



CMS32C020 用户手册

基于 ARM® Cortex®-M0+的 32 位微控制器

V0.9.1

请注意以下有关 CMS 知识产权政策

* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司 MCU 或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn

文档使用说明

本手册是 CMS32C020 微控制器产品的技术参考手册，技术参考手册是有关如何使用本系列产品的应用说明资料，包含各个功能模块的结构、功能描述、工作模式以及寄存器配置等详细信息,并对每种功能模块都有专门的章节进行介绍。

技术参考手册是针对这一系列产品所有功能模块的说明，若要了解特定型号产品的特征说明（即功能搭载情况），可参考相应的数据手册。

数据手册信息如下：

CMS32C020: CMS32C020_datasheet_vx.x.x.pdf

通常在芯片选型的初期，首先要看数据手册，以评估该产品是否能够满足设计上的功能需求；在基本选定所需产品后，需要查看技术参考手册，以确定各功能模块的工作模式是否符合要求；在确定选型进入编程设计阶段时，需要详细阅读技术参考手册，以获知各项功能的具体实现方式和寄存器配置。在设计硬件时可参考数据手册以获得电压，电流，驱动能力以及管脚分配等信息。

关于 Cortex-M0+核心、SysTick 定时器和 NVIC 的详细说明，请参照对应 ARM 的文档。

目录

文档使用说明	2
第 1 章 CPU	16
1.1 概述	16
1.2 Cortex-M0+ 内核特性	16
1.3 调试特性	16
1.4 SWD 接口引脚及调试控制	18
1.5 LOCKUP 控制寄存器 (LOCKCTL)	20
1.6 保护寄存器 (PRCR)	20
1.7 ARM 参考文档	20
第 2 章 引脚功能	21
2.1 端口功能	21
2.2 端口复用功能	21
2.3 寄存器映射	23
2.3.1 控制功能寄存器映射	23
2.3.2 输出输入复用功能寄存器映射	24
2.3.3 特殊功能端口 RESINB 控制寄存器映射	26
2.4 寄存器说明	27
2.4.1 端口寄存器 (Px)	28
2.4.2 端口模式寄存器 (PMx)	30
2.4.3 上拉电阻选择寄存器 (PUx)	32
2.4.4 下拉电阻选择寄存器 (PDx)	34
2.4.5 端口输出模式寄存器 (POMx)	36
2.4.6 端口模式控制寄存器 (PMCx)	38
2.4.7 端口置位控制寄存器 (PSETx)	40
2.4.8 端口清零控制寄存器 (PCLR _x)	42
2.4.9 端口回读寄存器 (PREAD _x)	44
2.4.10 端口模式选择寄存器 (PMS)	46
2.4.11 TTL 和施密特输入选择 (PTTL _x)	46
2.4.12 端口输出复用功能配置寄存器 (PS_XX)	48
2.4.13 特殊功能端口 RESINB 说明 (RSTM)	59
第 3 章 系统结构	60
3.1 概述	60
3.2 系统地址划分	61
第 4 章 时钟发生电路	63
4.1 时钟发生电路的功能	63
4.2 时钟发生电路的结构	64
4.3 寄存器映射	65
4.4 寄存器说明	66
4.4.1 系统时钟控制寄存器 (CKC)	66

4.4.2	时钟运行状态控制寄存器 (CSC)	67
4.4.3	时钟运行模式控制寄存器(CMC)	68
4.4.4	外围允许寄存器 (PER0、PER10、PER11、PER12、PER13)	69
4.4.5	15 位间隔定时器运行时钟选择寄存器 (OSMC)	72
4.4.6	高速内部振荡器的频率选择寄存器 (HOCODIV)	72
4.4.7	低速振荡器时钟选择寄存器 (SUBCKSEL)	73
4.5	系统时钟振荡电路	74
4.5.1	XT1 振荡电路	74
4.5.2	高速内部振荡器	74
4.5.3	低速内部振荡器	75
4.6	时钟发生电路的运行	76
4.7	时钟控制	77
4.7.1	高速内部振荡器的设置例子	77
4.7.2	XT1 振荡电路的设置例子	78
4.7.3	CPU 时钟的状态转移图	79
4.7.4	CPU 时钟转移前的条件和转移后的处理	81
4.7.5	CPU 时钟和主系统时钟的切换所需时间	82
4.7.6	时钟振荡停止前的条件	82
第 5 章	15 位间隔定时器	83
5.1	15 位间隔定时器的功能	83
5.2	15 位间隔定时器的结构	83
5.3	寄存器映射	84
5.4	寄存器说明	85
5.4.1	外围允许寄存器 0 (PER0)	85
5.4.2	15 位间隔定时器的控制寄存器 (CON)	86
5.4.3	15 位间隔定时器的计数时钟控制寄存器 (LPTIMCL)	86
5.5	15 位间隔定时器的运行	87
5.5.1	15 位间隔定时器的运行时序	87
5.5.2	从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移	88
第 6 章	高级控制定时器 TIM1	89
6.1	TIM1 介绍	89
6.2	TIM1 主要特性	89
6.3	TIM1 功能说明	91
6.3.1	时基单元	91
6.3.2	计数器模式	93
6.3.3	重复计数器	102
6.3.4	时钟选择	104
6.3.5	捕获/比较通道	107
6.3.6	输入捕获模式	109
6.3.7	PWM 输入模式	110
6.3.8	强制输出模式	110
6.3.9	输出比较模式	111

6.3.10	PWM 模式.....	112
6.3.11	互补输出和死区插入.....	115
6.3.12	使用刹车功能.....	117
6.3.13	发生外部事件时清除 OCxREF 信号.....	119
6.3.14	生成 6 步 PWM.....	120
6.3.15	单脉冲模式.....	121
6.3.16	编码器接口模式.....	123
6.3.17	定时器输入异或功能.....	125
6.3.18	连接霍尔传感器.....	125
6.3.19	TIM1 与外部触发同步.....	127
6.3.20	定时器同步.....	130
6.3.21	调试模式.....	130
6.4	寄存器映射.....	131
6.5	寄存器说明.....	132
6.5.1	TIM1 控制寄存器 1 (TIM1_CR1).....	132
6.5.2	TIM1 控制寄存器 2(TIM1_CR2).....	134
6.5.3	TIM1 从模式控制寄存器(SMCR).....	136
6.5.4	TIM1 DMA/中断使能寄存器 (TIM1_DIER).....	138
6.5.5	TIM1 状态寄存器 (TIM1_SR).....	139
6.5.6	TIM1 事件生成寄存器 (TIM1_EGR).....	141
6.5.7	TIM1 捕获/比较模式寄存器 1 (TIM1_CCMR1).....	142
6.5.8	TIM1 捕获/比较模式寄存器 1 (TIM1_CCMR2).....	145
6.5.9	TIM1 捕获/比较使能寄存器 (TIM1_CCER).....	146
6.5.10	TIM1 计数器 (TIM1_CNT).....	148
6.5.11	TIM1 预分频器 (TIM1_PSC).....	148
6.5.12	TIM1 自动重载寄存器 (TIM1_ARR).....	148
6.5.13	TIM1 重复计数器寄存器 (TIM1_RCR).....	148
6.5.14	TIM1 捕获/比较寄存器 1 (TIM1_CCR1).....	149
6.5.15	TIM1 捕获/比较寄存器 2 (TIM1_CCR2).....	149
6.5.16	TIM1 捕获/比较寄存器 3 (TIM1_CCR3).....	149
6.5.17	TIM1 捕获/比较寄存器 4 (TIM1_CCR4).....	150
6.5.18	TIM1 刹车和死区寄存器 (TIM1_BDTR).....	150
6.5.19	TIM1 DMA 控制寄存器 (TIM1_DCR).....	152
6.5.20	TIM1 全传输 DMA 地址 (TIM1_DMAR).....	152

第 7 章 通用定时器 TIM3.....153

7.1	TIM3 简介.....	153
7.2	TIM3 主要特性.....	153
7.3	TIM3 功能说明.....	155
7.3.1	时基单元.....	155
7.3.2	计数器模式.....	157
7.3.3	时钟选择.....	168
7.3.4	捕获/比较通道.....	170
7.3.5	输入捕获模式.....	172

7.3.6	PWM 输入模式	173
7.3.7	强制输出模式	174
7.3.8	输出比较模式	174
7.3.9	PWM 模式	176
7.3.10	单脉冲模式	179
7.3.11	发生外部事件时清除 OCxREF 信号	180
7.3.12	编码器接口模式	181
7.3.13	定时器输入异或功能	183
7.3.14	定时器与外部触发同步	183
7.3.15	定时器同步	186
7.3.16	调试模式	191
7.4	寄存器映射	192
7.5	寄存器说明	193
7.5.1	TIM3 控制寄存器 1 (TIM3_CR1)	193
7.5.2	TIM3 控制寄存器 2 (TIM3_CR2)	195
7.5.3	TIM3 从模式控制寄存器 (TIM3_SMCR)	196
7.5.4	TIM3 DMA/中断使能寄存器 (TIM3_DIER)	197
7.5.5	TIM3 状态寄存器 (TIM3_SR)	198
7.5.6	TIM3 事件生成寄存器 (TIM3_EGR)	199
7.5.7	TIM3 捕获/比较模式寄存器 1 (TIM3_CCMR1)	200
7.5.8	TIM3 捕获/比较模式寄存器 2 (TIM3_CCMR2)	203
7.5.9	TIM3 计数器 (TIM3_CNT)	205
7.5.10	TIM3 预分频器 (TIM3_PSC)	205
7.5.11	TIM3 捕获/比较使能寄存器 (TIM3_CCER)	206
7.5.12	TIM3 自动重载寄存器 (TIM3_ARR)	207
7.5.13	TIM3 捕获/比较寄存器 1 (TIM3_CCR1)	207
7.5.14	TIM3 捕获/比较寄存器 2 (TIM3_CCR2)	207
7.5.15	TIM3 捕获/比较寄存器 3 (TIM3_CCR3)	207
7.5.16	TIM3 捕获/比较寄存器 4 (TIM3_CCR4)	208
7.5.17	TIM3 DMA 控制寄存器 (TIM3_DCR)	208
7.5.18	TIM3 全传输 DMA 地址 (TIM3_DMAR)	208

第 8 章 通用定时器 TIM14.....209

8.1	TIM14 简介	209
8.2	TIM14 主要特性	209
8.3	TIM14 功能说明	210
8.3.1	时基单元	210
8.3.2	计数器模式	212
8.3.3	时钟选择	215
8.3.4	捕获/比较通道	216
8.3.5	输入捕获模式	218
8.3.6	强制输出模式	219
8.3.7	输出比较模式	220
8.3.8	PWM 模式	221

8.3.9	单脉冲模式.....	222
8.3.10	调试模式.....	223
8.4	寄存器映射	224
8.5	寄存器说明	225
8.5.1	TIM14 控制寄存器 1 (TIM14_CR1)	225
8.5.2	TIM14 中断使能寄存器 (TIM14_DIER)	225
8.5.3	TIM14 状态寄存器 (TIM14_SR).....	226
8.5.4	TIM14 事件生成寄存器 (TIM14_EGR).....	226
8.5.5	TIM14 捕获/比较模式寄存器 1 (TIM14_CCMR1).....	227
8.5.6	TIM14 捕获/比较使能寄存器 (TIM14_CCER).....	229
8.5.7	TIM14 计数器 (TIM14_CNT)	229
8.5.8	TIM14 预分频器 (TIM14_PSC)	230
8.5.9	TIM14 自动重载寄存器 (TIM14_ARR)	230
8.5.10	TIM14 捕获/比较寄存器 1 (TIM14_CCR1)	230
8.5.11	TIM14 选项寄存器(TIM14_OR)	230
第 9 章	通用定时器 (TIM16/17)	231
9.1	TIM16 /TIM17 主要特性.....	231
9.2	TIM16/TIM17 功能描述.....	233
9.2.1	时基单元.....	233
9.2.2	计数器模式.....	235
9.2.3	重复计数器.....	239
9.2.4	时钟源.....	240
9.2.5	捕捉/比较通道.....	240
9.2.6	输入捕获模式.....	242
9.2.7	强置输出模式.....	242
9.2.8	输出比较模式.....	243
9.2.9	PWM（脉冲宽度调节）模式	244
9.2.10	使用刹车功能.....	245
9.2.11	单脉冲模式.....	247
9.3	寄存器映射	248
9.4	寄存器说明	249
9.4.1	TIM16/17 控制寄存器 1 (TIMx_CR1)	249
9.4.2	TIM16/17 控制寄存器 2 (TIM16/17_CR2).....	250
9.4.3	TIM16/17 DMA/中断使能寄存器(TIM16/17_DIER).....	250
9.4.4	TIM16/17 状态寄存器 (TIM16/17_SR)	251
9.4.5	TIM16/17 事件产生寄存器(TIM16/17_EGR)	252
9.4.6	TIM16/17 捕获/比较模式寄存器 1(TIM16/17_CCMR1).....	253
9.4.7	TIM16/17 捕获/比较使能寄存器 (TIM16/17_CCER).....	256
9.4.8	TIM16/17 计数器(TIM16/17_CNT)	257
9.4.9	TIM16/17 预分频器(TIM16/17_PSC)	257
9.4.10	TIM16/17 自动重装载寄存器 (TIM16/17_ARR).....	257
9.4.11	TIM16/17 周期计数寄存器(TIM16/17_RCR)	257

9.4.12	TIM16/17 捕获/比较寄存器 1(TIM16/17_CCR1)	258
9.4.13	TIM16/17 刹车和死区寄存器 (TIM16/17_BDTR).....	258
9.4.14	TIM16/17 DMA 控制寄存器(TIM16/17_DCR).....	260
9.4.15	连续模式的 DMA 地址 (TIM16/17_DMAR).....	261
第 10 章	窗口看门狗定时器 WWDG	262
10.1	窗口看门狗定时器简介	262
10.2	窗口看门狗定时器主要特性	262
10.3	窗口看门狗定时器功能说明	262
10.4	设置看门狗超时	264
10.5	寄存器映射	265
10.6	寄存器说明	265
10.6.1	WWDG 定时器控制寄存器 (WWDG_CR)	265
10.6.2	WWDG 配置寄存器 (WWDG_CFR)	266
10.6.3	WWDG 状态寄存器(WWDG_SR)	266
第 11 章	独立看门狗定时器 IWDG	267
11.1	概述	267
11.2	特性	267
11.3	功能说明	267
11.3.1	硬件看门狗.....	267
11.3.2	寄存器访问保护.....	267
11.3.3	调试模式.....	267
11.4	IWDG 寄存器映射.....	269
11.5	IWDG 寄存器说明.....	269
11.5.1	关键字寄存器(IWDG_KR).....	269
11.5.2	预分频器寄存器(IWDG_PR)	269
11.5.3	重载寄存器(IWDG_RLR)	270
11.5.4	状态寄存器(IWDG_SR)	271
第 12 章	通用串行通信单元.....	272
12.1	通用串行通信单元的功能	273
12.1.1	3 线串行 I/O (SSPI00)	273
12.1.2	带从属片选功能的简易 SPI.....	274
12.1.3	UART0.....	274
12.1.4	简易 I ² C (IIC00)	275
12.2	通用串行通信单元的结构	276
12.2.1	移位寄存器.....	278
12.2.2	串行数据寄存器 mn (SDRmn)	278
12.3	控制通用串行通信单元的寄存器	280
12.3.1	外围允许寄存器 (PER0)	282
12.3.2	串行时钟选择寄存器 m (SPSm)	283
12.3.3	串行模式寄存器 mn (SMRmn)	284
12.3.4	串行通信运行设定寄存器 mn (SCRmn)	285

12.3.5	串行数据寄存器 mn (SDRmn)	287
12.3.6	串行标志清除触发寄存器 mn (SIRmn)	288
12.3.7	串行状态寄存器 mn (SSRmn)	289
12.3.8	串行通道开始寄存器 m (SSm)	291
12.3.9	串行通道停止寄存器 m (STm)	292
12.3.10	串行通道允许状态寄存器 m (SEm)	293
12.3.11	串行输出允许寄存器 m (SOEm)	294
12.3.12	串行输出寄存器 m (SOm)	295
12.3.13	串行输出电平寄存器 m (SOLm)	296
12.3.14	输入切换控制寄存器 (ISC)	297
12.3.15	噪声滤波器允许寄存器 0 (NFEN0)	298
12.3.16	控制串行输入/输出引脚端口功能的寄存器	299
12.4	运行停止模式	300
12.4.1	以单元为单位停止运行的情况	300
12.4.2	按通道停止运行的情况	301
12.5	3 线串行 I/O (SSPI00) 通信的运行	302
12.5.1	主控发送	303
12.5.2	主控接收	312
12.5.3	主控的发送和接收	321
12.5.4	从属发送	330
12.5.5	从属接收	338
12.5.6	从属的发送和接收	344
12.5.7	传送时钟频率的计算	353
12.5.8	在 3 线串行 I/O (SSPI00) 通信过程中发生错误时的处理步骤	355
12.6	从属选择输入功能的时钟同步串行通信的运行	356
12.6.1	从属发送	359
12.6.2	从属接收	369
12.6.3	从属的发送和接收	376
12.6.4	传送时钟频率的计算	386
12.6.5	在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤	387
12.7	UART0 通信的运行	388
12.7.1	UART 发送	389
12.7.2	UART 接收	398
12.7.3	波特率的计算	405
12.7.4	在 UART0 通信过程中发生错误时的处理步骤	409
12.8	简易 I ² C (IIC00) 通信的运行	410
12.8.1	地址段发送	411
12.8.2	数据发送	416
12.8.3	数据接收	419
12.8.4	停止条件的产生	423
12.8.5	传送速率的计算	424

第 13 章 SPI 通信单元.....428

13.1 简介	428
13.2 SPI	428
13.2.1 SPI 特性	428
13.3 SPI 功能说明	429
13.3.1 一般说明	429
13.3.2 把 SPI 配置成从器件	433
13.3.3 把 SPI 配置成主器件	435
13.3.4 配置 SPI 进行半双工通信	437
13.3.5 数据发送和接收过程	438
13.3.6 CRC 计算	446
13.3.7 状态标志	448
13.3.8 关闭 SPI	449
13.3.9 使用 DMA（直接存储器寻址）进行 SPI 通信	450
13.3.10 错误标志	453
13.3.11 SPI 中断	455
13.4 SPI 寄存器映射	456
13.5 SPI 寄存器说明	456
13.5.1 SPI 控制寄存器 1 (SPI_CR1)	456
13.5.2 SPI 控制寄存器 2 (SPI_CR2)	458
13.5.3 SPI 状态寄存器 (SPI_SR)	459
13.5.4 SPI 数据寄存器 (SPI_DR)	460
13.5.5 SPI CRC 多项式寄存器 (SPI_CRCPR)	461
13.5.6 SPI RX CRC 寄存器 (SPI_RXCRCR)	461
13.5.7 SPI TX CRC 寄存器 (SPI_TXCRCR)	461
第 14 章 I2C 接口	462
14.1 I2C 简介	462
14.2 I2C 主要特性	462
14.3 I2C 功能说明	464
14.3.1 模式选择	464
14.3.2 I2C 从模式	466
14.3.3 I2C 主模式	468
14.3.4 错误条件	472
14.3.5 可编程噪声滤波器	473
14.3.6 SDA/SCL 线控制	473
14.3.7 SMBus	474
14.3.8 DMA 请求	476
14.3.9 数据包错误校验	477
14.4 I2C 中断	478
14.5 I2C 调试模式	479
14.6 寄存器映射	479
14.7 寄存器说明	480
14.7.1 I2C 控制寄存器 1 (I2C_CR1)	480

14.7.2	I2C 控制寄存器 2 (I2C_CR2).....	482
14.7.3	I2C 自有地址寄存器 1 (I2C_OAR1).....	483
14.7.4	I2C 自有地址寄存器 2 (I2C_OAR2).....	483
14.7.5	I2C 数据寄存器 (I2C_DR).....	483
14.7.6	I2C 状态寄存器 1 (I2C_SR1)	484
14.7.7	I2C 状态寄存器 2 (I2C_SR2)	487
14.7.8	I2C 时钟控制寄存器 (I2C_CCR).....	488
14.7.9	I2C TRISE 寄存器 (I2C_TRISE).....	488
14.7.10	I2C FLTR 寄存器 (I2C_FLTR)	489
第 15 章 串行同步异步通信单元 USART		490
15.1	USART 介绍.....	490
15.2	USART 主要特性	490
15.3	USART 功能描述	492
15.3.1	USART 字符说明.....	494
15.3.2	发送器.....	495
15.3.3	接收器.....	498
15.3.4	小数波特率的产生.....	503
15.3.5	USART 接收器对时钟偏差的容差	505
15.3.6	多处理器通信.....	506
15.3.7	奇偶校验控制.....	508
15.3.8	LIN（局域互连网络）模式	509
15.3.9	USART 同步模式.....	512
15.3.10	单线半双工通信.....	515
15.3.11	IrDA SIR ENDEC 模块.....	515
15.3.12	使用 DMA 进行连续通信.....	518
15.3.13	硬件流控制.....	520
15.4	USART 中断	522
15.5	USART 模式配置	523
15.6	寄存器映射	524
15.6.1	状态寄存器 (USART_SR)	524
15.6.2	数据寄存器 (USART_DR).....	526
15.6.3	波特率寄存器 (USART_BRR)	526
15.6.4	控制寄存器 1 (USART_CR1).....	527
15.6.5	控制寄存器 2 (USART_CR2).....	529
15.6.6	控制寄存器 3 (USART_CR3).....	530
15.6.7	保护时间和预分频器寄存器 (USART_GTPR)	532
第 16 章 增强型 DMA		533
16.1	DMA 的功能	533
16.2	DMA 的结构	535
16.3	寄存器映射	536
16.4	控制 DMA 的寄存器	537
16.4.1	DMA 控制数据区和 DMA 向量表区的分配.....	537

16.4.2	控制数据的分配.....	538
16.4.3	向量表.....	540
16.4.4	外围允许寄存器 10 (PER10)	541
16.4.5	DMA 通道触发源选择寄存器 j (DMA_CHj_SEL) (j=0~1,3)	542
16.4.6	DMA 控制寄存器 j (DMACRj) (j=0~1,3)	543
16.4.7	DMA 块大小寄存器 j (DMBLSj) (j=0~1,3)	544
16.4.8	DMA 传送次数寄存器 j (DMACTj) (j=0~1,3)	545
16.4.9	DMA 传送次数重加载寄存器 j (DMRLDj) (j=0~1,3)	546
16.4.10	DMA 源地址寄存器 j (DMSARj) (j=0~1,3)	547
16.4.11	DMA 目标地址寄存器 j (DMDARj) (j=0~1,3)	547
16.4.12	DMA 启动允许寄存器 i (DMAEN0)	547
16.4.13	DMA 基址寄存器 (DMABAR)	548
16.4.14	DMAEN0 置位寄存器 (DMASET0)	548
16.4.15	DMAEN0 复位寄存器 (DMACLR0)	548
16.4.16	DMA 的运行	549
16.4.17	启动源.....	549
16.4.18	正常模式.....	550
16.4.19	重复模式.....	553
16.4.20	链传送.....	556
16.5	使用 DMA 时的注意事项	558
16.5.1	DMA 控制数据和向量表的设置.....	558
16.5.2	DMA 控制数据区和 DMA 向量表区的分配.....	558
16.5.3	DMA 的执行时钟数.....	559
16.5.4	DMA 的响应时间.....	560
16.5.5	DMA 的启动源	560
16.5.6	待机模式中的运行.....	560
第 17 章	A/D 转换器	561
17.1	A/D 转换器的功能.....	561
17.2	寄存器映射	564
17.3	寄存器说明	565
17.3.1	A/D 转换器的模式寄存器 0 (ADM0)	565
17.3.2	A/D 转换器的模式寄存器 1 (ADM1)	569
17.3.3	A/D 转换器的模式寄存器 2 (ADM2)	570
17.3.4	A/D 转换器的触发模式寄存器 (ADTRG)	571
17.3.5	模拟输入通道指定寄存器 (ADS)	572
17.3.6	A/D 内部通道选择寄存器 (ADICHS)	574
17.3.7	12 位 A/D 转换通道标识符寄存器 (ADSCHS)	575
17.3.8	12 位 A/D 转换结果寄存器 (ADCR)	575
17.3.9	8 位 A/D 转换结果寄存器 (ADCRH)	576
17.3.10	A/D 采样时间延长寄存器 (ADSMPWAIT)	576
17.3.11	转换结果比较上限值设置寄存器 (ADUL)	576
17.3.12	转换结果比较下限值设置寄存器 (ADLL)	577
17.4	输入电压和转换结果	578

17.5 A/D 转换器的运行模式.....	579
17.5.1 软件触发模式（选择模式、连续转换模式）	579
17.5.2 软件触发模式（选择模式、单次转换模式）	580
17.5.3 软件触发模式（扫描模式、连续转换模式）	581
17.5.4 软件触发模式（扫描模式、单次转换模式）	582
17.5.5 硬件触发无等待模式（选择模式、连续转换模式）	583
17.5.6 硬件触发无等待模式（选择模式、单次转换模式）	584
17.5.7 硬件触发无等待模式（扫描模式、连续转换模式）	585
17.5.8 硬件触发无等待模式（扫描模式、单次转换模式）	586
17.5.9 硬件触发等待模式（选择模式、连续转换模式）	587
17.5.10 硬件触发等待模式（选择模式、单次转换模式）	588
17.5.11 硬件触发等待模式（扫描模式、连续转换模式）	589
17.5.12 硬件触发等待模式（扫描模式、单次转换模式）	590
17.6 A/D 转换器的设置流程图.....	591
17.6.1 软件触发模式的设置.....	591
17.6.2 硬件触发无等待模式的设置.....	592
17.6.3 硬件触发等待模式的设置.....	593
第 18 章 模拟比较器（ACMP0/1）	594
18.1 概述	594
18.2 特性	594
18.3 结构框图	595
18.4 寄存器映射	596
18.5 寄存器说明	597
18.5.1 模拟比较器 0 控制寄存器 0(C0CON0)	597
18.5.2 模拟比较器 0 控制寄存器 1(C0CON1)	597
18.5.3 模拟比较器 1 控制寄存器 0(C1CON0)	598
18.5.4 模拟比较器 1 控制寄存器 1(C1CON1)	599
18.5.5 模拟比较器事件控制寄存器(CEVCON).....	600
18.5.6 模拟比较器参考电压控制寄存器(CVRCON).....	601
18.5.7 模拟比较器写使能控制寄存器(ACMPLOCK).....	601
第 19 章 通用 CRC	602
19.1 CRC 概述	602
19.2 CRC 特性	602
19.3 功能说明	602
19.4 寄存器映射	603
19.5 寄存器说明	603
19.5.1 CRC 数据寄存器(CRC_DR).....	603
19.5.2 CRC 独立数据寄存器(CRC_IDR)	603
19.5.3 CRC 控制寄存器(CR).....	603
第 20 章 待机功能.....	604
20.1 待机功能	604

20.2 睡眠模式	605
20.2.1 睡眠模式的设定	605
20.2.2 睡眠模式的解除	606
20.3 深度睡眠模式	607
20.3.1 深度睡眠模式的设定	607
20.3.2 深度睡眠模式的解除	608
第 21 章 嵌套向量中断控制器(NVIC)	609
21.1 特性	609
21.2 异常模式和系统中断映射	609
21.3 向量表	610
21.4 外部中断结构	611
21.5 寄存器映射	612
21.6 寄存器说明	613
21.6.1 外部中断上升沿允许寄存器 (EGP0)	613
21.6.2 外部中断上升沿允许寄存器 (EGP1)	614
21.6.3 外部中断下降沿允许寄存器 (EGN0)	615
21.6.4 外部中断下降沿允许寄存器 (EGN1)	616
21.6.5 外部中断屏蔽寄存器 0(E_IMR0)	618
21.6.6 外部中断屏蔽寄存器 1(E_IMR1)	619
21.6.7 外部中断挂起寄存器 0(E_PR0)	620
21.6.8 外部中断挂起寄存器 1(E_PR1)	621
21.6.9 外部中断滤波控制寄存器(FILCON)	622
第 22 章 复位功能	623
22.1 复位时序	624
22.2 确认复位源的寄存器	626
22.2.1 寄存器映射	626
22.2.2 复位控制标志寄存器(RESF)	626
第 23 章 上电复位电路	628
23.1 上电复位电路的功能	628
23.2 上电复位电路的结构	629
23.3 上电复位电路的运行	630
第 24 章 电压检测电路	633
24.1 电压检测电路的功能	633
24.2 电压检测电路的结构	634
24.3 寄存器映射	634
24.4 控制电压检测电路的寄存器	635
24.4.1 电压检测寄存器 (LVIM)	635
24.4.2 电压检测电平寄存器 (LVIS)	636
24.5 电压检测电路的运行	637
24.5.1 用作复位模式时的设定	637
24.6 用作中断模式时的设定	638

24.7 用作中断&复位模式时的设定	640
24.8 电压检测电路的注意事项	645
第 25 章 安全功能	647
25.1 安全功能的概要	647
25.2 寄存器映射	648
25.3 安全功能的运行	649
25.3.1 闪存 CRC 运算功能（高速 CRC）	649
25.3.2 SFR 保护功能	651
25.3.3 A/D 测试功能	652
25.3.4 输入/输出引脚的数字输出信号电平检测功能	653
25.3.5 产品唯一身份标识寄存器	654
第 26 章 选项字节	655
26.1 选项字节的功能	655
26.1.1 用户选项字节（000C0H~000C2H）	655
26.2 闪存数据保护选项字节（000C3H, 500004H）	656
26.3 寄存器映射	656
26.4 用户选项字节	657
26.4.1 用户选项字节（000C0H）	657
26.4.2 用户选项字节（000C1H）	657
26.4.3 用户选项字节（000C2H）	660
26.4.4 闪存数据保护选项字节（000C3H）	661
26.4.5 闪存数据保护选项字节（500004H）	661
第 27 章 FLASH 控制	662
27.1 FLASH 控制功能描述	662
27.2 FLASH 存储器结构	662
27.3 寄存器映射	663
27.4 寄存器说明	664
27.4.1 FLASH 操作控制寄存器(FLCMD)	664
27.4.2 FLASH 密钥寄存器(FLKEY)	664
27.4.3 FLASH 数据寄存器 0(FLDATA0)	664
27.4.4 FLASH 地址寄存器(FLADDR)	664
27.4.5 FLASH 控制寄存器(FLCR)	665
27.4.6 FLASH 状态寄存器(FLSR)	666
27.4.7 FLASH 区域控制寄存器（FLREGION）	666
27.4.8 FLASH 保护寄存器（FLPRT）	667
27.5 FLASH 操作方法	668
27.5.1 编程	668
27.5.2 扇擦除	669
27.5.3 全片擦除（chip erase）	670
27.6 FLASH 操作的注意事项	671
第 28 章 附录 修订记录	672

第1章 CPU

1.1 概述

本章节简单介绍本制品搭载的 ARM Cortex-M0+内核的特性及调试特性，详情请参考 ARM 相关文档。

1.2 Cortex-M0+内核特性

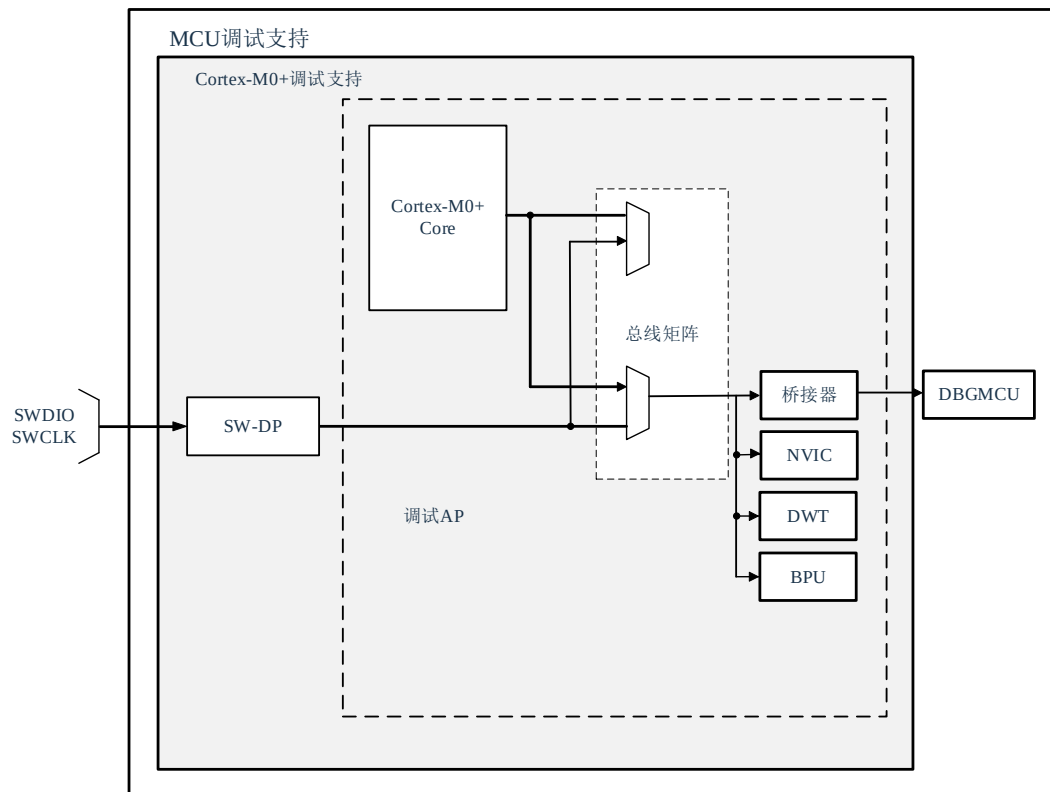
ARM Cortex-M0+处理器是 32 位 RISC 内核，采用 2 级流水线，仅支持特权模式

- 1 周期硬件乘法器
- 嵌套向量中断控制器 (NVIC)
 - 1 个不可屏蔽中断(NMI)
 - 支持 32 个可屏蔽中断请求(IRQ)
 - 4 个中断优先级
- 系统定时器 SysTick 是一个 24 位倒计时定时器，可选择 F_{CLK} 或 F_{IL} 计数时钟
- 向量表偏移寄存器(VTOR)
 - 软件可以写 VTOR 将向量表起始地址重新定位到不同的位置
 - 该寄存器的默认值为 0x0000_0000，低 8 位写忽略，读为零，也就是偏移量 256 字节对齐

1.3 调试特性

- 2 线 SWD 调试接口
- 支持暂停、恢复和单步执行程序
- 访问处理器的内核寄存器和特殊功能寄存器
- 4 个硬件断点(BPU)
- 无限个软件断点(BKPT 指令)
- 2 个数据观察点(DWT)
- 内核执行的时候访问存储器

图 1-1 Cortex-M0+的调试框图



注：SWD 在 Deep Sleep 模式下不能工作，请在 active 和 sleep 模式下进行调试操作。

1.4 SWD 接口引脚及调试控制

本产品的 2 个 GPIO 可用作 SWD 接口引脚，这些引脚在所有的封装里都存在。

表 1-1 SWD 调试端口引脚

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	PA14
SWDIO	串行数据输入 / 输出	PA13

不使用 SWD 功能时，可以通过设置 debug 停止控制寄存器（DBGSTOPCR）来禁用 SWD。

DBGSTOPCR 地址为 0x4001_B004

位	符号	描述	复位值
31:25	-	保留	0
24	SWDIS	SWD 调试接口禁用 0: SWD 调试接口使能。在连接调试器的状态下，PA13/PA14 不能用作 GPIO（因为此时该 IOBUF 的 ENO 和 DOUT 由调试器控制） 1: SWD 调试接口禁用。PA13/PA14 可用作 GPIO	0
23:14	-	保留	0
13	IICDB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），IIC 动作/停止 0: IIC 动作 1: IIC 停止	0
12	TIM17DB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），TIM17 动作/停止 0: TIM17 动作 1: TIM17 停止	0
11	TIM16DB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），TIM16 动作/停止 0: TIM16 动作 1: TIM16 停止	0
10	TIM1DB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），TIM1 动作/停止 0: TIM1 动作 1: TIM1 停止	0
9:8	-	保留	0
7	TIM14DB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），TIM14 动作/停止 0: TIM14 动作 1: TIM14 停止	0
6	TIM3DB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），TIM3 动作/停止 0: TIM3 动作 1: TIM3 停止	0
5	WWDGDB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），WWDG 动作/停止 0: WWDG 动作 1: WWDG 停止	0
4	IWDGDB	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），IWDG 动作/停止 0: IWDG 动作 1: IWDG 停止	0

3:2	-	保留	0
1	SAUDB	在调试器连接的状态下，并且 CPU 处于调试状态时 (HALTED=1)，SAU 动作/停止 0: SAU 动作 1: SAU 停止	0
0	LPTIMDB	在调试器连接的状态下，并且 CPU 处于调试状态时 (HALTED=1)，LPTIM 动作/停止 0: LPTIM 动作 1: LPTIM 停止	0

1.5 LOCKUP 控制寄存器（LOCKCTL）

LOCKCTL 寄存器是 Cortex-M0+ LockUp 功能是否使能 TIM1/TIM16/TIM17 的刹车输入及是否引起独立看门狗定时器运行的配置寄存器，PRCR 是其写保护寄存器。

通过 8 位存储器操作指令设置 LOCKCTL 寄存器。

在产生复位信号后，LOCKCTL 寄存器的值变为“01H”。

LOCKCTL 地址为 0x4002_0405

位	符号	描述	复位值
7:2	-	保留	-
1	LOCKUP_BRKEN_TIM	LOCKUP 功能作为 TIM1/TIM16/TIM17 的刹车输入使能控制 0: 禁止 1: 使能	0
0	LOCKUP_RST	LOCKUP 功能配置 0: LOCKUP 不导致 IWDG 复位 1: LOCKUP 导致 IWDG 复位	1

1.6 保护寄存器（PRCR）

LOCKCTL 寄存器是 Cortex-M0+ LockUp 功能是否使能 TIM1/TIM16/TIM17 的刹车输入及是否引起独立看门狗定时器运行的配置寄存器，PRCR 是其写保护寄存器。

通过 8 位存储器操作指令设置 PRCR 寄存器。

在产生复位信号后 PRCR 寄存器的值变为“00H”。

PRCR 地址为 0x4002_0406

位	符号	描述	复位值
7:1	PRTKEY	PRCR 的写保护 78H: PRCR 可写 其他: PRCR 不可写	0x0
0	PRCR	LOCKUP 控制寄存器写保护 0: LOCKCTL 寄存器不可写 1: LOCKCTL 寄存器可写	0

1.7 ARM 参考文档

Cortex®-M0+ 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。相关文档请参考：

- Cortex®-M0+技术参考手册(TRM)
- ARM®调试接口V5
- ARM® CoreSight设计套件版本r1p1 技术参考手册
- ARM® CoreSight™ MTB-M0+ Technical Reference Manual

第2章 引脚功能

2.1 端口功能

具体端口功能参见各产品系列的数据手册。

2.2 端口复用功能

具体端口复用功能参见各产品系列的数据手册。详细端口复用功能参见下表 2-1。

表 2-1 端口复用功能数字映射表

	USART	SPI	SAU	I2C	TIM1	TIM3	TIM14	TIM16	TIM17
PA0	CTS1	MISO	SCK00/SCL00		CH1N/CH3				
PA1	RTS1	MOSI/SCK			CH2N/CH4				
PA2	TXD1	MOSI/SCK		SDA		CH1			
PA3	RXD1	MOSI	SDI00/SDA00/RXD0	SCL	CH1				
PA4	CK1	NSS	SDO00/TXD0			CH3	CH1		
PA6	CK1	MISO			BKIN	CH1		CH1	
PA7	TXD1	MISO/MOSI		SDA	CH1N	CH2	CH1		CH1
PA8	RXD1/CK1	MOSI	SS00	SCL	CH1				
PA9	RXD1/TXD1	SCK	SDI00/SDA00/RXD0	SDA/SCL	BKIN/CH2				
PA10	TXD1/RXD1	NSS	SDO00/TXD0	SCL/SDA	CH3				BKIN
PA12	RTS1	MOSI		SDA	ETR				
PA13	RXD1	MISO			CH2				
PA14	TXD1								
PB0		NSS			CH2N	CH3			
PB1					CH3N	CH4	CH1		
PB2	RXD1		SCK00/SCL00						
PB3	RTS1	SCK			CH2				
PB4	CTS1	MISO				CH1			BKIN
PB5	CK1	MOSI		SMBA		CH2		BKIN	
PF0	RXD1		SCK00/SCL00	SDA			CH1		
PF1	TXD1	NSS	SDI00/SDA00/RXD0	SCL			CH1		
PF2			SDO00/TXD0						

注 1: 本产品 IO 复用功能, 需要用户单独配置 PMC, PM 等寄存器。

注 2: 当选择到 IIC 功能时, 开漏功能自动开启。

注 3: 针对复用功能, 即可做输入又可做输出功能, 选择好 PS_XX 后, 输入通道自动开启。

表 2-2: 模拟功能和特殊功能管脚

	ADC	ACMP	OSC	SWD	NRST
PA0	ANI0	C0N0/C0OUT0			
PA1	ANI1	C0P0			
PA2	ANI2	C1N0/C1OUT0			
PA3	ANI3	C1P1			
PA4	ANI4				
PA6	ANI6	C0OUT1			
PA7	ANI7	C1OUT1			
PA8	ANI24				
PA9	ANI25		OSC32_OUT		
PA10	ANI26		OSC32_IN		
PA12	ANI28	C1OUT2			
PA13	ANI29			SWDIO	
PA14	ANI30			SWCLK	
PB0	ANI8	C0OUT2			
PB1	ANI9	C0N1			
PB2	ANI10	C0P1			
PB3	ANI11	C1N1			
PB4	ANI12	C1P2			
PB5	ANI13	C0OUT4			
PF0	ANI16				
PF1	ANI17				
PF2	ANI18				NRST

2.3 寄存器映射

2.3.1 控制功能寄存器映射

(端口控制寄存器的基地址=0x4004_0000)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PA	0x000	R/W	以位为单位, 设定端口输出锁存器值的寄存器; 输入模式中读此寄存器可以得到引脚电平, 在输出模式中读此寄存器可得到端口的输出锁存器的值	0x00
PB	0x002	R/W		0x00
PF	0x00A	R/W		0x00
PMA	0x020	R/W	在端口作为数字通道使用时, 以位为单位, 设定端口输入或输出的寄存器	0xFF
PMB	0x022	R/W		0xFF
PMF	0x02A	R/W		0xFF
PUA	0x030	R/W	端口内部上拉电阻选择寄存器	0x6000
PUB	0x032	R/W		0x00
PUF	0x03A	R/W		0x04
PDA	0x040	R/W	端口内部下拉电阻选择寄存器	0x00
PDB	0x042	R/W		0x00
PDF	0x04A	R/W		0x00
POMA	0x050	R/W	开漏模式寄存器, 只有在端口配置为输出模式时, N 沟道漏极开路才会开启	0x00
POMB	0x052	R/W		0x00
POMF	0x05A	R/W		0x00
PMCA	0x060	R/W	端口模式寄存器, 以位为单位设定端口作为数字或模拟通道使用	0x9FFF
PMCB	0x062	R/W		0xFFFF
PMCF	0x06A	R/W		0x18
PSETA	0x070	WO	以位为单位来置位端口输出锁存器的寄存器	0x00
PSETB	0x072	WO		0x00
PSETF	0x07A	WO		0x00
PCLRA	0x080	WO	以位为单位来清零端口输出锁存器的寄存器	0x00
PCLRB	0x082	WO		0x00
PCLRF	0x08A	WO		0x00
PREADA	0x090	RO	在端口作为数字 GPIO 时, 可以通过读此寄存器得到对应端口电平	0xx
PREADB	0x092	RO		0xx
PREADF	0x09A	RO		0xx
PTTLA	0x0A0	R/W	GPIO 的 TTL/SIMT 输入选择, 默认选择 SMIT 输入	0x00
PTTLF	0x0AA	R/W		

2.3.2 输出输入复用功能寄存器映射

(输出输入复用功能寄存器的基地址=0x40040800)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值	使用条件	优先级
PS_INTPO	0x00	R/W	外部中断 0 输入端口选择寄存器	0x3	PMC=0, PM=1	高
PS_INTP1	0x01	R/W	外部中断 1 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP2	0x02	R/W	外部中断 2 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP3	0x03	R/W	外部中断 3 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP4	0x04	R/W	外部中断 4 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP5	0x05	R/W	外部中断 5 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP6	0x06	R/W	外部中断 6 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP7	0x07	R/W	外部中断 7 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_INTP8	0x08	R/W	外部中断 8 输入端口选择寄存器	0x3	PMC=0, PM=1	
PS_SCL	0x10	R/W	IIC SCL 端口选择寄存器	0x7	PMC=0, PM=0	
PS_SDA	0x11	R/W	IIC SDA 端口选择寄存器	0xf	PMC=0, PM=0	
PS_SMBA	0x12	R/W	IIC SMBA 端口选择寄存器	0x1	PMC=0, PM=0	
PS_SCK	0x13	R/W	SPI SCK 端口选择寄存器	0x7	SPI 主机: PMC=0, PM=0 SPI 从机: PMC=0, PM=1	
PS_MISO	0x14	R/W	SPI MISO 端口选择寄存器	0x7	SPI 主机: PMC=0, PM=1 SPI 从机: PMC=0, PM=0	
PS_MOSI	0x15	R/W	SPI MOSI 端口选择寄存器	0x7	SPI 主机: PMC=0, PM=0 SPI 从机: PMC=0, PM=1	
PS_NSS	0x16	R/W	SPI NSS 端口选择寄存器	0x7	SPI 主机: PMC=0, PM=0 SPI 从机: PMC=0, PM=1	
PS_SCK00	0x17	R/W	SPI00 SCLK /IIC00 SCL 端口选择寄存器	0x7	SPI00 主机: PMC=0, PM=0 SPI00 从机: PMC=0, PM=1 IIC00 主/从机: PMC=0, PM=0	
PS_SDI00	0x18	R/W	SPI00 MISI/IIC00 SDA /UART0 RXD 端口选择寄存器	0x7	PMC=0, PM=1	
PS_SDO00	0x19	R/W	SPI00 MOSI/UART0 TXD 端口选择寄存器	0x7	PMC=0, PM=0	
PS_SS00	0x1A	R/W	SPI00 SS 端口选择寄存器	0x3	SPI00 从机: PMC=0, PM=1	
PS_TXD1	0x20	R/W	USART TXD 端口选择寄存器	0xf	PMC=0, PM=0	
PS_RXD1	0x21	R/W	USART RXD 端口选择寄存器	0xf	PMC=0, PM=1	
PS_CTS1	0x22	R/W	USART CTS 端口选择寄存器	0x3	PMC=0, PM=1	
PS_RTS1	0x23	R/W	USART RTS 端口选择寄存器	0x3	PMC=0, PM=0	
PS_CK1	0x24	R/W	USART CK 端口选择寄存器	0x7	PMC=0, PM=0	
PS_T1CH1	0x30	R/W	TIM1 CH1 端口选择寄存器	0x3	做输入时: PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T1CH2	0x31	R/W	TIM1 CH2 端口选择寄存器	0x3	做输入时: PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T1CH3	0x32	R/W	TIM1 CH3 端口选择寄存器	0x3	做输入时: PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T1CH4	0x33	R/W	TIM1 CH4 端口选择寄存器	0x3	做输入时: PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T1CH1N	0x34	R/W	TIM1 CH1N 端口选择寄存器	0x3	PMC=0, PM=0	
PS_T1CH2N	0x35	R/W	TIM1 CH2N 端口选择寄存器	0x3	PMC=0, PM=0	
PS_T1BK	0x36	R/W	TIM1 BKIN 端口选择寄存器	0x3	PMC=0, PM=1	
PS_T3CH1	0x37	R/W	TIM3 CH1 端口选择寄存器	0x3	做输入时: PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T3CH2	0x38	R/W	TIM3 CH2 端口选择寄存器	0x3	做输入时: PMC=0, PM=1 做输出时: PMC=0, PM=0	

PS_T3CH3	0x39	R/W	TIM3 CH3 端口选择寄存器	0x3	做输入时:PMC=0, PM=1 做输出时: PMC=0, PM=0	低
PS_T14CH1	0x3A	R/W	TIM14 CH1 端口选择寄存器	0x7	做输入时:PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T16CH1	0x3B	R/W	TIM16 CH1 端口选择寄存器	0x3	做输入时:PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T17CH1	0x3C	R/W	TIM17 CH1 端口选择寄存器	0x3	做输入时:PMC=0, PM=1 做输出时: PMC=0, PM=0	
PS_T17BK	0x3D	R/W	TIM17 BKIN 端口选择寄存器	0x3	PMC=0, PM=1	
PS_C0O	0x60	R/W	比较器 0 端口选择寄存器	0x7	PMC=0, PM=0	
PS_C1O	0x61	R/W	比较器 1 端口选择寄存器	0x3	PMC=0, PM=0	低
PMS	0x7B	R/W	端口读取模式选择寄存器; 在端口为输出模式时, 该寄存器选择读端口锁存器的值或引脚输出电平	0x00		

2.3.3 特殊功能端口 RESINB 控制寄存器映射

(寄存器的基地址=0x4002_0400)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
RSTM	0x0B	R/W	选择 RESINB(PF2)口作为外部复位口或 GPIO 口的寄存器	0x00

2.4 寄存器说明

通过以下寄存器控制端口：

- ◆ 端口寄存器 (Px)
- ◆ 端口模式寄存器 (PMx)
- ◆ 上拉电阻选择寄存器 (PUx)
- ◆ 下拉电阻选择寄存器 (PDx)
- ◆ 端口输出模式寄存器 (POMx)
- ◆ 端口模式控制寄存器 (PMCx)
- ◆ 端口置位控制寄存器 (PSETx)
- ◆ 端口清零控制寄存器 (PCLRx)
- ◆ 端口输入复用功能配置寄存器 (PS_XX)
- ◆ 端口电平选择寄存器 (PTTLx)
- ◆ 特殊功能端口RESINB控制寄存器(RSTM)

2.4.1 端口寄存器 (Px)

这是以位为单位设定端口输出锁存器值的寄存器 Px (x=A、B、F)。在输入模式中读此寄存器可以得到引脚电平，而在输出模式中读可以得到端口的输出锁存器的值。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	Px15	端口 x 的模式寄存器第 15 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
14	Px14	端口 x 的模式寄存器第 14 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
13	Px13	端口 x 的模式寄存器第 13 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
12	Px12	端口 x 的模式寄存器第 12 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
11	Px11	端口 x 的模式寄存器第 11 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
10	Px10	端口 x 的模式寄存器第 10 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
9	Px9	端口 x 的模式寄存器第 9 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
8	Px8	端口 x 的模式寄存器第 8 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
7	Px7	端口 x 的模式寄存器第 7 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
6	Px6	端口 x 的模式寄存器第 6 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
5	Px5	端口 x 的模式寄存器第 5 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
4	Px4	端口 x 的模式寄存器第 4 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
3	Px3	端口 x 的模式寄存器第 3 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
2	Px2	端口 x 的模式寄存器第 2 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
1	Px1	端口 x 的模式寄存器第 1 位 0: 输出模式时输出“0”；输入模式时输入低电平	0

		1: 输出模式时输出“1”；输入模式时输入高电平	
0	Px0	端口 x 的模式寄存器第 0 位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0

注：必须给未分配的位设定初始值。

2.4.2 端口模式寄存器 (PMx)

当端口作为数字通道使用时，这是以位为单位设定其输入/输出的寄存器 PMx (x=A、B、F)。在产生复位信号后，所有端口默认为输入状态。具体寄存器描述如下：

位	符号	描述	复位值
15	PMx15	端口 x 的模式寄存器第 15 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
14	PMx14	端口 x 的模式寄存器第 14 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
13	PMx13	端口 x 的模式寄存器第 13 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
12	PMx12	端口 x 的模式寄存器第 12 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
11	PMx11	端口 x 的模式寄存器第 11 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
10	PMx10	端口 x 的模式寄存器第 10 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
9	PMx9	端口 x 的模式寄存器第 9 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
8	PMx8	端口 x 的模式寄存器第 8 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
7	PMx7	端口 x 的模式寄存器第 7 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
6	PMx6	端口 x 的模式寄存器第 6 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
5	PMx5	端口 x 的模式寄存器第 5 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
4	PMx4	端口 x 的模式寄存器第 4 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
3	PMx3	端口 x 的模式寄存器第 3 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
2	PMx2	端口 x 的模式寄存器第 2 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1
1	PMx1	端口 x 的模式寄存器第 1 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1

0	PMx0	端口 x 的模式寄存器第 0 位 0: 输出模式(用作输出端口 (输出缓冲器 ON)) 1: 输入模式(用作输入端口 (输出缓冲器 OFF))	1

注：必须给未分配的位设定初始值。

2.4.3 上拉电阻选择寄存器 (PUx)

内部上拉电阻的选择寄存器 PUx (x=A、B、F)。只能在对应的 PMCx 对应位等于 0 的时候，设置上拉电阻有效。

在产生复位信号后，PA13、PF2 端口上拉功能自动打开，其他端口的上拉功能默认不打开。具体寄存器描述如下：

位	符号	描述	复位值
15	PUx15	Px 引脚的内部上拉电阻选择第 15 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
14	PUx14	Px 引脚的内部上拉电阻选择第 14 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
13	PUx13	Px 引脚的内部上拉电阻选择第 13 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	当 x=A 时，复位值为 1； 当 x=其他时，复位值为 0
12	PUx12	Px 引脚的内部上拉电阻选择第 12 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
11	PUx11	Px 引脚的内部上拉电阻选择第 11 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
10	PUx10	Px 引脚的内部上拉电阻选择第 10 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
9	PUx9	Px 引脚的内部上拉电阻选择第 9 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
8	PUx8	Px 引脚的内部上拉电阻选择第 8 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
7	PUx7	Px 引脚的内部上拉电阻选择第 7 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
6	PUx6	Px 引脚的内部上拉电阻选择第 6 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
5	PUx5	Px 引脚的内部上拉电阻选择第 5 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
4	PUx4	Px 引脚的内部上拉电阻选择第 4 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
3	PUx3	Px 引脚的内部上拉电阻选择第 3 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
2	PUx2	Px 引脚的内部上拉电阻选择第 2 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	当 x=F 时，复位值为 1； 当 x=其他时，复位值为 0

1	PUx1	Px 引脚的内部上拉电阻选择第 1 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
0	PUx0	Px 引脚的内部上拉电阻选择第 0 位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0

注：必须给未分配的位设定初始值。

2.4.4 下拉电阻选择寄存器 (PDx)

内部下拉电阻选择寄存器 PDx (x=A、B、F)。只能在对应的 PMCx 对应位等于 0 的时候, 设置下拉电阻有效。

在产生复位信号后, PA14 端口下拉功能自动打开, 其他端口的下拉功能默认不打开。具体寄存器描述如下:

位	符号	描述	复位值
15	PDx15	Px 引脚的内部下拉电阻选择第 15 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
14	PDx14	Px 引脚的内部下拉电阻选择第 14 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	当 x=A 时, 复位值为 1; 当 x=其他时, 复位值为 0
13	PDx13	Px 引脚的内部下拉电阻选择第 13 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
12	PDx12	Px 引脚的内部下拉电阻选择第 12 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
11	PDx11	Px 引脚的内部下拉电阻选择第 11 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
10	PDx10	Px 引脚的内部下拉电阻选择第 10 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
9	PDx9	Px 引脚的内部下拉电阻选择第 9 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
8	PDx8	Px 引脚的内部下拉电阻选择第 8 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
7	PDx7	Px 引脚的内部下拉电阻选择第 7 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
6	PDx6	Px 引脚的内部下拉电阻选择第 6 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
5	PDx5	Px 引脚的内部下拉电阻选择第 5 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
4	PDx4	Px 引脚的内部下拉电阻选择第 4 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
3	PDx3	Px 引脚的内部下拉电阻选择第 3 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
2	PDx2	Px 引脚的内部下拉电阻选择第 2 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
1	PDx1	Px 引脚的内部下拉电阻选择第 1 位 0: 不连接内部下拉电阻	0

		1: 连接内部下拉电阻	
0	PDx0	Px 引脚的内部下拉电阻选择第 0 位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0

注：必须给未分配的位设定初始值。

2.4.5 端口输出模式寄存器（POMx）

端口输出模式寄存器 POMx（x=A、B、F），只有在配置成输出模式 N 沟道漏极开路才会开启。端口复用为 IIC 功能时，会强制开启开漏模式。

在产生复位信号后，这些寄存器的值变为“00H”。

注意：对于设定 N 沟道漏极开路输出模式（POMmn=1）的位，硬件不自动连接内部上拉电阻，若用户需要开启上拉电阻，须手动开启。

具体寄存器描述如下：

位	符号	描述	复位值
15	POMx15	Px 引脚输出模式选择第 15 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
14	POMx14	Px 引脚输出模式选择第 14 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
13	POMx13	Px 引脚输出模式选择第 13 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
12	POMx12	Px 引脚输出模式选择第 12 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
11	POMx11	Px 引脚输出模式选择第 11 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
10	POMx10	Px 引脚输出模式选择第 10 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
9	POMx9	Px 引脚输出模式选择第 9 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
8	POMx8	Px 引脚输出模式选择第 8 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
7	POMx7	Px 引脚输出模式选择第 7 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
6	POMx6	Px 引脚输出模式选择第 6 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
5	POMx5	Px 引脚输出模式选择第 5 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
4	POMx4	Px 引脚输出模式选择第 4 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
3	POMx3	Px 引脚输出模式选择第 3 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
2	POMx2	Px 引脚输出模式选择第 2 位	0

		0: 通常输出模式 1: N 沟道漏极开路输出模式	
1	POMx1	Px 引脚输出模式选择第 1 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0
0	POMx0	Px 引脚输出模式选择第 0 位 0: 通常输出模式 1: N 沟道漏极开路输出模式	0

注：必须给未分配的位设定初始值。

2.4.6 端口模式控制寄存器 (PMCx)

端口模式寄存器 PMCx(x=A、B、F)，以位为单位设定端口作为数字（输入/输出）或者作为模拟（输入）通道。

在产生复位信号后，PA13、PA14、PF2、PF0、PF1 默认作为数字通道使用，其他端口默认作为模拟通道使用即 PMCx 对应位等于 1。具体寄存器描述如下：

位	符号	描述	复位值
15	PMCx15	Px 引脚数字（输入/输出）或模拟（输入）第 15 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
14	PMCx14	Px 引脚数字（输入/输出）或模拟（输入）第 14 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当 x=A 时，复位值为 0； 当 x=其他时，复位值为 1
13	PMCx13	Px 引脚数字（输入/输出）或模拟（输入）第 13 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当 x=A 时，复位值为 0； 当 x=其他时，复位值为 1
12	PMCx12	Px 引脚数字（输入/输出）或模拟（输入）第 12 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
11	PMCx11	Px 引脚数字（输入/输出）或模拟（输入）第 11 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
10	PMCx10	Px 引脚数字（输入/输出）或模拟（输入）第 10 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
9	PMCx9	Px 引脚数字（输入/输出）或模拟（输入）第 9 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
8	PMCx8	Px 引脚数字（输入/输出）或模拟（输入）第 8 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
7	PMCx7	Px 引脚数字（输入/输出）或模拟（输入）第 7 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
6	PMCx6	Px 引脚数字（输入/输出）或模拟（输入）第 6 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
5	PMCx5	Px 引脚数字（输入/输出）或模拟（输入）第 5 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
4	PMCx4	Px 引脚数字（输入/输出）或模拟（输入）第 4 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
3	PMCx3	Px 引脚数字（输入/输出）或模拟（输入）第 3 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
2	PMCx2	Px 引脚数字（输入/输出）或模拟（输入）第 2 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当 x=F 时，复位值为 0； 当 x=其他时，复位值为 1

1	PMCx1	Px 引脚数字（输入/输出）或模拟（输入）第 1 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当 x=F 时，复位值为 0； 当 x=其他时，复位值为 1
0	PMCx0	Px 引脚数字（输入/输出）或模拟（输入）第 0 位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当 x=F 时，复位值为 0； 当 x=其他时，复位值为 1

注：必须给未分配的位设定初始值。

2.4.7 端口置位控制寄存器（PSETx）

这是以位为单位来置位端口输出锁存器的寄存器 PSETx（x=A、B、F）。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	PSETx15	Px 引脚的置位控制第 15 位 0: 无操作 1: 对应的 Px15 置 1	0
14	PSETx14	Px 引脚的置位控制第 14 位 0: 无操作 1: 对应的 Px14 置 1	0
13	PSETx13	Px 引脚的置位控制第 13 位 0: 无操作 1: 对应的 Px13 置 1	0
12	PSETx12	Px 引脚的置位控制第 12 位 0: 无操作 1: 对应的 Px12 置 1	0
11	PSETx11	Px 引脚的置位控制第 11 位 0: 无操作 1: 对应的 Px11 置 1	0
10	PSETx10	Px 引脚的置位控制第 10 位 0: 无操作 1: 对应的 Px10 置 1	0
9	PSETx9	Px 引脚的置位控制第 9 位 0: 无操作 1: 对应的 Px9 置 1	0
8	PSETx8	Px 引脚的置位控制第 8 位 0: 无操作 1: 对应的 Px8 置 1	0
7	PSETx7	Px 引脚的置位控制第 7 位 0: 无操作 1: 对应的 Px7 置 1	0
6	PSETx6	Px 引脚的置位控制第 6 位 0: 无操作 1: 对应的 Px6 置 1	0
5	PSETx5	Px 引脚的置位控制第 5 位 0: 无操作 1: 对应的 Px5 置 1	0
4	PSETx4	Px 引脚的置位控制第 4 位 0: 无操作 1: 对应的 Px4 置 1	0
3	PSETx3	Px 引脚的置位控制第 3 位 0: 无操作 1: 对应的 Px3 置 1	0
2	PSETx2	Px 引脚的置位控制第 2 位 0: 无操作 1: 对应的 Px2 置 1	0
1	PSETx1	Px 引脚的置位控制第 1 位 0: 无操作 1: 对应的 Px1 置 1	0

0	PSETx0	Px 引脚的置位控制第 0 位 0: 无操作 1: 对应的 Px0 置 1	0

注：必须给未分配的位设定初始值。

2.4.8 端口清零控制寄存器（PCLR_x）

这是以位为单位来清零端口输出锁存器的寄存器 PCLR_x（x=A、B、F）。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	PCLR _x 15	Px 引脚的清零控制第 15 位 0: 无操作 1: 对应的 Px15 清零	0
14	PCLR _x 14	Px 引脚的清零控制第 14 位 0: 无操作 1: 对应的 Px14 清零	0
13	PCLR _x 13	Px 引脚的清零控制第 13 位 0: 无操作 1: 对应的 Px13 清零	0
12	PCLR _x 12	Px 引脚的清零控制第 12 位 0: 无操作 1: 对应的 Px12 清零	0
11	PCLR _x 11	Px 引脚的清零控制第 11 位 0: 无操作 1: 对应的 Px11 清零	0
10	PCLR _x 10	Px 引脚的清零控制第 10 位 0: 无操作 1: 对应的 Px10 清零	0
9	PCLR _x 9	Px 引脚的清零控制第 9 位 0: 无操作 1: 对应的 Px9 清零	0
8	PCLR _x 8	Px 引脚的清零控制第 8 位 0: 无操作 1: 对应的 Px8 清零	0
7	PCLR _x 7	Px 引脚的清零控制第 7 位 0: 无操作 1: 对应的 Px7 清零	0
6	PCLR _x 6	Px 引脚的清零控制第 6 位 0: 无操作 1: 对应的 Px6 清零	0
5	PCLR _x 5	Px 引脚的清零控制第 5 位 0: 无操作 1: 对应的 Px5 清零	0
4	PCLR _x 4	Px 引脚的清零控制第 4 位 0: 无操作 1: 对应的 Px4 清零	0
3	PCLR _x 3	Px 引脚的清零控制第 3 位 0: 无操作 1: 对应的 Px3 清零	0
2	PCLR _x 2	Px 引脚的清零控制第 2 位 0: 无操作 1: 对应的 Px2 清零	0
1	PCLR _x 1	Px 引脚的清零控制第 1 位 0: 无操作 1: 对应的 Px1 清零	0

0	PCLR _x 0	Px 引脚的清零控制第 0 位 0: 无操作 1: 对应的 Px0 清零	0

注：必须给未分配的位设定初始值。

2.4.9 端口回读寄存器(PREADx)

这是一个只读寄存器 PREADx (x=A、B、F)，在端口作为数字 GPIO 时，可以通过读此寄存器得到对应端口电平。

位	符号	描述	复位值
15	PREADx15	Px 引脚的端口电平第 15 位 0: 端口为低电平 1: 端口为高电平	x
14	PREADx14	Px 引脚的端口电平第 14 位 0: 端口为低电平 1: 端口为高电平	x
13	PREADx13	Px 引脚的端口电平第 13 位 0: 端口为低电平 1: 端口为高电平	x
12	PREADx12	Px 引脚的端口电平第 12 位 0: 端口为低电平 1: 端口为高电平	x
11	PREADx11	Px 引脚的端口电平第 11 位 0: 端口为低电平 1: 端口为高电平	x
10	PREADx10	Px 引脚的端口电平第 10 位 0: 端口为低电平 1: 端口为高电平	x
9	PREADx9	Px 引脚的端口电平第 9 位 0: 端口为低电平 1: 端口为高电平	x
8	PREADx8	Px 引脚的端口电平第 8 位 0: 端口为低电平 1: 端口为高电平	x
7	PREADx7	Px 引脚的端口电平第 7 位 0: 端口为低电平 1: 端口为高电平	x
6	PREADx6	Px 引脚的端口电平第 6 位 0: 端口为低电平 1: 端口为高电平	X
5	PREADx5	Px 引脚的端口电平第 5 位 0: 端口为低电平 1: 端口为高电平	x
4	PREADx4	Px 引脚的端口电平第 4 位 0: 端口为低电平 1: 端口为高电平	x
3	PREADx3	Px 引脚的端口电平第 3 位 0: 端口为低电平 1: 端口为高电平	x
2	PREADx2	Px 引脚的端口电平第 2 位 0: 端口为低电平 1: 端口为高电平	x
1	PREADx1	Px 引脚的端口电平第 1 位 0: 端口为低电平 1: 端口为高电平	x

0	PREADx0	Px 引脚的端口电平第 0 位 0: 端口为低电平 1: 端口为高电平	x

2.4.10 端口模式选择寄存器(PMS)

此寄存器选择在引脚为输出模式（端口模式寄存器（PMm）的 PMmn 位为“0”）时是读端口的输出锁存器的值还是读引脚的输出电平。

通过 8 位存储器操作指令设定 PMS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	-	保留	-
0	PMS0	在引脚为输出模式时读数据的选择 0: 读 Pmn 寄存器的值。 1: 读引脚的数字输出电平。	0

2.4.11 TTL 和施密特输入选择(PTTLx)

这是以位为单位来选择输入电平的寄存器 PTTLx（x=A、F）。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	PTTLx15	Px 引脚输入电平选择控制第 15 位 0: 施密特输入 1: TTL 输入	0
14	PTTLx14	Px 引脚输入电平选择控制第 14 位 0: 施密特输入 1: TTL 输入	0
13	PTTLx13	Px 引脚输入电平选择控制第 13 位 0: 施密特输入 1: TTL 输入	0
12	PTTLx12	Px 引脚输入电平选择控制第 12 位 0: 施密特输入 1: TTL 输入	0
11	PTTLx11	Px 引脚输入电平选择控制第 11 位 0: 施密特输入 1: TTL 输入	0
10	PTTLx10	Px 引脚输入电平选择控制第 10 位 0: 施密特输入 1: TTL 输入	0
9	PTTLx9	Px 引脚输入电平选择控制第 9 位 0: 施密特输入 1: TTL 输入	0
8	PTTLx8	Px 引脚输入电平选择控制第 8 位 0: 施密特输入 1: TTL 输入	0
7	PTTLx7	Px 引脚输入电平选择控制第 7 位 0: 施密特输入 1: TTL 输入	0
6	PTTLx6	Px 引脚输入电平选择控制第 6 位 0: 施密特输入	0

		1: TTL 输入	
5	PTTLx5	Px 引脚输入电平选择控制第 5 位 0: 施密特输入 1: TTL 输入	0
4	PTTLx4	Px 引脚输入电平选择控制第 4 位 0: 施密特输入 1: TTL 输入	0
3	PTTLx3	Px 引脚输入电平选择控制第 3 位 0: 施密特输入 1: TTL 输入	0
2	PTTLx2	Px 引脚输入电平选择控制第 2 位 0: 施密特输入 1: TTL 输入	0
1	PTTLx1	Px 引脚输入电平选择控制第 1 位 0: 施密特输入 1: TTL 输入	0
0	PTTLx0	Px 引脚输入电平选择控制第 0 位 0: 施密特输入 1: TTL 输入	0

注：必须给未分配的位设定初始值，仅仅 PA2/PA3，PF0/PF1 支持 TTL 和施密特输入选择。

2.4.12 端口输出复用功能配置寄存器（PS_XX）

端口输出复用功能配置寄存器 PS_XX 可实现将一部分外围模块的输出功能映射到端口，具体映射功能参见表 2-1；端口输出复用功能配置寄存器的复位值为“FFH”，此时端口为默认的兼用功能和 GPIO 功能。设置步骤如下：

- ◆ 设置PMCmn=0，选择数字功能
- ◆ 根据PS_XX寄存器功能，设置输入或输出功能（参考2.3.2列表信息）
- ◆ 设置PS_XX寄存器选择对应管脚
- ◆ 开启对应模块的功能（如IIC，SPI等）

注：

- （1）IIC_SCL/IIC_SDA 的功能按照上述步骤设置后会自动开启开漏功能，如果需要接上拉电阻，可以外接上拉或单独设置PUx寄存器
- （2）通过PS_XX寄存器选择GPIO，说明该功能具有唯一性，不能通过多个GPIO同时输出。
- （3）不同功能选择PS_XX后，输出具有优先级（PMx=0），优先级的顺序按照2.3.2表格所描述。

2.4.12.1 INTP0 输入端口配置寄存器（PS_INTP0）

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP0[1:0]	PS_INTP0 输入选择 0x00: PA0 对应 INTP0 的输入功能 0x01: PB0 对应 INTP0 的输入功能 0x02: PF0 对应 INTP0 的输入功能 其他: GPIO 功能	0x03

2.4.12.2 INTP1 输入端口配置寄存器（PS_INTP1）

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP1[1:0]	PS_INTP1 输入选择 0x00: PA1 对应 INTP1 的输入功能 0x01: PB1 对应 INTP1 的输入功能 0x02: PF1 对应 INTP1 的输入功能 其他: GPIO 功能	0x03

2.4.12.3 INTP2 输入端口配置寄存器（PS_INTP2）

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP2[1:0]	PS_INTP2 输入选择 0x00: PA2 对应 INTP2 的输入功能 0x01: PB2 对应 INTP2 的输入功能 0x02: PF2 对应 INTP2 的输入功能 其他: GPIO 功能	0x03

2.4.12.4 INTP3 输入端口配置寄存器 (PS_INTP3)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP3[1:0]	PS_INTP3 输入选择 0x00: PA3 对应 INTP3 的输入功能 0x01: PB3 对应 INTP3 的输入功能 0x02: 禁止配置 其他: GPIO 功能	0x03

2.4.12.5 INTP4 输入端口配置寄存器 (PS_INTP4)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP4[1:0]	PS_INTP4 输入选择 0x00: PA4 对应 INTP4 的输入功能 0x01: PB4 对应 INTP4 的输入功能 0x02: 禁止配置 其他: GPIO 功能	0x03

2.4.12.6 INTP5 输入端口配置寄存器 (PS_INTP5)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP5[1:0]	PS_INTP5 输入选择 0x00: 禁止配置 0x01: PB5 对应 INTP5 的输入功能 其他: GPIO 功能	0x03

2.4.12.7 INTP6 输入端口配置寄存器 (PS_INTP6)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP6[1:0]	PS_INTP6 输入选择 0x00: PA6 对应 INTP6 的输入功能 0x01: 禁止配置 其他: GPIO 功能	0x03

2.4.12.8 INTP7 输入端口配置寄存器 (PS_INTP7)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP7[1:0]	PS_INTP7 输入选择 0x00: PA7 对应 INTP7 的输入功能 0x01: 禁止配置 其他: GPIO 功能	0x03

2.4.12.9 INTP8 输入端口配置寄存器 (PS_INTP8)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_INTP8[1:0]	PS_INTP8 输入选择 0x00: PA8 对应 INTP8 的输入功能 0x01: 禁止配置 其他: GPIO 功能	0x03

2.4.12.10 IIC_SCL 端口配置寄存器(PS_SCL)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SCL [2:0]	IIC 通信的 SCL 端口选择 0x00: PA3 对应 IIC SCL 功能 0x01: PA8 对应 IIC SCL 功能 0x02: PA9 对应 IIC SCL 功能 0x03: PA10 对应 IIC SCL 功能 0x04: 禁止配置 0x05: 禁止配置 0x06: PF1 对应 IIC SCL 功能 其他: GPIO 功能	0x7

2.4.12.11 IIC_SDA 端口配置寄存器(PS_SDA)

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_SDA [3:0]	IIC 通信的 SDA 端口选择 0x00: PA2 对应 IIC SDA 功能 0x01: PA7 对应 IIC SDA 功能 0x02: PA9 对应 IIC SDA 功能 0x03: PA10 对应 IIC SDA 功能 0x04: PA12 对应 IIC SDA 功能 0x05: 禁止配置 0x06: 禁止配置 0x07: PF0 对应 IIC SDA 功能 其他: GPIO 功能	0xF

2.4.12.12 IIC_SMBA 端口配置寄存器(PS_SMBA)

位	符号	描述	复位值
7:1	--	保留	--
0	PS_SMBA	IIC 通信的 SMBA 端口选择 0x00: PB5 对应 IIC SMBA 功能 其他: GPIO 功能	0x1

2.4.12.13 SPI_SCK SPI 的 SCK 端口配置寄存器 (PS_SCK)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SCK[2:0]	SPI 通信的 SCK 端口选择 0x00: PA1 对应 SPI_SCK 功能 0x01: PA2 对应 SPI_SCK 功能 0x02: 禁止配置 0x03: PA9 对应 SPI_SCK 功能 0x04: PB3 对应 SPI_SCK 功能 其他: GPIO 功能	0x7

2.4.12.14 SPI_MISO SPI 主机输入/从机输出端口配置寄存器 (PS_MISO)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_MISO [2:0]	SPI 通信的 MISO 端口选择 0x00: PA0 对应 SPI_MISO 功能 0x01: PA6 对应 SPI_MISO 功能 0x02: PA7 对应 SPI_MISO 功能 0x03: 禁止配置 0x04: PA13 对应 SPI_MISO 功能 0x05: PB4 对应 SPI_MISO 功能 其他: GPIO 功能	0x7

2.4.12.15 SPI_MOSI SPI 主机输入/从机输出端口配置寄存器 (PS_MOSI)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_MOSI [2:0]	SPI 通信的 MOSI 端口选择 0x00: PA1 对应 SPI_MOSI 功能 0x01: PA2 对应 SPI_MOSI 功能 0x02: PA3 对应 SPI_MOSI 功能 0x03: PA7 对应 SPI_MOSI 功能 0x04: PA8 对应 SPI_MOSI 功能 0x05: PA12 对应 SPI_MOSI 功能 0x06: PB5 对应 SPI_MOSI 功能 其他: GPIO 功能	0x7

2.4.12.16 SPI_NSS SPI 从机片选信号端口配置寄存器 (PS_NSS)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_NSS [2:0]	SPI 通信的 NSS 端口选择 0x00: PA4 对应 SPI_NSS 功能 0x01: PA10 对应 SPI_NSS 功能 0x02: 禁止配置 0x03: PB0 对应 SPI_NSS 功能 0x04: PF1 对应 SPI_NSS 功能 0x05: 禁止配置 其他: GPIO 功能	0x7

2.4.12.17 串行通信单元 SPI00 SCK/IIC00 SCL 端口配置寄存器 (PS_SCK00)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SCK00[2:0]	SAU 通信的 SCK/SCL 端口选择 0x00: PA0 对应 SPI00 SCK, IIC00 SCL 功能 0x01: PB2 对应 SPI00 SCK, IIC00 SCL 功能 0x02: 禁止配置 0x03: PF0 对应 SPI00 SCK, IIC00 SCL 功能 其他: GPIO 功能	0x7

2.4.12.18 串行通信单元 SPI00 MISI/IIC00 SDA/ UART0 RXD 端口配置寄存器 (PS_SDI00)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI00[2:0]	SAU 通信的 MISI/SDA/RXD0 端口选择 0x00: PA3 对应 SPI00 MISI, IIC00 SDA, UART0 RXD 功能 0x01: PA9 对应 SPI00 MISI, IIC00 SDA, UART0 RXD 功能 0x02: 禁止配置 0x03: PF1 对应 SPI00 MISI, IIC00 SDA, UART0 RXD 功能 0x04: 禁止配置 其他: GPIO 功能	0x7

2.4.12.19 串行通信单元 SPI00 MOSO/ UART0 TXD 端口配置寄存 (PS_SDO00)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO00[2:0]	SAU 通信的 MOSO/TXD0 端口选择 0x00: PA4 对应 SPI00 MOSO, UART0 TXD 功能 0x01: PA10 对应 SPI00 MOSO, UART0 TXD 功能 0x02: 禁止配置 0x03: PF2 对应 SPI00 MOSO, UART0 TXD 功能 其他: GPIO 功能	0x3

2.4.12.20 串行通信单元 SPI00 SS 端口配置寄存器 (PS_SS00)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SS00 [1:0]	SAU 通信的 SS00 端口选择 0x00: PA8 对应 SPI00 SS 功能 0x01: 禁止配置 其他: GPIO 功能	0x3

2.4.12.21 USART_TXD1 USART 输出端口配置寄存器 (PS_TXD1)

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TXD1 [3:0]	USART 通信的 TXD1 端口选择 0x00: PA2 对应 USART TXD1 功能 0x01: PA7 对应 USART TXD1 功能 0x02: PA9 对应 USART TXD1 功能 0x03: PA10 对应 USART TXD1 功能 0x04: PA14 对应 USART TXD1 功能 0x05: 禁止配置 0x06: 禁止配置 0x07: PF1 对应 USART TXD1 功能 0x08: 禁止配置 其他: GPIO 功能	0xF

2.4.12.22 USART_RXD1 USART 输入端口配置寄存器 (PS_RXD1)

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_RXD1 [3:0]	USART 通信的 RXD1 端口选择 0x00: PA3 对应 USART RXD1 功能 0x01: PA8 对应 USART RXD1 功能 0x02: PA9 对应 USART RXD1 功能 0x03: PA10 对应 USART RXD1 功能 0x04: PA13 对应 USART RXD1 功能 0x05: 禁止配置 0x06: PB2 对应 USART RXD1 功能 0x07: 禁止配置 0x08: PF0 对应 USART RXD1 功能 其他: GPIO 功能	0xF

2.4.12.23 USART_CTS1 USART 硬件流输入端口配置寄存器 (PS_CTS1)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_CTS1 [1:0]	USART 通信的 CTS1 端口选择 0x00: PA0 对应 USART CTS1 功能 0x01: 禁止配置 0x02: PB4 对应 USART CTS1 功能 其他: GPIO 功能	0x3

2.4.12.24 USART_RTS1 UART 硬件流输出端口配置寄存器 (PS_RTS1)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_RTS1 [1:0]	USART 通信的 RTS1 端口选择 0x00: PA1 对应 USART RTS1 功能 0x01: PA12 对应 USART RTS1 功能 0x02: PB3 对应 USART RTS1 功能 其他: GPIO 功能	0x3

2.4.12.25 USART_CK1 USART 同步模式时钟端口配置寄存器 (PS_CK1)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_CK1 [2:0]	USART 通信的 CK1 端口选择 0x00: PA4 对应 USART CK1 功能 0x01: PA6 对应 USART CK1 功能 0x02: PA8 对应 USART CK1 功能 0x03: PB5 对应 USART CK1 功能 其他: GPIO 功能	0x7

2.4.12.26 TIM1_CH1 端口配置寄存器 (PS_T1CH1)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1CH1 [1:0]	TIM1 CH1 端口选择 0x00: PA3 对应 TIM1 CH1 功能 0x01: PA8 对应 TIM1 CH1 功能 其他: GPIO 功能	0x3

2.4.12.27 TIM1_CH2 端口配置寄存器 (PS_T1CH2)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1CH2 [1:0]	TIM1 CH2 端口选择 0x00: PA9 对应 TIM1 CH2 功能 0x01: PA13 对应 TIM1 CH2 功能 0x02: PB3 对应 TIM1 CH2 功能 其他: GPIO 功能	0x3

2.4.12.28 TIM1_CH3 端口配置寄存器 (PS_T1CH3)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1CH3 [1:0]	TIM1 CH3 端口选择 0x00: PA0 对应 TIM1 CH3 功能 0x01: PA10 对应 TIM1 CH3 功能 0x02: 禁止配置 其他: GPIO 功能	0x3

2.4.12.29 TIM1 CH4 端口配置寄存器 (PS_T1CH4)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1CH4 [1:0]	TIM1 CH4 端口选择 0x00: PA1 对应 TIM1 CH4 功能 0x01: 禁止配置 其他: GPIO 功能	0x3

2.4.12.30 TIM1 CH1N 端口配置寄存器 (PS_T1CH1N)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1CH1N[1:0]	TIM1 CH1N 端口选择 0x00: PA0 对应 TIM1 CH1N 功能 0x01: PA7 对应 TIM1 CH1N 功能 其他: GPIO 功能	0x3

2.4.12.31 TIM1 CH2N 端口配置寄存器 (PS_T1CH2N)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1CH2N [1:0]	TIM1 CH2N 端口选择 0x00: PA1 对应 TIM1 CH2N 功能 0x01: PB0 对应 TIM1 CH2N 功能 其他: GPIO 功能	0x3

2.4.12.32 TIM1 BKIN 端口配置寄存器 (PS_T1BK)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T1BKIN [1:0]	TIM1BKIN 端口选择 0x00: PA6 对应 TIM1 BKIN 功能 0x01: PA9 对应 TIM1 BKIN 功能 其他: GPIO 功能	0x3

2.4.12.33 TIM3 CH1 端口配置寄存器 (PS_T3CH1)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T3CH1 [1:0]	TIM3 CH1 端口选择 0x00: PA2 对应 TIM3 CH1 功能 0x01: PA6 对应 TIM3 CH1 功能 0x02: PB4 对应 TIM3 CH1 功能 其他: GPIO 功能	0x3

2.4.12.34 TIM3 CH2 端口配置寄存器 (PS_T3CH2)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T3CH2 [1:0]	TIM3 CH2 端口选择 0x00: 禁止配置 0x01: PA7 对应 TIM3 CH2 功能 0x02: PB5 对应 TIM3 CH2 功能 其他: GPIO 功能	0x3

2.4.12.35 TIM3 CH3 端口配置寄存器 (PS_T3CH3)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T3CH3 [1:0]	TIM3 CH3 端口选择 0x00: PA4 对应 TIM3 CH3 功能 0x01: PB0 对应 TIM3 CH3 功能 0x02: 禁止配置 其他: GPIO 功能	0x3

2.4.12.36 TIM14 CH1 端口配置寄存器 (PS_T14CH1)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_T14CH1 [2:0]	TIM14 CH1 端口选择 0x00: PA4 对应 TIM14 CH1 功能 0x01: PA7 对应 TIM14 CH1 功能 0x02: PB1 对应 TIM14 CH1 功能 0x03: PF0 对应 TIM14 CH1 功能 0x04: PF1 对应 TIM14 CH1 功能 其他: GPIO 功能	0x7

2.4.12.37 TIM16 CH1 输入端口配置寄存器 (PS_T16CH1)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T16CH1 [1:0]	TIM16 CH1 端口选择 0x00: PA6 对应 TIM16 CH1 功能 0x01: 禁止配置 其他: GPIO 功能	0x3

2.4.12.38 TIM17 CH1 端口配置寄存器 (PS_T17CH1)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_T17CH1 [1:0]	TIM17 CH1 端口选择 0x00: PA7 对应 TIM17 CH1 功能 0x01: 禁止配置 其他: GPIO 功能	0x3

2.4.12.39 TIM17 BKIN 端口配置寄存器 (PS_T17BKIN)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_BKIN [1:0]	TIM17 BKIN 端口选择 0x00: PA10 对应 TIM17 BKIN 功能 0x01: PB4 对应 TIM17 BKIN 功能 其他: GPIO 功能	0x3

2.4.12.40 比较器 0 输出端口配置寄存器 (PS_C0O)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_C0O[2:0]	比较器 0 输出端口选择 0x00: PA0 对应 C0OUT0 功能 0x01: PA6 对应 C0OUT1 功能 0x02: PB0 对应 C0OUT2 功能 0x03: 禁止配置 0x04: PB5 对应 C0OUT4 功能 其他: GPIO 功能	0x7

2.4.12.41 比较器 1 输出端口配置寄存器 (PS_C1O)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_C1O[1:0]	比较器 1 输出端口选择 0x00: PA2 对应 C1OUT0 功能 0x01: PA7 对应 C1OUT1 功能 0x02: PA12 对应 C1OUT2 功能 其他: GPIO 功能	0x3

2.4.13 特殊功能端口 RESINB 说明(RSTM)

本产品上电默认 RESINB（PF2）有效，如果需要把该口做 GPIO 使用，那需要通过寄存器关掉复位功能，寄存器描述如下：

位	符号	描述	复位值
7:1	--	保留为 0	0
0	RSTM	RESINB 引脚的外部复位功能屏蔽 0: RESINB 外部复位引脚 1: RESINB 做 GPIO 引脚	0

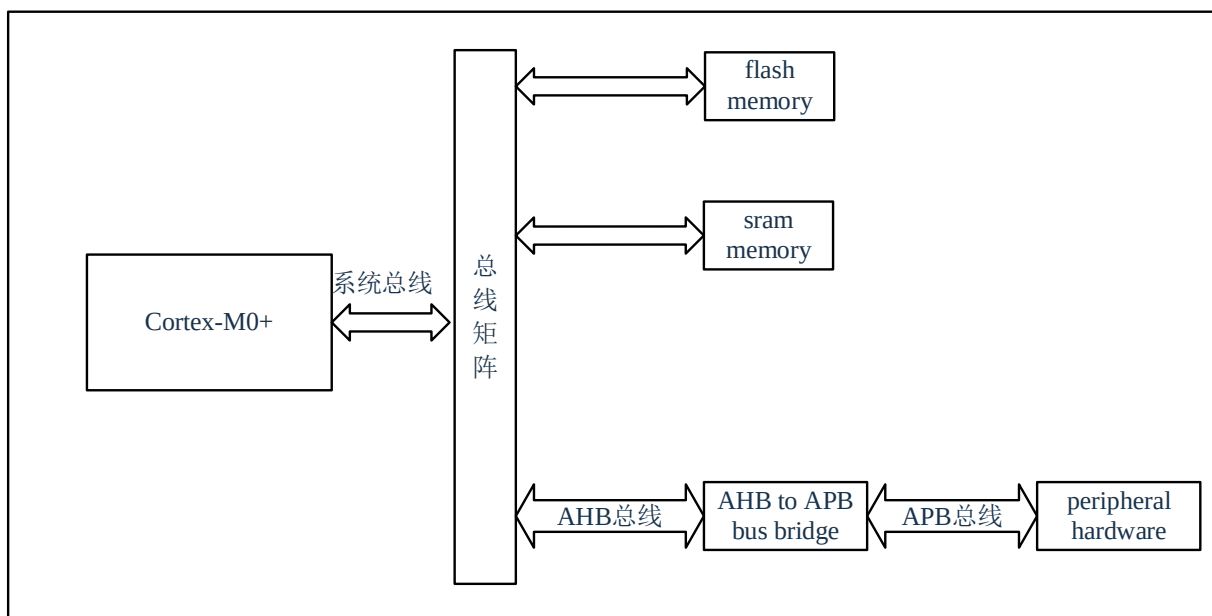
第3章 系统结构

3.1 概述

本产品系统由以下部分组成：

- 1 个 AHB 总线 Master:
 - Cortex-M0+
- 3个AHB总线Slaves:
 - FLASH存储器
 - SRAM存储器
 - AHB to APB Bridge, 包含所有APB接口外设

图 3-1 系统结构示意图



系统总线：此总线连接 Cortex-M0+内核的系统总线(外设总线)到总线矩阵。

总线矩阵：总线矩阵协调内核系统总线其他总线的访问。

AHB to APB Bridge: AHB to APB Bridge 在 AHB 和 APB 总线间提供同步连接。有关连接到每个桥的不同外设的地址映射请参考表 3-1。

3.2 系统地址划分

图 3-2 地址区域划分示意图



外设地址分配

表 3-1 外设的寄存器组起始地址

起始地址	外设	备注
0x4000_0000 - 0x4000_2FFF	保留	
0x4000_3000 - 0x4000_33FF	通用 CRC	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	保留	
0x4000_7000 - 0x4001_FFFF	保留	
0x4002_0000 - 0x4002_03FF	FLASH 控制	
0x4002_0400 - 0x4002_0FFF	时钟控制	
0x4002_1100 - 0x4002_17FF	保留	
0x4002_1800 - 0x4002_1BFF	高速 CRC	详见安全功能
0x4002_1C00 - 0x4002_1FFF	时钟控制	
0x4002_2000 - 0x4003_FFFF	保留	
0x4004_0000 - 0x4004_0F9F	GPIO	
0x4004_0FA5 - 0x4004_0FA6	保留	
0x4004_1500 - 0x4004_1543	串行通信单元 (SAU)	
0x4004_4B50 - 0x4004_4B50	LPTIME	
0x4004_5000 - 0x4004_53FF	AD 转换器	
0x4004_5400 - 0x4004_5AFF	保留	
0x4004_5B30 - 0x4004_5BFF	外部中断控制	
0x4004_5C00 - 0x4006_03FF	保留	
0x4006_0400 - 0x4006_07FF	TIM3	
0x4006_2000 - 0x4006_23FF	TIM14	
0x4006_2C00 - 0x4006_2FFF	WWDG	
0x4006_3000 - 0x4006_33FF	IWDG	
0x4006_3800 - 0x4006_3BFF	保留	
0x4006_4400 - 0x4006_47FF	保留	
0x4006_5400 - 0x4006_57FF	IIC	
0x4006_8200 - 0x4006_827F	ACMP	
0x4006_8360 - 0x4006_836F	-	-
0x4007_0000 - 0x4007_03FF	TIM1	
0x4007_1000 - 0x4007_13FF	USART	
0x4007_3000 - 0x4007_33FF	SPI	
0x4007_4400 - 0x4007_47FF	TIM16	
0x4007_4800 - 0x4007_4BFF	TIM17	
0x4006_9000 - 0x4006_901F	保留	
0x4008_0200 - 0xDFFF_FFFF	保留	

第4章 时钟发生电路

4.1 时钟发生电路的功能

时钟发生电路是产生给 CPU 和外围硬件提供时钟的电路。有以下 2 种系统时钟和时钟振荡电路。

(1) 主系统时钟

高速内部振荡器（高速OCO）：能通过选项字节（000C2H）对 $F_{HOCO}=48\text{MHz}$ 进行分频。 F_{IH} 为 F_{HOCO} 的分频。在解除复位后，CPU一定以此高速内部振荡器时钟开始运行。能通过进入深度睡眠模式或者设置HIOSTOP位（CSC寄存器的bit0）使振荡停止。能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改选项字节设置的频率。

(2) 副系统时钟

XT1振荡电路：能通过给XT1引脚和XT2引脚连接32.768KHz的谐振器使 $F_{XT}=32.768\text{KHz}$ 的时钟振荡，并且能通过设置XTSTOP位（时钟运行状态控制寄存器（CSC）的bit6）使振荡停止。

另外，能由 EXCLKS/XT2/PA9 引脚提供低速外部时钟（ $F_{EXS}=32.768\text{kHz}$ ），并且能通过设置 XTSTOP 位将低速外部时钟的输入置为无效。

低速内部振荡器（低速OCO）：能使 $F_{IL}=32\text{KHz(TYP.)}$ 的时钟振荡。

只有以下外围硬件才能通过低速内部振荡器时钟运行：

- 独立看门狗定时器
- 15位间隔定时器

注： F_{HOCO} ：高速内部振荡器的时钟频率

F_{IH} ：高速内部振荡器分频后的时钟频率

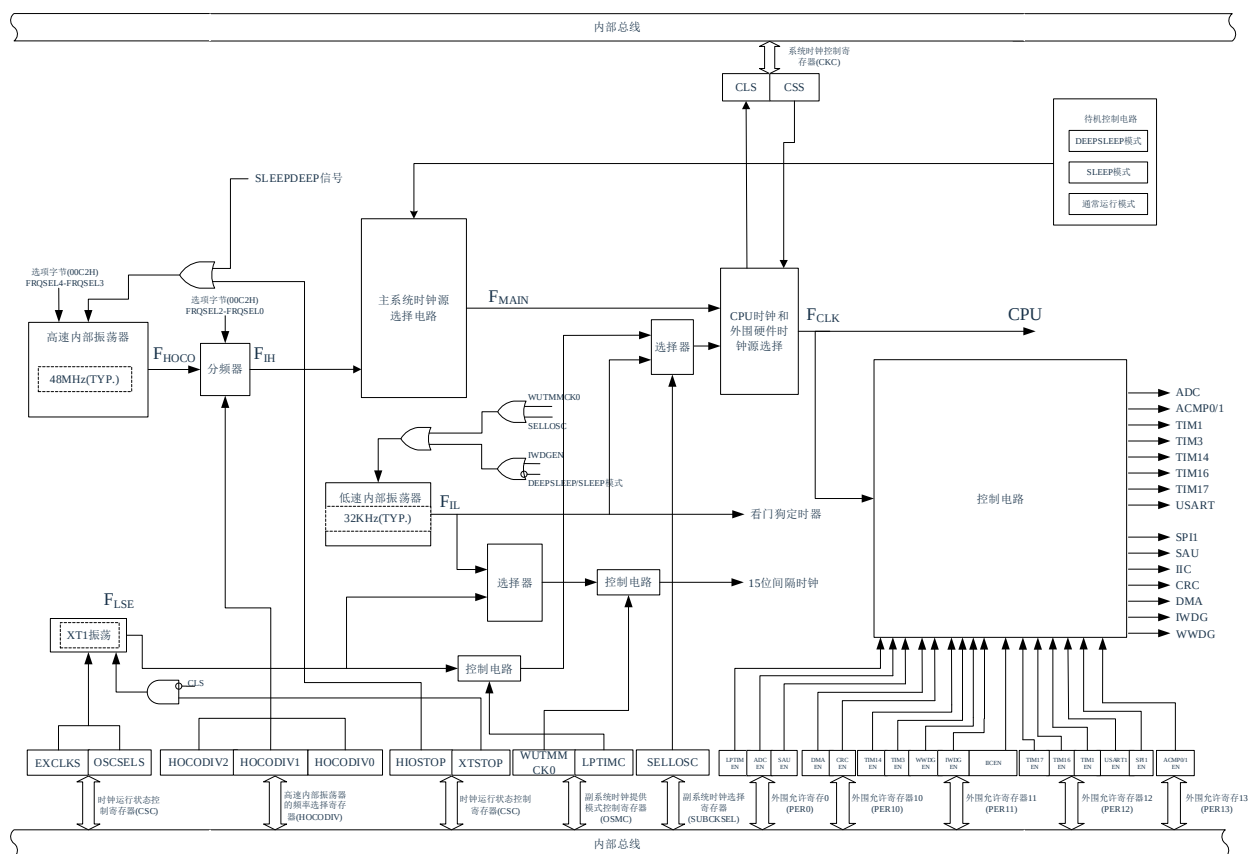
F_{IL} ：低速内部振荡器的时钟频率

时钟发生电路由以下硬件构成。

表 4-1 时钟发生电路的结构

项目	结构
控制寄存器	时钟运行模式控制寄存器（CMC） 系统时钟控制寄存器（CKC） 时钟运行状态控制寄存器（CSC） 外围允许寄存器（PER0、PER10、PER11、PER12、PER13） 高速内部振荡器的频率选择寄存器（HOCODIV） 低速内部振荡器时钟寄存器（SUBCKSEL）
振荡电路	高速内部振荡器 低速内部振荡器 XT1 振荡电路

图 4-1 时钟发生电路的框图



注: F_{HOCO} : 高速内部振荡器的时钟频率

F_{IH} : 高速内部振荡器分频后的时钟频率

F_{MAIN} : 主系统时钟频率

FCLK: CPU/外围硬件的时钟频率

F_{II} : 低速内部振荡器的时钟频率

FLSE: 低速外部振荡器的时钟频率

4.3 寄存器映射

(以下寄存器基地址 = 0x4002_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CMC	0x000	R/W	时钟运行模式控制寄存器	0x00
CSC	0x001	R/W	时钟运行状态控制寄存器	0xC0
CKC	0x004	R/W	系统时钟控制寄存器	0x00
SUBCKSEL	0x007	R/W	低速振荡器时钟选择寄存器	0x00
PER0	0x020	R/W	外围允许寄存器 0	0x00
OSMC	0x023	R/W	15 位间隔定时器运行时钟选择寄存器	0x00

(以下寄存器基地址 = 0x4002_0810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER10	0x000	R/W	外围允许寄存器 10	0x00
PER11	0x001	R/W	外围允许寄存器 11	0x00
PER12	0x002	R/W	外围允许寄存器 12	0x00
PER13	0x003	R/W	外围允许寄存器 13	0x00

(以下寄存器基地址 = 0x4002_1C00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
HOCODIV	0x024	R/W	高速内部振荡器的频率选择寄存器	选项字节 (000C2H) 的 FRQSEL[2:0]位的设置值

4.4 寄存器说明

4.4.1 系统时钟控制寄存器（CKC）

这是选择 CPU/外围硬件时钟和主系统时钟的寄存器。

通过 8 位存储器操作指令设置 CKC 寄存器。

位	符号	描述	复位值
7	CLS ^{注1}	CPU/外围硬件时钟(F _{CLK})的状态 0: 主系统时钟(F _{MAIN}) 1: 副系统时钟 (F _{SUB})	0
6	CSS	CPU/外围硬件时钟(F _{CLK})的选择 0: 主系统时钟(F _{MAIN}) 1: 副系统时钟 (F _{SUB})	0
5:0	--	保留	0x0

注1: bit7是只读位。

注2: 给CPU和外围硬件提供CSS位设置的时钟。如果更改CPU时钟，就同时更改外围硬件的时钟（15位间隔定时器时钟、独立看门狗定时器除外）。因此，如果要更改CPU/外围硬件的时钟，就必须停止各外围功能。

4.4.2 时钟运行状态控制寄存器（CSC）

这是控制高速内部振荡器时钟运行的寄存器。通过 8 位存储器操作指令设置 CSC 寄存器。
在产生复位信号后，此寄存器的值变为“C0H”。

位	符号	描述	复位值
7	--	保留	1
6	XTSTOP	XT1 振荡时钟运行控制 0: XT1 振荡电路运行 1: XT1 振荡电路停止	1
5:1	--	保留	0x0
0	HIOSTOP	高速内部振荡器时钟的运行控制 0: 高速内部振荡器运行 1: 高速内部振荡器停止	0

注意：

- 在解除复位后，必须在设置时钟运行模式控制寄存器（CMC）后设置 CSC 寄存器。
- 要通过设置 XSTOP 位开始 XT1 振荡时，必须通过软件等待外部低速时钟所需的振荡稳定时间。
- 不能通过 CSC 寄存器停止被选择为 CPU/外围硬件时钟（F_{CLK}）的时钟。
- 有关用于停止时钟振荡（外部时钟输入无效）的寄存器标志设置和停止前的条件，请参照表 4-2。

表 4-2 时钟停止方法

时钟	时钟停止前的条件（外部时钟输入无效）	设置 CSC 寄存器的标志
XT1 时钟	CPU/外围硬件时钟以副系统时钟以外的时钟运行。 (CLS=0)	XTSTOP=1
低速外部时钟		
高速内部振荡器时钟	CPU/外围硬件时钟以高速内部振荡器时钟以外的时钟运行。 (CLS=0 并且 MCS=1，或者 CLS=1)	HIOSTOP=1

4.4.3 时钟运行模式控制寄存器(CMC)

这是设置 XT1、XT2 引脚的运行模式的寄存器。

在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。能通过 8 位存储器操作指令读此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:4	--	保留	0
3:2	EXCLKS: OSCSELS	XT1 时钟引脚运行模式选择 00: XT1、XT2 端口用作端口模式 01: XT1 振荡模式，XT1/XT2 引脚连接晶体或者陶瓷谐振器 10: XT1、XT2 端口用作端口模式 11: 外部时钟输入模式，XT2 引脚用作外部时钟输入	0
1	--	保留	0x0
0	LSEDRV	XT1 驱动选择 0: 低驱动模式 1: 高驱动模式	0

注：EXCLKS,OSCSELS位，只在上电复位被初始化，而其他复位时保持不变。

注意：在解除复位后，只能通过8位存储器操作指令写1次CMC寄存器。当以初始值（“00H”）使用CMC寄存器时，为了防止程序失控时的误动作（如果误写“00H”以外的值就不能恢复），必须在解除复位后将CMC寄存器置“00H”。

4.4.4 外围允许寄存器（PER0、PER10、PER11、PER12、PER13）

这是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。当使用由这些寄存器控制的以下外围功能时，必须在进行外围功能的初始设置前将对应位置“1”。寄存器功能汇总：

PER0	15 位间隔定时器，ADC，SAU
PER10	DMA，CRC
PER11	TIM14，TIM3，WWDG，IWDG，IIC
PER12	TIM17，TIM16，TIM1，USART，SPI1
PER13	ACMP

通过 8 位存储器操作指令设置 PER0 寄存器和 PER10、PER11、PER12、PER13 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

外围允许寄存器 0（PER0）

位	符号	描述	复位值
7	LPTIMEN	提供 15 位间隔定时器输入时钟的控制 0: 停止提供输入时钟，不能写 15 位间隔定时器使用的 SFR。 1: 提供输入时钟，能读写 15 位间隔定时器使用的 SFR。	0
6	-	保留	0
5	ADCEN	提供 ADC 模块的输入时钟的控制 0: 停止提供输入时钟，ADC 不能运行 1: 提供输入时钟，ADC 能运行	0
4	-	保留	0
3	SAUEN	提供串行通信单元输入时钟的控制 0: 停止提供输入时钟，不能写 SAU 使用的 SFR。 1: 提供输入时钟，能读写 SAU 使用的 SFR。	0
2:0	-	保留	0

外围允许寄存器 10（PER10）

位	符号	描述	复位值
7:4	-	保留	0
3	DMAEN	提供 DMA 模块输入时钟的控制 0: 停止提供输入时钟，DMA 不能运行 1: 提供输入时钟，DMA 能运行	0
2	CRCEN	提供 CRC 模块输入时钟的控制 0: 停止提供输入时钟，CRC 不能运行 1: 提供输入时钟，CRC 能运行	0
1:0	-	保留	0

外围允许寄存器 11（PER11）

位	符号	描述	复位值
7	TIM14EN	提供 TIM14 模块输入时钟的控制 0: 停止提供输入时钟, TIM14 模块不能运行 1: 提供输入时钟, TIM14 模块能运行	0
6	TIM3EN	提供 TIM3 模块的输入时钟的控制 0: 停止提供输入时钟, TIM3 不能运行 1: 提供输入时钟, TIM3 能运行	0
5	WWDGEN	提供 WWDG 模块的输入时钟的控制 0: 停止提供输入时钟, WWDG 不能运行 1: 提供输入时钟, WWDG 能运行	0
4	IWDGEN	提供 IWDG 模块的输入时钟的控制 0: 停止提供输入时钟, IWDG 不能运行 1: 提供输入时钟, IWDG 能运行	0
3:1	-	保留	0
0	IICEN	提供 IIC 模块输入时钟的控制 0: 停止提供输入时钟, IIC 不能运行 1: 提供输入时钟, IIC 能运行	0

外围允许寄存器 12 (PER12)

位	符号	描述	复位值
7:5	--	保留	0
4	TIM17EN	提供 TIM17 模块的输入时钟的控制 0: 停止提供输入时钟, TIM17 不能运行 1: 提供输入时钟, TIM17 能运行	0
3	TIM16EN	提供 TIM16 模块的输入时钟的控制 0: 停止提供输入时钟, TIM16 不能运行 1: 提供输入时钟, TIM16 能运行	0
2	TIM1EN	提供 TIM1 模块输入时钟的控制 0: 停止提供输入时钟, TIM1 不能运行 1: 提供输入时钟, TIM1 能运行	0
1	USART1EN	提供 USART 模块输入时钟的控制 0: 停止提供输入时钟, USART 不能运行 1: 提供输入时钟, USART 能运行	0
0	SPI1EN	提供 SPI1 模块输入时钟的控制 0: 停止提供输入时钟, SPI1 不能运行 1: 提供输入时钟, SPI1 能运行	0

外围允许寄存器 13 (PER13)

位	符号	描述	复位值
7:2	-	保留	0
1	ACMPEN	提供 ACMP 模块输入时钟的控制 0: 停止提供输入时钟，ACMP 不能运行 1: 提供输入时钟，ACMP 能运行	0
0	-	保留	0

4.4.5 15 位间隔定时器运行时钟选择寄存器（OSMC）

通过 OSMC 寄存器选择 15 位间隔定时器运行时钟。

通过 8 位存储器操作指令设置 OSMC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	LPTIMC	深度睡眠模式和CPU以外低速时钟运行的睡眠模式中的设置 0: 允许给外围功能提供外部低速时钟 1: 停止给15位间隔定时器以外的外围功能提供外部低速时钟。	0
6:5	--	保留	0x0
4	WUTMMCK0	15 位间隔定时器的运行时钟的选择: 0: 低速振荡时钟停止给 15 位间隔定时器提供时钟 1: 低速振荡时钟为 15 位间隔定时提供时钟	0
3:0	--	保留	0x0

4.4.6 高速内部振荡器的频率选择寄存器（HOCODIV）

这是更改选项字节（000C2H）设置的高速内部振荡器频率的寄存器。通过 8 位存储器操作指令设置 HOCODIV 寄存器。

在产生复位信号后，此寄存器的值变为选项字节（000C2H）的 FRQSEL[2:0]位的设置值。

位	符号	描述	复位值
7:3	--	保留	0
2:0	HOCODIV[2:0]	高速内部振荡器时钟频率的选择 000: $F_{IH}=48\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 001: $F_{IH}=24\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 010: $F_{IH}=12\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 011: $F_{IH}=6\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 100: $F_{IH}=3\text{MHz}$ $F_{HOCO}=48\text{MHz}$ 101: $F_{IH}=1.5\text{MHz}$ $F_{HOCO}=48\text{MHz}$	选项字节（000C2H） 的 FRQSEL[2:0]位的 设置值

注 1: 必须在选择高速内部振荡器时钟（ F_{IH} ）作为 CPU/外围硬件时钟（ F_{CLK} ）的状态下设置 HOCODIV 寄存器。

注 2: 在通过 HOCODIV 寄存器更改频率后，经过以下转移时间之后进行频率切换：

- 以更改前的频率，最多进行 3 个时钟的运行。
- 以更改后的频率，最多等待 3 个 CPU/外围硬件的时钟。

4.4.7 低速振荡器时钟选择寄存器（SUBCKSEL）

SUBCKSEL 寄存器是选择副系统时钟 F_{SUB} 的寄存器。

通过 8 位存储器操作指令设置 SUBCKSEL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	--	保留	0x0
0	SELLOCO	低速振荡器时钟选择 0: 选择低速外部振荡器时钟 1: 选择低速内部振荡器时钟	0

4.5 系统时钟振荡电路

4.5.1 XT1 振荡电路

XT1 振荡电路通过连接 XT1 引脚和 XT2 引脚的晶体谐振器 (32.768KHz(TYP.)) 进行振荡。当使用 XT1 振荡电路时, 必须将时钟运行模式控制寄存器 (CMC) 的 bit2 (OSCSELS) 置“1”, 也能输入外部时钟, 此时必须给 EXCLKS 引脚输入时钟信号。

当使用 XT1 振荡电路时, 必须对时钟运行模式控制寄存器 (CMC) 的 bit3 和 bit2 (EXCLKS、OSCSELS) 进行以下的设置:

晶体振荡 : EXCLKS、OSCSELS=0、1

外部时钟输入 : EXCLKS、OSCSELS=1、1

当不使用 XT1 振荡电路时, 必须设置为输入端口模式 (EXCLKS、OSCSELS=0、0)。XT1 振荡电路的外接电路例子如图 4-2 所示。

图 4-2 XT1 振荡电路的外接电路例子



注意: XT1 振荡电路时, 为了避免布线电容等的影响, 必须通过以下方法对图 4-2 中的虚线部分进行布线:

- 必须尽量缩短布线。
- 不能和其他的信号线交叉, 并且不能接近有变化的大电流流过的布线。
- 必须始终保持振荡电路的电容器接地点和 VSS 同电位, 而且不能给大电流流过的接地图形接地。
- 不能从振荡电路取出信号。

4.5.2 高速内部振荡器

CMS32C020 内置高速内部振荡器。能通过选项字节 (000C2H) 从 $F_{IH}=48\text{MHz}$ 、24MHz、12MHz、6MHz、3MHz、1.5MHz 中选择频率。能通过时钟运行状态控制寄存器(CSC)的 bit0(HIOSTOP)控制振荡。

在解除上电复位后, 高速内部振荡器自动开始振荡。

4.5.3 低速内部振荡器

低速内部振荡器时钟用作独立看门狗定时器、15 位间隔定时器以及 SysTick 定时器的外部参考时钟，也可用作 CPU 时钟和外围模块时钟。

15 位间隔定时器运行时钟选择寄存器 (OSMC) 的 bit4 (WUTMMCK0) 和 15 位间隔定时器的计数时钟控制寄存器 (LPTIMCL) 的 bit1(CKSEL) 同时为“1”时，或者副系统时钟选择寄存器 (SUBCKSEL) 的 bit0(SELLOCO)为“1”时，低速内部振荡器振荡。

4.6 时钟发生电路的运行

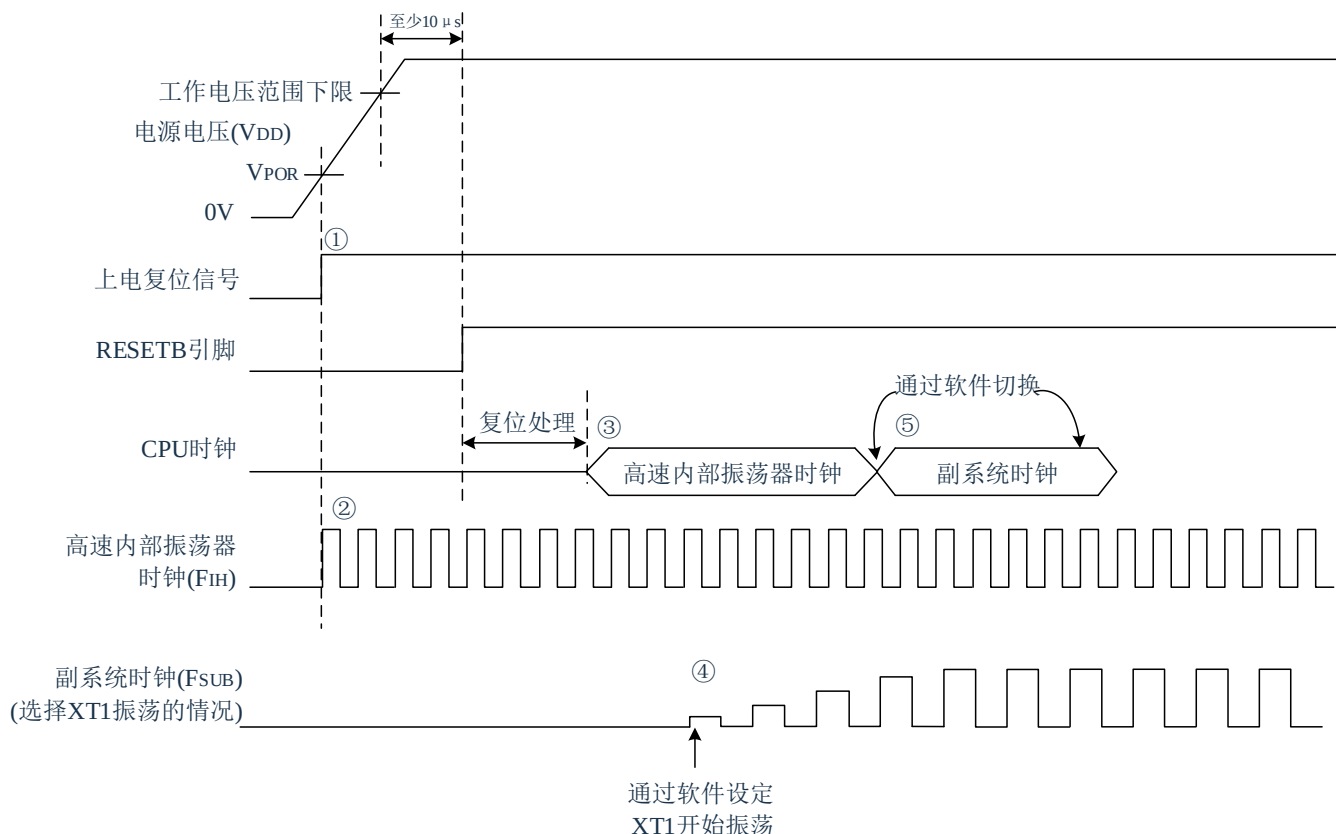
时钟发生电路产生以下所示各种时钟，并且控制待机模式等 CPU 的运行模式（参照图 4-1）。

F_{IH} : 高速内部振荡器时钟

F_{SUB} : 副系统时钟

CMS32C020 在解除复位后，CPU 通过高速内部振荡器的输出开始运行。接通电源时的时钟发生电路的运行如 4-3 所示。

图 4-3 接通电源时的时钟发生电路的运行



- (1) 在接通电源后，通过上电复位（POR）电路产生内部复位信号。但是，在达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态（上图是使用外部复位时的例子）。
- (2) 如果解除复位，高速内部振荡器就自动开始振荡。
- (3) 在解除复位后，进行电压稳定等待和复位处理，然后 CPU 以高速内部振荡器时钟开始运行。
- (4) 必须通过软件设置 XT1 时钟的开始振荡（参照 4.7.2 XT1 振荡电路的设置例子”）。
- (5) 如果要将 CPU 时钟切换到 XT1 时钟，就必须在等待时钟振荡稳定后通过软件设置切换（4.7.2 XT1 振荡电路的设置例子”）。

4.7 时钟控制

4.7.1 高速内部振荡器的设置例子

在解除复位后，CPU/外围硬件时钟（ F_{CLK} ）一定以高速内部振荡器时钟运行。能通过选项字节（000C2H）的 FRQSEL0~FRQSEL4 位，从 48MHz、24MHz、12MHz、6MHz、3MHz、1.5MHz 中选择高速内部振荡器的频率。另外，能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改频率。

选项字节 000C2 的设置

位	符号	描述			复位值
7:5	--	保留			--
4:0	FRQSEL[4:0]	高速内部振荡器时钟频率的选择			--
		FRQSEL[4:0]的值	F _{HOCO}	F _{IH}	
		00000	48MHz	48MHz	
		00001	48MHz	24MHz	
		00010	48MHz	12MHz	
		00011	48MHz	6MHz	
		00100	48MHz	3MHz	
		00101	48MHz	1.5MHz	
		上述以外	禁止设定		

【高速内部振荡器的频率选择寄存器（HOCODIV）的设置】

位	符号	描述	复位值
7:3	--	保留	0
2:0	HOCODIV[2:0]	高速内部振荡器时钟频率的选择 000: F_{IH} = 48MHz F_{HOCO} = 48MHz 001: F_{IH} = 24MHz F_{HOCO} = 48MHz 010: F_{IH} = 12MHz F_{HOCO} = 48MHz 011: F_{IH} = 6MHz F_{HOCO} = 48MHz 100: F_{IH} = 3MHz F_{HOCO} = 48MHz 101: F_{IH} = 1.5MHz F_{HOCO} = 48MHz	选项字节（000C2H）的 FRQSEL[2:0]位的设置值

4.7.2 XT1 振荡电路的设置例子

在解除复位后，CPU/外围硬件时钟（F_{CLK}）一定以高速内部振荡器时钟运行。此后，如果改为 XT1 振荡时钟，就通过副系统时钟提供模式控制寄存器（OSMC）、时钟运行模式控制寄存器（CMC）和时钟运行状态控制寄存器（CSC）进行振荡电路的设置和振荡开始的控制，并且通过系统时钟控制寄存器（CKC）将 XT1 振荡时钟设置为 F_{CLK}。

【寄存器的设置】必须按照①～⑤的顺序设置寄存器。

- ① 在深度睡眠模式或者 CPU 以外部低速时钟运行的睡眠模式中，当只要 15 位间隔定时器以外部低速时钟运行时，必须将 LPTIMC 位置“1”。

bit	7	6	5	4	3	2	1	0
OSMC	LPTIMC	0	0	WUTMMCK0	0	0	0	0

- ② 将 CMC 寄存器的 OSCSELS 位置“1”，使 XT1 振荡电路运行。

	7	6	5	4	3	2	1	0
CMC	0	0	0	0	EXCLKS	OSCSELS	0	LSEDRV

- ③ 将 CSC 寄存器的 XTSTOP 位清“0”，使 XT1 振荡电路开始振荡。

	7	6	5	4	3	2	1	0
CSC	1	XTSTOP	0	0	0	0	0	HIOSTOP

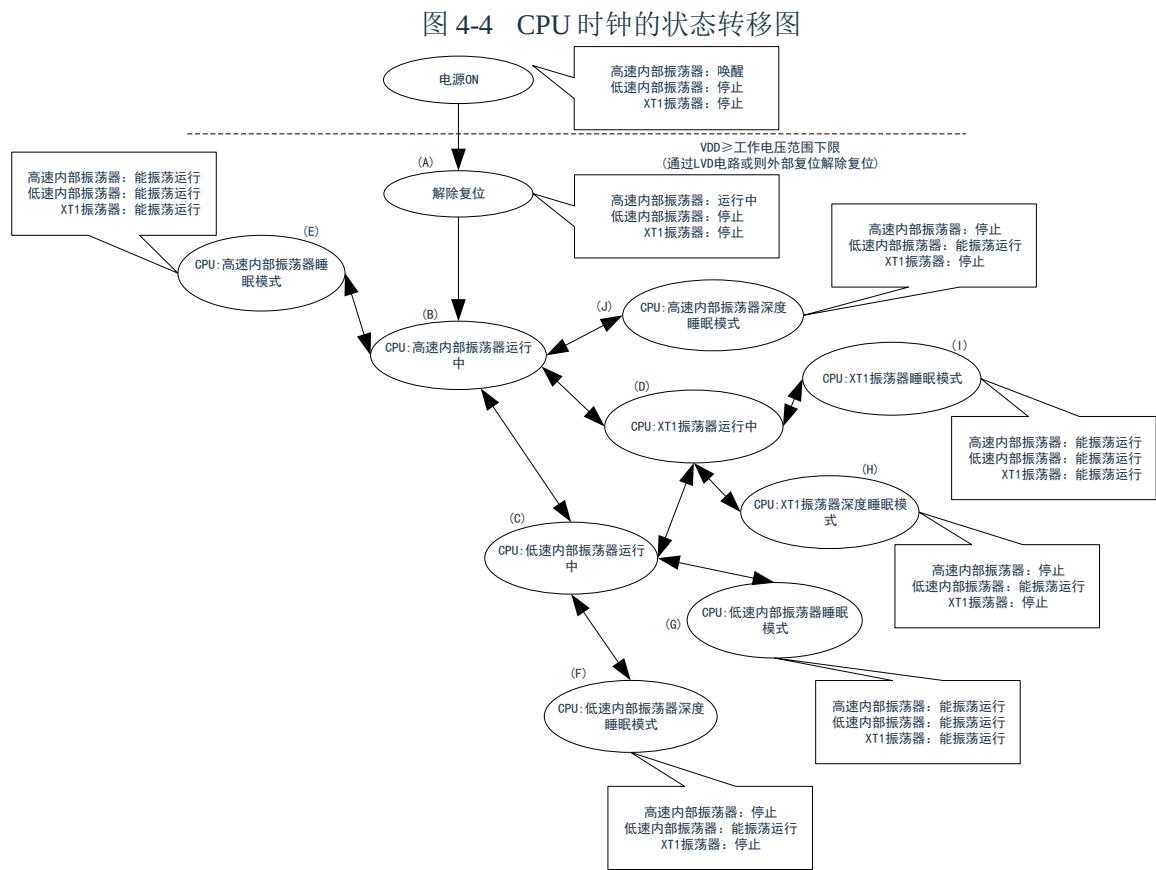
- ④ 必须通过软件和定时器功能等，等待外部低速时钟所需的振荡稳定时间。

- ⑤ 通过 CKC 寄存器的 CSS 位将 XT1 振荡时钟设置为 CPU/外围硬件时钟。

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	0	0	0	0	0	0

4.7.3 CPU 时钟的状态转移图

本产品的 CPU 时钟状态转移图如图 4-4 所示。



CPU 时钟的转移和 SFR 寄存器的设置例子等如表 4-3 所示。

表 4-3 CPU 时钟的转移和 SFR 寄存器的设置例子(1/3)

(1) 在解除复位(A)后，CPU 转移到高速内部振荡器时钟运行(B)。

状态转移	SFR 寄存器的设置
(A)→(B)	不需要设置 SFR 寄存器（解除复位后的初始状态）。

(2) 在解除复位(A)后，CPU 转移到低速内部振荡时钟运行(C)。

状态转移	SFR 寄存器的设置
(B)→(C) (D)→(C)	寄存器设置顺序:
	① SUBCKSEL 寄存器 SELLOC0 为 1
	② 配置 CKC 寄存器的 CSS 为 1，选择低速内部振荡时钟为 CPU 和外围电路的时钟
	③ 判断 CKC 寄存器的 CLS，等待该位为 1

(3) 在解除复位(A)后，CPU 转移到低速外部振荡时钟运行(D)。

状态转移	SFR 寄存器的设置
(B)→(D) (C)→(D)	寄存器设置顺序:
	①SUBCKSEL 寄存器 SELLOC0 为 0
	②配置 CKC 寄存器的 CSS 为 1，选择低速外部振荡时钟为 CPU 和外围电路的时钟
	③判断 CKC 寄存器的 CLS，等待该位为 1

(4) CPU 由其他时钟转移到高速内部振荡时钟运行(B)。

状态转移	SFR 寄存器的设置
(C)→(B) (D)→(B)	寄存器设置顺序:
	配置 CKC 寄存器的 CSS 为 0，选择主时钟作为 CPU 和外围电路的时钟
	判断 CKC 寄存器的 CLS，等待该位为 0

注 1：表 4-3 的(A)~(J)对应图 4-4 的(A)~(J)。

注 2：高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

表 4-3 CPU 时钟的转移和 SFR 寄存器的设置例子(2/3)

(5) CPU 从高速内部时钟运行(B)转移到睡眠模式(E)。

CPU 从低速内部时钟运行(C)转移到睡眠模式(G)。

CPU 从低速外部时钟运行(D)转移到睡眠模式(I)。

状态转移	设置内容
(B)→(E) (C)→(G) (D)→(I)	执行 WFI 指令。

注：表 4-3 的(A)~(J)对应图 4-4 的(A)~(J)。

表 4-3 CPU 时钟的转移和 SFR 寄存器的设置例子(3/3)

(6) CPU 从高速内部时钟运行(B)转移到深度睡眠模式(J)。

CPU 从低速内部时钟运行(C)转移到深度睡眠模式(F)。

CPU 从低速外部时钟运行(D)转移到深度睡眠模式(H)。

状态转移	设置内容	
(B)→(J) (C)→(F) (D)→(H)	停止不能在深度睡眠模式中运行的外围功能。	SCR 寄存器 bit2 (SLEEPDEEP) 置为 1，并执行 WFI 指令。

注：表 4-3 的(A)~(J)对应图 4-4 的(A)~(J)。

4.7.4 CPU 时钟转移前的条件和转移后的处理

CPU 时钟转移前的条件和转移后的处理如下所示。

表 4-4 有关 CPU 时钟的转移

CPU 时钟		转移前的条件	转移后的处理
转移前	转移后		
高速内部振荡时钟	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	如果停止高速内部振荡器的振荡（HIOSTOP=1），就能减小工作电流。
	XT1 振荡时钟	XT1 振荡稳定 • OSCSELS=1,EXCLKS=0,XTSTOP=0 • 经过振荡稳定时间后	
低速内部振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡 • HIOSTOP=0 • 经过振荡稳定时间后	若独立看门狗和 LPTIM 没有运行，可关闭低速内部时钟振荡(SELLOCO=0)就能减小工作电流。
	XT1 振荡时钟	XT1 振荡稳定 • OSCSELS=1,EXCLKS=0,XTSTOP=0 • 经过振荡稳定时间后	
XT1 振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡 • HIOSTOP=0 • 经过振荡稳定时间后	能停止 XT1 的振荡(XTSTOP=1)
	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	

4.7.5 CPU 时钟和主系统时钟的切换所需时间

能通过设置系统时钟控制寄存器(CKC)的 bit6 (CSS)进行 CPU 时钟的切换(主系统时钟和副系统时钟，即高速内部振荡时钟和低速内部振荡时钟及 XT1 振荡时钟)

在改写 CKC 寄存器后不立即进行实际的切换，而是在更改 CKC 寄存器后仍然以切换前的时钟继续运行数个时钟(参照表 4-5)。

能通过 CKC 寄存器的 bit7(CLS)、来判断 CPU 的运行时钟。

如果切换 CPU 时钟，就同时切换外围硬件时钟。

表 4-5 时钟切换所需要时钟数

切换后时钟 \ 切换前时钟	高速内部振荡器时钟(F _{IH})	低速内部振荡器时钟(F _{IL})	XT1 振荡时钟
高速内部振荡器时钟(F _{IH})		2.5 个 F _{IH} / F _{IL} 时钟	2.5 个 F _{IH} / F _{XT} 时钟
低速内部振荡器时钟(F _{IL})	2 个时钟		2 个时钟
XT1 振荡时钟	2 个时钟	2 个时钟	

注 1：表 4-5 中的时钟数是切换前的 CPU 时钟数。

注 2：表 4-5 中的时钟数是舍入小数部分的时钟数。

例：CPU 从主系统时钟切换到副系统时钟的情况（选择 F_{MAIN}=F_{IH}=1.5MHz、F_{LSE}= 32.768KHz 振荡的情况）

$$2.5F_{MAIN}/F_{LSE}=2.5(1500/32.768)=144.4\approx115 \text{ 个时钟}$$

4.7.6 时钟振荡停止前的条件

用于停止时钟振荡的寄存器标志设置和停止前的条件如下所示。

表 4-6 时钟振荡停止前的条件和标志设置

时钟	时钟停止前的条件	SFR 寄存器的标志
高速内部振荡器时钟	CLS=1 (CPU 以高速内部振荡器时钟以外的时钟运行)	HIOSTOP=1
低速内部振荡器时钟	CLS=0, WUMMCK0=0 (CPU 以低速内部振荡器时钟以外的时钟运行)	SELLOCO=0
XT1 振荡时钟	CLS=0 (CPU 低速外部时钟以外的时钟运行)	XTSTOP=1

第5章 15 位间隔定时器

5.1 15 位间隔定时器的功能

以事先设置的任意时间间隔产生中断，能用于从深度睡眠模式的唤醒。

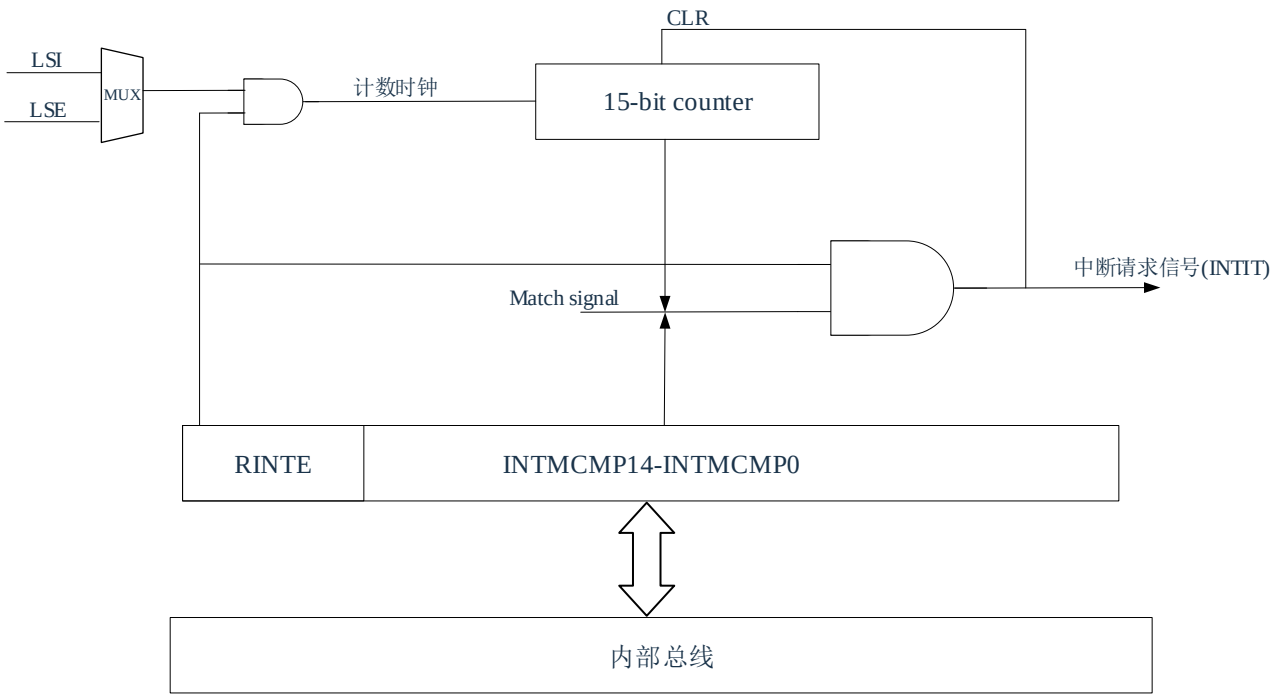
5.2 15 位间隔定时器的结构

15 位间隔定时器由以下硬件构成。

表 5-1 15 位间隔定时器的结构

项目	结构
计数器	15 位计数器
控制寄存器	外围允许寄存器 0（PER0）
	15 位间隔定时器的计数时钟控制寄存器（LPTIMCL）
	15 位间隔定时器的控制寄存器（CON）

图 5-1 15 位间隔定时器的框图



5.3 寄存器映射

(以下寄存器基地址 = 0x4002_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER0	0x020	R/W	外围允许寄存器 0	0x00

(以下寄存器基地址 = 0x4004_4B50) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CON	0x000	R/W	15 位间隔定时器控制寄存器	0x7FFF

(以下寄存器基地址 = 0x4004_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LPTIMCL	0x00C	R/W	15 位间隔定时器的计数时钟控制寄存器	0x00

5.4 寄存器说明

5.4.1 外围允许寄存器 0（PER0）

PER0 寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用 15 位间隔定时器时，必须将 bit7（LPTIMEN）置“1”。通过 8 位存储器操作指令设置 PER0 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	LPTIMEN	提供 15 位间隔定时器输入时钟的控制 0: 停止提供输入时钟，不能写 15 位间隔定时器使用的 SFR 1: 提供输入时钟，能读写 15 位间隔定时器使用的 SFR	0
6	-	保留	0
5	ADCEN	提供 ADC 模块的输入时钟的控制 0: 停止提供输入时钟，ADC 不能运行 1: 提供输入时钟，ADC 能运行	0
4	-	保留	0
3	SAUEN	提供串行通信单元输入时钟的控制 0: 停止提供输入时钟，不能写 SAU 使用的 SFR 1: 提供输入时钟，能读写 SAU 使用的 SFR	0
2:0	-	保留	0

5.4.2 15 位间隔定时器的控制寄存器（CON）

这是设置 15 位间隔定时器的运行开始和停止以及比较值的寄存器。通过 16 位存储器操作指令设置 CON 寄存器。

在产生复位信号后，此寄存器的值变为“7FFFH”。

位	符号	描述	复位值
15	RINTE	15 位间隔定时器的运行控制 0: 停止计数器的运行（清除计数） 1: 开始计数器的运行	0x0
14:0	ITCMP	15 位间隔定时器比较值的设定	0x7FFF

ITCMP14~ITCMP0	15 位间隔定时器比较值的设置
0001H	这些位产生“计数时钟周期 $\times$ （ITCMP 设置值+1）”的固定周期中断。
• • •	
7FFFH	
0000H	禁止设置。
ITCMP14~ITCMP0 为“0001H”或者“FFFH”时的中断周期例子	
•ITCMP14~ITCMP0=0001H，计数时钟：F _{LSE} =32.768KHz $1/32.768[\text{KHz}] \times (1+1) = 0.06103515625[\text{ms}] \approx 61.03[\mu\text{s}]$	
•ITCMP15~ITCMP0=7FFFH，计数时钟：F _{LSE} =32.768KHz $1/32.768[\text{KHz}] \times (32767+1) = 1[\text{S}]$	

注意：

1. 要将 RINTE 位从“1”改为“0”时，必须在通过中断屏蔽标志寄存器将 LPTIM 设置为禁止中断处理后进行改写。要重新开始运行（从“0”改为“1”）时，必须在清除中断标志后设置为允许中断处理。
2. RINTE 位的读取值在设置 RINTE 位后的 1 个计数时钟之后被反映。
3. 在从睡眠模式转移到通常运行模式后，如果要设置 CON 寄存器并且再次转移到睡眠模式，就必须在确认 CON 寄存器的写入值被反映后或者在设置 CON 寄存器后至少经过 1 个计数时钟之后再转移到睡眠模式。
4. 要更改 ITCMP14~ITCMP0 位的设置时，必须在 RINTE 位为“0”的状态下进行。

但是，能在将 RINTE 位从“0”改为“1”或者从“1”改为“0”的同时更改 ITCMP14~ITCMP0 位的设置。

5.4.3 15 位间隔定时器的计数时钟控制寄存器（LPTIMCL）

位	符号	描述	复位值
7:1	--	保留	0
0	CKSEL	15 位间隔定时器的计数时钟的时钟源的选择： 0: 选择低速外部振荡器时钟(LSE) 1: 选择低速内部振荡器时钟(LSI)	0

5.5 15 位间隔定时器的运行

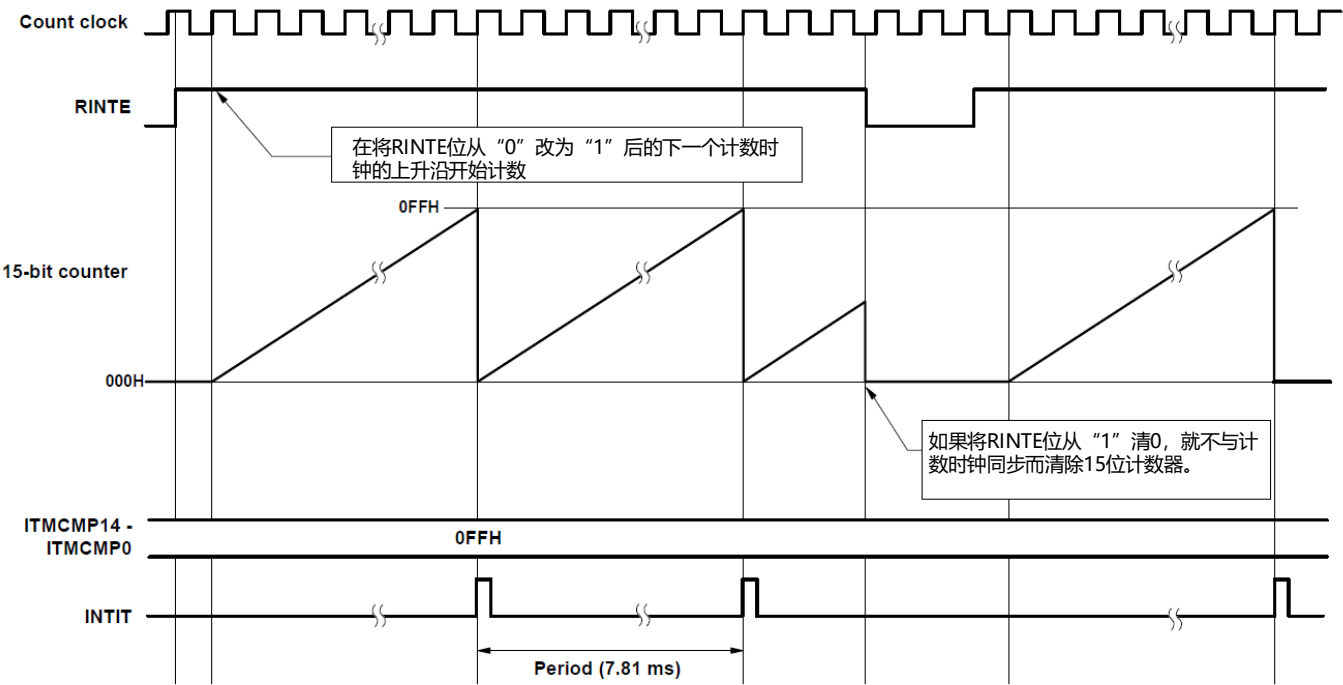
5.5.1 15 位间隔定时器的运行时序

以 ITCMP14~ITCMP0 位设置的计数值为间隔，作为重复产生中断请求的 15 位间隔定时器运行。如果将 RINTE 位置“1”，15 位计数器就开始计数。

当 15 位计数值和 ITCMP14~ITCMP0 位的设置值相同时，将 15 位计数值清“0”并且继续计数，同时产生中断请求信号。

15 位间隔定时器的基本运行如图 5-2 所示。

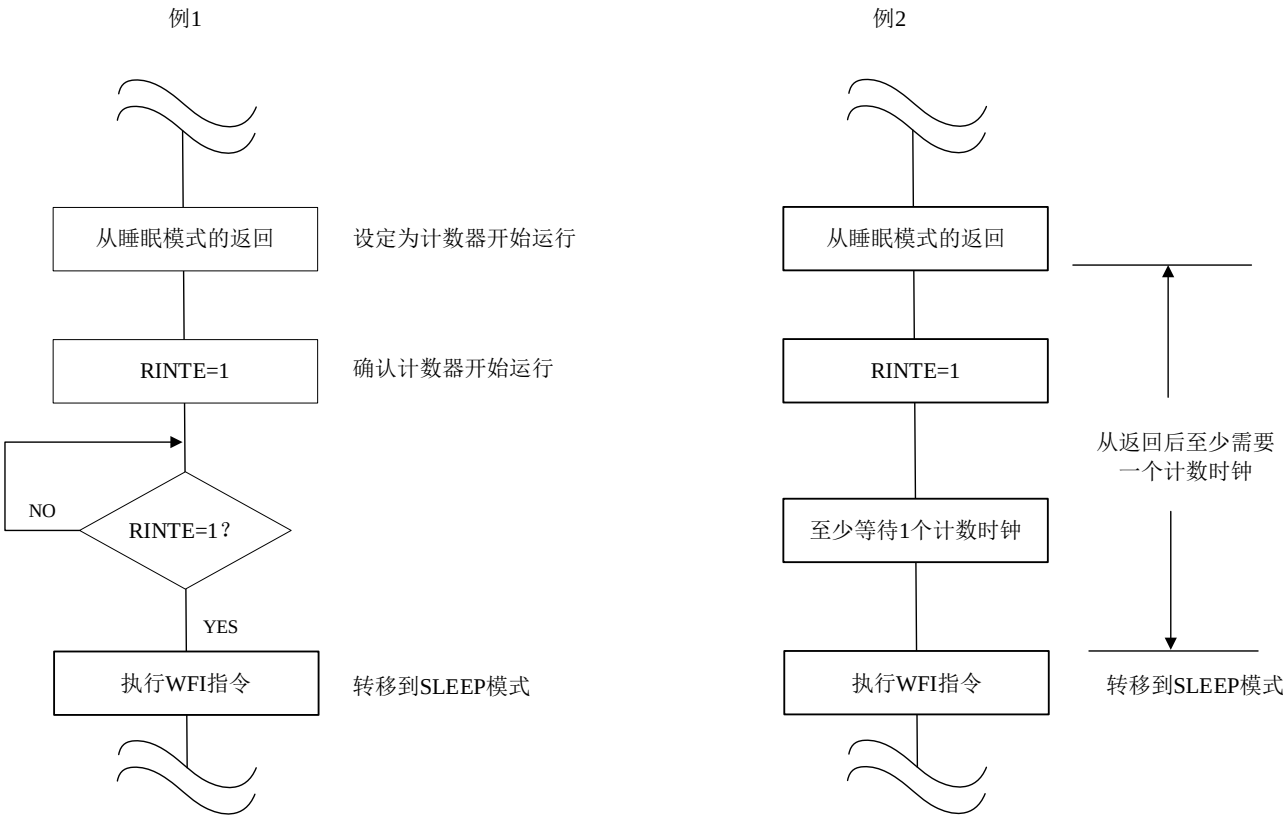
图 5-2 15 位间隔定时器的运行时序
(ITCMP14~ITCMP0=0FFH, 计数时钟: $F_{LSE}=32.768\text{KHz}$)



5.5.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移

在从睡眠模式返回后，如果要将 RINTE 位置“1”并且再次转移到睡眠模式，就必须在将 RINTE 位置“1”后确认 RINTE 位的写入值被反映，或者在返回后至少经过 1 个计数时钟的时间之后再转移到睡眠模式。

- 在将 RINTE 位置“1”后，通过轮询确认 RINTE 位变为“1”，然后转移到睡眠模式（参照下图的例 1）。
- 在将 RINTE 位置“1”后至少经过 1 个计数时钟的时间之后转移到睡眠模式（参照下图的例 2）。



第6章 高级控制定时器 TIM1

6.1 TIM1 介绍

高级控制定时器 TIM1 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。此定时器可用于各种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

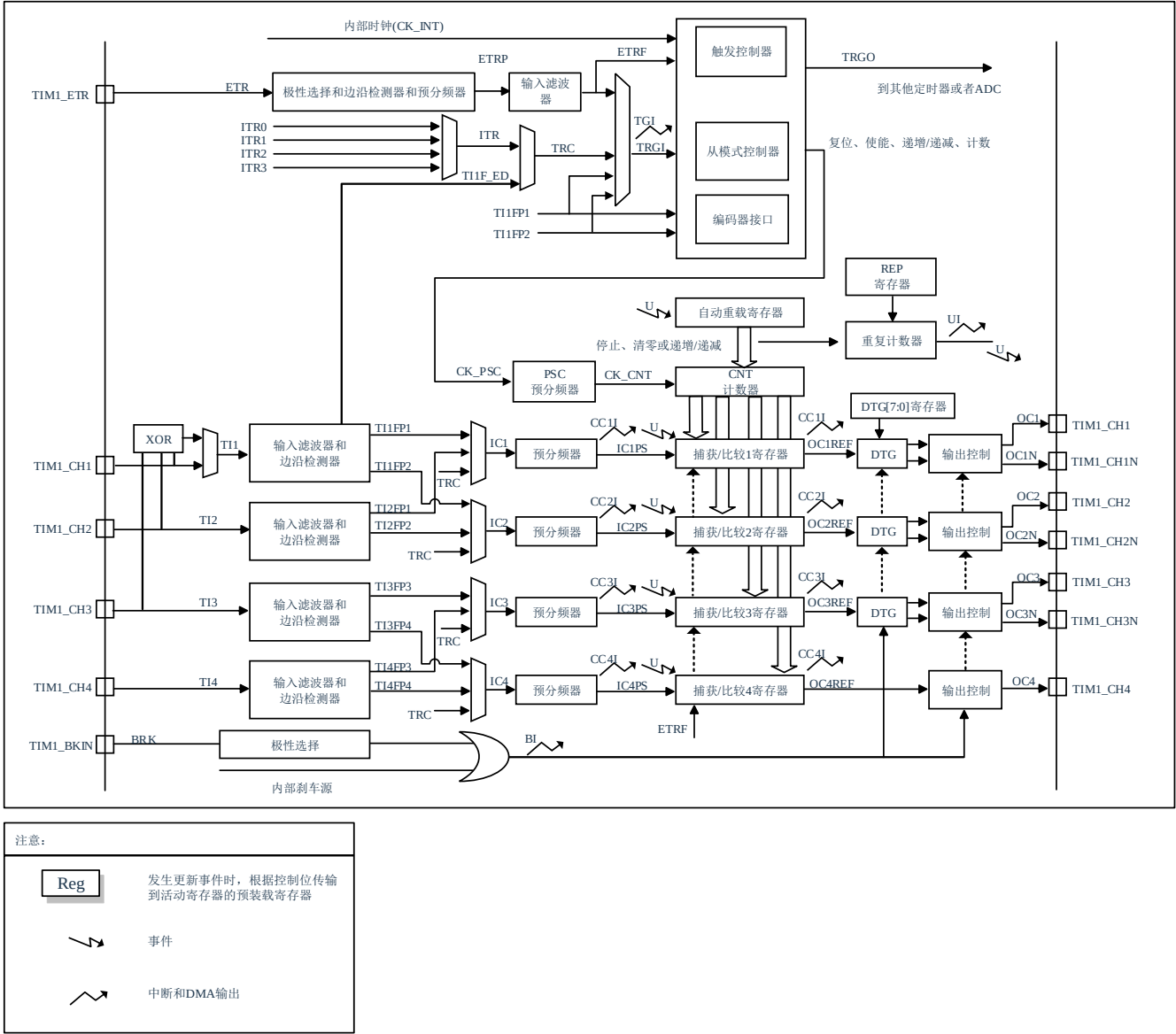
高级控制定时器（TIM1）和通用（TIMx）定时器是完全独立的，不共享任何资源。它们可以同步起来。

6.2 TIM1 主要特性

TIM1 定时器具有以下特性：

- 16 位向上、向下、向上向下自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 到 65536 之间。
- 多达 4 个独立通道，可用于：
 - 输入捕获
 - 输出比较
 - PWM 生成（边沿和中心对齐模式）
 - 单脉冲模式输出
- 带可编程死区的互补输出。
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 重复计数器，在计数指定周期数后，才更新时间寄存器
- 刹车输入可以将定时器的输出信号置为复位状态和已知状态
- 发生如下事件时生成中断/DMA 请求：
 - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
 - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
 - 输入捕获
 - 输出比较
 - 刹车输入
- 支持增量（正交）编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

图 6-1 TIM1 结构框图



6.3 TIM1 功能说明

6.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器向上计数、向下计数或向上向下双向计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIM1_CNT)
- 预分频器寄存器 (TIM1_PSC)
- 自动重载寄存器 (TIM1_ARR)
- 重复计数器寄存器 (TIM1_RCR)

自动装载寄存器是预先装载的，写或者读自动重载寄存器将访问预装载寄存器。根据在 TIM1_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到上溢出 (向下计数器时的下溢条件) 并当 TIM1_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIM1_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。

注意，在设置了 TIM1_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

6.3.1.1 预分频器说明

预分频器可以将计数器的时钟按 1 到 65535 之间的任意值分频。它是基于一个（在 TIM1_PSC 寄存器中的）16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 6-2 预分频器分频由 1 变为 2 的计数器时序图

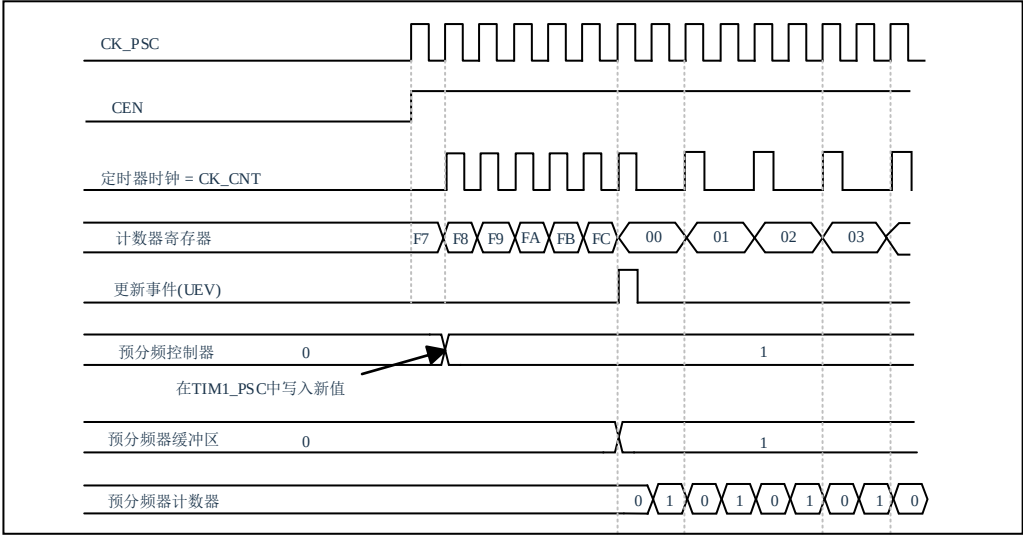
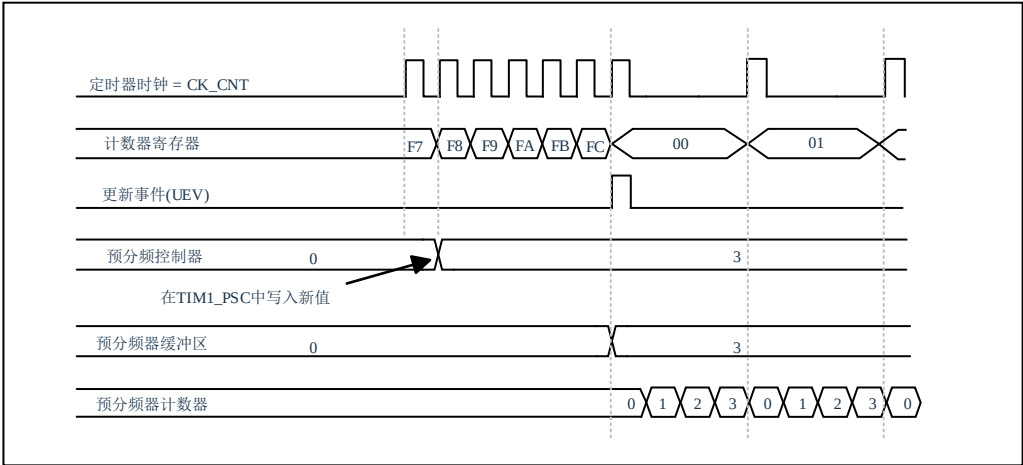


图 6-3 预分频器分频由 1 变为 4 的计数器时序图



6.3.2 计数器模式

6.3.2.1 向上计数模式

向上计数模式，是从 0 到自动装载值的计数器，然后又从 0 重新开始计数，并产生一个计数的溢出事件。如果重复计数器被使用，则在向上计数器重复几次（对重复计数器可编程）后，产生更新事件。否则，在每个计数溢出时，产生更新事件。

在 TIM1_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位也同样可以产生一个更新事件。设置 TIM1_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样也可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清零之前，将不产生更新事件。即使这样，在应该产生更新事件时，计数器仍会被清‘0’，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIM1_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时(依据 URS 位)设置更新标志位(TIM1_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIM1_RCR 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值(TIM1_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值(TIM1_PSC 寄存器的内容)。

以下各图以一些示例说明当 TIM1_ARR = 0x36 时不同时钟频率下计数器的行为。

图 6-4 计数器时序图，1 分频内部时钟

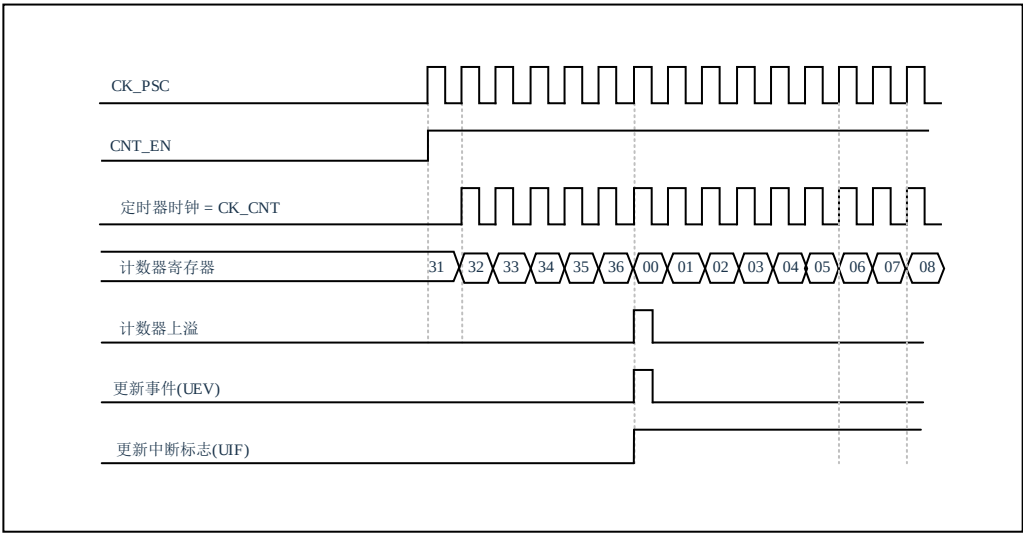


图 6-5 计数器时序图，2 分频内部时钟

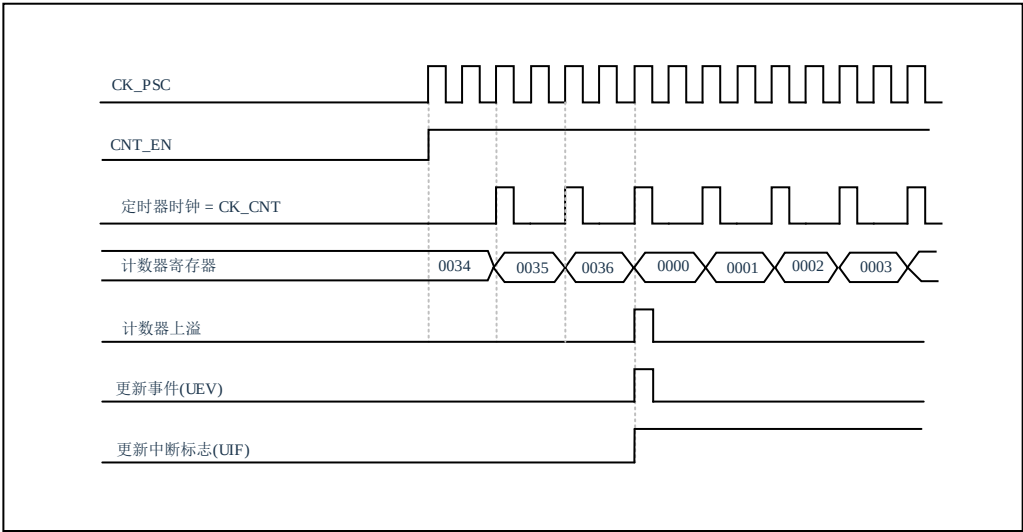


图 6-6 计数器时序图，4 分频内部时钟

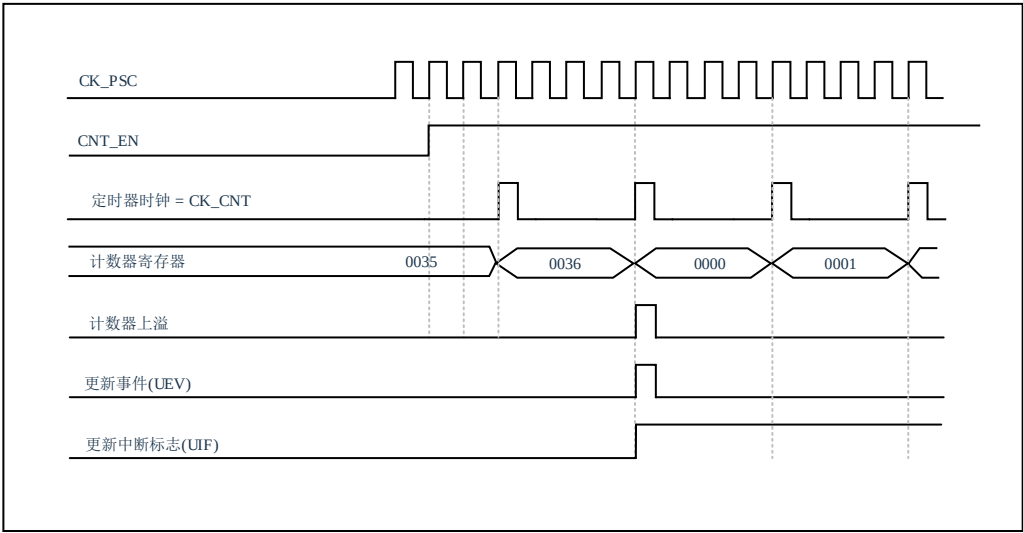


图 6-7 计数器时序图，N 分频内部时钟

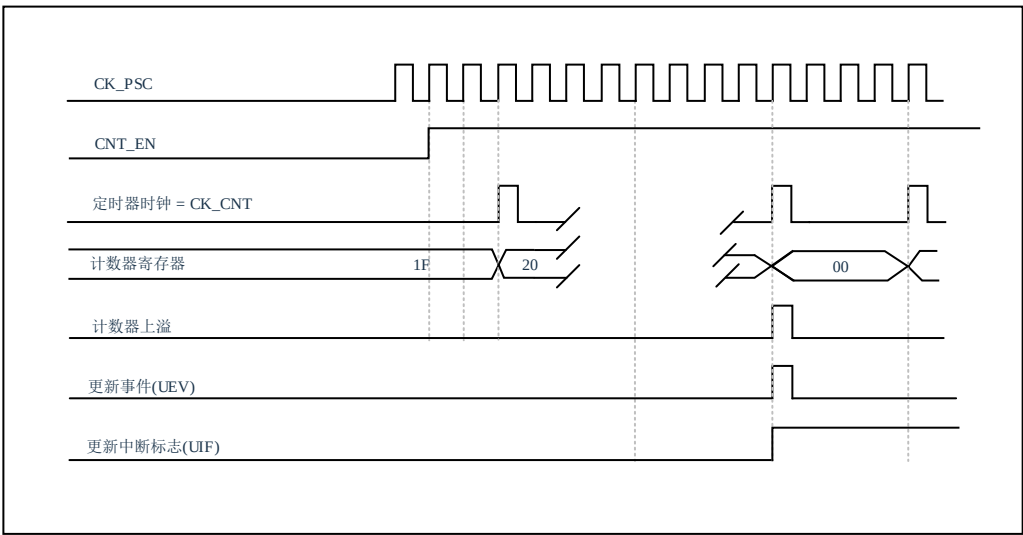


图 6-8 计数器时序图，APRE=0 时更新事件（TIM1_ARR 未预装载）

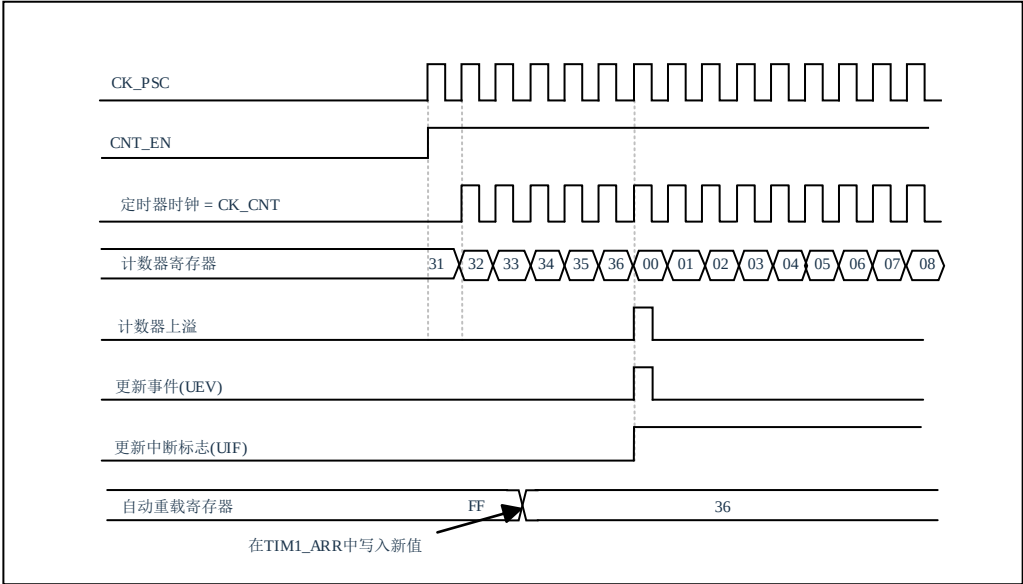
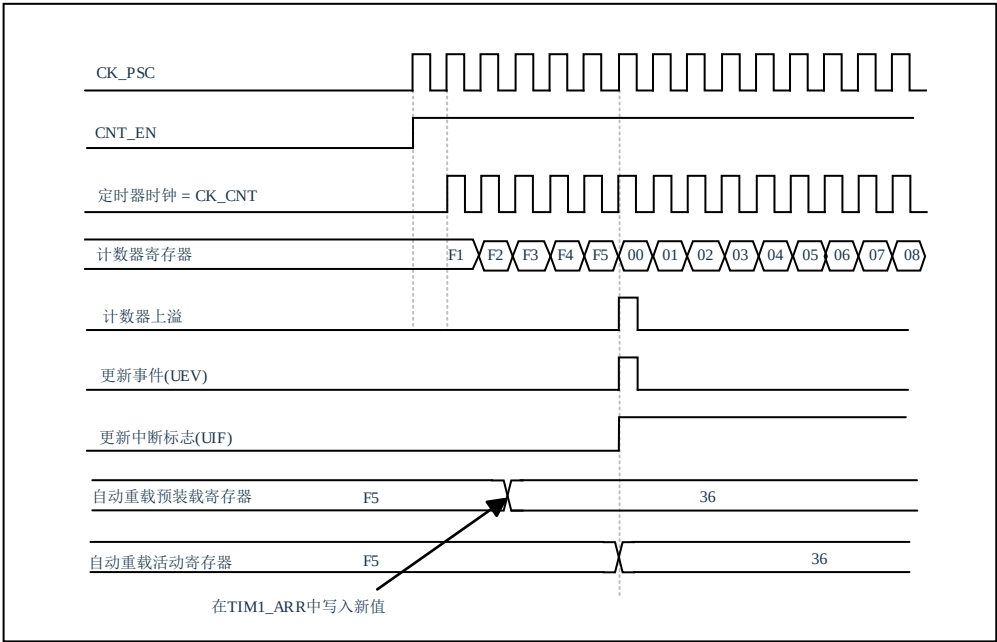


图 6-9 计数器时序图，APRE=1 时更新事件（TIM1_ARR 预装载）



6.3.2.2 向下计数模式

向下计数模式，从自动装载的值开始向下计数到 0，然后重新开始从自动装载的值向下计数，并产生一个向下溢出事件。

如果使用了重复计数器，当向下计数重复了重复计数寄存器(TIM1_RCR)中设定的次数后，将产生更新事件(UEV)，否则每次计数器下溢时才产生更新事件。

在 TIM1_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位，也同样可以产生一个更新事件。

设置 TIM1_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始(但预分频系数不变)。

此外，如果设置了 TIM1_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且(根据 URS 位的设置)更新标志位(TIM1_SR 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 TIM1_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载的值(TIM1_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值(TIM1_ARR 寄存器中的内容)。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下各图以一些示例说明当 TIM1_ARR = 0x36 时不同时钟频率下计数器的行为。

图 6-10 计数器时序图，1 分频内部时钟

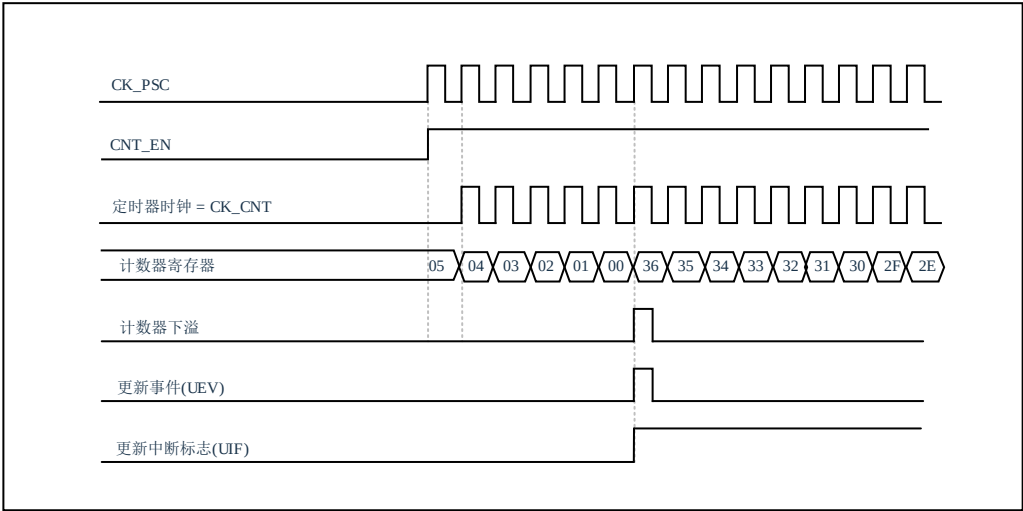
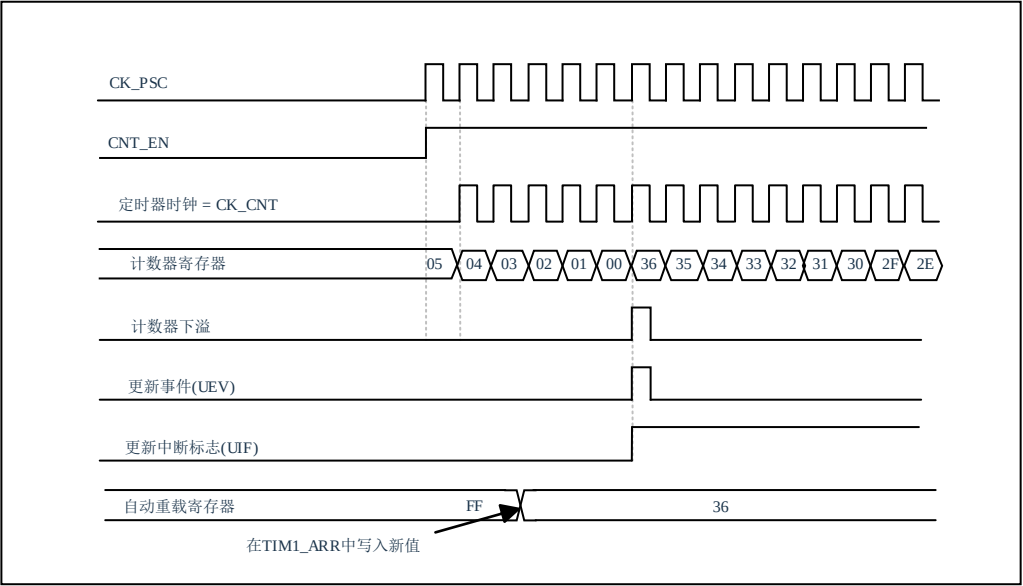


图 6-14 计数器时序图，未使用重复计数器时更新事件



6.3.2.3 中央对齐模式 (向上/向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值(TIM1_ARR 寄存器)-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

中央对齐模式在 TIM1_CR1 寄存器中的 CMS 不等于 0 时有效。通道在配置成输出模式时，输出比较中断标志将被置位，当：向下计数时（中央对齐模式 1，CMS= ” 01 ”），向上计数时（中央对齐模式 2，CMS= ” 10 ”）向上向下计数（中央对齐模式 3，CMS= ” 11 ”）。

在此模式下，不能写入 TIM1_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过(软件或者使用从模式控制器)设置 TIM1_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIM1_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIM1_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV

但不设置 UIF 标志(因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且(根据 URS 位的设置)更新标志位(TIM1_SR 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 TIM1_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载(TIM1_PSC 寄存器)的值。
- 当前的自动加载寄存器被更新为预装载寄存器的值(TIM1_PSC 寄存器的内容)。

注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值(计数器被装载为新的值)

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 6-15 计数器时序图，1 分频内部时钟，TIM1_ARR=0x06

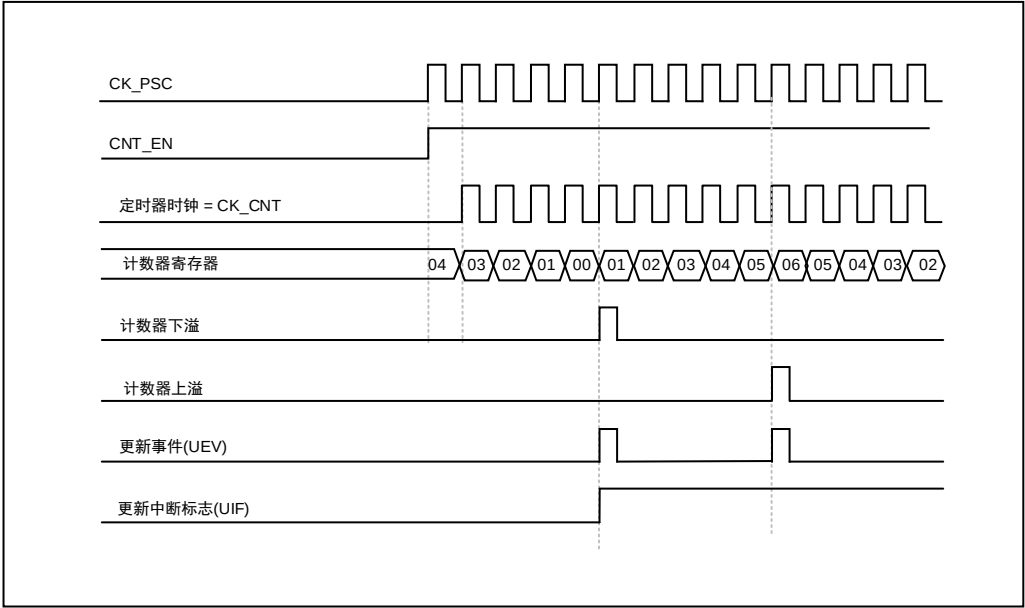


图 6-16 计数器时序图，2 分频内部时钟

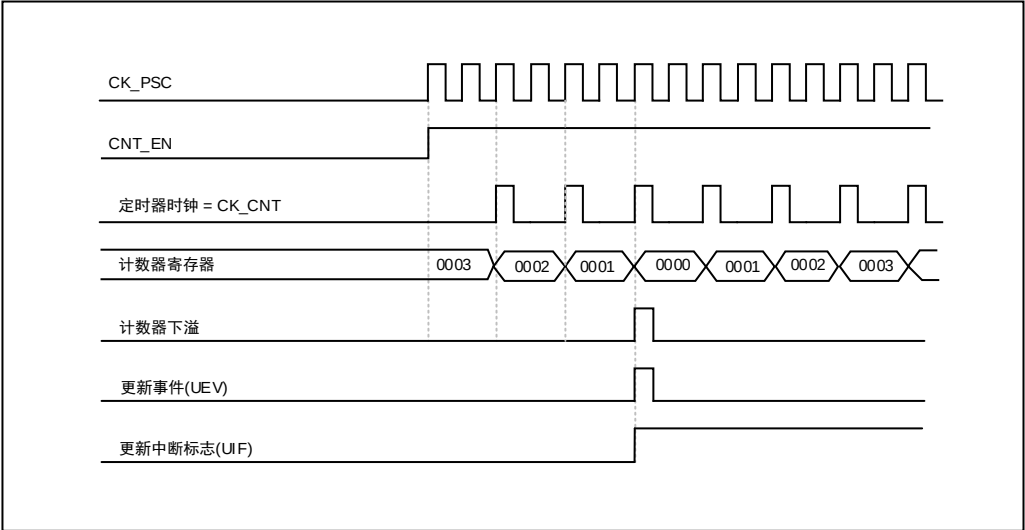


图 6-17 计数器时序图，4 分频内部时钟，TIM1_ARR=0x36

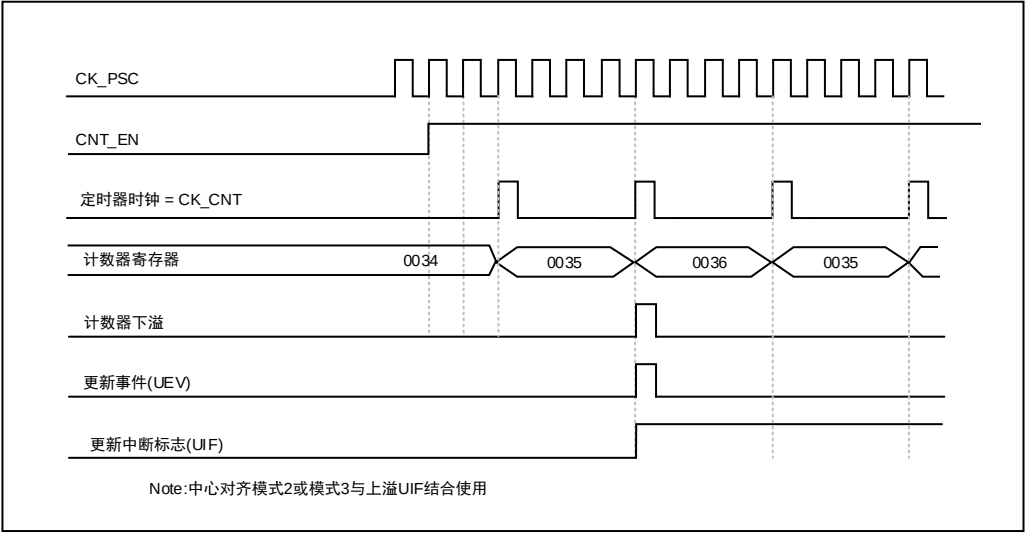


图 6-18 计数器时序图，N 分频内部时钟

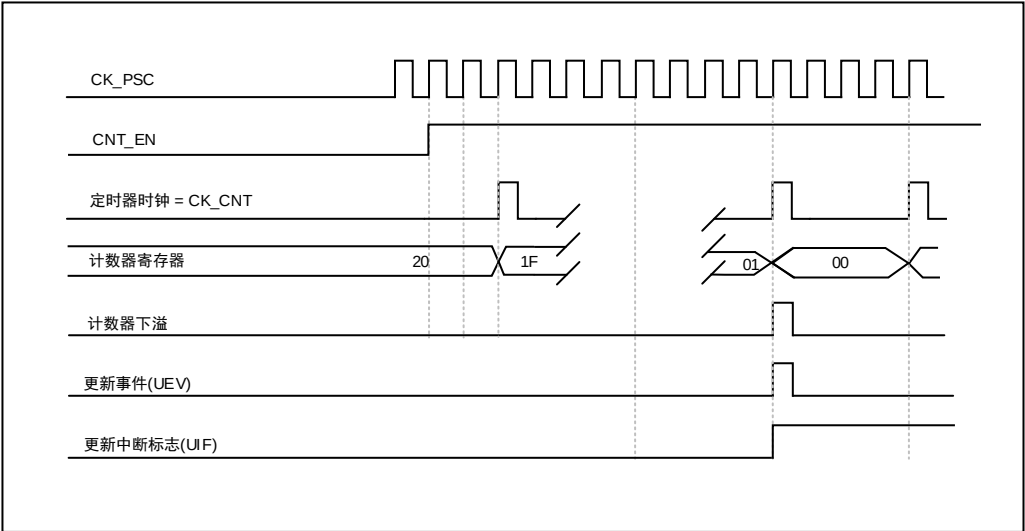


图 6-19 计数器时序图，APRE=1 时的更新事件（计数器下溢）

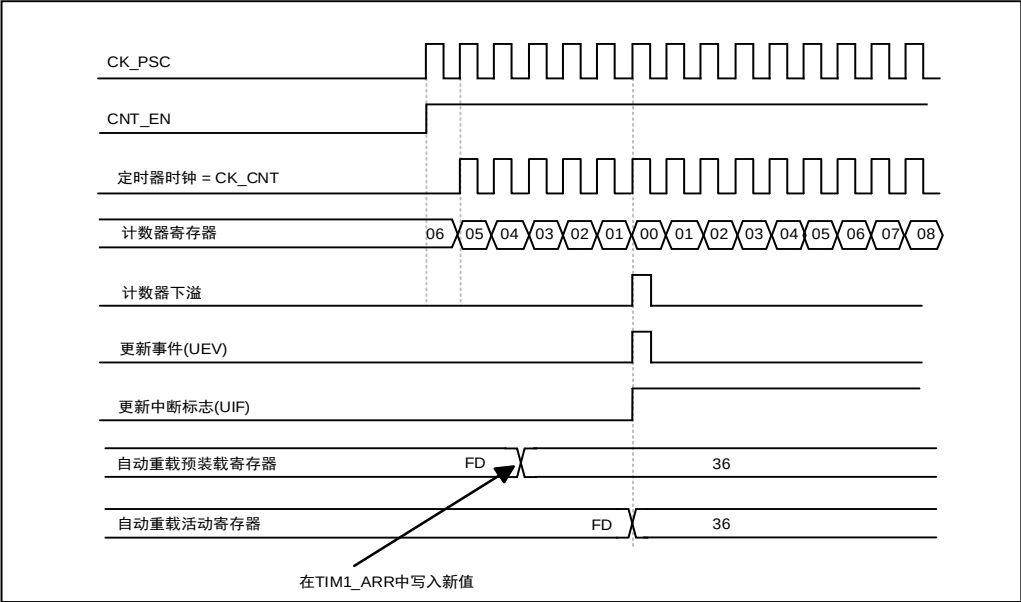
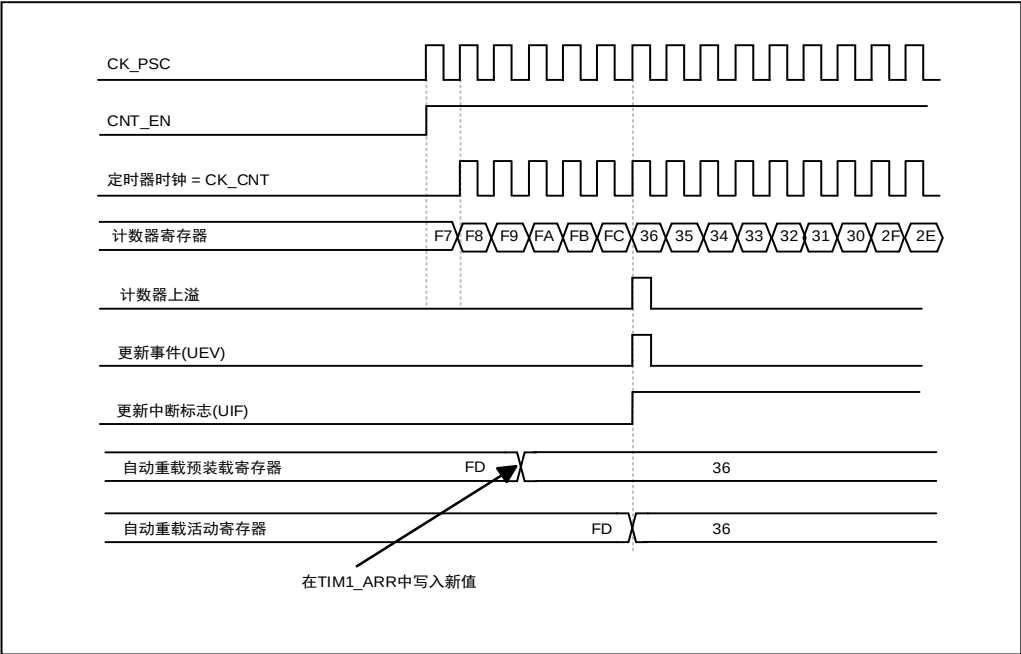


