

ARM Cortex-M0 内核微控制器

NCCM0660

用户手册



目录

目录	2
1. 概述	13
1.1. 说明	13
1.2. 产品特点	13
2. 特性	15
3. 料号信息	16
4. 管脚配置	17
5. 管脚描述	20
6. 系统及存储器架构	22
6.1. 系统架构	22
6.2. 存储器映射	22
6.3. 嵌入式 SRAM 和 FLASH 存储器	23
6.4. 引导配置	23
6.5. 中断向量管理	24
6.6. 系统滴答定时器 (SYSTICK)	25
6.6.1. 功能说明	25
6.6.2. SysTick 寄存器	26
7. 嵌入式 Flash(FLASH)	29
7.1. 简介	29
7.2. 主要特性	29
7.3. 功能说明	29
7.3.1. 闪存结构	29
7.3.2. 页擦除	30
7.3.3. 块擦除操作	30
7.3.4. 主存储块编程	30
7.3.5. FLASH 读操作	31
7.3.6. 用户配置区及保护配置	31
7.4. FLASH 中断	31
7.5. FLASH 寄存器	32
7.5.1. FLASH 寄存器映射	32
7.5.2. FLASH IP 版本号(IP_VERSION)	32
7.5.3. FLASH 地址寄存器(ADDR)	33
7.5.4. FLASH 数据寄存器 (DATA)	33
7.5.5. FLASH 控制寄存器(CTRL)	33
7.5.6. FLASH 等待寄存器(WAIT)	34
7.5.7. FLASH 配置重载入寄存器(CFG_RELOAD)	34
7.5.8. FLASH 中断使能寄存器(IEN)	35
7.5.9. FLASH 中断状态寄存器(ISR)	35
7.5.10. FLASH 配置信息区 0 寄存器(CFG_INFO0)	36
7.5.11. FLASH 配置信息区 1 寄存器(CFG_INFO1)	36

7.5.12. FLASH 读保护区 0 寄存器(RDPROTO)	37
7.5.13. FLASH 写保护区 0 寄存器(WRPROTO)	38
8. 电源控制(PWR)	40
8.1. 简介	40
8.2. 功能说明	40
8.2.1. 低功耗模式	40
8.2.2. 欠压复位	41
8.3. PWR 中断	41
8.4. PWR 寄存器	42
8.4.1. PWR 寄存器映射	42
8.4.2. PWR 控制寄存器(CR)	42
8.4.3. BOR 控制寄存器(BORCR)	43
9. 复位和时钟控制(RCC)	44
9.1. 复位控制	44
9.1.1. 电源复位	44
9.1.2. 系统复位	44
9.2. 时钟控制	44
9.3. 功能说明	44
9.3.1. 时钟树	45
9.3.2. 时钟输出功能	45
9.3.3. LSE 监测功能	46
9.4. RCC 寄存器	47
9.4.1. RCC 寄存器映射	47
9.4.2. 时钟控制寄存器 (CR)	48
9.4.3. 时钟配置寄存器 (CFGR)	48
9.4.4. 时钟中断使能寄存器 (CIER)	50
9.4.5. 时钟中断标记寄存器 (CIFR)	50
9.4.6. 时钟中断清除寄存器 (CICR)	51
9.4.7. AHB 周边复位寄存器 (AHBRSTR)	51
9.4.8. APB 周边复位寄存器 1 (APBRSTR1)	52
9.4.9. AHB 周边使能寄存器 (AHBENR)	53
9.4.10. APB 周边使能寄存器 1 (APBENR1)	54
9.4.11. AHB 周边 SLEEP 模式时钟使能寄存器 (AHBSMENR)	55
9.4.12. APB 周边 SLEEP 模式时钟使能寄存器 1 (APBSMENR1)	55
9.4.13. 外设时钟选择寄存器 (KSEL)	56
9.4.14. 复位状态寄存器 (RSTID)	57
9.4.15. LSI 控制寄存器 (LSICR)	58
9.4.16. LSE 控制寄存器 (LSECR)	58
9.4.17. 时钟监测控制寄存器 (SCMCR)	59
9.4.18. 时钟异常标志寄存器 (SCMFLAG)	59
9.4.19. APB 周边复位寄存器 2 (APBRSTR2)	60
9.4.20. APB 周边使能寄存器 2 (APBENR2)	61
9.4.21. APB 周边 SLEEP 模式时钟使能寄存器 2 (APBSMENR2)	62
10. 通用输入输出接口(GPIO)	63

10.1. 简介	63
10.2. 主要特性	63
10.3. 功能说明	63
10.3.1. GPIO 引脚配置	63
10.3.2. 引脚复用功能	64
10.3.3. 输入配置	66
10.3.4. 外部中断线/唤醒线	66
10.3.5. 输出配置	66
10.3.6. 模拟配置	67
10.3.7. 复用功能配置	68
10.3.8. GPIO 锁定功能	69
10.4. GPIO 寄存器	70
10.4.1. GPIO 寄存器映射	70
10.4.2. GPIOx 端口模式寄存器 (MODER) (x = A 到 C)	70
10.4.3. GPIOx 端口上拉/下拉寄存器 (PUPDR) (x = A 到 C)	71
10.4.4. GPIOx 端口输入数据寄存器 (IDR) (x = A 到 C)	71
10.4.5. GPIOx 端口输出数据寄存器 (ODR) (x = A 到 C)	71
10.4.6. GPIOx 端口位置位/复位寄存器 (BSRR) (x = A 到 C)	72
10.4.7. GPIOx 端口配置锁定寄存器 (LCKR) (x = A 到 C)	72
10.4.8. GPIOx 复用功能低位寄存器 (AFRL) (x = A 到 C)	72
10.4.9. GPIOx 复用功能高位寄存器 (AFRH) (x = A 到 C)	73
10.4.10. GPIOx 端口位复位寄存器 (BRR) (x = A 到 C)	73
11. 系统配置控制器 (SYSCFG)	74
11.1. 简介	74
11.2. SYSCFG 寄存器	74
11.2.1. SYSCFG 寄存器映射	74
11.2.2. SYSCFG 配置寄存器 1 (CFGR1)	74
11.2.3. SYSCFG 配置寄存器 2 (CFGR2)	75
11.2.4. NMI 中断标志寄存器 (NMI_SR)	76
12. 扩展中断与事件控制器(EXTI)	77
12.1. 简介	77
12.2. 主要特性	77
12.3. 功能说明	78
12.3.1. EXTI 框图	78
12.3.2. 外部中断与事件功能	78
12.4. EXTI 寄存器	79
12.4.1. EXTI 寄存器映射	79
12.4.2. EXTI 上升沿触发选择寄存器 (RTSR)	79
12.4.3. EXTI 下降沿触发选择寄存器 (FTSR)	80
12.4.4. EXTI 软件中断事件触发寄存器 (SWIER)	80
12.4.5. EXTI 上升沿挂起寄存器 (RPR)	80
12.4.6. EXTI 下降沿挂起寄存器 (FPR)	81
12.4.7. EXTI 外部中断选择寄存器(EXTICRx) x=1~4	81
12.4.8. EXTI 中断控制寄存器 (IMR)	82

12.4.9. EXTI 事件控制寄存器 (EMR)	82
13. 基本定时器(TIM6/TIM7)	83
13.1. 简介	83
13.2. 主要特性	83
13.3. BTIM 功能说明	84
13.3.1. 时基单元	84
13.3.2. 计数模式	84
13.3.3. 软件触发	86
13.3.4. BTIM 中断	86
13.4. BTIM 寄存器	87
13.4.1. BTIM 寄存器映射	87
13.4.2. BTIM IP 版本寄存器(VER)	87
13.4.3. BTIM 控制寄存器 1 (CR1)	88
13.4.4. BTIM 控制寄存器 2 (CR2)	89
13.4.5. BTIM 中断使能寄存器 (IER)	89
13.4.6. BTIM 状态寄存器 (SR)	90
13.4.7. BTIM 时间生成寄存器 (EGR)	90
13.4.8. BTIM 计数器 (CNT)	91
13.4.9. BTIM 预分频器 (PSC)	91
13.4.10. BTIM 自动重装载寄存器 (ARR)	92
14. 通用定时器(TIM2/TIM10/TIM16)	93
14.1. 简介	93
14.2. 主要特性	93
14.3. 功能说明	94
14.3.1. 模式及功能选项说明	94
14.3.2. 时基单元	94
14.3.3. 计数器模式	95
14.3.4. TIMx 时钟源	97
14.3.5. 输出比较/输入捕获	97
14.3.6. PWM 模式	98
14.3.7. 互补输出和死区插入	98
14.3.8. 触发单脉冲模式 1	100
14.3.9. 触发单脉冲模式 2	100
14.3.10. 中断及异常说明	101
14.4. GTIM 寄存器	102
14.4.1. GTIM 寄存器映射	102
14.4.2. GTIM 版本寄存器(VER)	103
14.4.3. GTIM 控制寄存器 (CR1)	103
14.4.4. GTIM 控制寄存器 2 (CR2)	104
14.4.5. GTIM 从模式控制寄存器 (SMCR)	105
14.4.6. GTIM 中断使能寄存器 (IER)	106
14.4.7. GTIM 状态寄存器 1 (SR1)	107
14.4.8. GTIM 状态寄存器 2 (SR2)	109
14.4.9. GTIM 事件生成寄存器 (EGR)	109

14.4.10. GTIM 计数器 (CNT)	111
14.4.11. GTIM 预分频器 (PSC)	111
14.4.12. GTIM 自动重载寄存器 (ARR)	112
14.4.13. GTIM 捕获比较模式寄存器[复用] (CCMRn)	112
14.4.14. GTIM 捕获比较使能寄存器 (CCERn)	114
14.4.15. GTIM 捕获比较寄存器 (CCRn)	115
14.4.16. TIMx 死区寄存器 (DTRn)	116
15. 模数转换器(ADC)	118
15.1. 简介	118
15.2. 主要特性	118
15.3. 功能说明	119
15.3.1. ADC 框图	119
15.3.2. ADC 时钟	119
15.3.3. 触发模式	119
15.3.4. 单次转换(CONT=0)	120
15.3.5. 连续转换(CONT=1)	120
15.3.6. ADC 开始转换	121
15.3.7. 比较输出	121
15.4. ADC 中断	121
15.5. ADC 寄存器	122
15.5.1. ADC 寄存器映射	122
15.5.2. ADC 版本寄存器(VER)	122
15.5.3. ADC 控制寄存器(CTL)	122
15.5.4. ADC 数据寄存器(DATA)	124
15.5.5. ADC 中断使能寄存器(IER)	124
15.5.6. ADC 中断和状态寄存器(ISR)	125
15.5.7. ADC 中断和状态清零寄存器(ICR)	125
15.5.8. ADC 配置寄存器(CFG)	126
15.5.9. ADC 高阈值参考数据寄存器(REFH)	126
15.5.10. ADC 低阈值参考数据寄存器(REFL)	127
16. 循环冗余校验计算单元 (CRC)	128
16.1. 简介	128
16.2. 主要特性	128
16.3. 功能说明	128
16.3.1. CRC 框图	128
16.3.2. CRC 操作	129
16.4. CRC 寄存器	130
16.4.1. CRC 寄存器总表	130
16.4.2. CRC 版本寄存器(IP_VERSION)	130
16.4.3. CRC 数据寄存器 (DR)	131
16.4.4. CRC 控制寄存器 (CR)	131
16.4.5. CRC 初始值 (INIT)	132
17. 独立看门狗(IWDG)	133
17.1. 简介	133

17.2. 主要特性	133
17.3. 功能说明	133
17.4. IWDG 寄存器	135
17.4.1. IWDG 寄存器映射	135
17.4.2. IWDG 版本寄存器(VER)	135
17.4.3. IWDG 控制寄存器(CR)	135
17.4.4. IWDG 键值寄存器(KR)	136
18. 窗口看门狗(WWDG)	137
18.1. 简介	137
18.2. 主要特性	137
18.3. 功能说明	137
18.3.1. WWDG 框图	138
18.3.2. 使能看门狗	138
18.3.3. 控制递减计数器	138
18.3.4. 看门狗中断	138
18.3.5. 调试模式	139
18.4. WWDG 中断	139
18.5. WWDG 寄存器	139
18.5.1. WWDG 寄存器映射	139
18.5.2. WWDG 版本寄存器(VER)	139
18.5.3. WWDG 控制寄存器 (CR)	140
18.5.4. WWDG 配置寄存器 (CFR)	140
18.5.5. WWDG 状态寄存器 (SR)	141
18.5.6. WWDG 写保护寄存器(WPR)	141
19. 内部集成电路 (I2C) 接口	142
19.1. 简介	142
19.2. 主要特性	142
19.3. I2C 功能说明	142
19.3.1. I2C 框图	143
19.3.2. 模式选择	143
19.3.3. 通信流程	144
19.3.4. I2C 初始化	144
19.3.5. 软件复位	147
19.3.6. 数据传输	147
19.3.7. I2C 从模式	149
19.3.8. I2C 主模式	155
19.3.9. I2C_TIMINGR 寄存器配置示例	164
19.3.10. SMBUS 特性	166
19.3.11. SMBUS 初始化	168
19.3.12. SMBUS 通信	169
19.3.13. 错误条件	176
19.3.14. 中断及异常说明	178
19.4. I2C 寄存器	180
19.4.1. I2C 寄存器映射	180

19.4.2. I2C 版本寄存器(VER)	180
19.4.3. I2C 控制寄存器 1 (CR1)	181
19.4.4. I2C 控制寄存器 2 (CR2)	183
19.4.5. I2C 自身地址 1 寄存器 (OA1)	185
19.4.6. I2C 自身地址 2 寄存器 (OA2)	185
19.4.7. I2C 时序寄存器 (TIMINGR)	186
19.4.8. I2C 超时寄存器 (TIMEOUTR)	187
19.4.9. I2C 中断和状态寄存器 (ISR)	188
19.4.10. I2C 中断清零寄存器 (ICR)	190
19.4.11. I2C PEC 寄存器 (PECR)	191
19.4.12. I2C 接收数据寄存器 (RXDR)	191
19.4.13. I2C 发送数据寄存器 (TXDR)	192
20. 串行外设接口 (SPI)	193
20.1. 简介	193
20.2. 主要特性	193
20.3. 功能说明	194
20.3.1. SPI 时序与数据帧格式	194
20.3.2. SPI 配置流程	195
20.3.3. 多主机模式	196
20.3.4. 多从机模式	196
20.3.5. NSS 功能	197
20.3.6. TI 模式	197
20.3.7. 匹配唤醒	197
20.4. SPI 标志与中断	198
20.5. SPI 寄存器	199
20.5.1. SPI 寄存器映射	199
20.5.2. SPI IP 版本号 (IP_VERSION)	199
20.5.3. SPI 控制寄存器 1 (CR1)	200
20.5.4. SPI 控制寄存器 2 (CR2)	201
20.5.5. SPI 中断使能寄存器 (IER)	202
20.5.6. SPI 使能寄存器 (ENR)	203
20.5.7. SPI 状态寄存器 (SR)	204
20.5.8. SPI 主时钟分频寄存器 (DIVR)	205
20.5.9. SPI FIFO 控制寄存器 (FCR)	205
20.5.10. SPI 复位寄存器 (RSTR)	206
20.5.11. SPI 数据匹配寄存器 (DMR)	206
20.5.12. SPI 发送数据寄存器 (TDR)	207
20.5.13. SPI 接收数据寄存器 (RDR)	207
21. 通用同步异步收发器(USART)	208
21.1. 简介	208
21.2. 主要特性	208
21.3. 功能说明	209
21.3.1. USART 框图	209
21.3.2. 发送与接收字符	209

21.3.3. 过采样	210
21.3.4. 波特率	213
21.3.5. 多处理器通信	213
21.3.6. MODBUS 通信	215
21.3.7. 同步模式	216
21.3.8. 单线半双工模式	216
21.3.9. LIN 模式	217
21.3.10. IrDA SIR 模式	217
21.4. USART 中断	219
21.5. USART 寄存器	220
21.5.1. USART 寄存器映射	220
21.5.2. USART 控制寄存器 (CTL)	220
21.5.3. USART 配置寄存器 (CFG)	223
21.5.4. USART 波特率寄存器 (BR)	224
21.5.5. USART 超时寄存器 (RTO)	224
21.5.6. USART 请求寄存器 (RQ)	225
21.5.7. USART 中断和状态使能寄存器 (IE)	225
21.5.8. USART 中断和状态寄存器 (IS)	227
21.5.9. USART 中断和状态清零寄存器 (IC)	230
21.5.10. USART 接收数据寄存器 (RD)	231
21.5.11. USART 发送数据寄存器 (TD)	231
22. 低功耗通用异步接收器 (LPUART)	233
22.1. 简介	233
22.2. 主要特性	233
22.3. LPUART 功能说明	234
22.3.1. LPUART 框图	234
22.3.2. LPUART 信号	234
22.3.3. LPUART 字符说明	234
22.3.4. LPUART 发送器	235
22.3.5. LPUART 接收器	237
22.3.6. 奇偶校验	239
22.3.7. 波特率生成	239
22.3.8. 自动波特率	241
22.3.9. 单线半双工	241
22.3.10. 多处理器模式	242
22.3.11. 空闲线路检测	242
22.3.12. 待机/停止模式唤醒	244
22.3.13. 中断及异常说明	244
22.4. LPUART 寄存器	246
22.4.1. LPUART 寄存器映射	246
22.4.2. LPUART 版本寄存器(VER)	246
22.4.3. LPUART 控制寄存器 1 (CR1)	246
22.4.4. LPUART 控制寄存器 2 (CR2)	249
22.4.5. LPUART 控制寄存器 3 (CR3)	250

22.4.6. LPUART 波特率寄存器 (BRR)	251
22.4.7. LPUART 请求寄存器 (RQR)	251
22.4.8. LPUART 状态寄存器 (SR)	251
22.4.9. LPUART 清零寄存器 (CLR)	254
22.4.10. LPUART 接收数据寄存器 (RDR)	254
22.4.11. LPUART 发送数据寄存器 (TDR)	255
22.4.12. LPUART 超时寄存器 (RTOR)	255
23. 唤醒定时器和蜂鸣器产生电路(AWU)	256
23.1. 简介	256
23.2. 主要特性	256
23.3. 唤醒定时器功能说明	256
23.4. 蜂鸣器产生电路功能说明	257
23.5. AWU 寄存器	258
23.5.1. AWU 寄存器映射	258
23.5.2. BUZ 控制寄存器 (BUZCON)	258
23.5.3. AWU 控制寄存器 (AWUCON)	258
24. 触摸(TOUCH)	260
24.1. 简介	260
24.2. 主要特性	261
24.3. 功能说明	261
24.3.1. 单个按键连续检测模式	261
24.3.2. 多个按键连续检测模式	262
24.3.3. 低功耗唤醒	262
24.3.4. 触发模式	262
24.3.5. 中断说明	262
24.4. TOUCH 寄存器	263
24.4.1. TOUCH 寄存器映射	263
24.4.2. TOUCH 版本寄存器(VER)	263
24.4.3. TOUCH 控制寄存器 (CR)	263
24.4.4. TOUCH 控制寄存器 (CFGR1)	264
24.4.5. TOUCH 控制寄存器 (CFGR2)	265
24.4.6. TOUCH 控制寄存器 (CHSEL)	266
24.4.7. TOUCH 控制寄存器 (MATCR)	267
24.4.8. TOUCH 控制寄存器 (ISR)	267
24.4.9. TOUCH 控制寄存器 (ICR)	269
24.4.10. TOUCH 控制寄存器 (MATxTH)(x=0~15)	270
25. 液晶显示 (LCD)	271
25.1. 简介	271
25.2. 主要特性	271
25.3. 功能说明	271
25.3.1. LCD 框图	271
25.3.2. 工作模式	271
25.3.3. 电阻型 LCD 驱动器	272
25.3.4. 电容型 LCD 驱动器	272

25.3.5. LED 驱动器	274
25.3.6. 闪烁	274
25.3.7. LCD 波形	274
25.3.8. LCD 数据	278
25.4. LCD 寄存器	279
25.4.1. LCD 寄存器映射	279
25.4.2. LCD 版本寄存器(VER)	279
25.4.3. LCD 控制寄存器 (CR)	279
25.4.4. LCD 闪烁控制寄存器 (FCR)	281
25.4.5. LCD COM 选择控制寄存器 (CENR)	281
25.4.6. LCD SEGEN 选择控制寄存器 (SENR)	282
25.4.7. LCD SEG 数据寄存器 (SDRx) (x=0-31)	282
26. 调试(DBG)	283
26.1. 简介	283
26.2. 功能说明	283
26.2.1. 对低功耗模式的调试支持	283
26.2.2. 对定时器、看门狗的调试支持	283
26.3. DBG 寄存器	284
26.3.1. DBG 寄存器映射	284
26.3.2. DBG 版本寄存器(VER)	284
26.3.3. DBG 控制寄存器 (CR)	284
26.3.4. DBG 冻结寄存器 1 (FZ1)	285
26.3.5. DBG 冻结寄存器 2 (FZ2)	286
27. 电气特性	287
27.1. 极限参数	287
27.2. DC 特性	287
27.3. BOD/BOR 特性	288
27.4. ADC 特性	288
27.5. 内部振荡器特性	289
27.6. 内部低频振荡器特性	289
27.7. 外部振荡器特性	289
27.8. 存储器工作特性	289
27.9. 外部复位及中断管脚特性	290
27.10. ESD 特性	290
27.11. SYSTEM ESD 特性	290
27.12. EFT 特性	290
27.13. EMI 特性	291
27.14. Latch up 特性	291
28. 封装信息	292
28.1. LQFP48	292
28.2. LQFP32	293
28.3. LQFP44	293
29. 文档约定	294
29.1. 寄存器相关缩写词列表	294

29.2. 词汇表	294
30. 版本历史	295



1. 概述

1.1. 说明

NCCM0660是一款内嵌64K字节FLASH的32位单片机。主要有如下特点：具有64K字节代码存储空间，8K字节SRAM；具有全套外设，内部集成48M高频时钟及32K低频时钟，外部支持32.768KHZ晶振，支持BOD/BOR功能；高速12位ADC，先进且灵活的定时器，通讯接口（例如USART，SPI，I2C），内置低功耗电容触控模块，LCD/LED驱动模块，非常适合通用工业控制电子产品应用。

1.2. 产品特点

1. 4路16位带互补输出和死区插入的通用定时器模块；
2. 12位ADC；
3. 具有用于遥控的红外接口；
4. 内置16通道触控；
5. LCD/LED 驱动模块。

NCCM0660 Cortex-M0

Memory

64KB Flash

8KB RAM

System

IHRC 48M

ILRC 32K

XTAL 32.768K

BOR/BOD

Timer

16bit Btimer x2

16bit Gtimer x3

AWU x1

Connectivity

USART x1

LPUART x1

I2C x1

SPI x1

Safety

IWDG x1

WWDG x1

CRC x1

Analog

12bit x 30ch ADC

TOUCH x1

Other

Buzzer x1

LCD x1

2. 特性

CPU 特性

Cortex-M0, 最高主频 48MHz
最高 48MHz 系统频率

ROM

64KB FLASH

RAM

8KB SRAM

GPIO

最多支持 46 个 GPIO
支持任意端口变化中断

定时器

3 个 16bit 通用定时器, 支持捕获和 PWM
2 个 16bit 基本定时器
2 个看门狗定时器
1 个唤醒定时器

通信

支持 1 个 USART
支持 1 个 LPUART
支持 1 个 I2C
支持 1 个 SPI

模拟模块

支持上电/低压复位
支持 28 路 12 位 ADC 外部输入, 2 路内部输入
内置低功耗电容触控模块, 最大支持 16 个通道
内置液晶驱动模块, 最大支持 4X32 段
支持 LED 驱动模块, 最大支持 8X28
支持内置 48MHz 振荡器, 出厂校准至 1%

支持内部 32KHz 低频振荡器
支持外部 32.768KHz 晶振

工作模式

正常工作模式
睡眠模式(SLEEP)
停机模式(STOP)

工作条件

工作频率: 最大 48MHz
工作电压: 2.0~5.5V
工作温度: -40~85 摄氏度

封装

LQFP48-7x7
LQFP44-10x10
LQFP32-7x7

3. 料号信息

芯片型号	封装	引脚数量
NCCM0660-44FT	LQFP44 10x10	44
NCCM0660-32LT	LQFP32 7x7	32
NCCM0660-48LT	LQFP48 7x7	48



4. 管脚配置

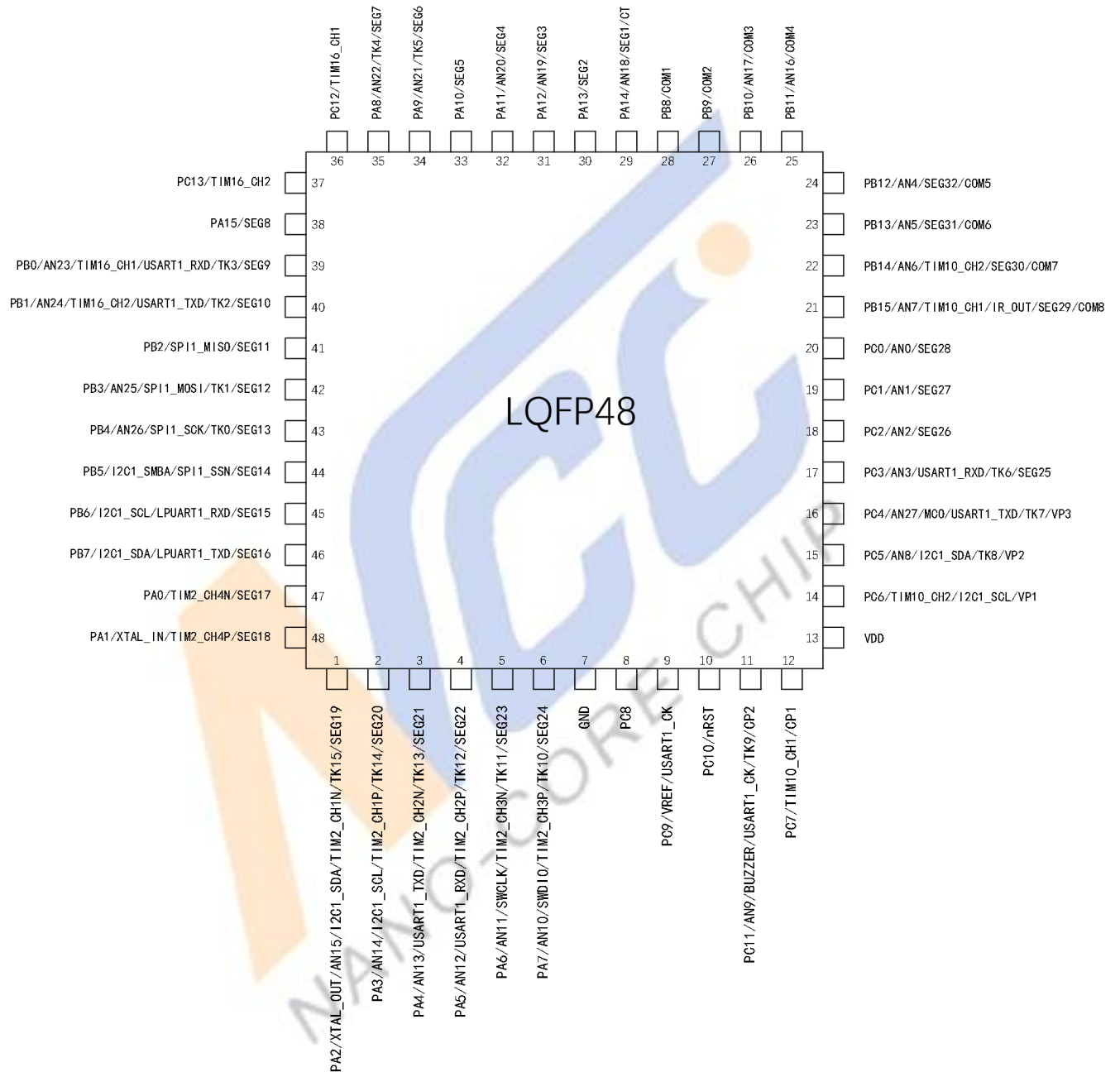


图 4-1 LQFP48 引脚图

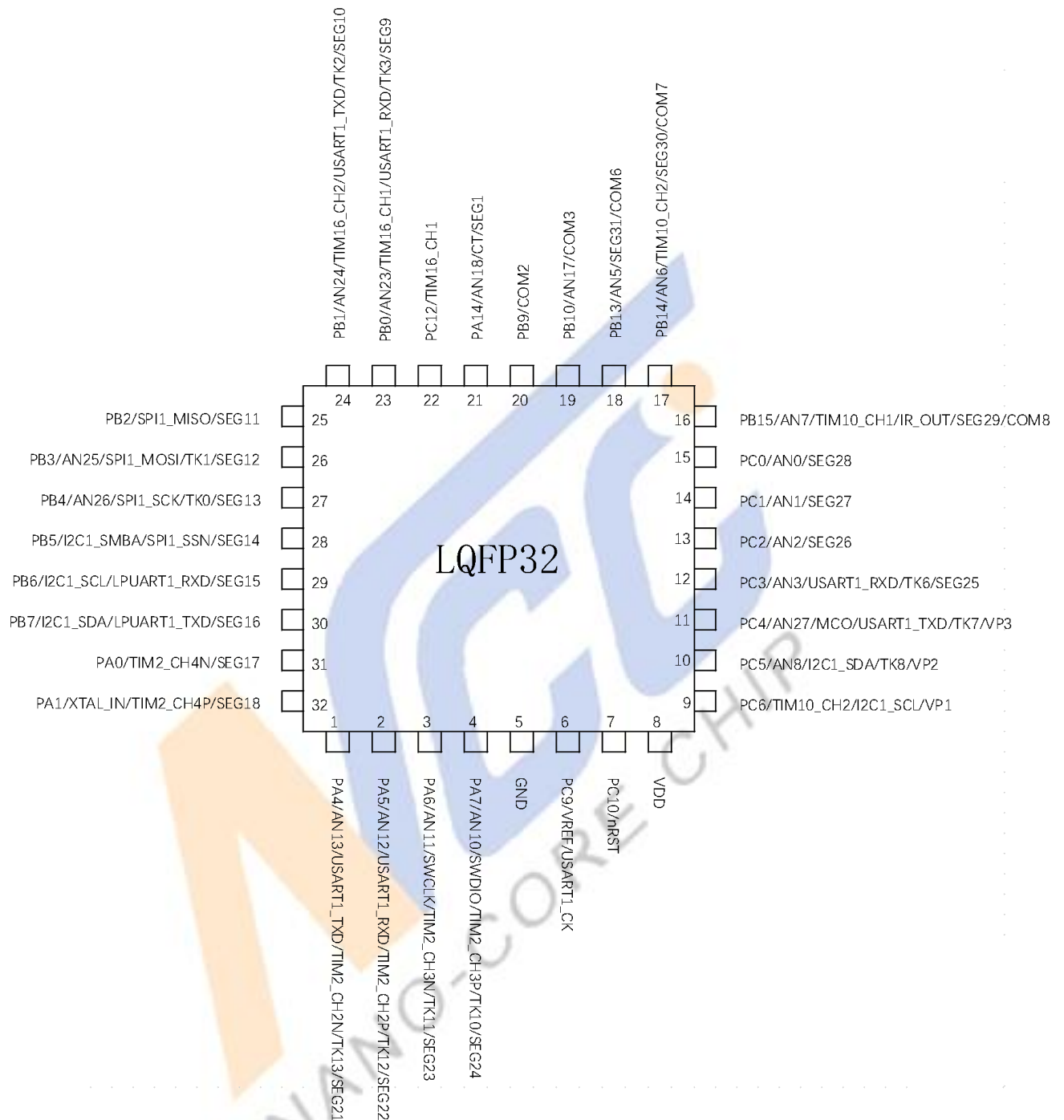


图 4-2 LQFP32 引脚图

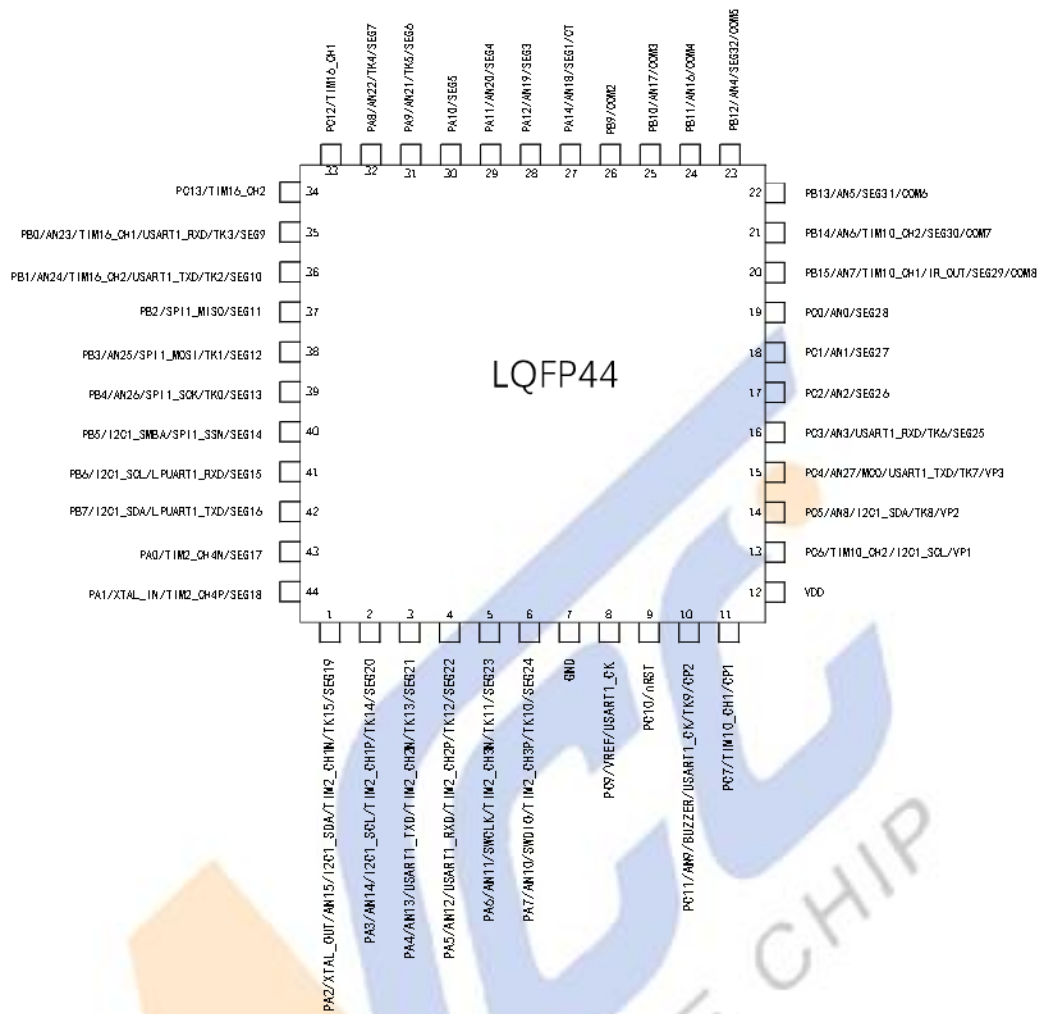


图 4-3 LQFP44 引脚图

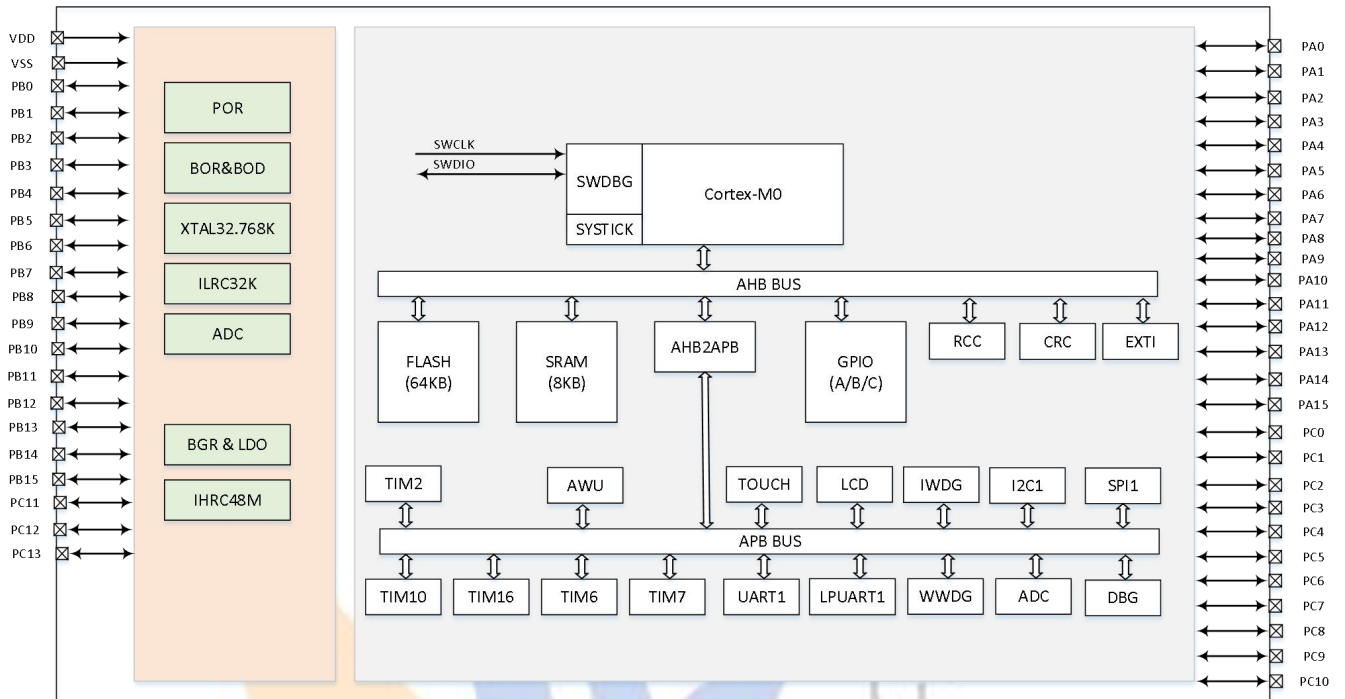
5. 管脚描述

LQFP48	LQFP32	LQFP44	名称	类型	附加功能	复用功能
1		1	PA2	I/O	XTAL_OUT/AN15/SEG19/TK15	I2C1_SDA/TIM2_CH1N/LED_S19
2		2	PA3	I/O	AN14/SEG20/TK14	I2C1_SCL/TIM2_CH1P/LED_S20
3	1	3	PA4	I/O	AN13/SEG21/TK13	USART1_TXD/TIM2_CH2N/LED_S21
4	2	4	PA5	I/O	AN12/SEG22/TK12	USART1_RXD/TIM2_CH2P/LED_S22
5	3	5	PA6	I/O	AN11/SEG23/TK11	SWCLK/TIM2_CH3N/LED_S23
6	4	6	PA7	I/O	AN10/SEG24/TK10	SWDIO/TIM2_CH3P/LED_S24
7	5	7	VSSIO	GND		
8			PC8	I/O		
9	6	8	PC9	I/O	VREF	USART1_CK
10	7	9	PC10	I/O	nRST	
11		10	PC11	I/O	AN9/CP2/TK9	BUZZER/ USART1_CK
12		11	PC7	I/O	CP1	TIM10_CH1
13	8	12	VDDIO	PWR		
14	9	13	PC6	I/O	VP1	TIM10_CH2/I2C1_SCL
15	10	14	PC5	I/O	AN8/VP2/TK8	I2C1_SDA
16	11	15	PC4	I/O	AN27/VP3/TK7	MCO/USART1_TXD
17	12	16	PC3	I/O	AN3/SEG25/TK6	USART1_RXD/ LED_S25
18	13	17	PC2	I/O	AN2/SEG26	LED_S26
19	14	18	PC1	I/O	AN1/SEG27	LED_S27
20	15	19	PC0	I/O	AN0/SEG28	LED_S28
21	16	20	PB15	I/O	AN7/SEG29/COM8	TIM10_CH1/IR_OUT/LED_C8
22	17	21	PB14	I/O	AN6/SEG30/COM7	TIM10_CH2/ LED_C7
23	18	22	PB13	I/O	AN5/SEG31/COM6	LED_C6

LQFP48	LQFP32	LQFP44	名称	类型	附加功能	复用功能
24		23	PB12	I/O	AN4/SEG32/COM5	LED_C5
25		24	PB11	I/O	AN16/COM4	LED_C4
26	19	25	PB10	I/O	AN17/COM3	LED_C3
27	20	26	PB9	I/O	COM2	LED_C2
28			PB8	I/O	COM1	LED_C1
29	21	27	PA14	I/O	AN18/SEG1/CT	LED_S1
30			PA13	I/O	SEG2	LED_S2
31		28	PA12	I/O	AN19/SEG3	LED_S3
32		29	PA11	I/O	AN20/SEG4	LED_S4
33		30	PA10	I/O	SEG5	LED_S5
34		31	PA9	I/O	AN21/SEG6/TK5	LED_S6
35		32	PA8	I/O	AN22/SEG7/TK4	LED_S7
36	22	33	PC12	I/O		TIM16_CH1
37		34	PC13	I/O		TIM16_CH2
38			PA15	I/O	SEG8	LED_S8
39	23	35	PB0	I/O	AN23/SEG9/TK3	TIM16_CH1/USART1_RXD/ LED_S9
40	24	36	PB1	I/O	AN24/SEG10/TK2	TIM16_CH2/USART1_TXD/ LED_S10
41	25	37	PB2	I/O	SEG11	SPI1_MISO/ LED_S11
42	26	38	PB3	I/O	AN25/SEG12/TK1	SPI1_MOSI/ LED_S12
43	27	39	PB4	I/O	AN26/SEG13/TK0	SPI1_SCK/ LED_S13
44	28	40	PB5	I/O	SEG14	I2C1_SMBA/SPI1_SSN/ LED_S14
45	29	41	PB6	I/O	SEG15	I2C1_SCL/LPUART1_RXD/ LED_S15
46	30	42	PB7	I/O	SEG16	I2C1_SDA/LPUART1_TXD/ LED_S16
47	31	43	PA0	I/O	SEG17	TIM2_CH4N/ LED_S17
48	32	44	PA1	I/O	XTAL_IN/SEG18	TIM2_CH4P/ LED_S18

6. 系统及存储器架构

6.1. 系统架构



6.2. 存储器映射

表 6-1 存储器映射表显示了器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。

表 6-1 存储器映射表

类型	地址范围	大小	存储区
AHB	0x4800 0800 - 0x4800 0BFF	1KB	GPIOC
	0x4800 0400 - 0x4800 07FF	1KB	GPIOB
	0x4800 0000 - 0x4800 03FF	1KB	GPIOA
	0x4002 3000 - 0x4002 33FF	1KB	CRC
	0x4002 1800 - 0x4002 1BFF	1KB	EXTI
	0x4002 1000 - 0x4002 13FF	1KB	RCC
	0x4002 0C00 - 0x4002 0FFF	1KB	FLASH
	0x4001 7000~0x4001 73FF	1KB	TK
	0x4001 6000~0x4001 63FF	1KB	LCD
	0x4001 5800 - 0x4001 5BFF	1KB	DBG

类型	地址范围	大小	存储区
	0x4001 4400 - 0x4001 47FF	1KB	TIM16
	0x4001 4000 - 0x4001 43FF	1KB	TIM10
	0x4001 3800 - 0x4001 3BFF	1KB	USART1
	0x4001 3000 - 0x4001 33FF	1KB	SPI1
	0x4001 2400 - 0x4001 27FF	1KB	ADC
	0x4001 0000 - 0x4001 03FF	1KB	SYSCFG
	0x4000 7000 - 0x4000 73FF	1KB	PWR
	0x4000 5400 - 0x4000 57FF	1KB	I2C1
	0x4000 5000 - 0x4000 53FF	1KB	LPUART1
	0x4000 3000 - 0x4000 33FF	1KB	IWDG
	0x4000 2C00 - 0x4000 2FFF	1KB	WWDG
	0x4000 1C00 - 0x4000 1FFF	1KB	AWU
	0x4000 1400 - 0x4000 17FF	1KB	TIM7
	0x4000 1000 - 0x4000 13FF	1KB	TIM6
	0x4000 0400 - 0x4000 07FF	1KB	TIM2
SRAM	0x2000 0000 - 0x2000 1FFF	8KB	SRAM
代码	0x1FFF_0000-0x1FFF_03FF	1KB	Option bytes
	0x0800 0000 -0x0800 FFFF	64KB	主 Flash
	0x0000 0000 - 0x0000 FFFF	64KB	主 Flash 或 SRAM (取决于 BOOT 配置)

6.3. 嵌入式 SRAM 和 FLASH 存储器

微控制器最大包含一个 64KB 的可编程 FLASH 和 8KB 的 SRAM。

6.4. 引导配置

系统复位后，CPU 将从 0x0000 0000 地址获取栈顶值，然后从始于 0x0000 0004 的自举存储器开始执行代码。

用户自定义 Bootloader 时，需将 NVIC 中断向量表复制到 SRAM 首地址，然后通过 SYSCFG 配置寄存器 CFGR1[MEM_MODE]位修改使用的存储器。

6.5. 中断向量管理

位置	优先级	优先级类型	缩略语	说明	地址
-	-	-	-	保留	0x0000_0000
-	-3	固定	复位	复位	0x0000_0004
-	-2	固定	NMI_Handler	不可屏蔽中断，LSE CSS 与 NMI 向量相链接	0x0000_0008
-	-	-	-	保留	0x0000_0010 0x0000_0014 0x0000_0018 0x0000_001C 0x0000_0020 0x0000_0024 0x0000_0028
-	3	可设置	SVC_Handler	通过 SWI 指令调用的系统服务	0x0000_002C
-	-	-	-	保留	0x0000_0030 0x0000_0034
-	5	可设置	PendSV_Handler	可挂起的系统服务请求	0x0000_0038
-	6	可设置	SysTick_Handler	系统滴答定时器	0x0000_003C
0	7	可设置	WWDG	窗口看门狗中断	0x0000_0040
1	8	可设置	PWR	PWR 中断(与 EXTI 16 结合使用)	0x0000_0044
2	9	-	-	-	0x0000_0048
3	10	可设置	FLASH	Flash 全局中断	0x0000_004C
4	11	可设置	RCC	RCC 全局中断	0x0000_0050
5	12	可设置	EXTI0_1	EXTI 线 0 和 1 中断	0x0000_0054
6	13	可设置	EXTI2_3	EXTI 线 2 和 3 中断	0x0000_0058
7	14	可设置	EXTI4_15	EXTI 线 4 和 15 中断	0x0000_005C
8	15	-	-	-	0x0000_0060
9	16	-	-	-	0x0000_0064
10	17	-	-	-	0x0000_0068
11	18	-	-	-	0x0000_006C
12	19	可设置	ADC	ADC 全局中断	0x0000_0070
13	20	可设置	TOUCH	TOUCH 全局中断(与 EXTI 19	0x0000_0074

位置	优先级	优先级类型	缩略语	说明	地址
				结合使用)	
14	21	可设置	AWU	AWU 全局中断(与 EXTI 20 结合使用)	0x0000_0078
15	22	-	-	-	0x0000_007C
16	23	可设置	TIM2	TIM2 全局中断	0x0000_0080
17	24	可设置	TIM6	TIM6 全局中断	0x0000_0084
18	25	可设置	TIM7	TIM7 全局中断	0x0000_0088
19	26	-	-	-	0x0000_008C
20	27	可设置	TIM10	TIM10 全局中断	0x0000_0090
21	28	可设置	TIM16	TIM16 全局中断	0x0000_0094
22	29	可设置	LPUART1	LPUART1 全局中断(与 EXTI 17 结合使用)	0x0000_0098
23	30	可设置	I2C1	I2C1 全局中断	0x0000_009C
24	31	-	-	-	0x0000_00A0
25	32	可设置	SPI1	SPI1 全局中断(与 EXTI 18 结合使用)	0x0000_00A4
26	33	-	-	-	0x0000_00A8
27	34	可设置	USART1	USART1 全局中断	0x0000_00AC
28	35	-	-	-	0x0000_00B0
29	36	-	-	-	0x0000_00B4
30	37	-	-	-	0x0000_00B8
31	38	-	-	-	0x0000_00BC

6.6. 系统滴答定时器（SYSTICK）

6.6.1. 功能说明

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减至 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常。该异常事件会引起 SysTick 异常处理的执行。

SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询计数标志。

6.6.2. SysTick 寄存器

6.6.2.1.SysTick 寄存器映射

基地址为：0xE000E010

表 6-2 FLASH 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	SysTick 控制和状态寄存器 (CTRL)	rw	0x0000_0000
0x04	SysTick 重装载值寄存器 (LOAD)	rw	0x00FF_FFFF
0x08	SysTick 当前值寄存器 (VAL)	rw	0x00FF_FFFF
0x0C	SysTick 校准值寄存器 (CALIB)	r	0x0000_1770

6.6.2.2. SysTick 控制和状态寄存器 (CTRL)

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

位域	功能
31-17	保留
16 COUNTFLAG	当 SysTick 定时器计数到 0 时，该位变为 1，读取寄存器会被清零
15-3	保留
2 CLKSOURCE	值为 1 表示 SysTick 定时器使用内核时钟，否则会使用参考时钟频率
1 TICKINT	SysTick 中断使能，当该位置位时，SysTick 定时器计数减至 0 时会产生异常
0 ENABLE	置 1 时 SysTick 定时器使能，否则计数会被禁止

6.6.2.3.SysTick 重装载值寄存器 (LOAD)

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

位域	功能
31-24	保留
23-0 RELOAD	指定 SysTick 定时器的重装载值

6.6.2.4.SysTick 当前值寄存器 (VAL)

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

位域	功能
31-24	保留
23-0 CURRENT	读取值为 SysTick 定时器的当前数值，写入任何值都会清除寄存器，COUNTFLAG 也会清零（不会引起 SysTick 异常）

6.6.2.5.SyTick 校准值寄存器(CALIB)

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

位域	功能
31	如果读取值为 1，就表示由于没有外部参考时钟，SysTick 定时器总是使用内核时钟；如果为 0，则表示有

位域	功能
NOREF	外部参考时钟可供使用。
30 SKEW	如果为 1，则表示 TENMS 域不准确
29-24	保留
23-0 TENMS	10 毫秒校准值



7. 嵌入式 Flash(FLASH)

7.1. 简介

Flash 接口可管理 CPU (Cortex®-M0) AHB 对 Flash 进行的访问。该接口可针对 Flash 执行擦除和编程操作，读写保护和安全机制。

7.2. 主要特性

- FLASH 包含高达 64KB 的片上闪存，可用于存储指令和数据
- 按页和块进行管理，每个页 1KB 字节，每个块 4KB
- 支持编程、页擦除、块擦除操作
- 支持连续编程操作，最大可连续编程 4K 字节
- 支持 0/1/2 等待周期，最高读取速度 16MHz

7.3. 功能说明

注意：对 flash 进行写操作和擦除操作前，需先关闭全局中断。

7.3.1. 闪存结构

闪存块大小为 4KB。主存储闪存的每块都可以单独擦除。闪存结构细节见表 7-1 FLASH 构成。

表 7-1 FLASH 构成

闪存块	名称	大小 (字节)	地址
主储存块	第 0 块	4K	0x0800 0000 - 0x0800 0FFF
	第 1 块	4K	0x0800 1000 - 0x0800 1FFF
	第 2 块	4K	0x0800 2000 - 0x0800 2FFF
	...	...	...
	第 15 块	4K	0x0800 F000 - 0x0800 FFFF

7.3.2. 页擦除

FLASH 的页擦除功能使得主存储闪存的页内容初始化为高电平。每一页都可以被独立擦除，而不影响其他页内容。页擦除操作具体步骤如下：

1. 将待擦除的页地址写入 ADDR 寄存器；
2. 配置 CTRL 寄存器为 0x653B_0005 启动页擦除；
3. 至少插入 6 个 NOP 指令
4. 读取 ISR[BUSY]位判断擦除是否结束；
5. 读取 ISR[OPERR]位判断是否出现错误。

注意：

如果擦除地址在保护扇区，相应的操作不会进行，ISR[OPERR] = 0x2，表示擦除受保护的页。

如果擦除地址在用户配置区，且 CTRL[CFGER_EN] = 0，则相应操作不会执行，ISR[OPERR] = 0x4，表示用户配置区不可擦除。

如果擦除地址在用户配置区，且 CTRL[CFGER_EN] = 1，但保护功能已经开启，则相应操作不会执行，ISR[OPERR] = 0x5，表示配置区域因保护不可擦除。

7.3.3. 块擦除操作

块擦除操作具体步骤如下：

1. 将待擦除的块地址写入 ADDR 寄存器；
2. 配置 CTRL 寄存器为 0x653B_0007 启动块擦除；
3. 至少插入 6 个 NOP 指令
4. 读取 CTRL[START]位判断擦除操作是否结束；
5. 读取 ISR[OPERR]位判断是否出现错误。

注意：

如果待擦除扇区 ADDR 地址落在用户配置区，相应的擦除动作不会执行，ISR[OPERR] = 0x6，代表无法擦除非保护扇区。

如果擦除地址在保护扇区，则相应擦除动作不会执行，ISR[OPERR] = 0x3，代表无法擦除保护扇区。

7.3.4. 主存储块编程

主存储块支持 32 位数据编程操作，具体步骤如下：

1. 将编程地址写入 ADDR 寄存器，编程地址需要 4 字节对齐；

2. 配置 CTRL 寄存器为 0x653B_0001 启动编程；
3. 将待编程数据写入 DATA 寄存器中；
4. 读取 ISR[BUSY]位判断编程是否结束；
5. 读取 ISR[OPERR]位判断是否出现错误。

注意：

如果编程地址在保护扇区，相应的操作不会进行，ISR[OPERR] = 0x1，表示编程受保护的区域。

如果编程地址超出芯片 FLASH 地址范围时，ISR[OPERR] = 0x9，表示超出地址范围。

如果在启动编程后 8 个时钟周期内未将数据写入 DATA 寄存器，ISR[OPERR] = 0x8，表示操作超时。

通过将编程数据长度写入 CTRL[LENGTH]位可以实现连续编程模式 (写入的长度 = 编程数据字长 - 1)。在连续编程模式下，只需要连续将编程数据写入 DATA 寄存器即可，可通过读取 CTRL[LENGTH]获取剩余编程字长。

7.3.5. FLASH 读操作

FLASH 速度较慢，在主频较高时需要设置读等待时间。

CPU 主频	读等待周期
<16MHz	0
>=16MHz, <32MHz	1
>=32MHz, <=48MHz	2

7.3.6. 用户配置区及保护配置

系统上电时将会自动载入用户配置。如果用户在正常程序执行过程中修改了用户配置区的内容，可以通过 CFG_RELOAD 寄存器重新载入用户配置区。

用户的保护信息可以查看 FLASH_RDPORT 和 FLASH_WRPROT 寄存器。

用户配置信息可以查看 CFG_INFO0 和 CFG_INFO1 寄存器。

7.4. FLASH 中断

中断事件	事件标志	使能控制位
FLASH 操作完成 或者发生错误	INTF	IEN

7.5. FLASH 寄存器

7.5.1. FLASH 寄存器映射

表 7-2 FLASH 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	FLASH IP 版本号 (IP_VERSION)	r	0x0000_0001
0x04	FLASH 地址寄存器 (ADDR)	rw	0x0000_0000
0x08	FLASH 数据寄存器 (DATA)	rw	0x0000_0000
0x0C	FLASH 控制寄存器 (CTRL)	rw	0x0000_0000
0x10	FLASH 等待寄存器 (WAIT)	rw	0x0000_0002
0x14	FLASH 配置重载入寄存器 (CFG_RELOAD)	rc_wl	0x0000_0000
0x18	FLASH 中断使能寄存器 (IEN)	rw	0x0000_0000
0x1C	FLASH 中断状态寄存器 (ISR)	rc_wl	0x0000_0000
0x380	FLASH 配置信息区 0 寄存器 (CFG_INFO0)	r	0x0000_FFFF
0x384	FLASH 配置信息区 1 寄存器 (CFG_INFO1)	r	0x0000_000D
0x390	FLASH 读保护区 0 寄存器 (RDPROT0)	r	0x0000_0000
0x3A0	FLASH 写保护区 0 寄存器 (WRPROT0)	r	0x0000_0000

7.5.2. FLASH IP 版本号(IP_VERSION)

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

位域	功能
31-16	保留
15-8 MVER	Main Version Number
7-0 SVER	Sub Version Number

7.5.3. FLASH 地址寄存器(ADDR)

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

位域	功能
31-0 ADDR	FLASH 编程地址寄存器

7.5.4. FLASH 数据寄存器 (DATA)

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

位域	功能
31-0 DATA	FLASH 编程数据寄存器

7.5.5. FLASH 控制寄存器(CTRL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LENGTH										-	CFGER_ EN	OP_MODE		START	
rw										-	rw	rw		rc_w1	

位域	功能
31-16 KEY	写入 0x653B 才能执行 START 命令

位域	功能
15-6 LENGTH	写数据长度位
5	保留
4 CFGER_EN	用户配置区擦除使能控制位，置 1 使能 若该位未使能，擦除该区域将报错并产生中断 若该位使能，同时所有区域均未设置保护，则可擦除该区域，若有区域设置为保护，擦除该区域将会报错并产生中断
3-1 OP_MODE	操作模式使能 0x1: 编程使能 0x2: 页擦除使能 0x3: sector 擦除使能 0x4: 主程序区全擦除，仅测试模式有效 0x5: 主程序区及用户配置区全擦除，仅测试模式有效 其他: 无效
0 START	写操作时: 写 1 自动清零，启动 OP_MODE 定义的操作，写 0 无效 读操作时: 显示当前 FLASH 操作是否完成，1 代表正在操作，0 代表完成

7.5.6. FLASH 等待寄存器(WAIT)

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

位域	功能
31-2	保留
1-0 WAIT	"FLASH 读等待周期 <16MHz 时， WAIT 可设置为 0 个等待周期 ≥16MHz, <32MHz 时， WAIT 可配置为 1 个等待周期 ≥32MHz, ≤48MHz 时， WAIT 可配置为 2 个等待周期"

7.5.7. FLASH 配置重载入寄存器(CFG_RELOAD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
																-
																-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																RELOAD
																rc_w1

位域	功能
----	----

位域	功能
31-1	保留
0 RELOAD	写 1 清零，重新载入用户信息区

7.5.8. FLASH 中断使能寄存器(IEN)

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

位域	功能
31-1	保留
0 IEN	FLASH 操作中断使能位

7.5.9. FLASH 中断状态寄存器(ISR)

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

位域	功能
31 BUSY	FLASH 操作状态位 1: 忙操作 0: 操作完成
30-8	保留
7-4 OPERR	FLASH 错误标志位 0: 操作正常完成，无错误 1: 编程受保护的区域 2: 擦除受保护的页 3: 擦除受保护的扇区 4: 用户配置区擦除未使能时，擦除用户配置区 5: 用户配置区擦除使能，但有其他区域受保护时，擦除用户配置区 6: 禁止按扇区擦除 NVR1 扇区 7: 用户模式下执行全擦除指令

位域	功能
	8: FLASH 写操作超时 9: 编程或者擦除 FLASH 地址不在地址空间内
3-1	保留
0 INTF	读操作时: FLASH 中断标志位, 只在 IEN 为高时有效, 表示 FLASH 操作发生错误或者操作完成。 写操作时: 写 1 清零, 清除中断标志位, 同时错误信息也会清除

7.5.10. FLASH 配置信息区 0 寄存器(CFG_INFO0)

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

位域	功能
31-16	保留
15-8 CFG_WDT	IWDG 配置区域
7-0 CFG_BOR	BOR 配置区域

7.5.11. FLASH 配置信息区 1 寄存器(CFG_INFO1)

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

位域	功能
31-3	保留
2 SWD_DIS	1: 禁止 SWD 调试 0: 使能 SWD 调试"
1 BOOT_SEL	1: 选择 SRAM 作为 BOOT 0: 选择 FLASH 作为 BOOT
0	1: nRST 端口作为 GPIO

位域	功能
PIN_RST	0: nRST 端口作为 RST

7.5.12. FLASH 读保护区 0 寄存器(RDPROT0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDPROT_SEC15	RDPROT_SEC14	RDPROT_SEC13	RDPROT_SEC12	RDPROT_SEC11	RDPROT_SEC10	RDPROT_SEC9	RDPROT_SEC8	RDPROT_SEC7	RDPROT_SEC6	RDPROT_SEC5	RDPROT_SEC4	RDPROT_SEC3	RDPROT_SEC2	RDPROT_SEC1	RDPROT_SEC0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	功能
31-16	保留
15 RDPROT_SEC15	置位时 0x0800F000-0x0800FFFF 4K 空间读保护
14 RDPROT_SEC14	置位时 0x0800E000-0x0800EFFF 4K 空间读保护
13 RDPROT_SEC13	置位时 0x0800D000-0x0800DFFF 4K 空间读保护
12 RDPROT_SEC12	置位时 0x0800C000-0x0800CFFF 4K 空间读保护
11 RDPROT_SEC11	置位时 0x0800B000-0x0800BFFF 4K 空间读保护
10 RDPROT_SEC10	置位时 0x0800A000-0x0800AFFF 4K 空间读保护
9 RDPROT_SEC9	置位时 0x08009000-0x08009FFF 4K 空间读保护
8 RDPROT_SEC8	置位时 0x08008000-0x08008FFF 4K 空间读保护
7 RDPROT_SEC7	置位时 0x08007000-0x08007FFF 4K 空间读保护
6 RDPROT_SEC6	置位时 0x08006000-0x08006FFF 4K 空间读保护
5 RDPROT_SEC5	置位时 0x08005000-0x08005FFF 4K 空间读保护
4 RDPROT_SEC4	置位时 0x08004000-0x08004FFF 4K 空间读保护
3 RDPROT_SEC3	置位时 0x08003000-0x08003FFF 4K 空间读保护
2	置位时 0x08002000-0x08002FFF 4K 空间读保护

位域	功能
RDPROT_SEC2	
1 RDPROT_SEC1	置位时 0x08001000-0x08001FFF 4K 空间读保护
0 RDPROT_SEC0	置位时 0x08000000-0x08000FFF 4K 空间读保护

7.5.13. FLASH 写保护区 0 寄存器(WRPROT0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPROT_SEC15	WRPROT_SEC14	WRPROT_SEC13	WRPROT_SEC12	WRPROT_SEC11	WRPROT_SEC10	WRPROT_SEC9	WRPROT_SEC8	WRPROT_SEC7	WRPROT_SEC6	WRPROT_SEC5	WRPROT_SEC4	WRPROT_SEC3	WRPROT_SEC2	WRPROT_SEC1	WRPROT_SEC0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	功能
31-16	保留
15 WRPROT_SEC15	置位时 0x0800F000-0x0800FFFF 4K 空间写保护
14 WRPROT_SEC14	置位时 0x0800E000-0x0800EFFF 4K 空间写保护
13 WRPROT_SEC13	置位时 0x0800D000-0x0800DFFF 4K 空间写保护
12 WRPROT_SEC12	置位时 0x0800C000-0x0800CFFF 4K 空间写保护
11 WRPROT_SEC11	置位时 0x0800B000-0x0800BFFF 4K 空间写保护
10 WRPROT_SEC10	置位时 0x0800A000-0x0800AFFF 4K 空间写保护
9 WRPROT_SEC9	置位时 0x08009000-0x08009FFF 4K 空间写保护
8 WRPROT_SEC8	置位时 0x08008000-0x08008FFF 4K 空间写保护
7 WRPROT_SEC7	置位时 0x08007000-0x08007FFF 4K 空间写保护
6 WRPROT_SEC6	置位时 0x08006000-0x08006FFF 4K 空间写保护
5 WRPROT_SEC5	置位时 0x08005000-0x08005FFF 4K 空间写保护
4	置位时 0x08004000-0x08004FFF 4K 空间写保护

位域	功能
WRPROT_SEC4	
3 WRPROT_SEC3	置位时 0x08003000-0x08003FFF 4K 空间写保护
2 WRPROT_SEC2	置位时 0x08002000-0x08002FFF 4K 空间写保护
1 WRPROT_SEC1	置位时 0x08001000-0x08001FFF 4K 空间写保护
0 WRPROT_SEC0	置位时 0x08000000-0x08000FFF 4K 空间写保护



8. 电源控制(PWR)

8.1. 简介

PWR 电源控制主要用途有：

- 停止模式时，外设工作模式配置
- 欠压复位功能配置

8.2. 功能说明

8.2.1. 低功耗模式

系统复位或上电复位后，MCU 处于运行状态。实现较低功耗的方法有两种：减慢系统时钟、关闭未使用的外设的时钟。此外，低功耗模式可以实现更低的功耗，分别是睡眠模式和停止模式。

8.2.1.1. 睡眠模式

睡眠模式和内核的 SLEEPING 模式相对应。在睡眠模式下，仅关闭内核时钟。如需进入睡眠模式，只要清除内核的系统控制寄存器中的 SLEEPDEEP 位，并执行一条 WFI 或 WFE 指令即可。如果通过执行 WFI 指令进入睡眠模式，任何中断都可以唤醒系统；如果通过执行 WFE 指令进入睡眠模式，任何唤醒事件都可以唤醒系统（如果 SEVONPEND 为 1，任何中断都可以唤醒系统）。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间是最短的。

根据内核的系统控制寄存器的 SLEEPONEXIT 位，有两种睡眠进入机制可选：

Sleep-now: 如果 SLEEPONEXIT = 0，一旦执行 WFI 或 WFE 指令，MCU 立即进入睡眠模式；

Sleep-on-exit: 如果 SLEEPONEXIT = 1，当系统从最低优先级的中断处理程序返回时，MCU 立即进入睡眠模式。

8.2.1.2. 停止模式

停止模式与内核的 SLEEPDEEP 模式相对应。在停止模式下，内部高速时钟 HSI 关闭，SRAM 和寄存器中的内容被保留，FLASH、BANDGAP、LDO 可根据需要打开或关闭。在进入停止模式之前，先将内核系统控制寄存器的 SLEEPDEEP 位置 1，然后再执行 WFI 或 WFE 指令即可进入停止模式。如果通过执行 WFI 指令进入停止模式，任何 EXTI 中断都可以唤醒系统；如果通过执行 WFE 指令进入停止模式，任何来自 EXTI 的事件都可以唤醒系统（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统）。

如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

表 8-1 停止模式汇总

模式	睡眠	停止
描述	仅关闭内核时钟	关闭 HSI
LDO 状态	开启	开启/低功耗模式
配置	SLEEPDEEP = 0	SLEEPDEEP = 1
进入指令	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（或 SEVONPEND=1 时的中断）均可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒； 若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒
唤醒延迟	无	如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间

8.2.2. 欠压复位

欠压复位功能用于电压低于设定阈值时保持系统处于复位状态，防止程序运行异常。

8.3. PWR 中断

PWR 中断使用时，如果系统电压低于 BOD 设定阈值，系统将进入 PWR 中断，PWR 中断挂接在 EXTI。

8.4. PWR 寄存器

8.4.1. PWR 寄存器映射

表 8-2 PWR 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	PWR 控制寄存器 (CR)	rw	0x0000_0000
0x04	BOR 控制寄存器 (BORCR)	rw	0x0000_0021

8.4.2. PWR 控制寄存器(CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								LP_MOD			BG_STO		HLDO_L		LPDS
-								E		-	P	-	PMOD	-	
-								rw		-	rw	-	rw	-	rw

位域	功能
31-8	保留
7 LP_MODE	ADC 模块进入低功耗模式选择信号 0: ADC 模块不进入低功耗模式 1: ADC 模块进入低功耗模式
6-5	保留
4 BG_STOP	STOP模式后, 关闭Bandgap 0: 进入STOP模式时, Bandgap保持开启 1: 进入 STOP 模式后, 关闭 Bandgap
3	保留
2 HLDO_LPMOD	STOP模式后, HLDO进入低功耗模式选择信号 0: HLDO不进入低功耗模式 1: HLDO进入低功耗模式 该配置不控制HLDO的开关, 只控制HLDO是否进入低功耗模式。
1	保留
0 LPDS	STOP模式后, HLDO开关 0: 进入STOP模式时, HLDO开启 1: 进入STOP模式后, HLDO关闭 该配置控制 HLDO 的开关, 不控制 HLDO 的模式

8.4.3. BOR 控制寄存器(BORCR)

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

位域	描述
31-16	保留
15-12 BOD_SEL	BOD 电压选择 0000: 2.0V 0001: 2.2V 0010: 2.4V 0011: 2.6V 0100: 2.8V 0101: 3.0V 0110: 3.7V 0111: 4.3V 1xxx: 保留
11-9	保留
8 BOD_EN	0: 禁止BOD 1: 使能BOD
7-6	保留
4-5 BOR_LEVEL	BOR 电压选择 00: 2.1V 01: 2.4V 1x: 1.9V
3-1	保留
0 BOR_EN	0: 禁止BOR 1: 使能 BOR

9. 复位和时钟控制(RCC)

9.1. 复位控制

复位控制包括两种控制方式：电源复位和系统复位

9.1.1. 电源复位

当发生上电复位（POR）或者欠压复位（BOR）事件，就会产生电源复位。

9.1.2. 系统复位

当发生以下任一事件时，产生系统复位：

- 上电复位（POR_RSTn）
- 外部引脚复位（NRST）
- 窗口看门狗复位（WWDG_RSTn）
- 独立看门狗复位（IWDG_RSTn）
- 软件复位（SW_RSTn）

