

译文

TMPM037FWUG

本资料是为了参考的目的由原始文档翻译而来。

使用本资料时，请务必确认原始文档关联的最新信息，并遵守其相关指示。

原本: “TMPM037FWUG” 2014-11-5

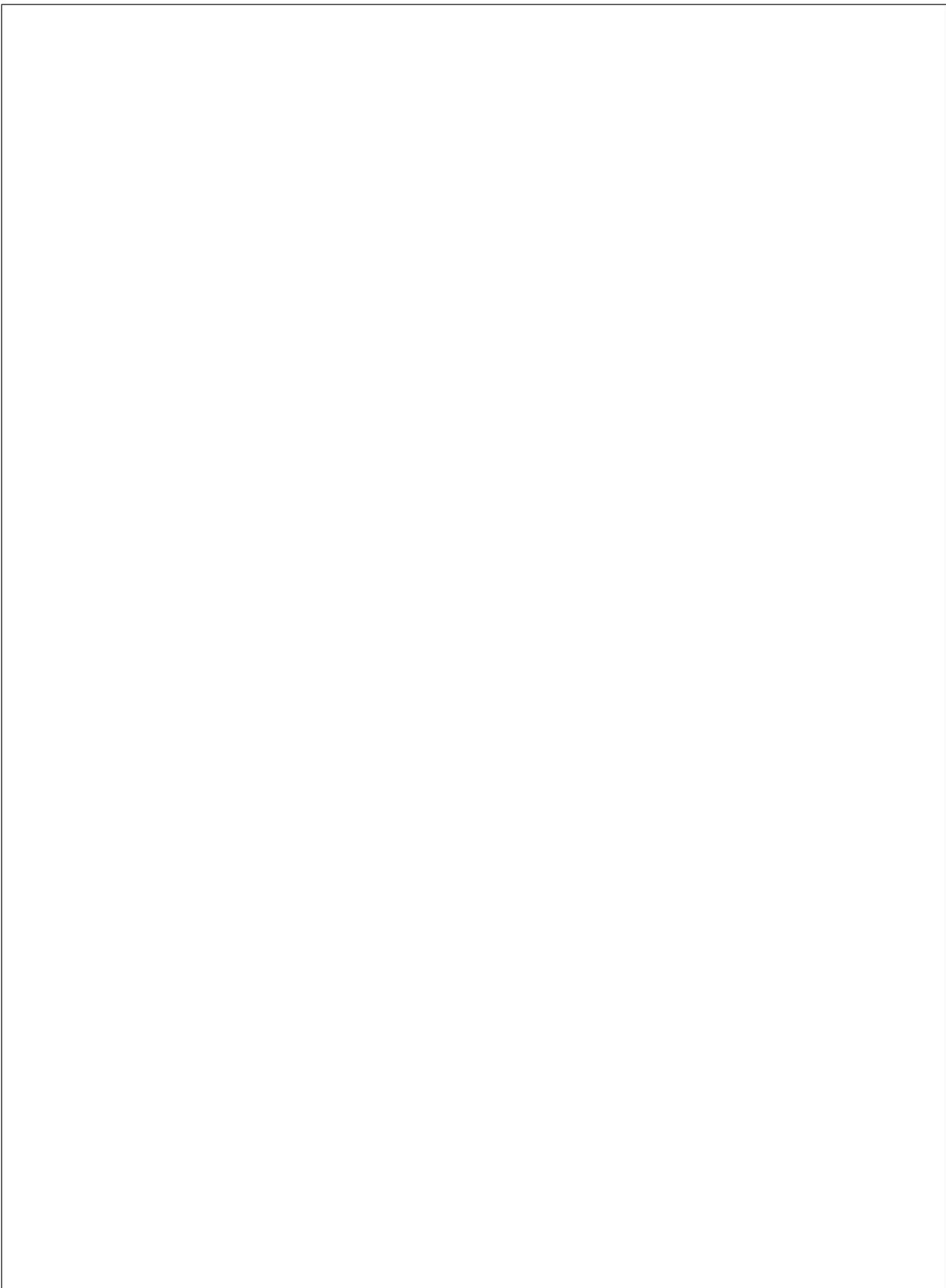
翻译日: 2015-5-11

TOSHIBA

32 位RISC微控制器
TX00系列

TMPM037FWUG

东芝公司
半导体&存储产品公司



ARM、Cortex和Thumb为ARM有限公司(或其子公司)在欧盟和/或其它地区的注册商标。版权所有，不得翻印。



概述:关于本规范中SFR(特殊功能寄存器)描述的说明

SFR(特殊功能寄存器)是一种外围电路控制寄存器(IP)。

IP的SFR地址见内存地址节次，SFR详见各IP节次。

本规范中采用的SFR的定义遵照下列规则。

a. 各IP的SFR表举例

各IP节次中的SFR表列出了寄存器名称，地址和简要说明。

所有寄存器具有一个 32-位的唯一地址。除一些例外情况外，寄存器地址定义如下：“基址 + (唯一地址)”

		基址 = 0x0000_0000
寄存器名称		地址(基+)
控制寄存器	SAMCR	0x0004
		0x000C

注：SAMCR寄存器地址为 32 位，即地址0x0000_0004(基地址(0x00000000)+唯一地址(0x0004))。

注：上述寄存器只是出于解释目的而给出的例子，不是为了证明目的。在该微控制器中，该寄存器并不存在。

b. SFR(寄存器)

每个寄存器基本上包含一个 32-位寄存器(有些例外)。

每个寄存器的说明部分包括位，比特符号，类型，复位初始值和功能等内容。

1.2.2 SAMCR(控制寄存器)

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

位	比特符号	类型	功能
31-10	-	R	能读取"0"。
9-7	MODE[2:0]	R/W	操作模式设置 000: 采样方式 0 001: 采样方式 1 010: 采样方式 2 011: 采样方式 3 非上述设置:保留
6-0	TDATA[6:0]	W	发送的数据

注:类型分为如下三类。

- R/W READ WRITE
- R READ
- W WRITE

c. 数据描述

说明中采用的符号的意义如下。

- x: 通道号/端口
- n, m: 位数

d. 寄存器描述

寄存器说明如下

- 寄存器名称 <比特符号号>
示例: SAMCR<MODE> = "000"或SAMCR<MODE[2:0]> = "000"
<MODE[2:0]>指在比特符号号模式时 位 2~位 0 (3 位宽)。
- 寄存器名称 [位]
示例:SAMCR[9:7] = "000"
它指寄存器SAMCR的位 9~位 7 (32 位宽)。

译文

修订记录

日期	修订版	备注
2014/11/5	1	第一次发布

目录

概述:关于本规范中SFR（特殊功能寄存器）描述的说明

TMPM037FWUG	1
1.1 特点	1
1.2 方块图	4
1.3 引脚分配(顶视图)	5
1.4 引脚名称与功能	6
1.4.1 各外设功能，控制引脚和电源引脚的名称与功能	6
1.4.1.1 外设功能	6
1.4.1.2 调试功能	6
1.4.1.3 控制功能	7
1.4.1.4 电源引脚	7
1.4.2 TMPM037FWUG 的引脚名称与功能	8
1.4.2.1 引脚名称与功能列表的详细数据	8
1.4.2.2 端口/调试引脚	9
1.4.2.3 控制引脚	11
1.4.2.4 电源引脚	11
2. 产品信息	12
2.1 各外设功能相关信息	13
2.1.1 DMA 控制器(DMAC)	13
2.1.1.1 DMA 请求表	13
2.1.1.2 用外设至外设传输支持的外设功能	14
2.1.1.3 DMA 请求控制寄存器(DMARQCTL)	14
2.1.1.4 DMACREdge(DMAC 请求设置寄存器)	15
2.1.1.5 DMACRCLR(DMAC 请求清除寄存器)	15
2.1.2 16-位定时器/事件计数器(TMRB)	15
2.1.3 16-位定时器 A(TMR16A)	16
2.1.4 串行通道(SIO/UART)	16
2.1.5 I2C 总线(I2C)	17
2.1.6 模拟/数字转换器(ADC)	17
2.1.7 调试接口	18
3. 处理器内核	19
3.1 处理器内核相关信息	19
3.2 可配置选项	19
3.3 异常/中断	20
3.3.1 中断输入的数目	20
3.3.2 SysTick	20
3.3.3 SYSRESETREQ	20
3.3.4 LOCKUP	20
3.4 事件	20
3.5 电源管理	20
4. 内存地址图	23
4.1 内存地址	23
4.2 总线矩阵	25
4.2.1 结构	26
4.2.1.1 单芯片模式	26
4.2.1.2 单启动模式	26
4.2.2 连接表	27

4.2.2.1 代码区/SRAM 区	27
4.2.2.2 外设区	28
4.2.3 外设功能的地址列表	29
5. 复位操作	30
5.1 冷复位	31
5.1.1 通过 RESET 引脚实现冷复位	31
5.1.2 用上电复位电路实现冷复位	32
5.2 热复位	32
5.3 复位后	32
6. 时钟/模式控制	34
6.1 特点	34
6.2 寄存器	35
6.2.1 寄存器列表	35
6.2.2 CGSYSCR(系统控制寄存器)	36
6.2.3 CGOSCCR(振荡控制寄存器)	37
6.2.4 CGSTBYCR(待机控制寄存器)	39
6.2.5 CGPLLSEL(PLL 选择寄存器)	40
6.2.6 CGPROTECT (保护寄存器)	41
6.3 时钟控制	42
6.3.1 时钟型	42
6.3.2 复位后初始值	42
6.3.3 时钟系统图	43
6.3.4 预热功能	44
6.3.5 时钟倍增电路(PLL)	46
6.3.5.1 如何配置 PLL 功能	46
6.3.5.2 PLL 设置的顺序	47
6.3.6 系统时钟	48
6.3.6.1 时钟设置	49
6.3.6.2 在使用外部振荡器时	50
6.3.7 预分频器时钟控制	50
6.4 模式与模式转换	51
6.4.1 模式转换	51
6.5 操作模式	52
6.5.1 NORMAL 模式	52
6.6 低功耗模式	52
6.6.1 IDLE 模式	52
6.6.2 STOP1 模式	53
6.6.3 低功耗模式设置	54
6.6.4 各模式下的操作状态	55
6.6.5 解除低功耗模式	56
6.6.6 预热	57
6.6.7 模式转换时的时钟操作	58
6.6.7.1 运行模式的转换: NORMAL → STOP1 → NORMAL	58
6.6.8 转换为低功耗模式时的注意事项	59
6.6.8.1 MCU 进入 IDLE 或 STOP1 模式时的情况	59
7. 异常	60
7.1 概述	60
7.1.1 异常类型	60
7.1.2 处理流程图	61
7.1.2.1 异常请求与检测	62
7.1.2.2 异常处理和分支转移至中断服务程序(预清空)	63
7.1.2.3 执行中断服务程序(ISR)	64
7.1.2.4 异常出口	64
7.2 复位异常	66
7.3 非屏蔽中断(NMI)	67
7.4 SysTick	67
7.5 中断	68

7.5.1	中断源	68
7.5.1.1	中断路径	68
7.5.1.2	生成	69
7.5.1.3	发送	69
7.5.1.4	使用外部中断引脚时的注意事项	69
7.5.2	中断源列表	70
7.5.2.1	作用电平	71
7.5.3	中断处理	72
7.5.3.1	流程图	72
7.5.3.2	准备工作	73
7.5.3.3	通过时钟发生器进行检测	75
7.5.3.4	通过 CPU 进行检测	75
7.5.3.5	CPU 处理	76
7.5.3.6	中断服务例行程序(ISR)	76
7.6	与异常/中断相关的寄存器	77
7.6.1	寄存器表	77
7.6.2	NVIC 寄存器	78
7.6.2.1	SysTick 控制与状态寄存器	78
7.6.2.2	SysTick 重新加载值寄存器	79
7.6.2.3	SysTick 正确值寄存器	79
7.6.2.4	SysTick 校准值寄存器	80
7.6.2.5	中断设置启用寄存器 1	81
7.6.2.6	中断清除使能寄存器 1	82
7.6.2.7	中断设置挂起寄存器 1	83
7.6.2.8	中断清除挂起寄存器 1	84
7.6.2.9	中断优先权寄存器	85
7.6.2.10	应用中断和复位控制寄存器	86
7.6.2.11	系统处理程序优先级寄存器	87
7.6.2.12	系统处理程序控制和状态寄存器	88
7.6.3	时钟生成器寄存器	89
7.6.3.1	CGIMCGA(CG 中断模式控制寄存器 A)	89
7.6.3.2	CGIMCGB(CG 中断模式控制寄存器)	91
7.6.3.3	CGICRCG(C 中断请求清除寄存器)	92
7.6.3.4	CGRSTFLG(复位标志寄存器)	93
7.6.3.5	CGNMIFLG(NMI 标志寄存器)	95
8	DMA 控制器(DMAC)	96
8.1	概述	96
8.2	DMA 传送类型	97
8.3	方块图	98
8.4	寄存器描述	99
8.4.1	DMAC 寄存器列表	99
8.4.2	DMACxIntStatus (DMAC 中断状态寄存器)	100
8.4.3	DMACxIntTCStatus (DMAC 中断端子计数状态寄存器)	101
8.4.4	DMACxIntTCClear (DMAC 中断端子计数清除寄存器)	102
8.4.5	DMACxIntErrorStatus (DMAC 中断错误状态寄存器)	103
8.4.6	DMACxIntErrClr (DMAC 中断错误清除寄存器)	104
8.4.7	DMACxRawIntTCStatus (DMAC 原中断端子计数状态寄存器)	105
8.4.8	DMACxRawIntErrorStatus (DMAC Raw 错误中断状态寄存器)	106
8.4.9	DMACxEnbldChns (DMAC 使能通道寄存器)	107
8.4.10	DMACxSoftBReq (DMAC 软件脉冲串请求寄存器)	108
8.4.11	DMACxSoftSReq (DMAC 软件单个请求寄存器)	110
8.4.12	DMACx 配置(DMAC 配置寄存器)	112
8.4.13	DMACxCnSrcAddr (DMAC 通道 x 源地址寄存器)	113
8.4.14	DMACxCnDestAddr (DMAC 通道 x 目的地址寄存器)	114
8.4.15	DMACxCnLLI (DMAC 通道 x 链接表项目寄存器)	115
8.4.16	DMACxCnControl (DMAC 通道 n 控制寄存器)	116
8.4.17	DMACxCnConfiguration (DMAC 通道 n 配置寄存器)	118
8.5	特殊功能	120
8.5.1	分散/聚集功能	120
8.5.2	链接表操作	121
9	输入/输出端口	123
9.1	寄存器	123
9.1.1	寄存器列表	124

9.1.2 端口功能和设置列表	125
9.1.2.1 PORT A	126
9.1.2.2 PORT B	127
9.1.2.3 PORT C	128
9.1.2.4 PORT D	129
9.1.2.5 PORT E	130
9.1.2.6 PORT F	131
9.1.2.7 PORT G	132
9.1.3 端口的方块图	133
9.1.3.1 端口类型	133
9.1.3.2 类型 FT1	134
9.1.3.3 类型 FT4	135
9.1.3.4 类型 FT5	136
9.1.3.5 类型 FT6	137
10.16 位计时器/事件计数器(TMRB)	138
10.1 概要	138
10.2 方块图	139
10.3 寄存器	140
10.3.1 寄存器列表	140
10.3.2 TBxEN(启用寄存器)	141
10.3.3 TBxRUN(RUN 寄存器)	142
10.3.4 TBxCR(控制寄存器)	143
10.3.5 TBxMOD(模式寄存器)	144
10.3.6 TBxFFCR(触发器控制寄存器)	145
10.3.7 TBxST(状态寄存器)	146
10.3.8 TBxIM(中断屏蔽寄存器)	147
10.3.9 TBxUC(上升计数器捕获寄存器)	148
10.3.10 TBxRG0(计时器寄存器 0)	149
10.3.11 TBxRG1(计时器寄存器 1)	149
10.3.12 TBxCP0(捕获寄存器 0)	150
10.3.13 TBxCP1(捕获寄存器 1)	150
10.3.14 TBxDMA(DMA 请求启用寄存器)	151
10.4 操作描述	152
10.4.1 预分频器	152
10.4.2 上升计数器(UC)	152
10.4.2.1 源时钟	152
10.4.2.2 计数器启动/停止	152
10.4.2.3 计数器清除	153
10.4.2.4 递增计数器溢出	153
10.4.3 计时器寄存器(TBxRG0, TBxRG1)	153
10.4.4 捕获控制	153
10.4.5 捕获寄存器(TBxCP0, TBxCP1)	153
10.4.6 上升计数器捕获寄存器(TBxUC)	154
10.4.7 比较器(CP0, CP1)	154
10.4.8 计时器触发器(TBxFF0)	154
10.4.9 捕获中断(INTTBxCAP0, INTTBxCAP1)	154
10.4.10 DMA 请求	154
10.5 各模式操作描述	155
10.5.1 间隔计时器模式	155
10.5.2 事件计数器模式	155
10.5.3 可编程脉冲发生(PPG)输出模式	156
10.5.4 可编程脉冲发生(PPG)外部触发脉冲输出模式	158
10.6 捕获功能应用	160
10.6.1 频率测量	160
10.6.2 脉冲宽度测量	162
11.16 位计时器 A (TMR16A Ver.B)	164
11.1 概要	164
11.2 方块图	164
11.3 寄存器	165
11.3.1 寄存器列表	165
11.3.1.1 T16AxEN(启用寄存器)	166

11.3.1.2 T16AxRUN(RUN 寄存器).....	166
11.3.1.3 T16AxCR(控制寄存器).....	167
11.3.1.4 T16AxRG(计时器寄存器).....	168
11.3.1.5 T16AxCP(捕获寄存器).....	168
11.4 操作描述	169
11.4.1 计时器操作	169
11.4.2 T16AxOUT 控制.....	169
11.4.3 读取捕获.....	169
11.4.4 自动停止.....	169
12. 使用 4 字节 FIFO 的串行的通道(SIO/UART).....	171
12.1. 概述	171
12.2. 配置	172
12.3 寄存器描述	173
12.3.1 寄存器列表	173
12.3.2 SCxEN(启用寄存器).....	174
12.3.3 SCxBUF(缓冲寄存器).....	175
12.3.4 SCxCR(控制寄存器).....	176
12.3.5 SCxMOD0(模式控制寄存器 0).....	178
12.3.6 SCxMOD1(模式控制寄存器 1).....	179
12.3.7 SCxMOD2(模式控制寄存器 2).....	180
12.3.8 SCxBRCR(波特率发生器控制寄存器).....	182
12.3.9 SCxBRADD(波特率发生器控制寄存器 2).....	183
12.3.10 SCxFCNF(FIFO 配置寄存器).....	184
12.3.11 SCxRFC(接收 FIFO 配置寄存器).....	186
12.3.12 SCxTFC(发送 FIFO 配置寄存器).....	187
12.3.13 SCxRST(接收 FIFO 状态寄存器).....	188
12.3.14 SCxTST(发送 FIFO 状态寄存器).....	189
12.3.15 SCxDMA(DMA 请求启用寄存器).....	190
12.4 各模式下的操作	191
12.5 数据格式	192
12.5.1 数据格式列表.....	192
12.5.2 奇偶控制.....	193
12.5.2.1 发送.....	193
12.5.2.2 接收.....	193
12.5.3 STOP 位长度.....	193
12.6 时钟控制	194
12.6.1 预分频器.....	194
12.6.2 串行时钟生成电路.....	194
12.6.2.1 波特率发生器.....	194
12.6.2.2 时钟选择电路.....	196
12.6.3 发送接收缓冲器和 FIFO.....	198
12.6.3.1 配置.....	198
12.6.3.2 发送/接收缓冲器.....	198
12.6.3.3 初始化发送缓冲器.....	199
12.6.3.4 FIFO.....	199
12.7 状态标志	200
12.8 错误标志	200
12.8.1 OERR 标志	200
12.8.2 PERR 标志.....	201
12.8.3 FERR 标志.....	201
12.9 接收	202
12.9.1 接收计数器.....	202
12.9.2 接收控制单元.....	202
12.9.2.1 I/O 接口模式.....	202
12.9.2.2 UART 模式.....	202
12.9.3 接收操作.....	202
12.9.3.1 接收缓冲器.....	202
12.9.3.2 接收 FIFO 操作.....	203
12.9.3.3 I/O 接口模式 (带有时钟输出模式).....	204
12.9.3.4 读取接收数据.....	205
12.9.3.5 唤醒功能.....	205
12.9.3.6 超程错误.....	205

12.10 发送	206
12.10.1 发送计数器	206
12.10.2 发送控制	206
12.10.2.1 在 I/O 接口模式下	206
12.10.2.2 在 UART 模式下	206
12.10.3 发送操作	207
12.10.3.1 发送缓冲器的操作	207
12.10.3.2 发送 FIFO 操作	208
12.10.3.3 在 I/O 接口模式(带有时钟输出模式)下发送	209
12.10.3.4 I/O 接口模式下最后一位输出后的 SCxTXD 引脚电平	209
12.10.3.5 欠载运行错误	210
12.10.3.6 在 I/O 接口模式(带有时钟输入模式)下的数据保持时间	210
12.11 握手功能	211
12.12 中断/错误生成定时	212
12.12.1 接收中断	212
12.12.1.1 单缓冲器/双缓冲器	212
12.12.1.2 FIFO	212
12.12.2 发送中断	213
12.12.2.1 单缓冲/双缓冲	213
12.12.2.2 FIFO	214
12.12.3 错误生成	214
12.12.3.1 UART 模式	214
12.12.3.2 I/O 接口模式	214
12.13 DMA 请求	215
12.13.2 使用 FIFO	215
12.14 软件复位	216
12.15 各模式下的操作	217
12.15.1 模式 0 (I/O 接口模式)	217
12.15.1.1 发送	217
12.15.1.2 接收	221
12.15.1.3 发送和接收(全双工)	224
12.15.2 模式 1 (7 位 UART 模式)	228
12.15.3 模式 2 (8 位 UART 模式)	228
12.15.4 模式 3 (9 位 UART 模式)	229
12.15.4.1 唤醒功能	229
12.15.4.2 协议	230
13. I2C 总线接口	231
13.1 配置	232
13.2 I2C 总线模式	233
13.2.1 I2C 总线模式数据格式	233
13.3 寄存器	235
13.3.1 各通道的寄存器	235
13.3.2 I2CxCR1(控制寄存器 1)	235
13.3.3 I2CxDBR(串行总线接口数据缓冲寄存器)	238
13.3.4 I2CxAR(I2C 总线地址寄存器)	239
13.3.5 I2CxCR2(控制寄存器 2)	240
13.3.6 I2CxSR(状态寄存器)	241
13.3.7 I2CxPRS(预分频器时钟设置寄存器)	242
13.3.8 I2CxIE(中断启用寄存器)	243
13.3.9 I2CxIR(中断寄存器)	243
13.4 在 I2C 总线模式下的控制	244
13.4.1 串行时钟	244
13.4.1.1 时钟源	244
13.4.1.2 时钟同步	246
13.4.2 从机设备地址匹配检测或一般调用检测的选择	246
13.4.3 设置确认模式	247
13.4.4 设置每次传输的位数	247
13.4.5 从机设备地址和地址识别模式	247
13.4.6 将 I2C 设置为主机设备或从机设备	248
13.4.7 将 I2C 配置为发送器或接收器	248
13.4.8 生成启动和停止条件	249
13.4.9 中断服务请求和解除	251

13.4.10 I2C 总线模式	251
13.4.11 软件复位	251
13.4.12 仲裁丢失检测监控器	251
13.4.13 从机设备地址匹配检测监控程序	253
13.4.14 一般调用检测监控程序	253
13.4.15 最后接收位监控程序	253
13.4.16 数据缓冲缓冲器 (I2CxDBR)	254
13.5 I2C 总线模式下的数据传输规程	255
13.5.1 装置初始化	255
13.5.2 生成启动条件和从设备地址	255
13.5.2.1 主设备模式	255
13.5.2.2 从设备模式	256
13.5.3 传输一个数据字	258
13.5.3.1 主设备模式(<MST> = "1")	258
13.5.3.2 从设备模式(<MST> = "0")	262
13.5.4 生成停止条件	264
13.5.5 重新启动规程	264
13.6 使用注意事项	266
13.6.1 软件复位后的寄存器值	266
14. 10-位模拟/数字转换器 (ADC)	267
14.1 概要	267
14.2 配置	267
14.3 寄存器	269
14.3.1 寄存器列表	269
14.3.2 ADCLK (转换时钟设置寄存器)	270
14.3.3 ADMOD0 (模式控制寄存器 0)	271
14.3.5 ADMOD2 (模式控制寄存器 2)	273
14.3.6 ADMOD3 (模式控制寄存器 3)	275
14.3.7 ADMOD4 (模式控制寄存器 4)	276
14.3.8 ADMOD5 (模式控制寄存器 5)	277
14.3.9 ADMOD6 (模式控制寄存器 6)	278
14.3.10 ADREGn (转换结果寄存器 n: n = 0 ~ 11)	279
14.3.11 ADREGSP (AD 转换结果寄存器 SP)	280
14.3.12 ADCMP0 (AD 转换结果比较寄存器 0)	281
14.3.13 ADCMP1 (AD 转换结果比较寄存器 1)	281
14.4 操作描述	283
14.4.1 模拟参考电压	283
14.4.2 AD 转换模式	283
14.4.2.1 标准 AD 转换	283
14.4.2.2 最高优先级 AD 转换	284
14.4.3 AD 监控功能	284
14.4.4 选择输入通道	285
14.4.5 AD 转换详细描述	285
14.4.5.1 启动 AD 转换	285
14.4.5.2 AD 转换	285
14.4.5.3 在标准 AD 转换期间进行最高优先级 AD 转换	286
14.4.5.4 停止重复转换模式	286
14.4.5.5 重新激活标准 AD 转换	286
14.4.5.6 转换完成	287
14.4.5.7 中断生成计时和 AD 转换结果存储寄存器	288
14.4.5.8 DMA 请求	290
14.4.5.9 注意事项	290
15. 低电压检测电路(LVD)	292
15.1 配置	292
15.2 寄存器	293
15.2.1 寄存器列表	293
15.2.2 LVDCR1 (LVD 检测控制寄存器 1)	293
15.3 操作	295
15.3.1 选择检测电压及使能电压检测操作	295
15.3.2 通过检测电源电压复位	295

15.3.3 通过检测电源电压实现中断.....	295
15.3.4 检测状态.....	295
16. 看门狗计时器(WDT)	296
16.1 配置	296
16.2 寄存器	297
16.2.1 寄存器列表.....	297
16.2.2 WDMOD(看门狗计时器模式寄存器).....	297
16.2.3 WDCR(看门狗计时器控制寄存器).....	298
16.3 操作描述	299
16.3.1 基本操作.....	299
16.3.2 操作模式和状态.....	299
16.3.3 在检测到故障(失控)时的操作情况.....	299
16.3.3.1 INTWDT 中断生成.....	299
16.3.3.2 内部复位生成.....	300
16.4 看门狗计时器的控制	300
16.4.1 禁用控制.....	300
16.4.2 启用控制.....	300
16.4.3 看门狗计时器清除控制.....	300
16.4.4 看门狗计时器的检测时间.....	300
17. 闪存操作	302
17.1 特征	302
17.1.1 存储器容量和配置.....	302
17.1.2 功能.....	303
17.1.3 操作模式.....	303
17.1.3.1 描述.....	303
17.1.3.2 模式确定.....	304
17.1.4 存储器映象.....	305
17.1.5 保护/安全功能.....	305
17.1.5.1 保护功能.....	306
17.1.5.2 安全功能.....	306
17.1.6 寄存器.....	307
17.1.6.1 寄存器列表.....	307
17.1.6.2 FCSR(闪存状态寄存器).....	307
17.1.6.3 FCSECBIT(安全位寄存器).....	308
17.1.6.4 FCPSRA(闪存保护状态寄存器).....	308
17.2 闪存的详细描述	309
17.2.1 功能.....	309
17.2.2 闪存的操作模式.....	309
17.2.3 硬件复位.....	309
17.2.4 如何执行命令.....	310
17.2.5 命令描述.....	310
17.2.5.1 自动页面程序.....	310
17.2.5.2 自动芯片擦除.....	311
17.2.5.3 自动存储块擦除.....	311
17.2.5.4 自动保护位程序.....	312
17.2.5.5 自动保护位擦除.....	312
17.2.5.6 ID-READ.....	312
17.2.5.7 读取命令和读取/复位命令(软件复位).....	313
17.2.6 命令时序.....	314
17.2.6.1 命令时序列表.....	314
17.2.6.2 总线周期中的地址位配置.....	315
17.2.6.3 存储块地址(BA).....	315
17.2.6.4 如何指定保护位(PBA).....	316
17.2.6.5 ID 读取代码(IA, ID).....	316
17.2.6.6 命令时序示例.....	317
17.2.7 流程图.....	318
17.2.7.1 自动程序.....	318
17.2.7.2 自动擦除.....	319
17.3 如何用单引导模式重新编程闪存	320
17.3.1 模式设置.....	320
17.3.2 接口规格.....	320
17.3.3 对内部存储器的限制.....	321

17.3.4 操作命令.....	321
17.3.4.1 RAM 传输.....	322
17.3.4.2 闪存芯片擦除和保护位擦除.....	322
17.3.5 普通操作（不考虑命令）.....	322
17.3.5.1 连续操作模式的确定.....	322
17.3.5.2 应响应数据.....	325
17.3.5.3 密码确定.....	326
17.3.5.4 CHECK SUM 计算.....	327
17.3.6 RAM 传送时的传送格式.....	329
17.3.7 闪存芯片擦除和保护位擦除的转换格式.....	331
17.3.8 Boot 程序完整流程图.....	333
17.3.9 片装 BOOT ROM 中使用重编程序算法重编闪存程序.....	334
17.3.9.1 步骤-1.....	334
17.3.9.2 步骤-2.....	334
17.3.9.3 步骤-3.....	335
17.3.9.4 步骤-4.....	335
17.3.9.5 步骤-5.....	336
17.3.9.6 步骤-6.....	336
17.4 用户引导模式下编程.....	337
17.4.1 (1-A)编程程序存储于闪存的步骤.....	337
17.4.1.1 步骤-1.....	337
17.4.1.2 步骤-2.....	338
17.4.1.3 步骤-3.....	338
17.4.1.4 步骤-4.....	339
17.4.1.5 步骤-5.....	339
17.4.1.6 步骤-6.....	340
17.4.2 (1-B)编程程序自外部主机进行传送的步骤.....	341
17.4.2.1 步骤-1.....	341
17.4.2.2 步骤-2.....	342
17.4.2.3 步骤-3.....	342
17.4.2.4 步骤-4.....	343
17.4.2.5 步骤-5.....	343
17.4.2.6 步骤-6.....	344
18. 调试接口.....	345
18.1 规范概述.....	345
18.2 SWJ-DP.....	345
18.3 在停止模式时的外设功能.....	345
18.4 与调试工具的连接.....	346
18.4.1 关于与调试工具的连接.....	346
18.4.2 将调试接口引脚用作通用端口时的要点.....	346
19. 端口部分等效电路示意图.....	347
19.1 PORT 引脚.....	348
19.2 模拟引脚.....	348
19.3 控制引脚.....	349
19.4 时钟引脚.....	349
19.5 测试引脚.....	349
20. 电气特性.....	351
20.1 绝对最大额定值.....	351
20.2 DC 电气特性(1/2).....	352
20.3 DC 电气特性(2/2).....	354
20.4 10-位 AD 转换器电气特性.....	355
20.5 AC 电气特性.....	356
20.5.1 串行通道(SIO/UART).....	356
20.5.1.1 AC 测量条件.....	356
20.5.1.2 AC 电气特性(I/O 接口模式).....	356
20.5.2 I2C 接口(I2C).....	358
20.5.2.1 AC 测量条件.....	358
20.5.2.2 AC 电气特性.....	358
20.5.3 16-位计时器/事件计数器(TMRB).....	360

20.5.3.1 事件计数器	360
20.5.3.2 捕捉	360
20.5.4 外部中断	361
20.5.4.1 AC 测量条件	361
20.5.4.2 AC 电气特性	361
20.5.5 调试通信	362
20.5.5.1 AC 测量条件	362
20.5.5.2 SWD 接口	362
20.5.6 片上振荡器特性	363
20.5.7 外部振荡器	363
20.5.8 外部时钟输入	364
20.5.9 闪存特性	364
20.5.10 噪声滤波器特性	364
20.6 建议采用的振荡电路	365
20.6.1 陶瓷振荡器	365
20.6.2 印刷电路板设计时的注意事项	365
20.7 使用注意事项	366
20.7.1 通电电源电压电平	366
20.7.2 操作期间发生的电压下降现象	366
21. 封装尺寸	367

CMOS #!未定义!#位微控制器

TMPM037FWUG

TMPM037FWUG 32-位RISC微处理器系列采用ARM®Cortex®-M0 微处理器内核。
TMPM037FWUG的特点如下。

1.1 特点

1. ARM®Cortex®-M0 微处理器内核

- a. 通过使用拇指®-2指令，提高了编码效率。
- b. 已实现高性能，低功耗。
 - [高性能]
 - 可利用一个时钟脉冲执行32-位乘法(32 x 32=32 位)。
 - [低功耗]
 - 优化设计（采用低功耗库）
 - 待机功能（可使微控制器内核停止运行）
- c. 高速中断响应有利于实时控制
 - 一条可中断的长指令。
 - 由硬件自动处理推栈。

2. 字节序：低端在前格式

3. 片装程序存储器和数据存储器

片装闪存ROM：128 K字节

片装RAM：16 K字节

4. DMA控制器 (DMAC)： 2 通道/1 单元

传输模式：内置存储器，外设功能和外部存储器。

5. 时钟发生器 (CG)

外部时钟输入/外部振荡 (8 MHz ~ 20 MHz)

片装PLL (2x)

时钟变速器功能：该高速时钟可分为1/1, 1/2, 1/4, 1/8 和 1/16。

6. 待机模式

IDLE, STOP1

7. 中断源：可将该优先顺序设置为 4 种电平。

内部：24 个因数

外部：6 个因数

8. 输入/输出端口(PORT): 52 引脚
输入/输出引脚: 51 引脚
输出引脚: 1 引脚
9. 16-位计时器 (TMRB): 8 通道
16-位区间计时器模式
16-位事件计数器模式
16-位PPG输出 (4 阶段同步模式)
输入捕捉功能
10. 16位计时器(TMR16A): 2 通道
11. 看门狗计时器 (WDT): 1 通道
看门狗计时器 (WDT) 可生成复位或非可屏蔽中断 (NMI)。
12. 通用串行接口(SIO/UART): 5 通道
可选择UART模式或同步模式
发送FIFO: 4-阶段 8-位宽度,接收FIFO: 4-阶段 8-位宽度
13. I2C总线接口(I2C) 1 通道
通讯速率 100 kbps / 400 kbps
14. 10位AD转换器(ADC): 1 单元 (8 通道)
固定通道/扫描模式
单/重复模式
支持重复转换
AD监控
15. LVD/POR功能: 1 单元
16. 最大工作频率: 20 MHz
17. 调试接口
支持SWD。

18. 工作电压: 2.3 ~ 3.6 V

19. 温度范围

-40 °C ~ 85 °C (闪存W/E期间除外)

0 °C ~ 70 °C (闪存W/E期间)

20. 封装

LQFP64 (10 mm × 10 mm, 0.5 mm间距)

1.2 方块图

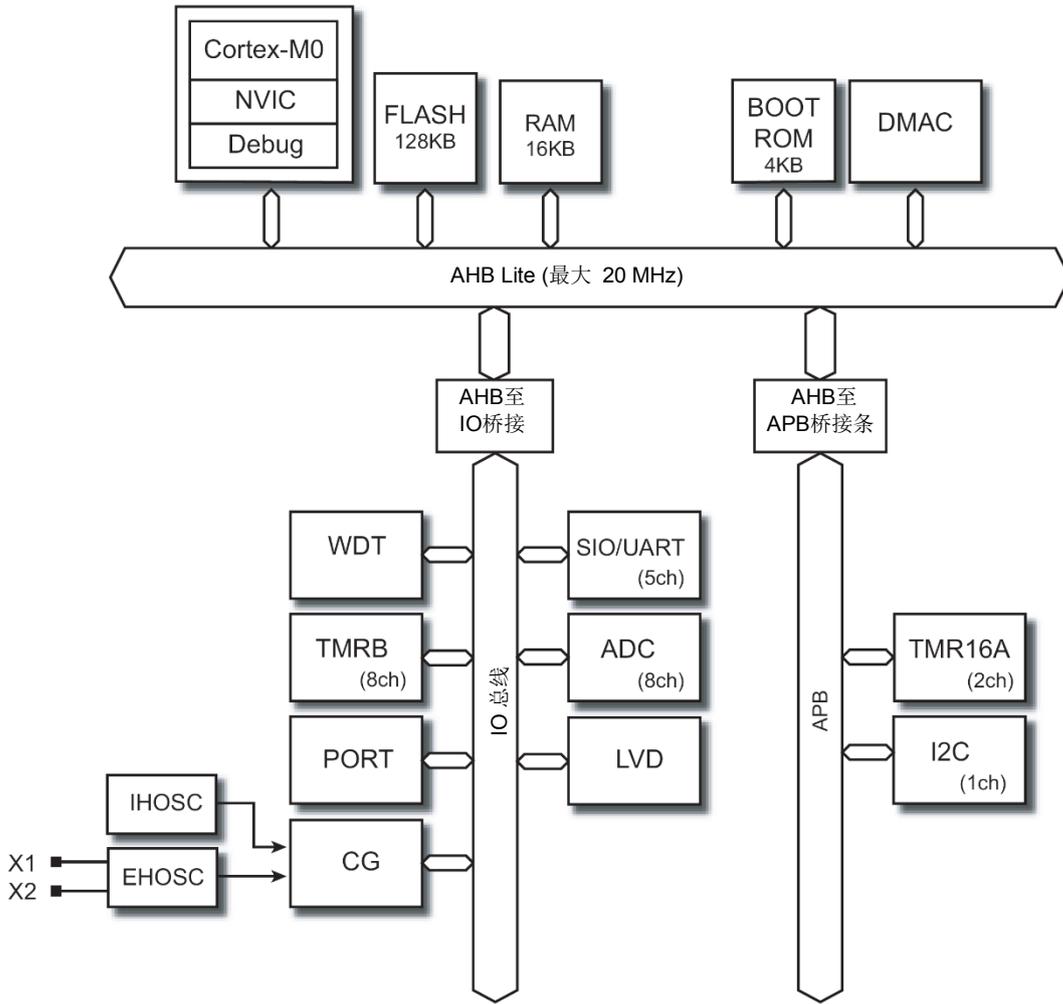


图 1-1 方块图

1.3 引脚分配(顶视图)

图 1-2 给出了TMPM037FWUG的引脚布局。

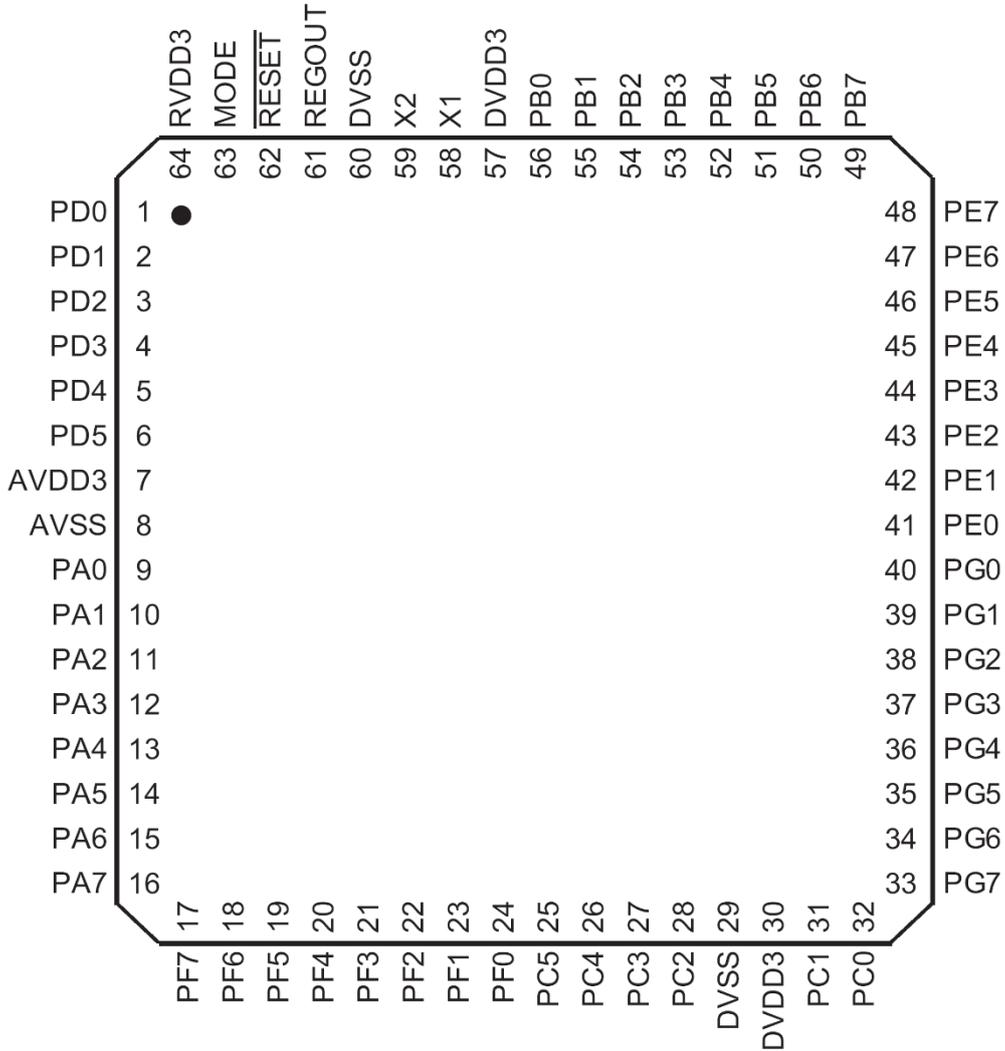


图 1-2 引脚布局(LQFP64 顶视图)

1.4 引脚名称与功能

1.4.1 各外设功能，控制引脚和电源引脚的名称与功能

1.4.1.1 外设功能

表 1-1 引脚数目和引脚名称

外设功能	引脚名称	输入 或 输出	功能
外部中断	INTx	输入	外部中断输入引脚x 外部中断输入引脚x带有一个噪声滤波器(滤波器宽度 30 ns(典型值))。
16 位计时器(TMRB)	TBxIN	输入	输入捕获输入引脚
	TBxOUT	输出	输出引脚
16 位计时器(TMR16A)	T16AxOUT	输出	输出引脚
串行通道(SIO/UART)	SCxTXD	输出	数据输出引脚
	SCxRXD	输入	数据输入引脚
	SCxSCLK	I/O	时钟输入/输出引脚
I2C总线接口(I2C)	I2CxSDA	I/O	数据输入/输出引脚
	I2CxSCL	I/O	时钟输入/输出引脚
模拟-数字转换器	AINx	输入	模拟输入引脚

1.4.1.2 调试功能

表 1-2 引脚名称与功能

引脚名称	输入 或 输出	功能
SWDIO	I/O	串行线数据输入与输出引脚
SWCLK	输入	串行线时钟输入引脚

1.4.1.3 控制功能

表 1-3 引脚名称与功能

引脚名称	输入 或 输出	功能
x1	输入	连接到高速振荡器。
x2	输出	连接到高速振荡器。
MODE	输入	MODE引脚 必须将MODE引脚连接至GND。
$\overline{\text{RESET}}$	输入	复位输入引脚
$\overline{\text{BOOT}}$	输入	BOOT模式控制引脚 在复位信号输入引脚的上升沿进行BOOT模式控制引脚的取样。 在BOOT模式控制引脚为“低”时，TMPM037FWUG改为单引导模式。 在引导模式控制引脚为“高”时，TMPM037FWUG改为单芯片模式。 有关详细数据请参看“闪存”一节。

1.4.1.4 电源引脚

表 1-4 引脚名称与功能

电源引脚名称	功能
REGOUT	引脚与稳压器的电容器相连。
RVDD3	稳压器的电源引脚。
DVDD3	数字电路的电源引脚。
DVSS	数字电路的GND引脚。
AVDD3	模拟电路的电源引脚。
AVSS	模拟电路的GND引脚。

1.4.2 TMPM037FWUG 的引脚名称与功能

1.4.2.1 引脚名称与功能列表的详细数据

表中符号的含义如以下所述。

1. 功能A

无需设置功能寄存器即可指定该功能，如该单元格内所述。

2. 功能B

需设置功能寄存器才可指定该功能，如该单元格内所述。该单元格中的编号与功能寄存器的编号对应。

3. 引脚规格

表中符号的含义如以下所述。

SMT/CMOS: 输入门的类型

SMT: Schmitt输入

COMS: CMOS输入

OD: 可编程开路漏极输出支持

是: 支持

N/A: 不支持

PU/PD: 可编程上拉/下拉

PU: 支持可编程上拉

PD: 支持可编程下拉

10 mA: 10 mA电流驱动器端口

是: 支持

N/A: 不支持

1.4.2.2 端口/调试引脚

表 1-5 引脚名称与功能<按PORT分类>

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTA											
9	PA0	AIN0						PU/PD	是	SMT	N/A
10	PA1	AIN1						PU/PD	是	SMT	N/A
11	PA2	AIN2						PU/PD	是	SMT	N/A
12	PA3	AIN3						PU/PD	是	SMT	N/A
13	PA4	AIN4						PU/PD	是	SMT	N/A
14	PA5	AIN5						PU/PD	是	SMT	N/A
15	PA6	AIN6						PU/PD	是	SMT	N/A
16	PA7	AIN7						PU/PD	是	SMT	N/A

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTB											
56	PB0	BOOT						PU/PD	是	SMT	N/A
55	PB1		SC4 RXD	SW CLK				PU/PD	是	SMT	N/A
54	PB2		SC4 TXD	SWDIO				PU/PD	是	SMT	N/A
53	PB3		SC4 SCLK					PU/PD	是	SMT	N/A
52	PB4							PU/PD	是	SMT	N/A
51	PB5	INT0						PU/PD	是	SMT	N/A
50	PB6	INT1						PU/PD	是	SMT	N/A
49	PB7	INT2						PU/PD	是	SMT	N/A

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTC											
32	PC0		I2C0 SCL					PU/PD	是	SMT	N/A
31	PC1		I2C0 SDA					PU/PD	是	SMT	N/A
28	PC2		TB2 OUT					PU/PD	是	SMT	是
27	PC3		TB0 OUT					PU/PD	是	SMT	是
26	PC4		T16A00 UT					PU/PD	是	SMT	N/A
25	PC5		TB0IN					PU/PD	是	SMT	N/A

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTD											
1	PD0		TB3 OUT					PU/PD	是	SMT	N/A
2	PD1		SC0 SCLK					PU/PD	是	SMT	N/A
3	PD2		SC0 RXD					PU/PD	是	SMT	N/A
4	PD3		SC0 TXD					PU/PD	是	SMT	N/A
5	PD4		TB3IN					PU/PD	是	SMT	N/A
6	PD5							PU/PD	是	SMT	N/A

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTE											
41	PE0							PU/PD	是	SMT	N/A
42	PE1							PU/PD	是	SMT	N/A
43	PE2		SC2 SCLK					PU/PD	是	SMT	N/A
44	PE3		SC2 RXD					PU/PD	是	SMT	N/A
45	PE4		SC2 TXD					PU/PD	是	SMT	N/A
46	PE5	INT5						PU/PD	是	SMT	N/A
47	PE6	INT4						PU/PD	是	SMT	N/A
48	PE7	INT3						PU/PD	是	SMT	N/A

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTF											
24	PF0		TB7IN					PU/PD	是	SMT	N/A
23	PF1		SC3 SCLK					PU/PD	是	SMT	N/A
22	PF2		SC3 RXD					PU/PD	是	SMT	N/A
21	PF3		SC3 TXD					PU/PD	是	SMT	N/A
20	PF4		TB7 OUT					PU/PD	是	SMT	N/A
19	PF5		T16A10 UT					PU/PD	是	SMT	N/A
18	PF6							PU/PD	是	SMT	N/A
17	PF7							PU/PD	是	SMT	N/A

引脚编号	端口	功能A	功能B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/CMOS	10 mA
PORTG											
40	PG0		SC1 SCLK					PU/PD	是	SMT	N/A
39	PG1		SC1 RXD					PU/PD	是	SMT	N/A
38	PG2		SC1 TXD					PU/PD	是	SMT	N/A
37	PG3		TB1IN					PU/PD	是	SMT	N/A
36	PG4		TB1 OUT					PU/PD	是	SMT	N/A
35	PG5		TB4 OUT					PU/PD	是	SMT	N/A
34	PG6		TB5 OUT					PU/PD	是	SMT	是
33	PG7		TB6 OUT					PU/PD	是	SMT	是

1.4.2.3 控制引脚

表 1-6 引脚数目和引脚名称

引脚编号	控制功能引脚名称
58	X1
59	X2
62	RESET
63	MODE
56	BOOT

1.4.2.4 电源引脚

表 1-7 引脚数目和引脚名称

引脚编号	电源引脚名称
61	REGOUT
64	RVDD3
30, 57	DVDD3
29, 60	DVSS
7	AVDD3
8	AVSS

2. 产品信息

本章主要对与外设功能相关的通道或单元数目，引脚信息，以及产品特殊功能信息等进行说明。本章应与“外设功能”一节配套使用。

- ”2.1.1 DMA控制器(DMAC)”
- ”2.1.2 16-位计时器/事件计数器(TMRB)”
- ”2.1.3 16-位计时器A(TMR16A)”
- ”2.1.4 串行通道(SIO/UART)”
- ”2.1.5 I2C总线(I2C)”
- ”2.1.6 模拟/数字转换器(ADC)”
- ”2.1.7 调试接口”

2.1 各外设功能相关信息

2.1.1 DMA 控制器(DMAC)

TMPM037FWUG 设有一个内置DMA控制器单元。

2.1.1.1 DMA 请求表

DMA请求表给出如下。

表 2-1 DMA请求表

请求编号	相应外设	
	Ch0, Ch1	
	突发	单
0	SIO/UART0 接收	-
1	SIO/UART0 发送	-
2	SIO/UART1 接收	-
3	SIO/UART1 发送	-
4	SIO/UART2 接收	-
5	SIO/UART2 发送	-
6	SIO/UART3 接收	-
7	SIO/UART3 发送	-
8	I2C0 发送/接收	-
9	-	-
10	SIO/UART4 接收	-
11	SIO/UART4 发送	-
12	TMRB (ch0-3)	-
13	TMRB (ch4-7)	-
14	最高优先级AD转换完成/ AD监控器 0/ AD监控器 1	-
	AD转换完成	-

2.1.1.2 用外设至外设传输支持的外设功能

用外设至外设传输支持的外设功能(寄存器)如以下所列。

表 2-2 所支持的外设功能

源	目标
外设寄存器	SCxBUF (x=0~4)
	TBxRG0~1 (x=0~7)
	TBxCP0~1 (X=0~7)
ScxBUF (x=0~4)	外设寄存器
TBxRG0~1 (X=0~7)	
TBxCP0~1 (x=0~7)	

2.1.1.3 DMA 请求控制寄存器(DMARQCTL)

DMA请求控制寄存器和I2C的地址如以下所述。

基地址= 0x4005_F000

寄存器名称		地址(基+)
DMAC请求设置寄存器	DMACREDGE	0x0000
DMAC请求清除寄存器	DMACRCLR	0x0004

2.1.1.4 DMACREDGE(DMAC 请求设置寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	I2CDMAC
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-1	-	R	读作零。
0	I2CDMAC	R/W	DMA请求在I2C总线模式下的设置。 0: 高激活 1: 保留 仅写入“0”。

2.1.1.5 DMACRCLR(DMAC 请求清除寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	DCLR0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-1	-	R	读作零。
0	DCLR0	W	在I2C总线模式下清除该请求信号。 0: - 1: DMA请求清除 在DMAC传输终止中断的服务例行程序中设置为“1”，并清除该DMA请求。 读作零。

2.1.2 16-位计时器/事件计数器(TMRB)

TMPM037FWUG设有 8 个TMRB通道。

表 2-3 引脚规格

通道	TBxOUT	TBxIN
TMRB0	PC3	PC5
TMRB1	PG4	PG3
TMRB2	PC2	-
TMRB3	PD0	PD4
TMRB4	PG5	-
TMRB5	PG6	-
TMRB6	PG7	-
TMRB7	PF4	PF0

表 2-4 同步启动规格

主通道	从通道
TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6, TMRB7

表 2-5 捕捉触发规格

触发输入通道	触发输出
TMRB0	TB6OUT
TMRB1	
TMRB2	
TMRB3	TB7OUT
TMRB4	
TMRB5	

2.1.3 16-位计时器 A(TMR16A)

TMPM037FWUG设有 2 个TMR16A通道。

表 2-6 引脚规格

通道	T16AxOUT
T16A0	PC4
T16A1	PF5

2.1.4 串行通道(SIO/UART)

TMPM037FWUG设有 5 个SIO通道。

表 2-7 引脚规格

通道	SCxTXD	SCxRXD	SCxSCLK
SC0	PD3	PD2	PD1
SC1	PG2	PG1	PG0
SC2	PE4	PE3	PE2
SC3	PF3	PF2	PF1
SC4	PB2	PB1	PB3

表 2-8 传输时钟规格

时钟输入通道	时钟输出
SC0	TB0OUT
SC1	TB1OUT

注：TMPM037FWUG不具备握手功能(SCxCTS引脚)。

2.1.5 I2C 总线(I2C)

TMPM037FWUG设有 1 个I2C通道。

表 2-9 引脚规格

通道	I2CxSDA	I2CxSCL
I2C0	PC1	PC0

2.1.6 模拟/数字转换器(ADC)

TMPM037FWUG设有 1 个ADC单元。

表 2-10 引脚规格

模拟输入	AIN0 ~ 7
端口	PA0 ~ 7

注：TMPM037FWUG模拟输入引脚是AIN0 至AIN7 的 8 个通道。

表 2-11 内部启动触发选择

类型	内部触发器
标准AD转换启动触发器	与TMRB ch0 匹配
最高优先级AD转换启动触发器	与TMRB ch1 匹配

2.1.7 调试接口

TMPM037FWUG支持串行线调试端口。

表 2-12 引脚规格

IF	SWDIO	SWCLK
串行线	PB2	PB1

3. 处理器内核

该TX00系列带有一个高性能的 32-位处理器内核(ARM Cortex-M0 处理器内核)。有关该处理器内核操作的相关信息，请参看ARM有限公司所发布的《Cortex-M0 技术参考手册》。本节主要对该文件中未予说明的TX00系列的专属功能进行说明。

3.1 处理器内核相关信息

下表给出了TMPM037FWUG中的处理器内核的修改内容。

请参看该CPU内核与CPU体系结构的详细信息，并请参看以下URL中的ARM手册《Cortex-M系列处理器》：
<http://infocenter.arm.com/help/index.jsp>

产品名称	内核版次
TMPM037FWUG	r0p0-03

3.2 可配置选项

该Cortex-M0 内核带有可选程序块。下表给出了TMPM037FWUG中的可配置选项。

可配置的选项	执行
中断	32
数据字节存储次序	小字节序
SysTick计时器	有
检测点比较器的数目	2
断点比较器的数目	4
暂停调试支持	有
乘法器	快

注：该快速乘法器采用 32-位 x 32-位乘法，可输出最低有效的 32-位。

3.3 异常/中断

以下将对异常与中断进行说明。

3.3.1 中断输入的数目

在Cortex-M0 内核中，可任意定义中断输入的数目。

TMPM037FWUG有 32 个中断输入。

3.3.2 SysTick

TMPM037FWUG带有一个可生成SysTick异常的SysTick计时器。

关于SysTick异常的详细信息，参见NVIC寄存器的SysTick异常和寄存器的“SysTick”一节。

3.3.3 SYSRESETREQ

在对应用中断和复位控制寄存器的<SYSRESETREQ>位进行设置时，该Cortex-M0 内核可输出SYSRESETREQ信号。

在输出SYSRESETREQ信号时，TMPM037FWUG也会进行同样的操作。

3.3.4 LOCKUP

在不能修补的异常生成时，Cortex-M0 内核可输出LOCKUP信号，用以指示软件中所含的严重错误。

TMPM037FWUG不使用该信号。需使用非可屏蔽中断(NMI)或复位，才能从LOCKUP状态返回。

3.4 事件

该Cortex-M0 内核有事件输出信号和事件输入信号。事件输出信号由SEV指令执行输出。如果输入了某事件，则该内核会从WFE指令所导致的低功耗模式返回。

TMPM037FWUG不使用事件输出信号和事件输入信号。请不要使用SEV指令与WFE指令。

3.5 电源管理

Cortex-M0 内核配备使用SLEEPING信号和SLEEPDEEP信号的电源管理系统。

在设置系统控制寄存器的<SLEEPDEEP>位时，可输出SLEEPDEEP信号。

在以下情况下会输出这些信号：

- 等待中断(WFI)指令的执行
- 等待事件(WFE)指令的执行

- 若设置了系统控制寄存器的<SLEEPONEXIT>位，当中断服务程序(ISR)退出时。
TMPM037FWUG不使用SLEEPDEEP信号，因此不必设置 <SLEEPDEEP>位。其也不使用事件信号，因此请不要使用WFE指令。
有关电源管理的详细资料，请参看“时钟/模式控制”一节。



4. 内存地址图

4.1 内存地址

TMPM037FWUG的内存地址基于ARM Cortex-M0 处理器内核内存地址。

内部ROM被地址至Cortex-M0 内核存储器的代码，内部RAM被地址至SRAM区，特殊功能寄存器(SFR)分别被地址至外设区。

SRAM区域，特殊功能寄存器(SFR)被地址到外设区域。

特殊功能寄存器(SFR)指示外设功能的I/O端口和控制寄存器。

TMPM037FWUG具备与Cortex-M3 相当的位带部件，且该位带区包含了TMPM037FWUG的SRAM和SFR区。

CPU寄存器区域是处理器内核的内部寄存器区域。

有关各区的详细资料，见《Cortex-M0 技术参考手册》。

注意，对已注明“故障”的区域进行访问可导致发生硬故障。不要访问该供应商特定区和该保留区。

图 4-1 给出了TMPM037FWUG的内存地址。

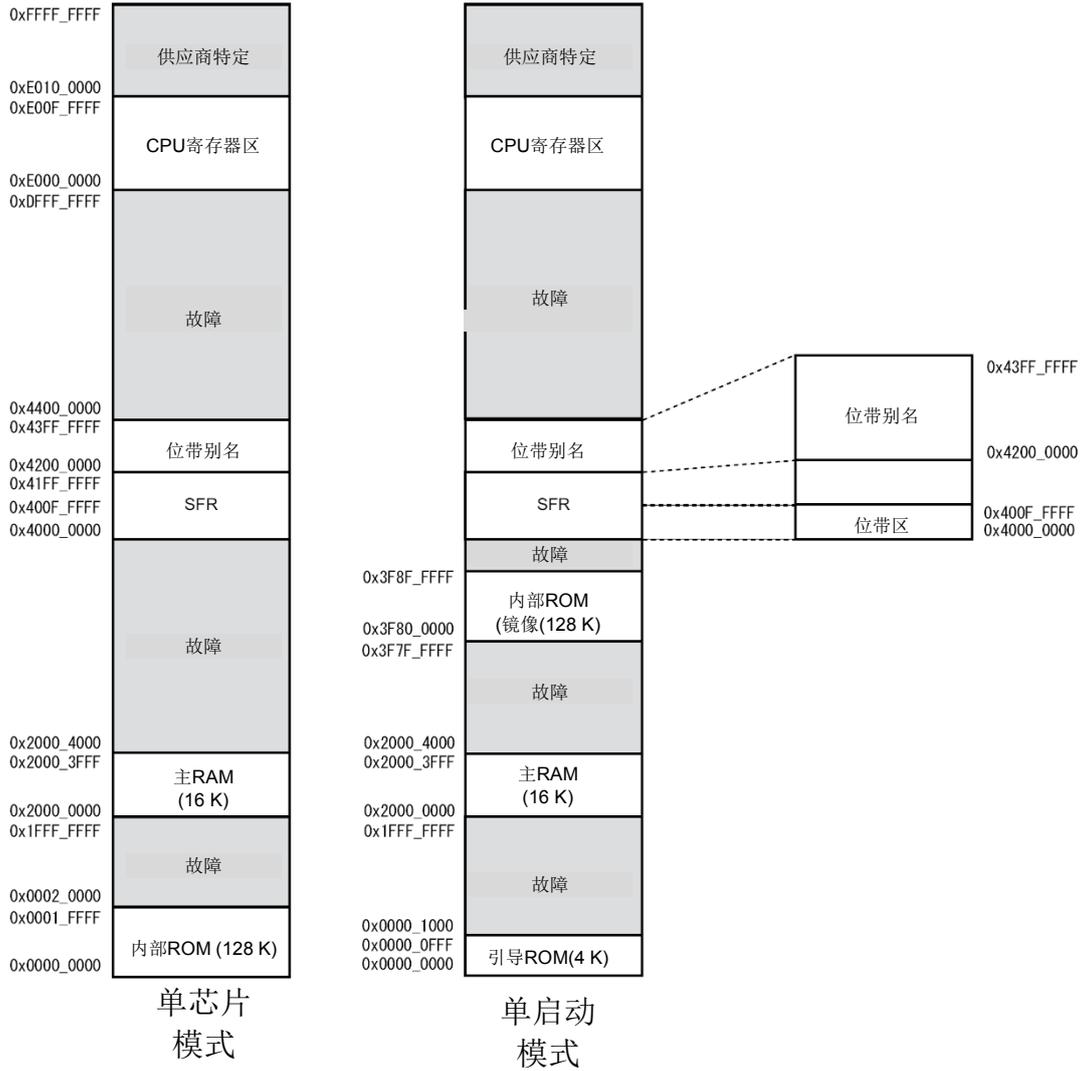


图 4-1 TMPM037FWUG内存地址

4.2 总线矩阵

TMPM037FWUG内含两个总线主设备(例如CPU内核和DMA控制器)。

总线主设备连接至总线矩阵的从设备端口(S0,S1)。在该总线矩阵中,主端口(M0~M7)可通过下图中所给出的连接(o)连接至外设功能。

同时,多个从设备被连接至该总线矩阵中的同一总线主设备线,如果同时生成多个从设备访问,则优先权会被给予来自其从设备编号最小的主设备的访问。

4.2.1 结构

4.2.1.1 单芯片模式

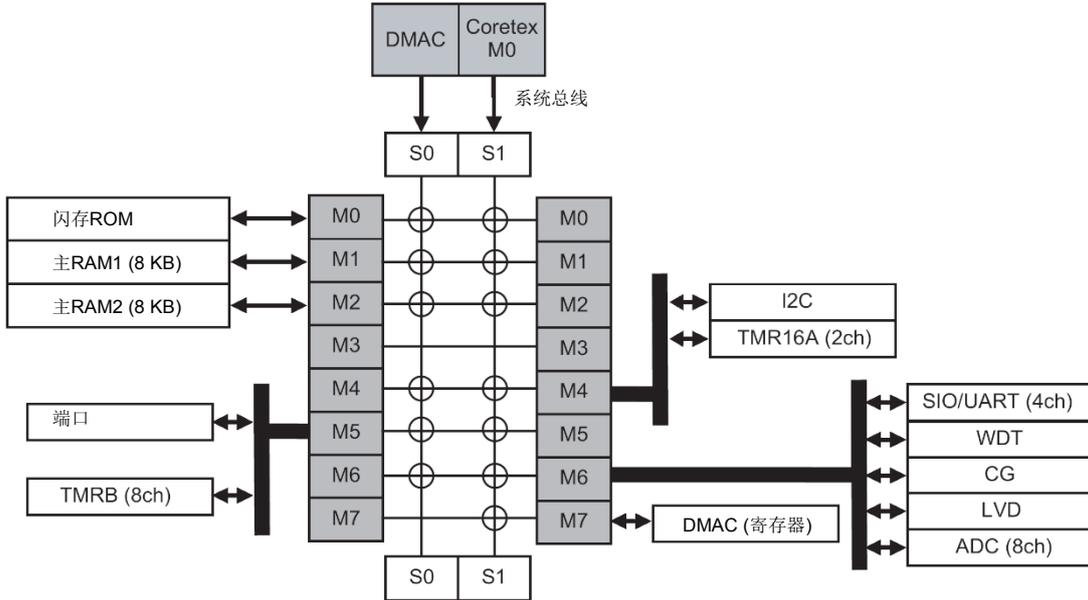


图 4-2 TMPM037FWUG (单芯片模式)

4.2.1.2 单启动模式

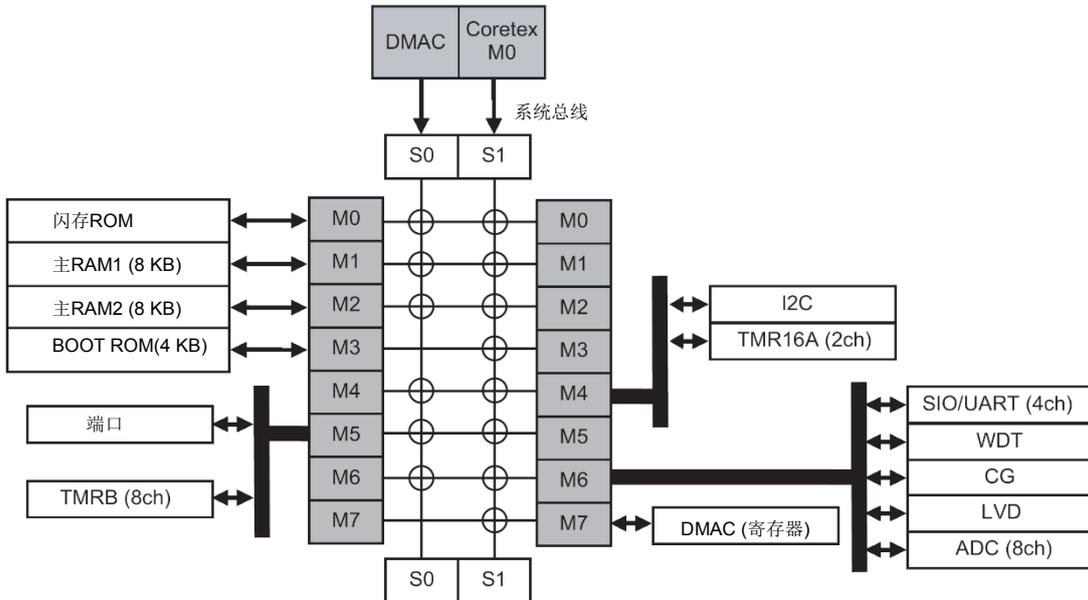


图 4-3 TMPM037FWUG (单引导模式)

4.2.2 连接表

4.2.2.1 代码区/SRAM 区

(1) 单芯片模式

起始地址			DMAC	核
			S0	S1
0x0000_0000	闪存ROM	M0	○	○
0x0002_0000	故障	-	故障	故障
0x2000_0000	主RAM0	M1	○	○
0x2000_2000	主RAM1	M2	○	○
0x2000_4000	故障	-	故障	故障

(2) 单启动模式

起始地址			DMAC	核
			S0	S1
0x0000_0000	BOOT ROM	M3	故障	○
0x0000_1000	故障	-	故障	故障
0x2000_0000	主RAM0	M1	○	○
0x2000_2000	主RAM1	M2	○	○
0x2000_4000	故障	-	故障	故障
0x3F80_0000	闪存ROM(镜像)	M0	○	○
0x3F82_0000	故障	-	故障	故障

4.2.2.2 外设区

起始地址			DMAC	核
			S0	S-Bus S1
0x4000_0000	DMAC	M7	-	0
0x4000_1000	故障	-	故障	故障
0x4005_F000	DMARC	M4	0	0
0x4006_0000	保留	-	-	-
0x4008_D000	TMR16A	M4	0	0
0x4008_F000	保留	-	-	-
0x400A_0000	I2C	M4	0	0
0x400A_1000	故障	-	故障	故障
0x400C_0000	端口	M5	0	0
0x400C_0800	保留	-	-	-
0x400C_4000	TMRB	M5	0	0
0x400C_4800	故障	-	故障	故障
0x400E_1000	SIO/UART	M6	0	0
0x400E_1500	保留	-	-	-
0x400F_2000	WDT	M6	-	0
0x400F_2100	保留	-	-	-
0x400F_3000	CG	M6	-	0
0x400F_3100	保留	-	-	-
0x400F_4000	LVD	M6	-	0
0x400F_4100	保留	-	-	-
0x400F_C000	ADC	M6	0	0
0x400F_C100	保留	-	-	-
0x41FF_FF00	SFR(FLASH)	M6	-	0
0x4200_0000	位带别名	-	-	0

4.2.3 外设功能的地址列表

不要访问外设区中的地址(控制寄存器除外)。有关各控制寄存器的详细资料, 请参看各外设功能所在节次。

外设功能		基址
DMA控制器(DMAC)		0x4000_0000
DMA请求控制器(DMARC)		0x4005_F000
16-位计时器(TMR16A)	ch0	0x4008_D000
	ch1	0x4008_E000
I2C串行总线接口(I2C)	ch0	0x400A_0000
输入/输出端口(PORT)	端口A	0x400C_0000
	端口B	0x400C_0100
	端口C	0x400C_0200
	端口D	0x400C_0300
	端口E	0x400C_0400
	端口F	0x400C_0500
	端口G	0x400C_0600
16-位计时器/事件计数器(TMRB)	ch0	0x400C_4000
	ch1	0x400C_4100
	ch2	0x400C_4200
	ch3	0x400C_4300
	ch4	0x400C_4400
	ch5	0x400C_4500
	ch6	0x400C_4600
	ch7	0x400C_4700
串行通道(SIO/UART)	ch0	0x400E_1000
	ch1	0x400E_1100
	ch2	0x400E_1200
	ch3	0x400E_1300
	ch4	0x400E_1400
看门狗计时器(WDT)		0x400F_2000
时钟/模式控制(CG)		0x400F_3000
低电压检测电路(LVD)		0x400F_4000
模拟/数字转换器(ADC)		0x400F_C000
SFR(FLASH)		0x41FF_FF00

5. 复位操作

复位操作源如下几种。

上电复位

复位引脚(RESET)

低电压检测电路(LVD)

看门狗计时器(WDT)

通过CPU和来自复位寄存器位<SYSRESETREQ>的信号实现应用中断

通过检查“异常”一节所述时钟发生器寄存器中的CGRSTFG，即可识别复位源。

有关通过低电压检测电路实现的复位，请参看“低电压检测电路”一节。

有关通过WDT实现的复位，见“看门狗计时器”一节。

有关通过<SYSRESETREQ>实现的复位，请参看《Cortex-M0 技术参考手册》。

注：复位操作一旦进行，内部RAM数据将无法保证。

5.1 冷复位

5.1.1 通过 $\overline{\text{RESET}}$ 引脚实现冷复位

在上电时， $\overline{\text{RESET}}$ 引脚必须保持为“低”。

在上电时，需考虑内置稳压器的稳定时间。在本产品中，内部稳压器至少需要 1 ms 才能实现安定。在冷复位时， $\overline{\text{RESET}}$ 引脚必须在足以让内部稳压器实现安定的持续时间内保持为“低”。

在 $\overline{\text{RESET}}$ 引脚变为“高”之后约 0.8 ms 时，内部 $\overline{\text{RESET}}$ 即被解除。

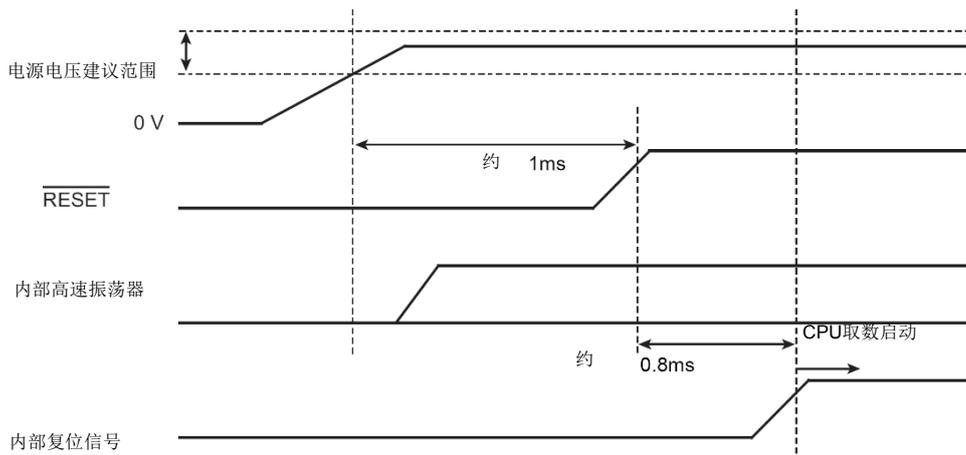


图 5-1 冷复位操作时序

5.1.2 用上电复位电路实现冷复位

如果使用的是上电复位电路，则对电源线的上升时间有限制。电源引脚应在 1.0 ms内启动电源并达到所建议的操作电压范围。

在电源电压进入所建议操作电压范围之后约 1.8 ms之后，内部复位即被解除。

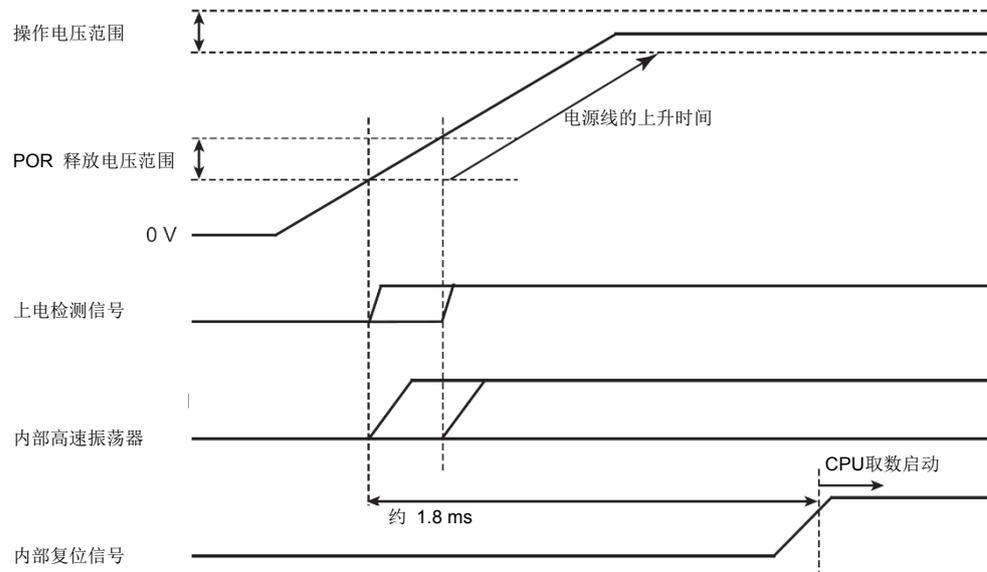


图 5-2 用上电复位电路实现冷复位

5.2 热复位

在对TMPM037FWUG进行复位时，必须具备以下条件；电源电压处于操作范围内； $\overline{\text{RESET}}$ 引脚至少在 12个内部高速时钟内保持为“低”。在 $\overline{\text{RESET}}$ 引脚变为“高”之后约 0.8 ms时，内部复位即被解除。

如果是WDT复位或<SYSRESETREQ>复位，则内部复位会在复位后约 30 个内部高速时钟时被解除。

5.3 复位后

大部分内核控制寄存器和外设功能控制寄存器(SFR)可通过热复位实现初始化。

该内核的系统调试部件寄存器(FPB, DWT, 以及ITM),以及与闪存相关的寄存器中的FCSECBIT, 则仅可通过冷复位实现初始化。

重置解除后, MCU根据内部高速振荡器的时钟开始运行。如有必要, 应设置外部时钟和PLL复接。



6. 时钟/模式控制

6.1 特点

通过该时钟/模式控制，可选择PLL时钟乘法电路和振荡器的时钟齿轮,预分频器时钟和预热。
还有一个低功耗模式，该模式可通过模式推移减少功耗。
本节主要对时钟工作模式与模式推移进行了说明。

该时钟/模式控制程序块具备以下功能：

- 控制系统时钟
- 控制预分频器时钟
- 控制PLL倍频电路
- 控制预热计时器

除NORMAL模式外，TMPM037FWUG还可按照其使用条件操作低功率模式，以减少功耗。

6.2 寄存器

6.2.1 寄存器列表

下表给出了与时钟/模式控制相关的各寄存器和地址。
有关基准地址，请参看“存储器”一节的“外设基准地址列表”。

寄存器名称		地址(基+)
系统控制寄存器	CGSYSCR	0x0000
振荡控制寄存器	CGOSCCR	0x0004
待机控制寄存器	CGSTBYCR	0x0008
PLL选择寄存器	CGPLLSEL	0x000C
保护寄存器	CGPROTECT	0x003C

6.2.2 CGSYSCR(系统控制寄存器)

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

位	比特符号	类型	功能
31-21	-	R	读作“0”。
20	-	R/W	写入“0”。
19-18	-	R	读作“0”。
17-16	-	R/W	写入“01”。
15-14	-	R	读作“0”。
13	-	R/W	写入“0”。
12	FPSEL	R/W	选择fperiph源时钟。 0: fgear 1: fc 将源时钟规定为fperiph。 选择fc以固定fperiph(不考虑时钟齿轮)。
11	-	R	读作“0”。
10-8	PRCK[2:0]	R/W	预分频器时钟 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: 保留 011: fperiph/8 111: 保留 将该预定标器时钟指定给外设I/O。
7-3	-	R	读作“0”。
2-0	GEAR[2:0]	R/W	高速齿轮时钟(fc)。 000: fc 100: (fc/2) 001: 保留 101: (fc/4) 010: 保留 110: (fc/8) 011: 保留 111: (fc/16)

6.2.3 CGOSCCR(振荡控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	WUODR							
复位后	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	WUODR				HWUPSEL	EHOSCSEL	OSCSEL	XEN2
复位后	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	OSCF	XEN1
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	PLLON	WUEF	WUEON
复位后	0	0	1	1	0	0	0	0

位	比特符号	类型	功能
31-20	WUODR[11:0]	R/W	预热计数器设置值。 对于上 12-位计数器值的计时器，设置 16-位计时器。
19	HWUPSEL	R/W	选择高速预热计数器时钟 0: 内部OSC(IHOSC) 1: 外部OSC (feosc) 选择预热计时器的OSC时钟。预热计时器按所选择的时钟进行计数。
18	EHOSCSEL	R/W	选择外部OSC源 0: 外部时钟输入(EHCLKIN) 1: 外部振荡器(EHOSC)
17	OSCSEL	R/W	选择高振荡器(注 2) 0: 内部(IHOSC) 1: 外部(EHOSC)
16	XEN2	R/W	内部高速振荡器操作控制 0: 停止 1: 振动
15-14	-	R/W	写入"0"。
13	-	R	读作"0"。
12	-	R/W	写入"0"。
11-10	-	R	读作"0"。
9	OSCF	R	所选择高速振荡器的状态。 0: 内部高速振荡器 1: 外部高速振荡器
8	XEN1	R/W	外部OSC操作控制 0: 停止 1: 振动
7-3	-	R/W	仅写入"00110"。
2	PLLON	R/W	PLL(乘法电路)操作控制(注 3) 0: 停止 1: 振动
1	WUEF	R	预热计时器的操作状态 0: 预热完成 1: 可通过该预热计时器的状态，监控预热活动的情况。
0	WUEON	W	预热计时器(WUP)的操作 0: 忽略 1: 预热计时器启动 启用以启动该预热计时器 读作"0"。

注 1: 有关预热设定，请参看"6.3.4 预热功能"。

注 2: 在选择外部振荡器时, 应在设置<EHOSCSEL>之后选择<OSCSEL>.(不要同时改变<EHOSCSEL>和<OSCSEL>).