图 6-20 计数器时序图，APRE=1 时的更新事件（计数器上溢）



6.3.3 重复计数器

时基单元描述了关于计数器向上、向下溢出的更新事件如何产生的。它实际上仅当重复计数器计数到零才产生。这也是当产生 PWM 信号时很有用的。

这意味着在每 N 次计数上溢或下溢时，数据被从预装载寄存器传送到影子寄存器（TIM1_ARR 自动重载入寄存器，TIM1_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 TIM1_CCRx），N 是 TIM1_RCR 重复计数寄存器中的值。

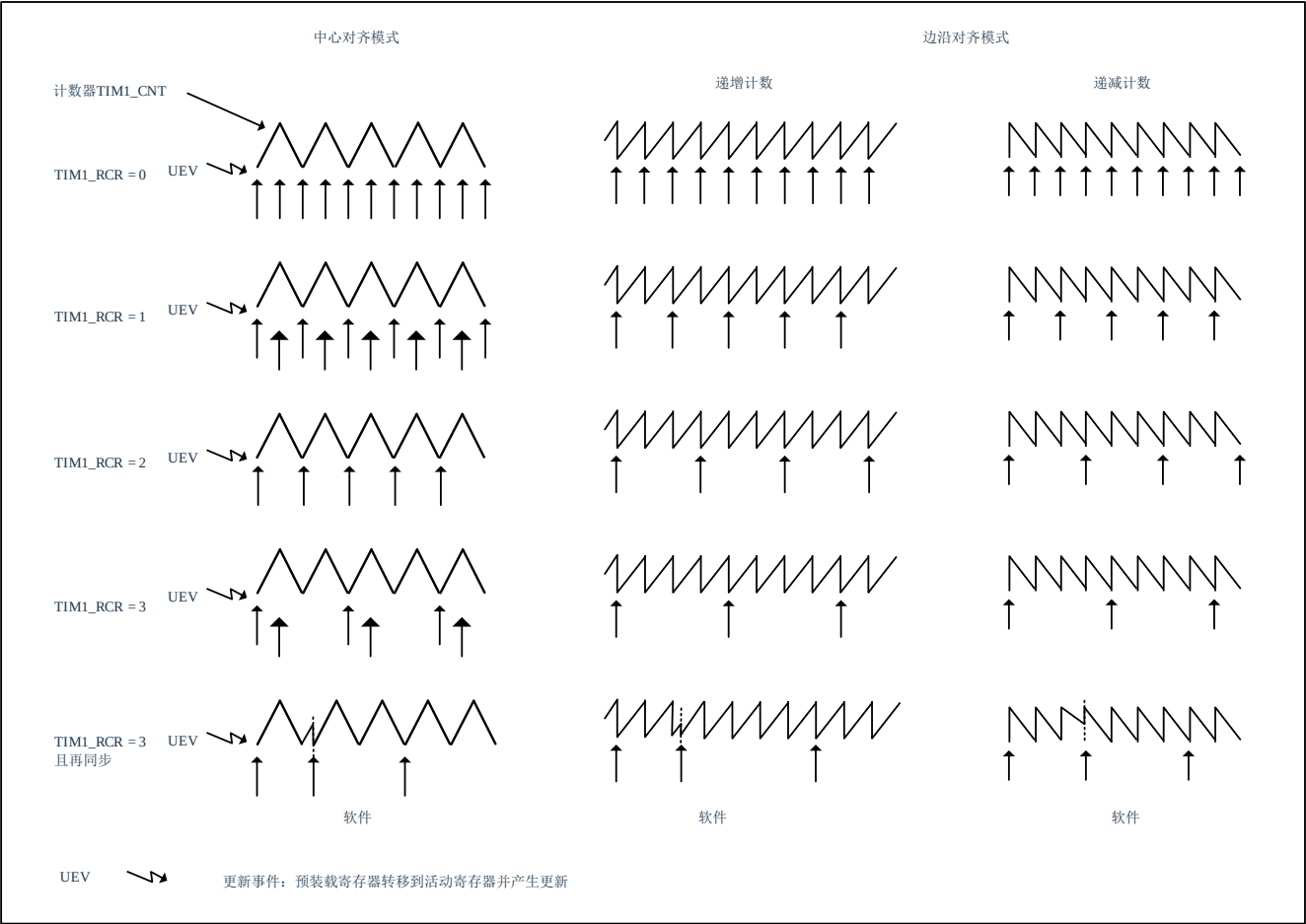
重复计数器在下述任何一条件成立时递减：

- ◆ 向上计数模式下每次计数器溢出时
- ◆ 向下计数模式下每次计数下溢时
- ◆ 中央对齐模式下每次上溢和每次下溢时。虽然这样限制了 PWM 的最大循环周期位 128，但它能够在每个 PWM 周期 2 次更新占空比。在中央对齐模式下，因为波形是对称的，如果每个 PWM 周期中仅刷新一次比较寄存器，则最大的分辨率为 $2 \cdot T_{ck}$ 。

重复计数器是自动加载的，重复速率是由 TIM1_RCR 寄存器的值定义。当更新事件由软件产生（通过设置 TIM1_EGR 中的 UG 位）或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIM1_RCR 寄存器中的内容被重载如到重复计数器。

在中央对齐模式下，对于 RCR 的奇数值，取决于当 RCR 寄存器被写入和当计数器开始，出现上溢、或者下溢，则更新事件产生。如果在启动计数器之前写 RCR，在上溢时产生更新事件。例如，对于 $RCR = 3$ 时，更新事件被产生在第 4 个上溢或者下溢事件（取决于 RCR 被写入的值）。

图 6-21 不同模式下设置不同 TIM1_RCR 寄存器值时的更新频率示例



6.3.4 时钟选择

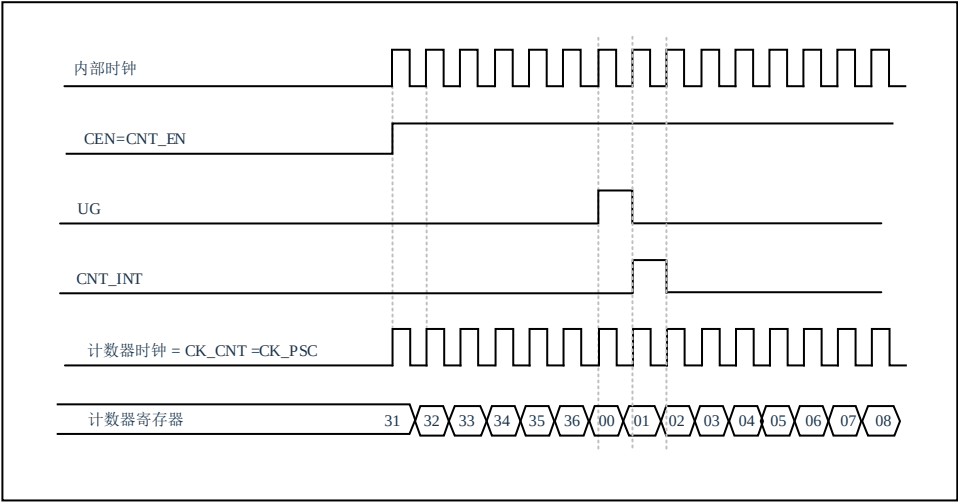
计数器的时钟可以由以下时钟源提供：

- ◆ 内部时钟（CK_INT）
- ◆ 外部时钟模式 1：外部输入引脚
- ◆ 外部时钟模式 2：外部触发输入 ETR
- ◆ 内部触发输入（ITRx）：使用一个定时器作为另一个定时器的预分频器。例如，可以配置一个定时器 TIM1 作为另一个定时器 TIM3 的预分频器。

6.3.4.1 内部时钟源（CK_INT）

如果从模式控制器被禁止，则 CEN、DIR（TIM1_CR1 寄存器）和 UG 位（TIM1_EGR 寄存器）是事实上的控制位，并且只能被软件修改。只要 CEN 位被写成 1，预分频器的时钟就由内部时钟 CK_INT 提供。

图 6-22 正常模式下的控制电路，1 分频内部时钟



6.3.4.2 外部时钟源模式 1

当 TIM1_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图 6-23 TI2 外部时钟连接示例

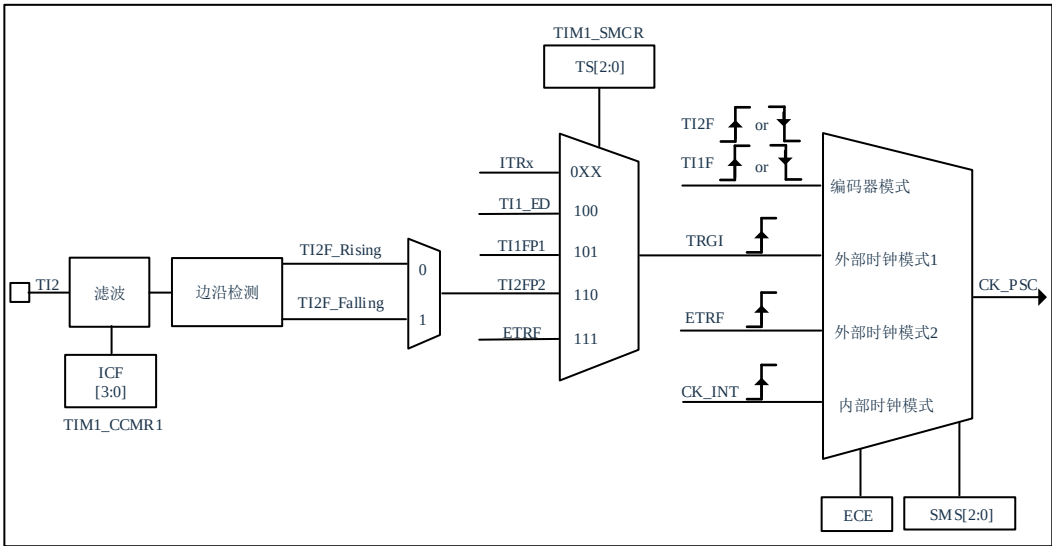
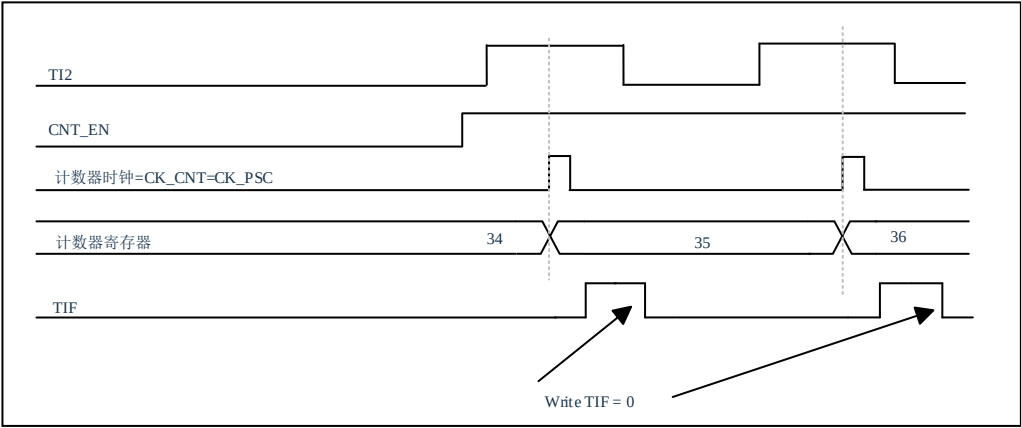


图 6-24 外部时钟模式 1 下的控制电路



6.3.4.3 外部时钟源模式 2

通过写 TIM1_SMCR 寄存器的 ECE 为 1，选定此模式。计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

图 6-25 外部触发输入模块

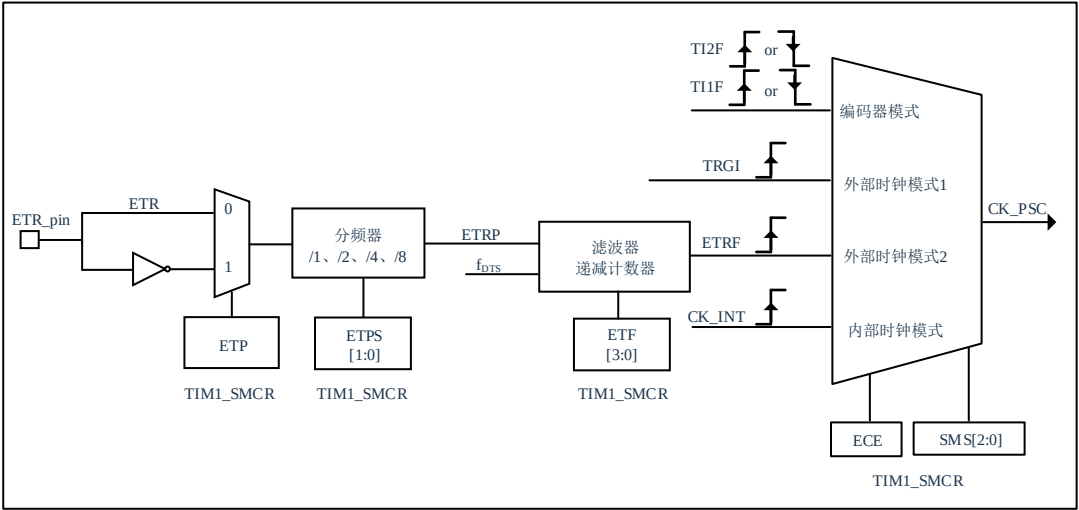


图 6-26 外部时钟模式 2 下的控制电路

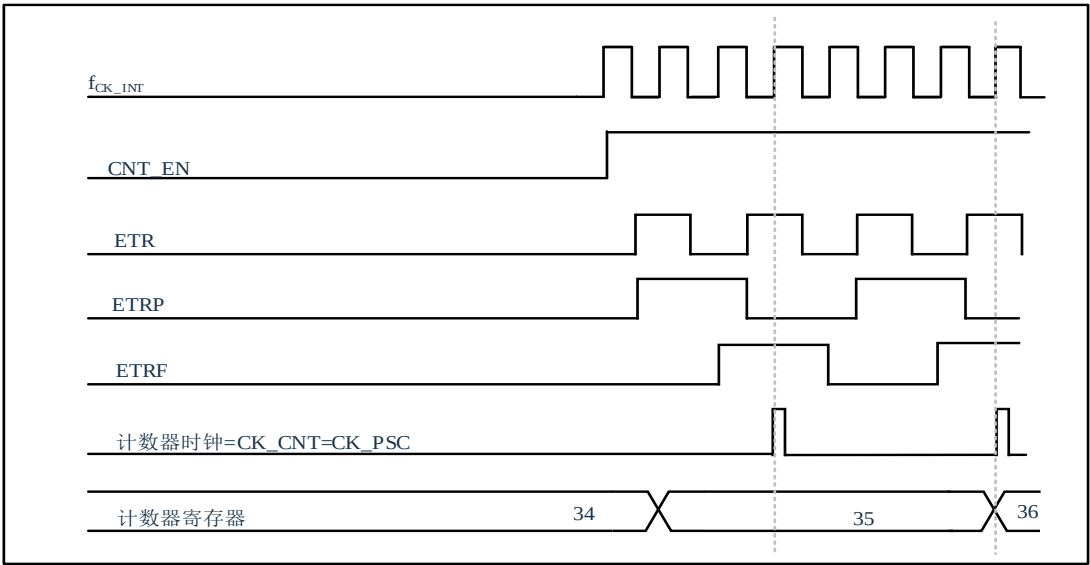


图 6-29：捕获/比较通道的输出阶段（通道 1 到 3）

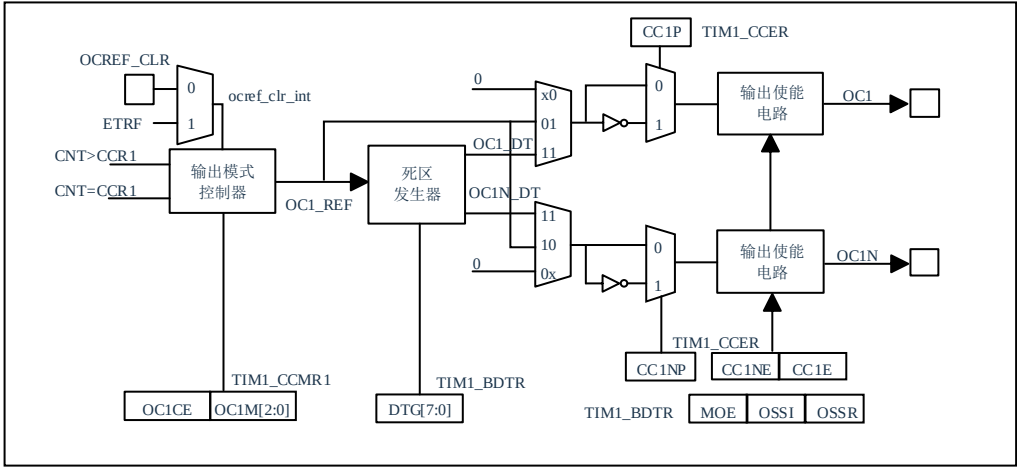
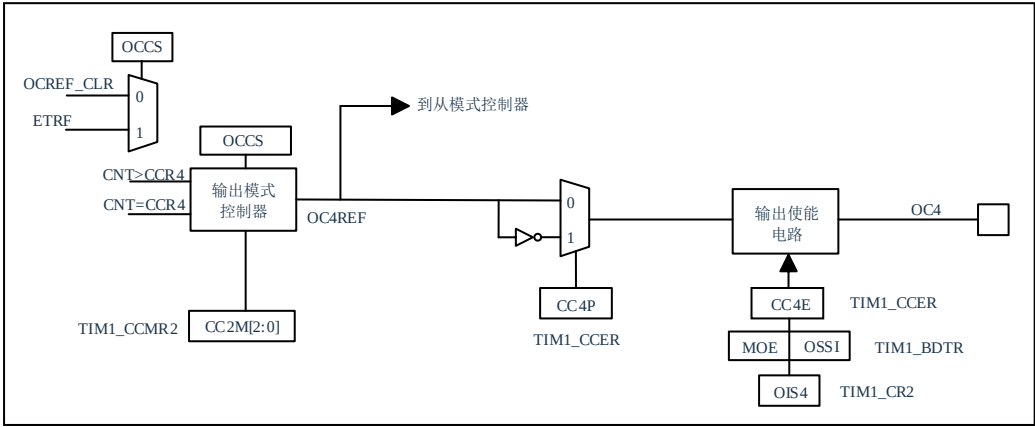


图 6-30 捕获/比较通道的输出阶段（通道 4）



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。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再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被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

6.3.6 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器中。当发生捕获事件时，相应的 CCxIF 标志（TIM1_SR 寄存器）被置 1，如果中断和 DMA 操作被打开，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，则重复捕获标志 CCxOF（TIM1_SR 寄存器）被置 1。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIM1_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM1_CCR1 寄存器中，步骤如下：

- ◆ 选择有效输入端：TIM1_CCMR1 必须连接到 TI1 输入，所以写入 TIM1_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为“00”，通道被配置为输入，并且 TIM1_CCR1 寄存器变为只读。
- ◆ 根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 TIx 时，输入滤波器控制位是 TIM1_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以(以 Fck_int 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM1_CCMR1 寄存器中写入 IC1F=0011。
- ◆ 选择 TI1 通道的有效转换边沿，在 TIM1_CCER 寄存器中写入 CC1P=0(上升沿)。
- ◆ 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIM1_CCMR1 寄存器的 IC1PS=00)。
- ◆ 设置 TIM1_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- ◆ 如果需要，通过设置 TIM1_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIM1_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- ◆ 产生有效的电平转换时，计数器的值被传送到 TIM1_CCR1 寄存器。
- ◆ CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- ◆ 如设置了 CC1IE 位，则会产生一个中断。
- ◆ 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIM1_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

6.3.7 PWM 输入模式

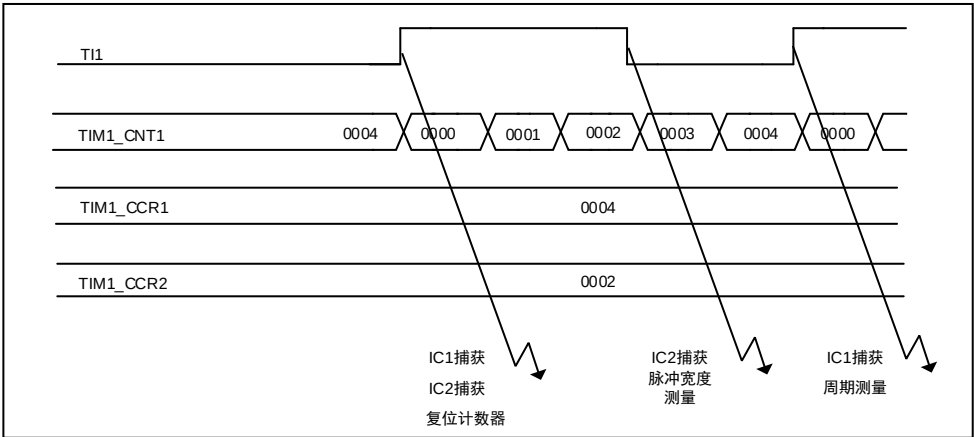
该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- ◆ 两个 ICx 信号被映射到同一个 TIx 输入。
- ◆ 这 2 个 ICx 信号为边沿有效，但是极性相反。
- ◆ 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，当需要测量输入到 TI1 上的 PWM 信号的长度(TIM1_CCR1 寄存器)和占空比(TIM1_CCR2 寄存器)时，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)

- ◆ 选择 TIM1_CCR1 的有效输入：置 TIM1_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- ◆ 选择 TI1FP1 的有效极性(用来捕获数据到 TIM1_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- ◆ 选择 TIM1_CCR2 的有效输入：置 TIM1_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- ◆ 选择 TI1FP2 的有效极性(捕获数据到 TIM1_CCR2)：置 CC2P=1(下降沿有效)。
- ◆ 选择有效的触发输入信号：置 TIM1_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- ◆ 配置从模式控制器为复位模式：置 TIM1_SMCR 中的 SMS=100。
- ◆ 使能捕获：置 TIM1_CCER 寄存器中 CC1E=1 且 CC2E=1。

图 6-31 PWM 输入模式时序



6.3.8 强制输出模式

在输出模式(TIM1_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。置 TIM1_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。置 TIM1_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIM1_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下文的输出比较模式一节中介绍。

6.3.9 输出比较模式

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

- ◆ 将输出比较模式(TIM1_CCMRx 寄存器中的 OCxM 位)和输出极性(TIM1_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- ◆ 设置中断状态寄存器中的标志位(TIM1_SR 寄存器中的 CCxIF 位)。
- ◆ 若设置了相应的中断屏蔽(TIM1_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- ◆ 若设置了相应的使能位(TIM1_DIER 寄存器中的 CCxDE 位，TIM1_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

TIM1_CCMRx 中的 OCxPE 位选择 TIM1_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

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

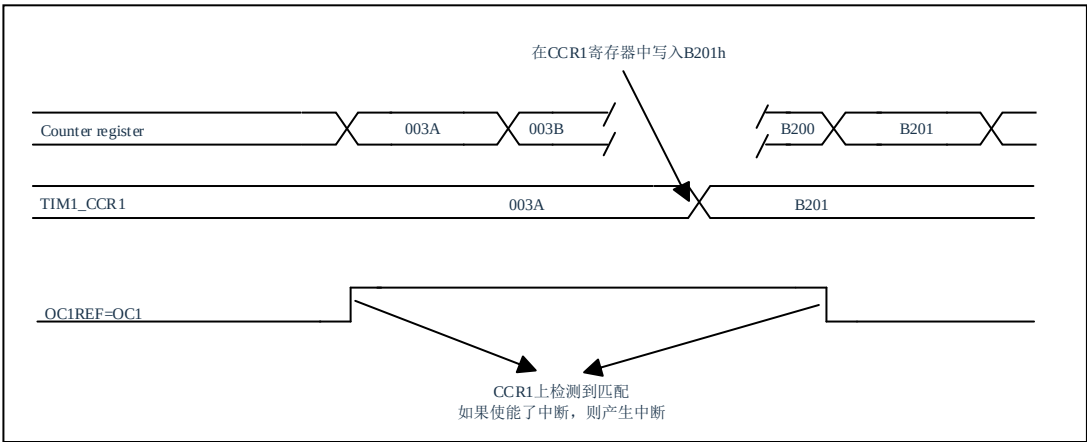
输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。
2. 将相应的数据写入 TIM1_ARR 和 TIM1_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
 - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
 - 置 OCxPE = 0 禁用预装载寄存器
 - 置 CCxP = 0 选择极性为高电平有效
 - 置 CCxE = 1 使能输出

设置 TIM1_CR1 寄存器的 CEN 位启动计数器

TIM1_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OCxPE= ‘0’，否则 TIM1_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 6-32 输出比较模式，翻转 OC1



6.3.10 PWM 模式

脉冲宽度调制模式可以允许产生一个由 TIM1_ARR 寄存器确定频率、由 TIM1_CCRx 寄存器确定占空比的信号。

在 TIM1_CCMRx 寄存器中的 OCxM 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2），能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIM1_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIM1_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中)使能自动重载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIM1_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIM1_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIM1_CCER 和 TIM1_BDTR 寄存器中)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIM1_CCER 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下，TIM1_CNT 和 TIM1_CCRx 始终在进行比较，(依据计数器的计数方向)以确定是否符合 $TIM1\_CCRx \leq TIM1\_CNT$ 或者 $TIM1\_CNT \leq TIM1\_CCRx$ 。

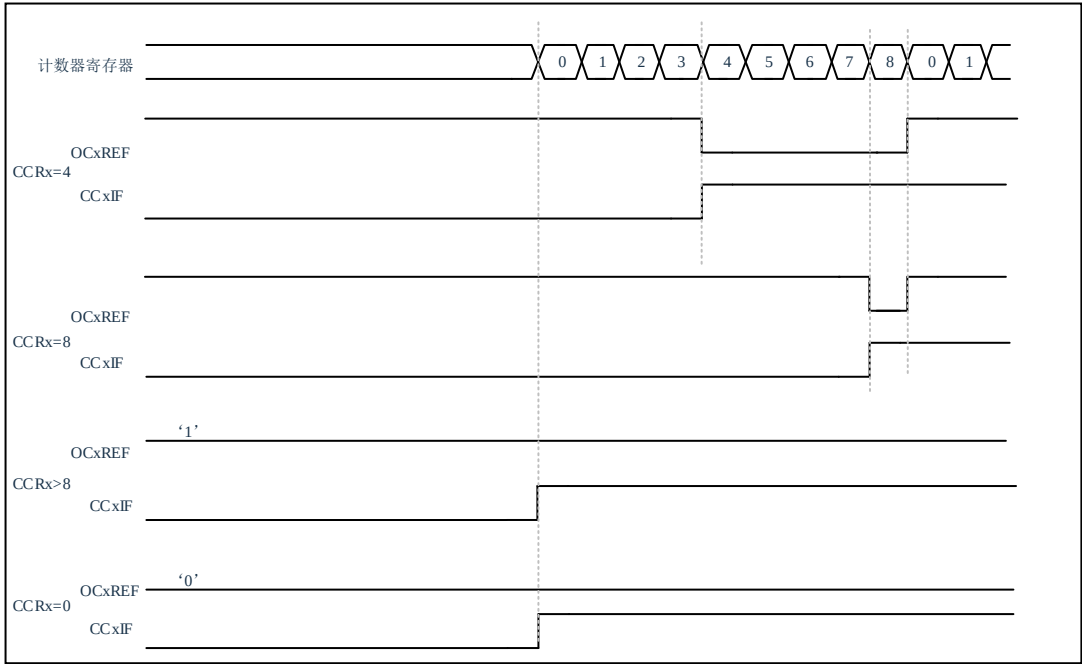
根据 TIM1_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

6.3.10.1 PWM 边沿对齐模式

● 向上计数配置

当 TIM1_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看下面是一个 PWM 模式 1 的例子。当 $TIM1\_CNT < TIM1\_CCR_x$ 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIM1_CCRx 中的比较值大于自动重载值(TIM1_ARR)，则 OCxREF 保持为 '1'。如果比较值为 0，则 OCxREF 保持为 '0'。下图为 TIM1_ARR=8 时边沿对齐的 PWM 波形实例

图 6-33 边沿对齐模式的 PWM 波形（ARR=8）



● 向下计数的配置

当 TIM1_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 $TIM1\_CNT > TIM1\_CCR_x$ 时参考信号 OCxREF 为低，否则为高。如果 TIM1_CCRx 中的比较值大于 TIM1_ARR 中的自动重载值，则 OCxREF 保持为 '1'。该模式下不能产生 0% 的 PWM 波形。

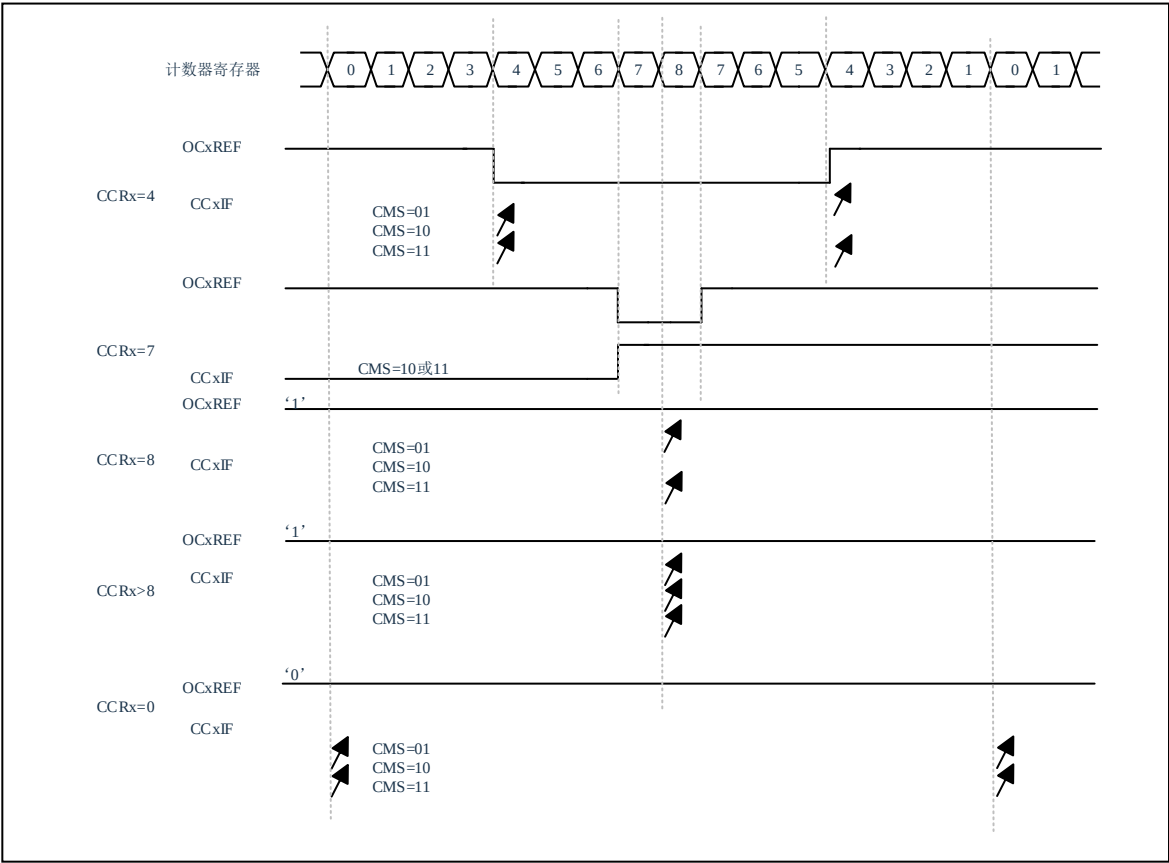
6.3.10.2 PWM 中央对齐模式

当 TIM1_CR1 寄存器中的 CMS 位不为 ‘00’ 时为中央对齐模式(所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIM1_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。

下图给出一些中央对齐的 PWM 波形的例子

- ◆ TIM1_ARR = 8
- ◆ PWM 模式 1
- ◆ TIM1_CR1 寄存器的 CMS=01，在中央对齐模式下，当计数器向下计数时设置比较标志

图 6-34 中心对齐模式 PWM 波形 (ARR=8)



使用中央对齐模式的提示：

- ◆ 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIM1_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- ◆ 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
 - 如果写入计数器的值大于自动重加载的值(TIM1_CNT>TIM1_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
 - 如果将 0 或者 TIM1_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- ◆ 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设置 TIM1_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

6.3.11 互补输出和死区插入

高级控制定时器(TIM1)能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性(电平转换的延时、电源开关的延时等)来调整死区时间。

配置 TIM1_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性(主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIM1_CCER 寄存器的 CCxE 和 CCxNE 位，TIM1_BDTR 和 TIM1_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见表 7-3 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别是，在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 8 位的死区发生器 DTG[7:0]。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效：

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

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

图 6-35 带死区插入的互补输出

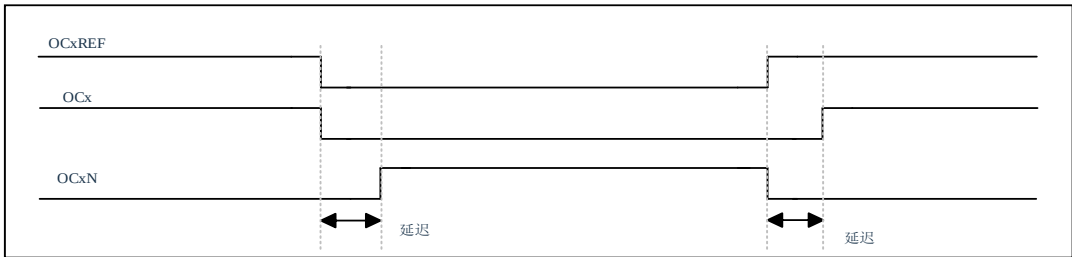


图 6-36 延迟时间大于负脉冲宽度的死区波形

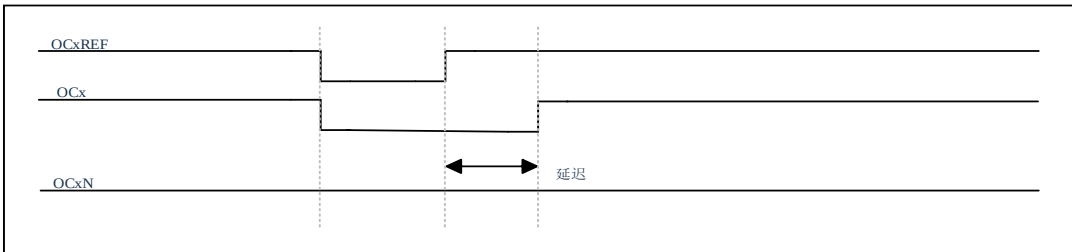
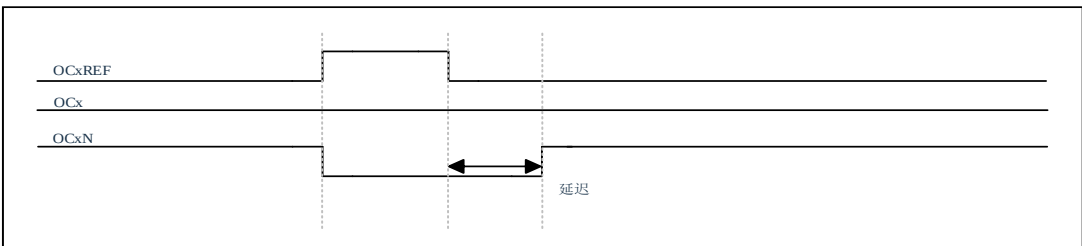


图 6-37 延迟时间大于正脉冲宽度的死区波形



每一个通道的死区延时都是相同的，是由 TIM1_BDTR 寄存器中的 DTG 位编程配置。

6.3.11.1 重定向 OCxREF 到 OCx 或 OCxN

在输出模式下(强置、输出比较或 PWM)，通过配置 TIM1_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形(例如 PWM 或者静态有效电平)。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

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

6.3.12 使用刹车功能

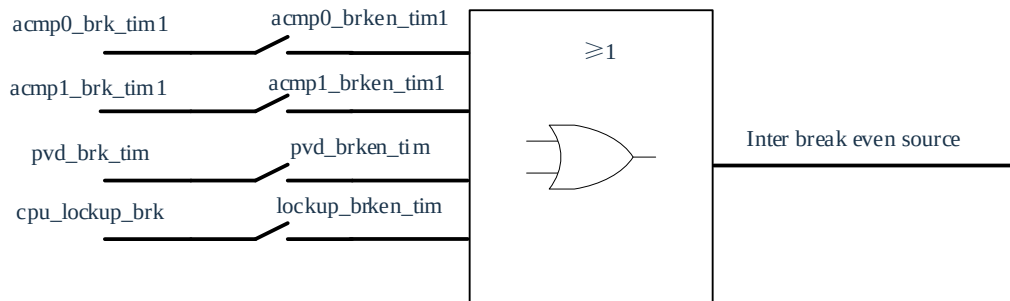
当使用刹车功能时，依据额外的控制位，输出使能信号和无效电平信号都会被修改。无论什么情况下，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。

刹车源既可以是刹车输入引脚，或者以下内部源：

- ◆ CPU 的 LOCKUP 输出
- ◆ PVD 输出
- ◆ 来自比较器的输出

刹车内部源框图如下：

图 6-38 刹车内部源框图



系统复位后，刹车电路被禁止，MOE 位为低。设置 TIM1_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号(作用在输出端)和同步控制位(在 TIM1_BDTR 寄存器中)之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时(空指令)才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

- ◆ MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSS1 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- ◆ 一旦 MOE=0，每一个输出通道输出由 TIM1_CR2 寄存器中的 OISx 位设定的电平。如果 OSS1=0，则定时器释放使能输出，否则使能输出始终为高。
- ◆ 当使用互补输出时：
 - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些(大约 2 个 ck_tim 的时钟周期)。
 - 如果 OSS1=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCCxNE 之一变高时，使能输出变为高。
- ◆ 如果设置了 TIM1_DIER 寄存器中的 BIE 位，当刹车状态标志(TIM1_SR 寄存器中的 BIF 位)为'1'时，则产生一个中断。
- ◆ 如果设置了 TIM1_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置'1'；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

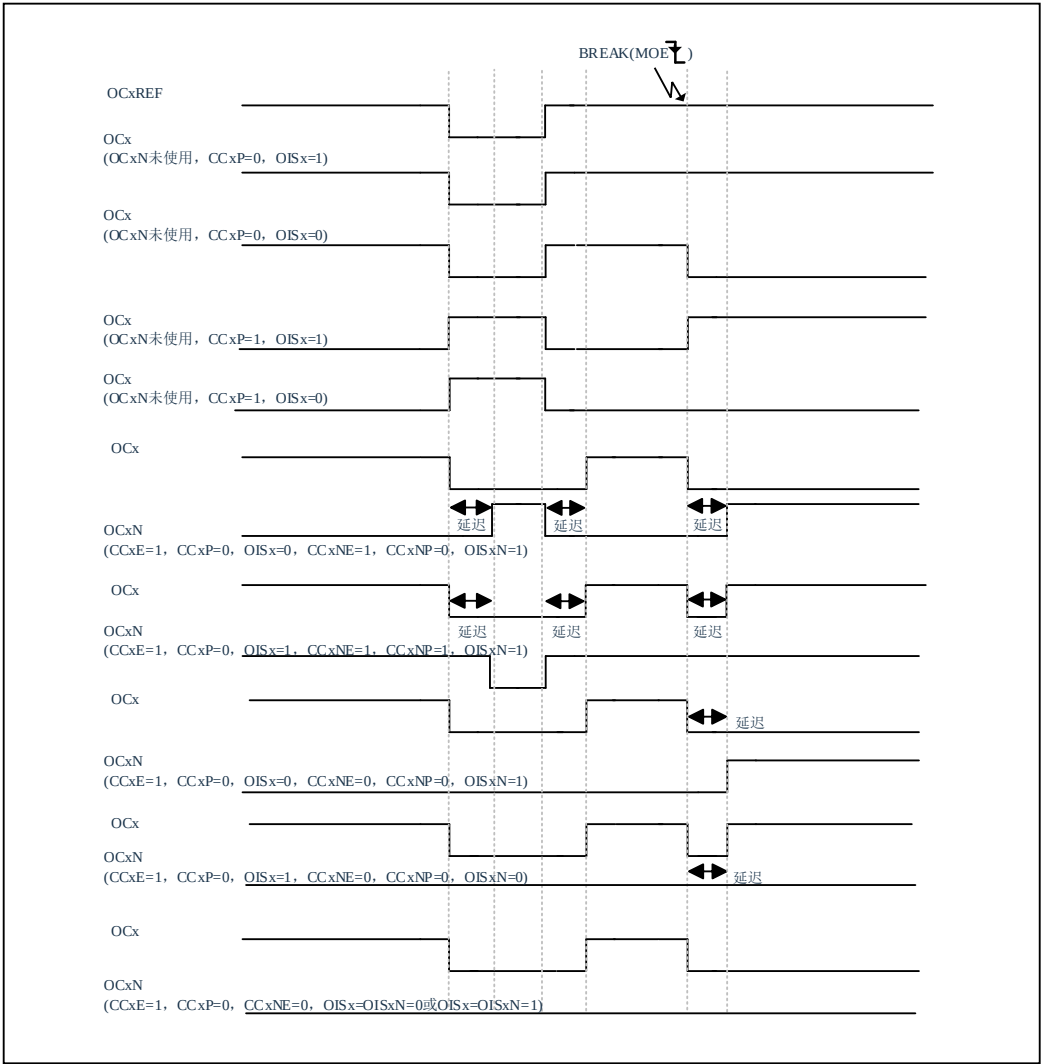
注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时(自动地或者通过软件)设置 MOE。同时，状态标志 BIF 不能被清除。

刹车可以由 BRK 输入产生，它的有效极性是可编程的，且由 TIM1_BDTR 寄存器中的 BKE 位开启。

除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。用户可以通过 TIM1_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

图 6-39 输出的刹车响应行为



6.3.13 发生外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIM1_CCMRx 寄存器中对应的 OCxCE 位为 1，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低电平，直到下一次的更新事件 UEV。

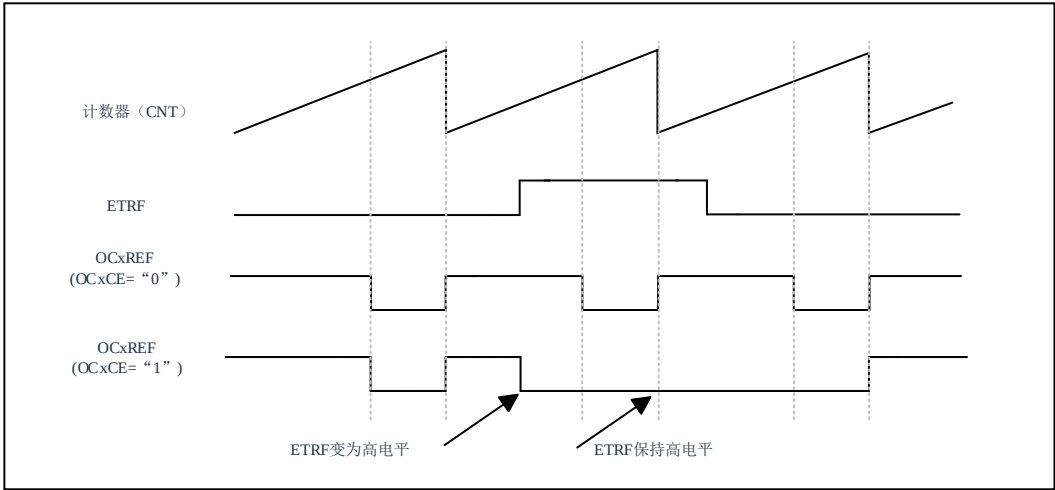
该功能只能用于输出比较和 PWM 模式，而不能用于强制模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

- 1. 外部触发预分频器必须处于关闭：TIM1_SMCR 寄存器中的 ETPS[1:0]=00。
- 2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE=0。
- 3. 外部触发极性(ETP)和外部触发滤波器(ETF)可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIM1 被置于 PWM 模式。

图 6-40：清除 TIM1 的 OCxREF



注意：如果 PWM 的占空比为 100% (CCR_x>ARR)，则下次计数器溢出时会再次使能 OCxREF。

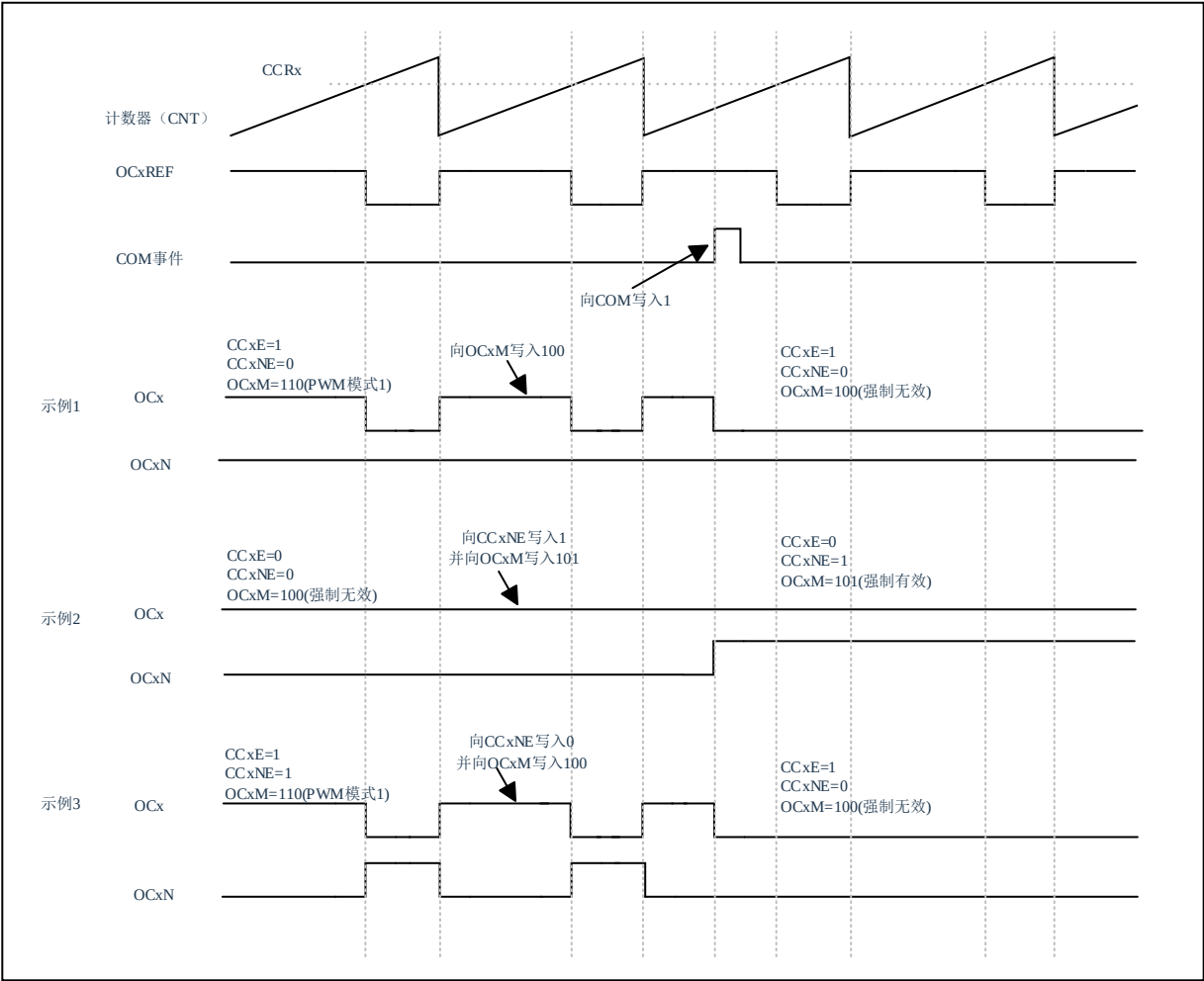
6.3.14 生成 6 步 PWM

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM commutation 事件时，这些预装载位被传送到影子寄存器位。这样就可以预先设置好下一步骤配置，并在同一个时刻同时修改所有通道的配置。COM 可以通过设置 TIM1_EGR 寄存器的 COM 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位(TIM1_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIM1_DIER 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIM1_DIER 寄存器的 COMDE 位，则产生一个 DMA 请求。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出

图 6-41 COM 事件生成 6 步 PWM 的示例（OSSR=1）



6.3.15 单脉冲模式

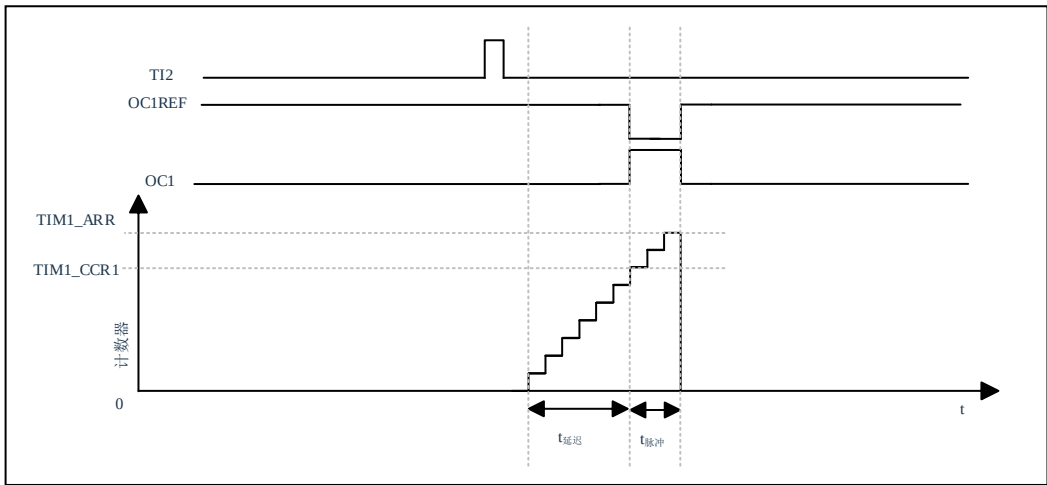
单脉冲模式（OPM）是之前所述众多模式中的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后，产生一个脉宽可被程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIM1_CR1 寄存器的 OPM 位将选择单脉冲模式，这样可以让计数器自动的在产生下一个更新事件 UEV 时停止。

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

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

图 6-42 单脉冲模式示例



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

使用 TI2FP2 作为触发 1：

- ◆ 置 TIM1_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- ◆ 置 TIM1_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- ◆ 置 TIM1_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发(TRGI)。
- ◆ 置 TIM1_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)

- ◆ t_{DELAY} 由 TIM1_CCR1 寄存器中的值定义。
- ◆ t_{PULSE} 由自动装载值和比较值之间的差值定义($TIM1\_ARR - TIM1\_CCR1$)。
- ◆ 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形；

首先要置 TIM1_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIM1_CCMR1 中的 OC1PE=1 和 TIM1_CR1 寄存器中的 ARPE；然后在 TIM1_CCR1 寄存器中填写比较值，在 TIM1_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIM1_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIM1_CR1 寄存器中的 OPM=1，在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIMx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。

如果要以最小延时输出波形，可以设置 TIM1_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx)直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

6.3.16 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIM1_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。

通过设置 TIM1_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参见表 6-1，假定计数器已经启动(TIM1_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIM1_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIM1_ARR 寄存器的自动装载值之间连续计数(根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIM1_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 6-1 计数方向与编码器信号的关系

有效边沿	对应相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI1FP2 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- ◆ CC1S= ‘01’(TIM1_CCMR1 寄存器，TI1FP1 映射到 TI1)
- ◆ CC2S= ‘01’(TIM1_CCMR2 寄存器，TI1FP2 映射到 TI2)
- ◆ CC1P= ‘0’(TIM1_CCER 寄存器，TI1FP1 不反相，TI1FP1=TI1)
- ◆ CC2P= ‘0’(TIM1_CCER 寄存器，TI1FP2 不反相，TI1FP2=TI2)
- ◆ SMS= ‘011’(TIM1_SMCR 寄存器，所有的输入均在上升沿和下降沿有效)
- ◆ CEN= ‘1’(TIM1_CR1 寄存器，计数器使能)

图 6-43 编码器接口模式下的计数器工作示例

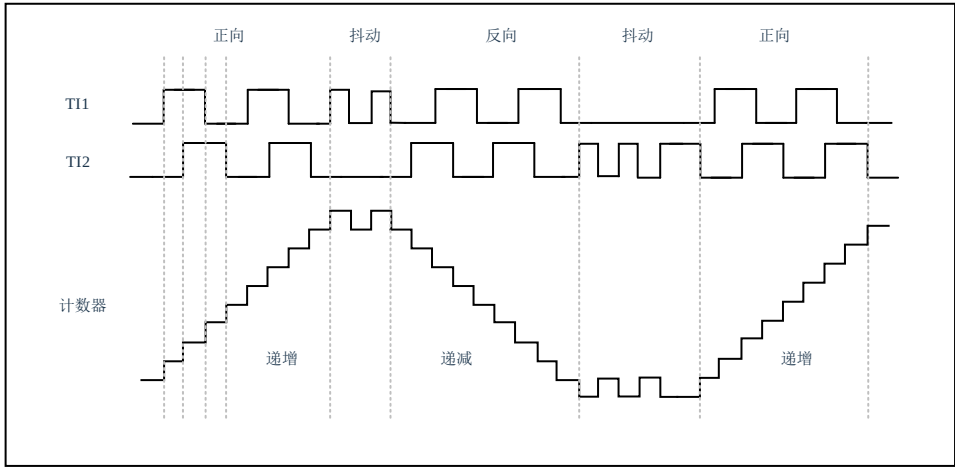
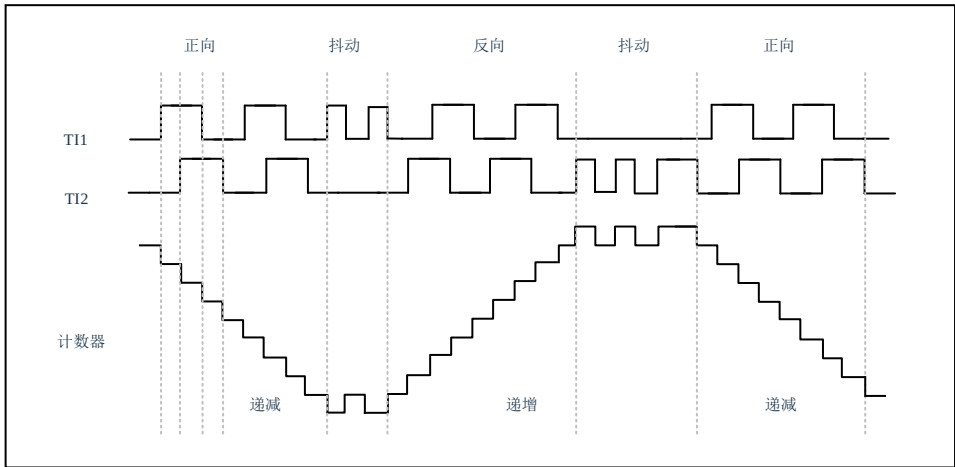


图 6-44 TI1FP1 极性反相时的编码器接口模式示例



当定时器配置成编码器接口模式时，提供传感器当前的位置信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息（速度、加速度、减速度）。指示机械零点的编码器输出可被用作此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的，并且可以由另一个定时器产生）；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

6.3.17 定时器输入异或功能

TIM1_CR2 寄存器的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIM1_CH1、TIM1_CH2 和 TIM1_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

6.3.18 连接霍尔传感器

使用高级控制定时器（TIM1）产生 PWM 信号驱动马达时，可以使用另一个通用定时器（TIM3）作为“接口定时器”来连接霍尔传感器。TIM3 的定时器输入脚（CC1、CC2、CC3）通过一个异或门连接到 TI1 输入通道（通过设置 TIM3_CR2 寄存器中的 TI1S 位来选择），“接口定时器”捕获这个信号。

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

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

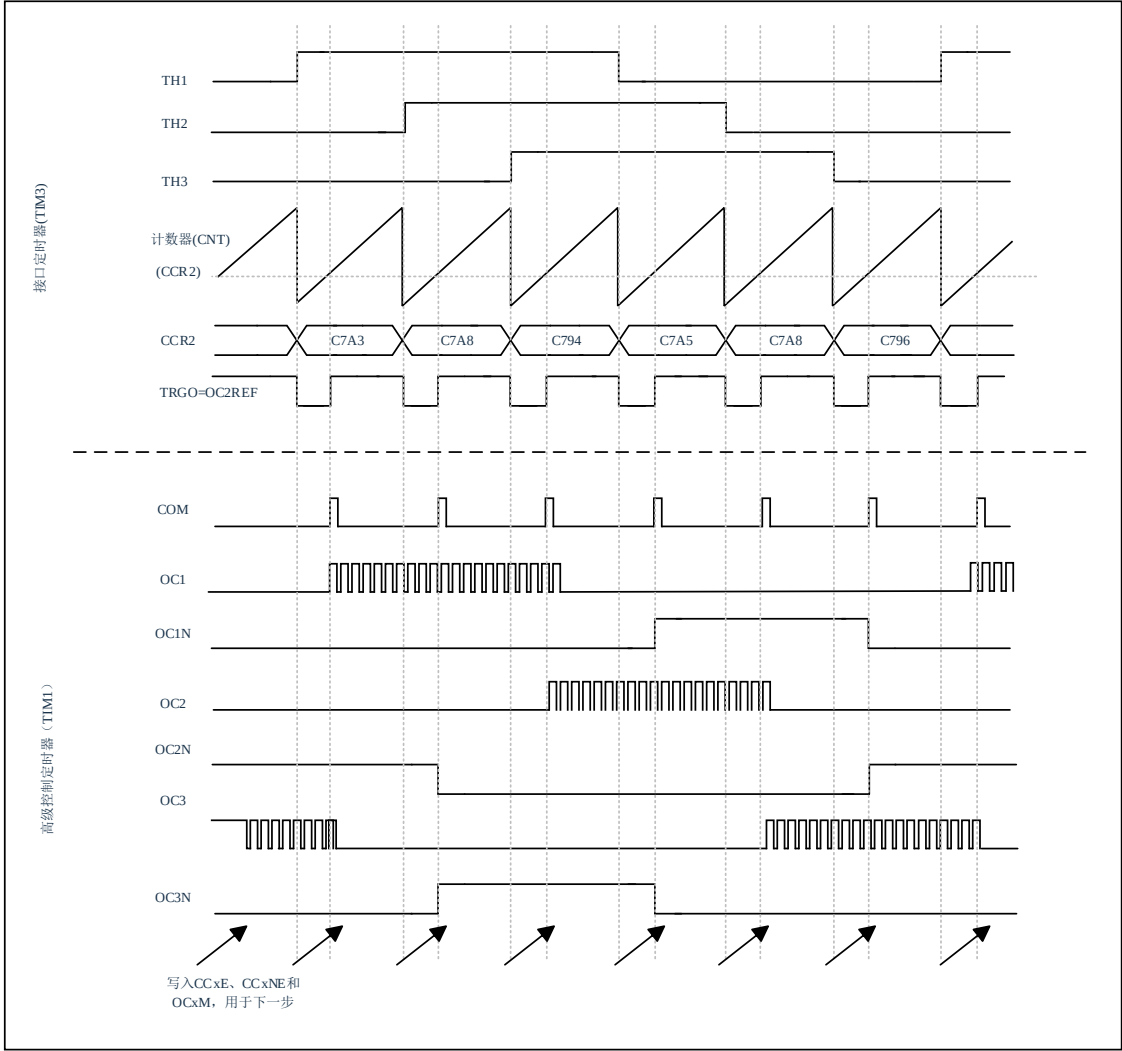
接口定时器可以用来在输出模式产生一个脉冲，这个脉冲可以（通过触发一个 COM 事件）用于改变高级控制定时器 TIM1 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此接口定时器通道必须编程为一个指定的延迟（输出比较或 PWM 模式）之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1。

举例：霍尔输入连接到 TIM3 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIM1 的 PWM 配置。

- ◆ 置 TIM3_CR2 寄存器的 TI1S 位为‘1’，配置三个定时器输入逻辑或到 TI1 输入。
- ◆ 时基编程：置 TIM3_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- ◆ 设置通道 1 为捕获模式(选中 TRC)：置 TIM3_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- ◆ 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIM3_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- ◆ 选择 OC2REF 作为 TRGO 上的触发输出：置 TIM3_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIM1_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件(TIM1_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一步的 PWM 控制位(CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

图 6-45 霍尔传感器接口的示例



6.3.19 TIM1 与外部触发同步

TIM1 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

6.3.19.1 从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIM1_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIM1_ARR, TIM1_CCRx)都被更新了。

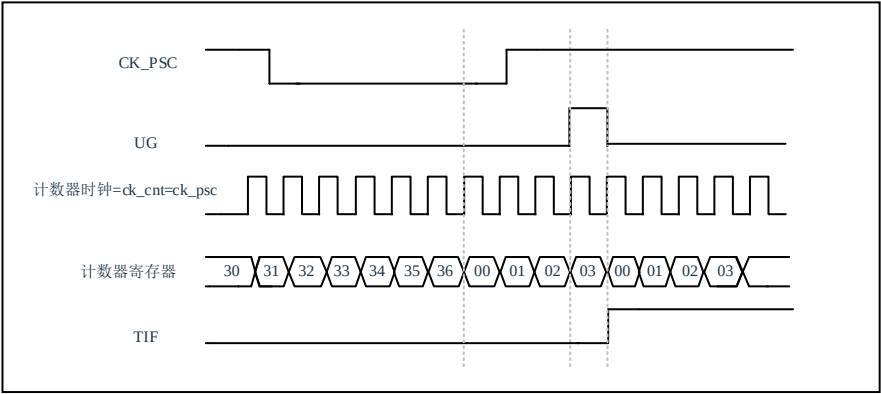
在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- ◆ 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIM1_CCMR1 寄存器中 CC1S=01。置 TIM1_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- ◆ 置 TIM1_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- ◆ 置 TIM1_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIM1_SR 寄存器中的 TIF 位)被设置，根据 TIM1_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIM1_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

图 6-46 复位模式下的控制电路

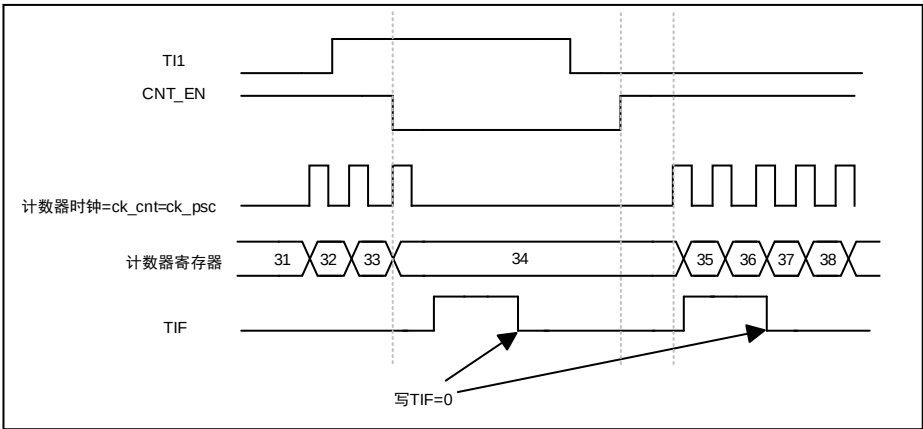


6.3.19.2 从模式：门控模式

- 按照选中的输入端电平使能计数器。
- 在如下的例子中，计数器只在 TI1 为低时向上计数：
- ◆ 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 IC1F=0000)。
 - 触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIM1_CCMR1 寄存器中 CC1S=01。置 TIM1_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
 - ◆ 置 TIM1_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
 - ◆ 置 TIM1_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。
- 只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIM1_SR 中的 TIF 标志。

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

图 6-47 门控模式下的控制电路



6.3.19.3 从模式:触发模式

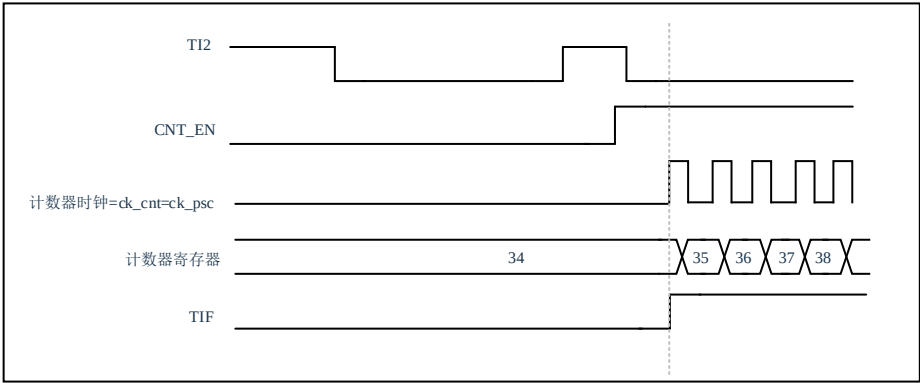
输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- ◆ 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIM1_CCMR1 寄存器中 CC2S=01。置 TIM1_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- ◆ 置 TIM1_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM1_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 6-48 触发模式下的控制电路



6.3.19.4 从模式：外部时钟模式 2 + 触发模式

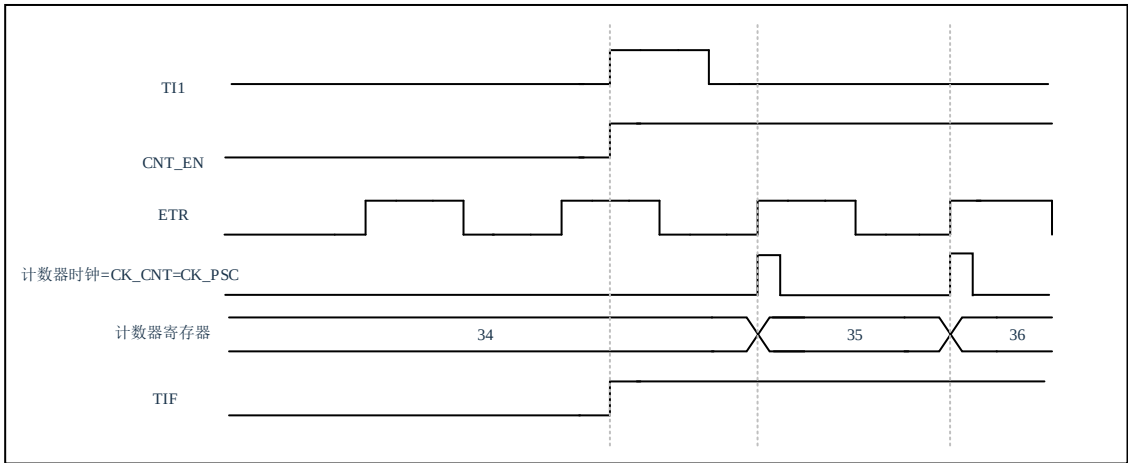
外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIM1_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

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

- 1. 通过 TIM1_SMCR 寄存器配置外部触发输入电路：
 - ETF=0000：没有滤波
 - ETPS=00：不用预分频器
 - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。
- 2. 按如下配置通道 1，检测 TI 的上升沿：
 - IC1F=0000：没有滤波
 - 触发操作中不使用捕获预分频器，不需要配置
 - 置 TIM1_CCMR1 寄存器中 CC1S=01，选择输入捕获源
 - 置 TIM1_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)
- 3. 置 TIM1_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

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

图 6-49 外部时钟模式 2 + 触发模式下的控制电路



6.3.20 定时器同步

TIM 定时器在内部相连，用于定时器的同步或者链接功能。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或时钟等操作。有关详细信息，请参考第 7.3.15 节：定时器同步。

6.3.21 调试模式

当芯片进入调试模式时，根据 1.4 中的设置，TIM1 计数器可以继续正常工作或者停止工作。

6.4 寄存器映射

(TIM1 基地址 = 0x4007_0000) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
CR1	0x00	R/W	TIM1 控制寄存器 1	0x00
CR2	0x04	R/W	TIM1 控制寄存器 2	0x00
SMCR	0x08	R/W	TIM1 从模式控制寄存器	0x00
DIER	0x0C	R/W	TIM1 DMA/中断使能寄存器	0x00
SR	0x10	R/W	TIM1 状态寄存器	0x00
EGR	0x14	R/W	TIM1 事件生成寄存器	0x00
CCMR1	0x18	R/W	TIM1 捕获/比较模式寄存器 1	0x00
CCMR2	0x1C	R/W	TIM1 捕获/比较模式寄存器 2	0x00
CCER	0x20	R/W	TIM1 捕获/比较使能寄存器	0x00
CNT	0x24	R/W	TIM1 计数器	0x00
PSC	0x28	R/W	TIM1 预分频器	0x00
ARR	0x2C	R/W	TIM1 自动重载寄存器	0xFFFF
RCR	0x30	R/W	TIM1 重复计数器寄存器	0x00
CCR1	0x34	R/W	TIM1 捕获/比较寄存器 1	0x00
CCR2	0x38	R/W	TIM1 捕获/比较寄存器 2	0x00
CCR3	0x3C	R/W	TIM1 捕获/比较寄存器 3	0x00
CCR4	0x40	R/W	TIM1 捕获/比较寄存器 4	0x00
BDTR	0x44	R/W	TIM1 刹车和死区寄存器	0x00
DCR	0x48	R/W	TIM1 DMA 控制寄存器	0x00
DMAR	0x4C	R/W	TIM1 全传输 DMA 地址	0x00

6.5 寄存器说明

6.5.1 TIM1 控制寄存器 1 (TIM1_CR1)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9:8	CKD[1:0]	时钟分频 (Clock division) 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (ETR、TIX) 所使用的死区及采样时钟 (t _{DTS}) 之间的分频比 00: t _{DTS} = t _{CK_INT} 01: t _{DTS} = 2 * t _{CK_INT} 10: t _{DTS} = 4 * t _{CK_INT} 11: 保留，不要设置成此值	0x0
7	ARPE	自动重载预装载使能 0: TIM1_ARR 寄存器不进行缓冲 1: TIM1_ARR 寄存器进行缓冲	0
6:5	CMS[1:0]	中心对齐模式选择 00: 边沿对齐模式。边沿对齐模式。计数器依据方向位(DIR) 向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIM1_CCMRx 寄存器 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置 10: 中央对齐模式 2。计数器交替地向上和向下计数。计数器交替地向上和向下计数。配置为输出的通道(TIM1_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置 11: 中央对齐模式 3。计数器交替地向上和向下计数。计数器交替地向上和向下计数。配置为输出的通道(TIM1_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。	0x0
4	DIR	方向 0: 计数器向上计数 1: 计数器向下计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读	0
3	OPM	单脉冲模式 0: 计数器在发生更新事件时不会停止计数 1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止	0
2	URS	更新请求源 此位由软件置 1 和清零，用以选择 UEV 事件源 0: 如果允许产生更新中断或 DMA 请求，则下述任一事件产生 — 一个更新中断或 DMA 请求 — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件 1: 如果允许产生更新中断或 DMA 请求，如果允许产生更新中断或 DMA 请求，则只有计数器上溢/下溢产生一个更新中断或 DMA 请求	0
1	UDIS	更新禁止 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0: 允许 UEV。更新 (UEV) 事件可通过以下事件之一生成 — 计数器上溢/下溢 — 将 UG 位置 1 — 从模式控制器产生的更新 被缓存的寄存器被装入它们的预装载值 1: 禁止 UEV。不产生更新事件，影子寄存器 (ARR,PSC,CCRx)保持它们的值。 如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。	0

0	CEN 计数器使能	计数器使能 0: 禁止计数器 1: 开启计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。	0

6.5.2 TIM1 控制寄存器 2(TIM1_CR2)

位	符号	描述	复位值
31:15	--	保留，须为 0	0x0
14	OIS4	输出空闲状态 4（OC4 输出），参见 OIS1 位	0
13	OIS3N	输出空闲状态 3（OC3N 输出），参见 OIS1N 位	0
12	OIS3	输出空闲状态 3 参见 OIS1 位	0
11	OIS2N	输出空闲状态 2（OC2N 输出）参见 OIS1N 位	0
10	OIS2	输出空闲状态 2（OC2 输出）参见 OIS1 位	0
9	OIS1N	输出空闲状态 1（OC1N 输出） 0: 当 MOE=0 时，经过死区时间后 OC1N=0 1: 当 MOE=0 时，经过死区时间后 OC1N=1 注：已经设置了 LOCK(TIM1_BDTR 寄存器)级别 1、2 或 3 后，该位不能被修改。	0
8	OIS1	输出空闲状态 1（OC1 输出） 0: 输出当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0 1: 当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1 注：已经设置了 LOCK(TIM1_BDTR 寄存器)级别 1、2 或 3 后，该位不能被修改。	0
7	TI1S	TI1 选择 0: TIM1_CH1 引脚连接到 TI1 输入 1: TIM1_CH1、TIM1_CH2 和 TIM1_CH3 管脚经异或后连接到 TI1 输入	0
6:4	MMS[1:0]	主模式选择 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: 复位—TIM1_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。 001: 允许 - 计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO)。有时需要在同一时间启动多个定时器或控制从定时器的一个窗口。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIM1_SMCR 寄存器中 MSM 位的描述)。 010: 更新 - 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲—一旦发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即是它已经为高)，触发输出送出一个正脉冲(TRGO) 100: 比较—OC1REF 信号用作触发输出 (TRGO) 101: 比较—OC2REF 信号用作触发输出 (TRGO) 110: 比较—OC3REF 信号用作触发输出 (TRGO) 111: 比较—OC4REF 信号用作触发输出 (TRGO)	0x0
3	CCDS	捕获/比较 DMA 选择 0: 发生 CCx 事件时,发送 CCx DMA 请求 1: 发生更新事件时,发送 CCx DMA 请求	0
2	CCUS	捕获/比较控制更新选择 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们 1: 如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们 注：该位只对具有互补输出的通道起作用。	0

1	--	保留，须为 0	0
0	CCPC	捕获/比较预装载控制 0: CCxE、CCxNE 和 OCxM 位不是预装载的。 1: CCxE、CCxNE 和 OCxM 位是预装载的；设置该位后， 它们只在设置了 COM 位后被更新。 注意：此位仅对具有互补输出的通道有效	0

6.5.3 TIM1 从模式控制寄存器(SMCR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	ETR	外部触发极性 此位可选择将 ETR 还是 ETR 反向用于触发操作 0: ETR 不进行反向，高电平或上升沿有效 1: ETR 反向，低电平或下降沿有效。	0
14	ECE	外部时钟使能，此位可使能外部时钟模式 2 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2。计数器工作在 ETRF 信号的有效沿	0
13:12	ETPS[1:0]	外部触发预分频器 外部触发信号 ETRP 频率不得超过 TIM1CLK 频率的 1/4。预分频器可以被使能，以降低 ETRP 的频率。 当输入快速外部时钟是有效的。 00: 预分频器关闭 01: 2 分频 ETRP 频率 10: 4 分频 ETRP 频率 11: 8 分频 ETRP 频率	0x0
11:8	ETF[3:0]	外部触发滤波。 这些位定义采样 ETRP 信号的频率和应用在 ETRP 的数字滤波长度。这个数字滤波由一个事件计数器组成，在改计数器里，N 个连续的事件被需要使输出的边沿有效 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: $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$ 1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$ 1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$ 1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$ 1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$ 1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$ 1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$ 1111: $f_{SAMPLING}=f_{DTS}/32$, $N=8$	0x0
7	MSM	主/从模式 0: 不执行任何操作 1: 触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的当前定时器和从定时器间的同步(通过 TRGO)。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的	0
6:4	TS[2:0]	触发选择，此位域可选择将要用于同步计数器的触发输入 000: 保留(ITR0) 001: 保留(ITR1) 010: TIM3 (ITR2) 011: TIM17 (ITR3) 100: TI1 边沿检测器 (TI1F_ED) 101: 滤波后的定时器输入 1 (TI1FP1) 110: 滤波后的定时器输入 2 (TI2FP2) 111: 外部触发输入 (ETRF) 注意：这些位只能在未使用的情况下进行更改，以避免转换时出现错误的边沿检测。	0x0
3	OCCS	OCREF 清除选择位。该位用于选择 OCREF 的清除源 0: OCREF_CLR_INT 连接到 OCREF_CLR 输入	0

		1: OCREF_CLR_INT 连接到 ETRF	
2:0	SMS	从模式选择 选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所 选的极性相关（请参见输入控制寄存器和控制寄存器说明）。 000: 关闭从模式 如果 CEN=1，则预分频器直接由内部时钟驱动。 001: 编码器模式 1 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/ 下计数。 010: 编码器模式 2 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/ 下计数。 011: 编码器模式 3 模式 1 和模式 2 的综合。 100: 复位模式 选中的触发输入(TRGI)的上升沿重新初始化计数 器，并且产生一个更新寄存器的信号 101: 门控模式 当触发输入(TRGI)为高时，计数器的时钟开启。一 旦触发输入变为低，则计数器停止(但不复位)。计 数器的启动和停止都是受控的。 110: 触发模式 计数器在触发输入 TRGI 的上升沿启动(但不复 位)，只有计数器的启动是受控的 111: 外部时钟模式 1 选中的触发输入(TRGI)的上升沿驱动计数器 注：如果 TI1F_EN 被选为触发输入(TS=100)时，不要使用门控 模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲， 然而门控模式是要检查触发输入的电平	0x0

表 7-2 TIMx 内部触发连接

从机 TIM	ITRO(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	保留	保留	TIM3	TIM17

6.5.4 TIM1 DMA/中断使能寄存器 (TIM1_DIER)

位	符号	描述	复位值
31:15	--	保留，须为 0	0x0
14	TDE	触发 DMA 请求使能 0: 禁止 DMA 请求 1: 使能 DMA 请求	0
13	COMDE	COM DMA 请求使能 0: 禁止 COM DMA 请求 1: 使能 COM DMA 请求	0
12	CC4DE	捕获/比较 4 DMA 请求使能 0: 禁止 CC4 DMA 请求 1: 使能 CC4 DMA 请求	0
11	CC3DE	捕获/比较 3 DMA 请求使能 0: 禁止 CC3 DMA 请求 1: 使能 CC3 DMA 请求	0
10	CC2DE	捕获/比较 2 DMA 请求使能 0: 禁止 CC2 DMA 请求 1: 使能 CC2 DMA 请求	0
9	CC1DE	捕获/比较 1 DMA 请求使能 0: 禁止 CC1 DMA 请求 1: 使能 CC1 DMA 请求	0
8	UDE	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求	0
7	BIE	刹车中断使能 0: 禁止刹车中断 1: 使能刹车中断	0
6	TIE	触发信号 (TGRI) 中断使能 0: 禁止触发信号 (TGRI) 中断 1: 使能触发信号 (TGRI) 中断	0
5	COMIE	COM 中断使能 0: 禁止 COM 中断 1: 使能 COM 中断	0
4	CC4IE	CC4IE: 捕获/比较 4 中断使能 0: 禁止 CC4 中断 1: 使能 CC4 中断	0
3	CC3IE	CC3IE: 捕获/比较 3 中断使能 0: 禁止 CC3 中断 1: 使能 CC3 中断	0
2	CC2IE	CC2IE: 捕获/比较 2 中断使能 0: 禁止 CC2 中断 1: 使能 CC2 中断	0
1	CC1IE	CC1IE: 捕获/比较 1 中断使能 0: 禁止 CC1 中断 1: 使能 CC1 中断	0
0	UIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断	0

6.5.5 TIM1 状态寄存器 (TIM1_SR)

位	符号	描述	复位值
31:13	--	保留，须为 0	0x0
12	CC4OF	捕获/比较 4 过捕获标志 请参见 CC1OF 说明	0
11	CC3OF	捕获/比较 3 过捕获标志 请参见 CC1OF 说明	0
10	CC2OF	捕获/比较 2 过捕获标志 请参见 CC1OF 说明	0
9	CC1OF	捕获/比较 1 过捕获标志 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“0”可将该位清零 0: 未检测到过捕获。 1: TIM1_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1	0
8	--	保留，须为 0	0
7	BIF	刹车中断标志 只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。 0: 未发生刹车事件 1: 在刹车输入上检测到有效电平。	0
6	TIF	触发器中断标记 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿）时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生 1: 触发器中断等待响应	0
5	COMIF	COM 中断标志 此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。通过软件清零。 0: 未发生 COM 事件 1: COM 中断等待响应	0
4	CC4IF	CC4IF: 捕获/比较 4 中断标志 请参见 CC1IF 说明	0
3	CC3IF	CC3IF: 捕获/比较 3 中断标志 请参见 CC1IF 说明	0
2	CC2IF	CC2IF: 捕获/比较 2 中断标志 请参见 CC1IF 说明	0
1	CC1IF	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，中心对齐模式下除外（请参见 TIM1_CR1 寄存器中的 CMS 位说明）。通过软件清零。 0: 无匹配发生。 1: TIM1_CNT 的值与 TIM1_CCR1 的值匹配 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 TIM1_CCR1 寄存器将该位清零。 0: 未发生输入捕获事件 1: 输入捕获产生并且计数器值已装入 TIM1_CCR1(在 IC1 上检测到与所选极性相同的边沿)	0
0	UIF	更新中断标志 该位在发生更新事件时通过硬件置 1。需要通过软件清零 0: 未发生更新 1: 更新中断等待响应。该位在以下情况下更新寄存器时由硬件置 1 – 若 TIM1_CR1 寄存器的 UDIS=0，当 REP_CNT=0 时产生更新事件(重复向下计数器上溢或下溢时)；	0

		- 若 TIM1_CR1 寄存器的 UDIS=0、URS=0，当 TIM1_EGR 寄存器的 UG=1 时产生更新事件(软件对 CNT 重新初始化); - 若 TIM1_CR1 寄存器的 UDIS=0、URS=0，当 CNT 被触发事件重初始化时产生更新事件。	

6.5.6 TIM1 事件生成寄存器 (TIM1_EGR)

位	符号	描述	复位值
31x:8	--	保留，须为 0	0x0
7	BG	刹车生成 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: 生成刹车事件。MOE 位清零且 BIF 标志置 1。若开启对应的中断和 DMA，则产生相应的中断和 DMA。	0
6	TG	生成触发信号 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: TIM1_SR 寄存器中的 TIF 标志置 1。若开启对应的中断和 DMA，则产生相应的中断和 DMA。	0
5	COMG	捕获/比较控制更新生成 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: CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位 注意：此位仅对具有互补输出的通道有效	0
4	CC4G	CC4G: 捕获/比较 4 生成 请参见 CC1G 说明	0
3	CC3G	CC3G: 捕获/比较 3 生成 请参见 CC1G 说明	0
2	CC2G	CC2G: 捕获/比较 2 生成 请参见 CC1G 说明	0
1	CC1G	CC1G: 捕获/比较 1 生成 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： 若开启对应的中断和 DMA，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果通道 CC1 配置为输入： TIM1_CCR1 寄存器中将捕获到计数器当前值。若开启对应的中断和 DMA，CC1IF 标志置 1 并发送相应的中断 或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1	0
0	UG	更新生成 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，并产生一个更新事件。注意：预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0，若 DIR=1(向下计数)则计数器装载 TIM1_ARR 的值。	0

6.5.7 TIM1 捕获/比较模式寄存器 1 (TIM1_CCMR1)

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

输出比较模式

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	OC2CE	输出比较 2 清零使能，参见 OC1CE 说明	0
14:12	OC2M[2:0]	输出比较 2 模式，参见 OC1M 说明	0x0
11	OC2PE	输出比较 2 预装载使能，参见 OC1PE 说明	0
10	OC2FE	输出比较 2 快速使能，参见 OC1FE 说明	0
9:8	CC2S[1:0]	捕获/比较 2 选择 此位域定义通道方向（输入/输出）以及所使用的输入 00: CC2 通道配置为输出 01: CC2 通道配置为输入，IC2 映射到 TI2 上 10: CC2 通道配置为输入，IC2 映射到 TI1 上 11: CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIM1_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM1_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。	0x0
7	OC1CE	输出比较 1 清零使能 0: OC1REF 不受 ETRF 输入影响 1: ETRF 输入上检测到高电平时，OC1Ref 立即清零	0
6:4	OC1M	这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 000: 冻结。输出比较寄存器 TIM1_CCR1 与计数器 TIM1_CNT 间的比较对 OC1REF 不起作用 001: 匹配时设置通道 1 为有效电平。当计数器 TIM1_CNT 的值与捕获 / 比较寄存器 1(TIM1_CCR1) 相同时，强制 OC1REF 为高 010: 匹配时设置通道 1 为无效电平。当计数器 TIM1_CNT 的值与捕获 / 比较寄存器 1(TIM1_CCR1)相同时，强制 OC1REF 为低 011: 翻转。当 TIM1_CCR1=TIM1_CNT 时，翻转 OC1REF 的电平 100: 强制为无效电平。强制 OC1REF 为低 101: 强制为有效电平。强制 OC1REF 为高 110: PWM 模式 1 – 在向上计数模式下，只要 TIM1_CNT<TIM1_CCR1，通道 1 便为有效电平，否则为无效电平。在向下计数模式下，只要 TIM1_CNT>TIM1_CCR1，通道 1 便为无效电平 (OC1REF=0)，否则为有效电平 (OC1REF=1)。 111: PWM 模式 2 – 在向上计数模式下，只要 TIM1_CNT<TIM1_CCR1，通道 1 便为无效电平，否则为有效电平。在向下计数模式下，只要 TIM1_CNT>TIM1_CCR1，通道 1 便为有效电平，否则为无效电平。 注 1：一旦 LOCK 级别设为 3(TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变	0x0
3	OC1PE	输出比较 1 预装载使能 0: 禁止 TIM1_CCR1 寄存器的预装载功能，可随时写入 TIM1_CCR1 寄存器，且新值马上起作用	0

		1: 开启 TIM1_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1 的预装载值在更新事件到来时被载入当前寄存器中 注 1: 一旦 LOCK 级别设为 3(TIM1_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下, 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定	
2	OC1FE	输出比较 1 快速使能 此位用于加快触发输入事件对 CC 输出的影响 0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OC1FE 的只在通道被配置成 PWM1 或 PWM2 模式时起作用	0
1:0	CC1S	捕获/比较 1 选择 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 上 10: CC1 通道配置为输入, IC1 映射到 TI2 上 11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM1_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM1_CCER 中的 CC1E = “0”), 才可向 CC1S 位写入数据	0x0

输入捕获模式

位	符号	描述	复位值
31:16	--	保留, 须为 0	0x0
15:12	IC2F	输入捕获 2 滤波, 参见 IC1F 说明	0x0
11:10	IC2PSC[1:0]	输入捕获 2 预分频器, 参见 IC1PSC 说明	0x0
9:8	CC2S	CC2S: 捕获/比较 2 选择 此位域定义通道方向 (输入/输出) 以及所使用的输入 00: CC2 通道配置为输出 01: CC2 通道配置为输入, IC2 映射到 TI2 上 10: CC2 通道配置为输入, IC2 映射到 TI1 上 11: CC2 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM1_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM1_CCER 中的 CC2E = “0”), 才可向 CC2S 位写入数据	0x0
7:4	IC1F[3:0]	输入捕获 1 滤波器 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成, 每 N 个事件才视为一个有效沿 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: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8	0x0

		1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=5 1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=6 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=8	
3:2	IC1PSC	IC1PSC: 输入捕获 1 预分频器 此位域定义 CC1 输入 (IC1) 的预分频比。 只要 CC1E=“0” (TIM1_CCER 寄存器), 预分频器便立即复位 00: 无预分频器, 捕获输入上每检测到一个边沿便执行捕获。 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获。 11: 每发生 8 个事件便执行一次捕获。	0x0
1:0	CC1S	捕获/比较 1 选择 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 上 10: CC1 通道配置为输入, IC1 映射到 TI2 上 11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM1_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM1_CCER 中的 CC1E =“0”), 才可向 CC1S 位写入数据	0x0

6.5.8 TIM1 捕获/比较模式寄存器 1 (TIM1_CCMR2)

输出比较模式

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	OC4CE	输出比较 4 清零使能，参见 OC1CE 说明	0x0
14:12	OC4M[2:0]	输出比较 4 模式，参见 OC1M 说明	0x0
11	OC4PE	输出比较 4 预装载使能，参见 OC1PE 说明	0
10	OC4FE	输出比较 4 快速使能，参见 OC1FE 说明	0
9:8	CC4S[1:0]	捕获/比较 4 选择 此位域定义通道方向（输入/输出）以及所使用的输入 00: CC4 通道配置为输出 01: CC4 通道配置为输入，IC4 映射到 TI4 上 10: CC4 通道配置为输入，IC4 映射到 TI3 上 11: CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIM1_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM1_CCER 中的 CC4E=“0”），才可向 CC4S 位写入数据。	0x0
7	OC3CE	输出比较 3 清零使能，参见 OC1CE 说明	0
6:4	OC3M	输出比较 3 模式，参见 OC1M 说明	0x0
3	OC3PE	输出比较 3 预装载使能，参见 OC1PE 说明	0
2	OC3FE	输出比较 3 快速使能，参见 OC1FE 说明	0
1:0	CC3S	捕获/比较 3 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入，IC1 映射到 TI3 上 10: CC3 通道配置为输入，IC1 映射到 TI4 上 11: CC3 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIM1_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM1_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据	0x0

输入捕获模式

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:12	IC4F	输入捕获 4 滤波器，参见 IC1F 说明	0x0
11:10	IC4PSC	输入捕获 4 预分频器，参见 IC1PSC 说明	0x0
9:8	CC4S[1:0]	捕获/比较 4 选择 此位域定义通道方向（输入/输出）以及所使用的输入 00: CC4 通道配置为输出 01: CC4 通道配置为输入，IC4 映射到 TI4 上 10: CC4 通道配置为输入，IC4 映射到 TI3 上 11: CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIM1_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM1_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。	0x0
7:4	IC3F	输入捕获 3 滤波器，参见 IC1F 说明	0x0
3:2	IC3PSC	输入捕获 3 预分频器，参见 IC1PSC 说明	0x0
1:0	CC3S	捕获/比较 3 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入，IC3 映射到 TI3 上 10: CC3 通道配置为输入，IC3 映射到 TI4 上 11: CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIM1_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM1_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据	0x0

6.5.9 TIM1 捕获/比较使能寄存器 (TIM1_CCER)

位	符号	描述	复位值
31:14	--	保留,须为 0	0x0
13	CC4P	捕获/比较 4 输出极性 参见 CC1P 说明	0
12	CC4E	捕获/比较 4 输出使能, 请参见 CC1E 说明	0
11	CC3NP	捕获/比较 3 互补输出极性, 请参见 CC1NP 说明	0
10	CC3NE	捕获/比较 3 互补输出使能, 请参见 CC1NE 说明	0
9	CC3P	捕获/比较 3 输出极性, 请参见 CC1P 说明	0
8	CC3E	捕获/比较 3 输出使能, 请参见 CC1E 说明	0
7	CC2NP	捕获/比较 2 互补输出极性, 请参见 CC1NP 说明	0
6	CC2NE	捕获/比较 2 互补输出使能, 请参见 CC1NE 说明	0
5	CC2P	捕获/比较 2 输出极性, 请参见 CC1P 说明	0
4	CC2E	捕获/比较 2 输出使能, 请参见 CC1E 说明	0
3	CC1NP	捕获/比较 1 互补输出极性 0: OC1N 高电平有效 1: OC1N 低电平有效 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S=00(通道配置为输出)则该位不能被修改。	0
2	CC1NE	捕获 / 比较 1 互补输出使能 0: 关闭— OC1N 禁止输出, 因此 OC1N 的输出电平依赖于 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 位的值 1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 位的值。	0
1	CC1P	捕获/比较 1 输出极性 CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号 0: 不反相: 捕获发生在 IC1 的上升沿; 当用作外部触发器时, IC1 不反相 1: 反相: 捕获发生在 IC1 的下降沿; 当用作外部触发器时, IC1 反相。 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LCCK 位)设为 3 或 2, 则该位不能被修改	0
0	CC1E	捕获 / 比较 1 输出使能 CC1 通道配置为输出: 0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N、CC1NE 位的值。 1: 开启— OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N、CC1NE 位的值。 CC1 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 1 (TIM1_CCR1) 中。 0: 禁止捕获。 1: 使能捕获。	0

注意: 与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。

表 6-3: 具有中断功能的互补 OCX 和 OCXN 通道的输出控制

控制位					输出状态	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	1	0	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	OCxREF+极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		0	1	0	OCxREF+极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	1	1	OCREF+极性+死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+极性+死区 OCxN_EN=1
		1	0	0	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP、OCx_EN=1	OCxREF+极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		1	1	0	OCxREF+极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出使能为无效状态） OCxN=CCxNP、 OCxN_EN=1
		1	1	1	OCREF+极性+死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+极性+死区 OCxN_EN=1
0	0	X	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=OCxNP、 OCxN_EN=0
	0		0	1	禁止输出（不由定时器驱动） 异步：OCx=CCxP、OCxEN=0、OCxN=CCxNP、 OCxN_EN=0 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而 假定 OISx 和 OISxN 在有效状态下与 OCx 和 OCxN 不对应	
	0		1	0		
	0		1	1	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	
	1		0	0		
	1		0	1	关闭状态（输出使能为无效状态） 异步：OCx=CCxP、OCxEN=1、OCxN=CCxNP、 OCxN_EN=1 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而 假定 OISx 和 OISxN 在有效状态下与 OCx 和 OCxN 不对应	
	1		1	0		
	1		1	1		

6.5.10 TIM1 计数器 (TIM1_CNT)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CNT[15:0]	计数器值 (Counter value)	0x0

6.5.11 TIM1 预分频器 (TIM1_PSC)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	PSC[15:0]	预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ PSC 包含了当更新事件产生时装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清 0 或被工作在复位模式的从控制器清 0。	0x0

6.5.12 TIM1 自动重载寄存器 (TIM1_ARR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	ARR[15:0]	自动重载值 ARR 包含了将要装载入实际的自动重载寄存器的值。当自动重载的值为 0 时，计数器不工作。	0xFFFF

6.5.13 TIM1 重复计数器寄存器 (TIM1_RCR)

位	符号	描述	复位值
31:8	--	保留，必须为 0	0x0
7:0	REP[7:0]	周期计数器的值 开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率（即周期性地从预装载寄存器传输到当前寄存器）；如允许产生更新中断，则会同时影响产生更新中断的速率。每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIM1_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在 PWM 模式中，(REP+1)对应着： — 在边沿对齐模式下，PWM 周期的数目； — 在中心对称模式下，PWM 半周期的数目；	

6.5.14 TIM1 捕获/比较寄存器 1 (TIM1_CCR1)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR1[15:0]	捕获/比较 1 的值 若 CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。如果在 TIM1_CCMR1 寄存器(OC1PE 位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较 1 寄存器中。当前捕获/比较寄存器包含了与计数器 TIM1_CNT 比较的值，并且在 OC1 端口上输出信号。 若 CC1 通道配置为输入： CCR1 包含了由上一次输入捕获 1 事件（IC1）传输的计数器值。	0x0

6.5.15 TIM1 捕获/比较寄存器 2 (TIM1_CCR2)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR2[15:0]	捕获/比较 2 的值 若 CC2 通道配置为输出： CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。如果在 TIM1_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较 2 寄存器中。当前捕获/比较寄存器包含了与计数器 TIM1_CNT 比较的值，并且在 OC 端口上输出信号。 若 CC2 通道配置为输入： CCR2 包含了由上一次输入捕获 2 事件（IC2）传输的计数器值。	0x0

6.5.16 TIM1 捕获/比较寄存器 3 (TIM1_CCR3)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR3[15:0]	捕获/比较 3 的值 若 CC3 通道配置为输出： CCR3 包含了装入当前捕获/比较 3 寄存器的值（预装载值）。如果在 TIM1_CCMR3 寄存器(OC3PE 位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较 3 寄存器中。当前捕获/比较寄存器包含了与计数器 TIM1_CNT 比较的值，并且在 OC 端口上输出信号。 若 CC3 通道配置为输入： CCR3 包含了由上一次输入捕获 3 事件（IC3）传输的计数器值。	0x0

6.5.17 TIM1 捕获/比较寄存器 4 (TIM1_CCR4)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR4[15:0]	捕获/比较 4 的值 若 CC4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值（预装载值）。 如果在 TIM1_CCMR4 寄存器(OC4PE 位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较 4 寄存器中。当前捕获/比较寄存器包含了与计数器 TIM1_CNT 比较的值，并且在 OC 端口上输出信号。 若 CC4 通道配置为输入： CCR4 包含了由上一次输入捕获 4 事件（IC4）传输的计数器值	0x0

6.5.18 TIM1 刹车和死区寄存器 (TIM1_BDTR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	MOE	主输出使能 一旦刹车输入有效，该位被硬件异步清 0。 根据 AOE 位的值，可由软件清 0 或自动置 1。它仅对配置为输出通道有效。 0: OC 和 OCN 输出禁止或被强制为空闲状态。 1: 如果 OC 和 OCN 相应输出使能位（TIM1_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，使能 OC 和 OCN 输出。	0
14	AOE	自动输出使能 (Automatic output enable) 0: MOE 只能由软件置 1 1: MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果刹车输入无效) 注意：只要编程了 LOCK（TIM1_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改	0
13	BKP	刹车极性 (Break polarity) 0: 刹车输入为低电平有效 1: 刹车输入为高电平有效 注意：只要编程了 LOCK（TIM1_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。	0
12	BKE	刹车使能 (Break enable) 0: 禁止刹车输入 1: 使能刹车输入 注意：编程了 LOCK（TIM1_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。	0
11	OSSR	运行模式下的关闭状态选择 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明（捕获/比较使能寄存器(TIM1_CCER)） 0: 当定时器不工作时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。 1: 当定时器不工作时，一旦 CCxE 或 CCxNE=1,便使能 OC/OCN 输出并将其设为无效电平，然后设置 OC/OCN 使能输出信号=1 注意：编程了 LOCK（TIM1_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。	0
10	OSSI	空闲模式下的关闭状态选择 该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说明（捕获/比较使能寄存器(TIM1_CCER)）	0

		0: 当定时器不工作时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0） 1: 当定时器不工作时，一旦 CcxNE=1 或 CcxNE=1，OC/OCN 首先输出其空闲电平。OC/OCN 使能输出信号=1 注意：编程了 LOCK（TIM1_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。	
9:8	LOCK	锁定配置 (Lock configuration) 这些位用于针对软件错误提供写保护。 00: 锁定关闭，寄存器无写保护。 01: 锁定级别 1，不能写入 TIM1_BDTR 寄存器的 DTG/BKE/BKP/AOE 位、TIM1_CR2 寄存器的 OISx/OISxN 位。 10: 锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位（一旦相关通道通过 CCxS 位设为输出，TIM1_CCER 寄存器的 CCxP/CCNxP 位）以及 OSSR/OSSI 位 11: 锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位（一旦相关通道通过 CCxS 位设为输出，TIM1_CCMRx 寄存器的 OCxM/OCxPE 位）。 注意：注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIM1_BDTR 寄存器，则其内容冻结直至复位。	0x0
7:0	DTG	配置死区发生器 (Dead-time generator setup) 此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应 DTG[7:5]=0xx => DT=DTG[7:0]*t_{dtg}，其中 t_{dtg}=t_{DTS}。DTG[7:5]=10x => DT=(64+DTG[5:0])*t_{dtg}，其中 T_{dtg}=2xt_{DTS}。 DTG[7:5]=110 => DT=(32+DTG[4:0])*t_{dtg}，其中 T_{dtg}=8xt_{DTS}。 DTG[7:5]=111 => DT=(32+DTG[4:0])*t_{dtg}，其中 T_{dtg}=16xt_{DTS}。 示例：如果 TDTS=125ns (8MHz)，则可能的死区值为： 0 到 15875 ns（步长为 125 ns）， 16 us 到 31750 ns（步长为 250 ns）， 32 us 到 63us（步长为 1 us）， 64 us 到 126 us（步长为 2 us） 注意：只要编程了 LOCK（TIM1_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位域即无法修改	0x0

6.5.19 TIM1 DMA 控制寄存器 (TIM1_DCR)

位	符号	描述	复位值
31:13	-	保留	0x0
12:8	DBL	DMA 连续传送长度 (DMA burst length) 该 5 位定义了 DMA 在连续模式下的传送长度（当对 TIM1_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 00011: 4 次传输 10001: 18 次传输	0x0
7:5	-	保留	0x0
4:0	DBA	DMA 基址 (DMA base address) 该 5 位向量定义 DMA 传输的基址（通过 TIM1_DMAR 地址进行读/写访问时）。DBA 定义为从 TIM1_CR1 寄存器地址开始计算的偏移量。	0x0

6.5.20 TIM1 全传输 DMA 地址 (TIM1_DMAR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	DMAB	DMA 连续传送寄存器 对 TIM1_DMAR 寄存器执行读或写操作将访问如下地址的寄存器： TIM1_CR1 地址 +(DBA + DMA 引索)*4，其中： “TIM1_CR1 地址”是控制寄存器 1 的地址； “DBA”是 TIM1_DCR 寄存器中定义的 DMA 基地址； “DMA 引索”由 DMA 传输自动控制，其范围介于 0 到 DBL(TIM1_DCR 寄存器中定义的 DBL)之间。	0x0

第7章 通用定时器 TIM3

7.1 TIM3 简介

通用定时器包含一个 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）。

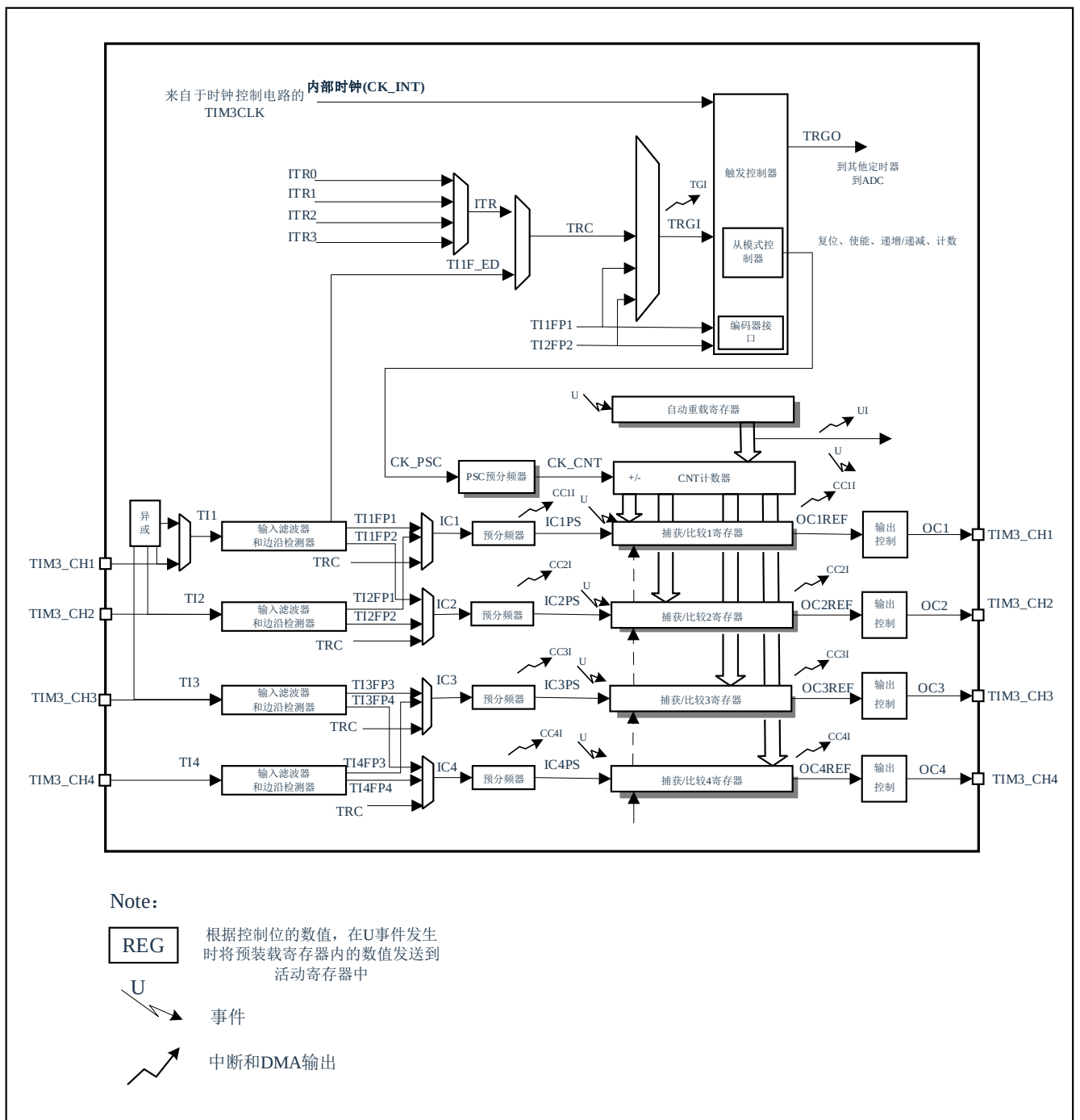
使用定时器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

7.2 TIM3 主要特性

通用 TIM3 定时器具有以下特性：

- 16 位 TIM3 向上、向下、向上向下自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 到 65536 之间。
- 多达 4 个独立通道，可用于：
 - 输入捕获
 - 输出比较
 - PWM 生成（边沿和中心对齐模式）
 - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 发生如下事件时生成中断/DMA 请求：
 - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
 - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
 - 输入捕获
 - 输出比较
- 支持增量（正交）编码器和霍尔传感器电路
- 外部时钟触发输入或逐周期电流管理

图 7-1 通用定时器 3 框图



7.3 TIM3 功能说明

7.3.1 时基单元

可编程定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。此计数器可向上计数、向下计数或向上向下双向计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIM3_CNT)
- 预分频器寄存器 (TIM3_PSC)
- 自动重载寄存器 (TIM3_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，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 TIM3_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（或者在递减计数时达到下溢值）并且 TIM3_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIM3_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，真正的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器可对计数器时钟频率进行分频，分频系数介于 1 到 65536 之间。该预分频器基于 16 位寄存器（TIM3_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 7-2 预分频器分频由 1 变为 2 时的计数器时序图

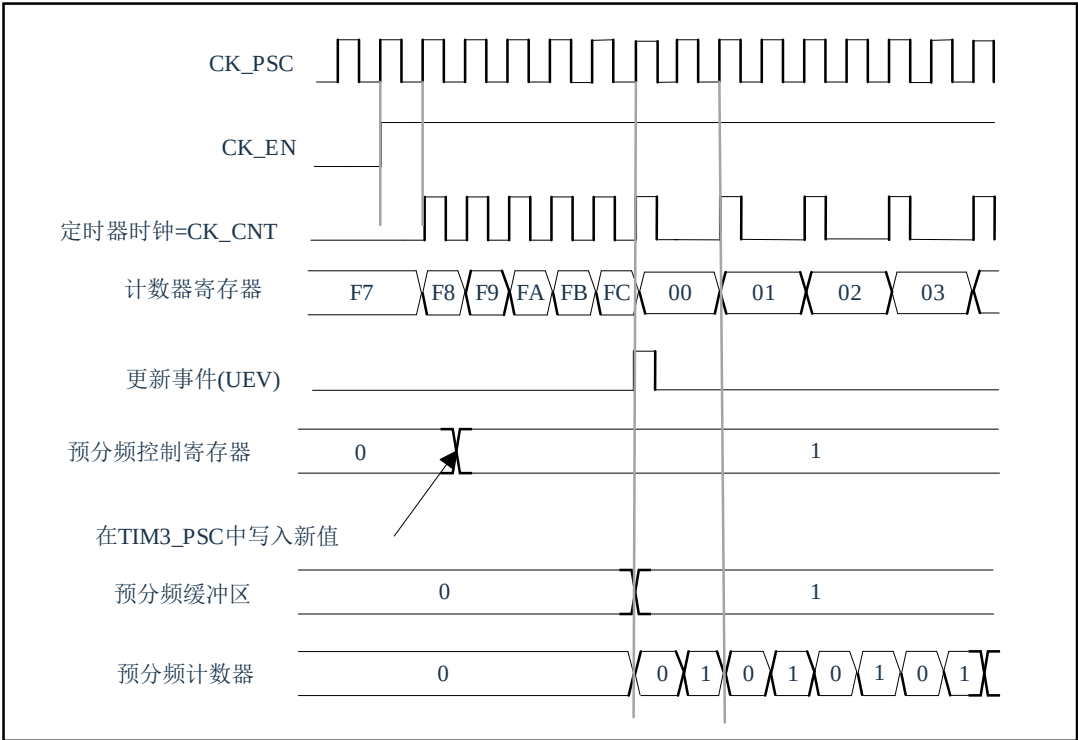
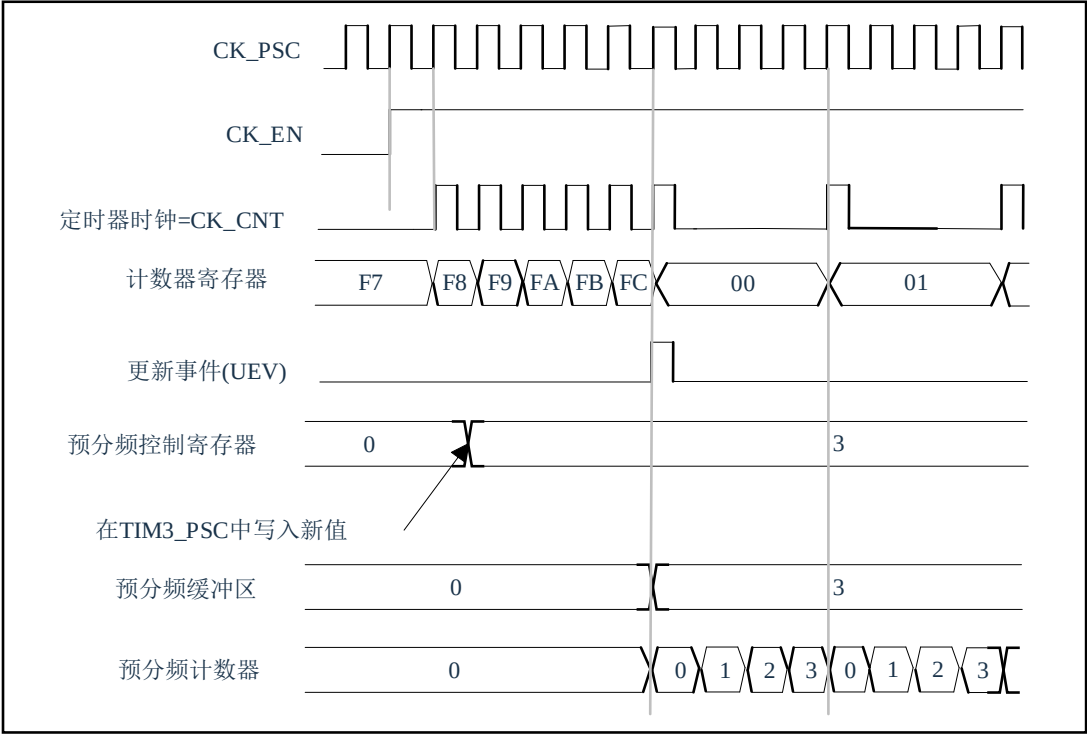


图 7-3 预分频器分频由 1 变为 4 时的计数器时序图



7.3.2 计数器模式

7.3.2.1 向上计数模式

在向上计数模式下，计数器从 0 计数到自动重载值（TIM3_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 TIM3_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIM3_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，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）。此外，如果 TIM3_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM3_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIM3_PSC 寄存器的内容）
- 自动重载影子寄存器将以预装载值进行更新

以下各图以一些示例说明当 TIM3_ARR=0x36 时不同时钟频率下计数器的行为。

图 7-4 计数器时序图，1 分频内部时钟

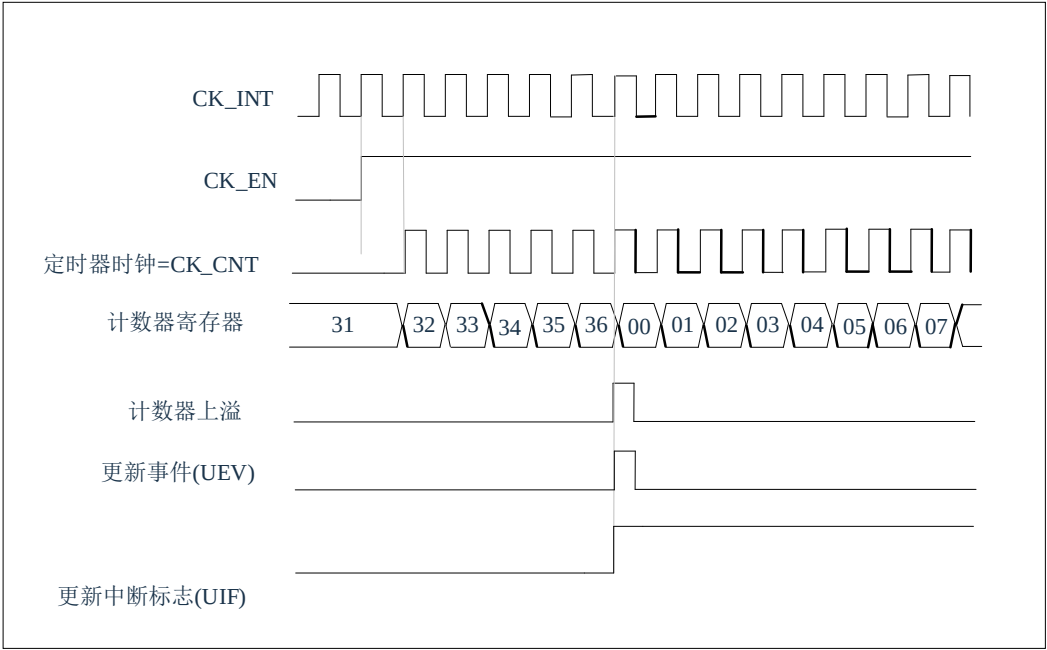


图 7-5 计数器时序图，2 分频内部时钟

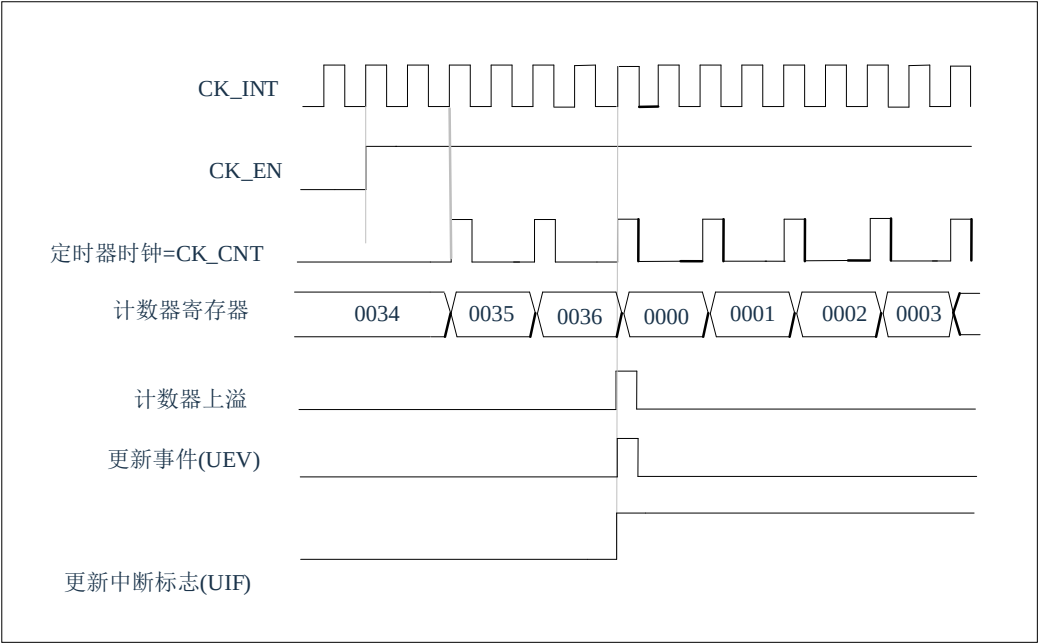


图 7-6 计数器时序图，4 分频内部时钟

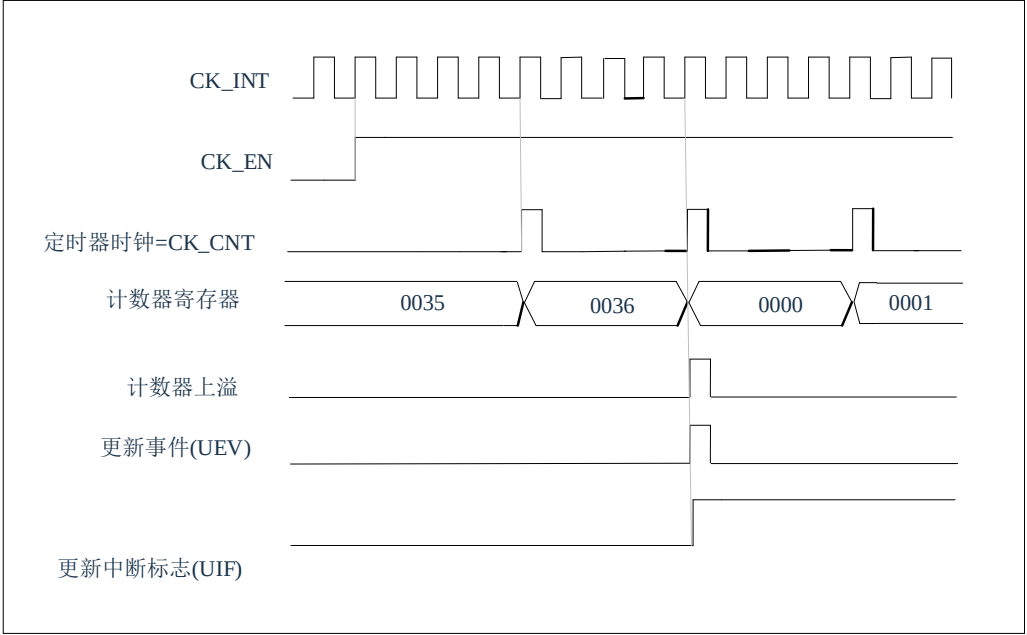


图 7-7 计数器时序图，N 分频内部时钟

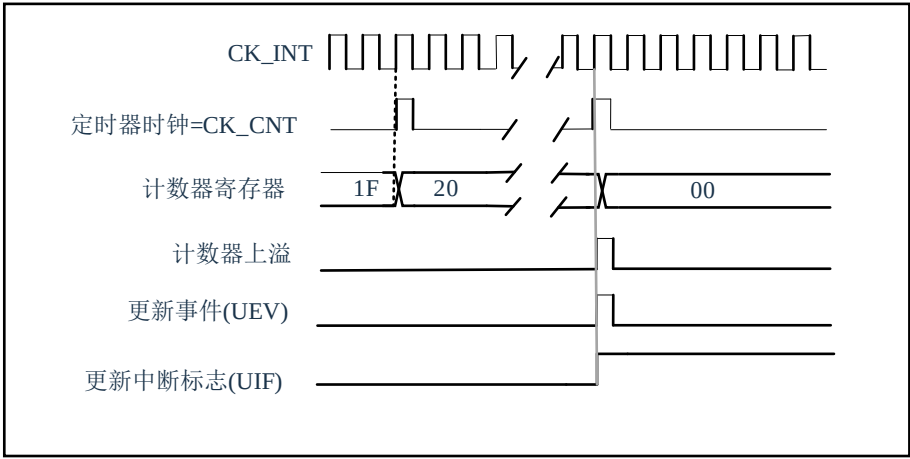


图 7-8 计数器时序图，ARPE=0 时更新事件（TIM3_ARR 未预装载）

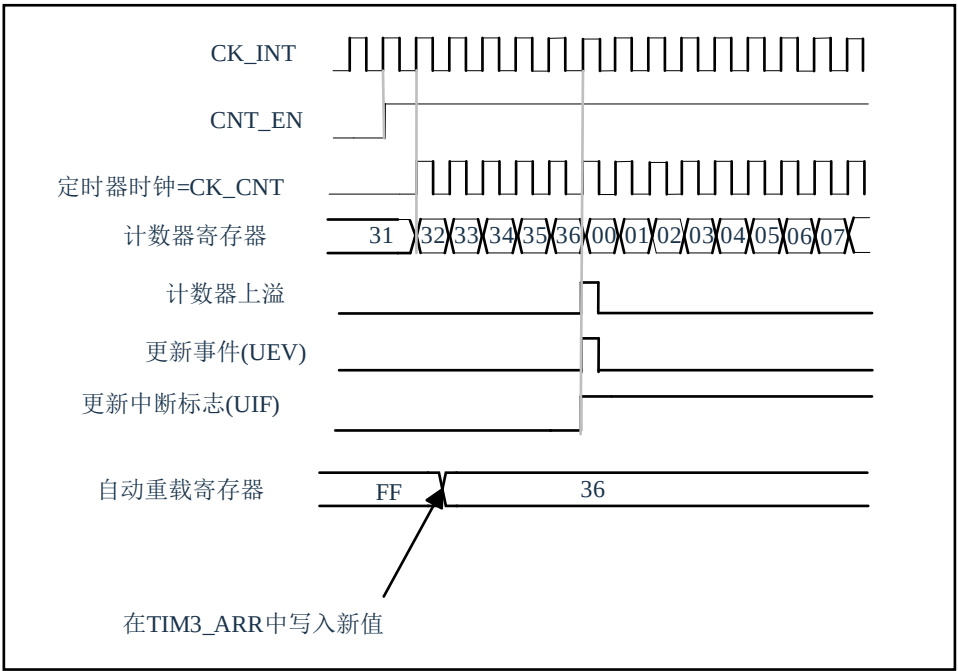
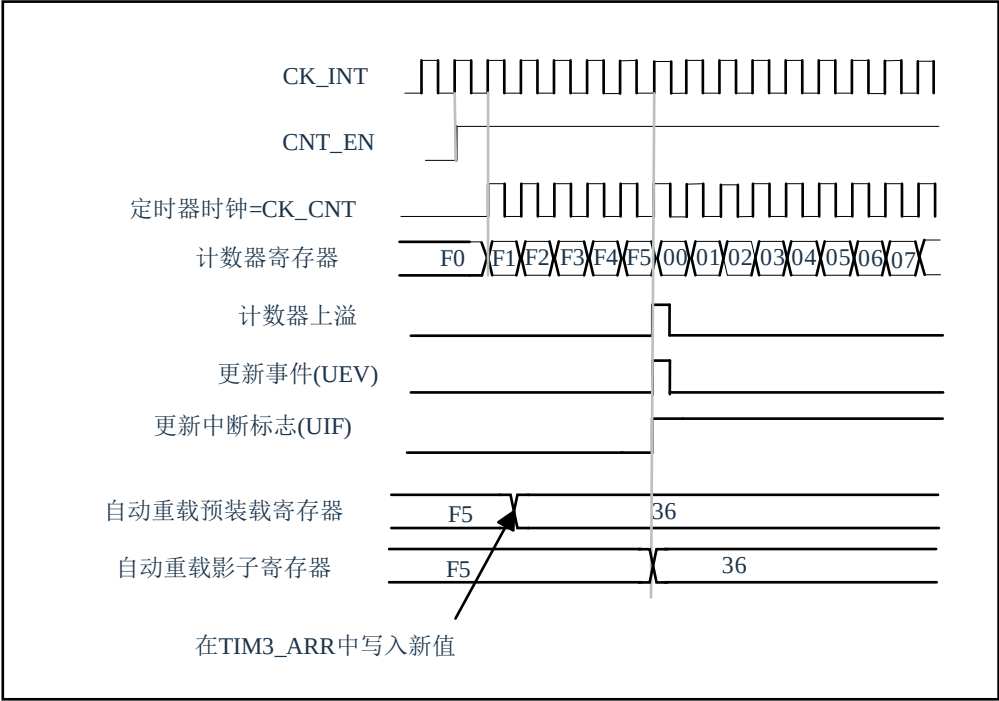


图 7-9 计数器时序图，ARPE=1 时更新事件（TIM3_ARR 已预装载）



7.3.2.2 向下计数模式

在向下计数模式下，计数器从自动重载值开始向下计数到 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。每次发生计数器下溢时会生成更新事件，或将 TIM3_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIM3_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIM3_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM3_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIM3_PSC 寄存器的内容）。
- 自动重载活动寄存器将以预装载值（TIM3_ARR 寄存器的内容）进行更新。

注意：自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度。

图 7-10 计数器时序图，1 分频内部时钟

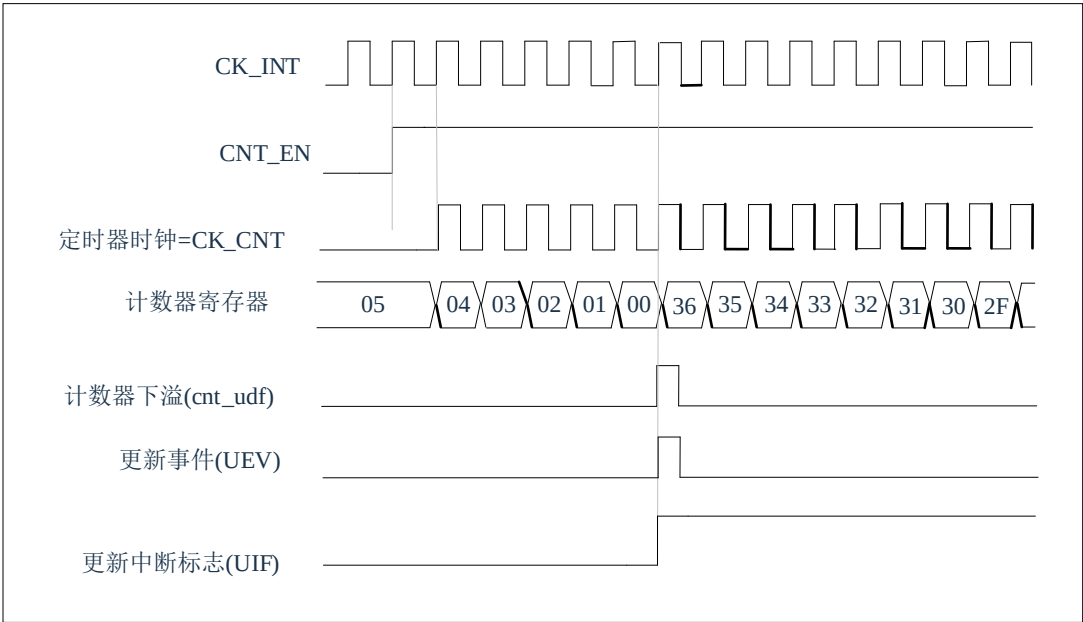


图 7-11 计数器时序图，2 分频内部时钟

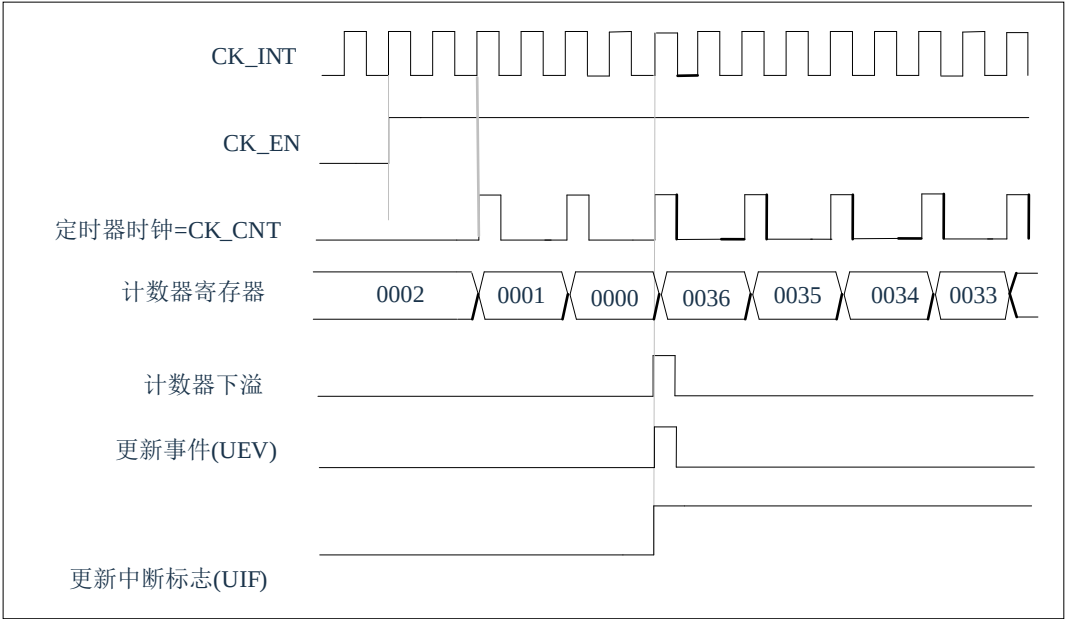


图 7-12 计数器时序图，4 分频内部时钟

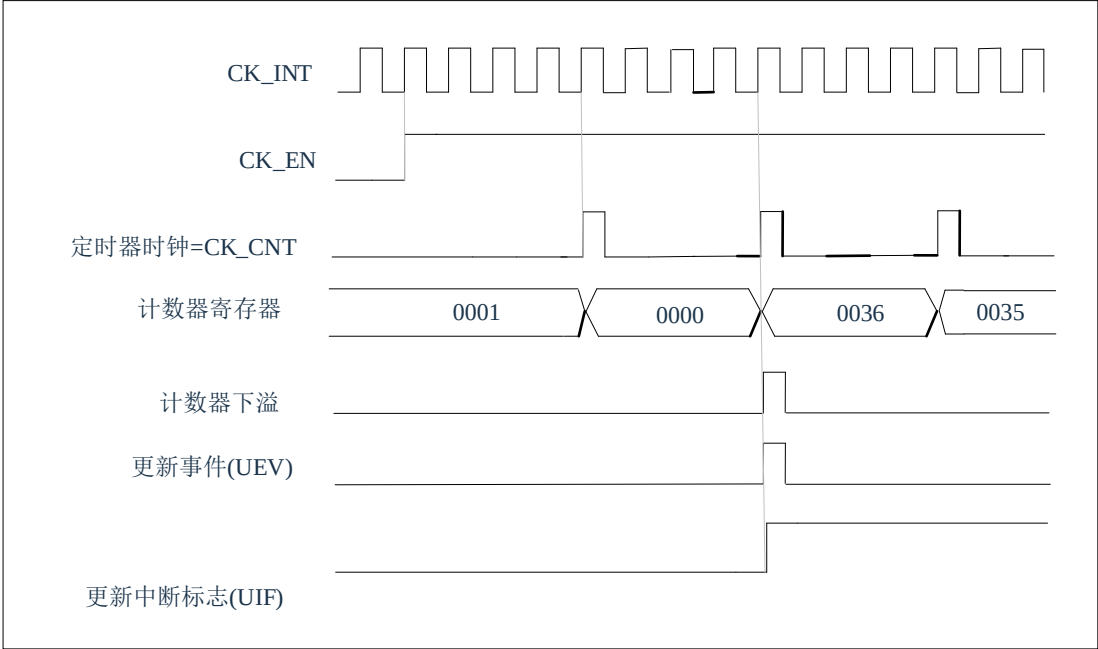


图 7-13 计数器时序图，N 分频内部时钟

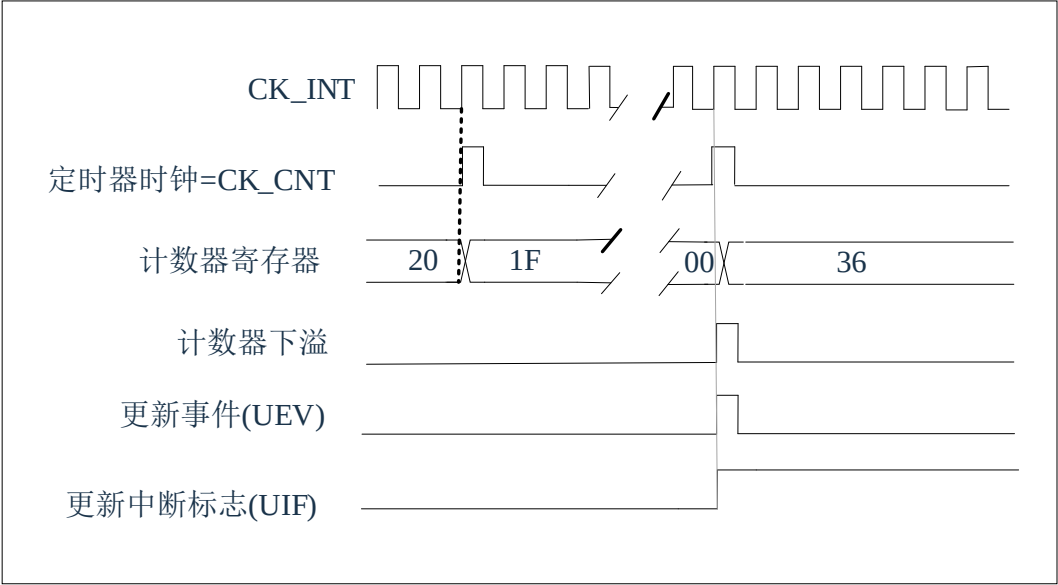
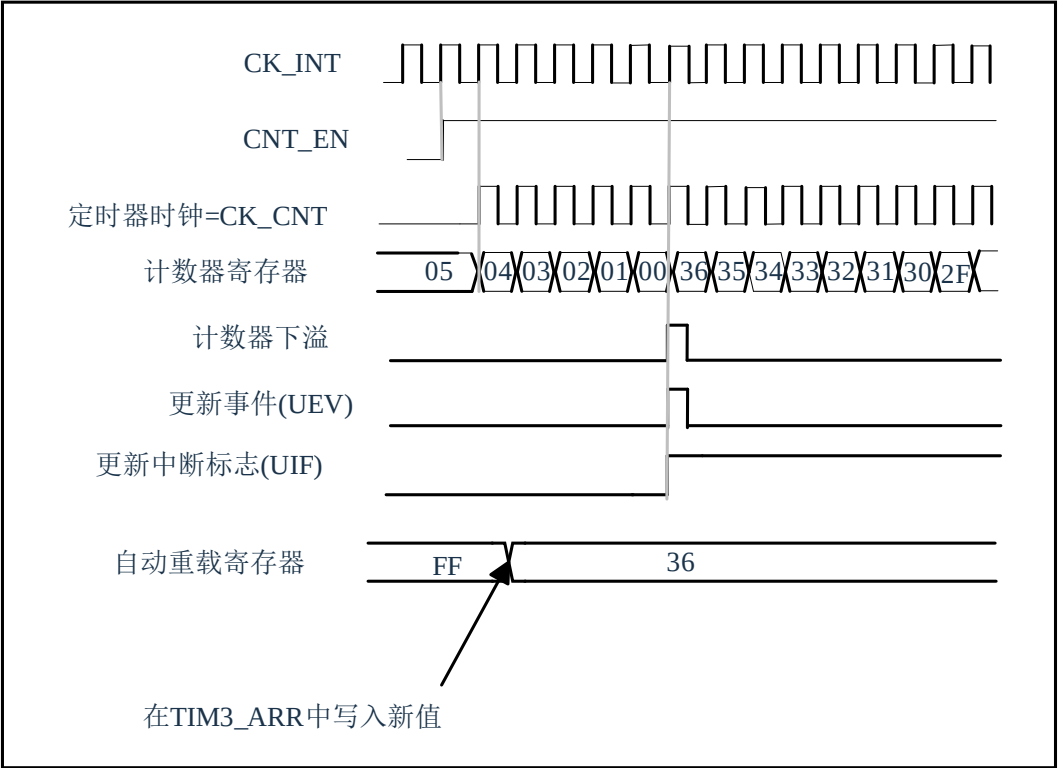


图 7-14 计数器时序图，更新事件



7.3.2.3 中心对齐模式（向上/向下计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIM3_ARR 寄存器的内容）减 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIM3_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS =“01”）、计数器递增计数（中心对齐模式 2，CMS =“10”）以及计数器递增/递减计数（中心对齐模式 3，CMS =“11”）。

此模式下无法写入方向位（TIM3_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIM3_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIM3_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM3_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

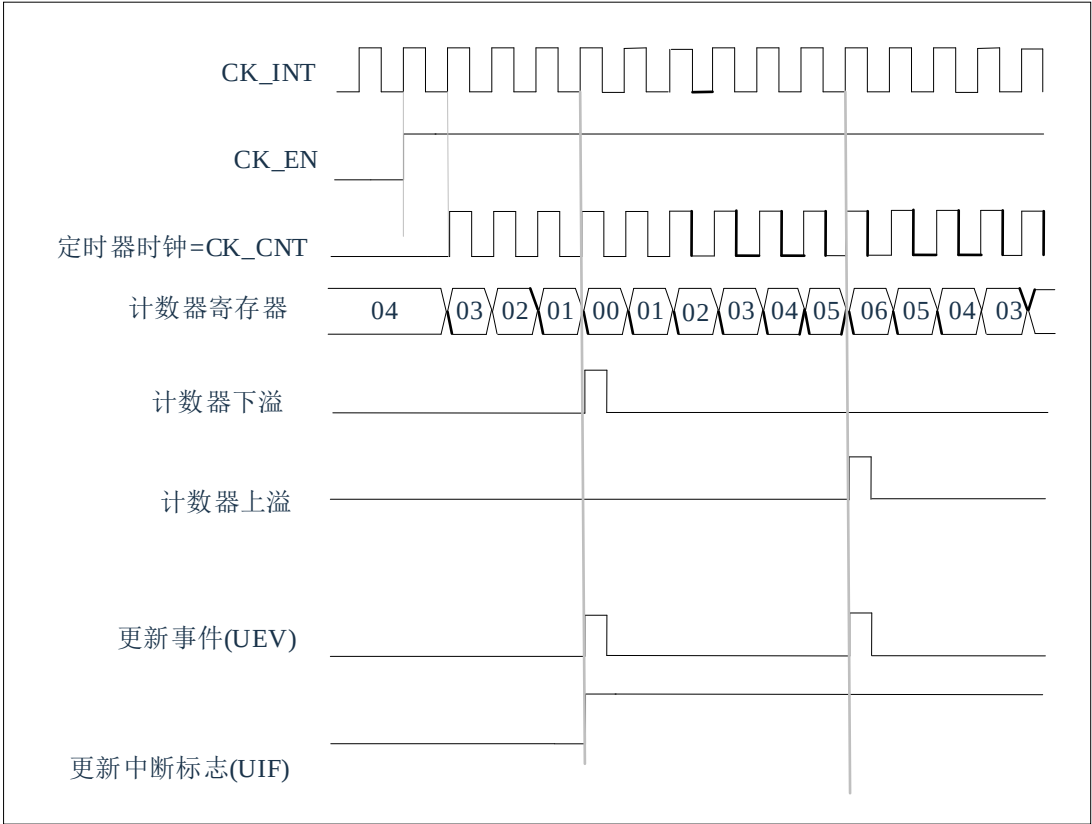
发生更新事件时，将更新所有寄存器且将更新标志（TIM3_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIM3_PSC 寄存器的内容）。
- 自动重载活动寄存器将以预装载值（TIM3_ARR 寄存器的内容）进行更新。

注意：如果更新操作是由计数器上溢触发的，则自动重载寄存器在重载计数器之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 7-15 计数器时序图，1 分频内部时钟，TIM3_ARR=0x6



注：此处使用了中心对齐模式 1。

图 7-16 计数器时序图，2 分频内部时钟

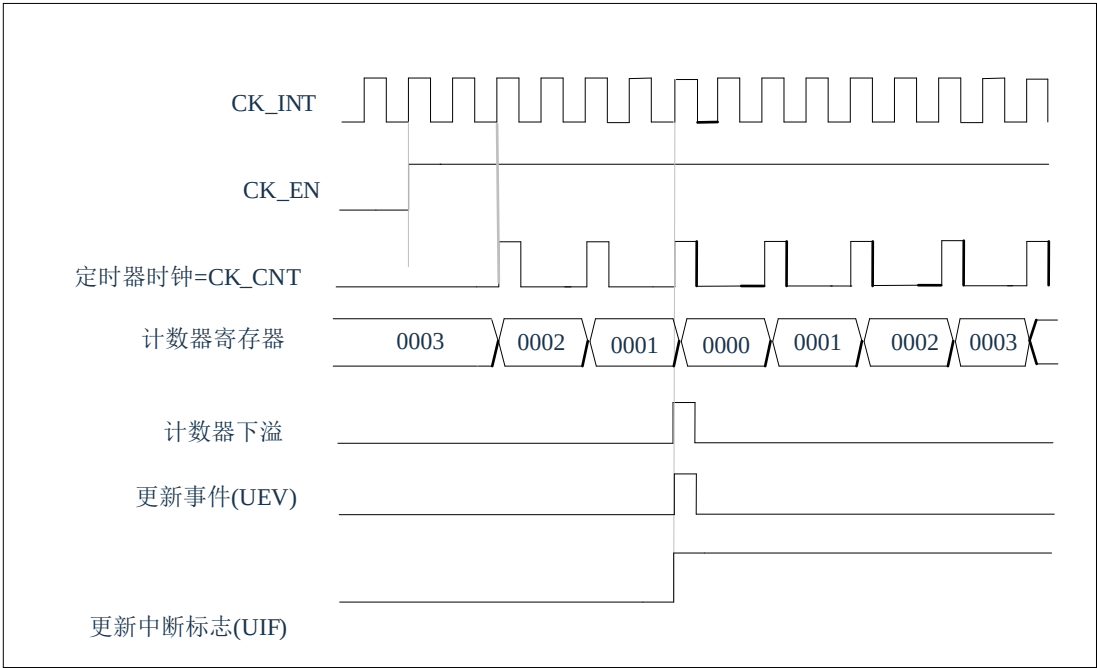
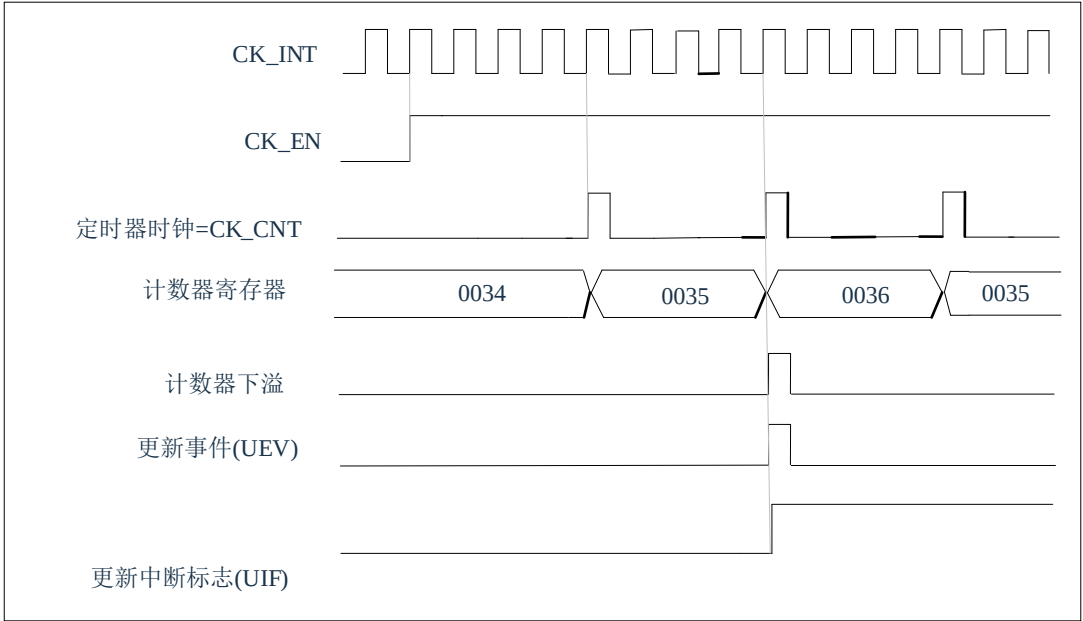


图 7-17 计数器时序图，4 分频内部时钟，TIM3_ARR=0x36



注：此处使用了中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 7-18 计数器时序图，N 分频内部时钟

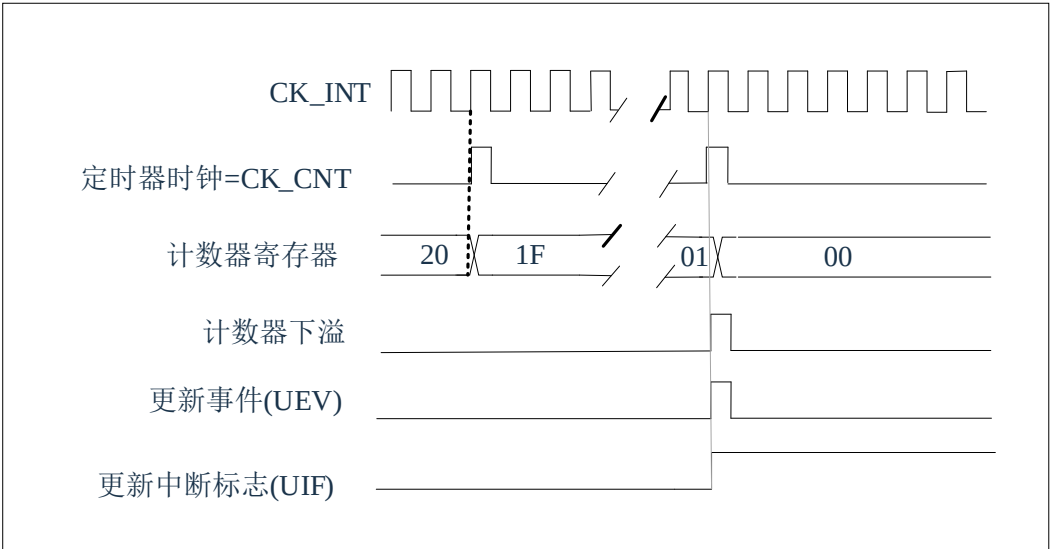


图 7-19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

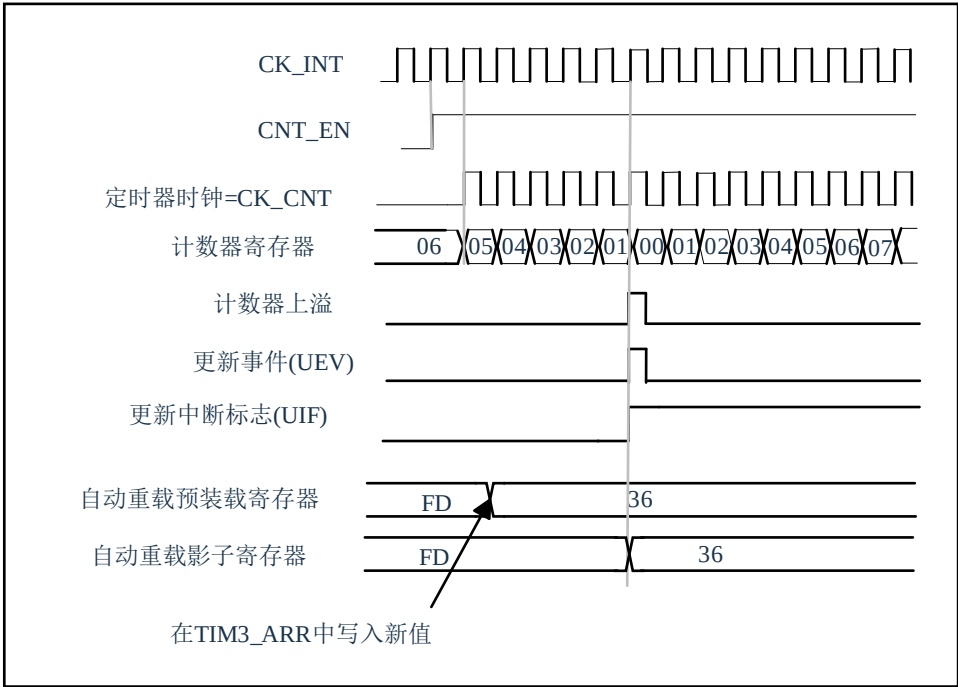
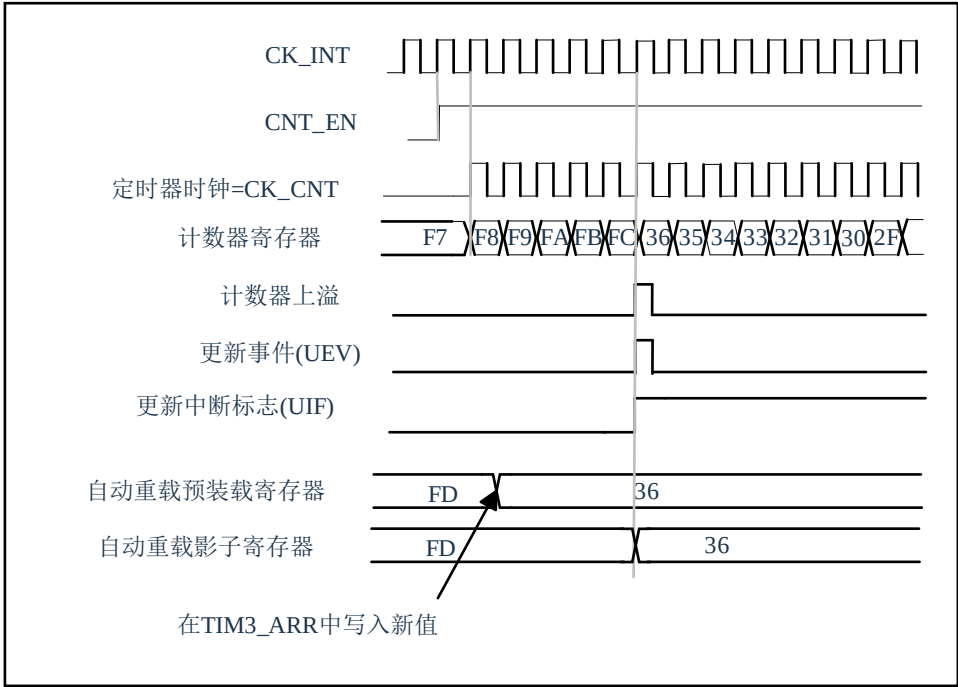


图 7-20 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



7.3.3 时钟选择

计数器时钟可由下列时钟源提供：

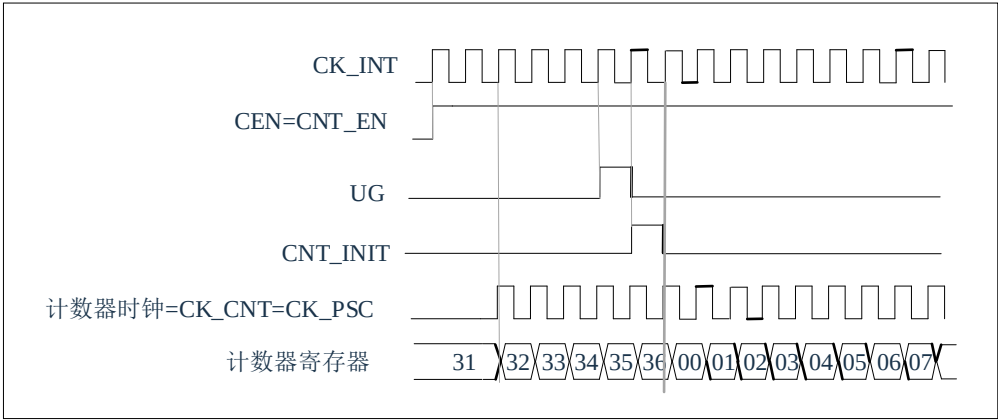
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器 1 置为定时器 3 预分频器。

7.3.3.1 内部时钟源 (CK_INT)

如果禁止从模式控制器，则 CEN 位、DIR 位（TIM3_CR1 寄存器中）和 UG 位（TIM3_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 7-21 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

图 7-21 正常模式下的控制电路，1 分频内部时钟



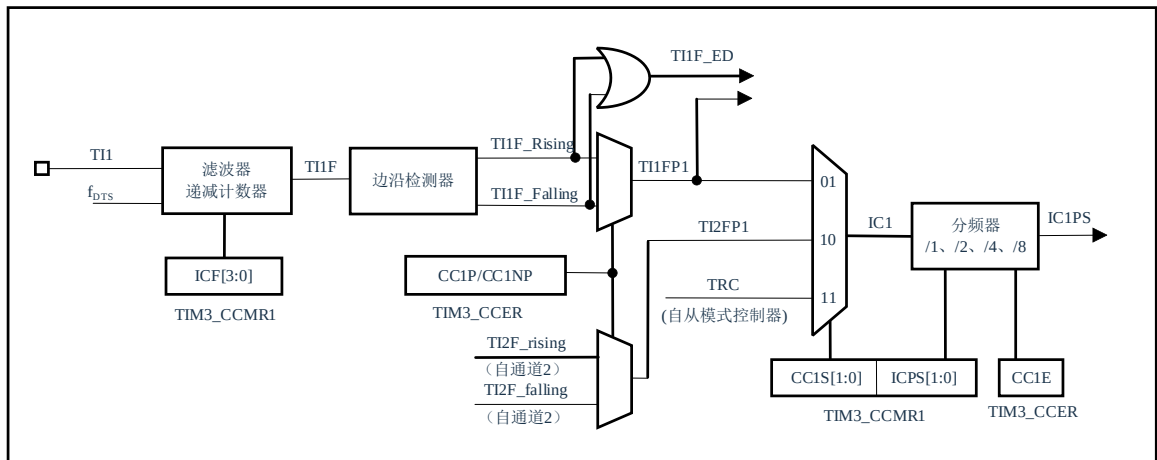
7.3.4 捕获/比较通道

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。

下图概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF 。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，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 (ICxPS)，而后再进入捕获寄存器。

图 7-24 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 7-25 捕获/比较通道 1 主电路

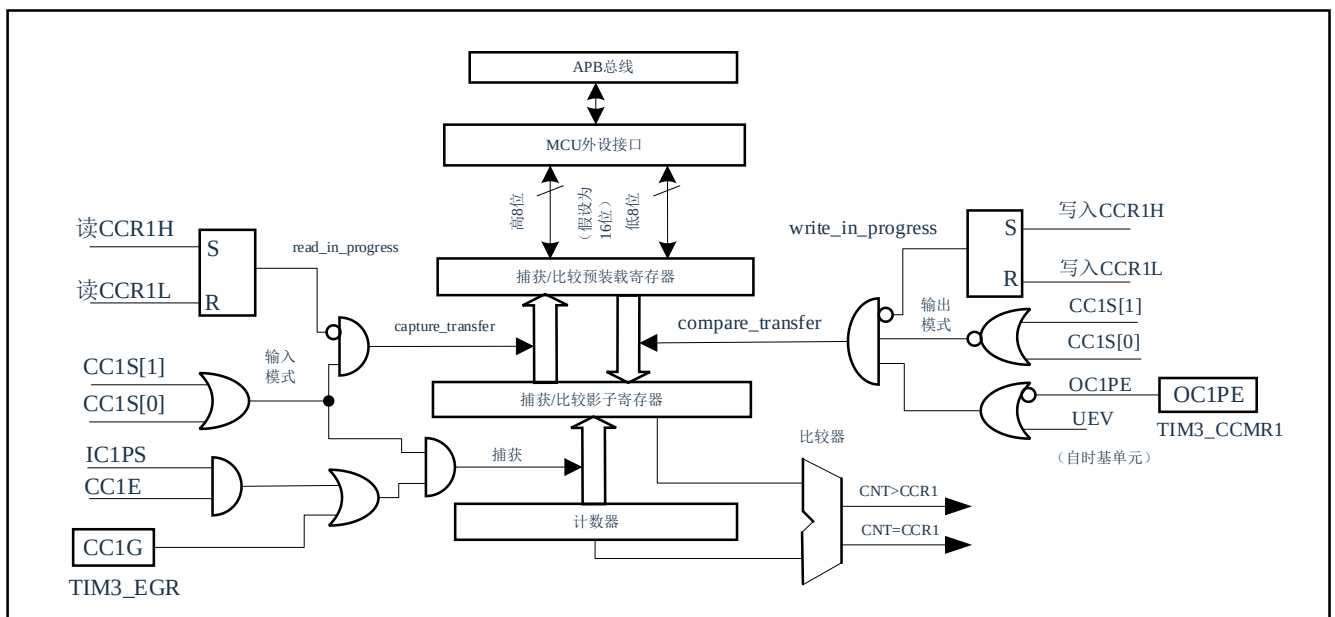
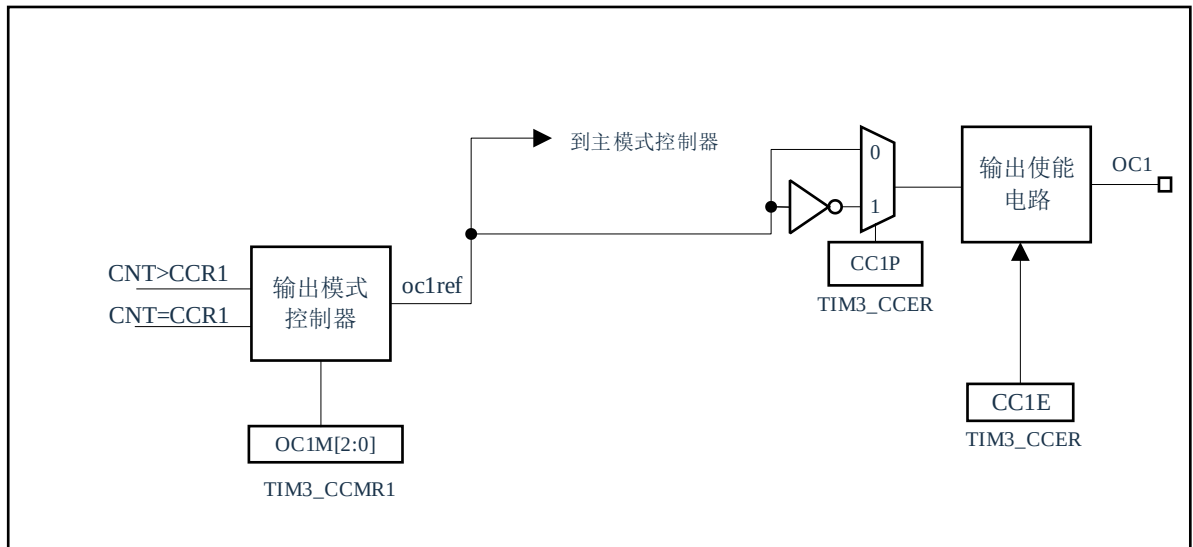


图 7-26 捕获/比较通道的输出阶段（通道 1）



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。

7.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器来锁存计数器的值。发生捕获事件时，会将相应的 CCxIF 标志（TIM3_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIM3_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIM3_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIM3_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIM3_CCR1 必须连接到 TI1 输入，因此向 TIM3_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIM3_CCR1 寄存器将处于只读状态。
- 根据连接到定时器的信号，配置输入滤波器为所需要的带宽（即输入为 TIx 时，输入滤波器控制位是 TIM3_CCMRx 寄存器中的 ICxF 位）。假设信号变化时，输入信号最多在 5 个内部时钟周期内发生抖动。我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIM3_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过向 TIM3_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿（本例中为上升沿）。
- 配置输入预分频器。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIM3_CCMR1 寄存器中的 IC1PS 位写入 00）。
- 通过将 TIM3_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIM3_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIM3_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

为了处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

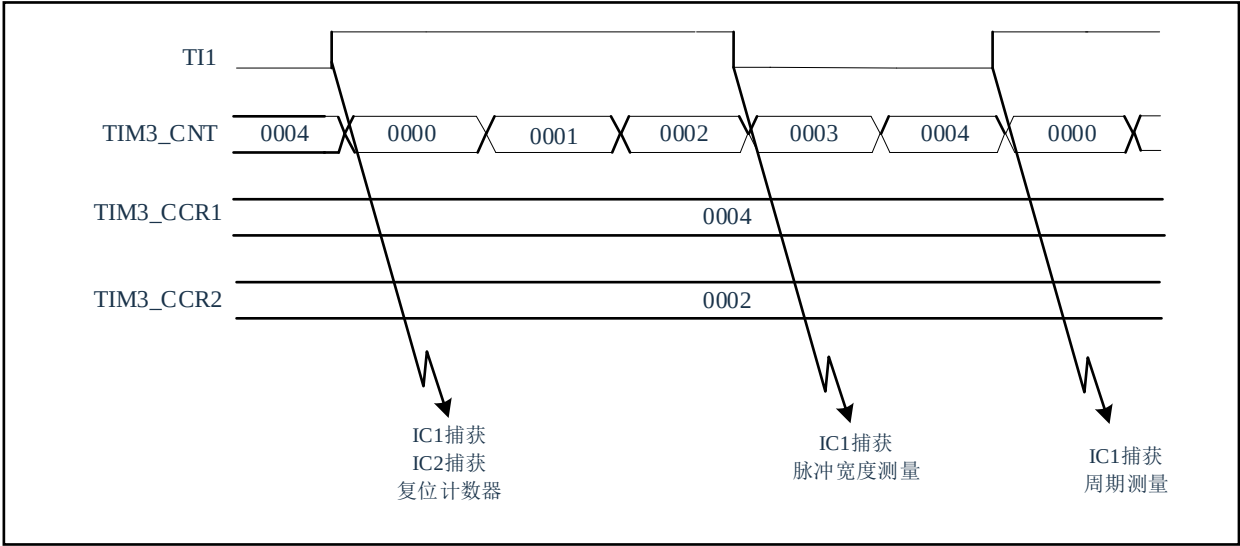
注意：通过软件将 TIM3_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。

7.3.6 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。
例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIM3_CCR1 寄存器中）和占空比（位于 TIM3_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：
- 选择 TIM3_CCR1 的有效输入：向 TIM3_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于 TIM3_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 选择 TIM3_CCR2 的有效输入：向 TIM3_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于 TIM3_CCR2 中的捕获）：向 CC2P 位写入“1”（下降沿有效）。
- 选择有效触发输入：向 TIM3_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIM3_SMCR 寄存器中的 SMS 位写入 100。
- 使能捕获：向 TIM3_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 7-27 PWM 输入模式时序



7.3.7 强制输出模式

在输出模式（TIM3_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）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 (OCxREF/OCx) 强制设置为有效电平，只需向相应 TIM3_CCMRx 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIM3_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。无论如何，TIMx_CCRx 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和 DMA 请求。输出比较模式一节对此进行了介绍。

7.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一段时间。当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将输出比较模式（TIM3_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM3_CCER 寄存器中的 CCxP 位）定义的值输出到对应引脚上。在比较匹配时，输出引脚既可保持其电平 (OCxM=000)，也可设置为有效电平 (OCxM=001)、无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIM3_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIM3_DIER 寄存器中的 CCxIE 位）置 1，将生成中断。
- 如果相应 DMA 使能位（TIM3_DIER 寄存器的 CCxDE 位，TIM3_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIM3_CCMRx 寄存器中的 OCxPE 位，可将 TIM3_CCRx 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 OCxREF 和 OCx 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）。

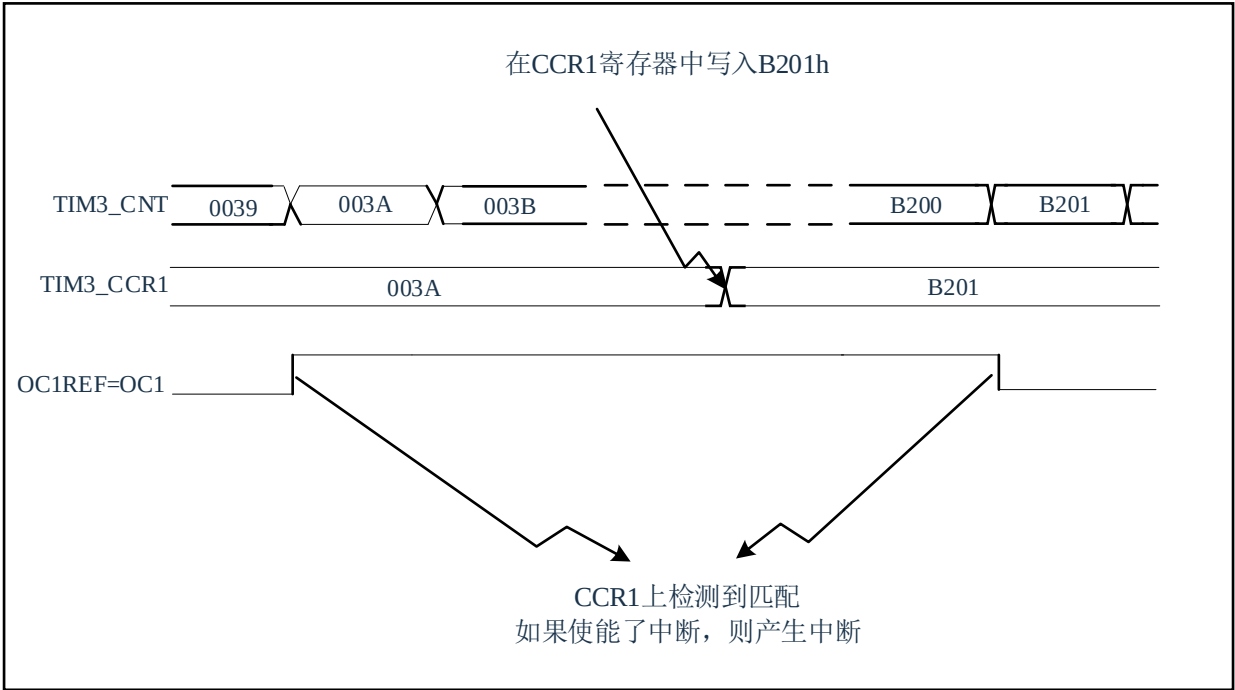
步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIM3_ARR 和 TIM3_CCRx 寄存器中写入所需数据。
3. 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，
 - 要求 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
 - 置 OCxPE=0 禁用预装载寄存器
 - 置 CCxP=0 选择极性为高电平有效
 - 置 CCxE=1 使能输出
5. 通过将 TIM3_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIM3_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIM3_CCRx 影子寄存器）。

图 7-28 列出了相关示例。

图 7-28 输出比较模式，翻转 OC1



7.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIM3_ARR 寄存器值决定，其占空比则由 TIM3_CCRx 寄存器值决定。

通过向 TIM3_CCMRx 寄存器中的 OCxM 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 TIM3_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIM3_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

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 TIM3_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可使用 TIM3_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。OCx 输出通过将 TIM3_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIM3_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIM3_CNT 始终与 TIM3_CCRx 进行比较，以确定是 $TIM3\_CCR_x \leq TIM3\_CNT$ 还是 $TIM3\_CNT \leq TIM3\_CCR_x$ （取决于计数器计数方向）。不过，为了与 ETRF 相符（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF），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（TIM3_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“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。

根据 TIM3_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

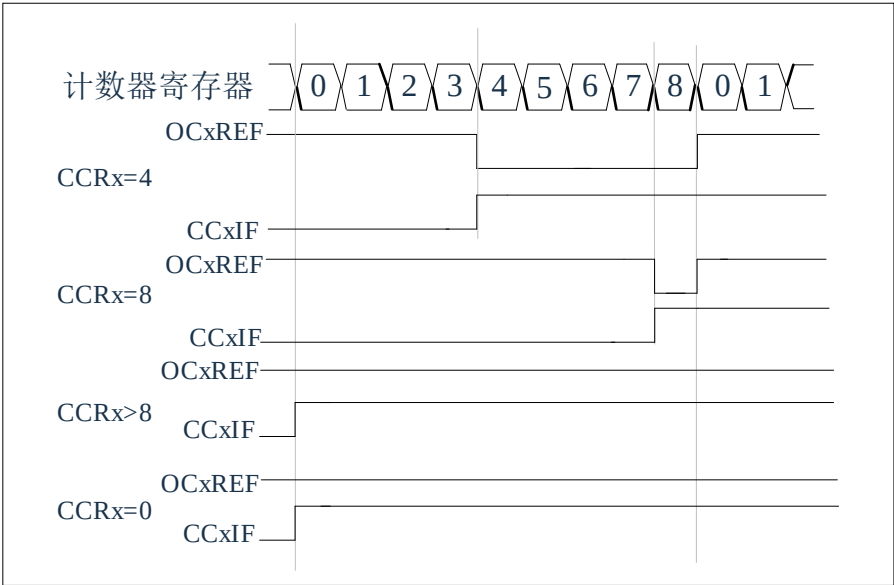
7.3.9.1 PWM 边沿对齐模式

向上计数配置

当 TIM3_CR1 寄存器中的 DIR 位为低时执行向上计数。

以下以 PWM 模式 1 为例。只要 TIM3_CNT < TIM3_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIM3_CCRx 中的比较值大于自动重载值（TIM3_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。图 7-29 举例介绍边沿对齐模式的一些 PWM 波形 (TIM3_ARR=8)。

图 7-29 边沿对齐模式 PWM 波形 (TIM3_ARR=8)



向下计数配置

当 TIM3_CR1 寄存器中的 DIR 位为高时执行向上计数。

在 PWM 模式 1 下，只要 TIM3_CNT > TIMx_CCRx，参考信号 OCxREF 便为低电平，否则为高电平。如果 TIM3_CCRx 中的比较值大于 TIM3_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

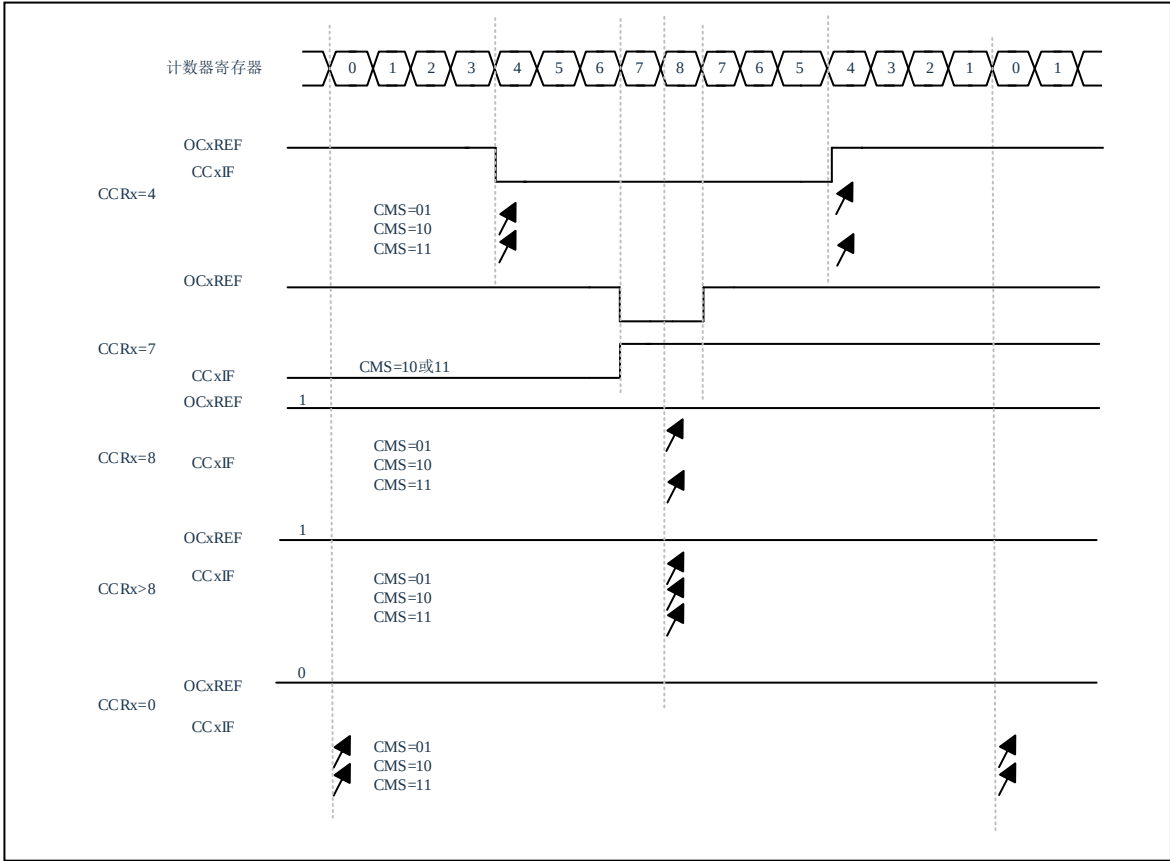
7.3.9.2 PWM 中心对齐模式

当 TIM3_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 OCxREF/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器向上计数、向下计数或同时向上和向下计数时将比较标志置 1。TIM3_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。

图 7-30 显示了中心对齐模式的 PWM 波形，在此例中：

- TIM3_ARR=8,
- PWM 模式为 PWM 模式 1,
- 在根据 TIM3_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 7-30 中心对齐模式 PWM 波形 (TIM3_ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的向上/向下计数配置。这意味着计数器将根据写入 TIM3_CR1 寄存器中 DIR 位的值进行向上或向下计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
 - 如果写入计数器中的值大于自动重载值 (TIM3_CNT>TIM3_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
 - 如果向计数器写入 0 或 TIM3_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIM3_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

