_CNT 和 TIMx\_CCRx 始终在进行比较, (依据计数器的计数方向)以确定是否符合  $TIMx\_CCRx \leq TIMx\_CNT$  或者  $TIMx\_CNT \leq TIMx\_CCRx$ 。根据 TIMx\_CR1 寄存器中 CMS 位的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

## 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。图 8-10 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装载值(TIMx\_ARR)，则 OCxREF 保持为'1'。如果比较值为 0，则 OCxREF 保持为'0'。

图 8-10 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。

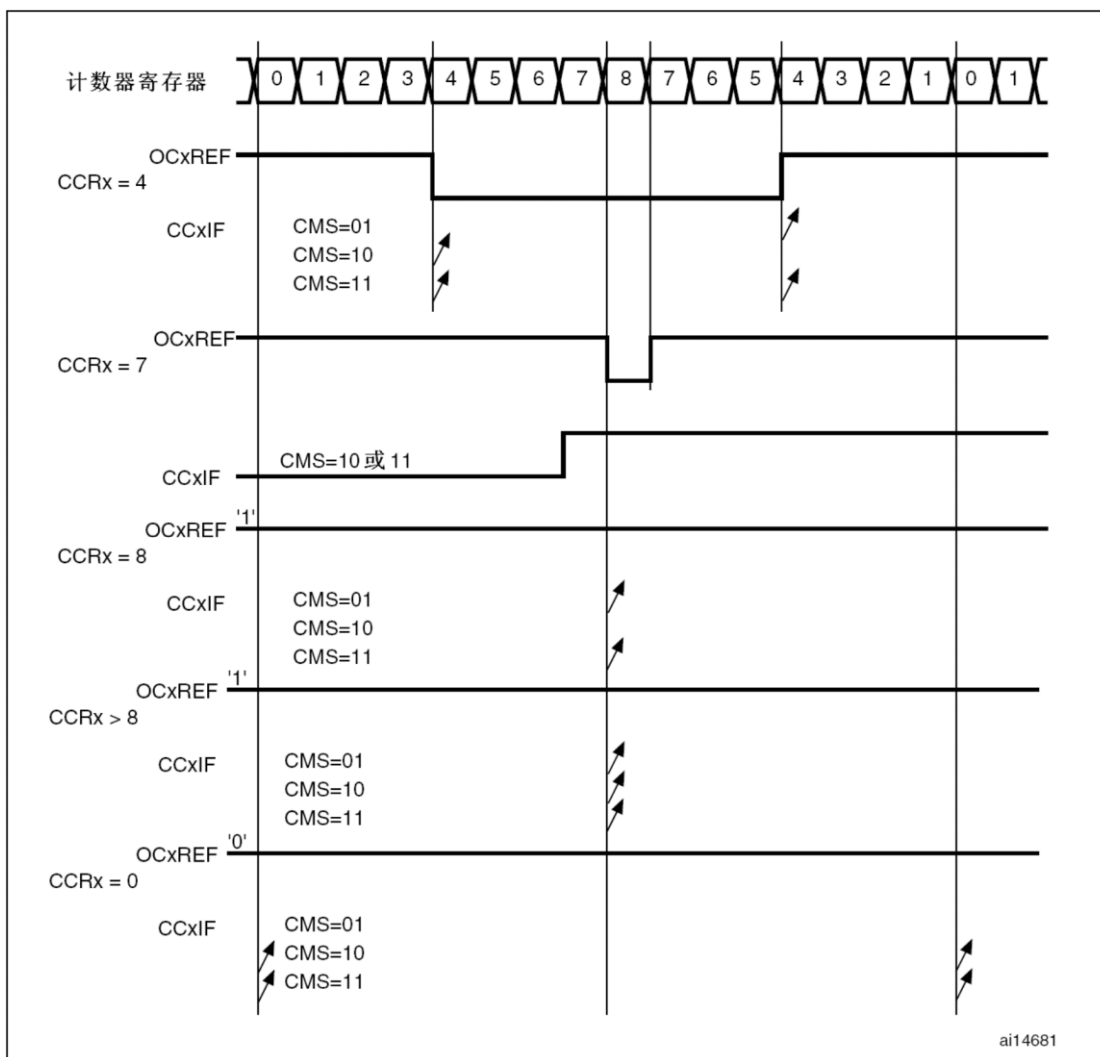
图 8-11 给出了一些中央对齐的 PWM 波形的例子

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 8-11 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。

### 8.3.9. 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，详见表格 8-4。带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是，在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

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

**如果 OCx 和 OCxN 为高有效：**

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

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

图 8-12 带死区插入的互补输出

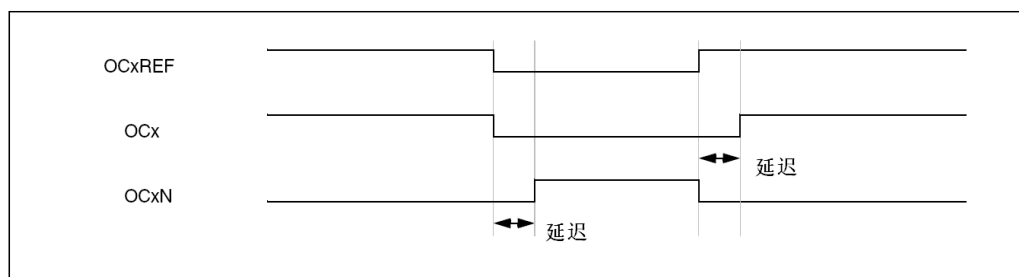


图 8-13 死区波形延迟大于负脉冲

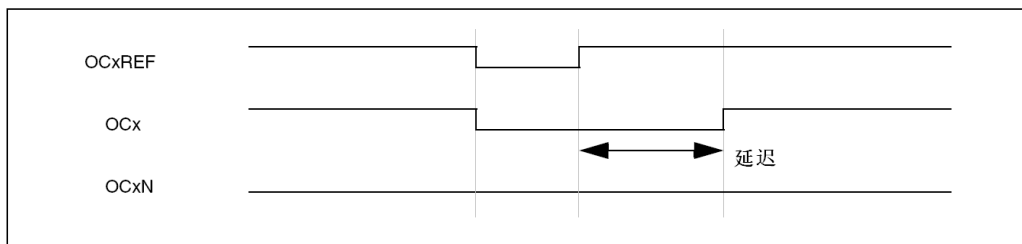
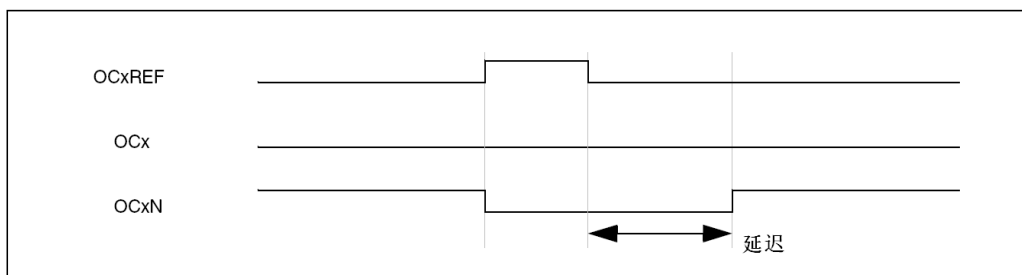


图 8-14 死区波形延迟大于正脉冲



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。

### 8.3.10. 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。

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 OSSI 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 OSSI=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 OSSI=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。

详情请参考表格 8-2。

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。图 8-15 显示响应刹车的输出实例。图 8-16 显示了刹车输入信号以及极性选择。

图 8-15 响应刹车的输出

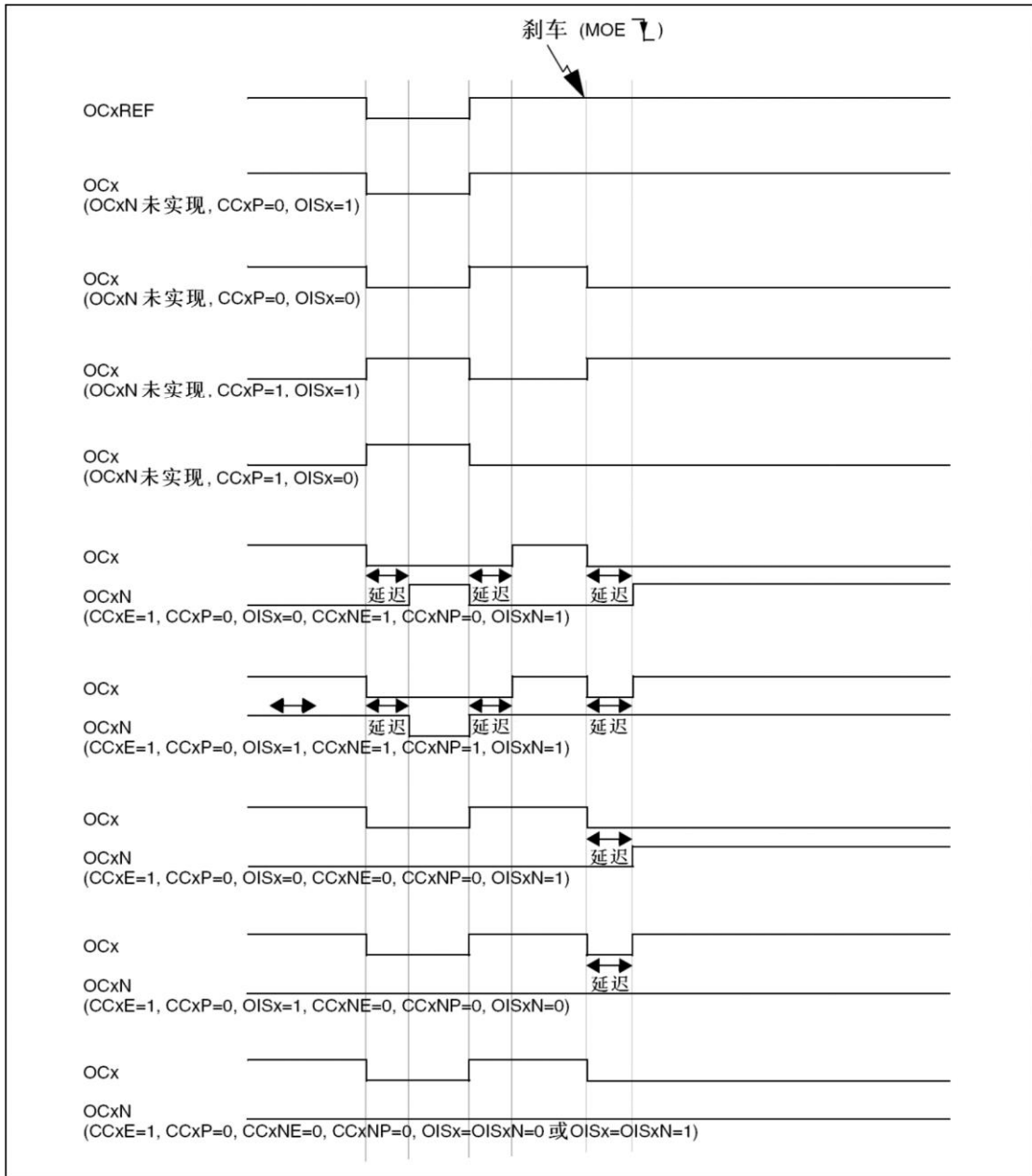
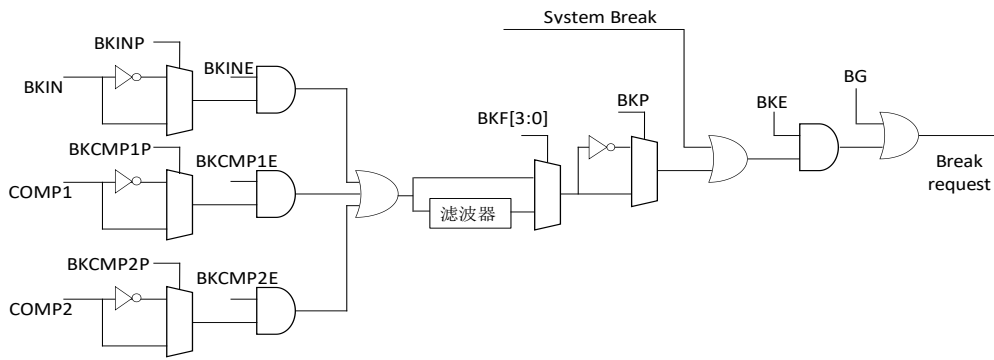


图 8-16 刹车控制



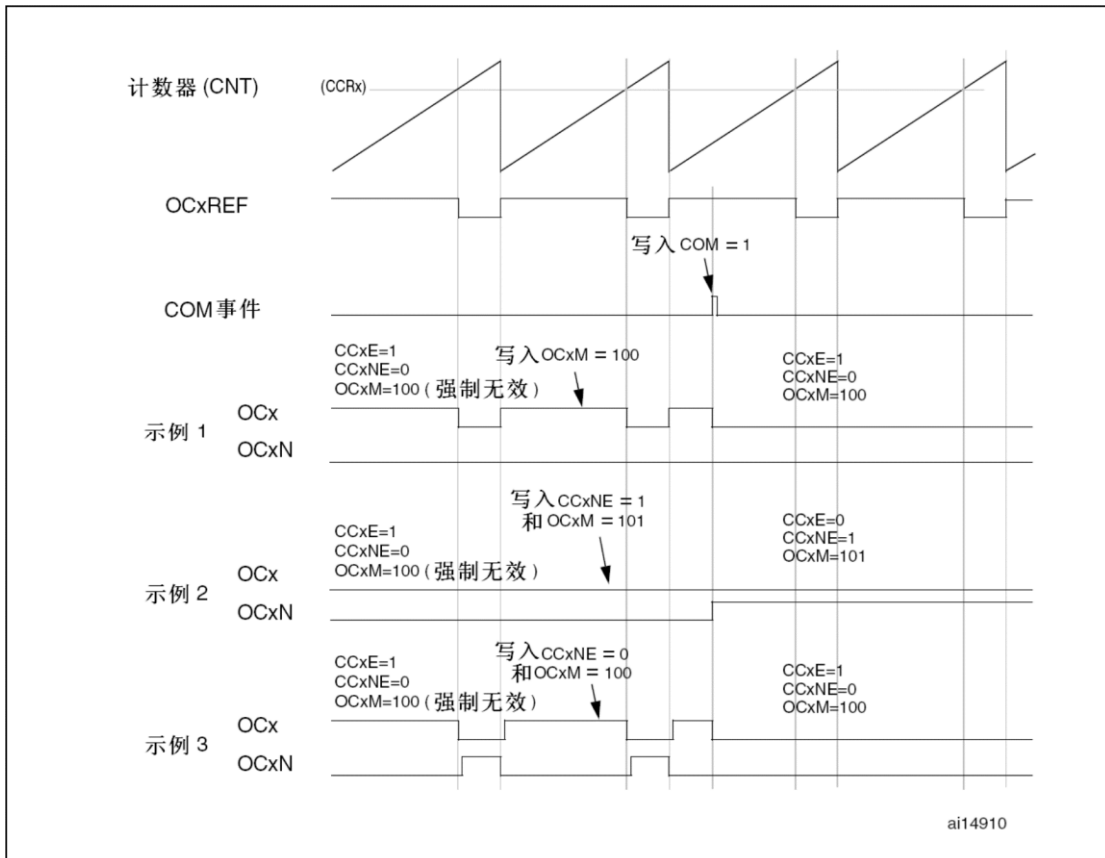
### 8.3.11. 六步 PWM 输出

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 8-17 产生六步 PWM，使用 COM 的例子(OSSR=1)



### 8.3.12. 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。表格 8-1 列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表格 8-1 计数器方向和编码器信号的关系

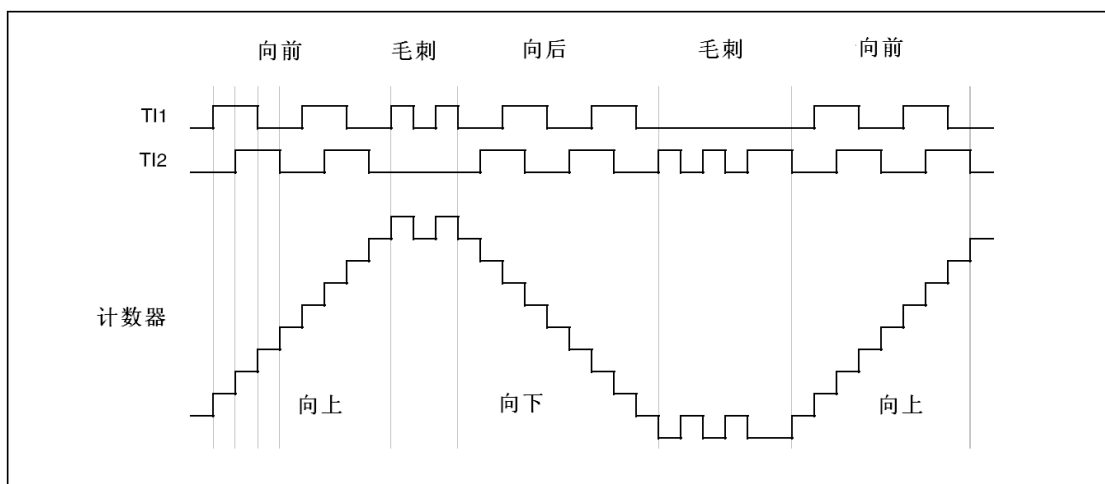
有效边沿	相对信号的电平 (TI1FP1 对应 TI2, TI2FP2 对 应 TI1)	TI1FP1		TI2FP2	
		上升	下降	上升	下降
仅在 TI1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 TI2 计数	高	不计数	不计数	向上计数	向下计数

	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分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。**错误!未找到引用源。**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 8-18 编码器模式下计数器操作实例



### 8.3.13. 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，见图 8-19，3 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，计数器从新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较通道 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。

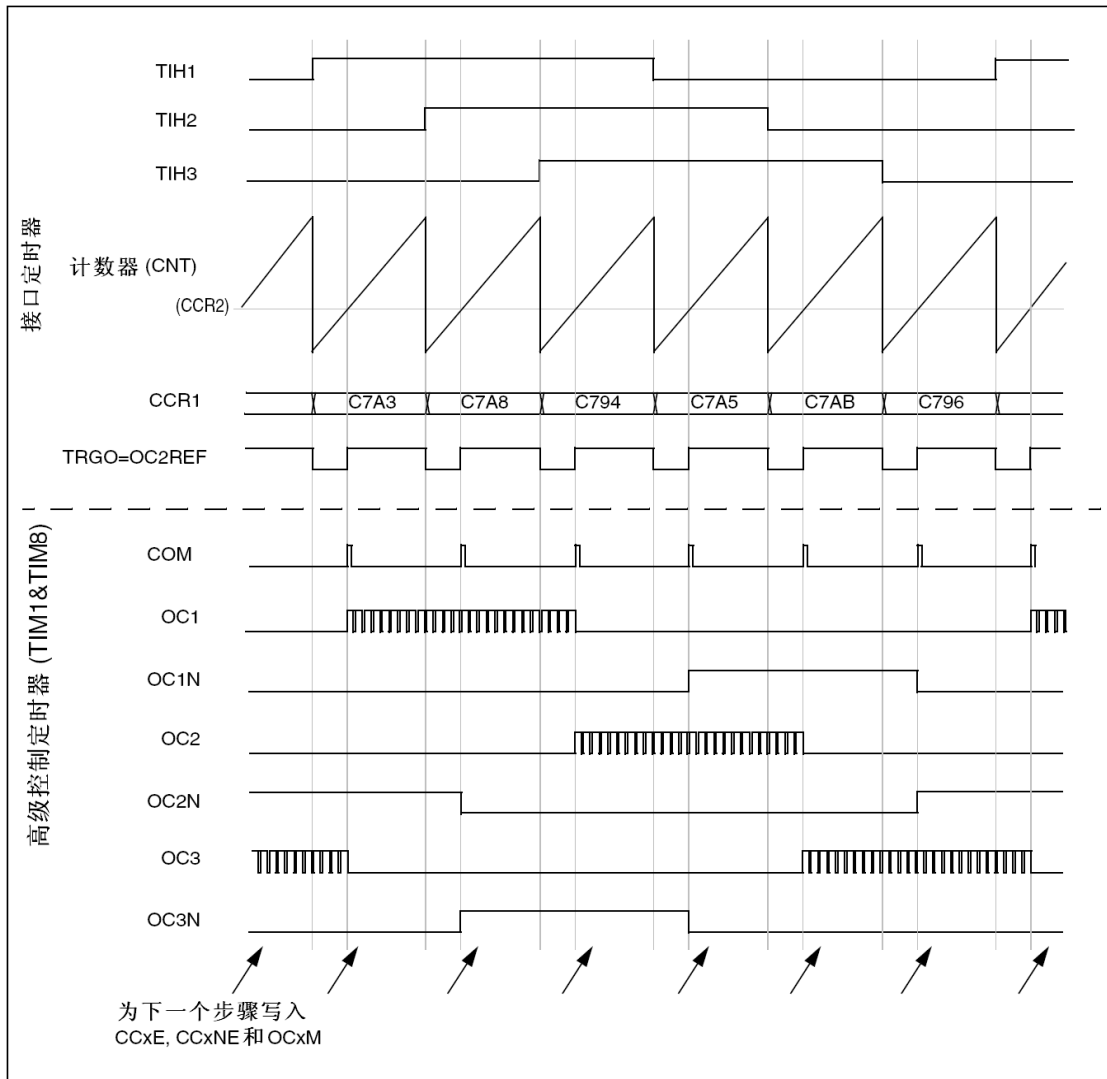
设置通道 1 为捕获模式(选中 TRC)：置 TIMx\_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。

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 TIM1 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 8-19 霍尔传感器接口的实例



### 8.3.14. 单脉冲模式

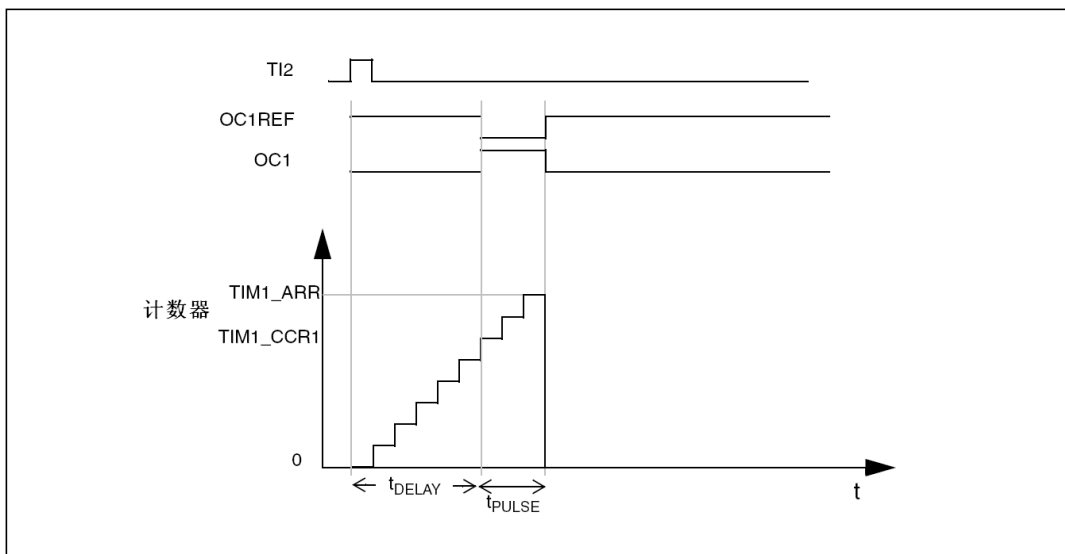
单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx\_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以使计数器自动地在产生下一个更新事件 UEV 时停止。

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 8-20 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  $t_{\text{DELAY}}$  之后，在 OC1 上产生一个长度为  $t_{\text{PULSE}}$  的正脉冲。假定 TI2FP2 作为触发 1:

置 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)

$t_{\text{DELAY}}$  由 TIMx\_CCR1 寄存器中的值定义。

$t_{\text{PULSE}}$  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。

### 8.3.15. 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：

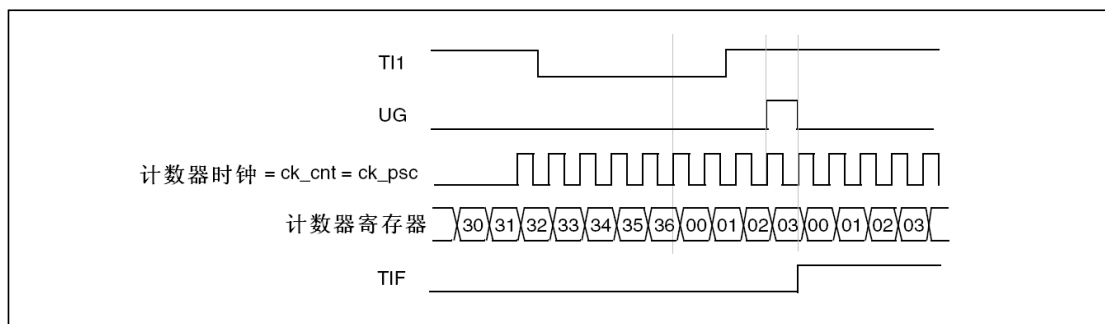
配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx\_CCMR1 寄存器中 CC1S=01。置 TIMx\_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。

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 8-21 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)。

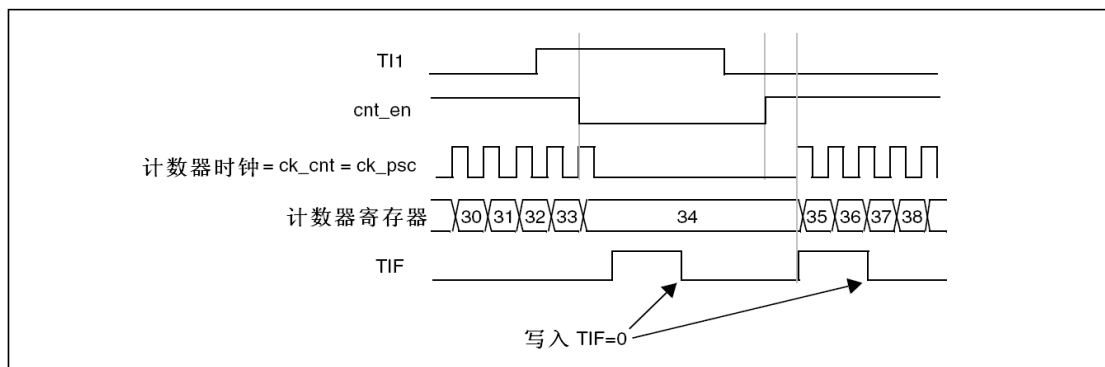
置 TIMx\_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx\_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

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 标志。

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 8-22 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：

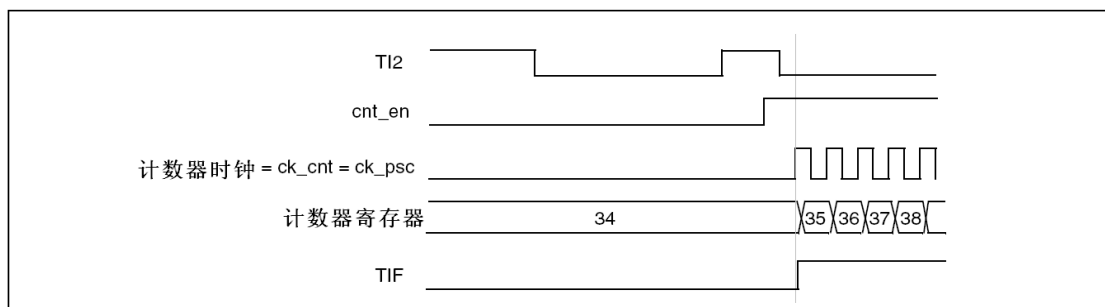
配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx\_CCMR1 寄存器中 CC2S=01。置 TIMx\_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。

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 8-23 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 寄存器配置外部触发输入电路: —ETF=0000: 没有滤波 —ETPS=00: 不用预分频器 —ETP=0: 检测 ETR 的上升沿, 置 ECE=1 使能外部时钟模式 2。

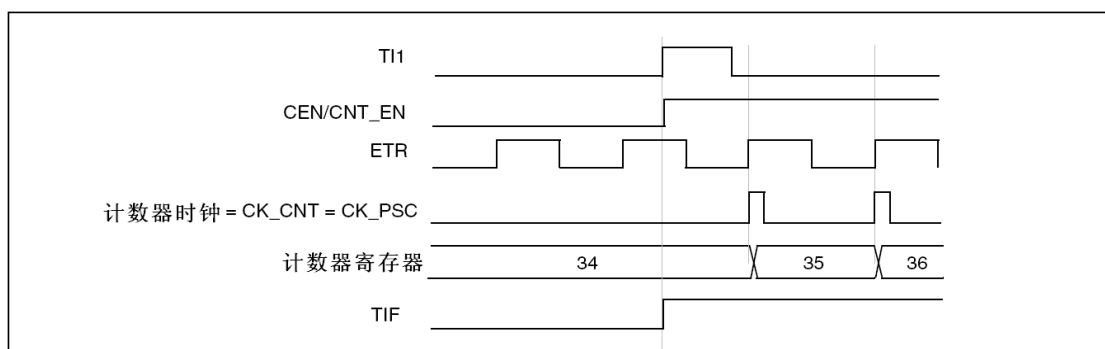
按如下配置通道 1, 检测 TI 的上升沿: —IC1F=0000: 没有滤波 — 触发操作中不使用捕获预分频器, 不需要配置 — 置 TIMx\_CCMR1 寄存器中 CC1S=01, 选择输入捕获源 — 置 TIMx\_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)

置 TIMx\_SMCR 寄存器中 SMS=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复位间的延时, 取决于 ETRP 输入端的重同步电路。

图 8-24 外部时钟模式 2+触发模式控制电路



### 8.3.16. 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。有两个跟定时器 DMA 模式相关的寄存器：TIMx\_DCR 和 TIMx\_DMAR。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMx 会给 DMA 发送请求。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，这个内部寄存器由 TIMx\_DCR 寄存器中的 DBL 来指定。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。

## 8.4. 寄存器描述

TIM1 寄存器基地址：0x40012c00

偏置	名称	描述
0x00	TIM1_CR1	TIM1 控制寄存器 1
0x04	TIM1_CR2	TIM1 控制寄存器 2
0x08	TIM1_SMCR	TIM1 从模式控制寄存器
0x0C	TIM1_DIER	TIM1 DMA/中断使能寄存器
0x10	TIM1_SR	TIM1 状态寄存器
0x14	TIM1_EGR	TIM1 事件产生寄存器
0x18	TIM1_CCMR1	TIM1 捕获/比较模式寄存器 1
0x1C	TIM1_CCMR2	TIM1 捕获/比较模式寄存器 2
0x20	TIM1_CCER	TIM1 捕获/比较使能寄存器
0x24	TIM1_CNT	TIM1 计数器
0x28	TIM1_PSC	TIM1 预分频器
0x2C	TIM1_ARR	TIM1 自动装载寄存器
0x30	TIM1_RCR	TIM1 重复计数寄存器
0x34	TIM1_CCR1	TIM1 捕获比较寄存器 1
0x38	TIM1_CCR2	TIM1 捕获比较寄存器 2
0x3C	TIM1_CCR3	TIM1 捕获比较寄存器 3
0x40	TIM1_CCR4	TIM1 捕获比较寄存器 4
0x44	TIM1_BDTR	TIM1 刹车和死区控制寄存器
0x48	TIM1_DCR	TIM1 DMA 控制寄存器
0x4C	TIM1_DMAR	TIM1 连续模式的 DMA 地址
0x60	TIM1_AF1	TIM1 复用功能选择寄存器
0x68	TIM1_TISEL	TIM1 输入选择寄存器
0x6C	TIM1_DBER	TIM1 DMA 请求类型选择寄存器

## 1.1.1. TIM1 控制寄存器 1 (TIM1\_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	0000	<p>刹车滤波器 (Break filter)</p> <p>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:</p> <p>0000: 无滤波器, 以 fDTS 采样</p> <p>100:: 0: 采样频率 fSAMPLING=fDTS/8, N=6</p> <p>0001: 采样频率 fSAMPLING=fCK_INT, N=2</p> <p>1001: 采样频率 fSAMPLING=fDTS/8, N=8</p> <p>0010: 采样频率 fSAMPLING=fCK_INT, N=4</p> <p>1010: 采样频率 fSAMPLING=fDTS/16, N=5</p> <p>0011: 采样频率 fSAMPLING=fCK_INT, N=8</p> <p>1011: 采样频率 fSAMPLING=fDTS/16, N=6</p> <p>0100: 采样频率 fSAMPLING=fDTS/2, N=6</p> <p>1100: 采样频率 fSAMPLING=fDTS/16, N=8</p> <p>0101: 采样频率 fSAMPLING=fDTS/2, N=8</p> <p>1101: 采样频率 fSAMPLING=fDTS/32, N=5</p> <p>0110: 采样频率 fSAMPLING=fDTS/4, N=6</p> <p>1110: 采样频率 fSAMPLING=fDTS/32, N=6</p> <p>0111: 采样频率 fSAMPLING=fDTS/4, N=8</p> <p>1111: 采样频率 fSAMPLING=fDTS/32, N=8</p>
9:8	CKD	RW	00	<p>时钟分频因子</p> <p>死区发生器和数字滤波器所用的采样时钟与定时器时钟 (CK_INT) 的分频比例。</p> <p>00: tDTS=tCK_INT</p> <p>01: tDTS=2 x tCK_INT</p> <p>10: tDTS=4 x tCK_INT</p> <p>11:保留</p>
7	ARPE	RW	0	<p>自动重装载预装载允许位</p> <p>0:TIM1_ARR 寄存器没有缓冲</p> <p>1:TIM1_ARR 寄存器被装入缓冲器</p>
6:5	CMS	RW	00	计数模式

				<p>00:边沿对齐模式，计数器根据方向位（DIR）向上或向下计数。</p> <p>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(TIM1_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。</p> <p>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(TIM1_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。</p> <p>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(TIM1_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。</p> <p>注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。</p>
4	DIR	RW	0	<p>方向控制位</p> <p>0: 计数器向上计数；</p> <p>1: 计数器向下计数。</p> <p>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。</p>
3	OPM	RW	0	<p>单脉冲模式</p> <p>0: 在发生更新事件时，计数器不停止；</p> <p>1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止。</p>
2	URS	RW	0	<p>更新请求源</p> <p>软件通过该位选择 UEV 事件的源</p> <p>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：</p> <ul style="list-style-type: none"> <li>- 计数器溢出/下溢</li> <li>- 设置 UG 位</li> <li>- 从模式控制器产生的更新</li> </ul> <p>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。</p>
1	UDIS	RW	0	禁止更新

				<p>软件通过该位允许/禁止 UEV 事件的产生</p> <p>0: 允许 UEV。更新(UEV)事件由下述任一事件产生:</p> <ul style="list-style-type: none"> <li>- 计数器溢出/下溢</li> <li>- 设置 UG 位</li> <li>- 从模式控制器产生的更新</li> </ul> <p>具有缓存的寄存器被装入它们的预装载值。(译注:更新影子寄存器)</p> <p>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。</p>
0	CEN	RW	0	<p>使能计数器</p> <p>0: 禁止计数器;</p> <p>1: 使能计数器。</p> <p>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。</p>

### 1.1.2. TIM1 控制寄存器 2 (TIM1\_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
15	-	-	-	保留, 始终读为 0。
14	OIS4	RW	0	输出空闲状态 4(OC4 输出)。参见 OIS1 位。
13	OIS3N	RW	0	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。
12	OIS3	RW	0	输出空闲状态 3(OC3 输出)。参见 OIS1 位。
11	OIS2N	RW	0	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。
10	OIS2	RW	0	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	RW	0	<p>输出空闲状态 1(OC1N 输出) (Output Idle state 1)</p> <p>0: 当 MOE=0 时, 死区后 OC1N=0;</p> <p>1: 当 MOE=0 时, 死区后 OC1N=1。</p> <p>注: 已经设置了 LOCK(TIM1_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。</p>
8	OIS1	RW	0	<p>输出空闲状态 1(OC1 输出) (Output Idle state 1)</p> <p>0: 当 MOE=0 时, 如果实现了 OC1N, 则死区后</p>

				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(TIM1_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
7	TIIS	RW	0	<p>TI1 选择 (TI1 selection)</p> <p>0: TIM1_CH1 引脚连到 TI1 输入;</p> <p>1: TIM1_CH1、TIM1_CH2 和 TIM1_CH3 引脚经异或后连到 TI1 输入。</p>
6:4	MMS	RW	000	<p>主模式选择 (Master mode selection)</p> <p>这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下:</p> <p>000: 复位 – TIM1_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。</p> <p>001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式 (见 TIM1_SMCR 寄存器中 MSM 位的描述)。</p> <p>010: 更新 – 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。</p> <p>011: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。</p> <p>100: 比较 – OC1REF 信号被用于作为触发输出 (TRGO)。</p> <p>101: 比较 – OC2REF 信号被用于作为触发输出 (TRGO)。</p> <p>110: 比较 – OC3REF 信号被用于作为触发输出 (TRGO)。</p> <p>111: 比较 – OC4REF 信号被用于作为触发输出 (TRGO)。</p>

3	CCDS	RW	0	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	0	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	0	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。

### 1.1.3. TIM1 从模式控制寄存器 (TIM1\_SMCR 偏移: 0x08)

比特	名称	属性	复位值	描述
15	ETP	RW	0	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	0	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	0	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 TIM1CLK 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	0000	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	0	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	RW	000	<p>触发选择 (Trigger selection)</p> <p>这 3 位选择用于同步计数器的触发输入。</p> <p>000: 内部触发 0(ITR0)</p> <p>100: TI1 的边沿检测器(TI1F_ED)</p> <p>001: 内部触发 1(ITR1)</p> <p>101: 滤波后的定时器输入 1(TI1FP1)</p> <p>010: 内部触发 2(ITR2)</p> <p>110: 滤波后的定时器输入 2(TI2FP2)</p> <p>011: 内部触发 3(ITR3)</p> <p>111: 外部触发输入(ETRF)</p> <p>注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。</p>
3	-	-	-	保留, 始终读为 0。
2:0	SMS	RW	000	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明)</p> <p>000: 关闭从模式 – 如果 CEN=1, 则预分频器直接由内部时钟驱动。</p> <p>001: 编码器模式 1 – 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数。</p> <p>010: 编码器模式 2 – 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。</p> <p>011: 编码器模式 3 – 根据另一个信号的输入电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。</p> <p>100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。</p> <p>101: 门控模式 – 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。</p> <p>110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。</p> <p>111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上</p>

				<p>升沿驱动计数器。</p> <p>注：如果 TI1F_EN 被选为触发输入(TS=100)时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。</p>
--	--	--	--	--

表格 8-2 TIM1 从模式互连-ITRx

ITR0	TIM4_TRGO
ITR1	TIM2_TRGO
ITR2	TIM3_TRGO
ITR3	TIM17_OC1

表格 8-3 TIM1 从模式互连-ETRx

ETR 输入	
ETRSEL =00	PA12
ETRSEL =01	COMP1
ETRSEL =02	COMP2
ETRSEL =03	AWD

#### 1.1.4. TIM1 DMA/中断使能寄存器 (TIM1\_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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断; 1: 允许 COM 中断。
4	CC4IE	RW	0	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	0	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	0	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	0	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	0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

### 1.1.5. TIM1 状态寄存器 (TIM1\_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
15:13	-	-	-	保留, 始终读为 0。
12	CC4OF	RC_W0	0	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	RC_W0	0	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	RC_W0	0	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	RC_W0	0	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 TIM1_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	RC_W0	0	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	RC_W0	0	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	RC_W0	0	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	RC_W0	0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	RC_W0	0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	RC_W0	0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	RC_W0	0	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 TIM1_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生; 1: TIM1_CNT 的值与 TIM1_CCR1 的值匹配。 当 TIM1_CCR1 的内容大于 TIM1_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 TIM1_CCR1 清'0'。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIM1_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	RC_W0	0	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':

				<ul style="list-style-type: none"> <li>- 若 TIM1_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)。</li> <li>- 若 TIM1_CR1 寄存器的 URS=0、UDIS=0，当设置 TIM1_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。</li> <li>- 若 TIM1_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。</li> </ul>
--	--	--	--	---

### 1.1.6. TIM1 事件产生寄存器 (TIM1\_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	0	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	0	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: TIM1_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	0	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	0	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	WO	0	产生捕获/比较 3 事件 (Capture/Compare 3 generation)

				参考 CC1G 描述。
2	CC2G	WO	0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0	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 TIM1_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	0	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 TIM1_ARR 的值。

### 1.1.7. TIM1 捕获/比较模式寄存器 1 (TIM1\_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	0	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	000	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0	输出比较 2 预装载使能 (Output Compare 2 preload enable)

10	OC2FE	RW	0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	00	<p>捕获/比较 2 选择。(Capture/Compare 2 selection)</p> <p>该位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC2 通道被配置为输出;</p> <p>01: CC2 通道被配置为输入, IC2 映射在 TI2 上;</p> <p>10: CC2 通道被配置为输入, IC2 映射在 TI1 上;</p> <p>11: CC2 通道被配置为输入, IC2 映射在 TRC 上。</p> <p>此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC2S 仅在通道关闭时(TIM1_CCER 寄存器的 CC2E=0)才是可写的。</p>
7	OC1CE	RW	0	<p>输出比较 1 清'0'使能 (Output Compare 1 clear enable)</p> <p>0: OC1REF 不受 ETRF 输入的影响;</p> <p>1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。</p>
6:4	OC1M	RW	000	<p>输出比较 1 模式 (Output Compare 1 mode)</p> <p>该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。</p> <p>000: 冻结。输出比较寄存器 TIM1_CCR1 与计数器 TIM1_CNT 间的比较对 OC1REF 不起作用;</p> <p>001: 匹配时设置通道 1 为有效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1 (TIM1_CCR1) 相同时, 强制 OC1REF 为高。</p> <p>010: 匹配时设置通道 1 为无效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1 (TIM1_CCR1) 相同时, 强制 OC1REF 为低。</p> <p>011: 翻转。当 TIM1_CCR1=TIM1_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为无效电平。强制 OC1REF 为低。</p> <p>101: 强制为有效电平。强制 OC1REF 为高。</p> <p>110: PWM 模式 1— 在向上计数时, 一旦 TIM1_CNT&lt;TIM1_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</p>

				<p>TIM1_CNT&gt;TIM1_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM 模式 2— 在向上计数时, 一旦 TIM1_CNT&lt;TIM1_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 TIM1_CNT&gt;TIM1_CCR1 时通道 1 为有效电平, 否则为无效电平。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIM1_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。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
3	OC1PE	RW	0	<p>输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>0: 禁止 TIM1_CCR1 寄存器的预装载功能, 可随时写入 TIM1_CCR1 寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启 TIM1_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIM1_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。</p> <p>注 2: 仅在单脉冲模式下(TIM1_CR1 寄存器的 OPM=1), 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定。</p>
2	OC1FE	RW	0	<p>输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被</p>

				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	00	<p>捕获/比较 1 选择。(Capture/Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: CC1 通道被配置为输入, IC1 映射在 TRC 上。</p> <p>此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC1S 仅在通道关闭时(TIM1_CCER 寄存器的 CC1E=0)才是可写的。</p>

## 输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC2F	RW	000	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	00	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	00	<p>捕获/比较 2 选择 (Capture/Compare 2 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC2 通道被配置为输出;</p> <p>01: CC2 通道被配置为输入, IC2 映射在 TI2 上;</p> <p>10: CC2 通道被配置为输入, IC2 映射在 TI1 上;</p> <p>11: CC2 通道被配置为输入, IC2 映射在 TRC 上。</p> <p>此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC2S 仅在通道关闭时(TIM1_CCER 寄存器的 CC2E=0)才是可写的。</p>
7:4	IC1F	RW	0000	<p>输入捕获 1 滤波器 (Input capture 1 filter)</p> <p>这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变:</p> <p>0000: 无滤波器, 以 fDTS 采样</p> <p>1000: 采样频率 fSAMPLING=fDTS/8, N=6</p> <p>0001: 采样频率 fSAMPLING=fCK_INT, N=2</p>

				<p>1001: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, <math>N=8</math></p> <p>0010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, <math>N=4</math></p> <p>1010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=5</math></p> <p>0011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, <math>N=8</math></p> <p>1011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=6</math></p> <p>0100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, <math>N=6</math></p> <p>1100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=8</math></p> <p>0101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, <math>N=8</math></p> <p>1101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=5</math></p> <p>0110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, <math>N=6</math></p> <p>1110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=6</math></p> <p>0111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, <math>N=8</math></p> <p>1111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=8</math></p>
3:2	IC1PSC	RW	00	<p>输入/捕获 1 预分频器 (Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 <math>\text{CC1E}=0</math>(TIM1_CCER 寄存器中), 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每 2 个事件触发一次捕获;</p> <p>10: 每 4 个事件触发一次捕获;</p> <p>11: 每 8 个事件触发一次捕获。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择 (Capture/Compare 1 Selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: CC1 通道被配置为输入, IC1 映射在 TRC 上。</p> <p>此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。</p>

### 1.1.8. TIM1 捕获/比较模式寄存器 2 (TIM1\_CCMR2 偏移: 0x1C)

输出比较模式:

比特	名称	属性	复位值	描述
----	----	----	-----	----

15	OC4CE	RW	0	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M	RW	000	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	RW	0	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	RW	0	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S	RW	00	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 TIM1_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM1_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	RW	0	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M	RW	000	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	RW	0	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	RW	0	输出比较 3 快速使能 (Output compare 3 fast enable)
1:0	CC3S	RW	00	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 TIM1_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM1_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	000	输入捕获 4 滤波器 (Input capture 4 filter)

11:10	IC4PSC	RW	00	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	00	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 TIM1_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM1_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	RW	0000	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	00	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	00	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 TIM1_SMCR 寄存器的 TS 位选择)。

### 1.1.9. TIM1 捕获/比较使能寄存器 (TIM1\_CCER 偏移: 0x20)

比特	名称	属性	复位值	描述
15	CC4NP	RW	0	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	0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0	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	0	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	0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0	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	0	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	0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0	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 (TIM1_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	0	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	0	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

				<p>1: OC1 低电平有效</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>10: 保留, 不使用此配置。</p> <p>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)</p> <p>注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 3 或 2, 则该位不能被修改。</p>
0	CC1E	RW	0	<p>输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出:</p> <p>0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>1: 开启— OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>CC1 通道配置为输入: 该位决定了计数器的值是否能捕获入 TIM1_CCR1 寄存器。</p> <p>0: 捕获禁止</p> <p>1: 捕获使能</p>

表格 8-4 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)	输出禁止 (与定时器断开)

					开) OCx=0, OCx_EN=0	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 ⊕ CCxNP OCxN_EN=1
		0	1	0	OCxREF + 极性, OCx=OCxREF ⊕ 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 ⊕ CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性, OCx=OCxREF ⊕ 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)	
	0		1	0	异步地: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0,	
	0		1	1	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
	1		0	0	输出禁止（与定时器断开） OCx=CCxP, OCx_EN=0
	1		0	1	关闭状态（输出使能且为无效电平）
	1		1	0	异步地：OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1,
	1		1	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。

### 1.1.10. TIM1 计数器（TIM1\_CNT 偏移：0x24）

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0000	计数器的值 (Counter value)

### 1.1.11. TIM1 预分频器（TIM1\_PSC 偏移：0x28）

比特	名称	属性	复位值	描述
15:0	PSC	RW	0x0000	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'

### 1.1.12. TIM1 自动加载寄存器（TIM1\_ARR 偏移：0x2c）

比特	名称	属性	复位值	描述
----	----	----	-----	----

15:0	ARR	RW	0x0000	<p>自动重载的值 (Auto Reload value)</p> <p>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。</p>
------	-----	----	--------	--

### 1.1.13. TIM1 重复计数寄存器 (TIM1\_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	0x00	<p>重复计数器的值 (Repetition counter value)</p> <p>开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIM1_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。这意味着在 PWM 模式中，(REP+1)对应着：</p> <ul style="list-style-type: none"> <li>— 在边沿对齐模式下，PWM 周期的数目；</li> <li>— 在中心对称模式下，PWM 半周期的数目；</li> </ul>

### 1.1.14. TIM1 捕获/比较寄存器 1 (TIM1\_CCR1 偏移: 0x34)

比特	名称	属性	复位值	描述
15:0	CCR1	RW	0x0000	<p>捕获/比较通道 1 的值 (Capture/Compare 1 value)</p> <p>若 CC1 通道配置为输出：CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。如果在 TIM1_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。当前捕获/比较寄存器参与同计数器 TIM1_CNT 的比较，并在 OC1 端口上产生输出信</p>

				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。
--	--	--	--	--

### 1.1.15. TIM1 捕获/比较寄存器 2 (TIM1\_CCR2 偏移: 0x38)

比特	名称	属性	复位值	描述
15:0	CCR2	RW	0x0000	捕获/比较通道 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出：CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。如果在 TIM1_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。当前捕获/比较寄存器参与同计数器 TIM1_CNT 的比较，并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入：CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

### 1.1.16. TIM1 捕获/比较寄存器 3 (TIM1\_CCR3 偏移: 0x3C)

比特	名称	属性	复位值	描述
15:0	CCR3	RW	0x0000	捕获/比较通道 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出：CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。如果在 TIM1_CCMR3 寄存器(OC3PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。当前捕获/比较寄存器参与同计数器 TIM1_CNT 的比较，并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入：CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。

## 1.1.17. TIM1 捕获/比较寄存器 4 (TIM1\_CCR4 偏移: 0x40)

比特	名称	属性	复位值	描述
15:0	CCR4	RW	0x0000	捕获/比较通道 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出: CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。如果在 TIM1_CCMR4 寄存器(OC4PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 4 寄存器中。当前捕获/比较寄存器参与同计数器 TIM1_CNT 的比较, 并在 OC4 端口上产生输出信号。若 CC4 通道配置为输入: CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。

## 1.1.18. TIM1 刹车和死区寄存器 (TIM1\_BDTR 偏移: 0x44)

比特	名称	属性	复位值	描述
15	MOE	RW	0	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(TIM1_CCER 寄存器的 CCxE、CCxNE 位), 则开启 OC 和 OCN 输出。
14	AOE	RW	0	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(TIM1_BDTR 寄存器中的 LOCK 位)设为'1', 则该位不能被修改。
13	BKP	RW	0	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。