9.2. 时钟控制

芯片提供以下时钟源：

- 内部 48M RC 时钟（HSI）
- 内部 32K RC 时钟（LSI）
- 外部 32.768K 低速晶体振荡器时钟（LSE）

从电源复位启动之后，默认选择 HSI 时钟 12 分频（4M）作为系统时钟源。

9.3. 功能说明

9.3.1. 时钟树

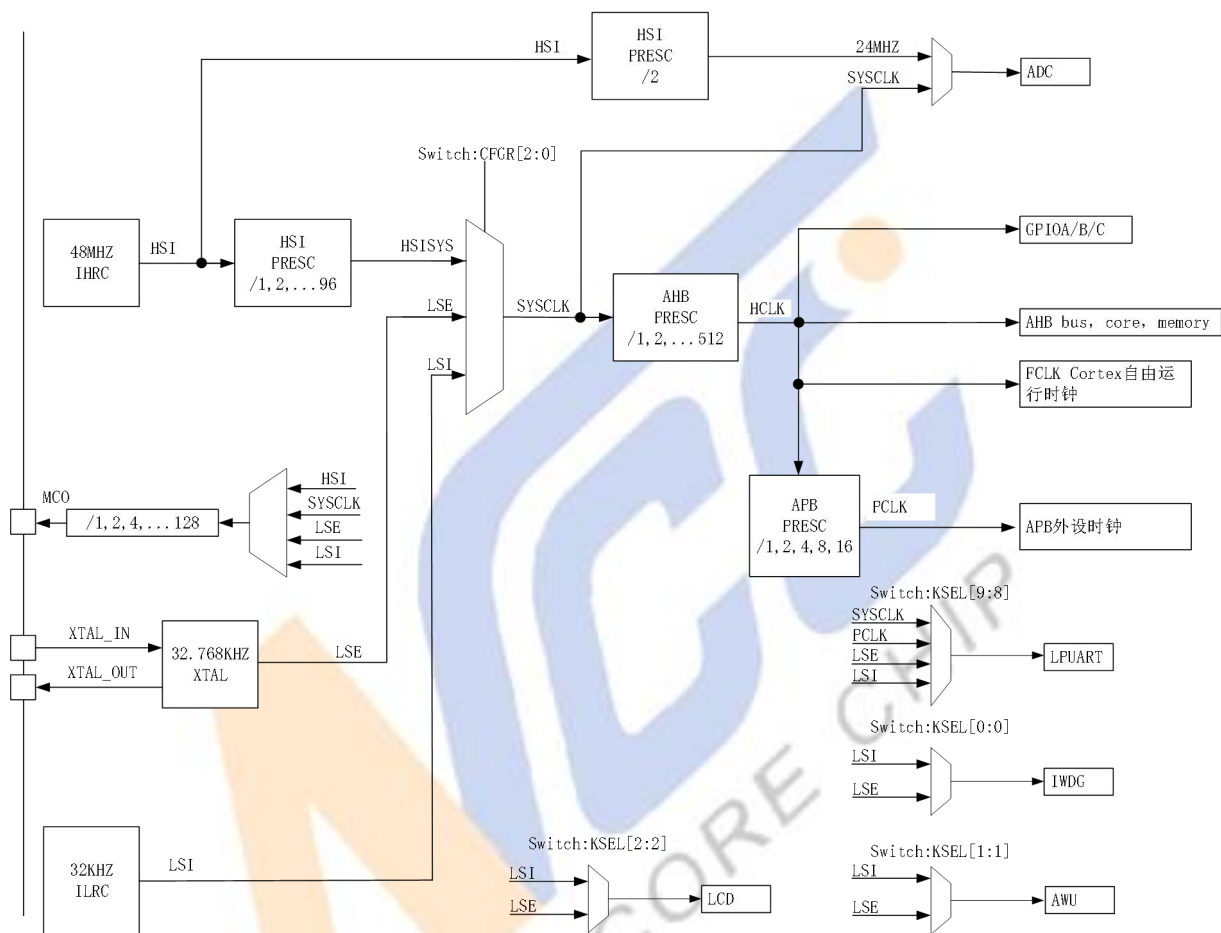


图 9-1 时钟树

9.3.2. 时钟输出功能

微控制器时钟输出 (MCO) 功能允许将时钟输出到外部 MCO 引脚上。可选择以下时钟之一作为 MCO 时钟：

- LSI
- LSE
- HSI
- SYSCLK

此选择由 CFGR[MCOSSEL]位控制。所选时钟可通过 CFGR[MCOPRE]进行分频后输出。

9.3.3. LSE 监测功能

LSE 监测功能在外部时钟停振或发生故障时，系统将产生异常标志，时钟切换为内部振荡器并触发 NMI 中断。

注意：若要监测 LSE 时钟，需在使能 LSE 前先打开 LSI；若要读取异常标志，需先将时钟监测过慢/过快中断使能。



9.4. RCC 寄存器

9.4.1. RCC 寄存器映射

表 9-1 RCC 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	时钟控制寄存器 (CR)	rw	0x0000_3F21
0x04	时钟配置寄存器 (CFGR)	rw	0x0000_0000
0x08	时钟中断使能寄存器 (CIER)	rw	0x0000_0000
0x0C	时钟中断标志寄存器 (CIFR)	r	0x0000_0008
0x10	时钟中断标志清除寄存器 (CICR)	w	0x0000_0000
0x14	AHB 周边复位寄存器 (AHBRSTR)	rw	0x0000_0000
0x18	APB 周边复位寄存器 1 (APBRSTR1)	rw	0x0000_0000
0x1C	AHB 周边时钟使能寄存器 (AHBENR)	rw	0x0000_0000
0x20	APB 周边时钟使能寄存器 1 (APBENR1)	rw	0x0000_0000
0x24	AHB 周边 SLEEP 模式下时钟使能寄存器 (AHBSMENR)	rw	0x0000_00FF
0x28	APB 周边 SLEEP 模式下时钟使能寄存器 1 (APBSMENR1)	rw	0xFFFF_FFFF
0x2C	外设时钟选择寄存器 (KSEL)	rw	0x0000_0000
0x30	复位状态寄存器 (RSTID)	rw	0x8800_0000
0x34	LSI 控制寄存器 (LSICR)	rw	0x0000_0000
0x38	LSE 控制寄存器 (LSECR)	rw	0x0000_0000
0x40	时钟监测控制寄存器 (SCMCR)	rw	0x0000_0000
0x44	时钟异常标志寄存器 (SCMFLAG)	rw	0x0000_0000
0x48	APB 周边复位寄存器 2 (APBRSTR2)	rw	0x0000_0000
0x4C	APB 周边时钟使能寄存器 2 (APBENR2)	rw	0x0000_0000
0x50	APB 周边 SLEEP 模式下时钟使能寄存器 2 (APBSMERN2)	rw	0xFFFF_FFFF

9.4.2. 时钟控制寄存器 (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	HSIRDY_TIM						-	-	HSIDIV			-	HSI KERON	HSION
-	-	rw						-	-	rw			-	rw	rw

位域	功能
31-14	保留
13-8 HSIRDY_TIM	HSI 稳定时间 稳定时间=HSI_RDY_TIME* 4 * T _{HSI_48M}
7-6	保留
5-3 HSIDIV	HSI 48M 时钟分频设置 000: 不分频 001: 2 分频 010: 4 分频 011: 6 分频 100: 12 分频 101: 24 分频 110: 48 分频 111: 96 分频
2	保留
1 HSIKERON	始终为外设内核使能 HSI48M 0: HSI48M 作为周边设备独立时钟源时总是关 1: HSI48M 作为周边设备独立时钟源时总是开
0 HSION	HSI48M 时钟使能 0: HSI48M 关 1: HSI48M 开

9.4.3. 时钟配置寄存器 (CFGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	MCPRE			-	MCOSEL			-	-	-	-	-	-	-	-
-	rw			-	rw			-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	PPRE			HPRE			-	-	SWS			SW			
-	rw			rw			-	-	r			rw			

位域	功能
31	保留

位域	功能
30-28 MCOPRE	时钟输出预分频 000: 不分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频
27	保留
26-24 MCOSEL	时钟输出源选择 000: 无 001: SYSCCLK 010: 无 011: HSI 100: 无 101: 无 110: LSI 111: LSE
23-15	保留
14-12 PPRE	APB 预分频 0xx: 1 分频 100: 2 分频 101: 4 分频 110: 8 分频 111: 16 分频
11-8 HPRE	AHB 预分频 0xxx: 1 分频 1000: 2 分频 1001: 4 分频 1010: 8 分频 1011: 16 分频 1100: 64 分频 1101: 128 分频 1110: 256 分频 1111: 512 分频
7-6	保留
5-3 SWS	系统时钟切换状态 000: HSISYS 001: 保留 010: 保留 011: LSI 100: LSE Others:保留
2-0 SW	系统时钟切换 000: HSISYS 001: 保留 010: 保留

位域	功能
	011: LSI 100: LSE Others:保留

9.4.4. 时钟中断使能寄存器 (CIER)

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

位域	功能
31-4	保留
3 HSIRDYIE	HSI48M 有效中断使能 0: 无效 1: 使能
2	保留
1 LSERDYIE	LSE 有效中断使能 0: 无效 1: 使能
0 LSIRDYIE	LSI 有效中断使能 0: 无效 1: 使能

9.4.5. 时钟中断标记寄存器 (CIFR)

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

位域	功能
31-4	保留
3	HSI48M 有效中断标记 0: 无中断

位域	功能
HSIRDYF	1: 有中断
2	保留
1 LSERDYF	LSE 有效中断标记 0: 无中断 1: 有中断
0 LSIRDYF	LSI 有效中断标记 0: 无中断 1: 有中断

9.4.6. 时钟中断清除寄存器 (CICR)

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

位域	功能
31-4	保留
3 HSIRDYC	HSI48M 有效中断清除 0: 无效 1: 清除中断标记
2	保留
1 LSERDYC	LSE 有效中断清除 0: 无效 1: 清除中断标记
0 LSIRDYC	LSI 有效中断清除 0: 无效 1: 清除中断标记

9.4.7. AHB 周边复位寄存器 (AHBRSTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	GPIOCRS	GPIOBRS	GPIOARS	CRCRST
-	-	-	-	-	-	-	-	-	-	-	-	T	T	T	
-	-	-	-	-	-	-	-	-	-	-	-	rw	rw	rw	rw

位域	功能
----	----

位域	功能
31-4	保留
3 GPIOCRST	GPIOC 复位 0: 无效 1: 复位
2 GPIOBRST	GPIOB 复位 0: 无效 1: 复位
1 GPIOARST	GPIOA 复位 0: 无效 1: 复位
0 CRCRST	CRC 复位 0: 无效 1: 复位

9.4.8. APB 周边复位寄存器 1 (APBRSTR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-		-	-	-	-	-	-	-	-	-	-	-	LPUART1RST
-	-	-		-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	PWRRST	-	I2C1RST	-	-	-	WWDGRST	IWDGRST	AWURST	TIM7RST	TIM6RST	-	-	TIM2RST
-	-	rw	-	rw	-	-	-	rw	rw	rw	rw	rw	-	-	rw

位域	功能
31-17	保留
16 LPUART1RST	LPUART1 复位 0: 无效 1: 复位
15-14	保留
13 PWRRST	PWR 复位 0: 无效 1: 复位
12	保留
11 I2C1RST	I2C1 复位 0: 无效 1: 复位
10-8	保留
7 WWDGRST	WWDG 复位 0: 无效 1: 复位
6 IWDGRST	IWDG 复位 0: 无效 1: 复位

位域	功能
5 AWURST	AWU 复位 0: 无效 1: 复位
4 TIM7RST	TIM7 复位 0: 无效 1: 复位
3 TIM6RST	TIM6 复位 0: 无效 1: 复位
2-1	保留
0 TIM2RST	TIM2 复位 0: 无效 1: 复位

9.4.9. AHB 周边使能寄存器 (AHBENR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	GPIOCEN	GPIOBEN	GPIOAEN	CRCEN
-	-	-	-	-	-	-	-	-	-	-	-	N	N	N	N
-	-	-	-	-	-	-	-	-	-	-	-	rw	rw	rw	rw

位域	功能
31-4	保留
3 GPIOCEN	GPIOC 使能 0: 无效 1: 使能
2 GPIOBEN	GPIOB 使能 0: 无效 1: 使能
1 GPIOAEN	GPIOA 使能 0: 无效 1: 使能
0 CRCEN	CRC 使能 0: 无效 1: 使能

9.4.10. APB 周边使能寄存器 1 (APBENR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LPUART1EN
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	PWREN	-	I2C1EN	-	-	-	WWDGEN	IWDGEN	AWUEN	TIM7EN	TIM6EN	-	-	TIM2EN
-	-	rw	-	rw	-	-	-	rw	rw	rw	rw	rw	-	-	rw

位域	功能
31-17	保留
16 LPUART1EN	LPUART1 使能 0: 无效 1: 使能
15-14	保留
13 PWREN	PWR 使能 0: 无效 1: 使能
12	保留
11 I2C1EN	I2C1 使能 0: 无效 1: 使能
10-8	保留
7 WWDGEN	WWDG 使能 0: 无效 1: 使能
6 IWDGEN	IWDG 使能 0: 无效 1: 使能
5 AWUEN	AWU 使能 0: 无效 1: 使能
4 TIM7EN	TIM7 使能 0: 无效 1: 使能
3 TIM6EN	TIM6 使能 0: 无效 1: 使能
2-1	保留
0 TIM2EN	TIM2 使能 0: 无效 1: 使能

9.4.11. AHB 周边 SLEEP 模式时钟使能寄存器 (AHBSMENR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	GPIOC	GPIOB	GPIOA	CRC
-	-	-	-	-	-	-	-	-	-	-	-	MEN	MEN	MEN	SME
-	-	-	-	-	-	-	-	-	-	-	-	rw	rw	rw	rw

位域	功能
31-4	保留
3 GPIOCMEN	GPIOC SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
2 GPIOBMEN	GPIOB SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
1 GPIOAMEN	GPIOA SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
0 CRCMEN	CRC SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能

9.4.12. APB 周边 SLEEP 模式时钟使能寄存器 1 (APBSMENR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LPUART1SMEN
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	PWRSME	-	I2C1SME	-	-	-	WWDGS	IWDGSM	AWUSM	TIM7SM	TIM6SM	-	-	TIM2SM
-	-	N	-	N	-	-	-	MEN	EN	EN	EN	EN	-	-	EN
-	-	rw	-	rw	-	-	-	rw	rw	rw	rw	rw	-	-	rw

位域	功能
31-17	保留
16 LPUART1EN	LPUART1 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
15-14	保留

位域	功能
13 PWREN	PWR SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
12	保留
11 I2C1EN	I2C1 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
10-8	保留
7 WWDGEN	WWDG SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
6 IWDGEN	IWDG SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
5 AWUEN	AWU SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
4 TIM7EN	TIM7 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
3 TIM6EN	TIM6 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
2-1	保留
0 TIM2EN	TIM2 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能

9.4.13. 外设时钟选择寄存器 (KSEL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	ADCSEL	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	rw	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	LPUART1SEL	-	-	-	-	-	-	LCD_SE	AWU_SE	IWDG_S
-	-	-	-	-	-	rw	-	-	-	-	-	-	L	L	EL
-	-	-	-	-	-	rw	-	-	-	-	-	-	-	rw	rw

位域	功能
31	保留
30 ADCSEL	ADC 时钟选择 0: 系统时钟 1: 24MHz 时钟 (HSI 时钟两分频)
29-10	保留
9-8	LPUART1 时钟选择

位域	功能
LPUARTISEL	00: PCLK 01: SYSCLK 10: LSI 11: LSE
7-3	保留
2 LCD_SEL	LCD 时钟选择 0: LSI 1: LSE
1 AWUSEL	AWU 时钟选择 0: LSI 1: LSE
0 IWDGSEL	IWDG 时钟选择 0: LSI 1: LSE

9.4.14. 复位状态寄存器 (RSTID)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFGRST	WWDGR	IWDGRS	SFTRSTF	PWRSTF	PINRSTF	-	-	RMVF	-	-	-	-	-	-	-
F	STF	TF				-	-	W	-	-	-	-	-	-	-
r	r	r	r	r	r	-	-	w	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位域	功能
31 CFGRSTF	FLASH 信息区载入成功标志 0: 无复位 1: 有复位
30 WWDGSTF	窗口看门狗复位标志 0: 无复位 1: 有复位
29 IWDGSTF	独立看门狗复位标志 0: 无复位 1: 有复位
28 SFTRSTF	软件复位标志 0: 无复位 1: 有复位
27 PWRSTF	POR/BOR/PDR 复位标志 0: 无复位 1: 有复位
26 PINRSTF	NRST 管脚复位标志 0: 无复位 1: 有复位
25-24	保留

位域	功能
23 RMVF	清除复位标志: 0: 无效 1: 将复位标志清除
22-0	保留

9.4.15. LSI 控制寄存器 (LSICR)

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

位域	功能
31-6	保留
5-4 LSI_RDY_TIME	LSI 稳定时间 LSI 稳定时间=LSI_RDY_TIME * 500 ms
3-1	保留
0 LSION	LSI 振荡器使能 0: 禁止 1: 使能

9.4.16. LSE 控制寄存器 (LSECR)

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

位域	功能
31-6	保留
5-4 LSE_RDY_TIME E	LSE 稳定时间 LSE 稳定时间=LSE_RDY_TIME * 500 ms
3-1	保留
0 LSEON	LSE 振荡器使能 0: 禁止 1: 使能

9.4.17. 时钟监测控制寄存器 (SCMCR)

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

位域	功能
31-4	保留
3 MT	监测周期 0: 一个监测周期 1: 半个监测周期
2 FASTIE	LSE 频率过快中断使能 0: 禁止中断 1: SCMFLAG[FASTF]=1 时生成中断
1 SLOWIE	LSE 频率过慢中断使能 0: 禁止中断 1: SCMFLAG [SLOWF]=1 时生成中断
0 LSEMEN	LSE 时钟异常监测使能 0: 禁止 1: 使能

9.4.18. 时钟异常标志寄存器 (SCMFLAG)

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

位域	功能
31-2	保留
1 FASTF	LSE 频率过快标志 0: 无故障 1: 有故障
0 SLOWF	LSE 频率过慢标志 0: 无故障 1: 有故障

9.4.19. APB 周边复位寄存器 2 (APBRSTR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TKRST
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG RST	SYSCFG RST	LCD RST	ADC RST	-	SPI1 RST	USART1 RST	-	-	-	TIM16 RST	-	-	-	-	TIM10 RST
rw	rw	rw	rw	-	rw	rw	-	-	-	rw	-	-	-	-	rw

位域	功能
31-17	保留
16 TKRST	TK 复位 0: 无效 1: 复位
15 DBG RST	DBG 复位 0: 无效 1: 复位
14 SYSCFG RST	SYSCFG 复位 0: 无效 1: 复位
13 LCD RST	LCD 复位 0: 无效 1: 复位
12 ADC RST	ADC 复位 0: 无效 1: 复位
11	保留
10 SPI1 RST	SPI1 复位 0: 无效 1: 复位
9 USART1 RST	USART1 复位 0: 无效 1: 复位
8-6	保留
5 TIM16 RST	TIM16 复位 0: 无效 1: 复位
4-1	保留
0 TIM10 RST	TIM10 复位 0: 无效 1: 复位

9.4.20. APB 周边使能寄存器 2 (APBENR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TKEN
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBGEN	SYSCFG EN	LCDEN	ADCEN	-	SPI1EN	USART1 EN	-	-	-	TIM16EN	-	-	-	-	TIM10EN
rw	rw	rw	rw	-	rw	rw	-	-	-	rw	-	-	-	-	rw

位域	功能
31-17	保留
16 TKEN	TK 使能 0: 无效 1: 使能
15 DBGEN	DBG 使能 0: 无效 1: 使能
14 SYSCFGEN	SYSCFG 使能 0: 无效 1: 使能
13 LCDEN	LCD 使能 0: 无效 1: 使能
12 ADCEN	ADC 使能 0: 无效 1: 使能
11	保留
10 SPI1EN	SPI1 使能 0: 无效 1: 使能
9 USART1EN	USART1 使能 0: 无效 1: 使能
8-6	保留
5 TIM16EN	TIM16 使能 0: 无效 1: 使能
4-1	保留
0 TIM10EN	TIM10 使能 0: 无效 1: 使能

9.4.21. APB 周边 SLEEP 模式时钟使能寄存器 2 (APBSMENR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TKSMEN
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBGSMEN	SYSCFG	LCDSME	ADCSME	-	SPI1SME	USART1	-	-	-	TIM16S	-	-	-	-	TIM10S
N	SMEN	N	N	-	N	SMEN	-	-	-	MEN	-	-	-	-	MEN
rw	rw	rw	rw	-	rw	rw	-	-	-	rw	-	-	-	-	rw

位域	功能
31-17	保留
16 TKSMEN	TK 使能 0: 无效 1: SLEEP 模式时钟使能
15 DBGSMEN	DBG SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
14 SYSCFGSMEN	SYSCFG SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
13 LCDSMEN	LCD 使能 0: 无效 1: SLEEP 模式时钟使能
12 ADCSMEN	ADC SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
11	保留
10 SPI1SMEN	SPI1 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
9 USART1SMEN	USART1 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
8-6	保留
5 TIM16SMEN	TIM16 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能
4-1	保留
0 TIM10SMEN	TIM10 SLEEP 模式时钟使能 0: 无效 1: SLEEP 模式时钟使能

10. 通用输入输出接口(GPIO)

10.1. 简介

通用 I/O 引脚(GPIO), 分别为 PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC13, 各片上设备用其来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。

GPIO 引脚上的外部中断在中断/事件控制器 (EXTI) 中有相关的控制和配置寄存器。

GPIO 端口和外设功能共用引脚, GPIO 引脚通过配置复用功能选择可以用作外设功能输入/输出引脚。

每个 GPIO 引脚可以由软件配置为输出(推挽)、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。

10.2. 主要特性

- 输出状态: 推挽
- 从输出数据寄存器(ODR)或周边设备(复用功能)输出数据
- 每个 I/O 速率选择
- 输入状态: 浮空, 上/下拉, 模拟
- 输入数据到输入数据寄存器(IDR)或周边设备(复用功能)的输入端
- 位设置/复位寄存器(BSRR)提供对 ODR 的位操作
- 锁定机制(LCKR)能锁定 I/O 的配置
- 模拟功能
- 复用功能选择寄存器

10.3. 功能说明

10.3.1. GPIO 引脚配置

在复位期间及复位刚刚完成后, 复用功能尚未激活, 大多数 I/O 端口被配置为模拟模式。

复位后, 调试引脚处于复用功能上拉/下拉状态:

PA6: SWCLK 处于下拉状态

PA7: SWDIO 处于上拉状态

注: PA6 与 PA7 复用功能需要小心操作, 如果在程序中将其复用, 需要用外部复位后才能进入调试模式。

当引脚配置为输出后，写入到输出数据寄存器 (ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下使用输出驱动器。

输入数据寄存器 (IDR) 每个 AHB 时钟周期捕获一次 I/O 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 PUPDR 寄存器中的值来打开/关闭。

10.3.2. 引脚复用功能

有两个寄存器可用来从每个 I/O 可用的复用功能输入/输出中进行选择。借助这些寄存器，用户可根据应用程序的要求将某个复用功能连接到其它某个引脚。

这意味着可使用 AFRL 和 AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为指定 I/O 的复用功能输入/输出选择一个通道即可。

表 10-1 GPIO 引脚复用

Name	AF0	AF1	AF2	AF3
PA0			LED_S17	
PA1			LED_S18	
PA2	I2C1_SDA	TIM2_CH1N	LED_S19	
PA3	I2C1_SCL	TIM2_CH1P	LED_S20	
PA4	USART1_TXD	TIM2_CH2N	LED_S21	
PA5	USART1_RXD	TIM2_CH2P	LED_S22	
PA6	SWCLK	TIM2_CH3N	LED_S23	
PA7	SWDIO	TIM2_CH3P	LED_S24	
PA8			LED_S7	
PA9			LED_S6	
PA10			LED_S5	
PA11			LED_S4	
PA12			LED_S3	
PA13			LED_S2	
PA14			LED_S1	
PA15			LED_S8	
PB0	TIM16_CH1	USART1_RXD	LED_S9	
PB1	TIM16_CH2	USART1_TXD	LED_S10	
PB2		SPI1_MISO	LED_S11	
PB3		SPI1_MOSI	LED_S12	
PB4		SPI1_SCK	LED_S13	
PB5	I2C1_SMBA	SPI1_SSN	LED_S14	
PB6	I2C1_SCL	LPUART1_RXD	LED_S15	
PB7	I2C1_SDA	LPUART1_TXD	LED_S16	
PB8			LED_C1	
PB9			LED_C2	
PB10			LED_C3	
PB11			LED_C4	
PB12			LED_C5	
PB13			LED_C6	
PB14	TIM10_CH2		LED_C7	
PB15	TIM10_CH1	IR_OUT	LED_C8	
PC0			LED_S28	
PC1			LED_S27	
PC2			LED_S26	
PC3		USART1_RXD	LED_S25	
PC4	MCO	USART1_TXD		
PC5		I2C1_SDA		
PC6	TIM10_CH2	I2C1_SCL		
PC7	TIM10_CH1			
PC8				

Name	AF0	AF1	AF2	AF3
PC9	USART1_CK			
PC10				
PC11	BUZZER	USART1_CK		
PC12	TIM16_CH1			
PC13	TIM16_CH2			

10.3.3. 输入配置

当 GPIO 引脚配置为输入时：

- 施密特触发器输入使能
- 可选择是否打开上拉和下拉电阻
- 输入数据寄存器即为当前 IO 状态
- 输出缓冲器禁止

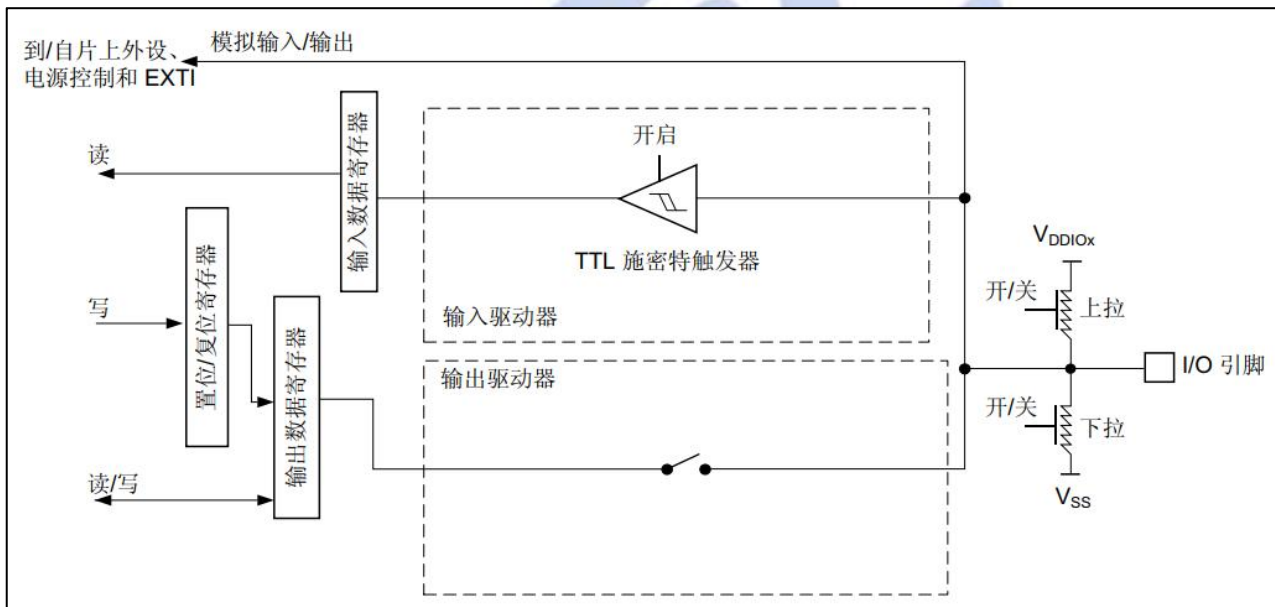


图 10-1 输入浮空/上拉/下拉配置

10.3.4. 外部中断线/唤醒线

所有端口都具有外部中断功能。为了使用外部中断线，端口必须配置为输入模式。

请参见 [扩展中断与事件控制器 \(EXTI\)](#)。

10.3.5. 输出配置

当 GPIO 引脚配置为输出时：

- 输出缓冲器被打开：
 - 推挽模式：输出寄存器设置为“0”时，I/O 引脚输出低电平；输出寄存器设置为“1”时，I/O 引脚输出高电平

- 施密特触发器输入被打开
- 根据 PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 对输入数据寄存器的读访问可获取当前 I/O 状态
- 对输出数据寄存器的读访问可获取最后的写入值

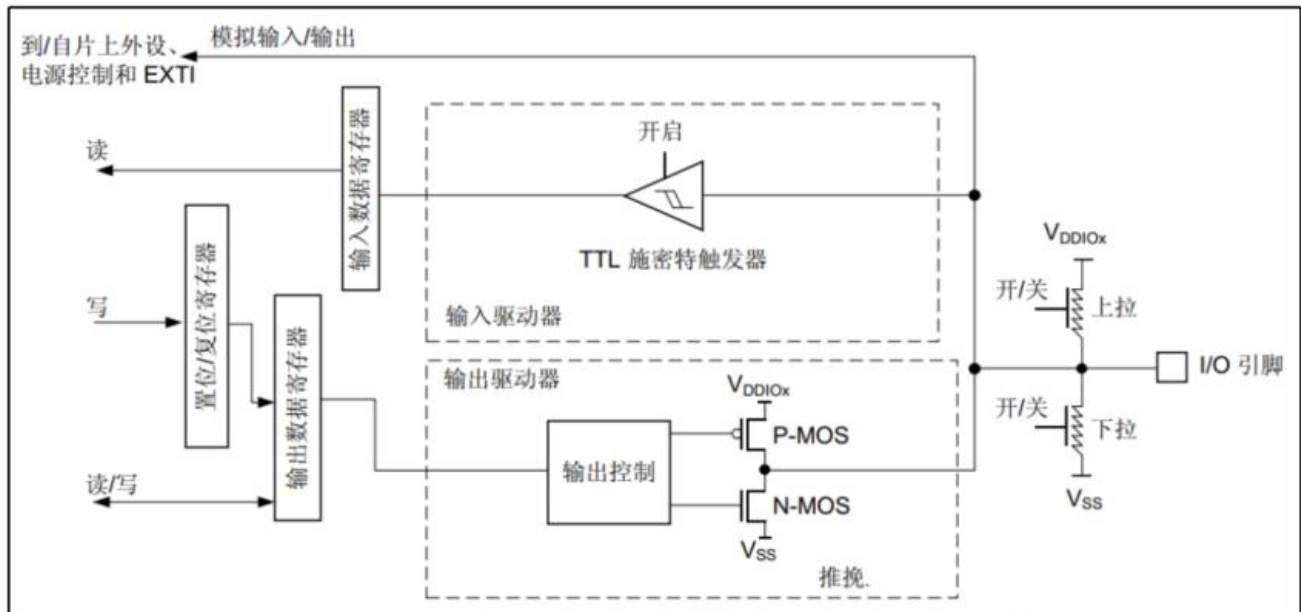


图 10-2 输出配置

10.3.6. 模拟配置

将 I/O 端口编程为模拟配置时：

- 输出缓冲器被禁止
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)
- 弱上拉和下拉电阻被硬件关闭
- 对输入数据寄存器的读访问值为“0”

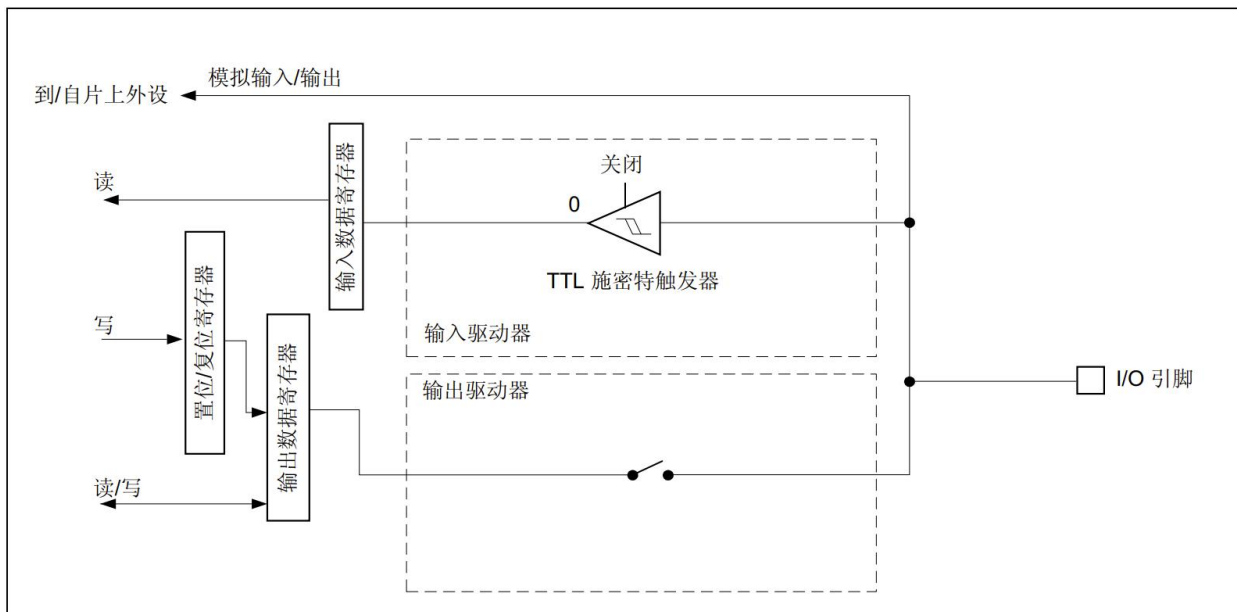


图 10-3 模拟配置

10.3.7. 复用功能配置

将 I/O 端口编程为复用功能时：

- 可将输出缓冲器配置为推挽模式
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）
- 施密特触发器输入被打开
- 根据 PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 对输入数据寄存器的读访问可获取 I/O 状态

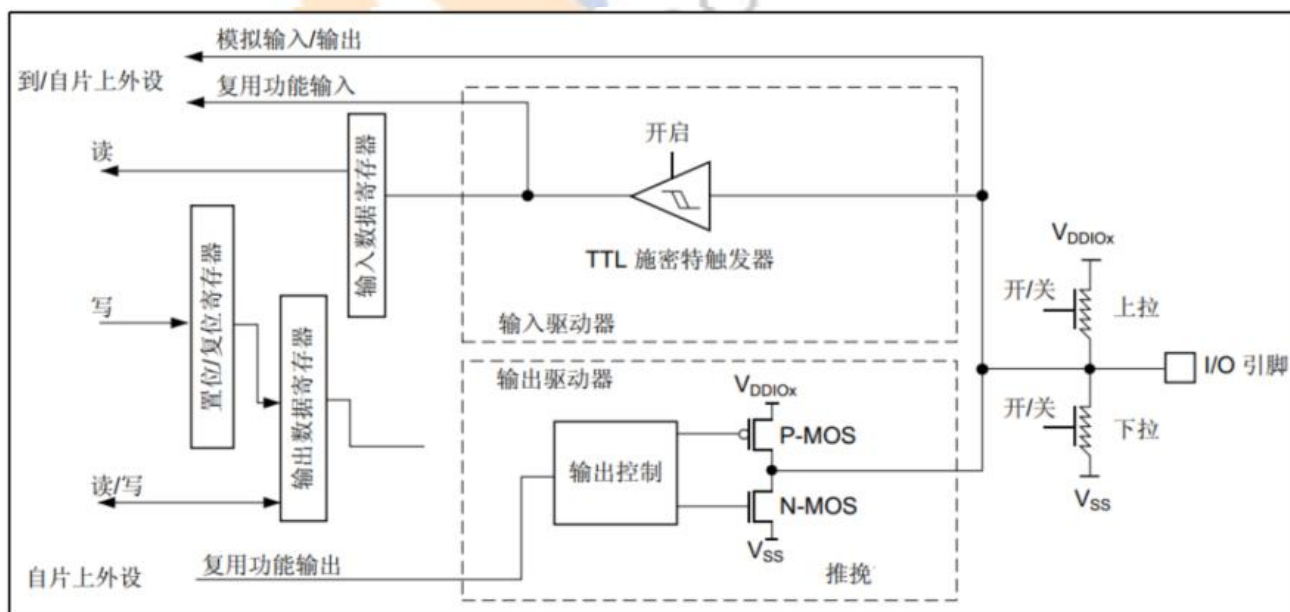


图 10-4 复用功能配置

10.3.8. GPIO 锁定功能

通过将特定的写序列应用到 LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 MODER、PUPDR、AFRL 和 AFRH。

要对 LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。每个 LCKR 位都会冻结控制寄存器（MODER、PUPDR、AFRL 和 AFRH）中的对应位。

LOCK 序列只能通过对 LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 LCKR 的第 16 位必须与 [15:0] 位同时置位。

10.4. GPIO 寄存器

10.4.1. GPIO 寄存器映射

表 10-2 GPIO 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	GPIOx 端口模式寄存器 (MODER)	rw	0xFFFF_AFFF (端口 A) 0xFFFF_FFFF (端口 B) 0x0FFF_FFFF (端口 C)
0x0c	GPIOx 端口上拉/下拉寄存器 (PUPDR)	rw	0x0020_6000 (端口 A) 0x8208_0820 (端口 B) 0x0210_0020 (端口 C)
0x10	GPIOx 端口输入数据寄存器 (IDR)	r	0x0000_XXXX
0x14	GPIOx 端口输出数据寄存器 (ODR)	rw	0x0000_0000
0x18	GPIOx 端口位置位/复位寄存器 (BSRR)	w	0x0000_0000
0x1c	GPIOx 端口配置锁定寄存器 (LCKR)	rw	0x0000_0000
0x20	GPIOx 复用功能低位寄存器 (AFRL)	rw	0x0000_0000
0x24	GPIOx 复用功能低位寄存器 (AFRH)	rw	0x0000_0000
0x28	GPIOx 端口位复位寄存器 (BRR)	w	0x0000_0000

10.4.2. GPIOx 端口模式寄存器 (MODER) (x=A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15	MODE14	MODE13	MODE12	MODE11	MODE10	MODE9	MODE8								
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0								
rw	rw	rw	rw	rw	rw	rw	rw								

位域	功能
31-0 MODEy	配置端口 x 的 y 引脚工作模式 00: GPIO 输入 01: GPIO 输出 (此时输入使能也是有效的) 10: 复用功能 (此时输入使能也是有效的) 11: 模拟通道 (default)

10.4.3. GPIOx 端口上拉/下拉寄存器 (PUPDR) (x = A 到 C)

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

位域	功能
31-0 OSPEEDy	配置端口 x 的 y 引脚上拉或下拉 00: 无上拉或下拉 01: 上拉 10: 下拉 11: 保留

10.4.4. GPIOx 端口输入数据寄存器 (IDR) (x = A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	功能
15-0 IDy	端口 x 的 y 引脚的输入值

10.4.5. GPIOx 端口输出数据寄存器 (ODR) (x = A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
15-0 ODy	端口 x 的 y 引脚的输出值

10.4.6. GPIOx 端口位置位/复位寄存器 (BSRR) (x = A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
位域	功能														
31-16 BRy	复位端口 x 的 y 引脚 0: 无效操作 1: 复位 Pin y 引脚														
15-0 BSy	置位端口 x 的 y 引脚 0: 无效操作 1: 置位 Pin y 引脚														

10.4.7. GPIOx 端口配置锁定寄存器 (LCKR) (x = A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位域	功能														
16 LCKK	锁定键写序列: WR LCKR[16] = '1' + LCKR[15:0] WR LCKR[16] = '0' + LCKR[15:0] WR LCKR[16] = '1' + LCKR[15:0] RD LCKR RD LCKR[16] = '1' (此读操作作为可选操作, 但它可确认锁定已激活)														
15-0 LCKy	端口 x 的引脚 y 锁定 0: 未锁定 1: 已锁定														

10.4.8. GPIOx 复用功能低位寄存器 (AFRL) (x = A 到 C)

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

AFSEL3	AFSEL2	AFSEL1	AFSEL0
rw	rw	rw	rw

位域	功能
31-0 AFSELy	端口 x 的 y 引脚复用功能选择 AFSELy 选择: 0000: AF0 0001: AF1 0010: AF2 0011: AF3

10.4.9. GPIOx 复用功能高位寄存器 (AFRH) (x = A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15				AFSEL14				AFSEL13				AFSEL12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11				AFSEL10				AFSEL9				AFSEL8			
rw				rw				rw				rw			

位域	功能
31-0 AFSELy	端口 x 的 y 引脚复用功能选择, AFSELy 选择: 0000: AF0 0001: AF1 0010: AF2 0011: AF3

10.4.10. GPIOx 端口位复位寄存器 (BRR) (x = A 到 C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	功能
15-0 BRy	端口 x 的引脚 y 复位 0: 无操作 1: 复位 Pin y 引脚

11. 系统配置控制器 (SYSCFG)

11.1. 简介

器件配有一组配置寄存器。系统配置控制器的主要用途如下：

- 配置 IR 输出模式及其输出极性
- 定时器和 ADC 主从模式触发源配置
- 重新映射位于代码区域开头的存储空间
- 标记晶振异常错误

11.2. SYSCFG 寄存器

11.2.1. SYSCFG 寄存器映射

表 11-1 SYSCFG 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	SYSCFG 配置寄存器 1 (CFGR1)	rw	0x0000_0000
0x04	SYSCFG 配置寄存器 2 (CFGR2)	rw	0x0000_0000
0x08	NMI 中标志寄存器 (NMI_SR)	rw	0x0000_0000

11.2.2. SYSCFG 配置寄存器 1 (CFGR1)

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

位域	功能
31-2	保留
1-0 MEM_MODE	存储器映射选择位 00: 将主 Flash 映射到地址 0x0000 0000 01: 保留 10: 将主 Flash 映射到地址 0x0000 0000 11: 将 SRAM 映射到地址 0x0000 0000

11.2.3. SYSCFG 配置寄存器 2 (CFGR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	IR_POL	IR_MODE
-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	ADC_TSEL			-	TIM16_TSEL			-	TIM10_TSEL			-	TIM2_TSEL		
-	rw			-	rw			-	rw			-	rw		

位域	功能
31-18	保留
17 IR_POL	IR 输出极性选择 0: IRTIM (IR_OUT) 的输出反相 1: IRTIM (IR_OUT) 的输出非反相 注意: IRTIM 输出信号的极性由 IR_POL 位控制, 可以通过将此位置 1 来反相。
16 IR_MODE	IR 调制包络信号选择 0: TIM16 1: USART1 注意: 所有标准 IR 脉冲调制模式都可通过编程两个定时器输出比较通道获得。 TIM10 用于生成高频载波信号, 而 TIM16 (或者 USART1) 生成调制包络, 具体需遵循 IR_MOD 位的设置。
15	保留
14-12 ADC_TSEL	ADC Trigger 选择: 000: 无 001: TIM2 010: TIM6 011: TIM7 100: 无 101: TIM10 110: TIM16 111: EXTI11
11	保留
10-8 TIM16_TSEL	TIM16 Trigger 选择: 000: 无 001: 无 010: TIM2 011: TIM6 100: TIM7 101: 无 110: TIM10 111: 无
7	保留
6-4 TIM10_TSEL	TIM10 Trigger 选择: 000: 无 001: 无 010: TIM2 011: TIM6

位域	功能
	100: TIM7 101: 无 110: TIM16 111: 无
3	保留
2-0 TIM2_TSEL	TIM2 Trigger 选择: 000: 无 001: 无 010: TIM6 011: TIM7 100: 无 101: TIM10 110: TIM16 111: 无

11.2.4. NMI 中断标志寄存器 (NMI_SR)

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

位域	功能
31-3	保留
2 SCM_ERR	LSE 晶振异常标志位 0: 无异常 1: 有异常 <i>注意: 此标志位需搭配 RCC 中 SCMCRR 寄存器使用, 只有时钟过快或过慢监测中断使能, 且晶振异常时, 该标志位才会置起。</i>
1-0	保留

12. 扩展中断与事件控制器(EXTI)

12.1. 简介

EXTI 包括相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI 有两种触发类型：上升沿触发和任意沿触发。EXTI 中的每一个边沿检测电路都可以独立配置和屏蔽。

12.2. 主要特性

- 可由任何输入作为事件唤醒系统。
- 可为没有唤醒标记的源周边设备产生唤醒标记和中断。
- 可配置事件（从 I/O，没有中断挂起的周边设备）。
 - 支持上升沿触发选择
 - 独立的上升沿中断挂起标志位
 - 分离的中断和事件生成控制，用于生成 CPU 唤醒，中断及事件
 - 支持软件触发
- 直接事件（有中断挂起的周边设备）
 - 固定的上升沿触发
 - EXTI 中没有中断挂起标志位
 - 分离的中断和事件生成控制，用于生成 CPU 唤醒及事件
 - 无软件触发
- 支持 I/O 端口选择

12.3. 功能说明

12.3.1. EXTI 框图

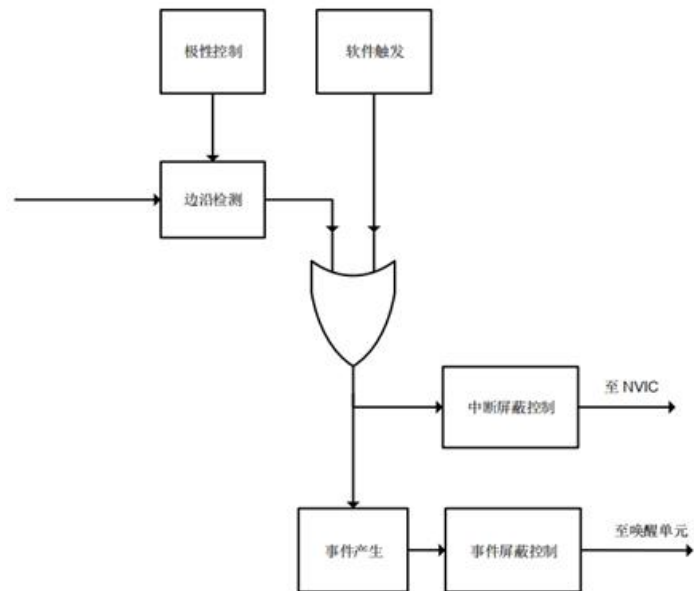


图 12-1 EXTI 框图

12.3.2. 外部中断与事件功能

EXTI 触发源包括来自 I/O 管脚的 16 根线以及来自内部模块的 5 根线。

线类型为可配置表示可以通过 RTSR 和 FTSR 配置触发边沿，线类型为直接表示不需要进行触发边沿配置。

表 12-1 EXTI 线连接

EXTI 线	线源	线类型
0-15	GPIO	可配置
16	PWR	可配置
17	LPUART1 wakeup	直接
18	SPI1 wakeup	直接
19	TOUCH wakeup	直接
20	AWU wakeup	直接

EXTI 唤醒时间： $t = 100\mu s$ 的 $hldo$ 建立时间 + $(2^7) \cdot T_{hsi}$ 。

12.4. EXTI 寄存器

12.4.1. EXTI 寄存器映射

表 12-2 EXTI 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	EXTI 上升沿触发选择寄存器 (RTSR)	rw	0x0000_0000
0x04	EXTI 下降沿触发选择寄存器 (FTSR)	rw	0x0000_0000
0x08	EXTI 软件中断事件触发寄存器 (SWIER)	rw	0x0000_0000
0x0C	EXTI 上升沿挂起寄存器 (RPR)	rw	0x0000_0000
0x10	EXTI 下降沿挂起寄存器 (FPR)	rw	0x0000_0000
0x60	EXTI 外部中断选择寄存器 1 (EXTICR1)	rw	0x0000_0000
0x64	EXTI 外部中断选择寄存器 2 (EXTICR2)	rw	0x0000_0000
0x68	EXTI 外部中断选择寄存器 3 (EXTICR3)	rw	0x0000_0000
0x6C	EXTI 外部中断选择寄存器 4 (EXTICR4)	rw	0x0000_0000
0x80	EXTI 中断控制寄存器 (IMR)	rw	0x0000_0000
0x84	EXTI 事件控制寄存器 (EMR)	rw	0x0000_0000

12.4.2. EXTI 上升沿触发选择寄存器 (RTSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															RT16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-17	保留
16-0 RTx	可配置线 x 的上升沿触发事件配置位 (x= 16 to 0) 每一位对应一条可配置中断/事件线 0: 禁止 1: 使能

12.4.3. EXTI 下降沿触发选择寄存器 (FTSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															FT16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-17	保留
16-0 FTx	可配置线 x 的下降沿触发事件配置位 (x= 16 to 0) 每一位对应一条可配置中断/事件线 0: 禁止 1: 使能

12.4.4. EXTI 软件中断事件触发寄存器 (SWIER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															SWI16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-17	保留
16-0 SWIx	可配置线 x 的软件上升沿事件触发 (x= 16 to 0) 每一位对应一条可配置中断/事件线，独立于 EXTI_RTSR 和 EXTI_FTSR，由硬件自动清零 0: 无效果 1: 上升沿触发

12.4.5. EXTI 上升沿挂起寄存器 (RPR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															RPIF16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPIF15	RPIF14	RPIF13	RPIF12	RPIF11	RPIF10	RPIF9	RPIF8	RPIF7	RPIF6	RPIF5	RPIF4	RPIF3	RPIF2	RPIF1	RPIF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
----	----

位域	功能
31-17	保留
16-0 RPIF _x	可配置线 x 的软件上升沿事件挂起 (x= 16 to 0) 每一位对应一条可配置中断/事件线 0: 无上升沿触发 1: 有上升沿触发

12.4.6. EXTI 下降沿挂起寄存器 (FPR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															FPIF16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPIF15	FPIF14	FPIF13	FPIF12	FPIF11	FPIF10	FPIF9	FPIF8	FPIF7	FPIF6	FPIF5	FPIF4	FPIF3	FPIF2	FPIF1	FPIF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-17	保留
16-0 FPIF _x	可配置线 x 的软件下降沿事件挂起 (x= 16 to 0) 每一位对应一条可配置中断/事件线 0: 无下降沿触发 1: 有下降沿触发

12.4.7. EXTI 外部中断选择寄存器(EXTICR_x) x=1~4

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

位域	功能
31-24 EXTIm+3	EXTIm+3 GPIO 端口选择(m = 4 * (x-1)) 0x00: PA[m+3] pin 0x01: PB[m+3] pin 0x02: PC[m+3] pin Others: reserved
23-16 EXTIm+2	EXTIm+2 GPIO 端口选择(m = 4 * (x-1)) 0x00: PA[m+2] pin 0x01: PB[m+2] pin 0x02: PC[m+2] pin Others: reserved
15-8	EXTIm+1 GPIO 端口选择(m = 4 * (x-1))

位域	功能
EXTIm+1	0x00: PA[m+1] pin 0x01: PB[m+1] pin 0x02: PC[m+1] pin Others: reserved
7-0 EXTIm	EXTIm+0 GPIO 端口选择($m = 4 * (x-1)$) 0x00: PA[m+0] pin 0x01: PB[m+0] pin 0x02: PC[m+0] pin Others: reserved

12.4.8. EXTI 中断控制寄存器 (IMR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											IM20	IM19	IM18	IM17	IM16
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-21	保留
20-0 IMx	中断唤醒 CPU 使能位($x=20:0$) 0: 中断唤醒禁止 1: 中断唤醒使能

12.4.9. EXTI 事件控制寄存器 (EMR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											EM20	EM19	EM18	EM17	EM16
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-21	保留
20-0 EMx	事件唤醒 CPU 使能位($x=20:0$) 0: 事件唤醒禁止 1: 事件唤醒使能

13. 基本定时器(TIM6/TIM7)

13.1. 简介

基本定时器 TIM6/TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。此定时器可用作通用定时器以生成时基。基本定时器 TIM6/TIM7 在本章统称为 BTIM。

13.2. 主要特性

基本定时器具有以下特性：

- TIM6/TIM7 为 16 位计数器
- 重载寄存器位数和计数器相同
- 16 位预分频器
- 计数器支持递增或递减模式
- 可配外部或系统时钟作为计数器计时时钟
- 可屏蔽的中断：更新中断，计数器上溢/下溢

基本定时器的设计框图如图所示。



图 13-1 基本定时器设计框图

13.3. BTIM 功能说明

13.3.1. 时基单元

可编程定时器的主要模块由一个 16 位递增/递减计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位(ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。详见本章寄存器描述。计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位(CEN) 置 1 时，才会启动计数器。

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位寄存器 TIMx_PSC 所控制的 16 位计数器。计数器时钟频率 $f_{CK_CNT} = f_{CK_INT}/(PSC[15:0]+1)$ 。

13.3.2. 计数模式

计数器从 0 计数到自动重载值 (TIMx_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以产生更新事件。通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这样，直到 UDIS 位中写入 0 前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。

BTIM 可以工作在递增计数模式或递减计数模式：

13.3.2.1. 递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值 (TIMx_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。

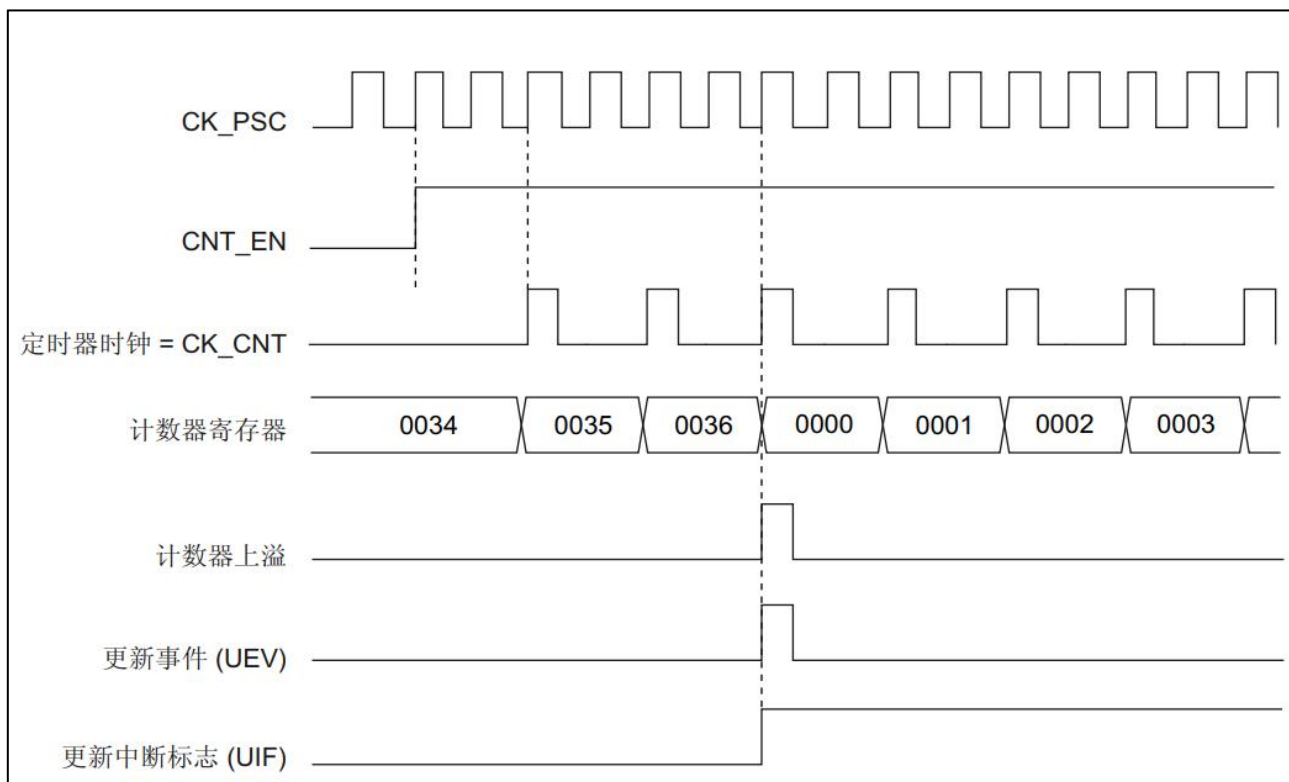


图 13-2 递增计数模式

13.3.2.2.递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

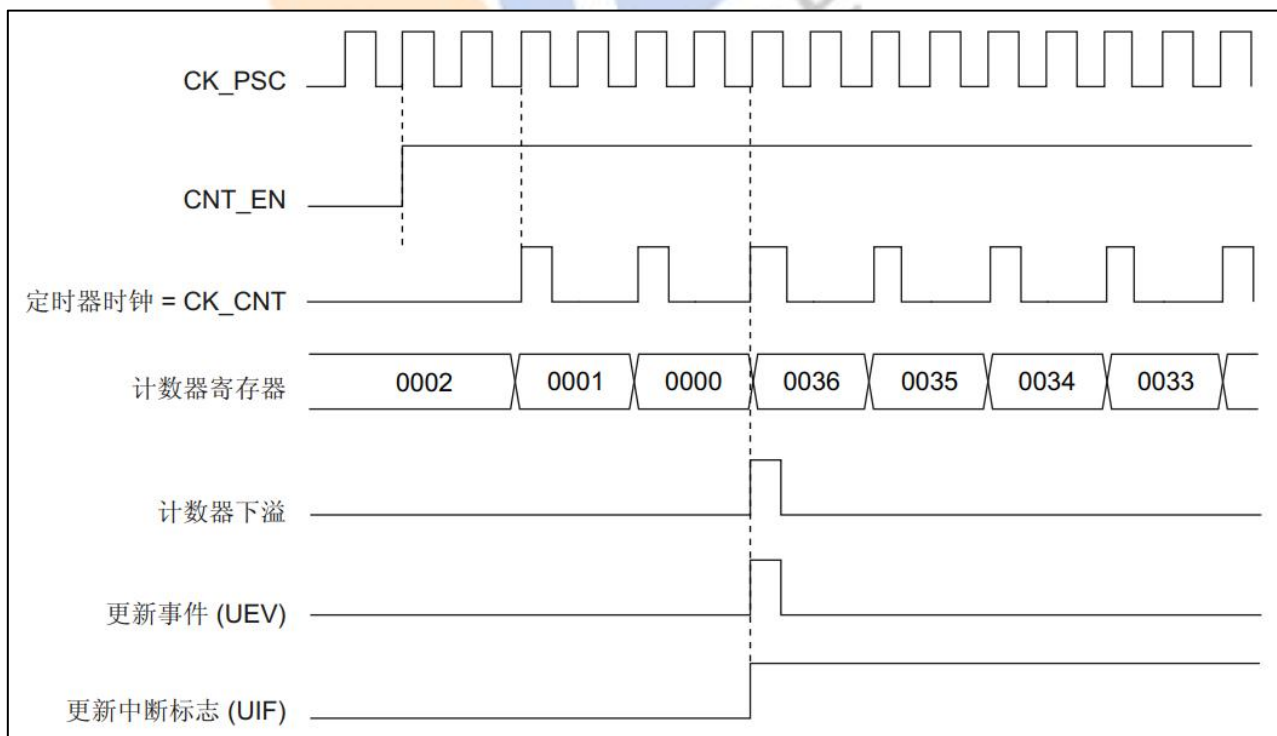


图 13-3 递减计数模式

13.3.3. 软件触发

将 TIMx_CR1 寄存器中的 OWM 位置 1，即可选择一次模式。这样，在 CEN 位置高后，发生下一更新事件 UEV 时，计数器将自动停止。该模式下只产生一个单脉冲波形。重新将 CEN 位置高后，定时器又会产生一个单脉冲波形。

将 TIMx_CR1 寄存器中的 OWM 位置 0，即可选择连续模式。这样，在 CEN 位置高后，计数器在发生更新事件时不会停止。该模式下会产生连续波形。直到 CEN 被拉低，计数器停止计数。

13.3.4. BTIM 中断

下表给出了 BTIM 的中断请求列表

表 13-1 BTIM 中断请求

中断事件	事件标志	事件标志/ 中断清除方法	中断使能控制位
更新中断	UIF	该位在发生更新事件时 由硬件置1，需通过软件清0	UIE

13.4. BTIM 寄存器

13.4.1. BTIM 寄存器映射

表 13-2 BTIM 寄存器总表

偏移	寄存器名	访问	复位值
0x00	BTIM IP 版本寄存器 (VER)	r	0x0000_0001
0x04	BTIM 控制寄存器 1 (CR1)	rw	0x0000_0000
0x08	BTIM 控制寄存器 2 (CR2)	rw	0x0000_0000
0x10	BTIM 中断使能寄存器 (IER)	rw	0x0000_0000
0x14	BTIM 状态寄存器 (SR)	rc_wl	0x0000_0000
0x1C	BTIM 事件生成寄存器 (EGR)	w	0x0000_0000
0x20	BTIM 计数器 (CNT)	r	0x0000_0000
0x24	BTIM 预分频器 (PSC)	rw	0x0000_0000
0x28	自动重载寄存器 (ARR)	rw	0x0000_0000

13.4.2. BTIM IP 版本寄存器(VER)

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

位域	功能
31-16	保留
15-8 MVER	主版本号
7-0 SVER	子版本号

13.4.3. BTIM 控制寄存器 1 (CR1)

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

位域	功能
31-6	保留
5 APRE	自动重载预装载使能 (Auto-reload preload enable) 0: TIMx_ARR 寄存器不进行缓冲 1: TIMx_ARR 寄存器进行缓冲, 该寄存器在每次发生更新事件时进行重载。
4 DIR	方向 (Direction) 0: 计数器递增计数 1: 计数器递减计数
3 OWM	一次模式 (One-wave mode) 0: 计数器在发生更新事件时不会停止计数 1: 计数器在发生下一更新事件时停止计数 (将 CEN 位清零)
2 URS	更新请求源 (Update request source) 此位由软件置 1 和清零, 用以选择 UEV 事件源。 0: 使能后, 所有以下事件都会生成更新中断: 1) 递增模式时的计数器上溢(上溢在 CNT = ARR) 2) 递减模式时的计数器下溢(下溢在 CNT = 0) 3) 将 UG 位置 1 1: 使能后, 计数器上溢/下溢会生成更新中断, 将 UG 位置 1 不生成中断。
1 UDIS	更新禁止 (Update disable) 此位由软件置 1 和清零, 用以使能或禁止更新事件(UEV) 生成。 0: 使能 UEV。然后更新影子寄存器的值。如果 UG 位置 1, 则会重新初始化计数器和预分频器。 1: 禁止 UEV。不会生成 UEV, 各影子寄存器的值保持不变。UG 位置 1 无效。
0 CEN	计数器使能 (Counter enable) 0: 禁止计数器 1: 使能计数器 在单脉冲模式下, 当发生更新事件时会自动将 CEN 位清零。

13.4.4. BTIM 控制寄存器 2 (CR2)

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

位域	功能
31-5	保留
4 CLKSEL	定时器时钟选择 (clock selection) 0: 选择 PCLK 作为系统时钟 1: 保留
3 CLKEN	定时器时钟使能 (clock enable) 0: 禁止定时器时钟 1: 打开定时器时钟。 在配置 CEN=1 前, 必须将 CLKEN 配置为 1, 否则计数器时钟将无法正常工作。
2-0 MMS	主模式选择 (Master mode selection) 这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下: 000: 不输出。TRGO 保持低电平。 001: 复位——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRGO 上的信号相比实际复位会有延迟。 010: 使能——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。 011: 更新——选择更新事件作为触发输出 (TRGO)。 100, 101, 110, 111: 保留

13.4.5. BTIM 中断使能寄存器 (IER)

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

位域	功能
31-1	保留

位域	功能
0 UIE	更新中断使能 (Update interrupt enable) 0: 禁止更新中断。 1: 使能更新中断。

13.4.6. BTIM 状态寄存器 (SR)

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

位域	功能
31-1	保留
0 UIF	更新中断标志 (Update interrupt flag) 该位在发生更新事件时通过硬件置 1。但需要通过软件写 1 清零。 0: 未发生更新。 1: 更新中断挂起。该位在以下情况下由硬件置 1: - 上溢且当 TIMx_CR1 寄存器中 UDIS = “0” 时。 - 当由于 TIMx_CR1 寄存器中 URS = “0” 且 UDIS = “0” 而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

13.4.7. BTIM 时间生成寄存器 (EGR)

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

位域	功能
31-1	保留
0 UG	更新生成 (Update generation) 该位可通过软件置 1，并由硬件自动清零。 0: 不执行任何操作

位域	功能
	1: 重新初始化计数器并生成寄存器更新事件。预分频器计数器也将清零, 但预分频比不受影响。计数器清零。使能后可发生相关中断

13.4.8. BTIM 计数器 (CNT)

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

位域	功能
31-16	保留
15-0 CNT	计数器值 (Counter value)

13.4.9. BTIM 预分频器 (PSC)

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

位域	功能
31-16	保留
15-0 PSC	预分频器值 (Prescaler value) 计数器时钟频率 $f_{CK_CNT} = f_{CK_INT}/(PSC[15:0]+1)$ 。 PSC 代表实际预分频寄存器的值。只有 CEN 为 0 时可以写入。

13.4.10. BTIM 自动重载寄存器（ARR）

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

位域	功能
31-16	保留
15-0 ARR	自动重载值 (Auto-reload value)ARR 为要装载到实际自动重载寄存器的值。当自动重载值为空时，计数器不工作。

14. 通用定时器(TIM2/TIM10/TIM16)

14.1. 简介

通用定时器(TIM2/TIM10/TIM16)包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获）或生成输出波形（输出比较和 PWM）。使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。这些定时器彼此完全独立，不共享任何资源。通用定时器 TIM2/TIM10/TIM16 在本章统称为 GTIM。

14.2. 主要特性

通用定时器 TIM2/TIM10/TIM16 具有以下特性：

- 16 位自动重载计数器，计数器支持递增或递减模式。
- 16 位可编程预分频器，可用于对计数器时钟频率进行分频
- 可配置输出：单脉冲或 PWM
- 可配系统时钟作为计数器计时时钟
- 可选软件/硬件输入触发
- 可选软件/硬件输入复位计数器和预分频器
- TIM2 硬件可配 0~4 个通道，TIM10 和 TIM16 硬件分别可配 0~2 个通道，每个通道均支持：
 - 输入捕获，支持输入信号数字滤波及事件滤波
 - 比较输出，输出极性可配
- 可屏蔽的中断：
 - 每个通道独立的输入捕获中断
 - 每个通道独立的输出匹配中断
 - 触发中断
 - 更新中断，计数器上溢/下溢

GTIM 通用定时器框图如下：

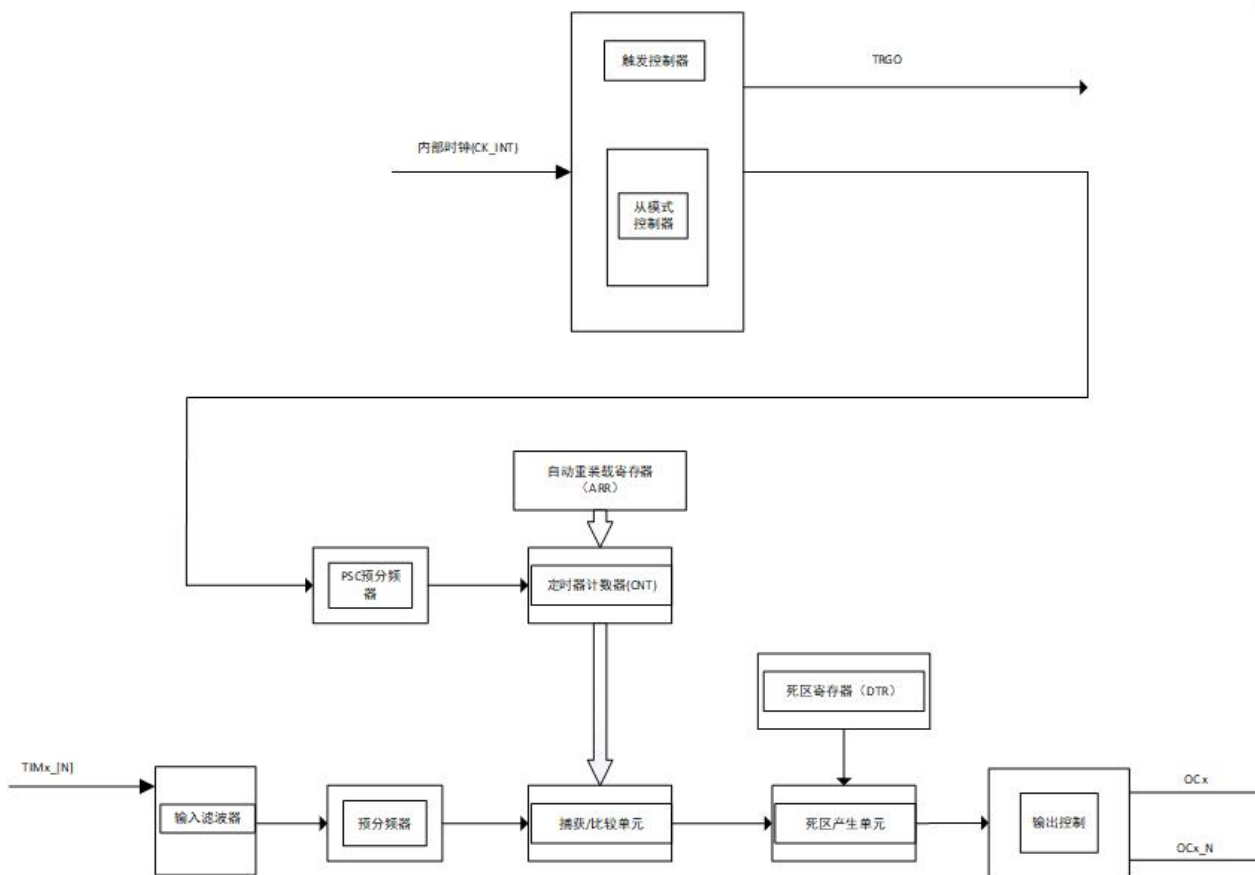


图 14-1 GTIM 通用定时器框图

14.3. 功能说明

14.3.1. 模式及功能选项说明

定时器名称	TIM2	TIM10	TIM16
计数模式	向上，向下	向上，向下	向上，向下
捕获比较通道数	4 路捕获比较通道	2 路捕获比较通道	2 路捕获比较通道
互补输出和死区插入	√	×	×
从设备控制器	TIM6/TIM7/ TIM10/TIM16	TIM2/TIM6/ TIM7 /TIM16	TIM2/TIM6/ TIM7/ TIM10

14.3.2. 时基单元

可编程定时器的主要模块由一个 16 位递增/递减计数器及其相关的自动重载寄存器组成。计数器的时钟可通

过预分频器进行分频。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。详见本章寄存器描述。计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器。

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位寄存器 TIMx_PSC 所控制的 16 位计数器。计数器时钟频率 $f_{CK_CNT} = f_{CK_INT} / (PSC[15:0] + 1)$ 。

14.3.3. 计数器模式

计数器从 0 计数到自动重载值 (TIMx_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1 (通过软件或使用从模式控制器) 也可以产生更新事件。通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这样，直到 UDIS 位中写入 0 前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数 (而预分频比保持不变)。

GTIM 可以工作在递增计数模式或递减计数模式：

14.3.3.1. 递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值 (TIMx_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。

