

译文

TMPM061FWFG

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: "TMPM061FWFG" 2014-03-10

翻译日: 2015-06-30

译文

TOSHIBA

32 位RISC微控制器
TX00系列

TMPM061FWFG

东芝公司

半导体&存储产品公司

译文

ARM, ARM Powered, AMBA, ADK, ARM9TDMI, TDMI, PrimeCell, RealView, Thumb, Cortex, Coresight, ARM9, ARM926EJ-S, 嵌入式跟踪宏单元, ETM, AHB, APB, KEIL是ARM 有限公司在EU及其它国家的注册商标或商标。

重要注意事项

使用该产品前，请务必阅读本章内容。

1 串行总线接口

使用多主机功能时，对I2C总线模式的使用存在一些限制。

1.1 描述

在I2C总线模式下使用多主功能时，如果这些主设备同时启动通信，则可能会出现以下现象：

1. 通信会被锁住。
2. SCL脉冲宽度缩短；因此，这些脉冲未必满足I2C规范。

1.2 条件

仅在I2C总线模式下使用多主功能时出现这些现象。如使用单主，则不会出现这些现象。

1.3 解决办法

目前没有针对这些现象的解决办法。可通过软件执行恢复进程。

1.4 出现这些现象时如何恢复

可通过软件执行恢复进程。

通过使用计时器，添加超时过程以检查通信是否处于锁定状态。

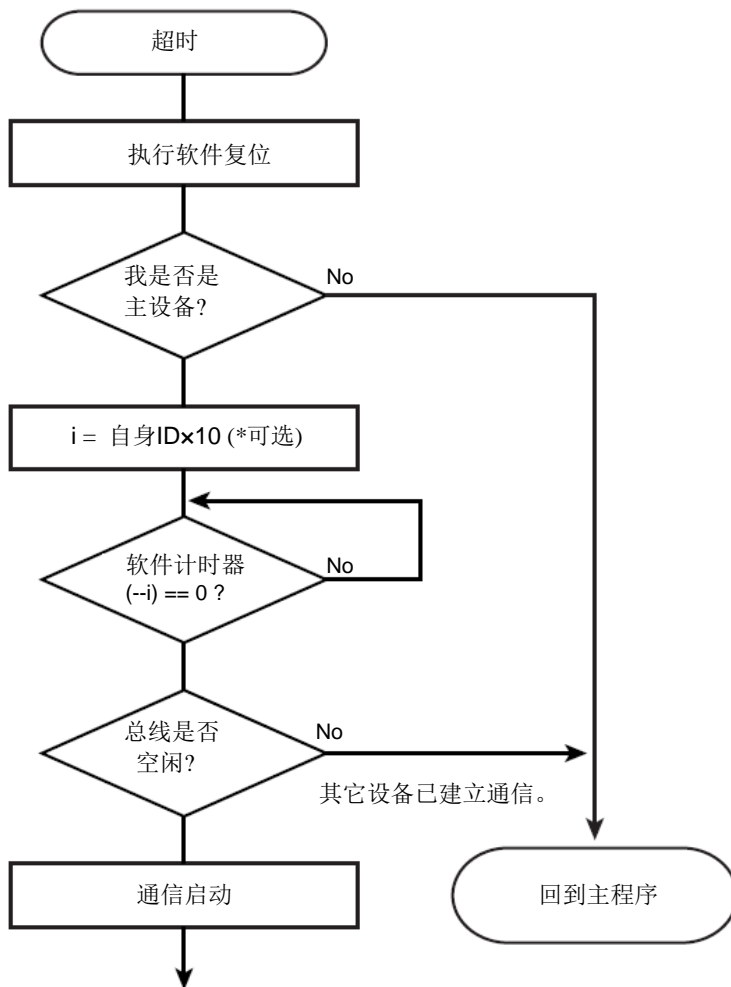
恢复进程示例：

1. 启动计时器计数，同时启动发送。
2. 如果在某个周期内未发生串行接口中断(INTSBIx)，则MCU可测定该超时。
3. 当MCU测定该超时，则通信会被锁住。对接口电路执行软件复位。该电路即被初始化，以解除通信的锁定状态。
4. 重新发送传输数据。

通常，进程 1 ~ 4 足以实现恢复；不过，如果同一总线上接有多个产品，则应在执行进程 4 (重新发送数据)前，往各产品的恢复进程之间应各新增一段延迟时间。该延迟可在各主设备之间形成时间差，可避免在重新发送数据时发生总线冲突。

译文

示例：超时被检测到之后的恢复进程。



介绍：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地址见存储器地址章节，SFR详见各IP章节。

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章节中的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)

- 各寄存器基本由 32-位寄存器组成(有些例外)。
- 每个寄存器对位，比特符号，类型，复位后的初值以及功能进行了描述。

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	可读取 "0"。
9-7	MODE[2:0]	R/W	工作模式设置 000: 样品模式 0 001: 样品模式 1 010: 样品模式 2 011: 样品模式 3 非上述设置: 保留
6-0	TDATA[6:0]	W	发送的数据

注: 类型分成下列三类。

R / W	READ WRITE
R	READ
W	WRITE

c. 数据描述

SFR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>
示例: SAMCR<MODE> = "000"或SAMCR<MODE[2:0]> = "000"
<MODE[2:0]>指在比特符号模式时位 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

译文

修订记录

日期	修订版	备注
2012/2/4	试行版 1	试行版的第一版
2012/2/8	试行版 2	修订内容
2012/6/26	试行版 3	修订内容
2013/8/2	1	第一版
2014/3/10	2	修订内容

目录

概述:关于介绍：关于本规范中SFR(特殊功能寄存器)描述的说明

TMPM061FWFG

1.1 特征	1
1.2 方块图	4
1.3 引脚布置图(顶视图).....	6
1.4 引脚名称与功能	7
1.5 引脚编号和电源引脚	12

2 产品信息

2.1 16-位计时器/事件计数器(TMRB)	13
2.2 16-位计时器 A (TMR16A)	13
2.3 串行通道(SIO/UART)	14
2.4 模拟/数字转换器(ADC).....	15
2.4.1 非可用功能	15
2.4.2 转换通道	15
2.5 $\Delta\Sigma$ 模拟/数字转换器(DSADC).....	16

3. 处理器内核

3.1 处理器内核该相关信息	17
3.2 可配置选项	17
3.3 异常/中断	18
3.3.1 中断输入的数目	18
3.3.2 SysTick	18
3.3.3 SYSRESETREQ	18
3.3.4 LOCKUP	18
3.4 事件	18
3.5 电源管理	18

4. 存储器地址

4.1 存储器地址	20
4.2 总线结构	22
4.3 外设功能的地址列表	23

5. 复位工作	
5.1 冷复位	24
5.2 热复位	24
5.3 复位后	26

6. 时钟/模式控制	
6.1 特征	27
6.2 寄存器	28
6.2.1 寄存器列表	28
6.2.2 CGSYSCR(系统控制寄存器)	28
6.2.3 CGOSCCR(振荡控制寄存器)	30
6.2.4 CGSTBYCR(待机控制寄存器)	32
6.2.5 CGEHCLKSEL (外部高速时钟选择寄存器)	33
6.2.6 CGCKSEL(系统时钟选择寄存器)	33
6.3 时钟控制	34
6.3.1 时钟类型	34
6.3.2 复位后初始值	34
6.3.3 时钟系统图	35
6.3.4 预热功能	36
6.3.5 系统时钟	37
6.3.5.1 系统时钟切换	
6.3.5.2 转换为 SLOW, STOP 和 SLEEP 模式时的注意事项	
6.3.5.3 时钟设置	
6.3.6 预分频器时钟控制	42
6.3.7 系统时钟引脚输出功能	42
6.4 模式与模式转换	44
6.4.1 模式转换	44
6.5 工作模式	45
6.5.1 NORMAL 模式	45
6.5.2 SLOW 模式	45
6.6 低功耗模式	45
6.6.1 IDLE 模式	46
6.6.2 SLEEP 模式	46
6.6.3 STOP 模式	46
6.6.4 低功耗模式设置	47
6.6.5 各模式下的工作状态	48
6.6.6 解除低功耗模式	49
6.6.7 预热	50
6.6.8 模式转换时的时钟工作	51
6.6.8.1 工作模式的转换: NORMAL → STOP → NORMAL	
6.6.8.2 工作模式的转换: NORMAL → SLEEP → NORMAL	
6.6.8.3 工作模式的转换: SLOW → STOP → SLOW	
6.6.8.4 工作模式的转换: SLOW → SLEEP → SLOW	

7. 异常	
7.1 概述	54
7.1.1 异常类型	54
7.1.2 处理流程图	55
7.1.2.1 异常请求与检测	
7.1.2.2 异常处理和分支转移至中断服务程序(抢占)	
7.1.2.3 执行 ISR	
7.1.2.4 异常出口	
7.2 复位异常	59
7.3 非屏蔽中断(NMI)	59

7.4 SysTick	59
7.5 中断	60
7.5.1 中断源	60
7.5.1.1 中断路径	
7.5.1.2 生成	
7.5.1.3 传输	
7.5.1.4 使用外部中断引脚时的注意事项	
7.5.1.5 中断源列表	
7.5.1.6 激活电平	
7.5.2 中断处理	64
7.5.2.1 流程图	
7.5.2.2 准备	
7.5.2.3 通过时钟生成器检测	
7.5.2.4 CPU 检测	
7.5.2.5 CPU 处理	
7.5.2.6 中断服务程序(ISR)	
7.6 与异常/中断相关的寄存器	70
7.6.1 寄存器列表.....	70
7.6.2 NVIC 寄存器	71
7.6.2.1 SysTick 控制与状态寄存器	
7.6.2.2 SysTick 重新加载值寄存器	
7.6.2.3 SysTick 当前值寄存器	
7.6.2.4 SysTick 校准值寄存器	
7.6.2.5 中断设置-启用寄存器	
7.6.2.6 中断清除清除-使能寄存器	
7.6.2.7 中断设置-挂起寄存器	
7.6.2.8 中断清除-挂起寄存器	
7.6.2.9 中断优先级寄存器	
7.6.2.10 应用中断和复位控制寄存器	
7.6.2.11 系统处理程序优先级寄存器	
7.6.2.12 系统处理器控制和状态寄存器	
7.6.3 时钟生成器寄存器.....	81
7.6.3.1 CGIMCGA(CG 中断模式控制寄存器 A)	
7.6.3.2 CGIMCGB(CG 中断模式控制寄存器 B)	
7.6.3.3 CGICRCG(CG 中断请求清除寄存器)	
7.6.3.4 CGRSTFLG (复位标志寄存器)	

8. 输入/输出端口

8.1 端口功能	86
8.1.1 功能列表	86
8.2 端口寄存器一般说明	89
8.2.1 PxDATA : 端口 x 数据寄存器.....	89
8.2.2 PxCR : 端口 x 输出控制寄存器.....	89
8.2.3 PxFRn: 端口 x 功能寄存器 n	90
8.2.4 PxOD : 端口 x 漏极开路控制寄存器.....	91
8.2.5 PxPUP : 端口 x 上拉控制寄存器	91
8.2.6 PxPDN : 端口 x 下拉控制寄存器.....	92
8.2.7 PxIE : 端口 x 输入控制寄存器	92
8.3 寄存器列表	94
8.4 功能详细	95
8.4.1 端口 A	95
8.4.2 端口 B	95
8.4.3 端口 C	96
8.4.4 端口 D	96
8.4.5 端口 E.....	97
8.4.6 端口 F.....	97
8.4.7 端口 G	98
8.4.8 端口 H	98
8.4.9 端口 I.....	99
8.4.10 端口 J.....	99
8.4.11 端口 K.....	100
8.5 端口方块图	101
8.5.1 端口类型	101
8.5.2 类型 FT1	102
8.5.3 类型 FT2	103

8.5.4 类型 FT3	104
8.5.5 类型 FT4	105
8.5.6 类型 FT5	106
8.5.7 类型 FT6	107
8.5.8 类型 FT7	108
8.5.9 类型 FT8	109
8.5.10 类型 FT9	110
8.5.11 类型 FT10	111
8.6 附录(端口设置列表)	112
8.6.1 I/O 端口的设置	112
8.6.2 输入专用端口的设置	112
8.6.3 输出专用端口的设置	112
8.6.4 外设 I/O 端口的设置	113
8.6.4.1 端口 A 设置	
8.6.4.2 端口 B 设置	
8.6.4.3 端口 C 设置	
8.6.4.4 端口 D 设置	
8.6.4.5 端口 E 设置	
8.6.4.6 端口 F 设置	
8.6.4.7 端口 G 设置	
8.6.4.8 端口 H 设置	
8.6.4.9 端口 I 设置	
8.6.4.10 端口 J 设置	
8.6.4.11 端口 K 设置	

9. 16-位计时器/事件计数器(TMRB)

9.1 概要	121
9.2 方块图	122
9.3 寄存器	123
9.3.1 寄存器列表	123
9.3.2 TBxEN (启用寄存器)	124
9.3.3 TBxRUN (RUN 寄存器)	125
9.3.4 TBxCR (控制寄存器)	126
9.3.5 TBxMOD (模式寄存器)	127
9.3.6 TBxFFCR (触发器控制寄存器)	128
9.3.7 TBxST (状态寄存器)	129
9.3.8 TBxIM (中断屏蔽寄存器)	130
9.3.9 TBxUC (上升计数器捕获寄存器)	131
9.3.10 TBxRG0 (计时器寄存器 0)	132
9.3.11 TBxRG1 (计时器寄存器 1)	132
9.3.12 TBxCP0(捕获寄存器 0)	133
9.3.13 TBxCP1(捕获寄存器 1)	133
9.4 工作说明	134
9.4.1 预分频器	134
9.4.2 上升计数器(UC)	134
9.4.2.1 源时钟	
9.4.2.2 计数器启动/停止	
9.4.2.3 计时至清除 UC	
9.4.2.4 UC 溢出	
9.4.3 计时器寄存器(TBxRG0, TBxRG1)	135
9.4.4 捕获控制	135
9.4.5 捕获寄存器(TBxCP0, TBxCP1)	135
9.4.6 上升计数器捕获寄存器(TBxUC)	136
9.4.7 比较器(CP0, CP1)	136
9.4.8 计时器触发器(TBxFF0)	136
9.4.9 捕获中断(INTCAPx0, INTCAPx1)	136
9.5 各模式工作说明	137
9.5.1 间隔计时器模式	137
9.5.2 事件计数器模式	137
9.5.3 可编程脉冲发生(PPG)输出模式	138
9.5.4 可编程脉冲发生(PPG)外部触发器输出模式	140
9.6 使用捕获功能的应用	142
9.6.1 频率测量	142

9.6.2 脉冲宽度测量.....	144
-------------------	-----

10. 16-位计时器 A (TMR16A)

10.1 概要	147
10.2 方块图	147
10.3 寄存器	148
10.3.1 寄存器列表.....	148
10.3.2 寄存器的详细.....	148
10.3.2.1 T16AxEN (启用寄存器)	
10.3.2.2 T16AxRUN (RUN 寄存器)	
10.3.2.3 T16AxCR (控制寄存器)	
10.3.2.4 T16AxRG (计时器寄存器)	
10.3.2.5 T16AxCP (捕获寄存器)	
10.4 工作说明	152
10.4.1 计时器运行.....	152
10.4.2 T16AxOUT 控制	152
10.4.3 读取捕获.....	152
10.4.4 自动停止.....	152

11. 串行通道(SIO/UART)

11.1 概述.....	154
11.2 配置.....	155
11.3 寄存器描述.....	156
11.3.1 寄存器列表.....	156
11.3.2 SCxEN (启用寄存器).....	157
11.3.3 SCxBUF (缓冲器寄存器).....	157
11.3.4 SCxCR (控制寄存器).....	159
11.3.5 SCxMOD0 (模式控制寄存器 0).....	160
11.3.6 SCxMOD1 (模式控制寄存器 1).....	161
11.3.7 SCxMOD2 (模式控制寄存器 2).....	162
11.3.8 SCxBRCR (波特率发生器控制寄存器), SCxBRADD (波特率发生器控制寄存器 2)	164
11.4 各模式下的工作.....	166
11.5 数据格式.....	167
11.5.1 数据格式列表.....	167
11.5.2 奇偶校验控制.....	168
11.5.2.1 传输	
11.5.2.2 接收数据	
11.5.3 STOP 位长度	168
11.6 时钟控制.....	169
11.6.1 预分频器.....	169
11.6.2 串行时钟生成电路.....	169
11.6.2.1 波特率发生器	
11.6.2.2 时钟选择电路	
11.6.3 传输/接收缓冲器.....	173
11.6.3.1 配置	
11.6.3.2 传输/接收缓冲器	
11.7 状态标志.....	174
11.8 错误标志.....	174
11.8.1 OERR 标志	174
11.8.2 PERR 标志.....	174
11.8.3 FERR 标志.....	175
11.9 接收	176
11.9.1 接收计数器.....	176
11.9.2 接收控制单元.....	176
11.9.2.1 I/O 接口模式	
11.9.2.2 UART 模式	
11.9.3 接收工作.....	176

11.9.3.1 接收缓冲器	
11.9.3.2 带有 SCLK 输出的 I/O 接口模式	
11.9.3.3 读取所接收的数据	
11.9.3.4 唤醒功能	
11.9.3.5 溢出错误	
11.10 传输	178
11.10.1 传输计数器	178
11.10.2 传输控制	178
11.10.2.1 I/O 接口模式	
11.10.2.2 UART 模式	
11.10.3 传输工作	178
11.10.3.1 传输缓冲器的工作	
11.10.3.2 I/O 接口模式/通过 SCLK 输出进行传输	
11.10.3.3 欠载运行错误	
11.11 握手功能	180
11.12 使用 IR 载波脉冲输出信号	181
11.13 中断/错误生成计时	182
11.13.1 RX 中断	182
11.13.2 TX 中断	182
11.13.3 错误生成	183
11.13.3.1 UART 模式	
11.13.3.2 I/O 接口模式	
11.14 软件复位	183
11.15 各模式下的工作	184
11.15.1 模式 0 (I/O 接口模式)	184
11.15.1.1 数据传输	
11.15.1.2 接收	
11.15.1.3 传输和接收(全双工)	
11.15.2 模式 1 (7-位 UART 模式)	196
11.15.3 模式 2 (8-位 UART 模式)	196
11.15.4 模式 3 (9-位 UART 模式)	197
11.15.4.1 唤醒功能	
11.15.4.2 协议	

12. 串行总线接口(I2C/SIO)

12.1 概要	199
12.2 方块图	199
12.3 I2C 总线模式数据格式	200
12.4 寄存器	201
12.4.1 寄存器列表	201
12.5 I2C 总线模式下的控制寄存器	202
12.5.1 SBIxCR0(控制寄存器 0)	202
12.5.2 SBIxCR1(控制寄存器 1)	202
12.5.3 SBIxCR2(控制寄存器 2)	204
12.5.4 SBIxSR (状态寄存器)	205
12.5.5 SBIxBR0(串行总线接口波特率寄存器 0)	206
12.5.6 SBIxDBR (串行总线接口数据缓冲寄存器)	206
12.5.7 SBIxI2CAR (I2C 总线地址寄存器)	207
12.6 I2C 总线模式下的控制	208
12.6.1 设置工作模式	208
12.6.2 串行时钟	208
12.6.2.1 时钟源	
12.6.2.2 时钟同步	
12.6.3 设置应答模式	209
12.6.4 设置每次发送的位数	209
12.6.5 从属寻址和地址识别模式	210
12.6.6 将 SBI 作为主设备模式或从设备模式进行配置	210
12.6.7 将 SBI 配置为发送器或接收器	210
12.6.7.1 主设备模式	
12.6.7.2 从设备模式	
12.6.8 总线忙监控器	211
12.6.9 中断服务请求和解除	212

12.6.10 仲裁丢失检测监控器	212
12.6.11 从设备地址匹配检测监控器	213
12.6.12 一般呼叫检测监控器	213
12.6.13 最后接收位监控程序	213
12.6.14 数据缓冲寄存器(SBIXDBR)	214
12.6.15 波特率寄存器(SBIXBR0)	214
12.6.16 软件复位	214
12.7 I2C 总线模式下的数据传送规程	215
12.7.1 设备初始化	215
12.7.2 生成启动条件和从设备地址	215
12.7.3 传送数据字	216
12.7.3.1 主模式(<MST> = "1")	
12.7.3.2 从设备模式 (<MST> = "0")	
12.7.4 生成停止条件	221
12.7.5 重启程序	222
12.8 SIO 模式控制寄存器	224
12.8.1 SBIXCR0(控制寄存器 0)	224
12.8.2 SBIXCR1(控制寄存器 1)	225
12.8.3 SBIXDBR (数据缓冲寄存器)	226
12.8.4 SBIXCR2(控制寄存器 2)	227
12.8.5 SBIXSR (状态寄存器)	228
12.8.6 SBIXBR0 (波特率寄存器 0)	229
12.9 SIO 模式控制	230
12.9.1 串行时钟	230
12.9.1.1 时钟源	
12.9.1.2 移位沿	
12.9.2 传送模式	232
12.9.2.1 8-位传输模式	
12.9.2.2 8-位接收模式	
12.9.2.3 8-位传输/接收模式	
12.9.2.4 传输结束时最后位的数据保留时间	

13. 10-位模拟/数字转换器(ADC)

13.1 概述	237
13.2 配置	237
13.3 寄存器	239
13.3.1 寄存器列表	239
13.3.2 ADCLK (转换时钟设置寄存器)	240
13.3.3 ADMOD0 (模式控制寄存器 0)	241
13.3.4 ADMOD1 (模式控制寄存器 1)	242
13.3.5 ADMOD2 (模式控制寄存器 2)	243
13.3.6 ADMOD3 (模式控制寄存器 3)	245
13.3.7 ADMOD4 (模式控制寄存器 4)	246
13.3.8 ADMOD5 (模式控制寄存器 5)	247
13.3.9 ADREGn (转换结果寄存器 n: n = 0 ~11)	248
13.3.10 ADREGSP (AD 转换结果寄存器 SP)	249
13.3.11 ADCMP0 (AD 转换结果比较寄存器 0)	250
13.3.12 ADCMP1 (AD 转换结果比较寄存器 1)	250
13.4 工作描述	251
13.4.1 模拟参考电压	251
13.4.2 AD 转换模式	251
13.4.2.1 普通 AD 转换	
13.4.2.2 最高优先级 AD 转换	
13.4.3 AD 监控功能	252
13.4.4 选择输入通道	253
13.4.5 AD 转换细节	253
13.4.5.1 启动 AD 转换	
13.4.5.2 AD 转换	
13.4.5.3 标准 AD 转换期间的最高优先级 AD 转换	
13.4.5.4 停止重复转换模式	
13.4.5.5 重新激活标准 AD 转换	
13.4.5.6 转换完成	
13.4.5.7 中断生成计时和 AD 转换的结果存储寄存器	

14. 24-位 $\Delta\Sigma$ 模拟/数字转换器 (DSADC)

14.1 特征	258
14.2 方块图	259
14.3 寄存器	260
14.3.1 寄存器列表	260
14.3.2 寄存器详细	260
14.3.2.1 DSADCLK (转换时钟设置寄存器)	
14.3.2.2 DSADCR0(控制寄存器 0)	
14.3.2.3 DSADCR1(控制寄存器 1)	
14.3.2.4 DSADCR2(控制寄存器 2)	
14.3.2.5 DSADCR3(控制寄存器 3)	
14.3.2.6 DSADCR4(控制寄存器 4)	
14.3.2.7 DSADADJ (校正寄存器)	
14.3.2.8 DSADST (转换状态寄存器)	
14.3.2.9 DSADRES (转换结果寄存器)	
14.4 工作说明	267
14.4.1 启动和停止步骤	267
14.4.1.1 启动	
14.4.1.2 停止	
14.4.2 转换时钟(ADCLK).....	271
14.4.2.1 转换时间	
14.4.2.2 转入低功耗模式	
14.4.3 转换模式.....	271
14.4.4 转换启动.....	271
14.4.5 转换状态.....	271
14.4.6 转换停止.....	272
14.4.7 转换结束.....	272
14.4.8 转换结果.....	272
14.5 同步启动功能	273
14.5.1 启动	273
14.5.2 停止	275
14.6 转换启动校正功能	275

15 温度传感器(TEMP)

15.1 概述	276
15.2 方块图	276
15.3 寄存器	277
15.3.1 寄存器列表.....	277
15.3.2 寄存器详细	277
15.3.2.1 TEMPEN(启用寄存器)	
15.3.2.2 TEMPCR(控制寄存器)	
15.4 工作说明	279

16. 实时时钟(RTC)

16.1 功能	280
16.2 方块图	280
16.3 寄存器详细描述	281
16.3.1 寄存器列表.....	281
16.3.2 控制寄存器.....	281
16.3.3 控制寄存器的详细描述	283
16.3.3.1 RTCSECR (秒钟列寄存器(仅适用于 PAGE0))	
16.3.3.2 RTCMINR (分钟列寄存器(PAGE0/1))	
16.3.3.3 RTCHOURR (小时列寄存器(PAGE0/1))	
16.3.3.4 RTCDAYR(周日列寄存器(PAGE0/1))	
16.3.3.5 RTCDATER (日列寄存器(仅适用于 PAGE0/1))	

16.3.3.6 RTCMONTHR (月列寄存器(仅适用于 PAGE0))	
16.3.3.7 RTCMONTHR(24-小时时钟或 12-小时时钟的选择(仅适用于 PAGE1))	
16.3.3.8 RTCYEARR(年份列寄存器(仅适用于 PAGE0))	
16.3.3.9 RTCYEARR(闰年寄存器(仅适用于 PAGE1))	
16.3.3.10 RTCPAGER(PAGE 寄存器(PAGE0/1))	
16.3.3.11 RTCRESTR(复位寄存器(适用于 PAGE0/1))	
16.3.3.12 RTCPROTECT(保护寄存器)	
16.3.3.13 RTCADJCTL(校正功能控制寄存器)	
16.3.3.14 RTCADJDAT(校正值寄存器)	
16.4 工作说明	292
16.4.1 读取时钟数据	292
16.4.2 写入时钟数据	292
16.4.3 进入低功耗模式	294
16.5 报警功能	295
16.5.1 报警功能的使用	295
16.5.2 1, 2, 4, 8 或 16 Hz 周期“低”脉冲	296
16.6 时钟校正功能	297
16.7 1Hz 时钟输出功能	297

17. LCD 驱动器

17.1 配置	299
17.2 寄存器	300
17.2.1 寄存器列表	300
17.2.2 寄存器详细	301
17.2.2.1 LCDEN (启用寄存器)	
17.2.2.2 LCDCR1 (控制寄存器 1)	
17.2.2.3 LCDCR2 (控制寄存器 2)	
17.2.2.4 LCDBUF00(缓冲寄存器 00)	
17.2.2.5 LCDBUF01(缓冲寄存器 01)	
17.2.2.6 LCDBUF02(缓冲寄存器 02)	
17.2.2.7 LCDBUF03(缓冲寄存器 03)	
17.2.2.8 LCDBUF04(缓冲寄存器 04)	
17.2.2.9 LCDBUF05(缓冲寄存器 05)	
17.2.2.10 LCDBUF06(缓冲寄存器 06)	
17.2.2.11 LCDBUF07(缓冲寄存器 07)	
17.2.2.12 LCDBUF08(缓冲寄存器 08)	
17.2.2.13 LCDBUF09(缓冲寄存器 09)	
17.2.2.14 LCDBUF10(缓冲寄存器 10)	
17.2.2.15 LCDBUF11(缓冲寄存器 11)	
17.2.2.16 LCDBUF12(缓冲寄存器 12)	
17.2.2.17 LCDBUF13(缓冲寄存器 13)	
17.2.2.18 LCDBUF14(缓冲寄存器 14)	
17.2.2.19 LCDBUF15(缓冲寄存器 15)	
17.2.2.20 LCDBUF16(缓冲寄存器 16)	
17.2.2.21 LCDBUF17(缓冲寄存器 17)	
17.2.2.22 LCDBUF18(缓冲寄存器 18)	
17.2.2.23 LCDBUF19(缓冲寄存器 19)	
17.3 功能	314
17.3.1 LCD 显示控制	314
17.3.2 复位时的工作	314
17.3.3 SLEEP/STOP 模式下的工作	316
17.3.4 SLOW 模式下的工作	316
17.3.5 故障安全	316
17.3.6 LCD 驱动方法(LCDCR1<DUTY>)	317
17.3.7 帧频(LCDCR1<SLF>)	318
17.3.8 内部/外部泄放电阻切换控制	319
17.3.9 低内部泄放电阻连接时间选择(LCDCR2<LRSE>)	319
17.3.10 高内部泄放电阻选择(LCDCR2<BRH>)	321
17.3.11 LCD 显示工作	321
17.3.12 显示数据设置	322
17.4 LCD 驱动器控制方式示例	324
17.4.1 初始化	324
17.4.2 显示数据设置	324
17.4.3 驱动输出示例	328

18. 低电压检测电路(LVD)	
18.1 结构	333
18.2 寄存器	334
18.2.1 寄存器列表	334
18.2.2 LVDICR (LVD-INT 控制寄存器)	334
18.2.3 LVDSR (状态寄存器)	335
18.3 工作说明	336
18.3.1 检测电压选择和启用/禁用工作	336
18.3.2 低电压检测	336
19. 看门狗计时器(WDT)	
19.1 配置	337
19.2 寄存器	338
19.2.1 寄存器列表	338
19.2.2 WDMOD (看门狗计时器模式寄存器)	338
19.2.3 WDCR(看门狗计时器控制寄存器)	339
19.3 工作描述	340
19.3.1 基本运行	340
19.3.2 运行模式和状态	340
19.3.3 在检测到故障(失控)时的运行情况	340
19.3.3.1 INTWDT 中断生成	
19.3.3.2 内部复位生成	
19.4 看门狗计时器的控制	341
19.4.1 禁用控制	341
19.4.2 启用控制	341
19.4.3 看门狗计时器清除控制	341
19.4.4 看门狗计时器的检测时间	341
20.闪存工作	
20.1 特征	343
20.1.1 存储器容量和配置	343
20.1.2 功能	344
20.1.3 运行模式	344
20.1.3.1 模式说明	
20.1.3.2 模式确定	
20.1.4 存储器地址	346
20.1.5 保护/安全功能	346
20.1.5.1 保护功能	
20.1.5.2 安全功能	
20.1.6 寄存器	349
20.1.6.1 寄存器列表	
20.1.6.2 FCFLCS (闪存控制寄存器)	
20.1.6.3 FCSECBIT (安全位寄存器)	
20.2 闪存的详细描述	351
20.2.1 功能	351
20.2.2 闪存的工作模式	351
20.2.3 如何执行命令	351
20.2.4 命令描述	353
20.2.4.1 自动页面程序	
20.2.4.2 自动芯片擦除	
20.2.4.3 自动存储块擦除	
20.2.4.4 自动保护位程序	
20.2.4.5 自动保护位擦除	
20.2.4.6 ID-Read	
20.2.4.7 读取命令和读取/复位命令(软件复位)	
20.2.5 命令顺序	357

20.2.5.1 命令顺序列表	
20.2.5.2 总线周期内的地址位配置	
20.2.5.3 存储块地址(BA)	
20.2.5.4 如何指定保护位(PBA)	
20.2.5.5 ID-读取代码(IA, ID)	
20.2.5.6 命令顺序示例	
20.2.6 流程图	361
20.2.6.1 自动程序	
20.2.6.2 自动擦除	
20.3 如何利用单启动模式对闪存进行重新编程	363
20.3.1 模式设置	363
20.3.2 接口规格	363
20.3.3 对内部存储器的限制	365
20.3.4 工作命令	365
20.3.4.1 RAM 传送	
20.3.4.2 闪存芯片擦除和保护位擦除	
20.3.5 共用工作不考虑命令	367
20.3.5.1 串行工作模式确定	
20.3.5.2 应答响应数据	
20.3.5.3 密码确定	
20.3.5.4 CHECK SUM 计算	
20.3.6 RAM 传送时的传送格式	375
20.3.7 闪存芯片擦除和保护位擦除的转换格式	377
20.3.8 启动程序完整流程图	379
20.3.9 芯片上 BOOT ROM 中使用重编程算法重编闪存程序	380
20.3.9.1 步骤-1	
20.3.9.2 步骤-2	
20.3.9.3 步骤-3	
20.3.9.4 步骤-4	
20.3.9.5 步骤-5	
20.3.9.6 步骤-6	
20.4 用户启动模式下编程	383
20.4.1 (1-A)编程程序存储于闪存的步骤	383
20.4.1.1 步骤-1	
20.4.1.2 步骤-2	
20.4.1.3 步骤-3	
20.4.1.4 步骤-4	
20.4.1.5 步骤-5	
20.4.1.6 步骤-6	
20.4.2 (1-B)编程程序自外部主机进行传送的步骤	387
20.4.2.1 步骤-1	
20.4.2.2 步骤-2	
20.4.2.3 步骤-3	
20.4.2.4 步骤-4	
20.4.2.5 步骤-5	
20.4.2.6 步骤-6	

21. 调试接口

21.1 规范概述	391
21.2 引脚功能	391
21.3 复位向量中止	391
21.4 调试接口引脚用于通用用途时的注意事项	391
21.5 调试启用引脚	391
21.6 在停止模式时的外设功能	393
21.7 关于与调试工具的连接	393

22. 电气特性

22.1 绝对最大额定值	394
22.2 DC 电气特性(1/3)	395
22.3 DC 电气特性(2/3)	396

22.4 DC 电气特性(3/3)	396
22.5 10-位 ADC 电气特性	397
22.6 24-位 $\Delta\Sigma$ADC 电气特性	397
22.7 温度传感器特性	399
22.8 LCD 特性	399
22.9 AC 电气特性	400
22.9.1 AC 测量条件	400
22.9.2 串行通道(SIO/UART)	400
22.9.2.1 I/O 接口模式	
22.9.3 串行总线接口(I2C/SIO).....	402
22.9.3.1 I2C 模式	
22.9.3.2 时钟同步 8-位 SIO 模式	
22.9.4 事件计数器.....	404
22.9.5 捕捉	404
22.9.6 外部中断.....	404
22.9.7 SCOUT 引脚 AC 特性	405
22.9.8 调试通信.....	405
22.9.9 闪存特性.....	406
22.9.10 芯片上振荡器	406
22.10 建议采用的振荡电路	407
22.10.1 陶瓷振荡器.....	407
22.10.2 水晶振荡器.....	407

23. 端口部分等效电路示意图

23.1 PORT 引脚	409
23.2 模拟引脚	410
23.3 控制引脚	410
23.4 时钟引脚	411

24. 封装尺寸

CMOS 32-位微控制器

TMPM061FWFG

TMPM061FWFG为 32-位RISC微处理器系列，带有一个ARM Cortex™-M0 微处理器内核。
TMPM061FWFG的特征如下：

1.1 特征

1. ARM Cortex-M0 微处理器内核

a. 通过采用Thumb®-2 指令，实现了编码效率的改善。

b. 同时实现了高性能与低功耗。

[高性能]

- 用一个时钟就能执行 32-位乘法($32 \times 32 = 32$ 位)

[低功耗]

- 用低功耗库优化了设计
- 待机功能适时停止微处理器核心的运行

c. 高速中断响应有利于实时控制

- 可中断的长指令
- 硬件自动处理压栈

2. 字节序：低位优先

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：8 K字节

4. 功率计算引擎(PCE)：功率计算用DSP功能

5. 16-位计时器 (TMRB)：2 通道

- 16-位间隔计时器模式
- 16-位计数器模式
- 16-位PPG输出
- 输入捕获功能
- 同步模式

6. 16-位计时器(TMR16A)：7 通道

7. 通用串行接口(SIO/UART)：4 通道

可选择UART模式或同步模式

- * 可实现端口选择(1 通道)
- * 可实现IR载波脉冲输出(3 通道)

8. 串行总线接口(I2C/SIO)：1 通道

可以选择I2C总线模式或同步模式。

9. 10-位AD转换器(ADC): 1 单元
- 固定通道/扫描模式
 - 单/重复模式
 - AD监控 2ch
 - 转换速度
 - 16.2 μ sec (AVDD = 2.7 ~ 3.6 V)
 - 32.4 μ sec (AVDD = 1.8 ~ 3.6 V)
10. 24-位 $\Delta\Sigma$ AD转换器(ADC): 3 单元
- 样品频率: 3 KHz, 6 KHz
 - 输入电压范围:-0.375 ~ +0.375 V
 - 可编程增益放大器: 可选择 $\times 1$, $\times 2$, $\times 4$, $\times 8$ 或 $\times 16$ 。
 - 转换模式: 单或重复
 - 转换启动的调节
 - 多单元的不同步启动
11. 温度传感器(TEMP)
12. 实时时钟(RTC): 1 通道
- 时钟(时, 分, 秒)
 - 日历(月, 周, 日期与闰年)
 - 时钟调节(通过软件)
13. LCD驱动/控制器(LCDD)
- 可实现LCD直接驱动(40 seg \times 4 com)
 - 可选择1/4, 1/3, 1/2 占空比或静态驱动
 - 配备有泄放电阻(也可使用外部泄放电阻)
14. 电压检测电路(LVD)
15. 看门狗计时器(WDT): 1 通道
- 看门狗计时器(WDT)可生成复位或非屏蔽中断(NMI)。
16. 中断源: 可将优先级顺序设置为 4 种水平。
- 内部: 28 个因数
 - 外部: 4 个因数
17. 输入/输出端口(PORT): 64 引脚
- 备好三个可耐受 5 V的输入端
18. 待机模式
- 待机模式: IDLE, SLOW, SLEEP, STOP
19. 时钟发生器(CG)

时钟齿轮功能：高速时钟分为 1/1, 1/2, 1/4, 1/8 或 1/16

20. 最高运行频率：16 MHz

21. 工作电压：1.8~3.6 V

2.9 ~ 3.6 V：可使用 24-位 $\Delta\Sigma$ AD转换器

- 2.2 ~ 3.6 V: 可使用LCD驱动器/控制器
- 2.7 ~ 3.6 V: 可实现闪存ROM的写入/擦除

22. 温度范围

- -40 ~ 85 度(闪存写入/擦除期间除外)
- 0 ~ 70 度(闪存写入/擦除期间)

23. 封装

LQFP100 (14mm × 14mm, 0.5mm 间距)

1.2 方块图

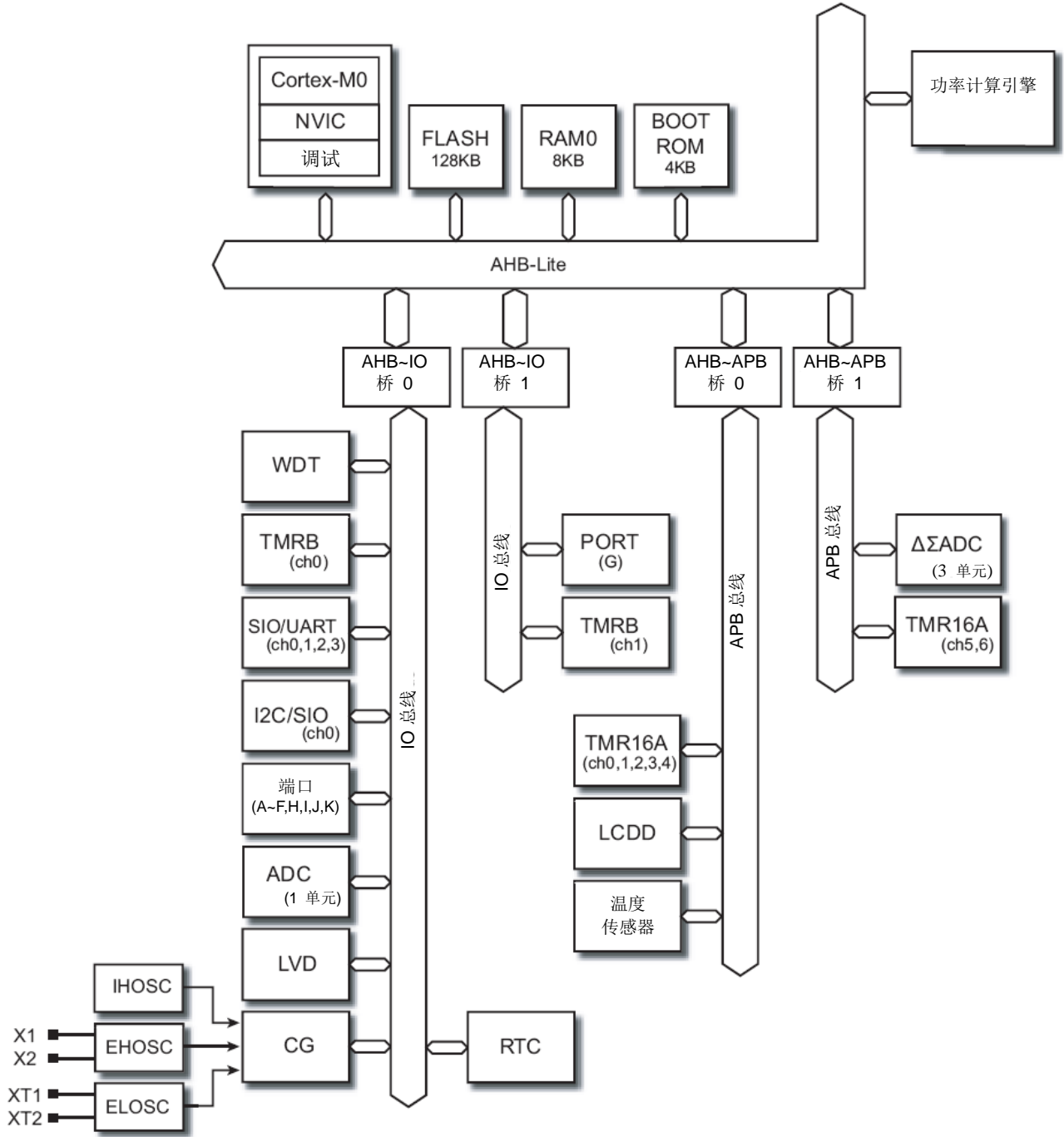


图 1-1 方块图

1.3 引脚布置图(顶视图)

图 1-2 给出了TMPM061FWFG的引脚布置图。

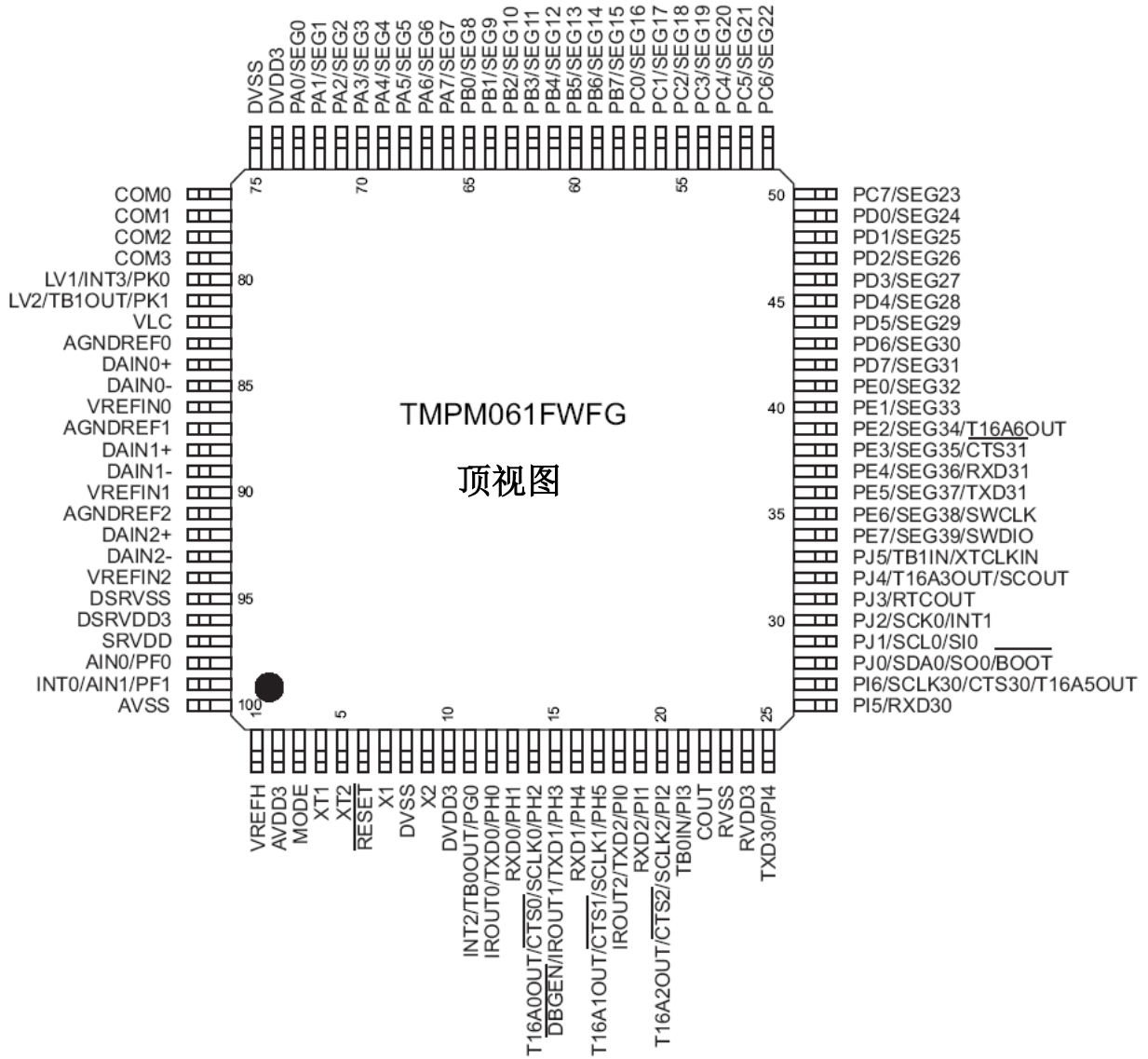


图 1-2 引脚布置图

1.4 引脚名称与功能

TMPM061FWFG的输入/输出引脚的名称与功能给出如下：

表 1-1 按引脚(1/6)分类的引脚名称与功能

类型	引脚编号	引脚名称	输入/ 输出	功能
PS	1	VREFH	-	为 10 位 AD 转换器提供参考电源。(注 1)
PS	2	AVDD3	-	为 10 位 AD 转换器提供电源。(注 1)(注 3)
控制	3	MODE	I	MODE 引脚 (注) 必须将 MODE 引脚连接至 GND。
时钟	4	XT1	I	连接到低速振荡器。
时钟	5	XT2	O	连接到低速振荡器。
控制	6	$\overline{\text{RESET}}$	I	复位输入引脚 (注) 带有一个上拉和一个噪声滤波器
时钟	7	X1	I	已连接至高速振荡器 高速时钟输入引脚。
PS	8	DVSS	-	GND 引脚
时钟	9	X2	O	已连接至高速振荡器
PS	10	DVDD3	-	电源引脚(注 3)
功能	11	PG0 TB0OUT INT2	I/O O I	I/O 端口 计时器 B 输出 外部中断引脚
功能	12	PH0 TXD0 IROUT0	I/O O O	I/O 端口 传输串行数据 用载波脉冲传输串行数据
功能	13	PH1 RXD0	I/O I	I/O 端口 接收串行数据
功能	14	PH2 SCLK0 $\overline{\text{CTS0}}$ T16A0OUT	I/O I/O I O	I/O 端口 串行时钟输入/输出 握手输入引脚 计时器 16A 输出
功能	15	PH3 TXD1 IROUT1 $\overline{\text{DBGEN}}$	I/O O O I	I/O 端口 发送串行数据 用载波脉冲发送串行数据 调试启用 (注) 通过在 $\overline{\text{RESET}}$ 信号上升时样品“低”，该引脚被固定为调试功能。
功能	16	PH4 RXD1	I/O I/O	I/O 端口 接收串行数据
功能	17	PH5 SCLK1 $\overline{\text{CTS1}}$ T16A1OUT	I/O I/O I O	I/O 端口 串行时钟输入/输出 握手输入引脚 计时器 16A 输出

表 1-1 按引脚(2/6)分类的引脚名称与功能

类型	引脚编号	引脚名称	输入/ 输出	功能
功能	18	PI0 TXD2 IROUT2	I/O O O	I/O 端口 发送串行数据 用载波脉冲发送串行数据
功能	19	PI1 RXD2	I/O I	I/O 端口 接收串行数据
功能	20	PI2 SCLK2 CTS2 T16A2OUT	I/O I/O I O	I/O 端口 串行时钟输入/输出 握手输入引脚 计时器 16A 输出
功能	21	PI3 TB0IN	I/O I	I/O 端口 输入计时器 B 捕获触发
PS	22	COUT	-	稳压器输出引脚 将 1 μ F 电容器连接到 COUT 和 RVSS 之间。
PS	23	RVSS	-	GND 引脚
PS	24	RVDD3	-	电源引脚(注 3)
功能	25	PI4 TXD30	I/O O	I/O 端口 发送串行数据
功能	26	PI5 RXD30	I/O I	I/O 端口(可耐受 5 V 的输入端)(注 7) 接收串行数据
功能	27	PI6 SCLK30 CTS30 T16A5OUT	I/O I/O I O	I/O 端口(可耐受 5 V 的输入端)(注 7) 串行时钟输入/输出 握手输入引脚 计时器 16A 输出
功能	28	PJ0 SDA0 SIO BOOT	I/O I/O I/O I	I/O 端口 I2C 模式: 数据引脚 SIO 模式: 数据引脚 设置单启动模式: (注) 通过在 RESET 信号上升时样品“低”, TMPM061FWFG 可进入单启动模式。
功能	29	PJ1 SCL0 SIO	I/O I/O I/O	I/O 端口 I2C 模式: 时钟引脚 SIO 模式: 数据引脚
功能	30	PJ2 SCK0 INT1	I/O I/O I	I/O 端口(可耐受 5 V 的输入端)(注 7) 如果串行总线接口以 SIO 模式运行, 则输入与输出时钟。 外部中断引脚
功能	31	PJ3 RTCOUT	I/O O	I/O 端口 RTC 输出
功能	32	PJ4 T16A3OUT SCOUT	I/O O O	I/O 端口 计时器 16A 输出 系统时钟输出
功能	33	PJ5 TB1IN XTCLKIN	I I I	输入端口 输入计时器 B 捕获触发 低速时钟输入引脚

表 1-1 按引脚(3/6)分类的引脚名称与功能

类型	引脚编号	引脚名称	输入/ 输出	功能
功能/ 调试	34	PE7 SEG39 SWDIO	I/O O I/O	I/O端口 LCD节段输出引脚 调试引脚
功能/ 调试	35	PE6 SEG38 SWCLK	I/O O I	I/O端口 LCD节段输出引脚 调试引脚
功能	36	PE5 SEG37 TXD31	I/O O O	I/O端口 LCD节段输出引脚 发送串行数据
功能	37	PE4 SEG36 RXD31	I/O O I	I/O端口 LCD节段输出引脚 接收串行数据
功能	38	PE3 SEG35 SCLK31 CTS31	I/O O I/O I	I/O端口 LCD节段输出引脚 握手输入引脚 串行时钟输入/输出
功能	39	PE2 SEG34 T16A6OUT	I/O O O	I/O端口 LCD节段输出引脚 计时器 16A输出
功能	40	PE1 SEG33	I/O O	I/O端口 LCD节段输出引脚
功能	41	PE0 SEG32	I/O O	I/O端口 LCD节段输出引脚
功能	42	PD7 SEG31	I/O O	I/O端口 LCD节段输出引脚
功能	43	PD6 SEG30	I/O O	I/O端口 LCD节段输出引脚
功能	44	PD5 SEG29	I/O O	I/O端口 LCD节段输出引脚
功能	45	PD4 SEG28	I/O O	I/O端口 LCD节段输出引脚
功能	46	PD3 SEG27	I/O O	I/O端口 LCD节段输出引脚
功能	47	PD2 SEG26	I/O O	I/O端口 LCD节段输出引脚
功能	48	PD1 SEG25	I/O O	I/O端口 LCD节段输出引脚
功能	49	PD0 SEG24	I/O O	I/O端口 LCD节段输出引脚
功能	50	PC7 SEG23	I/O O	I/O端口 LCD节段输出引脚
功能	51	PC6 SEG22	I/O O	I/O端口 LCD节段输出引脚

表 1-1 按引脚(4/6)分类的引脚名称与功能

类型	引脚编号	引脚名称	输入/ 输出	功能
功能	52	PC5 SEG21	I/O O	I/O端口 LCD节段输出引脚
功能	53	PC4 SEG20	I/O O	I/O端口 LCD节段输出引脚
功能	54	PC3 SEG19	I/O O	I/O端口 LCD节段输出引脚
功能	55	PC2 SEG18	I/O O	I/O端口 LCD节段输出引脚
功能	56	PC1 SEG17	I/O O	I/O端口 LCD节段输出引脚
功能	57	PC0 SEG16	I/O O	I/O端口 LCD节段输出引脚
功能	58	PB7 SEG15	I/O O	I/O端口 LCD节段输出引脚
功能	59	PB6 SEG14	I/O O	I/O端口 LCD节段输出引脚
功能	60	PB5 SEG13	I/O O	I/O端口 LCD节段输出引脚
功能	61	PB4 SEG12	I/O O	I/O端口 LCD节段输出引脚
功能	62	PB3 SEG11	I/O O	I/O端口 LCD节段输出引脚
功能	63	PB2 SEG10	I/O O	I/O端口 LCD节段输出引脚
功能	64	PB1 SEG9	I/O O	I/O端口 LCD节段输出引脚
功能	65	PB0 SEG8	I/O O	I/O端口 LCD节段输出引脚
功能	66	PA7 SEG7	I/O O	I/O端口 LCD节段输出引脚
功能	67	PA6 SEG6	I/O O	I/O端口 LCD节段输出引脚
功能	68	PA5 SEG5	I/O O	I/O端口 LCD节段输出引脚
功能	69	PA4 SEG4	I/O O	I/O端口 LCD节段输出引脚
功能	70	PA3 SEG3	I/O O	I/O端口 LCD节段输出引脚
功能	71	PA2 SEG2	I/O O	I/O端口 LCD节段输出引脚
功能	72	PA1 SEG1	I/O O	I/O端口 LCD节段输出引脚

表 1-1 按引脚(5/6)分类的引脚名称与功能

类型	引脚编号	引脚名称	输入/ 输出	功能
功能	73	PA0 SEG0	I/O O	I/O端口 LCD节段输出引脚
PS	74	DVDD3	-	电源引脚(注 3)
PS	75	DVSS	-	GND引脚
功能	76	COM0	O	LCD共用输出引脚
功能	77	COM1	O	LCD共用输出引脚
功能	78	COM2	O	LCD共用输出引脚
功能	79	COM3	O	LCD共用输出引脚
功能	80	PK0 INT3 LV1	I/O I -	I/O端口 外部中断引脚 连接至泄放电阻。
功能	81	PK1 TB1OUT LV2	I O -	I/O端口 计时器B 输出 连接至泄放电阻。
PS	82	VLC	-	LCD电源引脚(注 3)
PS	83	AGNDREF0	-	24 位 $\Delta\Sigma$ AD转换器:GND引脚(注 4)(注 5)
功能	84	DAIN0+	I	24 位 $\Delta\Sigma$ AD转换器模拟输入
功能	85	DAIN0-	I	24 位 $\Delta\Sigma$ AD转换器模拟输入
PS	86	VREFIN0	-	为 24 位 $\Delta\Sigma$ AD转换器提供电源。(注 4)(注 5)
PS	87	AGNDREF1	-	24 位 $\Delta\Sigma$ AD转换器:GND引脚 (注) 即使未使用 $\Delta\Sigma$ AD转换器, 也必须将AGNDREF1 连接到GND。
功能	88	DAIN1+	I	24 位 $\Delta\Sigma$ AD转换器模拟输入
功能	89	DAIN1-	I	24 位 $\Delta\Sigma$ AD转换器模拟输入
PS	90	VREFIN1	-	为 24 位 $\Delta\Sigma$ AD转换器提供电源。(注 4)(注 5)
PS	91	AGNDREF2	-	24 位 $\Delta\Sigma$ AD转换器: GND引脚(注 4)(注 5)
功能	92	DAIN2+	I	24 位 $\Delta\Sigma$ AD转换器模拟输入
功能	93	DAIN2-	I	24 位 $\Delta\Sigma$ AD转换器模拟输入
PS	94	VREFIN2	-	为 24 位 $\Delta\Sigma$ AD转换器提供电源。(注 4)(注 5)
PS	95	DSRVSS	-	电压参考电路: GND引脚(注 6)
PS	96	DSRVDD3	-	为 $\Delta\Sigma$ AD转换器的放大器电路提供电源。(注 6)

表 1-1 按引脚(6/6)分类的引脚名称与功能

类型	引脚编号	引脚名称	输入/ 输出	功能
PS	97	SRVDD	-	为电压参考电路提供电源。(注 6)
功能	98	PF0 AIN0	I/O I	I/O端口 10 位AD转换器模拟输入
功能	99	PF1 AIN1 INT0	I/O I I	I/O端口 10 位AD转换器模拟输入 外部中断引脚
PS	100	AVSS	-	10 位AD转换器：GND引脚(注 2)

注 1: 即使未使用 10 位AD转换器, 也必须将AVDD3 和VREH连接到电源。

注 2: 即使未使用 10 位AD转换器, 也必须将AVSS连接到GND。

注 3: 为DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD, VLC提供的电压必须相同。

注 4: 在使用24位 $\Delta\Sigma$ AD转换器时, 应提供如下引脚处理:

- 请勿将VREFINx连接到参考电压。
- 将AGNDREFx连接到DVSS电平。
- 将 1 μ F 电容器连接到VREFINx和AGNDREFx之间。

注 5: 在未使用 24 位 $\Delta\Sigma$ AD转换器时, 必须采用以下设置。

- 将AGNDREFx连接到DVSS电平。

注 6: 在也未使用温度传感器时, 参考电压电路必须采用以下设置。

- 将DSRVDD3 和SRVDD3连接到DVDD3。
- 将DSRVSS3连接到DVSS。

注 7: 只有在输入启用时, 这些引脚才能承受 5 V输入。

注意当用作开漏输出时, 这些引脚不能上拉到电源电压之上。

1.5 引脚编号和电源引脚

表 1-2 引脚编号和电源引脚

电源	电压范围	引脚编号	引脚名称
AVDD3	1.8 ~ 3.6 V	2	1~2 引脚, 98~100 引脚
DVDD3		8, 74	3~82 引脚
SRVDD		97	95~97 引脚
VREFIN0	-	86	83~86 引脚
VREFIN1		90	87~90 引脚
VREFIN2		94	91~94 引脚

2. 产品信息

本章描述了有关外部设备的产品特定信息。在使用本章时，应参考外部设备一章所述的相关内容。

2.1 16-位计时器/事件计数器(TMRB)

TMPM061FWFG包含2条TMRB通道。以下是该产品的特定功能：

1. 计时器触发器输出可用作捕获触发器。
而 16-位计时器A(TMR16A)的输出可被用作TMRB的捕获触发器。
 - T16A3OUT → TMRB0
 - T16A6OUT → TMRB1
2. 在计时器同步模式下启动触发器
可实现多通道的同步启动。
 - TMRB0 → 同时启动TMRB0和TMRB1。

表 2-1 描述了 TMRB 的不同点。

表 2-1 TMRB的不同点(按通道)(n: 通道编号)