注 3: 一旦设置该PLL值, CGOSCCR<PLLON>即在约 100 μ s的PLL的初始时间消逝之后被设置为"0", PLL的状态随即启动。

注 4: 在经改动的CGOSCCR<PLLON>被设置为"1"之后, CGPLLSEL<PLLSEL >应在预热时间结束时被设置为"1".

注 5: 在从STOP1 模式返回时, 寄存器CGOSCCR和CGPLLSEL<PLLSEL>的相关位

<HWUPSEL>,<OSCSEL>,<XEN2>,<XEN1>和<PLLON>即被初始化, 且内部高速振荡器启动。

注 6: 当内部高速振荡器(IHOSC) 用作系统时钟时, 切勿使用PLL乘法。

注 7: 在使用内部高速振荡器(IHOSC)时, 不要将其用作对高精度保证有要求的系统时钟。

6.2.4 CGSTBYCR(待机控制寄存器)

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

位	比特符号	类型	功能
31-20	-	R	读作"0"。
19-18	-	R/W	写入"0"。
17	-	R/W	写入"0"。
16	-	R	读作"0"。
15-3	-	R	读作"0"。
2-0	STBY[2:0]	R/W	低功耗模式控制。 000: 保留 001: STOP1 010: 保留 011: IDLE 100: 保留 101: 保留 110: 保留 111: 保留

6.2.5 CGPLLSEL(PLL 选择寄存器)

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

位	比特符号	类型	功能
31-19	-	R	读作"0"。
18	PLLST	R	PLL中所选择时钟的状态 0: fosc 1: fPLL
17-16	-	R	读作"0"。
15-1	PLLSET	R/W	PLL乘法值(切勿使用非下列值) 0x609F: 2 次
0	PLLSEL	R/W	PLL的使用 0: fosc使用 1: fPLL使用 规定使用或不用已被乘以PLL的时钟。 fosc可在重设后自动进行设定。在使用该PLL时必须进行设置。

注 1: 选择表 6-2 所列的PLL乘积值。

注 2: 在CGOSCCR<PLLON>0 (PLL停止)时, 选择PLL乘积值。

注 3: 一旦设置该PLL值, CGOSCCR<PLLON>即在约 100 μs的PLL的初始时间消逝之后被设置为"0", PLL的状态随即启动。

注 4: 在从STOP1 模式返回时, 寄存器CGOSCCR和CGPLLSEL<PLLSEL>的相关位

<HWUPSEL>,<OSCSEL>,<XEN2>,<XEN1>和<PLLON>即被初始化, 且内部高速振荡器启动。

注 5: 当内部高速振荡器(IHOSC) 用作系统时钟时, 切勿使用PLL乘法。

6.2.6 CGPROTECT (保护寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7-0	CGPROTECT	R/W	寄存器保护控制 0xC1: 寄存器写入启用 0xC1除外: 寄存器写入禁用 初始值是"0xC1"，原因是写入可启用各寄存器，且各寄存器（CGPROTECT寄存器除外）在写入"0xC1"以外的数值时均无法被写入。

6.3 时钟控制

6.3.1 时钟型

每个时钟的定义如下：

fosc	: 来自内部振荡器的时钟,或来自X1&X2 引脚的输入。
f _{PLL}	: 时钟 (乘以PLL(2 x))。
fc	: CGPLLSEL<PLLSEL>所规定的时钟 (高速时钟)
fgear	: CGSYSCR<GEAR[2:0]>所指定的时钟(系统时钟)。
fsys	: CGSYSCR<GEAR[2:0]>所指定的时钟(系统时钟)。
fperiph	: CGSYSCR<FPSEL[2:0]>所指定的时钟
φT0	: CGSYSCR<PRCK[2:0]> 所规定的时钟(预定标器时钟)

齿轮时钟fgear和预分频器时钟φT0 可分频如下。

齿轮时钟	: fc, fc/2, fc/4, fc/8, fc/16
预分频时钟	: fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16, fperiph/32

6.3.2 复位后初始值

复位操作可初始化该时钟配置(如以下所述)。

内部高速振荡器	: 振荡
外部高速振荡器	: 停止
PLL (锁相回路电路)	: 停止
高速齿轮时钟	: fc (无分频)

复位操作可导致所有时钟配置均变为与fosc的相同。

fc = fosc
fsys = fosc
φT0 = fosc

6.3.3 时钟系统图

图 6-1 显示时钟系统的示意图。

在复位后，箭头所示的选择器输入时钟被设为默认值。

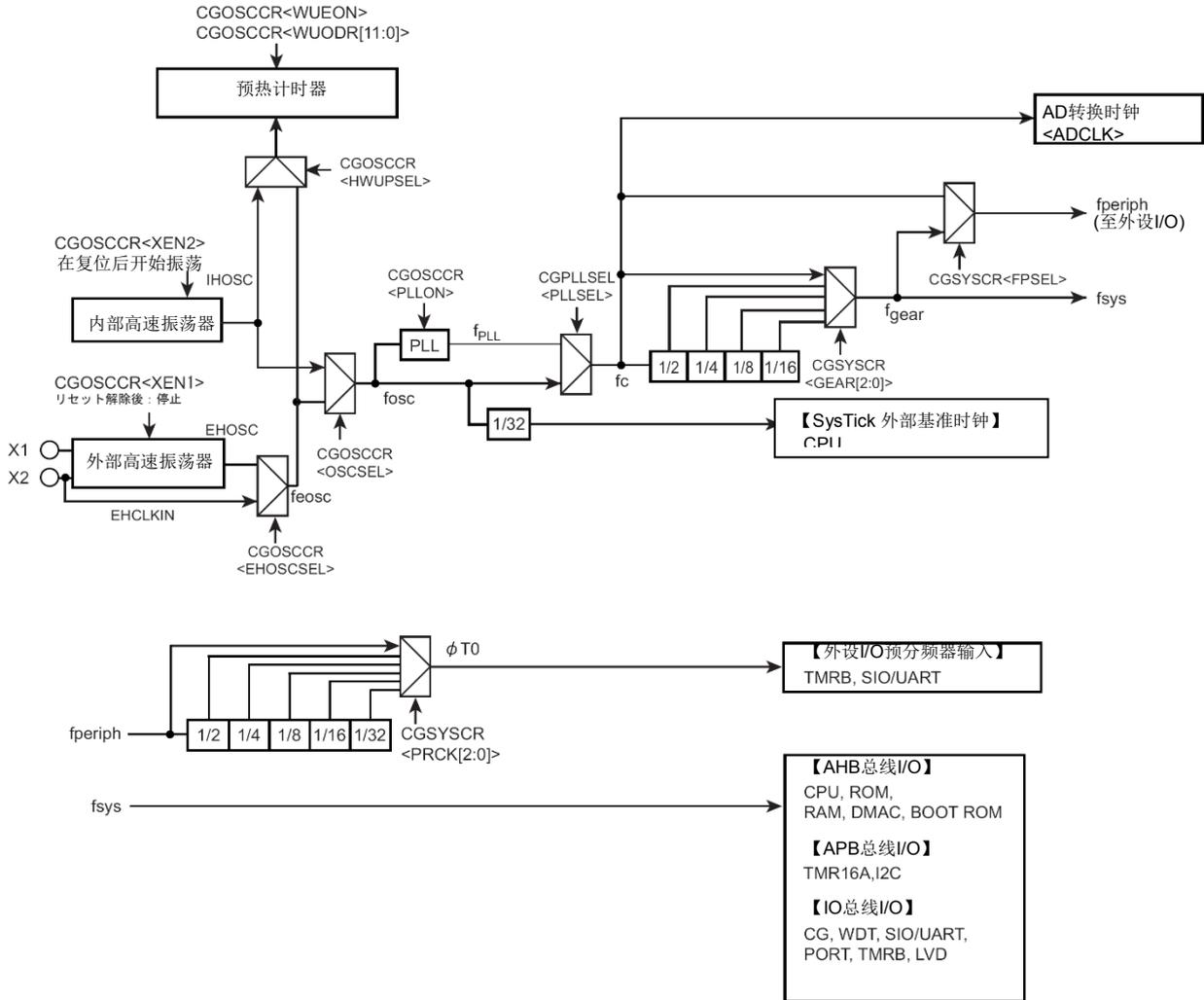


图 6-1 时钟方块图

6.3.4 预热功能

该预热功能可利用该预热计时器，固定该振荡器稳定时间与PLL。在使用外部时钟输入期间，在使用稳定外部时钟时无需使用预热功能。

有关详细资料，请参看“6.6.6 预热”。

其中对预热功能的配置方式进行了说明。

1. 指定递增计数时钟

在CGOSCCR<HWUPSEL>中规定预热计数器的向上计数时钟

2. 指定预热计数器值

可通过以下公式计算预热时间，然后将下部 4 位舍去，并设置到<WUODR[11:0]>位。

$$\text{预热周期的数目} = \frac{\text{设置的预热时间}}{\text{输入频率周期(s)}}$$

在使用 8 MHz 的高速振荡器时，按以下所述将预热时间设置为 5 ms。

$$\frac{\text{设置的预热时间}}{\text{输入频率周期(s)}} = \frac{5 \text{ ms}}{1/8 \text{ MHz}} = 40,000 \text{ 周期} = 0x9C40$$

舍入下 4 位，将 0x9C4 设为CGOSCCR<WUODR[11:0]>

3. 启动预热功能，并确认预热完成

如果是通过软件启动预热（即将CGOSCCR<WUEON>设置为“1”）的，则预热会启动递增计数。采用CGOSCCR <WUEF> 确认预热的启动和完成情况。

<WUEF>“1”表示正在预热，<WUEF>“0”则表示预热已完成

注 1：将预热计数值设置为CGOSCCR<WUDOR[11:0]>，等待至该值被反射，然后通过执行“WFI”命令，转入低功耗模式。

注 2：预热计时器根据振荡器时钟操作，所以当振荡频率发生波动时，则有可能导致预热计时器出现错误。因此，该预热时间应被视为近似时间。

预热功能设定示例给出如下。

表 6-1 预热功能设定示例

CGOSCCR<WUODR[11:0]> = "0x9C4"	: 指定预热时间
⌚ CGOSCCR<WUODR[11:0]> 读数	: 确认预热时间 重复至所读出的数据为"0x9C4"。
CGOSCCR<XEN2> = "1"	: 内部高速振荡器(IHOSC)启用
CGOSCCR<WUEON> = "1"	: 开启预热计时器(WUP)
⌚ CGOSCCR<WUEF>读数	: 读数保持等待, 直至状态变为"0"(预热完成)

注 1: 为了保持稳定, 在使用外部时钟时。

注 2: 预热计时器根据振荡器时钟操作, 所以当振荡频率发生波动时, 则有可能导致预热计时器出现错误。因此, 该预热时间应被视为近似时间。

注 3: 在将预热计数值设置为CGOSCCR<WUODR[11:0]>之后, 等待至该值的确认被反射, 然后通过WFI指令转换为待机模式。

注 4: 在从STOP1 模式返回时, 相关的位即CGPLLSEL<PLLSEL>,CGOSCCR<HWUPSEL>,<OSCSEL>,< XEN2>,<XEN1>和<PLLON>即被初始化, 以启动内部高速振荡器, 但CGOSCCR<WUODR[11:0]>未被初始化。

6.3.5 时钟倍增电路(PLL)

该电路可输出该高速振荡输出时钟的 f_{PLL} 时钟（被乘以 2）。结果，振荡器输入频率可能是低频率，内部时钟变成高速。

6.3.5.1 如何配置 PLL 功能

在复位之后该PLL被禁用。

通过将CGPLLSEL<PLLSET> 设置为乘积值，且CGOSCCR<PLLON>为”0”，即可启用该PLL。

在PLL的初始化时间 100 μ s消逝后，将<PLLON>设置为”1”。在为期 100 μ s的锁住时间消逝后，将设置CGPLLSEL<PLLSEL>为“1”，并使用来自fosc的 f_{PLL} （已被乘以2）。

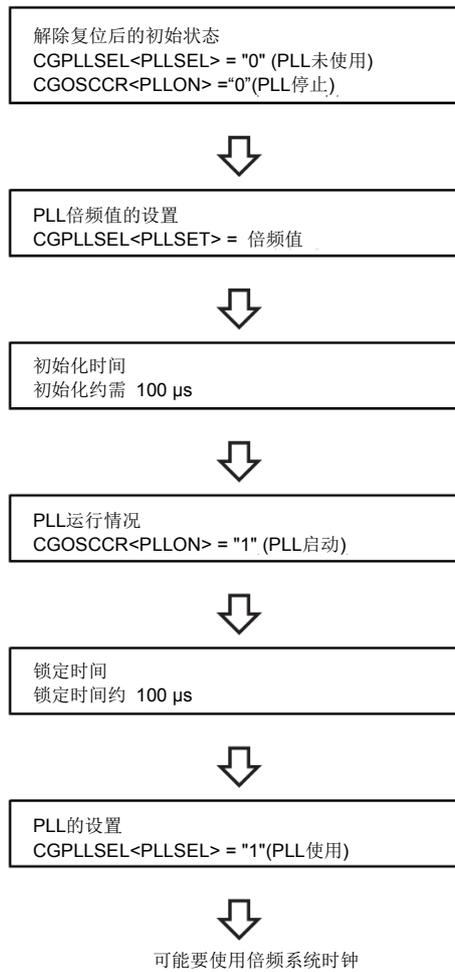
PLL功能需要一定时间稳定，可通过预热功能或其他方法保证稳定时间。

注 1：当内部高速振荡器(IHOSC) 用作系统时钟时，切勿使用PLL乘法。

至于该 2 乘积值，则仅允许采用以下设置。

乘法	<PLLSET>
2	0x609F

6.3.5.2 PLL 设置的顺序



6.3.6 系统时钟

可将内部高速振荡时钟和外部高速振荡时钟(属于振荡器连接或输入时钟)用作系统时钟的源时钟。在使用内部高速振荡时钟时，不要将其用作对高精度保证有要求的系统时钟。在使用外部高速振荡时钟时，可通过相乘使用该PLL功能。

源时钟		频率	使用PLL
内部高速振荡(f _{IHOSC})		10 MHz	无法使用
外部高速振荡	振荡器(f _{EHOSC})	8 ≤ f _{OSC} ≤ 10 MHz	2 相乘
		10 < f _{OSC} ≤ 20 MHz	无法使用
	输入时钟(f _{EHCLKIN})	8 ≤ f _{OSC} ≤ 10 MHz	2 相乘
		10 < f _{OSC} ≤ 20 MHz	无法使用

通过相乘用PLL生成的该时钟，可用作系统时钟和ADC时钟。各自使用的频率如下所列。ĀB

	系统时钟	ADC时钟
工作频率(MHz)	1~ 20	20 (最大值)

系统时钟通过由CGSYSCR<GEAR>进行分频。虽然在工作时可改变设置，但是实际切换稍有延时。表 6-2 给出了通过设置PLL和时钟齿轮而实现工作频率的示例。

表 6-2 PLL被乘以 2 时的系统时钟频率

外部 振荡器 (MHz)	外部 时钟 输入 (MHz)	PLL 倍频	最高 工作 频率 (fc) (MHz)	时钟齿轮(CG)					时钟齿轮(CG)				
				PLL = ON					PLL = OFF				
				1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
8	8	2	16	16	8	4	2	1	8	4	2	1	-
10	10		20	20	10	5	2.5	1.25	10	5	2.5	1.25	-
12	12	-	12	-	-	-	-	-	12	6	3	1.5	-
16	16		16	-	-	-	-	-	16	8	4	2	1
20	20		20	-	-	-	-	-	20	10	5	2.5	1.25

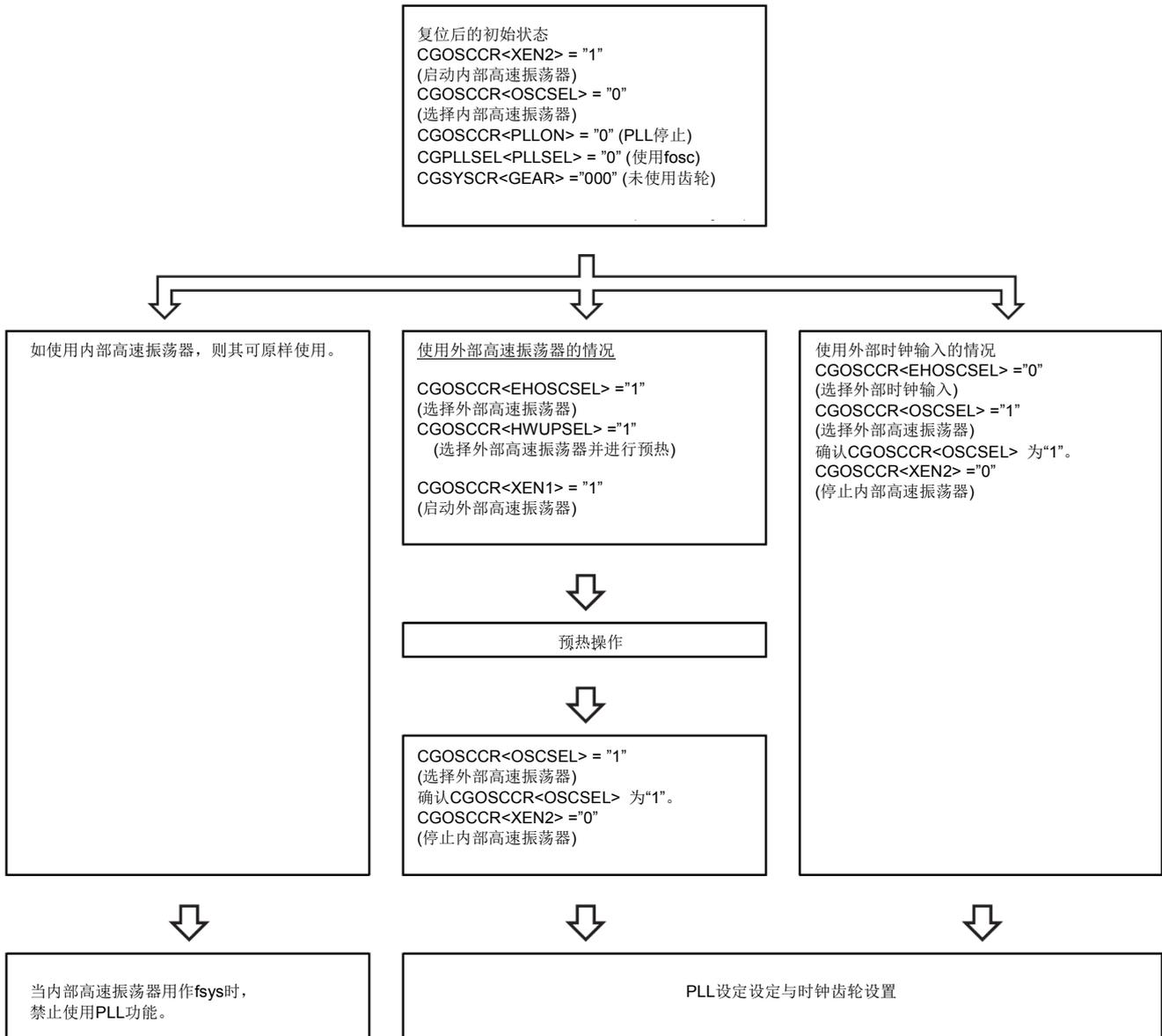
↑ 复位后的初始值

6.3.6.1 时钟设置

可通过CGOSCCR选择该系统时钟。在选中该时钟之后，即可利用PLLSEL和CGOSCCR完成PLL设置（如有必要）。并且用CGSYSCR设置时钟齿轮。

时钟设置顺序如下所示。

时钟设置顺序



6.3.6.2 在使用外部振荡器时

本产品解除复位后自内部高速振荡器激活。如使用外部高速振荡器和时钟乘法电路(PLL)，则其可按照”6.3.5 时钟乘法电路(PLL)”和”6.3.6.1 时钟设置”的程序对其进行设置。

用外部高速振荡器和时钟倍频电路时的切换如下图所示。

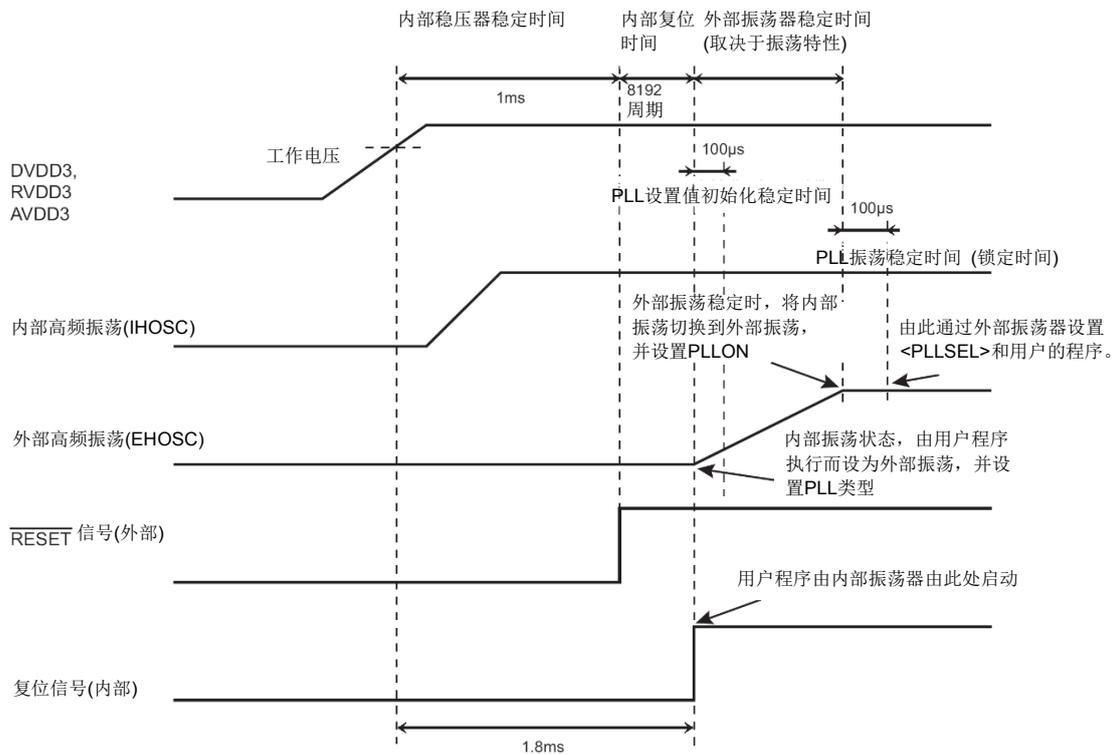


图 6-2 在用PLL设置高速振荡器时进行转换

6.3.7 预分频器时钟控制

外设功能(TMRB,SIO/UART)带有一个时钟划分用预分频器。在将时钟φT0 输入到各预分频器时，可按照CGSYSCR<PRCK[2:0]>中的设置，划分CGSYSCR<FPSEL>中的所指定的该”fperiph”时钟。

复位后，fperiph/1 即被选为φT0。

注：在定时计数器或其它外设功能运行期间，不要切换该时钟齿轮。

6.4 模式与模式转换

6.4.1 模式转换

IDLE模式和STOP1 模式能用作低功耗模式，该模式通过停止处理器内核的运行而降低功耗。

图 6-3 给出了模式转换图。

有关退出时睡眠的详细资料，请参看《技术参考手册》。

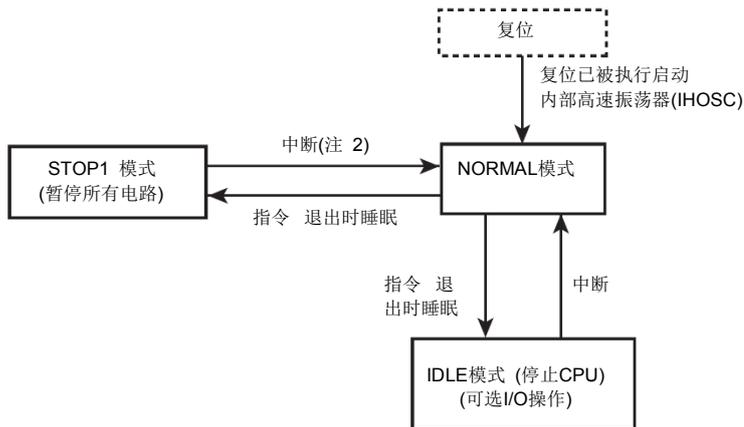


图 6-3 模式转换图

注 1: 在从STOP1 模式返回时，相关的位即CGPLLSEL<PLLSEL>、CGOSCCR<HWUPSEL>、<OSCSEL>、<XEN2>、<XEN1>和<PLLON>即被初始化，但CGOSCCR<WUODR[11:0]>不会被初始化。

注 2: 在从该STOP1 模式返回时，其转移至中断系数的中断服务程序。

6.5 操作模式

6.5.1 NORMAL 模式

该模式可利用高速时钟运行该CPU内核与外设硬件。
在复位之后，其即被转入NORMAL模式。

6.6 低功耗模式

TMPM037FWUG有低功耗模式：IDLE，STOP1。如需切换到低功耗模式，请在系统控制寄存器CGSTBYCR<STBY[2:0]>内设置模式，并执行WFI（等待中断）指令。在这种情况下，通过执行复位或生成中断即可解除该模式。在通过中断进行解除时，必须提前做好设置。有关详细资料见“异常”一节。

注 1: TMPM037FWUG不提供任何可用于解除低功耗模式的事件。禁止通过执行WFE(等待事件)指令转换至低功耗模式。AB

注 2: TMPM037FWUG不支持用Cortex-M0 内核中的SLEEPDEEP位配置的低功耗模式。禁止设置该系统控制寄存器的<SLEEPDEEP> 位。

IDLE和STOP1 模式的特性如下。

6.6.1 IDLE 模式

在该模式下，仅CPU被停止。每个外设功能在其控制寄存器中都设有一位用来启用或禁用空闲模式下的运行。在进入IDLE模式时，在IDLE模式下其运行被禁止的各外设功能停止运行，并保持在当时的状态。

以下外设功能可在该IDLE模式下被启用或被禁用。有关设置的详细资料，见各外设功能相关节次。

- 16-位定时器/事件计数器(TMRB)
- 16-位定时器(TMR16A)
- 串行通道(SIO/UART)
- 模拟数字转换器(ADC)
- 看门狗定时器(WDT)

注：在IDLE模式下，注意看门狗定时器的功能不能被CPU清除。

6.6.2 STOP1 模式

在STOP1 模式下，所有内部电路包括内部振荡器均被停止。在解除该STOP1 模式之后，该内部振荡器可触发该时钟，然后转入Normal模式。

该STOP1 模式可通过设置该端口寄存器，使能选择该引脚状态。表 6-3 给出了STOP1 模式下的引脚状态。

表 6-3 STOP1 模式下的引脚状态

功能	引脚名称	I/O	STOP1
控制引脚	$\overline{\text{RESET}}_{\text{MODE}}$	输入	o
振荡器	X1/ EHCLKIN	输入	x
	X2	输出	"高"电平输出
端口	PB1 (SWCLK) (调试I/F设置, 适用于 PxFRn<PxmFn>="1"的情形)	输入	取决于PxIE[m]
	PB2 (SWDIO) (调试I/F设置, 适用于 PxFRn<PxmFn>="1"的情形)	输入	取决于PxIE[m]
		输出	在数据有效时启用 在数据无效时禁用
	PB5, PB6, PB7, PE5, PE6, PE7 (INT0~5) (中断设置, 适用于 PxFRn<PxmFn>="1",且 PxIE<PxmIE>="1"的情形)	输入	o
	若使用非上述引脚	输入	取决于PxIE[m]。
输出		取决于PxCR[m]。	

o: 有效输入或输出。

x: 无效输入或输出。

6.6.3 低功耗模式设置

可通过待机控制寄存器CGSTBYCR<STBY[2:0]>的设置，指定该低功耗模式。

表 6-4 给出了<STBY[2:0]>中的模式设置。

表 6-4 低功耗模式设置

模式	CGSTBYCR <STBY[2:0]>
STOP1	001
IDLE	011

注：除了以上<STBY[2:0]>中的以外，不要设置任何值。

6.6.4 各模式下的操作状态

表 6-5 给出了各模式下的操作状态。

表 6-5 各模式下的操作状态

模块	NORMAL 内部高速振荡器 (IHOSC)	NORMAL 外部高速振荡器 (EHOSC)	IDLE 内部高速振荡器 (IHOSC)	IDLE 外部高速 振荡器 (EHOSC)	STOP1 (注 1)
处理器内核	o	o	-	-	-
DMAC	o	o	o	o	-
IO端口	o	o	o	o	o
SIO/UART	o	o	Δ	Δ	-
I2C	o	o	o	o	-
TMRB	o	o	Δ	Δ	-
TMR16A	o	o	Δ	Δ	-
WDT	o	o	Δ(注 3)	Δ(注 3)	-
10-位ADC	o	o	Δ	Δ	-
CG	o	o	o	o	o
PLL	o	o	Δ	Δ	-
外部高速振荡器(EHOSC)	o	o	Δ	o	-
LVD	o	o	o	o	-
内部高速振荡器(IHOSC)	o	o(注 2)	o	o(注 2)	-
主RAM	o	o	o	o	o

o: 在目标模式时, 操作可用。

-: 当切换到目标模式时, 模块时钟自动停止。

Δ: 在目标模式时, 能通过软件选择启用或禁用模块操作。

注 1: 在转换为STOP1 模式之前, 应停止“-”的外设功能。其可通过停止AD转换器的基准电压来减小泄漏电流。

注 2: 在复位或STOP1 模式被解除后, 时钟由内部高速振荡器提供。

注 3: 在IDLE模式下, 注意看门狗计时器的功能不能被CPU清除。

6.6.5 解除低功耗模式

可通过中断请求，非屏蔽中断 (NMI)或复位解除低功耗模式。可通过所选择的低功耗模式，确定可使用的解除源。

详细资料见表 6-6。

表 6-6 在各模式下解除源

低功耗模式		IDLE	STOP1	
解除源	中断	INT0~5 (注 2)	o	o
		INTTB0~7	o	x
		INTTTMR16A0~1	o	x
		INTCAP00~71	o	x
		INTRX0~4, INTTX0~4	o	x
		INTI2C0	o	x
		INTAD/INTADHP/INTADM0~1	o	x
		INTDMAC0TC INTDMAC0ERR	o	x
	SysTick中断	o	x	
	不可屏蔽中断(INTWDT)	o	x	
	不可屏蔽中断(INLVD)	o	x	
	RESET ($\overline{\text{RESET}}$ 引脚)	o	o	

o: 在该模式被解除之后，启动中断处理 (该复位可初始化LSI)

x: 不可用

注 1: 如转入该低功耗模式，可设置CPU以禁止除该解除解除源以外的所有中断。否则，未规定的中断会执行解除。

注 2: 在通过中断电平模式解除IDLE,STOP1 模式时，在中断处理开始前应保持该电平。若电平在此之前改变，则正确的中断处理无法启动。

通过中断请求解除

通过中断解除低功耗模式时，必须提前设置CPU来检测中断。除CPU中的设置外，还必须设置时钟发生器，以检测用于解除STOP1模式的中断。

通过不可屏蔽中断(NMI)解除

NMI有两种形式：WDT中断(INTWDT), LVD中断(INTLVD)。

INTWDT,INTLVD仅可用于IDLE模式。

通过复位解除

任何低功耗模式都可以通过 $\overline{\text{RESET}}$ 引脚进行解除。

解除之后该模式切换到NORMAL模式，所有寄存器被初始化，过程与正常复位相同。

注意经复位返回到STOP1 模式不会引起自动预热。应输入和冷复位一样的复位信号。

通过SysTick中断解除

SysTick中断只能在IDLE模式下使用。

详见”中断”。

6.6.6 预热

在进行模式推移时，可能会要求进行预热，以使得内部振荡器可提供稳定振荡。

在进行从STOP1 到NORMAL的模式转换时，暖机计数器和内部振荡器会被自动激活。然后在经过一段预热时间后，系统时钟输出启动。

为了进入STOP1 模式，在执行指令前，必须在CGOSCCR<WUODR[11:0]>中设置预热时间。

注：在从STOP1 模式返回时，相关的位即CGPLLSEL<PLLSEL>、CGOSCCR<HWUPSEL>、<OSCSSEL>、<XEN2>、<XEN1>和<PLLON>即被初始化，但CGOSCCR<WUODR[11:0]>不会被初始化。

表 6-7 给出了各模式转换的暖机设置是否有必要等情形。

表 6-7 模式转换时的暖机设置

模式推移	预热设置
NORMAL → IDLE	不需要
NORMAL → STOP1	不需要
IDLE → NORMAL	不需要
STOP1 → NORMAL	自动预热(注)

注：在执行复位解除时，不会执行自动暖机。输入某复位，直至振荡器稳定。

6.6.7 模式转换时的时钟操作

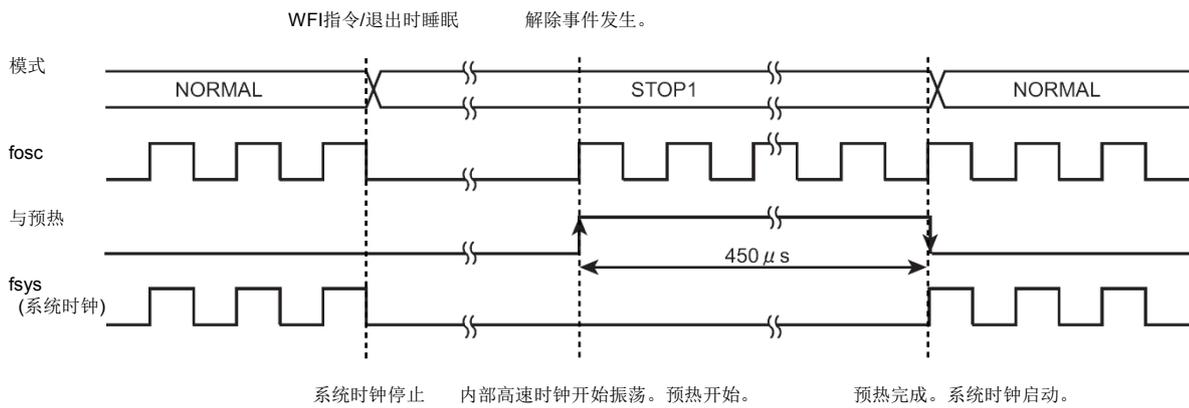
模式切换时的时钟操作如下。

6.6.7.1 运行模式的转换：NORMAL → STOP1 → NORMAL

在STOP1 模式返回NORMAL模式时，会自动触发预热。

在这种情况下，需将暖机设置CGOSCCR<WUODR[11:0]>=0x119为 片装闪存ROM进入STOP1 模式前的稳定时间(450μs)。

在执行复位解除时，不会执行自动暖机。输入复位，直至振荡器稳定。



系统时钟停止 内部高速时钟开始振荡。预热开始。 预热完成。系统时钟启动。

6.6.8 转换为低功耗模式时的注意事项

6.6.8.1 MCU 进入 IDLE 或 STOP1 模式时的情况

1. 在MCU进入STOP1 模式之前，用CGOSCCR<HWUPSEL>选择时钟，其与用CGOSCCR<OSCSEL>选择的时钟相同，让暖机计数器和fosc使用相同的源时钟。
2. 非可屏蔽中断仅在IDLE模式下用于解除。
3. 不要将非可屏蔽中断用作STOP1 模式的解除系数。在MCU进入STOP1 模式之前，禁止非可屏蔽中断，并规定如下：
(停止该看门狗计时器，停止该LVD)
4. 如果MCU未进入STOP1 模式，则CGOSCCR<HWUPSEL>, <OSCSEL>, <XEN2>, <XEN1>, <PLLSEL>和CGPLLSEL <PLLON>不会被初始化。在进入STOP1 模式之前，这些寄存器保持在此前的状况。

7. 异常

本节对各异常的特点，类型与处理进行说明。

异常与CPU内核之间有着密切的关系。如有必要，请参看《Cortex-M0 技术参考手册》。

7.1 概述

异常造成CPU停止当前正执行的进程并处理另一个进程。

有两类异常：一种异常是在出现某个错误条件时或执行生成异常的指令时产生的。另一种由硬件产生的异常，例如来自某个外部引脚或外设功能的中断请求信号。

所有异常都可以按照相应优先级通过CPU内的嵌套向量中断控制器(NVIC)处理。在发生异常时，CPU会将当前状态存储到该堆栈并转到相应的中断服务程序(ISR)。一旦该ISR完成，此前被存储该堆栈的信息即自动恢复。

7.1.1 异常类型

以下异常类型存在于Cortex-M0。

有关各异常的详细说明，请参看《Cortex-M0技术参考手册》。

复位

不可屏蔽中断(NMI)

硬故障

SVCall(管理程序调用)

PendSV

SysTick

外部中断

7.1.2 处理流程图

以下给出了异常/中断的处理方式。在下列说明中，表示硬件处理。表示软件处理。

将在本节后文部分对各步骤进行描述。

处理	说明	参见
由 CG/CPU 进行检测	CG/CPU 检测到异常请求。	7.1.2.1
↓		
由CPU进行处理	CPU 处理该异常请求。	7.1.2.2
↓		
转入ISR	CPU 转入到相应的中断服务程序(ISR)。	
↓		
ISR 的执行	进行必要的处理。	7.1.2.3
↓		
从异常返回	CPU 转入到另一 ISR，或返回到前一程序。	7.1.2.4

7.1.2.1 异常请求与检测

(1) 异常发生

异常源包括由CPU指令执行，存储器存取，以及来自外部中断引脚或外设功能的中断请求。

当CPU执行一个导致异常的指令时，或在指令执行过程中出现错误时，会出现异常。

另外，当从从不执行(XN)区中提取指令或非法访问故障区时，也会出现异常。

外部中断引脚或外设功能可导致生成中断请求。对于用于解除待机模式的中断而言，必须在时钟发生器中进行相关设置。详情参见”7.5 中断”。

(2) 异常检测

如果同时发生多个异常，则CPU会取优先级最高的异常。

表 7-1 给出了各异常的优先级。“可配置”表示异常的优先级可以手工指定。内存管理,总线故障和使用故障异常可以根据需要启用或禁用。

如果发生某被禁用异常，其将被作为硬故障接受处理。

表 7-1 异常类型和优先级

编号	异常类型	优先级	说明
1	复位	-3 (最高)	复位引脚, WDT, POR, LVD或SYSRETRQ
2	非屏蔽中断	-2	WDT
3	硬故障	-1	因正处理一个更高优先级的故障或因被禁用而无法激活的故障
4~10	保留	-	
11	SVCcall	可配置	用SVC指令进行系统服务调用
12~13	保留	-	
14	PendSV	可配置	未决系统服务请求
15	SysTick	可配置	源自系统计时器的通知
自16	外部中断	可配置	外部中断引脚或外设功能(注 2)

注：各产品外部中断的源与数目各不相同。有关详细说明，见”7.5.2 中断源列表”。

(3) 优先级设置

优先级

外部中断优先级可被设置为中断优先权寄存器，而其它异常会被设置到系统处理器优先寄存器中的 <PRI_n>位。

TMPM037FWUG具备<PRI_n>的 2-位。

该<PRI_n>的配置为两个位，因此，可在0至3的范围内该优先级。最高优先级为”0”。如果相同优先级的多个项目同时存在，则值越小优先级越高。

7.1.2.2 异常处理和分支转移至中断服务程序(预清空)

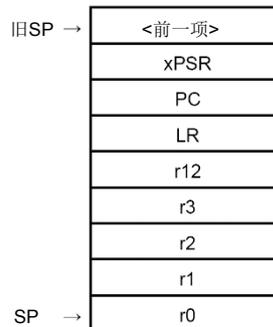
在异常发生时，CPU会中止当前正在执行的处理，并转到中断服务程序。这就是所谓的”先占”。

(1) 堆栈

在CPU检测到异常时，其会按以下顺序，将以下八个寄存器的内容压入到该堆栈：

1. 程序计数器 (xPSR)
2. 程序状态寄存器 (PC)
3. 链路寄存器(LR)
4. r12
5. r3 ~ r0

在堆栈推送结束之前SP由 8 个字确定。寄存器内容推送后的堆栈状态如下所示：



(2) 取出ISR

CPU启用指令以取出该中断处理，并将数据存储至寄存器。

为每个异常分别准备包含ISR项地址的一个向量表。

该向量表还应包含该主堆栈的初始值。

(3) 迟后到达

如果CPU在执行上一异常的ISR之前检测到优先级较高的异常，则CPU会首先处理该优先级较高的异常。这就是所谓的“迟来”。

后至异常会导致CPU提取新的向量地址并转发到相应的ISR，但是CPU不会重新将寄存器内容推送到堆栈。

(4) 向量表

该向量表的配置如下所示。

始终应设置首四字(栈顶地址,复位ISR地址,NMI ISR地址,以及硬故障ISR地址)。

对于其他异常，可根据需要设置ISR地址。

偏移量	异常	内容	设置
0x00	复位	主堆栈的初始值	要求
0x04	复位	ISR地址	要求
0x08	非屏蔽中断	ISR地址	要求
0x0C	硬故障	ISR地址	要求
0x10 ~ 0x28	保留	-	-
0x2C	SVCcall	ISR地址	要求
0x30 ~ 0x34	保留	-	-
0x38	PendSV	ISR地址	要求
0x3C	SysTick	ISR地址	要求
0x40	外部中断	ISR地址	要求

7.1.2.3 执行中断服务程序(ISR)

ISR可执行相应异常的必要处理。ISRs必须由用户编制。

ISR需要包括清除中断请求的代码，这样返回到正常程序执行时不会再出现同样的中断。

关于中断处理的详细信息，参见“7.5节中断”。

如果在当前异常的ISR执行期间发生优先级较高的异常，则CPU会放弃当前正在执行的ISR，并检修最近检测到的异常。

7.1.2.4 异常出口

(1) 从ISR返回后的执行

在从ISR返回时，CPU会采取以下动作的其中之一：

尾链

如果存在待决异常，且无堆叠异常，或待决异常的优先级高于所有堆叠异常的优先级，则CPU返回到待决异常的ISR中。

此时，当CPU从一个ISR退出进入到另一个ISR时，会跳过 8 个寄存器的入栈和出栈。这就是所谓的“尾链”。

返回到上一堆栈ISR

如果不存在待决异常，或者堆叠异常中的最高优先级高于待决异常中的最高优先级，则CPU返回到最后进入堆栈的ISR。

返回到上一个程序

如果不存在待决或已堆栈异常，则CPU会返回前一程序。

(2) 异常出口时序

在从某ISR返回时，CPU会执行以下操作：

弹出式寄存器

从该堆栈弹出八个寄存器 (PC, xPSR, r0 ~ r3, r12与LR) ，并调节该SP。

加载当前激活的中断编号

从已堆栈xPSR加载当前的活动中断号。CPU会用其进行跟踪，并确定返回到哪个中断。

选择SP

如果返回到某异常(处理器模式)，则SP是SP_main。如果返回到线程模式，则SP可以为主要SP或进程SP。

7.2 复位异常

可从以下源生成复位异常。

使用时钟发生器的该复位标志(CGRSTFLG)寄存器标识某个复位的源。

外部复位引脚

在某外部复位引脚从“低”变为“高”时，会发生复位异常。

POR异常

POR具备复位生成功能。有关详细说明，见POR相关节次。

由WDT导致的复位异常

看门狗计时器(WDT)具备复位生成功能。有关详细资料见WDT相关节次。

由SYSRESETREQ导致的复位异常

通过设置NVIC的应用中断和复位控制寄存器的SYSRESETREQ位可以进行复位。

通过LVD复位异常

低压检测电路 (LVD)具备复位生成功能。有关详细说明，见LVD相关节次。

7.3 非屏蔽中断(NMI)

可从以下源生成非可屏蔽中断。

可利用时钟发生器的NMI标志(CGNMIFLG)寄存器，标识非屏蔽中断的源。

WDT非屏蔽中断

看门狗计时器(WDT)具备非屏蔽中断生成功能。有关详细资料见WDT相关节次。

通过LVD实现非可屏蔽中断

低压检测电路(LVD)具备复位生成功能。有关详细说明，见LVD相关节次。

7.4 SysTick

SysTick具备中断功能(采用CPU的系统计时器)。

当设置SysTick重新加载值寄存器并在SysTick控制和状态寄存器中启用SysTick特点时，计数器会加载重新加载值寄存器的设置值，开始向下计数。当计数器达到“0”，会出现SysTick异常。可以将异常待决，并根据某个标志了解计时器合适达到“0”。

注：在本产品中，由(CGOSCCR<OSCSSEL><EHOSCSSEL>)按 32 选择的fosc被用作外部基准时钟。

7.5 中断

本节对中断的路径，源与必要设置进行说明。

源自各中断源的中断信号可将中断请求通知CPU。其可设置各中断的优先级，并处理具备最高优先级的中断请求。

可通过时钟发生器，将待机模式清除用中断请求通知CPU。因此时钟生成器必须正确设置。

7.5.1 中断源

7.5.1.1 中断路径

中断请求路径如图 7-1 所示。

未被用于解除待机的外设功能所发布的中断，被直接输入到该CPU (路径 1)。

对于解除备用模式的外设功能（路径 2）发出的中断以及来自外部中断引脚（路径 3）的中断，输入到时钟生成器，并通过备用解除逻辑输入到CPU（路径 4 和 5）。

如果源自外部中断引脚的中断未被用于解除待机，则其会被直接输入到CPU(不通过待机解除逻辑)(路径 6)。

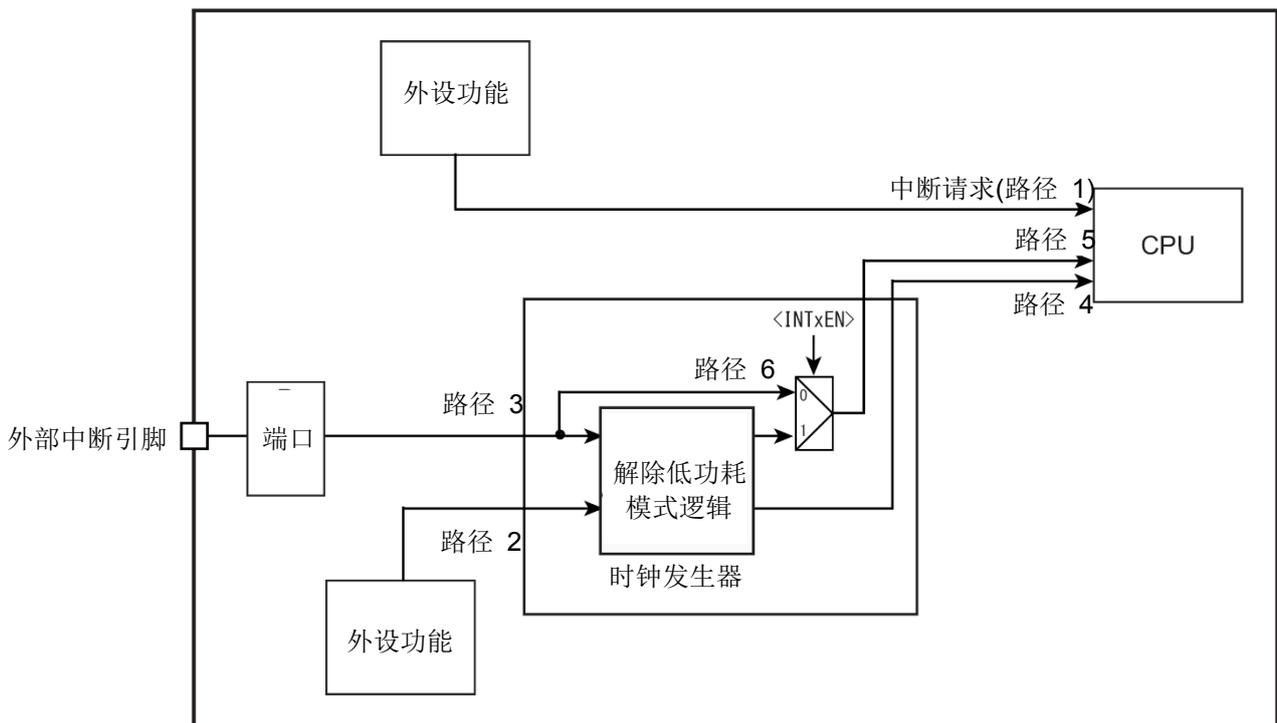


图 7-1 中断路径

7.5.1.2 生成

可从被指定为中断源的某外部引脚或外设功能，或通过设置NVIC的中断设置待决寄存器，生成一个中断请求。

- 来自外部引脚
设置该端口控制寄存，使得该外部引脚可用作中断功能引脚。
- 来自外设函数
设置该外设功能，使之可输出中断请求。
有关详细资料见各外设功能相关章节。
- 通过设置中断挂起寄存器(强制挂起)
通过设置中断设置挂起寄存器的相关位，可生成一个中断请求。
CPU会将该中断请求信号的”H”电平识别为中断请求。

7.5.1.3 发送

可将源自某外部引脚或外设功能的某中断信号直接发送给CPU，但其被用于退出待机模式的情形除外。

源自可用于待机模式清除的中断源的中断请求，会通过时钟发生器被发送到CPU。对于这些中断源，必须预先在时钟发生器中进行适当的设置。对于未被用于退出待机模式的外部中断，则无需设置时钟发生器即可使用。

7.5.1.4 使用外部中断引脚时的注意事项

如果操作员使用外部中断，则应注意以下内容，以免生成意外中断。

如果输入被禁用($PxIE < PxmIE = 0$)，则源自外部中断引脚的输入为”高”。此外，如果外部中断未被用作待机解除用触发器(“图 7-1 中断路径”的路径 6)，则来自外部中断引脚的输入信号会被直接发送到CPU。因为CPU将”高”输入当作中断，如果CPU启用相应的中断做为输入时被禁用，会出现中断。

将中断引脚输入设置为”低”并启用，即可在未将外部中断设置为待机触发信号的情况下使用该外部中断。即可启用CPU中断。

7.5.2 中断源列表

表 7-2给出了中断源列表。

表 7-2 中断源列表

编号	中断源		用以解除低功耗模式所需的激活电平					CG中断模式 控制寄存器
			“低” 电平	“高” 电平	上升沿	下降沿	两边沿	
0	INT0	外部中断引脚0	o	o	o	o	o	CGIMCGA
1	INT1	外部中断引脚1	o	o	o	o	o	
2	INT2	外部中断引脚2	o	o	o	o	o	
3	INT3	外部中断引脚3	o	o	o	o	o	
4	INT4	外部中断引脚4	o	o	o	o	o	
5	INT5	外部中断引脚5	o	o	o	o	o	CGIMCGB
6	INTRX0	串行接收(通道 0)						
7	INTTX0	串行传输(通道 0)						
8	INTRX1	串行接收(通道 1)						
9	INTTX1	串行传输(通道 1)						
10	保留	-						
11	保留	-						
12	INTI2C0	I2C0中断						
13	INTDMAC							
	INTDMACTC	DMAC传输完成中断						
	INTDMACERR	DMAC传输错误中断						
14	INTT16A0	16-位TMR16A匹配检测(通道 0)						
15	INTT16A1	16-位TMR16A匹配检测(通道 1)						
16	INTTMRB0	16-位TMRB(通道 0)						
	INTTB0	匹配检测/溢出						
	INTTB0CAP0	输入捕捉 0						
	INTTB0CAP1	输入捕捉 1						
17	INTTMRB1	16-位TMRB(通道 1)						
	INTTB1	匹配检测/溢出						
	INTTB1CAP0	输入捕捉 0						
	INTTB1CAP1	输入捕捉 1						
18	INTTMRB2	16-位TMRB(通道 2)						
	INTTB2	匹配检测/溢出						
	INTTB2CAP0	输入捕捉 0						
	INTTB2CAP1	输入捕捉 1						
19	INTTMRB3	16-位TMRB(通道 3)						
	INTTB3	匹配检测/溢出						
	INTTB3CAP0	输入捕捉 0						
	INTTB3CAP1	输入捕捉 1						
20	INTTMRB4	16-位TMRB(通道 4)						
	INTTB4	匹配检测/溢出						
	INTTB4CAP0	输入捕捉 0						
	INTTB4CAP1	输入捕捉 1						
21	INTTMRB5	16-位TMRB(通道 5)						
	INTTB5	匹配检测/溢出						
	INTTB5CAP0	输入捕捉 0						
	INTTB5CAP1	输入捕捉 1						

表 7-2 中断源列表

编号	中断源		用以解除低功耗模式所需的激活电平					CG中断模式控制寄存器
			“低”电平	“高”电平	上升沿	下降沿	两边沿	
22	INTTMRB6	16-位TMRB (通道 6)						
	INTTB6	匹配检测/溢出						
	INTTB6CAP0	输入捕捉 0						
	INTTB6CAP1	输入捕捉 1						
23	INTTMRB7	16-位TMRB (通道 7)						
	INTTB7	匹配检测/溢出						
	INTTB7CAP0	输入捕捉 0						
	INTTB7CAP1	输入捕捉 1						
24	INTRX2	串行接收(通道 2)						
25	INTTX2	串行传输(通道 2)						
26	INTRX3	串行接收(通道 3)						
27	INTTX3	串行传输(通道 3)						
28	INTRX4	串行接收(通道 4)						
29	INTTX4	串行传输(通道 4)						
30	INTADC							
	INTADHP	最高优先级AD转换完成中断						
	INTADM0	AD转换监控功能中断 0						
	INTADM1	AD转换监控功能中断 1						
31	INTAD	AD转换完成中断						

7.5.2.1 作用电平

有效电平表明中断源信号的哪种改变会导致中断。CPU识别“高”位的中断信号是中断。系统将对从周围函数发送至CPU的中断信号配置到“高电平”输出，表示一次中断请求。

有效电平，其可被用作待机解除用触发信号。来自外设功能的中断请求设置为上升边或下降边触发。源自中断引脚的中断请求可被设置为电平敏感型(“高”或“低”) 或边沿触发型(上升或下降)。

如果某中断源被用于清除待机模式，则必须设置相关时钟发生器寄存器。启用CGIMCGx<INTxEN>位，并规定CGIMCGx<EMCGx> 位中的有效电平。用户必须从表 7-2 所示的各外设功能设置中断请求的激活电平。

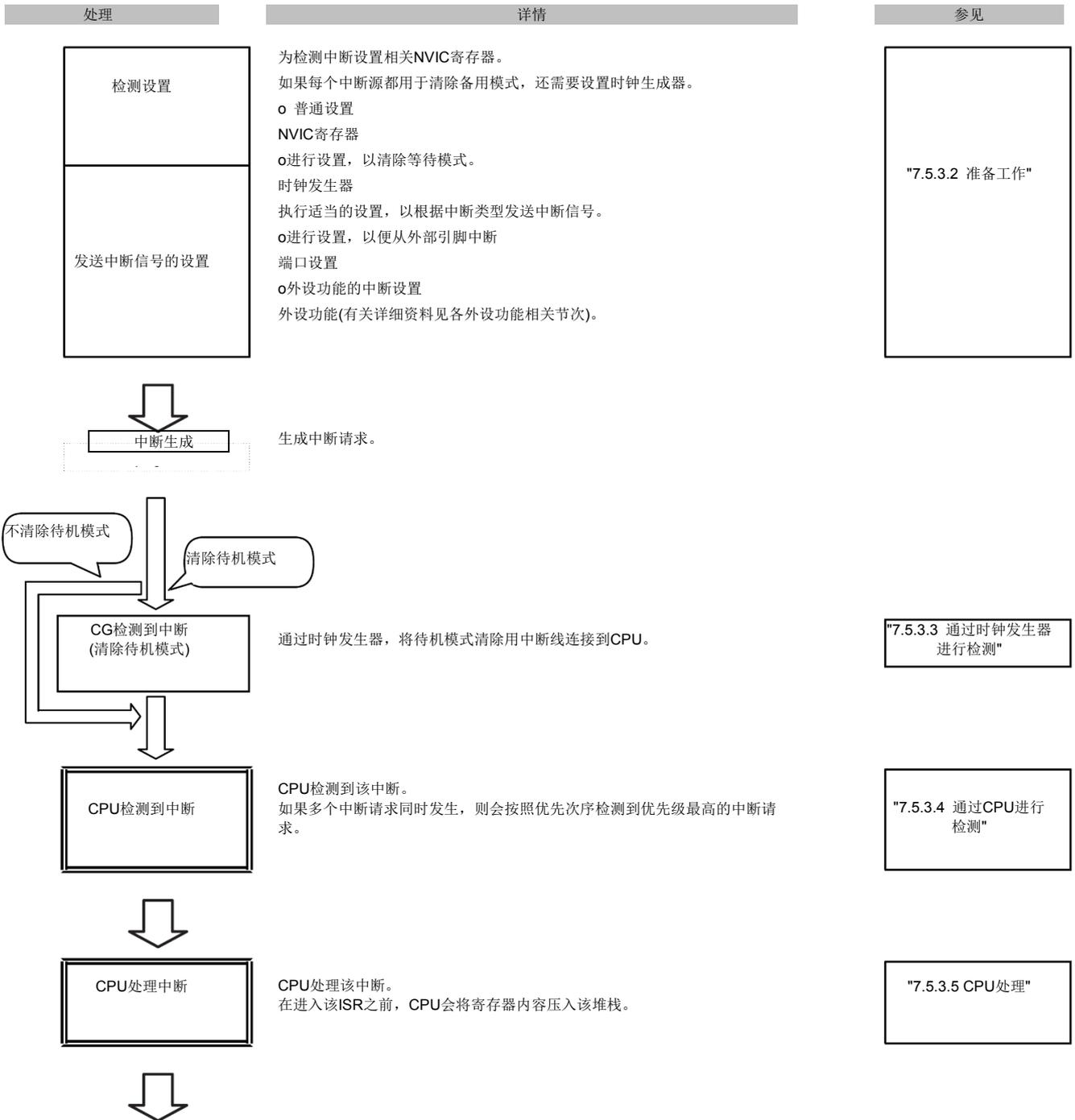
可用一个“高”电平信号将时钟发生器所检测到的中断请求通知CPU。

7.5.3 中断处理

7.5.3.1 流程图

下文给出了中断的处理方式。

在以下说明中， 表示硬件处理。 表示软件处理。





7.5.3.2 准备工作

在针对某个中断进行准备时，操作员需要注意配置顺序，以防止中途出现任何意外中断。

发起中断或变更配置时，一般需要遵守如下顺序。通过CPU禁用该中断。从CPU按最远的路径进行配置。然后通过CPU启用中断。

配置时钟发生器时，必须按照如下顺序操作，否则会导致意外中断。首先，配置预置条件。其次，清除时钟发生器中与该中断相关的数据，然后启用该中断。

以下各节均按中断处理的顺时序出，并将对其配制方法进行说明。

1. 通过CPU禁用中断
2. CPU寄存器设置
3. 预配置(1)(从外部引脚中断)
4. 预配置(2)(来自外设功能的中断)
5. 预配置(3)(中断设置挂起寄存器)
6. 配置时钟发生器
7. 由CPU启用的中断

(1) 通过CPU禁用中断

如需使CPU不接受任何中断，则可将"1"写到PRIMASK寄存器的相应位。所有中断与异常(非可屏蔽中断与硬故障除外)均会被屏蔽。

用"MSR"指令设置该寄存器。

中断屏蔽寄存器		
PRIMASK	←	"1"(中断停用)

注 1: 用户存取级不能修改PRIMASK寄存器。

注 2: 如果在"1"被设置到PRIMASK寄存器导致发生故障，则其会被作为硬故障处理。

(2) CPU寄存器设置

通过写入到NVIC寄存器的中断优先权寄存器中的<PRI_n>字段，操作员可指定某个优先级。

各中断源均具备两个位，用于指定从 0 到 3 的优先级。优先级 0 为最高优先级。如果多个中断源的优先级相同，最小编号的中断源的优先级最高。

NVIC寄存器		
<PRI_n>	←	优先级

注：“n”指相应的异常/中断。

(3) 预配置 (1) (从外部引脚中断)

将“1”设置到相应引脚的端口功能寄存器，用于从外部引脚实现中断。通过设置PxIE[m]，即可允许将该引脚用作输入端口。

端口寄存器		
PxIE<PxmlE>	←	"1"

注：x：端口号 / m：对应位

将PxIE设置为启用，即可启用相应的中断输入。注意不得启用未使用的中断。

(4) 预配置(2) (从外设功能中断)

该设置随所用外设功能的不同而不同。有关详细资料见各外设功能相关章节。

(5) 预配置 (3) (中断设置未决寄存器)

在利用该中断设置挂起寄存器生成中断时，需将“1”设置到该寄存器的对应位。

NVIC寄存器		
<SETPEND[m]>	←	"1"

注：m：对应位

(6) 配置时钟发生器

对于拟用于待机模式退出的中断源，操作员需在时钟发生器的CGIMCG寄存器中设置有效电平，并启用中断。CGIMCG寄存器可设置每个中断源。有关该激活电平，请参看“表 7-2 中断源列表”。

在启用某中断之前，清除已保留的相应中断请求。这样可避免发生意外中断。将与拟使用的中断相当的值写入到CGICRCG寄存器，即可清除相应的中断请求。有关各值见“7.6.3.3 CGICRCG(CG中断请求清除寄存器)”。

在未设置时钟发生器的情况下即可使用源自外部引脚的中断请求，但前提是其未被用于待机模式的退出。然而，“高”脉冲或“高”电平信号必须输入，便于CPU可以检测为中断请求。

另外，需注意“7.5.1.4 外部中断使用注意事项”。

时钟发生器寄存器		
CGIMCGn<EMCGm>	←	有效电平
CGICRCG<ICRCG>	←	拟使用中断的对应值
CGIMCGn<INTmEN>	←	"1"(中断已启用)

注：n：寄存器编号 / m:中断源分配的编号

(7) 由CPU启用的中断

通过CPU启用中断，如以下所述。

清除中断清除挂起寄存器中的已暂停中断。用该中断设置启动寄存器启用预定的中断。将寄存器的各位指定到单个中断源。

通过将“1”写入到该中断清除挂起寄存器的对应位，即可清除该已暂停的中断。

将“1”写入到中断设置-启用寄存器的对应位可以启用所需中断。

在该中断设置挂起寄存器设定中生成中断时，如果清除了各挂起中断，则中断触发用因数即告丢失。因此，不必进行该操作。

最后，PRIMASK寄存器被清零。

NVIC寄存器		
中断清除-未决<CLRPEND[m]>	←	"1"
中断设置启用<SETENA[m]>	←	"1"
中断屏蔽寄存器		
PRIMASK	←	"0"

注1：M：对应位

7.5.3.3 通过时钟发生器进行检测

如果用中断源退出待机模式，则会按照时钟发生器中所规定的有效电平进行检测，并通知CPU。

一旦检测到边触发中断请求，该中断请求即被保留在该时钟发生器中。程度敏感的中断请求在检测到之前必须在活动级别上保留，否则当信号电平从活动切换到不活动时中断请求会消失。

当检测到中断请求时，时钟生成器继续以“高”位向CPU发送中断信号，直到中断请求从CG中断请求清除(CGICRCG)寄存器中清除。如果在未清除该中断请求的情况下退出待机模式，则在恢复正常运行时会再次检测到同一中断。务必清除ISR中的各中断请求。

7.5.3.4 通过CPU进行检测

CPU可检测到具备最高优先级的中断请求。

7.5.3.5 CPU 处理

一旦检测到一个中断，CPU就会将xPSR, PC, LR, R12 和r3 至r0 的内容推至存储栈，然后输入ISR。

7.5.3.6 中断服务例行程序(ISR)

ISR需按照拟使用的应用进行特定的编程。本节将对服务程序编程时的建议项目，以及源清除方式进行说明。

(1) ISR期间的程序

ISR一般会按要求将寄存器内容压入到堆栈，并处理某一中断。该Cortex- M0 内核会将xPSR, PC, LR, R12 和r3 至r0 的内容自动推到该存储栈。无需对其进行额外编程。

如果需要，还可压入其它寄存器的内容。

即使正在执行某ISR，也会接收优先级较高的中断请求，以及NMI等异常。我们建议操作员压入可能会被重写的通用寄存器。

(2) 清除一个中断源

如拟用某中断源清除某待机模式，则必须用该CG中断请求清除(CGICRCG)寄存器清除各中断请求。

如果中断设置为边沿敏感，通过设置CGICRCG寄存器的相应值来清除中断请求。如果有效边再次出现，将会检测到新的中断请求。

如某中断源被设置为电平敏感型，则其会继续存在，直至其在某源时被清除。因此，必须清除该中断源。自动清除该中断源则清除了来自时钟发生器的中断请求信号。

7.6 与异常/中断相关的寄存器

7.6.1 寄存器表

本章介绍的CPU的NVIC寄存器和时钟生成器寄存器以及相关地址如下表所示：

有关基地址的详细情况，参照“内存地址图”一章的“外部功能地址表”。

NVIC寄存器 基址= 0xE000_E000

寄存器名称	地址
SysTick控制器与状态寄存器	0x0010
SysTick重新加载值寄存器	0x0014
SysTick当前值寄存器	0x0018
SysTick校准值寄存器	0x001C
中断设置使能寄存器	0x0100
中断清除-使能寄存器	0x0180
中断设置-未决寄存器	0x0200
中断清除-未决寄存器	0x0280
中断优先权寄存器	0x0400~ 0x041F
应用中断与复位控制寄存器	0x0D0C
系统处理器优先寄存器	0x0D1C, 0x0D20
系统处理器控制器与状态寄存器	0x0D24

外设功能名称:CG

寄存器名称	地址
CG中断模式控制寄存器A	CGIMCGA 0x0040
CG中断模式控制寄存器B	CGIMCGB 0x0044
CG中断请求清除寄存器	CGICRCG 0x0060
复位标志寄存器	CGRSTFLG 0x0064
NMI标志寄存器	CGNMIFLG 0x0068

7.6.2 NVIC 寄存器

7.6.2.1 SysTick 控制与状态寄存器

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

位	比特符号	类型	功能
31-17	-	R	读作"0"。
16	COUNTFLAG	R/W	0: 计时器不会计数至 0 1: 计时器计数至 0 如果在上次读取后, 计时器计数至"0", 则返回"1"。 清除SysTick控制器与状态寄存器的任何已读取部分。
15-3	-	R	读作"0"。
2	CLKSOURCE	R/W	0: 外部基准时钟(fosc/32) (注) 1: CPU时钟(fsyst)
1	TICKINT	R/W	0: 不要让SysTick处于挂起状态 1: 让SysTick处于挂起状态
0	ENABLE	R/W	0: 禁用 1: 启用 如果已设置"1", 则其会重新加载该重新加载值寄存器的值, 并开始运行。

注: 在本产品中, 由CGOSCCR<OSCSEL><EHOSCSEL>按 32 选择的fosc被用作外部基准时钟。

7.6.2.2 SysTick 重新加载值寄存器

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

位	比特符号	类型	功能
31-24	-	R	读作"0"。
23-0	RELOAD	RW	重新加载值 设置该值，使之在计时器达到"0"时加载到SysTick当前值寄存器。

7.6.2.3 SysTick 正确值寄存器

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

位	比特符号	类型	功能
31-24	-	R	读作"0"。
23-0	CURRENT	RW	[读取]当前SysTick计时器值 [写入] 清除 通过将任意值写入到该寄存器，均可将其清零。 在清除该寄存器的同时，也会清除SysTick控制器与状态寄存器的<COUNTFLAG>位。

7.6.2.4 SysTick 校准值寄存器

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

位	比特符号	类型	功能
31	NOREF	R	0: 具备基准时钟 1: 无基准时钟
30	SKEW	R	0: 校准值是 10 ms。 1: 校准值不为 10 ms。
29-24	-	R	读作"0"。
23-0	TENMS	R	校准值(注)

注：本产品不提供校准值。

7.6.2.5 中断设置启用寄存器 1

	31	30	29	28	27	26	25	24
比特符号	SETENA (中断 31)	SETENA (中断 30)	SETENA (中断 29)	SETENA (中断 28)	SETENA (中断 27)	SETENA (中断 26)	SETENA (中断 25)	SETENA (中断 24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	SETENA (中断 23)	SETENA (中断 22)	SETENA (中断 21)	SETENA (中断 20)	SETENA (中断 19)	SETENA (中断 18)	SETENA (中断 17)	SETENA (中断 16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	SETENA (中断 15)	SETENA (中断 14)	SETENA (中断 13)	SETENA (中断 12)	SETENA (中断 11)	SETENA (中断 10)	SETENA (中断 9)	SETENA (中断 8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SETENA (中断 7)	SETENA (中断 6)	SETENA (中断 5)	SETENA (中断 4)	SETENA (中断 3)	SETENA (中断 2)	SETENA (中断 1)	SETENA (中断 0)
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	SETENA	R/W	<p>中断号[31:0]</p> <p>[写入]</p> <p>1: 启用</p> <p>[读取]</p> <p>0: 禁用</p> <p>1: 启用</p> <p>每个位均与所指定的中断号对应。</p> <p>通过将“1”写入到该寄存器中的某个位，即可启用相应的中断。写入“0”无任何影响。</p> <p>通过读取各位，即可查看相应中断的启用/禁用条件。</p>

注：有关中断和中断编号的描述，见“7.5.2中断源列表”一节。

7.6.2.6 中断清除使能寄存器 1

	31	30	29	28	27	26	25	24
比特符号	CLRENA (中断 31)	CLRENA (中断 30)	CLRENA (中断 29)	CLRENA (中断 28)	CLRENA (中断 27)	CLRENA (中断 26)	CLRENA (中断 25)	CLRENA (中断 24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	CLRENA (中断 23)	CLRENA (中断 22)	CLRENA (中断 21)	CLRENA (中断 20)	CLRENA (中断 19)	CLRENA (中断 18)	CLRENA (中断 17)	CLRENA (中断 16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	CLRENA (中断 15)	CLRENA (中断 14)	CLRENA (中断 13)	CLRENA (中断 12)	CLRENA (中断 11)	CLRENA (中断 10)	CLRENA (中断 9)	CLRENA (中断 8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CLRENA (中断 7)	CLRENA (中断 6)	CLRENA (中断 5)	CLRENA (中断 4)	CLRENA (中断 3)	CLRENA (中断 2)	CLRENA (中断 1)	CLRENA (中断 0)
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	CLRENA	R/W	<p>中断号[31:0]</p> <p>[写入]</p> <p>1: 禁用</p> <p>[读取]</p> <p>0: 禁用</p> <p>1: 启用</p> <p>每个位均与所指定的中断号对应。其可用于启用中断，以及检测各中断是否被禁用。</p> <p>通过将“1”写入到该寄存器中的某个位，即可启用相应的中断。写入“0”无任何影响。</p> <p>通过读取各位，即可查看相应中断的启用/禁用条件。</p>

注：有关中断和中断编号的描述，见“7.5.2中断源列表”一节。

7.6.2.7 中断设置挂起寄存器 1

	31	30	29	28	27	26	25	24
比特符号	SETPEND (中断 31)	SETPEND (中断 30)	SETPEND (中断 29)	SETPEND (中断 28)	SETPEND (中断 27)	SETPEND (中断 26)	SETPEND (中断 25)	SETPEND (中断 24)
复位后	未定义							
	23	22	21	20	19	18	17	16
比特符号	SETPEND (中断 23)	SETPEND (中断 22)	SETPEND (中断 21)	SETPEND (中断 20)	SETPEND (中断 19)	SETPEND (中断 18)	SETPEND (中断 17)	SETPEND (中断 16)
复位后	未定义							
	15	14	13	12	11	10	9	8
比特符号	SETPEND (中断 15)	SETPEND (中断 14)	SETPEND (中断 13)	SETPEND (中断 12)	SETPEND (中断 11)	SETPEND (中断 10)	SETPEND (中断 9)	SETPEND (中断 8)
复位后	未定义							
	7	6	5	4	3	2	1	0
比特符号	SETPEND (中断 7)	SETPEND (中断 6)	SETPEND (中断 5)	SETPEND (中断 4)	SETPEND (中断 3)	SETPEND (中断 2)	SETPEND (中断 1)	SETPEND (中断 0)
复位后	未定义							

位	比特符号	类型	功能
31-0	SETPEND	RW	<p>中断号[31:0]</p> <p>[写入]</p> <p>1: 挂起</p> <p>[读取]</p> <p>0: 未挂起</p> <p>1: 挂起</p> <p>各位对应于指定的编号，强迫各中断进入挂起状态，并可确定当前处于挂起状态的中断。</p> <p>通过将“1”写入到该寄存器中的某个位，即可使相应的中断进入挂起状态。不过，对于已处于挂起状态或已被禁用的中断而言，写入“1”对其无任何影响。写入“0”无任何影响。</p> <p>通过读取该位，即可返回相应中断的当前状态。</p> <p>通过将“1”写入到中断清除挂起寄存器中的某个对应位，即可清除该寄存器中的该位。</p>

注：有关中断和中断编号的描述，见“7.5.2中断源列表”一节。

7.6.2.8 中断清除挂起寄存器 1

	31	30	29	28	27	26	25	24
比特符号	CLRPEND (中断 31)	CLRPEND (中断 30)	CLRPEND (中断 29)	CLRPEND (中断 28)	CLRPEND (中断 27)	CLRPEND (中断 26)	CLRPEND (中断 25)	CLRPEND (中断 24)
复位后	未定义							
	23	22	21	20	19	18	17	16
比特符号	CLRPEND (中断 23)	CLRPEND (中断 22)	CLRPEND (中断 21)	CLRPEND (中断 20)	CLRPEND (中断 19)	CLRPEND (中断 18)	CLRPEND (中断 17)	CLRPEND (中断 16)
复位后	未定义							
	15	14	13	12	11	10	9	8
比特符号	CLRPEND (中断 15)	CLRPEND (中断 14)	CLRPEND (中断 13)	CLRPEND (中断 12)	CLRPEND (中断 11)	CLRPEND (中断 10)	CLRPEND (中断 9)	CLRPEND (中断 8)
复位后	未定义							
	7	6	5	4	3	2	1	0
比特符号	CLRPEND (中断 7)	CLRPEND (中断 6)	CLRPEND (中断 5)	CLRPEND (中断 4)	CLRPEND (中断 3)	CLRPEND (中断 2)	CLRPEND (中断 1)	CLRPEND (中断 0)
复位后	未定义							

位	比特符号	型号	功能
31-0	CLRPEND	RW	<p>中断号[31:0]</p> <p>[写入]</p> <p>1: 清除挂起中断</p> <p>[读取]</p> <p>0: 未挂起</p> <p>1: 挂起</p> <p>各位对应于指定的编号, 强迫各中断进入挂起状态, 并可确定当前处于挂起状态的中断。</p> <p>通过将"1"写入到该寄存器中的某个位, 即可清除相应的挂起中断。然而, 写入"1"对于已操作的中断无效。写入"0"无任何影响。</p> <p>通过读取该位, 即可返回相应中断的当前状态。</p>

注: 有关中断和中断编号的描述, 见"7.5.2中断源列表"一节。

7.6.2.9 中断优先权寄存器

以下给出了各中断优先权寄存器的地址(与中断号对应)。

	31	24	23	16	15	8	7	0
0xE000_E400	PRI_3		PRI_2		PRI_1		PRI_0	
0xE000_E404	PRI_7		PRI_6		PRI_5		PRI_4	
0xE000_E408	PRI_11		PRI_10		PRI_9		PRI_8	
0xE000_E40C	PRI_15		PRI_14		PRI_13		PRI_12	
0xE000_E410	PRI_19		PRI_18		PRI_17		PRI_16	
0xE000_E414	PRI_23		PRI_22		PRI_21		PRI_20	
0xE000_E418	PRI_27		PRI_26		PRI_25		PRI_24	
0xE000_E41C	PRI_31		PRI_30		PRI_29		PRI_28	

Cortex-M0 内核用两个位即可指定优先级。

以下显示针对中断数目 0至3 的中断优先寄存器的域。此中断优先寄存器对所有其它中断数目具有同等的域。未使用的位在读取时会返回”0”，且写入未使用的位将不起任何作用。

	31	30	29	28	27	26	25	24
比特符号	PRI_3		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	PRI_2		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	PRI_1		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PRI_0		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-30	PRI_3	R/W	中断数 3 的优先权
29-24	-	R	读作”0”。
23-22	PRI_2	R/W	中断数 2 的优先权
21-16	-	R	读作”0”。
15-14	PRI_1	R/W	中断数 1 的优先权
13-8	-	R	读作”0”。
7-6	PRI_0	R/W	中断数 0 的优先权
5-0	-	R	读作”0”。

7.6.2.10 应用中断和复位控制寄存器

	31	30	29	28	27	26	25	24
比特符号	VECTKEY/VECTKEYSTAT							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	VECTKEY/VECTKEYSTAT							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENDIANESS	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	SYSRESET REQ	VECTCLR ACTIVE	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	VECTKEY (已写入)/ VECTKEYST AT(已读取)	R/W	寄存器键 [写入] 写入到寄存器要求<VECTKEY>字段中的 0x5FA [读取] 读作 0xFA05
15	ENDIANESS	R/W	字节顺序: (注 1) 1: 从大到小 0: 低端在前格式
14-3	-	R	读作"0"。
2	SYSRESET REQ	R/W	系统复位请求 1=CPU输出一个 SYSRESETREQ 信号.(注 2)
1	VECTCLR ACTIVE	R/W	清除活动矢量位 1: 清除活动NMI, 故障与中断的所有状态信息。 0: 不清除。 该位可自行清除。 应用程序负责重新启动堆叠。
0	-	R	读作"0"。

注 1: 本产品仅可被用作从小到大存储器格式。

注 2: 在SYSRESETREQ被输出时, 即会对本产品执行复位。通过复位即可清除<SYSRESETREQ>。

7.6.2.11 系统处理程序优先级寄存器

以下显示了对应于每个异常的系统处理器优先寄存器的地址。

31	24 23	16 15	8 7	0
0xE000_ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12

Cortex-M0 内核用两个位即可指定优先级。

以下给出了存储器管理，总线故障与使用故障用系统处理器优先寄存器的各字段。未使用的位在读取时会返回“0”，且写入未使用的位将不起任何作用。

	31	30	29	28	27	26	25	24
比特符号	PRI_15		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	PRI_14		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	PRI_13		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PRI_12		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-29	PRI_15	R/W	SysTick的优先级
28-24	-	R	读作“0”。
23-21	PRI_14	R/W	PendSV的优先级
20-16	-	R	读作“0”。
15-13	PRI_13	R/W	保留
12-8	-	R	读作“0”。
7-5	PRI_12	R/W	保留
4-0	-	R	读作“0”。

7.6.2.12 系统处理程序控制和状态寄存器

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

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15	SVCALL PENDED	R/W	SVCALL 0: 未挂起状态 1: 进入挂起状态
14-0	-	R	读作"0"。

注：在清除或设置各激活位时用户必须高度小心，原因是清除和设置这些位不会修复栈内容。

7.6.3 时钟生成器寄存器

7.6.3.1 CGIMCGA(CG 中断模式控制寄存器 A)

	31	30	29	28	27	26	25	24
比特符号	-	EMCG3			EMST3		-	INT3EN
复位后	0	0	1	0	0	0	-	0
	23	22	21	20	19	18	17	16
比特符号	-	EMCG2			EMST2		-	INT2EN
复位后	0	0	1	0	0	0	-	0
	15	14	13	12	11	10	9	8
比特符号	-	EMCG1			EMST1		-	INT1EN
复位后	0	0	1	0	0	0	-	0
	7	6	5	4	3	2	1	0
比特符号	-	EMCG0			EMST0		-	INT0EN
复位后	0	0	1	0	0	0	-	0

位	比特符号	类型	功能
31	-	R	读作"0"。
30-28	EMCG3[2:0]	R/W	INT3待机清除请求的激活电平设置(101至111: 禁止设置) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双边沿
27-26	EMST3[1:0]	R	INT3待机清除请求的有效电平 00: - 01: 上升沿 10: 下降沿 11: 双边沿
25	-	R	读作未定义。
24	INT3EN	R/W	INT3归零输入 0: 禁用 1: 启用
23	-	R	读作"0"。
22-20	EMCG2[2:0]	R/W	INT2待机清除请求的激活电平设置(101至111: 禁止设置) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双边沿
19-18	EMST2[1:0]	R	INT2待机清除请求的有效电平 00: - 01: 上升沿 10: 下降沿 11: 双边沿
17	-	R	读作未定义。
16	INT2EN	R/W	INT2 归零输入 0: 禁用 1: 启用
15	-	R	读作"0"。