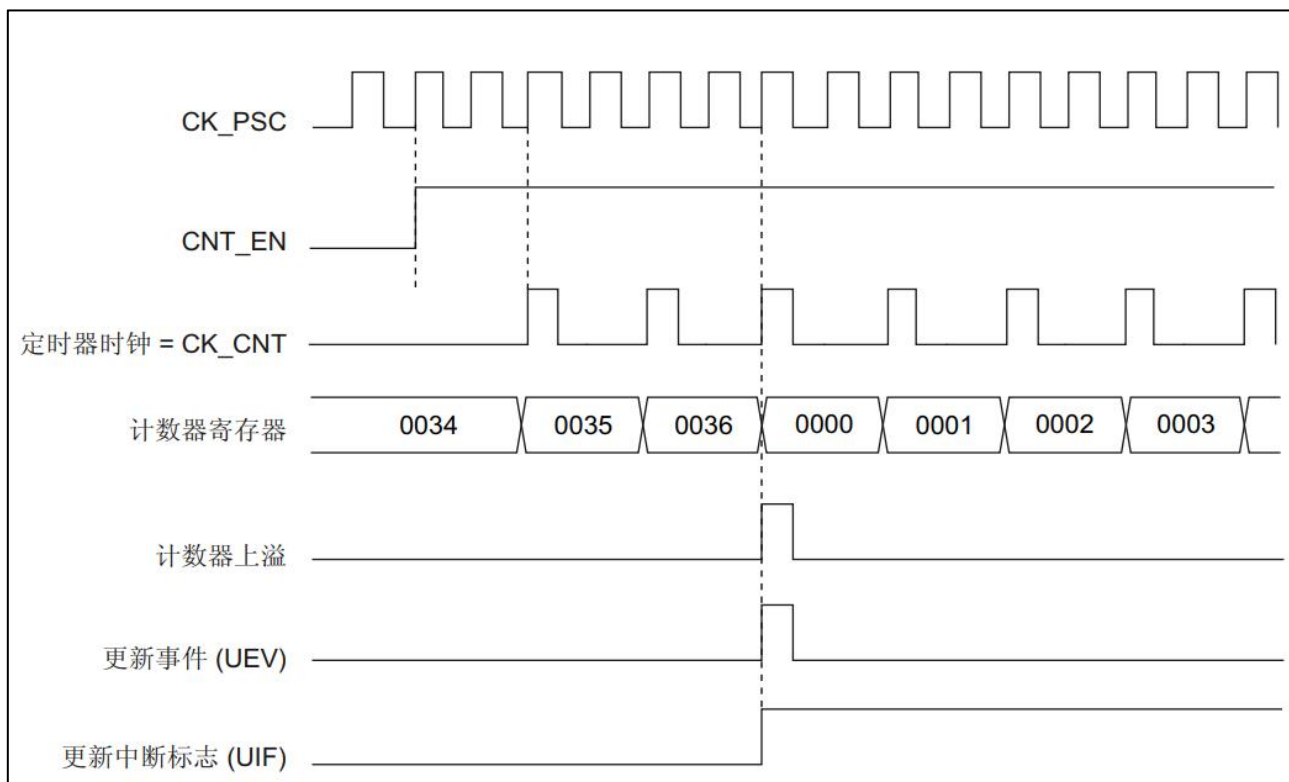


图 14-2 递增计数模式 计数器时序图

14.3.3.2.递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

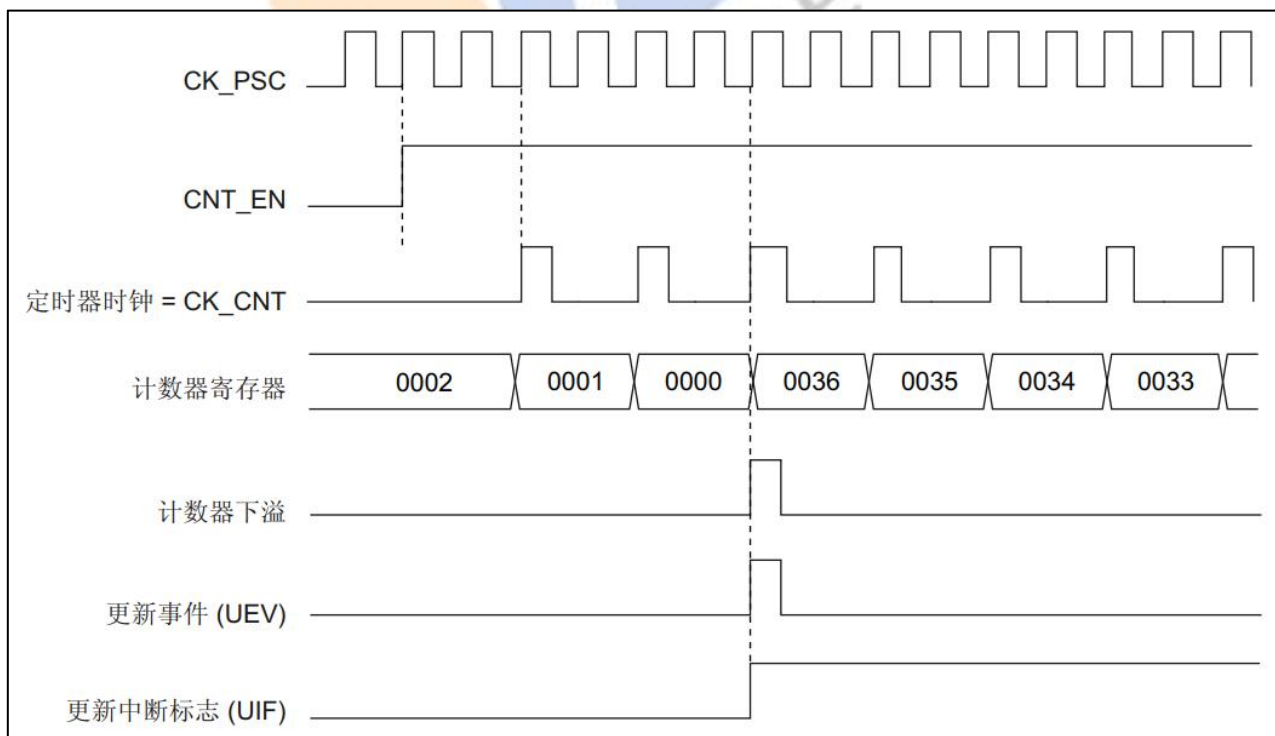


图 14-3 递减计数模式 计数器时序图

14.3.4. TIMx 时钟源

定时器计数器的预分频计数器时钟由定时器时钟 CK_INT 提供, 此 IC 无外部时钟, CK_INT 时钟即为 PCLK 时钟。

14.3.5. 输出比较/输入捕获

每个捕获/比较通道均围绕一个捕获/比较寄存器(包括一个影子寄存器)、一个捕获输入阶段(数字滤波、多路复用和预分频器)和一个输出阶段(比较器和输出控制)构建而成。输入阶段对相应的 ICn 输入进行采样, 生成一个滤波后的信号 ICnF。该信号可用作从模式控制器的触发输入, 也可用作捕获命令。该信号先进行预分频, 而后再进入捕获寄存器。捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。在捕获模式下, 捕获实际发生在影子寄存器中, 然后将影子寄存器的内容复制到预装载寄存器中。在比较模式下, 预装载寄存器的内容将复制到影子寄存器中, 然后将影子寄存器的内容与计数器进行比较。

14.3.5.1. 输入捕获

在输入捕获模式下, 当相应的 ICn 信号检测到跳变沿后, 将使用捕获/比较寄存器(TIMx_CCRn)来锁存计数器的值。发生捕获事件时, 会将相应的 CCnIF 标志(TIMx_SR1 寄存器)置 1。如果发生捕获事件时 CCnIF 标志已处于高位, 则会将重复捕获标志 CCnOF (TIMx_SR2 寄存器)置 1。可通过软件方法向 CCnIF 写入 1 来给 CCnIF 清零, 或读取存储在 TIMx_CCRn 寄存器中的已捕获数据。CCnOF 在写入 1 时清零。

14.3.5.2. 输出比较

此功能用于控制输出波形。当捕获/比较寄存器与计数器之间相匹配时, 输出比较功能:

- 将为相应的输出引脚分配一个可编程值, 该值由输出比较模式(TIMx_CCMRn 寄存器中的 OCnM 位)和输出极性(TIMx_CCER 寄存器中的 CCnP 位)定义。匹配时, 输出引脚既可保持其电平(OCnM = “000”), 也可设置为有效电平(OCnM = “001”)、无效电平(OCnM = “010”)或进行翻转(OCnM = “011”)。
- 将中断状态寄存器中的标志置 1(TIMx_SR 寄存器中的 CCxIF 位)。
- 如果相应中断使能位(TIMx_IER 寄存器中的 CCnIE 位)置 1, 将生成中断。

使用 TIMx_CCMRn 寄存器中的 OCnPE 位, 可将 TIMx_CCRn 寄存器配置为带或不带预装载寄存器。

在输出比较模式下, 更新事件 UEV 对 OCnREF 和 OCn 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲(在单脉冲模式下)。

14.3.5.3. 强制输出

在输出模式(TIMx_CCMRn 寄存器中的 CCnS 位 = ‘00’)下, 可直接由软件将每个输出比较信号(OCnREF 和 OCn)强制设置为有效电平或无效电平, 而无需考虑输出比较寄存器和计数器之间的任何比较结果。要

将输出比较信号 (OCnREF/OCn) 强制设置为有效电平, 只需向相应 TIMx_CCMRn 寄存器中的 OCnM 位写入“101”。OCnREF 进而强制设置为高电平 (OCnREF 始终为高电平有效), 同时 OCn 获取 CCnP 极性的相反值。例如: CCnP=“0” (OCn 高电平有效) 将 OCn 强制设置为高电平。通过向 TIMx_CCMRn 寄存器中的 OCnM 位写入“100”, 可将 OCnREF 信号强制设置为低电平。无论如何, TIMx_CCRn 影子寄存器与计数器之间的比较仍会执行, 而且允许将标志置 1。因此可发送相应的中断请求。

14.3.5.4. 特殊情况

在单脉冲模式下, TIx 输入的边沿检测会将 CEN 位置 1, 表示使能计数器。然后, 在计数器与比较值之间发生比较时, 将切换输出。但是, 完成这些操作需要多个时钟周期, 这会限制可能的最小延迟 (t_{DELAY} 最小值)。如果要输出延迟时间最短的波形, 可以将 TIMx_CCMRn 寄存器中的 OCnFE 位置 1。这会强制 OCnREF (和 OCn) 对激励信号做出响应, 而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时, OCnFE 才会起作用。

14.3.6. PWM 模式

PWM 脉冲宽度调制模式可以生成一个信号, 该信号频率由 TIMx_ARR 寄存器值决定, 其占空比则由 TIMx_CCRn 寄存器值决定。PWM 输出时的计数器也可以采用递增、递减模式。通过向 TIMx_CCMRn 寄存器中的 OCnM 位写入 110 (PWM 模式 1) 或 111 (PWM 模式 2), 可以独立选择各通道 (每个 OCx 输出对应一个 PWM) 的 PWM 模式。必须通过将 TIMx_CCMRn 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器, 最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器 (在递增计数下)。由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器, 因此启动计数器之前, 必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程 (使用 TIMx_CCER 寄存器的 CCxP 位)。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCnE 位置 1 来使能。请参见本章 TIMx_CCERn 寄存器说明。

在 PWM 模式 (1 或 2) 下, TIMx_CNT 总是与 TIMx_CCRn 进行比较, 以确定是 $\text{TIMx_CCRn} \leq \text{TIMx_CNT}$ 还是 $\text{TIMx_CNT} \leq \text{TIMx_CCRn}$ (取决于计数器计数方向)。不过, 为符合 OCREF_CLR 功能 (在下一个 PWM 周期之前, ETR 信号上的一个外部事件能够清除 OCREF), OCREF 信号仅在以下情况下变为有效状态:

- 比较结果发生改变
- 输出比较模式 (TIMx_CCMRn 寄存器中的 OCnM 位) 从“冻结”配置 (不进行比较, OCnM=“000”) 切换为任一 PWM 模式 (OCxM=“110”或“111”)。定时器运行期间, 可以通过软件强制 PWM 输出。

14.3.7. 互补输出和死区插入

通用定时器 TIMx 的每个通道可以输出两路互补信号, 并管理输出的关断与接通瞬间。这段时间通常称为死区, 用户必须根据与输出相连接的器件以及特性 (电平转换器的固有延迟、开关器件产生的延迟) 来调整

死区时间。每路输出可以独立选择输出极性，可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。可通过配置 TIMx_CCER 寄存器中的 DT_EN 位插入死区。

如果 DT_EN=1, CC1P=0（高电平有效），CC1NP=1（低电平有效），OC1M=110（PWM 模式 1）：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟
- 输出信号 OCxN 与参考信号相反，只是其上升沿相对参考下降沿存在延迟

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为 CC1P=0（高电平有效）时，CC1NP=1（低电平有效），带死区的输出信号与参考信号 OCxREF 之间的关系：

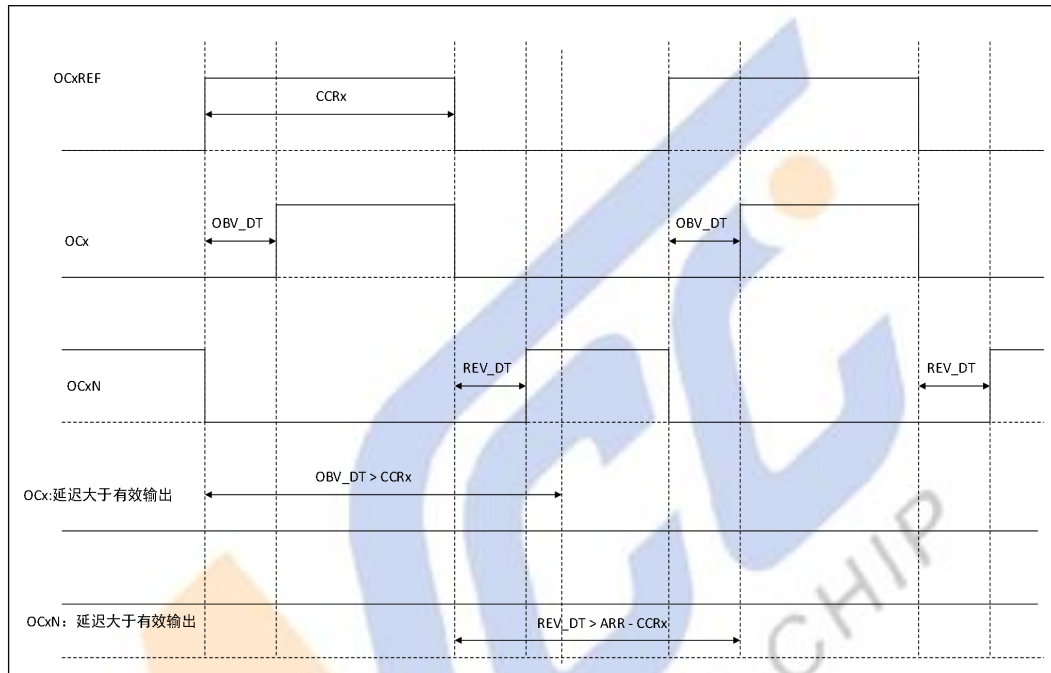


图 14-4 (CC1P=0,CC1NP=1)

如果 DT_EN=1, CC1P=1（低电平有效），CC1NP=0（高电平有效），OC1M=110（PWM 模式 1）：

- 输出信号 OCx 与参考信号相反，只是其下降沿相对参考上升沿存在延迟
- 输出信号 OCxN 与参考信号相同，只是其下降沿相对参考下降沿存在延迟

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为 CC1P=1（低电平有效）时，CC1NP=0（高电平有效），带死区的输出信号与参考信号 OCxREF 之间的关系：

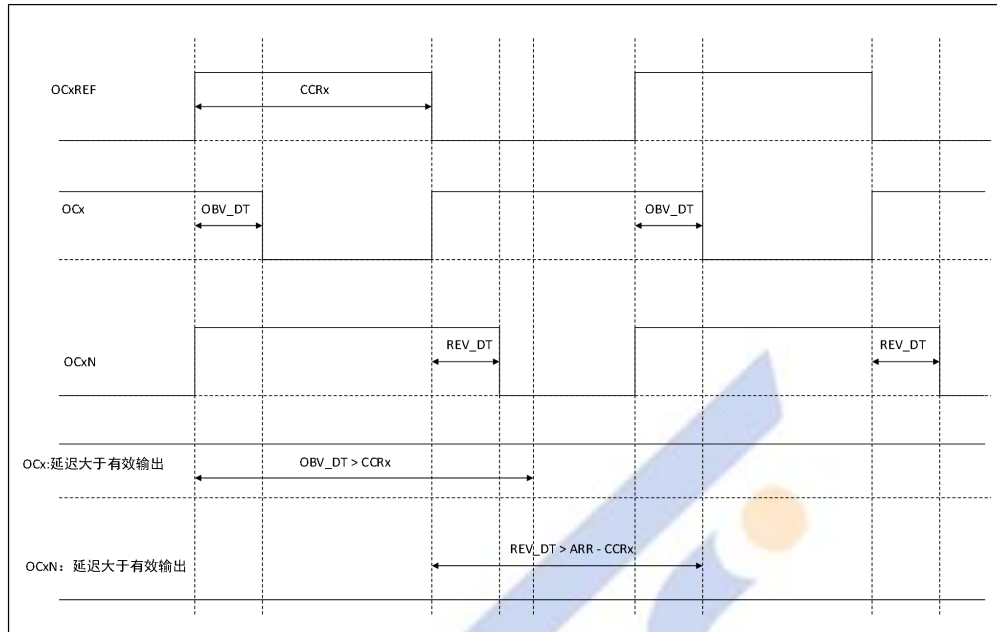


图 14-5 (CC1P=1,CC1NP=0)

14.3.8. 触发单脉冲模式 1

单脉冲模式 1: 第一个选择的触发源会按单脉冲模式启动计数器, 再次出现的触发输入会重新初始化计数器, 生成一个寄存器更新事件并按单脉冲模式启动计数器, 计数器停止后 CEN 不清 0。当定时器已启动时, 到达的触发事件将复位并重新启动计数器, 这将延长一次单脉冲计数的时间。

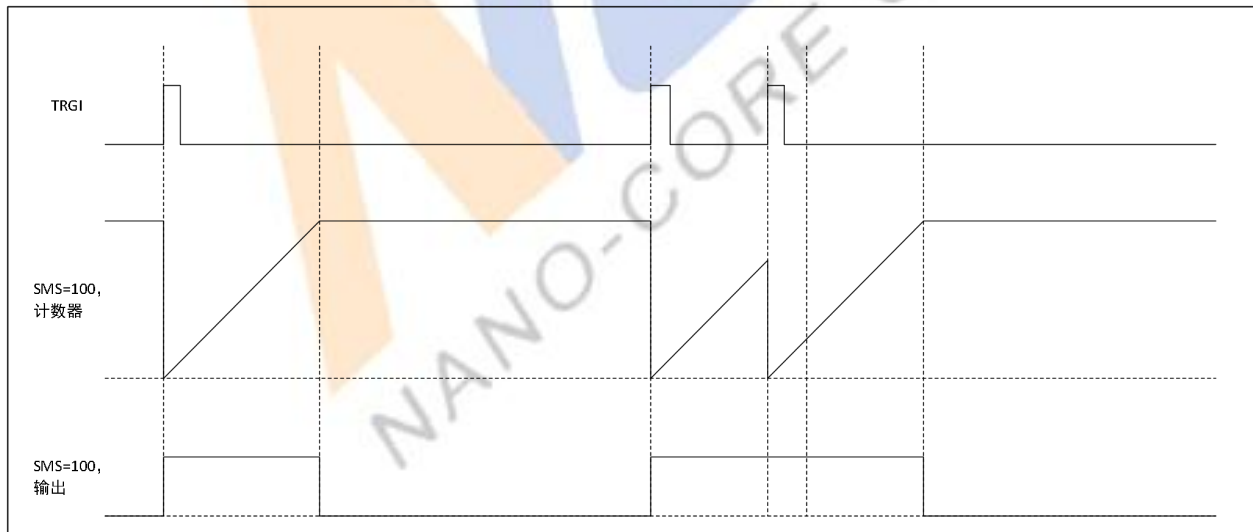


图 14-6 触发单脉冲模式 1 示意图

14.3.9. 触发单脉冲模式 2

单脉冲模式 2: 第一个选择的触发源会按单脉冲模式启动计数器, 再次出现的触发输入不会重新初始化计数器, 生成一个寄存器更新事件并按单脉冲模式启动计数器, 计数器停止后 CEN 不清 0。

定时器已启动时到达的触发事件将被忽略，那么每一次单脉冲计数时间都相同。

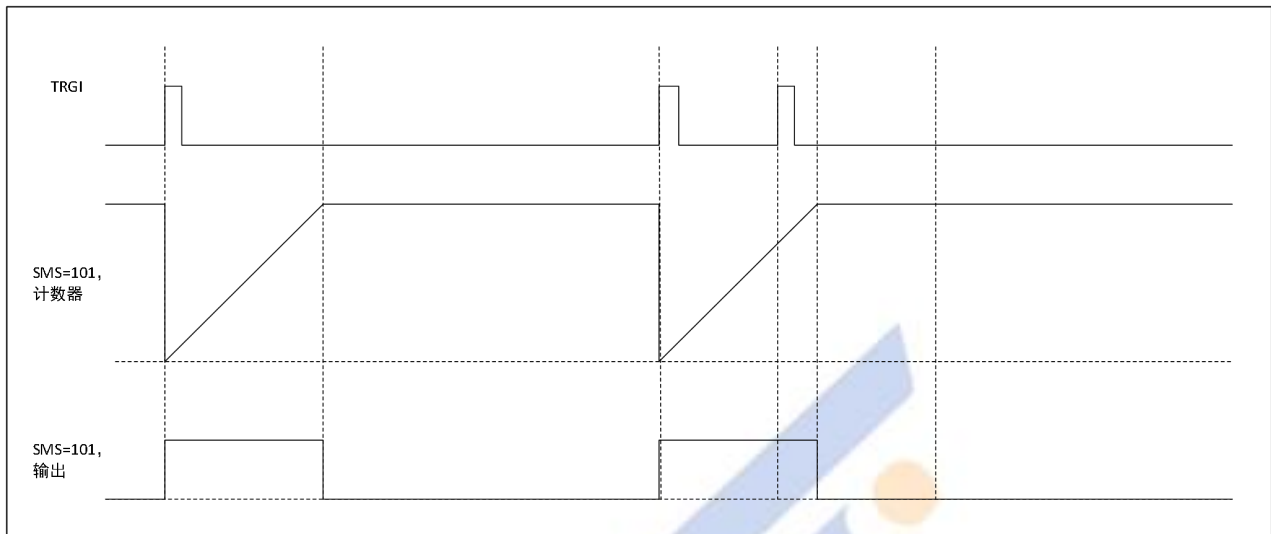


图 14-7 触发单脉冲模式 2 示意图

14.3.10. 中断及异常说明

下表给出了通用定时器的中断请求列表。

表 14-1 表通用定时器中断请求

中断事件	事件标志	事件标志/ 中断清除方法	中断使能控制位
更新中断 (TIMx)	TIMx_SR1 UIF标志位	该位在发生更新事件时 由硬件置1，需通过软件写1 清0	UIE
触发中断(TIMx)	TIMx_SR1 TIF标志位	该位在发生触发事件时 由硬件置1，需通过软件写1 清0	TIE
捕获比较中断(TIMx)	TIMx_SR1 CCnIF标志位 n:为发生中断的 通道	该位在发生捕获比较事 件时由硬件置1,通过软件写1 或者读取CCR1寄存器将该 位清零	CCnIE n:为使能的通道

注意：如果 CCnIF 已经置位 ($CCnIF = 1$)，再次产生捕获比较中断时 TIMx_SR2 的重复捕获比较中断标志位 CCnOF 将会置 1。通过软件写 1 可以清除 CCnOF 标志位。

14.4. GTIM 寄存器

14.4.1. GTIM 寄存器映射

表 14-2 GTIM 寄存器映射

偏移	寄存器名	访问	复位值
0x00	GTIM 版本寄存器 (VER)	r	0x0000_0008
0x04	GTIM 控制寄存器 1 (CR1)	rw	0x0000_0000
0x08	GTIM 控制寄存器 2 (CR2)	rw	0x0000_0000
0x0C	GTIM 从模式控制寄存器 (SMCR)	rw	0x0000_0000
0x10	中断使能寄存器 (IER)	rw	0x0000_0000
0x14	状态寄存器 1 (SR1)	rc_wl	0x0000_0000
0x18	状态寄存器 2 (SR2)	rc_wl	0x0000_0000
0x1C	事件生成寄存器 (EGR)	r	0x0000_0000
0x20	计数器 (CNT)	r	0x0000_0000
0x24	预分频器 (PSC)	rw	0x0000_0000
0x28	自动重载寄存器 (ARR)	rw	0x0000_0000
0x40	捕获/比较模式寄存器 1 (CCMR1)	rw	0x0000_0000
0x44	捕获/比较使能寄存器 1 (CCER1)	rw	0x0000_0000
0x48	捕获/比较寄存器 1 (CCR1)	rw	0x0000_0000
0x4C	TIMx 死区寄存器 1 (DTR1)	rw	0x0000_0000
0x60	捕获/比较模式寄存器 2 (CCMR2)	rw	0x0000_0000
0x64	捕获/比较使能寄存器 2 (CCER2)	rw	0x0000_0000
0x68	捕获/比较寄存器 2 (CCR2)	rw	0x0000_0000
0x6C	TIMx 死区寄存器 2 (DTR2)	rw	0x0000_0000
0x80	捕获/比较模式寄存器 3 (CCMR3)	rw	0x0000_0000
0x84	捕获/比较使能寄存器 3 (CCER3)	rw	0x0000_0000
0x88	捕获/比较寄存器 3 (CCR3)	rw	0x0000_0000
0x8C	TIMx 死区寄存器 3 (DTR3)	rw	0x0000_0000
0xA0	捕获/比较模式寄存器 4 (CCMR4)	rw	0x0000_0000
0xA4	捕获/比较使能寄存器 4 (CCER4)	rw	0x0000_0000
0xA8	捕获/比较寄存器 4 (CCR4)	rw	0x0000_0000
0xAC	TIMx 死区寄存器 4 (DTR4)	rw	0x0000_0000

14.4.2. GTIM 版本寄存器(VER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MVER								SVER							
r								r							
位域	功能														
31-16	保留														
15-8 MVER	主版本号														
7-0 SVER	子版本号 0x08														

14.4.3. GTIM 控制寄存器 (CR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	CKD	ARPE	DIR	OWM	URS	UDIS	CEN
-	-	-	-	-	-	-	-	-	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-8	保留
7-6 CKD	时钟分频 (Clock division) 此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIX) 之间的分频比, 及死区时间所用的时钟 tDTS 和定时器时钟 (CK_INT)间的分频比 00: $t_{DTS} = t_{CK_INT}$ 01: $t_{DTS} = 2 \times t_{CK_INT}$ 10: $t_{DTS} = 4 \times t_{CK_INT}$ 11: 保留
5 APRE	自动重载预装载使能 (Auto-reload preload enable) 0: TIMx_ARR 寄存器不进行缓冲 1: TIMx_ARR 寄存器进行缓冲, 该寄存器在每次发生更新事件时进行重载。
4 DIR	方向 (Direction) 0: 计数器递增计数 1: 计数器递减计数
3 OWM	一次模式 (One-wave mode) 0: 计数器在发生更新事件时不会停止计数 1: 计数器在发生下一更新事件时停止计数 (将 CEN 位清零)
2	更新请求源 (Update request source)

位域	功能
URS 0: 使能后，所有以下事件都会生成更新中断： 1) 递增模式时的计数器上溢(上溢在 CNT = ARR) 2) 递减模式时的计数器下溢(下溢在 CNT = 0) 3) 将 UG 位置 1 1: 使能后，前 2 种模式会生成更新中断，将 UG 位置 1 不生成中断	
1 UDIS 更新禁止 (Update disable) 此位由软件置 1 和清零，用以使能/ 禁止更新事件 (UEV) 生成。 0: 使能 UEV。然后更新影子寄存器的值。如果 UG 位置 1，则会重新初始化计数器和预分频器。 1: 禁止 UEV。不会生成 UEV，各影子寄存器的值 (ARR、CCR _x) 保持不变。UG 位置 1 无效。	
0 CEN 计数器使能 (Counter enable) 0: 禁止计数器 1: 使能计数器 在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。	

14.4.4. GTIM 控制寄存器 2 (CR2)

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

位域	功能
31-8	保留
7 CLKSEL 定时器时钟选择 (clock selection) 0: 选择 PCLK 作为系统时钟 1: 保留	
6-4 CHSEL 当 MMS=100，比较脉冲模式或者 MMS=101，比较模式时，通道选择： 000: 通道 1 001: 通道 2 010: 通道 3 011: 通道 4 100: 保留 101: 保留 110: 保留 111: 保留	
3 CLKEN 定时器时钟使能 (clock enable) 0: 禁止定时器时钟 1: 打开定时器时钟。 在配置 CEN=1 前，必须将 CLKEN 配置为 1，否则计数器时钟将无法正常工作。	
2-0 MMS 主模式选择 (Master mode selection) 这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下： 000: 不输出。TRGO 保持低电平。	

001: 复位——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。
010: 使能——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。
011: 更新——选择更新事件作为触发输出 (TRGO)。
100: 比较脉冲——一旦某通道发生输入捕获或比较匹配事件，触发输出都会发送一个正脉冲。(TRGO)
101: 比较——某通道的比较输出参考信号 OCxREF 用作触发输出 (TRGO)
110, 111: 保留

14.4.5. GTIM 从模式控制寄存器 (SMCR)

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

位域	功能
31-8	保留
7-4 TS	触发信号 (TRGI) 输入选择 (Trigger selection) 此位域可选择将要用于同步计数器的触发输入。 0000: 不经过同步逻辑的内部触发，上升沿有效 0001: 经过同步逻辑的内部触发，上升沿有效 0010: 保留 0011: 滤波后的定时器输入 1 (ch1fp)，触发极性由 CC1NP/CC1P 控制 0100: 滤波后的定时器输入 2 0101: 滤波后的定时器输入 3 0110: 滤波后的定时器输入 4 0111: 保留 1000: 保留 1001: 保留 1010: 保留 1011,1100,1101,1110,1111: 保留
3 TRGO_SYNC	TRGO 同步使能 (TRGO synchronize enable) 0: 不执行任何操作，当前定时器与其从定时器至少有 2 个定时器时钟的延迟（若从定时器 TS=0000，则有 2 个时钟的延迟；若从定时器 TS=0001，则有 4 个时钟的延迟） 1: 当前定时器的使能或复位动作被延迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）的情况。 <i>注意：若要实现完美同步，其从定时器应选择不经过同步逻辑的内部触发。</i>
2-0 SMS	从模式选择 (Slave mode selection) 000: 禁止从模式 001: 复位模式——在出现所选触发输入的有效边沿时，重新初始化计数器并生成一个寄存器更新事件

位域	功能
	010: 门控模式——触发输入(TRGI) 为有效电平时使能计数器时钟。只要触发输入变为无效电平, 计数器立即停止计数(但不复位)。计数器的启动和停止都是受控的 011: 触发连续模式——触发信号 TRGI 出现有效边沿时启动计数器, 计数器连续计数 注意: 如果触发模式启动并且 OWM 为 1, 那么启动计数器之后只会计数一个周期。 100: 触发单脉冲模式 1 101: 触发单脉冲模式 2 110: 保留 111: 保留 注: 建议在 CEN 为 0 时配置从模式选择, 在 CEN=1 时配置可能无法实现正确的功能。

14.4.6. GTIM 中断使能寄存器 (IER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	CC4IE	CC3IE	CC2IE	CC1IE	-	-	TIE	UIE
-	-	-	-	-	-	-	-	rw	rw	rw	rw	-	-	rw	rw

位域	功能
31-8	保留
7 CC4IE	捕获/ 比较 4 中断使能 (Capture/Compare 4 interrupt enable) 0: 禁止 CC4 中断。 1: 使能 CC4 中断。
6 CC3IE	捕获/ 比较 3 中断使能 (Capture/Compare 3 interrupt enable) 0: 禁止 CC3 中断。 1: 使能 CC3 中断。
5 CC2IE	捕获/ 比较 2 中断使能 (Capture/Compare 2 interrupt enable) 0: 禁止 CC2 中断。 1: 使能 CC2 中断。
4 CC1IE	捕获/ 比较 1 中断使能 (Capture/Compare 1 interrupt enable) 0: 禁止 CC1 中断。 1: 使能 CC1 中断。
3-2	保留
1 TIE	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断。 1: 使能触发中断。
0 UIE	更新中断使能 (Update interrupt enable) 0: 禁止更新中断。 1: 使能更新中断。

14.4.7. GTIM 状态寄存器 1 (SR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	CC4IF	CC3IF	CC2IF	CC1IF	-	-	TIF	UIF
-	-	-	-	-	-	-	-	rc_wl	rc_wl	rc_wl	rc_wl	-	-	rc_wl	rc_wl

位域	功能
31-8	保留
7 CC4IF	捕获/比较 4 中断标志 (Capture/compare 4 interrupt flag) 如果通道 CC4 配置为输出： 当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件写 1 清零。 0：不匹配。 1：TIMx_CNT 计数器的值与 TIMx_CCR4 寄存器的值匹配。当 TIMx_CCR4 的值大于 TIMx_ARR 的值时，CC4IF 位将在计数器发生上溢时变为高电平。 如果通道 CC4 配置为输入： 此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR4 寄存器将该位清零。 0：未发生输入捕获事件。 1：TIMx_CCR4 寄存器中已捕获到计数器值。
6 CC3IF	捕获/比较 3 中断标志 (Capture/compare 3 interrupt flag) 如果通道 CC3 配置为输出： 当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件写 1 清零。 0：不匹配。 1：TIMx_CNT 计数器的值与 TIMx_CCR3 寄存器的值匹配。当 TIMx_CCR3 的值大于 TIMx_ARR 的值时，CC3IF 位将在计数器发生上溢时变为高电平。 如果通道 CC3 配置为输入： 此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR3 寄存器将该位清零。 0：未发生输入捕获事件。 1：TIMx_CCR3 寄存器中已捕获到计数器值。
5 CC2IF	捕获/比较 2 中断标志 (Capture/compare 2 interrupt flag) 如果通道 CC2 配置为输出： 当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件写 1 清零。 0：不匹配。 1：TIMx_CNT 计数器的值与 TIMx_CCR2 寄存器的值匹配。当 TIMx_CCR2 的值大于 TIMx_ARR 的值时，CC2IF 位将在计数器发生上溢时变为高电平。 如果通道 CC2 配置为输入： 此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR2 寄存器将该位清零。 0：未发生输入捕获事件。 1：TIMx_CCR2 寄存器中已捕获到计数器值。

位域	功能
4 CC1IF	捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag) 如果通道 CC1 配置为输出： 当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件写 1 清零。 0：不匹配。 1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。 如果通道 CC1 配置为输入： 此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。 0：未发生输入捕获事件。 1：TIMx_CCR1 寄存器中已捕获到计数器值。
3-2	保留
1 TIF	触发中断标志 (Trigger interrupt flag) 该标志在发生触发事件时将由硬件置 1。但需要通过软件写 1 清零。 0：未发生触发事件。 1：触发中断挂起。该位在以下情况下由硬件置 1： - 门控模式时，在计数器启动或停止时置 1 - 在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1 - 软件配置 TIMx_EGR 的 TG 位
0 UIF	更新中断标志 (Update interrupt flag) 该位在发生更新事件时通过硬件置 1。但需要通过软件写 1 清零。 0：未发生更新。 1：更新中断挂起。该位在以下情况下由硬件置 1： - 上溢且当 TIMx_CR1 寄存器中 UDIS = “0” 时。 - 当由于 TIMx_CR1 寄存器中 URS = “0” 且 UDIS = “0” 而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。 - 复位模式时，所选触发信号 TRGI 出现有效边沿时置 1，该触发会重新初始化计数器 - 触发单脉冲模式 1 时，所选触发信号 TRGI 出现有效边沿时，并且计数器处于运行状态时置 1，该触发会重新初始化计数器

14.4.8. GTIM 状态寄存器 2 (SR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	CC4OF	CC3OF	CC2OF	CC1OF
-	-	-	-	-	-	-	-	-	-	-	-	rc_wl	rc_wl	rc_wl	rc_wl

位域	功能
31-4	保留
3 CC4OF	捕获/ 比较 1 重复捕获标志 (Capture/Compare 4 overcapture flag) 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入 1 可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR4 寄存器中已捕获到计数器值且 CC4IF 标志已置 1。
2 CC3OF	捕获/ 比较 1 重复捕获标志 (Capture/Compare 3 overcapture flag) 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入 1 可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR3 寄存器中已捕获到计数器值且 CC3IF 标志已置 1。
1 CC2OF	捕获/ 比较 2 重复捕获标志 (Capture/Compare 2 overcapture flag) 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入 1 可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR2 寄存器中已捕获到计数器值且 CC2IF 标志已置 1。
0 CC1OF	捕获/ 比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag) 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入 1 可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

14.4.9. GTIM 事件生成寄存器 (EGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	CC4G	CC3G	CC2G	CC1G	-	-	TG	UG
-	-	-	-	-	-	-	-	w	w	w	w	-	-	w	w

位域	功能
----	----

位域	功能
31-8	保留
7 CC4G	捕获/比较 4 生成 (Capture/Compare 4 generation) 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: 通道 4 上生成捕获/比较事件: 如果通道 CC4 配置为输出: 使能后，CC4IF 标志置 1 并发送相应中断。 如果通道 CC4 配置为输入: TIMx_CCR4 寄存器中将捕获到计数器的当前值。使能后，CC4IF 标志置 1 并发送相应中断。如果 CC4IF 标志已为高电平，CC4OF 标志将置 1。
6 CC3G	捕获/比较 3 生成 (Capture/Compare 3 generation) 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: 通道 3 上生成捕获/比较事件: 如果通道 CC3 配置为输出: 使能后，CC3IF 标志置 1 并发送相应中断。 如果通道 CC3 配置为输入: TIMx_CCR3 寄存器中将捕获到计数器的当前值。使能后，CC3IF 标志置 1 并发送相应中断。如果 CC3IF 标志已为高电平，CC3OF 标志将置 1。
5 CC2G	捕获/比较 2 生成 (Capture/Compare 2 generation) 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: 通道 2 上生成捕获/比较事件: 如果通道 CC2 配置为输出: 使能后，CC2IF 标志置 1 并发送相应中断。 如果通道 CC2 配置为输入: TIMx_CCR2 寄存器中将捕获到计数器的当前值。使能后，CC2IF 标志置 1 并发送相应中断。如果 CC2IF 标志已为高电平，CC2OF 标志将置 1。
4 CC1G	捕获/比较 1 生成 (Capture/Compare 1 generation) 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: 通道 1 上生成捕获/比较事件: 如果通道 CC1 配置为输出: 使能后，CC1IF 标志置 1 并发送相应中断。 如果通道 CC1 配置为输入: TIMx_CCR1 寄存器中将捕获到计数器的当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。
3-2	保留
1 TG	触发生成 (Trigger generation) 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断。若没有真正的触发源，计数器不启动。
0	更新生成 (Update generation)

位域	功能
UG	该位可通过软件置 1，并由硬件自动清零。 0：不执行任何操作 1：重新初始化计数器并生成寄存器更新事件。预分频器计数器也将清零，但预分频比不受影响。计数器清零，使能后可发生相关中断。

14.4.10. GTIM 计数器（CNT）

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

位域	功能
31-16	保留
15-0 CNT	计数器值

14.4.11. GTIM 预分频器（PSC）

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

位域	功能
31-16	保留
15-0 CNT	预分频器值 (Prescaler value) 计数器时钟频率 $CK_CNT = f_{CK_CNT} / (PSC[15:0] + 1)$。 PSC 代表实际预分频寄存器的值。只有 CEN 为 0 时可以写入。

14.4.12. GTIM 自动重载寄存器（ARR）

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

位域	功能
31-16	保留
15-0 ARR	自动重载值 (Auto-reload value) ARR 为要装载到实际自动重载寄存器的值。 当自动重载值为空时，计数器不工作。

14.4.13. GTIM 捕获比较模式寄存器[复用]（CCMRn）

该寄存器可用于输入捕获模式或输出比较模式，
用于输出比较模式时：

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

位域	功能
31-7	保留
6-4 OCnM	输出比较模式选择(Output Compare mode) 这些位定义提供 OCn 和 OCnN 的输出参考信号 OCnREF 的行为。OCnREF 为高电平有效，而 OCn 和 OCnN 的有效电平则分别取决于 CCnP 位和 CCnNP 位。 000：冻结——输出比较寄存器 TIMx_CCRn 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响，该模式用于生成以 2*TIMx_ARR 为周期的时基。 001：将通道设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCRn) 匹配时，OCnREF 信号强制变为高电平。 010：将通道设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCRn) 匹配时，OCnREF 信号强制变为低电平。 011：翻转——TIMx_CNT=TIMx_CCRn 时，OCnREF 发生翻转 100：强制变为无效电平——OCnREF 强制变为低电平 101：强制变为有效电平——OCnREF 强制变为高电平 110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCRn，通道 1 便为有效状态，否则

位域	功能
	为无效状态。在递减计数模式下，只要 $TIMx_CNT > TIMx_CCRn$，通道 1 便为无效状态（$OCnREF = 0$），否则为有效状态（$OCnREF = 1$） 111: PWM 模式 2——在递增计数模式下，只要 $TIMx_CNT < TIMx_CCRn$，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 $TIMx_CNT > TIMx_CCRn$，通道 1 便为有效状态，否则为无效状态。
3 OCnPE	输出比较预装载使能 (Output Compare preload enable) 0: 禁止与 $TIMx_CCRn$ 相关的预装载寄存器。可随时向 $TIMx_CCRn$ 写入数据，写入后将立即使用新值。 1: 使能与 $TIMx_CCRn$ 相关的预装载寄存器。可读/写访问预装载寄存器。$TIMx_CCRn$ 预装载值在每次生成更新事件时都会装载到活动寄存器中 <i>注意: 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 ($TIMx_CR1$ 寄存器中的 OWM 位置 1)。其它情况下则无法保证该行为。</i>
2 OCnFE	输出比较快速使能 (Output Compare fast enable) 此位用于加快触发输入事件对 CC 输出的影响。 0: 即使触发开启，CC 也将根据计数器和 $CCRn$ 值正常工作。 1: 触发输入上出现有效边沿相当于 CCn 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。仅当通道配置为 PWM1 或 PWM2 模式时，OCnFE 才会起作用。
1-0 CCnS	捕获/比较选择 (Capture/Compare 1 Selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CCn 通道关闭 01: CCn 通道配置为输出 10: CCn 通道配置为输入，ICn 映射到 TI_n 上 11: CCn 通道配置为输入，ICn 映射到 TXOR 上 $TXOR = TI \wedge T2 \wedge T3$, TI,T2,T3 异或之后产生的信号 <i>注意 1: 仅当通道关闭时 ($TIMx_CCER$ 中的 $CCnE = 0$)，才可向 CCnS 位写入数据。</i> <i>注意 2: 对于 CC1S,CC2S,CC3S,CC4S 来说, $TXOR = T1 \wedge T2 \wedge T3$; 对于 CC5S,CC6S,CC7S,CC8S 来说, $TXOR = T5 \wedge T6 \wedge T7$;</i>

用于输入捕获模式时:

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

位域	功能
31-8	保留
7-4 ICnF	输入捕获滤波器 (Input capture filter) 此位域可定义 TI_n 输入的采样频率和适用于 TI_n 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿： 0000: 无滤波器，按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 0100: $f_{SAMPLING} = f_{CK_INT}/2$, N=6

位域	功能
	0101: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/2$, $N=8$ 0110: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/4$, $N=6$ 0111: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/4$, $N=8$ 1000: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/8$, $N=6$ 1001: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/8$, $N=8$ 1010: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/16$, $N=5$ 1011: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/16$, $N=6$ 1100: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/16$, $N=8$ 1101: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/32$, $N=5$ 1110: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/32$, $N=6$ 1111: $f_{\text{SAMPING}} = f_{\text{CK_INT}}/32$, $N=8$
3-2 ICNPSC	输入捕获预分频器 (Input capture prescaler) 此位域定义 CCn 输入(ICn) 的预分频比。 00: 无预分频器, 捕获输入上每检测到一个有效边沿便执行捕获 01: 每发生 2 个事件边沿便执行一次捕获 10: 每发生 4 个事件边沿便执行一次捕获 11: 每发生 8 个事件边沿便执行一次捕获
1-0 CCnS	捕获/比较选择 (Capture/Compare Selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CCn 通道关闭 01: CCn 通道配置为输出 10: CCn 通道配置为输入, ICn 映射到输入通道 TIn 上 11: 保留

14.4.14. GTIM 捕获比较使能寄存器 (CCERn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-							DT_EN	CCnNP	CCnNE	CCnP	CCnE
-	-	-	-	-							rw	rw	rw	rw	rw

位域	功能
31-5	保留
4 DT_EN	插入死区使能 (dead_time enable) 0: 禁止插入死区 1: 允许插入死区 <i>注意: 只有 TIMx_CCMR 的[6:4]OCnM=110,111 的时候允许插入死区有效。</i>
3 CCnNP	捕获/比较互补输出极性 (Capture/Compare complementary output Polarity) CCn 通道配置为输入: CCnNP 与 CCnP 配合使用可定义 TI1FPn/TI2FPn 的极性 (请参见 CCnP 说明)。 CCnN 通道配置为输出: 0: OCnN 高电平有效。

位域	功能
	1: OCnN 低电平有效。
2 CCnNE	捕获/ 比较互补输出使能 (Capture/Compare complementary output enable)。 CCn 通道配置为输出: 0: 关闭——OCnN 未激活。 1: 开启—— 在相应输出引脚上输出 OCnN 信号。
1 CCnNP	捕获/比较输出极性 (Capture/Compare output Polarity)。 CCn 通道配置为输出: 0: OCn 高电平有效。 1: OCn 低电平有效。 CCn 通道配置为输入: CCnNP/CCnN 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。 00: 未反相/上升沿触发 电路对 TIxFPn 上升沿敏感 (在复位模式、触发模式下执行捕获或触发操作), TIxFPn 未反相。 01: 反相/下降沿触发 电路对 TIxFPn 下降沿敏感 (在复位模式、触发模式下执行捕获或触发操作), TIxFPn 反相。 10: 保留, 不使用此配置。 11: 未反相/上升沿和下降沿均触发 注: 电路对 TIxFPn 上升沿和下降沿都敏感 (在复位模式、触发模式下执行捕获或触发操作), TIxFPn 未反相 (在门控模式下执行触发操作)。
0 CCnE	捕获/比较输出使能 (Capture/Compare output enable)。 CCn 通道配置为输出: 0: 关闭——OCn 未激活。 1: 开启——在相应输出引脚上输出 OCn 信号。 CCn 通道配置为输入: 此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 (TIMx_CCRn) 中。 0: 禁止捕获。 1: 使能

14.4.15. GTIM 捕获比较寄存器 (CCRn)

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

位域	功能
31-16	保留
15-0 CCRn	捕获/比较值 (Capture/Compare value) 如果通道 CCn 配置为输出:

位域	功能
	CCRN 是捕获/比较寄存器 n 的预装载值。 如果没有通过 TIMx_CCMRn 寄存器中的 OCnPE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OCn 输出上发出信号的值。 如果通道 CCn 配置为输入： CCRN 为上一个输入捕获事件 (ICn) 发生时的计数器值。

14.4.16. TIMx 死区寄存器 (DTRn)

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

位域	功能
31-16	保留
15-8 REV_DT	反面死区值 (reverse dead-time)，只在 DT_EN 为 0 时配置有效。t_{DTS} 该值定义插入到互补输出之间的下降沿死区持续时间。 REV_DT[7:5]=0xx，死区时间=REV_DT[7:0] * t_{DTS}。 REV_DT[7:5]=10x，死区时间=(64+REV_DT[5:0]) * t_{DTS} * 2。 REV_DT[7:5]=110，死区时间=(32+REV_DT[4:0]) * t_{DTS} * 8。 REV_DT[7:5]=111，死区时间=(32+REV_DT[4:0]) * t_{DTS} * 16。 例如：如果 t_{DTS}=125ns (8MHz)，则可能的死区值为： 0 到 15875ns (步长为 125ns) 16us 到 31750ns (步长为 250ns) 32us 到 63us (步长为 1us) 64us 到 126us (步长为 2us) 死区误差最大为±CK_INT，死区时间最小为 1。
7-0 OBV_DT	正面死区值 (obverse dead-time)，只在 DT_EN 为 0 时配置有效。 该值定义插入到互补输出之间的上升沿死区持续时间。 OBV_DT[7:5]=0xx，死区时间=OBV_DT[7:0] * t_{DTS}。 OBV_DT[7:5]=10x，死区时间=(64+OBV_DT[5:0]) * t_{DTS} * 2。 OBV_DT[7:5]=110，死区时间=(32+OBV_DT[4:0]) * t_{DTS} * 8。 OBV_DT[7:5]=111，死区时间=(32+OBV_DT[4:0]) * t_{DTS} * 16。 例如：如果 t_{DTS}=125ns (8MHz)，则可能的死区值为： 0 到 15875ns (步长为 125ns) 16us 到 31750ns (步长为 250ns) 32us 到 63us (步长为 1us) 64us 到 126us (步长为 2us) 死区误差最大为±CK_INT，死区时间最小为 1。



15. 模数转换器(ADC)

15.1. 简介

12 位 ADC 可测量来自 28 个外部源和 2 个内部源的信号。各通道的 A/D 转换可以配置成单次、连续、触发模式。

15.2. 主要特性

- 转换速度高达 500Ksps
- ADC 转换时钟可配，分频后不超过 8M
- 支持最多 28 个外部输入模拟通道以及 2 个内部电压通道
- 支持内部参考及外部参考
- ADC 采样时钟可配置
- 支持硬件触发和软件触发
- 支持延迟触发
- 支持单通道单次转换和连续转换
- 支持转换结果自动匹配比较

触发到实际发生转换时的时间间隔。如果 CTL 寄存器中的连续转换 CONT 位置 1，在首次外部触发发生后，ADC 将开始连续转换。

当 ADC 正在转换时，发生硬件触发或者软件触发，这些触发都会被忽略。

15.3.4. 单次转换(CONT=0)

当 CTL 寄存器中的 CONT=0 时，选择此模式。可通过以下方式开始转换：

- 将 CTL 寄存器中的 ADGO 位置 1
- 硬件触发事件

每次转换完成后：

- 转换后的数据会存储在 DATA 寄存器中
- EOCIF(转换结束)标志置 1
- EOCIE 位置 1 时将产生中断

随后，ADC 会停止工作，直至发生新的外部触发事件或 ADGO 位再次置 1。

15.3.5. 连续转换(CONT=1)

ADC 在受到内部触发或者外部触发时，开始第一次转换过程，在当前转化结束后，ADC 等待 CFG 寄存器中设置的 TRGDL 位所对应的延迟后自动开启下一次转换。当 CTL 寄存器中的 CONT=1 时，可选择此模式。可通过以下方式开始转换：

- 将 CTL 寄存器中的 ADGO 位置 1
- 硬件触发事件

每次转换完成后：

- 转换后的数据会存储在 DATA 寄存器中
- EOCIF(转换结束)标志置 1
- EOCIE 位置 1 时将产生中断

退出连续转换模式只需将 EN 位拉低，在当前 ADC 转换结束后自动停止连续转换。

15.3.6. ADC 开始转换

ADC 可以通过以下方式开始转换：

- 在 CFG 寄存器的 TRGS 位为 0 时，将 CTL 寄存器中的 ADGO 位置 1 会开始转换（软件触发）
- 在 CFG 寄存器的 TRGS 位为 1 时，可根据 SYSCFG 模块的 CFGR2 寄存器里的 ADC_TSEL 来选择外部触发源，每个触发源可以通过 CFG 寄存器中的 EDGE 位选择上升沿触发还是下降沿触发

ADGO 位还可用于指示当前是否正在进行 ADC 操作。当 ADC 正在转换时 ADGO=1。

15.3.7. 比较输出

可通过配置 CTL 寄存器中的 REF_CMP 来选择是上限比较、下限比较或是区间比较；

然后写比较上限寄存器 REFH 和比较下限寄存器 REFL；

当满足比较条件，且 REFIE 位置 1 时将产生中断，REFIF 将置起。

15.4. ADC 中断

表 15-1 ADC 中断表

中断事件	事件标志	使能控制位
ADC 阈值比较中断	REFIF	REFIE
ADC 转换结束中断	EOCIF	EOCIE

15.5. ADC 寄存器

15.5.1. ADC 寄存器映射

表 15-2 ADC 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	ADC 版本寄存器(VER)	r	0x0000_0005
0x04	ADC 控制寄存器(CTL)	rw	0x0000_00E0
0x08	ADC 数据寄存器(DATA)	r	0x0000_0000
0x0C	ADC 中断使能寄存器(IER)	rw	0x0000_0000
0x10	ADC 中断和状态寄存器(ISR)	r	0x0000_0000
0x14	ADC 中断和状态清零寄存器(ICR)	w	0x0000_0000
0x18	ADC 配置寄存器(CFG)	rw	0x0000_0000
0x1C	ADC 高阈值参考数据寄存器(REFH)	rw	0x0000_0000
0x24	ADC 低阈值参考数据寄存器(REFL)	rw	0x0000_0000

15.5.2. ADC 版本寄存器(VER)

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

位域	功能
31-16	保留
15-0 VER	IP 版本号

15.5.3. ADC 控制寄存器(CTL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VREF_SEL		-			ECH_TY PE_SEL	ICH_SEL		-		REF_CMP		-			
rw		-			rw	rw		-		rw		-			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP				ECH_SEL				CLK_SEL			CONT	VBUF_SEL		EN	ADGO
rw				rw				rw			rw	rw		rw	rw

位域	功能
31-30 VREF_SEL	ADC 参考电压选择 0: AVDD 5V 1: 外部参考电压 2: 内部 BUF 电压（内部 BUF 使能打开） 3: AVDD 5V
29-27	保留
26 ECH_TYPE_SE L	ADC 外部输入通道高段选择 0: 选择低段，通道 0 到通道 15 1: 选择高段，通道 16 到通道 27
25-24 ICH_SEL	ADC 内部通道选择 00: 外部输入通道（由 ECH_SEL 选择） 01: 1/2 AVDD 10: 内部 BUF 电压（内部 BUF 使能打开） 11: 保留
23-22	保留
21-20 REF_CMP	ADC 参考阈值比较模式 00: 禁止阈值比较 01: ADC 数据低于低阈值数据 10: ADC 数据在低阈值及高阈值数据之间 11: ADC 数据高于高阈值数据
19-16	保留
15-12 SAMP	ADC 采样周期控制 0x0 时采样周期为 1 个周期，0xF 时采样周期为 16 个周期
11-8 ECH_SEL	ADC 外部输入通道选择 0000: 通道 0 或通道 16 0001: 通道 1 或通道 17 ... 1010: 通道 10 或通道 26 1011: 通道 11 或通道 27 1100: 通道 12 1101: 通道 13 1110: 通道 14 1111: 通道 15
7-5 CLK_SEL	ADC 时钟选择位 000: 系统时钟 2 分频 001: 系统时钟 4 分频 010: 系统时钟 6 分频 011: 系统时钟 8 分频 100: 系统时钟 12 分频 101: 系统时钟 16 分频 110: 系统时钟 20 分频 111: 系统时钟 32 分频
4 CONT	ADC 连续转换控制位 0: ADC 连续转换禁止 1: ADC 连续转换使能
3-2 VBUF_SEL	ADC 内部 BUF 电压选择 00: 保留

位域	功能
	01: 选择 1.8V 作为内部 BUF 电压 10: 选择 2.5V 作为内部 BUF 电压 11: 选择 3.0V 作为内部 BUF 电压
1 EN	ADC 使能控制位 1: ADC 使能 0: ADC 禁止
0 ADGO	写操作时作为 ADGO 使用, 软件触发 ADC 转换, 读时表示 ADC BUSY 备注: 只有 EN=1 时写 ADGO 有效, 否则写 ADGO 无效。

15.5.4. ADC 数据寄存器(DATA)

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

位域	功能
31-12	保留
11-0 DATA	DATA 寄存器

15.5.5. ADC 中断使能寄存器(IER)

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

位域	功能
31-2	保留
1 REFIE	ADC 阈值比较中断使能
0 EOCIE	ADC 转换结束中断使能

15.5.6. ADC 中断和状态寄存器(ISR)

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

位域	功能
31-2	保留
1 REFIF	ADC 转换值处于阈值比较模式范围中断标志
0 EOCIF	转换结束标志位

15.5.7. ADC 中断和状态清零寄存器(ICR)

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

位域	功能
31-2	保留
1 REFIC	ADC 阈值比较中断标志清除，写 1 清零
0 EOCIC	ADC 转换结束中断标志清除，写 1 清零

15.5.8. ADC 配置寄存器(CFG)

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

位域	功能
31-5	保留
4 TRGS	ADC 触发源选择 0: 选择内部触发源(软件触发) 1: 选择外部触发源（硬件触发），外部触发源由 SYSCFG 选择
3 EDGE	ADC 外部触发沿选择： 0: 上升沿触发 1: 下降沿触发
2-0 TRGDL	外部触发延迟时间/连续转换延迟时间 000: 1 个 ADC_MCLK 时钟延迟 001: 2 个 ADC_MCLK 时钟延迟 010: 4 个 ADC_MCLK 时钟延迟 011: 8 个 ADC_MCLK 时钟延迟 100: 12 个 ADC_MCLK 时钟延迟 101: 16 个 ADC_MCLK 时钟延迟 110: 24 个 ADC_MCLK 时钟延迟 111: 32 个 ADC_MCLK 时钟延迟

15.5.9. ADC 高閾值參考數據寄存器(REFH)

[illegible]

位域	功能
31-12	保留
11-0 REFH	ADC 比较上限值

15.5.10. ADC 低阈值参考数据寄存器(REFL)

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

位域	功能
31-12	保留
11-0 REFL	ADC 比较下限值

16. 循环冗余校验计算单元 (CRC)

16.1. 简介

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位/16 位/32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

16.2. 主要特性

- 支持 CRC-32-IEEE: 802.3
- 支持 CRC-8: 8210
- 支持 CRC-16-CCIT
- 支持 CRC-16-IBM
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- AHB 总线
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期(HCLK)内完成

16.3. 功能说明

16.3.1. CRC 框图

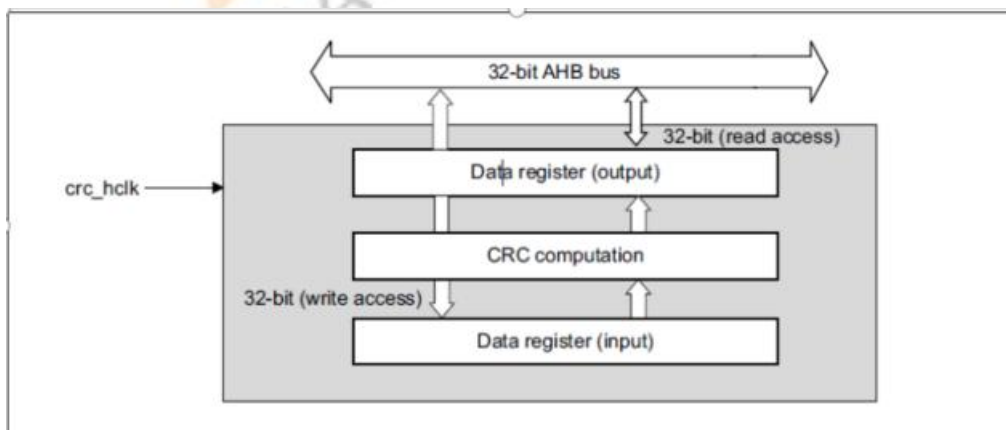


图 16-1 CRC 框图

16.3.2. CRC 操作

16.3.2.1.CRC 输入输出

CRC 计算单元具有单个 32 位读/写数据寄存器(DR)。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其它寄存器，只允许进行 32 位访问。

计算时间取决于输入数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期

可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

16.3.2.2.CRC 数据翻转

输入数据的顺序可反转，以管理各种数据存放方式（双字/单字/字节，大端/小端等等）。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CR[REV_IN]位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

- 按字节执行位反转的 0x58D43CB2
- 按半字执行位反转的 0xD458B23C
- 按全字执行位反转的 0xB23CD458

通过将 CR[REV_OUT]置 1 也可以将输出数据反转。该操作按位进行：

- 对于 CRC8，读取数据时，将输出数据的低 8 bit 按位反转
- 对于 CRC16，读取数据时，将输出数据的低 16 bit 按位反转
- 对于 CRC32，读取数据时，将输出数据 32 位按位反转，例如，输出数据 0x11223344 将转换为 0x22CC4488。

16.3.2.3.CRC 初始值编程

使用 CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 INIT 寄存器对 CRC 初始值进行编程。

16.3.2.4.CRC 操作步骤

- 1) 配置 CR 寄存器，确定 CRC 多项式，输入输出顺序反转等参数。
- 2) 写 INIT 寄存器编程 CRC 初始值。
- 3) 向 CR[RESET]位写 1，将 CRC 计算器初始化为 INIT 中的值。
- 4) 向 DR 中写入要计算的数值，可按字、右对齐半字和右对齐字节进行访问。
- 5) 写入的数据从高位到低位进行计算，例如，对于 CRC8，如果写入的数据为 0x11223344，则 CRC8 计算相当于对于 0x11、0x22、0x33、0x44 依次计算。
- 6) 等待相应的计算时间，可以写入接下来要计算的数值或者读 DR 寄存器获得最终的 CRC 结果。

16.4. CRC 寄存器

16.4.1. CRC 寄存器总表

表格 16-1 CRC 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	CRC 版本寄存器(IP_VERSION)	r	0xA368_0008
0x04	CRC 数据寄存器 (DR)	rw	0xFFFF_FFFF
0x08	CRC 控制寄存器 (CR)	rw	0x0000_0000
0x0C	CRC 初始值 (INIT)	rw	0xFFFF_FFFF

16.4.2. CRC 版本寄存器(IP_VERSION)

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

位域	功能
31-16 Vender	版本号 0xA386
15-8 MVER	主版本号 0x00

位域	功能
15-8 SVER	子版本号 0x08

16.4.3. CRC 数据寄存器 (DR)

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

位域	功能
31-0 DR	CRC 数据 该寄存器用于向 CRC 计算器写入新数据。 读取寄存器时可读出之前的 CRC 计算结果。 如果数据大小小于 32 位, 则最低有效位可用于写入/读取正确值。

16.4.4. CRC 控制寄存器 (CR)

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

位域	功能
31-8	保留
7 REV_OUT	反转输出数据 (Reverse output data) 该位用于控制输出数据位顺序的反转。 0: 不影响位顺序 1: 位反转输出格式
6-5 REV_IN[1:0]	反转输入数据(Reverse input data) 这些位用于控制输入数据位顺序的反转。 00: 不影响位顺序 01: 按字节执行位反转 10: 按半字执行位反转 11: 按字执行位反转
4-3 POLYSIZE	CRC 多项式选择位 00: CRC-32-IEEE 802.3

位域	功能
	01: CRC-16-CCIT 10: CRC-16-IBM 11: CRC-8: 8210
2-1	保留
0 RESET	RESET 位 此位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。 此位只能置 1，将由硬件自动进行清零。

16.4.5. CRC 初始值 (INIT)

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

位域	功能
31-0 CRC_INIT	可编程 CRC 初始值 (Programmable initial CRC value) 此寄存器用于写入 CRC 初始值

17. 独立看门狗(IWDG)

17.1. 简介

独立看门狗由内部低速时钟 LSI 时钟源或外部低速时钟 LSE 时钟源驱动，因此即便主时钟失效，它仍然能保持工作状态。这非常适合于需要独立工作环境且对时间精度要求不高的场合。

17.2. 主要特性

- 16 位递减计数器
- 当递减计数器值小于 0 时复位（如果看门狗已激活）

17.3. 功能说明

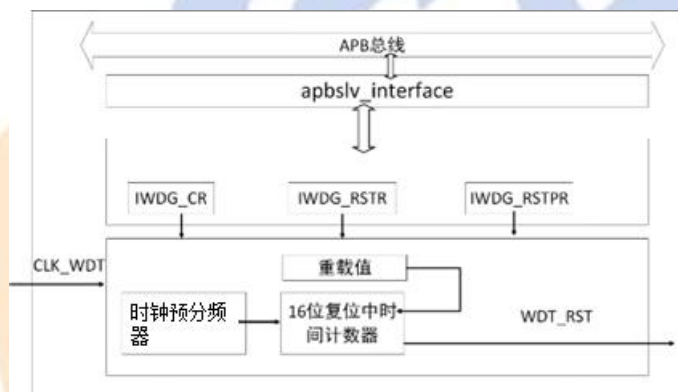


图 17-1 IWDG 框图

独立看门狗包含一个 4 级的预分频器和一个 16 位的向下递减计数器。

当控制寄存器 CR[EN] = 1 时可以开启独立看门狗，计数器开始从复位值 0xFFFF 递减计数。当计数器计数到 0 时会产生一个复位信号。

任何时候向 KR[KR]中写入 0xCAFE 都可以重载计数器，重载值来源于 CR[RSTDAT]，从而避免看门狗复位。

CR 寄存器具有写保护功能（IWDG 使能位 EN 除外），在写 CR 寄存器前，需要将 KR[KR] = 0x5A3C 才能临时解锁一次写操作，写操作后 CR 寄存器会重新锁定。

当器件进入调试模式时（内核停止），IWDG 计数器会根据 DBG 寄存器中的相应位的配置选择继续正常工作或者停止工作。

CFG_INFO0 配置有效后，无需设置 CR[EN]，IWDG 常开。如果 CFG_INFO0 配置无效，需要设置 CR[EN]开启 IWDG。IWDG 一旦开启，则需要发生系统复位才能关闭（需要 CFG_INFO0 的 IWDG 无效）。

由于 IWDG 时钟来源于低速时钟（LSI/LSE），当通过 RCC 模块中的 LSICR[LSION]位关闭 LSI 时钟或 BDCR[LSEON]位关闭 LSE 时，IWDG 将不再工作，因此在低功耗停止模式时，如果保持芯片在休眠期间 IWDG 继续正常工作应保持 LSI/LSE 时钟正常。

注意： IWDG 在 SLEEP 模式下继续计数，正常复位；在 STOP 模式下继续计数，但不会产生复位。



17.4. IWDG 寄存器

17.4.1. IWDG 寄存器映射

表 17-1 IWDG 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	IWDG 版本寄存器 (VER)	r	0xA368_0008
0x04	IWDG 控制寄存器 (CR)	rw	0xFFFF_0000
0x08	IWDG 键值寄存器 (KR)	w	0x0000_0000

17.4.2. IWDG 版本寄存器(VER)

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

位域	功能
31-0	IP 版本号

17.4.3. IWDG 控制寄存器(CR)

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

位域	功能
31-16 RSTDAT	看门狗计数器重载值 这些位受写保护，每次对 KR[KR]中写入 0xCAFE 时，这个值就会重载到看门狗计数器中。
15-6	保留
5-4 RSTTIME	预分频系数选择 分频后的频率 $f_{CLK} = f_{IWDG} / 2^{(RSTTIME * 2)}$ 这些位受写保护。通过写 KR[KR] = 0x5A3C 可解锁一次 00: 1 分频 01: 4 分频 10: 16 分频 11: 64 分频

位域	功能
3-1	保留
0 EN	看门狗使能位 此位由软件置“1”，系统复位时清零

17.4.4. IWDG 键值寄存器(KR)

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

位域	功能
31-16	保留
15-0 KR	只可写,写入不同的值产生不同的功能 0xCAFE 重载计数器 0x5A3C 解锁 CR 寄存器写保护

18. 窗口看门狗(WWDG)

18.1. 简介

窗口看门狗 (WWDG) 通常被用来监测由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非程序在 CR[TDAT] 位变成 0 前刷新递减计数器的值, 否则看门狗电路在达到预置的时间周期时, 会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值, 也会产生 MCU 复位。这意味着必须在限定的时间窗口内刷新计数器。

WWDG 时钟由 APB 时钟经预分频后提供, 通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用程序。

18.2. 主要特性

- 可编程的自由运行递减计数器
- 复位条件
 - 当递减计数器值小于 0x40 时复位 (如果看门狗已激活)
 - 在窗口之外重载递减计数器时复位 (如果看门狗已激活)
- 提前唤醒中断 (EWI): 当递减计数器等于 0x40 时触发 (如果已使能且看门狗已激活)

18.3. 功能说明

如果激活看门狗 (CR[EN] = 1), 则当 7 位递减计数器 CR[TDAT] 从 0x40 递减到 0x3F (CR[TDAT] 已清零) 时会引发复位。当计数器值大于窗口寄存器中所存储的值时, 如果软件重载计数器, 则会产生复位。

应用程序在正常运行过程中必须定期地写入 CR 寄存器以防止 MCU 发生复位。只有当计数器值低于窗口寄存器值且高于 0x3F 时, 才能执行此操作。要存储到 CR 寄存器中的值必须介于 0xFF 和 0xC0 之间。

18.3.1. WWDG 框图

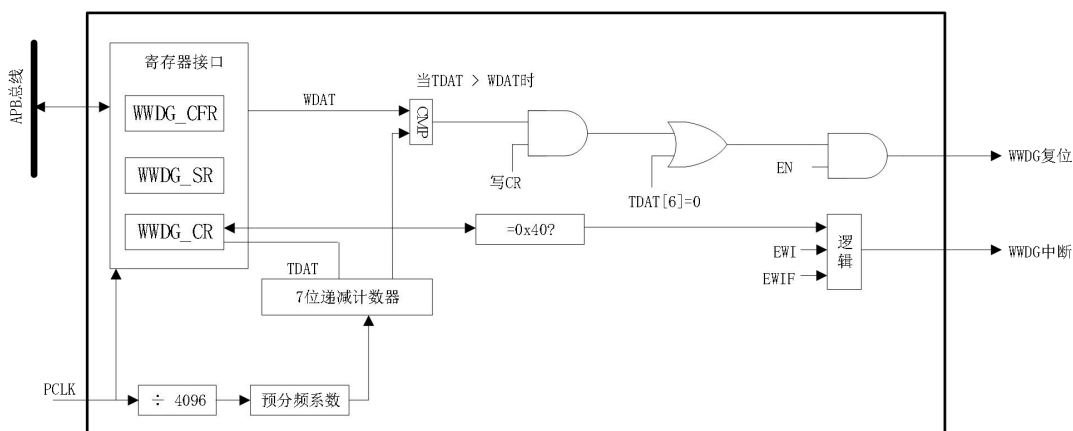


图 18-1 看门狗框图

18.3.2. 使能看门狗

系统复位后，窗口看门狗总是处于关闭状态。可通过 CR[EN]位来开启或者关闭窗口看门狗。

CR[EN]位受写保护，需要连续对 WPR 寄存器写入 0x5A3C、0xCAFE 后解锁保护。

18.3.3. 控制递减计数器

递减计数器在看门狗使能后开始递减计数，当递减至 TDAT[6]为 0 时，会自动载入 0x3F。当看门狗关闭时，递减计数器不持续递减。

当使能看门狗时，必须将 TDAT[6] 位置 1，以防止立即复位。T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频器的状态是未知的。配置寄存器 (WWDG_CFR) 包含窗口的上限：为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x3F 时必须重载。

注： 可使用 TDAT 6 位产生软件复位（将 EN 位置 1 并将 TDAT[6] 位清零）。

18.3.4. 看门狗中断

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断(EWI)。通过设置 WWDG_CFR 寄存器中的 EWIF 位使能 EWIF 中断。当递减计数器的值为 0x40 时，将生成 EWIF 中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWIF 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序 (ISR) 可用来重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

18.3.5. 调试模式

当器件进入调试模式时，WWDG 计数器会根据 DBG 模块中的配置位选择继续正常工作或者停止工作。

18.4. WWDG 中断

提前唤醒中断(EWI): 通过设置 WWDG_CFR 寄存器中的 EWI 位使能 EWI 中断, 当递减计数器的值为 0x40 时, 将生成 EWI 中断。

18.5. WWDG 寄存器

18.5.1. WWDG 寄存器映射

表 18-1 WWDG 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	WWDG 版本寄存器 (VER)	r	0xA368_0008
0x04	WWDG 控制寄存器 (CR)	rw	0x0000_007F
0x08	WWDG 配置寄存器 (CFR)	rw	0x0000_007F
0x0C	WWDG 状态寄存器 (SR)	rc_w1	0x0000_0000
0x10	WWDG 写保护寄存器 (WPR)	w	0x0000_0000