通道	引脚(端口/引脚编号)		计时器之间的触发器功能	
	外部时钟/ 捕获触发器 输入引脚 TBnIN	计时器触发器 输出引脚 TBnOUT	捕获触发器	同步启动 触发器通道
TMRB0	PI3 (21)	PG0 (11)	T16A3OUT	-
TMRB1	PJ5 (33)	PK1 (81)	T16A6OUT	TMRB0

2.2 16-位计时器 A (TMR16A)

TMPM061FWFG包含 7 条TMR16A通道。通道 4 不向该矩形波引脚提供匹配中断信号与输出信号。

表 2-2 给出了TMR16A的矩形波输出引脚列表。

表2-2 TMR16A所使用的引脚(n: 通道编号)

通道	T16AnOUT引脚
TMR16A0	PH2 (14)
TMR16A1	PH5 (17)
TMR16A2	PI2 (20)
TMR16A3	PJ4 (32)
TMR16A4	-
TMR16A5	PI6 (27)
TMR16A6	PE2 (39)

2.3 串行通道(SIO/UART)

TMPM061FWFG包含4条SIO通道。以下是该产品的特定功能:

1. 计时器输出被用作传送时钟

在UART模式下, 计时器输出可被用作传送时钟。

- T16A0OUT → SIO0, SIO1
- T16A1OUT → SIO2, SIO3

2. 计时器输出被用作载波脉冲

计时器触发器输出可被用于传输(利用载波脉冲)。但是, SIO3不具备该功能。

- T16A4OUT → SIO0, SIO1, SIO2

表 2-3 描述了SIO的不同点。

SIO3可选择拟使用的端口。所有端口均必须有效。SIO3 不具备IROUTn的功能(载波脉冲传输带有数据)。

表 2-3 按照通道 SIO 的不同点 (n: 通道编号)

通道	引脚 (端口/引脚编号)			计时器输出	
	TXDn IROUTn	RXDn	SCLKn CTS _n	用作时钟	用作载波 脉冲
SIO0	PH0 (12)	PH1 (13)	PH2 (14)	T16A0OUT	T16A4OUT
SIO1	PH3 (15)	PH4 (16)	PH5 (17)		
SIO2	PI0 (18)	PI1 (19)	PI2 (20)	T16A1OUT	
SIO3	PI4 (25)	PI5 (26)	PI6 (27)		
	PE5 (36)	PE4 (37)	PE3 (38)	-	

2.4 模拟/数字转换器(ADC)

2.4.1 非可用功能

在 TMPM061FWFG 中，不能使用以下 ADC 功能。请勿设置相关的寄存器。

功能	寄存器
最高优先级转换	ADMOD2, ADREGSP
AD 监视功能	ADMOD3, ADMOD5, ADCMP0, ADCMP1
AD 启动通过硬件	ADMOD4 <ADHTG> <ADHS> <HADHTG> <HADHS>

2.4.2 转换通道

在 TMPM061FWFG 中，从 0 到 3 的 4 条通道被用作 AD 转换器的输入通道。被输入到各通道的模拟信号如下所列。

表 2-4 ADC输入

通道	输入
通道 0	AIN0引脚(PF0/98引脚)
通道 1	AIN1引脚(PF1/99引脚)
通道 2	温度传感器输出
通道 3	$\Delta\Sigma$ ADC参考电压电路(BGR)输出

可用 ADMOD0<SCAN>, ADMOD1<ADSCN>和<ADCH>指定各转换通道。表 2-5 描述了可用的设置。

表 2-5 标准转换时的通道选择

		ADMOD1<ADCH[3:0]>				0100~1111
		0000	0001	0010	0011	
ADMOD0 <SCAN>=0	固定通道	AIN0	AIN1	AIN2	AIN3	不可用
ADMOD0 <SCAN>=1	ADMOD1<ADSCN>=00 4-通道扫描	AIN0	AIN0~AIN1	AIN0~AIN2	AIN0~AIN3	
	ADMOD1<ADSCN>=01 8-通道扫描	AIN0	AIN0~AIN1	AIN0~AIN2	AIN0~AIN3	
	ADMOD1<ADSCN>=10 12-通道扫描	AIN0	AIN0~AIN1	AIN0~AIN2	AIN0~AIN3	

2.5 $\Delta\Sigma$ 模拟/数字转换器(DSADC)

TMPM061FWFG包含3个DSADC单元。

下表描述了DSADC的同步启动功能中，主设备单元和从属设备单元的分配情况。

表 2-6 主设备/从设备分配

主机	从机
单元A	单元B 单元C

3. 处理器内核

TMPM061FWFG系列配备有一个高性能的 32-位处理器内核(ARM Cortex-M0 处理器内核)。有关该处理器内核运行方面的信息，请参照ARM有限公司所发布的“Cortex-M0 技术参考手册”。本章给出了专属于TMPM061FWFG系列的功能不拟在本文件中阐述这些功能。

3.1 处理器内核该相关信息

下表给出了TMPM061FWFG中处理器内核的修订说明。

请参照该CPU内核与CPU体系结构的详细信息，并参照以下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
TMPM061FWFG	r0p0-03

3.2 可配置选项

Cortex-M0 内核带有可选存储块。下表描述了以下对象的可配置选项

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

注：该快速乘法器可提供一个 32-位 × 32-位逻辑乘可得出最低有效 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

可在 Cortex-M0 内核中任意定义该中断输入数目。

TMPM061FWFG 有 32 个中断输入。

3.3.2 SysTick

TMPM061FWFG 的 SysTick 计时器可生成 SysTick 异常。

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

3.3.3 SYSRESETREQ

在设置了应用中断和复位控制寄存器的<SYSRESETREQ>位之后，该 Cortex-M0 内核可输出 SYSRESETREQ 信号。

在 SYSRESETREQ 信号输出时，TMPM061FWFG 可提供同样的工作。

注：在 SLOW 模式下，无法通过<SYSRESETREQ>实现复位工作。

3.3.4 LOCKUP

在无法修复的异常生成的情况下，Cortex-M0 内核会输出 LOCKUP 信号，表示软件中出现了严重错误。

TMPM061FWFG 不使用该信号。如拟从 LOCKUP 状态返回，则需使用非屏蔽中断(NMI)或复位。

3.4 事件

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

TMPM061FWFG 不使用事件输出信号和事件输入信号。请勿使用 SEV 指令与 WFE 指令。

3.5 电源管理

Cortex-M0 内核具备需使用 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

-中断服务例行程序(ISR)在系统控制寄存器的<SLEEPONEXIT>位被设置的情况下退出时的计时。

TMPM061FWFG不使用SLEEPDEEP信号，因此不必设置<SLEEPDEEP>位。也不使用事件信号，因此请勿使用WFE指令。

有关电源管理的详细，请参照"时钟/模式控制"一节。

4. 存储器地址

4.1 存储器地址

TMPM061FWFG的存储器地址基于ARM Cortex-M0 处理器内核存储器地址。

内部ROM被地址到Cortex-M0 内核存储器的代码，内部RAM被地址到SRAM区域，而特殊功能寄存器(SFR)则被地址到外设区域。

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。TMPM061FWFG具备与Cortex-M3 等效的位-带功能，且TMPM061FWFG的SRAM和SFR区域均被包含在该位带区内。

CPU寄存器区域是处理器内核的内部寄存器区域。

有关各区域的详细，见“Cortex-M0 技术参考手册”。

注意对已被标注为“故障”的区域进行访问，可导致发生硬故障。请勿访问供应商特定区域和保留区域。

图 4-1 给出了TMPM061FWFG的存储器地址。

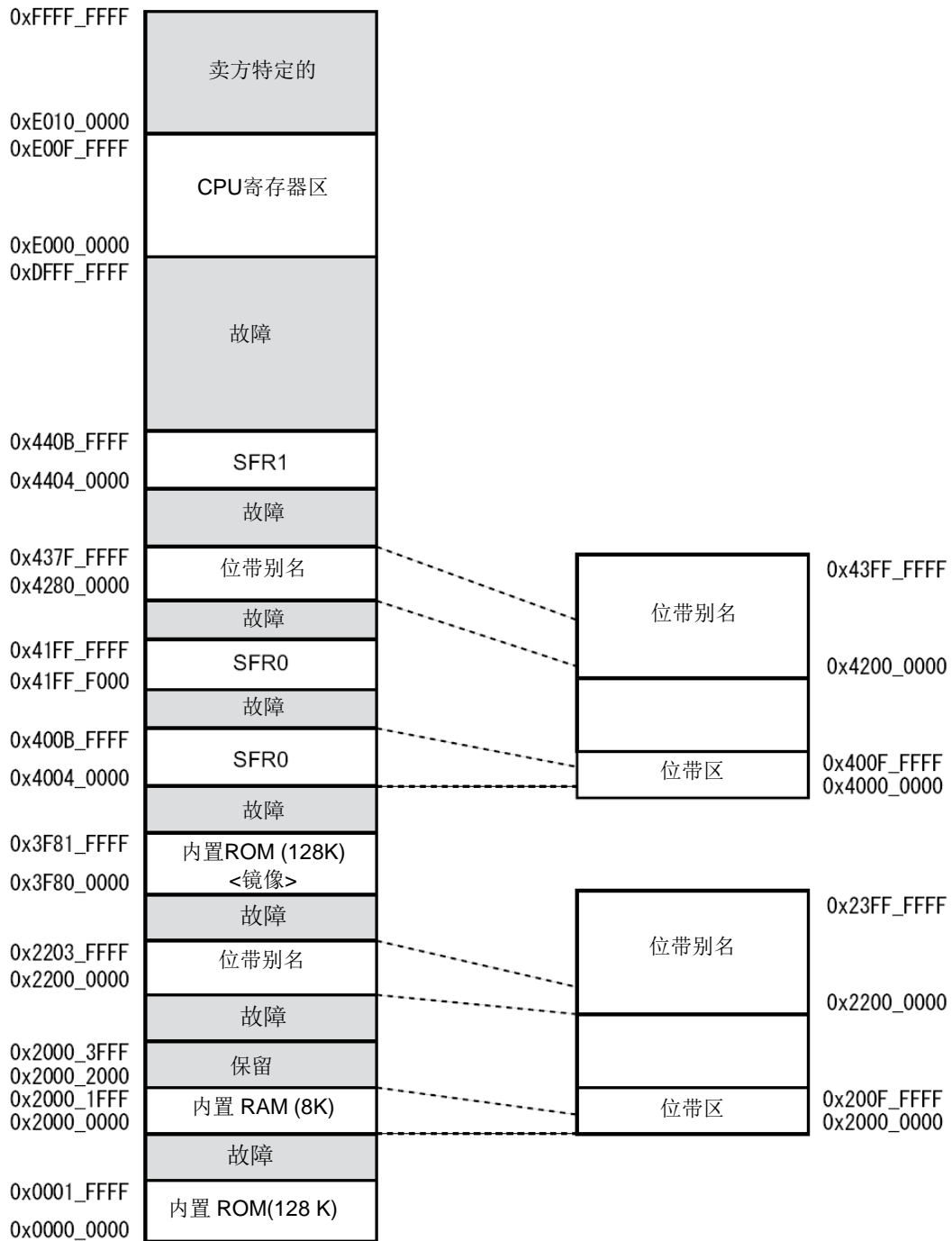


图 4-1 存储器地址

4.2 总线结构

TMPM061FWFG包含Cortex-M0 内核和用作总线主设备的PCE。

从设备则是内置ROM, 内置RAM0/1, 内置BOOT ROM, 桥0/1从AHB-Lite总线到APB总线和桥0/1 从AHB-Lite总线到IO总线。0 号桥被分配到SFR0 区, 而 1 号桥则被分配到SFR1 区。各外设功能均需通过桥与APB总线或IO总线相连。

只要该访问的对象不是同一从设备, 就可执行从内核或PCE到从设备的访问。

注: PCE无法访问内置BOOT ROM。

图4-2给出了总线结构。

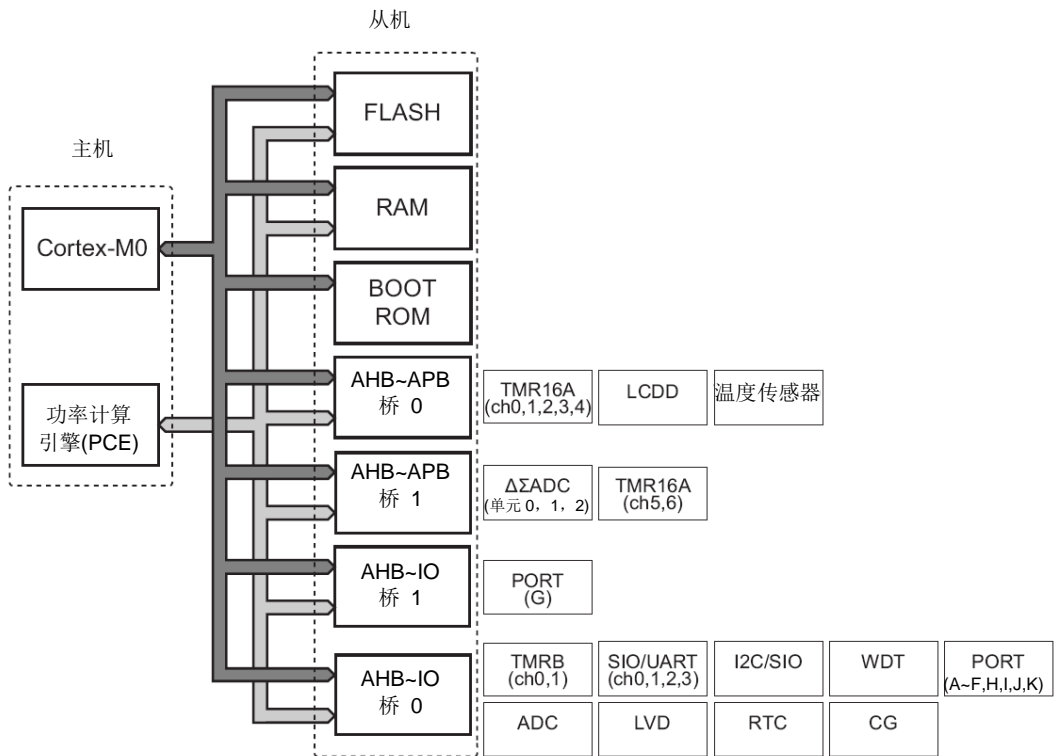


图 4-2 总线结构

4.3 外设功能的地址列表

外设功能的参考地址如以下所列。注意观察该外设功能被分配到了SFR0 或SFR1 的哪个区。

请勿访问SFR区中的各地址各控制寄存器除外。有关各控制寄存器的详细，请参照各外设功能有关章节。

外设功能		基址	区域
时钟/模式控制(CG)		0x400F_3000	SFR0
输入/输出端口	PORTA	0x400C_0000	SFR0
	PORTB	0x400C_0100	SFR0
	PORTC	0x400C_0200	SFR0
	PORTD	0x400C_0300	SFR0
	PORTE	0x400C_0400	SFR0
	PORTF	0x400C_0500	SFR0
	PORTG	0x440C_0600	SFR1
	PORTH	0x400C_0700	SFR0
	PORTI	0x400C_0800	SFR0
	PORTJ	0x400C_0900	SFR0
	PORTK	0x400C_0A00	SFR0
16-位计时器/事件计数器 (TMRB)	ch0	0x400C_4000	SFR0
	ch1	0x440C_4100	SFR1
16-位计时器 A (TMR16A)	ch0	0x4008_D000	SFR0
	ch1	0x4008_E000	SFR0
	ch2	0x4008_F000	SFR0
	ch3	0x4009_0000	SFR0
	ch4	0x4009_1000	SFR0
	ch5	0x4409_2000	SFR1
	ch6	0x4409_3000	SFR1
串行通道 (SIO/SIO)	ch0	0x400E_1000	SFR0
	ch1	0x400E_1100	SFR0
	ch2	0x400E_1200	SFR0
	ch3	0x400E_1300	SFR0
串行总线接口(I2C/SIO)		0x400E_0000	SFR0
模拟/数字转换器 (ADC)		0x400F_C000	SFR0
$\Delta\Sigma$ 模拟/数字转换器 (DSADC)	单元 0	0x4406_7000	SFR1
	单元 1	0x4406_8000	SFR1
	单元 2	0x4406_9000	SFR1
温度传感器(TEMP)		0x4005_D000	SFR0
实时时钟(RTC)		0x400C_C000	SFR0
LCD驱动器 (LCD)		0x4006_E000	SFR0
电压检测器(LVD)		0x400F_4000	SFR0
看门狗计时器(WDT)		0x400F_2000	SFR0
闪存/调试(FC)		0x41FF_F000	SFR0

5. 复位工作

复位工作源如下几种：

- RESET引脚($\overline{\text{RESET}}$)
- 看门狗计时器 (WDT)
- CPU和复位寄存器<SYSRESETREQ>位发出的信号导致的应用中断

在确认复位源时，需检查时钟发生器寄存器中的CGRSTFG见“异常”一章所述。

有关通过WDT进行复位，请参照“看门狗计时器”一章。

有关通过<SYSRESETREQ>进行复位，请参照“Cortex-M0 技术参考手册”。

注 1：复位工作完成后，内部RAM数据有可能会丢失。

注 2：在该SLOW模式下，请勿通过<SYSRESETREQ>进行复位。

5.1 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计时间。在TMPM061FWFG中，内部稳压器至少需 1 ms才能实现稳定。在冷复位时， $\overline{\text{RESET}}$ 引脚必须在足够长的时段内持续保持为“低”，以确保内部稳压器可实现稳定。约 1.6 ms之后， $\overline{\text{RESET}}$ 引脚变为“高”，内部复位即被解除。

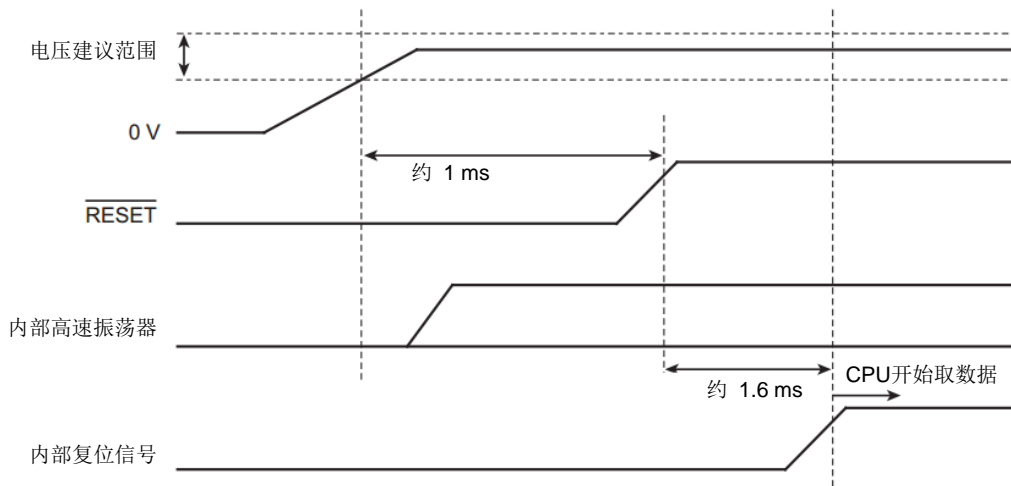


图 5-1 冷复位工作顺序

5.2 热复位

在复位TMPM061FWFG时必须具备以下条件：电源电压位于工作范围内； $\overline{\text{RESET}}$ 引脚保持为“低”至少12个内部高速时钟。约 1.6 ms之后， $\overline{\text{RESET}}$ 引脚变为“高”，内部复位即被解除。

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



5.3 复位后

内核所有控制寄存器和外设功能控制寄存器(SFR)，均可通过复位实现初始化。

在复位被解除时，TPM061FWFG可由内部高速振荡器的时钟启动运行。如有必要，应设置外部时钟。

6. 时钟/模式控制

6.1 特征

时钟/模式控制存储块能够选择时钟齿轮，预分频器时钟和振荡器的预热。

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。

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
- 控制预热计时器

除NORMAL模式外，TMPM061FWFG还可在多种低功率模式下运行，以根据其使用情况降低功耗。

6.2 寄存器

6.2.1 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“存储器地址”一章的“外部功能地址表”。

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
外部高速时钟选择寄存器	CGEHCLKSEL	0x000C
系统时钟选择寄存器	CGCKSEL	0x0010

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	-	-	-	FCSTOP	-	-	SCOSEL	
复位后	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
比特符号	-	-	FPSEL1	FPSEL0	-	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。
23	-	R/W	写入“0”。
22-21	-	R	读作 0。
20	FCSTOP	R/W	ADC时钟 0: 激活 1: 停止 利用该位，可停止向AD转换器和ΣΔAD转换器提供时钟。 在复位后，就可提供ADC时钟。 在将“1”(停止)设置到该寄存器前，确认AD转换已停止或已完成。
19-18	-	R	读作 0。
17-16	SCOSEL[1:0]	R/W	SCOUT输出 00: fs 01: fsys/2 10: fsys 11: φT0 指定拟从SCOUT引脚输出的时钟。
15-14	-	R	读作 0。
13-12	FPSEL0	R/W	fperiph 0 0: fgear 1: fc 不管时钟齿轮模式，选择fc就能固定fperiph。



位	比特符号	类型	功能
13-12	FPSEL1	R/W	fperiph 1 0: $\phi T0$ 1: fsys
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	高速时钟齿轮(fc)齿轮(注) 000: fc 100: fc/2 001: 保留 101: fc/4 010: 保留 110: fc/8 011: 保留 111: fc/16

注: 在 SysTick 计时器正使用期间, 用户无法选择 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	WUPT							
复位后	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	WUPT				WUPSEL2	HOSCEN	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	WUPTL		-	-	EHCLKEN	LOSCSEL	XTEN	XEN1
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	WUPSEL1	-	WUEF	WUEON
复位后	0	0	1	1	0	0	0	0

位	比特符号	类型	功能
31-20	WUPT[11:0]	R/W	预热计数器设置值。(注 1) 对于上 12-位计数器值的计时器, 设置 16-位计时器。
19	HWUPSEL2	R/W	预热时钟。(注 2) 0: 内部(f_{IHOSC}) 1: 外部(f_{EHOSC}) 由高速振荡器选择预热计数器。 在使用STOP/SLEEP模式时, 请在进入STOP/SLEEP模式前, 选择与<OSCSEL>-<WUPSEL2>相同的时钟源。
18	HOSCEN	R/W	外部高速振荡器 0: 未使用X1/X2(f_{EHOSC}) 1: 使用X1/X2(f_{EHOSC}) 设置“1”, 以使用外部高速振荡器(X1/X2)。

位	比特符号	类型	功能
17	OSCSEL	R/W	高速振荡器 0: 内部(f_{IHOSC}) 1: 外部(f_{EHOSC}) 该高速振荡器可在内部和外部之间更改。切换到外部振荡器后再停止该内部振荡器, 可减少功耗。
16	XEN2	R/W	内部高速振荡器运行(注 3) 0: 停止 1: 振荡
15-14	WUPTL[1:0]	R/W	预热计数器设定值(注 1) 如选择高速振荡器, 则<WUODR[1:0]>即被设置为“00”。
13-12	-	R/W	写入 "0"。
11	EHCLKEN	R/W	外部高速时钟输入启用 0: 禁用 1: 启用
10	LOSCSEL	R/W	低速时钟 0: 低速振荡器(f_{ELOSC}) 1: 低速时钟($f_{ELCLKIN}$)
9	XTEN	R/W	外部低速振荡器模式 0: 停止 1: 振荡
8	XEN1	R/W	外部高速振荡器模式 0: 停止 1: 振荡
7-4	-	R/W	写入 "0011"
3	WUPSEL1	R/W	预热时钟 0: 高速时钟 1: 低速时钟 选择预热计时器的源时钟。高速时钟后面带有<WUPSEL2>。通过CGOSCCR<LOSCSEL>选择低速时钟。
2	-	R/W	写入"0"。
1	WUEF	R	振荡器预热计时器 (WUP)的运行 0: WUP完成 1: WUP激活 监控预热计时器的状态。
0	WUEON	W	振荡器预热计时器 (WUP)的运行 0: 忽略 1: WUP 启动 启用以启动该预热计时器。 读作 0。

注 1: 有关预热设定, 请参照“6.3.4 预热功能”一节。

注 2: 不能将外部高速时钟输入($f_{EHCLKIN}$)用作预热时钟。

注 3: 在使用内部高速振荡器(IHOSC)时, 请勿将其用作必须确保高精度的系统时钟。

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	-	-	-	-	-	-	-	DRVE
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	RXTEN	RXTEN
复位后	0	0	0	0	0	0	0	1
	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。
19-17	-	R/W	在复位后写入"0"。
16	DRVE	R/W	STOP模式下的引脚状态 0: 在STOP模式下处于不激活状态 1: 在STOP模式下处于激活状态
15-10	-	R	读作 0。
9	RXTEN	R/W	低速振荡器在解除STOP模式之后。 0: 停止 1: 振荡 在低速振荡器未被用作系统时钟的情况下, 该位可在解除STOP模式之后指定低速振荡器运行。
8	RXEN	R/W	高速振荡器在解除STOP模式之后。 0: 停止 1: 振荡 如果该高速振荡器未被用作系统时钟, 则该位可在解除STOP时指定该高速振荡器的运行。通过设置OSCCR<OSCSEL>, 即可启用内部振荡或外部振荡。
7-3	-	R	读作 0。
2-0	STBY[2:0]	R/W	低功耗模式 000: 保留 001: STOP 010: SLEEP 011: IDLE 100: 保留 101: 保留 110: 保留 111: 保留

6.2.5 CGEHCLKSEL (外部高速时钟选择寄存器)

	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	-	-	-	-	-	-	-	-
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	1	1	1	0	0	1	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	EHCLKSEL
复位后	0	0	0	1	1	1	1	0

位	比特符号	类型	功能
31-16	-	R	读作 0。
15-12	-	R/W	写入"0111"。
11	-	R	读作 0。
10-1	-	R/W	写入"0100001111"。
0	EHCLKSEL	R/W	外部高速时钟 0: 使用通过CGOSCCR<OSCSSEL>选择的时钟 1: 使用外部高速时钟输入

6.2.6 CGCKSEL(系统时钟选择寄存器)

	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	-	-	-	-	-	-	-	-
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	-	-	-	-	-	-	SYSCK	SYSCKFLG
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	SYSCK	R/W	系统时钟 0: 高速 1: 低速
0	SYSCKFLG	R/W	系统时钟的状态 0: 高速 1: 低速

6.3 时钟控制

6.3.1 时钟类型

各时钟的定义如下：

f_{EHOSC}	: 来自与X1, X2 连接的振荡器的外部高速振荡器时钟输入
$f_{EHCLKIN}$: 来自X1 的外部高速时钟输入
f_{IHOSC}	: 来自内部振荡器的内部高速振荡器时钟输入
f_s	: 来自与XT1, XT2 连接的振荡器的外部低速振荡器时钟输入
$f_{ELCLKIN}$: 来自PJ5 (33引脚)的外部低速时钟输入
f_{OSC}	: CGOSCCR<OSCSEL>指定的高速时钟
f_c	: 通过CGEHCLKSEL<EHCLKSEL> (高速时钟)指定的时钟
f_{gear}	: CGSYSCR<GEAR[2:0]>指定的时钟
f_{sys}	: 由CGCKSEL<SYSCK>(系统时钟)指定的时钟。
f_{periph}	: 由CGSYSCR<FPSEL0>指定的时钟。
$\phi T0$: CGSYSCR<PRCK[2:0]> 所指定的时钟(预分频器时钟)

高速时钟 f_c 和预分频器时钟 $\phi T0$ 的分割情况如下：

高速时钟: $f_c, f_c/2, f_c/4, f_c/8, f_c/16$

预分频时钟: $f_{periph}, f_{periph}/2, f_{periph}/4, f_{periph}/8, f_{periph}/16, f_{periph}/32$

6.3.2 复位后初始值

复位工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 (锁相环电路)	: 停止
高速时钟齿轮	: f_c (无分频)

复位工作可导致全部时钟配置与 f_{IHOSC} 相同。

$f_c = f_{IHOSC}$

$f_{sys} = f_c (=f_{IHOSC})$

$f_{periph} = f_c (=f_{IHOSC})$

$\phi T0 = f_{periph} (=f_{IHOSC})$

6.3.3 时钟系统图

图 6-1 给出了时钟系统图。

复位之后，显示时带有箭头记号的选择器的输入时钟即被设置为默认值。

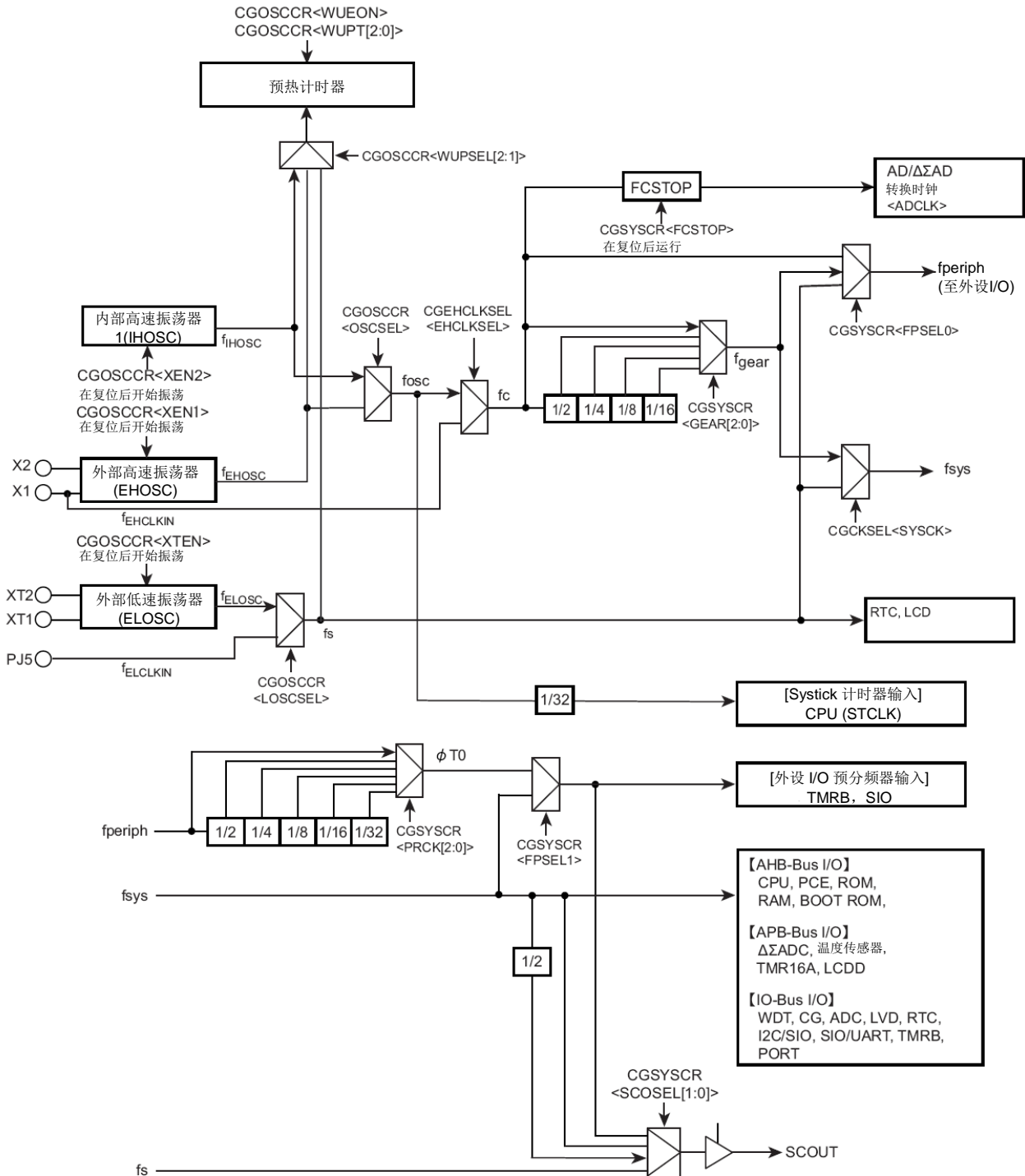


图 6-1 时钟方块图

6.3.4 预热功能

预热功能可利用预热计时器固定振荡器的稳计时间。在使用稳定的外部时钟时，无需使用预热功能。

在从 STOP/SLEEP 模式返回时，也可使用该预热功能。此时，从低功耗模式返回导致的中断会触发自动计时器计时。达到指计时间后，会输出系统时钟，同时 CPU 开始运行。

注：禁止在预热计时器运行期间转换为低功耗模式。

如何配置该预热功能。

1. 指定递增计数时钟

在CGOSCCR <WUPSEL2> <WUPSEL1>位中，指定该预热计数器的递增计数时钟。

注：不能将该外部高速时钟输入($f_{EHCLKIN}$)用作预热时钟。如拟将 $f_{EHCLKIN}$ 用作系统时钟，需将 f_{IOsc} (内部高速振荡)指定为预热时钟。在这种情况下，用户不能停止 f_{IOsc} 。

2. 指定预热计数数值

CGOSCCR<WUPT[11:0]><WUPTL[1:0]>可使用<WUPT>和<WUPTL>和低速时钟进行计数，也可使用<WUPT> 和高速时钟进行计数。将"00"设置到<WUPTL>，即可使用高速时钟。

可利用以下公式计算各设置值，并舍入下部四位。

注：将预热计数数值设置到CGOSCCR<WUPT><WUPTL>，等待至该值被反射，然后执行命令"WFI"即可转入待机模式。

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

注：预热计时器根据振荡器时钟工作，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。

<示例 1> 当使用高速振荡器 8MHz，并设置预热时间 5ms时。

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


舍入下部的 4 位，并将0x9C4 设置到CGOSCCR<WUPT[11:0]>。

3. 确认预热的启动和完成情况

通过软件(指令),利用 CGOSCCR<WUEON><WUEF>确认预热的开始与完成。当 CGOSCCR<WUEON>设为"1"时,预热开始递增计数。可用 CGOSCCR<WUEF>确认预热的完成情况。

预热功能设置的示例。

表 6-1 <示例> 从 STOP 模式到 NORMAL 模式转换(已选择内部高速振荡器)

CGOSCCR<WUPT> = "0x9C4"	: 指定预热时间
 CGOSCCR<WUPT>读取	: 确认反映重复的预热时间 直至读出数据为"0x9C4"
CGOSCCR<XEN2> = "1"	: 高速振荡器(fosc)启用
CGOSCCR<WUEON> = "1"	: 启动预热计时器
CGOSCCR<WUEF> 读取	: 等待状态变为"0"(预热完成)

6.3.5 系统时钟

以下时钟之一可被用作该系统时钟的源时钟: 内部高速振荡时钟, 外部高速振荡时钟(与振荡器或时钟输入连接)和外部低速时钟(与振荡器或时钟输入连接)。

如果必须具备高精度, 则不应使用内部高速振荡。

源时钟		频率
内部高速振荡(fIHOSC)		10MHz
外部高速振荡	振荡器(fEHOSC)	8 ~ 16MHz
	输入时钟(fEHCLKIN)	
外部低速振荡	振荡器(fELOSC)	30 ~ 34kHz
	输入时钟(fELCLKIN)	

在使用高速振荡器时, CGSYSCR<GEAR>可对系统时钟进行分频。尽管在运行期间可改变设置, 在轻微延迟之后还是会发生实际切换。

表 6-2 给出了通过时钟齿轮设置配置的工作频率示例。

表 6-2 系统时钟频率

fosc 频率 (MHz)	时钟变速器(CG)				
	1/1	1/2	1/4	1/8	1/16
8	8	4	2	1	-
10	10	5	2.5	1.25	-
16	16	8	4	2	1

↑复位后初始值

注: 在使用SysTick时, 请勿使用 1/16。



6.3.5.1 系统时钟切换

图 6-2 给出了系统时钟的切换方式。

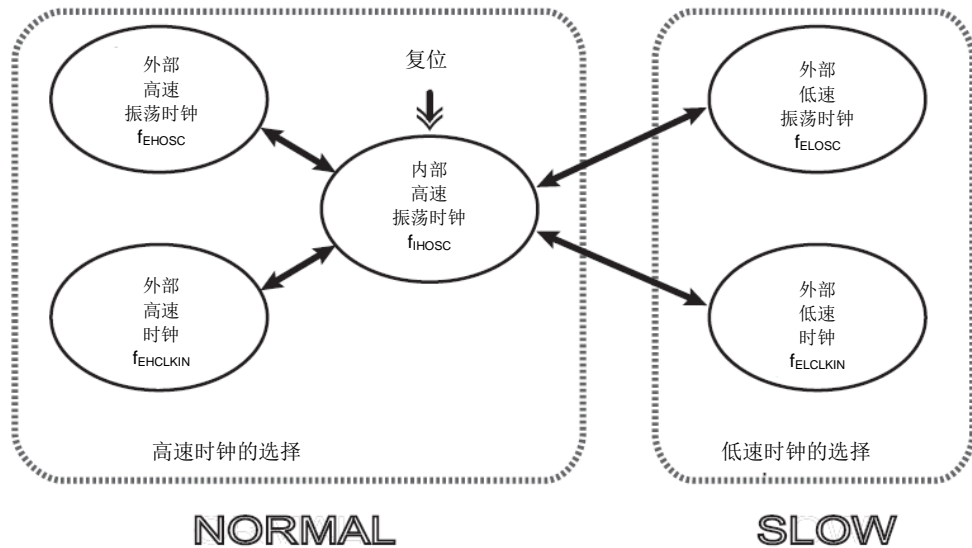


图6-2 系统时钟切换

在解除该复位信号之后，该内部高速振荡时钟即被设置为系统时钟。可从内部高速振荡时钟，外部高速振荡时钟和外部高速时钟中，选择一种高速时钟。可从外部低速振荡时钟和外部低速时钟中，选择一个低速时钟。不能在外部高速振荡时钟和外部高速时钟，以及外部低速振荡时钟和外部低速时钟之间切换时钟。

6.3.5.2 转换为 SLOW，STOP 和 SLEEP 模式时的注意事项

如拟转换为 SLOW 模式，STOP 模式或 SLEEP 模式，需首先通过设置 CGOSCCR<OSCSSEL>，将系统时钟切换到内部高速振荡时钟。在使用外部高速振荡时钟或外部高速时钟期间，不能切换到 STOP 或 SLEEP 模式。

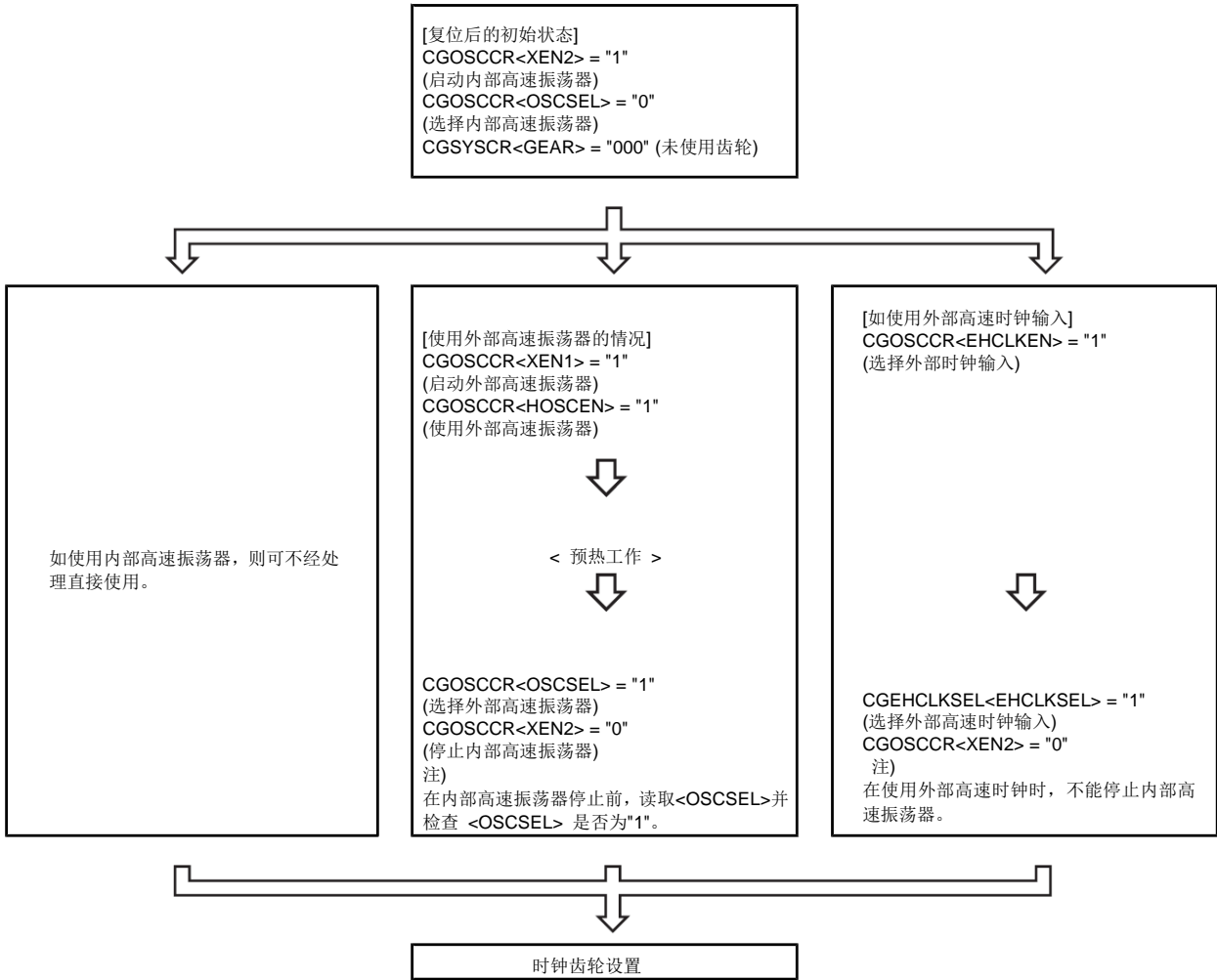
6.3.5.3 时钟设置

通过设置 CGOSCCR 和 CGEHCLKSEL，即可选择系统时钟。在选择该时钟之后，如有必要，需通过设置该 CGSYSCR 来设置该时钟齿轮。

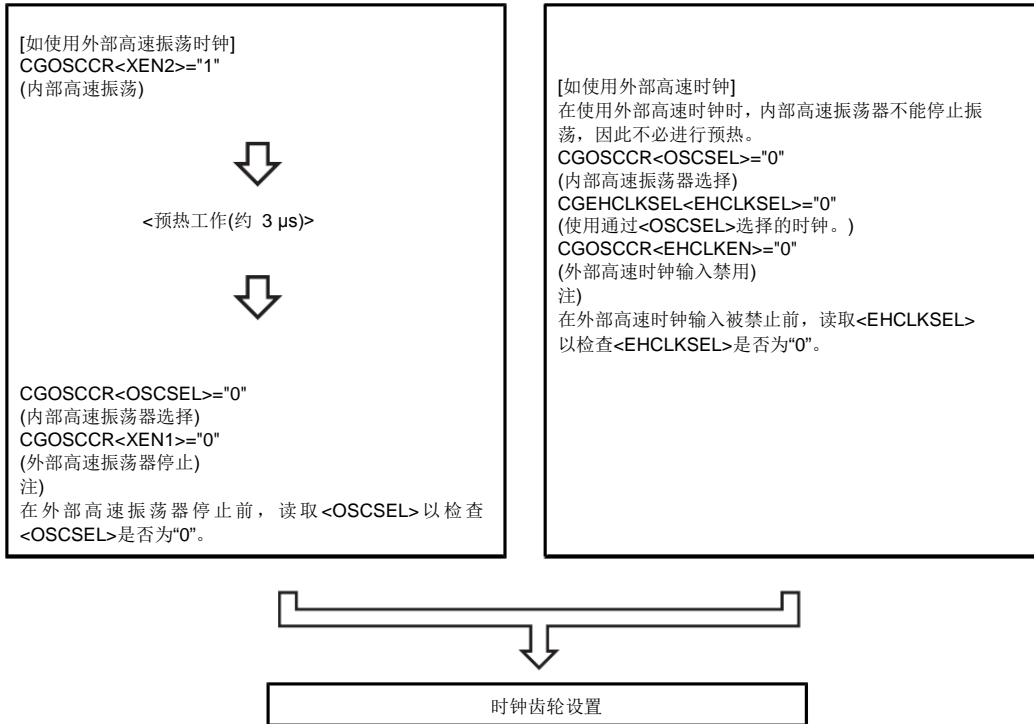
时钟设定顺序给出如下。

如将某低速时钟设置为系统时钟，则首先需在 NORMAL 模式下设置低速时钟，然后转入 SLOW 模式。

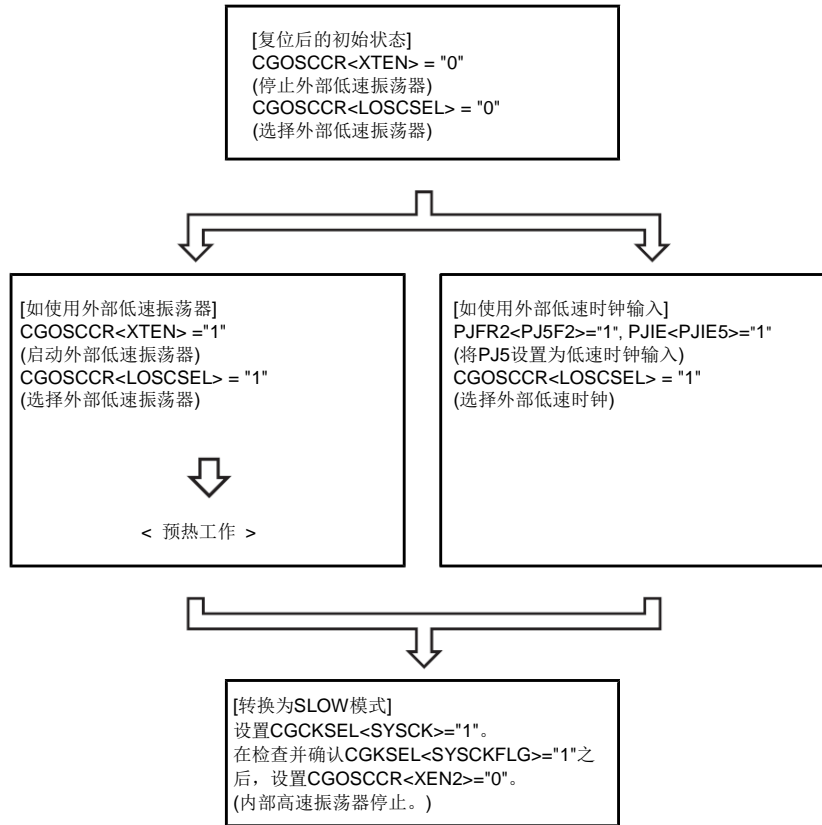
1. 高速时钟设定顺序(初始设置)



2. 高速时钟设置顺序(从外部高速时钟切换到内部高速时钟)



3. 低速时钟设定顺序



6.3.6 预分频器时钟控制

外设 IO (TMRB,SIO)带有一个时钟分频用时钟预分频器。在将时钟 $\phi T0$ 输入到各预分频器时,可按照 CGSYSCR<PRCK[2:0]>中的设置,对在 CGSYSCR<FPSEL0>中指定的"fperiph"时钟进行分频。在控制器复位之后,"fperiph/1"即被选作 $\phi T0$ 。

在时钟 f_s 被用作系统时钟 f_{sys} 时,通过将"1"设置到 CGSYSCR<FPSEL1>,即可将 f_s 用于预分频器时钟。

注:使用时钟变速器时,必须确保时间设置中每个外设功能的预分频器输出 ϕT_n 比 f_{sys} 慢($\phi T_n < f_{sys}$)。在计时器计数器或其它外设功能运行期间,请勿切换该时钟齿轮。

6.3.7 系统时钟引脚输出功能

TMPM061FWFG 允许从某引脚输出系统时钟。该 SCOUT 引脚可输出系统时钟 f_{sys} 和 $f_{sys}/2$, 以及外设 I/O $\phi T0$ 的预分频器输入时钟。

注:不保证SCOUT的系统时钟输出和内部时钟之间的相差(AC时序)。

通过将端口 J 寄存器, PJCR<PD4C>和 PJFR2<PJ4F2>设置为"1",该 PJ4 引脚即可变为 SCOUT 输出引脚。通过设置 CGSYSCR<SCOSEL[1:0]>,就可选择输出时钟。

表 6-3 给出了当 SCOUT 引脚设置至 SCOUT 输出时不同模式下的引脚状态

表 6-3 不同模式下 SCOUT 输出状态

SCOUT的选择 CGSYSCR	模式	低功耗模式		
	NORMAL	IDLE	SLEEP	STOP
<SCOSEL[1:0]> = "00"	保留			
<SCOSEL[1:0]> = "01"	输出f _{sys} /2 时钟			
<SCOSEL[1:0]> = "10"	输出f _{sys} 时钟			
<SCOSEL[1:0]> = "11"	输出φ _{T0} 时钟	固定为"0"或"1"。		

注 1: 无法保证SCOUT所输出的系统时钟输出与内部时钟之间的相差(AC时序)。

注 2: 如果为SCOUT选用f_{sys}, 则输出波形会在时钟齿轮切换时发生畸变。

6.4 模式与模式转换

6.4.1 模式转换

在 NORMAL 模式和 SLOW 模式下，系统时钟分别使用高速和低速时钟。

可将 IDLE, SLEEP 和 STOP 模式用作可通过暂停处理器内核运行降低功耗的低功耗模式。

在不使用低速时钟时，不能使用 SLOW 和 SLEEP 模式。

图 6-3 给出了模式转换图。

有关退出时睡眠的详细描述，请参照"Cortex-M0 技术参考手册"。

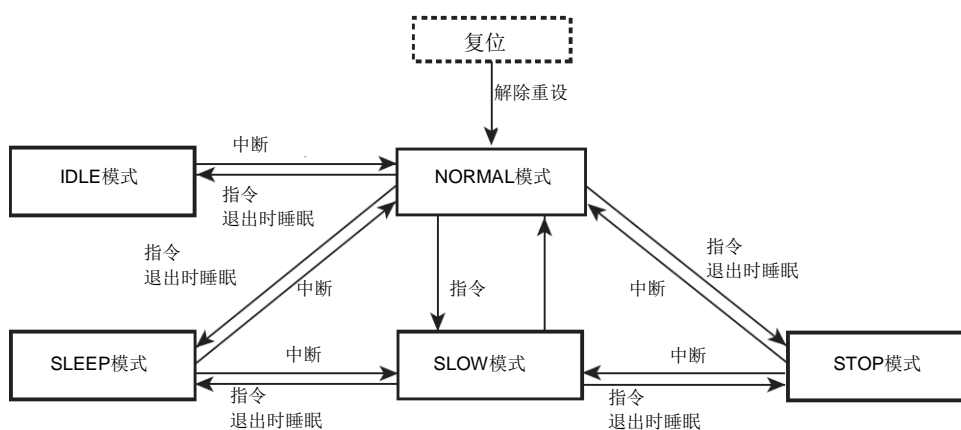


图 6-3 模式转换图

注：在使用外部高速振荡时钟(f_{EHOSC})和外部高速时钟($f_{EHCLKIN}$)时，不能将工作模式转换到STOP或SLEEP模式。首先需将时钟切换到内部高速振荡时钟(f_{IHOSC})。

6.5 工作模式

两种工作模式即NORMAL和SLOW可用。各模式的特征如以下所述。

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。也可使用低速时钟。

6.5.2 SLOW 模式

此模式可在高速时钟停止时用低速时钟工作 CPU 内核和外围硬件。与 NORMAL 模式相比，SLOW 模式降低了功耗。

在 SLOW 模式下，可工作的外设功能会受到限制。以下外设功能是可工作的：I/O 端口(PORT)，功率计算引擎(PCE)，计时器(TMRB 和 TMR16A)，看门狗计时器(WDT)，实时时钟(RTC)，LCD 驱动器，低电压检测(LVD)和温度传感器(TEMP)。

注 1：在SLOW模式下，在转换到SLOW模式前，会停止除可工作外设以外的全部功能。

注 2：在该慢速模式下，不得用Cortex-M0 NVIC寄存器的应用中断和复位控制寄存器<SYSRESETREG>执行复位工作。

6.6 低功耗模式

TMPM061FWFG有三种低功耗模式：IDLE，SLEEP和STOP。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，执行复位或生成该中断即可解除该模式。在通过中断实现解除前，必须提前进行设置。有关详细见"异常"一节。

注 1：TMPM061FWFG不会为低功耗模式的解除提供任何事件。并禁止通过执行WFE(等待事件)指令切换到低功耗模式。

注 2：TMPM061FWFG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。

注 3：禁止在预热计时器运行期间切换到低功耗模式。

注 4：在使用外部高速振荡时钟(f_{EHOSC})和外部高速时钟($f_{EHCLKIN}$)期间，不能切换到STOP或SLEEP模式。首先需将时钟切换到内部高速振荡时钟(f_{IHOSC})。

注 5：在SLEEP模式下使用LCD驱动器时，需将工作模式从SLOW模式转换到SLEEP模式。

IDLE，SLEEP和STOP模式的特征描述如下。

6.6.1 IDLE 模式

在该模式下,处理器内核的时钟处于停止状态。各外设功能的控制寄存器中,均带有一个用于在 IDLE 模式下启用或禁用运行的位。在进入 IDLE 模式时,在 IDLE 模式下其运行已被禁用的外设功能会停止运行,并保持当时的状态。

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,见各外设功能相关章节。

- 16-位计时器/事件计数器(TMRB)
- 16-位计时器 A(TMR16A)
- 串行通道(SIO/UART)
- 串行总线接口(I2C/SIO)
- 模拟数字转换器(ADC)
- 看门狗计时器(WDT)

6.6.2 SLEEP 模式

在该模式下,高速时钟停止工作。用低速时钟信号运行的实时时钟(RTC)计数器则保持运行。

通过解除该 SLEEP 模式,该设备会返回到先前的 SLEEP 模式,并开始运行。

6.6.3 STOP 模式

在 STOP 模式下,所有时钟包括内部振荡器都会停止运行。

通过解除该 STOP 模式,该设备会返回到先前的 STOP 模式,并开始运行。在解除该 STOP 模式之后,设置 200 μ s 或以上作为预热时间。

STOP 模式允许通过设置 CGSTBYCR<DRVE>选择引脚状态。表 6-4 显示 STOP 模式下的引脚状态。

表 6-4 STOP 模式下的引脚状态

用法	功能	端口	配置	I/O	<DRVE> = 0	<DRVE> = 1
端口	SWCLK	PE6	PEFR2<PE6F2>=1, PEIE<PE6IE>=1 或 DBGEN="0"在RESET的上升沿时	输入	o	
	SWDIO	PE7	PEF21<PE7F2>=1, PE7C<PE7C>=1, PEIE<PE7IE>=1 或 DBGEN="0"在RESET的上升沿时	输入	o	
				输出	在数据有效时启用 在数据无效时禁用	
	INT0 INT1 INT2 INT3	PF1 PJ2 PG0 PK0	PxFRn<PxmFn>=1 PxIE<PxmIE>=1	输入	o	
	上述或通用输入/输出端口以外的功能		PxIE<PxmIE>=1 PxCR<PxmC>=1	输入	x	o
		输出		x	o	
其它 端口	RESET, MODE			仅输入	o	
	COM0, COM1, COM2, COM3			仅输出	o	
	X1, XT1			仅输入	x	
	X2, XT2			仅输出	"高"电平输出	

o: 有效输入或输出。
x: 无效输入或输出。

注: x: 端口号 / m: 对应位

6.6.4 低功耗模式设置

通过设置待机控制寄存器 CGSTBYCR<STBY[2:0]>, 可指定低功耗模式。

表 6-5 给出了<STBY[2:0]>中的模式设置。

表 6-5 低功耗模式的设置

模式	CGSTBYCR <STBY[2:0]>
STOP	001
SLEEP	010
IDLE	011

注: 请勿设置除以上<STBY[2:0]>中所示以外的任何值。

6.6.5 各模式下的工作状态

表 6-6 给出了各模式下的工作状态。

表 6-6 各模式下的工作状态

模块	NORMAL			IDLE			SLOW		SLEEP		STOP
	内部 高速 振荡器 使用 (f _{IHOSC})	外部 高速 振荡器 使用 (f _{EHOSC})	外部 高速 时钟 使用 (f _{EHCLKIN})	内部 高速 振荡器 使用 (f _{IHOSC})	外部 高速 振荡器 使用 (f _{EHOSC})	外部 高速 时钟 使用 (f _{EHCLKIN})	外部 低速 振荡器 使用 (f _{ELOSC})	外部 低速 时钟 使用 (f _{ELCLKIN})	外部 低速 振荡器 使用 (f _{ELOSC})	外部 低速 时钟 使用 (f _{ELCLKIN})	
内部高速 振荡器	o	Δ	o (注 1)	o	Δ	o (注 1)	Δ (注 2)		-		-
外部高速 振荡器	Δ	o	Δ	Δ	o	Δ	Δ (注 2)		-		-
外部低速 振荡器	Δ	Δ	Δ	Δ	Δ	Δ	o	Δ	o	Δ	-
CG	o			o			o		o		-
处理器内核	o			-			o(注 6)		-		-
功率计算 引擎	o			o			o		-		-
IO端口	o			o			o		o		o(注 3)
SIO/UART	o			Δ			#(注 5)		-		-
I2C/SIO	o			Δ			#(注 5)		-		-
TMRB	o			Δ			o		-		-
TMR16A	o			Δ			o		-		-
WDT	o			Δ(注 7)			o		-		-
ADC	o			Δ			#(注 5)		-		-
ΔΣADC	o			o			#(注 5)		-		-
RTC	o			o			o		o		-
LCD驱动器	o			o			o		o(注 4)		-
温度 传感器	o			o			o		o		-
LVD	o			o			o		o		o

o: 在目标模式时, 工作可用。

- : 当切换到目标模式时, 模块时钟自动停止。

: 在转换为目标模式时, 必须停止该功能。

Δ: 在目标模式下, 允许通过软件选择禁用模块运行。

注 1: 在使用某个外部高速时钟时, 内部高速时钟不能停止。

注 2: 在工作模式从NORMAL模式转换到SLOW模式时, 高速振荡器(EHOSC和IHOSC)不会自动停止。设置CGOSCCR<XEN1>或<XEN2>, 以停止振荡。在进行从SLOW模式到NORMAL模式的转换之前, 设置 CGOSCCR<XEN1>或<XEN2>以启动振荡, 原因是该高速振荡器(EHOSC, IHOSC)不会自动启动振荡。

注 3: 该设置视CGSTBYCR<DRVE>设置而定。

注 4: 如拟在SLEEP模式下使用LCD驱动器, 则需将工作模式从SLOW模式转换为SLEEP模式。

注 5: 在SLOW模式下停止ADC, ΔΣADC, SIO/UART和I2C/SIO。

注 6: SysTick计时器不能用于SLOW模式。

注 7: 在IDLE模式下, 不能通过处理器内核清除看门狗计时器。

6.6.6 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。

详见表 6-7。

表 6-7 各模式的解除源

解除源		低功耗模式		
		IDLE	SLEEP	STOP
中断	INT0 ~ 3 (注 1), INTLVD	o	o	o
	INTRTC	o	o	x
	其它中断	o	x	x
SysTick中断		o	x	x
不可屏蔽中断(INTWDT)		o	x	x
RESET (RESET 引脚)		o	o	o

o: 在该模式被解除后启动中断处理(复位对LSI进行了初始化)

x: 不可用

注 1: 在通过中断电平模式从低功耗模式解除时，应保持电平，直至中断处理启动。如果该电平在中断处理启动之前已被改变，则无法启动正确的中断处理。

注 2: 如拟转入该低功耗模式，可设置CPU以禁止除该解除源以外的所有中断。否则，未指定的中断会执行解除。

- 通过中断请求解除

为了通过中断来解除低功耗模式，必须事先设置CPU，以检测中断。除了对CPU的设置，还应设置时钟生成器，以检测解除停止模式时需使用的中断。

- 通过SysTick中断实现解除

SysTick中断只能在IDLE模式下使用。

- 通过不可屏蔽中断(NMI)实现解除

INTWDT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RESET引脚进行解除。此后，模式切换到NORMAL模式，所有寄存器按正常复位的情况初始化。