7.3.10 单脉冲模式

单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIM3_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 向上计数模式下： $CNT < CCRx \leq ARR$ （特别注意， $0 < CCRx$ ）
- 向下计数模式下： $CNT > CCRx$

单脉冲模式示例

例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 TIM3_CCMR1 寄存器中写入 $CC2S=01$ ，将 TI2FP2 映射到 TI2。
- 在 TIM3_CCER 寄存器中写入 $CC2P=0$ 和 $CC2NP="0"$ ，使 TI2FP2 能够检测上升沿。
- 在 TIM3_SMCR 寄存器中写入 $TS=110$ ，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 在 TIM3_SMCR 寄存器中写入 $SMS="110"$ （触发模式），使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIM3_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值 ($TIM3\_ARR - TIM3\_CCR1$) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIM3_CCMR1 寄存器中写入 $OC1M=111$ ，以使能 PWM 模式 2。如果需要，可选择在 TIM3_CCMR1 寄存器的 OC1PE 和 TIM3_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIM3_CCR1 寄存器中写入比较值并在 TIM3_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIM3_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM3_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。

特例：OCx 快速使能：

在单脉冲模式下，在 TIx 输入的边沿检测逻辑设置 CEN 位以启动计数器。然后，在计数器值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 TIM3_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF（和 OCx）对激励信号做出响应，而不再考虑比较的结果。输出的波形与比较匹配时的波形一样。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

7.3.11 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 TIM3_CCMRx 寄存器中的 OCxCE 使能位 置“1”），可 使 OCxREF 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更 新事件 (UEV)。

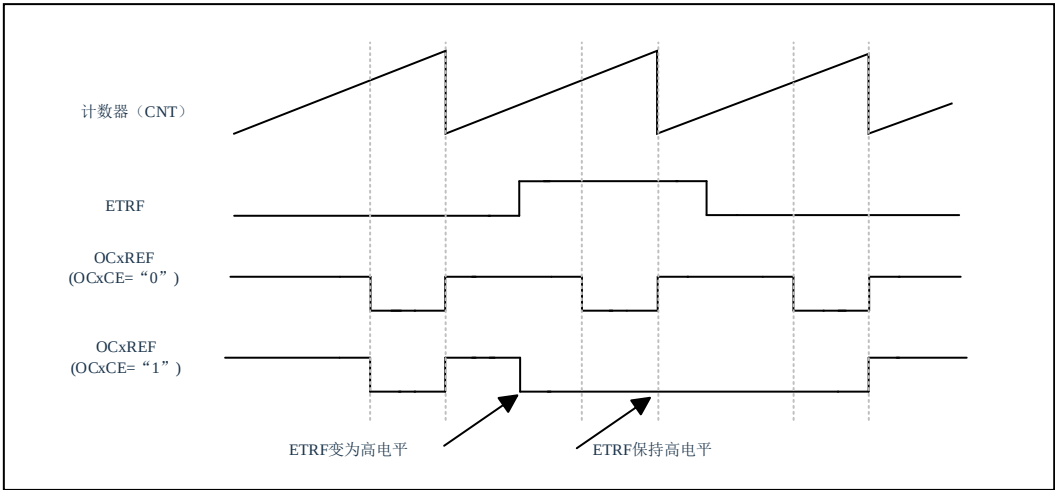
此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如，ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

- 1. 必须关闭外部触发预分频器：TIM3_SMCR 寄存器中的 ETPS[1:0] 位置 “00”。
- 2. 必须禁止外部时钟模式 2：TIM3_SMCR 寄存器中的 ECE 位置 “0”。
- 3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据应用需要进行配置。

图 7-31 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行 为。在本例中，定时器 TIM3 编程为 PWM 模式。

图 7-31 清除 TIMx 的 OCxREF



注：如果 PWM 的占空比为 100% (CCR_x>ARR),则下次计数器上溢出时会再次使能 OCxREF。

7.3.12 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIM3_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过设置 TIM3_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。

TI1 和 TI2 两个输入用于连接增量编码器。请参见表 7-1。如果使能计数器（在 TIM3_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应向上或向下计数，同时硬件对 TIM3_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIM3_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIM3_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 7-1 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。编码器输出的第三个信号表示机械零点，可与外部中断输入相连，用以触发计数器复位。

图 7-32 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIM3_CCMR1 寄存器，IC1FP1 映射到 TI1 上）。
- CC2S=“01”（TIM3_CCMR2 寄存器，IC1FP2 映射到 TI2 上）。
- CC1P=“0”，CC1NP=“0”，IC1F=“0000”（TIM3_CCER 寄存器，IC1FP1 未反相，IC1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”，IC2F=“0000”（TIM3_CCER 寄存器，IC2FP2 未反相，IC1FP2=TI2）。
- SMS=“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN = 1（TIM3_CR1 寄存器，使能计数器）。

图 7-32 编码器接口模式下的计数器工作示例

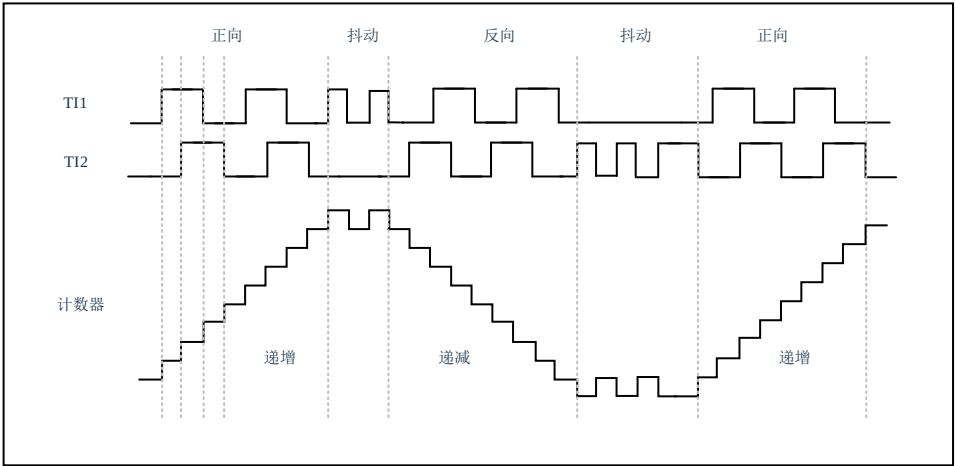
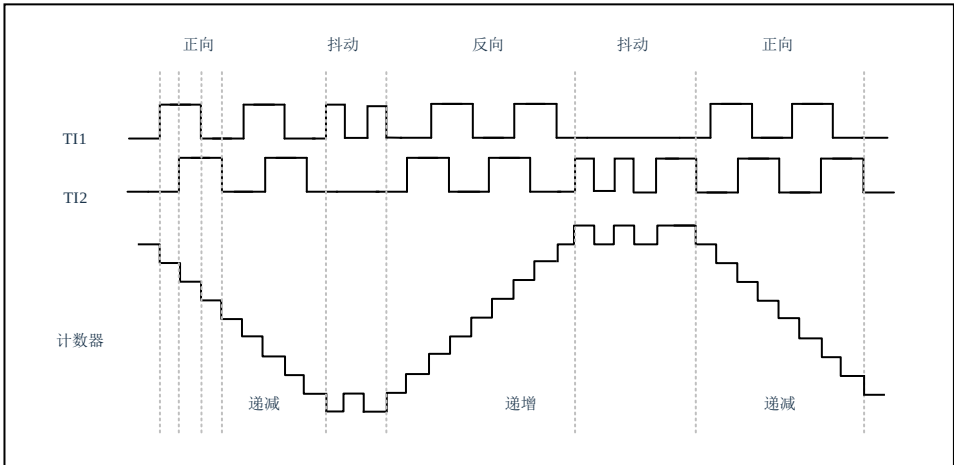


图 7-33 TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零点的编码器输出即可用于此目的。根据两个事件之间的时间间隔，可以按照固定的时间读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

7.3.13 定时器输入异或功能

TIM3_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIM3_CH1、TIM3_CH2、TIM3_CH3。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

7.3.14 定时器与外部触发同步

TIM3 定时器以下列模式与外部触发实现同步：复位模式、门控模式和触发模式。

7.3.14.1 从模式：复位模式

在发生一个触发输入事件时，计数器及其预分频器可重新初始化。此外，如果 TIM3_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIM3_ARR 和 TIM3_CCRx）都将更新。

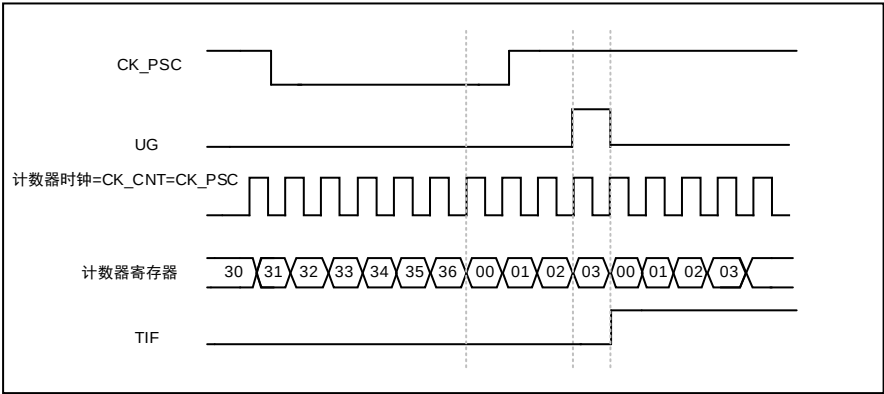
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM3_CCMR1 寄存器中的 CC1S = 01。在 TIM3_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
- 在 TIM3_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIM3_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIM3_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIM3_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIM3_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIM3_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 7-34 复位模式下的控制电路



7.3.14.2 从模式：门控模式

输入信号的电平可用来使能计数器。
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM3_CCMR1 寄存器中的 CC1S=01。在 TIM3_CCER 寄存器中写入 CC1P=1，以确定极性（仅检测低电平）。
- 在 TIM3_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIM3_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIM3_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIM3_SR 寄存器中的 TIF 标志都会置 1。

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

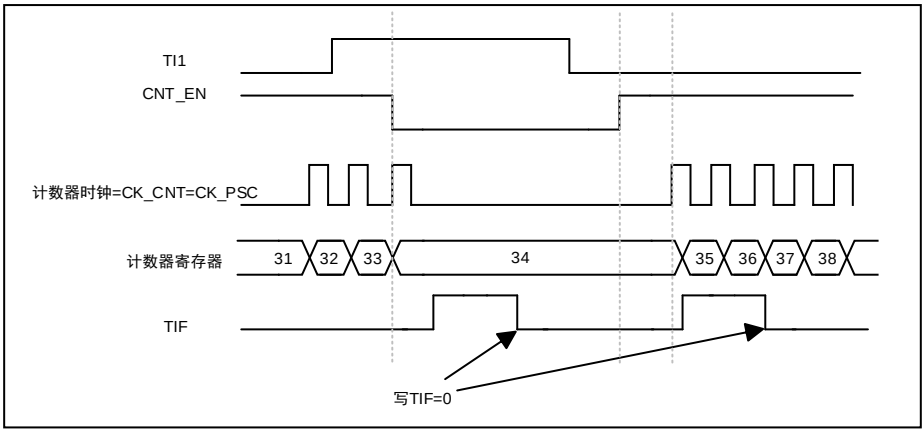
7.3.14.3 从模式：触发模式

所选输入上发生某一事件时可以启动计数器。
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIM3_CCMR1 寄存器中的 CC2S=01。在 TIM3_CCER 寄存器中写入 CC2P=1，以确定极性（仅检测低电平）。
- 在 TIM3_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIM3_SMCR 寄存器中写入 TS=110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。
TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 7-35 门控模式下的控制电路



7.3.14.4 从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIM3_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

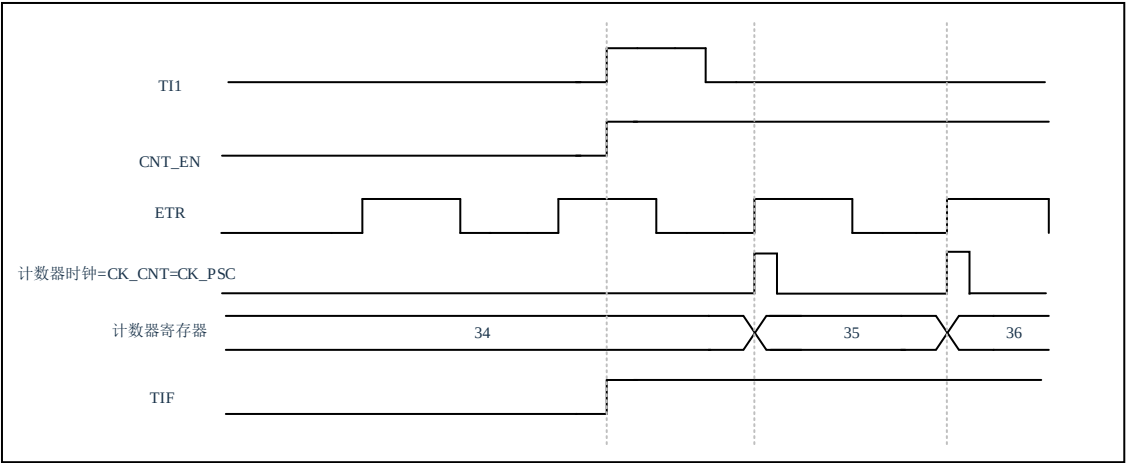
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

- 1. 通过对 TIM3_SMCR 寄存器进行如下编程，配置外部触发输入电路：
 - ETF = 0000：无滤波器
 - ETPS = 00：禁止预分频器
 - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
- 2. 如下配置通道 1，以检测 TI 的上升沿：
 - IC1F = 0000：无滤波器
 - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
 - TIM3_CCMR1 寄存器中的 CC1S = 01，只选择输入捕获源
 - TIM3_CCER 寄存器中的 CC1P = 0，以确定极性（仅检测上升沿）。
- 3. 在 TIM3_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIM3_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

注：该芯片不支持此模式。

图 7-36 触发模式下的控制电路



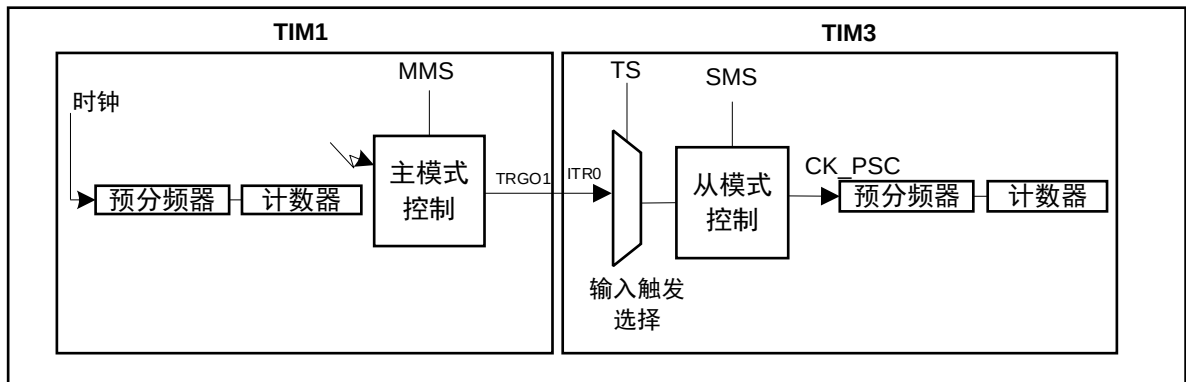
7.3.15 定时器同步

TIM3 定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

图 7-37：主/从定时器示例简要介绍了触发选择和主模式选择框图。

将一个定时器用作另一个定时器的预分频器

图 7-37 主/从定时器示例



例如，可以将定时器 1 配置为定时器 3 的预分频器。请参见图 7-37。为此：

- 将定时器 1 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIM1_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO1 都会输出一个上升沿。
- 要将定时器 1 的 TRGO1 输出连接到定时器 3，必须将定时器 3 配置为从模式，使用 ITR0 作为内部触发。通过 TIM3_SMCR 寄存器中的 TS 位（写入 TS=000）可对此进行选择。
- 然后将从模式控制器设为外部时钟模式 1（在 TIM3_SMCR 寄存器中写入 SMS=111）。这样一来，定时器 3 的时钟将由定时器 1 周期性触发信号的上升沿（与定时器 1 的计数器上溢对应）提供。
- 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1，确保先启动 Timer3,再启动 Timer1。

注意：如果选择定时器 1 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动定时器 3 的计数器。

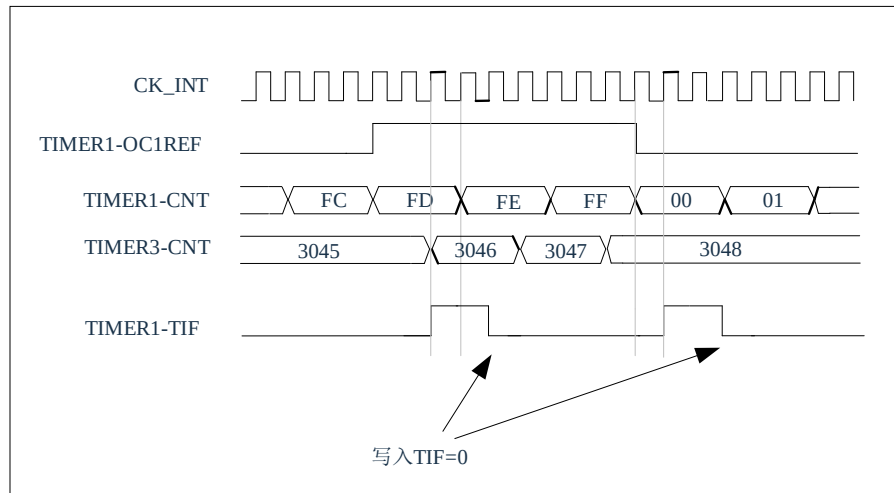
7.3.15.1 使用一个定时器使能另一个定时器

本例中通过定时器 1 的输出比较 1 来使能定时器 3。相关连接图，请参见图 7-37。仅当定时器 1 的 OC1REF 为高电平时，定时器 3 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

- 将定时器 1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 (TIM1_CR2 寄存器中的 MMS=100)。
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)。
- 配置定时器 3 以接收来自定时器 1 的输入触发 (TIM3_SMCR 寄存器中的 TS=000)。
- 将定时器 3 配置为门控模式 (TIM3_SMCR 寄存器中的 SMS=101)。
- 通过向 CEN 位 (TIM3_CR1 寄存器) 写入“1”使能定时器 3。
- 通过向 CEN 位 (TIM1_CR1 寄存器) 写入“1”启动定时器 1。

注意：计数器 3 的时钟与计数器 1 不同步，此模式仅影响定时器 3 的计数器使能信号。

图 7-38 使用定时器 1 的 OC1REF 对定时器 3 实施门控控制

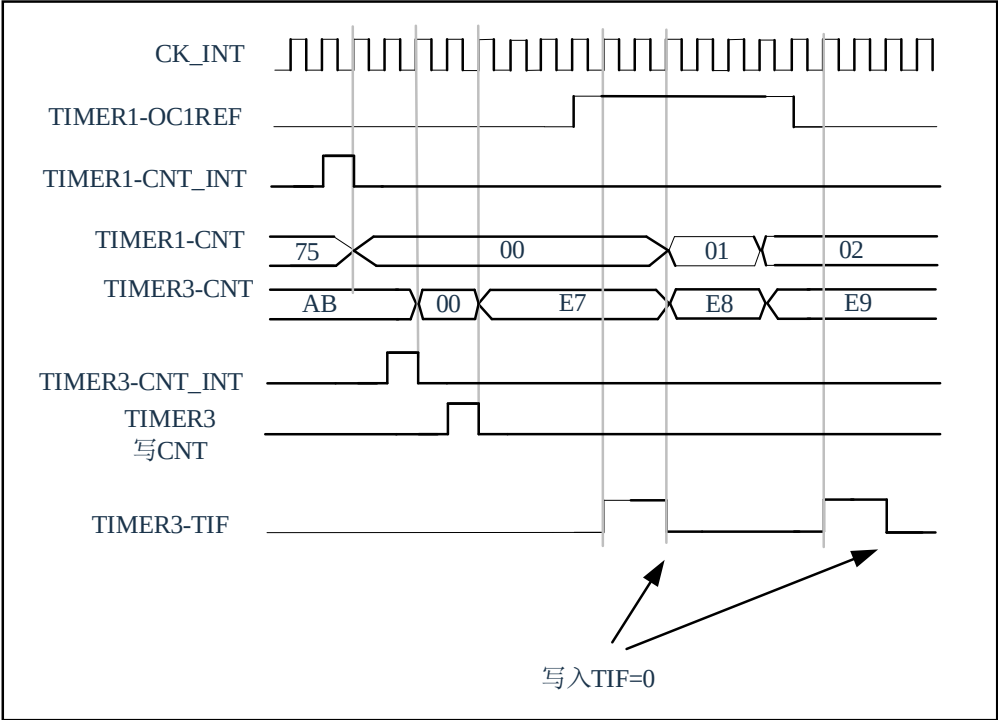


在图 7-38 的示例中，定时器 3 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 1 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中，定时器 1 与定时器 3 同步。定时器 1 为主模式，从 0 开始计数。定时器 3 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。在 TIM1_CR1 寄存器中通过向 CEN 位写入“0”来禁止定时器 1 时，定时器 3 将停止：

- 将定时器 1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 (TIM1_CR2 寄存器中的 MMS=100)。
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)。
- 配置定时器 3 以接收来自定时器 1 的输入触发 (TIM3_SMCR 寄存器中的 TS=000)。
- 将定时器 3 配置为门控模式 (TIM3_SMCR 寄存器中的 SMS=101)。
- 通过向 UG 位 (TIM1_EGR 寄存器) 写入“1”复位定时器 1。
- 通过向 UG 位 (TIM3_EGR 寄存器) 写入“1”复位定时器 3。
- 通过在定时器 3 的计数器 (TIM3_CNT) 中写入“0xE7”使定时器 3 初始化为 0xE7。
- 通过向 CEN 位 (TIM3_CR1 寄存器) 写入“1”使能定时器 3。
- 通过向 CEN 位 (TIM1_CR1 寄存器) 写入“1”启动定时器 1。
- 通过向 CEN 位 (TIM1_CR1 寄存器) 写入“0”停止定时器 1。

图 7-39 使用定时器 1 的使能信号对定时器 3 实施门控控制

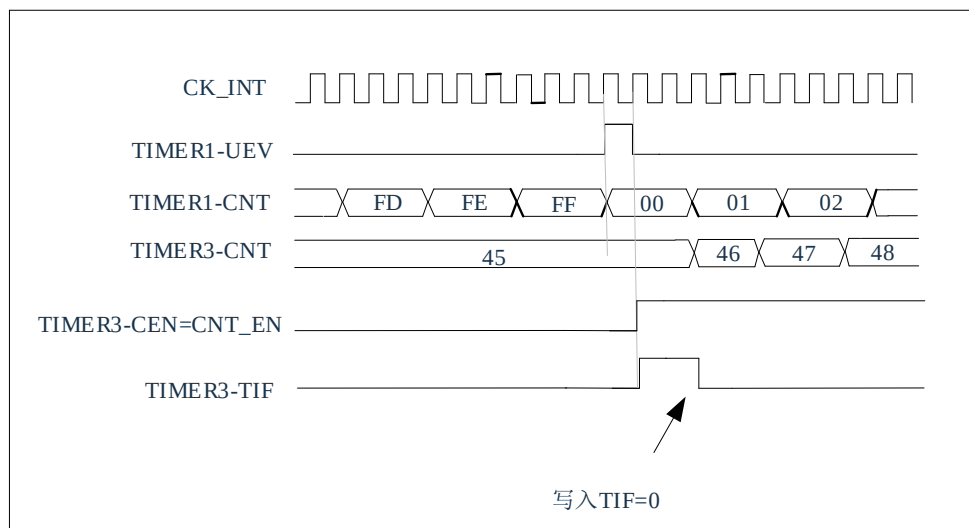


7.3.15.2 使用一个定时器启动另一个定时器

本例中使用定时器 1 的更新事件使能定时器 3。相关连接图，请参见图 7-40。只要定时器 1 生成更新事件，定时器 3 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 3 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM3_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

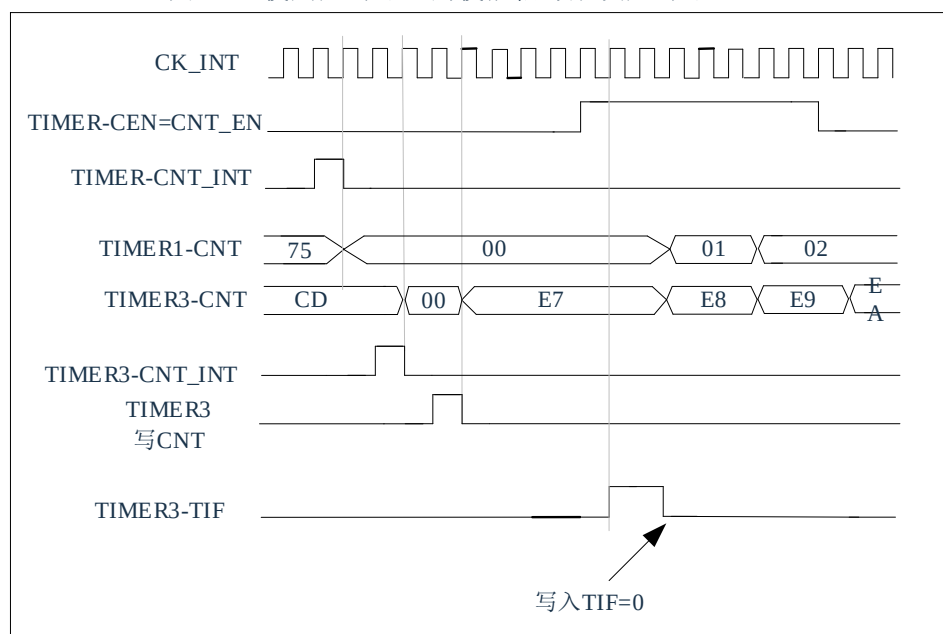
- 将定时器 1 配置为主模式，发送其更新事件 (UEV) 作为触发输出 (TIM1_CR2 寄存器中的 MMS=010)。
- 配置定时器 1 的周期 (TIM1_ARR 寄存器)。
- 配置定时器 3 以接收来自定时器 1 的输入触发 (TIM3_SMCR 寄存器中的 TS=000)。
- 将定时器 3 配置为触发模式 (TIM3_SMCR 寄存器中的 SMS=110)。
- 通过向 CEN 位 (TIM1_CR1 寄存器) 写入“1”启动定时器 1。

图 7-40 使用定时器 1 的更新时间触发定时器 3



如上述示例所示，用户可以在开始计数之前初始化两个计数器。图 7-41 显示了与图 7-40 具有相同配置，只不过处于触发模式 (TIM3_SMCR 寄存器中的 SMS=110) 而非门控模式的计数行为。

图 7-41 使用定时器 1 的使能信号触发定时器 3



7.3.15.3 将一个定时器用作另一个定时器的预分频器

例如，可以将定时器 1 配置为定时器 3 的预分频器。相关连接图，请参见图 7-37。为此：

- 将定时器 1 配置为主模式，发送其更新事件 (UEV) 作为触发输出 (TIM1_CR2 寄存器中的 MMS=010)。这样便会在计数器每次发生上溢时输出一个周期性信号。
- 配置定时器 1 的周期 (TIM1_ARR 寄存器)。
- 配置定时器 3 以接收来自定时器 1 的输入触发 (TIM3_SMCR 寄存器中的 TS=000)。
- 将定时器 3 配置为外部时钟模式 (TIM3_SMCR 寄存器中的 SMS=111)。
- 通过向 CEN 位 (TIM3_CR1 寄存器) 写入“1”启动定时器 3。
- 通过向 CEN 位 (TIM1_CR1 寄存器) 写入“1”启动定时器 1。

7.3.15.4 使用一个外部触发同步的启动 2 个定时器

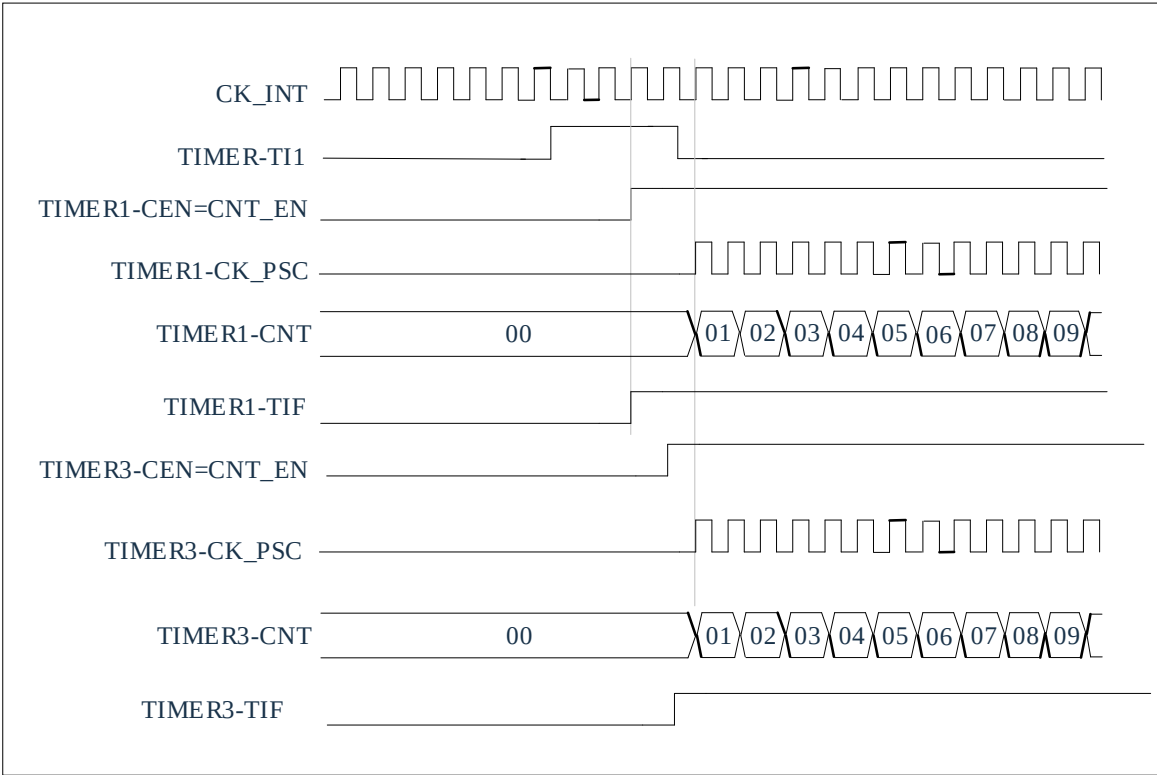
本例中，定时器 1 的 TI1 输入出现上升沿时使能定时器 1，使能定时器 1 的同时使能定时器 3。相关连接图，请参见图 7-37。要确保两个计数器对齐，定时器 1 必须配置为主/从模式 (对应的 TI1 为从，对应定时器 3 为主)：

- 将定时器 1 配置为主模式，发送其使能信号作为触发输出 (TIM1_CR2 寄存器中的 MMS=001)。
- 将定时器 1 配置为从模式以接收来自 TI1 的输入触发 (TIM1_SMCR 寄存器中的 TS=100)。
- 将定时器 1 配置为触发模式 (TIM1_SMCR 寄存器中的 SMS=110)。
- 通过写入 MSM=1 (TIM1_SMCR 寄存器) 将定时器 1 配置为主/从模式。
- 配置定时器 3 以接收来自定时器 1 的输入触发 (TIM3_SMCR 寄存器中的 TS=000)。
- 将定时器 3 配置为触发模式 (TIM3_SMCR 寄存器中的 SMS=110)。

当 TI1 (定时器 1) 出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

注意：本例中，两个定时器都在启动之前进行了初始化 (通过将各自的 UG 位置 1)。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (TIMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在定时器 1 的 CNT_EN 与 CK_PSC 之间产生了延迟。

图 7-42 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 3



7.3.16 调试模式

当芯片进入调试模式时，根据 1.4 中的设置，TIM3 计数器可以继续正常工作或者停止工作。

7.4 寄存器映射

(TIM3 基地址 = 0x4006_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
CR1	0x00	R/W	TIM3 控制寄存器 1	0x00
CR2	0x04	R/W	TIM3 控制寄存器 2	0x00
SMCR	0x08	R/W	TIM3 从模式控制寄存器	0x00
DIER	0x0C	R/W	TIM3 DMA/中断使能寄存器	0x00
SR	0x10	R/W	TIM3 状态寄存器	0x00
EGR	0x14	R/W	TIM3 事件生成寄存器	0x00
CCMR1	0x18	R/W	TIM3 捕获/比较模式寄存器 1	0x00
CCMR2	0x1C	R/W	TIM3 捕获/比较模式寄存器 2	0x00
CCER	0x20	R/W	TIM3 捕获/比较使能寄存器	0x00
CNT	0x24	R/W	TIM3 计数器	0x00
PSC	0x28	R/W	TIM3 预分频器	0x00
ARR	0x2C	R/W	TIM3 自动重载寄存器	0xFFFF
CCR1	0x34	R/W	TIM3 捕获/比较寄存器 1	0x00
CCR2	0x38	R/W	TIM3 捕获/比较寄存器 2	0x00
CCR3	0x3C	R/W	TIM3 捕获/比较寄存器 3	0x00
CCR4	0x40	R/W	TIM3 捕获/比较寄存器 4	0x00
DCR	0x48	R/W	TIM3 DMA 控制寄存器	0x00
DMAR	0x4C	R/W	TIM3 全传输 DMA 地址	0x00

7.5 寄存器说明

7.5.1 TIM3 控制寄存器 1 (TIM3_CR1)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9:8	CKD[1:0]	时钟分频 (Clock division) 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (ETR、TIx) 所使用的死区及采样时钟 (tdts) 之间的分频比 00: $tdts = tck\_int$ 01: $tdts = 2 * tck\_int$ 10: $tdts = 4 * tck\_int$ 11: 保留，禁止设置成此值	0x0
7	ARPE	自动重载预装载使能 0: TIM3_ARR 寄存器不进行缓冲 1: TIM3_ARR 寄存器进行缓冲	0
6:5	CMS[1:0]	中心对齐模式选择: 00: 边沿对齐模式。边沿对齐模式。计数器依据方向位 (DIR) 向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道 (TIM3_CCMRx 寄存 CCxS=00) 的输出比较中断标志位，只在计数器向下计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计数。计数器交替地向上和向下计数。配置为输出的通道 (TIM3_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。计数器交替地向上和向下计数。配置为输出的通道 (TIM3_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时 (CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。	0x0
4	DIR	方向: 0: 计数器向上计数 1: 计数器向下计数 注：当计数器配置为中央对齐模式或编码器模式时，该位只读	0
3	OPM	单脉冲模式: 0: 计数器在发生更新事件时不会停止计数 1: 在发生下一次更新事件 (清除 CEN 位) 时，计数器停止计数	0
2	URS	更新请求源: 此位由软件置 1 和清零，用以选择 UEV 事件源 0: 如果允许产生更新中断或 DMA 请求，则下述任一事件产生一个更新中断或 DMA 请求 — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件 1: 如果允许产生更新中断或 DMA 请求，如果允许产生更新中断或 DMA 请求，则只有计数器上溢/下溢产生一个更新中断或 DMA 请求	0
1	UDIS	更新禁止: 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0: 允许 UEV。更新 (UEV) 事件可通过以下事件之一生成 — 计数器上溢/下溢 — 将 UG 位置 1 — 从模式控制器产生的更新 被缓存的寄存器被装入它们的预装载值 1: 禁止 UEV。不产生更新事件，影子寄存器	0

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

7.5.2 TIM3 控制寄存器 2 (TIM3_CR2)

位	符号	描述	复位值
31:8	--	保留，须为 0	0
7	TI1S	TI1 选择： 0: TIM3_CH1 引脚连接到 TI1 输入 1: TIM3_CH1、TIM3_CH2 和 TIM3_CH3 管脚经异或后连到 TI1 输入	0
6:4	MMS[1:0]	主模式选择： 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: 复位 — TIM3_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。 001: 允许 — 计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO)。有时需要在同一时间启动多个定时器或控制从定时器的一个窗口。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式 (TIM3_SMCR 寄存器中 MSM 位的描述)。 010: 更新 — 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 — 一旦发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时 (即是它已经为高)，触发输出送出一个正脉冲 (TRGO) 100: 比较 — OC1REF 信号用作触发输出 (TRGO) 101: 比较 — OC2REF 信号用作触发输出 (TRGO) 110: 比较 — OC3REF 信号用作触发输出 (TRGO) 111: 比较 — OC4REF 信号用作触发输出 (TRGO)	0x0
3	CCDS	捕获/比较 DMA 选择： 0: 发生 CCx 事件时,发送 CCx DMA 请求 1: 发生更新事件时,发送 CCx DMA 请求	0
2:0	--	保留，须为 0	0x0

7.5.3 TIM3 从模式控制寄存器 (TIM3_SMCR)

位	符号	描述	复位值
31:8	--	保留，须为 0。	0x0
7	MSM	主/从模式： 0: 不执行任何操作 1: 触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的当前定时器和从定时器间的同步（通过 TRGO）。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的	0
6:4	TS[2:0]	触发选择，此位域可选择将要用于同步计数器的触发输入 000: Internal Trigger 0(ITR0) 001: Internal Trigger 1(ITR1) 010: Internal Trigger 2(ITR2) 011: Internal Trigger 3(ITR3) 100: TI1 边沿检测器 (TI1F_ED) 101: 滤波后的定时器输入 1 (TI1FP1) 110: 滤波后的定时器输入 2 (TI2FP2) 111: 外部触发输入 (ETRF) 注意：这些位只能在未使用的情况下进行更改，以避免转换时出现错误的边沿检测。	0x0
3	OCCS	OCREF 清除选择位：（该位用于选择 OCREF 的清除源） 0: OCREF_CLR_INT 连接到 OCREF_CLR 输入 1: OCREF_CLR_INT 连接到 ETRF	0
2:0	SMS	从模式选择： 选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入 控制寄存器和控制寄存器说明）。 000: 关闭从模式 如果 CEN=1，则预分频器直接由内部时钟驱动。 001: 编码器模式 1 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。 010: 编码器模式 2 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。 011: 编码器模式 3 模式 1 和模式 2 的综合。 100: 复位模式 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号 101: 门控模式 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的 111: 外部时钟模式 1 选中的触发输入(TRGI)的上升沿驱动计数器 注：如果 TI1F_EN 被选为触发输入(TS=100)时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平	0x0

表 7-2 TIM3 内部触发连接

从定时器	ITT0(TS=000)	ITT0(TS=001)	ITT0(TS=010)	ITT0(TS=011)
TIM3	TIM1	保留	保留	TIM14 OC1

7.5.4 TIM3 DMA/中断使能寄存器 (TIM3_DIER)

位	符号	描述	复位值
31:15	--	保留，须为 0	0x0
14	TDE	触发 DMA 请求使能 0: 禁止 DMA 请求 1: 使能 DMA 请求	0
13	--	保留，须为 0	0
12	CC4DE	捕获/比较 4 DMA 请求使能 0: 禁止 CC4 DMA 请求 1: 使能 CC4 DMA 请求	0
11	CC3DE	捕获/比较 3 DMA 请求使能 0: 禁止 CC3 DMA 请求 1: 使能 CC3 DMA 请求	0
10	CC2DE	捕获/比较 2 DMA 请求使能 0: 禁止 CC2 DMA 请求 1: 使能 CC2 DMA 请求	0
9	CC1DE	捕获/比较 1 DMA 请求使能 0: 禁止 CC1 DMA 请求 1: 使能 CC1 DMA 请求	0
8	UDE	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求	0
7	--	保留，须为 0	0
6	TIE	触发信号 (TGRI) 中断使能 0: 禁止触发信号 (TGRI) 中断 1: 使能触发信号 (TGRI) 中断	0
5	--	保留，须为 0	0
4	CC4IE	CC4IE: 捕获/比较 4 中断使能 0: 禁止 CC4 中断 1: 使能 CC4 中断	0
3	CC3IE	CC3IE: 捕获/比较 3 中断使能 0: 禁止 CC3 中断 1: 使能 CC3 中断	0
2	CC2IE	CC2IE: 捕获/比较 2 中断使能 0: 禁止 CC2 中断 1: 使能 CC2 中断	0
1	CC1IE	CC1IE: 捕获/比较 1 中断使能 0: 禁止 CC1 中断 1: 使能 CC1 中断	0
0	UIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断	0

7.5.5 TIM3 状态寄存器 (TIM3_SR)

位	符号	描述	复位值
31:13	--	保留，须为 0	0x0
12	CC4OF	捕获/比较 4 过捕获标志 请参见 CC1OF 说明	0
11	CC3OF	捕获/比较 3 过捕获标志 请参见 CC1OF 说明	0
10	CC2OF	捕获/比较 2 过捕获标志 请参见 CC1OF 说明	0
9	CC1OF	捕获/比较 1 过捕获标志 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“0”可将该位清零 0: 未检测到过捕获。 1: TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1	0
8:7	--	保留，须为 0	0x0
6	TIF	触发器中断标记 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿）时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生 1: 触发器中断等待响应	0
5	--	保留，须为 0	0
4	CC4IF	CC4IF: 捕获/比较 4 中断标志 请参见 CC1IF 说明	0
3	CC3IF	CC3IF: 捕获/比较 3 中断标志 请参见 CC1IF 说明	0
2	CC2IF	CC2IF: 捕获/比较 2 中断标志 请参见 CC1IF 说明	0
1	CC1IF	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，中心对齐模式下除外（请参见 TIM3_CR1 寄存器中的 CMS 位说明）。通过软件清零。 0: 无匹配发生。 1: TIM3_CNT 的值与 TIM3_CCR1 的值匹配 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 TIM3_CCR1 寄存器将该位清零。 0: 未发生输入捕获事件 1: 输入捕获产生并且计数器值已装入 TIM1_CCR1(在 IC1 上检测到与所选极性相同的边沿) 注：当 CEN 打开，该位也会被置位	0
0	UIF	更新中断标志 该位在发生更新事件时通过硬件置 1。需要通过软件清零 0: 未发生更新 1: 更新中断等待响应。该位在以下情况下更新寄存器时由硬件置 1 – 若 TIM3_CR1 寄存器的 UDIS=0，当 REP_CNT=0 时产生更新事件(重复向下计数器上溢或下溢时)； – 若 TIM3_CR1 寄存器的 UDIS=0、URS=0，当 TIM3_EGR 寄存器的 UG=1 时产生更新事件(软件对 CNT 重新初始化)； – 若 TIM3_CR1 寄存器的 UDIS=0、URS=0，当 CNT 被触发事件重初始化时产生更新事件。	0

7.5.6 TIM3 事件生成寄存器 (TIM3_EGR)

位	符号	描述	复位值
31:7	--	保留，须为 0	0x0
6	TG	生成触发信号 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: TIM1_SR 寄存器中的 TIF 标志置 1。若开启对应的中断和 DMA，则产生相应的中断和 DMA。	0
5	--	保留，须为 0	0
4	CC4G	CC4G: 捕获/比较 4 生成 请参见 CC1G 说明	0
3	CC3G	CC3G: 捕获/比较 3 生成 请参见 CC1G 说明	0
2	CC2G	CC2G: 捕获/比较 2 生成 请参见 CC1G 说明	0
1	CC1G	CC1G: 捕获/比较 1 生成 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： 若开启对应的中断和 DMA，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果通道 CC1 配置为输入： TIM3_CCR1 寄存器中将捕获到计数器当前值。若开启对应的中断和 DMA，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果 CC1IF 标志已为高电平，CC1OF 标志将置 1	0
0	UG	更新生成：（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，并产生一个更新事件。注意：预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0，若 DIR=1(向下计数)则计数器装载 TIM3_ARR 的值。	0

7.5.7 TIM3 捕获/比较模式寄存器 1 (TIM3_CCMR1)

与 TIM1 类似，此寄存器的所有其它位在输入模式和输出模式下的功能均不同。
输出比较模式时。

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	OC2CE	输出比较 2 清零使能，参见 OC1CE 说明	0
14:12	OC2M[2:0]	输出比较 2 模式，参见 OC1M 说明	0x0
11	OC2PE	输出比较 2 预装载使能，参见 OC1PE 说明	0
10	OC2FE	输出比较 2 快速使能，参见 OC1FE 说明	0
9:8	CC2S[1:0]	捕获/比较 2 选择： 此位域定义通道方向（输入/输出）以及所使用的输入 00: CC2 通道配置为输出 01: CC2 通道配置为输入，IC2 映射到 TI2 上 10: CC2 通道配置为输入，IC2 映射到 TI1 上 11: CC2 通道配置为输入，IC2 映射到 TRC 上。此模式 仅在通过 TS 位（TIM3_SMCR 寄存器）选择内部触 发输入时有效 注意：仅当通道关闭时（TIM3_CCER 中的 CC2E “0”），才可 向 CC2S 位写入数据。	0x0
7	OC1CE	输出比较 1 清零使能： 0: OC1REF 不受 ETRF 输入影响 1: ETRF 输入上检测到高电平时，OC1REF 立即清零	0
6:4	OC1M	这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。 OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 000: 冻结。输出比较寄存器 TIM3_CCR1 与计数器 TIM3_CNT 间的比较对 OC1REF 不起作用。 001: 匹配时设置通道 1 为有效电平。当计数器 TIM3_CNT 的值与捕获 / 比较寄存器 1(TIM3_CCR1) 相同时，强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIM3_CNT 的值与捕获 / 比较寄存器 1(TIM3_CCR1) 相同时，强制 OC1REF 为低。 011: 翻转。当 TIM3_CCR1=TIM3_CNT 时，翻 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1 - 在向上计数模式下，只要 TIM3_CNT<TIM3_CCR1，通道 1 便为有效电平，否 则为无效电平。在向下计数模式下，只要 TIM3_CNT>TIM3_CCR1，通道 1 便为无效电平 (OC1REF=0)，否则为有效电平 (OC1REF=1)。 111: PWM 模式 2 - 在向上计数模式下，只要 TIM3_CNT<TIM3_CCR1，通道 1 便为无效电平，否 则为有效电平。在向下计数模式下，只要 TIM3_CNT>TIM3_CCR1，通道 1 便为有效电平 (OC1REF=0)，否则为无效电平 (OC1REF=1)。 注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果 改变 了或在输出比较模式中从冻结模式切换到 PWM 模式时， OC1REF 电平才改变	0x0
3	OC1PE	输出比较 1 预装载使能： 0: 禁止 TIM3_CCR1 寄存器的预装载功能，可随时写 入 TIM3_CCR1 寄存器，且新值马上起作用 1: 开启 TIM3_CCR1 寄存器的预装载功能，读写操作 仅对预装载寄存器操作，TIM3_CCR1 的预装载值	0

		在更新事件到来时被载入当前寄存器中 注 1：仅在单脉冲模式下，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定	
2	OC1FE	输出比较 1 快速使能： 此位用于加快触发输入事件对 CC 输出的影响 0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期 1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OC1FE 的只在通道被配置成 PWM1 或 PWM2 模式时起作用	0
1:0	CC1S	捕获/比较 1 选择： 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上 11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据	0x0

输入捕获模式时。

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:12	IC2F	输入捕获 2 滤波器，参见 IC1F 说明	0x0
11:10	IC2PSC[1:0]	输入捕获 2 预分频器，参见 IC1PSC 说明	0x0
9:8	CC2S	捕获/比较 2 选择： 此位域定义通道方向（输入/输出）以及所使用的输入 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上 10：CC2 通道配置为输入，IC2 映射到 TI1 上 11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIM3_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM3_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据	0x0
7:4	IC1F[3:0]	输入捕获 1 滤波器： 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿 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 _{DTS} /2，N=6 0101：f _{SAMPLING} =f _{DTS} /2，N=8 0110：f _{SAMPLING} =f _{DTS} /4，N=6 0111：f _{SAMPLING} =f _{DTS} /4，N=8 1000：f _{SAMPLING} =f _{DTS} /8，N=6 1001：f _{SAMPLING} =f _{DTS} /8，N=8 1010：f _{SAMPLING} =f _{DTS} /16，N=5 1011：f _{SAMPLING} =f _{DTS} /16，N=6 1100：f _{SAMPLING} =f _{DTS} /16，N=8 1101：f _{SAMPLING} =f _{DTS} /32，N=5 1110：f _{SAMPLING} =f _{DTS} /32，N=6 1111：f _{SAMPLING} =f _{DTS} /32，N=8	0x0
3:2	IC1PSC	输入捕获 1 预分频器：	0x0

		此位域定义 CC1 输入 (IC1) 的预分频比。 只要 CC1E=“0” (TIM3_CCER 寄存器)，预分频器便立即复位 00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获。 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获。 11: 每发生 8 个事件便执行一次捕获。	
1:0	CC1S	捕获/比较 1 选择: 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 上 10: CC1 通道配置为输入, IC1 映射到 TI2 上 11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM3_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM3_CCER 中的 CC1E =“0”), 才可向 CC1S 位写入数据	0x0

7.5.8 TIM3 捕获/比较模式寄存器 2 (TIM3_CCMR2)

输出比较模式。

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	OC4CE	输出比较 4 清零使能，参见 OC1CE 说明	0
14:12	OC4M[2:0]	输出比较 4 模式，参见 OC1M 说明	0x0
11	OC4PE	输出比较 4 预装载使能，参见 OC1PE 说明	0
10	OC4FE	输出比较 4 快速使能，参见 OC1FE 说明	0
9:8	CC4S[1:0]	捕获/比较 4 选择： 此位域定义通道方向（输入/输出）以及所使用的输入 00: CC4 通道配置为输出 01: CC4 通道配置为输入，IC4 映射到 TI4 上 10: CC4 通道配置为输入，IC4 映射到 TI3 上 11: CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIM3_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM3_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。	0x0
7	OC3CE	输出比较 3 清零使能，参见 OC1CE 说明	0
6:4	OC3M	输出比较 3 模式，参见 OC1M 说明	0x0
3	OC3PE	输出比较 3 预装载使能，参见 OC1PE 说明	0
2	OC3FE	输出比较 3 快速使能，参见 OC1FE 说明	0
1:0	CC3S	捕获/比较 3 选择： 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入，IC3 映射到 TI3 上 10: CC3 通道配置为输入，IC3 映射到 TI4 上 11: CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIM3_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM3_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据	0x0

输入捕获模式

位	符号	描述	复位值
31:12	IC4F	输入捕获 4 滤波器，参见 IC1F 说明	0x0
11:10	IC4PSC	输入捕获 4 预分频器，参见 IC1PSC 说明	0x0
9:8	CC4S[1:0]	捕获/比较 4 选择： 此位域定义通道方向（输入/输出）以及所使用的输入 00: CC4 通道配置为输出 01: CC4 通道配置为输入，IC4 映射到 TI4 上 10: CC4 通道配置为输入，IC4 映射到 TI3 上 11: CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIM3_SMCR 寄存器）选择内部触发输入时有效 注意：仅当通道关闭时（TIM3_CCER 中的 CC4E “0”），才可向 CC4S 位写入数据。	0x0
7:4	IC3F	输入捕获 3 滤波器： 这几位定义了 TI3 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变 0000: 无滤波器，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 _{DTS} /2，N=6 0101: f _{SAMPLING} =f _{DTS} /2，N=8 0110: f _{SAMPLING} =f _{DTS} /4，N=6 0111: f _{SAMPLING} =f _{DTS} /4，N=8	0x0

		1000: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=6$ 1001: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=8$ 1010: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=5$ 1011: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=6$ 1100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$ 1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$ 1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$	
3:2	IC3PSC	输入捕获 3 预分频器: 此位域定义 CC3 输入 (IC3) 的预分频比。 只要 CC3E=“0” (TIM3_CCER 寄存器), 预分频器便立即复位 00: 无预分频器, 捕获输入上每检测到一个边沿便执行捕获。 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获。 11: 每发生 8 个事件便执行一次捕获。	0x0
1:0	CC3S	捕获/比较 3 选择: 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入, IC3 映射到 TI3 上 10: CC3 通道配置为输入, IC3 映射到 TI4 上 11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIM1_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM3_CCER 中的 CC3E “0”), 才可向 CC3S 位写入数据	0x0

7.5.9 TIM3 计数器 (TIM3_CNT)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CNT[15:0]	计数器值 (Counter value)	0x0

7.5.10 TIM3 预分频器 (TIM3_PSC)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	PSC[15:0]	预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ PSC 包含每次发生更新事件时要装载到活动预分频器寄存器的值。	0x0

7.5.11 TIM3 捕获/比较使能寄存器 (TIM3_CCER)

位	符号	描述	复位值
31:16	--	保留,须为 0	0x0
15	CC4NP	捕获/比较 4 互补输出极性, 请参见 CC1PN 说明	0
14	--	保留,须为 0	0
13	CC4P	捕获/比较 4 输出极性参见 CC1P 说明	0
12	CC4E	捕获/比较 4 输出使能, 请参见 CC1E 说明	0
11	CC3NP	捕获/比较 3 互补输出极性, 请参见 CC1PN 说明	0
10	--	保留,须为 0	0
9	CC3P	捕获/比较 3 输出极性, 请参见 CC1P 说明	0
8	CC3E	捕获/比较 3 输出使能, 请参见 CC1E 说明	0
7	CC2NP	捕获/比较 2 互补输出极性,请参见 CC1PN 说明	0
6	--	保留,须为 0	0
5	CC2P	捕获/比较 2 输出极性, 请参见 CC1P 说明	0
4	CC2E	捕获/比较 2 输出使能, 请参见 CC1E 说明	0
3	CC1NP	CC1 通道配置为输出时: 必须保持为零。 CC1 通道配置为输入时: 与 CC1P 配合使用	0
2	--	保留,须为 0	0
1	CC1P	捕获/比较 1 输出极性 CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: CC1NP/CC1P 两位选择是 TI1FP1 还是 TI2FP1 的极性信号作为触发或捕获信号 不反相/上升沿: 捕获发生在 TIxFP1 的上升沿(捕获, 复位 00: 触发, 外部时钟或触发模式); TIxFP1 不反相(门触发模式, 编码模式)。 反相/下降沿: 捕获发生在 TIxFP1 的下降沿(捕获, 复位触 01: 发, 外部时钟或触发模式); TIxFP1 反相(门触发模式, 编码模式)。 10: 保留, 无效配置。 11: 不反向, 双边沿	0
0	CC1E	捕获 / 比较 1 输出使能 CC1 通道配置为输出: 0: 关闭— OC1 禁止输出 1: 开启— OC1 信号输出到对应的输出引脚。 CC1 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 1 (TIM3_CCR1) 中。 0: 禁止捕获。 1: 使能捕获。	0

表 7-3 标准 OCx 通道的输出控制位

CCx 位	OCx 通道输出状态
0	禁止输出 (OCx=0、OCx_EN=0)
1	OCx=OCxREF+极性、OCx_EN=1)

注意: 与标准 OCx 通道相连的外部 IO 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。

7.5.12 TIM3 自动重载寄存器 (TIM3_ARR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	ARR[15:0]	自动重载值 ARR 为要装载到实际自动重载寄存器的值。 有关 ARR 更新和行为的详细信息，请参见时基单元。当自动重载值为空时，计数器不工作。	0xFFFF

7.5.13 TIM3 捕获/比较寄存器 1 (TIM3_CCR1)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR1[15:0]	捕获/比较 1 值 如果通道 CC1 配置为输出： CCR1 是捕获/比较寄存器 1 的预装载值。 如果没有通过 TIM3_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。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 1）。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 OC1 输出上发出信号的值。 如果通道 CC1 配置为输入： CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值	0x0

7.5.14 TIM3 捕获/比较寄存器 2 (TIM3_CCR2)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR2[15:0]	捕获/比较 2 值 如果通道 CC2 配置为输出： CCR2 是捕获/比较寄存器 2 的预装载值。 如果没有通过 TIM3_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。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 2）。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 OC2 输出上发出信号的值。 如果通道 CC2 配置为输入： CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值	0x0

7.5.15 TIM3 捕获/比较寄存器 3 (TIM3_CCR3)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR3[15:0]	捕获/比较 3 值 如果通道 CC3 配置为输出： CCR3 为要装载到实际捕获/比较 3 寄存器的值(预装载值)。 如果没有通过 TIM3_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。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 OC3 输出上发出信号的值 如果通道 CC3 配置为输入： CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值	0x0

7.5.16 TIM3 捕获/比较寄存器 4 (TIM3_CCR4)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR4[15:0]	捕获/比较 4 值 如果通道 CC4 配置为输出： CCR4 为要装载到实际捕获/比较 4 寄存器的值(预装载值)。 如果没有通过 TIM3_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 OC4 输出上发出信号的值。 如果通道 CC4 配置为输入： CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值	0x0

7.5.17 TIM3 DMA 控制寄存器 (TIM3_DCR)

位	符号	描述	复位值
31:13	--	保留	0x0
12:8	DBL	DMA 连续传送长度 (DMA burst length) 该 5 位定义了 DMA 在连续模式下的传送长度（当对 TIM3_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 00011: 4 次传输: 10001: 18 次传输	0x0
7:5	--	保留	0x0
4:0	DBA	DMA 基址 (DMA base address) 该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIM3_CR1 寄存器地址开始计算的偏移量。示例： 00000: TIM1_CR1 00001: TIM1_CR2 00010: TIM1_SCMR:	0x0

7.5.18 TIM3 全传输 DMA 地址 (TIM3_DMAR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	DMAB	DMA 连续传送寄存器 (DMA register for burst accesses) 对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器 (TIM3_CR1 地址) + (DBA + DMA 索引) x 4, 其中 TIM3_CR1 地址为控制寄存器 1 的地址，DBA 为 TIM3_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIM3_DCR 寄存器中配置的 DBL）之间。	0x0

第8章 通用定时器 TIM14

8.1 TIM14 简介

TIM14 通用定时器包含一个 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），或者生成输出波形（输出比较、PWM）。

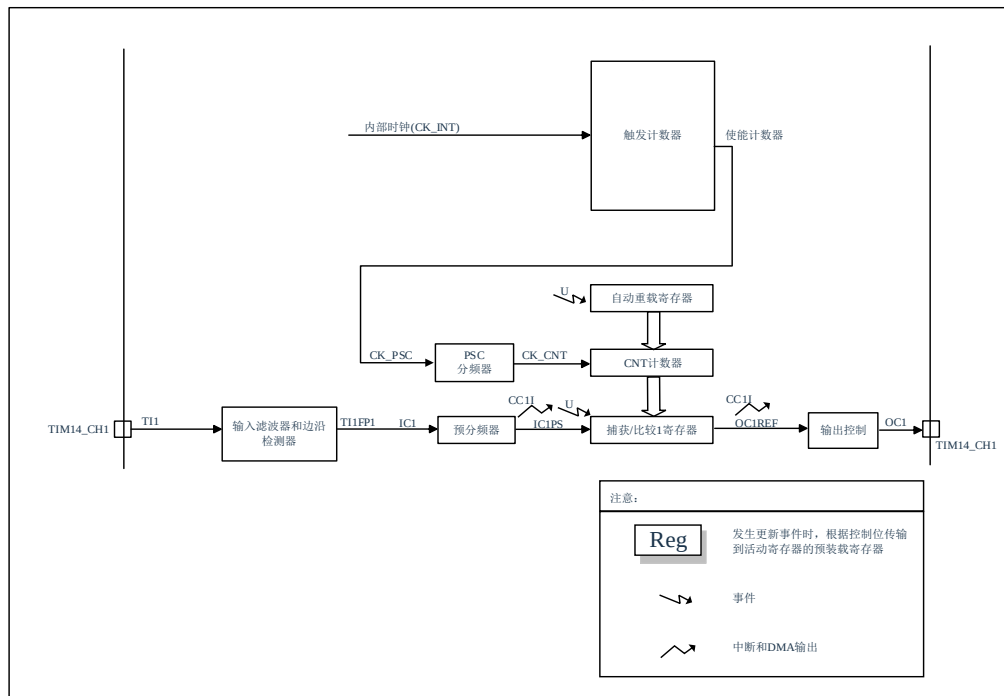
使用定时器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

8.2 TIM14 主要特性

TIM14 具有以下特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 和 65536 之间
- 独立通道，可用于：
 - 输入捕获
 - 输出比较
 - PWM 生成（边沿对齐模式）
 - 单脉冲模式输出
- 发生如下事件时生成中断：
 - 更新：计数器上溢、计数器初始化（通过软件）
 - 输入捕获
 - 输出比较

图 8-1 通用定时器框图 (TIM14)



8.3 TIM14 功能说明

8.3.1 时基单元

定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。计数器采用向上计数方式计数。计数器的时钟可通过预分频器进行分频。计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIM14_CNT)
- 预分频器寄存器 (TIM14_PSC)
- 自动重载寄存器 (TIM14_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，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 TIM14_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 TIM14_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIM14_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器。

注意：计数器将在 TIM14_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数

8.3.1.1 预分频器说明

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 TIM14_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

以下各图以一些示例说明在预分频比实时变化时计数器的行为

图 8-2 预分频器由 1 变为 2 时的计数器时序图

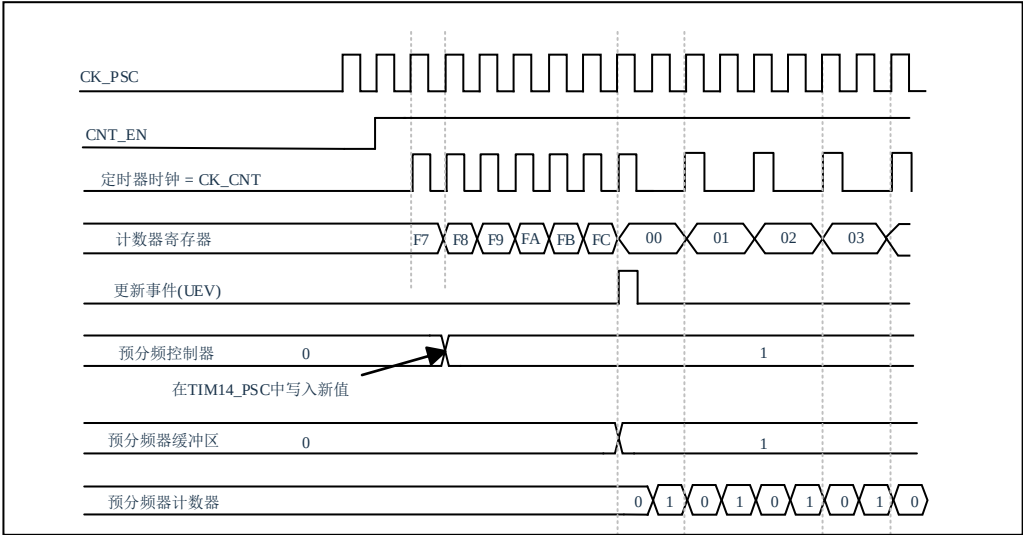
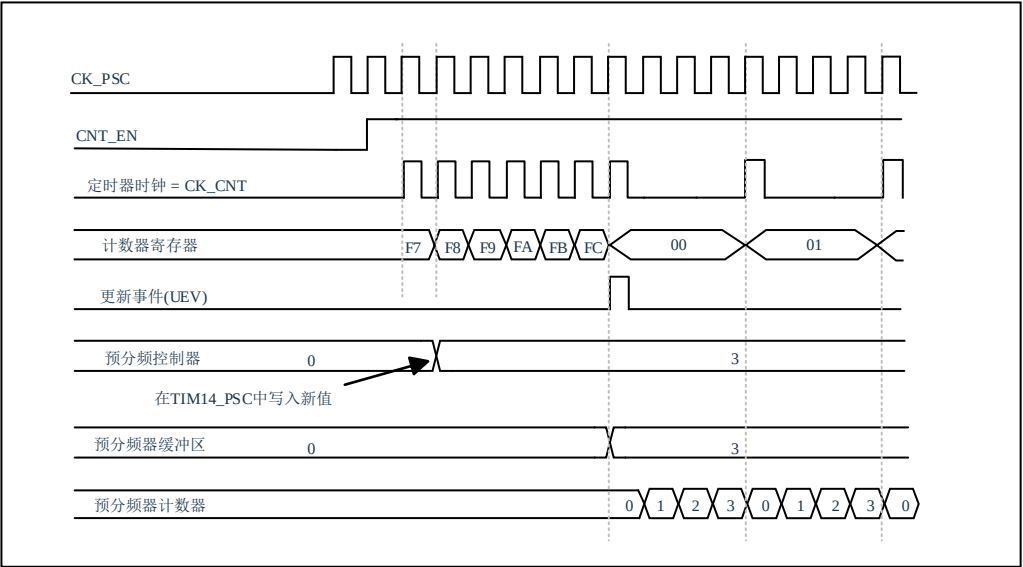


图 8-3 预分频器由 1 变为 4 时的计数器时序图



8.3.2 计数器模式

8.3.2.1 向上计数模式

在向上计数模式下，计数器从 0 计数到自动重载值（TIM14_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。

将 TIM14_EGR 寄存器中的 UG 位置 1 也会生成更新事件。

通过软件将 TIM14_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，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）。此外，如果 TIM14_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM14_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 自动重载影子寄存器将以预装载值 (TIM14_ARR) 进行更新
- 预分频器的缓冲区中将重新装载预装载值（TIM14_PSC 寄存器的内容）

以下各图以一些示例说明当 TIM14_ARR=0x36 时不同时钟频率下计数器的行为。

图 8-4 计数器时序图，1 分频内部时钟

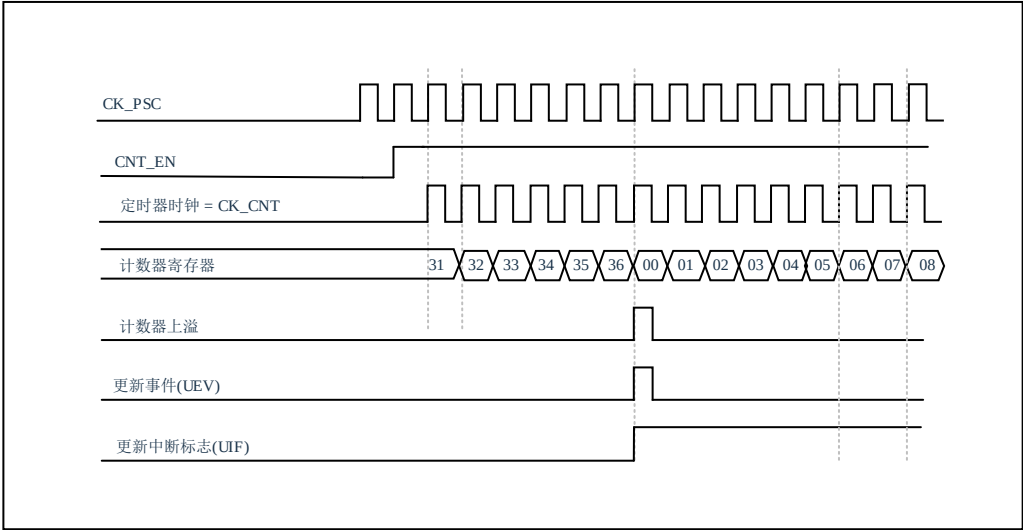


图 8-5 计数器时序图，2 分频内部时钟

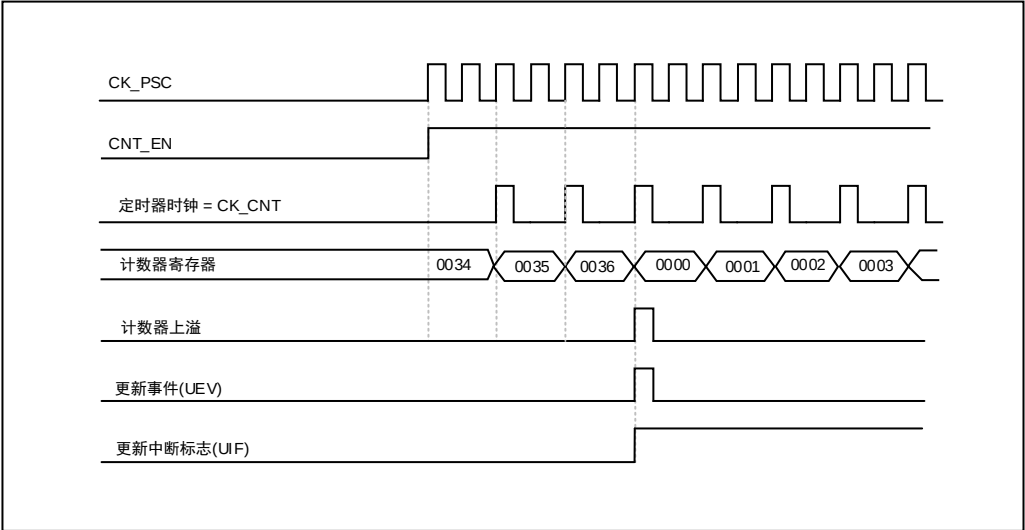


图 8-6 计数器时序图，4 分频内部时钟

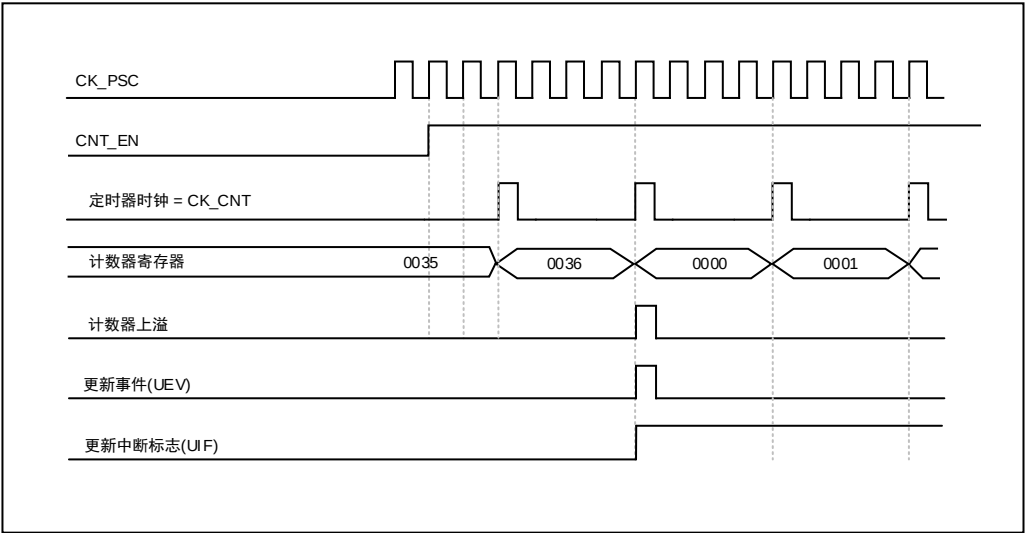


图 8-7 计数器时序图，N 分频内部时钟

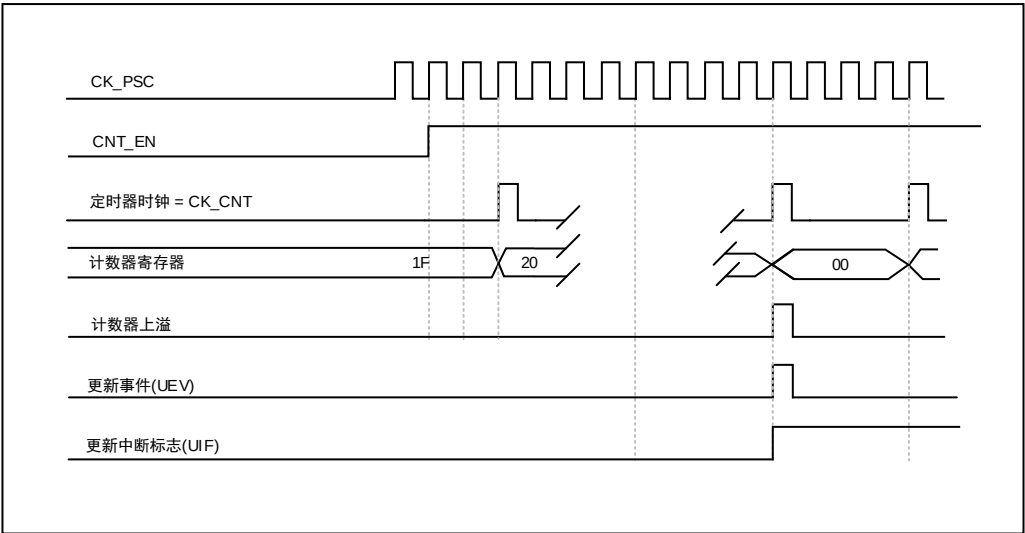


图 8-8 计数器时序图，ARPE=0 时更新事件（TIM14_ARR 未预装载）

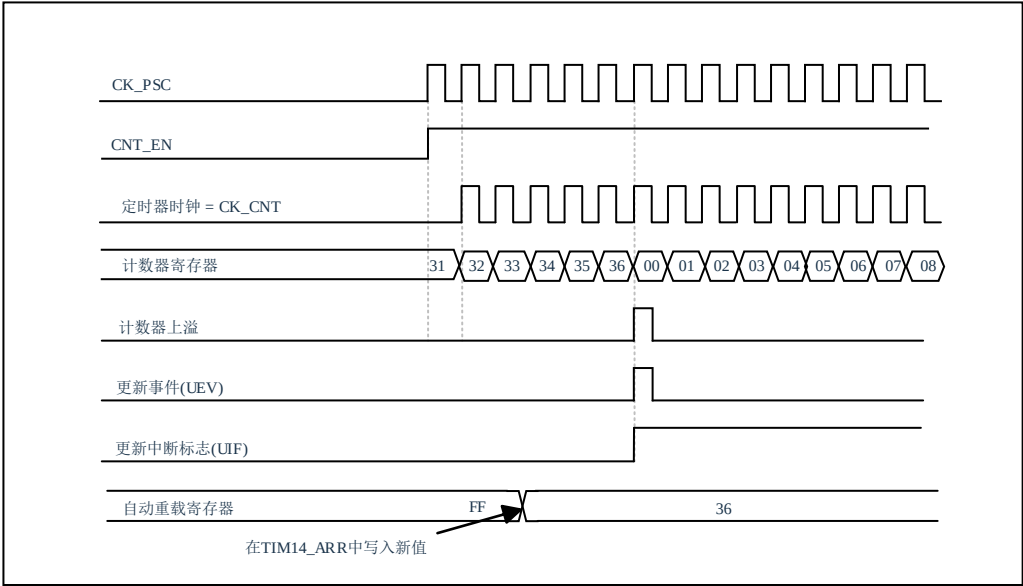
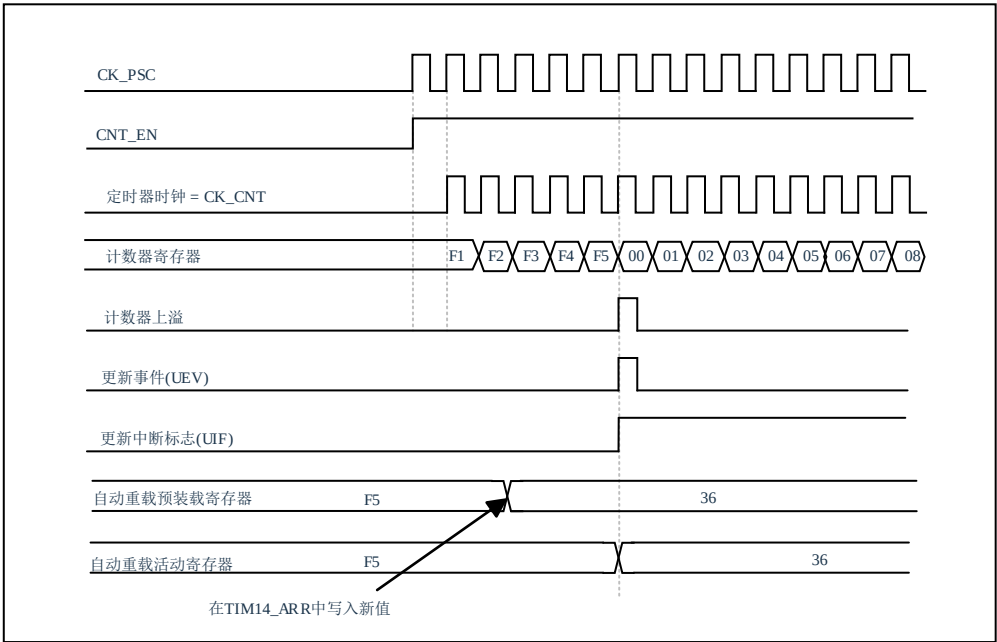


图 8-9 计数器时序图，ARPE=1 时更新事件（TIM14_ARR 预装载）

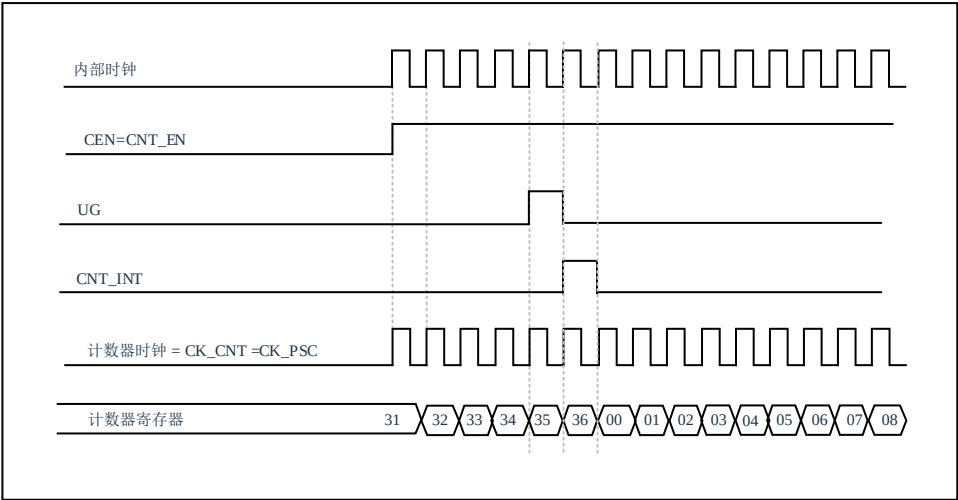


8.3.3 时钟选择

计数器的时钟由内部时钟（CK_INT）提供。TIM14_CR1 寄存器的 CEN 位和 TIM14_EGR 寄存器的 UG 位是实际的控制位（除了 UG 位被自动清除外），只能通过软件改变他们。一旦置 CEN 位为 1，内部时钟即向分频器提供时钟。

图 8-10 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

图 8-10 正常模式下的控制电路，1 分频内部时钟



8.3.4 捕获/比较通道

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。

图 8-11 到图 8-13 对捕获/比较通道做了简要说明。

输入阶段对相应的 TI1 输入进行采样，生成一个滤波后的信号 TI1F。然后，带有极性选择功能的边沿检测器生成一个信号 (TI1FP1)，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 (IC1PS)，而后再进入捕获寄存器。

图 8-11 捕获/比较通道（如：通道 1 输入阶段）

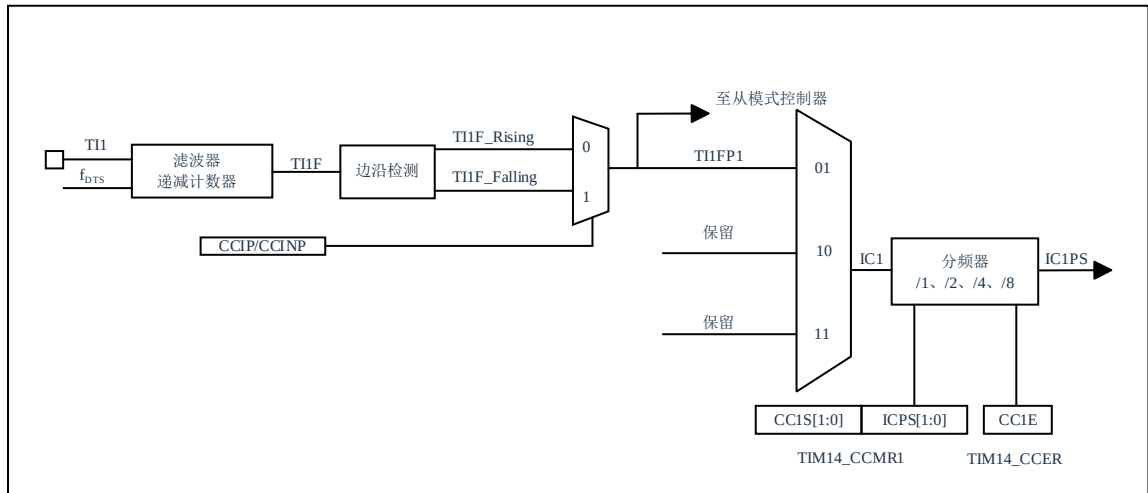


图 8-12 捕获/比较通道 1 主电路

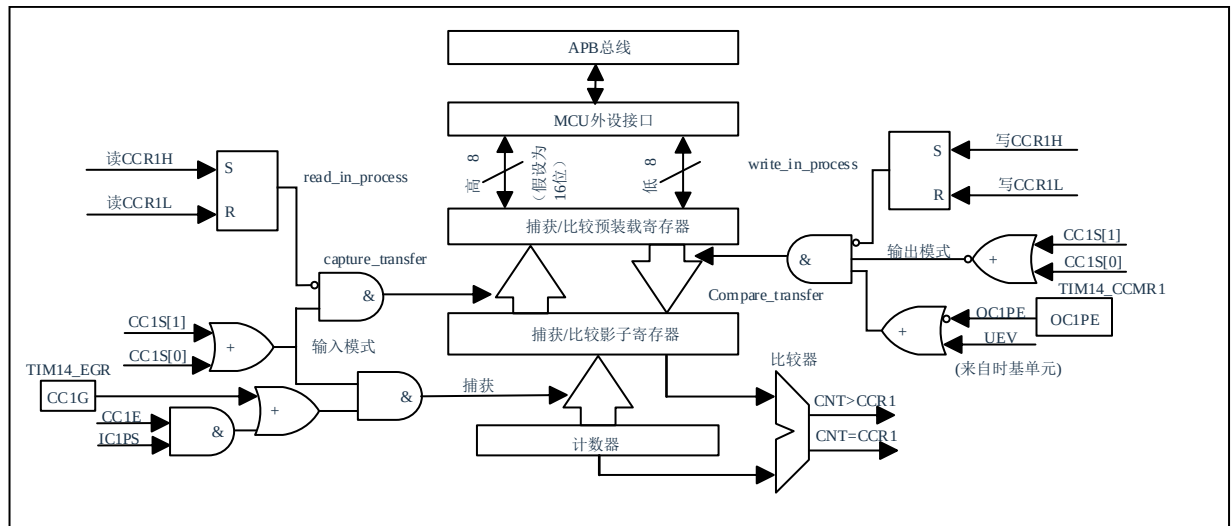
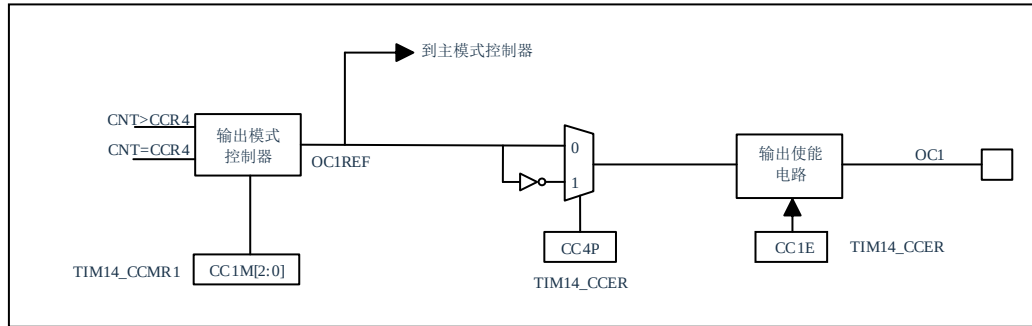


图 8-13 捕获/比较通道的输出阶段（通道 1）



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。

8.3.5 输入捕获模式

在输入捕获模式下，当相应的 IC1 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIM14_CCR1) 来锁存计数器的值。发生捕获事件时，会将相应的 CC1IF 标志 (TIM14_SR 寄存器) 置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CC1IF 标志已处于高位，则会将重复捕获标志 CC1OF (TIM14_SR 寄存器) 置 1。可通过软件向 CC1IF 写入 0 来给 CC1IF 清零，或读取存储在 TIM14_CCR1 寄存器中的已捕获数据。向 CC1OF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIM14_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIM14_CCR1 必须连接到 TI1 输入，因此向 TIM14_CCMR1 寄存器中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 TIM14_CCR1 寄存器将处于只读状态。
2. 根据输入信号的特点，配置输入滤波器为所需的宽度（如果输入为 TI1 输入之一，则对 TIM14_CCMR1 寄存器中的 IC1F 位进行编程）。假设输入信号最多在 5 个内部时钟周期内发生抖动，我们必须将滤波时间设置为大于 5 个内部时钟周期。可以设置检测到 8 个具有新电平连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIM14_CCMR1 寄存器中的 IC1F 位写入“0011”。
3. 通过在 TIM14_CCER 寄存器中将 CC1P 位和 CC1NP 位编程为“00”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
4. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIM14_CCMR1 寄存器中的 IC1PS 位写入“00”）。
5. 通过将 TIM14_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。如果需要，可通过将 TIM14_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIM14_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注意：通过软件将 TIM14_EGR 寄存器中的相应 CC1G 位置 1 可生成 IC 中断请求。

8.3.6 强制输出模式

在该模式（TIM14_CCMR1 寄存器中的 CC1S 位 =“00”）下，可直接由软件将每个输出比较信号（OC1REF 和 OC1）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 (OC1REF/OC1) 强制设置为有效电平，只需向相应 TIM14_CCMR1 寄存器中的 OC1M 位写入“101”。OC1REF 进而强制设置为高电平（OC1REF 始终为高电平有效），同时 OC1 获取 CC1P 极性位的相反值。

例如：CC1P=“0”（OC1 高电平有效）则 将 OC1 强制设置为高电平。

通过向 TIM14_CCMR1 寄存器中的 OC1M 位写入“100”，可将 OC1REF 信号强制设置为低电平。

在该模式下，TIM14_CCR1 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。因此可相应发送中断请求。下面的输出比较模式一节对此进行了介绍。

8.3.7 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 1. 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（TIM14_CCMR1 寄存器中的 OC1M 位）和输出极性（TIM14_CCER 寄存器中的 CC1P 位）定义。在比较匹配时，输出引脚既可保持其电平（OC1M=“000”），也可设置为有效电平（OC1M=“001”）、无效电平（OC1M=“010”）或进行翻转（OC1M=“011”）。
- 2. 将中断状态寄存器中的标志置 1（TIM14_SR 寄存器中的 CC1IF 位）。
- 3. 如果相应中断使能位（TIM14_DIER 寄存器中的 CC1IE 位）置 1，将生成中断。

使用 TIM14_CCMR1 寄存器中的 OC1PE 位，可将 TIM14_CCR1 寄存器配置为带或不带预装载寄存器。

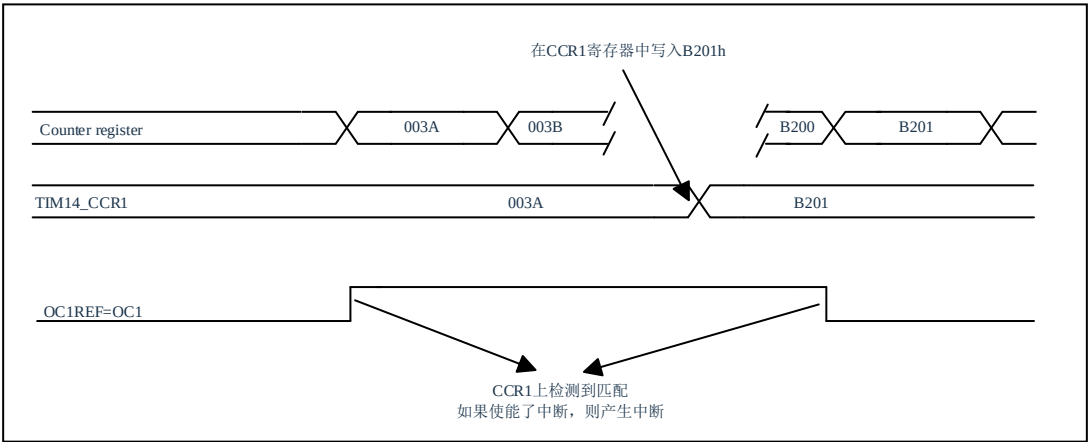
在输出比较模式下，更新事件 UEV 对 OC1REF 和 OC1 输出毫无影响。同步的精度可以达到计数器的一个计数周期。

步骤：

- 1. 选择计数器时钟（内部、外部、预分频器）。
- 2. 在 TIM14_ARR 和 TIM14_CCR1 寄存器中写入所需数据。
- 3. 如果要生成中断请求，则需将 CC1IE 位置 1。
- 4. 选择输出模式。例如：
 - 当 CNT 与 CCR1 匹配时，写入 OC1M=“011”以翻转 OC1 输出引脚
 - 写入 OC1PE=“0”以禁止预装载寄存器
 - 写入 CC1P=“0”以选择高电平有效极性
 - 写入 CC1E=“1”以使能输出
- 5. 通过将 TIM14_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIM14_CCR1 寄存器以控制输出波形，前提是未使能预加载寄存器（OC1PE=“0”，否则仅当发生下一个更新事件 UEV 时，才会更新 TIM14_CCR1 影子寄存器）。图 8-14 给出了一个示例。

图 8-14 输出比较模式，翻转 OC1



8.3.8 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIM14_ARR 寄存器值决定，其占空比则由 TIM14_CCR1 寄存器值决定。

通过向 TIM14_CCMR1 寄存器中的 OC1M 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 OC1 输出对应一个 PWM）的 PWM 模式。必须通过将 TIM14_CCMR1 寄存器中的 OC1PE 位置 1 使能相应预装载寄存器，最后通过将 TIM14_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在向上计数或中心对齐模式下）。

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 TIM14_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OC1 极性可使用 TIM14_CCER 寄存器的 CC1P 位来编程。既可以设为高电平有效，也可以设为低电平有效。OC1 输出通过将 TIM14_CCER 寄存器中的 CC1E 位置 1 来使能。

在 PWM 模式（1 或 2）下，TIM14_CNT 总是与 TIM14_CCR1 进行比较，以确定是 $TIM14\_CNT \leq TIM14\_CCR1$ 。

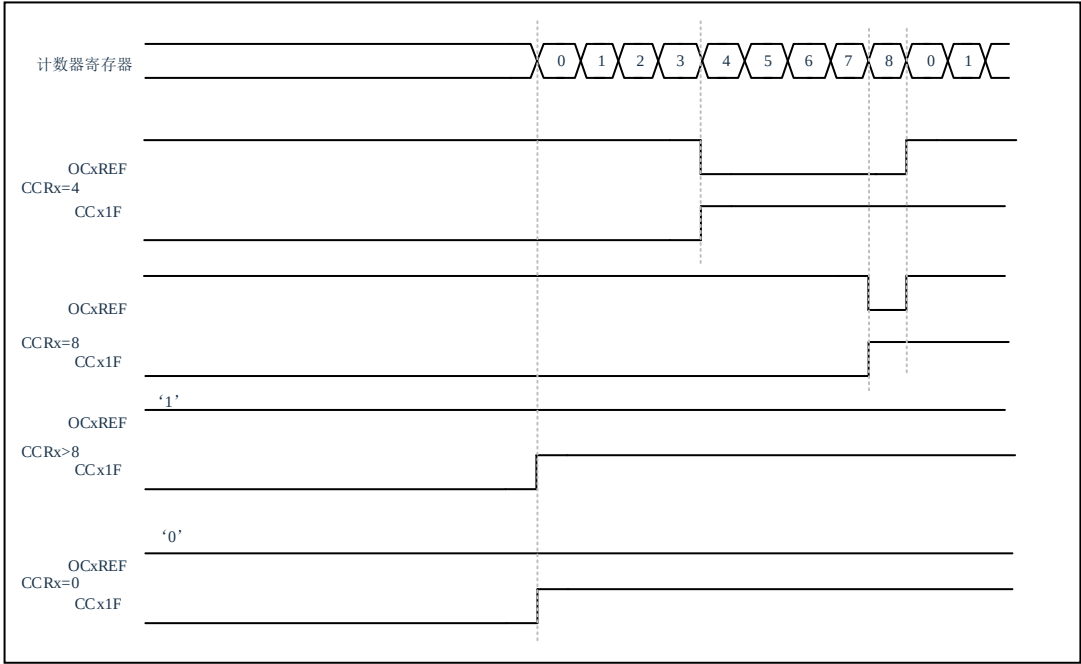
因为计数器采用向上计数方式计数，所以定时器能够在边沿对齐模式下生成 PWM。

8.3.8.1 PWM 边沿对齐模式

以下以 PWM 模式 1 为例。只要 $TIM14\_CNT < TIM14\_CCR1$ ，PWM 参考信号 OC1REF 便为高电平，否则为低电平。如果 TIM14_CCR1 中的比较值大于自动重载值（TIM14_ARR 中），则 OC1REF 保持为“1”。如果比较值为 0，则 OC1REF 保持为“0”。

图 8-15 举例介绍边沿对齐模式的一些 PWM 波形 (TIM14_ARR=8)。

图 8-15 边沿对齐模式的 PWM 波形（ARR=8）



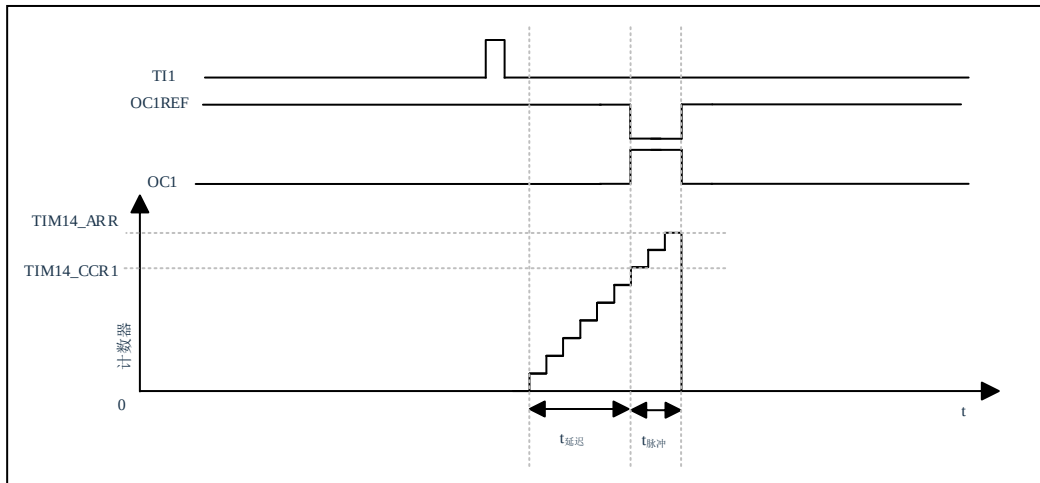
8.3.9 单脉冲模式

单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIM14_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置： $CNT < CCR1 \leq ARR$ （特别注意， $0 < CCR1$ ）

图 8-16 单脉冲模式示例



OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIM14_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值 (TIM14_ARR - TIM14_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIM14_CCMR1 寄存器中写入 OC1M=“111”，以使能 PWM 模式 2。如果需要，可选择在 TIM14_CCMR1 寄存器的 OC1PE 和 TIM14_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIM14_CCR1 寄存器中写入比较值并在 TIM14_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI1 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM14_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIM14_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OC1 快速使能

在单脉冲模式下，TI1 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值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 TIM14_CCMR1 寄存器中的 OC1FE 位置 1。这样会强制 OC1REF（和 OC1）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，OC1FE 才会起作用。

8.3.10 调试模式

当芯片进入调试模式时，根据 1.4 中的设置，TIM14 计数器可以继续正常工作或者停止工作。

8.4 寄存器映射

(TIM14 基地址 = 0x4006_2000) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
CR1	0x00	R/W	TIM14 控制寄存器 1	0x00
DIER	0x0C	R/W	TIM14 DMA/中断使能寄存器	0x00
SR	0x10	R/W	TIM14 状态寄存器	0x00
EGR	0x14	R/W	TIM14 事件生成寄存器	0x00
CCMR1	0x18	R/W	TIM14 捕获/比较模式寄存器 1	0x00
CCER	0x20	R/W	TIM14 捕获/比较使能寄存器	0x00
CNT	0x24	R/W	TIM14 计数器	0x00
PSC	0x28	R/W	TIM14 预分频器	0x00
ARR	0x2C	R/W	TIM14 自动重载寄存器	0xFFFF
CCR1	0x34	R/W	TIM14 捕获/比较寄存器 1	0x00
OR	0x50	R/W	TIM14 选项寄存器	0x00

8.5 寄存器说明

8.5.1 TIM14 控制寄存器 1 (TIM14_CR1)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9:8	CKD[1:0]	时钟分频 (Clock division) 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (ETR、Tl _x) 所使用的死区及采样时钟 (t _{DTS}) 之间的分频比 00: t _{DTS} = t _{CK_INT} 01: t _{DTS} = 2 * t _{CK_INT} 10: t _{DTS} = 4 * t _{CK_INT} 11: 保留，不要设置成此值	0x0
7	ARPE	自动重载预装载使能 0: TIM14_ARR 寄存器不进行缓冲 1: TIM14_ARR 寄存器进行缓冲	0
6:4	--	保留，须为 0	0x0
3	OPM	单脉冲模式： 0: 计数器在发生更新事件时不会停止计数 1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止	0
2	URS	更新请求源 此位由软件置 1 和清零，用以选择 UEV 事件源 0: 如果允许产生更新中断或 DMA 请求，则下述任一事件产生一个更新中断或 DMA 请求 — 计数器上溢/下溢 — 设置 UG 位 1: 如果允许产生更新中断或 DMA 请求，则只有计数器上溢/下溢产生一个更新中断或 DMA 请求	0
1	UDIS	更新禁止 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0: 允许 UEV。更新 (UEV) 事件可通过以下事件之一生成 — 计数器上溢/下溢 — 设置 UG 位 被缓存的寄存器被装入它们的预装载值 1: 禁止 UEV。不产生更新事件，影子寄存器 (ARR,PSC,CCR _x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。	0
0	CEN 计数器使能	计数器使能 0: 禁止计数器 1: 开启计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。	0

8.5.2 TIM14 中断使能寄存器 (TIM14_DIER)

位	符号	描述	复位值
31:2	--	保留，须为 0	0x0
1	CC1IE	CC1IE: 捕获/比较 1 中断使能 0: 禁止 CC1 中断 1: 使能 CC1 中断	0
0	UIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断	0

8.5.3 TIM14 状态寄存器 (TIM14_SR)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9	CC1OF	捕获/比较 1 过捕获标志 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“0”可将该位清零 0: 未检测到过捕获。 1: TIM14_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1	0
8:2	--	保留，须为 0	0x0
1	CC1IF	捕获/比较 1 中断标志 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。通过软件清零。 0: 无匹配发生。 1: TIM14_CNT 的值与 TIM14_CCR1 的值匹配 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 TIM14_CCR1 寄存器将该位清零。 0: 未发生输入捕获事件 1: 输入捕获产生并且计数器值已装入 TIM14_CCR1(在 IC1 上检测到与所选极性相同的边沿)	0
0	UIF	更新中断标志 该位在发生更新事件时通过硬件置 1。需要通过软件清零 0: 未发生更新 1: 更新中断等待响应。该位在以下情况下更新寄存器时由硬件置 1 – 若 TIM14_CR1 寄存器的 UDIS=0，产生更新事件上溢 – 若 TIM14_CR1 寄存器的 UDIS=0、URS=0，当 TIM1_EGR 寄存器的 UG=1 时产生更新事件(软件对 CNT 重新初始化)；	0

8.5.4 TIM14 事件生成寄存器 (TIM14_EGR)

位	符号	描述	复位值
31:2	--	保留，须为 0	0x0
1	CC1G	捕获/比较 1 生成 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 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断请求。 若通道 CC1 配置为输入： 当前的计数器值捕获至 TIM14_CCR1 寄存器，设 CC1IF=1，若开启对应的中断，则产生相应的中断请求。 若 CC1IF 已经为 1，则设置 CC1OF=1。	0
0	UG	更新生成 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一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。	0

8.5.5 TIM14 捕获/比较模式寄存器 1 (TIM14_CCMR1)

输出比较模式

位	符号	描述	复位值
31:7	--	保留，须为 0	0x0
6:4	OC1M	输出比较 1 模式 这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 000: 冻结。输出比较寄存器 TIM14_CCR1 与计数器 TIM14_CNT 间的比较对 OC1REF 不起作用 001: 匹配时设置通道 1 为有效电平。当计数器 TIM14_CNT 的值与捕获 / 比较寄存器 1(TIM14_CCR1) 相同时，强制 OC1REF 为高 010: 匹配时设置通道 1 为无效电平。当计数器 TIM14_CNT 的值与捕获 / 比较寄存器 1(TIM14_CCR1)相同时，强制 OC1REF 为低 011: 翻转。当 TIM1_CCR1=TIM1_CNT 时，翻转 OC1REF 的电平 100: 强制为无效电平。强制 OC1REF 为低 101: 强制为有效电平。强制 OC1REF 为高 110: PWM 模式 1 — TIM14_CNT<TIM14_CCR1 时，通道 1 为有效电平，否则为无效电平。 111: PWM 模式 2 — TIM14_CNT<TIM14_CCR1 时，通道 1 为无效电平，否则为有效电平。 注 1: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变	0x0
3	OC1PE	输出比较 1 预装载使能 0: 禁止 TIM14_CCR1 寄存器的预装载功能，可随时写入 TIM14_CCR1 寄存器，且新值马上起作用 1: 开启 TIM14_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM14_CCR1 的预装载值在更新事件到来时被载入当前寄存器中 注 1: 仅在单脉冲模式下，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定	0
2	OC1FE	输出比较 1 快速使能 此位用于加快触发输入事件对 CC 输出的影响 0: 根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OC1FE 的只在通道被配置成 PWM1 或 PWM2 模式时起作用	0
1:0	CC1S	捕获/比较 1 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC1 通道配置为输出 01: CC1 通道配置为输入，IC1 映射到 TI1 上 10: 保留 11: 保留 注意: 仅当通道关闭时（TIM14_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据	0

输入捕获模式

位	符号	描述	复位值
31:8	--	保留，须为 0	0x0
7:4	IC1F[3:0]	输入捕获 1 滤波器 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。 数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿 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: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	0x0
3:2	IC1PSC	输入捕获 1 预分频器 此位域定义 CC1 输入 (IC1) 的预分频比。 只要 CC1E=“0” (TIM14_CCER 寄存器)，预分频器便立即复位 00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获。 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获。 11: 每发生 8 个事件便执行一次捕获。	0x0
1:0	CC1S	捕获/比较 1 选择 00: CC1 通道配置为输出 01: CC1 通道配置为输入，IC1 映射到 TI1 上 10: 保留 11: 保留 注意：仅当通道关闭时 (TIM14_CCER 中的 CC1E =“0”)，才可向 CC1S 位写入数据	0x0

8.5.6 TIM14 捕获/比较使能寄存器 (TIM14_CCER)

位	符号	描述	复位值
31:4	--	保留,须为 0	0x0
3	CC1NP	捕获/比较 1 互补输出极性 CC1 通道配置为输出: CC1NP 必须保持 0 CC1 通道配置为输入: 此位与 CC1P 配合使用,用以定义 TI1FP1 极性。请参见 CC1P 说明。	0
2	---	保留,须为 0	0
1	CC1P	捕获/比较 1 输出极性 CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 的有效极性 00: 非反相/上升沿触发 电路对 TI1FP1 上升沿敏感(在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TI1FP1 未反相(在门控模式或编码器模式下执行触发操作) 01: 反相/下降沿触发 电路对 TI1FP1 下降沿敏感(在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TI1FP1 反相(在门控模式或编码器模式下执行触发操作) 10: 保留,不使用此配置 11: 未反相/边沿触发 电路对 TI1FP1 上升沿和下降沿都敏感(在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TI1FP1 未反相(在门控模式下执行触发操作)。	0
0	CC1E	捕获 / 比较 1 输出使能 CC1 通道配置为输出: 0: 关闭 —OC1 输出禁止 1: 开启 —OC1 信号是相应输出引脚上的输出 CC1 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 1 (TIM14_CCR1) 中。 0: 禁止捕获。 1: 使能捕获。	0

表 8-1 标准 OC1 通道的输出控制位

CC1E 位	OC1 输出状态
0	禁止输出 (OC1= “0”, OC1NE= “0”)
1	OC1=OC1REF + 极性, OC1EN = “1”

8.5.7 TIM14 计数器 (TIM14_CNT)

位	符号	描述	复位值
31:16	--	保留, 须为 0	0x0
15:0	CNT[15:0]	计数器值 (Counter value)	0x0

8.5.8 TIM14 预分频器 (TIM14_PSC)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	PSC[15:0]	预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ PSC 包含每次发生更新事件时要装载到活动预分频器寄存器的值。	0x0

8.5.9 TIM14 自动重载寄存器 (TIM14_ARR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	ARR[15:0]	自动重载值 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。	0xFFFF

8.5.10 TIM14 捕获/比较寄存器 1 (TIM14_CCR1)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR1[15:0]	捕获/比较 1 值 如果通道 CC1 配置为输出： CCR1 是捕获/比较寄存器 1 的预装载值。 如果没有通过 TIM14_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。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 1）。实际捕获/比较寄存器中包含要与计数器 TIM14_CNT 进行比较并在 OC1 输出上发出信号的值。 如果通道 CC1 配置为输入： CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值	0x0

8.5.11 TIM14 选项寄存器(TIM14_OR)

位	符号	描述	复位值
31:2	--	保留，须为 0	0x0
1:0	TI1_RMP	定时器输入 1 重映射 通过软件置位和清零。 00: TIM14 通道 1 连接到 GPIO,具体参考数据手册的复用功能。 其他: 保留	0x0

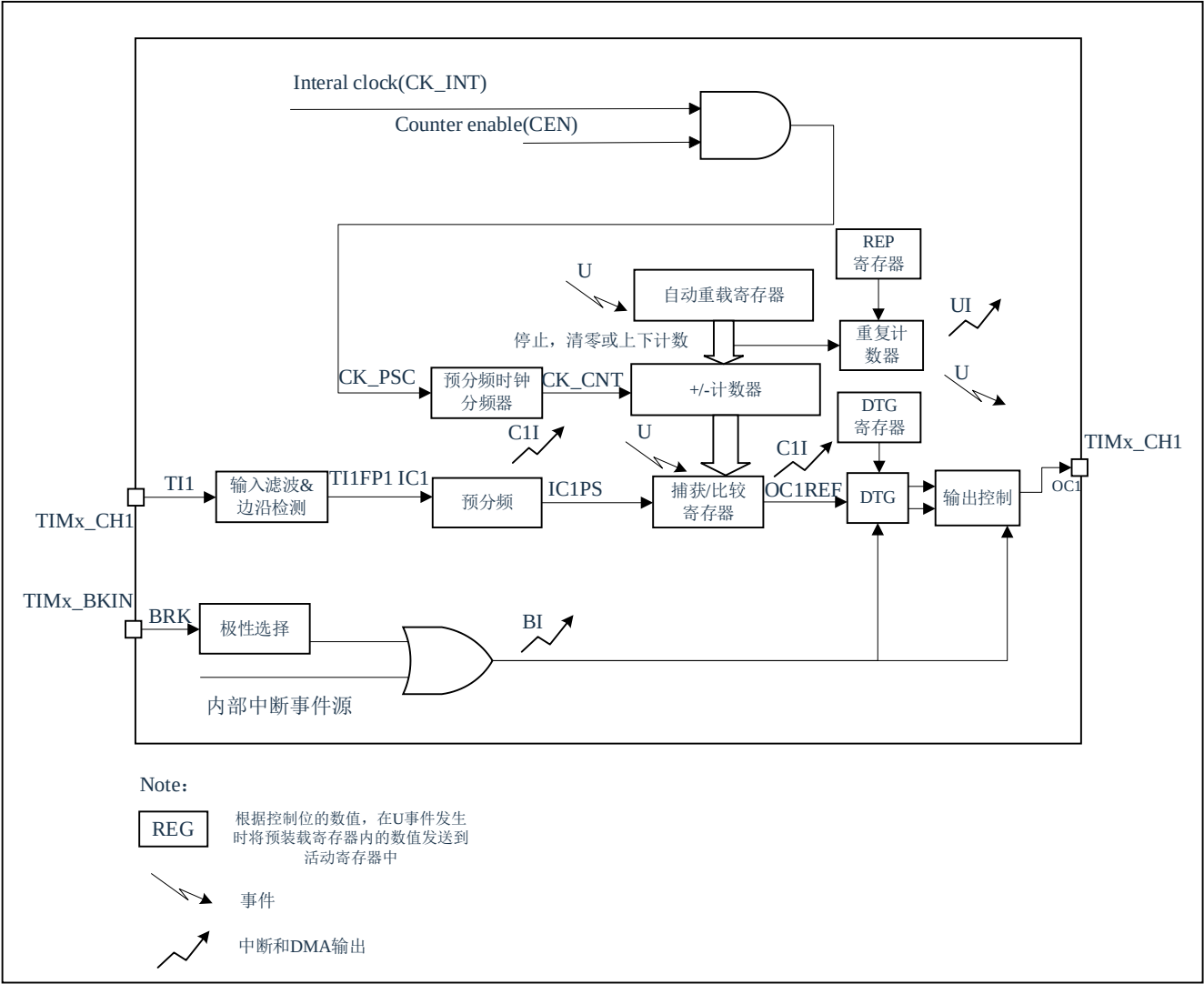
第9章 通用定时器 (TIM16/17)

TIM16 和 TIM17 功能完全一样。

9.1 TIM16 /TIM17 主要特性

- ◆ 位自动装载向上计数器
- ◆ 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值
- ◆ 一个通道作为：
 - 输入捕获
 - 输出比较
 - PWM 生成（边缘对齐模式）
 - 单脉冲模式输出
- ◆ 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- ◆ 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- ◆ 如下时间发生时产生中断/DMA：
 - 更新：计数器上溢
 - 输入捕获
 - 输出比较
 - 刹车信号输入

图 9-1通用定时器框图



9.2 TIM16/TIM17 功能描述

9.2.1 时基单元

这个可编程定时器的主要部分是一个带有自动重载的 16 位的向上计数器，计数器的时钟通过一个预分频器得到。

软件可以读写计数器、自动重载寄存器和预分频寄存器，即使计数器运行时也可以操作。时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频寄存器 (TIMx_PSC)
- 自动重载 (TIMx_ARR)
- 重复计数寄存器 (TIMx_RCR)

自动装载寄存器是预先装载的，写或者读自动重载寄存器 (TIMx_ARR) 将访问预装载寄存器 (preload register)。根据 TIMx_CR1 寄存器中的自动装载使能位 (ARPE) 的设置，预装载寄存器的内容可以被一直或者在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到上溢出并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。

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

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

预分频描述：

预分频器可以将计数器的时钟按 1 到 65535 之间的任意值分频。它是基于一个（在 TIMx_PSC 寄存器中的）16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

下图给出了在预分频器运行时，更改计数器参数的例子。

图 9-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

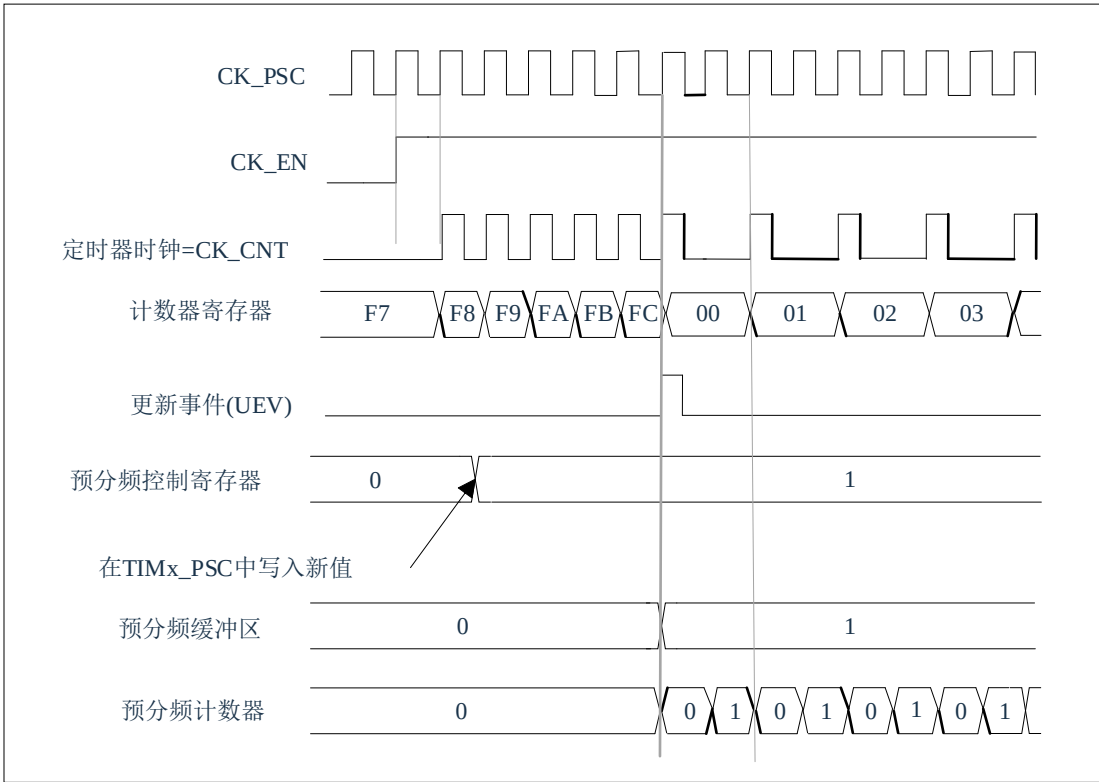
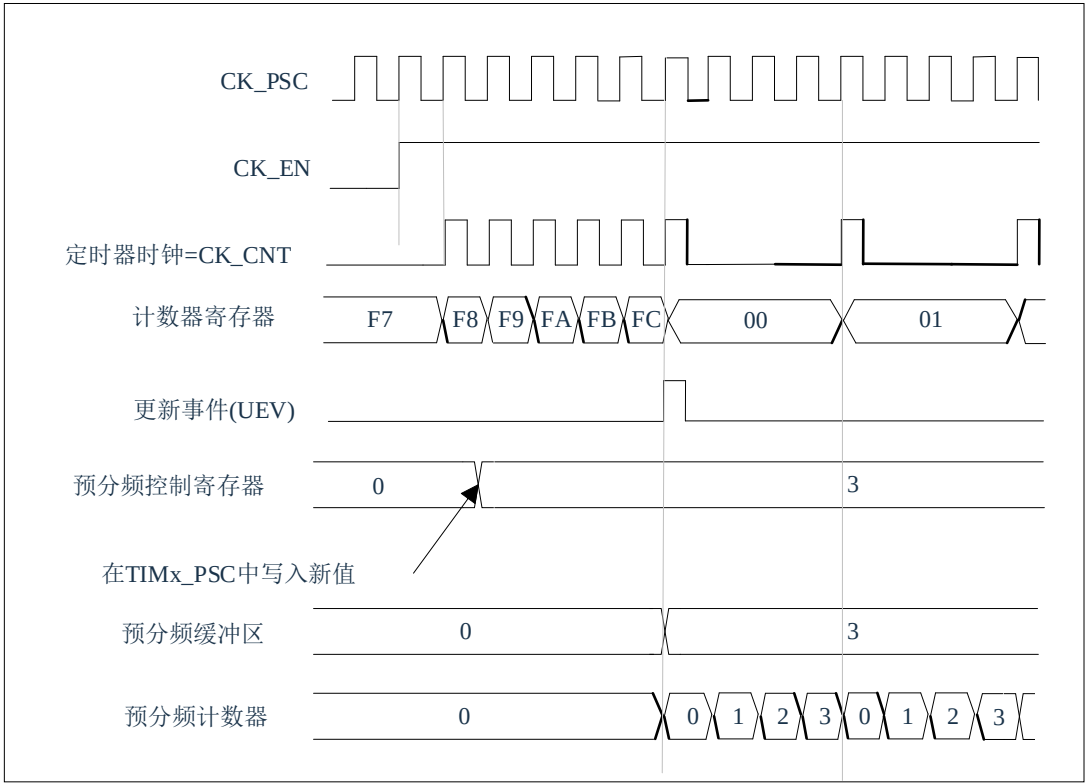


图 9-3 当预分频器的参数从 1 变到 4 时，计数器的时序图



9.2.2 计数器模式

计数器从 0 计数到自动装载值（TIMx_ARR 寄存的值），然后又从 0 重新开始计数，并产生一个计数器溢出事件。

如果重复计数器被使用，则在向上计数器重复几次（对重复计数器可编程）后，产生更新事件。否则，在每个计数溢出时，产生更新事件。

在 TIMx_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位也同样可以产生一个更新事件。设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样也可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清零之前，将不产生更新事件。虽然如此，但是计数器依旧从 0 开始，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。当发生一个更新事件时，所有的寄存器都被更新，硬件同时(依据 URS 位)设置更新标志位(TIMx_SR 寄存器中的 UIF 位)。

- 重复计数器被 TIMx_RCR 寄存器中的值重新装载。
- 自动装载影子寄存器被重新置入预装载寄存器的值(TIMx_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值(TIMx_PSC 寄存器的内容)。下图显示了几个在不同频率下的计数器行为，当 TIMx_ARR=0X36。

图 9-4 计数器时序图，内部时钟分频因子为 1

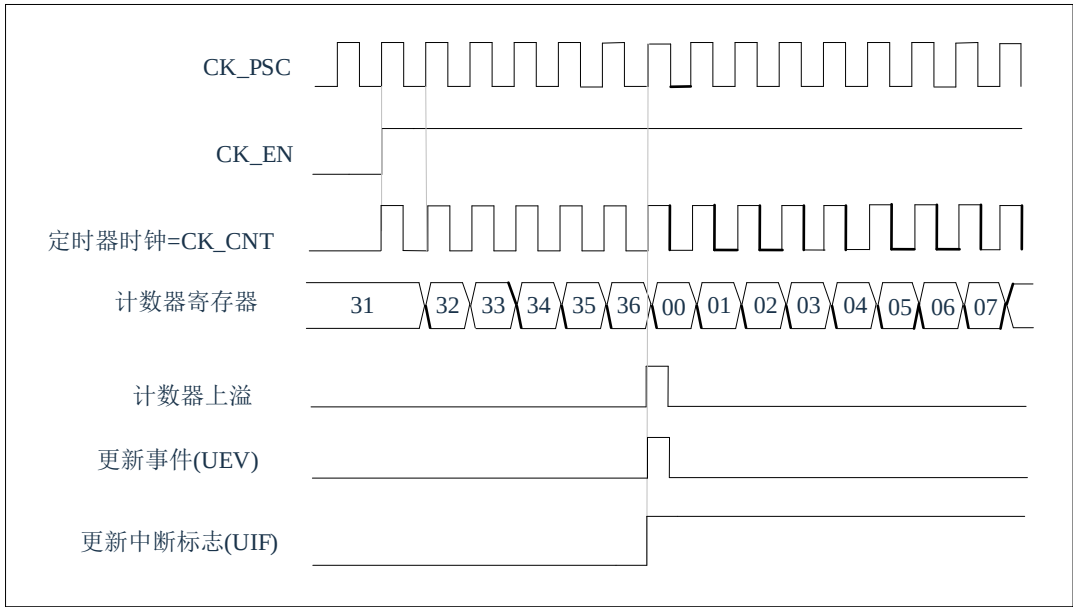


图 9-5 计数器时序图，内部时钟分频因子为 2

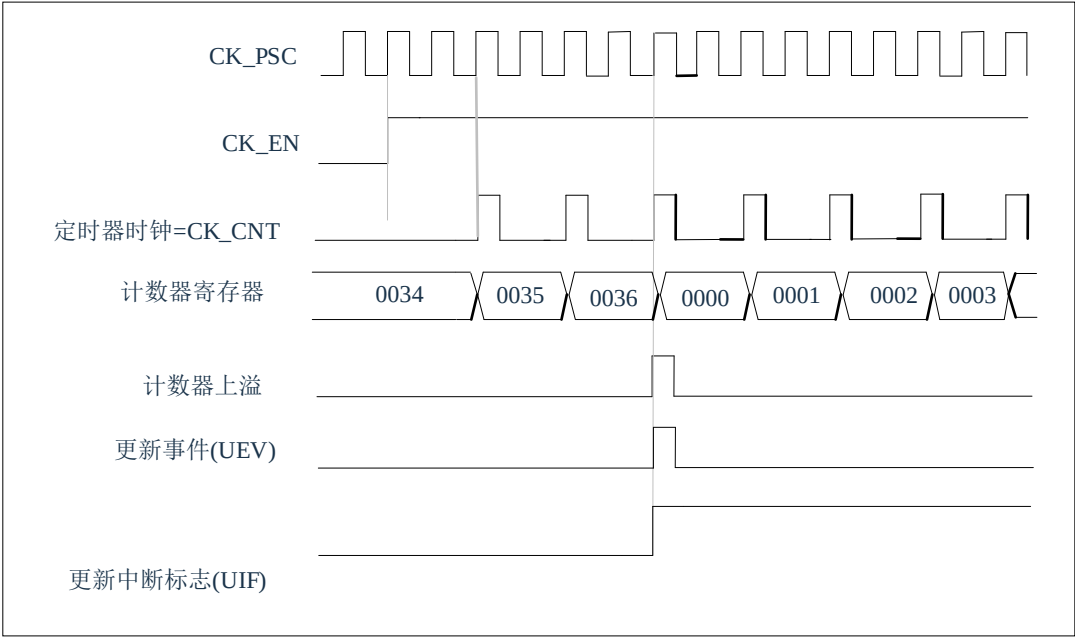


图 9-6 计数器时序图，内部时钟分频因子为 4

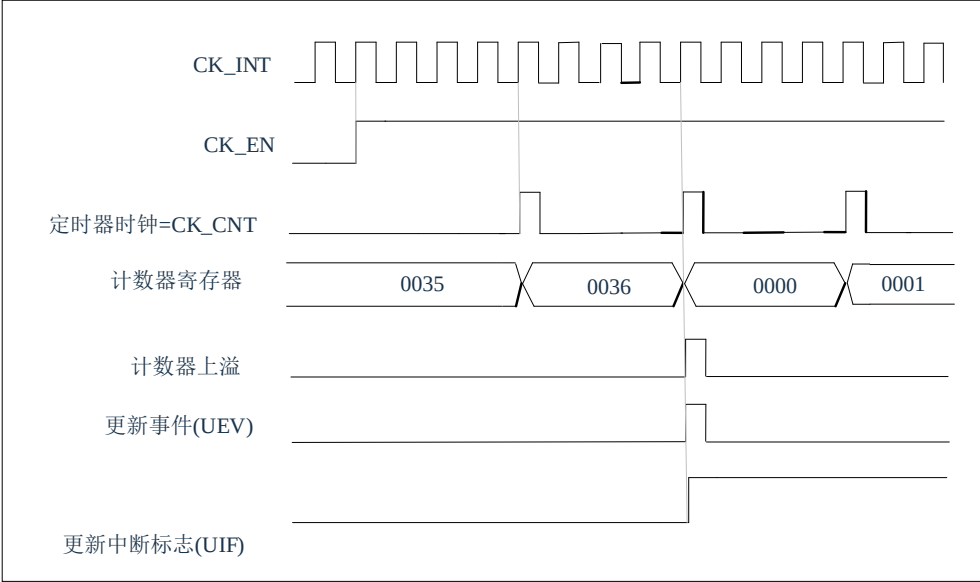


图 9-7 计数器时序图，内部时钟分频因子为 N

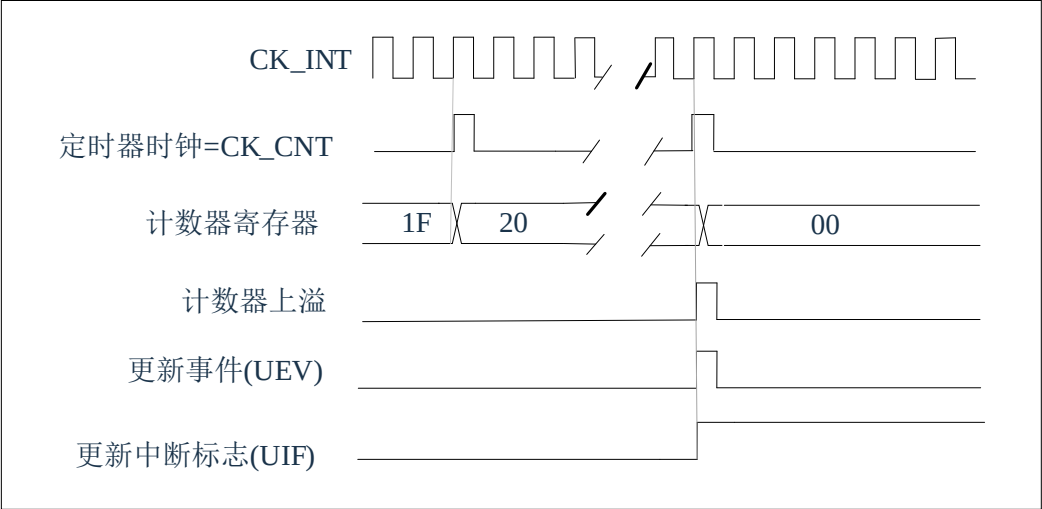


图 9-8 计数器时序图，当 ARPE=0 时的更新事件（TIMx_ARR 没有预装入）

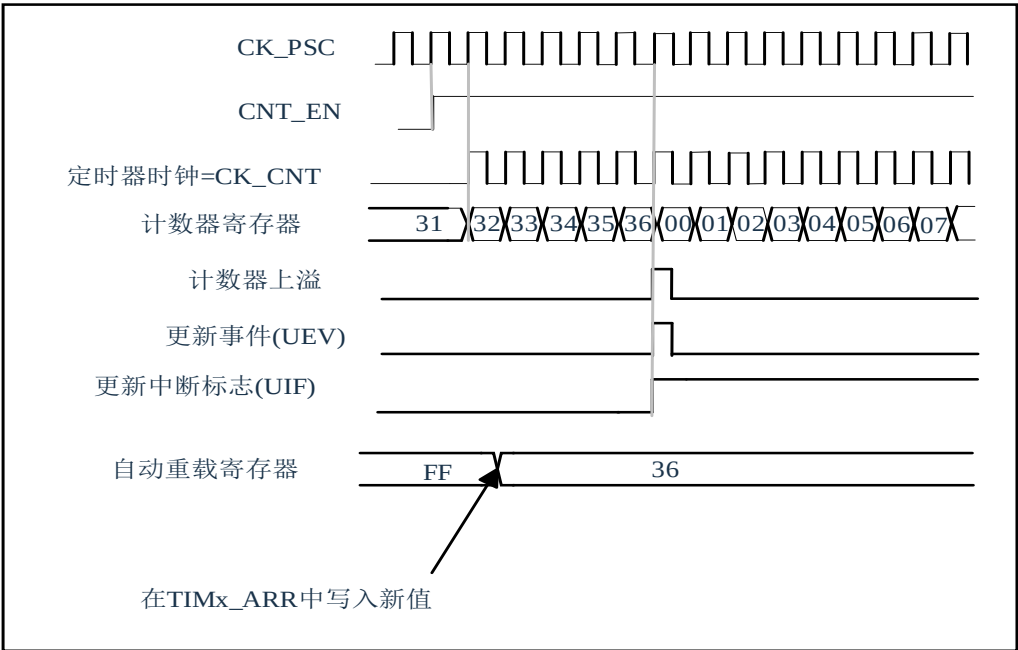
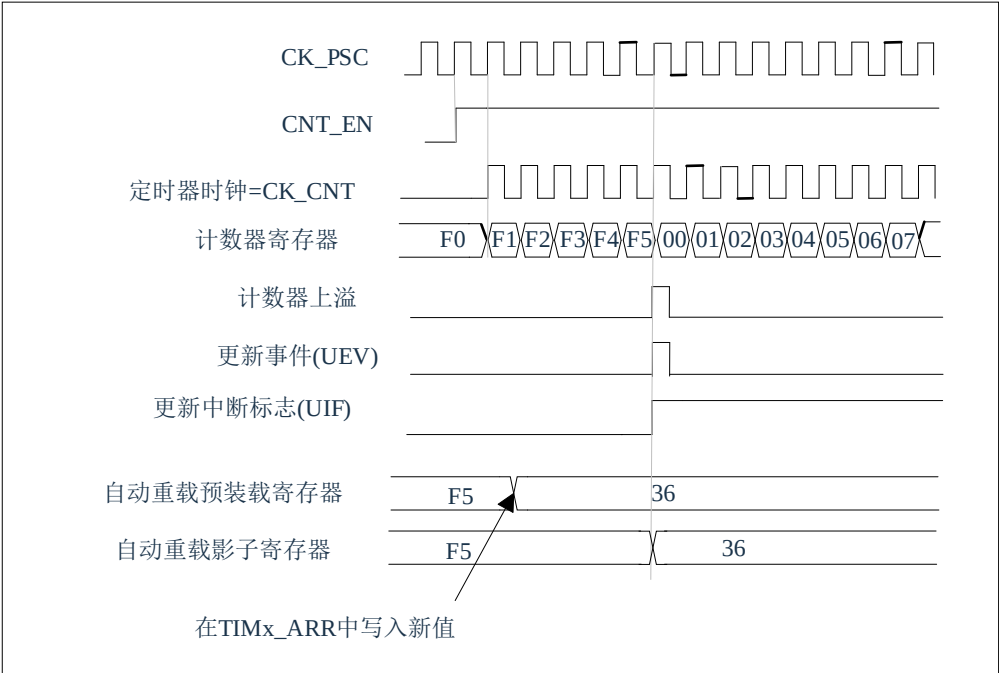


图 9-9 计数器时序图，当 ARPE=1 时的更新事件（TIMx_ARR 没有预装入）



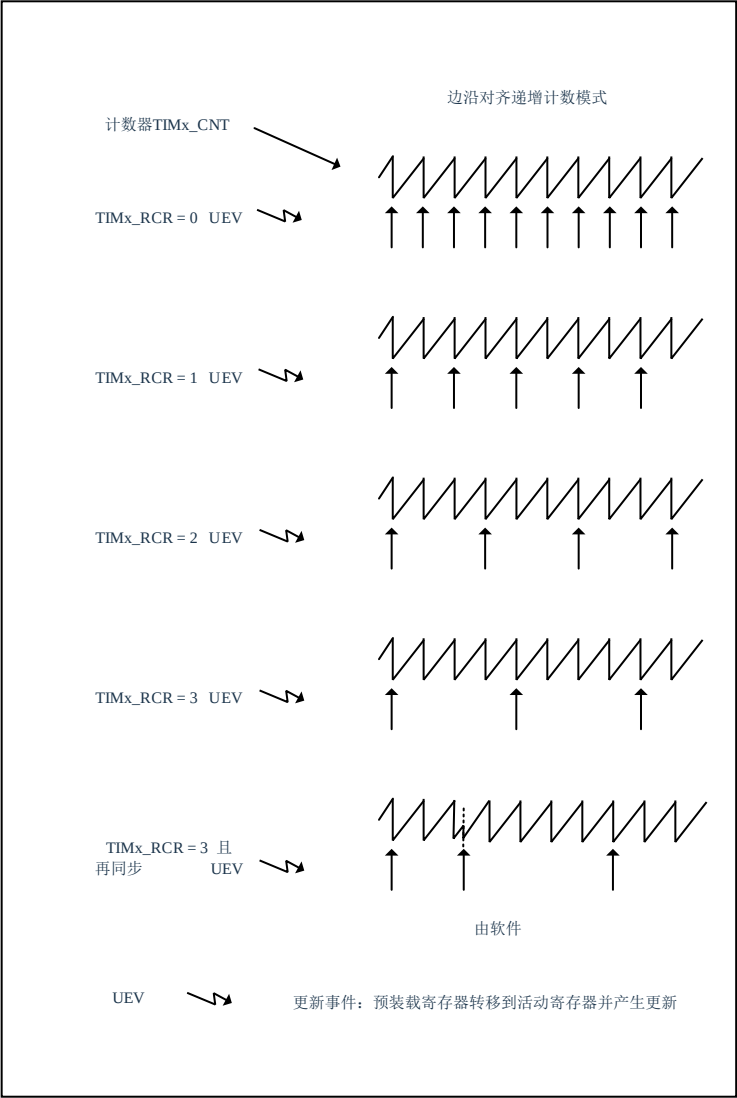
9.2.3 重复计数器

时基单元描述了计数器上溢时更新事件（UEV）是如何产生的，然而实际上它只能在重复计数器达到 0 的时候产生。这个特性对产生 PWM 信号有用。

这意味着在每 N 次计数上溢时，数据从预装载寄存器传输到影子寄存器（TIMx_ARR 自动重载寄存器，TIMx_PSC 预分频寄存器，还有在比较模式下的捕获/比较寄存器 TIMx_CCRx），N 是 TIMx_RCR 重复计数器寄存器中的值。重复计数器，在向上计数器模式的每个计数上溢时递减。

重复计数器是自动重载的，重复速率是由 TIMx_RCR 寄存器的值定义。当更新事件由软件（通过设置 TIMx_EGR 中的 UG 位）或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMx_RCR 寄存器中的内容被重载到重复计数器中。

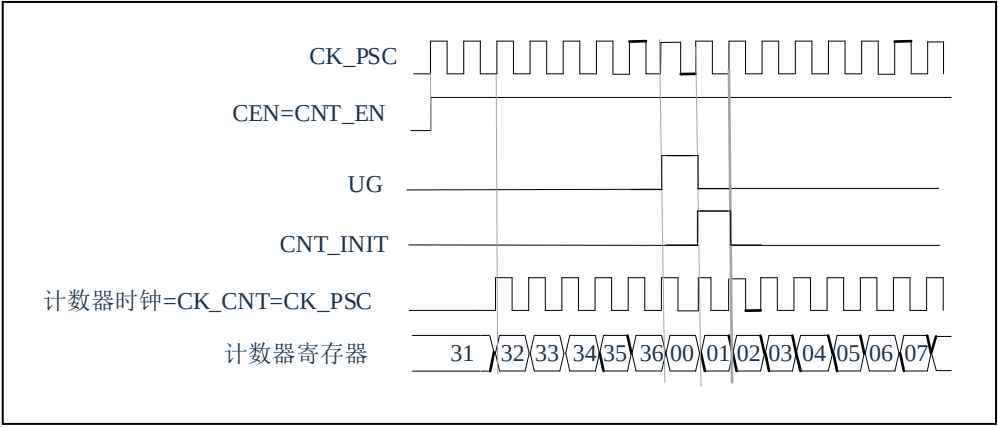
图 9-10 不同模式下更新速率的例子，及 TIMx_RCR 的寄存器设置



9.2.4 时钟源

计数器的时钟由内部时钟（CK_INT）提供。TIMx_CR1 寄存器的 CEN 位和 TIMx_EGR 寄存器的 UG 位是实际的控制位（除了 UG 位被自动清除外），只能通过软件改变他们。一旦置 CEN 位为 1，内部时钟即向分频器提供时钟。

图 9-11 一般模式下的控制电路，内部时钟分频因子为 1



9.2.5 捕捉/比较通道

每一个捕捉/比较通道都是围绕着一个捕捉/比较寄存器(包含影子寄存器)，包括捕捉的输入部分(数字滤波、多路复用和预分频器)，和输出部分(比较器和输出控制)。

下面几张图是一个捕捉/比较通道概览。

输入部分对相应的 TI1 输入信号采样，并产生一个滤波后的信号 TI1F。然后，一个带极性选择的边缘检测器产生一个信号(TI1FP1)，它可以作为从模式控制器的输入触发或者作为捕捉控制。该信号通过预分频进入捕捉寄存器(IC1PS)。

图 9-12 捕捉/比较通道（如：通道 1 输入部分）

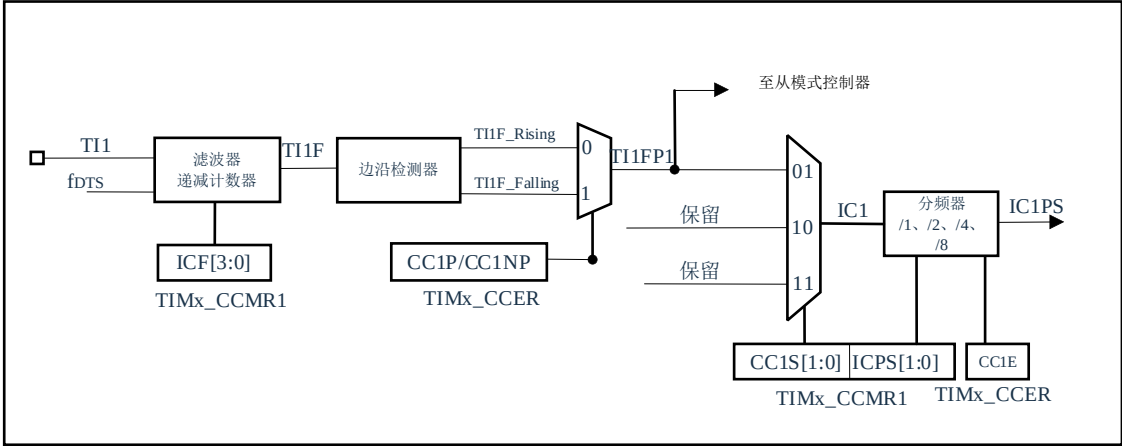


图 9-13 捕获/比较通道 1 的主电路

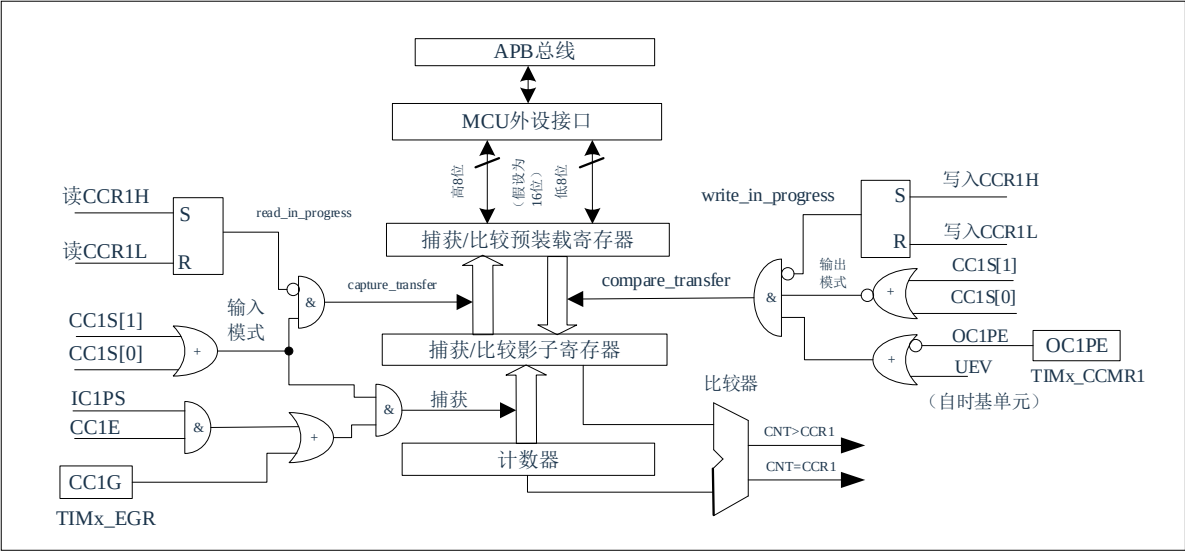
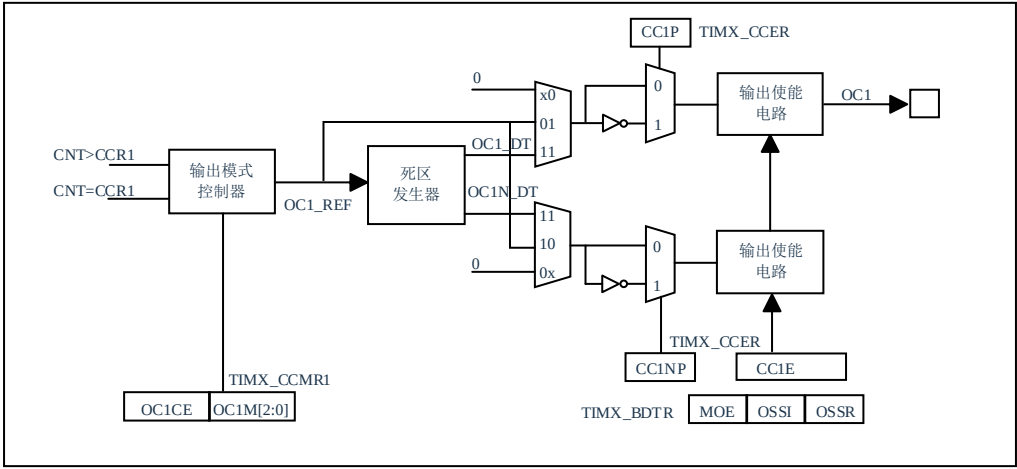


图 9-14 捕获/比较通道的输出部分



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。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再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被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

9.2.6 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCR1）中。当捕获事件发生时，相应的 CCxIF 标志（TIMx_SR 寄存器）被置 1。如果捕获事件发生时，CCxIF 标志已经为高，那么重复捕获标志 CCxOF（TIMx_SR 寄存器）被置 1。写 CCxIF 可清除 CCxIF，或读取存储中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

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

- 选择有效输入端：TIMx_CCR1 必须连接到 TI1 输入，所以写入 TIMx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为 00 时，通道被配置为输入，并且 TIMx_CCMR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的宽度（即输入为 TI1 时，输入滤波器控制位是 TIMx_CCMR1 寄存器中的 ICxF 位）。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们必须配置滤波器的宽度长于 5 个时钟周期。因此，我们可（以 f_{DTS} 频率）连续采样 8 次，已确认在 TI1 上一次真是的边沿变化，然后在 TIMx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0（上升沿）。
- 配置输入预分频器。在这个例子中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PS=00）。
- 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。

如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 个连续捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断请求。
- 如设置了 CC1DE 位，则会产生一个 DMA 请求。

为了处理中断溢出，建议在读出中断溢出标志之前，读取数据。这是为了避免丢失在读出中断溢出标志之后和读取数据之前可能产生的中断溢出信息。

注：输入捕获中断请求能通过软件设置在 TIMx_EGR 中相应的 CCxG 位来产生。

9.2.7 强置输出模式

在该模式下（TIMx_CCMR1 寄存器中 CC1S bits = 00）下，输出比较信号（OC1REF 和相应的 OC1）能够直接由软件置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

写 TIMx_CCMR1 寄存器中相应的 OC1M=101，即可强制输出比较信号（OC1REF/OC1）为有效状态。这样 OC1REF 被强置为高电平（OC1REF 始终为高电平有效），同时 OC1 得到 CC1P 极性位相反的值。

例如：CC1P=0(OC11 高电平有效)，则 OC1 被强置为高电平。

置 TIMx_CCMR1 寄存器中的 OC1M=100，可强置 OC1REF 信号为低。

该模式下，在 TIMx_CCR1 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

9.2.8 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较工作做如下操作：

- 将输出比较模式(TIMx_CCMR1 寄存器中的 OC1M 位)和输出极性(TIMx_CCER 寄存器中的 CC1P 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OC1M=000)、被设置成有效电平(OC1M=001)、被设置成无效电平(OC1M=010)或进行翻转(OC1M=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的 CC1IF 位)。
- 若设置了相应的中断屏蔽(TIMx_DIER 寄存器中的 CC1IE 位)，则产生一个中断。TIMx_CCMR1 中的 OC1PE 位选择 TIMx_CCR1 寄存器是否需要使用预装载寄存器。

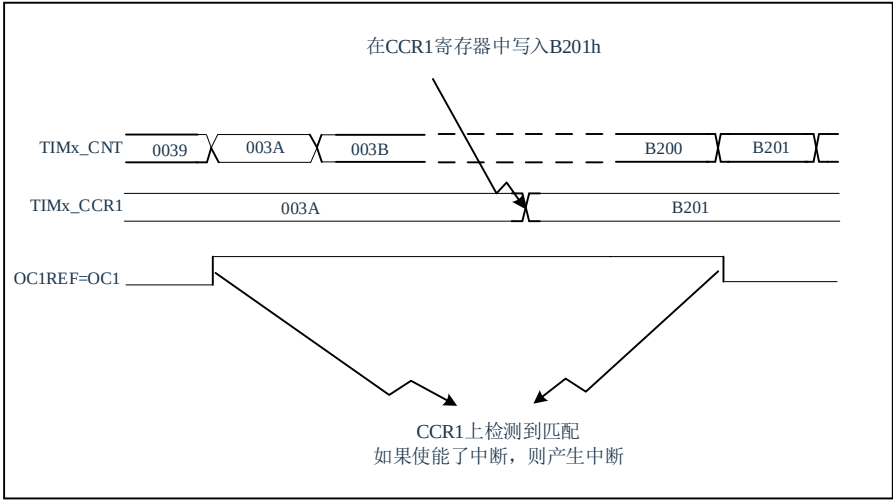
在输出比较模式下，更新事件 UEV 对 OC1REF 和 OC1 输出没有影响。同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCR1 寄存器中。
3. 如果要产生一个中断请求，设置 CC1IE 位。
4. 选择输出模式，例如：
 - 要求计数器与 CCR1 匹配时翻转 OC1 的输出引脚，设置 OC1M=011
 - 置 OC1PE = 0 禁用预装载寄存器
 - 置 CC1P = 0 选择极性为高电平有效
 - 置 CC1E = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCR1 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OC1PE='0'，否则 TIMx_CCR1 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 9-15 输出比较模式，OC1 上翻转



9.2.9 PWM（脉冲宽度调节）模式

脉冲宽度调节模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCR1 寄存器确定占空比的信号。

在 TIMx_CCMR1 寄存器中的 OC1M 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2），能够独立地设置 OC1 输出通道产生一路 PWM。必须设置 TIMx_CCMR1 寄存器 OC1PE 位以使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器中的 ARPE 位（在向上计数或中心对称模式中）使能自动重载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OC1 的极性可以通过软件在 TIMx_CCER 寄存器中的 CC1P 位设置，它可以设置为高电平有效或者低电平有效。TIMx_CCER 寄存器中的 CC1E 位控制 OC1 输出使能。

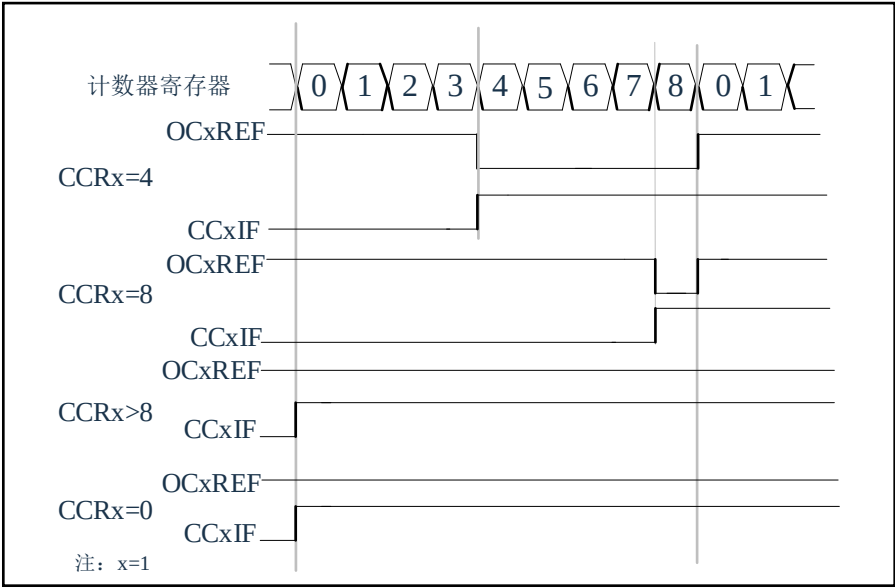
在 PWM 模式（模式 1 或者模式 2），TIMx_CNT 和 TIMx_CCR1 始终在进行比较，以确定是否符合 $TIMx\_CNT \leq TIMx\_CCR1$ 。

定时器仅当计数器是向上计数时才能够产生边沿对齐模式的 PWM。

9.2.9.1 PWM 边沿对齐模式

下面是一个 PWM 模式 1 的例子。当 $TIMx\_CNT < TIMx\_CCR1$ 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCR1 中的比较值大于自动重载值(TIMx_ARR)，则 OC1REF 保持为‘1’。如果比较值为 0，则 OC1REF 保持为‘0’。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

图 9-16 边沿对齐的 PWM 波形（ARR=8）



9.2.10 使用刹车功能

当使用刹车功能时，依据额外的控制位（TIMx_BDTR 寄存器中的 MOE,OSSI 和 OSSR，TIMx_CR2 寄存器中的 OIS1 和 OIS1N），输出使能信号和无效电平信号都会被修改。

刹车源（BRK）信号通道可以是外部输入信号（BKIN）或者如下内部信号：

- 内核 LOCKUP 输出
- PVD 输出
- 来自比较器的输出

内部刹车信号框图如下。



系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

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

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。

- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OIS1 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志(TIMx_SR 寄存器中的 BIF 位)为'1'时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置'1'；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时(自动地或者通过软件)设置 MOE。同时，状态标志 BIF 不能被清除。

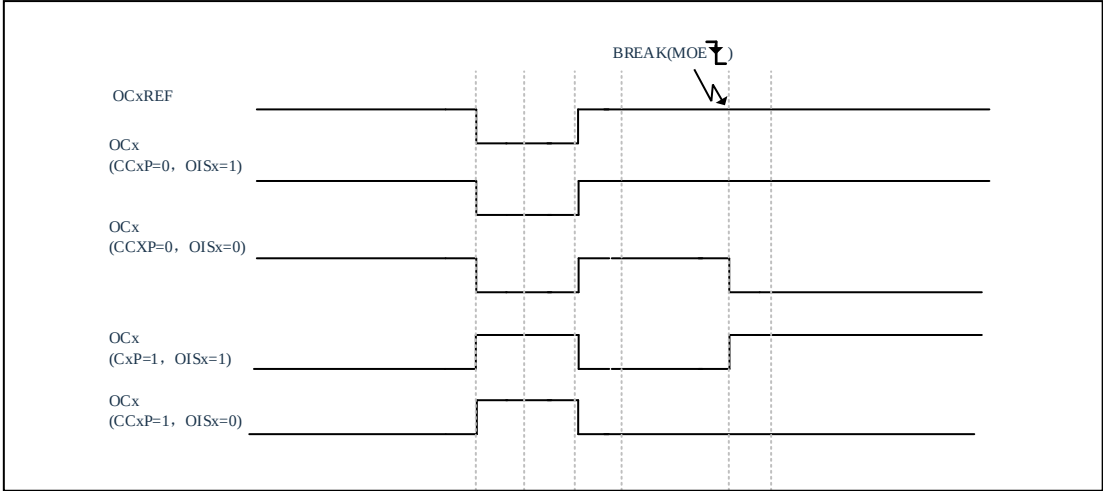
刹车可以由 BRK 输入产生，它的有效极性是可编程的，且由 TIMx_BDTR 寄存器中的 BKE 位开启。这里有两种方式产生刹车：

- 通过可编程极性的 BKR 输入，同时在 TIMx_BDTR 寄存器中使能 BKE 。
- 通过软件设置 TIMx_EGR 中的 BG 位。

除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(OC1 极性和被禁止的状态，OC1M 配置，刹车使能和极性)。用户可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参见 TIMx 刹车和死区寄存器(TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

图 9-21 响应刹车的输出



9.2.11 单脉冲模式

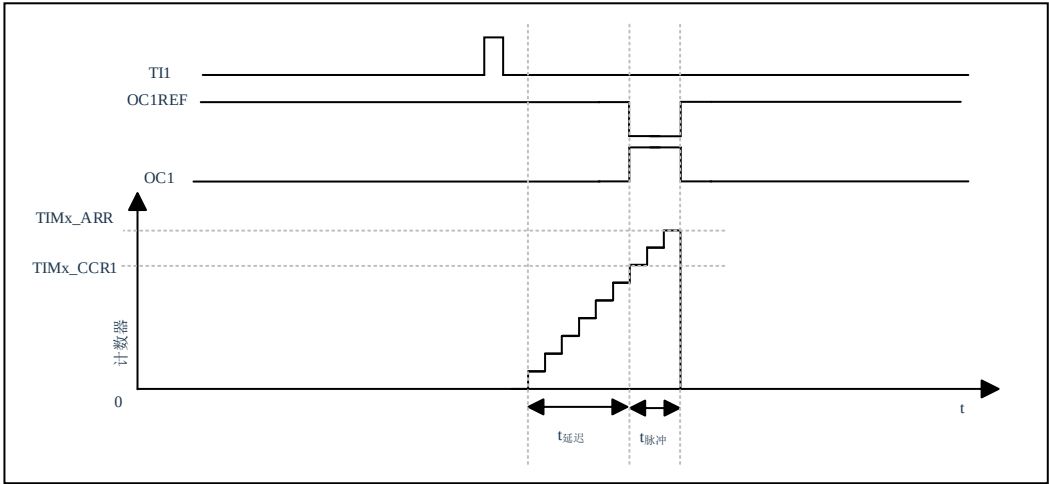
单脉冲模式（OPM）是之前所述众多模式中的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后，产生一个脉宽可被程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器的 OPM 位将选择单脉冲模式，这样可以使计数器自动的在产生下一个更新事件 UEV 时停止。

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

- 在向上计数模式中：计数器 $CNT < CCR1 \leq ARR$ (特别地, $0 < CCR1$)
- 在向下计数模式中： $CNT > CCR1$

图 9-22 单脉冲模式的例子



特殊情况：OC1 快速使能：

在单脉冲模式下，在 TI1 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。如果要以最小延时输出波形，可以设置 TIMx_CCMR1 寄存器中的 OC1FE 位；此时 OC1REF(和 OC1)直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OC1FE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

9.3 寄存器映射

(TIM16 基地址 = 0x4007_4400) RO: 只读, WO: 只写, R/W: 读写

(TIM17 基地址 = 0x4007_4800) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
CR1	0x00	R/W	TIMx 控制寄存器 1	0x00
CR2	0x04	R/W	TIMx 控制寄存器 2	0x00
DIER	0x0C	R/W	TIMxDMA/中断使能寄存器	0x00
SR	0x10	R/W	TIMx 状态寄存器	0x00
EGR	0x14	R/W	TIMx 事件生成寄存器	0x00
CCMR1	0x18	R/W	TIMx 捕获/比较模式寄存器 1	0x00
CCER	0x20	R/W	TIMx 捕获/比较使能寄存器	0x00
CNT	0x24	R/W	TIMx 计数器	0x00
PSC	0x28	R/W	TIMx 预分频器	0x00
ARR	0x2C	R/W	TIMx 自动重载寄存器	0xFFFF
RCR	0x30	R/W	TIMx 重复计数器寄存器	0x00
CCR1	0x34	R/W	TIMx 捕获/比较寄存器 1	0x00
DCR	0x48	R/W	TIMx DMA 控制寄存器	0x00
DMAR	0x4C	R/W	TIMx 全传输 DMA 地址	0x00

9.4 寄存器说明

9.4.1 TIM16/17 控制寄存器 1 (TIMx_CR1)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9:8	CKD[1:0]	时钟分频 (Clock division): 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (ETR、TI1) 所使用的死区及采样时钟 (tDTS) 之间的分频比 00: tDTS = tCK_INT 01: tDTS = 2 * tCK_INT 10: tDTS = 4 * tCK_INT 11: 保留，不要设置成此值	0x0
7	ARPE	自动重载预装载使能: 0: TIMx_ARR 寄存器不进行缓冲 1: TIMx_ARR 寄存器进行缓冲	0
6:4	--	保留，须为 0	0x0
3	OPM	单脉冲模式: 0: 计数器在发生更新事件时不会停止计数 1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止	0
2	URS	更新请求源: 此位由软件置 1 和清零，用以选择 UEV 事件源 0: 如果允许产生更新中断或 DMA 请求，则下述任一事件产生一个更新中断或 DMA 请求 — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件 1: 如果允许产生更新中断或 DMA 请求，如果允许产生更新中断或 DMA 请求，则只有计数器上溢/下溢产生一个更新中断或 DMA 请求	0
1	UDIS	更新禁止: 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0: 允许 UEV。更新 (UEV) 事件可通过以下事件之一生成 — 计数器上溢/下溢 — 将 UG 位置 1 — 从模式控制器产生的更新 被缓存的寄存器被装入它们的预装载值 1: 禁止 UEV。不产生更新事件，影子寄存器 (ARR,PSC,CCR1)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。	0
0	CEN 计数器使能	计数器使能: 0: 禁止计数器 1: 开启计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。	0

9.4.2 TIM16/17 控制寄存器 2 (TIM16/17_CR2)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9	--	保留	0
8	OIS1	输出空闲状态 1（OC1 输出）： 0: 当 MOE=0 时，OC1=0 1: 当 MOE=0 时，OC1=1 注：已经设置了 LOCK(TIMx_BDTR 寄存器)级别 1、2 或 3 后，该位不能被修改。	0
7:4	--	保留，须为 0	0x0
3	CCDS	捕获/比较 DMA 选择： 0: 发生 CC1 事件时,发送 CC1 DMA 请求 1: 发生更新事件时,发送 CC1 DMA 请求	0
2:1	--	保留，须为 0	0x0
0	CCPC	捕获/比较预装载控制： 0: CC1E 和 OC1M 位不是预装载的。 1: CC1E 和 OC1M 位是预装载的；设置该位后，它们只在设置了 COM 位后被更新。	0

9.4.3 TIM16/17 DMA/中断使能寄存器(TIM16/17_DIER)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9	CC1DE	捕获/比较 1 DMA 请求使能： 0: 禁止 CC1 DMA 请求 1: 使能 CC1 DMA 请求	0
8	UDE	更新 DMA 请求使能： 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求	0
7	BIE	刹车中断使能： 0: 禁止刹车中断 1: 使能刹车中断	0
6	--	保留，须为 0	0
5	COMIE	COM 中断使能： 0: 禁止 COM 中断 1: 使能 COM 中断	0
4:2	--	保留，须为 0	0x0
1	CC1IE	捕获/比较 1 中断使能： 0: 禁止 CC1 中断 1: 使能 CC1 中断	0
0	UIE	更新中断使能： 0: 禁止更新中断 1: 使能更新中断	0

9.4.4 TIM16/17 状态寄存器 (TIM16/17_SR)

位	符号	描述	复位值
31:10	--	保留，须为 0	0x0
9	CC1OF	捕获/比较 1 过捕获标志（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“0”可将该位清零）： 0: 未检测到过捕获。 1: TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1	0
8	--	保留，须为 0	0
7	BIF	刹车中断标志：（只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。） 0: 未发生刹车事件 1: 在刹车输入上检测到有效电平。	0
6	--	保留，须为 0	0
5	COMIF	COM 中断标志： 此标志在发生 COM 事件时（捕获/比较控制位 CC1E、CC1NE 和 OC1M 已更新时）由硬件置 1。但需要通过软件清零。 0: 未发生 COM 事件 1: COM 中断等待响应	0
4:2	--	保留，须为 0	0x0
1	CC1IF	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，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。 0: 不匹配。 1: TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。 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（IC1 上已检测到与所选极性匹配的边沿）	0
0	UIF	更新中断标志： 该位在发生更新事件时通过硬件置 1。但需要通过软件清零 0: 未发生更新 1: 更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1 —TIMx_CR1 寄存器中的 UDIS=0，产生更新事件(计数器上溢)。 —若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当 TIMx_EGR 寄存器的 UG=1 时产生更新事件(软件对 CNT 重新初始化)	0

9.4.5 TIM16/17 事件产生寄存器(TIM16/17_EGR)

位	符号	描述	复位值
31:8	--	保留，须为 0	0x0
7	BG	产生刹车事件： 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: 生成刹车事件。MOE 位清零且 BIF 标志置 1。若开启对应的中断和 DMA，则产生相应的中断和 DMA。	0
6	--	保留，须为 0	0
5	COMG	捕获/比较控制更新生成： 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: CCPC 位置 1 时，可更新 CC1E 和 OC1M 位	0
4:2	--	保留，须为 0	0x0
1	CC1G	CC1G: 捕获/比较 1 生成： 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： 若开启对应的中断和 DMA，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果通道 CC1 配置为输入： TIMx_CCR1 寄存器中将捕获到计数器当前值。若开启对应的中断和 DMA，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果 CC1IF 标志已为高电平，CC1OF 标志将置 1	0
0	UG	更新生成： 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一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。	0

9.4.6 TIM16/17 捕获/比较模式寄存器 1(TIM16/17_CCMR1)

输出比较模式

位	符号	描述	复位值
31:7	--	保留，须为 0	0x0
6:4	OC1M	这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P 位。 000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用 001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1(TIMx_CCR1) 相同时，强制 OC1REF 为高 010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1(TIMx_CCR1)相同时，强制 OC1REF 为低 011: 翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平 100: 强制为无效电平。强制 OC1REF 为低 101: 强制为有效电平。强制 OC1REF 为高 110: PWM 模式 1 ——一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平 (OC1REF=0)，否则为无效电平 (OC1REF=1)。 111: PWM 模式 2 ——一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式 1 中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变	0x0
3	OC1PE	输出比较 1 预装载使能: 0: 禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，且新值马上起作用 1: 开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被载入当前寄存器中 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定	0
2	OC1FE	输出比较 1 快速使能: 此位用于加快触发输入事件对 CC 输出的影响 0: 根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OC1FE 的只在通道被配置成 PWM1 或 PWM2 模式时起作用	0
1:0	CC1S	捕获/比较 1 选择: 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC1 通道配置为输出 01: CC1 通道配置为输入，IC1 映射到 TI1 上 10: 保留 11: 保留 注意: 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据	0x0

输入捕获模式。

位	符号	描述	复位值
31:7	--	保留，须为 0	0x0
7:4	IC1F[3:0]	输入捕获 1 滤波器： 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿 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: $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$ 1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$ 1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$ 1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$ 1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$ 1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$ 1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$ 1111: $f_{SAMPLING}=f_{DTS}/32$, $N=8$	0x0
3:2	IC1PSC	输入捕获 1 预分频器： 此位域定义 CC1 输入 (IC1) 的预分频比。 只要 CC1E=“0” (TIMx_CCER 寄存器)，预分频器便立即复位 00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获。 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获。 11: 每发生 8 个事件便执行一次捕获。	0x0
1:0	CC1S	捕获/比较 1 选择： 00: CC1 通道配置为输出 01: CC1 通道配置为输入，IC1 映射到 TI1 上 10: 保留 11: 保留 注意：仅当通道关闭时 (TIMx_CCER 中的 CC1E=“0”)，才可向 CC1S 位写入数据	0x0

表 9-1 刹车和输出控制位对于 OCx 通道的影响

控制位					输出状态
MOE	-	OSSR	CCxE	-	OCx 输出状态
1	-	0	0	-	输出未使能（未被 TIMER 驱动）OCx=0，OCx_EN=0
		0	1	-	OCxPEF+Polarity OCx=OCxPEF 异或 OCxP OCx_EN=1
		1	0	-	-
		1	1	-	-
0	-	X	0	-	-
	-		1	-	-

注：x=1。

9.4.7 TIM16/17 捕获/比较使能寄存器 (TIM16/17_CCER)

位	符号	描述	复位值
31:4	--	保留,须为 0	0x0
3	CC1NP	捕获/比较 1 互补输出极性: CC1 通道配置为输出: C1NP 必须保持 0 CC1 通道配置为输入: 此位与 CC1P 配合使用,用以定义 TI1FP1 极性。请参见 CC1P 说明。 注意: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 2 或 3 且 CC1S= “00” (通道配置为输出), 此位立即变为不可写状态。	0
2	-	保留,须为 0	0
1	CC1P	捕获/比较 1 输出极性: CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 的有效极性 00: 非反相/上升沿触发 电路对 TI1FP1 上升沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TI1FP1 未反相 (在门控模式或编码器模式下执行触发操作) 01: 反相/下降沿触发 电路对 TI1FP1 下降沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TI1FP1 反相 (在门控模式或编码器模式下执行触发操作) 10: 保留, 不使用此配置 11: 未反相/边沿触发 电路对 TI1FP1 上升沿和下降沿都敏感 (在复位模式、外部时钟模式或触发模式下执行捕获 或触发操作), TI1FP1 未反相 (在门控模式下执行触发操作)。编码器模式下不得使用此配置。 注意: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 2 或 3, 此位立即变为不可写状态	0
0	CC1E	捕获 / 比较 1 输出使能: CC1 通道配置为输出: 0: 关闭 —OC1 禁止输出, 1: 开启 —OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE, OSSI, OSSR, OIS1 位的值。 CC1 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 1 (TIMx_CCR1) 中。 0: 禁止捕获。 1: 使能捕获。	0

9.4.8 TIM16/17 计数器(TIM16/17_CNT)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CNT[15:0]	计数器值 (Counter value)	0x0

9.4.9 TIM16/17 预分频器(TIM16/17_PSC)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	PSC[15:0]	预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。	0x0

9.4.10 TIM16/17 自动重载寄存器 (TIM16/17_ARR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	ARR[15:0]	自动重载值 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。	0xFFFF

9.4.11 TIM16/17 周期计数寄存器(TIM16/17_RCR)

位	符号	描述	复位值
31:8	--	保留，必须为 0	0x0
7:0	REP[7:0]	REP[7:0]：重复计数器值 使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。 这意味着 PWM 模式下 (REP+1) 相当于： - 边沿对齐模式下的 PWM 周期数 - 中心对齐模式下的 PWM 半周期数	0x0

9.4.12 TIM16/17 捕获/比较寄存器 1(TIM16/17_CCR1)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	CCR1[15:0]	捕获/比较 1 值 如果通道 CC1 配置为输出： CCR1 是捕获/比较寄存器 1 的预装载值。 如果没有通过 TIMx_CCMR 1 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。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 1）。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 OC1 输出上发出信号的值。 如果通道 CC1 配置为输入： CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值	0x0

9.4.13 TIM16/17 刹车和死区寄存器 (TIM16/17_BDTR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15	MOE	主输出使能： 一旦刹车输入有效，该位被硬件异步清 0。根据 AOE 位的值，可由软件清 0 或自动置 1。它仅对配置为输出通道有效。 0: OC 输出禁止或被强制为空闲状态。 1: 如果 OC 相应输出使能位（TIMx_CCER 寄存器中的 CC1E 位）均置 1，使能 OC 输出。	0
14	AOE	自动输出使能 (Automatic output enable): 0: MOE 只能由软件置 1 MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果刹车输入无效) 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改	0
13	BKP	刹车极性 (Break polarity): 0: 刹车输入为低电平有效 1: 刹车输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。	0
12	BKE	刹车路使能 (Break enable): 0: 禁止刹车输入（BRK 及 BRK_ACTH） 1: 使能刹车输入（BRK 及 BRK_ACTH） 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。	0
11	OSSR	运行模式下的关闭状态选择： 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC 使能的详细说明（捕获/比较使能寄存器(TIMx_CCER)） 0: 当定时器不工作时，禁止 OC 输出（OC 使能输出信号=0）。 1: 当定时器不工作时，一旦 CCxE=1，开启 OC 输出并输出无效电平。OC 使能输出信号=1 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。	0
10	OSSI	空闲模式下的关闭状态选择： 该位用于当 MOE=0 且通道设为输出时。 参考 OC 使能的详细说明（捕获/比较使能寄存器	0

		(TIMx_CCER)) 0: 当定时器不工作时, 禁止 O 输出 (OC 使能输出信号=0) 1: 当定时器不工作时, 一旦 CCxE=1, OC 首先输出其空闲电平。OC 使能输出信号=1 注意: 编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 2 后, 此位即无法修改。	
9:8	LOCK	锁定配置 (Lock configuration): 这些位用于针对软件错误提供写保护。 00: 锁定关闭, 寄存器无写保护。 01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 BKE/BKP/AOE 位、TIMx_CR2 寄存器的 OIS1 位。 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位 (一旦相关通道通过 CC1S 位设为输出, TIMx_CCER 寄存器的 CCx1P/CCN1P 位) 以及 OSSR/OSSI 位 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位 (一旦相关通道通过 CC1S 位设为输出, TIMx_CCMR1 寄存器的 OC1M/OC1PE 位)。 注意: 注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIMx_BDTR 寄存器, 则其内容冻结直至复位。。	0x0
7:0	-	保留	0x0

9.4.14 TIM16/17 DMA 控制寄存器(TIM16/17_DCR)

位	符号	描述	复位值
31:13	-	保留	0x0
12:8	DBL	DMA 连续传送长度 (DMA burst length) 该 5 位定义了 DMA 在连续模式下的传送长度（当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 00011: 4 次传输 10001: 18 次传输	0x0
7:5	-	保留	0x0
4:0	DBA	DMA 基址 (DMA base address) 该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。 示例： 00000: TIMx_CR1， 00001: TIMx_CR2， 00010: 保留 ...	0x0

9.4.15 连续模式的 DMA 地址 (TIM16/17_DMAR)

位	符号	描述	复位值
31:16	--	保留，须为 0	0x0
15:0	DMAB	DMA 连续传送寄存器 (DMA register for burst accesses) 对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器 (TIMx_CR1 地址) + (DBA + DMA 索引) x 4, 其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制， 其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。	0x0

第10章 窗口看门狗定时器 WWDG

10.1 窗口看门狗定时器简介

窗口看门狗通常被用来监测，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。除非递减计数器的值在 T6 位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生 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。

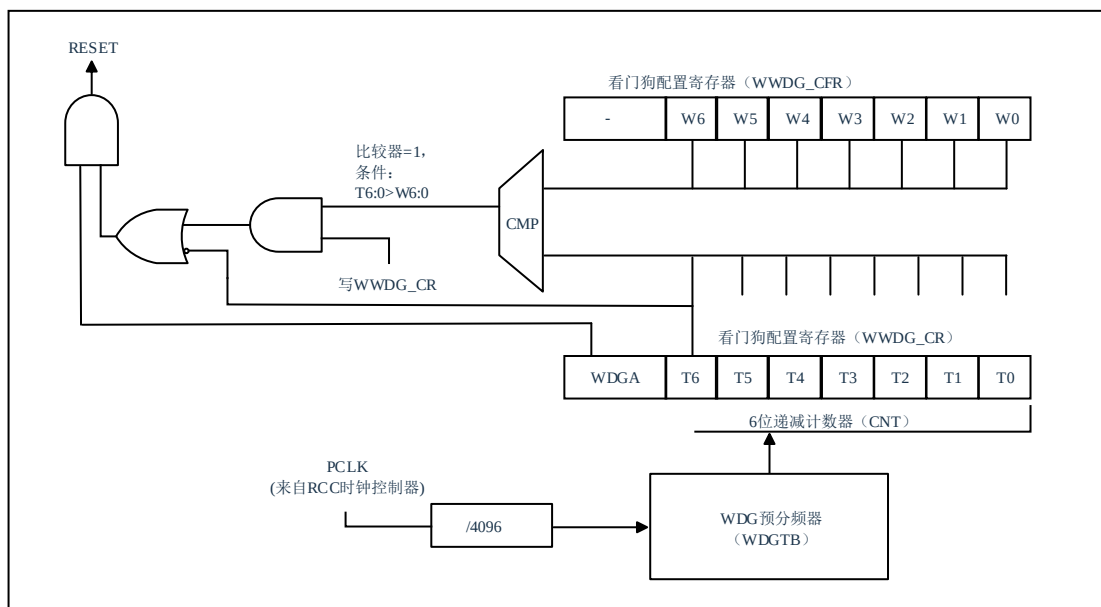
10.2 窗口看门狗定时器主要特性

- 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 (EWD)：当递减计数器等于 0x40 时触发（如果已使能且看门狗已激活）

10.3 窗口看门狗定时器功能说明

如果激活看门狗（WWDG_CR 寄存器中的 WDGA 位置 1），则当 7 位递减计数器（T[6:0] 位）从 0x40 滚动到 0x3F（T6 已清零）时会引发复位。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。

图 10-1 看门狗框图



应用程序在正常运行过程中必须定期地写入 WWDG_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时，才能执行此操作。存储在 WWDG_CR 寄存器中的值必须介于 0xFF 和 0xC0 之间。

10.3.1.1 使能看门狗

在系统复位后，看门狗总是处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

10.3.1.2 控制递减计数器

递减计数器处于自由运行状态：即使禁止看门狗，递减计数器仍继续递减计数。当使能看门狗时，必须将 T6 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，预分频值是未知的。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。介绍了窗口看门狗的工作过程。

注意：可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。

10.3.1.3 看门狗中断高级特性

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 EWI 位使能 EWI 中断。当递减计数器的值为 0x40 时，将生成 EWI 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 EWI 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。

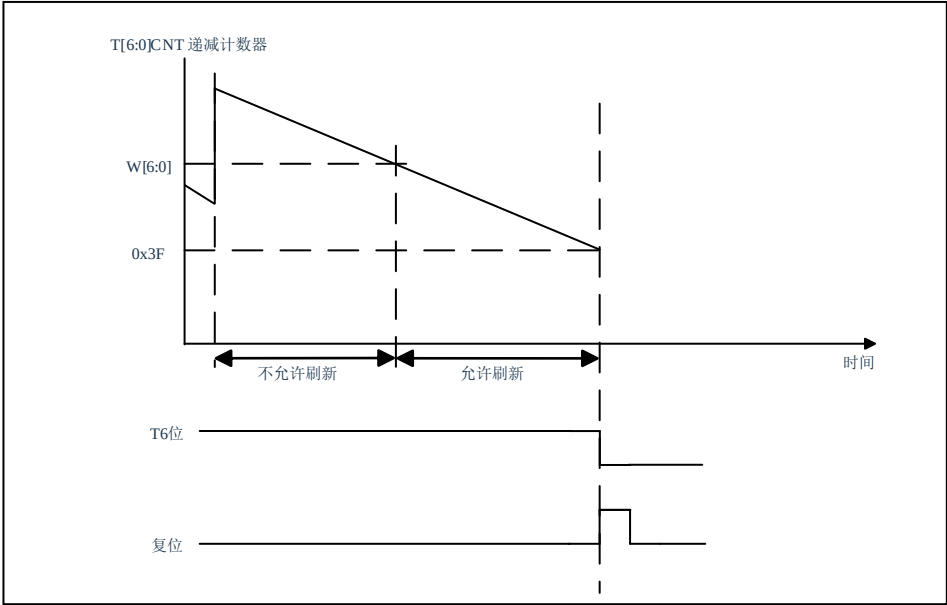
注意：当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。

10.4 设置看门狗超时

可以使用图 10-2 中的公式来计算 WWDG 超时

注意：写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。

图 10-2 窗口看门狗时序图



超时值的计算公式如下：

$$t_{\text{WWDG}} = t_{\text{PCLK}} * 4096 * 2^{\text{WDGTB}} * (t[5:0] + 1) \text{ (ms)}$$

其中：

t_{WWDG} ：WWDG 超时

t_{PCLK} ：APB 时钟周期，以 ms 为测量单位

有关 T_{WWDG} 的最小值和最大值，请参见下表。

表 10-1 24MHz（F_{PCLK}）时的超时值

预分频器	WDGTB	最小超时（us） T[5:0] = 0x00	最大超时（ms） T[5:0] = 0x3F
1	0	170.67	10.92
2	1	341.34	21.85
4	2	682.67	43.69
8	3	1365.34	87.39

10.5 寄存器映射

(WWDG 基地址 = 0x4006_2C00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
WWDG_CR	0x000	R/W	WWDG 控制寄存器	0x7f
WWDG_CFR	0x004	R/W	WWDG 配置寄存器	0x7f
WWDG_SR	0x008	R/W	WWDG 状态寄存器	0x0

10.6 寄存器说明

10.6.1 WWDG 定时器控制寄存器 (WWDG_CR)

位	符号	描述	复位值
31:8	-	保留	0x0
7	WDGA	激活位 (Activation bit) 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 WDGA = 1 时，看门狗可产生复位。	0
6:0	T[6:0]	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 $(4096 \times 2^{WDG_{TB}})$ PCLK 个周期递减一次。当它从 0x40 滚动到 0x3F (T6 清零) 时会产生复位	0x7f