18.5.2. WWDG 版本寄存器(VER)

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

位域	功能
31-0	IP 版本号

18.5.3. WWDG 控制寄存器 (CR)

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

位域	功能
31-8	保留
7 EN	WWDG 激活使能位 0: 禁止看门狗 1: 使能看门狗 此位由软件置 1，只有复位后才由硬件清零。当 EN = 1 时，看门狗开始工作。
6-0 TDAT	7 位计数器 (MSB 到 LSB) 这些位用来存储看门狗计数器的值。它每隔 (4096×2^{WDGSEL}) PCLK 个周期递减一次。当它从 0x40 递减到 0x3F (TDAT[6] = 0) 时会产生复位。

18.5.4. WWDG 配置寄存器 (CFR)

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

位域	功能
31-10	保留
9 EWI	提前唤醒中断 (Early wakeup interrupt) 置 1 后，只要计数器值达到 0x40 就会产生中断。 注：此位只有在 EN=0 时才能进行设置。
8-7 WDGSEL	定时器时基 (Timer base) 可按如下方式修改预分频器的时基： 00: CK 计数器时钟 (PCLK div 4096) 分频器 1 01: CK 计数器时钟 (PCLK div 4096) 分频器 2 10: CK 计数器时钟 (PCLK div 4096) 分频器 4 11: CK 计数器时钟 (PCLK div 4096) 分频器 8 注：此位域只有在 EN=0 时才能进行设置。
6-0 WDAT	7 位窗口值 (7-bit window value) 这些位包含用于与递减计数器进行比较的窗口值。 注：此位只有在 EN=0 时才能进行设置。

18.5.5. WWDG 状态寄存器 (SR)

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

位域	功能
31-1	保留
0 EWIF	提前唤醒中断标志 (Early wakeup interrupt flag) 当计数器值达到 0x40 时此位由硬件置 1。它必须由软件通过写入 1 来清零。写入 0 不起作用。如果不使能中断，此位也会被置 1。

18.5.6. WWDG 写保护寄存器(WPR)

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

位域	功能
31-16	保留
15-0 WPR	写保护 当连续写入 0x5A3C 和 0xCAFE，才允许对 WWDG_CR 寄存器中的 EN 位执行写操作。

19. 内部集成电路 (I2C) 接口

19.1. 简介

I2C（内部集成电路）总线接口处理微控制器与串行 I2C 总线间的通信。它提供多主模式功能，可以控制所有 I2C 总线特定的序列、协议、仲裁和时序。它支持标准模式 (Sm)、快速模式(Fm)和超快速模式(Fm+)。它还与 SMBus（系统管理总线）和 PMBus（电源管理总线）兼容。

19.2. 主要特性

- 兼容 I2C 总线规范第 03 版：
 - 从模式和主模式
 - 多主模式功能
 - 标准速度模式（高达 100 kHz）
 - 快速模式（高达 400 kHz）
 - 超快速模式（高达 1 MHz）
 - 7 位和 10 位寻址模式
 - 多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
 - 所有 7 位地址应答模式
 - 广播呼叫
 - 总线上的数据建立和保持时间可软件配置
 - 方便易用的事件管理
 - 可选的时钟延长
 - 软件复位
 - 可编程模拟和数字噪声滤波器

还可额外提供以下特性，具体取决于产品实现：

- 兼容 SMBus 规范第 3.1 版：
 - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
 - 命令和数据应答控制
 - 支持地址解析协议 (ARP)
 - 支持主机和从设备
 - SMBus 报警
 - 超时和空闲条件检测
- 兼容 PMBus 第 1.1 版标准

19.3. I2C 功能说明

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I2C 总线。它可以连接到标准速度（高达 100 kHz）、

快速(高达 400 kHz)或超快速(高达 1 MHz) I2C 总线。该接口也可通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 SMBus。如果支持 SMBus 功能: 还可使用额外的可选 SMBus 报警引脚 (SMBA)。

19.3.1. I2C 框图

I2C 接口的框图如图所示。

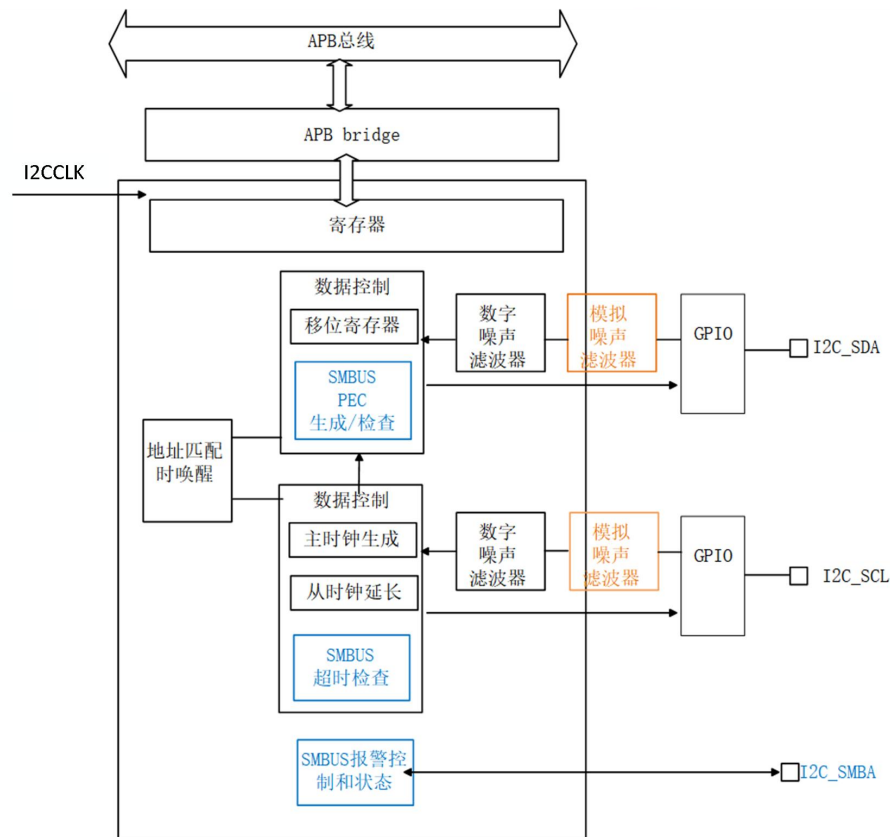


图 19-1 I2C 接口框图

19.3.2. 模式选择

该接口在工作时可选用以下四种模式之一:

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下, 它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式, 并在出现仲裁丢失或生成停止位时从主模式切换为从模式, 从而实现多主模式功能。

19.3.3. 通信流程

在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。保留的 SMBus 地址也可由软件使能。数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下传送。

在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发送器发送一个应答位。请参见下图。

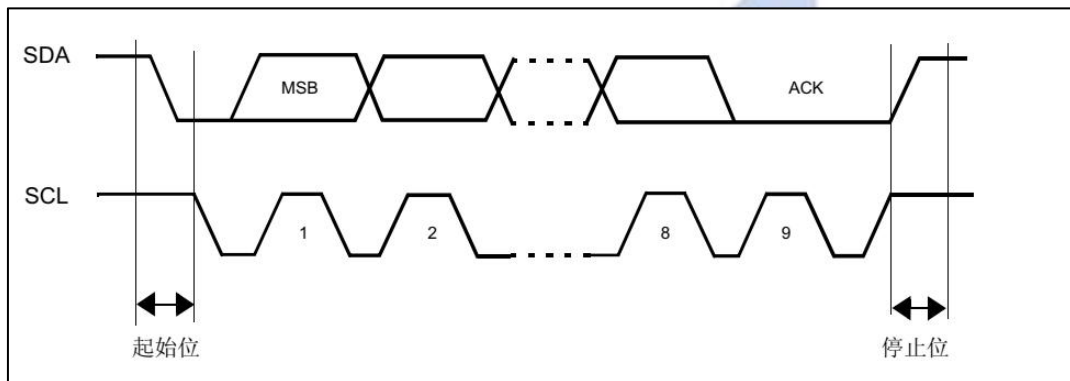


图 19-2 I2C 通信流程

应答位可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

19.3.4. I2C 初始化

使能和禁止外设

I2C 外设时钟必须在时钟控制器中进行配置和使能。然后可通过将 I2C_CR1 寄存器中的 PE 位置 1 使能 I2C。当禁止 I2C (PE=0) 时，I2C 将执行软件复位。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C 外设之前，如有必要，用户必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上集成了模拟噪声滤波器。该模拟滤波器符合 I2C 规范，此规范要求快速模式和超快速模式下对脉宽在 50ns 以下的脉冲都要抑制。用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，通过配置 I2C_CR1 寄存器中的 DNF[3:0] 位来选择数字滤波器。使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 $DNF \times I2CCLK$ 个周期后才会发生内部变化。从而可抑制的尖峰脉宽在 1 到 15 个 I2CCLK 周期可编程。

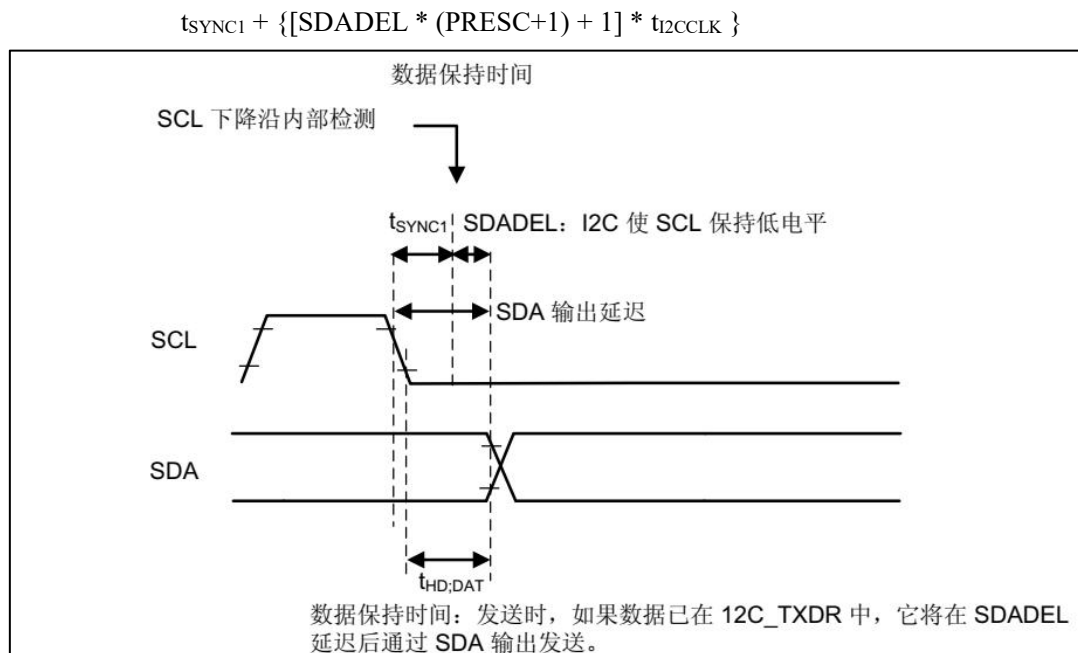
注意：使能 I2C 时不允许更改滤波器配置。

I2C 时序

必须配置时序，以便保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0] 和 SDADEL[3:0] 位。

• 当内部检测到 SCL 下降沿时会在发送 SDA 输出之前插入一段延时。该延时为 $t_{SDADEL} = SDADEL \times t_{PRESC} + t_{I2CCLK}$ ，其中 $t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$ 。tSDADEL 会影响保持时间 tHD;DAT。

SDA 总输出延时为：



t_{SYNC1} 持续时间取决于以下参数:

- SCL 下降斜率
- 模拟滤波器 (使能时) 引入的输入延时: $t_{\text{AF}(\text{min})} < t_{\text{AF}} < t_{\text{AF}(\text{max})}$
- 数字滤波器 (使能时) 引入的输入延时: $t_{\text{DNF}} = \text{DNF} * t_{\text{I2CCLK}}$
- SCL 与 I2CCLK 时钟建立同步而产生的延时 (2 到 3 个 I2CCLK 周期) 为了桥接 SCL 降沿的未定义区域, 用户编程 SDADEL 时必须遵循以下条件:

$$\{t_{\text{f}(\text{max})} + t_{\text{HD,DAT}(\text{min})} - t_{\text{AF}(\text{min})} - [(\text{DNF} + 3) * t_{\text{I2CCLK}}]\} / \{(\text{PRESC} + 1) * t_{\text{I2CCLK}}\} \leq \text{SDADEL}$$

$$\text{SDADEL} \leq \{t_{\text{VD,DAT}(\text{max})} - t_{\text{AF}(\text{max})} - [(\text{DNF} + 4) * t_{\text{I2CCLK}}]\} / \{(\text{PRESC} + 1) * t_{\text{I2CCLK}}\}$$

标准模式、快速模式和超快速模式下的 $t_{\text{HD,DAT}}$ 最大值分别可达 3.45 μs 、0.9 μs 和 0.45 μs , 但必须小于 $t_{\text{VD,DAT}}$ 最大值 (差值为跳变时间)。只有器件未延长 SCL 信号的低电平周期(t_{LOW}) 时, 才必须满足该最大值条件。如果时钟延长 SCL, 数据必须在建立时间内保持有效, 之后才能释放时钟。

SDA 上升沿通常为最坏情况, 因此在这种情况下, 上述公式变成如下形式:

$$\text{SDADEL} \leq \{t_{\text{VD,DAT}(\text{max})} - t_{\text{r}(\text{max})} - 260 \text{ ns} - [(\text{DNF} + 4) * t_{\text{I2CCLK}}]\} / \{(\text{PRESC} + 1) * t_{\text{I2CCLK}}\}。$$

注意: NOSTRETCH=0 时会违反该条件, 这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间, 以保证建立时间。

- 在 t_{SDADEL} 延时后, 或在因数据未写入 I2C_TXDR 寄存器而导致从器件必须延长时钟的情况下发送 SDA 输出后, SCL 线会在建立时间内保持低电平。该建立时间为 $t_{\text{SCLDEL}} = (\text{SCLDEL} + 1) * t_{\text{PRESC}}$, 其中 $t_{\text{PRESC}} = (\text{PRESC} + 1) * t_{\text{I2CCLK}}$ 。 t_{SCLDEL} 会影响建立时间 $t_{\text{SU,DAT}}$ 。

为了桥接 SDA 跳变 (上升沿通常为最坏情况) 的未定义区域, 编程 SCLDEL 时必须遵循以下条件:

$$\{[t_{\text{r}(\text{max})} + t_{\text{SU,DAT}(\text{min})}] / [(\text{PRESC} + 1)] * t_{\text{I2CCLK}}\} - 1 \leq \text{SCLDEL}。$$

如果从模式下 NOSTRETCH=1, 则 SCL 不会延长。因此, 编程 SDADEL 时还必须确保提供充足的建立时间。

表 19-1 I2C-SMBUS 规范数据建立和保持时间

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{\text{HD,DAT}}$	数据保持时间	0	-	0	-	0	-	0.3	-	

$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	μs
$t_{su;DAT}$	数据建立时间	250	-	100		50		250		ns
t_r	SDA 和 SCL 信号的上升时间	-	1000		300	-	120	-	1000	
t_f	SDA 和 SCL 信号的下 降时间	-	300		300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0] 和 SCLL[7:0] 位来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。该延时为

$$t_{SCLL} = (SCLL + 1) \times t_{PRESC}, \text{ 其中 } t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}。$$

t_{SCLL} 会影响 SCL 低电平时间 t_{LOW} 。

- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 $t_{SCLH} = (SCLH + 1) \times t_{PRESC}$ ，其中 $t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$ 。 t_{SCLH} 会影响 SCL 高电平时间 t_{HIGH} 。

更多详细信息，请参见 I2C 主模式初始化。

注意：使能 I2C 后，不允许更改时序配置。此外，还必须在使能从设备前，对 NOSTRETCH 进行配置。更多详细信息，请参见 I2C 从模式初始化。

注意：使能 I2C 后，不允许更改 NOSTRETCH 配置。

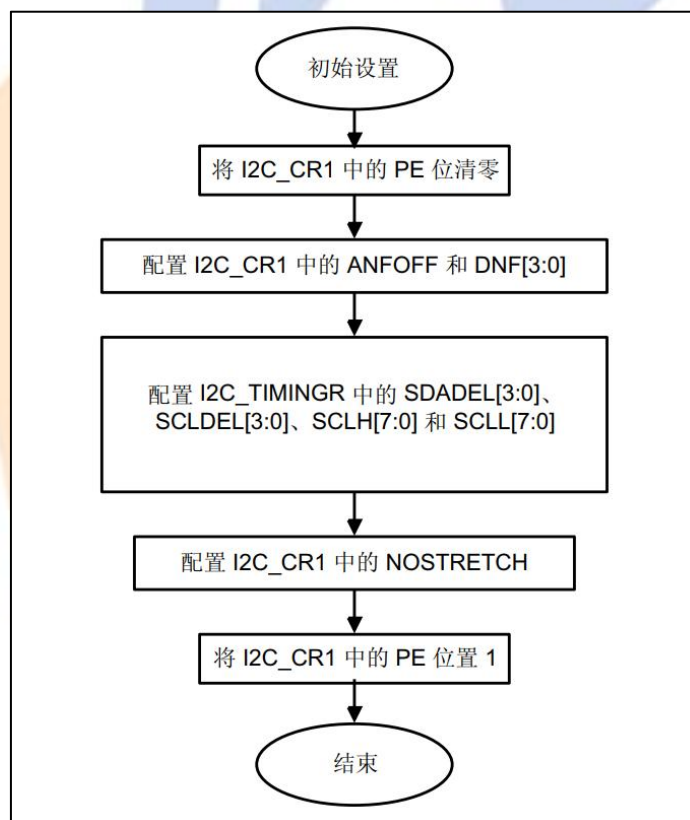


图 19-4 I2C 初始化流程

19.3.5. 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR。

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少 3 个 APB 时钟周期，才能成功执行软件复位。写入以下软件序列可确保这一点：- 写入 PE=0 - 检查 PE=0 - 写入 PE=1。

19.3.6. 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第 8 个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空(RXNE=0)，则移位寄存器的内容会复制到其中。如果 RXNE=1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第 8 个和第 9 个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

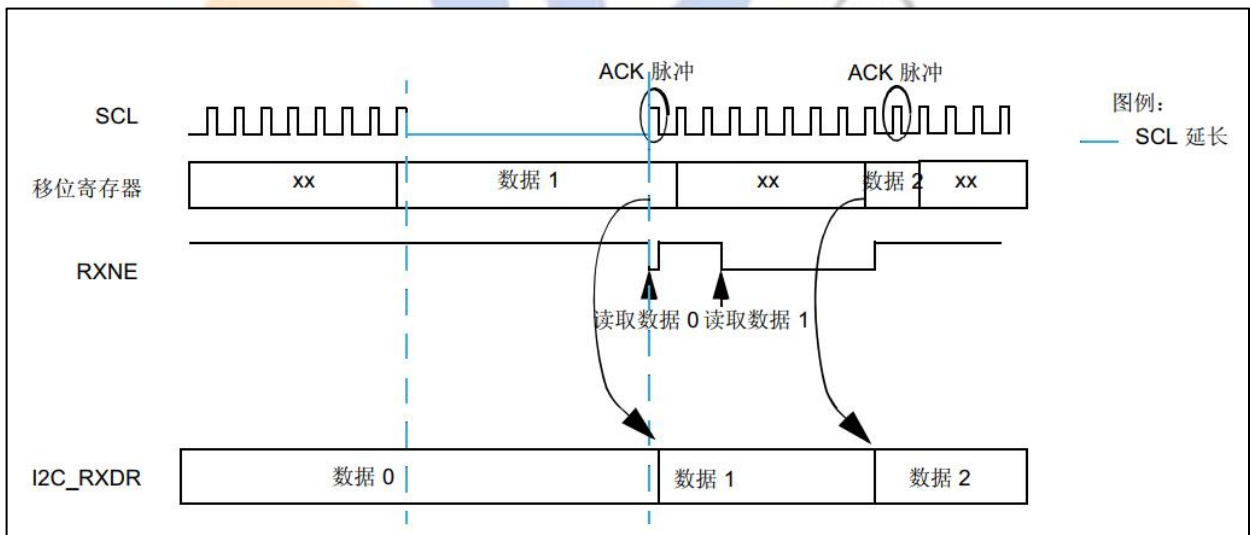


图 19-5 数据接收

发送

如果 I2C_TXDR 寄存器不为空(TXE=0)，则其内容会在第 9 个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。如果 TXE=1（意味 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第 9 个 SCL 脉冲后进行延长。

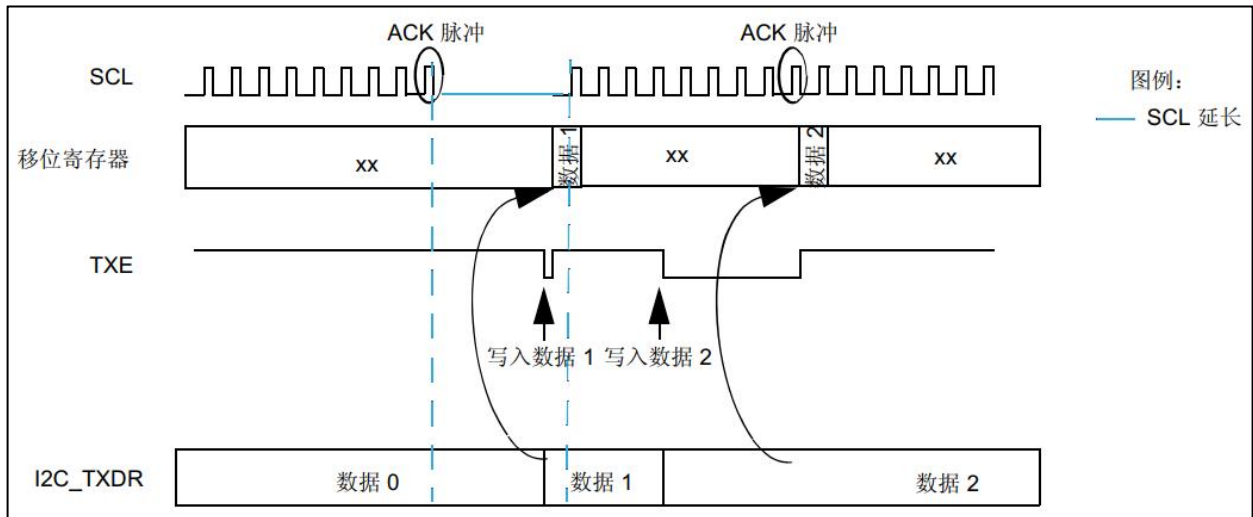


图 19-6 数据发送

硬件传输管理

I2C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 RESTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器通常在主模式下使用。在从模式下，字节计数器默认为禁止状态，但可以通过软件来使能，方法是将 I2C_CR2 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0] 位域中进行编程。如果待传输的字节数(NBYTES)大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。当往 NBYTES 写入一个非零值时，TCR 由软件清零。

在往 NBYTE 中设置最后一次传输的字节数前，必须把 RELOAD 位清零。

当主模式下 RELOAD=0 时，可在以下 2 种模式下使用计数器：

- **自动结束模式**（I2C_CR2 寄存器中的 AUTOEND = “1”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，主器件便会自动发送停止位。
- **软件结束模式**（I2C_CR2 寄存器中的 AUTOEND = “0”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件把 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主器件要发送重复起始位时，必须使用该模式。

注意：当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 19-2 I2C 配置表

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0
从 Tx/Rx 接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

19.3.7. I2C 从模式

I2C 从模式初始化

要在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。
- 如果需要额外的从地址，可配置第 2 个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0] 位置 1 最多可屏蔽 7 个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。只要 OA2MSK 不等于 0，OA2 的地址比较器便会排除 I2C 保留地址（0000 XXX 和 1111 XXX），这些地址将不会得到应答。如 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。OA2 始终为 7 位地址。

如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK=0，则它们可以在通过特定使能位使能后得到应答。通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。

- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDR1E 位置 1 时将生成中断。

默认情况下，从器件使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主器件不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0] 位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式 (NOSTRETCH = 0)

在默认模式下，I2C 从器件会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDR1E 位置 1 以清零 ADDR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零 (TXE=1) 时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从器件字节控制模式和重载模式 (SBC=1 且 RELOAD=1) 下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0] 字段写入一个非零值以将 TCR 清零时，将释放该时钟延展。
- 在 SCL 下降沿检测之后，I2C 会延长 SCL 的低电平时间（不超过 $[(SCLDEL+SCLDEL+1) * (PRESC+1) +$

1] * t_{I2CCLK})。

不带时钟延长的从模式 (NOSTRETCH = 1)

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I2C 从器件不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则应提供 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从器件字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从器件字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式 (RELOAD=1)。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第 8 个和第 9 个脉冲之间的低电平时间。用户可以 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注意： SBC 位只能在 I2C 被禁止时、从器件不被寻址时或 ADDR=1 时配置。

ADDR=1 或 TCR=1 时，可以更改 RELOAD 位的值。

注意： 从器件字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH=1 时将 SBC 位置 1。

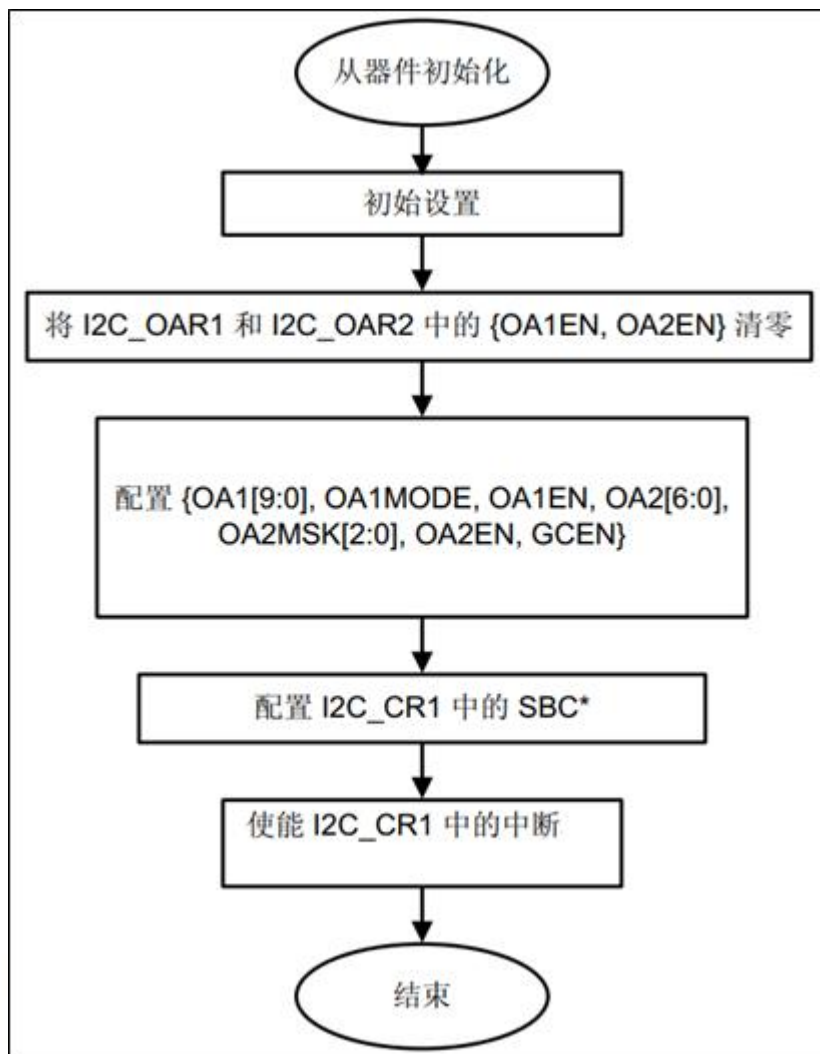


图 19-7 从器件初始化流程

从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态 (TXIS)。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将生成中断。I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从器件自动释放 SCL 和 SDA 线，以使主器件执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从地址 (ADDR=1) 时 TXE = 0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从器件字节控制模式 (SBC=1) 下，必须在地址匹配中断子程序(ADDR=1) 中向 NBYTE 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

注意：如果 NOSTRETCH=1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第

一个数据字节:

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节,可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器,从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位,以确保在地址应答之后,第一次数据传输开始之前执行这些操作。

如果第一次数据传输开始时 STOPF 仍置 1, 则将生成下溢错误 (OVR 标志置 1)。

如果需要 TXIS 事件 (发送中断请求), 用户必须将 TXE 位和 TXIS 位均置 1, 以便生成 TXIS 事件。

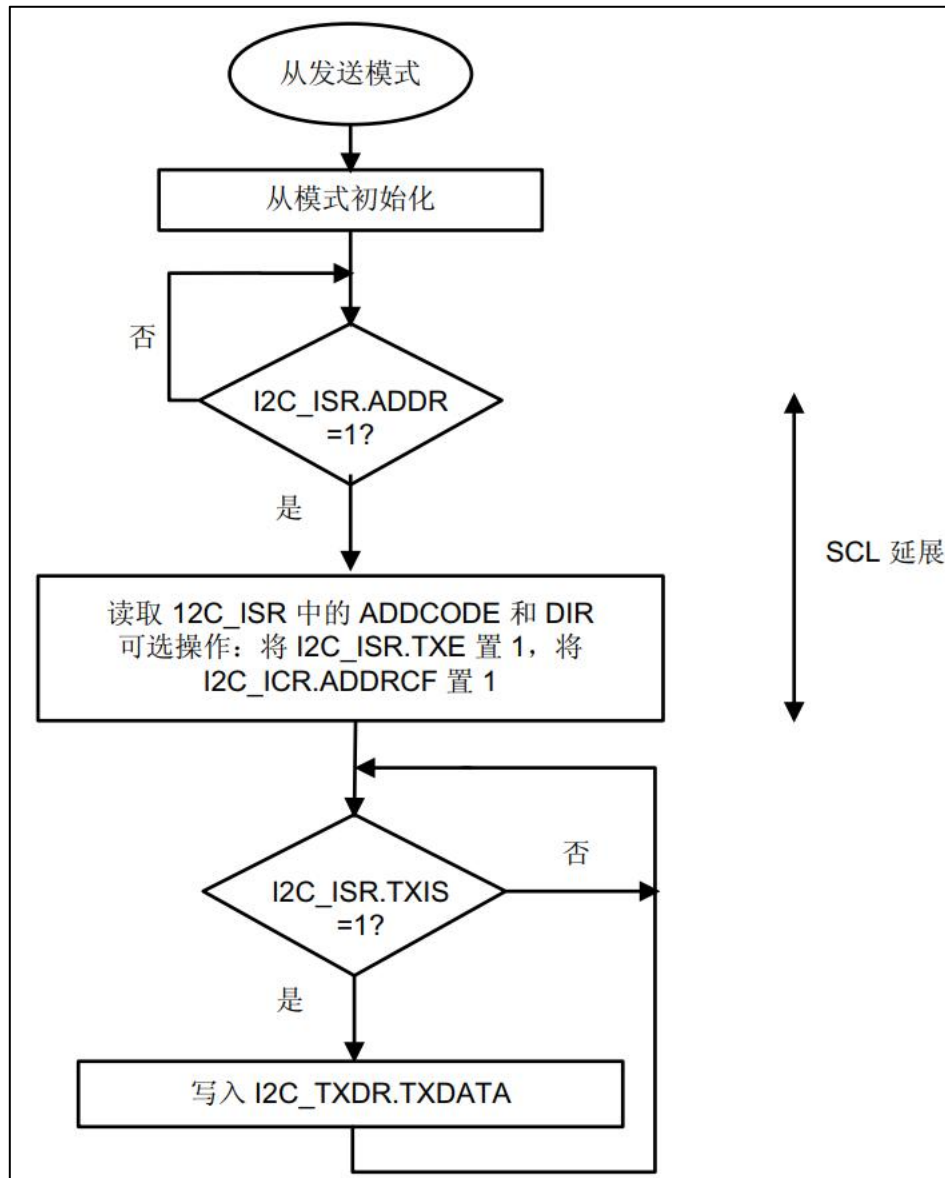
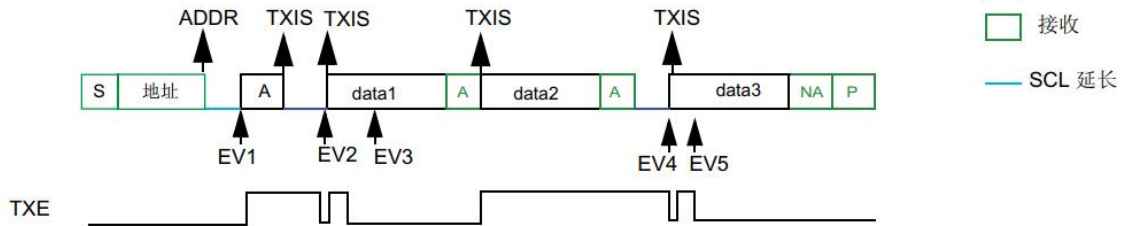


图 19-8 从发送器传输序列流程图(NOSTRETCH=0)

示例：I2C 从器件发送 3 个字节，刷新第 1 个数据，
NOSTRETCH=0:



EV1: ADDR ISR: 检查 ADDCODE 和 DIR, 将 TXE 置 1, 将 ADDRCONF 置 1

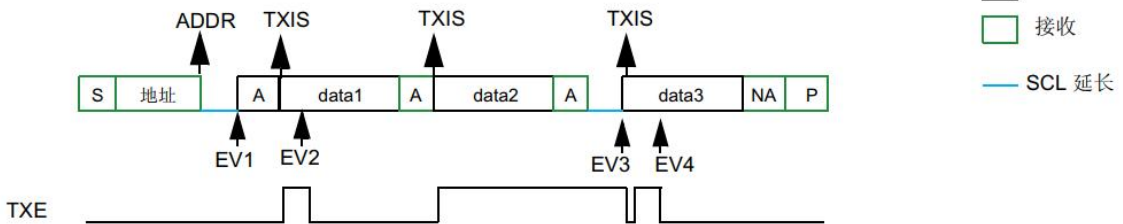
EV2: TXIS ISR: 写入 data1

EV3: TXIS ISR: 写入 data2

EV4: TXIS ISR: 写入 data3

EV5: TXIS ISR: 写入 data4 (不发送)

示例：I2C 从器件发送 3 个字节，不刷新第 1 个数据，
NOSTRETCH=0:



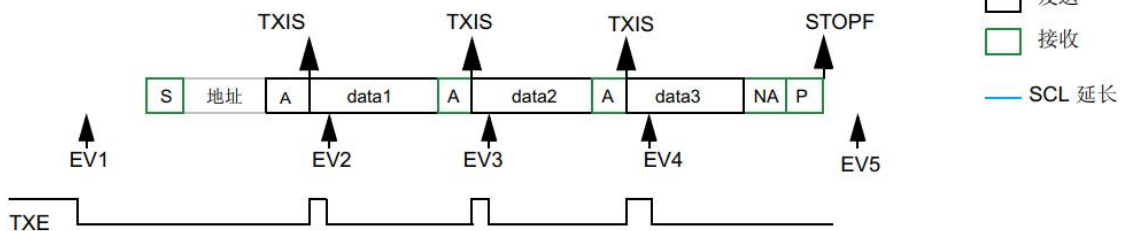
EV1: ADDR ISR: 检查 ADDCODE 和 DIR, 将 ADDRCONF 置 1

EV2: TXIS ISR: 写入 data2

EV3: TXIS ISR: 写入 data3

EV4: TXIS ISR: 写入 data4 (不发送)

示例：I2C 从器件发送 3 个字节，NOSTRETCH=1:



EV1: 写入 data1

EV2: TXIS ISR: 写入 data2

EV3: TXIS ISR: 写入 data3

EV4: TXIS ISR: 写入 data4 (不发送)

EV5: STOPF ISR: (可选操作: 将 TXE 和 TXIS 置 1), 将 STOPCF 置 1

图 19-9 从发送器传输总线图

从接收器

当 I2C_RXDR 满时, I2C_ISR 中的 RXNE 将置 1, 如果 I2C_CR1 中的 RXIE 置 1, 还将生成中断。读取 I2C_RXDR 时, 将清零 RXNE。接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时, I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

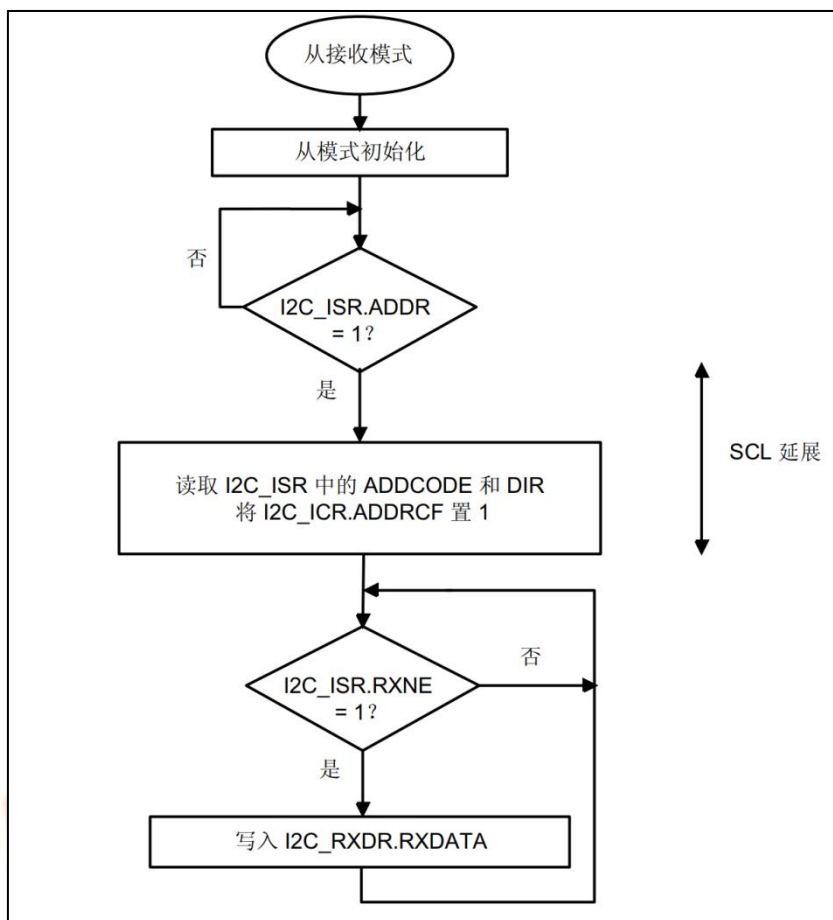


图 19-10 从接收器传输序列流程 (NOSTRETCH=0)

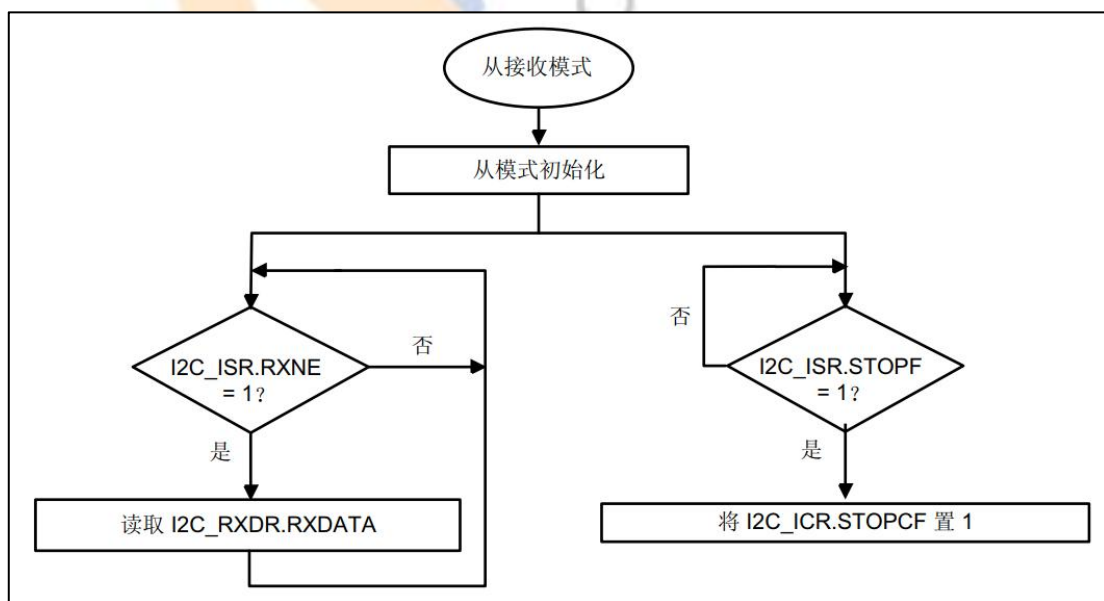


图 19-11 从接收器传输序列流程 (NOSTRETCH=1)

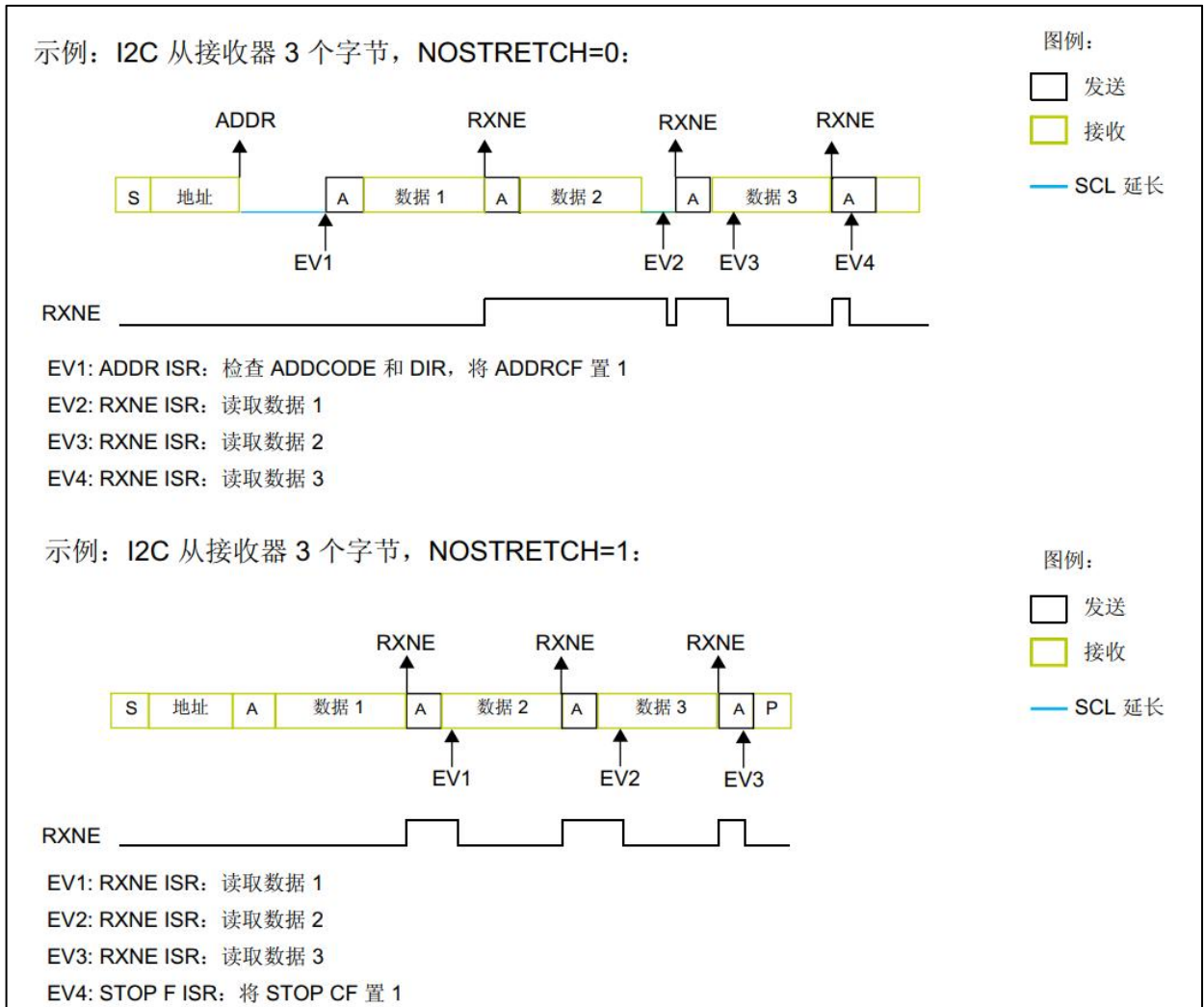


图 19-12 从接收器传输总线图

19.3.8. I2C 主模式

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。为了实现时钟同步，需执行以下操作：

使用 SCLL 计数器从 SCL 低电平内部检测开始对时钟的低电平进行计数。

使用 SCLH 计数器从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器 SCLL[7:0] 位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存 SCLH[7:0] 位中编程的值，I2C 便会使 SCL 变为低电平。因此，主时钟周期为：

$$t_{sc1} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+1) + (SCLL+1)] * (PRESC+1) * t_{2CLK}\}$$

t_{SYNC1} 的持续时间取决于以下参数：

-SCL 下降斜率

- 模拟滤波器（使能时）引入的输入延时。
 - 数字滤波器（使能时）引入的输入延时： $DNF * t_{I2CCLK}$
 - SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）
- t_{SYNC2} 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时。
- 数字滤波器（使能时）引入的输入延时： $DNF * t_{I2CCLK}$

SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

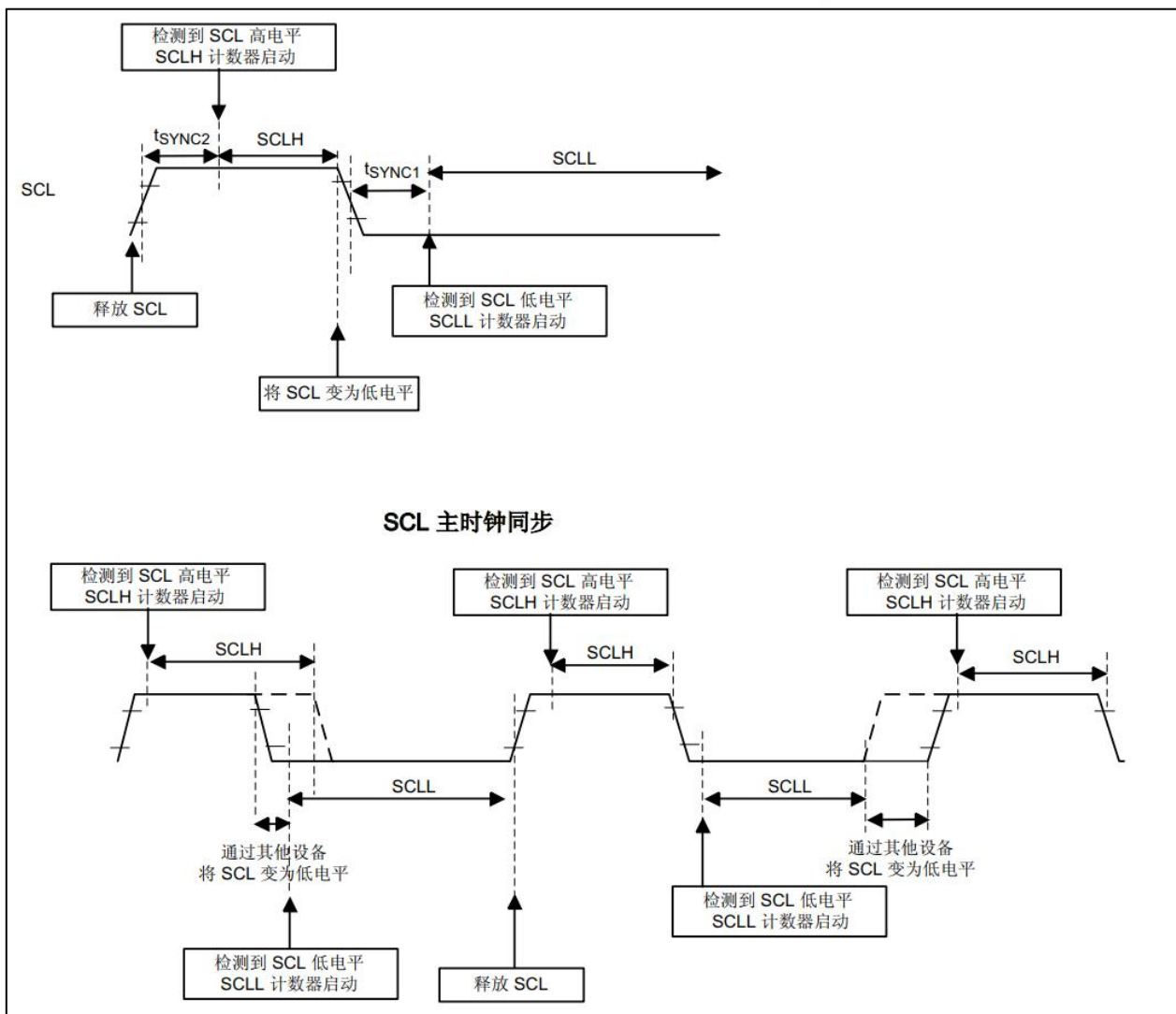


图 19-13 I2C 主时钟生成

注意：为了符合 I2C 或 SMBus 规范，主时钟必须遵循下表中给出的时序：

表 19-3 I2C-SMBUS 规范时钟时序

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小 值	最大 值	最小 值	最大 值	最小 值	最大 值	最小 值	最大 值	
f_{SCL}	SCL 时钟频率		100		400		1000		100	kHz
$t_{HD:STA}$	(重复) 起始条件的保持时间	4.0	-	0.6		0.26	-	4.0	-	μs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6		0.26	-	4.7	-	μs
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6		0.26	-	4.0	-	μs
t_{BUF}	停止条件和起始条件之间的 总线空闲时间	4.7	-	1.3		0.5	-	4.7	-	μs
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3		0.5	-	4.7	-	μs
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6		0.26	-	4.0	50	μs
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300		120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300		120	-	300	ns

注：SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。SCLH 还用于生成 $t_{HD:STA}$ 和 $t_{SU:STO}$ 时序。

主模式通信初始化（地址阶段）

要发起通信，用户必须在 I2C_CR2 寄存器中为寻址的从器件编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址 SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0] 填充为 0xFF。

然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

之后，当主器件检测到总线空闲 (BUSY = 0) 时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从器件地址。

仲裁丢失时，主器件将自动切换回从模式，如果作为从器件被寻址，还可对其自身地址进行应答。

注意：无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。如果当 START 位置 1 时，I2C 作为从器件 (ADDR=1) 被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCONF 位置 1 时清零。

注意：该步骤同样适用于重复起始位。在这种情况下，BUSY=1。

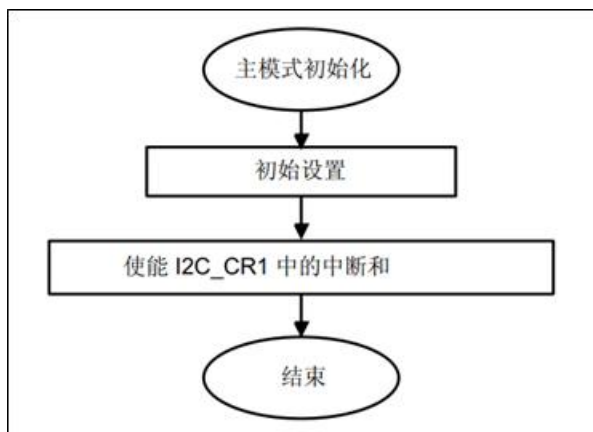


图 19-14 主模式初始化流程图

主接收器寻址 10 位地址从器件的初始化过程

- 如果从地址采用 10 位格式, 用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下, 主器件会在 START 位置 1 后自动发送以下完整序列: (重复) 起始位 + 带写方向的从器件 10 位地址头字节 + 从器件地址第 2 个字节 + 重复起始位 + 带读方向的从器件 10 位地址头字节。



图 19-15 10 位地址读访问(HEADER10R = 0)

- 如果主器件对 10 位地址从器件进行寻址、向该从器件发送数据、然后再从该从器件读取数据，则必须首先完成主器件发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R=1。在这种情况下，主器件发送以下序列：重复起始位 + 从地址 10 位头读取。

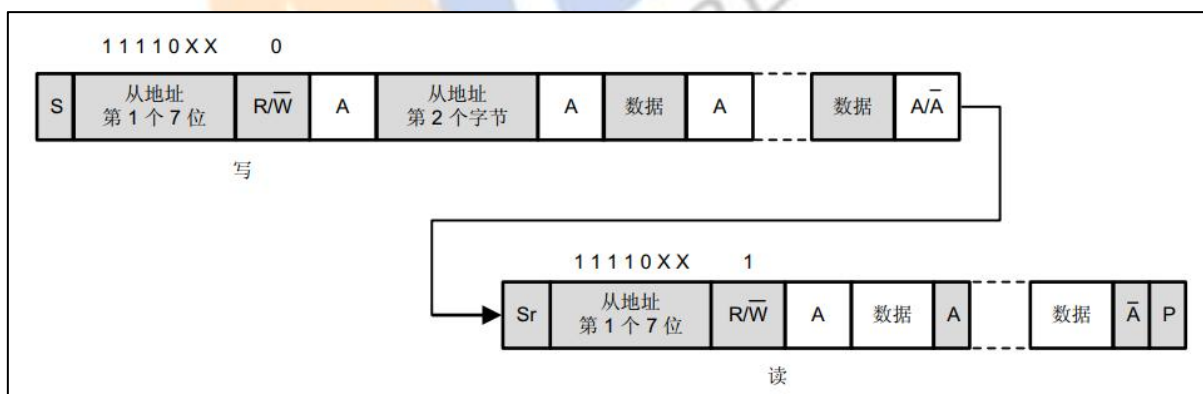


图 19-16 10 位地址读访问 (HEADER10R = 1)

主发送器

写传输时，在发送完每个字节（即第 9 个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1, TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时, 该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0]中编程的值。如果待发送的数据字节总数大 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD=0 且 NBYTES 数据传输完成时：
 - 在自动结束模式 (AUTOEND=1) 下，将自动发送停止位。
 - 在软件结束模式 (AUTOEND=0) 下，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。

- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

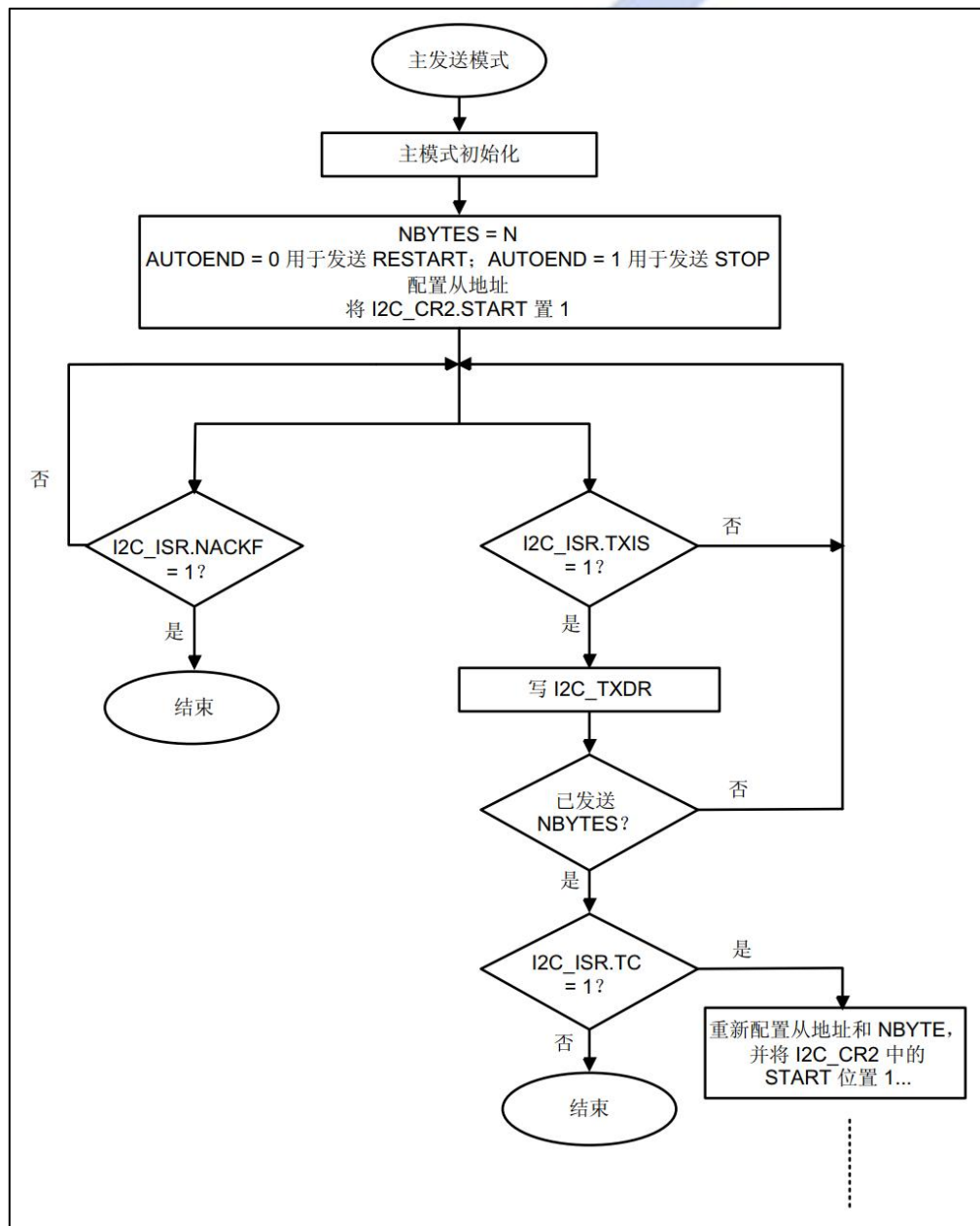


图 19-17 I2C 主发送器的传输序列流程图 (N≤255 字节)

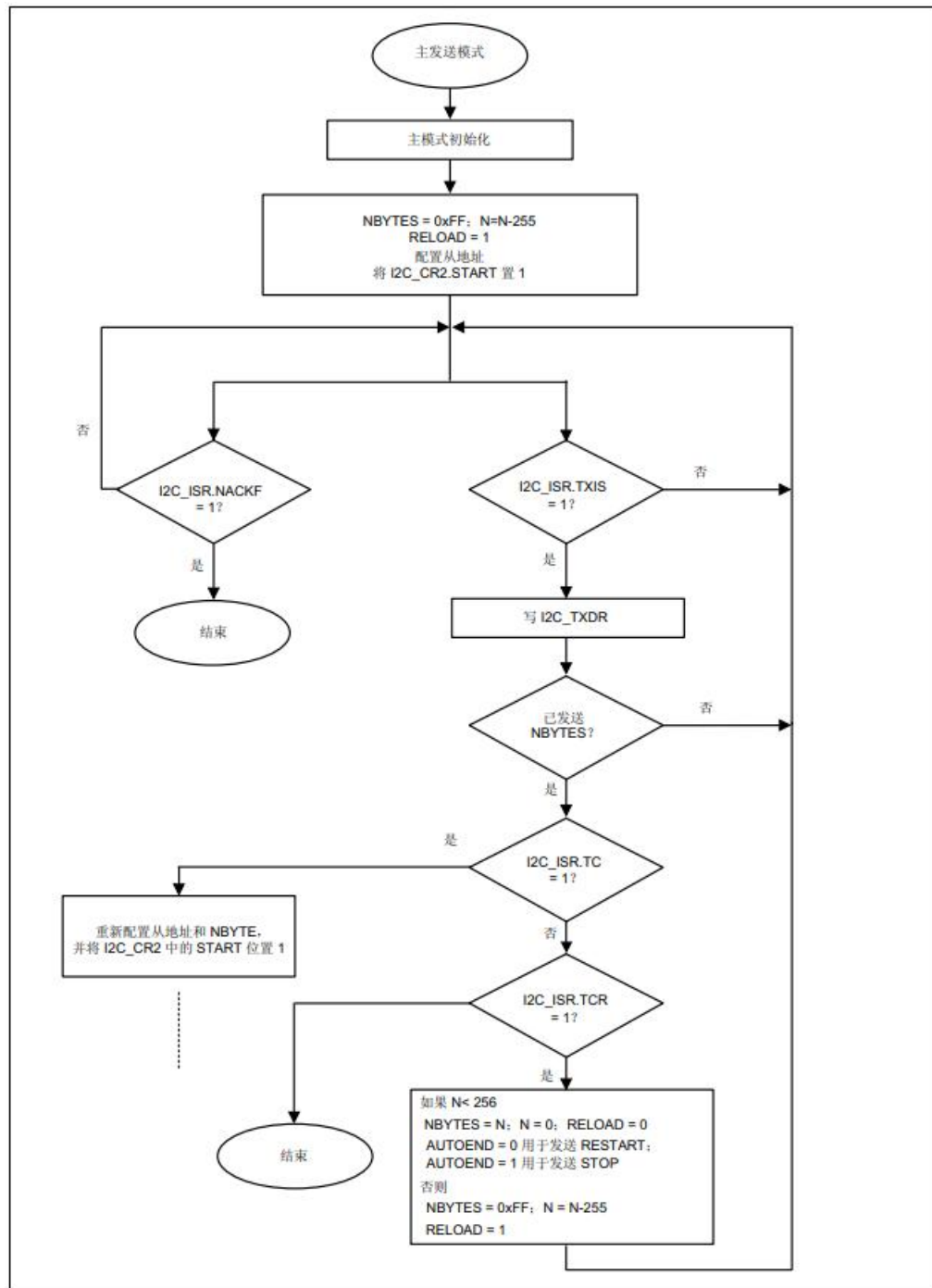
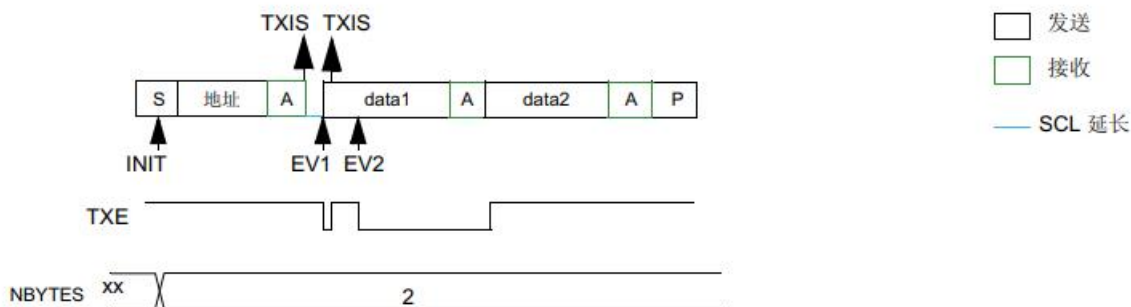


图 19-18 I2C 主发送器的传输序列流程图 (N>255 字节)

示例：I2C 主器件发送 2 个字节，自动结束模式 (STOP)

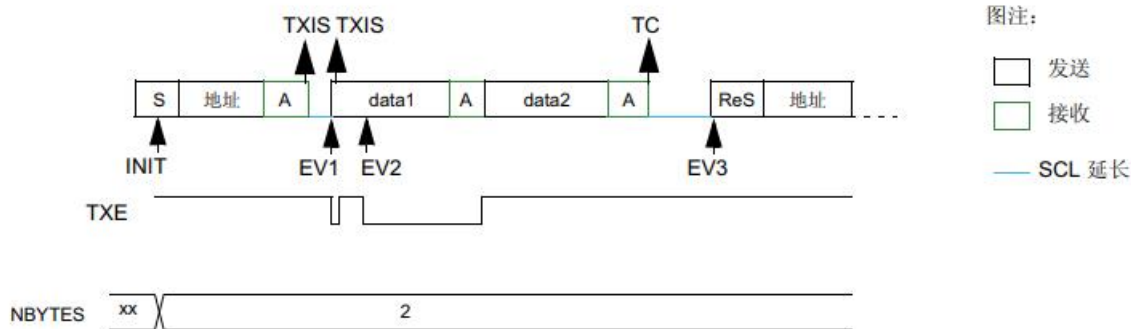


INIT: 设置从地址, 设置 NBYTES = 2, AUTOEND=1, 将 START 置 1

EV1: TXIS ISR: 写入 data1

EV2: TXIS ISR: 写入 data2

示例：I2C 主器件发送 2 个字节，软件结束模式 (RESTART)



INIT: 设置从地址, 设置 NBYTES = 2, AUTOEND=0, 将 START 置 1

EV1: TXIS ISR: 写入 data1

EV2: TXIS ISR: 写入 data2

EV3: TC ISR: 设置从地址, 设置 NBYTES = N, 将 START 置 1

图 19-19 I2C 主发送器的传输总线图

主接收器

读传输时, 在接收到每个字节 (即第 8 个 SCL 脉冲) 后, RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1, RXNE 事件将生成中断。读取 I2C_RXDR 时, 将清零该标志。如果待接收的数据字节总数大于 255, 则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下, 当 NBYTES[7:0] 数据传输完成时, TCR 标志将置 1, 并且 SCL 线的低电平将被延展, 直到 NBYTES[7:0] 被写入非零值。• 当 RELOAD=0 且 NBYTES[7:0] 数据传输完成时: – 在自动结束模式 (AUTOEND=1) 下, 接收到最后一个字节后, 将自动发送 NACK 和 停止位。– 在软件结束模式 (AUTOEND=0) 下, 接收到最后一个字节后, 将自动发送 NACK, TC 标志将置 1 且 SCL 线的低电平将被延展, 以便执行以下

软件操作： 可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数 来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起 始位，后跟从地址。 可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。

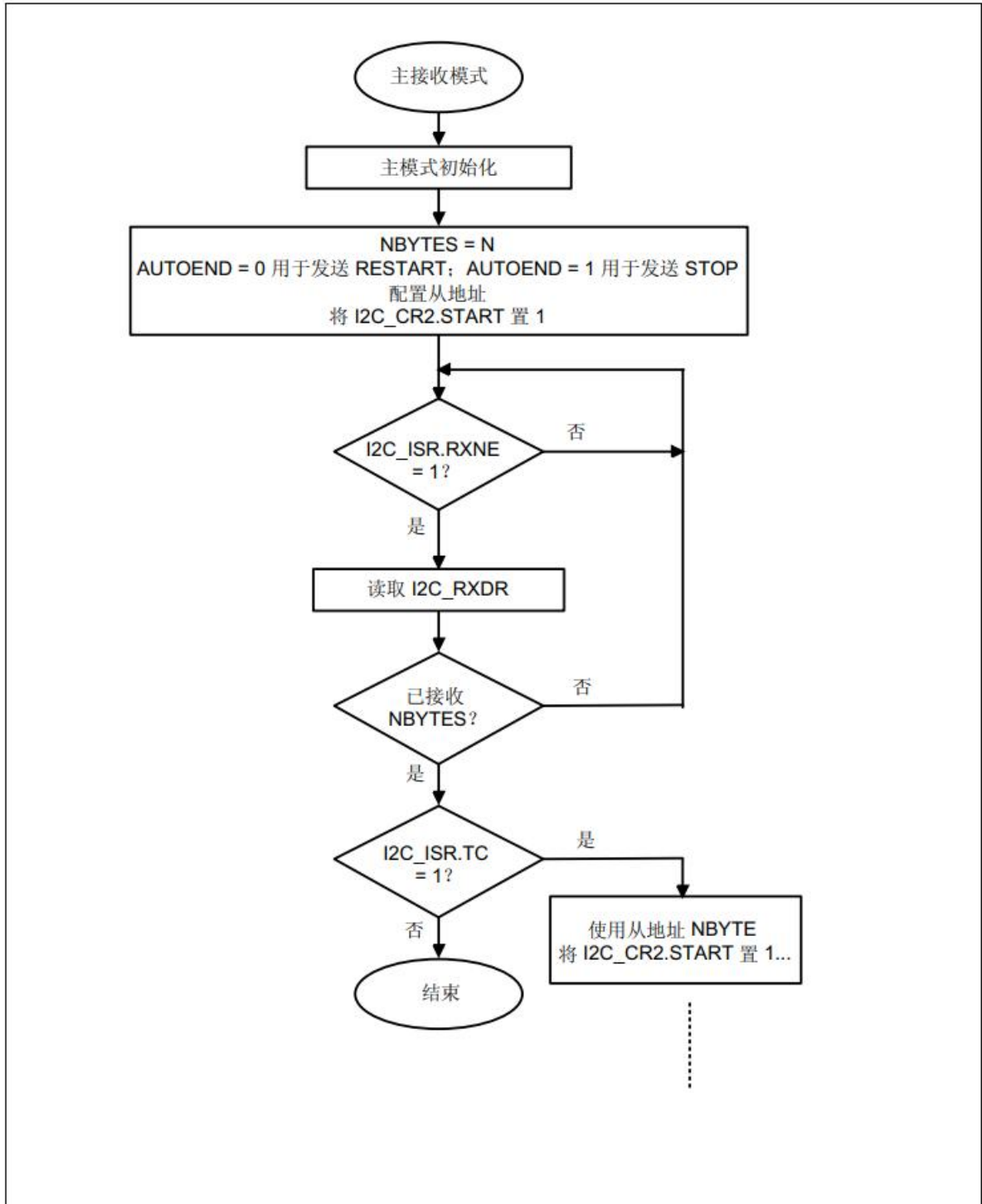


图 19-20 I2C 主接收器的传输序列流程图 (N≤255 字节)