详见"中断"。

6.6.7 预热

在模式转换过程中，可能需进行预热，以实现内部振荡器的稳定振荡。

在从 STOP 到 NORMAL/SLOW，或从 SLEEP 到 NORMAL 的模式转换过程中，预热计数器会被自动激活。然后在经过一段预热时间后，系统时钟输出启动。在执行该指令并进入 STOP/SLEEP 模式之前，需在 CGOSCCR<WUPSEL2><WUPSEL1> 中设置某个预热时间，并在 CGOSCCR<WUPT><WUPTL> 中设置该预热时间。

注：不得将外部高速时钟输入($f_{EHCLKIN}$)用作预热时钟。

当低速振荡器被禁用时，在从 NORMAL 到 SLOW/SLEEP 的转换过程中，必须进行预热，以确保内部振荡器实现稳定。启用低速振荡器，然后通过软件激活预热。

在从 SLOW 到 NORMAL 的转换过程中，在高速振荡器已被禁用的情况下，需启用高速振荡器，然后激活预热。

表 6-8 显示了各模式转换是否需要进行预热。

表 6-8 模式转换预热设置

模式转换	预热设置
NORMAL → IDLE	不需要
NORMAL → SLEEP	(注 1)
NORMAL → SLOW	(注 1)
NORMAL → STOP	不需要
IDLE → NORMAL	不需要
SLEEP → NORMAL	用通过CGOSCCR<WUPSEL2>选择的时钟启动自动预热
SLEEP → SLOW	不需要
SLOW → NORMAL	(注 2)
SLOW → SLEEP	不需要
SLOW → STOP	不需要
STOP → NORMAL (注 3)	用通过CGOSCCR<WUPSEL2>选择的时钟启动自动预热。
STOP → SLOW (注 3)	用通过CGOSCCR<LOSCSEL>选择的时钟启动自动预热。

注 1：对低速振荡器被禁用时，则启用低速振荡器，然后通过软件激活预热。

注 2：当高速振荡器被禁用时，则启用高速振荡器，然后通过软件激活预热。

注 3：在解除该STOP模式之后，设置 200 μ s或以上作为预热时间。

6.6.8 模式转换时的时钟工作

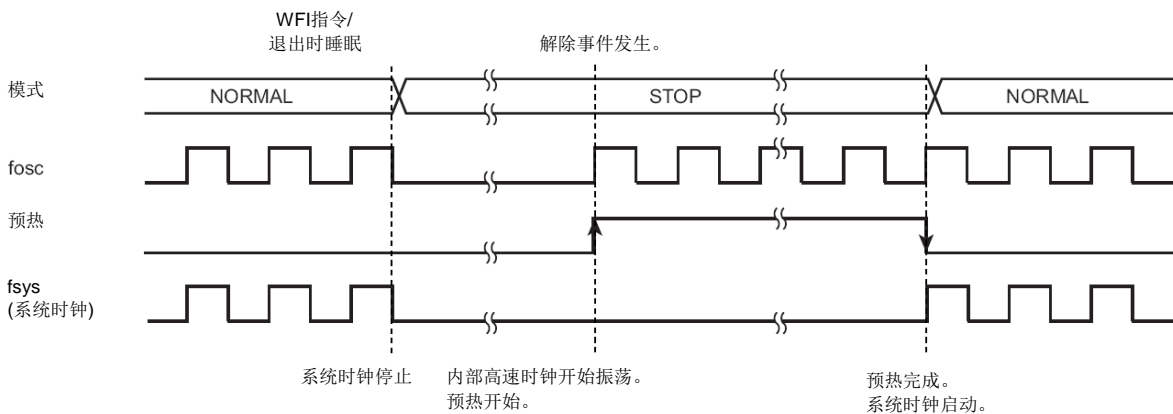
模式切换时的时钟工作如下。

6.6.8.1 工作模式的转换：NORMAL → STOP → NORMAL

当从STOP模式返回到NORMAL模式时，预热自动开启。需将预热时间设置到CGOSCCR<WUPT>中，并在进入STOP模式之前，选择与CGOSCCR<OSCSEL>~<WUPSEL2><WUPSEL1>相同的时钟源。

注：在使用外部高速振荡时钟(f_{EHOSC})和外部高速时钟($f_{EHCLKIN}$)时，不得将工作模式转换到STOP模式或SLEEP模式。首先需将时钟切换到内部高速振荡时钟(f_{IHOSC})。

在通过复位返回到NORMAL模式时，不会引发自动预热。使复位信号保持在被断言状态，直至振荡器运行变为稳定。

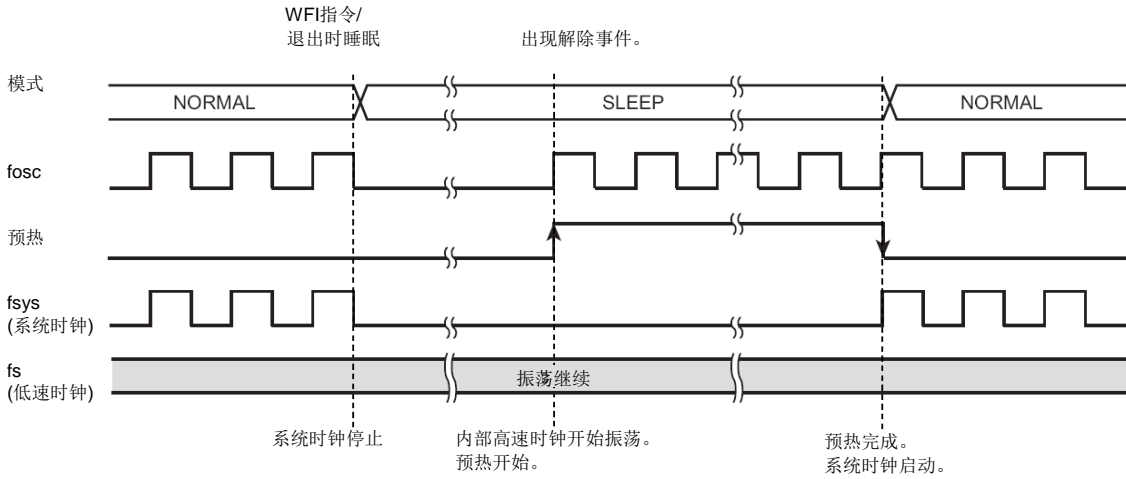


6.6.8.2 工作模式的转换：NORMAL → SLEEP → NORMAL

在从SLEEP模式返回到NORMAL模式时，预热会被自动激活。在进入SLEEP模式之前，需将预热时间设置到CGOSCCR<WUPT>中，并选择与CGOSCCR<OSCSEL>~<WUPSEL2><WUPSEL1>相同的时钟源。

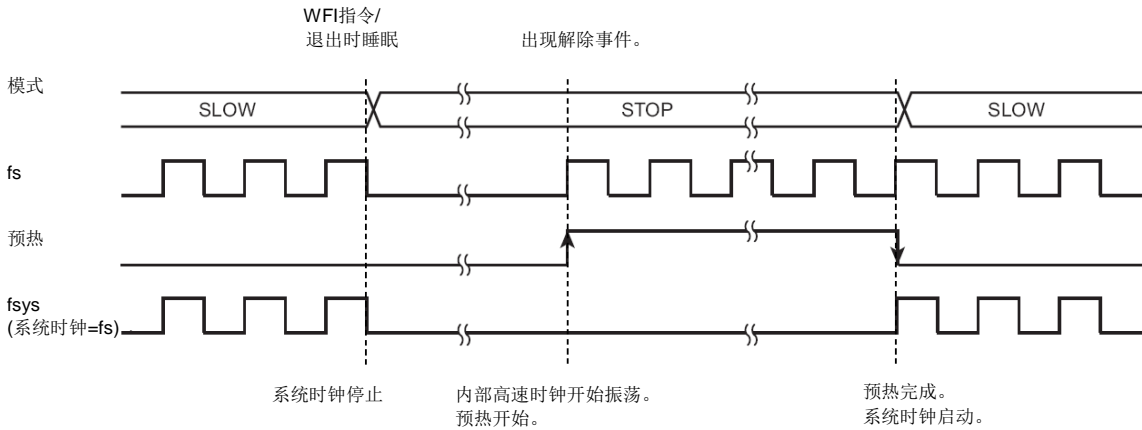
注：在使用外部高速振荡时钟(f_{EHOSC})和外部高速时钟($f_{EHCLKIN}$)时，不得将工作模式转换到STOP模式或SLEEP模式。首先需将时钟切换到内部高速振荡时钟(f_{IHOSC})。

在通过复位返回到NORMAL模式时，不会引发自动预热。使复位信号保持在被断言状态，直至振荡器运行变为稳定。



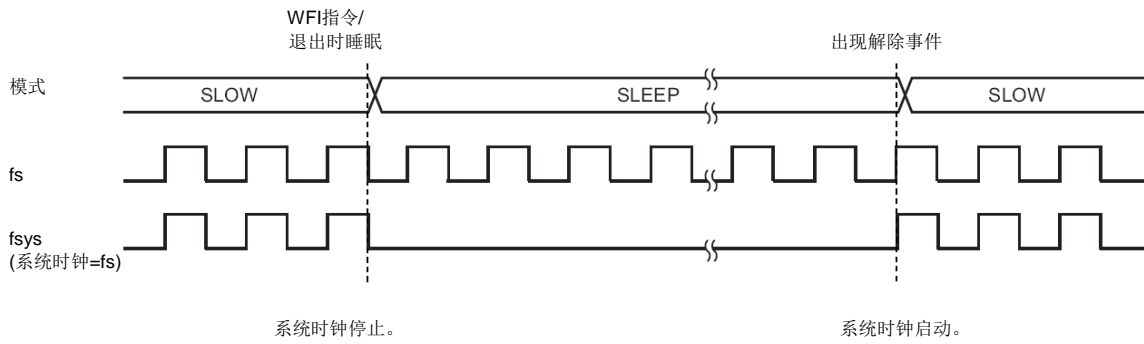
6.6.8.3 工作模式的转换：SLOW → STOP → SLOW

预热自动激活。在进入停止模式前需要提前设置预热时间。



6.6.8.4 工作模式的转换: SLOW → SLEEP → SLOW

在睡眠模式下，低速时钟持续振荡。无需做预热设置。



7. 异常

本节对各异常的特征，类型与处理进行说明。

异常与CPU内核之间有着密切的关系。如有需要，请参照“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。

所有异常由CPU中的嵌套向量中断控制器(NVIC)按照各自优先等级进行处理。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。


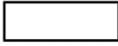
7.1.1 异常类型

Cortex-M0 中存在以下各类型的异常。

有关各异常的详细说明，请参照“Cortex-M3 技术参考手册”。

- 复位
- 不可屏蔽中断(NMI)
- 硬故障
- SVCcall(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。

各步骤将在本节后面的内容中进行说明。

处理	说明	参见
 由 CG/CPU 进行检测	CG/CPU 检测到异常请求。	 章节 7.1.2.1
		
 由CPU进行处理	CPU 处理该异常请求。	 章节 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。	 章节 7.1.2.3
		
 从异常返回	CPU 转入到另一 ISR，或返回到前一程序。	 章节 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CPU的指令执行，存储器访问，以及来自外部中断引脚或外设功能的中断请求。

在CPU执行可引发异常的某条指令时，或在指令执行期间出现错误状态时，就会发生异常。

从不执行(XN)区域取出指令，或故障区域的访问破坏，也会导致发生异常。

可从外部中断引脚或外设功能生成中断请求。对于拟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。有关详细请参照“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。

异常的优先级如表 7-1 所示。“可配置”表示异常的优先级可以手工指定。

表 7-1 异常类型和优先级

序号	异常类型	优先级	说明
1	复位	-3 (最高)	复位引脚, WDT或SYSRETRREQ
2	非屏蔽中断	-2	$\overline{\text{NMI}}$ 引脚或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	可挂起系统服务申请
15	SysTick	可配置	源自系统计时器的通知
16~	外部中断	可配置	外部中断引脚或外设功能 (注 2)

注: 各产品的外部中断的来源和数目各不相同。相关的详细,

参照“7.5.1.5中断源列表”。

(3) 优先级设置

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。

<PRI_n>的配置是两个位, 因此, 优先级的配置范围为 0 ~ 3。最高优先级为“0”。如存在具备相同优先级的多个要素, 则编号越小, 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(抢占)

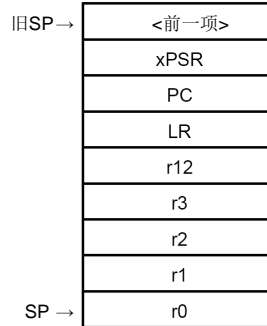
在异常发生时, CPU 会暂停当前正在执行的进程, 并转移到中断服务程序。这就是所谓的“抢占”。

(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:

- 程序计数器(PC)
- 程序状态寄存器(xPSR)
- r0 - r3
- r12
- 链接寄存器(LR)

到压栈完成时, SP渐减八个字。在寄存器内容已被压栈后, 堆栈状态如下所示。



(2) 读取 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顶地址的一个向量表。

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”。

迟后到达型异常可导致CPU取用新的向量地址，以转至相应的ISR，但CPU不会将寄存器内容推至存储栈。

(4) 向量表

该向量表的配置如下所示。

用户必须始终设置前四个字(栈顶地址，复位ISR地址，NMI ISR地址，以及硬故障ISR地址)。如有必要，可设置其它异常的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	SVCall	ISR地址	可选
0x30 ~ 0x34	保留		
0x38	PendSV	ISR地址	可选
0x3C	SysTick	ISR地址	可选
0x40	外部中断	ISR地址	可选

7.1.2.3 执行 ISR

ISR 可执行相应异常的必要处理。ISR 必须由用户编制。

ISR 可能需包括中断请求清除用代码，以确保在恢复正常程序执行时不会再次发生相同的中断。

关于中断处理的详细信息，参见“7.5 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

从某个ISR返回后，CPU会采取如下行动：

- 尾链

如果存在某个挂起异常且不存在任何堆栈异常，或该挂起异常的优先级高于所有堆栈异常，则CPU会返回到该挂起异常的ISR。

在这种情况下，CPU会在退出一个ISR并进入另一个ISR时，跳过八个寄存器的堆栈上托和八个寄存器的推栈。这就是所谓的“尾链”。

- 返回到上一堆栈ISR

如果不存在任何挂起异常，或优先级最高的堆栈异常的优先级高于最高优先级挂起异常，则CPU会返回到上一堆栈ISR。

- 返回到前一程序

如果不存在任何挂起或堆栈异常，则CPU会返回到前一程序。

(2) 异常出口顺序

在从某ISR返回时，CPU会执行以下工作：

- 弹出八个寄存器

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

复位异常产生于以下三个来源：

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

在某个外部复位引脚从“低”变为“高”时，会发生外部复位引脚A复位。

- WDT所引发的复位异常

看门狗计时器(WDT)具备复位生成功能。有关详细，见WDT相关节次。

- 由SYSRESETREQ导致的复位异常

通过在NVIC的应用中断和复位控制寄存器中设置SYSRESETREQ位，可生成复位。

7.3 非屏蔽中断(NMI)

看门狗计时器(WDT)具备非屏蔽中断生成功能。有关详细，见WDT相关节次。

7.4 SysTick

SysTick具备中断功能采用CPU的系统计时器。

当在SysTick重新加载值寄存器中设置某个值，并在SysTick控制和状态寄存器中启用SysTick特征时，计数器会加载重新加载值寄存器的设置值，开始递减计数。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何时达到“0”。

注：在本产品中，systick计时器可根据寄存器CGOSCCR的各<OSCSSEL>位所选择的fosc进行计数。

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。

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

用于解除待机的外设功能(路径 2)，以及源自外部中断引脚(路径 3)的中断被输入到时钟发生器，并通过待机解除逻辑(路径 4 与 5)被输入到 CPU。

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)。

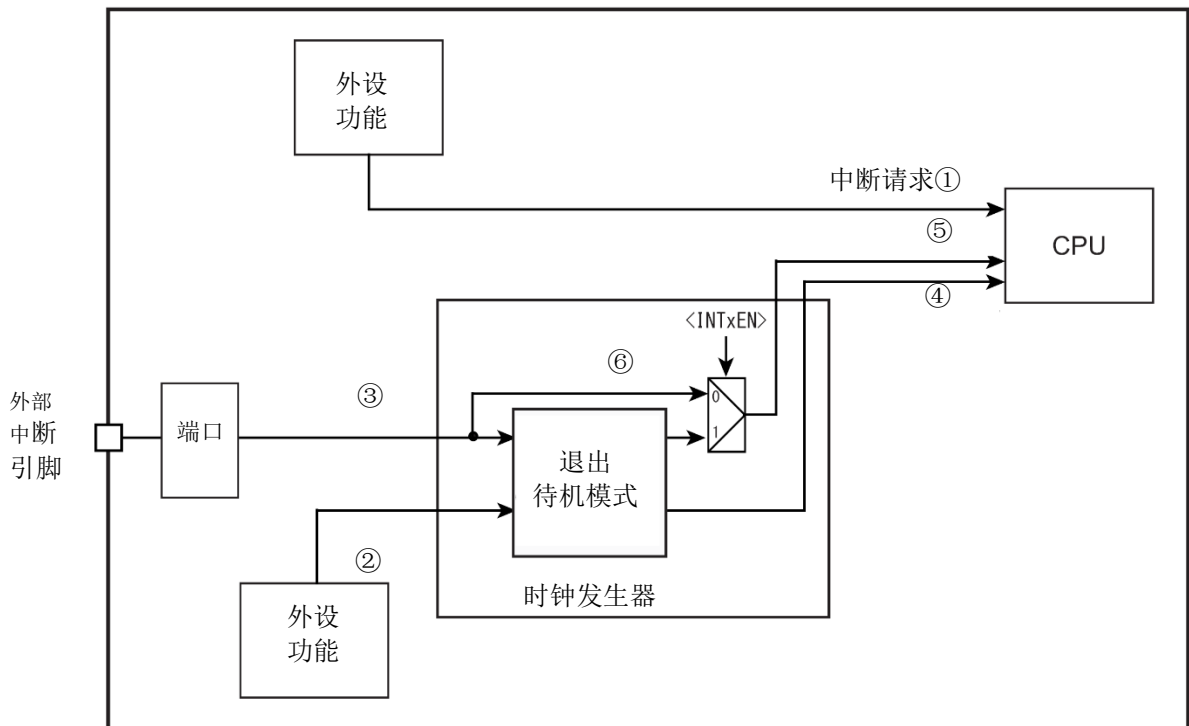


图 7-1 中断路径

7.5.1.5 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		激活电平 (解除待机)	CG中断模式 控制寄存器
0	INTPCE	PCE		
1	INTDSAD0	$\Delta\Sigma/A/D$ 转换完成中断 (单元 0)		
2	INTDSAD1	$\Delta\Sigma/A/D$ 转换完成中断 (单元 1)		
3	INTDSAD2	$\Delta\Sigma/A/D$ 转换完成中断 (单元 2)		
4	INTLVD	LVD中断	高沿	CGIMCGA
5	INT0	中断引脚 0	高/低 缘/电平 可选	
6	INT1	中断引脚 1		
7	INT2	中断引脚 2		
8	INT3	中断引脚 3		CGIMCGB
9	INTRX0	串行接收 (通道 0)		
10	INTTX0	串行传输 (通道 0)		
11	INTRX1	串行接收 (通道 1)		
12	INTTX1	串行传输 (通道 1)		
13	INTRX2	串行接收 (通道 2)		
14	INTTX2	串行传输 (通道 2)		
15	INTRX3	串行接收 (通道 3)		
16	INTTX3	串行传输 (通道 3)		
17	INTSBI	串行总线接口		
18	INTTB0	16位TMRB匹配检测 0		
19	INTTB1	16位TMRB匹配检测 1		
20	INTCAP00	16位TMRB输入捕获 0 通道 0)		
21	INTCAP01	16位TMRB输入捕获 1 通道 0)		
22	INTCAP10	16位TMRB输入捕获 0(通道 1)		
23	INTCAP11	16位TMRB输入捕获 1(通道 1)		
24	INTT16A0	16位TMR16A匹配检测(通道 0)		
25	INTT16A1	16位TMR16A匹配检测 (通道 1)		
26	INTT16A2	16位TMR16A 匹配检测 (通道 2)		
27	INTT16A3	16位TMR16A 匹配检测 (通道 3)		
28	INTT16A5	16位TMR16A 匹配检测 (通道 5)		
29	INTT16A6	16位TMR16A 匹配检测 (通道 6)		
30	INTRTC	实时时钟	低沿	CGIMCGB
31	INTAD	A/D转换完成中断		

7.5.1.6 激活电平

激活电平表明中断源信号的何种改变会导致中断。CPU 会将“高”电平中断信号识别为中断。被从外设功能直接发送到 CPU 的中断信号会接受配置，以输出“高”并指示中断请求。

激活电平，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会被设置为上升沿或下降沿触发。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)。

对于用于清除备用模式的中断源，还必须设置相应的时钟生成器寄存器。启用 CGIMCGx<INTxEN> 位，并指定 CGIMCGx<EMCGx> 位中的激活电平。用户必须从表 7-2 所述的各外设功能设置各中断请求的激活电平。

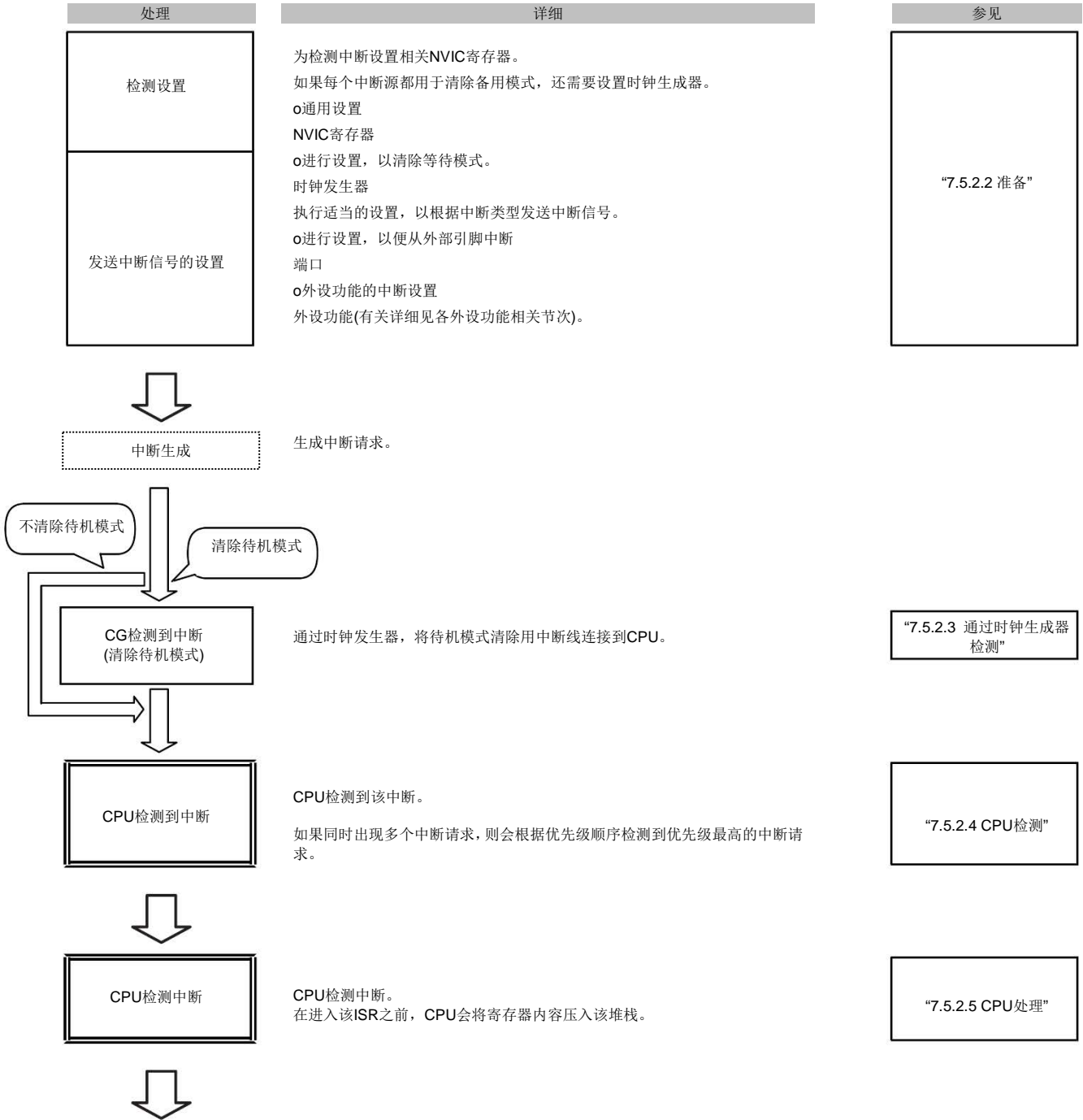
可用一个"高"电平信号将时钟发生器所检测到的中断请求通知 CPU。

7.5.2 中断处理

7.5.2.1 流程图

下文给出了中断的处理方式。

在以下说明中， 指硬件处理。 指软件处理。



处理

ISR 执行



返回到前一程序

详情

ISR的程序
如有必要，可清除该中断源。

配置以返回ISR的前一程序。

参见

"7.5.2.6 中断服务程序 (ISR)"

7.5.2.2 准备

在为中断做准备时，需注意配置顺序，以避免中途发生意外的中断。

一般而言，必须按以下顺序初始化某个中断或改变其配置。通过 CPU 禁用该中断。从 CPU 按最远的路径进行配置。然后该 CPU 使能该中断。

在配置该时钟发生器时，用户必须遵循本节所述的次序，以免导致任何意外中断。首先，配置预置条件。其次，清除与时钟生成器内中断有关的所有数据，然后启用中断。

以下各节均按中断处理的顺序出现，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. 预置(1) (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

为了使CPU不接受任何中断，应向PRIMASK寄存器写入"1"。所有中断与异常非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)

(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，工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 0~3 的优先级。优先级 0 属于最高优先级。如果多个源具备相同的优先级，则编号最小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) 预置(1) (外部引脚中断)

将"1"设置到相应引脚的端口功能函数寄存器。通过设置PxFRn[m]，即可允许将该引脚用作功能引脚。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		
PxFRn<PxmFn>	←	"1"
PxIE<PxIE>	←	"1"

注：x: 端口号 / m: 对应位 / n: 功能寄存器编号

在STOP模式以外的各模式中，通过设置PxIE以使能输入，即使能相应的中断输入，且不考虑PxFR设置。注意不得启用未使用的中断。另外，需注意“7.5.1.4 外部中断使用注意事项”。

(4) 预配置 (2) (来自外设功能的中断)

该设置随所用外设功能的不同而不同。有关详细，见各外设功能相关章节。

(5) 预配置 (3) (中断设置挂起寄存器)

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		
中断设置挂起[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，工作人员需在时钟发生器的CGIMCG寄存器中设置激活电平，并启用中断。CGIMCG寄存器能配置各源。

在启用某中断之前，清除已保留的相应中断请求。这种处理可防止意外中断。为清除相应的中断请求，应向CGICRCG寄存器中写入拟用中断对应的数值。有关各数值的具体信息，参见“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。但必须输入“高”脉冲信号或“高”电平信号，以便CPU检测出该信号为中断请求。此外，还请参照“7.5.1.4 外部中断引脚使用注意事项”。

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

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。用中断设置使能寄存器使能该预定中断。将寄存器的各位指定到单个中断源。

向中断清除挂起寄存器的相应位写入"1"，即可解除该被暂停的中断。向中断设置使能寄存器的相应位写入"1"，即可使能该中断。

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，不必进行该工作。

最后，PRIMASK寄存器被清零。

NVIC寄存器		
中断 清零-挂起 [m]	←	"1"
中断 设置-启用 [m]	←	"1"
中断屏蔽寄存器		
PRIMASK	←	"0"

注: m: 相应位

7.5.2.3 通过时钟生成器检测

如果用中断源退出待机模式，则会按照时钟发生器中所指定的激活电平进行检测，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。在检测到电平敏感型中断请求前，必须将其保持在激活电平；否则，当信号电平由激活变为不激活时，中断请求将不存在。

在该时钟发生器检测到某个中断请求时，其会不断按"高"电平向 CPU 发送该中断信号，直至该中断请求在该 CG 中断请求清除(CGICRCG)寄存器中被清除。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。

7.5.2.4 CPU 检测

CPU 可检测到具备最高优先级的中断请求。

7.5.2.5 CPU 处理

一旦检测到中断，CPU 就会在进入该 ISR 时将 PC, PSR, r0-r3, r12 和 LR 的内容压入到该堆栈中。

7.5.2.6 中断服务程序(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) 在 ISR 期间加入

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。Cortex-M3 内核会将 PC, PSR, r0-r3, r12 和 LR 的内容自动推向该存储栈。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。我们建议工作员压入可能会被重写的通用寄存器。

(2) 清除一中断源

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。通过自动清除该中断源，即可从时钟发生器解除该中断请求信号。

若某中断信号被设为边缘敏感型，可设定 CGICRCG 请求中的相应数值，从而清除中断请求。在某个激活沿再次发生时，就会检测到新的中断请求。

7.6 异常/中断相关的寄存器

本章所述CPU的 NVIC寄存器和时钟发生寄存器如以下所列附有相应的地址。

7.6.1 寄存器列表

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-启用寄存器	0x0100
中断清除-启用寄存器	0x0180
中断设置-挂起寄存器	0x0200
中断清除-挂起寄存器	0x0280
中断优先权寄存器	0x0400 ~ 0x0430
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

时钟发生器寄存器 基址 = 0x400F_3000

寄存器名称	地址
CG中断模式控制寄存器A	CGIMCGA 0x0020
CG中断模式控制寄存器B	CGIMCGB 0x0024
CG中断请求清除寄存器	CGICRCG 0x0014
复位标志寄存器	CGRSTFLG 0x001C

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: 外部参考时钟(fosc/32) 1: CPU时钟(fsyst)
1	TICKINT	R/W	0: 请勿让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: 在该产品中, fosc(其由寄存器CGOSCCR的<OSCSEL>位选择)被用作外部参考时钟。

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	R/W	重新加载值 在计时器达到“0”时，将拟加载的值设置到SysTick当前值寄存器中。

7.6.2.3 SysTick 当前值寄存器

	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	R/W	[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，均可将其清为0。 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	1	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	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	NOREF	R	0: 具备参考时钟 1: 无参考时钟
30	SKEW	R	0: 校准值是 10 ms。 1: 校准值非 10 ms。
29-24	-	R	读作 0。
23-0	TENMS	R	校准值(注)

注：TMPM061FWFG不会确定该校准值。

7.6.2.5 中断设置-启用寄存器

	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:有关中断和中断号的描述,见“7.5.1.5 中断源列表”一节。

7.6.2.6 中断清除-使能寄存器

	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：有关中断和中断号的描述，见“7.5.1.5 中断源列表”一节。

7.6.2.7 中断设置-挂起寄存器

	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	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	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	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	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	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	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	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
31-0	SETPEND	R/W	<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: 有关中断和中断号的描述, 见“7.5.1.5 中断源列表”一节。

7.6.2.8 中断清除-挂起寄存器

	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	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	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	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	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	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	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	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
31-0	CLRPEND	R/W	<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。但是, 对于已处于使用状态的中断而言, "1"的写入不会产生任何作用。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: 有关中断和中断号的描述, 见“7.5.1.5 中断源列表”一节。

7.6.2.9 中断优先级寄存器

以下给出了各中断优先级寄存器的地址(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 内核有两个位用于指定优先级。

以下给出了中断号 0~3 中断优先级寄存器的各字段。其它所有中断号的中断优先级寄存器均具备完全相同的字段。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_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	中断编号 3 的优先级
29-24	-	R	读作 0。
23-22	PRI_2	R/W	中断编号 2 的优先级
21-16	-	R	读作 0。
15-14	PRI_1	R/W	中断编号 1 的优先级
13-8	-	R	读作 0。
7-6	PRI_0	R/W	中断编号 0 的优先级
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 (写入/ VECTKEY- STAT(读取))	R/W	寄存器键 [写入] 如拟写入到该寄存器, 则 <VECTKEY>字段中必须有0x5FA字样。 [读取] 读作0xFA05。
15	ENDIANESS	R/W	字节存储次序位 (注1) 1: 大端字节顺序 0: 低位优先
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	清除活性向量位 1: 清零所有活性NMI, 故障, 及中断的状态信息。 0: 未清除。 该位可自行清除。 应用的责任是为了重初始堆栈。
0	-	R	读作 0。

注 1: 本产品仅可采用小端字节存储次序存储器格式。

注 2: 在SYSRESETREQ被输出的同时, 即对本产品实施复位。通过复位, <SYSRESETREQ>即被清除。

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 内核有两个位用于指定优先级。

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会失效。

	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
31-30	PRI_15	R/W	SysTick的优先级
29-24	-	R	读作 0。
23-22	PRI_14	R/W	PendSV的优先级。
21-16	-	R	读作 0。
15-14	PRI_13	R/W	保留
13-8	-	R	读作 0。
7-6	PRI_12	R/W	保留
5-0	-	R	读作 0。

7.6.2.12 系统处理器控制和状态寄存器

	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 1: 挂起
14-0	-	R	读作 0。

注：由于这些位的清除和设置不会修复堆栈内容，所以在清除或设置时必须十分小心。

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	未定义	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	未定义	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	未定义	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	未定义	0

位	比特符号	类型	功能
31	-	R	读作 0。
30-28	EMCG3[2:0]	R/W	INT2待机清除请求的激活电平设置(101~111: 设置已被禁止) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双沿
27-26	EMST3[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: 双沿
25	-	R	读作未定义。
24	INT3EN	R/W	INT2清除输入 0: 禁用 1: 启用
23	-	R	读作 0。
22-20	EMCG2[2:0]	R/W	INT1待机清除请求的激活电平设置。(101~111: 设置已被禁止) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双沿
19-18	EMST2[1:0]	R	INT1待机清除请求的激活电平 00: - 01: 上升沿 10: 下降沿 11: 双沿
17	-	R	读作未定义。
16	INT2EN	R/W	INT1清除输入 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	INT0 待机清除请求的激活电平设置(101~111: 设置已被禁止) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双沿
11-10	EMST1[1:0]	R	INT0 待机清除请求的激活电平 00: - 01: 上升沿 10: 下降沿 11: 双沿
9	-	R	读作未定义。
8	INT1EN	R/W	INT0 清除输入 0: 禁用 1: 启用
7	-	R	读作0。
6-4	EMCG0[2:0]	R/W	INTLVD待机清除请求的激活电平设置。 (注) 按以下所示对其进行设置。 011: 上升沿
3-2	EMST0[1:0]	R	INTLVD待机清除请求的活动电平 00: - 01: 上升沿 10: 下降沿 11: 双沿
1	-	R	读作未定义。
0	INT0EN	R/W	INTLVD清除输入 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。可参照<EMSTx>检查待机复位用激活电平。如果用CGICRCG寄存器清除了各中断, 则<EMSTx>也同时被清除。

注 2: 请首先指定供边使用的位, 然后再指定供<INTxEN>使用的位。禁止同时对两者进行设置。

7.6.3.2 CGIMCGB(CG 中断模式控制寄存器 B)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	1	0	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	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	未定义	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	未定义	0

位	比特符号	类型	功能
31	-	R	读作 0。

位	比特符号	类型	功能
30-28	-	R/W	写入任何值。
27-25	-	R	读作 0。
24	-	R/W	写入 0。
23	-	R	读作 0。
22-20	-	R/W	写入任何值。
19-17	-	R	读作 0。
16	-	R/W	写入 0。
15	-	R	读作 0。
14-12	EMCG5[2:0]	R/W	对INTRTC待机清除请求的激活电平设定 (注) 按以下所示对其进行设置。 010: 下降沿
11-10	EMST5[1:0]	R	INTRTC待机清除请求的激活电平 00: - 01: 上升沿 10: 下降沿 11: 双沿
9	-	R	读作未定义。
8	INT5EN	R/W	INTRTC清除输入 0: 禁用 1: 启用
7	-	R	读作 0。
6-4	EMCG4[2:0]	R/W	INT3 等待清除请求的激活电平设置。(101~111:设置已被禁止) 000: "低"电平 001: "高"电平 010: 下降沿 011: 上升沿 100: 双沿
3-2	EMST4[1:0]	R	INT3 待机清除请求的激活电平 00: - 01: 上升沿 10: 下降沿 11: 双沿
1	-	R	读作未定义。
0	INT4EN	R/W	INT3 清除输入 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。可参照<EMSTx>检查待机复位用激活电平。如果用CGICRCG寄存器清除了各中断, 则<EMSTx>也同时被清除。

注 2: 请首先指定供边使用的位, 然后再指定供<INTxEN>使用的位。禁止同时对两者进行设置。

7.6.3.3 CGICRCG(CG 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: INTLVD 0_0001: INT0 0_0010: INT1 0_0011: INT2 0_0100: INT3 0_0101: INTRTC 0_0110 ~ 1_1111: 设置已被禁止 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	-	-	-	DBGSTF	-	WDTRSTF	-	PINRSTF
复位后	0	0	0	0	0	0	0	1

位	比特符号	类型	功能
31-6	-	R	读作 0。
5	-	R	写入 0。
4	DBGSTF	R/W	调试复位标志 (注 1) 0: 写入"0" 1: 从SYSRESETREQ复位
3	-	R	写入 0。
2	WDTRSTF	R/W	WDT复位标志 0: 写入"0" 1: 从WDT复位
1	-	R	写入 0。
0	PINRSTF	R/W	$\overline{\text{RESET}}$ 引脚标志 0: 写入"0" 1: 从 $\overline{\text{RESET}}$ 引脚复位

注 1: 该标志表示由应用中断的SYSRESETREQ位和CPU NVIC的复位控制寄存器生成的复位。

注 2: 寄存器不会被自动清除。写入"0"寄存器即被清除。

8. 输入/输出端口

8.1 端口功能

8.1.1 功能列表

这些端口还可用作各内置外设功能的输入/输出引脚。表 8-1 给出了端口功能列表。

表 8-1 端口功能列表(端口A ~ 端口K)

端口	引脚名称	输入/输出	施密特输入	噪声滤波器	功能引脚
端口 A					
	PA0	I/O	o	-	SEG0
	PA1	I/O	o	-	SEG1
	PA2	I/O	o	-	SEG2
	PA3	I/O	o	-	SEG3
	PA4	I/O	o	-	SEG4
	PA5	I/O	o	-	SEG5
	PA6	I/O	o	-	SEG6
	PA7	I/O	o	-	SEG7
端口 B					
	PB0	I/O	o	-	SEG8
	PB1	I/O	o	-	SEG9
	PB2	I/O	o	-	SEG10
	PB3	I/O	o	-	SEG11
	PB4	I/O	o	-	SEG12
	PB5	I/O	o	-	SEG13
	PB6	I/O	o	-	SEG14
	PB7	I/O	o	-	SEG15
端口 C					
	PC0	I/O	o	-	SEG16
	PC1	I/O	o	-	SEG17
	PC2	I/O	o	-	SEG18
	PC3	I/O	o	-	SEG19
	PC4	I/O	o	-	SEG20
	PC5	I/O	o	-	SEG21
	PC6	I/O	o	-	SEG22
	PC7	I/O	o	-	SEG23
端口 D					

表 8-1 端口功能列表(端口A ~ 端口K)

端口	引脚名称	输入/输出	施密特输入	噪声滤波器	功能引脚
	PD0	I/O	o	-	SEG24
	PD1	I/O	o	-	SEG25
	PD2	I/O	o	-	SEG26
	PD3	I/O	o	-	SEG27
	PD4	I/O	o	-	SEG28
	PD5	I/O	o	-	SEG29
	PD6	I/O	o	-	SEG30
	PD7	I/O	o	-	SEG31
端口E					
	PE0	I/O	o	-	SEG32
	PE1	I/O	o	-	SEG33
	PE2	I/O	o	-	SEG34/T16A6OUT
	PE3	I/O	o	-	SEG35/SCLK31/ $\overline{\text{CTS31}}$
	PE4	I/O	o	-	SEG36/RXD31
	PE5	I/O	o	-	SEG37/TXD31
	PE6	I/O	o	-	SEG38/SWCLK
	PE7	I/O	o	-	SEG39/SWDIO
端口F					
	PF0	I/O	o	-	AIN0
	PF1	I/O	o	o	AIN1/INT0
端口G					
	PG0	I/O	o	o	TB0OUT/INT2
端口H					
	PH0	I/O	o	-	TXD0/IROUT0
	PH1	I/O	o	-	RXD0
	PH2	I/O	o	-	SCLK0/ $\overline{\text{CTS0}}$ /T16A0OUT
	PH3	I/O	o	-	TXD1/IROUT1
	PH4	I/O	o	-	RXD1
	PH5	I/O	o	-	SCLK1/ $\overline{\text{CTS1}}$ /T16A1OUT
端口I					
	PI0	I/O	o	-	TXD2/IROUT2
	PI1	I/O	o	-	RXD2
	PI2	I/O	o	-	SCLK2/ $\overline{\text{CTS2}}$ /T16A2OUT
	PI3	I/O	o	-	TB0IN
	PI4	I/O	o	-	TXD30
	PI5	I/O	o	-	RXD30
	PI6	I/O	o	-	SCLK30/ $\overline{\text{CTS30}}$ /T16A5OUT
端口J					
	PJ0	I/O	o	-	SDA0/SO0
	PJ1	I/O	o	-	SCL0/SI0
	PJ2	I/O	o	o	SCK0/INT1
	PJ3	I/O	o	-	RTCOUT
	PJ4	I/O	o	-	T16A3OUT/SCOUT
	PJ5	I	o	-	TB1IN/XTCLKIN
端口K					

表 8-1 端口功能列表(端口A ~ 端口K)

端口	引脚名称	输入/输出	施密特输入	噪声滤波器	功能引脚
	PK0	I/O	o	o	INT3
	PK1	I/O	o	-	TB1OUT

注：噪声滤波器在典型条件下的噪声消除宽度约为 30 ns。

8.2 端口寄存器一般说明

使用端口寄存器时，必须设置以下寄存器。

所有寄存器均为 32-位。配置的差异取决于端口位数目和功能分配。

在以下检验中，对 8 位端口进行了说明。有关端口的配置和各端口中的初始值，请参看各端口的所在章节。

注：“x”表示端口的名称，而“n”表示以下说明中给出的功能编号。

8.2.1 PxDATA：端口 x 数据寄存器

寄存器可读取/写入端口数据。

	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	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
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	Px7-Px0	R/W	端口x 数据寄存器

8.2.2 PxCR：端口 x 输出控制寄存器

该寄存器可控制输出。

用 PxIE 寄存器启用 / 禁用输入

	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	Px7C	Px6C	Px5C	Px4C	Px3C	Px2C	Px1C	Px0C
复位后	0	0	0	0	0	0	1	0

位	比特符号	类型	功能
31-8	-	R	读作 "0"。
7-0	Px7C-Px0C	R/W	输出 0: 禁用 1: 启用

8.2.3 PxFRn: 端口 x 功能寄存器 n

该寄存器可设置功能。

可通过设置"1"启用已分配的功能。该寄存器用将个功能分配至端口。如具备若干功能，则仅一种功能可被启用。

	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	Px7Fn	Px6Fn	Px5Fn	Px4Fn	Px3Fn	Px2Fn	Px1Fn	Px0Fn
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	Px7Fn-Px0Fn	R/W	0: PORT 1: 功能

8.2.4 PxOD : 端口 x 开漏控制寄存器

该寄存器可控制可编程漏极输出。

可用 PxOD 设置可编程漏极输出。在输出数据为"1"时，输出缓冲器即被禁用，并变为准开漏输出。

	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	Px7OD	Px6OD	Px5OD	Px4OD	Px3OD	Px2OD	Px1OD	Px0OD
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	Px7OD-Px0OD	R/W	0: 推-拉输出 1: 开漏输出

8.2.5 PxPUP : 端口 x 上拉控制寄存器

该寄存器可控制可编程上拉。

	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	Px7UP	Px6UP	Px5UP	Px4UP	Px3UP	Px2UP	Px1UP	Px0UP
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	Px7UP-Px0UP	R/W	上拉 0: 禁用 1: 启用

8.2.6 PxPDN：端口 x 下拉控制寄存器

寄存器可控制可编程下拉。

	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	Px7DN	Px6DN	Px5DN	Px4DN	Px3DN	Px2DN	Px1DN	Px0DN
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	Px7DN-Px0DN	R/W	下拉 0: 禁用 1: 启用

8.2.7 PxIE：端口 x 输入控制寄存器

寄存器可控制输入。

	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	Px7IE	Px6IE	Px5IE	Px4IE	Px3IE	Px2IE	Px1IE	Px0IE
复位后	0	0	0	0	0	0	1	1

位	比特符号	类型	功能
31-8	-	R	读作 "0"。
7-0	Px7IE-Px0IE	R/W	输入 0: 禁用 1: 启用



8.3 寄存器列表

端口	基址
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	0x440C_0600
端口H	0x400C_0700
端口I	0x400C_0800
端口J	0x400C_0900
端口K	0x400C_0A00

寄存器名称	地址 (基+)	端口A	端口B	端口C	端口D	端口E	端口F
数据寄存器	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA	PFDATA
输出控制寄存器	0x0004	PACR	PBCR	PCCR	PDCR	PECR	PFCR
功能寄存器 1	0x0008	PAFR1	PBFR1	PCFR1	PDFR1	PEFR1	PFFR1
功能寄存器 2	0x000C	-	-	-	-	PEFR2	-
功能寄存器 3	0x0010	-	-	-	-	PEFR3	-
开漏控制寄存器	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD	PFOD
上拉控制寄存器	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP	PFPUP
下拉控制寄存器	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN	PFPDN
输入控制寄存器	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE	PFIE

寄存器名称	地址 (基+)	端口G	端口H	端口I	端口J	端口K
数据寄存器	0x0000	PGDATA	PHDATA	PIDATA	PJDATA	PKDATA
输出控制寄存器	0x0004	PGCR	PHCR	PICR	PJCR	PKCR
功能寄存器 1	0x0008	PGFR1	PHFR1	PIFR1	PJFR1	PKFR1
功能寄存器 2	0x000C	PGFR2	PHFR2	PIFR2	PJFR2	-
功能寄存器 3	0x0010	-	PHFR3	PIFR3	-	-
开漏控制寄存器	0x0028	PGOD	PHOD	PIOD	PJOD	PKOD
上拉控制寄存器	0x002C	PGPUP	PHPUP	PIPUP	PJPUP	PKPUP
下拉控制寄存器	0x0030	PGPDN	PHPDN	PIPDN	PJPDN	PKPDN
输入控制寄存器	0x0038	PGIE	PHIE	PIIE	PJIE	PKIE

8.4 功能详细

本章说明寄存器的配置，初始值和通过功能寄存器分配的功能。寄存器的位 31 ~ 8，以及带有阴影线的位被读作"0"。向其写入不会产生影响。

8.4.1 端口 A

	7	6	5	4	3	2	1	0
PADATA	0	0	0	0	0	0	0	0
PAIE	0	0	0	0	0	0	0	0
PACR	0	0	0	0	0	0	0	0
PAPUP	0	0	0	0	0	0	0	0
PAPDN	0	0	0	0	0	0	0	0
PAOD	0	0	0	0	0	0	0	0
PAFR1	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
	0	0	0	0	0	0	0	0

注：在端口A用作LCD节段输出时，应将"1"设置到PAFR1，并将其它寄存器清除为"0"。

8.4.2 端口 B

	7	6	5	4	3	2	1	0
PBDATA	0	0	0	0	0	0	0	0
PBIE	0	0	0	0	0	0	0	0
PBCR	0	0	0	0	0	0	0	0
PBPUP	0	0	0	0	0	0	0	0
PBPDN	0	0	0	0	0	0	0	0
PBOD	0	0	0	0	0	0	0	0
PBFR1	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
	0	0	0	0	0	0	0	0

注：在端口B用作LCD节段输出时，应将"1"设置到PBFR1，并将其它寄存器清除为"0"。

8.4.3 端口 C

	7	6	5	4	3	2	1	0
PCDATA	0	0	0	0	0	0	0	0
PCIE	0	0	0	0	0	0	0	0
PCCR	0	0	0	0	0	0	0	0
PCPUP	0	0	0	0	0	0	0	0
PCPDN	0	0	0	0	0	0	0	0
PCOD	0	0	0	0	0	0	0	0
PCFR1	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
	0	0	0	0	0	0	0	0

注：在端口C用作LCD节段输出时，可将"1"设置到PCFR1，并将其它寄存器清除为"0"。

8.4.4 端口 D

	7	6	5	4	3	2	1	0
PDDATA	0	0	0	0	0	0	0	0
PDIE	0	0	0	0	0	0	0	0
PDCR	0	0	0	0	0	0	0	0
PDPUP	0	0	0	0	0	0	0	0
PDPDN	0	0	0	0	0	0	0	0
PDOD	0	0	0	0	0	0	0	0
PDFR1	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
	0	0	0	0	0	0	0	0

注：在端口D用作LCD节段输出时，可将"1"设置到PDFR1，并将其它寄存器清除为"0"。

8.4.5 端口 E

	7	6	5	4	3	2	1	0
PEDATA	0	0	0	0	0	0	0	0
PEIE	1	1	0	0	0	0	1(注 3)	1(注 3)
PECR	1	0	0	0	0	0	1(注 3)	0
PEPUP	1	0	0	0	0	0	1(注 3)	0
PEPDN	0	1	0	0	0	0	0	1(注 3)
PEOD	0	0	0	0	0	0	0	0
PEFR1	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
	0	0	0	0	0	0	0	0
PEFR2	SWDIO	SWCLK	TXD31	RXD31	SCLK31	T16A6OUT	-	-
	1	1	0	0	0	0	1(注 2)	1(注 2)
PEFR3					$\overline{\text{CTS31}}$			
					0			

注 1: 在端口E用作LCD节段输出时, 可将"1"设置到PEFR1, 并将其它寄存器清除为"0".

注 2: 写作 "0".