位	比特符号	类型	功能
14-12	EMCG1[2:0]	R/W	INT1待机清除请求的激活电平设置(101至111: 禁止设置) 000: “低”电平 001: “高”电平 010: 下降沿 011: 上升沿 100: 双边沿
11-10	EMST1[1:0]	R	INT1待机清除请求的有效电平 00: - 01: 上升沿 10: 下降沿 11: 双边沿
9	-	R	读作未定义。
8	INT1EN	R/W	INT1归零输入 0: 禁用 1: 启用
7	-	R	读作“0”。
6-4	EMCG0[2:0]	R/W	INT0待机清除请求的激活电平设置(101至111: 禁止设置) 000: “低”电平 001: “高”电平 010: 下降沿 011: 上升沿 100: 双边沿
3-2	EMST0[1:0]	R	INT0待机清除请求的有效电平 00: - 01: 上升沿 10: 下降沿 11: 双边沿
1	-	R	读作未定义。
0	INT0EN	R/W	INT0归零输入 0: 禁用 1: 启用

注 1: <EMSTx> 仅在上升沿与下降沿的<EMCGx[2:0]>均被设置为“100”时才有效。可通过提交<EMSTx>, 来检查待机复位有效电平。如果用CGICRCG寄存器清除了各中断, 则<EMSTx>也同时被清除。

注 2: 请首先指定供该边使用的位, 然后再指定供<INTxEN>使用的位。禁止对它们进行同时设定。

7.6.3.2 CGIMCGB(CG 中断模式控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	1	0	0	0	-	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	1	0	0	0	-	0
	15	14	13	12	11	10	9	8
比特符号	-	EMCG5			EMST5		-	INT5EN
复位后	0	0	1	0	0	0	-	0
	7	6	5	4	3	2	1	0
比特符号	-	EMCG4			EMST4		-	INT4EN
复位后	0	0	1	0	0	0	-	0

位	比特符号	类型	功能
31	-	R	读作"0"。
30-28	-	R/W	写入"000"。
27-25	-	R	读作未定义。
24	-	R/W	写入"0"。
23	-	R	读作"0"。
22-20	-	R/W	写入"000"。
19-17	-	R	读作未定义。
16	-	R/W	写入"0"。
15	-	R	读作"0"。
14-12	EMCG5[2:0]	R/W	INT5待机清除请求的激活电平设置(101至111: 禁止设置) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双边沿
11-10	EMST5[1:0]	R	INT5待机清除请求的有效电平 00: - 01: 上升沿 10: 下降沿 11: 双边沿
9	-	R	读作未定义。
8	INT5EN	R/W	INT5归零输入 0: 禁用 1: 启用
7	-	R	读作"0"。
6-4	EMCG4[2:0]	R/W	INT4待机清除请求的激活电平设置(101至111: 禁止设置) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双边沿
3-2	EMST4[1:0]	R	INT4待机清除请求的有效电平 00: - 01: 上升沿 10: 下降沿 11: 双边沿
1	-	R	读作未定义。

位	比特符号	类型	功能
0	INT4EN	R/W	INT4归零输入 0: 禁用 1: 启用

注 1: <EMSTx> 仅在上升沿与下降沿的<EMCGx[2:0]>均被设置为“100”时才有效。可通过提交<EMSTx>, 来检查待机复位有效电平。如果用CGICRCG寄存器清除了各中断, 则<EMSTx>也同时被清除。

注 2: 请首先指定供该边使用的位, 然后再指定供<INTxEN>使用的位。禁止对它们进行同时设定。

7.6.3.3 CGICRCG(C 中断请求清除寄存器)

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

位	比特符号	类型	功能
31-5	-	R	读作“0”。
4-0	ICRCG[4:0]	W	清除中断请求 0_0000: INT0 0_0001: INT1 0_0010: INT2 0_0011: INT3 0_0100: INT4 0_0101: INT5 0_0110~1_1111: 禁止设置 读作“0”。

7.6.3.4 CGRSTFLG (复位标志寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	LVDSTF	-	SYRSTF	-	WDTRSTF	PINRSTF	PONRSTF
复位后	0	-	0	-	0	-	-	1

位	比特符号	类型	功能
31-7	-	R	读作"0"。
6	LVDRSTF	R/W	LVD复位标志 [读取] 0: - 1: 从LVD复位。 [写入] 0: 清除复位标志(注 2) 1: 忽略
5	-	R	读作"0"。
4	SYSRSTF	R/W	调试复位标志 (注 1) [读取] 0: - 1: 从SYSRESETREQ复位 [写入] 0: 清除复位标志(注 2) 1: 忽略
3	-	R	读作"0"。
2	WDTRSTF	R/W	WDT复位标志 [读取] 0: - 1: 从WDT复位 [写入] 0: 清除复位标志(注 2) 1: 忽略
1	PINRSTF	R/W	$\overline{\text{RESET}}$ 引脚标志 [读取] 0: - 1: 自 $\overline{\text{RESET}}$ 引脚复位 [写入] 0: 清除复位标志(注 2) 1: 忽略
0	PONRSTF	R/W	上电复位标志 [读取] 0: - 1: 上电复位实现复位 [写入] 0: 清除复位标志(注 2) 1: 忽略

注 1: 可显示由CPU和设置位控制寄存器<SYSRESETREQ>的NVIC中的应用中断生成的复位。

注 2: 该位不会被自动清除。因此, 可将对应的位清除为"0", 从而清除之。

7.6.3.5 CGNMIFLG(NMI 标志寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	NMIFLG3	NMIFLG2	-	NMIFLG0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-5	-	R	读作"0"。
4	-	R	读作"0"。
3	NMIFLG3	R	NMI源生成标志 0: N/A 1: 在低电压返回时从LVD生成。
2	NMIFLG2	R	NMI源生成标志 0: N/A 1: 从LVD生成, 仅低于电压下降时的设置电压。
1	-	R	读作"0"。
0	NMIFLG0	R	NMI源生成标志 0: N/A 1: 从WDT中生成

8. DMA 控制器(DMAC)

8.1 概述

下表列出了其主要功能。

表 8-1 DMA控制器功能 (1 单元)

项目	功能		说明
通道数	2ch		-
DMA请求数目	16		-
DMA启动	硬件启动		以外设电路的DMA请求开始
触发器	软件启动		以写入DMACxSoftBReq寄存器开始。
总线主控装置	32 位 x 1 (AHB)		-
优先级	高: CH0 低: CH1		固定
FIFO	4 字组 x 2ch (1 字组 = 32 位)		-
总线宽度	8/16/32 位		可单独设定用于传输源和目的地。
突发量	1/4/8/16/32/64/128/256		-
传输数	高达 4095		-
地址	传输源地址	增加未增加	可以说明来源和目的地址是否应增加或不应增加。(不支持地址包。)
	传输目的地址	增加未增加	
字节存	支持小端字节序。		-
传输方式	外设-存储器 存储器-外设 存储器-存储器 外设-外设		在"存储器至存储器"被选中时, 不支持DMA启动所需的硬件启动。更多信息, 参考DMACxCnConfiguration。 在选用"外设-外设"时, 可把特殊外设指定为来源或目的地。有关所指定的外设, 请参看"产品信息"一节。
中断功能	传输结束中断(INTDMACxTC) 错误中断(INTDMACxERR)		-
特殊功能	分散/聚集功能		-

8.2 DMA 传送类型

表 8-2 DMA传送类型

编号	DMA传输方式	电路产生DMA请求	DMA请求方式	说明									
1	存储器-外设	外设(目标)	突发请求	1 字组发送时，将DMA控制器的解除突发量设置至“1”。									
2	外设-存储器	外设(源头)	突发请求/ 单一请求	如传输数据量不是突发量的整数倍，则可使用突发请求和单一请求。 如传输数据量大于或等于突发量，则忽略单一请求，使用突发请求。 如果传输数据量低于突发量，则使用单一传输。									
3	存储器-至-存储器	DMAC	无	在无DMAC请求的情况下，启用DMAC，启动数据传输。(选择存储器-存储器模式，把DMACxCnConfiguration<E> 设置为“1”) 在所有传输数据被全部传输时或DMAC通道被禁用时，DMAC停止。									
4	外设-外设	外设(源头)	突发请求/ 单一请求	<table border="1"> <thead> <tr> <th>传输大小</th> <th>源头</th> <th>目标</th> </tr> </thead> <tbody> <tr> <td>(1)丛集尺寸的整倍数</td> <td>突发请求</td> <td>突发请求</td> </tr> <tr> <td>(2)突发量的非整数倍</td> <td>突发请求/单一请求</td> <td>-</td> </tr> </tbody> </table>	传输大小	源头	目标	(1)丛集尺寸的整倍数	突发请求	突发请求	(2)突发量的非整数倍	突发请求/单一请求	-
		传输大小	源头		目标								
		(1)丛集尺寸的整倍数	突发请求		突发请求								
(2)突发量的非整数倍	突发请求/单一请求	-											
外设(目标)	突发请求												

注： 当大量的数据在存储器-存储器下传输时，建议使用低优先通道。如使用较低优先级通道，则在较低优先级通道进行传输时，可启动较高优先级通道。如使用较高优先级通道，则在较高优先级通道进行传输时，不可启动较低优先级通道。

8.3 方块图

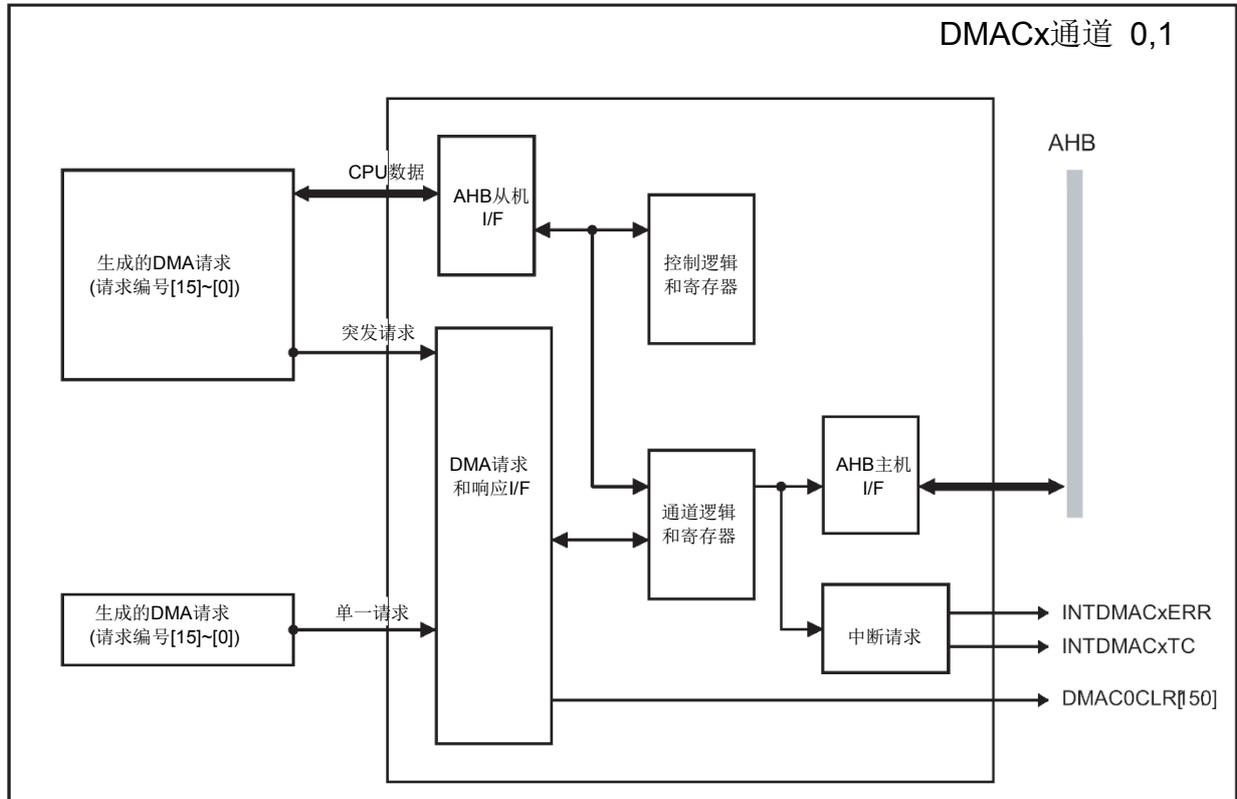


图 8-1 DMAC方块图

8.4 寄存器描述

8.4.1 DMAC 寄存器列表

每个寄存器的功能和地址如下所示：

有关基地址的详细情况，参照“内存地址图”一章的“外部功能地址表”。

寄存器名称		地址(基+)
DMAC请求状态寄存器	DMACxIntStaus	0x0000
DMAC中断端子计数状态寄存器	DMACxIntTCStatus	0x0004
DMAC中断端子计数清除寄存器	DMACxIntTCClear	0x0008
DMAC中断错误状态寄存器	DMACxIntErrorStatus	0x000C
DMAC中断错误清除寄存器	DMACxIntErrClr	0x0010
DMAC写后读中断端子计数状态寄存器	DMACxRawIntTCStatus	0x0014
DMAC写后读错误中断状态寄存器	DMACxRawIntErrorStatus	0x0018
DMAC启用通道寄存器	DMACxEnbldChns	0x001C
DMAC软件突发请求寄存器	DMACxSoftBReq	0x0020
DMAC软件单一请求寄存器	DMACxSoftSReq	0x0024
DMAC配置寄存器	DMACxConfiguration	0x0030
DMAC通道 0 源地址寄存器	DMACxC0SrcAddr	0x0100
DMAC通道 0 目的地址寄存器	DMACxC0DestAddr	0x0104
DMAC通道 0 链表项目寄存器	DMACxC0LLI	0x0108
DMAC通道 0 控制寄存器	DMACxC0Control	0x010C
DMAC通道 0 配置寄存器	DMACxC0Configuration	0x0110
DMAC通道 1 源地址寄存器	DMACxC1SrcAddr	0x0120
DMAC通道 1 目的地址寄存器	DMACxC1DestAddr	0x0124
DMAC通道 1 链表项目寄存器	DMACxC1LLI	0x0128
DMAC通道 1 控制寄存器	DMACxC1Control	0x012C
DMAC通道 1 配置寄存器	DMACxC1Configuration	0x0130

注 1：通过采用字符(32 位)读出和字符写入访问寄存器。

注 2：对于为每个通道准备的寄存器，如通道结构是相同的，则单位编号标示为“x”，通道编号标示为“n”。

注 3：当分配各个通道的寄存器是在未分配各个通道的寄存器被写入之后读出的，则要在指令之间插入机器周期，或对未分配各个通道的寄存器读出两次。

8.4.2 DMACxIntStatus (DMAC 中断状态寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	IntStatus1	IntStatus0
复位后	未定义	未定义	未定义	未定义	未定义	未定义	0	0

位	比特符号	类型	功能
31-2	-	-	写作零。
1	IntStatus1	R	DMAC通道 1 传输末端中断状态。 0: 未请求中断 1: 请求中断 在通过传输末端中断使能寄存器和错误中断使能寄存器后, DMAC中断生成的状态。在出现传输错误或计数器完成计数时, 请求中断。
0	IntStatus0	R	DMAC通道 0 中断生成状态。 0: 未请求中断 1: 请求中断 在通过传输末端中断使能寄存器和错误中断使能寄存器后, DMAC中断生成的状态。在出现传输错误或计数器完成计数时, 请求中断。

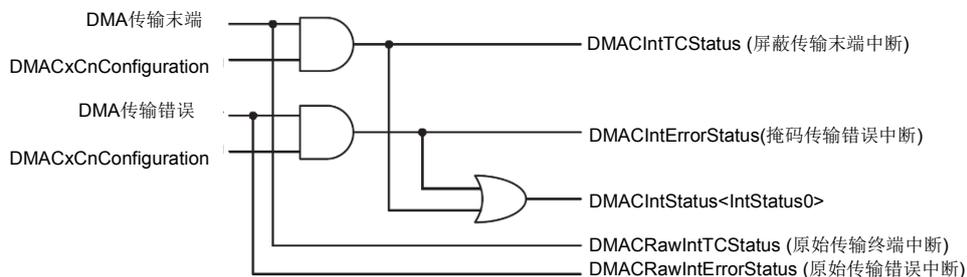


图 8-2 与中断相关的方块图

8.4.3 DMACxIntTCStatus (DMAC 中断端子计数状态寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	IntTCStatus1	IntTCStatus0
复位后	未定义	未定义	未定义	未定义	未定义	未定义	0	0

位	比特符号	类型	功能
31-2	-	-	写作零。
1	IntTCStatus1	R	DMAC通道 1 传输末端中断状态。 0: 未请求中断 1: 请求中断 后启用传输末端中断生成状态。
0	IntTCStatus0	R	DMAC通道 0 传输末端中断状态。 0: 未请求中断 1: 请求中断 后启用传输末端中断生成状态。

8.4.4 DMACxIntTCClear (DMAC 中断端子计数清除寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	IntTCClear1	IntTCClear0
复位后	未定义	未定义	未定义	未定义	未定义	未定义	0	0

位	比特符号	类型	功能
31-2	-	-	写作0。
1	IntTCClear1	W	清除DMAC通道 1 传输末端中断。 0: 不操作 1: 清除 当写入"1"时, DMACxIntTCStatus<IntTCStatus1>将被清除。
0	IntTCClear0	W	清除DMAC通道 0 传输末端中断。 0: 不操作 1: 清除 在写入"1"时, DMACxIntTCStatus<IntTCStatus0>将被清除。

8.4.5 DMACxIntErrorStatus (DMAC 中断错误状态寄存器)

	31	30	29	28	27	26	25	24	
比特符号	-	-	-	-	-	-	-	-	
复位后	未定义								
	23	22	21	20	19	18	17	16	
比特符号	-	-	-	-	-	-	-	-	
复位后	未定义								
	15	14	13	12	11	10	9	8	
比特符号	-	-	-	-	-	-	-	-	
复位后	未定义								
	7	6	5	4	3	2	1	0	
比特符号	-	-	-	-	-	-	-	IntErrStatus1	IntErrStatus0
复位后	未定义	0	0						

位	比特符号	类型	功能
31-2	-	-	写作零。
1	IntErrStatus1	R	DMAC通道 1 错误中断生成的状态。 0: 未请求中断 1: 请求中断 启用后, 显示错误中断状态。
0	IntErrStatus0	R	DMAC通道 0 错误中断生成的状态。 0: 未请求中断 1: 请求中断 启用后, 显示错误中断状态。

8.4.6 DMACxIntErrClr (DMAC 中断错误清除寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	IntErrClr1	IntErrClr0
复位后	未定义	未定义	未定义	未定义	未定义	未定义	0	0

位	比特符号	类型	功能
31-2	-	-	写作零。
1	IntErrClr1	W	清除DMAC通道 1 传输末端中断。 0: 不操作 1: 清除 在写入"1"时, DMACxIntErrorStatus<IntErrStatus1>被清除
0	IntErrClr0	W	清除DMAC通道 0 传输末端中断。 0: 不操作 1: 清除 在写入"1"时, DMACxIntErrorStatus<IntErrStatus0>被清除。

8.4.7 DMACxRawIntTCStatus (DMAC 原中断端子计数状态寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	RawIntTCS1	RawIntTCS0
复位后	未定义	未定义	未定义	未定义	未定义	未定义	0	0

位	比特符号	类型	功能
31-2	-	-	写作零。
1	RawIntTCS1	R	DMAC通道 1 启用前传输末端中断生成状态 0: 未请求中断 1: 请求中断
0	RawIntTCS0	R	DMAC通道 0 启用前传输末端中断生成状态 0: 未请求中断 1: 请求中断

8.4.8 DMACxRawIntErrorStatus (DMAC Raw 错误中断状态寄存器)

	31	30	29	28	27	26	25	24	
比特符号	-	-	-	-	-	-	-	-	
复位后	未定义								
	23	22	21	20	19	18	17	16	
比特符号	-	-	-	-	-	-	-	-	
复位后	未定义								
	15	14	13	12	11	10	9	8	
比特符号	-	-	-	-	-	-	-	-	
复位后	未定义								
	7	6	5	4	3	2	1	0	
比特符号	-	-	-	-	-	-	-	RawIntErrS1	RawIntErrS0
复位后	未定义	0	0						

位	比特符号	类型	功能
31-2	-	-	写作零。
1	RawIntErrS1	R	DMAC 通道 1 启用前错误中断的状态 0: 未请求中断 1: 请求中断
0	RawIntErrS0	R	DMAC 通道 0 启用前错误中断的状态 0: 未请求中断 1: 请求中断

8.4.9 DMACxEnbldChns (DMAC 使能通道寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义						
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	EnabledCH1	EnabledCH0
复位后	未定义	未定义	未定义	未定义	未定义	未定义	0	0

位	比特符号	类型	功能
31-2	-	-	写作 0。
1	EnabledCH1	R	DMA通道 1 使能状态。 0: 禁用 1: 启用 在完成DMACxCn控制寄存器(值变为 0)中的所有传输总次数后, 该标志被清除。
0	EnabledCH0	R	DMA通道 0 使能状态。 0: 禁用 1: 启用 在完成DMACxCn控制寄存器(值变为 0)中的所有传输总次数后, 该标志被清除。

8.4.10 DMACxSoftBReq (DMAC 软件脉冲串请求寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	15	14	13	12	11	10	9	8
比特符号	SoftBReq15	SoftBReq14	SoftBReq13	SoftBReq12	SoftBReq11	SoftBReq10	SoftBReq9	SoftBReq8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SoftBReq7	SoftBReq6	SoftBReq5	SoftBReq4	SoftBReq3	SoftBReq2	SoftBReq1	SoftBReq0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	-	写作 0。
15	SoftBReq15	R/W	通过软件提出DMA脉冲串请求(请求编号[15]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
14	SoftBReq14	R/W	通过软件提出DMA脉冲串请求(请求编号[14]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
13	SoftBReq13	R/W	通过软件提出DMA脉冲串请求(请求编号[13]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
12	SoftBReq12	R/W	通过软件提出DMA脉冲串请求(请求编号[12]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
11	SoftBReq11	R/W	通过软件提出DMA脉冲串请求(请求编号[11]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
10	SoftBReq10	R/W	通过软件提出DMA脉冲串请求(请求编号[10]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
9	SoftBReq9	R/W	通过软件提出DMA脉冲串请求(请求编号[9]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求

位	比特符号	类型	功能
8	SoftBReq8	R/W	通过软件提出DMA脉冲请求(请求编号[8]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
7	SoftBReq7	R/W	通过软件提出DMA脉冲请求(请求编号[7]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
6	SoftBReq6	R/W	通过软件提出DMA脉冲请求(请求编号[6]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
5	SoftBReq5	R/W	通过软件提出DMA脉冲请求(请求编号[5]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
4	SoftBReq4	R/W	通过软件提出DMA脉冲请求(请求编号[4]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
3	SoftBReq3	R/W	通过软件提出DMA脉冲请求(请求编号[3]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
2	SoftBReq2	R/W	通过软件提出DMA脉冲请求(请求编号[2]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
1	SoftBReq1	R/W	通过软件提出DMA脉冲请求(请求编号[1]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求
0	SoftBReq0	R/W	通过软件提出DMA脉冲请求(请求编号[0]) 读取: 0: 停止DMA突发传输 1: 运行DMA突发传输 写入: 0: 无效 1: DMA突发请求

注 1: 请勿同时通过软件和硬件执行DMA请求。

注 2: 有关DMA请求编号, 请参看“产品信息”一节。对没有突发请求的DMA请求号所对应的位清“0”。

8.4.11 DMACxSoftSReq (DMAC 软件单个请求寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	15	14	13	12	11	10	9	8
比特符号	SoftSReq15	SoftSReq14	SoftSReq13	SoftSReq12	SoftSReq11	SoftSReq10	SoftSReq9	SoftSReq8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SoftSReq7	SoftSReq6	SoftSReq5	SoftSReq4	SoftSReq3	SoftSReq2	SoftSReq1	SoftSReq0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	-	写作0。
15	SoftSReq15	R/W	通过软件提出DMA单个请求(请求编号[15]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
14	SoftSReq14	R/W	通过软件提出DMA单个请求(请求编号[14]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
13	SoftSReq13	R/W	通过软件提出DMA单个请求(请求编号[13]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
12	SoftSReq12	R/W	通过软件提出DMA单个请求(请求编号[12]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
11	SoftSReq11	R/W	通过软件提出DMA单个请求(请求编号[11]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
10	SoftSReq10	R/W	通过软件提出DMA单个请求(请求编号[10]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
9	SoftSReq9	R/W	通过软件提出DMA单个请求(请求编号[9]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
8	SoftSReq8	R/W	通过软件提出DMA单个请求(请求编号[8]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求

位	比特符号	类型	功能
7	SoftSReq7	R/W	通过软件提出DMA单个请求(请求编号[7]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
6	SoftSReq6	R/W	通过软件提出DMA单个请求(请求编号[6]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
5	SoftSReq5	R/W	通过软件提出DMA单个请求(请求编号[5]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
4	SoftSReq4	R/W	通过软件提出DMA单个请求(请求编号[4]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
3	SoftSReq3	R/W	通过软件提出DMA单个请求(请求编号[3]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
2	SoftSReq2	R/W	通过软件提出DMA单个请求(请求编号[2]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
1	SoftSReq1	R/W	通过软件提出DMA单个请求(请求编号[1]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求
0	SoftSReq0	R/W	通过软件提出DMA单个请求(请求编号[0]) 读取: 0: 停止DMA单一传输 1: 运行DMA单一传输 写入: 0: 无效 1: DMA单一请求

注 1: 请勿同时通过软件和硬件执行DMA请求。

注 2: 有关DMA请求编号, 请参看"产品信息"一节。将没有突发请求的DMA请求号所对应的位清"0"。

8.4.12 DMACx 配置(DMAC 配置寄存器)

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

位	比特符号	类型	功能
31-2	-	-	写作 0。
1	-	R/W	写作 0。
0	E	R/W	DMA电路控制 0: 停止 1: 操作 当电路停止时，DMA电路寄存器不得写入或读出。在操作DMA时，应始终设置为<E>="1"。

8.4.13 DMACxCnSrcAddr (DMAC 通道 x 源地址寄存器)

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

位	比特符号	类型	功能										
31-0	SrcAddr[31:0]	R/W	<p>设置DMA传输源地址。</p> <p>在设置前，确保对源地址和位宽度进行确认。</p> <p>下文是对源地址位宽度进行设置的限制条件。</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 50%;">源地址位宽度</th> <th style="width: 50%;">最低有效地址的设置</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">DMACxCnControl<Swidth[2:0]></td> <td></td> </tr> <tr> <td>000: 字节(8 位)</td> <td>无限制</td> </tr> <tr> <td>001: 半字符(16 位)</td> <td>设置为 2 的倍数, (0x0,0x02,0x4,0x06,0x8,0xA,0xC...)</td> </tr> <tr> <td>010: 字符(32 位)</td> <td>设置为 4 的倍数, (0x0,0x4,0x8,0xC...)</td> </tr> </tbody> </table>	源地址位宽度	最低有效地址的设置	DMACxCnControl<Swidth[2:0]>		000: 字节(8 位)	无限制	001: 半字符(16 位)	设置为 2 的倍数, (0x0,0x02,0x4,0x06,0x8,0xA,0xC...)	010: 字符(32 位)	设置为 4 的倍数, (0x0,0x4,0x8,0xC...)
源地址位宽度	最低有效地址的设置												
DMACxCnControl<Swidth[2:0]>													
000: 字节(8 位)	无限制												
001: 半字符(16 位)	设置为 2 的倍数, (0x0,0x02,0x4,0x06,0x8,0xA,0xC...)												
010: 字符(32 位)	设置为 4 的倍数, (0x0,0x4,0x8,0xC...)												

由于启用通道"n" (DMACxCnConfiguration<E>="1")会对写入寄存器内的数据进行更新，因此应在启用该通道前对DMACxCnSrcAddr进行设置。

当DMA在运行时，DMACxCnSrcAddr寄存器内的数值连续发生变化，因此读取数值是不固定的。

在传输期间，请勿更新DMACxCnSrcAddr。如需更改DMACxCnSrcAddr，在更改前，请务必禁用通道"n"(DMACxCnConfiguration<E>="0")。

8.4.14 DMACxCnDestAddr (DMAC 通道 x 目的地址寄存器)

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

位	比特符号	类型	功能								
31-0	DestAddr[31:0]	R/W	<p>设置DMA传输目的地址。</p> <p>在设置前，确保对目的地址和位宽度进行确认。</p> <p>下文是对目的地址位宽度进行设置的限制条件。</p> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="width: 50%;">目的地址位宽度 DMACxControl<Dwidth[2:0]></th> <th style="width: 50%;">最低有效地址的设置</th> </tr> </thead> <tbody> <tr> <td>000: 字节(8 位)</td> <td>无限制</td> </tr> <tr> <td>001: 半字符(16 位)</td> <td>设置为 2 的倍数, (0x0,0x02,0x4,0x06,0x8,0xA,0xC...)</td> </tr> <tr> <td>010: 字符(32 位)</td> <td>设置为 4 的倍数, (0x0,0x4,0x8,0xC...)</td> </tr> </tbody> </table>	目的地址位宽度 DMACxControl<Dwidth[2:0]>	最低有效地址的设置	000: 字节(8 位)	无限制	001: 半字符(16 位)	设置为 2 的倍数, (0x0,0x02,0x4,0x06,0x8,0xA,0xC...)	010: 字符(32 位)	设置为 4 的倍数, (0x0,0x4,0x8,0xC...)
目的地址位宽度 DMACxControl<Dwidth[2:0]>	最低有效地址的设置										
000: 字节(8 位)	无限制										
001: 半字符(16 位)	设置为 2 的倍数, (0x0,0x02,0x4,0x06,0x8,0xA,0xC...)										
010: 字符(32 位)	设置为 4 的倍数, (0x0,0x4,0x8,0xC...)										

请勿在传输期间更新DMACxCnDestAddr。如需更改DMACxCnDestAddr，在更改前，请务必禁用通道 "n" (DMACxCnConfiguration<E>="0")。

8.4.15 DMACxLnLLI (DMAC 通道 x 链接表项目寄存器)

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

位	比特符号	类型	功能
31-2	LLI[29:0]	R/W	设置该传送信息的首个地址。 当<LLI> = 0时, LLI是最后一个链。DMA传输完成后, DMA通道禁用。
1-0	-	R/W	写作 0。

注: 有关<LLI>的详细操作, 见"8.5 特殊功能"。

8.4.16 DMACxCnControl (DMAC 通道 n 控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	I	-	-	-	DI	SI	-	-
复位后	0	未定义	未定义	未定义	0	0	未定义	未定义
	23	22	21	20	19	18	17	16
比特符号	Dwidth			Swidth			DBSize	
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	DBSize	SBSize			TransferSize			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TransferSize							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	I	R/W	启用传输中断寄存器。 0: 禁用 1: 启用 通过设置<I>="1"以及DMACxCnConfiguration<ITC>="1", 将生成传输端中断。当分散/聚集功能在最后一次传输DMAC设置流程中使用, 通过设置这个位进行启用, 传输末端中断生成仅在最后一次传输中是启用的。将该位设置为"1"并转换为使能模式, 即可在标准传送期间生成中断。
30-28	-	W	写作 0。
27	DI	R/W	增加传输目的地址 0: 无需增加 1: 增量
26	SI	R/W	增加传输源地址 0: 无需增加 1: 增量
25-24	-	W	写作 0。
23-21	Dwidth[2:0]	R/W	传输目的地位宽度。 000: 字节(8 位) 001: 半字符(16 位) 010: 字符(32 位) 其它: 保留 有关该设置值, 请参看表 8-3。
20-18	Swidth[2:0]	R/W	传输源位宽度 000: 字节(8 位) 001: 半字符(16 位) 010: 字符(32 位) 其它: 保留 有关该设置值, 请参看表 8-3。
17-15	DBSize[2:0]	R/W	传输目的地突发量: (注 1) 000: 1 拍 100: 32 位 001: 4 拍 101: 64 位 010: 8 拍 110: 128 位 011: 16 拍 111: 256 位 有关该设置值, 请参看表 8-3。

位	比特符号	类型	功能
14-12	SBSize[2:0]	R/W	<p>传输源突发量：(注 1)</p> <p>000：1 拍 100：32 位</p> <p>001：4 拍 101：64 位</p> <p>010：8 拍 110：128 位</p> <p>011：16 拍 111：256 位</p> <p>有关该设置值，请参看表 8-3。</p>
11-0	TransferSize [11:0]	R/W	<p>设置传输的总数。</p> <p>把由传输源位宽(4 个字节/2 个字节/1 个字节)指定的每个位宽的传输数据量设置为<传输大小[11:0]>。因为突发量会内部显示在每次DMA请求时的传输数据量，所以在传输源位宽和传输总次数不发生变化时，即使设置任何突发量，传输数据量绝不会发生变化。</p> <p>通过DMA传输，把<传输大小[11:0]>的值增加为 0。</p> <p>如果读取了该数值，则该值(即不会在DMA运行时传送的数据的数目)会在DMA停止时被读作零。</p> <p>传输总次数用作传输源位宽的单元。</p> <p>例如：</p> <p>当< Swidth >= "000" (8 位)时，传输数量以字节为单位表示。</p> <p>当< Swidth >= "001" (16 位)时，传输数量以半字符为单位表示。</p> <p>当< Swidth >= "010" (32 位)时，传输数量以字符为单位表示。</p>

注：拟以DBSize和SBSize设置的突发量与AHB总线HBURST无任何关系。

表 8-3 如何确定<Dwidth[2:0]>, <Swidth[2:0]>, <DBSize[2:0]>, <SBSize[2:0]>的值

<Dwidth[2:0]>/ <Swidth[2:0]>	<p>设置数字以满足下述表达：</p> <p>传输源位宽度x传输总量= 传输目标位宽度x N (N: 整数)</p> <p>(示例 1)传输源位宽：8 位，传输目的地位宽：32 位，传输总次数：25 次</p> <p>8 位 x 25 倍 = 200 位 (25 字节)</p> <p>$N = 200 \div 32 = 6.25$ 字符</p> <p>因为 6.25 不是一个整数，所以上述设置是无效的。</p> <p>如果传输源位宽小于传输目的地位宽，在设置传输总次数时必须小心。</p> <p>(示例 2)传输源位宽：32 位，传输目的地位宽：16 位，传输总次数：13 次</p> <p>32 位 x 13 倍 = 416 位(13 字符)</p> <p>$N = 416 \div 16 = 26$ 半字符</p> <p>因为 26 是整数，所以上述设置是有效的。</p>
<DBSize[2:0]>/ < SBSize[2:0]>	<p>在执行"外设至存储器"或"存储器至外设"时，外围电路可生成DMA请求信号，用以指示准备工作已就绪。触发此信号，以执行数据传输。(如是"存储器-存储器"的传输方式，仅使用软件启动。)</p> <p>设置突发量，以定义外设在每个DMA请求信号时传输的数据量。本寄存器可与可容纳多样数据的FIFO缓冲器配套使用。</p>

8.4.17 DMACxCnConfiguration (DMAC 通道 n 配置寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	Halt	Active	Lock
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	15	14	13	12	11	10	9	8
比特符号	ITC	IE	FlowCntrl			-	DestPeripheral	
复位后	0	0	0	0	0	未定义	0	0
	7	6	5	4	3	2	1	0
比特符号	DestPeripheral		-	SrcPeripheral			e.	
复位后	0	0	未定义	0	0	0	0	0

位	比特符号	类型	功能												
31-19	-	W	写作 0。												
18	Halt	R/W	控制装置接受DMA请求 0: 接受DMA请求 1: 忽略DMA请求												
17	Active	R	说明通道FIFO中是否存在数据。 0: FIFO中不存在数据 1: FIFO中存在数据												
16	Lock	R/W	设置锁定传输(不分开传输)。 0: 禁用被锁定传输 1: 启用锁定传输 在已锁定传输被使能时,会在不解除该总线的情况下连续执行尽可能多的脉冲串传送。有关详细操作,见“特殊功能”。												
15	ITC	R/W	传输末端中断使能寄存器。 0: 禁用中断 1: 启用中断												
14	IE	R/W	错误中断启用寄存器 0: 禁用中断 1: 启用中断												
13-11	FlowCntrl[2:0]	R/W	设置传输方法 <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 40%;"><FlowCntrl[2:0]> 设定值</th> <th style="width: 60%;">传输方法</th> </tr> </thead> <tbody> <tr> <td>000:</td> <td>存储器-存储器(注)</td> </tr> <tr> <td>001:</td> <td>存储器-外设</td> </tr> <tr> <td>010:</td> <td>外设-存储器</td> </tr> <tr> <td>011:</td> <td>外设-外设</td> </tr> <tr> <td>100 ~ 111:</td> <td>保留</td> </tr> </tbody> </table>	<FlowCntrl[2:0]> 设定值	传输方法	000:	存储器-存储器(注)	001:	存储器-外设	010:	外设-存储器	011:	外设-外设	100 ~ 111:	保留
<FlowCntrl[2:0]> 设定值	传输方法														
000:	存储器-存储器(注)														
001:	存储器-外设														
010:	外设-存储器														
011:	外设-外设														
100 ~ 111:	保留														
10	-	W	写作 0。												
9-6	DestPeripheral [3:0]	R/W	设置传输目的地外设(注 2) 当存储器是传输目的地时,此设定被忽略。												
5	-	W	写作 0。												
4-1	SrcPeripheral [3:0]	R/W	设置传输源外设(注 2) 当存储器是传输源时,此设定被忽略。												

位	比特符号	类型	功能
0	E	R/W	<p>通道启用</p> <p>0: 禁用</p> <p>1: 启用</p> <p>可以使用该位来启用/禁用通道。(在选用"存储器-存储器"传输方式时, 此位作为启动位。)</p> <p>在完成由DMACxCnControl<TransferSize>指定的传输数据量时, 对应的<E>自动清零。</p> <p>在传输期间禁用通道会损失FIFO中的数据。在重启前对所有通道进行初始化。</p> <p>如需暂停传, 可用<HALT>停止该DMA请求, 在该<Active>变为"0"之前查询该数据, 然后该<E>位禁用该通道。</p>

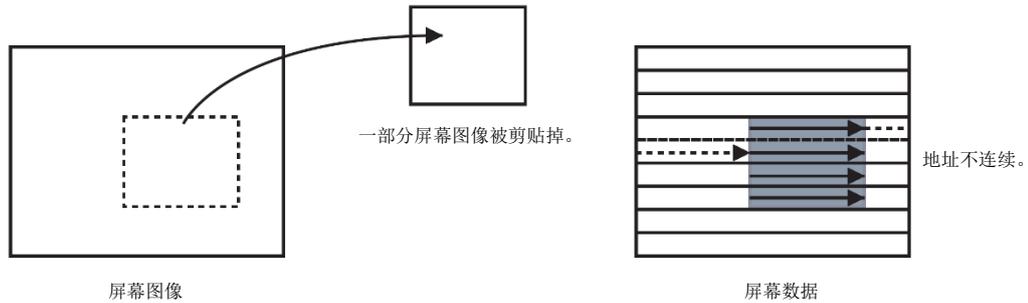
注 1: 当选定 "存储器-存储器"时, 将不支持用硬件启动来进行DMA启动。写入"1"<E>, 开始传输。

注 2: 在DMACxENableChns<EnabledCHx>被使能, 且相应的DMACxCnConfiguration<Halt>被设置为"1"时, 在通道使能位 (E:位0)被清除为"0"之后将其写入。在没有这样做的情况下, 如在写入它们时发生从站错误, 则错误会通过复位恢复。关于从站错误, 当传输宽度和地址失配时, 会发生这种错误。

8.5 特殊功能

8.5.1 分散/聚集功能

在删除部分图像数据并对其进行传输时，不得把图像数据作为连续数据处理，而且地址会根据特殊规则发生突然变化。因为DMA只能通过使用连续地址传输数据，所以必需在地址发生变化的位置按要求进行设定。



分散/聚集功能每次可通过重新载入DMA设置的方式，对它们进行连续操作(传输源地址，目的地址，传输次数和传输总线宽度)，指定的DMA执行次数已通过预设的"链表"(此处的CPU不需要控制操作)完成。在DMACxCnLLI 寄存器中设置为"1"来启用/禁用操作。

可以链表设置的项目要配置下列 4 个字符：

1. DMACxCnSrcAddr
2. DMACxCnDestAddr
3. DMACxCnLLI
4. DMACxCnControl

它们可与中断操作一起使用。

中断取决于DMACxCnn控寄存器的计数末端中断启用位，可在每个LLI结束时产生。在使用此位时，可使用LLI增加条件，即使是在传输期间，以执行分支操作等。为清除中断，要控制DMACxIntTCClear寄存器的适当位。

8.5.2 链接表操作

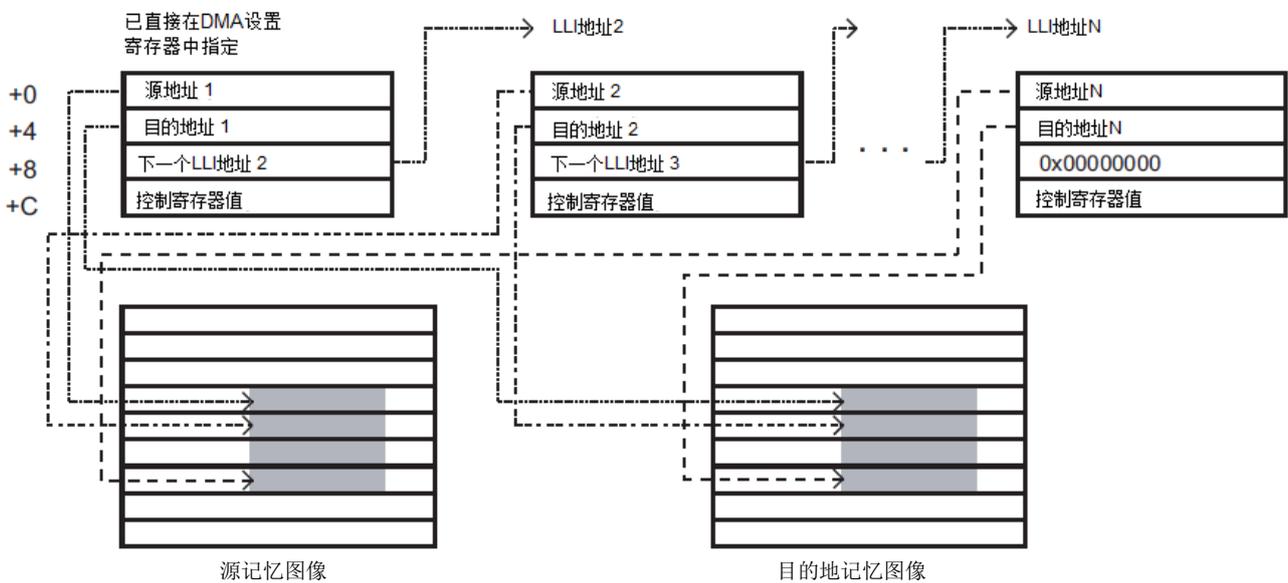
位操作分散/聚集功能，传输源和源数据区域需要首先通过创建一组链表进行确定。

每个设置称作LLI(链表)。

每个LLI控制一个数据块的传输。每个LLI表明正常DMA设置和连续数据的控制传输。每当每个DMA传输完成时，会加载下一个LLI设置，以继续DMA操作(菊花链)。

设置示例如下所示。

1. 应直接在DMA寄存器中进行首个DMA传送的设置。
2. 应将第二个及其后的DMA传送设置写入“下一LLI的地址X”中的存储器设置中。
3. 如需停止直至第N次DMA传送，则可将“下一LLI地址X”设置为 0x0000_0000。

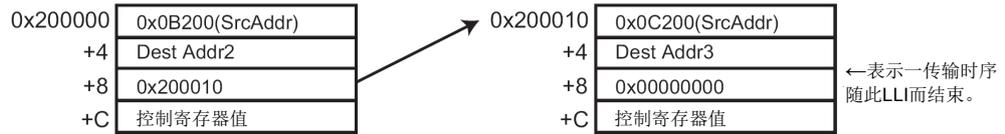


在由方形封闭的区域内传输数据时

	0x002000	0x00E000
0x0A000		
0x0B000		
0x0C000		

设置寄存器	设置参数
+0 DMACxCnSrcAddr	:0x0A200
+4 DMACxCnDestAddr	:目的地址 1
+8 DMACxCnLLI	:0x200000
+C DMACxCnControl	:设置脉冲串传输的次数和传输的次数等。

链接表



9. 输入/输出端口

本节主要对与端口相关的寄存器、其设置和电路进行说明。

9.1 寄存器

在使用这些端口寄存器时，必须设置以下寄存器。

所有寄存器均为 32 位。各配置会随着端口位个数和功能指定的不同而有所不同。

在以下描述中，"x"表示的端口的名称，"n"则表示功能编号。

寄存器名称		设置值	
PxDATA	数据寄存器	0 或 1	该寄存器可读取/写入端口数据。
PxCR	输出控制寄存器	0: 输出禁用 1: 输出启用	寄存器可控制输出。
PxFRn	功能寄存器 n	0: PORT 1: 功能	该寄存器可设置该功能。 通过设置"1"，可使能该指定功能。 该寄存器存在时，可用于被指定给该端口的各功能。如具备若干功能，则仅一种功能可被使能。
PxOD	漏极开路控制寄存器	0: CMOS 1: 开漏	该寄存器可控制可编程漏极输出。可用PxOD设置各可编程漏极输出。在时输出数据为"1"时，输出缓冲器即被禁用，并变为一个虚拟的漏极输出。
PxPUP	上拉控制寄存器	0: 上拉禁用 1: 上拉启用	该寄存器可控制可编程上拉。
PxPDN	下拉控制寄存器	0: 下拉禁用 1: 下拉启用	该寄存器可控制可编程下拉。
PxIE	输入控制寄存器	0: 输入禁用 1: 输入启用	该寄存器可控制输入。 在使能PxIE后、外部数据在PxDATA中被反射之前，必须等待一段时间。

9.1.1 寄存器列表

有关基地址的详细情况，参照“内存印象图”一章的“外部功能地址表”。

无功能的位被读作“0”，且将数据写入该位无效。

寄存器名称	地址 (基+)	端口A	端口B	端口C	端口D	端口E
数据寄存器	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA
输出控制寄存器	0x0004	PACR	PBCR	PCCR	PDCR	PECR
功能寄存器 1	0x0008	-	PBFR1	PCFR1	PDFR1	PEFR1
功能寄存器 2	0x000C	-	PBFR2	-	-	-
漏极开路控制寄存器	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD
上拉控制寄存器	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP
下拉控制寄存器	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN
输入控制寄存器	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE

寄存器名称	地址(基+)	端口F	端口G
数据寄存器	0x0000	PFDATA	PGDATA
输出控制寄存器	0x0004	PFCR	PGCR
功能寄存器 1	0x0008	PFFR1	PGFR1
功能寄存器 2	0x000C	-	-
漏极开路控制寄存器	0x0028	PFOD	PGOD
上拉控制寄存器	0x002C	PFPUP	PGPUP
下拉控制寄存器	0x0030	PFPDN	PGPDN
输入控制寄存器	0x0038	PFIE	PGIE

注：在该地址被显示为“-”时，其不可访问。

9.1.2 端口功能和设置列表

各端口的功能和设置寄存器列表给出如下。

- "表 9-1 端口A设置列表"
- "表 9-2 端口B设置列表"
- "表 9-3 端口C设置列表"
- "表 9-4 端口D设置列表"
- "表 9-5 端口E设置列表"
- "表 9-6 端口F设置列表"
- "表 9-7 端口G设置列表"

在单元格PxFRn中，给出了在选择某功能时必须设置的功能寄存器。如果该寄存器被设置为"1"，则相应的功能即被PxFRn使能。

(在以下描述中，"x"表示各端口的名称，而"n"则表示功能编号)

位于充满阴影线的单元格内的位被读作"0"，且将数据写入该位无效。

该表中所给出的"0"或"1"表示已被设置到该寄存器的值。"0/1"则表示可将该可选值设置到该寄存器。

9.1.2.1 PORT A

表 9-1 端口A设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PADATA	PAGR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN0	输入	FT5	0/1	0		0/1	0	0	0
PA1	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN1	输入	FT5	0/1	0		0/1	0	0	0
PA2	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN2	输入	FT5	0/1	0		0/1	0	0	0
PA3	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN3	输入	FT5	0/1	0		0/1	0	0	0
PA4	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN4	输入	FT5	0/1	0		0/1	0	0	0
PA5	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN5	输入	FT5	0/1	0		0/1	0	0	0
PA6	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN6	输入	FT5	0/1	0		0/1	0	0	0
PA7	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AIN7	输入	FT5	0/1	0		0/1	0	0	0

9.1.2.2 PORT B

表 9-2 端口B设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0 (注)	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
PB1	复位后			0	0	PBFR2	0	0	1	1
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC4RXD	输入	FT1	0/1	0	PBFR1	0/1	0/1	0/1	1
	SWCLK	输入	FT2	0/1	0	PBFR2	0	0	1	1
PB2	复位后			0	1	PBFR2	0	1	0	1
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC4RXD	输出	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
	SWDIO	I/O	FT2	0/1	1	PBFR2	0	1	0	1
PB3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC4SCLK	输入	FT1	0/1	0	PBFR1	0/1	0/1	0/1	0/1
输出		0/1		1	PBFR1	0/1	0/1	0/1	0/1	1
PB4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
PB5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	INT0	输入	FT4	0/1	0		0/1	0/1	0/1	1
PB6	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	INT1	输入	FT4	0/1	0		0/1	0/1	0/1	1
PB7	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	INT2	输入	FT4	0/1	0		0/1	0/1	0/1	1

注：PB0 工作时具备BOOT功能。在 **RESET** 引脚为“低”时，其可被输入及上拉。在该复位信号的上升沿，如果PB0 为“高”，则该设备进入单片模式，并从片装闪存存储器引导。如果PB0 为“低”，则该设备进入单引导模式，并从内部BOOT程序引导。

9.1.2.3 PORT C

表 9-3 端口C设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE
PC0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	I2C0SCL	I/O	FT1	0/1	1	PCFR1	1	0/1	0/1	1
PC1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	I2C0SDA	I/O	FT1	0/1	1	PCFR1	1	0/1	0/1	1
PC2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB2OUT	输出	FT1	0/1	1	PCFR1	0/1	0/1	0/1	0
PC3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB0OUT	输出	FT1	0/1	1	PCFR1	0/1	0/1	0/1	0
PC4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	T16A0OUT	输出	FT1	0/1	1	PCFR1	0/1	0/1	0/1	0
PC5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB0IN	输入	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1

9.1.2.4 PORT D

表 9-4 端口D设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE
PD0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB3OUT	输出	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
PD1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC0SCK	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
输出		0/1		1	PDFR1	0/1	0/1	0/1	0	
PD2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC0RXD	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
PD3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC0TXD	输出	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
PD4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB3IN	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
PD5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0

9.1.2.5 PORT E

表 9-5 端口E设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
PE1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
PE2	复位后			0	1	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC2SCK	输入	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
输出		0/1		1	PEFR1	0/1	0/1	0/1	0	
PE3	复位后			0	1	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC2RXD	输入	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC2TXD	输出	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	INT5	输入	FT4	0/1	0		0/1	0/1	0/1	1
PE6	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	INT4	输入	FT4	0/1	0		0/1	0/1	0/1	1
PE7	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	INT3	输入	FT4	0/1	0		0/1	0/1	0/1	1

9.1.2.6 PORT F

表 9-6 端口F设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB7IN	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
PF1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC3SCK	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
输出		0/1		1	PFFR1	0/1	0/1	0/1	0	
PF2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC3RXD	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
PF3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC3TXD	输出	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB7OUT	输出	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	T16A1OUT	输出	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF6	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
PF7	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0

9.1.2.7 PORT G

表 9-7 端口G设置列表

PO RT	复位状态	输入/输出	PORT 类型	控制寄存器						
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PG0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC1SCK	输入	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
输出		0/1		1	PGFR1	0/1	0/1	0/1	0	
PG1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC1RXD	输入	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
PG2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	SC1TXD	输出	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0
PG3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB1IN	输入	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
PG4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB1OUT	输出	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0
PG5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB4OUT	输出	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0
PG6	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB5OUT	输出	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0
PG7	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	TB6OUT	输出	FT1	0/1	1	PGFR1	0/1	0/1	0/1	0

9.1.3 端口的方块图

9.1.3.1 端口类型

端口分类如下。有关各端口类型所对应的框图，请参见后续几页。图中虚线框表示"端口框图"所示等效电路部分。

表 9-8 功能列表

类型	功能	上拉	下拉	模拟
	输入/输出			
FT1	输入/输出	R	R	-
FT4	输入(INT)	R	R	-
FT5	输入(AIN)	R	R	o
FT6	输入($\overline{\text{BOOT}}$)	EnR	R	-

int: 中断输入

-: 不存在

o: 存在

R: 在复位时强制禁用

EnR: 在复位时强制使能

9.1.3.2 类型 FT1

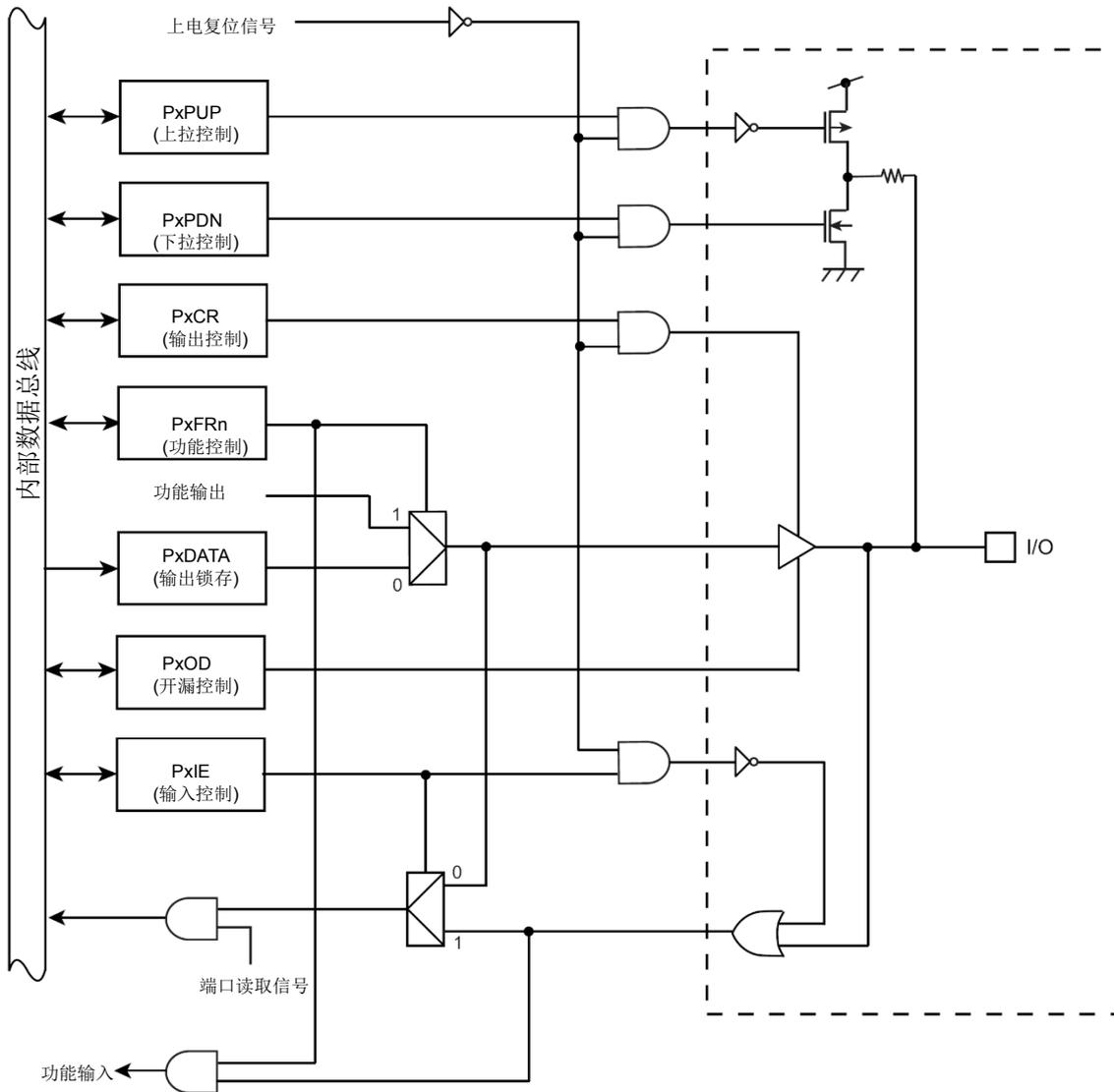


图 9-1 端口类型 FT1

9.1.3.3 类型 FT4

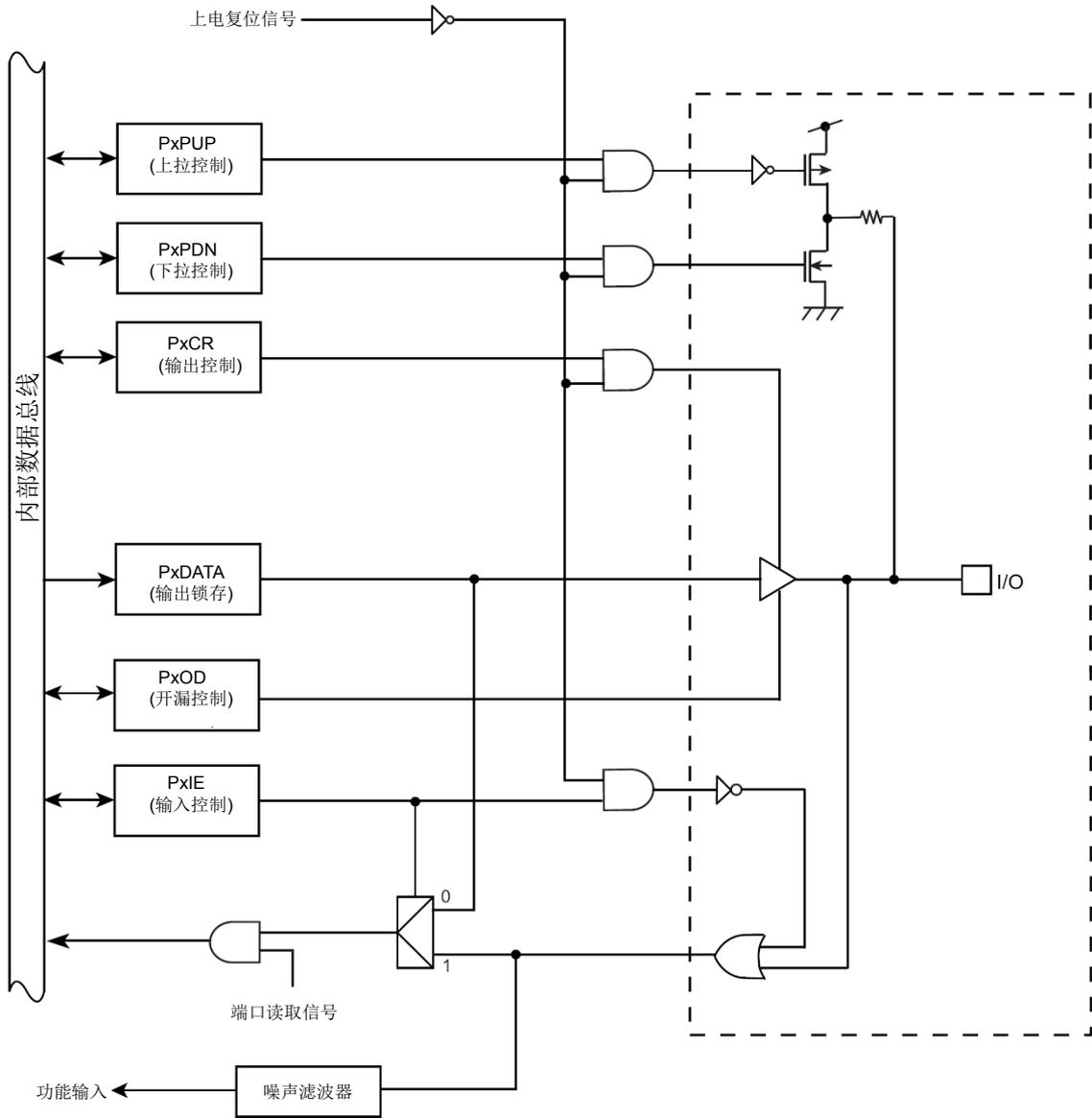


图 9-2 端口类型 FT4

9.1.3.4 类型 FT5

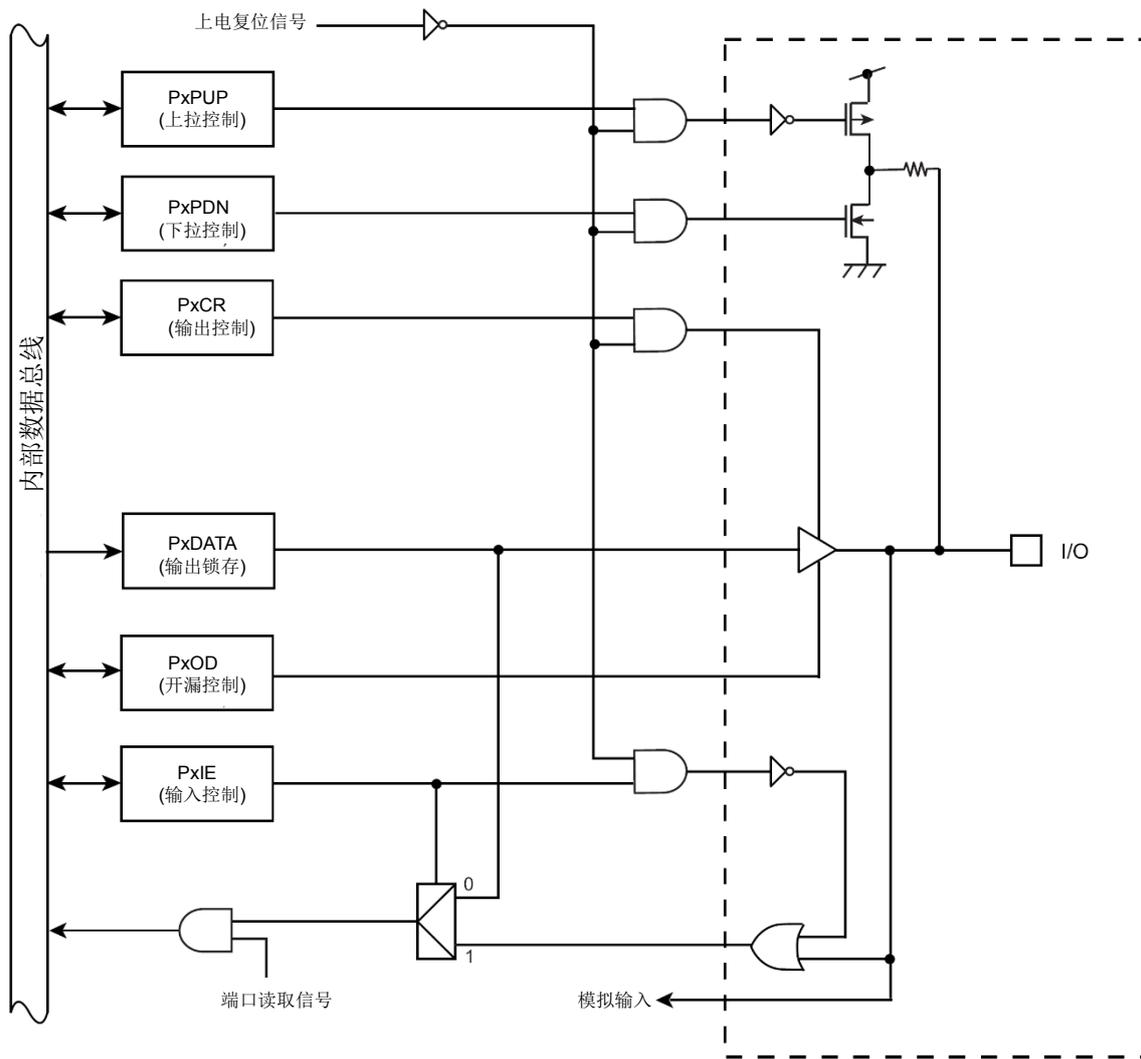


图 9-3 端口类型 FT5

9.1.3.5 类型 FT6

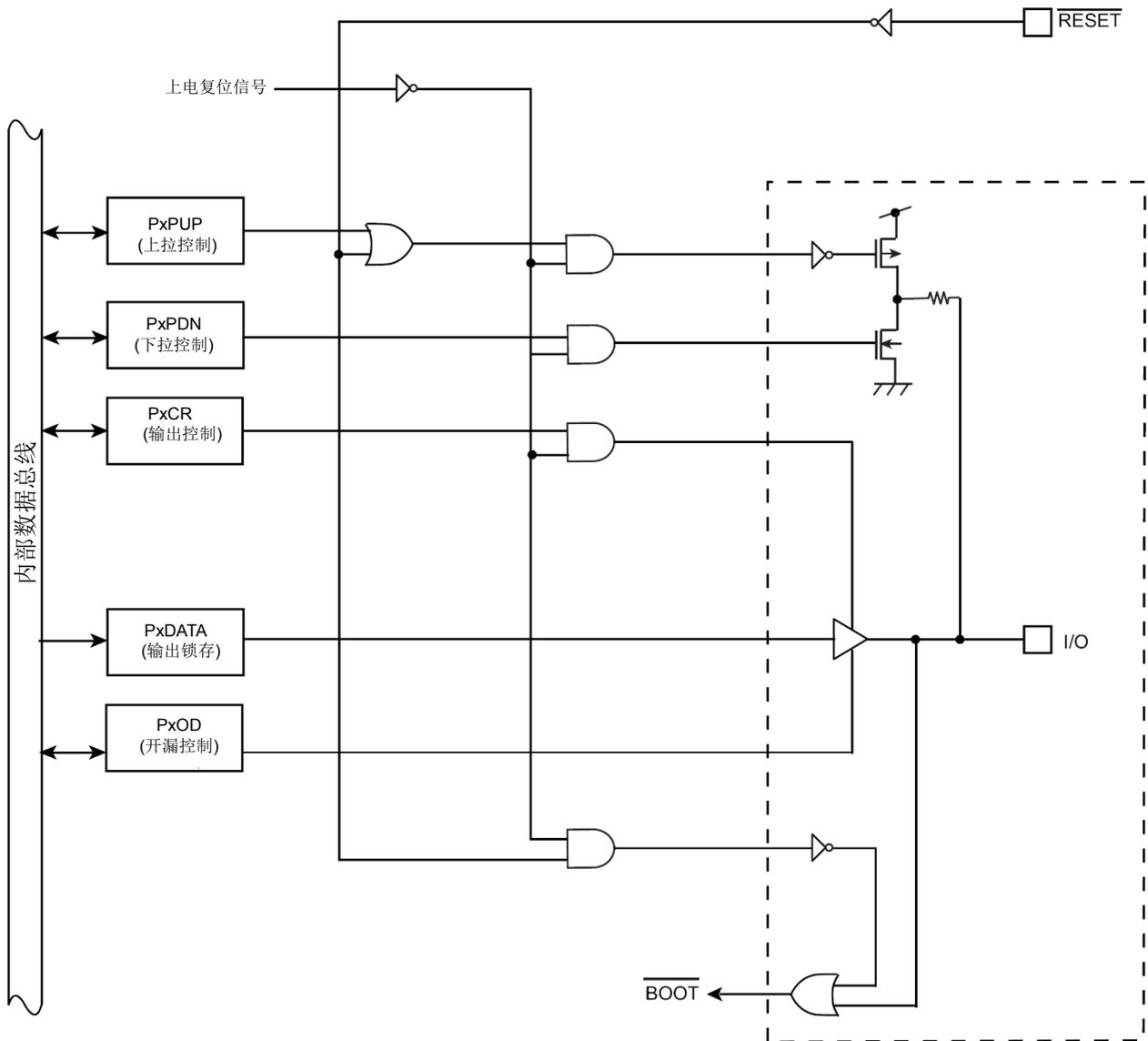


图 9-4 端口类型 FT6

10.16 位计时器/事件计数器(TMRB)

10.1 概要

TMRBx 具有下列工作模式。

- 间隔计时器模式
- 事件计数器模式
- 可编程脉冲发生(PPG)模式
- 可编程脉冲发生(PPG)外部触发器模式

使用捕获功能使 TMRBx 能执行下列测量。

- 频率测量
- 脉冲宽度测量

10.2 方块图

TMRBx 由 1 个 16 位递增计数器，2 个 16 位计时器寄存器(双缓冲)，2 个 16 位捕获寄存器，2 个比较器，捕获输入控制，1 个计时器触发器及其相关控制电路组成。计时器工作模式和计时器触发器受寄存器控制。

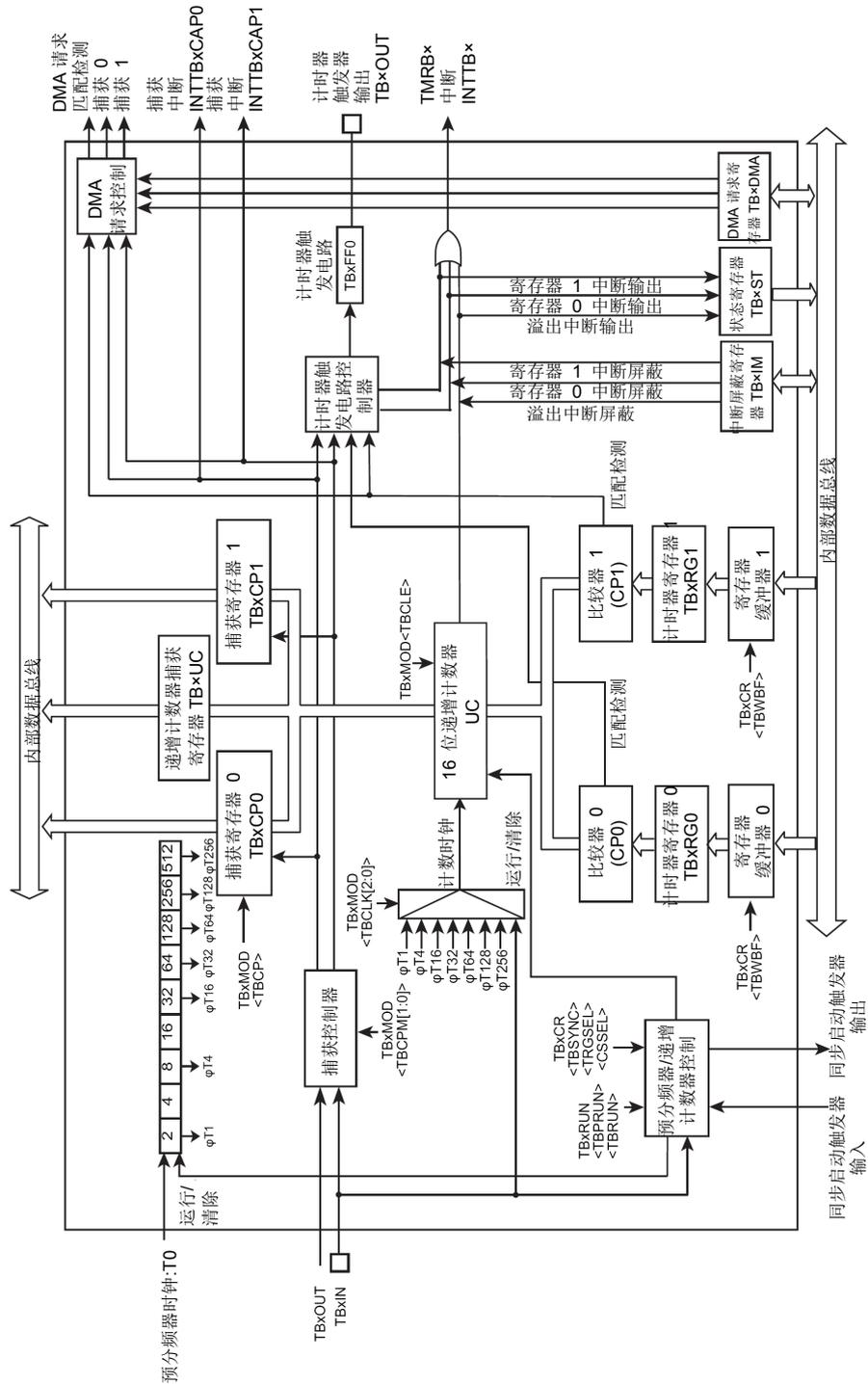


图 10-1 TMRBx 方块图

10.3 寄存器

10.3.1 寄存器列表

下表显示控制寄存器及其地址。

基址详见"内存地址"章节"外设功能地址列表"。

寄存器名称		地址(基+)
启用寄存器	TBxEN	0x0000
RUN 寄存器	TBxRUN	0x0004
控制寄存器	TBxCR	0x0008
模式寄存器	TBxMOD	0x000C
触发电路控制寄存器	TBxFFCR	0x0010
状态寄存器	TBxST	0x0014
中断屏蔽寄存器	TBxIM	0x0018
递增计数器捕获寄存器	TBxUC	0x001C
计时器寄存器 0	TBxRG0	0x0020
计时器寄存器 1	TBxRG1	0x0024
捕获寄存器 0	TBxCP0	0x0028
捕获寄存器 1	TBxCP1	0x002C
DMA 请求启用寄存器	TBxDMA	0x0030

注：在计时器工作时，不要修改计时器控制寄存器、计时器模式寄存器和计时器触发器控制寄存器。用户可在计时器停止工作后修改它们。

10.3.2 TBxEN(启用寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TBEN	R/W	<p>TMRBx 操作</p> <p>0: 禁用</p> <p>1: 启用</p> <p>规定 TMRBx 的操作。当操作禁用时, 不向 TMRBx 中的其他寄存器提供时钟。可降低功耗。(除 TBxEN 寄存器外, 禁用读取和写入其它寄存器。)</p> <p>为了使用 TMRB, 应在 TMRBx 中各寄存器编程前使能 TMRBx(设为"1")。当执行 TMRBx 操作后被禁用时, 应将设置保持在各寄存器中。</p>
6	TBHALT	R/W	<p>在调试停止期间时钟运行</p> <p>0: 运行</p> <p>1: 停止</p> <p>指定在使用中, 当调试工具转到 HALT 模式时, TMRBx 时钟的运行或停止设置。</p>
5-0	-	R	读作"0"。

10.3.3 TBxRUN(RUN 寄存器)

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

位	比特符号	类型	功能
31-3	-	R	读作"0"。
2	TBPRUN	RW	预分频器操作 0: 停止&清除 1: 计数
1	-	R	读作"0"。
0	TBRUN	RW	计数操作 0: 停止&清除 1: 计数

10.3.4 TBxCR(控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TBWBFB	-	TBSYNC	-	I2TB	-	TRGSEL	CSSEL
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TBWBFB	R/W	双缓冲器 0: 禁用 1: 启用
6	-	R/W	写入"0"。
5	TBSYNC	R/W	同步模式切换 0: 个别(各通道) 1: 同步
4	-	R	读作"0"。
3	I2TB	R/W	在 IDLE 模式下运行 0: 停止 1: 运行
2	-	R/W	写入"0"。
1	TRGSEL	R/W	当使用外部触发器时, 选择边沿。 0: 上升 1: 下降 当外部触发器(TBxIN)启动计数时, 选择边沿。
0	CSSEL	R/W	选择计数起动 0: 由软件起动 1: 由外部触发器起动

10.3.5 TBxMOD(模式寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	TBCP	TBCPM		TBCLE	TBCLK		
复位后	0	1	0	0	0	0	0	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	-	R/W	写入"0"。
6	TBCP	W	通过软件捕获控制 0: 通过软件捕获 1: 忽略 写入"0"时, 捕获寄存器 0 (TBxCP0)捕获计数值。读作"1"。
5-4	TBCPM[1:0]	R/W	捕获计时 00: 禁用 01: 保留 10: TBxIN \uparrow TBxIN \downarrow 在捕获寄存器 0 (TBxCP0)的 TBxIN 输入的上升沿, 捕获计数值。 在捕获寄存器 1 (TBxCP1)的 TBxIN 输入的下降沿, 捕获计数值。 11: TBxFF0 \uparrow TBxFF0 \downarrow 在捕获寄存器 0 (TBxCP0)的 TBxFF0 输入的上升沿捕获计数值。 在捕获寄存器 1 (TBxCP1)的 TBxFF0 输入的下降沿捕获计数值。
3	TBCLE	R/W	递增计数器控制 0: 禁用递增计数器的清除 1: 启用递增计数器的清除。 清除并控制该递增计数器。 当写入"0"时升计数器清零功能被禁用。写入"1"时, 它在递增计数器与计时器寄存器 1 (TBxRG1)匹配时清除递增计数器。
2-0	TBCLK[2:0]	R/W	选择该 TMRBx 源时钟。 000: TBxIN 引脚输入 001: ϕ T1 010: ϕ T4 011: ϕ T16 100: ϕ T32 101: ϕ T64 110: ϕ T128 111: ϕ T256

注: 当 TMRBx 正运行时, 不要改变 TBxMOD 寄存器。

10.3.6 TBxFFCR(触发器控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	TBC1T1	TBC0T1	TBE1T1	TBE0T1	TBFF0C	
复位后	1	1	0	0	0	0	1	1

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7-6	-	R	读作"1"。
5	TBC1T1	R/W	递增计数器值纳入 TBxCP1 时的 TBxFF0 反向触发器。 0: 禁用触发器 1: 启用触发器 通过设置"1", 当递增计数器值纳入捕获寄存器 1 (TBxCP1)时, 计时器触发器反转。
4	TBC0T1	R/W	递增计数器值纳入 TBxCP0 时的 TBxFF0 反向触发器。 0: 禁用触发器 1: 启用触发器 通过设置"1", 当递增计数器值纳入捕获寄存器 0 (TBxCP0)时, 计时器触发器反转。
3	TBE1T1	R/W	递增计数器值与 TBxRG1 匹配时的 TBxFF0 反向触发器。 0: 禁用触发器 1: 启用触发器 通过设置"1", 当递增计数器值与计时器寄存器 1 (TBxRG1)匹配时, 计时器触发器反转。
2	TBE0T1	R/W	递增计数器值与 TBxRG0 匹配时的 TBxFF0 反向触发器。 0: 禁用触发器 1: 启用触发器 通过设置"1", 当递增计数器值与计时器寄存器 0 (TBxRG0)匹配时, 计时器触发器反转。
1-0	TBFF0C[1:0]	R/W	TBxFF0 控制 00: 反转 使 TBxFF0 值反转(用软件反转)。 01: 设置 将 TBxFF0 设为"1"。 10: 清除 将 TBxFF0 清"0"。 11: 忽略 始终读作"11"。

10.3.7 TBxST(状态寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	INTTBOF	INTTB1	INTTB0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-3	-	R	读作"0"。
2	INTTBOF	R	溢出中断请求标志 0: 未发生溢出 1: 发生溢出 在某递增计数器溢出时, 即设置"1"。
1	INTTB1	R	匹配(TBxRG1)中断请求标志 0: 未检测到匹配 1: 检测到与 TBxRG1 匹配 当检测到与计时器寄存器 1 (TBxRG1)匹配时, 设置"1"。
0	INTTB0	R	匹配(TBxRG0)中断请求标志 0: 未检测到匹配 1: 检测到与 TBxRG0 匹配 当检测到与计时器寄存器 0 (TBxRG0)匹配时, 设置"1"。