				<p>注：一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。</p> <p>注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。</p>
12	BKE	RW	0	<p>刹车功能使能 (Break enable)</p> <p>0: 禁止刹车输入(BRK 及 CCS 时钟失效事件);</p> <p>1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。</p> <p>注：当设置了 LOCK 级别 1 时(TIM1_BDTR 寄存器中的 LOCK 位)，该位不能被修改。</p> <p>注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。</p>
11	OSSR	RW	0	<p>运行模式下“关闭状态”选择 (Off-state selection for Run mode)</p> <p>该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明。</p> <p>0: 当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0);</p> <p>1: 当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OC/OCN 使能输出信号=1。</p> <p>注：一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。</p>
10	OSSI	RW	0	<p>空闲模式下“关闭状态”选择 (Off-state selection for Idle mode)</p> <p>该位用于当 MOE=0 且通道设为输出时。参考 OC/OCN 使能的详细说明。</p> <p>0: 当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0);</p> <p>1: 当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 首先输出其空闲电平，然后 OC/OCN 使能输出信号=1。</p> <p>注：一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。</p>
9:8	LOCK	RW	00	<p>锁定设置 (Lock configuration) 该位为防止软件错误</p>

				<p>而提供写保护。</p> <p>00: 锁定关闭, 寄存器无写保护;</p> <p>01: 锁定级别 1, 不能写入 TIM1_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIM1_CR2 寄存器的 OISx/OISxN 位;</p> <p>10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出, CC 极性位是 TIM1_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位;</p> <p>11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出, CC 控制位是 TIM1_CCMRx 寄存器的 OCxM/OCxPE 位);</p> <p>注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIM1_BDTR 寄存器, 则其内容冻结直至复位。</p>
7:0	UTG	RW	0x00	<p>死区发生器设置 (Dead-time generator setup)</p> <p>这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间:</p> <p><math>DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times Tdtg, Tdtg = TDTS;</math></p> <p><math>DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times Tdtg, Tdtg = 2 \times TDTS;</math></p> <p><math>DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg, Tdtg = 8 \times TDTS;</math></p> <p><math>DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg, Tdtg = 16 \times TDTS;</math></p> <p>例: 若 <math>TDTS = 125ns(8MHZ)</math>, 可能的死区时间为:</p> <p>0 到 15875ns, 若步长时间为 125ns;</p> <p>16us 到 31750ns, 若步长时间为 250ns;</p> <p>32us 到 63us, 若步长时间为 1us;</p> <p>64us 到 126us, 若步长时间为 2us;</p> <p>注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3, 则不能修改这些位。</p>

## 1.1.19. TIM1 DMA 控制寄存器 (TIM1\_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	00000	<p>DMA 连续传送长度 (DMA burst length)</p> <p>这些位定义了 DMA 在连续模式下的传送长度(当对 TIM1_DMAR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的次数, 传输可以是半字(双字节)或字节:</p> <p>00000: 1 次传输  00001: 2 次传输  00010: 3 次传输  .....  10001: 18 次传输</p> <p>例: 我们考虑这样的传输: DBL=7,  DBA=TIM1_CR1 - 如果 DBL=7, DBA=TIM1_CR1 表示待传输数据的地址, 那么传输的地址由下式给出: (TIM1_CR1 的地址)+DBA+(DMA 索引), 其中 DMA 索引 = DBL 其中(TIM1_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(TIM1_CR1 的地址)+DBA 开始的 7 个寄存器。</p> <p>根据 DMA 数据长度的设置, 可能发生以下情况:</p> <ul style="list-style-type: none"> <li>- 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。</li> <li>- 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。</li> </ul>
7:5	-	-	-	位 7:5 保留, 始终读为 0。
4:0	DBA	RW	00000	<p>这些位定义了 DMA 在连续模式下的基地址(当对 TIM1_DMAR 寄存器进行读或写时), DBA 定义为从 TIM1_CR1 寄存器所在地址开始的偏移量:</p> <p>00000: TIM1_CR1,</p>

				00001: TIM1_CR2, 00010: TIM1_SMCR, ...
--	--	--	--	--

### 1.1.20. TIM1 连续模式的 DMA 地址 (TIM1\_DMAR 偏移: 0x4C)

比特	名称	属性	复位值	描述
15:0	DMAB	RW	0x0000	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIM1_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: TIM1_CR1 地址 + DBA + DMA 索引, 其中: “TIM1_CR1 地址”是控制寄存器 1(TIM1_CR1)所在的地址; “DBA”是 TIM1_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, 它取决于 TIM1_DCR 寄存器中定义的 DBL。

### 1.1.21. TIM1 复用功能选择寄存器 (TIM1\_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
15:14	ETRSEL	RW	00	ETR 输入源选择 00: GPIO 01: COMP1 10: COMP2 11: AWD
13:12	-	-	-	保留, 始终为 0。
11	BKCOMP2P	RW	0	比较器 2 输入极性控制 0: 不反相 1: 被反相
10	BKCOMP1P	RW	0	比较器 1 输入极性控制 0: 不反相 1: 被反相
9	BKINP	RW	0	刹车输入极性控制

				0: 不反相 1: 被反相
8:3	-	-	-	保留, 始终为 0。
2	BKCOMP2E	RW	0	比较器 2 输入使能控制 0: 禁止 1: 使能
1	BKCOMP1E	RW	0	比较器 1 输入使能控制 0: 禁止 1: 使能
0	BKINE	RW	0	刹车输入使能控制 0: 禁止 1: 使能

### 1.1.22. TIM1 输入选择寄存器 (TIM1\_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
15:9	-	-	-	保留, 始终为 0。
8	T2SEL	RW	0	TI2 输入选择 0: TIM1_CH2 1: COMP2
7:1	-	-	-	保留, 始终为 0。
0	T1SEL	RW	0	TI1 输入选择 0: TIM1_CH1 1: COMP1

### 1.1.23. TIM1 DMA 请求类型选择寄存器 (TIM1\_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	0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0	COM 事件的 DMA 请求类型

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

## 9. 通用定时器（TIM2）

### 9.1. 概述

通用定时器 TIM2 由一个 32 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。

### 9.2. 主要特性

- 32 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



### 9.3.4. 捕获比较通道

请参考 TIM1 相应章节。

### 9.3.5. 强制输出模式

请参考 TIM1 相应章节。

### 9.3.6. PWM 模式

请参考 TIM1 相应章节。

### 9.3.7. 正交编码器模式

请参考 TIM1 相应章节。

### 9.3.8. 与霍尔传感器的接口

请参考 TIM1 相应章节。

### 9.3.9. 单脉冲模式

请参考 TIM1 相应章节。

### 9.3.10. 定时器互连

### 9.3.11. DMA 功能

请参考 TIM1 相应章节。

## 9.4. 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
0x30	-	保留
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
0x44	-	保留
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
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

#### 1.1.24. 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	00	时钟分频因子 死区发生器和数字滤波器所用的采样时钟与定时器时

				<p>钟 (CK_INT) 的分频比例。</p> <p>00: tDTS=tCK_INT</p> <p>01: tDTS=2 x tCK_INT</p> <p>10: tDTS=4 x tCK_INT</p> <p>11:保留</p>
7	ARPE	RW	0	<p>自动重载预装载允许位</p> <p>0:TIM2_ARR 寄存器没有缓冲</p> <p>1:TIM2_ARR 寄存器被装入缓冲器</p>
6:5	CMS	RW	00	<p>计数模式</p> <p>00:边沿对齐模式，计数器根据方向位 (DIR) 向上或向下计数。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。</p>
4	DIR	RW	0	<p>方向控制位</p> <p>0: 计数器向上计数；</p> <p>1: 计数器向下计数。</p> <p>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。</p>
3	OPM	RW	0	<p>单脉冲模式</p> <p>0: 在发生更新事件时，计数器不停止；</p> <p>1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止。</p>
2	URS	RW	0	更新请求源

				<p>软件通过该位选择 UEV 事件的源</p> <p>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:</p> <ul style="list-style-type: none"> <li>- 计数器溢出/下溢</li> <li>- 设置 UG 位</li> <li>- 从模式控制器产生的更新</li> </ul> <p>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。</p>
1	UDIS	RW	0	<p>禁止更新</p> <p>软件通过该位允许/禁止 UEV 事件的产生</p> <p>0: 允许 UEV。更新(UEV)事件由下述任一事件产生:</p> <ul style="list-style-type: none"> <li>- 计数器溢出/下溢</li> <li>- 设置 UG 位</li> <li>- 从模式控制器产生的更新</li> </ul> <p>具有缓存的寄存器被装入它们的预装载值。(译注: 更新影子寄存器)</p> <p>1: 禁止 UEV。不产生更新事件, 影子寄存器 (ARR、PSC、CCR<sub>x</sub>)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。</p>
0	CEN	RW	0	<p>使能计数器</p> <p>0: 禁止计数器;</p> <p>1: 使能计数器。</p> <p>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。</p>

### 1.1.25. TIM2 控制寄存器 2 (TIM2\_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
15: 8	-	-	-	保留, 始终读为 0。
7	TI1S	RW	0	<p>TI1 选择 (TI1 selection)</p> <p>0: TIM2_CH1 引脚连到 TI1 输入;</p> <p>1: TIM2_CH1、TIM2_CH2 和 TIM2_CH3 引脚经异</p>

				或后连到 TI1 输入。
6:4	MMS	RW	000	<p>主模式选择 (Master mode selection)</p> <p>这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下:</p> <p>000: 复位 – 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。</p> <p>001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式 (见 TIM2_SMCR 寄存器中 MSM 位的描述)。</p> <p>010: 更新 – 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。</p> <p>011: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。</p> <p>100: 比较 – OC1REF 信号被用于作为触发输出 (TRGO)。</p> <p>101: 比较 – OC2REF 信号被用于作为触发输出 (TRGO)。</p> <p>110: 比较 – OC3REF 信号被用于作为触发输出 (TRGO)。</p> <p>111: 比较 – OC4REF 信号被用于作为触发输出 (TRGO)。</p>
3	CCDS	RW	0	<p>捕获/比较的 DMA 选择 (Capture/compare DMA selection)</p> <p>0: 当发生 CCx 事件时, 送出 CCx 的 DMA 请求;</p> <p>1: 当发生更新事件时, 送出 CCx 的 DMA 请求。</p>
2: 0	-	-	-	保留, 始终读为 0。

## 1.1.26. 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
15	ETP	RW	0	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	0	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	00	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	0000	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

				<p>0010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, <math>N=4</math></p> <p>1010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=5</math></p> <p>0011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, <math>N=8</math></p> <p>1011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=6</math></p> <p>0100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, <math>N=6</math></p> <p>1100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=8</math></p> <p>0101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, <math>N=8</math></p> <p>1101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=5</math></p> <p>0110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, <math>N=6</math></p> <p>1110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=6</math></p> <p>0111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, <math>N=8</math></p> <p>1111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=8</math></p>
7	MSM	RW	0	<p>主/从模式 (Master/slave mode)</p> <p>0: 无作用;</p> <p>1: 触发输入(TRGI)上的事件被延迟了, 以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。</p>
6:4	TS	RW	000	<p>触发选择 (Trigger selection)</p> <p>这 3 位选择用于同步计数器的触发输入。</p> <p>000: 内部触发 0(ITR0)</p> <p>100: TI1 的边沿检测器(TI1F_ED)</p> <p>001: 内部触发 1(ITR1)</p> <p>101: 滤波后的定时器输入 1(TI1FP1)</p> <p>010: 内部触发 2(ITR2)</p> <p>110: 滤波后的定时器输入 2(TI2FP2)</p> <p>011: 内部触发 3(ITR3)</p> <p>111: 外部触发输入(ETRF)</p> <p>注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。</p>
3	-	-	-	保留, 始终读为 0。
2:0	SMS	RW	000	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明)</p>

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

表格 9-1 TIM2 从模式互联: ITRx

ITR0	TIM1_TRGO
ITR1	TIM15_TRGO
ITR2	TIM3_TRGO
ITR3	TIM4_TRGO

表格 9-2 TIM2 从模式互联: ETRx

ETRSEL =00	PA0/5/15/PC4
ETRSEL =01	COMP1
ETRSEL =02	COMP2
ETRSEL =03	LSE

## 1.1.27. 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	0	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	-	-	-	保留, 始终读为 0。
12	CC4DE	RW	0	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	0	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	0	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	0	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	0	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	0	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
5	-	-	-	保留, 始终读为 0。
4	CC4IE	RW	0	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	0	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	0	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	0	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	0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

### 1.1.28. TIM2 状态寄存器 (TIM2\_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
15:13	-	-	-	保留, 始终读为 0。
12	CC4OF	RC_W0	0	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	RC_W0	0	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	RC_W0	0	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	RC_W0	0	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	RC_W0	0	<p>触发器中断标记 (Trigger interrupt flag)</p> <p>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'。</p> <p>0: 无触发器事件产生； 1: 触发中断等待响应。</p>
5	-	-	-	保留，始终读为 0。
4	CC4IF	RC_W0	0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	RC_W0	0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	RC_W0	0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	RC_W0	0	<p>捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag)</p> <p>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。</p> <p>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</p> <p>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'。</p> <p>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)。</p>
0	UIF	RC_W0	0	<p>更新中断标记 (Update interrupt flag)</p> <p>当产生更新事件时该位由硬件置'1'。它由软件清'0'。</p>

				<p>0: 无更新事件产生;</p> <p>1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1':</p> <ul style="list-style-type: none"> <li>- 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)。</li> <li>- 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。</li> <li>- 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。</li> </ul>
--	--	--	--	--

### 1.1.29. 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	0	<p>产生触发事件 (Trigger generation)</p> <p>该位由软件置'1', 用于产生一个触发事件, 由硬件自动清'0'。</p> <p>0: 无动作;</p> <p>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。</p>
5	-	-	-	保留, 始终读为 0。
4	CC4G	WO	0	<p>产生捕获/比较 4 事件 (Capture/Compare 4 generation)</p> <p>参考 CC1G 描述。</p>
3	CC3G	WO	0	<p>产生捕获/比较 3 事件 (Capture/Compare 3 generation)</p> <p>参考 CC1G 描述。</p>
2	CC2G	WO	0	<p>产生捕获/比较 2 事件 (Capture/Compare 2 generation)</p> <p>参考 CC1G 描述。</p>
1	CC1G	WO	0	<p>产生捕获/比较 1 事件 (Capture/Compare 1 generation)</p> <p>该位由软件置'1', 用于产生一个捕获/比较事件, 由硬件自动清'0'。</p> <p>0: 无动作;</p> <p>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</p>

				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	0	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。

### 1.1.30. 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	0	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	000	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	RW	0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	00	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	0	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	000	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	0	<p>输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p>
2	OC1FE	RW	0	<p>输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择。(Capture/Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: CC1 通道被配置为输入, IC1 映射在 TRC 上。</p> <p>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)。</p>

				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	000	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	00	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	00	<p>捕获/比较 2 选择 (Capture/Compare 2 selection)</p> <p>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)。</p> <p>注：CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E=0)才是可写的。</p>
7:4	IC1F	RW	0000	<p>输入捕获 1 滤波器 (Input capture 1 filter)</p> <p>这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变：</p> <p>0000: 无滤波器，以 fDTS 采样            1000: 采样频率 <math>f_{SAMPLING}=f_{DTS}/8</math>, N=6            0001: 采样频率 <math>f_{SAMPLING}=f_{CK\_INT}</math>, N=2            1001: 采样频率 <math>f_{SAMPLING}=f_{DTS}/8</math>, N=8            0010: 采样频率 <math>f_{SAMPLING}=f_{CK\_INT}</math>, N=4            1010: 采样频率 <math>f_{SAMPLING}=f_{DTS}/16</math>, N=5            0011: 采样频率 <math>f_{SAMPLING}=f_{CK\_INT}</math>, N=8            1011: 采样频率 <math>f_{SAMPLING}=f_{DTS}/16</math>, N=6            0100: 采样频率 <math>f_{SAMPLING}=f_{DTS}/2</math>, N=6            1100: 采样频率 <math>f_{SAMPLING}=f_{DTS}/16</math>, N=8            0101: 采样频率 <math>f_{SAMPLING}=f_{DTS}/2</math>, N=8            1101: 采样频率 <math>f_{SAMPLING}=f_{DTS}/32</math>, N=5            0110: 采样频率 <math>f_{SAMPLING}=f_{DTS}/4</math>, N=6</p>

				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	0	输入/捕获 1 预分频器 (Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 $CC1E=0$ (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	0	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)。

### 1.1.31. 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	0	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M	RW	000	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	RW	0	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	RW	0	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S	RW	00	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	0	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M	RW	000	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	RW	0	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	RW	0	输出比较 3 快速使能 (Output compare 3 fast enable)
1:0	CC3S	RW	00	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	000	输入捕获 4 滤波器 (Input capture 4 filter)
11:10	IC4PSC	RW	00	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	00	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	0000	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	00	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	00	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)。

### 1.1.32. 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
15	CC4NP	RW	0	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	0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0	输入/捕获 3 互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。
10	-	-	-	保留, 始终读为 0。
9	CC3P	RW	0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0	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	0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0	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	-	-	-	保留，始终读为 0。
1	CC1P	RW	0	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	0	<p>输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出：</p> <p>0： 关闭— OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>1： 开启— OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>CC1 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。</p> <p>0： 捕获禁止；</p> <p>1： 捕获使能。</p>
---	------	----	---	---

### 1.1.33. TIM2 计数器 (TIM2\_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
31:0	CNT	RW	0x00000000	计数器的值 (Counter value)

### 1.1.34. 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	0x0000	<p>预分频器的值 (Prescaler value)</p> <p>计数器的时钟频率(CK_CNT)等于 <math>f_{CK\_PSC}/(PSC[15:0]+1)</math>。</p> <p>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'</p>

### 1.1.35. TIM2 自动重载寄存器 (TIM2\_ARR 偏移: 0x2C)

比特	名称	属性	复位值	描述
----	----	----	-----	----

31:0	ARR	RW	0x00000000	自动重载的值 (Auto Reload 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。
------	-----	----	------------	---

### 1.1.36. 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	0x0000	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。

### 1.1.37. 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	0x0000	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。

**1.1.38. 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	0x0000	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。

**1.1.39. 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	0x0000	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。

**1.1.40. 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	00000	<p>DMA 连续传送长度 (DMA burst length)</p> <p>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：</p> <p>00000: 1 次传输</p> <p>00001: 2 次传输</p> <p>00010: 3 次传输</p> <p>.....</p> <p>10001: 18 次传输</p> <p>例：我们考虑这样的传输：DBL=7，</p> <p>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。</p> <p>根据 DMA 数据长度的设置，可能发生以下情况：</p> <ul style="list-style-type: none"> <li>- 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。</li> <li>- 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。</li> </ul>
7:5	-	-	-	位 7:5 保留，始终读为 0。
4:0	DBA	RW	00000	<p>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：</p> <p>00000: TIM2_CR1，</p> <p>00001: TIM2_CR2，</p> <p>00010: TIM2_SMCR，</p> <p>...</p>

**1.1.41. 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
15:0	DMAB	RW	0x0000	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, 它取决于 TIM2_DCR 寄存器中定义的 DBL。

**1.1.42. TIM2 复用功能选择寄存器 (TIM2\_AF1 偏移: 0x60)**

比特	名称	属性	复位值	描述
15:14	ETRSEL	RW	00	ETR 输入源选择 00: GPIO 01: COMP1 10: COMP2 11: AWD
13:0	-	-	-	保留, 始终为 0。

**1.1.43. TIM2 输入选择寄存器 (TIM2\_TISEL 偏移: 0x68)**

比特	名称	属性	复位值	描述
15:9	-	-	-	保留, 始终为 0。
8	T2SEL	RW	0	TI2 输入选择 0: TIM_CH2 1: COMP2
7:1	-	-	-	保留, 始终为 0。

0	T1SEL	RW	0	TI1 输入选择 0: TIM_CH1 1: COMP1
---	-------	----	---	------------------------------------

#### 1.1.44. 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	0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;
4	CC4BE	RW	0	捕获/比较 4 事件的 DMA 请求类型 0: Single; 1: Burst;
3	CC3BE	RW	0	捕获/比较 3 事件的 DMA 请求类型 0: Single; 1: Burst;
2	CC2BE	RW	0	捕获/比较 2 事件的 DMA 请求类型 0: Single; 1: Burst;
1	CC1BE	RW	0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

## 10. 通用定时器（TIM3/TIM4）

### 10.1. 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)。

### 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
- 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



### 10.3.5. 强制输出模式

请参考 TIM1 相应章节。

### 10.3.6. PWM 模式

请参考 TIM1 相应章节。

### 10.3.7. 正交编码器模式

请参考 TIM1 相应章节。

### 10.3.8. 与霍尔传感器的接口

请参考 TIM1 相应章节。

### 10.3.9. 单脉冲模式

请参考 TIM1 相应章节。

### 10.3.10. 定时器互连

请参考 TIM1 相应章节。

### 10.3.11. DMA 功能

请参考 TIM1 相应章节。

## 10.4. 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
0x30	-	保留
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	-	保留
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
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
15:10	-	-	-	保留, 始终读为 0。
9:8	CKD	RW	00	时钟分频因子 死区发生器和数字滤波器所用的采样时钟与定时器时钟 (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	0	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	00	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	0	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	0	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	0	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	0	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。(译注: 更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器 (ARR、PSC、CCR <sub>x</sub> )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	0	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
15: 8	-	-	-	保留, 始终读为 0。
7	TIIS	RW	0	<p>TI1 选择 (TI1 selection)</p> <p>0: TIMx_CH1 引脚连到 TI1 输入;</p> <p>1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。</p>
6:4	MMS	RW	000	<p>主模式选择 (Master mode selection)</p> <p>这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下:</p> <p>000: 复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。</p> <p>001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式 (见 TIMx_SMCR 寄存器中 MSM 位的描述)。</p> <p>010: 更新 – 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。</p> <p>011: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。</p> <p>100: 比较 – OC1REF 信号被用于作为触发输出 (TRGO)。</p> <p>101: 比较 – OC2REF 信号被用于作为触发输出 (TRGO)。</p> <p>110: 比较 – OC3REF 信号被用于作为触发输出 (TRGO)。</p> <p>111: 比较 – OC4REF 信号被用于作为触发输出</p>

				(TRGO)。
3	CCDS	RW	0	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: 0	-	-	-	保留, 始终读为 0。

## 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
15	ETP	RW	0	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	0	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	00	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	0000	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

				<p>0010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, <math>N=4</math></p> <p>1010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=5</math></p> <p>0011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, <math>N=8</math></p> <p>1011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=6</math></p> <p>0100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, <math>N=6</math></p> <p>1100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, <math>N=8</math></p> <p>0101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, <math>N=8</math></p> <p>1101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=5</math></p> <p>0110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, <math>N=6</math></p> <p>1110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=6</math></p> <p>0111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, <math>N=8</math></p> <p>1111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, <math>N=8</math></p>
7	MSM	RW	0	<p>主/从模式 (Master/slave mode)</p> <p>0: 无作用;</p> <p>1: 触发输入(TRGI)上的事件被延迟了, 以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。</p>
6:4	TS	RW	000	<p>触发选择 (Trigger selection)</p> <p>这 3 位选择用于同步计数器的触发输入。</p> <p>000: 内部触发 0(ITR0)</p> <p>100: TI1 的边沿检测器(TI1F_ED)</p> <p>001: 保留</p> <p>101: 滤波后的定时器输入 1(TI1FP1)</p> <p>010: 内部触发 2(ITR2)</p> <p>110: 滤波后的定时器输入 2(TI2FP2)</p> <p>011: 内部触发 3(ITR3)</p> <p>111: 外部触发输入(ETRF)</p> <p>注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。</p>
3	-	-	-	保留, 始终读为 0。
2:0	SMS	RW	000	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明)</p>

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

表格 10-1 TIM3 从模式互联: ITRx

ITR0	TIM1_TRGO
ITR1	TIM2_TRGO
ITR2	TIM15_TRGO
ITR3	TIM4_TRGO

表格 10-2 TIM3 从模式互联: ETRx

ETRSEL =00	PD2
ETRSEL =01	COMP1
ETRSEL =02	COMP2
ETRSEL =03	AWD

表格 10-3 TIM4 从模式互联: ITRx

ITR0	TIM1_TRGO
ITR1	TIM2_TRGO
ITR2	TIM3_TRGO

ITR3	TIM15_TRGO
------	------------

表格 10-4 TIM4 从模式互联: ETRx

ETRSEL =00	
ETRSEL =01	COMP1
ETRSEL =02	COMP2
ETRSEL =03	LSE

## 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	0	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	-	-	-	保留, 始终读为 0。
12	CC4DE	RW	0	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	0	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	0	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	0	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	0	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	0	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
5	-	-	-	保留，始终读为 0。
4	CC4IE	RW	0	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	0	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	0	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	0	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	0	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
15:13	-	-	-	保留，始终读为 0。
12	CC4OF	RC_W0	0	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	RC_W0	0	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	RC_W0	0	捕获/比较 2 重复捕获标记 (Capture/Compare 2

				overcapture flag) 参见 CC10F 描述。
9	CC10F	RC_W0	0	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	RC_W0	0	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	-	-	-	保留, 始终读为 0。
4	CC4IF	RC_W0	0	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	RC_W0	0	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	RC_W0	0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	RC_W0	0	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

				<p>TIMx_CCR1 清'0'。</p> <p>0: 无输入捕获产生;</p> <p>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)。</p>
0	UIF	RC_WO	0	<p>更新中断标记 (Update interrupt flag)</p> <p>当产生更新事件时该位由硬件置'1'。它由软件清'0'。</p> <p>0: 无更新事件产生;</p> <p>1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1':</p> <ul style="list-style-type: none"> <li>- 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)。</li> <li>- 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。</li> <li>- 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。</li> </ul>

#### 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: 7	-	-	-	保留, 始终读为 0。
6	TG	WO	0	<p>产生触发事件 (Trigger generation)</p> <p>该位由软件置'1', 用于产生一个触发事件, 由硬件自动清'0'。</p> <p>0: 无动作;</p> <p>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。</p>
5	-	-	-	保留, 始终读为 0。
4	CC4G	WO	0	<p>产生捕获/比较 4 事件 (Capture/Compare 4 generation)</p> <p>参考 CC1G 描述。</p>
3	CC3G	WO	0	<p>产生捕获/比较 3 事件 (Capture/Compare 3 generation)</p> <p>参考 CC1G 描述。</p>
2	CC2G	WO	0	<p>产生捕获/比较 2 事件 (Capture/Compare 2 generation)</p>

				参考 CC1G 描述。
1	CC1G	WO	0	<p>产生捕获/比较 1 事件 (Capture/Compare 1 generation)</p> <p>该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。</p> <p>0: 无动作；</p> <p>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。</p>
0	UG	WO	0	<p>产生更新事件 (Update generation)</p> <p>该位由软件置'1'，由硬件自动清'0'。</p> <p>0: 无动作；</p> <p>1: 重新初始化计数器，并产生一个更新事件。</p> <p>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。</p>

### 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。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

比特	名称	属性	复位值	描述
15	OC2CE	RW	0	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14:12	OC2M	RW	000	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	RW	0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	00	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	0	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	000	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；

				<p>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。</p> <p>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。</p> <p>011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为无效电平。强制 OC1REF 为低。</p> <p>101: 强制为有效电平。强制 OC1REF 为高。</p> <p>110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT&lt;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&gt;TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT&lt;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&gt;TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平。</p> <p>注 1: 一旦 LOCK 级别设为 3(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。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
3	OC1PE	RW	0	<p>输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>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。</p> <p>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。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中</p>

				<p>的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。</p> <p>注 2: 仅在单脉冲模式下(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。</p>
2	OC1FE	RW	0	<p>输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OC1FE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 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)。</p> <p>注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。</p>

## 输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC2F	RW	000	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	00	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	00	<p>捕获/比较 2 选择 (Capture/Compare 2 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2 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;</p>

				<p>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)。</p> <p>注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。</p>
7:4	IC1F	RW	0000	<p>输入捕获 1 滤波器 (Input capture 1 filter)</p> <p>这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变:</p> <p>0000: 无滤波器, 以 fDTS 采样</p> <p>1000: 采样频率 fSAMPLING=fDTS/8, N=6</p> <p>0001: 采样频率 fSAMPLING=fCK_INT, N=2</p> <p>1001: 采样频率 fSAMPLING=fDTS/8, N=8</p> <p>0010: 采样频率 fSAMPLING=fCK_INT, N=4</p> <p>1010: 采样频率 fSAMPLING=fDTS/16, N=5</p> <p>0011: 采样频率 fSAMPLING=fCK_INT, N=8</p> <p>1011: 采样频率 fSAMPLING=fDTS/16, N=6</p> <p>0100: 采样频率 fSAMPLING=fDTS/2, N=6</p> <p>1100: 采样频率 fSAMPLING=fDTS/16, N=8</p> <p>0101: 采样频率 fSAMPLING=fDTS/2, N=8</p> <p>1101: 采样频率 fSAMPLING=fDTS/32, N=5</p> <p>0110: 采样频率 fSAMPLING=fDTS/4, N=6</p> <p>1110: 采样频率 fSAMPLING=fDTS/32, N=6</p> <p>0111: 采样频率 fSAMPLING=fDTS/4, N=8</p> <p>1111: 采样频率 fSAMPLING=fDTS/32, N=8</p>
3:2	IC1PSC	RW	00	<p>输入/捕获 1 预分频器 (Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMx_CCER 寄存器中), 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每 2 个事件触发一次捕获;</p> <p>10: 每 4 个事件触发一次捕获;</p> <p>11: 每 8 个事件触发一次捕获。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择 (Capture/Compare 1 Selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选</p>

				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	0	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M	RW	000	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	RW	0	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	RW	0	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S	RW	00	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	0	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M	RW	000	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	RW	0	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	RW	0	输出比较 3 快速使能 (Output compare 3 fast enable)
1:0	CC3S	RW	00	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;

				<p>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。</p>
--	--	--	--	---

## 输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC4F	RW	000	输入捕获 4 滤波器 (Input capture 4 filter)
11:10	IC4PSC	RW	00	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S	RW	00	<p>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。</p>
7:4	IC3F	RW	0000	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC	RW	00	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S	RW	00	<p>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)。</p>

## 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
15	CC4NP	RW	0	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	0	输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	RW	0	输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	RW	0	输入/捕获 3 互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。
10	-	-	-	保留, 始终读为 0。
9	CC3P	RW	0	输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	RW	0	输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	RW	0	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	0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0	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	-	-	-	保留, 始终读为 0。

1	CC1P	RW	0	<p>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。</p> <p>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）</p> <p>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。</p>
0	CC1E	RW	0	<p>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。</p> <p>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。</p>

表格 10-5 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
--------	----------

0	输出禁止 (OCx=0, OCx_EN=0)
1	OCx=OCxREF + 极性, OCx_EN=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	0x0000	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	0x0000	<p>预分频器的值 (Prescaler value)</p> <p>计数器的时钟频率(CK_CNT)等于 <math>f_{CK\_PSC}/(PSC[15:0]+1)</math>。</p> <p>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'</p>

#### 10.4.12. TIMx 自动重装载寄存器 (TIMx\_ARR 偏移: 0x2C)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0000	<p>自动重装载的值 (Auto Reload value)</p> <p>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。</p>

#### 10.4.13. 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	0x0000	<p>捕获/比较通道 1 的值 (Capture/Compare 1 value)</p> <p>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。</p>
------	------	----	--------	---

#### 10.4.14. 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	0x0000	<p>捕获/比较通道 2 的值 (Capture/Compare 2 value)</p> <p>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。</p>

#### 10.4.15. 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	0x0000	<p>捕获/比较通道 3 的值 (Capture/Compare 3 value)</p> <p>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。</p>
------	------	----	--------	---

#### 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	0x0000	<p>捕获/比较通道 4 的值 (Capture/Compare 4 value)</p> <p>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。</p>

#### 10.4.16. 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	00000	<p>DMA 连续传送长度 (DMA burst length)</p> <p>这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，即：定义传输的次数，传输可以是半字(双字节)或字节：</p> <p>00000: 1 次传输</p> <p>00001: 2 次传输</p> <p>00010: 3 次传输</p> <p>.....</p> <p>10001: 18 次传输</p> <p>例：我们考虑这样的传输：DBL=7，</p> <p>DBA=TIM2_CR1 - 如果 DBL=7，DBA=TIM2_CR1 表示待传输数据的地址，那么传输的地址由下式给出：(TIMx_CR1 的地址)+DBA+(DMA 索引)，其中 DMA 索引 =DBL 其中(TIMx_CR1 的地址)+DBA 再加上 7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址(TIMx_CR1 的地址)+DBA 开始的 7 个寄存器。</p> <p>根据 DMA 数据长度的设置，可能发生以下情况：</p> <ul style="list-style-type: none"> <li>- 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。</li> <li>- 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。</li> </ul>
7:5	-	-	-	位 7:5 保留，始终读为 0。
4:0	DBA	RW	00000	<p>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：</p> <p>00000: TIMx_CR1，</p> <p>00001: TIMx_CR2，</p> <p>00010: TIMx_SMCR，</p> <p>...</p>

## 10.4.17. 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
15:0	DMAB	RW	0x0000	<p>DMA 连续传送寄存器 (DMA register for burst accesses)</p> <p>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, 它取决于 TIMx_DCR 寄存器中定义的 DBL。</p>

## 10.4.18. TIMx 复用功能选择寄存器 (TIMx\_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
15:14	ETRSEL	RW	00	<p>ETR 输入源选择</p> <p>00: GPIO</p> <p>01: COMP1</p> <p>10: COMP2</p> <p>11: AWD</p>
13:0	-	-	-	保留, 始终为 0。

## 10.4.19. TIMx 输入选择寄存器 (TIMx\_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
15:9	-	-	-	保留, 始终为 0。
8	T2SEL	RW	0	<p>TI2 输入选择</p> <p>0: TIM_CH2</p> <p>1: COMP2</p>

7:1	-	-	-	保留，始终为 0。
0	T1SEL	RW	0	TI1 输入选择 0: TIM_CH1 1: COMP1

#### 10.4.20. 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	0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;
4	CC4BE	RW	0	捕获/比较 4 事件的 DMA 请求类型 0: Single; 1: Burst;
3	CC3BE	RW	0	捕获/比较 3 事件的 DMA 请求类型 0: Single; 1: Burst;
2	CC2BE	RW	0	捕获/比较 2 事件的 DMA 请求类型 0: Single; 1: Burst;
1	CC1BE	RW	0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

## 11. 基本定时器（TIM6/TIM7）

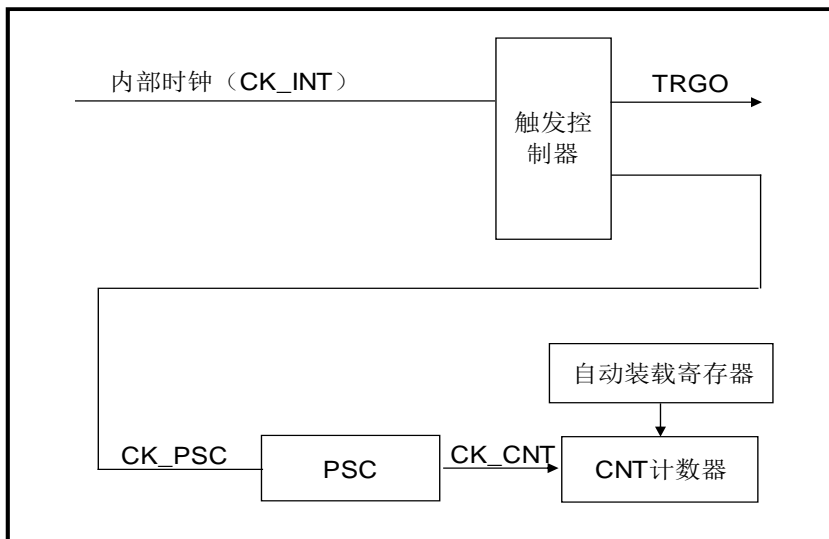
### 11.1. 概述

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。

### 11.2. 主要特性

- 16 位自动重载累加计数器
- 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

图 11-1 基本定时器 TIM6/TIM7 结构框图



注：TIM6 的 TRGO 可作为 ADC 的外部触发源，TIM7 的 TRGO 可作为 ADC 的外部触发源，详见 ADC/DAC 章节。定时器之间的互联以及刹车输入信号源，详见航芯 ACM32F4 和 F3 芯片定时器差异说明.docx 的定时器互联章节以及本章的刹车功能章节。

### 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

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。

### 11.3.4. 定时器互连

请参考 TIM1 相应章节。

### 11.3.5. DMA 功能

请参考 TIM1 相应章节。

## 11.4. 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	-	保留
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

## 11.4.1. TIMx 控制寄存器 1 (TIMx\_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15:8	-	-	-	保留, 始终读为 0。
7	ARPE	RW	0	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	0	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, 计数停止。
2	URS	RW	0	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	0	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。(译注: 更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器 (ARR、PSC、CCR <sub>x</sub> )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	0	使能计数器 0: 禁止计数器;

				<p>1: 使能计数器。</p> <p>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。</p>
--	--	--	--	--

## 11.4.2. TIMx 控制寄存器 2 (TIMx\_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
15:7	-	-	-	保留, 始终读为 0。
6:4	MMS	RW	000	<p>主模式选择 (Master mode selection)</p> <p>这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下:</p> <p>000: 复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。</p> <p>001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式 (见 TIMx_SMCR 寄存器中 MSM 位的描述)。</p> <p>010: 更新 – 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。</p> <p>011: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。</p> <p>100: 比较 – OC1REF 信号被用于作为触发输出 (TRGO)。</p> <p>101: 比较 – OC2REF 信号被用于作为触发输出 (TRGO)。</p> <p>110: 比较 – OC3REF 信号被用于作为触发输出 (TRGO)。</p> <p>111: 比较 – OC4REF 信号被用于作为触发输出 (TRGO)。</p>
3:0	-	-	-	保留, 始终读为 0。

## 11.4.3. 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	0	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	0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

## 11.4.4. TIMx 状态寄存器 (TIMx\_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
15: 1	-	-	-	保留, 始终读为 0。
0	UIF	RW	0	<p>更新中断标记 (Update interrupt flag)</p> <p>当产生更新事件时该位由硬件置'1'。它由软件清'0'。</p> <p>0: 无更新事件产生;</p> <p>1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1':</p> <ul style="list-style-type: none"> <li>- 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)。</li> <li>- 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。</li> <li>- 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。</li> </ul>

## 11.4.5. 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	0	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。

## 11.4.6. TIMx 计数器 (TIMx\_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0000	计数器的值 (Counter value)

## 11.4.7. 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	0x0000	<p>预分频器的值 (Prescaler value)</p> <p>计数器的时钟频率(CK_CNT)等于 <math>f_{CK\_PSC}/(PSC[15:0]+1)</math>。</p> <p>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'</p>

## 11.4.8. TIMx 自动重载寄存器 (TIMx\_ARR 偏移: 0x2C)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0000	<p>自动重载的值 (Auto Reload value)</p> <p>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。</p>

## 12. 通用定时器（TIM14）

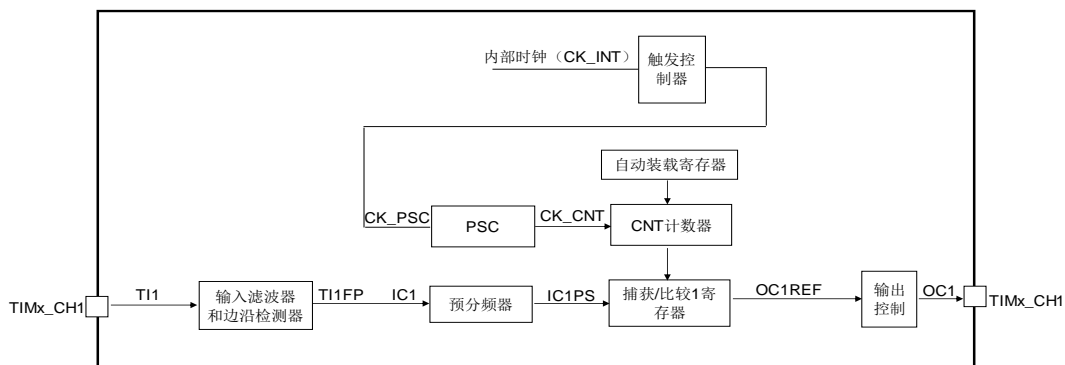
### 12.1. 概述

通用定时器 TIM14 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。

### 12.2. 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 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成
  - 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

图 12-1 通用定时器 TIM14 结构框图



## 12.3. 功能描述

### 12.3.1. 计数单元

请参考 TIM1 相应章节。

### 12.3.2. 预分频器

请参考 TIM1 相应章节。

### 12.3.3. 时钟源选择

计数器的时钟由内部时钟(CK\_INT)提供。请参考 TIM1 相应章节。

### 12.3.4. 捕获比较通道

请参考 TIM1 相应章节。

### 12.3.5. 强制输出模式

请参考 TIM1 相应章节。

### 12.3.6. PWM 模式

请参考 TIM1 相应章节。

### 12.3.7. 单脉冲模式

请参考 TIM1 相应章节。

### 12.3.8. 定时器互连

请参考 TIM1 相应章节。

### 12.3.9. DMA 功能

请参考 TIM1 相应章节。

## 12.4. 寄存器描述

寄存器基地址：0x4000\_2000

偏置	名称	描述
0x00	TIMx_CR1	TIMx 控制寄存器 1
0x04	-	保留
0x08	-	保留
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	-	保留
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	-	保留
0x34	TIMx_CCR1	TIMx 捕获比较寄存器 1

## 12.4.1. TIMx 控制寄存器 1 (TIMx\_CR1 偏移: 0x00)

比特	名称	属性	复位值	描述
15:10	-	-	-	保留，读始终为 0。
9:8	CKD	RW	00	时钟分频因子 死区发生器和数字滤波器所用的采样时钟与定时器时钟 (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	0	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
6:3	-	-	-	保留，读始终为 0。
2	URS	RW	0	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	0	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。(译注：更新影子寄存器) 1: 禁止 UEV。不产生更新事件，影子寄存器 (ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器

				和预分频器被重新初始化。
0	CEN	RW	0	<p>使能计数器</p> <p>0: 禁止计数器;</p> <p>1: 使能计数器。</p> <p>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。</p>

## 12.4.2. 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
15:2	-	-	-	保留, 始终读为 0。
1	CC1IE	RW	0	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	0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

## 12.4.3. TIMx 状态寄存器 (TIMx\_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
15:10	-	-	-	保留, 始终读为 0。
9	CC1OF	RC_W0	0	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'。
8:2	-	-	-	位 8 保留, 始终读为 0。
1	CC1IF	RC_W0	0	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

				<p>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'。</p> <p>0： 无输入捕获产生；</p> <p>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)。</p>
0	UIF	RC_W0	0	<p>更新中断标记 (Update interrupt flag)</p> <p>当产生更新事件时该位由硬件置'1'。它由软件清'0'。</p> <p>0： 无更新事件产生；</p> <p>1： 更新中断等待响应。当寄存器被更新时该位由硬件置'1'：</p> <ul style="list-style-type: none"> <li>- 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)。</li> <li>- 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。</li> <li>- 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。</li> </ul>

## 12.4.4. TIMx 事件产生寄存器 (TIMx\_EGR 偏移: 0x14)

比特	名称	属性	复位值	描述
15:2	-	-	-	保留，始终读为 0。
1	CC1G	WO	0	<p>产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。</p> <p>0: 无动作； 1: 在通道 CC1 上产生一个捕获/比较事件：若通道 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。</p>
0	UG	WO	0	<p>产生更新事件 (Update generation) 该位由软件置'1'，由硬件自动清'0'。</p> <p>0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。 注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'；若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。</p>

### 12.4.5. 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。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输入模式下的功能。因此必须注意, 同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式:

比特	名称	属性	复位值	描述
15:7	-	-	-	保留, 始终读为 0。
6:4	OC1M	RW	000	<p>输出比较 1 模式 (Output Compare 1 mode)</p> <p>该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。</p> <p>000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用;</p> <p>001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时, 强制 OC1REF 为高。</p> <p>010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时, 强制 OC1REF 为低。</p> <p>011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为无效电平。强制 OC1REF 为低。</p> <p>101: 强制为有效电平。强制 OC1REF 为高。</p> <p>110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT&lt;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&gt;TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT&lt;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</p>

				<p>TIMx_CNT&gt;TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。</p> <p>注 1：一旦 LOCK 级别设为 3(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。</p> <p>注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。</p>
3	OC1PE	RW	0	<p>输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p>
2	OC1FE	RW	0	<p>输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择。(Capture/Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出；</p>

				01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 其它: 保留 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
15:8	-	-	-	保留, 始终读为 0。
7:4	IC1F	RW	0000	<p>输入捕获 1 滤波器 (Input capture 1 filter)</p> <p>这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变:</p> <p>0000: 无滤波器, 以 fDTS 采样</p> <p>1000: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, N=6</p> <p>0001: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=2</p> <p>1001: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, N=8</p> <p>0010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=4</p> <p>1010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=5</p> <p>0011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=8</p> <p>1011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=6</p> <p>0100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, N=6</p> <p>1100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=8</p> <p>0101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, N=8</p> <p>1101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=5</p> <p>0110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, N=6</p> <p>1110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=6</p> <p>0111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, N=8</p> <p>1111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=8</p>
3:2	IC1PSC	RW	00	<p>输入/捕获 1 预分频器 (Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMx_CCER 寄存器中), 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每 2 个事件触发一次捕获;</p> <p>10: 每 4 个事件触发一次捕获;</p>

				11: 每 8 个事件触发一次捕获。
1:0	CC1S	RW	00	<p>捕获/比较 1 选择 (Capture/Compare 1 Selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>其它: 保留</p> <p>注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。</p>

#### 12.4.6. 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
15:4	-	-	-	保留, 始终读为 0。
3	CC1NP	RW	0	<p>输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity)</p> <p>0: OC1N 高电平有效;</p> <p>1: OC1N 低电平有效。</p> <p>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。</p> <p>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。</p>
2	-	-	-	保留, 始终读为 0。
1	CC1P	RW	0	<p>输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出:</p> <p>0: OC1 高电平有效;</p> <p>1: OC1 低电平有效。</p> <p>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。</p> <p>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。</p> <p>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</p>

				<p>式或编码器模式下触发操作，TIxFP1 反相。</p> <p>10: 保留，不使用此配置。</p> <p>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）</p> <p>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。</p>
0	CC1E	RW	0	<p>输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出：</p> <p>0: 关闭— OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>1: 开启— OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>CC1 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。</p> <p>0: 捕获禁止；</p> <p>1: 捕获使能。</p>

表格 12-1 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	输出禁止 (OCx=0, OCx_EN=0)
1	OCx=OCxREF + 极性, OCx_EN=1

#### 12.4.7. TIMx 计数器 (TIMx\_CNT 偏移: 0x24)

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0000	计数器的值 (Counter value)

## 12.4.8. 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	0x0000	<p>预分频器的值 (Prescaler value)</p> <p>计数器的时钟频率(CK_CNT)等于 <math>f_{CK\_PSC}/(PSC[15:0]+1)</math>。</p> <p>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'</p>

## 12.4.9. TIMx 自动重载寄存器 (TIMx\_ARR 偏移: 0x2C)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0000	<p>自动重载的值 (Auto Reload value)</p> <p>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。</p>

## 12.4.10. 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	0x0000	<p>捕获/比较通道 1 的值 (Capture/Compare 1 value)</p> <p>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。</p> <p>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。</p>

## 13. 通用定时器（TIM15/TIM16/TIM17）

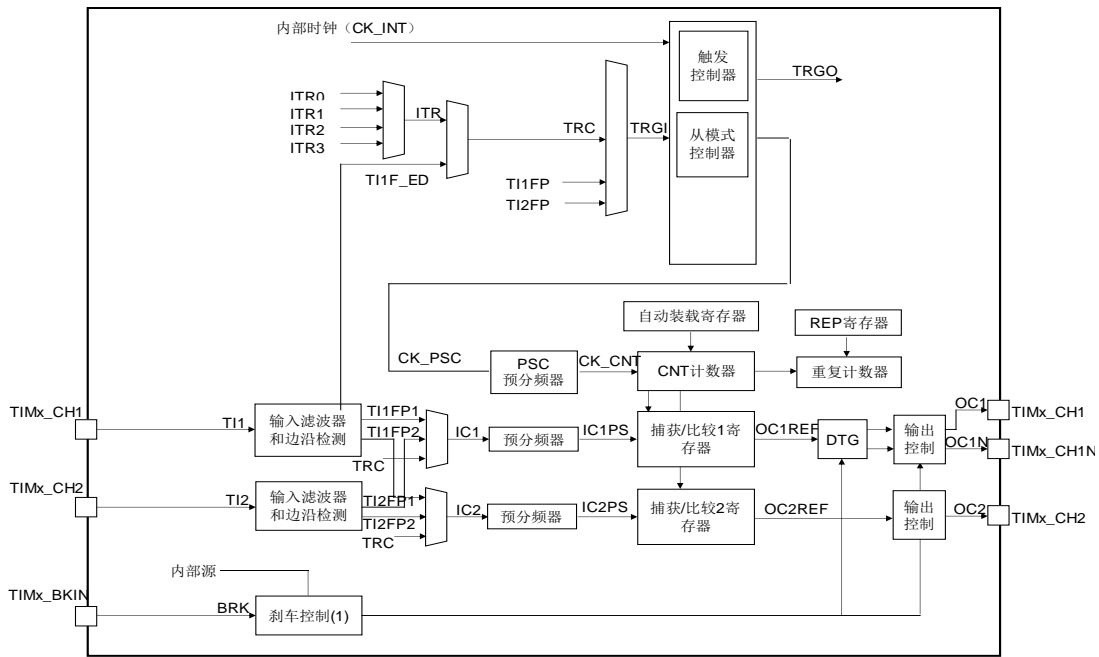
### 13.1. 概述

通用定时器 TIM15/TIM16/TIM17 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。

### 13.2. 主要特性

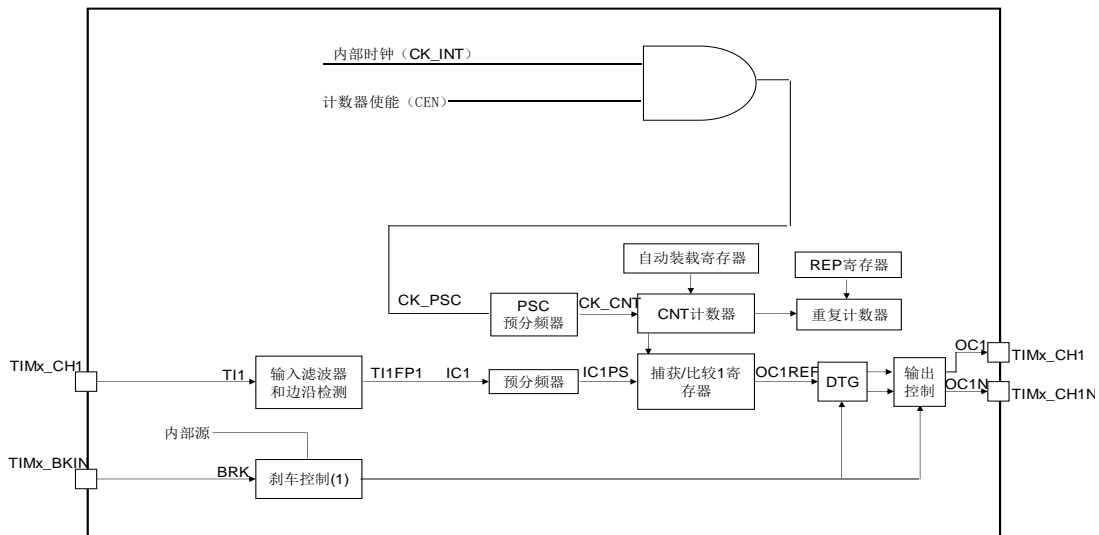
- 16 位向上自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值
- 1 个/2 个独立通道（TIM15 两个通道，TIM16 和 TIM17 各一个通道）：
  - 输入捕获
  - 输出比较
  - PWM 生成
  - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路（TIM15）
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢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