注 3: 必要时可改变设置。

8.4.6 端口 F

在本端口用作模拟输入时, 所有寄存器的值均即被设置为初始值。

	7	6	5	4	3	2	1	0
PFDATA							0	0
PFIE							0	0
PFCR							0	0
PFPUP							0	0
PFPDN							0	0
PFOD							0	0
PFFR1							INT0	
							0	

注: 在除STOP模式以外的各模式下, 如果PxIE里的输入被启用, 则中断输入即被启用, 且不考虑PxFRn寄存器设置。当给装置编程时, 应确保禁用未使用的中断。

8.4.7 端口 G

	7	6	5	4	3	2	1	0
PGDATA								0
PGIE								0
PGCR								0
PGPUP								0
PGPDN								0
PGOD								0
PGFR1								TB0OUT
								0
PGFR2								INT2
								0

注：在除STOP模式以外的各模式下，如果PxIE中的输入被启用，则中断输入即被启用，且不考虑PxFRn寄存器设置。当给装置编程时，应确保禁用未使用的中断。

8.4.8 端口 H

	7	6	5	4	3	2	1	0
PHDATA			0	0	0	0	0	0
PHIE			0	0	0	0	0	0
PHCR			0	0	0	0	0	0
PHPUP			0	0	0	0	0	0
PHPDN			0	0	0	0	0	0
PHOD			0	0	0	0	0	0
PHFR1			SCLK1	RXD1	TXD1	SCLK0	RXD0	TXD0
			0	0	0	0	0	0
PHFR2			$\overline{\text{CTS1}}$		IROUT1	$\overline{\text{CTS0}}$		IROUT0
			0		0	0		0
PHFR3			T16A1OUT			T16A0OUT		
			0			0		

注： $\overline{\text{DBGEN}}$ 功能可共享PH3。在 $\overline{\text{RESET}}$ 引脚为低时，启用后可被输入及上拉。

8.4.9 端口 I

	7	6	5	4	3	2	1	0
PIDATA		0	0	0	0	0	0	0
PIIE		0	0	0	0	0	0	0
PICR		0	0	0	0	0	0	0
PIPUP		0	0	0	0	0	0	0
PIPDN		0	0	0	0	0	0	0
PIOD		0	0	0	0	0	0	0
PIFR1		SCLK30	RXD30	TXD30	TB0IN	SCLK2	RXD2	TXD2
		0	0	0	0	0	0	0
PIFR2		$\overline{\text{CTS30}}$				$\overline{\text{CTS2}}$		IROUT2
		0				0		0
PIFR3		T16A5OUT				T16A2OUT		
		0				0		

注 1: 在 $\overline{\text{RESET}}$ 引脚为低时, PIO 启用后可被输入及上拉。

注 2: PI5 和 PI6 仅在输入启用时允许 5 V 输入。

注意当用作开漏输出时, 这些引脚不能上拉到电源电压之上。

8.4.10 端口 J

	7	6	5	4	3	2	1	0
PJDATA			0	0	0	0	0	0
PJIE			0	0	0	0	0	0
PJCR				0	0	0	0	0
PJPUP				0	0	0	0	0
PJPDN				0	0	0	0	0
PJOD				0	0	0	0	0
PJFR1			TB1IN	T16A3OUT	RTCOUT	SCK0	SCL0/SI0	SDA0/SO0
			0	0	0	0	0	0
PJFR2			XTCLKIN	SCOUT		INT1		
			0	0		0		

注 1: $\overline{\text{BOOT}}$ 功能可共享 PJ0 。在 $\overline{\text{RESET}}$ 引脚为“低”时, 启用后可被输入及上拉。

注 2: PJ5 仅用作输入端口。

注 3: 在除 STOP 模式以外的各模式下, 如果输入在 PxIE 中被启用, 则中断输入即被启用, 且不考虑 PxFRn 寄存器设置。当给装置编程时, 应确保禁用未使用的中断。

注 4: PJ2 仅在输入被启用时允许 5V 输入。

注意当用作开漏输出时, 这些引脚不能上拉到电源电压之上。

8.4.11 端口 K

在泄放电阻器被连接到端口时，所有寄存器的值均即设置为初始值。

	7	6	5	4	3	2	1	0
PKDATA							0	0
PKIE							0	0
PKCR							0	0
PKPUP							0	0
PKPDN							0	0
PKOD							0	0
PKFR1							TB1OUT	INT3
							0	0

注：在除STOP模式以外的各模式下，如果PxIE中的输入被启用，则中断输入即被启用，且不考虑PxFRn寄存器设置。当给装置编程时，应确保禁用未使用的中断。

8.5 端口方块图

8.5.1 端口类型

端口分类如下。有关各端口类型所对应的方块图，请参照后续几页。图中的虚线框表示“端口方块图”中所述等效电路的一部分。

各引脚所用的端口类型见第 8.6 节所述。

表 8-2 功能列表

类型	I/O	功能				上拉	下拉	模拟
		启用信号 存在/不存在		在 STOP 模式下运行				
		输入	输出	输入	输出			
FT1	I/O	-	-	-	-	R	R	-
FT2	I/O	-	o	o	o	EnR	EnR	-
FT3	I/O	-	o	-	-	R	R	-
FT4	输入(int)	-	-	o	-	R	R	-
FT5	输入	-	-	-	-	R	R	o
FT6	I/O	o	o	-	o	R	R	-
FT7	输入	-	-	o	-	R	R	-
FT8	I/O	-	-	-	o	R	R	-
FT9	I/O	-	o	-	o	R	R	-
FT10	输出(LCD)	-	-	-	o	R	R	-

int: 中断输入

-: 不存在/已被禁用

o: 存在/已被启用

R: 复位期间强制禁用

EnR: 复位期间强制启用

8.5.2 类型 FT1

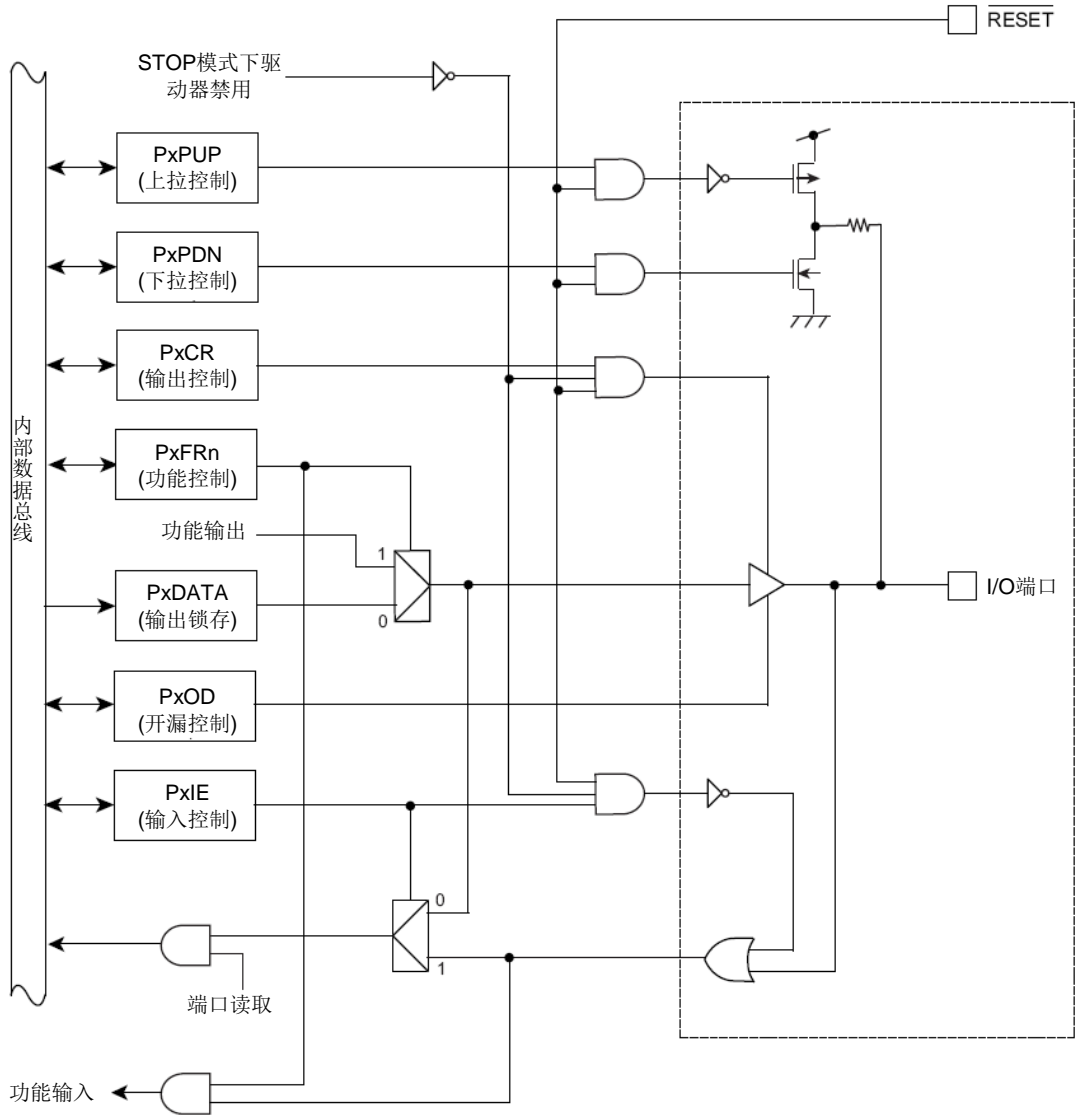


图 8-1 端口类型FT1

8.5.3 类型 FT2

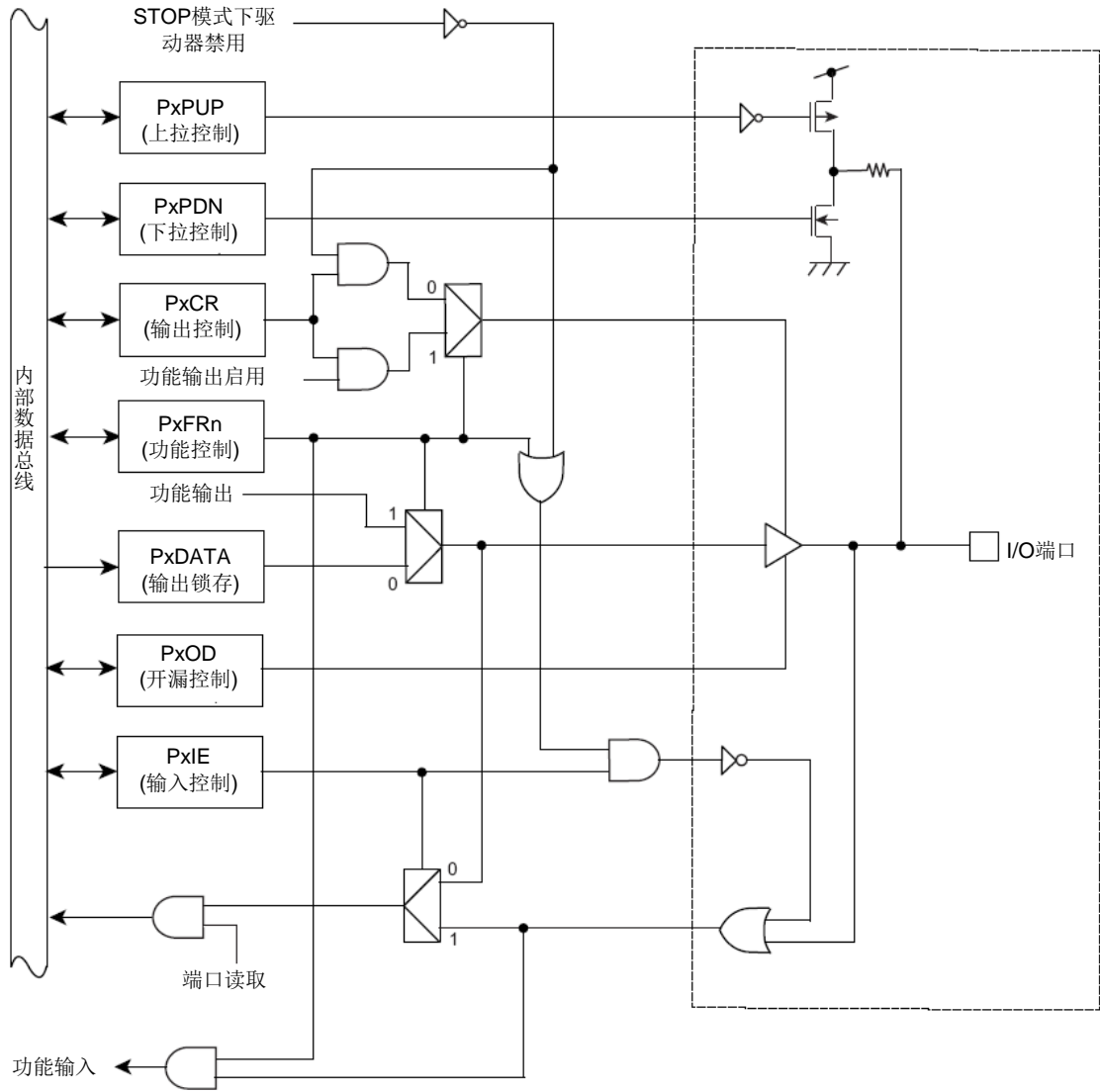


图 8-2 端口类型FT2

8.5.4 类型 FT3

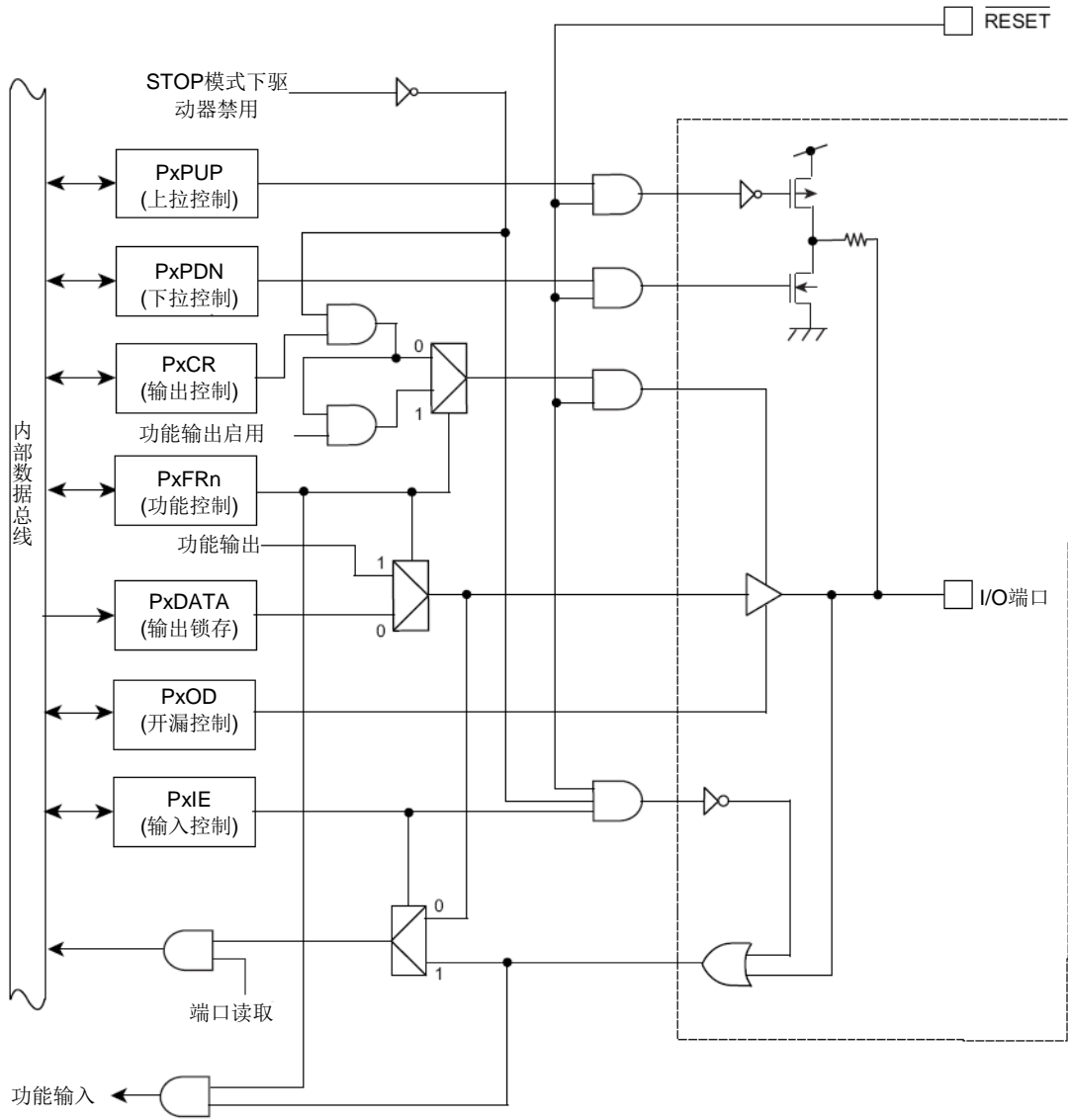


图 8-3 端口类型FT3

8.5.5 类型 FT4

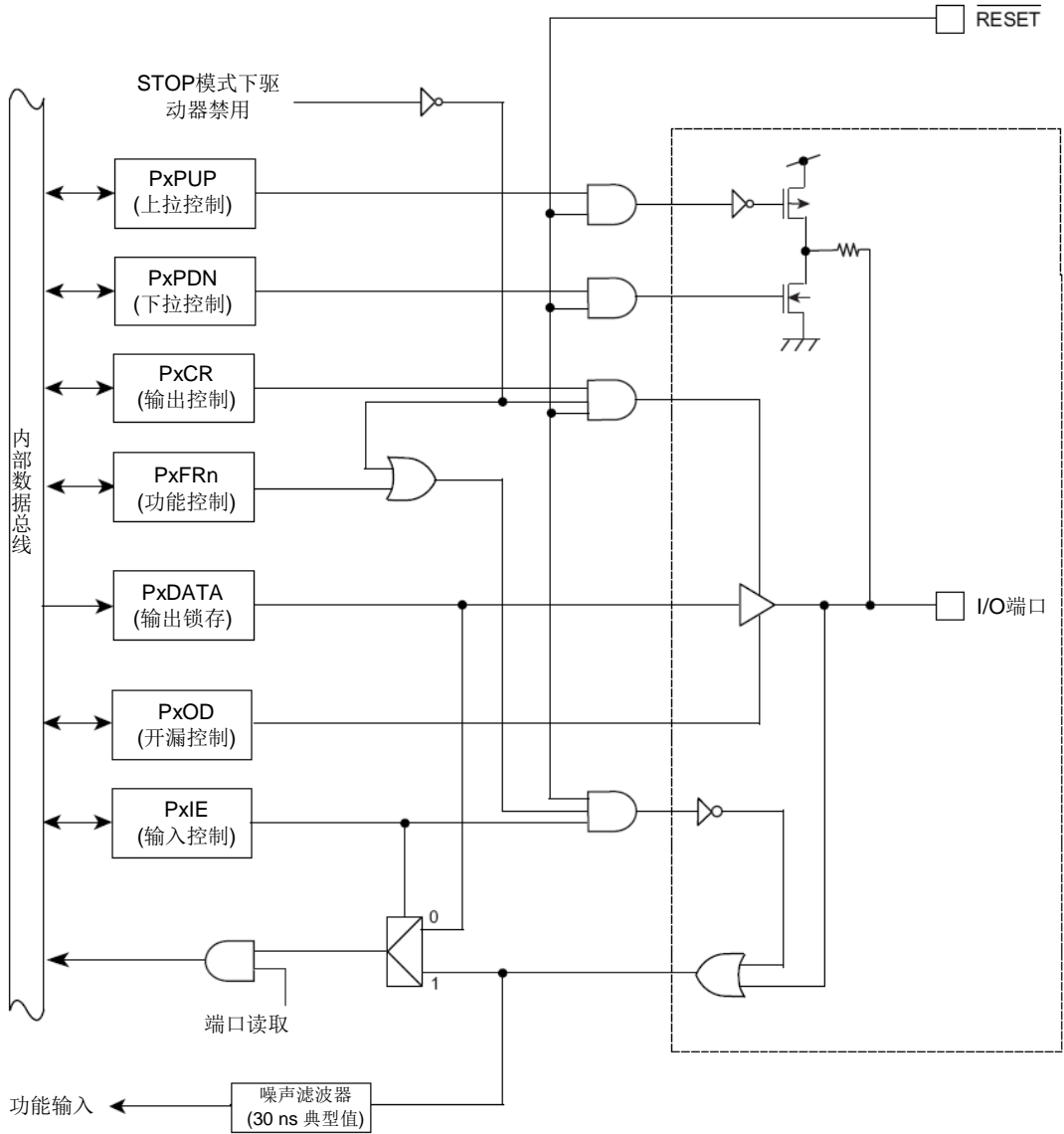


图 8-4 端口类型FT4

8.5.6 类型 FT5

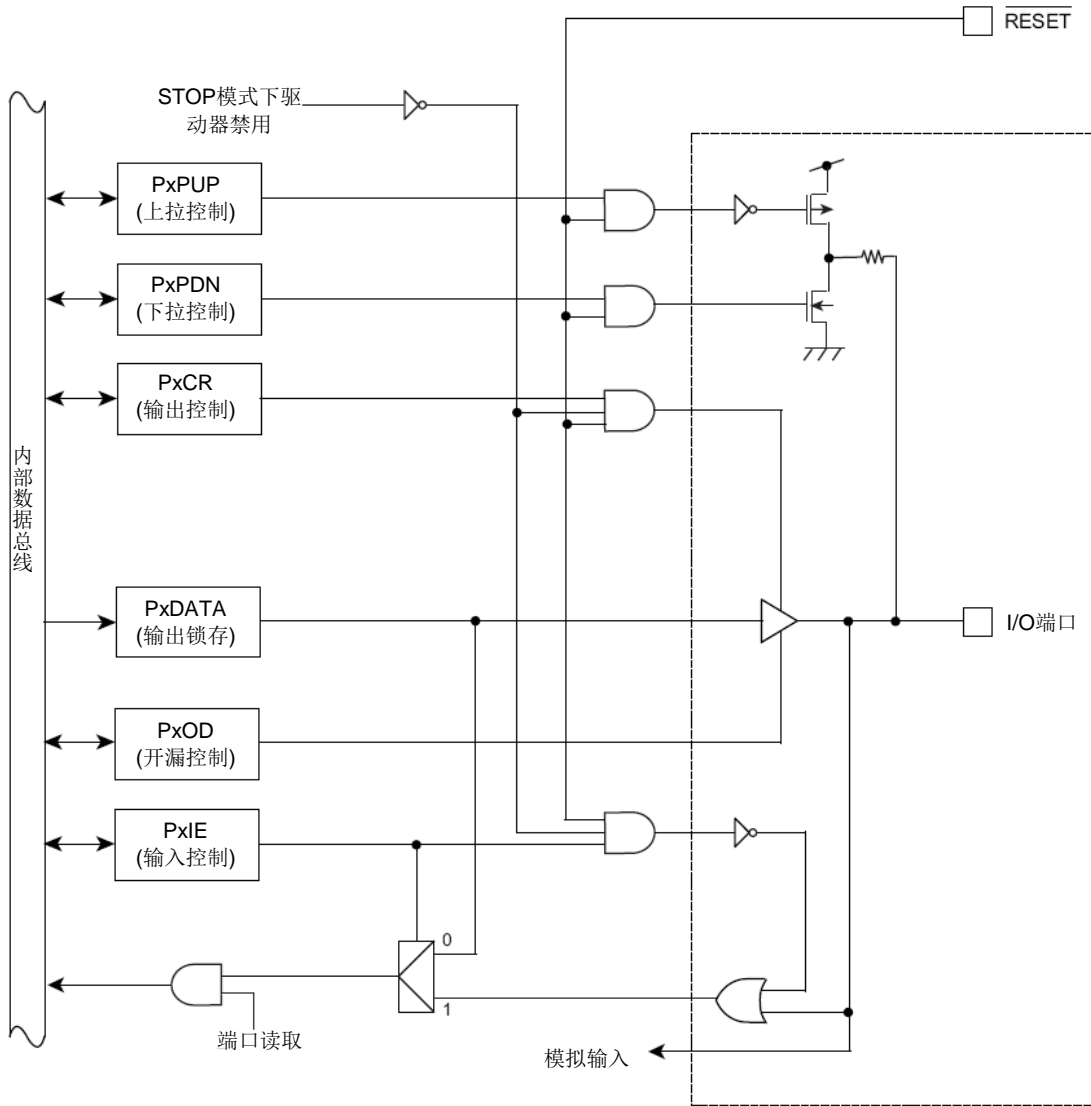


图 8-5 端口类型FT5

8.5.7 类型 FT6

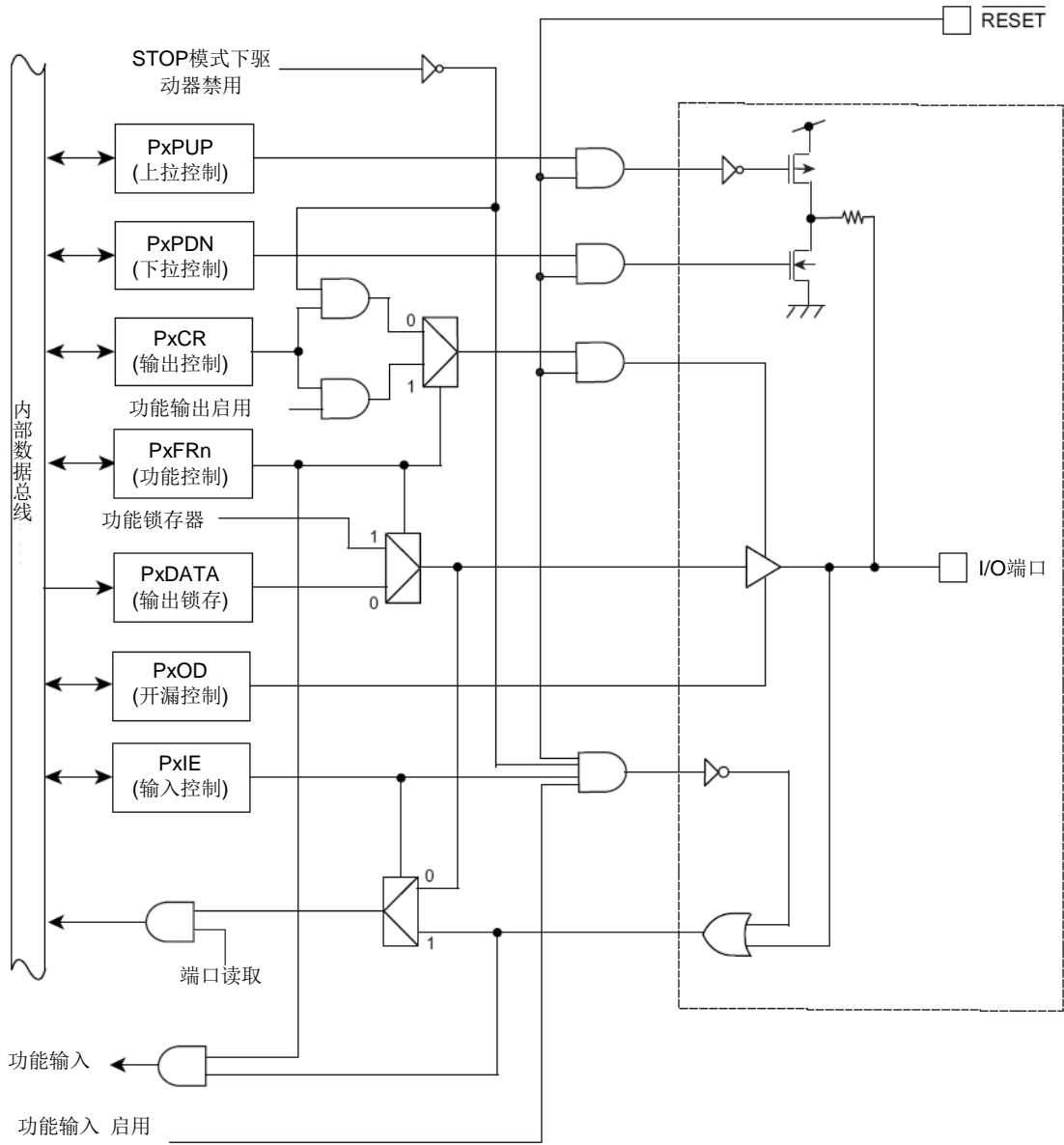


图 8-6 端口类型FT6

8.5.8 类型 FT7

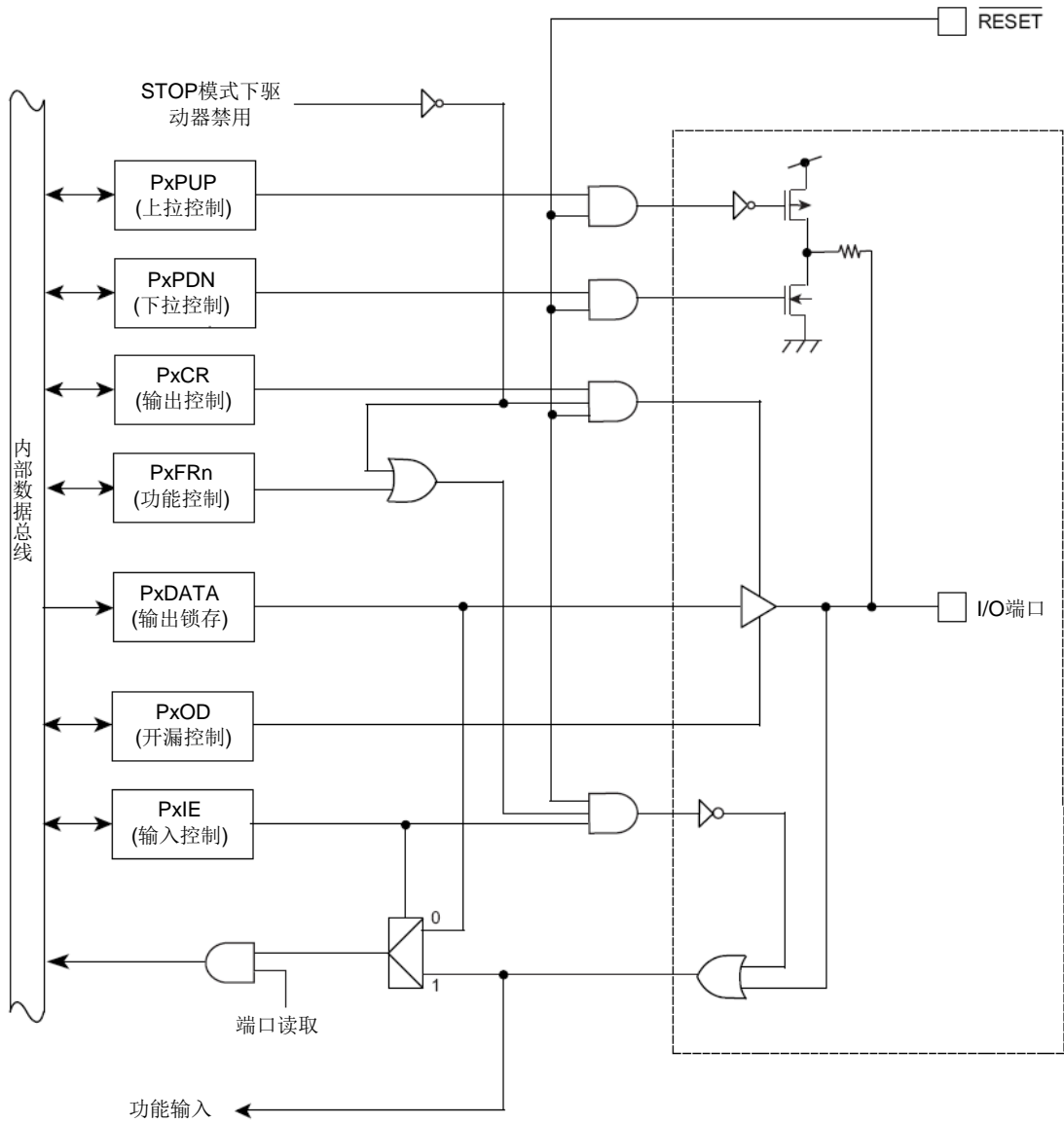


图 8-7 端口类型FT7

8.5.9 类型 FT8

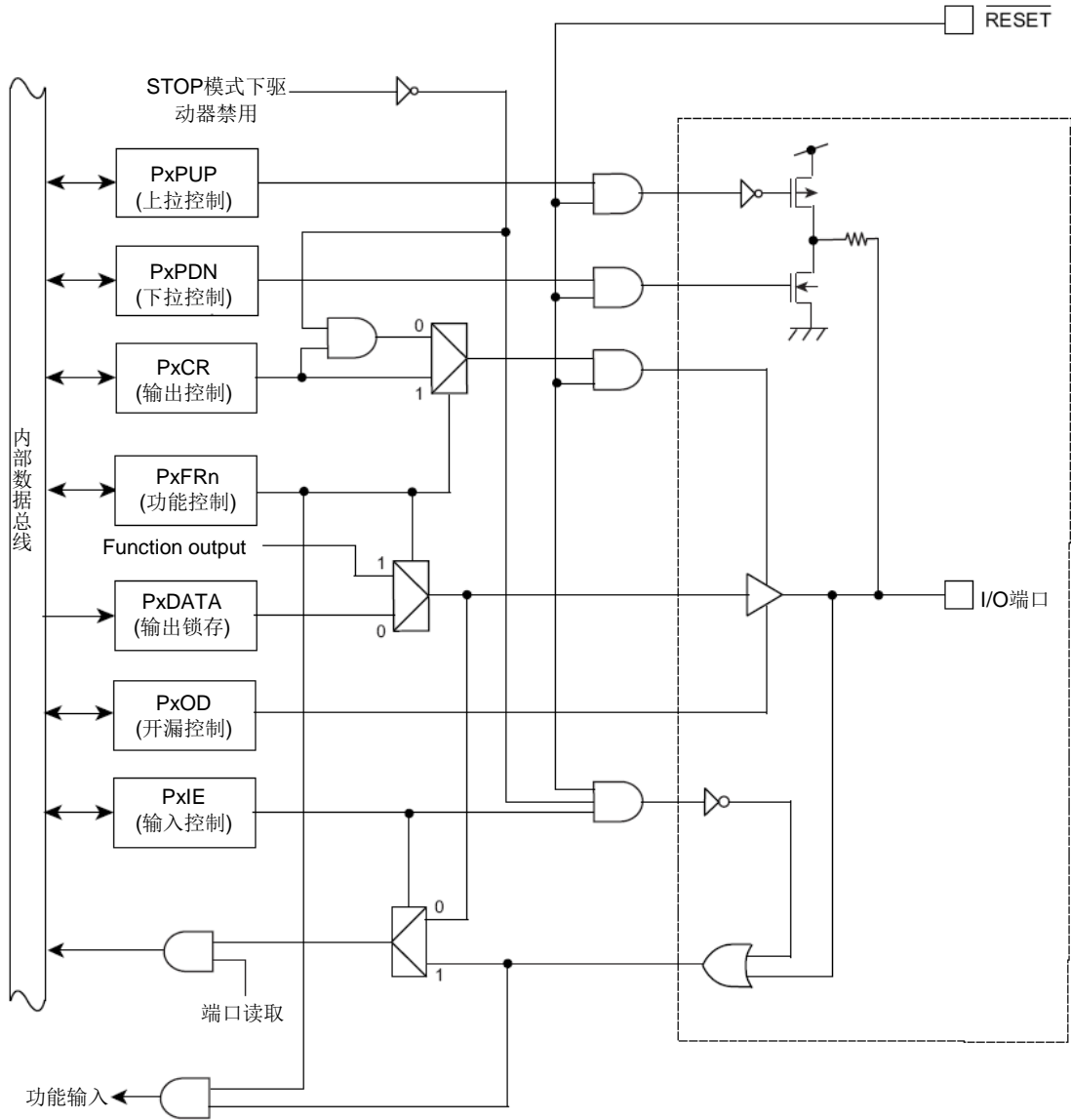


图 8-8 端口类型FT8

8.5.10 类型 FT9

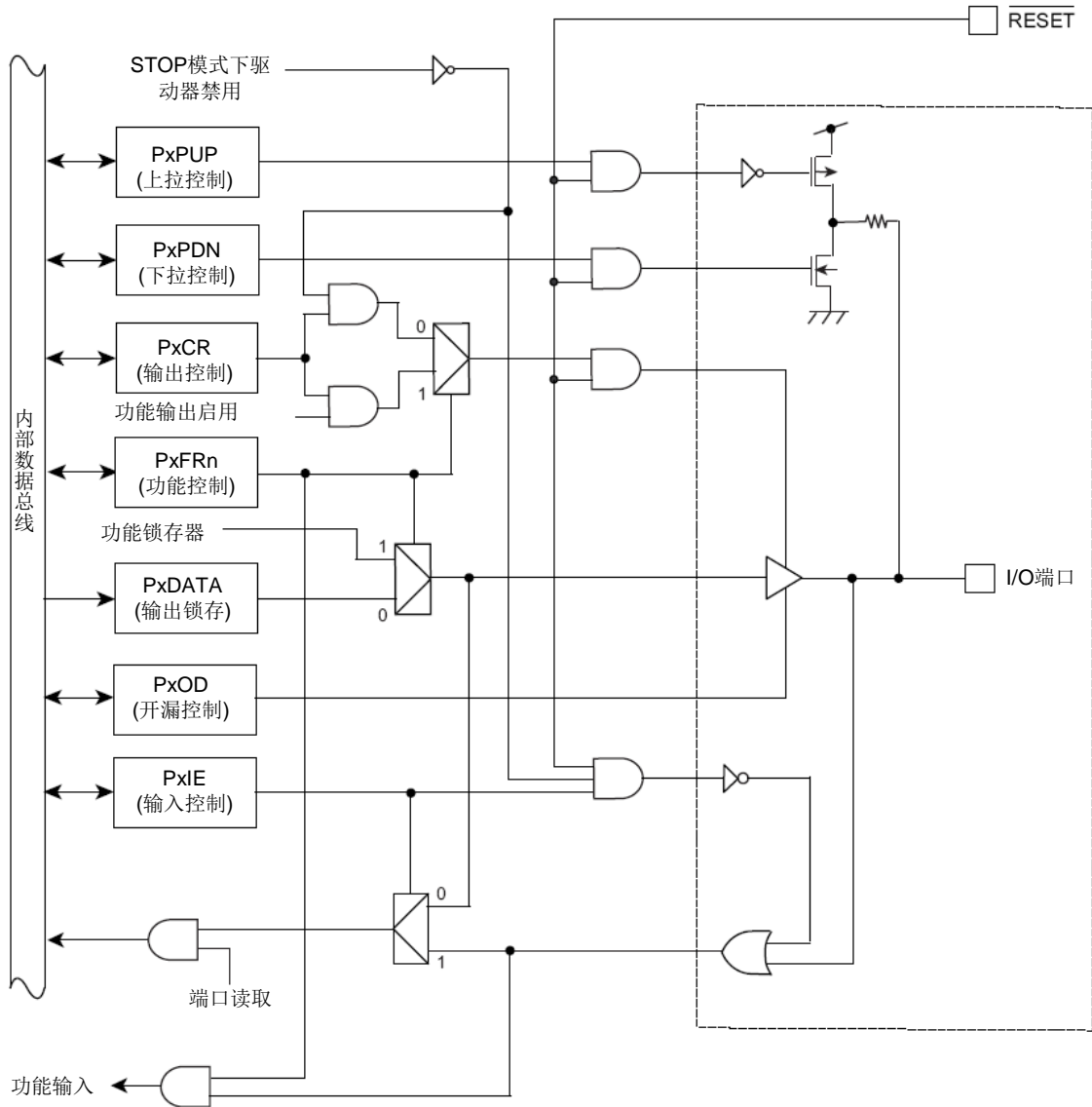


图 8-9 端口类型FT9

8.5.11 类型 FT10

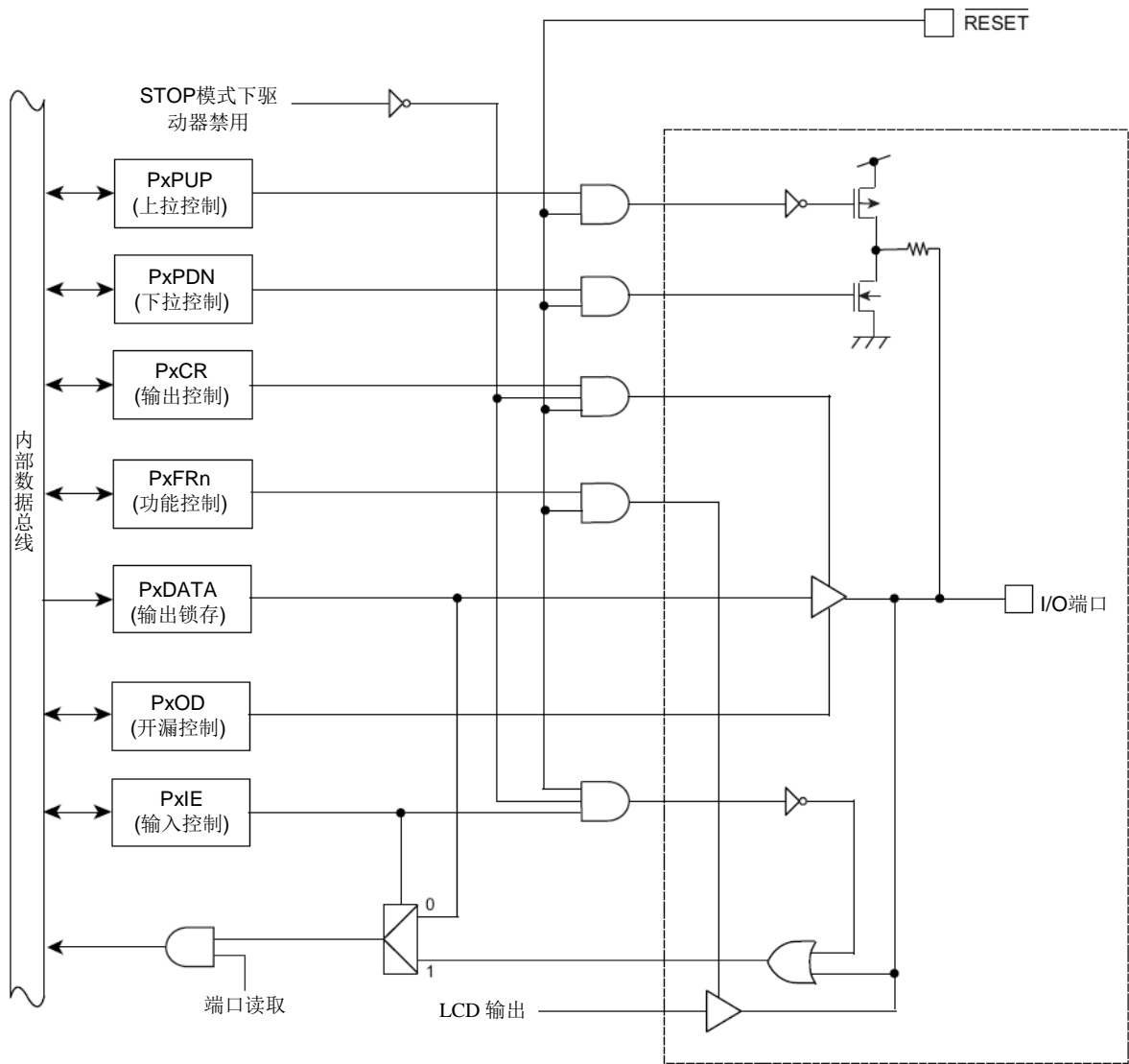


图 8-10 端口类型FT10

8.6 附录(端口设置列表)

下表给出了各引脚中的端口寄存器设置。

以下寄存器名称中的数字"0"和 "1"表示设置值，"x"则表示任选。

8.6.1 I/O 端口的设置

在 I/O 端口被用作输入端口或输出端口时，应按以下所述设置其寄存器：

引脚名称	端口类型	功能	复位后	Px CR	Px FRn	Px OD	Px PUP	Px PDN	Px IE
Pxn	-	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0

8.6.2 输入专用端口的设置

在使用输入专用端口时，按以下所述设置其寄存器：

引脚名称	端口类型	功能	复位后	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	输入端口		0	x	x	x	1

8.6.3 输出专用端口的设置

在使用输出专用端口时，应按以下所述设置其寄存器：

引脚名称	端口类型	功能	复位后	Px CR	Px FRn	Px OD	Px PUP	Px PDN
Pxn	-	输出端口		1	0	x	x	x
		输出端口 (Hi-Z 输出)		0	0	x	x	x

8.6.4 外设 I/O 端口的设置

本节对端口被用于外设功能时的设置进行说明。

在几乎所有的引脚中，端口寄存器的初始端口状态均全为"0"，并被禁止在复位后输入/输出。某些引脚会在复位后被指定用于某项功能。在这种情况下，会在下表的"复位后"一列中对符号"o"进行说明。

PxFRn 列表示需设置的功能寄存器。

下表所列各位的阴影区表示"0"的读取和写入无意义。

8.6.4.1 端口 A 设置

引脚名称	端口类型	功能	复位后	PA CR	PA FRn	PA OD	PA PUP	PA PDN	PA IE
PA0	FT10	SEG0 (输出)		0	PA0 FR1	0	0	0	0
PA1	FT10	SEG1 (输出)		0	PA1 FR1	0	0	0	0
PA2	FT10	SEG2 (输出)		0	PA2 FR1	0	0	0	0
PA3	FT10	SEG3 (输出)		0	PA3 FR1	0	0	0	0
PA4	FT10	SEG4 (输出)		0	PA4 FR1	0	0	0	0
PA5	FT10	SEG5 (输出)		0	PA5 FR1	0	0	0	0
PA6	FT10	SEG6 (输出)		0	PA6 FR1	0	0	0	0
PA7	FT10	SEG7 (输出)		0	PA7 FR1	0	0	0	0

8.6.4.2 端口 B 设置

引脚名称	端口类型	功能	复位后	PB CR	PB FRn	PB OD	PB PUP	PB PDN	PB IE
PB0	FT10	SEG8 (输出)		0	PB0 FR1	0	0	0	0
PB1	FT10	SEG9 (输出)		0	PB1 FR1	0	0	0	0
PB2	FT10	SEG10 (输出)		0	PB2 FR1	0	0	0	0
PB3	FT10	SEG11 (输出)		0	PB3 FR1	0	0	0	0
PB4	FT10	SEG12 (输出)		0	PB4 FR1	0	0	0	0
PB5	FT10	SEG13 (输出)		0	PB5 FR1	0	0	0	0
PB6	FT10	SEG14 (输出)		0	PB6 FR1	0	0	0	0
PB7	FT10	SEG15 (输出)		0	PB7 FR1	0	0	0	0

8.6.4.3 端口 C 设置

引脚名称	端口类型	功能	复位后	PC CR	PC FRn	PC OD	PC PUP	PC PDN	PC IE
PC0	FT10	SEG16 (输出)		0	PC0 FR1	0	0	0	0
PC1	FT10	SEG17 (输出)		0	PC1 FR1	0	0	0	0
PC2	FT10	SEG18 (输出)		0	PC2 FR1	0	0	0	0
PC3	FT10	SEG19 (输出)		0	PC3 FR1	0	0	0	0
PC4	FT10	SEG20 (输出)		0	PC4 FR1	0	0	0	0
PC5	FT10	SEG21 (输出)		0	PC5 FR1	0	0	0	0
PC6	FT10	SEG22 (输出)		0	PC6 FR1	0	0	0	0
PC7	FT10	SEG23 (输出)		0	PC7 FR1	0	0	0	0

8.6.4.4 端口 D 设置

引脚名称	端口类型	功能	复位后	PD DR	PD FRn	PD OD	PD PUP	PD PDN	PD IE
PD0	FT10	SEG24 (输出)		0	PD0 FR1	0	0	0	0
PD1	FT10	SEG25 (输出)		0	PD1 FR1	0	0	0	0
PD2	FT10	SEG26 (输出)		0	PD2 FR1	0	0	0	0
PD3	FT10	SEG27 (输出)		0	PD3 FR1	0	0	0	0
PD4	FT10	SEG28 (输出)		0	PD4 FR1	0	0	0	0
PD5	FT10	SEG29 (输出)		0	PD5 FR1	0	0	0	0
PD6	FT10	SEG30 (输出)		0	PD6 FR1	0	0	0	0
PD7	FT10	SEG31 (输出)		0	PD7 FR1	0	0	0	0

8.6.4.5 端口 E 设置

引脚名称	端口类型	功能	复位后	PE CR	PE FRn	PE OD	PE PUP	PE PDN	PE IE
PE0	FT10	SEG32 (输出)		0	PE0 FR1	0	0	0	0
PE1	FT10	SEG33 (输出)		0	PE1 FR1	0	0	0	0
PE2	FT10	SEG34 (输出)		0	PE2 FR1	0	0	0	0
	FT1	T16A6OUT (输出)		1	PE2 FR2	x	x	x	0
PE3	FT10	SEG35 (输出)		0	PE3 FR1	0	0	0	0
	FT1	SCLK31 (输入)		0	PE3 FR2	x	x	x	1
		SCLK31 (输出)		1	PE3 FR2	x	x	x	0
	FT1	$\overline{\text{CTS31}}$ (输出)		1	PE3 FR3	x	x	x	0
PE4	FT10	SEG36 (输出)		0	PE4 FR1	0	0	0	0
	FT1	RXD31 (输入)		0	PE4 FR2	x	x	x	1
PE5	FT10	SEG37 (输出)		0	PE5 FR1	0	0	0	0
	FT1	TXD31 (输入)		1	PE5 FR2	x	x	x	0
PE6	FT10	SEG38 (输出)		0	PE6 FR1	0	0	0	0
	FT2	SWCLK(输入)	o	0	PE6 FR2	0	0	1	1
PE7	FT10	SEG39 (输出)		0	PE7 FR1	0	0	0	0
	FT2	SWDIO (I/O)	o	1	PE7 FR2	0	1	0	1

8.6.4.6 端口 F 设置

引脚名称	端口类型	功能	复位后	PF CR	PF FRn	PF OD	PF PUP	PF PDN	PF IE
PF0	FT5	AIN0	0	0	0	0	0	0	0
PF1	FT5	AIN1	0	0	0	0	0	0	0
	FT4	INT0 (输入)		0	PF1 FR1	x	x	x	1

8.6.4.7 端口 G 设置

引脚名称	端口类型	功能	复位后	PG CR	PG GRn	PG OD	PG PUP	PG PDN	PG IE
PG0	FT1	TB0OUT(输出)		1	PG0 GR0	x	x	x	0
	FT4	INT2 (输入)		0	PG0 GR1	x	x	x	1

8.6.4.8 端口 H 设置

引脚名称	端口类型	功能	复位后	PH CR	PH FRn	PH OD	PH PUP	PH PDN	PH IE
PH0	FT1	TXD0 (输出)		1	PH0 FR1	x	x	x	0
	FT1	IROUT0 (输出)		1	PH0 FR2	x	x	x	0
PH1	FT1	RXD0 (输入)		0	PH1 FR1	x	x	x	1
PH2	FT1	SCLK0 (输入)		0	PH2 FR1	x	x	x	1
	FT1	SCLK0 (输出)		1	PH2 FR1	x	x	x	0
	FT1	$\overline{\text{CTS0}}$ (输出)		1	PH2 FR2	x	x	x	0
	FT1	T16A0OUT (输出)		1	PH2 FR3	x	x	x	0
PH3	FT1	TXD1 (输出)		1	PH3 FR1	x	x	x	0
	FT1	IROUT1 (输出)		1	PH3 FR2	x	x	x	0
PH4	FT1	RXD1(输入)		0	PH4 FR1	x	x	x	1
PH5	FT1	SCLK1(输入)		0	PH5 FR1	x	x	x	1
	FT1	SCLK1(输出)		1	PH5 FR1	x	x	x	0
	FT1	$\overline{\text{CTS1}}$ (输出)		1	PH5 FR2	x	x	x	0
	FT1	T16A1OUT (输出)		1	PH5 FR3	x	x	x	0

8.6.4.9 端口 I 设置

引脚名称	端口类型	功能	复位后	PI CR	PI FRn	PI OD	PI PUP	PI PDN	PI IE
PI0	FT1	TXD2(输出)		1	PI0 FR1	x	x	x	0
	FT1	IROUT2 (输出)		1	PI0 FR2	x	x	x	0
PI1	FT1	RXD2(输入)		0	PI1 FR1	x	x	x	1
PI2	FT1	SCLK2 (输入)		0	PI2 FR1	x	x	x	1
	FT1	SCLK2 (输出)		1	PI2 FR1	x	x	x	0
	FT1	$\overline{\text{CTS2}}$ (输出)		1	PI2 FR2	x	x	x	0
	FT1	T16A2OUT (输出)		1	PI2 FR3	x	x	x	0
PI3	FT1	TB0IN (输入)		0	PI3 FR1	x	x	x	1
PI4	FT1	TXD30 (输出)		1	PI4 FR1	x	x	x	0
PI5	FT1	RXD30 (输入)		0	PI5 FR1	x	x	x	1
PI6	FT1	SCLK30 (输入)		0	PI6 FR1	x	x	x	1
	FT1	SCLK30 (输出)		1	PI6 FR1	x	x	x	0
	FT1	$\overline{\text{CTS30}}$ (输出)		1	PI6 FR2	x	x	x	0
	FT1	T16A5OUT (输出)		1	PI6 FR3	x	x	x	0

8.6.4.10 端口 J 设置

引脚名称	端口类型	功能	复位后	PJ CR	PJ FRn	PJ OD	PJ PUP	PJ PDN	PJ IE
PJ0	FT1	SDA0(I/O)		1	PJ0 FR1	1	x	x	1
		SO0(输出)		1	PJ0 FR1	x	x	x	0
PJ1	FT1	SCL0(I/O)		1	PJ0 FR1	1	x	x	1
		SI0(输入)		0	PJ0 FR1	x	x	x	1
PJ2	FT1	SCK0(输入)		0	PJ2 FR1	x	x	x	1
		SCK0(输出)		1	PJ2 FR1	x	x	x	0
	FT4	INT1(输入)		0	PJ2 FR2	x	x	x	1
PJ3	FT1	RTCOUT (输出)		1	PJ3 FR1	x	x	x	0
PJ4	FT1	T16A3OUT (输出)		1	PJ4 FR1	x	x	x	0
	FT1	SCOUT (输出)		1	PJ4 FR2	x	x	x	0
PJ5	FT1	TB1IN (输入)			PJ5 FR1				1
	FT1	XTCLKIN (输入)			PJ5 FR2				1

8.6.4.11 端口 K 设置

引脚名称	端口类型	功能	复位后	PK CR	PK FRn	PK OD	PK PUP	PK PDN	PK IE
PK0	FT5	LV1 (输入)	o	0	0	0	0	0	0
	FT4	INT3 (输入)		0	PK0 FR1	x	x	x	1
PK1	FT5	LV2 (输入)	o	0	0	0	0	0	0
	FT1	TB1OUT(输出)		1	PK1 FR1	x	x	x	0

9. 16-位计时器/事件计数器(TMRB)

9.1 概要

TMRB具备以下所述的各工作模式。

- 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

捕获功能的使用，TMRB执行以下测量。

- 频率测量
- 脉冲宽度测量

在以下说明中，“x”表示通道编号。

9.2 方块图

TMRB由一个 16-位上升计数器，两个 16-位计时器寄存器(双缓冲)，两个 16-位捕获寄存器，两个比较器，一个捕获输入控制，一个计时器触发器及其相关控制电路组成。计时器工作模式及计时器触发器是由寄存器控制的。

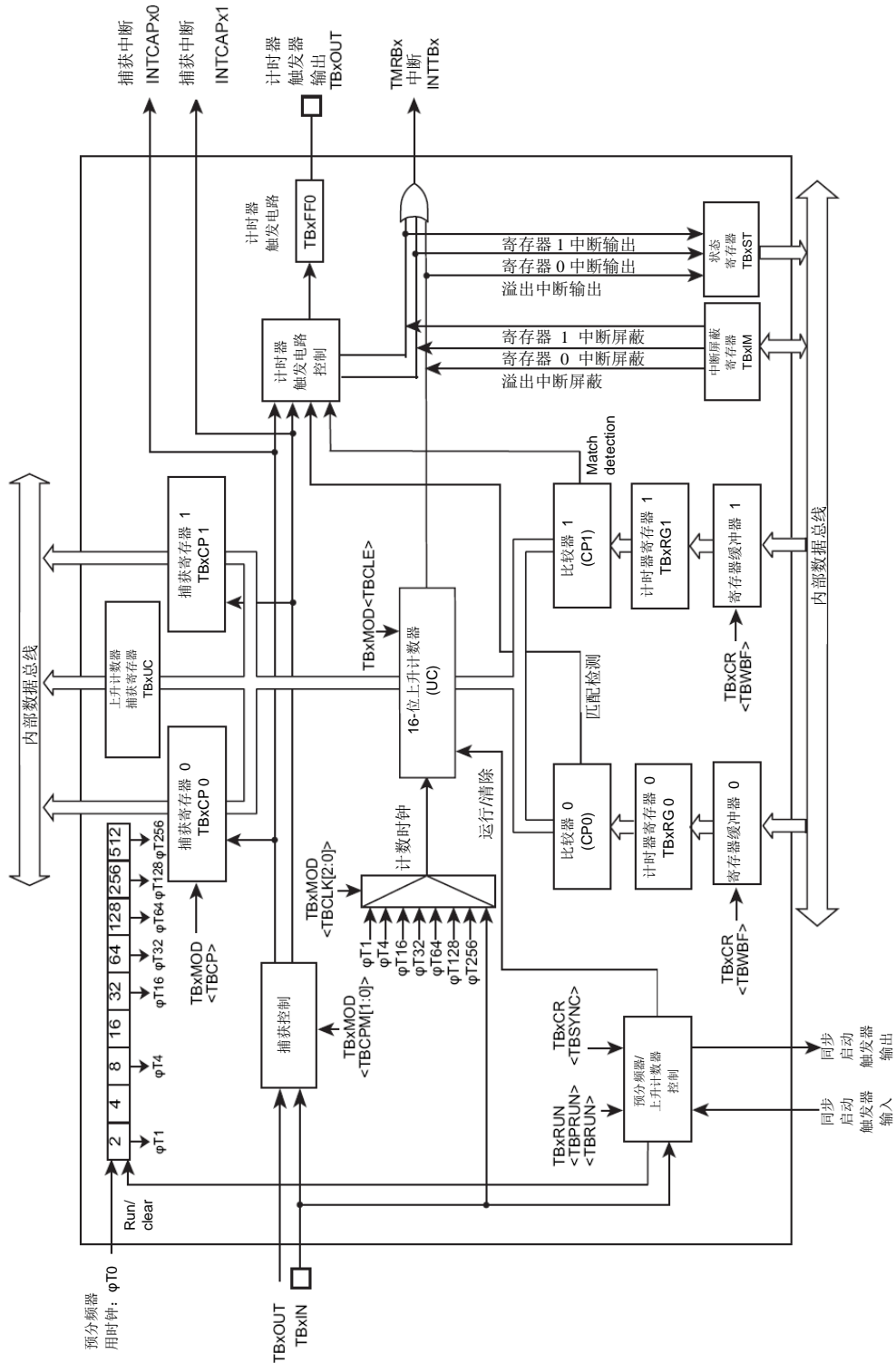


图9-1 TMRBx方块图

9.3 寄存器

9.3.1 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“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址(基+)
启动寄存器	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
上升计数器捕获寄存器	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

注：在计时器运行期间，不应修改时钟控制寄存器，计时器模式寄存器和计时器触发器控制寄存器。应在停止计时器运行之后再对其进行修改。

9.3.2 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 工作</p> <p>0: 禁用</p> <p>1: 启用</p> <p>规定 TMRB 运行。在该运行被禁用时，不会向 TMRB 模块中的其它寄存器提供任何时钟脉冲。可降低功耗。(其可禁止从 TBxEN 寄存器以外的其它寄存器读取及向这些寄存器写入。)</p> <p>如需使用 TMRB，可在对该 TMRB 模块中的各寄存器进行编程之前，启动该 TMRB 运行(即将其设置为"1")。如 TMRB 在被禁用之前执行了运行，则各寄存器中仍将保留该设置。</p>
6	TBHALT	R/W	<p>调试 HALT 期间的时钟工作。</p> <p>0: 运行</p> <p>1: 停止</p> <p>规定 TMRB 时钟在调试工具于使用中过渡到 HALT 模式时，设置为运行或停止。</p>
5-0	-	R	读作 "0"。

9.3.3 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	R/W	预分频器工作 0: 停止&清除 1: 计数
1	-	R	读作"0"。
0	TBRUN	R/W	计数工作 0: 停止&清除 1: 计数

9.3.4 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	TBWBF	-	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	TBWBF	R/W	双缓冲 0: 禁用 1: 启用
6	-	R/W	写入 "0"
5	TBSYNC	R/W	同步模式切换 0: 单个(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	选择外部寄存器。 0: 上升 1: 下降 在外部触发器被选中时控制该沿选择(发送给 TBxIN 引脚的信号)。
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

9.3.5 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 在写入 "0"时, 捕获寄存器 0(TBxCP0)取计数值。读作 "1"。
5-4	TBCPM[1:0]	R/W	捕获计时 00: 禁用捕获 01: 保留 10TBxIN↑, TBxIN↓ 一旦 TBxIN 引脚输入上升, 即将各计数值纳入捕获寄存器 0 (TBxCP0)。 一旦 TBxIN 引脚输入下降, 即将各计数值纳入捕获寄存器 1(TBxCP1)。 11: TBxFF0↑, TBxFF0↓ 将计数值存储到捕获寄存器 0(TBxCP0)一旦 TBxFF0 上升和捕获寄存器 1(TBxCP1)一旦 TBxFF0 下降。
3	TBCLE	R/W	上升计数器控制器 0: 禁用上升计数器的清除。 1: 启用上升计数器的清除。 清除并控制该上升计数器。 在已写入"0"时, 其可禁用上升计数器的清除。在"1"被写入时, 其可清除上升计数器前提是上升计数器可匹配计时器寄存器 1(TBxRG1)。
2-0	TBCLK[2:0]	R/W	选择该 TMRBx 源时钟。 000: TBxIN 引脚输入 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

注: 在TMRBx正在运行期间, 请勿对TBxMOD寄存器进行任何改动。

9.3.6 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	在该上升计数器值被纳入 TBxCP1 时, TBxFF0 反转触发。 0: 禁用触发器 1: 启用触发器 通过设置"1", 该计时器触发电路会在上升计数器被纳入该捕获寄存器 1(TBxCP1)时反转。
4	TBC0T1	R/W	在该上升计数器值被纳入 TBxCP0 时, TBxFF0 反转触发。 0: 禁用触发器 1: 启用触发器 通过设置"1", 该计时器触发电路会在上升计数器被纳入该捕获寄存器 0(TBxCP0)时反转。
3	TBE1T1	R/W	在该上升计数器值与 TBxRG1 匹配时, TBxFF0 反转触发。 0: 禁用触发器 1: 启用触发器 通过设置"1", 该计时器触发电路会在上升计数器匹配计时器寄存器 1(TBxRG1)时反转。
2	TBE0T1	R/W	在该上升计数器值与 TBxRG0 匹配时, TBxFF0 反转触发。 0: 禁用触发器 1: 启用触发器 通过设置"1", 该计时器触发器可在上升计数器匹配计时器寄存器 0 (TBxRG0)时反转。
1-0	TBFF0C[1:0]	R/W	TBxFF0 控制 00: 反转 反转 TBxFF0 的值(使用软件反转)。 01: 设置 把 TBxFF0 设置为"1"。 10: 清除 把 TBxFF0 清为 "0"。 11: 忽略 始终读作 "11"。

9.3.7 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: 无溢出发生 1: 发生溢出 在某上升计数器溢出时, 即设置"1"。
1	INTTB1	R	匹配(TBxRG1)中断请求标志 0: 无匹配被检测到。 1: 检测到与 TBxRG1 之间存在的一处匹配 在检测到计时器寄存器 1(TBxRG1)的匹配时, 即设置"1"。
0	INTTB0	R	匹配(TBxRG0)中断请求标志 0: 无匹配被检测到。 1: 检测到与 TBxRG0 之间存在的一处匹配 在检测到计时器寄存器 0(TBxRG0)的匹配时, 即设置"1"。

注 1: 即使TBxIM寄存器进行的屏蔽配置有效, 该状态仍会被设置为TBxST寄存器。

注 2: 当TBxIM寄存器的对应位禁用中断屏蔽配置时, 向CPU发出中断。

注 3: 在清除该标志时, 应读取TBxST寄存器。

9.3.8 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 将该上升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 用计时器寄存器 1(TBxRG1)设置该匹配中断请求屏蔽, 以启用或禁用。
0	TBIM0	R/W	匹配(TBxRG0)中断请求屏蔽 0: 禁用 1: 启用 用计时器寄存器 0(TBxRG0)设置该匹配中断请求屏蔽, 以启用或禁用。

注: 即使通过TBxIM寄存器进行的屏蔽配置有效, 该状态仍会被设置到TBxST寄存器。

9.3.9 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	通过读取上升计数器输出，捕获某个值。 如果 TBxUC 在计数器运行期间被读取，则上升计数器的当前值会被捕获。

9.3.10 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上升计数器。

9.3.11 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上升计数器。

9.3.12 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	已读取从该上升计数器捕获到的某个值。

9.3.13 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	已读取从该上升计数器捕获到的某个值。

9.4 工作说明

9.4.1 预分频器

有预分频器可用于生成上升计数器的源时钟。

预分频器输入 $\phi T0$ 是 $f_{periph}/1$, $f_{periph}/2$, $f_{periph}/4$, $f_{periph}/8$, $f_{periph}/16$ 或 $f_{periph}/32$ 在 CG 电路中所选择的 $CGSYSCR<PRCK[2:0]>$ 。外设时钟是 $CGSYSCR<FPSEL>$ 在 CG 电路中选择时钟 f_{gear} , 或 f_c 其在被该时钟齿轮分频之前是时钟。

可用 $TBxRUN<TBPRUN>$ 设置预分频器的运行或停止, 其中, 写入"1"即可开始状态计数, 写入"0"即可清除和停止计数。

9.4.2 上升计数器(UC)

UC 为 16-位二进制计数器。

9.4.2.1 源时钟

可通过 $TBxMOD<TBCLK[2:0]>$ 指定 UC 的源时钟。

可从预分频器输出时钟 $-\phi T1$, $\phi T4$, $\phi T16$, $\phi T32$, $\phi T64$, $\phi T128$ 和 $\phi T256-$, 或 $TBxIN$ 引脚的外部时钟中进行选择。

9.4.2.2 计数器启动 / 停止

计数器启动方式有软件启动, 外部触发器启动和同步启动。

1. 软件启动

如果 $<TBRUN>$ 被设置为"1", 计数器随即启动。如果"0"即被设置到 $<TBRUN>$, 则计数器将停止, 且上升计数器同时被清除。

2. 外部触发器启动

在外部触发器模式下, 可通过外部信号启动计数器。

如果 $TBxCR<CSSEL>$ 被设置为"1", 则被设置为外部触发器启动模式。此时, 如果 $<TBRUN>$ 被设置为"1", 计数器的条件则为触发器等待。计数器将在 $TBxIN$ 的上升/下降沿启动。

$TBxCR<TRGSEL>$ 位可指定外部触发器沿的切换状态。

$<TRGSEL>="0"$: $TBxIN$ 的上升沿被选中。

$<TRGSEL>="1"$: $TBxIN$ 的下降沿被选中。

如果 $<TBRUN>$ 被设置为"0", 则计数器将停止, 且上升计数器同时被清除。

3. 同步启动

在计时器同步模式下, 可使用同步启动计时器。如果计时器同步模式被用于 PPG 输出模式, 则可实现马达驱动应用。

已根据具体产品, 确定主设备通道和从设备通道的组合。有关本产品主设备通道和从设备通道的组合, 请参看“产品信息”一节。

TBxCR<TBSYNC>位可指定同步模式的切换状态。如果从设备通道的<TBSYNC>位被设置为"1", 则计数器会与软件或主设备通道的外部触发器启动装置同步启动/停止。无需设置从设备通道的 TBxRUN<TBPRUN, TBRUN>位。必须将主设备通道的 TBSYNC>位设置为"0"。