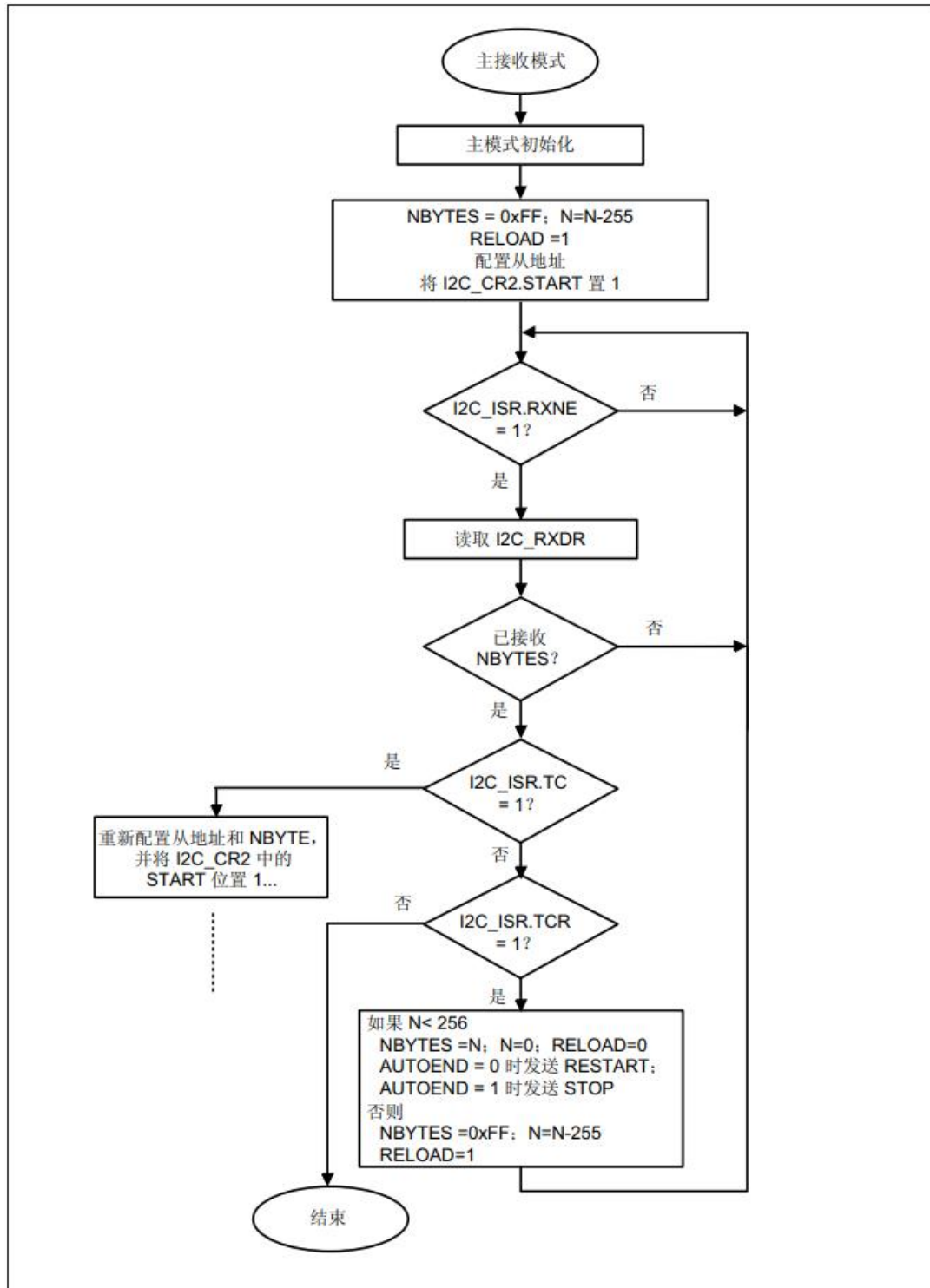


图 19-21 I2C 主接收器的传输序列流程图 (N>255 字节)

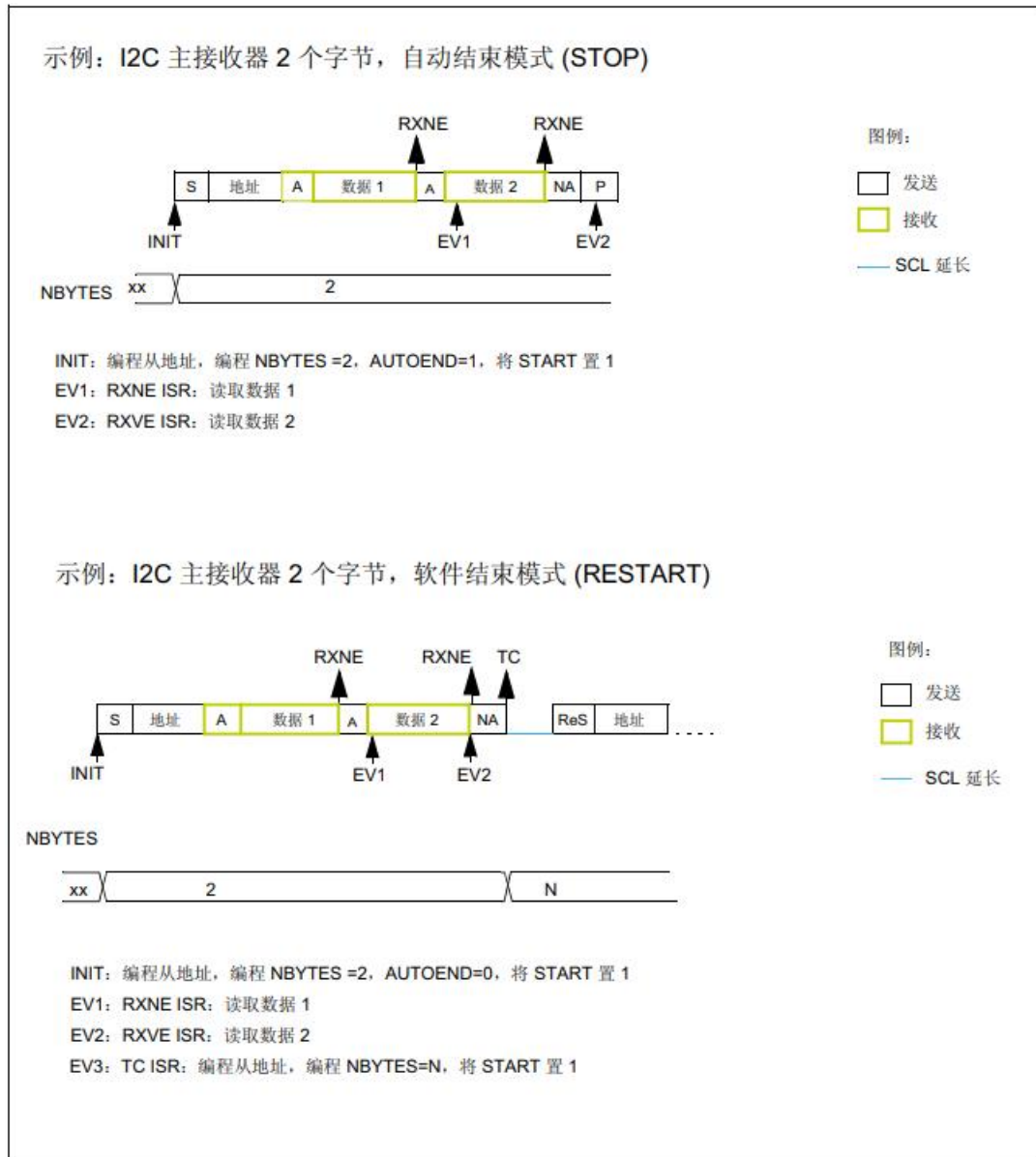


图 19-22 I2C 主接收器的传输总线图

19.3.9. I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I2C 规范的时序。

表 19-4 $f_{I2CCLK} = 8 \text{ MHz}$ 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10KHZ	100KHZ	400KHZ	500KHZ
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	$200 \times 250\text{ns} = 50\mu\text{s}$	$20 \times 250\text{ns} = 5.0\mu\text{s}$	$10 \times 125\text{ns} = 1250\text{ns}$	$7 \times 125\text{ns} = 875\text{ns}$
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	$196 \times 250\text{ns} = 49\mu\text{s}$	$16 \times 250\text{ns} = 4.0\mu\text{s}$	$4 \times 125\text{ns} = 500\text{ns}$	$4 \times 125\text{ns} = 500\text{ns}$

$t_{SCL}^{(1)}$	约 100 $\mu s^{(2)}$	约 10 $\mu s^{(2)}$	约 2500 ns ⁽³⁾	约 2500 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2 x 250ns = 500ns	2 x 250ns = 500ns	1 x 125ns = 125ns	0ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250ns = 1250ns	5 x 250ns = 1250ns	4 x 125ns = 500ns	2 x 125ns = 250ns

1. 由于 SCL 内部检测存在延时, SCL 周期 t_{SCL} 大于 $t_{SCLL} + t_{SCLH}$ 。为 t_{SCL} 提供的值仅用于举例说明。
2. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 500$ ns。 $t_{SYNC1} + t_{SYNC2} = 1000$ ns 时的示例。
3. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 500$ ns。 $t_{SYNC1} + t_{SYNC2} = 750$ ns 时的示例。
4. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 500$ ns。 $t_{SYNC1} + t_{SYNC2} = 655$ ns 时的示例。

表 19-5 $f_{I2CCLK} = 16$ MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10KHZ	100KHZ	400KHZ	500KHZ
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250ns = 50us	20 x 250ns = 5.0us	10 x 125ns = 1250ns	5x 62.5ns = 312.5ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250ns = 49us	16 x 250ns = 4.0us	4 x 125ns = 500ns	3 x 62.5ns = 187.5ns
$t_{SCL}^{(1)}$	约 100 $\mu s^{(2)}$	约 10 $\mu s^{(2)}$	约 2500 ns ⁽³⁾	约 1000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250ns = 500ns	2 x 250ns = 500ns	2x 125ns = 250ns	0ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250ns = 1250ns	5 x 250ns = 1250ns	4 x 125ns = 500ns	3 x 62.5ns = 187.5ns

1. 由于 SCL 内部检测存在延时, SCL 周期 t_{SCL} 大于 $t_{SCLL} + t_{SCLH}$ 。为 t_{SCL} 提供的值仅用于举例说明。
2. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 250$ ns。 $t_{SYNC1} + t_{SYNC2} = 1000$ ns 时的示例。
3. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 250$ ns。 $t_{SYNC1} + t_{SYNC2} = 750$ ns 时的示例。
4. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 250$ ns。 $t_{SYNC1} + t_{SYNC2} = 500$ ns 时的示例。

表 19-6 $f_{I2CCLK} = 48$ MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10KHZ	100KHZ	400KHZ	500KHZ
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	200 x 250ns = 50us	20 x 250ns = 5.0us	10 x 125ns = 1250ns	4 x 125ns = 500ns
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	196 x 250ns = 49us	16 x 250ns = 4.0us	4 x 125ns = 500ns	2 x 125ns = 250ns
$t_{SCL}^{(1)}$	约 100 $\mu s^{(2)}$	约 10 $\mu s^{(2)}$	约 2500 ns ⁽³⁾	约 875 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	2 x 250ns = 500ns	2 x 250ns = 500ns	3x 125ns = 375ns	0ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250ns = 1250ns	5 x 250ns = 1250ns	4 x 125ns = 500ns	3 x 125ns = 250ns

5. 由于 SCL 内部检测存在延时, SCL 周期 t_{SCL} 大于 $t_{SCLL} + t_{SCLH}$ 。为 t_{SCL} 提供的值仅用于举例说明。
6. $t_{SYNC1} + t_{SYNC2}$ 最小值为 $4 \times t_{I2CCLK} = 83.3$ ns。 $t_{SYNC1} + t_{SYNC2} = 1000$ ns 时的示例。

7. $t_{\text{SYNC1}} + t_{\text{SYNC2}}$ 最小值为 $4 \times t_{\text{I2CCLK}} = 83.3 \text{ ns}$ 。 $t_{\text{SYNC1}} + t_{\text{SYNC2}} = 750 \text{ ns}$ 时的示例。
8. $t_{\text{SYNC1}} + t_{\text{SYNC2}}$ 最小值为 $4 \times t_{\text{I2CCLK}} = 83.3 \text{ ns}$ 。 $t_{\text{SYNC1}} + t_{\text{SYNC2}} = 250 \text{ ns}$ 时的示例。

19.3.10. SMBUS 特性

仅当支持 SMBus 功能时，才涉及本节内容。

简介

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBUS 规范第 2.0 版兼容 (<http://smbus.org>)。系统管理总线规范涉及三类器件。

- 从器件，用于接收或响应命令。
- 主器件，用于发出命令、生成时钟和中止传输。
- 主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主-从器件功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。该外设可配置为主器件或从器件，也可配置为主机。

SMBUS 以 I2C 规范第 3.1 版为基础。

总线协议

任何给定器件都有十一种可用命令协议。器件既可以在这十一种协议中任选其一，也可以使用全部十一种协议进行通信。这十一种协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议应通过用户软件实施。

地址解析协议 (ARP)

通过为各个从器件动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符(UDID)。该 128 位数字由软件实现。

该外设支持地址解析协议(ARP)。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。ARP 命令应通过用户软件实现。此外，还将在从模式下执行仲裁以支持 ARP。

接收的命令和数据应答控制

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见“[从器件字节控制模式](#)”。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址(0b0001000)。使用该协议时，器件作为主器件，而主机作为从器件。

SMBus 报警

器件支持 SMBus ALERT 可选信号。只具备从功能的器件可通过 SMBALERT 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址(0b0001 100)同时访问所有 SMBALERT 器件。只有那些将 SMBALERT 拉到低电平的器件会应答报警响应地址。如果配置为从器件(SMBHEN=0)，则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。如果配置为主机 (SMBHEN=1)，则当 SMBA 引脚上检测到下降沿且 ALERTEN=1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码 (PEC)。PEC 的计算方式是对所有消息字节 (包括地址和读/写位) 使用 CRC-8 多项式 $C(x) = x^8 + x^2 + x + 1$ 。外设内置了硬件 PEC 计算器, 可在接收到的字节与硬件计算的 PEC 不匹配时自动发送否定应答号。

超时

该外设内置了硬件定时器, 以便符合 SMBus 规范中定义的 3 个超时。

表 19-7 SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
t_{TIMEOUT}	检测时钟低电平超时	25	35	ms
$t_{\text{LOW:SEXT}}^{(1)}$	累积时钟低电平延长时间 (从器件)	-	25	ms
$t_{\text{LOW:MEXT}}^{(2)}$	累积时钟低电平延长时间 (主器件)	-	10	ms

$t_{\text{LOW:SEXT}}$ 是一段累积时间, 即给定从器件在一条消息的最初起始到停止期间时钟信号可延展的时间。其它从器件或主器件也可能延长时钟, 进而导致时钟低电平总延长时间超过 $t_{\text{LOW:SEXT}}$ 。因此, 测量该参数时该器件应该是全速主器件寻址的唯一器件。

$t_{\text{LOW:MEXT}}$ 是一段累积时间, 即主器件在消息的每个字节 (定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP) 内时钟信号可延展的时间。从器件或其它主器件也可能延长时钟, 进而导致时钟低电平总时间超过 $t_{\text{LOW:MEXT}}$ (针对给定字节)。因此, 测量该参数时该全速主器件只寻址一个从器件。

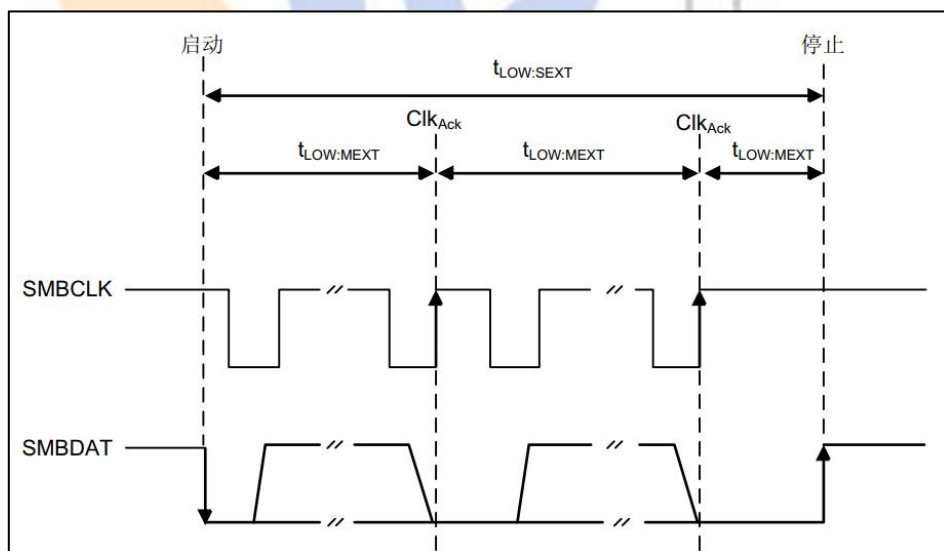


图 19-23 超时间隔

总线空闲检测

如果主器件检测到时钟和数据信号的高电平时间已达 t_{IDLE} (超过 $t_{\text{HIGH,Max}}$), 则认为总线空闲。

该时序参数已考虑如下情况: 主器件已动态添加至总线, 但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下, 主器件必须等待足够长的时间, 以确定当前未进行传输。外设支持硬件总线空闲检测。

19.3.11. SMBUS 初始化

仅当支持 SMBus 功能时，才涉及本节内容。

除了 I2C 初始化之外，还必须进行一些其它的特定初始化，以便执行 SMBus 通信。

接收的命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见本章“[从器件字节控制模式](#)”。

特定地址（从模式）

必要时应使能特定的 SMBus 地址。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址 (0b0001 000)。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址 (0b0001100)。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器（I2C_CR2 寄存器中的 NBYTES[7:0]）来管理 PEC 传输。使能 I2C 之前，必须配置 PECEN 位。PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES-1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

注意：使能 I2C 时，不允许更改 PECEN 配置。

表 19-8 带 PEC 的 SMBUS 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOCTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范第 2.0 版规定的时间最大值之前检测出超时情况。

• t_{TIMEOUT} 检查

要使能 t_{TIMEOUT} 检查，必须将 12 位 TIMEOUTA[11:0] 位编程为定时器重载值，以检查 t_{TIMEOUT} 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。然后，通过将 I2C_TIMEOCTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 的低电平持续时间超过 (TIMEOUTA+1)* 2048 * t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

注意：TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

• t_{LOW:SEXT} 和 t_{LOW:MEXT} 检查

必须根据外设配置为主器件还是从器件来配置 TIMEOUTB 定时器，以便为从器件校验 $t_{\text{LOW:SEXT}}$ ，为主器件校验 $t_{\text{LOW:MEXT}}$ 。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。

如果 SMBus 外设延展 SCL 的累积时间超过 $(\text{TIMEOUTB}+1) * 2048 * t_{\text{I2CCLK}}$ ，并且达到超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMEOUTEN 位置 1 来使能定时器。如果 SCL 和 SDA 线的高电平持续时间超过 $(\text{TIMEOUTA}+1) * 4 * t_{\text{I2CCLK}}$ ，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见表 19-11 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{\text{IDLE}} = 50\mu\text{s}$ ）。

注意：TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

19.3.12. SMBUS 通信

仅当支持 SMBus 功能时，才涉及本节内容。

- 将 t_{TIMEOUT} 的最大持续时间配置为 25 ms:

表 19-9 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{\text{TIMEOUT}} = 25\text{ms}$ ）

f_{I2CCLK}	TIMEOUTA[11:0]	TIDLE	TIMEOUTEN	t_{TIMEOUT}
8 MHz	0x61	0	1	$98 * 2048 * 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 * 2048 * 62.5 \text{ ns} = 25 \text{ ms}$
32 MHz	0x186	0	1	$391 * 2048 * 31.25 \text{ ns} = 25 \text{ ms}$

- 将 $t_{\text{LOW:SEXT}}$ 和 $t_{\text{LOW:MEXT}}$ 的最大持续时间配置为 8 ms:

表 19-10 不同 I2CCLK 频率下的 TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0]	TEXTEN	$t_{\text{LOW:EXT}}$
8 MHz	0x1F	1	$32 * 2048 * 125 \text{ ns} = 8 \text{ ms}$
16 MHz	0x3F	1	$64 * 2048 * 62.5 \text{ ns} = 8 \text{ ms}$
32 MHz	0x7C	1	$125 * 2048 * 31.25 \text{ ns} = 8 \text{ ms}$

- 将 t_{IDLE} 的最大持续时间配置为 $50 \mu\text{s}$

表 19-11 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{\text{IDLE}} = 50\mu\text{s}$ ）

f_{I2CCLK}	TIMEOUTA[11:0]	TIDLE	TIMEOUTEN	t_{IDLE}
8 MHz	0x63	1	1	$100 * 4 * 125 \text{ ns} = 50 \mu\text{s}$
16 MHz	0xC7	1	1	$200 * 4 * 62.5 \text{ ns} = 50 \mu\text{s}$

32 MHz	0x18F	1	1	$400 * 4 * 31.25 \text{ ns} = 50 \mu\text{s}$
--------	-------	---	---	---

SMBus 从发送器

在 SMBus 模式下使用 IP 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0]中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES-1，如果主器件在完成 NBYTES-1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

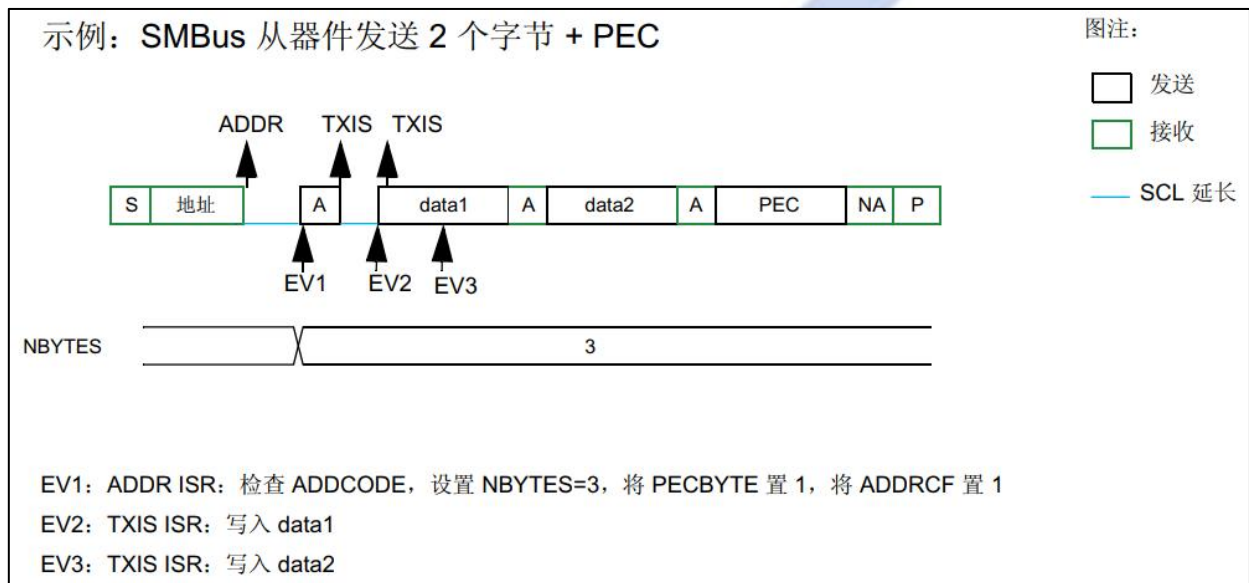


图 19-24 SMBus 从发送器传输总线图 (SBC=1)

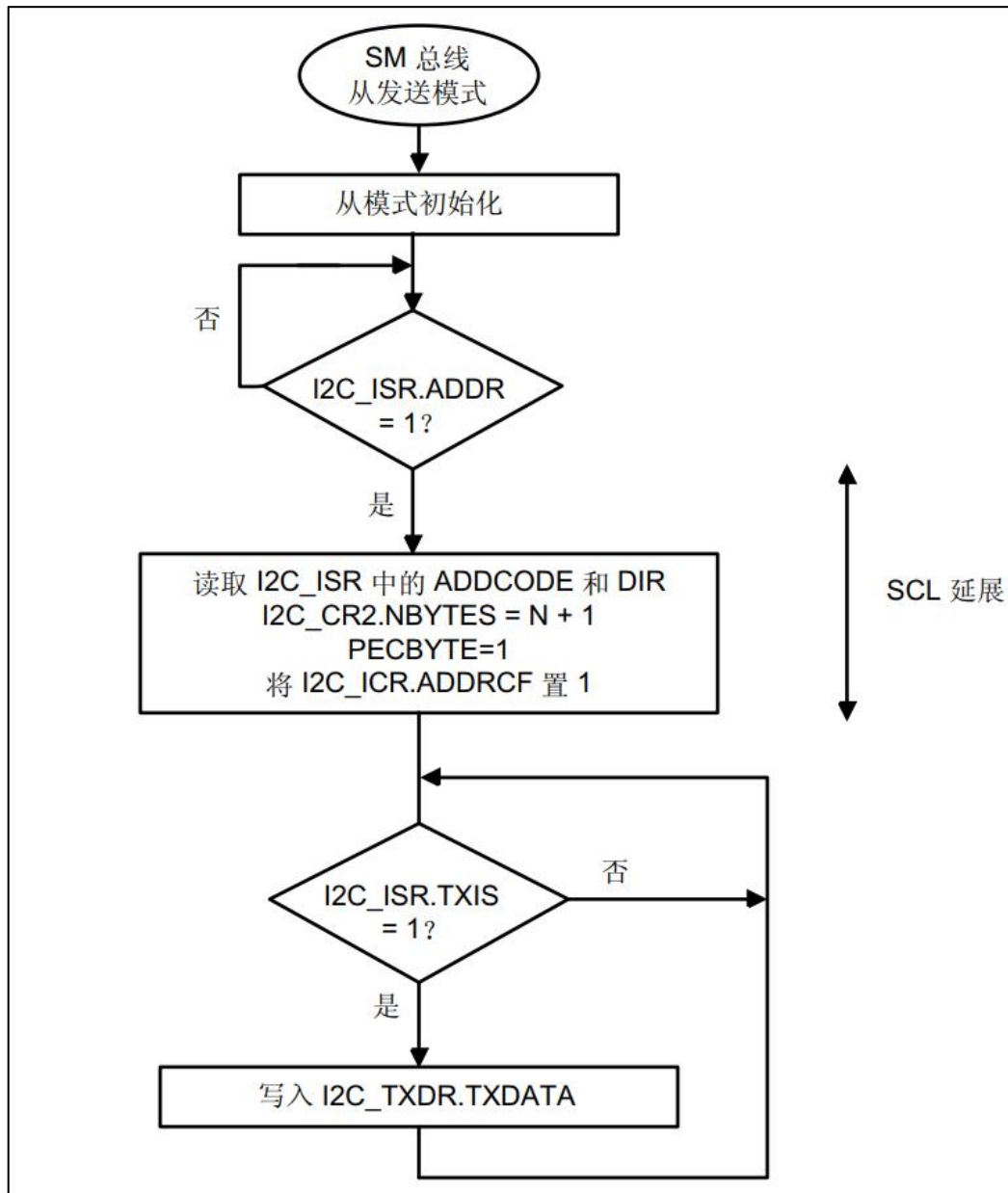


图 19-25 SMBus 从发送器传输序列流程图（N 字节+PEC）

SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式(RELOAD=1)。更多详细信息，请参见“从器件字节控制模式”。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES-1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其它数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。如果无需 ACK 软件控制，用户可编程 PECBYTE=1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。

接收到 NBYTES-1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

注意：当 RELOA 位置 1 时，PECBYTE 位将不起作用。

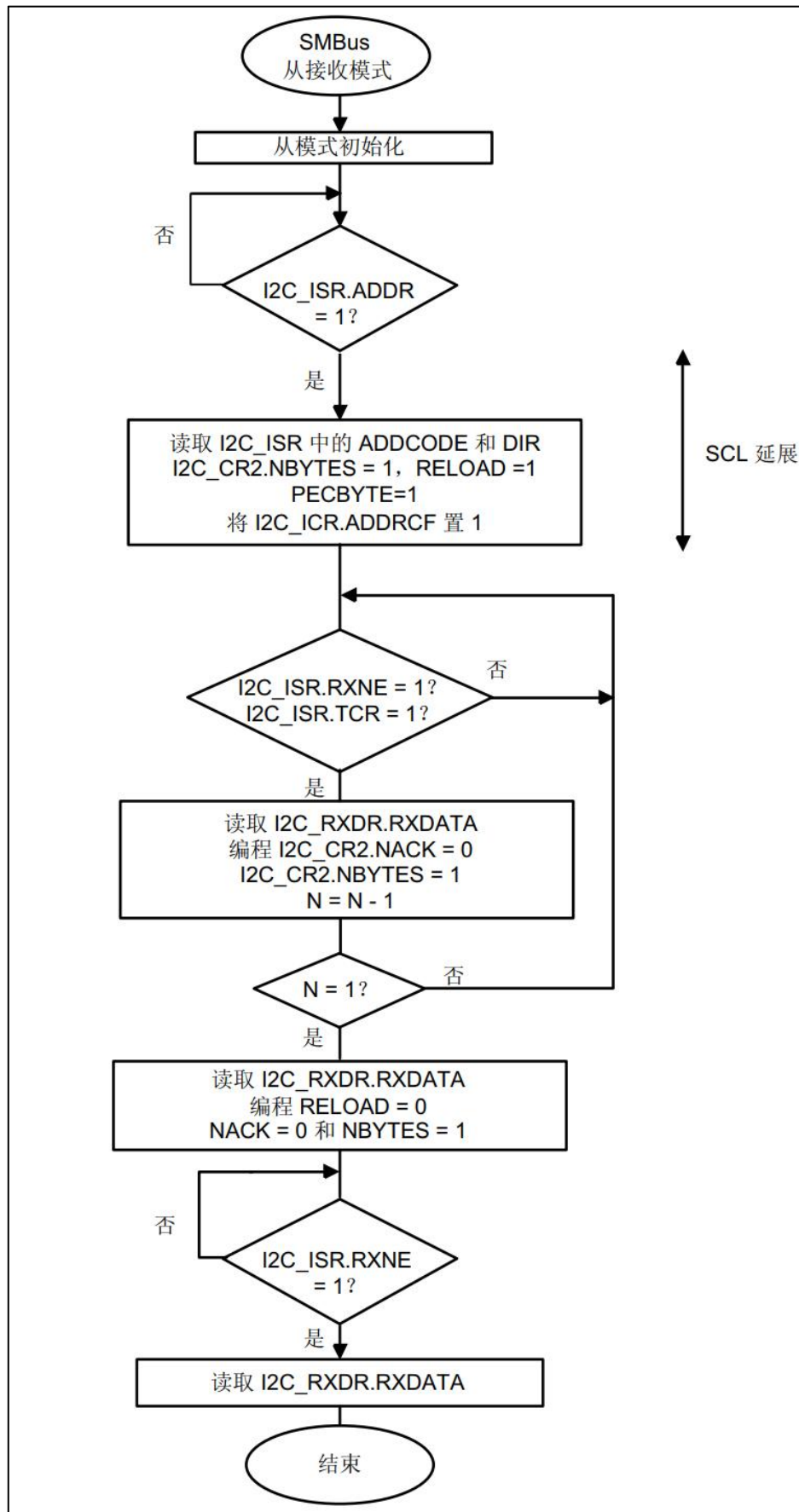
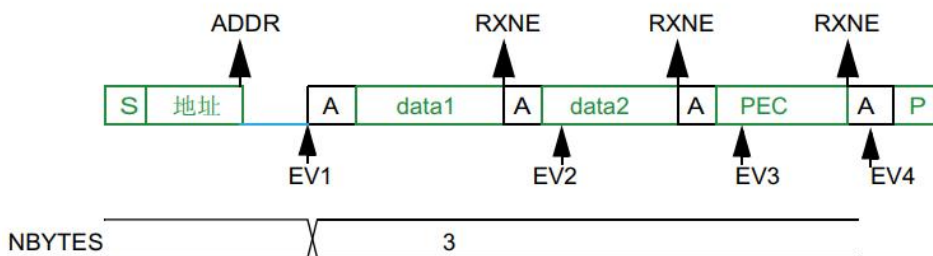


图 19-26 SMBus 从接收器传输序列流程图（N 字节+PEC）

示例：SMBus 从器件接收 2 个字节 + PEC

图注：

- 发送
- 接收
- SCL 延长

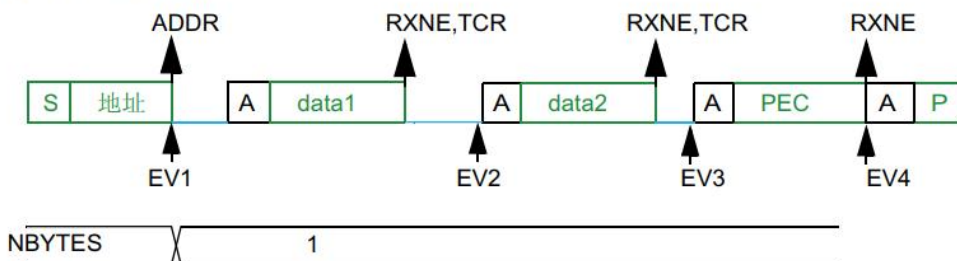


- EV1: ADDR ISR: 检查 ADDCODE 和 DIR, 设置 NBYTES = 3, PECBYTE=1, RELOAD=0, 将 ADDRCONF 置 1
- EV2: RXNE ISR: 读取 data1
- EV3: RXNE ISR: 读取 data2
- EV4: RXNE ISR: 读取 PEC

示例：SMBus 从器件接收 2 个字节 + PEC, 具有 ACK 控制 (RELOAD=1/0)

图注：

- 发送
- 接收
- SCL 延长



- EV1: ADDR ISR: 检查 ADDCODE 和 DIR, 设置 NBYTES = 1, PECBYTE=1, RELOAD=1, 将 ADDRCONF 置 1
- EV2: RXNE-TCR ISR: 读取 data1, 设置 NACK=0, NBYTES = 1
- EV3: RXNE-TCR ISR: 读取 data2, 设置 NACK=0, NBYTES = 1, RELOAD=0
- EV4: RXNE-TCR ISR: 读取 PEC

图 19-27 SMBus 从接收器总线传输图

仅当支持 SMBus 功能时, 才涉及本节内容。

除 I2C 主模式传输管理 (请参见本章: I2C 主模式) 之外, 还提供了一些额外的软件流程图来支持 SMBus。

SMBus 主发送器

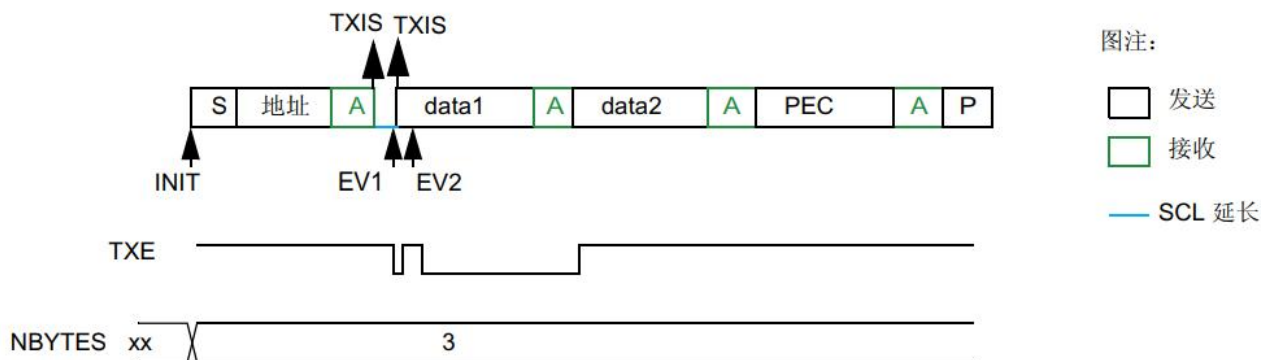
当 SMBus 主器件想要发送 PEC 时, 必须在 START 位置 1 前, 将 PECBYTE 位置 1 并在 NBYTES[7:0] 字段中设置字节数。在这种情况下, 总 TXIS 中断数为 NBYTES-1。因此, 如果 PECBYTE 位在 NBYTES=0x1 时置 1, 则将自动发送 I2C_PECR 寄存器的内容。

如果 SMBus 主器件想要在 PEC 后发送停止位, 则应选择自动结束模式 (AUTOEND=1)。在这种情况下, 传输 PEC 后将自动发送停止位。

如果 SMBus 主器件想要在 PEC 后发送重复起始位，则必须选择软件模式 (AUTOEND=0)。在这种情况下，发送 NBYTES-1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

示例：SMBus 主器件发送 2 个字节 + PEC，自动结束模式 (STOP)

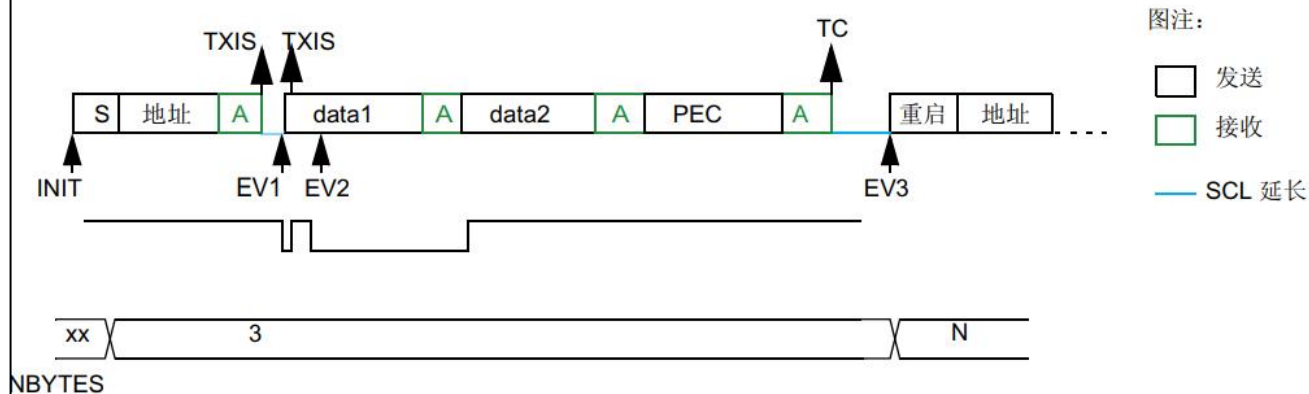


INIT: 设置从地址，设置 NBYTES = 3，AUTOEND=1，将 PECBYTE 置 1，将 START 置 1

EV1: TXIS ISR: 写入 data1

EV2: TXIS ISR: 写入 data2

示例：SMBus 主器件发送 2 个字节 + PEC，软件结束模式 (RESTART)



INIT: 设置从地址，设置 NBYTES = 3，AUTOEND=0，将 PECBYTE 置 1，将 START 置 1

EV1: TXIS ISR: 写入 data1

EV2: TXIS ISR: 写入 data2

EV3: TC ISR: 设置从地址，设置 NBYTES = N，将 START 置 1

图 19-28 SMBus 主发送器总线传输图

SMBus 主接收器

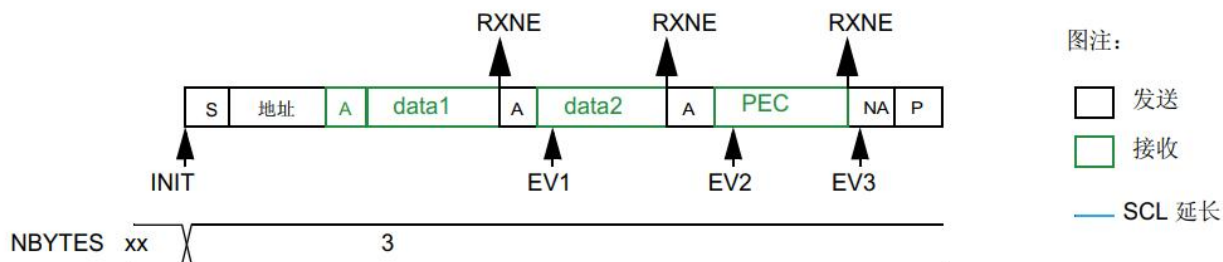
当 SMBus 主器件想要接收 PEC，并在传输结束后接收 STOP 时，可选择自动结束模式(AUTOEND=1)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。PEC 字节（其后跟有停止位）将得到 NACK 响应。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后接收重复起始位时，必须选择软件模式(AUTOEND=0)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。



示例：SMBus 主器件接收 2 个字节 + PEC，自动结束模式 (STOP)



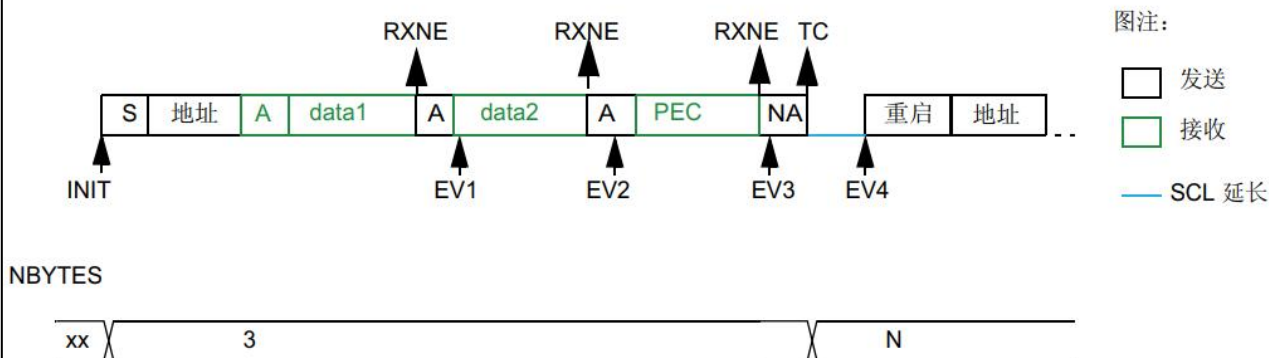
INIT: 设置从地址, 设置 NBYTES = 3, AUTOEND=1, 将 PECBYTE 置 1, 将 START 置 1

EV1: RXNE ISR: 读取 data1

EV2: RXNE ISR: 读取 data2

EV3: RXNE ISR: 读取 PEC

示例：SMBus 主器件接收 2 个字节 + PEC，软件结束模式 (RESTART)



INIT: 设置从地址, 设置 NBYTES = 3, AUTOEND=0, 将 PECBYTE 置 1, 将 START 置 1

EV1: RXNE ISR: 读取 data1

EV2: RXNE ISR: 读取 data2

EV3: RXNE ISR: 读取 PEC

EV4: TC ISR: 设置从地址, 设置 NBYTES = N, 将 START 置 1

图 19-29 SMBus 主接收器总线传输图

19.3.13. 错误条件

以下错误条件可能导致通信失败。

总线错误 (BERR)

当检测到起始位或停止位但不位于第 9N 个 SCL 时钟脉冲之后时, 会检测到总线错误。当 SDA 边沿出现且

SCL 为高电平时，会检测到起始或停止位。只有当 I2C 在传输过程中用作主器件或被寻址为从器件时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。在从模式下误检测到起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失 (ARLO)

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主器件自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误 (OVR)

当满足 NOSTRETCH=1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 在接收过程中接收到一个新的字节，但 RXDR 寄存器的值还未被读取。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
 - 当 STOPF=1 且应发送第一个数据字节时。TXE=0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
 - 应发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误 (PECERR)

仅当支持 SMBus 功能时，才涉及本节内容。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误 (TIMEOUT)

仅当支持 SMBus 能时，才涉及本节内容。满足以下任何条件均会出现超时错误：

- TIDLE=0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0]位中定义的时间：这用于检测 SMBus 超时。
- TIDLE=1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0]位中定义的时间：这用于检测总线空闲情况。
- 主器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0]位中定义的时间（SMBus $t_{low:MEXT}$ 参数）
- 从器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{low:SEXT}$ 参数）

当在主模式下检测到超时时，将自动发送停止位。当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警 (ALERT)

仅当支持 SMBus 功能时，才涉及本节内容。

当 I2C 接口配置为主机 (SMBHEN=1)、使能了报警引脚检测 (ALERTEN=1) 并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

19.3.14. 中断及异常说明

下表给出了 I2C 中断请求列表。

表 19-12 I2C 中断请求

中断事件	事件标志	事件标志/ 中断清除方法	中断使能控制位
接收缓冲区非空 (Receive buffer not empty)	RXNE	读取 I2C_RXDR 寄存器	RXIE
发送缓冲区中断状态	TXIS	写入 I2C_TXDR 寄存器	TXIE
停止位检测中断标志	STOPF	写入 STOPCF=1	STOPIE
传输完成等待重载	TCR	写入 I2C_CR2 (NBYTES[7:0] $\neq$ 0)	TCIE
传输完成	TC	写入 START=1 或 STOP=1	
地址匹配	ADDR	写入 ADDRCONF=1	ADDRIE
接收到 NACK 应答	NACKF	写入 NACKCF=1	NACKIE
总线错误 (Bus error)	BERR	写入 BERRCF=1	ERRIE
仲裁丢失	ARLO	写入 ARLOCF=1	
上溢/下溢 (Overrun/Underrun)	OVR	写入 OVRCONF=1	
PEC 错误	PECERR	写入 PECERRCONF=1	
超时/ t_{LOW} 错误	TIMEOUT	写入 TIMEOUTCONF=1	
SMBus 报警	ALERT	写入 ALERTCONF=1	

上述所有中断共享同一个中断向量 (I2C 全局中断)

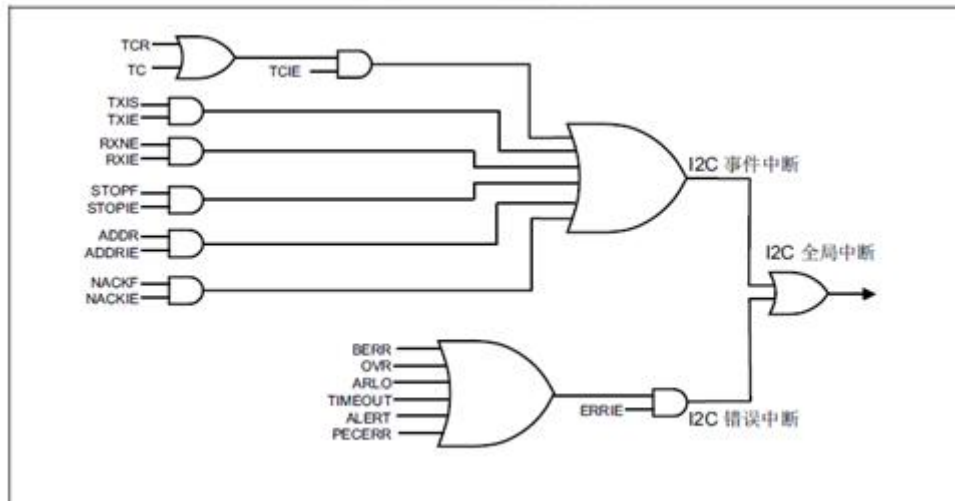


图 19-30 中断映射图

19.4. I2C 寄存器

19.4.1. I2C 寄存器映射

表 19-13 I2C 寄存器总表

偏移	寄存器名	访问	复位值
0x00	I2C 版本寄存器 (VER)	r	0xa368_0010
0x04	I2C 控制寄存器 1 (CR1)	rw	0x0000_0000
0x08	I2C 控制寄存器 2 (CR2)	rw	0x0000_0000
0x0C	I2C 自身地址 1 寄存器 (OA1)	rw	0x0000_0000
0x10	I2C 自身地址 2 寄存器 (OA2)	rw	0x0000_0000
0x14	I2C 时序寄存器 (TIMINGR)	rw	0x0000_0000
0x18	I2C 超时寄存器 (TIMEOUTR)	rw	0x0000_0000
0x1C	I2C 中断和状态寄存器 (ISR)	r	0x0000_0000
0x20	I2C 中断清零寄存器 (ICR)	w	0x0000_0000
0x28	I2C PEC 寄存器 (PECR)	rw	0x0000_0000
0x2C	I2C 接收数据寄存器 (RXDR)	rw	0x0000_0000
0x30	I2C 发送数据寄存器 (TXDR)	rw	0x0000_0000

19.4.2. I2C 版本寄存器(VER)

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

位域	功能
31-16 Vender	版本号 0xA386
15-8 MVER	主版本号
7-0 SVER	子版本号

19.4.3. I2C 控制寄存器 1 (CR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	-	NOSTRETCH	SBC
-	-	-	-	-	-	-	-	rw	rw	rw	rw	rw	-	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	ANFOFF	DNF				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
-	-	-	rw	rw				rw	rw	rw	rw	RXIE	rw	rw	rw

位域	功能
31-24	保留
23 PECEN	PEC enable (PEC 计算使能) 0: 禁止 PEC 计算 1: 使能 PEC 计算 <i>注意: 如果不支持 SMBus 功能, 该位保留, 并由硬件强制为 “0”。</i>
22 ALERTEN	SMBus alert enable(SMBus 报警使能) 从机模式 (SMBHEN=0): 0: 将 SMBA 引脚释放为高电平并禁止报警响应地址头: 0001100x 后跟 NACK。 1: 将 SMBA 引脚驱动为低电平并使能报警响应地址头: 0001100x 后跟 ACK。 主机模式 (SMBHEN=1): 0: 不支持 SMBus 报警引脚 (SMBA)。 1: 支持 SMBus 报警引脚(SMBA)。 <i>注意: 当 ALERTEN=0 时, SMBA 引脚可用作标准 GPIO。</i> <i>如果不支持 SMBus 功能, 该位保留, 并由硬件强制为 “0”。</i>
21 SMBDEN	SMBus Device Default address enable (SMBus 期间默认地址使能) 0: 禁止器件默认地址。不对地址 0b1100001x 应答。 1: 使能器件默认地址。对地址 0b1100001x 应答。 <i>注意: 如果不支持 SMBus 功能, 该位保留, 并由硬件强制为 “0”。</i>
20 SMBHEN	SMBus Host address enable(SMBus 主机地址使能) 0: 禁止主机地址。不对地址 0b0001000x 应答。 1: 使能主机地址。对地址 0b0001000x 应答。 <i>注意: 如果不支持 SMBus 功能, 该位保留, 并由硬件强制为 “0”。</i>
19 GCEN	General call enable (广播呼叫使能) 0: 禁止广播呼叫。不对地址 0b00000000 应答。 1: 使能广播呼叫。对地址 0b00000000 应答。
18	保留
17 NOSTRETCH	Clock stretching disable (时钟延长禁止) 该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。 0: 使能时钟延长 1: 禁止时钟延长 <i>注: 该位只能在 I2C 禁止时 (PE = 0) 编程。</i>
16 SBC	Slave byte control (从设备模式下的字节控制) 该位用于在从设备模式下使能硬件字节控制。 0: 禁止从设备模式下的字节控制 1: 使能从设备模式下的字节控制

位域	功能
15-13	保留
12 ANFOFF	Analog noise filter OFF（模拟噪声滤波器关闭） 0：使能模拟噪声滤波器 1：禁止模拟噪声滤波器 注意：该位只能在 I2C 禁止时 (PE = 0) 编程。
11-8 DNF	Digital noise filter（数字噪声滤波器） 这些位用于配 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可滤除脉宽 DNF[3:0] * t_{I2CCLK} 以下的尖峰： 0000：禁止数字滤波器 0001：使能数字滤波器，可滤除的噪声尖峰脉宽可达 1 t_{I2CCLK} ... 1111：使能数字滤波器，可滤除的噪声尖峰脉宽可达 15 t_{I2CCLK} 注意：如果模拟滤波器也已使能，数字滤波将叠加在模拟滤波之上。该滤波器只能在 I2C 禁止时 (PE = 0) 编程。
7 ERRIE	Error interrupt enable（错误中断使能） 0：禁止错误检测中断 1：使能错误检测中断 注：以下任一错误均会生成中断： 仲裁丢失 (ARLO) 总线错误检测 (BERR) 上溢/下溢 (OVR) 超时检测 (TIMEOUT) PEC 错误检测 (PECERR) 报警引脚事件检测 (ALERT)
6 TCIE	Transfer Complete interrupt enable（传输完成中断使能） 0：禁止传输完成中断 1：使能传输完成中断 注意：以下任一事件均会生成中断： 传输完成 (TC) 传输完成等待重载 (TCR)
5 STOPIE	STOP detection interrupt enable（停止位检测中断使能） 0：禁止停止位检测 (STOPF) 中断 1：使能停止位检测 (STOPF) 中断
4 NACKIE	Not Acknowledge received interrupt enable（接收到否定应答中断使能） 0：禁止接收到否定应答 (NACKF) 中断 1：使能接收到否定应答 (NACKF) 中断
3 ADDRIE	Address match interrupt enable（slave only）（地址匹配中断使能，仅从模式） 0：禁止地址匹配 (ADDR) 中断 1：使能地址匹配 (ADDR) 中断
2 RXIE	Receive (RXNE) interrupt enable（RX 中断使能） 0：禁止接收 (RXNE) 中断 1：使能接收 (RXNE) 中断
1 TXIE	Transmit (TXIS) interrupt enable（TX 中断使能） 0：禁止发送 (TXIS) 中断 1：使能发送 (TXIS) 中断
0 PE	Peripheral enable（外设使能） 0：禁止外设 1：使能外设 注意：当 PE=0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须

位域	功能
	保持低电平状态至少 3 个 APB 时钟周期。

19.4.4. I2C 控制寄存器 2 (CR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	PECBYTE	AUTOEND	RELOAD	NBYTES							
-	-	-	-	-	rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD10	ADD10	RD_WRN	SADD[9:8]		SADD[7:1]						SADD[0]	
rw	rw	rw	rw	rw	rw	rw		rw						rw	

位域	功能
31-27	保留
26 PECBYTE	数据包错误校验字节(PEC error checking byte) 此位由软件置 1，并可在接收到停止条件或匹配地址时、或者 PE=0 时由硬件清零。 0：不传输 PEC。 1：请求 PEC 发送/接收。 注意：向该位写入“0”不起作用。 当 RELOAD 置 1 时，该位不起作用。 当 SBC=0 时，该位在从模式下不起作用。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。
25 AUTOEND	自动结束模式，主模式(Automatic end mode, master mode) 此位由软件置 1 和清零。 0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。 1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。 注意：在从模式下，或当 RELOAD 位置 1 时，该位不起作用。1：支持 SMBus 报警引脚 (SMBA)。 注意：当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。
24 RELOAD	NBYTES 重载模式(NBYTES reload mode) 此位由软件置 1 和清零。 0：传输 NBYTES 数据（后跟停止位或重复起始位）之后即完成传输。 1：传输 NBYTES 数据之后未完成传输（将重载 NBYTES）。当 NBYTES 数据传输完成时，TCR 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。
23-16 SMBHEN	字节数(Number of bytes) 在此设置待发送/接收的字节数。在从模式下，当 SBC=0 时，该字段为无关字段。 注意：START 位置 1 时，不允许更改这些位。
15 NACK	NACK 生成，从模式(NACK generation, slave mode) 此位由软件置 1，并可在发送 NACK 时、接收到停止条件或匹配地址时、或者 PE=0 时由硬件清零。 0：在当前接收的字节后发送 ACK。 1：在当前接收的字节后发送 NACK。 注意：向该位写入“0”不起作用。

位域	功能
	该位仅在从模式下使用：在主接收器模式下，无论NACK位的值为何，最后一个字节（后跟停止位或重复起始位）后都将自动生成NACK。 当从接收器NOSTRETCH模式下发生上溢时，无论NACK位的值为何，都将自动生成NACK。 使能硬件PEC校验时 (PECBYTE=1)，PEC应答值与NACK值无关。
14 STOP	停止位生成，主模式(Stop generation master mode) 该位由软件置1，并可在检测到停止位时或PE=0时由硬件清零。在主模式下： 0：不生成停止位。 1：在当前字节传输完成后生成停止位。 注意：向该位写入“0”不起作用。
13 START	起始位生成(Start generation, master mode) 该位由软件置1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者PE=0时由硬件清零。它也可由软件清零，方法是向I2C_ICR寄存器中的ADDRCF位写入“1”。 0：不生成起始位。 1：生成重复起始/起始位： - 如果I2C已处于主模式下且AUTOEND=0，则将该位置1会在NBYTES传输结束后且RELOAD=0的情况下生成重复起始位。 - 否则，将该位置1会在总线释放后立即生成起始位。 注意：向该位写入“0”不起作用。 即使总线繁忙或I2C处于从模式，也可将START位置1。 当RELOAD置1时，该位不起作用。
12 HEAD10R	读方向传输时只发送10位地址的前7位头字节，主接收器模式(10-bit address header only read direction master, receive mode) 0：主器件发送完整的10位从地址读序列：起始位 + 带写方向的2字节10位地址 + 重复起始位 + 带读方向的10位地址的前7位。 1：主器件只发送10位地址的前7位，后跟读方向。 注意：START位置1时，不允许更改此位。
11 ADD10	10位寻址模式，主模式(10-bit address mode, master mode) 0：主器件工作在7位寻址模式下 1：主器件工作在10位寻址模式下 注意：START位置1时，不允许更改此位。
10 RD_WRN	传输方向，主模式(Transfer direction, master mode) 0：主器件请求写传输。 1：主器件请求读传输。 注意：START位置1时，不允许更改此位。
9-8 SADD[9:8]	从地址9：8，主模式(Slave address bit9:8, master mode) 在7位寻址模式 (ADD10=0) 下：这些位无意义 在10位寻址模式 (ADD10=1) 下：这些位应写入待发送从地址的第8和第9位 注意：START位置1时，不允许更改这些位。
7-1 SADD[7:1]	从地址7：1，主模式(Slave address bit7:1, master mode) 在7位寻址模式 (ADD10=0) 下：这些位应写入待发送的7位从地址。 在10位寻址模式 (ADD10=1) 下：这些位应写入待发送从地址的第1到第7位 注意：START位置1时，不允许更改这些位。
0 SADD[0]	从地址位0，主模式(Slave address bit0, master mode) 在7位寻址模式 (ADD10=0) 下：该位无意义 在10位寻址模式 (ADD10=1) 下：该位应写入待发送从地址的第0位 注意：START位置1时，不允许更改这些位。

19.4.5. I2C 自身地址 1 寄存器（OA1）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	-	-	-	-	OA1MODE	OA1[9:8]	OA1[7:1]								OA1[0]
rw	-	-	-	-	rw	rw	Rw								rw

位域	功能
31-16	保留
15 OA1EN	设备自身地址 1 使能(Own Address 1 enable) 0: 禁止设备自身地址 1。不对接收的从地址 OA1 应答。 1: 使能设备自身地址 1。对接收的从地址 OA1 应答。
14-11	保留
10 OA1MODE	设备自身地址 1 10 位模式(Own Address 1 10-bit mode) 0: 设备自身地址 1 为 7 位地址。 1: 设备自身地址 1 为 10 位地址。 <i>注意: 仅可在 OA1EN=0 时写入该位。</i>
9-8 OA1[9:8]	接口地址 9: 8 位(Interface address) 7 位寻址模式: 无关 10 位寻址模式: 地址位 9:8 <i>注意: 仅可在 OA1EN=0 时写入这些位。</i>
7-1 OA1[7:1]	接口地址 7: 1 位(Interface address) 地址位 7:1 <i>注意: 仅可在 OA1EN=0 时写入这些位。</i>
0 OA1[0]	接口地址位 0(Interface address) 7 位寻址模式: 无关 10 位寻址模式: 地址位 0 <i>注意: 仅可在 OA1EN=0 时写入该位。</i>

19.4.6. I2C 自身地址 2 寄存器（OA2）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	-	-	-	-	OA2MSK			OA2							
rw	-	-	-	-	rw			rw							

位域	功能
31-16	保留
15 OA2EN	设备自身地址 2 使能(Own Address 1 enable) 0: 禁止设备自身地址 2。不对接收的从地址 OA2 应答。 1: 使能设备自身地址 2。对接收的从地址 OA2 应答。
14-11	保留
10-8 OA2MSK	设备自身地址 2 屏蔽位(Own Address 2 masks) 000: 无屏蔽 001: OA2[1] 被屏蔽, 为无关位。仅比较 OA2[7:2]。 010: OA2[2:1] 被屏蔽, 为无关位。仅比较 OA2[7:3]。 011: OA2[3:1] 被屏蔽, 为无关位。仅比较 OA2[7:4]。 100: OA2[4:1] 被屏蔽, 为无关位。仅比较 OA2[7:5]。 101: OA2[5:1] 被屏蔽, 为无关位。仅比较 OA2[7:6]。 110: OA2[6:1] 被屏蔽, 为无关位。仅比较 OA2[7]。 111: OA2[7:1] 被屏蔽, 为无关位。不进行比较, 对接收到的全部 7 位地址 (保留位除外) 应答。 <i>注意: 仅可在 OA2EN=0 时写入这些位。</i> <i>只要 OA2MSK 不等于 0, 即使比较匹配, 也不会对保留的 I2C 地址 (0b0000xxx 和 0b1111xxx) 应答。</i>
7-1 OA2	接口地址 7: 1 位(Interface address) 地址位 7:1 <i>注意: 仅可在 OA2EN=0 时写入这些位。</i>
0	保留

19.4.7. I2C 时序寄存器 (TIMINGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC				-	-	-	-	SCLDEL				SDADEL			
rw				-	-	-	-	rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH								SCLL							
rw								rw							

位域	功能
31-28 PRESC	时钟预分频因子(Timing prescaler) 该字段用于对 I2CCLK 进行预分频, 以生成用于数据建立和保持计数器 (本章 “I2C 时序”) 以 SCL 高电平和低电平计数器 (请参见本章 “I2C 主模式初始化”) 的时钟周期 t _{PRESC} 。 $t_{PRESC} = (PRESC + 1) * t_{I2CCLK}$
27-24	保留
23-20 SCLDEL	数据建立时间(Data setup time) 该字段用于在 SDA 边沿和 SCL 上升沿之间生成延时 t _{SCLDEL} 。在主模式和从模式下, 如果 NOSTRETCH = 0, 则 SCL 线的低电平时间将在 t _{SCLDEL} 期间延长。 $t_{SCLDEL} = (SCLDEL + 1) * t_{PRESC}$ <i>注: t_{SCLDEL} 用于生成 t_{SU:DAT} 时序。</i>
19-16	数据保持时间(Data hold time)

位域	功能
SDADEL	该字段用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL} 。在主模式和从模式下，如果 $NOSTRETCH = 0$ ，则 SCL 线的低电平时间将在 t_{SCDAEL} 期间延长。 $t_{SDADEL} = SDADEL * t_{PRESC}$ <i>注意：SDADEL 用于生成 $t_{HD:DAT}$ 时序。</i>
15-8 SCLH	SCL 高电平周期，主模式(SCL high period,master) 在主模式下，该字段用于生成 $t_{SU:STO}$ 成 SCL 高电平周期。 $t_{SCLH} = (SCLH+1) * t_{PRESC}$ <i>注意：SCLH 还用于生成 $t_{SU:STO}$ 和 $t_{HD:STA}$ 时序。</i>
7-0 SCLL	SCL 低电平周期，主模式(SCL low period,master) 在主模式下，该字段用于生成 SCL 低电平周期。 $t_{SCLL} = (SCLL+1) * t_{PRESC}$ <i>注意：SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。</i>

19.4.8. I2C 超时寄存器 (TIMEOUTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	-	-	-	TIMOUTB											
rw	-	-	-	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUTEN	-	-	TIDLE	TIMEOUTA											
rw	-	-	rw	rw											

位域	功能
31 TEXTEN	时钟延展超时使能(Extended clock timeout enable) 0: 禁止时钟信号延展超时检测 1: 使能时钟信号延展超时检测 当 I2C 接口执行 SCL 延展的累积时间超过 $t_{LOW:EXT}$ 时,将检测到超时错误 (TIMEOUT=1)。
30-28	保留
27-16 TIMEOUTB	总线超时 B (Bus timeout B) 该字段用于配置累积时钟延展超时: 在主模式下，将检测主器件的累积时钟低电平延展时间 ($t_{LOW:MEXT}$) 在从模式下，将检测从器件的累积时钟低电平延展时间 ($t_{LOW:SEXT}$) $t_{LOW:EXT} = (TIMEOUTB+1) * 2048 * t_{I2CCLK}$ <i>注意：仅可在 TEXTEN=0 时写入这些位。</i>
15 TIMEOUTEN	时钟超时使能(Clock timeout enable) 0: 禁止 SCL 超时检测 1: 使能 SCL 超时检测: 当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE=0)，或 SCL 的高电平时间超过 t_{IDLE} (TIDLE=1) 时，将检测到超时错误 (TIMEOUT=1)。
14-13	保留
12	空闲时在超时检测(Idle clock timeout detection)

位域	功能
TIDLE	0: TIMEOUTA 用于检测 SCL 低电平超时 1: TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件） 注意：仅可在 TIMOUTEN=0 时写入该位。
11-0 TIMEOUTA	总线超时 A (Bus timeout A) 该字段用于配置： - SCL 低电平超时条件 t_{TIMEOUT}（当 TIDLE=0 时） $t_{\text{TIMEOUT}} = (\text{TIMEOUTA} + 1) * 2048 * t_{\text{I2CCLK}}$ - 总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE=1 时） $t_{\text{IDLE}} = (\text{TIMEOUTA} + 1) * 4 * t_{\text{I2CCLK}}$ 注意：仅可在 TIMOUTEN=0 时写入这些位。

19.4.9. I2C 中断和状态寄存器（ISR）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DIR
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	-	ALERT	TIMEOUT	PECERR	OVR	ARLO	BEER	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r	-	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	功能
31-24	保留
23-17 ADDCODE	地址匹配码，从模式(Address match code) 发生地址匹配事件时 (ADDR = 1)，这些位更新为接收到的地址。 在 10 位地址的情况下，ADDCODE 提供 10 位地址的头字节，后跟地址的 2 个 MSB。
16 DIR	传输方向，从模式(Transfer direction, Slave mode) 该标志在发生地址匹配事件时 (ADDR=1) 更新。 0: 写传输，从器件进入接收器模式。 1: 读传输，从器件进入发送器模式。
15 BUSY	总线忙标识(Bus busy) 该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置 1。当检测到停止位或 PE = 0 时，该位由硬件清零。
14	保留
13 ALERT	SMBus 报警(SMBus alert) 当 SMBHEN=1（SMBus 主机配置）、ALERTEN=1 且在 SMBA 脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。 注：当 PE=0 时，该位由硬件清零。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。
12 TIMEOUT	超时或 t_{LOW} 检测标志(Timeout detection flag) 发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。

位域	功能
	注：当 $PE=0$ 时，该位由硬件清零。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。
11 PECERR	接收期间的 PEC 错误(PEC error in detection) 当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该标志由软件清零，方法是将 PECF 位置 1。 注：当 $PE=0$ 时，该位由硬件清零。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。
10 OVR	上溢/下溢，从模式(Overflow/Underflow, Slave mode) 在从模式下且 NOSTRETCH=1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该标志由软件清零，方法是将 OVRF 位置 1。 注意：当 $PE=0$ 时，该位由硬件清零。
9 ARLO	仲裁丢失(Arbitration lost) 发生仲裁丢失时，该标志由硬件置 1。该标志由软件清零，方法是将 ARLOF 位置 1。 注：当 $PE=0$ 时，该位由硬件清零。
8 BERR	总线错误(Bus error) 当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该标志由软件清零，方法是将 BERRF 位置 1。 注：当 $PE=0$ 时，该位由硬件清零。
7 TCR	传输完成等待重载(Transfer Complete Reload) 当 RELOAD=1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。 注：当 $PE=0$ 时，该位由硬件清零。 该标志单独用于主模式，SBC 位置 1 时单独用于从模式。
6 TC	传输完成，主模式(Transfer Complete, Master mode) 当 RELOAD=0、AUTOEND=0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。 注意：当 $PE=0$ 时，该位由硬件清零。
5 STOPF	停止位检测标志(Stop detection flag) 当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1： – 外设作为主器件，该位置位的前提是外设已经发出停止位。 – 外设作为从器件，该位置位的前提条件是此次传输的寻址对象就是该外设。该标志由软件清零，方法是将 STOPCF 位置 1。 注意：当 $PE=0$ 时，该位由硬件清零。
4 NACKF	接收到否定应答标志(Not Acknowledge received flag) 传输完字节后接收到 NACK 时，该标志由硬件置 1。该标志由软件清零，方法是将 NACKCF 位置 1。 注意：当 $PE=0$ 时，该位由硬件清零。
3 ADDR	地址匹配(Address matched, slave only) 接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 ADDRCF 位置 1。 注意：当 $PE=0$ 时，该位由硬件清零。
2 RXNE	接收数据寄存器不为空，接收器(Receive data register not empty, receivers) 当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。

位域	功能
	可以向该位写入“1”以生成 RXNE 事件 <i>注意：当 PE=0 时，该位由硬件清零。</i>
1 TXIS	发送中断状态，发送器(Transmit interrupt status,Transmitters) 当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。 该位只能在 NOSTRETCH=1 时由软件写入“1”，以生成 TXIS 事件（TXIE=1 时为中断）。 可以向该位写入“0”以清除 TXIS 标志 <i>注意：当 PE=0 时，该位由硬件清零。</i>
0 TXE	发送数据寄存器为空(Transmit data register empty,Transmitters) 当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。 该位可由软件写入“1”，以刷新发送数据寄存器 I2C_TXDR。 <i>注：当 PE=0 时，该位由硬件置 1。</i>

19.4.10. I2C 中断清零寄存器（ICR）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	-	-	STOPCF	NACKF	ADDRCF	-	-	-
-	-	w	w	w	w	w	w	-	-	w	w	w	-	-	-

位域	功能
31-14	保留
13 ALERTCF	SMBus 报警标志位清零(SMBus alert flag clear) 将 1 写入此位时，I2C_ISR 寄存器中的 ALERT 标志将清零。 <i>注：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。</i>
12 TIMOUTCF	SMBus 超时检测标志位清零(Timeout detection flag clear) 将 1 写入此位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。 <i>注：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。</i>
11 PECCF	PEC 错误检测标志位清零(PEC error detection flag clear) 将 1 写入此位时，I2C_ISR 寄存器中的 PECERR 标志将清零。 <i>注：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。</i>
10 OVRCF	上溢/下溢标志位清零(Overflow/Underflow flag clear) 将 1 写入此位时，I2C_ISR 寄存器中的 OVR 标志将清零。
9 ARLOCF	仲裁丢失标志位清零(Arbitration lost flag clear) 将 1 写入此位时，I2C_ISR 寄存器中的 ARLO 标志将清零。
8 BERRCF	总线错误标志位清零(Bus error flag clear) 将 1 写入此位时，I2C_ISR 寄存器中的 BERRF 标志将清零。位 7:6 保留，必须保持复位值。
7-6	保留

位域	功能
5 STOPCF	停止条件检测标志位清零(Stop detection flag clear) 将 1 写入此位时, I2C_ISR 寄存器中的 STOPF 标志将清零。
4 NACKCF	NACK 接收标志位清零(Not Acknowledge received flag clear) 将 1 写入此位时, I2C_ISR 寄存器中的 ACKF 标志将清零。
3 ADDRCF	地址匹配标志位清零(Address matched flag clear) 将 1 写入此位时, I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入此位时, I2C_CR2 寄存器中的 START 位也将清零。
2-0	保留

19.4.11. I2C PEC 寄存器 (PECR)

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

位域	功能
31-8	保留
7-0 PEC	数据包错误校验寄存器(Packet error checking register) 当 PECEN=1 时, 此字段包含内部 PEC。当 PE=0 时, PEC 由硬件清零。 <i>注意: 如果不支持 SMBus 功能, 该寄存器保留, 并由硬件强制为 “0x00000000”。</i>

19.4.12. I2C 接收数据寄存器 (RXDR)

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

位域	功能
31-8	保留
7-0 RXDATA	8 位接收数据寄存器(8-bit receive data) 从 I2C 总线接收的数据字节。

19.4.13. I2C 发送数据寄存器 (TXDR)

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

位域	功能
31-8	保留
7-0 TXDATA	8 位发送数据寄存器(8-bit transmit data) 待发送到 I2C 总线的数据字节。 <i>注意：仅可在 TXE=1 时写入这些位。</i>

20. 串行外设接口 (SPI)