10.6.2 WWDG 配置寄存器（WWDG_CFR）

位	符号	描述	复位值
31:10	-	保留	0x0
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。此中断只有在复位后才由硬件清零。	0
8:7	WDGTB	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	0
6:0	W	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。	0x7f

10.6.3 WWDG 状态寄存器(WWDG_SR)

位	符号	描述	复位值
31:1	-	保留	0x0
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 0 来清零。写入 1 不起作用。如果不使能中断，此位也会被置 1。	0

第11章 独立看门狗定时器 IWDG

11.1 概述

独立看门狗(IWDG)由其专用低速时钟(LSI)驱动，因此即便在主时钟发生故障时仍然保持工作状态。

当系统产生 LOCKUP 时，IWDG 计数使能强制开启，LOCKUP 撤销后，关闭 IWDG 计数使能。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低场合。

11.2 特性

- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在睡眠模式和深度睡眠模式下运行）
- 当递减计数器值达到 0x000 时产生复位（如果看门狗已激活）

11.3 功能说明

图 11-1 给出了独立看门狗模块的功能框图

当通过对关键字寄存器(IWDG_KR)写入值 0xCCCC 启动独立看门狗时，计数器开始从复位值 0xFFFF 递减计数。当计数器计数到终值(0x000)时会产生一个复位信号（IWDG 复位）。任何时候将关键字 0xAAAA 写到 IWDG_KR 寄存器中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

11.3.1 硬件看门狗

如果通过器件选项字节 0 使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向关键字寄存器写入相应的值，则系统会产生复位。

11.3.2 寄存器访问保护

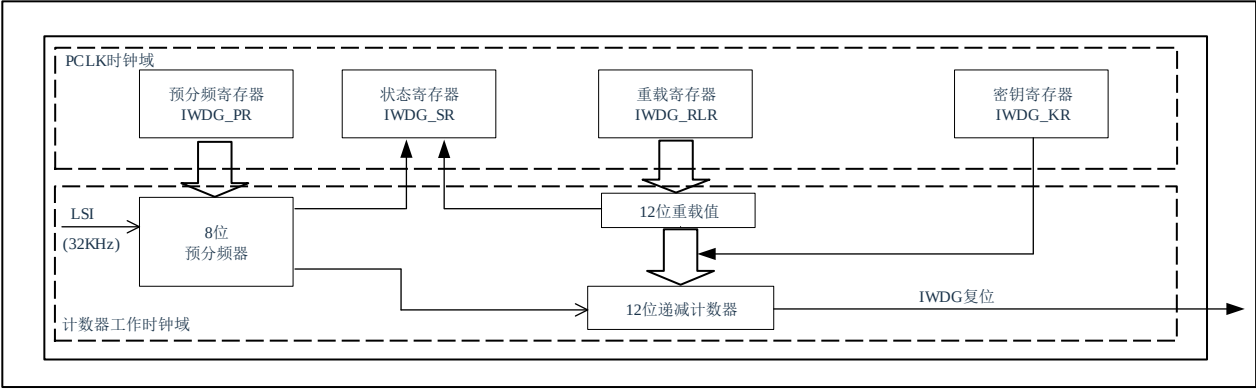
IWDG_PR 和 IWDG_RLR 寄存器具有写访问保护。若要修改寄存器，必须首先对 IWDG_KR 寄存器写入代码 0x5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这意味着重装载操作（即写入 0xAAAA）也会启动写保护功能。

状态寄存器指示预分频值和递减计数器是否正在被更新。

11.3.3 调试模式

当微控制器进入调试模式时（内核停止），IWDG 计数器会根据 1.4 章节配置位选择继续正常工作或者停止工作。

图 11-1 独立看门狗框图



注意：看门狗功能由 VDD 电压域供电，在停止模式和待机模式下仍能工作。

表 11-1 32KHz(LSI)频率下 IWDG 超时周期的最小值/最大值

预分频器	PR[2:0]位	最小超时 (ms) RL[11:0] = 0x000	最大超时 (ms) RL[11:0] = 0xFFFF
/4	0	0.125	512
/8	1	0.25	1024
/16	2	0.5	2048
/32	3	1	4096
/64	4	2	8192
/128	5	4	16384
/256	6	8	32768

注意：这些时间均针对 32KHz 时钟给出。实际上，MCU 内部的 RC 频率会在 20KHz 到 40KHz 之间变化。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的。

11.4 IWDG 寄存器映射

(IWDG 基地址 = 0x4006_3000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
IWDG_KR	0x000	R/W	IWDG 关键字寄存器	0x0
IWDG_PR	0x004	R/W	IWDG 预分频寄存器	0x0
IWDG_RLR	0x008	R/W	IWDG 重载寄存器	0xff
IWDG_SR	0x00C	R/W	IWDG 状态寄存器	0x0

11.5 IWDG 寄存器说明

外设寄存器可支持半字（16 位）或字（32 位）访问。

11.5.1 关键字寄存器(IWDG_KR)

位	符号	描述	复位值
31:16	-	保留	0x0
15:0	KEY	键值(Key value)（只写位，读为 0000h）： 必须每隔一段时间便通过软件对这些位写入键值 AAAAh，否则当计数器计数到 0 时，看门狗会产生复位。 0x5555: 使能对 IWDG_PR 和 IWDG_RLR 寄存器的访问 0xCCCC: 启动看门狗（选中硬件看门狗选项的情况除外）	0

11.5.2 预分频器寄存器(IWDG_PR)

位	符号	描述	复位值
31:3	-	保留	0x0
2:0	PR	预分频器(Prescaler divider): 000: 4 分频 001: 8 分频 010: 16 分频 011: 32 分频 100: 64 分频 101: 128 分频 110: 256 分频 111: 256 分频 这些位受写访问保护，请参见第 12.3.2 节。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，IWDG_SR 的 PVU 位必须为 0。	0x0

注意：读取该寄存器会返回 VDD 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG_SR 寄存器中的 PVU 位为 0 时，从寄存器读取的值才有效。

11.5.3 重载寄存器(IWDG_RLR)

位	符号	描述	复位值
31:12	-	保留	0x0
11:0	RL	看门狗计数器重载值(Watchdog counter reload value) 这些位受写访问保护。这个值由软件设置，每次对 IWDR_KR 寄存器写入值 AAAAh 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。 若要更改重载值，IWDG_SR 中的 RVU 位必须为 0。	0xffff

注意：读取该寄存器会返回 VDD 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG_SR 寄存器中的 RVU 位为 0 时，从寄存器读取的值才有效。

11.5.4 状态寄存器(IWDG_SR)

位	符号	描述	复位值
31:2	-	保留	0x0
1	RVU	看门狗计数器重载值更新 可通过硬件将该位置 1 以指示重载值正在更新。当在 VDD 电压域下完成重载值更新操作后（需要多达 5 个 RC 32KHz 周期），会通过硬件将该位复位。 重载值只有在 RVU 位为 0 时才可更新。	0
0	PVU	看门狗预分频器值更新 可通过硬件将该位置 1 以指示预分频器值正在更新。当在 VDD 电压域下完成预分频器值更新操作后（需要多达 5 个 RC 32KHz 周期），会通过硬件将该位复位。预分频器值只有在 PVU 位为 0 时才可更新。	0

注意：如果应用使用多个重载值或预分频器值，则必须等到 RVU 位被清零后才能更改重载值，而且必须等到 PVU 位被清零后才能更改预分频器值。但是，在更新预分频器和/或重载值之后，则无需等到 RVU 或 PVU 复位后再继续执行代码（即便进入低功耗模式，也会继续执行写操作至完成）

第12章 通用串行通信单元

本产品搭载1个通用串行通信单元，单元0有2个串行通道，各通道能实现3线串行(SSPI)、UART 和简易 I²C 的通信功能，其通道功能分配如下：

表12-1：通用串行通道单元通道功能分配

单元	通道	用作 SSPI	用作 UART	用作简易 I ² C
0	0	SSPI00	UART0	IIC00
	1	-		-

注意：

1. “-”表示在此型号产品中不支持；
2. 在单元 0 的通道 0 和通道 1 使用 UART0 时，不能使用 SSPI00 和 IIC00；

12.1 通用串行通信单元的功能

本产品支持的各串行接口的特征如下所示：

12.1.1 3 线串行 I/O（SSPI00）

与主控设备输出的串行时钟（SCLK）同步进行数据的发送和接收。

这是使用1条串行时钟（SCLK）、1条发送串行数据（SDO）和1条接收串行数据（SDI）共3条通信线进行通信的时钟同步通信功能。

有关具体的设定例子，请参照“12.5 3线串行 I/O（SSPI00）通信的运行”。

[数据的发送和接收]

- 7位、8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率

主控通信：Max.6MHz

从属通信：Max.6MHz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注意：必须在满足 SCLK 周期时间（ T_{KCY} ）特性的范围内使用。详细内容请参照数据手册。

12.1.2 带从属片选功能的简易 SPI

支持从属片选输入功能的 SPI 串行通信接口。这是使用一个从属片选输入（SS）、1条串行时钟（SCLK）、1条发送串行数据（SDO）和1条接收串行数据（SDI）共4条通信线进行通信的时钟同步通信接口。

[数据的发送和接收]

- 7位、8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

- 输入/输出时钟的相位控制
- 由预分频器和通道内部计数器产生的传送周期
- 最大传送速率

从属通信：6MHz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

12.1.3 UART0

这是通过串行数据发送（TxD）和串行数据接收（RxD）共2条线进行异步通信的功能。使用这2条通信线，按数据帧（由起始位、数据、奇偶校验位和停止位构成）与其他通信方进行异步（使用内部波特率）的数据发送和接收。能通过使用发送专用（偶数通道）和接收专用（奇数通道）共2个通道来实现全双工 UART 通信。

有关具体的设定例子，请参照“12.7 UART0通信的运行”。

[数据的发送和接收]

- 7位、8位的数据长度
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定、反相的选择
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误或者溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

12.1.4 简易 I²C (IIC00)

这是通过串行时钟 (SCL) 和串行数据 (SDA) 共2条线与多个设备进行时钟同步通信的功能。因为此简易 I²C 是为了与闪存、A/D 转换器等设备进行单通信而设计的, 所以只用作主控设备。

对于开始条件和停止条件, 必须遵守 AC 规格, 在操作控制寄存器的同时通过软件进行处理。有关具体的设定例子, 请参照“12.8 简易 I²C (IIC00) 通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收 (仅限单主控的主控功能)
- ACK 输出功能注、ACK 检测功能
- 8 位数据长 (在发送地址时, 用高 7 位指地址, 用最低位进行 R/W 控制)
- 通过软件产生开始条件和停止条件

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误、溢出错误

[简易 I²C 不支持的功能]

- 从属发送、从属接收
- 仲裁失败检测功能
- 等待检测功能

注: 在接收最后的数据时, 如果给 SOEmn 位 (串行输出允许寄存器 m (SOEm)) 写“0”来停止串行通信数据的输出, 就不输出 ACK。详细内容请参照“12.8.3(2) 处理流程”。

备注: 在使用全功能的 I²C 总线时, 请参照“第 14 章 I²C 接口”。

12.2 通用串行通信单元的结构

通用串行通信单元由以下硬件构成。

表12-2：通用串行通信单元的结构

项目	结构
移位寄存器	16位
缓冲寄存器	串行数据寄存器mn（SDRmn） ^注
串行时钟输入/输出	SCLKOI00引脚（用于3线串行I/O）、 SCL00引脚（用于简易I2C）
串行数据输入	SDI00引脚（用于3线串行I/O）、 RxD0引脚（用于UART）
串行数据输出	SDO00引脚（用于3线串行I/O）、 TxD0引脚（用于UART）
串行数据输入/输出	SDA00引脚（用于简易I2C）
从属选择输入	SAU0_SS引脚（用于串行通信单元0做SPI从属选择输入功能）
控制寄存器	<单元设定部的寄存器> 外围允许寄存器（PER0） 串行时钟选择寄存器m（SPSm） 串行通道允许状态寄存m（SEm） 串行通道开始寄存器m（SSm） 串行通道停止寄存器m（STm） 串行输出允许寄存器m（SOEm） 串行输出寄存器m（SOM） 串行输出电平寄存器m（SOLm） 输入切换控制寄存器（ISC） 噪声滤波器允许寄存器0（NFEN0）
	<各通道部的寄存器> 串行数据寄存器mn（SDRmn） 串行模式寄存器mn（SMRmn） 串行通信运行设定寄存器mn（SCRmn） 串行状态寄存器mn（SSRmn） 串行标志清除触发寄存器mn（SIRmn）
	端口复用功能配置寄存器（PS_xx） 端口输出模式寄存器（POMxx） 端口模式寄存器（PMxx） 端口寄存器（Pxx）

注：SEmn=1的动作期间；

备注：m：单元号（m=0）；

n：通道号（n=0、1）；

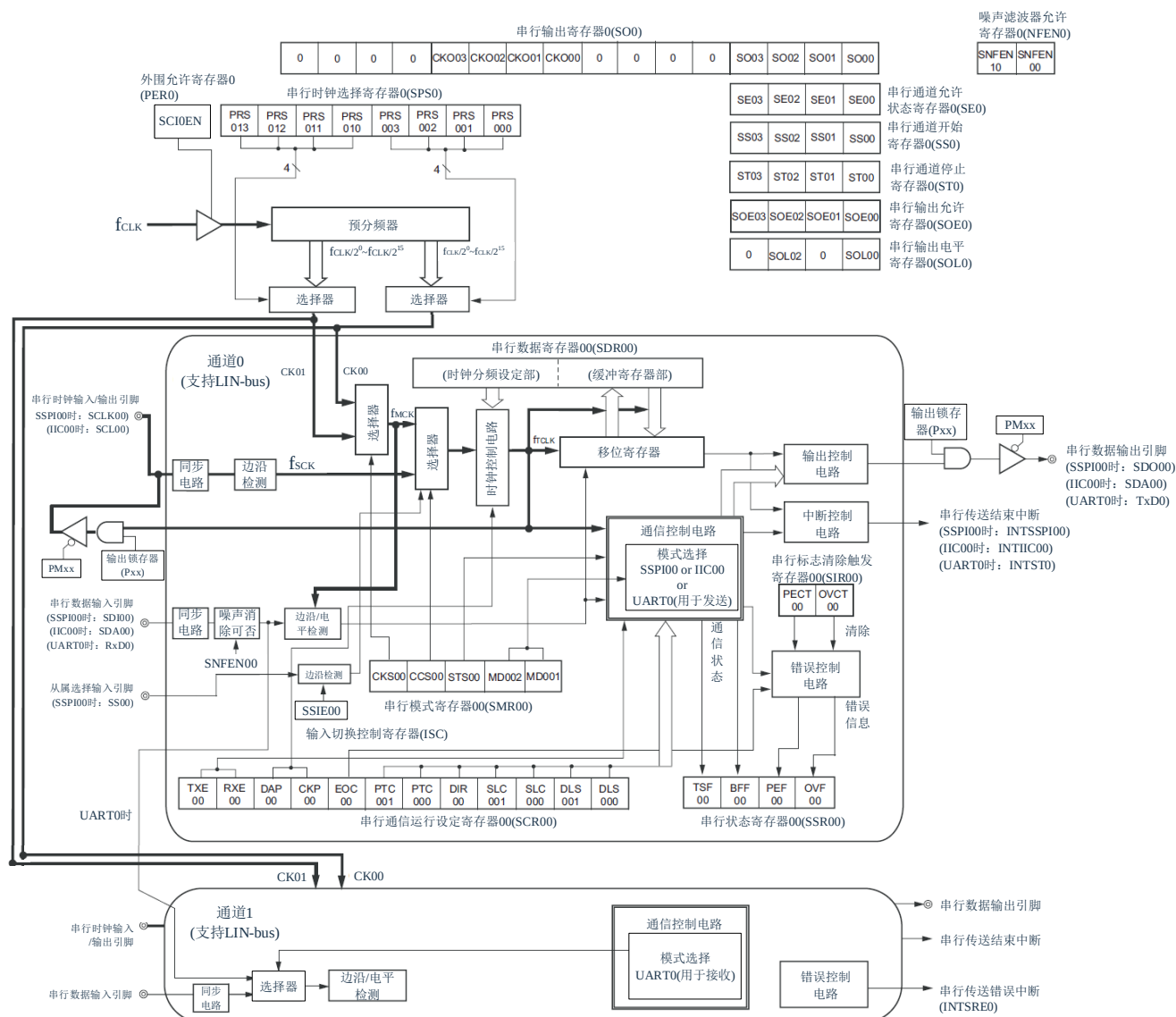
p：SSPI 号（p=00）；

q：UART 号（q=0）；

r：IIC 号（r=00）。

通用串行通信单元的框图如图12-1所示：（以单元0为例）

图12-1: 通用串行通信单元0的框图



12.2.1 移位寄存器

这是进行并行和串行相互转换的16位寄存器。

在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出。不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，在运行期间(SEmn=1)使用串行数据寄存器 mn(SDRmn)。

表12-3: 移位寄存器的格式

位	符号	描述	复位值
15:0	SDRmn	串行数据移位寄存器	0x0

12.2.2 串行数据寄存器 mn (SDRmn)

SDRmn 寄存器是通道 n 的发送和接收数据寄存器（16位）。

当运行停止（SEmn=0）时，bit15~9用作运行时钟（FMCK）的分频设定寄存器。在运行期间（SEmn=1）bit15~9用作发送接收缓冲寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到串行数据寄存器 SDRmn；在发送数据时，将被传送到移位寄存器的发送数据设置到串行数据寄存器 SDRmn。

与数据的输出顺序无关，根据串行通信运行设定寄存器 mn（SCRmn）的 bit1~bit0（DLSmn1~DLSmn0）的设定，保存到 SDRmn 寄存器的数据如下所示：

- 7位数据长度（保存在 SDRmn 寄存器的 bit0~6）；
- 8位数据长度（保存在 SDRmn 寄存器的 bit0~7）；

能以16位为单位读写 SDRmn 寄存器。

SEmn=1时，SDRmn 寄存器的低8位可以作为 SDRmnL 以8位为单位进行读写^注。

根据通信模式，能用以下 SFR 名称，读写 SDRmnL 寄存器。

- SSPIp 通信.....SDIOpL
- UARTq 接收.....RXDq（UARTq 接收数据寄存器）
- UARTq 发送.....TXDq（UARTq 发送数据寄存器）
- IICr 通信.....SDIOr（IICr 数据寄存器）

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

注：在运行停止（SEmn=0）时，禁止通过8位存储器操作指令改写 SDRmn[7:0]（否则，SDRmn[15:9]全部被清“0”）；

备注：m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）；

q：UART 号（q=0）；

r：IIC 号（r=00）。

表12-4：串行数据寄存器 mn（SDRmn）的格式

位	符号	描述	复位值
15:0	SDRmn	串行数据寄存器	0x0

备注：

- 有关 SDRmn 寄存器的高7位的功能，请参照“12.3 控制通用串行通信单元的寄存器”；
- m：单元号（m=0）；
- n：通道号（n=0、1）。

12.3 控制通用串行通信单元的寄存器

控制通用串行通信单元的寄存器如下所示：

1. 外围允许寄存器0 (PER0)
2. 串行时钟选择寄存器 m (SPSm)
3. 串行模式寄存器 mn (SMRmn)
4. 串行通信运行设定寄存器 mn (SCRmn)
5. 串行数据寄存器 mn (SDRmn)
6. 串行标志清除触发寄存器 mn (SDIRmn)
7. 串行状态寄存器 mn (SSRmn)
8. 串行通道开始寄存器 m (SSm)
9. 串行通道停止寄存器 m (STm)
10. 串行通道允许状态寄存器 m (SEm)
11. 串行输出允许寄存器 m (SOEm)
12. 串行输出电平寄存器 m (SOLm)
13. 串行输出寄存器 m (SOM)
14. 输入切换控制寄存器 (ISC)
15. 噪声滤波器允许寄存器0 (NFEN0)
16. 端口复用功能配置寄存器 (PS_xx)
17. 端口输出模式寄存器 (POMx)
18. 端口模式寄存器 (PMx)
19. 端口寄存器 (Px)

备注：m：单元号 (m=0)；

n：通道号 (n=0、1)。

串行通信单元寄存器列表:

寄存器基地址: 0x40041500

表12-5: 寄存器列表

偏移地址	寄存器名称	R/W	复位值
0x000	SSRm0	R	0000H
0x002	SSRm1	R	0000H
0x008	SIRm0	R/W	0000H
0x00A	SIRm1	R/W	0000H
0x010	SMRm0	R/W	0020H
0x012	SMRm1	R/W	0020H
0x018	SCRm0	R/W	0083H
0x01A	SCRm1	R/W	0083H
0x020	SEm	R/O	0000H
0x022	SSm	R/W	0000H
0x024	STm	R/W	0000H
0x026	SPSm	R/W	0000H
0x028	SOM	R/W	0303H
0x02A	SOEm	R/W	0000H
0x034	SOLm	R/W	0000H
0x040	SDRm0	R/W	0000H
0x042	SDRm1	R/W	0000H

备注: 单元号 m=0。

(以下寄存器基地址 = 0x4004_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
NFEN0	0x00	R/W	噪声滤波允许寄存器 0	0x00
ISC	0x03	R/W	输入切换控制寄存器	0x00

12.3.1 外围允许寄存器（PER0）

PER0是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用通用串行通信单元0时，必须将 PER0的 bit3（SAUEN）置“1”。

表12-6: 外围允许寄存器0（PER0）的格式

位	符号	描述	复位值
7	LPTIMEN	提供 15 位间隔定时器输入时钟的控制 0: 停止提供输入时钟，不能写 15 位间隔定时器使用的 SFR。 1: 提供输入时钟，能读写 15 位间隔定时器使用的 SFR。	0
6	-	保留	0
5	ADCEN	提供 ADC 模块的输入时钟的控制 0: 停止提供输入时钟，ADC 不能运行 1: 提供输入时钟，ADC 能运行	0
4	-	保留	0
3	SAUEN	提供串行通信单元输入时钟的控制 0: 停止提供输入时钟，不能写 SAU 使用的 SFR。 1: 提供输入时钟，能读写 SAU 使用的 SFR。	0
2:0	-	保留	0

注意：要设定通用串行通信单元 m 时，必须先在 SAUmEN 位为“1”的状态下设定以下的寄存器。当 SAUmEN 位为“0”时，忽视通用串行通信单元 m 的控制寄存器的写操作，而且读取值都为初始值（输入切换控制寄存器（ISC）、噪声滤波器允许寄存器0（NFEN0）、端口复用功能配置寄存器（PS_xx）、端口输出模式寄存器（POMx）、端口模式寄存器（PMx）、端口模式控制寄存器（PMCx）以及端口寄存器（Px）除外）：

1. 串行时钟选择寄存器 m（SPSm）；
2. 串行模式寄存器 mn（SMRmn）；
3. 串行通信运行设定寄存器 mn（SCRmn）；
4. 串行数据寄存器 mn（SDRmn）；
5. 串行标志清除触发寄存器 mn（SIRmn）；
6. 串行状态寄存器 mn（SSRmn）；
7. 串行通道开始寄存器 m（SSm）；
8. 串行通道停止寄存器 m（STm）；
9. 串行通道允许状态寄存器 m（SEm）；
10. 串行输出允许寄存器 m（SOEm）；
11. 串行输出电平寄存器 m（SOLm）；
12. 串行输出寄存器 m（SOM）。

12.3.2 串行时钟选择寄存器 m (SPSm)

SPSm 寄存器是16位寄存器，选择提供给各通道的2种公共运行时钟（CKm0、CKm1）。通过 SPSm 寄存器的 bit7~4选择 CKm1，通过 bit3~0选择 CKm0。

禁止在运行过程中（SEmn=1）改写 SPSm 寄存器。

通过16位存储器操作指令设定 SPSm 寄存器。

能用 SPSmL 并且通过8位存储器操作指令设定 SPSm 寄存器的低8位。

在产生复位信号后，SPSm 寄存器的值变为“0000H”。

表12-7：串行时钟选择寄存器 m (SPSm) 的格式

位	符号	描述	复位值
15:8	-	保留	0x0
7:4	PRSm13~PRSm10	运行时钟CKm1的选择 0000: FCLK 0001: FCLK/2 0010: FCLK/2 ² 0011: FCLK/2 ³ 0100: FCLK/2 ⁴ 0101: FCLK/2 ⁵ 0110: FCLK/2 ⁶ 0111: FCLK/2 ⁷ 1000: FCLK/2 ⁸ 1001: FCLK/2 ⁹ 1010: FCLK/2 ¹⁰ 1011: FCLK/2 ¹¹ 1100: FCLK/2 ¹² 1101: FCLK/2 ¹³ 1110: FCLK/2 ¹⁴ 1111: FCLK/2 ¹⁵	0x0
3:0	PRSm03~PRSm00	运行时钟CKm0的选择 0000: FCLK 0001: FCLK/2 0010: FCLK/2 ² 0011: FCLK/2 ³ 0100: FCLK/2 ⁴ 0101: FCLK/2 ⁵ 0110: FCLK/2 ⁶ 0111: FCLK/2 ⁷ 1000: FCLK/2 ⁸ 1001: FCLK/2 ⁹ 1010: FCLK/2 ¹⁰ 1011: FCLK/2 ¹¹ 1100: FCLK/2 ¹² 1101: FCLK/2 ¹³ 1110: FCLK/2 ¹⁴ 1111: FCLK/2 ¹⁵	0x0

注：要在通用串行通信单元（SCI）运行过程中更改被选择为 FCLK 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止 SCI 的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改；

注意：必须将 bit15~8置“0”；

备注：

FCLK: CPU/外围硬件的时钟频率；

m: 单元号（m=0）；

k=0、1。

12.3.3 串行模式寄存器 mn (SMRmn)

SMRmn 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟 (F_{MCK}) 的选择、能否使用串行时钟 (F_{SCLK}) 输入的指定、开始触发的设定、运行模式 (SSPI、UART、简易 I²C) 的设定以及中断源的选择。另外，只在 UART 模式中设定接收数据的反相电平。

禁止在运行过程中 (SEmn=1) 改写 SMRmn 寄存器，但是能在运行过程中改写 MDmn0 位。

通过 16 位存储器操作指令设定 SMRmn 寄存器。

在产生复位信号后，SMRmn 寄存器的值变为“0020H”。

表12-8：串行模式寄存器 mn (SMRmn) 的格式

位	符号	描述	复位值
15	CKSmn	通道n运行时钟 (F _{MCK}) 的选择 0: SPSm寄存器设定的运行时钟CKm0 1: SPSm寄存器设定的运行时钟CKm1 运行时钟 (F _{MCK}) 用于边沿检测电路。通过设定CCSmn位和SDRmn寄存器的高7位，生成传送时钟 (F _{TCLK})。	0
14	CCSmn	通道n传送时钟 (F _{TCLK}) 的选择 0: CKSmn位指定的运行时钟F _{MCK} 的分频时钟 1: 来自SCLKp引脚的输入时钟F _{SCLK} (SSPI模式的从属传送) 传送时钟F _{TCLK} 用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当CCSmn位为“0”时，通过SDRmn寄存器的高7位进行运行时钟 (F _{MCK}) 的分频设定。	0
13:10	-	-	-
8	STSmn ^{注1}	开始触发源的选择 0: 只有软件触发有效 (在SSPI、UART发送、简易I ² C时选择)。 1: RxDq引脚的有效边沿 (在UART接收时选择) 在将SSm寄存器置“1”后满足上述条件时，开始传送。	0
7	-	-	-
6	SISmn0 ^{注1}	UART模式中的通道n接收数据的电平反相控制 0: 将下降沿检测为起始位。 不将输入的通信数据进行反相。 1: 将上升沿检测为起始位。 将输入的通信数据进行反相。	0x0
5:3	-	-	0x4
2:1	MDmn2~MDmn1	通道n运行模式的设定 00: SSPI模式 01: UART模式 10: 简易I ² C模式 11: 禁止设定。	0x0
0	MDmn0	通道n中断源的选择 0: 传送结束中断 1: 缓冲器空中断 (在数据从SDRmn寄存器传送到移位寄存器时发生) 在连续发送时，如果MDmn0位为“1”并且SDRmn的数据为空，就写下一个发送数据。	0

注1：只限于 SMR01、SMR03、SMR11、SMR21、SMR31、SMR41 寄存器；

注意：必须将 bit13~9、7、4、3 (SMR00、SMR02、SMR10、SMR20、SMR30、SMR40 寄存器为 bit13~6、4、3) 置“0”，并且将 bit5 置 1”。

备注：m：单元号 (m=0)；

n：通道号 (n=0、1)；

p：SSPI 号 (p=00)；

q：UART 号 (q=0)；

r：IIC 号 (r=00)。

12.3.4 串行通信运行设定寄存器 mn (SCRmn)

SCRmn 寄存器是通道 n 的通信运行设定寄存器，设定数据发送和接收模式、数据和时钟相位、是否屏蔽错误信号、奇偶检验位、起始位、停止位和数据长度等。

禁止在运行过程中 (SEmn=1) 改写 SCRmn 寄存器。

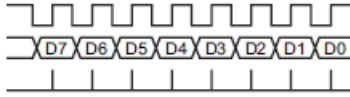
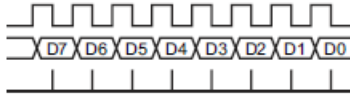
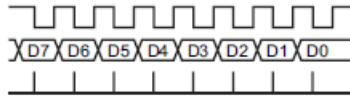
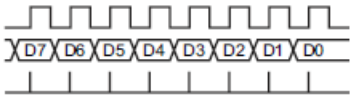
通过16位存储器操作指令设定 SCRmn 寄存器。

在产生复位信号后，SCRmn 寄存器的值变为“0083H”。

表12-9：串行通信运行设定寄存器 mn (SCRmn) 的格式

位	符号	描述	复位值
15:14	TXEmn,RXEmn	通道n运行模式的设定 00: 禁止通信。 01: 只进行接收。 10: 只进行发送。 11: 使能发送和接收。	0x0
13:12	DAPmn,CKPmn	SSPI模式中的数据和时钟的相位选择 00: 类型1 01: 类型2 10: 类型3 11: 类型4 在UART模式中，必须将DAPmn和CKPmn位都置0	0x0
11	-	-	0
10	EOCmn	错误中断信号 (INTSREx (x=0~1)) 的屏蔽控制 0: 禁止产生错误中断INTSREx (产生INTSRx)。 1: 允许产生错误中断INTSREx (在发生错误时不产生INTSRx)。 在SSPI模式和简易I ² C模式中或者在UART发送时，必须将EOCmn位置“0” ^{注2} 。	0x0
9:8	PTCmn1~PTCmn0	UART模式中的奇偶校验位的设定 发送 00: 不输出奇偶校验位。 01: 输出奇偶校验 ^{注3} 。 10: 输出偶校验。 11: 输出奇校验。 接收 接收时没有奇偶校验。 不判断奇偶校验。 判断偶校验。 判断奇校验。 在SSPI模式和简易I ² C模式中，必须将PTCmn1位和PTCmn0位都置“0”。	0x0
7	DIRmn	SSPI和UART模式中的数据传送顺序的选择 0: 进行MSB优先的输入/输出。 1: 进行LSB优先的输入/输出。 在简易I ² C模式中，必须将DIRmn位置“0”。	0x1
6	-	-	0x0
5:4	SLCmn1~SLCmn0	UART模式中的停止位的设定 00: 无停止位 01: 停止位长度=1位 10: 停止位长度=2位 (只限于mn=00) 11: 禁止设定。 如果选择了传送结束中断，就在传送完所有停止位后产生中断。 在UART接收时或者在简易I ² C模式中，必须设定为1个停止位 (SLCmn1、SLCmn0=0、1)。 在SSPI模式中，必须设定为无停止位 (SLCmn1、SLCmn0=0、0)。 在UART发送时，必须设定为1位 (SLCmn1、SLCmn0=0、1) 或者2位 (SLCmn1、SLCmn0=1、0)。	0x0
3:2	-	保留	-
1:0	DLSmn1~DLSmn0	SSPI模式中的数据长度的设定 10: 7位数据长度 (保存在SDRmn寄存器的bit0~6) 串行功能对应 SSPI UART IIC ○ ○ ×	0x3

		11: 8位数据长度（保存在SDRmn寄存器的bit0~7） 其他 禁止设置	○ ○ ○	
		在简易I ² C模式中，必须设置DLSmn1~DLSmn0=11B。		

DAPmn	CKPmn	SSPI 模式中的数据与时钟的相位选择	类型
0	0	SCLKp	1
0	1	SCLKp	2
1	0	SCLKp	3
1	1	SCLKp	4
在 UART 模式和简易 I ² C 模式中，必须将 DAPmn 位和 CKPmn 位都置“0”。			

注1：只限于 SCR00寄存器；

注2：在 EOCmn 位为“0”并且不使用 SSPImn 时，有可能产生错误中断 INTSREn；

注意：必须将 bit6、10、11置“0”（也必须将 SCR01、寄存器的 bit5置“0”）；

备注：m：单元号（m=0）；

 n：通道号（n=0、1）；

 p：SSPI 号（p=00）。

12.3.5 串行数据寄存器 mn (SDRmn)

SDRmn 寄存器是通道 n 发送和接收的数据寄存器（16 位）。

当运行停止（SEmn=0）时，bit15~9 用作运行时钟（FMCK）的分频设定寄存器。在运行期间（SEmn=1）bit15~9 用作发送接收缓冲寄存器。

如果将串行模式寄存器 mn（SMRmn）的 CCSmn 位置“0”，由 SDRmn 寄存器的 bit15~9（高 7 位）设定的运行时钟的分频时钟就用作传送时钟。

通过 16 位存储器操作指令设定 SDRmn 寄存器。

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

表12-10：串行数据寄存器 mn（SDRmn）的格式

位	符号	描述	复位值
15:9		运行时钟分频的传送时钟设定 7'h0: FMCK 7'h1: FMCK/2 7'h2: FMCK/3 7'h3: FMCK/4 . 7'h7E: FMCK/127 7'h7F: FMCK/128 当运行停止（SEmn=0）时，bit15~9 用作运行时钟（FMCK）的分频设定寄存器。在运行期间（SEmn=1）bit15~9 用作发送接收缓冲寄存器	0x0
8:0		在运行期间（SEmn=1）用作发送接收缓冲寄存器	0x0

注意：

1. 运行停止（SEmn=0）时，必须将 bit8~0 清零；
2. 在使用 UART 时，禁止将 SDRmn[15:9] 设定为“0000000B”和“0000001B”；
3. 在使用简易 I²C 时，禁止将 SDRmn[15:9] 设定为“0000000B”，SDRmn[15:9] 的设定值必须大于等于“0000001B”；
4. 在运行停止（SEmn=0）时，禁止通过 8 位存储器操作指令改写 SDRmn[7:0]（否则，SDRmn[15:9] 全部被清“0”）；

备注：

SDRmn 寄存器在运行期间的功能，请参照“12.2 通用串行通信单元的结构”；

m：单元号（m=0）；

n：通道号（n=0、1）。

12.3.6 串行标志清除触发寄存器 mn（SIRmn）

这是用于清除通道 n 各错误标志的触发寄存器。

如果将各位（FECTmn、PECTmn、OVCTmn）置“1”，就将串行状态寄存器 mn（SSRmn）的对应位（FEFmn、PEFmn、OVFmn）清“0”。因为 SIRmn 寄存器是触发寄存器，所以如果清除 SSRmn 寄存器的对应位，也会立即清除 SIRmn 寄存器。

通过16位存储器操作指令设定 SIRmn 寄存器。

能用 SIRmnL 并且通过8位存储器操作指令设定 SIRmn 寄存器的低8位。

在产生复位信号后，SIRmn 寄存器的值变为“0000H”。

表12-11：串行标志清除触发寄存器 mn（SIRmn）的格式

位	符号	描述	复位值
15:3	-	-	0x0
2	FECTmn ^{注1}	通道n帧错误标志的清除触发 0: 不清除。 1: 将SSRmn寄存器的FEFmn位清“0”。	0
1	PECTmn	通道n奇偶校验错误标志的清除触发 0: 不清除。 1: 将SSRmn寄存器的PEFmn位清“0”。	0
0	OVCTmn	通道n溢出错误标志的清除触发 0: 不清除。 1: 将SSRmn寄存器的OVFmn位清“0”。	0

注1：只限于 SIR01寄存器；

注意：必须将 bit15~3（SIR00、SIR10寄存器为 bit15~2）置“0”；

备注：

m: 单元号（m=0）；

n: 通道号（n=0、1）；

SIRmn 寄存器的读取值总是“0000H”。

12.3.7 串行状态寄存器 mn (SSRmn)

SSRmn 寄存器表示通道 n 的通信状态和发生错误的情况。表示的错误为帧错误、奇偶校验错误和溢出错误。通过16位存储器操作指令读取 SSRmn 寄存器。

能用 SSRmnL 并且通过8位存储器操作指令读取 SSRmn 寄存器的低8位。

在产生复位信号后，SSRmn 寄存器的值变为“0000H”。

表12-12：串行状态寄存器 mn (SSRmn) 的格式(1/2)

位	符号	描述	复位值
15:7	-	-	0x0
6	TSFmn	通道n通信状态的表示标志 0: 通信停止状态或者通信待机状态 1: 通信运行状态 [清除条件] • 当将STmn寄存器的STmn位置“1”（通信停止状态）或者将SSmn寄存器的SSmn位置“1”（通信待机状态）时 • 当通信结束时 [置位条件] • 当开始通信时	0x0
5	BFFmn	通道n缓冲寄存器的状态表示标志 0: SDRmn寄存器没有保存有效数据。 1: SDRmn寄存器保存了有效数据。 [清除条件] • 在发送过程中传送完从SDRmn寄存器到移位寄存器的发送数据时 • 在接收过程中从SDRmn寄存器读完接收数据时 • 当将STmn寄存器的STmn位置“1”（通信停止状态）或者将SSmn寄存器的SSmn位置“1”（通信允许状态）时 [置位条件] • 在SCRmn寄存器的TXEmn位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给SDRmn寄存器写发送数据时 • 在SCRmn寄存器的RXEmn位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到SDRmn寄存器时 • 当发生接收错误时	0x0
4:3	-	-	0x0
2	FEFmn ^{注1}	通道n帧错误的检测标志 0: 没有发生错误。 1: 发生错误（UART接收时）。 [清除条件] • 当给SIRmn寄存器的FECTmn位写“1”层面上名称时 [置位条件] • 在UART接收结束时没有检测到停止位时	0x0
1	PEFmn	通道n奇偶校验错误的检测标志 0: 没有发生错误。 1: 发生错误（UART接收时）或者未检测到ACK（I ² C发送时）。 [清除条件] • 当给SIRmn寄存器的PECTmn位写“1”时 [置位条件] • 在UART接收结束时发送数据的奇偶校验和奇偶校验位不同（奇偶校验错误）时 • 在I ² C发送时并且在ACK接收时序从属方没有返回ACK信号（未检测到ACK）时	0x0 0x0
0	OVFmn	通道n溢出错误的检测标志 0: 没有发生错误。 1: 发生错误。 [清除条件] • 当给SIRmn寄存器的OVCTmn位写“1”时 [置位条件]	0x0

		- 在SCRmn寄存器的RXEmn位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下，虽然接收数据被保存在SDRmn寄存器，但是没有读接收数据而写发送数据或者写下一个接收数据时 - 在SSPI模式的从属发送或者从属发送和接收过程中未准备好发送数据时	

注1：只限于 SSR01寄存器。

备注：如果在 BFFmn 位为“1”时写 SDRmn 寄存器，就会丢弃被保存的发送或者接收数据，并且检测到溢出错误（OVEmn=1）。

m：单元号（m=0）

n：通道号（n=0、1）

12.3.8 串行通道开始寄存器 m (SSm)

SSm 寄存器是设定允许各通道的通信/开始计数的触发寄存器。

如果给各位 (SSmn) 写“1”，就将串行通道允许状态寄存器 m (SEm) 的对应位 (SEmn) 置“1”（运行允许状态）。因为 SSmn 位是触发位，所以如果 SEmn 位为“1”就立即清除 SSmn 位。

通过16位存储器操作指令设定 SSm 寄存器。

能用 SSmL 并且通过8位存储器操作指令设定 SSm 寄存器的低8位。

在产生复位信号后，SSm 寄存器的值变为“0000H”。

表12-13: 串行通道开始寄存器 m (SSm) 的格式

位	符号	描述	复位值
15:2	-	-	0x0
1	SSm1	通道n运行开始的触发 0: 没有触发。 1: 将SEmn位置“1”，转移到通信待机状态 ^注 。	0
0	SSm0	通道n运行开始的触发 0: 没有触发。 1: 将SEmn位置“1”，转移到通信待机状态 ^注 。	0

注：如果在通信过程中将 SSmn 位置“1”，就停止通信进入待机状态。此时，控制寄存器和移位寄存器的值、SCLKmn 引脚和 SDOmn 引脚、FEFmn 标志、PEFmn 标志和 OVFn 标志保持状态；

注意：必须将 SS0寄存器的 bit15~2置“0”；

备注：

m: 单元号 (m=0)；

n: 通道号 (n=0、1)；

SSm 寄存器的读取值总是“0000H”。

12.3.9 串行通道停止寄存器 m (STm)

STm 寄存器是设定允许各通道的通信/停止计数的触发寄存器。

如果给各位 (STmn) 写“1”，就将串行通道允许状态寄存器 m (SEm) 的对应位 (SEmn) 清“0” (运行停止状态)。因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

通过16位存储器操作指令设定 STm 寄存器。

能用 STmL 并且通过8位存储器操作指令设定 STm 寄存器的低8位。

在产生复位信号后，STm 寄存器的值变为“0000H”。

表12-14: 串行通道停止寄存器 m (STm) 的格式

位	符号	描述	复位值
15:2	-	-	0x0
1	STm1	通道n运行的停止触发 0: 没有触发。 1: 将SEmn位清“0”，停止通信运行 ^注 。	0
0	STm0	通道n运行的停止触发 0: 没有触发。 1: 将SEmn位清“0”，停止通信运行 ^注 。	0

注：控制寄存器和移位寄存器的值、SCLKmn 引脚和 SDOmn 引脚以及 FEFmn 标志、PEFmn 标志和 OVFMn 标志保持状态；

注意：必须将 ST0 寄存器的 bit15~2 置“0”；

备注：

m: 单元号 (m=0)；

n: 通道号 (n=0、1)；

STm 寄存器的读取值总是“0000H”。

12.3.10 串行通道允许状态寄存器 m (SEm)

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

如果给串行开始允许寄存器 m (SSm) 的各位写“1”，就将其对应位置“1”。如果给串行通道停止寄存器 m (STm) 的各位写“1”，就将其对应位清“0”。

对于允许运行的通道 n，无法通过软件改写后述的串行输出寄存器 m (SOM) 的 CKOmn 位（通道 n 的串行时钟输出）的值，而从串行时钟引脚输出由通信运行反映的值。

对于停止运行的通道 n，能通过软件设定 SOM 寄存器的 CKOmn 位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过16位存储器操作指令读取 SEm 寄存器。

能用 SEmL 并且通过8位存储器操作指令读取 SEm 寄存器的低8位。

在产生复位信号后，SEm 寄存器的值变为“0000H”。

表12-15：串行通道允许状态寄存器 m (SEm) 的格式

位	符号	描述	复位值
15:2	-	-	0x0
1	SEm1	通道n运行的允许或者停止状态的表示 0: 运行停止状态 1: 运行允许状态	0
0	SEm0	通道n运行的允许或者停止状态的表示 0: 运行停止状态 1: 运行允许状态	0

备注：m：单元号（m=0）；

n：通道号（n=0、1）。

12.3.11 串行输出允许寄存器 m (SOEm)

SOEm 寄存器设定允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道 n，无法通过软件改写后述的串行输出寄存器 m (SOm) 的 SOmn 位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道 n，能通过软件设定 SOm 寄存器的 SOmn 位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过16位存储器操作指令设定 SOEm 寄存器。

能用 SOEmL 并且通过8位存储器操作指令设定 SOEm 寄存器的低8位。

在产生复位信号后，SOEm 寄存器的值变为“0000H”。

表12-16：串行输出允许寄存器 m (SOEm) 的格式

位	符号	描述	复位值
15:2	-	-	0x0
1	SOEm1	通道n串行输出的允许或者停止 0: 停止串行通信的输出。 1: 允许串行通信的输出。	0
0	SOEm0	通道n串行输出的允许或者停止 0: 停止串行通信的输出。 1: 允许串行通信的输出。	0

备注：SOE0寄存器的 bit15~2置“0”；

m: 单元号 (m=0)；

n: 通道号 (n=0、1)。

12.3.12 串行输出寄存器 m (SOM)

SOM 寄存器是各通道串行输出的缓冲寄存器。

从通道 n 的串行数据输出引脚输出此寄存器的 SOMn 位的值。

从通道 n 的串行时钟输出引脚输出此寄存器的 CKOm 位的值。

只有在禁止串行输出时 (SOEmn=0) 才能通过软件改写此寄存器的 SOMn 位。当允许串行输出 (SOEmn=1) 时, 忽视通过软件的改写而只能通过串行通信更改此寄存器的 SOMn 位的值。

只有在通道停止运行时 (SEmn=0) 才能通过软件改写此寄存器的 CKOm 位。当允许通道运行 (SEmn=1) 时, 忽视通过软件的改写而只能通过串行通信更改此寄存器的 CKOm 位的值。

要将串行接口的引脚用作端口功能等非串行接口功能时, 必须将相应的 CKOm 位和 SOMn 位置“1”。

通过 16 位存储器操作指令设定 SOM 寄存器。

在产生复位信号后, SOM 寄存器的值变为“0303H”。

表 12-17: 串行输出寄存器 m (SOM) 的格式

位	符号	描述	复位值
15:10	-	-	0x0
9	CKOm1	通道 n 的串行时钟输出 0: 串行时钟的输出值为“0”。 1: 串行时钟的输出值为“1”。	1
8	CKOm0	通道 n 的串行时钟输出 0: 串行时钟的输出值为“0”。 1: 串行时钟的输出值为“1”。	1
7:2			0x0
1	SOM1	通道 n 的串行数据输出 0: 串行数据的输出值为“0”。 1: 串行数据的输出值为“1”。	1
0	SOM0	通道 n 的串行数据输出 0: 串行数据的输出值为“0”。 1: 串行数据的输出值为“1”。	1

注意: 必须将 SOM 寄存器的 bit15~10 和 bit7~2 置“0”;

备注: m: 单元号 (m=0);

n: 通道号 (n=0、1)。

12.3.13 串行输出电平寄存器 m (SOLm)

SOLm 寄存器是设定各通道的数据输出电平反相的寄存器。

只有在 UART 模式中才能设定此寄存器。在 SSPI 模式和简易 I²C 模式中，必须将对应位置“0”。只在允许串行输出时 (SOEmn=1)，将此寄存器的各通道 n 反相设定反映到引脚输出。在禁止串行输出时 (SOEmn=0)，将 SOmn 位的值直接输出。禁止在运行过程中 (SEmn=1) 改写 SOLm 寄存器。

通过16位存储器操作指令设定 SOLm 寄存器。

能用 SOLmL 并且通过8位存储器操作指令设定 SDOLm 寄存器的低8位。

在产生复位信号后，SOLm 寄存器的值变为“0000H”。

表12-18： 串行输出电平寄存器 m (SOLm) 的格式

位	符号	描述	复位值
15:1	-	-	0x0
0	SOLm0	UART模式中的通道n发送数据电平反相的选择 0: 将通信数据直接输出。 1: 将通信数据反相输出。	0x0

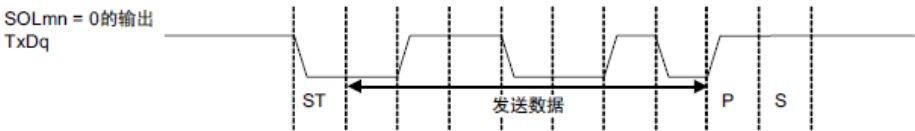
注意：必须将 SOL0寄存器的 bit15~bit1置“0”；

备注：m：单元号（m=0）；

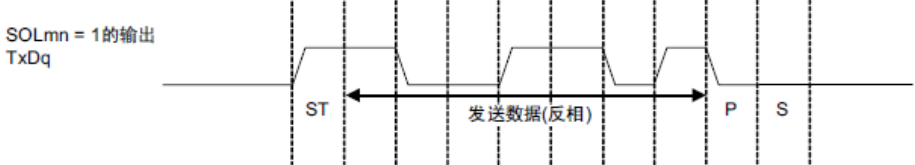
n：通道号（n=0、1）。

当进行 UART 发送时，发送数据的电平反相例子如：发送数据的电平反相例子

(a) 正相输出 (SOLmn = 0)



(b) 反相输出 (SOLmn = 1)



备注：m：单元号（m=0）；

n：通道号（n=0、1）。

12.3.14 输入切换控制寄存器（ISC）

在通过 UART0实现 LIN-bus 通信时，ISC 寄存器的 ISC1位和 ISC0位用于外部中断和定时器阵列单元的协调。如果将 bit0置“1”，就选择串行数据输入（RxD0）引脚的输入信号作为外部中断的输入（INTP0），因此能通过 INTP0中断检测唤醒信号。

如果将 bit1置“1”，就选择串行数据输入（RxD0）引脚的输入信号作为定时器的输入，因此能通过定时器检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽。

SSIE00位在 SSPI00通信的从属模式中控制通道0的 SS00引脚输入。在给 SS00引脚输入高电平的期间，即使输入串行时钟，也不进行发送和接收；在给 SS00引脚输入低电平的期间，如果输入串行时钟，就根据各模式的设定进行发送和接收。

通过8位存储器操作指令设定 ISC 寄存器。

在产生复位信号后，ISC 寄存器的值变为“00H”。

表12-19：输入切换控制寄存器（ISC）的格式

位	符号	描述	复位值
7	--	保留	0
6	SSIE00	SSPI00通信的从属模式中通道0的SS00输入的设置 0: SS00引脚输入无效。 1: SS00引脚输入有效。	0
5:0	--	保留	0

注意：必须将bit7,bit5~0置“0”。

12.3.15 噪声滤波器允许寄存器 0 (NFEN0)

NFEN0寄存器设定噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于 SSPI 或者简易 I²C 通信的引脚，必须将对应位置“0”，使噪声滤波器无效。对于用于 UART 通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，在通过对象通道的运行时钟（F_{MCK}）进行同步后检测2个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟（F_{MCK}）进行同步。

通过8位存储器操作指令设定 NFEN0寄存器。

在产生复位信号后，NFEN0寄存器的值变为“00H”。

表12-20：噪声滤波器允许寄存器0 (NFEN0) 的格式

位	符号	描述	复位值
7:3	-	保留	0x0
2	SNFEN10	RxD0引脚的噪声滤波器的使用与否 0: 噪声滤波器OFF 1: 噪声滤波器ON 当用作RxD0引脚时，必须将SNFEN10位置“1”。 当用作RxD0引脚以外的功能时，必须将SNFEN10位置“0”。	0x0
1:0	-	保留	0x0

注意：必须将 bit7~3，bit1~0置“0”。

12.3.16 控制串行输入/输出引脚端口功能的寄存器

在使用通用串行通信单元时，必须设定复用端口功能的控制寄存器（端口模式寄存器（PM_{xx}）、端口复用功能配置寄存器（PS_{xx}）、端口输出模式寄存器（POM_{xx}）和端口模式控制寄存器（PMC_{xx}））。

详细内容请参照“第2章 引脚功能”。

在将串行数据输出引脚或者串行时钟输出引脚的复用端口用作串行数据输出或者串行时钟输出时，必须将各端口对应的端口模式控制寄存器（PMC_{xx}）的位和端口模式寄存器（PM_{xx}）的位置“0”。此时，端口寄存器（P_{xx}）的位可以是“0”或者“1”。

另外，当用于 N 沟道漏极开路输出模式时，必须将各端口对应的端口输出模式寄存器（POM_{xx}）的位置“1”。

在将串行数据输入引脚或者串行时钟输入引脚的复用端口用作串行数据输入或者串行时钟输入时，必须将各端口对应的端口模式寄存器（PM_{xx}）的位置“1”，并且将端口模式控制寄存器（PMC_{xx}）的位置“0”。此时，端口寄存器（P_{xx}）的位可以是“0”或者“1”。

12.4 运行停止模式

通用串行通信单元的各串行接口有运行停止模式。在运行停止模式中不能进行串行通信，因此能降低功耗。另外，在运行停止模式中能将用于串行接口的引脚用作端口功能。

12.4.1 以单元为单位停止运行的情况

通过外围允许寄存器（PER0）设定以单元为单位的停止运行。

PER0寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过给不使用的硬件停止提供时钟，以降低功耗和噪声。

要停止通用串行通信单元0时，必须将 PER0的 bit3（SAUEN）置“0”；

表12-21：以单元为单位停止运行时的外围允许寄存器0（PER0）的设定

位	符号	描述	复位值
7	LPTIMEN	提供 15 位间隔定时器输入时钟的控制 0: 停止提供输入时钟，不能写 15 位间隔定时器使用的 SFR。 1: 提供输入时钟，能读写 15 位间隔定时器使用的 SFR。	0
6	-	保留	0
5	ADCEN	提供 ADC 模块的输入时钟的控制 0: 停止提供输入时钟，ADC 不能运行 1: 提供输入时钟，ADC 能运行	0
4	-	保留	0
3	SAUEN	提供串行通信单元输入时钟的控制 0: 停止提供输入时钟，不能写 SAU 使用的 SFR。 1: 提供输入时钟，能读写 SAU 使用的 SFR。	0
2:0	-	保留	0

备注：

- 当 SAUmEN 位为“0”时，忽视通用串行通信单元 m 的控制寄存器的写操作，而且读取值都为初始值。但是，以下的寄存器除外：
 - 输入切换控制寄存器（ISC）
 - 噪声滤波器允许寄存器 0（NFEN0）
 - 端口复用功能配置寄存器（PS_xx）
 - 端口输出模式寄存器（POMx）
 - 端口模式寄存器（PMx）
 - 端口寄存器（Px）
- ×：这是通用串行通信单元未使用的位（取决于其他外围功能的设定）；
0/1：根据用户的用途置“0”或者“1”。

12.4.2 按通道停止运行的情况

通过以下各寄存器设定按通道停止运行。

(1) 串行通道停止寄存器 m (STm) 这是设定允许各通道的通信/停止计数的寄存器。

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3	STm2	STm1	STm0

①将 SEmn 位清“0”并且停止通信运行

②因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

(2) 串行通道允许状态寄存器 m (SEm) 此寄存器表示各通道的数据发送和接收的运行或者停止状态。

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3	SEm2	SEm1	SEm0

①0: 运行停止状态

②SEm 寄存器是只读状态寄存器，通过 STm 寄存器来停止运行。对于已经停止运行的通道，能通过软件设定 SOm 寄存器的 CKOmn 位的值。

(3) 串行输出允许寄存器 m (SOEm) 这是设定允许或者停止各通道串行通信输出的寄存器。

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0

①0: 通过串行通信运行来停止输出

②对于已经停止串行输出的通道，能通过软件设定 SOm 寄存器的 SOmn 位的值。

(4) 串行输出寄存器 m (SOM) 这是各通道串行输出的缓冲寄存器。

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOM3	SOM2	SOM1	SOM0

①CKOm3/2/1/0: 1: 串行时钟的输出值为“1”; 0: 串行时钟的输出值为“0”

②SOM3/2/1/0: 1: 串行数据的输出值为“1”; 0: 串行数据的输出值为“0”

③当将各通道对应的引脚用作端口功能时，必须将相应的 CKOmn 位和 SOMn 位置“1”。

备注：只限于通用串行通信单元0；

m: 单元号 (m=0)；

n: 通道号 (n=0、1)；

0/1: 根据用户的用途置“0”或者“1”。

12.5 3 线串行 I/O（SSPI00）通信的运行

这是通过串行时钟（SCLK）和串行数据（SDI 和 SDO）共3条线实现的时钟同步通信功能。

[数据的发送和接收]

- 7位、8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率
- 主控通信：Max.6MHz
- 从属通信：Max.6MHz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

SCI0的通道0是支持3线串行 I/O（SSPI00）的通道。

3线串行 I/O（SSPI00）有以下6种通信运行：

- 主控发送（参照12.5.1）
- 主控接收（参照12.5.2）
- 主控的发送和接收（参照12.5.3）
- 从属发送（参照12.5.4）
- 从属接收（参照12.5.5）
- 从属的发送和接收（参照12.5.6）

注意：必须在满足 SCLK 周期时间（ T_{KCY} ）特性的范围内使用。详细内容请参照数据手册。

12.5.1 主控发送

主控发送是指本产品输出传送时钟并且将数据发送到其他设备的运行。

表12-22：主控发送

3线串行I/O	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDO00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	无
传送数据长度	7位、8位
传送速率 ^注	Max.6M[Hz] Min.F _{CLK} /(2x2 ¹¹ x128)[Hz] F _{CLK} : 系统时钟频率
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0: 正相 CKPmn=1: 反相
数据方向	MSB优先或者LSB优先

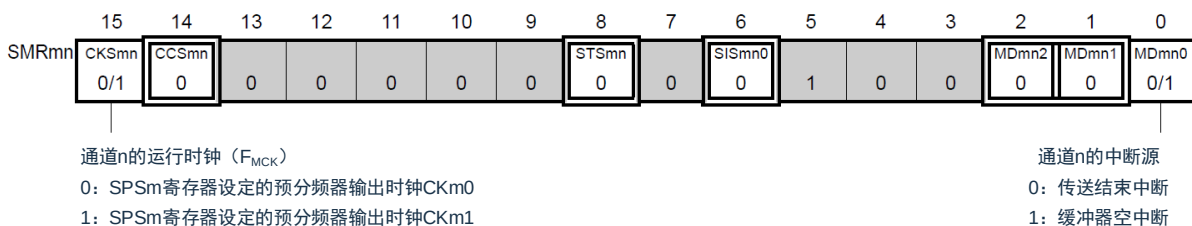
注：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

m: 单元号（m=0）；
n: 通道号（n=0、1）；
mn=00~01。

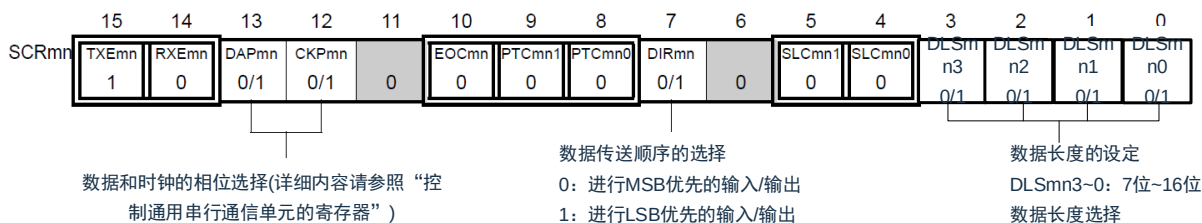
(1)寄存器的设定

图12-4：3线串行 I/O（SSPI00）主控发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

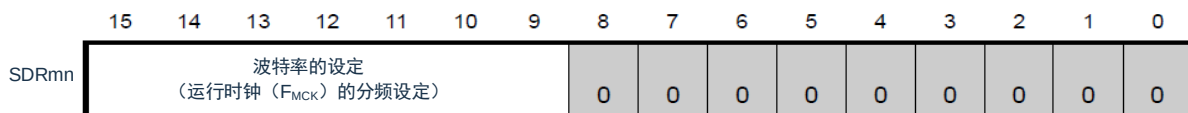


(b) 串行通信运行设定寄存器mn(SCRmn)

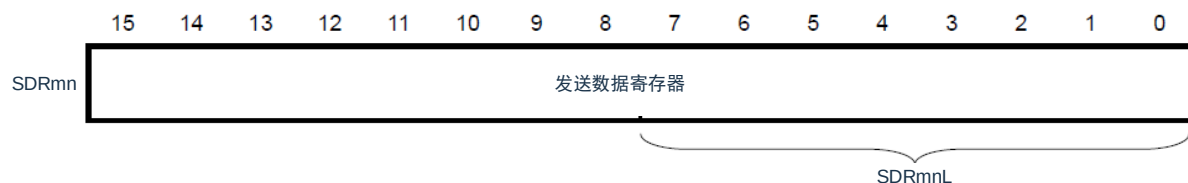


(c) 串行数据寄存器mn(SDRmn)

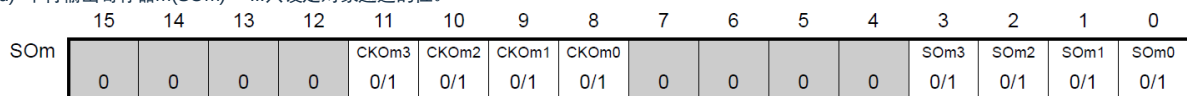
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)

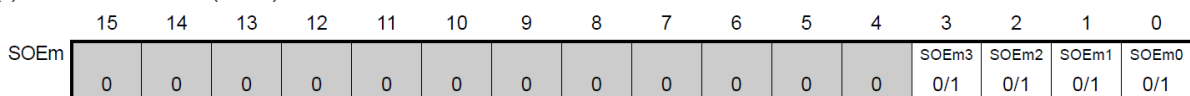


(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。

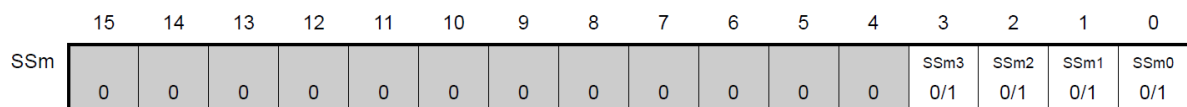


当时钟相位为“正相”(SCRmn寄存器的CKPmn=0)时,“1”为开始通信;当时钟相位为“反相”(CKPmn=1)时,“0”为开始通信。

(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0) n: 通道号 (n=0、1)

■: 不能设定(设定初始值)。0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图12-5: 主控发送的初始设置步骤

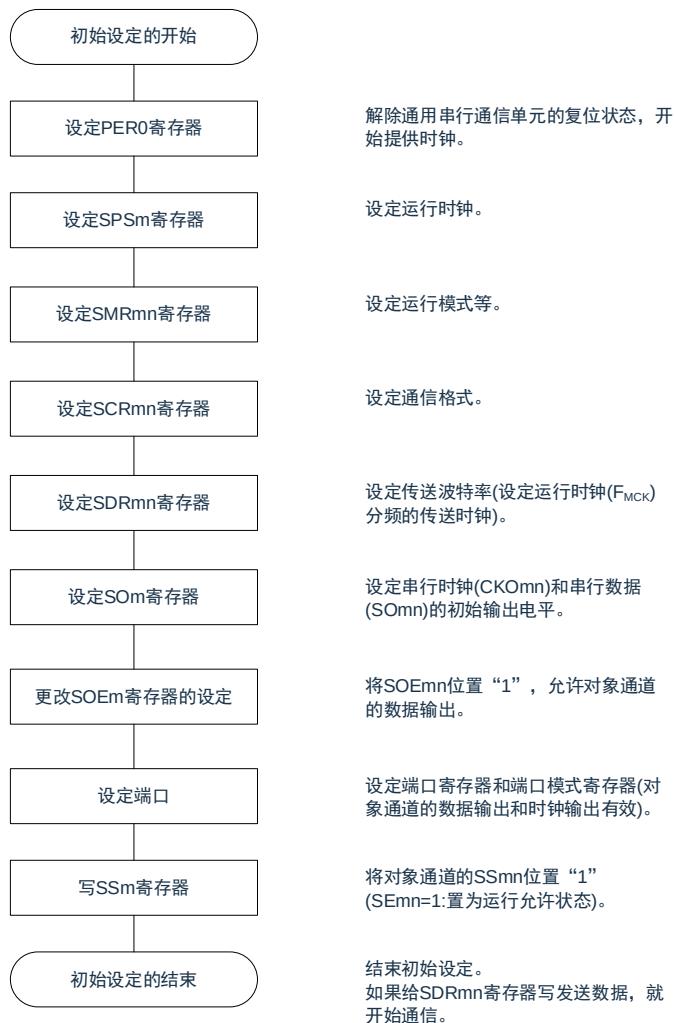


图12-6：主控发送的中止步骤

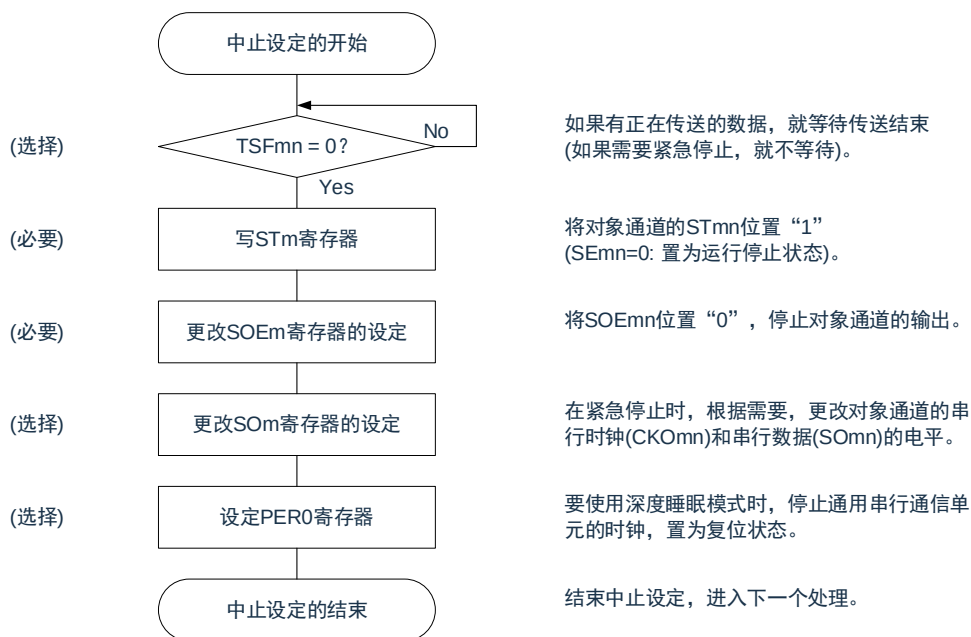


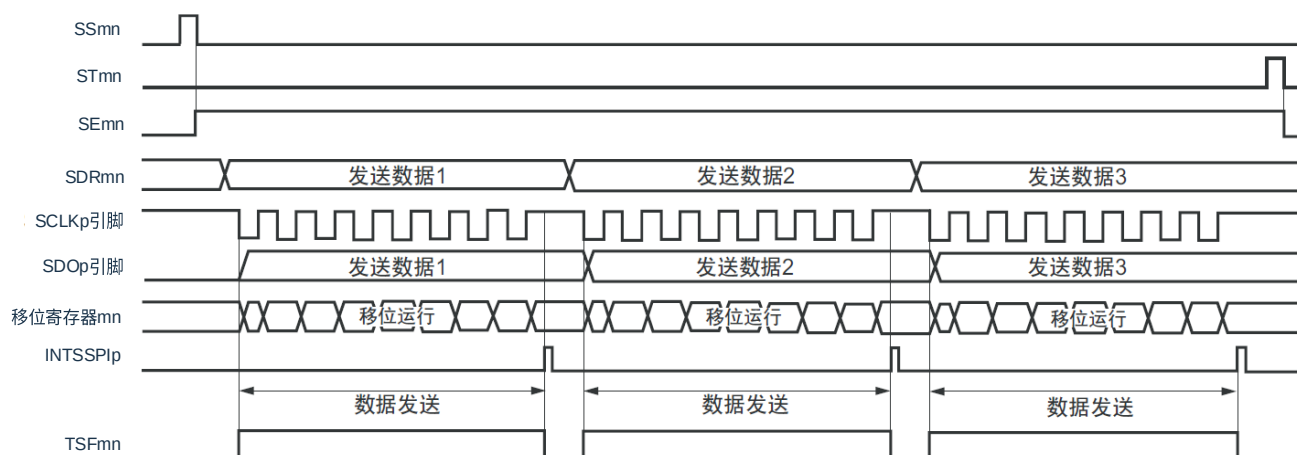
图12-7：重新开始主控发送的设定步骤



注意：如果在中止设定中改写 PER0来停止提供时钟，就必须在等到通信对象（从属设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

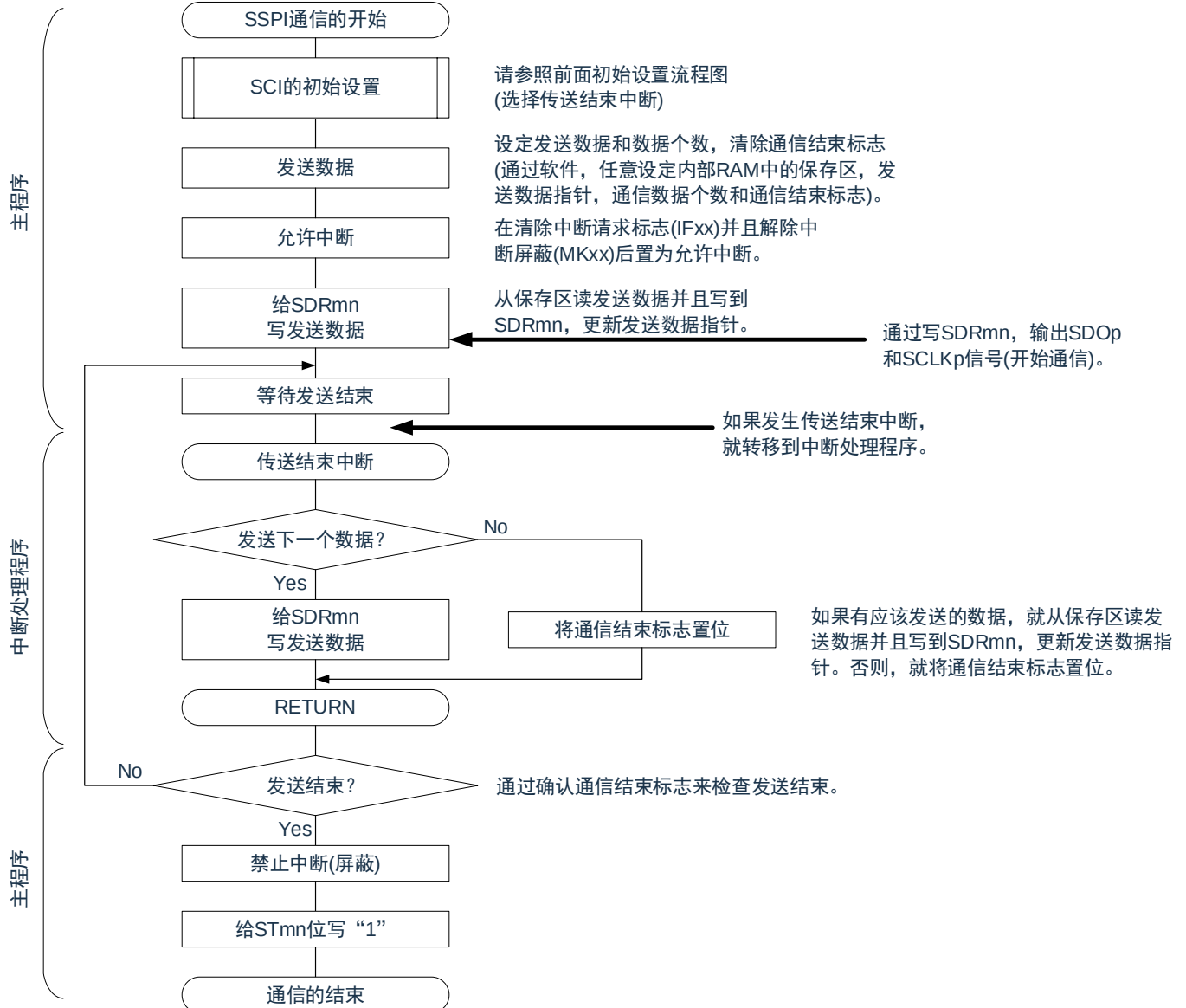
(3) 处理流程（单次发送模式）

图12-8：主控发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



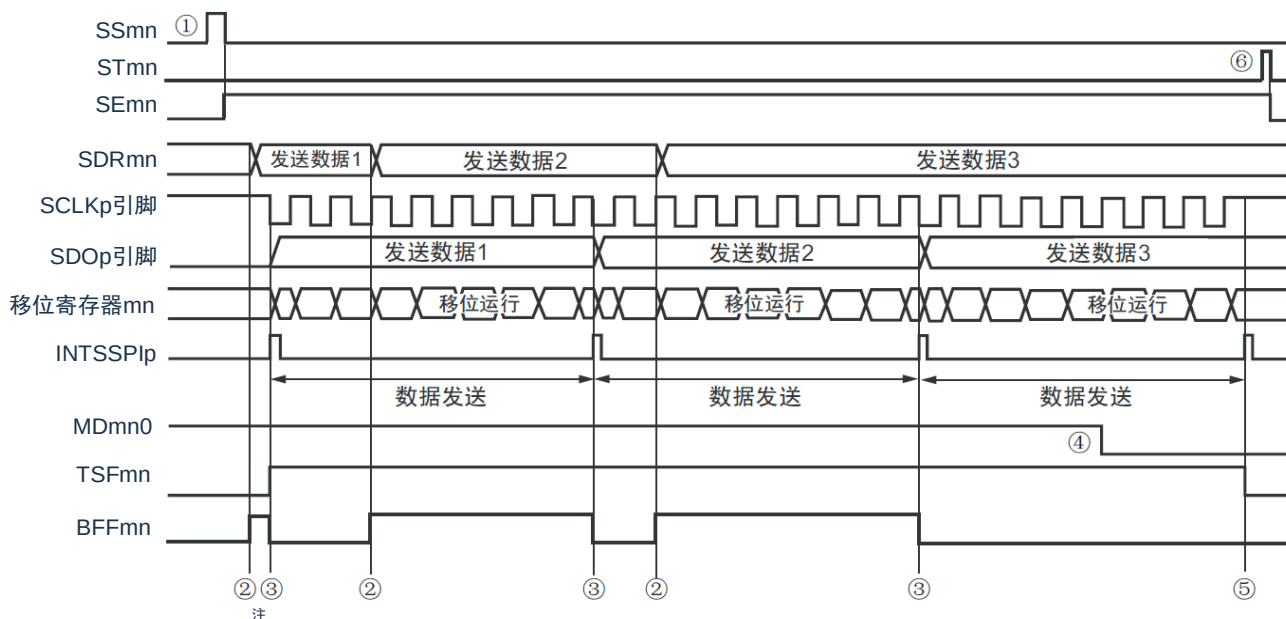
备注： m: 单元号 (m=0) ;
n: 通道号 (n=0、1) ;
p: SSPI 号 (p=00) 。

图12-9：主控发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图12-10：主控发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据；

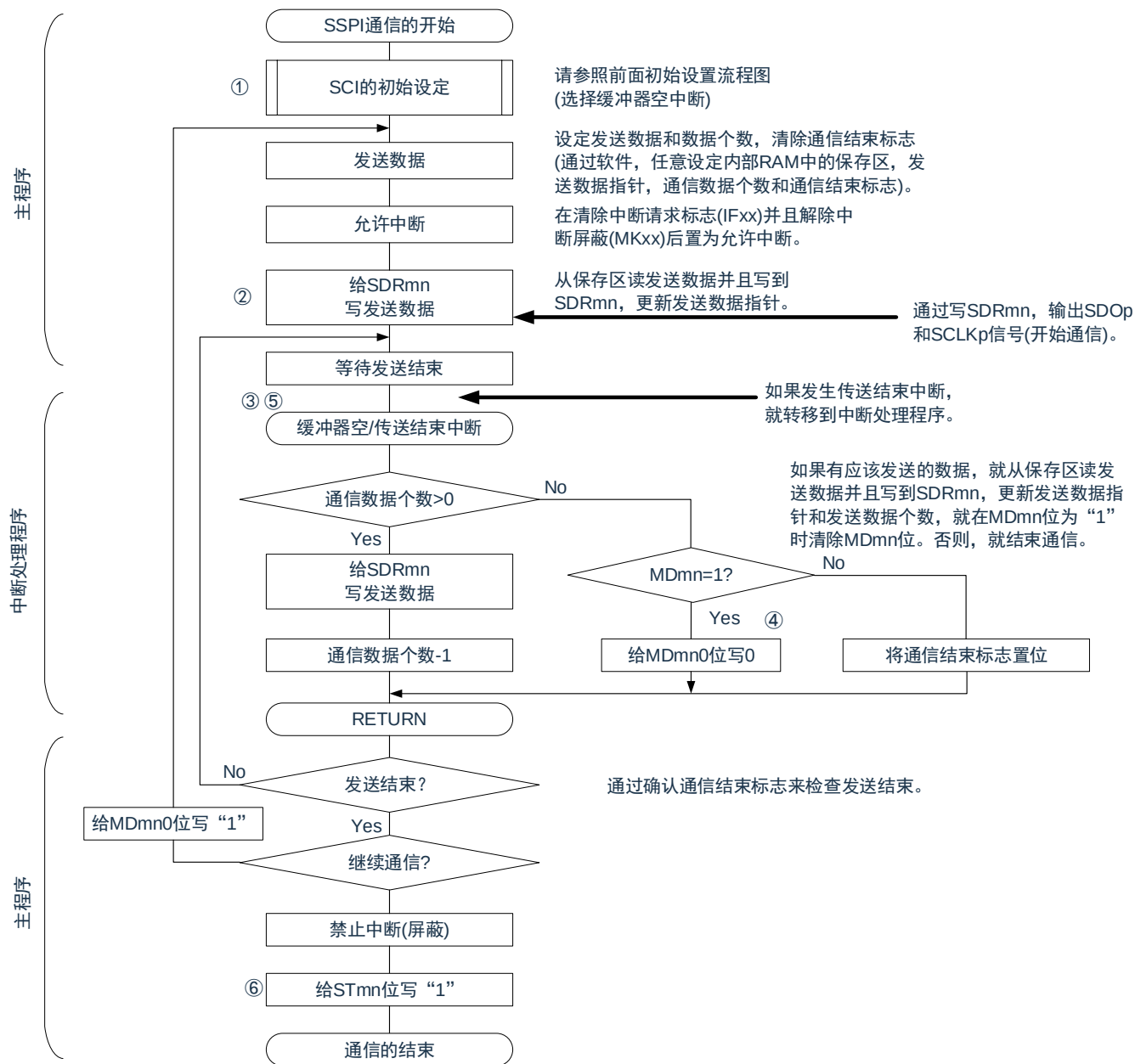
注意：即使在运行中能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写；

备注：m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）。

图12-11: 主控发送（连续发送模式）的流程图



备注: 图中的①~⑥对应“图12-10 主控发送（连续发送模式）的时序图”中的①~⑥。

12.5.2 主控接收

主控接收是指本产品输出传送时钟并且从其他设备接收数据的运行。

表12-23：主控接收

3线串行 I/O	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率 ^注	Max.6M[Hz] Min.FCLK/(2x2 ¹¹ x128)[Hz]FCLK：系统时钟频率
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0：在串行时钟开始运行时，开始数据输出。 DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0：正相 CKPmn=1：反相
数据方向	MSB 优先或者 LSB 优先

注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用；

备注：m：单元号（m=0）；

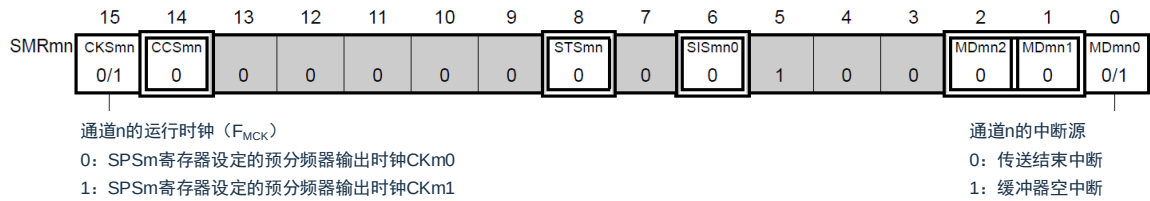
n：通道号（n=0、1）；

p：SSPI 号（p=00）。

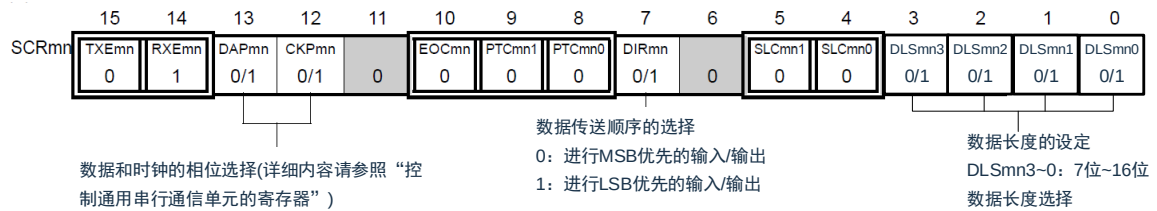
(1)寄存器的设定

图12-12：3线串行 I/O（SSPI00）主控接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

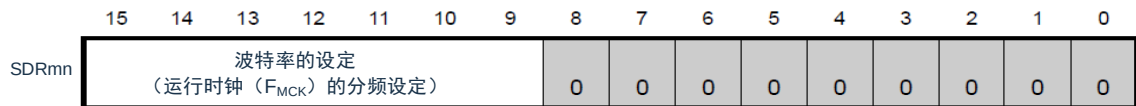


(b) 串行通信运行设定寄存器mn(SCRmn)

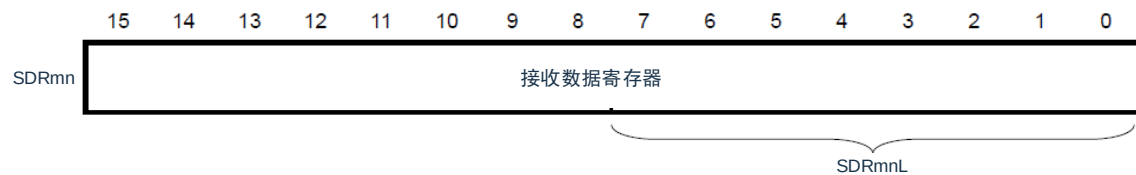


(c) 串行数据寄存器mn(SDRmn)

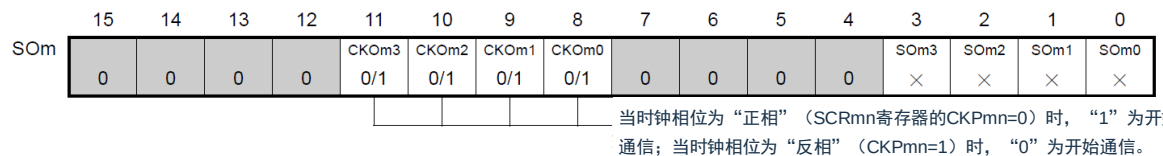
(1) 运行停止 (SEmn=0) 时



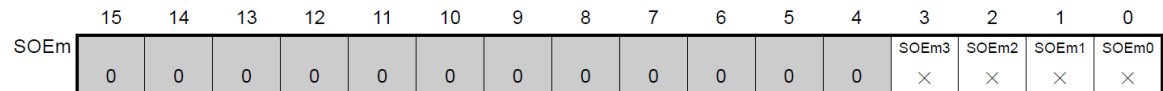
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



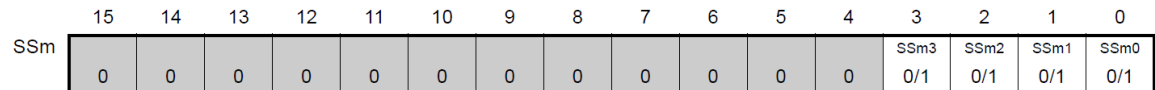
(d) 串行输出寄存器m(SOm) 只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) 在此模式中不使用



(f) 串行通道开始寄存器m(SSm) 只将对象通道的位置“1”。



备注: m: 单元号 (m=0);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00);

□ : 在 SSPI 主控接收模式中为固定设定; ■ : 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值);

0/1: 根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-13: 主控接收的初始设定步骤



图12-14：主控接收的中止步骤

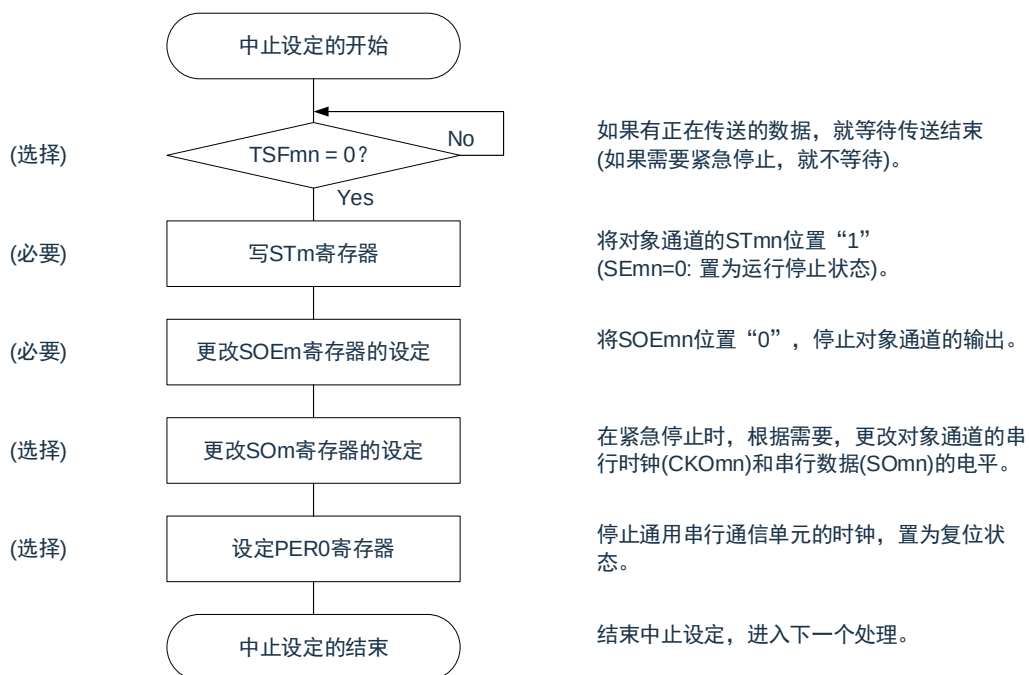


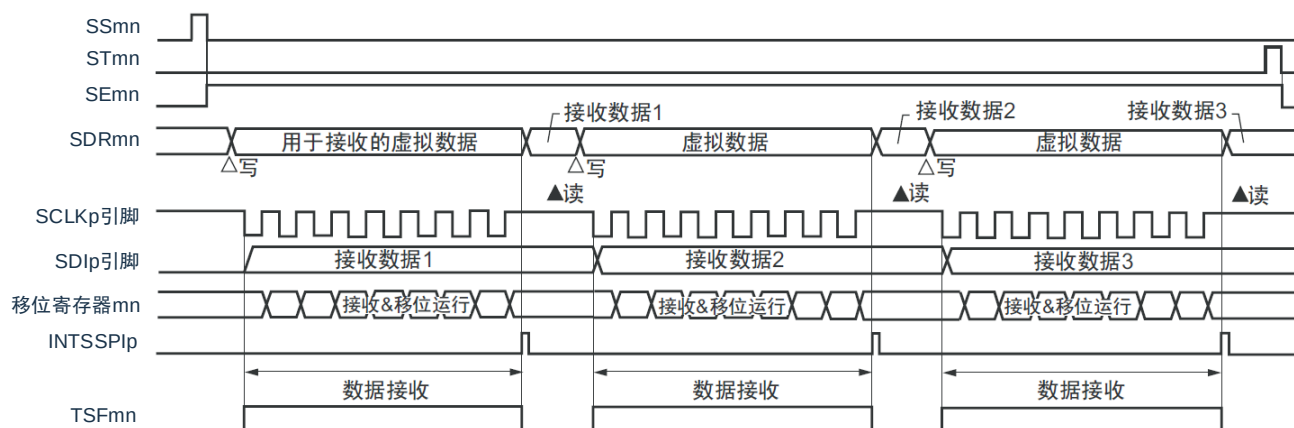
图12-15：重新开始主控接收的设定步骤



注意：如果在中止设定中改写 PER0来停止提供时钟，就必须在等到通信对象（从属设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

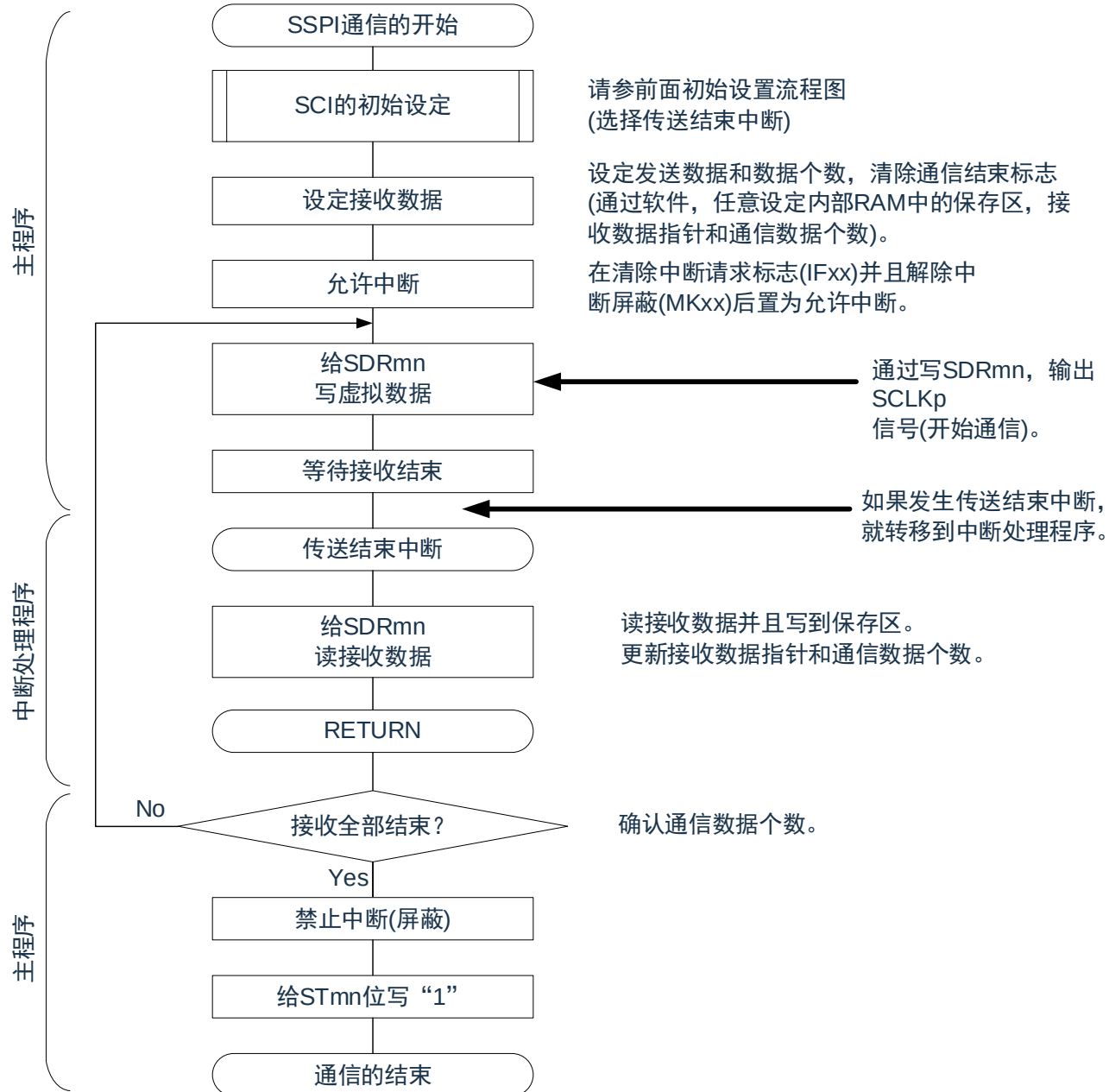
(3) 处理流程（单次接收模式）

图12-16：主控接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



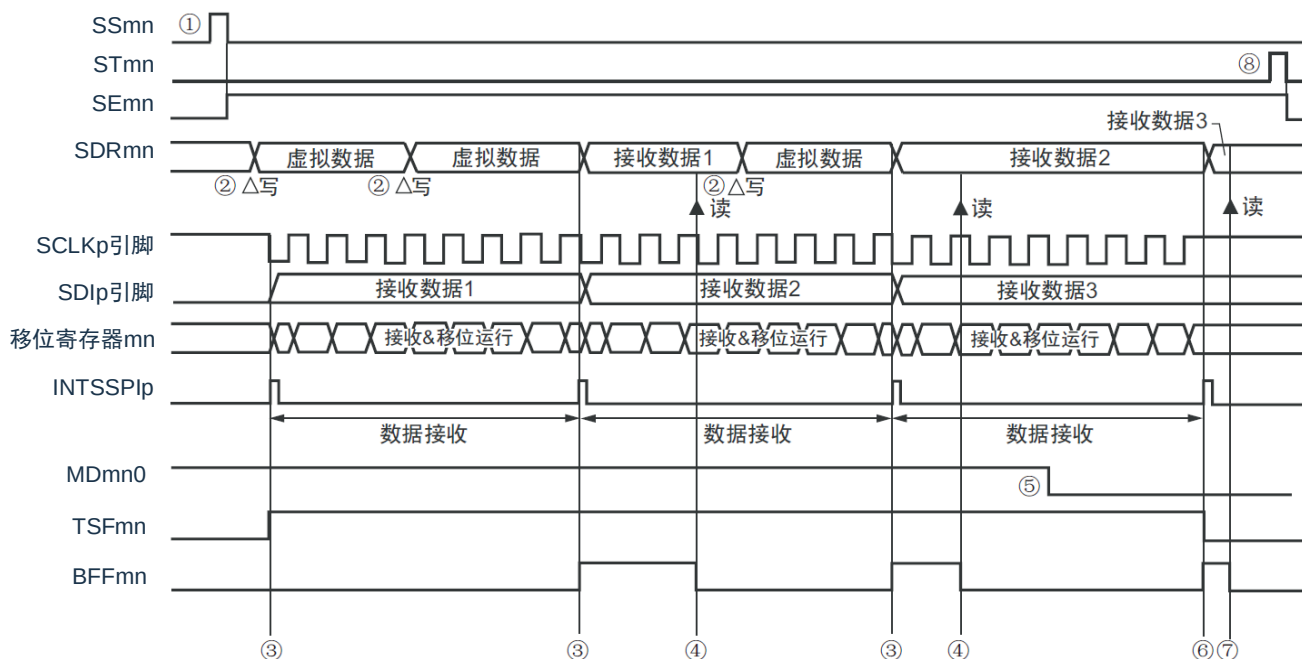
备注：m：单元号（m=0）；
n：通道号（n=0、1）；
p：SSPI 号（p=00）。

图12-17：主控接收（单次接收模式）的流程图



(4)处理流程（连续接收模式）

图12-18：主控接收（连续接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注意：即使在运行过程中也能改写 MDmn0 位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写；

备注：

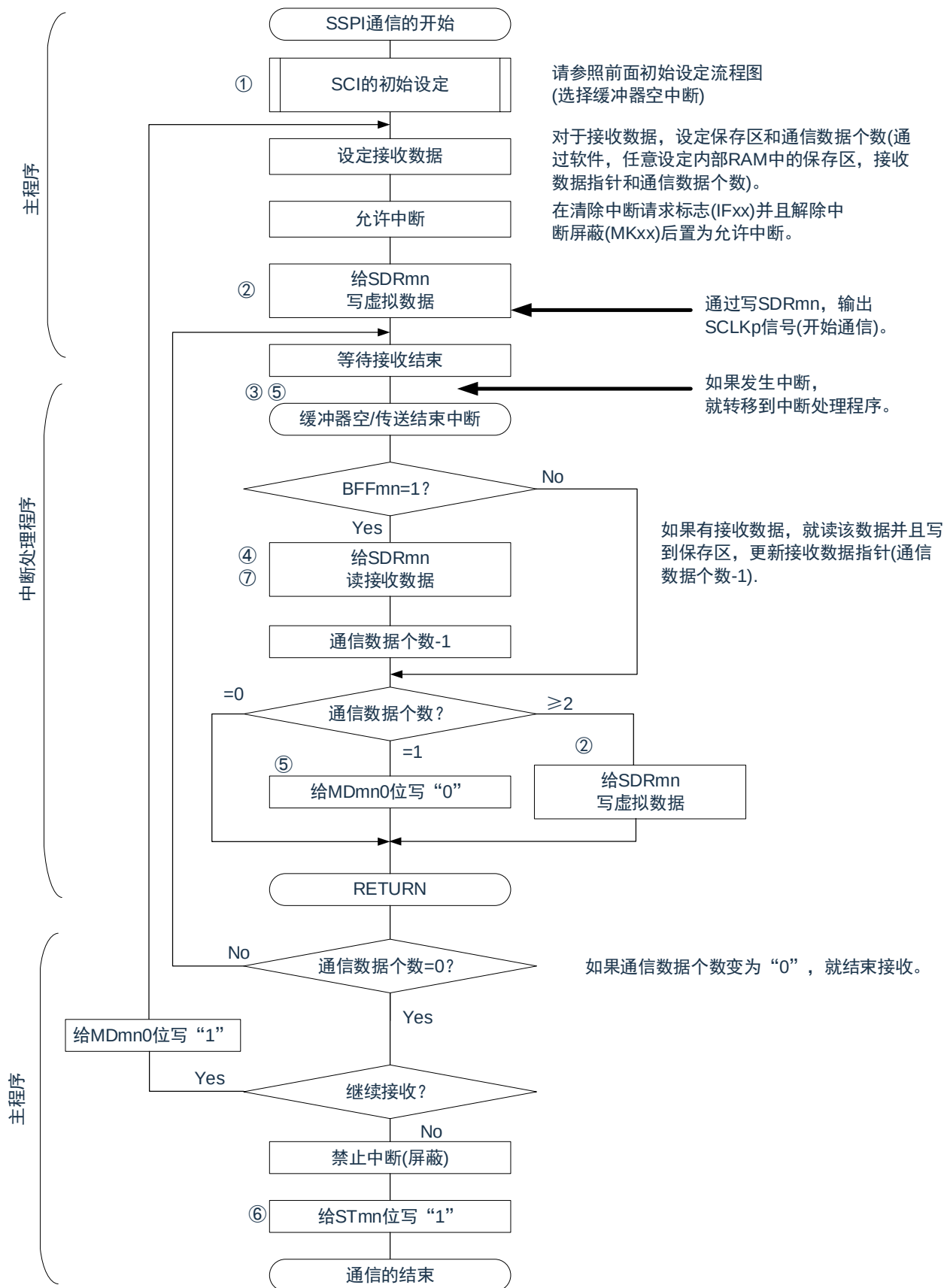
图中的①~⑧对应“图 12-19 主控接收（连续接收模式）的流程图”中的①~⑧；

m: 单元号 (m=0)；

n: 通道号 (n=0、1)；

p: SSPI 号 (p=00)。

图12-19: 主控接收（连续接收模式）的流程图



备注: 图中的①~⑧对应“图12-18 主控接收（连续接收模式）的时序图”中的①~⑧。

12.5.3 主控的发送和接收

主控的发送和接收是指本产品输出传送时钟并且和其他设备进行数据发送和接收的运行。

表12-24：主控的发送和接收

3线串行 I/O	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00、SDO00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率 ^注	Max.6M[Hz] Min. $F_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] F_{CLK} : 系统时钟频率
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0: 在串行时钟开始运行时，开始数据输出。 DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0: 正相 CKPmn=1: 反相
数据方向	MSB 优先或者 LSB 优先

注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用；

备注：m：单元号（m=0）；

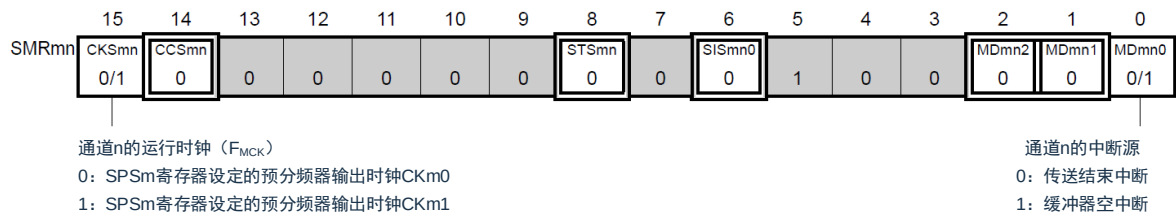
n：通道号（n=0、1）；

p：SSPI 号（p=00）。

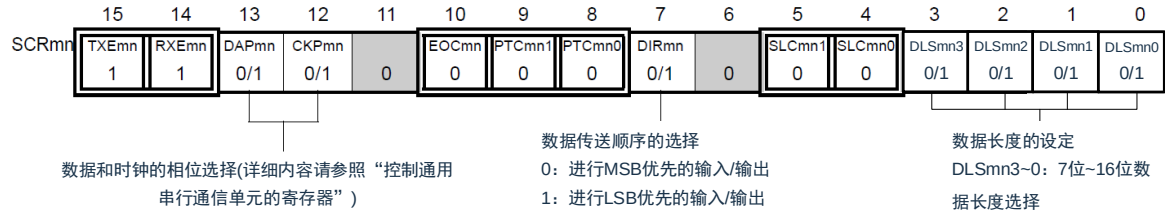
(1) 寄存器的设定

图12-20：3线串行 I/O（SSPI00）主控发送和接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

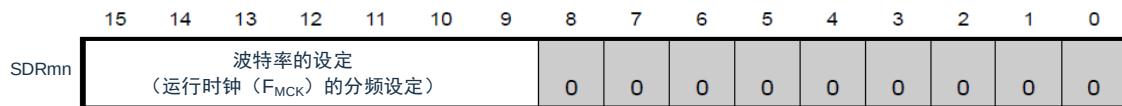


(b) 串行通信运行设定寄存器mn(SCRmn)

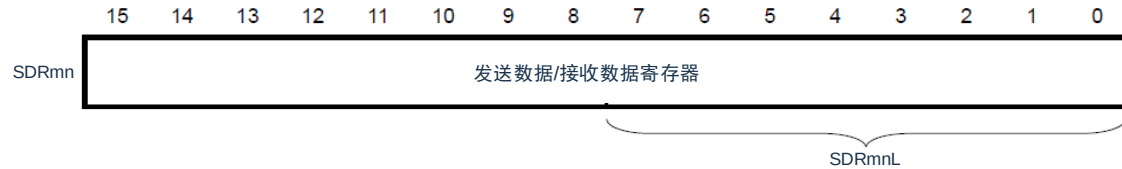


(c) 串行数据寄存器mn(SDRmn)

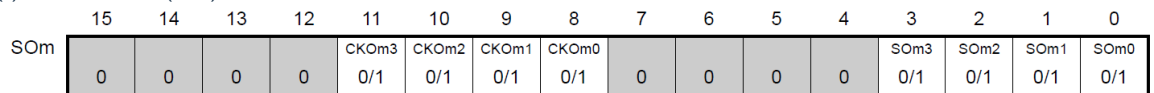
(1) 运行停止（SEmn=0）时



(2) 运行期间（SEmn=1）（低8位：SDRmnL）

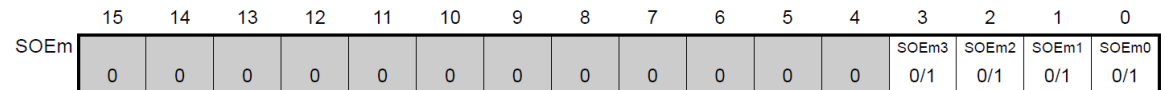


(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。

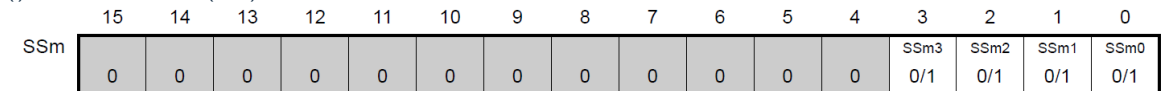


当时钟相位为“正相”（SCRmn寄存器的CKPmn=0）时，“1”为开始通信；当时钟相位为“反相”（CKPmn=1）时，“0”为开始通信。

(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注：m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI号（p=00）；

□：在 SSPI 主控接收模式中为固定设定； ■：不能设定（设定初始值）；

0/1：根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-21：主控发送和接收的初始设定步骤

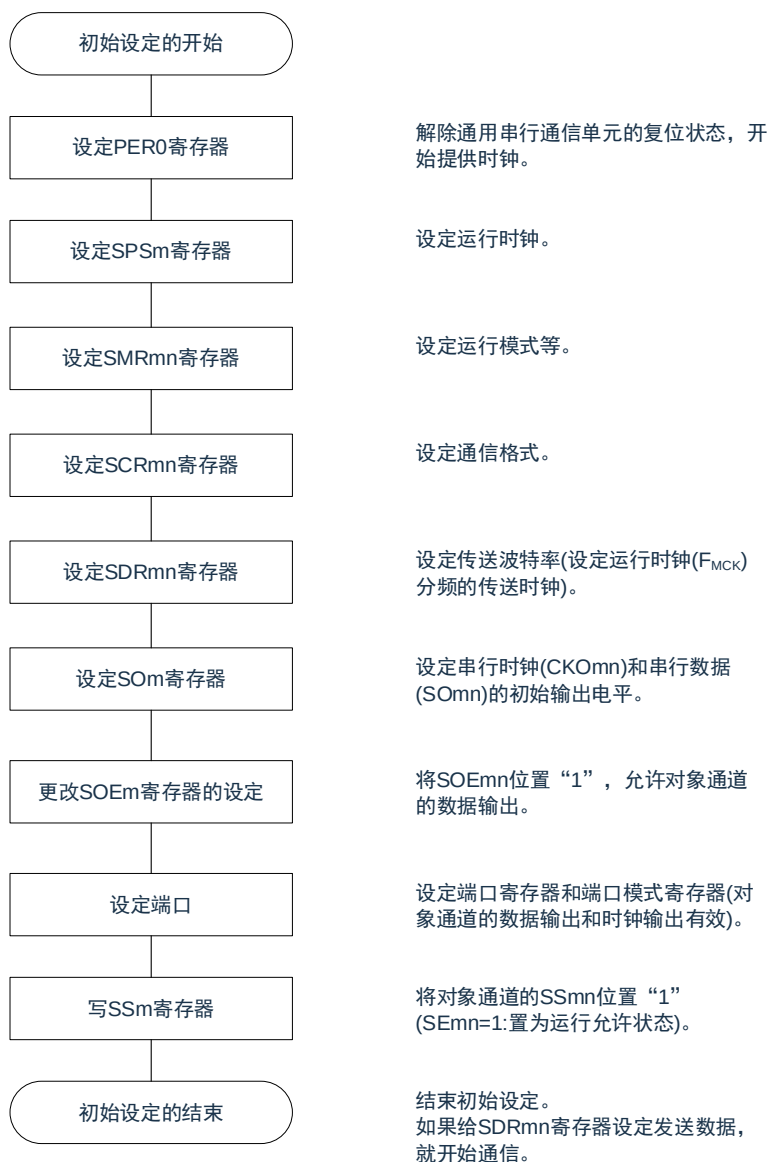


图12-22：主控发送和接收的中止步骤

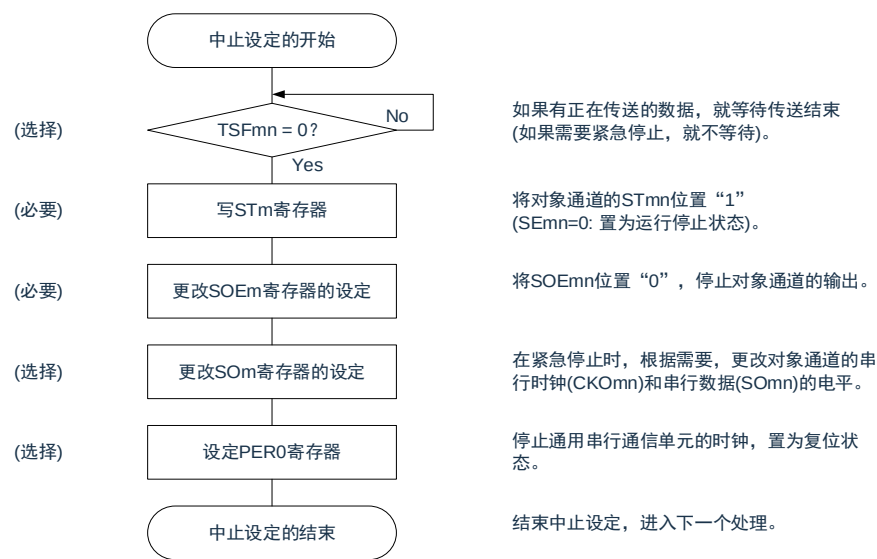
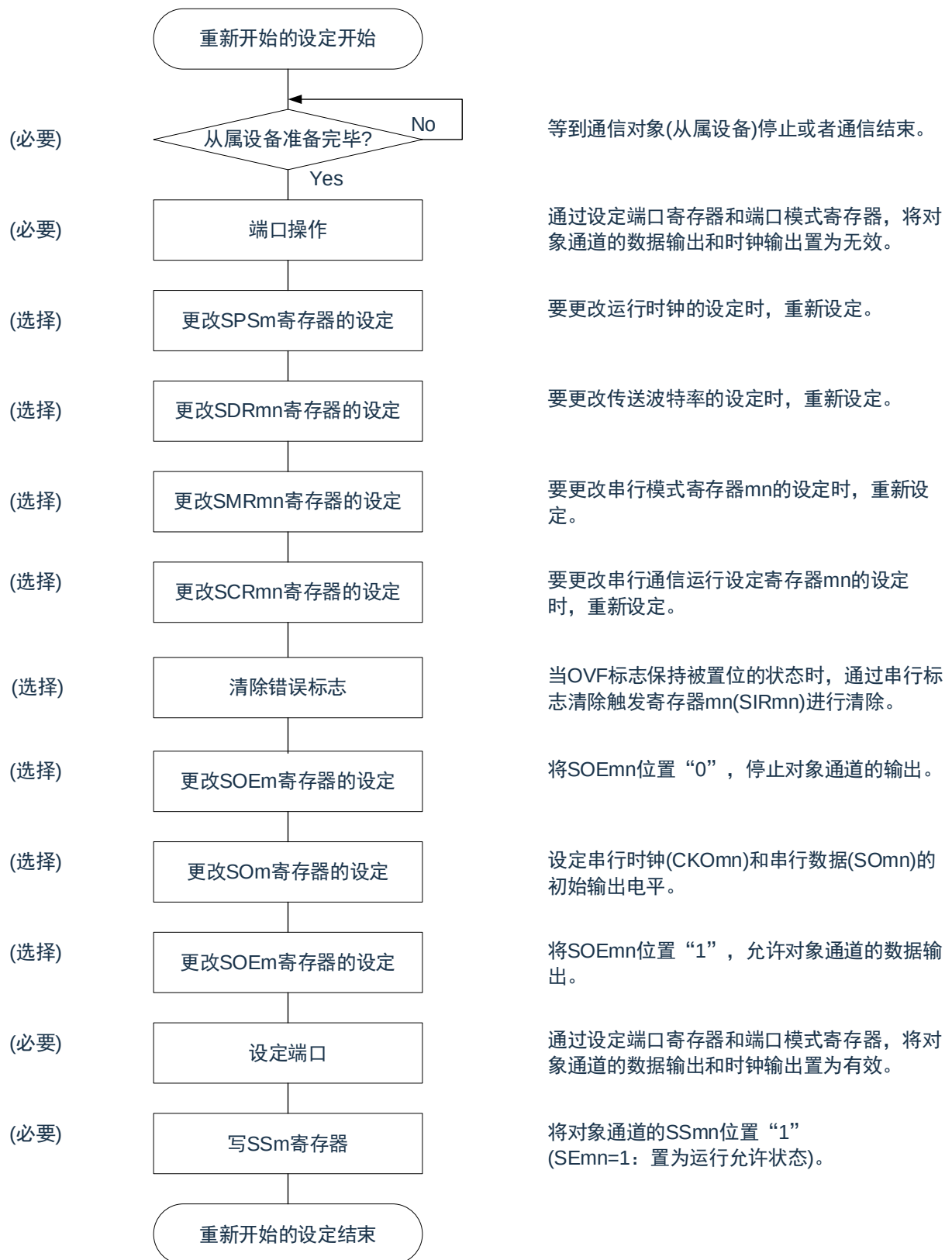
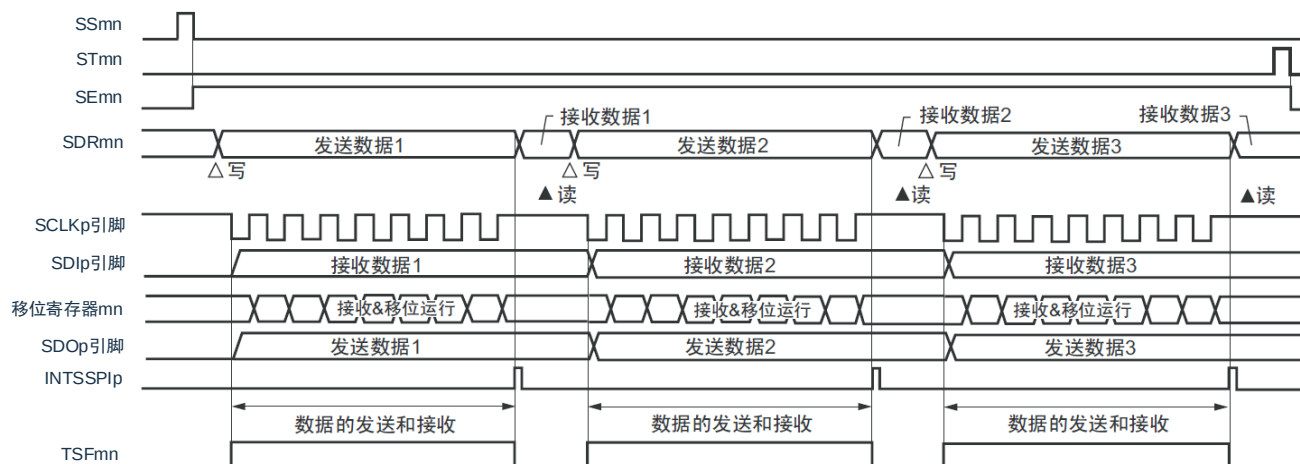


图12-23: 重新开始主控发送和接收的设定步骤



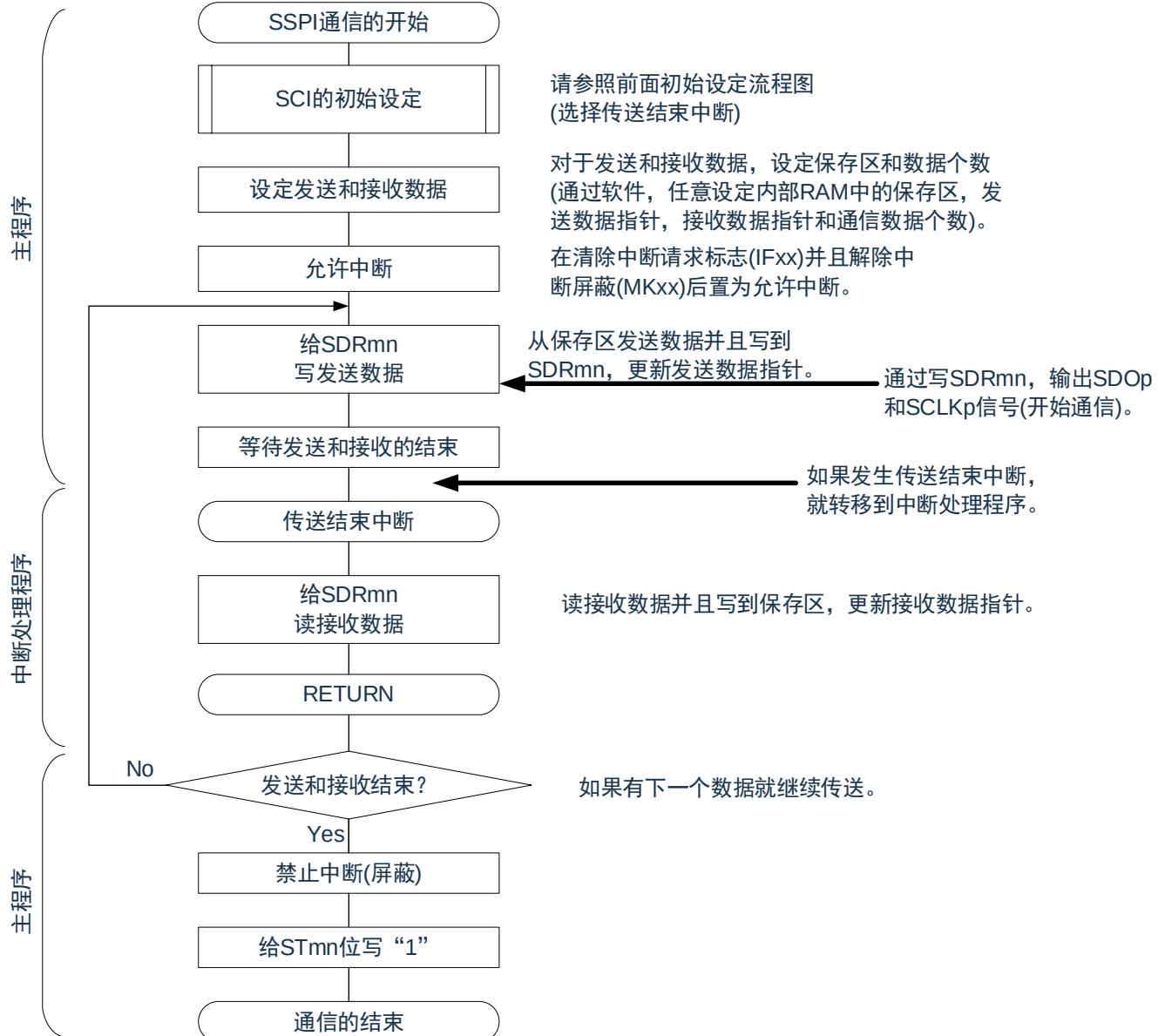
(3) 处理流程（单次发送和接收模式）

图12-24： 主控发送和接收（单次发送和接收模式）的时序图（类型1： DAPmn=0、CKPmn=0）



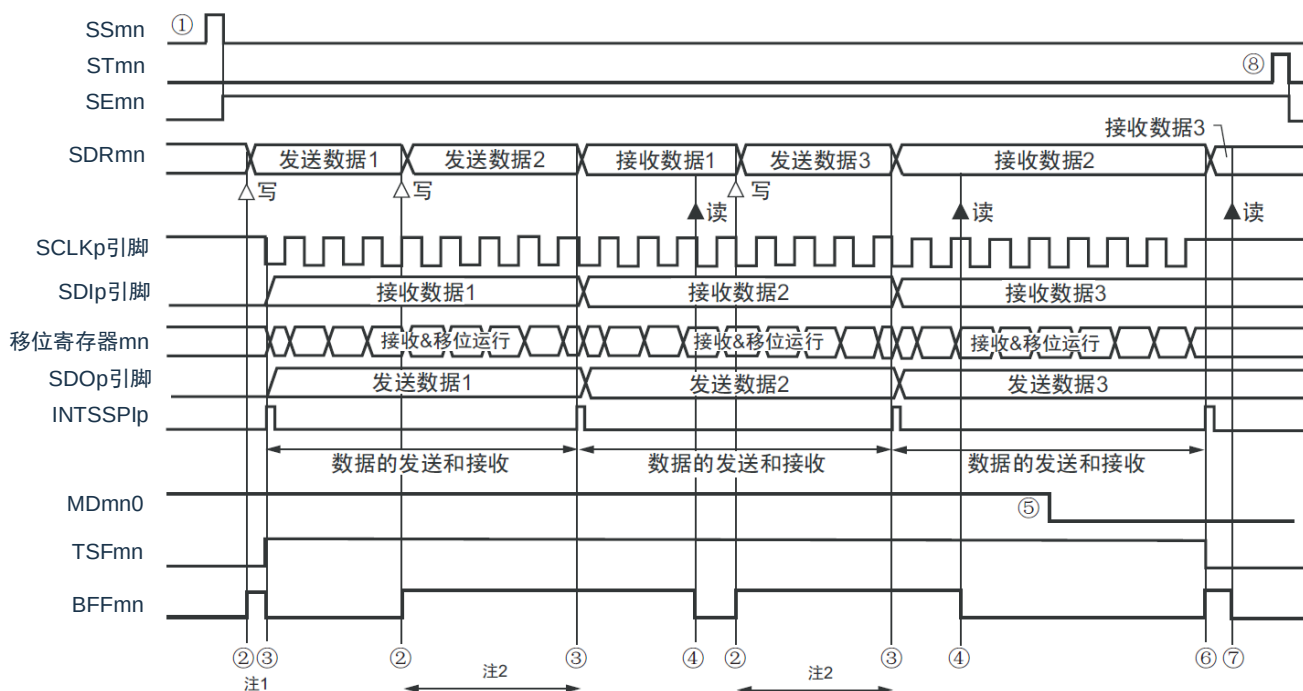
备注：m：单元号（m=0）；n：通道号（n=0、1）；p：SSPI号（p=00）。

图12-25: 主控发送和接收（单次发送和接收模式）的流程图



(4) 处理流程（连续发送和接收模式）

图12-26：主控发送和接收（连续发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注1：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注：

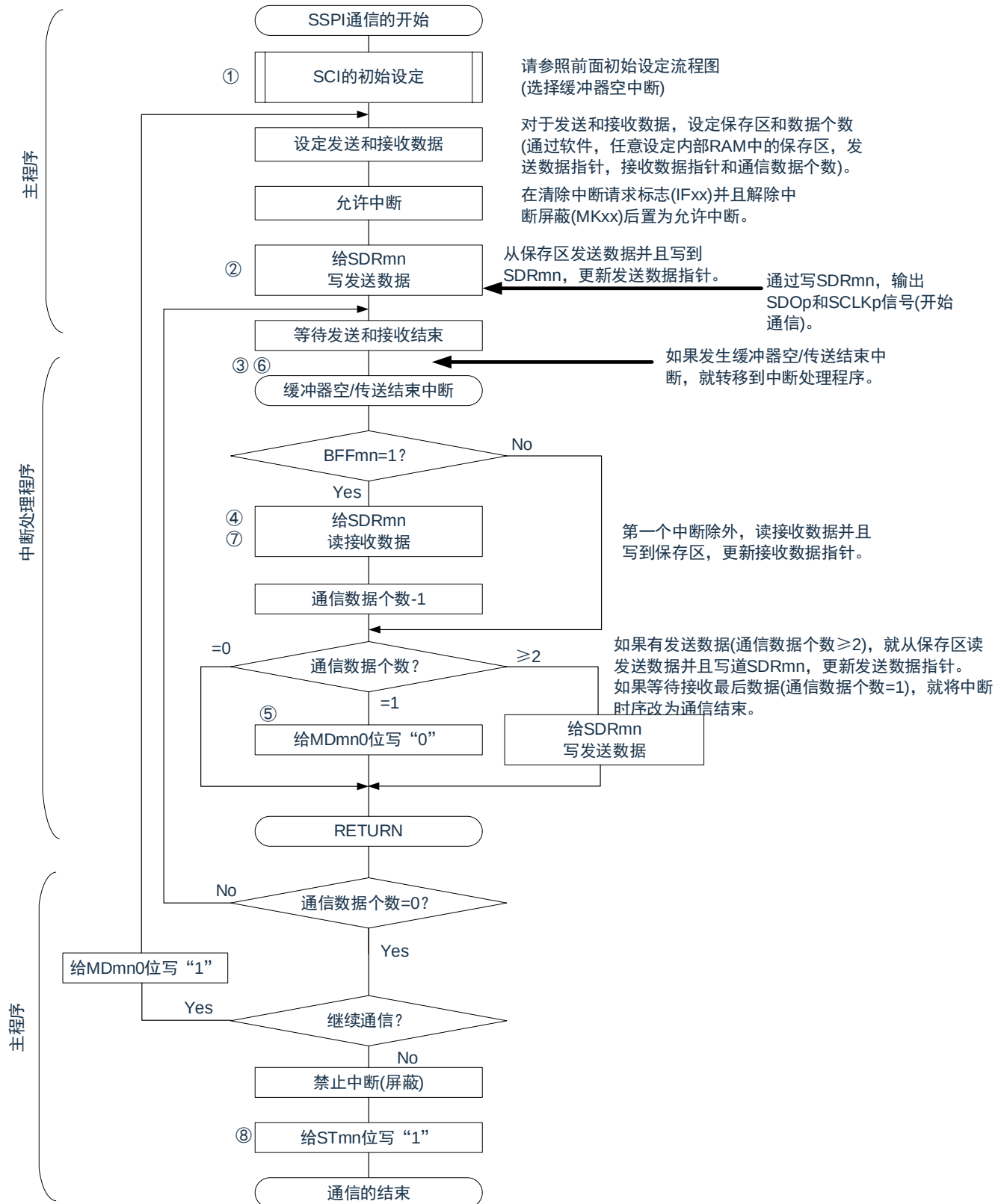
图中的①~⑧对应“图 12-27 主控发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）。

图12-27：主控发送和接收（连续发送和接收模式）的流程图



备注：图中的①~⑧对应“图12-26 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

12.5.4 从属发送

从属发送是指在从其他设备输入传送时钟的状态下微控制器将数据发送到其他设备的运行。

表12-25：从属发送

3线串行I/O	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDO00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0：在串行时钟开始运行时，开始数据输出。 DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0：正相 CKPmn=1：反相
数据方向	MSB优先或者LSB优先

注1：因为在内部对 SCLKOI00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为6M [Hz]；

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用；

备注：

F_{MCK}：对象通道的运行时钟频率；

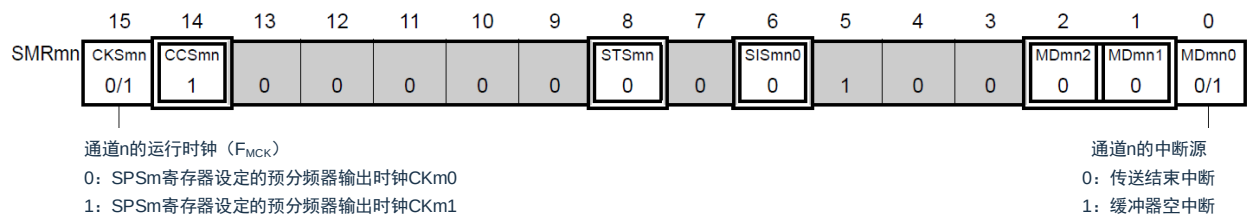
m：单元号（m=0）；

n：通道号（n=0、1）。

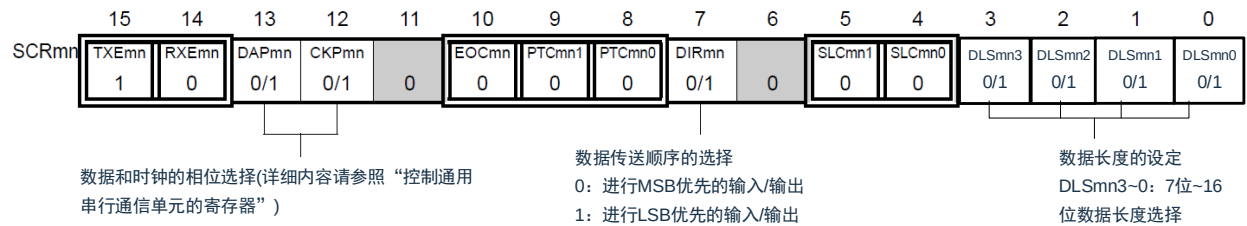
(1) 寄存器的设定

图12-28：3线串行 I/O（SSPI00）从属发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

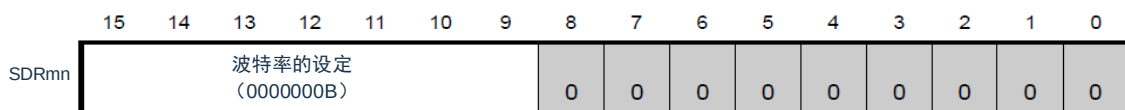


(b) 串行通信运行设定寄存器mn(SCRmn)

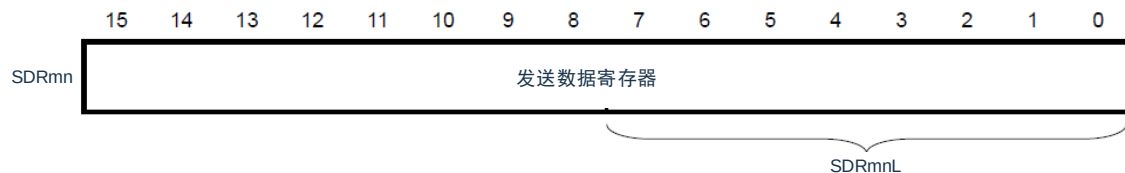


(c) 串行数据寄存器mn(SDRmn)

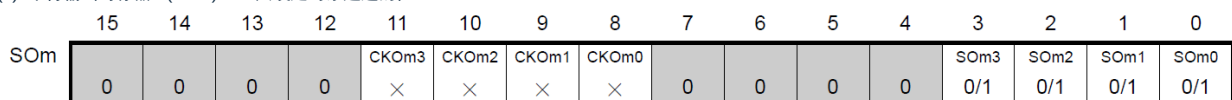
(1) 运行停止 (SEmn=0) 时



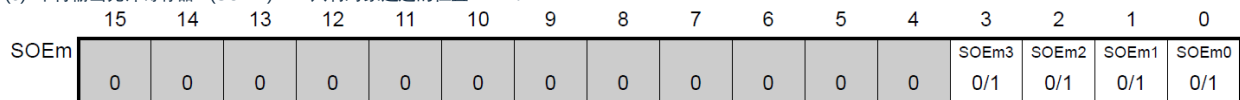
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



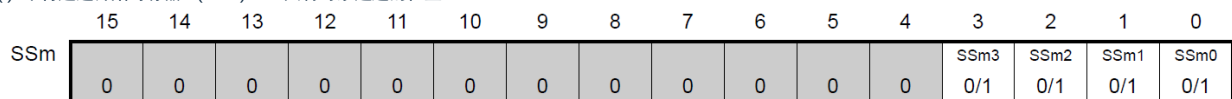
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00);

□ : 在 SSPI 主控接收模式中为固定设定; ■ : 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值);

0/1: 根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-29：从属发送的初始设定步骤



图12-30：从属发送的中止步骤

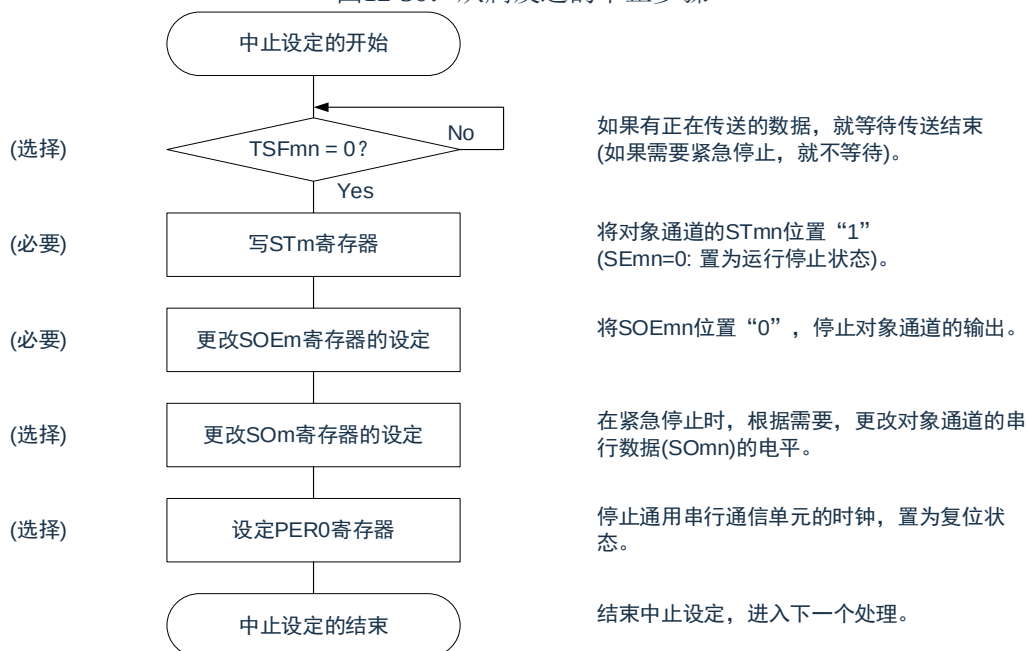
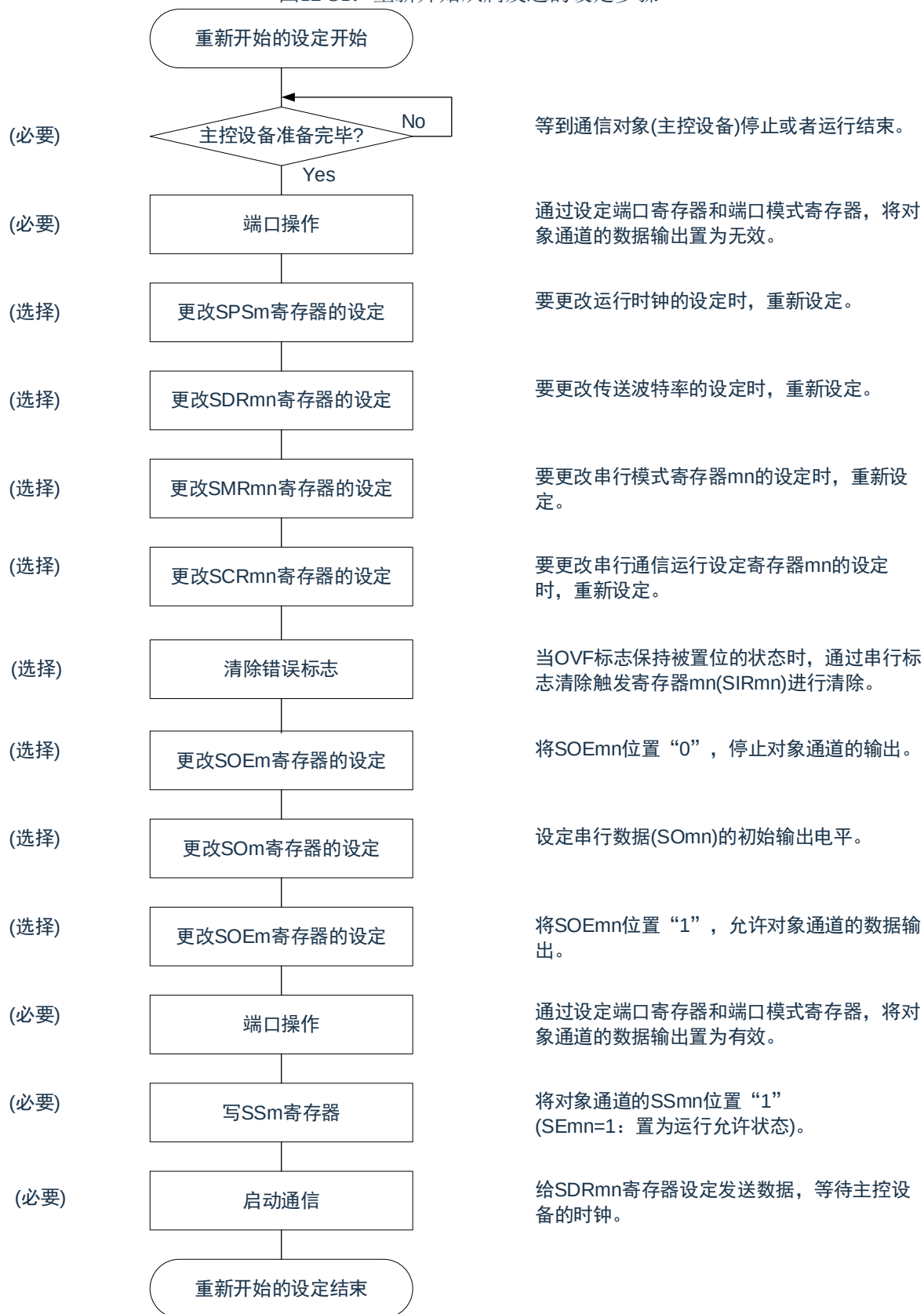


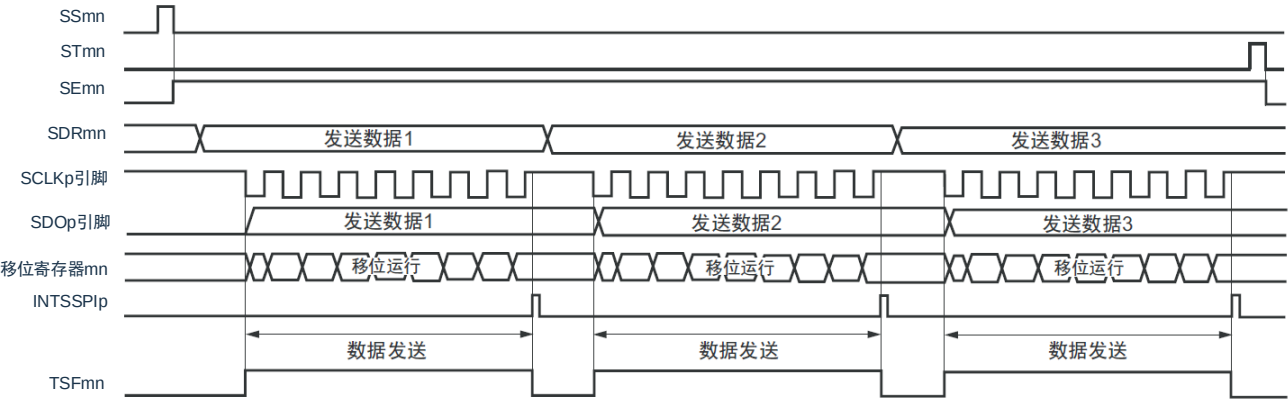
图12-31：重新开始从属发送的设定步骤



注意：如果在中止设定中改写 PER0来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

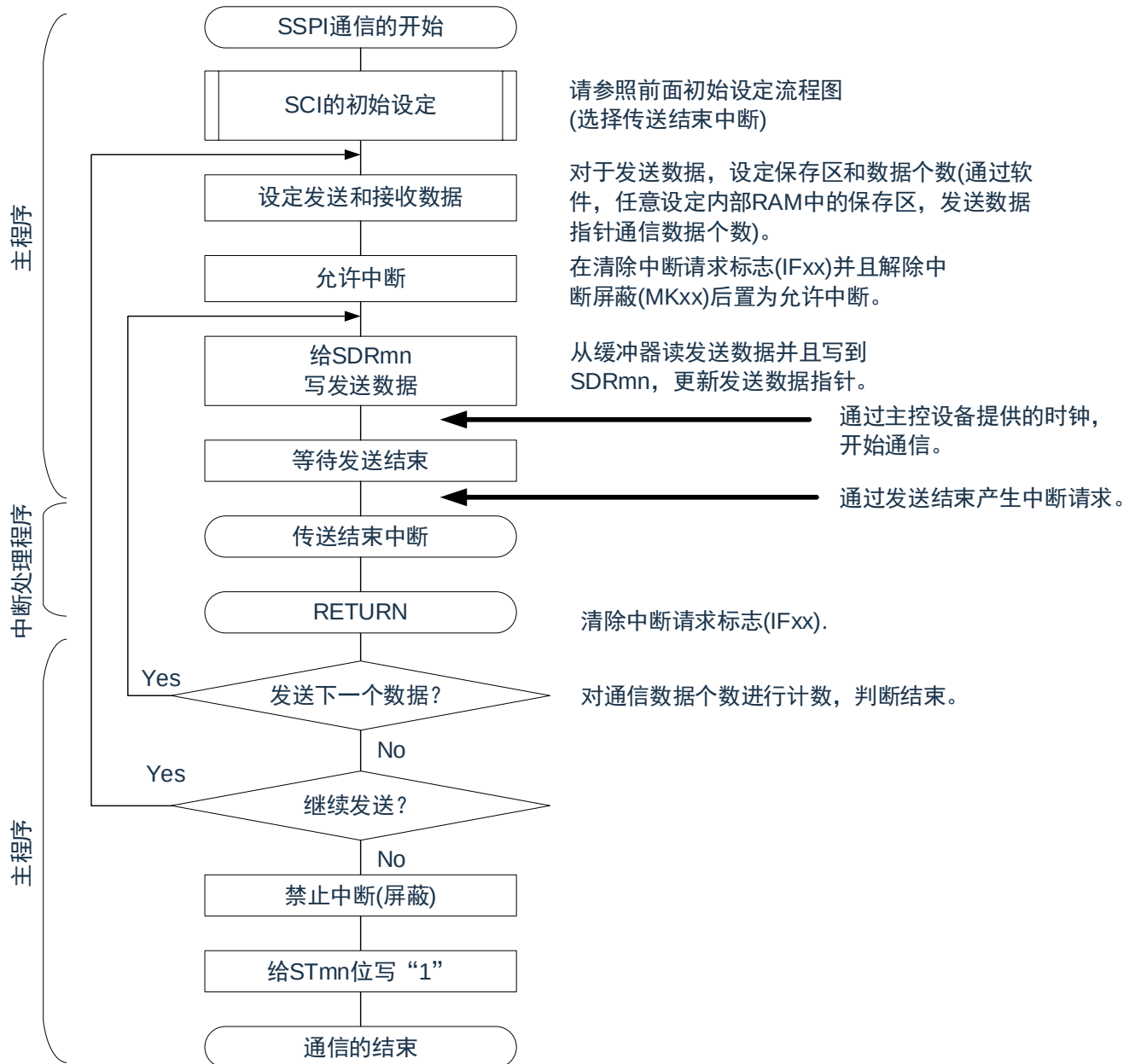
(3)处理流程（单次发送模式）

图12-32：从属发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



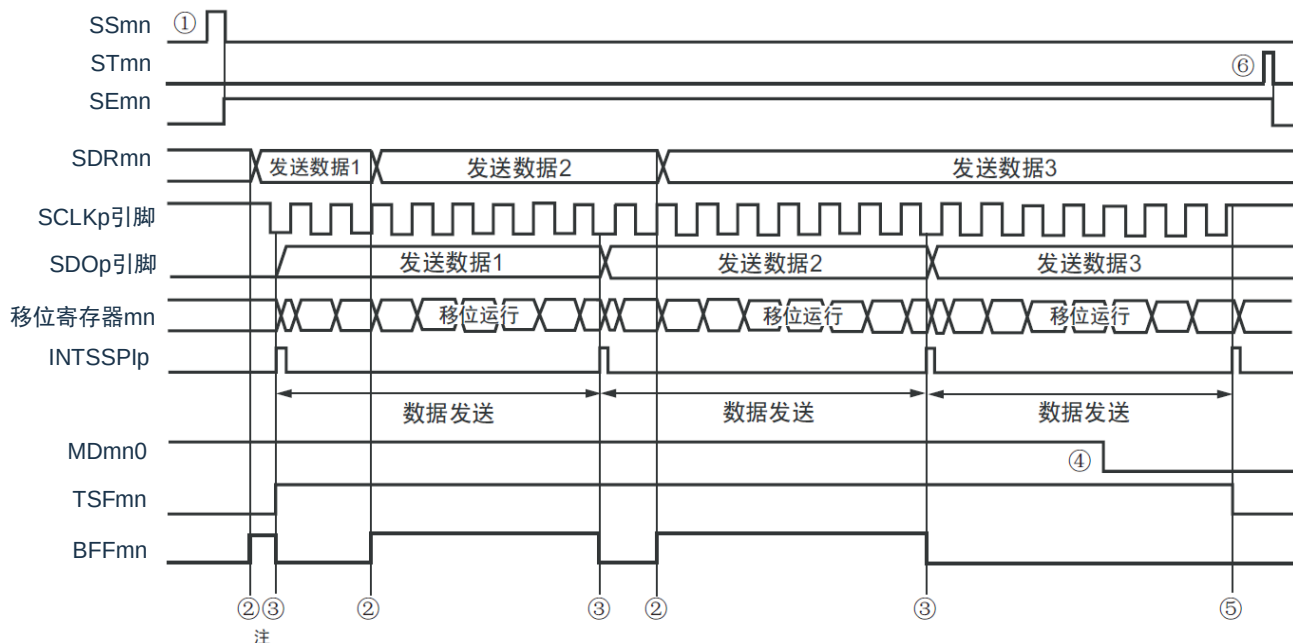
备注： m： 单元号（m=0）；
n： 通道号（n=0、1）；
p： SSPI 号（p=00）。

图12-33: 从属发送（单次发送模式）的流程图



(4)处理流程（连续发送模式）

图12-34：从属发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

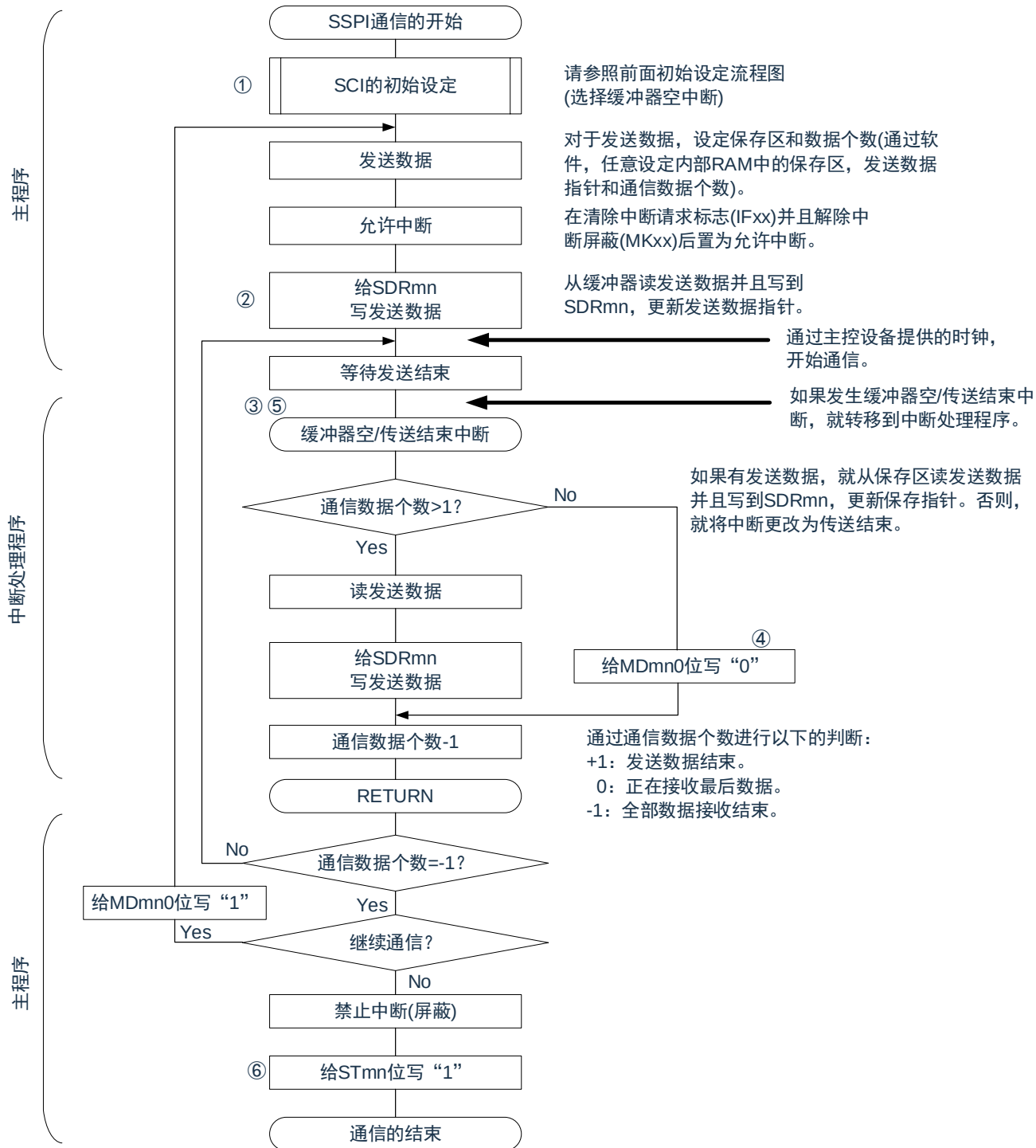
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注：m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）。

图12-35：从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“

图12-34 从属发送（连续发送模式）的时序图”中的①~⑥。

12.5.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

表12-26：从属接收

3线串行I/O	SSPI00
对象通道	SCIO的通道0
使用的引脚	SCLKOI00、SDI00
中断	INTSSPI00
	只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6MHz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。
	DAPmn=0: 在串行时钟开始运行时，开始数据输出。 DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。
	CKPmn=0: 正相 CKPmn=1: 反相
数据方向	MSB优先或者LSB优先

注1：因为在内部对 SCLKOI00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为6M[Hz]；

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用；

F_{MCK}: 对象通道的运行时钟频率；

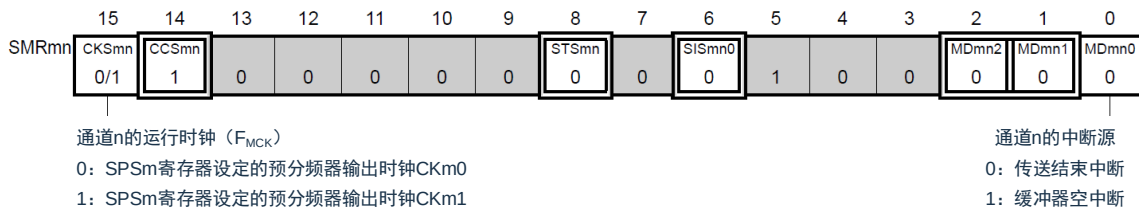
m: 单元号（m=0）；

n: 通道号（n=0、1）。

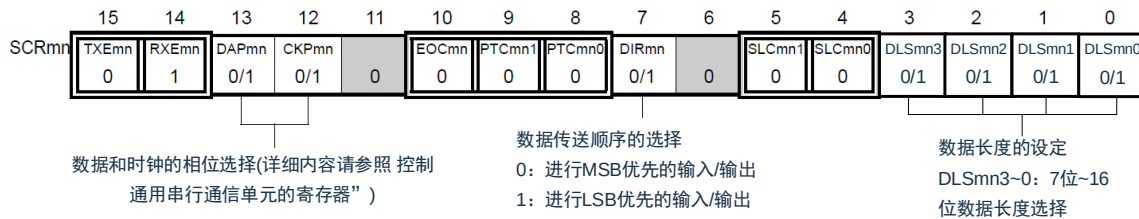
(1) 寄存器的设定

图12-36：3线串行 I/O（SSPI00）从属接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

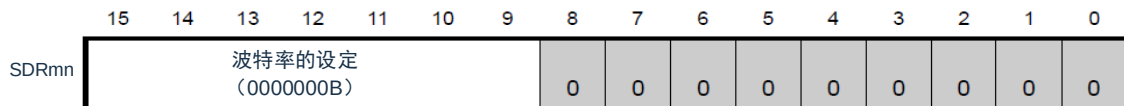


(b) 串行通信运行设定寄存器mn(SCRmn)

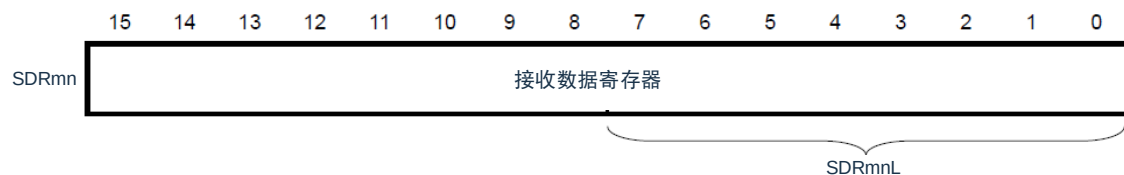


(c) 串行数据寄存器mn(SDRmn)

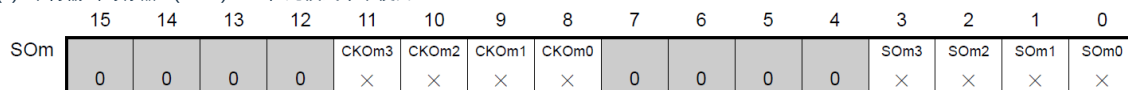
(1) 运行停止 (SEmn=0) 时



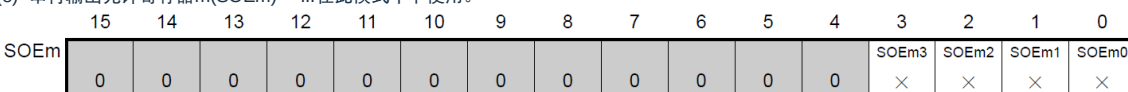
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



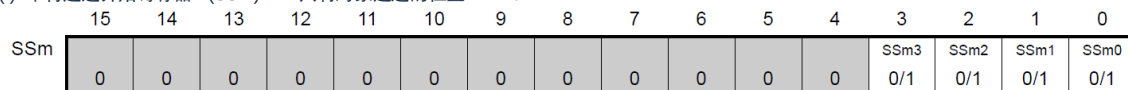
(d) 串行输出寄存器m(SOm) ……在此模式中不使用。



(e) 串行输出允许寄存器m(SOEm) ……在此模式中不使用。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00);

□ : 在 SSPI 主控接收模式中为固定设定; ■ : 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值);

0/1: 根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-37：从属接收的初始设定步骤



图12-38：从属接收的中止步骤

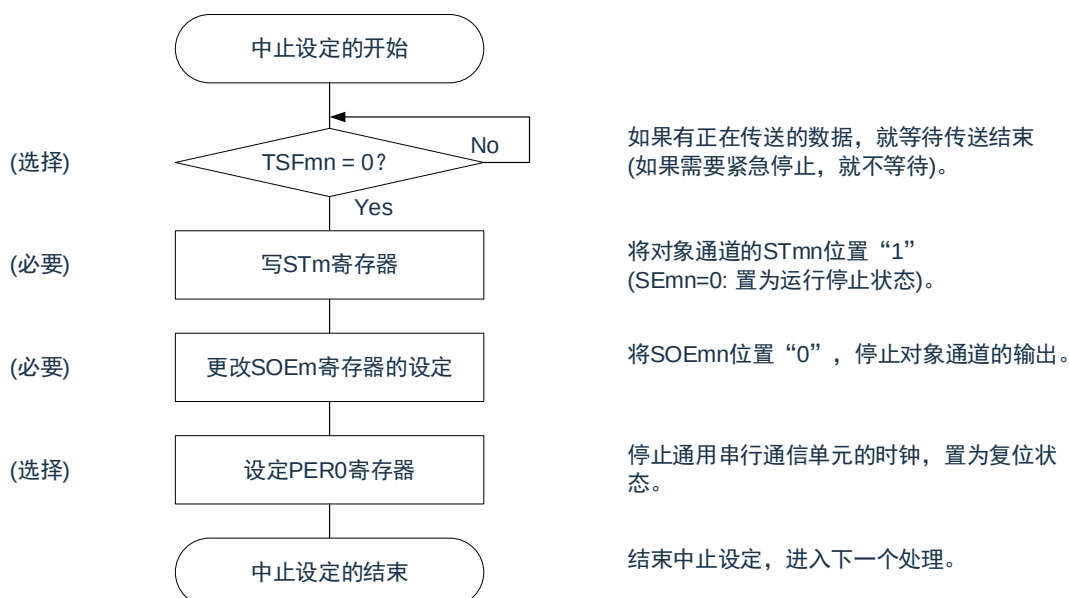


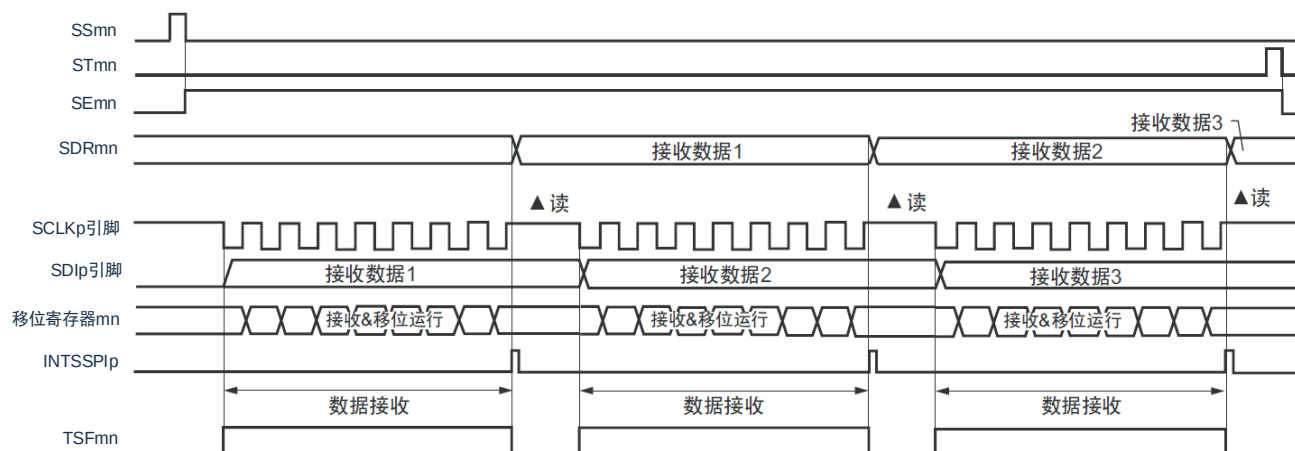
图12-39：重新开始从属接收的设定步骤



注意：如果在中止设定中改写 PER0来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

(3) 处理流程（单次接收模式）

图12-40：从属接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）

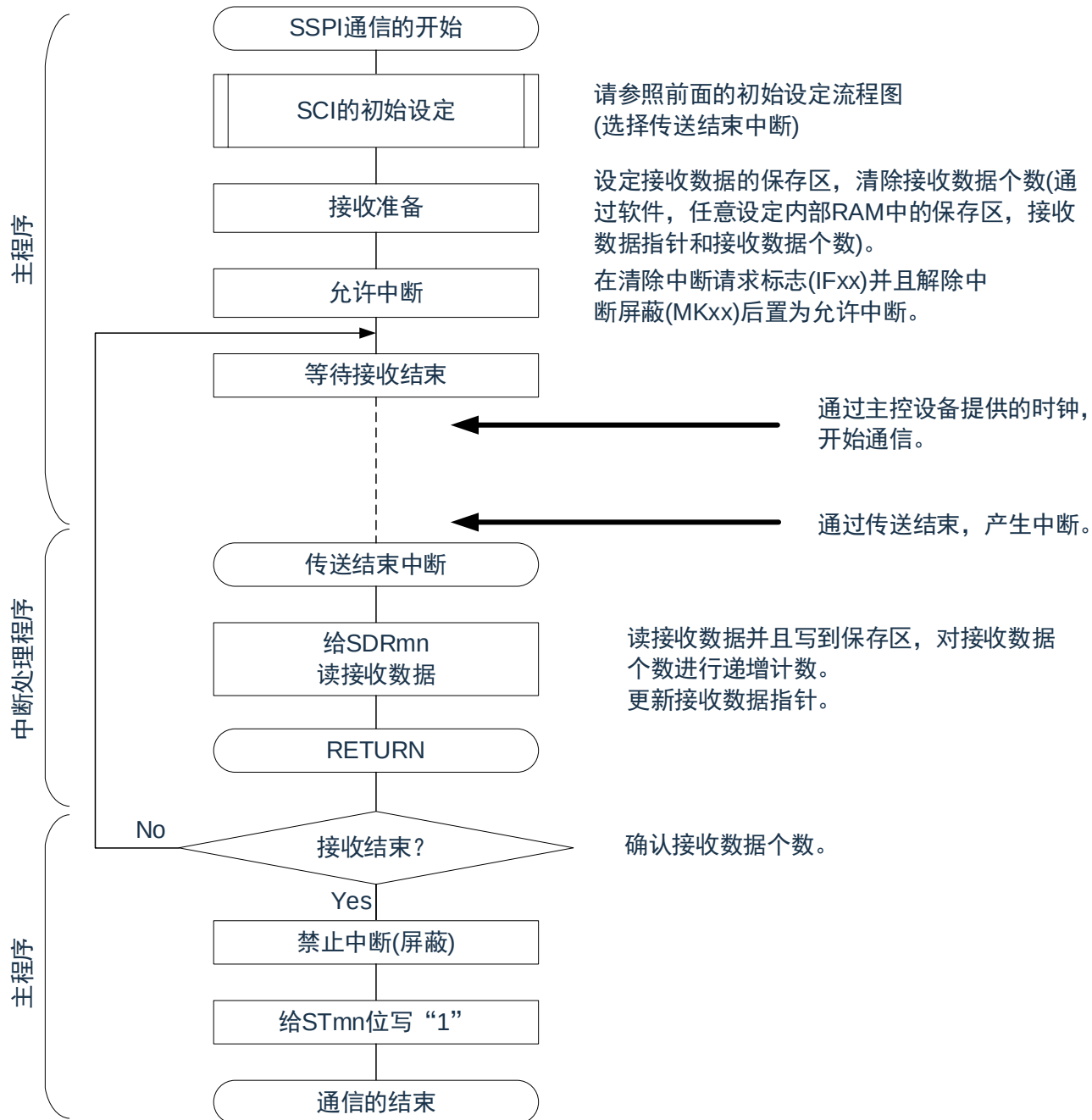


备注：m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）。

图12-41：从属接收（单次接收模式）的流程图



12.5.6 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品微控制器和其他设备进行数据发送和接收的运行。

表12-27：从属的发送和接收

3线串行I/O	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00、SDO00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。
	DAPmn=0：在串行时钟开始运行时，开始数据输入/输出。 DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输入/输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。
	CKPmn=0：正相 CKPmn=1：反相
数据方向	MSB优先或者LSB优先

注1：因为在内部对 SCLKOI00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率6M[Hz]；

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用；

备注：

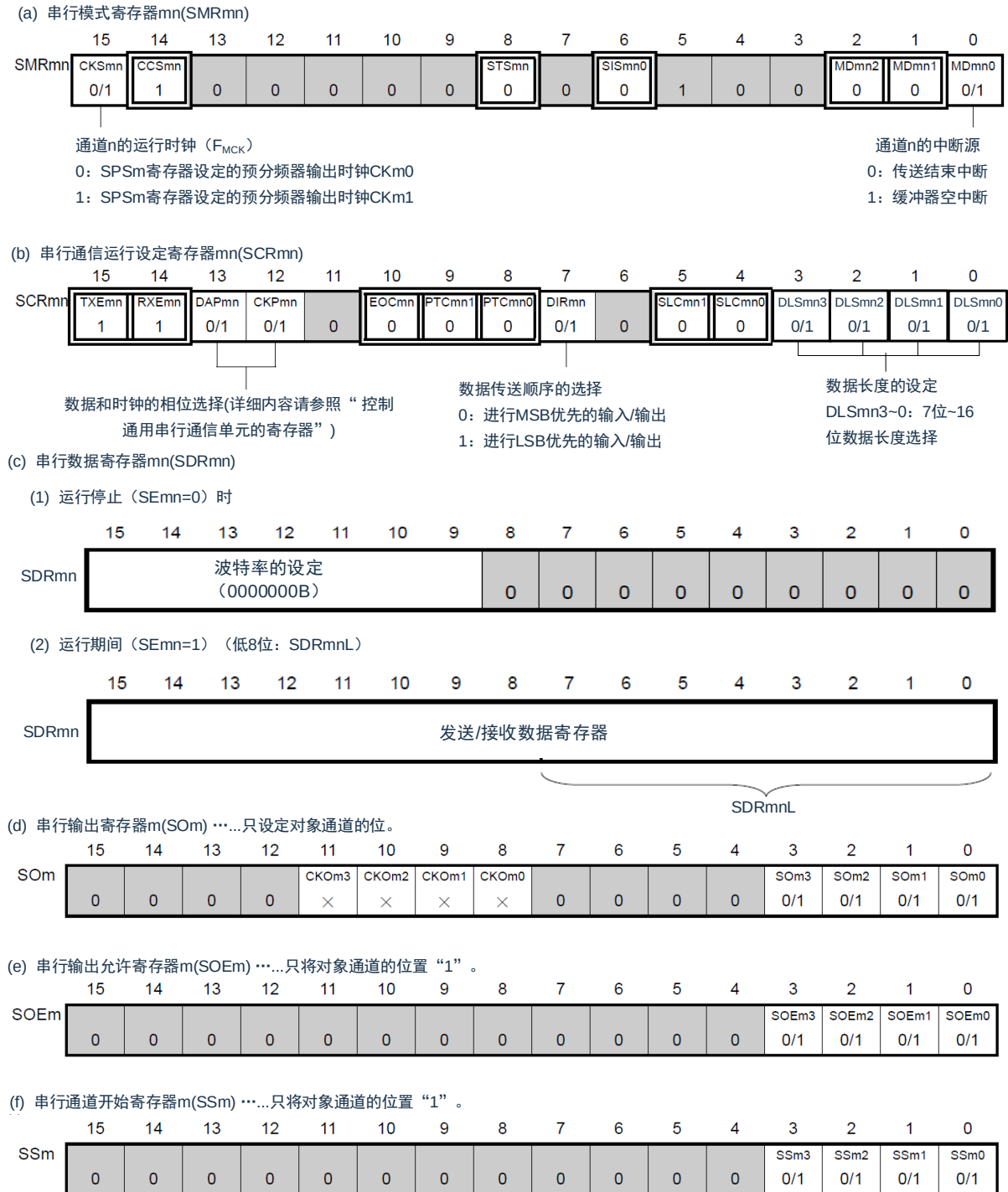
F_{MCK}：对象通道的运行时钟频率；

m：单元号（m=0）；

n：通道号（n=0、1）。

(1) 寄存器的设定

图12-42：3线串行 I/O（SSPI00）从属发送和接收时的寄存器设定内容例子



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注： m: 单元号 (m=0)；

n: 通道号 (n=0、1)；

p: SSPI 号 (p=00)；

□：在 SSPI 主控接收模式中为固定设定； ■：不能设定（设定初始值）；

×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）；

0/1: 根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-43: 从属发送和接收的初始设定步骤



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

图12-44：从属发送和接收的中止步骤

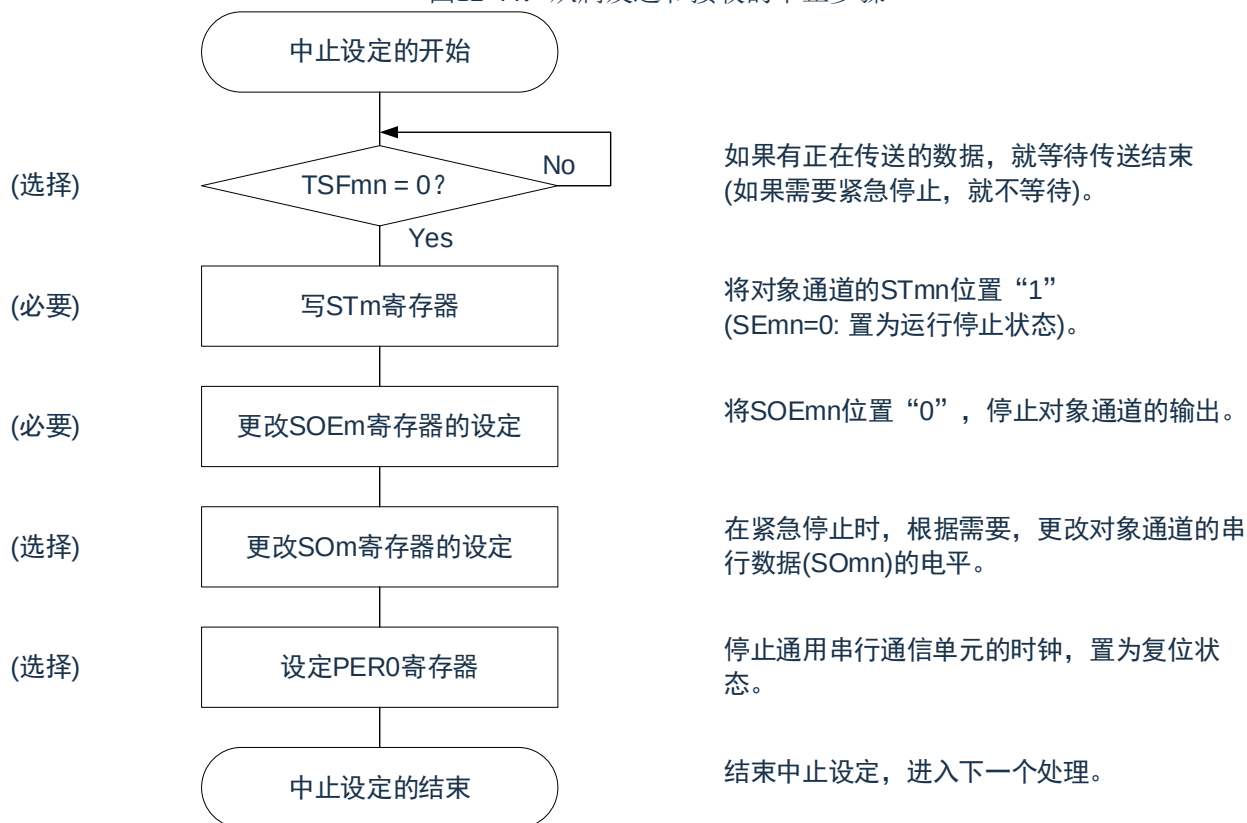
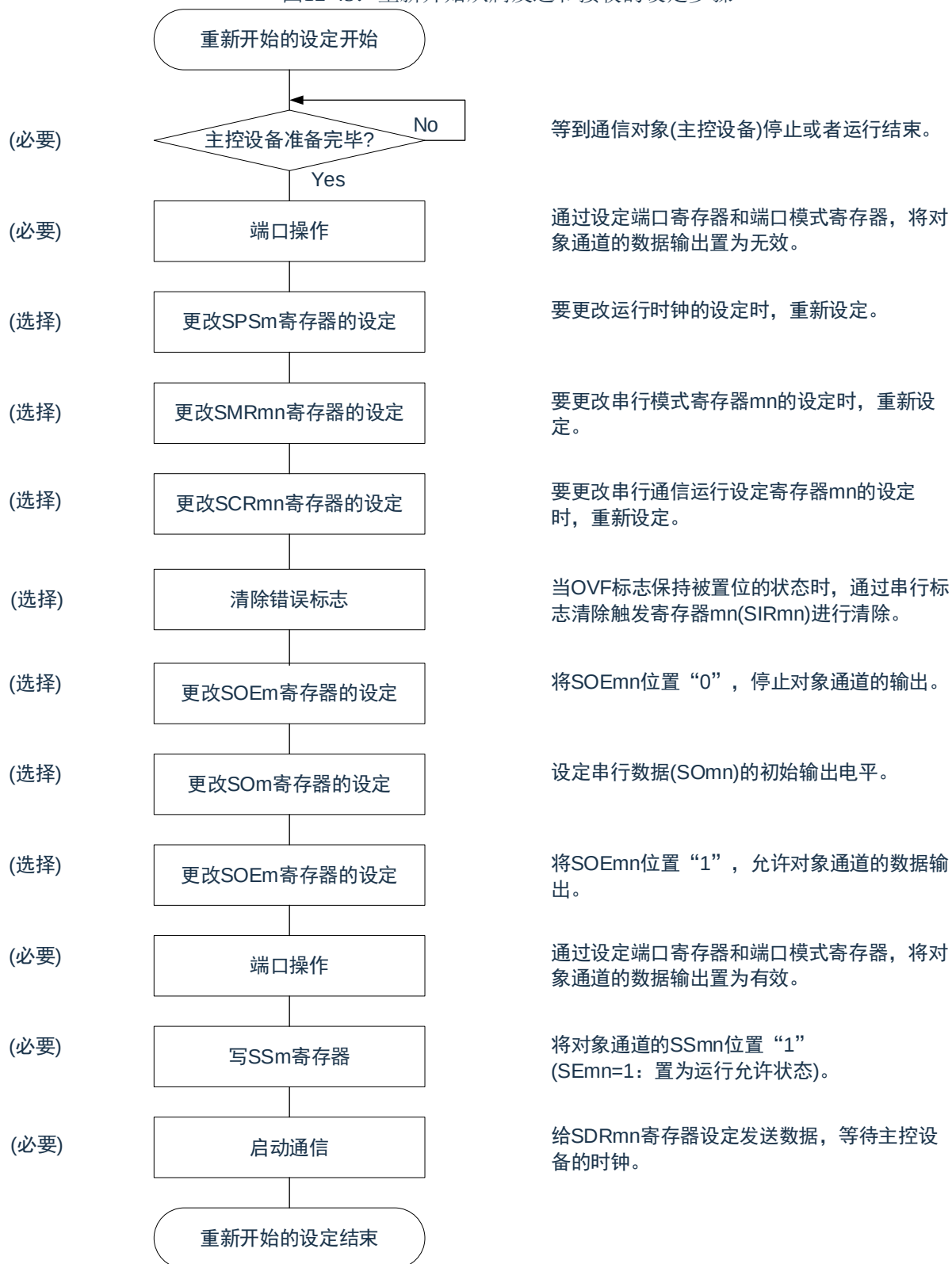


图12-45：重新开始从属发送和接收的设定步骤

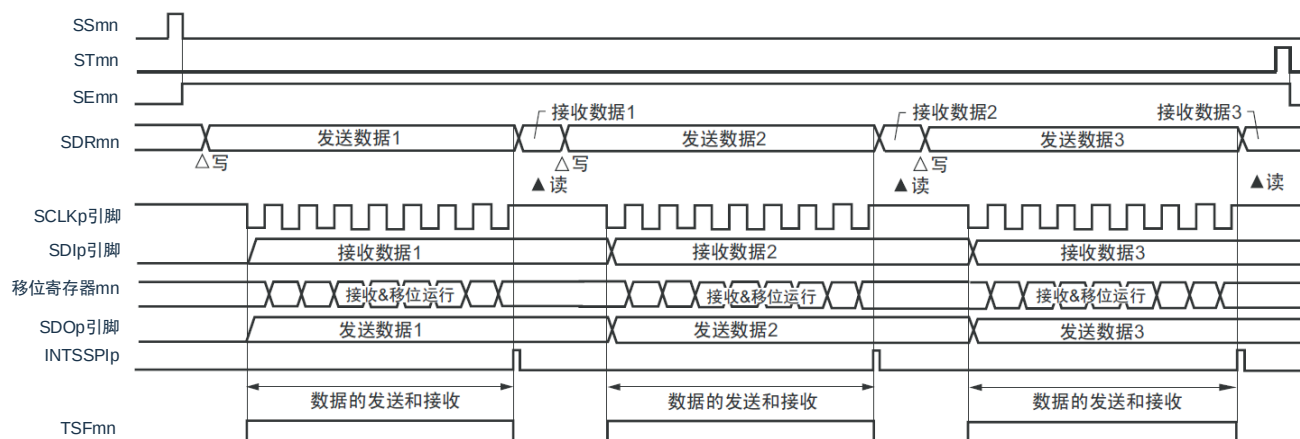


注意：

在主机设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主机设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