注 1: 即使 TBxIM 寄存器的屏蔽配置有效, 状态仍设为 TBxST 寄存器。

注 2: 当中断屏蔽配置由 TBxIM 寄存器的对应位禁用时, 中断被发给 CPU。

注 3: 为了清除标志, 读取 TBxST 寄存器。

10.3.8 TBxIM(中断屏蔽寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	TBIMOF	TBIM1	TBIM0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-3	-	R	读作"0"。
2	TBIMOF	R/W	溢出中断请求屏蔽 0: 禁用 1: 启用 将该递增计数器溢出中断设置为禁用或启用。
1	TBIM1	R/W	匹配(TBxRG1)中断请求屏蔽 0: 禁用 1: 启用 将与计时器寄存器 1 (TBxRG1)的匹配中断请求屏蔽设为启用或禁用。
0	TBIM0	R/W	匹配(TBxRG0)中断请求屏蔽 0: 禁用 1: 启用 将与计时器寄存器 0 (TBxRG0)的匹配中断请求屏蔽设为启用或禁用。

注：即使 TBxIM 寄存器的屏蔽配置有效，状态仍设为 TBxST 寄存器。

10.3.9 TBxUC(上升计数器捕获寄存器)

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

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	TBUC[15:0]	R	通过读取递增计数器输出，捕获某个值。 若在计数器工作时读取 TBxUC，就会捕获递增计数器当前值。

10.3.10 TBxRG0(计时器寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TBRG0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TBRG0							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	TBRG0[15:0]	R/W	将某比较值设置到该递增计数器。

10.3.11 TBxRG1(计时器寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TBRG1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TBRG1							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	TBRG1[15:0]	R/W	将某比较值设置到该递增计数器。

10.3.12 TBxCP0(捕获寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TBCP0							
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	7	6	5	4	3	2	1	0
比特符号	TBCP0							
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	TBCP0[15:0]	R	已读取从该递增计数器捕获到的某个值。

10.3.13 TBxCP1(捕获寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TBCP1							
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	7	6	5	4	3	2	1	0
比特符号	TBCP1							
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	TBCP1[15:0]	R	已读取从该递增计数器捕获到的某个值。

10.3.14 TBxDMA(DMA 请求启用寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	TBDMAEN2	TBDMAEN1	TBDMAEN0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-3	-	R	读作"0"。
2	TBDMAEN2	R/W	选择 DMA 请求。比较与 TBxRG1 的匹配。(注 1) 0: 禁用 1: 启用
1	TBDMAEN1	R/W	选择 DMA 请求。输入捕获 1 0: 禁用 1: 启用
0	TBDMAEN0	R/W	选择 DMA 请求。输入捕获 0 0: 禁用 1: 启用

注 1: 当 TBxIM<TBIM1>寄存器的掩码配置有效, 即使用<TBDMAEN2>启用 DMA 请求, DMA 请求仍不会发生。

注 2: DMA 请求因素的分配随通道不同。请参考"产品信息"章节。

10.4 操作描述

10.4.1 预分频器

TMRB_x 具有 4 位预分频器，以生成上升计数器的源时钟。

预分频器输入时钟 $\phi T0$ 为 CG 电路中 CGSYSCR<PRCK[2:0]>所选的 fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 或 fperiph/32。外设时钟为 fgear(CG 电路中 CGSYSCR<FPSEL>所选时钟)或 fc(时钟齿轮分频前的时钟)。

预分频器工作或停止由 TB_xRUN<TBPRUN>设定。将"1"写入位元，开始计数；将"0"写入位元，停止计数。

10.4.2 上升计数器(UC)

UC 为 16 位二进制计数器。

10.4.2.1 源时钟

上升计数器的源时钟由 TB_xMOD<TBCLK[2:0]>指定。

可从预分频器输出时钟- $\phi T1, \phi T4, \phi T16, \phi T32, \phi T64, \phi T128, \phi T256$ -或 TB_xIN 引脚外部时钟中选择。

10.4.2.2 计数器启动/停止

可通过软件启动、外部触发器启动和同步启动来启动计数器。

1. 软件启动

若<TBRUN>设为"1"，计数器启动。若<TBRUN>设为"0"，计数器停止，同时上升计数器被清除。

2. 外部触发器启动

在外部触发器模式下，计数器通过外部信号启动。

若 TB_xCR<CSSEL>设为"1"，则设为外部触发器模式。同时，若<TBRUN>设为"1"，计数器的状况为触发等待。计数器在 TB_xIN 的上升/下降沿启动。

TB_xCR<TRGSEL>位指定外部触发器边沿的切换。

<TRGSEL> = "0": 选择 TB_xIN 的上升沿。

<TRGSEL> = "1": 选择 TB_xIN 的下降沿。

若<TBRUN>设为"0"，计数器停止，同时上升计数器被清除。

3. 同步启动

在计时器同步模式下，同步启动计时器是可能的。若在 PPG 输出模式下使用计时器同步模式，就能实现电机驱动应用。

视产品而定，已确定主机通道和从机通道的组合。本产品主机通道和从机通道的组合见"产品信息"章节。

TBxCR<TBSYNC>位指定同步模式的切换。若从机通道的<TBSYNC>位设为"1", 计数器将随主机通道的软件或外部触发器启动而同步启动/停止。不必设置从机通道的TBxRUN<TBPRUN, TBRUN>位。主机通道的<TBSYNC>位必须设为"0"。

注意若同时设置外部触发器计数器模式和计时器同步模式, 则计时器同步模式优先。

10.4.2.3 计数器清除

上升计数器在以下时间被清除:

1. 当检测到与 TBxRG1 匹配时

通过设置 TBxMOD<TBCLE> = "1", 若比较器检测到 UC 和 TBxRG1 匹配, 则上升计数器被清除。

2. 当上升计数器停止时

当 TBxRUN<TBRUN> = "0"时, 上升计数器停止并被清除。

10.4.2.4 递增计数器溢出

若上升计数器溢出, 则产生 INTTBx 溢出中断。

10.4.3 计时器寄存器(TBxRG0, TBxRG1)

TBxRG0 和 TBxRG1 是与上升计数器值比较的寄存器。这两个寄存器被嵌入各通道。若比较器检测到计时器寄存器中的设置值和上升计数器中的值匹配, 则比较器输出匹配检测信号。

TBxRG0 和 TBxRG1 是与寄存器缓冲器配对的双缓冲配置。在初始状态时, 该双缓冲被禁用。

双缓冲的禁用或启用由 TBxCR<TBWBF>指定。若<TBWBF> = 0, 双缓冲被禁用; 若<TBWBF> = "1", 则启用。

当双缓冲启用并且上升计数器与 TBxRG1 匹配时, 数据从寄存器缓冲器发送到计时器寄存器 (TBxRG0/1)。

当上升计数器停止时, 即使双缓冲启用, 双缓冲仍作为单缓冲器工作, 数据能直接写入 TBxRG0 和 TBxRG1。

10.4.4 捕获控制

这是一个对锁存值从 UC 进入 TBxCP0 和 TBxCP1 的计时进行控制的电路。上升计数器的捕获计时由 TBxMOD<TBCPM[1:0]>指定。

软件也能将上升计数器值捕入捕获寄存器。每当"0"写入 TBxMOD<TBCP>时, 上升计数器值就会被捕入 TBxCP0。

10.4.5 捕获寄存器(TBxCP0, TBxCP1)

这些寄存器捕获上升计数器值。

10.4.6 上升计数器捕获寄存器(TBxUC)

若在计数器工作时读取 TBxUC 寄存器，就会捕获上升计数器当前值，并读值。当计数器正停止时，保持最后捕获的值。

10.4.7 比较器(CP0, CP1)

这些电路比较上升计数器和 TBxRG0/1 设置值，以检测是否匹配。若检测到匹配，则发生 INTTBx。

10.4.8 计时器触发器(TBxFF0)

计时器触发器(TBxFF0)由来自比较器的匹配信号和至捕获寄存器的锁存信号反转。设置 TBxFFCR<TBC1T1, TBC0T1, TBC1T1, TBC1T0>，即可启用或禁用反转。

在复位后，TBxFF0 值变成未定义状态。将"00"写入 TBxFFCR<TBFF0C[1:0]>，即可反转触发器。写入"01"即可将其设置为"1"，通过写入"10"即可清除该设置。

TBxFF0 值能输出到计时器输出引脚(TBxOUT)。若进行计时器输出，应事先对相应的端口设置进行编程。

10.4.9 捕获中断(INTTBxCA0, INTTBxCAP1)

在锁存值从上升计数器进入 TBxCP0 和 TBxCP1 时，就能产生 INTTBxCA0 和 INTTBxCAP1。

10.4.10 DMA 请求

在匹配中断或捕获中断产生时，DMA 请求被发给 DMAC。当进行 DMA 传输时，用 TBxDMA 寄存器的对应位设置要启用的 DMA 请求。

注：若已用 TBxIM<TBIM1>设置中断屏蔽配置，则即使用 TBxDMAREQ<TBDMAEN2>启用 DMA 请求，DMA 请求仍不会发生。

10.5 各模式操作描述

10.5.1 间隔计时器模式

当以恒定间隔产生中断时，应设置计时器寄存器(TBxRG1)的间隔时间，以产生 INTTBx 中断。

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx。
TBxRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	将对应位设为“1”，即可允许产生 INTTBx 中断。
TBxFFCR	←	X	X	0	0	0	0	1	1	将 TBQ6FF0 反向触发器设为禁用
TBxMOD	←	X	1	0	0	0	*	*	*	改为将预定标器输出时钟用作输入时钟。禁用俘获功能。
							(***) = 001 ~ 111)			
TBxRG1	←	*	*	*	*	*	*	*	*	指定时间间隔。(16 位)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	X	X	X	X	X	1	X	1	启动预分频器和计数器。

注: X: 忽略, *: 可选值, -: 不要改变

10.5.2 事件计数器模式

可以通过将源时钟用作外部时钟(TBxIN 引脚输入)而使 TMRBx 成为事件计数器。

在 TBxIN 引脚输入的上升沿，上升计数器递增计数。上升计数器值能用软件捕获。可以通过读取捕获值来读取计数值。

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx。
TBxRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。
										将对应端口分配给 TBxIN。
TBxFFCR	←	X	X	0	0	0	0	1	1	将 TBxFF0 反向触发器设为禁用。
TBxMOD	←	X	1	0	0	0	0	0	0	设为源时钟，作为 TBxIN 引脚输入。
TBxRUN	←	X	X	X	X	X	1	X	1	启动预分频器和计数器。
TBxMOD	←	X	0	-	-	-	-	-	-	软件捕获已完成。

注: X: 忽略, *: 可选值, -: 不要改变

10.5.3 可编程脉冲发生(PPG)输出模式

方波能以任何频率和占空输出。输出脉冲可以是低有源的或高有源的。

当上升计数器与 TBxRG0 和 TBxRG1 设置值匹配时, TBxFF0 反转。TBxFF0 能从 TBxOUT 引脚输出。

注意 TBxRG0 和 TBxRG1 必须满足下列要求。

TBxRG0 设置值 < TBxRG1 设置值

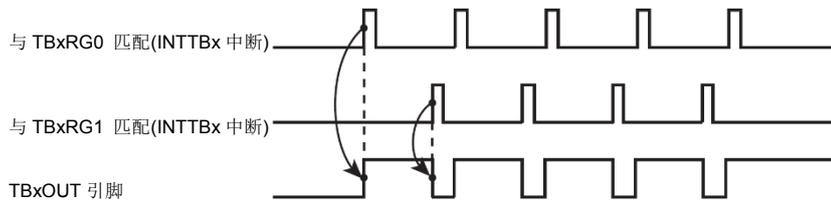


图 10-2 可编程脉冲发生输出示例

在该模式下,启用双缓冲,当 UC 与 TBxRG1 值匹配时,寄存器缓冲器 0 和 1 的值就会被移入 TBxRG0 和 TBxRG1。

这使得在不担心 TBxRG0 和 TBxRG1 计时变更的情况下修改频率和占空。

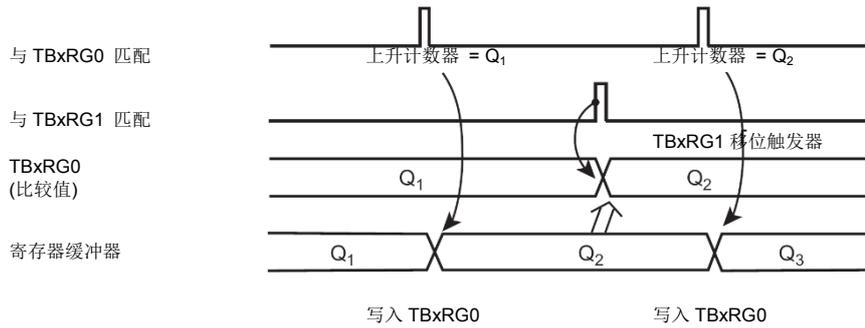


图 10-3 寄存器缓冲操作

该模式的方块图如以下所示。

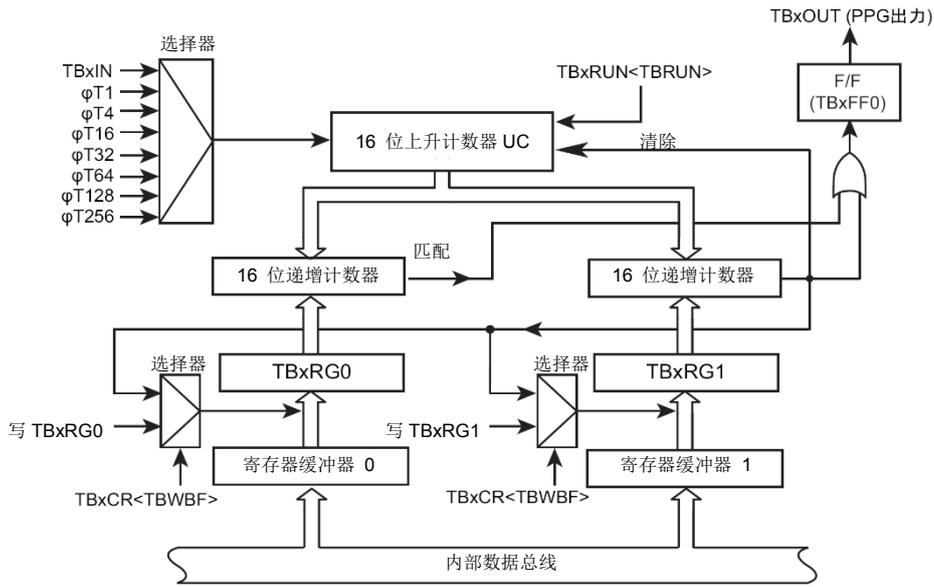


图 10-4 16 位 PPG 模式方块图

16 位 PPG 输出模式下各寄存器应编程如下。

	7	6	5	4	3	2	1	0	
TBxEN	← 1	X	X	X	X	X	X	X	启用 TMRBx。
TBxRUN	← X	X	X	X	X	0	X	0	停止预分频器和计数器。
TBxCR	← 1	0	X	X	X	0	X	X	启用双缓冲。
TBxRG0	← *	*	*	*	*	*	*	*	设置占空。
TBxRG1	← *	*	*	*	*	*	*	*	设置周期。
TBxFFCR	← X	X	0	0	1	1	1	0	通过检测 TBxRG0 或 TBxRG1 和上升计数器之间的匹配， 设置 TBxFF0 使信号反转；通过捕获 TBxCP0 或 TBxCP1， 设置 TBxFF0 使信号不反转；将 TBxFF0 初始值设为"0"。
TBxMOD	← X	1	0	0	1	*	*	*	将预分频器输出时钟指定为输入时钟，并禁用捕获功能。

(*** = 001 ~ 111)

将对应端口分配给 TBxOUT。

TBxRUN	← X	X	X	X	X	1	X	1	启动预分频器和计数器。
--------	-----	---	---	---	---	---	---	---	-------------

注: X: 忽略, *: 可选值, -: 不要改变

10.5.4 可编程脉冲发生(PPG)外部触发脉冲输出模式

在 PPG(可编程脉冲发生)输出模式下, 当用外部触发器计数启动模式启动 TMRB_x 时, 能输出短暂延时的 PPG 波。

在外部触发器计数启动模式下(延时的)单触发脉冲输出示例如下。

为了在外部触发器计数启动模式下启动 TMRB, 当 16 位计数器已停止时, 将 TB_xCR<CSSEL>设为"1", 将 TB_xCR<TRGSEL>设为"0"而在上升沿递增计数 TB_xIN。

TB_xRG0 设置外部触发信号的延时时间(d)。TB_xRG1 设置值(d)+(p), 其中延时时间(d)被加到单触发脉冲宽度(p)中。

当上升计数器与 TB_xRG0/1 匹配时, 为了反转 TB_xFF0, 将 TB_xFFCR<TBE1T1>, <TBE0T1>设为"1"。

为了启动上升计数器, 将 TB_xRUN<TBPRUN>, <TBRUN>设为"1"。

此时, 若外部触发脉冲被输入 TB_xIN, 则上升计数器在外部触发脉冲上升沿启动。

当上升计数器递增计数到(d)时, TB_xFF0 反转。它先与 TB_xRG0 匹配, 然后 TB_xFF0 变成"高"电平。

当上升计数器递增计数到(d)+(p)时, TB_xFF0 反转。它先与 TB_xRG1 匹配, 然后 TB_xFF0 变成"低"电平。

为了避免上升计数器与 TB_xRG1 匹配时发生的 INTTB_x 改变 TB_xFF0 的电平, 将 TB_xFFCR<TBE1T1>, <TBE0T1>清"0"或者由 TB_xRUN<TBPRUN>, <TBRUN>停止上升计数器。

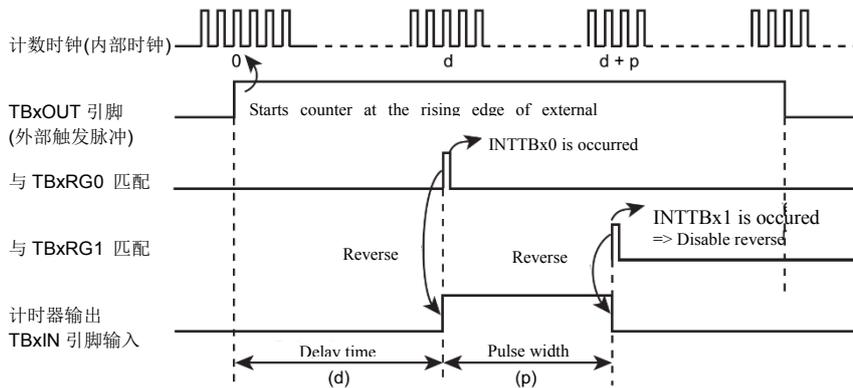


图 10-5 在外部触发器计数启动模式下延时的单触发脉冲输出

在 3 ms 后, 通过触发上升沿的 TB_xIN 输入, 输出 2 ms 宽度的单触发脉冲的情况如下所示。在本例中, 源时钟为 φT1。

7 6 5 4 3 2 1 0

[主处理]

将对应端口分配给 TBxIN。

TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx 运行。
TBxRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。
TBxRG0	←	*	*	*	*	*	*	*	*	设置计数值。(3ms/φT1)
TBxRG0	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	设置计数值。(3+2)ms/φT1
TBxRG1	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	0	0	1	1	1	0	通过检测 TBxRG0 或 TBxRG1 和上升计数器之间的匹配, 设置 TBxFF0 使信号反转; 通过捕获 TBxCP0 或 TBxCP1, 设置 TBxFF0 使信号不反转。将 TBxFF0 初始值设为"0"。
TBxMOD	←	X	1	0	0	0	0	0	1	以自由运行方式启动上升计数器。选择源时钟 φT1。禁用捕获上升计数器值。

将对应端口分配给 TBxOUT。

TBxIM	←	X	X	X	X	X	1	0	1	除 TBxRG1 中断外, 进行屏蔽。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	将 INTTBx 中断对应位设为"1", 允许产生中断。
TBxRUN	←	X	X	X	X	X	1	X	1	启动预分频器和计数器。

[INTTBx 中断服务程序的处理] 禁用输出

TBxFFCR	←	X	X	-	-	0	0	-	-	清除 TBxFF0 反向触发器设置值
TBxRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。

注: X: 忽略, *: 可选值, -: 不要改变

10.6 捕获功能应用

在许多应用中可使用捕获功能。
应用如下。

1. 频率测量
2. 脉冲宽度测量

10.6.1 频率测量

外部时钟频率测量示例如下。

在本节中，TMRB_m 用作 16 位间隔计时器模式，TMRB_n 用作 16 位事件计数器模式。

为了用外部时钟自由地计数 TMRB_n 的上升计数器，将 TM_nMOD<TBCLK> 设为 "000"，将 TB_nRUN<TBE1T1><TBE0T1> 设为 "11"。

为了在 TMRB_m 的上升计数器与 TB_mRG0 和 TB_mRG1 匹配时反转 TB_mFF0，将 TB_mFFCR<TBE1T1><TBE0T1> 设为 "11"。

为了在 TB_mFF0 上升沿将上升计数器值捕入 TB_nCP0 及在 TB_mFF0 下降沿将上升计数器值捕入 TB_nCP1，将 TB_nMOD<TBCPM[1:0]> 设为 "11"。

将 TB_mRG0/1 设为计数外部时钟数，启动 TMRB_m。

当 TMRB_m 上升计数器值与 TB_mRG0 匹配时，TB_mFF0 升高，TMRB_n 的上升计数器值被捕入 TB_nCP0。当 TMRB_m 的上升计数器与 TB_mRG1 匹配时，TB_mFF0 下降，TMRB_n 的上升计数器值被捕入 TB_nCP1。

用 INTTB_m 就能从 $(TB_{n}CP1 - TB_{n}CP0) \div (TB_{m}RG1 - TB_{m}RG0)$ 中测定频率。

例如，TB_mRG1 和 TB_mRG0 之差为 0.5 s，TB_nCP1 和 TB_nCP0 之差为 100，则频率为 200 Hz ($100 \div 0.5 \text{ s} = 200 \text{ Hz}$)。

根据 TB_mFF0 变化的计时，TB_nCP1 - TB_nCP0 的结果为负。若 TB_nCP1 - TB_nCP0 为负，则对值进行修正。

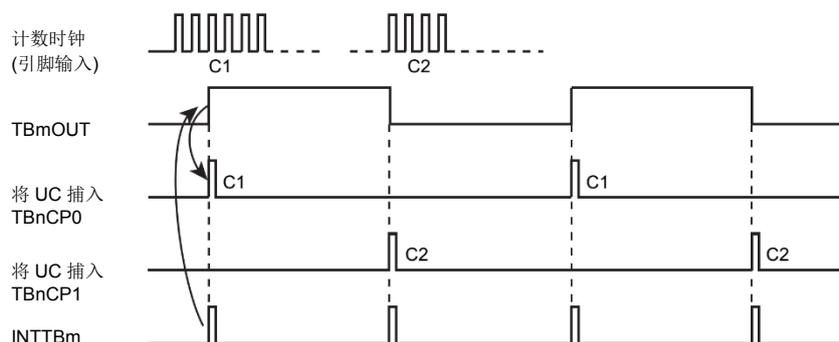


图 10-6 频率测量

测得的脉冲输入 TB_nIN 的情况如下所示。在本例中，源时钟为 φT1。

		7	6	5	4	3	2	1	0	
[主处理] 用 TBmFF0 捕获设置值										
		7	6	5	4	3	2	1	0	
将对端口分配给 TBnIN。										
TBmEN	←	1	X	X	X	X	X	X	X	启用 TMRBm 操作。
TBmRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。
TBnEN	←	1	X	X	X	X	X	X	X	启用 TMRBn。
TBnRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。
TBmCR	←	1	0	X	X	X	0	X	X	启用双缓冲。
TBmRG0	←	*	*	*	*	*	*	*	*	设置外部时钟测量时间 1。
	←	*	*	*	*	*	*	*	*	
TBmRG1	←	*	*	*	*	*	*	*	*	设置外部时钟测量时间 2。
	←	*	*	*	*	*	*	*	*	
TBmFFCR	←	X	X	0	0	1	1	1	0	通过检测 TBmRG0 或 TBmRG1 和上升计数器之间的匹配, 设置 TBmFF0 使信号反转。通过捕获 TBmCP0 或 TBmCP1, 设置 TBmFF0 不使信号反转。将 TBmFF0 初始值设为"0"。
TBnMOD	←	0	1	1	1	0	0	0	0	在上升/下降沿捕获。清除并禁用上升计数器。将输入时钟设为 TBnIN。
TBmIM	←	X	X	X	X	X	1	0	1	除 TBxRG1 中断外, 进行屏蔽。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	为了启用中断, 将 INTTBm 中断对应位设为"1"。
TBnRUN	←	X	X	X	X	X	1	X	1	启动预分频器和计数器。
TBmRUN	←	X	X	X	X	X	1	X	1	启动预分频器和计数器。
[INTTBm 中断服务程序的处理]										
TBmFFCR	←	X	X	-	-	0	0	-	-	清除 TBxFF0 反向触发器设置值
中断赋能清除寄存器	←	*	*	*	*	*	*	*	*	将"1"设置至与 INTTBm 对应的位, 来禁用中断

读出 TBnCP0 和 TBnCP1, 并计算频率。

注: X: 忽略, *: 可选值, -: 不要改变, m, n: 可选通道号

10.6.2 脉冲宽度测量

能测量外部脉冲“高”电平宽度。

为了在 TBxIN 上升沿将上升计数器值插入 TBxCP0 及在 TBxIN 下降沿将上升计数器值插入 TBxCP1，将 TBxMOD<TBxCPM> 设为“10”。

启用 INTTBxCAP1 中断。

启用 TMRBx。

若外部脉冲上升沿被输入 TBxIN，则上升计数器值被插入 TBxCP0。若外部脉冲下降沿被输入 TBxIN，则上升计数器值被插入 TBxCP1，并发生 IN-TTBxCAP1 中断。

用 TBxCP0 与 TBxCP1 之差乘以预分频器输出时钟的时钟周期，可计算出外部脉冲“高”电平宽度。

例如，若 TBxCP0 与 TBxCP1 之差为 100，预分频器输出时钟的周期为 $0.5 \mu\text{s}$ ，则脉冲宽度为 $100 \times 0.5 \mu\text{s} = 50 \mu\text{s}$ 。

若脉冲宽度超过上升计数器最大计数时间，则进行修正。

此外，能测量外部脉冲“低”电平宽度。

计算“低”电平宽度时，启用 INTTBxCAP0 中断并用“图 10-7 脉冲宽度测量”中第 2 次 INTTBxCAP0 时第一个 C2 与第 2 个 C1 之差乘以预分频器输出时钟周期。

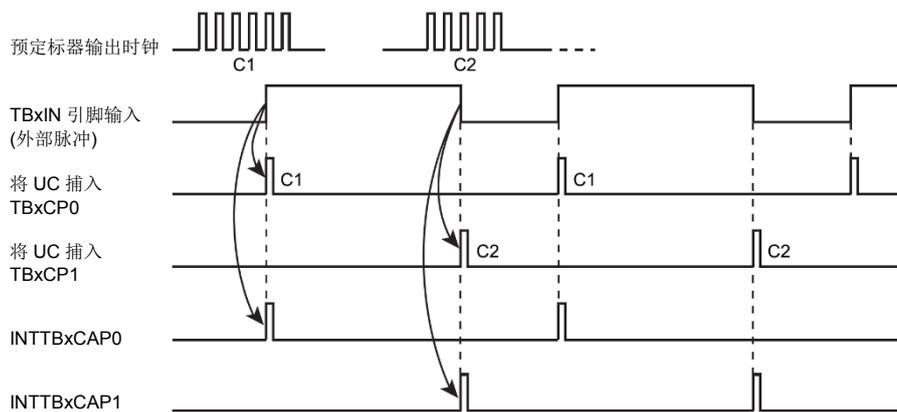


图 10-7 脉冲宽度测量

输入 TBxIN 的外部脉冲“高”电平宽度测量示例如下所示。在本例中，源时钟为 $\phi T1$ 。

		7	6	5	4	3	2	1	0	
[主处理] 捕获设置 TBxIN。										
将对应端口分配给 TBxIN。										
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx
TBxRUN	←	X	X	X	X	X	0	X	0	停止预分频器和计数器。
TBxFFCR	←	X	X	0	0	0	0	1	0	若 TMRBx 检测到 TBxRG0 或 TBxRG1 和上升计数器之间的匹配，或者当捕获 TBxCP0 或 TBxCP1 时，设置 TBxFF0 不使信号反转。 将 TBxFF0 初始值设为"0"。
TBxMOD	←	X	1	1	0	0	0	0	1	以自由运行方式启动上升计数器。选择源时钟的 φT1。上升计数器在 TBxIN 上升沿被捕入 TBxCP0。上升计数器在 TBxIN 下降沿被捕入 TBxCP1。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	为了启用中断，将 INTTBxCAP1 中断对应位设为"1"。
TBxRUN	←	X	X	X	X	X	1	X	1	启动预分频器和计数器。
[处理 INTTBxCAP1 中断服务程序] 计算"高"电平宽度。										
中断赋能清除寄存器	←	*	*	*	*	*	*	*	*	为了禁用中断，将 INTTBxCAP1 中断对应位设为"1"。
通过读取 TBxRG0 和 TBxRG1 来计算"高"电平。										
注: X; 忽略, *; 可选值, -; 不要改变										

11.16 位计时器 A (TMR16A Ver.B)

11.1 概要

TMR16A 包含下列功能：

- 匹配中断
- 方波形输出
- 读取捕获

在本章中，"x"指通道号。

11.2 方块图

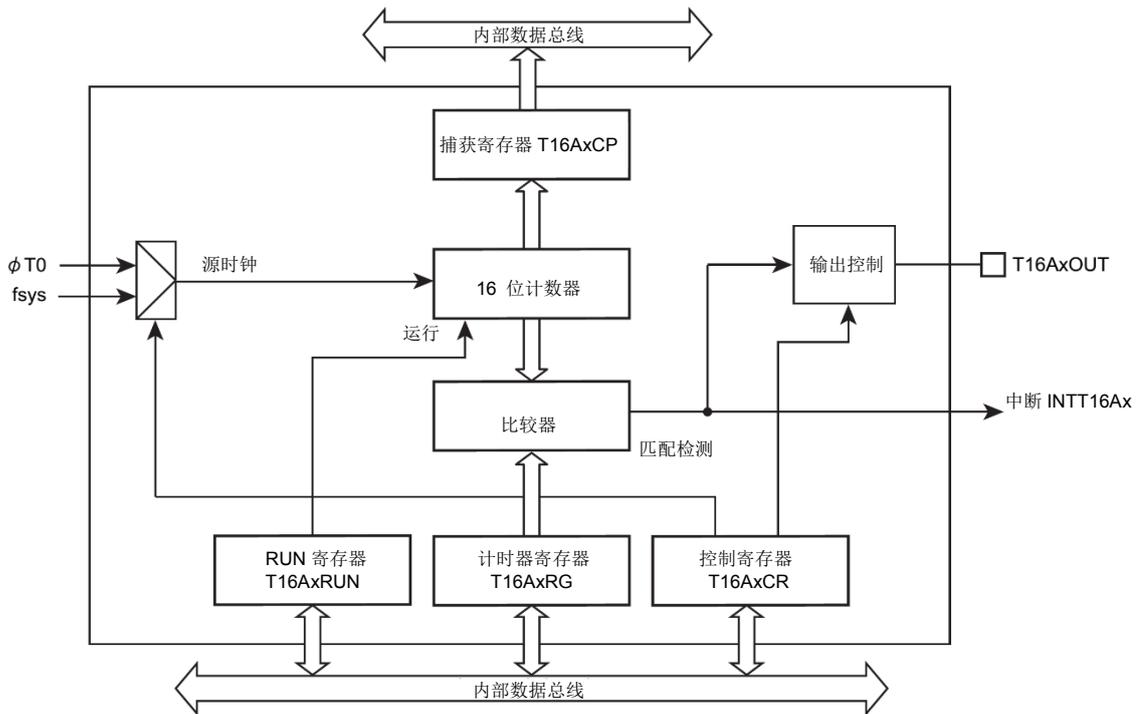


图 11-1 TMR16A 方块图

11.3 寄存器

11.3.1 寄存器列表

下表显示控制寄存器及其地址。

有关基地址的详细情况，参照“内存印象图”一章的“外部功能地址表”。

寄存器名称		地址(基+)
启用寄存器	T16AxEN	0x0000
RUN 寄存器	T16AxRUN	0x0004
控制寄存器	T16AxCR	0x0008
计时器寄存器	T16AxRG	0x000C
捕获寄存器	T16AxCP	0x0010

注：当 T16ARUN<RUN> 设为“1”时，不要修改 T16AxEN, T16AxCR, T16AxRG 和 T16AxCP。

11.3.1.1 T16AxEN(启用寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	HALT	I2T16A
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-2	-	R	读作"0"。
1	HALT	R/W	在停止模式调试时工作状态 0: 操作 1: 停止 指定停止模式调试时工作状态。将"1"写入该位, 停止工作。
0	I2T16A	R/W	在 IDLE 模式时工作状态 0: 停止 1: 操作 指定 IDLE 模式时工作状态。将"1"写入该位, 继续工作。

11.3.1.2 T16AxRUN(RUN 寄存器)

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

位	比特符号	类型	功能
31-1	-	R	读作"0"。
0	RUN	R/W	计数器工作状态 0: 停止 1: 操作

11.3.1.3 T16AxCR(控制寄存器)

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

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15	-	R/W	写入"0"。
14-8	-	R	读作"0"。
7	FFEN	R/W	T16AxOUT 反转 0: 禁用 1: 启用 当计数器与 T16ARG 匹配时, 将"1"写入该位, 使 T16AxOUT 反转。
6	-	R	读作"0"。
5-4	FFCR[1:0]	W	T16AxOUT 控制 00: 反转 01: 设置 10: 清除 11: 无操作 将数值写入该位, 用软件控制 T16AxOUT。读作"11"。
3-1	-	R	读作"0"。
0	CLK	R/W	源时钟 0: fsys 1: $\Phi T0$ 指定源时钟。

11.3.1.4 T16xRG(计时器寄存器)

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

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	RG[15:0]	R/W	设置与计数器比较的值

注：不要设置"0x0000"。

11.3.1.5 T16xCP(捕获寄存器)

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

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	CP[15:0]	R	计数器值 [读取] 读取当前计数器值。

11.4 操作描述

11.4.1 计时器操作

1. 准备工作

用 T16AxCR<CLK> 选择源时钟。写入 "0" 设置 fsys 或者写入 "1" 设置 $\Phi T0$ 。将 T16AxRG<RG[15:0]> 设为计数器值。

2. 计数器工作状态

在开始进行计数器操作前，将 T16AxCP<CP> 设为 "0x0000"，以清除计数器。

为了开始递增计数，将 T16AxRUN<RUN> 设为 "1"。若计数器值与 T16AxRG<RG[15:0]> 值匹配，它将被清除到 "0x0000"，并继续递增计数。

3. 匹配检测中断产生

若计数器值与 T16AxRG<RG[15:0]> 值匹配，则输出匹配检测中断 INTT16Ax。

4. 停止

为了停止计数，将 T16AxRUN<RUN> 设为 "0"。保持计数器值。然后在开始计数前，将 <RUN> 设为 "1"，清除计数器。

注：当计数器正停止时(T16AxRUN<RUN>设为"0")，必须修改 T16AxCR, T16AxRG 和 T16AxCP。

11.4.2 T16AxOUT 控制

用寄存器设置值或者使计数器与 T16AxRG 匹配，就可修改 T16AxOUT。

T16AxOUT 初始状态为 "0"。

1. 软件控制

用 T16AxCR<FFCR[1:0]> 设置值，就能指定 T16AxOUT；"1" 为设置，"0" 为清除，并且还能进行反转设置。

当计数器停止时(T16AxRUN<RUN>为"0")，修改 T16AxCR。

2. 因匹配计数器产生的反转

将 "1" 写入 T16ACR<FFEN>，就能使 T16AxOUT 反转。当 T16AxRG<RG[15:0]> 与计数器值匹配时，T16AxOUT 就会反转。当计数器停止时，T16AxOUT 的状态就会得到保持。

11.4.3 读取捕获

读取 T16AxCP<[15:0]>，就能捕获计数器当前值。

11.4.4 自动停止

用 T16AxEN<I2T16A> 或 <HALD> 设置值，TMR16A 在下列条件下自动停止：

1. 切换到 IDLE 模式/自 IDLE 模式切换

用 T16AxEN<I2T16A>设置值，就能指定 IDLE 模式时 TMR16Ax 工作状态。若设置"1"，则切换到 IDLE 模式时，TMR16A 自动停止。若 TMR16Ax 自 IDLE 模式返回，则它重启上升计数。

2. 调试停止

用 T16AxEN<HALT>设置值，就能指定调试时 TMR16Ax 工作状态。若设置"0"，则切换到调试停止模式时，TMR16Ax 自动停止上升计数。若取消内核调试停止模式，则重启上升计数。

12. 使用 4 字节 FIFO 的串行的通道(SIO/UART)

12.1. 概述

串行通道(SIO/UART) 具有如下所示的模式。

- 同步通信模式(I/O 接口模式)
- 异步通信模式(UART 模式)

其特点给出如下。

- 传送时钟
 - 预分频器将频率从外设时钟($\phi T0$)频率分频成 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128。
 - 能从预分频器输出时钟频率分频成 1 ~ 16。
 - 能从预分频器输出时钟频率分频成 $N + m / 16$ ($N = 2 \sim 15$, $m = 1 \sim 15$)。(仅 UART 模式)
 - 可用的系统时钟(fsyst)(仅 UART 模式)。
- 缓冲器
 - 可用的双缓冲器功能。
 - 能清除发送缓冲器。
- FIFO
 - 可用的 4 字节浮点输入浮点输出，包括发送与接收。
- I/O 接口模式
 - 传输模式：半双工(发送/接收)，全双工
 - 时钟：输出(固定上升沿)/输入(上升或下降沿可选)
 - 能指定连续发送的间隔时间。
 - 在输出最后一位后 SCxTXD 引脚状态的选择如下：
 - 保持"高"电平，"低"电平或最后一位的状态
 - 当在时钟输入模式下发生欠载错误时，SCxTXD 引脚状态的选择如下：
 - 保持"高"电平或"低"电平
 - 能在时钟输入模式下指定 SCxTXD 引脚最后一位保持时间。
- UART 模式
 - 数据长度：7 位, 8 位, 9 位
 - 增加奇偶校验位(针对 9 位数据长度)
 - 唤醒功能用串行链路
 - $\overline{\text{SCxCTS}}$ 引脚的握手功能
 - SCxRXD 引脚的噪声消除

在下列解释中，"x"代表通道号。

12.2. 配置

串行通道方块图和串行时钟发生器电路图如下所示。

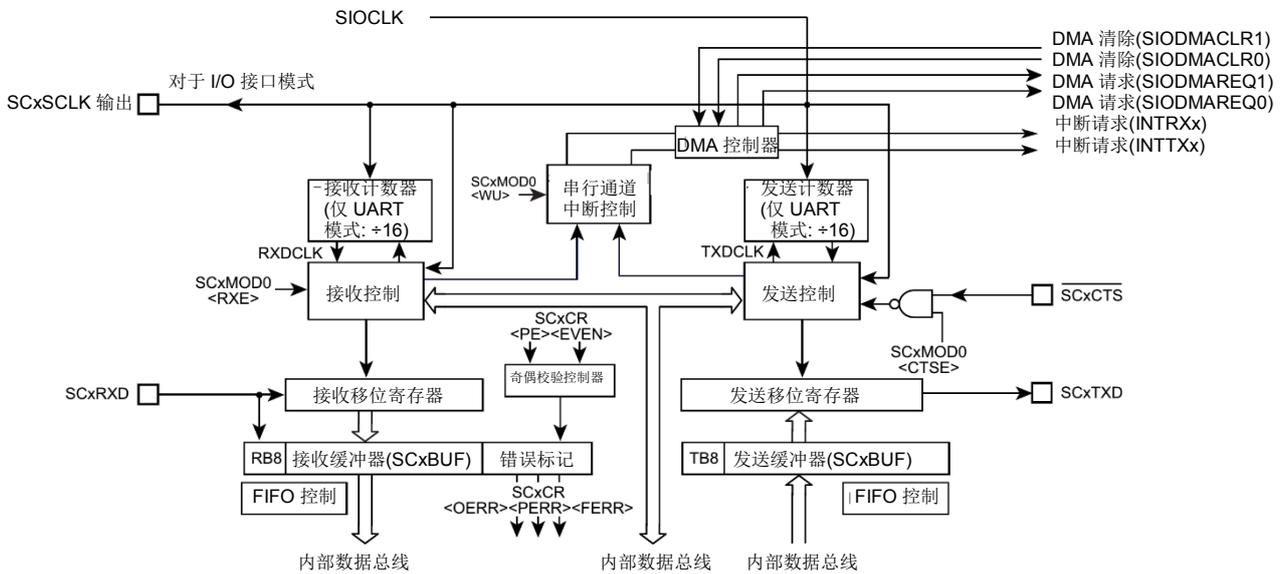


图 12-1 串行通道框图

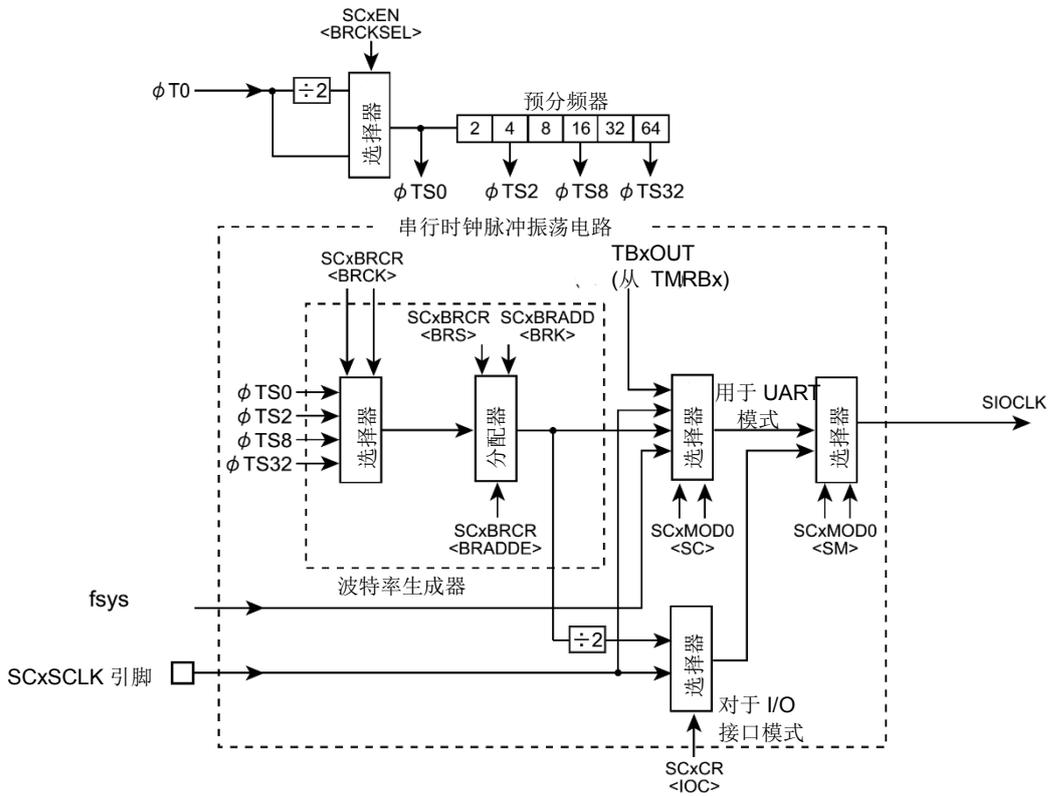


图 12-2 串行时钟产生电路方块图

12.3 寄存器描述

12.3.1 寄存器列表

下表显示控制寄存器及其地址。

有关基地址的详细情况，参照“内存印象图”一章的“外部功能地址表”。

寄存器名称		地址(基+)
启用寄存器	SCxEN	0x0000
缓冲寄存器	SCxBUF	0x0004
控制寄存器	SCxCR	0x0008
模式控制寄存器 0	SCxMOD0	0x000C
波特率发生器控制寄存器	SCxBRCR	0x0010
波特率发生器控制寄存器 2	SCxBRADD	0x0014
模式控制寄存器 1	SCxMOD1	0x0018
模式控制寄存器 2	SCxMOD2	0x001C
接收浮点输入浮点输出配置寄存器	SCxRFC	0x0020
发送浮点输入浮点输出配置寄存器	SCxTFC	0x0024
接收浮点输入浮点输出状态寄存器	SCxRST	0x0028
发送浮点输入浮点输出状态寄存器	SCxTST	0x002C
FIFO 配置寄存器	SCxFCNF	0x0030
DMA 请求启用寄存器	SCxDMA	0x0034

注：不要在数据发送或接收期间修改任何控制寄存器

12.3.2 SCxEN(启用寄存器)

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

位	比特符号	类型	功能
31-2	-	R	读作"0"。
1	BRCKSEL	R/W	选择预定标器的输入时钟。 0: $\phi T0/2$ 1: $\phi T0$
0	SIOE	R/W	串行通道操作 0: 禁用 1: 启用 指定的串行通道操作。 要使用串行通道, 设置<SIOE>="1"。 操作禁用时, 串行通道模块中无时钟可提供给另一个寄存器。此可降低功耗。 通道操作被执行随即禁用, 该设定值将保持在每个寄存器中。

12.3.3 SCxBUF(缓冲寄存器)

对于写操作, SCxBUF 作为发送缓冲器或 FIFO 工作, 或者对于读操作, 作为接收缓冲器或 FIFO 工作。

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7-0	TB[7:0] / RB [7:0]	R/W	[写] TB: 发送缓冲器或 FIFO [读] RB: 接收缓冲器或 FIFO

12.3.4 SCxCR(控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	EHOLD			-	TXDEMP	TIDLE	
复位后	0	0	0	0	0	1	1	0
	7	6	5	4	3	2	1	0
比特符号	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-15	-	R	读作"0"。
14-12	EHOLD[2:0]	R/W	在时钟输入模式下 SCxTXD 引脚最后一位保持时间(仅 I/O 接口模式) 设置最后一位保持时间和 SCLK 周期, 以保持最后一位保持时间等于或小于 SCLK 周期/2。 000: 2/fsys 100: 32/fsys 001: 4/fsys 101: 64/fsys 010: 8/fsys 110: 128/fsys 011: 16/fsys 111: 保留
11	-	R	读作"0"。
10	TXDEMP	R/W	在时钟输入模式下发生欠载错误时 SCxTXD 引脚状态。(仅 I/O 接口模式) 0: "低"电平输出 1: "高"电平输出
9-8	TIDLE[1:0]	R/W	在输出最后一位后 SCxTXD 引脚状态(仅 I/O 接口模式) 当<TIDLE[1:0]>设为"10"时, 将<EHOLD[2:0]>设为"000"。 00: 保持"低"电平输出 01: 保持"高"电平输出 10: 保持最后一位 11: 保留
7	RB8	R	接收数据第 8 位(仅 UART 模式) 以 9-位通用异步收发器模式接收数据的第 9 位。
6	EVEN	R/W	奇偶校验(仅 UART 模式) 选择偶或奇校验。奇偶检验位仅可用于 7-或 8-位通用异步收发器模式中。 0: 奇数 1: 偶数 选择偶或奇校验。
5	PE	R/W	增加奇偶校验(仅 UART 模式) 控制奇偶校验的禁用或启用。奇偶检验位仅可用于 7-或 8-位通用异步收发器模式中。 0: 禁用 1: 启用
4	OERR	R	超载运行出错标志(注) 0: 正常操作 1: 错误
3	PERR	R	奇偶校验/欠载运行出错标志(注) 0: 正常操作 1: 错误
2	FERR	R	成帧错误标志(注) 0: 正常操作 1: 错误

位	比特符号	类型	功能
1	SCLKS	R/W	<p>选择输入时钟沿(适用于输入/输出接口模式)</p> <p>在时钟脉冲输出模式下, 设置为"0"。</p> <p>0: 发送缓冲器的数据在 SCxRXD 引脚下降沿被发送到 SCxTXD 引脚的每一位。</p> <p>SCxRXD 引脚的数据在 SCxRXD 引脚上升沿被接收缓冲器的每一位接收。</p> <p>在这种情况下, SCxRXD 引脚状态从"高"电平开始。(上升沿模式)</p> <p>1: 发送缓冲器的数据在 SCxSCLK 引脚上升沿被发送到 SCxTXD 引脚的每一位。</p> <p>SCxRXD 引脚的数据在 SCxSCLK 引脚下降沿被接收缓冲器的每一位接收。在这种情况下, SCxSCLK 引脚状态从"低"电平开始。</p>
0	IOC	R/W	<p>选择时钟(适用于输入/输出接口模式)</p> <p>0: 时钟输出模式(传输时钟从 SCxSCLK 引脚输出。)</p> <p>1: 时钟输入模式(传输时钟被输入 SCxSCLK 引脚。)</p>

注: 当读取时, <OERR>, <PERR>和<FERR>清"0"。

12.3.5 SCxMOD0(模式控制寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TB8	CTSE	RXE	WU	SM		SC	
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TB8	R/W	发送数据第 8 位(仅 UART 模式) 以 9-位通用异步收发器模式写入发送数据的第 9 位。
6	CTSE	R/W	握手功能控制(仅 UART 模式) 0: CTS 禁用 1: CTS 启用 控制握手功能。 设置"1"启用 $\overline{\text{SCxCTS}}$ 引脚的握手功能。
5	RXE	R/W	接收控制(注 1)(注 2) 0: 禁用 1: 启用
4	WU	R/W	唤醒功能(仅 UART 模式) 0: 禁用 1: 启用 该功能仅在 9 位 UART 模式下可用。在其它模式下, 该功能无意义。当启用时, 仅在 9 位 UART 模式下 RB9 = "1" 时才会发生中断。
3-2	SM[1:0]	R/W	指定传送模式。 00: I/O 接口模式 01: 7-位 UART 模式 10: 8-位 UART 模式 11: 9-位 UART 模式
1-0	SC[1:0]	R/W	串行传输时钟(仅 UART 模式) 00: TMRB 输出 01: 波特率生成器 10: 系统时钟(fsys) 11: 外部时钟(SCxSCLK 引脚输入) (对于 I/O 接口模式, I/O 接口模式下的传输时钟由 SCxCR<IOC>选择。)

注 1: 首先确定所有模态控制寄存器, 然后<RXE>。

注 2: 正在接收数据时, 不要(通过将 SCxMOD0<RXE>设为"0")停止接收操作。

12.3.6 SCxMOD1(模式控制寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	I2SC	FDPX		TXE	SINT			-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	I2SC	R/W	IDLE 0: 停止 1: 操作 指定 IDLE 模式下的工作情况。
6-5	FDPX[1:0]	R/W	传送模式设置 00: 传送被禁用 01: 半双工(接收) 10: 半双工(发送) 11: 全双工 可在 I/O 接口模式下配置该传送模式。 并且当 FIFO 启用时, 指定 FIFO 的配置。在 UART 模式下, 仅指定 FIFO 的配置。
4	TXE	R/W	发送控制器(注 1)(注 2) 0: 禁用 1: 启用 该位可启用发送, 且在所有传送模式下均有效。
3-1	SINT[2:0]	R/W	连续发送的间隔时间(适用于 I/O 接口模式) 000: 无 001: 1 x SCLK 周期 010: 2 x SCLK 周期 011: 4 x SCLK 周期 100: 8 x SCLK 周期 101: 16 x SCLK 周期 110: 32 x SCLK 周期 111: 64 x SCLK 周期 当选择 SCLK 输出模式时, 该参数仅对 I/O 接口模式才有效。在其它模式中, 该参数没有意义。 当双缓冲或 FIFO 在 I/O 接口模式下启用时, 指定连续发送的间隔时间。
0	-	R/W	写作"0"。

注 1: 首先确定所有模式控制寄存器, 然后启用<TXE>。

注 2: 当正在发送数据时, 不要(通过将<TXE>设置到"0")停止发送操作。

12.3.7 SCxMOD2(模式控制寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TBEMP	RBFL	TXRUN	SBLN	DRCHG	WBUF	SWRST	
复位后	1	0	0	0	0	0	0	0

位	比特符号	类型	功能											
31-8	-	R	读作"0"。											
7	TBEMP	R	发送缓冲器为空标志 0: 满 1: 空 如果双缓冲被禁用, 则该标志可忽略。 该标志表示该传输双缓冲器为空。 在该传输双缓冲器中的数据被移动到传输移位寄存器, 且该双缓冲器为空时, 可将该位设置为"1"。 通过将数据重新写入到该双缓冲器, 即可将该位设置为"0"。											
6	RBFL	R	接收缓冲器已满标志。 0: 空 1: 满 如果双缓冲被禁用, 则该标志可忽略。 该标志表示接收双缓冲器已满。 当完成接收操作, 并且接收数据从接收移位寄存器移到接收双缓冲器时, 该位变为"1"。当读取接收缓冲器时, 该位清"0"。											
5	TXRUN	R	在传输标志中 0: 停止 1: 操作 该状态标志表示数据传输正在进行中。 <TXRUN>与<TBEMP>位可指示以下状态。 <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 15%;"><TXRUN></th> <th style="width: 15%;"><TBEMP></th> <th style="width: 70%;">状态</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>-</td> <td>传输中</td> </tr> <tr> <td rowspan="2">0</td> <td>1</td> <td>发送完成</td> </tr> <tr> <td>0</td> <td>等待状态, 数据在发送缓冲器中。</td> </tr> </tbody> </table>	<TXRUN>	<TBEMP>	状态	1	-	传输中	0	1	发送完成	0	等待状态, 数据在发送缓冲器中。
<TXRUN>	<TBEMP>	状态												
1	-	传输中												
0	1	发送完成												
	0	等待状态, 数据在发送缓冲器中。												
4	SBLN	R/W	停止位长度(适用于通用异步收发器模式) 0: 1 位 1: 2 位 这样可在 UART 模式下指定传输停止位的长度。 在接收侧, 不管<SBLN>如何, 仅使用单一位元作出决策。											
3	DRCHG	R/W	设置传送方向 0: LSB 先 1: MSB 先 指定数据传输方向。 在 UART 模式下, 首先将该位设置为 LSB。											
2	WBUF	R/W	启用双缓冲器 0: 禁用 1: 启用 该参数启用或禁用发送/接收双缓冲器在 I/O 接口模式下发送(在 SCLK 输出/输入模式)和接收(在 SCLK 输出模式)数据及在 UART 模式下发送数据。 当在 I/O 接口模式(时钟输入模式)和 UART 模式下接收数据, 不管<WBUF>如何, 双缓冲均会启用。											

位	比特符号	类型	功能										
1-0	SWRST [1:0]	R/W	<p>软件复位</p> <p>用"01"盖写"10"即可触发软件复位。</p> <p>当执行软件复位时，下列各位被初始化，发送/接收电路和 FIFO 变为初始状态(注 1)(注 2)。</p> <table border="1" style="margin-left: 20px; border-collapse: collapse; width: 80%;"> <thead> <tr> <th style="width: 40%;">寄存器</th> <th style="width: 60%;">位</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">SCxMOD0</td> <td style="text-align: center;"><RXE></td> </tr> <tr> <td style="text-align: center;">SCxMOD1</td> <td style="text-align: center;"><TXE></td> </tr> <tr> <td style="text-align: center;">SCxMOD2</td> <td style="text-align: center;"><TBEMP>, <RBFLL>, <TXRUN></td> </tr> <tr> <td style="text-align: center;">SCxCR</td> <td style="text-align: center;"><OERR>, <PERR>, <FERR></td> </tr> </tbody> </table>	寄存器	位	SCxMOD0	<RXE>	SCxMOD1	<TXE>	SCxMOD2	<TBEMP>, <RBFLL>, <TXRUN>	SCxCR	<OERR>, <PERR>, <FERR>
寄存器	位												
SCxMOD0	<RXE>												
SCxMOD1	<TXE>												
SCxMOD2	<TBEMP>, <RBFLL>, <TXRUN>												
SCxCR	<OERR>, <PERR>, <FERR>												

注 1: 在数据传输进行期间，任何软件复位操作均必须连续执行两次。

注 2: 在识别结束和软件复位指令执行开始之间的时间，软件复位要求 2 时钟的持续时间。

12.3.8 SCxBRCR(波特率发生器控制寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	-	R/W	写入"0"。
6	BRADDE	R/W	N + (16 - K) / 16 分频器功能(仅 UART 模式) 0: 禁用 1: 启用
5-4	BRCK[1:0]	R/W	选择波特率发生器的输入时钟。 00: φTS0 01: φTS2 10: φTS8 11: φTS32
3-0	BRS[3:0]	R/W	分频比"N" 0000: N = 16 0001: N = 1 0010: N = 2 ... 1111: N = 15

注 1: 当在 UART 模式下使用"N + (16 - K)/16"分频功能时, 1 ("0001")或 16 ("0000") 作为分频比, 不能应用于 N。

注 2: 只有在 I/O 接口模式并采用双缓冲时, 才能指定波特率发生器分频比"1"。

12.3.9 SCxBRADD(波特率发生器控制寄存器 2)

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

位	比特符号	类型	功能
31-4	-	R	读作"0"。
3-0	BRK[3:0]	R/W	指定" $N + (16 - K)/16$ "分频中的 K(仅 UART 模式) 0000: 禁止 0001: K = 1 0010: K = 2 ... 1111: K = 15

波特率发生器分频比的设置如表 12-1 所示。

表 12-1 设置分频比

	<BRADDE> = "0"	<BRADDE> = "1" (注 1) (仅 UART 模式)
<BRS>	指定"N" (注 2) (注 3)	
<BRK>	无设置要求	指定"K" (注 4)
分频比	除以 N	$N + \frac{(16 - K)}{16}$ 分频

注 1: 为了使用" $N + (16 - K)/16$ "分频功能, 应保证在将<BRK>设为 K 值后, 将<BRADDE>设为"1"。" $N + (16 - K)/16$ "分频功能仅能用在 UART 模式。

注 2: 禁用指定"K = 0"。

12.3.10 SCxFCNF(FIFO 配置寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	RFST	TFIE	RFIE	RXTXCNT	CNFG
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能						
31-8	-	R	读作"0"。						
7-5	-	R/W	务必写入"000"。						
4	RFST	R/W	接收 FIFO 中使用的字节。 0: 最高 1: 与接收 FIFO 的填充深度相同 选择要用的接收 FIFO 字节数。(注 1) 0: 已配置的 FIFO 最大字节数(亦见<CNFG>)。 1: 与 SC0RFC <RIL[1:0]>指定的产生接收中断的填充深度相同。						
3	TFIE	R/W	指定发送 FIFO 的发送中断。 0: 禁用 1: 启用 当启用发送浮点输入浮点输出时, 则由该参数启用发送中断信号或禁用发送中断信号。						
2	RFIE	R/W	指定接收 FIFO 的接收中断。 0: 禁用 1: 启用 当启用接收浮点输入浮点输出时, 则由该参数启用接收中断信号或禁用接收中断信号。						
1	RXTXCNT	R/W	RXE/TXE 的自动禁用 0: 无 1: 自动禁用 控制发送和接收的自动禁用。 设置"1"则启用如下操作。 <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 30%;">半双工(接收)</td> <td>当接收移位寄存器, 接收缓冲器和接收 FIFO 被填充到指定的有效字节数时, SCxMOD0<RXE>自动设为"0", 以禁用进一步接收。</td> </tr> <tr> <td>半双工(传输)</td> <td>当发送移位寄存器, 发送缓冲器和发送 FIFO 为空时, SCxMOD1<TXE>自动设为"0", 以禁用进一步发送。</td> </tr> <tr> <td>全双工</td> <td>当满足上述两种条件中的任一种时, <TXE>和<RXE>自动设为"0", 以禁用进一步发送和接收。</td> </tr> </table>	半双工(接收)	当接收移位寄存器, 接收缓冲器和接收 FIFO 被填充到指定的有效字节数时, SCxMOD0<RXE>自动设为"0", 以禁用进一步接收。	半双工(传输)	当发送移位寄存器, 发送缓冲器和发送 FIFO 为空时, SCxMOD1<TXE>自动设为"0", 以禁用进一步发送。	全双工	当满足上述两种条件中的任一种时, <TXE>和<RXE>自动设为"0", 以禁用进一步发送和接收。
半双工(接收)	当接收移位寄存器, 接收缓冲器和接收 FIFO 被填充到指定的有效字节数时, SCxMOD0<RXE>自动设为"0", 以禁用进一步接收。								
半双工(传输)	当发送移位寄存器, 发送缓冲器和发送 FIFO 为空时, SCxMOD1<TXE>自动设为"0", 以禁用进一步发送。								
全双工	当满足上述两种条件中的任一种时, <TXE>和<RXE>自动设为"0", 以禁用进一步发送和接收。								
0	CNFG	R/W	浮点输入浮点输出启用。 0: 禁用 1: 启用 启用 FIFO (注 2) 当<CNFG>设为"1"时, FIFO 启用。若 FIFO 启用, 则 SCMOD1 <FDPX[1:0]>的设置自动配置 FIFO 如下: <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td style="width: 30%;">半双工(接收)</td> <td>接收 FIFO 4 字节</td> </tr> <tr> <td>半双工(传输)</td> <td>发送 FIFO 4 字节</td> </tr> <tr> <td>全双工</td> <td>接收 FIFO 2 字节, 发送 FIFO 2 字节</td> </tr> </table>	半双工(接收)	接收 FIFO 4 字节	半双工(传输)	发送 FIFO 4 字节	全双工	接收 FIFO 2 字节, 发送 FIFO 2 字节
半双工(接收)	接收 FIFO 4 字节								
半双工(传输)	发送 FIFO 4 字节								
全双工	接收 FIFO 2 字节, 发送 FIFO 2 字节								

注 1: 关于发送 FIFO, 配置的最大字节数总是可用的。(另见<CNFG>。)

注 2: 浮点输入浮点输出不能用于 9 位通用异步收发器模式。

12.3.11 SCxRFC(接收 FIFO 配置寄存器)

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

位	比特符号	类型	功能															
31-8	-	R	读作"0"。															
7	RFCS	W	接收浮点输入浮点输出清除(注 1) 1: 清除 当 SCxRFC<RFCS>设为"1"时, 接收 FIFO 被清除, SCxRST<RLVL[2:0]>为"000"。并且读指针也被初始化。 读作"0"。															
6	RFIS	R/W	选择中断产生条件 0: 当 FIFO 填充深度(SCxRST<RLVL[2:0]>) = 产生接收中断的接收 FIFO 填充深度(<RIL [1:0]>)时 1: 当 FIFO 填充深度(SCxRST<RLVL[2:0]>) ≥ 产生接收中断的接收 FIFO 填充深度(<RIL [1:0]>)时 中断条件详见"12.12.1.2 FIFO"															
5-2	-	R	读作"0"。															
1-0	RIL[1:0]	R/W	产生接收中断的接收 FIFO 填充深度 <table border="1" style="margin-left: 20px; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>半双工</th> <th>全双工</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>4 字节</td> <td>2 字节</td> </tr> <tr> <td>01</td> <td>1 字节</td> <td>1 字节</td> </tr> <tr> <td>10</td> <td>2 字节</td> <td>2 字节</td> </tr> <tr> <td>11</td> <td>3 字节</td> <td>1 字节</td> </tr> </tbody> </table>		半双工	全双工	00	4 字节	2 字节	01	1 字节	1 字节	10	2 字节	2 字节	11	3 字节	1 字节
	半双工	全双工																
00	4 字节	2 字节																
01	1 字节	1 字节																
10	2 字节	2 字节																
11	3 字节	1 字节																

注: 为了使用发送/接收 FIFO 缓冲器, 在设置 SIO 传输模式(半双工/全双工) 并启用 FIFO (SCxFCNF<CNFG> = "1")后, 发送/接收 FIFO 必须被清除。

12.3.12 SCxTFC(发送 FIFO 配置寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	TBCLR
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TFCS	TFIS	-	-	-	-	TIL	
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能															
31-9	-	R	读作"0"。															
8	TBCLR	W	发送缓冲器清除 0: 忽略 1: 清除 当 SCxTFC<TBCLR>设为"1"时, 发送缓冲器被清除。 读作"0"。															
7	TFCS	W	发送浮点输入浮点输出清除(注 1) 0: 忽略 1: 清除 当 SCxTFC<TFCS>设为"1"时, 发送 FIFO 被清除, SCxTST<TLVL[2:0]>为"000"。并且写指针也被初始化。 读作"0"。															
6	TFIS	R/W	选择中断产生条件。 0: 当 FIFO 填充深度(SCxTST<TLVL[2:0]>) = 产生发送中断的发送 FIFO 填充深度(<TIL [1:0]>)时 1: 当 FIFO 填充深度(SCxTST<TLVL[2:0]>) ≤ 产生发送中断的发送 FIFO 填充深度(<TIL [1:0]>)时 中断条件详见"12.12.2.2 FIFO"															
5-2	-	R	读作"0"。															
1-0	TIL[1:0]	R/W	填入引起发生发送中断信号的能级。 <table border="1" style="margin-left: 20px;"> <tr> <td></td> <td>半双工</td> <td>全双工</td> </tr> <tr> <td>00</td> <td>空</td> <td>空</td> </tr> <tr> <td>01</td> <td>1 字节</td> <td>1 字节</td> </tr> <tr> <td>10</td> <td>2 字节</td> <td>空</td> </tr> <tr> <td>11</td> <td>3 字节</td> <td>1 字节</td> </tr> </table>		半双工	全双工	00	空	空	01	1 字节	1 字节	10	2 字节	空	11	3 字节	1 字节
	半双工	全双工																
00	空	空																
01	1 字节	1 字节																
10	2 字节	空																
11	3 字节	1 字节																

注 1: 为了使用发送/接收 FIFO 缓冲器, 在设置 SIO 传输模式(半双工/全双工) 并启用 FIFO (SCxFCNF<CNFG> = "1")后, 发送/接收 FIFO 必须被清除。

注 2: 若 SCxEN<SIOE> = "0"(停止 SIO/UART 操作)或者操作模式变为 IDLE 模式, SCxMOD<I2SC> = "0"(在 IDLE 模式下停止 SIO/UART 操作), 则 SCxTFC 被重新初始化。在您执行下列操作后, 请重新配置 SCxTFC 寄存器。

12.3.13 SCxRST(接收 FIFO 状态寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	ROR	R	接收 FIFO 超载运行。(注) 0: 未生成。 1: 生成。
6-3	-	R	读作"0"。
2-0	RLVL[2:0]	R	接收 FIFO 填入电平的状态。 000: 空 001: 1 字节 010: 2 字节 011: 3 字节 100: 4 字节

注: 当从 SCxBUF 读取接收数据时, <ROR>清"0"。

12.3.14 SCxTST(发送 FIFO 状态寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TUR	R	发送浮点输入浮点输出欠载运行。(注) 0: 未生成。 1: 生成。
6-3	-	R	读作"0"。
2-0	TLVL[2:0]	R	发送 FIFO 深度状态 000: 空 001: 1 个字节 010: 2 个字节 011: 3 个字节 100: 4 个字节

注: 当发送数据被写入 SCxBUF 时, <TUR>清"0"。

12.3.15 SCxDMA(DMA 请求启用寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	DMAEN1	DMAEN0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-2	-	R	读作"0"。
1	DMAEN1	R/W	启用 DMA 请求。由接收中断 INTRX 生成 DMA 请求。 0: 禁用 1: 启用
0	DMAEN0	R/W	启用 DMA 请求。DMA 请求由发送中断 INTTX 产生。 0: 禁用 1: 启用

注：在 DMA 传输期间生成 DMA 请求时，其不会得以保持并嵌套。

12.4 各模式下的操作

模式如表 12-2 所示。

表 12-2 模式

模式	类型	数据长度	传输方向	指定是否使用奇偶校验位	STOP 位长度(传输)
模式 0	同步通信模式 (I/O 接口模式)	8 位	LSB 先/MSB 先	-	-
模式 1	异步通信模式 (UART 模式)	7 位	LSB 先	o	1 位或 2 位
模式 2		8 位		o	
模式 3		9 位		x	

模式 0 为同步通信，能用于扩展 I/O。该模式与 SCLK 时钟同步发送和接收数据。SCLK 时钟既可用于输入，也可用于输出。可从 LSB 先或 MSB 先中选择数据传输方向。该模式不得使用奇偶校验位或 STOP 位。

模式 1，模式 2 和模式 3 为异步模式，传输方向只能选用 LSB 先。

奇偶校验位能在模式 1 和模式 2 时添加。模式 3 具有唤醒功能，在该功能中，主机控制器经串行链路启动从机控制器(多控制器系统)。可从 1 位和 2 位中选择发送中的 STOP 位。接收中的 STOP 位长度固定为一位。

12.5 数据格式

12.5.1 数据格式列表

数据格式如图 12-3 所示。

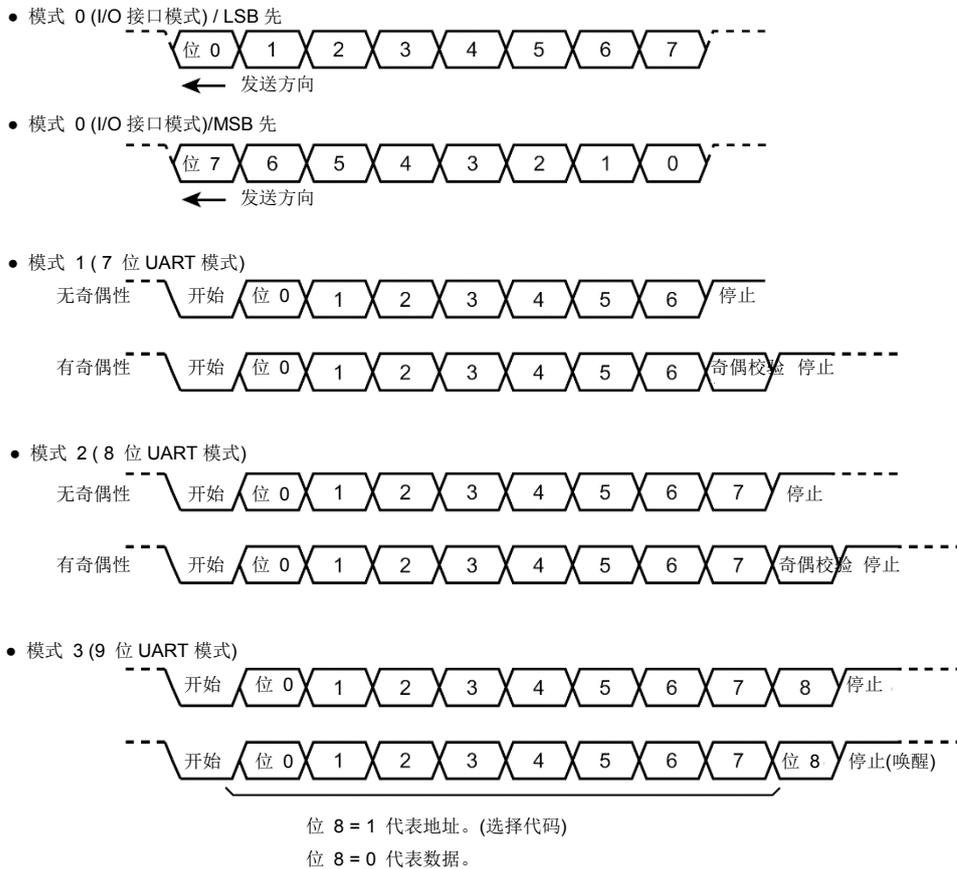


图 12-3 数据格式

12.5.2 奇偶控制

发送数据仅在 7 位或 8 位 UART 模式下才能添加奇偶校验位。

接收的奇偶校验位能与产生的奇偶校验位比较。

将 SCxCR<PE>设为"1"就可启用奇偶校验。SCxCR<EVEN>选择偶校验或奇校验。

12.5.2.1 发送

在数据传输后，奇偶控制电路自动生成奇偶性，数据位于发送缓冲器。奇偶校验位在 7 位 UART 模式下储存在 SCxBUF<TB7>中，在 8 位 UART 模式下储存在 SCxMOD<TB8>中。

<PE>和<EVEN> 的设置必须在数据写入发送缓冲器之前完成。

12.5.2.2 接收

若接收的数据从接收移位寄存器移到接收缓冲器，则会生成奇偶性。

在 7 位 UART 模式下，产生的奇偶校验位与 SCxBUF<RB7>中储存的奇偶校验位比较；在 8 位 UART 模式下，它与 SCxCR<RB8>中的比较。

若有差异，就会发生奇偶校验错误，SCxCR<PERR>设为"1"。

在 FIFO 的使用中，<PERR>表明，奇偶校验误差产生于所接收的数据之一中。

12.5.3 STOP 位长度

通过设置 SCxMOD2<SBLEN>，就能从一位或两位中选择 UART 发送模式下的 STOP 位长度。当收到 STOP 位数据时，不管该位的设置如何，其长度确定为一位。

12.6 时钟控制

12.6.1 预分频器

有一个 7-位预定标器，用 1, 2, 4, 8, 16, 32, 64 与 128 来分频预定标器输入时钟 $\phi T0$ 。
用时钟/模式控制块中的 CGSYSCR 和 SCxEN<BRCKSEL>选择预分频器的输入时钟。
仅在 SCxMOD0<SC[1:0]> = "01"而将波特率发生器选作传送时钟时，预分频器才被激活。

12.6.2 串行时钟生成电路

串行时钟产生电路是一个用于产生发送和接收时钟的块(SIOCLK)，由通过设置波特率发生器和模式来选择时钟的电路组成。

12.6.2.1 波特率发生器

波特率发生器生成发送和接收时钟，以确定串行通道传输率。

(1) 波特率发生器输入时钟

波特率发生器的输入时钟从预定标器输出中选取，这些输出除以 1, 4, 16 与 64。
该输入时钟通过设置 SCxEN<BRCKSEL>与 SCxBRCR<BRCK>来选取。

SCxEN<BRCKSEL>	SCxBRCR<BRCK>	波特率发生器 输入时钟 ϕT_x
0	00	$\phi T0/2$
0	01	$\phi T0/8$
0	10	$\phi T0/32$
0	11	$\phi T0/128$
1	00	$\phi T0$
1	01	$\phi T0/4$
1	10	$\phi T0/16$
1	11	$\phi T0/64$

(2) 波特率发生器输出时钟

波特率发生器中的输出时钟的分频比由 SCxBRCR 和 SCxBRADD 设置。

可使用以下分频比率：I/O 接口模式中的 1/N 分频，UART 模式中的 1/N 或 1/(N+(16-K)/16)分频。

能选择的分频比如下表所示。

模式	分频功能设置 SCxBRCR<BRADDE>	N 分频 SCxBRCR<BRS[3:0]>	K 分频 SCxBRADD<BRK[3:0]>
I/O 接口	除以 N	1 ~ 16 (注)	-
UART	除以 N	1 ~ 16	-
	$N + (16 - K) / 16$ 分频	2 ~ 15	1 ~ 15

注：只有当启用双缓冲器时才可使用 $1/N$ ($N=1$)分频比

波特率发生器的分频器输入时钟为 ϕ_{Tx} ，在 $1/N$ 和 $N+(16-K)/16$ 的情况下，波特率如下所示。

N 分频

$$\text{波特率} = \frac{\phi_{Tx}}{N}$$

$N + (16-K)/16$ 分频

$$\text{波特率} = \frac{\phi_{Tx}}{N + \frac{(16 - K)}{16}}$$

12.6.2.2 时钟选择电路

通过设置模式和寄存器选择时钟。

通过设置 SCxMOD0<SM[1:0]>指定模式。

通过设置 SCxCR<IOC><SCLKS>，便可选择在 I/O 接口模式时的时钟。

通过设置 SCxMOD0<SC[1:0]>，便可选择在 UART 模式时的时钟。

(1) 在 I/O 接口模式时的传输时钟

在 I/O 接口模式时的时钟选择如表 12-3 所示。

表 12-3 在 I/O 接口模式时的时钟选择

模式 SCxMOD0<SM[1:0]>	输入/输出选择 SCxCR<IOC>	时钟边沿选择 SCxCR<SCLKS>	使用的时钟
"00" (I/O 接口模式)	"0" (时钟输出模式)	"0" (传输: 下降沿, 接收: 上升沿)	波特率发生器输出由 2 分频
	"1" (时钟输入模式)	"0" (传输: 下降沿, 接收: 上升沿)	SCxSCLK 引脚输入
		"1" (传输: 上升沿, 接收: 下降沿)	SCxSCLK 引脚输入

为了使用 SCxSCLK 输入，必须满足下列条件。

- 若使用双缓冲器
-SCLK 周期 > 6/fsys
- 若未使用双缓冲器
-SCLK 周期 > 8/fsys

(2) UART 模式中的传输时钟

在 UART 模式时的时钟选择如表 12-4 所示。

在 UART 模式时，所选时钟在使用前在接收计数器或发送计数器中进行 16 分频。

表 12-4 在 UART 模式时的时钟选择

模式 SCxMOD0<SM[1:0]>	时钟选择 SCxMOD0<SC[1:0]>
UART 模式("01", "10", "11")	00: TMRB 输出
	01: 波特率生成器
	10: fsys
	11: SCxSCLK 引脚输入

为了使用 SCxSCLK 引脚输入，必须满足下列条件。

- SCLK 周期 > 2/fsys

启用计时器输出时，当计数器和 TBxRG1 的数值相匹配时，计数器反向输出。SIOCLK 时钟频率为"TBxRG1 设置值 x 2"。

波特率由下述公式计算所得。

波特率的计算

$$\text{传输率} = \frac{\text{由 CGSYSCR<PRCK[1:0]>选择的时钟脉冲频率}}{(\text{TBxRG1} \times 2) \times 2 \times 16}$$

↑ 如选择了计时器预分频器时钟 ΦT1 (2 分频比)。

└ 一个时钟周期为计时器触发器反向两次的时间。

12.6.3 发送接收缓冲器和 FIFO

12.6.3.1 配置

发送缓冲器，接收缓冲器和 FIFO 配置如图 12-4 所示。

使用缓冲器和 FIFO，必须进行适当的设置。配置可根据模式预先定义。

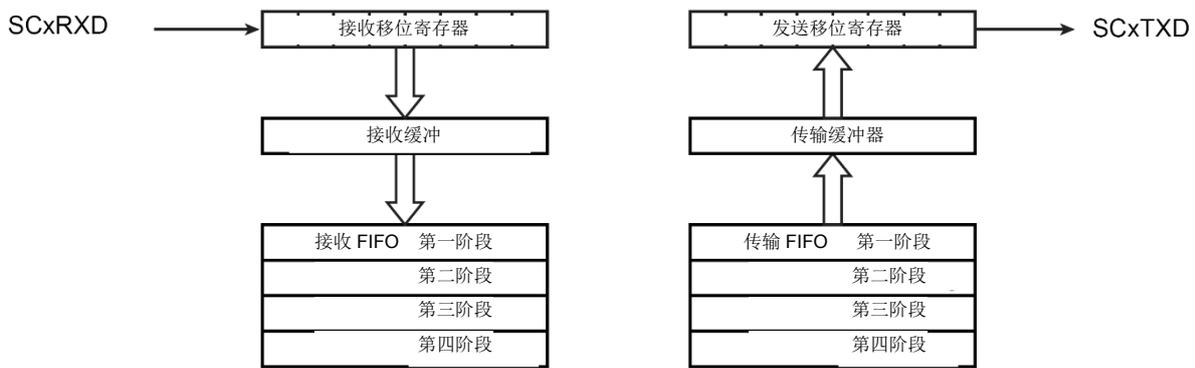


图 12-4 缓冲器和 FIFO 配置

12.6.3.2 发送/接收缓冲器

对发送缓冲器和接收缓冲器进行双缓冲。缓冲器配置由 SCxMOD2<WBUF>指定。

当串行通道以接收方式工作时，若它以 I/O 接口模式中的时钟输入模式工作，或者以 UART 模式工作，则不管<WBUF>的设置，仍进行双缓冲。

在其它模式时，它按照<WBUF>的设置。

模式和缓冲器之间的相关性如表 12-5 所示。

表 12-5 模式和缓冲器的组成

模式		SCxMOD2<WBUF>	
		"0"	"1"
UART 模式	传输	单	双
	接收	双	双
I/O 接口模式 (时钟输入模式)	传输	单	双
	接收	双	双
I/O 接口模式 (时钟输出模式)	传输	单	双
	接收	单	双