图 13-1 通用定时器 TIM15 结构框图



注：TIM15 的 TRGO 可作为 DAC 的外部触发源，详见 ADC/DAC 章节。定时器之间的互联以及刹车输入信号源，详见航芯 ACM32F4 和 F3 芯片定时器差异说明.docx 的定时器互联章节以及本章的刹车功能章节。

图 13-2 通用定时器 TIM16/TIM17 结构框图



### 13.3. 功能描述

#### 13.3.1. 计数单元

请参考 TIM1 相应章节。

### 13.3.2. 预分频器

请参考 TIM1 相应章节。

### 13.3.3. 时钟源选择

请参考 TIM1 相应章节。

### 13.3.4. 捕获比较通道

请参考 TIM1 相应章节。

### 13.3.5. 强制输出模式

请参考 TIM1 相应章节。

### 13.3.6. PWM 模式

请参考 TIM1 相应章节。

### 13.3.7. 互补输出和死区插入

请参考 TIM1 相应章节。

### 13.3.8. 刹车功能

请参考 TIM1 相应章节。

### 13.3.9. 六步 PWM 输出

请参考 TIM1 相应章节。

### 13.3.10. 单脉冲模式

请参考 TIM1 相应章节。

### 13.3.11. 定时器互连

请参考 TIM1 相应章节。

### 13.3.12. DMA 功能

请参考 TIM1 相应章节。

## 13.4. 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
0x1C	-	保留
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	-	保留
0x40	-	保留
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
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

## 13.4.1. 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, 读始终为 0。
13:10	BKF	RW	0000	<p>刹车滤波器 (Break filter)</p> <p>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:</p> <p>0000: 无滤波器, 以 fDTS 采样</p> <p>100:: 0: 采样频率 fSAMPLING=fDTS/8, N=6</p> <p>0001: 采样频率 fSAMPLING=fCK_INT, N=2</p> <p>1001: 采样频率 fSAMPLING=fDTS/8, N=8</p> <p>0010: 采样频率 fSAMPLING=fCK_INT, N=4</p> <p>1010: 采样频率 fSAMPLING=fDTS/16, N=5</p> <p>0011: 采样频率 fSAMPLING=fCK_INT, N=8</p> <p>1011: 采样频率 fSAMPLING=fDTS/16, N=6</p> <p>0100: 采样频率 fSAMPLING=fDTS/2, N=6</p> <p>1100: 采样频率 fSAMPLING=fDTS/16, N=8</p> <p>0101: 采样频率 fSAMPLING=fDTS/2, N=8</p> <p>1101: 采样频率 fSAMPLING=fDTS/32, N=5</p> <p>0110: 采样频率 fSAMPLING=fDTS/4, N=6</p> <p>1110: 采样频率 fSAMPLING=fDTS/32, N=6</p> <p>0111: 采样频率 fSAMPLING=fDTS/4, N=8</p> <p>1111: 采样频率 fSAMPLING=fDTS/32, N=8</p>
9:8	CKD	RW	00	<p>时钟分频因子</p> <p>死区发生器和数字滤波器所用的采样时钟与定时器时钟 (CK_INT) 的分频比例。</p> <p>00:tDTS=tCK_INT</p> <p>01: tDTS=2 x tCK_INT</p> <p>10: tDTS=4 x tCK_INT</p> <p>11:保留</p>
7	ARPE	RW	0	<p>自动重装载预装载允许位</p> <p>0:TIMx_ARR 寄存器没有缓冲</p> <p>1:TIMx_ARR 寄存器被装入缓冲器</p>
6:4	-	-	-	保留, 读始终为 0。

3	OPM	RW	0	<p>单脉冲模式</p> <p>0: 在发生更新事件时, 计数器不停止;</p> <p>1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。</p>
2	URS	RW	0	<p>更新请求源</p> <p>软件通过该位选择 UEV 事件的源</p> <p>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:</p> <ul style="list-style-type: none"> <li>- 计数器溢出/下溢</li> <li>- 设置 UG 位</li> <li>- 从模式控制器产生的更新</li> </ul> <p>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。</p>
1	UDIS	RW	0	<p>禁止更新</p> <p>软件通过该位允许/禁止 UEV 事件的产生</p> <p>0: 允许 UEV。更新(UEV)事件由下述任一事件产生:</p> <ul style="list-style-type: none"> <li>- 计数器溢出/下溢</li> <li>- 设置 UG 位</li> <li>- 从模式控制器产生的更新</li> </ul> <p>具有缓存的寄存器被装入它们的预装载值。(译注: 更新影子寄存器)</p> <p>1: 禁止 UEV。不产生更新事件, 影子寄存器 (ARR、PSC、CCR<sub>x</sub>)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。</p>
0	CEN	RW	0	<p>使能计数器</p> <p>0: 禁止计数器;</p> <p>1: 使能计数器。</p> <p>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。</p>

## 13.4.2. TIMx 控制寄存器 2 (TIMx\_CR2 偏移: 0x04)

比特	名称	属性	复位值	描述
15:11	-	-	-	保留, 始终读为 0。
10	OIS2	RW	0	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	RW	0	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	0	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	000	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

				<p>输出送出一个正脉冲(TRGO)。</p> <p>100: 比较 – OC1REF 信号被用于作为触发输出 (TRGO)。</p> <p>101: 比较 – OC2REF 信号被用于作为触发输出 (TRGO)。</p> <p>110: 比较 – OC3REF 信号被用于作为触发输出 (TRGO)。</p> <p>111: 比较 – OC4REF 信号被用于作为触发输出 (TRGO)。</p>
3	CCDS	RW	0	<p>捕获/比较的 DMA 选择 (Capture/compare DMA selection)</p> <p>0: 当发生 CCx 事件时, 送出 CCx 的 DMA 请求;</p> <p>1: 当发生更新事件时, 送出 CCx 的 DMA 请求。</p>
2	CCUS	RW	0	<p>捕获/比较控制更新选择 (Capture/compare control update selection)</p> <p>0: 如果捕获/比较控制位是预装载的(CCPC=1), 只能通过设置 COM 位更新它们;</p> <p>1: 如果捕获/比较控制位是预装载的(CCPC=1), 可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注: 该位只对具有互补输出的通道起作用。</p>
1	-	-	-	保留, 始终读为 0。
0	CCPC	RW	0	<p>捕获/比较预装载控制位 (Capture/compare preloaded control)</p> <p>0: CCxE, CCxNE 和 OCxM 位不是预装载的;</p> <p>1: CCxE, CCxNE 和 OCxM 位是预装载的;</p> <p>设置该位后, 它们只在设置了 COM 位后被更新。</p>

### 13.4.3. 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
15:8	-	-	-	保留, 始终读为 0。
7	MSM	RW	0	<p>主/从模式 (Master/slave mode)</p> <p>0: 无作用;</p> <p>1: 触发输入(TRGI)上的事件被延迟了, 以允许在当</p>

				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	RW	000	<p>触发选择 (Trigger selection)</p> <p>这 3 位选择用于同步计数器的触发输入。</p> <p>000: 保留</p> <p>100: TI1 的边沿检测器(TI1F_ED)</p> <p>001: 内部触发 1(ITR1)</p> <p>101: 滤波后的定时器输入 1(TI1FP1)</p> <p>010: 内部触发 2(ITR2)</p> <p>110: 滤波后的定时器输入 2(TI2FP2)</p> <p>011: 内部触发 3(ITR3)</p> <p>111: 保留</p> <p>注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。</p>
3	-	-	-	保留, 始终读为 0。
2:0	SMS	RW	000	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明)</p> <p>000: 关闭从模式 – 如果 CEN=1, 则预分频器直接由内部时钟驱动。</p> <p>001: 保留</p> <p>010: 保留</p> <p>011: 保留</p> <p>100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。</p> <p>101: 门控模式 – 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。</p> <p>110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。</p> <p>111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。</p> <p>注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不</p>

				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。
--	--	--	--	--

表格 13-1 TIM15 从模式互联：ITRx

ITR0	TIM2_TRGO
ITR1	TIM3_TRGO
ITR2	TIM16_OC1
ITR3	TIM17_OC1

#### 13.4.4. 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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

### 13.4.5. TIMx 状态寄存器 (TIMx\_SR 偏移: 0x10)

比特	名称	属性	复位值	描述
15:11	-	-	-	保留, 始终读为 0。
10	CC2OF	W0C	0	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	W0C	0	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	0	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	0	<p>触发器中断标记 (Trigger interrupt flag)</p> <p>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'。</p> <p>0: 无触发器事件产生;</p> <p>1: 触发中断等待响应。</p>
5	COMIF	W0C	0	<p>COM 中断标记 (COM interrupt flag)</p> <p>一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置'1'。它由软件清'0'。</p> <p>0: 无 COM 事件产生;</p> <p>1: COM 中断等待响应。</p>
4:3	-	-	-	保留, 始终读为 0。
2	CC2IF	W0C	0	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	W0C	0	<p>捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag)</p> <p>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;</p> <p>1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。</p> <p>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</p> <p>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'。</p> <p>0: 无输入捕获产生;</p> <p>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)。</p>
0	UIF	W0C	0	<p>更新中断标记 (Update interrupt flag)</p> <p>当产生更新事件时该位由硬件置'1'。它由软件</p>

			<p>清'0'。</p> <p>0: 无更新事件产生;</p> <p>1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1':</p> <ul style="list-style-type: none"> <li>- 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)。</li> <li>- 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。</li> <li>- 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。</li> </ul>
--	--	--	---

## 13.4.6. 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	0	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	0	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	0	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	0	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0	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	0	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。

### 13.4.7. 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。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

比特	名称	属性	复位值	描述
15	-	-	-	保留，读始终为 0。
14:12	OC2M	RW	000	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	RW	0	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	RW	0	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S	RW	00	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，读始终为 0。
6:4	OC1M	RW	000	<p>输出比较 1 模式 (Output Compare 1 mode)</p> <p>该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。</p> <p>000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；</p> <p>001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为高。</p> <p>010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为低。</p> <p>011: 翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。</p> <p>100: 强制为无效电平。强制 OC1REF 为低。</p> <p>101: 强制为有效电平。强制 OC1REF 为高。</p> <p>110: PWM 模式 1— 在向上计数时，一旦 TIMx_CNT&lt;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&gt;TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0)，否则为有效电平(OC1REF=1)。</p> <p>111: PWM 模式 2— 在向上计数时，一旦 TIMx_CNT&lt;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&gt;TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。</p> <p>注 1: 一旦 LOCK 级别设为 3(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。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。</p>
3	OC1PE	RW	0	输出比较 1 预装载使能 (Output Compare 1 preload

				<p>enable)</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p>
2	OC1FE	RW	0	<p>输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择。(Capture/Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: CC1 通道被配置为输入, IC1 映射在 TRC 上。</p> <p>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)。</p> <p>注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。</p>

输入捕获模式:

比特	名称	属性	复位值	描述
15:12	IC2F	RW	000	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC	RW	00	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S	RW	00	<p>捕获/比较 2 选择 (Capture/Compare 2 selection)</p> <p>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;</p> <p>01: CC2 通道被配置为输入, IC2 映射在 TI2 上;</p> <p>10: CC2 通道被配置为输入, IC2 映射在 TI1 上;</p> <p>11: CC2 通道被配置为输入, IC2 映射在 TRC 上。</p> <p>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)。</p> <p>注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。</p>
7:4	IC1F	RW	0000	<p>输入捕获 1 滤波器 (Input capture 1 filter)</p> <p>这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变:</p> <p>0000: 无滤波器, 以 fDTS 采样</p> <p>1000: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, N=6</p> <p>0001: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=2</p> <p>1001: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, N=8</p> <p>0010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=4</p> <p>1010: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=5</p> <p>0011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=8</p> <p>1011: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=6</p> <p>0100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, N=6</p> <p>1100: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=8</p> <p>0101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, N=8</p> <p>1101: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=5</p> <p>0110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, N=6</p> <p>1110: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=6</p> <p>0111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, N=8</p> <p>1111: 采样频率 <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=8</p>
3:2	IC1PSC	RW	00	<p>输入/捕获 1 预分频器 (Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦</p>

				<p>CC1E=0(TIMx_CCER 寄存器中), 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每 2 个事件触发一次捕获;</p> <p>10: 每 4 个事件触发一次捕获;</p> <p>11: 每 8 个事件触发一次捕获。</p>
1:0	CC1S	RW	00	<p>捕获/比较 1 选择 (Capture/Compare 1 Selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: CC1 通道被配置为输入, IC1 映射在 TRC 上。</p> <p>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)。</p>

#### 13.4.8. 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
15:8	-	-	-	保留, 始终读为 0。
7	CC2NP	RW	0	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	0	输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	RW	0	输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	RW	0	<p>输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity)</p> <p>0: OC1N 高电平有效;</p> <p>1: OC1N 低电平有效。</p> <p>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。</p> <p>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)</p>

				则该位不能被修改。
2	CC1NE	RW	0	<p>输入/捕获 1 互补输出使能 (Capture/Compare 1 complementary output enable)</p> <p>0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。</p> <p>1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。</p>
1	CC1P	RW	0	<p>输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出:</p> <p>0: OC1 高电平有效;</p> <p>1: OC1 低电平有效。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>10: 保留, 不使用此配置。</p> <p>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)</p> <p>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。</p>
0	CC1E	RW	0	<p>输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出:</p> <p>0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>1: 开启— OC1 信号输出到对应的输出引脚, 其输</p>

				<p>出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>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。</p> <p>0：捕获禁止；</p> <p>1：捕获使能。</p>
--	--	--	--	---

表 13-1 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

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

					OCxN_EN=1	
		1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
0	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）	
	0		1	0	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	1	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）	
	1		1	0	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	1		

### 13.4.9. TIMx 计数器（TIMx\_CNT 偏移：0x24）

比特	名称	属性	复位值	描述
15:0	CNT	RW	0x0000	计数器的值 (Counter value)



## 13.4.10. 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	0x0000	<p>预分频器的值 (Prescaler value)</p> <p>计数器的时钟频率(CK_CNT)等于 <math>f_{CK\_PSC}/(PSC[15:0]+1)</math>。</p> <p>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'</p>

## 13.4.11. TIMx 自动重载寄存器 (TIMx\_ARR 偏移: 0x2C)

比特	名称	属性	复位值	描述
15:0	ARR	RW	0x0000	<p>自动重载的值 (Auto Reload value)</p> <p>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。</p>

## 13.4.12. 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	0x00	<p>重复计数器的值 (Repetition counter value)</p> <p>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:</p> <ul style="list-style-type: none"> <li>— 在边沿对齐模式下, PWM 周期的数目;</li> <li>— 在中心对称模式下, PWM 半周期的数目;</li> </ul>

## 13.4.13. 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	0x0000	<p>捕获/比较通道 1 的值 (Capture/Compare 1 value)</p> <p>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。</p>

## 13.4.14. 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	0x0000	<p>捕获/比较通道 2 的值 (Capture/Compare 2 value)</p> <p>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。</p>

## 13.4.15. 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	0	<p>主输出使能 (Main output enable)</p> <p>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。</p> <p>0: 禁止 OC 和 OCN 输出或强制为空闲状态;</p> <p>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。</p>
14	AOE	RW	0	<p>自动输出使能 (Automatic output enable)</p> <p>0: MOE 只能被软件置'1';</p> <p>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)。</p> <p>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。</p>
13	BKP	RW	0	<p>刹车输入极性 (Break polarity)</p> <p>0: 刹车输入低电平有效;</p> <p>1: 刹车输入高电平有效。</p> <p>注: 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。</p> <p>注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。</p>
12	BKE	RW	0	<p>刹车功能使能 (Break enable)</p> <p>0: 禁止刹车输入(BRK 及 CCS 时钟失效事件);</p> <p>1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。</p> <p>注: 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。</p> <p>注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。</p>
11	OSSR	RW	0	<p>运行模式下“关闭状态”选择 (Off-state selection for Run mode)</p> <p>该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明。</p>

				<p>0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0);</p> <p>1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, 首先开启 OC/OCN 并输出无效电平, 然后置 OC/OCN 使能输出信号=1。</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。</p>
10	OSSI	RW	0	<p>空闲模式下“关闭状态”选择 (Off-state selection for Idle mode)</p> <p>该位用于当 MOE=0 且通道设为输出时。参考 OC/OCN 使能的详细说明。</p> <p>0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0);</p> <p>1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, OC/OCN 首先输出其空闲电平, 然后 OC/OCN 使能输出信号=1。</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。</p>
9:8	LOCK	RW	00	<p>锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。</p> <p>00: 锁定关闭, 寄存器无写保护;</p> <p>01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位;</p> <p>10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出, CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位;</p> <p>11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出, CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位);</p> <p>注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIMx_BDTR 寄存器, 则其内容冻结直至复位。</p>
7:0	UTG	RW	0x00	死区发生器设置 (Dead-time generator setup)

			<p>这些位定义了插入互补输出之间的死区持续时间。 假设 DT 表示其持续时间： DTG[7:5]=0xx =&gt; DT=DTG[7:0] × Tdtg, Tdtg = TDS; DTG[7:5]=10x =&gt; DT=(64+DTG[5:0]) × Tdtg, Tdtg = 2 × TDS; DTG[7:5]=110 =&gt; DT=(32+DTG[4:0]) × Tdtg, Tdtg = 8 × TDS; DTG[7:5]=111 =&gt; DT=(32+DTG[4:0]) × Tdtg, Tdtg = 16 × TDS;</p> <p>例：若 TDS = 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；</p> <p>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。</p>
--	--	--	---

### 13.4.16. 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	00000	<p>DMA 连续传送长度 (DMA burst length)</p> <p>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</p> <p>例：我们考虑这样的传输：DBL=7， DBA=TIM2_CR1 - 如果 DBL=7，DBA=TIM2_CR1</p>

				<p>表示待传输数据的地址，那么传输的地址由下式给出：<math>(TIMx\_CR1 \text{ 的地址}) + DBA + (DMA \text{ 索引})</math>，其中 <math>DMA \text{ 索引} = DBL</math> 其中<math>(TIMx\_CR1 \text{ 的地址}) + DBA</math> 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<math>(TIMx\_CR1 \text{ 的地址}) + DBA</math> 开始的 7 个寄存器。</p> <p>根据 DMA 数据长度的设置，可能发生以下情况：</p> <ul style="list-style-type: none"> <li>- 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。</li> <li>- 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。</li> </ul>
7:5	-	-	-	位 7:5 保留，始终读为 0。
4:0	DBA	RW	00000	<p>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：</p> <p>00000: TIMx_CR1,  00001: TIMx_CR2,  00010: TIMx_SMCR,  ...</p>

## 13.4.17. 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
15:0	DMAB	RW	0x0000	<p>DMA 连续传送寄存器 (DMA register for burst accesses)</p> <p>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, 它取决于 TIMx_DCR 寄存器中定义的 DBL。</p>

## 13.4.18. TIMx 复用功能选择寄存器 (TIMx\_AF1 偏移: 0x60)

比特	名称	属性	复位值	描述
15:14	ETRSEL	RW	00	ETR 输入源选择 00: GPIO 01: COMP1 10: COMP2 11: AWD
13:12	-	-	-	保留, 始终为 0。
11	BKCOMP2P	RW	0	比较器 2 输入极性控制 0: 不反相 1: 被反相
10	BKCOMP1P	RW	0	比较器 1 输入极性控制 0: 不反相 1: 被反相
9	BKINP	RW	0	刹车输入极性控制 0: 不反相 1: 被反相
8:3	-	-	-	保留, 始终为 0。
2	BKCOMP2E	RW	0	比较器 2 输入使能控制 0: 禁止 1: 使能
1	BKCOMP1E	RW	0	比较器 1 输入使能控制 0: 禁止 1: 使能
0	BKINE	RW	0	刹车输入使能控制 0: 禁止 1: 使能

## 13.4.19. TIMx 输入选择寄存器 (TIMx\_TISEL 偏移: 0x68)

比特	名称	属性	复位值	描述
15:9	-	-	-	保留, 始终为 0。
8	T2SEL	RW	0	TI2 输入选择

				0: TIM_CH2 1: COMP2
7:1	-	-	-	保留, 始终为 0。
0	T1SEL	RW	0	TI1 输入选择 0: TIM_CH1 1: COMP1

### 13.4.20. 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	0	触发事件的 DMA 请求类型 0: Single; 1: Burst;
5	COMBE	RW	0	COM 事件的 DMA 请求类型 0: Single; 1: Burst;
4	CC4BE	RW	0	捕获/比较 4 事件的 DMA 请求类型 0: Single; 1: Burst;
3	CC3BE	RW	0	捕获/比较 3 事件的 DMA 请求类型 0: Single; 1: Burst;
2	CC2BE	RW	0	捕获/比较 2 事件的 DMA 请求类型 0: Single; 1: Burst;
1	CC1BE	RW	0	捕获/比较 1 事件的 DMA 请求类型 0: Single; 1: Burst;
0	UBE	RW	0	更新事件的 DMA 请求类型 0: Single; 1: Burst;

## 14. 看门狗（WDT）

### 14.1. 概述

看门狗模块（WatchDog TIMER）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。

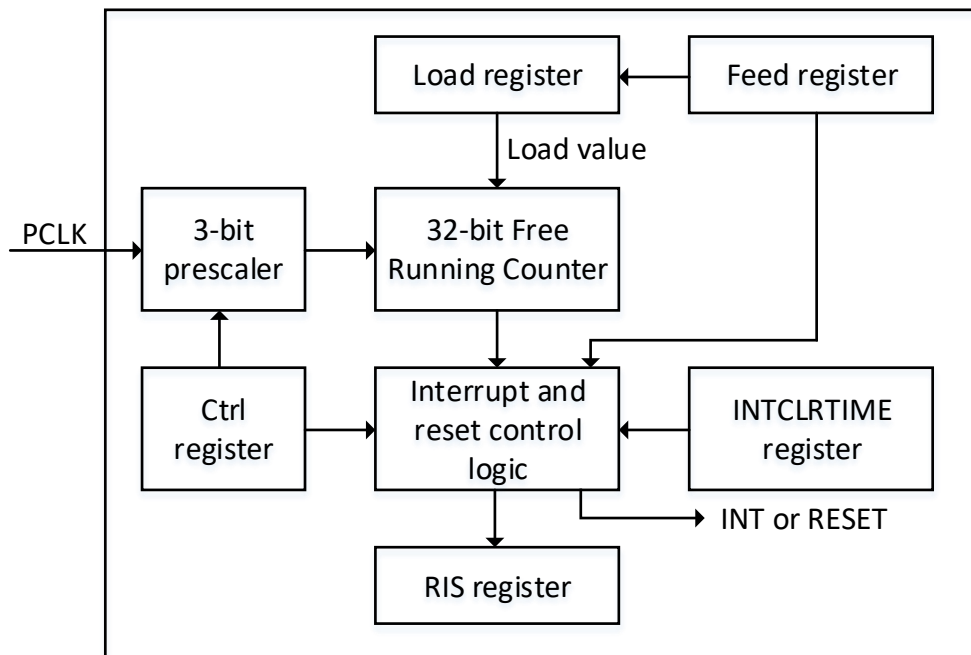
### 14.2. 主要特性

- 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

## 14.3. 功能描述

### 14.3.1. 功能框图

图 14-1 WDT 功能框图



### 14.3.2. 看门狗模式

看门狗模块可以通过设置控制寄存器 WDTCTRL 的 MODE 位选择看门狗模式，包括：

- 中断模式：当计数器计数减为 0，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。
- 复位模式：当计数器计数减为 0，则看门狗模块输出复位信号，复位系统。

### 14.3.3. 中断清除时限

看门狗模块可以通过设置一个 32 位的中断清除时限寄存器 WDTINTCLRTIME 配置中断清除时限，当看门狗模块工作在中断模式时，当计数器计数减为 0，会触发看门狗中断(即使中断未使能)，若在中断清除时限寄存器定义的时间内没有去喂狗，WDT 将会产生复位信号。

## 14.4. 寄存器描述

WDT 寄存器基地址：0x4000\_2C00

偏置	名称	描述
0x00	WDTLOAD	加载寄存器
0x04	WDTCOUNT	当前计数寄存器
0x08	WDTCTRL	控制寄存器
0x0C	WDTFEED	喂狗寄存器
0x10	WDTINTCLRTIME	中断清除时限寄存器
0x14	WDTRIS	原始中断状态寄存器

### 14.4.1. WDTLOAD 加载寄存器(偏移：00h)

比特	名称	属性	复位值	描述
31:0	LOAD	RW	0xFFFFFFFF	32 位寄存器，用来存入加载值。

### 14.4.2. WDTCOUNT 当前计数寄存器(偏移：04h)

比特	名称	属性	复位值	描述
31: 0	COUNT	RO	0xFFFFFFFF	32 位寄存器，看门狗当前计数值。

### 14.4.3. WDTCTRL 控制寄存器(偏移：08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	EN	RW	0	看门狗使能位：0 禁止，1 使能。
6	MODE	RW	0	看门狗模式位：0 复位，1 中断。
5	RSV	-	-	保留
4	INTEN	RW	0	看门狗中断使能位：0 禁止，1 使能。
3	RSV	-	-	保留
2:0	DIVISOR	RW	110	看门狗时钟预分频：

				3'b000: 不分频 3'b001: 2 分频 3'b010: 4 分频 3'b011: 8 分频 3'b100: 16 分频 3'b101: 32 分频 3'b110: 64 分频 3'b111: 128 分频
--	--	--	--	--

#### 14.4.4. WDTFEED 喂狗寄存器(偏移: 0Ch)

比特	名称	属性	复位值	描述
31:0	FEED	WO	0x0	32 位寄存器，喂狗寄存器，写入特征值 0xAA55A55A 使计数器从装载寄存器加载装载值，加载动作会清除 WDT 中断。

#### 14.4.5. WDTINTCLRTIME 中断清除时限寄存器 (偏移: 10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	INTCLRT	RW	0x1000	32 位寄存器，中断清除时限寄存器，中断产生后(即使中断未使能)，在这个寄存器定义的时间内没有去喂狗，WDT 会产生复位信号。

#### 14.4.6. WDTRIS 原始中断状态寄存器(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。

## 14.5. 使用流程

### 14.5.1. 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。

### 14.5.2. 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。

## 15. 独立看门狗（IWDT）

### 15.1. 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。

### 15.2. 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

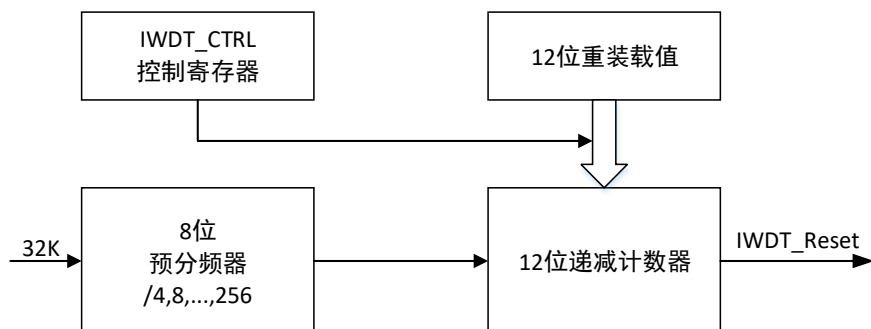
### 15.3. 功能描述

#### 15.3.1. 功能框图

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 0xFFF 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。

IWDT 整体框图如下：

图 15-1 IWDT 框图



### 15.3.2. 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。写打开后可以一次对多个受保护寄存器改写。以不同的值写入这个寄存器将会中断操作流程，寄存器将重新被保护。重载操作(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。

### 15.3.3. 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，重载向下计数器可以避免复位。

IWDT\_WINR 的默认值为 0xFFFF，与重载值 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。

### 15.3.4. 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 RPMU 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。

## 15.4. 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

### 15.4.1. IWDT\_CMDR 命令寄存器(偏移：00h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CMD	WO	0x0	独立看门狗定时器命令 只写寄存器 软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。 写入 0x5555 表示允许访问 IWDT_PR、IWDT_RLR 和 IWDT_WINR 寄存器。 写入 0xCCCC，启动看门狗工作。 写入 0xEF01ABCD，关闭看门狗工作。 写入 0x6666，使能唤醒功能 写入 0x9999，禁止唤醒功能 注：请在启动看门狗后再配置寄存器

## 15.4.2. IWDT\_PR 预分频寄存器(偏移: 04h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	PR	RW	000	<p>独立看门狗定时器预分频因子 (Prescaler divider)</p> <p>这些位具有写保护设置。通过设置这些位来选择计数器时钟的预分频因子。要改变预分频因子, IWDT_SR 寄存器的 PVU 位必须为 0。</p> <p>3'b000: 4 分频 3'b001: 8 分频 3'b010: 16 分频 3'b011: 32 分频 3'b100: 64 分频 3'b101: 128 分频 3'b110: 256 分频 3'b111: 256 分频</p> <p>注意: 如果写操作正在进行, 则读回的值可能是无效的。因此, 只有当 IWDT_SR 寄存器的 PVU 位为 0 时, 读出的值才有效。</p>

## 15.4.3. IWDT\_RLR 重装载寄存器(偏移: 08h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	PL	RW	0xFFFF	<p>独立看门狗定时器重装载值</p> <p>这些位具有写保护功能。用于定义看门狗计数器的重装载值, 每当向 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。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。</p> <p>只有当 IWDT_SR 寄存器中的 RVU 位为 0 时, 才能对此寄存器进行修改。</p>

				注：如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDT_SR 寄存器的 RVU 位为 0 时，读出的值才有效。这些位有写保护功能。
--	--	--	--	---

#### 15.4.4. IWDT\_SR 状态寄存器(偏移：0Ch)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
4	RLF	RO	0	独立看门狗定时器喂狗完成 此位由硬件置 1 用来指示喂狗正在进行中。当喂狗结束后，此位由硬件清 0。 关闭 IWDT 时钟或者进入 STOP 模式前，需要等待完成。
3	WTU	RO	0	独立看门狗定时器唤醒值更新 此位由硬件置 1 用来指示唤醒值的更新正在进行中。当唤醒值更新结束后，此位由硬件清 0（最多需 5 个 32kHz 时钟周期）。唤醒值只有在 WTU 位被清 0 后才可更新。
2	WVU	RO	0	独立看门狗定时器窗口值更新 此位由硬件置 1 用来指示窗口值的更新正在进行中。当窗口值更新结束后，此位由硬件清 0（最多需 5 个 32kHz 时钟周期）。窗口值只有在 WVU 位被清 0 后才可更新。
1	RVU	RO	0	独立看门狗定时器重装载值更新 此位由硬件置 1 用来指示重装载值的更新正在进行中。当重装载更新结束后，此位由硬件清 0（最多需 5 个 32kHz 时钟周期）。重装载值只有在 RVU 位被清 0 后才可更新。
0	PVU	RO	0	独立看门狗定时器预分频值更新 此位由硬件置 1 用来指示预分频值的更新正在进行中。当预分频值更新结束后，此位由硬件清 0(最多需 5 个 32kHz 时钟周期)。预分频值只有在 PVU 位被清 0 后才可更新。

## 15.4.5. IWDT\_WINR 窗口寄存器(偏移: 10h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	WIN	RW	0xFFFF	<p>独立看门狗定时器窗口值</p> <p>这些位有写保护功能。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[11:0]中值，重装载操作会引起复位。只有当 IWDT_SR 寄存器中的 WVU 位为 0 时，才能对此寄存器进行修改。</p> <p>注：如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDT_SR 寄存器的 WVU 位为 0 时，读出的值才有效。</p> <p>在写这些位之前需向 IWDT_CMDR 寄存器中写 0x5555。</p>

## 15.4.6. IWDT\_WUTR 窗口寄存器(偏移: 14h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	WUT	RW	0xFFFF	<p>独立看门狗定时器唤醒值</p> <p>这些位有写保护功能。定义唤醒值，用于定义唤醒信号产生的计数值，只产生一个预分频周期的 WakeUp 信号。只有当 IWDT_SR 寄存器中的 WTU 位为 0 时，才能对此寄存器进行修改。</p> <p>注：如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDT_SR 寄存器的 WTU 位为 0 时，读出的值才有效。</p> <p>在写这些位之前需向 IWDT_CMDR 寄存器中写 0x5555。</p>

## 15.5. 使用流程

### 15.5.1. 不带窗口

1. 对 IWDT\_CMDR 写入 0xCCCC 使能 IWDT
2. 对 IWDT\_CMDR 写入 0x5555 打开写保护
3. 写 IWDT\_PR 配置预分频值 4~256
4. 写 IWDT\_RLR 配置重装载值
5. 等待 IWDT\_SR 为 0x00000000
6. 对 IWDT\_CMDR 写入 0xAAAA 进行重装载操作

### 15.5.2. 带窗口

1. 对 IWDT\_CMDR 写入 0xCCCC 使能 IWDT
2. 对 IWDT\_CMDR 写入 0x5555 打开写保护
3. 写 IWDT\_PR 配置预分频值 4~256
4. 写 IWDT\_RLR 配置重装载值
5. 等待 IWDT\_SR 为 0x00000000
6. 写 IWDT\_WINR 配置窗口值，会自动进行重装载操作

## 16. 实时时钟（RTC）

### 16.1. 概述

RTC 提供年、月、日、周、时、分、秒的信息，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，用以达到更高的日历精度。

### 16.2. 主要特性

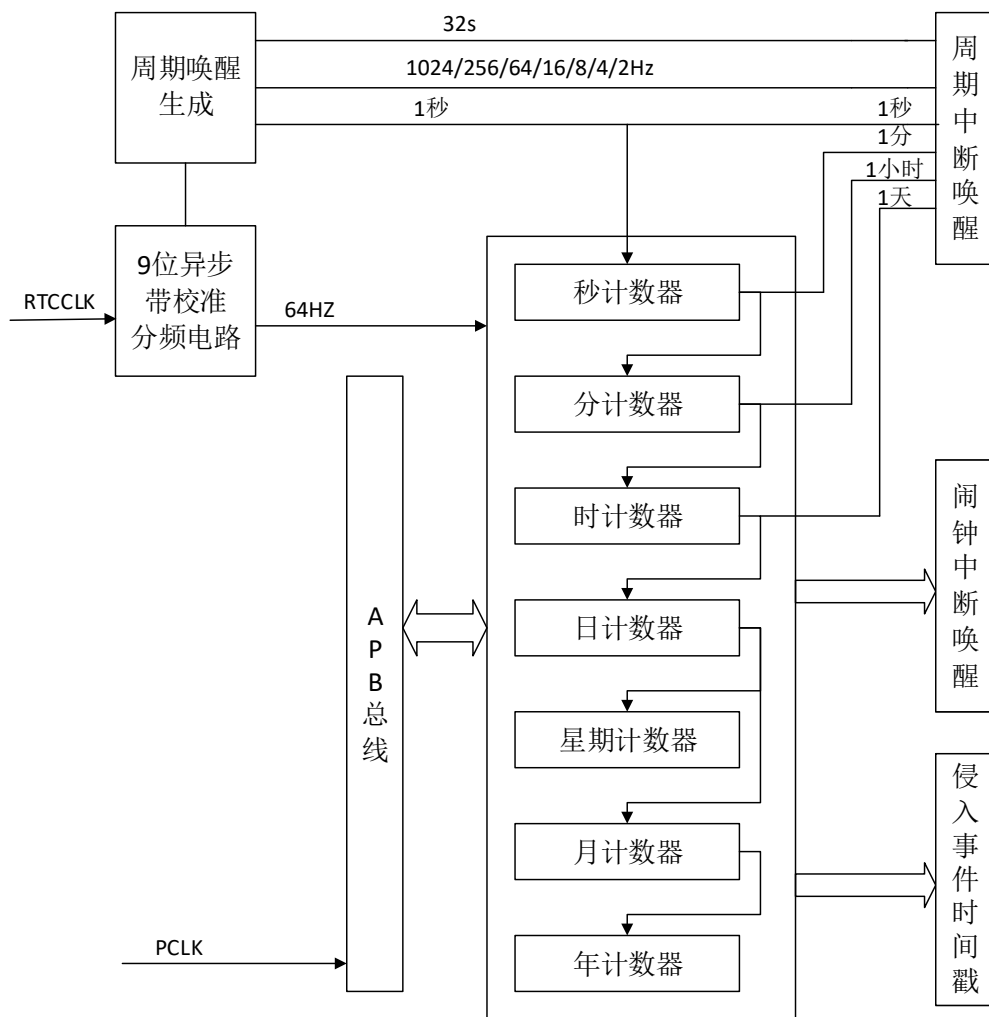
- 可计算 00~99 年之间的年、月、日、周、时、分、秒
- BCD 时间显示
- 自动闰年调整
- 数字校准功能：通过调整最小时间单位（最大可调精度 0.95ppm）来进行日历校准，调校后理论精度 $\pm 0.477\text{ppm}$
- 周期唤醒中断
- 闹钟功能
- 可从管脚输出 XTLF 时钟信号供用户校准
- 2 路输入上下沿(侵入)时间戳功能
- 5 个 32 位 (共 20 字节) 通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位

### 16.3. 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 16-1 RTC 内部结构框图



### 16.3.1. 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：

- 1、等待秒中断时间发生
- 2、打开写保护
- 3、连续写入年月日时分秒寄存器
- 4、读出时间寄存器进行校验

## 5、使能写保护

注：年、月、日、周、时、分、秒不会被复位

### 16.3.2. RTC 时间读取

#### 方式 1：任意时刻读取方式

- 1、读出年、月、日、周、时、分、秒计数寄存器值。
- 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。

### 16.3.3. 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、CR.ALM\_EN=0，闹钟关闭。
- 2、设定 ALM\_IE=1，闹钟中断禁止。
- 3、周闹钟 ALMWEEK 设定，设置每周几闹钟工作。秒闹钟 ALMSEC、分闹钟 ALMMIN、时闹钟 ALMHOUR。
- 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。

### 16.3.4. 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內。

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： $\text{Correction}(\text{ppm}) = (C * 32 + D) * 30.517 / 32000000$

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。

### 16.3.5. 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。

### 16.3.6. 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。

### 16.3.7. 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。

## 16.4. 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_DATE	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_CLKSR1	上升沿时间戳 1
0x48	RTC_CALSR1	上升沿日历戳 1
0x4C	RTC_CLKSF1	下降沿时间戳 1
0x50	RTC_CALPF1	下降沿日历戳 1
0x54	RTC_CLKSR2	上升沿时间戳 2
0x58	RTC_CALSR2	上升沿日历戳 2
0x5C	RTC_CLKSF2	下降沿时间戳 2
0x60	RTC_CALPF2	下降沿日历戳 2
0x70~80	RTC_BAKUP0~4	备份寄存器 0 ~ 4

## 16.4.1. RTC\_WP 写保护寄存器(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。

## 16.4.2. RTC\_IE 中断使能寄存器(偏移: 04h)

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
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

### 16.4.3. RTC\_SR 中断标志寄存器(偏移: 08h)

比特	名称	属性	复位值	描述
----	----	----	-----	----

31:12	RSV	-	-	保留
16	STP2RIF	RC_W1	0	RTC TAMP2 上升沿事件中断标志 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 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 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 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

#### 16.4.4. RTC\_SEC 秒计数寄存器(偏移: 0Ch)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
6:0	BCDSEC	RW	X	秒时间数值, BCD 格式。

#### 16.4.5. RTC\_MIN 时计数寄存器(偏移: 10h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留

6:0	BCDMIN	RW	X	分钟时间数值，BCD 格式。
-----	--------	----	---	----------------

#### 16.4.6. RTC\_HOUR 分计数寄存器(偏移: 14h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:0	BCD HOUR	RW	X	小时数值，BCD 格式。

#### 16.4.7. RTC\_DAY 日计数寄存器(偏移: 18h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
5:0	BCDDATE	RW	X	天数数值，BCD 格式。

#### 16.4.8. RTC\_WEEK 周计数寄存器(偏移: 1Ch)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
2:0	BCDWEEK	RW	X	周数值，BCD 格式。

#### 16.4.9. RTC\_MONTH 月计数寄存器(偏移: 20h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
4:0	BCDMONTH	RW	X	月数值，BCD 格式。

## 16.4.10. RTC\_YEAR 年计数寄存器(偏移: 24h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
7:0	BCDYEAR	RW	X	年数值, BCD 格式。

## 16.4.11. RTC\_ALM 闹钟寄存器(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	ALMWEEK/ ALMDAY	RW	0x7F	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	0x0	闹钟的小时 BCD 格式。
15	RSV	-	-	保留
14:8	ALMMIN	RW	0x0	闹钟的分 BCD 格式。
7	RSV	-	-	保留
6:0	ALMSEC	RW	0x0	闹钟的秒 BCD 格式

## 16.4.12. RTC\_CR 控制寄存器(偏移: 2Ch)

比特	名称	属性	复位值	描述
31:23	RSV	-	-	保留
22	TAMPFLTCLK	RW	1	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_MSKEH	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	RW	0	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: RFU 4'b0001: RFU 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: RFU 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: RFU 4'b1111: 输出 RTC 内部秒时标方波
--	--	--	--	---

#### 16.4.13. RTC\_ADJUST 时钟误差补偿寄存器(偏移: 30h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	ADJSIGN	RW	0	补偿调整方向 0: 增加 1: 减少
8:0	ADJVALUE	RW	0x0	补偿调整数值

#### 16.4.14. RTC\_CLKSTAMP1 时间戳 1(偏移: 44h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
21:16	HRSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 秒寄存器的值

#### 16.4.15. RTC\_CALSTAMP1 日历戳 1(偏移: 48h)

比特	名称	属性	复位值	描述
31:21	YEARSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留

10:8	WKSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP1	RO	0x0	检测到 TAMP1 后存储 BCD 日寄存器的值

#### 16.4.16. RTC\_CLKSTAMP2 时间戳 2(偏移: 4Ch)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
21:16	HRSTP2	RO	0x0	检测到 TAMP2 后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP2	RO	0x0	检测到 TAMP2 后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP2	RO	0x0	检测到 TAMP2 后存储 BCD 秒寄存器的值

#### 16.4.17. RTC\_CALSTAMP2 日历戳 2(偏移: 50h)

比特	名称	属性	复位值	描述
31:24	YEARSTP2	RO	0x0	检测到 TAMP1 后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP2	RO	0x0	检测到 TAMP2 后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP2	RO	0x0	检测到 TAMP2 后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP2	RO	0x0	检测到 TAMP2 后存储 BCD 日寄存器的值

#### 16.4.18. RTC\_BAKUP0~4 备份寄存器 0~4(偏移: 70~80h)

比特	名称	属性	复位值	描述
31:0	BAKUP	RW	0x0	备份寄存器



## 17. 待机区电源管理（RPMU）

### 17.1. 概述

待机区域的模拟和电源管理由 RPMU(RTC PMU)模块完成。RPMU 模块主要负责 LPBGR (Low power bandgap reference) 的 TRIM, XTL 和 RC32K 两个低速时钟的管理, BOR (Brown out reset) 控制, 待机区管脚 PC13/PC14/PC15 的控制以及 STANDBY 唤醒管理。

STANDBY 模式唤醒主要有四种唤醒源: WKUP 管脚的上升或下降沿, RTC 的周期唤醒/闹钟/时间戳事件, RSTN 管脚复位以及 IWDT 复位。唤醒后, 芯片产生一次系统复位; 复位后, Core 从 0x00 地址重新取值。

WKUP 管脚使用上升或下降沿唤醒, 因此对于唤醒使能的 WKUP 管脚, 板级建议增加上拉/下拉电阻, 防止悬空。如果待机模式下 RTCCLK 存在, 可以开启数字滤波功能, 连续两拍信号变化相同才认为 WKUP 管脚状态变化。RTC 的唤醒事件选择与 RTC 中断一致, 由 RTC\_IE 中使能, 可以选择周期唤醒 (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), 或者闹钟事件, 或者时间戳功能 (可以配合侵入检测)。RSTN 管脚复位或 IWDT 复位也可以唤醒待机模式。

(注: 当 EWUPx 位使能时, 如果 WKUP 管脚为高, 则立刻会产生一个 WUK 标志。)

### 17.2. 功能描述

#### 17.2.1. 唤醒引脚配置

芯片提供了 6 个唤醒引脚, 并配有唤醒引脚滤波功能。滤波为一拍 RTC Clock。唤醒引脚上升沿、下降沿可选。

表 17-1 唤醒引脚对应表

唤醒源	唤醒引脚
Wakeup1	PA0
Wakeup2	PC13
Wakeup3	PA2
Wakeup4	PC5
Wakeup5	PB5

Wakeup6	PB15
---------	------

### 17.2.2. RTC Clock 管理

RTC 时钟选择控制, RTC 可选时钟片上 RC32K 或片外低速晶振 XTL, 通过 RPMU\_ANACR 使能 RC32K 或 XTL, 通过 RPMU\_CR 寄存器选择 RTC 时钟。

### 17.2.3. RPMU 区引脚控制

PC13、PC14、PC15 引脚属于 RPMU 特殊控制引脚, RPMU\_IOSEL、RPMU\_IOCRR 设置优先级高于 GPIO 寄存器和系统寄存器。

若 RPMU\_IOSEL 寄存器配置中不做主区 GPIO 功能, 则以 RPMU\_IOSEL 寄存器为最高优先级, 此时系统寄存器中 PCSEL2 中对 PC13、PC14、PC15 的配置将会忽略。

## 17.3. 寄存器描述

RPMU 寄存器基地址: 0x4000\_2400

偏置	名称	描述
0x00	RPMU_CR	RTC PMU 控制寄存器
0x04	RPMU_SR	RTC PMU 状态寄存器
0x08	RPMU_IOSEL	RTC PMU IO 复用寄存器
0x0C	PRMU_IOCRR	RTC PMU IO 控制寄存器
0x10	RPMU_ANACR	RTC PMU 模拟控制寄存器(XTL/RC32K/LPBGR)
0x14	RPMU_CR2	RTC PMU 控制寄存器 2

### 17.3.1. RPMU\_CR/RTC 域控制寄存器(偏移: 00h)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:25	WU6~2FILEN	RW	0	WKUP6~2 管脚滤波使能, 使用 RTCCLK 滤波, 滤波 1 拍毛刺 0: WKUP6~2 管脚滤波不使能

				1: WKUP6~2 管脚滤波使能
24	WU1FILEN	RW	0	WKUP1 管脚滤波使能, 使用 RTCCLK 滤波, 滤波 1 拍毛刺 0: WKUP1 管脚滤波不使能 1: WKUP1 管脚滤波使能
23:22	RSV	-	-	保留
21:17	EWUP6~2	RW	0	WKUP6~2 管脚唤醒功能使能 0: WKUP6~2 管脚唤醒功能不使能 1: WKUP6~2 管脚唤醒功能使能, 滤波功能由 WUXFIL 位选择。
16	EWUP1	RW	0	WKUP1 管脚唤醒功能使能 0: WKUP1 管脚唤醒功能不使能 1: WKUP1 管脚唤醒功能使能, 滤波功能由 WU1FIL 位选择。
15:13	RSV	-	-	保留
12	BORRST_EN	RW	0	BOR 复位使能 0: BOR 复位不使能 1: BOR 复位使能
11:9	WK_TIME	RW	010	RTC 模块 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
8	STB_EN	RW	0	STANDBY 进入使能, 提供进入 STANDBY 模式的保护, 请写此位再进入 DeepSleep 0: 不能进入 STANDBY 模式 1: 允许进入 STANDBY 模式
7	RSV	-	-	保留
6	BDRST	RW	1	RTC 域软复位, 写 0 后下一拍自动变 1 0: 复位 RTC 域 1: 不复位 RTC 域 注: BOR_EN/BOR_CFG/BORRST_EN 不会被复位
5	RTCEN	RW	1	RTCCLK 使能

				0: 不使能 1: 使能
4	RSV	-	-	保留
3:2	RTCSEL	RW	00	RTCCLK 选择 00: RC32K 时钟 01: XTL 时钟 1x: 没有时钟
1	CWUF	WO	0	清除唤醒标志。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
0	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

### 17.3.2. RPMU\_SR/RTC 域状态寄存器(偏移: 04h)

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
15:12	RSV	-	-	保留
13	BORWUF	RO	0	BOR 唤醒标志。通过写 CSBF 位 (RPMU_CR 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
12	IWDTWUF	RO	0	IWDT 唤醒标志。通过写 CSBF 位 (RPMU_CR 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
11	RSTWUF	RO	0	RSTN 唤醒标志。通过写 CSBF 位 (RPMU_CR 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
10	RTCWUF	RO	0	RTC 唤醒标志。通过写 CSBF 位 (RPMU_CR 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
9:7	RSV	-	-	保留
6:2	WUP6-2F	RO	0	WKUP6~2 唤醒标志。WKUP6~2 管脚检测到一个唤醒事件, 可以通过写 CSBF 位 (RPMU_CR 寄存器) 清除 0: 没有 WKUP6~2 唤醒事件 1: WKUP6~2 管脚发生唤醒事件 注: WUPxF 标志如果为高, 进入 STANDBY 后会立刻唤醒
1	WUP1F	RO	0	WKUP1 唤醒标志。WKUP1 管脚检测到一个唤醒事件, 可以通过写 CSBF 位 (RPMU_CR 寄存器) 清除 0: 没有 WKUP1 唤醒事件 1: WKUP1 管脚发生唤醒事件 注: WUP1F 标志如果为高, 进入 STANDBY 后会立刻唤醒
0	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 (RPMU_CR 寄存器) 清除 0: 芯片未进入过 STANDBY 模式 1: 芯片未进入过 STANDBY 模式

### 17.3.3. RPMU\_IOSEL/RTC 域 IO 复用寄存器(偏移: 08h)

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
9	PC14_Value	RW	0	PC14_Value, 具体见 PC14_SEL
8	PC13_Value	RW	0	PC13_Value, 具体见 PC13_SEL

7	RSV	-	-	保留
6:5	PC15_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: 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

#### 17.3.4. RPMU\_IOCRR/RTC 域 IO 控制寄存器(偏移: Ch)

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
22	PC15_ADS	RW	1	PC15 数模选择 0: 配置为数字端口 1: 配置为模拟端口
21	PC15_IE	RW	1	PC15 输入使能 0: 禁止 1: 使能
20	PC15_SMT	RW	1	PC15 施密特使能 0: 禁止 1: 使能
19:18	PC15_STH	RW	0	PC15 输出驱动设置