(3) 处理流程（单次发送和接收模式）

图12-46：从属发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）

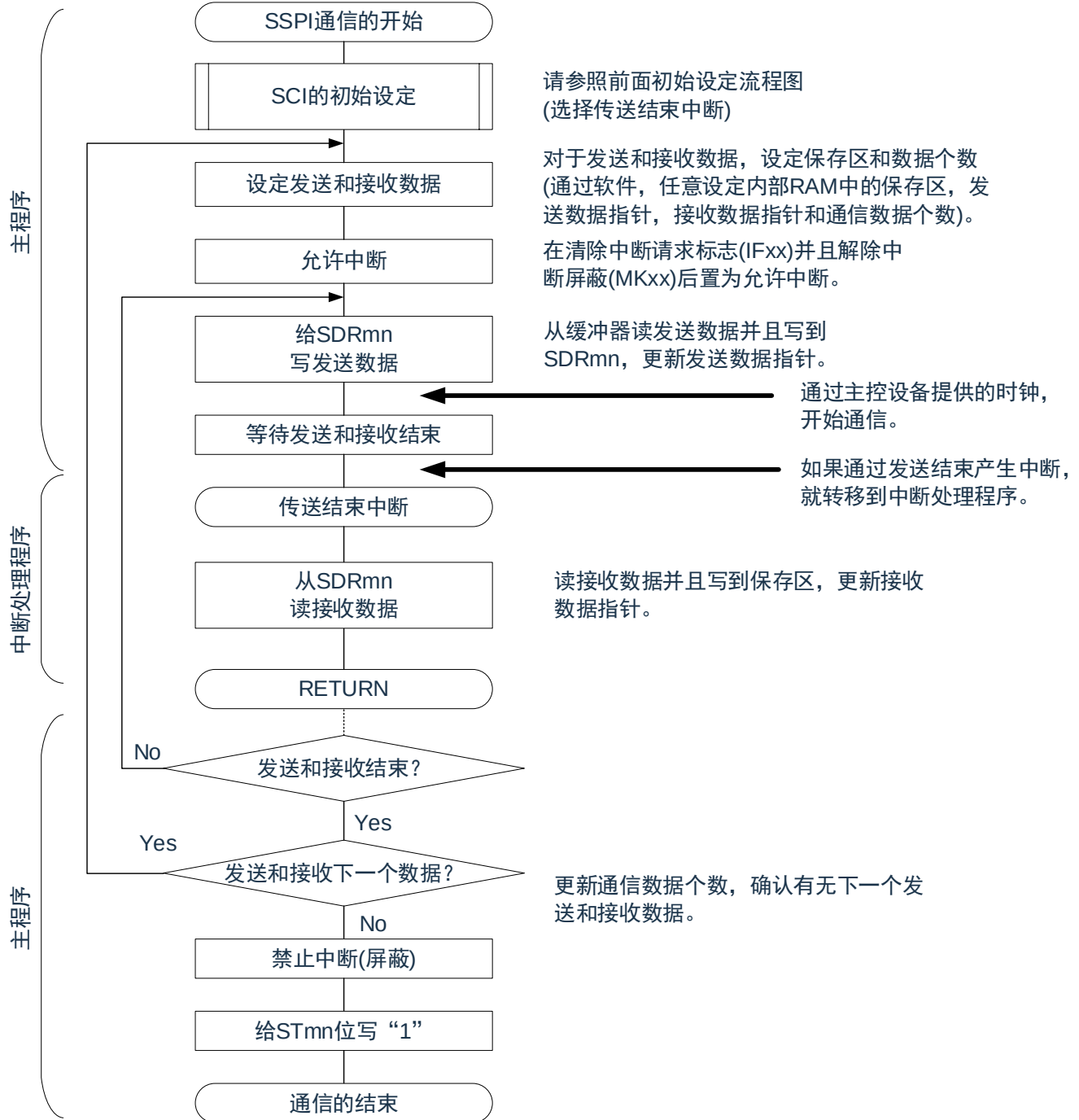


备注：m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）。

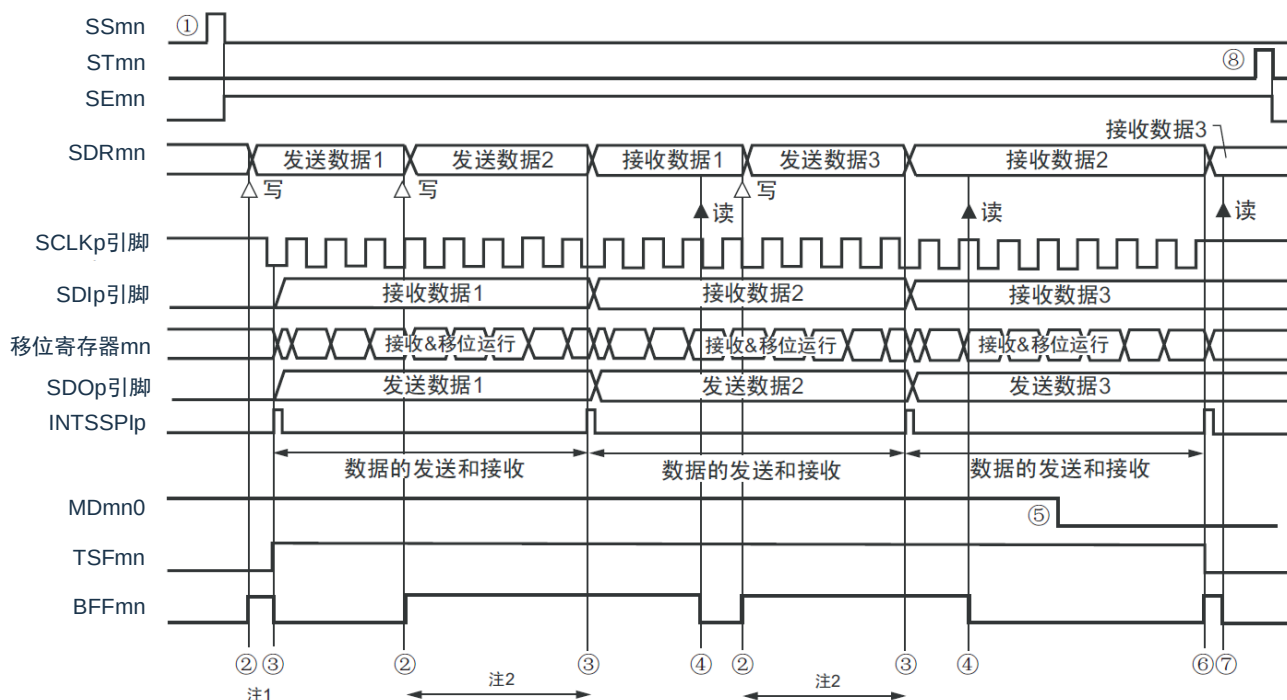
图12-47：从属发送和接收（单次发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

(4)处理流程（连续发送和接收模式）

图12-48：从属发送和接收（连续发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注1：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据；

注2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行；

注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写；

备注：

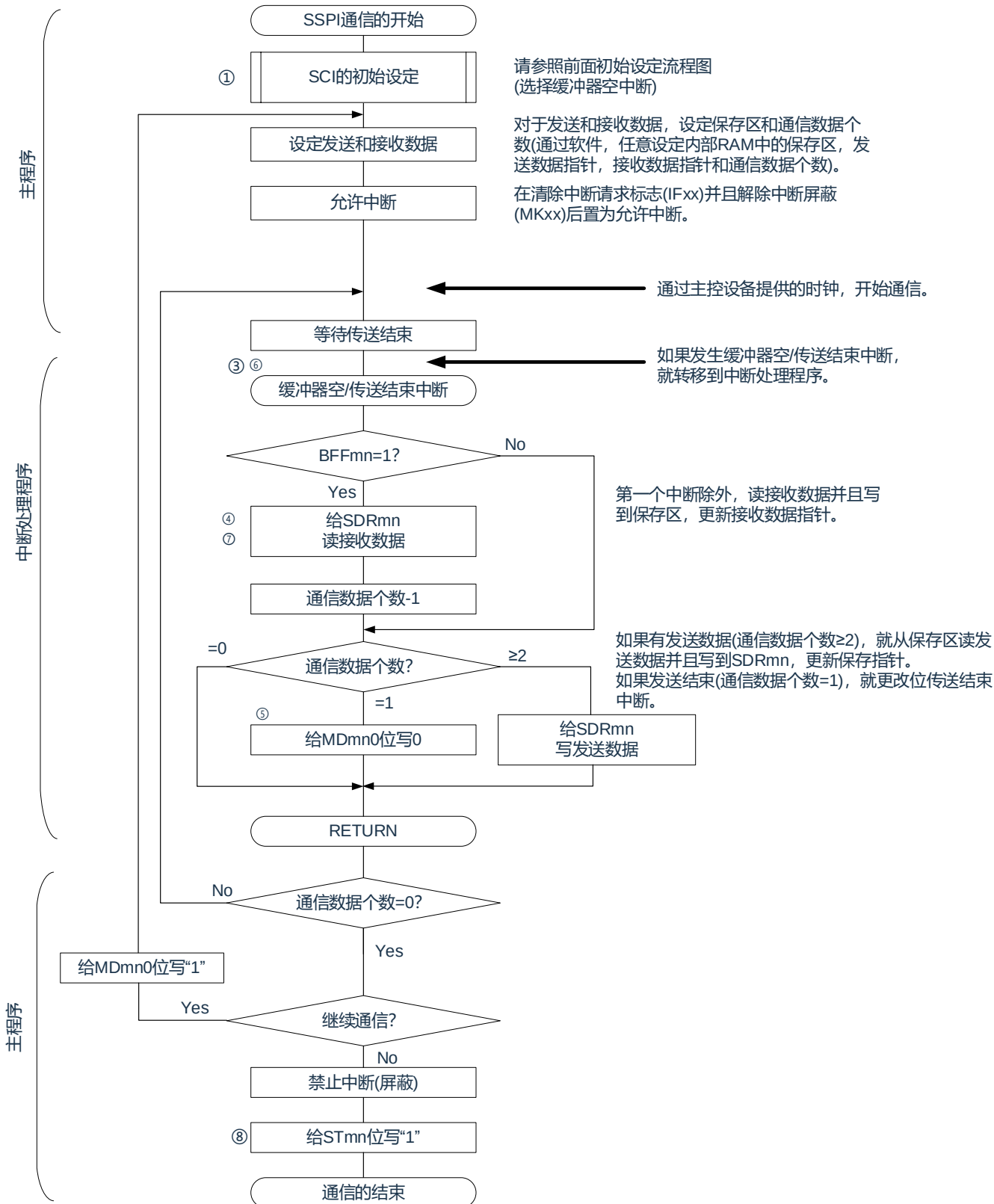
图中的①~⑧对应“12-49 从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧；

m：单元号（m=0）；

n：通道号（n=0、1）；

p：SSPI 号（p=00）。

图12-49：从属发送和接收（连续发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：图中的①~⑧对应“12-48从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

12.5.7 传送时钟频率的计算

3线串行 I/O（SSPI00）通信的传送时钟频率能用以下计算式进行计算：

主控设备

$$(\text{传送时钟频率}) = \{ \text{对象通道的运行时钟 (F}_{\text{MCK}}) \text{ 频率} \} / (\text{SDRmn}[15:9] + 1) / 2 [\text{Hz}]$$

从属设备

$$(\text{传送时钟频率}) = \{ \text{主控设备提供的串行时钟 (SCLK) 频率} \}^{\text{注}} [\text{Hz}]$$

备注：

容许的最大传送时钟频率为6MHz；

因为 SDRmn[15:9]的值为串行数据寄存器 mn（SDRmn）的 bit15~9的值（0000000B~1111111B），所以为0~127；

运行时钟（F_{MCK}）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn）。

表12-28：线串行 I/O 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (F _{MCK}) ^注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz 运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	5.86KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	2.93KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.46KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	732Hz
1	0	0	0	0	X	X	X	X	F _{CLK}	24MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	12MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	6MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	3MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	1.5MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	0.75MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	375KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	187.5KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	93.75KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	46.875KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	23.44KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	11.72KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	5.86KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	2.93KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.46KHz
	1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	732Hz

注：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改；

X：忽略；

m：单元号（m=0）；

n：通道号（n=0、1）。

12.5.8 在 3 线串行 I/O（SSPI00）通信过程中发生错误时的处理步骤

在 3 线串行 I/O（SSPI00）通信过程中发生错误时的处理步骤如表 12-29 所示：

表 12-29：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn（SDRmn）。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn（SDIRmn）写“1”。	→ 清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0）；
n：通道号（n=0、1）。

12.6 从属选择输入功能的时钟同步串行通信的运行

SCI0的通道0是支持从属选择输入功能的时钟同步串行通信的通道。

[数据的发送和接收]

- 7位、8位或9位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率注从属通信：Max.6MHz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

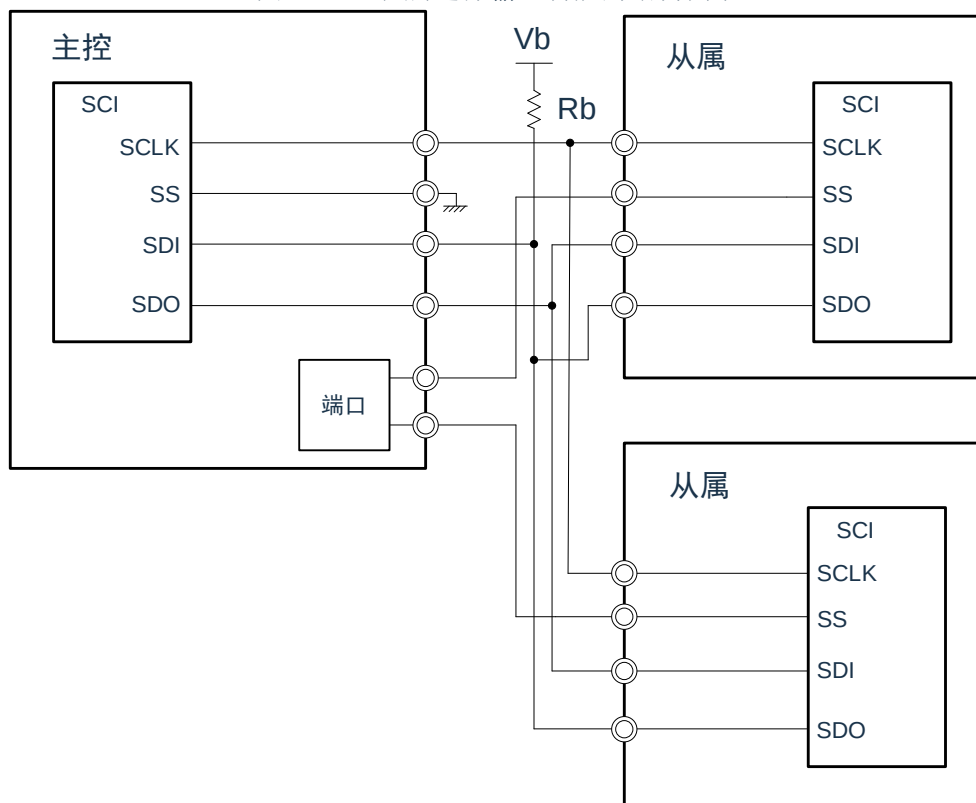
从属选择输入功能有以下3种通信运行：

- 从属发送（参照12.6.1）
- 从属接收（参照12.6.2）
- 从属的发送和接收（参照12.6.3）

注意：必须在满足 SCLK 周期时间（ T_{KCY} ）特性的范围内使用。详细内容请参照数据手册。

能通过使用从属选择输入功能，使1个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制 SDO 引脚的输出。当被选择为通信对象的从属设备时，SDO 引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SDO 引脚变为高电平输出，因此在连接多个从属设备的环境下需要将 SDO 引脚设定为 Nch-O.D 并且将该节点上拉。另外，即使输入主控设备的串行时钟也不进行发送和接收。

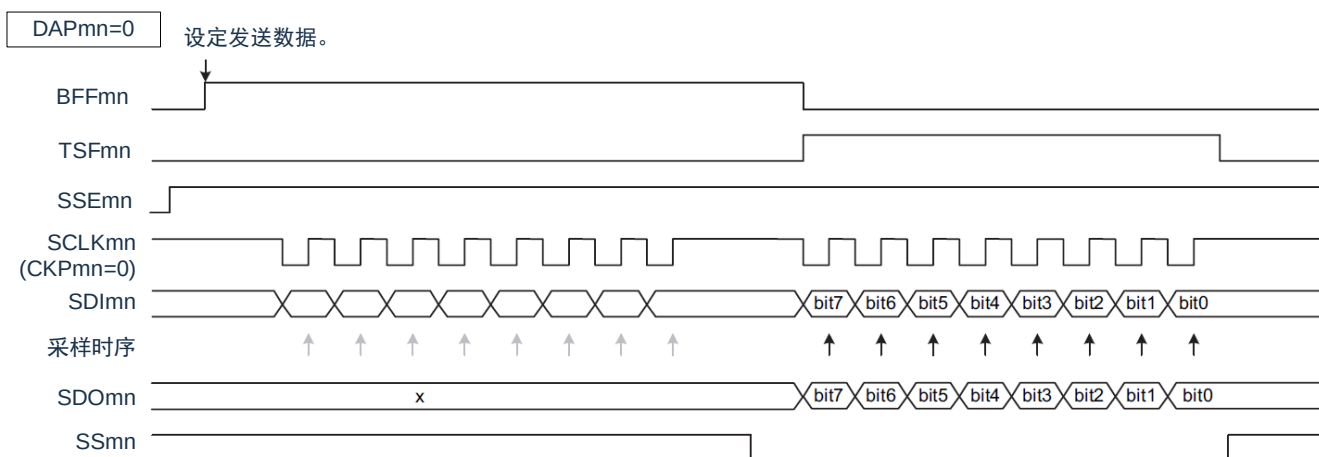
图12-50：从属选择输入功能的结构例子



注意：

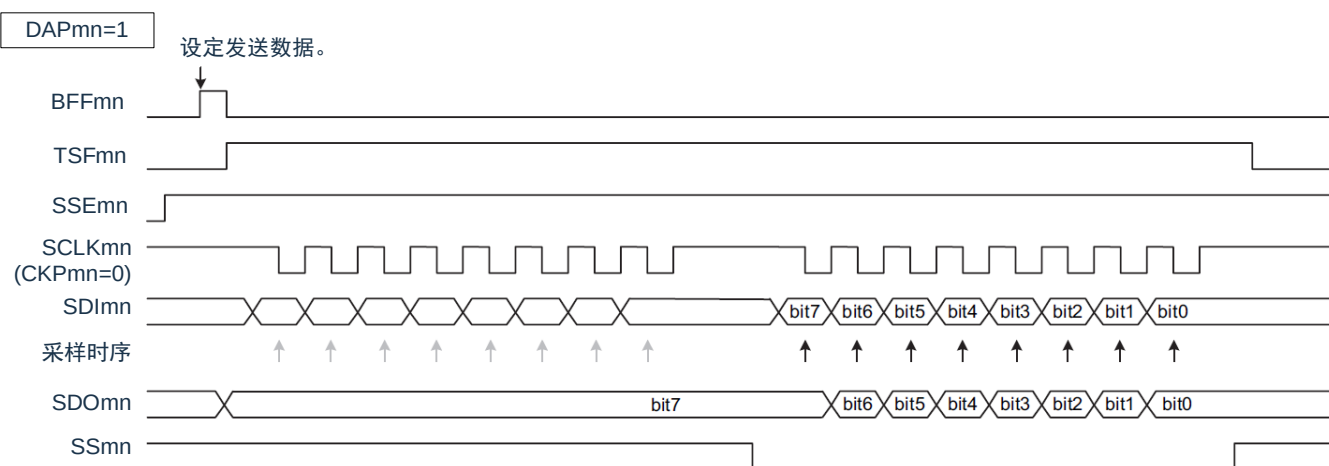
将 SDO00 引脚选择为 N 沟道漏极开路输出模式；
必须通过端口的操作输出从属选择信号。

图12-51：从属选择输入功能的时序图



在 SSmn 为高电平期间，即使在 SCKmn(串行时钟)的下降沿也不进行发送，而且也不进行与上升沿同步的接收数据的采样。

在 SSmn 为低电平期间，与串行时钟的下降沿同步输出数据(移位)并且与上升沿同步接收数据。



当 DAPmn 位为“1”时，如果在 SSmn 为高电平期间设定发送数据，就将最初的数据(bit7)提供给数据输出。但是，即使在 SCLKmn(串行时钟)的上升沿也不移位，而且也不进行与下降沿同步的接受数据的采样。如果 SSmn 变为低电平，就与下一个上升沿同步输出数据(移位)并且与下降沿同步接收数据。

备注：m：单元号（m=0）；

n：通道号（n=0）。

12.6.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下本产品将数据发送到其他设备的运行。

表12-30：从属发送

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0：在串行时钟开始运行时，开始数据输出。 DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0：正相 CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择功能的运行。

注1：因为在内部对 SCLK00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为6M[Hz]。

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

FMCK：对象通道的运行时钟频率

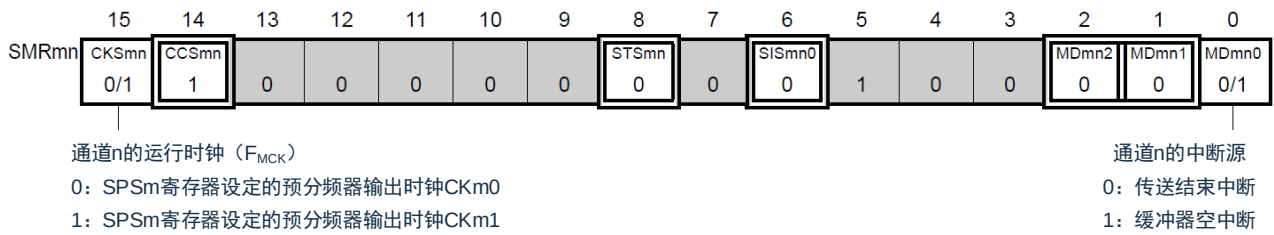
m：单元号（m=0）；

n：通道号（n=0）。

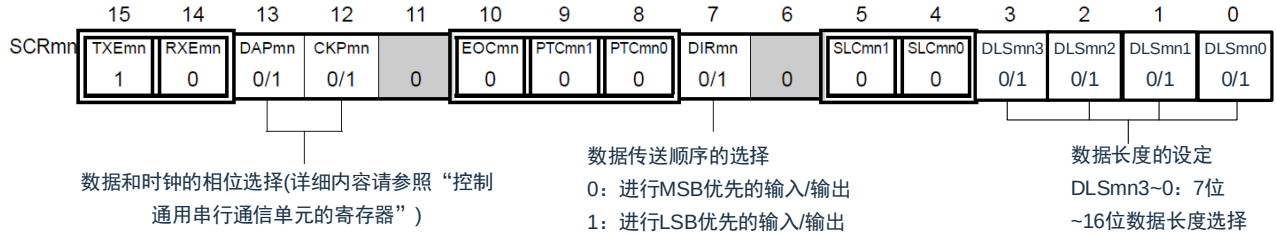
(1) 寄存器的设定

图12-52：从属选择输入功能（SSPI00）从属发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

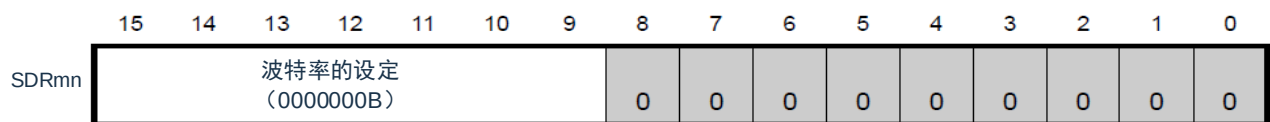


(b) 串行通信运行设定寄存器mn(SCRmn)

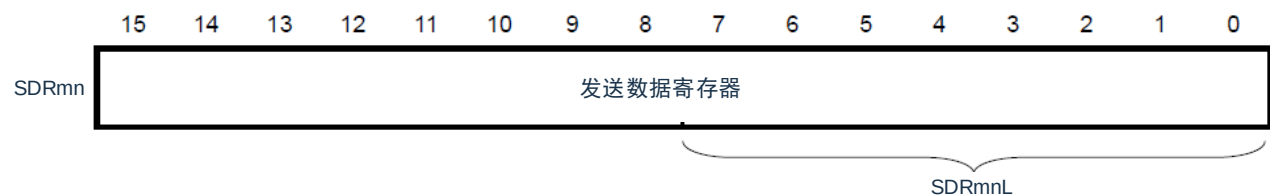


(c) 串行数据寄存器mn(SDRmn)

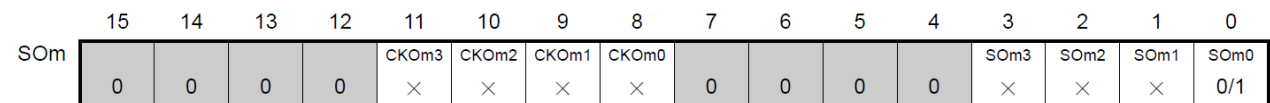
(1) 运行停止 (SEmn=0) 时



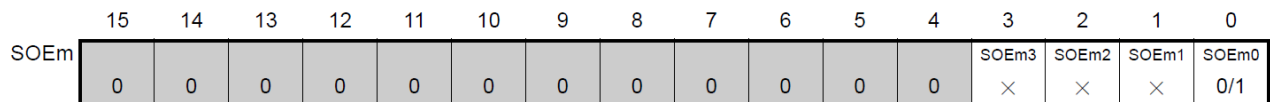
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0);

n: 通道号 (n=0);

p: SSPI 号 (p=00);

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值);

0/1: 根据用户的用途置“0”或者“1”。

图12-52：从属选择输入功能（SSPI00）从属发送时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

备注： m: 单元号（m=0）；

n: 通道号（n=0）；

p: SSPI 号（p=00）；

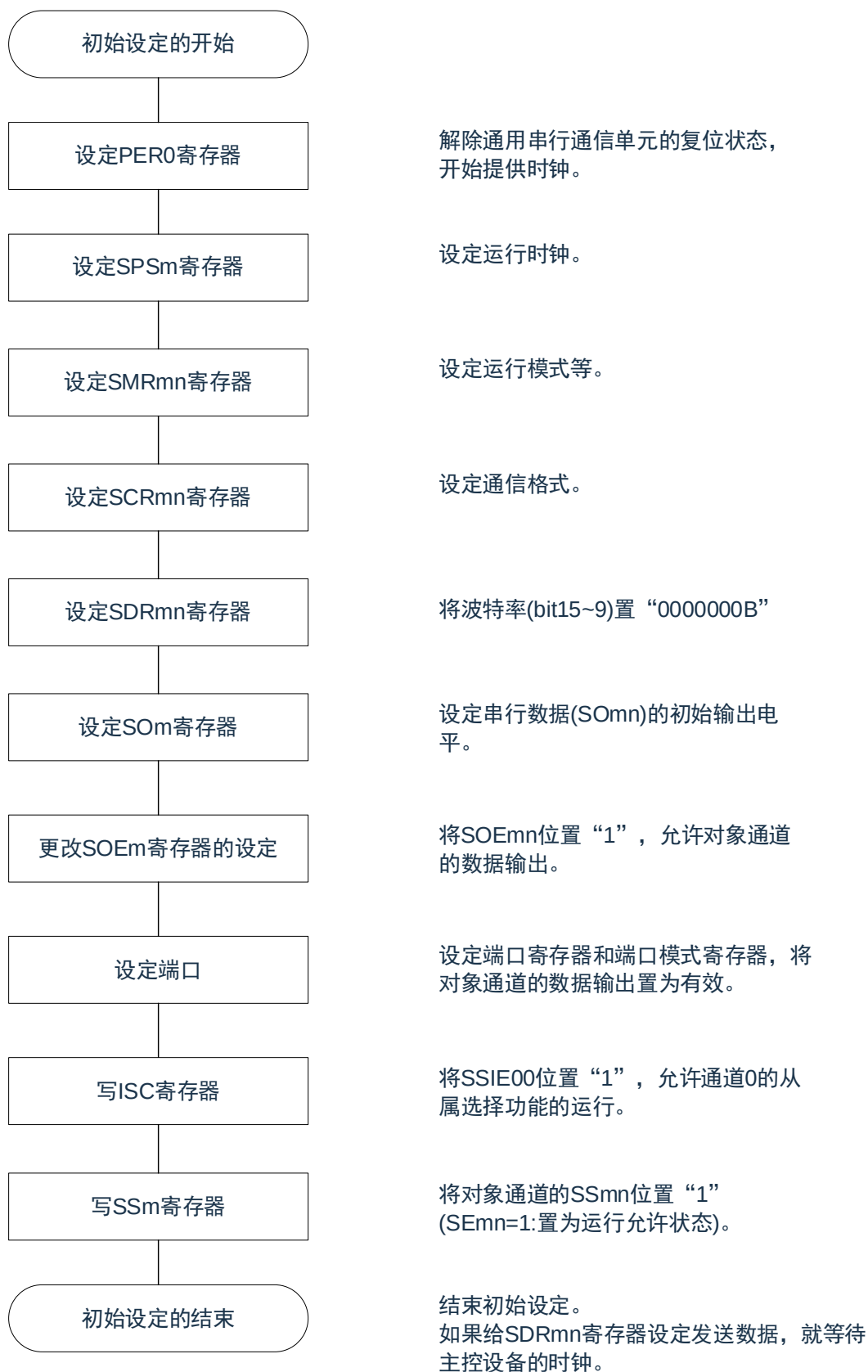
☐: 在 SSPI 主控接收模式中为固定设定； ☐: 不能设定（设定初始值）；

×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）；

0/1: 根据用户的用途置“0”或者“1”。

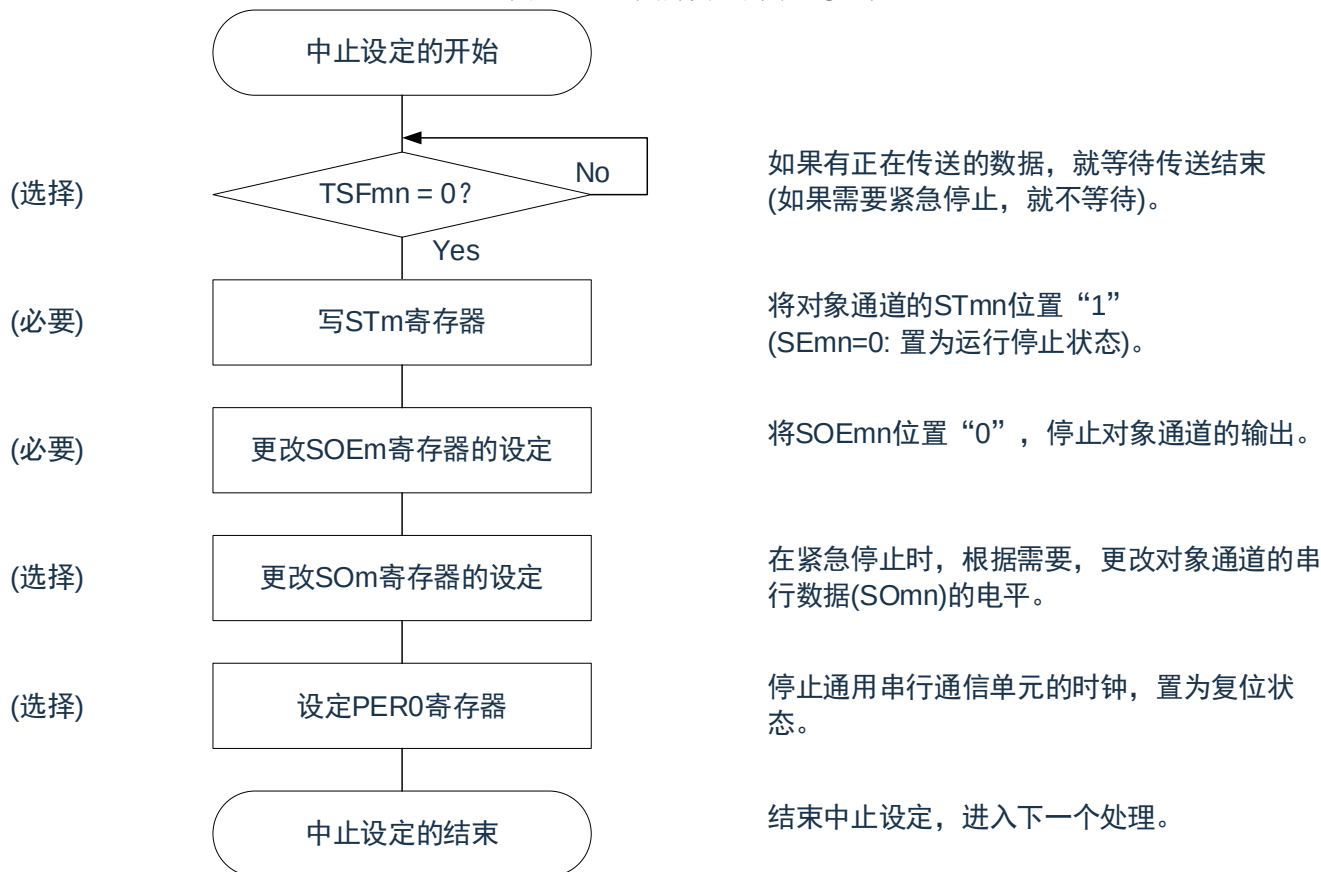
(2) 操作步骤

图12-53：从属发送的初始设定步骤



备注：m：单元号（m=0）；n：通道号（n=0）；p：SSPI号（p=00）。

图12-54：从属发送的中止步骤



注意：如果在中止设定中改写 PER0来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始设定；

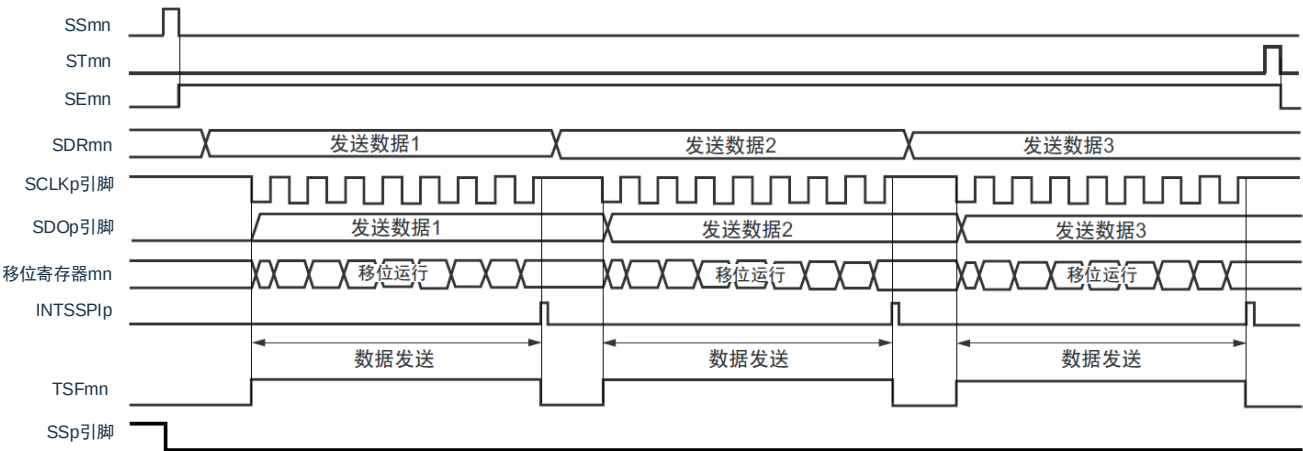
备注：m：单元号（m=0）； n：通道号（n=0）； p：SSPI 号（p=00）；

图12-55：重新开始从属发送的设定步骤



(3) 处理流程（单次发送模式）

图12-56：从属发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

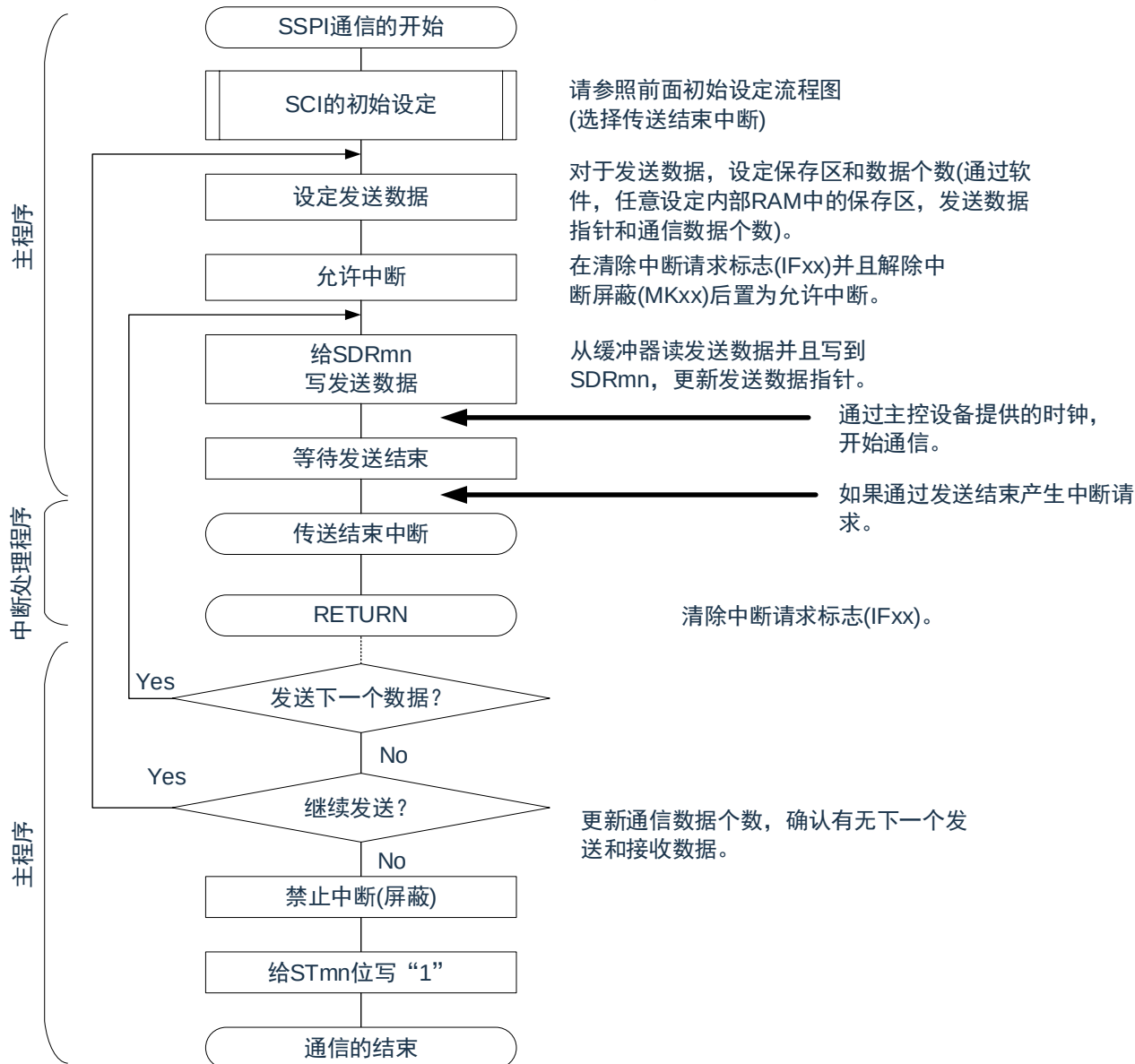


备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）。

图12-57：从属发送（单次发送模式）的流程图



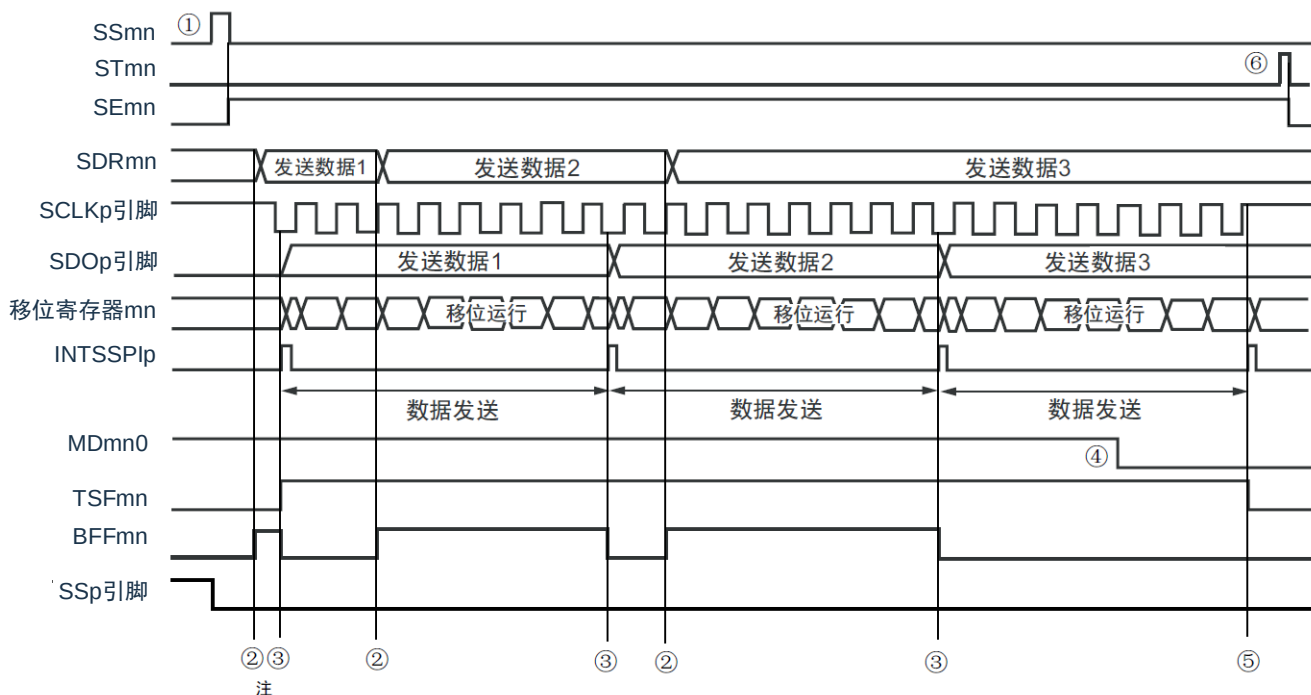
备注: m: 单元号 (m=0);

n: 通道号 (n=0);

p: SSPI 号 (p=00)。

(4) 处理流程（连续发送模式）

图12-58：从属发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

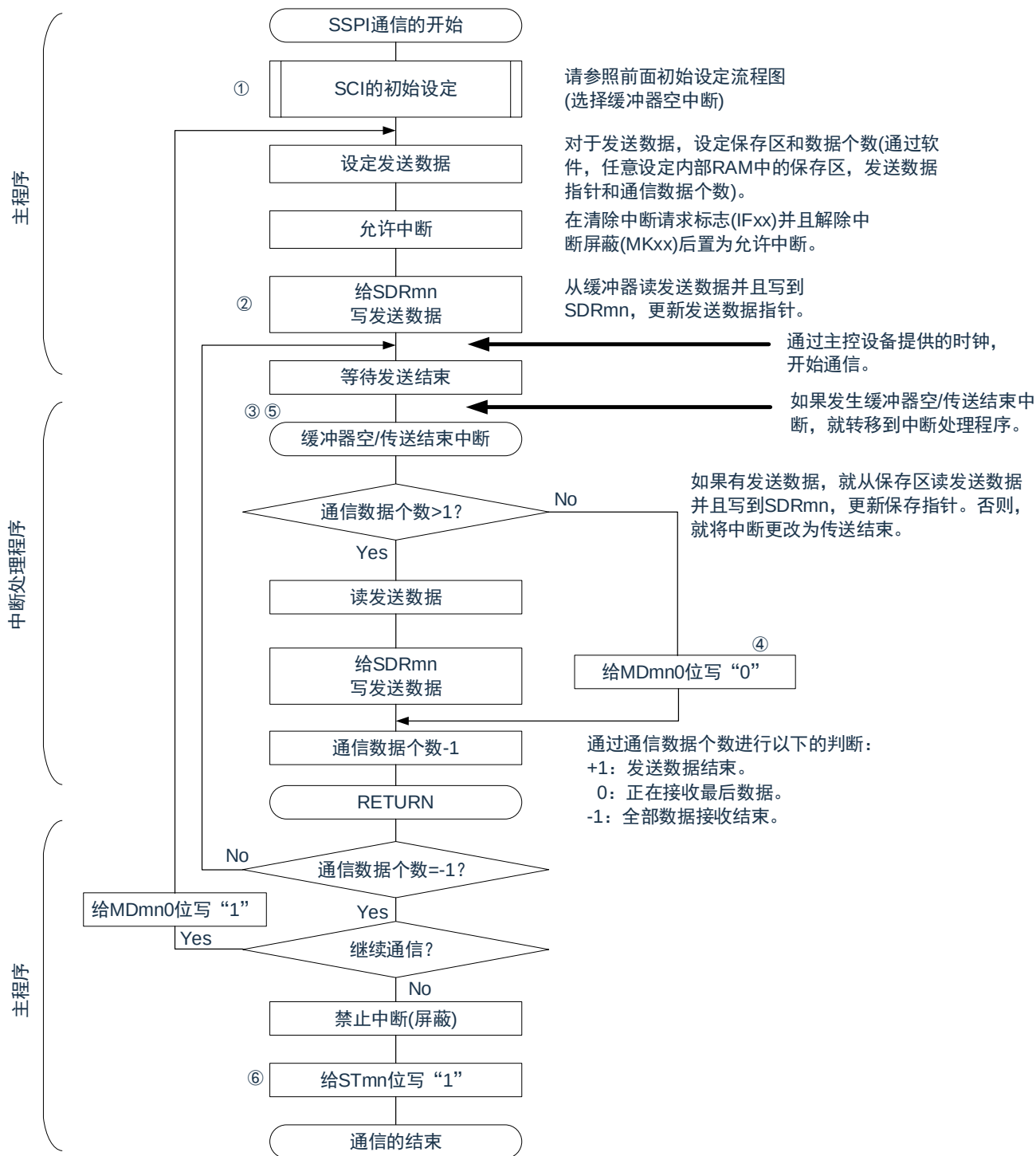
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）。

图12-59：从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“12-58 从属发送（连续发送模式）的时序图”中的①~⑥。

m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI号（p=00）。

12.6.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

表12-31：从属接收

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00、SS00
中断	INTSSPI00
	只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。
	DAPmn=0：在串行时钟开始运行时，开始数据输出。 DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。
	CKPmn=0：正相 CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注1：因为在内部对 SCLK00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为6M[Hz]。

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

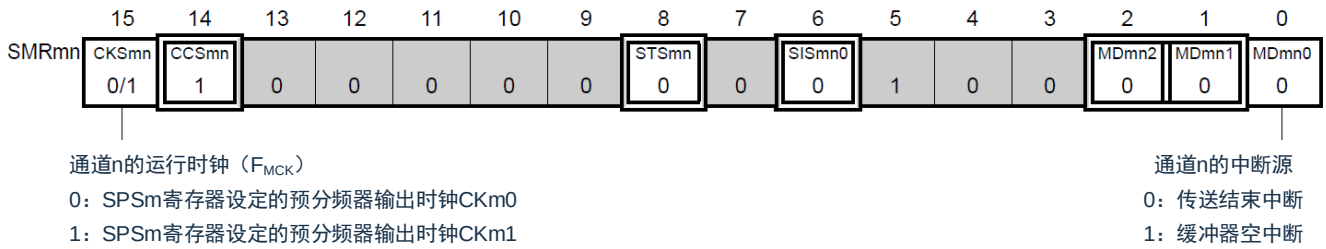
FMCK：对象通道的运行时钟频率

m：单元号（m=0） n：通道号（n=0）。

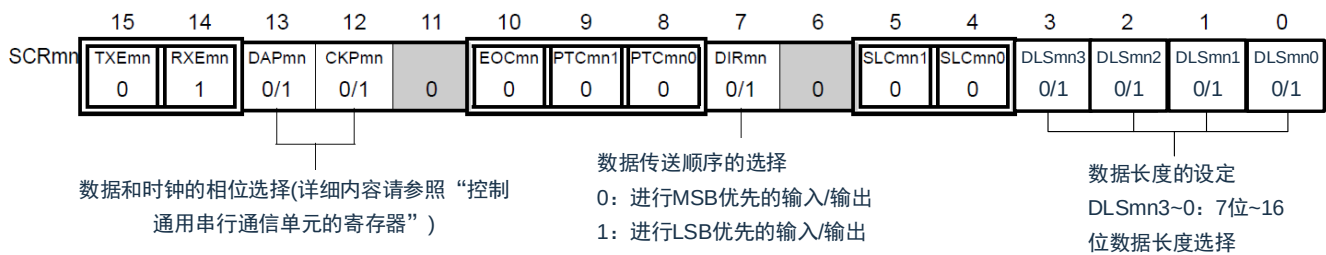
(1) 寄存器的设定

图12-60：从属选择输入功能（SSPI00）从属接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

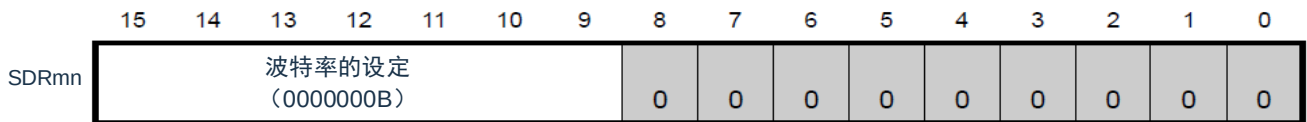


(b) 串行通信运行设定寄存器mn(SCRmn)

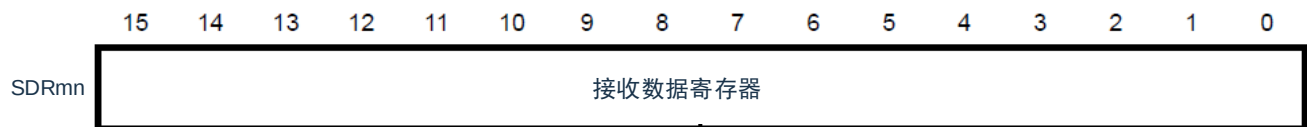


(c) 串行数据寄存器mn(SDRmn)

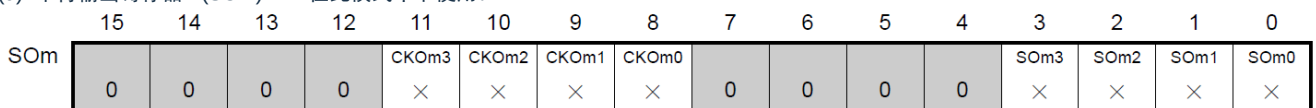
(1) 运行停止（SEmn=0）时



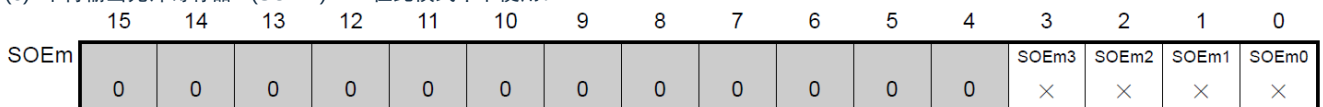
(2) 运行期间（SEmn=1）（低8位：SDRmnL）



(d) 串行输出寄存器m(SOm) ……在此模式中不使用。



(e) 串行输出允许寄存器m(SOEm) ……在此模式中不使用。



备注： m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI号（p=00）；

□：在SSPI主控接收模式中为固定设定； ■：不能设定（设定初始值）；

×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）；

0/1：根据用户的用途置“0”或者“1”。

图12-60：从属选择输入功能（SSPI00）从属接收时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

0: SS00引脚的输入值无效

1: SS00引脚的输入值有效

备注： m: 单元号（m=0）；

n: 通道号（n=0）；

p: SSPI 号（p=00）；

☐：在 SSPI 主控接收模式中为固定设定；

☐：不能设定（设定初始值）；

×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）；

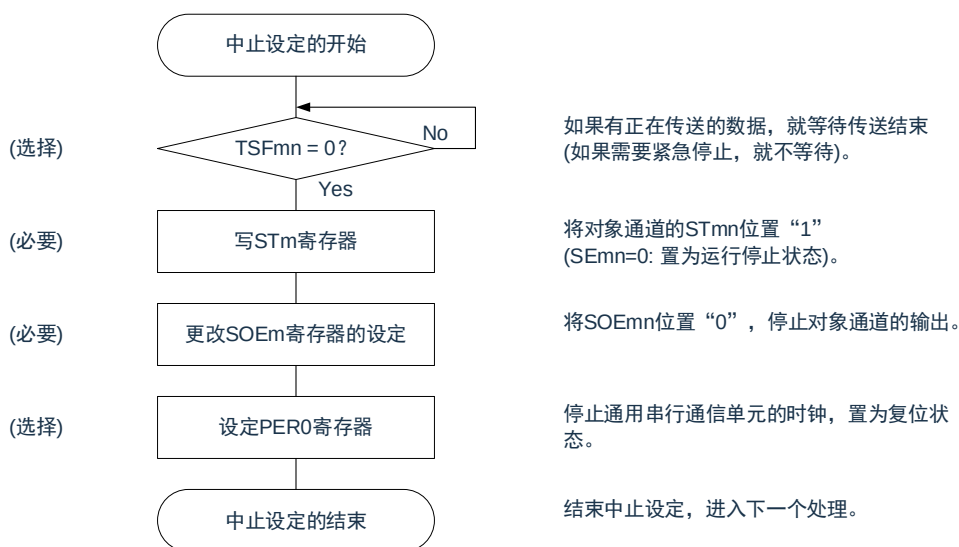
0/1: 根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-61：从属接收的初始设定步骤

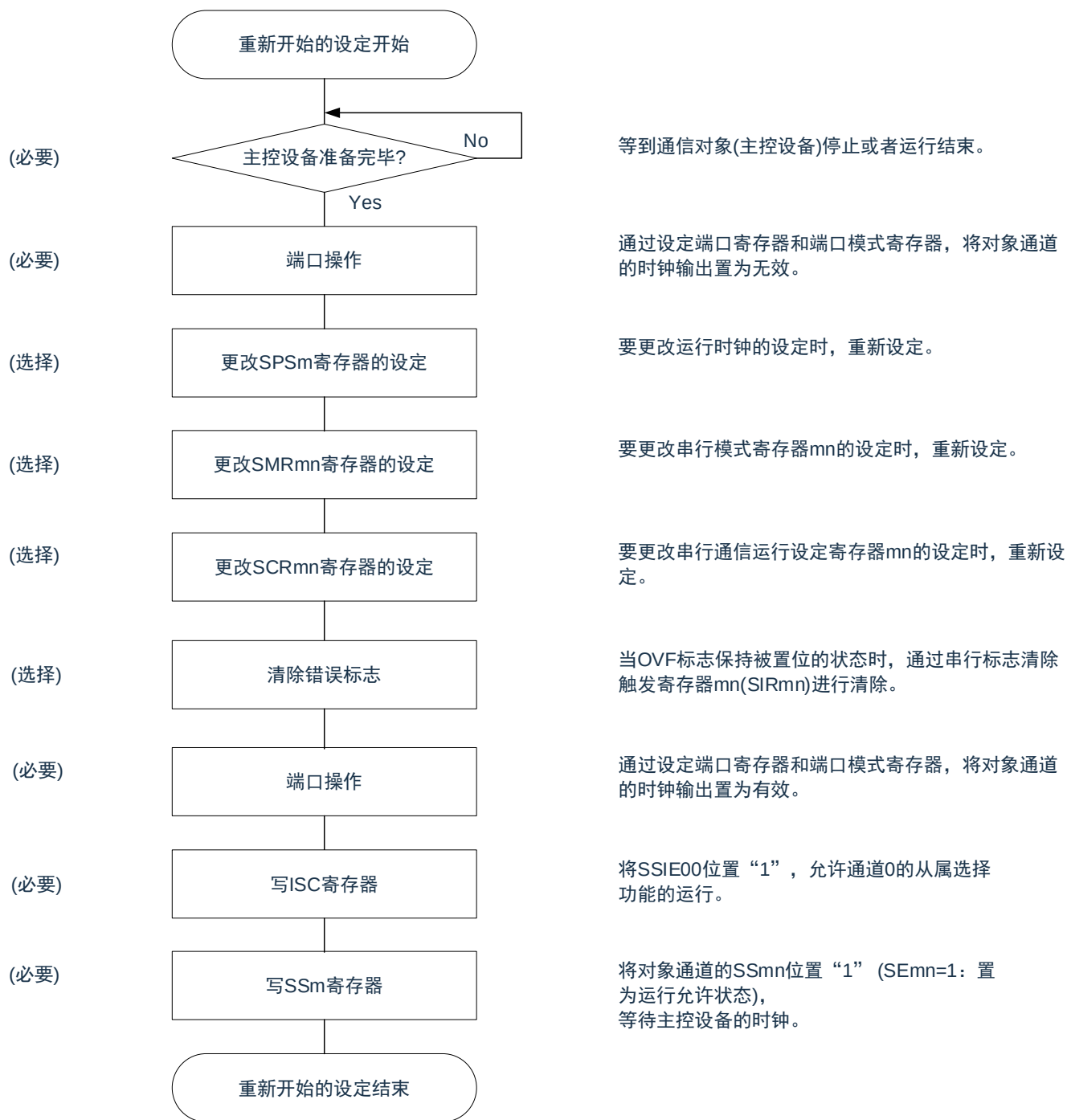


图12-62：从属接收的中止步骤



备注：m：单元号（m=0）；n：通道号（n=0）；p：SSPI号（p=00）。

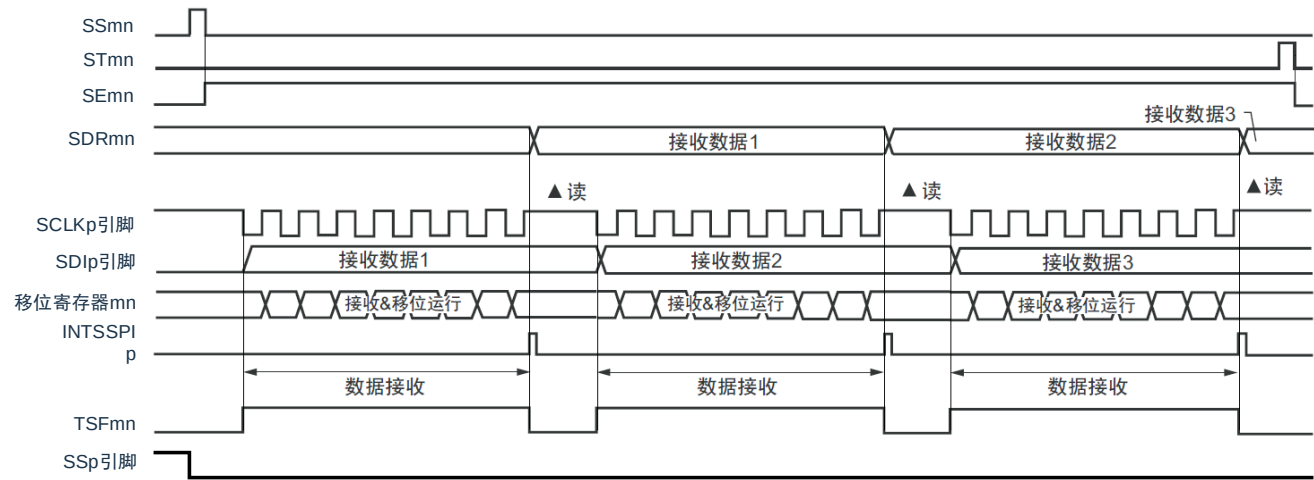
图12-63：重新开始从属接收的设定步骤



备注：m：单元号（m=0）；n：通道号（n=0）；p：SSPI号（p=00）。

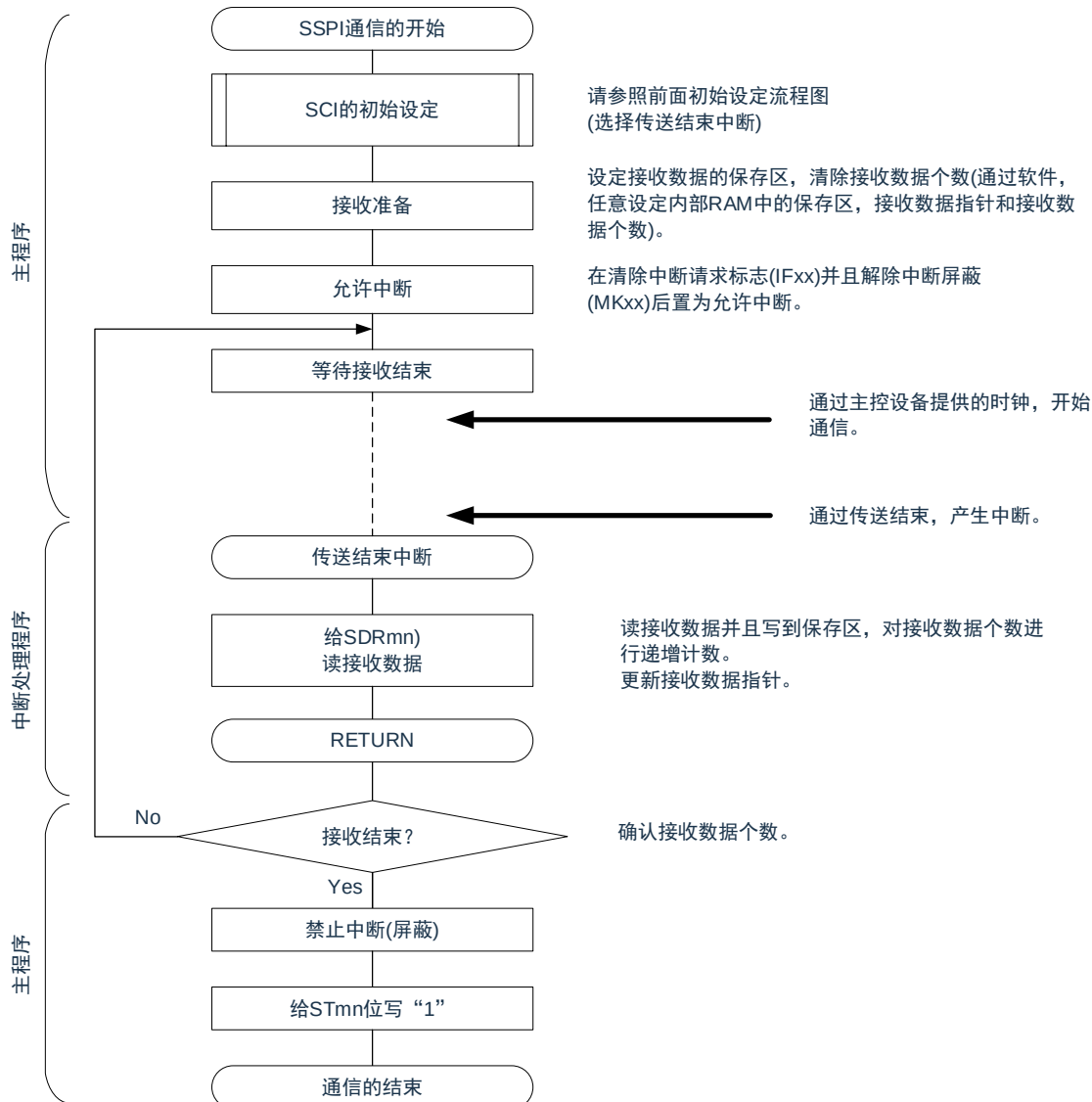
(3)处理流程（单次接收模式）

图12-64：从属接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0）；n：通道号（n=0）；p：SSPI号（p=00）。

图12-65：从属接收（单次接收模式）的流程图



12.6.3 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品和其他设备进行数据发送和接收的运行。

表12-32：从属的发送和接收

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00、SDO00、SS00
中断	INTSSPI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0：在串行时钟开始运行时，开始数据输出。 DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0：正相 CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注1：因为在内部对 SCLK00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为6M[Hz]。

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

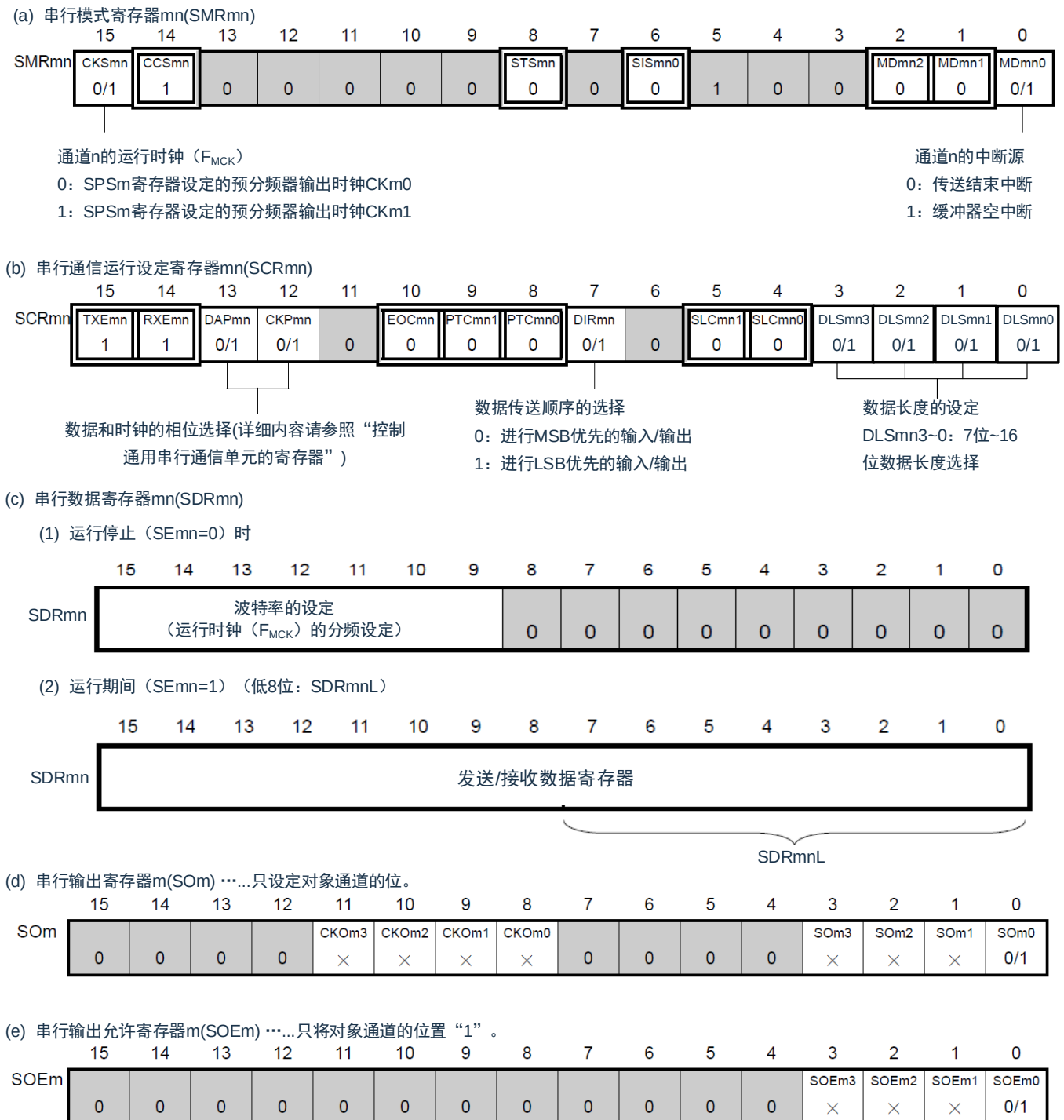
F_{MCK}：对象通道的运行时钟频率

m：单元号（m=0）；

n：通道号（n=0）。

(1) 寄存器的设定

图12-66：从属选择输入功能（SSPI00）从属发送和接收时的寄存器设定内容例子(1/2)



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：m: 单元号 (m=0)；

n: 通道号 (n=0)；

p: SSPI 号 (p=00)；

□: 在 SSPI 主控接收模式中为固定设定；

■: 不能设定 (设定初始值)；

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下，设定初始值)；

0/1: 根据用户的用途置“0”或者“1”。

图12-66：从属选择输入功能（SSPI00）从属发送和接收时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）；

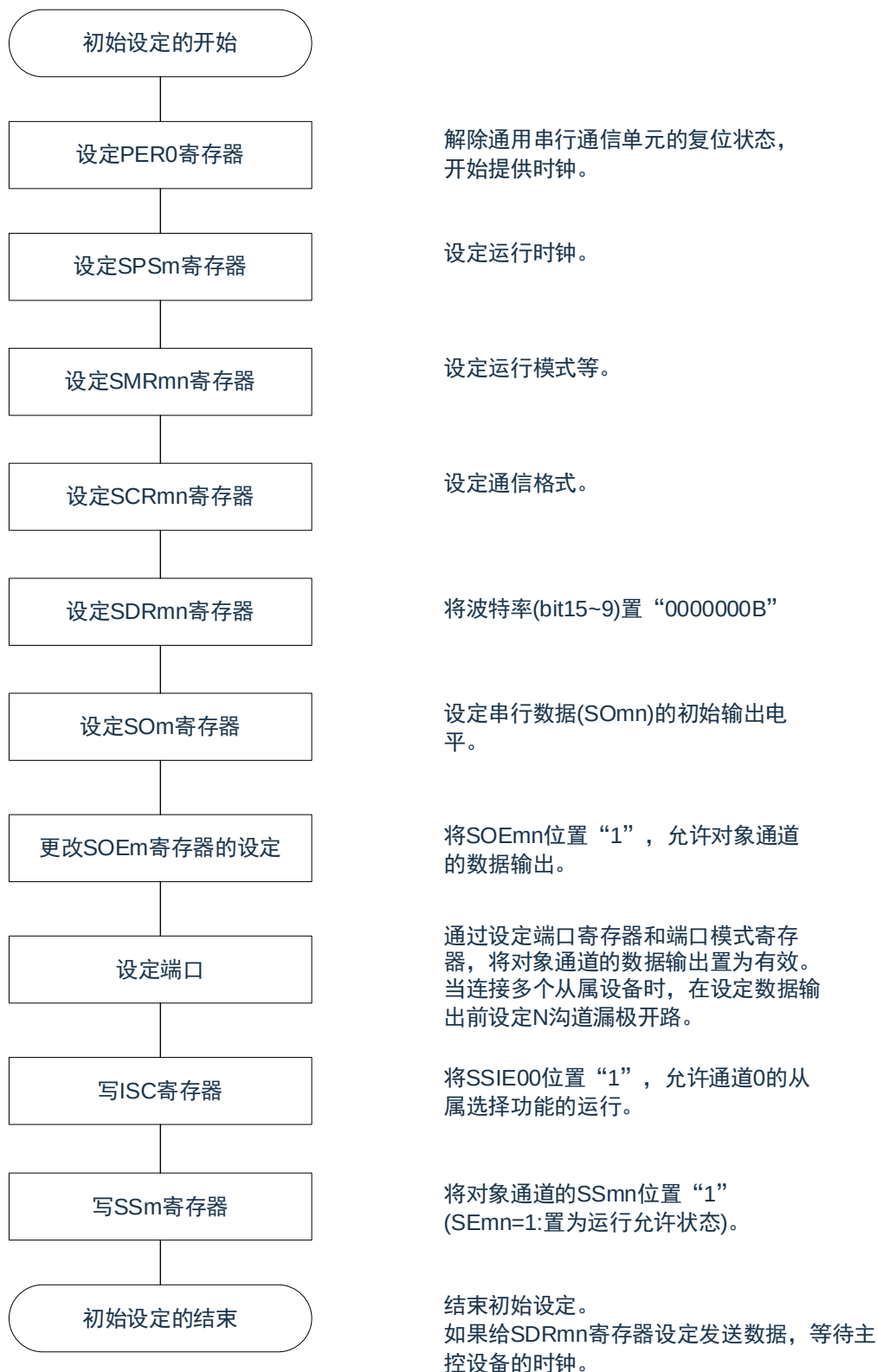
☐：在 SSPI 主控接收模式中为固定设定； ☐：不能设定（设定初始值）；

×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）；

0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图12-67：从属发送和接收的初始设定步骤



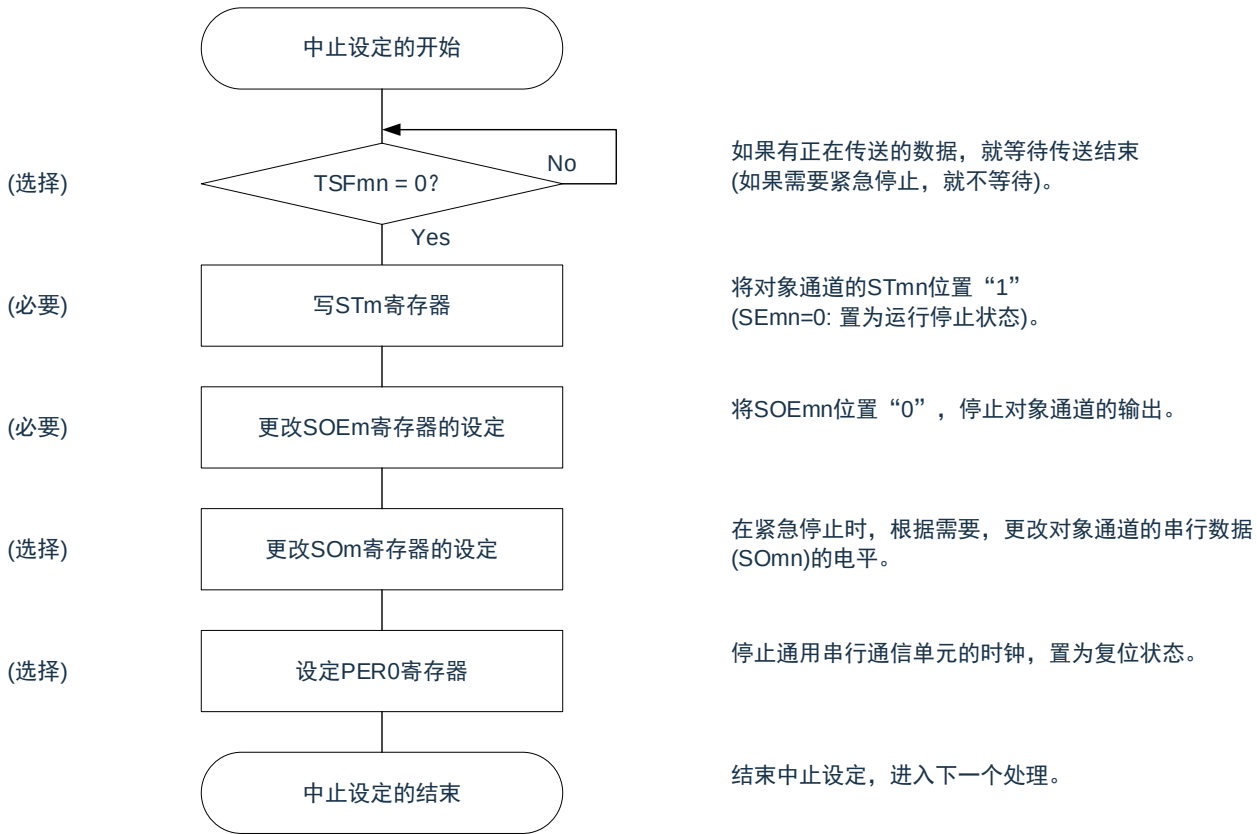
注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注： m: 单元号 (m=0)；

n: 通道号 (n=0)；

p: SSPI 号 (p=00)。

图12-68：从属发送和接收的中止步骤



备注： m：单元号（m=0）；
n：通道号（n=0）；
p：SSPI 号（p=00）。

图12-69：重新开始从属发送和接收的设定步骤



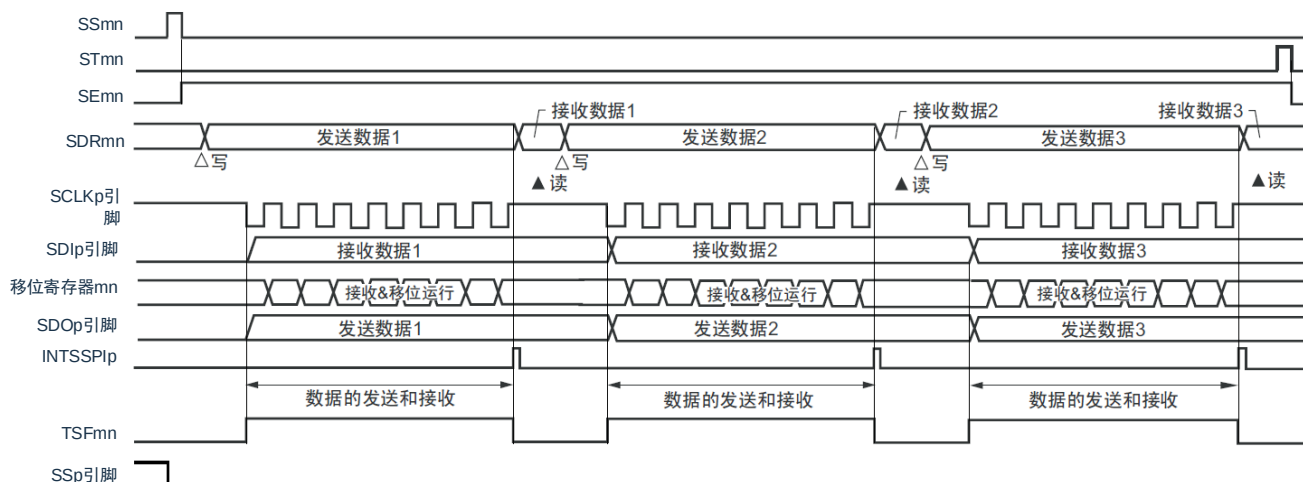
注意：

在主机设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主机设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

(3) 处理流程（单次发送和接收模式）

图12-70：从属发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）

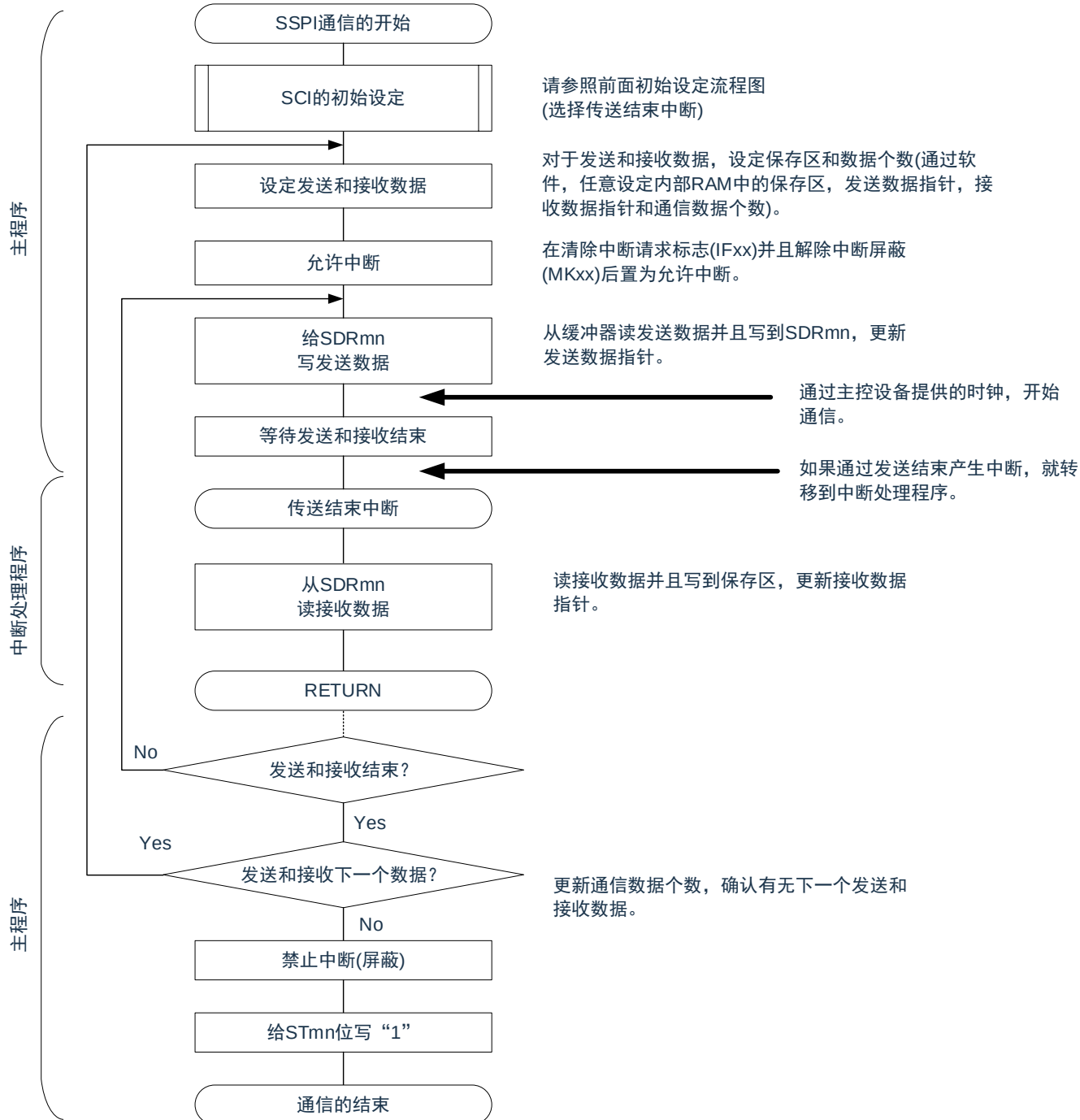


备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）。

图12-71：从属发送和接收（单次发送和接收模式）的流程图



注意：在主机设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

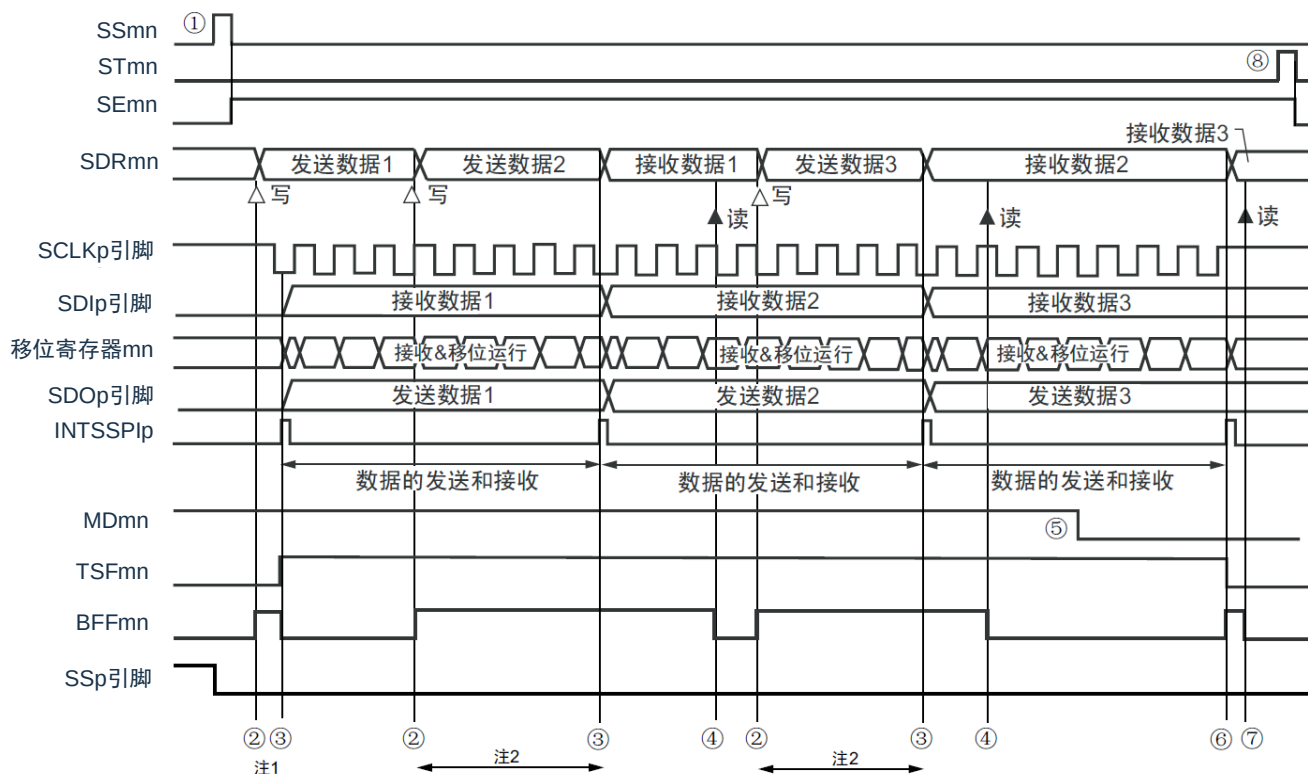
m: 单元号 (m=0)；

n: 通道号 (n=0)；

p: SSPI 号 (p=00)。

(4) 处理流程（连续发送和接收模式）

图12-72：从属发送和接收（连续发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



注1：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

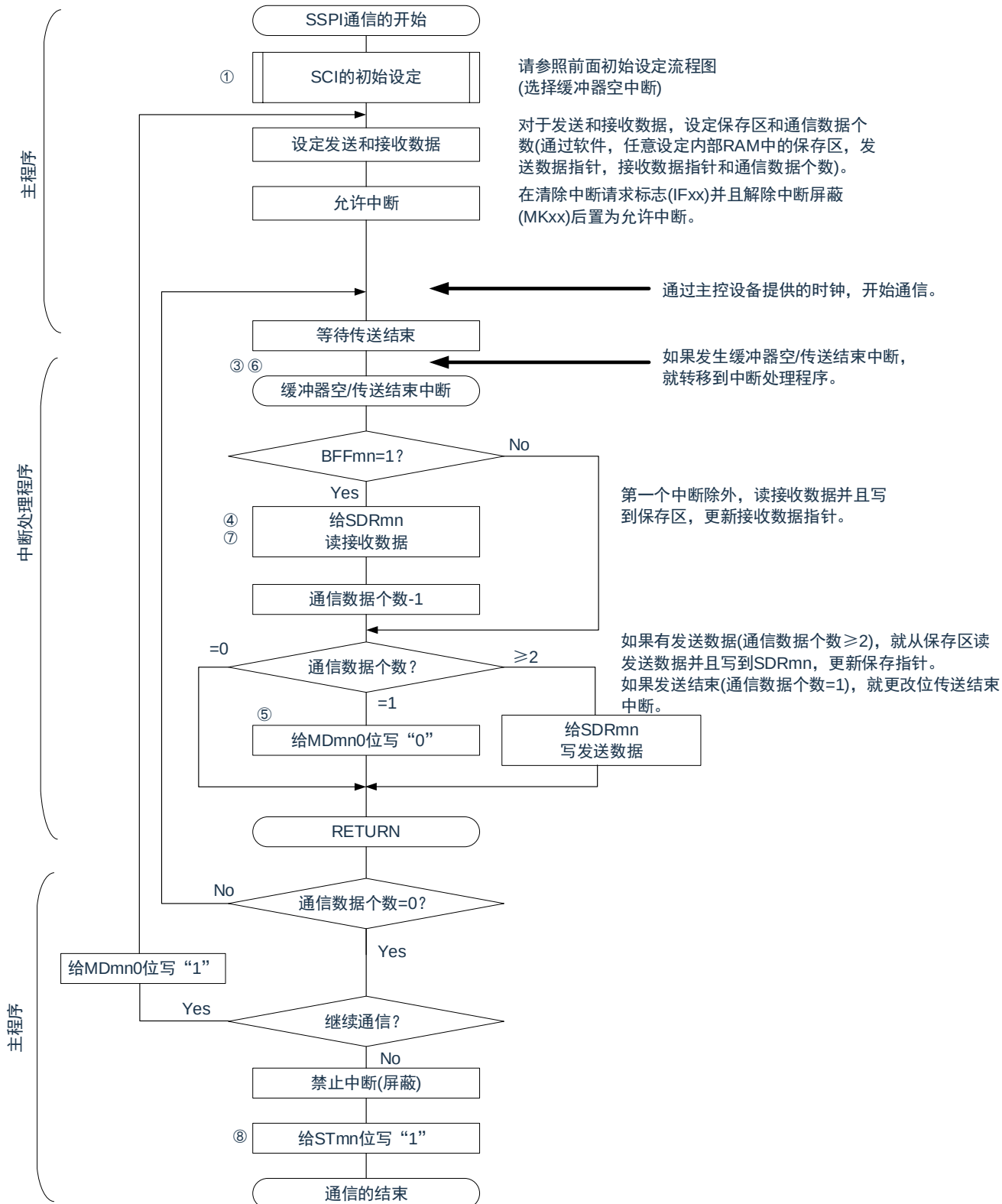
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注：

图中的①~⑧对应“12-73 从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

m：单元号（m=0）n：通道号（n=0）p：SSPI 号（p=00）。

图12-73：从属发送和接收（连续发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：

图中的①~⑧对应“12-72 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

m：单元号（m=0）n：通道号（n=0）p：SSPI 号（p=00）。

12.6.4 传送时钟频率的计算

从属选择输入功能（SSPI00）通信的传送时钟频率能用以下计算式进行计算。

从属设备：

$$(\text{传送时钟频率}) = \{\text{主控设备提供的串行时钟 (SCLK) 频率}\}^{2^p} [\text{Hz}]$$

注1： 容许的最大传送时钟频率为6MHz。

m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI 号 (p=00)。

表12-33: 从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟 (F _{MCK}) ^注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	5.86KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	2.93KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.46KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	732Hz

注：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

X: 忽略

m: 单元号 (m=0) n: 通道号 (n=0)。

12.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如表12-34所示：

表12-34：溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。	-	判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn（SDIRmn）写“1”。→	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0）n：通道号（n=0）。

12.7 UART0 通信的运行

这是通过串行数据发送（TxD）和串行数据接收（RxD）共2条线进行异步通信的功能。使用这2条通信线，按数据帧（由起始位、数据、奇偶校验位和停止位构成）与其他通信方进行异步（使用内部波特率）的数据发送和接收。能通过使用发送专用（偶数通道00/10）和接收专用（奇数通道01/11）共2个通道来实现全双工异步 UART 通信。

[数据的发送和接收]

- 7位、8位的数据长度
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定（选择电平是否反相）
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加、停止位的检测功能

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误和溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

UART0使用 SCIO的通道0和通道1。

各通道任意选择一个功能使用，除了所选功能以外，其他功能不能运行。

例如，在单元0的通道0和通道1使用 UART0时，不能使用 SSPI00和 IIC00。

UART 有以下2种通信运行：

- UART 发送 （参照12.7.1）
- UART 接收 （参照12.7.2）

注意：当用作 UART 时，发送方（偶数通道）和接收方（奇数通道）只能用于 UART。

12.7.1 UART 发送

UART 发送是本产品微控制器将数据异步发送到其他设备的运行。
UART 使用的2个通道中的偶数通道用于 UART 发送。

表12-35: UART 发送

UART	UART0
对象通道	SCI0的通道0
使用的引脚	TxD0
中断	INTST0
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	无
传送数据长度	7位、8位
传送速率	$\text{Max.F}_{\text{MCK}}/6[\text{bps}]$ （ $\text{SDRmn}[15:9] \geq 3$ ）、 $\text{Min.F}_{\text{CLK}}/(2 \times 2^{11} \times 128)[\text{bps}]^{\text{注}}$
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。
奇偶校验位	可选择以下内容： 无奇偶校验位。 附加零校验。 附加偶校验。 附加奇校验。
停止位	可选择以下内容： 附加1位。 附加2位。
数据方向	MSB优先或者LSB优先

注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

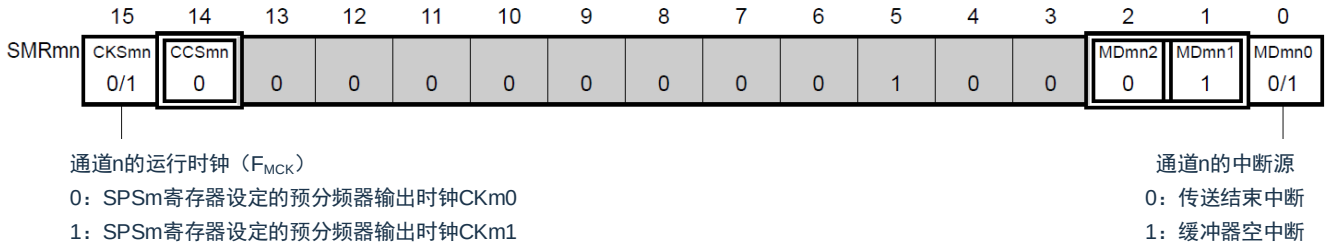
备注：

- 1. F_{MCK} ：对象通道的运行时钟频率
- 2. F_{CLK} ：系统时钟频率
- 3. m ：单元号（ $\text{m}=0$ ） n ：通道号（ $\text{n}=0$ ）。

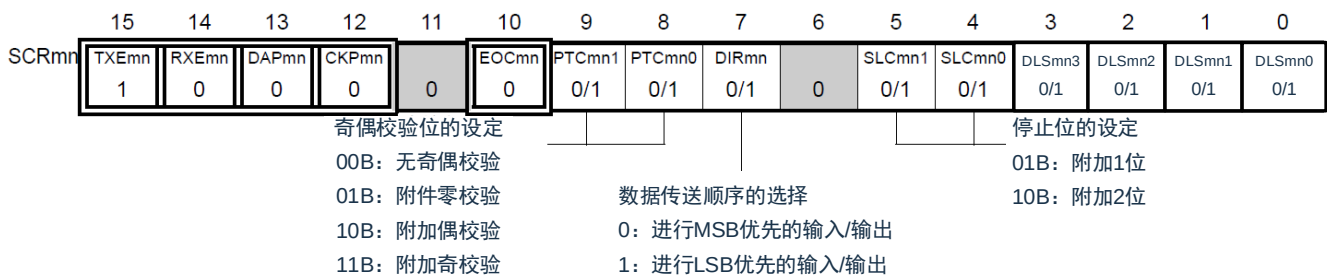
(1)寄存器的设定

图12-74: UART (UART0) 的 UART 发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

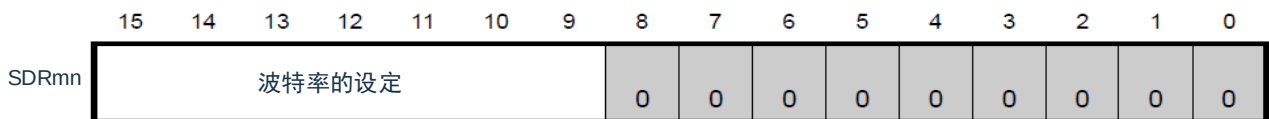


(b) 串行通信运行设定寄存器mn(SCRmn)

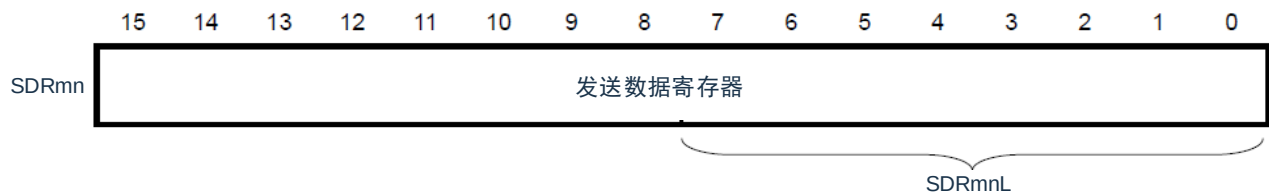


(c) 串行数据寄存器mn(SDRmn)

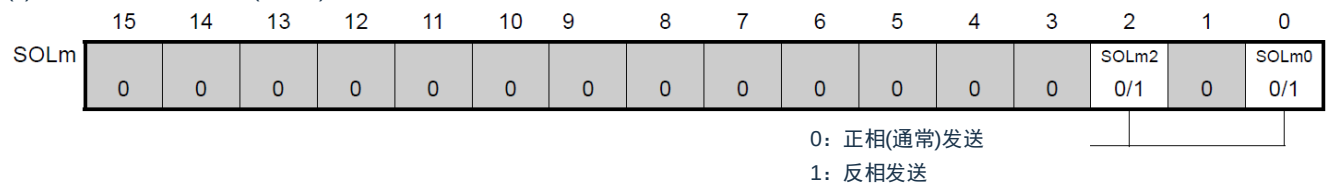
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(d) 串行输出电平寄存器m(SOLm) ……只设定对象通道的位。



备注: m: 单元号 (m=0) n: 通道号 (n=0) q: UART 号 (q=0)

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图12-74: UART (UART0) 的 UART 发送时的寄存器设定内容例子(2/2)

(e) 串行输出寄存器m(SOm) ……只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1 注

0: 串行数据输出值为“0”
1: 串行数据输出值为“1”

(f) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

(g) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

注意：在开始发送前，当对应通道的 SOLmn 位为“0”时，必须置“1”；当对应通道的 SOLmn 位为“1”时，必须置“0”。在通信过程中，值因通信数据而变。

备注：m: 单元号 (m=0) n: 通道号 (n=0) q: UART 号 (q=0)

☐ : 在 SSPI 主控接收模式中为固定设定；

☐ : 不能设定 (设定初始值)；

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下，设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-75: UART 发送的初始设定步骤

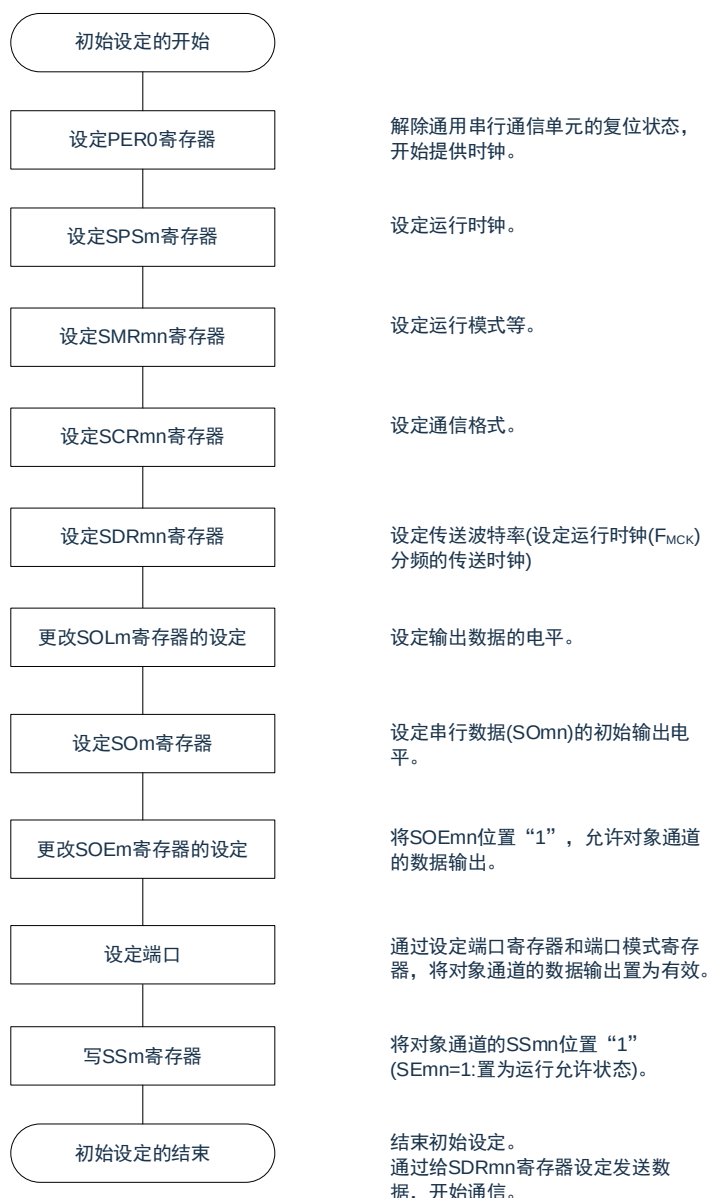


图12-76: UART 发送的中止步骤

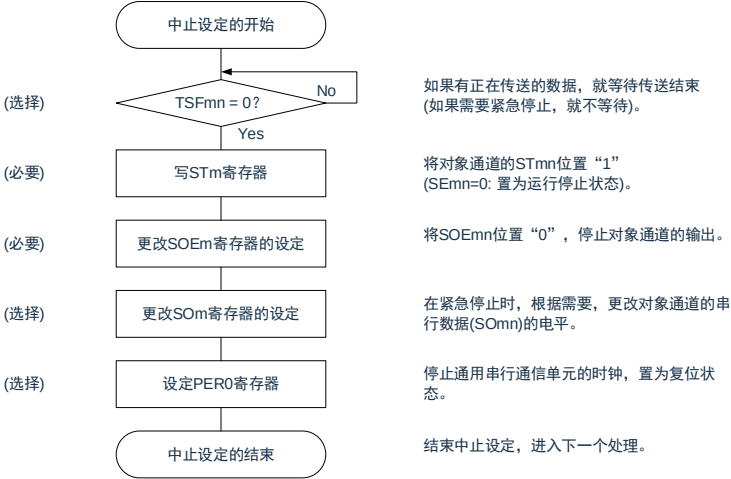
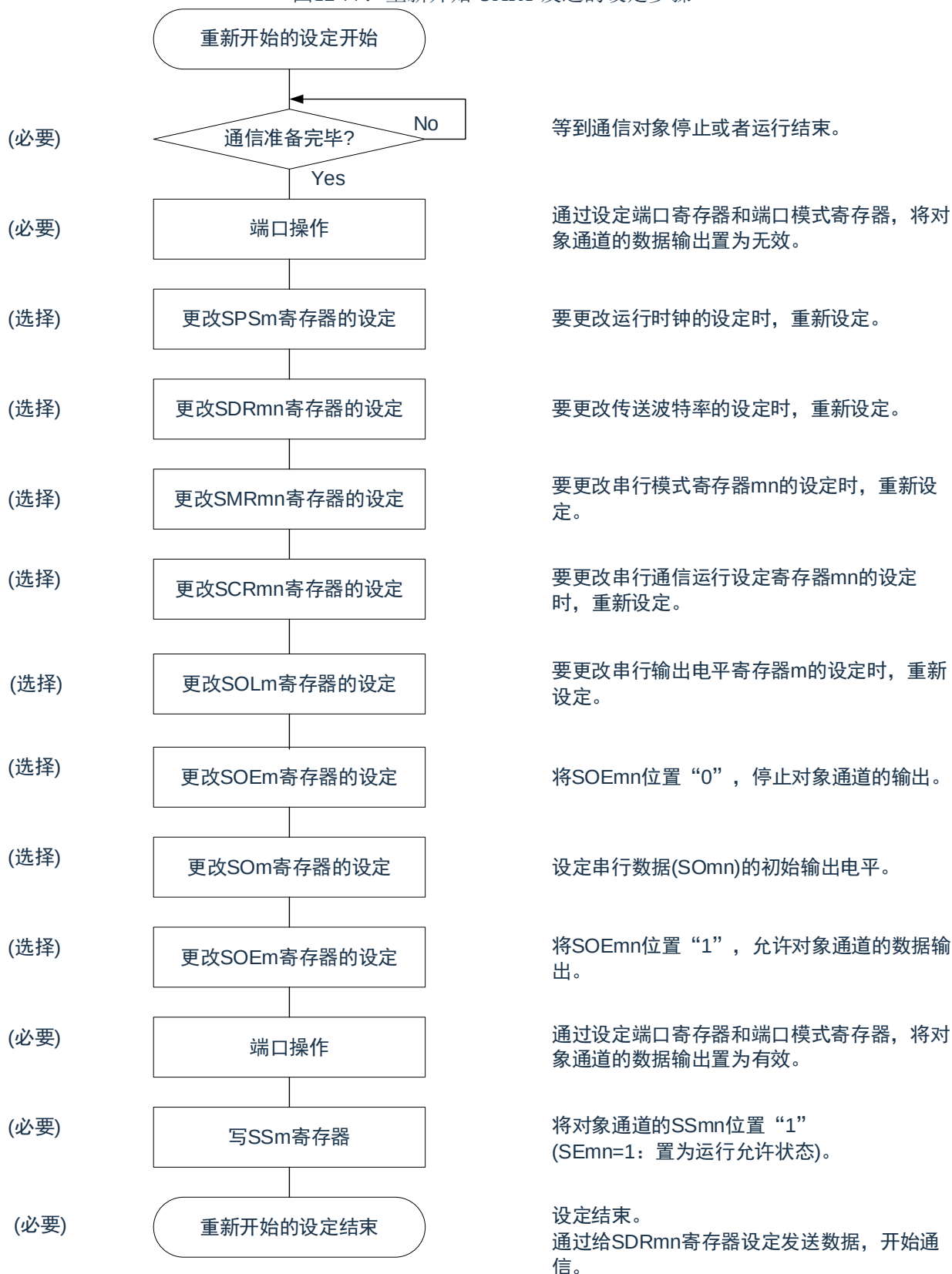


图12-77：重新开始 UART 发送的设定步骤



注意：如果在中止设定中改写 PER0来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始设定。

(3) 处理流程（单次发送模式）

图12-78: UART 发送（单次发送模式）的时序图

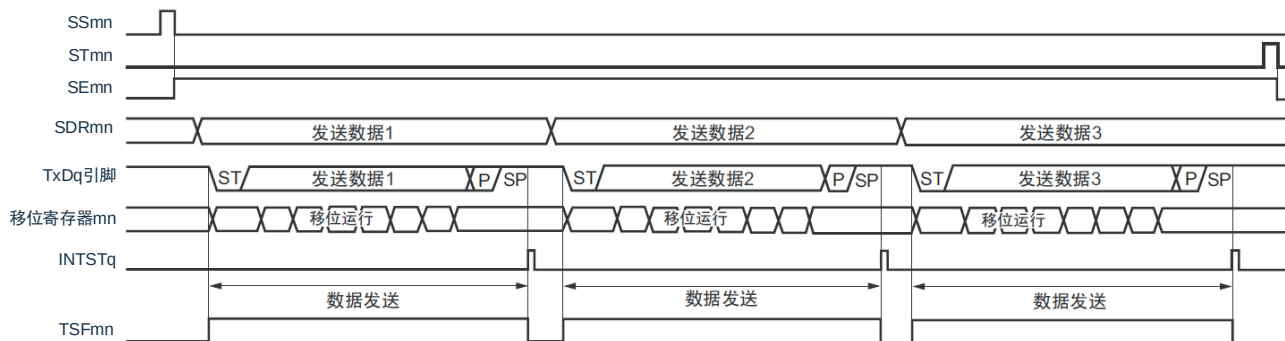
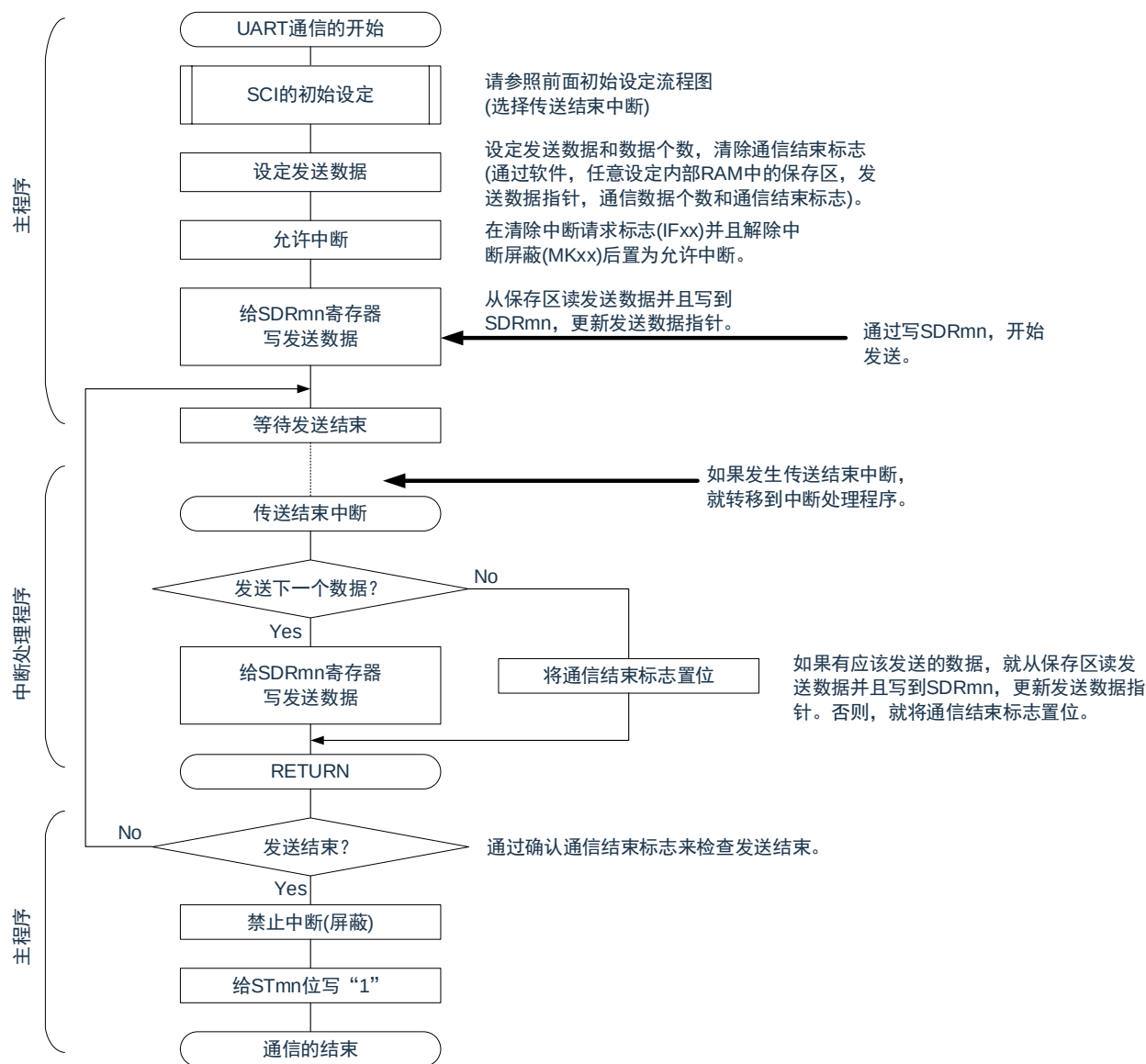


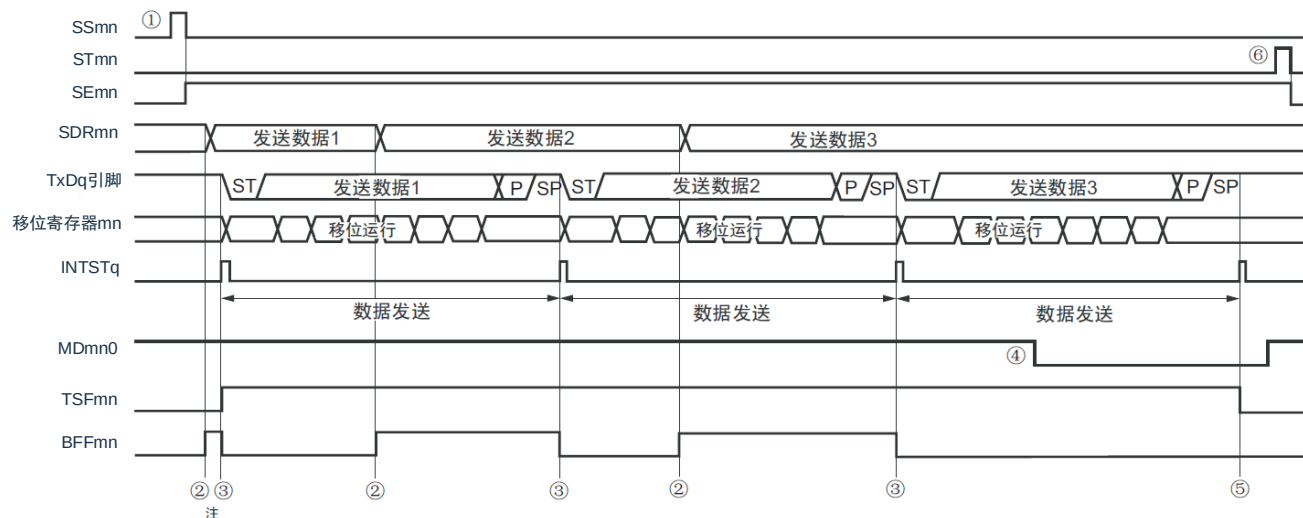
图12-79: UART 发送（单次发送模式）的流程图



备注: m: 单元号 (m=0) n: 通道号 (n=0、) q: UART 号 (q=0)。

(4)处理流程（连续发送模式）

图12-80：UART 发送（连续发送模式）的时序图

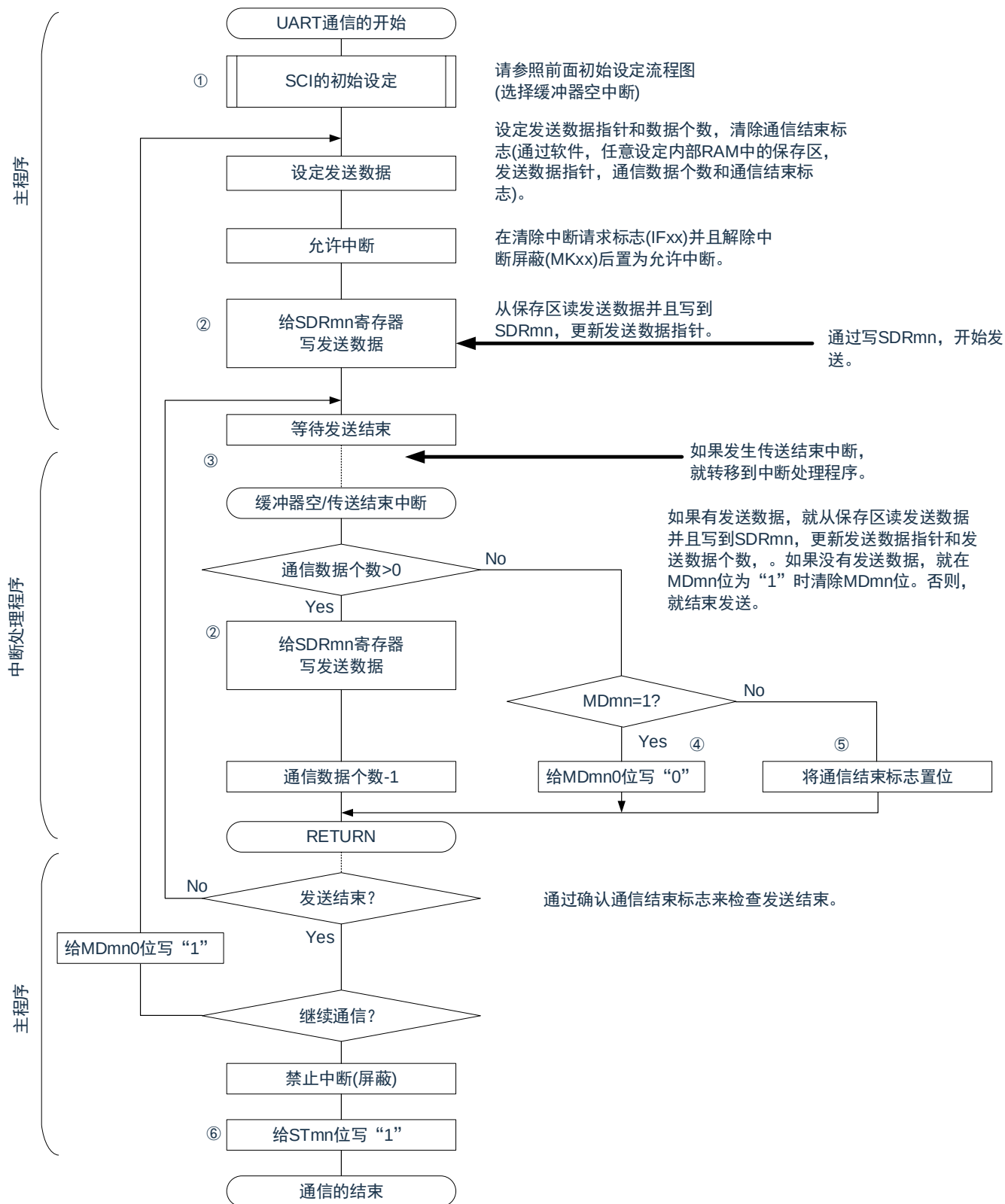


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注：m：单元号（m=0）n：通道号（n=0）q：UART 号（q=0）。

图12-81: UART 发送（连续发送模式）的流程图



备注：图中的①~⑥对应“12-80 UART 发送（连续发送模式）的时序图”中的①~⑥。

12.7.2 UART 接收

UART 接收是本产品微控制器其他设备异步接收数据的运行。
UART 使用的2个通道中的奇数通道用于 UART 接收。但是，需要设定奇数通道和偶数通道的 SMR 寄存器。

表12-36: UART 接收

UART	UART0
对象通道	SCI0的通道1
使用的引脚	RxD0
中断	INTSR0
	只限于传送结束中断（禁止设定缓冲器空中断）。
错误中断	INTSRE0
错误检测标志	帧错误检测标志（FEFmn） 奇偶校验错误检测标志（PEFmn） 溢出错误检测标志（OVFmn）
传送数据长度	7位、8位
传送速率	$\text{Max.F}_{\text{MCK}}/6[\text{bps}]$ （ $\text{SDRmn}[15:9] \geq 2$ ）、 $\text{Min.F}_{\text{CLK}}/(2 \times 2^{15} \times 128)[\text{bps}]$
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。
奇偶校验位	可选择以下内容： 无奇偶校验位（无奇偶校验）。 附加零校验（无奇偶校验）。 偶校验 奇校验
停止位	附加1位。
数据方向	MSB优先或者LSB优先

注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

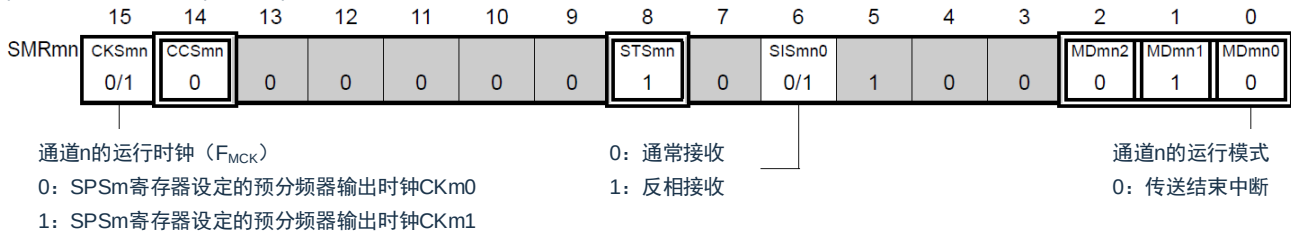
备注：

- F_{MCK}: 对象通道的运行时钟频率
- F_{CLK}: 系统时钟频率
- m: 单元号（m=0） n: 通道号（n=1、3）。

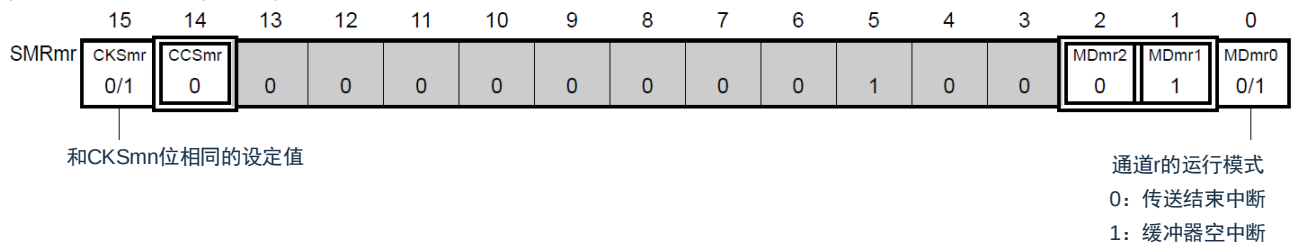
(1)寄存器的设定

图12-81: UART (UART0) 的 UART 接收时的寄存器设定内容例子(1/2)

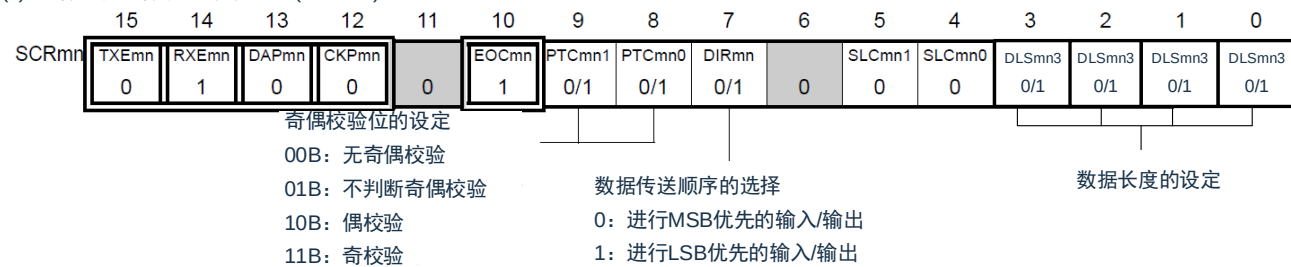
(a) 串行模式寄存器mn(SMRmn)



(b) 串行模式寄存器mr(SMRmr)

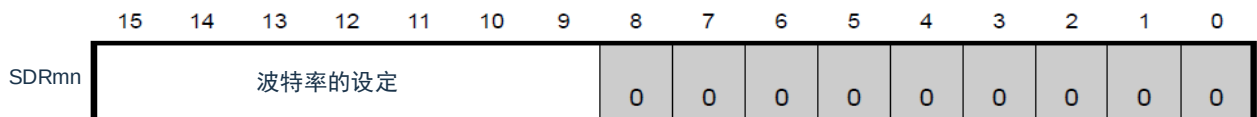


(c) 串行通信运行设定寄存器mn(SCRmn)

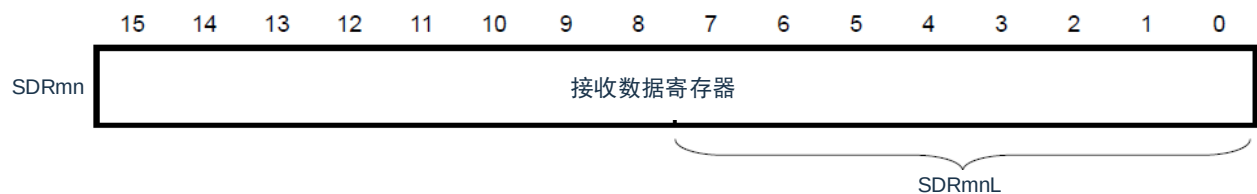


(d) 串行数据寄存器mn(SDRmn)

(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



注意: 在 UART 接收时, 还必须设定与通道 n 成对的通道 r 的 SMRmr 寄存器。

备注: m: 单元号 (m=0); n: 通道号 (n=1、3);

r: 通道号 (r=n-1); q: UART 号 (q=0~5)。

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图12-82: UART (UART0) 的 UART 接收时的寄存器设定内容例子(2/2)

(e) 串行输出寄存器m(SOm) ……在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	×	×

(f) 串行输出允许寄存器m(SOEm) ……在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	×

(g) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×

注意：在 UART 接收时，还必须设定与通道 n 成对的通道 r 的 SMRmr 寄存器。

备注：m：单元号（m=0）；n：通道号（n=1）；

r：通道号（r=n-1）；q：UART 号（q=0）。

☐ ：在此模式中为固定设定； ☐ ：不能设定（设定初始值）；

×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。

0/1：根据用户的用途置“0”或者“1”。

(2)操作步骤

图12-83: UART 接收的初始设定步骤



注意：必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔4个 F_{MCK} 时钟，然后将 SSmn 位置“1”。

图12-84: UART 接收的中止步骤

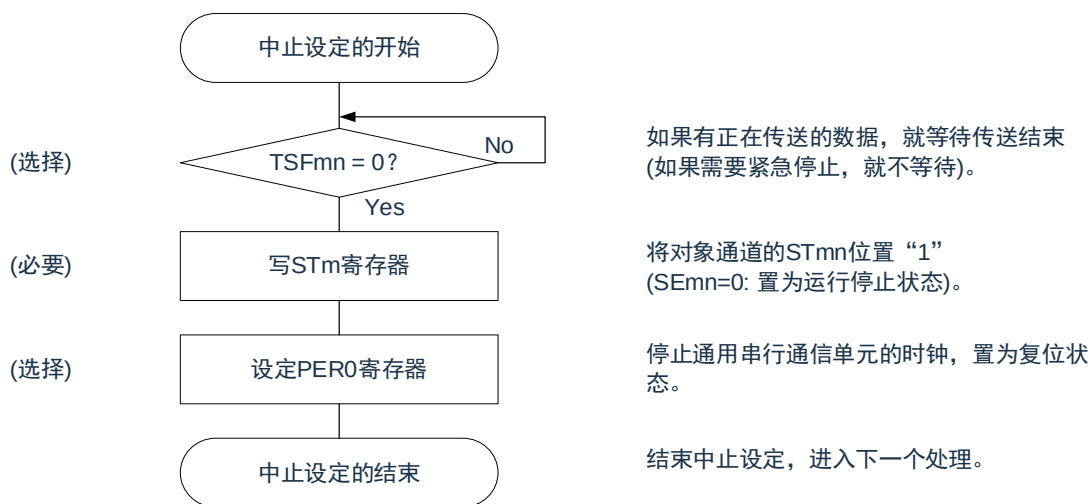


图12-85：重新开始 UART 接收的设定步骤



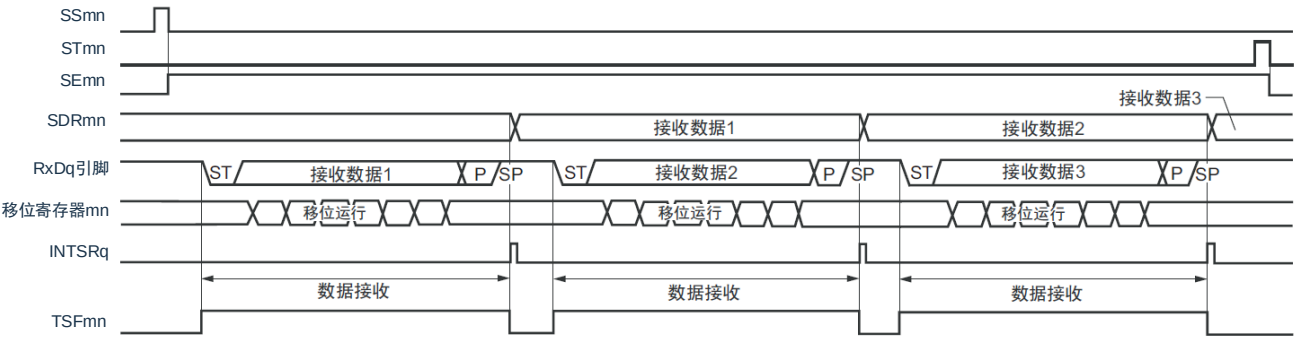
注意：

必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 FMCK 时钟，然后将 SSmn 位置“1”。

如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始设定。

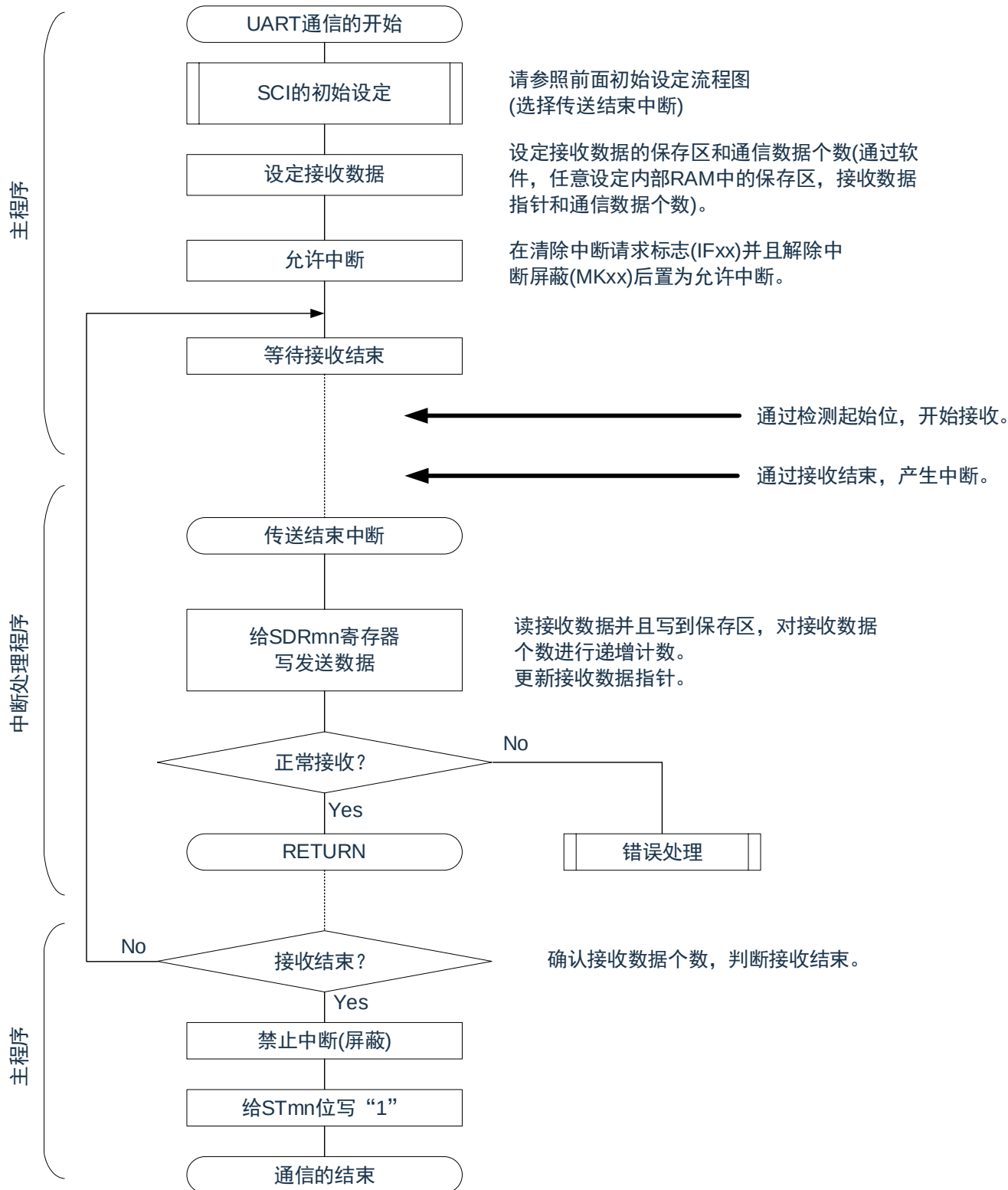
(3) 处理流程

图12-86: UART 接收的时序图



备注: m: 单元号 (m=0) ; n: 通道号 (n=1) ;
r: 通道号 (r=n-1) ; q: UART 号 (q=0) 。

图12-87: UART 接收的流程图



12.7.3 波特率的计算

波特率的计算式

UART0通信的波特率能用以下计算式进行计算：

$$(\text{波特率}) = \{\text{对象通道的运行时钟 (F}_{\text{MCK}}\text{) 频率}\} / (\text{SDRmn}[15:9] + 1) / 2 [\text{bps}]$$

运行时钟 (F_{MCK}) 取决于串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的 bit15 (CKSmn 位)。

注意：禁止将串行数据寄存器 mn (SDRmn) 的 SDRmn[15:9] 设定为“0000000B”和“0000001B”。

备注：因为在使用 UART 时 SDRmn[15:9] 的值为 SDRmn 寄存器的 bit15~9 的值 (0000010B~1111111B)，所以为 2~127。

m: 单元号 (m=0) n: 通道号 (n=0、1)。

表12-37: UART 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟 (F _{CLK}) ^注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	5.86KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	2.93KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.46KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	732Hz
1	0	0	0	0	X	X	X	X	F _{CLK}	24MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	12MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	6MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	3MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	1.5MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	0.75MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	375KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	187.5KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	93.75KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	46.875KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	23.44KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	11.72KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	5.86KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	2.93KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.46KHz
	1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	732Hz

注：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

X：忽略

m：单元号（m=0）n：通道号（n=0、1）。

发送时的波特率误差

UART0通信发送时的波特率误差能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

$$(\text{波特率误差}) = (\text{波特率的计算值}) / (\text{目标波特率的值}) \times 100 - 100[\%]$$

F_{CLK}=24MHz 时的 UART 波特率的设定例子如下所示：

UART波特率 (目标波特率)	F _{CLK} =24MHz			
	运行时钟 (F _{MCK})	SDRmn[15:9]	波特率的计算值	与目标波特率的误差
300bps	F _{CLK} /2 ⁹	77	300.48bps	+0.16%
600bps	F _{CLK} /2 ⁸	77	600.96bps	+0.16%
1200bps	F _{CLK} /2 ⁷	77	1201.92bps	+0.16%
2400bps	F _{CLK} /2 ⁶	77	2403.85bps	+0.16%
4800bps	F _{CLK} /2 ⁵	77	4807.69bps	+0.16%
9600bps	F _{CLK} /2 ⁴	77	9615.38bps	+0.16%
19200bps	F _{CLK} /2 ³	77	19230.8bps	+0.16%
31250bps	F _{CLK} /2 ³	47	31250.0bps	±0.0%
38400bps	F _{CLK} /2 ²	77	38461.5bps	+0.16%
76800bps	F _{CLK} /2	77	76923.1bps	+0.16%
153600bps	F _{CLK}	77	153846bps	+0.16%
312500bps	F _{CLK}	37	315789bps	±1.05%

备注：m：单元号（m=0）n：通道号（n=0、1）。

接收时的波特率容许范围

UART0通信接收时的波特率容许范围能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

$$(\text{可接收的最大波特率}) = \frac{2 * k * Nfr}{2 * k * Nfr - k + 2} \times Brate$$

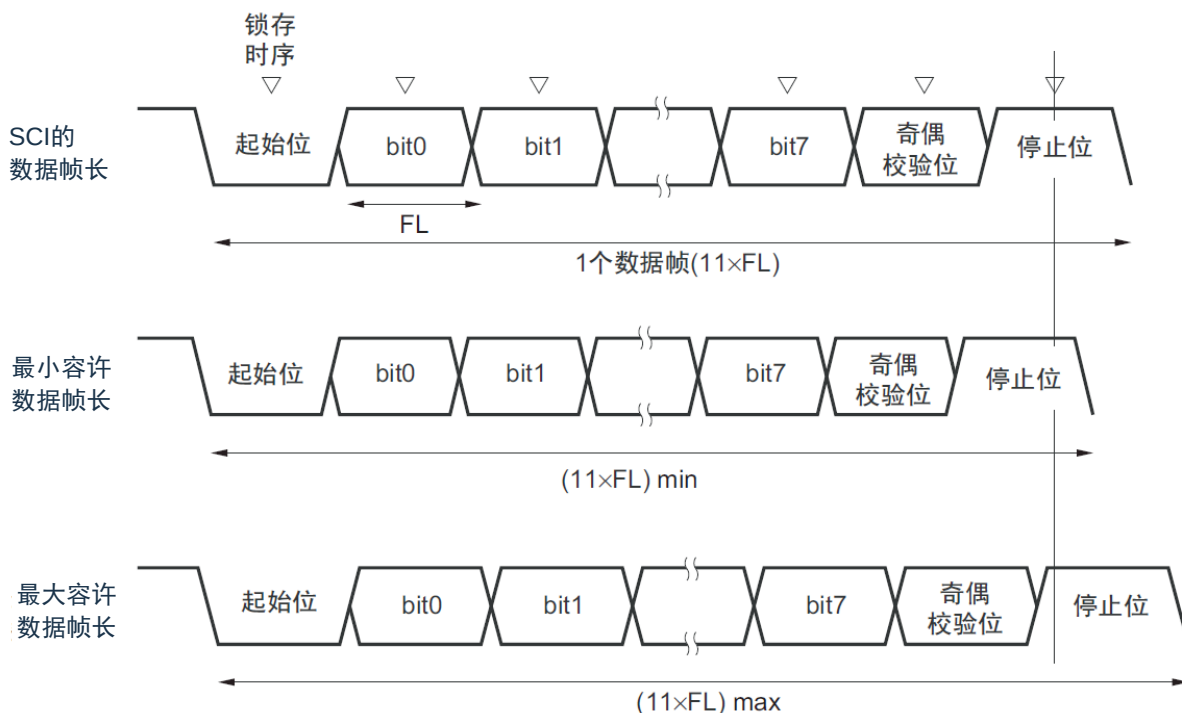
$$(\text{可接收的最小波特率}) = \frac{2 * k * (Nfr - 1)}{2 * k * Nfr - k - 2} \times Brate$$

Brate: 接收方的波特率的计算值（参照“12.7.4 (1)波特率的计算式”）。

K: SDRmn[15:9]+1。

Nfr: 1个数据的帧长[位] = (起始位) + (数据长度) + (奇偶校验位) + (停止位)。

图12-88: 接收时的波特率容许范围（1个数据的帧长=11位的情况）



如图12-88所示，在检测到起始位后，接收数据的锁存时序取决于串行数据寄存器（SDRmn）的 bit15~9设定的分频比。如果最后的数据（停止位）能赶上此锁存时序，就能正常接收。

备注：m: 单元号（m=0）；n: 通道号（n=0、1）。

12.7.4 在 UART0 通信过程中发生错误时的处理步骤

在 UART0通信过程中发生错误时的处理步骤如表12-38和表12-39所示：

表12-38：发生奇偶校验错误或者溢出错时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn（SDRmn）。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。	→ 清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

表12-39：发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn（SIRmn）。	→ 清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m（STm）的STmn位置“1”。	→ 串行通道允许状态寄存器m（SEm）的SEmn位为“0”并且通道n为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存m（SSm）的SSmn位置“1”。	→ 串行通道允许状态寄存器m（SEm）的SEmn位为“1”并且通道n为可运行状态。	

备注：m：单元号（m=0）n：通道号（n=0、1）。

12.8 简易 I²C（IIC00）通信的运行

这是通过串行时钟（SCL）和串行数据（SDA）共2条线与多个设备进行时钟同步通信的功能。因为此简易 I²C 是为了与 EEPROM、闪存、A/D 转换器等设备进行单通信而设计的，所以只用作主控设备。

对于开始条件和停止条件，必须遵守 AC 规格，在操作控制寄存器的同时通过软件进行处理。

[数据的发送和接收]

- 主控发送、主控接收（只限于单主控的主控功能）
- ACK 输出功能注、ACK 检测功能
- 8位数据长度（在发送地址时，用高7位指定地址，用最低位进行 R/W 控制）
- 通过软件产生开始条件和停止条件。

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误

[简易 I²C 不支持的功能]

- 从属发送、从属接收
- 多主控功能（仲裁失败检测功能）
- 等待检测功能

SCI0的通道0是支持简易 I²C(IIC00)的通道。

简易 I²C（IIC00）有以下4种通信运行：

- 地址段发送（参照12.8.1）
- 数据发送（参照12.8.2）
- 数据接收（参照12.8.3）
- 停止条件的产生（参照12.8.4）

注意：在接收最后的数据时，如果给 SDOEmn 位（SDOEm 寄存器）写“0”来停止串行通信数据的输出，就不输出 ACK。详

细内容请参照“12.8.3 (2)处理流程”。

备注：m：单元号（m=0）n：通道号（n=0、1）。

12.8.1 地址段发送

地址段发送是为了特别指定传送对象（从属设备）而在 I²C 通信时最先进行的发送运行。在产生开始条件后，将地址（7位）和传送方向（1位）作为1帧进行发送。

表12-42：地址段发送

简易I ² C	IIC00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SAU_SDA00 ^{注1}
中断	INTIIC00 只限于传送结束中断（不能选择缓冲器空中断）。
错误检测标志	ACK错误检测标志（PEFmn）
传送数据长度	8位（将高7位作为地址并且将低1位作为R/W控制进行发送）
传送速率 ^{注2}	Max.F _{MCK} /4[Hz]（SDRmn[15:9]≥1）F _{MCK} ：对象通道的运行时钟频率但是，必须在I ² C的各模式中满足以下条件： Max.1MHz（增强型快速模式） Max.400KHz（快速模式） Max.100KHz（标准模式）
数据电平	正相输出（默认值：高电平）。
奇偶校验位	无奇偶校验位。
停止位	附加1位（用于ACK接收）。
数据方向	MSB优先

注1：要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器（POM_{xx}）设定 N 沟道漏极开路输出模式（POM_{xx}=1）。

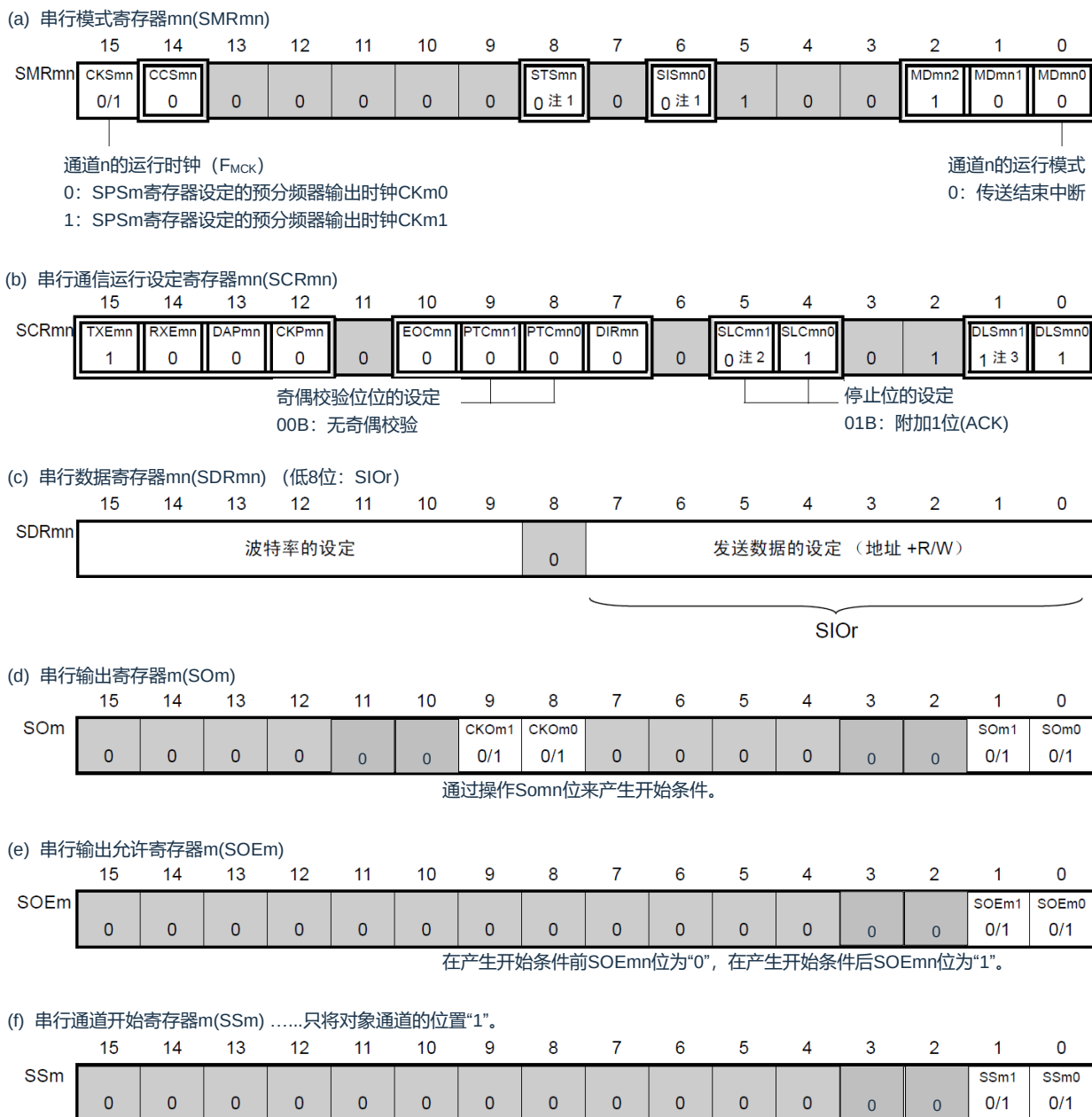
详细内容请参照“第2章 引脚功能”

注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注：m：单元号（m=0）n：通道号（n=0、1）

① 寄存器的设定

图12-94：简易 I²C（IIC00）地址段发送时的寄存器设定内容例子



注1: 只限于 SMR01。

注2: 只限于 SCR00。

注3: 只限于 SCR00寄存器和 SCR01寄存器，其他固定为“1”。

m: 单元号 (m=0) n: 通道号 (n=0、1) r: IIC 号 (r=00)

□ : 在 IIC 模式中为固定设定; ■ : 不能设定 (设定初始值);

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

② 操作步骤

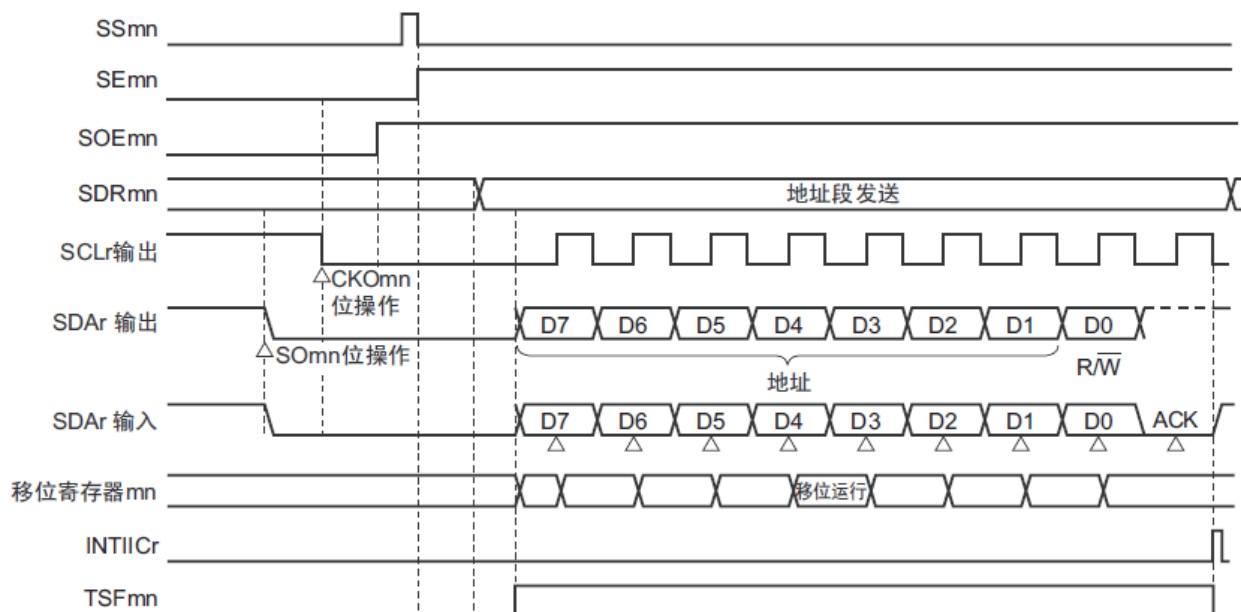
图12-95：地址段发送的初始设定步骤



备注：在初始设定结束时，简易 I²C（IIC00）为禁止输出并且处于运行停止状态。

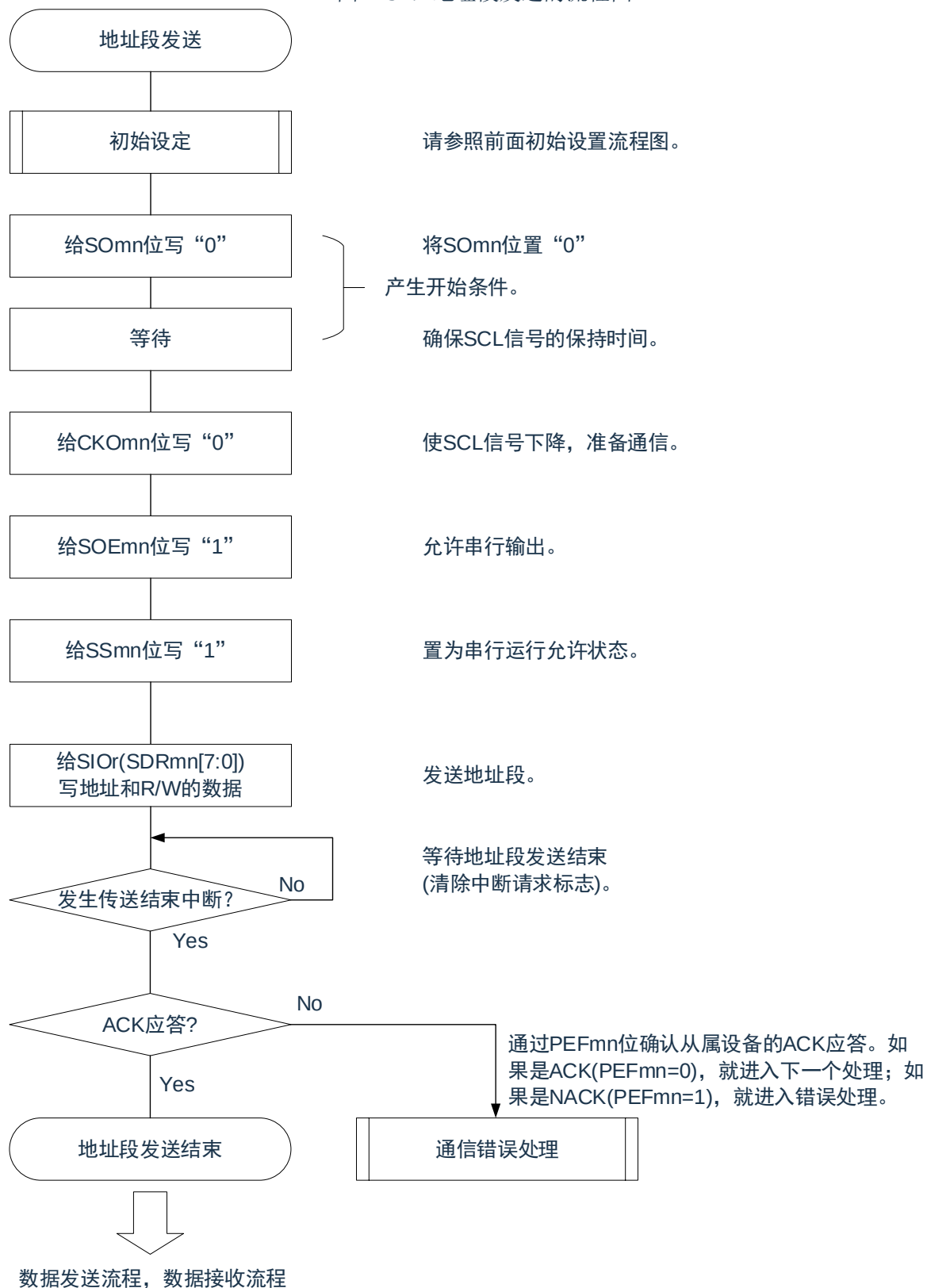
③ 处理流程

图12-96：地址段发送的时序图



备注：m：单元号（m=0）n：通道号（n=0、1）r：IIC号（r=00）

图12-97：地址段发送的流程图



12.8.2 数据发送

数据发送是在发送地址段后给该传送对象（从属设备）发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

表12-43：数据发送

简易I ² C	IIC00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SAU_SDA00 ^{注1}
中断	INTIIC00
	只限于传送结束中断（不能选择缓冲器空中断）。
错误检测标志	ACK错误标志（PEFmn）
传送数据长度	8位
传送速率 ^{注2}	Max.F _{MCK} /4[Hz]（SDRmn[15:9]≥1）F _{MCK} ：对象通道的运行时钟频率但是，必须在I ² C的各模式中满足以下条件： Max.1MHz（增强型快速模式） Max.400KHz（快速模式） Max.100KHz（标准模式）
数据电平	正相输出（默认值：高电平）。
奇偶校验位	无奇偶校验位。
停止位	附加1位（用于ACK接收）。
数据方向	MSB优先

注1：要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器（POMxx）设定 N 沟道漏极开路输出模式（POMxx=1）。详细内容请参照“2.2端口复用功能”和“2.3.2 输出输入复用功能寄存器映射”。

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注：m：单元号（m=0）n：通道号（n=0、1）

① 寄存器的设定

图12-98：简易 I²C 数据发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0 注 1	0	0 注 1	1	0	0	1	0	0

(b) 串行通信运行设定寄存器mn(SCRmn)……在数据发送和接收的过程中不操作TXEmn位和RXEmn位以外的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0 注 2	1	0	1	1 注 3	1

(c) 串行数据寄存器mn(SDRmn)（低8位：SIOr）……在数据发送和接收的过程中只有低8位有效(SIOr)。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定注4							0	发送数据的设定							
									SIOr							

(d) 串行输出寄存器m(SOm)……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1 注 5	0/1 注 5	0	0	0	0	0	0	0/1 注 5	0/1 注 5

(e) 串行输出允许寄存器m(SOEm)……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

(f) 串行通道开始寄存器m(SSm)……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

注1：只限于 SMR01寄存器。

注2：只限于 SCR00寄存器。

注3：只限于 SCR00和 SCR01寄存器，其他固定为“1”。

注4：因为已经在发送地址段时设定，所以不需要设定。

注5：在通信运行过程中，值因通信数据而变。

备注：m：单元号（m=0）n：通道号（n=0、1）r：IIC 号（r=00）

□：在 IIC 模式中为固定设定；■：不能设定（设定初始值）；

×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。

0/1：根据用户的用途置“0”或者“1”。

② 处理流程

图12-99：数据发送的时序图

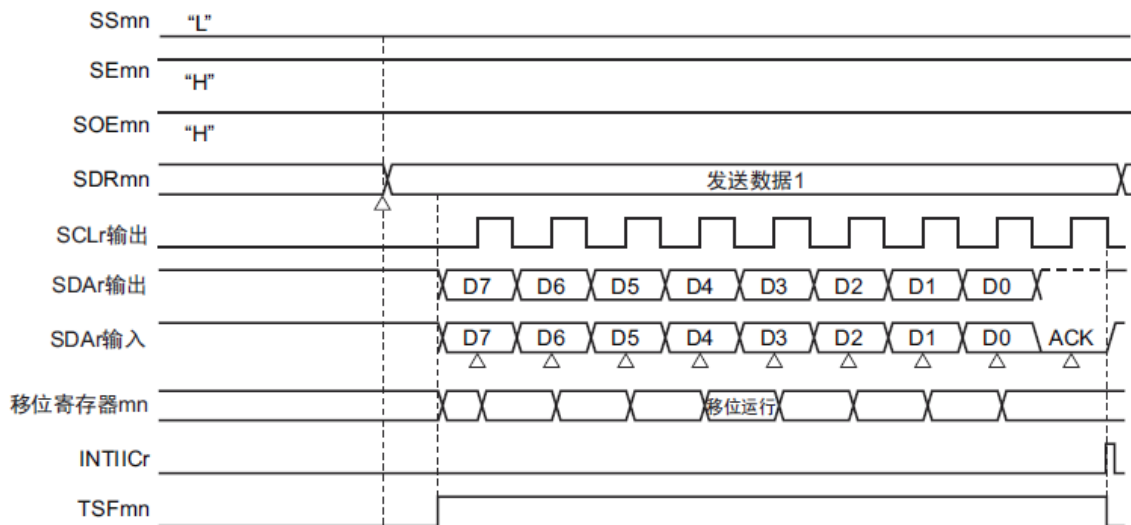
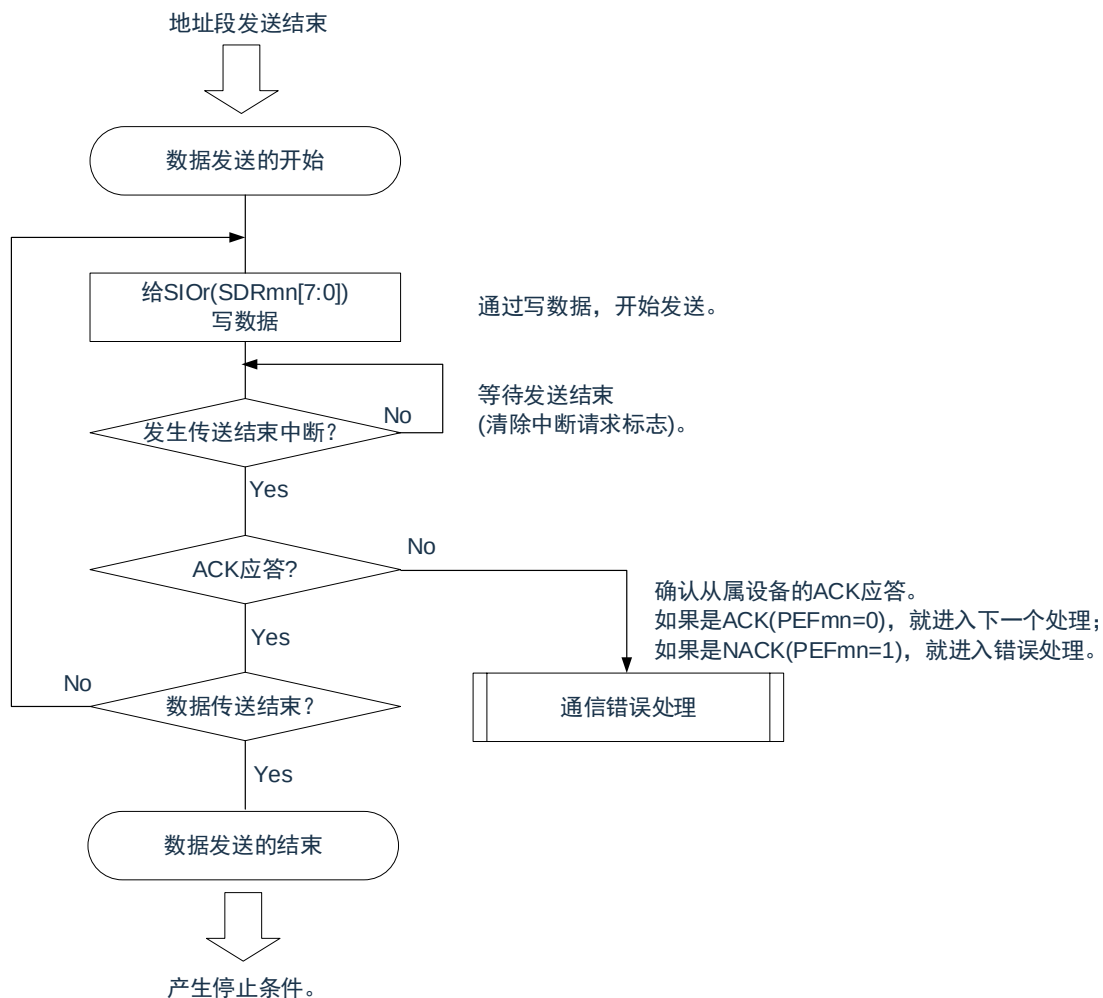


图12-100：数据发送的流程图



12.8.3 数据接收

数据接收是在发送地址段后从传送对象（从属设备）接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

表12-44：数据接收

简易I ² C	IIC00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、 SDI00 ^{注1}
中断	INTIIC00
	只限于传送结束中断（不能选择缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	8位
传送速率 ^{注2}	Max.FMCK/4[Hz]（SDRmn[15:9]≥1） FMCK：对象通道的运行时钟频率但是，必须在I ² C的各模式中满足以下条件： Max.1MHz（增强型快速模式） Max.400KHz（快速模式） Max.100KHz（标准模式）
数据电平	正相输出（默认值：高电平）。
奇偶校验位	无奇偶校验位。
停止位	附加1位（ACK发送）。
数据方向	MSB优先

注1：要通过简易I²C进行通信时，必须通过端口输出模式寄存器（POMxx）设定N沟道漏极开路输出模式（POMxx=1）。

详细内容请参照“2.2端口复用功能”和“2.3.2 输出输入复用功能寄存器映射”。

注2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注：m：单元号（m=0）n：通道号（n=0、1）

① 寄存器的设定

图12-101: 简易 I²C (IIC00) 数据接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0 注1	0	0 注1	1	0	0	1	0	0

(b) 串行通信运行设定寄存器mn(SCRmn)……在数据发送和接收的过程中不操作TXEmn位和RXEmn位以外的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLsmn1	DLsmn0
	0	1	0	0	0	0	0	0	0	0	0 注2	1	0	1	1 注3	1

(c) 串行数据寄存器mn(SDRmn) (低8位: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定注4							0	虚拟发送数据的设定（FFH）							
SIOr																

(d) 串行输出寄存器m(SOm) ……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1 注5	0/1 注5	0	0	0	0	0	0	0/1 注5	0/1 注5

(e) 串行输出允许寄存器m(SOEm) ……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(f) 串行通道开始寄存器m(SSm) ……在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

注1: 只限于 SMR01 寄存器。

注2: 只限于 SCR00 寄存器。

注3: 只限于 SCR00、SCR01 寄存器, 其他固定为“1”。

注4: 因为已经在发送地址段时设定, 所以不需要设定。

注5: 在通信运行过程中, 值因通信数据而变。

备注: m: 单元号 (m=0) n: 通道号 (n=0、1) r: IIC 号 (r=00)

□ : 在 IIC 模式中为固定设定; ■ : 不能设定 (设定初始值);

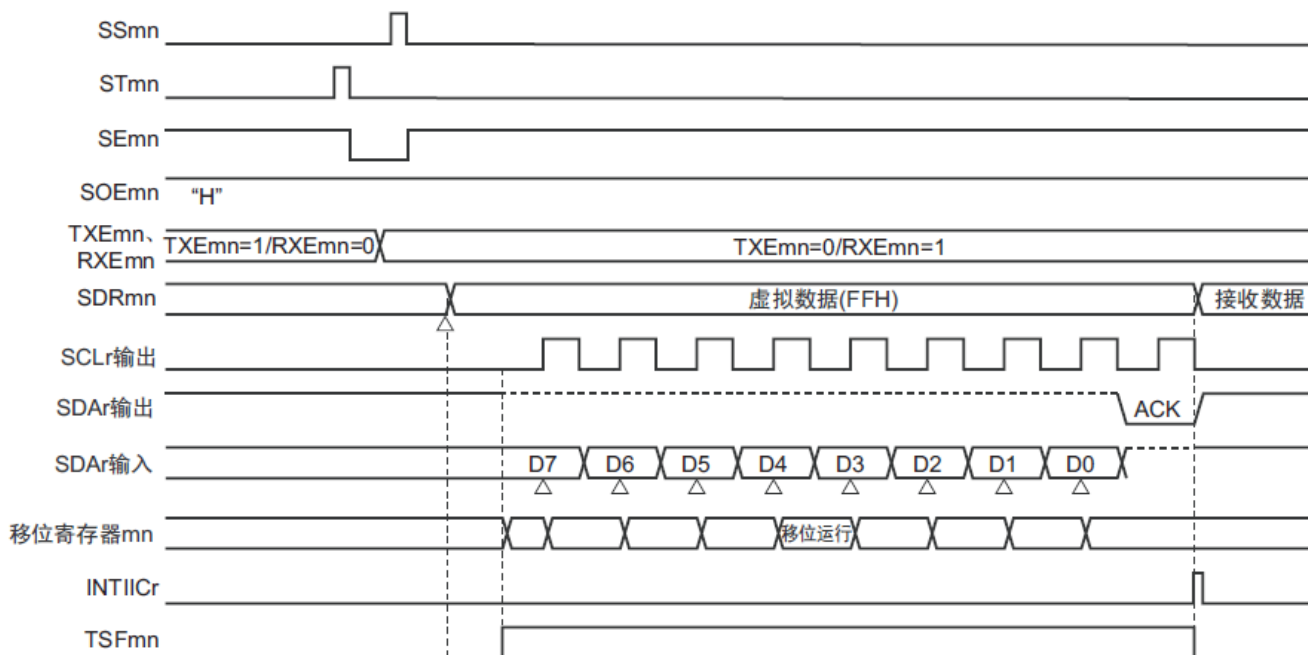
×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

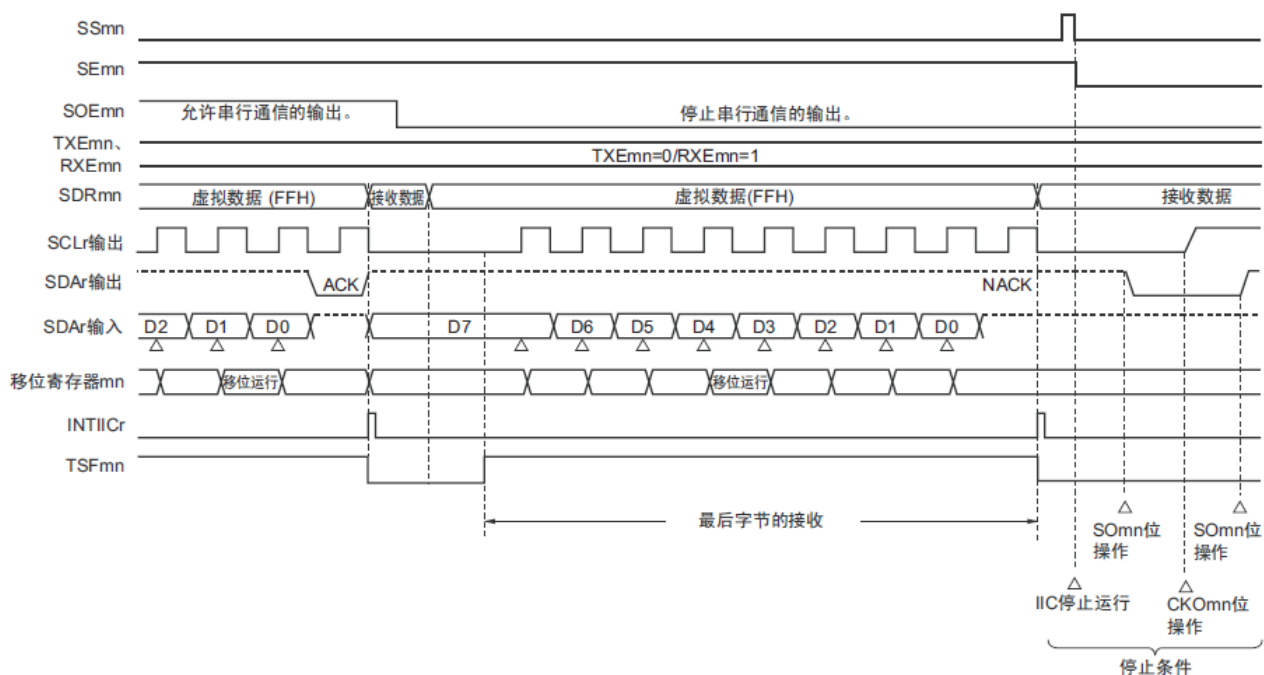
② 处理流程

图12-102：数据接收的时序图

- 开始接收数据的情况

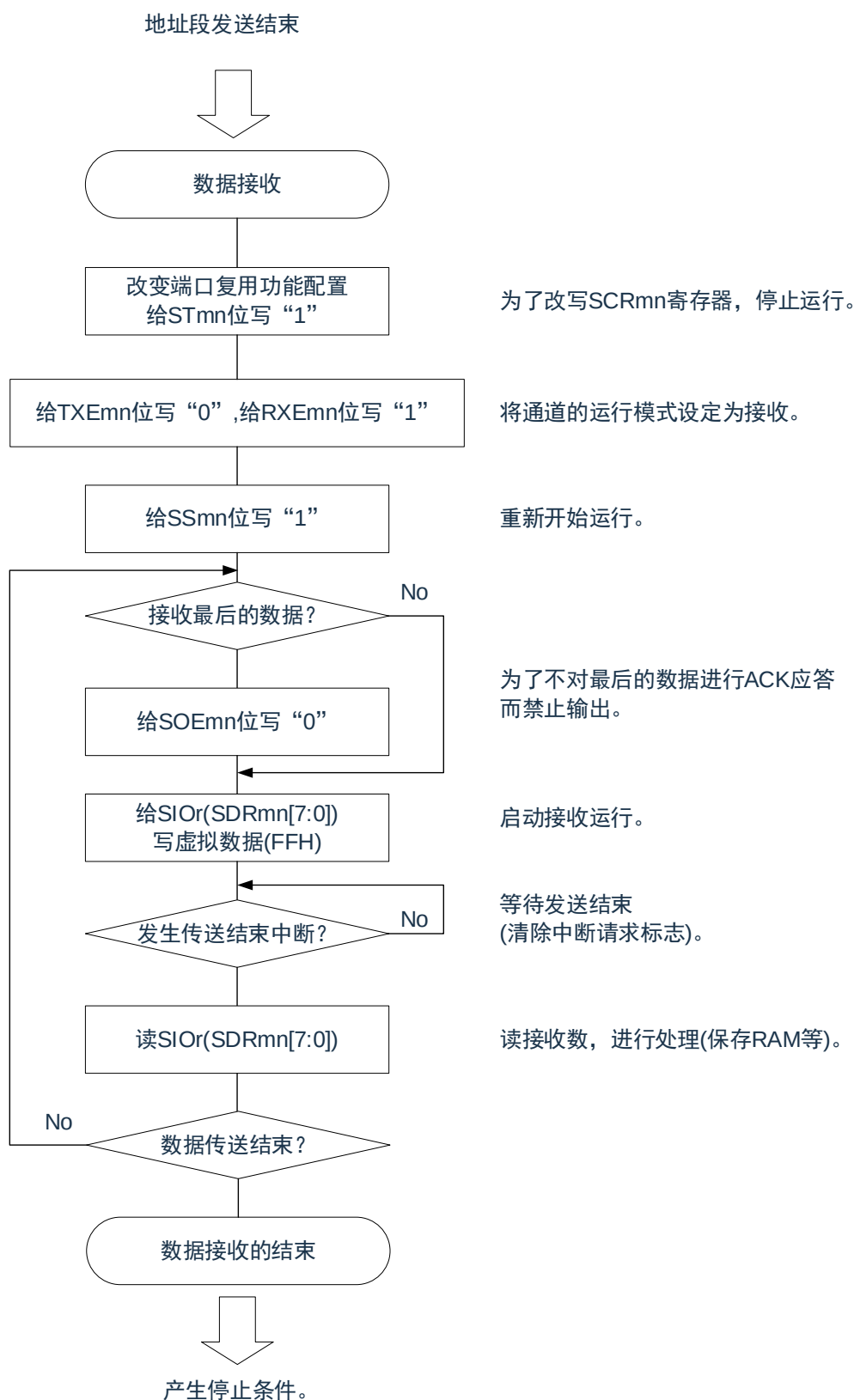


- 接收最后数据的情况



备注：m：单元号（m=0）n：通道号（n=0、1）r：IIC号（r=00）

图12-103：数据接收的流程图

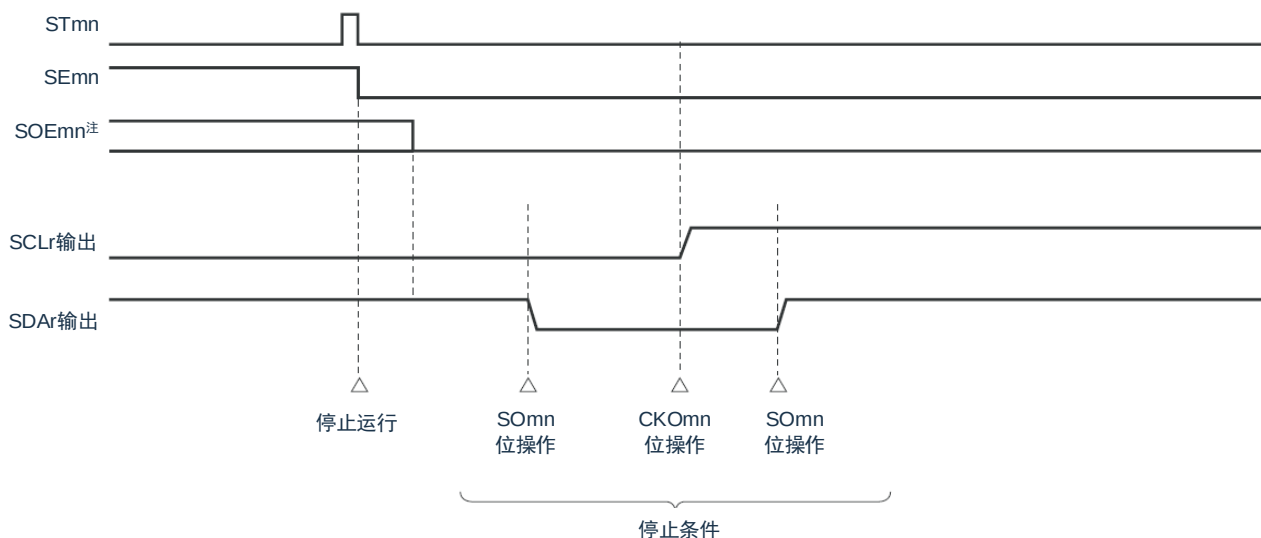


备注：在接收最后的数据时不输出 ACK（NACK）。此后，先通过将串行通道停止寄存器 m（STm）的 STmn 位置“1”来停止运行，然后产生停止条件来结束通信。

12.8.4 停止条件的产生

在与对象从属设备进行了全部数据的发送和接收后，产生停止条件并且释放总线。
处理流程

图12-104：产生停止条件的时序图



注：在接收时，已经在接收最后的数据前将串行输出允许寄存器 m（SOEm）的 SOEmn 位置“0”。

图12-105：产生停止条件的流程图



12.8.5 传送速率的计算

简易 I²C（IIC00）通信的传送速率能用以下计算式进行计算。

$$(\text{传送速率}) = \{ \text{对象通道的运行时钟 (F}_{\text{MCK}}) \text{ 频率} \} / (\text{SDRmn}[15:9] + 1) / 2$$

运行时钟（F_{MCK}）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn 位）。

注意：

禁止将 SDRmn[15:9] 设定为“0000000B”，SDRmn[15:9] 的设定值必须大于等于“0000001B”。简易 I²C 输出的 SCL 信号的占空比为 50%。在 I²C 总线规格中，SCL 信号的低电平宽度大于高电平宽度。因此，如果设定为快速模式的 400kbps 或者增强型快速模式的 1Mbps，SCL 信号输出的低电平宽度就小于 I²C 总线的规格值。必须给 SDRmn[15:9] 设定能满足 I²C 总线规格的值。

因为 SDRmn[15:9] 的值为串行数据寄存器（SDRmn）的 bit15~9 的值（0000001B~1111111B），所以为 1~127。

备注：m：单元号（m=0） n：通道号（n=0、1）

表12-45：简易 I²C 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟（F _{MCK} ） ^{注1}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =24MHz运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	24MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	12MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	6MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	3MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	1.5MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	0.75MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	375KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	187.5KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	93.75KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	46.875KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	23.44KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	11.72KHz
1	0	0	0	0	X	X	X	X	F _{CLK}	24MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	12MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	6MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	3MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	1.5MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	0.75MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	375KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	187.5KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	93.75KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	46.875KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	23.44KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	11.72KHz
上述以外									禁止设定。	

注1：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注：X：忽略

m：单元号（m=0）；n：通道号（n=0、1）。

$F_{MCK}=F_{CLK}=24\text{MHz}$ 时的 I²C 传送速率的设定例子如下所示：

表12-46：I²C 传输速率的设定

I ² C 传送模式 (期待的传送速率)	$F_{CLK}=24\text{MHz}$			
	运行时钟 (F_{MCK})	SDRmn[15:9]	计算的传送速率	与期待的传送速率的误差
100KHz	$F_{CLK}/2$	61	100KHz	3.2%
400KHz	F_{CLK}	30	380KHz	3.2% ^注
1MHz	F_{CLK}	12	0.92MHz	7.7% ^注

注：因为 SCL 信号的占空比为50%，所以不能将误差设定为“0”%左右。

12.8.5.1 在简易I²C（IIC00）通信过程中发生错误时的处理步骤

在简易 I²C（IIC00）通信过程中发生错误时的处理步骤如表12-47和表12-48所示。

表12-47：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn（SDRmn）。 →	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。 →	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

表12-48：在简易 I²C 模式中发生 ACK 错误时的处理步骤

软件操作	硬件状态	备注
读串行状态寄存器mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn（SDIRmn）。 →	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m（STm）的STmn位置“1”。 →	串行通道允许状态寄存器m（SEm）的SEmn位为“0”并且通道n为运行停止状态。	因为没有返回 ACK，所以从属设备没有做好接收的准备。因此，生成停止条件并且释放总线，再次从开始条件开始通信，或者也能生成重新开始条件并且从地址发送开始重新进行。
生成停止条件。		
生成开始条件。		
将串行通道开始寄存器m（SSm）的SSmn位置“1”。 →	串行通道允许状态寄存器m（SEm）的SEmn位为“1”并且通道n为可运行状态。	

备注： m：单元号（m=0） n：通道号（n=0、1） r：IIC号（r=00）

第13章 SPI 通信单元

13.1 简介

串行外设接口（SPI）可与外部器件进行半双工/全双工的同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从器件提供通信时钟（SCK）。该接口还能够的多主模式配置下工作。