12.6.3.3 初始化发送缓冲器

当因发送缓冲器内的某一数据而停止发送时，必须在新传输数据写入发送缓冲器前将发送缓冲器初始化。

传输操作停止后必须将发送缓冲器进行初始化操作。通过读取 SCxMOD2<TXRUN>，就能确认发送操作的停止情况。在确认发送操作停止后，将 SCxTFC<TBCLR>设为"1"，并初始化发送缓冲器。

当启动发送 FIFO 时，根据发送 FIFO 中的数据进行初始化操作。若发送 FIFO 有数据，则数据从发送 FIFO 传输到发送缓冲器。若它没有数据，则 SCxMOD2<RBEMP>设为"1"。

注：在带时钟输入模式的 I/O 接口模式下，时钟被异步输入。当传输操作停止后，不要输入时钟。

12.6.3.4 FIFO

除上述双缓冲功能外，还能使用 4 字节 FIFO。

为了启用 FIFO，应通过将 SCxMOD2<WBUF>设为"1"及将 SCxFCNF<CNFG>设为"1"，启用双缓冲器。FIFO 缓冲器配置由 SCxMOD1<FDPX[1:0]>指定。

注：为了使用发送/接收 FIFO 缓冲器，在设置 SIO 传输模式(半双工/全双工)并启用 FIFO(SCxFCNF<CNFG> = "1")后，发送/接收 FIFO 必须被清除。

模式和 FIFO 之间的相关性如表 12-6 所示。

表 12-6 模式和 FIFO 的组成

	SCxMOD1<FDPX[1:0]>	接收 FIFO	发送 FIFO
半双工(接收)	01	4 字节	-
半双工(传输)	"10"	-	4 字节
全双工	"11"	2 字节	2 字节

12.7 状态标志

SCxMOD2 有两类标志。只有在双缓冲器启用时，该位才有效。

<RBFL>是一个显示接收缓冲器已满的标志。当收到一帧数据，并且数据从接收移位寄存器移到接收缓冲器时，该位变为"1"，当读取接收缓冲器时，该位清"0"。

<TBEMP>表示发送缓冲器为空。当发送缓冲器中的数据移至发送移位寄存器时，该位设为"1"。当数据被设定在发送缓冲器时，该位清"0"。

12.8 错误标志

在 SCxCR 中设有三个错误标志。标志的含义随模式而变化。在各模式时的含义如下表所示。

在读取 SCxCR 寄存器后，这些标志清"0"。

模式	标志		
	<OERR>	<PERR>	<FERR>
UART 模式	溢出错误	奇偶错误	成帧错误
I/O 接口模式 (时钟输入模式)	溢出错误	欠载错误 (当使用双缓冲器和 FIFO 时)	固定到"0"
		固定为 0 (当不使用双缓冲器和 FIFO 时)	
I/O 接口模式 (时钟输出模式)	未定义	未定义	固定到"0"

12.8.1 OERR 标志

在 UART 和 I/O 接口模式下，在接收缓冲器被读取前完成下一帧的接收而产生错误时，该位设为"1"。若接收 FIFO 启用，则接收数据自动移到接收 FIFO，不会产生溢出错误，直到接收 FIFO 满(或者直到可用字节被完全占用)。

在带时钟输出模式的 I/O 接口模式下，SCxSCLK 引脚输出在设置标志后停止。

注：为了将带时钟输出模式的 I/O 接口模式切换到其它模式，应读取 SCxCR 并清除溢出标志。

12.8.2 PERR 标志

此标志表示 UART 模式中的奇偶校验错误和 I/O 接口模式中的在运行错误或传输完成。

在 UART 模式下，当接收数据产生的奇偶校验位不同于接收的奇偶校验位时，<PERR>设为"1"。

在 I/O 接口模式下，当双缓冲器启用时，在下列条件下<PERR>设为"1"。

在时钟输入模式下，在完成发送移位寄存器的数据输出，发送缓冲器中无数据后，当输入时钟时，<PERR>设为"1"。

在时钟输出模式下，在完成所有数据输出及时钟输出停止后，<PERR>设为"1"。

注：为了将带时钟输出模式的 I/O 接口模式切换到其它模式，应读取 SCxCR 并清除欠载标志。

12.8.3 FERR 标志

若在中心周围对相应的停止位取样，该位被确定为"0"，则会生成帧错误。不管 SCxMOD2<SBLEN>中停止位长度设定如何，停止位状态只能由第 1 停止位确定。

I/O 接口模式下，该位固定至"0"。

12.9 接收

12.9.1 接收计数器

接收计数器为 4 位二进制计数器，并由 SIOCLK 向上计数。

在 UART 模式下，用 16 个 SIOCLK 时钟脉冲接收单一数据位，在第 8 个脉冲对数据符号进行采样。

12.9.2 接收控制单元

12.9.2.1 I/O 接口模式

在时钟输出模式下（SCxCR <IOC> 设为"0"），在 SCxSCLK 引脚上升沿对 SCxRXD 引脚采样。

在时钟输入模式下（SCxCR <IOC> 设为"1"），根据 SCxCR <SCLKS>，在 SCxSCLK 引脚上升沿或下降沿对 SCxRXD 引脚采样。

12.9.2.2 UART 模式

接收控制器有一个起始位检测电路，该电路用于在检测到正常起始位时启动接收操作。

12.9.3 接收操作

12.9.3.1 接收缓冲器

接收的数据按 1 位存储在接收移位寄存器中。当已存储一整套位元时，就会产生中断 INTRXx。

当双缓冲器启用时，数据移到接收缓冲器(SCxBUF)，接收缓冲器满标志(SCxMOD2<RBFL>)设为"1"。通过读取接收缓冲器，接收缓冲器全满标志清"0"。当双缓冲中止后，接收缓冲器标志则没有任何意义。

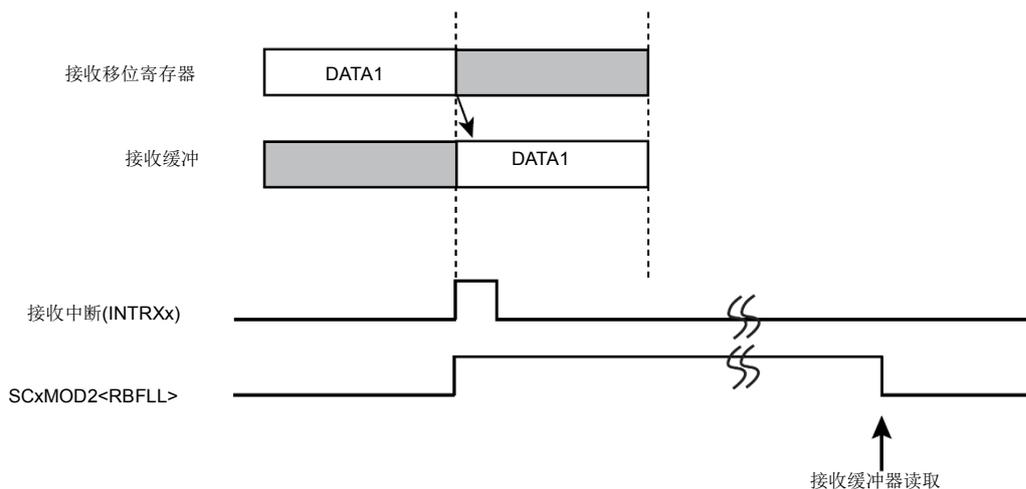


图 12-5 接收缓冲器操作

12.9.3.2 接收 FIFO 操作

启用 FIFO 时，接收的数据从接收缓冲器移到接收 FIFO，接收缓冲器全满标志被立即清除。根据 SCxRFC<RIL[1:0]> 产生中断。

注：当在 UART 模式中通过利用 FIFO 接收带奇偶校验位数据时，奇偶校验错误标志显示接收数据中发生的奇偶校验错误。

在半双工接收模式中的配置和操作说明如下。

- SCxMOD1<FDPx[1:0]> = "01" : 传输模式设为半双工模式
- SCxFCNF<RFST><TFIE><RFIE> : 在达到充满率后，自动禁用连续接收。
- <RXCNT><CNFG> = "10111": 在接收 FIFO 中使用的字节数与中断产生充满率相同。
- SCxRFC<RIL[1:0]> = "00" : 产生的接收中断设为 4 字节的 FIFO 填充深度
- SCxRFC<RFCS><RFIS> = "01" : 清除接收 FIFO，并设置中断产生的条件。

在上述 FIFO 配置设置后，数据接收通过将"1"写入 SCxMOD0<RXE>来启动。当所有数据存储在接受移位寄存器，接收缓冲器及接收 FIFO 时，SCxMOD0<RXE>自动清除，接收操作完成。

在上述条件下，若连续接收在达到填充深度后启用，则通过读取 FIFO 中的数据，就可能连续接收数据。

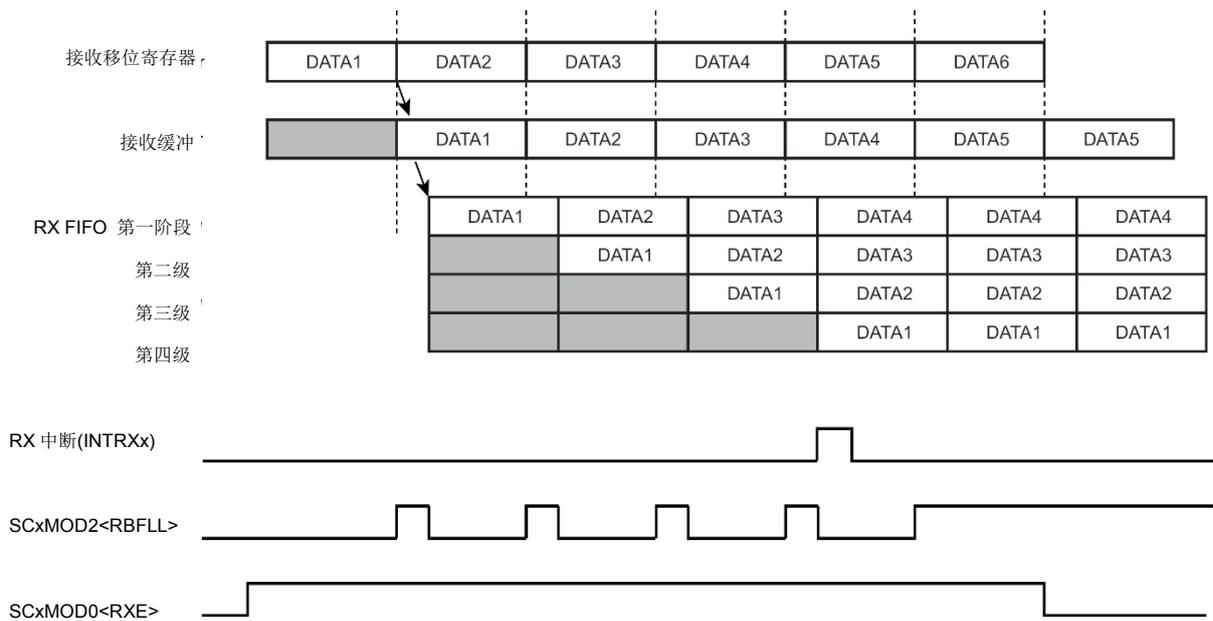


图 12-6 接收 FIFO 操作

12.9.3.3 I/O 接口模式（带有时钟输出模式）

在带时钟输出模式设置的 I/O 接口模式下，当所有接收数据存储在接受缓冲器和 FIFO 时，SCLK 输出停止。所以，在这种模式下，过运行错误标志没有意义。

SCLK 输出停止和再输出的时间取决于接收缓冲器和 FIFO。

(1) 单缓冲器的情况

在接收数据后，停止时钟输出。此模式下，I/O 接口通过握手可用传输装置传输各数据。

当读取缓冲器中的数据时，时钟输出被重启。

(2) 双缓冲器的情况

在数据被接收至接收移位寄存器和接收缓冲器后，停止时钟输出。

当读取数据时，时钟输出重启。

(3) FIFO 的情况

在数据被接收至移位寄存器，接收缓冲器和 FIFO 后，停止时钟输出。

当读取一个字节的的数据时，接收缓冲器内的数据被传输到 FIFO 内，接收移位寄存器内的数据被传输到接收缓冲器内，时钟输出重启。

若 SCxFCNF<RXTXCNT>设为"1"，则时钟停止，接收操作停止，并清除 SCxMOD0<RXE>。

12.9.3.4 读取接收数据

尽管启用或禁用 FIFO，仍会读取接收缓冲器(SCxBUF)的接收数据。

当接收 FIFO 被禁用时，通过读取，缓冲器满标志 SCxMOD2<RBFL>清"0"。在读取接收缓冲器数据前，接收移位寄存器能接收下一数据。在 8 位 UART 模式下要添加的奇偶校验位及在 9 位 UART 模式时最有效的位，将储存在 SCxCR<RB8>中。

当接收 FIFO 启用时，由于在 FIFO 中能存储高达 8 位数据，故禁用 9 位 UART 模式。在 8 位 UART 模式，虽然奇偶校验位丢失，但能确定奇偶校验错误，结果存储在 SCxCR<PERR>中。

12.9.3.5 唤醒功能

在 9 位 UART 模式下，将唤醒功能 SCxMOD0 <WU>设为"1"，从机控制器就能在唤醒模式下运行。在这种情况下，只有在 SCxCR <RB8>设为"1"时，才会产生中断 INTRXx。

12.9.3.6 超程错误

当禁用接收 FIFO 时，在接收下一个数据前未完成数据读取则会发生溢出错误。当发生溢出错误时，虽然接收缓冲器和 SCxCR<RB8>中的内容不丢失，但接收移位寄存器中的内容丢失。

当 FIFO 启用时，发生溢出错误，并且当 FIFO 满时，在下一数据移到接收缓冲器前，不读取 FIFO，由此设置溢出标志。在这种情况下，FIFO 未丢失。

在带时钟输出模式的 I/O 接口模式下，时钟输出自动停止，因此该标志无意义。

注：当模式从带时钟输出模式的 I/O 接口模式变为其它模式时，读取 SCxCR，并清除溢出标志。

12.10 发送

12.10.1 发送计数器

发送计数器是一个 4 位二进制计数器,和接收计数器的情况一样,由 SIOCLK 计数。UART 模式时,它在每第 16 个时钟脉冲时生成发送时钟(TXDCLK)。

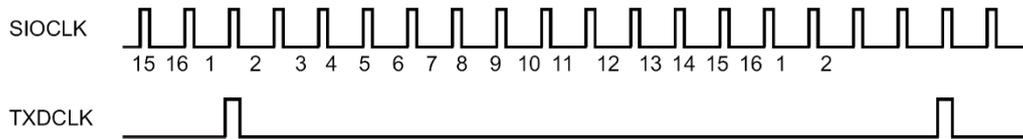


图 12-7 在 UART 模式下发送时钟的产生

12.10.2 发送控制

12.10.2.1 在 I/O 接口模式下

在 SCxCR<IOC>设为"0"的时钟输出模式下,发送缓冲器中各位数据在 SCxSCLK 引脚下降沿被输出到 SCxTXD 引脚。

在 SCxCR<IOC>设为"1"的时钟输入模式下,根据 SCxCR<SCLKS>,发送缓冲器中各位数据在 SCxSCLK 引脚下降沿被输出到 SCxTXD 引脚。

12.10.2.2 在 UART 模式下

当发送数据写入发送缓冲器时,数据发送在下一 TXDCLK 上升沿启动,并且也产生发送移位时钟信号。

12.10.3 发送操作

12.10.3.1 发送缓冲器的操作

若双缓冲被禁用, 则 CPU 将数据仅写入发送移位寄存器, 并在数据发送完成后, 产生发送中断 INTTXx。

若双缓冲启用(包括发送 FIFO 启用的情况), 则发送缓冲器中写入的数据移到发送移位寄存器。同时, 产生 INTTXx 中断, 发送缓冲器空标志(SCxMOD2<TBEMP>)设为"1"。该标志表示能写入下一发送数据。当下一数据写入发送缓冲器时, <TBEMP>标志清"0"。

图 12-8 发送缓冲器操作(双缓冲启用)

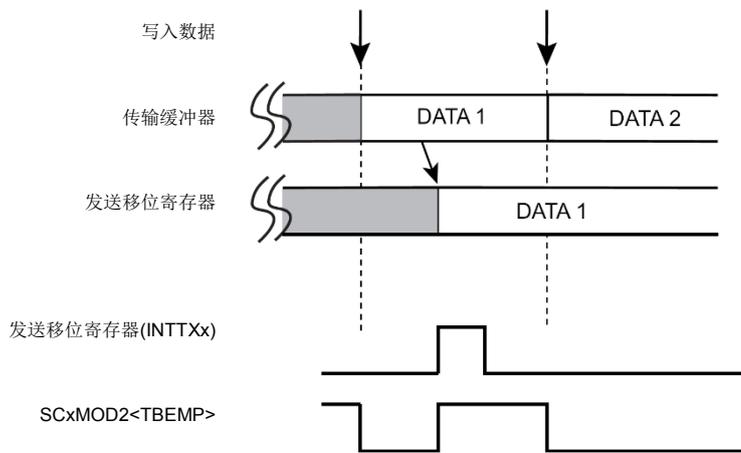


图 12-8 传输缓冲器的操作(双缓冲器已被启用)

12.10.3.2 发送 FIFO 操作

启用 FIFO，用发送缓冲器和 FIFO 能储存最多 5 字节的数据。一旦启用传输，数据就从发送缓冲器转移到发送移位寄存器，并开始传输。若在 FIFO 中存在数据，数据被立即移到发送缓冲器，<TBEMP>标志清"0"。

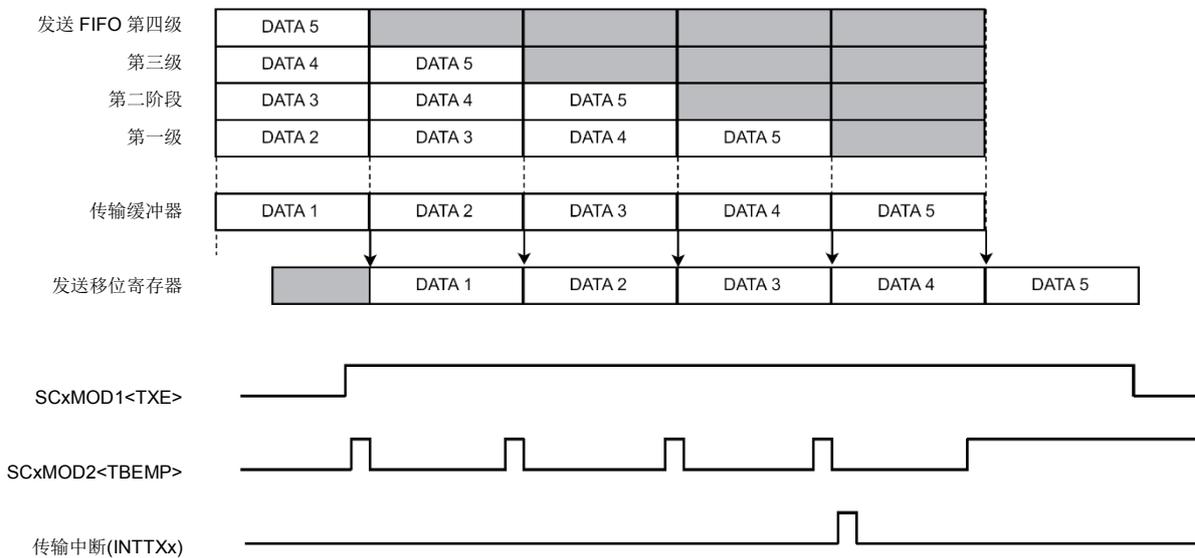
注：为了使用发送 FIFO 缓冲器，在设置 SIO 传输模式(半双工/全双工) 并启用 FIFO(SCxFCNF<CNFG> = "1")后发送 FIFO 必须被清除。

通过将传输模式设为半双工而发送 5 字节数据流的设置和操作如下所示。

- SCxMOD1<FDPX[1:0]> = "10" : 传输模式设为半双工。
- SCxFCNF<RFST><TFIE><RFIE> : 若 FIFO 为空，传输自动禁用。
- <RXTXCNT><CNFG> = "11011" : 在接收 FIFO 中使用的字节数与中断产生充满率相同。
- SCxTFC<TIL[1:0]> = "00" : 将中断产生充满率设为"0"。
- SCxTFC<TFCS><TFIS> = "11" : 清除接收 FIFO，并设置中断产生的条件。
- SCxFCNF<CNFG> = "1" : 启用 FIFO

在配置上述设置后，将 5 字节数据写入发送缓冲器和 FIFO，并将 SCxMOD1<TXE>位设为"1"，可启动数据发送当最后一个传输数据移动至发送缓冲器，产生传输中断。当最后一个数据发送完成时，时钟停止，发送顺序终止。

一旦进行了上述设置的配置，如果传输为设定为自动中止，则传输将持续写入传输数据。



12.10.3.3 在 I/O 接口模式(带有时钟输出模式) 下发送

在带时钟输出模式的 I/O 接口模式中，当所有数据发送完成并且发生在运行错误时，SCLK 输出自动停止。

时钟输出中止和恢复时序随缓冲器和 FIFO 使用情况不同。

(1) 单缓冲器

每当转移一帧数据时，时钟输出停止。可启用各数据与通信另一侧的握手。当下一数据写入缓冲器时，时钟输出恢复。

(2) 双缓冲器

在发送移位寄存器和发送缓冲器内的数据发送完成后，时钟输出停止。当下一数据写入缓冲器时，时钟输出恢复。

(3) FIFO

发送移位寄存器，发送缓冲器和 FIFO 中存储的所有数据发送完成，SCLK 输出停止。写入下一数据，时钟输出恢复。

若配置 SCxFCNF<RXTXCNT>，则在时钟停止的同时，清除 SCxMOD0<TXE>位，并且发送停止。

12.10.3.4 I/O 接口模式下最后一位输出后的 SCxTXD 引脚电平

在最后一位输出后，超过数据保持时间后的 SCxTXD 引脚电平由 SCxCR<TIDLE>指定。

当 SCxCR<TIDLE> 为"00"时，SCxTXD 引脚电平输出 "低"电平。当 SCxCR<TIDLE>为"01"时，SCxTXD 引脚电平输出"高"电平。当 SCxCR<TIDLE>为"10"时，SCxTXD 引脚输出最后一位的电平。

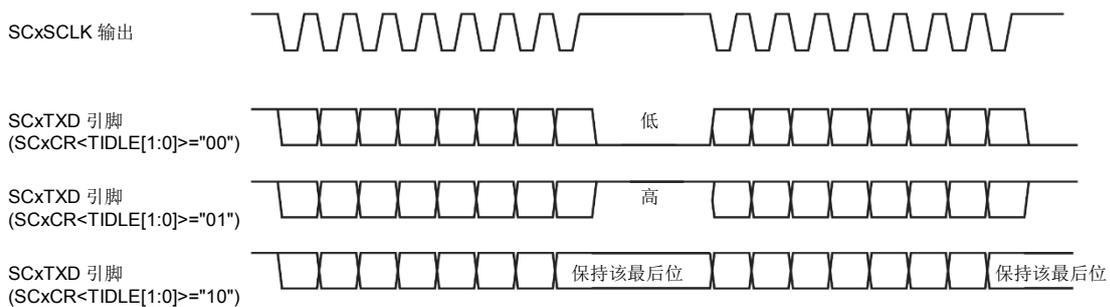


图 12-9 在最后一位被输出后 SCxTXD 引脚电平

12.10.3.5 欠载运行错误

在带时钟输入模式的 I/O 接口模式下及若 FIFO 为空及若在下一帧时钟输入前在发送缓冲器中未设置任何数据，并且这在发送移位寄存器的数据发送完成后发生，则发生欠载错误，SCxCR<PERR>设为"1"。

SCxTXD 引脚电平可由 SCxCR<TXDEMP>指定。当 SCxCR<TXDEMP>为"0"时，SCxTXD 引脚在数据输出期间输出"低"电平。当 SCxCR<TXDEMP>为"1"时，SCxTXD 引脚输出"高"电平。

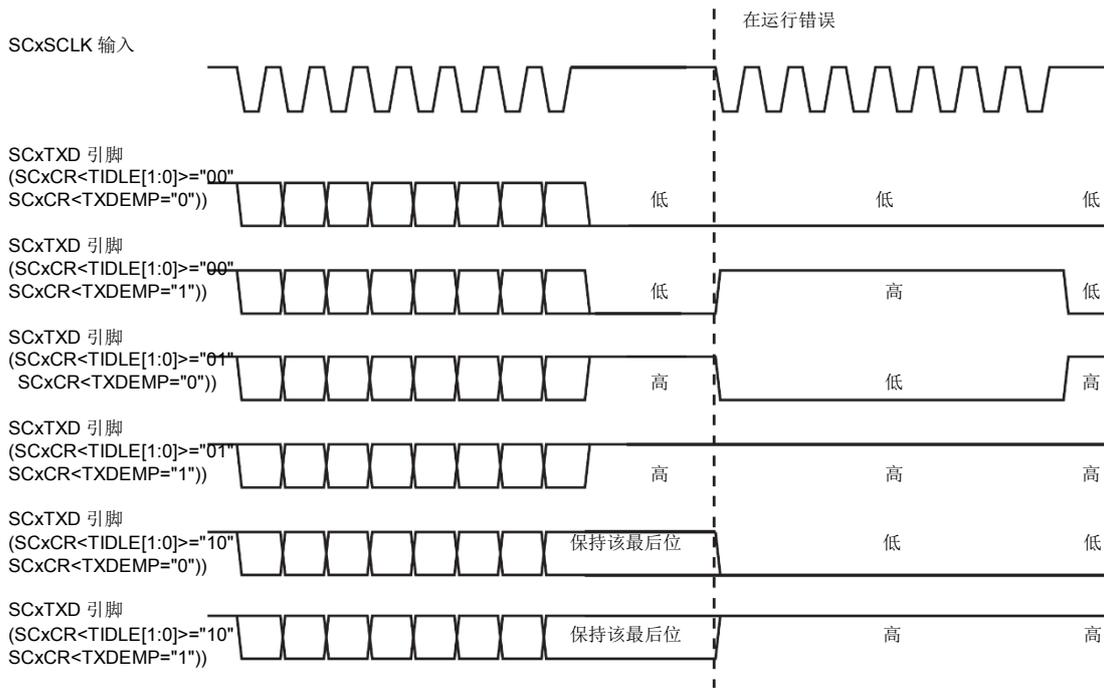


图 12-10 发生欠载错误时 SCxTXD 引脚

在带 SCLK 输出设置的 I/O 接口模式下，时钟输出自动停止，所以 SCxCR<PERR>没有意义。

注：在带时钟输出模式的 I/O 接口模式切换到其它模式前，读取 SCxCR 并清除欠载标志。

12.10.3.6 在 I/O 接口模式(带有时钟输入模式) 下的数据保持时间

在带时钟输入模式的 I/O 接口模式下，能用 SCxCR<EHOLD [2:0]>调整最后一位的数据保持时间。为了满足下列公式，应指定数据保持时间和 SCLK 周期。

$$\text{最后一位的数据保持时间} \leq \text{周期 SCLK} / 2$$

12.11 握手功能

握手功能使用 $\overline{\text{SCxCTS}}$ （清除发送）引脚启用逐帧数据发送，并防止溢出错误。该功能由 $\text{SCxMOD0}<\text{CTSE}>$ 启用或禁用。

当 $\overline{\text{SCxCTS}}$ 引脚设为"高"电平时，虽然能完成当前数据的发送，但下一数据的发送中止，直到 $\overline{\text{SCxCTS}}$ 引脚回到"低"电平。 INTTXx 中断在正常时序产生，下一发送数据写入发送缓冲器，并且它一直等到准备发送数据为止。

注 1: 若 $\overline{\text{CTS}}$ 信号在发送时设为"高"电平，则在当前发送完成后，下一数据的发送中止。

注 2: 在 $\overline{\text{CTS}}$ 设为"低"后，数据发送在 TXDCLK 时钟第一下降沿开始。

虽然未设有 $\overline{\text{RTS}}$ 引脚，但是通过给端口的一位分配 $\overline{\text{RTS}}$ 功能，就能轻易地实施握手控制功能。在数据接收完成后，通过（在接收中断程序中）将端口设为"高"电平，就能请求发送侧中止数据发送。

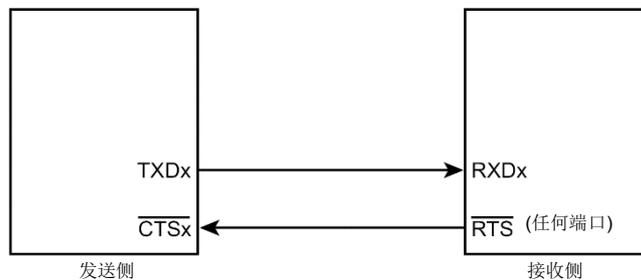


图 12-11 握手功能

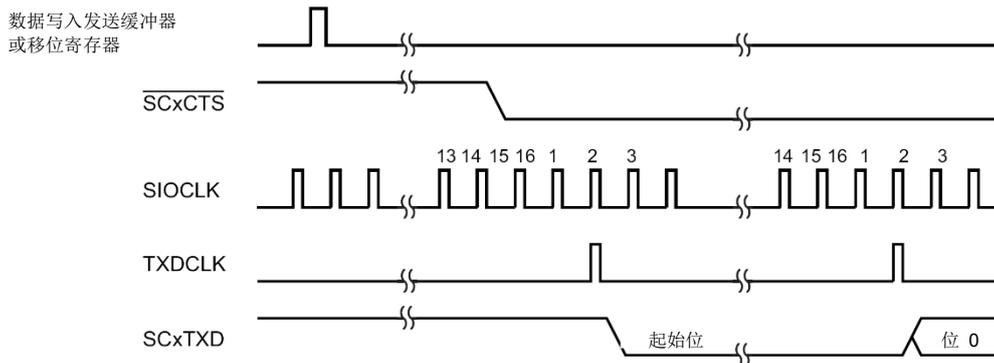


图 12-12 $\overline{\text{SCxCTS}}$ 信号时序

12.12 中断/错误生成定时

12.12.1 接收中断

接收操作的数据流及读取路径如图 12-13 所示。

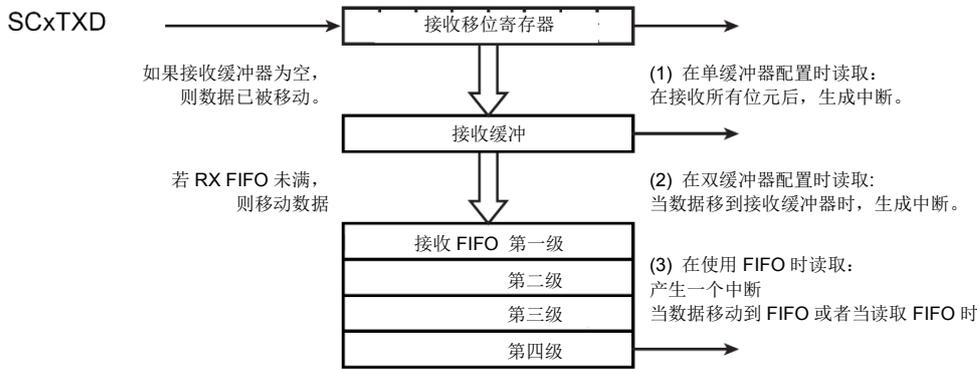


图 12-13 接收缓冲器/FIFO 配置框图

12.12.1.1 单缓冲器/双缓冲器

根据下列传输模式和缓冲器配置，在下列时间产生接收中断。

表 12-7 在使用单缓冲器/双缓冲器时接收中断条件

缓冲器配置	UART 模式	IO 接口模式
单缓冲器	-	恰在最后一个 SCxSCLK 引脚上升沿/下降沿后(上升或者下降根据 SCxCR<SCLKS>的设置确定。)
双缓冲器	当数据从接收移位寄存器传输到接收缓冲器时，发生接收中断。具体时序： <ul style="list-style-type: none"> 若在接收缓冲器中不存在数据，则在第 1 停止位中心附近发生接收中断。 若在接收移位寄存器和接收缓冲器中均存在数据，则当读取缓冲器时，发生接收中断。 	当数据从接收移位寄存器传输到接收缓冲器时，发生接收中断。具体时序： <ul style="list-style-type: none"> 若在接收缓冲器中不存在数据，则恰在最后一位的 SCxSCLK 引脚上升沿/下降沿发生接收中断。 (上升沿或下降沿的设置由 SCxCR<SCLKS>指定。) <ul style="list-style-type: none"> 若在接收移位寄存器和接收缓冲器中均存在数据，则当读取缓冲器时，发生接收中断。

注：当发生过运行错误时生成一个中断。

12.12.1.2 FIFO

当使用 FIFO 时，根据表 12-8 所述时序及 SCxRFC<RFIS>指定的条件，发生接收中断。

表 12-8 在使用 FIFO 时接收中断条件

SCxRFC<RFIS>	中断条件	中断产生时间。
"0"	当 FIFO 填充深度(SCxRST<RLVL[2:0]>) = 产生接收中断的接收 FIFO 填充深度<RIL[1:0]>时	<ul style="list-style-type: none"> 当接收数据从接收缓冲器传输到接收 FIFO 时 当读取接收 FIFO 的接收数据时
"1"	当 FIFO 填充深度(SCxRST<RLVL[2:0]>) > 产生接收中断的接收 FIFO 填充深度<RIL[1:0]>时	<ul style="list-style-type: none"> 当读取接收 FIFO 的接收数据时

12.12.2 发送中断

发送操作和读取路径数据流如图 12-14 所示。

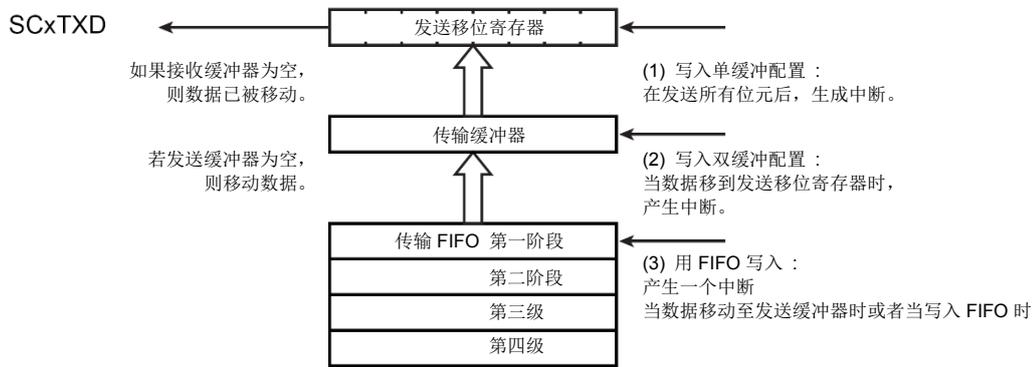


图 12-14 发送缓冲器/FIFO 配置图

12.12.2.1 单缓冲/双缓冲

根据下列传输模式和缓冲器配置，在下列时间产生发送中断。

表 12-9 在使用单缓冲器/双缓冲器时发送中断条件

配置	UART 模式	IO 接口模式
单缓冲器	恰在发送停止位前	恰在最后一个 SCxSCLK 引脚上升沿/下降沿后 (上升或者下降根据 SCxCR<SCLKS>的设置确定。)
双缓冲器	当数据从发送缓冲器移到发送移位寄存器时。 若发送移位寄存器为空，则不根据 SCxMOD1<TXE>产生发送中断，是因为写入发送缓冲器的数据已经从发送缓冲器移到发送移位寄存器。	

12.12.2.2 FIFO

在使用 FIFO 时，根据表 12-10 所述时序及 SCxTFC<TFIS>指定的条件，发生发送中断。

表 12-10 在使用 FIFO 时发送中断条件

SCxTFC<TFIS>	中断条件	中断产生时间
"0"	当 FIFO 填充深度(SCxTST<TLVL[2:0]>) = 产生发送中断的发送 FIFO 填充深度<TIL[1:0]>时	<ul style="list-style-type: none"> ■ 当发送数据从发送 FIFO 传输到发送缓冲器时 ■ 当发送数据写入发送 FIFO 时
"1"	当 FIFO 填充深度(SCxTST<TLVL[2:0]>) < 产生发送中断的发送 FIFO 填充深度<TIL[1:0]>时	<ul style="list-style-type: none"> ■ 当发送数据写入发送 FIFO 时

12.12.3 错误生成

12.12.3.1 UART 模式

错误	9 位	7 位 8 位 7 位 + 奇偶校验 8 位 + 奇偶校验
成帧错误 溢出错误	在停止位中心周围	
奇偶错误	-	在奇偶校验位中心周围

12.12.3.2 I/O 接口模式

溢出错误	恰在最后一个 SCxSCLK 引脚上升沿/下降沿后(上升或者下降根据 SCxCR<SCLKS>的设置确定。)
在运行错误	恰在下一 SCxSCLK 引脚上升沿/下降沿后。(上升或者下降根据 SCxCR<SCLKS>的设置确定。)

注：在时钟输出模式中溢出错误和欠载错误没有任何意义

12.13 DMA 请求

DMA 传输在中断请求时序启动。

当进行 DMA 传输时，请设置 SCxDMA 位。

本产品 DMA 请求所用通道见"产品信息"章节。

注：若通过发送或接收中断请求使用 DMA，则用 SCxMOD<SWRST>产生软件复位后，应启用 DMA 并设置发送和接收寄存器。

12.13.2 使用 FIFO

通过在 12.12.1.2 和 12.12.2.2 中的时序产生中断请求，DMA 传输启动。

在发送模式下，产生中断的填充深度，在开始发送前写入发送 FIFO 的数据最小数及 DMA 传输数据最大数如下所示。

表 12-11 产生中断的填充深度，在开始发送前写入发送 FIFO 的数据最小数及 DMA 传输数据最大数

SCxTFC <TIL[1:0]>	半双工		全双工	
	在开始发送前写入发送 FIFO 的数据最小数	DMA 传输数据最大数	在开始发送前写入发送 FIFO 的数据最小数	DMA 传输数据最大数
00	3	4 字节	3	2 字节
01	4	3 字节	4	1 字节
10	5	2 字节	3	2 字节
11	6	1 字节	4	1 字节

注：若发送移位寄存器为空，则不根据 SCxMOD1<TXE>产生发送中断，是因为写入发送缓冲器的数据已经从发送缓冲器移到发送移位寄存器。DMA 传输不得由该中断启动。DMA 传输详见 DMA 章节。

在接收模式下，产生中断的填充深度和 DMA 传输数据最大数如下所示。

表 12-12 产生中断的填充深度，在开始发送前写入发送 FIFO 的数据最小数及 DMA 传输数据最大数

SCxTFC <RIL[1:0]>	DMA 传输数据最大数(半双工)	DMA 传输数据最大数(全双工)
00	4 字节	2 字节
01	3 字节	1 字节
10	2 字节	2 字节
11	1 字节	1 字节

12.14 软件复位

将 SCxMOD2<SWRST[1:0]>写作"10"再后接"01", 就能产生软件复位。

结果是 SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFL><TXRUN>, SCxCR<OERR><PERR><FERR>被初始化。接收电路和发送电路为初始状态。

其它状态则被保持。

12.15 各模式下的操作

12.15.1 模式 0 (I/O 接口模式)

将 SCxMOD<SM[1:0]> 设为 "00", 可选择 I/O 接口模式。

模式 0 由两种模式组成, 即输出同步时钟(SCLK)的时钟输出模式和接收外部来源同步时钟(SCLK)的时钟输入模式。

在每种模式中中止 FIFO 的操作描述如下。FIFO 见前文述接收 FIFO 和发送 FIFO。

12.15.1.1 发送

(1) 时钟输出模式

若发送双缓冲器被禁止(SCxMOD2<WBUF> = "0")

每当 CPU 将数据写入发送缓冲器时, 数据从 SCxTXD 引脚输出, 时钟从 SCxSCLK 引脚输出。当所有数据被输出时, 生成中断 INTTXx。

若发送双缓冲器启用(SCxMOD2<WBUF> = "1")

当 CPU 将数据写入发送缓冲器, 而数据发送停止时, 或者当发送缓冲器(移位寄存器)的数据发送完成时, 数据从发送缓冲器移到发送移位寄存器。同时, SCxMOD2<TBEMP> 设为 "1", 并产生 INTTXx 中断。

若发送缓冲器无数据移到发送移位寄存器, 则不产生 INTTXx 中断, SCLK 输出停止。

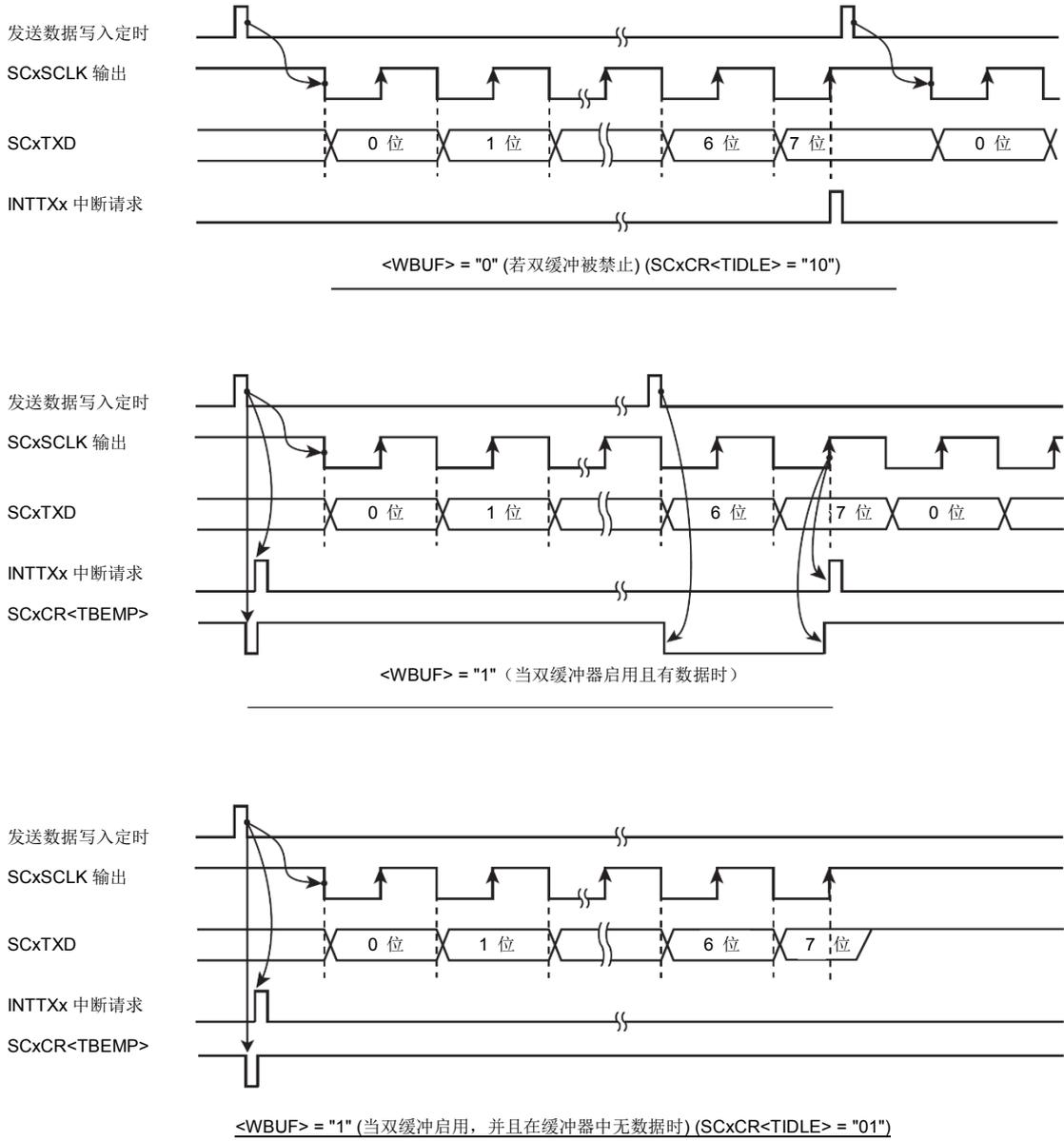


图 12-15 在 I/O 接口模式(时钟输出模式)下的发送操作

(2) 时钟输入模式

若双缓冲被禁止(SCxMOD2<WBUF> = "0")

若在数据写入发送缓冲器的条件下输入 SCLK，则从 SCxTXD 引脚输出 8 位数据。当输出所有数据时，产生中断 INTTXx。在图 12-16 所示时序点"A"前，必须写入下一发送数据。

若双缓冲器启用(SCxMOD2<WBUF> = "1")

当 CPU 在时钟输入激活前将数据写入发送缓冲器时，或者当发送移位寄存器的数据发送完成时，数据从发送缓冲器移到发送移位寄存器。同时，SCxMOD2<TBEMP>设为"1"，产生 INTTXx 中断。

若时钟输入激活，而发送缓冲器中无数据，则虽然内部位计数器启动，但仍会发生欠载错误，并且将 SCxCR<TXDEMP>指定的电平输出到 SCxTXD 引脚。

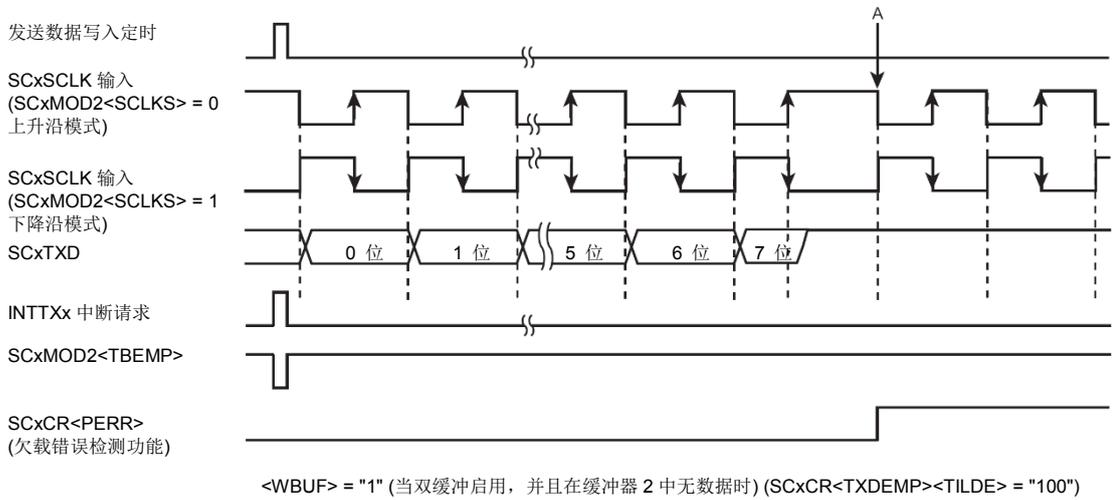
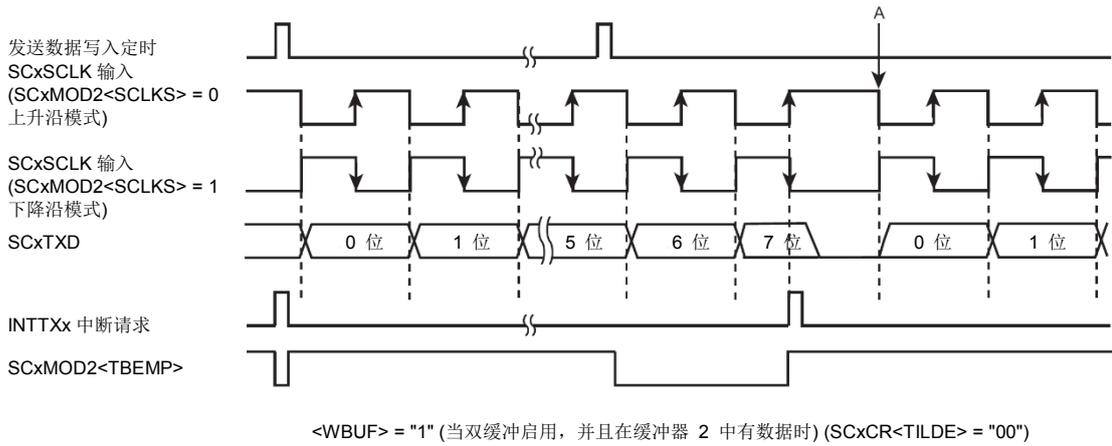
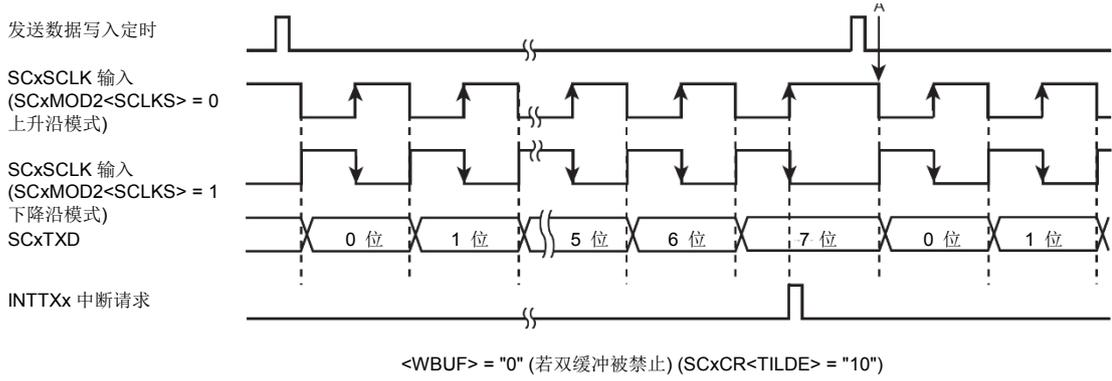


图 12-16 在 I/O 接口模式(时钟输入模式)下的发送操作

12.15.1.2 接收

(1) 时钟输出模式

将接收启用位 SCxMOD0<RXE>设为"1", 时钟输出开始。

若双缓冲器被禁止(SCxMOD2<WBUF> = "0")

每当 CPU 读取接收数据时, 就会从 SCxSCLK 引脚输出时钟, 并将下一数据存储在移位寄存器中。当接收到所有 8 位时, 产生 INTRXx 中断。

若双缓冲器启用(SCxMOD2<WBUF> = "1")

在移位寄存器中储存的数据移到接收缓冲器, 并且接收缓冲器能接收下一帧。数据从移位寄存器移到接收缓冲器, SCxMOD2<RBFL>设为"1", 产生 INTRXx 中断。

当接收缓冲器中有数据时, 若在下 8 位的接收完成前无法读取接收缓冲器的数据, 则不产生 INTRXx 中断, 时钟输出停止。在这种状态下, 读取接收缓冲器的数据可使移位寄存器中的数据移到接收缓冲器, 因此生成 INTRXx 中断, 数据接收恢复。

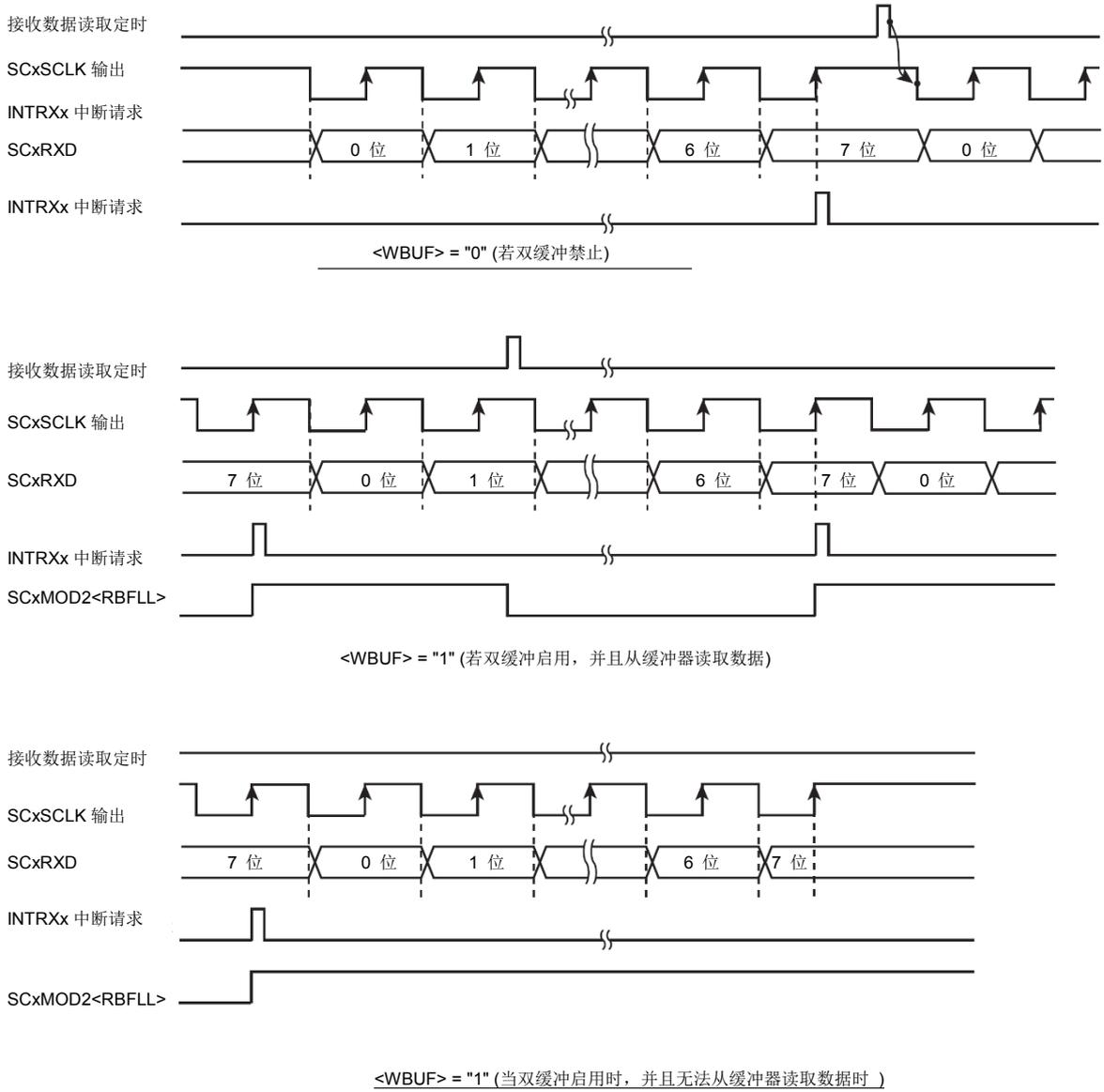


图 12-17 在 I/O 接口模式(时钟输出模式)下的接收操作

(2) 时钟输入模式

在时钟输入模式下，接收双缓冲始终启用，接收数据能从移位寄存器移到接收缓冲器，并且接收缓冲器能相继接收下一帧。

每当将接收数据移到接收缓冲器时都会产生 INTRXx 接收中断。

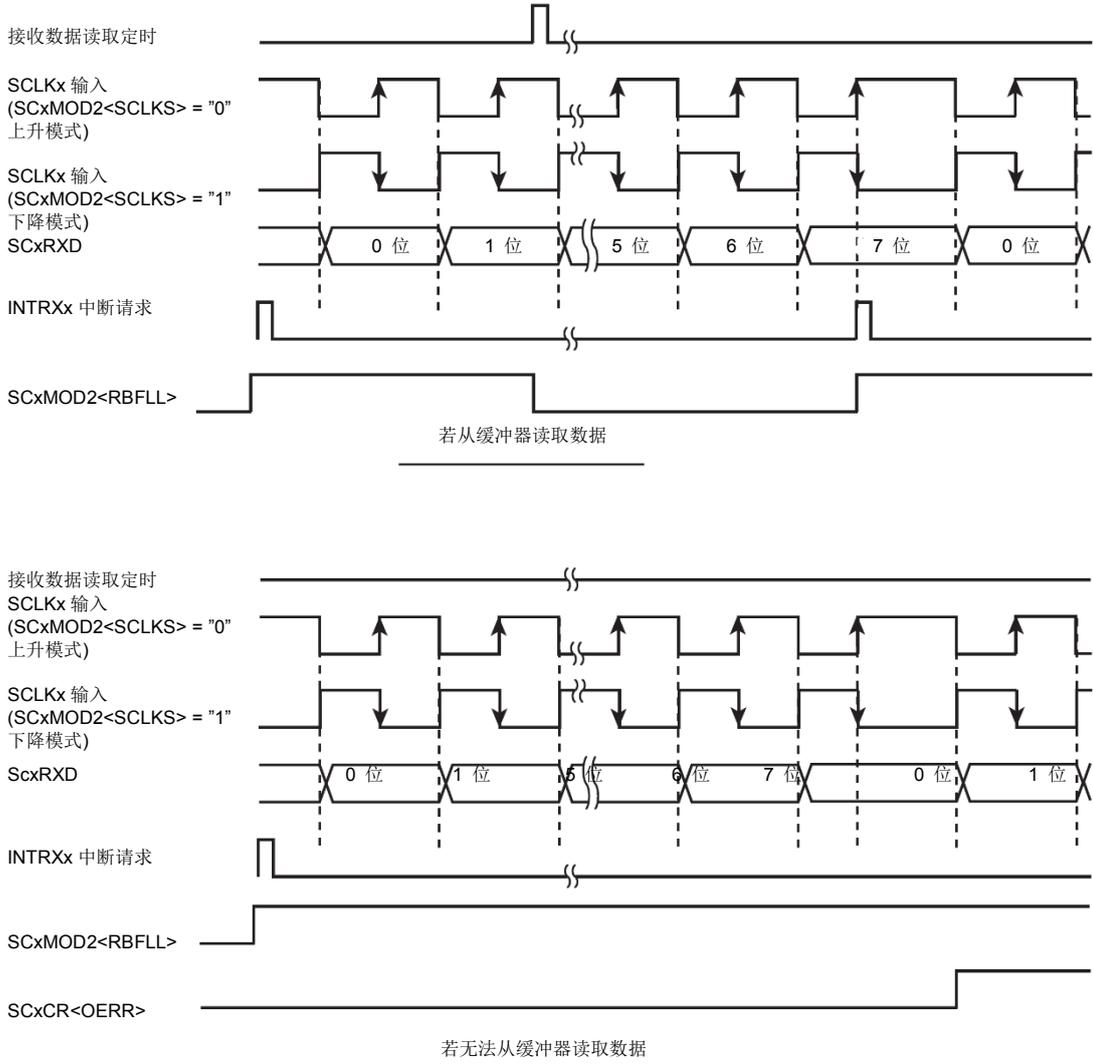


图 12-18 在 I/O 接口模式(时钟输入模式)下的接收操作

12.15.1.3 发送和接收(全双工)

(1) 时钟输出模式

若双缓冲器被禁止(SCxMOD2<WBUF> = "0")

当 CPU 将数据写入发送缓冲器时，输出时钟。

随后，数据移入接收缓冲器，产生 INTRX_x 接收中断。同时，当所有数据完成发送时，从 SCxTXD 引脚输出发送缓冲器中写入的数据，产生 INTTX_x 发送中断。然后，时钟输出停止。

当接收缓冲器的数据被读取，并且下一发送数据由 CPU 写入发送缓冲器时，就会开始下一轮数据发送和接收。能自由确定读取接收缓冲器和写入发送缓冲器的顺序。只有在满足这两个条件时，数据传输才能恢复。

若双缓冲器启用(SCxMOD2<WBUF> = "1")

当 CPU 将数据写入发送缓冲器时，输出时钟。

数据移入接收移位寄存器，移到接收缓冲器，并产生 INTRX_x 中断。当接收数据时，从 SCxTXD 引脚输出发送数据。当所有数据已发送时，产生 INTTX_x 中断，下一数据从发送缓冲器移到发送移位寄存器。

若发送缓冲器无数据要移到发送缓冲器(SCxMOD2<TBEMP> = "1")或者当接收缓冲器已满时(SCxMOD2<RBFL> = "1")，时钟输出停止。当满足接收数据被读取和发送数据被写入两种条件时，时钟输出恢复，开始下一轮数据发送和接收。

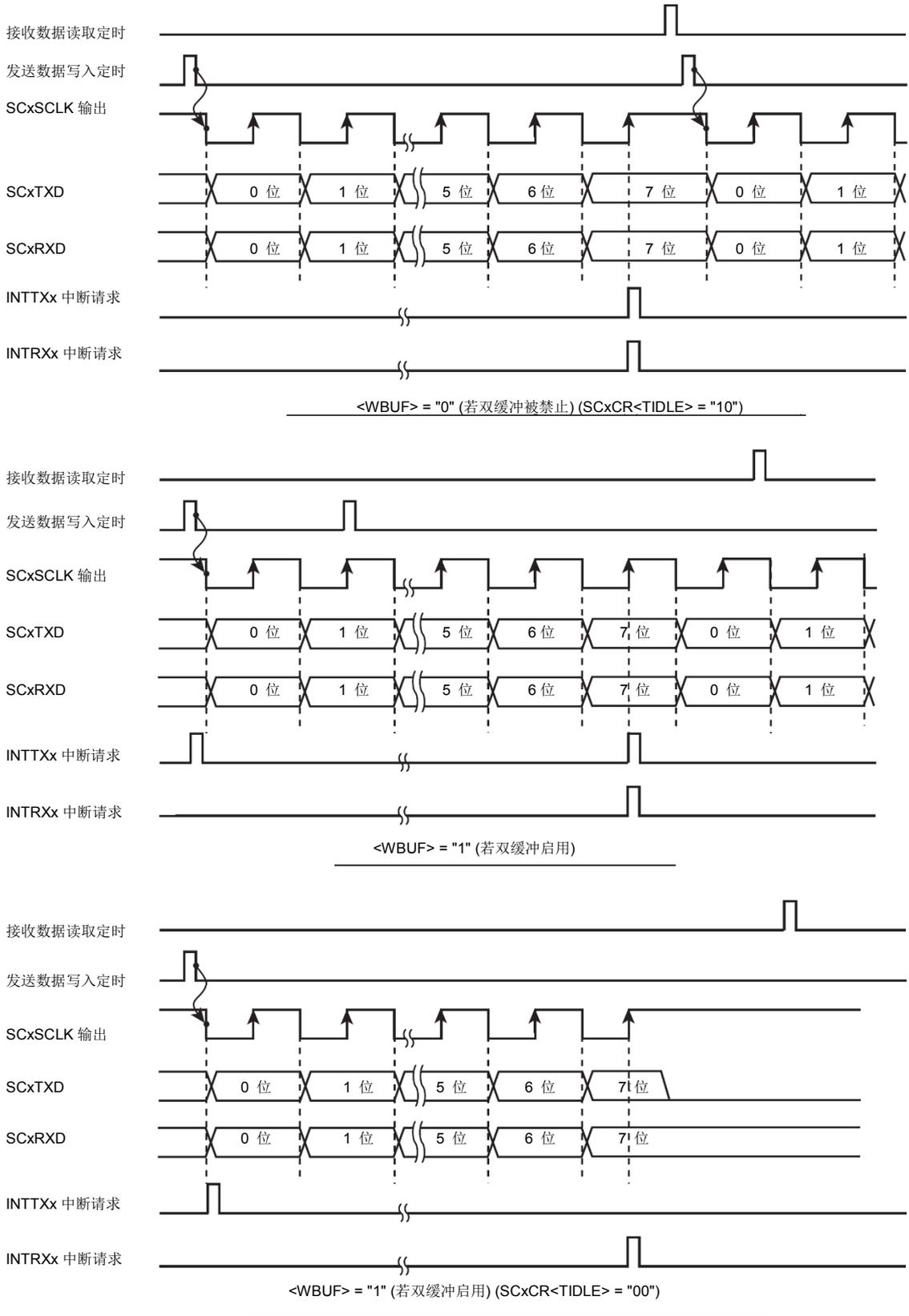


图 12-19 在 I/O 接口模式(SCLK 输出模式)下的发送/接收操作

(2) 时钟输入模式

当双缓冲中止时。(SCxMOD2<WBUF> = "0")

当接收数据时，不管 SCxMOD2 <WBUF>的设置如何，始终启用双缓冲。

当时钟输入激活时，从 SCxTXD 引脚输出发送缓冲器中写入的数据，数据移入接收缓冲器。在数据发送完成后，产生 INTTXx 中断。在数据接收完成后，当数据从移位寄存器移到接收缓冲器时，产生 INTRXx 中断。

注意发送数据必须在下一数据的时钟输入前写入发送缓冲器(数据必须在图 12-20 中的 A 点前写入)。在下一数据的接收完成前，必须读取数据。

当启用双缓冲时(SCxMOD2<WBUF> = "1")

在完成从发送移位寄存器的数据发送后，当发送缓冲器数据移到发送移位寄存器时，产生中断 INTTXx。同时，接收数据移入移位寄存器，并移到接收缓冲器，产生 INTRXx 中断。

注意发送数据必须在下一数据的时钟输入前写入发送缓冲器(数据必须在图 12-20 中的 A 点前写入)。在下一数据的接收完成前，必须读取数据。

在下一数据的时钟输入后，发送移位寄存器的发送开始(在该寄存器中，数据已从发送缓冲器移动)，同时接收数据移入接收移位寄存器。

当接收数据最后一位时，若接收缓冲器中的数据未被读取，则发生溢出错误。

当输入下一数据的时钟时，若无任何数据写入发送缓冲器，则发生欠载错误。SCxCR<TXDEMP>指定的电平被输出到 SCxTXD 引脚。

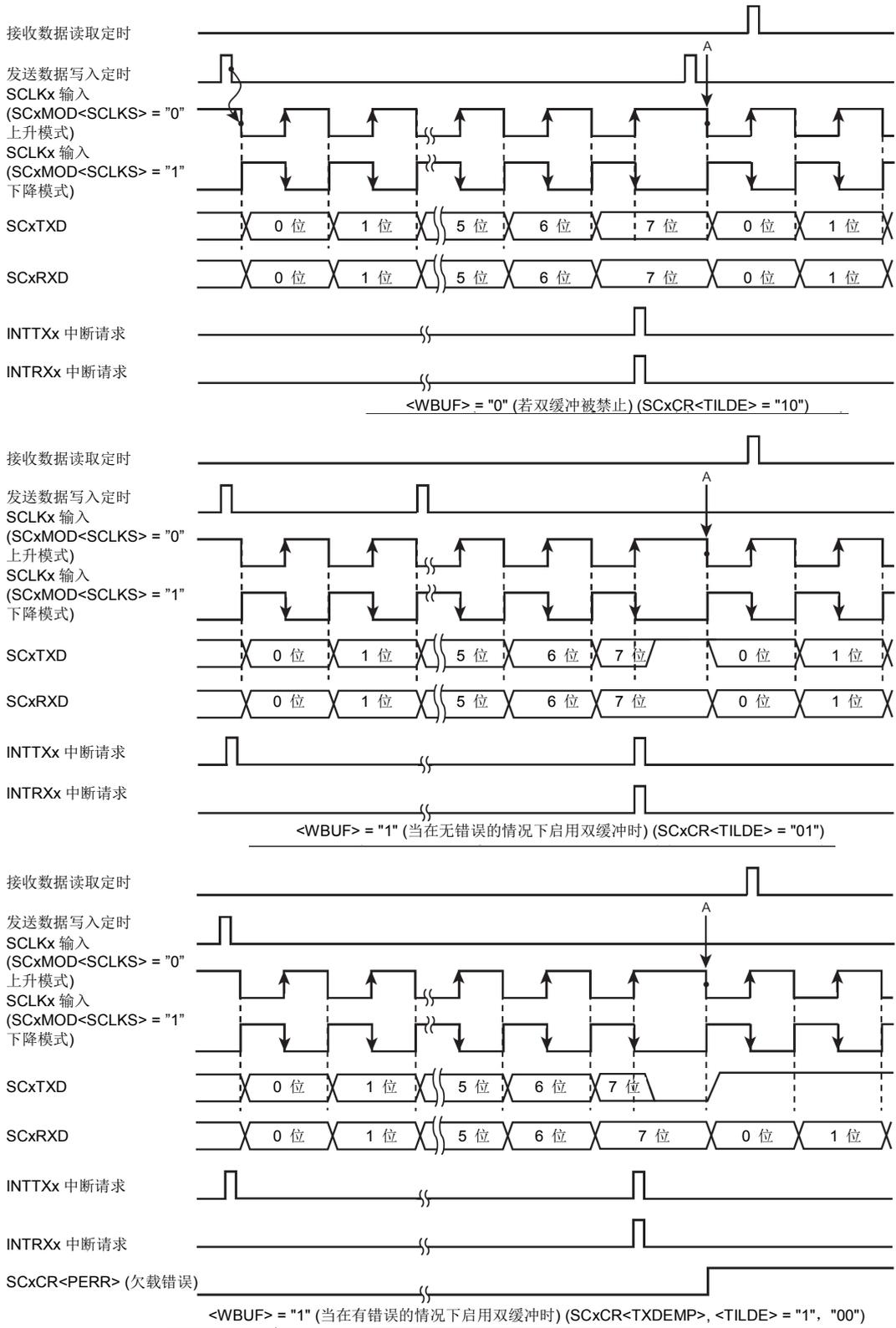


图 12-20 在 I/O 接口模式(时钟输入模式)下的发送/接收操作

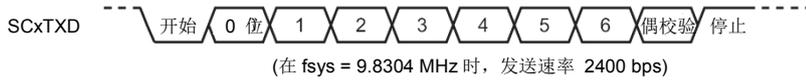
12.15.2 模式 1 (7 位 UART 模式)

将 SCxMOD<SM[1:0]>设为"01", 就能选择 7 位 UART 模式。

在该模式下, 奇偶校验位可加到发送数据流中; SCxCR<PE>控制奇偶校验的启用/禁止设置。

当<PE>设为"1"(启用)时, 可以用 SCxCR<EVEN>选择偶校验或奇校验。用 SCxMOD2<SBLEN>指定停止位长度。

以下列数据格式发送时, 控制寄存器的设置如下表所示。



计时条件	系统时钟:	高速(fc)
	高速时钟齿轮:	x 1 (fc)
	预分频时钟:	fperiph/ 2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	← x:	0	-	0	0	1	0	1		设置 7 位 UART 模式
SCxCR	← x:	1	1	x	x	x	0	0		偶校验启用
SCxBRCR	←	0	0	1	0	0	1	0	0	设置 2400 bps
SCxBUF	←	*	.*	.*	.*	.*	.*	.*	.*	设置发送数据

x:忽略 -:无变化

12.15.3 模式 2 (8 位 UART 模式)

将 SCxMOD0<SM[1:0]设为"10", 就能选择 8 位 UART 模式。在该模式下, 可增加奇偶校验位; 用 SCxCR<PE>控制奇偶校验的启用/禁用。若<PE> = "1" (启用), 则能用 SCxCR<EVEN>选择偶校验或奇校验。

按以下格式接收数据的控制寄存器设置如下:



计时条件	系统时钟:	高速(fc)
	高速时钟齿轮:	x 1 (fc)
	预分频时钟:	fperiph/ 2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	设置 8 位 UART 模式
SCxCR	←	x	0	1	x	x	x	0	0	奇校验启用
SCxBRCR	←	0	0	0	1	0	1	0	0	设置 9600 bps
SCxMOD0	←	-	-	1	-	-	-	-	-	接收启用

x:忽略 - :无变化

12.15.4 模式 3 (9 位 UART 模式)

将 SCxMOD0<SM[1:0]>设为"11", 就能选择 9 位 UART 模式。在该模式下, 奇偶校验位必须被禁止 (SCxCR<PE> = "0")。

为了发送数据, 最高有效位(第 9 位)写入 SCxMOD0<TB8>。为了接收数据, 数据存储在 SCxCR<RB8>中。

当数据写入缓冲器或者从缓冲器中读取时, 在写入 SCxBUF 或者从 SCxBUF 中读取前, 必须首先写入或读取最高有效位。

可用 SCxMOD2<SBLEN>指定停止位长度。

12.15.4.1 唤醒功能

在 9 位 UART 模式下, 将唤醒功能控制位 SCxMOD0<WU>设为"1", 从机控制器就能在唤醒模式下运行。

在这种情况下, 只有在 SCxCR<RB8>设为"1"时, 才会产生中断 INTRXx。

注: 从机控制器的 SCxTXD 引脚必须用 PxOD 寄存器设为开漏输出模式。

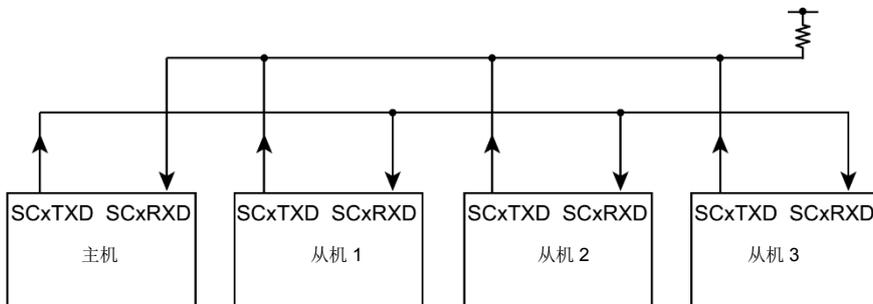
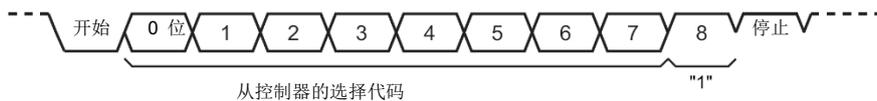


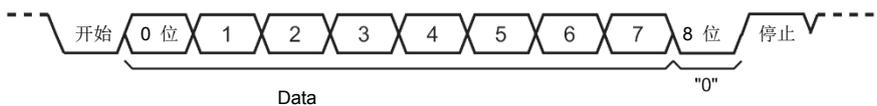
图 12-21 使用唤醒功能的串行链路

12.15.4.2 协议

1. 选择各主从控制器的 9 位 UART 模式。
2. 将从机控制器的 SCxMOD<WU>设为"1", 使其准备接收数据。
3. 主机控制器发送包括从机控制器选择代码(8 位)在内的单帧数据。在这种情况下, 最重要的位元(位 8) <TB8> 必须设为"1"。



4. 各从机控制器接收上述数据帧; 若接收的代码与控制器自己的选择代码匹配, 则它将<WU>清"0"。
5. 主控制器将数据发送到指定的从机控制器(SCxMOD<WU>清"0"的控制器)。在这种情况下, 最重要的位元(位 8) <TB8> 必须设为"0"。



6. 因为最高有效位(位 8) <TB8> 设为"0", 所以<WU>设为"1"的从机控制器忽视接收数据, 因此不产生中断(INTRXx)。此外, <WU>设为"0"的从机控制器能将数据发送到主机控制器, 通知数据已成功接收。

13. I2C 总线接口

TMPM037FWUG 包含符合常见 I2C 总线标准(飞利浦规范)的 I2C 总线接口。

主要特性如下。

可在主机和从机之间选择。

可在发送和接收之间选择。

支持多主机(仲裁, 时钟同步识别)。

波特率(支持标准模式和快速模式)。

仅支持 7 位寻址格式。

支持 1~8 位传输数据大小。

提供一次传输(发送或接收)完整中断(电平敏感)。

启用或禁用中断。

该模块也支持东芝专有数据格式"自由数据格式"。

表 13-1 I2C 总线标准规范

I2C 总线特性	I2C 标准	TMPM037FWUG
STANDARD 模式(高达 100 KHz)	要求	支持
FAST 模式(高达 400 KHz)	要求	支持
Hs(高速)模式(高达 3.4 Mbps)	要求	不支持
7 位寻址	要求	支持
10 位寻址	要求	不支持
START 字节	要求	支持
噪声消除器	要求	支持(数字)
斜度控制	要求	不支持
断电时 I/O	要求	支持
施密特(VIL/VHL)	$VDD \times 0.3 / VDD \times 0.7$	不支持
在 $VOL = 0.4 V$, $VDD > 2V$ 时输出电流	3mA	不支持

13.1 配置

配置如图 13-1 所示。

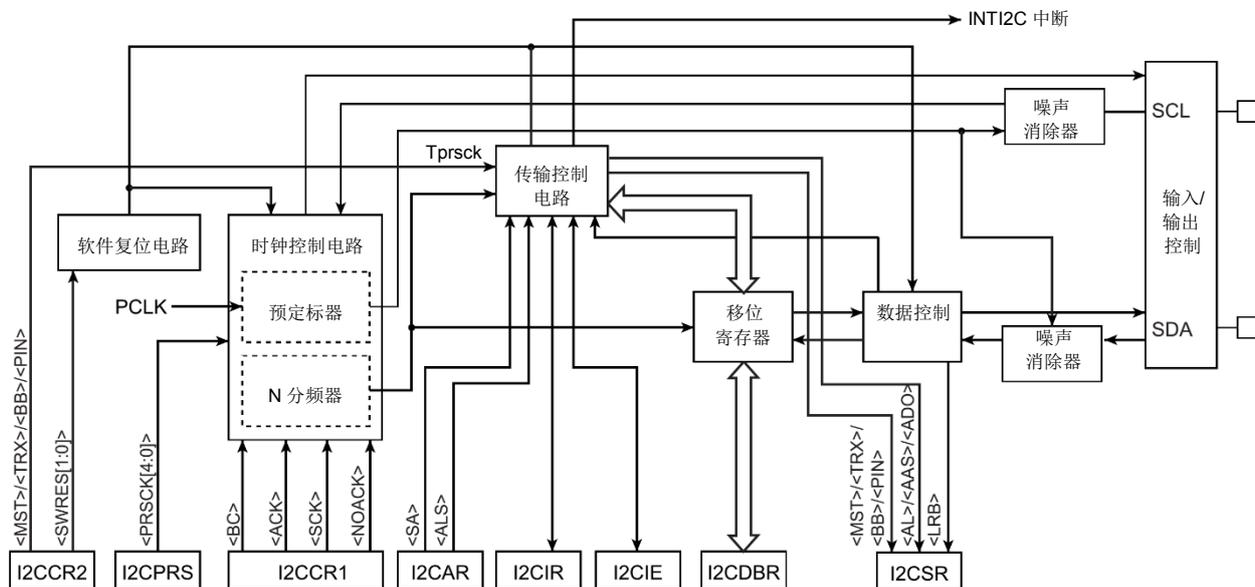


图 13-1 I2C 总线块

13.2 I2C 总线模式

I2C 总线经 SDA 和 SCL 引脚连至装置，能与多台设备通信。

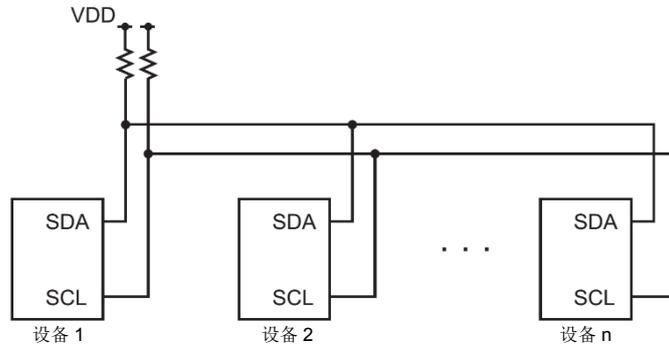


图 13-2

本模块作为 I2C 总线上主机或从机设备运行。主机设备驱动总线串行时钟线(SCL)，发送 8 位地址，并发送或接收 1~8 位数据。

与总线串行时钟同步，从机设备发送 8 位地址，并发送或接收 1~8 位串行数据。

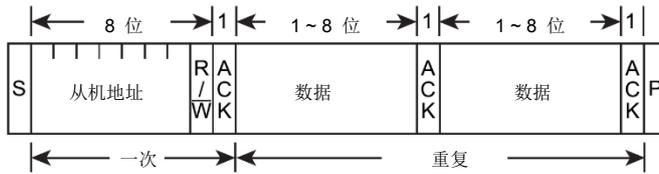
不管设备是主机还是从机，作为接收器操作的设备在接收串行数据后能输出确认信号，作为发送器操作的设备能接收该确认信号。主机设备能输出确认信号的时钟。