20.1. 简介

SPI 模块可以通过 SPI 协议与外部设备进行通信。

串行外设接口提供了基于 SPI 协议的全双工和单工串行通信功能，支持工作于主机或从机模式。接口还能以多主配置方式工作。

当工作在从机只接收模式时，接口支持接收到匹配的数据后将系统从低功耗模式唤醒。

20.2. 主要特性

- 支持 SPI Motorola 模式，SPI TI 模式
- 主机、从机可配
- 主机波特率可配，支持 2-256 的偶数分频
- 支持全双工、单工模式
- 支持多主机模式
- 8~32 位数据帧长度选择
- 独立的发送 FIFO 和接收 FIFO，大小为 4 字，
当数据帧等于 8 位时，FIFO 深度自动配置为 16；
当数据帧为 9~16 位时，FIFO 深度自动配置为 8；
当数据帧为 17~32 位时，FIFO 深度自动配置为 4
- 可编程的数据收发的字节序，低字节在前或高字节在前
- 可编程的时钟极性和相位
- 可编程的数据位顺序，低位在前或高位在前
- 支持软件或硬件 NSS 管理
- 支持停止模式下数据匹配唤醒功能
- 支持 FIFO 的空满中断以及达到水位线中断
- 可触发中断的从模式下的帧错误标志
- 可触发中断的多主模式下的主模式故障标志

20.3. 功能说明

20.3.1. SPI 时序与数据帧格式

20.3.1.1. 时钟相位和极性控制

通过 CR2[CPOL]和 CR2[CPHA]可以用软件选择四种可能的时序关系。CPOL（时钟极性）位决定了空闲状态时 SCK 的电平，CPHA（时钟相位）位决定了在 SCK 的第一个边沿还是第二个边沿采样数据。

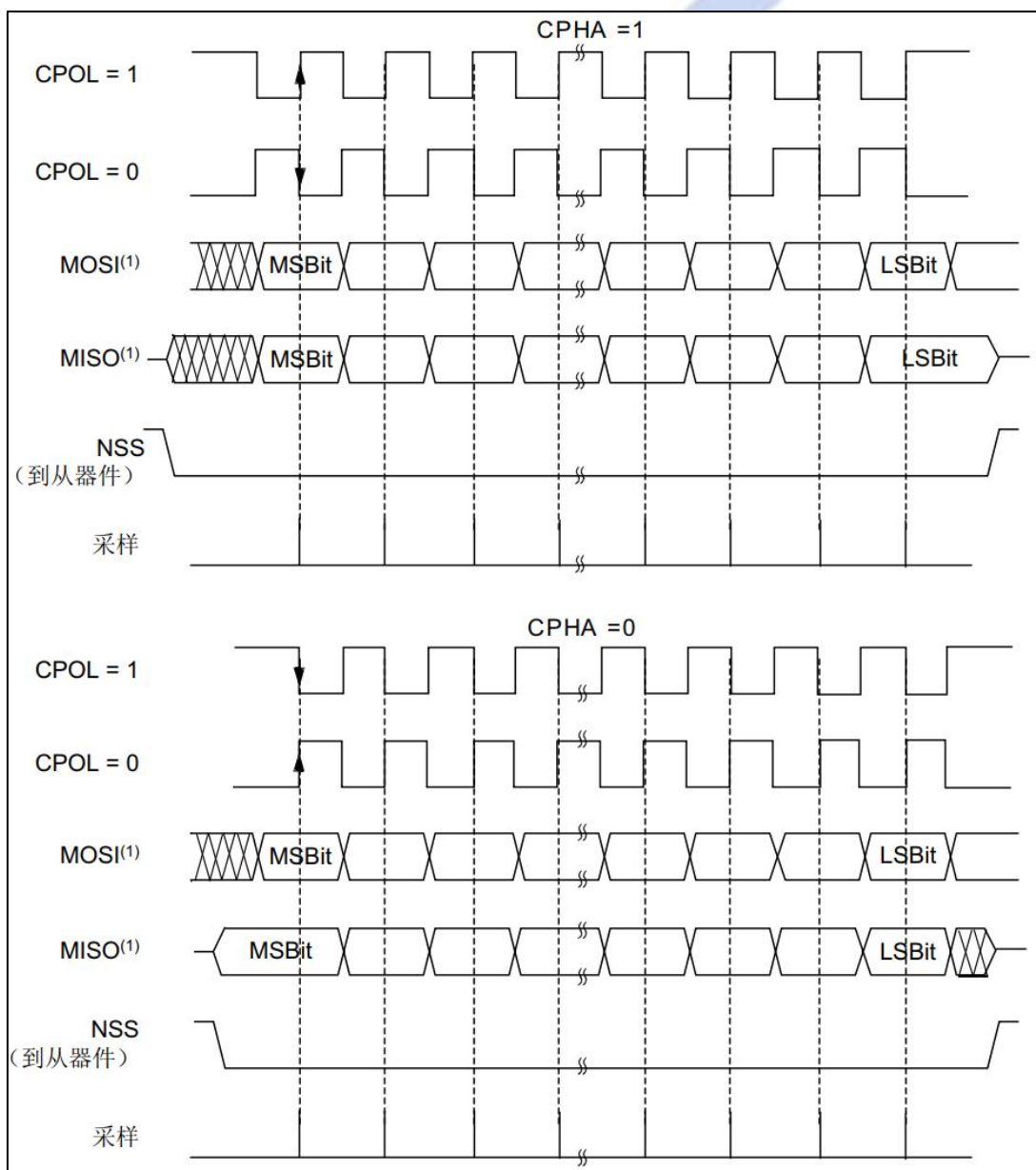


图 20-1 SPI 时钟相位和极性

20.3.1.2.数据帧格式

SPI 可通过 CR2[LSBFIRST]位配置数据顺序，当 CR2[LSBFIRST]=1 时，SPI 先发送 LSB，当 CR2[LSBFIRST]=0 时，SPI 先发送 MSB。

20.3.2. SPI 配置流程

20.3.2.1.SPI 初始化流程

在发送或者接收数据之前，应用程序应遵循如下的 SPI 初始化流程：

1. 配置 CR2[FCLKSEL] 选择 PCLK，并置位 CR2[FCLKEN]打开时钟；
2. 置位 RSTR[GRST]复位模块，确保上一次通讯之后的 FIFO 指针复位；
3. 配置主从模式 CR1[MSTR]，传输模式 CR1[TMOD]，NSS 片选空闲时电平 CR1[SSPOL]，工作模式 CR1[FRF]；
4. 配置时钟极性 CR2[CPOL]，时钟相位 CR2[CPHA]，帧格式 CR2[LSBFIRST]，字节序 CR2[REORDER]，帧长度 CR2[LENGTH]；
5. 工作在主机时，配置 SCK 时钟分频 DIVR[SCKDIV]，以产生需要的波特率，SPI 波特率=APB 时钟 /((DIVR[SCKDIV] + 1)*2)，从机模式可忽略此步骤；
6. 置位 ENR[SPE]打开 SPI 外设。

20.3.2.2.SPI 基本发送流程

SPI 在完成初始化流程后，SPI 保持在空闲状态。在主机模式下，当写入一个数据到发送缓冲区时，发送过程开始。在从机模式下，当 SCK 引脚上的 SCK 信号开始翻转，且 NSS 引脚为低电平，发送开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入到发送缓冲区中。

当 SPI 开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在此同时，TXE（发送缓冲区空）位置 1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写 TDR 寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入 TDR 寄存器中。当 SR[TXF]=1 时，不能再向发送缓冲区写入数据。

20.3.2.3.SPI 基本接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入接收缓冲区，且 SR[RXNE]（接收缓冲区非空）位置 1。软件通过读 RDR 寄存器获得接收到的数据，此操作会自动清除 SR[RXNE]位。

20.3.3. 多主机模式

在主机模式下（CR1[MSTR]=1）将 CR2[MMSEL]位置 1 将开启多主机模式，此时 NSS 信号作为输入，可通过 NSS 引脚（CR2[SSM]=0）或者通过软件模式（CR2[SSM]=1）来获得。如果正常通讯中检测到 NSS 引脚电平有效（根据 CR1[SSPOL]极性配置）或者在软件模式（CR2[SSM]=1）下 CR2[SSI]=0，就会产生多主机故障。如果使能多主机故障中断 IER[MMFIE]=1，将触发多主机故障中断。

若产生多主机故障，硬件会自动关闭 SPI 外设（ENR[SPE]=0），随后软件需要清除多主机故障标志，然后再把 SPI 外设配置成从机模式后重新打开外设。

多主机模式只有在 Motorola 模式下才有效，TI 模式无效。

20.3.4. 多从机模式

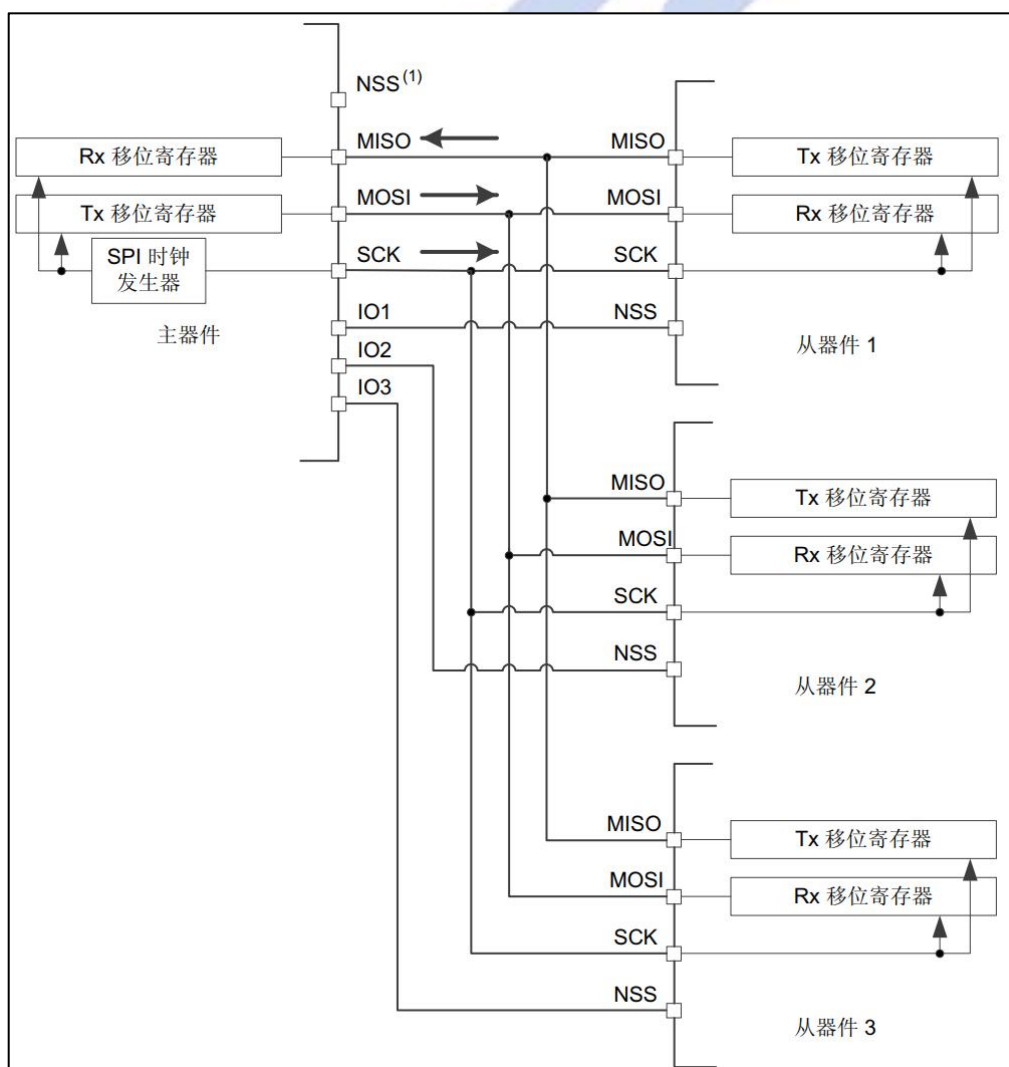


图 20-2 多从机模式

1. NSS 引脚在主机侧不使用。在此模式下必须配置 SSM=1，SSI=1 以防止多主机故障错误。

20.3.5. NSS 功能

从机模式：

1. 在硬件 NSS 模式 ($CR1[SSM]=0$) 下, NSS 引脚获取 NSS 信号;
2. 在软件 NSS 模式 ($CR1[SSM]=1$) 下, SPI 根据 $CR1[SSI]$ 位得到 NSS 信号, 此时 NSS 引脚空闲, 可供其它应用使用。

主机模式：

1. 使用多主机模式时 ($CR2[MMSEL]=1$), NSS 可以配置为硬件输入模式 ($CR1[SSM]=0$) 或者软件模式 ($CR1[SSM]=1$), 一旦 NSS 引脚 (在硬件输入模式) 或 $CR1[SSI]$ 位 (软件模式) 变为非空闲 (根据 $CR1[SSPOL]$ 极性配置), SPI 将自动进入从机模式, 并且置位多主机错误标志 $SR[MMF]$;
2. 不使用多主机模式时 ($CR2[MMSEL]=0$), 可将 NSS 配置为硬件输出模式, NSS 信号将在 NSS 引脚输出, 也可将 NSS 配置为软件模式, 可使用 GPIO 作为 NSS 引脚, 以实现更加灵活的应用。

20.3.6. TI 模式

SPI TI 模式将 NSS 作为一种特殊的帧头标志信号, 它的操作流程与常规的 Motorola 模式类似, 主要的区别是, 在 TI 模式中, 每一个数据帧的第一个时钟用于传输 NSS 脉冲, 时钟极性 $CR2[CPOL]$ 和时钟相位 $CR2[CPHA]$ 是没有意义的, TI 模式采用 SCK 的下降沿采样。

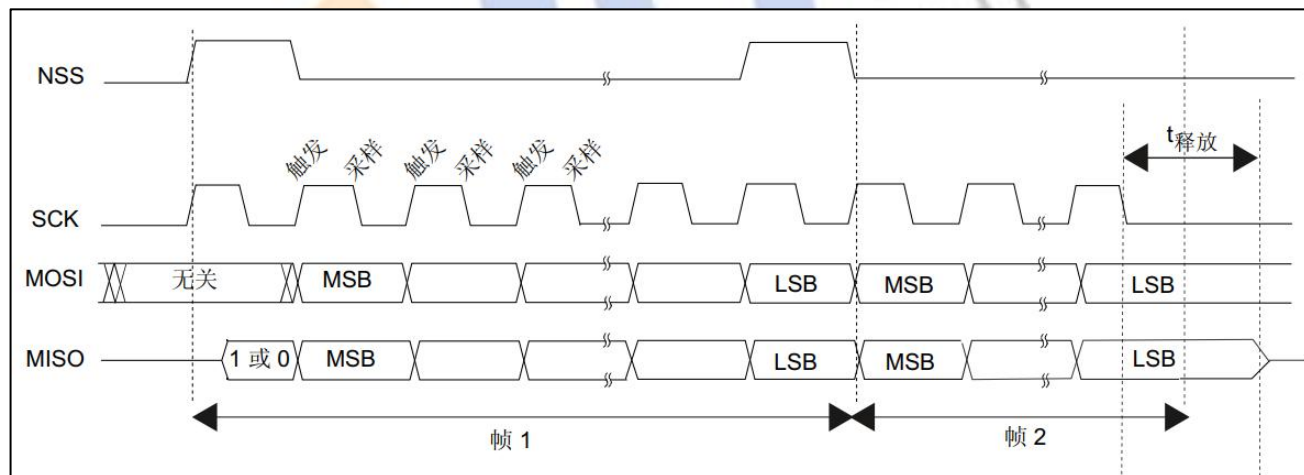


图 20-3 TI 模式传输

20.3.7. 匹配唤醒

当 SPI 工作在从机只接收模式下, 系统进入低功耗停止模式后, 如果接收到和 DMR 数据匹配寄存器相同的数据帧, 则会产生匹配唤醒中断唤醒系统工作。

20.4. SPI 标志与中断

表 20-1 SPI 中断请求

状态位	描述	中断	清中断	停止模式唤醒
TXE	发送 FIFO 空	是	硬件	无
TXF	发送 FIFO 满	是	硬件	无
TXWR	发送 FIFO 容量低于水位线	是	硬件	无
RXE	接收 FIFO 非空	是	硬件	无
RXF	接收 FIFO 满	是	硬件	无
RXWR	接收 FIFO 剩余容量低于水位线	是	硬件	无
RXOVER	接收溢出	是	写 1	无
MMF	多主机故障	是	写 1	无
FERR	帧错误	是	写 1	无
MAWU	匹配唤醒	是	写 1	是
FEND	帧结束	是	写 1	无
BUSY	忙状态	否	无	无

20.5. SPI 寄存器

20.5.1. SPI 寄存器映射

表 20-2 SPI 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	SPI IP 版本号 (IP_VERSION)	r	0x0000_0005
0x04	SPI 控制寄存器 1 (CR1)	rw	0x0000_0000
0x08	SPI 控制寄存器 2 (CR2)	rw	0x0000_0000
0x0C	SPI 中断使能寄存器 (IER)	rw	0x0000_0000
0x10	SPI 使能寄存器 (ENR)	rw	0x0000_0000
0x14	SPI 状态寄存器 (SR)	rc_wl	0x0000_0005
0x18	SPI 主时钟分频寄存器 (DIVR)	rw	0x0000_0000
0x20	SPI FIFO 控制寄存器 (FCR)	rw	0x0000_0000
0x24	SPI 复位寄存器 (RSTR)	w	0x0000_0000
0x28	SPI 数据匹配寄存器 (DMR)	rw	0x0000_0000
0x30	SPI 发送数据寄存器 (TDR)	w	0x0000_0000
0x40	SPI 接收数据寄存器 (RDR)	r	0x0000_0000

20.5.2. SPI IP 版本号 (IP_VERSION)

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

位域	功能
31-16	保留
15-8 MVER	主版本号
7-0 SVER	子版本号

20.5.3. SPI 控制寄存器 1 (CR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	FLTCL	WUEN	FRF	SSPOL	HDDIR	TMOD	MSTR		
-	-	-	-	-	-	-	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-9	保留
8 FLTCL	滤波关闭 (filter close)，只在从机或者多主机模式时有效。 0: 滤波打开 1: 滤波关闭 备注 1: 使能数据匹配唤醒时，滤波关闭；正常情况下滤波打开。 备注 2: 从机模式下，如果 $SCK \geq PCLK/4$ ，必须配置滤波关闭。
7 WUEN	数据匹配唤醒使能 (wake-up enable)，只在从机时有效。 0: 禁止数据匹配唤醒 1: 使能数据匹配唤醒
6-5 FRF	帧模式 (frame format) 00: Motorola SPI 模式 01: Motorola SPI 模式 10: TI 模式 11: 保留
4 SSPOL	片选极性 (slave select polarity)，主机从机都需要配置。 0: 空闲时保持高电平 1: 空闲时保持低电平 备注: 在 Motorola SPI 模式时，片选极性=0；在 TI 模式时，片选极性=1；在帧模式=0x 或 10 时，SSPOL 不需要配置。
3	保留
2-1 TMOD	传输模式 (transfer mode) 00: 全双工模式 01: 保留 10: 只接收模式 11: 只发送模式
0 MSTR	主模式选择 (master selection) 0: 从机 1: 主机

20.5.4. SPI 控制寄存器 2 (CR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	LENGTH					-	REORDER	LSBFIRST	MMSEL	SSM	SSI	CPOL	CPHA	FCLKSEL	FCLKEN
-	rw					-	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-15	保留
14-10 LENGTH	帧长度 (frame length) LENGTH=0x08 8bits LENGTH=0x09 9bits LENGTH=0x1F 31bits LENGTH=0x00 32bits 注: 如果 LENGTH=0x01 到 0x07, 还是会按照 LENGTH=8bits 进行传输
9	保留
8 REORDER	发送数据时的帧格式顺序选择 (byte reorder selection), 接收的数据顺序不作处理。 如果帧长度为 16bits, 0: 低 8 位, 高 8 位 1: 高 8 位, 低 8 位 如果帧长度为 24bits, 0: 低 8 位, 中 8 位, 高 8 位 1: 高 8 位, 中 8 位, 低 8 位 如果帧长度为 32bits, 0: 低 8 位, 中低 8 位, 中高 8 位, 高 8 位 1: 高 8 位, 中高 8 位, 中低 8 位, 低 8 位 帧长度为其他时, REORDER 无效
7 LSBFIRST	帧格式 (byte format) 0: MSB, 发送/接收数据时高位在前 1: LSB, 发送/接收数据时低位在前
6 MMSEL	多主机模式选择 (multi-master selection), 仅当 MSTR=1, 此位才有效。 0: 禁止多主机模式, NSS 作为输出 1: 允许多主机模式, NSS 作为输入, 也可以使用软件从器件管理来控制
5 SSM	软件从器件管理 (software slave management) 0: 禁止软件从器件管理 1: 使能软件从器件管理 备注 1: TI 模式下该位须为 0。 备注 2: 在 Motorola SPI 主机模式, 多从机工作环境下该位须为 1, NSS 由 SSI 控制。
4 SSI	内部从器件选择 (internal slave select) 仅当 SSM=1 时, 此位才有效。此位的值将作用到 NSS 引脚上, 并忽略 NSS 引脚的 I/O 值。 0: NSS=0 1: NSS=1

位域	功能
3 CPOL	时钟极性 (clock polarity) 0: 空闲状态时, SCK=0 1: 空闲状态时, SCK=1 只需要在 Motorola SPI 模式下配置; 其他模式有固定值。
2 CPHA	时钟相位 (clock phase) 0: 从第一个时钟边沿开始捕获数据 1: 从第二个时钟边沿开始捕获数据 只需要在 Motorola SPI 模式下配置; 其他模式有固定值。
1 FCLKSEL	FCLK 时钟选择 (functional clock selection) 0: FCLK=PCLK 1: 保留
0 FCLKEN	FCLK 时钟使能 (functional clock enable) 0: FCLK 关闭 1: FCLK 打开 备注: 通过 FCKSEL 切换时钟, 再打开 FCLKEN, 需要 2 个 CYCLE 的时钟稳定时间, 此时不建议配置 ENR 和 RSTR 寄存器。

20.5.5. SPI 中断使能寄存器 (IER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	FENDIE	-	MAWUIE	FERRIE	RXOVIE	RXWRIE	RXFIE	RXNEIE	MMFIE	TXWRIE	TXFIE	TXEIE
-	-	-	-	rw	-	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-12	保留
11 FENDIE	帧结束中断使能 (frame end interrupt enable) 0: 禁止 1: 允许
10	保留
9 MAWUIE	匹配唤醒中断使能 (match wake-up interrupt enable) 0: 禁止 1: 允许
8 FERRIE	帧错误中断使能 (Frame error interrupt enable) 0: 禁止 1: 允许
7 RXOVIE	接收溢出中断使能 (Receive overflow interrupt enable) 0: 禁止 1: 允许
6	接收水位线中断使能 (Receive above watermark interrupt enable)

位域	功能
RXWRIE	0: 禁止 1: 允许
5 RXFIE	接收 FIFO 满中断使能 (RX FIFO full interrupt enable) 0: 禁止 1: 允许
4 RXNEIE	接收 FIFO 非空中断使能 (RX FIFO not empty interrupt enable) 0: 禁止 1: 允许
3 MMFIE	多主机故障中断使能 (Multi-master fault interrupt enable) 0: 禁止 1: 允许
2 TXWRIE	发送水位线中断使能 (Transmit below watermark interrupt enable) 0: 禁止 1: 允许
1 TXFIE	发送 FIFO 满中断使能 (TX FIFO full interrupt enable) 0: 禁止 1: 允许
0 TXEIE	发送 FIFO 空中断使能 (TX FIFO empty interrupt enable) 0: 禁止 1: 允许

20.5.6. SPI 使能寄存器 (ENR)

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

位域	功能
31-1	保留
0 SPE	SPI 使能 (SPI enable) 0: 禁止外设 1: 使能外设 备注 1: 建议在 SR 寄存器中的 BUSY 位处于 0 时禁止外设。在传输未结束时禁止外设，需要在下一次传输前进行软复位，才能保证下一次传输正常。 备注 2: SPI 使能生效需要有 3 个稳定的 FCLK 周期，若从机匹配唤醒模式下，必须先在 FCLKEN 拉高的时候配置 SPE，然后稳定后才能关闭 FCLKEN。

20.5.7. SPI 状态寄存器 (SR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	FEND	BUSY	MAWU	FERR	RXOVER	RXWR	RXF	RXNE	MMF	TXWR	TXF	TXE
-	-	-	-	rc_wl	r	rc_wl	rc_wl	rc_wl	r	r	r	rc_wl	r	r	r

位域	功能
31-12	保留
11 FEND	帧结束 (Frame end flag) 0: 未结束或未传输状态 1: 帧结束状态 备注 1: TMOD = 00 时, 全双工模式, 帧结束为接收结束; TMOD=10 时, 只接收模式, 帧结束为接收结束; TMOD=11 时, 只发送模式, 帧结束为发送结束。 备注 2: 结束状态在下一帧开始后自动拉低, 如果没有下一帧, 结束状态保持高, 需要写 1 拉低该状态位。
10 BUSY	忙状态 (BUSY status) 0: 外设处于空闲状态 1: 外设处于忙状态
9 MAWU	匹配唤醒中断标志 (Match wake-up interrupt flag) 0: 无匹配唤醒 1: 有匹配唤醒
8 FERR	帧错误中断标志 (Frame error interrupt flag) 0: 未发生帧错误 1: 发生帧错误
7 RXOVER	接收溢出中断标志 (Receive overflow interrupt flag) 0: 未发生接收溢出 1: 发生接收溢出, 后续接收数据被丢弃
6 RXWR	接收水位线中断标志 (Receive above watermark interrupt flag) 0: 接收 FIFO 剩余容量未达到水位线 1: 接收 FIFO 剩余容量低于水位线
5 RXF	接收 FIFO 满中断标志 (RX FIFO full interrupt flag) 0: 接收 FIFO 未滿 1: 接收 FIFO 满
4 RXNE	接收 FIFO 非空中断标志 (RX FIFO not empty interrupt flag) 0: 接收 FIFO 空 1: 接收 FIFO 非空
3 MMF	多主机故障中断标志 (Multi-master fault interrupt flag) 0: 未发生多主机故障 1: 发生多主机故障
2 TXWR	发送低于水位线中断标志 (Transmit below watermark interrupt flag) 0: 发送 FIFO 容量未达到水位线 1: 发送 FIFO 容量低于水位线

位域	功能
1 TXF	发送 FIFO 满中断标志 (TX FIFO full interrupt flag) 0: 发送 FIFO 未 1: 发送 FIFO 满
0 TXE	发送 FIFO 空中断标志 (TX FIFO empty interrupt flag) 0: 发送 FIFO 非空 1: 发送 FIFO 空

20.5.8. SPI 主时钟分频寄存器 (DIVR)

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

位域	功能
31-7	保留
6-0 SCKDIV	主机 SCK 分频系数 (SCK divider) $f_{SCK} = f_{CLK}/SCKDIV$; 只支持偶数分频。 SCKDIV=0, 2 分频; SCKDIV=1, 4 分频; SCKDIV=2, 6 分频; SCKDIV=126, 254 分频 SCKDIV=127, 256 分频 最大支持 256 分频。

20.5.9. SPI FIFO 控制寄存器 (FCR)

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

位域	功能
31-8	保留
7-4	接收 FIFO 水位线 (receive FIFO watermark)

位域	功能
RX_WATER	若接收 FIFO 的剩余容量（FIFO 总深度-当前有效数据个数）小于等于接收 FIFO 水位线，则接收到达水位线标志置 1。
3-0 TX_WATER	发送 FIFO 水位线（transmit FIFO watermark） 若发送 FIFO 中的有效数据个数小于等于发送 FIFO 水位线，则发送到达水位线标志置 1。

20.5.10. SPI 复位寄存器 (RSTR)

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

位域	功能
31-3	保留
2 RRST	接收模块复位（rx part reset） 0：无 1：接收模块及 FIFO 复位
1 TRST	发送模块复位（tx part reset） 0：无 1：发送模块及 FIFO 复位
0 GRST	全模块复位（global reset） 0：无 1：全模块复位

20.5.11. SPI 数据匹配寄存器 (DMR)

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

位域	功能
31-0 MATCH	数据匹配值，若帧长度为 8bits，则[7:0]有效，若帧长度为 16bits,则[15:0]有效；以此类推。

20.5.12. SPI 发送数据寄存器 (TDR)

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

位域	功能
31-0 TXDATA	发送的数据，有效位由帧长度决定，因此必须在帧长度配置完成后才能写该寄存器。

20.5.13. SPI 接收数据寄存器 (RDR)

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

位域	功能
31-0 RXDATA	接收的数据，有效位由帧长度决定，因此必须在帧长度配置完成后才能读该寄存器。

21. 通用同步异步收发器(USART)

21.1. 简介

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器实现了多种波特率。

USART 支持半双工单线通信，以及 LIN（局域互连网络）、IrDA（红外线数据协会），还支持多处理器通信。

21.2. 主要特性

- 全双工异步通信
- 采用 NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样
- 48MHz 时钟频率和 8 倍过采样时，通用可编程收发波特率最高为 6Mbps
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 单线半双工通信
- 支持 8 bytes 深度发送接收 FIFO
- 发射器和接收器有单独的使能位
- 支持发送和接收的信号极性翻转
- 通信控制/错误检测标志
- 奇偶校验控制：
 - 发送奇偶校验位
 - 检查接收的数据字节的奇偶性
- 十七个具有标志位的中断源
- 支持多处理器通信，最高 9 位地址模式
- 从静默模式唤醒（通过空闲线检测或地址标记检测）
- LIN 主模式同步断路发送功能和 LIN 从模式断路检测功能
 - 当 USART 被配置为 LIN 使用时，可生成 13 位断路
- 支持 3/16 位持续时间的 IrDASIR 编解码器
- 支持 ModBus 通信
 - 超时功能
 - CR/LF 字符识别

21.3. 功能说明

21.3.1. USART 框图

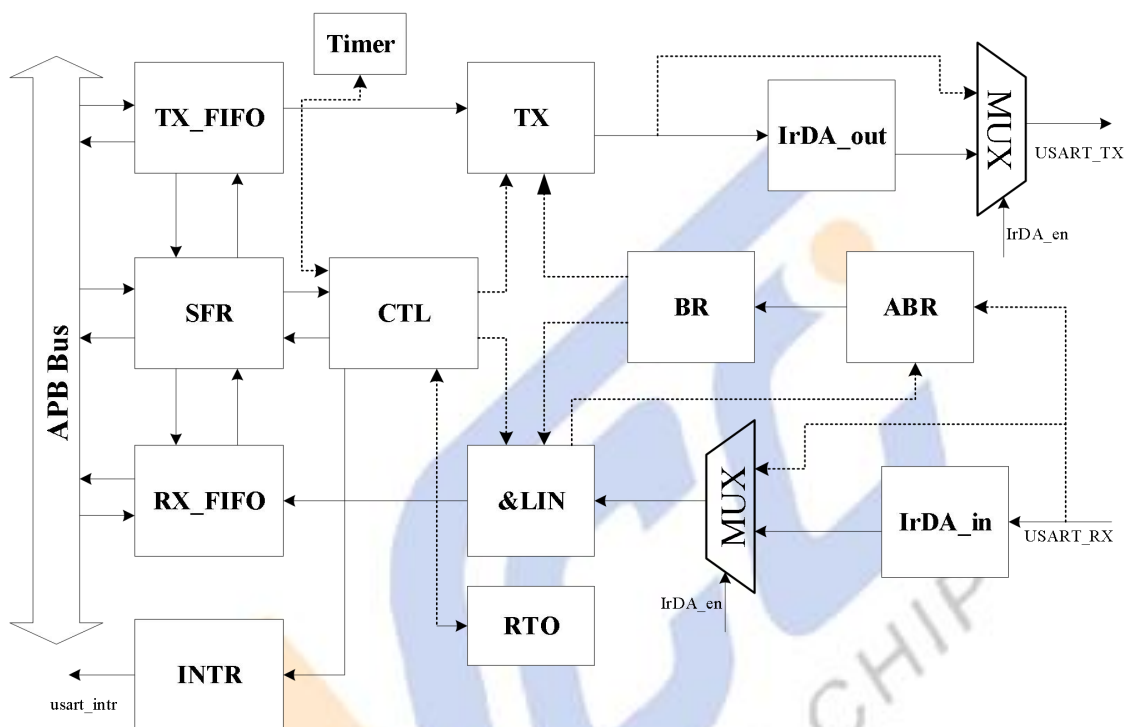


图 21-1 USART 框图

21.3.2. 发送与接收字符

USART 的每一帧数据由一个起始位、 n 个数据位、1 个校验位（可能有）、 m 个停止位构成（特殊的 LIN 断路和应答除外）；其中奇偶校验位通过 CTL[PCE]使能和 CTL[PS]位选择奇校验或者是偶校验。数据位长度通过 CTL[ML]来设置，停止位也可以通过 CTL[STOP]来设置。需要注意在不同的功能下，数据位和停止位的长度必须满足相应的协议规定。

表 21-1 帧格式

M 位	PCE 位	USART 帧
00	0	SB 8 位数据 STB
00	1	SB 7 位数据 PB STB
01	0	SB 9 位数据 STB
01	1	SB 8 位数据 PB STB
10	0	SB 7 位数据 STB
10	1	SB 6 位数据 PB STB

表格注释：

- SB：起始位；STB：停止位；P：奇偶校验位；
- 在数据寄存器中，PB 始终位于 MSB 位置（第 9 位、第 8 位或第 7 位，具体取决于 M 位的值）。

其中 STOP，停止位(STOP bits)，停止位选择：

- 00：1 个停止位
- 10：2 个停止位

通过极性配置控制，可以单独针对每个信号对这些值取反。发送时可以发送特殊的空闲字符和停止字符。空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”），空闲字符主要用于某些模式发送中途需要插入等待字符；中断帧字符可理解为在一个帧周期内接收到的电平均为“0”，发送器在中断帧的末尾插入 2 个停止位。

注意：7 位数据长度模式下，不支持 LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

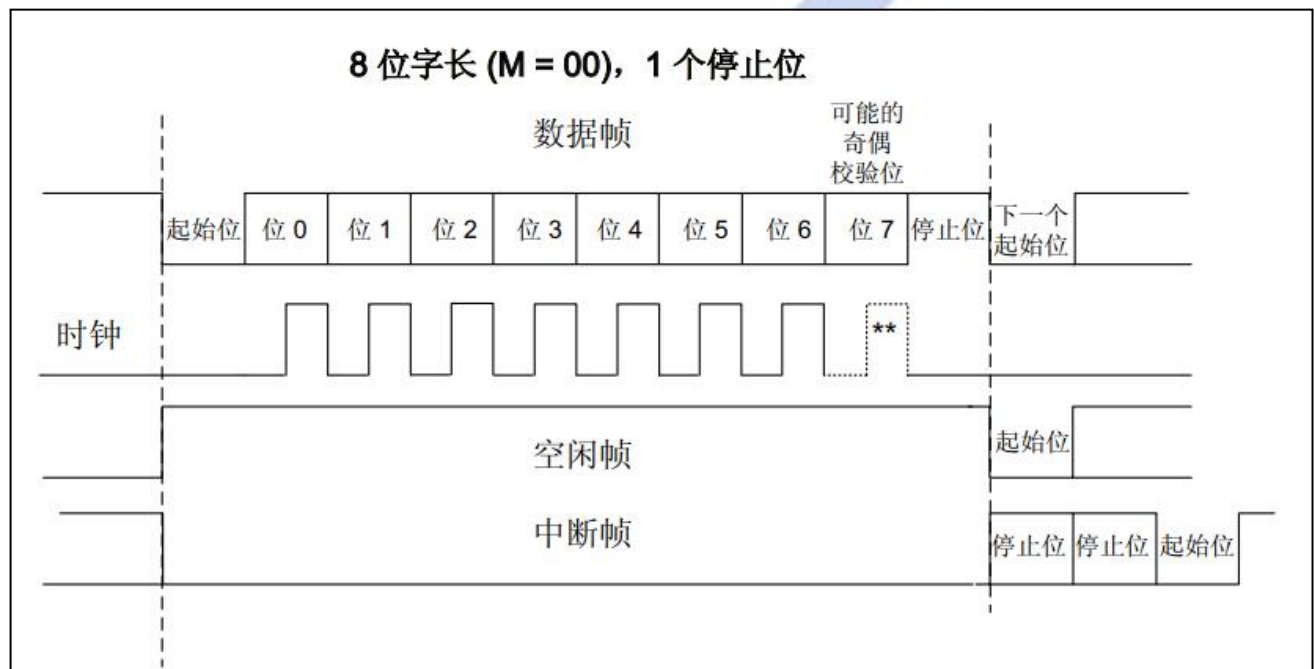


图 21-2 USART 帧数据时序图

21.3.3. 过采样

本设计支持 16 倍或 8 倍过采样，用于起始位序列检测和数据采样。在 USART 中，可根据 USART_CTL 寄存器中的 START_FLT 位来选择起始位是否滤波；当选择滤波功能后，检测起始位需要识别出特定序列。此序列为：1110X0X0X0X0X0。

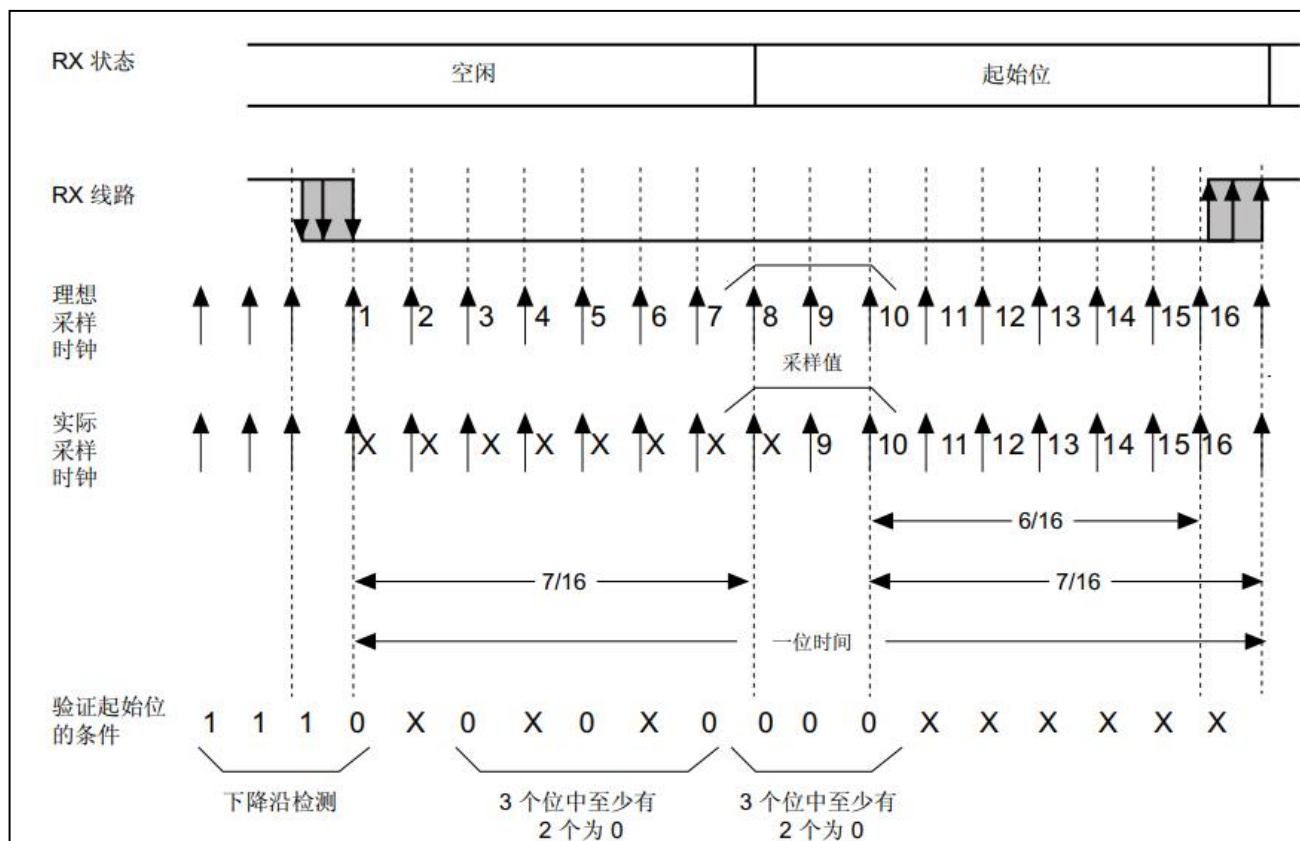


图 21-3 16 倍或 8 倍过采样时的起始位检测

如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。如果两次的 3 个采样位均为 0（针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为 0；针对第 8 位、第 9 位和第 10 位进行第二次采样时检测到这 3 位均为 0），可确认起始位。满足以下条件时，可验证起始位但 NF 噪声标志位置 1，如果条件 a 或条件 b 都不满足，则启动检测中止，接收器返回空闲状态。

- 对于两次采样，3 个采样位中有 2 位为 0（针对第 3 位、第 5 位和第 7 位进行采样；针对第 8 位、第 9 位和第 10 位采样）
- 如果其中一次采样时（对第 3 位、第 5 位和第 7 位进行采样或对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为 0。

可通过编程 CTL[OVER8]位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍。根据应用：

- 选择 8 倍过采样(OVER8=1)以获得更高的速度（高达 $f_{CK}/8$ ）。这种情况下接收器对时钟偏差的最大容差将会降低。
- 选择 16 倍过采样(OVER8=0)以增加接收器对时钟偏差的容差。这种情况下，最大速度限制为最高 $f_{CK}/16$ ，其中 f_{CK} 为时钟源频率。

可通过编程 CTL[ONEBIT]位选择用于评估逻辑电平的方法。有两种选择：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NF 位置 1。
- 在已接收位的中心进行单次采样。

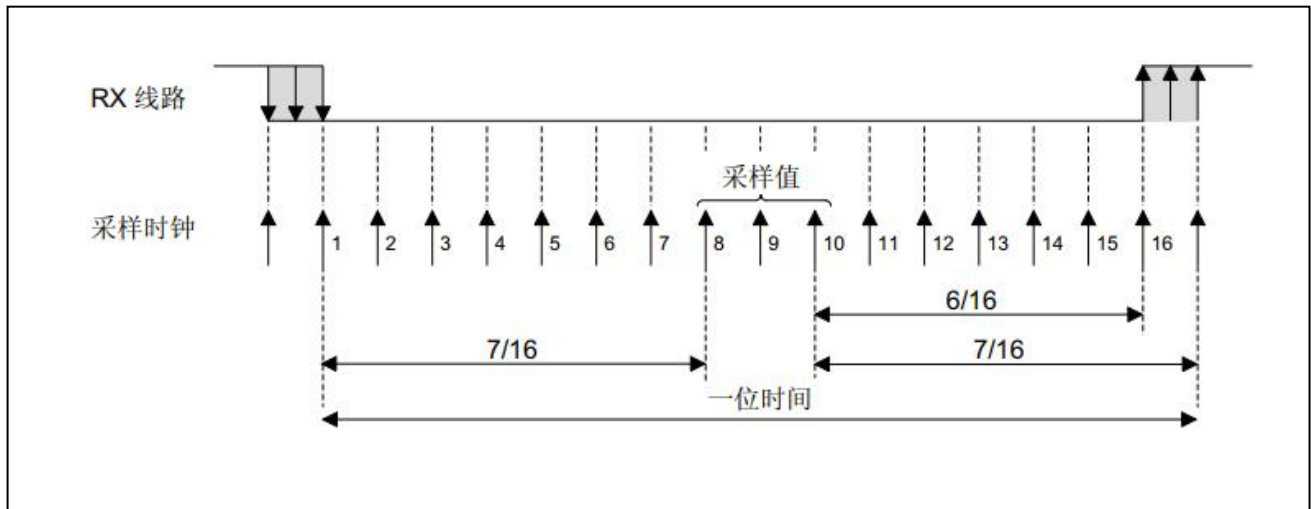


图 21-4 16 倍过采样时的数据采样

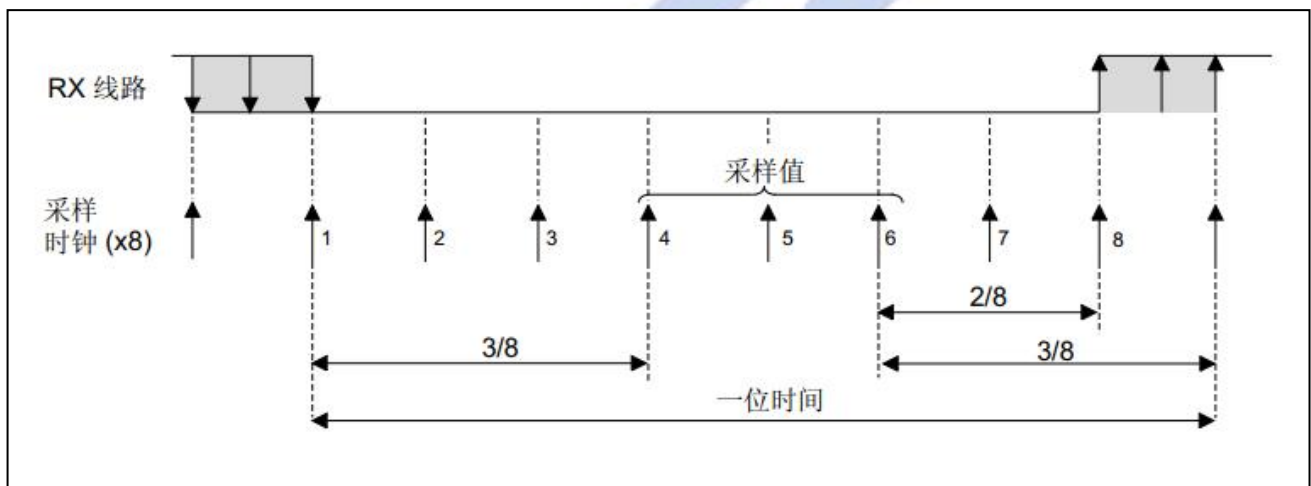


图 21-5 8 倍过采样时的数据采样

表 21-2 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误 (Framing error):

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

21.3.4. 波特率

接收器和发送器(Rx 和 Tx)的波特率设置为 BRR 寄存器中编程的相同值。适用于标准 USART 的波特率(OVER8=0 或 1)

16 倍过采样时的公式为:

$$\text{Baud_rate} = \frac{f_{\text{ck}}}{16 * \text{USARTDIV}[15:4] + \text{USARTDIV}[3:0]}$$

8 倍过采样时的公式为:

$$\text{Baud_rate} = \frac{f_{\text{ck}}}{8 * \text{USARTDIV}[15:4] + \text{USARTDIV}[2:0]}$$

注意: LIN 和 IrDA 模式(OVER8=0)下只能用 16 倍过采样

该波特率发生还带补偿功能,当所选波特率不能被 16 或 8 整除时,每一个 bit 发送时间具有一个补偿值。

USART 能够根据接收一个字符检测并自动设置 BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用:

- 事先不知道系统的通信速度
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。时钟源频率必须与预期通信速度匹配(16 倍过采样时,波特率介于 $f_{\text{CK}}/65535$ 与 $f_{\text{CK}}/16$ 之间;8 倍过采样时,波特率介于 $f_{\text{CK}}/65535$ 与 $f_{\text{CK}}/8$ 之间)。

激活自动波特率检测前,必须先选择自动波特率检测模式。根据不同的字符模式,存在各种检测模式。这些模式可通过 CTL[ABRMOD]字段选择。在这些自动波特率模式下,波特率在同步接收数据期间被多次测量,每次测量的结果都与前一次进行比较。

- 模式 0: 以 1 位开头的任意字符。这种情况下,USART 会测量起始位的持续时间(下降沿到上升沿)。
- 模式 1: 以 10xx 位模式开头的任意字符。这种情况下,USART 会测量起始位和第一个数据位的持续时间。测量在下降沿到下降沿期间完成,可在信号斜率较小时确保较高的精度。
- 模式 2: 0x7F 字符帧(可以是 LSB 在前模式下的 0x7F 字符,也可以是 MSB 在前模式下的 0xFE 字符)。这种情况下,先在起始位结束时更新波特率(BR),然后在位 6 结束时更新波特率(根据从下降沿到下降沿执行的测量:BR6)。以 BR 对位 0 到 6 进行采样,而以 BR6 对字符的其它位进行采样。
- 模式 3: 0x55 字符帧。这种情况下,先在起始位结束时更新波特率(BR),然后在位 0 结束时更新波特率(根据从下降沿到下降沿执行的测量:BR0),最后在位 6 结束时更新波特率(BR6)。以 BR 对位 0 进行采样,以 BR0 对位 1 到 6 进行采样,以 BR6 对字符的其它位进行采样。同时,对 RX 线路的每个中间转换执行其它检查。如果 RX 上的转换与接收器(基于根据位 0 计算的波特率的接收器)未充分同步,则生成错误。

激活自动波特率检测前,必须先通过向 BRR 寄存器写入非零的波特率值来初始化该寄存器。

21.3.5. 多处理器通信

在多处理器通信中,以下位要保持清零:CTL 寄存器中的 HDSEL 位,CFG 寄存器中的 LINEN 和 IREN 位。

可以通过 USART 进行多处理器通信(多个 USART 连接在一个网络中)。例如,其中一个 USART 可以是主 USART,其 TX 输出与其它 USART 的 RX 输入相连接。其它 USART 为从 USART,其各自的 TX 输出在逻辑上通过与运算连在一起,并与主 USART 的 RX 输入相连接。在多处理器配置中,理想情况下通常只有预期的消息接收方主动接收完整的消息内容,从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能,必须将 CTL[MME]位置 1。在静默模式下:

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- IS[RWU]位置 1。在某些情况下, RWU 可以由硬件或软件通过

RQ[MMRQ]位自动控制。根据 CTL[WAKE]位的设置, USART 可使用以下两种方法进入或退出静默模式:

- 如果 WAKE 位被复位,则进行空闲线路检测,
- 如果 WAKE 位置 1,则进行地址标记检测。

21.3.5.1.空闲线路检测(WAKE=0)

当向 MMRQ 位写入 1 且 RWU 位自动置 1 时, USART 进入静默模式。当检测到空闲帧时,它会被唤醒。此时 RWU 位会由硬件清零,但 IS[IDLE]位不会置 1。

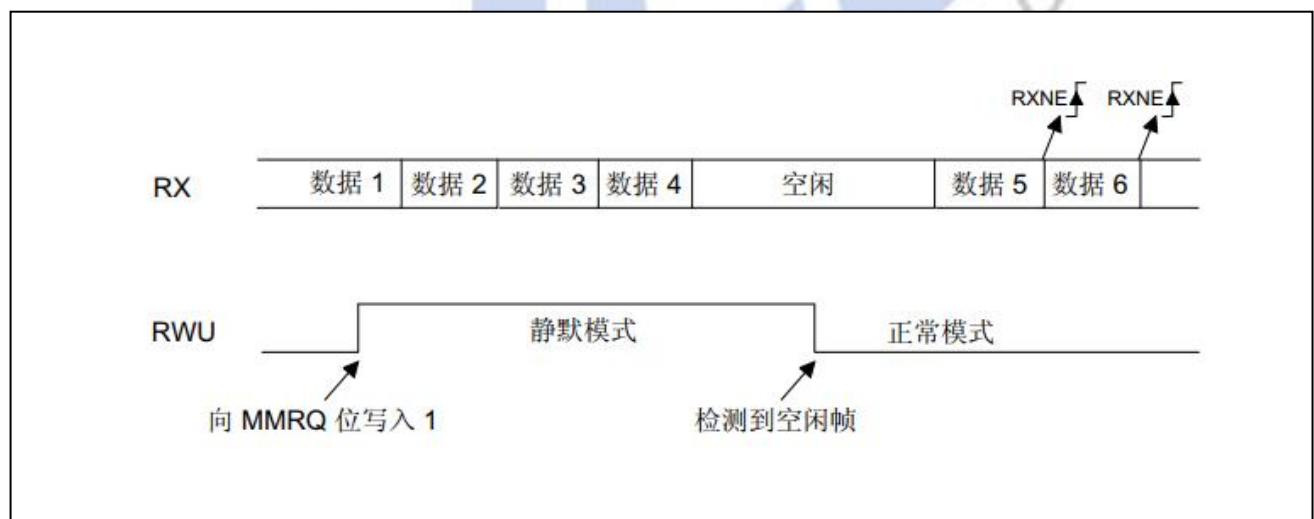


图 21-6 使用空闲线路检测时的静默模式

如果在 IDLE 字符已经过去时将 MMRQ 位置 1,将不会进入静默模式(RWU 未置 1)。如果在线路处于空闲状态时激活 USART,在一个 IDLE 帧持续时间后(不只在接收一个字符帧后)会检测到空闲状态。

21.3.5.2.4 位/7 位地址标记检测 (WAKE=1)

在此模式下,如果字节的 MSB 为 1,则将这些字节识别为地址,否则将其识别为数据。在地址字节中,目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其

地址进行比较，该接收器的地址在 CTL[ADD]位中进行设置。

注意：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0]和 ADD[7:0]）。

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断。当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE 位会针对地址字符置 1。下图给出了使用地址标记检测时静默模式行为的示例。

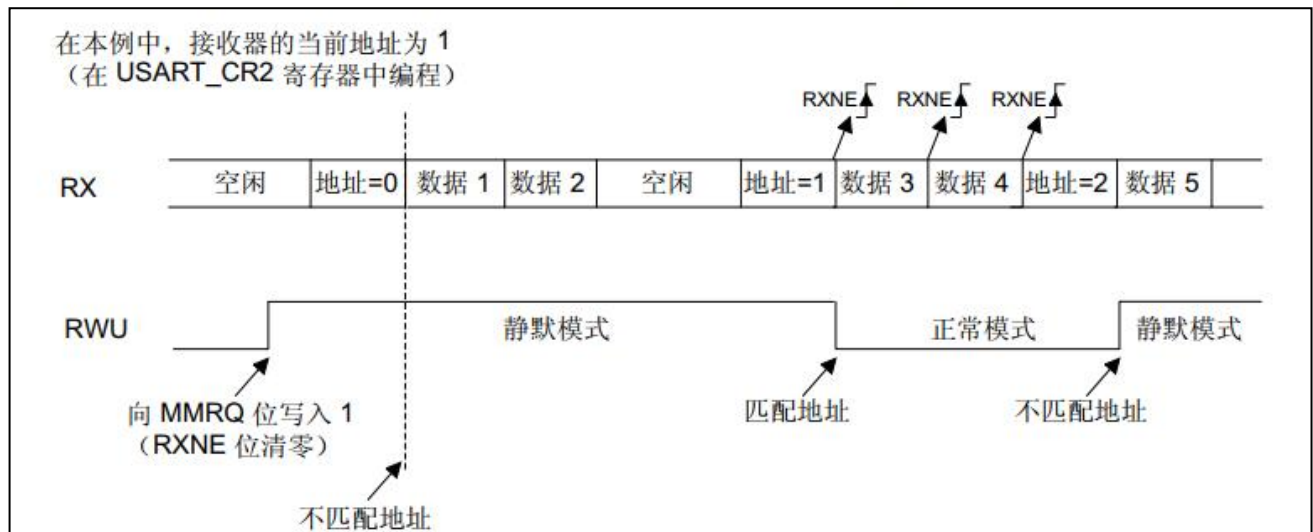


图 21-7 使用地址标记检测时的静默模式

21.3.6. MODBUS 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。USART 为块结束检测提供基本支持，无需软件开销或其它资源。

21.3.6.1.Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。超时功能和中断必须分别通过 USART_CTL 寄存器中的 RTOEN 位和 USART_IE [RTOIE]位激活。与 2 个字符时间（例如 22x 位持续时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

21.3.6.2.Modbus/ASCII

在此模式下，块结束通过特定(CR/LF)字符序列识别。USART 通过字符匹配功能管理此机制。通过在 ADD[7:0] 字段中编程 LFASCII 码以及激活字符匹配中断(CMIE=1)，软件可在接收到 LF 时获得通知。

21.3.7. 同步模式

在输出时，只需要将需要发送的数据写入 USART_TD 寄存器中，数据发送立刻开始，数据通过 USART_TX 端输出，时钟脉冲通过 USART_CK 输出，发送时 LSB 数据先输出。使能发送后只要写一个数据就会自动发送，当只使能接收时，必须先写一个数据才会接收一个数据，形成类 loopback 模式。同步发送的时钟极性和相位可由 CPOL 和 CPHA 来设置。

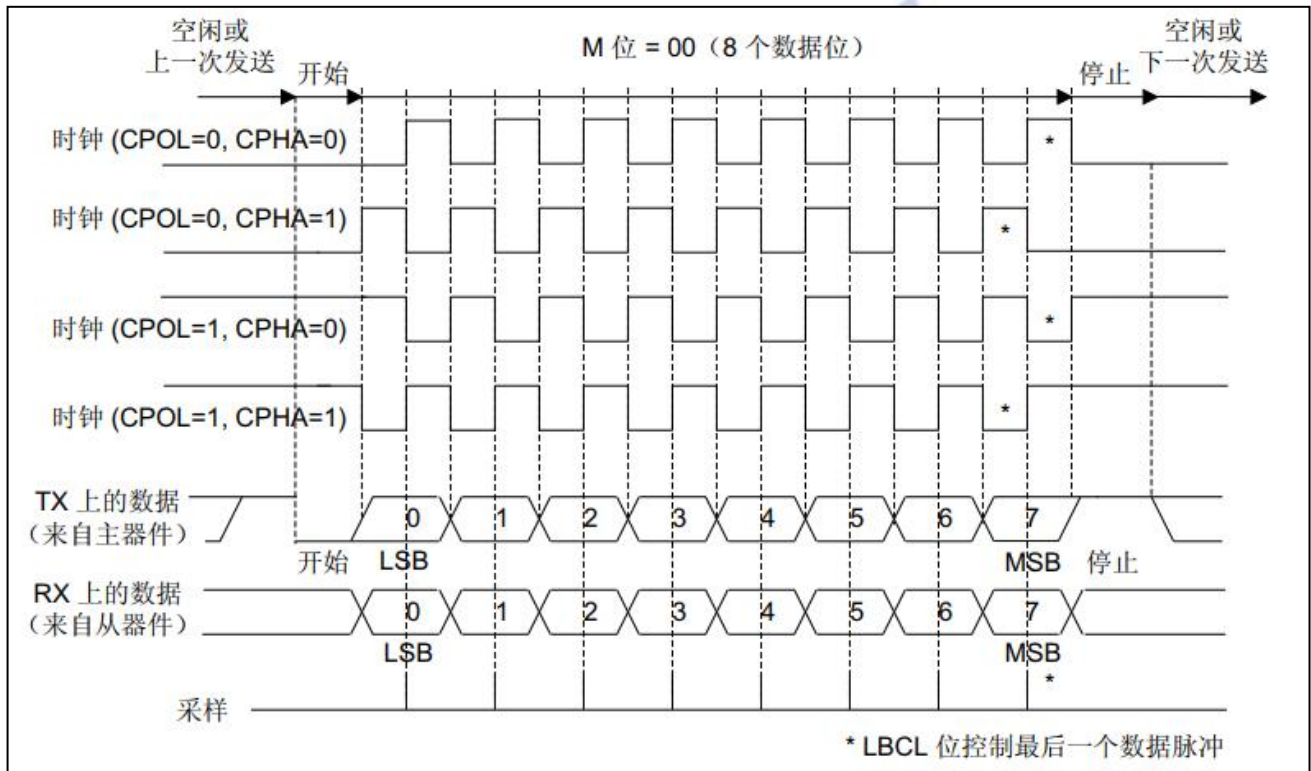


图 21-8 USART 数据始终时序图

当使用智能卡模式，同步模块可以产生智能卡所需的时钟，其中时钟分频由 CFG[PSC]（预分频值）确定。

21.3.8. 单线半双工模式

通过将 CTL[HDSEL] 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：CFG[LINEN]、CTL[CLKEN] 和 CFG[IREN] 位。USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 CTL[HDSEL] 可在半双工通信和全双工通信间进行选择。一旦向 HDSEL 位写入 1：

- TX 和 RX 线路从内部相连接
- 不能再使用 RX 引脚
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须将 TX 外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会

持续进行。

21.3.9. LIN 模式

通过将 CFG[LINEN]位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：CTL[STOP]、CTL[CLKEN]、CTL[HDSEL]和 CGF[IREN]位。LIN 发送与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用以下配置，同时还具有以下区别：

- ML 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBKRQ 位置 1 会发送 13 个“0”位作为断路字符。然后会发送值为“1”的 2 个位以进行下一启动检测。

21.3.9.1.LIN 接收

使能 LIN 模式后，将激活断路检测电路。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生断路即可检测出来。接收器（CTL[RE]=1）使能后，电路便开始监测启动信号的 RX 输入。检测起始位的方法与搜索断路字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 13 个连续位均检测为“0”，且其后跟随分隔符，则 IS[LBDIE]标志将置 1。如果 IE[LBDIE]位=1，则会生成中断。

在验证断路前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。如果在第 13 次采样前已对“1”采样，则断路检测电路会取消当前检测，并重新搜索起始位。如果禁止 LIN 模式(CFG[LINEN]=0)，接收器会作为正常的 USART 继续工作，不会再进行断路检测。

如果使能 LIN 模式(CGF[LINEN]=1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何断路帧中），接收器即会停止，直到断路检测电路接收到“1”（断路字不完整时）或接收到分隔符（检测到断路时）为止。

21.3.10. IrDA SIR 模式

通过将 CFG[IREN]位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：CFG[LINEN]、CTL[STOP]和 CTL[CLKEN]位和 CTL[HDSEL]位。

IrDASIR 物理层规定使用反相归零(RZI)调制方案，它以红外光脉冲表示逻辑 0。SIR 发送编码器用于调制 USART 发出的非归零(NRZ)位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。USART 支持的 SIRENDEC 比特率最高为 115.2Kbps。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 IrDA 码器的解码数据时），则 IrDA

不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。

- 0 作为高电平脉冲发送，而 1 作为 0 发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IrDA 规范要求脉冲容忍值要大于 $1.41\mu\text{s}$ 。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 CFG 寄存器中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，CTL[STOP]位必须配置为“1 个停止位”。

21.3.10.1.IrDA 低功耗模式

• 发送器

在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42MHz。通常此值是 1.8432MHz($1.42\text{MHz} < \text{PSC} < 2.12\text{MHz}$)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。

• 接收器

在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1 个 PSC 周期的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（CFG[PSC]值）时，才是有效低电平。

注意：宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10ms 的延迟（IrDA 是一个半双工协议）。

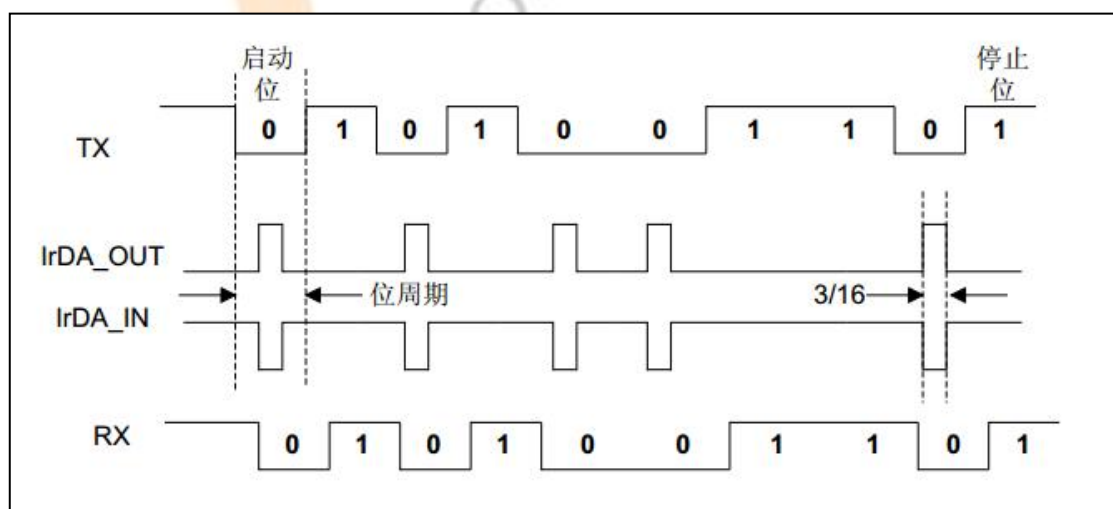


图 21-9 IrDA 数据调制 (3/16)——正常模式

21.4. USART 中断

表 21-3 中断事件表

中断事件	事件标志	清除标志	使能控制位
TX clear 完成	TXF	TXFCF	TXFIE
RX clear 完成	RXF	RXFCF	RXFIE
检测到空闲线路	IDLE	IDLECF	IDLEIE
从静默模式唤醒	WUF	WUCF	WUIE
字符匹配	CMF	CMCF	CMIE
自动波特率检测错误	ABRE	ABRECF	ABREIE
块结束	EOBF	EOBCF	EOBIE
接收器超时	RTOF	RTOCF	RTOIE
LIN 断路	LBDF	LBDCF	LBDIE
发送数据寄存器为空	TXEIF	TXEIFCF	TXEIE
发送完成	TC	TCCF	TCIE
接收数据寄存器不为空（已准备好读取数据）	RXNEIF	RXNEIFCF	RXNEIE
接收 FIFO 满	RXFULL	RXFULLCF	RXFULLIE
检测到溢出错误	ORE	ORECF	OREIE
多缓冲区通信中的噪声标志	NF	NCF	NIE
帧错误	FE	FECF	FEIE
奇偶校验错误	PE	PECF	PEIE

可通过 IS 寄存器对中断事件标志进行查询，通过使能 IE 寄存器的相应位允许产生中断。中断发生后可对 IC 寄存器的相应位写 1 清除相应中断标志。

USART 中断事件被连接到相同的中断向量。

- 发送期间：发送完成、清除以发送、发送数据寄存器为空或帧错误中断。
- 接收期间：空闲线路检测、上溢错误、接收数据寄存器不为空、奇偶校验错误、LIN 断路检测、噪声标志、帧错误、字符匹配等。如果相应的使能控制位置 1，则这些事件会生成中断。

21.5. USART 寄存器

21.5.1. USART 寄存器映射

表 21-4 USART 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	USART 版本寄存器(IP_VERSION)	rw	0xA368_0008
0x04	USART 控制寄存器 (CTL)	rw	0x0000_0000
0x08	USART 配置寄存器 (CFG)	rw	0x0000_0000
0x0C	USART 波特率寄存器 (BR)	rw	0x0000_0000
0x10	USART 超时寄存器 (RTO)	rw	0x0000_0000
0x14	USART 请求寄存器 (RQ)	w	0x0000_0000
0x18	USART 中断和状态使能寄存器 (IE)	rw	0x0000_0000
0x1C	USART 中断和状态寄存器 (IS)	r	0x0000_1200
0x20	USART 中断和状态清零寄存器 (IC)	w	0x0000_0000
0x24	USART 接收数据寄存器 (RD)	r	0x0000_0000
0x28	USART 发送数据寄存器 (TD)	w	0x0000_0000

21.5.2. USART 控制寄存器 (CTL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX_MBIT T	TX_MBIT T	ADDM7	LPBK	RTOEN	-	-	MBUS_E N	MME	WAKE	HDSEL	LBCL	CPOL	CPHA	CLKEN	START_F LT
rw	rw	rw	rw	rw	-	-	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ONEBIT	ABREN	ABRM0D	ML	STOP	PCE	PS	MSBFIRS T	DATAIN V	OVER8	TE	RE	UE			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31 RX_MBIT	当接收字长为 9bit 时，接收到的第 9 个 bit
30 TX_MBIT	当发送字长为 9bit 时，需要发送的第 9 个 bit
29 ADDM7	4 位/7 位地址选择 0: 7 位地址 1: 4 位地址
28 LPBK	Loopback 使能位 0: 禁止 1: 使能
27	接收器超时使能，此位由软件置 1 和清零。

位域	功能
RTOEN	0: 禁止接收器超时功能。 1: 使能接收器超时功能。 使能此功能后, 如果 RX 线路在 RTOR (接收器超时寄存器) 中编程的持续时间内处于空闲状态 (无接收), 则 IS[RTOF]标志置 1。
26-25	保留
24 MBUS_EN	modbus 功能使能位 0: 禁止 1: 使能
23 MME	静默模式使能(Mute mode enable) 此位用于激活 USART 的静默模式功能, 此位置 1 时, USART 可按 WAKE 位定义的方式在活动模式与静默模式之间切换。此位由软件置 1 和清零。 0: 接收器永久处于活动模式 1: 接收器可在静默模式和活动模式之间切换。
22 WAKE	接收器唤醒方法(Receiver wakeup method) 此位用于确定 USART 静默模式的唤醒方法。此位由软件置 1 或清零。 0: 空闲线路 1: 地址标记
21 HDSEL	半双工选择(Half-duplex selection) 选择单线半双工模式 0: 未选择半双工模式 1: 选择半双工模式
20 LBCL	最后一个位时钟脉冲>Last bit clock pulse) 此位用于在同步模式下选择与发送的最后一个数据位(MSB)关联的时钟脉冲是否必须在 CK 引脚上输出。 0: 最后一个数据位的时钟脉冲不在 CK 引脚上输出 1: 最后一个数据位的时钟脉冲在 CK 引脚上输出
19 CPOL	时钟极性(Clock polarity) 该位允许用户在同步模式下选择 CK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系 0: 空闲时 CK 引脚为低电平 1: 空闲时 CK 引脚为高电平
18 CPHA	时钟相位(Clock phase) 此位用于在同步模式下选择 CK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟 0: 从第一个时钟边沿开始采样数据 1: 从第二个时钟边沿开始采样数据
17 CLKEN	时钟使能, 用于同步模式和智能卡的时钟产生 该位允许用户使能 CK 引脚。 0: 禁止 CK 引脚 1: 使能 CK 引脚
16 START_FLT	检测起始位滤波使能位
15 ONEBIT	一个采样位方法使能(One sample bit method enable) 该位允许用户选择采样方法。选择一个采样位方法后, 将禁止噪声检测标志(NF)。 0: 三个采样位方法 1: 一个采样位方法 注意: <i>ONEBIT 功能仅适用于数据位, 不适用于起始位。</i>
14	自动波特率使能, 此位由软件置 1 和清零。

位域	功能
ABREN	0: 禁止自动波特率检测 1: 使能自动波特率检测
13-12 ABRM0D	自动波特率模式，这些位由软件置 1 和清零。 00: 通过测量起始位检测波特率。 01: 下降沿到下降沿的测量。（接收到的帧必须以一个等于 1 的位开头，即帧=10xxxxxx） 10: 0x7F 帧检测。 11: 0x55 帧检测。 <i>注意:</i> <i>如果 DATAINV=1 且/或 MSBFIRST=1，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA。</i>
11-10 ML	发送字长，软件置位选择。 00: 8 个数据位 01: 9 个数据位 10: 7 个数据位 11: 保留
9-8 STOP	停止位(STOPbits)，停止位选择。 00: 1 个停止位 01: 保留 10: 2 个停止位 11: 保留
7 PCE	奇偶校验使能位，此位由软件置 1 和清零 0: 禁止奇偶校验控制 1: 使能奇偶校验控制
6 PS	奇偶校验选择位，此位由软件置 1 和清零 该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。 0: 偶校验 1: 奇校验
5 MSBFIRST	最高有效位在前，此位由软件置 1 和清零 0: 发送/接收数据时位 0 在前，后跟起始位。 1: 发送/接收数据时 MSB（位 7/8/9）在前，后跟起始位。
4 DATAINV	二进制数据反向，此位由软件置 1 和清零 0: 按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L） 1: 按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。
3 OVER8	过采样模式 0: 16 倍过采样 1: 8 倍过采样 <i>注意:</i> <i>在 LIN 和 IrDA 模式下，此位必须保持清零</i>
2 TE	USART 发送器使能位，此位由软件置 1 和清零 0: 禁止发送器 1: 使能发送器 <i>注意: 传输期间 TE 位上的“0”脉冲会在当前字的后面发送一个空闲线路的报头。</i>
1 RE	USART 接收器使能位，此位由软件置 1 和清零 0: 禁止接收器 1: 使能接收器并开始搜索起始位
0 UE	USART 使能位 此位清零后，USART 预分频和输出将立即停止，并 abort 当前操作，但是 USART 的配置信息会保留，IS 寄存器则会恢复成 default 值。此位由软件置 1 和清零。