它可用于多种用途，包括基于双线的单工同步传输，其中一条可作为双向数据线，或使用 CRC 校验实现可靠通信。

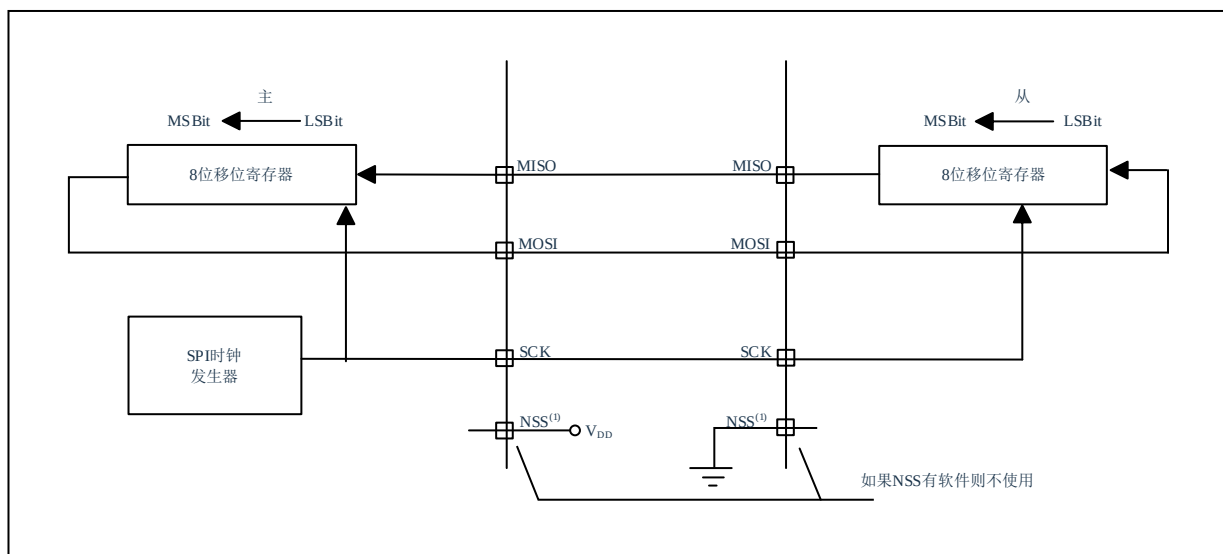
13.2 SPI

13.2.1 SPI 特性

- 基于三条线的全双工同步传输
- 基于双线的单工同步传输，其中一条可作为双向数据线
- 8 位或 16 位传输帧格式选择
- 主模式或从模式操作
- 多主模式功能
- 8 个主模式波特率预分频器（最大值为 $f_{\text{CLK}}/2$ ）
- 从模式频率（最大值为 6MHz）
- 对于主模式和从模式都可实现更快的通信
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- SPI TI 模式
- 用于确保可靠通信的硬件 CRC 功能：
 - 在发送模式下可将 CRC 值作为最后一个字节发送
 - 根据收到的最后一个字节自动进行 CRC 错误校验
- 可触发中断的主模式故障、上溢和 CRC 错误标志
- 具有 DMA 功能的 1 字节发送和接收缓冲器：发送和接收请求

图 13-2 给出了单个主器件和单个从器件之间的互连的基本示例。

图 13-2 单个主器件/单个从器件应用



此处，NSS 引脚配置为输入。

MOSI 引脚连接在一起，MISO 引脚连接在一起。通过这种方式，主器件和从器件之间以串行方式传输数据（最高有效位在前）。通信始终由主器件发起。当主器件通过 MOSI 引脚向从器件发送数据时，从器件同时通过 MISO 引脚做出响应。这是一个数据输出和数据输入都由同一时钟进行同步的 1 全双工通信过程。

从器件选择 (NSS) 引脚管理。

可以使用 SPI_CR1 寄存器中的 SSM 位设置硬件或软件管理从器件选择。

● 软件管理 NSS (SSM = 1)

从器件选择信息在内部由 SPI_CR1 寄存器中的 SSI 位的值驱动。外部 NSS 引脚空闲，可供其它应用使用。

● 硬件管理 NSS (SSM = 0)

根据 NSS 输出配置（SPI_CR1 寄存器中的 SSOE 位），硬件管理 NSS 有两种模式。

- NSS 输出使能（SSM = 0，SSOE = 1）仅当器件在主模式下工作时才使用此配置。当主器件开始通信时，NSS 信号驱动为低电平，并保持到 SPI 被关闭为止。
- NSS 输出禁止（SSM = 0，SSOE = 0）

对于在主模式下工作的器件，此配置允许许多主模式功能。对于设置为从模式的器件，NSS 引脚用作传统 NSS 输入：在 NSS 为低电平时片选该从器件，在 NSS 为高电平时取消对它的片选。

时钟相位和时钟极性。

通过 SPI_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传任何数据时的时钟电平状态。此位对主器件和从器件都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA（时钟相位）位置 1，则 SCK 引脚上的第二个边沿（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）对 MSBit 采样。即，在第二个时钟边沿锁存数据。如果复位 CPHA 位，则 SCK 引脚上的第一个边沿（如果将 CPOL 位置 1，则为下降沿；如果复位 CPOL 位，则为上升沿）对 MSBit 采样。即，在第一个时钟边沿锁存数据。

CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 13-3 显示了在 CPHA 和 CPOL 位的四种组合下的 SPI 传输。可以将该图解释为主器件或从器件时序图，其中 SCK 引脚、MISO 引脚、MOSI 引脚直接连接在主器件和从器件之间。

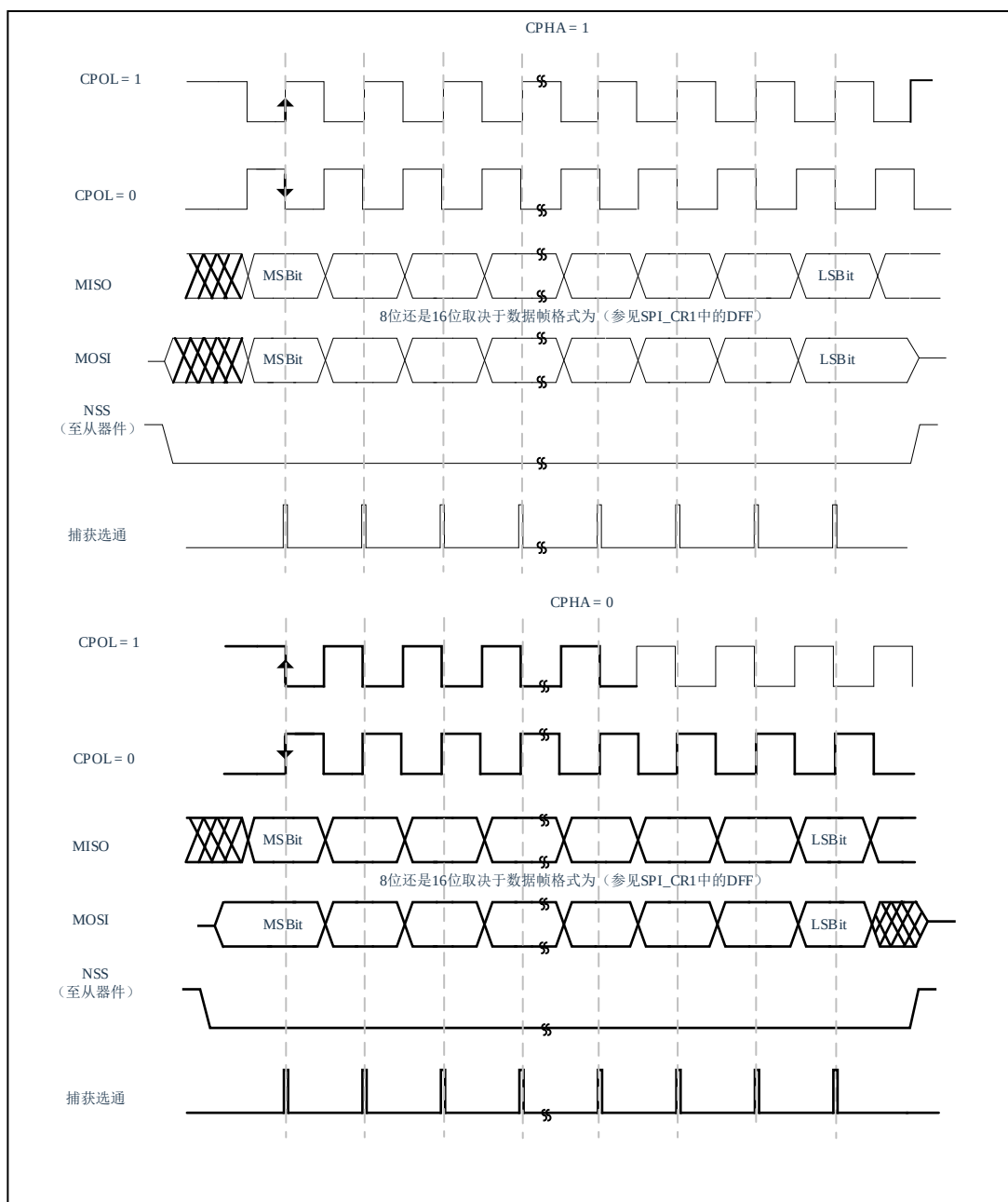
注意：在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来关闭 SPI。

必须以同一时序模式对主器件和从器件进行编程。

SCK 的空闲状态必须与 SPI_CR1 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。

通过 SPI_CR1 寄存器中的 DFF 位选择数据帧格式（8 或 16 位），该格式决定了发送/接收过程中的数据长度。

图 13-3 数据时钟时序图



1. 所示的时序为 SPI_CR1 寄存器中的 LSBFIRST 位复位时的时序。

数据帧格式

移出数据时 MSB 在前还是 LSB 在前取决于 SPI_CR1 寄存器中 LSBFIRST 位的值。每个数据帧的长度均为 8 位或 16 位，具体取决于使用 SPI_CR1 寄存器中的 DFF 位。所选的数据帧格式适用于发送和/或接收。

13.3.2 把 SPI 配置成从器件

在从模式配置中，从 SCK 引脚上接收主器件的串行时钟。SPI_CR1 寄存器的 BR[2:0] 位中设置的值不会影响数据传输率。

注意：建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。在主时钟的第一个边沿到来之前或者正在进行的通信结束之前，从器件的数据寄存器就需要准备就绪。在使能从器件和主器件之前，必须将通信时钟的极性设置为空闲时的时钟电平。

按照下面的步骤把 SPI 模式配置成从模式：

步骤：

1. 设置 DFF 位，以定义 8 或 16 位数据帧格式。
2. 选择 CPOL 和 CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）（参见图 13-3）。要实现正确的数据传输，必须以相同方式在从器件和主器件中配置 CPOL 和 CPHA 位。如果通过 SPI_CR2 寄存器中的 FRF 位选择 TI 模式，则不需要此步骤。
3. 帧格式（MSB 在前或 LSB 在前取决于 SPI_CR1 寄存器中 LSBFIRST 位的值）必须与主器件的帧格式相同。如果选择 TI 模式，则不需要此步骤。
4. 在硬件模式下（参见从器件选择 (NSS) 引脚管理），NSS 引脚在整个字节发送序列期间都必须连接到低电平。在 NSS 软件模式下，将 SPI_CR1 寄存器中的 SSM 位置 1，将 SSI 位清零。如果选择 TI 模式，则不需要此步骤。
5. 将 SPI_CR2 寄存器中的 FRF 位置 1，以选择 TI 模式协议进行串行通信。
6. 将 MSTR 位清零，并将 SPE 位置 1（两个位均在 SPI_CR1 寄存器中）。

在此配置中，MOSI 引脚为数据输入，MISO 引脚为数据输出。

13.3.2.1 发送序列

数据字节在写周期内被并行加载到发送缓冲区中。

当从器件在其 MOSI 引脚上收到时钟信号和数据的最高有效位时，发送序列开始。其余位（8 位数据帧格式中的 7 个位，16 位数据帧格式中的 15 个位）将加载到移位寄存器中。SPI_SR 寄存器中的 TXE 标志在数据从发送缓冲区传输到移位寄存器时置 1，并且在 SPI_CR2 寄存器中的 TXEIE 位置 1 时将生成中断。

13.3.2.2 接收序列

对于接收器，在数据传输完成时：

- 移位寄存器中的数据将传输到接收缓冲区，并且 RXNE 标志（SPI_SR 寄存器）置 1
- 如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，则生成中断。

在出现最后一个采样时钟边沿后，RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收缓冲区中。当读取 SPI_DR 寄存器时，SPI 外设将返回此缓冲值。

通过读取 SPI_DR 寄存器将 RXNE 位清零。

13.3.2.3 从模式下的 SPI TI 协议

在从模式下，SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置从 SPI 串行通信，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，这使用户无需通过设置 SPI_CR1 和 SPI_CR2 寄存器（例如 SSM、SSI、SSOE）来对 NSS 管理进行设置。

在从模式下（图 13-4：TI 模式—从模式，单次传输和图 13-5：TI 模式—从模式，连续传输），使用 SPI 波特率预分频器来控制 MISO 引脚状态切换到高阻态的时刻。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的时间 (t_{release}) 取决于芯片内部电路同步以及通过 SPI_CR1 寄存器的 BR[2:0] 设置的波特率值。具体公式如下：

$$\frac{t_{\text{baud\_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud\_rate}}}{2} + 6 \times t_{\text{pclk}}$$

注意：此特性不适用于 Motorola SPI 通信（FRF 位为 0）。

要在从器件发送器模式下使用错误中断 (ERRIE = 1) 检测 TI 帧错误，必须通过将 SPI_CR1 寄存器中的 BIDIMODE 和 BIDIOE 置 1 来将 SPI 配置为双线单向模式。当 BIDIMODE 置为 0 时，OVR 将置 1，因为始终不会读取数据寄存器从而始终生成溢出错误中断；而当 BIDIMODE 置 1 时，不会接收数据，也不会将 OVR 置 1。

图 13-4 TI 模式-从模式，单次传输

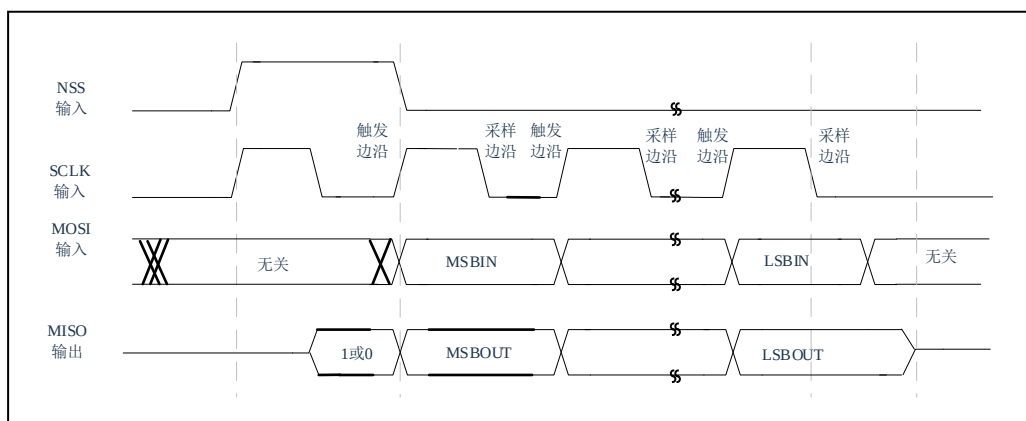
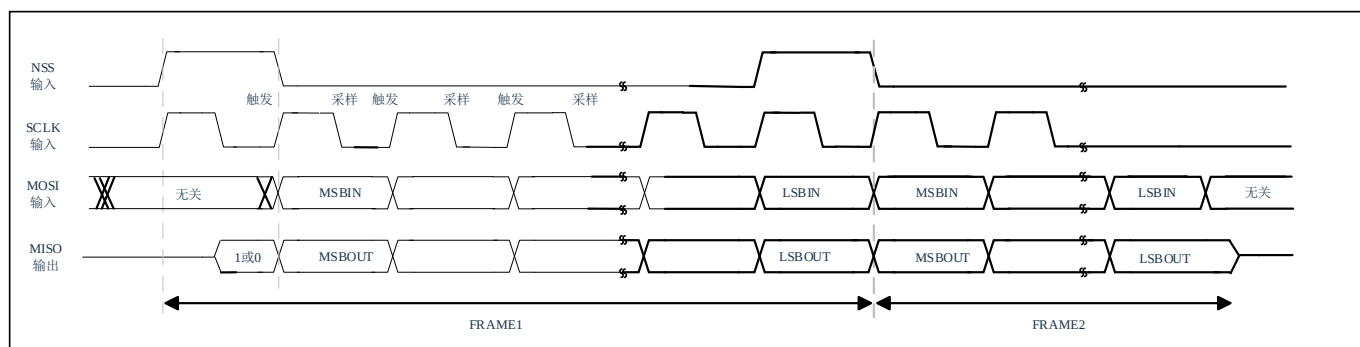


图 13-5 TI 模式-从模式，连续传输



13.3.3 把 SPI 配置成主器件

在主模式配置下，在 SCK 引脚上输出串行时钟。

步骤：

1. 设置 BR[2:0] 位以定义串行时钟波特率（参见 SPI_CR1 寄存器）。
2. 选择 CPOL 和 CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）（参见图 13-3）。如果选择 TI 模式，则不需要此步骤。
3. 设置 DFF 位，以定义 8 或 16 位数据帧格式
4. 配置 SPI_CR1 寄存器中的 LSBFIRST 位以定义帧格式。如果选择 TI 模式，则不需要此步骤。
5. 如果 NSS 引脚配置成输入，在 NSS 硬件模式下，NSS 引脚在整个字节发送序列期间都连接到高电平信号；在 NSS 软件模式下，将 SPI_CR1 寄存器中的 SSM 和 SSI 位置 1。如果 NSS 引脚配置成输出，只应将 SSOE 位置 1。如果选择 TI 模式，则不需要此步骤。
6. 将 SPI_CR2 中的 FRF 位置 1，以选择 TI 协议进行串行通信。
7. MSTR 和 SPE 位必须置 1（仅当 NSS 引脚与高电平信号连接时，这两个位才保持置 1）。

在此配置中，MOSI 引脚为数据输出，MISO 引脚为数据输入。

13.3.3.1 发送序列

在发送缓冲区中写入字节时，发送序列开始。

在第一个位传输期间，数据字节（从内部总线）并行加载到移位寄存器中，然后以串行方式移出到 MOSI 引脚，至于是 MSB 在前还是 LSB 在前则取决于 SPI_CR1 寄存器中的 LSBFIRST 位。TXE 标志在数据从发送缓冲区传输到移位寄存器时置 1，并且在 SPI_CR2 寄存器中的 TXEIE 位置 1 时将生成中断。

13.3.3.2 接收序列

对于接收器，在数据传输完成时：

- 移位寄存器中的数据将传输到接收缓冲区，并且 RXNE 标志置 1
- 如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，则生成中断

在出现最后一个采样时钟边沿时，RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收缓冲区中。当读取 SPI_DR 寄存器时，SPI 外设将返回此缓冲值。

通过读取 SPI_DR 寄存器将 RXNE 位清零。

如果在发送开始后将要发送的下一个数据置于发送缓冲区，则可保持连续的发送流。请注意，仅当 TXE 标志为 1 时，才可以对发送缓冲区执行写操作。

注意：如果与之通信的从器件需要在每个字节传输之间拉低片选信号，必须将该主器件的 NSS 配置成 GPIO，或使用另外 GPIO，通过软件控制从器件的片选。

13.3.3.3 主模式下的 SPI TI 协议

在主模式下，SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置主 SPI 串行通信，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，这使用户无需通过设置 SPI_CR1 和 SPI_CR2 寄存器（例如 SSM、SSI、SSOE）来对 NSS 管理进行设置。

图 13-6：TI 模式—主模式，单次传输和图 13-7：TI 模式—主模式，连续传输）显示了在主模式下选择 TI 模式时的 SPI 主模式通信波形。

图 13-6 TI 模式-主模式，单次传输

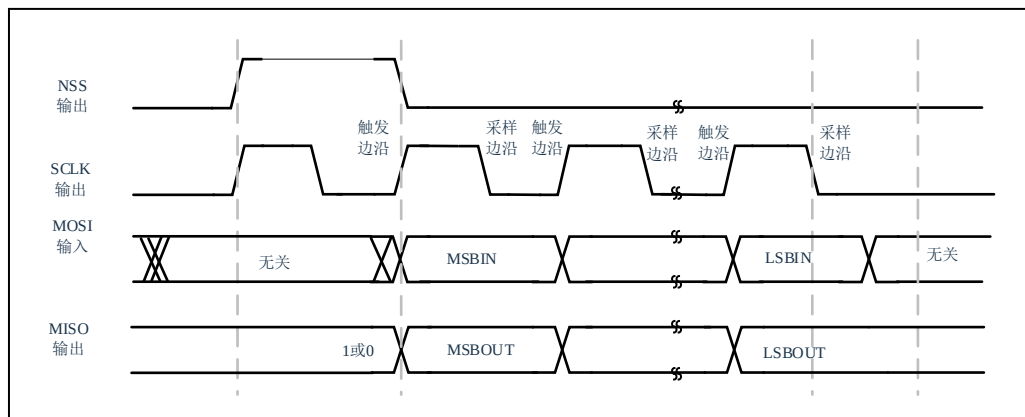
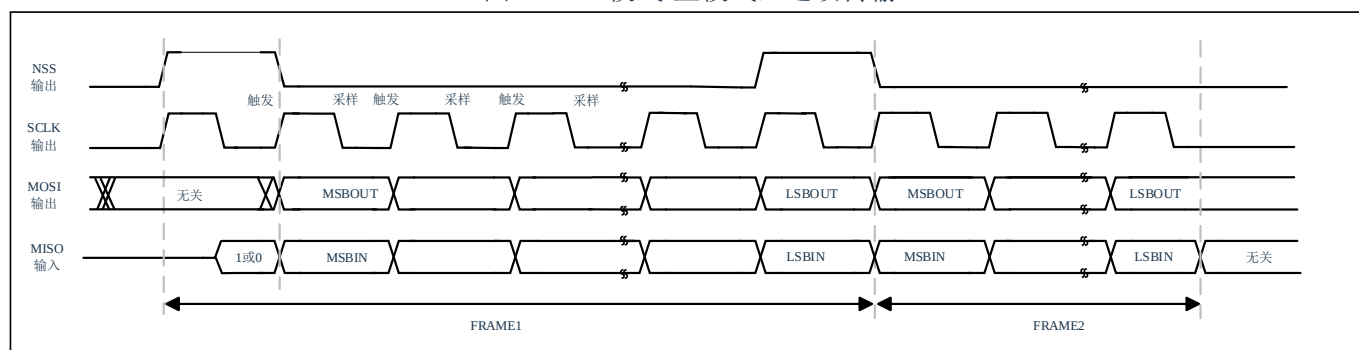


图 13-7 TI 模式-主模式，连续传输



13.3.4 配置 SPI 进行半双工通信

SPI 能够在以下两种配置中以半双工模式工作。

- 1 个时钟和 1 条双向数据线
- 1 个时钟和 1 条数据线（只接收或只发送）

1 个时钟和 1 条双向数据线 (BIDIMODE=1)

可将 SPI_CR1 寄存器中的 BIDIMODE 位置 1 来使能此模式。在此模式下，SCK 用于时钟，MOSI（主模式下）或 MISO（从模式下）用于数据通信。通过 SPI_CR1 寄存器中的 BIDIOE 位来选择传输方向（输入/输出）。当该位置 1 时，数据线为输出，否则为输入。

1 个时钟和 1 条单向数据线 (BIDIMODE=0)

在此模式下，应用程序可使用 SPI 的只发送或只接收功能。

- 只发送模式类似于全双工模式（BIDIMODE=0、RXONLY=0）：在发送引脚（主模式下的 MOSI 或从模式下的 MISO）上发送数据，接收引脚（主模式下的 MISO 或从模式下的 MOSI）可用作通用 IO。在这种情况下，应用程序只需要忽略接收缓冲区（即使读取数据寄存器，它也不包含接收值）。
- 只接收模式下，应用程序可将 SPI_CR2 寄存器中的 RXONLY 位置 1 来关闭 SPI 输出功能。在这种情况下，发送 IO 引脚（主模式下的 MOSI 或从模式下的 MISO）可用于其它用途。

要在只接收模式下开始通信，配置并使能 SPI：

- 在主模式下，通信会立即开始，并在 SPE 位清零且当前接收结束时停止。在此模式下无需读取 BSY 标志。在进行 SPI 通信时，该标志始终置 1。
- 在从模式下，只要 NSS 被拉低（或在 NSS 软件模式下将 SSI 位清零）并且一直有来自主器件的 SCK 输入，SPI 就会继续接收。

13.3.5 数据发送和接收过程

接收和发送缓冲区在接收过程中，数据收到后，先存储到内部接收缓冲区中；而在发送过程中，先将数据存储到内部发送缓冲区中，然后发送数据。

对 SPI_DR 寄存器的读访问将返回接收缓冲值，而对 SPI_DR 寄存器的写访问会将写入的数据存储到发送缓冲区中。

在主模式下启动通信序列

- 在全双工模式下（**BIDIMODE=0** 且 **RXONLY=0**）
 - 将数据写入到 SPI_DR 寄存器（发送缓冲区）时，通信序列启动。
 - 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
 - 同时，将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
- 在单向只接收模式下（**BIDIMODE=0** 且 **RXONLY=1**）
 - 只要 **SPE = 1**，通信序列就立即开始。
 - 只有接收器激活，并且在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
- 在双向模式下，进行发送时（**BIDIMODE=1** 且 **BIDIOE=1**）
 - 将数据写入到 SPI_DR 寄存器（发送缓冲区）时，通信序列启动。
 - 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
 - 不接收任何数据。
- 在双向模式下，进行接收时（**BIDIMODE=1** 且 **BIDIOE=0**）
 - 只要 **SPE=1** 且 **BIDIOE=0**，通信序列就立即开始。
 - 在 MOSI 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
 - 发送器没有激活，因此不会有数据以串行方式移出 MOSI 引脚。

在从模式下启动通信序列

- 在全双工模式下（**BIDIMODE=0** 且 **RXONLY=0**）
 - 当从器件在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
 - 同时，在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主器件启动传输前，软件必须已把要从器件发送的数据写入发送缓冲区。
- 在单向只接收模式下（**BIDIMODE=0** 且 **RXONLY=1**）
 - 当从器件在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
 - 发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。
- 在双向模式下，进行发送时（**BIDIMODE=1** 且 **BIDIOE=1**）
 - 当从器件收到时钟信号，并将发送缓冲区中的第一个位在 MISO 引脚上发送时，通信序列开始。

- 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主器件启动传输前，软件必须已把要从器件发送的数据写入发送缓冲区。
- 不接收任何数据。
- 在双向模式下，进行接收时（**BIDIMODE=1** 且 **BIDIOE=0**）
 - 当从器件在其 MISO 引脚上收到时钟信号和数据的第一位时，通信序列开始。
 - 在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
 - 发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。

13.3.5.1 处理数据发送与接收

将数据从发送缓冲区传输到移位寄存器时，TXE 标志（发送缓冲区为空）置 1。该标志表示内部发送缓冲区已准备好加载接下来的数据。如果 SPI_CR2 寄存器中的 TXEIE 位置 1，可产生中断。通过对 SPI_DR 寄存器执行写操作将 TXE 位清零。

注意：软件必须确保在尝试写入发送缓冲区之前 TXE 标志已置 1。否则，将覆盖之前写入发送缓冲区的数据。将数据从移位寄存器传输到接收缓冲区时，RXNE 标志（接收缓冲区非空）会在最后一个采样时钟边沿置 1。它表示已准备好从 SPI_DR 寄存器中读取数据。如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。通过读取 SPI_DR 寄存器将 RXNE 位清零。对于某些配置，可以在最后一次数据传输期间使用 BSY 标志来等待传输完成。

13.3.5.2 主模式或从模式下的全双工发送和接收过程（BIDIMODE=0 且 RXONLY=0）

软件必须遵循以下步骤来发送和接收数据（参见图 13-8 和图 13-9）：

1. 通过将 SPE 位置 1 来使能 SPI。
2. 将第一个要发送的数据项写入 SPI_DR 寄存器（此操作会将 TXE 标志清零）。
3. 等待至 TXE=1，然后写入要发送的第二个数据项。然后等待至 RXNE=1，读取 SPI_DR 以获取收到的第一个数据项（此操作会将 RXNE 位清零）。对每个要发送/接收的数据项重复此操作，直到第 n-1 个接收的数据为止。
4. 等待至 RXNE=1，然后读取最后接收的数据。
5. 等待至 TXE=1，然后等待至 BSY=0，再关闭 SPI。

此外，还可以使用在 RXNE 或 TXE 标志所产生的中断对应的各个中断子程序来实现该过程。

图 13-8 主/全双工模式（BIDIMODE=0 且 RXONLY=0）下的 TXE/RXNE/BSY 行为（在连续传输的情况下）

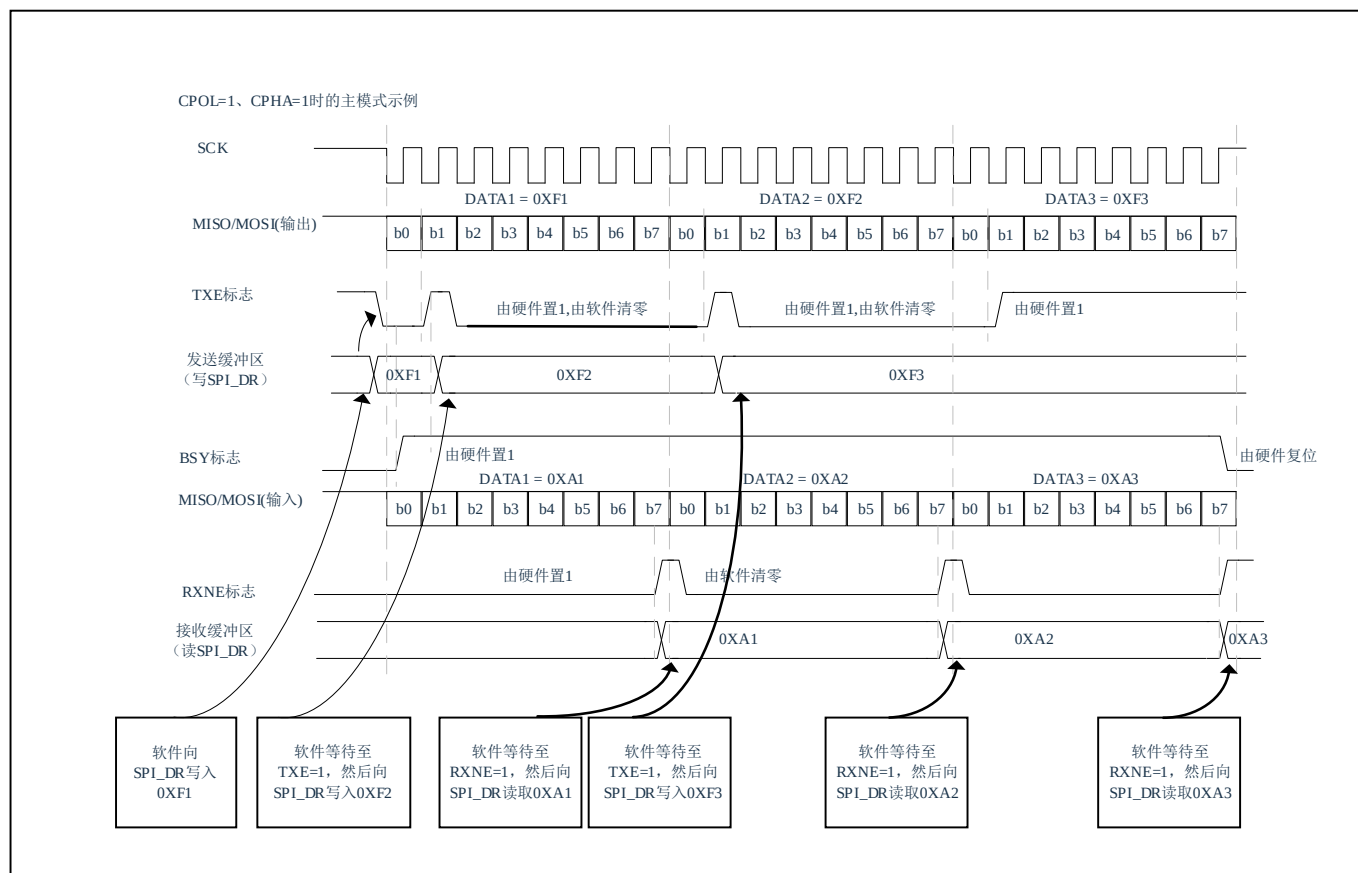
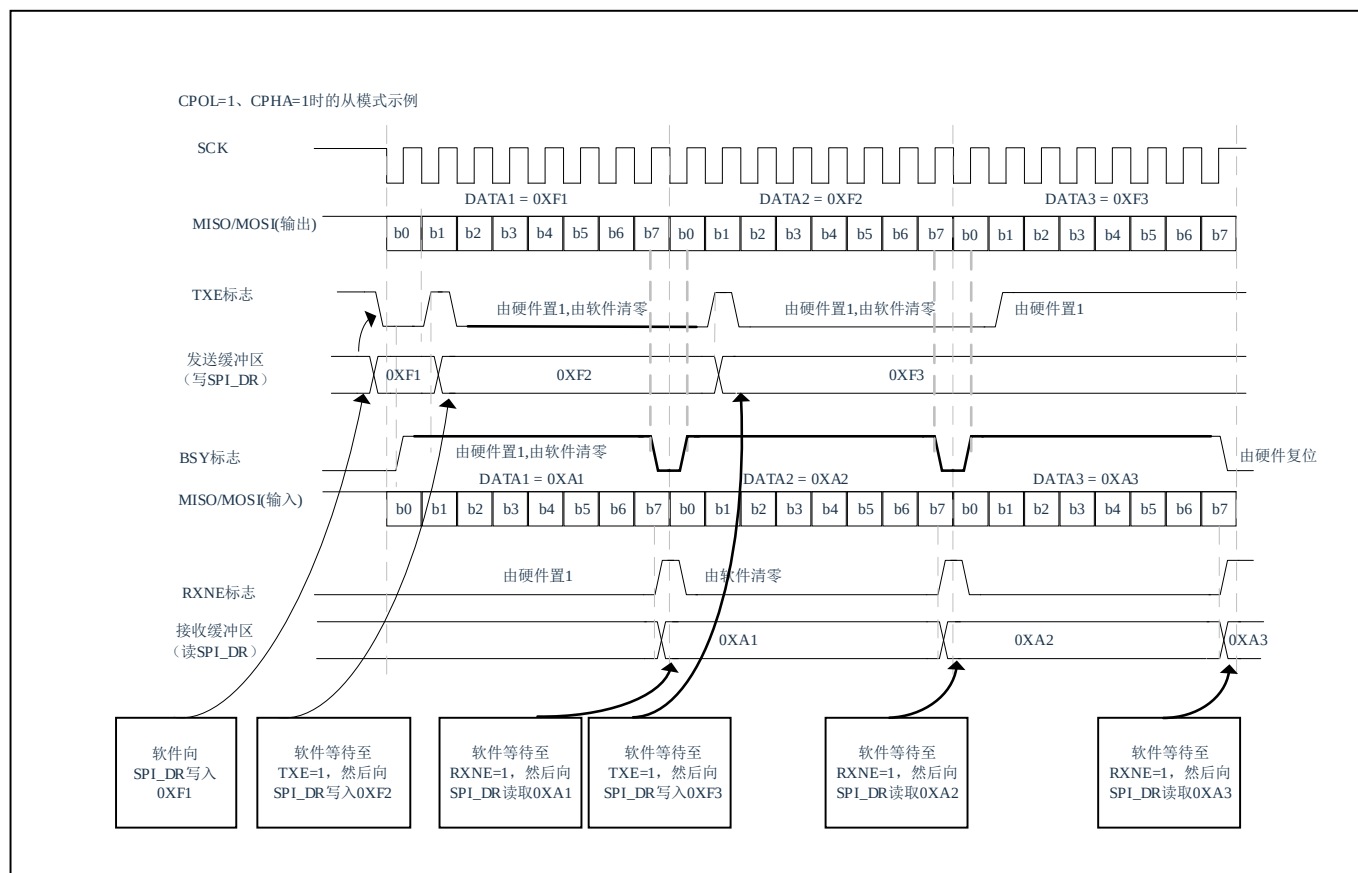


图 13-9 从/全双工模式（ $BIDIMODE=0$ 且 $RXONLY=0$ ）下的 TXE/RXNE/BSY 行为（在连续传输的情况下）



13.3.5.3 只发送模式下的数据发送过程 (BIDIMODE=0、RXONLY=0)

在此模式下，可以按下文所述简化过程，并且可使用 BSY 位等待发送完成（参见图 13-10 和图 13-11）。

1. 通过将 SPE 位置 1 来使能 SPI。
2. 将第一个要发送的数据项写入 SPI_DR 寄存器（此操作会将 TXE 标志清零）。
3. 等待至 TXE=1，然后写入下一个要发送的数据项。对每个要发送的数据项重复此步骤。
4. 将最后一个数据项写入 SPI_DR 寄存器后，等待至 TXE=1，然后等待至 BSY=0，这表示最后的数据发送完成。

此外，还可以使用在 TXE 标志所产生的中断对应的中断子程序来实现该过程。

注意：在不连续通信期间，在对 SPI_DR 执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，在只发送模式下，写入最后的数据后，必须先等待 TXE 位置 1，然后等待 BSY 位清零。在只发送模式下，发送两个数据项后，SPI_SR 寄存器中的 OVR 标志将置 1，因为始终不会读取接收的数据。

图 13-10 主设备只发送模式 (BIDIMODE=0 且 RXONLY=0) 下的 TXE/BSY 行为 (在连续传输的情况下)

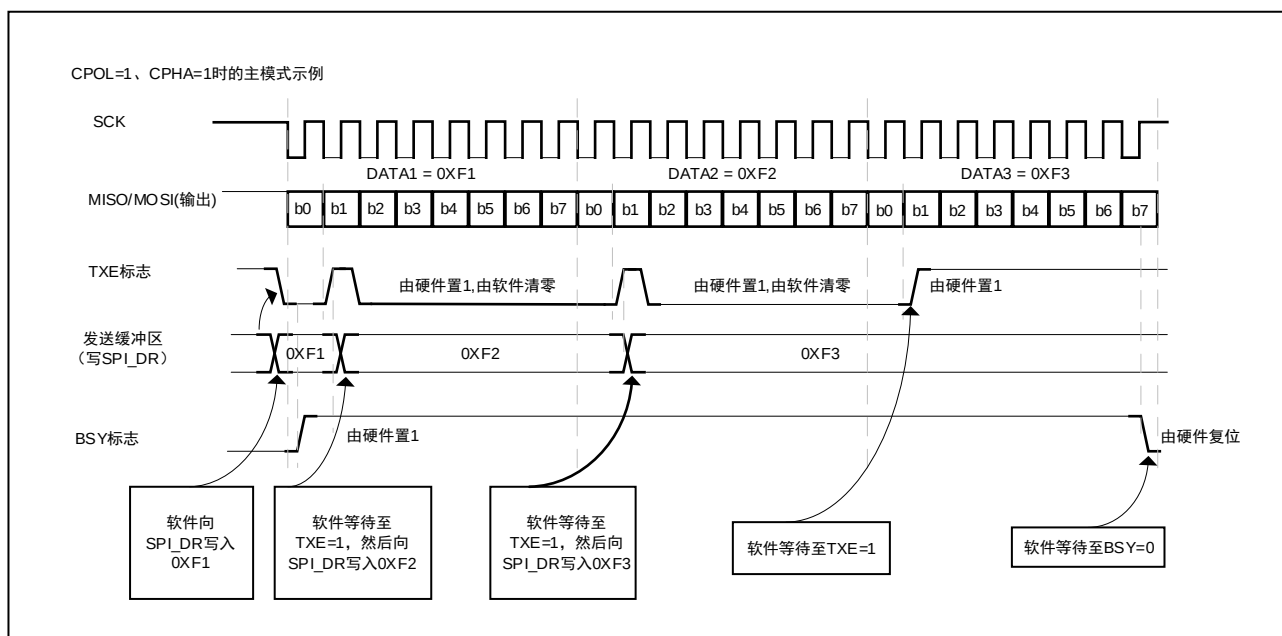
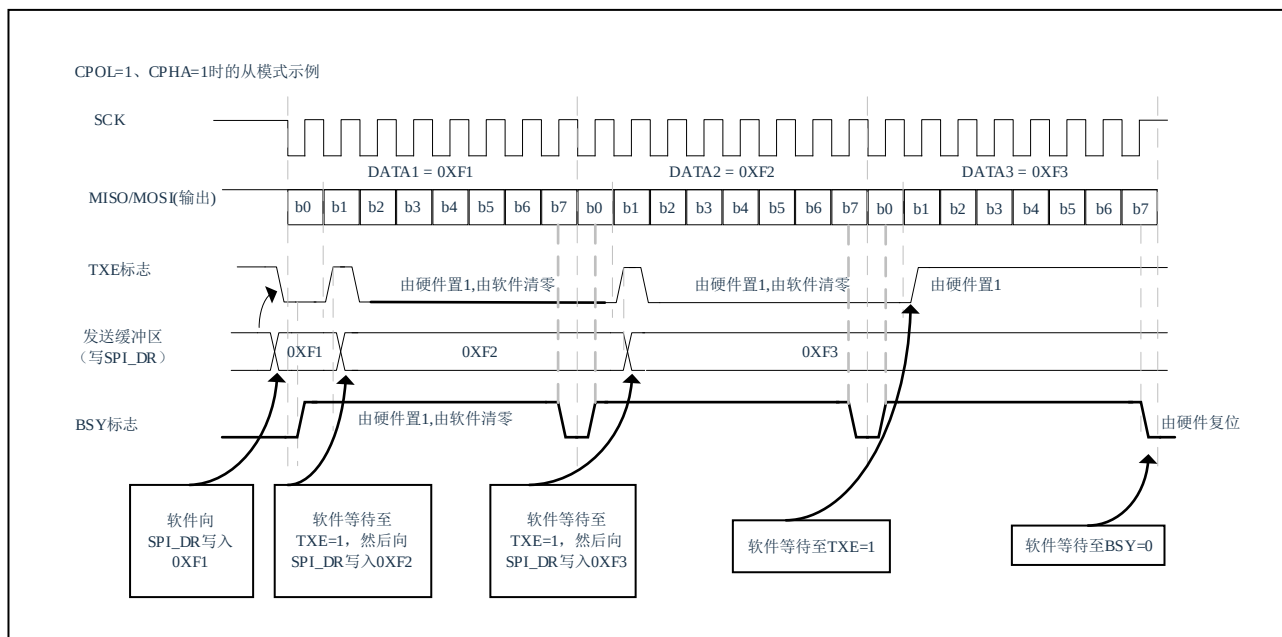


图 13-11 从设备只发送模式（ $BIDIMODE=0$ 且 $RXONLY=0$ ）下的 TXE/BSY 行为（在连续传输的情况下）



13.3.5.4 单线双向模式下的发送过程 (BIDIMODE=1 且 BIDIOE=1)

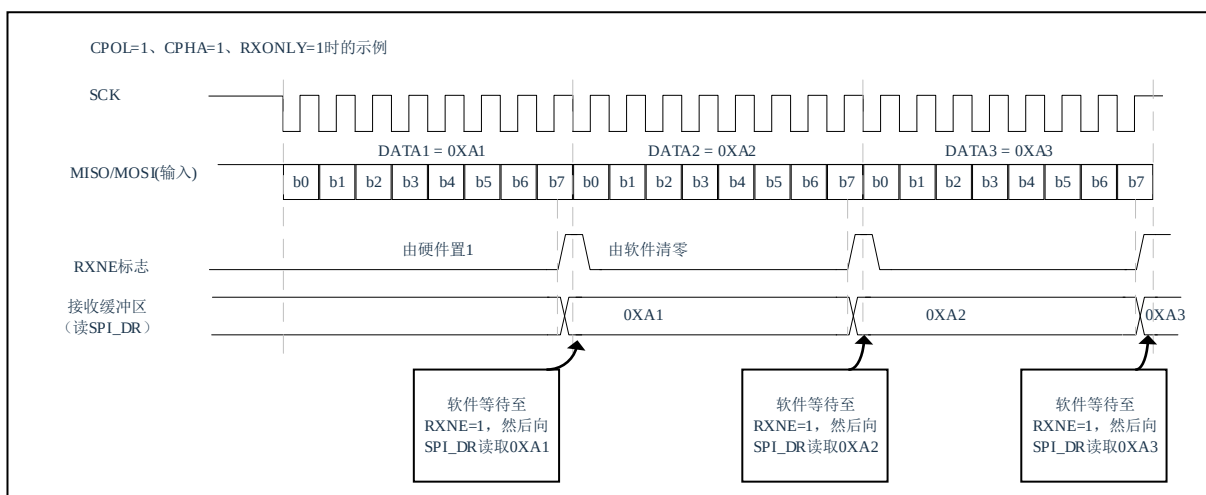
在此模式下，过程与只发送模式下的过程相似，除了在使用 SPI 前必须将 SPI_CR2 寄存器中的 BIDIMODE 位和 BIDIOE 位均置 1。

13.3.5.5 单向只接收过程 (BIDIMODE=0 且 RXONLY=1)

在此模式下，可以按如下所述简化过程（参见图 13-12）：

1. 将 SPI_CR2 寄存器中的 RXONLY 位置 1。
2. 通过将 SPE 位置 1 使能 SPI：
 - a) 在主模式下，这会立即激活 SCK 时钟的产生，并以串行方式接收数据，直到关闭 SPI (SPE=0)。
 - b) 在从模式下，当 SPI 主器件将该从器件的 NSS 驱动为低电平并输出 SCK 时钟时，接收数据。
3. 等待 RXNE=1，然后读取 SPI_DR 寄存器以获取接收的数据（此操作会将 RXNE 位清零）。对每个要接收的数据项重复此操作。此外，还可以使用在 RXNE 标志所产生的中断对应的中断子程序来实现该过程。

图 13-12 只接收模式 (BIDIMODE=0 且 RXONLY=1) 下的 RXNE 行为 (在连续传输的情况下)



13.3.5.6 单线双向模式下的接收过程 (BIDIMODE=1 和 BIDIOE=0)

在此模式下，过程与只接收模式的过程相似，除了在使用 SPI 前必须将 SPI_CR2 寄存器中的 BIDIMODE 位置 1 并将 BIDIOE 清零。

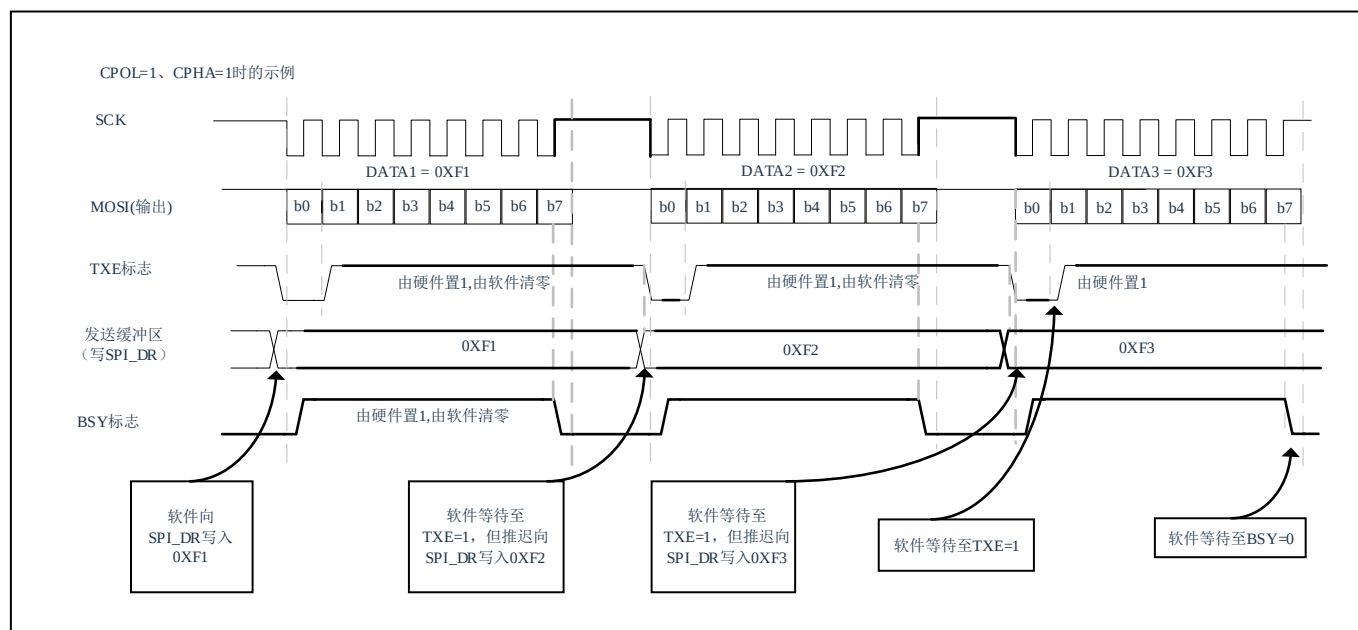
13.3.5.7 连续传输和间断传输

在主模式下发送数据时，如果软件速度快到能在下一个数据传输完成之前响应该数据从数据寄存器传输到移位寄存器所触发的 TXE 中断，并能立即完成再下一个数据的写 SPI_DR 操作，则这种通信称为连续通信。在这种情况下，各数据项之间 SPI 时钟的生成不会间断，并且各数据传输之间不会清零 BSY 位。

相反，如果软件速度不够快，则可能导致通信中断。在这种情况下，各数据传输之间会清零 BSY 位（参见图 13-13）。在主设备仅接收模式 (RXONLY=1) 下，通信始终是连续的，且 BSY 标志始终读为 1。

在从模式下，通信的连续性由 SPI 主器件决定。任何情况下（即使通信是连续的），在各个数据传输之间 BSY 标志都会短暂变为低电平，持续时间为一个 SPI 时钟周期（参见图 13-11）。

图 13-13 发送时 (BIDIMODE=0 且 RXONLY=1) 下的 TXE/BSY 行为 (在间断传输的情况下)



13.3.6 CRC 计算

为确保通信的可靠性，SPI 模块实现了硬件 CRC 功能。针对发送的数据和接收的数据分别实现 CRC 计算。使用可编程的多项式对每个位来计算 CRC。在由 SPI_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿采样每个位来进行计算。

注意：该 SPI 模块提供两种 CRC 计算标准，具体取决于为发送和/或接收选择的数据帧格式：8 位数据采用 CR8，16 位数据采用 CRC16(CRC16/XMODEM)。通过将 SPI_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 的计算。此操作将复位 CRC 寄存器（SPI_RXCRCR 和 SPI_TXCRCR）。在全双工或只发送模式下，如果传输由软件（CPU 模式）管理，则在将最后传输的数据写入 SPI_DR 后，必须立即对 CRCNEXT 位执行写操作。最后一次数据传输结束时，将发送 SPI_TXCRCR 值。

在只接收模式下，如果传输由软件（CPU 模式）管理，则在接收到倒数第二个数据后，必须对 CRCNEXT 位执行写操作。在收到最后一个数据后会收到 CRC，然后执行 CRC 校验。

如果传输过程中出现数据损坏，则在数据和 CRC 传输结束时，SPI_SR 寄存器中的 CRCERR 标志将置 1。

如果发送缓冲区中存在数据，则只有在发送数据字节后才会发送 CRC 值。在 CRC 发送期间，CRC 计算器处于关闭状态且寄存器值保持不变。

可通过以下步骤使用 CRC 进行 SPI 通信：

1. 对 CPOL、CPHA、LSBFirst、BR、SSM、SSI 和 MSTR 值进行编程。
2. 对 SPI_CRCPR 寄存器中的多项式进行编程。
3. 通过将 SPI_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。此操作还会将 SPI_RXCRCR 和 SPI_TXCRCR 寄存器清零。
4. 通过将 SPI_CR1 寄存器中的 SPE 位置 1 使能 SPI。
5. 启动并维持通信，直到只剩下一个字节或半字未发送或接收。
 - 在全双工或只发送模式下，如果传输由软件管理，则在向发送缓冲区写入最后一个字节或半字后，将 SPI_CR1 寄存器中的 CRCNEXT 位置 1，以表示在发送完最后一个字节后将发送 CRC。
 - 在只接收模式下，在接收倒数第二个数据后，立即将 CRCNEXT 位置 1，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在 CRC 传输期间，CRC 计算将冻结。
6. 传输完最后一个字节或半字后，SPI 进入 CRC 传输和校验阶段。在全双工模式或只接收模式下，将接收的 CRC 与 SPI_RXCRCR 值进行比较。如果两个值不匹配，则 SPI_SR 中的 CRCERR 标志将置 1，并且在 SPI_CR2 寄存器中的 ERRIE 位置 1 时会产生中断。

注意：当 SPI 处于从模式时，注意只能在时钟稳定（即，时钟处于空闲电平）时使能 CRC 计算。否则，可能导致 CRC 计算错误。因为，只要 CRCEN 位置 1，无论 SPE 位的值如何，只要有时钟输入，CRC 计算器就开始工作。

在 SPI 通信时钟频率较高的情况下，发送 CRC 时务必小心。由于在 CRC 传输阶段 CPU 应尽可能空闲，因此禁止在 CRC 发送阶段调用函数，以便避免最后的数据和 CRC 接收出错。实际上，在发送/接收最后的数据之前必须对 CRCNEXT 位执行写操作。

SPI 通信时钟频率较高时，建议使用 DMA 模式来避免由于 CPU 访问影响 SPI 带宽而导致 SPI 速度性能下降。

如果将器件配置为从器件，并且使用 NSS 硬件模式，则需要在数据阶段和 CRC 阶段之间将 NSS 引脚保持为低电平。

当 SPI 配置为从模式并且 CRC 功能已使能时，即使 NSS 引脚上为高电平，也会进行 CRC 计算。例如，在多从模式环境下可能出现这种情况，此时通信主器件会交替寻址从器件。

在对从器件片选的切换期间内，应在主器件和从器件两端同时将 CRC 值清零，以重新同步主从双方的 CRC 计算。要将 CRC 清零，请按以下步骤操作：

1. 关闭 SPI (SPE = 0)
2. 将 CRCEN 位清零
3. 将 CRCEN 位置 1
4. 使能 SPI (SPE = 1)

13.3.7 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

13.3.7.1 发送缓冲区为空 (TXE)

此标志置 1 时，表示发送缓冲区为空，可以将待发送的下一个数据加载到缓冲区中。对 SPI_DR 寄存器执行写操作时，将清零 TXE 标志。

13.3.7.2 接收缓冲区非空 (RXNE)

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPI_DR 时，将清零该标志。

BUSY 标志。

BSY 标志由硬件置 1 和清零（对此标志执行写操作没有任何作用）。BSY 标志用于指示 SPI 通信的状态。

BSY 置 1 时，表示 SPI 正忙于通信。

如果软件要关闭 SPI 并进入停止模式（或关闭外设时钟），可使用 BSY 标志检测传输是否结束以避免破坏最后一个数据的传输。为此，必须严格遵循下述步骤。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

传输开始时，BSY 标志将置 1。

在以下情况硬件将清零该标志：

- 传输完成时（主模式下的连续通信除外）
- 关闭 SPI 时

当通信不连续时，BSY 标志在各通信之间处于低电平。

当通信连续时：

- 在主模式下，BSY 标志在所有传输期间均保持高电平
- 在从模式下，BSY 标志在各传输之间的一个 SPI 时钟周期内变为低电平

注意：请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。

13.3.8 关闭 SPI

传输终止时，应用可通过关闭 SPI 外设来停止通信。这通过将 SPE 位清零来完成。

对于某些配置，在传输进行时关闭 SPI 并进入停止模式会导致当前传输受损，并且/或者 BSY 标志可能不可靠。

为避免上述后果，建议在关闭 SPI 时按以下步骤操作：

在主模式或全双工从模式（BIDIMODE=0、RXONLY=0）下：

1. 等待 RXNE=1 以接收最后的数据
2. 等待 TXE=1
3. 然后等待 BSY=0
4. 关闭 SPI (SPE=0)，最后进入停止模式（或关闭外设时钟）

在主模式或单向只发送从模式（BIDIMODE=0、RXONLY=0）或双向通信发送模式（BIDIMODE=1、BIDIOE=1）下：

在最后的数写入 SPI_DR 寄存器后：

1. 等待 TXE=1
2. 然后等待 BSY=0
3. 关闭 SPI (SPE=0)，最后进入停止模式（或关闭外设时钟）

在单向只接收主模式（MSTR=1、BIDIMODE=0、RXONLY=1）或双向通信接收模式（MSTR=1、BIDIMODE=1、BIDIOE=0）下：

必须以特殊方式管理这种情况，以避免多余的 SPI 数据传输。以下序列仅适用于 SPI Motorola 配置（FRF 位置 0）：

1. 等待倒数第二个数据（第 n-1 个）对应的 RxNE 标志置位
2. 然后等待一个 SPI 时钟周期（使用软件循环），才能关闭 SPI (SPE=0)
3. 再等待最后的 RXNE=1，然后进入停止模式（或关闭外设时钟）

当 SPI 配置为 TI 模式（FRF 位置 1）时，必须按以下步骤操作以避免当关闭 SPI 时在 NSS 上产生不需要的脉冲：

1. 等待倒数第二个数据（第 n-1 个）对应的 RxNE 标志置位
2. 在以下窗口帧中使用软件循环关闭 SPI (SPE = 0)：
 - 至少一个 SPI 时钟周期后，
 - LSB 数据开始传输前。

注意：在双向通信接收主模式（MSTR=1、BDM=1、BDOE=0）下，BSY 标志在传输期间保持低电平。

在只接收从模式（MSTR=0、BIDIMODE=0、RXONLY=1）或双向通信接收模式（MSTR=0、BIDIMODE=1、BIDIOE=0）下：

1. 可以随时关闭 SPI（写入 SPE=0）：当前传输完成后，SPI 才被真正关闭
2. 之后，如果要进入停止模式，则必须首先等待至 BSY = 0，然后才能进入停止模式（或关闭外设时钟）。

13.3.9 使用 DMA（直接存储器寻址）进行 SPI 通信

要以最大速度工作，需要给 SPI 不断提供要发送的数据，并及时读取接收缓冲区中的数据，以避免上溢。为加速传输，SPI 提供了 DMA 功能，以实现简单的请求/应答协议。当使能 SPI_CR2 寄存器中的使能位时，将请求 DMA 访问。发送缓冲区和接收缓冲区会发出各自的 DMA 请求（参见图 13-14 和图 13-15）：

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。DMA 随后对 SPI_DR 寄存器执行写操作（此操作会将 TXE 标志清零）。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。DMA 随后对 SPI_DR 寄存器执行读操作（此操作会将 RXNE 标志清零）。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，OVE 标志会置 1，因为未读取接收的数据。

当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。在发送模式下，DMA 完成了所有要发送数据的传输（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 BSY 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 或进入停止模式前必须执行此步骤，以避免损坏最后一次数据的发送。软件必须首先等待 TXE=1，再等待 BSY=0。

注意：在不连续通信期间，在对 SPI_DR 执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，必须在写入最后的数据后，先等待 TXE=1，再等待 BSY=0。

图 13-14 使用 DMA 进行发送

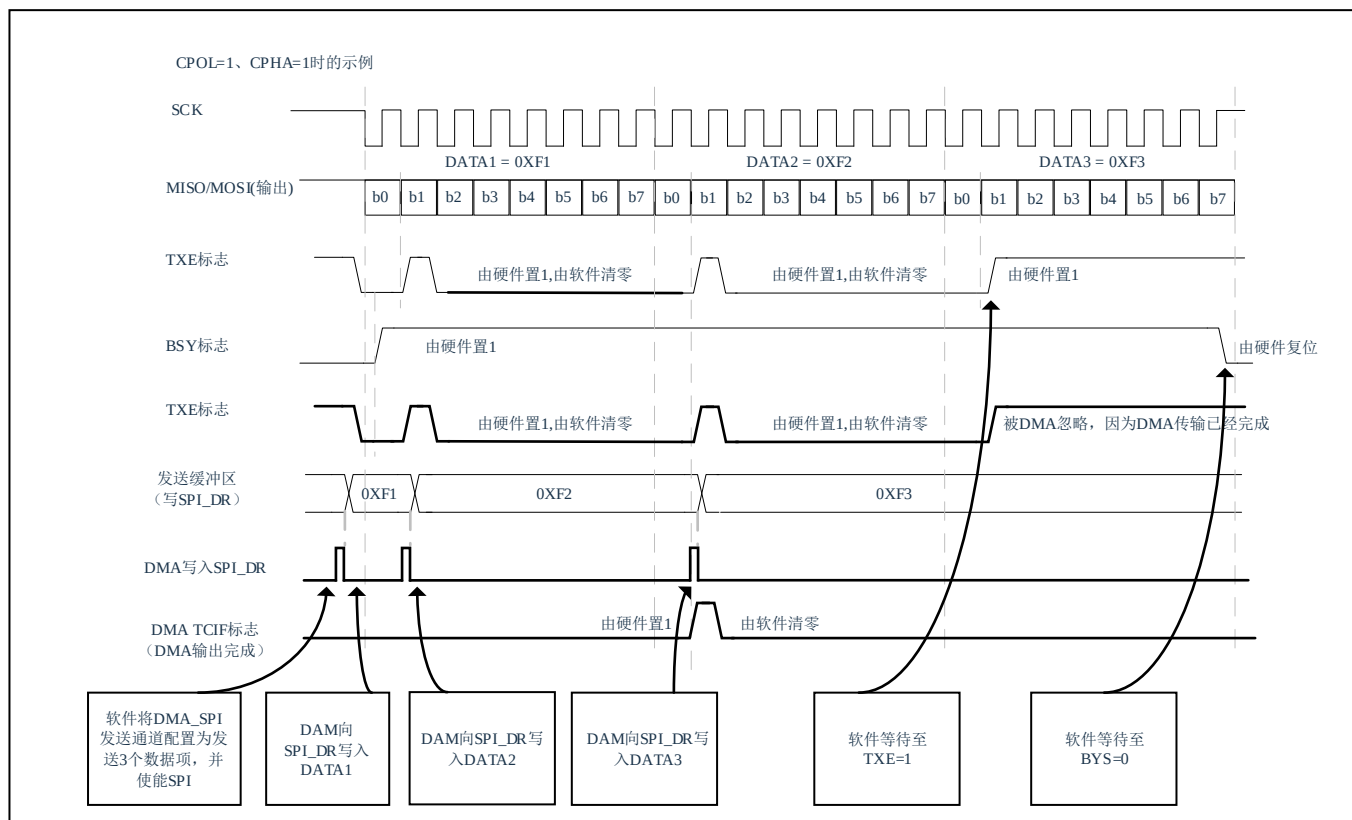
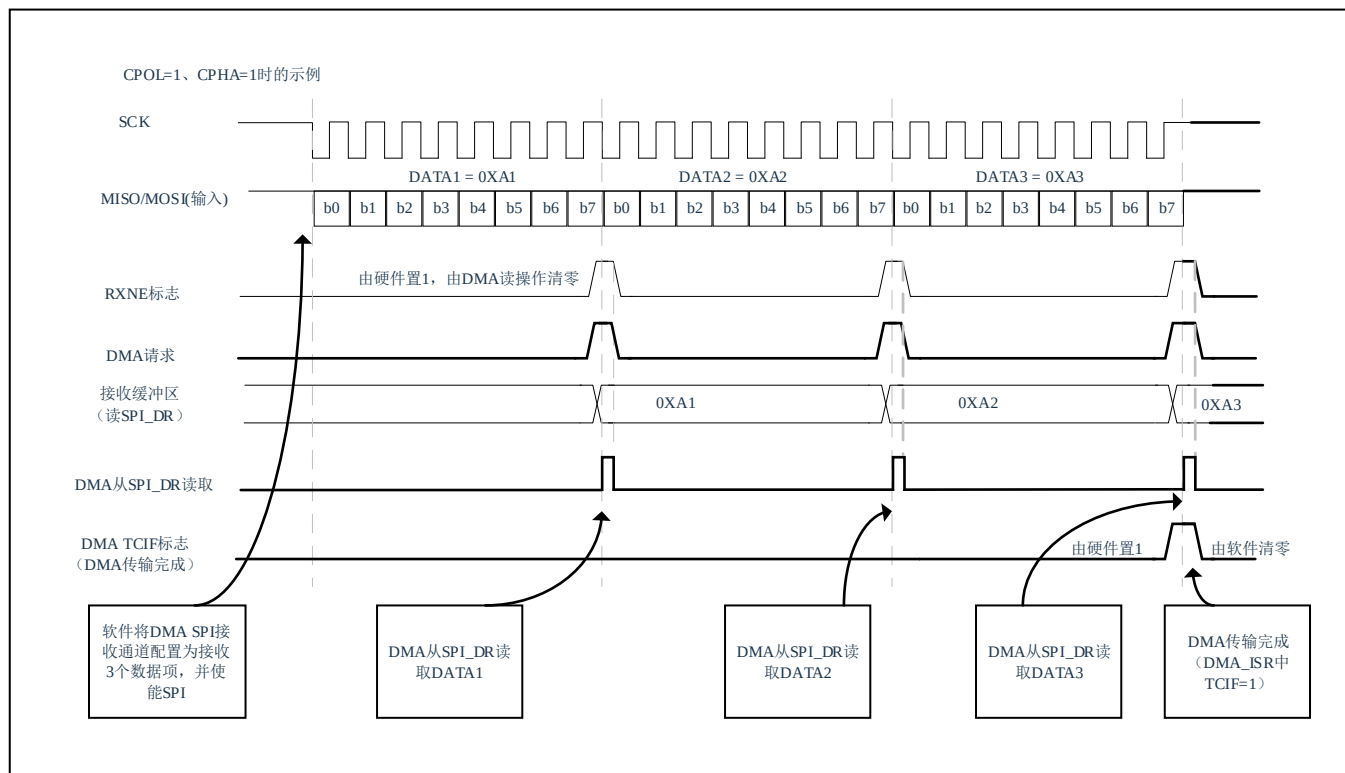


图 13-15 使用 DMA 进行接收



13.3.9.1 DMA 功能与 CRC

当使能的 SPI 通信支持 CRC 通信和 DMA 模式时，在通信结束时会自动发送和接收 CRC，无需使用 CRCNEXT 位。接收 CRC 后，必须在 SPI_DR 寄存器中读取 CRC，以将 RXNE 标志清零。

如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPI_SR 寄存器中的 CRCERR 标志将置 1。

13.3.10 错误标志

主模式故障 (MODF)

当主器件的 NSS 引脚拉低 (NSS 硬件模式下) 或 SSI 位为 0 (NSS 软件模式下) 时, 会发生主模式故障, 这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 外设:

- 如果 ERRIE 位置 1, MODF 位将置 1, 并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出, 并关闭 SPI 接口。
- MSTR 位清零, 从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零:

1. 在 MODF 位置 1 时, 对 SPI_SR 寄存器执行读或写访问。
2. 然后, 对 SPI_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突, 必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后, 可以将 SPE 和 MSTR 位恢复到原始状态。

作为安全措施, 硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。

在从器件中, 不能将 MODF 位置 1。但是, 在多主模式配置中, 器件可在 MODF 位置 1 时处于从模式。在这种情况下, MODF 位指示系统控制可能存在多主模式冲突。可使用中断程序从此状态完全恢复, 方法是执行复位或返回到默认状态。

溢出错误

当主器件发送完数据字节, 而从器件尚未将上一个收到的数据所产生的 RXNE 位清零时, 将出现溢出情况。出现溢出错误时:

- OVR 位置 1 并在 ERRIE 位置 1 时生成一个中断。

在这种情况下, 接收器缓冲区内容不会被来自主器件的新数据更新。读取 SPI_DR 寄存器将返回此字节。主器件后续发送的所有其它字节均将丢失。

依次读取 SPI_DR 寄存器和 SPI_SR 寄存器可将 OVR 清除。

CRC 错误

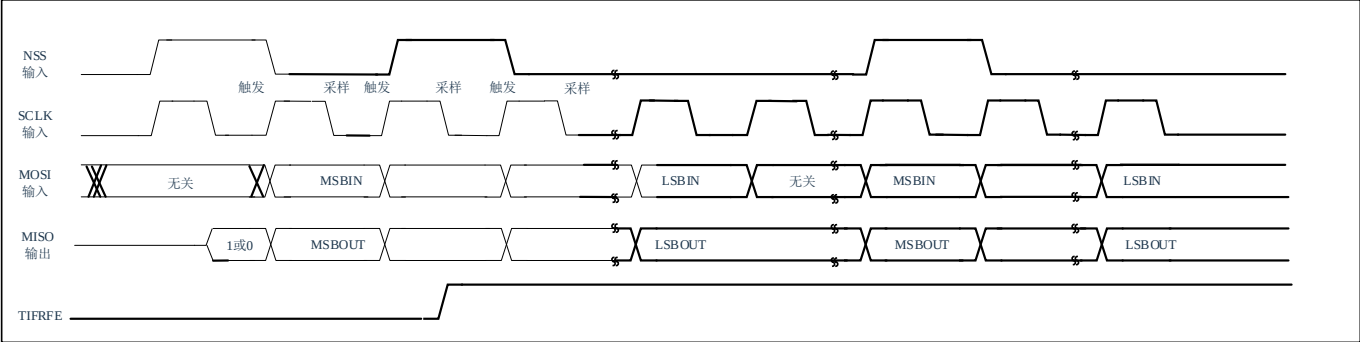
当 SPI_CR1 寄存器中的 CRCEN 位置 1 时, 此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPI_RXCRCR 的值不匹配, SPI_SR 寄存器中的 CRCERR 标志将置 1。

TI 模式帧格式错误

如果 SPI 在从模式下工作, 并配置为符合 TI 模式协议, 则在持续通信期间出现 NSS 脉冲时, 将检测到 TI 模式帧格式错误。出现此错误时, SPI_SR 寄存器中的 FRE 标志将置 1。发生错误时不会关闭 SPI, 但会忽略 NSS 脉冲, 并且 SPI 会等待至下一个 NSS 脉冲, 然后再开始新的传输。由于错误检测可能导致丢失两个数据字节, 因此数据可能会损坏。

读取 SPI_SR 寄存器时, 将清零 FRE 标志。如果 ERRIE 位置 1, 则检测到帧格式错误时将产生中断。在这种情况下, 由于无法保证数据的连续性, 应关闭 SPI, 并在重新使能从 SPI 后, 由主器件重新发起通信。

图 13-16 TI 模式帧格式错误检测



13.3.11 SPI 中断

表 13-1: SPI 中断请求

中断事件	事件标志	使能控制位
发送缓冲区为空	TXE	TXEIE
接收缓冲区非空	RXNE	PXNEIE
主模式故障	MODF	ERRIE
溢出错误	OVR	
CRC 错误	CRCERR	
TI 帧格式错误	FRE	ERRIR

13.4 SPI 寄存器映射

(SPI 基地址 = 0x4007_3000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
SPI_CR1	0x000	R/W	SPI 控制寄存器 1	0x0
SPI_CR2	0x004	R/W	SPI 控制寄存器 2	0x0
SPI_SR	0x008	R/W	SPI 状态寄存器	0x02
SPI_DR	0x00C	R/W	SPI 数据寄存器	0x0
SPI_CRCPR	0x010	R/W	SPI CRC 多项式寄存器	0x07
SPI_RXCRCR	0x014	RO	SPI RX CRC 寄存器	0x0
SPI_TXCRCR	0x018	RO	SPI TX CRC 寄存器	0x0

13.5 SPI 寄存器说明

外设寄存器可支持半字（16 位）或字（32 位）访问。

13.5.1 SPI 控制寄存器 1 (SPI_CR1)

位	符号	描述	复位值
15	BIDIMODE	双向通信数据模式使能 (Bidirectional data mode enable): 0: 选择双线单向通信数据模式 1: 选择单线双向通信数据模式	0
14	BIDIOE	双向通信模式下的输出使能: 0: 禁止输出（只接收模式） 1: 使能输出（只发送模式） 此位结合 BIDIMODE 位, 用于选择双向通信模式下的传输方向 注意: 在主模式下, 使用 MOSI 引脚; 在从模式下, 使用 MISO 引脚。	0
13	CRCEN	硬件 CRC 计算使能 (Hardware CRC calculation enable): 0: 禁止 CRC 计算 1: 使能 CRC 计算 注意: 为确保正确操作, 只应在禁止 SPI (SPE = "0") 时对此位执行写操作	0
12	CRCNEXT	下一次传输 CRC (CRC transfer next): 0: 数据阶段（无 CRC 阶段） 1: 下一次传输为 CRC (CRC 阶段) 注意: 当 SPI 配置为全双工或只发送模式时, 只要最后一个数据写入 SPI_DR 寄存器, 就必须对 CRCNEXT 执行写操作。当 SPI 配置为只接收模式时, 必须在接收到倒数第二个数据之后将 CRCNEXT 置 1。当传输由 DMA 管理时, 此位应保持清零状态。	0
11	DFF	数据帧格式 (Data frame format): 0: 为发送/接收选择 8 位数据帧格式 1: 为发送/接收选择 16 位数据帧格式 注意: 为确保正确操作, 只应在禁止 SPI (SPE = "0") 时对此位执行写操作	0
10	RXONLY	只接收 (Receive only) 0: 全双工（发送和接收） 1: 关闭输出（只接收模式） 此位结合 BIDIMODE 位, 用于选择双线单向模式下的传输方向。此位也适用于多从模式系统, 在此类系统中, 不会访问特	0

		定从器件，也不会损坏访问的从器件的输出。	
9	SSM	软件从器件管理: 0: 禁止软件从器件管理 1: 使能软件从器件管理 当 SSM 位置 1 时，NSS 引脚输入替换为 SSI 位的值。不适用于 SPI TI 模式	0
8	SSI	内部从器件选择 (Internal slave select): 仅当 SSM 位置 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 IO 。 不适用于 SPI TI 模式	0
7	LSBFIRST	帧格式 (Frame format): 0: 先发送 MSB 1: 先发送 LSB 注意：正在通信时不应更改此位。不适用于 SPI TI 模式	0
6	SPE	SPI 使能 (SPI enable): 0: 关闭外设 1: 使能外设 注意：关闭 SPI 时,请按照 13.3.8 关闭 SPI 中所述的步骤操作	0
5:3	BR	波特率控制 (Baud rate control): 000: fPCLK/2 100: fPCLK/32 001: fPCLK/4 101: fPCLK/64 010: fPCLK/8 110: fPCLK/128 011: fPCLK/16 111: fPCLK/256 注意：正在通信时不应更改这些位。	0x0
2	MSTR	主模式选择 (Master selection): 0: 从配置 1: 主配置 注意：正在通信时不应更改此位。	0
1	CPOL	时钟极性 (Clock polarity): 0: 空闲状态时，SCK 保持低电平 1: 空闲状态时，SCK 保持高电平 注意：正在通信时不应更改此位。	0
0	CPHA	时钟相位 (Clock phase): 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 注意：正在通信时不应更改此位。	0

13.5.2 SPI 控制寄存器 2 (SPI_CR2)

位	符号	描述	复位值
15:8	-	保留	0x0
7	TXEIE	发送缓冲区空中断使能: 0: 屏蔽 TXE 中断 1: 使能 TXE 中断。TXE 标志置 1 时产生中断请求。	0
6	RXNEIE	接收缓冲区非空中断使能: 0: 屏蔽 RXNE 中断 1: 使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。	0
5	ERRIE	错误中断使能: 0: 屏蔽错误中断 1: 使能错误中断 此位用于控制在错误状况发生时是否产生中断（SPI 模式中的 CRCERR、OVR、MODF，以及 OVR 和 FRE）。	0
4	FRF	帧格式 (Frame format): 0: SPI Motorola 模式 1: SPI TI 模式	0
3	-	保留	0
2	SSOE	SS 输出使能 (SS output enable): 0: 在主模式下禁止 SS 输出，可在多主模式配置下工作 1: 在主模式下使能 SS 输出，不能在多主模式环境下工作	0
1	TXDMAEN	发送缓冲区 DMA 使能 (Tx buffer DMA enable) 0: 关闭发送缓冲区 DMA 1: 使能发送缓冲区 DMA 当此位置 1 时，每当 TXE 标志置 1 时，即产生 DMA 请求	0
0	RXDMAEN	接收缓冲区 DMA 使能 (Rx buffer DMA enable) 0: 关闭接收缓冲区 DMA 1: 使能接收缓冲区 DMA 当此位置 1 时，每当 RXNE 标志置 1 时，即产生 DMA 请求。	0

13.5.3 SPI 状态寄存器 (SPI_SR)

位	符号	描述	复位值
15: 9	-	保留	0x0
8	FRE	帧格式错误标志 (Frame format error): 0: 没有帧格式错误 1: 出现帧格式错误 此标志由硬件置 1，当读取 SPIx_SR 寄存器时由软件清零。	0
7	BSY	忙标志 (Busy flag): 0: SPI 不繁忙 1: SPI 忙于通信或者发送缓冲区不为空 此标志由硬件置 1 和清零。注意：BSY 标志必须谨慎使用	0
6	OVR	上溢标志 (Overrun flag): 0: 未发生上溢 1: 发生上溢 此标志由硬件置 1，可由软件序列复位。	0
5	MODF	模式故障标志 (Mode fault): 0: 未发生模式故障 1: 发生模式故障 此标志由硬件置 1，可由软件序列复位。	0
4	CRCERR	CRC 错误标志 (CRC error flag): 0: 接收到的 CRC 值与 SPI_RXCRCR 值匹配 1: 接收到的 CRC 值与 SPI_RXCRCR 值不匹配 此标志由硬件置 1，通过软件写入 0 来清零。	0
3:2	-	保留	0
1	TXE	发送缓冲区为空标志 (Transmit buffer empty): 0: 发送缓冲区非空 1: 发送缓冲区为空	1
0	RXNE	接收缓冲区非空标志 (Receive buffer not empty): 0: 接收缓冲区为空 1: 接收缓冲区非空	0

13.5.4 SPI 数据寄存器 (SPI_DR)

位	符号	描述	复位值
15:0	DR	数据寄存器 (Data register) 已接收或者要发送的数据。	0x0

数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。

针对 SPI 模式的说明： 发送或接收的数据为 8 位或 16 位，具体取决于数据帧格式选择位（SPI_CR1 寄存器中的 DFF）。必须在使能 SPI 前进行此项选择，以确保操作正确。对于 8 位数据帧，缓冲区为 8 位，只有寄存器的 LSB(SPI_DR[7:0]) 用于发送/接收。在接收模式下，寄存器的 MSB(SPI_DR[15:8])强制为 0。对于 16 位数据帧，缓冲区为 16 位，整个寄存器 SPI_DR[15:0]均用于发送/接收。

13.5.5 SPI CRC 多项式寄存器 (SPI_CRCPR)

位	符号	描述	复位值
15:0	CRCPOLY	CRC 多项式寄存器 此寄存器包含用于 CRC 计算的多项式	0x0007

CRC 多项式 (0007h) 是此寄存器的复位值。可根据需要配置另一个多项式。

13.5.6 SPI RX CRC 寄存器 (SPI_RXCRCR)

位	符号	描述	复位值
15:0	RXCRC	接收 CRC 寄存器 (Rx CRC register)	0x0

使能 CRC 计算后，RxCRC[15:0] 位将包含后续接收字节在计算后所得到的 CRC 值。当 SPI_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPI_CRCPR 寄存器中编程的多项式连续计算。

数据帧格式设置为 8 位数据（SPI_CR1 的 DFF 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位数据帧格式（SPI_CR1 寄存器的 DFF 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

13.5.7 SPI TX CRC 寄存器 (SPI_TXCRCR)

位	符号	描述	复位值
15:0	TXCRC	发送 CRC 寄存器 (Tx CRC register)	0x0

使能 CRC 计算后，TxCRC[7:0] 位将包含后续发送字节在计算后所得到的 CRC 值。当 SPI_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPI_CRCPR 寄存器中编程的多项式连续计算。

数据帧格式设置为 8 位数据（SPI_CR1 的 DFF 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位数据帧格式（SPI_CR1 寄存器的 DFF 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

第14章 I2C 接口

14.1 I2C 简介

I2C（内部集成电路）总线接口用作微控制器和 I2C 串行总线之间的接口。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。它支持标准和快速模式。它还与 SMBus 2.0 兼容。它可以用于多种用途，包括 CRC 生成和验证、SMBus（系统管理总线）以及 PMBus（电源管理总线）。根据器件的不同，可利用 DMA 功能来减轻 CPU 的工作量。

14.2 I2C 主要特性

- 并行总线/I2C 协议转换器
- 多主模式功能：同一接口既可用作主模式也可用作从模式
- I2C 主模式特性：
 - 时钟生成
 - 起始位和停止位生成
- I2C 从模式特性：
 - 可编程 I2C 地址检测
 - 双寻址模式，可对 2 个从地址应答
 - 停止位检测
- 7 位/10 位寻址以及广播呼叫的生成和检测
- 支持不同的通信速度：
 - 标准速度（高达 100 KHz）
 - 快速速度（高达 400 KHz）
- 状态标志：
 - 发送/接收模式标志
 - 字节传输结束标志

— I2C 忙碌标志

- 错误标志：
 - 主模式下的仲裁丢失情况
 - 地址/数据传输完成后的应答失败
 - 检测误放的起始位和停止位
 - 禁止时钟延长后出现的上溢/下溢
- 2 个中断向量：
 - 一个中断由成功的地址/数据字节传输事件触发
 - 一个中断由错误状态触发
- 可选的时钟延长
- 带 DMA 功能的 1 字节缓冲
- 可配置的 PEC（数据包错误校验）生成或验证：
 - 在 Tx 模式下，可将 PEC 值作为最后一个字节进行传送
 - 针对最后接收字节的 PEC 错误校验

- SMBus 2.0 兼容性:
 - 25 ms 时钟低电平超时延迟
 - 10 ms 主器件累计时钟低电平延长时间
 - 25 ms 从器件累计时钟低电平延长时间
 - 具有 ACK 控制的硬件 PEC 生成/验证
 - 支持地址解析协议 (ARP)

- PMBus 兼容性

注意：上述某些特性可能不适用于某些产品。用户应参照产品数据手册来确定 I2C 接口实现所支持的特定特性。

14.3 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（高达 100 KHz）或快速（高达 400 KHz）I2C 总线

14.3.1 模式选择

该接口在工作时可选用以下四种模式之一：

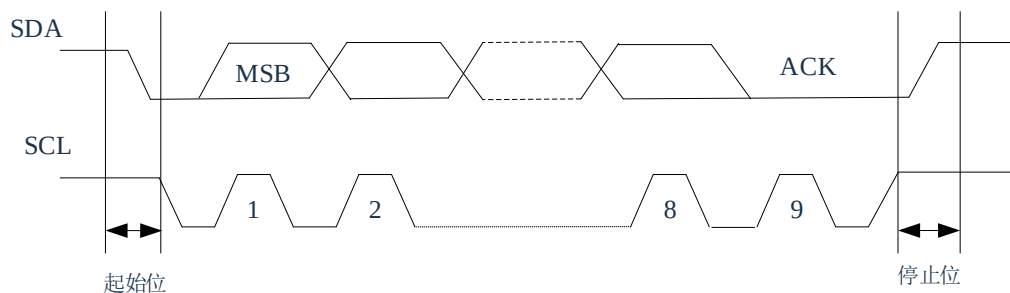
- 从发送器
- 从接收器
- 主发送器
- 主接收器

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。

通信流程

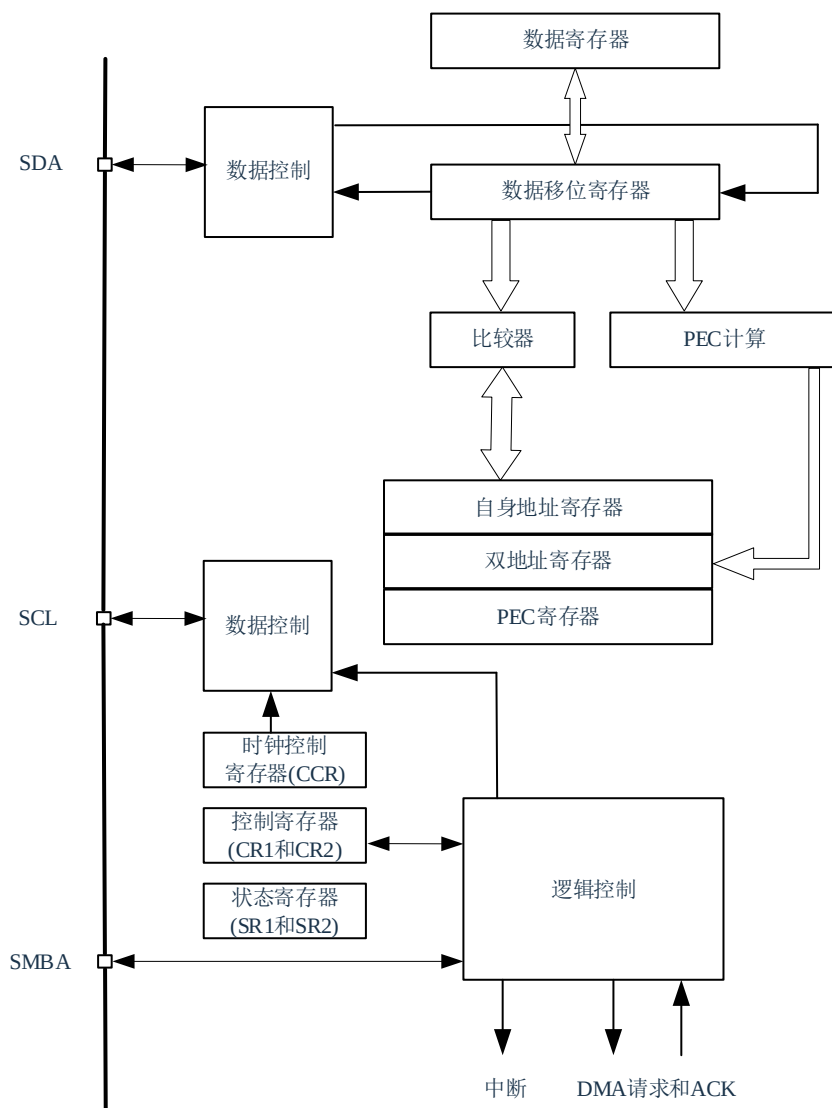
在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下传送。在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发送器发送一个应答位。请参见图 14-1。

图 14-1 IIC 总线协议



应答位可由软件使能或禁止。I2C 接口地址（7 位/10 位双寻址模式和/或广播呼叫地址）可通过软件进行选择。I2C 接口框图如图 14-2 所示

图 14-2 IIC 框图



14.3.2 I2C 从模式

默认情况下，I2C 接口在从模式下工作。要将工作模式由默认的从模式切换为主模式，需要生成一个起始位。为了生成正确的时序，必须在 I2C_CR2 寄存器中对外设输入时钟进行编程。外设输入时钟频率的下限为：

- 标准模式下 2 MHz
- 快速模式下 4 MHz

检测到起始位后，便会立即接收到来自 SDA 线的地址并将其送到移位寄存器。之后，会将其与接口地址 (OAR1) 和 OAR2（如果 ENDUAL=1）或者广播呼叫地址（如果 ENGCB = 1）进行比较。

注意：在 10 位寻址模式下，比较对象还包括头序列 (11110xx0)，其中，xx 表示该地址的两个最高有效位。

头或地址不匹配：接口会忽略它并等待下一个起始位。

头匹配（仅针对 10 位模式）：如果 ACK 位置 1，则接口会生成一个应答脉冲并等待 8 位从地址。

地址匹配：接口会依次：

- 发出应答脉冲（如果 ACK 位置 1）
- ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
- 如果 ENDUAL=1，则软件必须读取 DUALF 位状态来核对哪些从地址进行了应答。

在 10 位模式下，完成地址序列接收后，从模式始终处于接收模式。在接收到重复起始位以及一个匹配地址位和最低有效位均置 1 的头序列 (11110xx1) 后，它会进入发送模式。

TRA 位指示从设备是处于接收模式还是处于发送模式。

从发送器

在接收到地址并将 ADDR 清零后，从设备会通过内部移位寄存器将 DR 寄存器中的字节发送到 SDA 线。

从设备会延长 SCL 低电平时间，直到 ADDR 位清零且 DR 寄存器中填满待发送数据为止（请参见图 14-3 传输序列 EV1 EV3）。

接收到应答脉冲时：

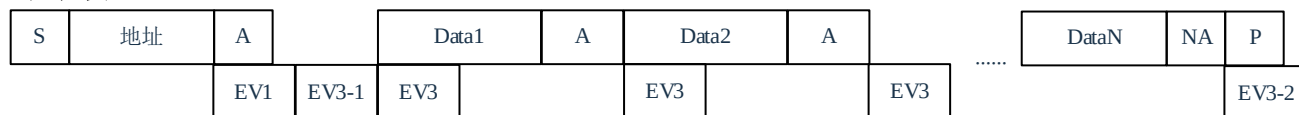
● TxE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。如果在下一次数据传输结束之前 TxE 位已置 1 但某些数据尚未写入 I2C_DR 寄存器，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，直到通过软件对 I2C_SR1 读操作，以及对 I2C_DR 写操作后，把 BTF 清零为止。

1. EV1 和 EV3_1 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。

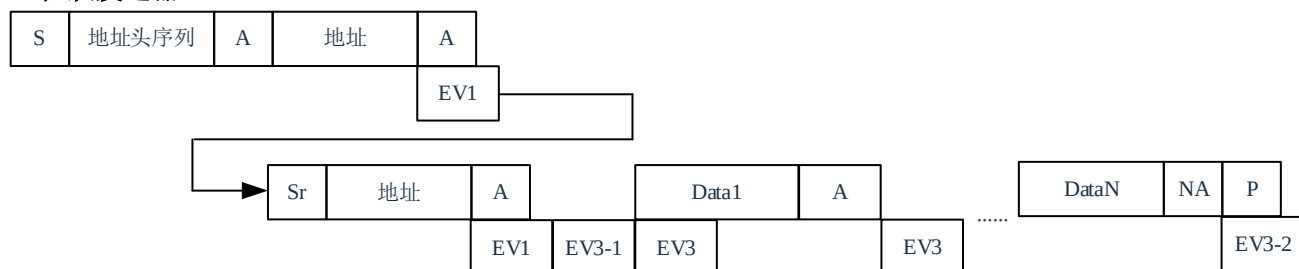
2. 如果软件序列在下一个字节传输结束之前未能完成，EV3 事件会延长 SCL 低电平时间

图 14-3 从发送器的传输序列图

7 位从发送器



10 位从发送器



图注：S = 起始位，Sr = 重复起始位，P = 停止位，A = 应答，NA = 非应答

EV_x = 事件（如果 ITEVFEN=1 则发生中断）

EV1: ADDR=1, 通过先读取 SR1 再读取 SR2 来清零。

EV3-1: TxE =1, 移位寄存器为空，数据寄存器为空，在 DR 中写入 Data1。

EV3: TxE =1, 移位寄存器非空，数据寄存器为空，通过对 DR 执行写操作来清零。

EV3-2: AF=1: 通过在 SR1 寄存器的 AF 位写入“0”将 AF 清零

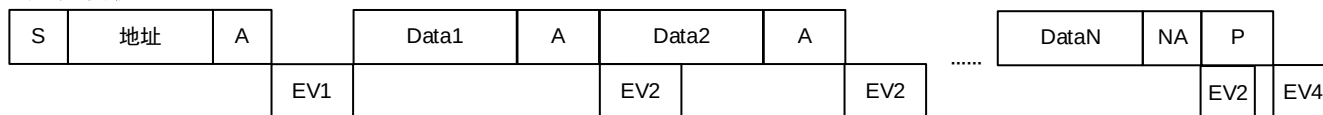
从接收器

在接收到地址并将 ADDR 位清零后，从设备会通过内部移位寄存器接收 SDA 线中的字节并将其保存到 DR 寄存器。在每个字节接收完成后，接口都会依次：

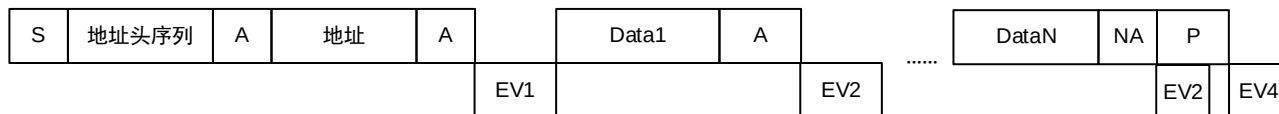
- 发出应答脉冲（如果 ACK 位置 1）
- RxNE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。如果在下一次数据接收结束之前 RxNE 位已置 1 但 DR 寄存器中的数据尚未读取，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，直到软件通过读取 I2C_DR 寄存器来把 BTF 清零（请参见图 14-4 传输序列）

图 14-4 从接收器的传输序列图

7 位从接收器



10 位从接收器



图注: S = 起始位, Sr = 重复起始位, P = 停止位, A = 应答, NA = 非应答

Ev_x = 事件（如果 ITEVFEN=1 则发生中断）

EV1: ADDR=1, 通过先读取 SR1 再读取 SR2 来清零。

EV2: RxNE =1, 通过读取 DR 寄存器清零

EV4: STOPF=1, 通过先读取 SR1 寄存器再写入 CR1 寄存器来清零。

1. EV1 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节接收结束之前未能完成，EV2 事件会延长 SCL 低电平时间。
3. 检查了 SR1 寄存器内容之后，用户应针对各个被置位的标志执行完整的清零序列。

对于 ADDR 和 STOPF 标志，需要在 I2C 中断程序中执行以下序列：

对 SR1 执行读操作

```
if (ADDR == 1) {对 SR1 执行读操作；对 SR2 执行读操作}
```

```
if (STOPF == 1) {对 SR1 执行读操作；对 CR1 执行写操作}
```

这样做的目的是为了确保 ADDR 和 STOPF 这两个标志都清零（如果二者均为被置位）。

关闭从设备通信

传输完最后一个数据字节之后，主设备会生成一个停止位。接口会检测此条件并：

- 将 STOPF 位置 1 并在 ITEVFEN 位置 1 时生成一个中断。

通过先读取 SR1 寄存器然后写入 CR1 寄存器的方式将 STOPF 位清零（请参见图 14-4：从接收器的传输序列图 EV4）。

14.3.3 I2C 主模式

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。只要通过 START 位在总线上生成了起始位，即会选中主模式。

在主模式中要求执行以下序列。

- 在 I2C_CR2 寄存器中对外设输入时钟进行编程，以生成正确的时序
- 配置时钟控制寄存器
- 配置上升时间寄存器
- 对 I2C_CR1 寄存器进行编程，以便使能外设
- 将 I2C_CR1 寄存器的 START 位置 1，以生成起始位

外设输入时钟频率的下限为：

- 标准模式下 2 MHz
- 快速模式下 4 MHz

起始位

START 位置 1 后，接口会在 BUSY 位清零后生成一个起始位并切换到主模式（M/SL 位置 1）。

注意：在主设备下，START 位置 1 后，接口会在当前字节传输结束后生成一个重复起始位。

起始位发出之后：

- SB 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。

接下来主设备会等待软件对 SR1 执行读操作，然后把从设备地址写入 DR 寄存器（请参见图 14-5 和图 14-6 传输序列 EV5）。

从地址传输

接下来从地址会通过内部移位寄存器发送到 SDA 线。

- 在 10 位寻址模式中，发送头序列会产生以下事件：
 - ADDR10 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。接下来主设备会等待软件读取 SR1，然后把第二个地址字节写入 DR 寄存器（请参见图 14-5 和图 14-6 传输序列）。
 - ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。

接下来主设备会等待对 SR1 寄存器执行读操作，然后对 SR2 寄存器执行读操作（请参见图 14-5 和图 14-6 传输序列）。

- 在 7 位寻址模式下，会发送一个地址字节。

地址字节被发出后，

- ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
- 接下来主设备会等待对 SR1 寄存器执行读操作，然后对 SR2 寄存器执行读操作（请参见图 14-5 和图 14-6 传输序列）。

主设备会根据发送的从地址字节 LSB 来决定是进入发送模式还是接收模式。

- 在 7 位寻址模式下，
 - 要进入发送模式，主设备会发送从地址并将 LSB 复位。
 - 要进入接收模式，主设备会发送从地址并将 LSB 置 1。
- 在 10 位寻址模式下，
 - 要进入发送模式，主设备会先发送头序列 (11110xx0)，然后发送从地址（其中 xx 表示该地址的两个最高有效位）。

- 要进入接收模式，主设备会先发送头序列 (11110xx0)，然后发送从地址。接下来会发送一个重复起始位，然后再发送头序列 (11110xx1)（其中 xx 表示地址的两个最高有效位）。

TRA 位指示主设备是处于接收模式还是处于发送模式。

主发送器

在发送出地址并将 ADDR 清零后，主设备会通过内部移位寄存器将 DR 寄存器中的字节发送到 SDA 线。

主设备会一直等待，直到首个数据字节被写入 I2C_DR 为止（请参见图 14-5 传输序列 EV8_1）。

接收到应答脉冲后，TxE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。

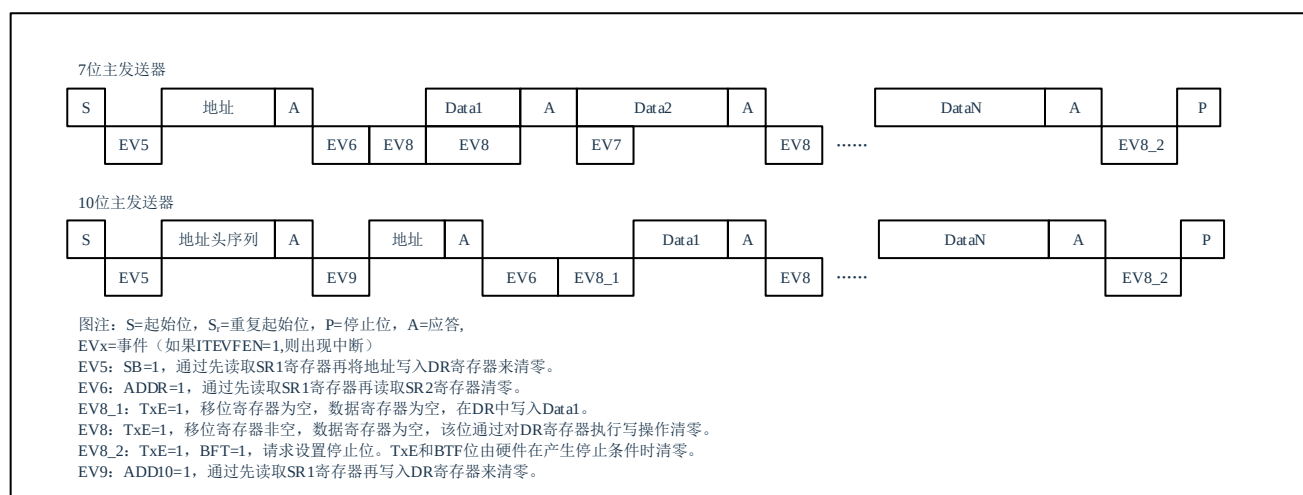
如果在上一次数据传输结束之前 TxE 位已置 1 但数据字节尚未写入 DR 寄存器，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，等待 I2C_DR 寄存器被写入，以将 BTF 清零。

结束通信

当最后一个字节写入 DR 寄存器后，软件会将 STOP 位置 1 以生成一个停止位（请参见图 14-5 传输序列 EV8_2）。接口会自动返回从模式（M/SL 位清零）。

注意：当 TxE 或 BTF 中的任何一个置 1 时，应在 EV8_2 事件期间对停止位进行编程。

图 14-5 主发送器的传输序列图



1. EV5、EV6、EV9、EV8_1 和 EV8_2 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节传输结束之前未能完成，EV8 事件会延长 SCL 低电平时间。

主接收器

完成地址传输并将 ADDR 位清零后，I2C 接口会进入主接收模式。在此模式下，接口会通过内部移位寄存器接收 SDA 线中的字节并将其保存到 DR 寄存器。在每个字节传输结束后，接口都会依次：

1. 发出应答脉冲（如果 ACK 位置 1）

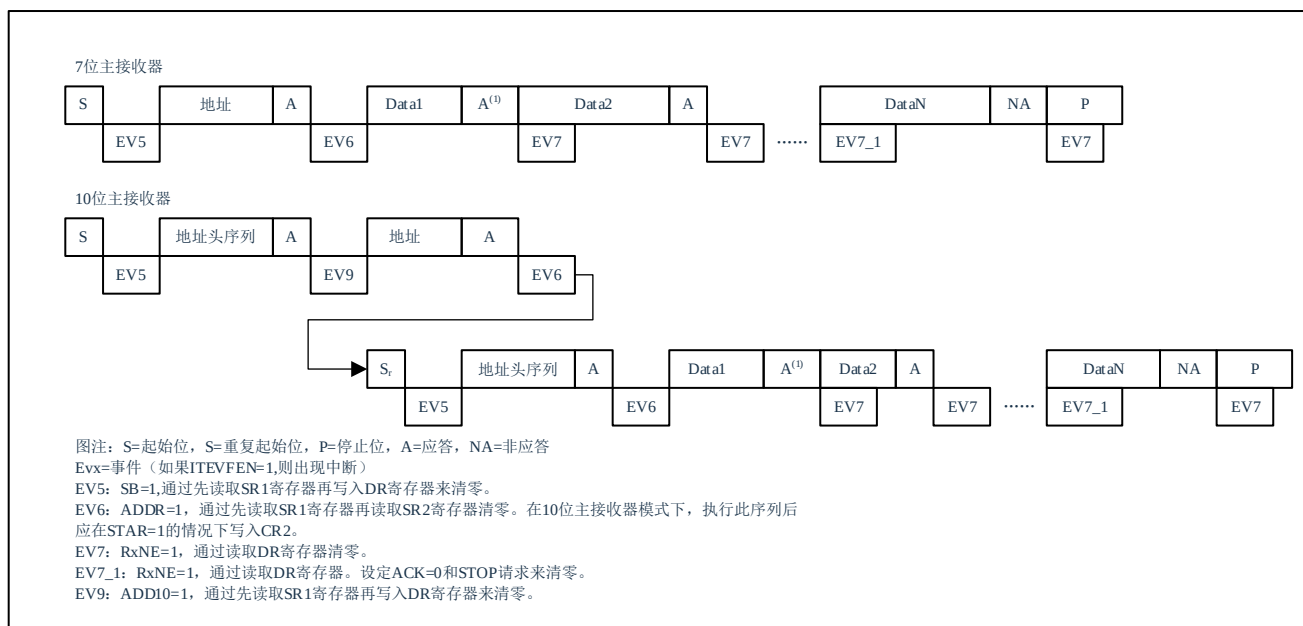
2. RxNE 位置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断（参见图 14-6 传输序列 EV7）。如果在上一次数据接收结束之前 RxNE 位已置 1 但 DR 寄存器中的数据尚未读取，则 BTF 位会由硬件置 1，而接口会一直延长 SCL 低电平，等待 I2C_DR 寄存器被写入，以将 BTF 清零。

结束通信

主设备会针对自从设备接收的最后一个字节发送 NACK。在接收到此 NACK 之后，从设备会释放对 SCL 和 SDA 线的控制。随后，主设备可发送一个停止位/重复起始位。

1. 为了在最后一个接收数据字节后生成非应答脉冲，必须在读取倒数第二个数据字节后（倒数第二个 RxNE 事件之后）立即将 ACK 位清零。
2. 要生成停止位/重复起始位，软件必须在读取倒数第二个数据字节后（倒数第二个 RxNE 事件之后）将 STOP/START 位置 1。
3. 在只接收单个字节的情况下，会在 EV6 期间（在 ADDR 标志清零之前）禁止应答并在 EV6 之后生成停止位。生成停止位后，接口会自动返回从模式（M/SL 位清零）

图 14-6 主接收器的传输序列图



1. 如果接收单个字节，则为 NA。
2. EV5、EV6 和 EV9 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
3. 如果软件序列在下一个字节接收结束之前未能完成，EV7 事件会延长 SCL 低电平时间。
4. EV7_1 软件序列必须在当前字节传输的 ACK 脉冲之前结束。

如果 EV7_1 软件序列未能在当前字节传输的 ACK 脉冲之前结束，建议采用下列步骤。

必须遵循以下步骤以确保：

- 在结束上一个数据接收之前及时将 ACK 位复位
- 在上一次数据接收后将 STOP 位置位且不接收补充数据。

针对 2 字节的接收：

- 等待 ADDR = 1（SCL 保持低电平，直到 ADDR 标志清零）
- 将 ACK 复位，POS 置位
- 将 ADDR 标志清零
- 等待 BTF = 1（数据 1 在 DR 中，数据 2 在移位寄存器中，SCL 保持低电平，直到读取了数据 1 为止）
- 将 STOP 置位
- 读取数据 1 和数据 2

针对 N > 2 的字节接收，从 N-2 数据接收开始

- 等待 $BTF = 1$ （数据 N-2 在 DR 中，数据 N-1 在移位寄存器中，SCL 保持低电平，直到读取了数据 N-2 为止）
- 将 ACK 复位
- 读取数据 N-2
- 等待 $BTF = 1$ （数据 N-1 在 DR 中，数据 N 在移位寄存器中，SCL 保持低电平，直到读取了数据 N-1 为止）
- 将 STOP 置位
- 读取数据 N-1 和数据 N

14.3.4 错误条件

以下错误条件可能导致通信失败。

总线错误 (BERR)

当 I2C 接口在传输地址或数据期间检测到外部停止位或起始位时，会出现此错误。在这种情况下：

- BERR 位置 1 并在 ITERREN 位置 1 时生成一个中断
- 在从模式下：硬件会丢弃数据并释放数据线：
 - 在误放起始位的情况下，从模式会认为它是重复起始位并会等待一个地址或停止位
 - 在误放停止位的情况下，从模式会按停止位操作且硬件会释放数据线
- 在主模式下：不会释放数据线且不会影响当前传输状态。由软件决定是否中止当前传输

应答失败 (AF)

当接口检测到未应答脉冲会出现此错误。在这种情况下：

- AF 位置 1 并在 ITERREN 位置 1 时生成一个中断
- 接收到 NACK 的发送器必须复位通信：
 - 在从模式下：硬件释放数据线
 - 在主模式下：必须由软件生成一个停止位或重复起始位

仲裁丢失 (ARLO)

当 I2C 接口检测到仲裁丢失时会出现此错误。在这种情况下，

- ARLO 位会由硬件置 1（并在 ITERREN 位置 1 时生成一个中断）
- I2C 接口会自动返回从模式（M/SL 位清零）。I2C 失去仲裁时，将无法在本次传输中对自身从地址进行应答，但是能够在赢得仲裁的主设备发出重复起始位后对自身设备进行应答。
- 硬件释放数据线

上溢/下溢错误 (OVR)

当时钟延长已禁止且 I2C 接口正在接收数据时，从模式中可能出现上溢错误。接口已经收到一个字节 (RxNE=1)，但是收到下一个字节之前 DR 中的数据未被读走。在这种情况下，

- 会丢失接收的最后一个字节。
- 在出现上溢错误时，软件应将 RxNE 位清零，而发送器应重新发送最后一个字节。

当时钟延长已禁止且 I2C 接口正在发送数据时，从模式中可能出现下溢错误。下一个字节的时钟信号到来之前，从设备还未把下一个要发送的数据写进 DR (TxNE=1)。在这种情况下，

- 会再次发送 DR 寄存器中的字节
- 用户应确保在出现下溢错误期间从接收器端接收到的数据被丢弃，并确保在 I2C 总线标准指定的时钟低电平时间内写入下一个字节。

对于要传送的首个字节，必须在 ADDR 清零之后且出现首个 SCL 上升沿之前将其写入 DR 寄存器。否则，接收器必须丢弃首个数据。

14.3.5 可编程噪声滤波器

在快速模式下，I2C 标准要求将 SDA 和 SCL 线上尖峰脉宽在 50 ns 以内的噪声都抑止掉。

SDA 和 SCL I/O 中采用了模拟噪声滤波器。此滤波器默认为使能，通过将 I2C_FLTR 寄存器中的 ANOFF 位置 1 可禁止它。

将 DNF[3:0] 位配置为一个非零值可使能数字噪声滤波器。这可以抑制 SDA 和 SCL 输入上脉宽小于 $DNF[3:0] * TPCLK1$ 的噪声。

使能数字噪声滤波器后，SDA 保持时间可增加 $(DNF[3:0] + 1) * TPCLK$ 。

为了符合 I2C 总线规范版本 2.1 (Thd:dat) 中对最大保持时间的要求，必须使用表 14-1 所示的限制条件对 DNF 位进行编程并假定模拟滤波器已禁止。

注意：DNF[3:0] 必须在 I2C 已禁止 (PE = 0) 的情况下配置。如果模拟滤波器也已使能，则需将数字滤波器添加到模拟滤波器中。

表 14-1 符合 Thd:dat(max)的 DNF[3:0]最大值

PCLK 频率	DNF 最大值	
	标准模式	快速模式
$2 <= F_{PCLK} <= 5$	2	0
$5 < F_{PCLK} <= 10$	12	0
$10 < F_{PCLK} <= 20$	15	1
$20 < F_{PCLK} <= 30$	15	7
$30 < F_{PCLK} <= 40$	15	13
$40 < F_{PCLK} <= 48$	15	15

注意：上表根据各个频率范围的最低频率给出了相应的限制条件。如果系统可以支持最大保持时间违例，则可使用更大的 DNF 值。

14.3.6 SDA/SCL 线控制

- 如果时钟延长已使能：
 - 发送模式：如果 TxE=1 且 BTF=1：接口会在发送数据之前保持时钟线为低电平，以等待微控制器将字节写入数据寄存器（缓冲寄存器和移位寄存器均为空）。
 - 接收模式：如果 RxNE=1 且 BTF=1：接口会在接收数据之后保持时钟线为低电平，以等待微控制器将字节读入数据寄存器（缓冲寄存器和移位寄存器均已满）。
- 如果从模式中的时钟延长已禁止：
 - RxNE=1，且在下一个数据接收完成之前还未读走 DR 中的数据就出现上溢错误。会丢失接收的最后一个字节。
 - TxE=1，且在下一个数据的始终到来之前还未把发送数据写到 DR 中，就出现下溢错误。会再次发送同一字节。
 - 不会对写冲突进行管理。

14.3.7 SMBus

前言

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。系统可使用 SMBus 与设备进行消息传递，而无需切换各个控制线。

系统管理总线规范涉及三类器件。从器件，用于接收或响应命令。主器件，用于发出命令、生成时钟和中止传输。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 - 从设备功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

SMBus 与 I2C 的相似之处

- 双线制总线协议（1 个时钟总线，1 个数据总线）+ 可选 SMBus 报警线
- 主从通信，主器件提供时钟
- 多主器件功能
- SMBus 数据格式与 I2C 7 位地址格式相似（图 14-1）。

SMBus 与 I2C 之间的差异

下表介绍了 SMBus 与 I2C 之间的差异

表 14-2 SMBus 与 I²C

SMBus	I ² C
最大速度 100KHz	最大速度 400KHz
最小时钟速度 10KHz	无最小时钟速度
35m 时钟低电平超时	无超时
逻辑电平固定	逻辑电平取决于 V _{DD}
地址类型不同（保留、动态等）	7 位、10 位和广播从模式地址类型
总线协议不同（快速命令、过程调用等）	无总线协议

SMBus 应用用途

通过系统管理总线，器件可以提供制造商信息、告诉系统它的型号/部件号、保存暂停事件的状态、报告不同的错误类型、接受控制参数并返回其状态。SMBus 可针对系统和电源管理相关的任务提供控制总线。

器件标识

系统管理总线中作为从器件的任何器件均具有一个唯一地址，被称为从地址。有关保留的从地址列表的信息，请参见 SMBus 规范版本 2.0 (<http://smbus.org/specs/>)。

总线协议

SMBus 规范支持多达 9 类总线协议。有关这些协议和 SMBus 地址类型的详细信息，请参见 SMBus 规范版本 2.0 (<http://smbus.org/specs/>)。这些协议应通过用户软件实施。

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。地址解析协议 (ARP) 具有以下属性：

- 地址分配采用标准的 SMBus 物理层仲裁机制
- 器件通电时，分配的地址保持不变；器件断电时，也允许保留地址

- 完成地址分配后不会带来额外的 SMBus 数据包开销。（即对已分配的从地址进行后续访问与访问固定地址的器件所产生的开销相同。）
- 任何 SMBus 主器件都可以遍历总线

唯一的器件标识符 (UDID)

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)。

有关 128 位 UDID 和 ARP 的详细信息，请参见 SMBus 规范版本 2.0 (<http://smbus.org/specs/>)。

SMBus 报警模式

SMBus 报警是带有中断线的可选信号，主要用于希望扩展它们的控制能力而牺牲一个引脚的器件。SMBA 是与 SCL 和 SDA 信号类似的线与信号。SMBA 与 SMBus 广播地址配合使用。使用 SMBus 调用的消息长度为 2 字节。

将 I2C_CR1 寄存器中的 ALERT 位置 1 后一个只具有从功能的器件可通过 SMBA 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（简称 ARA，其值为 0001 100X）同步访问所有 SMBA 器件。只有那些将 SMBA 拉到低电平的器件会确认报警响应地址。通过 I2C_SR1 寄存器中的 SMBALERT 状态标志可确定这一状态。主机会执行修改后的接收字节操作。由从发送器件提供的 7 位器件地址被放置在字节的 7 个最高有效位。第 8 位可以是 0 或 1。

如果有不止一个器件将 SMBA 拉为低电平，则在从地址传输期间，具有最高优先级（最低位地址）的器件会通过标准仲裁获得通信权限。在确认从地址之后，器件必须释放 SMBA。如果消息传输结束后主机检测到 SMBA 仍为低电平，会再次读取 ARA。未实现 SMBA 信号的主机会定期访问 ARA。

有关 SMBus 报警模式的详细信息，请参见 SMBus 规范版本 2.0 (<http://smbus.org/specs/>)。

超时错误

SMBus 和 I2C 之间存在一些定时规范方面的差异。

SMBus 定义了一个时钟低电平超时，TIMEOUT 为 35 ms。另外，SMBus 还指定 TLOW: SEXT 作为从器件的累积时钟低电平延长时间。SMBus 指定 TLOW: MEXT 作为主器件的累积时钟低电平延长时间。有关这些超时的详细信息，请参见 SMBus 规范版本 2.0(<http://smbus.org/specs/>)。

I2C_SR1 寄存器中的 Timeout 或 Tlow 错误状态标志表明了此特性的状态。

如何在 SMBus 模式下使用该接口

要从 I2C 模式切换到 SMBus 模式，应执行以下步骤。

- 将 I2C_CR1 寄存器中的 SMBus 位置 1
- 根据应用的要求配置 I2C_CR1 寄存器中的 SMBTYPE 和 ENARP 位

如果要将某个器件配置为主器件，请按照 I2C 主模式中介绍的起始位生成步骤进行操作。否则按照：I2C 从模式中的顺序操作。

该应用需要通过软件控制各种 SMBus 协议。

- ENARP=1 且 SMBTYPE=0 时使用 SMB 器件默认地址
- ENARP=1 且 SMBTYPE=1 时使用 SMB 主机头字段
- SMBALERT=1 时使用 SMB 报警响应地址

14.3.8 DMA 请求

DMA 请求（在使能后）仅用于数据传输。当发送数据寄存器变空以及接收数据寄存器变满时会生成 DMA 请求。进行 I2C 数据传输之前，必须先初始化并使能 DMA。I2C_CR2 寄存器中的 DMAEN 位必须在 ADDR 事件之前置 1。在主模式或从模式下，如果已使能时钟延长，DMAEN 位也可以在 ADDR 事件期间于 ADDR 标志清零之前置 1。结束当前字节传输之前，必须发出 DMA 请求。当传输的数据量达到相应 DMA 通道编程设定的值时，DMA 控制器会发送一个结束传输 EOT 信号给 I2C 接口，并生成一个传输完成中断（如果已使能）：

- 主发送器：在 EOT 中断后的中断程序中，禁止 DMA 请求，然后在等到 BTF 事件后设置停止位。
- 主接收器
 - 当要接收的字节数等于或大于 2 时，DMA 控制器会在收到倒数第二个数据字节（第 N-1 个数据时）发送一个硬件信号 EOT_1。如果 I2C_CR2 寄存器中的 LAST 位置 1，I2C 会在 EOT_1 后的下一个字节之后自动发送一个 NACK。用户可在 DMA 传输完成中断（如果已使能）程序中生成停止位。
 - 当必须接收单个字节时：必须在 EV6 事件期间于 ADDR 标志清零之前对 NACK 进行编程，即当 ADDR=1 时编程设定 ACK=0。接下来，用户可在 ADDR 标志清零之后或者在执行 DMA 传输完成中断程序时编程设定停止位。

使用 DMA 进行发送

将 I2C_CR2 寄存器中的 DMAEN 位置 1 可以使能 DMA 模式进行发送。当 TXE 位置 1 时，数据将由 DMA 从预置的存储区装载进 I2C_DR 寄存器。要映射一个 DMA 通道以便进行 I2C 发送，请按以下步骤操作：其中的 x 表示通道编号。

1. 设置 DMDARj 寄存器中的 I2C_DR 寄存器地址。每次发生 TxE 事件后，数据都会从存储器移动到此地址。
2. 设置 DMSARj 寄存器中的存储器地址。每次发生 TxE 事件后，数据都会从此存储器加载到 I2C_DR。
3. 在 DMACTj 寄存器中配置要传输的总字节数。在每次 TxE 事件后，此值都会递减。
4. 使用 DMA_CHj_SEL 寄存器中来选择通道的触发源，优先级由通道 0 到通道 3 递减。
5. 将 DMAEN0 寄存器中的 EN 位置 1 以激活通道。当传输的数据量达到 DMA 控制器中编程设定的值时，DMA 会在 DMA 通道中断向量上生成一个中断（如果已使能）：

注意：如果使用 DMA 进行传送，请勿使能 I2C_CR2 寄存器中的 ITBUFEN 位。

使用 DMA 进行接收

将 I2C_CR2 寄存器中的 DMAEN 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 I2C_DR 寄存器加载到使用 DMA 外配置的存储区中（参见 DMA 规范）。要映射一个 DMA 通道以便进行 I2C 接收，请按以下步骤操作：其中的 x 表示通道编号。

1. 设置 DMDARj 寄存器中的 I2C_DR 寄存器地址。每次发生 RxNE 事件后，数据都会从此地址移动到存储器。
2. 设置 DMSARj 寄存器中的存储器地址。每次发生 RXNE 事件后，数据都会从 I2C_DR 寄存器加载到此存储区。
3. 在 DMACTj 寄存器中配置要传输的总字节数。在每次 RxNE 事件后，此值都会递减。
4. 使用 DMA_CHj_SEL 寄存器中来选择通道的触发源，优先级由通道 0 到通道 3 递减。
5. 将 DMAEN0 寄存器中的 EN 位置 1 以激活通道。当传输的数据量达到 DMA 控制器中编程设定的值时，DMA 会在 DMA 通道中断向量上生成一个中断（如果已使能）：

注意：如果使用 DMA 进行接收，请勿使能 I2C_CR2 寄存器中的 ITBUFEN 位。

14.3.9 数据包错误校验

PEC 计算器的应用目的是提高通信的可靠性。PEC 对各个位使用 CRC-8 多项式 $C(x) = x^8 + x^2 + x + 1$ 公式进行串行计算。

- 将 I2C_CR1 寄存器中的 ENPEC 位置 1 即可使能 PEC 计算。PEC 是针对所有消息字节（包括地址和 R/W 位）的 CRC-8 计算。
 - 在发送过程中：在与最后一个字节对应的 TxNE 事件发生后，将 I2C_CR1 寄存器中的 PEC 传输位置 1。PEC 会在最后一个传输的字节之后进行传送
 - 在接收过程中：在与最后一个字节对应的 RxNE 事件发生之后，将 I2C_CR1 寄存器中的 PEC 位置 1，以便接收器在接收到的下一个字节不等于内部计算的 PEC 时发送一个 NACK。在主接收器中，无论校验结果如何，PEC 后都将发送 NACK。在从模式下，必须在 CRC 接收的 ACK 之前设置 PEC。在主模式下，必须在复位 ACK 时设置 PEC 位。
- I2C_SR1 寄存器中还提供 PECERR 错误标志/中断。
- 如果 DMA 和 PEC 计算均已使能：-
 - 在发送过程中：当 I2C 接口接收来自 DMA 控制器的 EOT 信号时，会在最后一个字节之后自动发送 PEC。
 - 在接收过程中：当 I2C 接口接收来自 DMA 控制器的 EOT_1 信号时，会自动将下一个字节视为 PEC 并对其进行校验。在 PEC 接收之后会生成一个 DMA 请求。
- 为了允许进行中间 PEC 传输，可使用 I2C_CR2 寄存器中的一个控制位（LAST 位）来确定它是否真的是最后一个 DMA 传输。如果确实是主接收器的最后一个 DMA 请求，则会在接收最后一个字节后自动发送一个 NACK。
- PEC 计算会因仲裁丢失而失效

14.4 I2C 中断

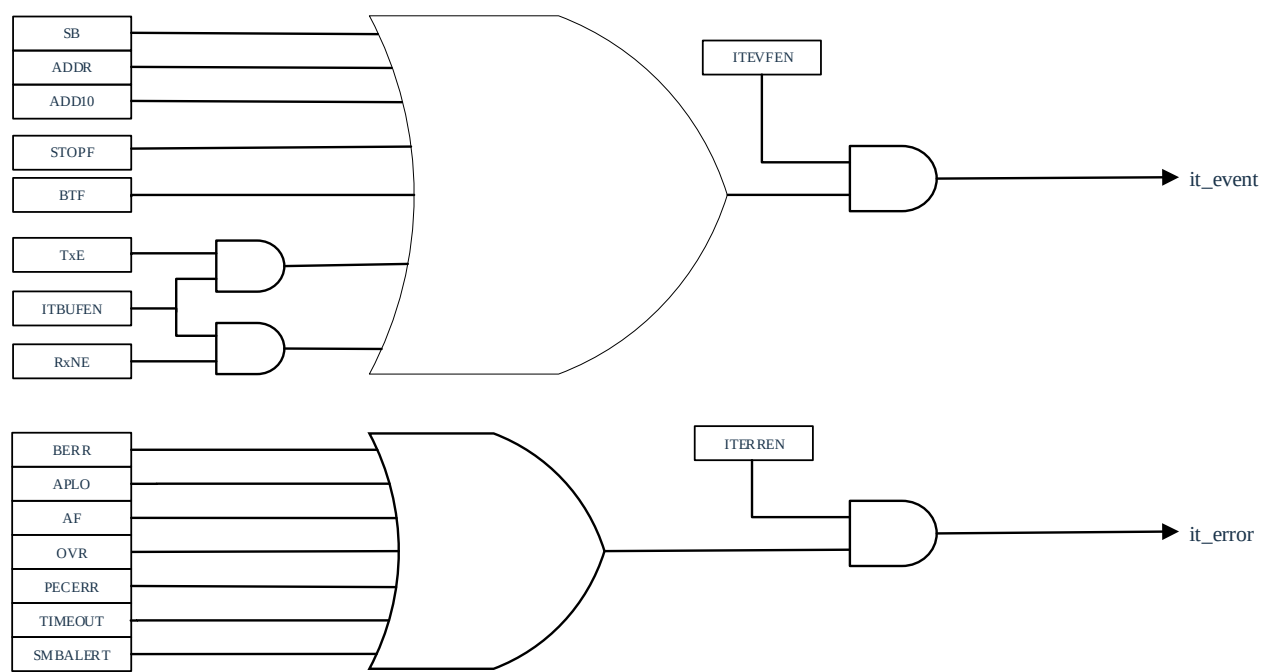
下表列出了 I2C 中断请求列表

表 14-3 符合 Thd:dat(max)的 DNF[3:0]最大值

中断事件	事件标志	使能控制位
发送起始位（主模式）	SB	ITEVFEN
地址已发送（主模式）或地址匹配（从模式）	ADDR	
10 位地址的头段已发送（主模式）	ADD10	
已收到停止位（从模式）	STOPF	
完成数据字节传输	BTF	
接收缓冲区非空	RxNE	ITEVFEN 和 ITBUFEN
发送缓冲区为空	TxE	
总线错误	BERR	ITERREN
仲裁丢失（主模式）	ARLO	
应答失败	AF	
上溢/下溢	OVR	
PEC 错误	PECERR	
超时/Tlow 错误	TIMEOUT	
SMBus 报警	SMBALERT	

注意：SB、ADDR、ADD10、STOPF、BTF、RxNE 和 TxE 通过逻辑或映射到同一个中断通道上。
BERR、ARLO、AF、OVR、PECERR、TIMEOUT 和 SMBALERT 通过逻辑或映射到同一个 中断通道上。

图 14-7 I²C 中断映射图



14.5 I2C 调试模式

当微控制器进入调试模式时，SMBUS 会根据 1.4 选择继续正常工作或者停止工作。

14.6 寄存器映射

(I2C 基地址 = 0x4006_5400) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
I2C_CR1	0x00	R/W	I2C 控制寄存器 1	0x00
I2C_CR2	0x04	R/W	I2C 控制寄存器 2	0x00
I2C_OAR1	0x08	R/W	I2C 自有地址寄存器 1	0x00
I2C_OAR2	0x0C	R/W	I2C 自有地址寄存器 2	0x00
I2C_DR	0x10	R/W	I2C 数据寄存器	0x00
I2C_SR1	0x14	R/W	I2C 状态寄存器 1	0x00
I2C_SR2	0x18	R/W	I2C 状态寄存器 2	0x00
I2C_CCR	0x1C	R/W	I2C 时钟控制寄存器	0x00
I2C_TRISE	0x20	R/W	I2C TRISE 寄存器	0x02
I2C_FLTR	0x24	R/W	I2C FLTR 寄存器	0x00

14.7 寄存器说明

14.7.1 I2C 控制寄存器 1 (I2C_CR1)

位	符号	描述	复位值
15	SWRST	软件复位 (Software reset) 当置 1 时, I2C 处于复位状态。在复位此位之前, 确保 I2C 线已释放且总线空闲。 0: I2C 外设未处于复位状态 1: I2C 外设处于复位状态 注意: 当出现错误或锁定状态后, 可使用此位重新初始化设。 例如, 如果 BUSY 位已置 1, 但因母线干扰而不能复位, 则可使用 SWRST 位退出此状态。	0
14	--	保留	--
13	ALERT	SMBus 报警 (SMBus alert) 此位由软件置 1 和清零, 并可在 PE=0 时由硬件清零 0: 释放 SMBA 引脚使其变成高电平。报警响应地址头后跟 NACK 1: 驱动 SMBA 引脚使其变成低电平。报警响应地址头后跟 ACK。	0
12	PEC	数据包错误校验 (Packet error checking) 此位由软件置 1 和清零, 并可在 PEC 传输完成时由硬件清零, 或者在 PE=0 时或在检测到起始位或停止位时由硬件清零。 0: 不传输 PEC 1: PEC 传输 (在 Tx 或 Rx 模式下) 注意: PEC 计算会因仲裁丢失而失效。	0
11	POS	应答/PEC 位置 (Acknowledge/PEC Position) (针对接收数据) 此位由软件置 1 和清零, 并可在 PE=0 时由硬件清零。 0: ACK 位控制移位寄存器中当前正在接收的字节的 (N)ACK。PEC 位指示移位寄存器中的当前字节是一个 PEC 1: ACK 位控制移位寄存器中要接收的下一个字节的 (N)ACK。PEC 位指示移位寄存器的下一个字节是一个 PEC 注意: POS 位只能用于主设备接收 2 个字节时。它必须在数据开始接收之前进行配置。	0
10	ACK	应答使能 (Acknowledge enable) 此位由软件置 1 和清零, 并可在 PE=0 时由硬件清零。 0: 不返回应答 1: 在接收一个字节 (匹配地址或数据) 之后返回应答	0
9	STOP	生成停止位 该位由软件置 1 和清零, 也可在检测到停止位时由硬件清零, 在检测到超时错误时由硬件置 1。 在主模式下: 0: 不生成停止位 1: 在传输当前字节或发送当前起始位后生成停止位 在从模式下: 0: 不生成停止位 1: 完成当前字节传输后释放 SCL 和 SDA 线	0
8	START	生成起始位 此位由软件置 1 和清零, 并可在起始位发送完成后或在从机模式下 PE=0 时由硬件清零。 在主模式下: 0: 不生成起始位	0

		1: 生成重复起始位 在从模式下: 0: 不生成起始位 1: 在总线空闲时生成起始位	
7	NOSTRETCH	禁止时钟延长 (Clock stretching disable) (从模式) 在从模式下, 当 ADDR 或 BTF 标志置 1 时, 此位用于禁止时钟延长, 直到软件将其复位为止。 0: 使能时钟延长 1: 禁止时钟延长	0
6	ENGCG	广播呼叫使能 (General call enable) 0: 禁止广播呼叫。不对地址 00h 应答 1: 使能广播呼叫。对地址 00h 应答	0
5	ENPEC	PEC 使能 (PEC enable) 0: 禁止 PEC 计算 1: 使能 PEC 计算	0
4	ENARP	ARP 使能 (ARP enable) 0: 禁止 ARP 1: 使能 ARP SMBTYPE=0 时识别 SMBus 器件默认地址 SMBTYPE=1 时识别 SMBus 主机地址	0
3	SMBTYPE	SMBus 类型 (SMBus type) 0: SMBus 器件 1: SMBus 主机	0
2	--	保留	--
1	SMBUS	SMBus 模式 (SMBus mode) 0: I2C 模式 1: SMBus 模式	0
0	PE	外设使能 (Peripheral enable) 0: 禁止外设 1: 使能外设 注意: 如果此位在通信进行过程中复位, 在结束本次通信后会带 IDLE 状态, 外设被禁止。 由于通信结束时 PE=0, 所有位均会复位 在主模式下, 此位不能在通信结束之前复位	0

注意: STOP、START 或 PEC 位置 1 之后, 在硬件将该位清零之前, 软件不能对 I2C_CR1 执行任何写操作。否则, 可能会再次发出 STOP、START 或 PEC 置位请求。

14.7.2 I2C 控制寄存器 2 (I2C_CR2)

位	符号	描述	复位值
15:13	--	保留	--
12	LAST	最后一次 DMA 传输 (DMA last transfer) 0: 下一个 DMA EOT 不是最后一次传输 1: 下一个 DMA EOT 是最后一次传输 注意: 此位用于主接收模式, 可对最后接收的数据生成 NACK。	0
11	DMAEN	DMA 请求使能 (DMA requests enable) 0: 禁止 DMA 请求 1: 当 TxE=1 或 RxNE =1 时使能 DMA 请求	0
10	ITBUFEN	缓冲中断使能 (Buffer interrupt enable) 0: TxE = 1 或 RxNE = 1 时不生成任何中断 1: TxE = 1 或 RxNE = 1 时生成事件中断 (与 DMAEN 状态无关)	0
9	ITEVTEN	事件中断使能 (Event interrupt enable) 0: 禁止事件中断 1: 使能事件中断 满足以下条件时将生成此中断: —SB = 1 (主模式) —ADDR = 1 (主/从模式) —ADD10= 1 (主模式) —STOPF = 1 (从模式) —BTF = 1, 无 TxE 或 RxNE 事件 —ITBUFEN = 1 且 TxE 事件置 1 —ITBUFEN = 1 且 RxNE 事件置 1	0
8	ITERREN	错误中断使能 (Error interrupt enable) 0: 禁止错误中断 1: 使能错误中断 满足以下条件时将生成此中断: —BERR = 1 —ARLO = 1 —AF = 1 —OVR = 1 —PECERR = 1 —TIMEOUT = 1 —SMBALERT = 1	0
7:6	--	保留	--
5:0	FREQ[5:0]	外设时钟频率 (Peripheral clock frequency) 外设时钟频率必须使用 APB 时钟频率进行配置 (I2C 外设连接到 APB)。允许的最小频率为 3 MHz, 最大频率则受限于 APB 最大频率 (48 MHz)。 0b000011: 3 MHz 0b000110: 6 MHz 0b001100: 12 MHz 0b010111: 24 MHz 0b101111: 48 MHz 其他: 不允许	0

14.7.3 I2C 自有地址寄存器 1 (I2C_OAR1)

位	符号	描述	复位值
15	ADDMODE	寻址模式 (Addressing mode) (从模式) 0: 7 位从地址 (无法应答 10 位地址) 1: 10 位从地址 (无法应答 7 位地址)	0
14:10	--	保留	--
9:8	ADD[9:8]	接口地址 7 位寻址 无意义 模式: 10 位寻址 地址的第 9:8 位 模式:	0
7:1	ADD[7:1]	接口地址 地址的第 7:1 位	0
0	ADD0	接口地址 7 位寻址 无意义 模式: 10 位寻址 地址的第 0 位 模式:	0

14.7.4 I2C 自有地址寄存器 2 (I2C_OAR2)

位	符号	描述	复位值
15:8	--	保留	--
7:1	ADD2[7:1]	接口地址 双寻址模式下的地址第 7:1 位	0
0	ENDUAL	双寻址模式使能 (Dual addressing mode enable) 0: 7 位寻址模式下仅对 OAR1 地址响应 1: 7 位寻址模式下能对 OAR1 和 OAR2 两个地址响应	0

14.7.5 I2C 数据寄存器 (I2C_DR)

位	符号	描述	复位值
15:8	--	保留	--
7:0	DR[7:0]	8 位数据寄存器 接收的字节或者要发送到总线的字节。 发送模式: 在 DR 寄存器中写入第一个字节时自动开始发送字节。如果在启动传送 (TxE=1) 后立即将下一个要传送的数据置于 DR 中, 则可以保持连续的传送流 接收模式: 将接收到的字节复制到 DR 中 (RxNE=1)。如果在接收下一个数据字节 (RxNE=1) 之前读取 DR, 则可保持连续的传送流 注意: 在从模式下, 地址并不会复制到 DR 中。 注意: 硬件不对写冲突进行管理 (TxE=0 时也可对 DR 执行写操作)。 注意: 如果发出 ACK 脉冲时出现 ARLO 事件, 则不会将接收到的字节复制到 DR 寄存器, 因而也无法读取字节	0

14.7.6 I2C 状态寄存器 1 (I2C_SR1)

位	符号	描述	复位值
15	SMBALERT	SMBus 报警 (SMBus alert) 在 SMBus 主机模式下: 0: 无 SMBALERT 1: 引脚上发生 SMBALERT 事件 在 SMBus 从机模式下: 0: 无 SMBALERT 响应地址头 1: 接收到指示 SMBALERT 低电平 SMBALERT 响应地址头 —由软件写入 0 来清零, 或在从机模式 PE=0 时由硬件清零。	0
14	TIMEOUT	超时或 Tlow 错误 (Timeout or Tlow error) 0: 无超时错误 1: SCL 低电平时长持续 25 ms (超时) 或主器件累计时钟低电平延长时间超过 10 ms (Tlow:mext) 或从器件累计时钟低电平延长时间超过 25 ms (Tlow:sext) — 在从模式下置 1 时: 从器件复位通信且硬件释放数据线 — 在主模式下置 1 时: 由硬件发送停止位 — 由软件写入 0 来清零, 或在从机模式 PE=0 时由硬件清零。 注意: 此功能仅在 SMBus 模式下可用。	0
13	--	保留	--
12	PECERR	接收期间 PEC 错误 (PEC Error in reception) 0: 无 PEC 错误: 接收器在接收 PEC 后返回 ACK (如果 ACK=1) 1: PEC 错误: 接收器在接收 PEC 后返回 NACK (无论 ACK 什么值) —由软件写入 0 来清零, 或在 PE=0 时由硬件清零。 —注意: 接收到错误的 CRC 时, 如果在结束 CRC 接收之前 PEC 控制位没有置 1, 则 PECERR 位在从模式下不会置 1。不过可以通过读取 PEC 值来判定接收到的 CRC 是否正确。	0
11	OVR	上溢/下溢 (Overrun/Underrun) 0: 未发生上溢/下溢 1: 上溢或下溢 —在从模式下由硬件置 1, 前提是满足 NOSTRETCH=1 且: —接收过程中接收到一个新字节 (包括 ACK 脉冲) 但尚未读取 DR 寄存器。新接收的字节将丢失。 —发送过程中将发送一个新字节但尚未向 DR 寄存器写入数据。同一字节发送两次。 —由软件写入 0 来清零, 或在 PE=0 时由硬件清零。 注意: 如果 DR 写操作时间与出现 SCL 上升沿的时间非常接近, 则发出的数据不确定, 并且出现数据保持时间错误	0
10	AF	应答失败 (Acknowledge failure) 0: 未发生应答失败 1: 应答失败 —无应答返回时由硬件置 1。 —由软件写入 0 来清零, 或在从机模式 PE=0 时由硬件清零。	0
9	ARLO	仲裁丢失 (Arbitration lost) (主模式) 0: 未检测到仲裁丢失	0

		1: 检测到仲裁丢失 —当接口在竞争总线是输给另一个主设备时, 由硬件将该位置 1 —由软件写入 0 来清零, 或在 PE=0 时由硬件清零。 —发生 ARLO 事件后, 接口会自动切换回从模式 (M/SL=0)。 注意: 在 SMBUS 中, 从模式下的数据仲裁仅发生在数据阶段或发送确认期间 (不适用于地址确认)	
8	BERR	总线错误 (Bus error) 0: 无误放的起始或停止位 1: 存在误放的起始或停止位 —SCL 为高电平时, 若接口在字节传输期间检测到某个无效位置出现 SDA 上升沿或下降沿, 则会由硬件将该位置 1。 —由软件写入 0 来清零, 或在从机模式 PE=0 时由硬件清零。	0
7	TxE	数据寄存器为空 (Data register empty) (发送器) 0: 数据寄存器非空 1: 数据寄存器为空 —发送过程中 DR 为空时该位置 1。TxE 不会在地址阶段置 1。 —由软件写入 DR 寄存器来清零, 或在出现起始、停止位或者在从机模式 PE=0 时由硬件清零。如果接收到 NACK 或要发送的下一个字节为 PEC (PEC=1), TxE 将不会置 1 注意: 写入第一个要发送的数据或在 BTF 置 1 时写入数据都无法将 TxE 清零, 因为这两种情况下数据寄存器仍为空。	0
6	RxNE	数据寄存器非空 (Data register not empty) (接收器) 0: 数据寄存器为空 1: 数据寄存器非空 —接收模式下数据寄存器非空时置 1。RxNE 不会在地址阶段置 1。 —由软件读取或写入 DR 寄存器来清零, 或在 PE=0 时由硬件清零。 发生 ARLO 事件时 RxNE 不会置 1。 注意: BTF 置 1 时无法通过读取数据将 RxNE 清零, 因为此时数据寄存器仍为满。	0
5	--	保留	--
4	STOPF	停止位检测 (Stop detection) (从模式) 0: 未检测到停止位 1: 检测到停止位 —从设备在应答脉冲后 (如果 ACK=1) 检测到停止位, 由硬件置 1。 —由软件分别对 SR1 寄存器和 CR1 寄存器执行读操作和写操作来清零, 或在 PE=0 时由硬件清零。 注意: 收到 NACK 后 STOPF 位不会置 1。 建议在 STOPF 置 1 后执行完整的清零序列 (首先读取 SR1, 然后写入 CR1)。	0
3	ADD10	发送 10 位头 (主模式) 0: 未发生 ADD10 事件 1: 主器件已发送第一个地址字节 (头) —主器件在 10 位地址模式下已发送第一个字节时由硬件置 1。 —由软件在读取 SR1 寄存器后在 DR 寄存器中写入第二个地址字节来清零, 或在 PE=0 时由硬件清零。 注意: 收到 NACK 后 ADD10 位不会置 1	0
2	BTF	字节传输完成 (Byte transfer finished) 0: 数据字节传输未完成 1: 数据字节传输成功完成 —由硬件置 1, 前提是满足 NOSTRETCH=0 且: —接收过程中接收到一个新字节 (包括 ACK 脉冲) 但尚未读取	0

		DR 寄存器 (RxNE=1)。 —发送过程中将发送一个新字节但尚未向 DR 寄存器写入数据 (TxNE=1)。 —由软件读或写 DR 寄存器来清零，或在发送过程中出现起始或停止位后由硬件清零，也可以在从机模式 PE=0 时由硬件清零。 注意：收到 NACK 后 BTF 位不会置 1 如果下一个要发送的字节为 PEC (I2C_SR2 寄存器中的 TRA=1, I2C_CR1 寄存器中的 PEC=1)，则 BTF 位不会置 1	
1	ADDR	地址已发送 (主模式)/地址匹配 (从模式) 由软件在读取 SR1 寄存器后读取 SR2 寄存器来清零，或在从机模式 PE=0 时由硬件清零。 地址匹配 (从模式)： 0: 地址不匹配或未接收到地址 1: 接收到的地址匹配 —当接收到的从地址与 OAR 寄存器内容、广播呼叫地址或 SMBus 器件默认地址匹配时，或者识别到 SMBus 主机或 SMBus 报警时，该位由硬件置 1。（根据配置确定何时使能）。 注意：在从模式下，建议在 ADDR 置 1 后执行完整的清零序列（首先读取 SR1，然后写入 SR2）。	0
		地址已发送 (主模式) 0: 地址发送未结束 1: 地址发送结束 —在 10 位寻址模式下，接收到第二个地址字节的 ACK 后该位置 1。 —在 7 位寻址模式下，接收到地址字节的 ACK 后该位置 1。 注意：收到 NACK 后 ADDR 位不会置 1	0
0	SB	起始位 (Start bit) (主模式) 0: 无起始位 1: 起始位已经发送 —生成启动条件时置 1。 —由软件在读取 SR1 寄存器后写入 DR 寄存器来清零，或在 PE=0 时由硬件清 0	0

14.7.7 I2C 状态寄存器 2 (I2C_SR2)

位	符号	描述	复位值
15:8	PEC[7:0]	数据包错误校验寄存器 ENPEC=1 时, 此寄存器包含内部 PEC	0
7	DUALF	双标志 (Dual flag) (从模式) 0: 接收到的地址与 OAR1 匹配 1: 接收到的地址与 OAR2 匹配 —出现停止位、重复起始位或 PE=0 时由硬件清零。	0
6	SMBHOST	SMBus 主机头 (SMBus host header) (从模式) 0: 无 SMBus 主机地址 1: SMBTYPE=1 且 ENARP=1 时接收到 SMBus 主机地址 —出现停止位、重复起始位或 PE=0 时由硬件清零。	0
5	SMBDEFAULT	SMBus 器件默认地址 (SMBus device default address) (从模式) 0: 无 SMBus 器件默认地址 1: ENARP=1 时接收到 SMBus 器件默认地址 —出现停止位、重复起始位或 PE=0 时由硬件清零。	0
4	GENCALL	广播呼叫地址 (General call address) (从模式) 0: 无广播呼叫 1: ENGCL=1 时接收到广播呼叫地址 —出现停止位、重复起始位或 PE=0 时由硬件清零。	0
3	--	保留	--
2	TRA	发送器/接收器 (Transmitter/receiver) 0: 接收器 1: 发送器 此位在整个地址阶段的结尾处根据地址字节的 R/W 位状态进行置 1。 同样, 检测到停止位 (STOPF=1)、重复起始位、总线仲裁丢失 (ARLO=1) 或当在从机模式 PE=0 时该位 也由硬件清零。	0
1	BUSY	总线忙碌 (Bus busy) 0: 总线上无通信 1: 总线正在进行通信 —检测到 SDA 或 SCL 低电平时由硬件置 1 —检测到停止位时由硬件清零。 该位指示总线上是否正在进行通信。即使禁止接口 (PE=0) 后此信息也会更新。	0
0	MSL	主/从模式 (Master/slave) 0: 从模式 1: 主模式 —接口进入主模式时 (SB=1) 由硬件置 1。 —检测到总线上的停止位、仲裁丢失 (ARLO=1) 或当在从机模式 PE=0 时由硬件清零	0

注意: 读取 I2C_SR1 后再读取 I2C_SR2 可将 ADDR 标志清零, 即使 ADDR 标志在读取 I2C_SR1 之后置 1 也如此。因此, 必须仅在 I2C_SR1 中的 ADDR 位已置 1 或者 STOPF 位已清零后 读取 I2C_SR2。

14.7.8 I2C 时钟控制寄存器 (I2C_CCR)

注意：F_{PCLK} 必须至少为 2 MHz，才能达到标准模式 I²C 频率。必须至少为 4 MHz 才能达到快速模式 I²C 频率。必须为 10MHz 的倍数才能实现最大 400 KHz 的 I²C 快速模式时钟。CCR 寄存器必须仅在禁止 I2C (PE = 0) 的情况下配置。

位	符号	描述	复位值
15	F/S	I2C 主模式选择 (I2C master mode selection) 0: 标准模式 I2C 1: 快速模式 I2C	0
14	DUTY	快速模式占空比 (Fast mode duty cycle) 0: 快速模式 T _{low} /T _{high} = 2 1: 快速模式 T _{low} /T _{high} = 16/9 (参见 CCR)	0
13:12	--	保留	--
11:0	CCR[11:0]	快速/标准模式下的时钟控制寄存器 (Clock control register in Fast/Standard mode) (主模式) 控制主模式下的 SCL 时钟。 标准模式或 SMBus 模式: Thigh = CCR * T _{PCLK} Tlow = CCR * T _{PCLK} 快速模式 如果 DUTY = 0: Thigh = CCR * T _{PCLK} Tlow = 2 * CCR * T _{PCLK} 如果 DUTY = 1: (达到 400 KHz) Thigh = 9 * CCR * T _{PCLK} Tlow = 16 * CCR * T _{PCLK} 例如: 要在标准模式下生成 100 KHz 的 SCL 频率: 如果 APB 最大频率为 8MHz, T _{PCLK} = 125 ns, 则必须将 CCR 编程为 0x28(0x28 <=> 40d x 125 ns = 5000 ns)。 注意: • 允许的最小值为 0x04, 但快速模式 (占空比 9/16) 例外, 其最小值为 0x01 • 这些时间均未经滤波 • CCR 寄存器必须仅在禁止 I2C (PE = 0) 的情况下配置	0

14.7.9 I2C TRISE 寄存器 (I2C_TRISE)

位	符号	描述	复位值
15:6	--	保留	--
5:0	TRISE[5:0]	快速/标准模式下的最大上升时间 (主模式) 这些位必须编程为 I2C 总线规范中给定的最大 SCL 上升时间加 1。 例如: 标准模式下允许的最大 SCL 上升时间为 1000 ns。 如果 APB 最大频率为 8MHz 且 T _{PCLK} = 125 ns, 则 TRISE[5:0] 位必须编程为 0x09。 (1000 ns / 125 ns = 8 + 1) 滤波器值也可以叠加到 TRISE[5:0]。 如果结果不为整数, 则 TRISE[5:0] 必须编程为整数部分, 以符合 t _{HIGH} 参数要求。 注意: TRISE[5:0] 必须仅在禁止 I2C (PE = 0) 的情况下配置。	2

14.7.10 I2C FLTR 寄存器 (I2C_FLTR)

位	符号	描述	复位值
15:5	--	保留	--
4	ANOFF	模拟噪声滤波器关闭 (Analog noise filter OFF) 0: 使能模拟噪声滤波器 1: 禁止模拟噪声滤波器 注意: ANOFF 必须仅在禁止 I2C (PE = 0) 的情况下配置。	0
3:0	DNF[3:0]	数字噪声滤波器 (Digital noise filter) 这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可抑制脉宽达 $DNF[3:0] * T_{PCLK}$ 以下的尖峰。 0000: 禁止数字噪声滤波器 0001: 使能数字噪声滤波器, 滤波能力可达 $1 * T_{PCLK}$... 1111: 使能数字噪声滤波器, 滤波能力可达 $15 * T_{PCLK}$ 注意: DNF[3:0] 必须仅在禁止 I2C (PE = 0) 的情况下配置。如果模拟滤波器也已使能, 则需将 数字滤波器添加到模拟滤波器中	0

第15章 串行同步异步通信单元 USART

15.1 USART 介绍

通用同步异步收发器 (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提供了多种波特率。

它支持同步单向通信和半双工单线通信; 还支持 LIN (局域互连网络)、IrDA (红外线数据协会) SIR ENDEC 规范, 以及调制解调器操作 (CTS/RTS)。而且, 它还支持多处理器通信。

通过配置多个缓冲区使用 DMA 可实现高速数据通信。

15.2 USART 主要特性

- 全双工异步通信
- NRZ 标准格式 (标记/空格)
- 小数波特率发生器系统
 - 通用可编程收发波特率。
- 数据字长度可编程 (8 位或 9 位)
- 停止位可配置 - 支持 1 或 2 个停止位
- LIN 主模式同步停止符号发送功能和 LIN 从模式停止符号检测功能
 - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- 用于同步发送的发送器时钟输出
- IrDA SIR 编码解码器
 - 正常模式下, 支持 3/16 位持续时间
- 单线半双工通信
- 使用 DMA (直接存储器访问) 实现可配置的多缓冲区通信
 - 使用 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发送器和接收器具有单独使能位
- 传输检测标志:
 - 接收缓冲区已满
 - 发送缓冲区为空
 - 传输结束标志
- 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
- 四个错误检测标志:
 - 溢出错误
 - 噪声检测
 - 帧错误
 - 奇偶校验错误
- 十个具有标志位的中断源:
 - CTS 变化

- LIN 停止符号检测
- 发送数据寄存器为空
- 发送完成
- 接收数据寄存器已满
- 接收到线路空闲
- 溢出错误
- 帧错误
- 噪声错误
- 奇偶校验错误
- 多处理器通信 - 如果地址不匹配，则进入静默模式
- 从静默模式唤醒（通过线路空闲检测或地址标记检测）
- 两个接收器唤醒模式：地址位（MSB，第 9 位），线路空闲

15.3 USART 功能描述

接口通过三个引脚从外部连接到其它设备（请参见图 15-1）。任何 USART 双向通信均需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据引脚输出 (TX)：

RX：接收数据输入引脚就是串行数据输入引脚。过采样技术可区分有效输入数据和噪声，从而用于恢复数据。

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（USART 电平下，随后在 SW_RX 上接收数据）。

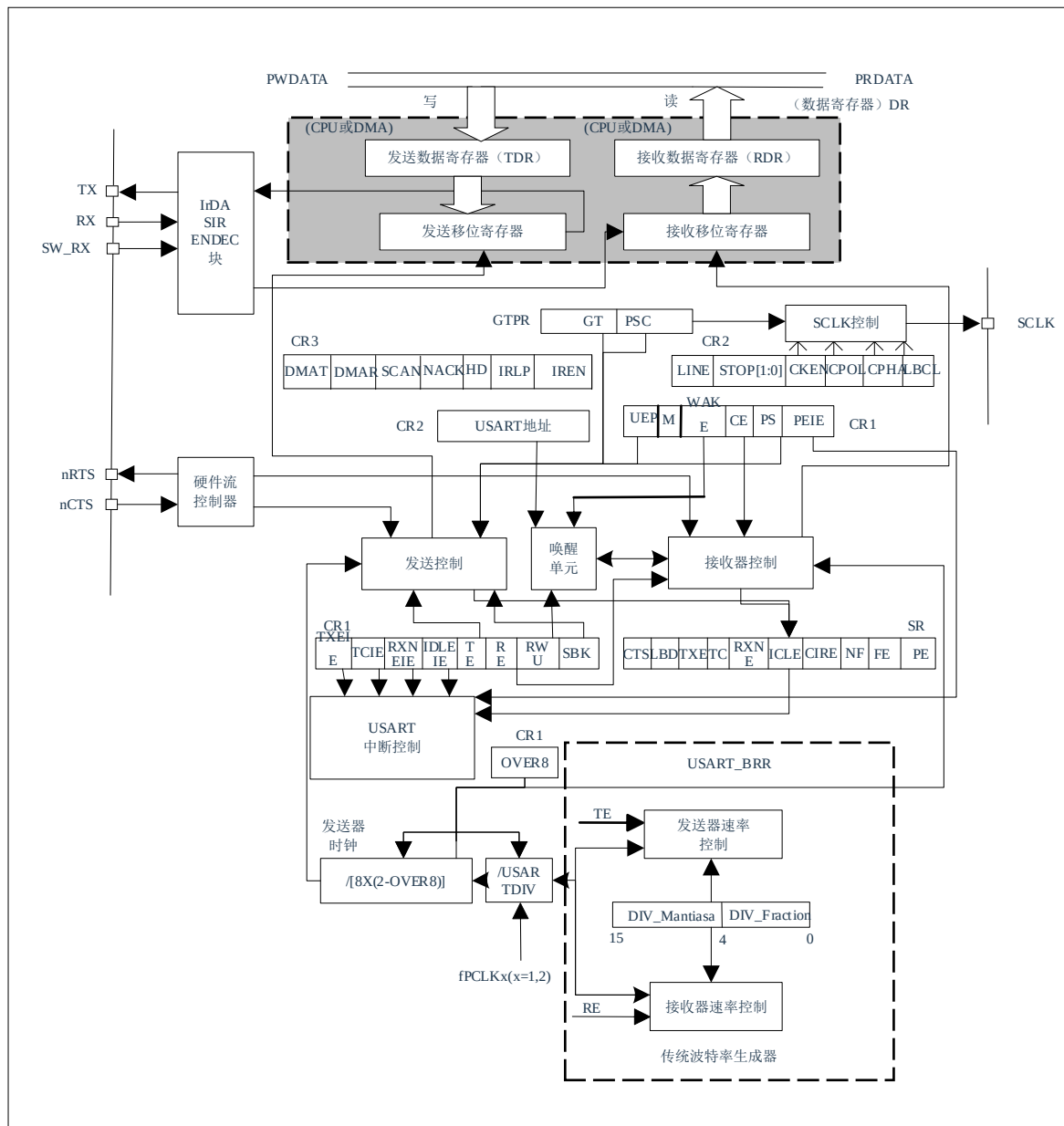
正常 USART 模式下，通过这些引脚以帧的形式发送和接收串行数据：

- 发送或接收前保持空闲线路
- 起始位
- 数据（字长 8 位或 9 位），最低有效位在前
- 用于指示帧传输已完成的 1 个、2 个停止位
- 该接口使用小数波特率发生器 - 带 12 位尾数和 4 位小数
- 状态寄存器 (USART_SR)
- 数据寄存器 (USART_DR)
- 波特率寄存器 (USART_BRR) - 12 位尾数和 4 位小数。

在同步模式下连接时需要以下引脚：

- **SCLK：**发送器时钟输出。该引脚用于输出发送器数据时钟，以便按照 SPI 主模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。RX 上可同步接收并行数据。这一点可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。在硬件流控制模式下需要以下引脚：
- **nCTS：**“清除以发送”用于在当前传输结束时阻止数据发送（高电平时）
- **nRTS：**“请求以发送”用于指示 USART 已准备好接收数据（低电平时）。

图 15-1 USART 框图



15.3.1 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位进行编程来选择 8 位或 9 位的字长（请参见图 15-2）。TX 引脚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。

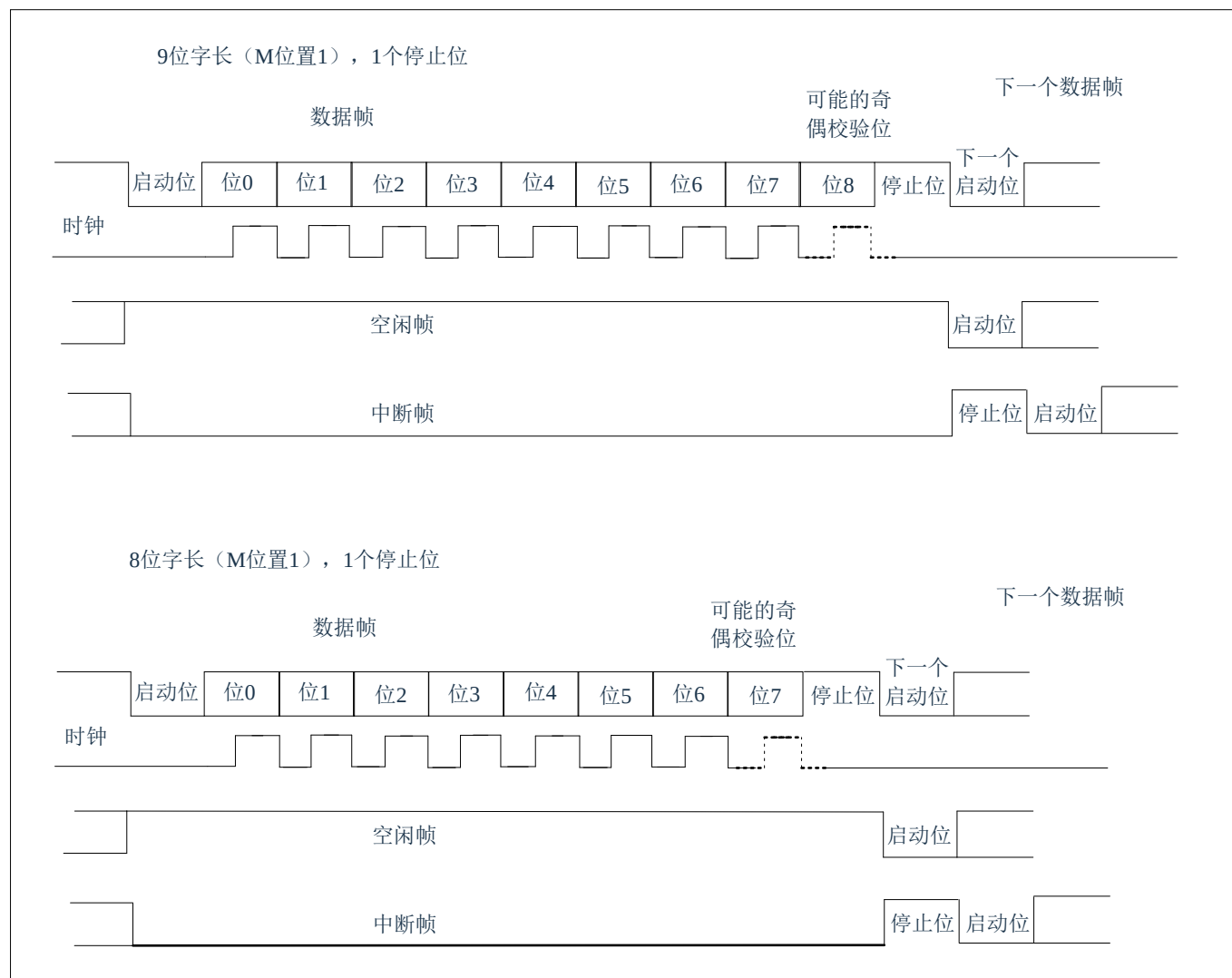
空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”），该字符后是下一个数据帧的起始位。

停止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 1 或 2 个停止位（逻辑“1”位）以确认起始位。

发送和接收由通用波特率发生器驱动，发送器和接收器的使能位分别置 1 时将生成相应的发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 15-2 字长编程



15.3.2 发送器

发送器可发送 8 位或 9 位的数据字，具体取决于 M 位的状态。发送使能位 (TE) 置 1 时，发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 SCLK 引脚输出。

15.3.2.1 字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位。该模式下，USART_DR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间（请参见图 15-1）。

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

USART 支持以下停止位：1 和 2 个停止位。

注 1：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，从而将损坏 TX 引脚上的数据。当前传输的数据将会丢失。

注 2：使能 TE 位后，将会发送空闲帧。

15.3.2.2 可配置的停止位

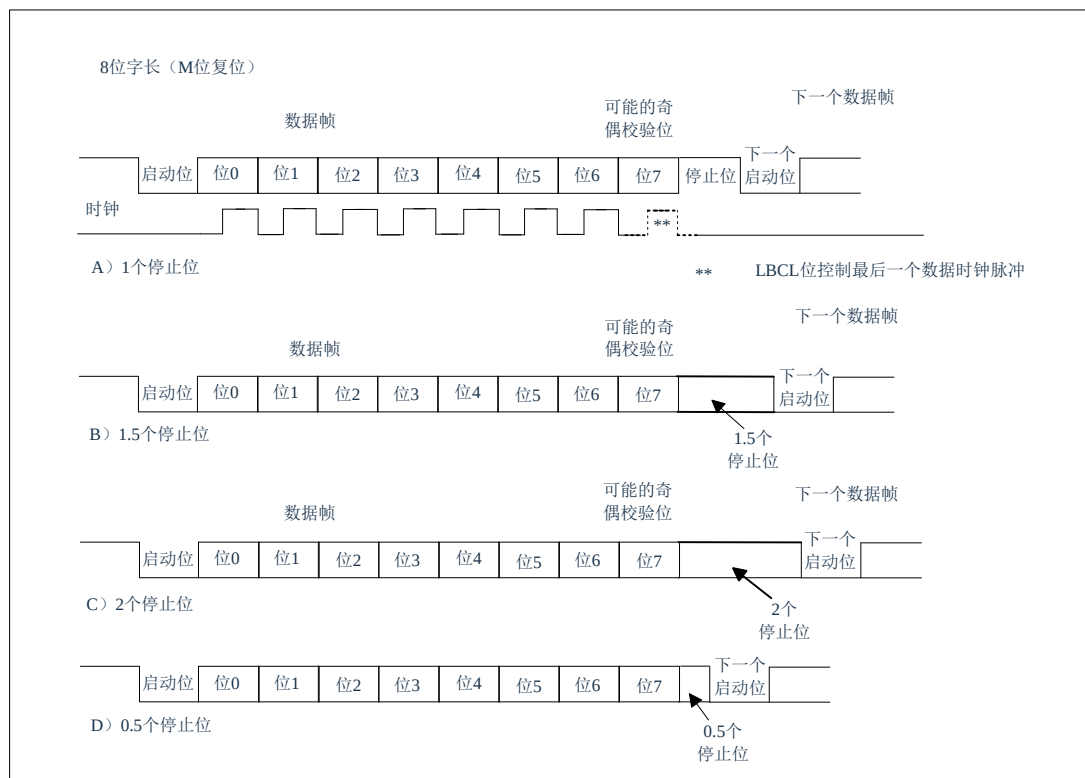
可以在控制寄存器 2 的位 13 编程将随各个字符发送的停止位的数量。

- 1 个停止位：这是停止位数量的默认值。
- 2 个停止位：正常 USART 模式、单线模式和调制解调器模式支持该值。

空闲帧发送将包括停止位。

$m = 0$ 时，中断发送是 10 个低电平位，然后是已配置数量的停止位； $m = 1$ 时，中断发送是 11 个低电平位，然后是已配置数量的停止位。无法传送长中断（中断长度大于 10/11 个低电平位）。

图 15-3 可配置的停止位



步骤:

1. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
2. 对 USART_CR1 中的 M 位进行编程以定义字长。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照多缓冲区通信中的解释说明配置 DMA 寄存器。
5. 使用 USART_BRR 寄存器选择所需波特率。
6. 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 USART_DR 寄存器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。
8. 向 USART_DR 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 USART 或进入暂停模式时需要此步骤，以避免损坏最后一次发送。

15.3.2.3 单字节通信

始终通过向数据寄存器写入数据来将 TXE 位清零。

TXE 位由硬件置 1，它表示：

- 数据已从 TDR 移到移位寄存器中且数据发送已开始。
- TDR 寄存器为空。
- USART_DR 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。

发送时，要传入 USART_DR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

未发送时，要传入 USART_DR 寄存器的写指令直接将数据置于移位寄存器中，数据发送开始时，TXE 位立即置 1。

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 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

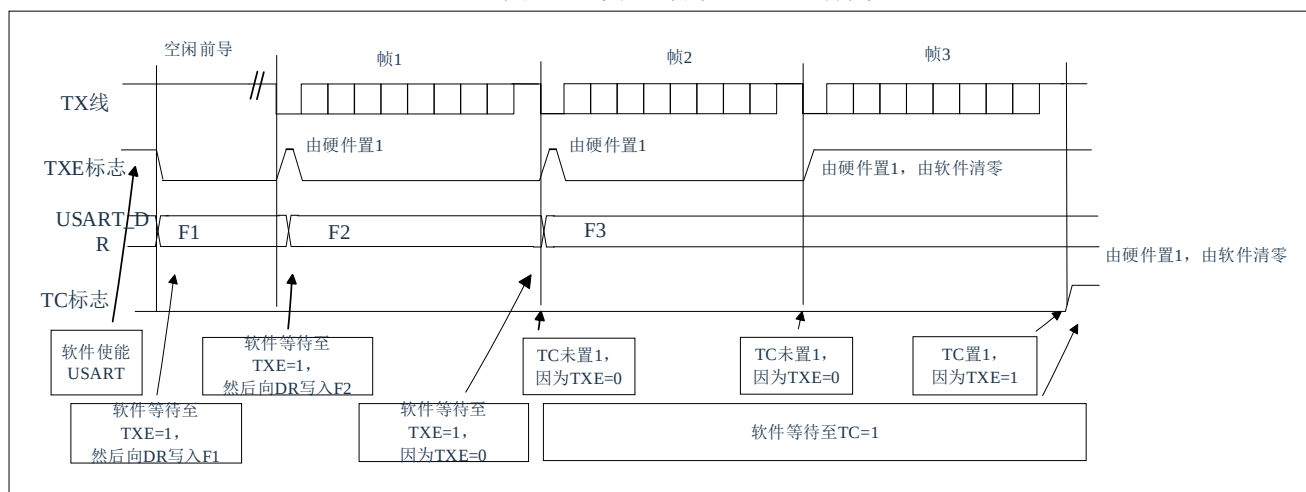
向 USART_DR 寄存器中写入最后一个数据后，必须等待至 TC=1，之后才可禁止 USART 或使微控制器进入低功耗模式（请参见图 15-4：发送时的 TC/TXE 行为）。

TC 位通过以下软件序列清零：

1. 从 USART_SR 寄存器读取数据
2. 向 USART_DR 寄存器写入数据

注意：还可通过向 TC 位写入“0”将其清零。建议仅在多缓冲区通信时使用此清零序列。

图 15-4 发送时的 TC/TXE 行为



15.3.2.4 中断字符

将 SBK 位置 1 将发送一个中断字符。中断帧的长度取决于 M 位（请参见图 15-2）。

如果 SBK 位置“1”，当前字符发送完成后，将在 TX 线路上发送一个中断字符。中断字符发送完成时（发送中断字符的停止位期间），该位由硬件复位。USART 在上一个中断帧的末尾插入一个逻辑“1”位，以确保识别下个帧的起始位。

注意：如果软件在中断发送开始前对 SBK 位进行了复位，将不会发送中断字符。对于两个连续的中断，应在上一个中断的停止位发送完成后将 SBK 位置 1。

15.3.2.5 空闲符号

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

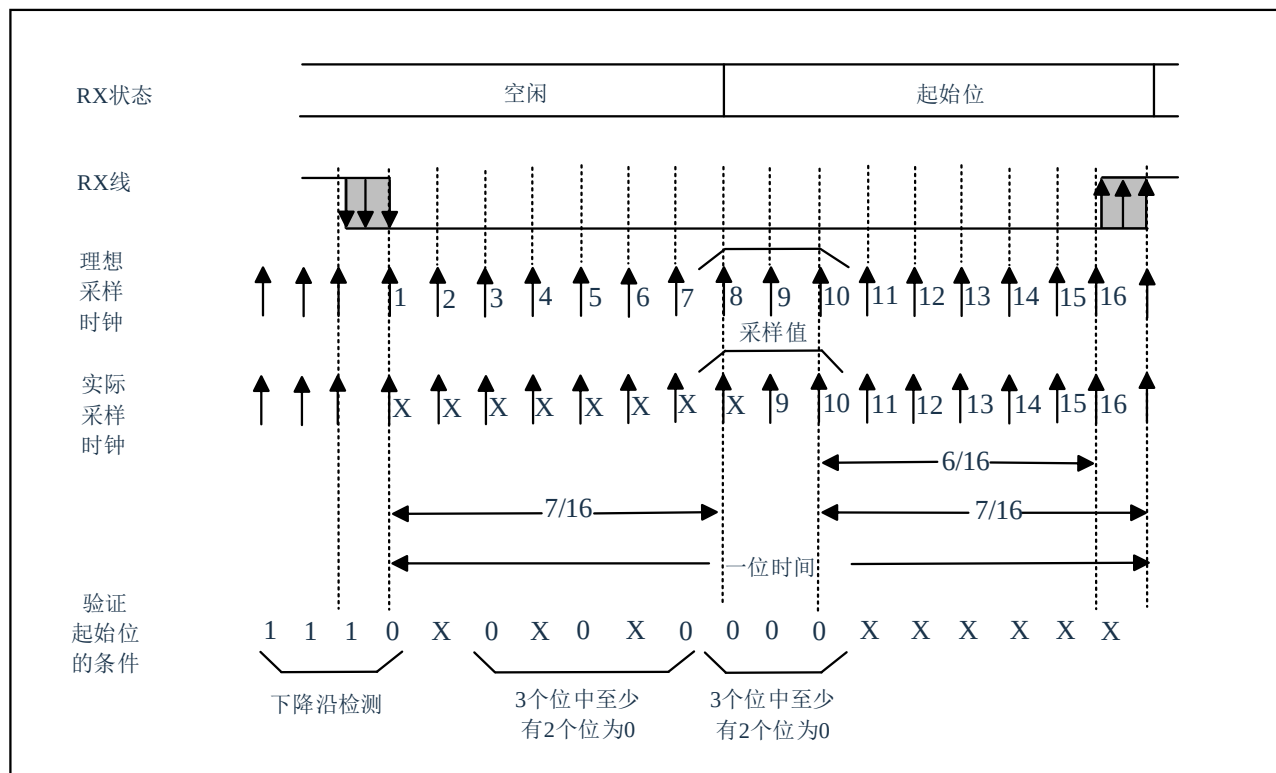
15.3.3 接收器

USART 可接收 8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

15.3.3.1 开始位检测

在 USART 中，识别出特定序列的采样时会检测起始位。该序列为：1 1 1 0 X 0 X 0 X 0 0 0 0。

图 15-5 16 倍过采样时的起始位检测



注意：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 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（RXNE 标志位置 1，RXNEIE=1 时生成中断）。

如果两次采样时（对第 3 位、第 5 位和第 7 位进行采样以及对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中至少有 2 个为 0，则可验证起始位（RXNE 标志位置 1，RXNEIE=1 时生成中断）但 NE 噪声标志位置 1。如果不满足此条件，则启动检测中止，接收器返回空闲状态（无标志位置 1）

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，则可验证起始位但 NE 噪声标志位置 1。

15.3.3.2 字符接收

USART 接收期间，首先通过 RX 引脚移入数据的最低有效位。该模式下，USART_DR 寄存器的缓冲区 (RDR) 位于内部总线和接收移位寄存器之间。

步骤：

1. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
2. 对 USART_CR1 中的 M 位进行编程以定义字长。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照多缓冲区通信中的解释说明配置 DMA 寄存器。
5. 使用波特率寄存器 USART_BRR 选择所需波特率
6. 将 RE 位 USART_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。这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）。
- 如果 RXNEIE 位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误或上溢错误，错误标志位可置 1。
- 在多缓冲区模式下，每接收到一个字节后 RXNE 均置 1，然后通过 DMA 对数据寄存器执行读操作清零。
- 在单缓冲区模式下，通过软件对 USART_DR 寄存器执行读操作将 RXNE 位清零。RXNE 标志也可以通过向该位写入零来清零。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。

注意：接收数据时，不应将 RE 位复位。如果接收期间禁止了 RE 位，则会中止接收当前字节。

15.3.3.3 中断字符

接收到中断字符时，USART 将会按照帧错误对其进行处理。

15.3.3.4 空闲字符

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 位为 1，则会产生中断。

15.3.3.5 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。当 RXNE 标志位是 1 时，如果再接收到下一数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：

- ORE 位置 1。
- RDR 中的内容不会丢失。对 USART_DR 执行读操作时可使用先前的数据。
- 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 与 DMAR 位均为 1，则会生成中断。
- 通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 ORE 位清除。

注意：ORE 位置 1 时表示至少 1 个数据丢失。存在两种可能：

- 如果 RXNE=1，则最后一个有效数据存储于接收寄存器 RDR 中并且可进行读取；
- 如果 RXNE=0，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。读取序列期间（在 USART_SR 寄存器读访问与 USART_DR 读访问之间）接收到新数据时也会发生该情况。

15.3.3.6 选择适合的过采样方法

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。可通过编程 USART_CR1 寄存器中的 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（请参见图 15-6 和图 15-7）。

注意：此产品不支持 8 倍过采样。

根据应用：

- 选择 8 倍过采样 (OVER8=1) 以获得更高的速度（高达 $f_{PCLK}/8$ ）。这种情况下接收器对时钟偏差的最大容差将会降低（请参见第 15.3.5 节：USART 接收器对时钟偏差的容差）。
- 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 $f_{PCLK}/16$ 。

可通过编程 USART_CR3 寄存器中的 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。

根据应用：

- 在噪声环境下工作时，请选择三次采样的多数表决法 (ONEBIT=0)；在检测到噪声时请拒绝数据（请参见表 15-1），因为这表示采样过程中产生了干扰。
- 线路无噪声时请选择单次采样法 (ONEBIT=1) 以增加接收器对时钟偏差的容差（请参见 USART 接收器对时钟偏差的容差）。这种情况下 NF 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位的上升沿时 NF 位置 1。
- 无效数据从移位寄存器传送到 USART_DR 寄存器。

- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 NF 位清零。

注意：IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位由硬件强制清零

图 15-6：16 倍过采样时的数据采样

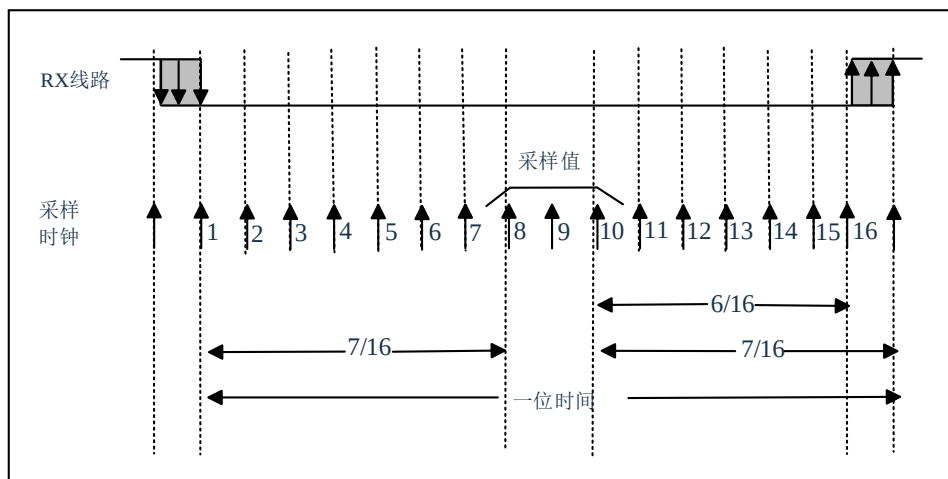


图 15-7：8 倍过采样时的数据采样

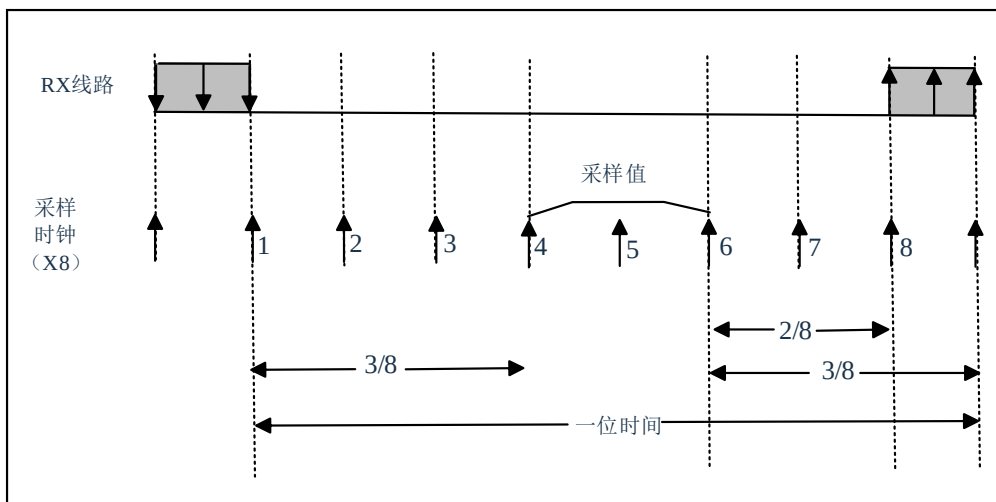


表 15-1 通过采样数据进行噪声检测

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

15.3.3.7 帧错误

以下情况下将检测到帧错误：

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。

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 USART_DR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 FE 位清零。