注意如果对外部触发器计数器模式和计时器同步模式都进行了设置, 则计时器同步模式可获得较高的优先级。

9.4.2.3 清除 UC 时序

1. 在检测到与 TBxRG1 之间存在的一处匹配时

通过设置 TBxMOD<TBCLE> = "1", UC 即被清除前提是比较器检测到 UC 和 TBxRG1 之间存在的匹配。

2. 在 UC 停止时

如果 TBxRUN<TBRUN> = "0", 则 UC 停止后随即会被清除。

9.4.2.4 UC 溢出

如果 UC 发生溢位, 会产生 INTTBx 溢位中断。

9.4.3 计时器寄存器(TBxRG0, TBxRG1)

寄存器 TBxRG0 和 TBxRG1 都用于设置与上升计数器值进行比较的值, 且这两个寄存器都被内置在相应的通道中。如果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设置值之间存在的一处匹配, 则比较器会输出匹配检测信号。

TBxRG0 和 TBxRG1 由与寄存器缓冲器配套的双缓冲配置组成。在初始状态时, 该双缓冲被禁用。

通过 TBxCR<TBWBF>可指定控制双缓冲器禁用或启用。如果<TBWBF> = 0, 则双缓冲变为禁用, 如果<TBWBF> = "1", 则其变为启用。

在双缓冲被启用时, 如果 UC 与 TBxRG1 匹配, 则从寄存器缓冲器向计时器寄存器(TBxRG0/1)传送数据。

如果即使双缓冲被启用 UC 仍被停止, 则双缓冲可作为单缓冲器运行, 且数据可被直接写入到 TBxRG0 和 TBxRG1。

9.4.4 捕获控制

电路可控制从 UC 进入 TBxCP0 和 TBxCP1 中的锁存器值的计时。可通过 TBxMOD<TBCPM[1:0]>指定 UC 的捕获计时。

软件还可捕获被发往捕获寄存器的 UC 值。"0"每被写入到 TBxMOD<TBCP>一次, UC 的值就会被发送到 TBxCP0 一次。

9.4.5 捕获寄存器(TBxCP0, TBxCP1)

寄存器可俘获 UC 的值。

9.4.6 上升计数器捕获寄存器(TBxUC)

如果 TBxUC 寄存器在计数器运行期间被读取,则上升计数器的当前值会被捕获,且该值会被读取。在该计数器处于停止状态期间,最后捕获的值会被保持。

9.4.7 比较器(CP0, CP1)

电路可与 UC 和被设置到 TBxRG0/1 的值进行比较,并检测到匹配。如果检测到匹配,则会出现 INTTBx。

9.4.8 计时器触发器(TBxFF0)

计时器触发器(TBxFF0)是通过由比较器发出的匹配信号及发送至捕获寄存器的锁存信号逆转的。通过设置 TBxFFCR<TBC1T1, TBC0T1, TBC1T1, TBC1T0>,可启用或禁止其逆转。

在复位之后, TBxFF0 的值变为未定义状态。通过将"00"写入到 TBxFFCR<TBFF0C[1:0]>,即可让触发电路发生反转。通过写入"01"可将其设置为"1",通过写入"10",可将其清除为"0"。

TBxFF0 的值可被输出到计时器输出引脚(TBxOUT)。如果计时器输出被执行,则应预先对端口设置进行编程。

9.4.9 捕获中断(INTCAPx0, INTCAPx1)

可按从 UC 发送到 TBxCP0 和 TBxCP1 的锁存值的计时,生成 INTCAPx0 和 INTCAPx1。

9.5 各模式工作说明

9.5.1 间隔计时器模式

如果是生成恒定周期中断，则可将间隔时间设置到该计时器寄存器(TBxRG1)，以生成 INTTBx 中断。

	7	6	5	4	3	2	1	0	
TBxEN	← 1	X	X	X	X	X	X	X	启用 TMRBx 工作。
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	← *	*	*	*	*	*	*	*	允许通过把对应位设置为"1"的方式生成 INTTBx 中断。
TBxFFCR	← X	X	0	0	0	0	1	1	禁用 TBxFF0 反向触发器
TBxMOD	← X	1	0	0	0	*	*	*	改为将预分频器输出时钟用作输入时钟。指定捕获功能为禁用。
(*** = 001 ~ 111)									
TBxRG1	← *	*	*	*	*	*	*	*	指计时间间隔。(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, -; 不要改动

9.5.2 事件计数器模式

通过将某个源时钟用作外部时钟(TBxIN 引脚输入)，即可将 TMRBx 成为事件计数器。

UC 会在 TBxIN 引脚输入的上升沿递增计数。可通过软捕获，来捕获 UC 的值。可通过读取之而读取该计数值。

	7	6	5	4	3	2	1	0	
TBxEN	← 1	X	X	X	X	X	X	X	启用 TMRBx 工作。
TBxRUN	← X	X	X	X	X	0	X	0	停止预分频器和计数器。
将相应的端口指定给 TBxIN。									
TBxFFCR	← X	X	0	0	0	0	1	1	禁用 TBxFF0 撤销触发器。
TBxMOD	← X	1	0	0	0	0	0	0	将其作为 TBxIN 引脚输入设置到某个源时钟。
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, -; 不要改动

9.5.3 可编程脉冲发生(PPG)输出模式

可输出具备任何频率和任何占空比的方波。输出脉冲的活动性可高可低。

在 UC 可匹配 TBxRG0 和 TBxRG1 的设定值时, TBxFF0 即被反转。可从 TBxOUT 引脚输出 TBxFF0。

注意: TBxRG0 和 TBxRG1 的设定值必须满足以下要求。

TBxRG0 的设定值 < TBxRG1 的设定值

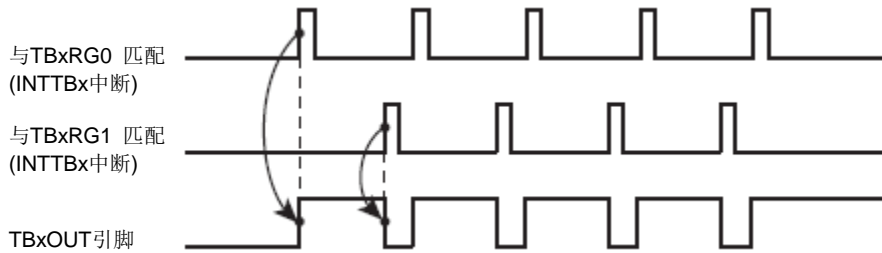


图 9-2 可编程脉冲发生输出示例

在该模式下, 通过启用该双缓冲, 寄存器缓冲器 0 和 1 的值在 UC 可匹配 TBxRG1 的值时被移位到 TBxRG0 和 1。

无需对 TBxRG0 和 TBxRG1 的修改进行计时, 即可修改频率和占空比。

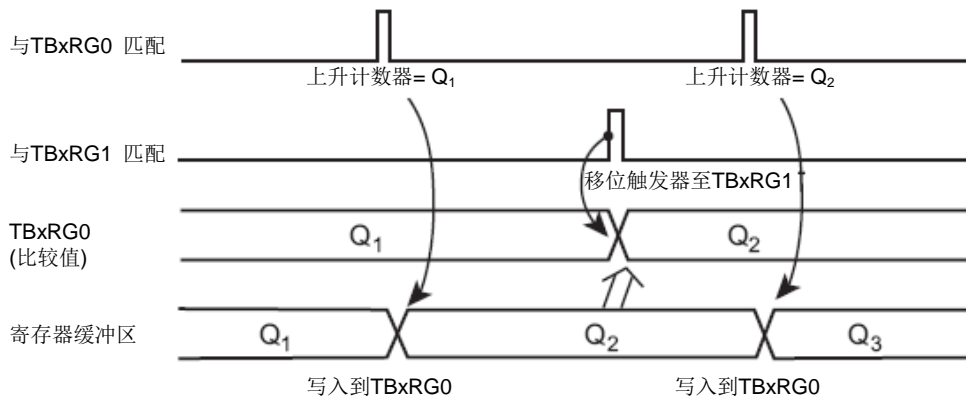


图 9-3 寄存器缓冲器工作

该模式的方块图如以下所示。

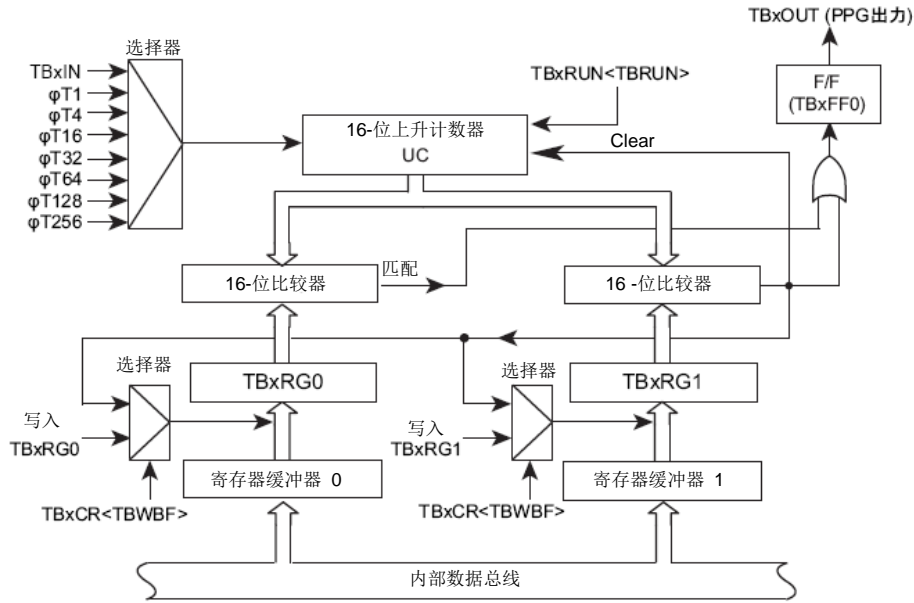


图 9-4 16-位PPG模式的方块图

处于 16-位 PPG 输出模式下的各寄存器的编程，均应按以下要求进行。

	7	6	5	4	3	2	1	0	
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。
TBxCR	← 1	0	X	X	X	0	X	X	启用双缓冲。
TBxRG0	← *	*	*	*	*	*	*	*	设置占空比。
	← *	*	*	*	*	*	*	*	
TBxRG1	← *	*	*	*	*	*	*	*	设置周期。
	← *	*	*	*	*	*	*	*	
TBxFFCR	← X	X	0	0	1	1	1	0	在检测到与 TBxRG0 或 TBxRG1 之间存在的一处匹配时，指定至触发器 TBxFF0 以实现逆转。并将 TBxFF0 的初始值设置为"0"。
TBxMOD	← X	1	0	0	0	*	*	*	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)

将某个相应端口分配给 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, -: 不要改动

9.5.4 可编程脉冲发生(PPG)外部触发器输出模式

通过使用外部触发器计数启动模式，可输出一个具备短延迟时间的 PPG 波。

以下给出了通过外部触发器计数启动模式实现一次性脉冲输出的示例。

在通过 TBxIN 的上升沿启动递增计数时，需将 TBxCR<CSSEL> 设置为"1"，并在停止 16-位上升计数器时将 TBxCR<TRGSEL>清除为"0"。

TBxRG0 可根据外部触发器信号设置延迟时间(d)。TBxRG1 可设置该(d)+(p)值，即一次性脉冲的延迟时间(d)加上宽度(p)。

如在 UC 可匹配 TBxRG0 和 TBxRG1 时逆转 TBxFF0，则需将 TBxFFCR<TBE1T1>和 TBxFFCR<TBE1T1> 设置为"1"。

通过将 TBxRUN<TBPRUN>和 TBxRUN<TBRUN>设置为"1"，UC 可启动 UC。

UC 可通过外部触发器的上升沿启动。

在 UC 计数达到(d)，且 UC 可匹配 TBxRG0 时，TBxFF0 即被反转。TBxFF0 属于“高”电平。

在 UC 计数达到(d)+(p)，且 UC 可匹配 TBxRG1 时，TBxFF0 即被反转。TBxFF0 属于“低”电平。

在固定 TBxFF0 的电平时，需将 TBxFFCR<TBE1T1>和 TBxFFCR<TBE0T1>清除为"0"，或通过 UC 可匹配 TBxRG1 时所生成的 INTTBx 中的 TBxRUN<TBPRUN><TBRUN>使 UC 停止运行。

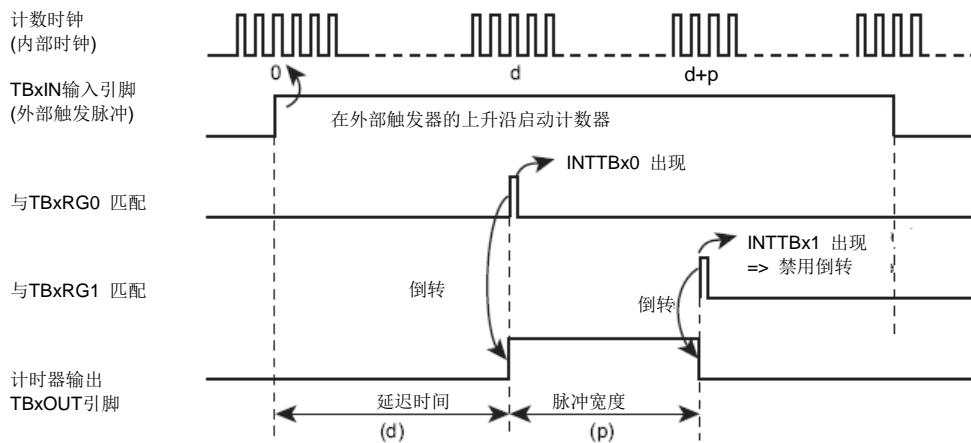


图 9-5 通过外部触发器启动带延迟的一次性脉冲输出

以下给出了通过在上升沿触发 TBxIN 输入，宽度为 2 ms 的一次性脉冲在 3 ms 后被输出时的设置。在该示例中，源时钟为 $\phi T1$ 。

		7	6	5	4	3	2	1	0	
[主处理]										
将相应的端口指定给 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。(3 ms/φT1)
TBxRG0	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	设置计数值。(3+2)ms/φT1)
TBxRG1	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	0	0	1	1	1	0	如果 UC 可匹配 TBxRG0 和 TBxRG1, 则逆转 TBxFF0。将 TBxFF0 清除为"0"。
TBxMOD	←	X	1	0	0	0	0	0	1	启动 UC (自由运行)。选择源时钟的 φT1。禁用捕获 UC。
将某个相应端口分配给 TBxOUT。										
TBxIM	←	X	X	X	X	X	1	0	1	TBxRG1 中断以外的屏蔽。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	通过将相应的位设置为"1", 从而允许生成 INTTBx 中断所指定的中断。
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]禁止输出										
TBxFFCR	←	X	X	-	-	0	0	-	-	清除 TBxFF0 倒转触发器设置。
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, -: 不要改动										

9.6 使用捕获功能的应用

许多应用都可使用捕获功能。
以下给出了这些应用。

1. 频率测量
2. 脉冲宽度测量

9.6.1 频率测量

可测量外部时钟的频率。

在测量频率时，TMRB_m 被用作 16-位间隔计时器模式，而 TMRB_n 被用作 16-位事件计数器模式。

在通过外部时钟对 TMRB_n 的 UC 自由计数时，需将 TM_nMOD<TBCLK> 设置为"000"，并将 TB_nRUN<TBE1T1><TBE0T1> 设置为"11"。

在 TMRB_m 的 UC 可匹配 TB_mRG0 和 TB_mRG1 的情况下逆转 TB_mFF0 时，需将 TB_mFFCR<TBE1T1><TBE0T1> 设置为"11"。

在 TB_mFF0 的上升沿将 UC 捕获至 TB_nCP0，以及在 TB_mFF0 的下降沿将 UC 捕获至 TB_nCP1 时，需将 TB_xMOD<TBCPM> 设置为"11"。

将 TB_mRG0 和 TB_mRG1 设置为 UC 对外部时钟进行计数并启动 TMRB_m 的时间。

在 TMRB_m 的 UC 可匹配 TB_mRG0，并可将 TMRB_n UC 的值捕捉到 TB_nCP0 时，TB_mFF0 上升。
在 TMRB_m 的 UC 可匹配 TB_mRG1，并可将 TMRB_n UC 的值捕捉到 TB_nCP1 时，TB_mFF0 下降。

可按照 INTTB_m 中的 $(\text{TB}_n\text{CP1} - \text{TB}_n\text{CP0}) \div (\text{TB}_m\text{RG1} - \text{TB}_m\text{RG0})$ 进行频率测量。

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

TB_nCP1 - TB_nCP0 可小于零，具体值取决于 TB_mFF0 的变化计时。如果 TB_nCP1 - TB_nCP0 小于零，则请校正该值。

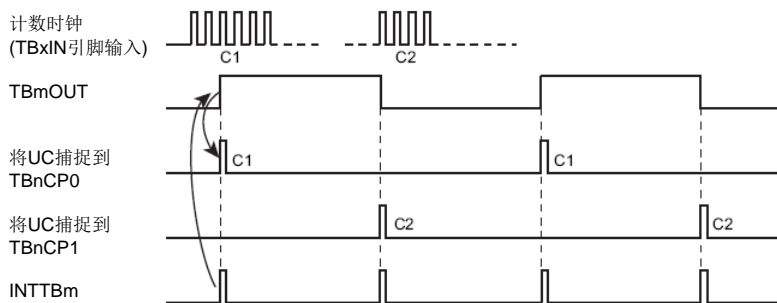


图 9-6 频率测量

以下给出了所测脉冲被输入到 TB_xIN 的情形。在该示例中，源时钟为 φT1。

7 6 5 4 3 2 1 0

[主处理] 用 TB_mFF0 捕获设置值

		7	6	5	4	3	2	1	0	
将对应端口分配给 TBxIN。										
TBmEN	←	1	X	X	X	X	X	X	X	启用 TMRBm 工作。
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 工作。
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/下降沿俘获。清除并禁用 UC。输入时钟为 TBxIN。
TBmIM	←	X	X	X	X	X	1	0	1	TBxRG1 中断以外的屏蔽。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	允许生成 INTTBm 指定的中断中断的对应位被设置为"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 倒转触发器设置。
中断启用清除寄存器	←	*	*	*	*	*	*	*	*	禁止生成 INTTBm 指定的中断对应位被设置为"1"。

TBnCP0 和 TBnCP1 被读出, 频率随即被计算出来。

注: X: 忽略, *: 可选值, -: 不要改动

9.6.2 脉冲宽度测量

可测量外部脉冲的“高”电平宽度。

在 TBxIN 的上升沿将 UC 捕获至 TBxCP0，以及在 TBxIN 的下降沿将 UC 捕获至 TBxCP1 时，需将 TBxMOD<TBCPM>设置为“10”。

启用 INTCAPx1 中断。

启用 TMRBx 运行。

在外部脉冲的上升沿进入到 TBxIN 之内时，将 UC 的值捕捉到 TBxCP0。在外部脉冲的下降沿进入到 TBxIN 之内，且 INTCAPx1 中断出现时，将 UC 的值捕捉到 TBxCP1。

通过将预分频器输出时钟的时钟周期乘以 TBxCP0 和 TBxCP1 之差，即可计算出该外部脉冲的“高”电平宽度。

例如，如果 TBxCP0 与 TBxCP1 之间的差为 100，预分频器输出时钟的周期为 0.5 μs ，那么脉冲宽度为 $100 \times 0.5 \mu\text{s} = 50 \mu\text{s}$ 。

如果所测出的脉冲宽度超过了 UC 的最大计数时间，则请校正该测量值。

也可测量外部脉冲的“低”电平宽度。

在这种情况下，可启用 INTCAPx0 中断。可将“图 9-7 脉冲宽度测量”所示 INTCAPx0 中断的两次进程中的第一次时所生成的 C2 和第二次时所生成的 C1 之间的差值，乘以该预分频器输出时钟的周期。

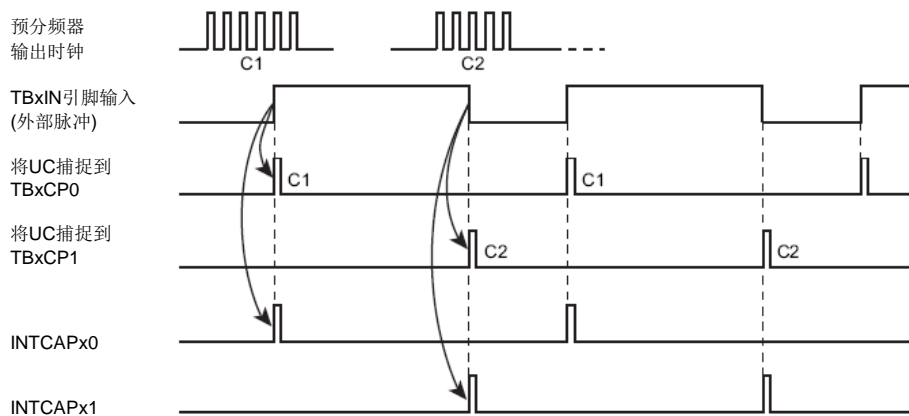


图 9-7 脉冲宽度测量

以下所示为进入到 TBxIN 之内的外部脉冲的“高”电平宽度的测量程序。在该示例中，源时钟为 $\phi T1$ 。

		7	6	5	4	3	2	1	0	
[主处理] 捕获设置 TBxIN。										
将相应的端口指定给 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。
TBxFFCR	←	X	X	0	0	0	0	1	0	清除 TBxFF0 逆转触发器和 TBxFF0。将 TBxFF0 初始值设为"0"。
TBxMOD	←	X	1	1	0	0	0	0	1	启动 UC (自由运行)。选择源时钟的 φT1。UC 被捕捉到 TBxIN 上升沿的 TBxCP0。UC 被捕捉到 TBxIN 下降沿的 TBxCP1。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	允许生成 INTCAPx1 指定的中断中断对应位被设置为"1"。
TBxRUN	←	X	X	X	X	X	1	X	1	启动预分频器 和计数器。
[处理 INTCAPx1 中断服务程序] 计算"高"电平的宽度。										
中断启用清除寄存器	←	*	*	*	*	*	*	*	*	禁止生成 INTCAPx1 指定的中断中断对应位被设置为"1"。
通过读取 TBxRG0 和 TBxRG1，计算“高”电平的宽度。										

注：X； 忽略，*； 任选值，-； 不要改动



10. 16-bit 定时器 A (TMR16A)

10.1 概要

TMR16A包含以下功能：

- 匹配中断
- 矩形波形输出
- 读取捕获

在本章中，“x”表示通道编号。

10.2 方块图

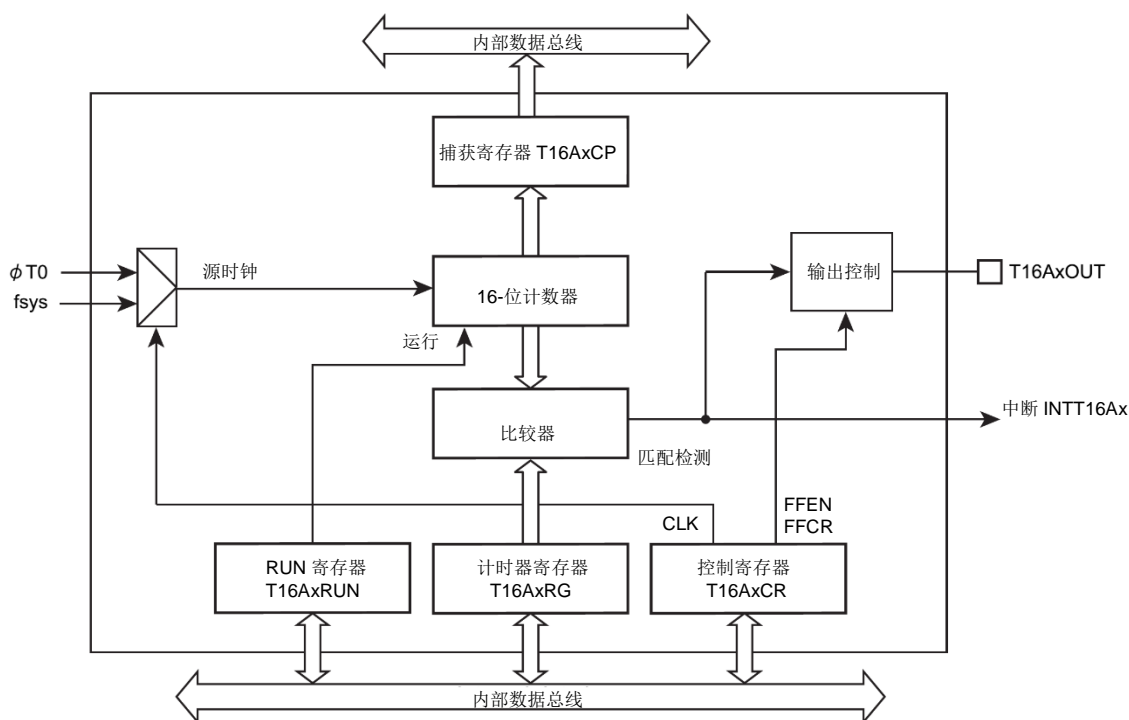


图 10-1 TMR16A的方块图

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。

有关基地址的详细情况，参照“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址 (基本+)
启动寄存器	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：在T16ARUN<RUN>被设置为“1”时，请勿修改T16AxEN，T16AxCR，T16AxRG和T16AxCP。

10.3.2 寄存器的详细

10.3.2.1 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	暂停模式调试期间的工作 0: 工作 1: 停止 指定暂停模式调试期间的工作。将"1"写入到该位，以停止该工作。
0	I2T16A	R/W	IDLE 模式期间的工作 0: 停止 1: 工作 指定 IDLE 停模式期间的工作。将"1"写入到该位，以延续该工作。

10.3.2.2 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 0: 停止 1: 工作

10.3.2.3 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	UCCR	-	-	-	-	-	-	-
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	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	R	读作 "0"。
15	UCCR	R/W	写入 "1"
14-8	-	R	读作 "0"。
7	FFEN	R/W	T16AxOUT 的反转 0: 禁用 1: 启用 在计数器可匹配 T16ARG 时, 将"1"写入该位, 以转换 T16AxOUT。
6	-	R	读作 "0"。
5-4	FFCR[1:0]	W	T16AxOUT 控制 00: 反转 01: 设置 10: 清除 11: 无工作 将某值写入到该位, 以通过软件控制 T16AxOUT。 读作"11"。
3-1	-	R	读作"0"。
0	CLK	R/W	源时钟 0: fsys 1: $\Phi T0$ 指定某个源时钟。

10.3.2.4 T16AxRG (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	设置某值, 以与计数器进行比较。

注: 请勿设置为"0x0000"。

10.3.2.5 T16AxCP (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/W	计数器值 [读取] 读取当前计数器值。 [写入] 设置计数器值。仅当计数器可匹配 T16AxRG<RG>时才清除计数器，因此，应在启动工作之前写入“0x0000”以清除该寄存器。

10.4 工作说明

10.4.1 计时器工作

1. 准备

将"1"设置到 T16AxCR<UCCR>。

用 T16AxCR<CLK>选择某个源时钟。写入"0"以设置 fsys，或写入"1"以设置 $\Phi T0$ 。将某个计数器值设置到 T16AxRG<RG>。

在计数器停止(T16AxRUN<RUN>为"0")期间，修改 T16AxCR 和 T16AxRG。

2. 计数器工作

在启动计数器工作之前，将"0x0000"设置到 T16AxCP<CP>，以清除该计数器。

在启动递增计数时，需将"1"设置到 T16AxRUN<RUN>。如果该计数器值可匹配 T16AxRG<RG>的值，则其可被清除为"0x0000"，并继续递增计数。

3. 匹配检测中断生成

如果某个计数器值可匹配 T16AxRG 的值，则将输出匹配检测中断 INTT16Ax。

4. 停止

如需停止计数，可将"0"设置到 T16AxRUN<RUN>。该计数器值会被保持。然后，在将"1"设置到<RUN>以启动计数之前，清除该计数器。

注：必须在计数器处于停止状态(T16AxRUN<RUN>被设置为"0")期间，执行对 T16AxCR，T16AxRG 和 T16AxCP 的修改。

10.4.2 T16AxOUT 控制

可通过寄存器设置，或通过将该计数器与 T16AxRG 匹配，来实现对 T16AxOUT 的修改。T16AxOUT 的初始状态为"0"。

1. 通过软件实现控制

利用 T16AxCR<T16AFFC>设置，可指定 T16AxOUT；"1"需设置，"0"需清除，还可实现反转设置。在该计数器停止 (T16AxRUN<RUN>为"0")期间修改 T16AxCR。

2. 因匹配计数器而出现的反转

将"1"写入到 T16ACR<FFEN>，以反转 T16AxOUT。在 T16AxRG 可匹配某个计数器值时，T16AxOUT 将反转。在该计数器停止时，T16AxOUT 的状态仍将保持不变。

10.4.3 读取捕获

通过读取 T16AxCP，可捕获该计数器的当前值。

10.4.4 自动停止

在 T16AxEN 设置后，TMR16A 会在以下条件下自动停止：

1. 转换为 IDLE 模式/从 IDLE 模式转换为其它模式

在 T16AEN<I2T16A>设置后，可指定 IDLE 模式期间的 TMR16A 运行状态。如设置为"1"，则 TMR16A 会在发生指向 IDLE 模式的转换时自动停止递增计数。如果 TMR16A 从 IDLE 模式返回，则其将重新启动递增计数运行。

2. 调试暂停

在 T16AEN<HALT>设置后，可指定调试暂停期间否认 TMR16A 运行状态。如果设置为"0"，则 TMR16A 会在指向调试 HALT 模式的转换发生时自动停止递增计数。如果内核的调试暂停模式被取消，则递增计数将重新启动。

11. 串行通道(SIO/UART)

11.1 概述

串行通道具备以下所述的模式。

- 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
 - 被预分频器从外设时钟($\phi T0$)频率分为 1/2, 1/8, 1/32, 1/128。
 - 使将预分频器输出时钟频率分为 1 ~ 16 成为可能。
 - 使将预分频器输出时钟频率分为 $N+m/16$ ($N=2 \sim 15$, $m=1 \sim 15$)成为可能。(仅UART模式)
 - 可用的系统时钟(仅UART模式)。
- 双缓冲
 - 可用的双缓冲器功能。
- I/O接口模式
 - 传送模式: 半双工(发送/接收), 全双工
 - 时钟: 输出(固定上升沿) /输入(可选上升/下降沿)
 - 使指定连续发送的间隔时间成为可能。
- UART模式
 - 数据长度: 7 位, 8 位, 9 位
 - 新增奇偶校验位(按 9 位数据长度)
 - 串行链接以使用唤醒功能
 - 握手功能用 $\overline{CTS_x}$ 引脚
- 使用IR载波脉冲的输出信号

在以下说明中, "x"表示通道编号。

11.2 配置

图 11-1 显示串行通道方块图。

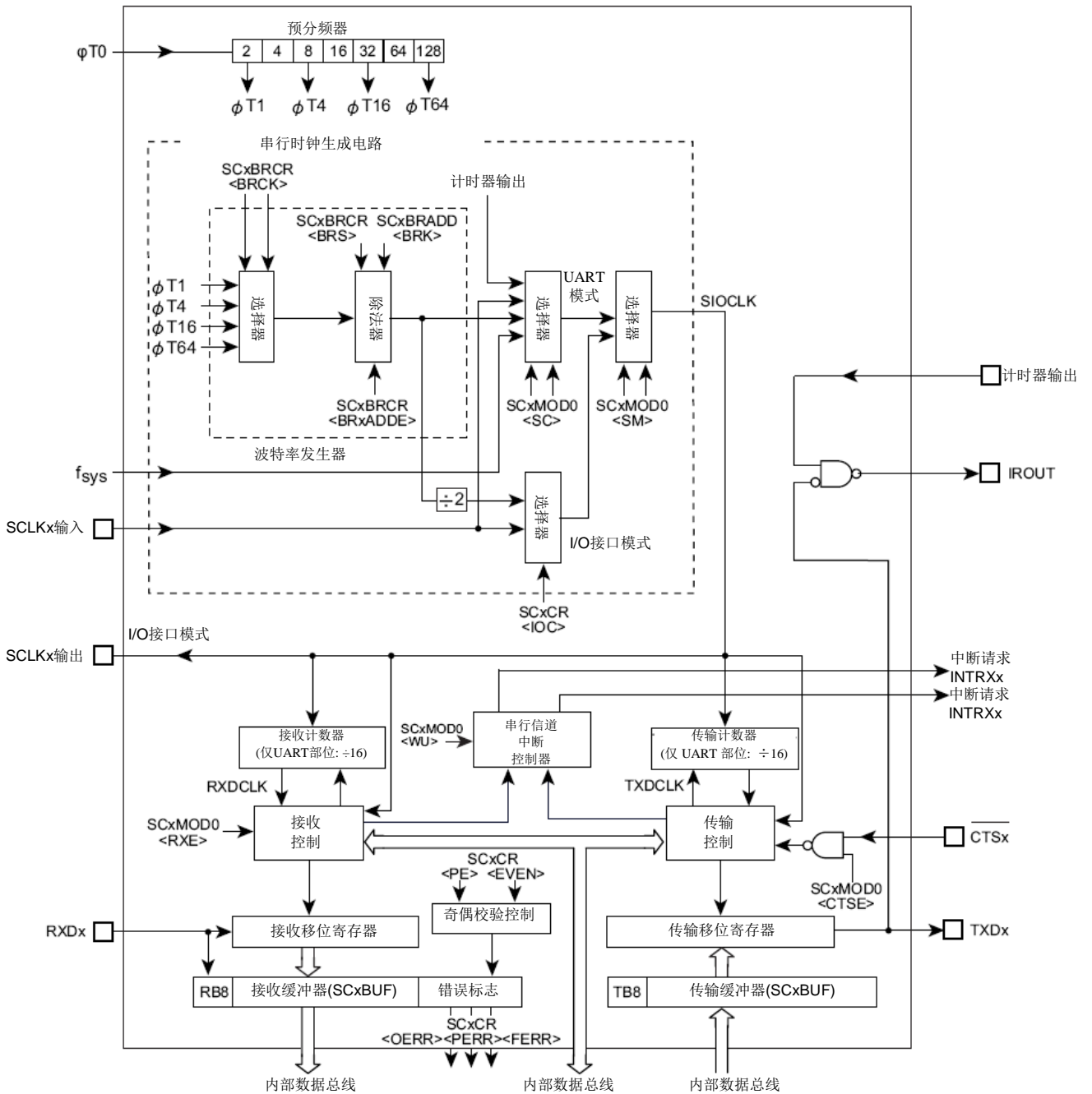


图 11-1 串行通道方块图

11.3 寄存器描述

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，请参照“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址(基+)
启动寄存器	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

注：当正在传输或接收数据时，请勿修改任何控制寄存器。

11.3.2 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	-	-	-	-	-	-	-	SIOE
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	SIOE	R/W	串行通道工作 0: 禁用 1: 启用 规定的串行通道工作。 要使用串行通道, 设置<SIOE> = "1"。 工作禁用时, 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。 通道工作被执行随即禁用, 该设定值将保持在每个寄存器中。

11.3.3 SCxBUF (缓冲器寄存器)

SCxBUF 可用作写入工作的发送缓冲器, 以及读取工作的接收缓冲器。

	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	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	[写入] TB: 传输缓冲器 [读取] RB: 接收缓冲器



11.3.4 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	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	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
31-8	-	R	读作 "0"。
7	RB8	R	接收数据位 8 (适用于 UART) 在 9-位 UART 模式下, 接收数据的第 9 位。
6	EVEN	R/W	奇偶性(适用于 UART) 0: 奇数 1: 偶数 选择偶或奇校验。 "0": 奇校验, "1": 偶校验 奇偶校验位仅可用于 7-或 8-位 UART 模式。
5	PE	R/W	加奇偶(适用于 UART) 0: 禁用 1: 启用 可控制启用/禁用奇偶性。 奇偶校验位仅可用于 7-或 8-位 UART 模式。
4	OERR	R	超载运行出错标志(注) 0: 正常工作 1: 错误
3	PERR	R	奇偶性/欠载工作错误标志(注) 0: 正常工作 1: 错误
2	FERR	R	成帧错误标志(注) 0: 正常工作 1: 错误
1	SCLKS	R/W	选择输入时钟边沿。(适于 I/O 接口) 在时钟输出模式下, 设置为 "0"。 0: 传输缓存器中的数据发送到 TXDx 引脚, 在 SCLKx 的下降沿一次发送一位。 在接收缓存器中接收 RXDx 引脚的数据, 在 SCLKx 的上升沿一次接收一位。 在这种情况下, SCLKx 从高电平启动。 1: 发送缓存器中的数据发送到 TXDx 引脚, 在 SCLKx 的上升沿一次发送一位。 在接收缓存器中接收 RXDx 引脚的数据, 在 SCLKx 的下降沿一次接收一位。 在这种情况下, SCLKx 从低电平启动。
0	IOC	R/W	选择时钟 (适用于 I/O 接口) 0: 波特率发生器 1: SCLK 引脚输入

注: 读取时, 任何错误标志 (OERR, PERR, FERR) 都会被清除至"0"。

11.3.5 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	传输数据位 8 (适用于 UART) 以 9-位 UART 模式写入传输数据的第 9 位。
6	CTSE	R/W	握手功能控制(适用于 UART) 0: CTS 禁用 1: CTS 启用 可控制握手功能。 设置 "1"使用 CTS 引脚启用信号交换功能。
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(适用于 UART) 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。 在其处于启用状态时, 中断仅适用于 9-位 UART 模式下 RB9 = "1"时的情况。
3-2	SM[1:0]	R/W	指定传送模式。 00: I/O 接口模式 01: 7-位长度 UART 模式 10: 8-位长度 UART 模式 11: 9-位长度 UART 模式
1-0	SC[1:0]	R/W	串行传送时钟(适用于 UART) 00: 计时器输出 01: 波特率发生器 10: 内部时钟 fsys 11: 外部时钟(SCLK 输入) (至于 I/O 接口模式, 可在控制寄存器(SCxCR)中设置串行传送时钟)

注 1: 在设置各模式寄存器(SCxMOD0, SCxMOD1和SCxMOD2)之后, 将<RXE>设置为"1"。

注 2: 当正在接收数据时, 请勿停止接收工作(通过将SCxMOD0<RXE>设置至 "0")。

11.3.6 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: 工作 指定 IDLE 模式工作。
6-5	FDPX[1:0]	R/W	传送模式设置 00: 传送被禁止 01: 半双工(接收) 10: 半双工(传输) 11: 全双工 可在 I/O 接口模式下配置该传送模式。
4	TXE	R/W	传输控制器(注 1)(注 2) 0: 禁用 1: 启用 该位可启用传输, 且在所有传送模式下均有效。
3-1	SINT[2:0]	R/W	连续传输的间隔时间(适用于 I/O 接口) 000: 无 001: 1 x SCLK 010: 2 x SCLK 011: 4 x SCLK 100: 8 x SCLK 101: 16 x SCLK 110: 32 x SCLK 111: 64 x SCLK 在已选择 SCLK 引脚的情况下, 该参数仅对 I/O 接口模式有效。在其它模式下, 该功能无意义。 当双缓冲以输入/输出接口模式启用时, 确定连续传输的间隔时间。
0	-	R/W	写作 "0"。

注 1: 首先指定所有模式, 然后启用该<TXE>位。

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

11.3.7 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	SBLEN	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	发送缓冲区为空标志 0: 满 1: 空 如果双缓冲被禁用, 则该标志可忽略。 该标志表示该发送双缓冲区为空。在该发送双缓冲区中的数据被移动到发送移位寄存器, 且该双缓冲区为空时, 可将该位设置为 "1"。 通过将数据重新写入到该双缓冲区, 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区已满。 在接收工作已完成, 且所接收的数据被从该接收移位寄存器移动到接收双缓冲器之后, 该位变为 "1", 而读取该位即可将其变成 "0"。											
5	TXRUN	R	在传输标志中 0: 停止 1: 工作 该状态标志表示数据发送正在进行中。 <TXRUN>与<TBEMP>位可指示以下状态。 <table border="1" style="margin: 5px 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>发送中</td> </tr> <tr> <td rowspan="2">0</td> <td>1</td> <td>发送已完成</td> </tr> <tr> <td>0</td> <td>等待状态数据在发送缓冲区中</td> </tr> </tbody> </table>	<TXRUN>	<TBEMP>	状态	1	-	发送中	0	1	发送已完成	0	等待状态数据在发送缓冲区中
<TXRUN>	<TBEMP>	状态												
1	-	发送中												
0	1	发送已完成												
	0	等待状态数据在发送缓冲区中												
4	SBLEN	R/W	STOP 位(适用于 UART) 0: 1-位 1: 2-位 其可在 UART 模式下指定发送停止位的长度。 对于接收侧而言, 仅利用与<SBLEN>设置无关的单个位即可做出该判定。											
3	DRCHG	R/W	设置传送方向 0: LSB 先 1: MSB 先 指定 I/O 接口模式下的数据传送方向。 在 UART 模式下, 首先将该位设置为 LSB。											
2	WBUF	R/W	双缓冲区 0: 禁用 1: 启用 该参数可启用或禁用该发送/接收双缓冲器在 I/O 接口模式下的数据发送(在 SCLK 输出/输入模式下)和接收(在 SCLK 输出模式下), 以及 UART 模式下的数据发送。 在接口模式(SCLK 输入)与 UART 模式下接收数据时, 双缓冲在 0 或 1 被设置到为<WBUF>位时均会启用。											

位	比特符号	类型	功能										
1-0	SWRST[1:0]	R/W	软件复位 若盖写"01"而不是"10", 则可引发软件复位。当执行该软件复位时, 以下位均初始化, 且传输接收电路成初始状态(注 1)(注 2)。 <table border="1" data-bbox="491 331 1134 555"> <thead> <tr> <th>寄存器</th> <th>位</th> </tr> </thead> <tbody> <tr> <td>SCxMOD0</td> <td><RXE></td> </tr> <tr> <td>SCxMOD1</td> <td><TXE></td> </tr> <tr> <td>SCxMOD2</td> <td><TBEMP>, <RBFLL>, <TXRUN></td> </tr> <tr> <td>SCxCR</td> <td><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, 任何软件复位工作均必须连续执行两次。

注 2: 单次软件复位需花费 2 段时钟持续时间均位于软件复位指令识别结束时间点和执行开始时间点之间。

11.3.8 SCxBRCR (波特率发生器控制寄存器), SCxBRADD (波特率发生器控制寄存器 2)

可在以下所列的寄存器中指定波特率发生器的分频比。

SCxBRCR

	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(适用于UART) 0: 禁用 1: 启用 该分频功能仅可用于UART模式。
5-4	BRCK[1:0]	R/W	选择波特率发生器的输入时钟。 00: $\phi T1$ 01: $\phi T4$ 10: $\phi T16$ 11: $\phi T64$
3-0	BRS[3:0]	R/W	分频比 "N" 0000: 16 0001: 1 0010: 2 ... 1111: 15

SCxBRADD

	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	为" $N + (16 - K)/16$ "分频(适用于UART) 0000: 禁止 0001: K = 1 0010: K = 2 ... 1111: K = 15

表 11-1 列出了波特率发生器分频比的各种设置。

表 11-1 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	除式 $N + \frac{(16 - K)}{16}$

注 1: 在使用" $N + (16 - K)/16$ "分频功能之前, 务必在将K值设置到<BRK>之后将<BRADDE>设置为"1"。" $N + (16 - K)/16$ "分频功能仅可用于UART模式。

注 2: 在UART模式下使用" $N + (16 - K)/16$ "分频功能时, 不得将作为分频比使用的 1 ("0001")或16 ("0000") 应用于N。

注 3: 仅在双缓冲被用于I/O接口模式时, 才可指定该波特率发生器的分频比"1"。

注 4: 禁止指定"K = 0"。

11.4 各模式下的工作

表 11-2 给出了各模式和各数据格式。

表 11-2 模式和数据格式

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，并可用于扩展I/O。该模式可与SCLK同步发送和接收数据。SCLK既用于输入，也用于输出。

可从LSB先和MSB先中选择数据传送方向。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，传送方向固定为LSB先。

在模式 1 和模式 2 下，可添加校验位。模式 3 有唤醒功能，其中的主设备控制器可通过串行链路(多控制器系统)启动从设备控制器。

可从 1 位和 2 位中选择传输中的STOP位。接收中的STOP位长度固定为一位。

11.5 数据格式

11.5.1 数据格式列表

图 11-2 显示数据格式。

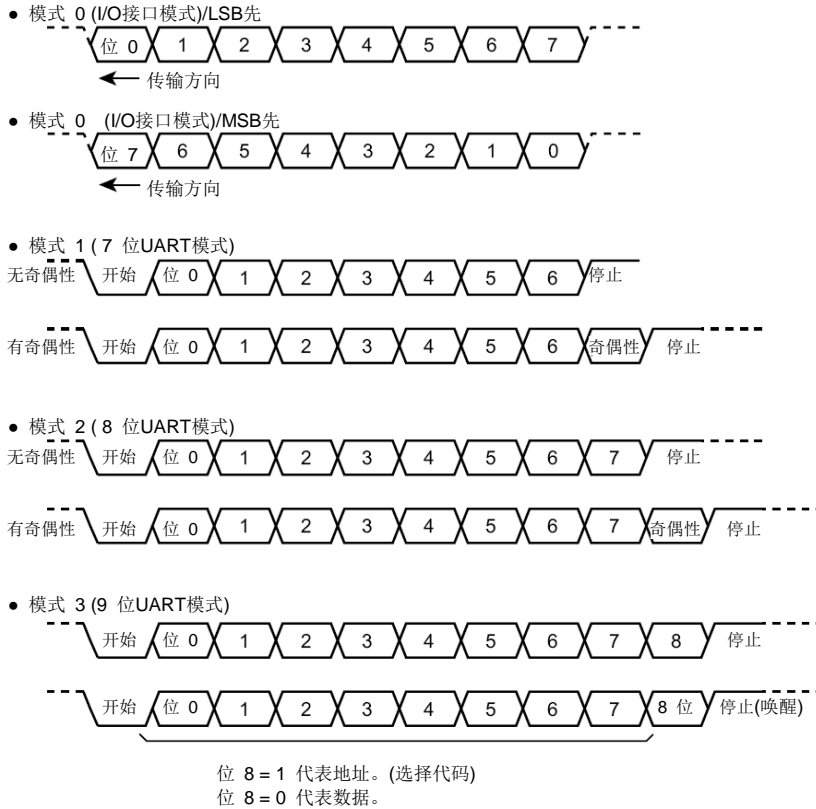


图 11-2 数据格式

11.5.2 奇偶校验控制

在 7-或 8-位 UART 模式下，仅可用发送数据新增奇偶校验位。

将 SCxCR<PE>设为"1"就可启用奇偶性。

SCxCR 的<EVEN>位选择偶数奇偶性或奇数奇偶性。

11.5.2.1 传输

在数据发送后，奇偶控制电路自动生成奇偶性，数据位于传输缓冲区。

奇偶校验位会被存储在 SCxBUF<TB7>中适用于 7-位模式或 SCxMOD<TB8>中适用于 8-位 UART 模式。

<PE>和<EVEN> 的设置必须在数据写入传输缓冲区之前完成。

11.5.2.2 接收数据

若接收的数据从接收移位寄存器移到接收缓冲区，则会生成奇偶性。

在 7-位 UART 模式中，生成的奇偶校验与储存在 SCxBUF<RB7>中的奇偶校验进行比较，在 8-位 UART 模式中，则与储存在 SCxCR<RB8>中的奇偶校验进行比较。

如有任何差异，则发生奇偶校验误差，且 SCxCR<PERR>设置到"1"。

11.5.3 STOP 位长度

在 UART 发送模式下，可通过设置 SCxMOD2<SBLLEN>，从一位或两位中选择 STOP 位的长度。在接收时，即可将 STOP 位数据的长度确定为一位，且不考虑该位的设置。

11.6 时钟控制

下图给出了串行时钟(SIOCLK)生成电路。在改变该串行时钟设置之前，应检查该设置是否满足AC电气特性。

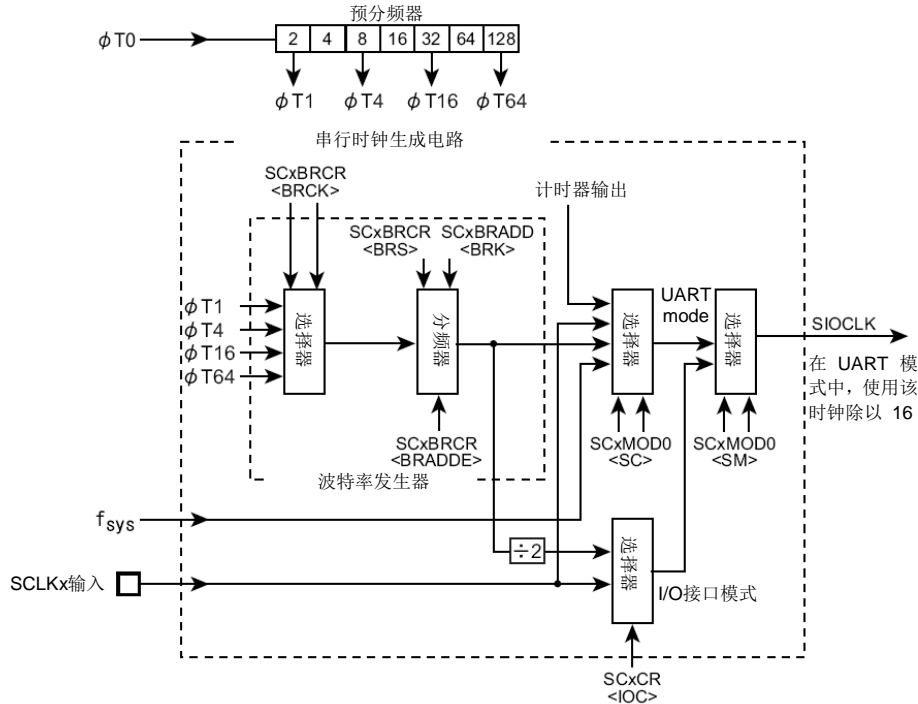


图 11-3 串行时钟生成电路

11.6.1 预分频器

把预分频器输入时钟 $\phi T0$ 除以 2, 8, 32 和 128 的 7-位预分频器。

用时钟/模式控制块中的 CGSYSCR 寄存器选择预分频器的输入时钟 $\phi T0$ 。

只有当波特率发生器由 $SCxMOD0<SC[1:0]> = "01"$ 选为传送时钟时，预分频器才能被激活。

11.6.2 串行时钟生成电路

串行时钟电路是一个存储块，用以生成传输与接收时钟(SIOCLK)，并由若干电路组成(在这些电路中可通过设置波特率发生器和模式来选择时钟)。

11.6.2.1 波特率发生器

波特率发生器生成传输和接收时钟，以确定串行通道传输率。

(1) 波特率发生器输入时钟

将预分频器输出除以 2, 8, 32 和 128，从所得结果中选择波特率发生器输入时钟。

通过设置 $SCxBRCR<BRCK>$ ，便可选择该输入时钟。

(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。

可使用的频率标度比如下所示：I/O 接口模式下为 1/N 分频，UART 模式下为 1/N 或 $N + (16-K)/16$ 分频。

可选择分频比如下表所示。

模式	分频功能设置 SCxBRCR<BRADDE>	由N分频 SCxBRCR<BRS>	由K分频 SCxBRADD<BRK>
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：仅可在双缓冲器已启用时采用 1/N (N=1) 分频比。

波特率发生器的分频器输入时钟为 ϕTx ，在 1/N 与 $N+(16-K)/16$ 情况下，波特率显示如下。

- 除以 N

$$\text{波特率} = \frac{\phi Tx}{N}$$

- $N+(16-K)/16$ 分频

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

11.6.2.2 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>指定模式。

通过设置 SCxCR 选择在 I/O 接口模式时的输入时钟。通过设置 SCxMOD0<SC>，便可选择在 UART 模式时的时钟。

(1) I/O 接口模式下的传送时钟

表 11-3 给出了时钟选择 I/O 接口模式下的时钟选择方式。

在启用 TMR16A 输出时，计时器输出可在计数器的值与 T16AxRG 的值匹配时反转。SIOCLK 时钟频率等于“T16AxRG 的设置值 × 2”。

波特率由下述公式计算所得。

$$\text{传送率} = \frac{\text{由 CGSYSCR<PRCK[1:0]>选择的时钟频率}}{(\text{TBxRG1} \times 2) \times 16}$$



一个时钟周期，指计时器输出可在其中反转两次的单个时段。

11.6.3 传输/接收缓冲器

11.6.3.1 配置

图 11-4 给出了传输缓冲器和接收缓冲器的配置。

在使用缓冲器之前，必须对其进行适当的设置。可按照模式对配置进行预定义。



图 11-4 缓冲器的配置

11.6.3.2 传输/接收缓冲器

对传输缓冲器和接收缓冲器进行双缓冲。缓冲器配置由 SCxMOD2<WBUF>指定。

如果是接收模式，如已在 I/O 接口模式下设置 SCLK 输入或已选择 UART 模式，则不论<WBUF>设置为何，其均属于双缓冲型。在其它模式下，其处理应按<WBUF>设置进行。

表 11-5 给出了各模式和缓冲器之间的相互关系。

表 11-5 模式和缓冲器组成

模式		SCxMOD2<WBUF>	
		"0"	"1"
UART	传输	单	双
	接收	双	双
I/O接口 (SCLK输入)	传输	单	双
	接收	双	双
I/O接口 (SCLK输出)	传输	单	双
	接收	单	双

11.7 状态标志

SCxMOD2 寄存器有两类标志。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"，而读取该位会使其变为"0"。

<TBEMP>表示传输缓冲器为空。在传输缓冲器中的数据被移到该传输移位寄存器时，该位即被设置为"1"。在数据即被设置到该传输缓冲器时，该位即被清除为"0"。

11.8 错误标志

在SCxCR寄存器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，这些标志被清除到"0"。

模式	标志		
	<OERR>	<PERR>	<FERR>
UART	欠载错误	奇偶错误	成帧错误
I/O接口 (SCLK输入)	欠载错误	欠程错误 (在使用双缓冲器时)	规定值为 0
		被固定为 0 (在未使用双缓冲器时)	
I/O接口 (SCLK输出)	未定义	未定义	规定值为 0

11.8.1 OERR 标志

在 UART 和 I/O 接口模式下，在接收缓冲器被读取之前，下一接收数据帧的接收完成导致发生错误时，该位即被设置为"1"。

带 SCLK 输出的 I/O 接口模式下，SCLK 输出在设置标志后停止。

注：在从I/O接口SCLK输出模式切换到其它模式时，应读取SCxCR寄存器，并清除该欠载标志。

11.8.2 PERR 标志

此标志表示 UART 模式中的奇偶校验错误和 I/O 接口模式中的欠载错误或传输完成。

UART 模式下，当接收的数据产生的奇偶性不同于接收的奇偶性时，将<PERR>设至"1"。

在 I/O 接口模式下，在双缓冲器被启用时，<PERR>会在以下条件下被设置为"1"。

SCLK 输入模式下，在完成传输移位寄存器的数据输出，并且在传输缓冲器中无数据后，当输入 SCLK 时，将<PERR>设至"1"。

在 SCLK 输出模式下，在完成所有数据的输出后，将<PERR> 设至"1"，SCLK 输出停止。

注：在将I/O接口SCLK输出模式切换到其它模式时，需读取SCxCR寄存器并清除该欠载标志。

11.8.3 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。在不考虑 SCxMOD2<SBLEN>寄存器中的 STOP 位长度设置的情况下，仅按 1 即可确定停止位状态。

I/O 接口模式下，该位固定至"0"。

11.9 接收

11.9.1 接收计数器

接收计数器为 4-位二进制计数器，并由 SIOCLK 递增计数。

在 UART 模式下，十六个 SIOCLK 时钟脉冲可被用于接收单个数据位，且数据符号的采样在第七、第八和第九个脉冲时进行。从这三个样本中，多数逻辑用于决定接收的数据。

11.9.2 接收控制单元

11.9.2.1 I/O 接口模式

在 SCLK 输出模式下，且 SCxCR <IOC> 被设置为 "0" 时，RXD_x 引脚的采样在被输出到 SCLK_x 引脚的移位时钟的上升沿进行。

在 CLK 输入模式下，且 SCxCR <IOC> 被设置为 "1" 时，串行接收数据 RXD_x 引脚的采样在 SCLK_x 引脚输入信号的上升或下降沿进行，视 SCxCR <SCLKS> 设置而定。

11.9.2.2 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工作。

11.9.3 接收工作

11.9.3.1 接收缓冲器

接收的数据按 1 位储存在接收移位寄存器中。在整组位已被存储时，中断 INTRX_x 即被生成。

在该双缓冲器被启用时，数据会被移动到接收缓冲器 (SCxBUF)，且接收缓冲器已满标志 (SCxMOD2 <RBFL>) 即被设置为 "1"。通过读取接收缓冲器，接收缓冲器全满标志被清除到 "0"。当双缓冲中止后，接收缓冲区标志则没有任何意义。

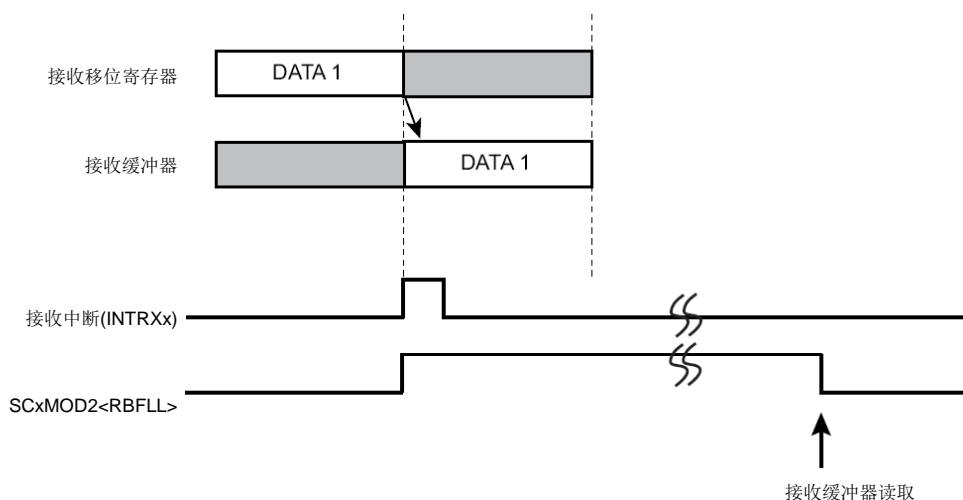


图 11-5 接收缓冲区工作

11.9.3.2 带有 SCLK 输出的 I/O 接口模式

在 I/O 接口模式下，且 SCLK 输出已设置的情况下，SCLK 输出会在所有已接收数据均被存储到该接收缓冲器中时停止。所以，在这种模式下，过运行错误标志没有意义。

SCLK 输出停止和重新输出的计时，取决于该接收缓冲器。

(1) 单缓冲器的情况

在接收数据后，停止 SCLK 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，SCLK 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 SCLK 输出。当读取数据时，SCLK 输出被重启。