在同一总线上存在多主机的多主机模式下，支持串行时钟同步和仲裁丢失，以保持串行数据一致性。

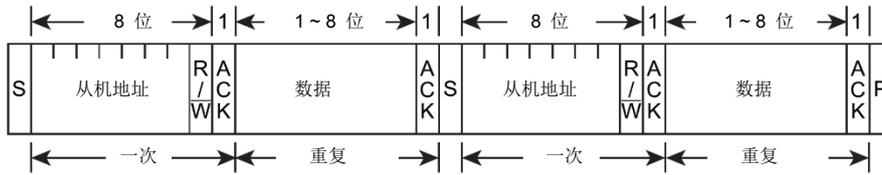
13.2.1 I2C 总线模式数据格式

I2C 总线模式使用的数据格式如图 13-3 所示。

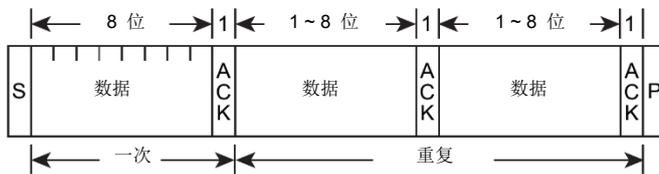
(a) 寻址格式



(b) 寻址格式 (重复启动条件)



(c) 自由数据格式 (主传输器-从接收器)



注)S: 起始条件
 R/W: 方向位
 ACK: 确认位
 P: 停止条件

图 13-3 I2C 总线模式数据格式

13.3 寄存器

13.3.1 各通道的寄存器

I2C 总线接口寄存器和地址如下表所示。
基址见"内存地址"章节中的"外设功能地址列表"。

寄存器名称		地址(基+)
控制寄存器 1	I2CxCR1	0x0000
数据缓冲寄存器	I2CxDBR	0x0004
I2C 总线地址寄存器	I2CxAR	0x0008
控制寄存器 2	I2CxCR2 (写入)	0x000C
状态寄存器	I2CxSR (读取)	
预分频器时钟设置寄存器	I2CxPRS	0x0010
中断启用寄存器	I2CxIE	0x0014
中断寄存器	I2CxIR	0x0018

注：这些寄存器能通过唯一的单词存取进行读取或写入。

13.3.2 I2CxCR1(控制寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	BC			ACK	NOACK	SCK		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能				
31-8	-	R	读作"0"。				
7-5	BC[2:0]	R/W	选择每次传输时的位数 (注 1)				
				在 <ACK> = 0 时	在 <ACK> = 1 时		
			<BC>	时钟周期的数目	数据长度	时钟周期的数目	数据长度
			000	8	8	9	8
			001	1	1	2	1
			010	2	2	3	2
			011	3	3	4	3
			100	4	4	5	4
			101	5	5	6	5
110	6	6	7	6			
111	7	7	8	7			



位	比特符号	型号	功能																
4	ACK:	R/W	主模式 0: 未生成确认时钟脉冲。 1: 生成了确认时钟脉冲。 从模式 0: 未计数确认时钟脉冲。 1: 已计数确认时钟脉冲。																
3	NOACK	R/W	从机地址匹配检测和一般调用检测。 0: 从机地址匹配检测或一般调用检测启用。 1: 从机地址匹配检测或一般调用检测被禁用。 当 I2CxAR<ALS> = '1' 时, 该位无意义。																
2-0	SCK[2:0]	R/W	选择内部 SCL 输出时钟频率(注 2)。 <table border="1" data-bbox="560 707 1048 887"> <tbody> <tr> <td>000</td> <td>n = 0</td> <td>000</td> <td>n = 4</td> </tr> <tr> <td>001</td> <td>n = 1</td> <td>001</td> <td>n = 5</td> </tr> <tr> <td>010</td> <td>n = 2</td> <td>010</td> <td>n = 6</td> </tr> <tr> <td>011</td> <td>n = 3</td> <td>011</td> <td>n = 7</td> </tr> </tbody> </table>	000	n = 0	000	n = 4	001	n = 1	001	n = 5	010	n = 2	010	n = 6	011	n = 3	011	n = 7
000	n = 0	000	n = 4																
001	n = 1	001	n = 5																
010	n = 2	010	n = 6																
011	n = 3	011	n = 7																

注 1: 在起始条件产生前或者在停止条件产生后或者在地址或数据传输中断发生瞬间和内部中断释放瞬间之间, 必须写入该寄存器。不要在地址或数据传输时写入该寄存器。

注 2: SCL 线时钟频率详见"13.4.1 串行时钟"。

注 3: 所选频率的初始值为<SCK [2:0]>= 000, 与读取的初始值无关。

13.3.3 I2CxDBR(串行总线接口数据缓冲寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作"0"。
7-0	DB[7:0]	R (接收)	接收数据
		W (传输)	传输数据
			该传输数据必须从 MSB (位 7) 写入寄存器。接收数据存储在 LSB 中。

当主机需要发送从机地址时，传输目标地址写入 I2CxDBR<DB [7:1]>，在 I2CxDBR<DB[0]>中传输方向指定如下：

<DB[0]> = 0：主机(传输) → 从机/接收

<DB[0]> = 1：主机(接收) ← 从机/传输

当 I2CxDBR 寄存器中的所有位写作"0"时，能在总线上发出一般调用。

在传输和接收模式下，在当前传输后，读写 I2CxDBR 寄存器就会释放内部中断，启动下一次传输。

I2CxDBR 作为传输/接收缓冲器，在发送模式下用作专用传输缓冲器，在接收模式下用作专用接收缓冲器。该寄存器应在逐次传输的基础上进行存取。

注：该寄存器仅通过软件复位进行初始化。软件复位不能初始化最后的数据，会保持它们。

13.3.4 I2CxAR(I2C 总线地址寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作"0"。
7-1	SA[6:0]	R/W	当 I2C 用作从机设备时，设置从机地址。
0	ALS	R/W	指定地址识别模式。 0: 识别其从机地址。 1: 不对其从地址予以确认（自由数据格式）。

注 1: 除了在您使用自由数据格式时，请将 I2C 总线地址寄存器 I2CxAR 第 0 位<ALS>设为"0"。当设为"1"时，它作为自由数据格式运行。选择拟被固定为发送功能的主机。选择从机，就能固定为接收。

注 2: 在从机模式下，不要将 I2CxAR 设为"0x00"。(若 I2CxAR 设为"0x00"，则识别在从机模式下，从机地址与接收的 I2C 标准的 START 字节("0x01")匹配。)

13.3.5 I2CxCR2(控制寄存器 2)

通过读取本寄存器，本寄存器用作 I2CxSR 寄存器。

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	MST	TRX	BB	PIN	I2CM	-	SWRES	
复位后	0	0	0	1	0	0	0	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	MST	W	选择主机/从机 0.从机模式 1.主机模式
6	TRX	W	选择发送/接收 0.接收 1.传输
5	BB	W	启动/停止条件生成 0.停止条件已生成 1.启动条件已生成
4	PIN	W	清除 INTI2Cx 中断请求 0. - 1.清除中断请求
3	I2CM	W	I2C 操作控制 0.禁用 1.启用
2	-	R	读作"0"。
1-0	SWRES[1:0]	W	软件重置产生 写入"10"再写入"01"以生成重置。 详见"13.4.11 软件复位"。

注 1: 通信会话时务必不要改变模式。保证操作模式切换到端口模式前，总线是自由的。保证操作模式从端口模式切换到 I2C 总线模式前，端口处于"高"电平。

注 2: 当正进行传输操作时，无法将 I2CxCR2<I2CM>清 0 来禁用 I2C 操作。在清除该位前，保证通过读取状态寄存器而使传输操作完全停止。

13.3.6 I2CxSR(状态寄存器)

通过向本寄存器写入数据，本寄存器用作 I2CxCR2。

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
复位后	0	0	0	1	0	0	0	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	MST	R	主机/从机选择监控程序 0: 从机模式 1: 主机模式
6	TRX	R	发送/接收选择监控程序 0: 接收 1: 传输
5	BB	R	I2C 总线状态监控程序 0: 空闲 1: 忙
4	PIN	R	INTI2Cx 中断请求监控 0: 产生中断请求 1: 中断请求已被清除
3	AL	R	仲裁丢失检测 0: - 1: 已检测
2	AAS	R	从机地址匹配检测 0: - 1: 已检测 (也可在检测到全呼叫时设置该位元。)
1	AD0	R	一般调用检测 0: - 1: 已检测
0	LRB	R	最后接收位监视器 0: 最后接收位"0" 1: 最后接收的位"1"

13.3.7 I2CxPRS(预分频器时钟设置寄存器)

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

位	比特符号	类型	功能
31-5	-	R	读作"0"。
4-0	PRSCK	R/W	用于产生串行时钟的预分频器时钟频率 00000: P = 32 分频 00001: P = 1 分频 ----- 11111: P = 31 分频

注：见"13.3.2 I2CxCR1(控制寄存器 1)", "13.4.1 串行时钟"。

13.3.8 I2CxIE(中断启用寄存器)

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

位	比特符号	类型	功能
31-1	-	R	读作"0"。
0	IE	R/W	I2C 中断启用或禁用设置。 0: 禁用 1: 启用

13.3.9 I2CxIR(中断寄存器)

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

位	比特符号	类型	功能
31-1	-	R	读作"0"。
0	ISIC	R	I2C 中断状态。 0: 不中断 1: 产生中断。 该位指示用 I2CxIE<IE> 屏蔽前的 I2C 中断状态
		W	清除 I2C 中断。 0: 无效 1: 清除 I2C 中断。 将"1"写入该位, 就会清除 I2C 中断输出(INTI2Cx)。写入"0"无效。

13.4 在 I2C 总线模式下的控制

13.4.1 串行时钟

13.4.1.1 时钟源

I2CxCR1<SCK[2:0]>用于设置主机模式下输出的串行时钟的高低周期。

<SCK[2:0]>	$t_{\text{HIGH}} (i / T_{\text{prsc}})$	$t_{\text{LOW}} (j / T_{\text{prsc}})$
	i	j
000	8	12
001	10	14
010	14	18
011	22	26
100	38	42
101	70	74
110	134	138
111	262	266

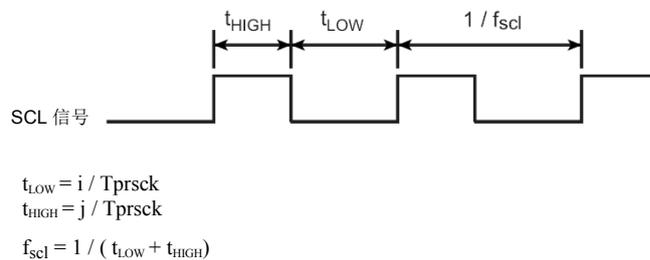


图 13-4 时钟源

注：根据总线负载电容和上拉寄存器的组合，若上升沿变钝， t_{HIGH} 周期随规定值不同。若使用的时钟同步功能是同步多个时钟的时钟，实际时钟周期随指定的设置不同。

在主机模式下，起始条件产生时的保持时间和停止条件产生时的建立时间被定义为 $t_{\text{HIGH}}[\text{S}]$ 。

当在从机模式下 I2CxCR2<PIN>设为"1"时，SCLx 释放时间被定义为 $t_{\text{LOW}}[\text{S}]$ 。

在主机和从机两种模式下，不管 I2CxCR1<SCK>的设置如何，对于外部输入的串行时钟，高电平周期必须大于等于 $4 / T_{\text{prsc}}[\text{s}]$ ，低电平周期必须大于等于 $5 / T_{\text{prsc}}[\text{s}]$ 。

主机输出的串行时钟速率通过 I2CxCR1<SCK[2:0]>和 I2CxPRS<PRSK[4:0]>设置。按照 I2CxPRS<PRSK[4:0]>分频的预分频器时钟用作产生串行时钟的基准时钟。预分频器时钟按照 I2CxCR1<SCK[2:0]>被进一步分频，并用作串行时钟。预分频器时钟默认设置为 1 分频(= f_{sys})。

<串行传输速率>

根据操作频率(fsys), 串行时钟速率(fSCL)由预分频器设置值"p" (I2CxPRS<PRCK[4:0]>, p = 1 ~ 32) 和串行时钟设置值"n" (I2xCRI<SCK[2:0]>, n = 0 ~ 7)确定如下:

$$\text{串行时钟速率: } f_{\text{scl}} \text{ (kHz)} = \frac{f_{\text{sys}} \text{ (MHz)}}{p \times (2^{n+2} + 16)} \times 1000$$

p: 预分频器设置值 I2CxPRS <PRCK[4:0]>, 1 ~ 32
n: 串行时钟设置值 I2xCRI <SCK[2:0]>, 0 ~ 7

预分频器设置值"p" (I2CxPRS<PRCK[4:0]>)允许范围随操作频率(fsys)不同, 必须满足下列条件。

50ns	<	预分频器时钟宽度:Tprsc (ns)	≤	150 ns
------	---	---------------------	---	--------

注: 在主机和从机两种模式下, 禁用将预分频器时钟宽度设置在该范围之外。

由于时钟同步功能, 串行时钟速率可以不恒定。

n: <SCK[2:0]>			p: <PRCK[4:0]>		
			00001 (1 分频)	01101 (13 分频)	00000 (32 分频)
			与 fsys 之比		
0	0	0	20	260	640
0	0	1	24	312	768
0	1	0	32	416	1024
0	1	1	48	624	1536
1	0	0	80	1040	2560
1	0	1	144	1872	4608
1	1	0	272	3536	8704
1	1	1	528	6864	16896

注: 在起始条件产生前或者在停止条件产生后, 必须写入这些位。在运输时写入这些位会造成意外运行。

<预分频器时钟宽度(=噪声消除宽度)>

根据操作频率(fsys), 预分频器时钟宽度(Tprsc)(=噪声消除宽度)由预分频器设置值"p" (I2CxPRS<PRCK[4:0]>, p = 1 ~ 32)确定如下:

$$\text{预分频器时钟宽度: } T_{\text{prsc}} \text{ (ns)} \text{ (=噪声消除宽度)} = \frac{1}{f_{\text{sys}} \text{ (MHz)}} \times 1000 \times p$$

13.4.1.2 时钟同步

因受其引脚结构影响，需采用配线 AND 连接驱动 I2C 总线。将其时钟线拖至“低”电平的首个主机，超越在其时钟线上产生“高”电平时其它主机。产生“高”电平的主机必须就此进行检测并应答。

甚至当总线上存在多主机时，为了保证正常传输，I2C 具有时钟同步操作。

例如，以下给出了带有两个主机的总线的时钟同步程序。

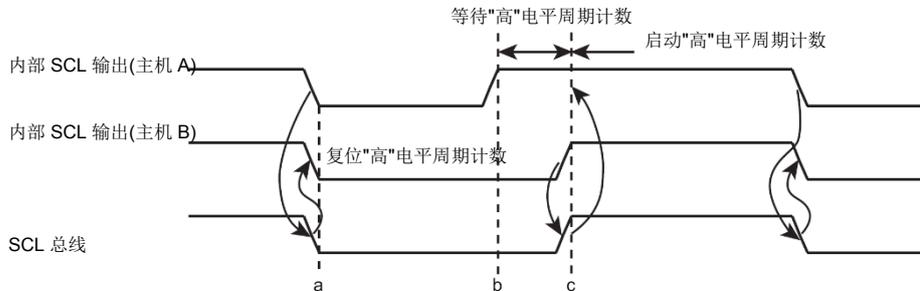


图 13-5 时钟同步示例

主机 A 在点 a 时将其内部 SCL 输出拖至“低”电平，使得 SCL 总线也变为“低”。主机 B 检测该跳变，使其“高”电平周期计数器复位，并将其内部 SCL 输出电平拉至“低”电平。

主机 A 在 b 点完成其“低”电平周期的计数，并将其内部 SCL 输出拉至“高”电平。然而，主机 B 仍使 SCL 总线保持在“低”电平，主机 A 停止其“高”电平周期的计数。在主机 A 检测到主机 B 在 c 点将其内部 SCL 输出拉至“高”电平并将 SCL 总线拉至“高”电平后，它开始其“高”电平周期的计数。

在该主机完成“高”电平周期的计数之后，主机将拖至“低”，SCL 总线随之变为“低”。

通过这种方法，即可依据已被连接到该总线的主机中“高”电平周期最短的主机，以及“低”电平周期最长的主机，确定该总线的时钟。

13.4.2 从机设备地址匹配检测或一般调用检测的选择

对于从机设备，I2CxCR1<NOACK>用于在从机模式下启用或禁用从机地址匹配检测和一般调用检测。当 I2CxCR1<NOACK> = 0 时，它用于启用从机地址匹配检测和一般调用检测。

若接收的地址与 I2CxAR 指定的其从机地址匹配或者等于一般调用地址，则在第 9 个时钟，I2C 将 SDA 线拉至“低”电平，并输出应答信号。

当 I2CxCR1<NOACK> = 1 时，它用于禁用从机地址匹配检测和一般调用检测。

若接收地址与 I2CxAR 指定的其从机地址匹配或者等于一般调用地址，则 I2C 将 SDA 线释放至“高”电平，并输出非应答信号。

从机设备忽略主机发送的从机地址和一般调用，并返回非应答。不产生 INTI2Cx 中断请求。
在主机模式下，I2CxCR1<NOACK>被忽略，对操作无影响。

注：若在从机模式下，I2CxCR1<NOACK>在数据传输时清"0"，则 I2CxCR1<NOACK>保持"1"，并且针对传输的数据，返回应答信号。

13.4.3 设置确认模式

将 I2CxCR1<ACK>设为"1"，就能选择应答模式。作为主机操作时，I2C 增加一个应答信号时钟。在从机模式下，可对应答信号的时钟进行计数。在发送器模式下，I2C 会在时钟周期期间释放 SDAx 引脚，以接收来自接收器的应答信号。在接收器模式下，I2C 在时钟周期期间将 SDAx 引脚拉至"低"电平，并产生应答信号。此外在从机模式下，若接收到一般调用地址，则 I2C 在时钟周期期间将 SDAx 引脚拉至"低"电平，并产生应答信号。

然而，根据第二个字节的内容，第二个字节必须由软件控制，以产生 ACK 信号。

通过将 <ACK>设置为"0"，即可触发非应答模式。作为主机操作时，I2C 不产生应答信号时钟。在从机模式下，会计数应答信号的时钟。

13.4.4 设置每次传输的位数

I2CxCR1<BC[2:0]>指定下一个将要发送或接收的数据的位数。

在起始条件下，<BC[2:0]>设为"000"，由此造成从机地址和方向位在 8 位信息包中传输。在其他时间，<BC [2:0]>保持先前编程的值。

注：必须将 I2CxCR1<ACK>设为"1"，发送/接收从机地址。若 I2CxCR1<ACK>被清除，则不能正确地进行从机地址匹配检测和方向位检测。

13.4.5 从机设备地址和地址识别模式

将 I2CxAR<ALS>设为"0"并在 I2CxAR<SA[6:0]>中设置从机地址，就能设置寻址格式，然后 I2C 识别主机设备发送的从机地址，并接收寻址格式的数据。

若<ALS>设为"1"，则 I2C 不识别从机地址，并接收自由数据格式的数据。在自由数据格式的情况下，不识别从机地址和方向位；在产生起始条件后，它们立即被识别为数据。

13.4.6 将 I2C 设置为主机设备或从机设备

将 I2CxCR2<MST>设为"1"，就能将 I2C 配置为主机设备。

将<MST>设为"0"，就能将 I2C 配置为从机设备。在其检测到总线上的该停止条件，或仲裁丢失时，<MST>即被硬件清除为"0"。

13.4.7 将 I2C 配置为发送器或接收器

将 I2CxCR2<TRX>设为"1"，就能将 I2C 配置为发送器。将<TRX>设为"0"，就能将 I2C 配置为接收器。

在从属模式时：

- 在数据被以寻址格式发送时。
- 当接收的从机地址与 I2CxCAR 的指定值匹配。
- 当接收到一般调用地址（在起始条件后的 8 位全为零。）

当该方向位（ $\overline{R/W}$ ）数值为“1”时，<TRX>通过硬件设置至“1”。如果该位为"0"，则<TRX>会被设置为"0"。

作为主机设备，I2C 接收从机设备的应答。若发送方向位"1"，<TRX>由硬件设为"0"。如果该方向位为"0"，则<TRX>会变为"1"。若 I2C 未接收到应答，<TRX>保持先前的值。

在其检测到总线上的该停止条件，或仲裁丢失时，<TRX>即被硬件清除为"0"。

注：当 I2CxCR1<NOACK> = 1 时，从机地址检测和一般调用检测被禁用，因此 I2CxSR<TRX>保持不变。

表 13-2 I2CxSR<TRX> 操作

模式	方向位	状态变化条件	在状态变化后改变的<TRX>
从模式	0	接收的从机地址 = I2CxAR<SA>	0
	1		1
主模式	0	接收到 ACK	1
	1		0

当以自由数据格式使用 I2C 时，不会识别从机地址和方向位，紧跟在起始条件后的位被处理为数据。因此，<TRX>不被硬件改变。

13.4.8 生成启动和停止条件

当 I2CxSR<BB>为"0"时，将"1"写入 I2CxCR2<MST, TRX, BB, PIN>就会造成 I2C 启动产生起始条件的顺序，并输出在数据缓冲器寄存器中可能写入的从机地址和方向位。必须将<ACK>预先设置为"1"。

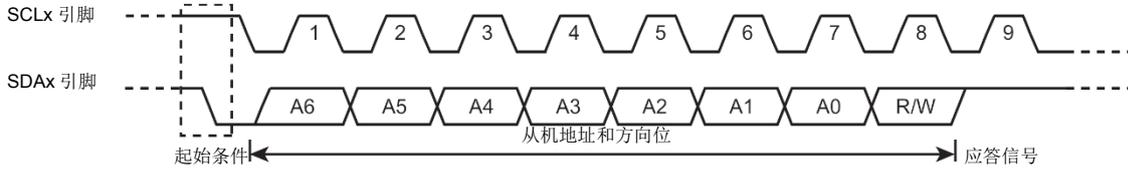


图 13-6 产生起始条件及从机地址

当<BB>为"1"时，将"1"写入<MST, TRX, PIN>及将"0"写入<BB>就会造成 I2C 启动在总线上产生停止条件的顺序。在总线上出现停止条件前，不应改变<MST, TRX, BB, PIN>的内容。

如果 SCL 总线在停止条件生成时被其它设备拖至"低"，则该停止条件会在 SCL 线路被解除时生成。

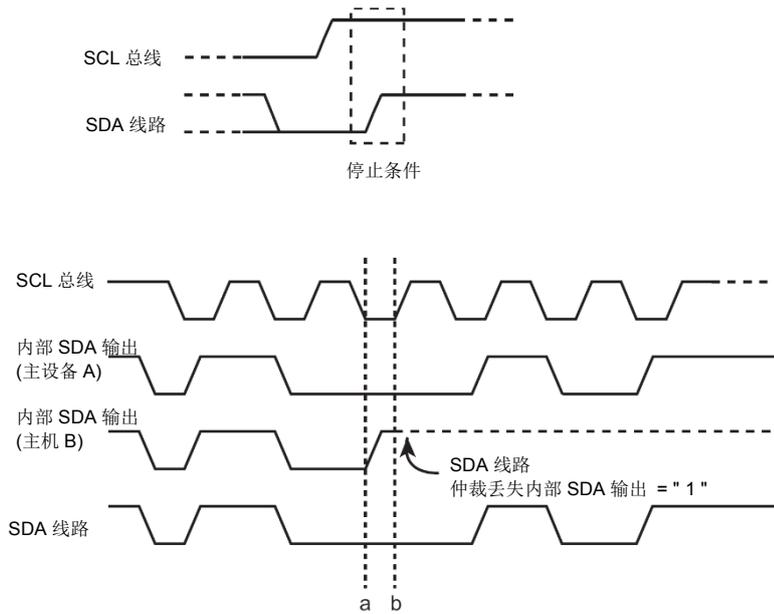


图 13-7 产生停止条件

能读取 I2CxSR<BB>，以检查总线状态。在总线上检测到启动条件时（总线忙），将<BB>设置至“1”；检测到停止条件时（总线空闲）将<BB>清“0”。

按照 I2CxSR 状态进行常见设置的示例如下表所示。

虽然 I2CxCR2<MST>, <TRX>, <BB>, <PIN>位被给予独立功能, 但是按照 I2CxSR 设置值, 它们用在常见组合中, 如下所示。

I2CxSR			I2CxCR2				操作
<MST>	<BB>	<PIN>	<MST>	<TRX>	<BB>	<PIN>	
0	0	1	0	0	0	0	等待作为从机的起始条件。
			1	1	1	1	产生起始条件。
1	1	0	1	1	0	1	产生停止条件。
			0	0	0	1	释放内部中断, 以便重启

注: 当写入这些位时, 不要错误地改变 I2CxCR2<I2CM>。

13.4.9 中断服务请求和解除

在主机模式下，当用 I2CxCR1<BC>和 I2CxCR1<ACK>设置的时钟周期数传输完成时，产生 I2C 总线接口请求(INTI2Cx)。

在从机模式下，在下列条件下产生 INTI2Cx 中断。

- 当 I2CxCR1<NOACK>为"0"时，在输出应答信号后。该应答信号是在接收的从机地址与 I2CxAR<SA[6:0]>中设置的从机地址匹配时产生。
- 当 I2CxCR1<NOACK>为"0"时，在接收到一般调用地址而产生应答信号后。
- 收到全呼地址后从地址匹配或数据传输完成时。

当产生中断请求(INTI2Cx)时，I2CxCR2<PIN>清"0"。当<PIN>清"0"时，I2C 将 SCL 线拉至"低"电平。当读写 I2CxDBR 时，<PIN>设为"1"。在<PIN>被设置为"1"之后，SCL 线路的解除需花费一段时期 t_{LOW} 。在程序将"1"写入到<PIN>时，其即被设置为"1"。然而，写入"0"不会将该位清"0"。

注：当从机地址和方向位在主机模式下传输而发生仲裁丢失时，<PIN>清"0"，并发生 INTI2Cx。这与从机地址是否与<SA>匹配无关。

13.4.10 I2C 总线模式

当 I2CxCR2<I2CM>设为"1"时，选择 I2C 总线模式，以保证在<I2CM>设为"1"前，I2C 总线接口引脚处于"高"电平。此外，还应确保在将操作模式切换到端口模式之前，总线处于空闲状态。

注：当 I2CxCR2<I2CM> = "0"时，除 I2CxCR2<I2CM>位外，无任何数据写入 I2CxCR2 寄存器中的各位。在设置 I2CxCR2 前，将"1"写入 I2CxCR2<I2CM>，以选择 I2C 总线模式。

13.4.11 软件复位

若 I2C 总线接口电路因外部噪声而锁定，用软件复位就能对其进行初始化。

将"10"后跟"01"写入 I2CxCR2<SWRES[1:0]>，就能产生复位信号，对 I2C 总线接口电路进行初始化。在复位后，除 I2CxCR2<I2CM>和 I2CxDBR 寄存器外，所有控制寄存器和状态标志被初始化为它们的复位值。当 I2C 总线接口被初始化时，<SWRES[1:0]>自动清"0"。

注：软件复位造成 I2C 操作模式从 I2C 模式切换到端口模式。

13.4.12 仲裁丢失检测监控器

I2C 总线具备多主设备功能（一条总线上有两台或多台主设备）并且要求执行总线仲裁程序，确保数据传输正确。

试图在总线占用期间生成该启动条件的某个主机丢失总线仲裁，SDA 和 SCL 线路上未出现启动条件。I2C 总线仲裁会发生在 SDA 线路上。某个启动条件在总线占用状态下被输出时，某个启动条件不会被输出到 SCL 或 SDA 线路，从而发生仲裁丢失。

一条总线上的两个主机的仲裁程序如以下所述。

在到达点 a 之前，主机 A 和主机 B 输出相同的数据。在点 a 时，主机 A 输出"低"电平，主机 B 输出"高"电平。

然后，主机 A 将 SDA 总线拖至"低"电平，原因是该线路具备配线 AND 连接。在 SCL 线路在点 b 变为高时，从机会读取 SDA 线路数据(即主机 A 所发送的数据)。此时，主机 B 所发送的数据变为无效。

主机 B 的这种情况被称为"仲裁丢失"。主机 B 释放其 SDA 引脚，以致它不影响另一个主机启动的数据传输。如果两个或更多主机已经发送完全相同的第一个数据字，则会对第二个数据字继续进行仲裁过程。

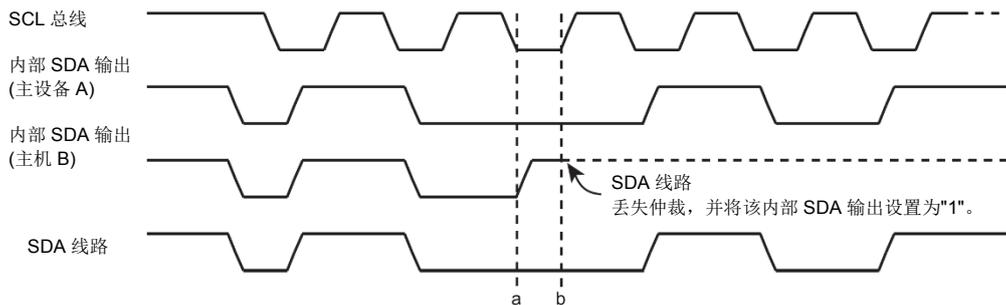


图 13-8 丢失的仲裁

主机会比较 SDA 总线电平和 SCL 线路上升部位的内部 SDA 输出电平。若这两个值有差别，则发生仲裁丢失，I2CxSR<AL>设为"1"。

当<AL>设为"1"时，I2CxSR<MST, TRX>清"0"，造成 I2C 作为从机接收器操作。因此，在<AL>被设置为"1"之后，串行总线接口电路会在数据传输期间停止该时钟输出。

当向/从 I2CxDBR 写入/读取数据时或向 I2CxCR2 写入数据时，<AL>清"0"。

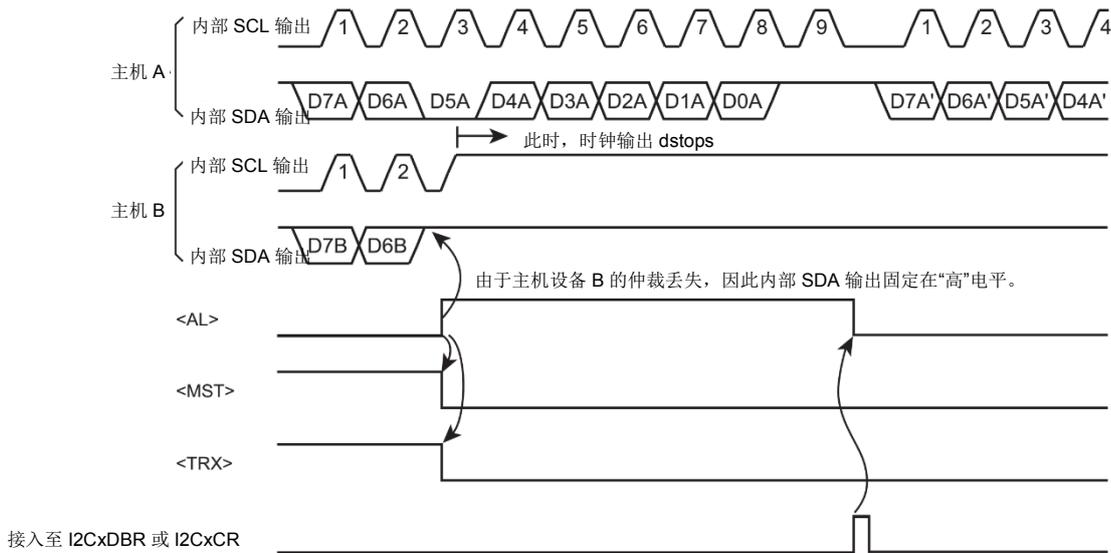


图 13-9 主机 B 丢失的仲裁示例(D7A = D7B, D6A = D6B)

13.4.13 从机设备地址匹配检测监控程序

在地址识别模式下，I2C 作为从机设备操作时(I2CxAR<ALS>="0")，在接收到一般调用地址或与 I2CxAR 指定的值匹配的从机地址后，I2CxSR<AAS>设为"1"。

I2CxCR1<NOACK>清"0"，就可启用从机地址匹配检测。当接收到一般调用或者主机发送的从机地址与 I2CxAR<SA>中设置的从机地址匹配时，I2CxSR<AAS>设为"1"。

将 I2CxCR1<NOACK>设为"1"，就能禁用从机地址匹配检测。即使接收到一般调用或者主机发送的从机地址与 I2CxAR<SA>中设置的从机地址匹配，I2CxSR<AAS>仍保持"0"。

当自由数据格式模式<ALS>设为"1"时，<AAS>在已收到第一个数据字时设为"1"。当数据写入 I2CxDBR 或者从 I2CxDBR 中读取时，<AAS>清"0"。

13.4.14 一般调用检测监控程序

I2C 作为从机设备运行时，当它接收到一般呼叫地址时(即在起始条件后的 8 位全为零)，I2CxSR<AD0>设为"1"。

在总线上检测到起始或停止条件时，<AD0>清"0"。

13.4.15 最后接收位监控程序

I2CxSR<LRB>设为在 SCL 线上升沿读取的 SDA 线值。

在应答模式下，若在 INTI2Cx 中断请求产生后立即读取 I2CxSR<LRB>，将导致 ACK 信号被读取。

13.4.16 数据缓冲缓冲器 (I2CxDBR)

读取或写入 I2CxDBR 将会启动读取接收数据或写入发送数据。

在 I2C 作为主机操作时，将从机地址和方向位设置到该寄存器，即可生成起始条件。

13.5 I2C 总线模式下的数据传输规程

13.5.1 装置初始化

在确保 SDA 和 SCL 引脚为高 (总线空闲)后, 将 I2CxCR2<I2CM>设为"1", 以启用 I2C。

接着, 将"1"写入 I2CxCR1<ACK>, 将"0"写入 I2CxCR1<NOACK>。同时将"000"写入 I2CxCR1<BC[2:0]>。

这些设置启用应答操作、从机地址匹配检测和一般调用检测以及将数据长度设为 8 位。

在 I2CxCR1<SCK>中设置"tHIGH"和"tLOW"。

然后, 通过在<SA[6:0]>中指定从机地址, 在<ALS>中指定地址识别模式而对 I2CxAR 进行编程。(在使用寻址格式时, 必须将<ALS>清"0")。

将 I2C 总线接口配置为从机接收器时, 应确保 I2C 总线接口引脚为"高"。最后, 将"0"写入 I2CxCR2<MST, TRX, BB>, 将"1"写入 I2CxCR2<PIN>, 将"00"写入 I2CxCR2<SWRES[1:0]>, 使设备配置为从机接收器。

注: 串行总线接口电路的初始化必须在总线连接的所有装置初始化后任何装置均未生成启动条件的时间内完成。如果不遵循该规则, 则可能不会正确的接收数据, 原因是其它设备可能在该串行总线接口电路的初始化完成之前开始传输。

		7	6	5	4	3	2	1	0	
I2CxCR2	←	0	0	0	1	1	0	0	0	启用 I2C
I2CxCR1	←	0	0	0	1	0	1	1	0	指定 ACK 和 SCL 时钟。
I2CxAR	←	x	x	x	x	x	x	x	x	指定一个从机地址和一种地址识别模式。
I2CxCR2	←	0	0	0	1	1	0	0	0	将 I2C 配置为从机接收器。

注: x: 忽略

13.5.2 生成启动条件和从设备地址

13.5.2.1 主设备模式

在主模式下, 必须执行以下步骤, 以生成该启动条件和一个从机地址。

首先, 确保总线处于空闲状态(<BB> = "0")。然后, 将"1"写入 I2CxCR1<ACK>, 以选择应答模式。将发送的从机地址和方向位写入 I2CxDBR。

当<BB> = "0"时, 将"1111"写入 I2CxCR2<MST, TRX, BB, PIN>, 可在总线上产生起始条件。在起始条件后, I2C 产生 SCL 引脚的 9 个时钟。I2C 用最先的 8 个时钟输出 I2CxDBR 指定的从机地址和方向位, 在第 9 个时钟释放 SDA 线, 以接收从机设备的应答信号。

INTI2Cx 中断请求在第 9 个时钟的下降沿产生, <PIN>清"0"。在主机模式下, I2C 保持 SCL 线在"低"电平, 同时<PIN> = "0"。若从机设备已返回应答信号, 则在 INTI2Cx 中断请求产生时, <TRX>根据发送的方向位改变其值。

注 1: 输出从机地址时, 用软件检查总线在写入 I2CxDBR 前是否为空闲状态。如果不遵循该规则, 则正在该总线上输出的数据可能会被破坏。

注 2: 在写入从机地址后设备产生起始条件的同时, 其他主机设备可以传输。因此, 在重新产生起始条件时, 您的程序应确认总线空闲。上述程序在写入从机地址后 98.0 μs (在 I2C 标准的 STANDARD 模式下, 最小发送时间)或 23.7 μs (在 I2C 标准的 FAST 模式下, 最小发送时间) 内完成。

主程序中的设置

		7	6	5	4	3	2	1	0	
Reg.	←	I2CxSR								
Reg.	←	Reg.AND 0x20								
如果 Reg.	≠	0x00								确保该总线处于空闲状态。
则										
I2xCxCR1	←	x	x	x	1	0	x	x	x	选择该确认模式。
I2CxDBR	←	x	x	x	x	x	x	x	x	指定所需的从机地址和方向。
I2CxCR2	←	1	1	1	1	1	0	0	0	生成启动条件。

INTI2C0 中断程序示例

清除中断请求。

处理

中断结束

13.5.2.2 从设备模式

在从机模式下, I2C 接收起始条件和从机地址。

在接收主机设备发送的起始条件后, 在 SCL 线上, I2C 在前 8 个时钟接收主机设备发送的从机地址和方向位。

若接收的地址与 I2CxAR 指定的其从机地址匹配或者等于一般调用地址, 则第 9 个时钟, I2C 将 SDA 线拉至"低"电平, 并输出应答信号。

INTI2Cx 中断请求在第 9 个时钟的下降沿产生, <PIN>清"0"。在从机模式下, I2C 将 SCL 线保持在"低"电平, 同时<PIN>为"0"。

注: 在下述条件下, 可通过 DMA 进行 I2C 总线传输。

- 主机和从机, 一对一通信
- 顺序连续传输或接收

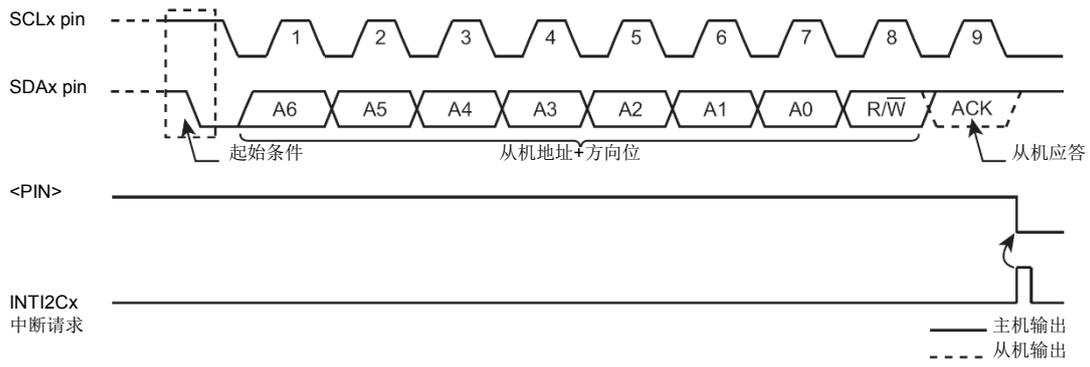


图 13-10 起始条件和从机地址的生成

13.5.3 传输一个数据字

在数据字传输结束时，生成 INTI2Cx 中断以测试 I2CxSR<MST>，借以确定 I2C 是在主机模式下还是在从机模式下。

13.5.3.1 主设备模式(<MST> = "1")

测试<TRX>以确定 I2C 是否被配置为发送器或接收器。

(1) 发送器模式(<TRX> = "1")

测试<LRB>。如果<LRB>为"1"，则表示接收器不要求提供更多的数据。

然后，主机生成后文所述的停止条件，以停止发送。

如果<LRB>为"0"，则表示接收器要求提供更多的数据。若下一发送数据有 8 位，则数据写入 I2CxDBR。若数据长度不同，则<BC[2:0]>和<ACK>被编程，发送数据写入 I2CxDBR。该数据的写入可使得<PIN>变为"1"，导致 SCL 引脚生成一个下一数据字传输用串行时钟，并导致 SDA 引脚传输该数据字。

传输完成后，产生 INTI2Cx 中断请求，<PIN>清"0"，且 SCL 引脚被拉至“低”电平。

在发送更多的数据字时，需重新测试<LRB>，并重复以上程序。

INTI2Cx 中断

如果 MST = 0

则转到从机模式处理。

如果 TRX = 0

则转到接收器模式处理。

如果 LRB = 0

则转到停止条件生成处理。

I2CxCR1	←	x	x	x	x	0	x	x	x	指定拟发送的位的数目，并指定是否需要 ACK。
I2CxDBR	←	x	x	x	x	x	x	x	x	写入该发送数据。

中断处理结束

注：x：忽略

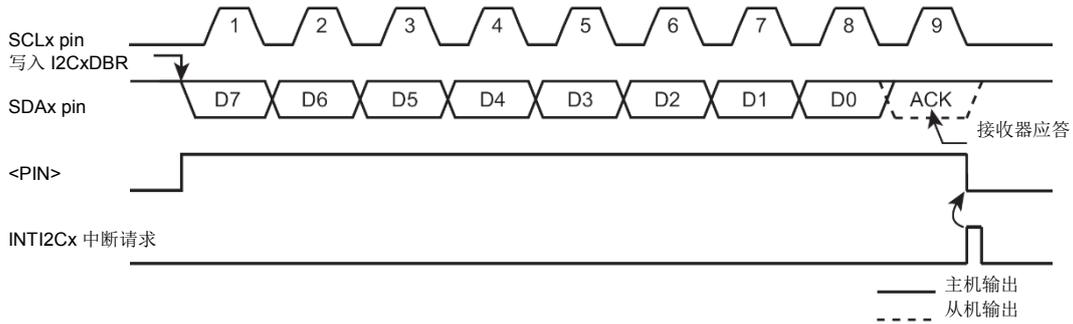


图 13-11 <BC[2:0]> = "000", <ACK> = "1" (发送器模式)

(2) 接收器模式(<TRX> = "0")

若下一发送数据有 8 位，发送数据写入 I2CxDBR。

若数据长度不同，则<BC[2:0]>和<ACK>被编程，接收数据从 I2CxDBR 中读取，以释放 SCL 线。(在从机地址发送后立即读取的数据未定义。) 在读取数据后，<PIN>设为"1"，串行时钟被输出到 SCL 引脚，以发送下一数据字。在最后一位，当应答信号变成"低"电平时，"0"被输出到 SDA 引脚。

此后，产生 INTI2Cx 中断请求，<PIN>清"0"，由此将 SCL 引脚拉到"低"电平。每次从 I2CxDBR 读取接收数据时，就会输出一字发送时钟及应答信号。

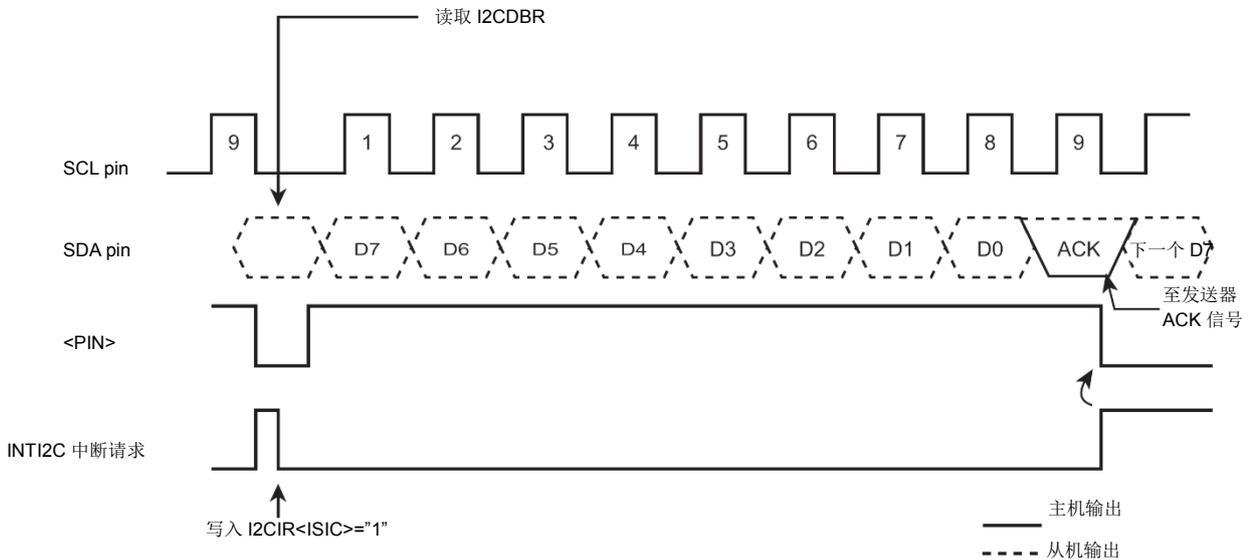


图 13-12 <BC[2:0]> = "000", <ACK> = "1" (接收器模式)

终止发送器的数据发送时，在读取倒数第二个数据字前，<ACK>必须立即清"0"。

这样可禁止生成最后数据字的确认时钟。

在传输完成时，一个中断请求即被生成。在中断处理完成之后，必须将<BC[2:0]>设置为"001"，并必须读取该数据，以确保生成 1 位传输的时钟。

此时，主接收器会将 SDA 总线保持在"高"电平，其会将对发送器传输的结束作为应答信号发出。

终止 1 位数据的接收时，在中断处理中，产生停止条件，以终止数据传输。

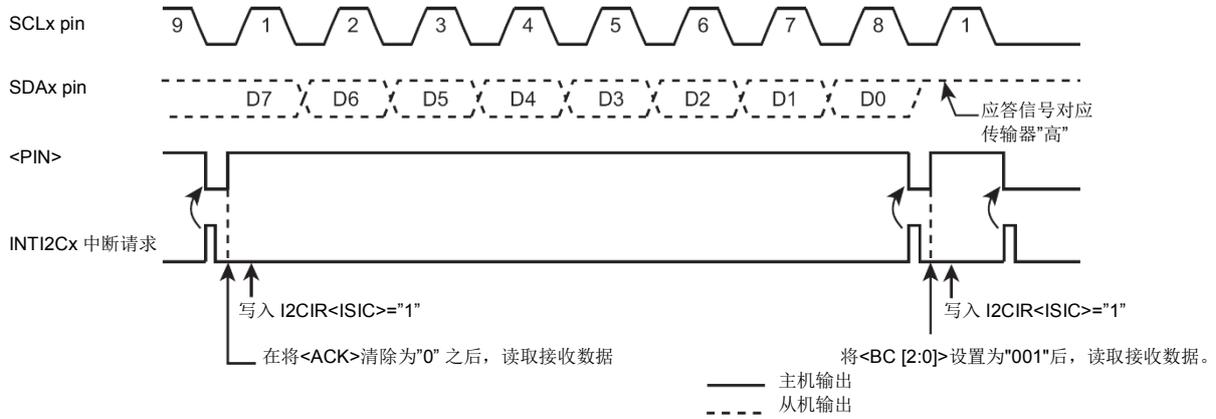


图 13-13 在主机接收器模式下终止数据发送

示例：在接收 N 数据字时

INTI2Cx 中断(在数据发送后)

		7	6	5	4	3	2	1	0	
I2CxCR1	←	x	x	x	x	0	x	x	x	设置拟接收的数据位的数目，并指定是否需要 ACK。
Reg	←	I2CxDBR								读取虚拟数据。
中断终止										

INTI2Cx 中断(第 1 个至第(N - 2)个数据的接收)

		7	6	5	4	3	2	1	0	
Reg	←	I2CxDBR								读取第 1 个 ~ 第(N - 2)个数据字。
中断终止										

INTI2Cx 中断(第(N - 1)个数据的接收)

		7	6	5	4	3	2	1	0	
I2CxCR1	←	x	x	x	0	0	x	x	x	禁止确认时钟的生成。
Reg	←	I2CxDBR								读取第(N - 1)个数据字。
中断终止										

INTI2Cx 中断(第 N 个数据的接收)

		7	6	5	4	3	2	1	0	
I2CxCR1	←	0	0	1	0	0	x	x	x	禁止确认时钟的生成。
Reg	←	I2CxDBR								读取第 N 个数据字。
中断终止										

INTI2Cx 中断(在完成数据接收后)

进行处理，以生成该停止条件。 终止数据发送。

中断终止

注：X：忽略

13.5.3.2 从设备模式(<MST> = "0")

在从机模式下，I2C 在四种情况下产生 INTI2Cx 中断请求：

- 1) 当 I2CxCR1<NOACK>为"0"，I2C 已接收到一般调用地址时。
- 2) 当 I2CxCR1<NOACK>为"0"，接收的从机地址与 I2CxAR<SA>地址匹配时。
- 3) 当响应接收的从机地址匹配或一般调用地址，已完成数据传输时。

同时，若在主机模式下 I2C 检测到仲裁丢失，则切换到从机模式。

若在数据字传输中检测到仲裁丢失，则在传输完成后，产生 INTI2Cx 中断请求，<PIN>清"0"，SCL 引脚被拉到"低"电平。

当数据写入 I2CxDBR 或者从 I2CxDBR 读取时，或者当<PIN>设为"1"时，SCLx 引脚在 t_{LOW} 时间后被释放。

在从属模式下，可执行正常从属模式处理或因仲裁而需进行的处理。

对 I2CxSR<AL>，<TRX>，<AAS>和<AD0>进行测试以确定所需处理。 "

从机模式状态及所需处理如表 13-3"在从机模式下的处理"所示。

示例：在从机接收器模式下，当接收的从机地址与 I2C 本身地址匹配并且方向位为"1"时。

INTI2Cx 中断

如果 TRX = 0

则转到其它处理。

如果 AL = 0

则转到其它处理。

如果 AAS = 0

则转到其它处理。

I2CxCR1 ← x x x 1 0 x x x 设置拟发送的位的数目。

I2CxDBR ← x x x x x x x x 设置该发送数据。

注：x： 忽略

表 13-3 在从机模式下处理

<TRX>	<AL>	<AAS>	<AD0>	状态	处理
1	1	1	0	当从机地址正在发送, I2C 接收到另一主机发送的从机地址及方向位"1"时, 检测到仲裁丢失。	将<BC[2:0]>设为数据字位数, 并将发送数据写入 I2CxDBR。
		1	0	在从机接收器模式下, I2C 接收到主机发送的从机地址及方向位"1"	
	0	0	0	在从机发送器模式下, I2C 已完成了一个数据字的发送。	测试 LRB。若它已设为"1", 则意味着接收器不需要进一步的数据。将<PIN>设为 1, 并将<TRX>复位为 0, 以释放总线。若<LRB>已复位为"0", 则意味着接收器需要进一步的数据。将<BC[2:0]>设为数据字位数, 并将发送数据写入 I2CxDBR。
0	1	1	1/0	当从机地址正在发送, I2C 接收到另一主机发送的从机地址及方向位"0"或一般调用地址时, 检测到仲裁丢失。	读取 I2CxDBR(假读), 以便将<PIN>设为 1, 或者将"1"写入<PIN>。
		0	0	当从机地址或数据字正在发送, 传输终止时, 检测到仲裁丢失。	
	0	1	1/0	在从机接收器模式下, I2C 接收主机发送的从机地址及方向位"0" 或者一般调用地址。	
		0	1/0	在从机接收器模式下, I2C 已完成一个数据字的接收。	将<BC[2:0]>设为数据字位数, 并读取 I2CxDBR 的接收数据。

注: 在从机模式下, 若 I2CxAR<SA[6:0]>设为"0x00", 则接收 I2C 总线标准的 START 字节(0x01), 检测从机地址匹配, 并将 I2CxSR<TRX>设为"1"。不要将 I2CxAR<SA[6:0]>设为"0x00"。

13.5.4 生成停止条件

当 I2CxSR<BB>为"1"时，将"1"写入 I2CxCR2<MST, TRX, PIN>，并将"0"写入<BB>会导致 I2C 开启一个可在总线上产生停止条件的时序。

不得改变<MST, TRX, BB, PIN>的内容，直到总线上出现停止条件。

若另一设备限制了 SCL 总线，则 I2C 需等至 SCL 线被释放。

之后，SDA 引脚走"高"，由此引起停止条件的产生，时间为" t_{HIGH} "。

		7	6	5	4	3	2	1	0	
I2CxCR2	←	1	1	0	1	1	0	0	0	生成停止条件。

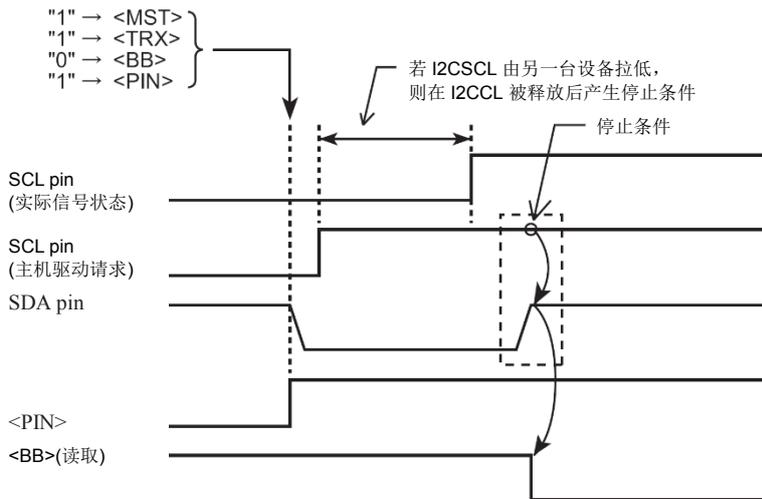


图 13-14 产生停止条件

13.5.5 重新启动规程

在主机未终止对某个从机的传输的情况下改变数据传输方向时，就需要进行重新启动。在主模式生成重新启动的程序如以下所述。

首先，将"0"写入 I2CxCR2<MST, TRX, BB>，及将"1"写入<PIN>，以释放总线。此时，SDAx 引脚保持在"高"电平，SCLx 引脚被释放。由于该总线上未生成停止条件，因此，其它设备会将该总线识别为占用。

然后测试 I2CxSR<BB>并等到它变为"0"，以确保 SCLx 引脚被释放。

接着，测试<LRB>并等到它变为"1"，以确保没有其他设备正将 SCLx 总线拉至"低"电平。

通过执行以上程序一旦确定总线空闲，则执行"13.5.2 产生起始条件和从机地址"中的程序以产生起始条件。

为了满足重启的设置时间，在确定总线空闲后，软件必须创建（标准模式下）至少 4.7 μs 等待时间。

注 1: 在其"0"时，不要将<MST> 写为"0"。(重新启动无法启动。)

注 2: 在主机作为接收器工作时, 必须在重新启动生成之前, 完成从作为发送器工作的从机进行的数据发送。为完成数据传输, 从机必须接收一个"高"电平应答信号。因此, 在产生重启前<LBR>变为"1", 甚至按照复位程序确认<LBR>= "1"时, SCL 线上升沿也检测不到。在检查 SCL 线的状态时, 需读取该端口。

<p>┌───┐</p> <p>└───┘</p> <p>┌───┐</p> <p>└───┘</p>	<p>← 7 6 5 4 3 2 1 0</p> <p>I2CxCR2 ← 0 0 0 1 1 0 0 0</p> <p>如果 I2CxSR<BB> ≠ 0</p> <p>则</p> <p>如果 I2CxSR<LRB> ≠ 1</p> <p>则</p> <p>4.7μs 等待</p> <p>I2CxCR1 ← x x x 1 0 x x x</p> <p>I2CxDBR ← x x x x x x x x</p> <p>I2CxCR2 ← 1 1 1 1 1 0 0 0</p> <p>注: x; 忽略</p>	<p>解除该总线</p> <p>检查并确认 SCL 引脚已被解除。</p> <p>检查并确认没有别的设备会将 SCL 引脚拖至"低"。</p> <p>选择该确认模式。</p> <p>设置所需的从机地址和方向。</p> <p>生成启动条件。</p>
---	---	---

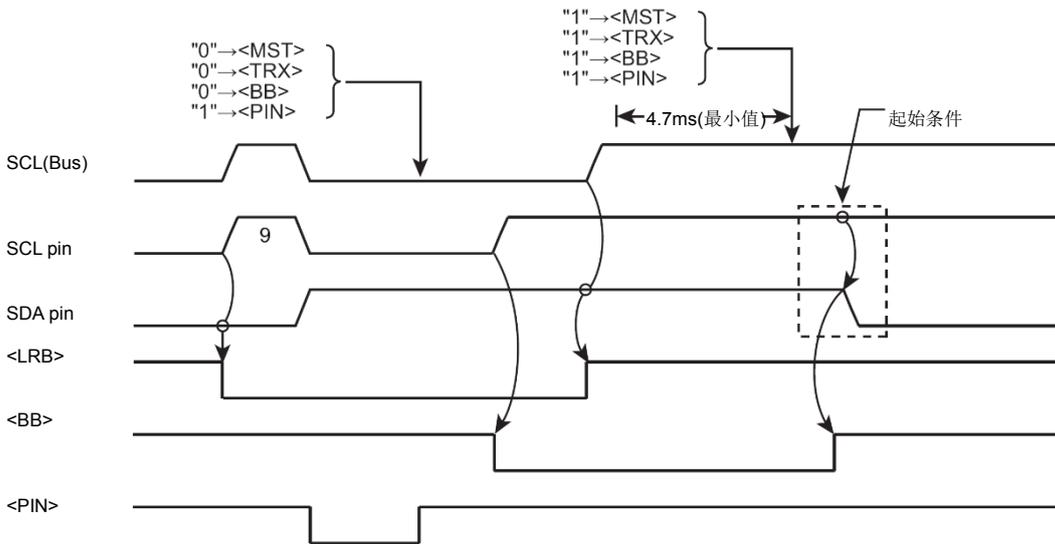


图 13-15 重启生成时序图

13.6 使用注意事项

13.6.1 软件复位后的寄存器值

软件复位使非 I2CxCR2<I2CM>的 I2C 寄存器及内部电路初始化,并释放 SCL 线和 SDA 线。(见"13.4.1.2 时钟同步"章节。)

然而, 根据软件复位后的读取时序, 读取 I2CxSR<LRB>可返回非初始值"0"的值。

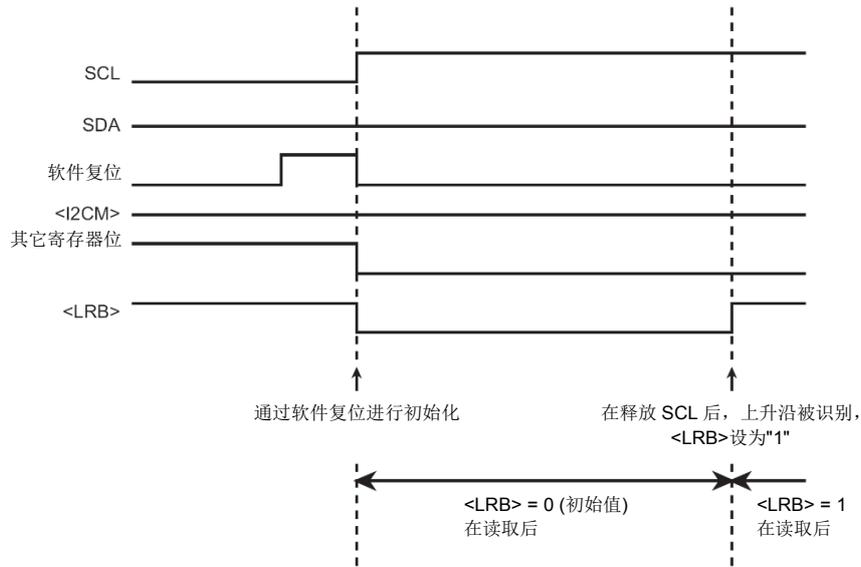


图 13-16 软件复位释放 SCL0 "0"到"1", 同时 SDA = "1"

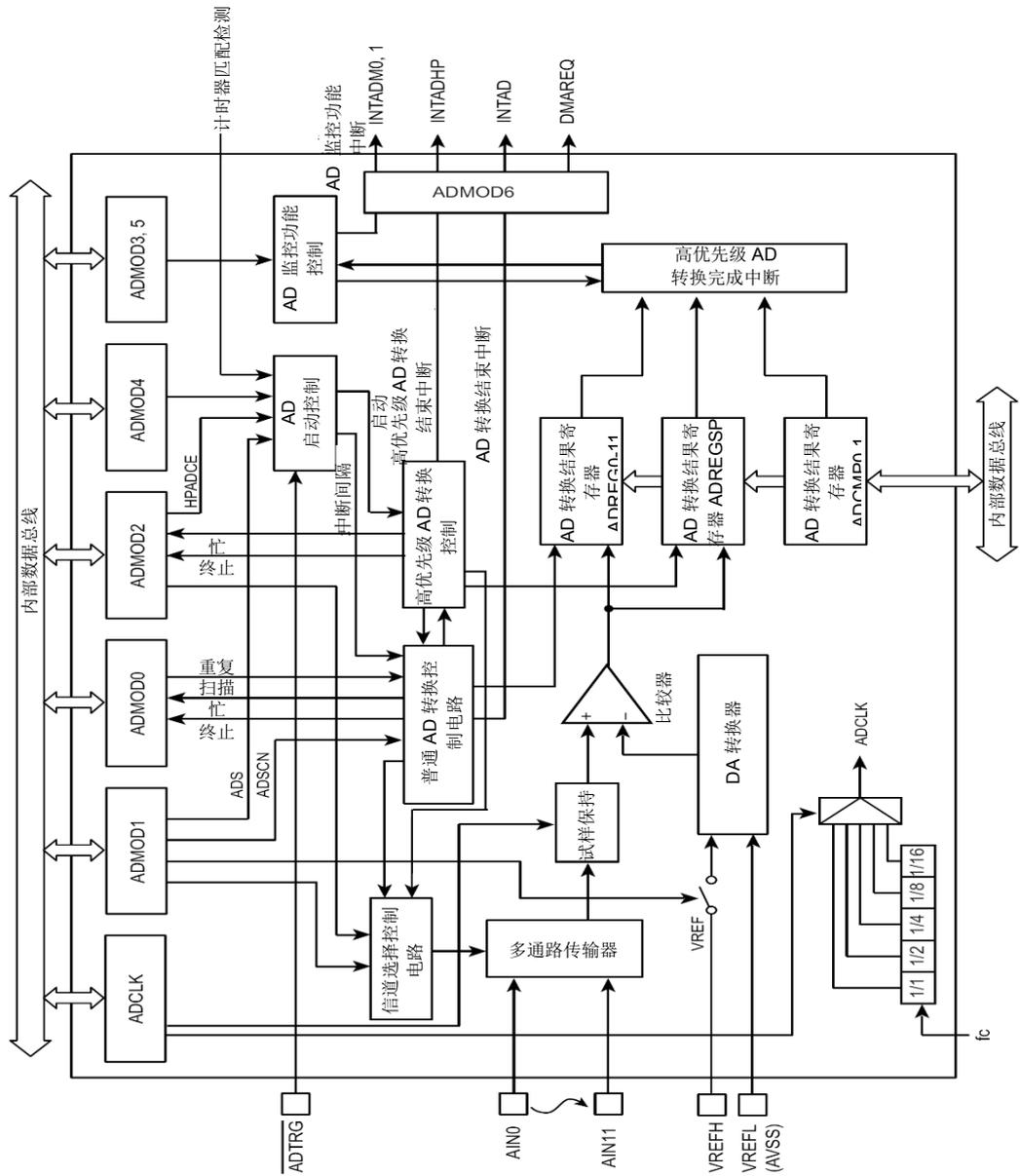
14. 10-位模拟/数字转换器 (ADC)

14.1 概要

TMPM037FWUG 内置 10 位顺次转换模拟/数字转换器 (AD 转换器)。为了确认使用通道和设置, 详见 "产品信息"。

14.2 配置

此 AD 转换器框图如图 14-1 所示。



注：共享 VREFH 和 AVDD3。共享 VREFL 和 AVSS。

图 14-1 10 位 ADI 转换器框图

14.3 寄存器

14.3.1 寄存器列表

控制寄存器和 AD 转换器地址如下。

有关基地址的详细情况，参照"内存印象图"一章的"外部功能地址表"。

寄存器名称		地址(基+)
转换时钟设置寄存器	ADCLK	0x0000
模式控制寄存器 0	ADMOD0	0x0004
模式控制寄存器 1	ADMOD1	0x0008
模式控制寄存器 2	ADMOD2	0x000C
模式控制寄存器 3	ADMOD3	0x0010
模式控制寄存器 4	ADMOD4	0x0014
模式控制寄存器 5	ADMOD5	0x0018
模式控制寄存器 6	ADMOD6	0x001C
转换结果寄存器 0	ADREG0	0x0030
转换结果寄存器 1	ADREG1	0x0034
转换结果寄存器 2	ADREG2	0x0038
转换结果寄存器 3	ADREG3	0x003C
转换结果寄存器 4	ADREG4	0x0040
转换结果寄存器 5	ADREG5	0x0044
转换结果寄存器 6	ADREG6	0x0048
转换结果寄存器 7	ADREG7	0x004C
转换结果寄存器 8	ADREG8	0x0050
转换结果寄存器 9	ADREG9	0x0054
转换结果寄存器 10	ADREG10	0x0058
转换结果寄存器 11	ADREG11	0x005C
转换结果寄存器 SP	ADREGSP	0x0060
转换结果比较寄存器 0	ADCMP0	0x0064
转换结果比较寄存器 1	ADCMP1	0x0068

14.3.2 ADCLK (转换时钟设置寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-4	ADCC[1:0]	R/W	选择 AD 转换时钟计数 00: 35.5 转换时钟 01: 42 转换时钟 10: 68 转换时钟 11: 81 转换时钟
5-4	-	R	读作 0。
3-0	ADCLK[3:0]	R/W	选择模拟数字转换时钟(注 1) (注 2) 0000: fc 1000: fc/6 0001: fc/2 1001: fc/12 0010: fc/4 1010: fc/24 0011: fc/8 1011: fc/48 0100: fc/16 1100: fc/96 0101 - 0111: 保留 0101-0111 & 1101 - 1111: 保留

注 1: AD 转换期间切勿改变 AD 转换设置。

注 2: ADCLK 的 AD 转换时钟设置小于等于 f_{sys} (系统时钟)。(ADCLK \leq f_{sys})

为了满足下列条件, 必须有时钟计数。

VREFH AVDD	转换时间
2.7 ~ 3.6 V	$\geq 16.2 \mu s$
2.3 ~ 3.6 V	$\geq 32.4 \mu s$

14.3.3 ADMOD0 (模式控制寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	EOCFN	ADBFN	-	ITM		REPEAT	SCAN	ADS
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7	EOCFN	R	正常 AD 转换完成标志(注 1) 0: 转换之前或期间 1: 完成 该位为"0", 读取时清除。
6	ADBFN	R	正常 AD 转换忙标志 0: 转换停止 1: 转换期间
5	-	R	读作 0。
4-3	ITM[1:0]	R/W	在固定通道重复转换模式下的中断 00: 每 1 次转换生成一次中断 01: 每 4 次转换时, 产生中断一次 10: 每 8 次转换时, 产生中断一次 11: 禁止设置 仅在固定通道重复模式(<REPEAT> = "1", <SCAN> = "0")中计时有效。
2	REPEAT	R/W	指定重复模式 0: 单次转换模式 1: 重复转换模式
1	SCAN	R/W	指定扫描模式 0: 固定通道模式 1: 通道扫描模式 如果选择通道扫描模式, 应将 ADMOD1<ADSCAN>设为通道号。
0	ADS	W	开始 AD 转换 0: 忽略 1: 开始转换 必须在设置该模式后进行转换。总读作"0"。

14.3.4 ADMOD1 (模式控制寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VREFON	I2AD	ADSCN		ADCH			
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7	VREFON	R/W	VREF 应用控制(注) 0: OFF 1: ON
6	I2AD	R/W	IDLE 模式中指定操作模式 0: 停止 1: 操作
5-4	ADSCN[1:0]	R/W	指定通道扫描模式中的运行模式 00: 4-通道扫描 01: 8-通道扫描 10: 12-通道扫描 11: 保留 当用 ADMOD0<SCAN>选择通道扫描模式时, 指定工作模式。转换通道通过<ADCH>设置值选择。见下表。
3-0	ADCH[3:0]	R/W	选择模拟输入通道(参阅下表。)

注: 在开始 AD 转换前, 将"1"写入<VREFON>位, 先等 3 μs, 在此期间内部参考电压应稳定, 然后将"1"写入 ADMOD0<ADS>。

模拟输入通道的选择

		<ADCH[3:0]>							
		0000	0001	0010	0011	0100	0101	0110	0111
ADMOD0 <SCAN>=0	固定通道	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7
ADMOD0 <SCAN>=1	<ADSCN>=00 4-通道扫描	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN4	AIN4~ AIN5	AIN4~ AIN6	AIN4~ AIN7
	<ADSCN>=01 8-通道扫描	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7
	<ADSCN>=10 12-通道扫描	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7

		<ADCH[3:0]>							
		1000	1001	1010	1011	1100	1101	1110	1111
ADMOD0 <SCAN>=0	固定通道	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15
ADMOD0 <SCAN>=1	<ADSCN>=00 4-通道扫描	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN4	AIN4~ AIN5	AIN4~ AIN6	AIN4~ AIN7
	<ADSCN>=01 8-通道扫描	AIN0	AIN0~ AIN1	AIN0~ AIN2	AIN0~ AIN3	AIN0~ AIN4	AIN0~ AIN5	AIN0~ AIN6	AIN0~ AIN7
	<ADSCN>=10 12-通道扫描	AIN0~ AIN8	AIN0~ AIN9	AIN0~ AIN10	AIN0~ AIN11	-	-	-	-

14.3.5 ADMOD2 (模式控制寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	EOCFHP	ADBFHP	HPADCE	-	HPADCH			
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7	EOCFHP	R	最优先 AD 转换完成标志(注) 0: 转换之前或期间 1: 完成
6	ADBFHP	R	最优先级 AD 转换 BUSY 标志 0: 转换暂停期间 1: 转换期间
5	HPADCE	R/W	激活最优先级转换 0: 忽略 1: 开始转换 总读作"0"。
4	-	R/W	写入"0"。
3-0	HPADCH[3:0]	R/W	当激活最优先级转换时, 选择模拟输入通道。(见下表)

注: 该位为"0", 读取时清除。

模拟输入通道的选择

HPADCH[3:0]	0000	0001	0010	0011	0100	0101	0110	0111
转换通道	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7

HPADCH[3:0]	1000	1001	1010	1011	1100	1101	1110	1111
转换通道	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15

14.3.6 ADMOD3 (模式控制寄存器 3)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	ADOBIC0	ADREGS0				ADOBSV0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7	-	R/W	写入"0"。
6	-	R	读作 0。
5	ADOBIC0	R/W	设置 AD 监控程序功能中断 0 0: 如果转换结果值小于比较寄存器 0, 则生成中断。 1: 如果转换结果值大于比较寄存器 0, 则生成中断。
4-1	ADREGS0[3:0]	R/W	使用 AD 监控程序功能 0 时选择目标转换结果寄存器(见下表)。
0	ADOBSV0	R/W	模拟数字监控程序功能 0 0: 禁用 1: 启用

<ADREGS0[3:0]>	待比较的转换结果 寄存器	<ADREGS0[3:0]>	待比较的转换结果 寄存器
0000	ADREG0	0100	ADREG4
0001	ADREG1	0101	ADREG5
0010	ADREG2	0110	ADREG6
0011	ADREG3	0111	ADREG7
-	-	1xxx	ADREGSP

14.3.7 ADMOD4 (模式控制寄存器 4)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	HADHS	HADHTG	ADHS	ADHTG	-	-	ADRST	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7	HADHS	R/W	激活最优先级 AD 转换的 H/W 源 0: 外触发 1: 与计时器寄存器匹配(注 1)
6	HADHTG	R/W	激活最优先级模拟数字转换的 H/W 0: 禁用 1: 启用
5	ADHS	R/W	用于激活正常 AD 转换的 H/W 源(注 2) 0: 外触发 1: 与计时器寄存器匹配(注 1)
4	ADHTG	R/W	用于激活正常 AD 转换的 HW 0: 禁用 1: 启用
3-2	-	R	读作 0。
1-0	ADRST[1:0]	W	用 01 重写 10, 可使 ADC 被软件复位。(注 3)

注 1: 为了确认 H/W 源, 详见"产品信息"。

注 2: 当外部触发器用于最优先 AD 转换的 H/W 激活时, 它就不能用于 AD 转换的 H/W 激活。

注 3: 软件复位初始化所有寄存器, ADCLK<ADCLK>除外。

14.3.8 ADMOD5 (模式控制寄存器 5)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	ADOBIC1	ADREGS1				ADOBSV1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-6	-	R	读作 0。
5	ADOBIC1	R/W	设置AD监控程序功能中断 1。 0: 若转换结果值小于比较寄存器1, 生成中断。 1: 若转换结果值大于比较寄存器1, 生成中断。
4-1	ADREGS1[3:0]	R/W	在使用该AD监控器功能 1 时, 选择某个目标转换结果寄存器(见下表)。
0	ADOBSV1	R/W	AD监控功能 1 0: 禁用 1: 启用

<ADREGS1[3:0]>	待比较的转换结果寄存器	<ADREGS1[3:0]>	待比较的转换结果寄存器
0000	ADREG0	0100	ADREG4
0001	ADREG1	0101	ADREG5
0010	ADREG2	0110	ADREG6
0011	ADREG3	0111	ADREG7
-	-	1xxx	ADREGSP

14.3.9 ADMOD6 (模式控制寄存器 6)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	ADM1DMA	ADM0DMA	ADHPDMA	ADDMA
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	-	R	读作 0。
3	ADM1DMA	R/W	指定AD监控器功能 1 DMA激活系数(通过INTADM1触发) 0: 禁用 1: 启用
2	ADM0DMA	R/W	指定AD监控器功能 0 DMA激活系数(通过INTADM0触发) 0: 禁用 1: 启用
1	ADHPDMA	R/W	指定最高优先级AD 转换DMA激活系数(通过INTADHP触发) 0: 禁用 1: 启用
0	ADDMA	R/W	指定标准AD转换DMA激活系数(通过INTAD触发) 0: 禁用 1: 启用

14.3.10 ADREGn (转换结果寄存器 n: n = 0 ~ 11)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADRn							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADRo		-	-	-	-	OVRn	ADRnRF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-6	ADRn[9:0]	R	AD转换结果 转换结果被保存。有关转换通道和转换结果寄存器之间的相关性信息，请参看表14-2和表14-3，“14.4.5.7 中断生成计时和AD 转换结果存储寄存器”。
5-2	-	R	读作 0。
1	OVRn	R	超限标志 0: 未生成。 1: 生成。 如果转换结果在读取<ADRo>时被覆盖，则设置成"1"。该位为"0"，读取时清除。
0	ADRnRF	R	AD转换结果保存标志 0: 转换结果未被保存。 1: 转换结果被保存。 如果转换结果已储存，则设置成"1"。 该位为"0"，在转换结果读取时清除。

注：寄存器的存取必须以半字或一字存取。

14.3.11 ADREGSP (AD 转换结果寄存器 SP)

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-6	ADRSP[9:0]	R	AD转换结果 最高优先级模拟数字转换结果已储存。
5-2	-	R	读作 0。
1	OVRSP	R	超限标志 0: 未生成。 1: 生成。 如果在读取<ADRSP>之前盖写了某次转换结果, 则“1”即被设置。该位为“0”, 读取时清除。
0	ADRSPRF	R	AD转换结果保存标志 0: 转换结果未被保存。 1: 转换结果被保存。 如果转换结果已储存, 则设置成“1”。 该位为“0”, 在转换结果读取时清除。

注: 寄存器的存取必须以半字或一字存取。

14.3.12 ADCMP0 (AD 转换结果比较寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADCOM0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADCOM0		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-6	ADCOM0[9:0]	R/W	在AD监控器功能 0 被启用时, 它会设置一个值, 用于与ADMOD3<ADREGS0>所指定的转换结果寄存器的值进行比较。
5-0	-	R	读作 0。

注: 在将值写至该寄存器时, 必须停用AD监控器功能 0(ADMOD3<ADBSV0>="0")。

14.3.13 ADCMP1 (AD 转换结果比较寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADCOM1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADCOM1		-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-6	ADCOM1[9:0]	R/W	在AD监控器功能1被启用时, 它会设置一个值, 用于与ADMOD1<ADREGS1>所指定的转换结果寄存器的值进行比较。
5-0	-	R	读作 0。

注：在将值写至该寄存器时，必须停用AD监控器功能 1(ADM0D5<ADBSV1>="0")。

14.4 操作描述

14.4.1 模拟参考电压

模拟基准电压的"高"电平应施加到VRFEH引脚，而"低"电平应施加到VREFL引脚。

要启动AD转换，你务必将"1"写入<VREFON>位，等待 3 μ s，期间内部参考电压应稳定，然后将"1"写入ADMOD0<ADS>位。

如果用户不使用ADC功能，则可将"0"写入到ADMOD1<DAON>中。模拟电路的消耗电流即被减小。

注：在TMPM037FWUG中，VREFH和AVDD3被共享。VREFL和AVSS也被共享。

14.4.2 AD 转换模式

支持两种类型的AD转换：标准AD转换和最高优先级AD转换。对于标准AD转换，支持以下四种操作模式。

14.4.2.1 标准 AD 转换

对于标准AD转换，以下四种操作模式均受到支持，且可用ADMOD0<REPEAT><SCAN>选择该操作模式。

- 固定通道单次转换模式
- 通道扫描单次转换模式
- 固定通道重复转换模式
- 通道扫描重复转换模式

对于通道扫描模式而言，以下三种操作模式均受到支持，且可用ADMOD1<ADSCN>选择操作模式。

- 4-通道扫描模式
- 8-通道扫描模式
- 12-通道扫描模式

(1) 固定通道单转换模式。

当ADMOD0<REPEAT, SCAN>被设置为"00"时，会在固定的通道单转换模式下执行AD转换。

在该模式下，会为每个所选通道执行一次AD转换。在AD转换完成之后，ADMOD0<EOCFN>即被设置为"1"，ADMOD0<ADBFN>即被清除为"0"，且会生成AD转换完成中断请求(INTAD)。<EOCFN>一旦被读取则清除为"0"。

(2) 通道扫描单次转换模式

如果ADMOD0 <REPEAT, SCAN>即被设置为"01"，并会在通道扫描单转换模式下执行AD转换。

在该模式下，会为每个所选通道执行一次AD转换。在AD扫描转换完成之后，ADMOD0<EOCFN>即被设置为"1"，ADMOD0<ADBFN>即被清除为"0"，且会生成AD转换完成中断请求(INTAD)。<EOCFN>一旦被读取则清除为"0"。

(3) 固定通道重复转换模式

当ADMOD0<REPEAT, SCAN>被设置为"10"时，在固定的通道重复会话模式下执行AD转换。

在该模式下，会为每个所选通道重复执行AD转换。在AD转换完成之后，ADMOD0<EOCFN>即被设置为"1"。ADMOD0<ADBFN>未被清除为"0"。其仍保留为"1"。通过将ADMOD0<ITM>设置为某种相关的设置，即可选择据以生成转换完成中断请求(INTAD)的计时。<EOCFN>即被用中断INTAD生成时的同一计时设置。

<EOCFN>一旦被读取则清除为"0"。

(4) 通道扫描重复转换模式

当ADMOD0<REPEAT, SCAN>被设置为"11"时，在固定的通道扫描重复模式下执行AD转换。在该模式下，会为每个所选的扫描通道重复执行AD转换。每完成一次AD扫描转换，ADMOD0<EOCFN>即被设置为"1"，还会生成转换完成中断请求(INTAD)。ADMOD0<ADBFN>未被清除为"0"。其仍保留为"1"。<EOCFN>一旦被读取则清除为"0"。