15.3.3.8 接收期间的可配置的停止位

可通过控制寄存器 2 中的控制位配置要接收的停止位的数量可以是 1 或 2 个（正常模式下）。

1. 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. 2 个停止位：采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。如果在第一个停止位期间检测到帧错误，则帧错误标志位将会置 1。发生帧错误时不检测第 2 个停止位。RXNE 标志将在第一个停止位末尾时置 1。

15.3.4 小数波特率的产生

对 USARTDIV 的尾数值和小数值进行编程时，接收器和发送器（Rx 和 Tx）的波特率均设置为相同值。

公式 1:适用于标准 USART(包括 SPI 模式)的波特率

$$\text{Tx/Rx 波特率} = f_{\text{CK}} / (8 \times (2 - \text{OVER8}) \times \text{USARTDIV})$$

公式 2: LIN 和 IrDA 模式下的波特率

$$\text{Tx/Rx 波特率} = f_{\text{CK}} / (16 \times \text{USARTDIV})$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8=0 时，小数部分编码为 4 位并通过 USART_BRR 寄存器中的 DIV_fraction[3:0] 位编程。
- 当 OVER8=1 时，小数部分编码为 3 位并通过 USART_BRR 寄存器中的 DIV_fraction[2:0] 位编程，此时 DIV_fraction[3] 位必须保持清零状态。

注意：对 USART_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。

OVER8=0 时如何从 USART_BRR 寄存器值中获取 USARTDIV：

示例 1：

如果 DIV_Mantissa = 0d27 且 DIV_Fraction = 0d12 (USART_BRR = 0x1BC)，
则尾数 (USARTDIV) = 0d27，小数 (USARTDIV) = 12/16 = 0d0.75，
因此 USARTDIV = 0d27.75。

示例 2：

要设定 USARTDIV = 0d25.62，
这将导致：DIV_Fraction = 16*0d0.62 = 0d9.92，
最接近的实数为 0d10 = 0xA，
DIV_Mantissa = 尾数 (0d25.620) = 0d25 = 0x19，
则 USART_BRR = 0x19A，因此 USARTDIV = 0d25.625。

示例 3：

要设定 USARTDIV = 0d50.99，
这将导致：DIV_Fraction = 16*0d0.99 = 0d15.84，
最接近的实数为 0d16 = 0x10 => DIV_frac[3:0] 溢出 => 尾数必须添加进位，
DIV_Mantissa = 尾数 (0d50.990 + 进位) = 0d51 = 0x33，
则 USART_BRR = 0x330，因此 USARTDIV = 0d51.000，

OVER8=1 时如何从 USART_BRR 寄存器中获取 USARTDIV：

示例 1：

如果 DIV_Mantissa = 0x27 且 DIV_Fraction[2:0]= 0d6 (USART_BRR = 0x1B6)，则
尾数 (USARTDIV) = 0d27，
小数 (USARTDIV) = 6/8 = 0d0.75，
因此 USARTDIV = 0d27.75。

示例 2:

要设定 USARTDIV = 0d25.62。

这将导致: $DIV\_Fraction = 8 * 0d0.62 = 0d4.96$,

最接近的实数为 $0d5 = 0x5$,

$DIV\_Mantissa = \text{尾数} (0d25.620) = 0d25 = 0x19$,

则 $USART\_BRR = 0x195 \Rightarrow USARTDIV = 0d25.625$ 。

示例 3:

要设定 USARTDIV = 0d50.99

这将导致: $DIV\_Fraction = 8 * 0d0.99 = 0d7.92$,

最接近的实数为 $0d8 = 0x8 \Rightarrow DIV\_frac[2:0]$ 溢出 $\Rightarrow$ 尾数必须添加进位,

$DIV\_Mantissa = \text{尾数} (0d50.990 + \text{进位}) = 0d51 = 0x33$,

则 $USART\_BRR = 0x0330 \Rightarrow USARTDIV = 0d51.000$ 。

表 15-2: 采用 16 倍过采样时, 在 $f_{PCLK}=12\text{MHz}$ 下编程波特率时的误差计算⁽¹⁾

波特率		$f_{PCLK}=12\text{MHz}$		
序号	所需值	实际值	波特率寄存器中编程的值	误差%
1	1.2KBps	1.2KBps	625	0
2	2.4KBps	2.4KBps	312.5	0
3	9.6KBps	9.6KBps	78.125	0
4	19.2KBps	19.2KBps	39.0625	0
5	38.4KBps	38.336KBps	19.5625	0.16
6	57.6KBps	57.692KBps	13	0.16
7	115.2KBps	115.385KBps	6.5	0.16
8	230.4KBps	230.769KBps	3.25	0.16
9	460.8KBps	461.538KBps	1.625	0.16
10	921.6KBps	NA	NA	NA
11	2MBps	NA	NA	NA
12	3MBps	NA	NA	NA

注 1: CPU 时钟越低, 特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 15-3: 采用 16 倍过采样时, 在 $f_{PCLK}=24\text{MHz}$ 下编程波特率时的误差计算⁽¹⁾

波特率		$f_{PCLK}=24\text{MHz}$		
序号	所需值	实际值	波特率寄存器中编程的值	误差%
1	1.2KBps	1.2	1250	0
2	2.4KBps	2.4	625	0
3	9.6KBps	9.6	156.25	0
4	19.2KBps	19.2	78.125	0
5	38.4KBps	38.4	39.0625	0
6	57.6KBps	57.554	26.0625	0.08
7	115.2KBps	115.385	13	0.16
8	230.4KBps	230.769	6.5	0.16
9	460.8KBps	461.538	3.25	0.16
10	921.6KBps	923.077	1.625	0.16
11	2MBps	NA	NA	NA
12	3MBps	NA	NA	NA

注 1: CPU 时钟越低, 特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

15.3.5 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。

影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

$DTRA + DQUANT + DREC + DTCL < \text{USART 接收器的容差}$ 。

对于正常接收数据，USART 接收器的容差等于所容许的最大偏差，具体取决于以下选择：

- 由 USART_CR1 寄存器中的 M 位定义的 8 位或 9 位字符长度。
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样。
- 是否使用小数波特率。
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值。

表 15-4: DIV_Faction 为 0 时的 USART 接收器容差

M 位	OVER8 位=0		OVER8 位=1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.75%	4.375%	2.50%	3.75%
1	3.41%	3.97%	2.27%	3.41%

表 15-5: DIV_Faction 不为 0 时的 USART 接收器容差

M 位	OVER8 位=0		OVER8 位=1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.33%	3.88%	2%	3%
1	3.03%	3.53%	1.82%	2.73%

15.3.6 多处理器通信

可以与 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。

可通过静音功能将未被寻址的器件置于静音模式下。在静音模式下：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- USART_CR1 寄存器中的 RWU 位置 1。RWU 可由硬件自动控制，或在特定条件下由软件写入。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静音模式：

- 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。

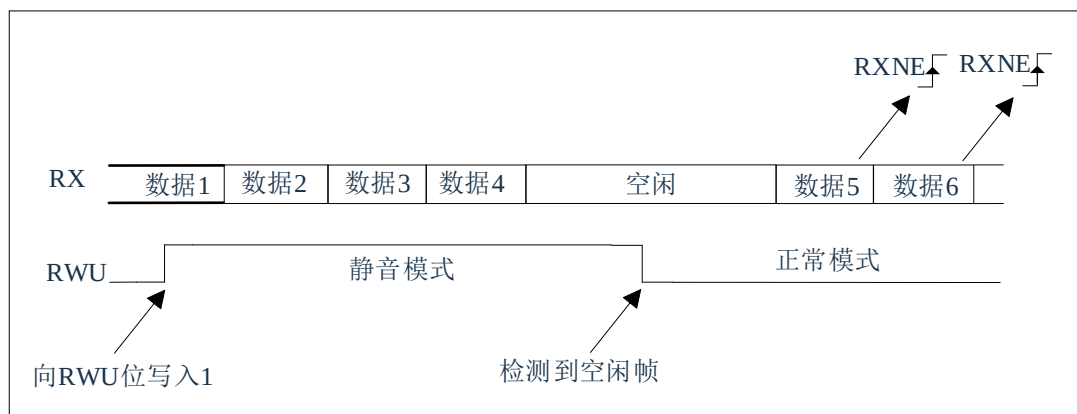
15.3.6.1 空闲总线检测(WAKE=0)

当向 RWU 位写入 1 时，USART 进入静音模式。

当检测到空闲帧时，它会被唤醒。此时 RWU 位会由硬件清零，但 USART_SR 寄存器中的 IDLE 位不会置 1。还可通过软件向 RWU 位写入 0。

图 15-8 中给出了使用空闲线路检测时静音模式行为的示例。

图 15-8 使用空闲线路检测时的静音模式



15.3.6.2 地址标记 (Address mark) 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 LSB 上。接收器会将此 4 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

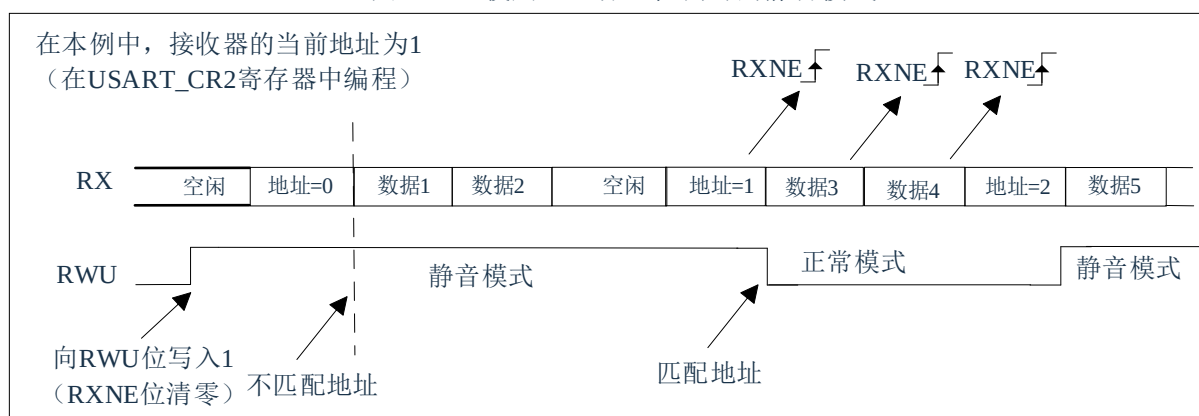
当接收到与其编程地址不匹配的地址字符时，USART 会进入静音模式。此时，RWU 位将由硬件置 1。由于当时 USART 已经进入了静音模式，所以 RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。

当接收到与编程地址匹配的地址字符时，它会退出静音模式。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。

当接收器的缓冲区不包含任何数据 (USART_SR 寄存器中 RXNE=0) 时，可向 RWU 位写入 0 或 1。否则会忽略写尝试。

图 15-9 中给出了使用地址标记检测时静音模式行为的示例。

图 15-9: 使用地址标记检测时的静音模式



15.3.7 奇偶校验控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，表 15-9 中列出了可能的 USART 帧格式。

表 15-9 帧格式

M 位	PCE 位	USART 帧 ⁽¹⁾
0	0	SB 8 位数据 STB
0	1	SB 7 位数据 PB STB
1	0	SB 9 位数据 STB
1	1	SB 8 位数据 PB STB

偶校验

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 7 个或 8 个 LSB 位组成，具体取决于 M 等于 0 还是 1）。

例如：数据=00110101；4 个位置 1 => 如果选择偶校验（USART_CR1 寄存器中的 PS 位 = 0），则校验位是 0。

奇校验

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 7 个或 8 个 LSB 位组成，具体取决于 M 等于 0 还是 1）。

例如：数据=00110101；4 个位置 1 => 如果选择奇校验（USART_CR1 寄存器中的 PS 位 = 1），则校验位是 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_SR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。PE 标志由软件序列清零（从状态寄存器中读取，然后对 USART_DR 数据寄存器执行读或写访问）。

注意：如果被地址标记唤醒：会使用数据的 MSB 位而非奇偶校验位来识别地址。此外，接收器不会对地址数据进行奇偶校验检查（奇偶校验出错时，PE 不置 1）。

发送时的奇偶校验生成

如果 USART_CR1 寄存器中的 PCE 位置 1，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=0)，则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数。

注意：用于管理发送过程的软件程序可以激活软件序列，进而将 PE 标志清零（从状态寄存器中读取，然后对数据寄存器执行读或写访问）。在半双工模式下工作时（具体取决于软件），这可能会导致 PE 标志意外清零。

15.3.8 LIN（局域互连网络）模式

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位，
- USART_CR3 寄存器中的 STOP、SCEN、HDSEL 和 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 第 16.3.2 节中介绍的相同步骤，同时还具有以下区别：

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

LIN 接收

断路检测电路在 USART 接口上实现。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（USART_CR1 寄存器中 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 10 个（USART_CR2 寄存器中 LBDL = 0 时）或 11 个（USART_CR2 寄存器中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_SR 寄存器中的 LBD 标志将会置 1。如果 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 10 或第 11 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 (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 (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。

图 15-10：LIN 模式下的断路检测（11 位断路长度—LBDL 位置 1）中显示了断路检测器状态机和断路标志的行为。

图 15-11：LIN 模式下的断路检测与帧错误检测中列出了断路帧的示例。

图 15-10 LIN 模式下的断路检测（11 位断路长度——LBDL 位置 1）

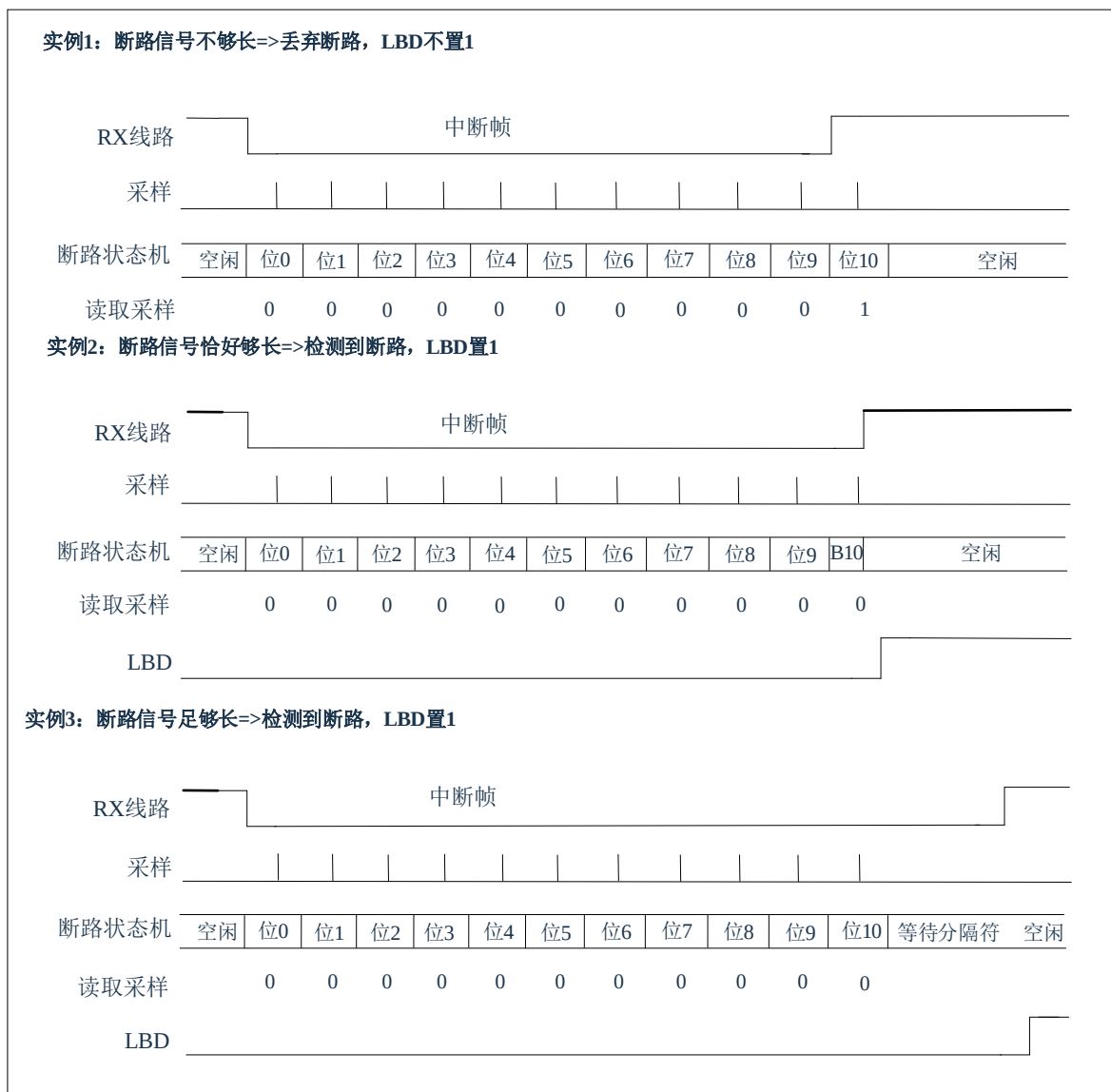
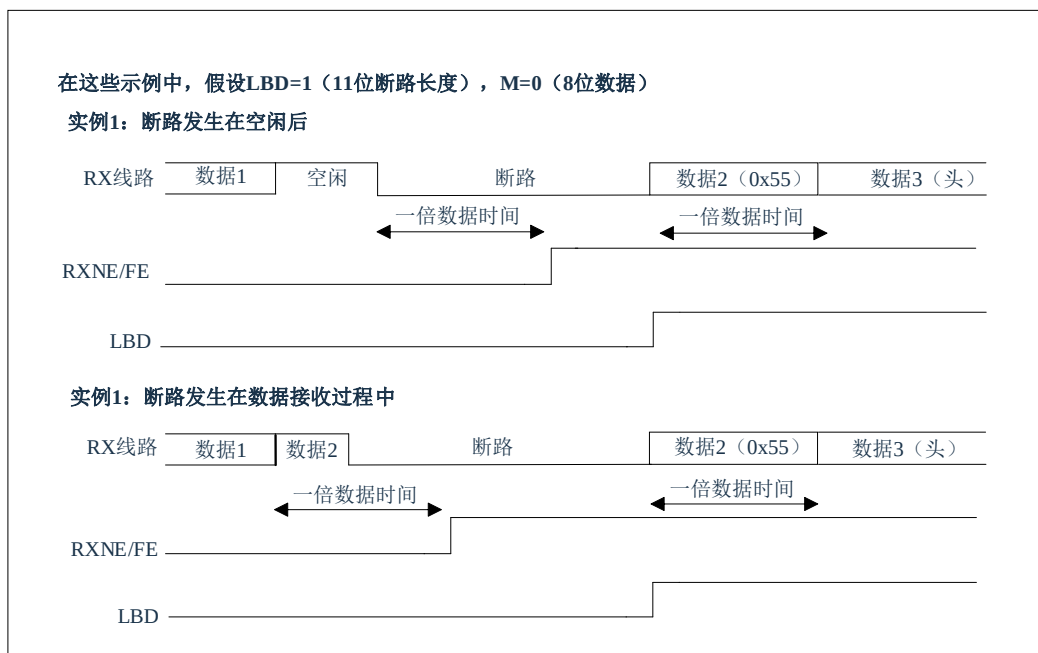


图 15-11 LIN 模式下的断路检测与帧错误检测



15.3.9 USART 同步模式

通过将 USART_CR2 寄存器中的 CLKEN 位写入 1 来选择同步模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

通过 USART，用户可以在主模式下控制双向同步串行通信。SCLK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，将会（也可能不会）生成时钟脉冲，这取决于 USART_CR2 寄存器中 LBCL 位的状态。通过 USART_CR2 寄存器中的 CPOL 位，用户可以选择时钟极性；通过 USART_CR2 寄存器中的 CPHA 位，用户可以选择外部时钟相位（参见图 15-12、图 15-13 和图 15-14）。

在空闲状态、报头模式和发送断路期间，外部 SCLK 时钟处于未激活状态。

USART 发送器在同步模式下的工作方式与异步模式下完全相同。但是由于 SCLK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在此模式下，USART 接收器的工作方式与异步模式下不同。如果 RE=1，则数据在 SCLK 上采样（上升或下降沿，取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保建立时间和保持时间（取决于波特率：1/16 位时间）符合要求。

注意：SCLK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器 (TE=1) 且正在发送数据时（对数据寄存器 USART_DR 已被写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。

当发送器和接收器 (TE=RE=0) 都被禁止时，必须选择 LBCL、CPOL 和 CPHA 位，以确保时钟脉冲正常工作。当使能发送器或接收器时，不得对这些位进行更改。

建议按照相同指令将 TE 和 RE 位置 1，以尽量缩短接收器的建立时间和保持时间。

USART 只支持主模式：它不能接收或发送与输入时钟相关的数据（SCLK 始终为输出）

图 15-12 USART 同步发送示例

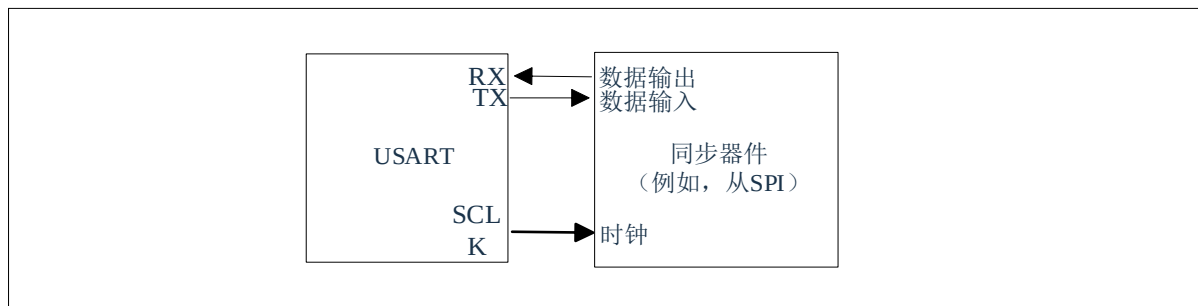


图 15-13: USART 数据时钟时序图 (M=0)

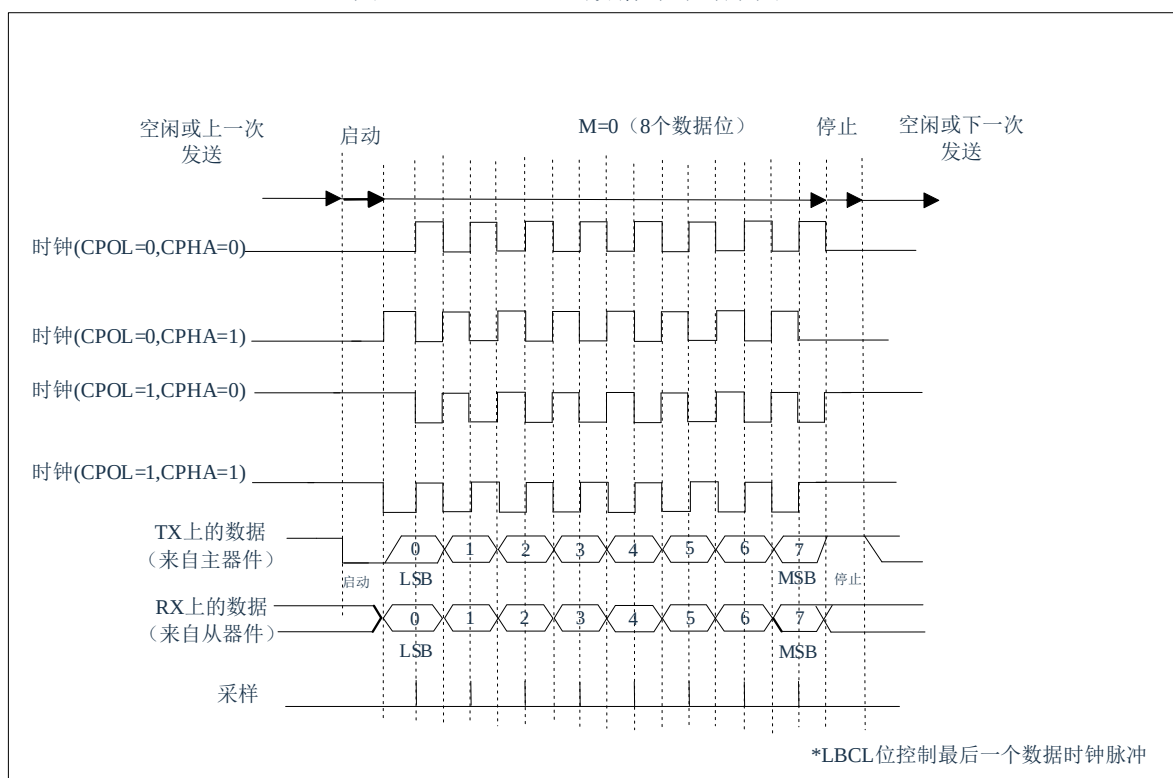


图 15-14: USART 数据时钟时序图 (M=1)

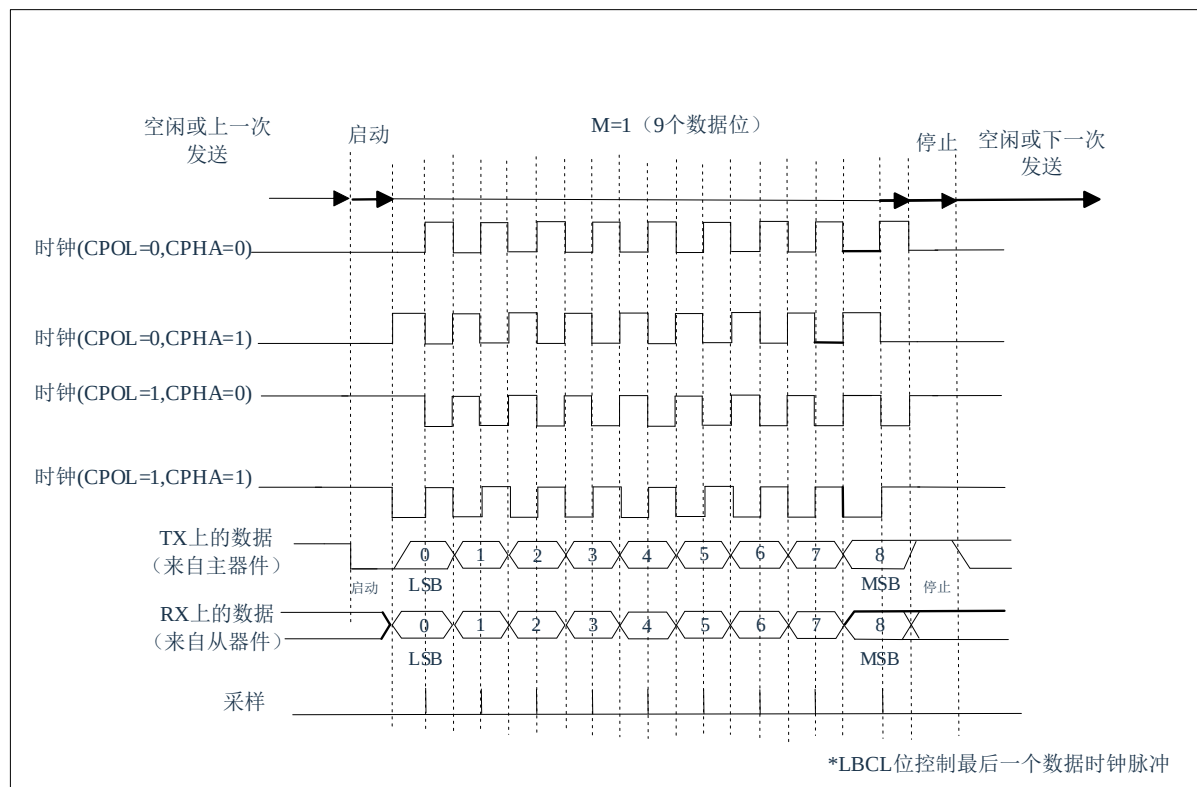
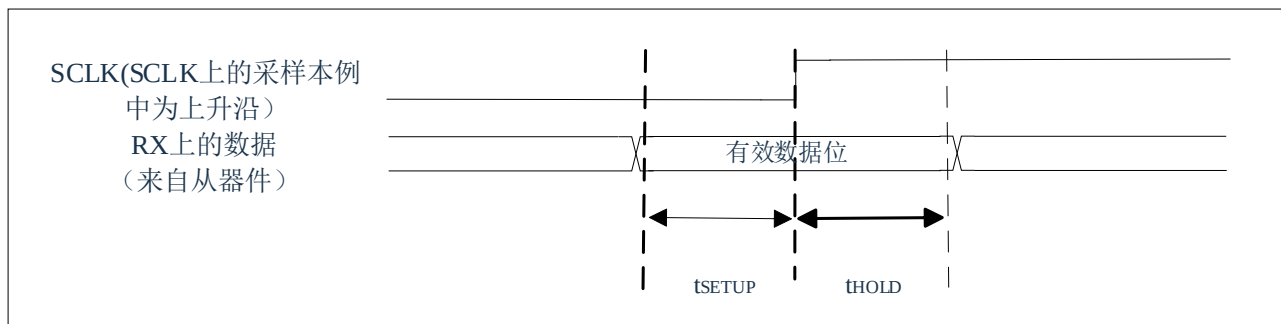


图 15-15 : RX 数据建立/保持时间



15.3.10 单线半双工通信

通过将 USART_CR3 寄存器中 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：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 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。使用控制位“HALF DUPLEX SEL”（USART_CR3 寄存器中的 HDSEL 位），可以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，必须对 I/O 进行配置，以便在未受 USART 驱动时，使 TX 成为浮空输入（或高电平开漏输出）除此之外，通信与正常 USART 模式下的通信相似。此线路上的冲突必须由软件进行管理（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。

15.3.11 IrDA SIR ENDEC 模块

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位；
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 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（参见图 15-18）。

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 SIR ENDEC 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（例如，USART 正在向 IrDA 编码器发送数据），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（例如，USART 正在接收来自 USART 的解码数据），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，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（参见图 15-19）。
- 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 us。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 IrDA 低功耗波特 寄存器 USART_GTPR 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，USART_CR2 寄存器中的 STOP 位必须配置为“1 个停止位”。

15.3.11.1 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 1.42 MHz。通常此值是 1.8432 MHz ($1.42 \text{ MHz} < \text{PSC} < 2.12 \text{ MHz}$)。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 $1/\text{PSC}$ 的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 寄存器中的 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 10 ms 的延迟（IrDA 是一个半双工协议）

图 15-18 IrDA SIR ENDEC—框图

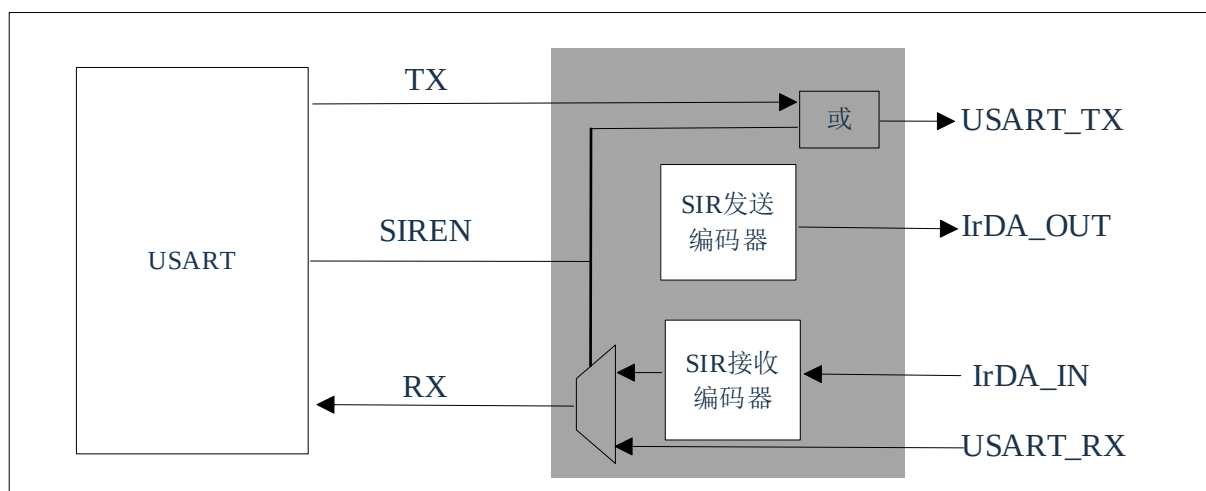
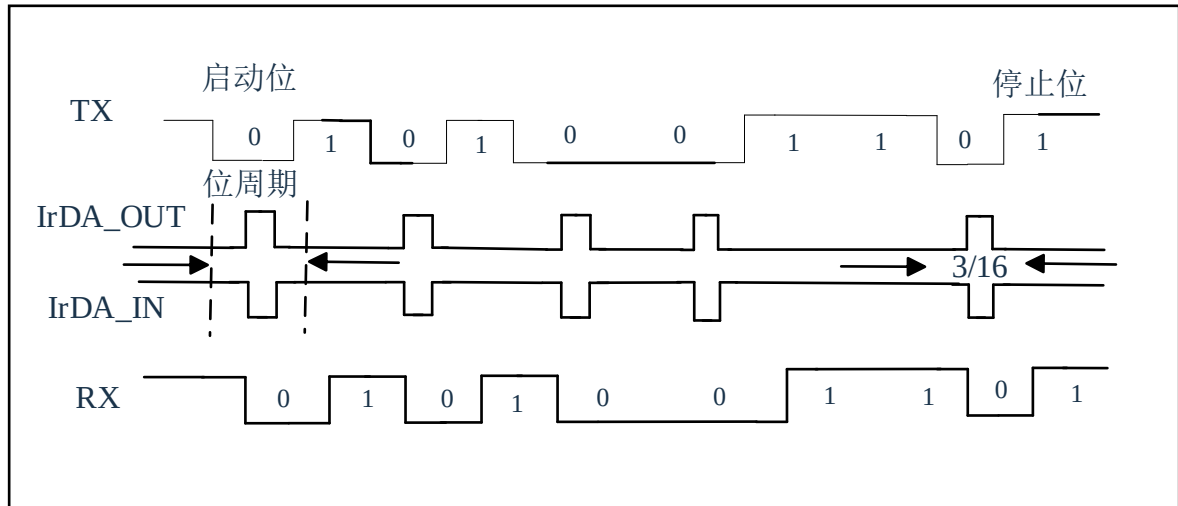


图 15-19 IrDA 数据调制（3/16）—正常模式



15.3.12 使用 DMA 进行连续通信

USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

注意：有关 DMA 控制器可用性的信息，请参见产品规范。如果 DMA 不适用于该产品，应按照第 16.3.2 节或 16.3.3 中的解释说明使用 USART。可以将 USART_SR 寄存器中的 TXE/RXNE 标志清零，从而实现连续通信。

15.3.12.1 使用 DMA 进行发送

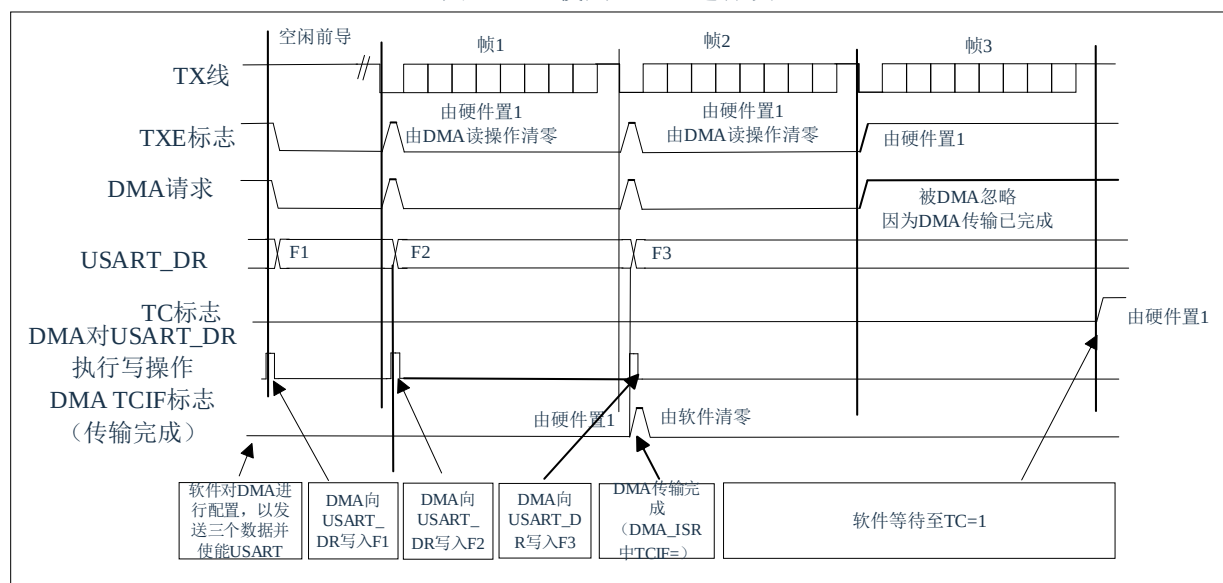
将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 位置 1 时，可将数据从 SRAM 区（通过 DMA 配置，参见 DMA 部分）加载到 USART_DR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_DR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE 事件后，数据都会从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE 事件后，数据都会从这个存储区域加载到 USART_DR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 向 SR 寄存器中的 TC 位写入 0，将其清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1。TC 标志在所有数据发送期间都必须保持清零状态，然后在最后一帧发送结束后由硬件置 1。

图 15-20 使用 DMA 进行发送



15.3.12.2 使用 DMA 进行接收

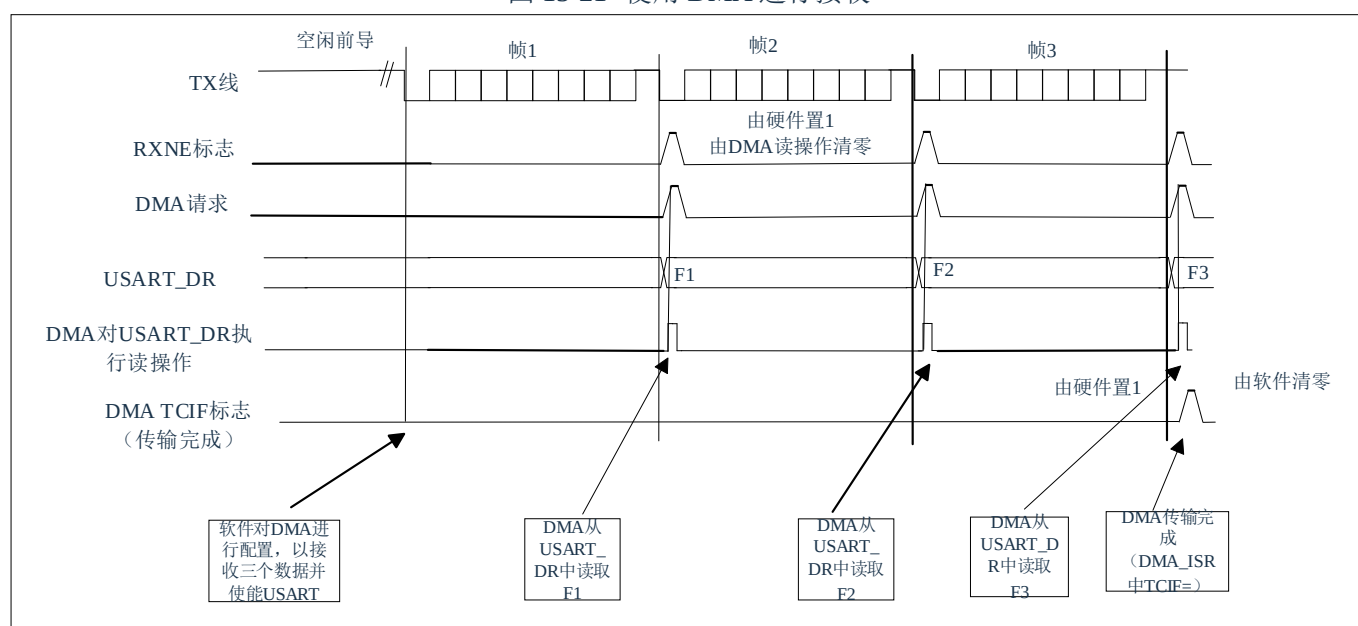
将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 USART_DR 寄存器加载到 SRAM 区域中（通过 DMA 配置，参见 DMA 规范）。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_DR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE 事件后，数据都会从此地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE 事件后，数据都会从 USART_DR 寄存器加载到此存储区。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。在中断子程序中，USART_CR3 寄存器中的 DMAR 位应由软件清零。

注意：如果 DMA 用于接收，则不要使能 RXNEIE 位。

图 15-21 使用 DMA 进行接收



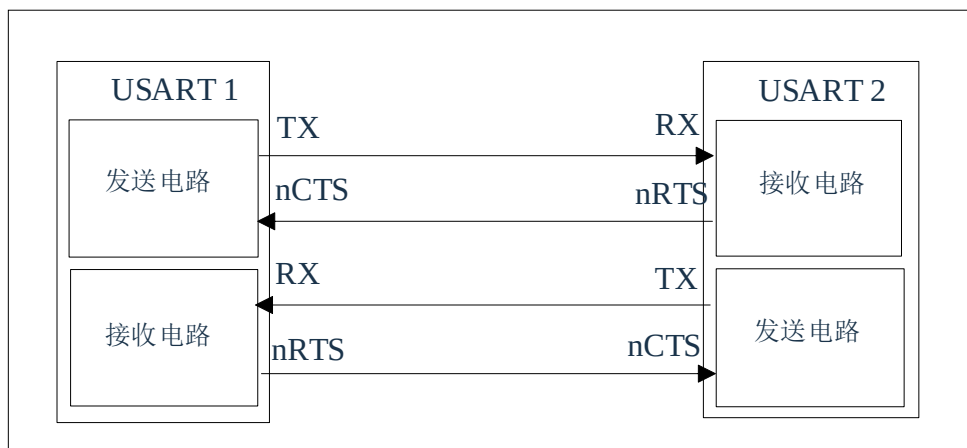
15.3.12.3 多缓冲区通信中的错误标志和中断生成

在多缓冲区通信中，如果事务中发生任何错误，都会在当前字节后放置错误标志。如果中断使能置 1，则会产生中断。在单字节接收过程中，与 RXNE 一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，则会因其中任何一个错误而在当前字节后产生中断。

15.3.13 硬件流控制

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。图 15-22 显示了在这种模式下如何连接 2 个器件

图 15-22: 2 个 USART 间的硬件流控制

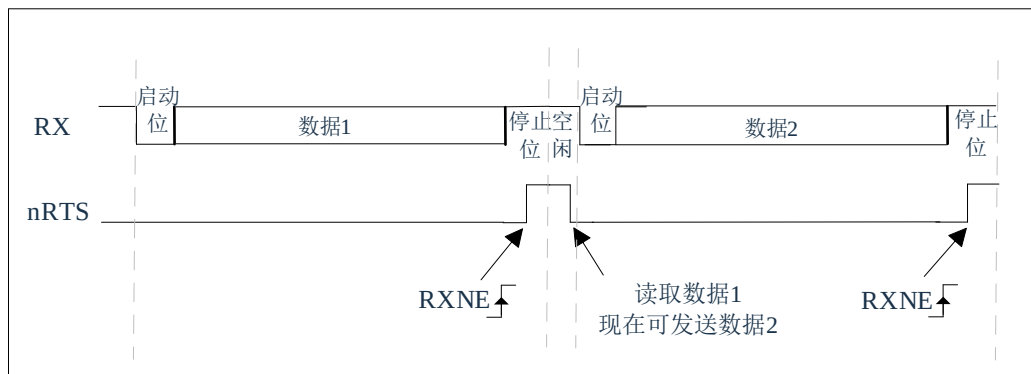


分别向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RTS 和 CTS 流控制

15.3.13.1 RTS 流控制

如果使能 RTS 流控制 (RTSE=1)，只要 USART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。图 15-23 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 15-23 RTS 流控制

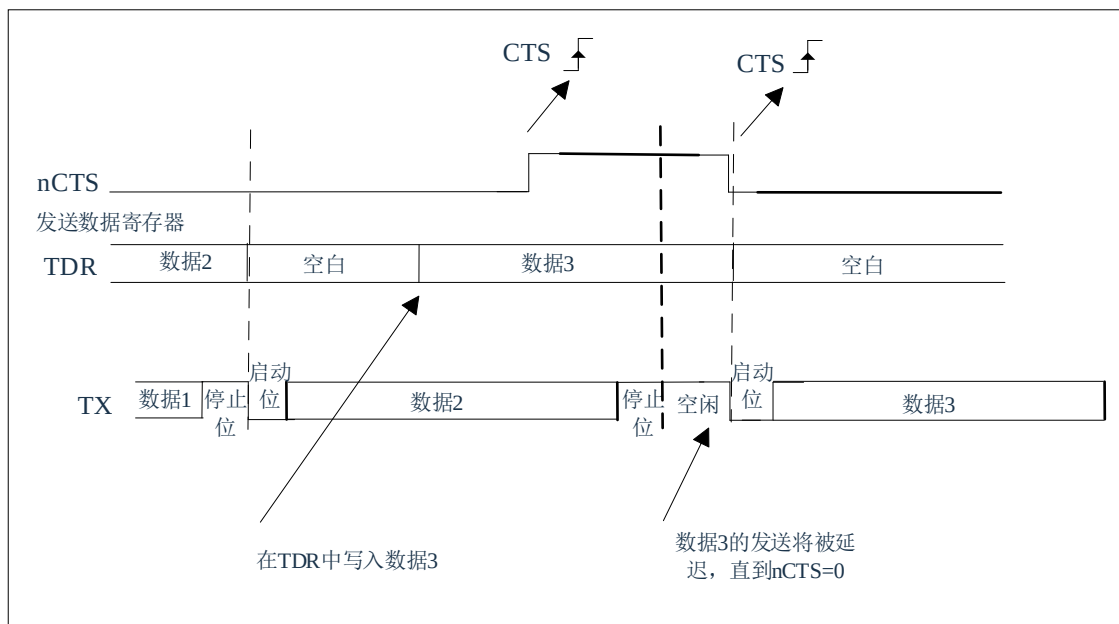


15.3.13.2 CTS 流控制

如果使能 CTS 流控制 (CTSE=1)，则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=1 时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。下图显示了在使能 CTS 流控制的情况下进行通信的示例。

图 15-24 CTS 流控制



注意：停止帧的特殊行为：使能 CTS 流后，发送器发送停止信号时将不检查 nCTS 输入状态。

15.4 USART 中断

表 15-10 USART 中断请求

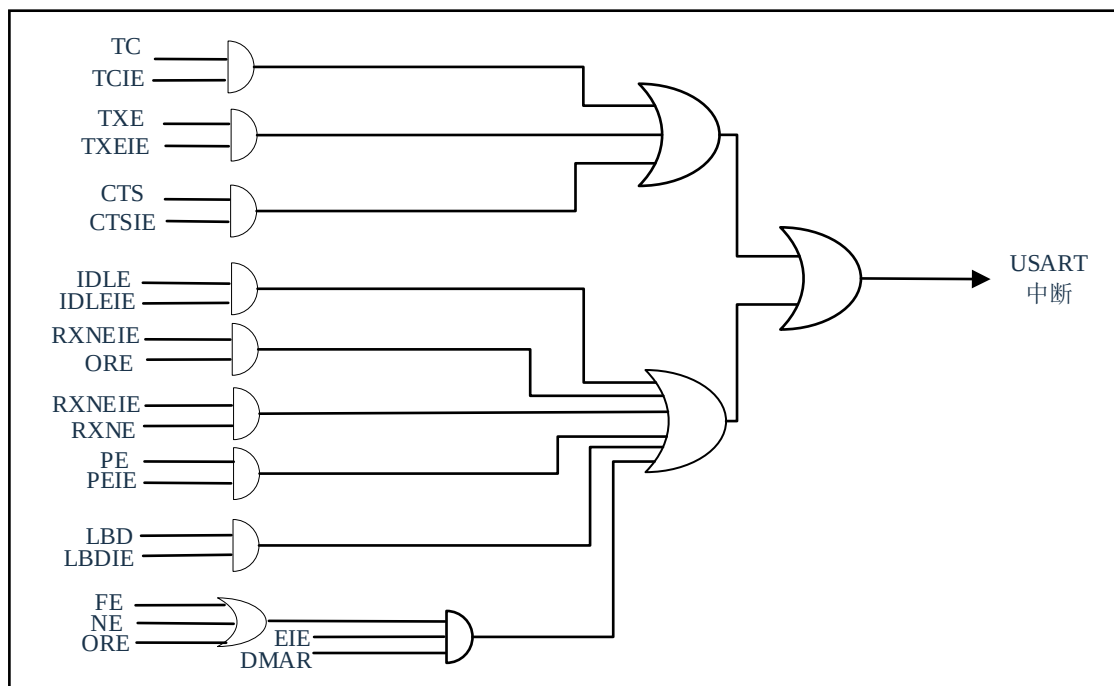
中断事件	事件标志	使能控制位
发送数据寄存器为空	TXE	TXEIE
CTS 标志	CTS	CTSIE
发送完成	TC	TCIE
准备好读取接收到的数据	RXNE	RXNEIE
检测到上溢错误	ORE	
检测到空闲线路	IDLE	IDLEIE
奇偶校验错误	PE	PEIE
断路标志	LBD	LBDIE
多缓冲区通信中的噪声标志、上溢错误和帧错误	NF 或 ORE 或 FE	EIE

USART 中断事件被连接到相同的中断向量（请参见图 15-25）。

- 发送期间：发送完成、清除以发送或发送数据寄存器为空中断。
- 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（仅限多缓冲区通信）和帧错误（仅限多缓冲区通信）

如果相应的使能控制位置 1，则这些事件会生成中断。

图 15-25 USART 中断映射图



15.5 USART 模式配置

表 15-11 USART 模式配置⁽¹⁾

USART 模式	USART
异步模式	X
硬件流控制	X
多缓冲区通信（DMA）	X
多处理器通信	X
同步	X
-	NA
半双工（单线模式）	X
IrDA	X
LIN	X

注 1： X=支持；NA=不适用。

15.6 寄存器映射

(USART 基地址 = 0x4007_1000) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
USART_SR	0x00	R/W	USART 状态寄存器	0xC0
USART_DR	0x04	R/W	USART 数据寄存器	0x00
USART_BRR	0x08	R/W	USART 波特率寄存器	0x00
USART_CR1	0x0C	R/W	USART 控制寄存器 1	0x00
USART_CR2	0x10	R/W	USART 控制寄存器 2	0x00
USART_CR3	0x14	R/W	USART 控制寄存器 3	0x00
USART_GTPR	0x18	R/W	USART 保护时间和预分频寄存器	0x00

15.6.1 状态寄存器 (USART_SR)

位	符号	描述	复位值
31:10	--	保留	--
9	CTS	CTS 标志 (CTS flag): 如果 CTSE 位置 1, 当 nCTS 输入变换时, 此位由硬件置 1。通过软件将该位清零 (通过向该位中写入 0)。如果 USART_CR3 寄存器中 CTSIE=1, 则会生成中断。 0: nCTS 状态线上未发生变化 1: nCTS 状态线上发生变化	0
8	LBD	LIN 断路检测标志 (LIN break detection flag): 检测到 LIN 断路时, 该位由硬件置 1。通过软件将该位清零 (通过向该位中写入 0)。如果 USART_CR2 寄存器中 LBDIE = 1, 则会生成中断。 0: 未检测到 LIN 断路 1: 检测到 LIN 断路 注意: 如果 LBDIE=1, 则当 LBD=1 时生成中断	0
7	TXE	发送数据寄存器为空 (Transmit data register empty): 当 TDR 寄存器的内容已传输到移位寄存器时, 该位由硬件置 1。如果 USART_CR1 寄存器中 TXEIE 位 = 1, 则会生成中断。通过对 USART_DR 寄存器执行写入操作将该位清零。 0: 数据未传输到移位寄存器 1: 数据传输到移位寄存器 注意: 单缓冲区发送期间使用该位。	1
6	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, 则该位由硬件置 1。如果 USART_CR1 寄存器中 TCIE = 1, 则会生成中断。该位由软件序列清零 (读取 USART_SR 寄存器, 然后写入 USART_DR 寄存器)。TC 位也可以通过向该位写入 '0' 来清零。建议仅在多缓冲区通信时使用此清零序列。 0: 传送未完成 1: 传送已完成	1
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 USART_DR 寄存器时, 该位由硬件置 1。如果 USART_CR1 寄存器中 RXNEIE = 1, 则会生成中断。通过对 USART_DR 寄存器执行读入操作将该位清零。RXNE 标志也可以通过向该位写入零来清零。建议仅在多缓冲区通信时使用此清零序列。 0: 未接收到数据	0

		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 USART_CR1 寄存器中 IDLEIE = 1, 则会生成中断。该位由软件序列清零 (读入 USART_SR 寄存器, 然后读入 USART_DR 寄存器)。 0: 未检测到空闲线路 1: 检测到空闲线路 注意: 直到 RXNE 位本身已置 1 时 (即, 当出现新的空闲线路时) IDLE 位才会被再次置 1。	0
3	ORE	上溢错误 (Overrun error): 在 RXNE = 1 的情况下, 当移位寄存器中当前正在接收的字准备好传输到 RDR 寄存器时, 该位由硬件置 1。如果 USART_CR1 寄存器中 RXNEIE = 1, 则会生成中断。该位由软件序列清零 (读入 USART_SR 寄存器, 然后读入 USART_DR 寄存器)。 0: 无上溢错误 1: 检测到上溢错误 注意: 当该位置 1 时, RDR 寄存器的内容不会丢失, 但移位寄存器会被覆盖。如果 EIE 位置 1, 则在多缓冲区通信时会对 ORE 标志生成一个中断。	0
2	NF	检测到噪声标志 (Noise detected flag): 当在接收的帧上检测到噪声时, 该位由硬件置 1。该位由软件序列清零 (读入 USART_SR 寄存器, 然后读入 USART_DR 寄存器)。 0: 未检测到噪声 1: 检测到噪声 注意: 如果 EIE 位置 1, 则在多缓冲区通信时, 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 位因 NF 标志而生成的时间相同。 注意: 当线路无噪声时, 可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NF 标志。	0
			0
1	FE	帧错误 (Framing error) 当检测到去同步化、过度的噪声或中断字符时, 该位由硬件置 1。该位由软件序列清零 (读入 USART_SR 寄存器, 然后读入 USART_DR 寄存器)。 0: 未检测到帧错误 1: 检测到帧错误或中断字符 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。如果当前正在传输的字同时导致帧错误和上溢错误, 则会传输该字, 且仅有 ORE 位被置 1。 如果 EIE 位置 1, 则在多缓冲区通信时会对 FE 标志生成一个中断。	0
			0
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。该位由软件序列清零 (读取状态寄存器, 然后对 USART_DR 数据寄存器执行读或写访问)。将 PE 位清零前软件必须等待 RXNE 标志被置 1。 如果 USART_CR1 寄存器中 PEIE = 1, 则会生成中断。 0: 无奇偶校验错误 1: 奇偶校验错误	0

15.6.2 数据寄存器 (USART_DR)

位	符号	描述	复位值
31:9	--	保留	--
8:0	DR[8:0]	数据值: 包含接收到数据字符或已发送的数据字符, 具体取决于所执行的操作是“读取”操作还是“写入”操作。 因为数据寄存器包含两个寄存器, 一个用于发送 (TDR), 一个用于接收 (RDR), 因此它具有 双重功能 (读和写)。 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。 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 (USART_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。 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。	0x0

15.6.3 波特率寄存器 (USART_BRR)

注意: 如果 TE 或 RE 位分别被禁止, 则波特计数器会停止计数。

位	符号	描述	复位值
31:16	--	保留	--
15:4	DIV_Mantissa[11:0]	USARTDIV 的尾数: 这 12 个位用于定义 USART 除数 (USARTDIV) 的尾数	0x0
3:0	DIV_Fraction[3:0]	USARTDIV 的小数: 这 4 个位用于定义 USART 除数 (USARTDIV) 的小数。 当 OVER8 = 1 时, 不考虑 DIV_Fraction3 位, 且必须将该位保持清零。	0x0

15.6.4 控制寄存器 1 (USART_CR1)

位	符号	描述	复位值
31:16	--	保留	--
15	OVER8	过采样模式 (Oversampling mode): 0: 16 倍过采样 1: 8 倍过采样(禁止使用)	0
14	--	保留	--
13	UE	USART 使能 (USART enable): 该位清零后, USART 预分频器和输出将停止, 并会结束当前字节传输以降低功耗。此位由软件置 1 和清零。 0: 禁止 USART 预分频器和输出 1: 使能 USART	0
12	M	字长 (Word length): 该位决定了字长。该位由软件置 1 或清零。 0: 1 起始位, 8 数据位, n 停止位 1: 1 起始位, 9 数据位, n 停止位 注意: 在数据传输 (发送和接收) 期间不得更改 M 位	0
11	WAKE	唤醒方法 (Wakeup method): 该位决定了 USART 唤醒方法, 该位由软件置 1 或清零。 0: 空闲线路 1: 地址标记	0
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)。 0: 禁止奇偶校验控制 1: 使能奇偶校验控制	0
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	0
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 USART_SR 寄存器中 PE=1 时, 生成 USART 中断	0
7	TXEIE	TXE 中断使能 (TXE interrupt enable):(此位由软件置 1 和清零) 0: 禁止中断 1: 当 USART_SR 寄存器中 TXE=1 时, 生成 USART 中断。	0
6	TCIE	传送完成中断使能 (Transmission complete interrupt enable): 此位由软件置 1 和清零。 0: 禁止中断 1: 当 USART_SR 寄存器中 TC=1 时, 生成 USART 中断	0
5	RXNEIE	RXNE 中断使能 (RXNE interrupt enable): 此位由软件置 1 和清零。 0: 禁止中断 1: 当 USART_SR 寄存器中 ORE=1 或 RXNE=1 时, 生成 USART 中断	0

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: 当 USART_SR 寄存器中 IDLE=1 时, 生成 USART 中断	0
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 注意: 1: 传送期间 TE 位上的“0”脉冲 (“0”后紧跟的是“1”) 会在当前字的后面发送一个报头 (空闲线路)。 2: 当 TE 置 1 时, 在发送开始前存在 1 位的时间延迟。	0
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	0
1	RWU	接收器唤醒 (Receiver wakeup): 该位决定 USART 是否处于静音模式。该位由软件置 1 和清零, 并可在识别出唤醒序列时由硬件清零。 0: 接收器处于活动模式 1: 接收器处于静音模式 注意: 1: 选择静音模式前 (通过将 RWU 位置 1), USART 必须首先接收一个数据字节, 否则当由空闲线路检测到唤醒时, 它无法于静音模式下正常工作。 2: 在地址标记检测唤醒配置 (WAKE 位 = 1) 中, RXNE 位置 1 时, RWU 位不能由软件进行修改。	0
			0
0	SBK	发送断路 (Send break): 该位用于发送断路字符。该位可由软件置 1 和清零。该位应由软件置 1, 并在断路停止位期间由硬件重置。 0: 不发送断路字符 1: 将发送断路字符	0

15.6.5 控制寄存器 2 (USART_CR2)

位	符号	描述	复位值
31:15	--	保留	--
14	LINEN	LIN 模式使能 (LIN mode enable): 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 USART_CR1 寄存器中的 SBK 位发送 LIN 同步断路 (13 个低位), 并可检测 LIN 同步断路。	0
13:12	STOP	停止位 (STOP bit): 这些位用于编程停止位。 00: 1 个停止位 01: 禁止设置 10: 2 个停止位 11: 禁止设置	0
11	CLKEN	时钟使能 (Clock enable): 该位允许用户使能 SCLK 引脚。 0: 禁止 SCLK 引脚 1: 使能 SCLK 引脚	0
10	CPOL	时钟极性 (Clock polarity): 该位允许用户在同步模式下选择 SCLK 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 SCLK 引脚为低电平 1: 空闲时 SCLK 引脚为高电平	0
9	CPHA	该位允许用户在同步模式下选择 SCLK 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/数据关系 (请参见图 258 至 259) 0: 在时钟第一个变化沿捕获数据 1: 在时钟第二个变化沿捕获数据	0
8	LBCL	最后一个位时钟脉冲 (Last bit clock pulse) 该位允许用户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 SCLK 引脚上输出。 0: 最后一个数据位的时钟脉冲不在 SCLK 引脚上输出 1: 最后一个数据位的时钟脉冲在 SCLK 引脚上输出 注意: 1: 最后一位为发送的第 8 或第 9 个数据位, 具体取决于 USART_CR1 寄存器中 M 位所选择的 8 位或 9 位格式。 2: 该位不适用于 UART4 和 UART5。	0
7	--	保留	--
6	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 USART_SR 寄存器中 LBD = 1 时, 生成中断	0
5	LBDL	lin 断路检测长度 (lin break detection length): 该位用于选择 11 位断路检测或 10 位断路检测。 0: 10 位断路检测 1: 11 位断路检测	0
4	--	保留	--
3:0	ADD[3:0]	USART 节点的地址: 该位域用于指定 USART 节点的地址。将在多处理器通信时于静音模式下使用该位域, 以通过地址标记检测进行唤醒。	0x0

注意: 使能发送器时不应对这 3 个位 (CPOL、CPHA、LBCL) 进行写操作。

15.6.6 控制寄存器 3 (USART_CR3)

位	符号	描述	复位值
31:12	--	保留	--
11	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(禁止使用)	0
10	CTSIE	CTS 中断使能 (CTS interrupt enable): 0: 禁止中断 1: 当 USART_SR 寄存器中 CTS = 1 时, 生成中断	0
9	CTSE	CTS 使能 (CTS enable): 0: 禁止 CTS 硬件流控制 1: 使能 CTS 模式, 仅当 nCTS 输入有效 (连接到 0) 时才发送数据。如果在发送数据时使 nCTS 输入无效, 会在停止之前完成发送。如果使 nCTS 有效时数据已写入数据寄存器, 则将延迟发送, 直到 nCTS 有效。	0
8	RTSE	RTS 使能 (RTS enable): 0: 禁止 RTS 硬件流控制 1: 使能 RTS 中断, 仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效 (连接到 0)。	0
7	DMAT	DMA 使能发送器 (DMA enable transmitter): 该位由软件置 1/ 复位。 0: 针对发送禁止 DMA 模式 1: 针对发送使能 DMA 模式	0
6	DMAR	DMA 使能接收器 (DMA enable receiver): 该位由软件置 1/ 复位。 0: 针对接收禁止 DMA 模式 1: 针对接收使能 DMA 模式	0
5:4		保留	0
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	0
2	IRLP	IrDA 低功耗 (IrDA low-power): 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	0

1	IREN	IrDA 模式使能 (IrDA mode enable): 0: 禁止 IrDA 1: 使能 IrDA	0
0	EIE	错误中断使能 (Error interrupt enable): 对于多缓冲区通信 (USART_CR3 寄存器中 DMAR = 1)，如果发生帧错误、上溢错误或出现噪声标志 (USART_SR 寄存器中 FE = 1 或 ORE = 1 或 NF = 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 USART_CR3 寄存器中的 DMAR = 1 并且 USART_SR 寄存器中的 FE = 1 或 ORE = 1 或 NF = 1 时，将生成中断	0

15.6.7 保护时间和预分频器寄存器 (USART_GTPR)

位	符号	描述	复位值
31:8	--	保留	--
7:0	PSC[7:0]	预分频器值： — 在 IrDA 低功耗模式下： PSC[7:0] = IrDA 低功耗波特率 用于编程预分频器，进行系统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 源时钟 1 分频 00000010: 源时钟 2 分频 ... — 在正常 IrDA 模式下： PSC 必须设置为 00000001。	0x0

第16章 增强型 DMA

16.1 DMA 的功能

DMA 是不使用 CPU 而在存储器之间进行数据传送的功能。通过外围功能中断启动 DMA 进行数据传送。当 DMA 和 CPU 同时访问 FLASH，SRAM 或外围模块中的同一单元时，其总线使用权高于 CPU。当 DMA 和 CPU 分别访问 FLASH，SRAM 外围模块中的不同单元时，两者互不干扰，可以并行执行。

DMA 的规格如表 16-1 所示。

表 16-1 DMA 的规格(1/2)

项目		规格
启动源		同时最大 3 个启动源
可分配的控制数据		3 组
可传送的地址空间	地址空间	全地址范围空间
	源	全地址范围空间可选
	目标	全地址范围空间可选
最大传送次数	正常模式	65535 次
	重复模式	65535 次
最大传送块大小	正常模式 (8 位传送)	65535 字节
	正常模式 (16 位传送)	131070 字节
	正常模式 (32 位传送)	262140 字节
	重复模式	65535 字节
传送单位		8 位/16 位/32 位
传送模式	正常模式	在进行 DMACTj 寄存器从“1”变为“0”的传送后结束。
	重复模式	在 DMACTj 寄存器从“1”变为“0”的传送结束后，对重复区的地址进行初始化，在将 DMRLDj 寄存器的值重新加载到 DMACTj 寄存器后继续传送。
地址控制	正常模式	固定或者递增
	重复模式	固定或者递增非重复区的地址。
启动源的优先级		参照“表 16-4 DMA 启动源和向量地址。

表 16-1 DMA 的规格(2/2)

项目		规格
中断请求	正常模式	在进行 DMACTj 寄存器从“1”变为“0”的数据传送时，向 CPU 请求启动源的中断，并进行中断处理。
	重复模式	在 DMACRj 寄存器的 RPTINT 位为“1”允许产生中断，并且进行 DMACTj 寄存器从“1”变为“0”的数据传送时，向 CPU 请求启动源的中断，并进行中断处理。
传送开始		如果将 DMAENi 寄存器的 DMAENi0~DMAENi1，DMAENi3 位置“1”（允许启动），就在每次发生 DMA 启动源时开始传送数据。
传送停止	正常模式	• 将 DMAENi0~DMAENi1，DMAENi3 位置“0”（禁止启动）。 • 当 DMACTj 寄存器从“1”变为“0”的数据传送结束时
	重复模式	• 将 DMAENi0~DMAENi1，DMAENi3 位置“0”（禁止启动）。 • 当 RPTINT 位为“1”（允许产生中断）并且 DMACTj 寄存器从“1”变为“0”的数据传送结束时

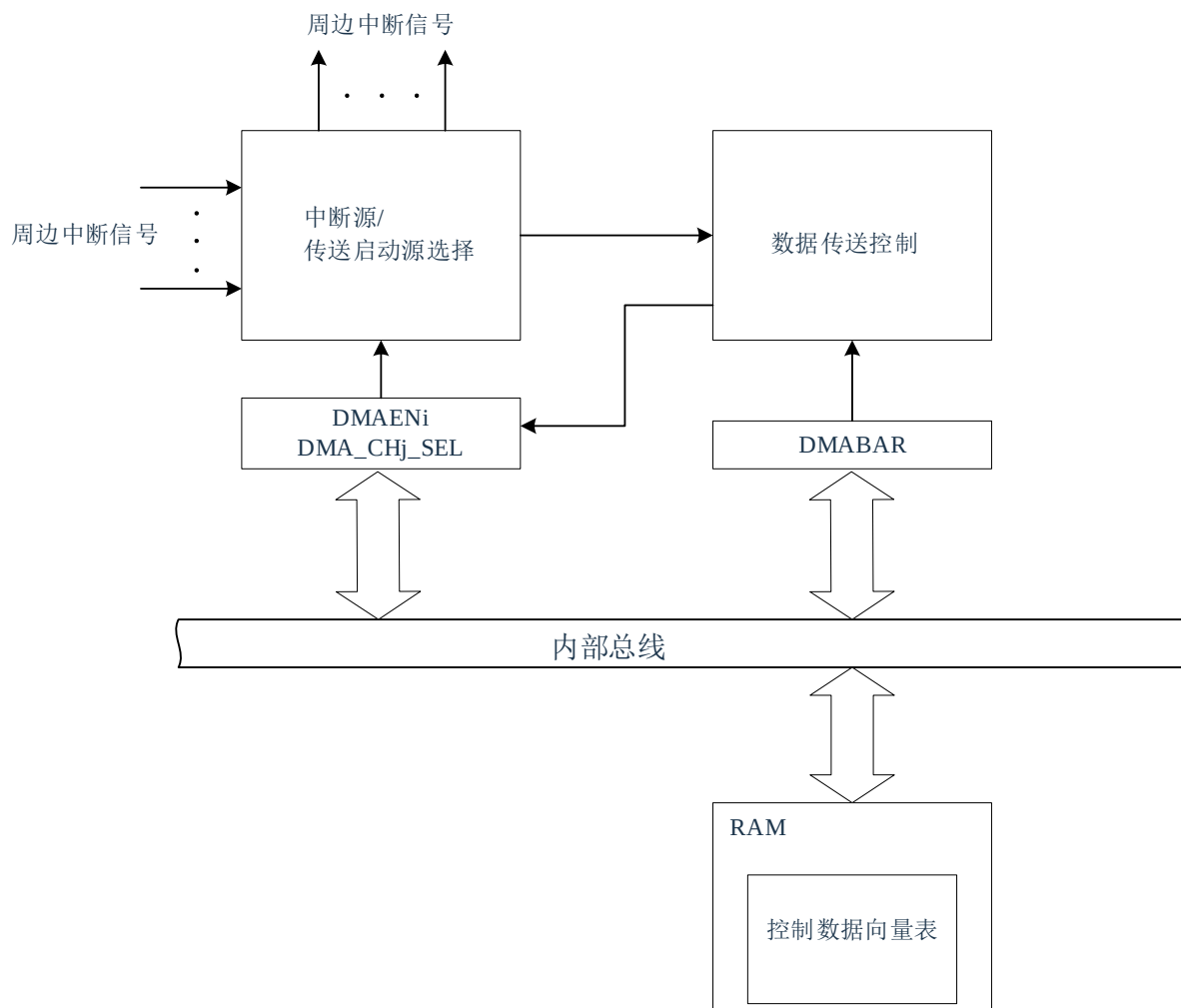
注 1：在深度睡眠模式，因为闪存停止运行，所以不能作为 DMA 传送源。

注 2：i=0，j=0~1，3。

16.2 DMA 的结构

DMA 的框图如图 16-1 所示

图 16-1 DMA 的框图



16.3 寄存器映射

(以下寄存器基地址 = 0x40020810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER10	0x000	R/W	外围允许寄存器 10	0x00

(以下寄存器基地址 = 0x40005000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
DMAEN0	0x000	R/W	DMA 启动允许寄存器 0	0x00
DMABAR	0x008	R/W	DMA 基址寄存器	0x00
DMASET0	0x018	WO	DMAEN0 置位寄存器	0x00
DMACLR0	0x020	WO	DMAEN0 复位寄存器	0x00

(以下寄存器基地址 = 0x40020D00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
DMA_CH0_SEL	0x000	R/W	DMA 通道 0 触发源选择寄存器	0x3F
DMA_CH1_SEL	0x001	R/W	DMA 通道 1 触发源选择寄存器	0x3F
DMA_CH3_SEL	0x003	R/W	DMA 通道 3 触发源选择寄存器	0x3F

DMA 的控制数据如表 16-2 所示。

DMA 的控制数据分配在 RAM 的 DMA 控制数据区。通过 DMABAR 寄存器设置 DMA 控制数据区和包含 DMA 向量表区（保存控制数据的起始地址）的 80 字节区域。

表 16-2 DMA 的控制数据

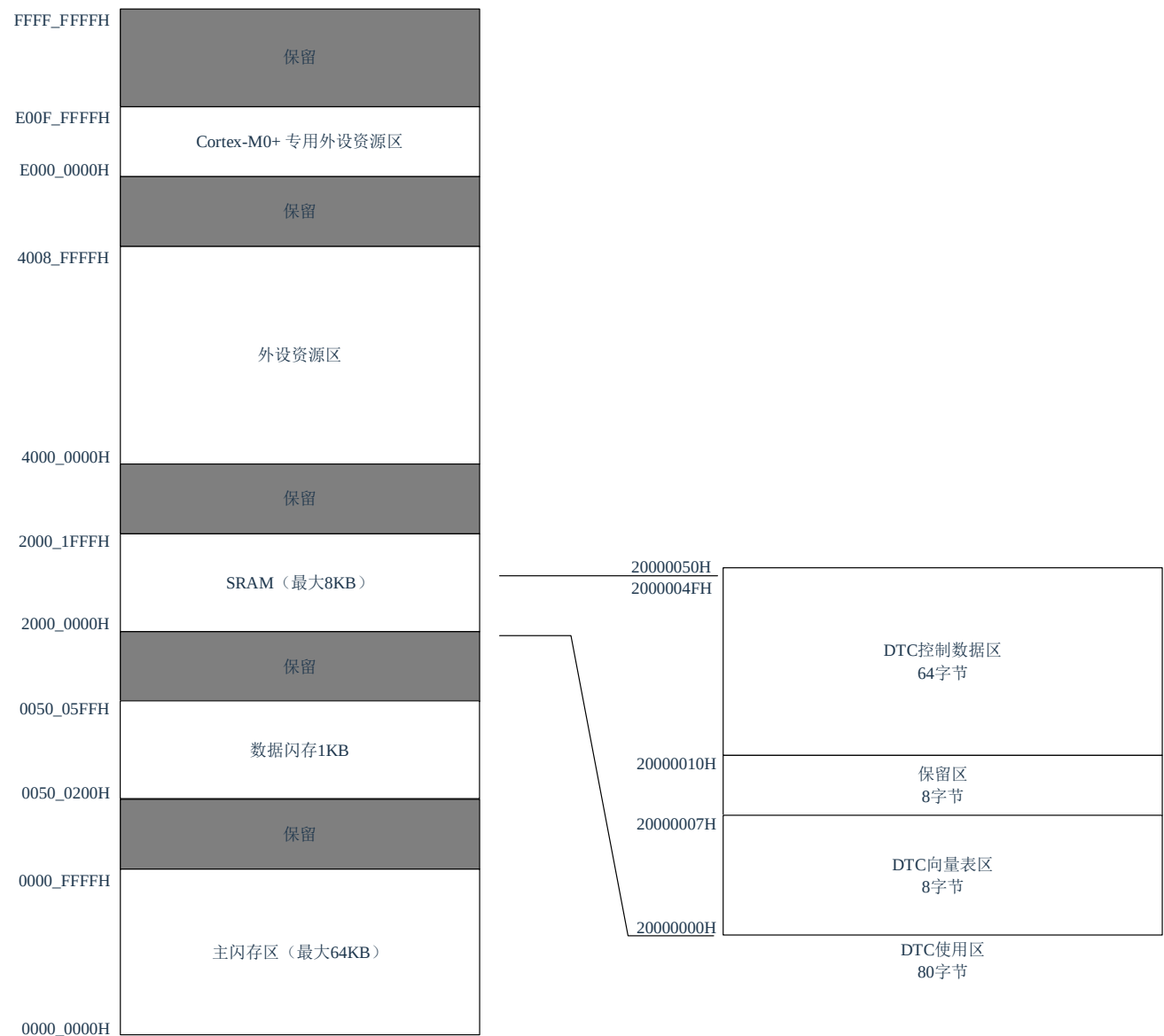
寄存器名	符号
DMA 控制寄存器 j	DMACRj
DMA 块大小寄存器 j	DMBSLj
DMA 传送次数寄存器 j	DMACTj
DMA 传送次数重加载寄存器 j	DMRLDj
DMA 源地址寄存器 j	DMSARj
DMA 目标地址寄存器 j	DMDARj

16.4 控制 DMA 的寄存器

16.4.1 DMA 控制数据区和 DMA 向量表区的分配

通过 DMABAR 寄存器将分配 DMA 的控制数据和向量表的 80 字节区域设置到 RAM 区。
DMABAR 寄存器的设置值为“20000000H”时的存储器映像例子如图 16-2 所示。
DMA 控制数据区的 80 字节中 DMA 不使用的空间能用作 RAM。

图 16-2 ： DMABAR 寄存器的设置值为“20000000H”时的存储器映像例子



16.4.2 控制数据的分配

从起始地址开始，按照 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj（j=0~1,3）寄存器的顺序分配控制数据。

起始地址由 DMABAR 寄存器设置，低 8 位由各启动源分配的向量表分别设置。

每个通道各自的触发源由 DMA_CHj_SEL（j=0~1,3）寄存器设置。

控制数据的分配如图 16-3 所示。

注 1：必须在对应的 DMAENi（i=0）的 DMAENi0~DMAENi2，DMAENi3 位为“0”（禁止启动）时更改 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj 寄存器的数据。

注 2：不能通过 DMA 传送进行 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj 和 DMDARj 的存取。

图 16-3 控制数据的分配(DMABAR 设置为 20000000H)

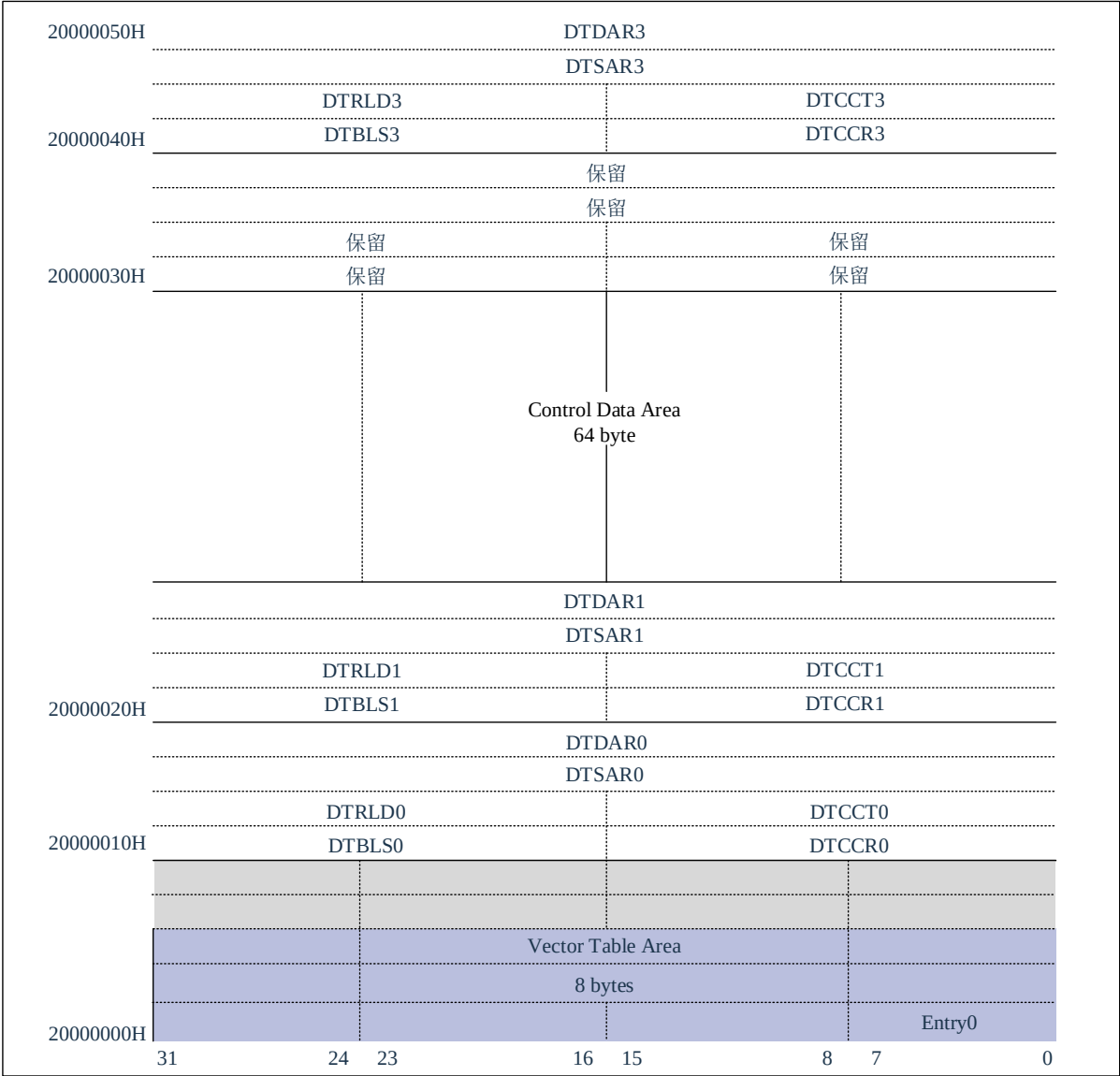


表 16-3 控制数据的起始地址

j	地址
3	baseaddr+40H
-	-
1	baseaddr+20H
0	baseaddr+10H

注：baseaddr：DMABAR 寄存器的设置值。

16.4.3 向量表

DMA 一旦启动，就通过从各启动源分配的向量表读取的数据来决定控制数据，读被分配在 DMA 控制数据区的控制数据。

DMA 启动源和向量地址如表 16-4 所示。各启动源的向量表有 1 字节，保存“00H”~“02H”，“03H”的数据，从 3 组的控制数据中选择 1 组数据。向量地址的高 24 位由 DMABAR 寄存器设置，低 8 位被分配了对应启动源的“00H”~“02H”，“03H”。

注：必须在对应的 DMAENi (i=0) 寄存器的 DMAENi0 位为“0”（禁止启动）时更改设置在向量表中的 DMA 控制数据区的起始地址。

图 16-4：控制数据的起始地址和向量表 DMABAR 寄存器的设置值为“20000000H”的情况（例）

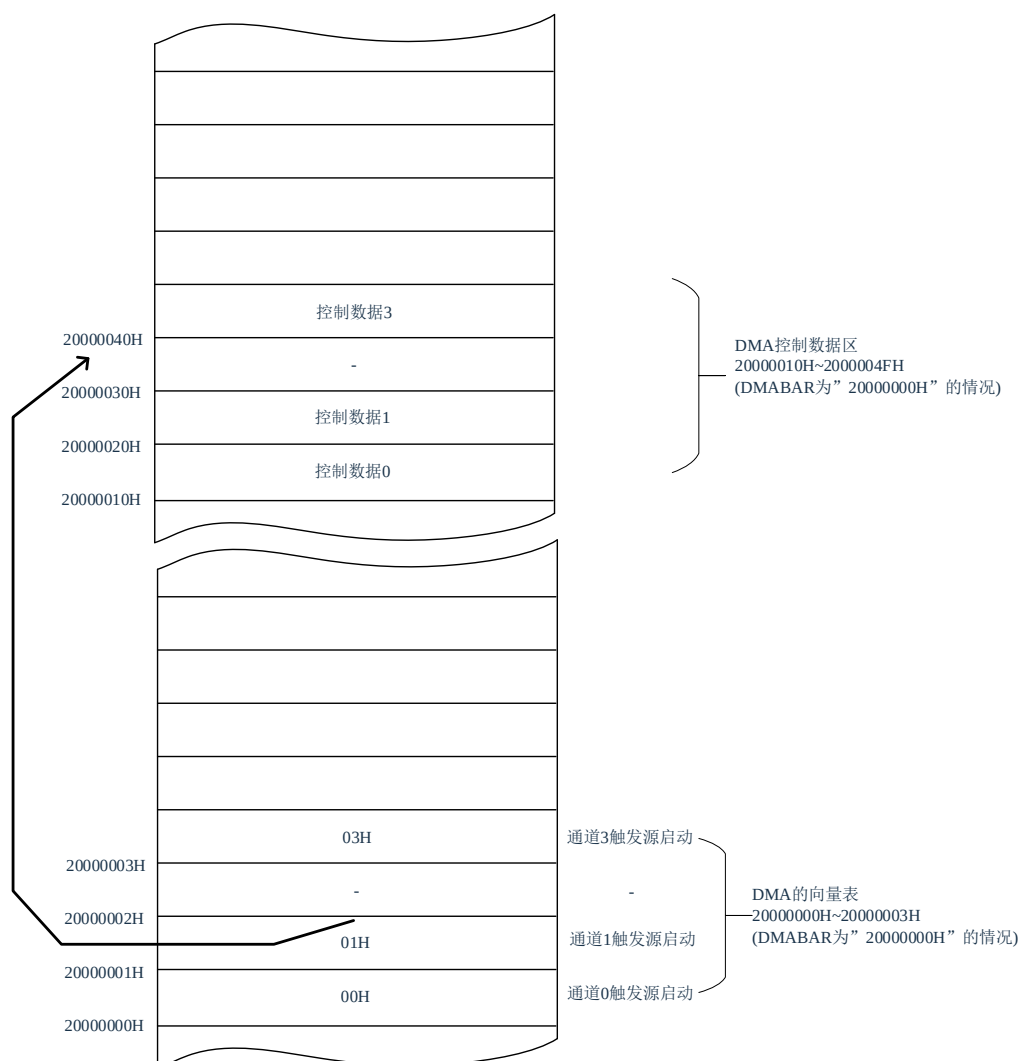


表 16-4 DMA 启动源和向量地址

DMA 启动源（中断请求发生源）	源号	向量地址	优先级
通道 0 触发源启动	0	DMABAR 寄存器的设置地址+00H	高 ↑ ↓ 低
通道 1 触发源启动	1	DMABAR 寄存器的设置地址+01H	
-	-	-	
通道 3 触发源启动	3	DMABAR 寄存器的设置地址+03H	

16.4.4 外围允许寄存器 10（PER10）

PER10 寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。要使用 DMA 时，必须将 bit3（DMAEN）置“1”。

通过 8 位存储器操作指令设置 PER10 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

外围允许寄存器 10（PER10）

位	符号	描述	复位值
7:4	-	保留	0
3	DMAEN	提供 DMA 模块输入时钟的控制 0: 停止提供输入时钟，DMA 不能运行 1: 提供输入时钟，DMA 能运行	0
2	CRCEN	提供 CRC 模块输入时钟的控制 0: 停止提供输入时钟，CRC 不能运行 1: 提供输入时钟，CRC 能运行	0
1:0	-	保留	0

16.4.5 DMA 通道触发源选择寄存器 j (DMA_CHj_SEL) (j=0~1,3)

位	符号	描述	复位值
7:6	-	保留	0x0
5:0	TRGSEL	DMA 通道 j 选择触发源(j=0~3): 0x00: ADC 0x01: SPI1_TX 0x02: SPI1_RX 0x03: SAU 模块 UART 发送、SPI/IIC 发送和接收结束 0x04: SAU 模块 UART 的接收结束 0x05: USART_TX 0x06: USART_RX 0x07: 保留 0x08: 保留 0x09: I2C_TX 0x0A: I2C_RX 0x0B: TIM1_CH1 0x0C: TIM1_CH2 0x0D: TIM1_CH3 0x0E: TIM1_CH4 0x0F: TIM1_COM 0x10: TIM1_UP 0x11: TIM1_TRIG 0x12: TIM3_CH1 0x13: TIM3_CH3 0x14: TIM3_CH4 0x15: TIM3_TRG 0x16: TIM3_UP 0x17: 保留 0x18: TIM16_CH1 0x19: TIM16_UP 0x1A: TIM17_CH1 0x1B: TIM17_UP 0x1C: TIM3_CH2 其他值: 禁止设置	0x3F

16.4.6 DMA 控制寄存器 j (DMACRj) (j=0~1,3)

DMACRj 寄存器控制 DMA 的运行模式。

地址：参照“16.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:8	-	保留	不定值
7:6	SZ	传送数据长度的选择 00: 8 位 01: 16 位 10: 32 位 11: 禁止设置	
5	RPTINT	重复模式中中断的允许/禁止 ^{注1} 0: 禁止发生中断。 1: 允许发生中断。	
4	CHNE	链传送的允许/禁止 ^{注2} 0: 禁止链传送。 1: 允许链传送。	
3	DAMOD	传送目标地址的控制 ^{注3} 0: 固定 1: 递增	
2	SAMOD	传送源地址的控制 ^{注4} 0: 固定 1: 递增	
1	RPTSEL	重复区的选择 ^{注5} 0: 传送目标为重复区。 1: 传送源为重复区。	
0	MODE	传送模式的选择 0: 正常模式 1: 重复模式	

注 1：在 MODE 位为“0”（正常模式）时，RPTINT 位的设置无效。

注 2：必须将 DMACR7 寄存器的 CHNE 位置“0”（禁止链传送）。

注 3：在 MODE 位为“1”（重复模式）并且 RPTSEL 位为“0”（传送目标为重复区）时，DAMOD 位的设置无效。

注 4：在 MODE 位为“1”（重复模式）并且 RPTSEL 位为“1”（传送源为重复区）时，SAMOD 位的设置无效。

注 5：在 MODE 位为“0”（正常模式）时，RPTSEL 位的设置无效。

注 6：不能通过 DMA 传送进行 DMACRj 寄存器的存取。

16.4.7 DMA 块大小寄存器 j (DMBLSj) (j=0~1,3)

此寄存器设置 1 次启动传送数据的块大小。

地址：参照“16.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMBLSj	设置 1 次启动传送数据的块大小	不定值

DMBLSj	传送块大小		
	8 位传送	16 位传送	32 位传送
00H	禁止设置	禁止设置	禁止设置
01H	1 字节	2 字节	4 字节
02H	2 字节	4 字节	8 字节
03H	3 字节	6 字节	12 字节
⋮	⋮	⋮	⋮
FDH	253 字节	506 字节	1012 字节
FEH	254 字节	508 字节	1016 字节
FFH	255 字节	510 字节	1020 字节
⋮	⋮	⋮	⋮
FFFFH	65535 字节	131070 字节	262140 字节

注 1：不能通过 DMA 传送进行 DMBLSj 寄存器的存取。

16.4.8 DMA 传送次数寄存器 j (DMACTj) (j=0~1,3)

此寄存器设置 DMA 的数据传送次数。每当启动 1 次 DMA 传送就减 1。

地址：参照“16.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMRLDj	设置重复模式中的传送次数寄存器的初始值	不定值

DMACTj	传送次数
00H	禁止设置
01H	1 次
02H	2 次
03H	3 次
⋮	⋮
FDH	253 次
FEH	254 次
FFH	255 次
⋮	⋮
FFFFH	65535 次

注 1：不能通过 DMA 传送进行 DMACTj 寄存器的存取。

16.4.9 DMA 传送次数重加载寄存器 j (DMRLDj) (j=0~1,3)

此寄存器设置重复模式中的传送次数寄存器的初始值。在重复模式中，因为将此寄存器的值重新加载到 DMACT 寄存器，所以设置值必须和 DMACT 寄存器的初始值相同。

地址：参照“16.4.2 控制数据的分配”。复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMRLDj	设置重复模式中的传送次数寄存器的初始值	不定值

注：不能通过 DMA 传送进行 DMRLDj 寄存器的存取。

16.4.10 DMA 源地址寄存器 j (DMSARj) (j=0~1,3)

此寄存器指定数据传送时的传送源地址。

当 DMACRj 寄存器的 SZ 位为“01”（16 位传送）时，忽视最低位而作为偶地址进行处理。

当 DMACRj 寄存器的 SZ 位为“10”（32 位传送）时，忽视低 2 位而作为 word 地址进行处理。

地址：参照“16.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
31:0	DMSARj	设置数据传送时的传送源地址	不定值

注：不能通过 DMA 传送进行 DMSARj 寄存器的存取

16.4.11 DMA 目标地址寄存器 j (DMDARj) (j=0~1,3)

此寄存器指定数据传送时的传送目标地址。

当 DMACRj 寄存器的 SZ 位为“01”（16 位传送）时，忽视最低位而作为偶地址进行处理。

当 DMACRj 寄存器的 SZ 位为“10”（32 位传送）时，忽视低 2 位而作为 word 地址进行处理。

地址：参照“16.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
31:0	DMDARj	设置数据传送时的目标地址	不定值

注：不能通过 DMA 传送进行 DMDARj 寄存器的存取。

16.4.12 DMA 启动允许寄存器 i (DMAEN0)

这是控制允许或者禁止通过各通道启动 DMA 的 8 位寄存器。通道和 DMAEN0 位的对应如表 16-5 所示。

能通过 8 位存储器操作指令设置 DMAEN 寄存器。

注 1：必须在不产生对应该位的启动源的位置更改 DMAENi0~DMAENi3 位。

注 2：不能通过 DMA 传送进行 DMAENi 寄存器的存取。

注 3：分配的功能因产品而不同，必须将没有分配功能的位置“0”。

位	符号	描述	复位值
31:4	--	保留，须为 0	0x0
3:0	DMAEN0n (n=0~3)	DMA 启动的允许 in ^{注1} (n=0~3) 0: 禁止启动。 1: 允许启动。	0x0

注 1：根据传送结束中断的发生条件，DMAENin 位变为“0”（禁止启动）。n=0,1,2,3。

注 2：DMAEN02 须为 0。

表 16-5 通道和 DMAEN00~DMAEN03 位的对应

寄存器	DMAENi3 位	DMAENi2 位	DMAENi1 位	DMAENi0 位
DMAEN0	通道 3 允许启动	-	通道 1 允许启动	通道 0 允许启动

注 1：必须将没有分配功能的位置“0”。

注 2：i=0

16.4.13 DMA 基址寄存器（DMABAR）

这是 32 位寄存器，设置保存 DMA 控制数据区起始地址的向量地址以及 DMA 控制数据区的地址。

注 1：必须在将全部的 DMA 启动源设置为禁止启动的状态下更改 DMABAR 寄存器。

注 2：只能改写 1 次 DMABAR 寄存器。

注 3：不能通过 DMA 传送进行 DMABAR 寄存器的存取。

注 4：有关 DMA 控制数据区和 DMA 向量表区的分配，请参照“表 16-4-1 DMA 控制数据区和 DMA 向量表区的分配”的注意。

注 5：设置该寄存器请保持 256Byte 对齐，也就是低 8 位设置为零。DMA 硬件忽略低 8 位。

注 6：该寄存器只能 WORD 访问，BYTE 和 HALFWORD 访问忽略。

位	符号	描述	复位值
31:8	DMABARj	设置数据传送时的目标地址高 24 位	0x0
7:0	DMABARj	设置数据传送时的目标地址低 8 位，因 256 字节对齐，故设置为 0	0x0

注：j=0。

16.4.14 DMAEN0 置位寄存器（DMASET0）

这是 DMA 启动允许寄存器 DMAEN0 的置位寄存器，设置相应的位为 1 则可以将 DMAEN0 的相应位置位为 1。

位	符号	描述	复位值
31:4	--	保留，须为 0	0x0
3:0	DMASET0n (n=0~3)	DMAEN0n 的置位寄存器 (n=0~3) 0: 不产生置位。 1: 将 DMAEN0 的位 n 置位为 1。	0x0

注：DMASET02 须为 0。

16.4.15 DMAEN0 复位寄存器（DMACLR0）

这是 DMA 启动允许寄存器 DMAEN0 的复位寄存器，设置相应的位为 1 则可以将 DMAEN0 的相应位复位为 0。

位	符号	描述	复位值
31:4	--	保留，须为 0	0x0
3:0	DMACLR0n (n=0~3)	DMAEN0n 的复位寄存器 (n=0~3) 0: 不产生复位。 1: 将 DMAEN0 的位 n 复位为 0。	0x0

注：DMACLR02 须为 0。

16.4.16 DMA 的运行

DMA 一旦启动，就从 DMA 控制数据区读控制数据，根据此控制数据进行数据传送，并且将数据传送后的控制数据回写到 DMA 控制数据区。能将 3 组控制数据保存到 DMA 控制数据区，并且进行 3 组数据的传送。传送模式有正常模式和重复模式，传送大小有 8 位传送，16 位传送和 32 位传送。在 DMACRj (j=0~1,3) 寄存器的 CHNE 位为“1”（允许链传送）时，通过 1 个启动源读多个控制数据进行连续的数据传送（链传送）。

通过 32 位 DMSARj 寄存器和 32 位 DMDARj 寄存器分别指定传送源地址和传送目标地址。在数据传送后，根据控制数据递增或者固定 DMSARj 寄存器和 DMDARj 寄存器的值。

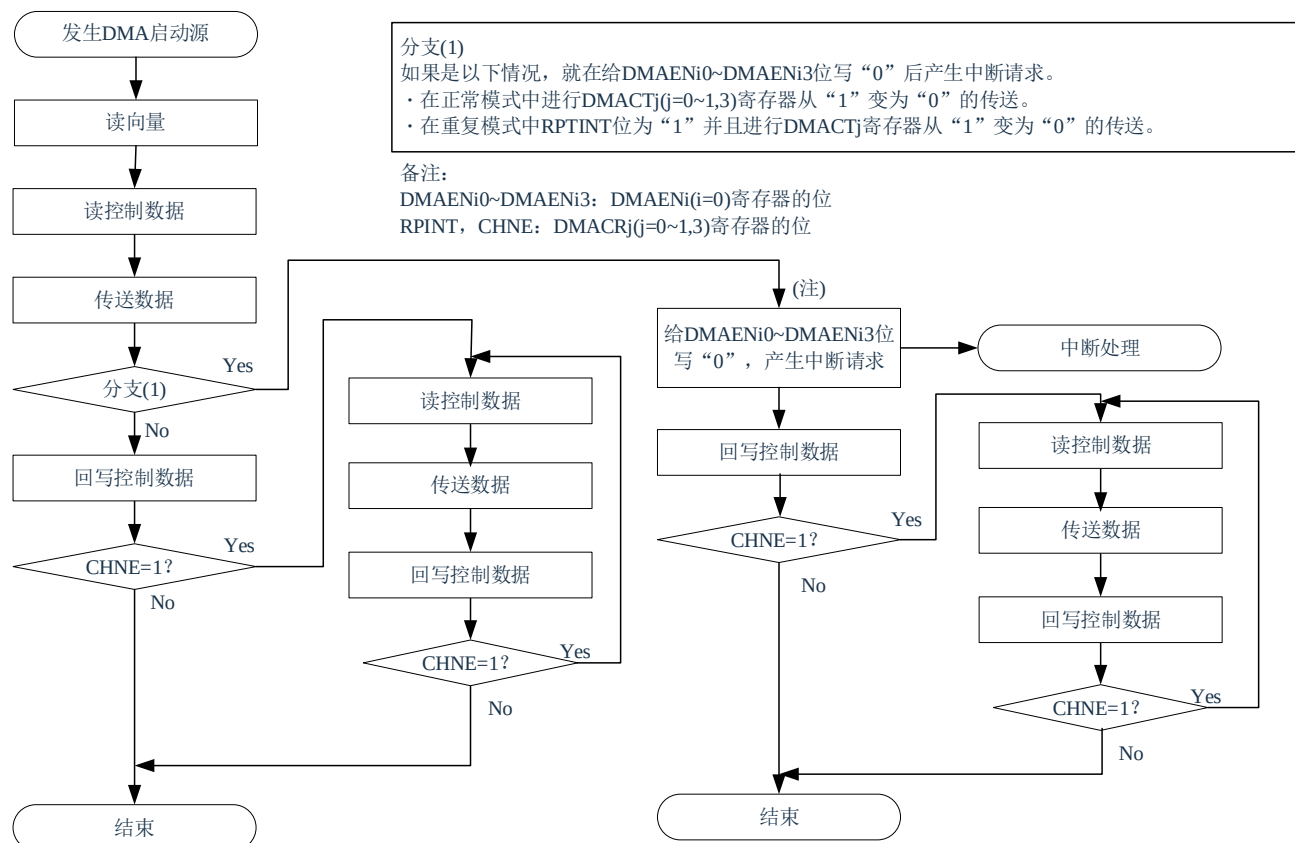
16.4.17 启动源

通过外围功能的中断信号启动 DMA，并且通过 DMAENi (i=0) 寄存器选择启动 DMA 的中断信号。当数据传送（在进行链传送时，连续进行最初的传送）的设置为下述两种情况时，就在 DMA 运行中将对应的 DMAENi 寄存器的 DMAENi0~DMAENi3 位置“0”（禁止启动）。

- 在正常模式中，进行 DMACTj (j=0~1,3) 寄存器变为“0”的传送。
- 在重复模式中，DMACRj 寄存器的 RPTINT 位为“1”（允许发生中断）并且进行 DMACTj 寄存器变为“0”的传送。

DMA 的内部运行流程图如图 16-5 所示。

图 16-5 DMA 的内部运行流程图



注：在通过允许链传送(CHNE=1)的设置启动的数据传送中，不给 DMAENi0~DMAENi3 位写“0”且不产生中断请求。

16.4.18 正常模式

在 8 位传送时，1 次启动的传送数据为 1~65535 字节；在 16 位传送时，1 次启动的传送数据为 2~131070 字节；在 32 位传送时，1 次启动的传送数据为 4~262140 字节。传送次数为 1~65535 次。如果进行 DMACTj（j=0~1,3）寄存器变为“0”的数据传送，就在 DMA 运行中向中断控制器产生对应启动源的中断请求，并且将对应的 DMAENi（i=0）寄存器的 DMAENi0~DMAENi3 位置“0”（禁止启动）。

正常模式的寄存器功能和数据传送分别如表 16-6 和图 16-6 所示

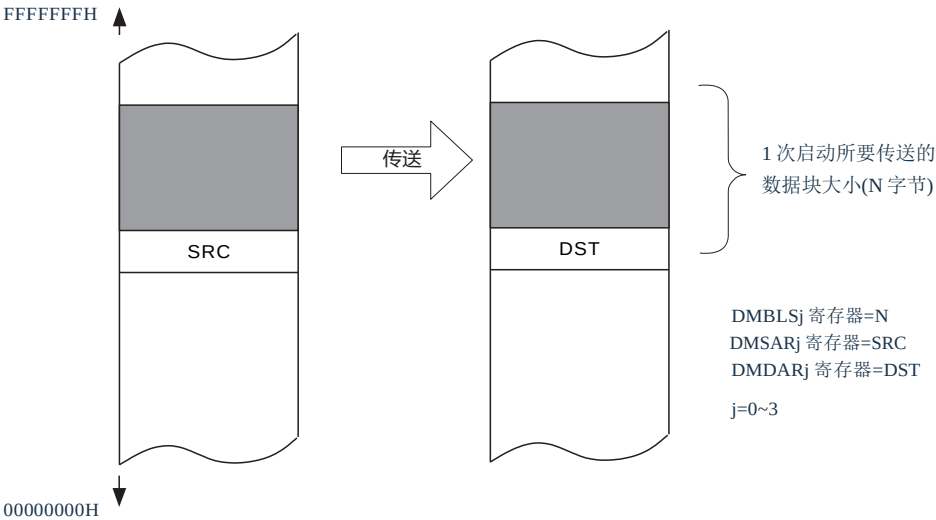
注：j=0

表 16-6 正常模式的寄存器功能

寄存器名	符号	功能
DMA 块大小寄存器 j	DMBLSj	1 次启动所要传送的数据块大小
DMA 传送次数寄存器 j	DMACTj	数据的传送次数
DMA 传送次数重加载寄存器 j	DMRLDj	不使用注。
DMA 源地址寄存器 j	DMSARj	数据的传送源地址
DMA 目标地址寄存器 j	DMDARj	数据的传送目标地址

注：j=0

图 16-6 正常模式的数据传送



DMACR 寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	固定	固定	SRC	DST
0	1	X	0	递增	固定	SRC+N	DST
1	0	X	0	固定	递增	SRC	DST+N
1	1	X	0	递增	递增	SRC+N	DST+N

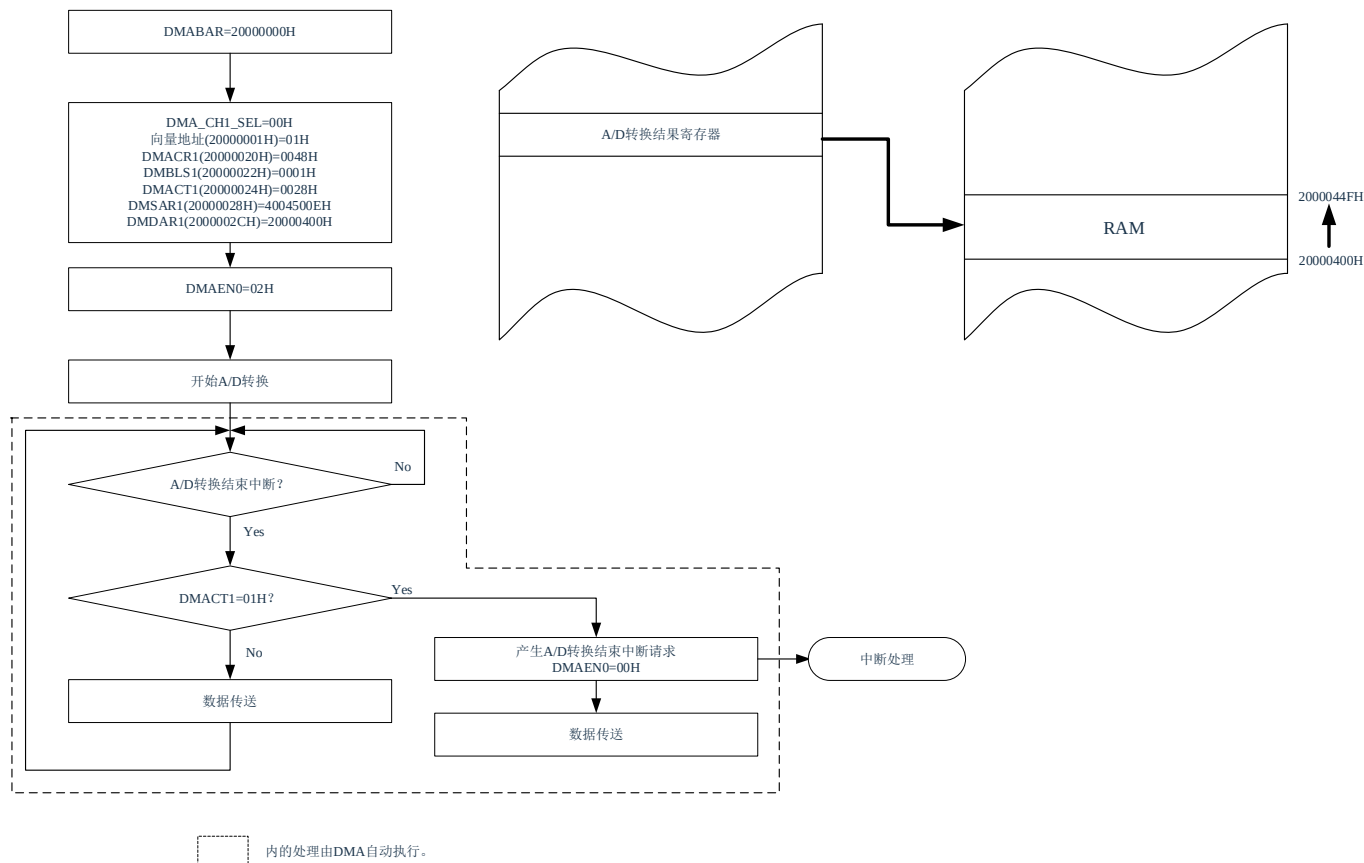
X: “0”或者“1”

（1）正常模式的使用例子 1：通道 1 连续取 A/D 转换结果

通过 A/D 转换结束中断启动 DMA，并且将 A/D 转换结果寄存器的值传送到 RAM。

- 向量地址分配在 20000001H，控制数据分配在 20000020H~2000002FH。
- 将 A/D 转换结果寄存器 ADCR（4004500EH）的 2 字节数据传送 40 次到 RAM 的 20000400H~2000044FH 的 80 字节。

图 16-7：正常模式的使用例子 1：连续取 A/D 转换结果

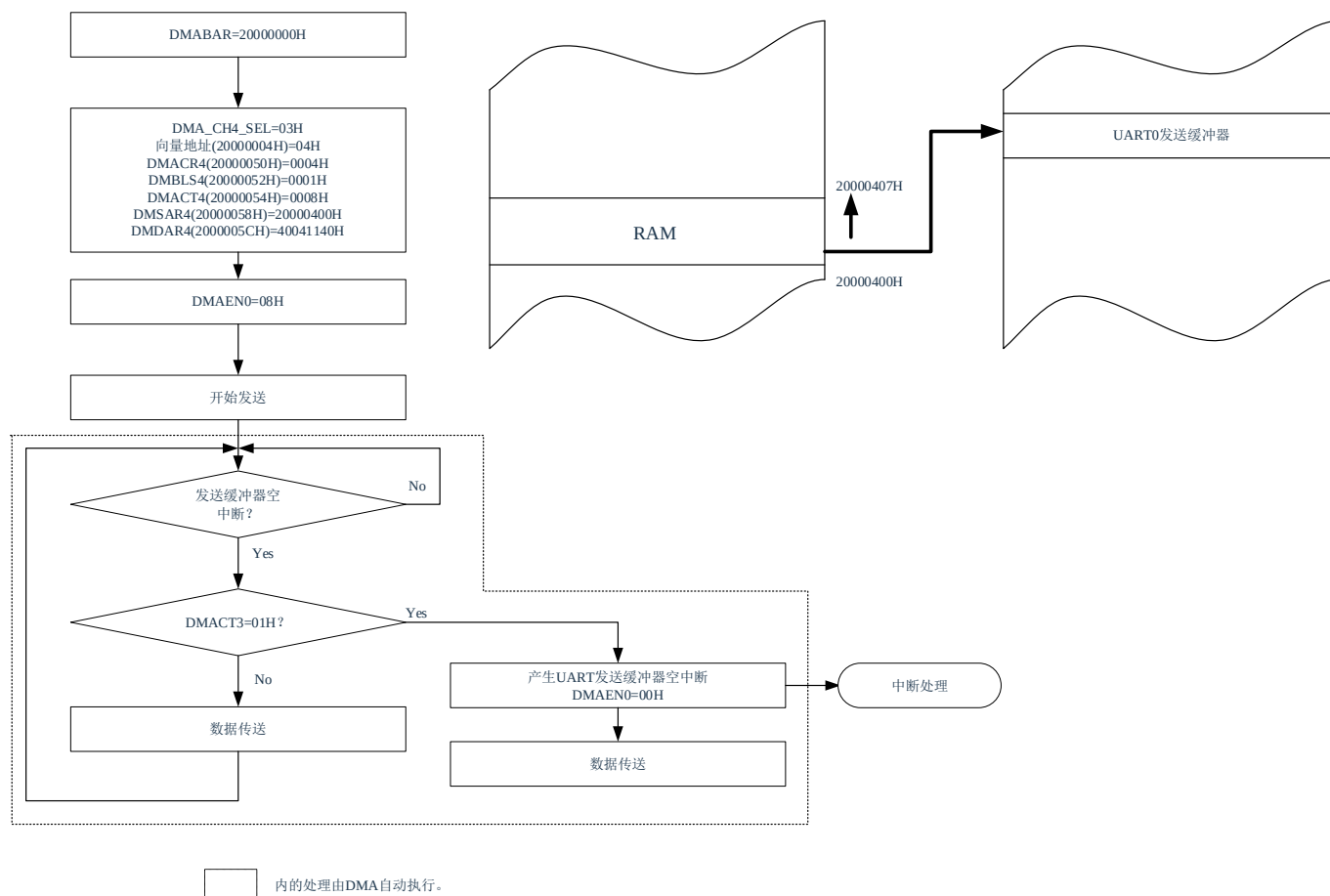


因为是正常模式，所以不使用 DMRLD1 寄存器的值。

通过 UART0 的发送缓冲器空中断启动 DMA，并且将 RAM 的值传送到 UART0 的发送缓冲器。

- 向量地址分配在 20000004H，控制数据分配在 20000050H~2000005FH。
- 将 RAM 的 20000400H~20000407H 的 8 字节传送到 UART0 的发送缓冲器 SDR00（40041140H）。

图 16-8 正常模式的使用例子 2: UART0 连续发送



因为是正常模式，所以不使用 DMRLD3 寄存器的值。必须通过软件开始第 1 次的 UART0 发送。通过发送传送结束启动 DMA，然后自动进行第 2 次以后的发送。

16.4.19 重复模式

1 次启动的传送数据为 1~65535 字节。将传送源或者传送目标指定为重复区，传送次数为 1~65535 次。一旦指定次数的传送结束，就对 DMACTj (j=0~1,3) 寄存器以及指定为重复区的地址进行初始化，然后重复进行传送。当 DMACRj 寄存器的 RPTINT 位为“1”（允许发生中断）并且进行 DMACTj 寄存器变为“0”的数据传送时，就在 DMA 运行中向中断控制器产生对应启动源的中断请求，并且将对应的 DMAENi (i=0) 寄存器的 DMAENi0~DMAENi3 位置“0”（禁止启动）。当 DMACRj 寄存器的 RPTINT 位为“0”（禁止发生中断）时，即使进行 DMACTj 寄存器变为“0”的数据传送，也不产生中断请求，而且 DMAENi0~DMAENi3 位不变为“0”。

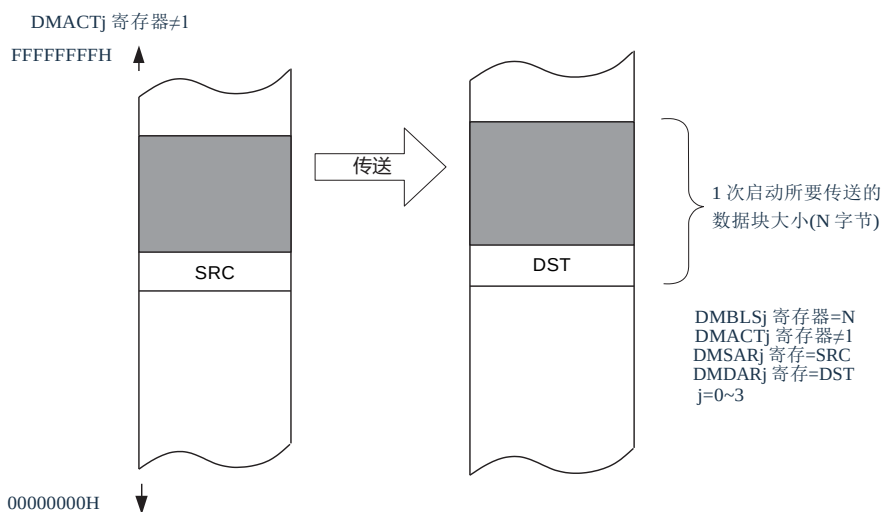
重复模式的寄存器功能和数据传送分别如表 16-7 和图 16-9 所示。

表 16-7：重复模式的寄存器功能

寄存器名	符号	功能
DMA 块大小寄存器 j	DMBLSj	1 次启动所要传送的数据块大小
DMA 传送次数寄存器 j	DMACTj	数据的传送次数
DMA 传送次数重加载寄存器 j	DMRLDj	将此寄存器的值重新加载到 DMACT 寄存器。 (对数据的传送次数进行初始化)
DMA 源地址寄存器 j	DMSARj	数据的传送源地址
DMA 目标地址寄存器 j	DMDARj	数据的传送目标地址

注：j=0~1,3

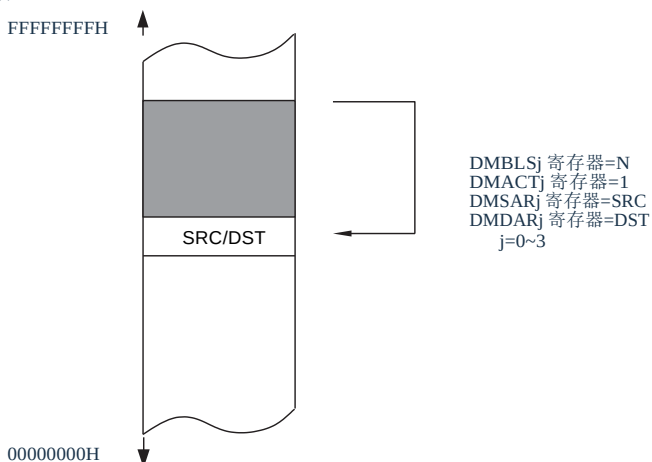
图 16-9 重复模式的数据传送



DMACR 寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC+N	DST
1	X	1	1	重复区	递增	SRC+N	DST+N
X	0	0	1	固定	重复区	SRC	DST+N
X	1	0	1	递增	重复区	SRC+N	DST+N

备注：X：“0”或者“1”

DMACTj 寄存器=1



DMACR 寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC	DST
1	X	1	1	重复区	递增	SRC	DST+N
X	0	0	1	固定	重复区	SRC	DST
X	1	0	1	递增	重复区	SRC+N	DST

备注：X：“0”或者“1”

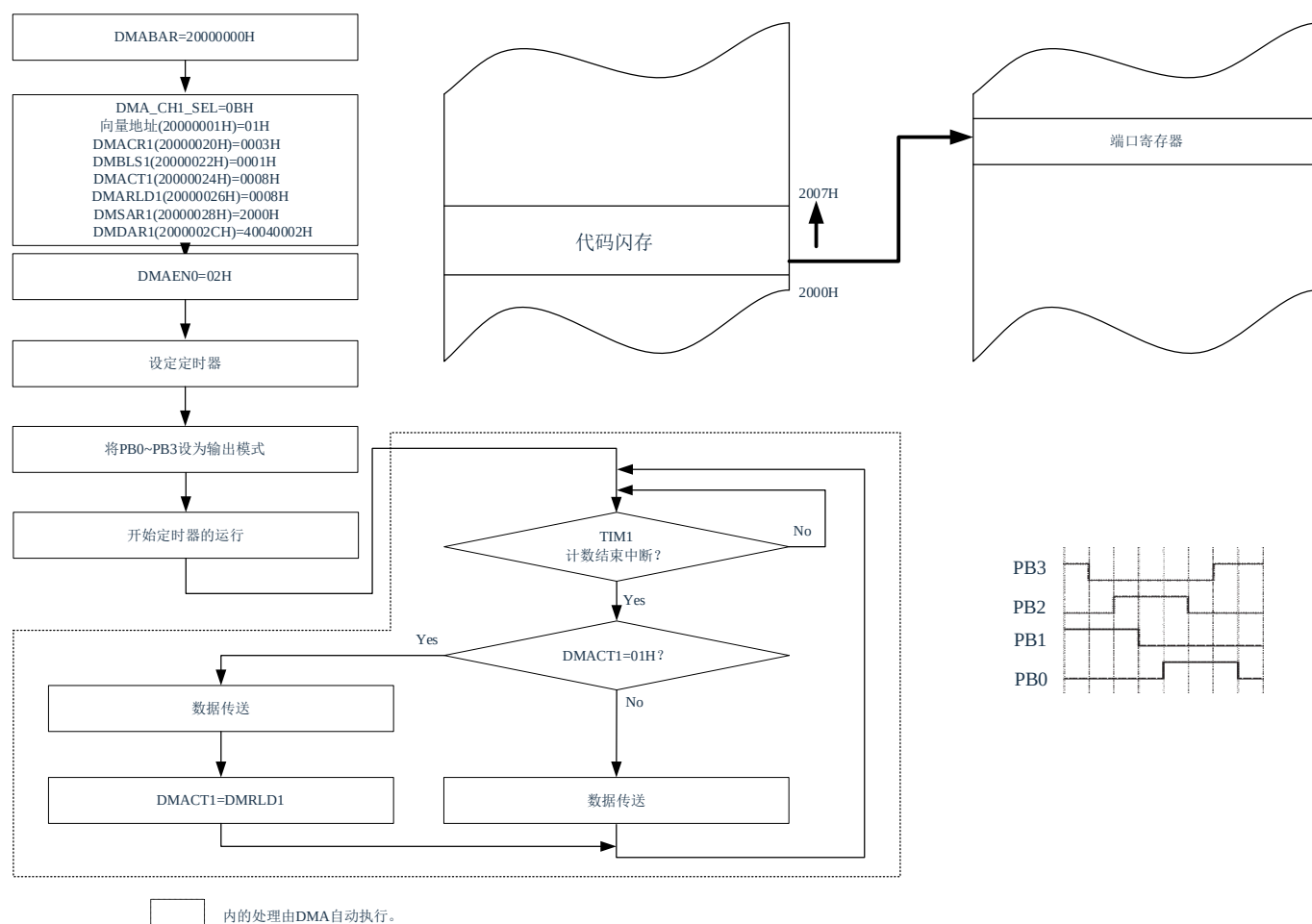
注：在使用重复模式时，必须将重复区的数据长度设置在 65535 字节以内。

1) 重复模式的使用例子：通道 1 使用端口的步进马达控制脉冲输出。

通道 1，当 TIM1 的通道 1 计数结束时，启动 DMA，并且将保存在代码闪存的马达控制脉冲的模式传送到通用端口 PB(0x40040002H)。

- 向量地址分配在 20000001H，控制数据分配在 20000020H~2000002FH。
- 将代码闪存的 02000H~02007H 的 8 字节传送到端口寄存器 1 (40040002H)。
- 禁止重复模式中断。

图 16-10：重复模式的使用例子 1：使用端口的步进马达控制脉冲输出



要停止输出时，必须在停止定时器的运行后清除 DMAEN0 的 bit1。

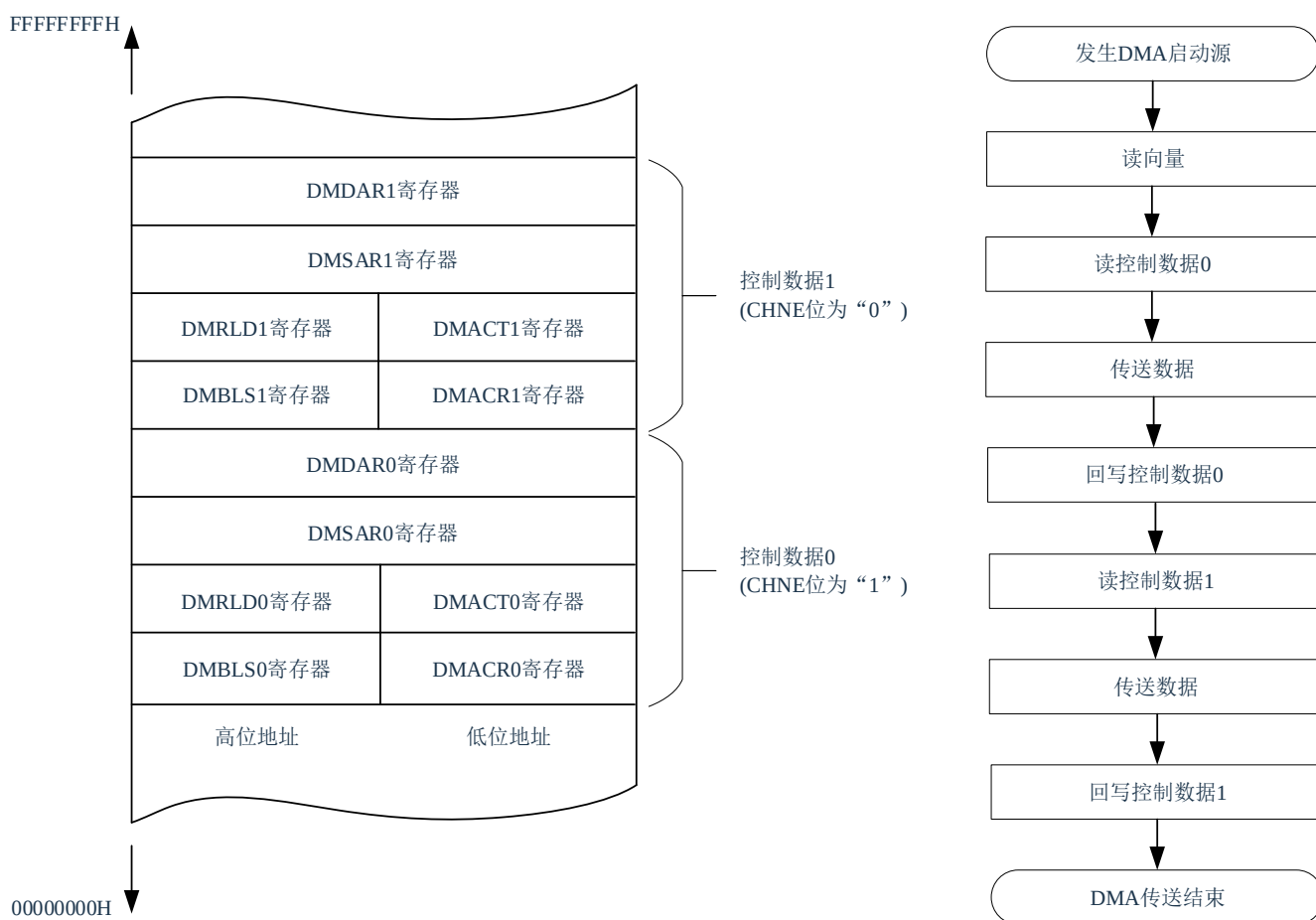
16.4.20 链传送

当 $DMACR_j$ ($j=0\sim1,3$) 寄存器的 CHNE 位为“1（允许链传送）”时，能通过 1 个通道的 1 个启动源连续进行多个数据的传送。

DMA 一旦启动，就通过从启动源对应的向量地址读取的数据来选择控制数据，读被分配在 DMA 控制数据区的控制数据。如果读到的控制数据的 CHNE 位为“1”（允许链传送），就在传送结束后读下一个被分配的控制数据，继续进行传送。重复此操作，直到 CHNE 位为“0”（禁止链传送）的控制数据传送结束为止。

链传送的流程图如图 16-11 所示。

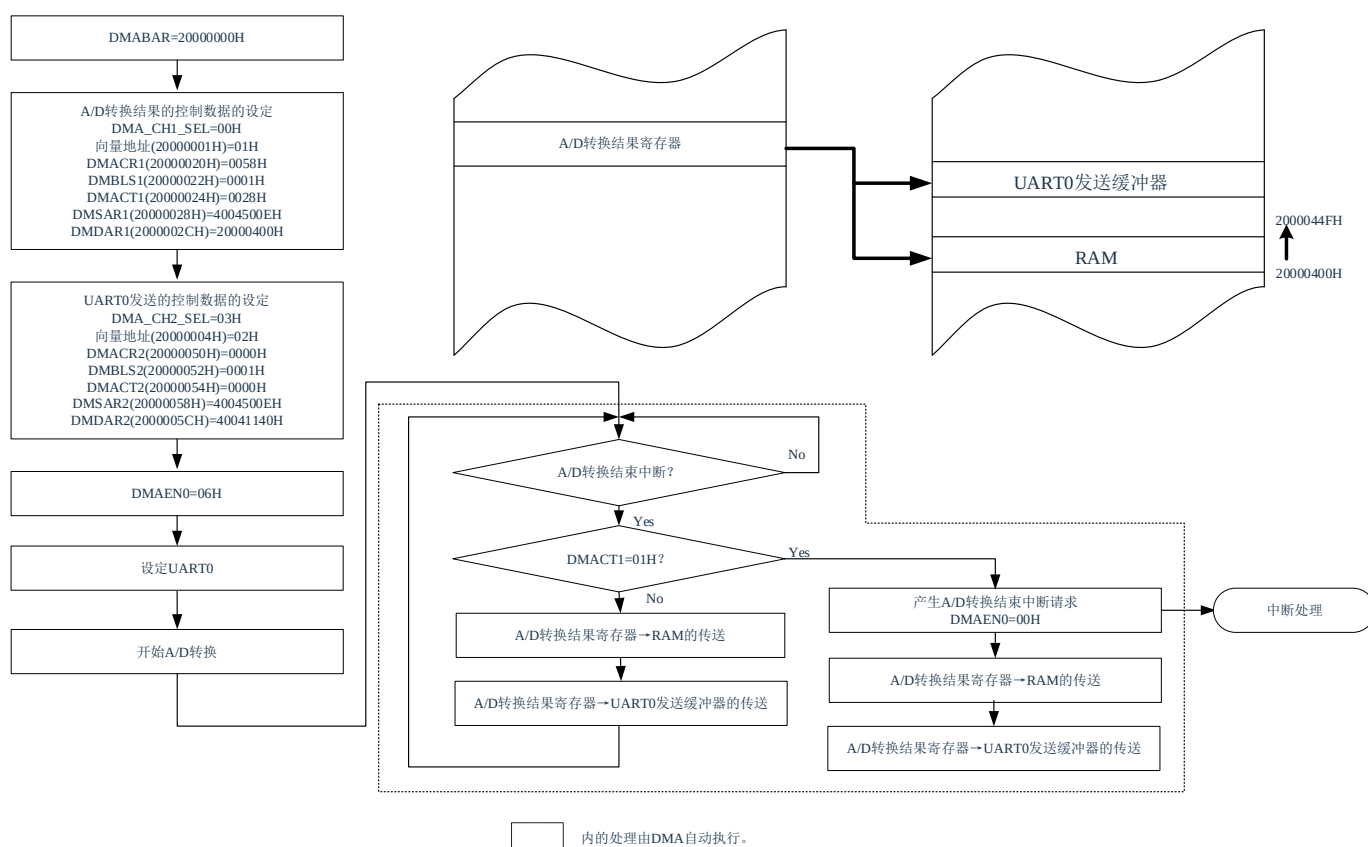
图 16-11： 链传送的流程图



注 1：在链传送的第 2 次以后的数据传送时， $DMAEN_i$ ($i=0$) 寄存器的 $DMAEN_i0\sim DMAEN_i3$ 位不变为“0”（禁止 DMA 启动），并且不产生中断请求。

- 链传送的使用例子：连续取 A/D 转换结果进行 UART0 发送
- 通道 1 通过 A/D 转换结束中断启动 DMA，并且将 A/D 转换结果传送到 RAM 进行 UART 发送。
- 向量地址分别为 20000001H 和 20000002H。
 - A/D 转换结果的控制数据分配在 20000020H~2000002FH。
 - UART0 发送的控制数据分配在 20000030H~2000003FH。
 - 将 A/D 转换结果寄存器 ADCR (4004500EH) 的 2 字节数据传送到 RAM 的 20000400H~2000044FH，并且将 A/D 转换结果寄存器的低位 1 字节 (4004500EH) 传送到 UART0 的发送缓冲器 SDR00 (40041140H)。

图 16-12 链传送的使用例子：连续取 A/D 转换结果进行 UART0 发送



16.5 使用 DMA 时的注意事项

16.5.1 DMA 控制数据和向量表的设置

- 必须在将全部的 DMA 启动源设置为禁止启动的状态下更改 DMA 基址寄存器（DMABAR）。
- 只能改写 1 次 DMA 基址寄存器（DMABAR）。
- 必须在对应的 DMAENi（i=0）寄存器的 DMAENi0~DMAENi3 位为“0”（禁止 DMA 启动）时更改 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj 寄存器的数据。
- 必须在对应的 DMAENi（i=0）寄存器的 DMAENi0~DMAENi3 位为“0”（禁止 DMA 启动）时更改设置在向量表中的 DMA 控制数据区的起始地址。

16.5.2 DMA 控制数据区和 DMA 向量表区的分配

- (1) 能分配 DMA 控制数据和向量表的区域因产品和使用条件而不同。
- (2) 堆栈区、DMA 控制数据区和 DMA 向量表区不能重叠。

16.5.3 DMA 的执行时钟数

DMA 启动时的执行情况和所需的时钟数如表 16-8 所示。

表 16-8: DMA 启动时的执行情况和所需的时钟数

读向量	控制数据		读数据	写数据
	读	回写		
1	4	注 1	注 2	注 2

注 1: 有关回写控制数据所需的时钟数, 请参照“表 16-9 回写控制数据所需的时钟数”

注 2: 有关读写数据所需的时钟数, 请参照“表 16-10 读写数据所需的时钟数”。

表 16-9: 回写控制数据所需的时钟数

DMACR 寄存器的设置				地址设置		控制寄存器的回写				时钟数
DAMOD	SAMOD	RPTSEL	MODE	源	目标	DMACTj 寄存器	DMRLDj 寄存器	DMSARj 寄存器	DMDARj 寄存器	
0	0	X	0	固定	固定	回写	回写	不回写	不回写	1
0	1	X	0	递增	固定	回写	回写	回写	不回写	2
1	0	X	0	固定	递增	回写	回写	不回写	回写	2
1	1	X	0	递增	递增	回写	回写	回写	回写	3
0	X	1	1	重复区	固定	回写	回写	回写	不回写	2
1	X	1	1		递增	回写	回写	回写	回写	3
X	0	0	1	固定	重复区	回写	回写	不回写	回写	2
X	1	0	1	递增		回写	回写	回写	回写	3

注: j=0~7, X: “0”或者“1”

表 16-10: 读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	特殊功能寄存器 (SFR)	扩展特殊功能寄存器 (2ndSFR)	
					无等待	等待
读数据	1	2	4	1	1	1+等待数
写数据	1	—	—	1	1	1+等待数

16.5.4 DMA 的响应时间

DMA 响应时间如表 16-11 所示。DMA 响应时间是指从检测到 DMA 启动源到开始 DMA 传送的时间，不包括 DMA 的执行时钟数。

表 16-11 DMA 的响应时间

	最短时间	最长时间
响应时间	3 个时钟	23 个时钟

但是，在以下情况下 DMA 的响应可能还会延迟。延迟的时钟数因条件而不同。

- 从内部 RAM 执行指令的情况最长响应时间：20 个时钟

注：1 个时钟：1/FCLK（FCLK：CPU/外围硬件时钟）

16.5.5 DMA 的启动源

- 不能在从输入 DMA 启动源到结束 DMA 传送的期间输入相同的启动源。
- 在产生 DMA 启动源的位置，不能操作该启动源对应的 DMA 启动允许位。
- 如果 DMA 启动源发送竞争，就在 CPU 接受 DMA 传送时判断优先级，决定启动启动源。有关启动源的优先级，请参照“16.4.3 向量表”。

16.5.6 待机模式中的运行

状态	DMA 运行
睡眠模式	能运行。
深度睡眠模式	能接受 DMA 启动源，并进行 DMA 传送注 1

注：在深度睡眠模式中，能在检测到 DMA 启动源后进行 DMA 传送，并且在传送结束后返回到深度睡眠模式。但是，因为在深度睡眠模式中代码闪存和数据闪存停止运行，所以不能将闪存设置为传送源。

第17章 A/D 转换器

17.1 A/D 转换器的功能

A/D 转换器是将模拟输入转换为数字值的转换器，最多能控制 26 个模拟通道的 A/D 转换（22 个引脚输入通道和 4 个内部通道）。

A/D 转换器有以下功能。

- 12 位分辨率的 A/D 转换

从 ANI0~ANI4、ANI6~ANI13、ANI16~ANI18、ANI24~ANI26、ANI28~ANI30，从 VDD、GND、BG、TS 中选择 1 个通道的模拟输入，重复进行 12 位分辨率的 A/D 转换。每结束 1 次 A/D 转换，就产生中断请求（INTAD）（选择模式的情况）。

能通过下述的模式组合设置各种 A/D 转换模式。

触发模式	软件触发	通过软件操作来开始转换。
	硬件触发无等待模式	通过检测硬件触发来开始转换。
	硬件触发等待模式	在切断电源的转换待机状态下，通过检测硬件触发来接通电源，在经过 A/D 电源稳定等待时间后自动开始转换。
通道选择模式	选择模式	选择 1 个通道的模拟输入，进行 A/D 转换。
	扫描模式	按顺序对 4 个通道的模拟输入进行 A/D 转换。能选择 ANI0~ANI4、ANI6~ANI13、ANI16~ANI18、ANI24~ANI26、ANI28~ANI30 中连续的 4 个
转换模式	单次转换模式	对所选通道进行 1 次 A/D 转换。
	连续转换模式	对所选通道进行连续的 A/D 转换，直到被软件停止为止。

图 17-1: A/D 转换器的框图

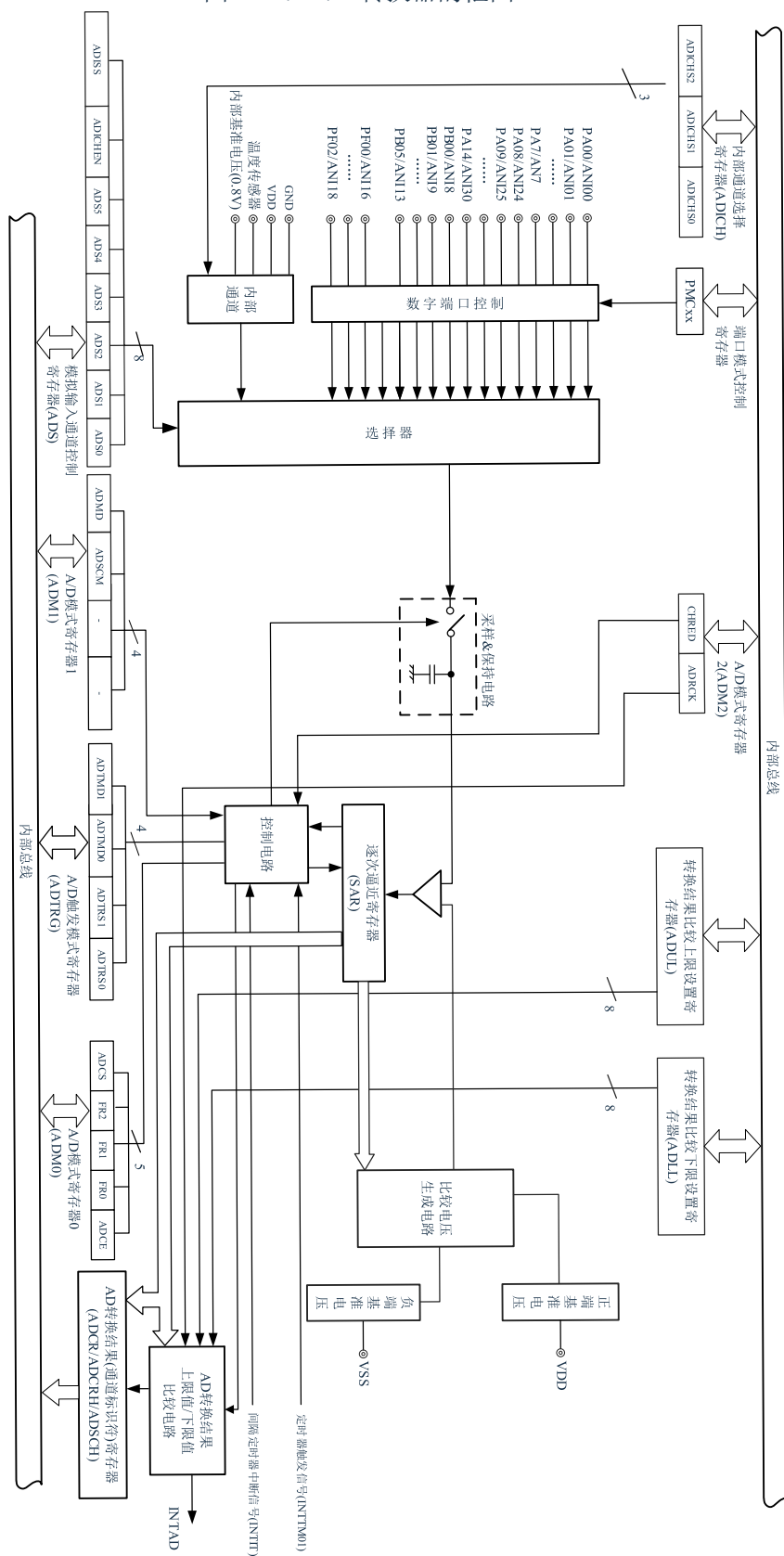
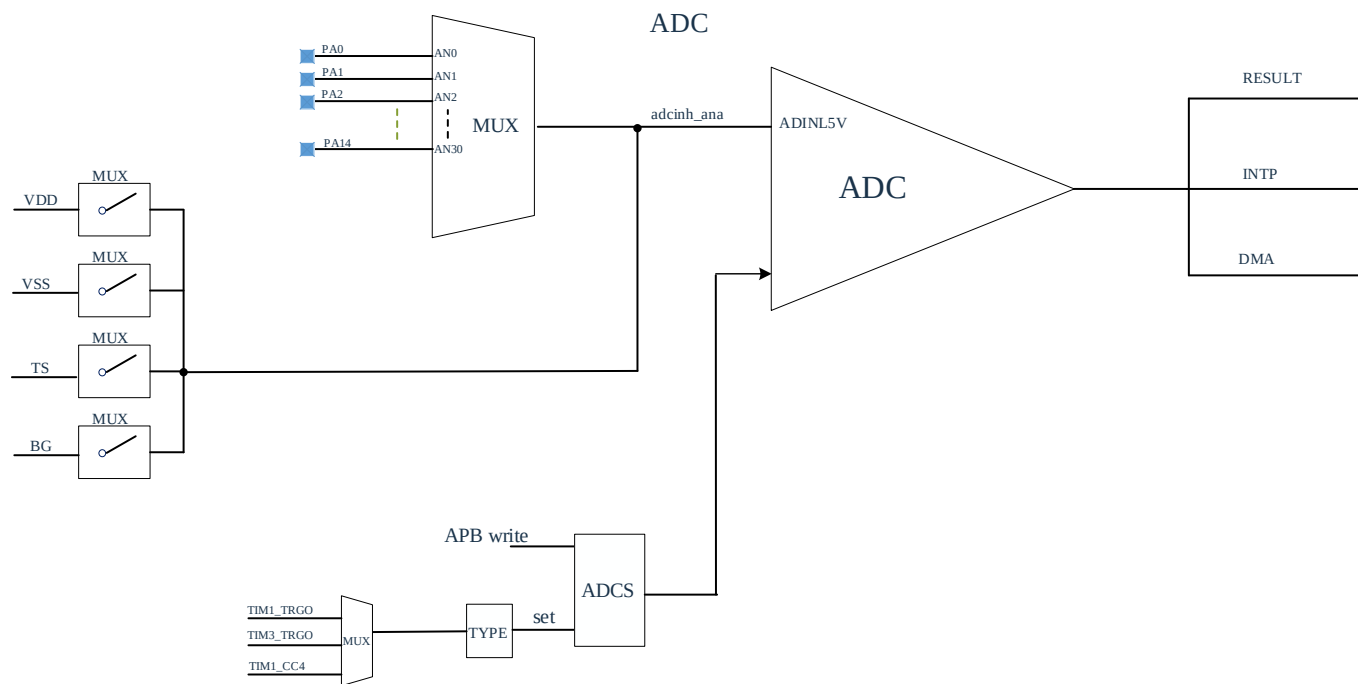


图 17-2 ADC 结构框图



17.2 寄存器映射

ADC 基地址 = 0x40045000 RO: 只读, WO: 只写, R/W: 读写

寄存器名称	偏移量	读写	寄存器描述	复位值
ADM0	0x00	R/W	A/D 转换器的模式寄存器 0	18H
ADM1	0x02	R/W	A/D 转换器的模式寄存器 1	00H
ADM2	0x04	R/W	A/D 转换器的模式寄存器 2	00H
ADTRG	0x06	R/W	A/D 转换器的触发模式寄存器	00H
ADS	0x08	R/W	模拟输入通道指定寄存器	00H
ADICHS	0x09	R/W	内部通道选择寄存器	00H
ADLL	0x0A	R/W	转换结果比较下限值设置寄存器	00H
ADUL	0x0B	R/W	转换结果比较上限值设置寄存器	FFH
ADSCH	0x0D	RO	扫描模式 AD 转换通道标识符寄存器	00H
ADCR	0x0E	RO	12 位 A/D 转换结果寄存器	0000H
ADCRH	0x0F	RO	8 位 A/D 转换结果寄存器	00H
ADSMPWAIT	0x15	R/W	A/D 转换器的采样时间延长控制寄存器	00H

17.3 寄存器说明

17.3.1 A/D 转换器的模式寄存器 0 (ADM0)

用于设置 A/D 转换时钟、转换开始或者停止的寄存器。通过 8 位存储器操作指令设置 ADM0 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	ADCS	A/D 转换运行控制 0: 写: 停止转换运行 读: 停止转换运行/待机状态 1: 写: 允许转换运行 读: 软件触发模式时, 转换运行状态; 硬件触发等待模式时, A/D 电源等待稳定状态+转换运行状态	0
6	-	保留, 必须为 0	-
5:3	FR<2:0>	ADC 时钟分频选择位 000: F _{CLK} /16 001: F _{CLK} /12 010: F _{CLK} /8 011: F _{CLK} /6 100: F _{CLK} /3 101: F _{CLK} /2 110: F _{CLK} /1 其他: 保留, 禁止选择	0x3
2:1	-	保留, 必须为 0	-
0	ADCE	A/D 电压比较器使能位注 2 0: 禁止 A/D 电压比较器 1: 使能 A/D 电压比较器	0

注 1: 有关 FR2~FR0 位 A/D 转换的详细内容, 请参照“表 17-3 A/D 转换时间的选择”。

注 2: A/D 转换器开始运行需要 1 μ s 稳定时间。在软件触发模式或者硬件触发无等待模式中, 在将 ADCE 位置“1”后至少经过 1 μ s 时间, 然后将 ADCS 位置“1”, 则本次转换结果有效。如果等待时间小于 1 μ s 而将 ADCS 位置“1”, 就必须忽视本次转换结果。在硬件触发等待模式中, 由设计保证 1 μ s 的等待时间。

注意:

1. 必须在转换停止状态 ADCS=0 下更改 FR2~FR0 位。
2. 禁止进行 ADCS=1、ADCE=0 的设置。
3. 禁止通过 8 位操作指令将 ADCS=0、ADCE=0 的状态设置为 ADCS=1、ADCE=1。必须按照“17.6 A/D 转换器的设置流程图”的步骤进行设置。

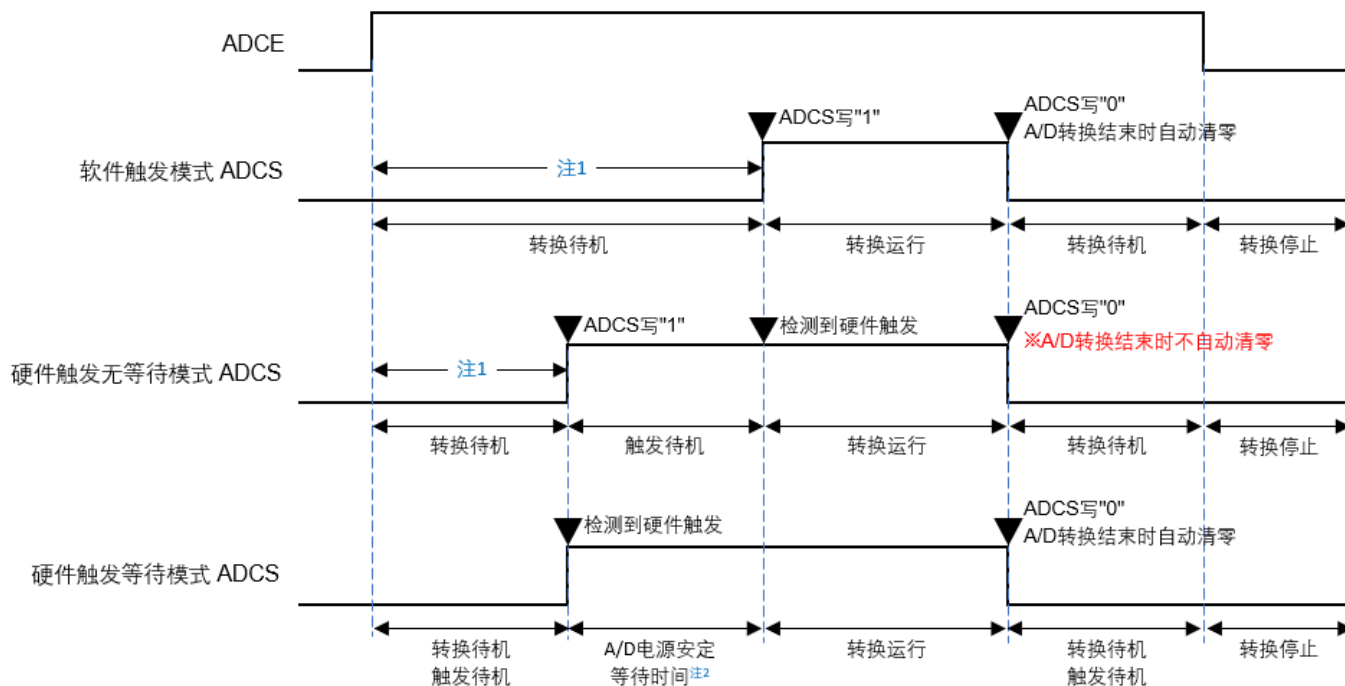
表 17-1 ADCS 位和 ADCE 位的设置

ADCS	ADCE	A/D 转换运行
0	0	转换停止状态
0	1	转换待机状态
1	0	禁止设置。
1	1	转换运行状态

表 17-2: ADCS 位的置位和清除条件

A/D 转换模式			置位条件	清除条件
软件触发	选择模式	连续转换模式	当给 ADCS 位写“1”时	当给 ADCS 位写“0”时
		单次转换模式		•当给 ADCS 位写“0”时 •在 A/D 转换结束时自动清“0”。
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		•当给 ADCS 位写“0”时 •当设置的 4 个通道转换结束时，自动清“0”。
硬件触发无等待模式	选择模式	连续转换模式	当给 ADCS 位写“1”时	当给 ADCS 位写“0”时
		单次转换模式		•当给 ADCS 位写“0”时
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		•当给 ADCS 位写“0”时
硬件触发等待模式	选择模式	连续转换模式	当输入硬件触发时	当给 ADCS 位写“0”时
		单次转换模式		•当给 ADCS 位写“0”时 •在 A/D 转换结束时自动清“0”。
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		•当给 ADCS 位写“0”时 •当设置的 4 个通道转换结束时，自动清“0”。

图 17-3 使用 A/D 各种模式时的动作状态图



注 1: 在软件触发模式或者硬件触发无等待模式中，为了稳定内部电路，从 ADCE 位上升到 ADCS 位上升的时间至少需要 1us。

注 2: 在硬件触发等待模式，A/D 电源稳定时间 1us 由设计保证。

注意:

1. 要使用硬件触发等待模式时，禁止将 ADCS 位置“1”（当检测到硬件触发信号时，自动切换为“1”）。但是，为了设置为 A/D 转换待机状态，能将 ADCS 位置“0”。
2. 必须在 ADCS 位为“0”（停止转换/转换待机状态）时改写 ADCE 位。

3. 为了结束 A/D 转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2 个 F_{CLK} 时钟+A/D 转换时间

硬件触发等待模式时：2 个 F_{CLK} 时钟+A/D 电源稳定等待时间+A/D 转换时间

备注： F_{CLK} ：CPU/外围硬件的时钟频率

表 17-3 A/D 转换时间的选择(1/2)

1. 无 A/D 电源稳定等待时间（软件触发模式/硬件触发无等待模式）

A/D 转换器的模式 寄存器 0 (ADM0)			A/D 采样时间延长寄存器 (ADSMPWAIT)	转换时钟 ADCLK 的频率 (F_{AD})	12 位分辨率的转换时间	
FR2	FR1	FR0	ADSMPWAIT		转换时钟数	转换时间
0	0	0	0	$F_{clk}/16$	16 个 ADCLK (采样时钟数：4 个 ADCLK)	$16/F_{AD}$
0	0	1		$F_{clk}/12$		
0	1	0		$F_{clk}/8$		
0	1	1		$F_{clk}/6$		
1	0	0		$F_{clk}/3$		
1	0	1		$F_{clk}/2$		
1	1	0		$F_{clk}/1$		
0	0	0	1	$F_{clk}/16$	20 个 ADCLK (采样时钟数：8 个 ADCLK)	$20/F_{AD}$
0	0	1		$F_{clk}/12$		
0	1	0		$F_{clk}/8$		
0	1	1		$F_{clk}/6$		
1	0	0		$F_{clk}/3$		
1	0	1		$F_{clk}/2$		
1	1	0		$F_{clk}/1$		

注意：1.要将 FR2~FR0 位改写为不同数据时，必须在转换停止状态（ADCS=0）下进行。

备注： F_{clk} ：CPU/外围硬件的时钟频率

表 17-3： A/D 转换时间的选择(2/2)

2. 有 A/D 电源稳定等待时间（硬件触发等待模式注 1）

A/D 转换器的模式 寄存器 0 (ADM0)			A/D 采样时间延长寄 存器 (ADSMPWAIT)	转换时钟 ADCLK 的频率(F_{AD})	A/D 电源 稳定等待 时间	转换时钟数	A/D 电源稳定 等待时间 +转换时间
FR2	FR1	FR0	ADSMPWAIT				
0	0	0	0	$F_{clk}/16$	$1\mu s$	16 个 ADCLK (采样时钟数：4 个 ADCLK)	$1\mu s+16/F_{AD}$
0	0	1		$F_{clk}/12$			
0	1	0		$F_{clk}/8$			
0	1	1		$F_{clk}/6$			
1	0	0		$F_{clk}/3$			
1	0	1		$F_{clk}/2$			
1	1	0		$F_{clk}/1$			
0	0	0	1	$F_{clk}/16$	$1\mu s$	20 个 ADCLK (采样时钟数：8 个 ADCLK) ADCLK)	$1\mu s+20/F_{AD}$
0	0	1		$F_{clk}/12$			
0	1	0		$F_{clk}/8$			
0	1	1		$F_{clk}/6$			
1	0	0		$F_{clk}/3$			

1	0	1		Fclk/2			
1	1	0		Fclk/1			

注 1：在连续转换模式时，仅在第 1 次检测到硬件触发后，发生 A/D 电源稳定等待时间（参照表 17-3）。

注意：

- 1. 要将 FR2~FR0 位改写为不同数据时，必须在转换停止状态（ADCS=0）下进行。
- 2. 硬件触发等待模式中的转换时间包含检测到硬件触发后的 A/D 电源稳定等待时间。

备注：F_{CLK}：CPU/外围硬件的时钟频率。

17.3.2 A/D 转换器的模式寄存器 1 (ADM1)

这是设置 A/D 转换模式的寄存器。

通过 8 位存储器操作指令设置 ADM1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	ADMD	A/D 通道选择模式 0: 选择模式 1: 扫描模式	0
6:4	-	保留，必须为 0	-
3	ADSCM	A/D 转换模式选择位 0: 连续转换模式 1: 单次转换模式	0
2:0	-	保留	-

注意：

1. 要改写 ADM1 寄存器时，必须在转换停止状态（ADCS=0，ADCE=0）下进行。
2. 为了正常结束 A/D 转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2 个 F_{CLK} 时钟+A/D 转换时间

硬件触发等待模式时：2 个 F_{CLK} 时钟+A/D 电源稳定等待时间+A/D 转换时间

备注 1：F_{CLK}：CPU/外围硬件的时钟频率。

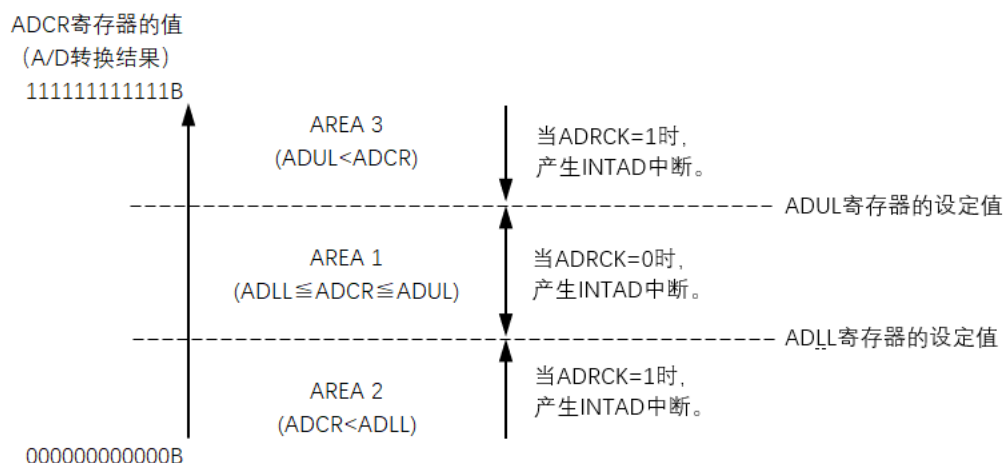
17.3.3 A/D 转换器的模式寄存器 2 (ADM2)

通过 8 位存储器操作指令设置 ADM2 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:4	-	保留，必须为 0	-
3	ADRCK	转换结果上限值和下限值的检查 AREA1~AREA3 的中断信号 (INTAD) 的产生范围如图 17-4 所示 0: 当 ADLL 寄存器≤ADCR 寄存器≤ADUL 寄存器 (AREA1) 时，产生中断信号 (INTAD)。 1: 当 ADCR 寄存器<ADLL 寄存器 (AREA2) 或者 ADUL 寄存器<ADCR 寄存器 (AREA3) 时，产生中断信号 (INTAD)。	0
2	-	保留，必须为 0	-
1	CHRED	A/D 转换器扫描模式时通道标识的输出使能 0: 扫描模式时，不输出转换通道标识号 1: 扫描模式时，输出转换通道标识号	0
0	-	保留，必须为 0	0

图 17-4 ADRCK 位的中断信号产生范围



注意：

1. 要改写 ADM2 寄存器时，必须在转换停止状态 (ADCS=0) 下进行。

备注：当不发生 INTAD 时，A/D 转换结果不保存到 ADCR 寄存器和 ADCRH 寄存器。

17.3.4 A/D 转换器的触发模式寄存器（ADTRG）

这是设置 A/D 转换触发模式和硬件触发信号的寄存器。

通过 8 位存储器操作指令设置 ADTRG 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:6	ADTMD<1:0>	A/D 转换触发模式选择 00: 软件触发 01: 软件触发 10: 硬件触发无等待模式 11: 硬件触发等待模式	0x0
5:2	-	保留，必须为 0	-
1:0	ADTRS<1:0>	硬件触发信号选择位 00: TIM1_TRGO 01: TIM1_CC4 10: 保留 11: TIM3_TRGO	0x0

注意：

1. 要改写 ADTRG 寄存器时，必须在转换停止状态（ADCS=0，ADCE=0）下进行。
2. 为了正常结束 A/D 转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2 个 F_{CLK} 时钟+A/D 转换时间

硬件触发等待模式时：2 个 F_{CLK} 时钟+A/D 电源稳定等待时间+A/D 转换时间

备注 1: F_{CLK} : CPU/外围硬件的时钟频率

17.3.5 模拟输入通道指定寄存器（ADS）

这是指定要进行 A/D 转换的模拟电压输入通道的寄存器。

通过 8 位存储器操作指令设置 ADS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	ADISS	详见下表	0
6	ADICHEN	详见下表	0
5:0	ADS<5:0>	详见下表	0x0

选择模式（ADM1.ADMD=0）

ADS 寄存器设置值								CH 选择
ADISS	ADICHEN	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	0	ANI0
0	0	0	0	0	0	0	1	ANI1
0	0	0	0	0	0	1	0	ANI2
0	0	0	0	0	0	1	1	ANI3
0	0	0	0	0	1	0	0	ANI4
0	0	0	0	0	1	0	1	ANI5
0	0	0	0	0	1	1	0	ANI6
0	0	0	0	0	1	1	1	ANI7
0	0							
0	0	0	1	1	1	1	0	ANI30
0	0	1	1	1	1	1	1	通道全关
0	1	0	0	0	0	0	0	内部通道使能
其他								禁止设置

备注：CMS32C020 无 ANI5,ANI14-ANI15、ANI18-ANI23、ANI27、AIN31-ANI32，禁止设置。

扫描模式（ADM1.ADMD=1）

ADISS	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0	模拟输入通道			
							扫描 0	扫描 1	扫描 2	扫描 3
0	0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	0	0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	0	0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	0	0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	0	0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
0	0	0	1	1	0	0	ANI12	ANI13	ANI14	ANI15
0	0	0	1	1	0	1	ANI13	ANI14	ANI15	ANI16
0	0	0	1	1	1	0	ANI14	ANI15	ANI16	ANI17
0	0	0	1	1	1	1	ANI15	ANI16	ANI17	ANI18
0	0	1	0	0	0	0	ANI16	ANI17	ANI18	ANI19
0	0	1	0	0	0	1	ANI17	ANI18	ANI19	ANI20
0	0	1	1	0	0	0	ANI24	ANI25	ANI26	ANI27
0	0	1	1	0	0	1	ANI25	ANI26	ANI27	ANI28
0	0	1	1	0	1	0	ANI26	ANI27	ANI28	ANI29
0	0	1	1	0	1	1	ANI27	ANI28	ANI29	ANI30
0	0	1	1	1	0	0	ANI28	ANI29	ANI30	ANI31
0	0	1	1	1	0	1	ANI29	ANI30	ANI31	ANI32
其他							禁止设置			

注意：

1. 必须将 bit4,bit5 和 bit6 置“0”。
2. 对于由 PMCx 寄存器设置为模拟输入的端口，才可以通过 ADS 指定为模拟输入进行 A/D 转换。
3. 对于由端口模式控制寄存器（PMCxx）设置为数字输入/输出的引脚，不能通过 ADS 寄存器进行设置。
4. 要改写 ADISS 位时，必须在转换停止状态（ADCS=0、ADCE=0）下进行。
5. 要转移到深度睡眠模式时或者要在 CPU 以副系统时钟运行中转移到睡眠模式时，不能将 ADISS 位置“1”。
6. CMS32C020 无 ANI5,ANI14-ANI15、ANI18-ANI23、ANI27、AIN31-ANI32，禁止设置。

17.3.6 A/D 内部通道选择寄存器（ADICHS）

这是设置 A/D 内部通道的寄存器。
通过 8 位存储器操作指令设置 ADICH 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:3	-	保留，必须为 0	-
2:0	ADICHS<2:0>	内部通道选择位（仅在 ADS 寄存器使能内部通道时 (ADS.ADICHEN=1)才有效） 000: VDD 001: GND 010: TS(温度传感器信号) 011: BG(0.8V) 其他: 保留，禁止选择	0x0

17.3.7 12 位 A/D 转换通道标识符寄存器（ADSCHS）

这是保存 A/D 转换通道标识符的 8 位寄存器，此寄存器只可读。每当 A/D 转换结束时，就输出当前 AD 转换结果对应通道。

此寄存器在选择模式时读出值固定为“0”，在扫描模式时可由 ADM2.CHRDE=1 配置为此次转换结果的通道号。

通过 9 位存储器操作指令读 ADCHO 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:6	-	保留	
5:0	ADSCH<5:0>	转换通道标识符 选择模式(ADM1.ADMD=0)时，ADCH[5:0]读出值始终为 6'b0； 扫描模式(ADM1.ADMD=1)时，且 ADM2.CHEDE=1，ADSCH[5:0]结果对应以通道。 000000: ANI0 000001: ANI1 000010: ANI2 011110: ANI30	0x0

备注：CMS32C020 无 ANI5,ANI14-ANI15、ANI18-ANI23、ANI27、AIN31-ANI32，禁止设置

17.3.8 12 位 A/D 转换结果寄存器（ADCR）

这是保存 A/D 转换结果的 16 位寄存器，此寄存器只可读。每当 A/D 转换结束时，就从逐次逼近寄存器（SAR）装入转换结果注。

通过 16 位存储器操作指令读 ADCR 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

注：如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设置（参照图 17-4））的设置值范围内，就不保存 A/D 转换结果。

位	符号	描述	复位值
15:12	-	保留	-
11:0	ADCR<11:0>	A/D 转换结果	0

注意：

- 1.如果在仅仅需要 8 位分辨率 A/D 转换结果，可以通过 ADCRH 寄存器读取转换结果的高 8 位。
- 2.当对 ADCR 寄存器进行 16 位存取时，能从 bit11 依次读转换结果的高 12 位。

17.3.9 8 位 A/D 转换结果寄存器（ADCRH）

这是保存 A/D 转换结果的 8 位寄存器，保存 12 位分辨率的高 8 位^注。

通过 8 位存储器操作指令读 ADCRH 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:0	ADCRH<7:0>	A/D 转换结果	0x0

注：如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设置（参照图 17-4）的设置值范围内，就不保存 A/D 转换结果。

注意：必须在转换结束后并且在配置 ADM0、ADS 寄存器前读转换结果。否则，就可能读不到正确的转换结果。

17.3.10 A/D 采样时间延长寄存器（ADSMPWAIT）

此寄存器用于延长 A/D 采样时间。

通过 8 位存储器操作指令设定 ADSMPWAIT 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	-	保留，必须为 0	-
0	ADSMPWAIT	A/D 采样时间选择位 0: A/D 采样时间为 4 个 ADCLK 1: A/D 采样时间为 8 个 ADCLK	0

17.3.11 转换结果比较上限值设置寄存器（ADUL）

这是用于检查 A/D 转换结果上限值的设置寄存器。

将 A/D 转换结果和 ADUL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2（ADM2）的 ADRCK 位的设置范围内（参照图 17-4）控制中断信号（INTAD）的产生。通过 8 位存储器操作指令设置 ADUL 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

注意：1.只将 12 位 A/D 转换结果寄存器（ADCR）的高 8 位和 ADUL 寄存器以及 ADLL 寄存器进行比较。

2.要改写 ADUL 寄存器和 ADLL 寄存器时，必须在转换停止状态（ADCS=0）下进行。

3.在设置 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL>ADLL。

位	符号	描述	复位值
7:0	ADUL<7:0>	转换结果比较上限值	0xff

17.3.12 转换结果比较下限值设置寄存器（ADLL）

这是用于检查 A/D 转换结果下限值的设置寄存器。

将 A/D 转换结果和 ADLL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2（ADM2）的 ADRCK 位的设置范围内（参照图 17-4）控制中断信号（INTAD）的产生。通过 8 位存储器操作指令设置 ADLL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:0	ADLL<7:0>	转换结果比较下限值	0x0

- 注意: 1.只将 12 位 A/D 转换结果寄存器（ADCR）的高 8 位和 ADUL 寄存器以及 ADLL 寄存器进行比较。
- 2.要改写 ADUL 寄存器和 ADLL 寄存器时，必须在转换停止状态（ADCS=0）下进行。
- 3.在设置 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL>ADLL。

17.4 输入电压和转换结果

模拟输入引脚（ANI0～ANI4、ANI6～ANI13、ANI16～ANI18、ANI24～ANI26、ANI28～ANI30）的模拟输入电压和理论上的 A/D 转换结果（12 位 A/D 转换结果寄存器（ADCR））有以下表达式的关系。

$$\text{ADCR} = \text{INT} (V_{\text{AIN}} / AV_{\text{REF}} \times 4096 + 0.5) \text{ or } (\text{ADCR} - 0.5) \times AV_{\text{REF}} / 4096 \leq V_{\text{AIN}} < (\text{ADCR} + 0.5) \times AV_{\text{REF}} / 4096$$

INT(): 将括号中的数值的整数部分返回的函数

V_{AIN} : 模拟输入电压

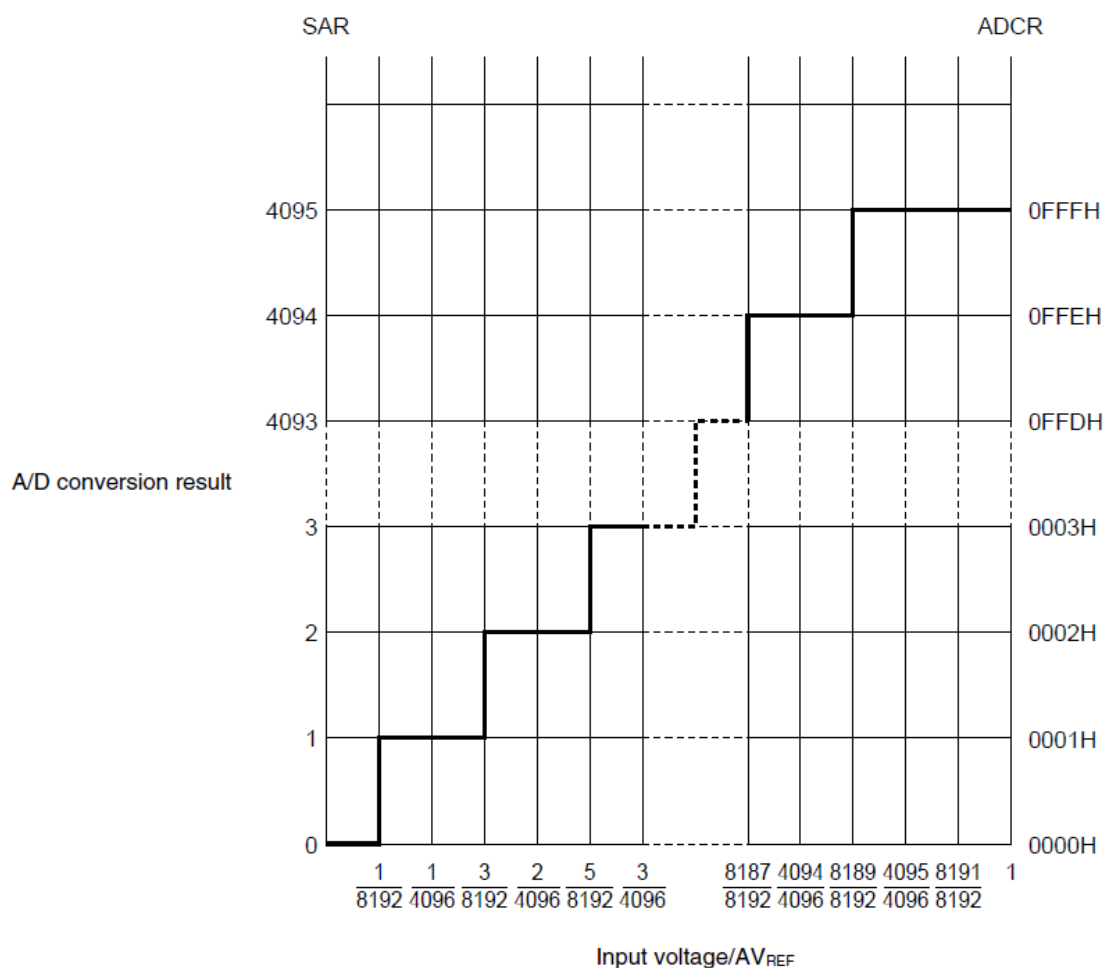
AV_{REF} : AV_{REF} 引脚电压

ADCR: A/D 转换结果寄存器（ADCR）的值

SAR: 逐次逼近寄存器

模拟输入电压和 A/D 转换结果的关系如图 17-5 所示。

图 17-5: 模拟输入电压和 A/D 转换结果的关系



备注: AV_{REF} 是 A/D 转换器的正 (+) 基准电压。

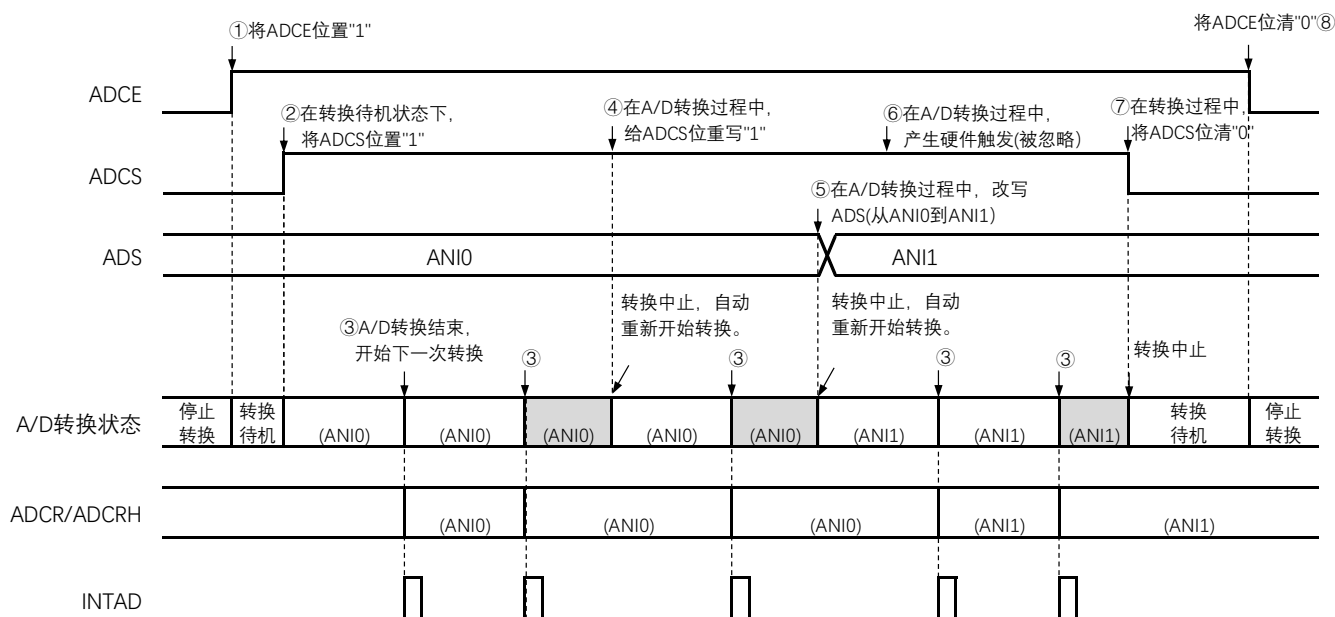
17.5 A/D 转换器的运行模式

A/D 转换器的各模式的运行如下所示。有关各模式的设置步骤，请参照“17.6 A/D 转换器的设置流程图”。

17.5.1 软件触发模式（选择模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1 μ S）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。
- ③ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换。
- ④ 如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后重新开始转换。
- ⑤ 如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。
- ⑥ 即使在转换过程中输入硬件触发也不开始 A/D 转换。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入 A/D 转换待机状态。
- ⑧ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。

图 17-6 软件触发模式（选择模式、连续转换模式）的运行时序例子



17.5.4 软件触发模式（扫描模式、单次转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。

在通过软件对稳定等待时间（1us）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描 0~扫描 3 的 4 个模拟输入通道进行 A/D 转换。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。

连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。

在 4 个通道的 A/D 转换结束后，ADCS 位自动清“0”，进入 A/D 转换待机状态。

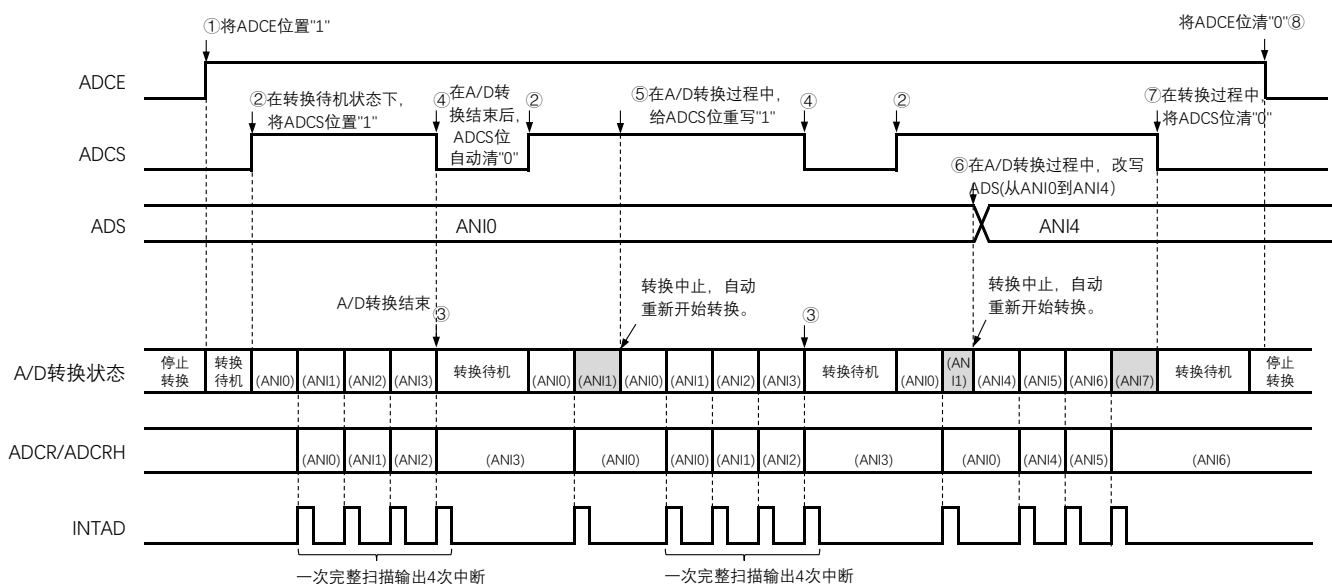
如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入 A/D 转换待机状态。

如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。即使在 A/D 转换待机的状态下输入硬件触发也不开始 A/D 转换。

图 17-9：软件触发模式（扫描模式、单次转换模式）的运行时序例子



17.5.5 硬件触发无等待模式（选择模式、连续转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。

在通过软件对稳定等待时间（1μs）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将 ADCS 位置“1”也不开始 A/D 转换。