21.5.3. USART 配置寄存器（CFG）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	PSC							
-	-	-	-	-	-	-	-	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD3				ADD7				LINEN	-	-	-	-	-	IREN	IRLP
rw				rw				rw	-	-	-	-	-	rw	rw

位域	功能
31-24	保留
23-16 PSC	预分频器值(Prescaler value) 在 IrDA 低功耗和正常的 IrDA 模式下： PSC[7:0]=IrDA 正常和低功耗波特率，用于编程预分频器，进行 USART 源时钟分频以获得低功耗频率，使用寄存器中给出的值（8 个有效位）对源时钟进行分频： 00000000：保留-不编程此值 00000001：源时钟 1 分频 00000010：源时钟 2 分频
15-12 ADD3	USART 节点的地址 此位域用于指定 USART 节点的地址或要识别的字符代码。 此位域在多处处理器通信时于静默模式或停止模式下使用，以通过 7 位地址标记检测进行唤醒。发送器发送字符的 MSB 应为 1。此位域还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0]值进行比较，如果匹配，CMF 标志将置 1。
11-8 ADD7	USART 节点的地址 此位域用于指定 USART 节点的地址或要识别的字符代码。此位域在多处处理器通信时于静默模式或停止模式下使用，以通过地址标记检测进行唤醒。
7 LINEN	LIN 模式使能，此位由软件置 1 和清零。 0：禁止 LIN 模式 1：使能 LIN 模式 LIN 模式可以使用 RQ[SBKRQ]位发送 LIN 同步断路（13 个低位），并可检测 LIN 同步断路。
6-2	保留
1 IREN	IrDA 模式使能，此位由软件置 1 和清零。 0：禁止 IrDA 1：使能 IrDA
0 IRLP	IrDA 低功耗模式，该位用于选择正常模式和低功耗 IrDA 模式 0：正常模式 1：低功耗模式

21.5.4. USART 波特率寄存器（BR）

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

位域	功能
31-16	保留
15-4 BRR[15:4]	波特率寄存器 BRR[15:4]=USARTDIV[15:4]
3-0 BRR[3:0]	波特率补偿位 当 OVER8=0 时： BRR[3:0]=USARTDIV[3:0]。 当 OVER8=1 时： BRR[2:0]=USARTDIV[3:0]，右移 1 位。 BRR[3]必须保持清零。

21.5.5. USART 超时寄存器（RTO）

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

位域	功能
31-24	保留
23-0 RTO	接收器超时值 此位域用于提供接收器的超时值（以位持续时间数为单位）。 在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。 注意： 每个接收到的字符只能对此值编程一次 可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。 如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。

21.5.6. USART 请求寄存器 (RQ)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
-	-	-	-	-	-	-	-	-	-	-	w	w	w	w	w

位域	功能
31-5	保留
4 TXFRQ	发送数据刷新请求 向该位写入 1 时会将 TXE 标志置 1。这可丢弃发送数据。
3 RXFRQ	接收数据刷新请求 向该位写入 1 时会将 RXNE 标志清零。这可以丢弃数据而不对其执行读取操作，并避免发生上溢。
2 MMRQ	静默模式请求 向此位写入 1 可将 USART 置于静默模式，并将 RWU 标志置 1。
1 SBKRQ	发送断路请求 向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。 <i>注意：这种情况下，应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。</i>
0 ABRRQ	自动波特率请求 向此位写入 1 可复位 IS[ABRF]标志，并请求对下一个接收到的数据帧进行自动波特率测量。

21.5.7. USART 中断和状态使能寄存器 (IE)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	TXFIE	RXFIE	IDLEIE	WUIE	-	-	CMIE	-	-
-	-	-	-	-	-	-	rw	rw	rw	rw	-	-	rw	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABREIE	EOBIE	RTOIE	-	-	LBDIE	-	TXEIE	TCIE	-	RXNEIE	RXFULL	OREIE	NIE	FEIE	PEIE
rw	rw	rw	-	-	rw	-	rw	rw	-	rw	rw	rw	rw	rw	rw

位域	功能
31-25	保留
24 TXFIE	TX clear 完成中断使能
23 RXFIE	RX clear 完成中断使能
22	IDLE 帧中断使能(IDLE interrupt enable)

位域	功能
IDLEIE	此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[IDLE]=1 时, 生成 USART 中断
21 WUIE	从静默模式唤醒中断使能(Wakeup from Stop mode interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[WUF]=1 时, 生成 USART 中断
20-19	保留
18 CMIE	字符匹配中断使能(Character match interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 如果 IS[CMF]位置 1, 则生成 USART 中断。
17-16	保留
15 ABREIE	自动波特率检测错误中断使能位 此位由软件置 1 和清零。 0: 禁止中断 1: 如果 IS[ABRE]位置 1, 则生成 USART 中断。
14 EOBIE	块结束中断使能(End of Block interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: IS[EOBF]标志置 1 时生成 USART 中断
13 RTOIE	接收器超时中断使能(Receiver timeout interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: IS[RTOF]位置 1 时生成 USART 中断。
12-11	保留
10 LBDIE	LIN 断路检测中断使能(LIN break detection interrupt enable) 断路中断屏蔽 (使用断路分隔符进行断路检测) 0: 禁止中断 1: 当 IS[LBDF]=1 时, 生成中断
9	保留
8 TXEIE	TXE 中断使能(interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[TXE]=1 时, 生成 USART 中断
7 TCIE	传输完成中断使能(Transfer complete interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[TC]=1 时, 生成 USART 中断
6	保留
5 RXNEIE	RXNEIF 标志中断使能(RXNE interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[RXNE]=1 时, 生成 USART 中断
4 RXFULLIE	RXFULL 中断使能 X, 此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[RXFULL]=1 时, 生成 USART 中断

位域	功能
3 OREIE	接受 FIFO 上溢中断使能位 0: 禁止中断 1: 当 IS[ORE]=1 时, 生成 USART 中断
2 NIE	噪声中断使能 0: 禁止中断 1: IS[NF]=1 时生成中断。
1 FEIE	错误中断使能(Error interrupt enable) 停止位错误中断使能 0: 禁止中断 1: IS[FE]=1 时生成中断。
0 PEIE	PE 奇偶校验错误中断使能(PE interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 IS[PE]=1 时, 生成 USART 中断

21.5.8. USART 中断和状态寄存器 (IS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	TXF	RXF	IDLE	WUF	RWU	SBKF	CMF	BUSY	ABRF
-	-	-	-	-	-	-	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRE	EOBF	RTOF	-	-	LBDF	TXE	TXEIF	TC	RXNE	RXNEIF	RXFULL	ORE	NF	FE	PE
r	r	r	-	-	r	r	r	r	r	r	r	r	r	r	r

位域	功能
31-25	保留
24 TXF	TX FIFO clear 完成标志位, 当对应中断使能位 TXFIE 开启, 会产生中断
23 RXF	RX FIFO clear 完成标志位, 当对应中断使能位 RXFIE 开启, 会产生中断
22 IDLE	检测到空闲线路(IDLE line detected) 检测到空闲线路时, 此位由硬件置 1。如果 IE[IDLEIE]=1, 则会生成中断。 通过向 IC[IDLECF]写入 1, 此位由软件清零。 0: 未检测到空闲线路 1: 检测到空闲线路
21 WUF	接收器检测到静默模式唤醒事件标志位 当对应的中断使能位(WUIE)置 1 时, 会产生中断 0: 未检测到唤醒事件 1: 检测到唤醒事件
20 RWU	接收器静默模式状态位 该位指示 USART 是否处于静默模式。 当识别出唤醒/静默序列时, 此位由硬件清零/置 1。静默模式控制序列(地址或 IDLE)通过 CTL[WAKE]位

位域	功能
	选择。当选择 IDLE 模式下唤醒时，该位只能通过用软件向 RQ[MMRQ]位写 1 的方式置 1。 0: 接收器处于活动模式 1: 接收器处于静默模式
19 SBKF	发送断路标志(Send break flag) 此位指示已请求发送断路字符。通过将 1 写入 RQ[SBKRQ]位，此位由软件置 1。此位在断路发送的停止位期间由硬件自动复位。 0: 不发送断路字符 1: 将发送断路字符
18 CMF	字符匹配标志(Character match flag) 接收到由 ADD[7:0]定义的字符后由硬件将此位置 1。如果 IE[CMIE]=1，则会生成中断。 通过向 IC[CMCF]写入 1，此位由软件清零。 0: 未检测到字符匹配 1: 检测到字符匹配
17 BUSY	忙标志(Busy flag) 此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。 在接收结束（成功或失败）时复位。 0: USART 处于空闲状态（无接收） 1: 正在接收
16 ABRF	自动波特率标志(Auto baud rate flag) 已设置自动波特率（RXNE 置 1，并在 RXNEIE=1 时生成中断），或者自动波特率操作未成功完成时，此位由硬件置 1(ABRE=1)（此时，ABRE、RXNE 和 FE 也置 1）。 为请求新的自动波特率检测，通过向 RQ[ABRRQ]写入 1，等待波特率检测完成，此位由硬件自动清零。
15 ABRE	自动波特率错误(Auto baud rate error) 如果波特率测量失败（波特率超出范围或字符比较失败），此位由硬件置 1。 通过将 1 写入 IC[ABRECF]位，此位由软件清零。
14 EOBF	块结束标志(End of block flag) 接收到完整块后，此位由硬件置 1（例如 T=1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN+4 时执行检测。如果 IE[EOBIE]=1，则会生成中断。 通过向 IC[EOBCF]写入 1，此位由软件清零 0: 未达到块结束 1: 已达到块结束（字符数）
13 RTOF	接收器超时(Receiver timeout) 已经过在 RTO 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。如果 IE[RTOIE]=1，则会生成中断。在智能卡模式下，该超时对应于 CWT 或 BWT 时间。 通过向 IC[RTOCF]写入 1，此位由软件清零。 0: 未达到超值 1: 已达到超时值，未接收到任何数据 <i>注意：</i> <i>如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。</i> <i>如果此时间大于该值+2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。</i> <i>即使 RE=0，计数器仍会计数，但 RTOF 仅在 RE=1 时置 1。</i> <i>如果 RE 置 1 时已经超时，则 RTOF 将置 1。</i>
12-11	保留
10 LBDF	LIN 断路检测标志(LIN break detection flag) 检测到 LIN 断路时，此位由硬件置 1。如果 IE[LBDIE]=1，则会生成中断。 通过向 IC[LBD CF]写入 1，此位由软件清零。 0: 未检测到 LIN 断路 1: 检测到 LIN 断路

位域	功能
9 TXE	FIFO 中发送数据寄存器为空(Transmit data register empty) 0: 发送数据 FIFO 中有数据 1: 发送数据 FIFO 为空
8 TXEIF	发送数据寄存器从有数据到无数据变化或者发送 FIFO 写满标志。 当对应的中断使能时, 会触发中断
7 TC	发送完成(Transmission complete) 如果已完成对包含数据的帧的发送并且 TXE 置 1, 则此位由硬件置 1。如果 IE[TCIE]=1, 则会生成中断。 通过向 ICR[TCCF]写入 1 或向 TDR 寄存器执行写操作, 此位由软件清零。 0: 传送未完成 1: 传送已完成 注意: TE 位复位且无任何发送正在进行, TC 位会立即置 1。
6 RXNE	读取数据寄存器不为空(Read data register not empty) 当 RDR 移位寄存器的内容已传输到 RDR 寄存器时, 此位由硬件置 1。如果 IE[RXNEIE]=1, 则会生成中断。 通过对 RDR 寄存器执行读入操作将该位清零。也可以通过将 RQ[RXFRQ]位置 1 将 RXNE 标志位清零。 0: 未接收到数据 1: 已准备好读取接收到的数据
5 RXNEIF	接受数据 FIFO 的状态发生变化, 从无数据到有数据进入 注意: 若 FIFO 不为空, 则 RXNEIF 不会置位。
4 RXFULL	接收数据 FIFO 为满的标志位 当接收数据 FIFO 满时, 此位由硬件置 1。如果 IE[RXFULLIE]=1, 则会生成中断。 通过向 IC[RXFULLCF]写入 1, 此位由软件清零。
3 ORE	溢出错误(Overrun error) 在 RXNE=1 的情况下, 当移位寄存器中当前正在接收的数据准备好传输到 RD 寄存器时, 此位由硬件置 1。 如果 IE[OREIE]=1, 则会生成中断。 通过向 IC[ORECF]写入 1, 此位由软件清零。 0: 无溢出错误 1: 检测到溢出错误 注意: 当此位置 1 时, RD 寄存器的内容不会丢失, 但移位寄存器会被覆盖。
2 NF	START 位噪声检测标志(START bit Noise detection flag) 当在接收的帧上检测到噪声时, 此位由硬件置 1。 通过向 IC[NFCF]写入 1, 此位由软件清零。 0: 未检测到噪声 1: 检测到噪声 注意: 该位不会生成中断, 因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。NIE 位置 1 后, 如果在多缓冲区通信中 NF 标志置 1, 则会生成中断。 当线路无噪声时, 可以通过将 CTL[ONEBIT]位编程为 1 提高 USART 对偏差的容差来禁止 NF 标志。
1 FE	帧错误(Framingerror) 当检测到去同步化、过度的噪声或中断字符时, 此位由硬件置 1。如果 IE[FEIE]=1, 则会生成中断。 通过向 IC[FE CF]写入 1, 此位由软件清零。 0: 未检测到帧错误 1: 检测到帧错误或中断字符
0 PE	奇偶校验错误(Parity error) 当在接收器模式下发生奇偶校验错误时, 此位由硬件置 1。如果 IE[PEIE]=1, 则会生成中断。 通过向 IC[PECF]写入 1, 此位由软件清零。 0: 无奇偶校验错误 1: 奇偶校验错误

21.5.9. USART 中断和状态清零寄存器 (IC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	TXFCF	RXFCF	IDLECF	WUCF	-	-	CMCF	-	-
-	-	-	-	-	-	-	w	w	w	w	-	-	w	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRECF	EOBCF	RTOCF	-	-	LBDCF	-	TXECF	TCCF	-	RXNECF	RXFULL	ORECF	NFCF	FECF	PECF
w	w	w	-	-	w	-	w	w	-	w	w	w	w	w	w

位域	功能
31-25	保留
24 TXFCF	TX FIFO clear 完成标志清零
23 RXFCF	TX FIFO clear 完成标志清零
22 IDLECF	检测到空闲线路清零标志(Idle line detected clear flag) 将 1 写入此位时, IS[IDLE]标志将清零
21 WUCF	从静默模式唤醒标志清零位 将 1 写入此位时, IS[WUF]标志将清零
20-19	保留
18 CMCF	字符匹配清零标志(Character match clear flag) 将 1 写入此位时, IS[CMF]标志将清零。
17-16	保留
15 ABRECF	自动波特率错误标志清位 将 1 写入此位时, IS[ABRE]标志将清零。
14 EOBCF	块结束清零标志(End of block clear flag) 将 1 写入此位时, IS[EOBF]标志将清零。 <i>注意: 如果 USART 不支持智能卡模式, 该位保留并由硬件强制清零</i>
13 RTOCF	接收器超时清零标志(Receiver timeout clear flag) 将 1 写入此位时, IS[RTOF]标志将清零。 <i>注意: 如果 USART 不支持接收器超时功能, 该位保留并由硬件强制清零。</i>
12-11	保留
10 LBDCF	LIN 中断检测清零标志(LIN break detection clear flag) 将 1 写入此位时, IS[LBDP]标志将清零。
9	保留
8 TXEFCF	TXEIF 标志清零位 将 1 写入此位时, IS[TXEIF]标志将清零
7 TCCF	发送完成清零标志(Transmission complete clear flag) 将 1 写入此位时, IS[TC]标志将清零。
6	保留

位域	功能
5 RXNEIFCF	RXNEIF 标志清零位 将 1 写入此位时, IS[RXNEIF]标志将清零。
4 RXFULLCF	接收数据 FIFO 满标志清零位 将 1 写入此位时, IS[RXFULL]标志将清零
3 ORECF	上溢错误清零标志(Overrun error clear flag) 将 1 写入此位时, IS[ORE]标志将清零。
2 NFCF	检测到噪声清零标志(Noise detected clear flag) 将 1 写入此位时, IS[NF]标志将清零。
1 FECF	帧错误清零标志(Framing error clear flag) 将 1 写入此位时, IS[FE]标志将清零。
0 PECF	奇偶校验错误清零标志(Parity error clear flag) 将 1 写入此位时, IS[PE]标志将清零

21.5.10. USART 接收数据寄存器 (RD)

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

位域	功能
31-9	保留
8-0 RDR	接收数据值(Transmit data value)

21.5.11. USART 发送数据寄存器 (TD)

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

位域	功能
----	----

位域	功能
31-9	保留
8-0 TDR	发送数据值(Transmit data value) 包含要发送的数据字符。TD 寄存器在内部总线和输出移位寄存器之间提供了并行接口。在使能奇偶校验位的情况下（CR1[PCE]位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。



22. 低功耗通用异步接收器 (LPUART)

22.1. 简介

LPUART 是一种 UART，允许在有限功耗下双向 UART 通信。仅需 32.768 kHz LSE 时钟即可进行高达 9600 波特率的 UART 通信。当 LPUART 由与 LSE 时钟不同的时钟源驱动时，可以达到更高的波特率。即使当微控制器处于低功耗模式，能耗极低时，LPUART 也会等待 UART 帧的到来。LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。它支持半双工单线通信，还支持多处理器通信。

22.2. 主要特性

- 全双工异步通信
- NRZ标准格式
- 可编程波特率
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 通信数据长度可配（7/8/9）
- 通信数据顺序可配(MSB在前/LSB在前)
- 停止位可配（1/2）
- 发送和接收的单独信号极性可配
- 支持奇偶校验
- 支持单线半双工
- 传输检测
 - 接收缓冲区已满（RXNE）
 - 发送缓冲区已空（TXE）
 - 发送结束标志（TC）
- 错误检测
 - 帧错误（FE）
 - 溢出错误（ORE）
 - 奇偶校验错误（PE）
- 多处理器通信
 - 如果地址不匹配，LPUART将进入静默模式
- 支持静默模式唤醒
 - 空闲线检测
 - 地址标记检测
- 支持待机/停止模式唤醒
- 自动波特率检测

22.3. LPUART 功能说明

22.3.1. LPUART 框图

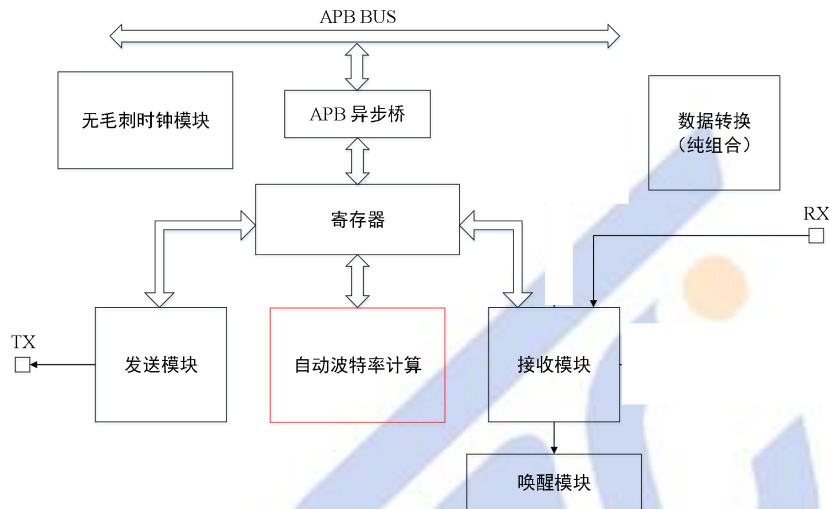


图 22-1 LPUART 功能框图

22.3.2. LPUART 信号

LPUART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

RX 为串行数据输入引脚。

- **TX**（发送数据输出引脚）

如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有待发送的数据，则 TX 引脚处于高电平。在单线模式下，该 I/O 用于发送和接收数据。注：TX/RX 引脚的 IO 端口配置时，需配置为上拉使能。

22.3.3. LPUART 字符说明

通过对 LPUART_CR1 中的 M[1:0] 位编程来选择 7 位、8 位或 9 位的字长。通过 LPUART_CR1 中的 PCE 和 PS 位控制奇偶校验使能。下表列出了可能的 LPUART 帧格式。

表 22-1 LPUART 位域

M	PCE	帧格式
00	0	SB 8 位数据 STB
00	1	SB 7 位数据 PB STB
01	0	SB 9 位数据 STB
01	1	SB 8 位数据 PB STB

10	0	SB 7 位数据 STB
10	1	SB 6 位数据 PB STB

注：SB：起始位，STB：停止位，PB：奇偶校验位。

默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。通过极性配置控制，可以单独针对每个信号对这些值取反。发送时可以发送特殊的空闲字符和中断字符。空闲字符为整个帧周期内电平均为“1”（停止位的电平也是“1”）；中断字符为在一个帧周期内接收到的电平均为“0”，发送器在中断帧的末尾插入2个停止位。



图 22-2 字长编程（以 9 位字长 M=01，1 个停止位 stop=0 为例）

注：以9位字长，1个停止位为例，bit8可能为奇偶校验位。

22.3.4. LPUART 发送器

发送器可发送7位、8位或9位的数据字，具体取决于M位的配置。要激活发送器功能，必须将发送使能位(TE)置1。发送移位寄存器中的数据输出到TX引脚。

22.3.4.1.字符发送

LPUART发送期间，首先通过TX引脚移出数据的最低有效位（默认配置）。每个字符前面都有一个起始位，其逻辑电平在一个位周期内为低电平。字符由可配置数量的停止位终止。LPUART 支持1个和2个停止位。

注：

1. 向 LPUART_TDR 中写入要发送的数据前，TE 和 UE 位必须先置 1。
2. 数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，从而损坏 TX 引脚上的数据。当前传输的数据将会丢失。
3. 使能 TE 位后，将会发送空闲帧。

22.3.4.2.可配置的停止位

通过对 LPUART_CR2中的 STOP 位进行编程来选择各个字符发送的停止位的数量（1个或者2个）。空闲帧发送包括停止位。中断帧发送是10个低电平位（M=00）、11个低电平位（M=01）或9个低电平（M=10），

然后是2个停止位。



图 22-3 可配置的停止位（以 9 位字长 M=01 为例）

22.3.4.3. 建议的发送步骤

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用 LPUART_BRR 选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 中的 UE 位写入 1 使能 LPUART。
5. 将 LPUART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
6. TXE=1 时在 LPUART_TDR 中写入要发送的数据（该操作将清零 TXE 位）。为每个要在单缓冲区模式下发送的数据重复这一步骤。
7. 向 LPUART_TDR 寄存器写入最后一个数据后，等待至 TC=1。这表明最后一个帧的传送已完成。禁止 LPUART 或进入低功耗模式时需要此步骤，以避免损坏最后一次发送。

22.3.4.4. 单字节通信

始终通过向发送数据寄存器写入数据将 TXE 位清零。

TXE 位由硬件置 1，它表示：

- 数据已从 LPUART_TDR 传送到移位寄存器中且数据发送已开始。
- LPUART_TDR 寄存器为空。
- LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

TXEIE 位置 1 时该标志位会生成中断。

如果帧已发送（停止位后）且 TXE 位置 1，TC 位将变为高电平。如果 LPUART_CR1 寄存器中的 TCIE 位置 1，将生成中断。向 LPUART_TDR 寄存器中写入最后一个数据后，必须等待至 TC=1，之后才可禁止 LPUART 或使微控制器进入低功率模式。

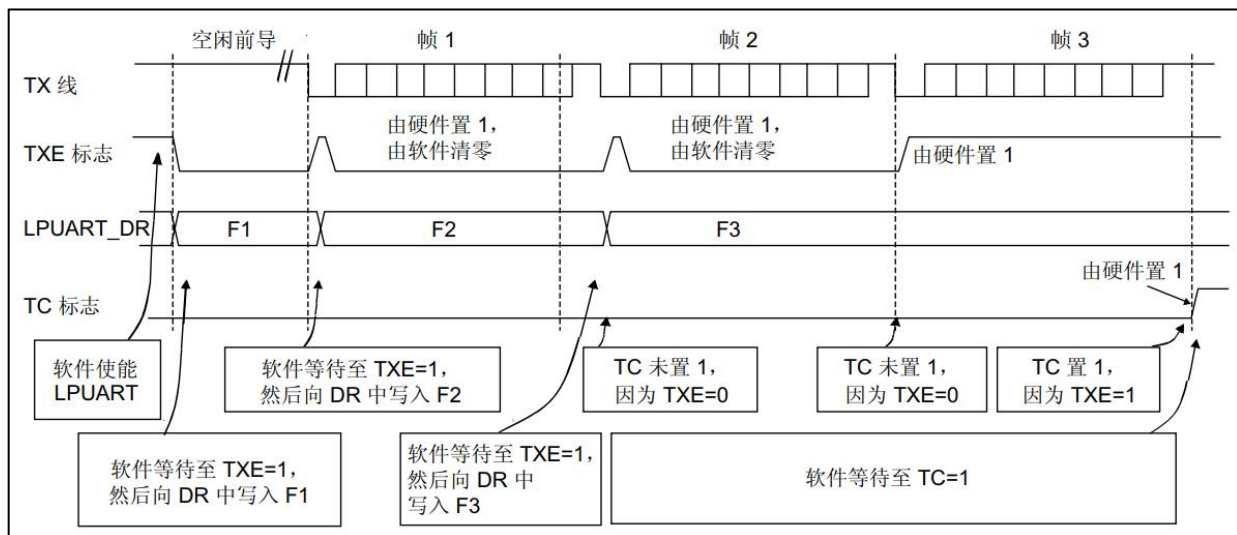


图 22-4 发送时的 TC/TXE

22.3.4.5. 中断和空闲字符

将SBKRQ位置1将发送一个中断字符。中断帧的长度取决于M位。如果将“1”写入 SBKRQ位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。通过写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间），此位由硬件复位。LPUART在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。这种情况下，应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到TXE标志=1后将SBKRQ位置1。

将TE位置1会驱动LPUART在第一个数据帧之前发送一个空闲帧。

22.3.5. LPUART 接收器

接收器可接收7位、8位或9位的数据字，具体取决于M位的配置。要激活接收器功能，必须将接收使能位(RE)置1。（接收器不进行噪声检查）

22.3.5.1. 起始位检测

在LPUART中，进行起始位检测时，首先应在RX引脚上检测到下降沿，然后接收器将正常采样所有传入位。

字符接收

LPUART接收期间，通过RX引脚逐位移入从发送端接收到的数据，默认配置下首先移入最低有效位，可通过软件更改MSB或LSB，请参见本章寄存器描述。

22.3.5.2. 建议的接收步骤

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 LPUART_BRR 选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入 1 使能 LPUART。
5. 将 RE 位 LPUART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时

- RXNE 位置 1。表明已接收到并可读取数据。
- 如果 RXNEIE 位置 1，则会生成中断。
- 在单缓冲区模式下，通过软件对 LPUART_RDR 执行读操作将 RXNE 位清零。也可以通过将 LPUART_RQR 中的 RXFRQ 位置 1 将 RXNE 位清零。RXNE 位必须在结束接收下一个字符前清零，以避免发生上溢错误。

从发送端接收到中断字符时，LPUART 将会按照帧错误对其进行处理。检测到空闲帧时，处理步骤与接收到数据的情况相同；如果 IDLEIE 位置 1，则会产生中断。

22.3.5.3. 上溢错误

如果在 RXNE 未复位时接收到字符，则会发生上溢错误。RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志位都将置 1。发生上溢错误时：

- ORE 位置 1。
- 对 LPUART_RDR 执行读操作时可读到先前的数据。
- 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
- 如果 RXNEIE 位置 1 或 EIE 位置 1，则会生成中断。
- 通过对 ORECF 位写 1 清零 ORE 位。

22.3.5.4. 选择时钟源

通过 RCC_KSEL 寄存器的 LPUART1SEL 选择 LPUART 的工作时钟源。时钟源必须在使能 LPUART (UE 位置 1) 前选择。

必须遵循以下两个条件选择时钟源：

- 可在低功耗模式下使用 LPUART
- 通信速度

时钟源频率为 f_{CK} 。时钟源可以是以下其中之一： f_{PCLK} （默认值）、外部时钟。选择外部时钟作为时钟源可允许 LPUART 在 MCU 处于低功耗模式时接收数据。时钟源还决定通信速度范围（尤其是最大通信速度）。接收器每 bit 数据采样 1 次。尽可能在中间进行采样。

注：LPUART 不进行数据噪声检测。

22.3.5.5. 帧错误

如果接收数据时未在预期时间内识别出停止位，则会检测到帧错误。

检测到帧错误时：

- FE位由硬件置1。
- 无效数据从移位寄存器传送到LPUART_RDR。
- LPUART_CR3寄存器中的EIE位置1时将发出中断。
- 通过将1写入FECF位来复位FE位。

可通过LPUART_CR2中的STOP控制位配置接收停止位的数量（1个或2个）。1个停止位：将在第8、第9和第10次采样时对1个停止位进行采样。2个停止位：将在2个停止位的中间进行采样。RXNE标志在此采样期间置1。

22.3.6. 奇偶校验

通过LPUART_CR1中的PCE位决定是否开启奇偶校验功能。校验模式通过LPUART_CR1中的PS位选择。支持2种校验模式：偶校验（PS=0）、奇校验（PS=1）。

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由6个、7个或8个LSB位组成，具体取决于M位的值）。例如，如果数=00110101且4个位置1，则在选择偶校验（PS=0）时，校验位为0。

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由6个、7个或8个LSB位组成，具体取决于M位的值）。例如，如果数据=00110101且4个位置1，则在选择奇校验（PS=1）时，校验位为1。

接收数据时，在奇偶校验使能模式下，如果奇偶校验检查失败，则LPUART_SR寄存器中的PE标志置1；如果LPUART_CR1寄存器中PEIE位置1，则会生成中断。通过软件向PE位写入“1”来清零PE标志。

发送数据时，在奇偶校验使能模式下，则在数据寄存器中所写入数据的MSB位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS=00），则“1”的数量为偶数；如果选择奇校验（PS=01），则“1”的数量为奇数）。

22.3.7. 波特率生成

接收器和发送器（Rx和Tx）的波特率设置为LPUART_BRR中编程的相同值。LPUART的波特率计算公式如下：

$$\text{Tx/Rx 波特率} = \frac{8 * f_{CK}}{\text{LPUART_BRR}[14:0]}$$

注：

1. 对LPUART_BRR执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。
2. 禁止在LPUART_BRR寄存器中写入小于0x18的值。
3. fck 必须在 [3 x 波特率, 4096 x 波特率] 范围内。

LPUART 时钟源为 32768Hz 时可达到的最大波特率为 9600 波特。LPUART使用高频时钟源时，可以达到更高的波特率。

发送器误差

表 22-2 LPUART 时钟频率=32768 KHz 时发送的波特率误差

波特率 = 9600bps			
M位，停止位	波特率寄存器值	实际波特率	误差%
8位，1个停止位	0x001B	9637.65 bps	0.39
9位，1个停止位		9485.47 bps	1.19
7位，1个停止位		9513.29 bps	0.90
8位，2个停止位		9485.47 bps	1.19
9位，2个停止位		9590.64 bps	0.10
7位，2个停止位		9637.65 bps	0.39
波特率 = 4800bps			
M位，停止位	波特率寄存器值	实际波特率	误差%
8位，1个停止位	0x0036	4818.82 bps	0.39
9位，1个停止位		4805.97 bps	0.12
7位，1个停止位		4834.62 bps	0.72
8位，2个停止位		4805.97 bps	0.12
9位，2个停止位		4854.51 bps	1.14
7位，2个停止位		4818.82 bps	0.39
波特率 = 2400bps			
M位，停止位	波特率寄存器值	实际波特率	误差%
8位，1个停止位	0x006D	2391.82 bps	0.34
9位，1个停止位		2387.07 bps	0.54
7位，1个停止位		2397.66 bps	0.10
8位，2个停止位		2387.07 bps	0.54
9位，2个停止位		2397.66 bps	0.10
7位，2个停止位		2391.82 bps	0.34

接收器容差

表 22-3 fCK = 32768KHz 时接收器容差 (M=01,1 个停止位)

Lpuart 时钟频率 = 32768 Hz		
寄存器配置的波特率	通信最小波特率 (容差)	通信最大波特率 (容差)
9600 bps	9362.3 bps (2.47 %)	9929.6 bps (3.43 %)
4800 bps	4681.1 bps (2.48 %)	5041.2 bps (5.03 %)
2400 bps	2291.5 bps (4.52 %)	2520.6 bps (5.03 %)
1200 bps	1145.7 bps (4.52 %)	1265.1 bps (5.43 %)

表 22-4 时钟偏差对接收器的影响 (M=01,1 个停止位)

Lpuart 计算的时钟频率 = 32768 Hz		
通信波特率 (精确)	时钟最快频率 (偏差)	时钟最慢频率 (偏差)
9600 bps	33600 Hz (2.54 %)	31680 Hz (3.32 %)
4800 bps	33600 Hz (2.54 %)	31200 Hz (4.78 %)

2400 bps	34320 Hz (4.74 %)	31200 Hz (4.78 %)
1200 bps	34320 Hz (4.74 %)	31080 Hz (5.15 %)

注:

从表格中可以观察到需要的时钟频率比较准确。当时钟频率偏差 5% 时, 如果仍按照 32768hz 的时钟频率进行配置波特率将无法接收。

22.3.8. 自动波特率

LPUART 能够根据接收一个 0x7F 字符帧 (可以是 LSB 在前模式下的 0x7F 字符, 也可以是 MSB 在前模式下的 0xFE 字符) 检测并自动设置 LPUART_BRR 寄存器的值。开启 LPUART 接收使能后, 通过将 LPUART_CR1 寄存器中的 ABREN 位置 1 来激活自动波特率检测。之后 LPUART 会测量起始位的持续时间 (上升沿到上升沿) 根据该持续时间推测通信的波特率。

通过将 LPUART_SR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路繁忙, 则无法保证正确的波特率检测。这种情况下, BRR 值可能会损坏, ABRE 错误标志位将置 1。可随时通过复位 ABRF 标志 (通过 ABRRQ 位写入 1) 重新启动自动波特率检测。

注: 如果在自动波特率操作期间禁止 LPUART (UE=0), 则可能损坏 BRR 值。

使用 LPUART 的自动波特率计算功能, 建议的寄存器操作如下:

1. 通过 LPUART_CR1 寄存器配置字长为 8bit 通信。
2. 通过 RCC_KSEL 寄存器对 LPUART 工作时钟源进行编程。
(若 LPUART 工作时钟有切换动作则软件需要等待时钟切换稳定时间, 2 个系统时钟周期 + 2 个 32K 时钟周期)
1. 通过向 LPUART_CR1 寄存器中的 UE 和 RE 位写入 1 使能 LPUART。
2. 通过将 LPUART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。
3. LPUART 等待接收一个 0x7F 字符帧 (可以是 LSB 在前模式下的 0x7F 字符, 也可以是 MSB 在前模式下的 0xFE 字符)。之后 LPUART 会测量起始位的持续时间 (上升沿到上升沿) 根据该持续时间推测通信的波特率。
4. 等待 LPUART_SR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。

22.3.9. 单线半双工

LPUART 可以配置为遵循单线半双工协议, 其中 TX 和 RX 线路从内部相连接。使用 LPUART_CR3 中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

一旦向 HDSEL 位写入 1:

- TX 和 RX 线路从内部相连接。
- 不需要再使用 RX 引脚。
- 无数据传输时, TX 引脚处于释放状态, 它在空闲状态或接收过程中用作标准 I/O。这意味着, 必须将 TX 外接上拉电阻。

除此之外, 通信协议与正常 LPUART 模式下的通信协议相似。此线路上的任何冲突必须由软件管理。尤其要注意, 发送过程永远不会被硬件封锁, 只要数据是在 TE 位置 1 的情况下写入, 发送就会持续进行。

22.3.10. 多处理器模式

可以通过LPUART进行多处理器通信。例如，其中一个LPUART可以是主器件，其TX输出与其它LPUART的RX 输入相连接。其它LPUART为从LPUART，其各自的TX输出在逻辑上通过与运算连在一起，并与主LPUART的 RX输入相连接。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余LPUART服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将LPUART_CR1寄存器中的MME位置1。

在静默模式下：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- LPUART_SR中的RWU位置1。在某些情况下，RWU可以由硬件或软件通过LPUART_RQR中的MMRQ位自动控制。

根据WAKE 位的设置，LPUART可使用以下两种方法进入或退出静默模式：

- 若 WAKE位=0，则进行空闲线路检测。
- 若 WAKE位=1，则进行地址标记检测。

使用 LPUART 的接收功能，建议的寄存器操作如下：

1. 通过LPUART_CR1寄存器配置字长和开启静默模式（MME 位置 1），选择静默唤醒模式（WAKE=0，空闲线路 / WAKE=1，地址标记）。
2. 通过LPUART_BRR寄存器设置所需波特率。
3. 通过LPUART_CR2寄存器配置停止位数量。
（若选择地址检测方式唤醒静默模式，则需配置ADD和ADDM7）
4. 通过向 LPUART_CR1 寄存器中的 UE和RE 位写入 1 使能 LPUART。
5. 通过向 LPUART_RQR 寄存器中的 MMRQ 位写入 1 进入静默模式。

22.3.11. 空闲线路检测

WAKE=0

当向 MMRQ位写入1且RWU位自动置1时，LPUART进入静默模式。当检测到空闲帧时，它会被唤醒。此时RWU 位会由硬件清零，但LPUART_SR寄存器中的IDLE 位不会置1。

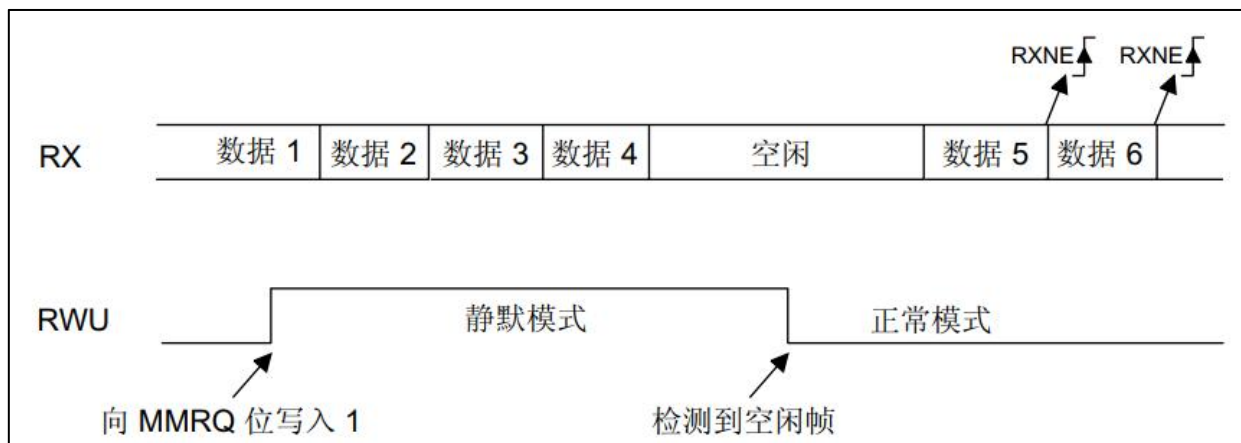


图 22-5 空闲线路检测时的静默模式

注：如果在空闲字符已经过去时将MMRQ位置1，将不会进入静默模式（RWU未置1）。

如果在线路处于空闲状态时激活 LPUART，在一个空闲帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

【空闲字符已经过去时，理解为RX线上高电平时间已经超过了空闲字符的长度】

4 位/7位地址标记检测 (WAKE=1)

在此模式下，如果字节的MSB为1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于4个或7个LSB中。7位或4位地址检测通过ADDM7位来选择。接收器会将此4位/7位字与其地址进行比较，该接收器的地址在LPUART_CR2的ADD位中进行设置。

注：在7位和9位数据模式下，地址检测分别在6位和8位地址上完成（ADD[5:0]和ADD[7:0]）。

当接收到与其编程地址不匹配的地址字符时，LPUART会进入静默模式。此时，RWU位将由硬件置1。

LPUART进入静默模式后，RXNE标志不会针对此地址字节置1。当向MMRQ位写入1时，LPUART也会进入静默模式。这种情况下，RWU位自动置1。当接收到与编程地址匹配的地址字符时，LPUART会退出静默模式。然后RWU位被清零，可以开始正常接收后续字节。由于RWU位已清零，RXNE位会针对地址字符置1。

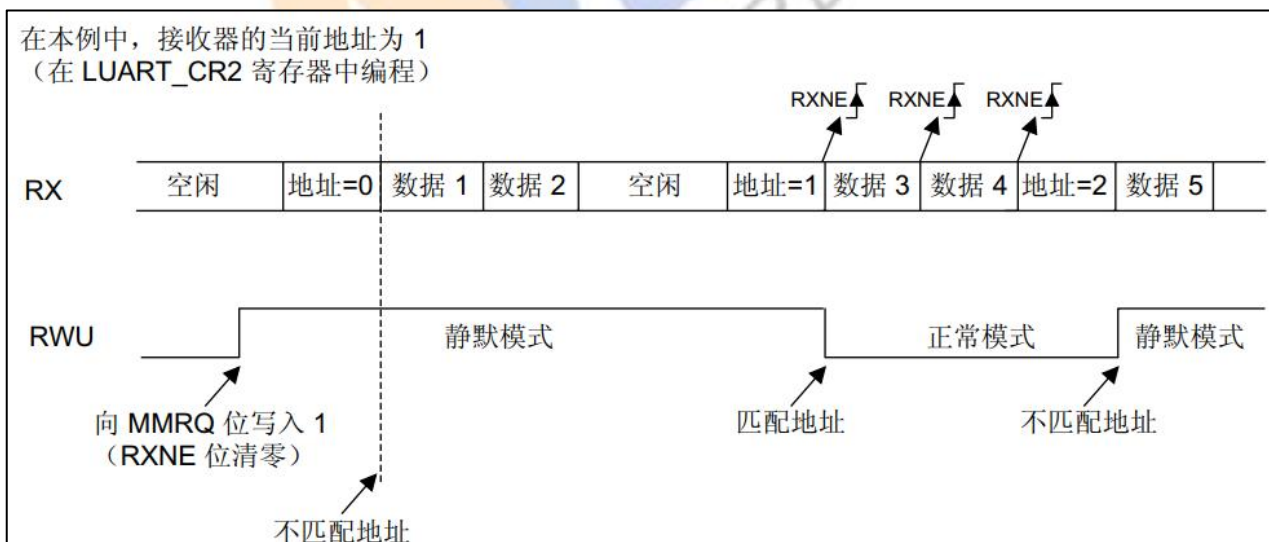


图 22-6 地址标记检测时的静默模式

22.3.12. 待机/停止模式唤醒

当UESM位置1且LPUART时钟设置为外部时钟时，LPUART能够将MCU从待机/停止模式唤醒。当接收到与编程地址匹配时将MCU唤醒。为了能够将MCU从停止模式唤醒，必须在进入停止模式前将LPUART_CR1中的UESM 位置1。检测到地址匹配后，WUF 标志会由硬件置1并在WUFIE位置1时生成一个唤醒中断。

注：在进入停止模式前，用户必须确保 LPUART未在执行传输。BUSY标志无法确保运行接收期间始终不进入停止模式。

22.3.13. 中断及异常说明

下表给出了 LPUART 中断请求列表。

表 22-5 LPUART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器为空	TXE	TXEIE
发送完成	TC	TCIE
接收数据寄存器不为空	RXNE	RXNEIE
检测到溢出错误	ORE	
检测到空闲字符	IDLE	IDLEIE
奇偶校验错误	PE	PEIE
多缓冲区通信中的溢出错误和帧错误	ORE或FE	EIE
正常模式字符匹配（8bit）	CMF	CMIE
从停止模式唤醒	WUF	WUFIE

LPUART 中断事件被连接到相同的中断向量。

- 发送：发送完成、发送数据寄存器为空中断。
 - 接收：空闲线路检测、上溢错误、接收数据寄存器不为空、奇偶校验错误、帧错误、字符匹配等。
- 如果相应的使能控制位置1，则这些事件会生成中断。

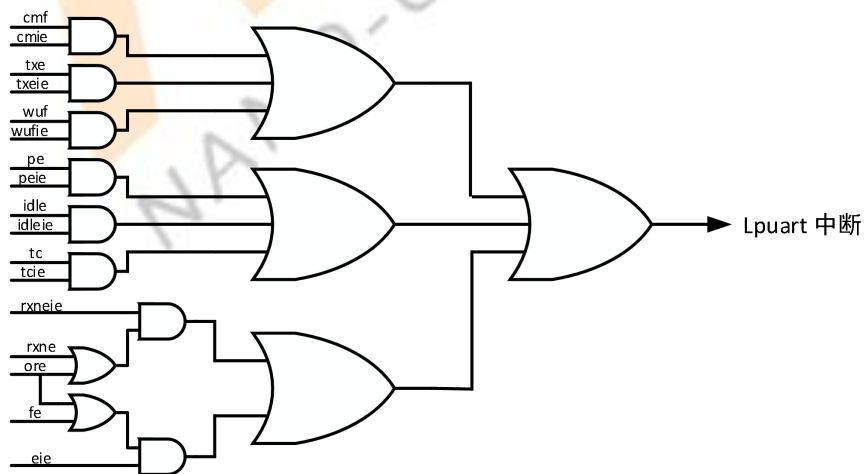


图 22-7 LPUART 中断映射图

LPUART 异常说明：

帧错误（FE）：接收模式下，未检测到停止位时发生帧错误。

溢出错误（ORE）：接收模式下，接收到下一个数据但尚未将上一个数据读走。

奇偶校验错误（PE）：接收模式下，接收到的校验位与硬件计算不符时发生奇偶校验错误。



22.4. LPUART 寄存器

22.4.1. LPUART 寄存器映射

表 22-6 LUPART 寄存器映射

偏移	寄存器名	访问	复位值
0x00	LPUART 版本寄存器 (VER)	r	0x0000_0001
0x04	LPUART 控制寄存器 1 (CR1)	rw	0x0000_0000
0x08	LPUART 控制寄存器 2 (CR2)	rw	0x0000_0000
0x0C	LPUART 控制寄存器 3 (CR3)	rw	0x0000_0000
0x10	LPUART 波特率配置寄存器 (BRR)	rw	0x0000_0000
0x14	LPUART 请求寄存器 (RQR)	rw	0x0000_0000
0x18	LPUART 状态寄存器 (SR)	r	0x0000_0080
0x1C	LPUART 清零寄存器 (CLR)	w	0x0000_0000
0x20	LPUART 置位寄存器 (SET)	w	0x0000_0000
0x28	LPUART 数据接收寄存器 (RDR)	r	0x0000_0000
0x2C	LPUART 数据发送寄存器 (TDR)	rw	0x0000_0000
0x30	LPUART 超时寄存器 (RTOR)	rw	0x0000_0000

22.4.2. LPUART 版本寄存器(VER)

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

位域	功能
31-0 ID	版本号

22.4.3. LPUART 控制寄存器 1 (CR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST	-	-	M1	-	-	-	-	-	-	CMEN	RTOEN	-	-	-	-
rw	-	-	rw	-	-	-	-	-	-	rw	rw	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ABREN	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31 RST	LPUART 复位信号(RESET), 写 1 清零
30-29	保留
28 M1	字长 (word length) 此位必须与位 12(M0)搭配使用来确定字长度。该位由软件置 1 或清零。 M[1:0]="00": 1 个起始位, 8 个数据位, n 个停止位 M[1:0]="01": 1 个起始位, 9 个数据位, n 个停止位 M[1:0]="10": 1 个起始位, 7 个数据位, n 个停止位 只有在禁止 LPUART(UE=0)时才能写入此位。
27-22	保留
21 CMEN	字符匹配模式使能 (Character match mode enable) 此位由软件置 1 和清零。 0: 禁止字符匹配模式。 1: 使能字符匹配模式。
20 RTOEN	接收器超时模式使能(Receiver Timeout Mode Enable) 0:禁止接收器超时模式 1:使能接收器超时模式
19-16	保留
15 ABREN	LPUART 自动波特率计算使能 (LPUART auto baudrate calculate mode enable) 此位使能或禁止 LPUART 自动波特率计算 0: 禁止 LPUART 自动波特率计算 1: 使能 LPUART 自动波特率计算
14 CMIE	字符匹配中断使能 (Character match interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断。 1: 如果 SR 寄存器中的 CMF 位置 1, 则生成 LPUART 中断。
13 MME	静默模式使能 (Mute mode enable) 此位用于激活 LPUART 的静默模式功能, 此位置 1 时, LPUART 可按 WAKE 位定义的方式在活动模式与静默模式之间切换。该位由软件置 1 和清零。 0: 接收器永久处于活动模式。 1: 接收器可在静默模式和活动模式之间切换。
12 M0	字长 (Word length) 此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零 (请参见位 28 (M1) 的说明)。 注: 只有在禁止 LPUART (UE=0) 时才能写入此位。
11 WAKE	接收器唤醒方法 (Receiver wakeup method) 此位用于确定 LPUART 静默模式的唤醒方法。该位由软件置 1 或清零。 0: 空闲线路 1: 地址标记 只有在禁止 LPUART (UE=0) 时才能写入此位域。
10 PCE	奇偶校验控制使能 (Parity control enable) 该位选择硬件奇偶校验控制 (生成和检测)。使能奇偶校验控制时, 计算出的奇偶校验位被插入到 MSB 位置 (如果 M=1, 则为第 9 位; 如果 M=0, 则为第 8 位), 并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1, PCE 在当前字节的后面处于活动状态 (在接收和发送时)。

位域	功能
	0: 禁止奇偶校验控制 1: 使能奇偶校验控制 只有在禁止 LPUART (UE=0) 时才能写入此位域。
9 PS	奇偶校验选择 (Parity selection) 该位用于在使能奇偶校验生成/检测 (PCE 位置 1) 时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。 0: 偶校验 1: 奇校验 只有在禁止 LPUART (UE=0) 时才能写入此位域
8 PEIE	PE 中断使能 (PE interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 SR 寄存器中的 PE=1 时, 生成 LPUART 中断
7 TXEIE	发送数据寄存器为空中断使能 (Transmit data register empty Interrupt Enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 SR 寄存器中 TXE/TXFNF=1 时, 生成 LPUART 中断
6 TCIE	传输完成中断使能 (Transfer complete interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 SR 寄存器中的 TC=1 时, 生成 LPUART 中断
5 RXNEIE	接收数据寄存器非空中断使能 (Receive data register not empty Interrupt Enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 SR 寄存器中的 ORE=1 或 RXNE=1 时, 生成 LPUART 中断
4 IDLEIE	IDLE 中断使能 (IDLE interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: SR 寄存器中的 IDLE=1 时, 生成 LPUART 中断
3 TE	发送器使能 (Transmitter enable) 该位使能发送器。该位由软件置 1 和清零。 0: 禁止发送器 1: 使能发送器
2 RE	接收器使能 (Receiver enable) 该位使能接收器。该位由软件置 1 和清零。 0: 禁止接收器 1: 使能接收器并开始搜索起始位
1 UESM	停止模式下的 LPUART 使能 (LPUART enable in Stop mode) 当此位清零时, LPUART 无法将 MCU 从低功耗模式唤醒。 当此位置 1 时, LPUART 能够将 MCU 从低功耗模式唤醒, 前提是 LPUART 时钟选择为 CLK32K 此位由软件置 1 和清零。 0: LPUART 无法将 MCU 从低功耗模式唤醒。 1: LPUART 能够将 MCU 从低功耗模式唤醒。前提是 LPUART 时钟选择为 CLK32K。 注: 如果该位仅在支持相关 function 的时候起作用。
0 UE	LPUART 使能 (LPUART enable) 0: 禁止 LPUART 预分频器和输出, 低功耗模式 1: 使能 LPUART

22.4.4. LPUART 控制寄存器 2 (CR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD								-	-	-	-	MSBFIRST	-	TXINV	RXINV
rw								-	-	-	-	rw	-	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	-	-	STOP	-	-	-	-	-	-	-	ADDM7	-	-	-	-
rw	-	-	rw	-	-	-	-	-	-	-	rw	-	-	-	-

位域	功能
31-24 ADD	LPUART 节点的地址 (Address of the LPUART node) ADD[7:4]: 这些位用于指定 LPUART 节点的地址或要识别的字符代码。 它在多处理器通信时于静默模式或停止模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。它们还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0]值进行比较，如果匹配，CMF 标志将置 1。 <i>仅在禁止接收 (RE = 0) 且禁止 LPUART (UE = 0) 时才能写入这些位。</i> ADD[3:0]: 这些位用于指定 LPUART 节点的地址或要识别的字符代码。 它在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。 <i>仅在禁止接收 (RE = 0) 且禁止 LPUART (UE = 0) 时才能写入这些位。</i>
23-20	保留
19 MSBFIRST	最高有效位在前(Most significant bit first) 此位由软件置 1 和清零。 0: 发送/接收数据时位 0 在前，后跟起始位。 1: 发送/接收数据时 MSB（位 7/8）在前，后跟起始位。 <i>只有在禁止 LPUART (UE=0) 时才能写入此位域。</i>
18	保留
17 TXINV	TX 引脚有效电平反向 (TX pin active level inversion) 此位由软件置 1 和清零。 0: TX 引脚信号使用标准逻辑电平（VDD = 1/空闲，Gnd = 0/标记）工作 1: 对 TX 引脚信号值取反。（VDD = 0/标记，Gnd=1/空闲）。 允许在 TX 线路上使用外部反相器。 <i>只有在禁止 LPUART (UE=0) 时才能写入此位域。</i>
16 RXINV	RX 引脚有效电平反向(RX pin active level inversion) 此位由软件置 1 和清零。 0: RX 引脚信号使用标准逻辑电平（VDD = 1/空闲，Gnd = 0/标记）工作 1: 对 RX 引脚信号值取反。（VDD = 0/标记，Gnd=1/空闲）。 允许在 RX 线路上使用外部反相器。 <i>只有在禁止 LPUART (UE=0) 时才能写入此位域。</i>
15 SWAP	交换 TX/RX 引脚 (Swap TX/RX pins) 此位由软件置 1 和清零。 0: 按标准引脚排列定义使用 TX/RX 引脚。 1: 交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。

位域	功能
	只有在禁止 LPUART (UE=0)时才能写入此位域。
14-13	保留
12 STOP	停止位 (STOP bits) 这些位用于编程停止位。 0: 1 个停止位 1: 2 个停止位 只有在禁止 LPUART (UE=0)时才能写入此位域。
11-5	保留
4 ADDM7	7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection) 此位用于选择 4 位地址检测或 7 位地址检测。 0: 4 位地址检测 1: 7 位地址检测 (在 8 位数据模式下) 只有在禁止 LPUART (UE=0) 时才能写入该位
3-0	保留

22.4.5. LPUART 控制寄存器 3 (CR3)

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

位域	功能
31-23	保留
22 WUFIE	从低功耗模式唤醒中断使能(Wakeup from low-power mode interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: 当 SR 寄存器中的 WUF=1 时, 生成 LPUART 中断
21-4	保留
3 HDSEL	半双工选择 (Half-duplex selection) 选择单线半双工模式 0: 未选择半双工模式 1: 选择半双工模式 只有在禁止 LPUART (UE=0)时才能写入此位。
2-1	保留
0 EIE	错误中断使能 (Error interrupt enable) 如果发生帧错误、上溢错误标志 (SR 寄存器中 FE = 1 或 ORE = 1), 则需要使用错误中断使能位来使能中断生成。 0: 禁止中断 1: 当 SR 寄存器中的 FE=1 或 ORE=1 时生成中断。

22.4.6. LPUART 波特率寄存器 (BRR)

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

位域	功能
31-15	保留
14-0 BRR	LPUART 波特率(LPUART baud rate) 禁止向 BRR 寄存器中写入小于 0x18 的值。 注意: 当使用高波特率通讯时, 必须谨慎选择 fck 值, fck 值必须在 $[3*baudrate, 4096*baudrate]$ 范围内。

22.4.7. LPUART 请求寄存器 (RQR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	--	-	-	-	--	-	-	-	-	--	-	-	-	-
-	-	-	--	-	-	-	--	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	--	-	-	-	-	-	-	-	-	-	-	RXFRQ	MMRQ	SBKRQ	ABRRQ
-	--	-	-	-	-	-	-	-	-	-	-	w	w	w	w

位域	功能
31-4	保留
3 RXFRQ	发送数据刷新请求 (Transmit data flush request) 向此位写 1 清零 RXNE 标志位 此位允许在没有读取数据时丢弃接收到的数据而避免 OVERRUN 条件。
2 MMRQ	禁默模式请求(Mute mode request) 向此位写入 1 可将 LPUART 置于静默模式, 并将 RWU 标志复位。
1 SBKRQ	发送中断请求 (Send break request) 向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK
0 ABRRQ	自动波特率请求(Auto baudrate request) 向此位写 1 清零 ABRF 标志, 并且产生下一帧接收数据的自动波特率测量请求。

22.4.8. LPUART 状态寄存器 (SR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	--	-	-	-	-	-	-	-	-	-	WUF	RWU	SBKF	CMF	BUSY

-		-	-	-	-	-	-	-	-	-	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	-	-	-	-	-	-	TXE	TC	RXNE	IDLE	ORE	-	FE	PE
r	r	-	-	-	-	-	-	r	r	r	r	r	-	r	r

位域	功能
31-21	保留
20 WUF	从停止模式唤醒标志 (Wakeup from stop mode flag) 当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。通过向 CLR 寄存器中的 WUCF 写入 1，此位由软件清零。 如果 CR3 寄存器中 WUFIE=1，则会生成中断
19 RWU	接收器从静默模式唤醒 (Receiver wakeup from mute mode) 此位指示 LPUART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。 静默模式控制序列（地址或 IDLE）通过 CR1 寄存器中的 WAKE 位选择。 当选择 IDLE 模式下唤醒时，该位只能通过用软件向 RQR 寄存器中的 MMRQ 位写 1 的方式置 1。 0：接收器处于工作模式 1：接收器处于静默模式
18 SBKF	发送中断标志 (Send break flag) 此位指示已请求发送中断字符。通过将 1 写入 CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。 0：不发送中断字符 1：将发送中断字符
17 CMF	字符匹配标志 (Character match flag) 接收到由 ADD[7:0]定义的字符后由硬件将此位置 1。通过向 CLR 寄存器中的 CMCF 写入 1，此位由软件清零。 如果 CR1 寄存器中 CMIE=1，则会生成中断。 0：未检测到字符匹配 1：检测到字符匹配
16 BUSY	忙标志 (Busy flag) 此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。 0：LPUART 处于空闲状态（无接收） 1：正在接收
15 ABRF	自动波特率标志 (Auto Baudrate flag) 在下列情形下，此位由硬件置 1 - 当自动波特率被置位时，（RXNE 也将被置位，并且如果 RXNEIE=1 时产生中断） - 当自动波特率操作不能成功完成（ABRE=1）（在这种情况下 ABRE，RXNE，FE 也被置位） 此位由软件清零。通过向 RQR 寄存器的 ABRRQ 位写 1 来产生自动波特率并清零该位。
14 ABRE	自动波特率错误 (Auto Baudrate Error) 如果发生波特率超出范围或者字符比较失败时，该位由硬件置 1。 此位通过向 ABRRQ 写 1，由软件清零

位域	功能
13-8	保留
7 TXE	发送数据寄存器为空 (Transmit data register empty) 此位在 LPUART TDR 寄存器中的内容被转移到移位寄存器中后由硬件置 1。 如果 TXEIE = 1 将会产生中断。 0: 数据没有被转移到移位寄存器 1: 数据已经被转移到移位寄存器
6 TC	数据发送完成标志位 (Data Transmission Complete) 在一帧数据被传送完成时由硬件置位。如果 TCIE = 1, 将会产生中断。 通过写 TDR 寄存器或 CLR[TCCF] = 1 清零此位。 如果 TCIE=1 将会产生中断。 0: 发送没有完成 1: 发送完成
5 RXNE	读取数据寄存器不为空 (Read data register not empty) 当 RDR 移位寄存器的内容已传输到 RDR 寄存器时, RXNE 位由硬件置 1。通过对 RDR 寄存器执行读取操作将该位清零。也可以通过将 RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。 如果 RXNEIE = 1, 则会生成中断。 0: 未接收到数据 1: 已准备好读取接收到的数据
4 IDLE	检测到空闲线路 (IDLE line detected) 检测到空闲线路时, 该位由硬件置 1。如果 IDLEIE=1, 则会生成中断。通过向 CLR 寄存器中的 IDLECF 写入 1, 此位由软件清零。 0: 未检测到空闲线路 1: 检测到空闲线路
3 ORE	溢出错误 (Overrun error) 在 RXNE=1 的情况下, 当移位寄存器中当前正在接收的数据准备好传输到 RDR 寄存器时, 此位由硬件置 1。 通过向 CLR 寄存器中的 ORECF 写入 1, 此位由软件清零。 如果 RXNEIE=1 或 EIE = 1, 则会生成中断。 0: 无溢出错误 1: 检测到溢出错误
2	保留
1 FE	帧错误 (Framing Error) 当检测到同步错误, 中断字符或者过度噪声时此位由硬件置 1。 通过向 CLR 寄存器的 FECF 位写 1, 此位由软件清零。 如果 EIE = 1 将会产生中断 0: 没有检测到帧错误 1: 检测到帧错误或者中断字符
0 PE	奇偶校验错误 (Parity error) 当在接收器模式下发生奇偶校验错误时, 该位由硬件置 1。通过向 CLR 寄存器中的 PECF 写入 1, 此位由软件清零。 如果 PEIE = 1, 则会生成中断。 0: 无奇偶校验错误 1: 奇偶校验错误

22.4.9. LPUART 清零寄存器（CLR）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	WUCF	-	-	CMFCF	-
-	-	-	-	-	-	-	-	-	-	-	w	-	-	w	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	TCCF	-	IDLECF	ORECF	-	FECF	PECF
-	-	-	-	-	-	-	-	-	w	-	w	w	-	w	w

位域	功能
31-21	保留
20 WUCF	从停止模式唤醒清除标志 (Wakeup from stop mode clear flag) 将 1 写入此位时，SR 寄存器中 WUF 标志将清零。
19-18	保留
17 CMFCF	字符匹配清零标志 (Character match clear flag) 将 1 写入此位时，SR 寄存器中 CMF 标志将清零。
16-7	保留
6 TCCF	数据发送完成清零标志 (Data Transmission Complete clear flag) 将 1 写入此位时，SR 寄存器中 IDLE 标志将清零。
5	保留
4 IDLECF	检测到空闲线路清零标志 (Idle line detected clear flag) 将 1 写入此位时，SR 寄存器中 IDLE 标志将清零。
3 ORECF	上溢错误清零标志 (Overrun error clear flag) 将 1 写入此位时，SR 寄存器中 ORE 标志将清零。
2	保留
1 FECF	帧错误清零标志 (Framing error clear flag) 将 1 写入此位时，SR 寄存器中 FE 标志将清零。
0 PECF	奇偶校验错误清零标志 (Parity error clear flag) 将 1 写入此位时，SR 寄存器中 PE 标志将清零。

22.4.10. LPUART 接收数据寄存器（RDR）

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

位域	功能
31-9	保留

位域	功能
8-0 RDR	接收数据值 (Receive data value) 包含接收到的数据字符。 RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口。在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位

22.4.11. LPUART 发送数据寄存器 (TDR)

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

位域	功能
31-9	保留
8-0 TDR	发送数据值 (Transmit data value) 包含要发送的数据字符。 TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口。在使能奇偶校验位的情况下 (CR1 寄存器中的 PCE 位被置 1) 进行发送时，由于 MSB 的写入值 (位 7 或位 8，具体取决于数据长度) 会被奇偶校验位所取代，因此该值不起任何作用。

22.4.12. LPUART 超时寄存器 (RTOR)

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

位域	功能
31-9	保留
8-0 RTO	超时值

23. 唤醒定时器和蜂鸣器产生电路(AWU)

23.1. 简介

芯片内部集成一个唤醒定时器和一个蜂鸣器产生电路，唤醒定时器和蜂鸣器产生电路均采用低频振荡器作为时钟，唤醒定时器采用一个 20 位的递增计数器作为定时器，当计数到设定值时可产生一个唤醒信号。蜂鸣器产生电路采用一个 5 位的递增计数器，可产生 500Hz 到 16KHz 的信号。

23.2. 主要特性

基本定时器具有以下特性：

- 可选低频时钟 LSI/LSE 作为时钟源
- AWU 采用 20 位递增计数器
- 可配置在 STOP 模式下运行并唤醒 STOP
- 可产生 500Hz 到 16KHz 的信号

23.3. 唤醒定时器功能说明

唤醒定时器可以产生周期性的中断信号。该定时器采用一个 20 位的计数器，其时钟源可通过设置 RCC 中 KSEL 寄存器 AWU_SEL 位选择 LSI/LSE 时钟。计时的时间可以配置为 $T_{AWUCLK} * (TIMEOUT+1) * 16384$ 。定时器计数到预设的匹配值时会归零重新计数，也可以通过写 AWUCON 中的 CLR 位清零。

唤醒定时器产生的中断信号维持一个 LSI/LSE 时钟周期。

其内部框图如下：

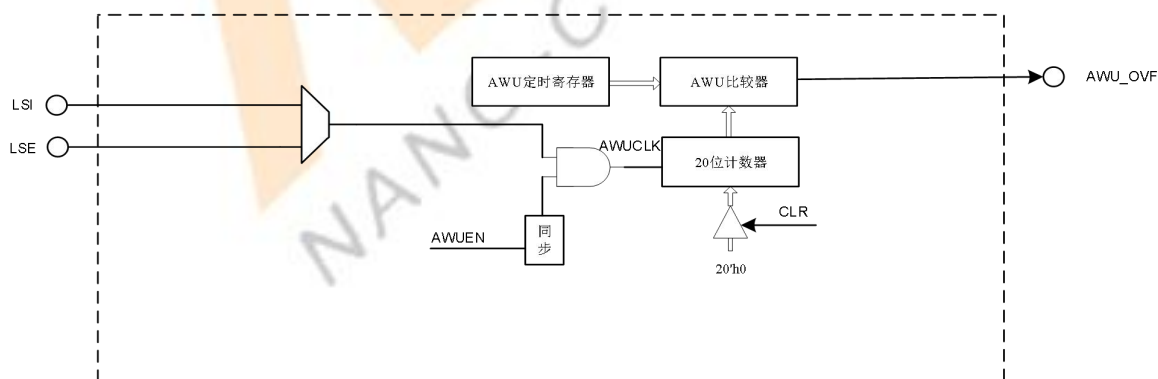


图 23-1 唤醒定时器内部框图

23.4. 蜂鸣器产生电路功能说明

BUZZER 电路输出的 BUZ 信号范围为 500~16KHz，其时钟源可通过设置 RCC 中 KSEL 寄存器 AWU_SEL 位选择 LSI/LSE 时钟，可由 BUZCON 中的 PSC 位确定输出频率。其内部框图如图如下：

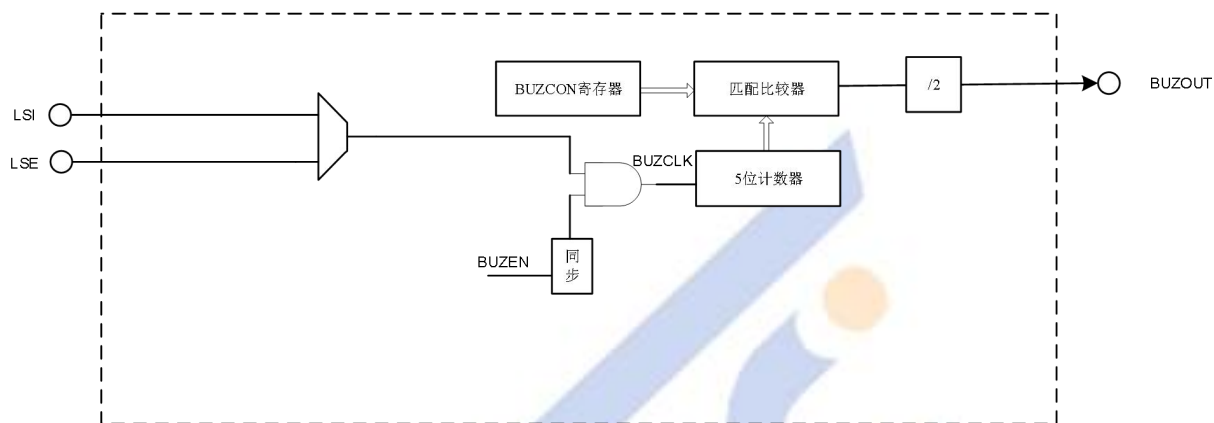


图 23-2 BUZZER 内部框图

23.5. AWU 寄存器

23.5.1. AWU 寄存器映射

表 23-1 AWU 寄存器总表

偏移	寄存器名	访问	复位值
0x00	BUZ 控制寄存器 (BUZCON)	rw	0x0000_0000
0x04	AWU 控制寄存器 (AWUCON)	rw	0x0000_0000

23.5.2. BUZ 控制寄存器 (BUZCON)

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

位域	功能
31-8	保留
7-3 PSC	频率分频 (BUZ frequency division) 分频频率 = $f_{BUZCLK} / (2 * (PSC + 1))$
2-1	保留
0 BUZEN	BUZ 使能 (BUZ enable) 0: 禁止 1: 使能

23.5.3. AWU 控制寄存器 (AWUCON)

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

位域	功能
31-8	保留
7-2 TIMEOUT	唤醒定时器超时设置 (AWU timeout) 超时时间= $T_{AWUCLK} * (TIMEOUT+1) * 16384$
1 CLR	唤醒定时器清零 写 1 有效
0 AWUEN	AWU 使能 (AWU enable) 0: 禁止 1: 使能



24. 触摸(TOUCH)

24.1. 简介

任何两个导电的物体之间都存在着感应电容，一个按键即一个焊盘与大地也可构成一个感应电容。电容触摸按键感应原理是利用人体的感应电容来检测是否有手指存在，在没有手指按下时，按键上由于分布电容的存在，因此按键对地存在一定的静态电容，在周围环境不变的情况下，该感应电容值是固定不变的微小值