11.9.3.3 读取所接收的数据

从该接收缓冲器(SCxBUF)读取所接收的数据。

通过该次读取，该缓冲器已满标志 SCxMOD2<RBFL>即被清除为"0"。在这种情况下，在读取接收缓冲器的数据前，在接收移位寄存器中能接收下一数据。拟在 8-位 UART 模式下新增的奇偶校验位，以及 9-位 UART 模式下的最高有效位，将被存储到 SCxCR<RB8>中。

11.9.3.4 唤醒功能

在 9-位 UART 模式中，从控制器可通过将唤醒功能 SCxMOD0<WU>设置至"1"，以唤醒模式工作。在这种情况下，只有当 SCxCR <RB8>设至"1"时，才会产生 INTRX_x。

11.9.3.5 溢出错误

此时会出现溢出错误，需在接收下一数据之前，在数据读取未完成的情况下进行超程设置。在发生溢出错误时，接收缓冲器和 SCxCR<RB8>的内容不会丢失，但接收移位寄存器的内容会丢失。带 SCLK 输出设置的 I/O 接口模式下，时钟输出自动停止，因此该标志无意义。

注：在从 I/O 接口 SCLK 输出模式变为另一模式时，会读取 SCxCR 并清除溢出标志。

11.10 传输

11.10.1 传输计数器

传输计数器为 4-位二进制计数器，和在接收计数器的情况一样，由 SIOCLK 计数。UART 模式下，它在每第 16 个时钟脉冲时生成传输时钟(TXDCLK)。



图 11-6 传输时钟的生成

11.10.2 传输控制

11.10.2.1 I/O 接口模式

在 SCLK 输出模式下，且 SCxCR<IOC>已被设置为"0"时，传输缓冲器中的各数据位会被输出到被从 SCLKx 引脚输出的移位时钟下降沿上的 TXDx 引脚。

在 SCLK 输入模式下，且 SCxCR<IOC>已被设置为"1"时，传输缓冲器中的各数据位会按照 SCxCR<SCLKS>设置，被输出到 SCLKx 引脚输入信号的上升沿或下降沿上的 TXDx 引脚。

11.10.2.2 UART 模式

当传输数据写入传输缓冲器时，数据传输在下一 TXDCLK 的上升沿启动，并且也生成传输移位时钟信号。

11.10.3 传输工作

11.10.3.1 传输缓冲器的工作

如果双缓冲已被禁用，则 CPU 只会将数据写入到传输移位寄存器，且会在数据传输完成时生成传输中断 INTTXx。

如果双缓冲被启用，则被写入到该传输缓冲器的数据会被移到该传输移位寄存器。同时，产生 INTTXx 中断，传输缓冲器空标志(SCxMOD2<TBEMP>)设为"1"。该标志表示能写入下一传输数据。当下一数据写入传输缓冲器时，<TBEMP>标志被清除到"0"。

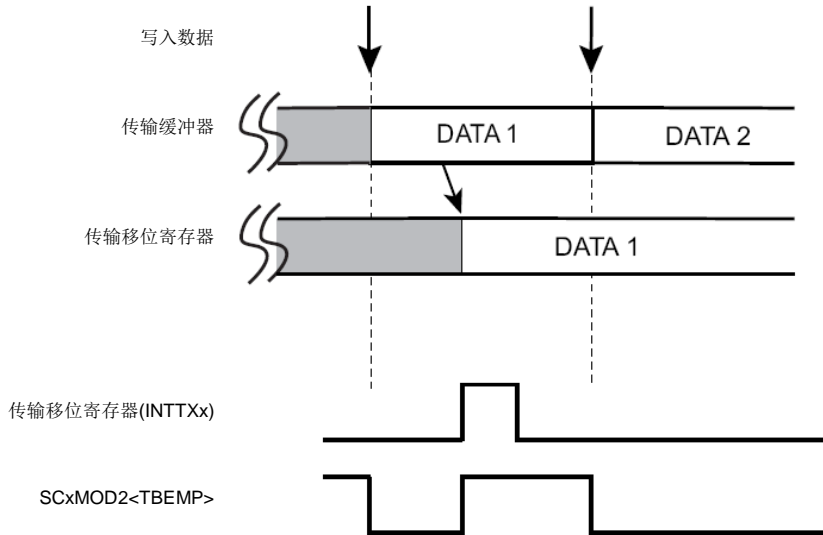


图 11-7 传输缓冲器的工作(双缓冲器已被启用)

11.10.3.2 I/O 接口模式/通过 SCLK 输出进行传输

若为了在 I/O 接口模式时生成时钟而设置 SCLK，则当所有数据传输完成时，SCLK 输出自动停止，不会发生欠载错误。

挂起计时和 SCLK 输出的恢复会有所不同，视缓冲器和 FIFO 使用而定。

(1) 单缓冲器

每当转移一帧数据时，SCLK 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，SCLK 输出恢复。

(2) 双缓冲器

在传输移位寄存器和传输缓冲器的数据传输完成后，SCLK 输出停止。当下一数据写入缓冲器时，SCLK 输出恢复。

11.10.3.3 欠载运行错误

在 I/O 接口 SCLK 输入模式下，且在下一帧时钟输入之前未在传输缓冲器中设置任何数据时其一般在源自传输移位寄存器的数据传输完成时即发生，会发生欠载错误，且 SCxCR<PERR>会被设置为"1"。

带 SCLK 输出设置的 I/O 接口模式下，时钟输出自动停止，因此该标志无意义。

注：在将 I/O 接口 SCLK 输出模式切换到其它模式之前，读取该 SCxCR 寄存器，并清除欠载标志。

11.11 握手功能

信号交换功能通过使用CTS (清除后传输)针来启用一帧接一帧的数据传输功能，并且避免溢出错误。功能由SCxMOD0<CTSE>启用或禁用。

在 \overline{CTSx} 引脚被设置为“高”电平时，当前数据传输可完成，但下一次数据传输会被暂停，直至该 \overline{CTSx} 引脚返回到“低”电平。然而在这种情况下，在正常时间生成INTTXx中断，下一传输数据写入传输缓冲器，并且它会等到准备传输数据时为止。

- 注 1: 如果该 \overline{CTS} 信号在传输期间被设置为“H”，则在当前传输完成之后，下一次数据传输即被暂停(图 11-9 中的点A)。
- 注 2: 在 \overline{CTS} 设定为“L”后，数据传输开始于TXDCLK 时钟的第一下降沿(图 11-9 中的点B)。

虽然未设有 \overline{RTS} 引脚，但是通过给端口的一位分配 \overline{RTS} 功能，就能轻易地实施握手控制功能。通过在数据接收完成时将该端口设置为“高”电平(在中断程序该接收中断程序中)，可请求该传输侧暂停数据传输。

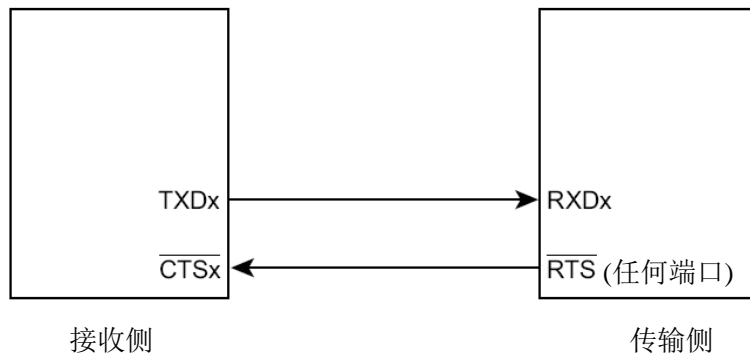


图 11-8 握手功能

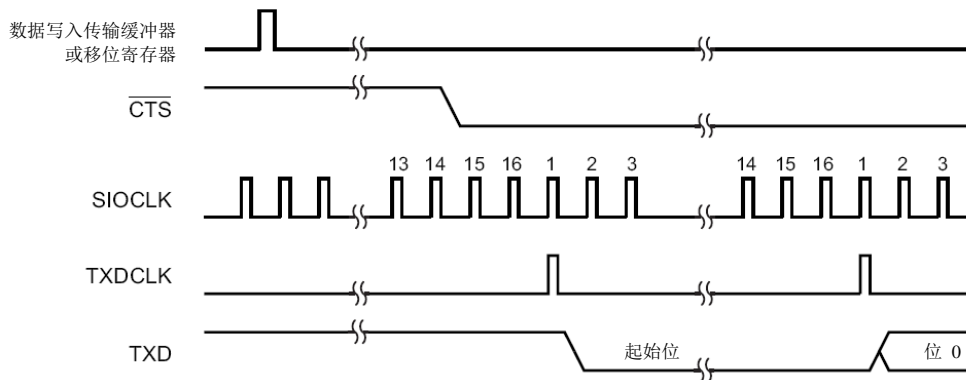


图 11-9 \overline{CTS} 信号时序

11.12 使用 IR 载波脉冲输出信号

IROUT引脚利用IR载波脉冲输出信号。

由计时器输出信号和TXD信号生成IROUT信号。由于用于生成脉冲的该计时器可随产品不同而不同，在选择计时器时请参照产品信息一节。

计时器经设置后，应在可设置的范围内，按所要求的周期生成脉冲输出。

在从某端口其中的IROUT引脚已分配输出IROUT信号时，应将功能寄存器(PxFRn)的功能设置指定给IROUT，然后启用拟由控制寄存器(PxCR)输出的IROUT信号。

SIO从计时器接收脉冲，并将该脉冲与TXD信号结合，向IROUT引脚输出一个信号。计时器输出和TXD信号并非同步，因此，在TXD信号被改变时会发生脉冲畸变。

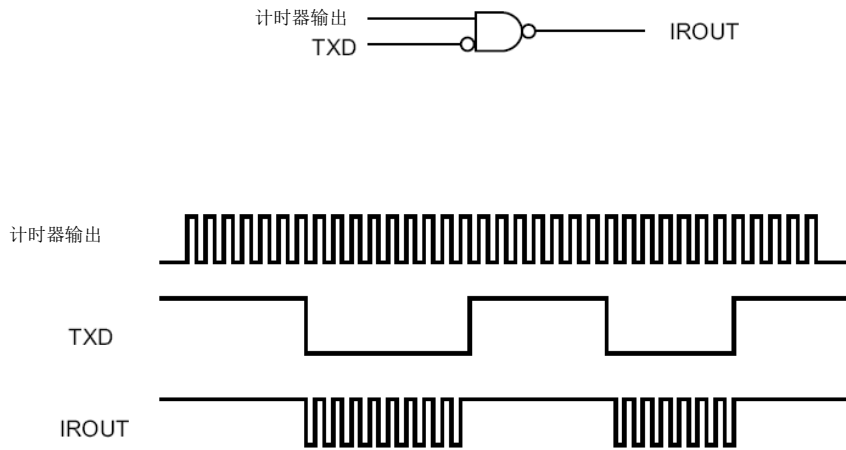


图 11-10 使用IR载波脉冲输出信号

11.13 中断/错误生成时序

11.13.1 RX 中断

图 11-11 给出了接收工作的数据流和读取路径。

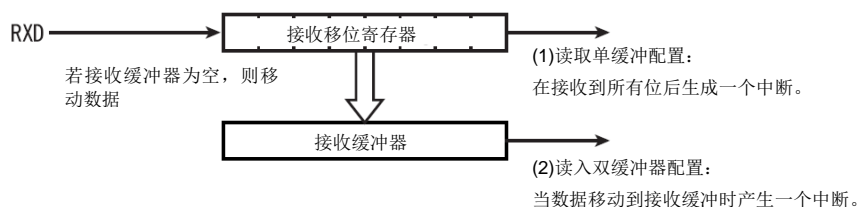


图 11-11 接收缓冲器配置图

在传送模式和缓冲器配置所确定的时间，生成 RX 中断，如下。

缓冲器配置	UART 模式	IO 接口模式
单缓冲器	-	· 紧跟在最后一个 SCLK 的上升/下降沿 (上升或下降按 SCxCR<SCLKS>设置确定)之后
双缓冲	· 首个停止位的中心周围	· 紧跟在最后一个 SCLK 的上升/下降沿 (上升或下降按 SCxCR<SCLKS>设置确定)之后 · 在通过读数据缓冲器从移位寄存器将数据传送到缓冲器时。

注：当发生溢出错误时不会生成中断。

11.13.2 TX 中断

图 11-12 给出了传输工作的数据流和读取路径。

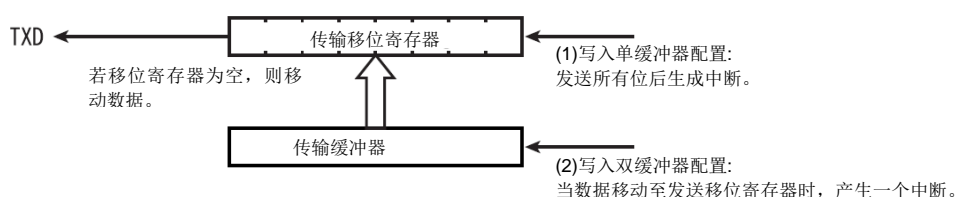


图 11-12 发送缓冲器配置图

如下表所示，在传输模式和缓冲器配置所确定的时间，生成 TX 中断。

缓冲器配置	UART 模式	IO 接口模式
单缓冲器	恰在传输停止位前	紧跟在最后一个 SCLK 的上升/下降沿之后 (上升或下降按照 SCxCR<SCLKS>的设置确定。)
双缓冲	当数据从传输缓冲器移到传输移位寄存器时。	

注：在启用双缓冲器时，当通过写入缓冲器将数据从缓冲器移动到移位寄存器的时候同样会生成中断。

11.13.3 错误生成

11.13.3.1 UART 模式

模式	9 位	7 位 8 位 7 位 + 奇偶校验 8 位 + 奇偶校验
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

11.13.3.2 I/O 接口模式

溢出错误	紧跟在最后一个SCLK的上升/下降沿之后 (上升或下降按照SCxCR<SCLKS>的设置确定。)
欠载错误	恰在下一SCLK上升沿或下降沿后。 (上升或下降按照SCxCR<SCLKS>的设置确定。)

注：在SCLK输出模式中溢出错误和欠载错误没有任何意义

11.14 软件复位

通过将SCxMOD2<SWRST[1:0]>写入为"10"其后带有"01"，即可生成软件复位。

软件复位后，SCxMOD0<RXE>， SCxMOD1<TXE>， SCxMOD2<TBEMP><RBFLL><TXRUN>， SCxCR<OERR><PERR><FERR>被初始化。接收电路和传输电路为初始状态。其它状态则被保持。

11.15 各模式下的工作

11.15.1 模式 0 (I/O 接口模式)

模式 0 由两种模式组成，即输出同步时钟的 SCLK 输出模式和接收外部来源的同步时钟的 SCLK 输入模式。

11.15.1.1 数据传输

(1) SCLK 输出模式

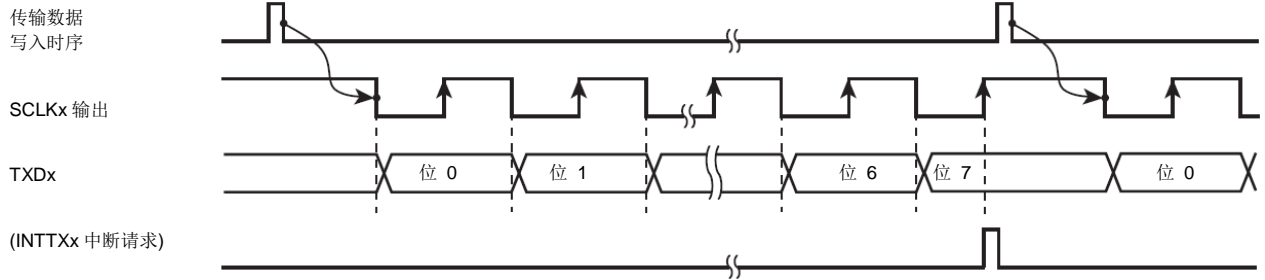
- 如果该发送双缓冲器已被禁用(SCxMOD2<WBUF> = "0")

数据从 TXD_x 引脚输出，每次 CPU 向传输缓冲器写入数据时，时钟从 SCLK_x 引脚输出。当所有数据被输出时，生成中断(INTTX_x)。

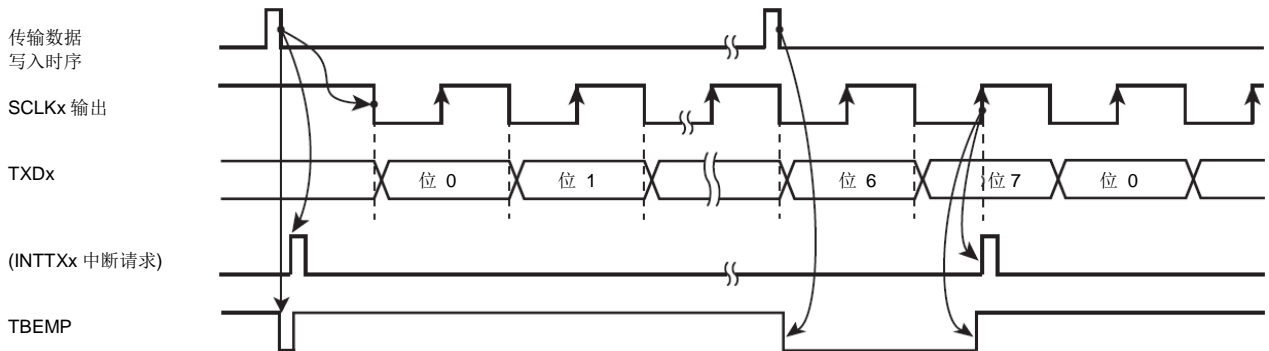
- 如果发送双缓冲器已启用(SCxMOD2<WBUF> = "1")

在 CPU 向传输缓冲器写入数据时(数据发送已被暂停，或当传输缓冲器(移位寄存器)的数据传输完成时)，数据会从传输缓冲器移至传输移位寄存器。同时，传输缓冲器空标志 SCxMOD2<TBEMP> 设为"1"， INTTX_x 中断生成。

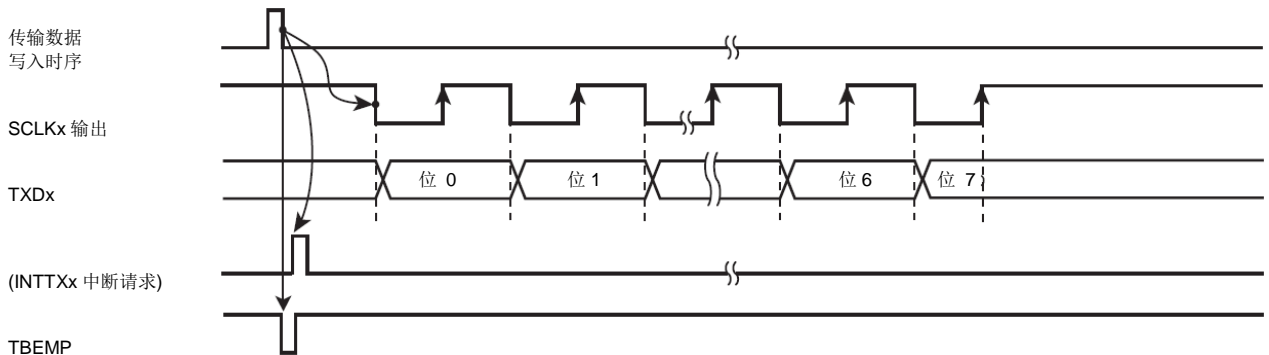
在数据被自传输缓冲器移至传输移位寄存器时，如果传输缓冲器无数据移至传输移位寄存器，则不会生成 INTTX_x 中断，SCLK 输出即停止。



$\langle WBUF \rangle = "0"$ (当双缓冲禁用时)



$\langle WBUF \rangle = "1"$ (当双缓冲启用, 且缓冲器中有数据时)



$\langle WBUF \rangle = "1"$ (当双缓冲启用, 且缓冲器中无数据时)

图 11-13 I/O接口模式(SCLK输出模式)下的传输工作

(2) SCLK 输入模式

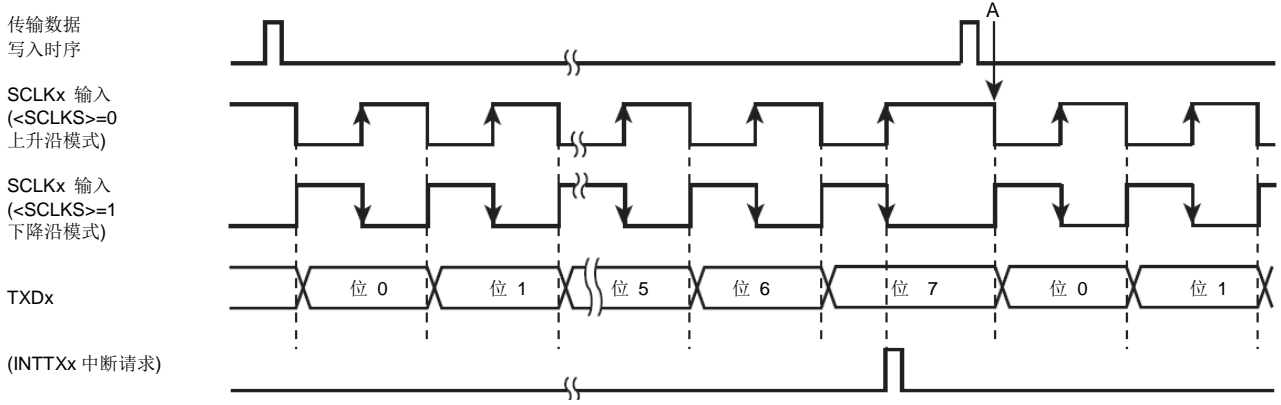
- 当双缓冲禁用时(SCxMOD2<WBUF> = "0")

当 SCLK 在数据已被写入传输缓冲器的情况下输入时，8-位数据从 TXD_x 引脚输出。全部数据输出后，中断 INTTX_x 即生成。必须将下一次的传输数据写入图 11-14 所示的计时点"A"之前。

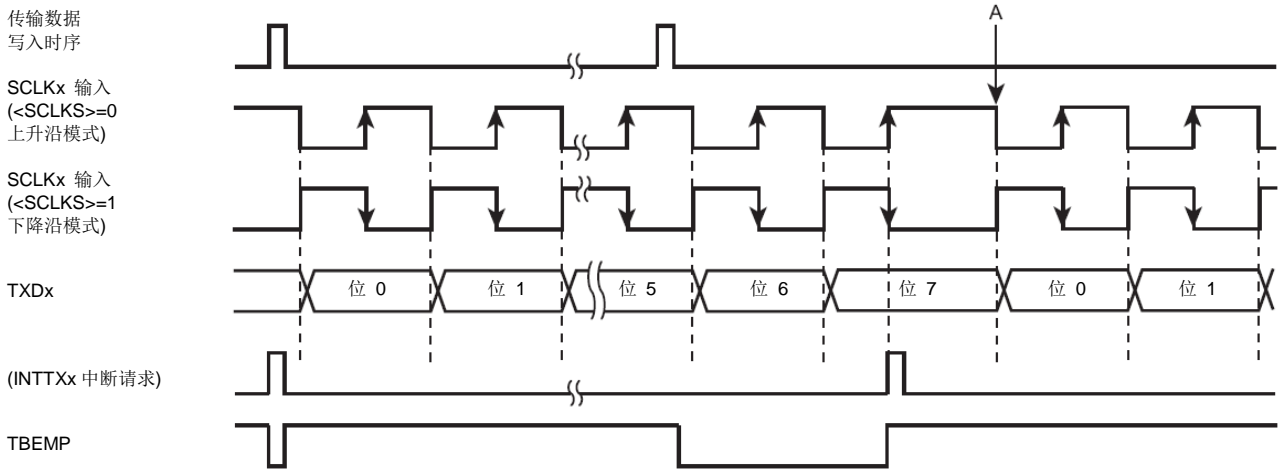
- 如果双缓冲器已被启用(SCxMOD2<WBUF> = "1")

当 CPU 在 SCLK 输入激活前，或在当传输移位寄存器的数据传输完成时向传输缓冲器写入数据时，数据会从传输缓冲器移至传输移位寄存器。同时，传输缓冲器空标志 SCxMOD2<TBEMP>设为"1"，INTTX_x 中断生成。

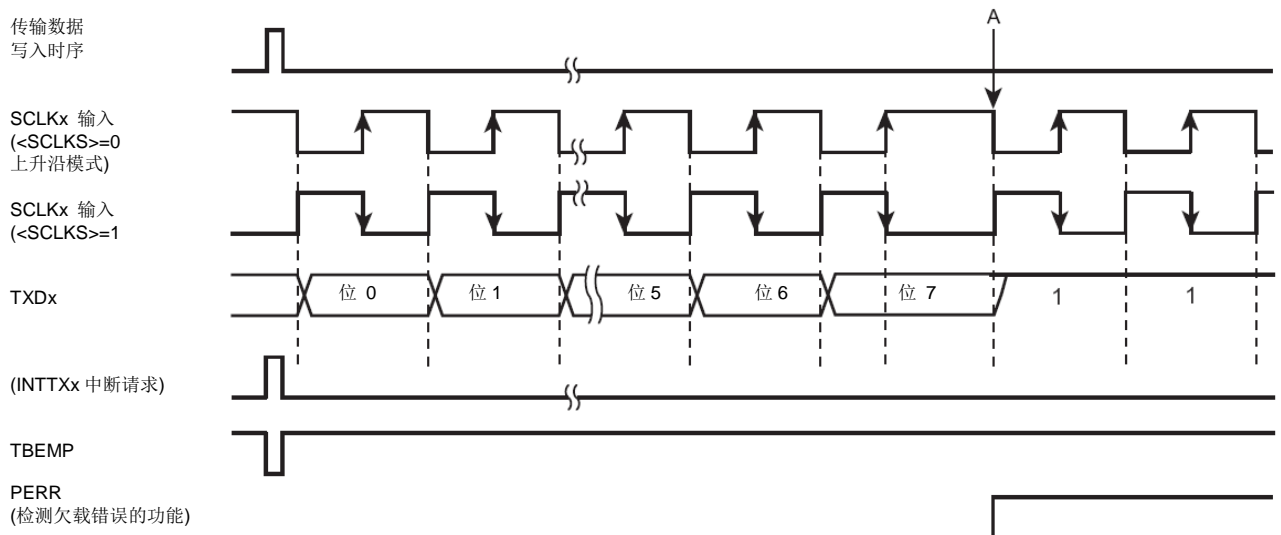
如果 SCLK 输入在发送缓冲器中无数据期间激活，则尽管内部计数器已被启动，仍会发生一个欠载错误，并会发送 8-位虚拟数据(0xFF)。



<WBUF> = "0" (若双缓冲禁用)



<WBUF> = "1" (如果双缓冲器已启用且缓冲器 2 中有数据)



<WBUF> = "1" (如果双缓冲器已启用且缓冲器 2 中没有数据)

图 11-14 I/O接口模式(SCLK输入模式)下的传输工作

11.15.1.2 接收

(1) SCLK 输出模式

将接收启用位 $SCxMOD0<RXE>$ 设为 "1", 可启用 SCLK 输出。

· 当双缓冲器禁用时($SCxMOD2<WBUF> = "0"$)

CPU 每读取一次所接收的数据, 就会有一个时钟脉冲从 SCLK_x 引脚输出, 且下一数据会存储到移位寄存器中。当接收到所有 8 位时, 生成 INTRX_x 中断。

· 如果双缓冲器已被启用($SCxMOD2<WBUF> = "1"$)

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至接收缓冲器, 接收缓冲器已满标志 $SCxMOD2<RBFL>$ 会被设置为 "1", INTRX_x 即被生成。

在数据位于接收缓冲器内期间, 如未能在下一个 8 位的接收完成之前从接收缓冲器读取数据, 则不会生成 INTRX_x 中断, 且 SCLK 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 INTRX_x 中断, 数据接收恢复。

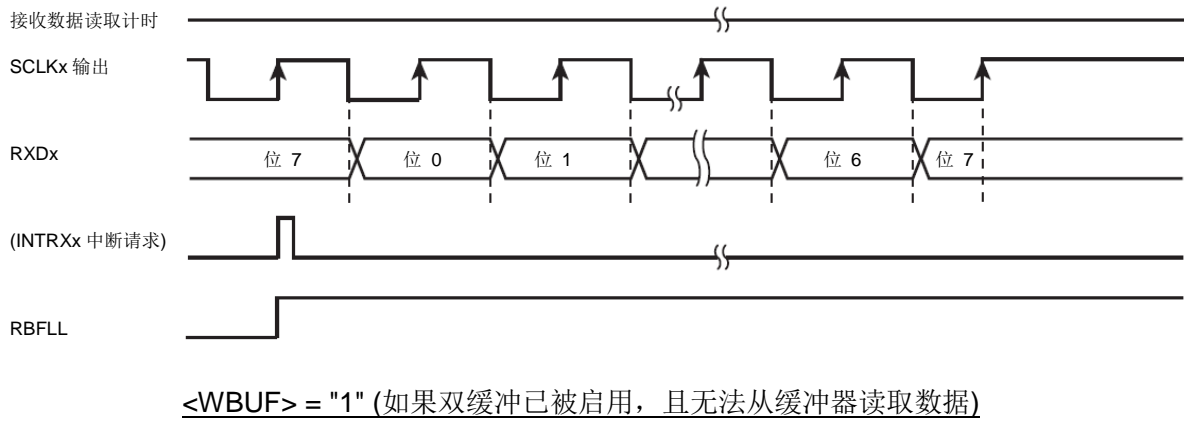
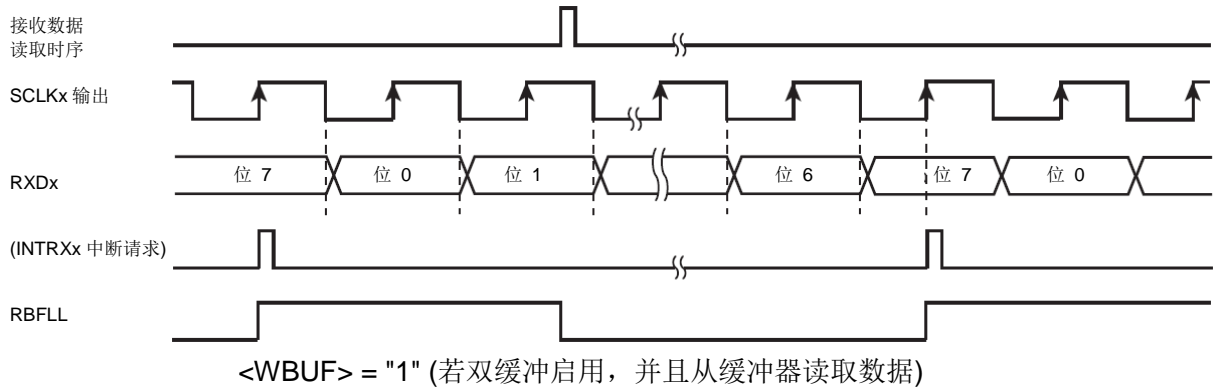
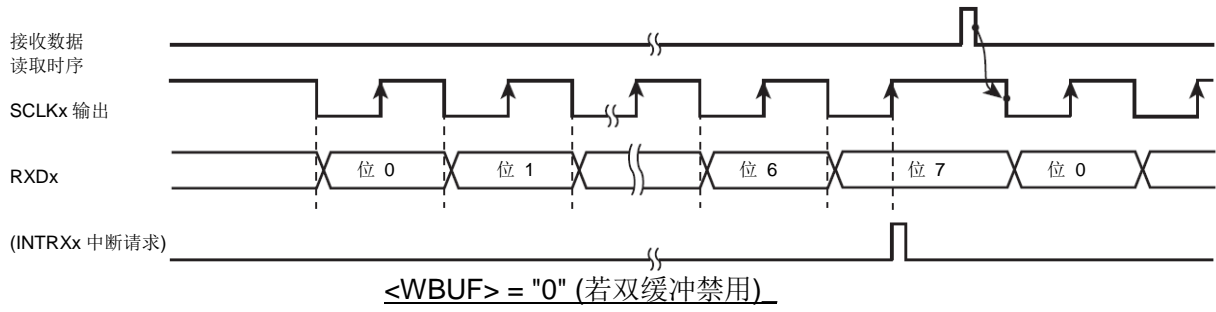
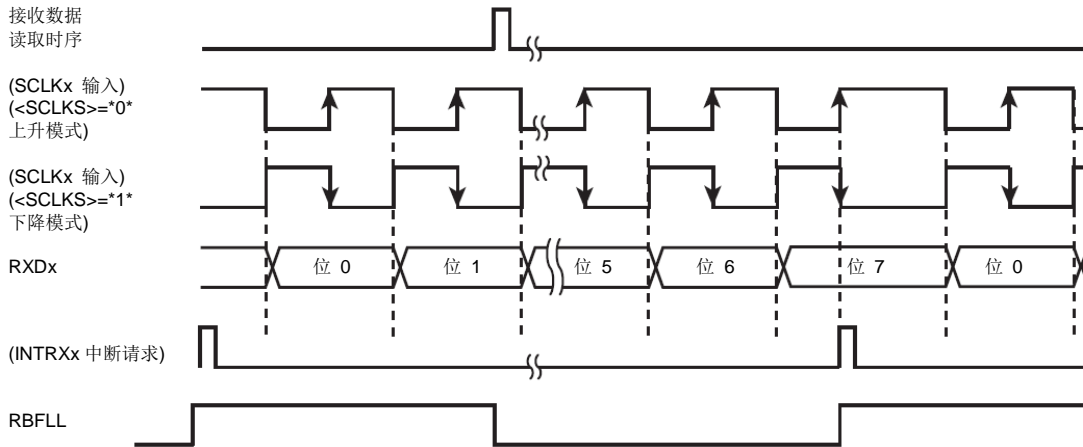


图 11-15 I/O接口模式(SCLK输出模式)下的接收工作

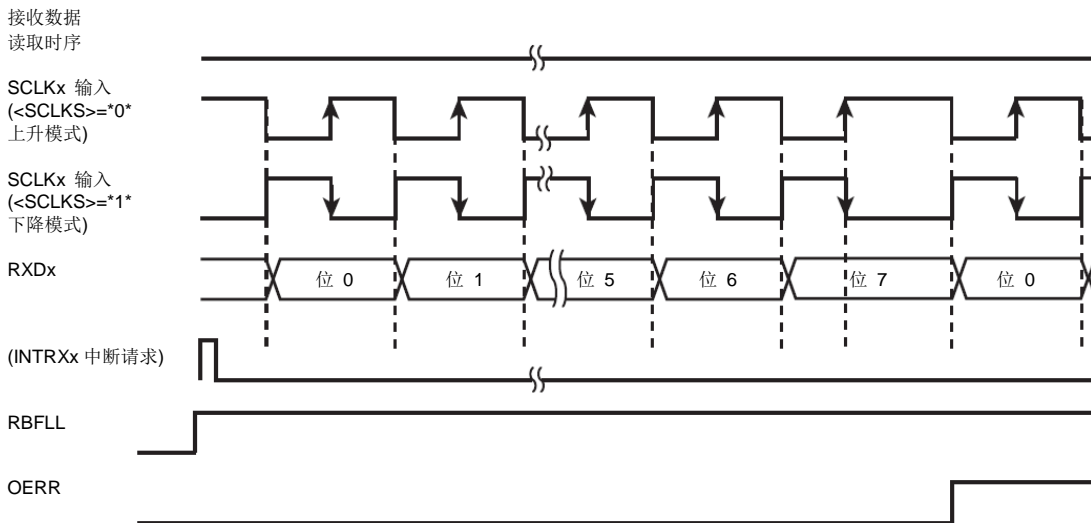
(2) SCLK 输入模式

在 SCLK 输入模式时，接收双缓冲始终启用，接收的帧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。

每一次将接收数据移至接收缓冲区时都会生成 INTRXx 中断。



若从缓冲器读取数据



若无法从缓冲器读取数据

图 11-16 I/O接口模式(SCLK输入模式)下的接收工作

11.15.1.3 传输和接收(全双工)

(1) SCLK 输出模式

- 如果双缓冲中止 (SCxMOD2<WBUF> = "0")

SCLK 当 CPU 把数据写入传输缓冲器时，输出 SCLK。

随后，数据中会有 8 位被移入接收缓冲器，INTRXx 接收中断生成。同时，写入到传输缓冲器的数据会有 8 位从 TXDx 引脚输出，在数据位的传输完成时，INTTXx 传输中断生成。然后，SCLK 输出停止。

在 CPU 从接收缓冲器读取数据，且将下一个传输数据写入到传输缓冲器时，下一轮的数据传输和接收开始。可自由确定从接收缓冲器读取，以及向传输缓冲器写入的次序。只有在满足这两个条件时，数据发送才能恢复。

- 当双缓冲启用(SCxMOD2<WBUF> = "1")时

SCLK 当 CPU 把数据写入传输缓冲器时，输出 SCLK。

8 位数据移至接收移位寄存器，至接收缓冲器，并生成 INTRXx 中断。当接收到 8 位数据后，从 TXDx 针输出 8 位传输数据。在全部数据位发送后，INTTXx 中断生成，且下一个数据从传输缓冲器移至传输移位寄存器。

若传输缓冲器无数据要移到传输缓冲器时(SCxMOD2<TBEMP> = "1")或者当接收缓冲器全满时(SCxMOD2<RBFULL> = "1")，SCLK 输出停止。在读取接收数据和写入传输数据等两个条件被满足后，SCLK 输出恢复，且下一轮数据传输和接收开始。

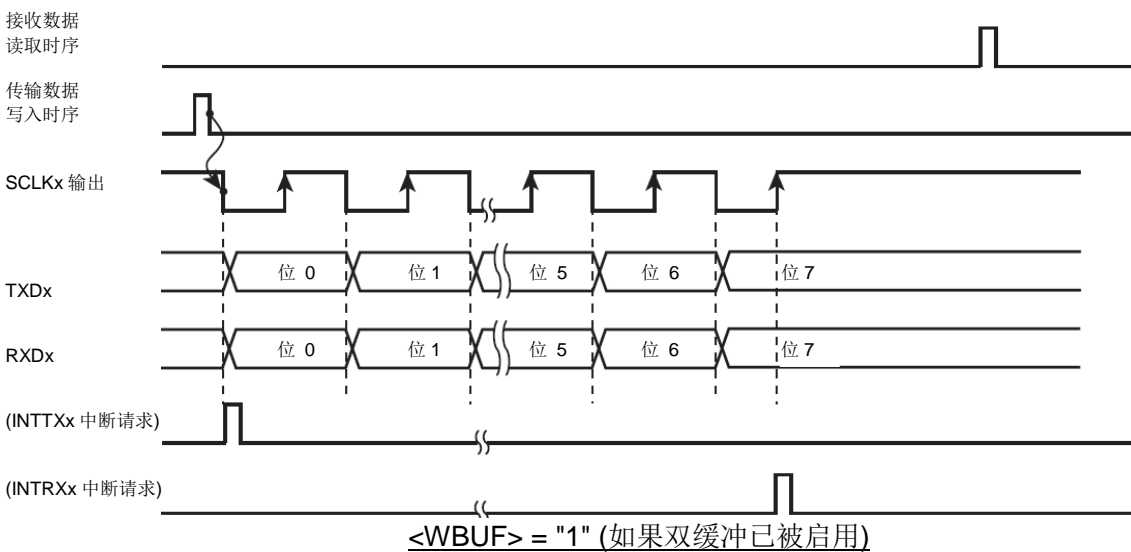
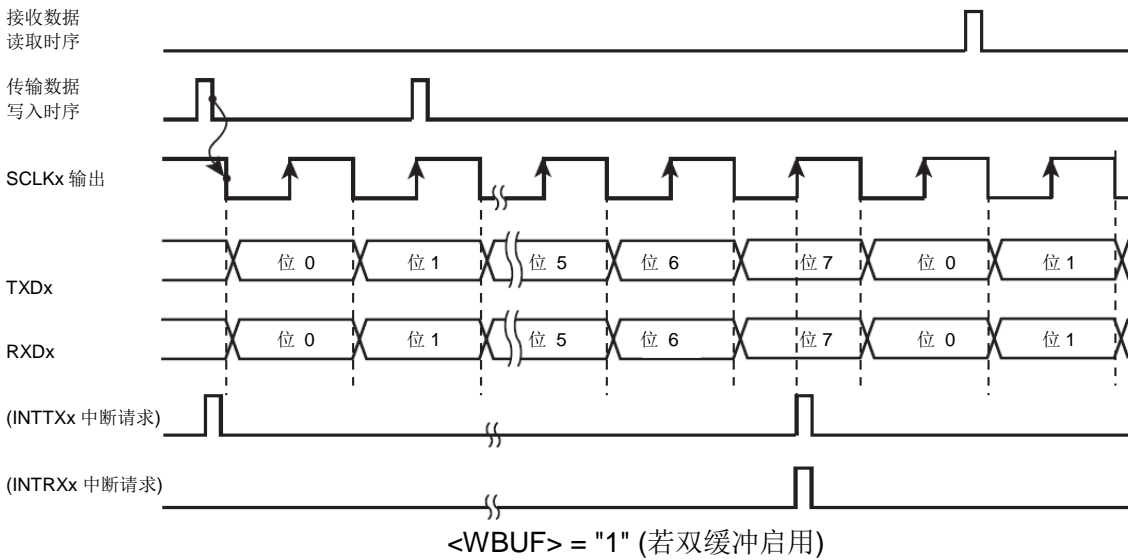
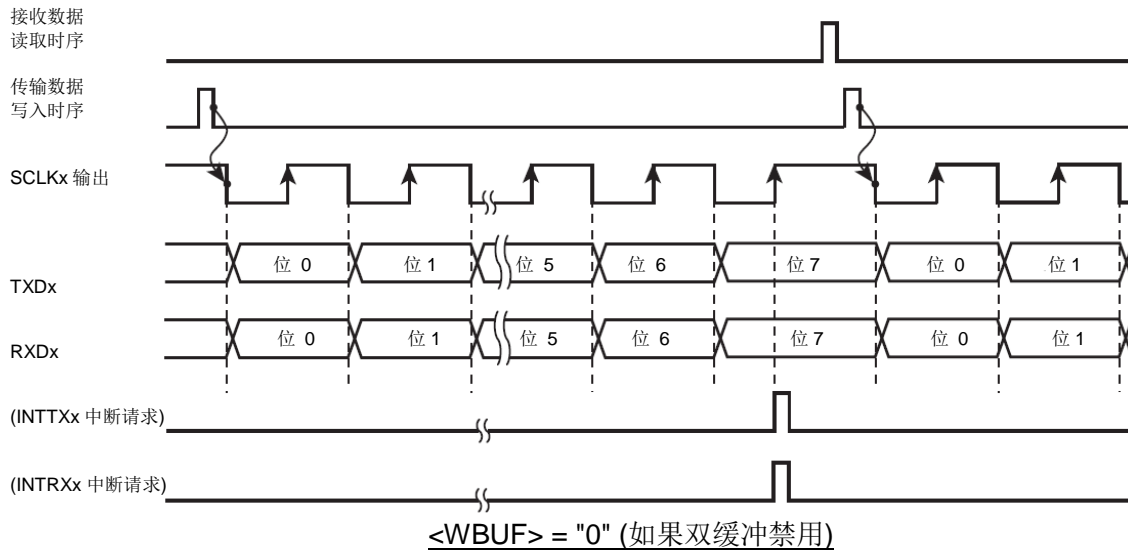


图 11-17 I/O接口模式(SCLK输出模式)下的传输/接收工作

(2) SCLK 输入模式

• 当双缓冲中止时。(SCxMOD2<WBUF> = "0")

当接收数据时，不考虑 SCxMOD2 <WBUF> 的设置，总是启用双缓冲。

在 SCLK 输入激活时，写入到传输缓冲器中的 8-位数据从 TXD_x 引脚输出，且数据中会有 8 位移入接收缓冲器。数据传输完成，INTTX_x 中断生成。数据接收完成时，在数据从移位寄存器移至接收缓冲器时生成 INTRX_x 中断。

注意必须在下一帧的 SCLK 输入之前将发送数据写入到传输缓冲器(必须在图 11-18 所示的点 A 之前写入数据)。在下一帧数据的接收完成前，必须读取数据。

• 当启用双缓冲时(SCxMOD2<WBUF> = "1")

在完成从传输移位寄存器的数据传输后，当传输缓冲区数据移动至传输移位寄存器时，产生中断 INTTX_x。同时，接收的数据移至移位寄存器，被移至接收缓冲器，INTRX_x 中断生成。

注意必须在下一帧的 SCLK 输入之前将传输数据写入到传输缓冲器(必须在图 11-18 所示的点 A 之前写入数据)。在下一帧数据的接收完成前，必须读取数据。

下一帧的 SCLK 输入完成时，启动源自欠载移位寄存器(其中的数据从传输缓冲器移出)的发送，接收数据同时移至接收移位寄存器中。

如果在接收帧的最后位时未读取接收缓冲器中的数据，则会发生溢位错误。同样，如果在下一帧的 SCLK 输入时无数据写入到传输缓冲器，则发生欠载错误，并输出虚拟数据(0xff)。

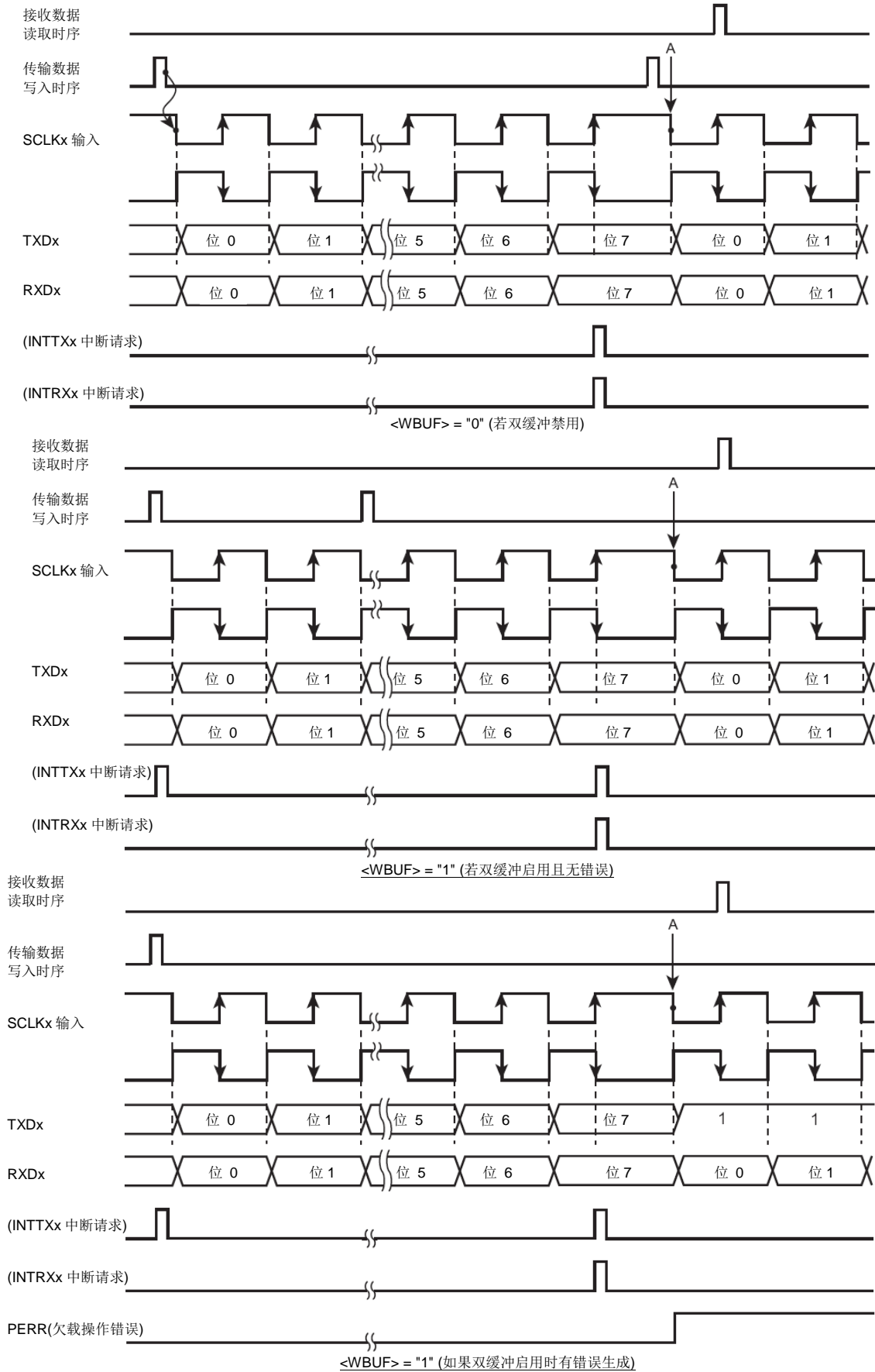


图 11-18 I/O接口模式(SCLK输入模式)下的传输接收工作



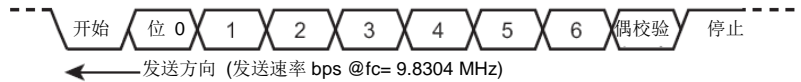
11.15.2 模式 1 (7-位 UART 模式)

通过设置模式控制寄存器 (SCxMOD<SM[1:0]>), 即可选择 7-位 UART 模式。

在模式下, 可将校验位添加到发送数据流; 串行模式控制寄存器(SCxCR<PE>)控制奇偶校验启用/禁用的设置。

当<PE>设为"1" (启用)时, 可利用 SCxCR<EVEN>位选择偶数或奇校验。用 SCxMOD2<SBLEN>指定停止位的长度。

以下列数据格式传输时, 控制寄存器的设置如下表如示。



时钟条件	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)

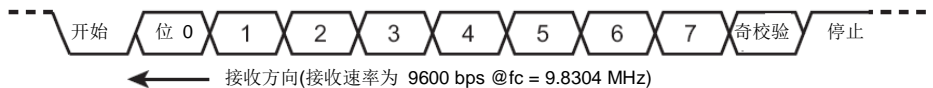
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	设置 7=位UART模式
SCxCR	←	x	1	1	x	x	x	0	0	偶校验启用
SCxBRCR	←	0	0	1	0	0	1	0	0	设置 2400 bps
SCxBUF	←	*	*	*	*	*	*	*	*	设置传输数据

x: 忽略-:无变化

11.15.3 模式 2 (8-位 UART 模式)

可通过设定 SCxMOD0<SM[1:0]> 为"10"来选择 8-位 UART 模式。模式下, 能添加奇偶校验位, 并能用 SCxCR<PE>控制奇偶校验的启用/禁用。若<PE> = "1" (启用), 则能用 SCxCR<EVEN>选择偶或奇校验。

为了以下列格式接收数据, 控制寄存器的设置如下:



时钟条件	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

11.15.4 模式 3 (9-位 UART 模式)

把 SCxMOD0<SM[1:0]>设置为"11", 可选择 9-位 UART 模式。在模式下, 奇偶校验位必须禁用 (SCxCR<PE> = "0")。

发送数据时, 将最高位(第 9 位)写入到 SCxMOD0<TB8>中。数据被存储在 SCxCR<RB8>中, 用于接收数据。

在向缓冲器写入数据, 或从缓冲器读出数据时, 必须在向 SCxBUF 写入数据, 或从 SCxBUF 读出数据之前写入或读取最高有效位。

可用 SCxMOD2<SBLEN>指定停止位长度。

11.15.4.1 唤醒功能

在 9-位 UART 模式下, 把唤醒功能控制位 SCxMOD0<WU>设置为"1", 可在唤醒模式下运行从属控制器。

在这种情况下, 只有在 SCxCR<RB8>设为"1"时, 才会生成中断 INTRXx。

注: 必须用 PxOD 寄存器将从设备控制器的 TXDx 引脚设置为开路漏极输出模式。

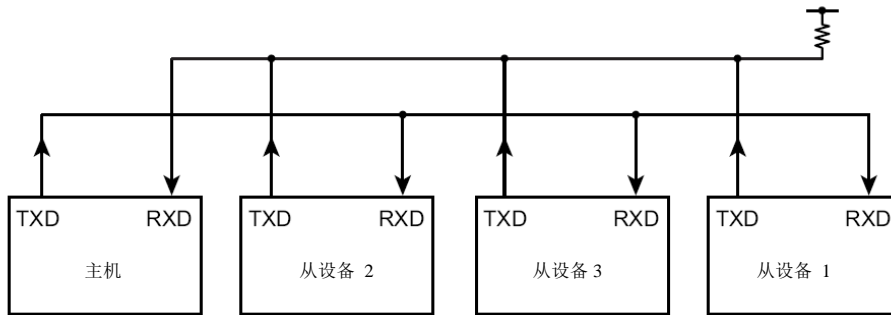
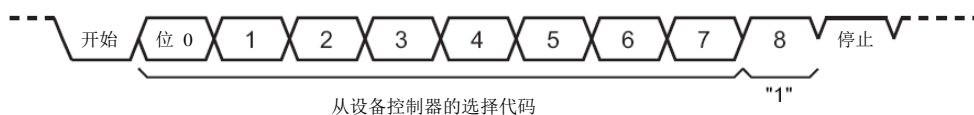


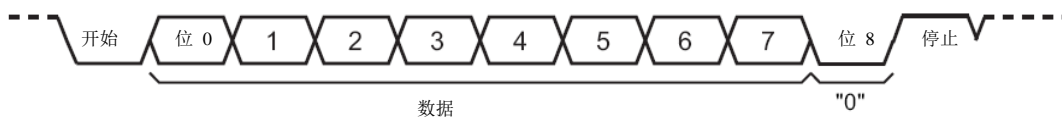
图 11-19 串行链路使用唤醒功能

11.15.4.2 协议

1. 选择各主从控制器的 9-位 UART 模式。
2. 将从控制器的 SCxMOD<WU>设置为"1", 使其随时可接收数据。
3. 主设备控制器应发送包含从设备控制器选择代码(8 位)的数据单帧。在这种情况下, 最重要的位元(位 8) <TB8> 必须设为"1"。



4. 各从设备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 位清除到"0"。
5. 主控制器发送数据给指定的从控制器(其 SCxMOD<WU>位被清除为"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. <WU>位设置为"1"的从设备控制器会忽略接收数据，原因是最高有效位(位 8)<RB8>已被设置为"0"，并因此不会生成如任何中断(INTRXx)。<WU>位设置为"0"的从设备控制器会向主设备控制器传输数据，以通知数据已被成功接收。

12. 串行总线接口(I2C/SIO)

12.1 概要

串行总线接口具备以下所述的两种工作模式。

- I2C总线模式
- 时钟同步 8-位SIO模式

在以下查验中，“x”表示通道编号。

12.2 方块图

方块图如图 12-1 所示。

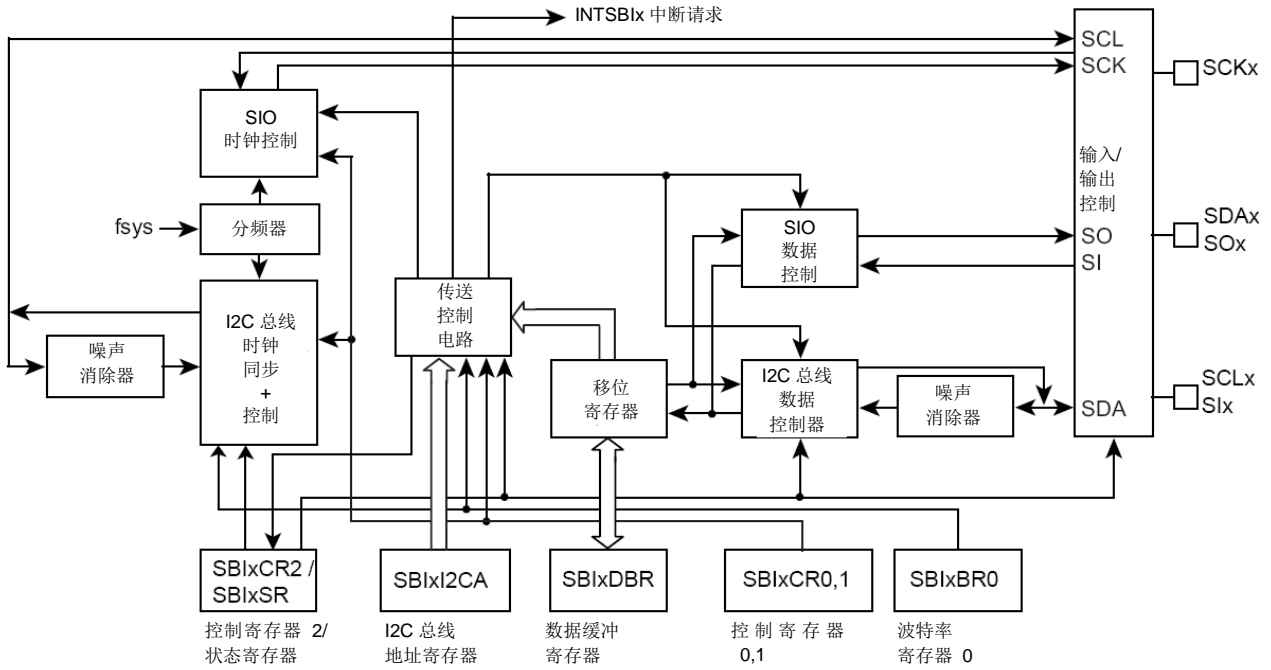
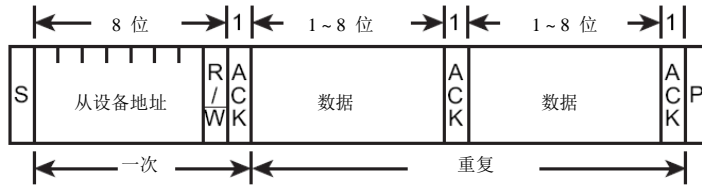


图12-1 串行总线接口方块图

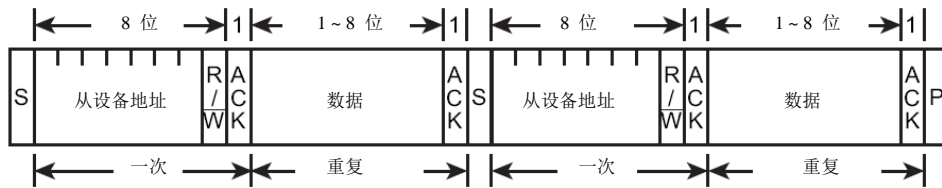
12.3 I2C 总线模式数据格式

图 12-2 给出了I2C总线模式下所采用的数据格式。

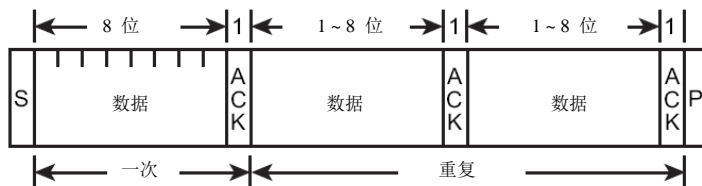
(a) 寻址格式



(b) 寻址格式(带重复起始条件)



(c) 自由数据格式(主发送器-从接收器)



注) S: 起始条件

R/ \bar{W} : 方向位

ACK: 确认位

P: 停止条件

图 12-2 I2C总线模式的数据格式

12.4 寄存器

12.4.1 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“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址(基+)
控制寄存器 0	SBIxCR0	0x0000
控制寄存器 1	SBIxCR1	0x0004
数据缓冲寄存器	SBIxDBR	0x0008
I2C总线地址寄存器	SBIxI2CAR	0x000C
控制寄存器 2	SBIxCR2 (写入)	0x0010
状态寄存器	SBIxSR (读取)	
波特率寄存器 0	SBIxBR0	0x0014