				具体定义参见 PA0_STH
17	PC15_PD	RW	0	PC15 下拉电阻使能 0: 禁止 1: 使能
16	PC15_PU	RW	1	PC15 上拉电阻使能 0: 禁止 1: 使能
15	RSV	-	-	保留
14	PC14_ADS	RW	1	PC14 数模选择 0: 配置为数字端口 1: 配置为模拟端口
13	PC14_IE	RW	1	PC14 输入使能 0: 禁止 1: 使能
12	PC14_SMT	RW	1	PC14 施密特使能 0: 禁止 1: 使能
11:10	PC14_STH	RW	0	PC14 输出驱动设置 具体定义参见 PA0_STH
9	PC14_PD	RW	0	PC14 下拉电阻使能 0: 禁止 1: 使能
8	PC14_PU	RW	1	PC14 上拉电阻使能 0: 禁止 1: 使能
7	RSV	-	-	保留
6	PC13_ADS	RW	1	PC13 数模选择 0: 配置为数字端口 1: 配置为模拟端口
5	PC13_IE	RW	1	PC13 输入使能 0: 禁止 1: 使能
4	PC13_SMT	RW	1	PC13 施密特使能 0: 禁止 1: 使能

3:2	PC13_STH	RW	0	PC13 输出驱动设置 具体定义参见 PA0_STH
1	PC13_PD	RW	0	PC13 下拉电阻使能 0: 禁止 1: 使能
0	PC13_PU	RW	1	PC13 上拉电阻使能 0: 禁止 1: 使能

### 17.3.5. RPMU\_ANACR/ RTC\_PMU 模拟控制寄存器 (偏移: 10h)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:27	TRIM_Regu18	RW	100	修调 Regu18 的信号, 默认为 100
26	EN_Regu18	RW	0	Regu18 使能信号, 高有效, 默认为低;
25:24	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
23	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。
22:20	LPBGR_TRIM	RW	100	LPBGR 的 TRIM 值
19:16	RSV	-	-	保留
15:10	RC32K_TRIM	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	RC32K 时钟使能 0: RC32K 不使能 1: RC32K 使能
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 低驱动 ..... 11: XTL 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

### 17.3.6. RPMU\_CR2/RTC 域控制寄存器 2(偏移: 14h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:1	WU6~2POL	RW	0	WKUP6~2 管脚唤醒极性选择 0: WKUP6~2 管脚上升沿唤醒 1: WKUP6~2 管脚下沿唤醒
0	WU1POL	RW	0	WKUP1 管脚唤醒极性选择

---

				0: WKUP1 管脚上升沿唤醒 1: WKUP1 管脚下降沿唤醒
--	--	--	--	--------------------------------------

## 18. 通用输入输出接口（GPIO）

### 18.1. 概述

GPIO 包含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。

GPIO 共有三组控制器，GPIO1、GPIO2、GPIO3。在命名规则上 GPIO1 可分为 PA0~PA15 和 PB0~PB15;GPIO2 可分为 PC0~PC15 和 PD0~PD15;GPIO3 可分为 PE0~PE15 和 PF0~PF4。

管脚所对应的功能详见《[航芯 ACM32F403\\_Datasheet](#)》。

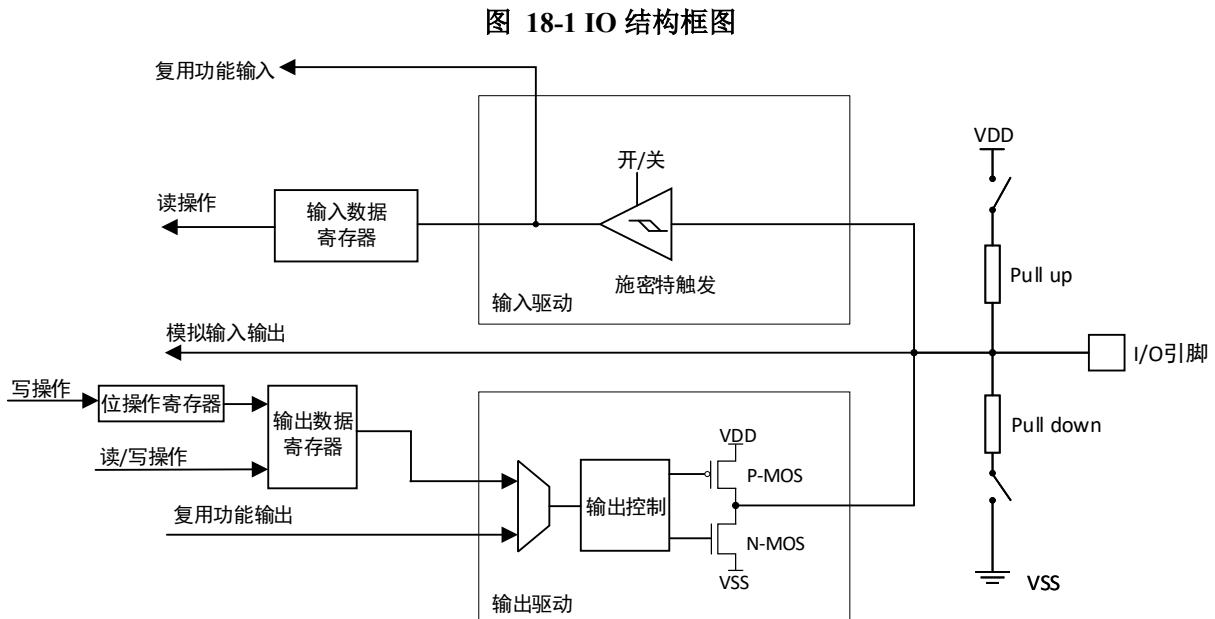
### 18.2. 主要特性

- 每个 GPIO\_IN 引脚可配置成边沿或电平方式触发中断。
- 每个 GPIO 引脚可以由软件配置为输出（推挽或开漏）、输入、外设复用功能或者模拟模式。每个 GPIO 引脚都可以配置为上拉、下拉或无上拉/下拉。

### 18.3. 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。大部分 GPIO 配置为模拟模式（数字功能失效，上下拉电阻失效）。

### 18.3.1. 结构框图



#### 18.3.1.1. 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 GPIO\_IDATA 寄存器中。

#### 18.3.2. 输出功能

- 施密特触发输入使能。
- 可选择弱上拉或弱下拉。
- 可选择推挽输出或开漏输出。
- 可通过配置 GPIO\_SET、GPIO\_CLR 控制 I/O 引脚状态或者通过 GPIO\_ODATA 控制 I/O 引脚状态。

### 18.3.3. 复用功能（AF）

当端口需要配置复用功能时，通过配置系统寄存器 **PASEL1**、**PASEL2**、**PBSEL1**、**PBSEL2** 等，设置为相应的复用功能，

端口复用功能分配的介绍详见《航芯 ACM32F403\_Datasheet》。

### 18.3.4. 开漏、驱动能力、上下拉、管脚复用、数字模拟配置

- 通过配置系统寄存器 PA/BODR、PC/DODR、PE/FODR 来使能开漏。
- 通过配置系统寄存器 PASTR、PBSTR、PCSTR、PDSTR、PESTR、PFSTR 来设置驱动能力。
- 通过配置系统寄存器 PA/BPUP、PA/BPDR；PC/DPUP、PC/DPDR；PE/FPUP、PE/FPDR 来设置上下拉电阻。
- 通过配置系统寄存器 PASEL1、PASEL2；PBSEL1、PBSEL2；PCSEL1、PCSEL2；PDSEL1、PDSEL2；PESEL1、PESEL2；PFSEL1 来设置复用功能。
- 通过配置系统寄存器 PA/BADS、PC/DADS、PE/FADS 来设置数字功能或者模拟功能。

### 18.3.5. 附加功能

部分引脚具有附加功能，附加功能主要包括两类：模拟功能和 STANDBY 唤醒功能。模拟功能的引脚，直接通过 PA/BADS、PC/DADS、PE/FADS 来配置即可，无需其他配置。对于 STANDBY 唤醒功能引脚，除 PC13 需要保持数字功能外，其他唤醒引脚无需配置，因为在 STANDBY 模式下主区已经断电。唤醒功能具体见 RPMU 模块。

注意：PC13/PC14/PC15 为 RTC 域引脚，最好不要当作普通的 IO 来使用。

## 18.4. 寄存器描述

GPIO1 寄存器基地址：0x4001\_F000

GPIO2 寄存器基地址：0x4001\_F400

GPIO3 寄存器基地址：0x4001\_F800

偏置	名称	描述
----	----	----

0x00	GPIO_DIR	GPIO 数据方向寄存器
0x08	GPIO_SET	GPIO 输出置位寄存器
0x0C	GPIO_CLR	GPIO 输出清零寄存器
0x10	GPIO_ODATA	GPIO 输出引脚映射寄存器
0x14	GPIO_IDATA	GPIO 输入引脚映射寄存器
0x18	GPIO_IEN	GPIO 中断使能寄存器
0x1C	GPIO_IS	GPIO 中断触发模式寄存器
0x20	GPIO_IBE	GPIO 中断触发模式寄存器
0x24	GPIO_IEV	GPIO 中断触发模式寄存器
0x28	GPIO_IC	GPIO 中断状态清除寄存器
0x2C	GPIO_RIS	GPIO 原始中断状态寄存器
0x30	GPIO_MIS	GPIO 屏蔽后中断状态寄存器

#### 18.4.1. 数据方向寄存器 GPIO\_DIR(偏移: 00h)

比特	名称	属性	复位值	描述
31: 0	GPIO_DIR	RW	0x00000000	32 位寄存器, GPIO 输入输出控制寄存器: 0: 输入; 1: 输出。

#### 18.4.2. 输出置位寄存器 GPIO\_SET(偏移: 08h)

比特	名称	属性	复位值	描述
31: 0	GPIO_SET	WO	0x00000000	32 位寄存器, GPIO 输出置位寄存器: 0: 无效操作; 1: 当 IO 为输出时, IO 置位。

#### 18.4.3. 输出清零寄存器 GPIO\_CLR(偏移: 0Ch)

比特	名称	属性	复位值	描述
----	----	----	-----	----

31: 0	GPIO_CLR	WO	0x00000000	32 位寄存器，GPIO 输出清零寄存器： 0：无效操作； 1：当 IO 为输出时，IO 清零。
-------	----------	----	------------	--

#### 18.4.4. GPIO 输出引脚映射寄存器 GPIO\_ODATA(偏移：10h)

比特	名称	属性	复位值	描述
31: 0	GPIO_ODATA	RW	0x00000000	32 位寄存器，GPIO 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。

#### 18.4.5. GPIO 输入引脚映射寄存器 GPIO\_IDATA(偏移：14h)

比特	名称	属性	复位值	描述
31: 0	GPIO_IDATA	RO	0x00000000	32 位寄存器，GPIO 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。

#### 18.4.6. GPIO 中断使能寄存器 GPIO\_IEN(偏移：18h)

比特	名称	属性	复位值	描述
31: 0	GPIO_IEN	RW	0x00000000	32 位寄存器，GPIO 中断使能寄存器： 0= 禁止相应引脚中断； 1= 使能相应引脚中断。

#### 18.4.7. GPIO 中断触发模式寄存器 GPIO\_IS(偏移：1Ch)

比特	名称	属性	复位值	描述
31: 0	GPIO_IS	RW	0x00000000	32 位寄存器，GPIO 中断模式： 0= 边沿检测； 1= 电平检测。

## 18.4.8. GPIO 中断触发模式寄存器 GPIO\_IBE(偏移: 20h)

比特	名称	属性	复位值	描述
31: 0	GPIO_IBE	RW	0x00000000	32 位寄存器, GPIO 中断模式: 0= 单边沿触发; 1= 双边沿触发。

## 18.4.9. GPIO 中断触发模式寄存器 GPIO\_IEV(偏移: 24h)

比特	名称	属性	复位值	描述
31: 0	GPIO_IEV	RW	0x00000000	32 位寄存器, GPIO 中断模式: 0= 下降沿/低电平触发; 1= 上升沿/高电平触发。

## 18.4.10. GPIO 中断状态清除寄存器 GPIO\_IC(偏移: 28h)

比特	名称	属性	复位值	描述
31: 0	GPIO_IC	WO	0x00000000	32 位寄存器, GPIO 中断清除寄存器: 0= 无效操作。 1= 清除对应引脚中断。

## 18.4.11. GPIO 原始中断状态寄存器 GPIO\_RIS(偏移: 2Ch)

比特	名称	属性	复位值	描述
31: 0	GPIO_RIS	RO	0x00000000	32 位寄存器, GPIO 原始中断寄存器: 0= 对应引脚无中断挂起。 1= 对应引脚有中断挂起。

### 18.4.12. GPIO 屏蔽后中断状态寄存器 GPIO\_MIS(偏移: 30h)

比特	名称	属性	复位值	描述
31: 0	GPIO_MIS	RO	0x00000000	32 位寄存器, GPIO 屏蔽后中断状态寄存器: 反映对应引脚屏蔽后的中断状态。

## 18.5. 使用流程

### 18.5.1. 输入输出 IO

- 配置 GPIO\_DIR 寄存器, 选择 GPIO 方向
- 可使用 GPIO\_SET/GPIO\_CLR 或 GPIO\_ODATA 来设置输出电平
- 使用 GPIO\_IDATA 来获取输入引脚电平

### 18.5.2. 中断触发模式

中断初始化过程:

1. 设置 GPIO\_DIR 为输入。
2. 清除 GPIO\_IE 以避免异常。
3. 配置寄存器 GPIO\_IS, 选择是边沿/电平触发类型。
4. 在单边沿触发方式下, 配置寄存器 GPIO\_IBE, 确定是单边触发还是双边触发。
5. 在单边沿触发方式下, 配置寄存器 GPIO\_IEV, 确定是哪种边沿触发类型。
6. 在电平触发方式下, 配置寄存器 GPIO\_IEV, 确定是哪种电平触发类型。
7. 配置寄存器 GPIO\_IC 来清除中断。
8. 配置寄存器 GPIO\_IE 使能相应位中断。

### 18.5.3. 清除中断

ISR 写 GPIO\_IC 来清除中断状态。如果在清除寄存器的同时有新的边沿触发中断产生, 这个新的中断将会保持有效直到下一次清除。读取中断状态操作应该在清 GPIO\_IE 之前进行, 清 GPIO\_IE 操作将清除相应中断状态。

## 19. 串行外设接口（SPI）

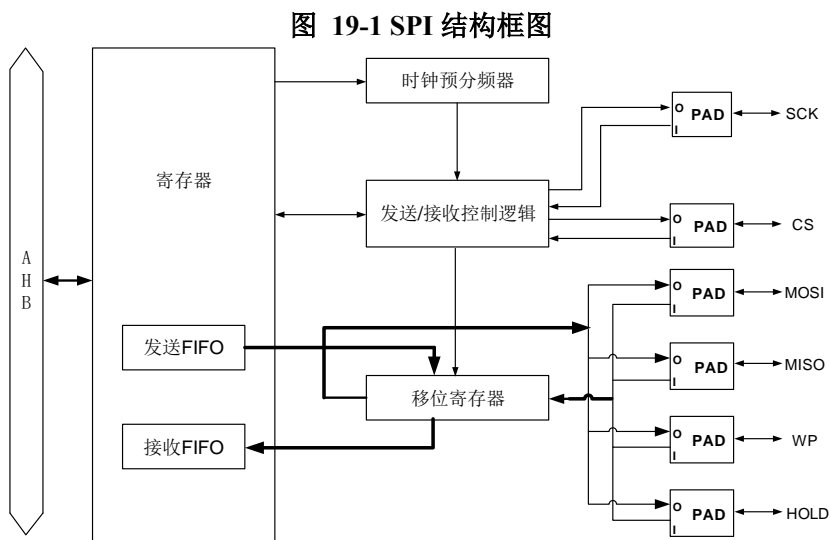
### 19.1. 概述

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。

### 19.2. 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；
- 其中 SPI3 支持存储器读取模式；

### 19.3. 结构框图



## 19.4. 功能描述

### 19.4.1. 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 19-2 SPI 一线模式时序图 (MSB)

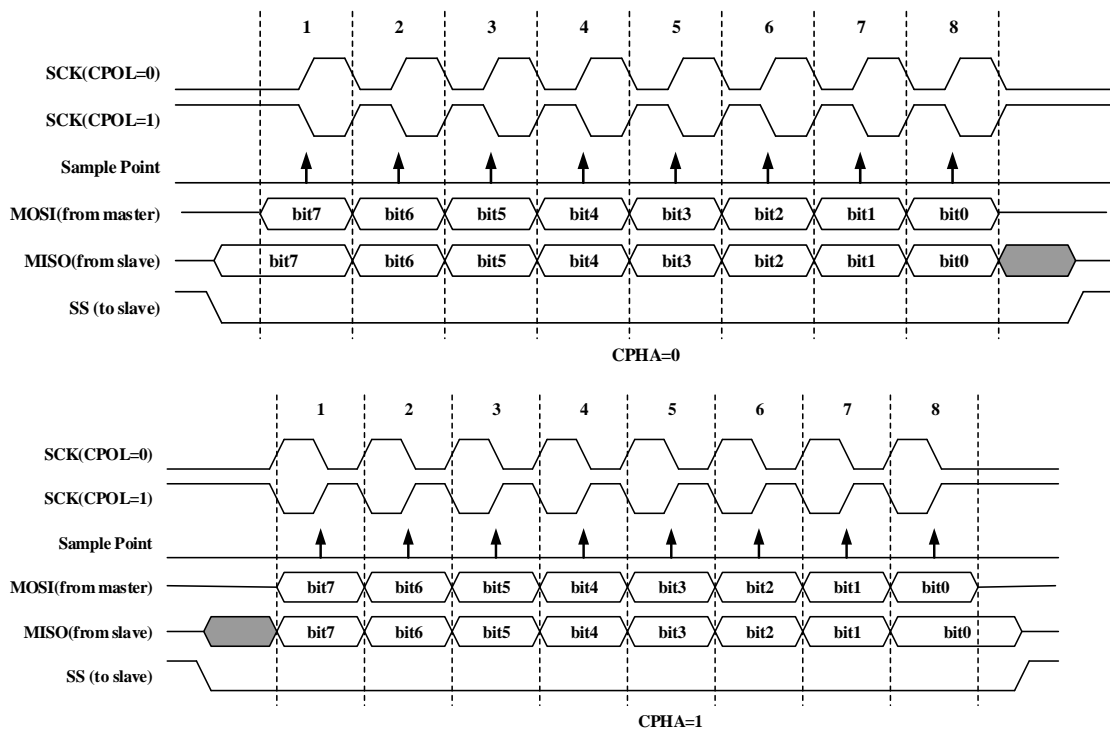
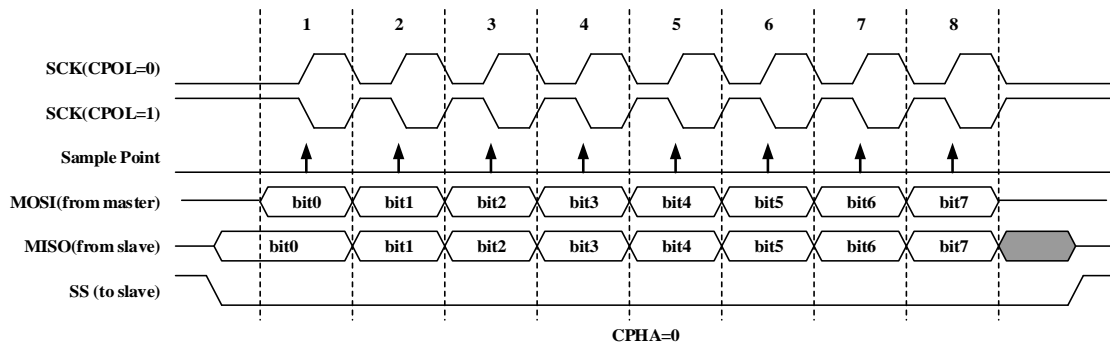
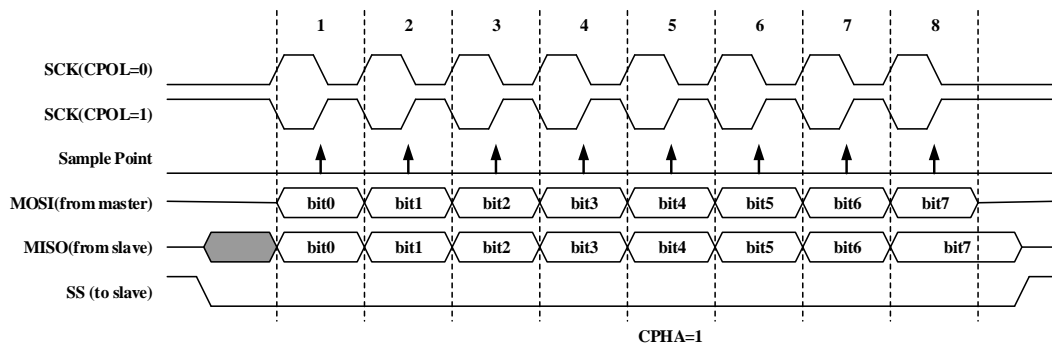


图 19-3 SPI 一线模式时序图 (LSB)





● 二线模式时序

图 19-4 SPI 二线模式时序图 (MSB)

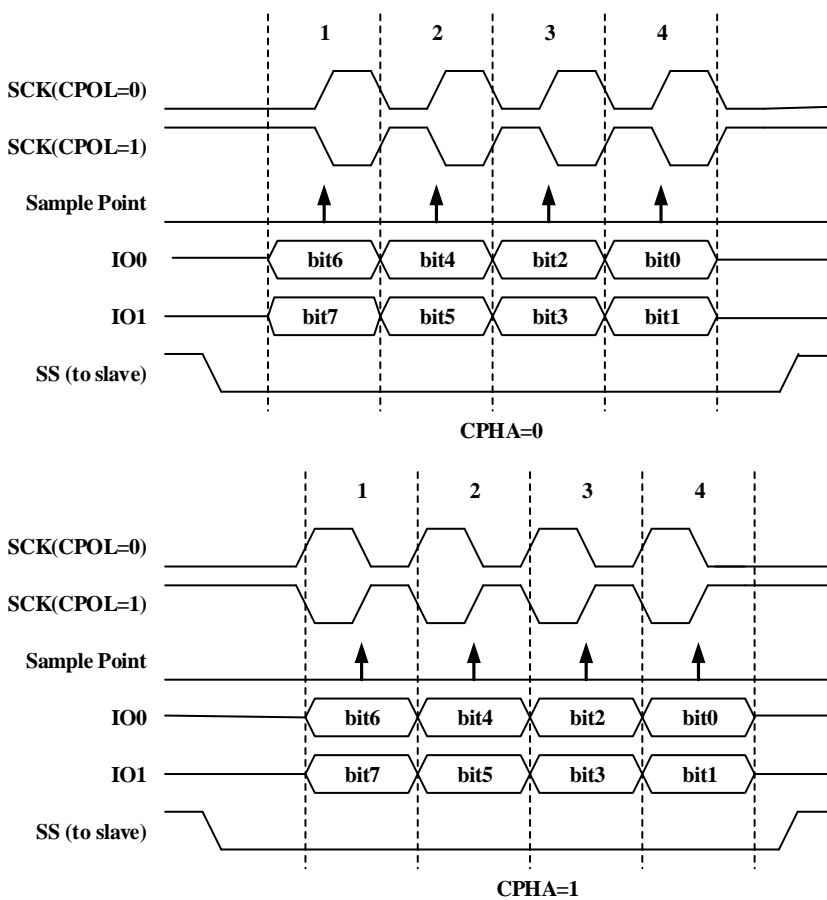
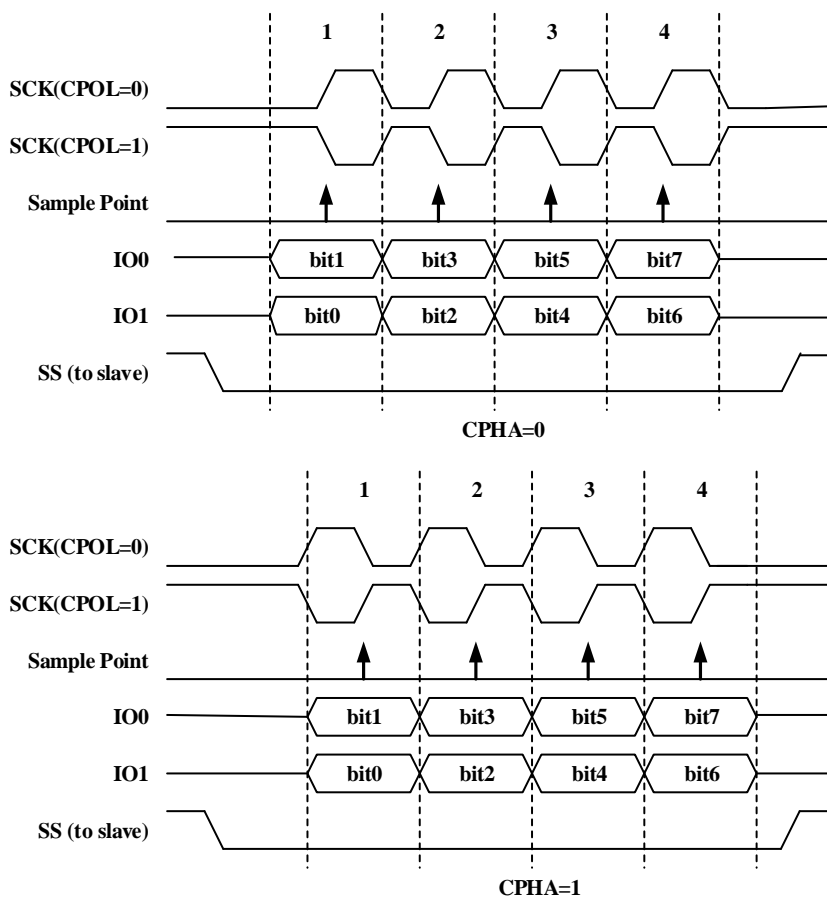
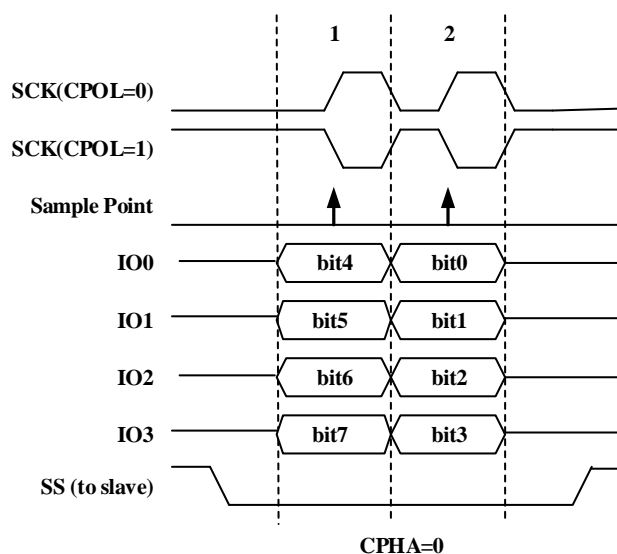


图 19-5 SPI 二线模式时序图 (LSB)



● 四线模式时序

图 19-6 SPI 四线模式时序图 (MSB)



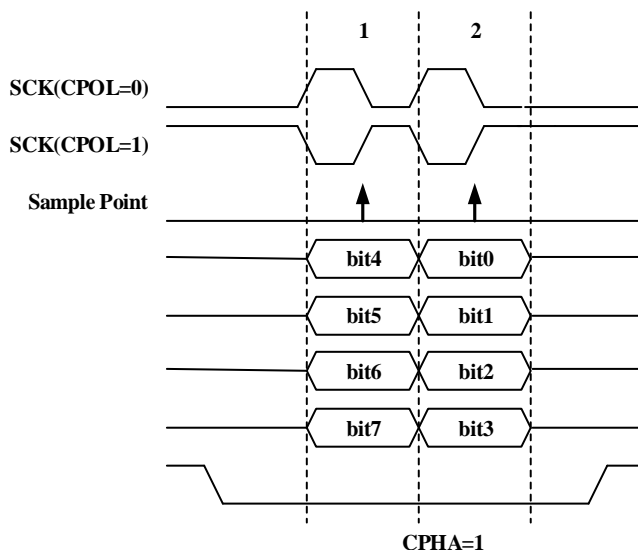
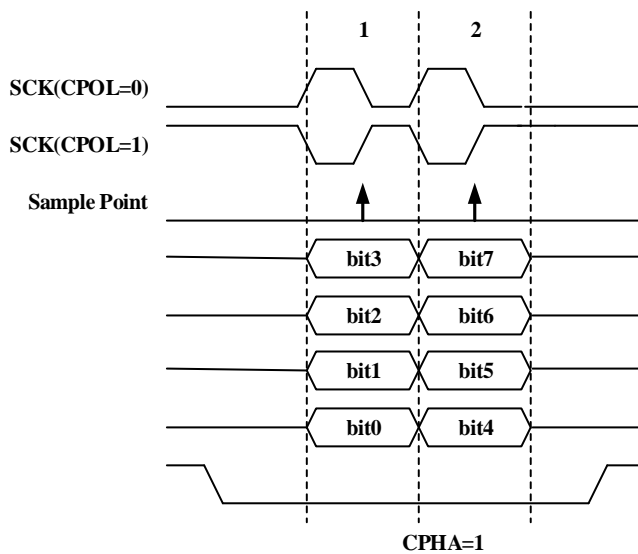
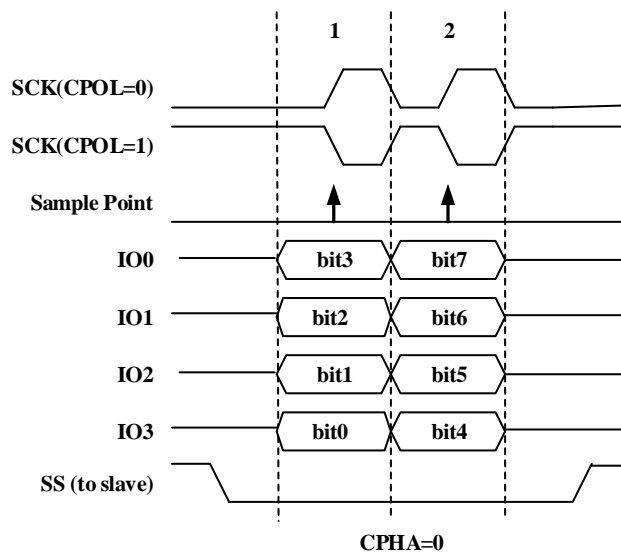


图 19-7 SPI 四线模式时序图 (LSB)



### 19.4.2. 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 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。

### 19.4.3. 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 寄存器中的 TX\_DMA\_REQ\_EN 位使能 SPI 的 DMA 发送，设置 SPI\_RX\_CTL 寄存器中的 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 寄存器中的 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 寄存器中的 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。

### 19.4.4. 存储器读取模式

其中一个 SPI 控制器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 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。

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。

## 19.5. 寄存器描述

SPI1 寄存器基地址: 0x4002\_0000

SPI2 寄存器基地址: 0x4002\_0400

SPI3 寄存器基地址: 0x4002\_0800

SPI4 寄存器基地址: 0x4002\_0C00

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_TXDelay	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	SPI3 取值控制寄存器
0x30	SPI_CMD	SPI3 取值命令寄存器
0x34	SPI_PARA	SPI3 取值参数寄存器

### 19.5.1. 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。
--	--	--	--	-----------------------------

### 19.5.2. 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。

### 19.5.3. 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。

$$\text{SPI 串行时钟:SPI\_CLK} = F_{\text{HCLK}} / (\text{DIV1} * (\text{DIV2}+1))$$

### 19.5.4. SPI 控制寄存器 SPI\_CTL(偏移: 08h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19	RSV	-	-	保留，此位须为 0
18:11	CS_TIME	RW	0x5	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	0	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	0	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	0	<p>从机收发逻辑使能位</p> <p>1: 使能 SPI 从机功能;</p> <p>0: 不使能 SPI 从机功能 ;</p> <p>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。</p>
7	IO_MODE	RW	0	<p>IO 方向模式选择位</p> <p>1: 硬件自动切换;</p> <p>0: 软件切换;</p>
6:5	X_Mode	RW	00	<p>多线模式控制位</p> <p>00: 1X 模式;</p> <p>01: 2X 模式;</p> <p>10: 4X 模式;</p> <p>11: 保留;</p>
4	LSB_first	RW	0	<p>MSB/LSB 在前选择位</p> <p>1: SPI 总线传输中 LSB 在前;</p> <p>0: SPI 总线传输中 MSB 在前;</p>
3	CPOL	RW	0	<p>时钟极性控制位</p> <p>1: SCLK 低电平有效。空闲状态下为高;</p> <p>0: SCLK 高电平有效。空闲状态下为低;</p>
2	CPHA	RW	0	<p>时钟相位控制位</p> <p>1: 在时钟 SCLK 的偶边沿采样数据。</p> <p>0: 在时钟 SCLK 的奇边沿采样数据。</p>
1	SFILTER	RW	0	<p>从机时钟滤毛刺选择位</p> <p>1: 开启从机时钟滤毛刺功能;</p> <p>0: 不开启从机时钟滤毛刺功能;</p> <p>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</p>
0	Mst_mode	RW	0	<p>主从模式选择位</p> <p>1: SPI 工作在主模式下;</p> <p>0: SPI 工作在从模式下;</p>

## 19.5.5. 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
31:16	RSV	-	-	保留
15:8	Dummy	RW	0x0	无效字节寄存器
7:4	TX_DMA_Level	RW	0000	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	0	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	0	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	0	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	0	发送使能位 1: TX 方向使能; 0: TX 方向禁止;

## 19.5.6. 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
31:8	RSV	-	-	保留
7:4	RX_DMA_Level	RW	0000	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	0	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	0	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	0	接收使能位 1: RX 方向使能; 0: RX 方向禁止;

### 19.5.7. 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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	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	0	发送 FIFO 空中断使能位 1: 中断使能; 0: 中断禁止;
1	BATCH_DONE_EN	RW	0	批量完成中断使能位 1: 中断使能; 0: 中断禁止;
0	RSV	-	-	保留

### 19.5.8. 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	0	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	0	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	0	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	0	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	0	接收 FIFO 非空标志位
10	CS_POS_Flg	RC_W1	0	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	0	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	1	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	0	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	1	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	0	接收 FIFO 满标志位 1: 接收 FIFO 中满; 0: 接收 FIFO 未满; 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	1	接收 FIFO 空标志位 1: 接收 FIFO 空; 0: 接收 FIFO 非空;
3	TX_FIFO_FULL	RO	0	发送 FIFO 满标志位 1: 发送 FIFO 中满; 0: 发送 FIFO 未滿;
2	TX_FIFO_EMPTY	RO	1	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	RC_W1	0	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。

### 19.5.9. 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	<code>SPI_TDY</code>	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。

### 19.5.10. 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	<code>RSV</code>	-	-	保留
19:0	<code>Batch_Number</code>	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。

### 19.5.11. 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	<code>RSV</code>	-	-	保留
1:0	<code>SPI_CS</code>	WO	00	置 1 将使设备选择信号 <code>SPI_CS[1:0]</code> 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。

### 19.5.12. 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	0	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	0	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	0	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	0	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

### 19.5.13. SPI3 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	0000	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。 1000: 8 bit 0000: 不使能 其他非 0: 8bit。
4	RVS	-	-	保留
3	Con_Rd_EN	RW	0	连读使能位。 1: 使能连读。 0: 不使能连读。
2	Para_Ord2	RW	0	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	0	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	0	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。

#### 19.5.14. SPI3 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。

### 19.5.15. SPI3 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。

## 19.6. 使用流程

### 19.6.1. SPI 主模式发送

#### 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 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、发送阶段

- 1) 设置 SPI 发送控制寄存器 SPI\_TX\_CTRL 中的 TX\_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\_CTRL 中的 TX\_EN 位。
- 8) 清除 SPI 从设备选择寄存器，结束发送。

### 19.6.2. 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、发送阶段

- 1) 设置 SPI 发送控制寄存器 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。

### 19.6.3. 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。

### 19.6.4. 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。

### 19.6.5. 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。

### 19.6.6. 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。

### 19.6.7. 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。

### 19.6.8. 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；再后设置 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。

最后 AHB 可以直接对 sram 地址进行写操作，硬件会自动开启对 SPI sram 的通信，在通信完成后 SPI 接口会将值写入 sram。

## 20. 通用异步收发器（UART）

### 20.1. 概述

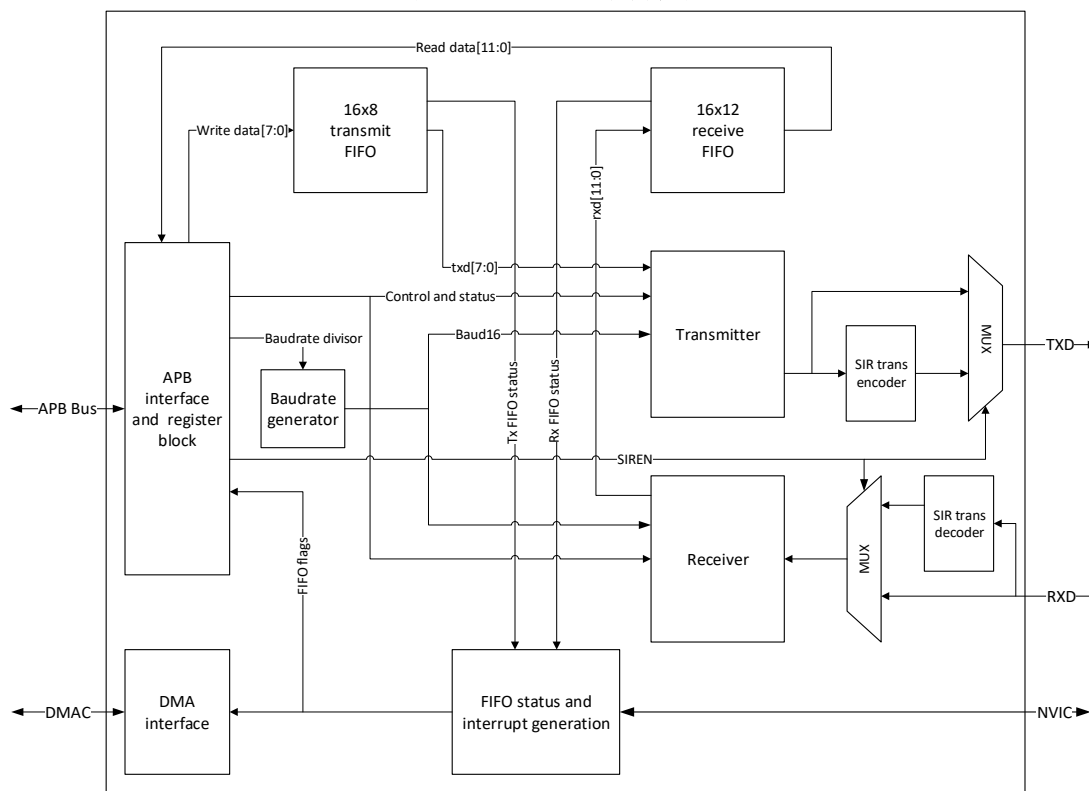
芯片上集成了四个 UART 串口模块。模块的收发端口可与 GPIO 管脚复用。

### 20.2. 主要特性

- 16 字节的硬件 FIFO
- 波特率支持整数和小数分频
- 支持 CTS, RTS 流控制
- Break 错误、奇偶或 0/1 校验位错误、帧格式错误检测
- 可编程位宽，奇偶或 0/1 校验，停止位个数
- 支持 LIN 和 IrDA 等功能
- 支持 DMA

## 20.3. 结构框图

图 20-1 UART 结构框图



## 20.4. 功能描述

### 20.4.1. 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。分频因子寄存器包含两个，其中 `UART_IBRD` 中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；寄存器 `UART_FBRD` 中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_LCRH` 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 5~8bit）；
- 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）；

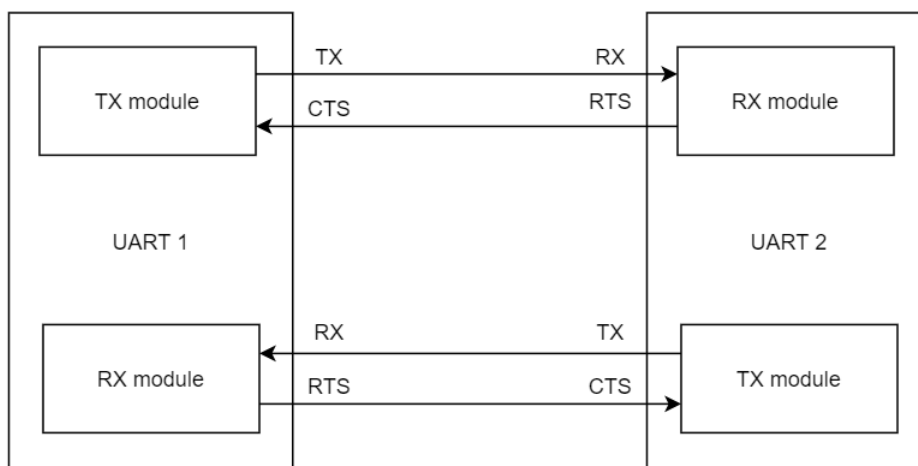
6、PEN 位配置校验使能；

完成基本的配置后，再选择是否使能中断。就可以进行串口数据的发送和接收了。

## 20.4.2. 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\_CR 的 CTSEn 和 RTSEn 位来使能硬件流控功能。

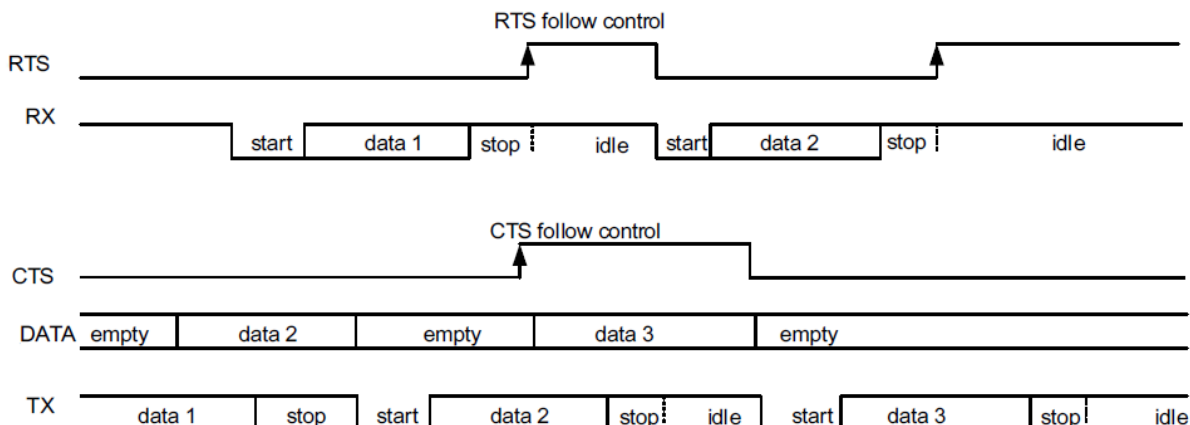
图 20-2 两个 UART 之间的硬件流控连接示意图



**RTS 流控：**由接收器输出，反映接收器缓冲区状态。，它用于反映接收缓冲区状态。当一帧数据接收完成，RTS 变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，RTS 保持高电平，可以通过读取 UART\_DR 中的数据来清除 RTS 状态。

**CTS 流控：**发送器监视 CTS 输入引脚来决定数据帧是否可以发送。如果 CTS 为低电平，发送器发送数据帧。在发送期间，若 CTS 信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

图 20-3 硬件流控时序图



### 20.4.3. DMA 请求

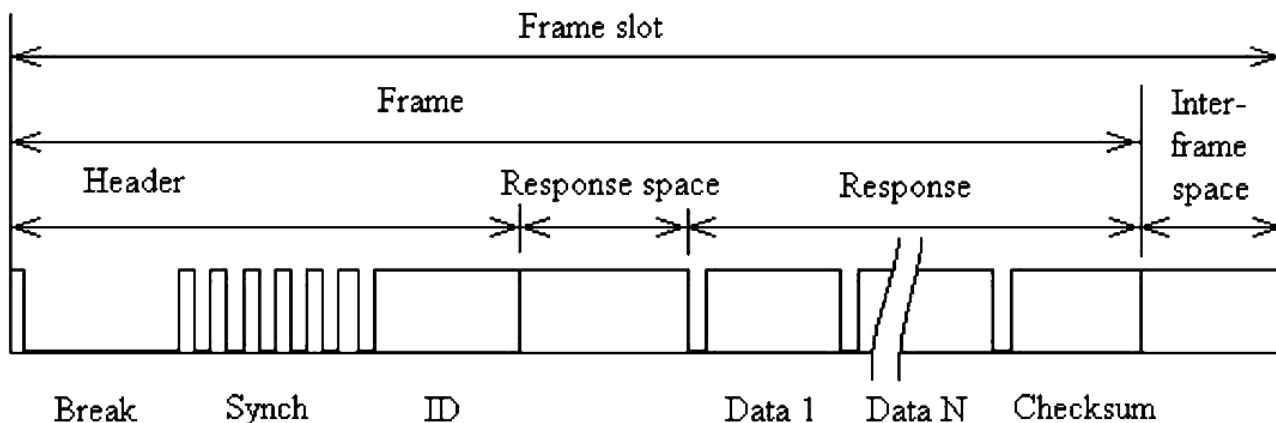
每个 UART 接口的 TX 和 RX 都支持 DMA 功能，都有其对应的 DMA 请求号。

设置 UART\_DMACR 寄存器中的 TXDMAE 位使能 DMA 发送，设置 RXDMAE 位使能 DMA 接收。

另外设置 DMAONERR 时可以忽略接收错误。

### 20.4.4. LIN 总线功能

图 20-4 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 20-5 LIN 间隔场格式

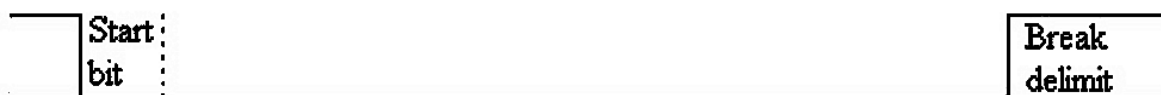
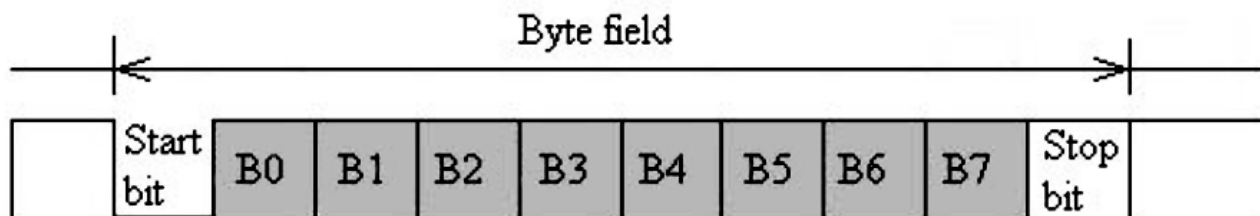


图 20-6 LIN 字符格式



设置 UART\_BCNT 寄存器的 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\_LCRH 寄存器的 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\_RIS 的 LBDI 位,或通过使能 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。

#### 20.4.5. IrDA SIR 功能

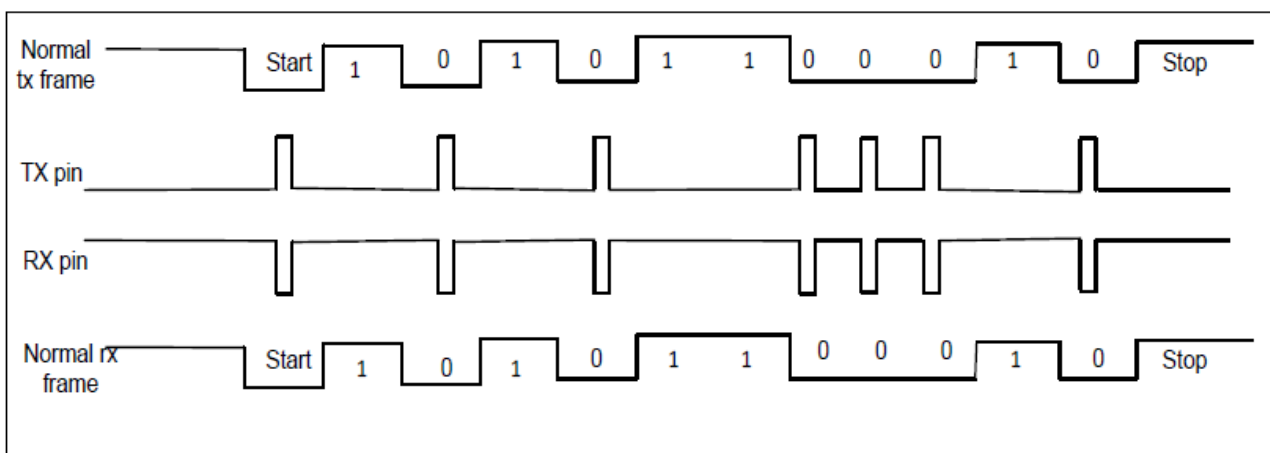
使能 UART\_CR 寄存器的 SIREN 位，即可打开 IrDA SIR 红外功能。

如若使用 IrDA SIR Low Power 模式，需要使能 UART\_CR 寄存器的 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。

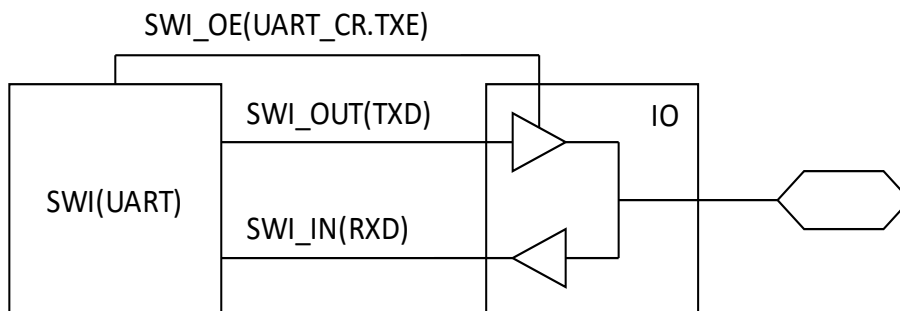
启用 Low Power 模式时，高电平脉冲宽度为为最高波特率 115200 时的一个位宽的 3/16。

图 20-7 IrDA SIR 数据调制解调



### 20.4.6. 单线模式

图 20-8 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 寄存器的 TXOE\_SEL 位和 RX\_SEL 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\_CR 的 TXE 为 1，开启 TX 引脚发送，此时需禁用 RXE。