14.4.2.2 最高优先级 AD 转换

中断进行中的标准模拟数字转换，即可执行最高优先级AD转换。

无论ADMOD0<REPEAT,SCAN>的设置为何，均可自动选择固定的通道单转换。在启动操作的条件得到满足时，可为ADMOD2<HPADCH>指定的通道仅执行一次转换。在转换完成时，最高优先级AD转换完成中断(INTADHP)即被生成，且ADMOD2<EOCFHP>（用以指示AD转换的完成）即被设置为"1"。<ADBFHP>恢复为"0"。一经读取，EOCFHP标志即被清除为"0"。

对最高优先级AD转换操作时激活的最高优先级AD模拟数字转换，则不予理会。

14.4.3 AD 监控功能

有两个AD监控程序功能通道。

如将ADMOD3<ADOBSV0>和ADMOD5<ADOBSV1>设置为"1"，则AD监控器功能即被使能。如果ADMOD3 <ADREGS0>和ADMOD5 <ADREGS1>所指定的转换结果寄存器的值变得大于或小于(拟由ADMOD3 <ADOBIC0>和ADMOD5 <ADBIC1>指定"大于"或 "小于")某个比较寄存器的值，则AD监控器功能中断 (INTADM0, INTADM1)即被生成。每当有一个结果被存储到相应的转换结果寄存器中，比较操作即被执行一次。

如果分配执行AD监控程序功能的转换结果寄存器连续使用而不读取转换结果，其转换结果即被覆盖。转换结果储存标志<ADR_xRF>与超限标志<OVR_x

14.4.4 选择输入通道

选择输入通道的方式，会随着拟采用AD转换器操作模式的不同而不同。

1. 正常AD转换模式

- 如果模拟输入通道被用于某固定状态(ADMOD0<SCAN> = "0")
通过将ADMOD1<ADCH>设置为某种合适的设定，即可从各模拟输入引脚中选择一个通道。
- 如果模拟输入通道被用于某扫描状态(ADMOD0<SCAN>="1")
通过将ADMOD1 <ADCH>和<ADSCN>设置为某种合适的设定，即可从各扫描模式中选择一种扫描模式。

2. 最优先级AD转换模式

通过将ADMOD2<HPADCH>设置为某种合适的设定，即可从各模拟输入引脚中选择一条通道。

14.4.5 AD 转换详细描述

14.4.5.1 启动 AD 转换

将ADMOD0<ADS>设置到"1"来激活标准AD转换。将ADMOD2<HPADCE>设置至"1"来激活最高优先级AD转换。

四种操作模式可适用于标准AD转换。在执行标准AD转换时，必须将ADMOD0<REPEAT,SCAN>某个合适的设定，然后选择这些操作模式的其中之一。对于最高优先级AD转换而言，仅可采用一种操作模式：固定通道单转换模式。

可利用ADMOD4 <ADHS>所选择的H/W激活源激活标准AD转换，并可利用ADMOD4 <HADHS>所选择的HW激活源激活最高优先级AD转换。如果<ADHS>和<HADHS> 的各位均为"0"，则可根据某个下降沿通过 **ADTRG** 引脚进行的输入激活标准和最高优先级AD转换。如果这些位为"1"，则可根据计时器的匹配检测激活转换。

如需允许H/W激活，则可将ADMOD4 <ADHTG>设置为"1"（适用于标准AD转换），并将ADMOD4 <HADHTG>设置为"1"（适用于最高优先级AD转换）。

即使在H/W激活已被允许之后，软件仍然有效。

注1：某些产品无 **ADTRG** 引脚。

注2：在某外触发器被用于某最高优先级AD转换的HW激活源时，外部触发器不能被设置为“激活标准AD转换HW”。

注3：详见“产品信息”以确定计时器的可用匹配检测。

14.4.5.2 AD 转换

在标准AD转换开始时，显示AD转换正在进行的模拟数字转换忙标志(ADMOD0<ADBFN>)被设置到"1"。

在最高优先级AD转换启动时，最高优先级AD转换忙标志(ADMOD2 <ADBFHP>)（用以指示AD转换正在进行中）即被设置为"1"。

此时，会保留标准AD转换的忙标志ADMOD0<ADBFN>在最高优先级AD转换启动之前的值。还会保留标准AD转换的转换完成标志ADMOD0<EOCFN>在最高优先级AD转换启动之前的值。

注：当最高优先级AD转换正在进行时，不可激活标准AD转换。

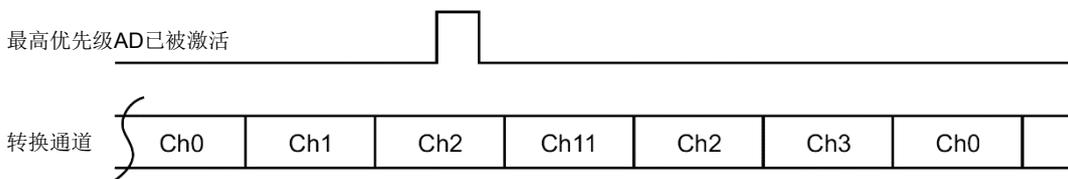
14.4.5.3 在标准AD转换期间进行的最高优先级AD转换

如果最高优先级AD转换已在标准AD转换期间被激活，则进行中的标准AD转换会被暂停，并会在最高优先级AD转换完成之后重新开始标准AD转换。

如果ADMOD2<HPADCE>在标准AD转换期间被设置为"1"，则进行中的标准AD转换会被暂停，最高优先级AD转换启动；具体地说就是可为ADMOD2<HPADCH>所指定的某个通道执行AD转换(固定通道单转换)。在最高优先级AD转换的结果被存储到存储寄存器ADREGSP中之后，标准AD转换即被继续执行。

如果最高优先级AD转换的H/W激活在标准AD转换期间得到许可，进行中的AD转换则在使用H/W激活源的激活要求满足时中止，则开始为ADMOD2<HPADCH>分派的通道进行最高优先级AD转换(固定通道单次转换)。在最高优先级AD转换的结果被存储到存储寄存器ADREGSP中之后，标准AD转换即被继续执行。

例如，如果为AIN0至AIN3激活了通道重复转换，且如果<HPADCE>在AIN2转换期间被设置为"1"，则AIN2转换即被暂停，并为<HPADCH> (AIN11如以下个案所示)所指定的某个通道执行转换。在转换结果被存储到ADREGSP中之后，继续从AIN2开始执行通道重复转换。



14.4.5.4 停止重复转换模式

将"0"写入到ADMOD0<REPEAT>，即可在转换模式(固定通道重复转换模式或通道扫描转换模式)下停止AD转换操作。在进行中的AD转换完成时，重复转换模式即终止，且ADMOD0<ADBFN>即被设置为"0"。

14.4.5.5 重新激活标准AD转换

如需在转换正在进行期间重新激活标准AD转换，则在AD转换开始之前必须执行软件复位(ADMOD3<ADRST>)。不得将H/W激活法用于重新激活标准AD转换。

14.4.5.6 转换完成

(1) 标准AD转换完成

在AD转换完成时，AD转换完成中断(INTAD)即被生成。AD转换的结果被保存到该存储寄存器中，且两个寄存器也会变为：寄存器ADMOD0 <EOCFN>（用于指示AD转换的完成情况）和寄存器ADMOD0 <ADBFN>。

有关详细资料，请参看表14-2和表14-3以确认与转换模式对应的存储寄存器。
中断请求与标记变更如下所示。

• 固定通道单次转换模式

在AD转换完成之后，ADMOD0<EOCFN>即被设置为"1"，ADMOD0<ADBFN>即被清除为"0"，中断请求即被生成。

• 通道扫描单次转换模式

在通道扫描转换完成之后，ADMOD0<EOCFN>即被设置为"1"，ADMOD0<ADBFN>即被清除为"0"，中断请求INTAD即被生成。

• 固定通道重复转换模式

ADMOD0<ADBFN>不清除至"0"。其仍保留为"1"。通过将ADMOD0<ITM> 设置为某合适的设定，即可选择生成中断请求INTAD时所使用的计时。ADMOD0<EOCFN>即被用中断INTAD生成时的同一计时设置。

• 通道扫描重复转换模式

每完成一次AD扫描转换，ADMOD0 <EOCF>即被设置为"1"，中断请求INTAD即被生成。ADMOD0<ADBFN>未被清除为"0"。其仍保留为"1"。

(2) 最高优先级AD转换完成

在AD转换完成之后，最高优先级AD转换完成中断(INTADHP)即被生成，且ADMOD2<EOCFHP>（用以指示最高优先级AD转换的完成情况）即被设置为"1"。
AD转换结果即被存储在AD转换结果寄存器SP中。

(3) 数据查询

利用数据查询，即可在不使用中断的情况下确认AD转换的完成情况。在AD转换完成时，ADMOD0<EOCFN>被设置到"1"。通过查询该位，即可确认AD转换的完成情况，并获得结果。

必须通过半字或整字存取读出转换结果存储寄存器。如果<OVRx> = "0"、且 <ADR_xRF> = "1"，则表明所得出的转换结果是正确的。

14.4.5.7 中断生成计时和 AD 转换结果存储寄存器

表14-1给出了以下三项的关系：AD转换模式、中断生成计时和标志操作。表14-2和表14-3显示了模拟通道输入和AD转换结果寄存器之间的关系。

表14-1 转换模式、中断生成计时和标志操作之间的关系

转换模式		扫描/重复模式设置			中断生成计时	ADMOD0<EOCFN>/ ADMOD2<EOCFHP> 设置计时 (注)	ADMOD0	ADMOD2
		ADMOD0 <REPEAT>	ADMOD0 <SCAIN>	ADMOD0 <ITM[1:0]>			<ADBFN> (在该中断被生成之后)	<ADBFHP>
标准转换	固定通道 单次转换	0	0	-	在生成完成之后	完成转换之后。	0	-
	固定通道 重复转换	1	0	00	每完成一次转换。	完成一次转换之后。	1	-
				01	每次完成四次转换时。	完成四次转换之后。	1	-
				10	每次完成八次转换时。	完成八次转换之后。	1	-
	通道扫描 单次转换	0	1	-	在扫描转换完成之后。	完成扫描转换之后。	0	-
通道扫描 重复转换	1	1	-	在一次扫描转换完成之后。	完成一次扫描转换之后。	1	-	
最高优先级转换		-	-	-	在转换完成之后。	完成转换	-	0

注：ADMOD0<EOCFN>与ADMOD2<EOCFHP>一经读取即清除。

表14-2 结果寄存器 (固定-通道重复转换模式)

<ITM[1:0]>	结果寄存器
00 每 1 次转换，生成一次中断	ADREG0
01 每 4 次转换，生成一次中断	ADREG0~ADREG3
10 每 8 次转换，生成一次中断	ADREG0~ADREG7

表14-3 结果寄存器(固定-通道重复转换模式除外)

ADMOD1 <ADCH[3:0]>	ADMOD0 <SCAN>=0		ADMOD0 <SCAN>=1					
	固定通道		<ADSCN>=00 4-通道扫描		<ADSCN>=00 8-通道扫描		<ADSCN>=00 12-通道扫描	
	转换通道	结果寄存器	转换通道	结果寄存器	转换通道	结果寄存器	转换通道	结果寄存器
0000	AIN0	ADREG0	AIN0	ADREG0	AIN0	ADREG0	AIN0	ADREG0
0001	AIN1	ADREG1	AIN0~AIN1	ADREG0~ADREG1	AIN0~AIN1	ADREG0~ADREG1	AIN0~AIN1	ADREG0~ADREG1
0010	AIN2	ADREG2	AIN0~AIN2	ADREG0~ADREG2	AIN0 ~ AIN2	ADREG0~ADREG2	AIN0~AIN2	ADREG0~ADREG2
0011	AIN3	ADREG3	AIN0~AIN3	ADREG0~ADREG3	AIN0 ~ AIN3	ADREG0~ADREG3	AIN0~AIN3	ADREG0~ADREG3
0100	AIN4	ADREG4	AIN4	ADREG4	AIN0 ~ AIN4	ADREG0~ADREG4	AIN0~AIN4	ADREG0~ADREG4
0101	AIN5	ADREG5	AIN4~AIN5	ADREG4~ADREG5	AIN0 ~ AIN5	ADREG0~ADREG5	AIN0~AIN5	ADREG0~ADREG5
0110	AIN6	ADREG6	AIN4~AIN6	ADREG4~ADREG6	AIN0 ~ AIN6	ADREG0~ADREG6	AIN0~AIN6	ADREG0~ADREG6
0111	AIN7	ADREG7	AIN4~AIN7	ADREG4~ADREG7	AIN0 ~ AIN7	ADREG0~ADREG7	AIN0~AIN7	ADREG0~ADREG7
1000	AIN8	ADREG0	AIN8	ADREG0	AIN8	ADREG0	AIN0~AIN8	ADREG0~ADREG8
1001	AIN9	ADREG1	AIN8~AIN9	ADREG0~ADREG1	AIN8 ~ AIN9	ADREG0~ADREG1	AIN0~AIN9	ADREG0~ADREG9
1010	AIN10	ADREG2	AIN8~AIN10	ADREG0~ADREG2	AIN8 ~ AIN10	ADREG0~ADREG2	AIN0~AIN10	ADREG0~ADREG10
1011	AIN11	ADREG3	AIN8~AIN11	ADREG0~ADREG3	AIN8 ~ AIN11	ADREG0~ADREG3	AIN0~AIN11	ADREG0~ADREG11
1100	AIN12	ADREG4	AIN12	ADREG4	AIN8 ~ AIN12	ADREG0~ADREG4	-	-
1101	AIN13	ADREG5	AIN12~AIN13	ADREG4~ADREG5	AIN8 ~ AIN13	ADREG0~ADREG5	-	-
1110	AIN14	ADREG6	AIN12~AIN14	ADREG4~ADREG6	AIN8 ~ AIN14	ADREG0~ADREG6	-	-
1111	AIN15	ADREG7	AIN12~AIN15	ADREG4~ADREG7	AIN8 ~ AIN15	ADREG0~ADREG7	-	-

14.4.5.8 DMA 请求

在AD转换完成中断或AD监控器功能中断生成的计时点，一个DMA请求被发送到DMAC。

在执行DMA传输时，对拟由ADMOD6寄存器相应位使能的DMA请求进行设置。

14.4.5.9 注意事项

注意事项

AD转换的结果值可随电源电压的波动而变化，也可受到杂波的影响。

在交替使用各模拟输入引脚和端口时，不要在转换期间读取和写入端口，以免因此而降低转换精度。此外，如果输出端口电流在AD转换期间发生波动，则也可能导致转换精度被降低。

请利用此程序采取对应措施，如对AD转换结果进行平均化。



15. 低电压检测电路(LVD)

通过检测正在下降的电压，低电压检测电路生成复位或中断(INTLVD)。

注：INTLVD为非可屏蔽中断(NMI)的系数。

15.1 配置

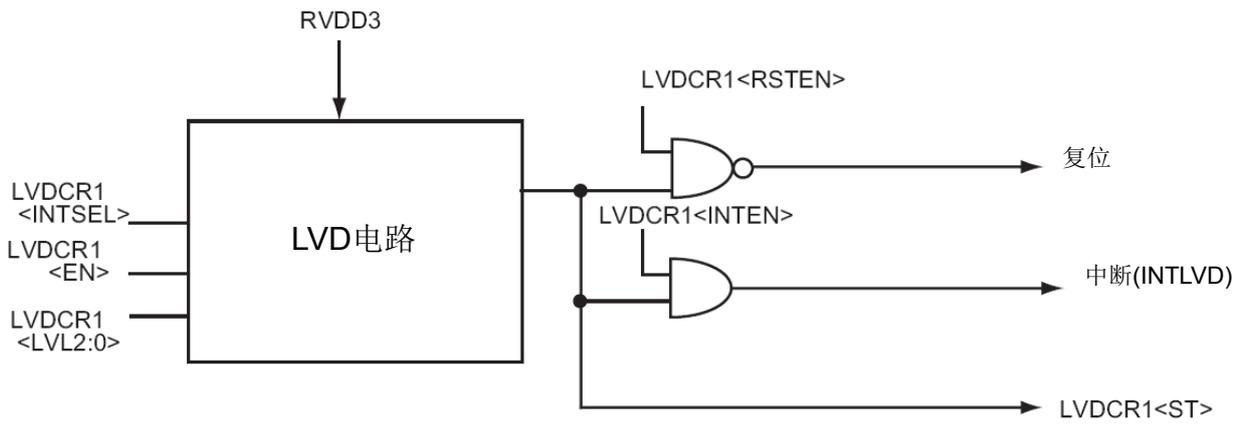


图15-1 LVD方块图

15.2 寄存器

有关基准地址，请参看“存储器地址”一节的“外设功能的地址列表”。

15.2.1 寄存器列表

寄存器名称	地址(基+)
保留	0x0000
LVD检测控制 1	LVDCR1 0x0004

15.2.2 LVDCR1 (LVD 检测控制寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ST	RSTEN	INTEN	INTSEL	LVL			EN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31 - 8	-	R	读作 0。
7	ST	R	LVL电压检测状态 0: 电源电压等于或高于检测电压。 1: 电源电压等于或低于检测电压。
6	RSTEN	R/W	控制RESET输出 0: 禁用 1: 启用
5	INTEN	R/W	控制INTLVD输出 0: 禁用 1: 启用
4	INTSEL	R	中断生成条件。 0: 在电压下降时仅低于设置电压。 1: 在电压下降时，低于及高于设置电压。 位可与<RSTEN>="0"和<INTEN>="1"配套使用。
3 - 1	LVL[2:0]	R/W	3 V电源检测电压 1 000: 保留 001: 保留 010: 保留 011: 2.5 ± 0.2 V 100: 2.6 ± 0.2 V 101: 2.7 ± 0.2 V 110: 2.8 ± 0.2 V 111: 2.9 ± 0.2 V
0	EN	R/W	电压检测操作 0: 禁用 1: 启用

注1: 通过上电复位和端子复位对LVDCRn进行初始化。

注2: 在LVD和解除电压之间不存在滞后现象。检测期间可能出现颤振现象, 视电源的斜率而定。

15.3 操作

15.3.1 选择检测电压及使能电压检测操作

在通过设置寄存器LVDCRn<LVL [2:0]>来选中拟检测的电压、且"1"被设置到LVDCRn<EN>之后，电压检测即被使能。

15.3.2 通过检测电源电压复位

在"1"被设置到LVDCRx<RSTEN>、且电源电压降低到设定检测电压以下时，即发生复位。

检测到电压下降量并生成复位需耗时约 100 μ s。如果电源电压下降到所检测到电压的时段比较短，则复位可能不会发生。

15.3.3 通过检测电源电压实现中断

在"0"被设置到LVDCEn<RSTEN>、且"1"被设置到LVDCRn<INTEN>时，即发生一次中断(INTLVD)

如果电源电压下降到设置检测电压电平以下或以上。

可通过LVDCRn<INTSEL>设置中断条件。

在"0"被设置到LVDCRx<INTSEL>、且电源电压下降到所设置的检测电压以下时，会发生一次中断。

在"1"被设置到LVDCRx <INTSEL >、且电源电压下降到所设置的检测电压以下或以上时，会发生一次中断。

检测到电压下降并生成一次中断需耗时约 100 μ s。如果电源电压下降到所检测到电压的时段比较短，则中断可能不会发生。

15.3.4 检测状态

读取LVDCRn<SR>时，可确认低电压检测的检测状态。

在LVDCRn<ST>为"0"时，电压等于或高于检测电压。

在LVDCRn<ST>为"1"时，电压等于或低于检测电压。

在低电压检测(LVD)的中断服务例行程序(ISR)中，在检测后读取LVDCRn<ST>的状态必定会导致发生电压移位

16.2 寄存器

16.2.1 寄存器列表

下表显示控制寄存器及其地址。

有关基地址的详细情况，参照“内存印象图”一章的“外部功能地址表”。

寄存器名称		地址(基+)
看门狗计时器寄存器地址	WDMOD	0x0000
看门狗计时器控制寄存器	WDCR	0x0004

16.2.2 WDMOD(看门狗计时器模式寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	WDTE	WDTP			-	I2WDT	RESCR	-
复位后	1	0	0	0	0	0	1	0

位	比特符号	型号	功能
31-8	-	R	读作"0"。
7	WDTE	R/W	启用/禁用控制 0:禁用 1:启用 禁用看门狗计时器保护故障进行错误写入时，则需将第一个<WDTE>设置为"0"，然后必须停用代码(0xB1)写入到WDCR。 将<WDTE>设置为"1"，即可将看门狗计时器的状态从"禁用"改为"启用"。
6-4	WDTP[2:0]	R/W	选择WDT检测时间 000:2 ¹⁵ /fsys 100:2 ²³ /fsys 001:2 ¹⁷ /fsys 101:2 ²⁵ /fsys 010:2 ¹⁹ /fsys 110:保留 011:2 ²¹ /fsys 111:保留
3	-	R	读作"0"。
2	I2WDT	R/W	IDLE模式操作。 0:停止 1:操作
1	RESCR	R/W	在检测到故障后操作 0:INTWDT中断请求即被生成。注： 1:复位
0	-	R/W	写作"0"。

注：INTWDT为非屏蔽中断的系数。

16.2.3 WDCR(看门狗计时器控制寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作"0"。
7-0	WDCR	W	停用/清除代码 0xB1: 停用代码 0x4E: 清除编码 其它: 保留

16.3 操作描述

16.3.1 基本操作

看门狗计时器由二进制计数器组成，二进制计数器工作时将系统时钟(fsys)用作输入。检测时间由WDMOD<WDTP[2:0]>在 2^{15} 、 2^{17} 、 2^{19} 、 2^{21} 、 2^{23} 和 2^{25} 之间选择。

在所规定的检测时间消逝后，看门狗计时器中断(INTWDT)即生成，且看门狗计时器输出引脚($\overline{\text{WDTOUT}}$)输出“低”。

如需检测杂波或其它干扰所引起的CPU故障(溢出)，则应在INTWDT中断被生成之前，通过软件指令清除看门狗计时器的二进制计数器。如果二进制计数器未被清除，则INTWDT即生成非可屏蔽中断。因此，CPU检测到故障(失控)，执行故障对策程序，并恢复正常运行。

此外，可能通过连接看门狗计时器输出引脚和外设复位引脚而解决CPU故障(失控)问题。

注：TMPM037FWUG无看门狗计时器输出引脚($\overline{\text{WDTOUT}}$)。

16.3.2 操作模式和状态

在某个复位被解除后，看门狗计时器立即开始操作。如果不使用看门狗计时器，则其应被禁用。

在高速频率时钟被停止时，无法使用看门狗计时器。在转换到以下操作模式之前，看门狗计时器应被禁用。

在IDLE模式下，其操作取决于WDMOD<I2WDT>设置。

- STOP1 模式

而且，调试模式下，二进制计数器自动停止。

16.3.3 在检测到故障(失控)时的操作情况

16.3.3.1 INTWDT 中断生成

图 16-2 显示已生成INTWDT中断的情形(WDMOD<RESCR>="0")。

在二进制计数器发生溢出时，INTWDT中断即被生成。其为非可屏蔽中断(NMI)的系数。因此，CPU可检测到非可屏蔽中断，并执行对策程序。不可屏蔽中断因素为复数。CGNMIFLG可识别非可屏蔽中断的系数。如果是INTWDT，则需设置CGNMIFLG<NMIFLG0>。

在INTWDT中断被生成时，看门狗计时器输出($\overline{\text{WDTOUT}}$)同时输出“低”。

通过执行看门狗计时器清除（即将清除代码 0x4E 写入到WDCR）， $\overline{\text{WDTOUT}}$ 变为“高”。

注：TMPM037FWUG不具备看门狗计时器输出引脚($\overline{\text{WDTOUT}}$)。

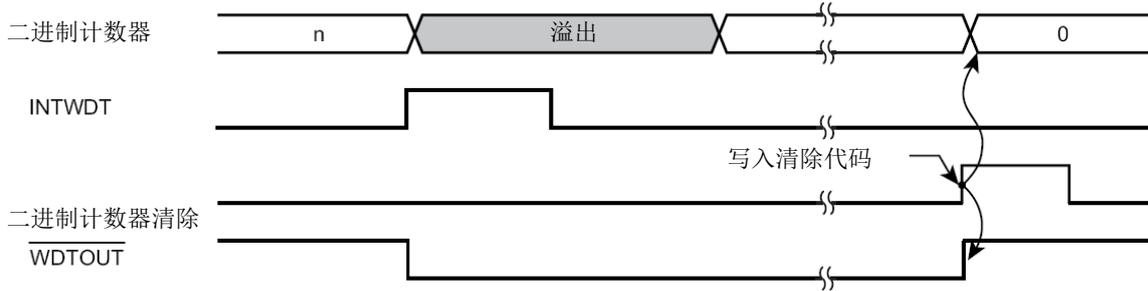


图 16-2 INTWDT 中断生成

16.3.3.2 内部复位生成

图 16-3 给出了内部复位生成(WDMOD<RESCR>="1")。

MCU通过二进制计数器的溢出而复位。在这种情况下，对于 32 种状态，复位状态继续。

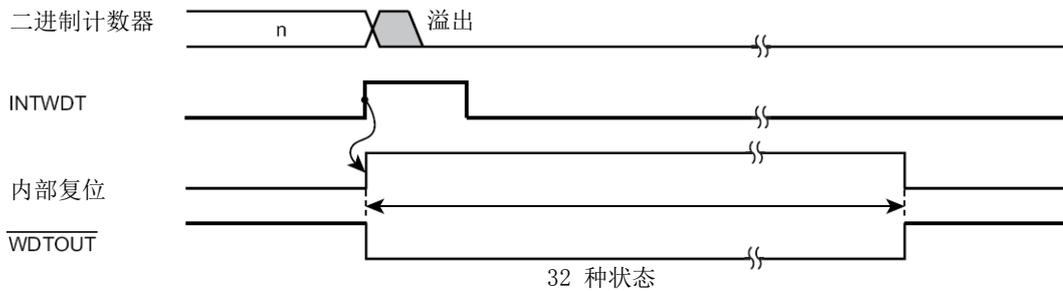


图 16-3 内部复位生成

16.4 看门狗计时器的控制

16.4.1 禁用控制

通过在将WDMOD<WDTE>设置为"0"之后，将禁用代码(0xB1)写入到WDCR，即可禁用看门狗计时器，并清除二进制计数器。

16.4.2 启用控制

将WDMOD<WDTE>设置为"1"。

16.4.3 看门狗计时器清除控制

通过将清除代码(0x4E)写入到WDCR，即可解除二进制计数器，且其重新开始计数。

16.4.4 看门狗计时器的检测时间

根据检测时间设置WDMOD<WDTP[2:0]>。

例如，在使用了 $2^21/f_{SYS}$ 的情况下，可将"011"设置到WDMOD<WDTP[2:0]>。



17. 闪存操作

本节主要对硬件配置与闪存的操作进行描述。在本节中，“1-字”表示 32 位。

17.1 特征

17.1.1 存储器容量和配置

表 17-1 和图 17-1 给出了内置存储器容量和TMPM037FWUG的配置。

表 17-1 存储器容量和配置

存储器容量	存储块配置				每页字数#	页数#	写入时间		擦除时间	
	128 KB	64 KB	32 KB	16 KB			1 页	整个区域	存储块擦除	芯片擦除
128 KB	-	-	4	-	32	1024	1.25 ms	1.28 sec	0.1 sec	0.2 sec

注：以上各值均为不计数据传输时间的理论值。每个芯片的写入时间，取决于用户所采用的写入方式。



图 17-1 块配置

闪存配置单元被称为“存储块”和“页面”。

- 页面
一个页面是 32 个字。单个页面内使用同一地址[31:7]。组的第一个地址为[6:0]=0，组的最后一个地址为[6:0] = 0x7F。
- 存储块
一个存储块为 32 KB，一个闪存存储器由四个存储块组成。

每个页面都会进行写入操作。每个页面的写入时间为 1.25 ms(典型值)。

每块均进行擦除(使用自动块擦除命令)或整个闪存均进行擦除(使用自动片擦除命令)。擦除时间随命令不同而变化如使用自动存储块命令，则擦除时间将会是每个存储块0.1 sec(典型值)。如使用该自动芯片擦除命令来擦除整个区域，则该时间会是 0.2 sec(典型值)。

此外，每个存储块都可使用保护功能。有关该保护功能的详细描述，请参看”17.1.5 保护/安全功能”。

17.1.2 功能

除了某些特定功能之外，闪存内置装置一般可满足JEDEC标准的要求。因此，如果用户当前是将闪存用作外部存储器的话，则易于对该装置执行该功能。此外，为便于进行写入或擦除操作，本产品还带有可自动执行写入或芯片擦除的专用电路。

符合JEDEC要求的功能	已被修改,新增或删除的功能
<ul style="list-style-type: none"> · 自动编程 · 自动芯片擦除 · 自动存储块擦除 · 数据查询/切换位 	<p><已被修改>存储块写入/擦除保护(仅支持软件保护)</p> <p><已被删除>擦除恢复-暂停功能</p>

17.1.3 操作模式

17.1.3.1 描述

本装置可提供单芯片模式和单引导模式。该单芯片模式包含标准模式和用户引导模式。图 17-2 给出了该模式转换。

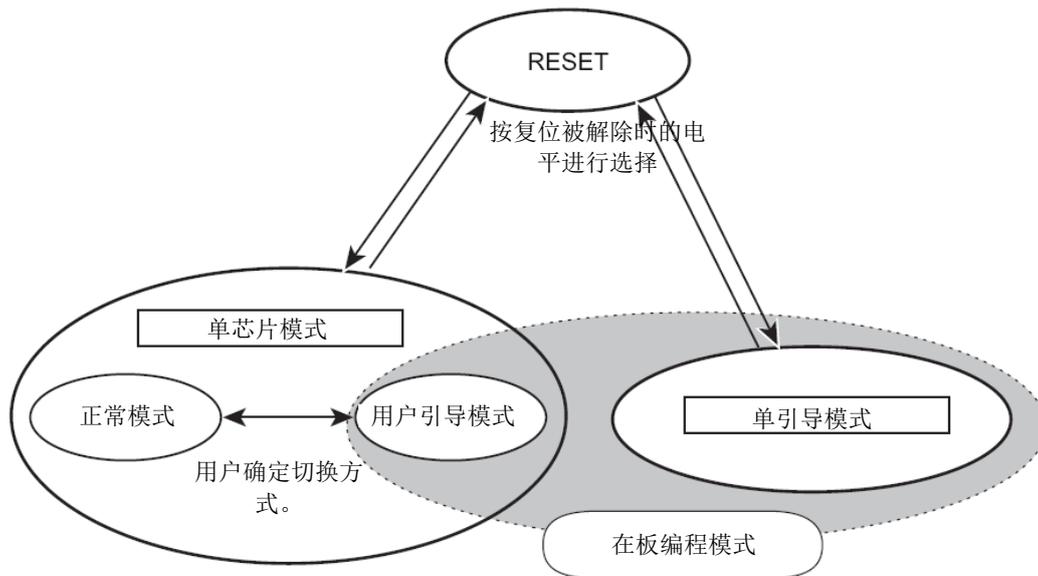


图 17-2 模式转换

(1) 单芯片模式

在单芯片模式下，该装置可在复位之后从闪存引导启动。该模式包含以下两种子模式。

标准模式

在该模式下，可执行用户应用程序。

用户引导模式

在该模式下，可按用户设置对闪存进行重新编程。

用户可从该标准模式自由切换为用户引导模式。例如，用户可做如下设置：如果端口A的PA0为“1”，则为标准模式。如果端口A的PA0为“0”，则为用户引导模式。用户必须在该应用程序中备好例行程序，用以确定切换。

(2) 单引导模式

在该模式下，闪存可在复位之后从内置BOOT ROM(掩模ROM)引导。

该BOOT ROM芯片所提供的算法，可按照用户的设置，通过本装置的串行端口对闪存进行重写。在已将串行端口连接到外部主机的情况下，即可在上述协议和重新编程的闪存存储器中执行数据传输。

(3) 在板编程模式

在该用户引导模式和单引导模式下，可按照用户的设置对闪存进行重新编程。这两种模式被称作“在板编程模式”。

17.1.3.2 模式确定

单片或单个启动操作模式在重置释放时均可依据 $\overline{\text{BOOT}}$ 引脚的电平来选择。

表 17-2 操作模式设置

操作模式	引脚	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
单芯片模式	0 → 1	1
单引导模式	0 → 1	0

17.1.4 存储器映象

图 17-3 给出了存储器地址在单芯片模式和单引导模式下的比较结果。在单引导模式下，内置闪存会被地址至 0x3F80_0000 和后续的地址，且该内置引导ROM芯片会通过 0x0000_0FFF 被地址至 0x0000_0000。

闪存和RAM地址如以下所示。

FLASH容量	RAM容量	FLASH地址	RAM地址
128 KB	16 KB	0x0000_0000~0x0001_FFFF(单芯片模式) 0x3F80_0000~0x3F81_FFFF(单引导模式)	0x2000_0000 ~ 0x2000_3FFF

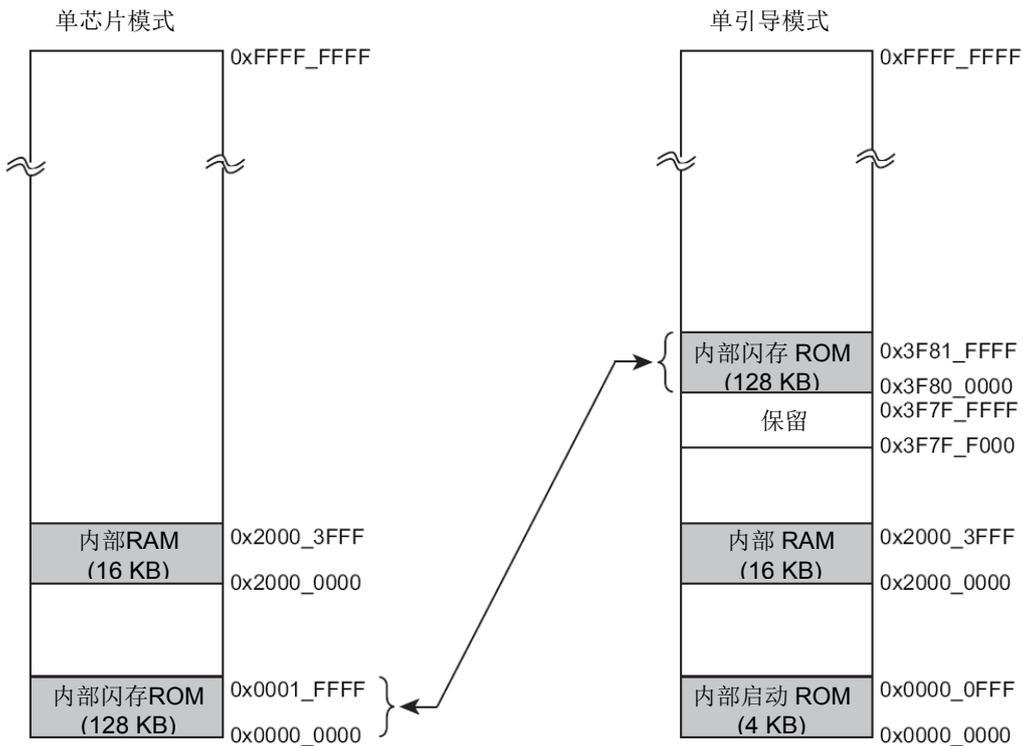


图 17-3 存储器地址的比较

17.1.5 保护/安全功能

该装置具备闪存保护和安全管理功能。

1. 保护功能
可按存储块禁止的写入/擦除操作。
2. 安全功能
可禁止从闪存编写程序进行读出操作。
使用调试功能限制

17.1.5.1 保护功能

该功能可按存储块禁止写入/擦除操作。

用该保护位程序命令，将某个存储块对应的某个保护位设为“1”，即可使能该保护功能。当利用保护位擦除命令将某保护位设置为“0”时，可取消块保护。可用FCPSRA<BLK[3:0]>监控保护位。

可通过1-位单元对保护位的某程序进行编程，并可通过4-位单元擦除保护位的某个程序。有关保护位编程/擦除的详细描述，请参看“17.2.5 命令描述”。

17.1.5.2 安全功能

表 17-3 给出了安全功能被使用时的操作步骤。

表 17-3 安全功能被使用时的操作步骤

项目	描述
读取闪存	CPU可读取闪存。
调试端口	JTAG、串行线或跟踪通信已被禁用。
对闪存执行命令	“写入到闪存”的命令未被接受。如果用户尝试擦除某个保护位，则会执行芯片擦除，且所有保护位均会被擦除。

在以下条件下，安全功能可被使用：

1. 将FCSECBIT<SECBIT>设置为“1”。
2. 将所有保护位(FCPSRA<BLK>)设置为“1”。

通过上电复位将FCSECBIT<SECBIT> 设置为“1”。FCSECBIT <SECBIT>的重写描述如下。

注：在以下写入操作项目 1 和 2 时，可使用 32-位传送指令。

1. 将指定代码(0xa74a9d23)写入到FCSECBIT
2. 项目1的操作结束后在 16 个时钟脉冲内写入数据。

17.1.6 寄存器

17.1.6.1 寄存器列表

下表给出了各控制寄存器和地址。

有关基地址的详细资料，请参看“存储器地址”一节的“外设功能的地址列表”。

寄存器名称		地址(基+)
安全位寄存器	FCSECBIT	0x0010
快闪状态寄存器	FCSR	0x0020
快闪保护状态寄存器A	FCPSRA	0x0030

17.1.6.2 FCSR(闪存状态寄存器)

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

位	比特符号	型号	功能
31-1	-	R	读作"0"。
0	RDY_BSY	R	准备就绪/忙(注1) "0"忙(在自动操作期间) "1"准备就绪(自动操作结束) 该位属于功能位，用于从CPU监控闪存。在闪存自动操作期间，如该位输出"0"则表明闪存忙。{一旦自动操作完成，该位变为就绪状态并输出"1"。此时，下一条命令即被接受。 若自动操作的结果不合格，则该位连续输出"0"。该位通过硬件复位返回为"1"。

注1：在发出命令之前，应确认闪存已准备就绪。当在忙碌期间发出某命令时，不仅命令不会被发送，而且后续的命令也不会被接受。在这种情况下，可利用硬件复位实现返回。无论系统时钟为何，硬件复位均需花费 0.5 μs 或以上的复位周期。此时，需耗时约 2 ms 才能在复位后启用读取。

17.1.6.3 FCSECBIT(安全位寄存器)

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

位	比特符号	型号	功能
31-1	-	R	读作"0"。
0	SECBIT	R/W	安全位 "0"安全功能设置已被禁用。 "1"安全功能设置已被启用。

注：由PowerOnReset对寄存器进行初始化。

17.1.6.4 FCPSRA(闪存保护状态寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	BLK3	BLK2	BLK1	BLK0
复位后	0	0	0	0	(注1)	(注1)	(注1)	(注1)

位	比特符号	型号	功能
31-4	-	R	读作"0"。
3-0	BLK3- BLK0	R	存储块 3~0 的保护状态 "0"不受保护 "1"受保护 保护位值与各存储块的保护状态对应。如果对应的位指示"1"，则表明对应的存储块处于该保护状态。 处于该保护状态的存储块是不能重新编程的。

注1：各值应与保护状态对应。

17.2 闪存的详细描述

在进行在板编程时，CPU可执行闪存重新编程或擦除命令。应由用户预先备好该重新编程/擦除控制程序。在闪存正在被写入或擦除期间不能读取闪存内容，因此，需在内置RAM上运行该重新编程/擦除控制程序。除复位外，不要生成中断/故障，以避免发生程序异常终止现象。

17.2.1 功能

除了某些特定功能之外，闪存一般而言均可满足JEDEC标准的要求。不过，操作命令地址命名方法不同于标准命令。

当执行写入/擦除操作时，用 32-位(1-字)存储指令命令将各命令输入到闪存中。在命令输入之后，会在内部自动执行写入或擦除操作。

表17-4 闪存功能

主要功能	描述
自动页面程序	自动写入数据。
自动芯片擦除	自动擦除闪存的整个区。
自动存储块擦除	自动擦除某个选定存储块。
写入/擦除保护	可分别禁止各存储块的写入或擦除操作。

注：检查FCSR<RDY_BSY>，以确认各命令时序终止，例如闪存写入、闪存擦除、保护位程序、保护位擦除。然后保持 200 μ s或以上，其后再从闪存存储器读取数据或开始指令取出。

17.2.2 闪存的操作模式

闪存主要有两种操作模式：

- 读取存储器数据的模式(读取模式)
- 自动擦除或重写存储器数据的模式(自动操作模式)

在接通电源，休眠或自动运行模式之后，闪存转变为读取模式。存储在闪存中的指令或数据的读取，均在读取模式执行。

如果在读取期间输入命令，则操作模式会变为自动操作模式。如果命令处理是以标准方式完成的，则操作模式会返回到读取模式(ID读命令除外)。在自动操作期间，不能执行数据读取以及存储在闪存中的指令。

如果命令过程的完成异常，则应以强制方式使操作模式返回到读取模式。在这种情况下，可使用该读取命令、读取/复位命令或硬件复位。

17.2.3 硬件复位

硬件复位表示在自动编程/擦除操作被强制取消、自动操作异常终止时，利用上电复位或热复位返回到读取模式。

如果在自动操作期间发生硬件复位，则闪存会停止自动操作，并返回到读取模式。如果在闪存自动编程/擦除操作期间生成硬件复位，则无论系统时钟为何，硬件复位的复位周期均需耗时 0.5 μ s或以上。此时，在复位之后需耗时约 2 ms才能使用读取。注意，如果在自动操作期间发生硬件复位，则不能正确执行数据写入操作。重新设置写入操作。

有关复位操作的详细描述，请参看“复位”。在给定的复位输入完成之后，CPU将读取复位向量数据，然后在复位后启动例行程序。

17.2.4 如何执行命令

在执行命令时，需使用存储指令将各命令时序写入到闪存中。闪存按照各输入地址和数据的组合，执行各自动操作命令。有关命令执行的详细描述，请参看“17.2.5 命令描述”。

对闪存执行存储指令，被称为“总线写入周期”。各命令均包含某些总线写入周期。在闪存中，在按所指定的顺序执行总线写周期的地址和数据时，执行自动命令操作。在按非指定顺序执行周期时，闪存会停止命令执行，并返回到读取模式。

如果用户在命令时序期间取消该命令或输入某个不同的命令时序，则会执行该读取命令或读取/复位命令。然后，闪存会停止命令执行，并返回到读取模式。该读取命令和读取/复位命令被称作“软件复位”。

在写入命令时序终止时，自动操作启动，FCSR<RDY_BSY> 被设置为“0”。在自动操作正常终止时，FCSR<RDY_BSY> = “1”即被设置，闪存会返回到读取模式。

在自动操作期间，不会接受新的命令时序。如果用户拟停止该命令操作，则可使用硬件复位。如果自动操作异常终止(FCSR<RDY_BSY>保持为“0”)，闪存保持在锁定状态，且不会返回到读取模式。利用硬件复位，即可返回到读取模式。如果硬件复位停止命令操作，这些命令则得不到正常执行。

关于命令执行的说明：

1. 识别命令时，命令时序发生器需在命令启动前处于读取模式。确认在各命令的首个总线写入周期之前，已设置为FCSR<RDY_BSY> = 1。建议连续执行该读取命令。
2. 从闪存外部执行各命令时序。
3. 由一字(32-位)中的数据传送指令，顺次执行各总线写周期。
4. 不要在各命令时序期间访问闪存。除复位外，不要生成任何中断或故障。
5. 如果任何地址或数据写入错误，则应在发出某命令之后，用软件复位确认已返回到读取模式。

17.2.5 命令描述

本节对各命令的内容进行说明。有关特定命令时序的详细资料，请参看“17.2.6命令时序”。

17.2.5.1 自动页面程序

(1) 操作描述

该自动页面程序可按页面写入数据。在该程序向多个页面写入数据时，需逐页执行某个页面命令。无法进行跨页面写入。

向闪存写入，意味着数据单元“1”变为数据“0”。无法从数据“0”变为数据单元“1”。需要执行擦除操作，才能从数据单元“0”变为“1”。

仅允许对已被擦除的各页面执行一次该自动页面程序。无论是数据单元“1”还是“0”，均不能被写入数据两次或以上。如拟对某个已被写入过一次的页面进行重写，则必须在自动存储块擦除或自动芯片擦除命令被执行之后，重新设置该自动页面程序。

注1：若在无擦除操作的情况下对同一页面执行页面程序两次或以上，则可能造成该装置受损。

注2：无法向受保护的存储块进行写入。

(2) 如何设置

第一至第三总线写入周期可指示该自动页面程序命令。

在第四个总线写入周期中，会写入该页面的第一个地址和数据。在第五个周期期间及之后，会顺次写入一个页面的数据。数据以一个字句单位(32位)写入。

如果部分页面被写入，则可将“0xFFFFFFFF”(其表示无需写入整个页面)为数据。

不会以内部方式在该装置中进行任何的自动验证操作。因此，必须读取已编程的数据，以确认其所写入正确。

如果该自动页面程序被异常终止，则该页面的写入即告失败。建议不使用装置，或不使用存储块(包括无效的地址)。

17.2.5.2 自动芯片擦除

(1) 操作描述

可对所有地址的存储器执行自动芯片擦除。如果内含受保护的存储块，则不会擦除这些存储块。如果所有的存储块都处于受保护状态，则不会执行自动芯片擦除操作，并将在某个命令时序被输入之后返回到读取模式。

(2) 如何设置

第一至第六总线写入周期可指示该自动芯片擦除命令。在输入该命令时序之后，即开始该自动芯片擦除操作。

不会以内部方式在该装置中进行任何的自动验证操作。因此，务必读取该数据，以确认其已被正确擦除。

17.2.5.3 自动存储块擦除

(1) 操作描述

该自动擦除命令可对指定的存储块执行擦除操作。如果该被指定的存储块处于受保护状态，则不会执行擦除操作。

(2) 如何设置

第一至第五总线写入周期可指示该自动存储块擦除命令。在第六个总线写入周期中，可指定拟被擦除的存储块。在输入命令时序之后，自动存储块擦除即开始。

不会以内部方式在该装置中进行任何的自动验证操作。因此，务必读取该数据，以确认其已被正确擦除。

17.2.5.4 自动保护位程序

(1) 操作描述

该自动保护位程序可一次将"1"写入到一个保护位。使用该自动保护位擦除命令，即可将"0"设置到某个保护位。

有关该保护功能的详细描述，请参看"17.1.5保护/安全功能"。

(2) 如何设置

第一至第六总线写入周期可指示该自动保护位程序命令。在第七个总线写入周期中，可指定拟被写入的保护位。在输入命令时序之后，该自动保护位程序即启动。检查写入操作是否以FCPSRA<BLK>正常终止。

17.2.5.5 自动保护位擦除

(1) 操作描述

该自动保护位擦除命令操作取决于安全状态。有关该安全状态的详细描述，请参看"17.1.5保护/安全功能"。

- 非安全状态

将指定保护位清除为"0"。在4-位单元中进行保护位擦除。

- 安全状态

在闪存的所有地址均被擦除之后，擦除所有保护位。

(2) 如何设置

第一至第六总线写入周期可指示该自动保护位擦除命令。在第七个总线写入周期中，可指定拟被擦除的保护位。在输入命令时序之后，该自动保护位擦除操作即开始。

在非安全状态时，会擦除所指定的保护位。检查擦除操作是否以FCPSRA<BLK>正常终止。

在安全状态下，闪存位的所有地址和所有保护都会被擦除。确认数据和保护位的擦除是否正常。如有必要，可执行自动保护位擦除、自动芯片擦除或自动存储块擦除。

所有情况均与其它命令相同，FCSR<RDY_BSY>在自动保护位擦除命令操作期间变为"0"。在操作完成之后，FCSR<RDY_BSY>变为"1"，闪存会返回其读取模式。只有硬件复位才能取消该操作。

17.2.5.6 ID-READ

(1) 操作描述

ID读取命令可读取信息(包括闪存类型和制造商代码、装置代码和宏代码等三种代码)。

(2) 如何设置

第一至第三总线写入周期可指示该ID读取命令。在第四个总线写入周期中，可指定拟被读取的代码。在第四个总线写入周期之后，在任意闪存区中进行读出操作时需取得相关代码。

该ID读取可连续执行。可重复执行第四个总线写入周期和ID值读取。

ID读取命令不会自动返回到读取模式。通过执行该读取命令、读取/复位命令或硬件复位，即可返回到读取模式。

17.2.5.7 读取命令和读取/复位命令(软件复位)

(1) 操作描述

利用该命令，可使闪存返回到读取模式。

在执行ID读取命令时，宏可停止在当前状态，且不会自动返回到读取模式。通过使用读取命令或读取/复位命令，可从这种情况返回到读取模式。其还可用于在命令输入期间，中途取消该命令。

(2) 如何设置

第一个总线周期可指示该读取命令。第一至第三总线写入周期可指示该读取/复位命令。在执行任一命令时序之后，闪存都会返回到读取模式。

17.2.6 命令时序

17.2.6.1 命令时序列表

表 17-5 给出了各命令中的总线写入周期的地址和数据。

除了ID读取命令的第五总线周期外，所有的命令周期均属于“总线写入周期”。由 32-位(1-字)数据传送指令执行总线写周期(下表仅给出了数据的下 8-位)。

有关地址的详细资料，请参看表 17-6。将以下值用于表 17-6 所列地址[15:9]栏中所述的“命令”。

注1) 应始终将“0”设置到地址位[1:0]。

注2) 根据闪存容量将以下值设置到地址位[19]。

存储器容量为 1 MB或以下：始终设置为“0”

存储器容量大于 1 MB：如果总线写入到1MB区或以下，则该位应被设置为“0”。

如果总线写入到 1 MB以上区，则将该位设置为“1”。

表 17-5 命令时序

命令	第一总线周期	第二总线周期	第三总线周期	第四总线周期	第五总线周期	第六总线周期	第七总线周期
	Addr.						
	Data						
读取	0xXX	-	-	-	-	-	-
	0xF0	-	-	-	-	-	-
读取/复位	0x55X	0xAAX	0x55X	-	-	-	-
	0xAA	0x55	0xF0	-	-	-	-
ID-READ	0x55X	0xAAX	0x55X	IA	0xXX	-	-
	0xAA	0x55	0x90	0x00	ID	-	-
自动页面程序	0x55X	0xAAX	0x55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
自动芯片擦除	0x55X	0xAAX	0x55X	0x55X	0xAAX	0x55X	-
	0xAA	0x55	0x80	0xAA	0x55	0x10	-
自动存储块擦除	0x55X	0xAAX	0x55X	0x55X	0xAAX	BA	-
	0xAA	0x55	0x80	0xAA	0x55	0x30	-
自动保护位程序	0x55X	0xAAX	0x55X	0x55X	0xAAX	0x55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
自动保护位擦除	0x55X	0xAAX	0x55X	0x55X	0xAAX	0x55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

补充说明

- IA:识别码地址
- ID:ID数据
- PA:程序页地址
- PD:程序数据(32-位数据)

在第四个总线周期之后，按地址的顺序按页面输入数据。

- BA:存储块地址 (见表 17-7)

- PBA:保护位地址(见表17-8)

17.2.6.2 总线周期中的地址位配置

表 17-6应与”表 17-5 命令时序”配套使用。

按照来自第一个总线周期的标准总线写入周期地址配置，设置该地址设置。

表 17-6 总线写入周期中的地址位配置

地址	Addr. [31:15]	Addr. [14]	Addr. [13:12]	Addr. [11:9]	Addr. [8:7]	Addr. [6:4]	Addr. [3:0]
正常命令	标准总线写入周期地址配置						
	闪存区	建议采用“0”。		命令	Addr[1:0] = “0” (固定) 其它位 = “0” (建议采用)		
ID-READ	IAID地址(ID-READ第四个总线写入周期的设置)						
	闪存区	建议采用“0”。	识别码地址	Addr[1:0] = “0” (固定) 其它位 = “0” (建议采用)			
存储块擦除	BA存储块地址(存储块擦除用第六个总线写入周期地址的设置)						
	存储块地址(表 17-7)	Addr[1:0] = “0” (固定) 其它位 = “0” (建议采用)					
自动页面程序	帕程序页地址(页面程序第四个总线写入周期地址的设置)						
	页面地址					Addr[1:0] = “0” (固定) 其它位 = “0” (建议采用)	
保护位程序	PBA:保护位地址(保护位程序第七个总线写入周期地址的设置)						
	闪存区	固定为“0”			保护位选择 (表 17-8)	Addr[1:0] = “0” (固定) 其它位 = “0” (建议采用)	

17.2.6.3 存储块地址(BA)

表 17-7 给出了各存储块地址。可指定拟在自动存储块擦除命令的第六个总线写入周期中被擦除的存储块所包含的任何地址。

表 17-7 存储块地址

模块	地址 (用户引导模式)	地址 (单引导模式)	容量 (K字节)
2	0x0001_8000 ~ 0x0001_FFFF	0x3F81_8000 ~ 0x3F81_FFFF	32
3	0x0001_0000 ~ 0x0001_7FFF	0x3F81_0000 ~ 0x3F81_7FFF	32
1	0x0000_8000 ~ 0x0000_FFFF	0x3F80_8000 ~ 0x3F80_FFFF	32
0	0x0000_0000 ~ 0x0000_7FFF	0x3F80_0000 ~ 0x3F80_7FFF	32

17.2.6.4 如何指定保护位(PBA)

可在 1-位单元中（编程时）、以及在 4-位单元中（擦除时）指定该保护位。

表 17-8 为自动保护位程序的保护位选择表。地址示例栏表示上面所述地址可用于使用引导模式，而下面的则可用于单引导模式。

该自动保护位擦除命令总共可擦除四个保护位。

表 17-8 保护位程序地址

模块	保护位	第七个写入周期的地址			地址示例 [31:0]
		地址 [14:9]	地址 [8]	地址 [7]	
Block0	<BLK[0]>	固定为 "0"	0	0	0x0000_0000 0x3F80_0000
Block1	<BLK[1]>		0	1	0x0000_0080 0x3F80_0080
Block2	<BLK[2]>		1	0	0x0000_0100 0x3F80_0100
Block3	<BLK[3]>		1	1	0x0000_0180 0x3F80_0180

17.2.6.5 ID 读取代码(IA, ID)

表 17-9 给出了用ID读命令指定某代码和内容的方法。

地址示例栏表示上面所述的地址可用于使用引导模式，而下面的则可用于单引导模式。

表 17-9 ID读命令代码和内容

代码	ID[7:0]	IA[13:12]	地址示例 [31:0]
制造代码	0x98	0b00	0x0000_0000 0x3F80_0000
装置代码	0x5A	0b01	0x0000_1000 0x3F80_1000
-	保留	0b10	-
宏代码	0x33	0b11	0x0000_3000 0x3F80_3000

17.2.6.6 命令时序示例

(1) 使用引导模式

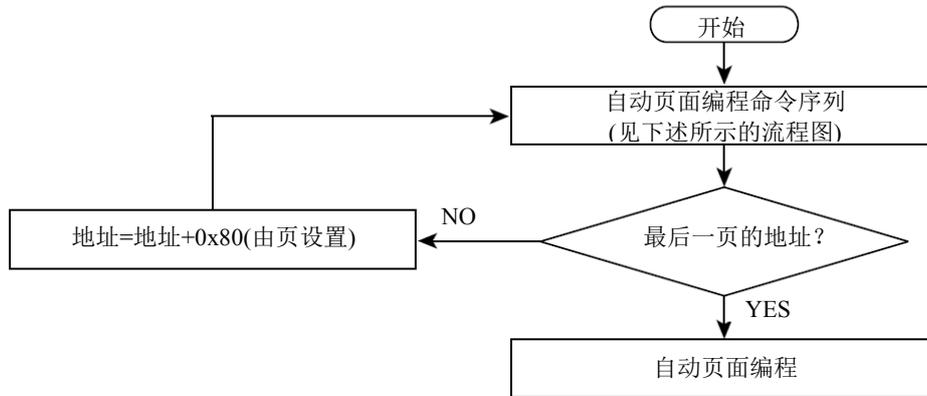
命令	总线周期							
		1	2	3	4	5	6	7
读取	地址	0x0000_0000	-	-	-	-	-	-
	数据	0x0000_00F0	-	-	-	-	-	-
读取/复位	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	-	-	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_00F0	-	-	-	-
ID-Read	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	-	-
自动页面程序	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	在以下周期中，按页面连续写入地址和数据。		
	数据	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
自动芯片擦除	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	-
自动存储块擦除	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	BA	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	-
自动保护位程序	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	PBA
	数据	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
自动保护位擦除	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550
	数据	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

(2) 数据单引导模式

命令	总线周期							
		1	2	3	4	5	6	7
读取	地址	0x3F80_0000	-	-	-	-	-	-
	数据	0x0000_00F0	-	-	-	-	-	-
读取/复位	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	-	-	-	-
	数据	0x0000_00AA	0x3F80_0055	0x3F80_00F0	-	-	-	-
ID-Read	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	IA	0x0000_0000	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	-	-
自动页面程序	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PA	在以下周期中，按页面连续写入地址和数据。		
	数据	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
自动芯片擦除	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	-
自动存储块擦除	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	BA	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	-
自动保护位程序	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PBA
	数据	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
自动保护位擦除	地址	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550
	数据	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

17.2.7 流程图

17.2.7.1 自动程序



自动页面编程命令时序(地址/命令)

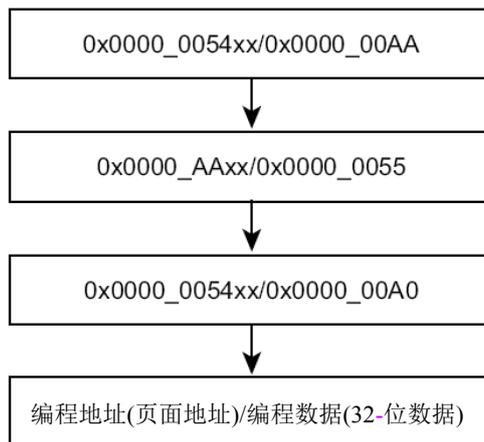


图 17-4 自动程序流程图

17.2.7.2 自动擦除

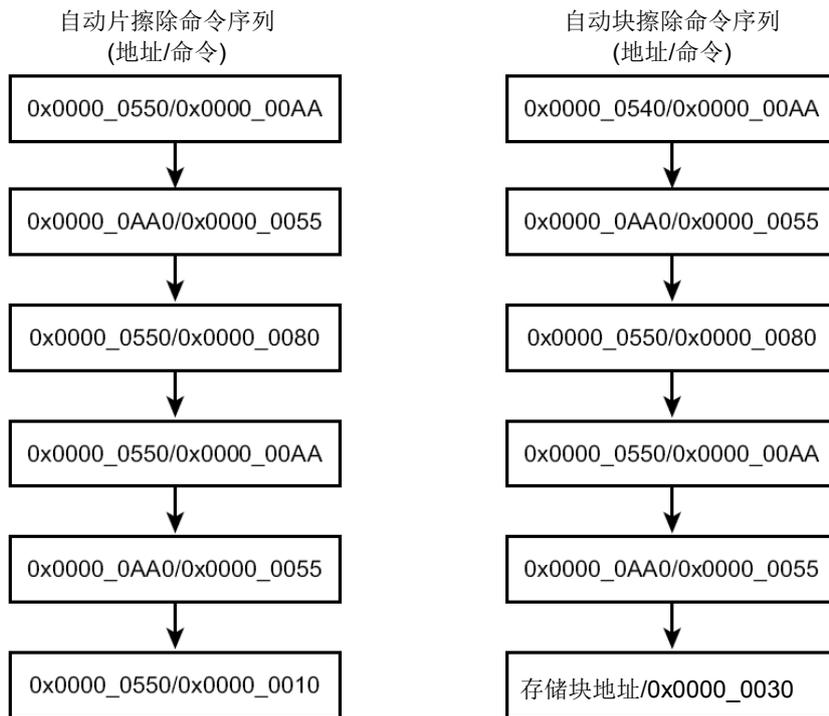
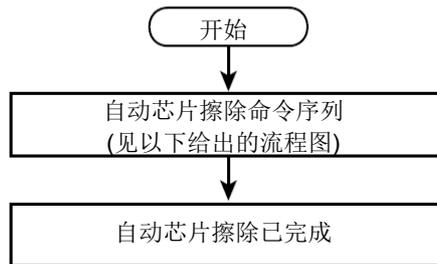


图 17-5 自动擦除流程图

17.3 如何用单引导模式重新编程闪存

单引导模式可运用重新编程闪存的内置引导ROM中所含的某个程序。在模式下，引导ROM被地址至内含中断向量表的区，而闪存即被地址至另一地址区（引导ROM区除外）。

在该引导模式下，可用串行命令/数据传送对闪存进行重新编程。将该装置的串行通道(SIO/UART)连接至外部主机，然后将重新编程程序从外部主机复制到内置RAM。即可执行该RAM中的重新编程例行程序，对闪存进行重新编程。有关与主机通信的详细描述，请遵循后文所述的协议。

即使在单引导模式下，除复位外也不要生成中断/故障，以避免发生程序异常终止现象。

在单芯片模式(标准操作模式)下保护闪存的内容时，建议一旦完成重新编程，即采取措施防止相关闪存存储块在后续的单芯片操作期间受到意外消磁的影响。

17.3.1 模式设置

为了执行在板编程，应在单引导模式下对该装置进行引导。以下设置可用于单引导模式设置。

$$\overline{\text{BOOT}} = 0$$
$$\overline{\text{RESET}} = 0 \rightarrow 1$$

在按上述要求提前对 $\overline{\text{BOOT}}$ 引脚进行设置的同时，应将 $\overline{\text{RESET}}$ 引脚设置为“0”。然后解除 $\overline{\text{RESET}}$ 引脚，随后设备就会在单引导模式下开始引导。

17.3.2 接口规格

本节主要对单引导模式下的SIO/UART通信格式进行说明。该串行操作支持UART(异步通信)和I/O接口方式。为执行在板编程，还需设置编程控制器的通信格式。

- UART通信
 - 通信通道：通道4
 - 串行传送模式：UART (非同步)，半双工，LSB优先
 - 数据长度：8 位
 - 奇偶检验位：无
 - 停止位：1-位1位
 - 波特率：任意波特率

- I/O 接口模式
 - 通信通道：通道4
 - 串行传送模式：I/O接口，全双工，LSB优先
 - 同步信号(SCLK4)：输入模式,上升沿设置
 - 握手信号：PB4(输出模式)
 - 波特率：任意波特率

该引导程序可将该时钟/模式控制块设置作为初始条件运行。有关时钟初始设置的详细描述，请参看“时钟/模式控制”。

如”17.3.5.1 连续操作模式确定”中所述，可由 16-位计时器(TMRB)确定某个波特率。在确定该波特率时，按某个预定波特率的 1/16 执行通信。因此，该通信波特率必须在该可测量的范围以内。计时器在 $\Phi T1$ ($f_c/2$)时运行。

I/O接口模式的握手引脚可在接收状态输出”低”等待，并在传输状态输出”高”。在通信之前应检查该握手引脚，并必须遵循该通信协议。

表 17-10 给出了引导程序中所使用的各引脚。该引导程序不使用这些引脚以外的其它引脚。

表 17-10 引脚连接

引脚		接口	
		UART	I/O接口模式
模式设置引脚	$\overline{\text{BOOT}}$	○	○
复位引脚	$\overline{\text{RESET}}$	○	○
通信引脚	TXD4 (PB2)	○	○
	RXD4 (PB1)	○	○
	SCLK4 (PB3)	x	○(输入模式)
	PB4	x	○(输出模式)

○:已使用 x未使用

17.3.3 对内部存储器的限制

注意，单引导模式对表 17-11 中所列的内置RAM和内置闪存设有限制条件。

表 17-11 单引导模式下对各存储器的限制条件

存储器	限制
内部RAM	引导程序将该存储器用作工作区（从 0x2000_0000 到 0x2000_03FF）。过RAM的结束地址存储该程序 0x2000_0400。该起始地址必须是偶校验地址。
内部闪存	以下地址被指定用于存储软件ID信息和密码。建议不要将程序存储到以下地址。 0x3F81_FFF0 到 0x3F81_FFFF

注：如果密码属于已擦除的数据(0xFF)，则难以保护数据安全，原因是密码很容易被猜出。即使不使用单引导模式，仍建议将某个唯一值设置为密码。

17.3.4 操作命令

该引导程序可提供以下操作命令。

表 17-12 操作命令数据

操作命令数据	操作模式
0x10	RAM传输
0x40	闪存芯片擦除和保护位擦除

17.3.4.1 RAM 传输

RAM传输用于将数据从控制器存储到内置RAM。在传输正常完成时，用户程序即启动。用户程序可将 0x2000_0400 或后续的存储器地址(0x2000_0000至0x2000_03FF除外)用于该引导程序。CPU将从RAM存储起始地址开始执行。该起始地址必须是偶校验地址。

该RAM传输功能可启用用户特定的在板程序控制。为由某个用户程序执行在板编程，应使用 17.2.6 中所述的闪存命令时序。

17.3.4.2 闪存芯片擦除和保护位擦除

闪存芯片擦除和保护位擦除命令可擦除闪存的整个存储块，并写入/擦除所有存储块的保护，且不考虑写入/擦除保护或安全状态。

17.3.5 普通操作（不考虑命令）

本节主要对引导程序执行时的普通操作进行说明。

17.3.5.1 连续操作模式的确定

在该控制器通过UART通信时，应在预定波特率时将第一字节设置为 0x86。在该控制器通过I/O接口模式通信时，应在预定波特率的 1/16 时将第一字节设置为 0x30。图 17-6 给出了各种情况下的波形。

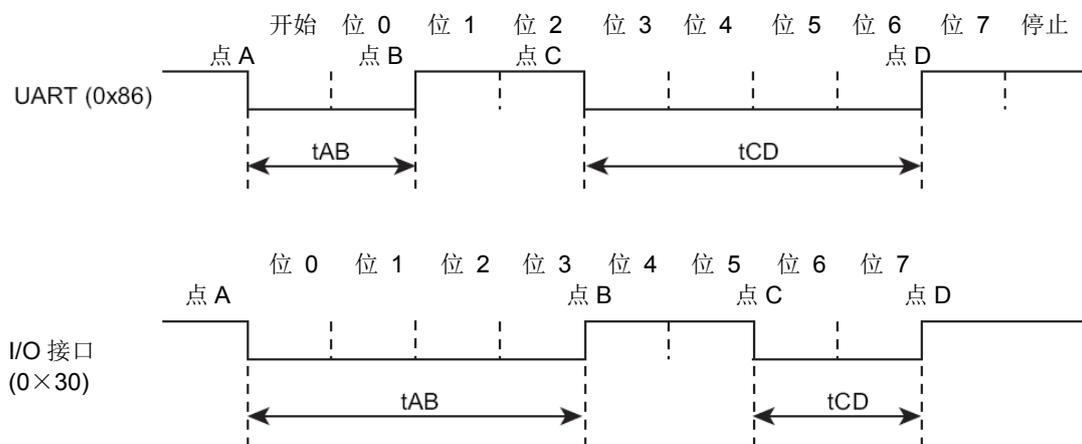


图 17-6 连续操作模式的确定数据

图 17-7 给出了引导程序的流程图。利用 16-位计时器(TMRB)(时间为 t_{AB} 、 t_{AC} 和 t_{AD})，可规定复位后连续操作模式确定数据(0x86, 0x30)的第一个字节。如图 17-7 所示，CPU可监控该接收引脚的电平，并可在该接收引脚的电平变化时立即获得某个计时器值。因此， t_{AB} 、 t_{AC} 和 t_{AD} 的计时器值具备某个误差容限。此外，还应注意的是，如果传输时的波特率较高，则CPU不能确定该接收引脚的电平。尤其是I/O接口易于发生该问题，原因是其波特率一般远高于UART的波特率。为避免发生这种现象，该控制器应在I/O接口模式下按预定波特率的 1/16 发送数据。

如图 17-8 中的流程图所示，可确定连续操作模式，并确定该接收引脚持续时间的长短。如果该长度为 $t_{AB} \leq t_{CD}$ ，则连续操作模式可被确定为UART模式。无论自动波特率设置是否被启用，均可使用该 t_{AD} 时间。如果该长度为 $t_{AB} > t_{CD}$ ，则连续操作模式可被确定为I/O接口模式。注意， t_{AB} 、 t_{AC} 和 t_{AD} 的计时器值可存在某个误差容限。如果该波特率高而工作频率“低”，则各计时器值会变小。这可能会导致发生出乎意料的确定(为防止发生该问题，可在该编程例行程序内复位UART)。

例如，在运用UART模式时，如果预计会在时间收到来自目标板的回应，则控制器应把超时时段考虑进去。控制器如果不能在允许的时间内得到回应，应当放弃通信。

例如，在模式为UART模式时，可将串行操作模式确定为I/O接口模式。为避免出现这种情况，在运用UART模式时，如果该时间预计可接收到来自目标板的回波信号，则控制器应允许超时时间出现。如果该控制器未能在该允许时间内该回波信号，则其应放弃通信。在使用I/O接口模式时，一旦第一个串行字节已被发送，则控制器应在某空闲时间之后发送SCLK时钟，以获取应答响应。如果所接收的应答响应不是 0x30，则控制器应放弃下一步的通信。

在预定模式是I/O接口模式时，只要 $t_{AB} > t_{CD}$ （如上所述），第一个字节就不必是 0x30。可将 0x91、0xA1或 0xB1 作为第一个字节代码进行发送，以确定点A和点C的下降缘、以及点B和点D的上升缘。如已确定 $t_{AB} > t_{CD}$ ，且该操作模式确定的分辨率已选定SIO，则第二字节代码应为 0x30，即便第一个字节的已发送代码不是 0x30 (用于确定I/O接口模式的第一个字节代码被描述为 0x30)。

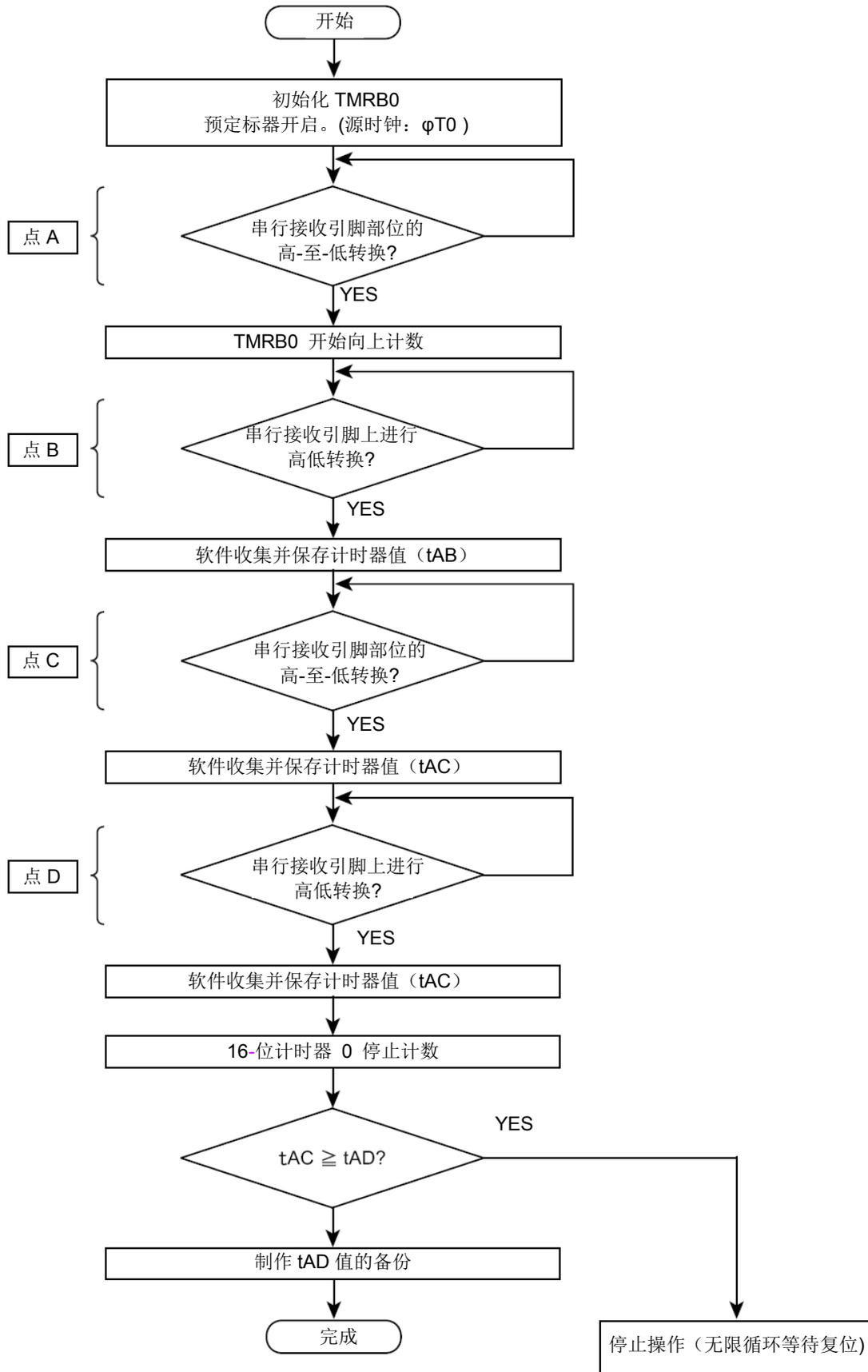


图 17-7 连续操作模式接收流程图

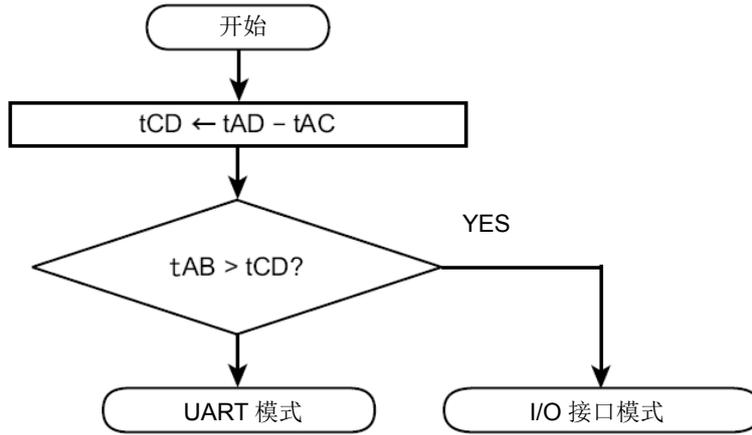


图 17-8 连续操作模式确定流程图

17.3.5.2 应答响应数据

引导程序用具体代码表示处理状态，并将其发送到控制器。表 17-13~表 17-16 给出了对各接收数据确认响应的值。

在表 17-14~表 17-16 中，确认响应的上四位字节等于操作命令数据的上四位字节。第三位可指示接收错误。第零位可指示无效操作命令错误、校验和错误或密码错误。第一位和第二位始终为“0”。在I/O接口模式下，不会执行接收误差检查。

表 17-13 串行操作确定数据ACK响应

传送数据	描述
0x86	已确定可进行UART通信。(注)
0x30	已确定可进行I/O接口通信。

注：在串行操作被确定为UART时，若波特率设置被确定为不合格，则引导程序放弃，且不会发回任何响应。

表 17-14 操作命令数据ACK响应

传送数据	描述
0x?8(注)	在操作命令数据中发生了接收错误
0x?1(注)	未定义操作命令数据接收正常。
0x10	已被确定为RAM传送命令
0x40	已被确定为闪存芯片擦除命令

注：该应答响应数据的上 4 位与前述命令数据的相同。

表 17-15 检查和数据的确认响应

传送数据	描述
0xN8 (注)	发生接收错误。
0xN1 (注)	发生检查错误和或密码错误。
0xN0 (注)	CHECK SUM值正确。

注：该应答响应数据的上 4 位与该操作命令数据的相同。

表 17-16 闪存芯片擦除和保护位擦除操作ACK响应

传送数据	描述
0x54	已被确定为擦除启用命令
0x4F	擦除命令已完成。
0x4C	擦除命令被异常终止。

注：即使在擦除命令被正常执行时，也可能通过ACK响应返回一个否定应答。检查 FCSR<RDY_BSY>，确认命令时序终止，保持200 μs或以上，然后再次确认清除状态。

17.3.5.3 密码确定

该引导程序使用以下区确定是否需要密码或用作密码。

区	地址
密码请求确定	0x3F81_FFF0 (1字节)
密码区	0x3F81_FFF4 ~ 0x3F81_FFFF (12字节)

RAM传送命令可执行密码验证，且不考虑必要性判定数据。闪存芯片擦除或保护位擦除命令，仅可在必要性判定已被确定为“必要”时执行密码验证。

密码要求设置	DATA
需要密码	0xFF除外
无密码	0xFF

如果密码被设置为 0xFF(已擦除的数据)，则难以保护数据安全，原因是密码很容易被猜出。即使不使用单引导模式，仍建议将某个唯一值设置为密码。

(1) 使用RAM传送命令进行密码验证

如果所有这些地址位置均包含同样的数据字节(0xFF除外)，则可将这种情况确定为密码区错误(如图 17-9 所示)。在这种情况下，引导程序会根据CHECK SUM值的第 17 个字节返回错误应答(0x11)，且不考虑密码验证。

该引导程序可验证接收数据的第五字节到第十六字节(密码数据)。如果并非所有 12 个字节均匹配，则会发生密码错误。如已确定密码错误，则对第 17 个检查和数据的应答响应数据即为密码错误。

即使安全功能启用，也可进行密码验证。

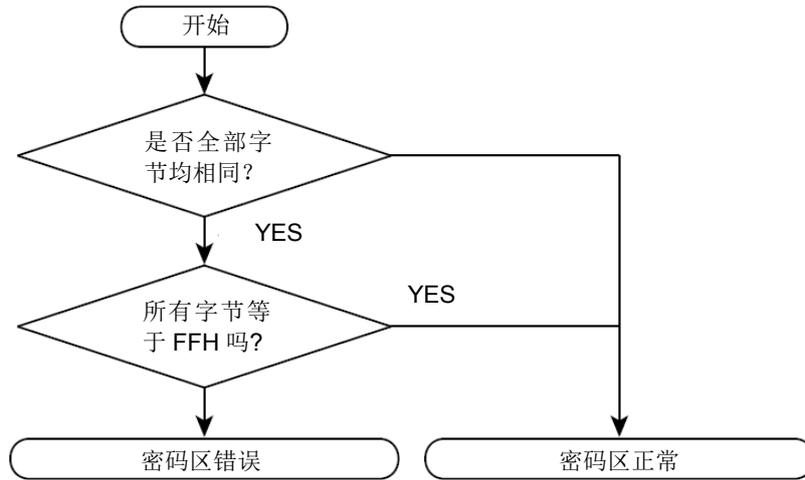


图 17-9 密码区检查流程图

(2) 对闪存芯片擦除和保护位擦除命令的密码验证

当密码在必须擦除确定区内（如图 17-10 所示）被启用、且这些密码是完全相同的数据时，就会发生密码区错误。当确定发生个密码区错误时，对检查和的第 17 个字节的ACK响应会发送 0x41，且不考虑密码验证。

该引导程序可验证接收数据的第五字节到第十六字节(密码数据)。如果并非所有 12 个字节均匹配，则会发生密码错误。如已确定密码错误，则对第 17 个检查和数据的应答响应数据即为密码错误。