12.5 I2C 总线模式下的控制寄存器

以下寄存器可在I2总线模式下控制串行总线接口，并提供其监控状态信息。

12.5.1 SBIXCR0(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	SBIEN	-	-	-	-	-	-	-
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	SBIEN	R/W	串行总线接口工作。 0: 禁用 1: 启用 在使用串行总线接口时，需启用第一个位。 在首次设置为启用时，可读取或写入相关的SBI寄存器。 位被禁用时所有时钟SBIXCR0除外均停止工作，因此，禁用位可减少功耗。 如果位在启用之后及被禁用，则各寄存器的设置均可保留。
6-0	-	R	读作 "0"。

12.5.2 SBIXCR1 (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	-	SCK2	SCK1	SCK0 / SWRMON
复位后	0	0	0	0	1	0	0	1(注 3)

位	比特符号	类型	功能
31-8	-	R	读作 0。

位	比特符号	类型	功能				
7-5	BC[2:0]	R/W	按传输选择位数(注 1)				
			<BC>	在 <ACK> = 0 时	在 <ACK> = 1 时		
				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
4	ACK	R/W	主设备模式 0: 未生成确认时钟脉冲。 1: 生成了确认时钟脉冲。 从设备模式 0: 未计数确认时钟脉冲。 1: 已计数确认时钟脉冲。				
3	-	R	读作 "1"。				
2-1	SCK[2:1]	R/W	选择内部SCL输出时钟频率(注 2)。				
0	SCK[0]	W	000	n = 5	$\left. \begin{array}{l} \text{系统时钟} f_{\text{sys}} \\ \text{时钟齿轮: } fc/1 \\ \text{频率} = \frac{f_{\text{sys}}}{2^n + 72} \text{ [Hz]} \end{array} \right\}$		
			001	n = 6			
			010	n = 7			
			011	n = 8			
			100	n = 9			
			101	n = 10			
			110	n = 11			
			111	保留			
SWRMON	R	读取时<SWRMON>: 软件复位状态监控程序 0: 软件复位工作在进行中。 1: 软件复位工作未在进行中。					

注 1: 在将运行模式切换为SIO模式之前, 将<BC[2:0]>清除为"000"。

注 2: 有关SCL线路时钟频率的详细, 请参照"12.6.2 串行时钟"。

注 3: 在复位之后, <SCK[0]/SWRMON>位即被读作"1"。但是, 如果在SBIXCR2寄存器选择SIO模式, 则<SCK[0]>的初始值为"0"。

注 4: 选择某个频率的初始值为<SCK[2:0]>=000, 且与所读取的初始值无关。

注 5: 在<BC[2:0]>="001", 且<ACK>="0"处于主设备模式时, SCL线可能会在STOP条件生成之后, 且另一设备无法使用总线时被SCL线的下降沿固定为"L"。如果是已与若干主设备连接的总线, 则应在STOP条件生成之前, 将每次发送的位数设置为等于或大于2。

12.5.3 SBIXCR2(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	MST	TRX	BB	PIN	SBIM		SWRST	
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	选择主设备/从设备 0: 从设备模式 1: 主设备模式
6	TRX	W	选择传输/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INTSBIX中断请求 0: - 1: 清除中断请求
3-2	SBIM[1:0]	W	选择串行总线接口工作模式 (注 1) 00: 端口模式(禁用串行总线接口输入/输出)(注 2) 01: SIO模式 10: I2C总线模式(注 3) 11: 保留
1-0	SWRST[1:0]	W	软件复位产生 写入"10"再写入"01"以生成复位。 在写入<SWRST[1:0]>时, <SBIM[1:0]>即被设置为"10"(适用于I2C总线模式)。

注 1: 在传送期间, 勿改变串行总线接口工作模式。

注 2: 在确认SDAx/SCLx引脚和SOx/SIx/SCKx引脚为"高"电平之后, 切换到端口模式。

注 3: 在确认SDAx/SCLx引脚和SOx/SIx/SCKx引脚为"高"电平之后, 从端口模式切换到I2C总线模式。

注 4: SBIXCR2分配到相同的地址(带有SBIXSR)。因此, 读取-修改-写入工作无法使用。

12.5.4 SBIXSR (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	MST	TRX	BB	PIN	AL	AAS	ADO	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	主设备/从设备选择监控程序 0: 从设备模式 1: 主设备模式
6	TRX	R	发送/接收选择监控程序 0: 接收 1: 发送
5	BB	R	I2C总线状态监控程序 0: 总线闲 1: 总线忙
4	PIN	R	INTSBIX中断请求监控程序 0: 中断请求已生成 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	从设备地址匹配检测 0: - 1: 已检测 (在同时检测到全呼的情况下, 会对位进行设置。)
1	ADO	R	全呼检测 0: - 1: 已检测
0	LRB	R	最后接收位监控程序 0: 最后接收位"0" 1: 最后接收的位"1"

12.5.5 SB1xBR0 (串行总线接口波特率寄存器 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	-	I2SBI	-	-	-	-	-	-
复位后	1	0	1	1	1	1	1	0

位	比特符号	类型	功能
31-8	-	R	读作 "0"。
7	-	R	读作 "1"。
6	I2SBI	R/W	空闲模式下的工作 0: 停止 1: 工作
5-1	-	R	读作 "1"。
0	-	R/W	写入 "0"。

12.5.6 SB1xDBR (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	类型	功能
31-8	-	R	读作 0。
7-0	DB[7:0]	R	接收数据
		W	传输数据

注 1: 必须将传输数据从MSB(位 7)写入到寄存器。已接收数据即可被存储到LSB中。

注 2: 因 SB1xDBR具独立读写缓冲, 写入数据无法读取。因此, 读取-修改-写入工作无法使用。

12.5.7 SBIXI2CAR (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	类型	功能
31-8	-	R	读作 "0"。
7-1	SA[6:0]	R/W	在SBI作为从设备运行时，设置从设备地址。
0	ALS	R/W	指定地址识别模式。 0: 识别其从设备地址。 1: 不识别其从设备地址(自由数据格式)。

注 1: 请将SBIXI2CAR<ALS> 设置为"0", 但用户使用空闲数据格式的情形除外。在其被设置为"1"时, 其作为自由数据格式运行。选择拟被固定为发送功能的主设备。选择拟被固定为接收功能的从设备。

注 2: 在从设备模式下, 请勿将SBIXI2CAR设置为"0x00"。(如果SBIXI2CAR被设置为"0x00", 则会认识到从设备地址可与在从设备模式下接收到的I2C标准的START字节("0x01")匹配。)

12.6 I2C 总线模式下的控制

12.6.1 设置工作模式

通过 $SBIxCR2\langle SBIM[1:0]\rangle$ 设置工作模式。在 SBI 被用作 I2C 总线模式时，将 $\langle SBIM[1:0]\rangle$ 设置为 "10"。

注1：在确认 $SDAx/SCLx$ 引脚和 $SOx/SIx/SCKx$ 引脚为 "高" 电平之后，切换到端口模式。

注2：在确认 $SDAx/SCLx$ 引脚和 $SOx/SIx/SCKx$ 引脚为 "高" 电平之后，即从端口模式切换到 I2C 总线模式。

12.6.2 串行时钟

12.6.2.1 时钟源

$SBIxCR1\langle SCK[2:0]\rangle$ 可指定串行时钟在主设备模式下从 $SCLx$ 引脚输出的最高频率。

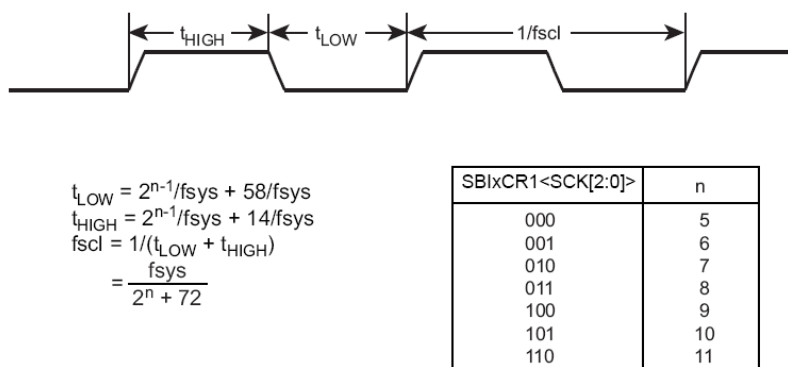


图 12-3 时钟源

注：按照通信标准，在标准模式和高速模式中，最大速度分别规定为 100 kHz 和 400 kHz。注意可依据所使用的 f_{sys} 和上文给出的计算公式，求出内部 SCL 时钟频率。

12.6.2.2 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" 电平的首个主设备，超越在其时钟线上产生 "高" 电平时其它主设备。产生 "高" 电平的主设备必须就此进行检测并应答。

时钟同步可保证具备两个或以上主设备的总线的正确数据传送。

例如，以下给出了带有两个主设备的总线的时钟同步程序。

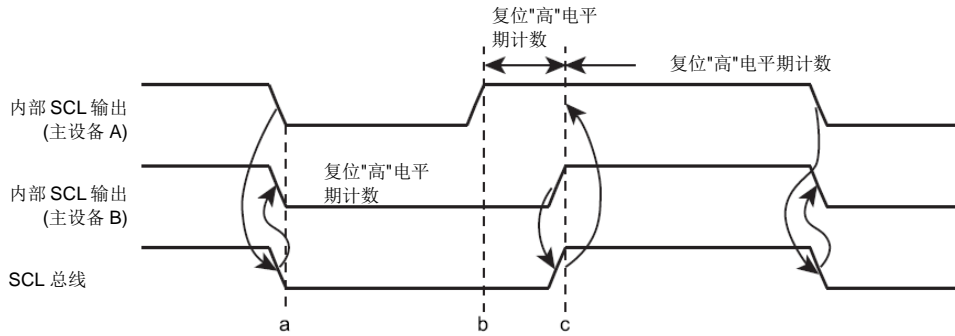


图 12-4 时钟同步示例

主设备 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”。主设备 B 检测到转换，复位其“高”电平周期计数器，并将其内部 SCL 输出电平拉至“低”电平。

主设备 A 完成点 B“低”电平周期的计数，并将内部 SCL 输出提至“高”电平。不过，主设备 B 仍会将 SCL 总线保持为“低”电平，且主设备 A 会停止其“高”电平周期计数。在主设备 A 检测到主设备 B 将其内部 SCL 输出提至“高”电平，并将点 c 的 SCL 总线提至“高”电平之后，其会启动其“高”电平周期的计数。

在主设备完成“高”电平周期的计数之后，主设备会将 SCLx 引脚拉至“低”，SCL 总线变为“低”。

通过这种方法，即可依据连接到母线的主设备中“高”电平周期最短的主设备，以及“低”电平周期最长的主设备，确定总线的时钟。

12.6.3 设置应答模式

将 $SBIxCR1<ACK>$ 设置为“1”，即可选择应答模式。

在作为主设备运行时，SBI 可添加一个应答信号。

在从设备模式下，可对确认信号的时钟进行计数。

在传输器模式下，SBI 可在时钟应答周期内解除 $SDAx$ 引脚，以从接收器接收应答信号。

在接收器模式下，SBI 可在时钟应答周期内将 $SDAx$ 引脚拉至“低”电平，并生成应答信号。如果在从设备模式下接收到全呼，则 SBI 可在时钟应答周期内将 $SDAx$ 引脚拉至“低”电平，并生成应答信号。

通过将 $<ACK>$ 设置为“0”，即可激活非应答模式。在运行某个主设备时，SBI 不会生成应答信号的时钟。在从设备模式下，不会对应答信号的时钟进行计数。

12.6.4 设置每次发送的位数

$SBIxCR1<BC[2:0]>$ 可指定拟传输或接收的下一数据的位数。

在启动条件下， $<BC[2:0]>$ 会被设置为“000”，从而通过八位数据包传送从设备地址和方向位。在其他时候， $<BC[2:0]>$ 会保持此前某个编程值。

12.6.5 从属寻址和地址识别模式

将"0"设置到 SBIxI2CAR<ALS>,并按寻址格式在 SBIxI2CAR<SA[6:0]>中设置某从设备地址,然后,SBI 可识别出通过主设备发送的从设备地址,并接收寻址格式的数据。

如果<ALS> 被设置为"1",则 SBI 无法识别从设备地址和接收空闲数据格式的数据。如果是空闲数据格式,则无法识别从设备地址和方向位;在启动条件生成之后即识别为数据。

12.6.6 将 SBI 作为主设备模式或从设备模式进行配置

通过 SBIxCR2<MST>将设置为"1",即可将 SBI 配置为主设备并进行工作。

通过将<MST>设置为"0",即可将 SBI 配置为从设备。

在 SBI 在总线上检测到停止条件或仲裁丢失时,<MST>清除为"0"。

12.6.7 将 SBI 配置为发送器或接收器

通过将 SBIxCR2<TRX>设置为"1",可将 SBI 配置为发送器。

通过将<TRX>设置为"0",可将 SBI 配置为接收器。

在 SBI 检测到总线上的停止条件或仲裁丢失时,<TRX>清除为"0"。

如果 SBI 被用于自由数据格式,则硬件不会改变<TRX>。如果 SBI 被用于寻址格式,则应按以下所述设置<TRX>。

12.6.7.1 主设备模式

在主设备模式下,如果 SBI 接收到发自某从设备的应答,则会由某硬件按下述要求对<TRX>进行设置。

如果 SBI 未应答,则<TRX>保持前一值。

- 在所发送的方向位为"1"时,<TRX>设置至"0"。
- 在所发送的方向位为"0"时,<TRX>设置至"1"。

12.6.7.2 从设备模式

在从设备模式下,如果是寻址格式,且满足以下条件,则会按照主设备所发送的方向位对<TRX>进行设置。

- 当所接收的从设备地址与 SBIxI2CAR 中所设置的值相同时。
- 在 SBI 接收到全呼

且<TRX> 的设置符合以下要求时。

- 当所接收的方向位为"1",且<TRX>设置为"1"时。
- 当所接收的方向位为"0",且<TRX>设置为"0"时。

12.6.8 总线忙监控器

在确认总线状态时，应读取 SBIxSR<BB>。

在 SBI 检测到总线上的启动条件时，<BB> 设置至 "1"；在 SBI 检测到总线上的停止条件时，清除至 "0"。

在 <BB> 为 "1" 时，称之为总线忙。在 <BB> 为 "0" 时，称之为总线闲。

主设备仅在总线闲时生成启动条件。应确认 <BB> 为 "0"。

在 <BB> 为 "1" 时，SBI 生成启动条件，该启动条件未生成，发生仲裁丢失。

12.6.9 中断服务请求和解除

在 INTSBIx 生成时，SBIxCR2<PIN>清除为 "0"，SBI 会处于中断服务请求状态。

SBI 将 SCLx 引脚拉至 "低" 电平。

在数据写入到 SBIxDBR 或从 SBIxDBR 读出时，<PIN> 即设置至 "1"。在程序向 <PIN> 写入 "1" 时，其设置至 "1"。不过，"0" 的写入不会将 <PIN> 清除至 "0"。

如果 <PIN> 设置至 "1"，则 SCLx 引脚被解除。从将 <PIN> 设置至 "1" 以解除 SCLx 引脚，所需的时间为 t_{Low} 。

注：在主设备模式下发生仲裁丢失时，如果所接收的从设备地址不匹配，则 <PIN> 不会清除至 "0"。但出现 INTSBIx。

12.6.10 仲裁丢失检测监控器

I2C 总线具备多主设备能力(一根总线上可设置两个或更多主设备)，并要求利用总线仲裁程序，以确保正确的数据发送。

如主设备试图在总线忙期间生成启动条件，则会丢失总线仲裁，SDA 和 SCL 总线上不会出现任何启动条件。SDA 总线上会发生 I2C 总线仲裁。

一条总线上的两个主设备的仲裁程序如以下所述。

在到达点 a 之前，主设备 A 和主设备 B 输出相同的数据。主设备 A 在点 a 时可输出 "低" 电平，而主设备 B 则输出 "高" 电平。

然后，主设备 A 将 SDA 总线拖至 "低" 电平，原因是线路具备配线 AND 连接。在 SCL 线在点 B 变为 "高" 时，从设备会读取 SDA 线数据比如主设备 A 所发送的数据。此时，主设备 B 所发送的数据会变为无效。

主设备 B 的这种情况称为 "仲裁丢失"。主设备 B 会解除其 SDAx 引脚，以确保其不影响另一主设备所启动的数据传送。如果两个或更多主设备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。

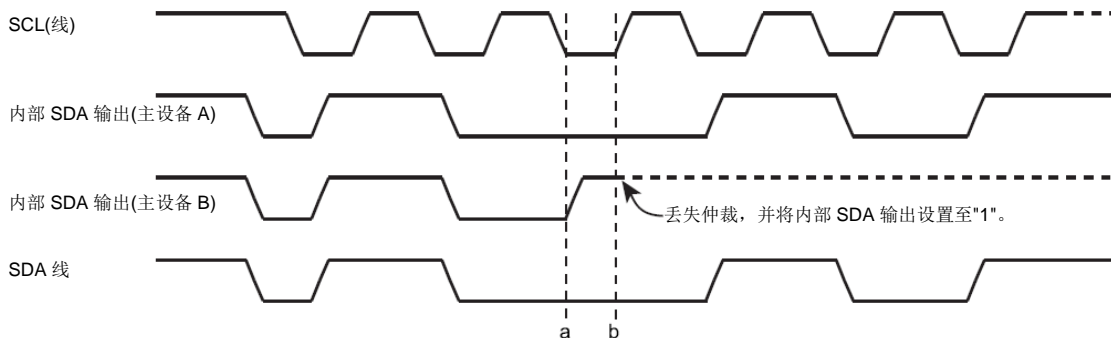


图 12-5 丢失仲裁

主设备会比较 SDA 总线电平和 SCL 线路上升部位的内部 SDA 输出电平。如果两个电平值之间存在差异，则仲裁丢失，SBIxSR<AL>设置至"1"。

在<AL>设置至"1"时，SBIxSR<MST, TRX>清除至"0"，导致 SBI 作为从设备接收器运行。因此，串行总线接口电路会在<AL>设置"1"之后的数据传送期间内停止时钟输出。

在数据写入到 SBIxCR2 或从 SBIxDBR 读出时，<AL>清除至"0"。

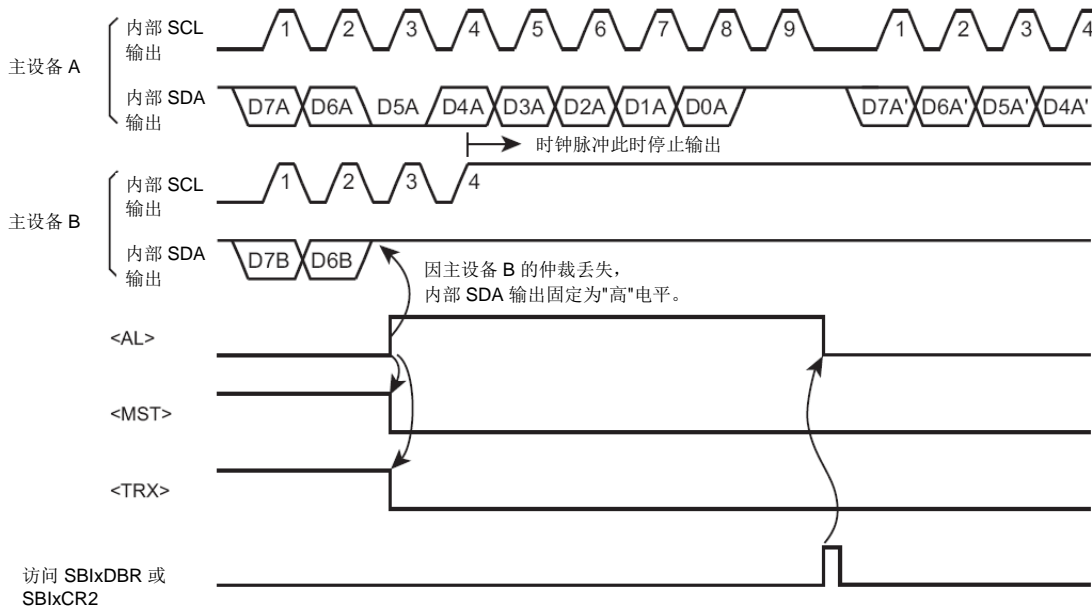


图 12-6 主设备B丢失仲裁示例(D7A = D7B, D6A = D6B)

12.6.11 从设备地址匹配检测监控器

在 SBI 按寻址格式(SBIxI2CAR<ALS>="0")作为从设备运行时，一旦接收到全呼或可匹配在 SBIxI2CAR 指定的值，从设备地址 SBIxSR<AAS>设置至"1"。

在<ALS>为"1"时，<AAS>在首个数据字接收时设置至"1"。

在数据写入或从 SBIxDBR 读出时，<AAS>清除至"0"。

12.6.12 一般呼叫检测监控器

在 SBI 作为从设备运行时，在其接收到全呼地址例如：启动条件后面的八位全部为零时，SBIxSR<ADO>设置至"1"。

在总线上的启/停条件检测到时，<ADO>清除至"0"。

12.6.13 最后接收位监控程序

SBIxSR<LRB>设置至 SCL 线路上升处的 SDA 线路值。

在应答模式下，如在 INTSBIx 中断请求生成之后立即读取 SBIxSR<LRB>，则会导致 ACK 信号读取。

12.6.14 数据缓冲寄存器(SBIxDBR)

在读取数据或写入数据时，会从 SBIxDBR 读取或写入 SBIxDBR。

在 SBI 处于主设备模式时，在向 SBIxDBR 中的寄存器写入从设备地址和方向位之后，启动条件生成，且 SBI 会向从设备发送从设备地址和方向位。

12.6.15 波特率寄存器(SBIxBR0)

SBIxBR0<I2SBI>寄存器可决定在其进入 IDLE 模式时，SBI 是否运行。

在执行一个指令以切换到待机模式之前，必须对寄存器进行编程。

12.6.16 软件复位

如果 SBI 因外部噪声而锁定，则可使用软件复位对其进行初始化。

向 SBIxCR2<SWRST[1:0]> 写入"10"其后带有"01"可生成可初始化 SBI 的 RESET 信号。在写入 SBIxCR2<SWRST[1:0]>时，将"10"设置到 SBIxCR2<SBIM[1:0]>可切换到 I2C 总线模式。在复位之后，所有控制寄存器和状态标志均会初始化为其复位值。在 SBI 初始化时，<SWRST>会自动清除至"00"。

注：软件复位将导致 SBI 运行模式从 I2C 模式切换为端口模式。

12.7 I2C 总线模式下的数据传送规程

12.7.1 设备初始化

首先，对 SBIxCR1<ACK, SCK[2:0]>进行编程。此时，将"000"写入到 SBIxCR1<BC[2:0]>。

然后，对 SBIxI2CAR 进行编程，方法是在<SA[6:0]>指定一个从设备地址，并在<ALS>指定地址识别模式(在使用寻址格式时，必须将<ALS>清除至"0")。

在将串行总线接口配置为从设备接收器时，首先需确保串行总线接口引脚处于"高"状态。然后，将"0"写入到 SBIxCR2<MST, TRX, BB>，将"1" 写入到<PIN>，将"10"写入到<SBIM[1:0]>并将"0"写入到位 1 和位 0。

注：必须是在与总线连接的所有设备均初始化后，任何设备都未生成启动条件的周期内，完成串行总线接口电路的初始化。如果不遵循规则，则可能不会正确的接收数据，原因是其它设备可能在串行总线接口电路的初始化完成之前开始传送。

	7	6	5	4	3	2	1	0		
SBIxCR1	←	0	0	0	x	0	x	x	x	指定ACK和SCL时钟。
SBIxI2CAR	←	x	x	x	x	x	x	x	x	指定一个从设备地址和一种地址识别模式。
SBIxCR2	←	0	0	0	1	1	0	0	0	将SBI配置为I2C总线模式和从设备接收器。

注：X；忽略

12.7.2 生成启动条件和从设备地址

在生成启动条件和从设备地址时，必须遵循以下步骤。

首先，确保总线处于总线闲(SBIxSR<BB> = "0")状态。其次，将 SBIxCR1<ACK>设置至"1"以选择应答模式。将拟传输的一个从设备地址和一个方向位写入到 SBIxDBR。

在<BB> = "0"时，向 SBIxCR2<MST, TRX, BB, PIN>写入"1111"，可在总线上生成启动条件。

根据启动条件，SBI 可从 SCLx 引脚生成九个时钟。

SBI 输出在 SBIxDBR 中指定的从设备地址和方向位，以及前八个时钟，并解除第九个时钟中的 SDA 线路，以接收一个来自从设备的确认信号。

在第九时钟的下降沿生成 INTSBIx 中断请求，<PIN>清除至"0"。在<PIN>为"0"期间，SBI 可将 SCL 线保持为"低"电平。<TRX>可按照 INTSBIx 中断请求生成时的发送方向位改变其值，但前提是从设备已返回一个应答信号。

注：输出从设备地址时，在写入SBIxDBR之前，需要用软件检查总线是否空闲。如果不遵守这一原则，在总线上待输出的数据有可能会损坏。如果不遵循规则，则正在总线上输出的数据可能会破坏。

主程序中的设置

```

    7 6 5 4 3 2 1 0
    Reg. ← SBIxSR
    Reg. ← Reg. e 0x20
    if Reg. ≠ 0x00
    Then
        SBIxCR1 ← X X X 1 0 X X X    选择应答模式。
        SBIxDBR ← X X X X X X X X    指定所需的从设备地址和方向。
        SBIxCR2 ← 1 1 1 1 1 0 0 0    生成启动条件。
    
```

INTSBI0 中断程序示例
清除中断请求。
处理
中断结束

确保总线处于空闲。

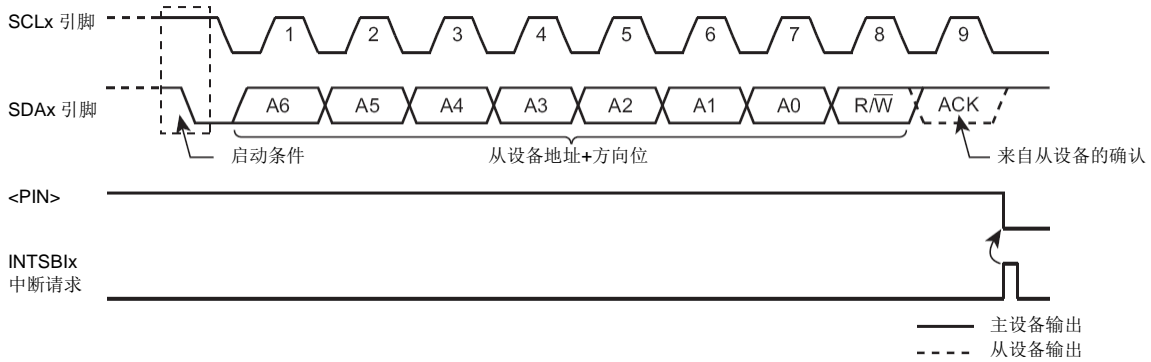


图 12-7 启动条件和从设备地址的生成

12.7.3 传送数据字

在数据字传送结束时，随生成 INTSBIx 中断以测试<MST>，从而确定 SBI 是否处于主设备或从设备模式。

12.7.3.1 主模式(<MST> = "1")

测试<TRX>以确定 SBI 是否配置为传输器或接收器。

(1) 传输器模式(<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，主设备生成后文所述的停止条件，以停止传输。如果<LRB>为"0"，则表示接收器要求提供更多的数据。

如果下一步拟传输的数据具备八位，则数据会写入到 SBIxDBR。如果数据有不同的长度，则 <BC[2:0]>会接受编程，传输数据则会写入到 SBIxDBR 中。

数据的写入可造成<PIN>变为"1", 导致 SCLx 引脚生成下一数据字传送, 以及 SDAx 引脚传送数据字所需的串行时钟。

在传送完成之后, INTSBIx 中断请求生成, <PIN>清除至"0", 且 SCLx 引脚会拖至"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。

INTSBIx中断

```

if MST = 0,
    则转到从属模式处理。
if TRX = 0,
    则转到接收器模式处理。
if LRB = 0,
    则转到停止条件生成处理。
SBIxCR1 ← 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。
SBIxDBR ← X X X X X X X X   写入发送数据。
中断处理结束。
    
```

注: X; 忽略

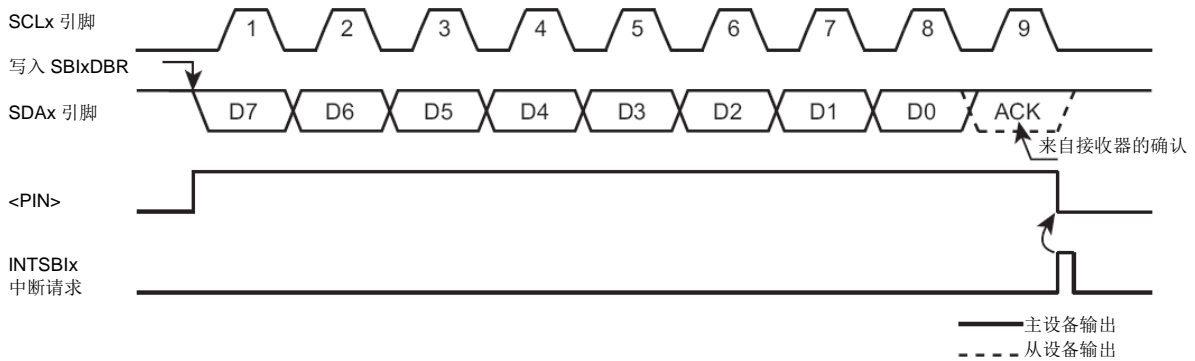


图 12-8 <BC[2:0]>= "000", <ACK>= "1" (传输器模式)

(2) 接收器模式(<TRX> = "0")

如果拟发送的下一数据有八位, 则应从 SBIxDBR 读取所接收的数据。

如果数据有不同的长度, 则应对<BC[2:0]>进行编程, 并应从 SBIxDBR 读取所接收的数据以解除 SCL 线(从设备地址的发送后立读取的数据将不赋值未定义)。

在读取数据时, <PIN>设置至"1", 且串行时钟会输出到 SCLx 引脚, 以传送下一个数据字。在最后位中, 在应答信号变为“低”电平时, "0"输出到 SDAx 引脚。

然后, INTSBIx 中断请求生成, <PIN>清除至"0", SCLx 引脚则随拖至"低"电平。每从 SBIxDBR 读取一次所接收的数据, 就会输出一字传送时钟和一个应答信号。

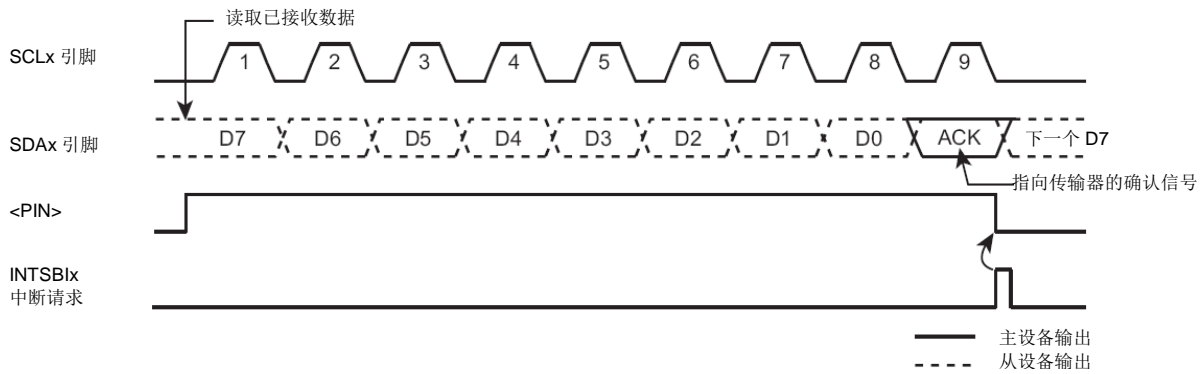


图 12-9 <BC[2:0]>= "000", <ACK>= "1" (接收器模式)

在终止源自发送器的数据发送时，<ACK> 会在将读取第二个至最后一个数据字之前清除至"0"。

这样可禁止生成最后数据字的应答时钟。

在传送完成时，一个中断请求生成。在进行中断处理时，必须将<BC[2:0]>设置至"001"，且必须读取数据，以确保生成 1-位传送所需的时钟。

此时，主设备为接收器，主设备可保持 SDA 线为"高"电平。传输器接收作为 ACK 信号的"高"电平，主设备接收器则可告知发送器次传送的结束时间。

在 1-位数据接收终止的中断处理期间，可生成 STOP 条件以终止数据传送。

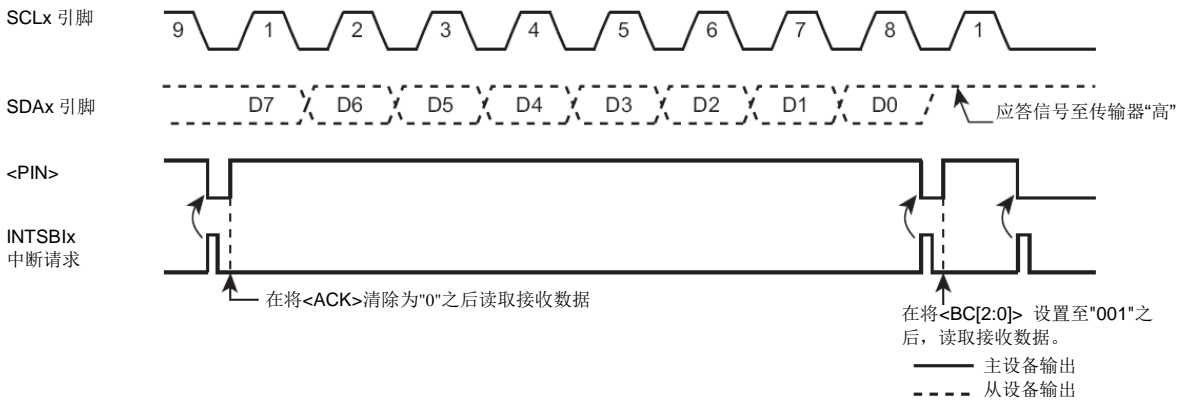


图12-10 主设备接收器模式下的终止数据传输

示例：在接收 N 数据字时

INTSBIx 中断(在数据传输之后)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	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.	←	SBIxDBR								读取虚拟数据。
中断结束										

INTSBIx 中断(第一个至第(N-2)个数据接收)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxDBR								读取第一个至第(N-2)个数据字。
中断结束										

INTSBIx 中断(第(N-1)个数据接收)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	x	x	x	0	0	x	x	x	禁止确认时钟的生成。
Reg.	←	SBIxDBR								读取第(N-1)个数据字。
中断结束										

INTSBIx 中断(第 N 个数据接收)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	x	x	x	将位数目设置至“1”。
Reg.	←	SBIxDBR								读取第N个数据字。
中断结束										

INTSBIx 中断(在完成数据接收之后)

		进行处理，以生成停止条件。								终止数据发送。
中断结束										

注：X；忽略

12.7.3.2 从设备模式 (<MST> =“0”)

在从设备模式下，在 SBI 接收到其从设备地址或全呼之后，在 SBI 完成传送数据时，SBI 可在接收来自主设备的任何从设备地址或全呼时生成 INTSBIx 中断请求。

此外，如果在主模式 SBI 检测到仲裁丢失，则切换至从设备模式。在数据字传送已在其中检测到仲裁丢失完成时，INTSBIx 中断请求生成。

在 INTSBIx 中断请求时，<PIN>会清除至“0”，且 SCLx 引脚会拖至“低”电平。

在数据写入或从 SBIxDBR 读取时，或在<PIN>设置至“1”时，SCLx 引脚会在 t_{LOW} 消逝后解除。

在从设备模式下，会进行标准从属模式处理，或作为仲裁丢失结果的处理，且其会从主设备模式切换到从设备模式。

会对 SBIxSR<AL>，<TRX>，<AAS>和<ADO>进行测试，以确定所需进行的处理。

"表 12-1 从设备模式下的处理"给出了从设备模式状态和所需处理。



示例：在所接收的从设备地址匹配 SBI 的从设备地址，且从设备接收器模式下的方向位为"1"时。

INTSBIx 中断

if TRX = 0

则转到其它处理。

if AL = 0

则转到其它处理。

if AAS = 0

则转到其它处理。

SBIxCR1 ← X X X 1 0 X X X 设置拟发送的位的数目。

SBIxDBR ← X X X X X X X X 设置发送数据。

注：X; 忽略

表12-1 从设备模式下的处理

<TRX>	<AL>	<AAS>	<ADO>	状态	处理
1.	1	1	0	在从设备地址正传输期间检测到仲裁丢失，且SBI收到了由另一主设备传输的方向位为"1"的从设备地址。	将数据字中的位数设置至<BC[2:0]>。并将发送数据写入到SBIxDBR中。
		1	0	在从设备接收器模式下，SBI接收到发自主设备的一个从设备地址连同方向位"1"。	
	0	0	0	在从设备传输器模式下，SBI已完成一个数据字的传输。	测试LRB。如果其设置至"1"，则表示接收器不要求提供更多的数据。将<PIN>设置至 1，并将<TRX>复位为 0，可解除总线。如果 <LRB>已复位为"0"，则表示接收器要求提供更多的数据。将数据字中的位数设置到<BC[2:0]>，并将发送数据写入到SBIxDBR中。
0	1	1	1/0	在从设备地址正传输期间检测到仲裁丢失，且SBI收到了由另一主设备传输的方向位为"0"的任一从设备地址或全呼地址。	读取SBIxDBR (虚拟读取)以将 <PIN>设置至 1, 或向<PIN>写入"1"。
		0	0	在从设备地址或数据字正传输期间检测到仲裁丢失，传送终止。	
	0	1	1/0	在从设备接收器模式下，SBI收到了由另一主设备发送的方向位为"0"的任一从设备地址或全呼地址。	
		0	1/0	在从设备接收器模式下，SBI已完成一个数据字的接收。	

12.7.4 生成停止条件

在 SBIxSR<BB>为"1"时，向 SBIxCR2<MST, TRX, PIN>写入"1"和向<BB>写入"0"会导致 SBI 启动总线上的 STOP 条件生成顺序。

在停止条件出现在总线上之前，请勿改动<MST, TRX, BB, PIN>的内容。

如果另一设备正在保持 SCL 总线，则 SBI 会等待，直至 SCL 线解除。

然后，SDAx 引脚走"高"，STOP 条件生成。

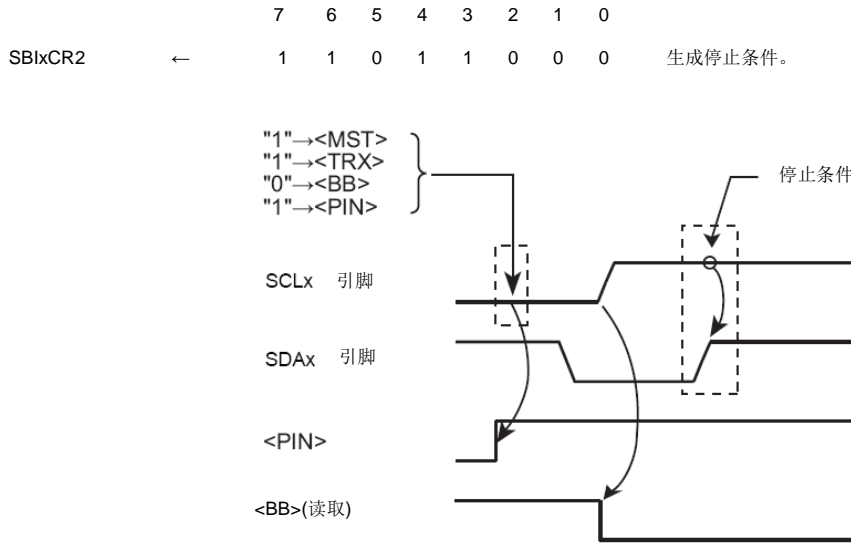


图 12-11 生成停止条件

12.7.5 重启程序

在主设备未终止对某个从设备的传送的情况下改变数据传送方向时，就需要进行重新启动。在主模式生成重新启动的程序如下所述。

首先，将 SBIxCR2<MST, TRX, BB> 写为“0”，并将“1”写入到<PIN>，可解除总线。此时，SDAx 引脚保持在“高”电平，SCLx 引脚则解除。由于总线上未生成停止条件，因此，其它设备会将总线识别为占用。

然后，测试 SBIxSR<BB> 并等待至其变为“0”，以确保 SCLx 引脚解除。

然后，测试<LRB> 并等待至其变为“1”，以确保任何其它设备都不会将 SCL 总线拖至“低”电平。

一旦决定拟按上述规程让总线保持在空闲状态，则应执行“12.7.2 生成启动条件和从设备地址”中所述的规程，以生成启动条件。

为满足重新启动的准备时间，在决定拟让总线保持空闲之后，必须通过软件创建时长至少为 4.7μs 的周期(在标准模式下)。

注 1：在其“0”时，不要将<MST> 写为“0”。(重新启动无法启动)。

注 2：在主设备作为接收器工作时，必须在重新启动生成之前，完成从作为传输器工作的从设备进行的数据传输。为完成数据发送，从设备必须接收一个“高”电平确认信号。为此，重新启动生成前的<LBR>变为“1”，使按重新启动规程确认<LBR>=“1”，SCL线的上升沿仍不能检测到。在检查SCL线路的状态时，需读取端口。



SBIxCR1	←	x	x	x	1	0	x	x	x	选择确认模式。
SBIxDBR	←	x	x	x	x	x	x	x	x	设置所需的从设备地址和方向。
SBIxCR2	←	1	1	1	1	1	0	0	0	生成启动条件。

注: X; 忽略

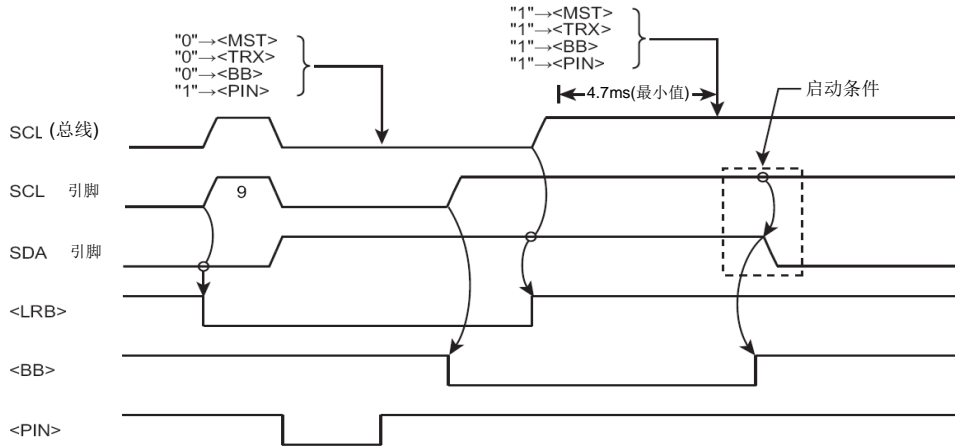


图 12-12 重新启动生成的时序表

12.8 SIO 模式控制寄存器

下述寄存器控制时钟同步 8-位SIO模式串行总线接口，带有状态信息，以便监控。

12.8.1 SBiXCR0(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	SBIEN	-	-	-	-	-	-	-
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	SBIEN	R/W	串行总线接口工作。 0: 禁用 1: 启用 使用串行总线接口前启用位。 若位禁用, 由于除SBiXCR0外所有时钟停止, 可减少功耗。 若串行总线接口工作启用后再禁用, 设置仍然保留于各寄存器。
6-0	-	R	读作 "0"。

12.8.2 SBIxCR1(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	SIOS	SIOINH	SIOM		-	SCK		
复位后	0	0	0	0	1	0	0	0(注 1)

位	比特符号	类型	功能																		
31-8	-	R	读作 "0"。																		
7	SIOS	R/W	传送启动/停止 0: 停止 1: 启动																		
6	SIOINH	R/W	传送 0: 继续 1: 强制终止																		
5-4	SIOM[1:0]	R/W	选择传送模式 00: 传输模式 01: 保留 10: 传输/接收模式 11: 接收模式																		
3	-	R	读作 "1"。																		
2-0	SCK[2:0]	R/W	开启写入<SCK[2:0]>: 选择串行时钟频率。(注 1) <table border="1" style="margin-left: 20px;"> <tr> <td>000</td> <td>n = 3</td> <td rowspan="11" style="font-size: 2em; vertical-align: middle;">} [系统时钟: fsys 时钟齿轮: fc/1 频率 = $\frac{fsys/2}{2^n}$ [Hz]]</td> </tr> <tr> <td>001</td> <td>n = 4</td> </tr> <tr> <td>010</td> <td>n = 5</td> </tr> <tr> <td>011</td> <td>n = 6</td> </tr> <tr> <td>100</td> <td>n = 7</td> </tr> <tr> <td>101</td> <td>n = 8</td> </tr> <tr> <td>110</td> <td>n = 9</td> </tr> <tr> <td>111</td> <td>-</td> <td>外部时钟</td> </tr> </table>	000	n = 3	} [系统时钟: fsys 时钟齿轮: fc/1 频率 = $\frac{fsys/2}{2^n}$ [Hz]]	001	n = 4	010	n = 5	011	n = 6	100	n = 7	101	n = 8	110	n = 9	111	-	外部时钟
000	n = 3	} [系统时钟: fsys 时钟齿轮: fc/1 频率 = $\frac{fsys/2}{2^n}$ [Hz]]																			
001	n = 4																				
010	n = 5																				
011	n = 6																				
100	n = 7																				
101	n = 8																				
110	n = 9																				
111	-		外部时钟																		

注 1: 复位后, <SCK[0]>位读作 "1"。但是, 如果在SBIxCR2寄存器选择SIO模式, 则初始值会读作"0"。本文中, "复位后"列所写数值为初始状态设置SIO模式后的数值。SBIxCR2 寄存器说明和SBIxSR寄存器说明相同。

注 2: 传送模式和串行时钟程序编制之前, 将 <SIOS> 设至"0"且 <SIOINH> 至"1"。

12.8.3 SBIxDBR (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	类型	功能
31-8	-	R	读作 0。
7-0	DB[7:0]	R	接收数据
		W	传输数据

注：因 SBIxDBR具独立读写缓冲器，写入数据无法读取。因此，读取-修改-写入工作无法使用。

12.8.4 SBiXCR2(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	-	-	-	-	SBIM		-	-
复位后	1(注 1)	1(注 1)	1(注 1)	1(注 1)	0	0	1(注 1)	1(注 1)

位	比特符号	类型	功能
31-8	-	R	读作 "0"。
7-4	-	R	读作 "1"。(注 1)
3-2	SBIM[1:0]	W	选择串行总线接口工作模式 (注 2) 00: 端口模式 01: SIO模式 10: I2C总线模式 11: 保留
1-0	-	R	读作 1。(注 1)

注 1: 本文中, "复位后"列所写数值为初始状态设置SIO模式后的数值。

注 2: 通信会话时务必不要改变模式。

注 3: 在确认SDAx/SCLx引脚和SOx/SIx/SCKx引脚为“高”电平之后, 即从端口模式切换到I2C总线模式。

注 4: SBiXCR2分配到相同的地址带有SBiXSR。因此, 读取-修改-写入工作无法使用。

12.8.5 SBIXSR (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	-	-	-	-	SIOF	SEF	-	-
复位后	1(注 1)	1(注 1)	1(注 1)	1(注 1)	0	0	1(注 1)	1(注 1)

位	比特符号	类型	功能
31-8	-	R	读作 "0"。
7-4	-	R	读作 "1"。(注)
3	SIOF	R	串行传送状态监控器。 0: 完成 1: 进行中
2	SEF	R	移位工作状态监控器 0: 完成 1: 进行中
1-0	-	R	读作 "1"。(注)

注：在本文件中，写入到“复位后”栏中的值，是在初始状态下设置SIO模式之后的值。

12.8.6 SBlxBR0 (波特率寄存器 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	-	I2SBI	-	-	-	-	-	-
复位后	1	0	1	1	1	1	1	0

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	-	R	读作"1"。
6	I2SBI	R/W	IDLE模式工作。 0: 停止 1: 工作
5-1	-	R	读作"1"。
0	-	R/W	写入"0"。

12.9 SIO 模式控制

12.9.1 串行时钟

12.9.1.1 时钟源

通过对 $SBIxCR1<SCK[2:0]>$ 进行编程，可选择内部或外部时钟。

(1) 内部时钟

内部时钟模式中，作为通过 $SCKx$ 引脚向外输出的串行时钟，可选择七个频率之一。

传送开始时， $SCKx$ 引脚输出成为“高”电平。

若程序写入发送数据或读取接受数据时无法保持串行时钟频率， SBI 自动进入等待时间。在时间段内，串行时钟自动停止，下一移位工作暂停，直至处理完成。

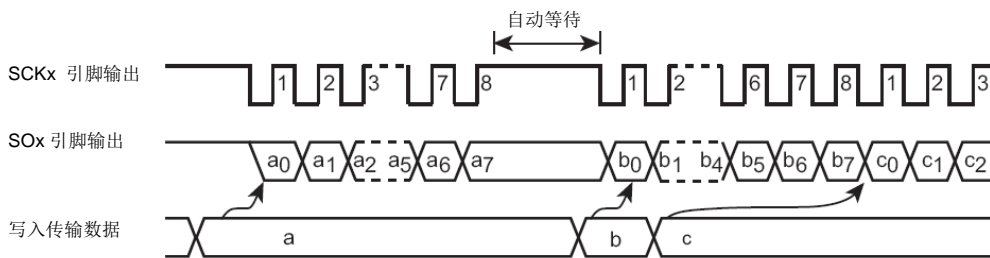


图 12-13 自动等待

(2) 外部时钟 ($<SCK[2:0]> = "111"$)

SBI 使用作为串行时钟自外供至 $SCKx$ 引脚的外部时钟。

正确移位工作，“高”和“低”阶的串行时钟须具备如下所示脉宽。

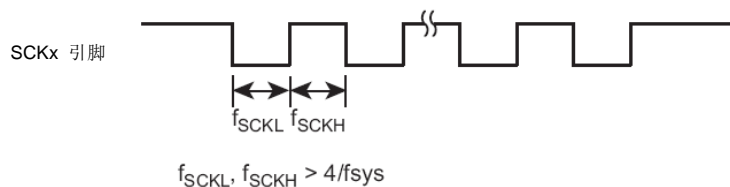


图 12-14 外部时钟输入的最高传送频率

12.9.1.2 移位沿

前沿移位用于传输。后沿移位用于接收。

- 前沿移位

数据于串行时钟前沿(或 SCKx 引脚输入/输出下降沿)移位。

- 后沿移位

数据于串行时钟后沿(或 SCKx 引脚输入/输出上升沿)移位。

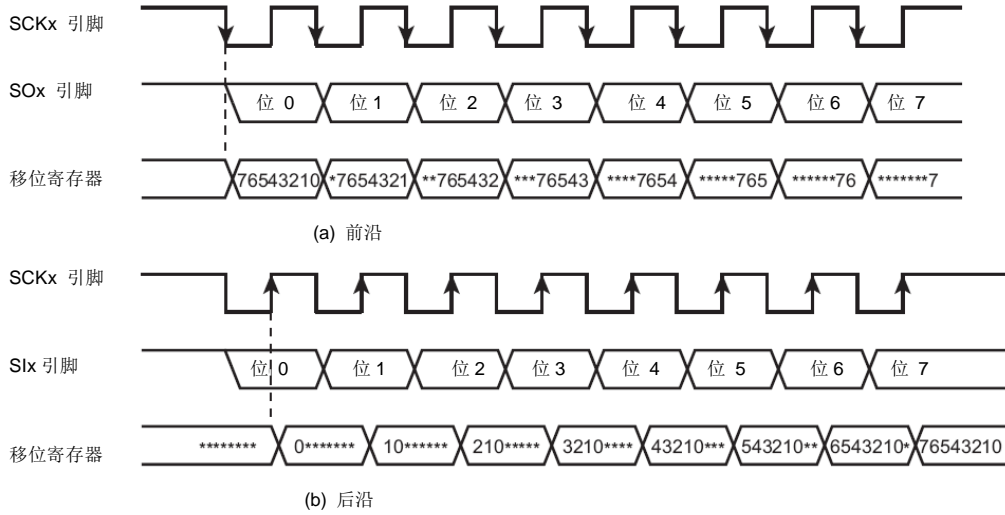


图 12-15 移位沿

12.9.2 传送模式

通过对 SBIxCR1<SIOM[1:0]>进行编程，可选择发传输式，接收模式或传输/接收模式。

12.9.2.1 8-位传输模式

将控制寄存器设为传输模式，将传输数据写入 SBIxDBR。

写入传输数据后，往 SBIxCR1<SIOS>写入"1"，开始传输。传输数据连同最小有效位(LSB)首先从 SBIxDBR 移动到移位寄存器，并输出到 SOx 引脚(与串行时钟同步)。传输数据传送到移位寄存器时，SBIxDBR 即为空，INTSBIx(缓冲器-为空)中断生成，并请求提供下一传输数据。

内部时钟模式中，8-位数据全部传输后，若加载下一数据，串行时钟将停止，自动进入等待状态。SBIxDBR 加载下一传输数据时，等待状态将清除。

外部时钟模式中，SBIxDBR 须在下一数据移位工作开始之前加载。因此，视乎产生中断请求时和 SBIxDBR 加载中断服务程序数据时之间的最大延迟，数据传送率各不相同。

传输开始时，前一传输数据最后位的同样数值于自将 SBIxSR<SIOF>设为"1"至 SCK 下降沿期间输出。

INTSBIx 中断服务程序中将<SIOS>清至"0"或将<SIOINH>设为"1"可终止发送。若<SIOS>清除，发送结束前输出剩余数据。程序检查 SBIxSR<SIOF>，以判断是否传输已结束。传输结束时，<SIOF>清除至"0"。若<SIOINH> 设为"1"，传输立即终止，<SIOF>清除至"0"。

外部时钟模式时，下一数据移位前，<SIOS>须清至"0"。若<SIOS>在下一数据移位前未清至"0"，SBI 输出虚拟数据并停止。

	7	6	5	4	3	2	1	0	
SBIxCR1	← 0	1	0	0	0	x	x	x	选择传输模式。
SBIxDBR	← x	x	x	x	x	x	x	x	写入传输数据。
SBIxCR1	← 1	0	0	0	0	x	x	x	开始传输。

INTSBIx中断

SBIxDBR	← x	x	x	x	x	x	x	x	写入传输数据。
---------	-----	---	---	---	---	---	---	---	---------

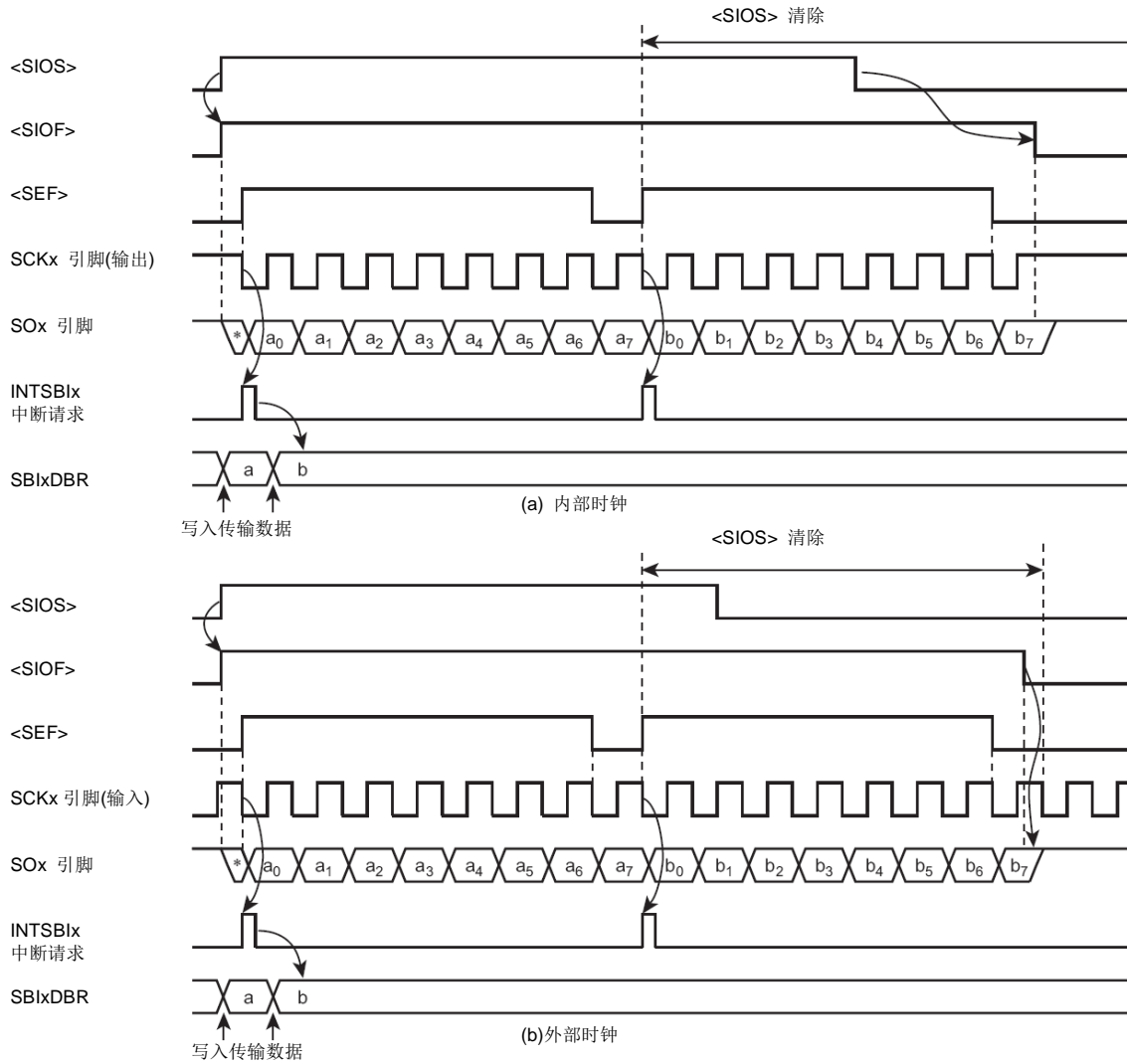


图 12-16 传输模式

示例：通过<SIOS>编程(外部时钟)以终止传输示例

```

    7 6 5 4 3 2 1 0
    → if SBIxSR<SIOF> ≠ 0          确认传输完成。
      则
    → if SCK ≠ 1                    通过监控端口，确认"1"设至SCK 引脚。
      则
    SBIxCR1 ← 0 0 0 0 0 1 1 1      设置<SIOS> = 0，完成传输。
  
```

12.9.2.2 8-位接收模式

将控制寄存器设为接收模式。然后，向 SBIxCR1<SIOS>写入"1"以启用接收。数据连同最小有效位(LSB)首先从 SIx 引脚送入移位寄存器与串行时钟同步。移位寄存器加载 8-位后，传送和接受数据至 SBIxDBR，INTSBIx (缓冲器-满)产生中断请求，请求读取接受数据。然后，中断服务程序自 SBIxDBR 读取接受数据。