如果在 ADCS 位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。

如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换。

如果在转换过程中输入硬件触发，当前的 A/D 转换立刻中止，然后重新开始转换。

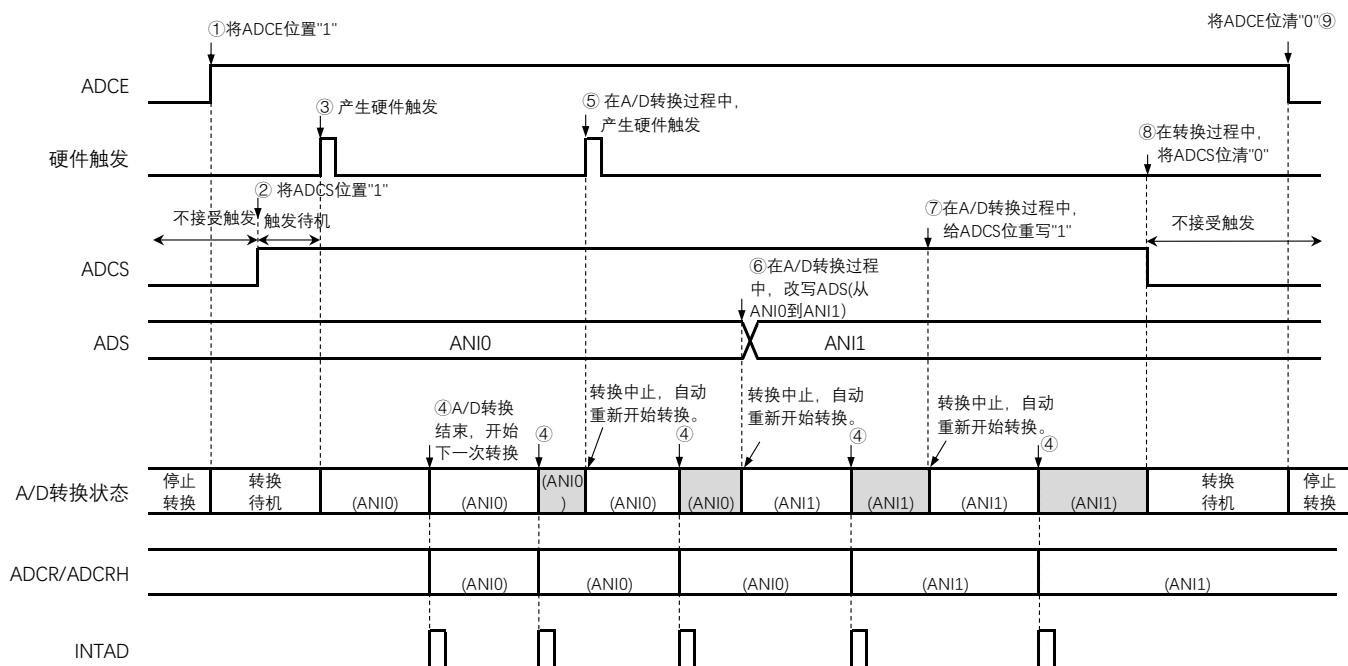
如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后对 ADS 寄存器重新指定的模拟输入进行 A/D 转换。

如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入 A/D 转换待机状态。但是，在此状态下 A/D 转换器不进入停止状态。

如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCS 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

图 17-10：硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



17.5.6 硬件触发无等待模式（选择模式、单次转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。

在通过软件对稳定等待时间（1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将 ADCS 位置“1”也不开始 A/D 转换。

如果在 ADCS 位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。

如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。

在 A/D 转换结束后，ADCS 位保持“1”的状态，进入 A/D 转换待机状态。

如果在转换过程中输入硬件触发，当前的 A/D 转换立刻中止，然后重新开始转换。

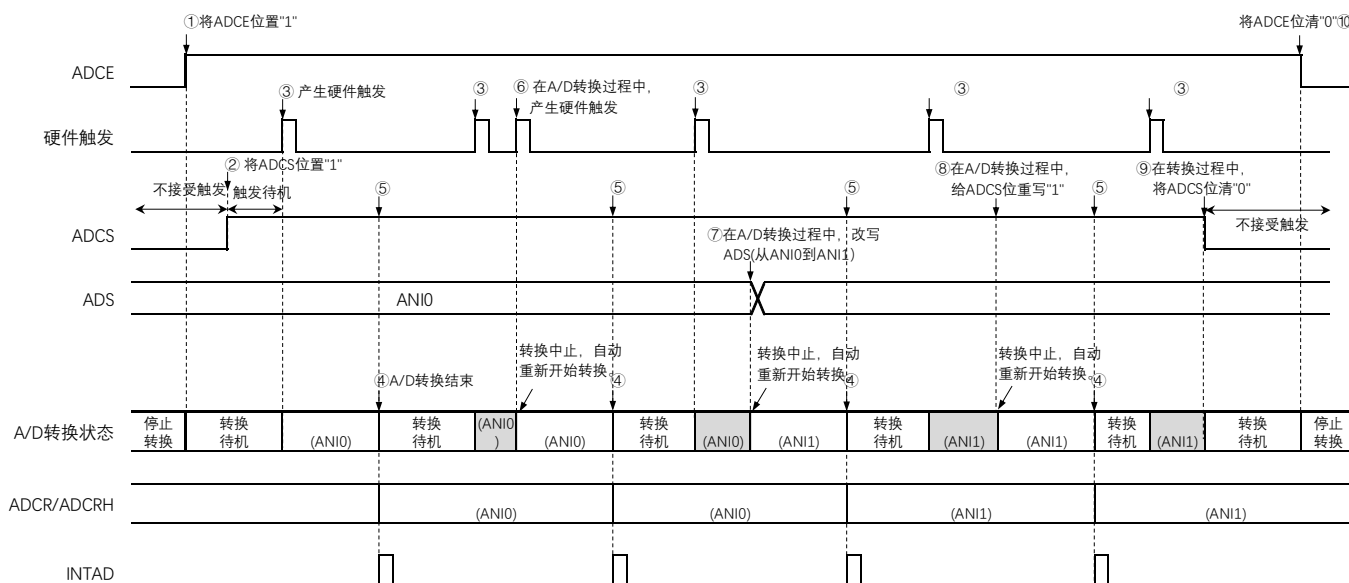
如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后对 ADS 寄存器重新指定的模拟输入进行 A/D 转换。

如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻停止，然后进入 A/D 转换待机状态。但是，在此状态下 A/D 转换器不进入停止状态。

如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCS 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

图 17-11 硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



17.5.9 硬件触发等待模式（选择模式、连续转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。

如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。

如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换（此时，不需要硬件触发）。

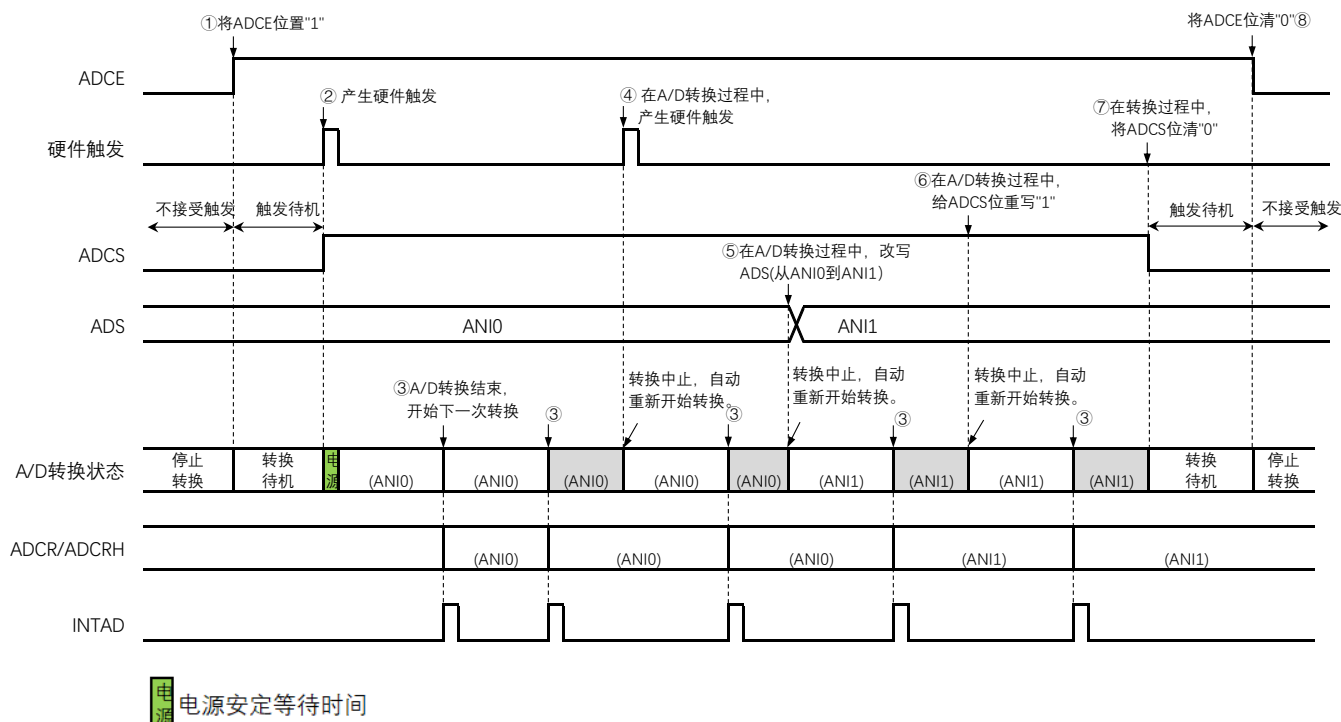
如果在转换过程中输入硬件触发，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后对 ADS 寄存器重新指定的模拟输入进行 A/D 转换。

如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

图 17-14 硬件触发等待模式（选择模式、连续转换模式）的运行时序例子



17.5.10 硬件触发等待模式（选择模式、单次转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。

如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。

如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。

在 A/D 转换结束后，ADCS 位自动清“0”，A/D 转换器进入停止状态。

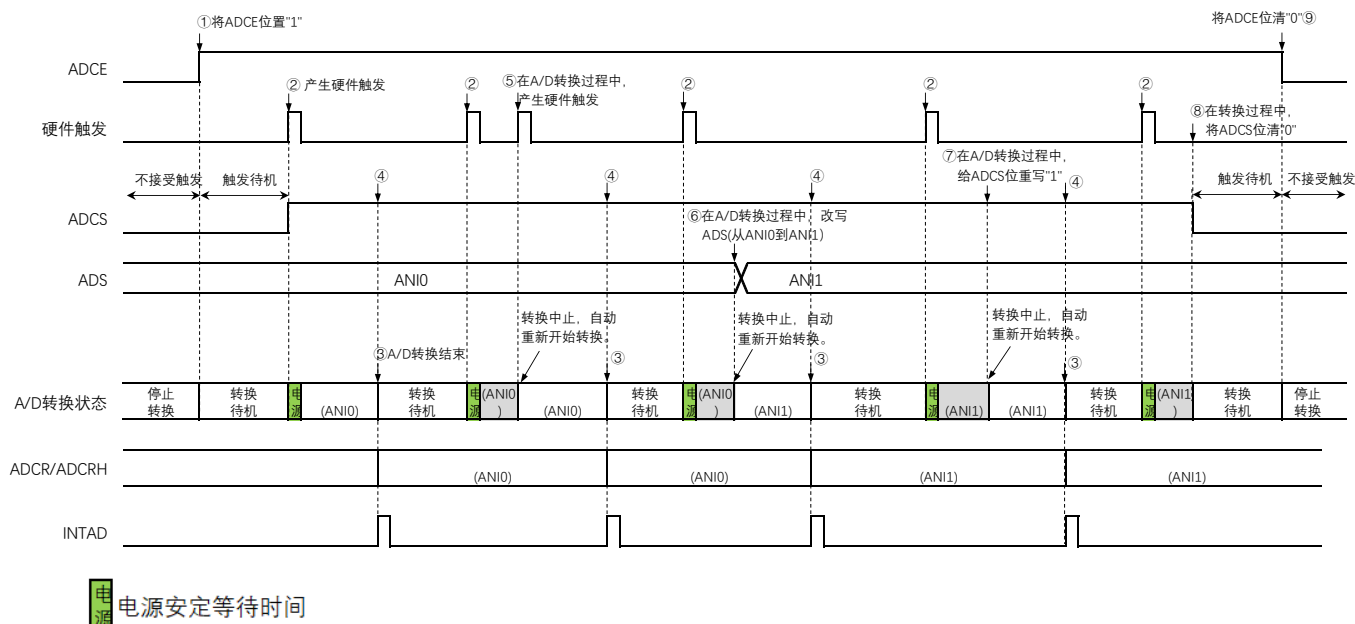
如果在转换过程中输入硬件触发，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后对 ADS 寄存器重新指定的模拟输入进行 A/D 转换。

如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后重新开始转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开 A/D 转换。

图 17-15：硬件触发等待模式（选择模式、单次转换模式）的运行时序例子



17.5.11 硬件触发等待模式（扫描模式、连续转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。

如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描 0~扫描 3 的 4 个模拟输入通道进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。

连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 4 个通道的 A/D 转换结束后立即从所设通道自动开始下一次 A/D 转换。

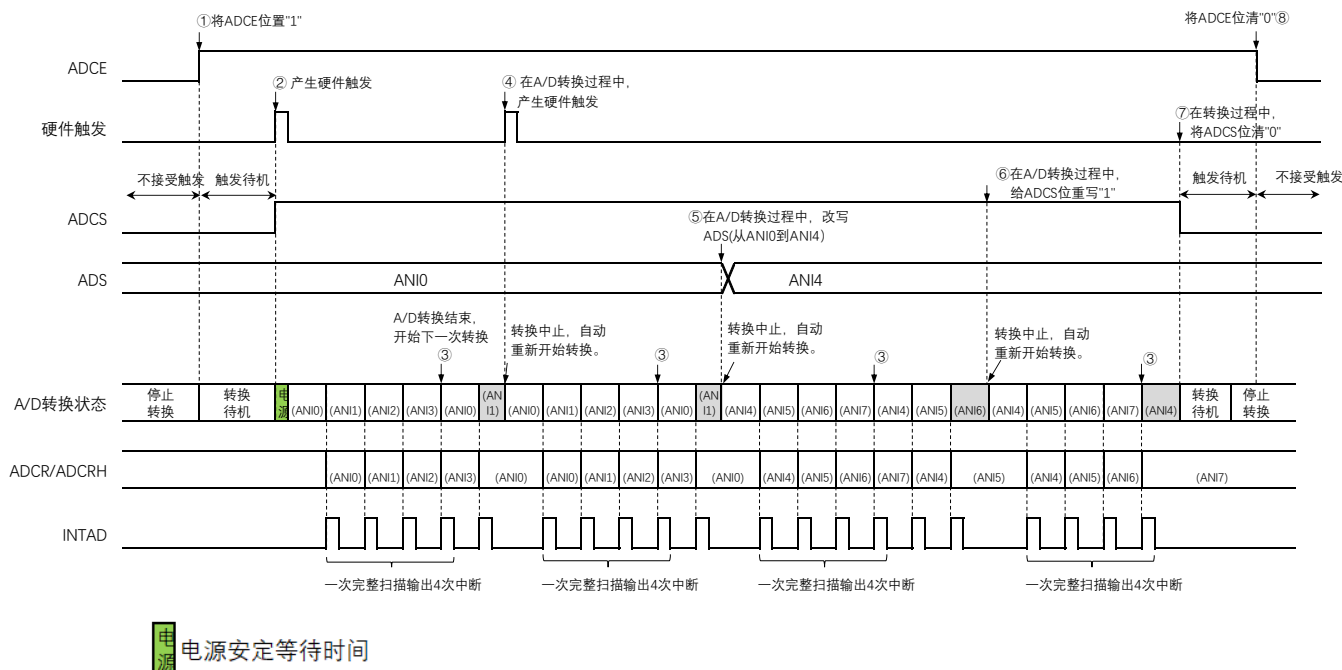
如果在转换过程中输入硬件触发，当前的 A/D 转换立刻中止，然后从最初的通道重新开始转换。

如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后从由 ADS 寄存器重新指定的通道开始进行扫描转换。

如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后从最初的通道重新开始转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

图 17-16: 硬件触发等待模式（扫描模式、连续转换模式）的运行时序例子



17.5.12 硬件触发等待模式（扫描模式、单次转换模式）

在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。

如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描 0~扫描 3 的 4 个模拟输入通道进行 A/D 转换。在输入硬件触发后自动将 ADM0 寄存器的 ADCS 位置“1”。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。

连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。

在 A/D 转换结束后，ADCS 位自动清“0”，A/D 转换器进入停止状态。

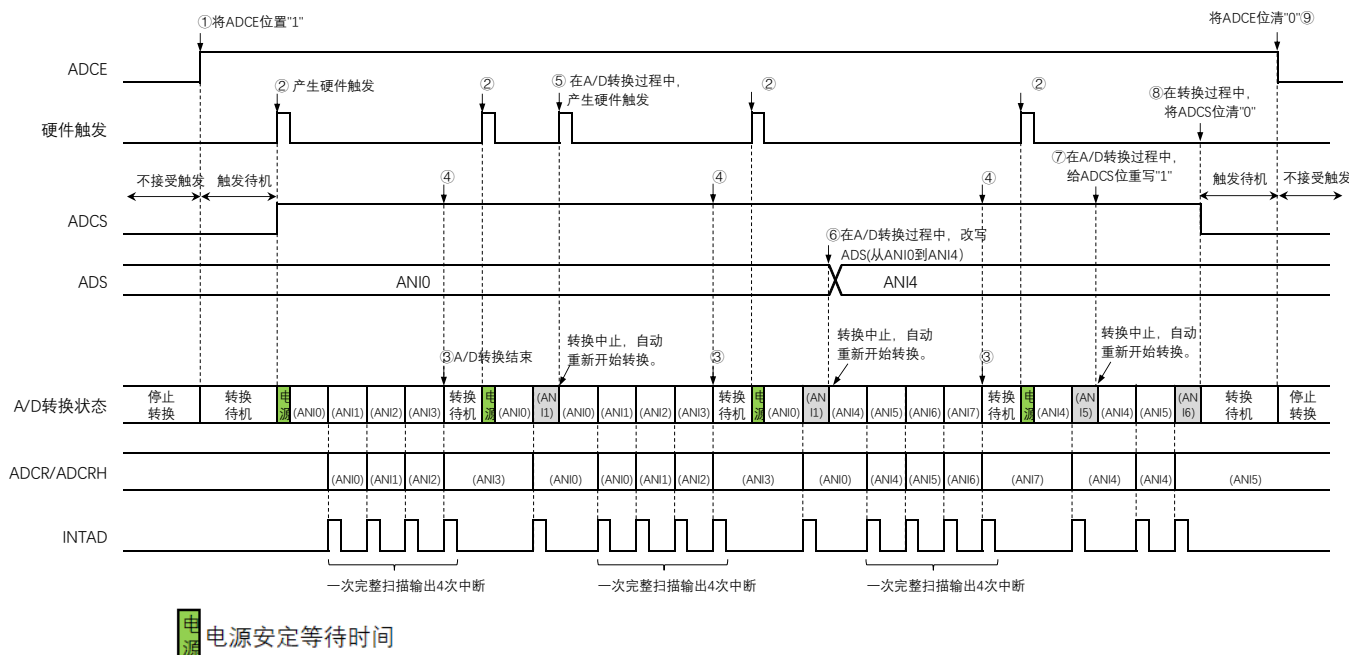
如果在转换过程中输入硬件触发，当前的 A/D 转换立刻中止，然后重新从最初的通道开始扫描转换。

如果在转换过程中改写或者重写 ADS 寄存器，当前的 A/D 转换立刻中止，然后从由 ADS 寄存器重新指定的通道开始扫描转换。

如果在转换过程中给 ADCS 位重写“1”，当前的 A/D 转换立刻中止，然后从最初的通道开始扫描转换。

如果在转换过程中将 ADCS 位置“0”，当前的 A/D 转换立刻中止，然后进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

图 17-17：硬件触发等待模式（扫描模式、单次转换模式）的运行时序例子

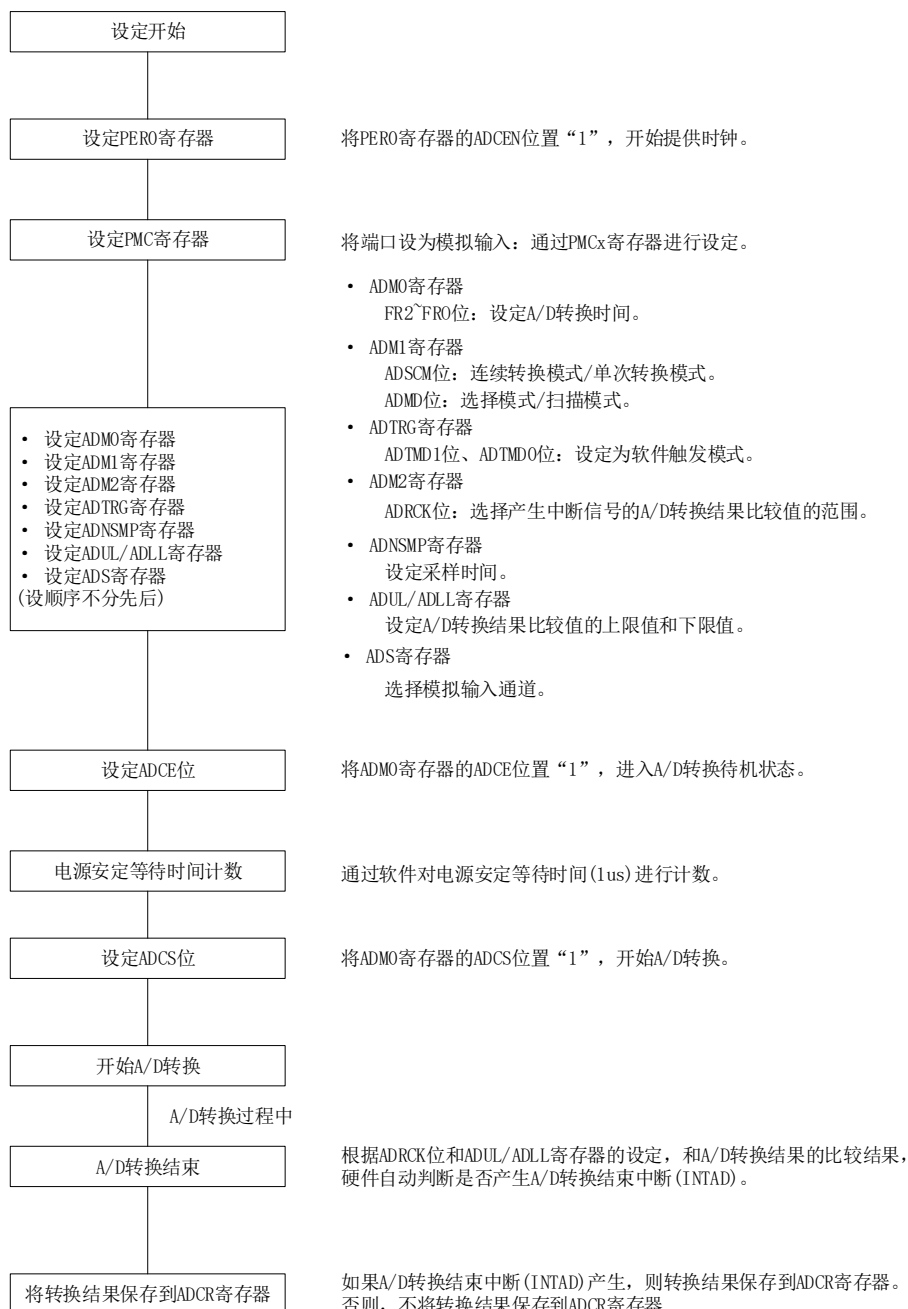


17.6 A/D 转换器的设置流程图

各运行模式的 A/D 转换器的设置流程图如下所示。

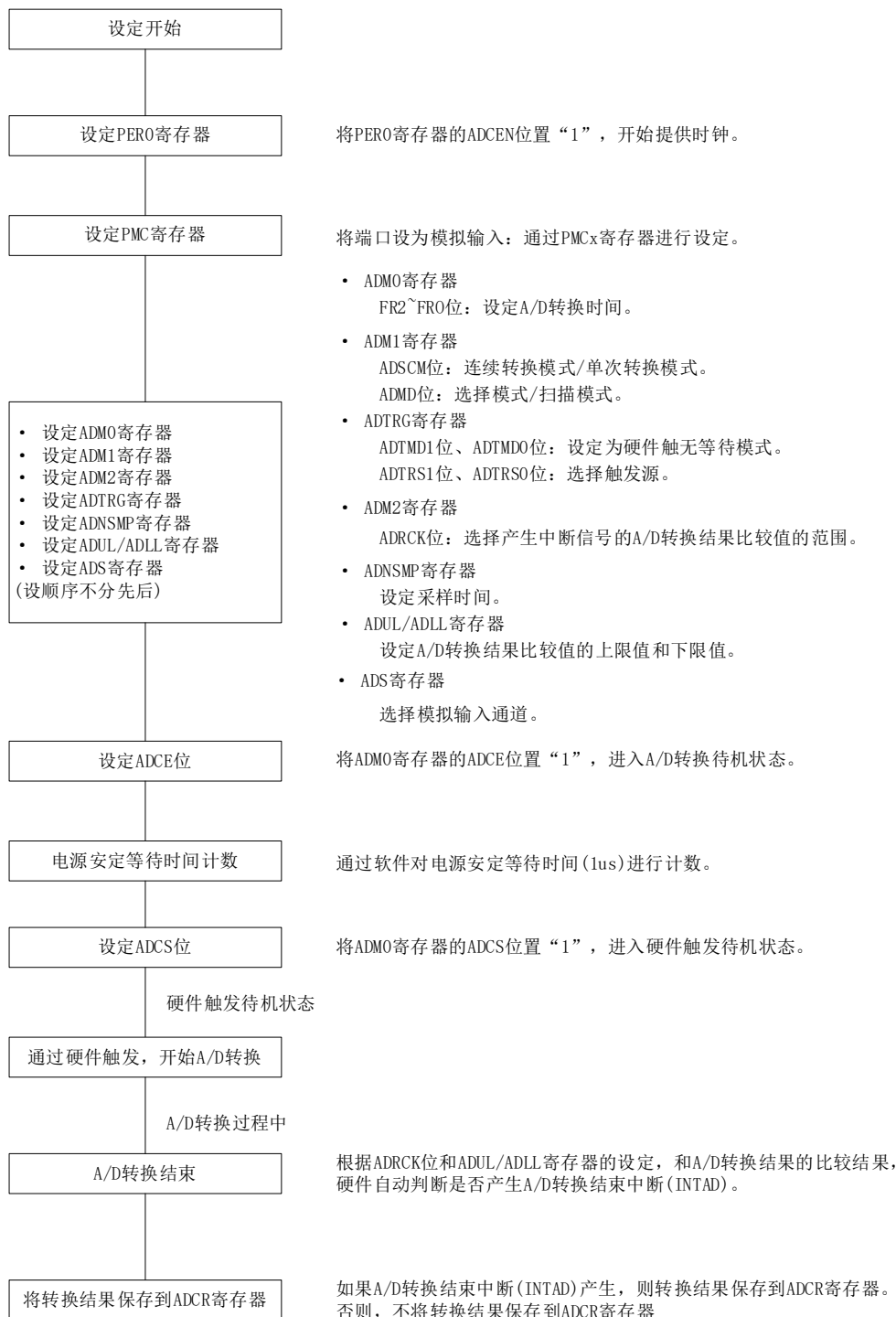
17.6.1 软件触发模式的设置

图 17-18：软件触发模式的设置



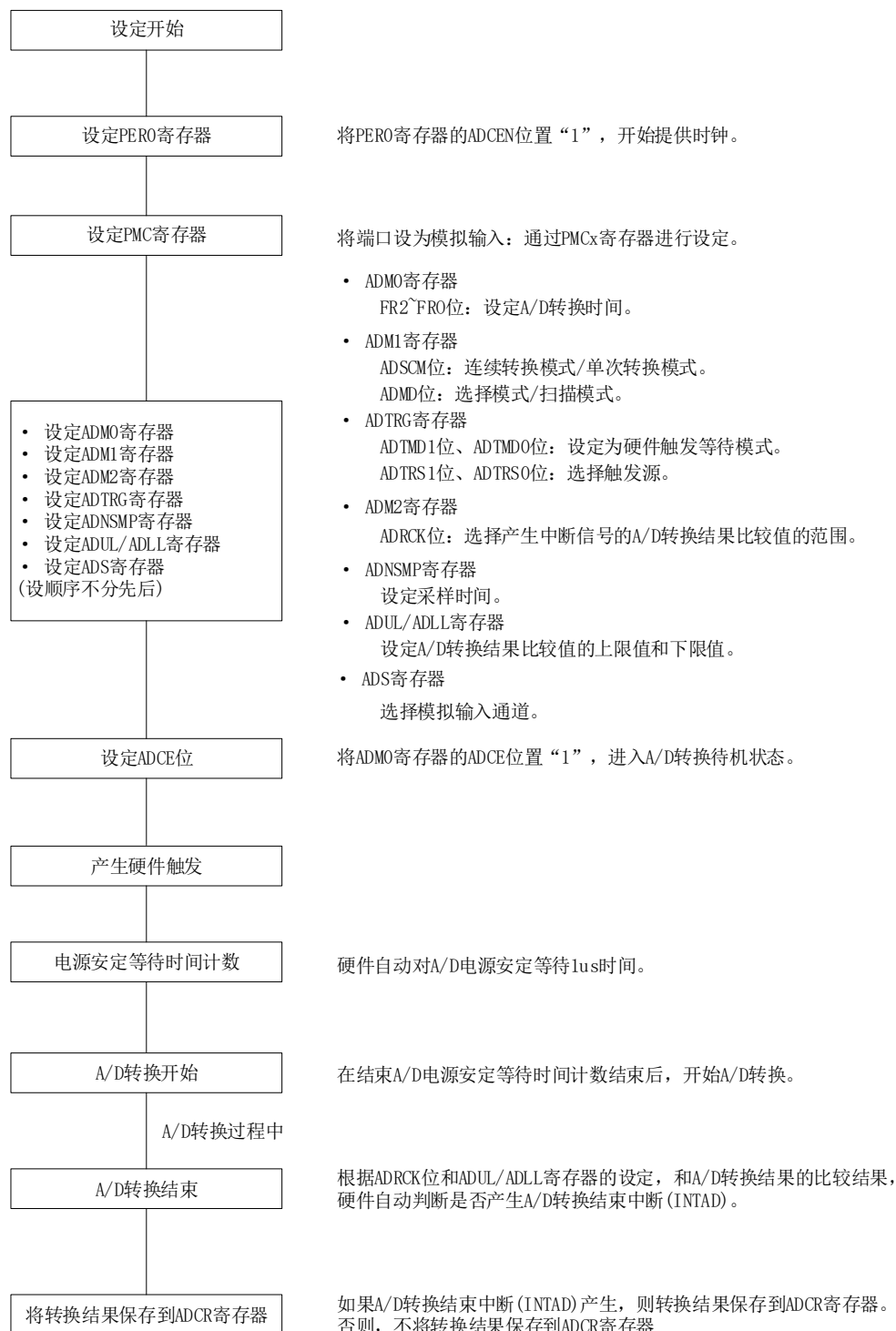
17.6.2 硬件触发无等待模式的设置

图 17-19：硬件触发无等待模式的设置



17.6.3 硬件触发等待模式的设置

图 17-20：硬件触发等待模式的设置



第18章 模拟比较器 (ACMP0/1)

18.1 概述

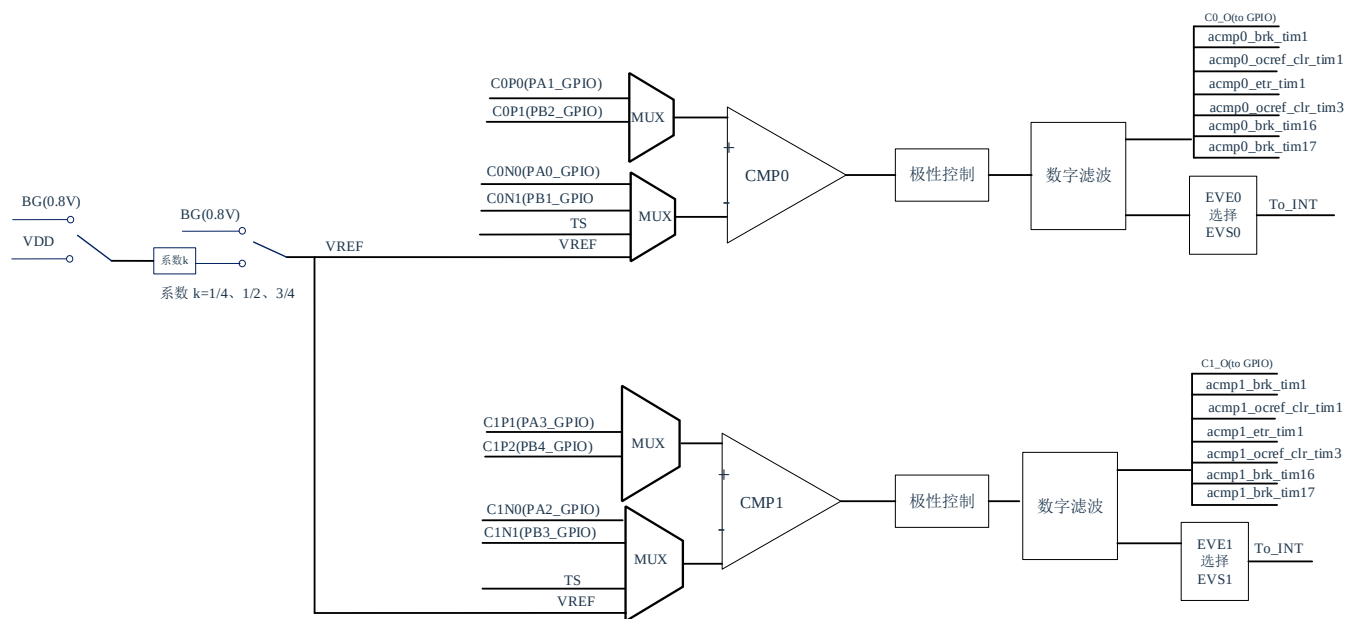
芯片内部包含两个模拟比较器。可按照比较器的配置适用于不同的应用场合。当正端电压大于负端电压时，比较器输出逻辑 1，反之输出 0，也可以通过输出极性选择位进行改变。当比较器输出值发生改变时，每路比较器都可通过配置产生中断。

18.2 特性

- 模拟输入电压范围：(0~VDD)V。
- 支持迟滞电压选择(0mV/20mV)。
- 每个比较器正、负端可多路选择
- 输出可滤波时间可选择：0~512* T_{sys} 。
- 比较器输出可作为定时器 1 的触发信号。
- 输出改变可产生中断。

18.3 结构框图

图 18-1: 比较器结构框图



注：若设置滤波使能，会产生以下几种情况：

滤波输入信号宽度<滤波设置时间：无信号输出。

滤波输入信号宽度=滤波设置时间：输出一个时钟周期的脉冲信号。

滤波输入信号宽度>滤波设置时间：正常滤波输出。

18.4 寄存器映射

(ACMP 基地址 = 0x40068200) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
C0CON0(P1B)	0x000	R/W	模拟比较器 0 控制寄存器 0	0x00
C0CON1(P1B)	0x004	R/W	模拟比较器 0 控制寄存器 1	0x00
C1CON0(P1B)	0x008	R/W	模拟比较器 1 控制寄存器 0	0x00
C1CON1(P1B)	0x00C	R/W	模拟比较器 1 控制寄存器 1	0x00
CEVCON(P1B)	0x010	R/W	模拟比较器事件控制寄存器	0x00
CVRCON(P1B)	0x014	R/W	模拟比较器电压参考控制寄存器	0x00
ACMPLOCK	0x018	R/W	模拟比较器写使能寄存器	0x00

注:

(P1B)标注的寄存器为被保护的寄存器。

(P1B): ACMPLOCK==55H 时, 标注的寄存器允许写入; =其他值, 禁止写入。

18.5 寄存器说明

18.5.1 模拟比较器 0 控制寄存器 0(C0CON0)

位	符号	描述	复位值
31:16	-	保留	0x0
15	C0_EN	模拟比较器 0 使能位 0: 禁止 1: 使能	0
14	C0_OEN	模拟比较器 0 输出使能位 0: 禁止 1: 使能	0
13:9	-	保留	0x0
8	C0_OUT	模拟比较器 0 结果位(只读)	0
7:6	-	保留	0
5:4	C0_P_S	模拟比较器 0 正端通道选择 00: C0P0(PA1) 01: C0P1(PB2) 10: 禁止设置 11: 禁止设置	0x0
3	--	保留	0
2:0	C0_N_S	模拟比较器 0 负端通道选择 000: C0N0(PA0) 001: C0N1(PB1) 010: 保留 011: TS 100: VREF 其他: 禁止设置	0x0

18.5.2 模拟比较器 0 控制寄存器 1(C0CON1)

位	符号	描述	复位值
31:14	-	保留	0x0
13:12	C0_HYSPN_S	模拟比较器 0 迟滞方式选择 00: 无迟滞 01: 正迟滞 10: 负迟滞 11: 正负迟滞	0x0
11	-	保留	0
10	C0_HYSV_S	模拟比较器 0 迟滞选择 0: 无迟滞 1: 20mV	0
9	C0_POS	模拟比较器 0 输出极性选择位 0: 正常输出 1: 反相输出	0
8	C0_FE	模拟比较器 0 输出滤波使能位 0: 禁止	0

		1: 使能	
7:4	-	保留	0x0
3:0	C0_FS	模拟比较器 0 输出滤波时间选择位 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk 其他: (0~1)*Tpclk	0x0

18.5.3 模拟比较器 1 控制寄存器 0(C1CON0)

位	符号	描述	复位值
31:16	-	保留	-
15	C1_EN	模拟比较器 1 使能位 0: 禁止 1: 使能	0
14	C1_OEN	模拟比较器 1 输出使能位 0: 禁止 1: 使能	0
13:9	-	保留	-
8	C1_OUT	模拟比较器 1 结果位(只读)	0
7:6	-	保留	0x0
5:4	C1_P_S	模拟比较器 1 正端通道选择 00: 禁止设置 01: C1P1(PA3) 10: C1P2(PB4) 11: 禁止设置	0x0
3	--	保留	
2:0	C1_N_S	模拟比较器 1 负端通道选择 000: C1N0(PA2) 001: C1N1(PB3) 010: 禁止设置 011: TS 100: VREF 其他: 禁止	0x0

18.5.4 模拟比较器 1 控制寄存器 1(C1CON1)

位	符号	描述	复位值
31:14	-	保留	-
13:12	C1_HYSPN_S	模拟比较器 1 迟滞方式选择 00: 无迟滞 01: 正迟滞 10: 负迟滞 11: 正负迟滞	0x0
11	-	保留	-
10	C1_HYSV_S	模拟比较器 1 迟滞电压选择 0: 无迟滞 1: 20mV	0
9	C1_POS	模拟比较器 1 输出极性选择位 0: 正常输出 1: 反相输出	0
8	C1_FE	模拟比较器 1 输出滤波使能位 0: 禁止 1: 使能	0
7:4	-	保留	-
3:0	C1_FS	模拟比较器 1 输出滤波时间选择位 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk 其他: (0~1)*Tpclk	0x0

18.5.5 模拟比较器事件控制寄存器(CEVCON)

位	符号	描述	复位值
31:18	-	保留	0
17	ACMP1_OCREFCLR INTEN_TIM3	模拟比较器 1 输出作 TIM3 的 OCREF_CLR_INT 使能控制位 0: 禁止 1: 使能	0
16	ACMP0_OCREFCLR INTEN_TIM3	模拟比较器 0 输出作 TIM3 的 OCREF_CLR_INT 使能控制位 0: 禁止 1: 使能	0
15	ACMP1_OCREFCLR INTEN_TIM1	模拟比较器 1 输出作 TIM1 的 OCREF_CLR_INT 使能控制位 0: 禁止 1: 使能	0
14	ACMP0_OCREFCLR INTEN_TIM1	模拟比较器 0 输出作 TIM1 的 OCREF_CLR_INT 使能控制位 0: 禁止 1: 使能	0
13:12	ETR_SRC_TIM1	TIM1 的 ETR 输入源选择 00: ETR 来源于 GPIO 01: ETR 来源于 COMP0 10: ETR 来源于 COMP1 11: 禁止	0
11	ACMP1_BRKEN_TIM17	模拟比较器 1 输出作 TIM17 的 break 使能控制位 0: 禁止 1: 使能	0
10	ACMP0_BRKEN_TIM17	模拟比较器 0 输出作 TIM17 的 break 使能控制位 0: 禁止 1: 使能	0
9	ACMP1_BRKEN_TIM16	模拟比较器 1 输出作 TIM16 的 break 使能控制位 0: 禁止 1: 使能	0
8	ACMP0_BRKEN_TIM16	模拟比较器 0 输出作 TIM16 的 break 使能控制位 0: 禁止 1: 使能	0
7	ACMP1_BRKEN_TIM1	模拟比较器 1 输出作 TIM1 的 break 使能控制位 0: 禁止 1: 使能	0
6	ACMP0_BRKEN_TIM1	模拟比较器 0 输出作 TIM1 的 break 使能控制位 0: 禁止 1: 使能	0
5:4	-	保留	0x0
3:2	EVS1	模拟比较器 1 事件产生条件选择位 00: 比较器 1 输出从 0->1 的跳变 01: 比较器 1 输出从 1->0 的跳变 10: 比较器 1 输出从 0->1 的跳变或从 1->0 的	0x0

		跳变 11: 保留	
1:0	EVS0	模拟比较器 0 事件产生条件选择位 00: 比较器 0 输出从 0->1 的跳变 01: 比较器 0 输出从 1->0 的跳变 10: 比较器 0 输出从 0->1 的跳变或从 1->0 的跳变 11: 保留	0x0

18.5.6 模拟比较器参考电压控制寄存器(CVRCON)

位	符号	描述	复位值
31:6	-	保留	0x0
5	CNDIVS	ACMP_VREF 参考源选择位 1: 选择 BG(0.8V)进行分压 0: 选择 VDD 进行分压	0
4	CNSVR	比较器负端内部电压 VREF 选择位 1: 选择 ACMP_VREF 0: 选择 BG(0.8V)	0
3:2	-	保留	0x0
1:0	CVS	模拟比较器参考电压分压系数 k 选择位 00: 1/4 01: 2/4 10: 3/4 11: 保留	0x0

18.5.7 模拟比较器写使能控制寄存器(ACMPLOCK)

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	ACMPLOCK[7:0]	ACMP 寄存器写使能控制; 0x55: 寄存器写使能; 其他: 写寄存器无效。	0x0

第19章 通用 CRC

19.1 CRC 概述

CRC（循环冗余校验）计算单元使用一个固定的多项式发生器从一个 32 位的数据字中产生 CRC 码。

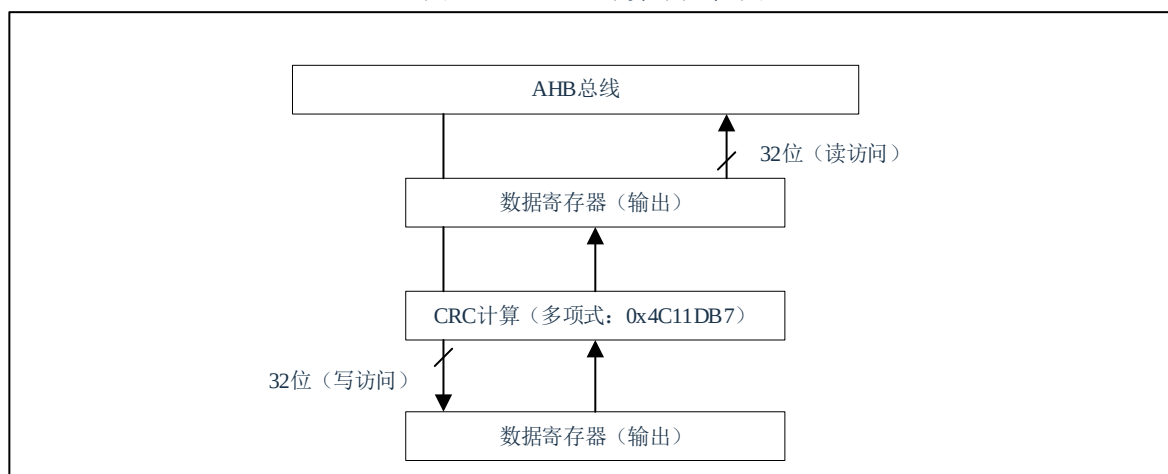
在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据 EN/IEC 60335-1 标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

19.2 CRC 特性

- 使用 CRC-32（以太网）多项式：0x4C11DB7
 - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 单输入/输出 32 位数据寄存器
- CRC 计算在 4 个 AHB 时钟周期 (HCLK) 内完成
- 8 位通用寄存器 (可用于临时存储)

框图如图 19-1 所示。

图 19-1: CRC 计算单元框图



19.3 功能说明

CRC 计算单元主要由单个 32 位数据寄存器组成，该寄存器：

- 用作输入寄存器，向 CRC 计算器中输入新数据（向寄存器写入数据时）
- 可保存之前的 CRC 计算结果（读取寄存器时）

对数据寄存器的每个写操作都会把当前新输入的数值和之前生成在数据寄存器中的 CRC 值再做一次 CRC 计算（CRC 计算针对整个 32 位数据字完成，而非逐字节进行）。

CRC 计算的时候，写操作被阻塞，因此允许执行背靠背写访问或连续的写读访问。

使用 CRC_CR 寄存器中的 RESET 控制位即可将 CRC 计算器复位为 0xFFFF FFFF。此操作不影响 CRC_IDR 寄存器的内容

19.4 寄存器映射

(CRC 基地址 = 0x4000_3000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CRC_DR	0x000	R/W	CRC 数据寄存器	0xFFFFFFFF
CRC_IDR	0x004	R/W	CRC 独立数据寄存器	0x0
CRC_CR	0x008	R/W	CRC 控制寄存器	0x0

19.5 寄存器说明

19.5.1 CRC 数据寄存器(CRC_DR)

位	符号	描述	复位值
31:0	DR	数据寄存器位。 向 CRC 计算器写入新数据时用作输入寄存器。 读取寄存器时可读出之前的 CRC 计算结果	0xFFFFFFFF

19.5.2 CRC 独立数据寄存器(CRC_IDR)

位	符号	描述	复位值
31:8	--	保留, 必须保持复位值	0x0
7:0	IDR[7:0]	通用的 8 位数据寄存器位。 可用作一个字节的临时存储单元。 此寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响。	0x0

19.5.3 CRC 控制寄存器(CR)

位	符号	描述	复位值
31:1	--	保留, 必须保持复位值	0x0
0	RESET	该位被软件置位, 用来复位 CRC 计算单元。该位只能被置位, 由硬件自动清零。	0x0

第20章 待机功能

20.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下 2 种模式。

1. 睡眠模式

睡眠模式是停止 CPU 运行时钟的模式。在设定睡眠模式前，如果高速内部振荡器或者低速内部振荡器正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到深度睡眠模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

2. 深度睡眠模式

深度睡眠模式是停止高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小 CPU 的工作电流。

因为深度睡眠模式能通过中断请求来解除，所以也能进行间歇运行。但是，因为在解除深度睡眠模式时需要确保振荡稳定的等待时间，所以如果需要通过中断请求立即开始处理，就必须选择睡眠模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设定为待机模式前的内容，并且还保持输入/输出端口的输出锁存器和输出缓冲器的状态。

注 1：在转移到深度睡眠模式时，必须在停止以系统时钟运行的外围硬件后执行 WFI 指令。

注 2：为了减小 A/D 转换器的工作电流，必须将 A/D 转换器的，模式寄存器 0（ADM0）bit7（ADCS）（ADCS 转换完成硬件清 0）清“0”后，再停止 A/D 转换运行后执行 WFI 指令。

20.2 睡眠模式

20.2.1 睡眠模式的设定

在 SCR 寄存器的 SLEEPDEEP 位为 0 时，执行 WFI 指令，就进入了睡眠模式。在睡眠模式，CPU 停止动作，但是内部寄存器的值仍被保持，周边模块也保持进入睡眠模式之前的状态。周边模块，振荡器等在睡眠模式下的状态见表 20-1。

无论设定前的 CPU 时钟是高速内部振荡器时钟或者低速内部振荡器时钟，都能设定睡眠模式。

注意：中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除睡眠模式。因此，即使在此情况下执行 WFI 指令，也不转移到睡眠模式。

表 20-1：睡眠模式中的运行状态

项目		睡眠模式
系统时钟	F _{IH}	继续运行
	F _{IL}	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态，保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM		停止运行
端口（锁存器）		保持休眠之前的设定
15 位间隔定时器		能运行
TIM1		能运行
TIM3		能运行
TIM14		能运行
TIM16/17		能运行
窗口看门狗定时器		能运行
独立看门狗定时器		能运行
SPI 通信单元		能运行
I2C 串行接口控制器		能运行
USART		能运行
数据传送控制器（DMA）		能运行
A/D 转换器		能运行
模拟比较器(ACMP0/1)		能运行
上电复位功能		能运行
电压检测功能		能运行
外部中断		能运行
CRC 运算	高速 CRC	能运行
	通用 CRC	停止运行
SFR 保护功能		停止运行

注 1：停止运行：在转移到睡眠模式时停止运行。

注 2：F_{IH}：高速内部振荡器时钟。

注 3：F_{IL}：低速内部振荡器时钟。

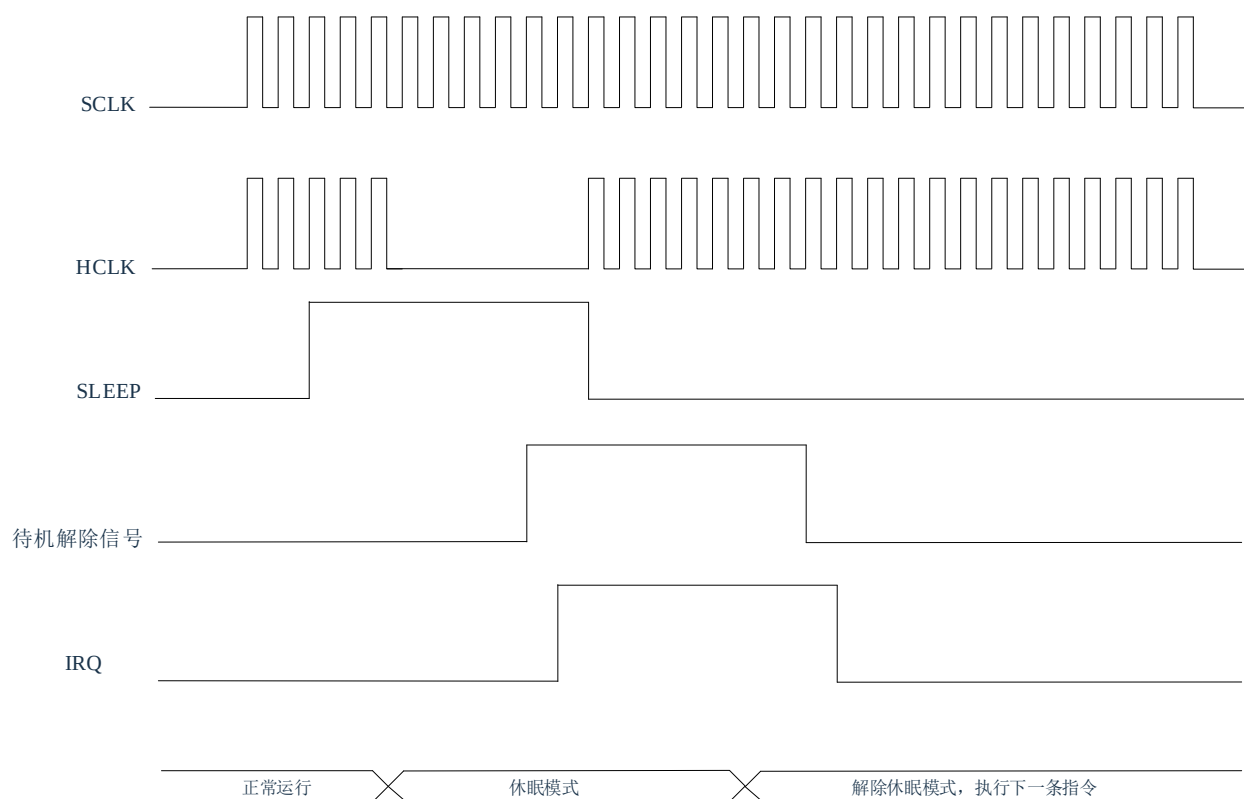
20.2.2 睡眠模式的解除

能通过以下 2 种方法解除睡眠模式。

1. 通过中断解除

当产生一个中断请求，且处于允许接受中断的状态时，睡眠模式就被解除，CPU 开始处理中断服务程序。

图 20-1 通过中断请求解除睡眠模式



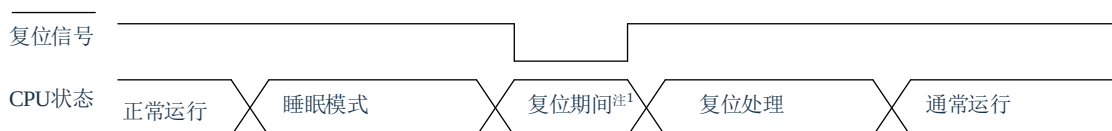
注 1：从待机解除信号产生到睡眠模式解除，开始执行中断服务程序，需要 16 个时钟。

注 2：部分待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

2. 通过复位解除

当有复位信号产生时，CPU 处于复位状态，睡眠模式被解除。和通常的复位相同，在转移到复位向量地址后执行程序。

图 20-2：通过复位解除睡眠模式



注 1：有关复位处理，请参照“第 22 章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理，请参照“第 23 章 上电复位电路”。

20.3 深度睡眠模式

20.3.1 深度睡眠模式的设定

在 SCR 寄存器的 SLEEPDEEP 位为 1 时，执行 WFI 指令，就进入了深度睡眠模式。在这个模式，CPU，大多数的周边模块，以及振荡器都停止运行。但是，CPU 内部寄存器的值，RAM 数据，周边模块，I/O 的状态被保持。周边模块，振荡器在深度睡眠模式的运行状态见表 20-2。

注意:当中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除深度睡眠模式。因此，如果在此情况下执行 WFI 指令，就在一旦进入深度睡眠模式后立即被解除。在执行 WFI 指令并且经过深度睡眠模式解除时间后返回到运行模式。

表 20-2: 深度睡眠模式中的运行状态

项目		深度睡眠模式
系统时钟	F _{IH}	停止运行
	F _{IL}	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态，保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM		停止运行
端口（锁存器）		保持休眠之前的设定
15 位间隔定时器		能运行
TIM1		禁止运行
TIM3		禁止运行
TIM14		禁止运行
TIM16/17		禁止运行
窗口看门狗定时器		禁止运行
独立看门狗定时器		能运行
SPI 通信单元		禁止运行
I2C 串行接口控制器		禁止运行
USART		禁止运行
数据传送控制器（DMA）		禁止运行
A/D 转换器		禁止运行
模拟比较器(ACMP0/1)		能运行
上电复位功能		能运行
电压检测功能		能运行
外部中断		能运行
CRC 运算	高速 CRC	停止运行
	通用 CRC	停止运行
SFR 保护功能		停止运行

注 1：停止运行：在转移到深度睡眠模式时停止运行。

禁止运行：在转移到深度睡眠模式前停止运行。

注 2：F_{IH}：高速内部振荡器时钟。

注 3：F_{IL}：低速内部振荡器时钟。

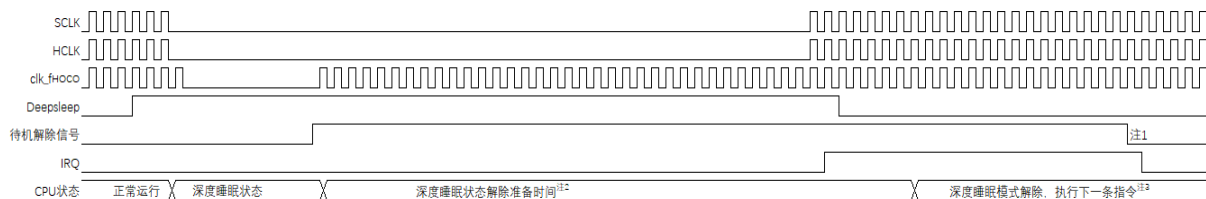
20.3.2 深度睡眠模式的解除

能通过以下 2 种方法解除深度睡眠模式。

1. 通过未屏蔽的中断请求进行的解除

如果发生 LVD 检测、INTP0-12, INTP13-14、15 位间隔定时器定时器或者 ACMP 中断请求, 就解除深度睡眠模式。在经过振荡稳定时间后, 如果处于允许接受中断的状态, 就进行向量中断的处理。如果处于禁止接受中断的状态, 就执行下一个地址的指令。

图 20-3 : 通过中断请求解除深度睡眠模式



注 1: 待机解除信号: 有关待机解除信号的详细内容, 请参照中断相关内容。

注 2: 深度睡眠状态解除准备时间: 进入深度睡眠模式前 CPU 时钟为高速内部振荡时时: 至少 100us。

注 3: 等待: 从 CPU.IRQ 有效到开始执行中断服务程序, 需要 16 个时钟。

备注: 高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

2. 通过产生复位信号进行的解除

通过产生复位信号来解除深度睡眠模式。然后, 和通常的复位相同, 在转移到复位向量地址后执行程序。

图 20-4: 通过复位解除深度睡眠模式



注 1: 有关复位处理, 请参照“第 22 章 复位功能”。有关上电复位 (POR) 电路和电压检测 (LVD) 电路的复位处理, 请参照“第 23 章 上电复位电路”。

第21章 嵌套向量中断控制器(NVIC)

Cortex®-M0+ CPU 提供一个用于中断处理的嵌套向量中断控制器（NVIC）。

21.1 特性

- 支持嵌套向量中断。
- 自动保存和恢复处理器状态。
- 动态改变优先级。
- 简化和确定的中断时间。

NVIC 按照优先级处理所有支持的异常。所有的异常在“Handler 模式”处理。NVIC 支持 32 个(IRQ[31:0])离散中断，每个中断支持 4 级中断优先级。所有的中断和大部分系统异常可以配置成不同的优先级。当一个中断发生时，NVIC 将比较新中断与当前中断的优先级，如果新中断优先级高，则立即处理新中断。

当接受一个中断后，中断服务程序（ISR）的开始地址可从内存中的向量表取得。软件不需要决定哪个中断被响应，也不用分配相关 ISR 的开始地址。当取得开始地址后，NVIC 将自动保存处理器状态寄存器（PC、PSR、LR、R0~R3、R12）的值到栈中。在 ISR 结束后，NVIC 将从栈中恢复相关寄存器的值，并运行在正常状态。因此花费少量且确定的时间处理中断请求。

NVIC 支持“末尾连锁”，可以有效的处理背对背中断，即无需保存和恢复当前状态，从而减少结束当前 ISR 切换到挂起 ISR 的延迟时间。NVIC 还支持“Late Arrival”，因此可以提高并发中断的效率。当较高优先级中断请求发生在当前 ISR 开始执行之前（保存处理器状态和获取起始地址阶段），NVIC 将立即处理更高优先级的中断，从而提高实时性。

更多详细信息，请参考“ARM®Cortex®-M0+ 技术参考手册”和“ARM®v6-M 架构参考手册”。

21.2 异常模式和系统中断映射

下表列出了该系列产品支持的异常模式。与所有中断一样，软件可以为其中一些异常设置 4 级优先级。用户可配置的最高优先级为 0，最低优先级为 3。所有用户可配置中断的默认优先级为 0。

异常名称	异常编号	优先级
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4~10	保留
SVCall	11	可配置
Reserved	12~13	保留
PendSV	14	可配置
SysTick	15	可配置
Interrupt (IRQ0~IRQ31)	16~47	可配置

注：优先级 0 在系统为第 4 优先级，排在“Reset”、“NMI”和“Hard Fault”三个系统异常之后。

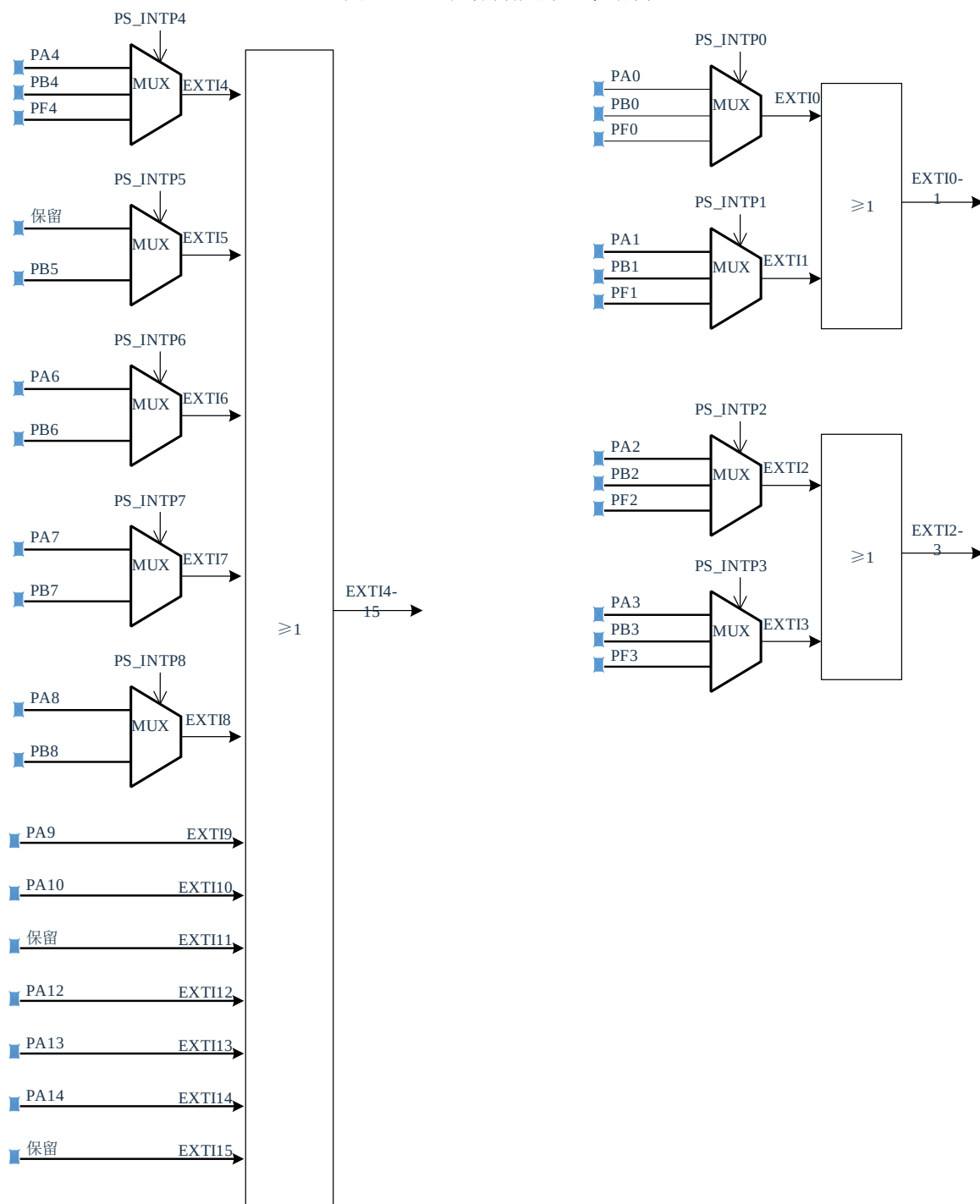
21.3 向量表

表 21-1 中断源一览表

编号	中断号	向量地址	异常类型	描述
1-15	-	0x00-0x3c	系统异常	
16	0	0x40	WWDG	窗口定时器中断
17	1	0x44	PVD	电源电压检测中断
18	2	0x48	保留	-
19	3	0x4c	FLASH	FLASH 编程完成/操作错误
20	4	0x50	保留	-
21	5	0x54	EXT0-1	外部中断 1_0 中断
22	6	0x58	EXT2-3	外部中断 3_2 中断
23	7	0x5c	EXT4-15	外部中断 15_4 中断
24	8	0x60	保留	-
25	9	0x64	DMA_0	DMA 通道 0 中断
26	10	0x68	DMA_1	DMA 通道 1 中断
27	11	0x6c	DMA_3	DMA 通道 3 中断
28	12	0x70	ADC	AD 转换结束中断
29	13	0x74	TIM1_BRK_UP_TRIG_COM	TIM1 断开、更新、触发和通信中断
30	14	0x78	TIM1_CC	TIM1 捕获/比较中断
31	15	0x7c	保留	-
32	16	0x80	TIM3	TIM3 全局中断
33	17	0x84	LPTIM	低功耗定时器中断
34	18	0x88	保留	-
35	19	0x8c	TIM14	TIM14 全局中断
36	20	0x90	保留	-
37	21	0x94	TIM16	TIM16 全局中断
38	22	0x98	TIM17	TIM17 全局中断
39	23	0x9c	I2C	I2C 全局中断
40	24	0xa0	保留	-
41	25	0xa4	SPI1	SPI1 全局中断
42	26	0xa8	SAU	SAU 全局中断
43	27	0xac	USART	USART 全局中断
44	28	0xb0	保留	-
45	29	0xb4	保留	-
46	30	0xb8	ACMP0	ACMP0 全局中断
47	31	0xbc	ACMP1	ACMP1 全局中断

21.4 外部中断结构

图 21-1 中断功能的基本结构



21.5 寄存器映射

(以下寄存器基地址 = 0x4004_5B30) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
EGP0	0x000	R/W	外部中断上升沿允许寄存器	0x0
EGN0	0x001	R/W	外部中断下降沿允许寄存器	0x0
EGP1	0x002	R/W	外部中断上升沿允许寄存器	0x0
EGN1	0x003	R/W	外部中断下降沿允许寄存器	0x0
FILCON	0x00A	R/W	外部中断滤波控制寄存器	0x0
E_IMR0	0x010	R/W	外部中断屏蔽寄存器0	0x0
E_PR0	0x011	R/W	外部中断挂起寄存器0	0x0
E_IMR1	0x012	R/W	外部中断屏蔽寄存器1	0x0
E_PR1	0x013	R/W	外部中断挂起寄存器1	0x0

21.6 寄存器说明

21.6.1 外部中断上升沿允许寄存器 (EGP0)

EGP0 和 EGN0 寄存器用于设定 INTP0~INTP7 的有效边沿。通过 8 位存储器操作指令设定 EGP0、EGN0 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

位	符号	描述	复位值
7	EGP7	INTP7 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
6	EGP6	INTP6 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
5	EGP5	INTP5 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
4	EGP4	INTP4 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
3	EGP3	INTP3 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
2	EGP2	INTP2 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
1	EGP1	INTP1 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
0	EGP0	INTP0 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0

21.6.2 外部中断上升沿允许寄存器（EGP1）

EGP1 和 EGN1 寄存器用于设定 INTP8~INTP10, INTP12~INTP14 的有效边沿。通过 8 位存储器操作指令设定 EGP1、EGN1 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

位	符号	描述	复位值
7	-	保留	0
6	EGP14	INTP14 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
5	EGP13	INTP13 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
4	EGP12	INTP12 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
3	-	保留	0
2	EGP10	INTP10 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
1	EGP9	INTP9 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
0	EGP8	INTP8 外部中断上升沿允许位 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0

注：1. PCLK 指 APB 总线时钟

21.6.3 外部中断下降沿允许寄存器 (EGN0)

位	符号	描述	复位值
7	EGN7	INTP7 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
6	EGN6	INTP6 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
5	EGN5	INTP5 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
4	EGN4	INTP4 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
3	EGN3	INTP3 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
2	EGN2	INTP2 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
1	EGN1	INTP1 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
0	EGN0	INTP0 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0

21.6.4 外部中断下降沿允许寄存器 (EGN1)

位	符号	描述	复位值
7	-	保留	0
6	EGN14	INTP14 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
5	EGN13	INTP13 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
4	EGN12	INTP12 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
3	-	保留	0
2	EGN10	INTP10 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
1	EGN9	INTP9 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
0	EGN8	INTP8 外部中断下降沿允许位 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0

EGPn	EGNn	INTPn 引脚的有效边沿选择 (n=0~10, 12-14)
0	0	禁止检测边沿。
0	1	下降沿
1	0	上升沿
1	1	上升和下降的双边沿

对应 EGPn 位和 EGNn 位的端口如表 21-3 所示。

表 21-3 对应 EGPn 位和 EGNn 位的中断请求信号

检测允许位		中断请求信号
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP5	EGN5	INTP5
EGP6	EGN6	INTP6
EGP7	EGN7	INTP7
EGP8	EGN8	INTP8
EGP9	EGN9	INTP9

EGP10	EGN10	INTP10
-	--	-
EGP12	EGN12	INTP12
EGP13	EGN13	INTP13
EGP14	EGN14	INTP14
-	--	--

注 1：如果将外部中断功能使用的输入端口切换到输出模式，就可能检测到有效边沿而产生 INTP_n 中断。当切换到输出模式时，必须在禁止检测边沿后（EGP_n、EGN_n=0、0）将端口模式寄存器（PM_{xx}）置“0”。

注 2：有关边沿检测的端口，请参照“2.1 端口功能”。

21.6.5 外部中断屏蔽寄存器 0(E_IMR0)

位	符号	描述	复位值
7	IMR7	INTP7 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
6	IMR6	INTP6 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
5	IMR5	INTP5 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
4	IMR4	INTP4 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
3	IMR3	INTP3 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
2	IMR2	INTP2 中断屏蔽控制: 0: 中断屏蔽 1: 中断未屏蔽	0
1	IMR1	INTP1 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
0	IMR0	INTP0 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0

21.6.6 外部中断屏蔽寄存器 1(E_IMR1)

位	符号	描述	复位值
7	-	保留	0
6	IMR14	INTP14 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
5	IMR13	INTP13 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
4	IMR12	INTP12 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
3	-	保留	0
2	IMR10	INTP10 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
1	IMR9	INTP9 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0
0	IMR8	INTP8 中断屏蔽控制 0: 中断屏蔽 1: 中断未屏蔽	0

21.6.7 外部中断挂起寄存器 0(E_PR0)

位	符号	描述	复位值
7	PR7	INTP7 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
6	PR6	INTP6 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
5	PR5	INTP5 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
4	PR4	INTP4 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
3	PR3	INTP3 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
2	PR2	INTP2 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零： 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
1	PR1	INTP1 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零： 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
0	PR0	INTP0 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0

21.6.8 外部中断挂起寄存器 1(E_PR1)

位	符号	描述	复位值
7	-	保留	0
6	PR14	INTP14 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
5	PR13	INTP13 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
4	PR12	INTP12 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
3	-	保留	0
2	PR10	INTP10 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零： 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
1	PR9	INTP9 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零： 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0
0	PR8	INTP8 中断挂起标志，硬件产生上升沿/下降沿中断时，该位置位。软件写 1 清零 0: 未产生中断请求 1: 产生上升沿/下降沿中断请求	0

21.6.9 外部中断滤波控制寄存器(FILCON)

INTP00~INTP10, INTP12~INTP14 共用一个数字滤波器。

位	符号	描述	复位值
7:4	---	保留，须为 0	0x0
3:2	FILCNT	滤波时钟个数 0x0: 1 个滤波时钟长度 0x1: 2 个滤波时钟长度 0x2: 3 个滤波时钟长度 0x3: 4 个滤波时钟长度	0x0
1:0	FILCLK	滤波时钟 filclk 频率设置 0x0: FILCLK = PCLK 0x1: FILCLK = PCLK/4 0x2: FILCLK = PCLK/16 0x3: FILCLK = PCLK/64	0x0

第22章 复位功能

以下 8 种方法产生复位信号。

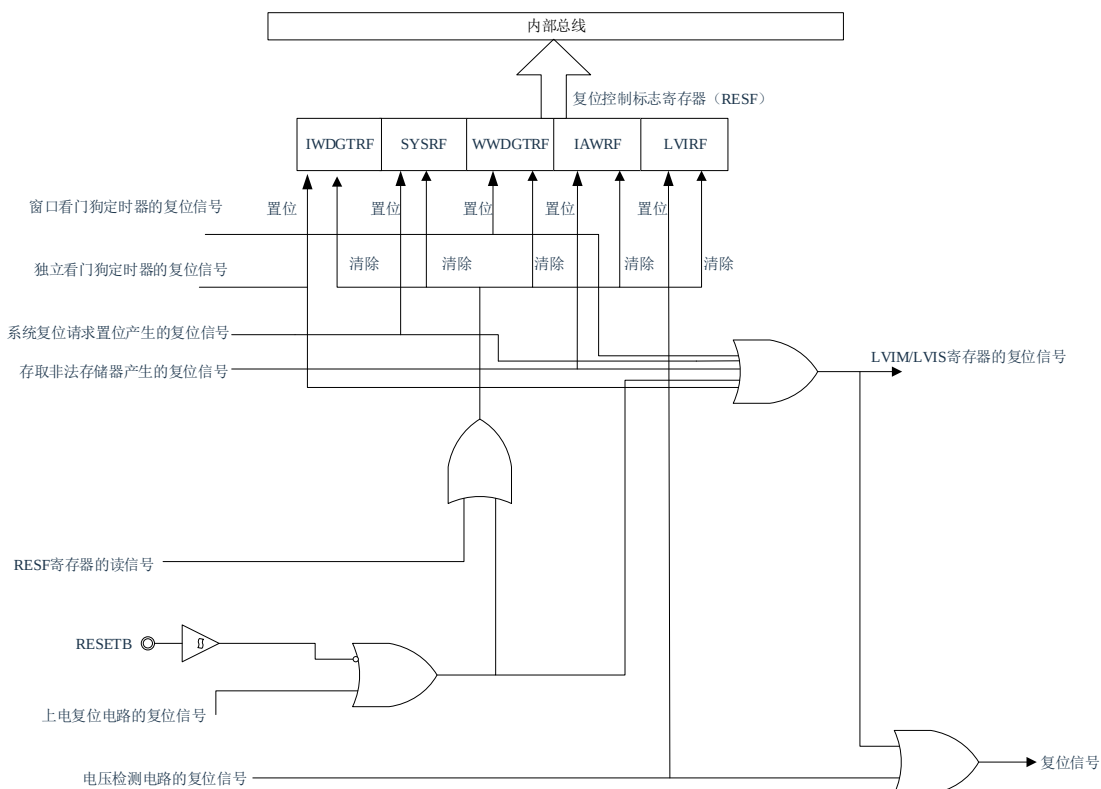
- 1) 通过 RESETB 引脚输入外部复位。
- 2) 通过独立看门狗定时器的程序失控检测产生内部复位 (IWDG)。
- 3) 通过窗口看门狗定时器的程序失控检测产生内部复位 (WWDG)。
- 4) 通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- 5) 通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- 6) 因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为 1 而产生内部复位。
- 7) 存取非法存储器而产生内部复位。
- 8) TRIM 修调的正反码复位。

内部复位和外部复位相同，在产生复位信号后，从用户自定义的程序起始地址开始执行程序。

当给 RESETB 引脚输入低电平，或者独立看门狗定时器检测到程序失控，或者窗口看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者系统复位请求位被置位，或者存取非法存储器时，产生复位并且各硬件变为如表 22-1 所示的状态。

注 1：在进行外部复位时，必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位，就必须在给 RESETB 引脚输入低电平后接通电源，而且在用户手册的 AC 特性所示的工作电压范围内至少保持 10us 的低电平，然后输入高电平。

图 22-1： 复位功能的框图



注 1：LVD 电路的内部复位不会对 LVD 电路进行复位。

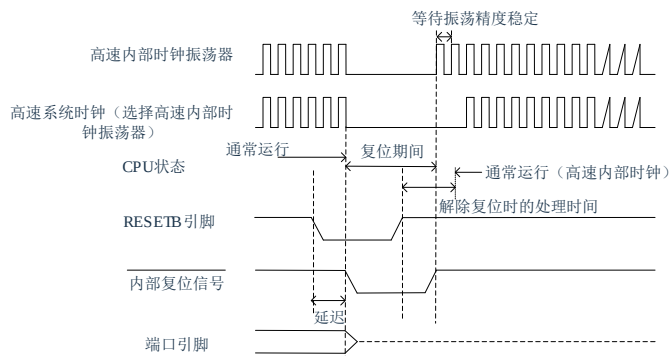
注 2：LVIM：电压检测寄存器。

注 3：LVIS：电压检测电平寄存器。

22.1 复位时序

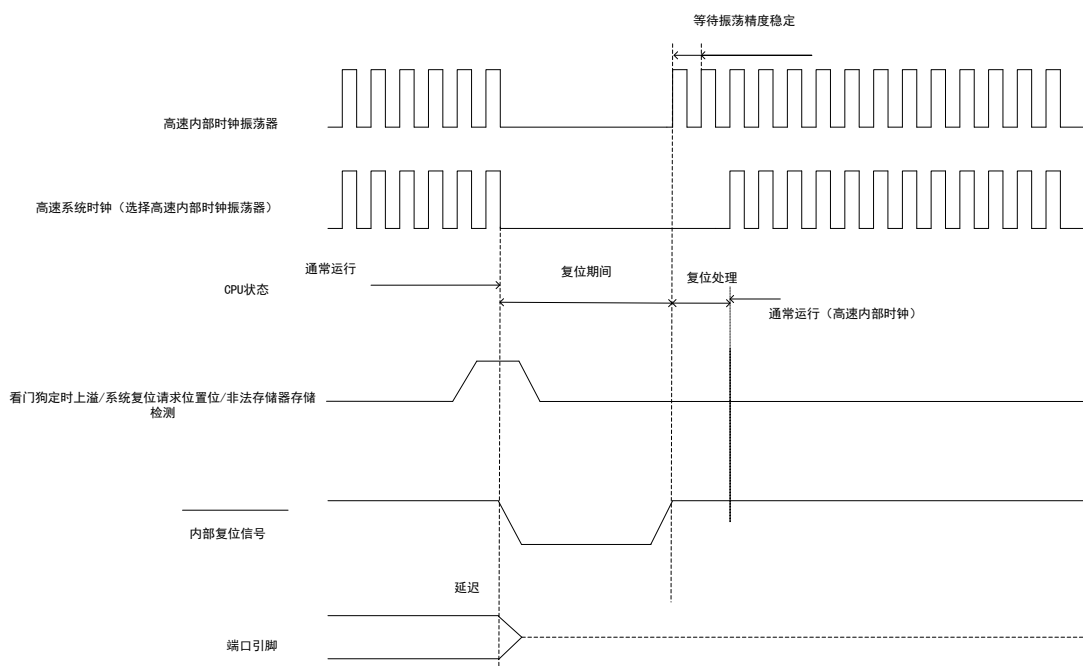
当给 RESETB 引脚输入低电平时，产生复位。然后，如果给 RESETB 引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 22-2：RESETB 输入的复位时序



对于 IWDG/WWDG 请求的复位、系统复位请求位的置位、TRIM 修调的正反码复位或者非法存储器存取检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 22-3 因 IWDG/WWDG 请求的复位、系统复位请求位的置位、TRIM 修调的正反码复位或者非法存储器存取检测而产生的复位时序。



注 1：端口引脚 PF2，PA13，PA14 变为以下状态：

- 在外部复位期间 PF2 为外部输入信号确定，PA13 为高电平，PA14 为低电平。
- POR 复位的期间 PA13 为高电平，PA14 为低电平。。

注 2：看门狗定时器也不例外，在发生内部复位时进行复位。

对于由 POR 电路和 LVD 电路的电压检测产生的复位，如果在复位后满足 $V_{DD} \geq V_{POR}$ 或者 $V_{DD} \geq V_{LVD}$ ，就解除复位状态，并且在复位处理后以高速内部振荡器时钟开始执行程序。详细内容请参照“第 23 章 上电复位电路”和“第 24 章 电压检测电路”。

备注： V_{POR} ：POR 电源电压上升检测电压

V_{LVD} ：LVD 检测电压。

表 22-1: 复位期间的运行状态

项目		复位期间						
		外部复位 RESINB	独立看门狗 复位	上电复位	窗口看门狗 复位	LVD 电压 检测复位	写复位寄存 器复位	存取非法 存储器复 位
系统时钟	f _{IH}	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
	f _{IL}	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
CPU		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
代码闪存		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SRAM		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
15 位间隔定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
TIM1		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
TIM3		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
TIM14		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
TIM16/17		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
窗口看门狗定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
独立看门狗定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SPI 通信单元		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
I2C 串行接口控制器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
USART		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
数据传送控制器 (DMA)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
A/D 转换器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
模拟比较器(ACMP0/1)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
上电复位功能		能进行检测 运行	能进行检测 运行	能进行检 测运行	能进行检 测运行	能进行检 测运行	能进行检 测运行	能进行检 测运行
电压检测功能		不能运行	不能运行	不能运行	不能运行	能运行	不能运行	不能运行
外部中断		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
CRC 运算功能	高速 CRC	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
	通用 CRC	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SFR 保护功能		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
复位完成后上电状态(选项字 节配置)		重新配置	重新配置	重新配置	重新配置	重新配置	重新配置	重新配置

注 1: PA13、PF2 上电后为高 (PF2 口在外部复位时为外部输入的状态), PA14 上电后为低, 其他脚复位前中后均为高阻态。

注 2: F_{IH}: 高速内部振荡器时钟; F_{IL}: 低速内部振荡器时钟。

22.2 确认复位源的寄存器

22.2.1 寄存器映射

(复位控制基地址 = 0x4002_0440) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
RESF	0x000	RO	复位控制标志寄存器	-

22.2.2 复位控制标志寄存器(RESF)

该微控制器存在多种内部复位发生源。复位控制标志寄存器（RESF）保存发生复位请求的复位源。能通过 8 位存储器操作指令读 RESF 寄存器。

通过 RESETB 的输入、上电复位（POR）电路的复位和 RESF 寄存器的读取，清除 SYSRF、WWDGRF、IWDGRF、OSDCRF、IAWRF、LVIRF 标志。要判断复位源时，必须将 RESF 寄存器的值保存到任意的 RAM，然后通过其 RAM 值进行判断。

位	符号	描述	复位值
7	SYSRF	系统复位请求位被置位而产生的内部复位请求 0: 没有产生内部复位请求或清除了 RESF 寄存器。 1: 产生内部复位请求。	-
6	-	保留	-
5	IWDGRF	独立看门狗定时器（IWDG）产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-
4	WWDGRF	窗口看门狗定时器（WWDG）产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-
3:2	-	保留	-
1	IAWRF	存取非法存储器产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-
0	LVIRF	电压检测电路（LVD）产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-

注：RESF 寄存器的状态因复位源而不同。请参照表 22-2。

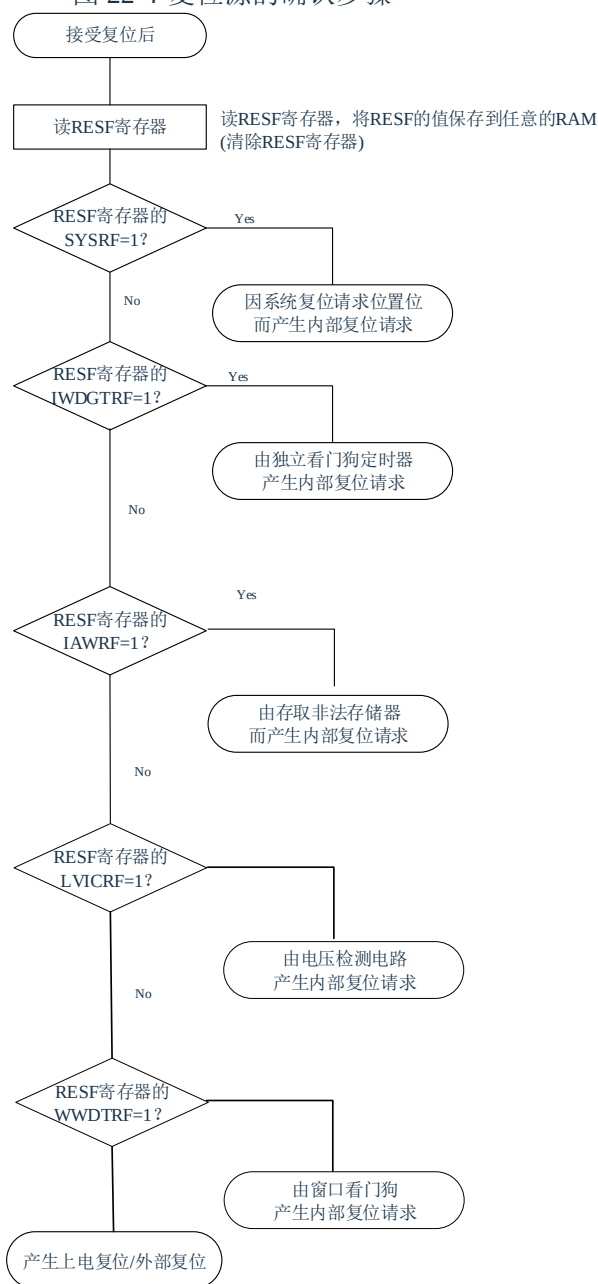
发生复位请求时的 RESF 寄存器状态如下表所示。

表 22-2：发生复位请求时的 RESF 寄存器状态

标志 \ 复位源	RESETB 输入	POR 产生的复位	系统复位请求位置位产生的复位	IWDG 产生的复位	存取非法存储器产生的复位	LVD 产生的复位	WWDG 产生的复位
SYSRF	清“0”	清“0”	置“1”	保持	保持	保持	保持
IWDGRF			保持	置“1”	置“1”		
IAWRF					保持	置“1”	置“1”
LVIRF					保持	保持	
WWDGRF							

复位源的确认步骤如图 22-4 所示。

图 22-4 复位源的确认步骤



第23章 上电复位电路

23.1 上电复位电路的功能

上电复位电路（POR）有以下功能：

在接通电源时产生内部复位信号。

- 如果电源电压（ V_{DD} ）超过检测电压（ V_{POR} ），就解除复位。但是，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态。
- 将电源电压（ V_{DD} ）和检测电压（ V_{PDR} ）进行比较。当 $V_{DD} < V_{PDR}$ 时，产生内部复位信号。但是，当电源电压下降时，必须在电源电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注 1：当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清“00H”。

注 2：芯片内置多个产生内部复位信号的硬件。当由窗口看门狗定时器（WWDG）、独立看门狗定时器（IWDG）、电压检测（LVD）电路、系统复位请求位置位或者非法存储器的存取而产生内部复位信号时，用于表示复位源的标志分配在 RESF 寄存器；当由 WWDG、IWDG、LVD、系统复位请求位的置位或者非法存储器的存取而产生内部复位信号时，不将 RESF 寄存器清“00H”，而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 22 章 复位功能”。

注 3： V_{POR} ：POR 电源电压上升检测电压

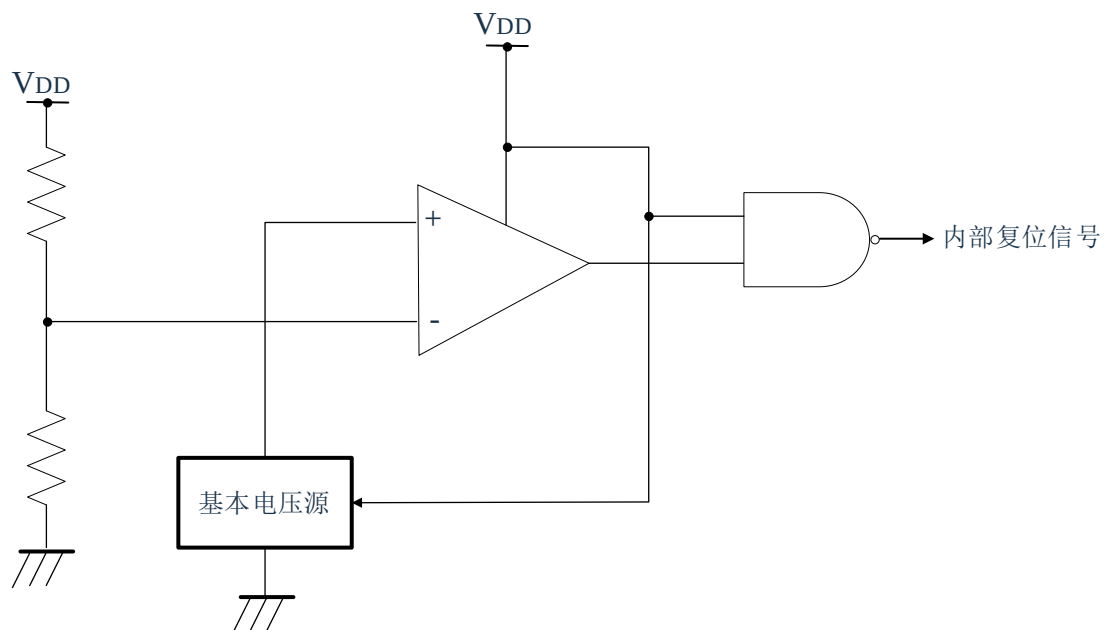
注 4： V_{PDR} ：POR 电源电压下降检测电压

注 5：有关 V_{POR} 、 V_{PDR} 详细内容请参照数据手册的 POR 电路特性。

23.2 上电复位电路的结构

上电复位电路的框图如图 23-1 所示。

图 23-1 上电复位电路的框图

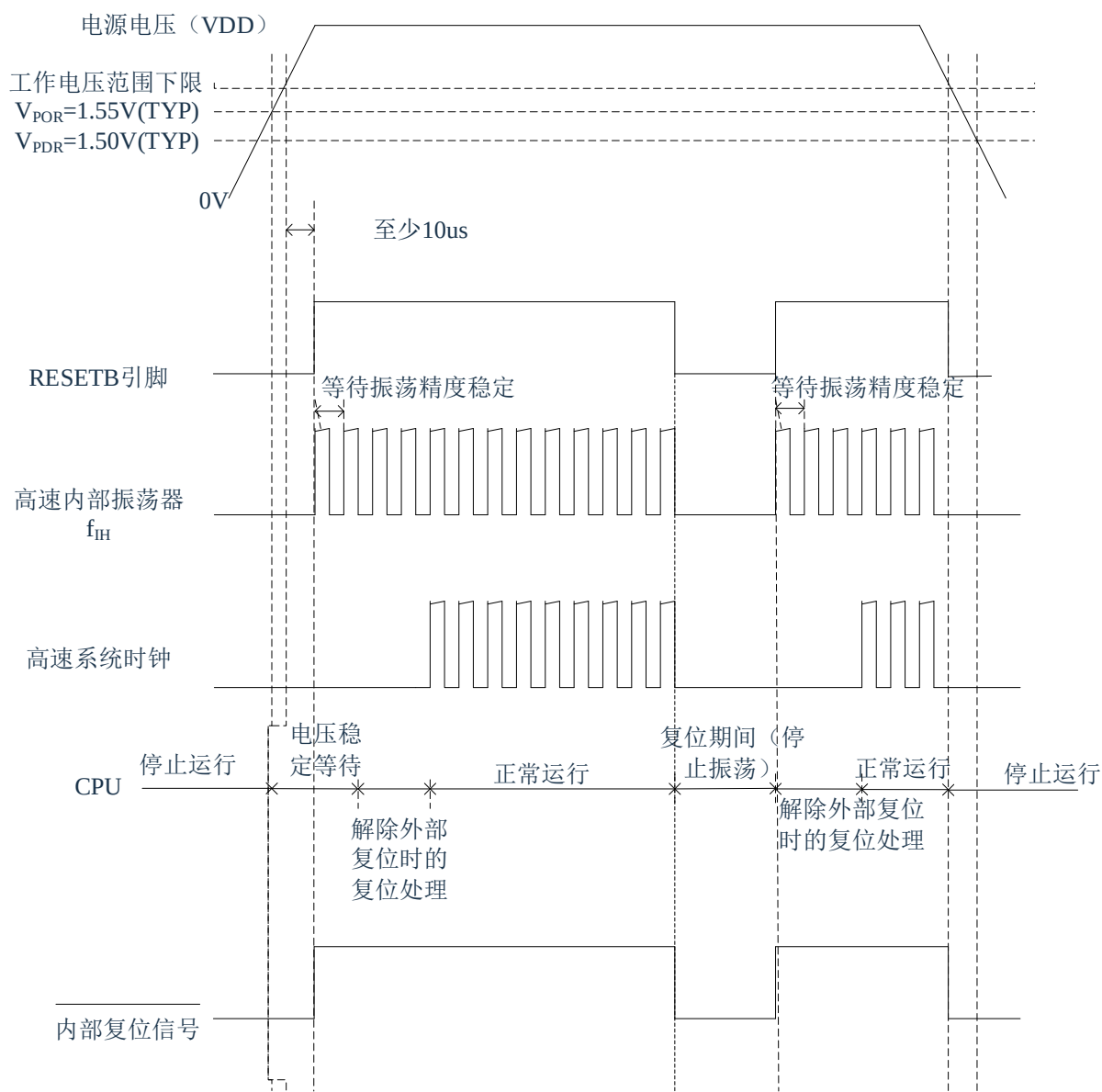


23.3 上电复位电路的运行

上电复位电路和电压检测电路的内部复位信号的产生时序如下所示。

图 23-2 上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

1. 使用 RESETB 引脚的外部复位输入的情况



注 1: 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注 2: 当电源电压上升时, 必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前, 通过外部复位保持复位状态; 当电源电压下降时, 必须在电源电压低于工作电压范围前, 通过深度睡眠模式的转移、电压检测电路或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

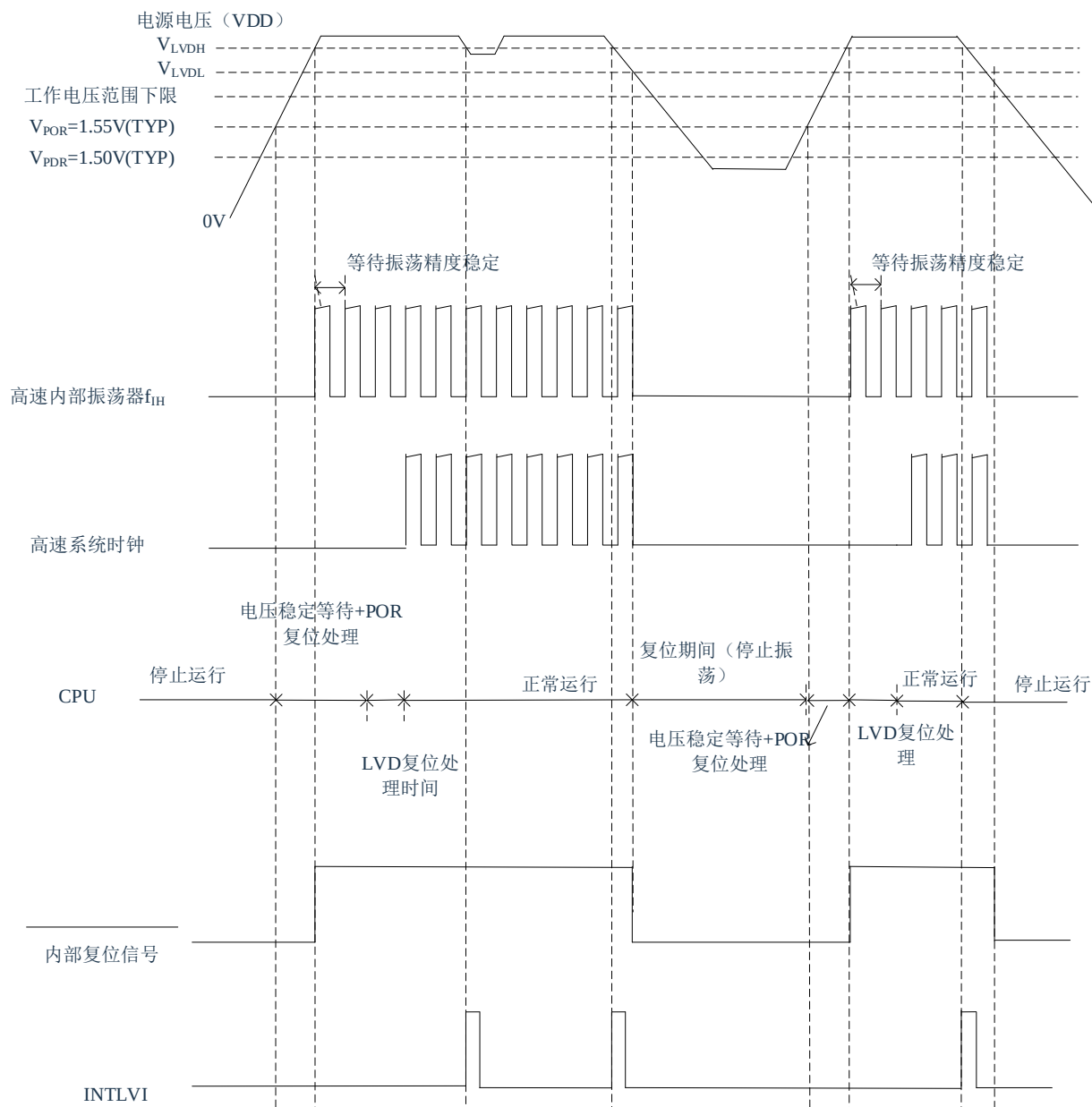
注 3: V_{POR} : POR 电源电压上升检测电压。

注 4: V_{PDR} : POR 电源电压下降检测电压。

注 5: 在 LVD 为 OFF 时, 必须使用 RESETB 引脚的外部复位。详细内容请参照“第 24 章 电压检测电路”。

图 23-2: 上电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

2. LVD 为中断&复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、0）



注 1: 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注 2: 在产生中断请求信号（INTLVI）后，自动将电压检测电寄存器（LVIS）的 LVILV 位和 LVIMD 位置“1”。因此，必须考虑可能出现电源电压在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高的情况，在产生 INTLVI 后按照“图 24-5 工作电压的确认/复位的设定步骤”和“图 24-6 中断&复位模式的初始设定步骤”进行设定。

注 3: 到开始通常运行为止的时间除了达到 V_{POR} （1.55V(TYP.)) 后的“电压稳定等待+POR 复位处理”以外，在达到 LVD 检测电平（ V_{LVDH} ）后还需要的“LVD 复位处理”。

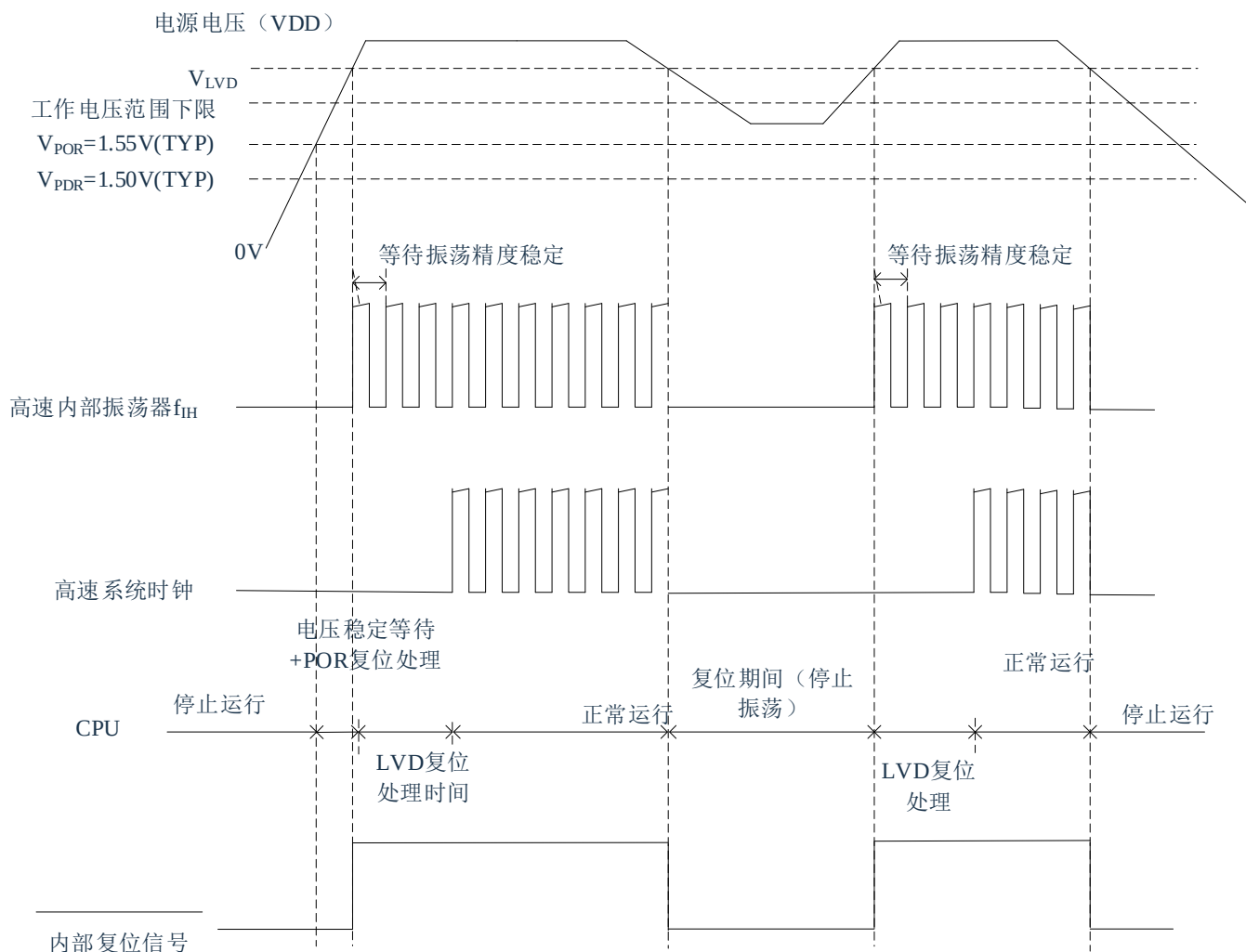
注 4: V_{LVDH} 、 V_{LVDL} : LVD 检测电压。

注 5: V_{POR} : POR 电源电压上升检测电压。

注 6: V_{PDR} : POR 电源电压下降检测电压。

图 23-2 上电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

3. LVD 复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、1）



注 1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注 2：到开始通常运行为止的时间除了达到 VPOR（1.55V(TYP)）后的“电压稳定等待+POR 复位处理”以外，在达到 LVD 检测电平（VLVD）后还需要“LVD 复位处理”。

注 3：在电源电压下降时，如果只在发生电压检测电路（LVD）的内部复位后恢复电源电压，就在达到 LVD 检测电平（VLVD）后需要“LVD 复位处理”。

注 4：VLVDH、VLVDL:LVD 检测电压。

注 5：VPOR：POR 电源上升检测电压。

注 6：VPDR：POR 电源下降检测电压。

注 7：当选择 LVD 中断模式（选项字节 000C1H 的 LVIMD1、LVIMD0=0、1）时，从接通电源到开始正常运行的时间和“图 23-2(3/3)LVD 复位模式的情况”的 LVD 复位处理的时间相同。

第24章 电压检测电路

24.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设定运行模式和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）。电压检测（LVD）电路有以下功能：

- 将电源电压（VDD）和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）进行比较，产生内部复位或者内部中断信号。
- 电源电压的检测电压（ V_{LVDH} 、 V_{LVDL} ）能通过选项字节选择 8 种检测电平（参照“第 26 章 选项字节”）。
- 也能在深度睡眠模式中运行。
- 当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节。

1. 中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）

通过选项字节 000C1H 选择 2 个检测电压（ V_{LVDH} 、 V_{LVDL} ），高电压检测电平（ V_{LVDH} ）用于解除复位或者产生中断，低电压检测电平（ V_{LVDL} ）用于产生复位。

2. 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）

将选项字节 000C1H 选择的 1 个检测电压（ V_{LVD} ）用于产生或者解除复位。

3. 中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）

将选项字节 000C1H 选择的 1 个检测电压（ V_{LVD} ）用于产生中断或者解除复位。在各模式中，产生以下的中断信号和内部复位信号。

模式	中断&复位模式	复位模式	中断模式
设置	LVIMDS1、LVIMDS0=1、0	LVIMDS1、LVIMDS0=1、1	LVIMDS1、LVIMDS0=0、1
运行过程	在工作电压下降时，当检测到 $VDD < V_{LVDH}$ 时，产生中断请求信号；当检测到 $VDD < V_{LVDL}$ 时，产生内部复位；当检测到 $VDD \geq V_{LVDH}$ 时，解除内部复位。	当检测到 $VDD \geq V_{LVD}$ 时，解除内部复位；当检测到 $VDD < V_{LVD}$ 时，产生内部复位。	在发生复位后，LVD 的内部复位状态继续保持到 $VDD \geq V_{LVD}$ 为止。当检测到 $VDD \geq V_{LVD}$ 时，解除 LVD 的内部复位。在解除 LVD 的内部复位后，如果检测到 $VDD < V_{LVD}$ 或者 $VDD \geq V_{LVD}$ 时，就产生中断请求信号（INTLVI）。

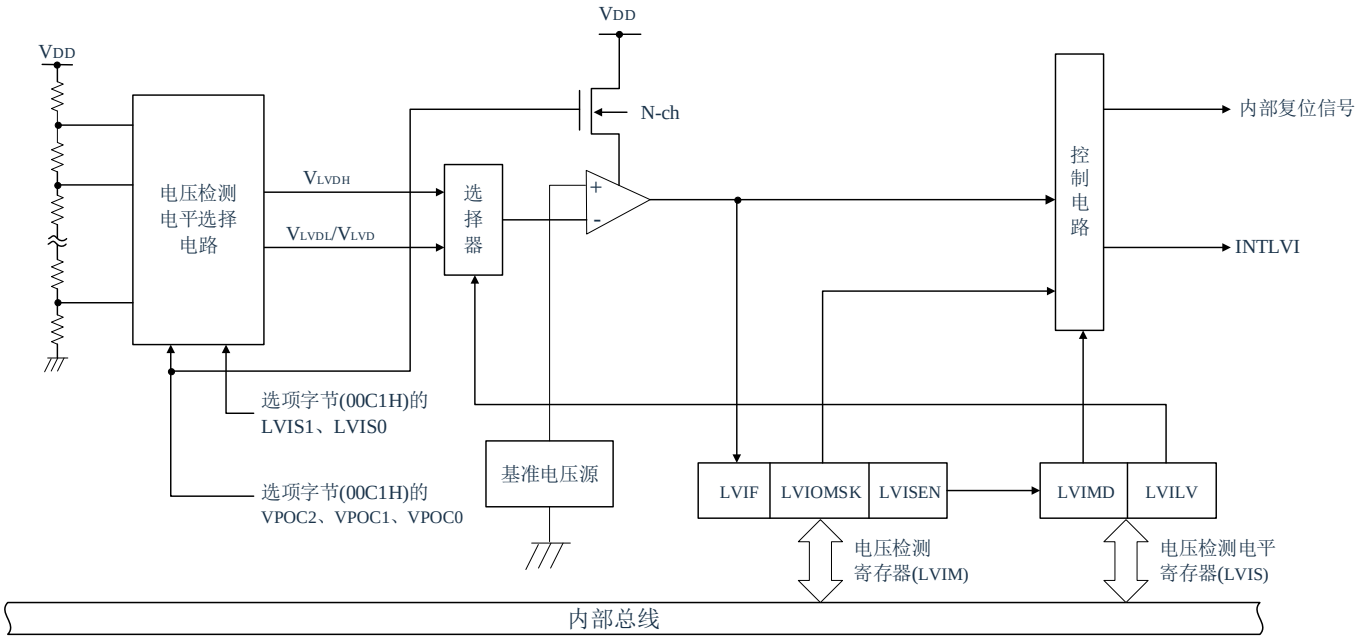
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

如果发生复位，就将复位控制标志寄存器（RESF）的 bit0（LVIRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 22 章 复位功能”。

24.2 电压检测电路的结构

电压检测电路的框图如图 24-1 所示。

图 24-1 ： 电压检测电路的框图



24.3 寄存器映射

(电压检测基地址 = 0x4002_0441) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LVIM	0x000	R/W	电压检测寄存器	0x0
LVIS	0x001	R/W	电压检测电平寄存器	--

24.4 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器（LVIM）
- 电压检测电平寄存器（LVIS）

24.4.1 电压检测寄存器（LVIM）

此寄存器设定允许或者禁止改写电压检测电平寄存器（LVIS），并且确认 LVD 输出的屏蔽状态。通过 8 位存储器操作指令 设定 LVIM 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	LVISEN ^{注1}	电压检测电平寄存器（LVIS）的允许/禁止改写的设定 0: 禁止 1: 允许	-
6:3	-	保留	-
2	LVD_BRKEN_TIM	LVD 输出用作 TIM1/TIM16/TIM17 刹车输入使能控制 0: 禁止 1: 允许	0
1	LVIOMSK	LVD 输出的屏蔽状态标志 0: 屏蔽无效 1: 屏蔽有效 ^{注2}	-
0	LVIF	电压检测标志 0: 电源电压（V _{DD} ）≥检测电压（V _{LVD} ）或者 LVD 为 OFF。 1: 电源电压（V _{DD} ）<检测电压（V _{LVD} ）	-

注 1: 只有在选择中断&复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时才能设定，在其他模式中不能更改初始值。

注 2: 只有在选择中断&复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时，LVIOMSK 位才在以下期间自动变为“1”，屏蔽 LVD 产生的复位或者中断。

- LVISEN=1 的期间
- 从发生 LVD 中断开始到 LVD 检测电压稳定为止的等待时间
- 从更改 LVILV 位的值到 LVD 检测电压稳定为止的等待时间

24.4.2 电压检测电平寄存器 (LVIS)

这是设定电压检测电平的寄存器。

通过 8 位存储器操作指令设定 LVIS 寄存器。在产生复位信号后，此寄存器的值变为“00H/01H/81H”^{注1}。

位	符号	描述	复位值
7	LVIMD ^{注1}	电压检测的运行模式 0: 中断模式 1: 复位模式	0
6:1	-	保留	-
0	LVILV ^{注2}	LVD 检测电平 0: 高电压检测电平 (VLVDH) 1: 低电压检测电平 (VLVDH 或者 VLVD)	0

注 1: 复位值因复位源和选项字节的设定而变。在发生 LVD 复位时，不将此寄存器清“00H”。

在发生 LVD 以外的复位时，此寄存器的值如下：

- 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
- 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
- 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H

注 2: 只有在选择中断&复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时才能写“0”。在其他情况下不能设定。在中断&复位模式中，通过产生复位或者中断自动进行值的替换。

注意 1: 要改写 LVIS 寄存器时，必须按照图 24-6 和图 24-7 的步骤进行。

2: 通过选项字节 000C1H 选择 LVD 的运行模式和各模式的检测电压 (VLVDH、VLVDL、VLVD)。用户选项字节 (000C1H) 的详细内容，请参照“第 26 章 选项字节”。

24.5 电压检测电路的运行

24.5.1 用作复位模式时的设定

通过选项字节 000C1H 设定运行模式（复位模式（LVIMDS1、LVIMDS0=1、1）和检测电压（ V_{LVD} ）。如果设定复位模式，就在以下初始设定的状态下开始运行。

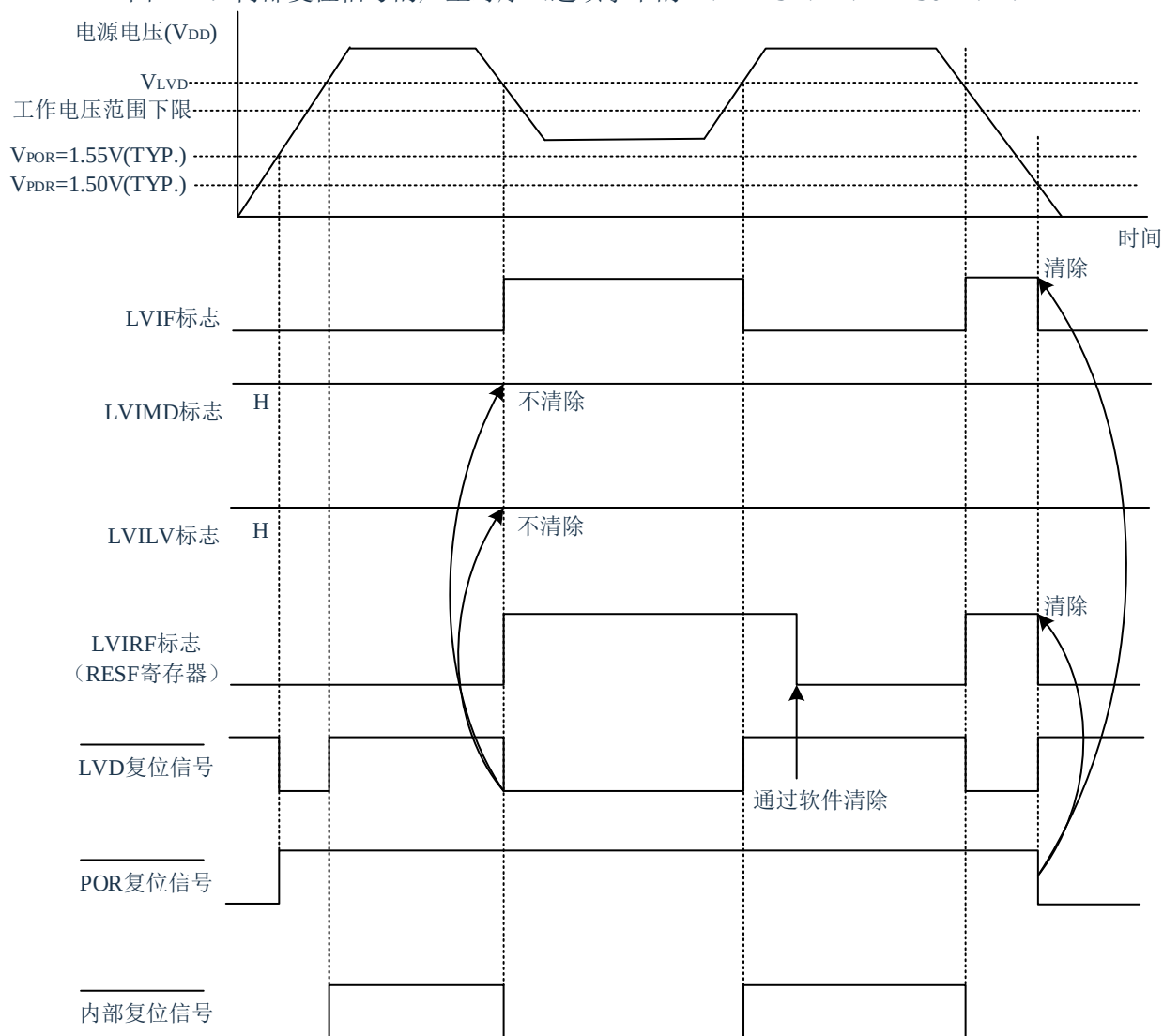
- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“81H”。bit7（LVIMD）为“1”（复位模式）。bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。
- LVD 复位模式的运行

当接通电源时，复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）在电源电压（VDD）超过电压检测电平（ V_{LVD} ）前保持 LVD 的内部复位状态。如果电源电压（VDD）超过电压检测电平（ V_{LVD} ），就解除内部复位。

当工作电压下降时，如果电源电压（VDD）低于电压检测电平（ V_{LVD} ），就产生 LVD 的内部复位。

LVD 复位模式的内部复位信号的产生时序如图 24-2 所示。

图 24-2：内部复位信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、1）



注 1： V_{POR} ：POR 电源电压上升检测电压

注 2： V_{PDR} ：POR 电源电压下降检测电压

24.6 用作中断模式时的设定

通过选项字节 000C1H 设定运行模式（中断模式（LVIMDS1、LVIMDS0=0、1）和检测电压（ V_{LVD} ）。如果设定中断模式，就在以下初始设定的状态下开始运行。

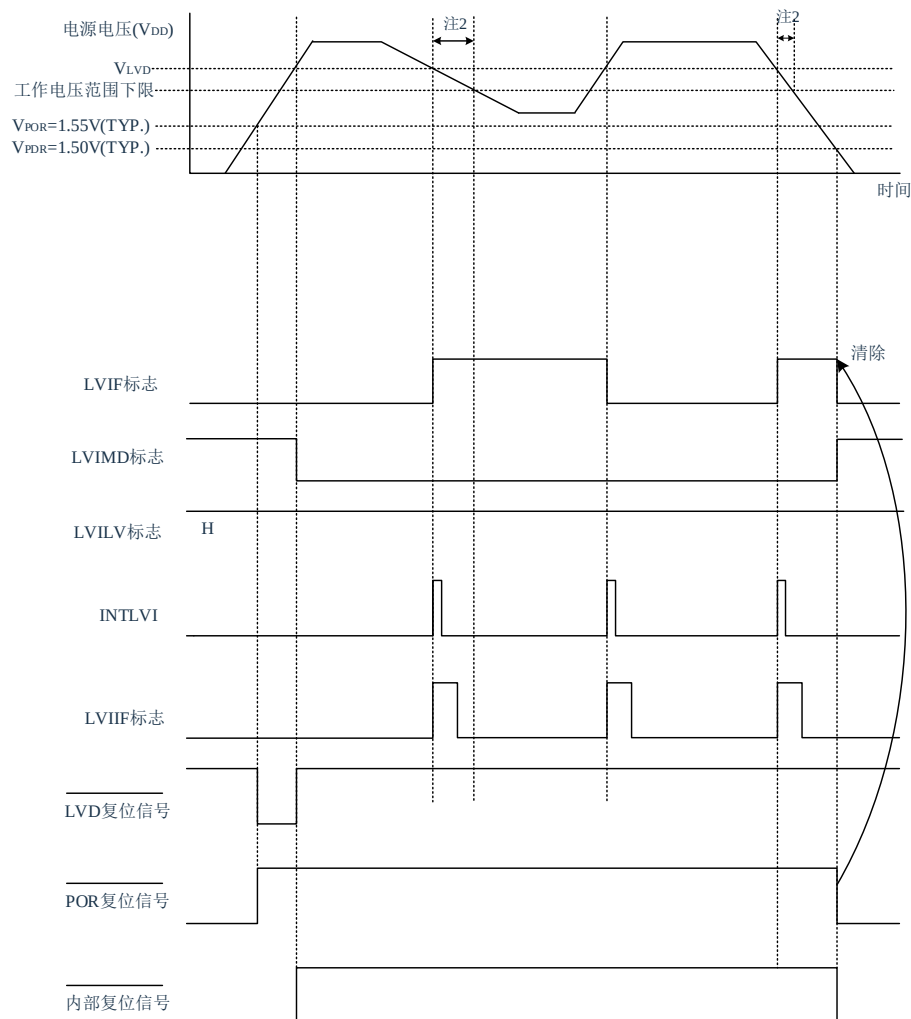
- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“01H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。
- LVD 中断模式的运行

在产生复位后，中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）在电源电压（VDD）超过电压检测电平（ V_{LVD} ）前保持 LVD 的内部复位状态。如果电源电压（VDD）超过电压检测电平（ V_{LVD} ），就解除 LVD 的内部复位。

在解除 LVD 的内部复位后，如果电源电压（VDD）超过电压检测电平（ V_{LVD} ），就产生 LVD 的中断请求信号（INTLVI）。当工作电压下降时，必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD 中断模式的中断请求信号的产生时序如图 24-3 所示。

图 24-3 中断信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=0、1）



注 1: 当工作电压下降时, 必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前, 通过深度睡眠模式的转移或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

备注: V_{POR} : POR 电源电压上升检测电压。

V_{PDR} : POR 电源电压下降检测电压。

24.7 用作中断&复位模式时的设定

通过选项字节 000C1H 设定运行模式（中断&复位模式（LVIMDS1、LVIMDS0=1、0））和检测电压（ V_{LVDH} 、 V_{LVDL} ）。

如果设定中断&复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“00H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“0”（高电压检测电平： V_{LVDH} ）。
- LVD 中断&复位模式的运行

当接通电源时，中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（VDD）超过高电压检测电平（ V_{LVDH} ）前保持 LVD 的内部复位状态。如果电源电压（VDD）超过高电压检测电平（ V_{LVDH} ），就解除内部复位。

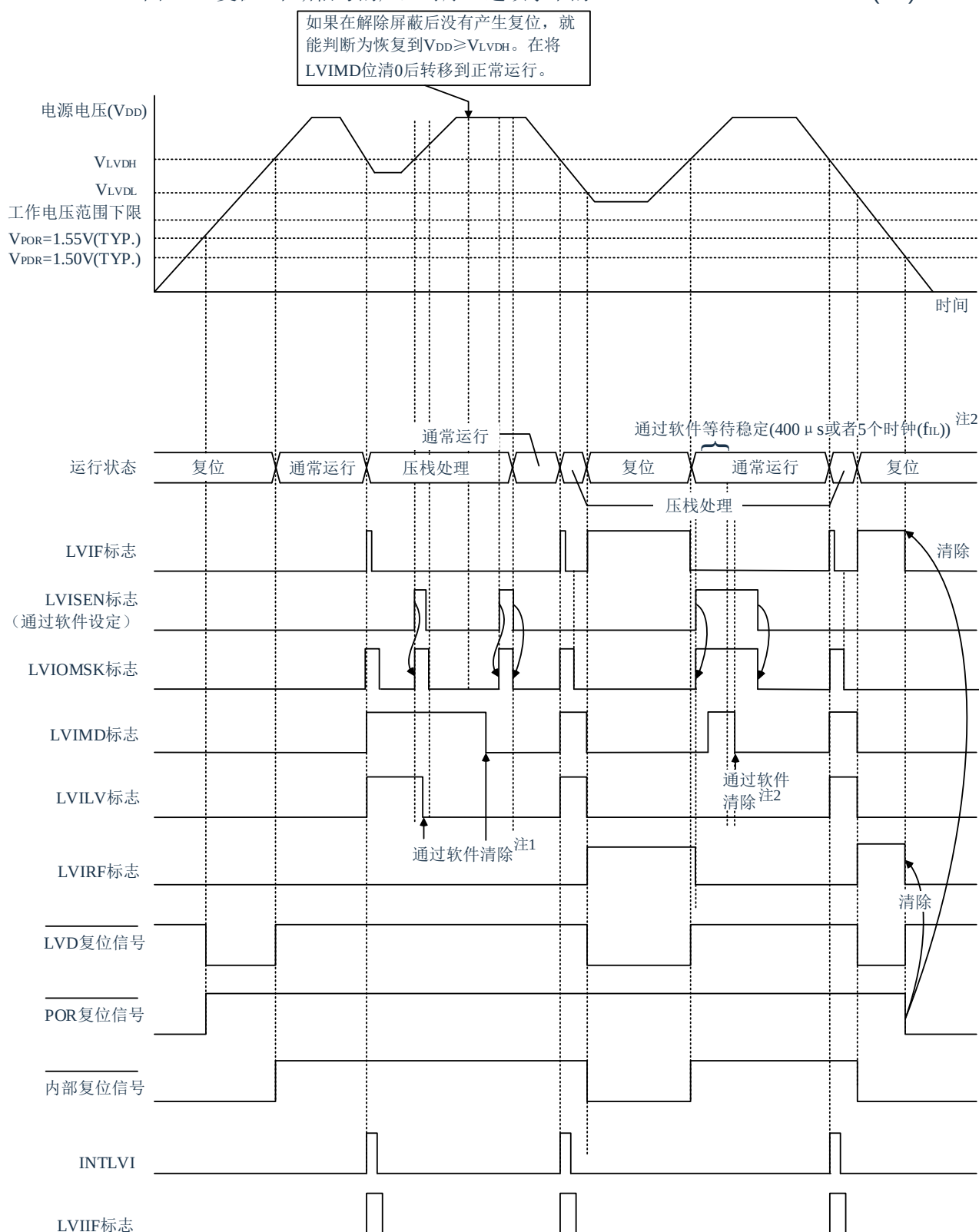
当工作电压下降时，如果电源电压（VDD）低于高电压检测电平（ V_{LVDH} ），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（VDD）低于低电压检测电平

（ V_{LVDL} ），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压（VDD）在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高，也不产生中断请求信号。

当使用 LVD 中断&复位模式时，必须按照“图 24-6 工作电压的确认/复位的设定步骤”和“图 24-7”所示的流程图的步骤进行设定。

LVD 中断&复位模式的内部复位信号和中断信号的产生时序如图 24-4 示。

图 24-4 复位&中断信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、0）(1/2)



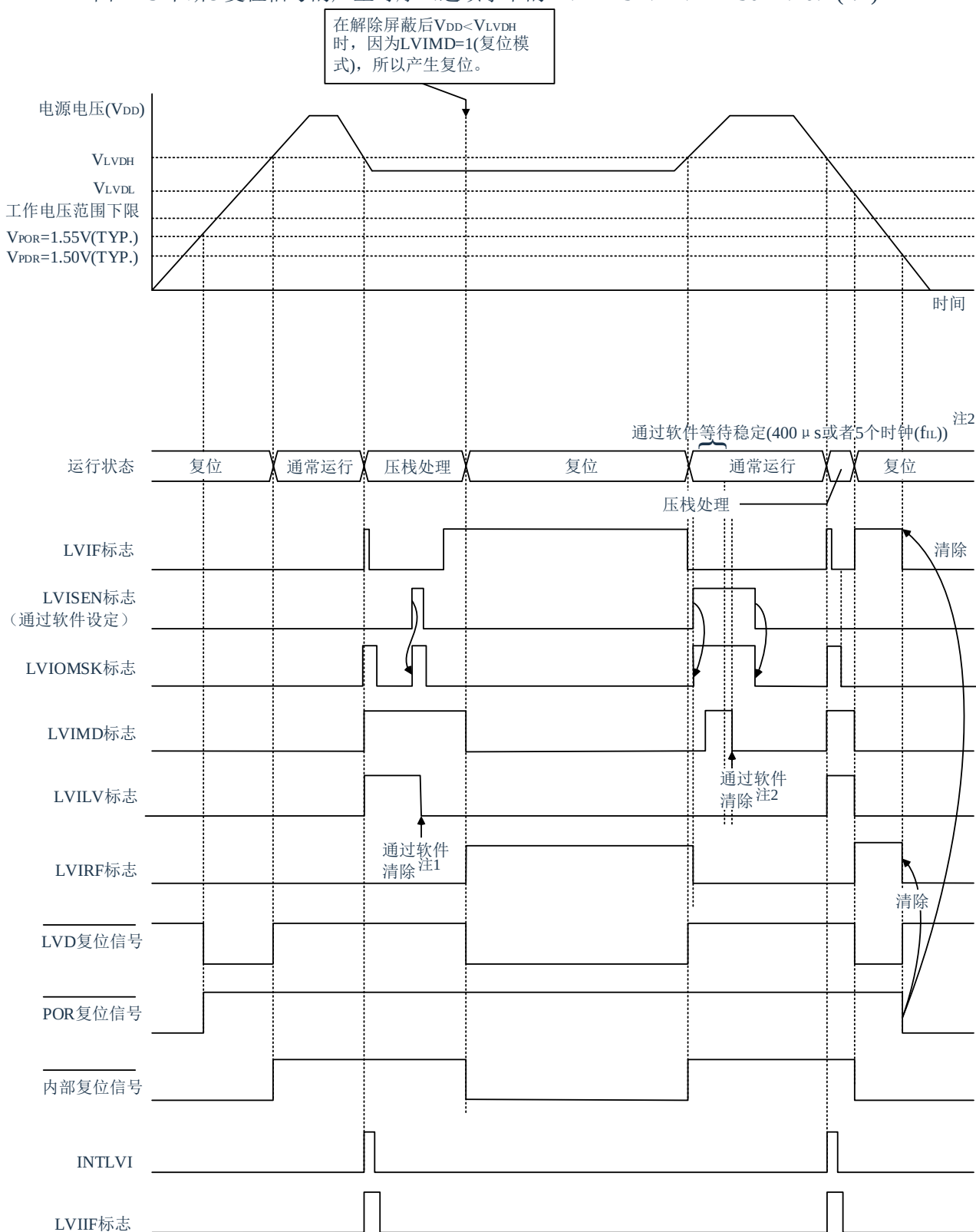
注 1：当使用中断&复位模式时，必须在发生中断后按照“图 24-6 工作电压的确认/复位的设定步骤”进行设定。

注 2：当使用中断&复位模式时，必须在解除复位后按照“图 24-7”进行设定。

注 3： V_{POR} ：POR 电源电压上升检测电压。

V_{PDR} ：POR 电源电压下降检测电压。

图 24-5 中断&复位信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、0）(2/2)



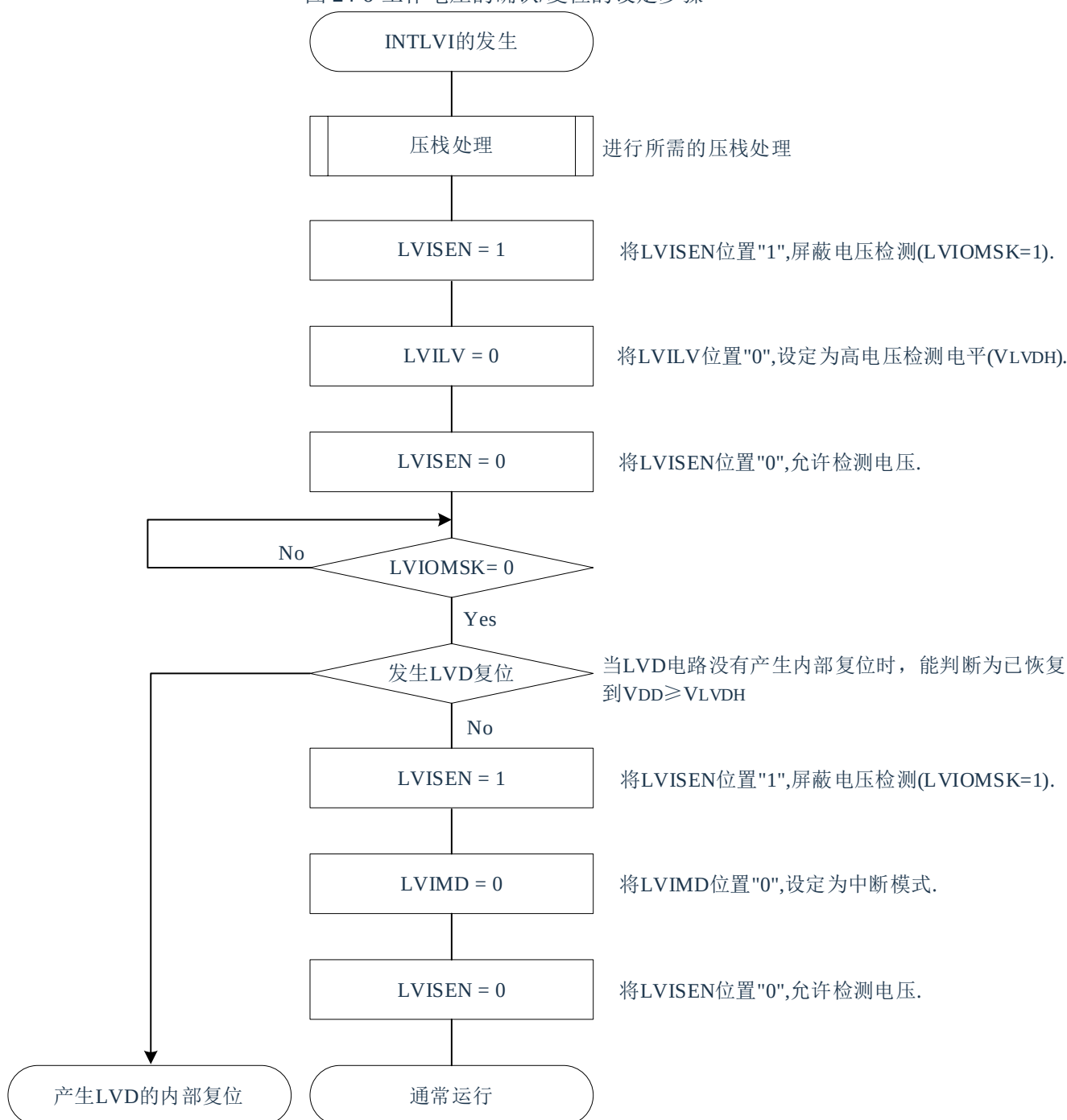
注 1：当使用中断&复位模式时，必须在发生中断后按照“图 24-6 工作电压的确认/复位的设定步骤”进行设定。

注 2：当使用中断&复位模式时，必须在解除复位后按照“图 24-7”进行设定。

注 3： V_{POR} ：POR 电源电压上升检测电压。

V_{PDR} ：POR 电源电压下降检测电压。

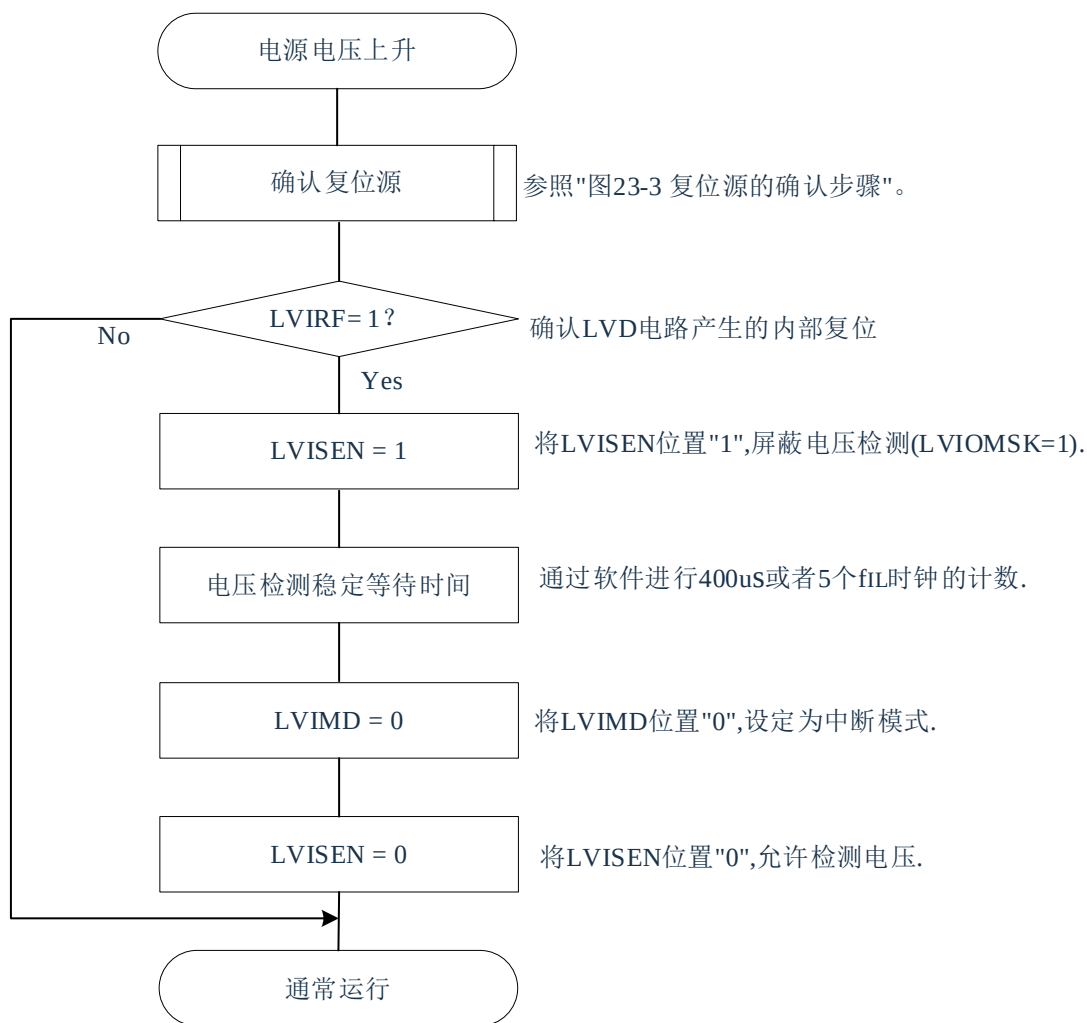
图 24-6 工作电压的确认/复位的设定步骤



如果设定中断&复位模式（LVIMDS1、LVIMDS0=1、0），就在解除 LVD 复位（LVIRF=1）后需要 400us 或者 5 个 f_{L} 时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将 LVIMD 位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写 LVIMD 位时，必须将 LVISEN 位置“1”，屏蔽 LVD 产生的复位或者中断的产生。

中断&复位模式的初始设定步骤如图 24-7 所示。

图 24-7 中断&复位模式的初始设定步骤



备注：f_{IL}：低速内部振荡器时钟频率

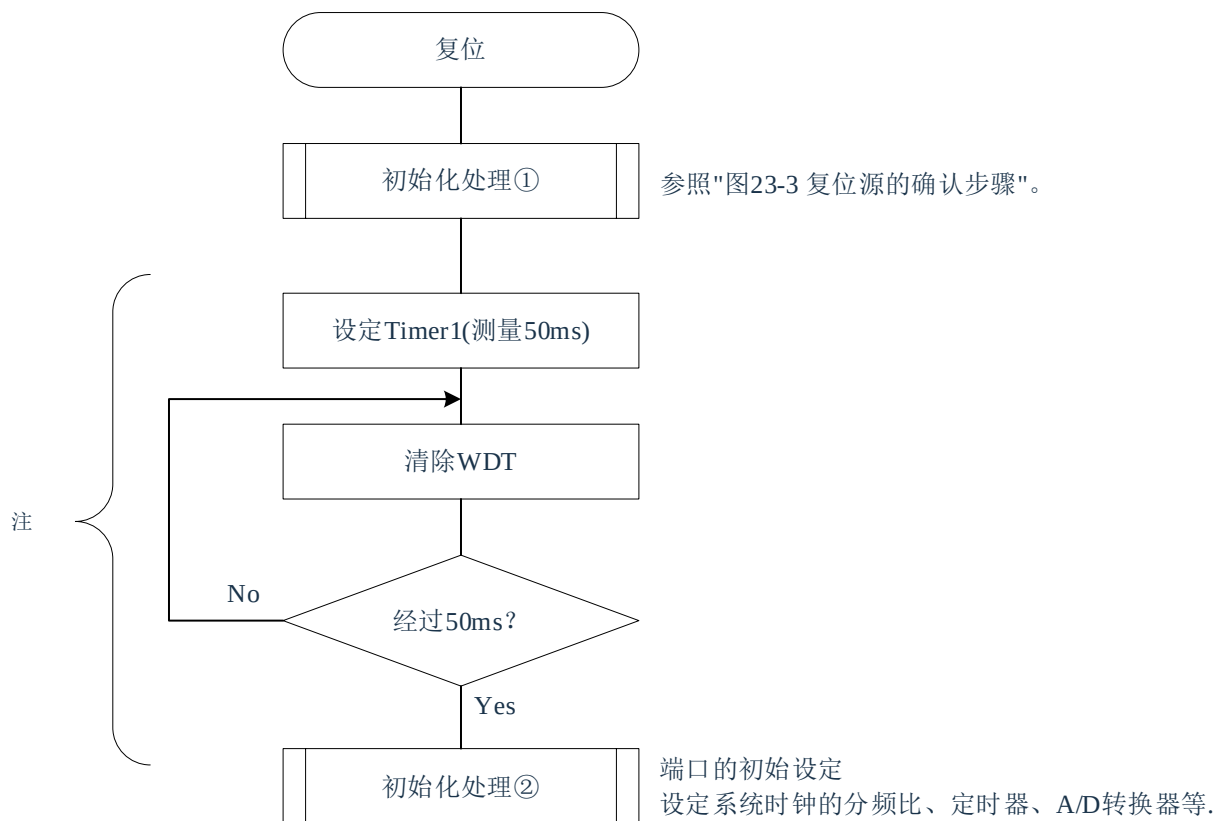
24.8 电压检测电路的注意事项

1. 有关接通电源时的电压波动

对于电源电压 (V_{DD}) 在 LVD 检测电压附近发生一定时间波动的系统, 有可能重复进入复位状态和复位解除状态。能通过以下的处理, 任意设定解除复位到单片机开始运行的时间。

在解除复位后, 必须通过使用定时器的软件计数器, 在等待各系统不同的电源电压波动时间后进行端口等的初始设定。

图 24-8: LVD 检测电压附近的电源电压波动不超过 50ms 时的软件处理例子

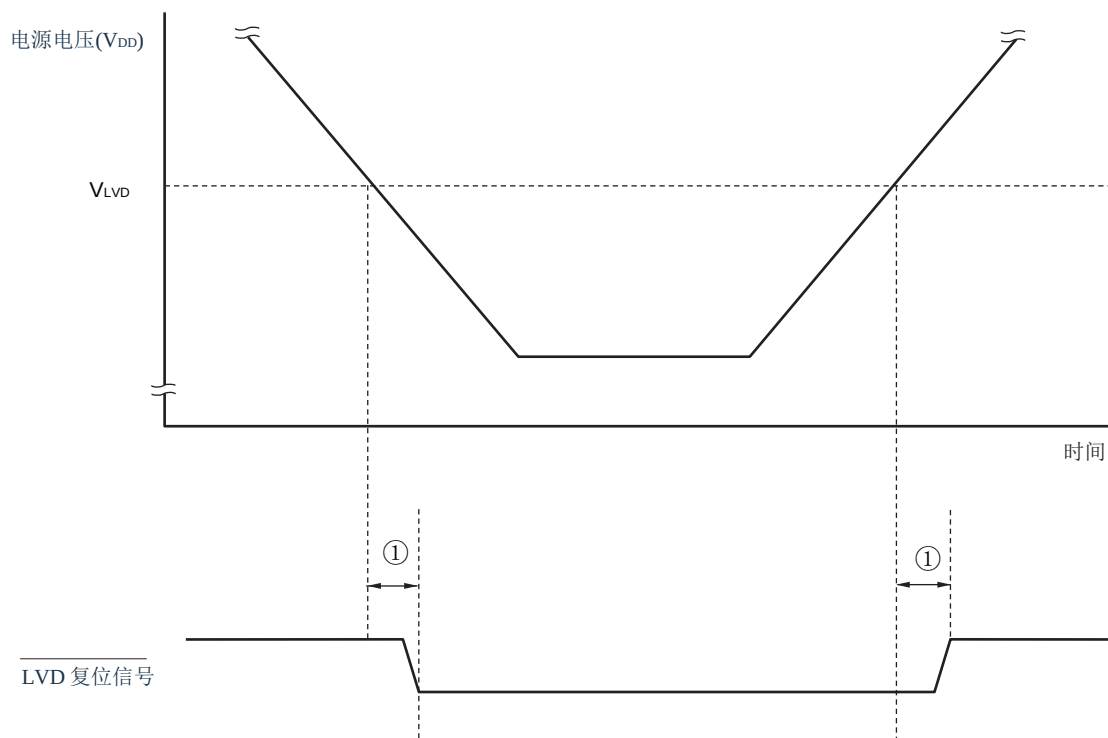


注 1: 如果在此期间再次发生复位, 就不转移到初始化处理②。

2. 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟

从满足电源电压 (V_{DD}) $< V_{LVD}$ 检测电压 (V_{LVD}) 到产生 LVD 复位为止会发生延迟。同样, 从 LVD 检测电压 (V_{LVD}) $\leq$ 电源电压 (V_{DD}) 到解除 LVD 复位为止也会发生延迟 (参照图 24-9)。

图 24-9: 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟



①: 检测延迟(300us(MAX.))

3. 有关将 LVD 置为 OFF 时接通电源的情况

当将 LVD 置为 OFF 时, 必须使用 RESETB 引脚的外部复位。

在进行外部复位时, 必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位, 就必须在给 RESETB 引脚输入低电平后接通电源, 而且在数据手册的 AC 特性所示的工作电压范围内至少保持 10us 的低电平, 然后输入高电平。

4. 有关将 LVD 置为 OFF 并且设定为 LVD 中断模式时工作电压下降的情况

在将 LVD 置为 OFF 并且设定为 LVD 中断模式的情况下, 如果工作电压下降, 就必须在的工作电压低于数据手册的 AC 特性所示的工作电压范围前, 通过深度睡眠模式的转移或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

第25章 安全功能

25.1 安全功能的概要

为了对应 IEC60730 安全标准，该芯片内置以下安全功能。

此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

1. 闪存 CRC 运算功能（高速 CRC）

通过 CRC 运算检测闪存的数据错误。

- “高速 CRC”... 在初始化程序中，能停止 CPU 的运行并且高速检查整个代码闪存区。

2. SFR 保护功能

防止因 CPU 失控而改写 SFR。

3. A/D 测试功能

能通过 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、内部基准电压输出的 A/D 转换进行 A/D 转换器的自检。

4. 输入/输出端口的数字输出信号电平检测功能

在输入/输出端口为输出模式时，能读引脚的输出电平。

25.2 寄存器映射

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
- 闪存 CRC 控制寄存器 (CRC0CTL) - 闪存 CRC 运算结果寄存器 (PGCRCL)	闪存 CRC 运算功能 (高速 CRC)
特殊 SFR 保护控制寄存器 (SFRGD)	SFR 保护功能
A/D 测试寄存器 (ADS)	A/D 测试功能
端口模式选择寄存器 (PMS)	输入/输出引脚的数字输出信号电平检测功能

(闪存 CRC 基地址 = 0x4002_1810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CRC0CTL	0x000	R/W	闪存 CRC 控制寄存器	0x0
PGCRCL	0x002	R/W	闪存 CRC 运算结果寄存器	0x0

(SFR 基地址 = 0x4004_0478) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
SFRGD	0x000	R/W	SFR 保护控制寄存器	0x0

(端口控制基地址 = 0x4004_087B) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PMS	0x000	R/W	端口模式选择寄存器	0x0

(UID 基地址 = 0x0050_0894) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
UID0	0x000	RO	产品唯一的身份标识的第[31:0]位	-
UID1	0x004	RO	产品唯一的身份标识的第[63:32]位	-
UID2	0x008	RO	产品唯一的身份标识的第[95:64]位	-
UID3	0x00C	RO	产品唯一的身份标识的第[127:96]位	-

25.3 安全功能的运行

25.3.1 闪存 CRC 运算功能（高速 CRC）

IEC60730 标准要求确认闪存中的数据，并且建议 CRC 为确认手段。此高速 CRC 能在初始设定（初始化）程序中检查整个代码闪存区。

高速 CRC 停止 CPU 的运行并且通过 1 个时钟从闪存读 32 位数据进行运算。因此，其特点是完成检查的时间较短

CRC 生成多项式对应 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。

以 bit31—bit0 的 MSB 优先进行运算。

25.3.1.1 闪存 CRC 控制寄存器（CRC0CTL）

这是设定高速 CRC 运算器的运行控制和运算范围的寄存器。通过 8 位存储器操作指令设定 CRC0CTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	CRC0EN	高速 CRC 运算器的运行控制 0: 停止运行 1: 通过执行 WFE 指令开始运算	0
6	-	保留	0
5:0	FEA	高速 CRC 的演算范围 00000: 00000H ~1FFBH(8K-4byte) 00001: 00000H ~3FFBH(16K-4byte) 00010: 00000H ~5FFBH(24K-4byte) 00011: 00000H ~7FFBH(32K-4byte) 其他: 保留	0x0

注 1: bit6、bit5、bit4、bit3，bit2 必须设置为 0。

注 2: 必须事先将用于比较的 CRC 运算结果期待值存入闪存的最后 4 字节，因此运算范围为减去 4 字节的范围。

25.3.1.2 闪存 CRC 运算结果寄存器 L (PGCRCL)

这是保存高速 CRC 运算结果的寄存器。

通过 16 位存储器操作指令设定 PGCRCL 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

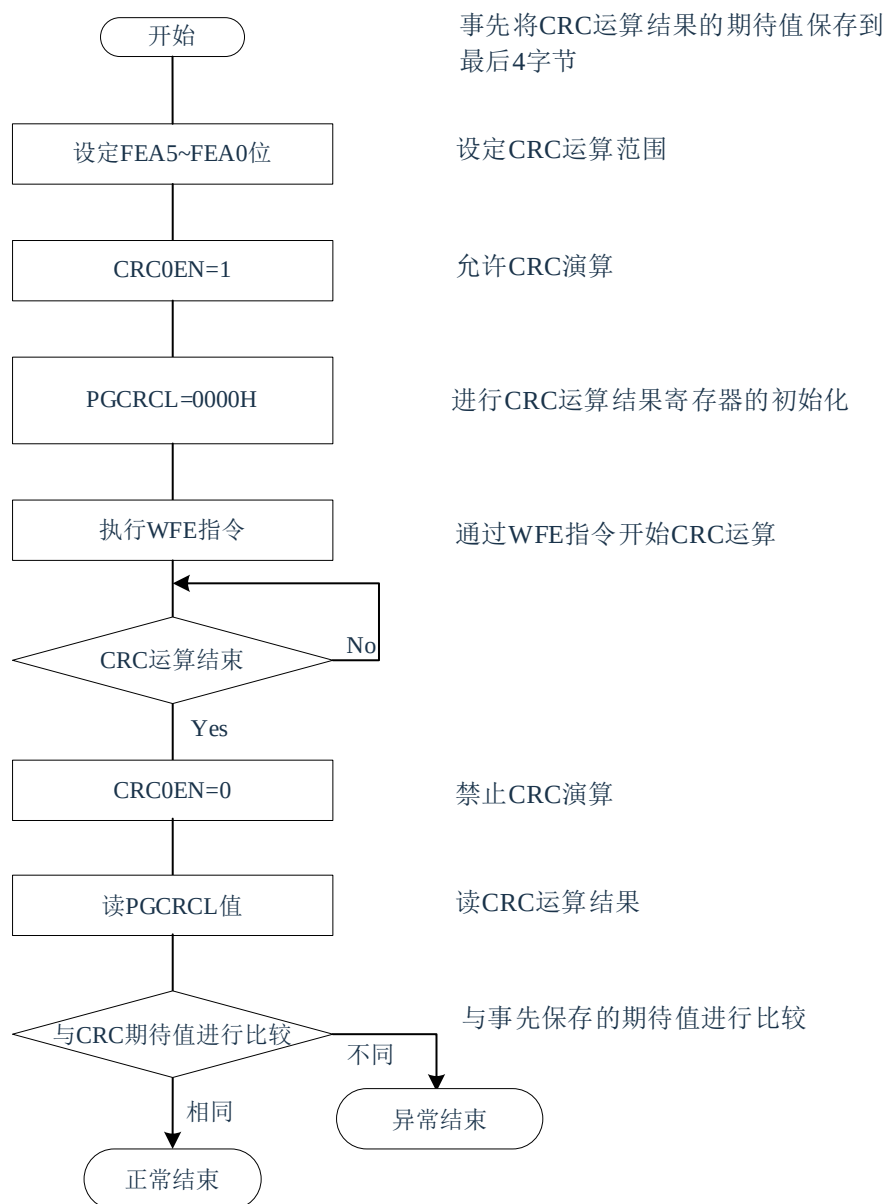
位	符号	描述	复位值
15:0	PGCRCL	保存高速 CRC 运算结果 (0000H ~ FFFFH)	0x0

注：只有在 CRC0EN（CRC0CTL 寄存器的 bit7）位为“1”时才能写 PGCRCL 寄存器。

闪存 CRC 运算功能（高速 CRC）的流程图如图 25-1 所示。

<操作流程>

图 25-1：闪存 CRC 运算功能（高速 CRC）的流程图



注 1：只以代码闪存为 CRC 运算的对象。

注 2：必须将 CRC 运算的期待值保存在代码闪存中的运算范围后的区域。

25.3.2 SFR 保护功能

为了必须保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护重要的 SFR，使其免遭改写。SFR 保护功能用于保护端口功能、时钟控制功能和电压检测电路的控制寄存器的数据。

如果设定为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

25.3.2.1 SFR 保护控制寄存器（SFRGD）

此寄存器控制 SFR 保护功能是否有效。

SFR 保护功能使用 GPORT 位和 GCSC 位。

通过 8 位存储器操作指令设定 SFRGD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:3	保留	-	-
2	GPORT	端口功能的控制寄存器的保护 0: 无效，能读写端口功能的控制寄存器。 1: 有效，端口功能的控制寄存器的写操作无效，能读。 [被保护的 SFR]PM _{xx} 、PU _{xx} 、PD _{xx} 、POM _{xx} 、 PMC _{xx} 、PS _{xx} ^注 。	0
1	-	保留	-
0	GCSC	时钟控制功能、电压检测电路的控制寄存器的保护 0: 无效，能读写时钟控制功能和电压检测电路的控制寄存器。 1: 有效。时钟控制功能和电压检测电路的控制寄存器的写操作无效，能读。 [被保护的 SFR]CMC、CSC、OSTS、CKC、PER _x 、 OSMC、LVIM、LVIS。	0

注：不保护 P_{xx}（端口寄存器）。

25.3.3 A/D 测试功能

IEC60730 标准要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (AN)、内部基准电压进行 A/D 转换，确认 A/D 转换器是否正常运行。

能通过以下步骤确认模拟多路转换器：

- 1) 通过 ADS 寄存器选择 ANIx 引脚作为 A/D 转换对象。
- 2) 对 ANx 引脚进行 A/D 转换 (转换结果 1-1)。
- 3) 通过 ADS 和 ADICHS 寄存器选择 A/D 转换器的负 (-) 基准电压作为 A/D 转换对象。
- 4) 对 A/D 转换器的负 (-) 基准电压进行 A/D 转换 (转换结果 2-1)。
- 5) 通过 ADS 寄存器选择 ANIx 引脚作为 A/D 转换对象。
- 6) 对 ANx 引脚进行 A/D 转换 (转换结果 1-2)。
- 7) 通过 ADS 和 ADICH 寄存器选择 A/D 转换器的正 (+) 基准电压作为 A/D 转换对象。
- 8) 对 A/D 转换器的正 (+) 基准电压进行 A/D 转换 (转换结果 2-2)。
- 9) 通过 ADS 寄存器选择 ANIx 引脚作为 A/D 转换对象。
- 10) 对 ANx 引脚进行 A/D 转换 (转换结果 1-3)。
- 11) 确认“转换结果 1-1”、“转换结果 1-2”和“转换结果 1-3”相同。
- 12) 确认“转换结果 2-1”的 A/D 转换结果全部为“0”并且“转换结果 2-2”的 A/D 转换结果全部为“1”。通过以上步骤，能选择模拟多路转换器以及确认布线没有断线。

注 1：在 1) ~ 10) 的转换过程中，如果模拟输入电压可变，就必须采用其他方法来确认模拟多路转换器。

注 2：转换结果含有误差，因此必须在比较转换结果时要适当考虑误差。

25.3.3.1 A/D 输入通道选择寄存器

通过 ADS 和 ADICHS 寄存器选择 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (ANxx)、内部基准电压 (0.8V) 作为 A/D 转换对象。

当用作 A/D 测试功能时，进行以下的设定：

- 在测量零刻度时，选择负 (-) 基准电压作为 A/D 转换对象。
- 在测量满刻度时，选择正 (+) 基准电压作为 A/D 转换对象。

25.3.4 输入/输出引脚的数字输出信号电平检测功能

IEC60730 标准要求确认 I/O 功能是否正常。

输入/输出引脚的数字输出信号电平检测功能在引脚为输出模式时，能读引脚的数字输出电平。

25.3.4.1 端口模式选择寄存器（PMS）

此寄存器选择在引脚为输出模式（端口模式寄存器（PMm）的 PMmn 位为“0”）时是读端口的输出锁存器的值还是读引脚的输出电平。

通过 8 位存储器操作指令设定 PMS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	-	保留	-
0	PMS0	在引脚为输出模式时读数据的选择 0: 读 Pmn 寄存器的值。 1: 读引脚的数字输出电平。	0

注：m=A、B、F，n=0~15。

25.3.5 产品唯一身份标识寄存器

产品唯一的身份标识非常适合：

用来作为序列号(例如 USB 字符序列号或者其他的终端应用)

用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器内的安全性。

用来激活带安全机制的自举过程

128 位的产品唯一身份标识所提供的参考号码对任意一个微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

产品唯一身份标识寄存器 0 (UID0)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[31:0]位，其值在出厂时编写	-

产品唯一身份标识寄存器 1 (UID1)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[63:32]位，其值在出厂时编写	-

产品唯一身份标识寄存器 2 (UID2)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[95:64]位，其值在出厂时编写	-

产品唯一身份标识寄存器 3 (UID3)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[127:96]位，其值在出厂时编写	-

第26章 选项字节

26.1 选项字节的功能

芯片的闪存 000C0H~000C3H, 500004H 为选项字节区。

选项字节由用户选项字节（000C0H~000C2H）和闪存数据保护选项字节（000C3H, 500004H）构成。在接通电源或者复位启动时，自动参照选项字节进行指定功能的设定。在使用本产品时，必须通过选项字节进行以下功能的设定。对于没有配置功能的位，不能更改初始值。

注意：是否使用各功能无关，必须设定选项字节。

26.1.1 用户选项字节（000C0H~000C2H）

1. 000C0H

IWDG 硬件使能控制

2. 000C1H

LVD 运行模式的设定

- 中断&复位模式
- 复位模式
- 中断模式
- LVD 为 OFF（使用 RESETB 引脚的外部复位输入）。

LVD 检测电平（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）的设定

注 1：当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

注 2：工作电压范围取决于用户选项字节（000C2H）的设定。

3. 000C2H

高速内部振荡器的频率设定

- 从 1.5MHz~48MHz 中选择。

26.2 闪存数据保护选项字节（000C3H，500004H）

片上调试时闪存数据保护的控制

Level0：允许通过 debugger 对闪存数据进行读出/写入/擦除操作

Level1：允许通过 debugger 对闪存数据进行 chip 全擦除操作，不允许进行读写操作。

Level2：不允许通过 debugger 对闪存数据进行操作。

26.3 寄存器映射

寄存器	地址	读/写	描述	复位值
选项字节 0	0x00C0H	R/W	看门狗控制寄存器	0xFF
选项字节 1	0x00C1H	R/W	LVD 状态控制寄存器	0xFF
选项字节 2	0x00C2H	R/W	高速内部振荡器频率控制寄存器	0xE6
选项字节 3	0x00C3H	R/W	片上调试闪存数据保护控制寄存器 1	0xFF
选项字节 4	0x500004H	R/W	片上调试闪存数据保护控制寄存器 2	0xFF

26.4 用户选项字节

26.4.1 用户选项字节（000C0H）

位	符号	描述	复位值
7:4	--	保留	0xF
3:0	IWDG_SW	IWDG 硬件使能控制 =A/F: 由软件控制 IWDG 计数使能 =其他: CONFIG 加载生效后, 强制使能 IWDG 计数	0xF

26.4.2 用户选项字节（000C1H）

位	符号	描述	复位值
7:5	VPOC[2:0]	检测电压设定	0x7
4	-	保留（须为 1）	1
3:2	LVIS[1:0]	检测电压设定	0x2
1:0	LVIMDS[1:0]	模式选择 10: 中断&复位模式 11: 复位模式 01: 中断模式	0x3

● LVD 检测电压设定（中断&复位模式）

检测电压			选项字节的设定值						
V _{LVDH}		V _{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降	下降						LVIMDS1	LVIMDS0
1.85V	1.8V	1.8V	0	0	1	1	1	1	0
1.95V	1.9V					1	0		
2.05V	2.0V					0	1		
3.15V	3.1V					0	0		
2.50V	2.45V	2.45V		1	0	1	1		
2.60V	2.55V					1	0		
2.70V	2.65V					0	1		
3.70V	3.65V					0	0		
2.80V	2.75V	2.75V		1	1	1	1		
2.90V	2.85V					1	0		
3.00V	2.95V					0	1		
4.05V	4.0V					0	0		
—			禁止设定上述以外的值。						

注 1: 必须给 bit4 写“1”。

注 2: 有关 LVD 电路的详细内容, 请参照“第 24 章 电压检测电路”。

注 3: 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

• LVD 检测电压设定（复位模式）

检测电压		选项字节的设定值						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
1.85V	1.8V	0	0	1	1	1	1	1
1.95V	1.9V		0	1	1	0		
2.05V	2.0V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.60V	2.55V		1	0	1	0		
2.70V	2.65V		1	0	0	1		
2.80V	2.75V		1	1	1	1		
2.90V	2.85V		1	1	1	0		
3.00V	2.95V		1	1	0	1		
3.15V	3.1V		0	1	0	0		
3.70V	3.65V		1	0	0	0		
4.05V	4.0V		1	1	0	0		
—		禁止设定上述以外的值。						

- 注 1：必须给 bit4 写“1”。
- 注 2：有关 LVD 电路的详细内容，请参照“第 24 章 电压检测电路”。
- 注 3：检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

● LVD 检测电压设定（中断模式）

检测电压		选项字节的设定值						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
1.85V	1.8V	0	0	1	1	1	0	1
1.95V	1.9V		0	1	1	0		
2.05V	2.0V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.60V	2.55V		1	0	1	0		
2.70V	2.65V		1	0	0	1		
2.80V	2.75V		1	1	1	1		
2.90V	2.85V		1	1	1	0		
3.00V	2.95V		1	1	0	1		
3.15V	3.1V		0	1	0	0		
3.70V	3.65V		1	0	0	0		
4.05V	4.0V		1	1	0	0		
—		禁止设定上述以外的值。						

注 1：必须给 bit4 写“1”。

注 2：有关 LVD 电路的详细内容，请参照“第 24 章 电压检测电路”。

注 3：检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

● LVD 为 OFF 时的设定（使用 RESETB 引脚的外部复位输入）

检测电压		选项字节的设定值						
VLVDH		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值。						

注意 1：必须给 bit4 写“1”。

注意 2：当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H）的设定。

注 1：×：忽略。

注 2：有关 LVD 电路的详细内容，请参照“第 24 章 电压检测电路”。

注 3：检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

26.4.3 用户选项字节（000C2H）

位	符号	描述	复位值
7:5	-	保留（须为 1）	0x7
4:0	FRQSE[4:0]	高速内部震荡器时钟频率选择	0x06

FRQSE4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率	
					f _{HOCO}	f _{IH}
0	0	0	0	0	48MHz	48MHz
0	0	0	0	1	48MHz	24MHz
0	0	0	1	0	48MHz	12MHz
0	0	0	1	1	48MHz	6MHz
0	0	1	0	0	48MHz	3MHz
0	0	1	0	1	48MHz	1.5MHz
上述以外					禁止设定。	

- 注 1：必须给 bit7~5 写“1”。
- 注 2：工作频率范围和工作电压范围因闪存各运行模式而不同。详细内容请参照数据手册的 AC 特性。

26.4.4 闪存数据保护选项字节（000C3H）

位	符号	描述	复位值
7:0	OCDEN[7:0]	闪存数据保护的 control	0xFF

26.4.5 闪存数据保护选项字节（500004H）

位	符号	描述	复位值
7:0	OCDM[7:0]	闪存数据保护的 control	0xFF

OCDM	OCDEN	闪存数据保护的 control
3C	C3	不允许通过 debugger 对闪存数据进行操作。
3C 以外的值	C3	允许通过 debugger 对闪存数据进行 chip 全擦除操作，不允许进行读写操作。
上记以外		允许通过 debugger 对闪存数据进行读出/写入/擦除操作

注：50_0004H 地址属于数据闪存区，如果使用该地址做数据存储用，需先确定数值不会引起保护选项的误设。

第27章 FLASH 控制

27.1 FLASH 控制功能描述

本制品包含一颗 32KByte 容量的 FLASH 存储器，共划分为 64 个 Sector，每个 Sector 容量为 512Byte。可做
为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。

27.2 FLASH 存储器结构



27.3 寄存器映射

(FLASH 控制基地址 = 0x4002_0000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
FLKEY	0x004	WO	FLASH 密钥寄存器	0x0
FLSR	0x00C	R/W	FLASH 状态寄存器	0x0
FLPRT	0x010	R/W	FLASH 保护寄存器	0x0
FLCMD	0x014	R/W	FLASH 操作控制寄存器	0x0
FLADDR	0x018	R/W	FLASH 地址寄存器	0x00ffffff
FLDATA0	0x01C	R/W	FLASH 数据寄存器 0	0xffffffff
FLCR	0x024	R/W	FLASH 控制寄存器	0x80000300
FLREGION	0x048	R/W	FLASH 区域控制寄存器	0x0

27.4 寄存器说明

27.4.1 FLASH 操作控制寄存器(FLCMD)

Flash 操作控制寄存器用于对 FLASH 操作进行选择。

位	符号	描述	复位值
31:8	-	保留	-
7:0	CMD	操作选择: 56H: CHIP 擦除 5CH: Erase without retry AAH: 没有预写入的写入	0x0

27.4.2 FLASH 密钥寄存器(FLKEY)

借助 Flash 密钥寄存器，可允许对 Flash 控制寄存器的访问，进而允许执行编程和擦除操作。

位	符号	描述	复位值
31:0	KEY	FLASH 寄存器解锁并允许对其执行编程/擦除操作必须顺序编程以下值: (a): KEY = 0x45670123 (b): KEY = 0xCDEF89AB	0x0

27.4.3 FLASH 数据寄存器 0(FLDATA0)

Flash 数据寄存器，将要写入的值写入该寄存器。

位	符号	描述	复位值
31:0	DATA0	FLASH 写入的数据	0xffffffff

27.4.4 FLASH 地址寄存器(FLADDR)

Flash 地址寄存器，将要写入或擦除的地址写入该寄存器。

位	符号	描述	复位值
31:24	-	保留	0x0
23:0	ADDR	写入或擦除的地址	0xffffffff

27.4.5 FLASH 控制寄存器(FLCR)

Flash 控制寄存器用于配置和启动 Flash 操作。

位	符号	描述	复位值
31	LOCK	该位只能写入 1。该位置 1 时，表示 FLASH_CR 寄存器已锁定。当检测到解锁序列时，由硬件将该位清零。如果解锁操作失败，该位仍保持置 1，直到下一次复位	1
30:18	-	保留	0
17	ERRIE	错误中断使能： 0: 禁止错误中断 1: 使能错误中断	0x0
16	EOPIE	操作结束中断使能： 0: 禁止中断 1: 使能中断	0x0
15:10	-	保留	0
9:8	PSIZE	写入数据字长： 00: X8 01: X16 10: X32 11: 禁止设定	0x11
7:1	-	保留	0x0
0	STRT	FLASH 擦写启动位： 0: 在 BSY 位清零后硬件清零 1: 触发擦除或写入操作(只能软件置 1)	0

27.4.6 FLASH 状态寄存器(FLSR)

Flash 状态寄存器提供正在执行的编程和擦除操作的相关信息。

位	符号	描述	复位值
31:17	-	保留	0x0
16	BSY	繁忙 ^{注 1} : 0: 当前未执行任何 Flash 操作 1: 正在执行 Flash 操作（该位指示 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时清零。）	0
15:7	-	保留	0x0
6	PGPERR	编程并行位数错误: 0: 编程并行位数正常 1: 代码在控制寄存器未正确配置的情况下对 Flash 执行写访问，将由硬件为该位置 1。需软件写 1 清零	0
5	-	保留	0
4	WRPERR	写保护错误 0: 写保护正常 1: 擦除/编程的地址属于 Flash 中处于写保护状态的区域，将由硬件为该位置 1。需软件写 1 清零	0
3:2	-	保留	0x0
1	OPERR	操作错误: 0: 操作正常 1: 检测到 Flash 操作（编程/擦除/读取）请求，但由于存在并行位数错误、对齐错误或写保护错误而无法运行，将由硬件对该位置 1。	0
0	EOP	操作结束: 0: 操作未完成 1: 当成功完成一个或多个 Flash 操作（编程/擦除）时，由硬件将该位置 1。需软件写 1 清零。	0

注 1: BSY 只读。

27.4.7 FLASH 区域控制寄存器（FLREGION）

Flash 区域选择寄存器用于选择 FLASH 访问区域。

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	REGION	FLASH 区域选择位: 55H: 程序闪存区 AAH: 数据闪存区	0x0

27.4.8 FLASH 保护寄存器（FLPRT）

Flash 保护寄存器用于保护程序区和数据区内容。

位	符号	描述	复位值
31	PRTALL	全空间保护控制位： 0: 全空间不写保护 1: 全空间写保护	0
30:25	-	保留	0x0
24	PRTDATA	数据区保护控制位： 0: 数据区不写保护 1: 数据区写保护	0
23:8	-	保留	0x0
7	PRTROM7	程序区 28K-32K(7000H-7FFFH)保护控制位： 0: 不写保护 1: 写保护	0
6	PRTROM6	程序区 24K-28K(6000H-6FFFH)保护控制位： 0: 不写保护 1: 写保护	0
5	PRTROM5	程序区 20K-24K(5000H-5FFFH)保护控制位： 0: 不写保护 1: 写保护	0
4	PRTROM4	程序区 16K-20K(4000H-4FFFH)保护控制位： 0: 不写保护 1: 写保护	0
3	PRTROM3	程序区 12K-16K(3000H-3FFFH)保护控制位： 0: 不写保护 1: 写保护	0
2	PRTROM2	程序区 8K-12K(2000H-2FFFH)保护控制位： 0: 不写保护 1: 写保护	0
1	PRTROM1	程序区 4K-8K(1000H-1FFFH)保护控制位： 0: 不写保护 1: 写保护	0
0	PRTROM0	程序区 0K-4K(0000H-0FFFH)保护控制位： 0: 不写保护 1: 写保护	0

27.5 FLASH 操作方法

27.5.1 编程

写入操作流程如下：

- (1) 解锁 FLASH 寄存器：按照顺序写

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; 解锁后 FLCR.LOCK=0;

- (2) 解除编程区域的保护：FLPRT.PRTROM_n=0(n=0-7)或 FLPRT.PRTDATA=0.

- (3) 设置写入模式 FLCMD=0xAA。

- (4) 设置编程区域 FLREGION。

- (5) 将目标地址写入 FLADDR。

- (6) 将写入数据写入 FLDATA0。

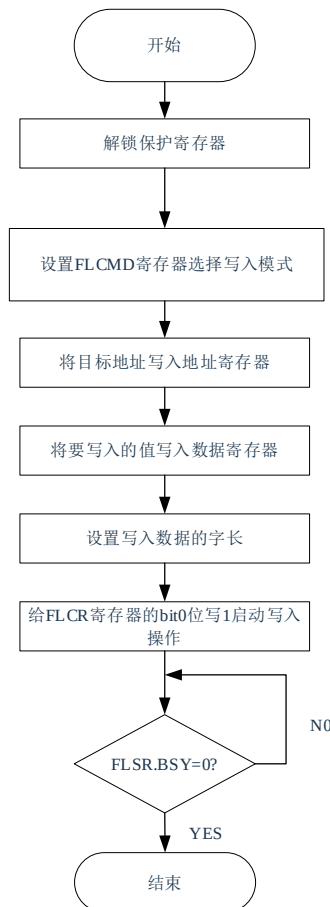
- (7) 设置要写入数据字长 FLCR.PSIZE。

- (8) 启动编程 FLCR.STRT=1。

- (9) 软件查询状态寄存器 FLSR.BSY =0 时，表示操作完成；或软件查询状态寄存器 FLSR.EOP=1，表示操作完成。

- (10) 进行下次操作前，软件置”1”来清除 FLSR.EOP。

- (11) 操作完毕后，FLCR.LOCK=1,重新锁定 FLCR。



27.5.2 扇擦除

sector 擦除，擦除时间由硬件实现。操作流程如下：

(1) 解锁 FLASH 寄存器：按照顺序写

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; 解锁后 FLCR.LOCK=0;

(2) 解除编程区域的保护：FLPRT.PRTROMn=0(n=0-7)或 FLPRT.PRTDATA=0.

(3) 设置扇擦除模式 FLCMD=0x5C。

(4) 设置编程区域 FLREGION。

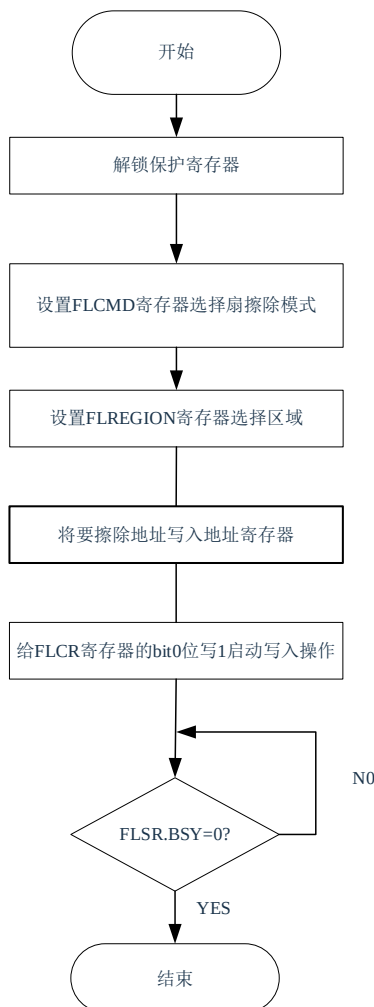
(5) 将目标地址写入 FLADDR（写擦除目标 sector 的首地址）。

(6) 启动编程 FLCR.STRT=1。

(7) 软件查询状态寄存器 FLSR.BSY =0 时，表示操作完成；或软件查询状态寄存器 FLSR.EOP=1，表示操作完成。

(8) 进行下次操作前，软件置”1”来清除 FLSR.EOP。

(9) 操作完毕后，FLCR.LOCK=1,重新锁定 FLCR。



27.5.3 全片擦除（chip erase）

chip 擦除，擦除时间由硬件实现。操作流程如下：

（1）解锁 FLASH 寄存器：按照顺序写

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; 解锁后 FLCR.LOCK=0;

（2）解除编程区域的保护：FLPRT.PRTROMn=0(n=0-7)或 FLPRT.PRTDATA=0,FLPRT.PRTALL=0。

（3）设置扇擦除模式 FLCMD=0x56。

（4）设置编程区域 FLREGION=0x55（全擦除只擦除程序闪存区）。

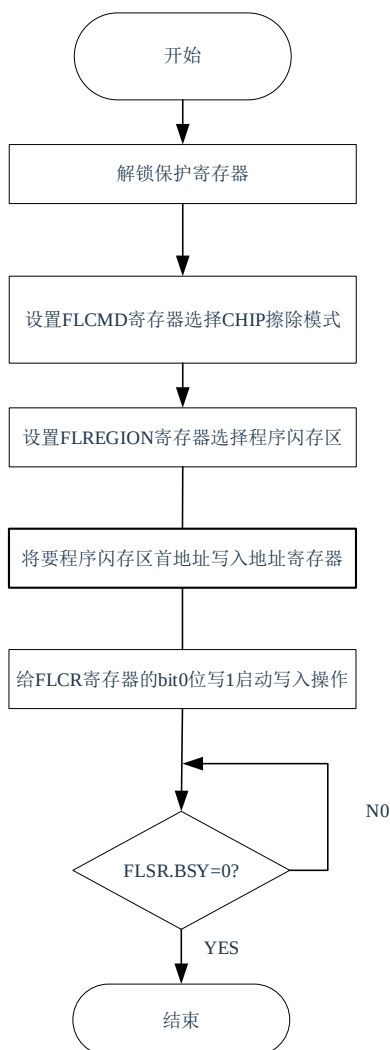
（5）将目标地址写入 FLADDR（写程序闪存区首地址）。

（6）启动编程 FLCR.STRT=1。

（7）软件查询状态寄存器 FLSR.BSY =0 时，表示操作完成；或软件查询状态寄存器 FLSR.EOP=1，表示操作完成。

（8）进行下次操作前，软件置”1”来清除 FLSR.EOP。

（9）操作完毕后，FLCR.LOCK=1,重新锁定 FLCR。



27.6 FLASH 操作的注意事项

FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。擦写参数的设置由硬件实现,无需设置参数寄存器。

如果擦写操作从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待操作完成后，继续下一条指令。如果该操作从 RAM 里执行，CPU 不会停止取指，当前可以继续下一条指令。

在 FLASH 处于编程操作中时，如果 CPU 执行进入深睡眠的指令，系统将等待编程动作结束后才会进入深睡眠。

第28章 附录 修订记录

版本	日期	修订内容
V0.9.0	2025 年 7 月	初始版本
V0.9.1	2025 年 8 月	1) 删除 FLASH 访问控制寄存器 (FLACR) 2) 修改 12.3.15 噪声滤波器允许寄存器 0 (NFEN0) 3) 修正 18.5.5 模拟比较器事件控制寄存器名字 4) 修改 TIM1_BDTR 寄存器