接收时，设置 UART\_CR 的 TXE 为 0，开启 TX 引脚接收，此时需启用 RXE。

## 20.5. 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_RSR	接收状态寄存器
0x18	UART_FR	标志寄存器
0x20	UART_ILPR	IrDA 低功耗分频因子寄存器
0x24	UART_IBRD	整数分频因子寄存器
0x28	UART_FBRD	小数分频因子寄存器
0x2C	UART_LCRH	线控制器寄存器
0x30	UART_CR	控制寄存器
0x34	UART_IFLS	FIFO 中断触发值寄存器
0x38	UART_IMSC	中断屏蔽使能/清除寄存器
0x3C	UART_RIS	原始中断状态寄存器
0x40	UART_MIS	屏蔽后的中断状态寄存器
0x44	UART_ICR	中断清除寄存器
0x48	UART_DMACR	DMA 控制寄存器
0x54	UART_CR2	控制寄存器 2
0x58	UART_BCNT	比特计时寄存器

### 20.5.1. 数据寄存器 UART\_DR(偏移: 00h)

比特	名称	属性	复位值	描述
15:12	RSV	-	-	保留
11	OE	RO	0	Overrun 错误 0: 无错误 1: 有错误
10	BE	RO	0	Break 错误 0: 无错误 1: 有错误
9	PE	RO	0	奇偶校验错误 0: 无错误 1: 有错误
8	FE	RO	0	帧格式错误 0: 无错误 1: 有错误
7:0	DATA	RW	0x0	发送或接收的数据 注:

				<p>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。</p> <p>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。</p>
--	--	--	--	--

### 20.5.2. 接收状态寄存器 UART\_RSR(偏移：04h)

比特	名称	属性	复位值	描述
15:4	RSV	-	-	保留
3	OE	RC_W1	0	<p>Overrun 错误</p> <p>0: 无错误</p> <p>1: 有错误</p> <p>注：OE 只在接收时有效</p> <p>开启 FIFO 功能时，当接收 FIFO 已满后再接收到数据时被置位。</p> <p>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</p> <p>写 1 清 0;</p>
2	BE	RC_W1	0	<p>Break 错误</p> <p>0: 无错误</p> <p>1: 有错误</p> <p>当接收数据持续为低超过传输 1 个 word 的时间时被置位</p> <p>写 1 清 0;</p>
1	PE	RC_W1	0	<p>奇偶或 0/1 校验位错误</p> <p>0: 无错误</p> <p>1: 有错误</p> <p>写 1 清 0;</p>
0	FE	RC_W1	0	<p>帧格式错误</p> <p>0: 无错误</p> <p>1: 有错误</p> <p>当停止位错误时被置位</p>

				写 1 清 0;
--	--	--	--	----------

### 20.5.3. 标志位寄存器 UART\_FR(偏移: 18h)

比特	名称	属性	复位值	描述
15:8	RSV	-	-	保留
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	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
2:1	RSV	-	-	保留
0	CTS	RO	0	CTS 输入管脚状态 0: CTS 输入高电平 1: CTS 输入低电平

## 20.5.4. IrDA 低功耗分频因子寄存器 UART\_ILPR(偏移: 20h)

比特	名称	属性	复位值	描述
15:8	RSV	-	-	保留
7:0	ILPD	RW	0x0	IrDA 低功耗分频分数因子。 ILRD =integer(F <sub>PCLK</sub> /(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。

## 20.5.5. 整数分频因子寄存器 UART\_IBRD(偏移: 24h)

比特	名称	属性	复位值	描述
15:0	UART_IBAUD	RW	0x0	波特率分频整数因子 UART_IBAUD = (integer(F <sub>PCLK</sub> /(16*BAUD))), integer 为取整操作

## 20.5.6. 小数分频因子寄存器 UART\_FBRD (偏移: 28h)

比特	名称	属性	复位值	描述
15:6	RSV	-	-	保留
5:0	UART_FBAUD	RW	0x0	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 <sub>PCLK</sub> /(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;

## 20.5.7. 线控制器寄存器 UART\_LCRH (偏移: 2Ch)

比特	名称	属性	复位值	描述
15:8	RSV	-	-	保留

7	SPS	RW	0	校验模式选择位 0: 奇/偶校验 1: 0/1 校验
6:5	WLEN	RW	00	字宽选择位 00: 5bits 01: 6bits 10: 7bits 11: 8bits
4	FEN	RW	0	FIFO 使能位 0: 禁止 FIFO 1: 使能 FIFO
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	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
0	BRK	RW	0	BREAK 发送使能位 0: 禁止 1: 使能

### 20.5.8. 控制寄存器 UART\_CR (偏移: 30h)

比特	名称	属性	复位值	描述
15	CTSEn	RW	0	CTS 流控制使能位 0: 禁止 1: 使能
14	RTSEn	RW	0	RTS 流控制使能位 0: 禁止 1: 使能

13:12	RSV	-	-	保留
11	RTS	RW	0	RTS 输出管脚状态 0: RTS 输出高电平 1: RTS 输出低电平
10	RSV	-	-	保留
9	RXE	RW	1	接收使能位 0: 禁止 1: 使能
8	TXE	RW	1	发送使能位: 0: 禁止 1: 使能
7:3	RSV	-	-	保留
2	SIRLP	RW	0	IrDA SIR 低功耗模式使能位: 0: 禁止 1: 使能
1	SIREN	RW	0	IrDA SIR ENDEC 模块使能位: 0: 禁止 1: 使能
0	UARTEN	RW	0	UART 使能位 0: 禁止 1: 使能

### 20.5.9. FIFO 中断触发寄存器 UART\_IFLS(偏移: 34h)

比特	名称	属性	复位值	描述
15:6	RSV	-	-	保留
5:3	RXIFLSEL	RW	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: 1/16 (接收 FIFO 收到 1 个数据)。
2:0	TXIFLSEL	RW	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)。

				<p>001: 1/4 (发送到 FIFO 中剩余 4 个数据)。</p> <p>010: 1/2 (发送到 FIFO 中剩余 8 个数据)。</p> <p>011: 3/4 (发送到 FIFO 中剩余 12 个数据)。</p> <p>100: 7/8 (发送到 FIFO 中剩余 14 个数据)。</p> <p>101: 发送 FIFO 由非空到空时 (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)。注意, 此种配置下, 如果 TX 使能且 FIFO 里一开始为空时只写入一个字节的数据, 该数据会不经过 FIFO 直接写到移位寄存器里, FIFO 不会有从非空到空的过程, 因此 TX 中断也不会产生。</p>
--	--	--	--	---

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出发点的瞬间, 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出发点的瞬间, 往发送 FIFO 中写入数据达到触发点时也不会产生中断。

### 20.5.10. 中断使能寄存器 UART\_IMSC (偏移: 38h)

比特	名称	属性	复位值	描述
15:13	RSV	-	-	保留
12	BCNTI	RW	0	Bit Count Timeout 中断使能位 0: 禁止 1: 使能
11	LBDI	RW	0	LIN Break Detection 中断使能位。 0: 禁止 1: 使能
10	OEI	RW	0	overrun 中断使能位 0: 禁止 1: 使能
9	BEI	RW	0	break error 中断使能位 0: 禁止

				1: 使能
8	PEI	RW	0	奇偶校验错误中断使能位 0: 禁止 1: 使能
7	FEI	RW	0	帧格式错误中断使能位 0: 禁止 1: 使能
6	RTI	RW	0	接收数据读取超时中断使能位 0: 禁止 1: 使能
5	TXI	RW	0	发送中断使能位 0: 禁止 1: 使能
4	RXI	RW	0	接收中断使能位 0: 禁止 1: 使能
3:0	RSV	-	-	保留

### 20.5.11. 原始中断状态寄存器 UART\_RIS (偏移: 3Ch)

比特	名称	属性	复位值	描述
15:13	RSV	-	-	保留
12	BCNTI	RO	0	Bit Count Timeout 原始中断 0: 无中断 1: 有中断
11	LBDI	RO	0	LIN Break Detection 原始中断 0: 无中断 1: 有中断
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
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
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
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
6	RTI	RO	0	接收数据读取超时原始中断 0: 无中断 1: 有中断 当接收 FIFO 非空时, 32 个 bit 传输时间内没有任何新数据进入接收 FIFO, 也没有任何读操作时此位被置位
5	TXI	RO	0	发送原始中断 0: 无中断 1: 有中断 注: 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: 将 UART_ICR[5]寄存器写 1 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 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)。 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: 写 UART_ICR[5]寄存器 发送未完成时, 写新的数据到 UART_DR 寄存器。
4	RXI	RO	0	接收原始中断

				<p>0: 无中断 1: 有中断</p> <p>注:</p> <p>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:</p> <p>写 UART_ICR[4]寄存器 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 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)。</p> <p>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:</p> <p>写 UART_ICR[4]寄存器 读 UART_DR 寄存器</p>
3:0	RSV	-	-	保留

### 20.5.12. MASK 后的中断状态寄存器 UART\_MIS (偏移: 40h)

比特	名称	属性	复位值	描述
15:13	RSV	-	-	保留
12	BCNTI	RO	0	Bit Count Timeout MASK 后的中断 0: 无中断 1: 有中断
11	LBDI	RO	0	LIN Break Detection MASK 后的中断 0: 无中断 1: 有中断
10	OEI	RO	0	Overrun MASK 后的中断 0: 无中断 1: 有中断
9	BEI	RO	0	break errorMASK 后的中断 0: 无中断

				1: 有中断
8	PEI	RO	0	奇偶校验错误 MASK 后的中断 0: 无中断 1: 有中断
7	FEI	RO	0	帧格式错误 MASK 后的中断 0: 无中断 1: 有中断
6	RTI	RO	0	接收数据读取超时 MASK 后的中断 0: 无中断 1: 有中断
5	TXI	RO	0	发送 MASK 后的中断 0: 无中断 1: 有中断
4	RXI	RO	0	接收 MASK 后的中断 0: 无中断 1: 有中断
3:0	RSV	-	-	保留

### 20.5.13. 中断状态清除 UART\_ICR (偏移: 44h)

比特	名称	属性	复位值	描述
15:13	RSV	-	-	保留
12	BCNTIC	WO	0	Bit Count Timeout 中断状态清除位 写 1 清
11	LBDIC	WO	0	LIN Break Detection 中断状态清除位 写 1 清
10	OEI	WO	0	overrun 中断状态清除位 写 1 清
9	BEI	WO	0	break error 中断状态清除位 写 1 清
8	PEI	WO	0	奇偶校验错误中断状态清除位 写 1 清
7	FEI	WO	0	帧格式错误中断状态清除位 写 1 清
6	RTI	WO	0	接收数据读取超时中断状态清除位

				写 1 清
5	TXI	WO	0	发送中断状态清除位 写 1 清
4	RXI	WO	0	接收中断状态清除位 写 1 清
3:0	RSV	-	-	保留

#### 20.5.14. DMA 控制寄存器 UART\_DMACR (偏移: 48h)

比特	名称	属性	复位值	描述
15:3	RSV	-	-	保留
2	DMAONERR	RW	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
1	TXDMAE	RW	0	发送 DMA 使能 0: 不使能 1: 使能
0	RXDMAE	RW	0	接收 DMA 使能 0: 不使能 1: 使能

#### 20.5.15. 控制寄存器 2UART\_CR2 (偏移: 54h)

比特	名称	属性	复位值	描述
15:2	RSV	-	-	保留
1	TXOE_SEL	RW	0	TX 通路输出使能控制 0: TX 输出一直使能 1: TX 输出在发送时输出, 用于单线半双工模式
0	RX_SEL	RW	0	接收通路选择 0: 从 RX 通路输入 1: 从 TX 通路输入, 用于单线半双工模式, 或

				者通路测试
--	--	--	--	-------

### 20.5.16. 比特计时寄存器 UART\_BCNT (偏移: 58h)

比特	名称	属性	复位值	描述
15:9	RSV	-	-	保留
8	BCNT_START	WO	0	比特计时开始。
7:0	BCNT_VALUE	RW	0xD	比特计时初值。

## 20.6. 使用流程

### 20.6.1. 串口的发送和接收

1. 配置波特率
2. 配置 FIFO(是否使用 FIFO)
3. 配置线控制寄存器(奇偶校验位等)
4. 配置控制寄存器(是否使用中断)
5. 使能 UART

### 20.6.2. CTS 和 RTS 控制流功能

- 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，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。

### 20.6.3. 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 UARTRIS 的 LBDI 位；
  - 使能 UARTIE 的 LBDI 位，然后触发中断，在中断查看 UARTRIS 的 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\_LCRH 的 WLEN 决定，建议 8 位。）
  - 校验位（0 位或 1 位校验位，由 UART\_LCRH 的 PEN 决定）
  - 停止位（1 位或 2 位停止位，由 UART\_LCRH 的 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\_LCRH 的 BRK 位和 UART\_BCNT 的 BCNT\_START 位。
  - 轮询访问 UARTRIS 的 BCNTI 位，等待置 1；或者使能中断，等待 BCNTI 中断。
  - 清零 UART\_LCRH 的 BRK 位。

### 20.6.4. IrDA SIR 功能使用流程

- 使能 UART\_CR 寄存器的 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\_CR 寄存器的 SIRLP 位。

### 20.6.5. 单线模式功能使用流程

- 使能 UART\_CR2 寄存器的 TXOE\_SEL 位和 RX\_SEL 位。

## 21. 内部集成电路总线接口（I2C）

### 21.1. 概述

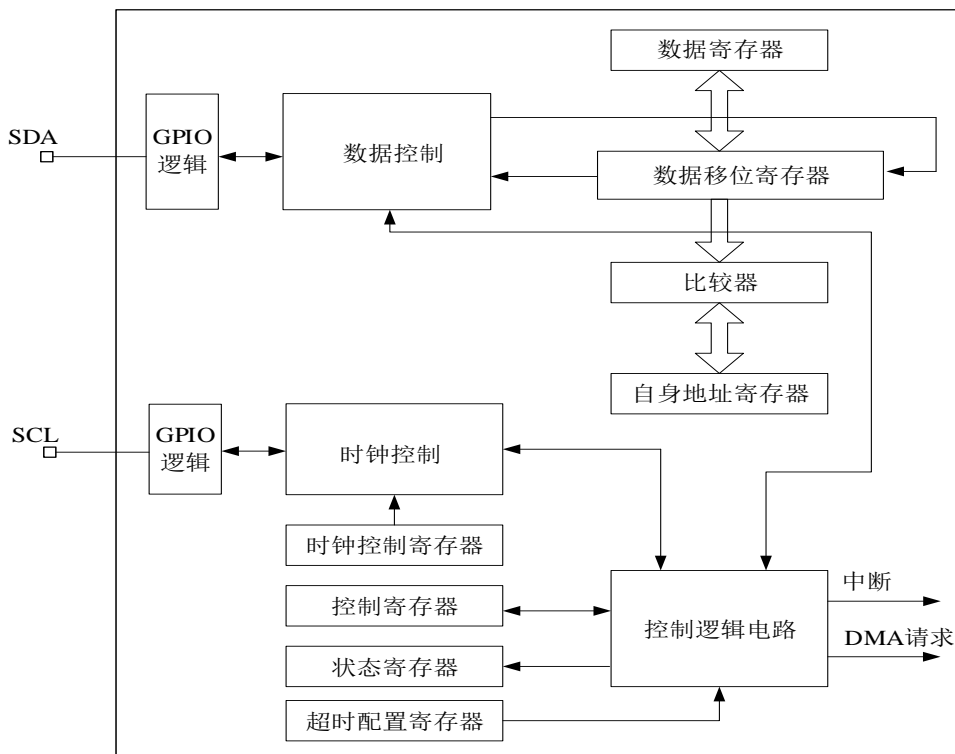
I2C 总线接口连接微控制器和串行 I2C 总线。I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。I2C 模块通过数据引脚 SDA 和时钟引脚 SCL 连接到 I2C 总线，控制所有 I2C 总线规定的时序。本模块支持主模式和从模式。

### 21.2. 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 标准模式/400K 快速模式/1M 快速加模式；
- 支持 7bit 设备地址；
- 支持主机仲裁模式；

## 21.3. 结构框图

图 21-1 I2C 结构框图

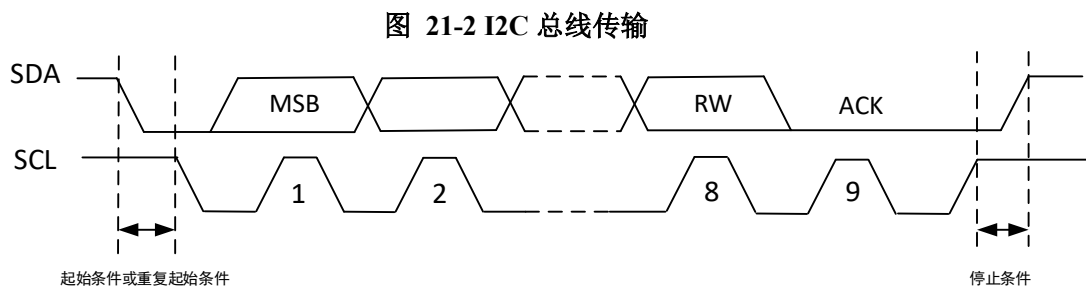


## 21.4. 功能描述

### 21.4.1. 模式选择

该模块默认为从模式，当软件写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：

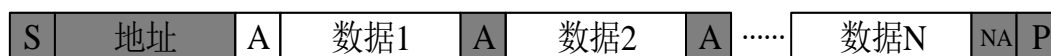


### 21.4.2. I2C 从模式

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 寄存器的 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 21-3 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，新收到的数据将不断覆盖移位寄存器中的数据但不会被加载到 DR 寄存器中，直到 DR 寄存器中的数据被读走然后才会将移位寄存器中的数据加载到 DR 中。

图 21-4 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

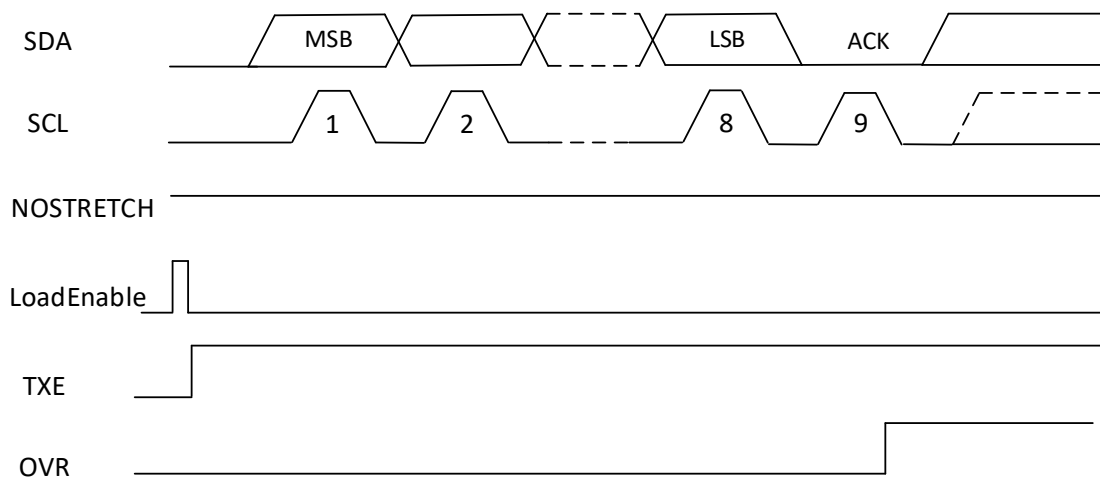
- 关闭从通信

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 21-5 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）



### 21.4.3. 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。

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 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 21-6 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

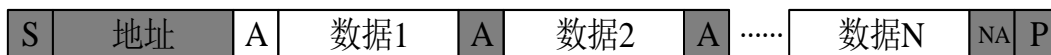
发送了地址后，I2C 接口从 SDA 线接收数据字节，并通过内部移位寄存器存储到数据寄存器，产生数据寄存器非空标志 RXNE，软件需要读出数据寄存器的值来清除 RXNE 标志。

若当前字节传输完成，不读取数据清除 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 RXNE 清 0，主机开始一次新的传输。

主设备在从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/重开始条件。

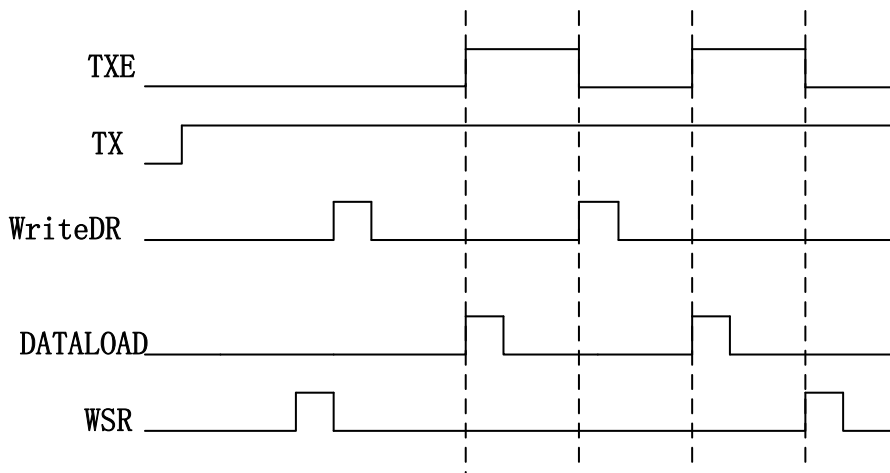
图 21-7 7 位主接收器的传送图



说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

#### 21.4.4. TXE 状态

图 21-8 TXE 功能说明图



TXE 反应了 I2C\_DR 数据是否被取走。

在接收状态下，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，写状态寄存器 TXE 位为 1 清 0；

PS: TX 1 为发送，0 为接收，DATALOAD 加载数据到发送引擎，WriteDR 写 I2C\_DR 寄存器，WSR 写状态寄存器。

### 21.5. 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
0x1C	I2C_FILTER	I2C 滤波寄存器

0x24	I2C_Timeout	I2C 超时配置寄存器
------	-------------	-------------

### 21.5.1. 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

### 21.5.2. 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 <sub>scl</sub> = (F <sub>PCLK</sub> ) / (4 * (I2C_CLK_DIV + 1)) 注： 1、F <sub>PCLK</sub> 为 APB 时钟频率。 2、I2C_CLK_DIV 的值必须大于 4。

### 21.5.3. 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	0	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	0	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	0	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	0	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	0	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	1	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	OD_MODE	RW	0	SCL 与 SDA 输出模式选择： 0:push-pull 模式输出 1:open-drain 模式输出 注： open-drain 模式下需要接上拉电阻
13	RSV	-	-	保留，此位须为 0
12	RX_ADDR2_INT_EN	RW	0	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	0	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	0	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	0	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	0	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	0	<p>传输应答位</p> <p>0: 接收一字节后, 在应答周期产生 ACK;</p> <p>1: 接收一字节后, 在应答周期产生 NACK;</p> <p>注:TACK 必须在应答周期前写入。</p>
5	STOP	RW	0	<p>结束条件产生位</p> <p>0: 发送完当前字节不产生结束条件;</p> <p>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</p>
4	START	RW	0	<p>起始条件产生位</p> <p>0: 主模式下不产生起始条件</p> <p>1: 主模式下产生起始条件</p> <p>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。</p>
3	TX	RW	0	<p>发送接收选择位</p> <p>0: 设备作为接收器</p> <p>1: 设备作为发送器</p> <p>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</p> <p>TX_AUTO_EN 位使能后 TX 位设置无效</p>
2	MASTER	RW	0	<p>主从设备选择位</p> <p>0: 从模式</p> <p>1: 主模式。</p>
1	NOSTRETCH	RW	0	<p>从模式禁止时钟延长</p> <p>0: 使能时钟延长</p> <p>1: 禁止时钟延长</p>
0	MEN	RW	0	<p>设备使能位</p> <p>0: 设备不使能</p> <p>1: 设备使能</p>

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。

## 21.5.4. 状态寄存器 I2C\_SR(偏移: 0x0Ch)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留位
16	TIMEOUTBF	RC_W1	0	TIMEOUTB 超时标志, 表示 SMBus EXT 超时 写 1 清 0
15	TIMEOUTAF	RC_W1	0	TIMEOUTA 超时标志, 表示 SMBus SCL Timeout 超 时 写 1 清 0
14	RX_ADDR3	RC_W1	0	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	RSV	-	-	保留
12	RX_ADDR2	RC_W1	0	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	0	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	RC_W1	0	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	RC_W1	0	<p>发送数据时数据寄存器状态位</p> <p>0: I2C_DR 数据未被发送器取走</p> <p>1: I2C_DR 数据被发送器取走</p> <p>硬件置位, 写 1 清 0, 写 DR 清 0</p> <p>注: 主机发送模式下, 写 DR 寄存器后主机才会发送数据</p>
8	RX_ADDR1	RC_W1	0	<p>从设备地址 1 匹配状态位</p> <p>0: 设备地址 1 和接收到的地址不相等</p> <p>1: 设备地址 1 和接收到的地址相等</p> <p>写 1 清 0</p> <p>注: 当地址匹配时, SRW 位表示了地址字节的 RW 位。</p>
7	MTF	RC_W1	0	<p>字节传输完成状态位</p> <p>0: 字节传输未完成</p> <p>1: 字节传输完成</p> <p>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</p>
6	MARLO	RO	0	<p>主模式仲裁丢失</p> <p>0: 没有检测到仲裁丢失</p> <p>1: 检测到仲裁丢失</p> <p>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</p>
5	TX_RX_FLAG	RC_W1	0	<p>主机发送转接收状态位</p> <p>0: 主机未由发送状态转成接收状态</p> <p>1: 主机由发送状态转为接收状态。</p> <p>硬件置 1, 写 1 清 0;</p> <p>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。</p>
4	BUS_BUSY	RO	0	<p>总线忙碌状态位</p> <p>0: 总线上无数据通信 (检测到总线上的结束条件, 此位清 0)</p> <p>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)</p>

3	SRW	RO	0	<p>从机读状态指示位</p> <p>0: 作为从设备接收器</p> <p>1: 作为从设备发送器</p> <p>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</p>
2	STOPF	RC_W1	0	<p>STOP 条件位检测位, 主从都会产生</p> <p>1: 检测到停止位</p> <p>0: 未检测到停止位</p> <p>只能写 1 清 0</p>
1	STARTF	RC_W1	0	<p>主机起始条件发送状态位</p> <p>1: 起始条件已发送</p> <p>0: 起始条件未发送</p> <p>写 1 清 0</p> <p>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。</p>
0	RACK	RO	1	<p>应答接收状态位</p> <p>0: 最近的发送应答周期接收到 ACK</p> <p>1: 最近的发送应答周期接收到 NACK</p> <p>只有 START 条件将清除 RACK 位</p> <p>注: 主机发送模式下收到 NACK 停止发送, 等待软件处理事件</p>

### 21.5.5. 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

### 21.5.6. 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

### 21.5.7. I2C 滤波寄存器 I2C\_FILTER (偏移: 0x1Ch)

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	SCL 滤波值设置位。 滤的毛刺的最大宽度为 $T_{pclk} * 8 * SCL\_FILTER$ 。T <sub>pclk</sub> 为 PCLK 周期

### 21.5.8. I2C 超时配置寄存器 I2C\_TIMEOUT (偏移: 0x24h)

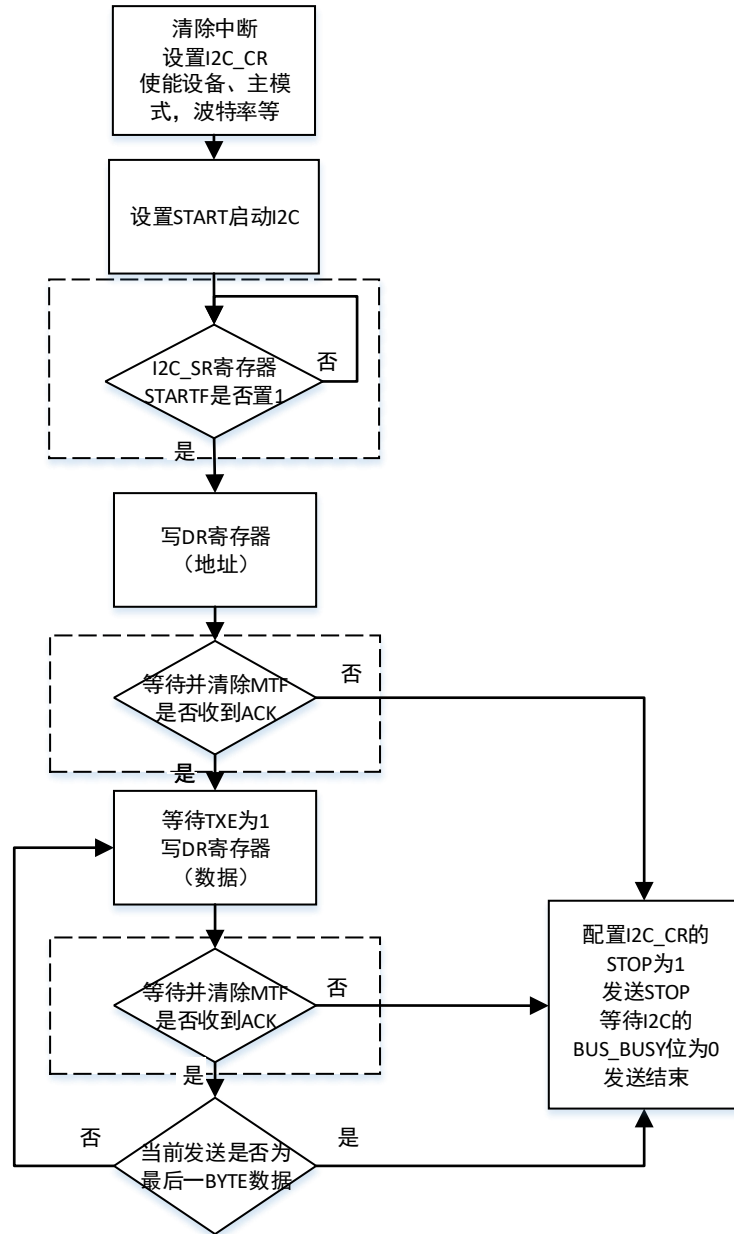
比特	名称	属性	复位值	描述
31	EXTEN			TimeoutB 定时使能，超出后产生 TIMEOUTB 标志
30	TOUTB_INTEN			TIMEOUTBF 标志中断使能
29	EXT_MODE			EXT 计数 (TimeoutB) 模式选择 0: 计数 SEXT 模式 1: 计数 MEXT 模式
28	RSV	-	-	保留
27:16	TIMEOUTB	RW	0x0	SMBus EXT 时间设置，以 PCLK 的 2048 倍

				为计时长度
15	TIMOUTEN			TimeoutA 定时使能，超出后产生 TIMEOUTA 标志
14	TOUTA_INTEN			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

## 21.6. 软件操作流程

### 21.6.1. 作为主发送器

图 21-9 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 的 STARTF 标志为 1(也可不等待)，把 I2C 要访问的 SLAVE 的 7 位地址

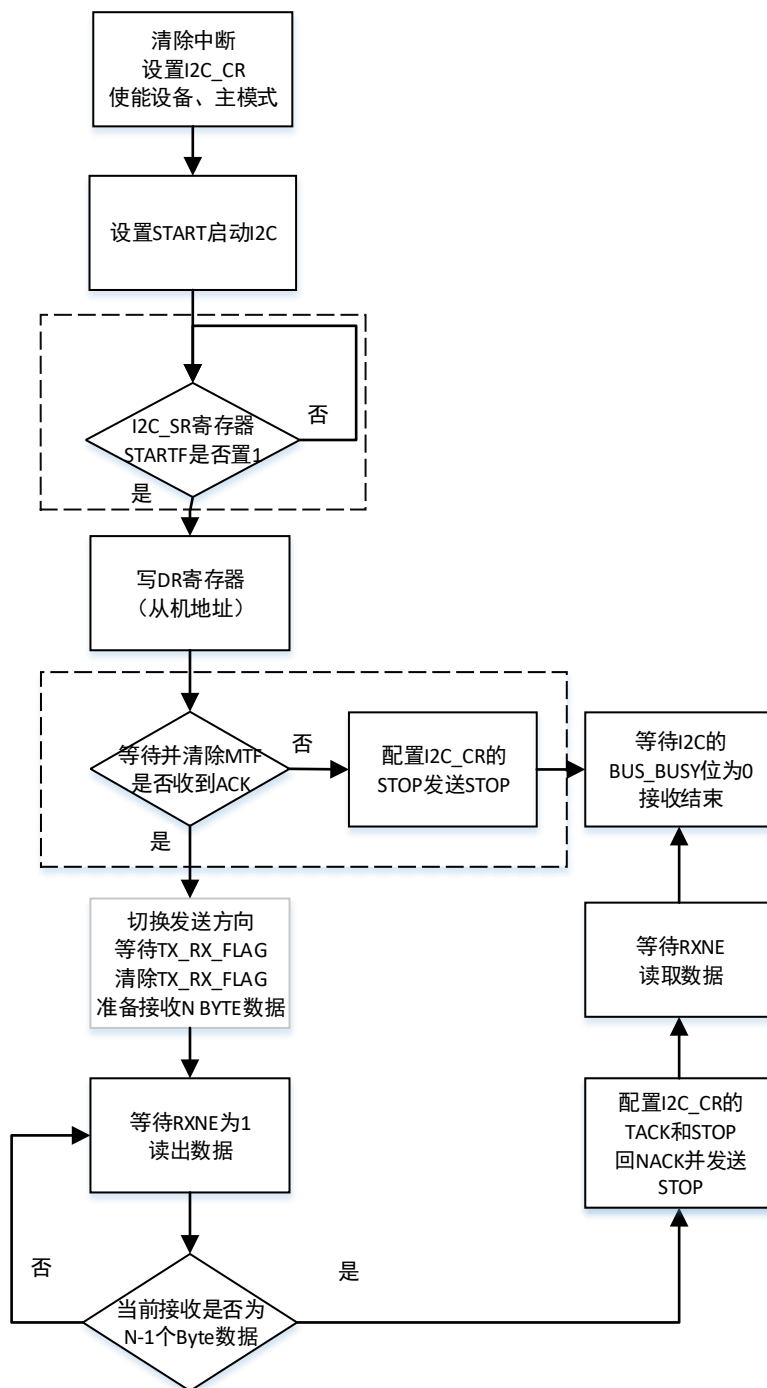
写入 I2C\_DR 寄存器中。(如若直接写数据或 start 之前数据以写入, STARTF 将被清 0)

5. 等待 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 NAK, 软件写 I2C\_CR 的 STOP 位为 1, 1, 结束发送并释放总线, 软件等待 I2C\_SR 的 BUS\_BUSY 为 0 后退出。
6. 等待 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 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 的 STOP 位为 1, 结束发送并释放总线, 软件等待 I2C\_SR 的 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 的 STOP 为 1 结束发送。STOP 也可以在最后字节发送完成后写入。

MTX\_AUTO\_EN 使能后,, I2C\_CR 位的 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。

## 21.6.2. 作为主接收器

图 21-10 主机接收流程图

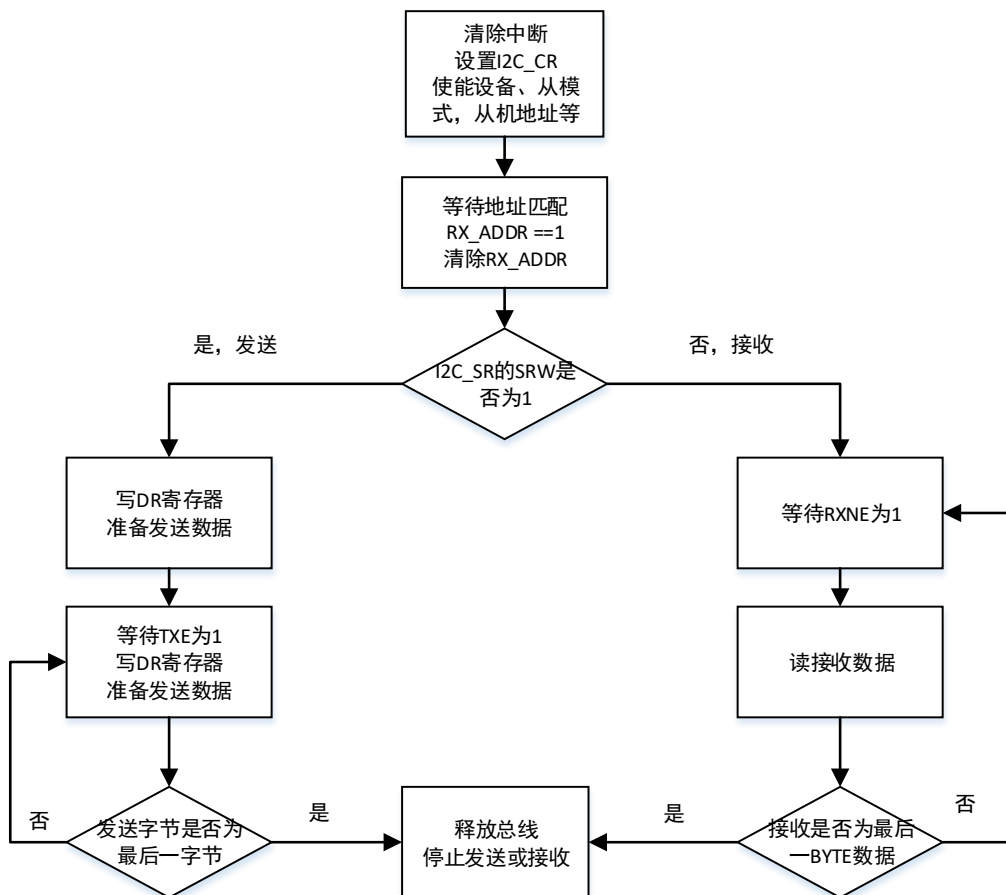


1. 清除中断标志
2. 写 I2C\_CLK\_DIV 寄存器的值确定 I2C 传输频率
3. 写 I2C\_CR 寄存器的 MEN、MASTER 和 START 为 1，发起 START 条件。
4. 等待 I2C\_SR 的 SRARTF 标志为 1(也可不等待)，把 I2C 要访问的 SLAVE 的 7 位地址

- 和 1bit 的 1 写入 I2C\_DR 寄存器中，表示作为接收器。
5. 等待 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 NAK，软件写 I2C\_CR 的 STOP 位，结束发送并释放总线，软件等待 I2C\_SR 的 BUS\_BUSY 位为 0 后退出。或者收到 NACK 后，软件写 I2C\_CR 的 START 位为 1，从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 的 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 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 的 BUS\_BUSY 为 0 后退出。

## 21.6.3. 作为从发送器

图 21-11 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 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 或 RX2\_ADDR2 (ADDR2\_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 寄存器的 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 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：MTX\_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。

#### 21.6.4. 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 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 或 RX\_ADDR2 (ADDR2\_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. 等待 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 的 BUS\_BUSY 为 0 后退出。

#### 21.6.5. 滤波算法

当 I2C\_FILTER 寄存器中的 SCL\_FILTER 不为 0 时，I2C 滤波功能开启。例如，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。其中  $T_{cntc}$  为 PCLK 的 8 分频时钟周期。

注：

1. 在使用过程中，设置 SDA\_IN\_DELAY 与 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  $T_{cntc} * I2C\_FILTER$  时间的高电平，低电平不滤波；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. 音频接口（I2S）

### 22.1. 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。

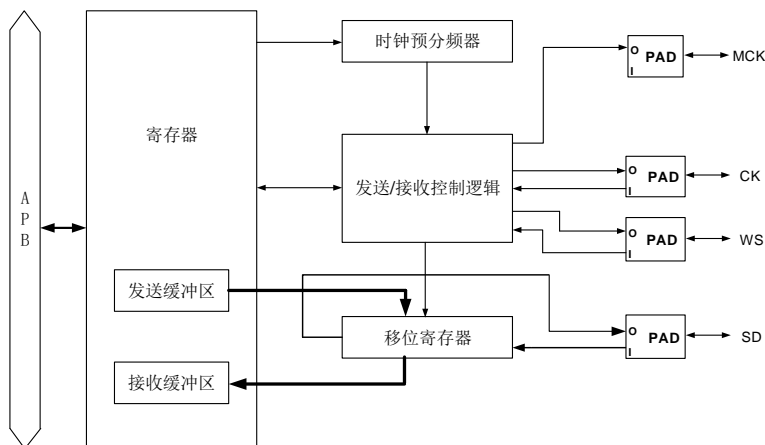
### 22.2. 主要特性

- 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。

## 22.3. 功能描述

### 22.3.1. I2S 结构框图

图 22-1 I2S 结构框图



### 22.3.2. I2S 音频标准

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，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。

有四种数据包类型可供选择：16 位数据打包成 16 位数据帧格式，16 位数据打包成 32 位数据帧格式，24 位数据打包成 32 位数据帧格式，32 位数据打包成 32 位数据帧格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

## I2S 飞利浦标准

对于 I2S 飞利浦标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的下降沿变化

图 22-2 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)

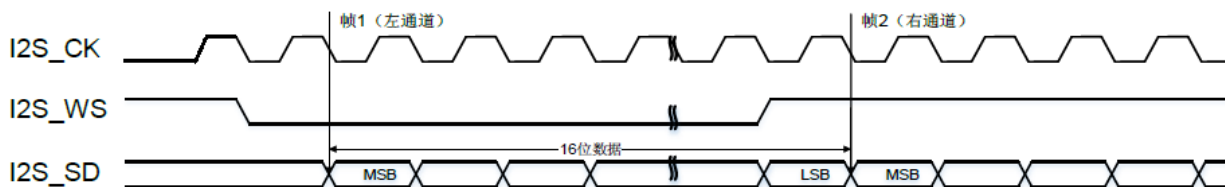


图 22-3 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)

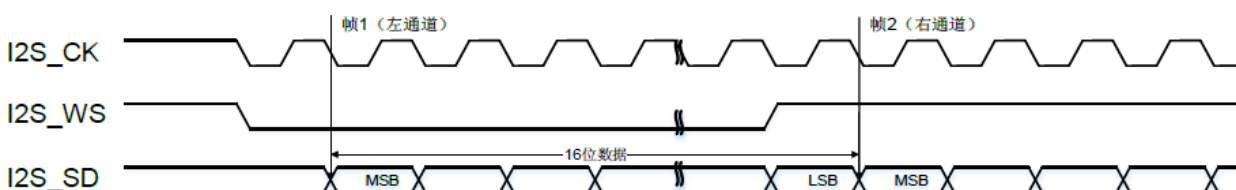


图 22-4 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)

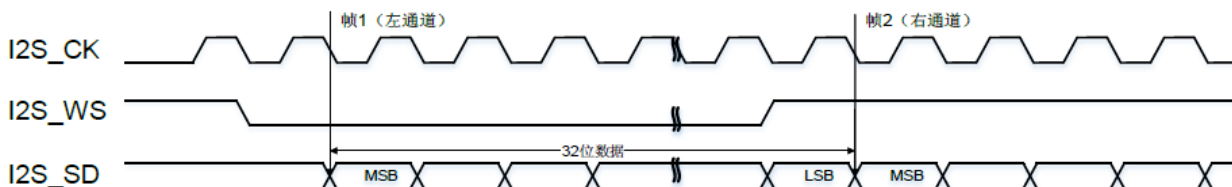


图 22-5 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)

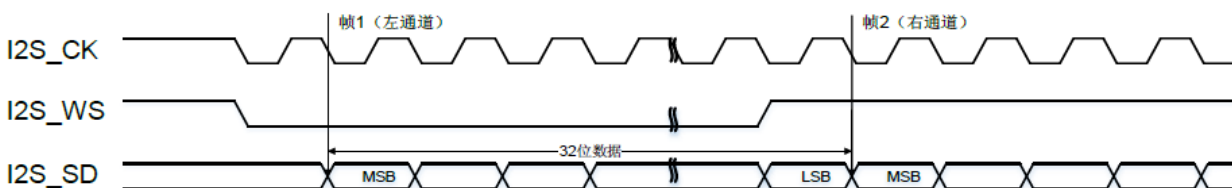


图 22-6 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

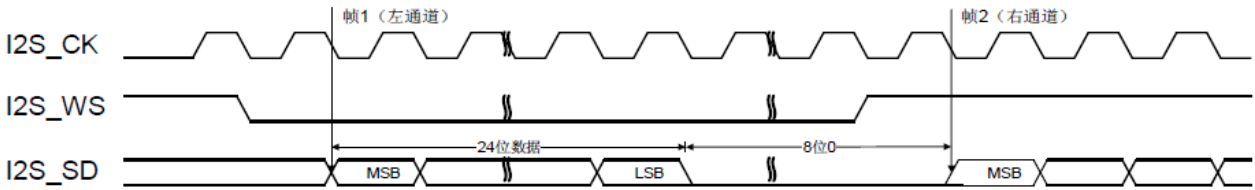


图 22-7 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)

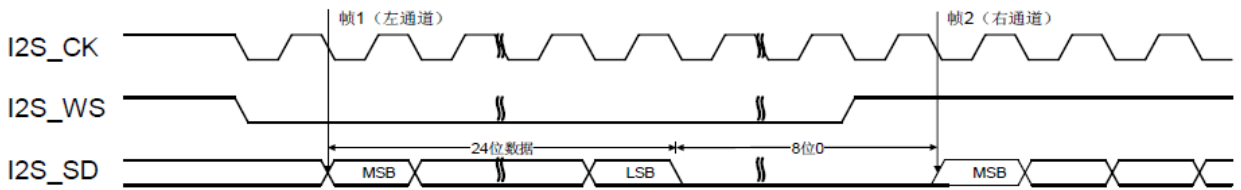


图 22-8 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

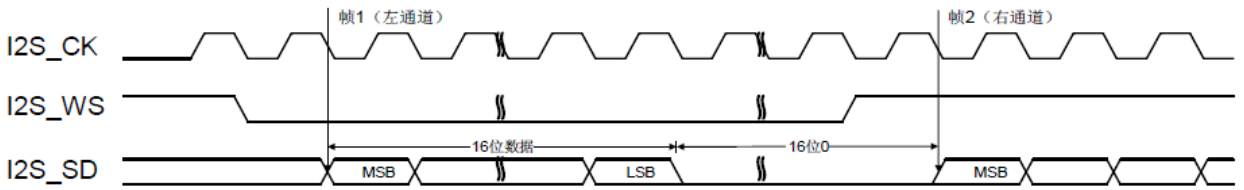
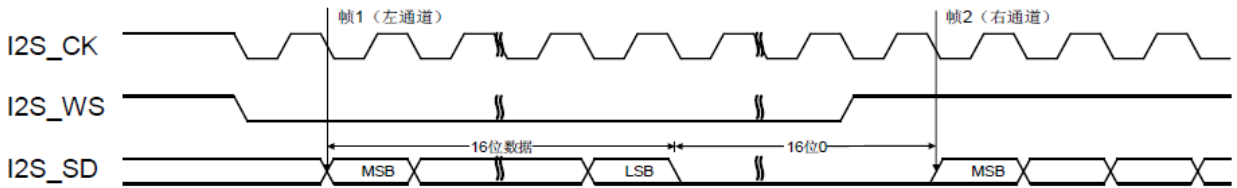


图 22-9 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



为了将该 16 位数据扩展成 32 位数据，剩下的 16 位被硬件强制填充为 0x0000。

### MSB 对齐标准

对于 MSB 对齐标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的下降沿变化

图 22-10 MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)

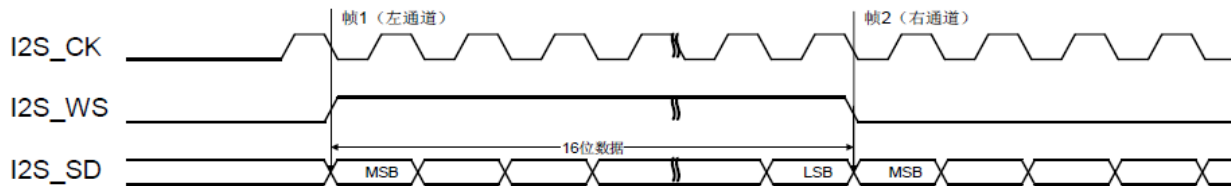


图 22-11 MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)

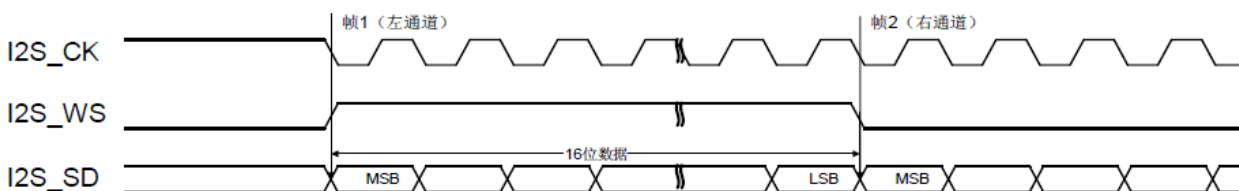


图 22-12 MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)

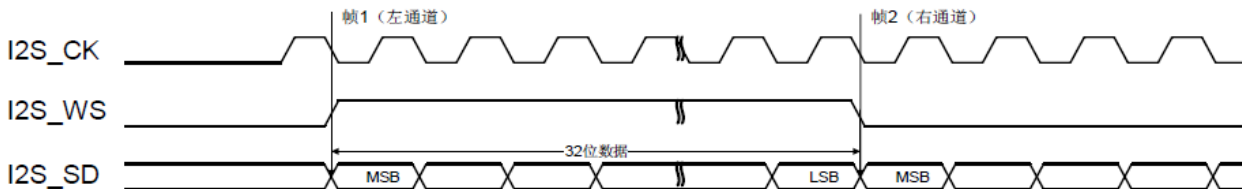


图 22-13 MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)

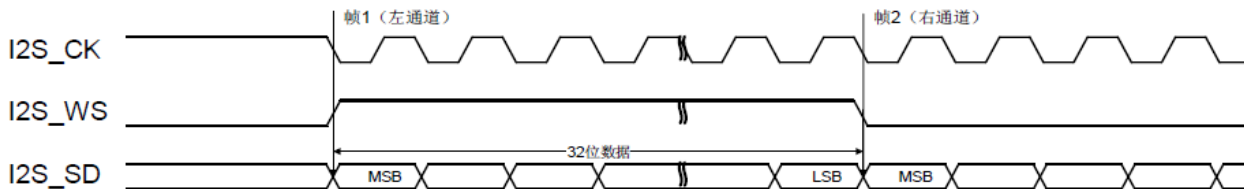


图 22-14 MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

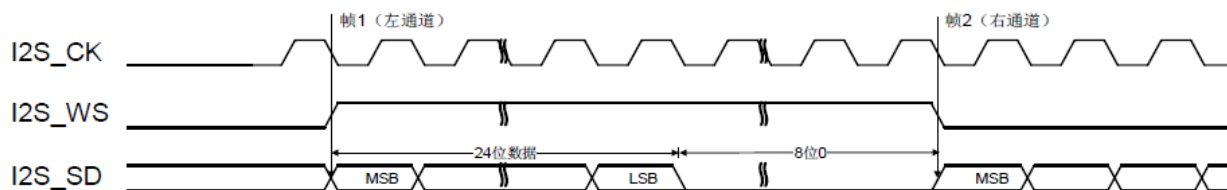


图 22-15 MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)