当人的手指按下或者接近按键时，人体的寄生电容将耦合到静态电容上，使按键的最终电容值变大，该变化的电容信号再输入到单片机进行信号转换，将变化的电容量转换成某种电信号的变化量，再由一定的算法来检测和判断这个变化量的程度，当这个变化量超过一定的域值时就认为手指按下，然后将输出某个按键被按下的确定信号。

如下图所示在 PCB 上构建的电容器，电容式触摸感应按键实际上只是 PCB 上的一小块“覆铜焊盘”，触摸按键与周围的“地信号”构成一个感应电容，当手指靠近电容上方区域时，它会干扰电场，从而引起电容相应变化。根据这个电容量的变化，可以检测是否有人体接近或接触该触摸按键。

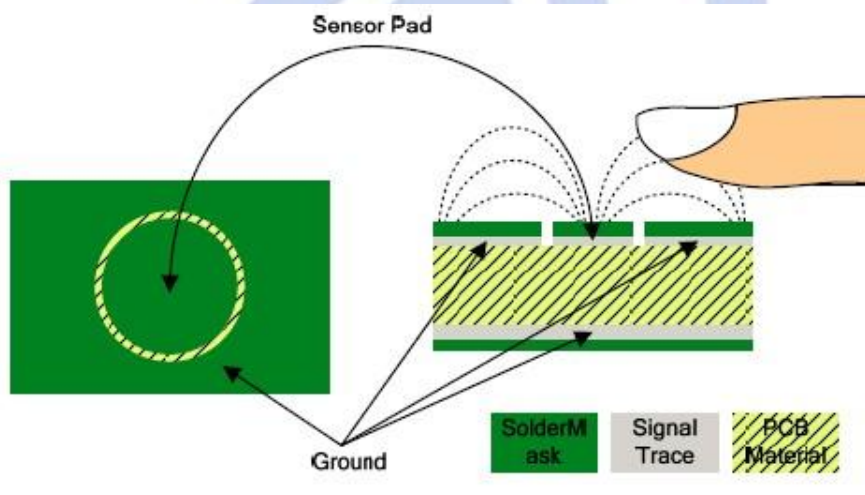


图 24-1 PCB 上构建开放式电容器示意图

接地板通常放置在按键板的下方，用于屏蔽其它电子产品产生的干扰。此类设计受 PCB 上的寄生电容和温度以及湿度等环境因素的影响，检测系统需持续监控和跟踪此变化并做出基准值调整。

基准电容值由特定结构的 PCB 产生，介质变化时，电容大小亦发生变化。

当人体（手指）接触金属感应片的时候，由于人体相当于一个接大地的电容，因此会在感应片和大地之间形成一个电容，感应电容量通常有几 pF 到几十 pF。利用这个最基本的原理，在外部搭建相关电路，就可以根据这个电容量的变化，检测是否有人体接触金属感应片。

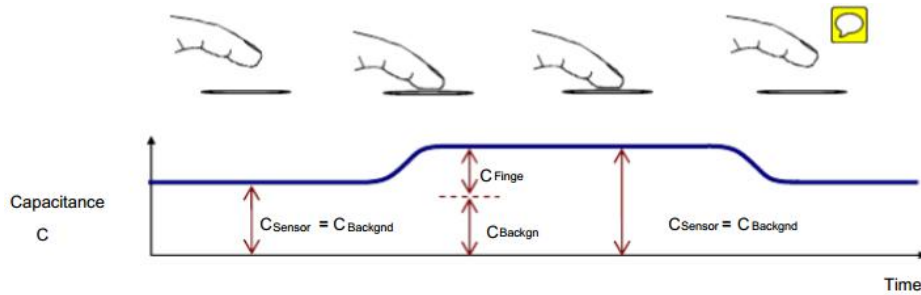


图 24-2 触摸按键的电容变化

24.2. 主要特性

TOUCH 控制模块主要有以下特性：

- 自动使能 ADC,并在按键扫描结束后硬件触发 ADC 开始转换
- 硬件自动触发 ADC 转换时提前使能 ADC 时间可配
- 按键检测时间间隔可配
- 按键充放电次数可配
- 正常模式：
 - 工作时钟为系统时钟
 - 比较输出，输出极性可配
 - 支持 8 档充放电频率（8MHz，6MHz，4MHz，2MHz，1MHz，500KHz，250KHz，125KHz）
 - 支持单个按键连续检测，检测时间间隔可设，充放电次数可设
 - 支持多个按键自动扫描，最多支持 16 个扫描通道，每个按键扫描时间间隔可设
- 低功耗模式：
 - 工作时钟为较快内部低振（512KHz）、较慢内部低振（32KHz）或外部时钟（32.768KHz）
 - 充放电频率为工作时钟的二分频
 - 单个按键唤醒连续检测，检测时间间隔可设，充放电次数可设
 - 支持多个按键轮询检测，最多支持 16 个通道，每个按键检测时间间隔可设
 - 低功耗唤醒时各通道的匹配唤醒阈值可配
 - 可查看 16 个按键匹配唤醒状态

24.3. 功能说明

24.3.1. 单个按键连续检测模式

TOUCH 可以工作在单个按键连续检测模式下。当 TOUCH 充放电结束后硬件会自动触发 ADC 转换，通过读 TK_CHWORK[3:0]位可知当前转换 ADC 数据对应的 TOUCH 通道。

通过配置 CHSEL，仅使能其中一个通道有效，即为单个按键连续检测模式。配置连续检测次数（NUM）、充放电频率（CHAREG_TIME）和间隔时间（INTERVAL_TIME）。开启 TKEN 使能 TOUCH 开始单个按键连续检测。退出单个按键连续检测模式只需将 TKEN 清零。硬件会自动在 TOUCH 充放电结束后触发 ADC 转换。

24.3.2. 多个按键连续检测模式

通过配置 CHSEL，使能多个通道有效，即为多个按键连续检测模式。配置连续检测次数（NUM）、充放电频率（CHAREG_TIME）和间隔时间（INTERVAL_TIME）。开启 TKEN 使能 TOUCH 开始多个按键依次连续检测。退出多个按键连续检测模式只需将 TKEN 清零。硬件会自动在 TOUCH 充放电结束后触发 ADC 转换。

24.3.3. 低功耗唤醒

当芯片进入低功耗模式，采用 TOUCH 唤醒功能时，在进入低功耗模式前，须先使能 LSI/LSE，将 TOUCH 时钟切换到 LSI/LSE，设置 ADC 模块进入低功耗，TOUCH 中断挂载在 EXTI_LINE19，使能外部中断，方可通过 TOUCH 唤醒。

24.3.4. 触发模式

24.3.4.1. 硬件触发模式

硬件触发模式可参考上述 24.3.1 及 24.3.2 章节中关于单个/多个按键连续检测模式所描述的内容，进行配置，此处不再赘述。

24.3.4.2. 软件触发模式

当 TOUCH 选择软件触发时，必须将 TOUCH 关闭再使能，完成一次充电，检测到充电完成标志后需立即使用软件触发 ADC 转换（操作 ADC_CTL[ADGO]=1），方可采到正确的触摸值。

需要注意的是，在设置软件触发模式时，必须将 ADC 采集通道设置成 CT 引脚的 ADC 通道 18。

24.3.5. 中断说明

24.3.5.1. 按键充放电检测完成中断

硬件触发 ADC 转换会在 ADC 转换完成时产生中断。

软件触发 ADC 转换会在 TOUCH 充电完成后产生中断。

24.3.5.2. 按键检测匹配中断

将转换的 ADC 数据与设定的阈值进行比较，若结果大于等于阈值则计数加一，若最终的结果符合设置的比较模式，可以产生中断。

24.4. TOUCH 寄存器

24.4.1. TOUCH 寄存器映射

表 24-1 TOUCH 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	TOUCH 版本寄存器 (VER)	r	0x0000_0001
0x04	TOUCH 控制寄存器 (CR)	rw	0x0000_0000
0x08	TOUCH 配置寄存器 1 (CFGR1)	rw	0x0000_0000
0x0C	TOUCH 配置寄存器 2 (CFGR2)	rw	0x0000_0000
0x10	TOUCH 通道选择寄存器 (CHSEL)	rw	0x0000_0000
0x14	TOUCH 匹配控制寄存器 (MATCR)	rw	0x0000_0000
0x18	TOUCH 中断状态寄存器 (ISR)	rw	0x0000_0000
0x1C	TOUCH 中断状态清除寄存器 (ICR)	rw	0x0000_0000
0x20~0 x5C	TOUCH 通道 x 阈值寄存器 (THx, x=0~15)	rw	0x0000_0000

24.4.2. TOUCH 版本寄存器(VER)

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

位域	功能
31-0	IP 版本号

24.4.3. TOUCH 控制寄存器 (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TK_NEC OVIE	TK_CLKSEL	TK_MAT IE	TK_CRGI E	-	TK_ADC TRG-	TK_DIS	TK_EN	
-	-	-	-	-	-	-	rw	rw	rw	rw	-	rw	rw	rw	

位域	功能
----	----

位域	功能
31-9	保留
8 TK_NECOVIE	充放电间隔时间中 ADC 转换未完成中断 0: 禁止充放电间隔时间中 ADC 转换未完成中断 1: 使能充放电间隔时间中 ADC 转换未完成中断
7-6 TK_CLKSEL	工作时钟选择 00: 选择系统时钟 01: 选择较快内部低速时钟 (512KHz) 10: 选择较慢内部低速时钟 (32KHz) 11: 选择外部时钟 (32.768KHz)
5 TK_MATIE	按键检测匹配中断使能 0: 禁止按键检测匹配中断 1: 使能按键检测匹配中断
4 TK_CRGIE	按键充放电检测完成中断使能 0: 禁止按键充放电检测完成中断 1: 使能按键充放电检测完成中断 备注: 硬件触发 ADC 转换会在 ADC 转换完成时产生中断 软件触发 ADC 转换会在 TOUCH 充电完成后产生中断
3	保留
2 TK_CRGIE	触发 ADC 模式: 0: 硬件自动触发 ADC 转换 1: 软件配置触发 ADC 转换
1 TK_CRGIE	软件配置放电 0: 不放电 1: 放电
0 TK_EN	TOUCH 使能: 0: 禁止 TOUCH 模块 1: 使能 TOUCH 模块

24.4.4. TOUCH 控制寄存器 (CFGR1)

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

位域	功能
31-16	保留
15-8 TK_CRGNUM	按键的充电次数配置 实际充电次数 = $64 * (TK_CRGNUM[7:0] + 1)$ 实际充电次数范围 64~16384
7-4 TK_MATIE	两通道一次完整充放电之间的间隔时间 0000: $ITVL_UNIT * 1 \approx 62.5\mu s$

位域	功能
	0001: ITVL_UNIT*2 $\approx$ 125us 0010: ITVL_UNIT*3 $\approx$ 187.5us 0011: ITVL_UNIT*4 $\approx$ 250us 0100: ITVL_UNIT*5 $\approx$ 312.5us 0101: ITVL_UNIT*6 $\approx$ 375us 0110: ITVL_UNIT*7 $\approx$ 437.5us 0111: ITVL_UNIT*8 $\approx$ 500us 1000: ITVL_UNIT*12 $\approx$ 750us 1001: ITVL_UNIT*16 $\approx$ 1ms 1010: ITVL_UNIT*24 $\approx$ 1.5ms 1011: ITVL_UNIT*32 $\approx$ 2ms 1100: ITVL_UNIT*40 $\approx$ 2.5ms 1101: ITVL_UNIT*48 $\approx$ 3ms 1110: ITVL_UNIT*56 $\approx$ 3.5ms 1111: ITVL_UNIT*64 $\approx$ 4ms
3	保留
2-0 TK_CRGIE	按键充放电信号频率（正常模式下有效） 000: 工作时钟的 6 分频（工作时钟为 48M, Fsw = 8MHz） 001: 工作时钟的 8 分频（工作时钟为 48M, Fsw = 6MHz） 010: 工作时钟的 12 分频（工作时钟为 48M, Fsw = 4MHz） 011: 工作时钟的 24 分频（工作时钟为 48M, Fsw = 2MHz） 100: 工作时钟的 48 分频（工作时钟为 48M, Fsw = 1MHz） 101: 工作时钟的 96 分频（工作时钟为 48M, Fsw = 500KHz） 110: 工作时钟的 192 分频（工作时钟为 48M, Fsw = 250KHz） 111: 工作时钟的 384 分频（工作时钟为 48M, Fsw = 125KHz） 备注：低功耗模式下充放电信号频率 Fsw 为工作时钟的二分频

24.4.5. TOUCH 控制寄存器 (CFGR2)

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

位域	功能
31-10	保留
9-0 TK_ENADCTIME	硬件自动触发 ADC 转换模式时的提前使能 ADC 时间 $Tenadc_time = (TK_ENADCTIME[9:0] + 1) * Tsw$ $Tsw = 1/Fsw$

24.4.6. TOUCH 控制寄存器 (CHSEL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TK15_EN	TK14_EN	TK13_EN	TK12_EN	TK11_EN	TK10_EN	TK9_EN	TK8_EN	TK7_EN	TK6_EN	TK5_EN	TK4_EN	TK3_EN	TK2_EN	TK1_EN	TK0_EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	功能
31-16	保留
15 TK15_EN	TOUCH 通道 15 使能 0: 禁止该通道 1: 使能该通道
14 TK14_EN	TOUCH 通道 14 使能 0: 禁止该通道 1: 使能该通道
13 TK13_EN	TOUCH 通道 13 使能 0: 禁止该通道 1: 使能该通道
12 TK12_EN	TOUCH 通道 12 使能 0: 禁止该通道 1: 使能该通道
11 TK11_EN	TOUCH 通道 11 使能 0: 禁止该通道 1: 使能该通道
10 TK10_EN	TOUCH 通道 10 使能 0: 禁止该通道 1: 使能该通道
9 TK9_EN	TOUCH 通道 9 使能 0: 禁止该通道 1: 使能该通道
8 TK8_EN	TOUCH 通道 8 使能 0: 禁止该通道 1: 使能该通道
7 TK7_EN	TOUCH 通道 7 使能 0: 禁止该通道 1: 使能该通道
6 TK6_EN	TOUCH 通道 6 使能 0: 禁止该通道 1: 使能该通道
5 TK5_EN	TOUCH 通道 5 使能 0: 禁止该通道 1: 使能该通道
4 TK4_EN	TOUCH 通道 4 使能 0: 禁止该通道

位域	功能
	1: 使能该通道
3 TK3_EN	TOUCH 通道 3 使能 0: 禁止该通道 1: 使能该通道
2 TK2_EN	TOUCH 通道 2 使能 0: 禁止该通道 1: 使能该通道
1 TK1_EN	TOUCH 通道 1 使能 0: 禁止该通道 1: 使能该通道
0 TK0_EN	TOUCH 通道 0 使能 0: 禁止该通道 1: 使能该通道

24.4.7. TOUCH 控制寄存器 (MATCR)

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

位域	功能
31-6	保留
5-4 TK_MATMODE	低功耗模式时按键的检测匹配选择: 00: 按键检测后的 ADC 转换数据都符合设定的阈值 01: 按键检测后的 ADC 转换数据最多 1 次未符合设定的阈值 10: 按键检测后的 ADC 转换数据最多 2 次未符合设定的阈值 11: 按键检测后的 ADC 转换数据最多 3 次未符合设定的阈值
3-0 TK_SWNUM	低功耗模式按键的 ADC 比较次数=SWNUM[3:0] + 4

24.4.8. TOUCH 控制寄存器 (ISR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	TK_NEC	TK_CH15	TK_CH14	TK_CH13	TK_CH12	TK_CH11
-	-	-	-	-	-	-	-	-	-	OVF	_MATF	_MATF	_MATF	_MATF	1_MATF
-	-	-	-	-	-	-	-	-	-	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TK_CH10	TK_CH9	TK_CH8	TK_CH7	TK_CH6	TK_CH5	TK_CH4	TK_CH3	TK_CH2	TK_CH1	TK_CH0	TK_CRG	TK_CH1			
_MATF	MATF	MATF	MATF	MATF	MATF	MATF	MATF	MATF	MATF	MATF	F	MATF			
r	r	r	r	r	r	r	r	r	r	r	r	r			

位域	功能
31-22	保留
21 TK_NECOVF	充放电间隔时间中 ADC 转换未完成中断标志 0: 充放电间隔时间中 ADC 转换已完成 1: 充放电间隔时间中 ADC 转换未完成
20 TK_CH15_MATF	TOUCH 通道 CH15 匹配中断标志 0: 低功耗模式下 CH15 未匹配通过 1: 低功耗模式下 CH15 匹配通过
19 TK_CH14_MATF	TOUCH 通道 CH14 匹配中断标志 0: 低功耗模式下 CH14 未匹配通过 1: 低功耗模式下 CH14 匹配通过
18 TK_CH13_MATF	TOUCH 通道 CH13 匹配中断标志 0: 低功耗模式下 CH13 未匹配通过 1: 低功耗模式下 CH13 匹配通过
17 TK_CH12_MATF	TOUCH 通道 CH12 匹配中断标志 0: 低功耗模式下 CH12 未匹配通过 1: 低功耗模式下 CH12 匹配通过
16 TK_CH11_MATF	TOUCH 通道 CH11 匹配中断标志 0: 低功耗模式下 CH11 未匹配通过 1: 低功耗模式下 CH11 匹配通过
15 TK_CH10_MATF	TOUCH 通道 CH10 匹配中断标志 0: 低功耗模式下 CH10 未匹配通过 1: 低功耗模式下 CH10 匹配通过
14 TK_CH9_MATF	TOUCH 通道 CH9 匹配中断标志 0: 低功耗模式下 CH9 未匹配通过 1: 低功耗模式下 CH9 匹配通过
13 TK_CH8_MATF	TOUCH 通道 CH8 匹配中断标志 0: 低功耗模式下 CH8 未匹配通过 1: 低功耗模式下 CH8 匹配通过
12 TK_CH7_MATF	TOUCH 通道 CH7 匹配中断标志 0: 低功耗模式下 CH7 未匹配通过 1: 低功耗模式下 CH7 匹配通过
11 TK_CH6_MATF	TOUCH 通道 CH6 匹配中断标志 0: 低功耗模式下 CH6 未匹配通过 1: 低功耗模式下 CH6 匹配通过
10 TK_CH5_MATF	TOUCH 通道 CH5 匹配中断标志 0: 低功耗模式下 CH5 未匹配通过 1: 低功耗模式下 CH5 匹配通过
9 TK_CH4_MATF	TOUCH 通道 CH4 匹配中断标志 0: 低功耗模式下 CH4 未匹配通过 1: 低功耗模式下 CH4 匹配通过
8 TK_CH3_MATF	TOUCH 通道 CH3 匹配中断标志 0: 低功耗模式下 CH3 未匹配通过 1: 低功耗模式下 CH3 匹配通过
7 TK_CH2_MATF	TOUCH 通道 CH2 匹配中断标志 0: 低功耗模式下 CH2 未匹配通过 1: 低功耗模式下 CH2 匹配通过

位域	功能
6 TK_CH1_MATF	TOUCH 通道 CH1 匹配中断标志 0: 低功耗模式下 CH1 未匹配通过 1: 低功耗模式下 CH1 匹配通过
5 TK_CH0_MATF	TOUCH 通道 CH 匹配中断标志 0: 低功耗模式下 CH0 未匹配通过 1: 低功耗模式下 CH0 匹配通过
4 TK_CRGF	按键充放电检测完成中断标志 0: 当前按键充放电未完成 1: 当前按键充放电已完成 备注: 硬件触发 ADC 转换会在 ADC 转换完成时产生中断 软件触发 ADC 转换会在 TOUCH 充电完成后产生中断
3-0 TK_CHWORK	当前 ADC 转换数据对应的通道 0000: 当前数据对应 TOUCH 通道 0 0001: 当前数据对应 TOUCH 通道 1 0010: 当前数据对应 TOUCH 通道 2 0011: 当前数据对应 TOUCH 通道 3 0100: 当前数据对应 TOUCH 通道 4 0101: 当前数据对应 TOUCH 通道 5 0110: 当前数据对应 TOUCH 通道 6 0111: 当前数据对应 TOUCH 通道 7 1000: 当前数据对应 TOUCH 通道 8 1001: 当前数据对应 TOUCH 通道 9 1010: 当前数据对应 TOUCH 通道 10 1011: 当前数据对应 TOUCH 通道 11 1100: 当前数据对应 TOUCH 通道 12 1101: 当前数据对应 TOUCH 通道 13 1110: 当前数据对应 TOUCH 通道 14 1111: 当前数据对应 TOUCH 通道 15

24.4.9. TOUCH 控制寄存器 (ICR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TK_NEC OVC	-	-	TK_MAT C	TK_CRG C	-	-	-	-
-	-	-	-	-	-	-	w	-	-	w	w	-	-	-	-

位域	功能
31-9	保留
8 TK_NECOV C	清除充放电间隔时间中 ADC 转换未完成中断标志 0: 无 1: 清除充放电间隔时间中 ADC 转换未完成中断标志

位域	功能
	写 1 硬件自动清 0
7-6	保留
5 TK_MATC	清除所有通道的匹配中断标志 0: 无 1: 清除所有通道的匹配中断标志 写 1 硬件自动清 0
4 TK_CRGC	清除按键充放电检测完成中断标志 0: 无 1: 清除按键充放电检测完成中断标志 写 1 硬件自动清 0
3-0	保留

24.4.10. TOUCH 控制寄存器 (MATxTH)(x=0~15)

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

TK_CHxTH

rw

位域	功能
31-12	保留
11-0 TK_CHxTH	TOUCH 低功耗模式时各通道的匹配唤醒阈值 若该按键通道的 ADC 转换数据大于该阈值，认为此按键被触摸，对应通道标志位置 1。

25. 液晶显示（LCD）

25.1. 简介

LCD 控制器支持控制驱动 4*32 或 5*31 的单色被动液晶显示屏。LED 模式下，支持控制驱动 4*28 或 8*28 矩阵。

25.2. 主要特性

- LCD 支持 4*32（1/4 占空比，1/3 偏压）
- LCD 支持 5*31（1/5 占空比，1/3 偏压）
- LCD 驱动支持电阻型和电容型偏压产生电路
- 电阻型 LCD 驱动支持软件对比度调节和快速充电模式以降低功耗
- 电容型 LCD 驱动分为电容偏压型和内建电压稳定器型两种驱动方式
- 闪烁功能：可配置频率 0.5/1/2/4Hz 闪烁
- LED 支持 4*28 或 8*28

25.3. 功能说明

25.3.1. LCD 框图

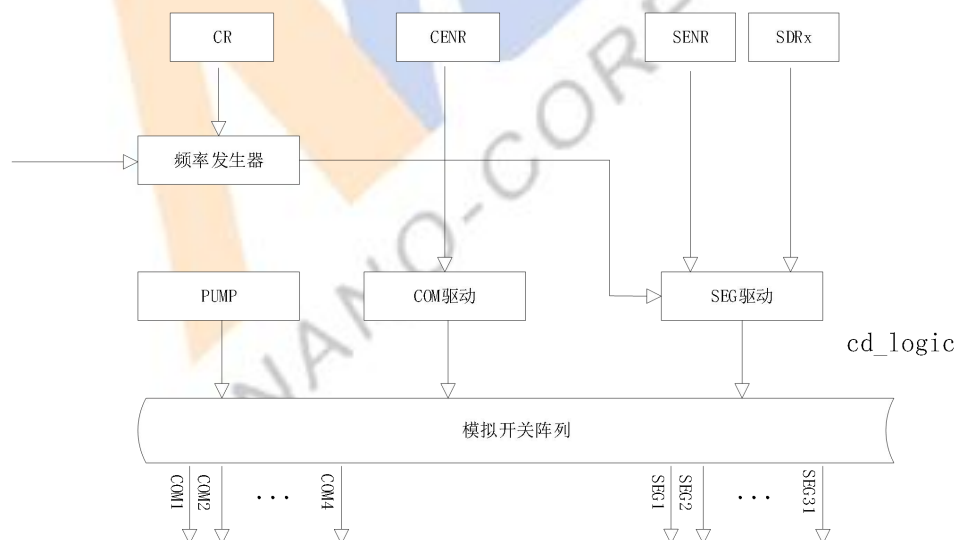


图 25-1 LCD 模块框图

25.3.2. 工作模式

支持两种不同方式的 LCD 驱动：电阻型 LCD 和电容型 LCD，通过控制寄存器 CR[TSEL]位来选择。其中电容型

LCD 驱动又分电容偏压型和内建电压稳定器型两种驱动方式，通过 CR[PPSEL]位来选择。

只有当 LCDON 位置 1 时，LCD 功能才会有效。

LCD 时钟源为低频 32KHz，LCD 帧频固定为 64Hz。

25.3.3. 电阻型 LCD 驱动器

CR[TSEL]位置 0，选择 LCD 驱动类型为电阻型驱动。

CR[MOD]位可选择 LCD 偏置电阻（RLCD）总和为 60k、450k 或 990k。

- 将 CR[MOD]位设置为 1，选择 60k 偏置电阻。
- 将 CR[MOD]位设置为 0，CR[RLCD]为设置为 0，选择 450k 偏置电阻。
- 将 CR[MOD]位设置为 0，CR[RLCD]为设置为 1，选择 990k 偏置电阻。

选择较小的偏置电阻可以得到较好的显示效果，但工作电流相对偏大，不适合低功耗的应用；反之，选择较大的偏置电阻，工作电流相对较小，但显示效果稍差。

因此，MCU 提供了兼顾低功耗和显示效果的显示模式：快速充电模式。设置 CR[MOD]= 1x 可以选择此种显示方式，在显示数据刷新时刻选择 60k 偏置电阻，提供较大的驱动电流，在数据保持期间选择 450k 或 990k 偏置电阻，提供较小的驱动电流。

由 CR[FCCTL]位选择充电时间为 LCD COM 周期的 1/4、1/8、1/16 或 1/32。

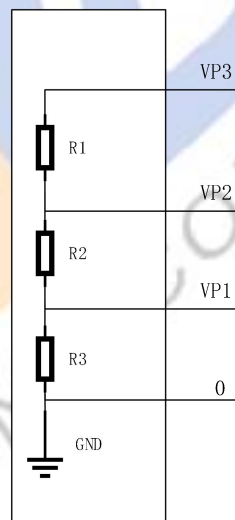


图 25-2 电阻型 LCD、1/3 偏置

25.3.4. 电容型 LCD 驱动器

CR[TSEL]位置 1，选择 LCD 驱动类型为电容型驱动

LCD_PPCK 为电容驱动方式的工作时钟，为 LCD 工作时钟的 8 分频。

电容型 LCD 驱动分为电容偏压型和内建电压稳定器型两种驱动方式，可以通过 CR[PPSEL]位控制。

注意：为更有效的使用电容型 LCD 驱动器，用户必须先设置 PUMPON，LCDON 位之外的所有控制位，然后设置 PUMPON 位，再打开 LCD，即 LCDON 位置 1，点亮 LCD 面板。

25.3.4.1.连接电容的引脚

表 25-1 电容型 LCD 特殊引脚说明

引脚	功能
PC11	CP2
PC7	CP1
PC6	VP1
PC5	VP2
PC4	VP3

CR[TSEL]置 1，选择电容型 LCD 驱动模式。此时 CP1-CP2、VP1-VP3 对应的引脚需要连接电容，建议都选择电容值为 100nF 的电容。如图 25-3 和图 25-4 所示，CP1 和 CP2 间接一个电容，VP1、VP2、VP3 分别接一个电容。

此时，CP1-CP2、VP1-VP3 对应的 IO 口应配置为模拟通道模式，因为这些 IO 口用来连接电容，所以此时不可以作它用。

25.3.4.2.电容偏压型 LCD 驱动

CR[PPSEL]置 0，选择电容偏压型 LCD，驱动电压 VLCD 等于 VDD。

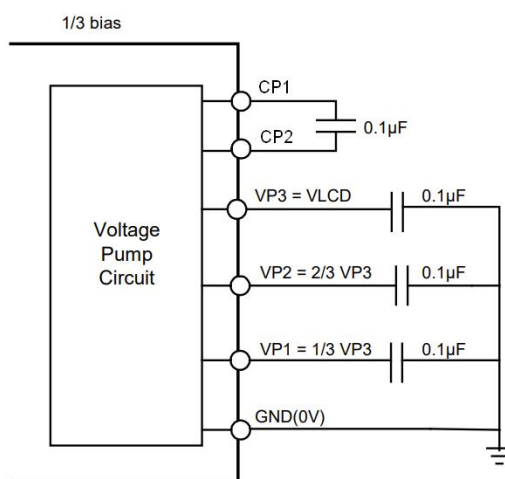


图 25-3 电容偏压型 LCD、1/3 偏置

电容偏压型 LCD 驱动器包含一个控制器，一个占空比发生器及 4/5 个 COM 输出引脚 32/31 个 SEG 输出引脚。SEG1- SEG32 和 COM1-COM5 脚可以当作 I/O 口使用。32 字节的 LCD 显示数据存储在寄存器 SDR0~SDR31 中。

25.3.4.3. 内建电压稳定器型 LCD 驱动器

CR[PPSEL]置 1，选择内建电压稳定型 LCD 驱动器，内建一个电压稳定器，当 VDD 介于 1.8V 及 3.6V 之间时，它可以产生一个稳定的电压。

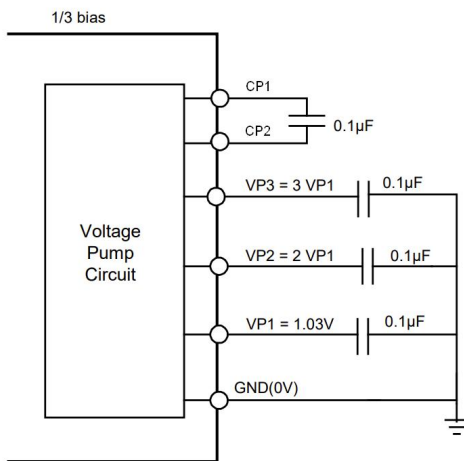


图 25-4 内建电压稳定器型 LCD、1/3 偏置

25.3.5. LED 驱动器

通过 CR[TSEL]=2 使能 LED 驱动器工作模式，可通过 CR[LEDPOL]选择 COM 口输出极性，并通过 CR[DUTY]配置占空比为 1/4 或 1/8，其它配置同 LCD，对应的 SEG 和 COM 需配置为 AF2 引脚复用。

25.3.6. 闪烁

驱动器还实现了一个可编程的闪烁功能，允许一些像素以特定的频率连续地接通。

可以通过 FCR[BLINKF]位配置闪烁频率；通过 FCR[BLINK]位配置闪烁模式，可以选择禁止闪烁，闪烁特定的像素(BLINKSEG 和 BLINKCOM 分别用来指定闪烁的 SEG 和 COM 位置)，或是闪烁某一个 SEG 的像素(BLINKSEG 用来指定闪烁的 SEG)，或是闪烁所有像素。

25.3.7. LCD 波形

COM 信号偏置

各个 COM 信号的相位不同，仅在周期的相应相位才具有最大振幅 VP3，而在其他相位的信号幅值为 VP1、VP2 或者 0（1/3 偏置情况下）。

某个像素点激活，意味着此相位的 COM 和区段线之间的电压差最大。

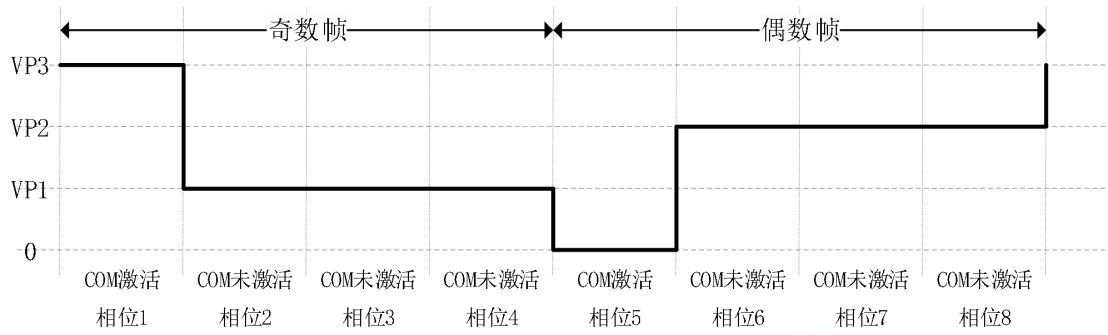
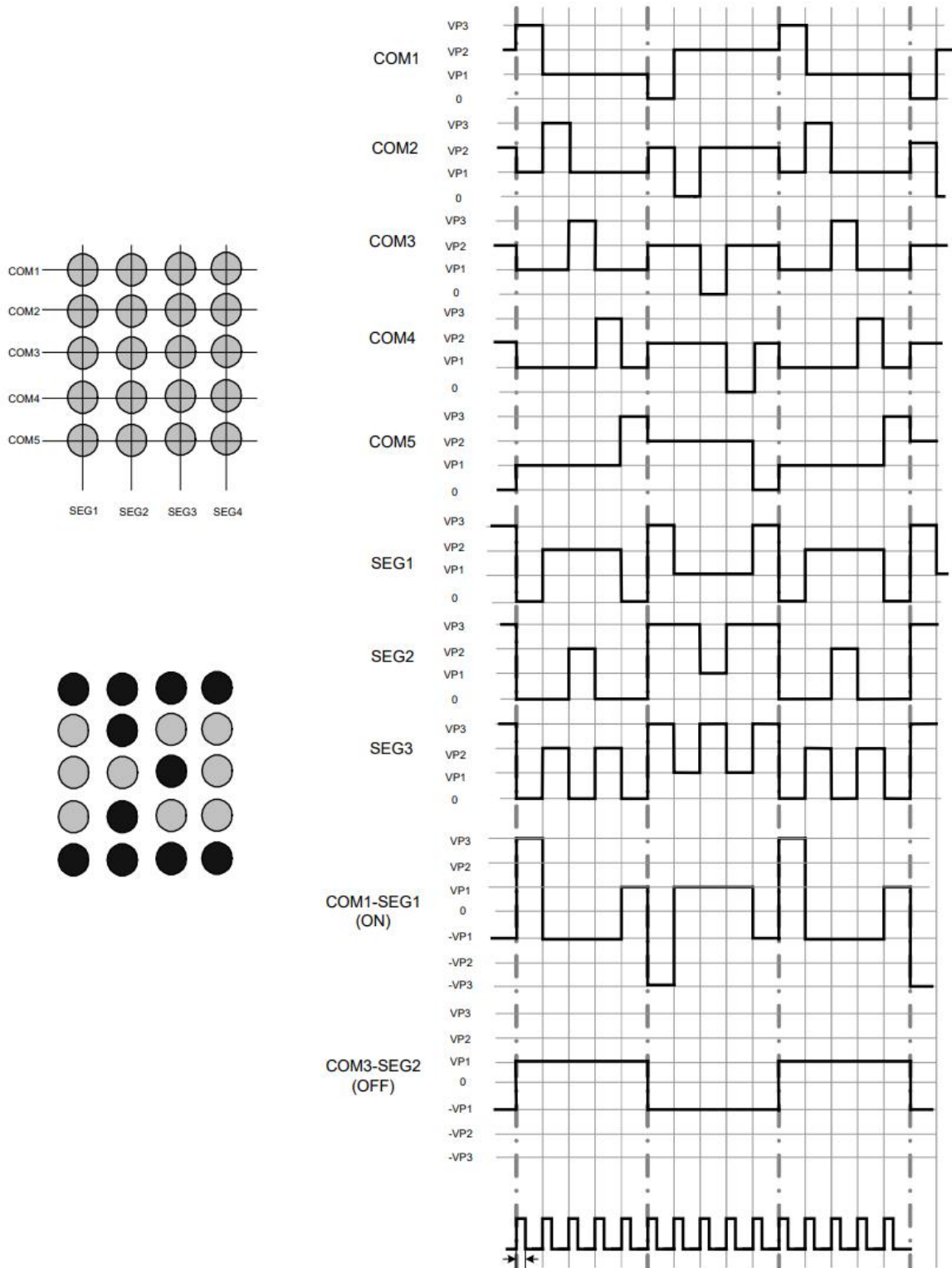


图 25-5 1/4 占空比、1/3 偏置的 COM 口波形示意图



LCD显示Σ (1/5占空比, 1/3偏置)

图 25-6 1/4 占空比、1/3 偏置 显示字符 51.8

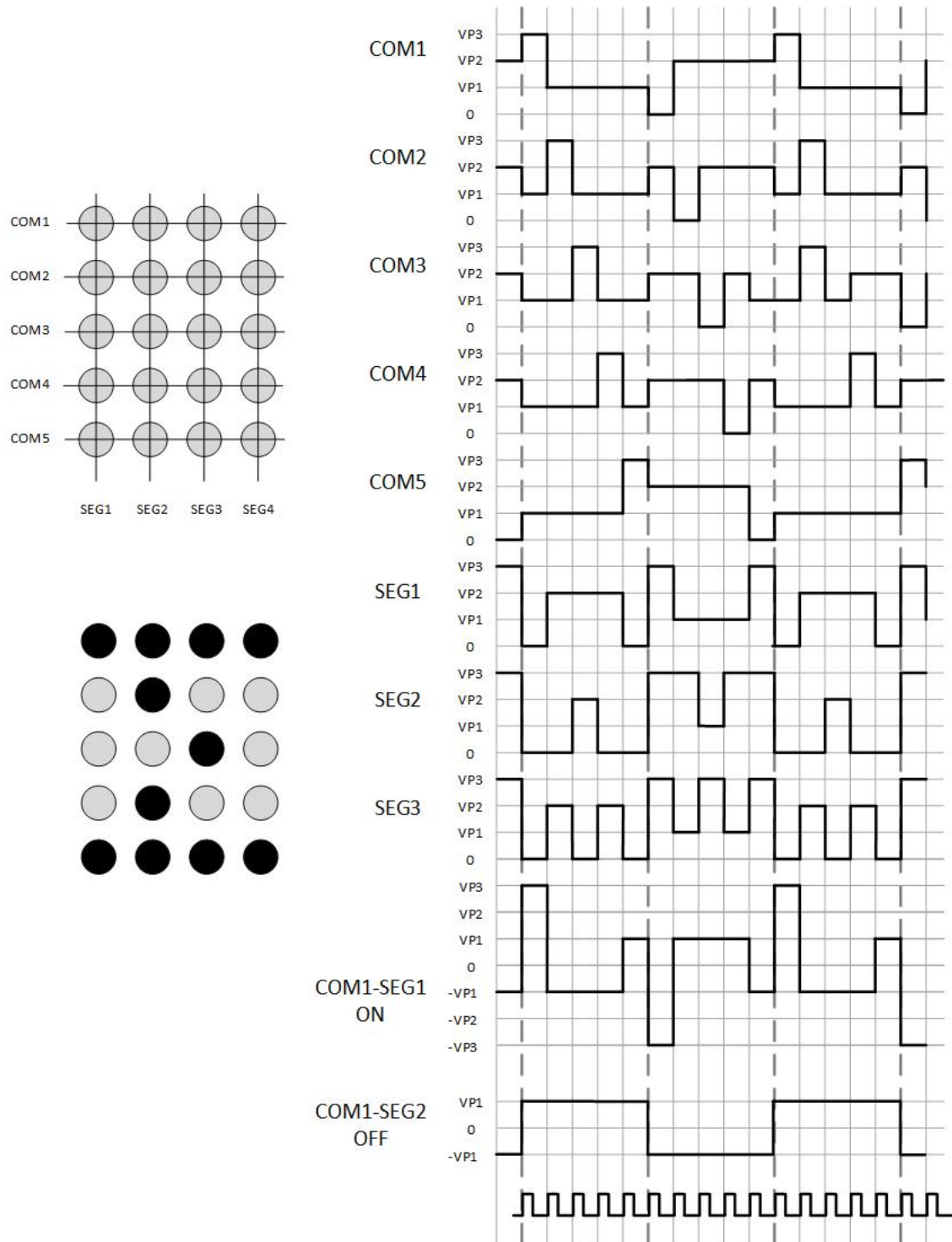


图 25-7 1/5 占空比、1/3 偏置 显示字符Σ

25.3.8. LCD 数据

表 25-2 LCD RAM 配置

地址	bit31-bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	LED_C8	LED_C7	LED_C6	COM5/ LED_C5	COM4/ LED_C4	COM3/ LED_C3	COM2/ LED_C2	COM1/ LED_C1
0x18	-	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
0x1c	-	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
0x20	-	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
0x24	-	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
0x28	-	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
0x2C	-	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
0x30	-	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
0x34	-	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
0x38	-	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
0x3C	-	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
0x40	-	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
0x44	-	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
0x48	-	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
0x4C	-	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
0x50	-	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
0x54	-	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
0x58	-	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
0x5C	-	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
0x60	-	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
0x64	-	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
0x68	-	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
0x6C	-	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
0x70	-	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
0x74	-	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
0x78	-	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
0x7C	-	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
0x80	-	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27
0x84	-	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
0x88	-	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29
0x8C	-	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30
0x90	-	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31
0x94	-	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32

注意：若使用 1/4 占空比，则使用 4 个 COM 口，32 个 SEG 口，此时 SDRx 寄存器只有 bit[0-3] 有效；

若使用 1/5 占空比，则使用 5 个 COM 口，SEG32 口作为 COM5 使用，此时 SDRx 寄存器 bit[0-4] 有效。

25.4. LCD 寄存器

25.4.1. LCD 寄存器映射

表 25-3 LCD 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	LCD 版本寄存器 (IP_VERSION)	r	0xA368_0008
0x04	LCD 控制寄存器 (CR)	rw	0x0000_0000
0x08	LCD 闪烁控制寄存器 (FCR)	rw	0x0000_0000
0x0C	LCD COM 选择控制寄存器 (CENR)	rw	0x0000_0000
0x10	LCD SEGEN 选择控制寄存器 (SENR)	rw	0x0000_0000
0x18-0x94	LCD SEG 数据寄存器, 对应 SEG1- SEG32 (SDR0- SDR31)	rw	0x0000_0000

25.4.2. LCD 版本寄存器(VER)

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

25.4.3. LCD 控制寄存器 (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEDPOL	TSEL	MOD	RLCD	FCCTL	VOL			PPSEL	DUTY	PUMPON	LCDON				
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw				

位域	功能
31-16	保留
15 LEDPOL	LED 驱动时, COM 口极性选择 0: LED COM 口低电平有效 1: LED COM 口高电平有效
14-13 TSEL	驱动类型选择 00: LCD 电阻型驱动 01: LCD 电容型驱动

	1x: LED 驱动
12-11 MOD	驱动模式选择位 00: 传统电阻型模式, 偏置电阻总和为 450k/990k 01: 传统电阻型模式, 偏置电阻总和为 60k 1x: 快速充电模式, 偏置电阻总和自动在 60k 和 450k/990k 之间切换 注意: 当选择电容型 LCD 驱动时, 此 2 位无效
10 RLCD	LCD 偏置电阻选择位 0: LCD 偏置电阻总和为 450k 1: LCD 偏置电阻总和为 990k 注意: 当选择电容型 LCD 驱动时, 此位无效
9-8 FCCTL	充电时间控制位 00: 1/4 LCD com 周期 01: 1/8 LCD com 周期 10: 1/16 LCD com 周期 11: 1/32 LCD com 周期 注意: 当选择电容型 LCD 驱动时, 此 2 位无效
7-4 VOL	LCD 对比度控制位: 电阻分压型 VDDH 分压比例选择信号 0000: 0.57 0001: 0.59 0010: 0.61 0011: 0.63 0100: 0.65 0101: 0.67 0110: 0.69 0111: 0.71 1000: 0.74 1001: 0.77 1010: 0.80 1011: 0.83 1100: 0.87 1101: 0.91 1110: 0.95 1111: 1.00 注意: 当选择电容型 LCD 驱动或 LED 驱动时, 此位域无效
3 PPSEL	电容型 LCD 驱动方式选择位 0: 选择电容偏压型 LCD 驱动方式 1: 选择内建电压稳定器型 LCD 驱动方式
2 DUTY	LCD 占空比选择位 0: 1/4 占空比, 1/3 偏置 1: 1/8 占空比(仅 LED 驱动时有效)
1 PUMPON	PUMP 打开/关闭控制位 0: 禁止 LCD PUMP 1: 允许 LCD PUMP 注意: PUMPON 必须为 1, 电容偏压型和电压稳定器型 LCD 才有效
0 LCDON	LCD 使能控制位 0: 禁止 LCD 驱动器 1: 允许 LCD 驱动器 通过软件将此位置 1 可使能 LCD 控制器/驱动器。通过软件将此位清零可在下一帧开始时关闭 LCD

25.4.4. LCD 闪烁控制寄存器（FCR）

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

位域	功能
31-13	保留
12-7 BLINKSEG	闪烁 SEG 选择
6-5 BLINKCOM	闪烁 COM 选择
3-2 BLINK	闪烁模式选择 00: 禁止闪烁 01: 在 SEG[BLINKSEG+1]、COM[BLINKCOM +1]上启用闪烁（1 个像素） 10: 在 SEG[BLINKSEG+1]、所有 COM 上启用闪烁（最多 5 个像素，取决于编程的占空比） 11: 在所有 SEG 和所有 COM 上启用闪烁（所有像素）
1-0 BLINKF	闪烁频率选择 00: 0.5Hz 01: 1Hz 10: 2Hz 11: 4Hz

25.4.5. LCD COM 选择控制寄存器（CENR）

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

位域	功能
31-5	保留
7-0 COM_EN	选择控制，COM[x] 0: 对应 COM[x+1]未选中 1: 对应 COM[x+1]选中 <i>注意：LCD 模式下，在 1/4 DUTY 时 COM_EN[3:0]有效，在 1/5 DUTY 时 com_en[4:0]有效。 在 LED 模式下，在 1/4 DUTY 时 COM_EN[3:0]有效；在 1/8 DUTY 时 COM_EN[7:0]有效。</i>

25.4.6. LCD SEGEN 选择控制寄存器（SENR）

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

位域	功能
31-0 SEG_EN	SEG 选择控制 SEG_EN[x] 0: 对应 SEG[x+1]未选中 1: 对应 SEG[x+1]选中 <i>注意: LED 模式下, 在 1/8 DUTY 时 SEG_EN[27:0]有效。</i>

25.4.7. LCD SEG 数据寄存器（SDRx）（x=0-31）

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

位域	功能
7-0 SDATAx	对应于 SEG[x], 每个位对应于 LCD 显示器的一个像素 0: 像素未激活 1: 像素激活

26. 调试(DBG)

26.1. 简介

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

26.2. 功能说明

26.2.1. 对低功耗模式的调试支持

在睡眠模式下， AHB 总线时钟没有关闭，可以在睡眠模式下调试。

将 CR[DBG_STOP]位置 1，并进入停止模式，调试程序可以在停止模式下继续调试。

26.2.2. 对定时器、看门狗的调试支持

当内核停止，并且在 DBG 冻结寄存器中的相应位置 1。对于不同外设有不同的动作：

对于定时器，计数器停止工作并进行调试；

对于看门狗，计数器时钟停止并进行调试。

26.3. DBG 寄存器

26.3.1. DBG 寄存器映射

表 26-1 DBG 寄存器映射和复位值

偏移	寄存器名	访问	复位值
0x00	DBG 版本寄存器 (VER)	r	0x0401_0003
0x04	DBG 控制寄存器 (CR)	rw	0x0000_0000
0x08	DBG 冻结寄存器 1 (FZ1)	rw	0x0000_0000
0x0C	DBG 冻结寄存器 2 (FZ2)	rw	0x0000_0000

26.3.2. DBG 版本寄存器(VER)

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

位域	功能
31-0	IP 版本号

26.3.3. DBG 控制寄存器 (CR)

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

位域	功能
31-1	保留
0 DBG_STOP	调试 STOP 模式 0: 关闭 FCLK, HCLK 1: 开启 FCLK, HCLK

26.3.4. DBG 冻结寄存器 1 (FZ1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	DBG_IWDG_STOP	DBG_WWDG_STOP	-	-	-	-	DBG_BUZ_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	-	-	DBG_TIM2_STOP	-
-	-	-	rw	rw	-	-	-	-	rw	rw	rw	-	-	rw	-

位域	功能
31-13	保留
12 DBG_IWDG_STOP	内核停止时 IWDG 计数器的时钟 此位用于开启/关闭核心停止时 IWDG 计数器的时钟 0: 开启 1: 关闭
11 DBG_WWDG_STOP	内核停止时 WWDG 计数器的时钟 此位用于开启/关闭核心停止时 WWDG 计数器的时钟 0: 开启 1: 关闭
10-7	保留
6 DBG_BUZ_STOP	内核停止时 BUZ 计数器的时钟 此位用于开启/关闭核心停止时 BUZ 计数器的时钟 0: 开启 1: 关闭
5 DBG_TIM7_STOP	内核停止时 TIM7 计数器的时钟 此位用于开启/关闭核心停止时 TIM7 计数器的时钟 0: 开启 1: 关闭
4 DBG_TIM6_STOP	内核停止时 TIM6 计数器的时钟 此位用于开启/关闭核心停止时 TIM6 计数器的时钟 0: 开启 1: 关闭
3-2	保留
1 DBG_TIM2_STOP	内核停止时 TIM2 计数器的时钟 此位用于开启/关闭核心停止时 TIM2 计数器的时钟 0: 开启 1: 关闭
0	保留

26.3.5. DBG 冻结寄存器 2 (FZ2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	DBG_TIM16_STOP	DBG_TIM10_STOP
-	-	-	-	-	-	-	-	-	-	-	-	-	-	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位域	功能
31-18	保留
17 DBG_TIM16_ST OP	内核停止时 TIM16 计数器的时钟 此位用于开启/关闭核心停止时 TIM16 计数器的时钟 0: 开启 1: 关闭
16 DBG_TIM10_ST OP	内核停止时 TIM10 计数器的时钟 此位用于开启/关闭核心停止时 TIM10 计数器的时钟 0: 开启 1: 关闭
15-0	保留

27. 电气特性

27.1. 极限参数

表 27-1 极限参数

标识	参数	范围	单位	其它
VDD	电源范围	-0.3~5.5	V	-
T _{STG}	存放温度	-40~125	°C	-
T _{opt}	工作温度	-40~85	°C	-
V _{IIN}	引脚输入电压	-0.3~VDD+0.3	V	-
I _{OLT}	引脚总灌电流	100	mA	最大灌电流
I _{OHT}	引脚总拉电流	-100	mA	最大拉电流

27.2. DC 特性

表 27-2 直流特性参数

(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
V _{DD}	工作电压	内部高频时钟 48MHz	2.0		5.5	V
I _{DD1}	工作电流	系统时钟 48MHz, ADC 关闭		10		mA
I _{DD2}	睡眠电流	系统时钟 48MHz, 模拟关闭 进入 Sleep Mode		3		mA
I _{DD31}	待机电流	所有模拟模块关闭		TBD		uA
I _{DD32}		低振 32K 运行, 唤醒定时器打开, 其他 模块关闭		TBD		uA
V _{IH}	输入高电压	VDD=5V			3.5	V
		VDD=3.3V			2	V
V _{IL}	输入低电压	VDD=5V	1.5			V
		VDD=3.3V	0.8			V
I _{IL}	(输入 0) 漏电流		-1			uA
I _{IH}	(输入 1) 漏电流				1	uA
V _{OH}	输出高电压	VDD=5V, I _{src} =10mA	4.4			V
		VDD=3.3V, I _{src} =4mA	2.4			V
V _{OL}	输出低电压	VDD=5V, I _{sink} =20mA			0.5	V
		VDD=3.3V, I _{sink} =4mA			0.4	V
R _{PUL}	上拉电阻 1 (除 PC10 外)			50		KΩ
	上拉电阻 2 (PC10)			100		KΩ
R _{PD}	下拉电阻			50		KΩ

27.3. BOD/BOR 特性

表 27-3 BOD/BOR 特性参数
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
BOD	BOD 电压	BOD Level=4.3V	4.1	4.3	4.5	V
		BOD Level=3.7V	3.5	3.7	3.9	V
		BOD Level=3.0V	2.85	3.0	3.15	V
		BOD Level=2.8V	2.65	2.8	2.95	V
		BOD Level=2.6V	2.5	2.6	2.7	V
		BOD Level=2.4V	2.3	2.4	2.5	V
		BOD Level=2.2V	2.1	2.2	2.3	V
		BOD Level=2.0V	1.9	2	2.1	V
	BOD 释放迟滞			100		mV
BOR	BOR 电压	BOR Level=2.4V	2.3	2.4	2.5	V
		BOR Level=2.1V	2.0	2.1	2.2	V
		BOR Level=1.9V	1.8	1.9	2.0	V
	BOR 释放迟滞			100		mV

27.4. ADC 特性

表 27-4 ADC 特性参数^[1]
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
V _{DD}	供电电压	VDD=5V	2.7		5.5	V
V _{IREF}	内部参考电压	VDD=5V	2.34	2.4	2.46	V
V _{EREF}	外部参考电压				5	V
V _{IN}	输入电压范围	VDD=5V	0		VREF	V
N _R	分辨率	VDD=5V		12		Bit
DNL	差分非线性误差	VDD=5V		2		LSB
INL	积分非线性误差	VDD=5V		±3		LSB
E _F ^[1]	满刻度误差	VDD=5V		±6		LSB
E _{AD} ^[1]	总绝对误差	VDD=5V		±8		LSB
E _O ^[1]	失调误差	VDD=5V		±4		LSB
F _{ADC_MAX1}	ADC 输入频率	4.5~5.5V		8		MHz
F _{ADC_MAX2}		3.0~5.5V		4		MHz
F _{ADC_MAX3}		2.7~5.5V		1		MHz
T _{ADC}	转换时间	VDD=5V		14		TCK
T _{SAMP1}	采样时间	VDD=5V, F _{ADC} =4MHz	1		16	TCK
T _{SAMP2}		VDD=3V, F _{ADC} =1MHz	1		16	TCK
C _{ADC}	ADC 采样和保持电容	VDD=5V		8		pf

[1]设计保证

27.5. 内部振荡器特性

表 27-5 IHRC 参数特性
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
V _{DD}	工作电压		2.0		5.5	V
F _{req}	振荡频率	出厂校准值, TA=25°C		48		MHz
F _{Ivar}	频率错误	TA=25°C	-1		1%	%
		-40°C~85°C	-3		3	%
DUTY	占空比		45		55	%
T _{HSIO}	起振时间			5		us

27.6. 内部低频振荡器特性

表 27-6 ILRC 参数特性
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
V _{DD}	工作电压		2.0		5.5	V
F _{req}	振荡频率	25°C, 5V		32		KHz
F _{Ivar}	频率错误	TA=25°C	-5		5	%
		-40°C~85°C	-10		10	%
DUTY	占空比		40		60	%
T _{LSIO}	起振时间			500		us

27.7. 外部振荡器特性

表 27-7 CRY 参数特性
(TA=25°C, 2.0V~5.5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
F _{req}	振荡频率	CL=10~15pf, TA=25° C		32.768		KHz
T _{st}	稳定时间			5		ms

[1]外部振荡器起振电路工作温度范围为-40~85°C

27.8. 存储器工作特性

表 27-8 参数特性

标识	参数	测试条件	MIN	TYP	MAX	单位
擦写次数	N _{END}	-40°C~85°C	100K			Cycles
数据保持时间	T _{DR}	-	10			Years

27.9. 外部复位及中断管脚特性

表 27-9 外部复位及中断管脚特性
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	MIN	TYP	MAX	单位
T _{INT}	外部中断脉宽	VDD=5.0V		20		us
T _{RST}	外部复位脉宽	VDD=5.0V		20		us

27.10. ESD 特性

表 27-10 ESD 特性
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	通过值	单位
V _{ESD(HBM)}	静电放电电压 (人体模型)	TA=25°C	±8000	V
V _{ESD(CDM)}	静电放电电压 (充电设备模型)		±2000	V

27.11. SYSTEM ESD 特性

表 27-11 SYSTEM ESD 特性
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	通过值	单位
V _{ESD}	静电放电电压 (静电枪测试)	F _{sys} =24MHz	±2000	V

注：在容易受 ESD 干扰的引脚串接小电阻，可以提高 ESD 的水平。

27.12. EFT 特性

表 27-12 EFT 特性
(若无其他说明, TA=25°C, VDD=5V)

标识	参数	测试条件	通过值	单位
V _{EFT}	瞬变脉冲群电压	F _{sys} =24MHz, BOR 关闭, VDD/GND	±3000	V
		F _{sys} =24MHz, BOR 关闭, IO	±2000	V

27.13. EMI 特性

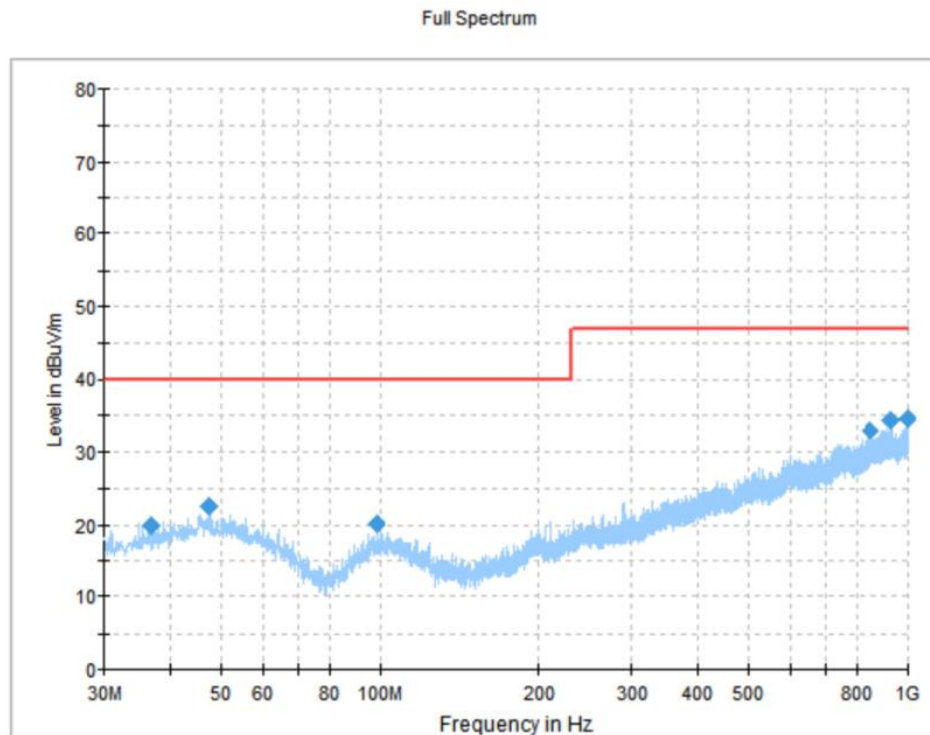


表 27- 13 EMI 测试结果

Frequency (MHz)	QuasiPeak (dBuV/m)	Limit (dBuV/m)	Margin (dB)	Meas. Time (ms)	Bandwidth (kHz)	Height (cm)	Pol	Azimuth (deg)	Corr. (dB)
36.984000	19.84	40.00	20.16	1000.0	120.000	217.0	H	343.0	18.5
47.363000	22.44	40.00	17.56	1000.0	120.000	133.0	H	132.0	20.8
98.385000	20.09	40.00	19.91	1000.0	120.000	114.0	H	32.0	18.7
847.613000	33.08	47.00	13.92	1000.0	120.000	238.0	H	269.0	30.2
926.231500	34.32	47.00	12.68	1000.0	120.000	126.0	H	139.0	31.4
997.720500	34.57	47.00	12.43	1000.0	120.000	134.0	H	100.0	31.4

27.14. Latch up 特性

表 27- 14 Latch up 特性
(若无其他说明，TA=25° C，VDD=5V)

标识	参数	测试条件	通过值	单位
LU	静态栓锁	TA=25°C,VDD=5V	±800	mA

28. 封装信息

28.1. LQFP48

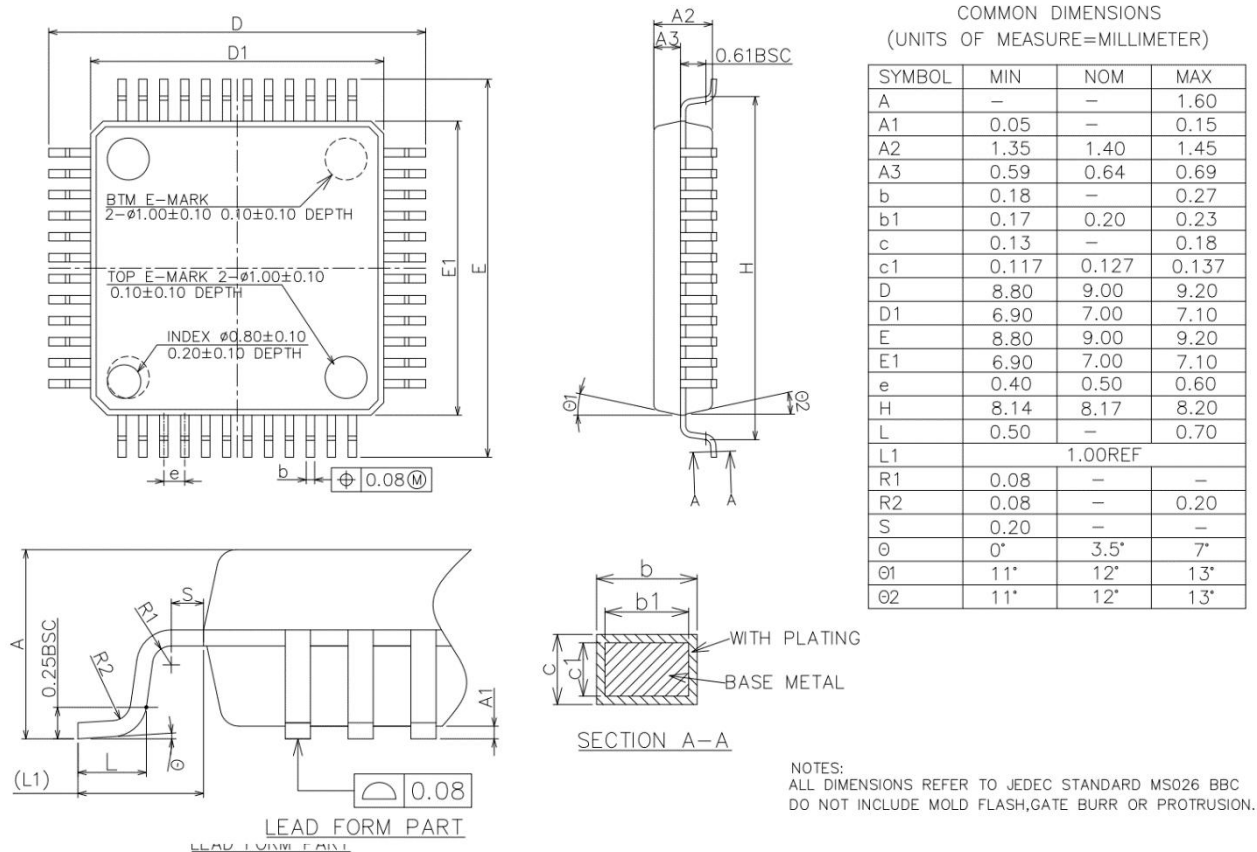


图 28-1 LQFP48-7*7 封装尺寸图

28.2. LQFP32

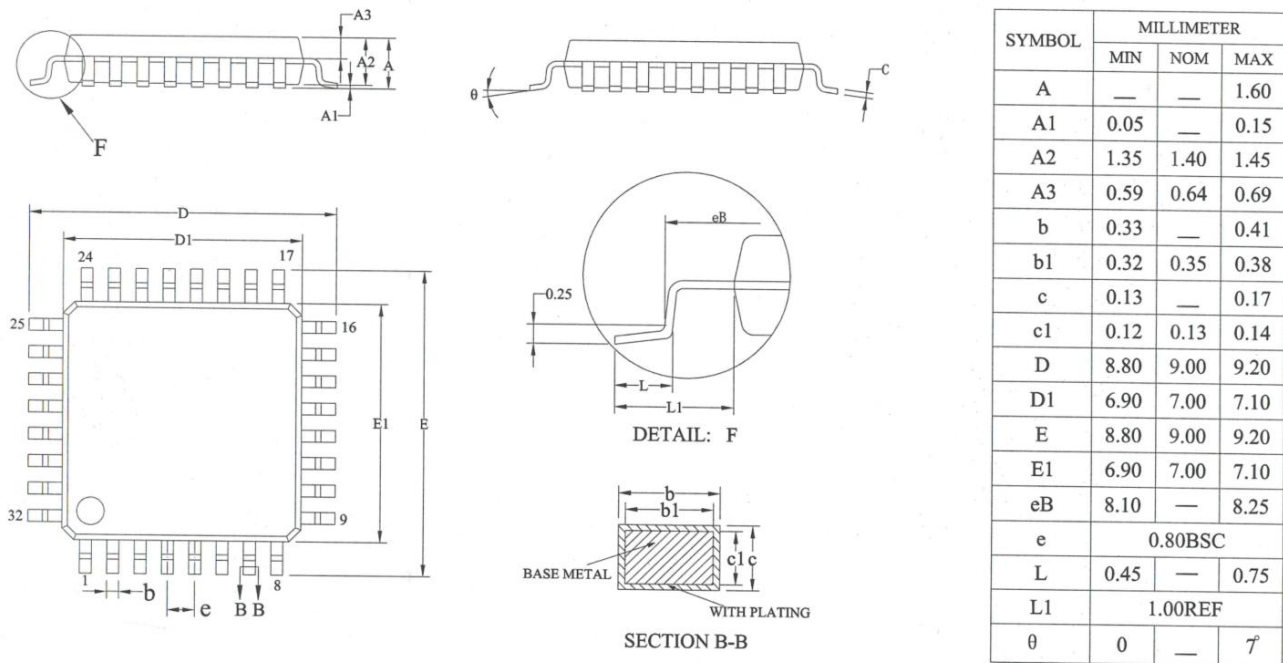


图 28-2 LQFP32-7*7 封装尺寸图

28.3. LQFP44

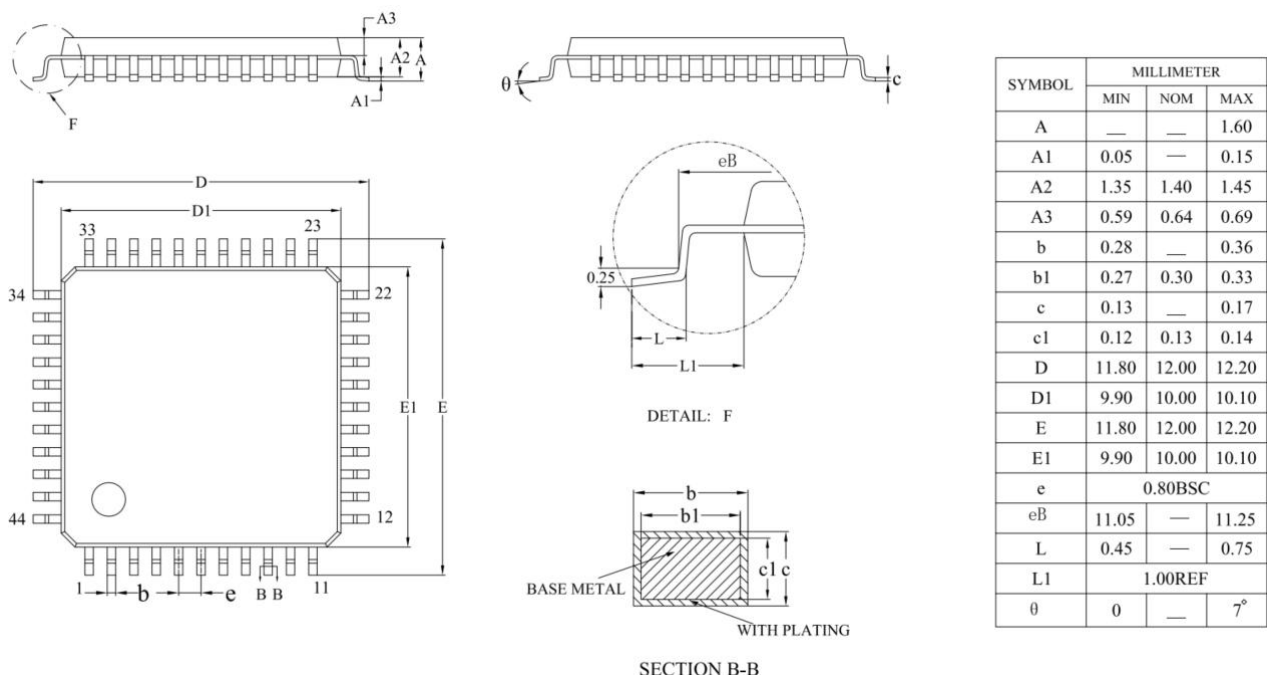


图 28-3 LQFP 44-10*10 封装尺寸图

29. 文档约定

29.1. 寄存器相关缩写词列表

读/写 (rw)	软件可以读写该位。
只读 (r)	软件只能读取该位。
只写 (w)	软件只能写入该位。读取该位时将返回复位值。
读取/写入 0 清零 (rc_w0)	软件可以读取该位，也可以通过写入 0 将该位清零。写入 1 对该位的值无影响。
读取/写入 1 清零 (rc_w1)	软件可以读取该位，也可以通过写入 1 将该位清零。写入 0 对该位的值无影响。
读取/写入清零 (rc_w)	软件可以读取该位，也可以通过写入寄存器将该位清零。写入该位的值并不重要。
读取/读取清零 (rc_r)	软件可以读取该位。读取该位时，将该位自动清零。写入该位对其值无影响。
读取/读取置位 (rs_r)	软件可以读取该位。读取该位时，将该位自动置 1。写入该位对其值无影响。
读取/置位 (rs)	软件可以读取该位，也可将其置 1。写入 0 对该位的值无影响。
读/仅可写入一次 (rwo)	软件仅可写入一次该位，但可随时读取该位。只能通过复位将该位返回到复位值。
切换 (t)	软件可以通过写入 1 来切换该位。写入 0 无影响。
只读，写触发 (rt_w1)	软件可以读取该位。写入 1 时，将触发事件，但不会影响该位的值。
保留 (-)	保留位，必须保持复位值。

29.2. 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- 字： 32 位数据。
- 半字： 16 位数据。
- 字节： 8 位数据。

30. 版本历史

表 30-1 版本历史

版本号	更新时间	更新内容
1.0	2023/11/30	初版发布
1.1	2024/4/9	1. 补充 EFT、SYSTEM ESD、EMI 特性 2. 补充存储器工作特性测试条件
1.2	2024/4/22	删除 GPIOx 端口输出速度寄存器
1.3	2024/6/28	1. 补充 IWDG 的使能禁止描述 2. 补充唤醒定时器中断信号描述
1.4	2025/3/28	1. 修改 I2C 框图

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Nano-core Chip and its subsidiaries (“NCC”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to NCC products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on NCC products before placing orders. NCC products are sold pursuant to NCC’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of NCC products and NCC assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by NCC herein.

Resale of NCC products with provisions different from the information set forth herein shall void any warranty granted by NCC for such product. NCC and the NCC logo are trademarks of NCC. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2022 HangZhou Nano-core Chip – All rights reserved