即使安全功能启用，也可进行密码验证。

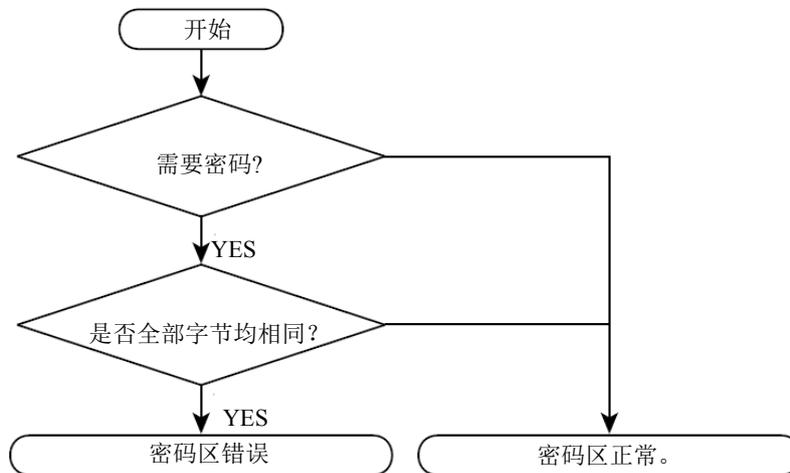


图 17-10 密码区检查流程图

17.3.5.4 CHECK SUM 计算

检查和的计算，就是对传输数据按 8-位加法进行计算，撤消各进位，并取两者的总和的余数。控制器必须在发送检查和字节时执行同样的检查和操作。

CHECK SUM示例

执行8-位加法，即可计算一系列 0xE5 和 0xF6 的检查和。

$$0xE5 + 0xF6 = 0x1DB$$

取该和的两者的余数加到下 8-位，其即为CHECK SUM值。因此，引导程序会将 0x25 发送给该控制器。

$$0 - 0xDB = 0x25$$

17.3.6 RAM 传送时的传送格式

本节给出了RAM传送命令格式。该表中的传送方向注明如下：

传送方向(C→T)：控制器至TMPM037FWUG

传送方向(C←T)：TMPM037FWUG至控制器

传送字节数	传送方向	传送数据	描述
1.	C→T	连续操作模式和波特率设置	发送数据确定连续操作模式。模式确定详情，参阅“17.3.5.1 连续操作模式确定”。
		[UART模式] 0x86	发送 0x86。如果UART模式已被确定，则该程序会确定是否可进行波特设置。如不可进行，则程序停止，而通信则停止工作。
		[I/O接口模式] 0x30	按预定波特率的 1/6 发送 0x30。同时也按预定波特率的 1/6 发送第二字节。用户可按所需的波特率，从第三字节或后续字节发送数据。
2	C←T	对连续操作模式的ACK响应	该传输数据的第二字节，是对应于串行操作设置模式数据的第一字节的ACK响应数据。如果该设置可行，则设置SIO/UART。应在传输缓冲器被写入到该数据之前，设置接收启用计时。
		[UART模式] 正常状态：0x86	如果该设置已被确定为可行，则发送 0x86。如被确定为不可行，则操作放弃，且不发回任何响应。 在控制器完成发送数据的第一字节时，要求等待某个超时时间(5秒)。如果数据(0x86)未能在超时时间内被正常接收，则通信不可进行。
		[I/O接口模式] 正常状态：0x30	将数据(0x30)写入到传输缓冲器，并等待SCLK0 时钟。在控制器完成发送数据的第一字节并等待若干ms(空闲时间)后，输出SCLK时钟。此时，波特率可被设置为预定波特率的 1/16。如果接收数据为 0x30，则可进行通信。在第三字节后，将所需的波特率设置到通信。
3	C→T	操作命令数据(0x10)	发送RAM传送命令数据(0x10)。
4	C←T	对操作命令的ACK响应	对操作命令的ACK响应数据。
		正常状态：0x10 异常状态：0xX1 通信错误：0xX8	首先，检查接收数据的第三字节是否有错误。(仅适用于UART模式) 如果存在接收错误，则发送表示通信异常的ACK响应数据 0xX8，并等待下一个操作命令(第三字节)。传输数据的上四位未定义(与紧挨在操作命令之前的上四位相同)。注意，不会在I/O接口中执行接收误差检查。 然后，如果接收数据的第三字节符合表 17-12 中的任一操作命令数据，则接收数据即被回送。如果是RAM传送，则 0x10 会被回送，且传送数据会转移到RAM传送服务例行程序。如果该数据与表 17-12 中的该命令不对应，则发送表示操作命令错误的ACK响应数据 0xX1，并等待下一条操作命令(第三字节)。传输数据的上四位未定义(与紧挨在操作命令数据之前的上四位相同)。
5 to 16	C→T	密码数据(12-byte) 0x3F81_FFF4 ~ 0x3F81_FFFF	检查密码区中的数据。有关密码区检查的详细描述，请参看“17.3.5.3 密码确定”。 将闪存数据的 0x3F81_FFF0 至 0x3F81_FFFF与接收数据的第 5 到第 16 字节进行比较。如果数据不匹配该地址，则应设置一个密码错误标志。
17	C→T	CHECK SUM值的第五至第十六字节	发送CHECK SUM值的第五至第十六字节。 有关CHECK SUM计算的详细描述，请参看 17.3.5.4。

传送字节数	传送方向	传送数据	描述
18	C←T	对CHECK SUM数值的ACK回应 正常状态: 0x10 异常状态: 0x11 通信错误: 0x18	首先, 检查是否接收数据的第 5~第 17 字节存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK回应数据 0x18, 意为通信异常, 等待下一操作指令(第 3 字节)。然后, 检查CHECK SUM数据的第 17 字节。若错误存在, 发送 0x11, 等待下一操作指令(第 3 字节)。 最后, 检查密码验证结果。若存在密码错误, 发送ACK回应数据 0x11 意为密码错误, 等待下一操作指令(第 3 字节)。 若所有步骤正常结束, 发送正常ACK回应数据 0x10。
19	C→T	RAM存储开始地址 31~24	发送块传送开始地址, 用于RAM存储。第 19 字节对应地址的第 31~24 位。第 22 字节对应地址的第 7~0 位。 通过RAM的最后地址, 将该地址指定为地址 0x2000_0400。 RAM该地址必须是偶校验地址。
20	C→T	RAM存储开始地址 23~16	
21	C→T	RAM存储开始地址 15~8	
22	C→T	RAM存储开始地址 7~0	
23	C→T	RAM存储字节数 15~8	设定字节数, 以进行块传送。第 23 字节对应传送字节的第 15~8 位。第 24 字节对应传送字节的第 7~0 位。指定拟存储于 自RAM地址 0x2000_0400 直至最后地址的数据。
24	C→T	RAM存储字节数 7~0	
25	C→T	CHECK SUM 数值的第 19~24 字节	发送CHECK SUM值的第 19~24 字节。
26	C←T	对CHECK SUM数值的ACK回应 正常状态: 0x10 异常状态: 0x11 通信错误: 0x18	首先, 检查是否接收数据的第 19~25 字节存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK回应数据 0x18, 意为通信异常, 等待下一操作指令(第 3 字节)。然后, 检查CHECK SUM数据的第 25 字节。若错误存在, 发送 0x11, 等待下一操作指令(第 3 字节)。 若所有步骤正常结束, 发送正常ACK回应数据 0x10。
27~m	C→T	RAM已存数据	发送用于RAM已存储数据, 于第 23~24 字节中指定数据的相同字节。
m+1	C→T	CHECK SUM 数值的第 27~m 字节	发送CHECK SUM值的第 27~m 字节
m+2	C←T	对CHECK SUM数值的ACK回应 正常状态: 0x10 异常状态: 0x11 通信错误: 0x18	首先, 检查是否接收数据的第 27~m+1 字节存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK回应数据 0x18, 意为通信异常, 等待下一操作指令(第 3 字节)。然后, 检查CHECK SUM数据的第 m+1 字节。若错误存在, 发送 0x11, 等待下一操作指令(第 3 字节)。若所有步骤正常结束, 发送正常ACK回应数据 0x10。
-	-	-	若ACK回应数据的第 m+2 字节为正常ACK回应数据, 传送数据分至第 19~22 字节指定的地址。

17.3.7 闪存芯片擦除和保护位擦除的转换格式

本节所示为闪存芯片擦除和保护位擦除命令。表中的传送方向注明如下：

传送方向(C→T)：控制器至TMPM037FWUG

传送方向(C←T)：TMPM037FWUG至控制器

传送字节数	传送方向	传送数据	描述
1	C→T	连续操作模式和波特率设置	发送数据以确定串行操作模式模式确定详情，参阅"17.3.5.1 连续操作模式确定"。
		[UART 模式] 0x86	发送 0x86。若UART模式确定，检查是否波特率设置可完成。若否，则进行停止通信操作。
		[I/O 接口模式] 0x30	于所需波特率的 1/16 发送 0x30。和第 1 字节相同，于所需波特率的 1/16 发送第 2 字节。所需波特率可用于第 3 字节之后。
2	C←T	对连续操作模式的ACK响应	该传输数据的第二字节，是对对应于串行操作设置模式数据的第一字节的ACK响应数据。如果该设置可行，则设置SIO/UART。应在传输缓冲器被写入到该数据之前，设置接收启用计时。
		[UART 模式] 正常状态：0x86	如果该设置已被确定为可行，则发送 0x86。如被确定为不可行，则操作放弃，且不发回任何响应。 在控制器完成发送数据的第一字节时，要求等待某个超时时间(5秒)。如果数据(0x86)未能在超时时间内被正常接收，则通信不可进行。
		[I/O 接口模式] 正常状态：0x30	将数据(0x30)写入到传输缓冲器，并等待SCLK0时钟。在控制器完成发送数据的第一字节并等待若干ms(空闲时间)后，输出SCLK时钟。此时，波特率可被设置为预定波特率的 1/16。如果接收数据为 0x30，则可进行通信。在第三字节后，将所需的波特率设置到通信。
3	C→T	操作命令数据(0x40)	发送闪存芯片擦除和保护位擦除命令数据(0x40)。
4	C←T	对操作命令的ACK响应	对操作命令的ACK响应数据。
		正常状态：0x40 异常状态：0xX1 通信错误：0xX8	首先，检查接收数据的第三字节是否有错误。(仅适用于UART模式) 如果存在接收错误，则发送表示通信异常的ACK响应数据 0xX8，并等待下一个操作命令(第三字节)。传输数据的上四位未定义(与紧挨在操作命令数据之前的上四位相同)。注意，不会在I/O接口中执行接收误差检查。 然后，如果接收数据的第三字节符合表 17-12 中的任一操作命令数据，则接收数据即被回送。当数据不符合表 17-12 中的命令时，发送一个表示操作命令错误的ACK 响应数据，并等待后续操作命令。(第 3 字节)传输数据高 4 位未定义。(使用了和紧接操作命令前的高4位。)
5 ~ 16	C→T	密码数据(12-byte) 0x3F81_FFF4 ~ 0x3F81_FFFF	若密码必要性设为"无"，数据为虚拟数据。 若密码必要性设为"必要"，检查密码区数据。密码区数据检查方法，参阅"17.3.5.3 密码确定"。 按顺序将接收数据的第 5 至第 16 个字节，与闪存数据的 0x3F81_FFF0 至 0x3F81_FFFF进行比较。若数据不匹配，设定密码错误标志。
17	C→T	第 5~16 个字节的CHECK SUM值	发送CHECK SUM值的第 5~16 字节。 CHECK SUM计算方法，参阅"17.3.5.4 CHECK SUM计算"。

传送字节数	传送方向	传送数据	描述
18	C←T	对CHECK SUM值的ACK回应 正常状态: 0x40 异常状态: 0x41 通信错误: 0x48	若密码必要性设为"无", 发送正常ACK回应数据 0x40。若密码必要性设为"必要", 首先检查是否 接收数据第 5~17 字节存在接收错误。(仅限于UART 模式)若接收错误存在, 发送ACK回应数据 0x48 意为通信异常, 等待下一个操作命令。(第 3 字节) 然后, 检查CHECK SUM数据的第 17 字节。若错误发生, 发送 0x41, 等待下一个操作命令(第 3 字节) 最后, 检查密码验证结果。若密码错误存在, 发送ACK回应数据 0x41, 意为密码错误, 等待下一个操作命令(第 3 字节) 若所有步骤正常结束, 发送正常ACK回应数据 0x40。
19	C→T	擦除启用命令数据(0x54)	发送启用命令数据(0x54)。
20	C←T	对擦除启用命令的ACK回应 正常状态: 0x54 异常状态: 0xX1 通信错误: 0x58	首先, 检查是否接收数据的第 19 字节存在错误。若接收错误存在, 发送ACK回应数据(第 3 位) 0x58, 意为通信异常, 等待下一个操作命令(第 3 字节)。 然后, 若接收数据的第 19 字节和擦除启用命令不符, 接收数据获回应(正常ACK回应数据)。这种情况下, 0x54 获回应, 传送数据分至闪存芯片擦除例行程序。 若数据和擦除启用命令不符, 发送ACK回应数据 (第 0 位) 0xX1, 等待下一个操作命令。传输数据的上 4 位未定义。(使用了和紧接操作命令前的高4位。)
21	C→T	对擦除命令的ACK回应 (注1) 正常状态: 0x4F 异常状态: 0x4C	若操作正常完成, 结束码(0x4F)被返回。若擦除错误发生, 错误码(0x4C)被返回。
-	-	-	等待下一个操作命令。

注1: 即使在擦除命令被正常执行时, ACK响应也可能返回一个否定应答。检查FCSR<RDY_BSY>, 确认命令时序终止, 保持 200 μs或以上, 然后再次确认清除状态。

17.3.8 Boot 程序完整流程图

本节显示引导程序的整个流程图。

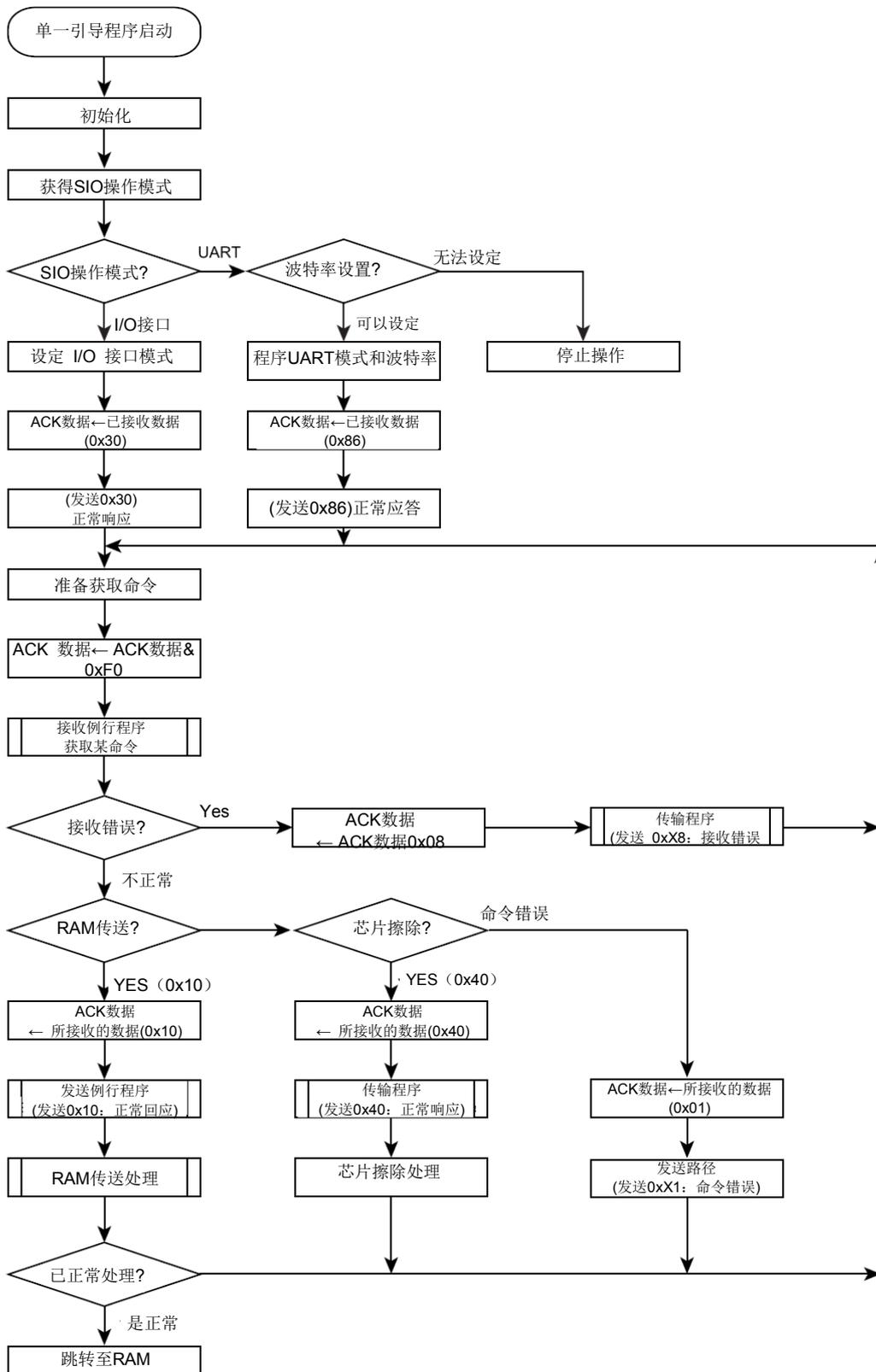


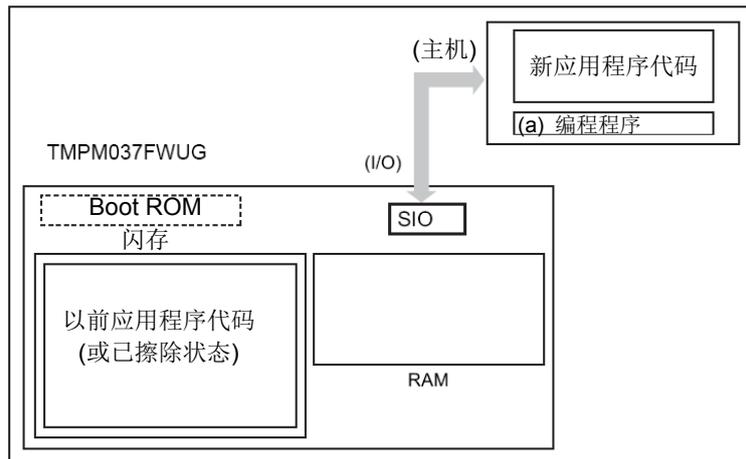
图 17-11 引导程序完整流程图

17.3.9 片装 BOOT ROM 中使用重编程序算法重编闪存程序

本节所述为片装启动ROM中使用重编程序算法重编闪存程序。

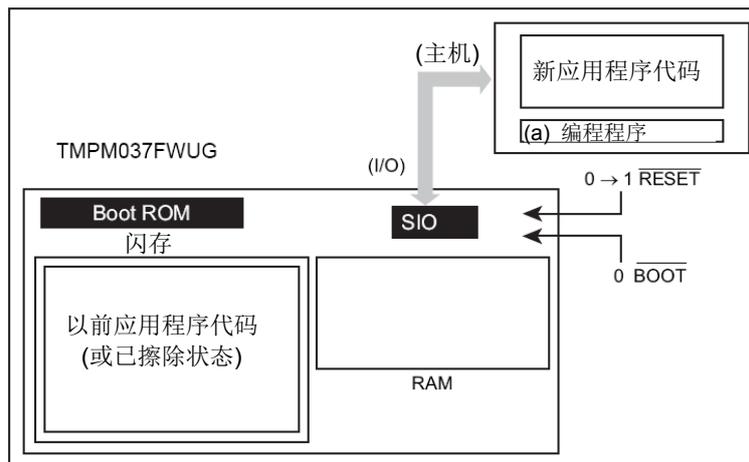
17.3.9.1 步骤-1

闪存条件不在乎是否前版构成的用户程序已写入或擦除。编程例程序和编程数据是通过SIO(SIO4)传送的，因此，必须将SIO4 连接至外部主机。编程程序(a)在主机上已准备好。



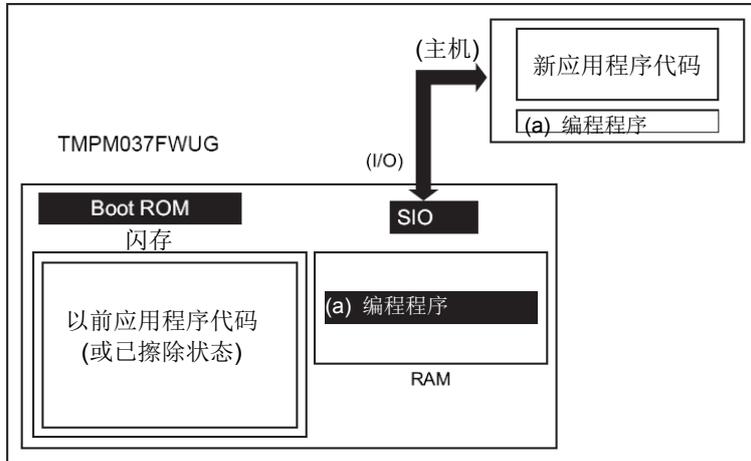
17.3.9.2 步骤-2

通过启动模式中的引脚条件设置解除复位，启动BOOT ROM。按照引导模式的程序，从源(主机)通过SIO4传送编程例程序(a)。用用户应用程序中的密码进行密码验证。(若闪存被擦除，通过密码对擦除数据(0xFF)进行处理。)



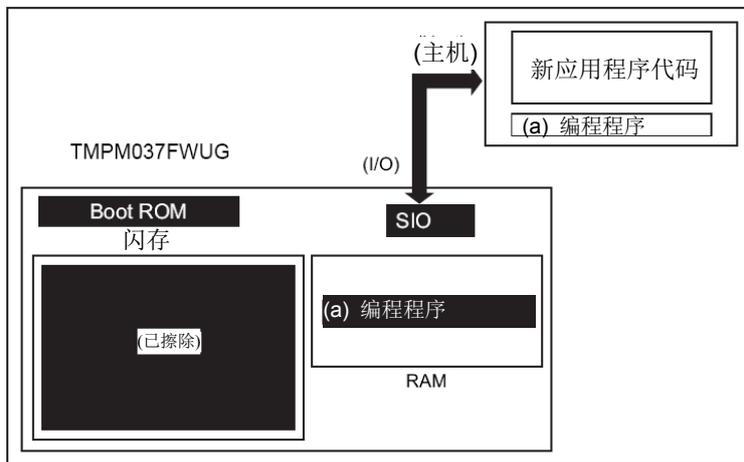
17.3.9.3 步骤-3

如果密码验证已完成，则引导程序会从主机将某个编程例行程序(a)传送到片装RAM。编程程序必须存入 0x2000_0400 到RAM终端地址之间的范围内。



17.3.9.4 步骤-4

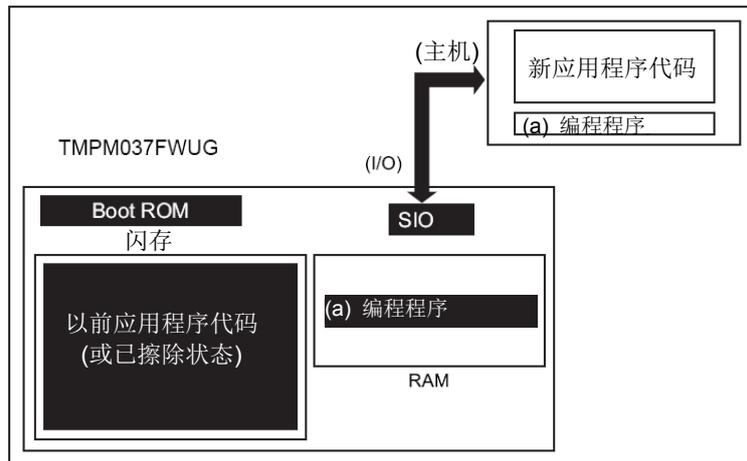
引导程序跳转至片装RAM中的编程路径(a)，以擦除内含旧应用程序代码的闪存存储块。使用块擦除或芯片擦除命令



17.3.9.5 步骤-5

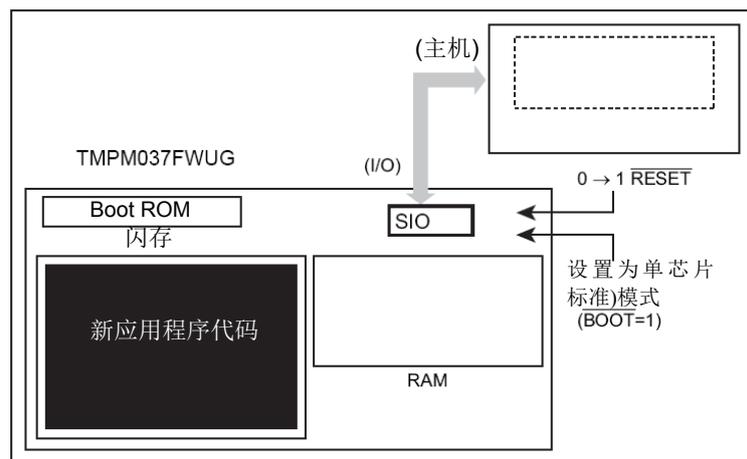
引导程序执行编程程序(a)自主机下载新应用程序代码，并将其编程至已擦除闪存区。编程完成时，须在用户程序中设定写入或擦除该闪存区保护。

在以下的示例中，新的程序代码通过与编程路径相同的SIO4 通道从同一主机出现。但是，一旦编程程序已开始执行，可自由更换传送路径和传送源。创建硬件板卡并编程程序以满足你的特定需求。



17.3.9.6 步骤-6

闪存编程完成时，令板卡断电，并断开主机和目标板卡之间的连接线。再次通电，令装置以单芯片机(正常)模式重启以执行新程序。



17.4 用户引导模式下编程

用户启动模式为使用用户定义的闪存编程程序。在用户应用上的闪存程序代码的数据传送总线不同于串行I/O时，就需要用到用户引导模式。其需在单芯片模式下运行；因此，必须从标准模式（其中的用户应用已在使用引导模式下激活）切换为编程闪存所需的用户引导模式。尤其是，在用户应用程序中增加了模式判断程序至复位服务程序。

需根据用户系统设置情况设定切换模式的条件。而且，用户独特编制的闪存编程程序需设定于新应用程序中。程序在切换至用户引导模式后用于编程。内置闪存的数据在擦除/重编程模式时无法读出。因此，重编程程序须在存储于闪存区以外的区中进行。一旦重编程完成，建议保护相关闪存块，以避免意外重编程。确保切勿生成复位外的中断/错误，以免用户引导模式中异常终止。

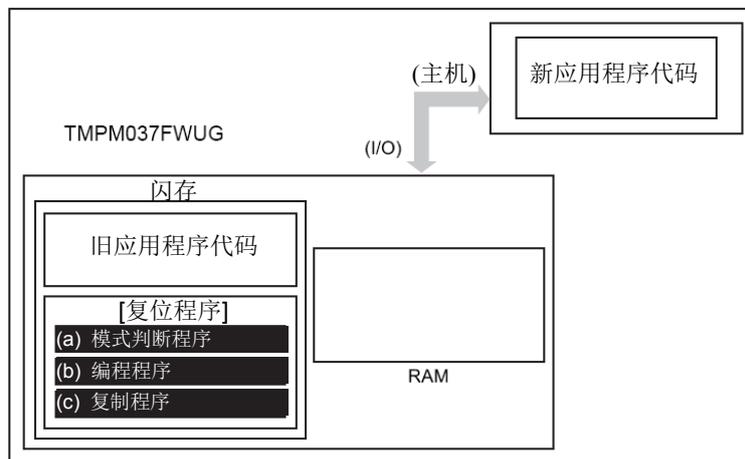
选取两种情况作为示例，如存储于闪存(1-A)，传送自外部装置(1-B)的重编程程序方法，下节对步骤予以阐述。编程/擦除至闪存详情，参阅“17.2 闪存详细描述”。

17.4.1 (1-A)编程程序存储于闪存的步骤

17.4.1.1 步骤-1

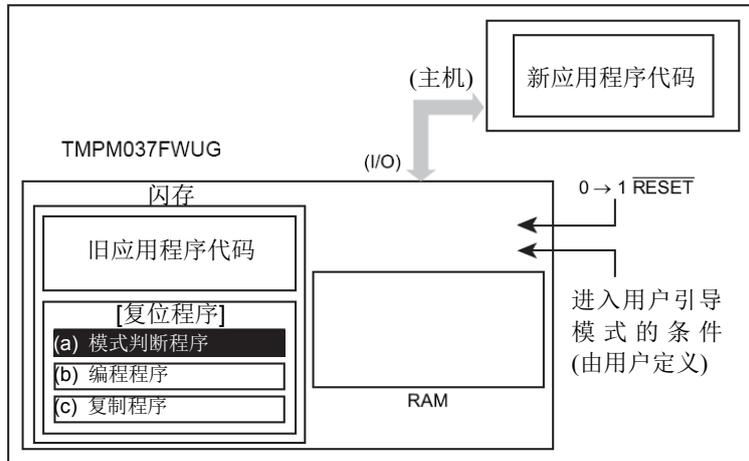
用户确定进入用户引导模式和拟用于传送数据的I/O总线的条件(如引脚状态)。然后创建适当的电路设计和程序。将装置安装于印刷电路板之前，用编程设备如闪存程序写入器将以下三个编程程序写入任意闪存块。

- (a) 模式确定路径： 确定是否切换至用户引导模式的程序
- (b) 闪存编程程序： 自主机控制器和重编程闪存下载新程序的程序
- (c) 复制程序： 一个用于将(A)中所述数据复制到内置RAM或外部存储器件的程序



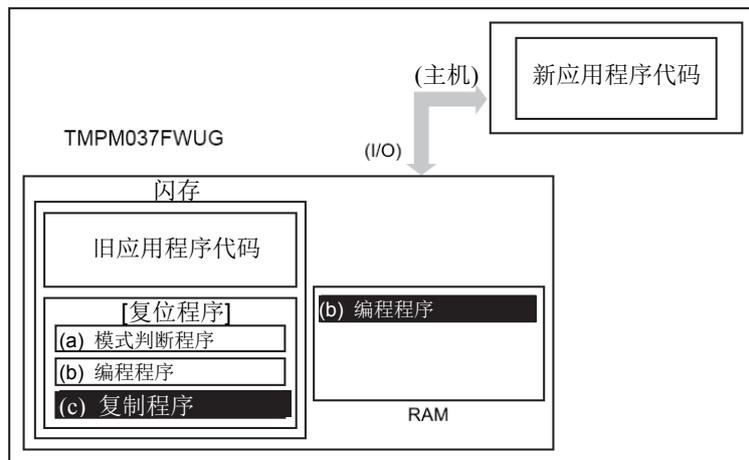
17.4.1.2 步骤-2

本节阐述了存储于复位程序的编程程序的情况。首先，复位路径确定进入用户引导模式。若模式切换条件符合，装置进入用户引导模式对数据进行重编程。



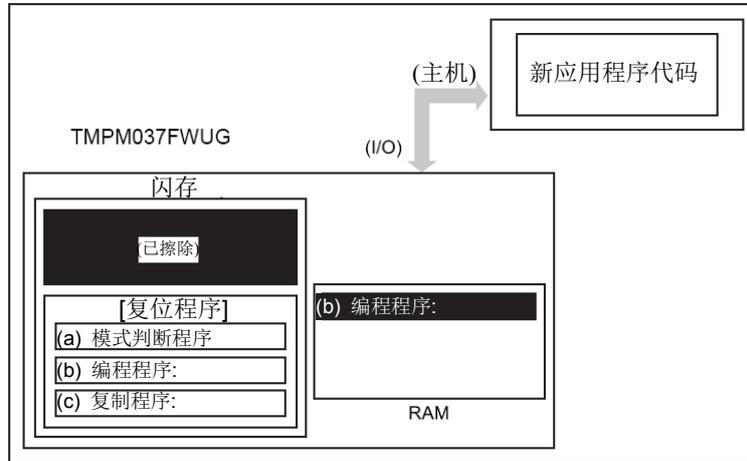
17.4.1.3 步骤-3

一旦器件进入用户引导模式，则执行复制路径(C)，以将闪存编程路径(b)从主机控制器下载到内置RAM。



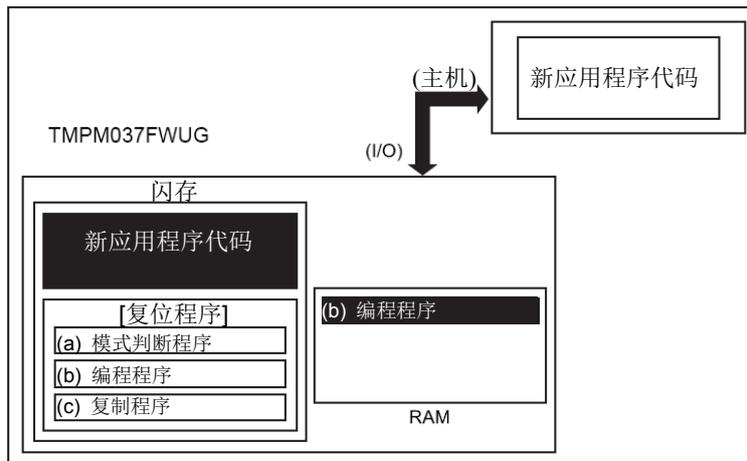
17.4.1.4 步骤-4

跳转至内置RAM中的重新编程路径，解除旧应用程序的写入/擦除保护，并擦除存储块单元中的某个闪存。



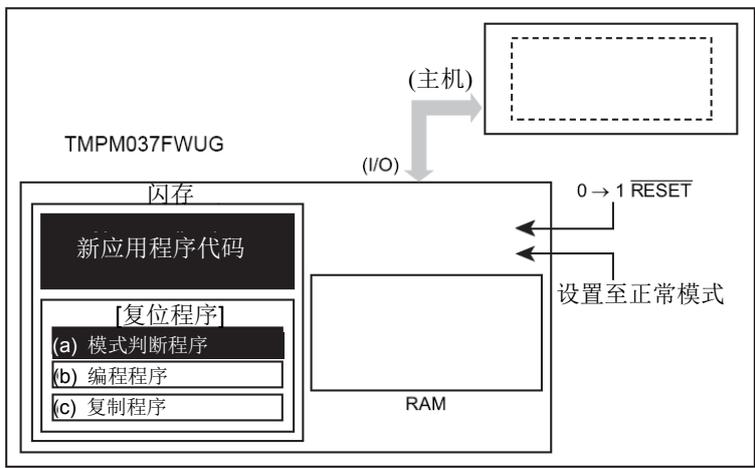
17.4.1.5 步骤-5

继续执行闪存编程路径，以从主机控制器下载新的程序数据，并将其编程到已被擦除的闪存存储块内。在编程完成时，必须对用户程序区中闪存存储块的写入/擦除保护进行设置。



17.4.1.6 步骤-6

设 $\overline{\text{RESET}}$ 为“0”。一旦复位，闪存设为正常模式。复位后，CPU将随新应用程序启动。



17.4.2 (1-B)编程程序自外部主机进行传送的步骤

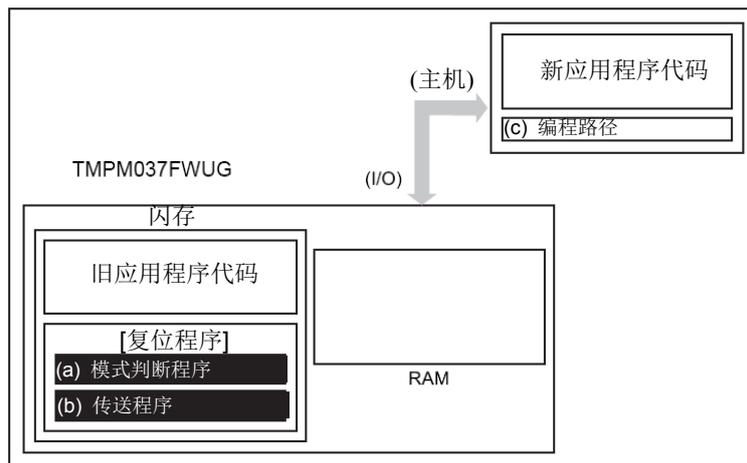
17.4.2.1 步骤-1

用户确定进入用户引导模式和拟用于传送数据的I/O总线的条件(如引脚状态)。然后创建适当的电路设计和程序。将装置安装于印刷电路板之前，用编程设备如闪存程序写入器将以下三个编程程序写入任意闪存块。

- (a) 模式确定路径: 确定是否切换至重编程操作的程序
- (b) 传送程序: 自外部装置获取重编程程序的程序

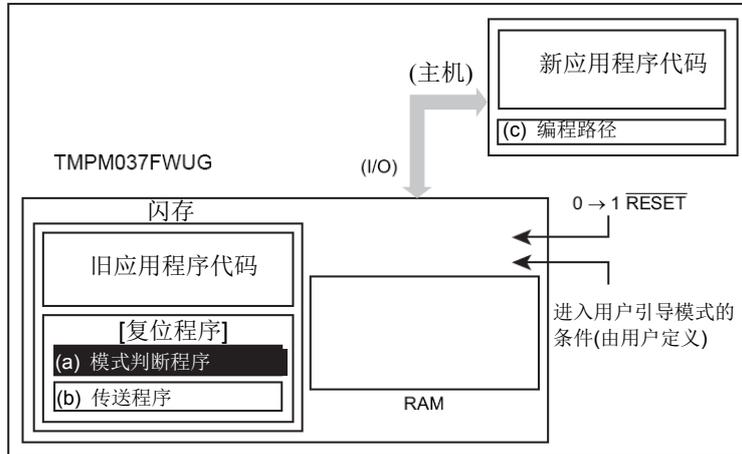
另外，准备须存储于主机控制器，如下所示的重编程程序。

- (c) 重编程程序: 对数据进行重编程的程序



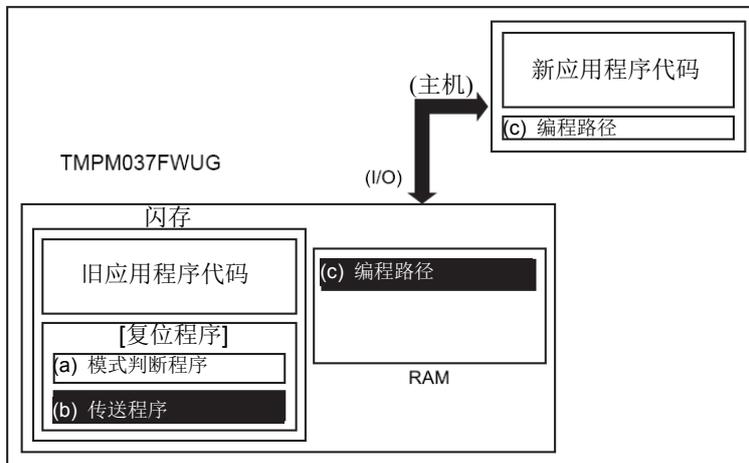
17.4.2.2 步骤-2

本节阐述了存储于复位程序的编程程序的情况。首先，复位路径确定进入用户引导模式。若模式切换条件符合，装置进入用户引导模式对数据进行重编程。



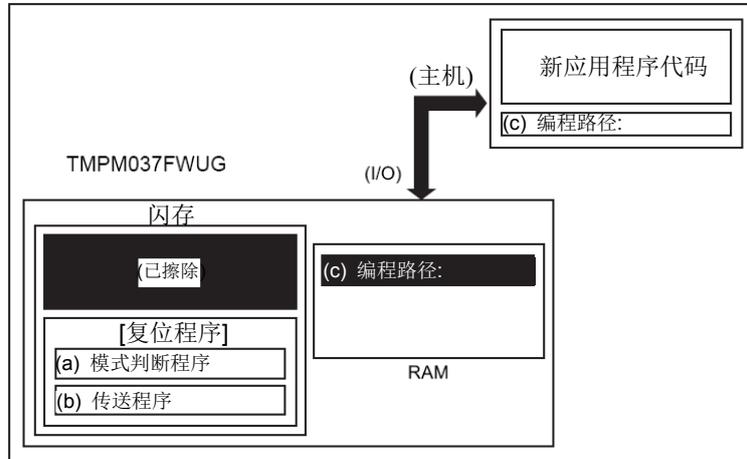
17.4.2.3 步骤-3

一旦器件进入用户引导模式，即可执行传送路径(b)，以将编程路径(c)从主机控制器下载到内置RAM中。



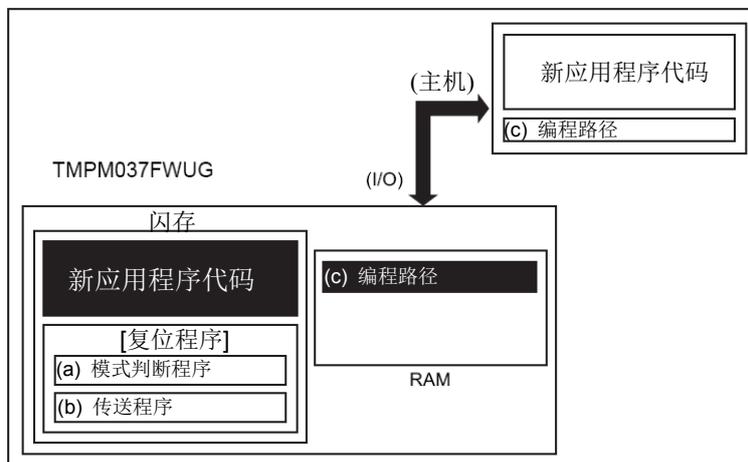
17.4.2.4 步骤-4

跳转至内置RAM中的重新编程路径，解除旧应用程序的写入/擦除保护，并擦除存储块单元中的某个闪存。



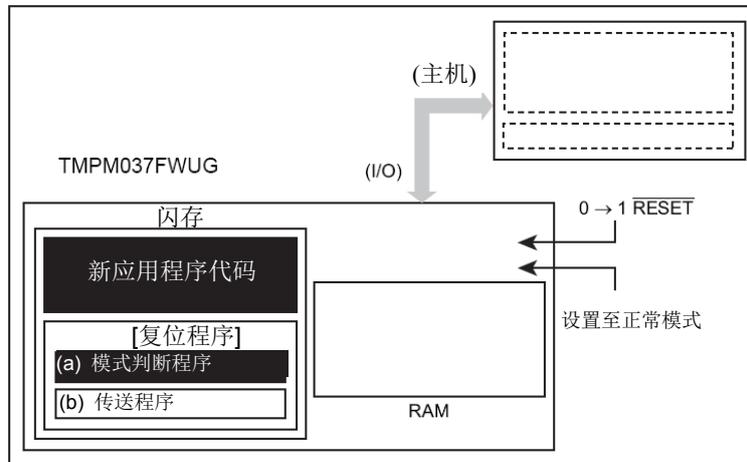
17.4.2.5 步骤-5

继续执行闪存编程路径(c)，以从主机控制器下载新的程序数据，并将其编程到已被擦除的闪存存储块内。在编程完成时，必须对用户程序区中闪存存储块的写入/擦除保护进行设置。



17.4.2.6 步骤-6

设 $\overline{\text{RESET}}$ 为“0”。一旦复位，闪存设为正常模式。复位后，CPU将随新应用程序启动。



18. 调试接口

18.1 规范概述

TMPM037FWUG带有用于连接调试工具的串行线JTAG调试端口(SWJ-DP)。有关SWJ-DP的详细资料，请参看“拟用于Cortex-M0的ARM文档”。

18.2 SWJ-DP

SWJ-DP支持串行线调试端口(SWCLK, SWDIO)。

引脚名称	功能	描述	I/O
SWDIO	承插焊	串行线数据输入/输出	I/O
SWCLK	承插焊	串行线时钟	输入

18.3 在停止模式时的外设功能

在Cortex-M0内核进入停机状态时，看门狗计时器(WDT)自动停止工作。16-位计时器(TMRB和TMR16A)可继续计数或停止计数。其它外设功能则继续工作。

18.4 与调试工具的连接

18.4.1 关于与调试工具的连接

与调试工具的连接，见制造商的建议。

调试接口引脚带有一个上拉电阻器和一个下拉电阻器。在调试接口引脚与外部上拉或下拉连接时，请输入电平。

注：严禁在STOP1 模式下连接调试工具来测量电耗。

18.4.2 将调试接口引脚用作通用端口时的要点

调试接口引脚也能用作通用端口。

复位解除后，各调试接口引脚特定引脚即被初始化为调试接口引脚。如有必要，则应将其它调试接口引脚改为调试接口引脚。

如果调试接口引脚被用作一般I/O端口，则应预先确定将一般I/O端口改为调试接口引脚的方法。

表 18-1 调试接口引脚使用示例表

	调试接口引脚	
	SWCLK	SWDIO
承插焊	o	o

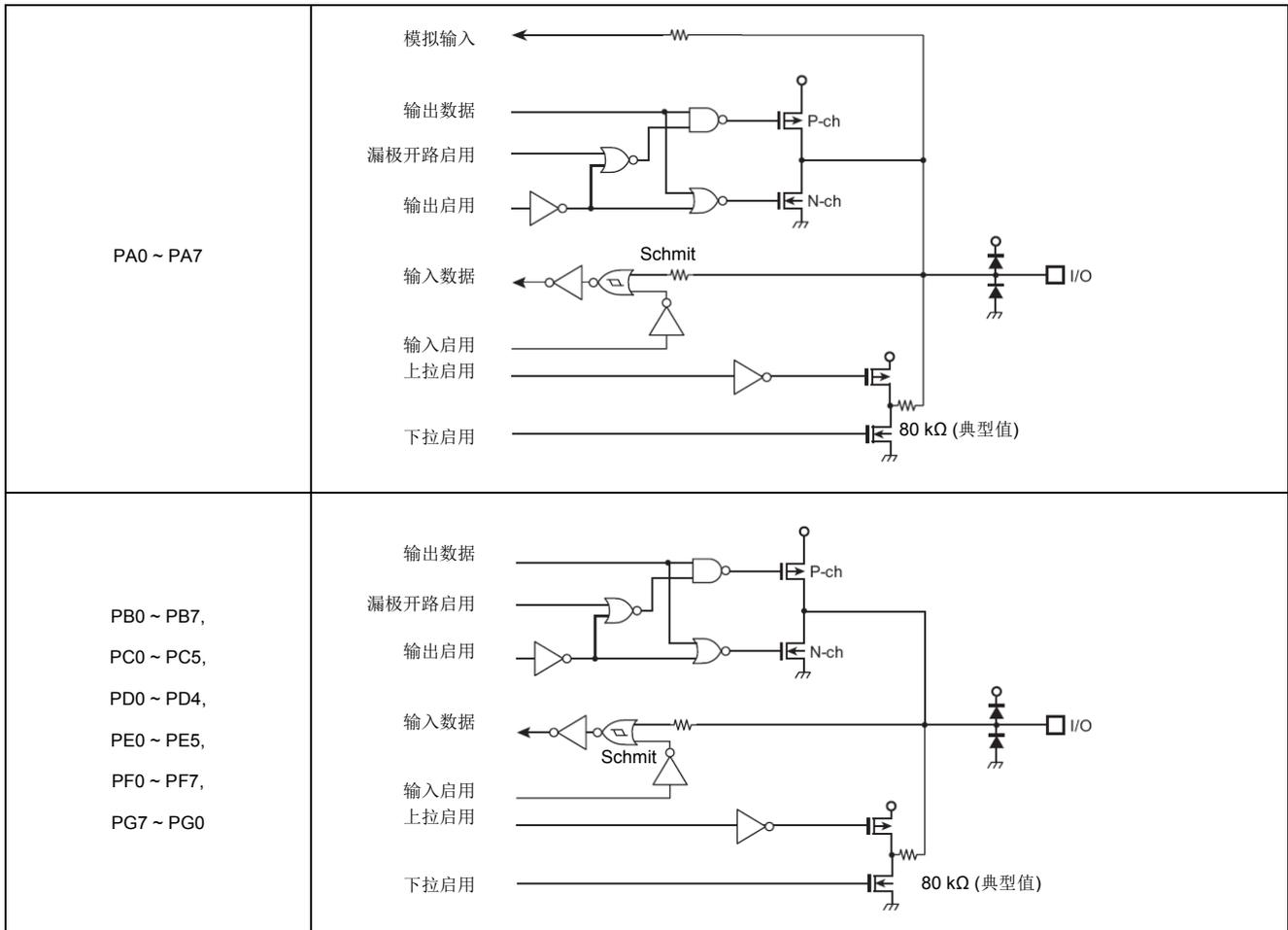
o: 启用 x: 禁用(可用作通用端口)

19. 端口部分等效电路示意图

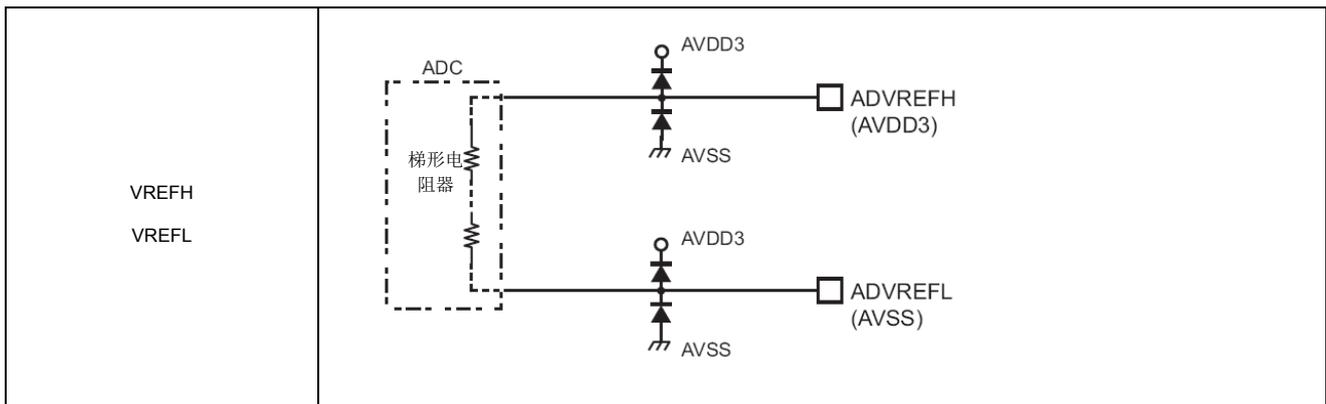
基本上，所写入的逻辑门符号和用于标准CMOS逻辑IC [74HCXX] 系列的相同。输入保护电阻有几十 Ω 至几百 Ω 不等。阻尼电阻器X2标注有典型值。

注：图中无值的电阻器则注明为输入保护电阻器。

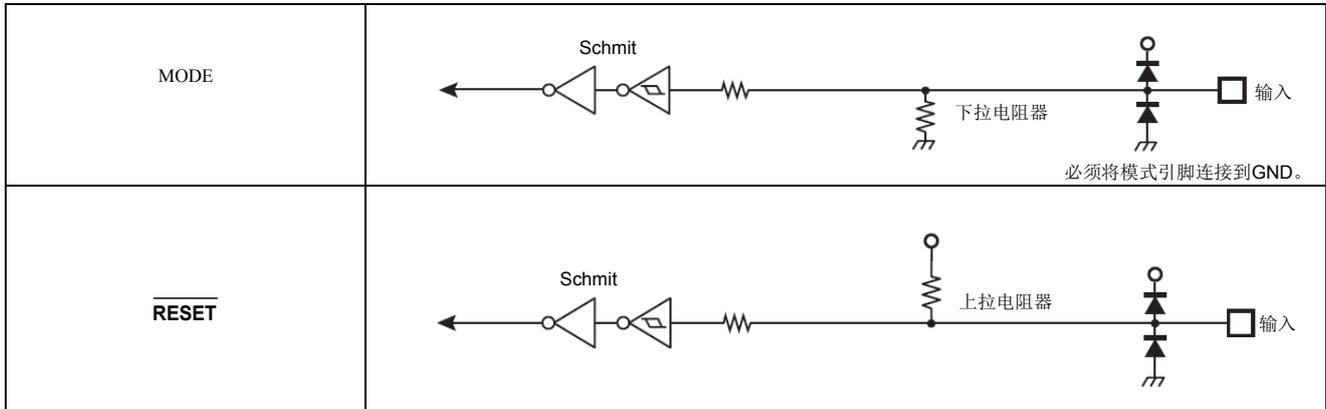
19.1 PORT 引脚



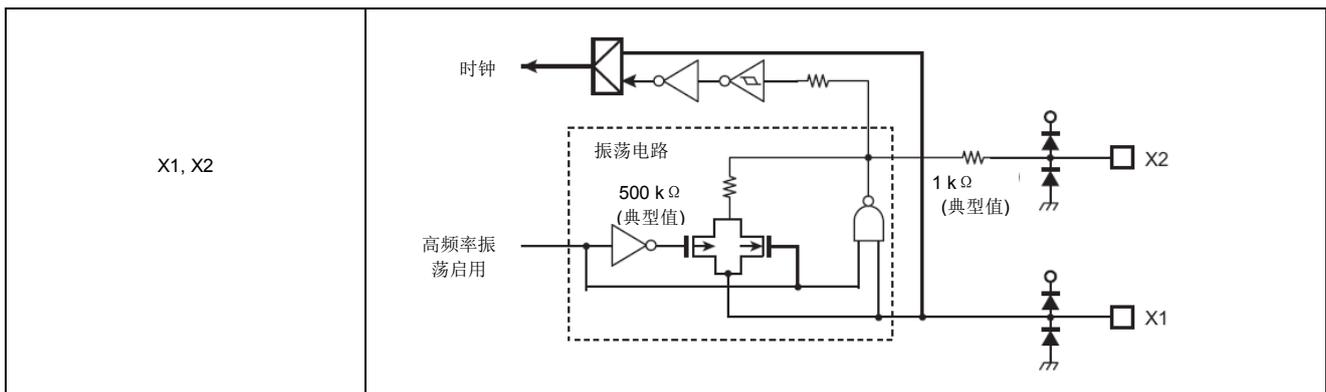
19.2 模拟引脚



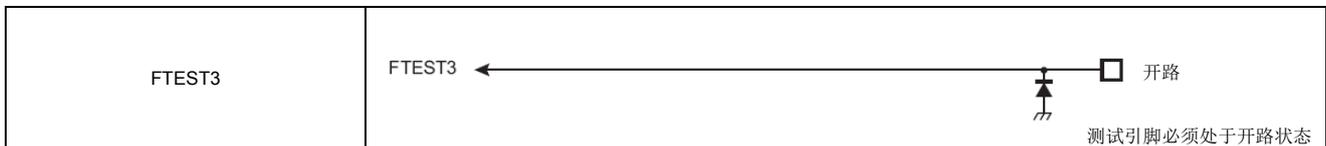
19.3 控制引脚



19.4 时钟引脚



19.5 测试引脚





20. 电气特性

20.1 绝对最大额定值

参数		符号	额定值	单位
电源电压		DVDD3	-0.3 ~ 3.9	V
		RVDD3	-0.3 ~ 3.9	
		AVDD3	-0.3 ~ 3.9	
输入电压	数字输入引脚	V _{IN1}	-0.3 ~ DVDD3 + 0.3	V
	模拟输入引脚	V _{IN2}	-0.3 ~ AVDD3 + 0.3	
低电平 输出电流	每个引脚(以下的除外)	I _{OL1}	5	mA
	PC2,PC3,PG6,PG7	I _{OL2}	20	
	合计	ΣI _{OL}	75	
高电平 输出电流	每个引脚(以下的除外)	I _{OH1}	-5	mA
	PC2,PC3,PG6,PG7	I _{OH2}	-20	
	合计	ΣI _{OH}	-75	
功耗 (Ta = 85 °C)		PD	600	mW
焊接温度(10 s)		T _{SOLDER}	260	°C
贮存温度		T _{STG}	-55 ~ 125	°C
工作温度		T _{OPR}	-40 ~ 85	°C

注：绝对最大额定值为最恶劣可能条件下不可超过的操作和环境条件极限值。设备制造厂商所采用的设计，应确保不超出任何绝对最大额定值，包括电流，电压，耗电，温度等。如暴露在超出上列值的条件下，则可能导致器件发生永久性损伤，或器件的可靠性受到影响，这可能增大相关人员因IC爆炸和/或燃烧而受伤的潜在风险。

20.2 DC 电气特性(1/2)

DVSS = RVSS = AVSS = 0 V
Ta = -40 ~ 85 °C

参数		符号	条件	最小值	典型值	最大值	单位
电源电压	DVDD3 RVDD3 AVDD3	DVDD3 RVDD3 AVDD3	fosc = 8 ~ 16 MHz fsys = 1 ~ 20 MHz	2.3	-	3.6	V
低电平输入电压	PB0 ~ 7, PC0 ~ 5, PD0 ~ 5, PE0 ~ 7, PF0 ~ PF7, PG0 ~ 7	V _{IL1}	Schmitt	-0.3	-	0.2 DVDD3	V
	PA0 ~ 7	V _{IL2}				0.2 AVDD3	
	X1, MODE, RESET	V _{IL3}				0.2 DVDD3	
高电平输入电压	PB0 ~ 7, PC0 ~ 5, PD0 ~ 5, PE0 ~ 7, PF0 ~ PF7, PG0 ~ 7	V _{IH1}	Schmitt	0.8 DVDD3	-	DVDD3+0.3	V
	PA0 ~ 7	V _{IH2}				AVDD3+0.3	
	X1, MODE, RESET	V _{IH3}				DVDD3+0.3	
低电平输出电压	PAx,PBx,PC0~1,PC4~5,PDx,PDx,PEx,PFx,PG0~5	V _{OL1}	I _{OL1} = 2 mA 2.3 ≤ DVDD3 ≤ 3.6V	-	-	0.4	V
	PC2,PC3,PG6,PG7	V _{OL2}	I _{OL2} = 10 mA 2.3 ≤ DVDD3 ≤ 3.6V			0.4	
高电平输出电压	PAx,PBx,PC0-1,PC4~5,PDx,PDx,PEx,PFx,PG0to5	V _{OH1}	I _{OH1} = -2 mA 2.3 ≤ DVDD3 ≤ 3.6V	DVDD3-0.4	-	DVDD3	V
	PC2,PC3,PG6,PG7	V _{OH2}	I _{OH2} = -10 mA 2.3 ≤ DVDD3 ≤ 2.7V	DVDD3-0.4	-	DVDD3	

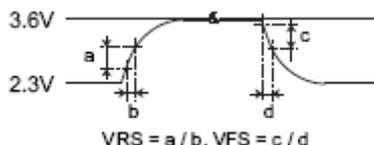
DVSS = RVSS = AVSS = 0V
Ta = -40 ~ 85 °C

参数	符号	条件	最小值	典型值(注1)	最大值	单位
输入泄漏流	-	$0.0 \leq V_{IN} \leq DVDD3$ $0.0 \leq V_{IN} \leq AVDD3$	-	0.02	±5	μA
	PC0, PC1	$0.0 \leq V_{IN} \leq DVDD3$ $0 \leq DVDD3 \leq 0.2$	-	-	±1.8	
输出泄漏电流	I _{LO}	$0.2 \leq V_{IN} \leq DVDD3 - 0.2$ $0.2 \leq V_{IN} \leq AVDD3 - 0.2$	-	0.05	±10	
Schmitt触发器输入宽度	VTH1	$2.7 V \leq DVDD3 \leq 3.6 V$	0.1 DVDD3	-	-	V
	VTH2	$2.3 V \leq DVDD3 \leq 2.7 V$	0.1 DVDD3	-		
复位时的上拉电阻器	RRST	$2.7 V \leq DVDD3 \leq 3.6 V$	25	50	75	kΩ
可编程上拉/下拉电阻器	PKH	$2.7 V \leq DVDD3 \leq 3.6 V$	25	50	75	kΩ
操作范围内的电源变化速率	VRS	RVDD3 = DVDD3	-	-	10	mV/μs
	VFS		-	-	-1.44	
引脚电容(电源引脚除外)	C _{IO}	f _c = 1 MHz	-	-	10	pF
低电平输出电流	I _{OL1}	每引脚: PAx,PBx,PC0~1,PC4~5, PDx,PEx,PF,PG0~5 $2.7 V \leq DVDD3 \leq 3.6 V$	-	-	2	mA
	I _{OL2}	每引脚: PC2,PC3,PG6,PG7 $2.7 V \leq DVDD3 \leq 3.6 V$	-	-	10	mA
	ΣI _{OL1}	每个端口:PA	-	-	10	mA
	ΣI _{OL2}	每个区: PBx,PC0~1,PEx,PGx	-	-	32	mA
	ΣI _{OL3}	每个区: PC2至5,PDx,PFx	-	-	32	mA
高电平输出电流	I _{OH1}	每引脚: PAx,PBx,PC0~1,PC4to5, PDx,PEx,PF,PG0~5 $2.7 V \leq DVDD3 \leq 3.6 V$	-	-	-2	mA
	I _{OH2}	每引脚: PC2,PC3,PG6,PG7 $2.7 V \leq DVDD3 \leq 3.6 V$	-	-	-10	mA
	ΣI _{OH1}	每个端口:PA	-	-	-10	mA
	ΣI _{OH2}	每个区: porPBx,PC0~1,PEx,PGx	-	-	-32	mA
	ΣI _{OH3}	每个区: PC2 ~ 5,PDx,PFx	-	-	-32	mA
输出电流	ΣI _O	合计, 所有端口	-	-	± 60	mA

注1: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = 3.3 V, 除非另有说明。

注2: DVDD3、RVDD3和AVDD3的供电电压必须相同。

注3: 作为特性, VRS(上升)和VFS(下降)应以严格标准进行测量。



20.3 DC 电气特性(2/2)

Ta = -40 ~ 85 °C

参数	符号	条件			最小值	典型值(注)	最大值	单位
		操作电压	高速振荡	工作条件				
NORMAL	IDD	DVDD3 = RVDD3 = AVDD3 = 3.6V	启用	所有外设功能均带有 CPU	-	-	13.1	mA
IDLE		DVDD3 = RVDD3 = AVDD3 = 3.3V	请参看表 20-1 按操作条件进行再分类		-	9	12	
STOP1			禁用	请参看表 20-1 按操作条件进行再分类	-	4.5	6.5	mA
						50	650	μA

(注: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = 3.3 V, 除非另有说明。)

表 20-1 IDD测量条件(引脚条件, 振荡器)

		NORMAL	IDEL	STOP1
引脚条件	DVDD3 = RVDD3 = AVDD3	3.3 V		
	X1, X2引脚	连接至高速振荡器 (10 MHz)		
	输入引脚	固定		
	输出引脚	开路		
操作条件 (振荡器)	系统时钟 (fsys)	20 MHz		禁用
	外部高速振荡器 (EHOSC)	启用		禁用
	内部高速振荡器 (IHOSC)	禁用		
	fsys的PLL	已启用(由2)		禁用
	外设功能	全部已启用	禁用	禁用

20.4 10-位 AD 转换器电气特性

AVSS = DVSS = 0 V, Ta = -40 ~ 85 °C

参数	符号	条件	最小值	典型值	最大值	单位
模拟参考电压(+)	AVDD	-	2.3	2.7	3.6	V
模拟输入电压	VAIN	-	AVSS	-	AVDD	V
模拟基准电压的	IREF	DVSS = AVSS	-	0.45	0.7	mA
电源电流			非AD转换	-	15	50
INL误差	-	AIN电阻 ≤ 300 Ω AIN负载电容 ≥ 0.1 μF 转换时间 ≥ 16.2 μs AVDD = 2.7~3.6 V	-	4.0	6.0	LSB
DNL误差			-	4.0	6.0	
零刻度误差			-	4.0	6.0	
满刻度误差			-	4.0	6.0	
总误差			-	4.0	6.0	
INL误差	-	AIN电阻 ≤ 300 Ω AIN负载电容 ≥ 0.1 μF 转换时间 ≥ 32.4 μs AVDD = 2.3~3.6 V	-	4.0	6.0	
DNL误差			-	4.0	6.0	
零刻度误差			-	4.0	6.0	
满刻度误差			-	4.0	6.0	
总误差			-	4.0	6.0	

注1: 1LSB = (AVDD - AVSS)/1024 [V]

注2: 特性仅在ADC运行时显现。

20.5 AC 电气特性

20.5.1 串行通道(SIO/UART)

20.5.1.1 AC 测量条件

可在以下条件下测量本节所述的AC特性。

- 输出电平：高 = $0.8 \times DVDD3$ ，低 = $0.2 \times DVDD3$
- 输入电平：高 = $0.8 \times DVDD3$ ，低 = $0.2 \times DVDD3$
- 负载容量：CL = 30 pF

20.5.1.2 AC 电气特性(I/O 接口模式)

下表中，字母x代表SIO操作时钟周期时间，和fsys周期时间相同。

(1) SCLK输入模式

[输入]

DVDD3=2.3 ~ 3.6 V

参数	符号	等式		fsys = 20 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK时钟高宽度(输入)	tSCH	4x	-	200	-	ns
SCLK时钟控制低宽度(输入)	tSCL	4x	-	200	-	
SCLK 周期	tSCY	tSCH + tSCL	-	400	-	
有效数据输入← SCLK上升/下降(注1)	tSRD	30	-	30	-	
SCLK上升/下降(注1)→ 输入数据保持	tHSR	x + 30	-	80	-	

[输出]

DVDD3=2.3 ~ 3.6 V

参数	符号	等式		fsys = 20 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK时钟高宽度(输入)	tSCH	4x	-	200 (注3)	-	ns
SCLK时钟控制低宽度(输入)	tSCL	4x	-	200 (注3)	-	
SCLK周期	tSCY	tSCH + tSCL	-	400	-	
输出数据← SCLK上升或下降(注1)	tOSS	tSCY/2 - 3x- 45	-	5 (注2)	-	
SCLK上升或下降→输出数据保持(注1)	tOHS	tSCY/2	-	105	-	

注1：SCLK上升/下降：SCLK上升模式使用SCLK的上升计时。SCLK下降模式使用SCLK的下降计时。

注2：使用在计算值保持为正的范围内SCLK的频率。

注3：值指示启用toss为零或模式的某个最小值。

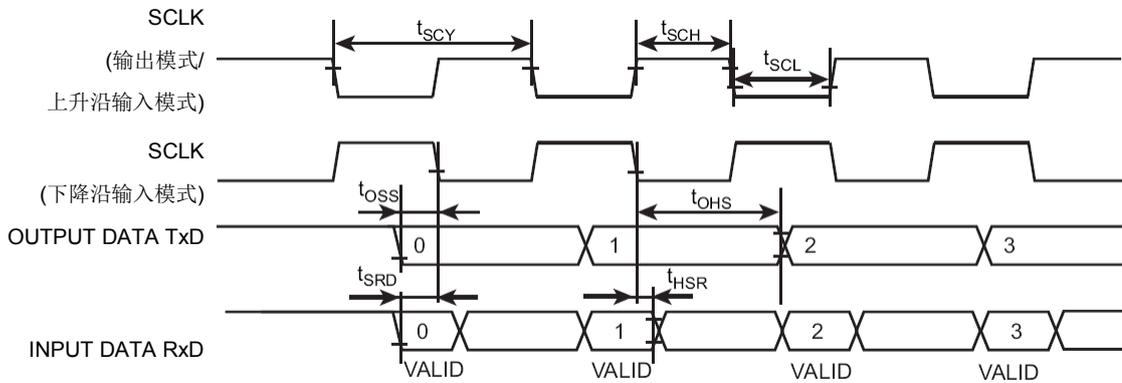
(2) SCLK输出模式

DVDD3=2.7~3.6 V

参数	符号	等式		f _{sys} = 20 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK周期(可编程)	t _{SCY}	4x	-	200	-	ns
输出数据←SCLK上升	t _{OSS}	t _{SCY} /2 - 30	-	70	-	
SCLK上升→输出数据保持	t _{OHS}	t _{SCY} /2 - 30	-	70	-	
有效数据输入←SCLK上升	t _{SRD}	45	-	45	-	
SCLK上升→输入数据保持	t _{HSR}	0	-	0	-	

DVDD3=2.3~2.7 V

参数	符号	等式		f _{sys} = 20 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK周期(可编程)	t _{SCY}	4x	-	200	-	ns
输出数据←SCLK上升	t _{OSS}	t _{SCY} /2 - 30	-	70	-	
SCLK上升→输出数据保持	t _{OHS}	t _{SCY} /2 - 50	-	50	-	
有效数据输入←SCLK上升	t _{SRD}	70	-	70	-	
SCLK上升→输入数据保持	t _{HSR}	0	-	0	-	



20.5.2 I2C 接口(I2C)

20.5.2.1 AC 测量条件

本节所述的AC特性数据均需在下述条件下测量，但另有描述的情形除外。

- 输出电平：高 = $0.8 \times DVDD3$ ，低 = $0.2 \times DVDD3$
- 输入电平：高 = $0.8 \times DVDD3$ ，低 = $0.2 \times DVDD3$
- 负载容量：CL = 30 pF

20.5.2.2 AC 电气特性

下表中，字母x代表I2C操作时钟周期时间，和fsys周期时间相同。其变化视时钟齿轮功能的编程而定。其变化视时钟齿轮功能的编程而定。

n表示被编程到I2CxCR1的SCK(SCL输出频率选择)字段中的n的值。

p表示被编程到I2CxPRS的PRsCK(刻度时钟选择)字段中的p的值。

参数	符号	等式		标准模式		快速模式		单位
		最小值	最大值	最小值	最大值	最小值	最大值	
SCL时钟频率	tSCL	0	-	0	100	0	400	kHz
START条件的保持定时器	tHD, STA	-	-	4.0	-	0.6	-	μs
SCL低宽度(输入)(注1)	tLOW	-	-	4.7	-	1.3	-	
SCL高宽度(输入)(注2)	tHIGH	-	-	4.0	-	0.6	-	
重复START条件的建立时间	tSU, STA	(注5)	-	4.7	-	0.6	-	
数据保持时间(输入)(注3, 4)	tHD, DAT	-	-	0.0	-	0.0	-	ns
数据建立时间	tSU, DAT	-	-	250	-	100	-	
STOP条件的建立时间	tSU, STO	-	-	4.0	-	0.6	-	μs
stop条件和start条件之间的总线空闲时间	tBUF	(注5)	-	4.7	-	1.3	-	

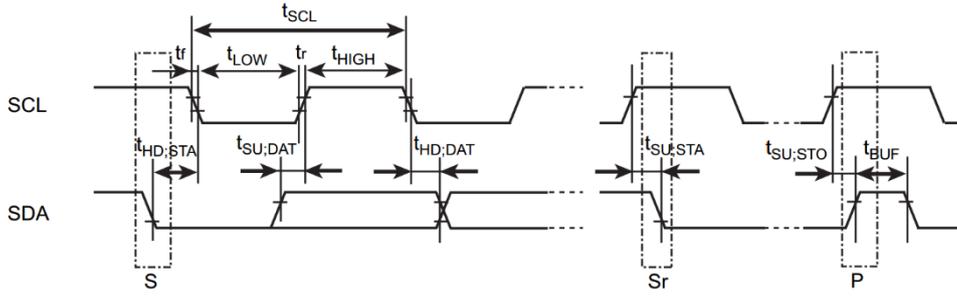
注1: SCL时钟低宽度(输出): $P \times (2^{n-1} + 6)/X$

注2: SCL时钟高宽度(输出): $P \times (2^{n-1} + 6)/X$ On I2C总线规格，标准模式的最高速度/快速模式为100 kHz/400 kHz。内部SCL频率设置应符合fsys和以上注1与注2的要求。(P: 取决于I2CxPROS<'RSCK'>, n: 取决于I2CxCR1<'SCK'>。)

注1: 输出数据保持时间等于预分频器时钟(Tprscck)从内部SCL的边缘起计的 4 个周期。

注4: 飞利浦I2C-总线规格为，设备应在内部提供相应保留时间最少为300 ns，来让SDA信号桥接SCL下降沿的未定义区。不过，I2C不符合本要求。此外，SCL的输出缓冲器也未采用下降沿的斜率控制装置；因此，设备制造厂商所采用的设计应确保满足表所述输入数据保持时间的要求，包括SCL和SDA线的tr/tf。

注1: 取决于软件



- S: 开始条件
- Sr: 重新启动条件
- P: 停止条件

20.5.3 16-位计时器/事件计数器(TMRB)

20.5.3.1 事件计数器

(1) AC测量条件

可在以下条件下测量本节所述的AC特性。

- 输入电平：高 = $0.8 \times DVDD3$, 低 = $0.2 \times DVDD3$
- 负载容量：CL = 30 pF

(2) AC电气特性

在下表中，字母x表示TMRB操作时钟周期时间。TMRB与f_{sys}使用同一时钟周期。因时钟齿轮功能编程而异。

参数	符号	等式		f _{sys} = 20 MHz		单位
		最小值	最大值	最小值	最大值	
时钟低脉宽	t _{vCKL}	2x + 100	-	200	-	ns
时钟高脉宽	t _{vCKH}	2x + 100	-	200	-	

20.5.3.2 捕捉

(1) AC测量条件

可在以下条件下测量本节所述的AC特性。

- 输入电平：高 = $0.8 \times DVDD3$, 低 = $0.2 \times DVDD3$
- 负载容量：CL = 30 pF

(2) AC电气特性

在下表中，字母x表示TMRB操作时钟周期时间。TMRB与f_{sys}使用同一时钟周期。因时钟齿轮功能编程而异。

参数	符号	等式		f _{sys} = 20 MHz		单位
		最小值	最大值	最小值	最大值	
低脉冲宽度	t _{CPL}	2x + 100	-	200	-	ns
高脉冲宽度	t _{CPH}	2x + 100	-	200	-	

20.5.4 外部中断

20.5.4.1 AC 测量条件

可在以下条件下测量本节所述的AC特性。

- 输入电平：高 = $0.8 \times DVDD3$ ，低 = $0.2 \times DVDD3$
- 负载容量：CL = 30 pF

20.5.4.2 AC 电气特性

下表中，字母x代表f_{sys}周期时间。

1. STOP1解除中断除外

参数	符号	等式		f _{sys} = 20 MHz		单位
		最小值	最大值	最小值	最大值	
低电平脉冲宽度	t _{INTAL}	x + 100	-	200	-	ns
高电平脉冲宽度	t _{INTAH}	x + 100	-	200	-	

2. STOP1解除中断

参数	符号	等式		f _{sys} = 20 MHz		单位
		最小值	最大值	最小值	最大值	
低电平脉冲宽度	t _{INTBL}	500	-	500	-	ns
高电平脉冲宽度	t _{INTBH}	500	-	500	-	

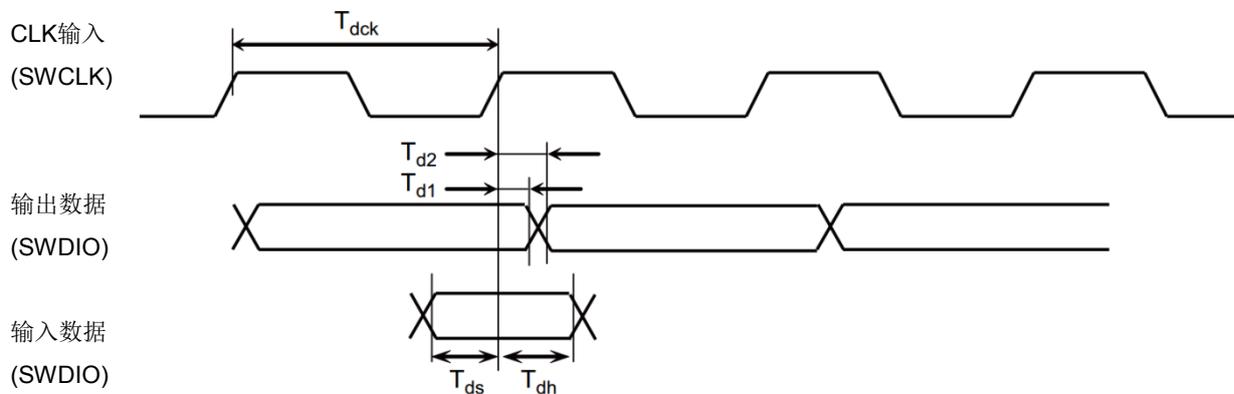
20.5.5 调试通信

20.5.5.1 AC 测量条件

- 输出电平：高 = $0.8 \times DVDD3$, 低 = $0.2 \times DVDD3$
- 输入电平：低 = $0.8 \times DVDD3$, 低 = $0.2 \times DVDD3$
- 负载容量：CL = 30 pF (SWDIO)

20.5.5.2 SWD 接口

参数	符号	最小值	最大值	单位
CLK周期	T_{dck}	100	-	ns
CLK上升→输出数据保留	T_{d1}	4	-	
CLK上升→至有效输出数据	T_{d2}	-	30	
有效输入数据→CLK上升	T_{ds}	20	-	
CLK上升→输入数据保持	T_{dh}	15	-	



20.5.6 片上振荡器特性

参数	符号	条件	最小值	典型值	最大值	单位
振荡频率	IHOSC	Ta = -40 ~ 85 °C	9.0	10	11	MHz

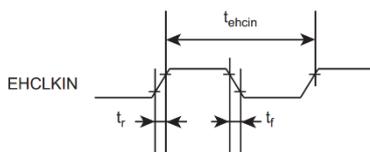
注：如必须使用高精度振荡频率，则不要将片装振荡器用作系统时钟。

20.5.7 外部振荡器

参数	符号	条件	最小值	典型值	最大值	单位
高频率振荡	EHOSC	Ta = -40 ~ 85 °C	8	-	20	MHz

20.5.8 外部时钟输入

参数	符号	最小值	典型值	最大值	单位
外部时钟频率	t_{ehcin}	8	-	20	MHz
外部时钟占空比	-	45	-	55	%
外部时钟输入上升时间	t_r	-	-	10	ns
外部时钟输入下降时间	t_f	-	-	10	ns



20.5.9 闪存特性

参数	条件	最小值	典型值	最大值	单位
闪存编程的保证数目	DVDD3 = RVDD3 = AVDD3 = 2.7 V ~ 3.6 V Ta = 0 ~ 70°C	-	-	100	次

20.5.10 噪声滤波器特性

参数	条件	最小值	典型值	最大值	单位
噪声滤波器的宽度	-	15	30	60	ns

20.6 建议采用的振荡电路

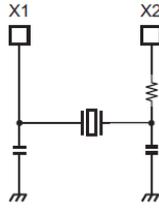


图 20-1 高频振荡连接

注：必须正确配置振荡器的负荷能力和位置，才能获得稳定振荡。由于这些因素可受到基板模板的严重影响，请使用用户所使用的基板评价振荡的稳定性。

以下振荡器供应商已对TMPM037FWUG进行过评价。在选择外部零件时，请参照本资料。

20.6.1 陶瓷振荡器

TMPM037FWUG建议采用村太制造股份有限公司出品的高频振荡器。详情请参阅以下URL
<http://www.murata.co.jp>

20.6.2 印刷电路板设计时的注意事项

所设计印刷电路板的晶体单元与其它振荡元件的连接方式，必须确保模板的长度尽可能的短，以防止因杂散电容和布线电感导致器特性发生劣化。对于多层线路板而言，不得将地线和其它信号式样接在振荡电路的正下方。有关详细资料，请参看振荡器供应商的URL。

20.7 使用注意事项

20.7.1 通电电源电压电平

上电时的电源上升量必须小于下表中所列的值。TMPM037FWUG具备一些电源引脚。它们须同时供电。

电源引脚 = DVDD3, AVDD3, RVDD3

Ta = -40 ~ 85 °C

参数	条件	最小值	典型值	最大值	单位
通电时的电源上升	0 V → 2.3 V ~ 3.6 V	-	-	10	mV/μs

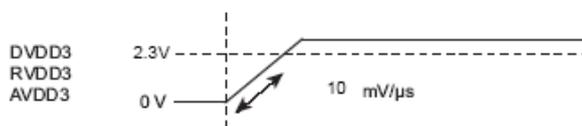


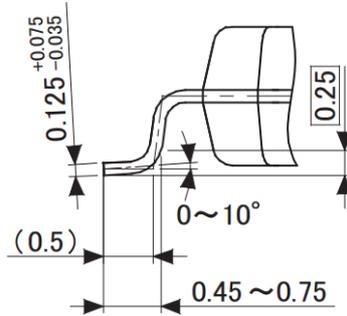
图 20-2 上电时的电源电压电平

20.7.2 操作期间发生的电压下降现象

在操作期间发生电源电压下降时，如果电源电压低于操作电压(电压过低)，则重新上电。

引脚详图

单位: mm



RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

译文