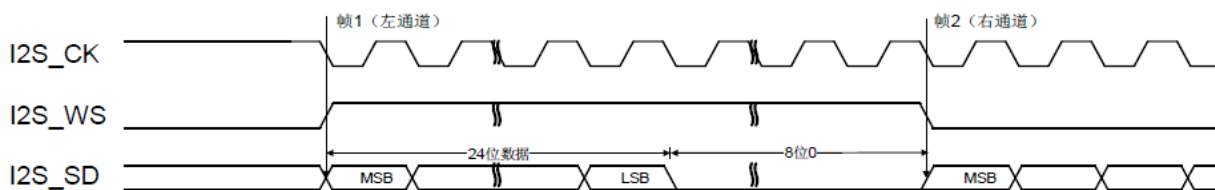


图 22-16 MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

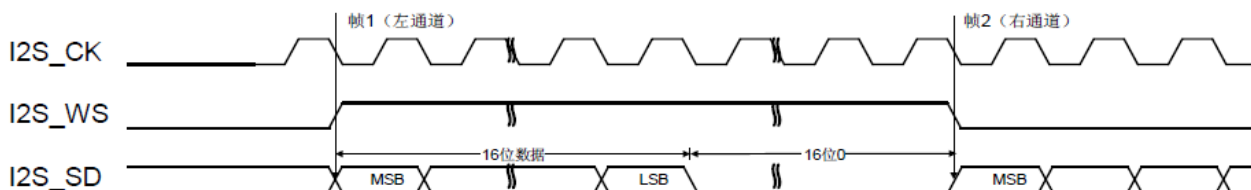
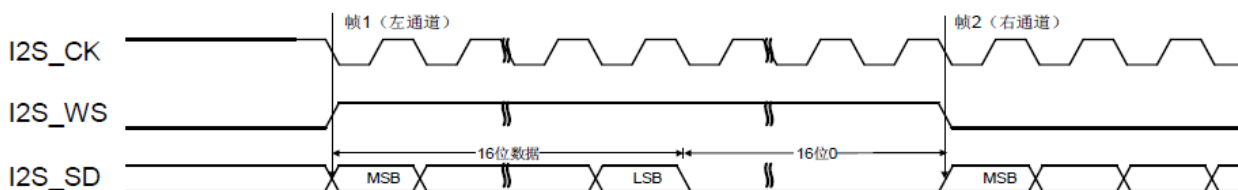


图 22-17 MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



### LSB 对齐标准

对于 LSB 对齐标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的下降沿变化。在通道长度与数据长度相同的情况下，LSB 对齐标准和 MSB 对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB 对齐标准的有效数据与最低位对齐，而 MSB 对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 22-18 LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

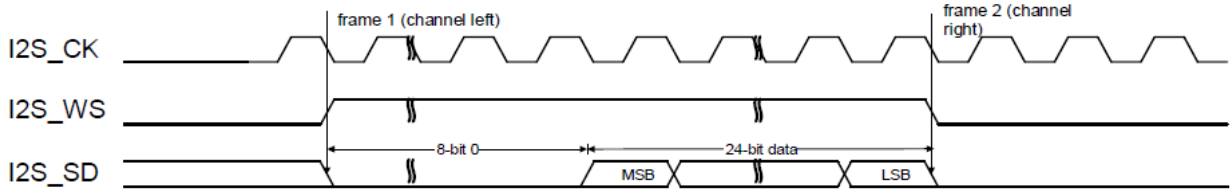


图 22-19 LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)

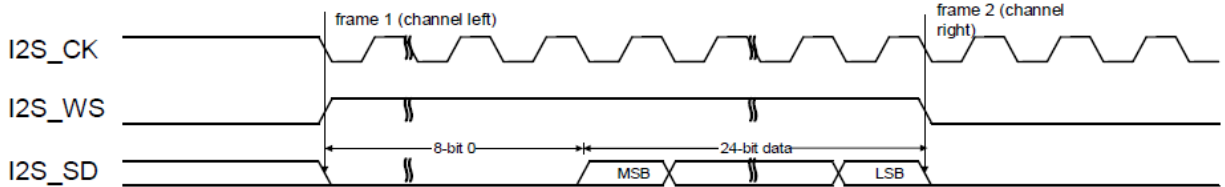


图 22-20 LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

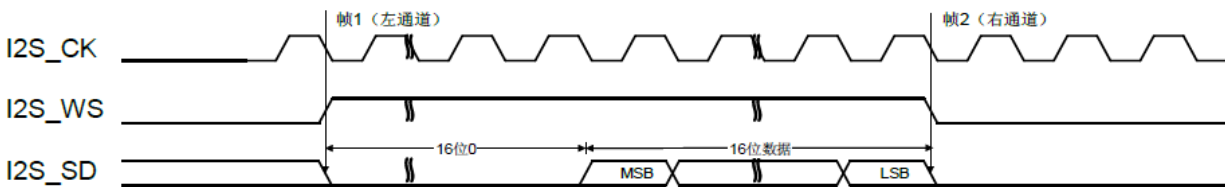
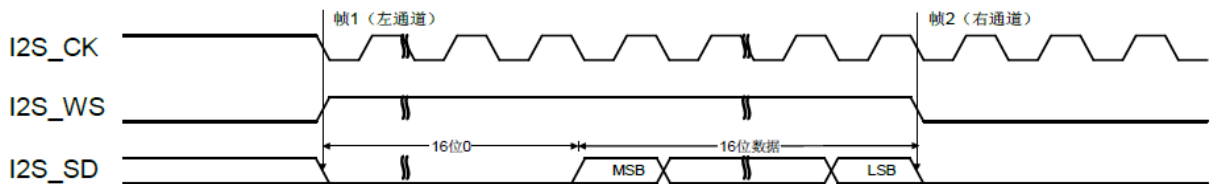


图 22-21 LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



## PCM 标准

对于 PCM 标准，I2S\_WS 和 I2S\_SD 在 I2S\_CK 的上升沿变化，I2S\_WS 信号表示帧同步信息。

图 22-22 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

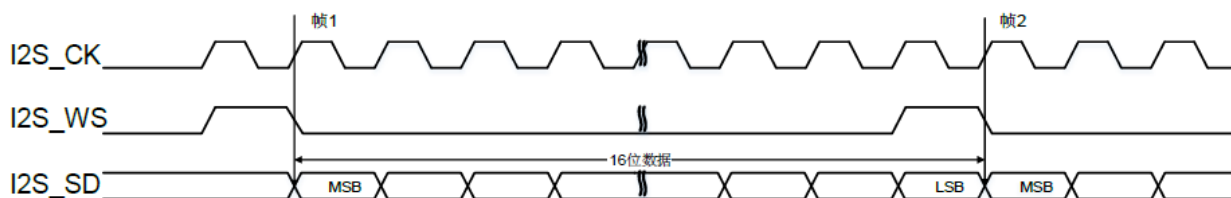


图 22-23 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

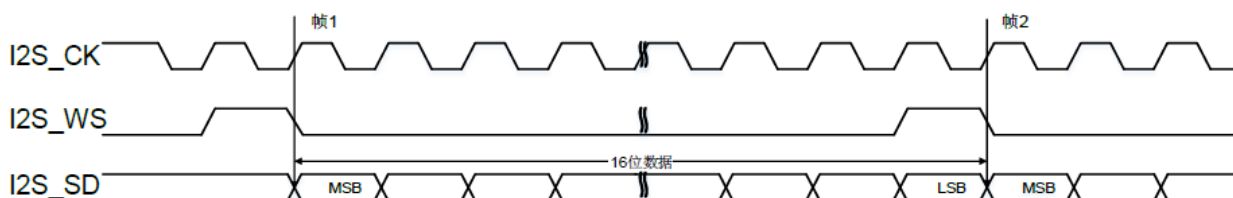


图 22-24 PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

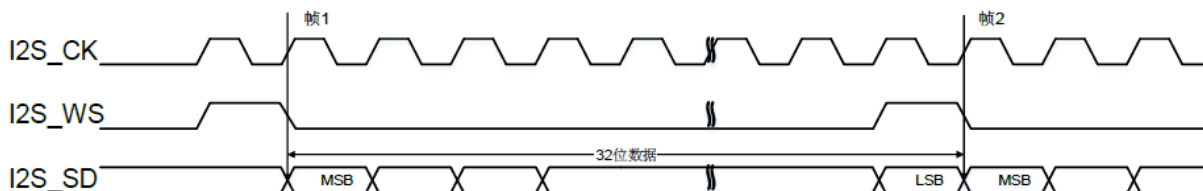


图 22-25 PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

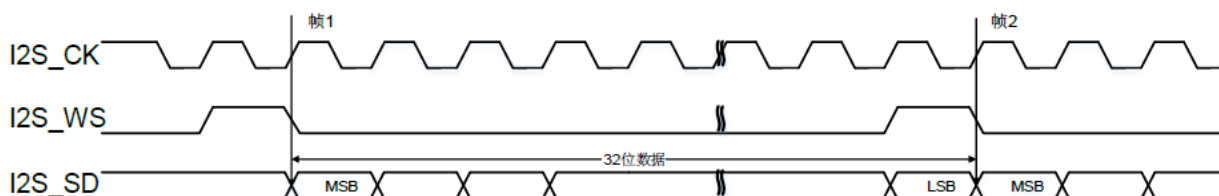


图 22-26 PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

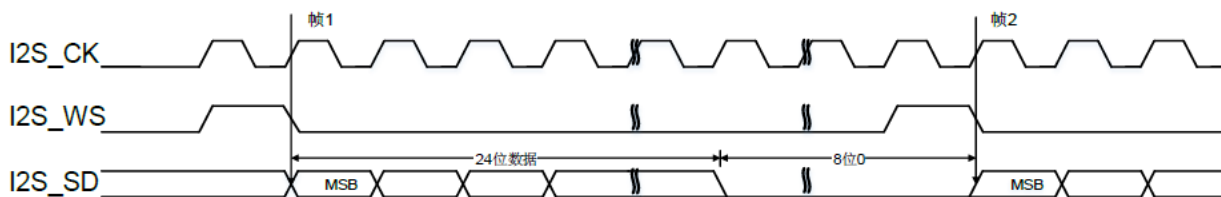


图 22-27 PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

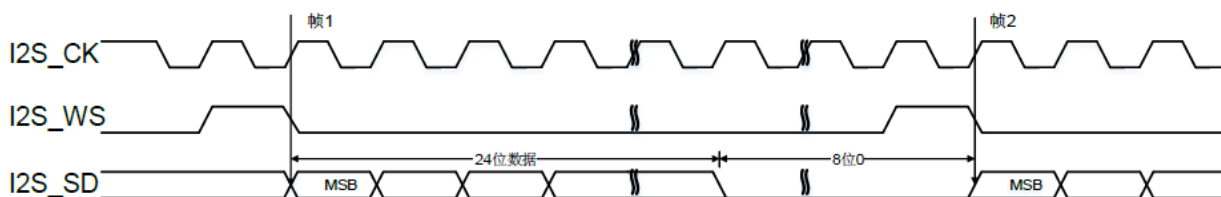


图 22-28 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

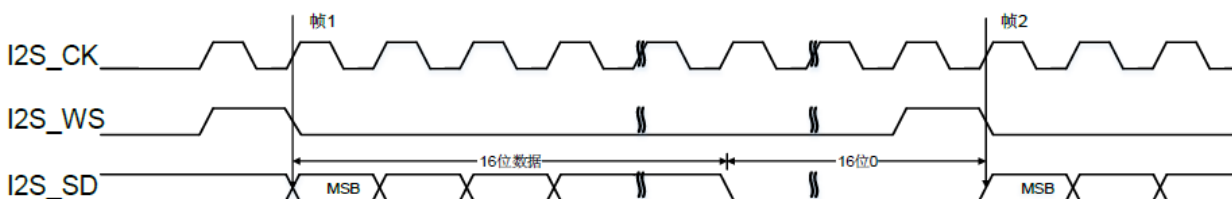
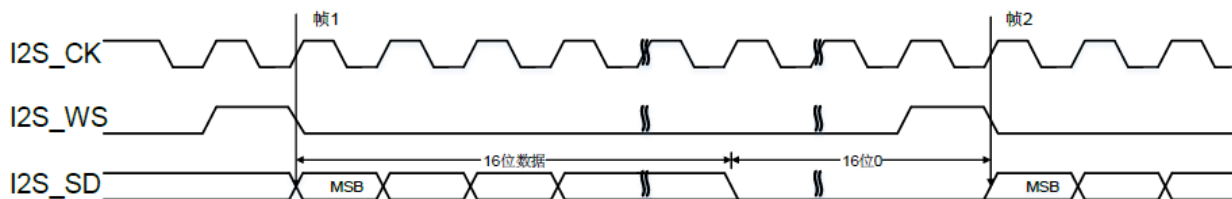


图 22-29 PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



PCM 标准长帧

图 22-30 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

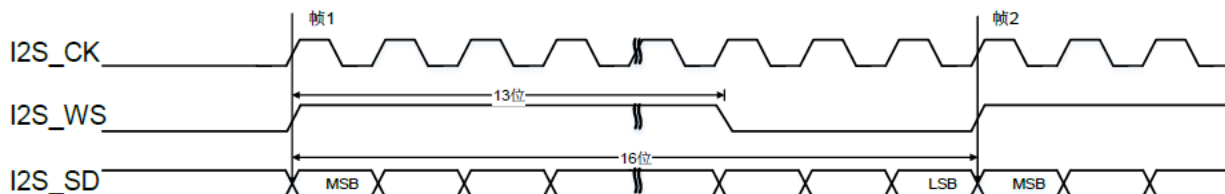


图 22-31 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

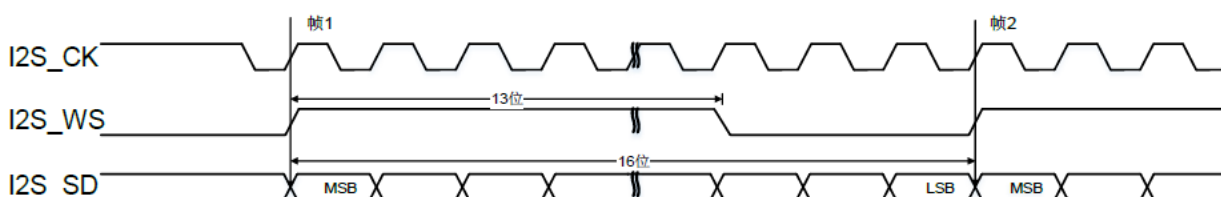


图 22-32 PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

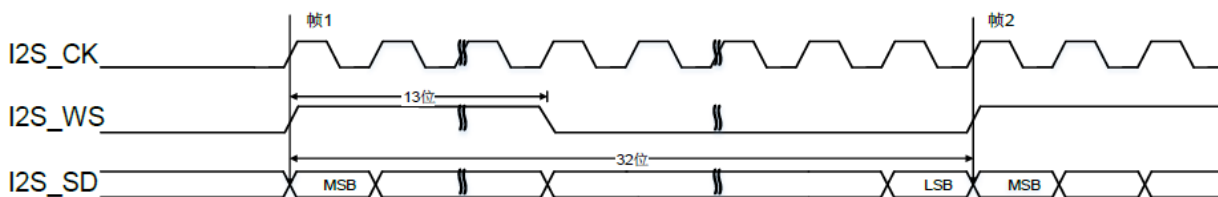


图 22-33 PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

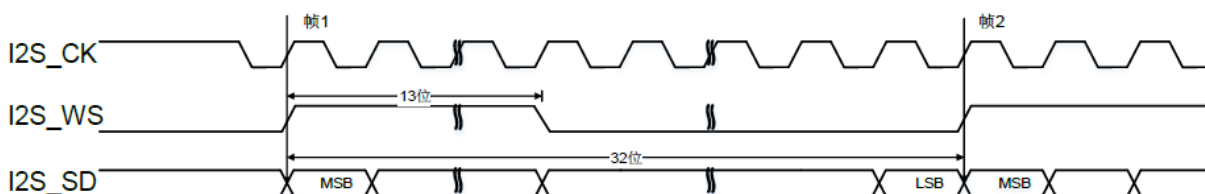


图 22-34 PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

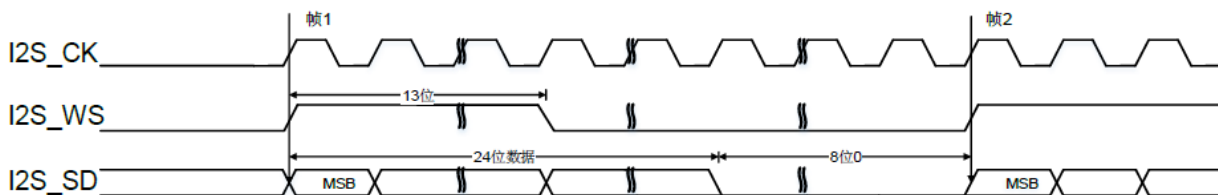


图 22-35 PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

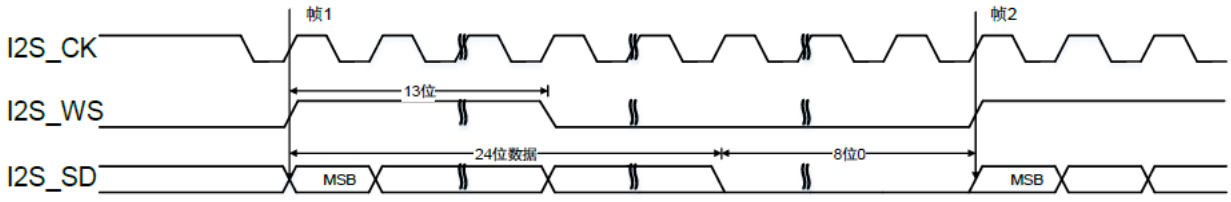


图 22-36 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

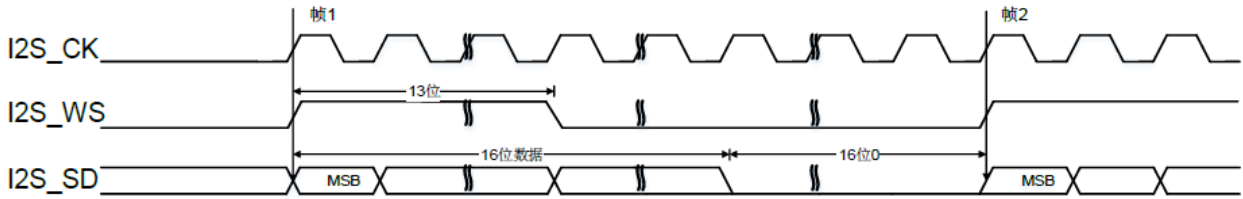
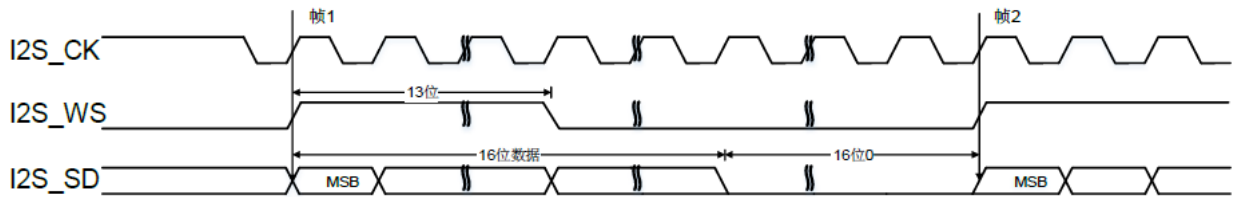
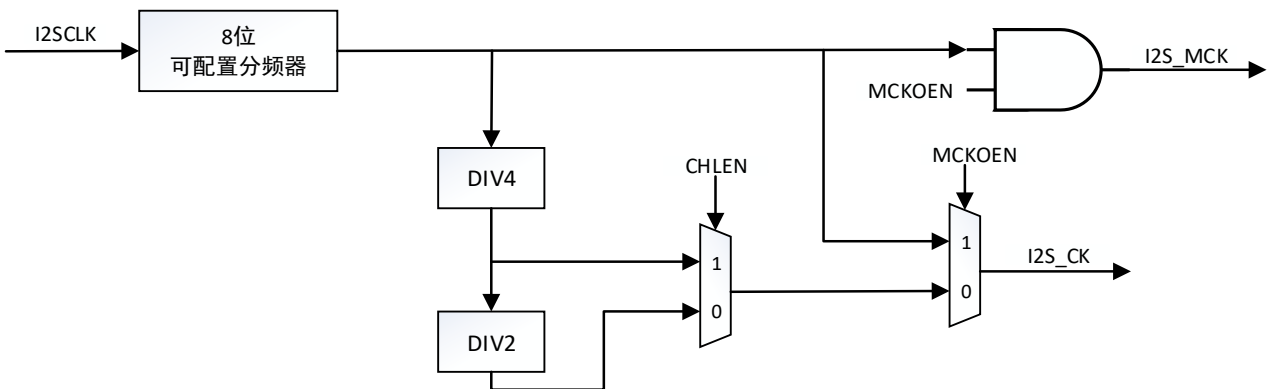


图 22-37 PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



### 22.3.3. I2S 时钟

图 22-38 I2S 时钟生成结构框图



I2S 接口时钟是通过 I2S\_PR 寄存器的 DIV 位, OF 位和 MCKOEN 位以及 I2S\_CR 寄存器的 CHLEN 位来配置的。I2S 的时钟源是 APB 总线时钟 (PCLK)。I2S 比特率可以通过表 22-1 所示的公式计算。

表格 22-1 比特率计算公式

MCKOEN	CHLEN	公式
0	0	$I2SCLK / (DIV * 2 + OF)$
0	1	$I2SCLK / (DIV * 2 + OF)$
1	0	$I2SCLK / (8 * (DIV * 2 + OF))$
1	1	$I2SCLK / (4 * (DIV * 2 + OF))$

音频采样率( $F_s$ )和 I2S 比特率的关系由如下公式定义:

$$F_s = \text{I2S 比特率} / (\text{通道长度} * \text{通道数})$$

所以, 为了得到期望的音频采样率, 时钟生成器需要按表 22-2 音频采样频率计算公式所列的公式进行配置

表格 22-2 音频采样率计算公式

MCKOEN	CHLEN	公式
0	0	$I2SCLK / (32 * (DIV * 2 + OF))$
0	1	$I2SCLK / (64 * (DIV * 2 + OF))$
1	0	$I2SCLK / (256 * (DIV * 2 + OF))$
1	1	$I2SCLK / (256 * (DIV * 2 + OF))$

## 22.4. 寄存器描述

I2S 寄存器基地址: 0x4001\_3000

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	中断使能控制寄存器
0x10	I2S_SR	状态寄存器

## 22.4.1. I2S 数据寄存器 I2S\_DR(偏移 00h)

比特	名称	属性	复位值	描述
31:0	I2S_DR[31:0]	RW	0x0	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。

## 22.4.2. I2S 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	I2SEN	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[1:0]	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[1:0]	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。

### 22.4.3. I2S 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	MCKOEN	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[8:0]	RW	0x0	预分频器的分频系数 实际分频系数是 $DIV * 2 + OF$ 。 DIV 不能为 0。 当 I2S 模式关闭时配置该位。

## 22.4.4. I2S DMA/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	DMATEN	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	DMAREN	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。

## 22.4.5. 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 该位由硬件置位, 可以通过写 0 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 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 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

## 22.5. 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。

### 22.5.1. 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。

### 22.5.2. 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。

### 22.5.3. 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。

### 22.5.4. 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。

## 23. 低功耗串口（LPUART）

### 23.1. 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。

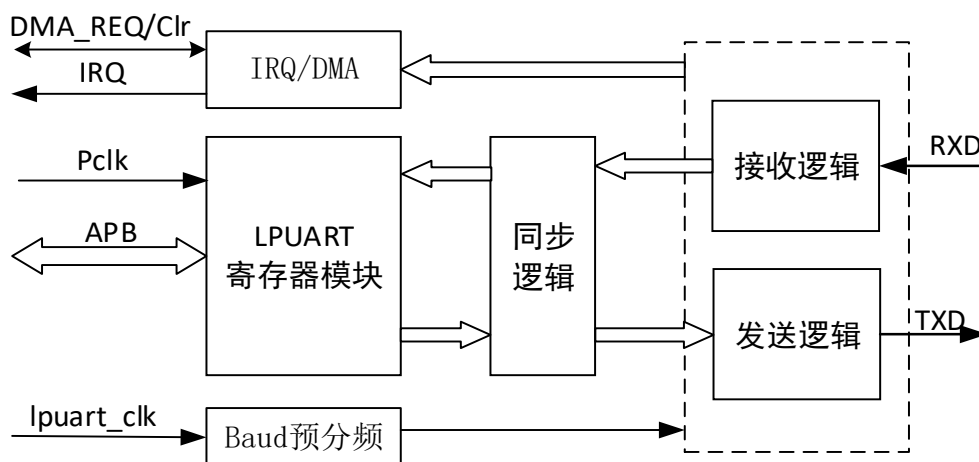
### 23.2. 主要特性

- 寄存器和通讯时钟独立
- 当使用 32.768KHz 作为工作时钟时，最大支持 9600 波特率
- 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
- 支持地址匹配功能，可用作多机通讯
- 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 工作

### 23.3. 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。整体框图如下：

图 23-1 LPURT 整体框图



### 23.3.1. 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

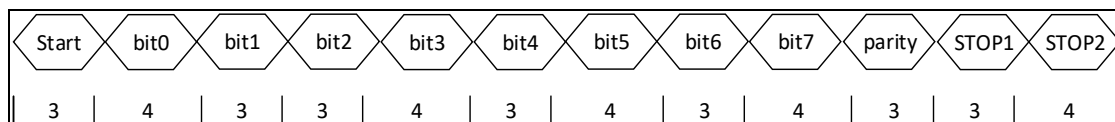
通过 SCU 的 CCR2 寄存器进行设置，当选择 RC32K 或 XTL 为时钟源时，可以作为低功耗串口进行工作，最大波特率支持 9600。当需要更高波特率时，可以选择 PCLK 分频作为时钟源，此时 LPUART 可以当做普通 UART 使用。

### 23.3.2. 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，每比特是否增加 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（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 23-2 LPURT BIT 分布



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。

波特率计算方法为：

- 1、波特率整数部分=（工作频率/需要波特率）的整数部分
- 2、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

如要波特率设为 115200，APB1 时钟为 90M，经过 32 分频的时钟作为产生 LPUART 工作时钟， $90M/32=2.8125M$  则：

- 1、波特率应为  $2812500/115200=24.4140625$
- 2、整数部分可以设置为 24 分频（寄存器值 23）
- 3、小数加权平均为 0.414,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  $11*0.414\approx 4$  个 1，可以设置为 12'b 0010 0100 1001。

### 23.3.3. 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，当接到到 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。

### 23.3.4. 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 DMA\_EN 位,使能 DMA 功能,利用 DMA 可以实现 LPUART 的发送/接受。

当传输完成 DMA 控制器指定的数据量时, DMA 控制器在该 DMA 通道的中断向量上产生一中断。

## 23.4. 寄存器描述

LPUART 寄存器基地址: 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

#### 23.4.1. 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	0x0	接收的数据

#### 23.4.2. 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	0x0	发送的数据

## 23.4.3. 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: 7bit
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

#### 23.4.4. 控制寄存器/LPUART\_CR (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	DMA_EN	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

#### 23.4.5. 波特率整数部分/LPUART\_IBAUD (偏移: 10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	RXSAM	RW	0x1	接收采样点设置, 一般设为 IBAUD>>1, 可以略微调整。
7:0	IBAUD	RW	0x2	波特率分频整数因子-1, 设置范围为 2~254 $IBAUD = (\text{integer}(F_{PCLK}/BAUD)) - 1$ , 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。

## 23.4.6. 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	0x0	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 (F <sub>PCLK</sub> /BAUD) - 1 - IBAUD

## 23.4.7. 中断使能寄存器/LPUART\_IE (偏移: 18h)

比特	名称	属性	复位值	描述
15:10	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

### 23.4.8. 状态寄存器 LPUART\_SR (偏移: 1Ch)

比特	名称	属性	复位值	描述
15:10	RSV	-	-	保留
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	RO	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

### 23.4.9. 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	0	匹配地址设置

## 23.5. 使用流程

### 23.5.1. 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

### 23.5.2. 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

## 24. 模数转换器（ADC）

### 24.1. 概述

ADC 控制模块通过 12 位逼近型模拟数字转换电路，实现模数转换控制，转换电路内嵌温度传感器实现温度检测。ADC 控制器支持单端信号转换和差分信号转换，最多达 21 个通道，包括 13 路外部普通通道，1 路外部 1/4 电源电压，2 路外部 BUFFER 通道，1 路内部 BUFFER 通道（内部 BGR 通道，仅校准时使用），1 路内部温度传感器，3 路内部 OPAMP 输出采样通道（使用的是普通通道 VIN7/VIN14/VIN15）。

ADC 控制器支持最大 16 次可设通道的规则转换，以及 1 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。

模拟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 控制模块支持 PCLK 的单时钟输入；ADC\_CLK 由内部分频产生。

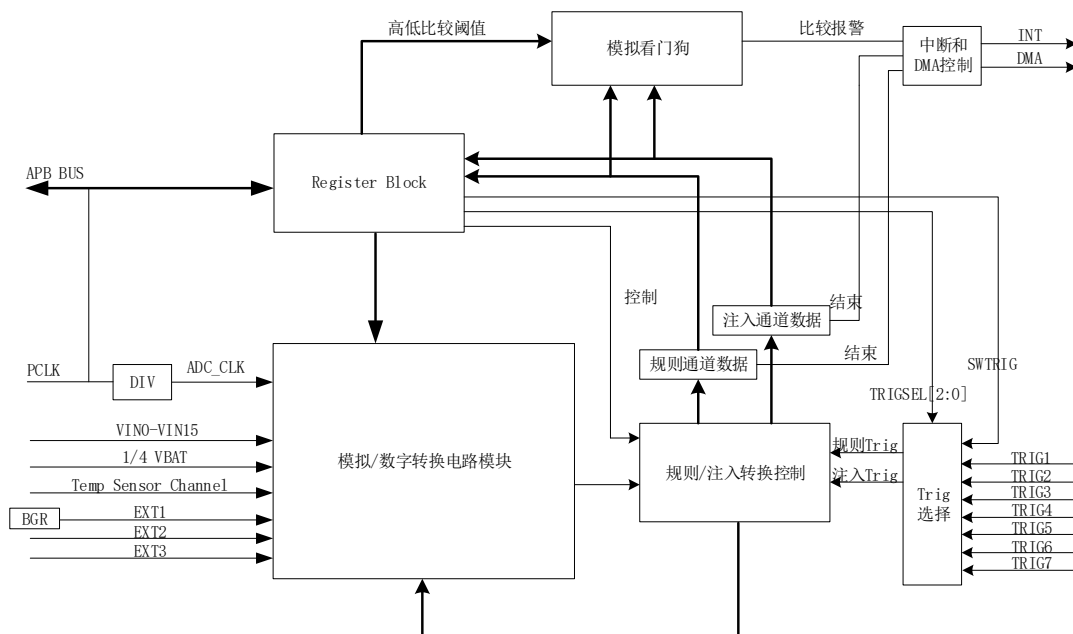
### 24.2. 主要特性

- 12 位分辨率
- 转换速率 2Msps
- 21 个可用通道，包括 13 路外部普通通道，1 路外部 1/4 电源电压，2 路外部 BUFFER 通道，1 路内部 BUFFER 通道（内部 BGR 通道，仅校准时使用），1 路内部温度传感器，3 路内部 OPAMP 输出采样通道（使用的是普通通道 VIN7/VIN14/VIN15）
- 支持单端信号转换和差分信号转换
- 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
- 单次、连续和间断转换模式
- 从序列 1 到序列 16 的规则扫描模式，序列通道可设
- 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
- APB 总线便于系统集成，同时实现高速的读写操作

- 支持过采样

## 24.3. 结构框图

图 24-1 ADC 框图



- Register Block

Register Block 通过 APB 接口实现对 ADC 模块内部的控制、中断和数据寄存器的读写译码访问。

- 转换控制

转换控制模块实现规则序列的转换，以及注入转换的插入。根据当前转换的通道号，通过对 ADC 转化电路的时序控制，完成一次转换，并把结果和通道号存到相应的数据寄存器。转换的触发源可以在外部 TRIG 和软件 TRIG 中选择，每个通道的转换定义独立的采样间隔时间。

- 模拟看门狗

模拟看门狗电路监控转化的结果，并与设置的高低阈值比较，超过阈值产生报警

- 中断控制

根据设置，可以产生单次转化结束中断、注入转换结束中断和模拟看门狗中断

- 时钟分频

DIV 模块实现 PCLK 分频产生 ADC\_CLK。在 ADC 控制模块内部，PCLK 和 ADC\_CLK 为全同频设计

## 24.4. 功能描述

### 24.4.1. 通道选择

一共有 21 个 ADC 通道(通道 0-通道 20)，见下表。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。
- 注入组只支持一个转换组成。注入通道在 ADC\_JSQR 寄存器中选择。

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。

表格 24-1 ADC 通道说明

通道号	差分模式	通道类型	说明备注
0 ~ 15	支持	普通	13 路普通外部输入通道 ADC_IN0~6/8~13  3 路内部 OPAMP 输出采样通道 ADC_IN7/14/15  ADC_IN7: OPAMP2 OUT  ADC_IN14: OPAMP3 OUT  ADC_IN15: OPAMP1 OUT
16	不支持	温度	内部模拟温度传感器
17	不支持	Vbat	1/4 Vbat, 外部输入 ADC_VBAT
18	不支持	带 BUFFER	内部 BGR 通道, 仅校准时使用
19~20	不支持	带 BUFFER	外部输入带 BUFFER 通道 ADC_EXT2/3

注：ADC 章节中提及的 BUFFER 是指缓冲放大器。

### 24.4.2. 单次转换模式

在规则组内，单次转换模式下，ADC 模块只执行一次转换。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：

- 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。

### 24.4.3. 连续转换模式

在规则组内，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。

### 24.4.4. 间断模式

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  $n(n \leq 16)$  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[3:0] 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。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。

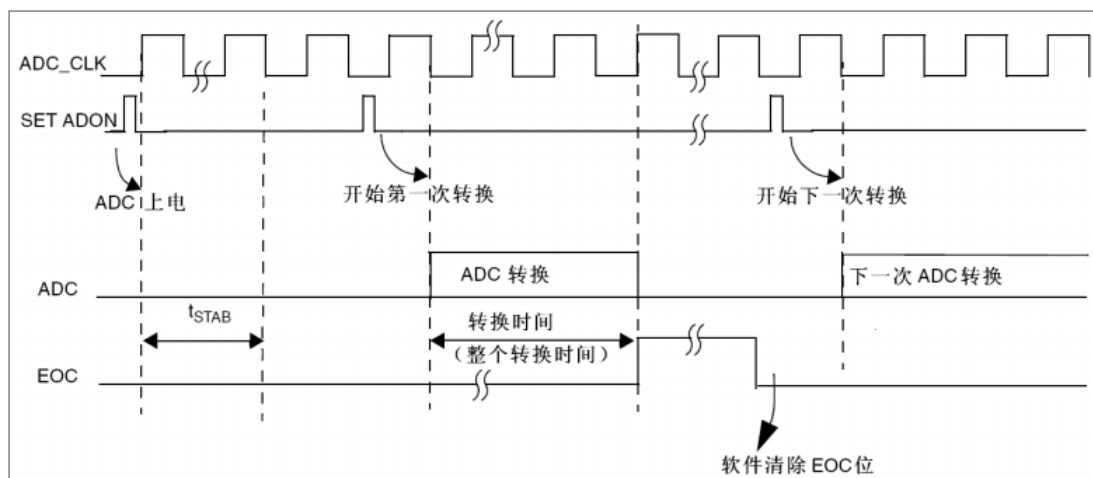
#### 24.4.5. 停止控制

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。

#### 24.4.6. 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。

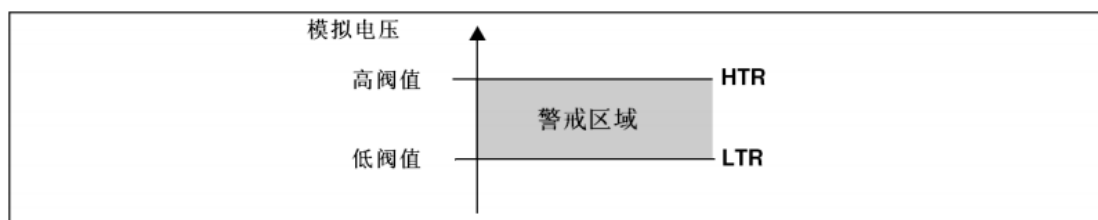
图 24-2 ADC 转换时序图



### 24.4.7. 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 DHT 位和 ADC\_LTR 寄存器的 DLT 位决定,通道应该设置为有符号数。通过设置 ADC\_IE 寄存器的 AWDIE 位以允许产生相应中断。

图 24-3 AWD 阈值图



表格 24-2 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

### 24.4.8. 注入通道

注入通道只能触发注入,不能自动插入到扫描序列中。

使能 ADC\_CR1 寄存器的 JEN 位,即可使用触发注入功能。

1. 利用外部触发或通过软件 TRIG 位，启动一组规则通道的转换。
2. 如果在规则通道转换期间产生注入触发，当前转换完成后(不等待采样间隔时间)，注入通道序列进行转换。
3. 结果放到注入数据寄存器 ADC\_JDR，JEOC 标志置位，如果设置了 JEOCIE 为，则产生 JEOC 中断。
4. 然后，恢复上次被中断的规则组通道转换。

#### 24.4.9. 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\_SMPR1/2/3 的 SMPx[3:0]位进行设置。

ADC 转换时间 TCONV 的计算如下：

$$T_{CONV} = T_{SAMP} + 17T_{ADC\_CLK}$$

例如：ADC\_CLK = 40MHz，TSAMP = 3TADC\_CLK，TCONV = (3+17) TADC\_CLK = 20 TADC\_CLK = 500ns。

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。

#### 24.4.10. 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[2:0]和 JEXTSEL[2:0]控制位允许应用程序选择 8 个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，只有它的上升沿可以启动转换。通道的触发选择，见下表：

表格 24-3 ADC 触发源选择表

名称	触发源	EXTSEL[2:0]	JEXTSEL[2:0]
SWSTART/JSWSTART	软件触发	000	000
TIG1	TIM1_TRGO	001	001
TIG2	TIM1_CC4	010	010

TIG3	TIM2_TRGO	011	011
TIG4	TIM3_TRGO	100	100
TIG5	TIM4_TRGO	101	101
TIG6	TIM6_TRGO	110	110
TIG7	EXTi Line 11	111	111

#### 24.4.11. 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。

#### 24.4.12. 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。

#### 24.4.13. 差分信号转换

ADC 比较电路支持差分信号转换，选择差分模式后，输入差分信号成对比较后再进行 AD 转换。差分信号分为 8 对：VIN0/VIN8、VIN1/VIN9、VIN2/VIN10、VIN3/VIN11、VIN4/VIN12、VIN5/VIN13、VIN6/VIN14、VIN7/VIN15。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。

为了得到正确的比较结果，差分转换时，根据应用选择结果为 16 位有符号数，在符号位为最高位 bit15，在未使用过采样功能的情况下，仅低 11 位数据有效。

#### 24.4.14. 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。

#### 24.4.15. 差分模式和有符号数

对应通道 ADC\_IN0~IN15 支持单端和差分输入两种模式，其他通道只支持单端模式；仅差分模式支持有符号数输出，其他通道只支持无符号。

单端模式下(DIFFx\_x=0)，输入 ADC\_INx 为无符号数，无符号输出(SIGNx\_x=0)为 DR[11:0]=ADC\_INx。

差分模式下(DIFFx\_x=1)，输入 ADC\_INx\_P 和 ADC\_INx\_N 对应有符号输出(SIGNx\_x=1)为 DR[15:0]=(ADC\_INx\_P-ADC\_INx\_N)/2。一个差分对中序号小的为 ADC\_INx\_P，序号大的为 ADC\_INx\_N。例如 ADC\_IN0=12'hfff, ADC\_IN8=12'h000，配置 DIFF0\_8=1/ SIGN0\_8=1，则 DR[15:0]=(ADC\_IN0-ADC\_IN8)/2=16'h07ff，反过来如果 ADC\_IN0=12'h000, ADC\_IN8=12'hfff，则 DR[15:0]=(ADC\_IN0-ADC\_IN8)/2=16'hf800。

#### 24.4.16. 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。

$$result = \frac{1}{M} \times \sum_{n=0}^{n=N-1} Conversion(t_n)$$

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。注入组过采样的触发模式由 JTOVS 控制，当 JTOVS =0 时，一次触发进行一次 ADC 转换；当 JTOVS =1 时，一次触发进行 N 次 ADC 转换。

#### 24.4.17. ADC 中断

规则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。

ADC\_SR 寄存器中有 2 个其他标志，但是它们没有相关联的中断：

- AFE\_VLDT (ADC 模拟 TS 转换结束信号)
- AFE\_EOC (ADC 模拟转换结束信号)

表格 24-4 ADC 中断控制表

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
模拟看门狗状态	AWD	AWDIE
规则转换数据溢出	OVERF	OVERFIE

## 24.5. 寄存器描述

ADC 寄存器基地址：0x4001\_2400

偏置	名称	属性	描述
0x00	ADC_SR	RO	ADC 状态寄存器
0x04	ADC_IE	RW	中断使能寄存器
0x08	ADC_CR1	RW	ADC 控制寄存器 1
0x0c	ADC_CR2	RW	ADC 控制寄存器 2
0x10	ADC_SMPR1	RW	ADC 转换时间寄存器 1
0x14	ADC_SMPR2	RW	ADC 转换时间寄存器 2
0x18	ADC_HTR	RW	ADC 看门狗高阈值寄存器
0x1c	ADC_LTR	RW	ADC 看门狗低阈值寄存器
0x20	ADC_SQR1	RW	ADC 规则序列寄存器 1
0x24	ADC_SQR2	RW	ADC 规则序列寄存器 2
0x28	ADC_SQR3	RW	ADC 规则序列寄存器 3
0x2c	ADC_JSQR	RW	ADC 注入序列寄存器
0x30	ADC_JDR	RO	ADC 注入数据寄存器
0x34	ADC_DR	RO	ADC 规则数据寄存器
0x38	ADC_DIFF	RW	ADC 单端/差分选择寄存器
0x3c	ADC_SIGN	RW	ADC 符号数选择寄存器
0x40	ADC_TSREF	RW	ADC 温度传感器/REF 寄存器
0x44	ADC_SMPR3	RW	ADC 转换时间寄存器 3

### 24.5.1. ADC 状态寄存器(ADC\_SR 偏移 00h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5	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。
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	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	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 0：转换未完成； 1：转换完成
1	EOC	RC_W1	0	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。
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

### 24.5.2. ADC 中断使能寄存器(ADC\_IE 偏移 04h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5	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。
4	OVERFIE	RW	0	允许产生规则通道数据溢出中断

				<p>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。</p> <p>0：溢出中断禁止</p> <p>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</p>
3	EOGIE	RW	0	<p>允许产生 EOG 中断</p> <p>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。</p> <p>0：禁止 EOG 中断；</p> <p>1：允许 EOG 中断。当硬件设置 EOG 位时产生中断</p>
2	JEOCIE	RW	0	<p>允许产生注入通道转换结束中断</p> <p>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。</p> <p>0：禁止 JEOC 中断；</p> <p>1：允许 JEOC 中断。当硬件设置 JEOC 位时产生中断</p>
1	EOCIE	RW	0	<p>允许产生 EOC 中断</p> <p>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。</p> <p>0：禁止 EOC 中断；</p> <p>1：允许 EOC 中断。当硬件设置 EOC 位时产生中断</p>
0	RSV	-	-	保留

### 24.5.3. ADC 控制寄存器 1 (ADC\_CR1 偏移 08h)

比特	名称	属性	复位值	描述
31:27	AWDJCH[4:0]	RW	00000	<p>注入通道模拟看门狗通道选择位</p> <p>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。</p> <p>00000：ADC 模拟输入通道 0</p> <p>00001：ADC 模拟输入通道 1</p> <p>.....</p> <p>01111：ADC 模拟输入通道 15</p> <p>10000：ADC 模拟温度传感器（输入通道 16）</p>

				<p>10001: ADC 模式 1/4 Vbat (输入通道 17)</p> <p>10010: 内部 BGR 通道, 仅校准时使用</p> <p>10011: ADC 模拟 BUF 输入通道 19 (ADC_EXT2)</p> <p>10100: ADC 模拟 BUF 输入通道 20 (ADC_EXT3)</p>
26:23	DISCNUM[3:0]	RW	0000	<p>规则通道中断模式通道计数</p> <p>软件通过这些定义在中断模式下, 收到外部触发后转换规则通道的数目。</p> <p>0000: 1 个通道</p> <p>0001: 2 个通道</p> <p>...</p> <p>1111: 16 个通道</p> <p>注: 需要小于或者等于 ADC_SQR1 中的 L 长度</p>
22	DISCEN	RW	0	<p>规则通道上的中断模式使能控制</p> <p>0: 规则通道组禁用中断模式</p> <p>1: 规则通道组使用中断模式</p>
21	CONT	RW	0	<p>连续转换</p> <p>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。</p> <p>0: 单次转换模式;</p> <p>1: 连续转换模式。</p>
20	SWSTART	RW	0	<p>开始转换规则通道</p> <p>如果在 EXTSEL[2:0]位中选择了 SWSTART 为触发事件, 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, 或者中断模式下, 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。</p> <p>0: 复位状态;</p> <p>1: 开始转换规则通道。</p>
19	JSWSTART	RW	0	<p>开始转换注入通道</p> <p>如果在 JEXTSEL[2:0]位中选择了 JSWSTART 为触发事件, 由软件设置该位用于启动一组注入通道的转换, 当 JEOC 置位时, 由硬件清除该位。</p> <p>0: 复位状态;</p> <p>1: 开始转换注入通道。</p>
18:16	EXTSEL[2:0]	RW	000	<p>选择启动规则通道组转换的外部触发</p>

				<p>选择用于启动规则通道组转换的外部触发</p> <p>000: SWSTART</p> <p>001: TIG1</p> <p>010: TIG2</p> <p>...</p> <p>111: TIG7</p>
15:13	JEXTSEL[2:0]	RW	000	<p>选择启动注入通道组转换的外部事件</p> <p>选择用于启动注入通道组转换的外部触发</p> <p>000: JSWSTART</p> <p>001: TIG1</p> <p>010: TIG2</p> <p>...</p> <p>111: TIG7</p>
12	DMA	RW	0	<p>直接存储器访问模式</p> <p>该位由软件设置和清除。详见 DMA 控制器章节。</p> <p>0: 不使用 DMA 模式;</p> <p>1: 使用 DMA 模式。</p>
11	AWDEN	RW	0	<p>在规则通道上开启模拟看门狗</p> <p>该位由软件设置和清除。</p> <p>0: 在规则通道上禁用模拟看门狗;</p> <p>1: 在规则通道上使用模拟看门狗。</p>
10	JAWDEN	RW	0	<p>在注入通道上开启模拟看门狗</p> <p>该位由软件设置和清除。</p> <p>0: 在注入通道上禁用模拟看门狗;</p> <p>1: 在注入通道上使用模拟看门狗。</p>
9	JEN	RW	0	<p>注入通道组转换使能位</p> <p>该位由软件设置和清除, 用于开启或关闭注入通道组转换使能</p> <p>0: 关闭注入通道组转换;</p> <p>1: 开启注入通道组转换</p>
8	AWDSGL	RW	0	<p>在一个单一的通道上使用看门狗</p> <p>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</p> <p>0: 在所有的通道上使用模拟看门狗;</p> <p>1: 在单一通道上使用模拟看门狗。</p>

7:5	RSV	-	-	保留
4:0	AWDCH[4:0]	RW	00000	<p>模拟看门狗规则通道选择位</p> <p>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。</p> <p>00000: ADC 模拟输入通道 0</p> <p>00001: ADC 模拟输入通道 1</p> <p>.....</p> <p>01111: ADC 模拟输入通道 15</p> <p>10000: ADC 模拟温度传感器（输入通道 16）</p> <p>10001: ADC 模式 1/4 Vbat（输入通道 17）</p> <p>10010: 内部 BGR 通道，仅校准时使用</p> <p>10011: ADC 模拟 BUF 输入通道 19（ADC_EXT2）</p> <p>10100: ADC 模拟 BUF 输入通道 20（ADC_EXT3）</p>

#### 24.5.4. 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	<p>ADC 快速转换模式(转换时间最短)</p> <p>0: 禁止</p> <p>1: 使能</p> <p>注：模拟温度传感器和 BUFFER 通道不支持快速转换模式</p>
26	AFE_RSTN	RW	0	<p>ADCAFE 复位信号,复位 AFE 内部数字部分。在 ADC_EN 使能后，等待 2us 释放（设为高）。</p> <p>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。</p>
25	JOVSE	RW	0	<p>注入组过采样功能使能</p> <p>0: 禁止</p> <p>1: 使能</p>
24	JTOVS	RW	0	<p>注入组过采样触发模式</p> <p>0: 一次触发进行 1 次 ADC 转换</p> <p>1: 一次触发进行 N 次 ADC 转换，N 是由 OVSr 决定的过采样率。</p>

23:20	OVSS[3:0]	RW	0000	<p>过采样移位系数</p> <p>0000: 不移位</p> <p>0001: 右移 1 位</p> <p>0010: 右移 2 位</p> <p>0011: 右移 3 位</p> <p>0100: 右移 4 位</p> <p>0101: 右移 5 位</p> <p>0110: 右移 6 位</p> <p>0111: 右移 7 位</p> <p>1000: 右移 8 位</p> <p>其他: 保留</p>
19:17	OVSR[2:0]	RW	000	<p>过采样率</p> <p>000: 2x</p> <p>001: 4x</p> <p>010: 8x</p> <p>011: 16x</p> <p>100: 32x</p> <p>101: 64x</p> <p>110: 128x</p> <p>111: 256x</p>
16	OVSE	RW	0	<p>规则组过采样功能使能</p> <p>0: 禁止</p> <p>1: 使能</p>
15:8	BUF_STIME	RW	0x40	<p>BUF 通道稳定时间，对于三个 BUF 通道 (10010~10100)，通道地址切换到转换开始需要额外的 3us 稳定时间。通过计数 ADC 时钟延迟产生，计数周期为 (BUF_STIME + 3) 个 ADC 时钟</p>
7	EN_BUF	RW	0	<p>ADC 模拟 BUF 使能，当 ADC_EN 为 0，内部模拟 BUF 也会无效。最大需要 8us 启动时间</p>
6:3	DIV [3:0]	RW	0001	<p>ADC_CLK 分频选择</p> <p>该位由软件设置和清除，决定 ADC_CLK 相对于 PCLK 的分频数</p> <p>0000: 不分频</p> <p>0001: 2 分频</p> <p>0010: 3 分频</p>

				... 1110: 15 分频 1111: 16 分频
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。

#### 24.5.5. 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: 3 周期 0001: 5 周期 0010: 7 周期 0011: 10 周期 0100: 13 周期 0101: 16 周期 0110: 20 周期 0111: 30 周期 1000: 60 周期 1001: 80 周期 1010: 100 周期 1011: 120 周期 1100: 160 周期 1101: 320 周期 1110: 480 周期 1111: 640 周期

#### 24.5.6. 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

#### 24.5.7. ADC 看门狗高阈值寄存器(ADC\_HTR 偏移 18h)

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，用有符号数表示。
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。

#### 24.5.8. ADC 看门狗低阈值寄存器(ADC\_LTR 偏移 1ch)

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。

#### 24.5.9. ADC 规则序列寄存器 1(ADC\_SQR1 偏移 20h)

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 (0~20)。 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 模拟温度传感器 (输入通道 16) 10001: ADC 模式 1/4 Vbat (输入通道 17) 10010: 内部 BGR 通道, 仅校准时使用 10011: ADC 模拟 BUF 输入通道 19 (ADC_EXT2) 10100: ADC 模拟 BUF 输入通道 20 (ADC_EXT3)
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

#### 24.5.10. ADC 规则序列寄存器 2(ADC\_SQR2 偏移 24h)

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

## 24.5.11. ADC 规则序列寄存器 3(ADC\_SQR3 偏移 28h)

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

## 24.5.12. ADC 注入通道寄存器(ADC\_JSQR 偏移 2ch)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:0	JSQ[4:0]	RW	00000	注入序列中转换 这些位由软件定义注入转换通道的编号(0~20)。

## 24.5.13. ADC 注入数据寄存器(ADC\_JDR 偏移 30h)

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。数据可以为有符号或者无符号

## 24.5.14. ADC 规则数据寄存器(ADC\_DR 偏移 34h)

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。数据可以为有符号或者无符号
--	--	--	--	-----------------------------------

#### 24.5.15. ADC 单端/差分选择寄存器 (ADC\_DIFF 偏移 38h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	DIFF7_15	RW	0	通道 7/15(VIN7/VIN15)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_14	RW	0	通道 6/14(VIN6/VIN14)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_13	RW	0	通道 5/13(VIN5/VIN13)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 4/12(VIN4/VIN12)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_11	RW	0	通道 3/11(VIN3/VIN11)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_10	RW	0	通道 2/10(VIN2/VIN10)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_9	RW	0	通道 1/9(VIN1/VIN9)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_8	RW	0	通道 0/8(VIN0/VIN8)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。

#### 24.5.16. ADC 符号数选择寄存器（ADC\_SIGN 偏移 3ch）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	SIGN7_15	RW	0	通道 7/15(VIN7/VIN15)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_14	RW	0	通道 6/14(VIN6/VIN14)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_13	RW	0	通道 5/13(VIN5/VIN13)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_12	RW	0	通道 4/12(VIN4/VIN12)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_11	RW	0	通道 3/11(VIN3/VIN11)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_10	RW	0	通道 2/10(VIN2/VIN10)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_9	RW	0	通道 1/9(VIN1/VIN9) 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_8	RW	0	通道 0/8(VIN0/VIN8)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。

#### 24.5.17. ADC 温度传感器/REF 寄存器 (ADC\_TSREF 偏移 40h)

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
26:25	VREFBI_SEL	RW	00	ADC 内嵌 VREF 电压选择 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
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
14:11	RSV	-	-	保留
10:9	ALG_MEAN[1:0]	RW	10	温度传感器转换次数选择，当通道选择温度传感器时， 转换时间变为转换次数*20(Teoc)个 ADC 时钟。 00:32 01:64 10:128 11:256
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
--	--	--	--	---

### 24.5.18. ADC 采样时间寄存器 3 (ADC\_SMPR3 偏移 44h)

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

## 24.6. 使用流程

### 24.6.1. ADC 操作流程

1. 设置 ADC 输入源:
2. 根据需要设置 ADC 输入源, 输入 GPIO 设为 GPIO\_ANALOG
3. 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)
4. 设置 ADC\_SQR1/2/3, 选择规则序列的长度和通道号
  5. 设置 ADC\_DIFF, 选择每个通道差分/单端模式
  6. 设置 ADC\_JSQR, 选择注入转换的通道号 (可选)
  7. 设置 ADC\_SMPR1/2/3, 配置每个通道的采样时间
  8. 设置 ADC\_HTR/LTR, 选择模拟看门狗的高低阈值 (可选)
  9. 设置 DMA 通道相关设置 (可选)
  10. 使能 ADC 转换电路(ADC\_EN)
  11. 等待外部 TRIG 或者触发软件 TRIG (SWSTART/ JSWSTART), 使能转换序列
  12. 等待相应的中断信号

### 24.6.2. 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 ALG\_MEAN 位, 选择合适的温度采样次数
4. 设置 ADC\_SQR1/2/3 寄存器, 选择通道 16
5. 设置 SMPR 时间

## 25. 数模转换器（DAC）

### 25.1. 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。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 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。

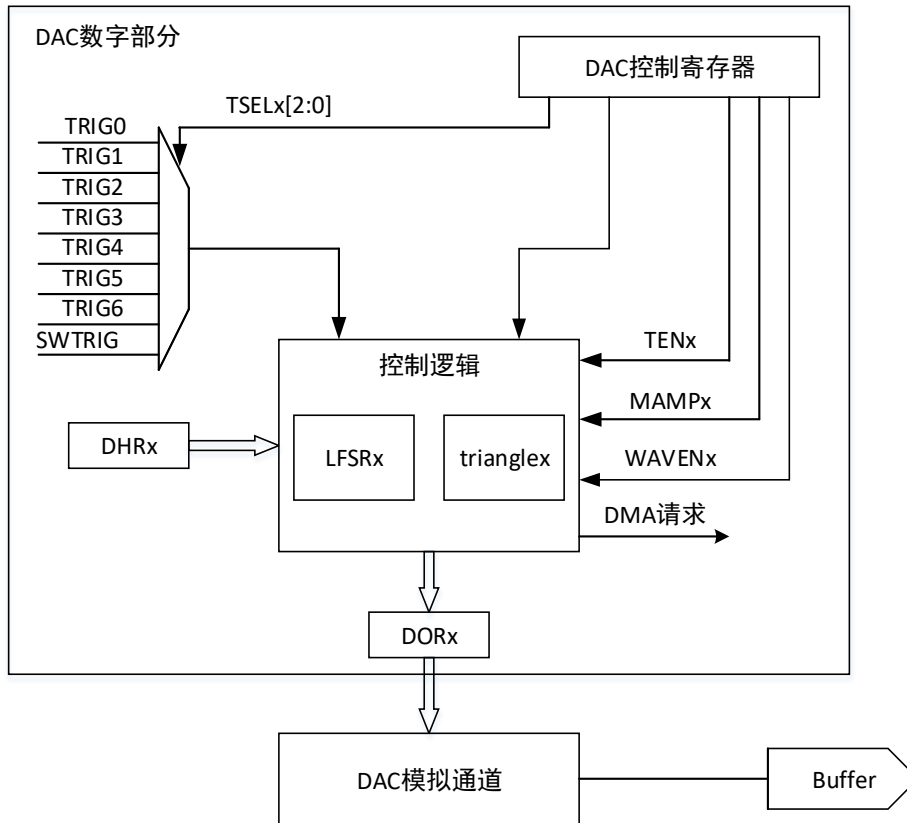
### 25.2. 主要特性

- 每个转换器对应 1 个输出通道，输出 BUFFER 可选
- 8 位或者 12 位分辨率
- 12 位模式下数据左对齐或者右对齐
- 同步更新功能
- 噪声波形生成
- 三角波形生成
- 双 DAC 通道同时或者分别转换
- 每个通道都有 DMA 功能
- 外部触发转换
- 输入参考电压 VREF+

### 25.3. 结构框图

下图为 DAC 控制模块单通道的框图

图 25-1 DAC 结构框图



## 25.4. 功能描述

### 25.4.1. 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 的寄存器开启或者关闭输出 BUFFER,默认为开启状态。

### 25.4.2. DAC 数据结构

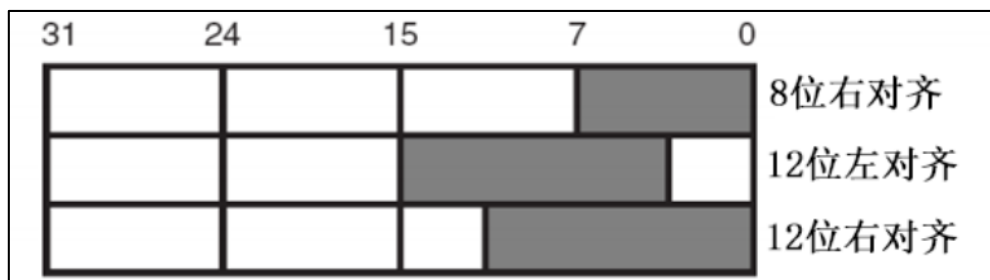
根据选择的配置模式,数据按照下文所述写入指定的寄存器:

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 25-2 DAC 通道模式的数据寄存器

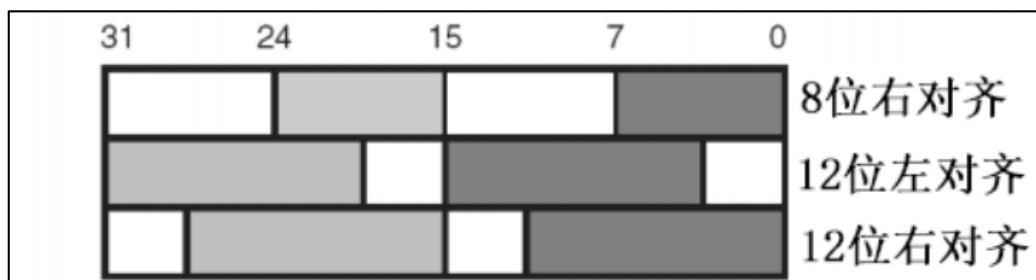


双 DAC 通道, 有 3 种情况:

- 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 数据写入寄存器 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 数据写入寄存器 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 25-3 双 DAC 通道模式的数据寄存器



### 25.4.3. 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：

$$\text{DAC 输出} = \text{VREF} * (\text{DOR} / 4095)。$$

#### 25.4.4. DAC 触发选择

如果  $\text{TEN}_x$  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  $\text{TSEL}_x[2:0]$  可以选择 8 个触发事件之一触发 DAC 转换。

表 25-1 DAC 触发源选择表

触发源	类型	$\text{TSEL}_x[2:0]$
TIMx_TRGO	内部定时器的触发信号	000
TIMx_TRGO		001
TIMx_TRGO		010
TIMx_TRGO		011
TIMx_TRGO		100
TIMx_TRGO		101
EXTI9	外部管脚	110
SWTRIG	软件触发位	111

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。

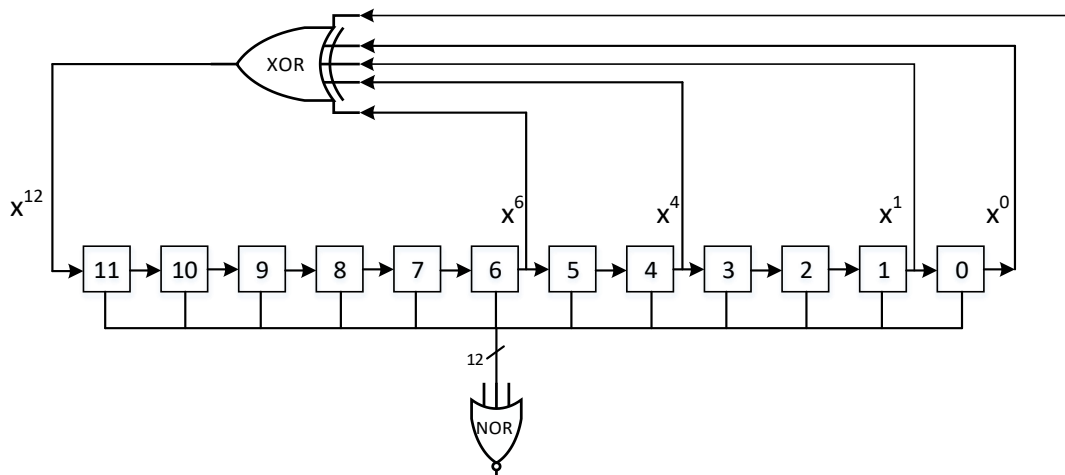
如果选择软件触发，一旦 SWTRIG 位置‘1’，转换即开始。在数据从 DAC\_DHRx 寄存器传送到 DAC\_DORx 寄存器后， SWTRIG 位由硬件自动清‘0’。

#### 25.4.5. 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。

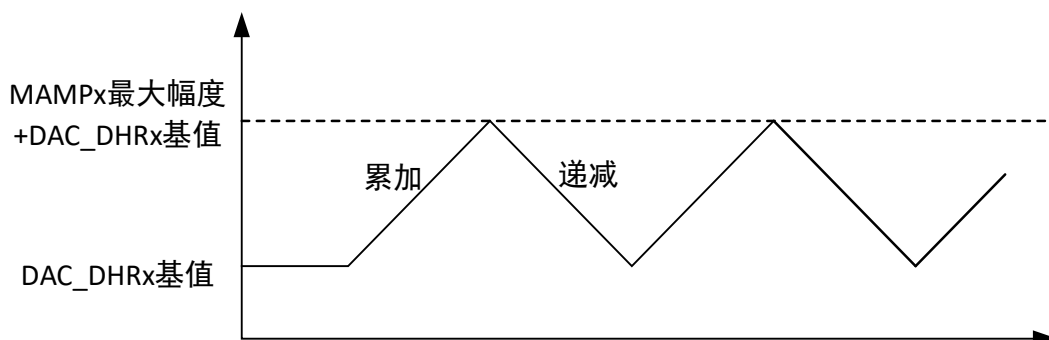
图 25-4 LFSR 噪声功能框图



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，这样的得到的 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'。

- 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 25-5 三角波噪声功能框图



将 WAVE<sub>x</sub>[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 TEN<sub>x</sub> 位为'1'。

#### 25.4.6. 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 DMAEN<sub>x</sub> 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\_DHR<sub>x</sub> 寄存器的数据被传送到 DAC\_DOR<sub>x</sub> 寄存器。

在双 DAC 模式下，如果 2 个通道的 DMAEN<sub>x</sub> 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 DMAEN<sub>x</sub> 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。

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 DMAUDR<sub>x</sub> (DAC\_SR 寄存器)。

#### 25.4.7. 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\_DOR<sub>x</sub> 的值将同时被更新。

有 3 个寄存器可被用于加载 DAC\_DHR<sub>x</sub> 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 TEN<sub>x</sub> 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 DMAEN<sub>x</sub> 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。

#### 25.4.8. 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



## 25.5. 寄存器描述

DAC 寄存器基地址：0x4000\_7400

偏置	名称	属性	描述
0x00	DAC_CR	RW	DAC 控制寄存器
0x04	DAC_SWTRIGR	RW	DAC 软件触发寄存器
0x08	DAC_DHR12R1	RW	DAC 通道 1 12 位右对齐数据保持寄存器
0x0c	DAC_DHR12L1	RW	DAC 通道 1 12 位左对齐数据保持寄存器
0x10	DAC_DHR8R1	RW	DAC 通道 1 8 位右对齐数据保持寄存器
0x14	DAC_DHR12R2	RW	DAC 通道 2 12 位右对齐数据保持寄存器
0x18	DAC_DHR12L2	RW	DAC 通道 2 12 位左对齐数据保持寄存器
0x1c	DAC_DHR8R2	RW	DAC 通道 2 8 位右对齐数据保持寄存器
0x20	DAC_DHR12RD	RW	双 DAC 12 位右对齐数据保持寄存器
0x24	DAC_DHR12LD	RW	双 DAC 12 位左对齐数据保持寄存器
0x28	DAC_DHR8RD	RW	双 DAC 8 位右对齐数据保持寄存器
0x2c	DAC_DOR1	RO	DAC 通道 1 数据输出寄存器
0x30	DAC_DOR2	RO	DAC 通道 2 数据输出寄存器
0x34	DAC_SR	RO	DAC 状态寄存器
0x38	DAC_CCR	RW	DAC 校准控制寄存器
0x3c	DAC_MCR	RW	DAC 模式控制寄存器
0x40	DAC_SHSR1	RW	DAC 通道 1 采样时间寄存器
0x44	DAC_SHSR2	RW	DAC 通道 2 采样时间寄存器
0x48	DAC_SHHR	RW	DAC 保持时间寄存器
0x4c	DAC_SHRR	RW	DAC 刷新时间寄存器

### 25.5.1. 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	0000	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	00	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; 1x: 使能三角波噪声发生器。
21:19	TSEL2[2:0]	RW	000	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。 000: TIM6 TRGO 事件; 001: TIM3 TRGO 事件 010: TIM7 TRGO 事件; 011: TIM15 TRGO 事件; 100: TIM2 TRGO 事件; 101: TIM1 TRGO 事件 110: 外部中断线 9; 111: 软件触发。 注意: 该位只能在 TEN1= 1(DAC 通道 1 触发使能)时有效
18	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

				<p>据在 1 个时钟周期后传入寄存器 DAC_DOR2;</p> <p>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。</p> <p>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。</p>
17	RSV	-	-	保留
16	EN2	RW	0	<p>DAC 模拟通道 2 使能</p> <p>0: 关闭 DAC 模拟通道 1;</p> <p>1: 使能 DAC 模拟通道 1。</p> <p>注: 需要设置 MODEx 后使能</p>
15	RSV	-	-	保留
14	CEN1	RW	0	DAC 通道 1 calibration 使能
13	DMAUDIE1	RW	0	<p>DAC 通道 1 Underrun 错误中断使能</p> <p>0: 关闭 DAC 通道 1 Underrun 错误中断</p> <p>1: 使能 DAC 通道 1 Underrun 错误中断</p>
12	DMAEN1	RW	0	<p>DAC 通道 1 DMA 使能</p> <p>0: 关闭 DAC 通道 1 DMA 模式;</p> <p>1: 使能 DAC 通道 1 DMA 模式</p>
11:8	MAMP1[3:0]	RW	0000	<p>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。</p> <p>0000: 不屏蔽 LFSR 位[0] / 三角波幅值等于 1;</p> <p>0001: 不屏蔽 LFSR 位[1:0] / 三角波幅值等于 3;</p> <p>0010: 不屏蔽 LFSR 位[2:0] / 三角波幅值等于 7;</p> <p>0011: 不屏蔽 LFSR 位[3:0] / 三角波幅值等于 15;</p> <p>0100: 不屏蔽 LFSR 位[4:0] / 三角波幅值等于 31;</p> <p>0101: 不屏蔽 LFSR 位[5:0] / 三角波幅值等于 63;</p> <p>0110: 不屏蔽 LFSR 位[6:0] / 三角波幅值等于 127;</p> <p>0111: 不屏蔽 LFSR 位[7:0] / 三角波幅值等于 255;</p> <p>1000: 不屏蔽 LFSR 位[8:0] / 三角波幅值等于 511;</p> <p>1001: 不屏蔽 LFSR 位[9:0] / 三角波幅值等于 1023;</p> <p>1010: 不屏蔽 LFSR 位[10:0] / 三角波幅值等于 2047;</p> <p>≥1011: 不屏蔽 LFSR 位[11:0] / 三角波幅值等于 4095</p>
7:6	WAVE1[1:0]	RW	00	<p>DAC 通道 1 LFSR 噪声/三角波生成使能</p> <p>00: 关闭波形生成;</p> <p>01: 使能 LFSR 噪声波形发生器;</p>

				1x: 使能三角波发生器。
5:3	TSEL1[2:0]	RW	000	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。 000: TIM6 TRGO 事件; 001: TIM3 TRGO 事件 010: TIM7 TRGO 事件; 011: TIM15 TRGO 事件; 100: TIM2 TRGO 事件; 101: TIM1 TRGO 事件 110: 外部中断线 9; 111: 软件触发。 注意: 该位只能在 TEN1= 1(DAC 通道 1 触发使能)时有效
2	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。
1	RSV	-	-	保留
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

### 25.5.2. DAC 软件触发寄存器(DAC\_SWTRIGR 偏移 04h)

比特	名称	属性	复位值	描述
31:2	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'。

### 25.5.3. 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
31:12	RSV	-	-	保留
11:0	DACC1DHR[11:0]	RW	0x0	DAC 通道 1 的 12 位右对齐数据, 表示 DAC 通道 1 的 12 位数据。

### 25.5.4. 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
31:16	RSV	-	-	保留
15:4	DACC1DHR[11:0]	RW	0x0	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

### 25.5.5. 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
31:8	RSV	-	-	保留
7:0	DACC1DHR[11:4]	RW	0x0	DAC 通道 1 的 8 位右对齐数据, 表示 DAC 通道 1 的高 8 位数据。

### 25.5.6. 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
----	----	----	-----	----

31:12	RSV	-	-	保留
11:0	DACC2DHR[11:0]	RW	0x0	DAC 通道 2 的 12 位右对齐数据，表示 DAC 通道 2 的 12 位数据。

### 25.5.7. DAC 通道 2 12 位左对齐数据保持寄存器(DAC\_DHR12L2 偏移 18ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:4	DACC2DHR[11:0]	RW	0x0	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

### 25.5.8. 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
31:8	RSV	-	-	保留
7:0	DACC2DHR[11:4]	RW	0x0	DAC 通道 2 的 8 位右对齐数据，表示 DAC 通道 2 的高 8 位数据。

### 25.5.9. 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	0x0	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	0x0	DAC 通道 1 的 12 位右对齐数据，表示 DAC 通道 1 的 12 位数据。

## 25.5.10. 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	0x0	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	0x0	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

## 25.5.11. 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	0x0	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	0x0	DAC 通道 1 的 8 位右对齐数据, 表示 DAC 通道 1 的高 8 位数据。

## 25.5.12. 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
31:12	RSV	-	-	保留
11:0	DACC1DOR[11:0]	RO	0x0	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。

## 25.5.13. 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
31:12	RSV	-	-	保留
11:0	DACC2DOR[11:0]	RO	0x0	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。
--	--	--	--	-----------------

### 25.5.14. 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	RC_W1		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。
28:25	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	RC_W1	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。
12:9	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

## 25.5.15. 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	0x0	DAC 通道 2 TRIM 值
15:5	RSV	-	-	保留
4:0	OTRIM1[4:0]	RW	0x0	DAC 通道 1 TRIM 值

## 25.5.16. DAC 模式控制寄存器(DAC\_MCR 偏移 3Ch)

比特	名称	属性	复位值	描述
31:19	RSV	-	-	保留
18:16	MODE2[2:0]	RW	000	DAC 通道 2 MODE 设置，在 DAC_ENx 使能前设置 具体定义参见 MODE1
15:3	RSV	-	-	保留
2:0	MODE1[2:0]	RW	000	DAC 通道 1 MODE 设置，在 DAC_ENx 使能前设置 正常模式： 000: DAC Buffer 使能，输出到管脚 001: DAC Buffer 使能，输出到管脚和内部 010: DAC Buffer 禁止，输出到管脚 011: DAC Buffer 禁止，输出到内部 采样保持模式： 100: DAC buffer 使能，输出到管脚 101: DAC Buffer 使能，输出到管脚和内部 110: DAC Buffer 禁止，输出到管脚和内部 111: DAC Buffer 禁止，输出到内部

注：DAC 章节中提及的 BUFFER 是指缓冲放大器。

## 25.5.17. 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	0x0	DAC 通道 1 采样时间周期设置 周期数为 TSAMPLE1+1
-----	---------------	----	-----	--------------------------------------

### 25.5.18. 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	0x0	DAC 通道 2 采样时间周期设置 周期数为 TSAMPLE2+1

### 25.5.19. 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	0x0	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	0x0	DAC 通道 1 保持时间周期设置 周期数为 THOLD1

### 25.5.20. 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	0x0	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	0x0	DAC 通道 1 刷新时间周期设置 周期数为 TREFRESH1

## 25.6. 使用流程

### 25.6.1. 单个 DAC 操作流程

1. 设置 DAC 通道的触发使能位  $TEN_x$  为'1'；
2. 通过设置  $TSEL_x[2:0]$ ，配置 DAC 通道的触发源；
3. 根据需要设置 DAC 通道的  $WAVEx[1:0]$ 位选择无噪、LFSR 噪声或者三角波噪声，并设  $MAMP_x[3:0]$ 为不同的 LFSR 屏蔽位或三角波幅值。
4. 通过设置  $DMAEN_x$  选择是否使能 DMA 模式
5. 通过 DMA 或者软件将 DAC 通道转换数据装入所需的 DHR 寄存器( $DHR12Rx$ 、 $DHR12Lx$  或  $DHR8Rx$ )。

### 25.6.2. 双 DAC 并发模式流程

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  $TSEL1[2:0]$ 和  $TSEL2[2:0]$ 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 噪声或者三角波噪声，并设  $MAMP_x[3:0]$ 为不同的 LFSR 屏蔽位或三角波幅值。
4. 通过设置  $DMAEN_x$  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  $DMAEN_x$  位被置位即可。
5. 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  $DHR8RD$ )。

## 26. 运算放大器（OPAMP）

### 26.1. 概述

OPAMP 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。

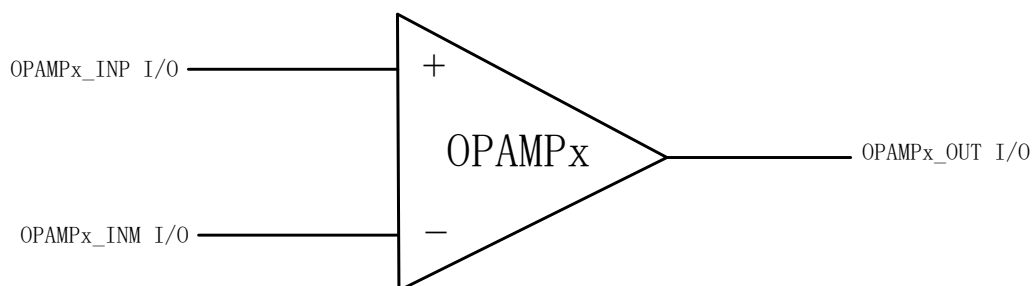
### 26.2. 主要特性

- 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；
- 运算放大器输出到管脚；
- 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管脚或者是 DAC 的输出；
- 运算放大器负端输入来自管脚、OPAMP 或者是 PGA 的反馈电阻网络输出。

### 26.3. 结构框图

下图为运放信号通路。

图 26-1 OPAMP 结构框图



## 26.4. 功能描述

### 26.4.1. 操作模式和校准

运放可工作在正常模式和校准模式。校准模式下软件可对运放的失调电压进行校准。

表格 26-1 运放工作模式控制和输出关系

Mode	Control bits			Output	
	EN	CAL NEN	CAL PEN	Vout	CALout flag
Normal	1	0	0	analog	0
		1	1	analog	0
Power down	0	x	x	z	0
Offset cal n	1	1	0	analog	x
Offset cal p	1	0	1	analog	x

### 26.4.2. 工作模式和负端输入

表格 26-2 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

注：若需要使用 PGA 模式，则只能选择 VINM1 作为负端输入。

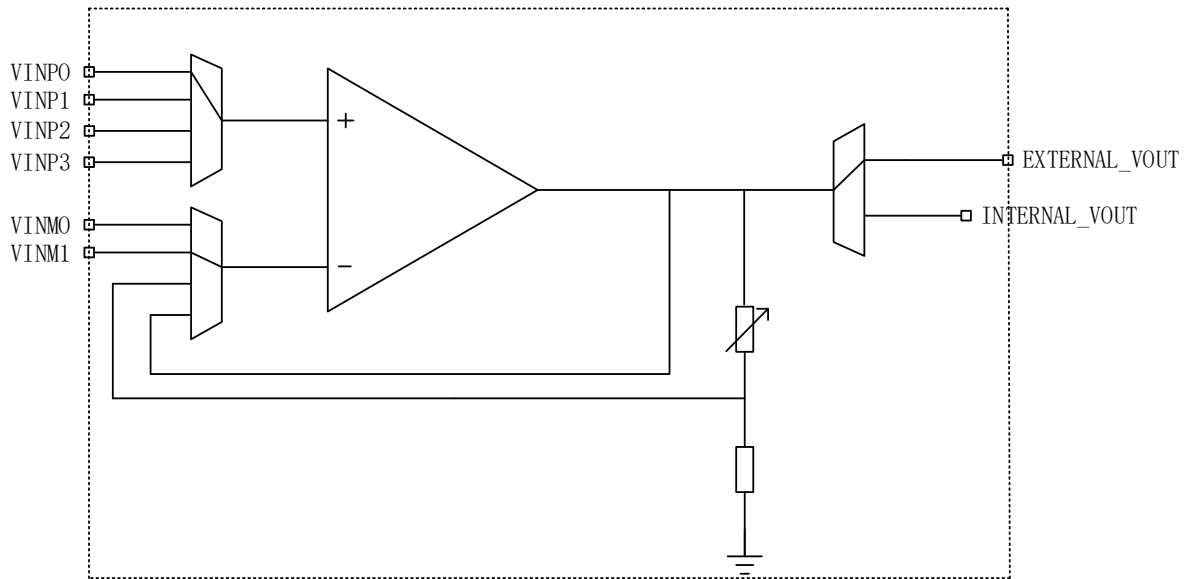
表格 26-3 负端输入管脚

OPAMP 编号	VINM1	VINM0
OPAMP1	PC5	PA3
OPAMP2	PC5	PA5
OPAMP3	PB10	PB2

### 26.4.3. 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：

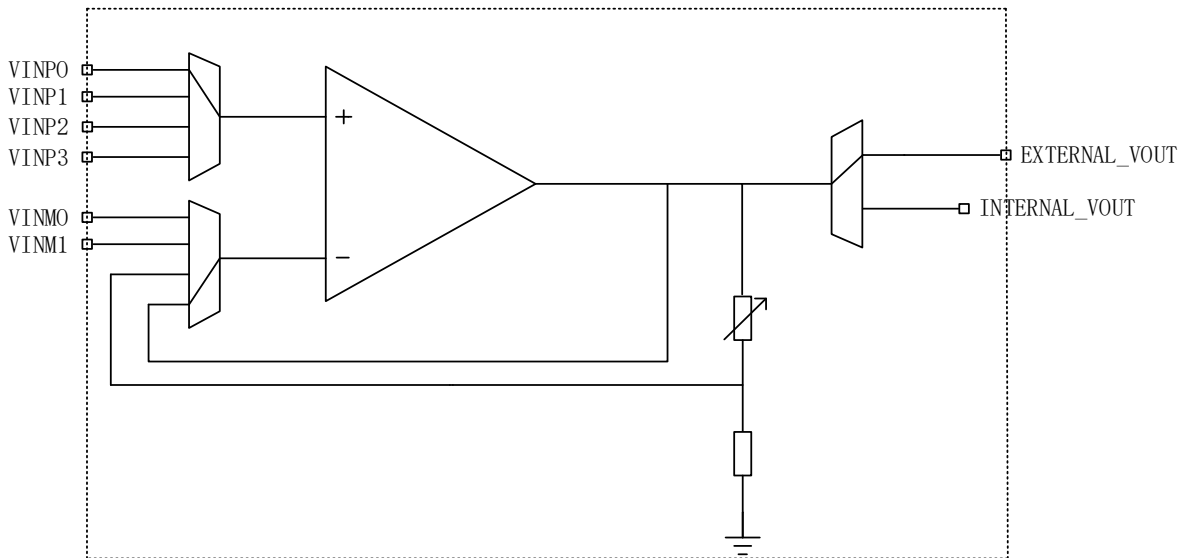
图 26-2 运放外置模式配置等效图



#### 26.4.4. 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 26-3 运放单位增益模式配置等效图



#### 26.4.5. 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 `OPAMPx_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 26-4 运放 PGA 同相模式配置等效图

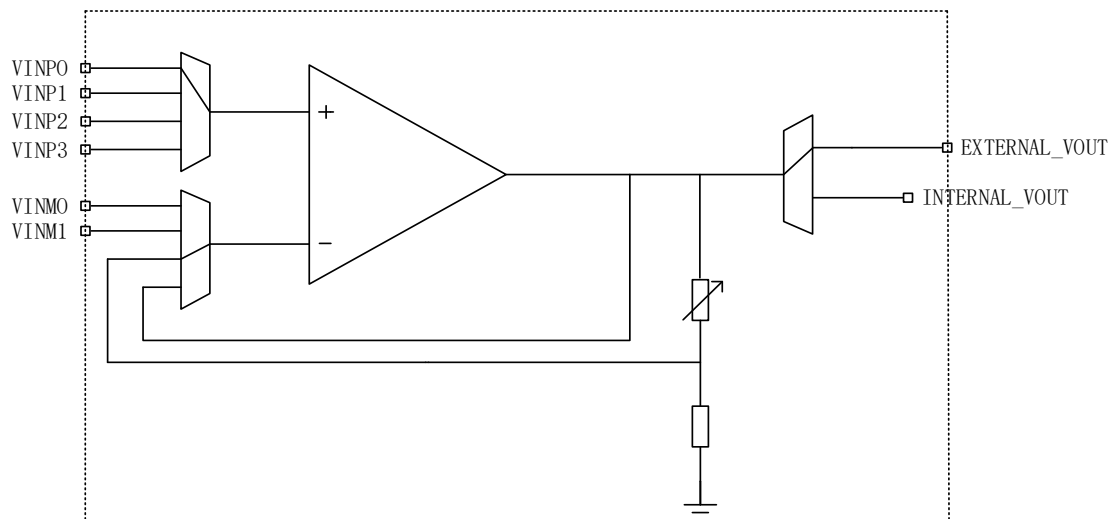
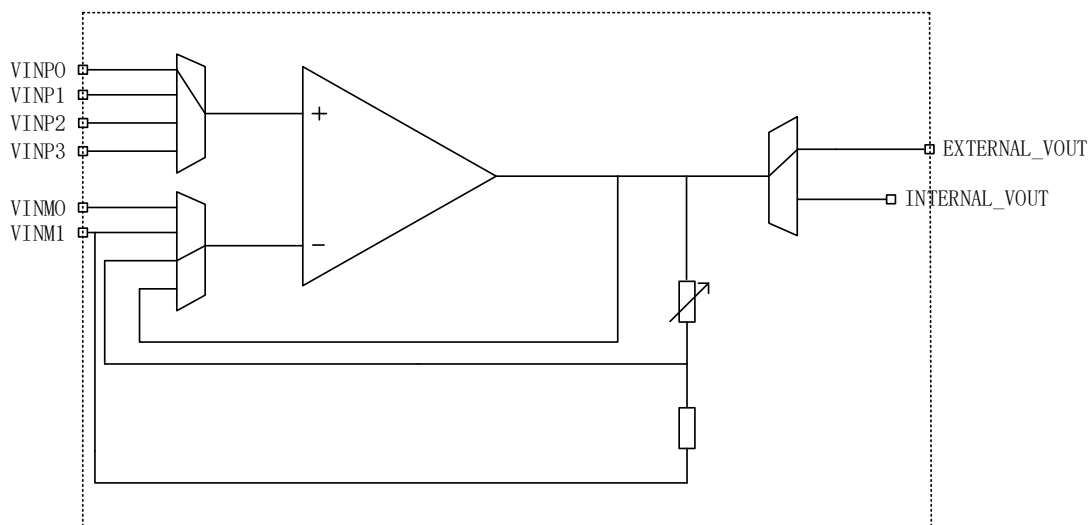


图 26-5 运放 PGA 反相模式配置等效图



## 26.5. 寄存器描述

OPAMP 寄存器基地址：0x40010300

偏置	名称	属性	描述
0x00	OPAMP1_CSR	RW	OPAMP1 控制/状态寄存器
0x04	OPAMP2_CSR	RW	OPAMP2 控制/状态寄存器
0x08	OPAMP3_CSR	RW	OPAMP3 控制/状态寄存器

### 26.5.1. OPAMP1 控制寄存器(OPAMP1\_CSR 偏移 00h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	OPAMP1_CSR 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 OPAMP1_CSR 寄存器 1: 禁止软件写入 OPAMP1_CSR 寄存器
30:29	REV	RO	00	保留位。
28	HSM	RW	0	驱动模式选择 低驱动电流消耗低，高驱动输出电流大。 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	00	极性选择 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	00	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	00	正端信号选择 00: PC4 01: PA3 10: PA7 11: DAC1 输出
15:14	OUT_SEL	RW	00	输出通道信号选择 00: 输出到 GPIO。 01: 输出到内部 ADC_IN15, 并且断开外部输出。 其他: 保留
13:9	TRIM_OSN	RW	0x0	OPAMP1 的 N 差分对管失调电压的修调值。
8:4	TRIM_OSP	RW	0x0	OPAMP1 的 P 差分对管失调电压的修调值。
3	CAL_OUT	RO	0	OPAMP1 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	OPAMP1 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	OPAMP1 的 P 差分对管失调电压修调使能 0: 禁止

				1: 使能
0	EN	RW	0	OPAMP1 使能位 0: 禁止 1: 使能

### 26.5.2. OPAMP2 控制寄存器(OPAMP2\_CSR 偏移 04h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	OPAMP2_CSR 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 OPAMP2_CSR 寄存器 1: 禁止软件写入 OPAMP2_CSR 寄存器
30:29	REV	RO	00	保留位。
28	HSM	RW	0	驱动模式选择 低驱动电流消耗低，高驱动输出电流大。 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	00	极性选择

				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	00	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	00	正端信号选择 00: PA7 01: PB0 10: OPAMP1 输出 11: DAC2 输出
15:14	OUT_SEL	RW	00	输出通道信号选择 00: 输出到 GPIO。 01: 输出到内部 ADC_IN7, 并且断开外部输出。 其他: 保留
13:9	TRIM_OSN	RW	0x0	OPAMP2 的 N 差分对管失调电压的修调值。
8:4	TRIM_OSP	RW	0x0	OPAMP2 的 P 差分对管失调电压的修调值。
3	CAL_OUT	RO	0	OPAMP2 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	OPAMP2 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	OPAMP2 的 P 差分对管失调电压修调使能 0: 禁止 1: 使能
0	EN	RW	0	OPAMP2 使能位 0: 禁止 1: 使能

## 26.5.3. OPAMP3 控制寄存器(OPAMP3\_CSR 偏移 08h)

比特	名称	属性	复位值	描述
31	LOCK	RW	0	OPAMP3_CSR 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 OPAMP3_CSR 寄存器 1: 禁止软件写入 OPAMP3_CSR 寄存器
30:29	REV	RO	00	保留位。
28	HSM	RW	0	驱动模式选择 低驱动电流消耗低，高驱动输出电流大。 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	00	极性选择 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	00	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	00	正端信号选择 00: PA4 01: PC4 10: PB0 11: DAC1 输出
15:14	OUT_SEL	RW	00	输出通道信号选择 00: 输出到 GPIO。 01: 输出到内部 ADC_IN14, 并且断开外部输出。 其他: 保留
13:9	TRIM_OSN	RW	0x0	OPAMP3 的 N 差分对管失调电压的修调值。
8:4	TRIM_OSP	RW	0x0	OPAMP3 的 P 差分对管失调电压的修调值。
3	CAL_OUT	RO	0	OPAMP3 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	OPAMP3 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	OPAMP3 的 P 差分对管失调电压修调使能 0: 禁止 1: 使能
0	EN	RW	0	OPAMP3 使能位 0: 禁止 1: 使能

## 26.6. 使用流程

### 26.6.1. 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。

### 26.6.2. 使用修调功能流程

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。

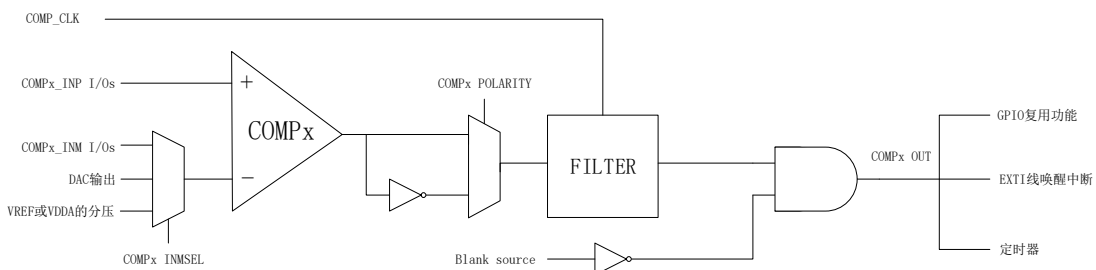
### 27.2. 主要特性

- 支持电压比较功能；
- 比较器负端输入可选择 I/O 或内部参考电压作为输入；
- 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；
- 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



## 27.4. 功能描述

### 27.4.1. 负端输入

负端输入可以选择来自外部, 内部 DAC 输出或者来自内部基准分压。通过配置 COMP\_CR 的 INMSEL 位域选择。

选择内部基准分压时, 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 FLTTIME 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 SCU 模块中的 CCR2 寄存器, 通过设置其中的 FLTCLK\_SEL 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。

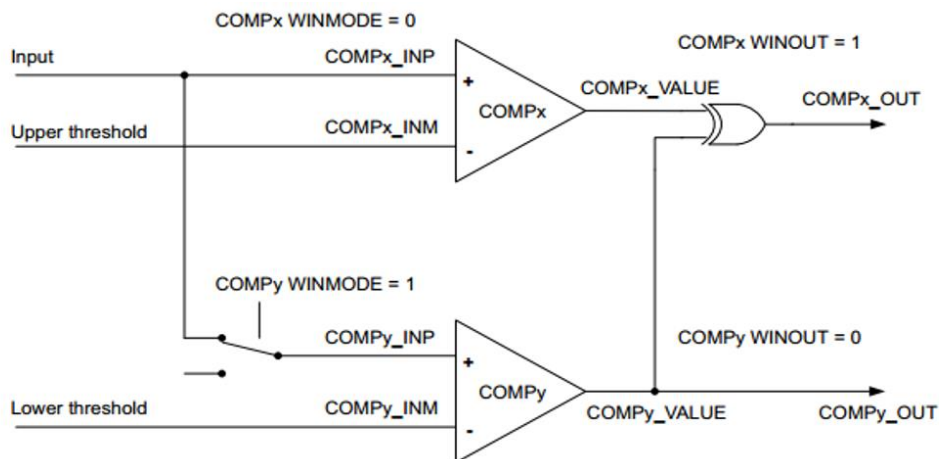
### 27.4.6. 窗口模式级联

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 COMP<sub>x</sub> 的输出设置 WINOUT 设为 1，此时输出为 VCOUT1 XOR VCOUT2；
- 需将 COMP<sub>y</sub> 的正端输入设置 WINMODE 设为 1，此时 COMP<sub>y</sub> 的正端输入连接到 COMP<sub>x</sub> 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_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	BLANKTIME	RW	00	BLANK WINDOW 的时间配置，宽度按 PCLK 计算。 00：32 个周期 01：64 个周期 10：128 个周期 11：256 个周期
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
15:12	BLANKSEL	RW	0000	比较器 1 切断源选择 0000: 不切断 xxx1: TIM1 OC4 xx1x: TIM2 OC3 x1xx: TIM3 OC3 1xxx: TIM15 OC2
11:8	INPSEL	RW	0000	比较器 1 正端信号选择 0000: PC4 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: 14mV 101: 24mV 110: 34mV 111: 44mV
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	BLANKTIME	RW	00	BLANK WINDOW 的时间配置，按 PCLK 计算。 00: 32 个周期 01: 64 个周期 10: 128 个周期 11: 256 个周期
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
15:12	BLANKSEL	RW	0000	比较器 2 切断源选择 0000: 不切断 xxx1: TIM1 OC4 xx1x: TIM2 OC3 x1xx: TIM3 OC3 1xxx: TIM15 OC2
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 100：14mV 101：24mV 110：34mV 111：44mV
0	EN	RW	0	比较器 2 使能位 0：禁止 1：使能

### 27.5.3. COMP 状态寄存器(COMP\_SR 偏移 08h)

比特	名称	属性	复位值	描述
31: 4	REV	RO	0x0	保留位。
3	VCOUT2_ORG	RO	0	比较器 2 原始输出状态
2	VCOUT1_ORG	RO	0	比较器 1 原始输出状态
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. 清除控制寄存器 COMP\_CR 的 LOCK 位解锁寄存器。
3. 配置比较器的正端信号选择 (INPSEL) 和负端信号选择 (INMSEL)。
4. 设置窗口模式正端输入选择 WINMODE (窗口模式可选)。
5. 设置输出模式 WINOUT (窗口模式可选)。
6. 设置输出极性 POLARITY。
7. 设置 CRV\_SEL 选择基准分压源 (内部基准源可选)。
8. 设置 CRV\_EN 使能基准分压 (基准分压可选)。
9. 设置分压系数 CRV\_CFG (基准分压可选)。
10. 设置 FLTEN 使能滤波功能 (滤波可选)。
11. 设置滤波时间配置 (FLTTIME) (滤波可选)。
12. 设置 HYS 配置迟滞比较窗口值 (迟滞比较可选)。
13. 设置 BLANKSEL 选择切断源 (切断可选)。
14. 设置和切断源对应的系统定时器实现触发功能 (切断可选)。
15. 设置 EN 使能比较器。
16. 设置控制寄存器 COMP\_CR 的 LOCK 位锁定寄存器。
17. 在 VOUT 端测量或者在 COMP\_SR 状态寄存器中读取输出信号。
18. 如需将输出用作定时器刹车输入功能, 请在定时器刹车输出相关寄存器中配置 (可选)。
19. 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，该芯片上集成了 2 路 CAN 模块。

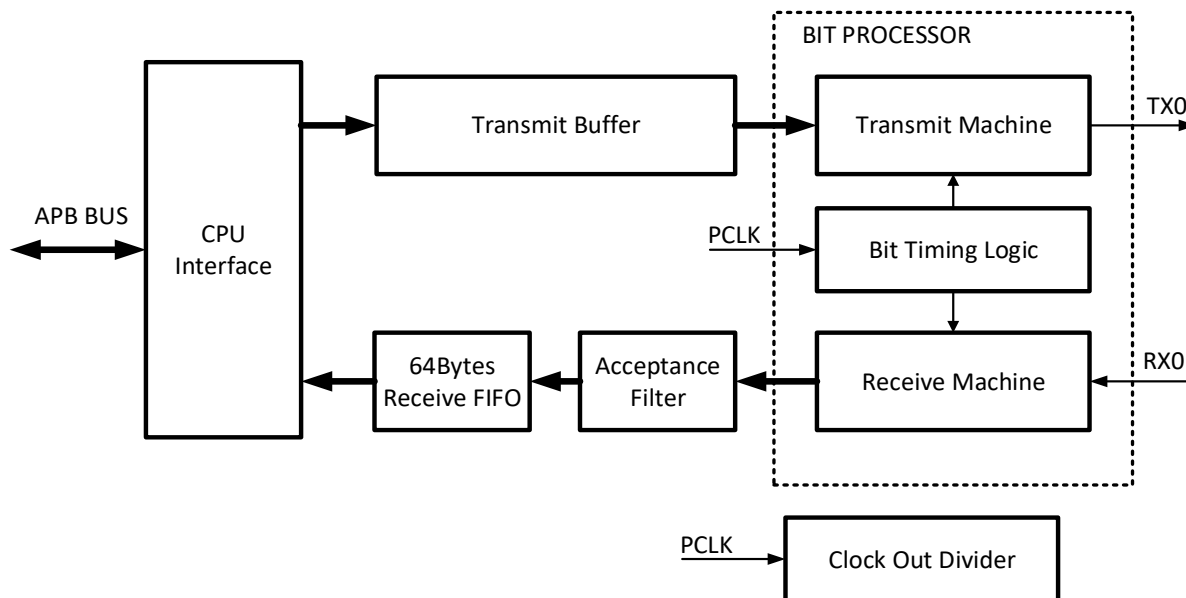
### 28.2. 主要特性

- 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。
- 通过 BOSCH CAN2.0 测试。

## 28.3. 功能描述

### 28.3.1. 功能框图

图 28-1 内部功能框图



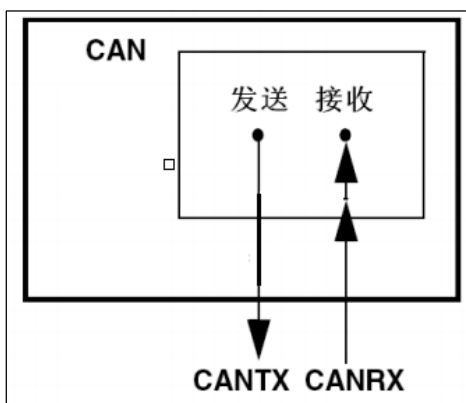
注：CPU 通过独立的地址线、输入数据线、输出数据线来访问 CAN 模块。发送的数据被放到 TxBuff 中去，并由发送器发送出去。接收的数据先通过接收滤波器过滤以后再放到 RxBuff 中。CPU 通过一个 13bytes 宽度的数据接口访问 RxFIFO。RxFIFO 总共有 64 个字节的缓存，最多可以一次性存储 5 组扩展帧，1 组放入到 RxBuff 中读取，4 组在 RxFIFO 中等待读取。Bit-Timing-Logic 模块负责用来产生波特率。TX0 用于发送，RX0 用于接收。

### 28.3.2. 操作模式

CAN 有两种工作模式：

- **Operating Mode:** 时可以正常发送接收 CAN 总线数据  
进入 Operating Mode 的方式是清除 MOD 的 BIT0。

图 28-2 正常模式示意图



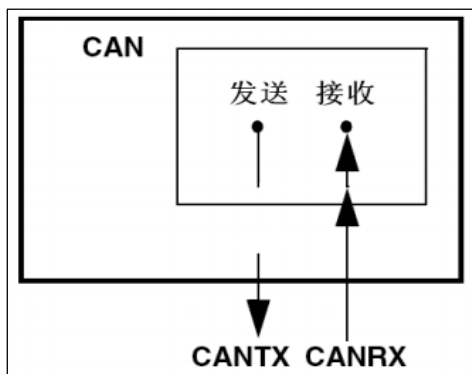
- **Reset Mode:** 可以修改时间参数和报文过滤参数。

进入 Reset Mode 的方式有两种：第一种就是执行一次硬件复位，第二种方式是写 MOD 寄存器的 BIT0。

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。

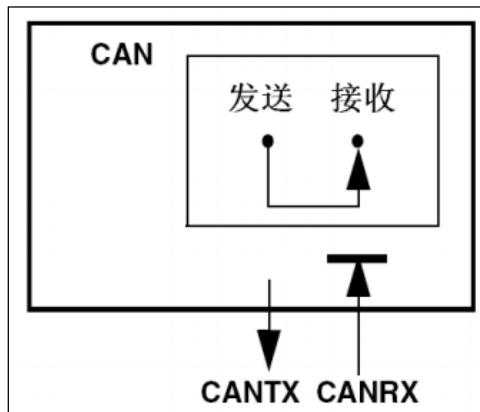
在监听模式下，CAN 只能用于接收数据，不能发送。CAN 在成功接收到一帧数据时也不会回复 ACK。并且会强制进入“Error Passive”模式。这种模式允许软驱动波特率检测，使 CAN 支持 hot plug-in 模式，能够自适应波特率。

图 28-3 监听模式示意图



在自测试模式下，CAN 通过自己接收的模式进行发送和接收，不需要从远程节点获取 ACK。使用自测试模式可以进行自检，而不需要外接任何 CAN 节点。

图 28-4 自测试模式示意图

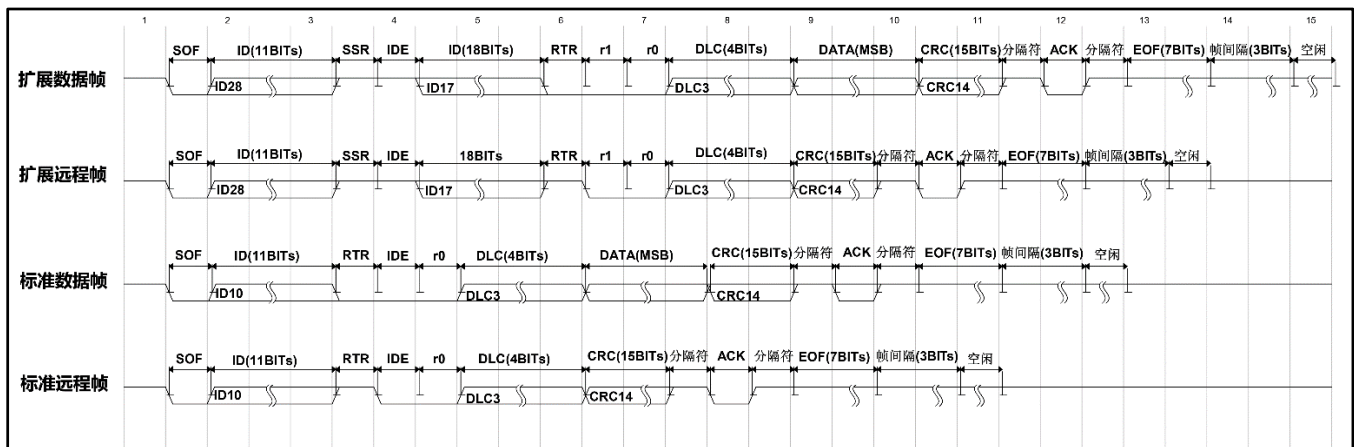


CAN 提供了一种时钟输出模式，但是只能在 Reset Mode 时需选择，TX0 用来输出发送的时钟而不是数据。

### 28.3.3. 发送

根据 CAN 数据帧的结构，如图所示。将需要发送的帧类型、帧长度、帧数据写入到 CAN 的 TxBuff 中，不管数据帧格式是标准帧还是扩展帧，TxBuff 由 13 字节的数据组成，从地址 0x10 到 0x1C，这样可以保证能写入一个数据长度为 8 字节的数据帧。注：在向 TxBuff 写数据之前需要先检查一下 SR.2 状态，确保 TxBuff 被释放了，否则写入的数据就丢失。

图 28-5 CAN 数据帧结构



为了将存放在 TxBuff 里面的数据发送出去，可以通过设置 CMR.0（发送请求）或者设置自接收请求 CMR.4。开始发送时，SR.5 被置为 1 并且发送请求被清除。

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 15bit 的 CRC 校验值，而 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，可以通过写 CMR.1 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。

TxBuff 的数据结构如图所示，由 13 个字节组成，发送时只需按照 TxBuff 结构填入相应的 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。

图 28-6 TxBuff 数据结构

Standard Frame Format (SFF)		Extended Frame Format (EFF)	
CAN Address	Field	CAN Address	Field
10h	TX Frame Information	10h	TX Frame Information
11h	TX Identifier 1	11h	TX Identifier 1
12h	TX Identifier 2	12h	TX Identifier 2
13h	TX Data Byte 1	13h	TX Identifier 3
14h	TX Data Byte 2	14h	TX Identifier 4
15h	TX Data Byte 3	15h	TX Data Byte 1
16h	TX Data Byte 4	16h	TX Data Byte 2
17h	TX Data Byte 5	17h	TX Data Byte 3
18h	TX Data Byte 6	18h	TX Data Byte 4
19h	TX Data Byte 7	19h	TX Data Byte 5
1Ah	TX Data Byte 8	1Ah	TX Data Byte 6
1Bh	(Unused)	1Bh	TX Data Byte 7
1Ch	(Unused)	1Ch	TX Data Byte 8

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	X (1)	X (1)	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	X (2)	X (1)	X (1)	X (1)	X (1)

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	X (1)	X (1)	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13
13h	ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5
14h	ID.4	ID.3	ID.2	ID.1	ID.0	X (2)	X (1)	X (1)

### 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 RxBuff。接收滤波器只会通过那些标识符合条件的报文。

数据放到 RxBuff 时，SR.4 寄存器就会被置位。一旦接收到数据，SR.0 就会被置位，同时才生一个接收中断。RxBuff 的数据结构如图所示，由 13 个字节组成，接收时只需按照 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。

图 28-7 RxBuff 数据结构

Standard Frame Format (SFF)		Extended Frame Format (EFF)	
CAN Address	Field	CAN Address	Field
10h	RX Frame Information	10h	RX Frame Information
11h	RX Identifier 1	11h	RX Identifier 1
12h	RX Identifier 2	12h	RX Identifier 2
13h	RX Data Byte 1	13h	RX Identifier 3
14h	RX Data Byte 2	14h	RX Identifier 4
15h	RX Data Byte 3	15h	RX Data Byte 1
16h	RX Data Byte 4	16h	RX Data Byte 2
17h	RX Data Byte 5	17h	RX Data Byte 3
18h	RX Data Byte 6	18h	RX Data Byte 4
19h	RX Data Byte 7	19h	RX Data Byte 5
1Ah	RX Data Byte 8	1Ah	RX Data Byte 6
1Bh	(Unused)	1Bh	RX Data Byte 7
1Ch	(Unused)	1Ch	RX Data Byte 8

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	RTR	0	0	0	0

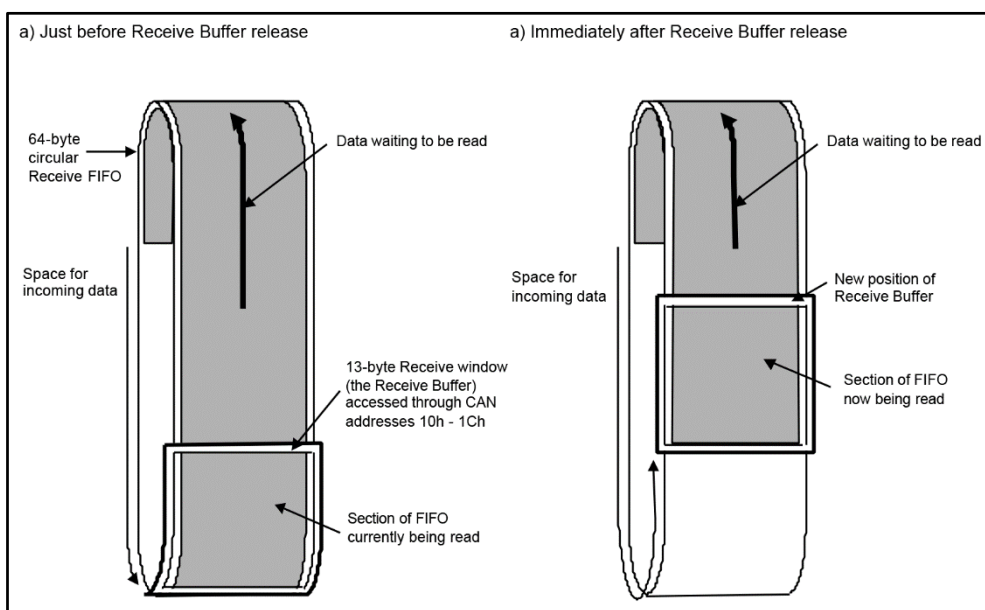
  

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13
13h	ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5
14h	ID.4	ID.3	ID.2	ID.1	ID.0	RTR	0	0

RxFIFO 是 64 字节深度，最多允许存放 5 组 EFF 数据。如果数据没有被读取，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，SR.1 会被置位，接收的数据也会被丢弃，也产生一个中断。

放在 RxBuff 中的数据通过一个 13 位宽的窗口读取，地址是 0x10~0x1C。为了读取所有的报文，CPU 需要移动窗口，这是通过设置 CMR.2 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 28-8 CAN 接收示意图



### 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 CMR.4 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。

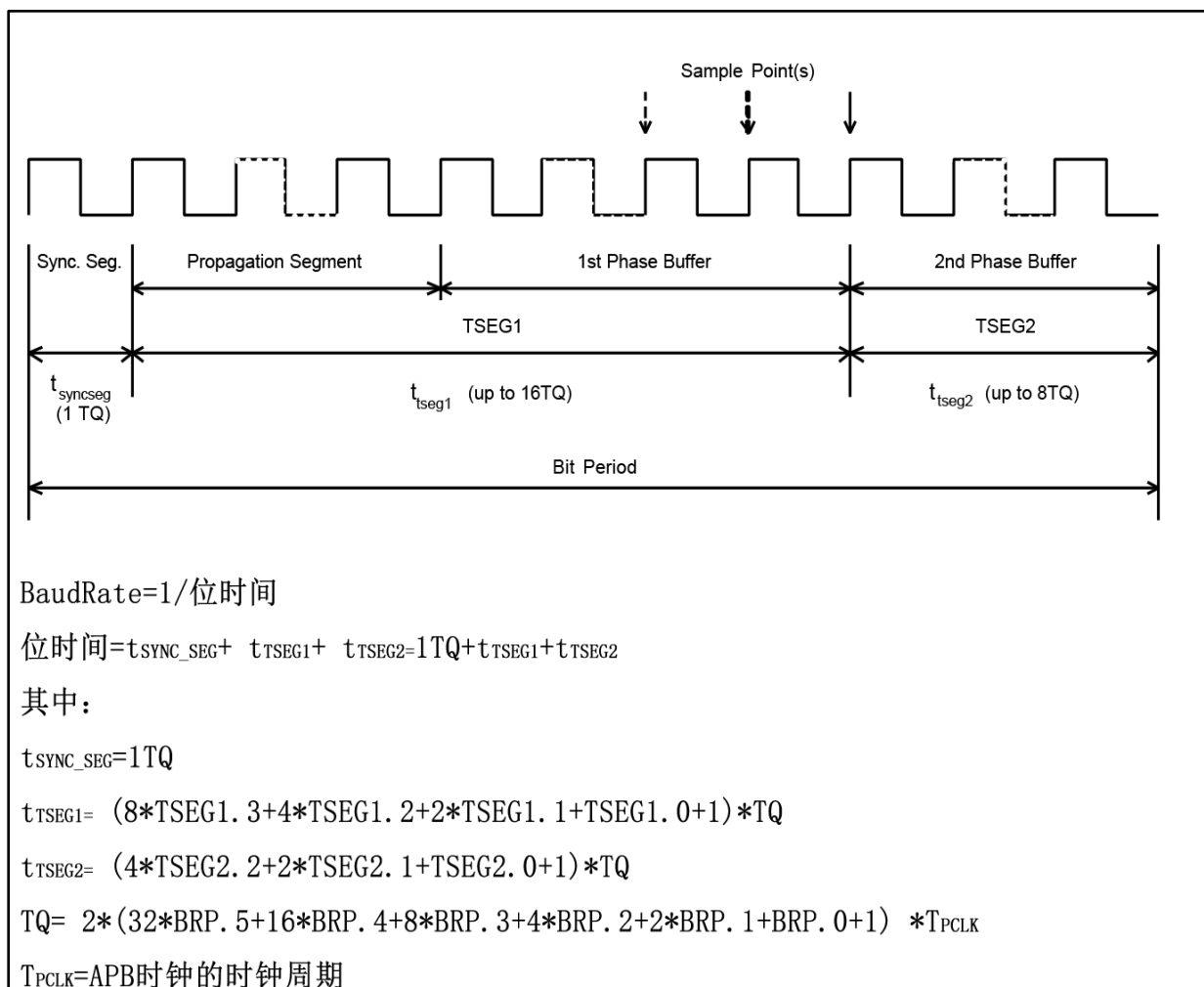
### 28.3.6. 接收过滤

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，通过一个 4 字节的接收过滤器实现。只有报文的标识符与过滤器匹配，才能被写入到 RxBuff。





图 28-11 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 180M, APB 时钟为 90M, 则  $\text{BRP}=8$ ,  $\text{TQ}=2*(8+1)*(1/90\text{M})=18*(1/90\text{M})=1/5\text{M}$ , 设置  $\text{TSEG1}=3$ ,  $\text{TSEG2}=4$ , 则数据位时间= $1+(3+1)+(4+1)=10\text{TQ}$ , 则  $\text{baud}=1/10\text{TQ}=1/(10*1/5\text{M})=1/(2\text{M})=500\text{Kbps}$ 。

### 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 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 Error-Code-Capture-Register 中看到。CAN 也包括一个 EWL 寄存器，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 SR.7 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 模块进入那 Sleep 模式。进入 Sleep 模式的方式是置位 MOD.4。TX0 在 Sleep 模式时为高。

可以通过以下方式唤醒：

1. 将 Sleep 模式设为 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。

## 28.4. 寄存器描述

CAN 寄存器基地址：参见系统说明

偏置	名称	正常模式	复位模式	描述
0x00	CAN_MOD	RW	RW	模式寄存器

0x04	CAN_CMCR	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	RW	RW	中断使能寄存器
0x18	CAN_BTR0	RO	RW	总线时序控制寄存器 0
0x1C	CAN_BTR1	RO	RW	总线时序控制寄存器 1
0x20	CAN_OCR	RO	RW	输出控制寄存器
0x2C	CAN_ALC	RO	RO	仲裁失败抓取寄存器
0x30	CAN_ECC	RO	RO	错误抓取寄存器
0x34	CAN_EWLR	RO	RW	错误警告上限寄存器
0x38	CAN_RXERR	RO	RW	接收错误计数寄存器
0x3C	CAN_TXERR	RO	RW	发送错误计数寄存器
0x40~0x70	CAN_TXBUFF	WO	--	发送缓存寄存器（写操作）
0x40~0x70	CAN_RXBUFF	RO	--	接收缓存寄存器（读操作）
0x40~0x4C	CAN_ACR0 ~ CAN_ACR3	--	RW	接收过滤寄存器
0x50~0x5C	CAN_AMR0~ CAN_AMR3	--	RW	接收屏蔽寄存器
0x74	CAN_RMC	RO	RO	接收报文计数器
0x78	CAN_RBSA	RO	RW	接收缓存起始位置寄存器
0x7C	CAN_CDR	RW	RW	时钟分频因子寄存器
0x80~0x17C	CAN_RX_FIFO	RO	RW	接收 FIFO
0x180~0x1B0	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
31:5	RSV	-	-	保留
4	SM	RW	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)
3	AFM	RW	0	接收过滤器模式 0: 双过滤模式, 两个较短字节的过滤器。 1: 单过滤模式, 单个 4 字节的过滤器。
2	STM	RW	0	自测试模式 0: 正常模式。 1: 自测模式。(不检查 ACK)
1	LOM	RW	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, CAN 接收到正确的数据帧也不会回发 ACK。强制进入 Error-passive 模式。
0	RM	RW	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\_CMCR(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 SR.1。
2	RRB	WO	0	释放接收缓存 0: 无效。

				1: 释放接收缓存。
1	AT	WO	0	终止发送 0: 无效 1: 如果发送请求还未处理, 就取消下一个发送的请求。否则无效。
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, 硬件自动清零, BEI 是脉冲信号, 一般软件很难读到 1, 需要结合 ECCR 寄存器判断出错原因。
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	RW	0	总线错误中断使能 0: 禁止。 1: 使能。
6	ALIE	RW	0	仲裁失败中断使能 0: 禁止。 1: 使能。
5	EPIE	RW	0	Error-Passive 中断使能 0: 禁止 1: 使能
4	WUIE	RW	0	唤醒中断使能 0: 禁止。 1: 使能。
3	DOIE	RW	0	数据过载中断使能 0: 禁止。 1: 使能。
2	EIE	RW	0	错误报警中断使能 0: 禁止。

				1: 使能。
1	TIE	RW	0	发送中断使能。 0: 禁止。 1: 使能。
0	RIE	RW	0	接收中断使能 0: 禁止。 1: 使能。

#### 28.4.6. 时序寄存器 CAN\_BTR0(偏移: 18h)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:6	SJW	RO	RW	0	同步偏移宽度。 定义每个比特在重新同步前可能偏差的最大时钟数。
5:0	BRP	RO	RW	0x0	波特率因子 TQ $TQ = 2 * (BRP + 1) * T_{PCLK}$

#### 28.4.7. 时序寄存器 CAN\_BTR1(偏移: 1Ch)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7	SAM	RO	RW	0	采样选择 0: 采样一次 1: 采样三次
6:4	TSEG2	RO	RW	0x0	采样点尾部 = TSEG2+1
3:0	TSEG1	RO	RW	0x0	采样点前部 = TSEG1+1

## 28.4.8. 输出寄存器 CAN\_OCR(偏移: 20h)

比特	名称	正常模式	复位模式	复位值	描述
31:6	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	000	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

## 28.4.9. 仲裁失败位置获取寄存器 CAN\_ALC(偏移: 2Ch)

比特	名称	正常模式	复位模式	复位值	描述
31:5	RSV	-		-	保留
4:0	ALC	RO	RO	0x0	仲裁失败位置值=ALC + 1

## 28.4.10. 错误代码获取寄存器 CAN\_ECC(偏移: 30h)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:6	ErrCode	RO	RO	00	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	0x0	出错的段代码，见下表
-----	---------	----	----	-----	------------

表 28-1 SegCode 出错段代码表

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

#### 28.4.11. 错误报警寄存器 CAN\_EWLR(偏移: 34h)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:0	EWL	RO	RW	0x60	错误报警值

## 28.4.12. 接收错误寄存器 CAN\_RXERR(偏移: 38h)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:0	RXERR	RO	RW	0x0	接收错误计数寄存器

## 28.4.13. 发送错误寄存器 CAN\_TXERR(偏移: 3Ch)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:0	TXERR	RO	RW	0x0	发送错误计数寄存器

## 28.4.14. 发送缓存写寄存器 CAN\_TXBUFFx(写偏移: 0x40~0x70)

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	--	0x0	发送缓存的字节 x

## 28.4.15. 接收缓存读寄存器 CAN\_RXBUFFx(读偏移: 0x40~0x70)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	-		-	保留
7:0	RXBUFFx	RO	--	0x0	接收缓存的字节 x

## 28.4.16. 接收过滤寄存器 CAN\_ACRx(偏移: 0x40~0x4C)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	--	--	-	保留

7:0	ACRx	--	RW	0x0	接收过滤字节 x
-----	------	----	----	-----	----------

#### 28.4.17. 接收过滤屏蔽寄存器 CAN\_AMRx(偏移: 0x50~0x5C)

比特	名称	正常模式	复位模式	复位值	描述
31:8	RSV	--	--	-	保留
7:0	AMR0	--	RW	0x0	接收过滤字节 0

#### 28.4.18. 接收报文计数寄存器 CAN\_RMC(偏移: 0x74)

比特	名称	正常模式	复位模式	复位值	描述
31:5	RSV	--	--	--	保留
4:0	RMC	RO	RO	0x0	接收 FIFO 中已接收的报文个数

#### 28.4.19. 接收缓存起始地址寄存器 CAN\_RBSA(偏移: 0x78)

比特	名称	正常模式	复位模式	复位值	描述
31:6	RSV	--	--	--	保留
5:0	RBSA	RO	RW	0x0	接收 FIFO 起始地址

#### 28.4.20. 时钟输出分频寄存器 CAN\_CDR(偏移: 0x7C)

比特	名称	正常模式	复位模式	复位值	描述
31:4	RSV	--	--	-	保留
3	Clock OFF	RW	RW	0	Clock Out 关闭 0: 使能 Clock Out 1: 禁止 Clock Out
2:0	CD	RW	RW	000	3'b000: fpclk/2 3'b001: fpclk/4 3'b010: fpclk/6

					3'b011: fpcclk/8 3'b100: fpcclk/10 3'b101: fpcclk/12 3'b110: fpcclk/14 3'b111: fpcclk
--	--	--	--	--	---

#### 28.4.21. 接收 FIFO 访问寄存器 CAN\_RXFIFO(偏移: 0x80~0x17C)

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	RW	0x0	接收 FIFO 字节 x (0~63)

#### 28.4.22. 发送 FIFO 访问寄存器 CAN\_TXFIFO(偏移: 0x180~0x1B0)

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	0x0	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\_TXBUFFx 中；

使能命令寄存器 CAN\_CMR 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\_RXBUFFx 中的数据（包括长度、ID、Data），接收完成；

配置命令寄存器 CAN\_CMR，释放接收缓存，等待下一次接收；

## 29. 通用串行总线 USB

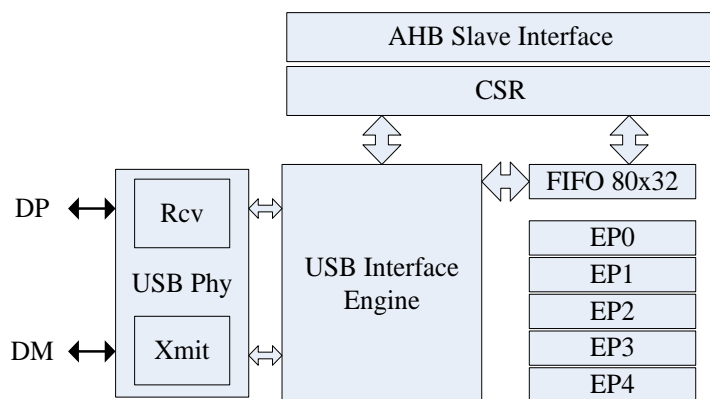
### 29.1. 概述

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。

### 29.2. 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 Bus Reset、Suspend 和 Resume 中断功能；
- 支持远程唤醒功能；
- 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 29-1 USB 结构框架图



## 29.3. 功能描述

### 29.3.1. 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。

### 29.3.2. 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。

### 29.3.3. 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。

#### 29.3.4. 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。

#### 29.3.5. 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,将会将会产生中断。

#### 29.3.6. 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。

#### 29.3.7. 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。

### 29.3.8. 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。

## 29.4. 寄存器描述

USB 寄存器基地址：0x4004\_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

#### 29.4.1. 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	0	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	0	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	0	<p>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。</p> <p>1: 使能此功能； 0: 不使能此功能。</p>
22	USB_DATA_NOEOP_NAK_En	RW	0	<p>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。</p> <p>1: 回复 NAK； 0: 不回复 NAK。</p> <p>此功能只有在 USB_DATA_NOEOP_En 为 1 时有效。</p>
21	USB_TOKEN_NOEOP_NAK_En	RW	0	<p>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。</p> <p>1: 回复 NAK； 0: 不回复 NAK。</p> <p>此功能只有在 USB_TOKEN_NOEOP_En 为 1 时有效。</p>
20	USB_REMOTE_WAKEUP	RW	0	<p>USB 远程唤醒控制位。</p> <p>1: 控制器发送 K 状态； 0: 控制器不发送 K 状态</p> <p>唤醒结束后，此位需要软件写 0 清除。</p>
19: 13	RSV	-	-	保留
12	USB_MORETHAN64_NAK_En	RW	0	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	0	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	0	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	RO	00	USB Dp/Dm 信号状态位。 8bit: DM; 9bit: DP。
7	RSV	-	-	保留
6	USB_DPPU_LO	RW	0	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	0	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	1	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	0	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	0	1: 设置 UsbPhy 为挂起状态; 0: 清除 UsbPhy 的挂起状态;
0	SPEED_MODE	RW	1	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。

#### 29.4.2. EP0 传输控制寄存器 EP0CSR (偏移: 0x4h)

比特	名称	属性	复位值	描述
31: 28	RSV	-	-	保留
27	EP0_SEND_HALT	WO	0	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	RSV	-	-	保留
25	EP0_RECEIVED_NAK	RO	0	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	0	EP0 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 EP0_DATA_START 写 1 可以清除该位。
23:21	RSV	-	-	保留
20	EP0_OUT_VALID	RC_W1	0	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	0	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	0	1: EP0_OUT_TOGGLE_WANT 中的值写入生效。 0: EP0_OUT_TOGGLE_WANT 值写入无效;
17	EP0_OUT_TOGGLE_WANT	RW	0	EP0 Out/Setup 数据包 Toggle 的对比值。 1: Data1; 0: Data0;
16	EP0_OUT_TOGGLE_VALUE	RO	0	EP0 收到的数据包的 Toggle 值。 1: Data1; 0: Data0;
15	EP0_IN_TOGGLE_CTRL_EN	WO	0	1: EP0_IN_TOGGLE_VALUE 中值生效。 0: EP0_IN_TOGGLE_VALUE 值无效;
14	EP0_IN_TOGGLE_VALUE	RW	0	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	RC_W1	0	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	0	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	0	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	0	发送 START。 1: 将 FIFO 中的数据发出; 0: 无操作;
9	EP0_FIFOCLR	WO	0	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	1	EP0 端点使能位。 1: 使能; 0: 不使能;
7:0	EP0_RECEIVED_BYTE	RO	0x00	EP0 接收到的数据 Byte 数目。此位需要小 于最大包长度。

### 29.4.3. EP1 传输控制寄存器 EP1CSR (偏移: 0x8h)

比特	名称	属性	复位值	描述
----	----	----	-----	----

31: 28	RSV	-	-	保留
27	EP1_SEND_HALT	WO	0	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	RSV	-	-	保留
25	EP1_RECEIVED_NAK	RO	0	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	0	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	RC_W1	0	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	0	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	0	1: EP1_OUT_TOGGLE_WANT 中的值生效; 0: EP1_OUT_TOGGLE_WANT 值写入无效。

17	EP1_OUT_TOGGLE_WANT	RW	0	EP1 Out/Setup 数据包 Toggle 的比对值。 1: Data1; 0: Data0;
16	EP1_OUT_TOGGLE_VALUE	RO	1	EP1 收到的数据包的 Toggle 值。 1: Data1; 0: Data0;
15	EP1_IN_TOGGLE_CTRL_EN	WO	0	1: EP1_IN_TOGGLE_VALUE 中写入值生效; 0: EP1_IN_TOGGLE_VALUE 值写入值无效。
14	EP1_IN_TOGGLE_VALUE	RW	0	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	RC_W1	0	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	0	STALL 发送控制位。 1: 发送 STALL; 0: 不发送 STALL; 一次写操作能发送多次 STALL, 直到清除该位。
11	EP1_RECEIVED_DONE	WO	0	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	0	发送 START。 1: 将 FIFO 中的数据发出； 0: 无操作；
9	EP1_FIFOCLR	WO	0	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	1	EP1 端点使能位。 1: 使能； 0: 不使能；
7:0	EP1_RECEIVED_BYTE	RO	0x00	EP1 接收到的数据 Byte 数目。此位需要小于最大包长度。

#### 29.4.4. EP2 传输控制寄存器 EP2CSR (偏移: 0xch)

比特	名称	属性	复位值	描述
31: 28	RSV	-	-	保留
27	EP2_SEND_HALT	WO	0	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	RSV	-	-	保留
25	EP2_RECEIVED_NAK	RO	0	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	0	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	RC_W1	0	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	0	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	0	1: EP2_OUT_TOGGLE_WANT 中的写入值生效。 0: EP2_OUT_TOGGLE_WANT 的写入值无效;
17	EP2_OUT_TOGGLE_WANT	RW	0	EP2 Out/Setup 数据包 Toggle 的对比值。 1: Data1; 0: Data0。
16	EP2_OUT_TOGGLE_VALUE	RO	1	EP2 收到的数据包的 Toggle 值。 1: Data1; 0: Data0。
15	EP2_IN_TOGGLE_CTRL_EN	WO	0	1: EP2_IN_TOGGLE_VALUE 中的写入值生效; 0: EP2_IN_TOGGLE_VALUE 的写入值无效。
14	EP2_IN_TOGGLE_VALUE	RW	0	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	RC_W1	0	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	0	STALL 发送控制位。 1: 发送 STALL; 0: 不发送 STALL。 一次写操作能发送多次 STALL，直到清除该位。
11	EP2_RECEIVED_DONE	WO	0	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	0	发送 START。 1: 将 FIFO 中的数据发出; 0: 无操作;
9	EP2_FIFOCLR	WO	0	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	1	EP2 端点使能位。 1: 使能; 0: 不使能;
7:0	EP2_RECEIVED_BYTE	RO	0x00	EP2 接收到的数据 Byte 数目。此位需要小于最大包长度。

## 29.4.5. EP3 传输控制寄存器 EP3CSR (偏移: 0x10h)

比特	名称	属性	复位值	描述
31: 28	RSV	-	-	保留
27	EP3_SEND_HALT	WO	0	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	RSV	-	-	保留
25	EP3_RECEIVED_NAK	RO	0	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	0	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	RC_W1	0	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	0	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	0	1: EP3_OUT_TOGGLE_WANT 中的写入

				值生效。 0: EP3_OUT_TOGGLE_WANT 值写入值无效。
17	EP3_OUT_TOGGLE_WANT	RW	0	EP3 Out/Setup 数据包 Toggle 的比对值。 1: Data1; 0: Data0;
16	EP3_OUT_TOGGLE_VALUE	RO	1	EP3 收到的数据包的 Toggle 值。 1: Data1; 0: Data0;
15	EP3_IN_TOGGLE_CTRL_EN	WO	0	1: EP3_IN_TOGGLE_VALUE 中的写入值生效; 0: EP3_IN_TOGGLE_VALUE 中的写入值无效。
14	EP3_IN_TOGGLE_VALUE	RW	0	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	RC_W1	0	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	0	STALL 发送控制位。 1: 发送 STALL; 0: 不发送 STALL; 一次写操作能发送多次 STALL, 直到清除该位。
11	EP3_RECEIVED_DONE	WO	0	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	0	发送 START。 1: 将 FIFO 中的数据发出; 0: 无操作;
9	EP3_FIFOCLR	WO	0	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	1	EP3 端点使能位。 1: 使能; 0: 不使能;
7:0	EP3_RECEIVED_BYTE	RO	0x00	EP3 接收到的数据 Byte 数目。此位需要小于最大包长度。

#### 29.4.6. EP4 传输控制寄存器 EP4CSR (偏移: 0x14h)

比特	名称	属性	复位值	描述
31: 28	RSV	-	-	保留
27	EP4_SEND_HALT	WO	0	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	RSV	-	-	保留
25	EP4_RECEIVED_NAK	RO	0	EP4 接收到 Host 的 NAK 握手包标志信号。 1: 接收到握手包;

				<p>0: 未接收到握手包;</p> <p>此位为只读位, USB Reset 或者向 EP4_DATA_START 写 1 可以清除该位。</p>
24	EP4_RECEIVED_ACK	RO	0	<p>EP4 接收到 Host 的 ACK 握手包标志信号。</p> <p>1: 接收到握手包;</p> <p>0: 未接收到握手包;</p> <p>此位为只读位, USB Reset 或者向 EP4_DATA_START 写 1 可以清除该位。</p>
23:21	-	-	-	-
20	EP4_OUT_VALID	RC_W1	0	<p>Ep4 Out 有效标志, 当有效数据进入 FIFO 时, 此位会置 1。写 1 清零。</p>
19	EP4_OUT_TOGGLE_STATE	RO	0	<p>EP4 Out/Setup 状态错误标志。</p> <p>1: Toggle 收到的与预期不符;</p> <p>0: Toggle 正常;</p>
18	EP4_OUT_TOGGLE_CTRL_En	WO	0	<p>1: EP4_OUT_TOGGLE_WANT 中的写入值生效;</p> <p>0: EP4_OUT_TOGGLE_WANT 中的写入值无效。</p>
17	EP4_OUT_TOGGLE_WANT	RW	0	<p>EP4 Out/Setup 数据包 Toggle 的比对值。</p> <p>1: Data1;</p> <p>0: Data0;</p>
16	EP4_OUT_TOGGLE_VALUE	RO	1	<p>EP4 收到的数据包的 Toggle 值。</p> <p>1: Data1;</p> <p>0: Data0;</p>
15	EP4_IN_TOGGLE_CTRL_EN	WO	0	<p>EP4_IN_TOGGLE_VALUE 中值生效使能。</p> <p>1: EP4_IN_TOGGLE_VALUE 值写入生效;</p>

14	EP4_IN_TOGGLE_VALUE	RW	0	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	RC_W1	0	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	0	STALL 发送控制位。 1: 发送 STALL; 0: 不发送 STALL; 一次写操作发送多次 STALL，直到清除该位。
11	EP4_RECEIVED_DONE	WO	0	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	0	发送 START。 1: 将 FIFO 中的数据发出; 0: 无操作;
9	EP4_FIFOCLR	WO	0	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	1	EP4 端点使能位。 1: 使能;

				0: 不使能;
7:0	EP4_RECEIVED_BYTE	RO	0x00	EP4 接收到的数据 Byte 数目。此位需要小于最大包长度。

#### 29.4.7. 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	0x00	USB 地址寄存器。

#### 29.4.8. 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	0x00000000	SETUP Data 包 Byte0~Byte3 寄存器。

#### 29.4.9. 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	0x00000000	SETUP Data 包 Byte4~Byte7 寄存器。

#### 29.4.10. 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。

## 29.4.11. 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	0000	当前接收的 USB 包的 PID 值。

## 29.4.12. 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。

## 29.4.13. 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	0x00	CRC 错误包的个数, 在 USB Reset 时复位。

## 29.4.14. 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	0x1FF	Reset/Resume/Suspend 检测阈值设定。设置时间为 USB_STATUS_DETECT_CNT*5.3us + 2.5us

## 29.4.15. EP0 发送数据数目寄存器 EPOSENDNB (偏移: 0x40h)

比特	名称	属性	复位值	描述
31: 8	RSV	-	-	保留
7:0	EPO_SEND_BYTE	RW	0x00	EP0 发送数据 Byte 数量寄存器。

## 29.4.16. 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	0x00	EP1 发送数据 Byte 数量寄存器。

## 29.4.17. 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	0x00	EP2 发送数据 Byte 数量寄存器。

## 29.4.18. 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	0x00	EP3 发送数据 Byte 数量寄存器。

## 29.4.19. 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	0x00	EP4 发送数据 Byte 数量寄存器。

## 29.4.20. 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	0x00000000	EP0 FIFO 入口地址, 只支持 32bit 访问。

## 29.4.21. 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	0x00000000	EP1 FIFO 入口地址, 只支持 32bit 访问。

## 29.4.22. 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	0x00000000	EP2 FIFO 入口地址, 只支持 32bit 访问。

## 29.4.23. 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	0x00000000	EP3 FIFO 入口地址, 只支持 32bit 访问。

## 29.4.24. 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	0x00000000	EP4 FIFO 入口地址, 只支持 32bit 访问。

## 29.4.25. 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	0	IN/OUT/Setup 操作发生 Toggle 错误时, 此位置 1。
30	NOEOP_ERR_RAW	RO	0	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	0	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	0	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	0	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	0	EP1 IN 操作，主机未成功返回 ACK 信号时，此位会置 1。
25	EP0_IN_HANDSHAKE_ERR_RAW	RO	0	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	0	收到的 DATA 数据包长度超过 64byte，此位会置 1。
23	CRC_ERR_RAW	RO	0	收到令牌包或者数据包的 CRC Error，此位会置 1。
22	SETADDR_RAW	RO	0	当 Host 设置 USB 设备地址完成，此位会置 1。
21	TURNAROUND_ERROR_RAW	RO	0	Host 回复 Ack 包发生 TimeOut 中断。
20	EP4_ACK_RAW	RO	0	EP4 Ack 状态，发送或者接收 Ack 包，此位会置 1。
19	EP4_OUT_RAW	RO	0	EP4 Out 中断，当有效数据进入 FIFO 时，此位会置 1。
18	EP4_IN_RAW	RO	0	EP4 接收到 IN 令牌包时，此位会置 1。
17	EP3_ACK_RAW	RO	0	EP3 Ack 状态，发送或者接收 Ack 包，此位会置 1。
16	EP3_OUT_RAW	RO	0	EP3 Out 中断，当有效数据进入 FIFO 时，此位会置 1。
15	EP3_IN_RAW	RO	0	EP3 接收到 IN 令牌包时，此位会置 1。
14	EP2_ACK_RAW	RO	0	EP2 Ack 状态，发送或者接收 Ack 包，此位会置 1。
13	EP2_OUT_RAW	RO	0	EP2 Out 中断，当有效数据进入 FIFO 时，此位会置 1。
12	EP2_IN_RAW	RO	0	EP2 接收到 IN 令牌包时，此位会置 1。

11	EP1_ACK_RAW	RO	0	EP1 Ack 状态，发送或者接收 Ack 包，此位会置 1。
10	EP1_OUT_RAW	RO	0	EP1 Out 中断，当有效数据进入 FIFO 时，此位会置 1。
9	EP1_IN_RAW	RO	0	EP1 接收到 IN 令牌包时，此位会置 1。
8	EP0_ACK_RAW	RO	0	EP0 Ack 状态，发送或者接收 Ack 包，此位会置 1。
7	EP0_OUT_RAW	RO	0	EP0 Out 中断，当有效数据进入 FIFO 时，此位会置 1。
6	EP0_IN_RAW	RO	0	EP0 接收到 IN 令牌包时，此位会置 1。
5	SUDAV_RAW	RO	0	接收到 Setup 数据包，此位会置 1。
4	SETUPTOK_RAW	RO	0	接收到 Setup 令牌包，此位会置 1。
3	SOF_RAW	RO	0	接收到 Sof 包，此位会置 1。
2	RESUME_RAW	RO	0	Host Resume，此位会置 1。
1	SUSPEND_RAW	RO	0	Host Suspend，此位会置 1。
0	BUS_RESET_RAW	RO	0	Host Reset，此位会置 1。

#### 29.4.26. 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	0	TOGGLE_STATE_ERR 中断使能。
30	NOEOP_ERR_EN	RW	0	NOEOP_ERR 中断使能。
29	EP4_IN_HANDSHAKE_ERR_EN	RW	0	EP4_IN_HANDSHAKE_ERR 中断使能。
28	EP3_IN_HANDSHAKE_ERR_EN	RW	0	EP3_IN_HANDSHAKE_ERR 中断使能。
27	EP2_IN_HANDSHAKE_ERR_EN	RW	0	EP2_IN_HANDSHAKE_ERR 中断使能。
26	EP1_IN_HANDSHAKE_ERR_EN	RW	0	EP1_IN_HANDSHAKE_ERR 中断使能。
25	EP0_IN_HANDSHAKE_ERR_EN	RW	0	EP0_IN_HANDSHAKE_ERR 中断使能。
24	DATA_BYTE_MORETHAN_64_EN	RW	0	收到的 DATA 数据包长度超过 64byte 中断使能。

23	CRC_ERR_EN	RW	0	CRC 错误中断使能。
22	SETADDR_EN	RW	0	当 Host 设置 USB 设备地址完成中断使能。
21	TURNAROUND_ERROR_EN	RW	0	Host 回复 Ack 包发生 TimeOut 中断使能。
20	EP4_ACK_EN	RW	0	EP4 Ack 状态，发送或者接收 Ack 包中断使能。
19	EP4_OUT_EN	RW	0	Ep4 Out 中断，当有效数据进入 FIFO 时产生中断使能。
18	EP4_IN_EN	RW	0	EP4 接收到 IN 令牌包中断使能。
17	EP3_ACK_EN	RW	0	EP3 Ack 状态，发送或者接收 Ack 包中断使能。
16	EP3_OUT_EN	RW	0	EP3 Out 中断，当有效数据进入 FIFO 时产生中断使能。
15	EP3_IN_EN	RW	0	EP3 接收到 IN 令牌包中断使能。
14	EP2_ACK_EN	RW	0	EP2 Ack 状态，发送或者接收 Ack 包中断使能。
13	EP2_OUT_EN	RW	0	EP2 Out 中断，当有效数据进入 FIFO 时产生中断使能。
12	EP2_IN_EN	RW	0	EP2 接收到 IN 令牌包中断使能。
11	EP1_ACK_EN	RW	0	EP1 Ack 状态，发送或者接收 Ack 包中断使能。
10	EP1_OUT_EN	RW	0	EP1 Out 中断，当有效数据进入 FIFO 时产生中断使能。
9	EP1_IN_EN	RW	0	EP1 接收到 IN 令牌包中断使能。
8	EP0_ACK_EN	RW	0	EP0 Ack 状态，发送或者接收 Ack 包中断使能。
7	EP0_OUT_EN	RW	0	EP0 Out 中断，当有效数据进入 FIFO 时产生中断使能。
6	EP0_IN_EN	RW	0	EP0 接收到 IN 令牌包中断使能。

5	SUDAV_EN	RW	0	接收到 Setup 数据包中断使能。
4	SETUPTOK_EN	RW	0	接收到 Setup 令牌包中断使能。
3	SOF_EN	RW	0	接收到 Sof 包中断使能。
2	RESUME_EN	RW	0	Host Resume 中断使能。。
1	SUSPEND_EN	RW	0	Host Suspend 中断使能。
0	BUS_RESET_EN	RW	0	Host Reset 中断使能。

#### 29.4.27. 中断清除寄存器 INT\_CLR (偏移: 0xFFF0h)

比特	名称	属性	复位值	描述
31	TOGGLE_STATE_ERR_CLR	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	EP2_IN_HANDSHAKE_ERR_RAW 中断寄存器清除位。 1: EP2_IN_HANDSHAKE_ERR_RAW 清 0;

				0: EP2_IN_HANDSHAKE_ERR_RAW 保持不变。
26	EP1_IN_HANDSHAKE_ERR_CLR	WO	0	EP1_IN_HANDSHAKE_ERR_RAW 中断寄存器清除位。 1: EP1_IN_HANDSHAKE_ERR_RAW 清 0; 0: EP1_IN_HANDSHAKE_ERR_RAW 保持不变。
25	EP0_IN_HANDSHAKE_ERR_CLR	WO	0	EP0_IN_HANDSHAKE_ERR_RAW 中断寄存器清除位。 1: EP0_IN_HANDSHAKE_ERR_RAW 清 0; 0: EP0_IN_HANDSHAKE_ERR_RAW 保持不变。
24	DATA_BYTE_MORETHAN_64_CLR	WO	0	DATA_BYTE_MORETHAN_64_RAW 中断寄存器清除位。 1: DATA_BYTE_MORETHAN_64_RAW 清 0; 0: DATA_BYTE_MORETHAN_64_RAW 保持不变。
23	CRC_ERR_CLR	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	0	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	WO	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	WO	0	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	WO	0	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	WO	0	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	WO	0	Host Reset 中断寄存器清除位。 1: BUS_RESET_RAW 清 0; 0: BUS_RESET_RAW 保持不变。
---	---------------	----	---	---

## 29.5. 软件流程

### 29.5.1. USB 连接

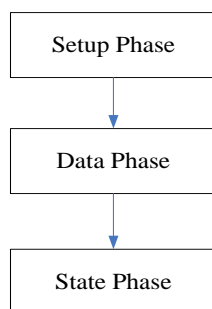
默认状态 USB 是不连接的，需要在完成初始化之后，将 WORKING\_MODE 中的寄存器 USB\_DPPU 和 USB\_DPPU\_LO 置位，反之可以断开 USB 连接。

### 29.5.2. 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 29-2 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

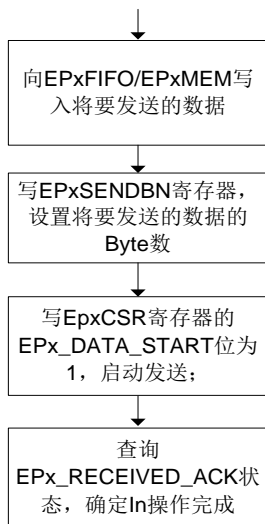
NAK packet;

- **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。

### 29.5.3. 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：

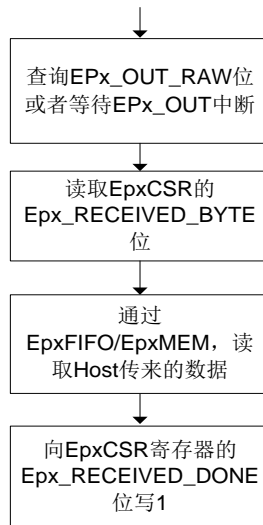
图 29-3 Endpoint In 传输



### 29.5.4. 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 29-4 Endpoint Out 传输



## 30. CRC 计算单元

### 30.1. 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。

### 30.2. 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;

### 30.3. 寄存器描述

CRC 寄存器基地址: 0x40010c00

#### 30.3.1. 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	0x00000000	<p>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</p> <p>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</p>

#### 30.3.2. 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

### 30.3.3. 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	0x00000000	写入 CRC 初始值

### 30.3.4. 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	0x00000000	写入结果异或值
------	--------	----	------------	---------

### 30.3.5. 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]

### 30.3.6. 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	0x00000000	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。

## 30.4. 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。

## 31. 算法库

### 31.1. 数据类型

算法库中的数据类型定义如下：

```
typedef unsigned char UINT8;
```

```
typedef unsigned int UINT32;
```

即 UINT8 表示 8 位无符号数，UINT32 表示 32 位无符号数。

### 31.2. HRNG

#### 31.2.1. 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；

#### 31.2.2. 库文件说明

驱动开发包源文件说明如下：

表格 31-1 库文件说明

hrng.lib	实现取随机数操作
hrng.h	hrng.lib 对应的头文件

#### 31.2.3. 函数说明

表格 31-2 HRNG 驱动函数说明

函数名	功能描述	参数	
hrng_initial	初始化函数用于启动随机数模块	输入：	无
		输出：	无
		返回值：	无
hrng_source_disable	关闭随机数模块	输入：	无
		输出：	无
		返回值：	无
get_hrng	取多个字节随机数	输入：	UINT32 byte_len: 取随机数的字节长度

		输出:	UINT8 *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

### 31.2.4. 注意事项

无。

## 31.3. AES

### 31.3.1. 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

### 31.3.2. 库文件说明

表格 31-3 库文件说明

aes.lib	实现 AES 加解密运算
aes.h	aes.lib 对应的头文件

### 31.3.3. 函数说明

表格 31-4 AES 驱动函数说明

函数名	功能描述	参数	
aes_set_key	配置 aes 密钥及对调模式	输入:	UINT32 *keyin: 存放密钥的起始地址
			UINT8 key_len: 密钥长度(128/192/256)
			UINT8 swap_en: 输入输出数据大小端对调使能 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
aes_crypt	aes 加解密函数	输入:	UINT32 *indata: 存放输入数据的起始地址
			UINT32 block_len: 加解密的块(128 bit)个数

			<p>UINT8 operation:加解密操作选择</p> <p>0: 解密</p> <p>1: 加密</p>
			<p>UINT8 mode: 加解密模式选择</p> <p>0: ECB 模式</p> <p>1: CBC 模式</p>
			<p>UINT32 *iv: 存放 CBC 模式初始向量的起始地址</p>
			<p>UINT32 security_mode: 安全模式选择</p> <p>0x12345678: 普通模式</p> <p>0: 安全模式</p>
		输出:	<p>UINT32 *outdata: 存放输出结果的起始地址</p>
		返回值:	<p>UINT32 加解密运行结果</p> <p>0: 运行失败</p> <p>0xa59ada68: 运行成功</p>
aes_set_key_u8	配置 aes 密钥及对调模式 (字节方式)	输入:	<p>UINT8 *keyin: 存放密钥的起始地址</p>
			<p>UINT8 key_len: 密钥长度(128/192/256)</p>
			<p>UINT8 swap_en: 输入输出数据大小端对调使能</p> <p>0: 输入输出数据大小端对调禁止</p> <p>1: 输入输出数据大小端对调使能</p>
		输出:	<p>无</p>
		返回值:	<p>无</p>
aes_crypt_u8	aes 加解密函数(字节方式)	输入:	<p>UINT8 *indata: 存放输入数据的起始地址</p>
			<p>UINT32 block_len: 加解密的块(128 bit)个数</p>
			<p>UINT8 operation:加解密操作选择</p> <p>0: 解密</p> <p>1: 加密</p>

		UINT8 mode: 加解密模式选择 0: ECB 模式 1: CBC 模式
		UINT8 *iv: 存放 CBC 模式初始向量的起始地址
		UINT32 security_mode: 安全模式选择 0x12345678: 普通模式 0: 安全模式
	输出:	UINT8 *outdata: 存放输出结果的起始地址
	返回值:	UINT32 加解密运行结果 0: 运行失败 0xa59ada68: 运行成功

### 31.3.4. 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 位字数据存储于数组的低位元素中，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 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 plain_text [4]= {0x44332211, 0x88776655, 0xccbbaa99, 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 plain_text[16] = {0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xaa,0xbb,0xcc,0x dd,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 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。

## 31.4. HASH

### 31.4.1. 主要特性

- 支持 SHA1/SHA256 算法；

### 31.4.2. 库文件说明

表格 31-5 库文件说明

sha1.lib	实现 sha1 运算
sha1.h	sha1.lib 对应的头文件
sha256.lib	实现 sha256 运算
sha256.h	sha256.lib 对应的头文件

### 31.4.3. 函数说明

表格 31-6 驱动函数说明

SHA1_hash	SHA1 运算	输入:	UINT8 *pDataIn: 待压缩数据的起始地址
			UINT32 DataLen: 待压缩数据的字节长度
		输出:	UINT8 *pDigest: 摘要值的起始地址
		返回值:	无
SHA256_hash	SHA256 运算	输入:	UINT8 *pDataIn: 待压缩数据的起始地址
			UINT32 DataLen: 待压缩数据的字节长度
		输出:	UINT8 *pDigest: 摘要值的起始地址
		返回值:	无

### 31.4.4. 注意事项:

- 1) 通过时钟分频寄存器配置好系统时钟和算法时钟；
- 2) 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;

## 31.5. FAU

### 31.5.1. 主要特性

- 24 位 CORDIC 旋转引擎
- 支持 sin/cos/atan2 等函数

### 31.5.2. 库文件说明

表格 31-7 库文件说明

fau.lib	实现 sin/cos/atan2/sqrt 运算
fau.h	fau.lib 对应的头文件

### 31.5.3. 函数说明

表格 31-8 驱动函数说明

函数名	功能描述	参数	
cordic_cos_sin_x x 为计算精度，范围(1~8)	cos_sin 运算	输入:	int angle_para: 以弧度为单位的角度值(除以 $\pi$ , Q31 格式)
		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
cordic_atan_sqrt_x x 为计算精度，范围(1~8) atan 与 sqrt 值在精度 1 时	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)

误差非常大，不建议使用		输出：	int* sqrt_data: xy 代表向量的长度(Q31 格式)
			int* atan_data: xy 代表向量的角度(Q31 格式)
		返回值：	无

#### 31.5.4. 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： $0.5 * 2^{31} = 1073741824 = 0x40000000$