内部时钟模式中，串行时钟将停止，自动处于等待状态，直至接受数据自 SBIxDBR 读取。

外部时钟模式中，和外部时钟同步执行移位工作。视乎产生中断请求时和读取接受数据之间的最大延迟，最大数据传送速率各不相同。

INTSBIx 中断服务程序中将<SIOS> 清至"0"或将<SIOINH>设为"1"，可终止接收。若 <SIOS> 清除，接收继续，直至所有接受数据位写入 SBIxDBR。程序检查 SBIxSR<SIOF>以判定接收是否已经终止。接收结束时，<SIOF>清除至"0"。确认接收完成后，最后接受数据读取。若<SIOINH>设为"1"，接收立终止，<SIOF>清除至"0"。(接受数据变为无效，无需读出。)

注：改变传送模式后，SBIxDBR内容将不保留。必须通过将<SIOS> 清除至"0"来完成正在进行的接收，且必须在传送模式改变之前读取最后接收的数据。

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	1	0	x	x	x	选择接收模式。
SBIxCR1	←	1	0	1	1	0	x	x	x	开始接收。

INTSBIx中断

Reg.	←	SBIxDBR	读取接受数据。
------	---	---------	---------

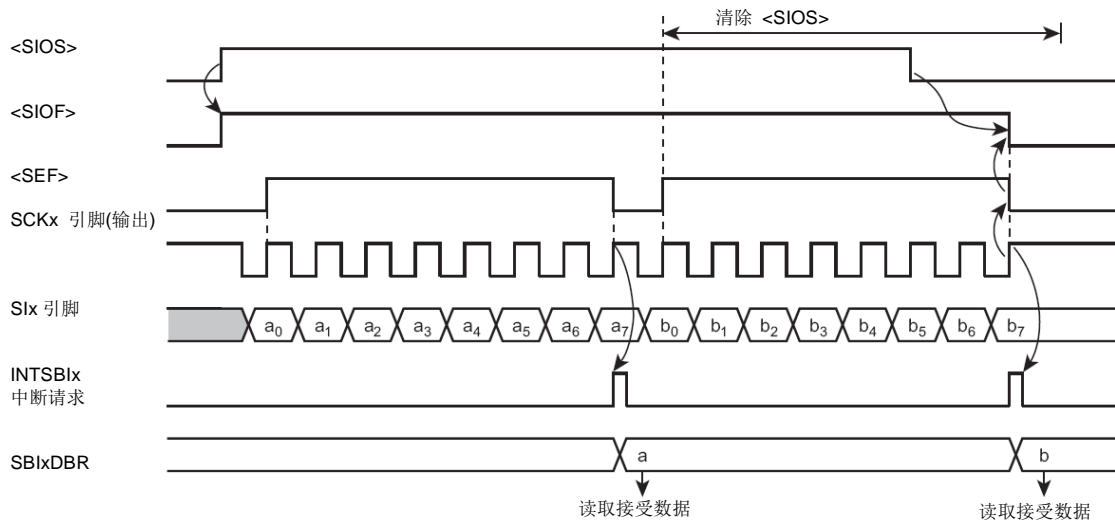


图 12-17 接收模式(示例：内部时钟)

12.9.2.3 8-位传输/接收模式

将控制寄存器设为传输/接收模式。则向 SBIxDBR 写入传输数据，并将 SBIxCR1<SIOS> 设置至"1"以启用传输和接收。传输数据会通过串行时钟的下降沿上的 SOx 引脚输出，且所接收的数据会连同最小有效位(LSB)通过串行时钟的上升沿上的 SI 引脚首先送入。移位寄存器加载 8-位数据时，会向 SBIxDBR 传送所接收的数据，INTSBIx 中断请求生成。中断服务程序可读取从数据缓冲寄存器接收的数据，并写入下一传输数据。SBIxDBR 共用于传输与接收工作，因此，必须在下一传输数据写入之前读取所接收的数据。

内部时钟工作中，串行时钟将自动处于等待状态，直至接受数据读取，下一传输数据写入。

外部时钟工作中，移位工作和外部串行时钟同步执行。因此，必须在下一次移位运算启动之前读取所接收的数据，并写入下一传输数据。外部时钟工作的最大数据传送速率，可随中断请求生成的时间和传输数据写入的时间两者之间的最大等待时间而变化。

在传输开始时，与早先传输数据最后位相同的值，会在从将<SIOF>设置至"1"到 SCKx 的下降沿之间的周期内输出。

INTSBIx 中断服务程序中，将<SIOS>清除至"0" 或将 SBIxCR1<SIOINH> 设为"1"，可终止传输和接收。若<SIOS>清除，传输和接收继续，直至接受数据完全传输至 SBIxDBR。程序将检查 SBIxSR<SIOF>，以判断传输和接收是否已结束。在传输和接收结束时，<SIOF>清除至"0"。如果<SIOINH>设置至"1"，则传输和接收会当异常终止，且<SIOF>随清除至"0"。

注：改变传送模式后，SBIxDBR 内容将不保留。必须在传送模式改变之前，通过将<SIOS>清除至"0"完成正在进行的传输和接收，并读取所接收的数据。

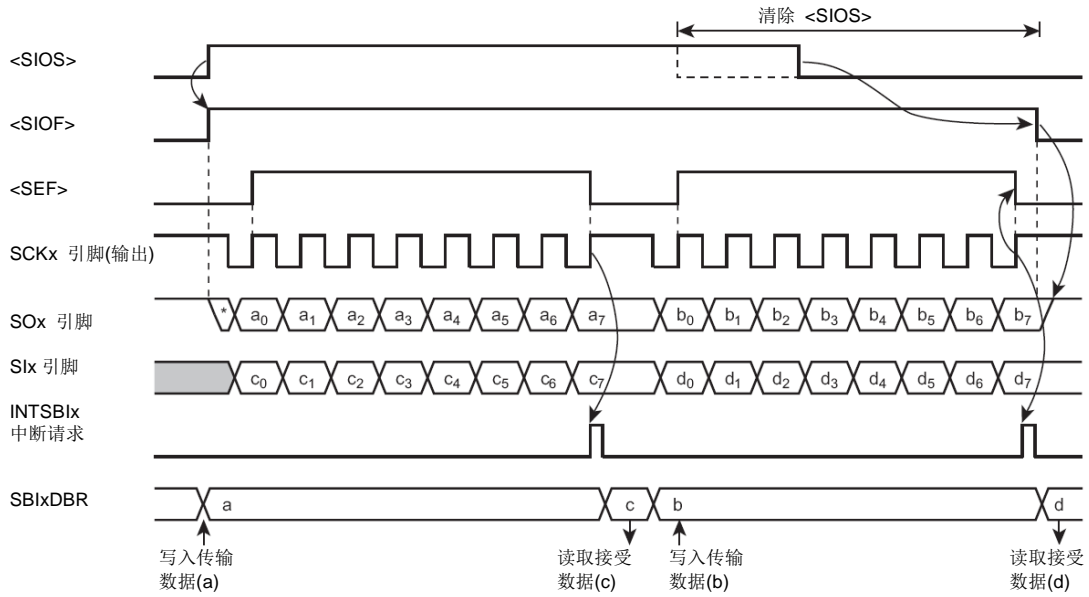


图12-18 传输/接收模式(示例：内部时钟)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	0	0	x	x	x	选择传输模式。
SBIxDBR	←	x	x	x	x	x	x	x	x	写入传输数据。
SBIxCR1	←	1	0	1	0	0	x	x	x	开始接收/传输。

INTSBIx中断

Reg.	←	SBIxDBR		读取接受数据					
SBIxDBR	←	x	x	x	x	x	x	x	写入传输数据。

12.9.2.4 传输结束时最后位的数据保留时间

在 SBIxCR1<SIOS>= "0"的条件下，传输数据的最后位可按以下所述保有 SCK 上升沿的数据。传输模式和传输/接收模式相同。

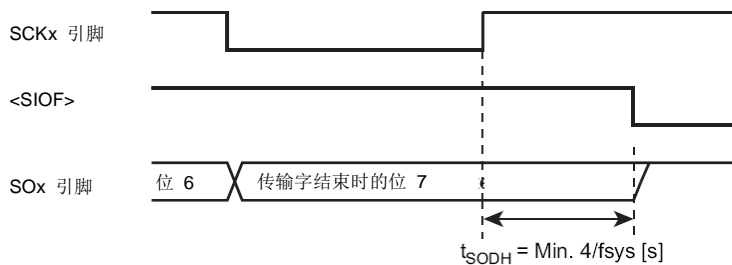


图 12-19 传输结束时最后位的数据保持时间

13. 10-位模拟/数字转换器(ADC)

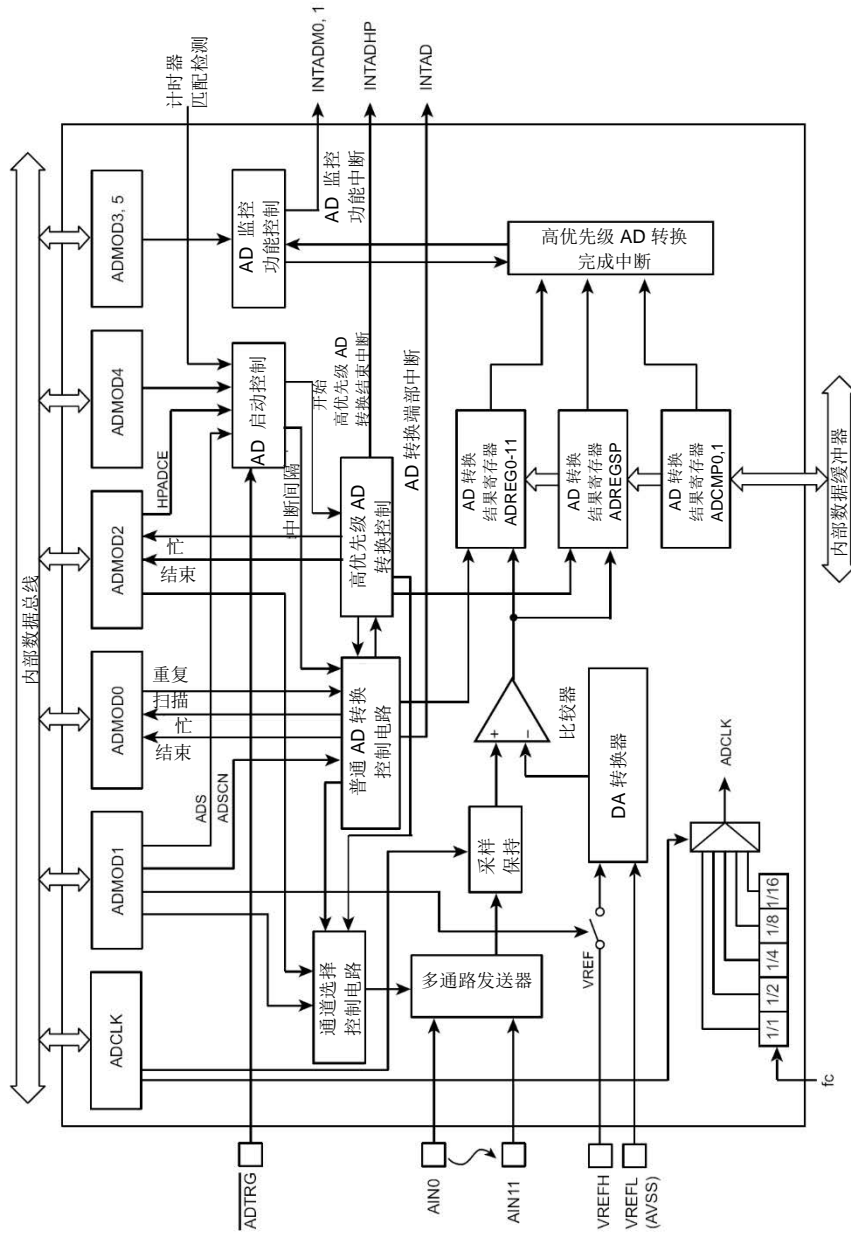
13.1 概述

TMPM061FWFG带有一个内置式 10-位，顺序转换，模拟/数字转换器(AD转换器)。

有关详细，请参照“产品信息”以确认各可用通道和设置。

13.2 配置

图 13-1 给出了AD转换器的方块图。



注：VREFL和AVSS处于共享状态。

图 13-1 10-位AD转换器方块图

13.3 寄存器

13.3.1 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，参照“存储器地址”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
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

13.3.2 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	类型	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
5-3	-	R	读作 0。
2-0	ADCLK[2:0]	R/W	选择AD转换时钟(注) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101-111: 保留

注：AD转换期间切勿改变AD转换设置。

需进行时钟计数以满足下述条件。

VREFH AVDD	转换时间
2.7 ~ 3.6 V	16.2 μs或以上
1.8 ~ 3.6 V	32.4μs或以上

13.3.3 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	类型	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 位为"0", 读取时清除。
6	ADBFN	R	普通AD转换BUSY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中断 00: 每隔 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: 设置禁止 仅在固定通道重复模式(<REPEAT> = "1", <SCAN> = "0")中规定时有效
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 如选择通道扫描模式, 则应将通道编号设置至ADMOD1<ADSCAN>。
0	ADS	W	启动AD转换开始 0: 忽略 1: 开始转换 必须在设置模式后进行转换。 总读作"0"。

13.3.4 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	类型	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 模式中指定工作模式 0: 停止 1: 工作
5-4	ADSCN[1:0]	R/W	规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 在通过ADMOD0<SCAN>选择通道扫描模式时, 应指定工作模式。 通过<ADCH>的设置选择转换通道。请参照下表。
3-0	ADCH[3:0]	R/W	选择模拟输入通道(请参照下表。)

注: 在启动AD转换之前, 将“1”写入<VREFON>位, 等待 3 μs(在此时段内, 内部参考电压应趋于稳定), 然后向ADMOD0<ADS>写入“1”。

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	-	-	-	-

13.3.5 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	类型	功能
31-8	-	R	读作 0。
7	EOCFHP	R	最高优先级AD转换完成标志(注) 0: 转换前或期间 1: 完成
6	ADBFHP	R	最高优先级AD转换BUSY标志 0: 转换暂停期间 1: 转换期间
5	HPADCE	R/W	激活最高优先级转换 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	在激活最高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

13.3.6 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	类型	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: 若转换结果值小于比较寄存器 0, 生成中断。 1: 若转换结果值大于比较寄存器 0, 生成中断。
4-1	ADREGS0[3:0]	R/W	在使用AD监控功能 0 (见下表)时, 选择一个目标转换结果寄存器。
0	ADOBSV0	R/W	AD监控功能 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	ADREG08	0100	ADREG4C
0001	ADREG19	0101	ADREG5D
0010	ADREG2A	0110	ADREG6E
0011	ADREG3B	0111	ADREG7F
-	-	1xxx	ADREGSP

13.3.7 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	类型	功能
31-8	-	R	读作 0。
7	HADHS	R/W	激活最高优先级AD转换的H/W源 0: 外部触发 1: 与定时器寄存器匹配(注 1)
6	HADHTG	R/W	激活最高优先级AD转换的H/W 0: 禁用 1: 启用
5	ADHS	R/W	标准AD转换激活用H/W源(注 2) 0: 外部触发 1: 与定时器寄存器匹配(注 1)
4	ADHTG	R/W	标准AD转换激活用HW 0: 禁用 1: 启用
3-2	-	R	读作 0。
1-0	ADRST[1:0]	W	用 01 重写 10, 允许ADC执行软件复位(注 3)。

注 1: 有关详细, 请参看“产品信息”以确认H/W源。

注 2: 在外部触发器被用于最高优先级AD转换的H/W激活时, 不得将其用于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。

13.3.8 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	类型	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AD监控功能 1 (见下表)时, 选择一个目标转换结果寄存器。
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	ADREG08	0100	ADREG4C
0001	ADREG19	0101	ADREG5D
0010	ADREG2A	0110	ADREG6E
0011	ADREG3B	0111	ADREG7F
-	-	1xxx	ADREGSP

13.3.9 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	类型	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互关系信息，请参看 13-2 和表 13-3 “13.4.5.7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	溢位标志 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：该寄存器必须以半字或整字存取。

13.3.10 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	类型	功能
31-16	-	R	读作 0。
15-6	ADRSP[9:0]	R	AD转换结果 最高优先级AD转换结果已储存。
5-2	-	R	读作 0。
1	OVRSP	R	溢位标志 0: 未生成 1: 生成 如果在读取<ADRSP>之前转换结果被重写, 则设置 "1"。 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: 该寄存器必须以半字或整字存取。

13.3.11 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	类型	功能
31-16	-	R	读作 0。
15-6	ADCOM0[9:0]	R/W	在AD监控功能 0 已被启用时, 其可设置某个值, 用于与通过ADMOD3<ADREGS0>指定的转换结果寄存器的值进行比较。
5-0	-	R	读作 0。

注: 在将各值写入该寄存器时, 必须禁用AD监控功能 0 (ADMOD3<ADBSV0>="0")。

13.3.12 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	类型	功能
31-16	-	R	读作 0。
15-6	ADCOM1[9:0]	R/W	在AD监控功能 1 启用时, 其可设置某个值, 用以与通过ADMODt<ADREGS1>指定的转换结果寄存器的值进行比较。
5-0	-	R	读作 0。

注: 在将各值写入到寄存器时, 必须禁用AD监控功能 1 (ADMOD5<ADBSV1>="0")。

13.4 工作描述

13.4.1 模拟参考电压

模拟参考电压的"高"电平应用于VRFEH引脚，而"低"电平应施加到VREFL引脚。

启动AD转换时，务必首先将"1"写入<VREFON>位，等待3 μs，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，则可向ADMOD1<DACON>写入"0"。这样可减少模拟电路的功耗电流。

注：VREFL和AVSS由TMPM061FWFG共享。

13.4.2 AD 转换模式

支持两种 AD 转换：普通 AD 转换和最高优先级 AD 转换。

对于普通AD转换，支持以下四种工作模式。

13.4.2.1 普通 AD 转换

对于标准 AD 转换，均支持以下四种工作模式，用户可通过 ADMOD0 <REPEAT> <SCAN>选择相关的工作模式。

- 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，支持以下三种模式，用户可通过 ADMOD1<ADSCN>选择相关的工作模式。

- 4-通道扫描模式
- 8-通道扫描模式
- 12-通道扫描模式

(1) 固定通道单次转换模式

如果 ADMOD0<REPEAT, SCAN>设置为"00"，则会按固定通道单次转换模式进行 AD 转换。

在该模式下，只要选择一条通道就会执行 AD 转换。在 AD 转换完成之后，ADMOD0<EOCFN>即设置为"1"，ADMOD0<ADBFN>即清除为"0"，且 AD 转换完成中断请求(INTAD)即生成。<EOCFN>一旦读取则清除到"0"。

(2) 通道扫描单次转换模式

如果 ADMOD0<REPEAT, SCAN>设置为"01"，则会按通道扫描单次转换模式进行 AD 转换。

在该模式下，只要选择一条通道就会执行 AD 转换。在 AD 扫描转换完成之后，ADMOD0<EOCFN>即设置为"1"，ADMOD0<ADBFN>即清除为"0"。且转换完成中断请求(INTAD)即生成。<EOCFN>一旦读取则清除到"0"。

(3) 固定通道重复转换模式

如果 ADMOD0<REPEAT, SCAN>设置为"10", 则会按固定通道重复转换模式进行 AD 转换。

在该模式下, 可为所选定的一条通道重复执行 AD 转换。在 AD 转换完成之后, ADMOD0<EOCFN>即设置为"1"。ADMOD0<ADBFN>不清除至 "0"。其值仍保留为"1"。可通过对 ADMOD0<ITM>进行适当的设置, 即可选择转换完成中断请求 INTAD 生成时所依据的计时。在中断 INTAD 生成时, 用相同的计时设置<EOCFN>。

<EOCFN>一旦读取则清除到 "0"。

(4) 通道扫描重复转换模式

如果 ADMOD0<REPEAT, SCAN>设置为"11", 则会按通道扫描重复转换模式进行 AD 转换。

在该模式下, 可为所选定的一条扫描通道重复执行 AD 转换。每完成一次 AD 扫描转换, ADMOD0<EOCFN>即设置为 "1" 一次, 且转换完成中断请求 (INTAD) 即生成一次。ADMOD0<ADBFN>不清除至"0"。保留为"1"。<EOCFN>一旦读取则清除到 "0"。

13.4.2.2 最高优先级 AD 转换

中断进行中的标准 AD 转换, 即可执行最高优先级 AD 转换。

不管 ADMOD0<REPEAT, SCAN>的设置为何, 都会自动选择固定通道单次转换。在满足启动工作的条件时, 会为 ADMOD2<HPADCH>所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)即生成, 用以指示 AD 转换完成的 ADMOD2<EOCFHP>即设置为 "1"。<ADBFHP>会返回为 "0"。一经读取, EOCFHP 标志即清除至 "0"。

对最高优先级 AD 转换运行时激活的最高优先级模拟数字转换, 则不予理会。

13.4.3 AD 监控功能

有两条AD监控程序功能通道。

如果ADMOD3<ADOBSV0>和ADMOD5<ADOBSV1>设置为"1", 则AD监控功能即启用。如果 ADMOD3<ADREGS0>和ADMOD5<ADREGS1>所指定的转换结果寄存器的值变为较大或较小("较大"或"较小"需由ADMOD3<ADOBIC0>和ADMOD5<ADBIC1>指定)比较寄存器的值, 则AD监控功能中断(INTADM0, INTADM1)即生成。每将一个结果存储到相应的转换结果寄存器中一次, 就要执行一次比较工作。

如果分配执行AD监控功能的转换结果寄存器连续使用而不读取转换结果, 其转换结果则覆盖。转换结果储存标志<ADR_xRF>和溢位标志<OVR_x>则仍保持在所设置的状态。

13.4.4 选择输入通道

输入通道的选择方式，可随拟使用 AD 转换器工作模式的不同而不同。

1. 标准 AD 转换模式

- 如果是在固定状态(ADMOD0<SCAN> = "0")使用模拟输入通道

通过对 ADMOD1<ADCH>进行适当的设置，从各模拟输入引脚中选择一条通道。

- 如果是在扫描状态(ADMOD0<SCAN> = "1")使用模拟输入通道

通过对 ADMOD1 <ADCH>和 ADSCN>进行适当的设置，从各扫描模式中选择一种扫描模式。

2. 最高优先级 AD 转换模式

通过对 ADMOD2<HPADCH>进行适当的设置，从各模拟输入引脚中选择一条通道。

13.4.5 AD 转换 细节

13.4.5.1 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。

四种工作模式被提供给标准 AD 转换。在执行标准 AD 转换时，必须通过对 ADMOD0<REPEAT, SCAN>进行适当的设置，选择其中一种工作模式。对于最高优先级 AD 转换，仅可使用一种工作模式：固定通道单次转换模式。

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"，则可根据通过 $\overline{\text{ADTRG}}$ 引脚实现的下降沿输入，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。

应将 ADMOD4 <ADHTG>设置为"1"以进行标准 AD 转换，并将 AD MOD4 <HADHTG>设置为"1" 以进行最高优先级 AD 转换，以允许进行 H/W 激活。

软件激活 是 H / W 激活已获准后仍然是有效的。

注 1: 有些产品不提供 $\overline{\text{ADTRG}}$ 引脚

注 2: 在外部触发器用于最高优先级AD转换的HW激活源时，无法针对标准AD转换HW的激活设置外部触发器。

注 3: 有关详细，请参看“产品信息”以确认计时器的可用匹配检测。

13.4.5.2 AD 转换

在标准 AD 转换开始时，显示 AD 转换正在进行的 AD 转换忙碌标志(ADMOD0<ADBFN>)设置到 "1"。

在最高优先级 AD 转换启动时，用以指示“AD 转换进行中”的最高优先级 AD 转换忙碌标志 (ADMOD2 <ADBFHP>)即设置为"1"。

此时,最高优先级转换启动前的标准 AD 转换忙碌标志 ADMOD0<ADBFN>的值会得以保留。此时,最高优先级转换启动前的标准 AD 转换完成标志 ADMOD0<EOCFN>的值会得以保留。

注:当最高优先级AD转换正在进行时,不可激活标准AD转换。

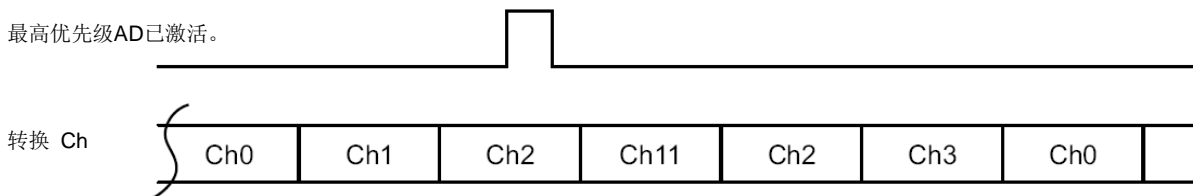
13.4.5.3 标准 AD 转换期间的最高优先级 AD 转换

如果最高优先级 AD 转换在标准 AD 转换期间已激活,则正在进行的标准 AD 转换会暂停,并会在最高优先级 AD 转换完成之后,重新启动标准 AD 转换。

如果 ADMOD2<HPADCE>在标准 AD 转换期间设置为"1",则正在进行的标准 AD 转换会暂停,最高优先级 AD 转换则启动;具体地说,会为 ADMOD2<HPADCH>指定的通道执行 AD 转换(固定通道单次转换)。在最高优先级 AD 转换的结果存储到存储寄存器 ADREGSP 中之后,标准 AD 转换即重新开始。

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 转换的结果存储到存储寄存器 ADREGSP 中之后,标准 AD 转换就会重新开始。

例如,如果为通道 AIN0~AIN3 激活了通道重复转换,且<HPADCE>已在 AIN2 转换期间设置为"1",则 AIN2 会暂停,并会针对<HPADCH>(在以下个案中为 AIN11)所指定的通道执行转换。在转换结果存储到 ADREGSP 中之后,通道重复转换会从 AIN2 重新开始。



13.4.5.4 停止重复转换模式

向 ADMOD0<REPEAT>写入"0",即可停止重复转换模式下的 AD 转换工作(固定通道重复转换模式,或通道扫描转换模式)。在进行中的 AD 转换完成时,重复转换模式终止,ADMOD0<ADBFN>即设置为"0"。

13.4.5.5 重新激活标准 AD 转换

如需在转换进行期间重新激活标准 AD 转换,必须在 AD 转换转换启动之前执行软件复位 (ADMOD3 <ADRST>)。不得将 H/W 激活法用于标准 AD 转换的重新激活。

13.4.5.6 转换完成

(1) 正常 AD 转换完成

在正常 AD 转换完成时，AD 转换完成中断 (INTAD)即生成。在 AD 转换的结果存储到存储寄存器，和改变两个寄存器：用以指示 AD 转换完成的寄存器 ADMOD0 <EOCFN>，以及寄存器 ADMOD0 <ADBFN>。

有关详细，请参看表 13-2 和表 13-3 以确认转换模式所对应的存储寄存器。

以下给出了中断请求和标志变化等内容。

- 固定通道单次转换模式

在 AD 转换完成之后，ADMOD0<EOCFN>即设置为"1"，ADMOD0<ADBFN>即清除为"0"，且中断请求即生成。

- 通道扫描单次转换模式

在通道扫描转换完成之后，ADMOD0<EOCFN>即设置为"1"，ADMOD0<ADBFN>即设置为"0"，且中断请求 INTAD 即生成。

- 固定通道重复转换模式

ADMOD0<ADBFN>不清除至"0"。保留为"1"。可通过对 ADMOD0<ITM>进行适当的设置，即可选择中断请求 INTAD 生成时所依据的计时。用与中断 INTAD 生成时相同的计时设置 ADMOD0<EOCFN>。

- 通道扫描重复转换模式

每当 AD 扫描转换完成一次，ADMOD0 <EOCF> 即设置为"1"，中断请求 INTAD 即生成。ADMOD0<ADBFN>不清除至"0"。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)即生成，可指示最高优先级 AD 转换完成情况的 ADMOD2<EOCFHP>即设置为"1"。

AD 转换结果 被存在 AD 转换结果寄存器 SP。

(3) 数据轮询

确认 AD 转换完成,使用中无被打断，数据轮询 可被使用。在 AD 转换完成时，ADMOD0<EOCFN>设置到"1"。确认 AD 转换完成和获得结果，查询此位。

必须通过半字或整字存取来读取 AD 转换结果存储寄存器。如果<OVRx> ="0"且<ADR_xRF> = "1"，则已获得正确的转换结果。

13.4.5.7 中断生成计时和 AD 转换的结果存储寄存器

表 13-1 给出了以下三项之间的关系：AD 转换模式，中断生成计时和标志工作。表 13-2 和表 13-3 给出了模拟通道输入和 AD 转换结果寄存器之间的关系。

表 13-1 转换模式，中断生成计时和标志工作之间的关系

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> (在中断之后 生成。)	<ADBFHP>
正常 转换	固定通道 单次转换	0	0	-	生成完成之后。	转换完成之后。	0	-
	固定通道重复转换	1	0	00	每完成一次 转换。	在一次转换完成之后。	1	-
				01	每次完成四个转 换时。	完成四个转换之后。	1	-
				10	每次完成八个转 换时。	在8次转换完成后。	1	-
	通道扫描单次转换	0	1	-	在扫描转换完成 之后。	在扫描转换完成之后。	0	-
	通道扫描重复转换	1	1	-	在一次扫描转换 完成之后。	在一次扫描转换完成 之后。	1	-
最高优先级转换	-	-	-	转换完成之后。	转换完成	-	0	

注：ADMOD0<EOCFN>与ADMOD2<EOCFHP>一经读取即清除。

表 13-2 结果寄存器(固定通道重复转换模式)

<ITM[1:0]>	结果寄存器
00 每隔 1 次转换生成一次中断	ADREG0
01 每 4 次转换时，产生中断一次	ADREG0 ~ ADREG3
10 每 8 次转换时，产生中断一次	ADREG0 ~ ADREG7

表 13-3 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	-	-

注意事项

AD转换的结果值可以根据电源电压的波动而变化，或者可以受噪音影响。

当使用模拟输入引脚和端口交替进行，由于转换精度可能会降低，转换过程中不可读取和写端口。如果在AD转换输出端口电流波动减小，转换精度也可能会降低。

请利用此程序采取对应措施，如对AD转换结果进行平均化。

14. 24-位 $\Delta\Sigma$ 模拟/数字转换器 (DSADC)

温度传感器可共享DSADC所采用的参考电压电路(BGR)，并需设置温度传感器的控制寄存器(TEMPEN)。

14.1 特征

DSADC具备以下特征：

- 转换启动
 - 软件通过启动转换
- 转换模式
 - 单次转换
 - 重复转换
- 状态标志
 - 转换结果存储标志
 - 溢位标志
 - 转换结束标志
 - 转换标志
- 可按以下比率对转换时钟进行分频：
 - fc/1, fc/2, fc/4, fc/8
- 转换结束中断输出
- 转换启动校正功能
- 多个单元用同步启动功能

在使用DSADC时，应按以下所述进行引脚处理：

- 请勿将VREFIN_x连接到参考电压。
- 将AGNDREF_x连接到DVSS电平。
- 将 1 μ F 电容器连接到VREFIN_x和AGNDREF_x之间。在不使用DSADC时，需进行以下设置。
- 将AGNDREF_x连接到DVSS电平。

同时在不使用温度传感器时，必须对参考电压电路进行以下设置。

- 将DSRVDD3 和SRVDD3连接到DVDD3。
- 将DSRVSS3连接到DVSS。

14.2 方块图

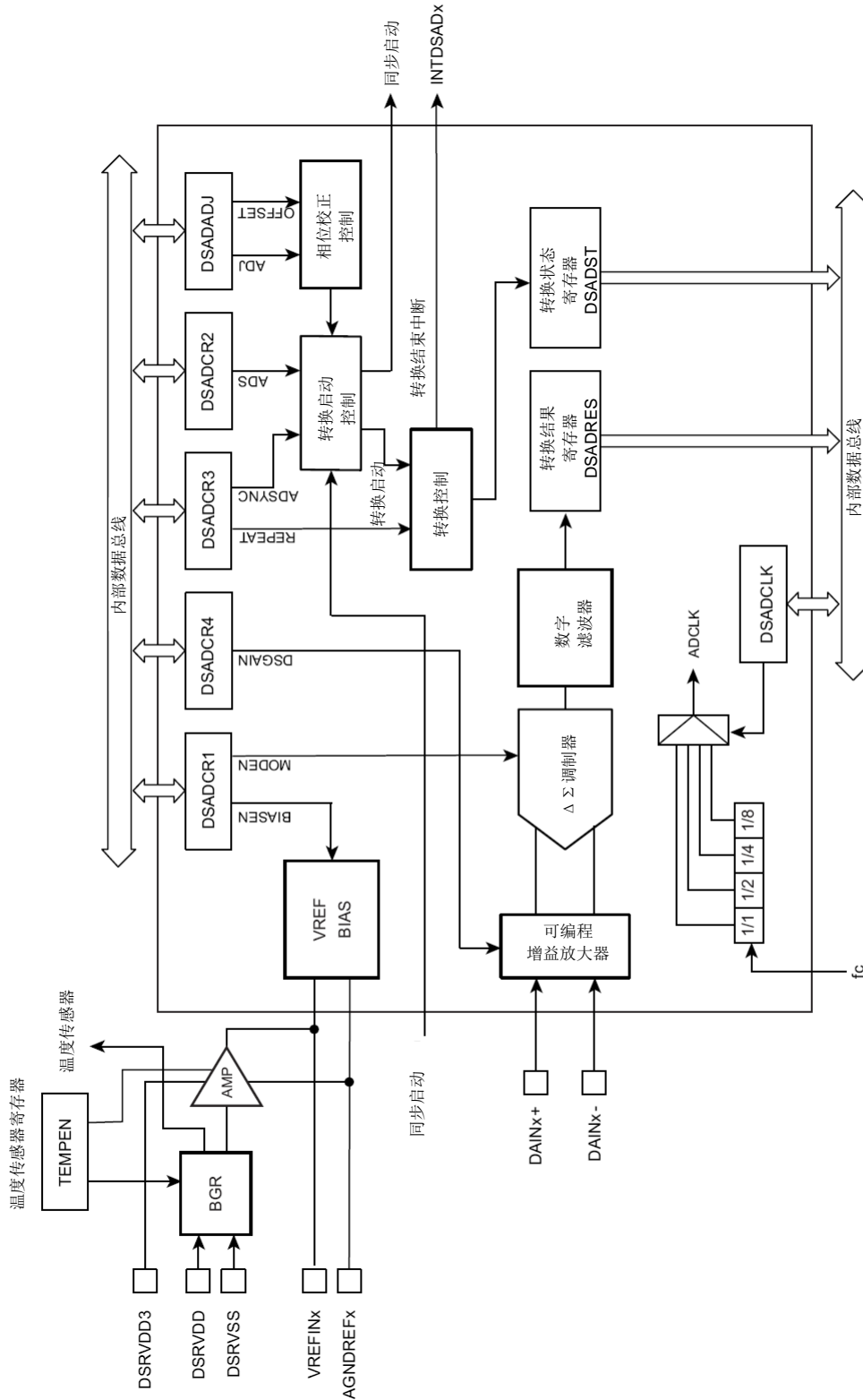


图 14-1 24-位ΔΣAD转换器方块图

14.3 寄存器

14.3.1 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“存储器地址一章的“外部功能地址表”。

寄存器名称		地址(基+)
时钟设置寄存器	DSADCLK	0x0000
控制寄存器 0	DSADCR0	0x0004
控制寄存器 1	DSADCR1	0x0008
控制寄存器 2	DSADCR2	0x000C
控制寄存器 3	DSADCR3	0x0010
控制寄存器 4	DSADCR4	0x0014
校正寄存器	DSADADJ	0x0030
转换状态寄存器	DSADST	0x0040
转换结果存储寄存器	DSADRES	0x0044

14.3.2 寄存器详细

14.3.2.1 DSADCLK (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	-	-	-	-	-	ADCLK		
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"。
2-0	ADCLK[2:0]	R/W	AD转换时钟选择 000: fc/1 001: fc/2 010: fc/4 011: fc/8 100-111: 禁止

注1: 在“DSADCR1<BIASEN>=<MODEN>=0 且AD转换停止”的条件下，改变<ADCLK[2:0]>寄存器。

注2: 在使用同步启动功能时，所有单元中均应选择相同的转换时钟。



14.3.2.2 DSADCR0(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	-	-	-	-	-	-	ADRST	
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-2	-	-	读作 "0"。
1-0	ADRST	W	软件复位(注) 写入"10"和"01"以生成软件复位。内部电路和寄存器DSADCLK除外初始化。

注：仅当DSADCR1<BIASEN>="1"时有效。

14.3.2.3 DSADCR1(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	-	-	-	-	-	-	BIASEN	MODEN
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-2	-	-	读作 "0"。
1	BIASEN	R/W	偏置控制 0: 停止 1: 工作
0	MODEN	R/W	调制器控制 0: 停止 1: 控制

14.3.2.4 DSADCR2 (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	-	-	-	-	-	-	-	ADS
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-1	-	-	读作 "0"。
0	ADS	W	转换启动 1: 开始转换 设置"1"以启动转换。写入"0"无效。 读作 "0"。

14.3.2.5 DSADCR3 (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	-	-	-	-	-	-	-	ADSYNC
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	REPEAT
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-9	-	-	读作 "0"。
8	ADSYNC	R/W	同步模式 0: 异步工作 1: 同步工作 设置"1"以在多个单元中同时启动转换。(注)
7-1	-	R	读作 "0"。
0	REPEAT	R/W	转换模式 0: 单次转换 1: 重复转换 指定一种转换模式。如果在重复转换模式工作期间设置单次转换，则转换将自动停止。

注：将"1"设置至仅用于从设备的单元设置将主设备单元设置为"0"。有关本产品的主设备通道和从设备通道组合，请参看"产品信息"一节。

14.3.2.6 DSADCR4 (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	-	-	-	-	-	DSGAIN		
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-3	-	-	读作 "0"。
2-0	DSGAIN[2:0]	R/W	放大器设置 000: ×1 001: ×2 010: ×4 011: ×8 100: ×16 101-111: 保留

14.3.2.7 DSADADJ (校正寄存器)

	31	30	29	28	27	26	25	24
比特符号	OFFSET							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	OFFSET							
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	-	-	-	-	-	-	-	ADJ
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	OFFSET	R/W	转换启动校正时间 设置从将"1"设置到DSADCR2<ADS>至启动转换之间的时间周期。 新增延迟<OFFSET>x1/fsys。
15-1	-	R	读作 "0"。
0	ADJ	R/W	转换启动校正 0: 无校正 1: 校正 在用<OFFSET>设置的时间延迟消逝后从转换启动起计, 设置"1"以启动转换。

14.3.2.8 DSADST (转换状态寄存器)

	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	-	-	-	-	-	-	EOCF	ADBF
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	EOCF	R	转换结束标志(注) 0: 转换进行中 1: 转换结束
0	ADBF	R	转换标志 0: 无转换启动 1: 转换启动

注: 通过读取DSADST寄存器清除该位。

14.3.2.9 DSADRES (转换结果寄存器)

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

位	比特符号	类型	功能
31-8	ADR[23:0]	R	转换结果 转换结果按两者的余数格式存储。 如果在转换期间读取该位，则会读取前一转换结果。
7-2	-	R	读作 "0"。
1	ADOVR	R	溢位标志(注) 0: 无生成 1: 生成 如果在读取<ADR>之前转换结果重写，则设置"1"。
0	ADRF	R	转换结果存储标志(注) 0: 无结果存储。 1: 结果被存储。 如果某转换结果存储在<ADR>中，则设置"1"。

注：通过读取DSADRES寄存器清除该位。

14.4 工作说明

14.4.1 启动和停止步骤

本节说明了DSADC启动步骤，以及转换到低功耗模式时需执行的DSADC停止步骤。下表给出了需设置的各寄存器。

寄存器	位	说明
TEMPEN	EN0, EN1	参考电压电路(注)
CGSYSCR	FCSTOP	时钟馈送对DSADC和ADC
DSADCLK	DSADCLK	转换时钟的分频
DSADCR4	DSGAIN	增益设置
DSADCR1	BIASEN, MODEN	偏置电路和调制器电路工作

注：温度传感器可共享参考电压电路。

14.4.1.1 启动

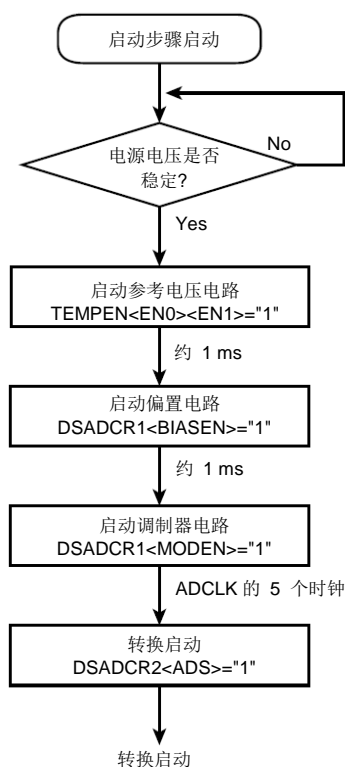


图14-2 启动步骤

在电源电压稳定时执行以下步骤。

1. 启动参考电压电路

向 TEMPEN<EN0><EN1>设置"1"，并等待 1ms 或以上直至稳定。

在 TEMPEN<EN0>启用时，TEMPEN<EN1>必须启用。能同时设置<EN0>和<EN1>。



2. 启动偏置电路

向 DSADCR1<BIASEN>设置"1", 并等待 1 ms 或以上直至稳定。

在设置 DSADCR1<BIASEN>之前, 馈送一次转换时钟。有关转换时钟的详细, 请参看“1.4.2 转换时钟(ADCLK)”。

3. 启动调制器电路

向 DSADCR1<MODEN>设置"1"。

转换将在 ADCLK 消逝 5 个时钟之后启用。在启动转换之前, 设置转换模式 (DSADCR3<REPEAT>)和增益调整 (DSADCR4<DSGAIN>)。

14.4.1.2 停止

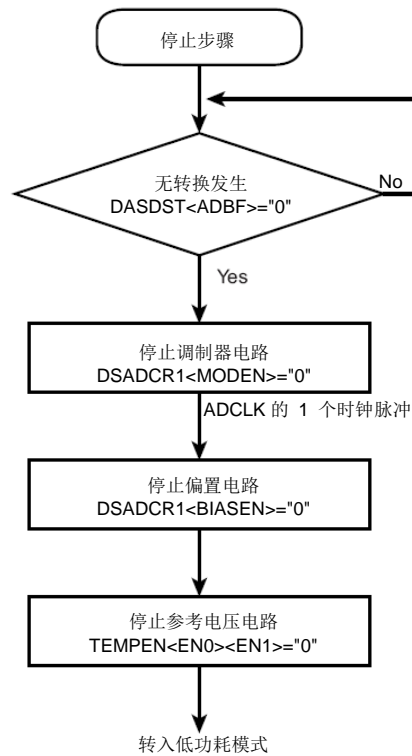


图14-3 停止步骤

在转换完成的同时执行以下步骤。(DSADST<ADBF>="0")

1. 停止调制器电路

向 DSADCR1<MODEN>设置"0"

2. 停止偏置电路

在 ADCLK 已消逝一个时钟或以上之后从调制器电路停止时起计, 向 DSADCR1<BIASEN>设置"0"。

3. 停止参考电压生成电路

向 TEMPEN<EN0><EN1>设置"0"



注：温度传感器可共享参考电压电路。在温度传感器运行期间，请勿停止TEMPEN<EN0> ("0"设置为停止)。

14.4.2 转换时钟(ADCLK)

顺序比较型AD转换器可共享馈送DSADC的转换时钟，且转换时钟可在复位之后运行。可用CGSYSCR<FCSTOP>停止时钟，并可用DSADCLK对其进行分频。

在DSADC处于停止状态时，改变或停止转换时钟 (DSADCR1<BIASEN>和<MODEN>均设置为"0")。

在使用同步启动功能时，所有单元中均应选择相同的转换时钟。

14.4.2.1 转换时间

可利用以下公式计算转换时间，其中 ADCLK 的频率为 f_{ADCLK} 。

$$\text{转换时间} = 1 / f_{ADCLK} \times 2640 + \text{固定延迟时间}[s]$$

对于重复转换中的首次转换和单次转换而言，固定延迟时间为 673 至 675 次时钟。在重复转换时，第二次转换或其后的转换的固定延迟时间为 0 次时钟。

例如，如果 $f_c/1$ 的选择按 $f_c=16 \text{ MHz}$ ，则第二次转换或其后转换所需的转换时间将是 $165 \mu\text{s}$ 。

14.4.2.2 转入低功耗模式

在发生指向 SLOW 模式的转换时，按“14.4.1 启动和停止步骤”所述的步骤停止 DSADC，然后用CGSYSCR<FCSTOP>停止对 DSADC 的时钟馈送。

在发生指向 STOP 或 SLEEP 模式的转换时，则应按停止步骤停止 DSADC。对 DSADC 的时钟馈送会自动停止。

14.4.3 转换模式

DSADC 可提供单次模式和重复模式等两种转换模式。在单次模式下，转换仅执行一次，而在重复模式下则连续执行转换。可用 DSADCR3<REPEAT>设置模式。

14.4.4 转换启动

向 DSADCR2<ADS>设置"1"可启动转换。

14.4.5 转换状态

可用 DSADST 检查转换状态。

在转换期间，DSADST<ADBF>变为"1"。转换后，DSADST<EOCF>变为"1"。读取 DSADST 可清除<EOCF>。

在进行重复转换时，DSADST<ADBF>可在工作期间保持为"1"。在重复转换完成时，DSADST<ADBF>清除为"0"。

14.4.6 转换停止

在单次模式下，DSADC 将在转换完成时自动停止。

在重复模式下，向 DSADCR3<REPEAT>设置"0"可停止转换。DSADC 将暂停当前转换，然后完全停止转换。此时，不会发生转换结束中断。

注：如果通过向<REPEAT>设置"0"完成重复转换，则不要修改DSADCR3 的其它位。

14.4.7 转换结束

在转换结束时，将发生转换结束中断。转换结果会存储到 DSADRES<ADR>中，且 DSADRES<ADRF>会设置为"1"。

如未使用转换结束中断，则查询 DSADST<EOCF>。如果<EOCF>为"1"，则转换已完成。

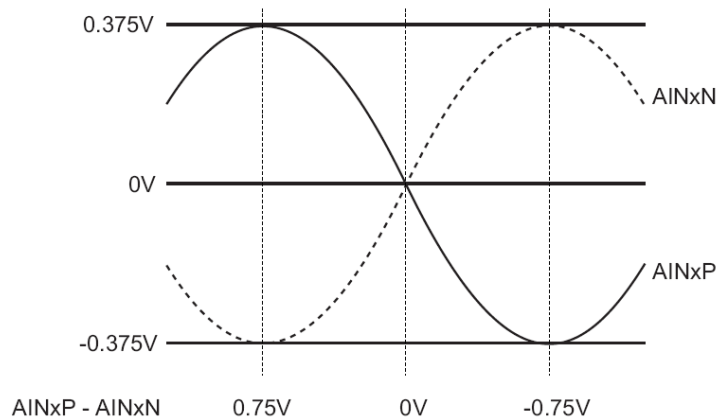
如果在读取当前值之前<ADR>写入到下一个结果，则<ADOVR>设置为"1"。读取 DSADRES，可清除<ADRF>和<ADOVR>。

14.4.8 转换结果

由于AINxP ~ AINxN的输入范围为 -0.375V~ 0.375V，最高输入电压的幅度为±0.75V。此时，转换结果如以下所述：

注：VREFINx = 2.75 V

AINxP - AINxN	转换结果
0.75 V	0x4FC2BF
0 V	0x000000
-0.75 V	0xB03D41



14.5 同步启动功能

主设备单元和从设备单元可同时启动转换。有关详细，请参看“产品信息”以确认本产品中的主设备和从设备分配情况。

14.5.1 启动

以下流程图给出了使用同步启动功能时的设置步骤。

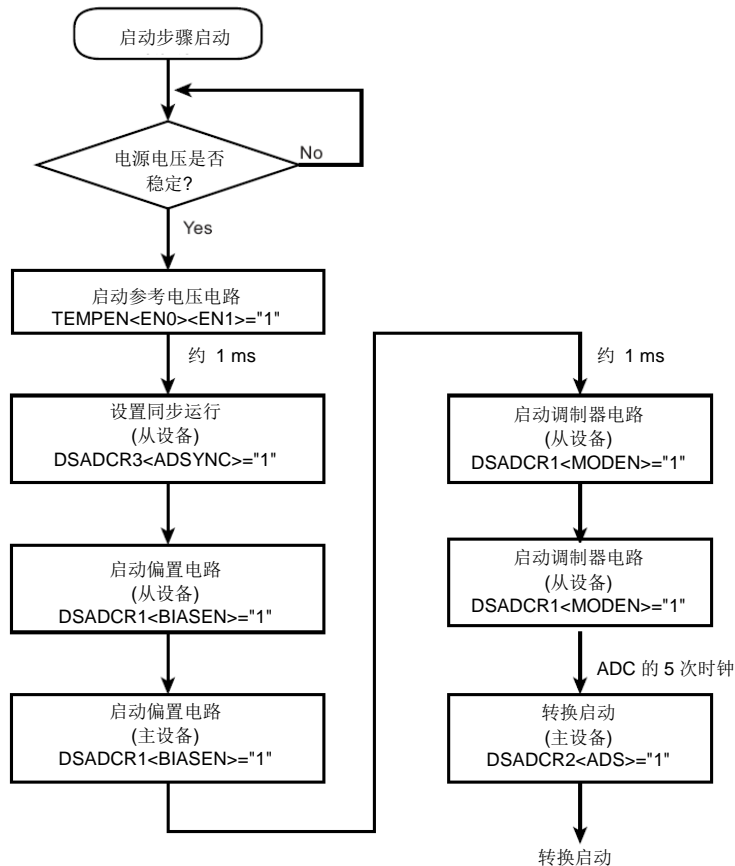


图14-4 同步启动功能步骤

1. 启动参考电压电路

向 TEMPEN<EN0><EN1>设置"1"，并等待 1 ms 或以上直至稳定。

在 TEMPEN<EN0>启用时，TEMPEN<EN1>必须启用。能同时设置<EN0>和<EN1>。

2. 设置同步运行

在使用同步启动功能时，将从设备单元 DSADCR3<ADSYNC>设置为"1"，并将主设备单元<ADSYNC>设置为"0"。

3. 启动偏置电路

向 DSADCR1<BIASEN>设置"1"，并等待 1 ms 或以上直至稳定。如果是<BIASEN>，则首先设置从设备侧，然后设置主设备侧。

在设置 DSADCR<BIASEN>之前，馈送一次转换时钟。有关转换时钟的详细，请参看"1.4.2 转换时钟(ADCLK)"。



4. 启动调制电路

向 DSADCR1<MODEN>设置"1"。

转换将在 ADCLK 消逝 5 个时钟之后启用。

可在各单元中设置转换模式 (DSADCR3<REPEAT>和增益调整 (DSADCR4<DSGAIN>)。在转换启动之前, 在各单元中对其进行设置。

14.5.2 停止

在单次转换模式下, DSADC将在各单元中的转换完成时停止。

在重复转换模式下, 应按以下步骤停止DSADC:

- 仅停止从设备侧

改变从设备的转换模式 (将DSADCR3<REPEAT> 设置为"0")或执行一次软件复位。
此时, 主设备侧会继续执行转换。

- 仅停止主设备侧

通过将主设备的DSADCR3<ADSYNC>设置为"0"取消同步运行, 然后通过改变转换模式或执行软件复位停止主设备侧。

此时, 从设备侧会继续执行转换。

- 同时停止主设备侧和从设备侧。

停止从设备侧, 然后通过改变转换模式或执行软件复位停止主设备侧。

14.6 转换启动校正功能

通过使用转换启动校正功能, 可从DSADCR2<ADS>设置为"1"的时间起计推迟转换启动时间。

通过向DSADADJ<ADJ>设置"1", 可启用功能。可用DSADADJ<OFF SET>设置延迟时间。在延迟时间<OFFSET> \times 1/fsys消逝之后, 转换将与ADCLK同步启动。

在同步运行时, 可在各单元中将所需的延迟时间设置为<OFFSET>。在<OFFSET>中定义的时间消逝之后从<ADS>主设备侧设置为"1"时起计, 转换将启动。

在从向<ADS>设置"1"到转换启动之间的时段内, 请勿修改<OFFSET>的值。

15 温度传感器(TEMP)

15.1 概述

MCU可利用温度传感器测量相对温度。

温度传感器可按照温度，根据参考电压电路 (BGR) 输出某个电压。输出电压输入到模拟/数字转换器 (ADC)中的通道 2。利用AD转换，可求出温度的相应数字值。

(注： $\Delta\Sigma$ 型模拟/数字转换器(DSADC)可共享参考电压电路(BGR)。

温度传感器输出电压差异，为涉及到温度变化的线性差。在某些条件下收集数据，可得出相对温度。

ADC的通道 3 输入 1 V来自BGR。在可变电源电压系统中，如果BGR电压为已执行的AD转换，则可按照结果相对地确定电源电压。

如未使用温度传感器或DSADC，则参考电压电路必须采用以下设置。

- 将DSRVDD3和SRVDD连接到DVDD3
- 将DSRVSS连接到DGND

15.2 方块图

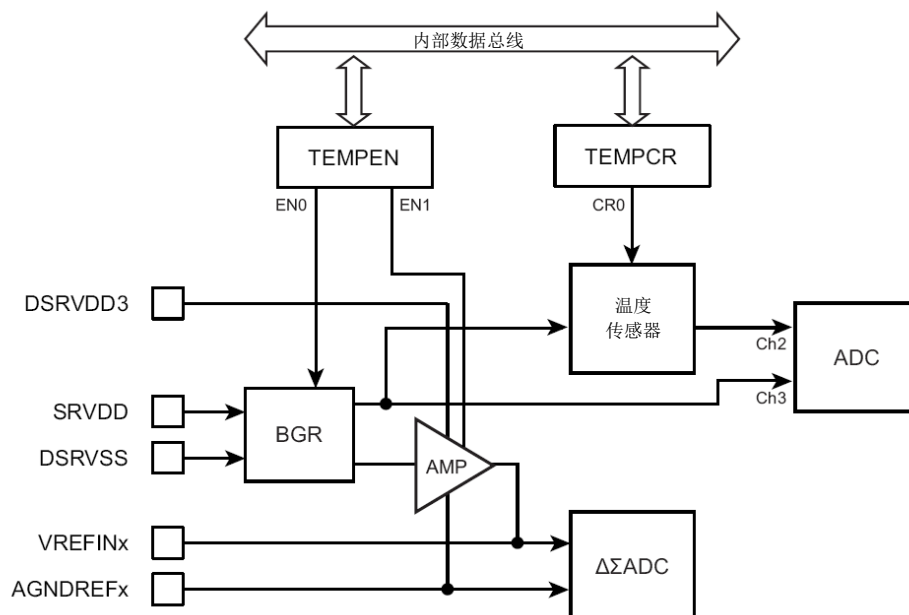


图15-1 温度传感器框图

15.3 寄存器

15.3.1 寄存器列表

下表列出了各控制寄存器及其地址。

有关基地址的详细情况，参照“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址 (基+)
启动寄存器	TEMPEN	0x0000
控制寄存器	TEMPCR	0x0004

15.3.2 寄存器详细

15.3.2.1 TEMPEN (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	-	-	-	-	-	-	EN1	EN0
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	EN1	R/W	AMP工作 0: 禁用 1: 启用 设置AMP以实现 $\Delta\Sigma$ ADC的启用/禁用。
0	EN0	R/W	BGR工作 0: 禁用 1: 启用 设置参考电压电路的启用/禁用。

注：在TEMPEN<EN0>启用时，TEMPEN<EN1>必须启用。可同时设置<EN0>和<EN1>。

15.3.2.2 TEMPCR(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	-	-	-	-	-	-	-	CR0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-1	-	-	读作 "0"。
0	CR0	R/W	温度传感器工作 0: 禁用 1: 启用 设置温度传感器的启用/禁用。

15.4 工作说明

1. 启动

在电源电压稳定时执行以下步骤。

1. 启动参考电压电路

向TEMPEN<EN0>设置"1", 并等待 1 ms或以上直至稳定。

2. 启动温度传感器

向TEMPCR<CR0>设置"1"

在约 10 μ s之后, 输出电压启用。

2. 停止

执行以下步骤。

1. 停止温度传感器

向TEMPCR<CR0>设置"0"

2. 停止参考电压电路

向TEMPEN<EN0>设置"0"

注: $\Delta\Sigma$ ADC可共享参考电压电路。在 $\Delta\Sigma$ ADC运行期间, 请勿将TEMPEN<EN0>设置为"0" (停止)。

16. 实时时钟(RTC)

16.1 功能

1. 时钟(时, 分与秒)
2. 日历(月, 周, 日与闰年)
3. 可选12 (am/ pm) 且24小时显示
4. 时间调整+或-30秒(通过软件)
5. 报警器(报警输出)
6. 报警中断
7. 时钟校正功能
8. 1 Hz时钟输出

16.2 方块图

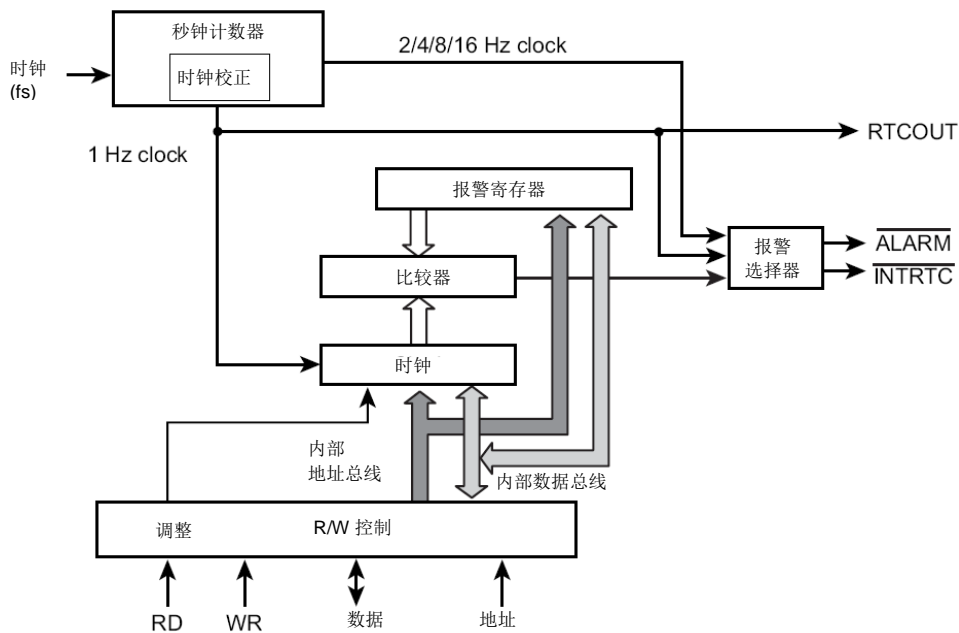


图16-1 方块图

注 1: 西历年列: 本产品仅使用年份的最后两个数字。年份99之后的年份为00年。在西历中进行年份处理时, 请将开头两个数字考虑在内。

注 2: 闰年: 指可 4 除尽的年份, 但不包括可 100 除尽的年份; 可 100 除尽的年份不视为闰年。可 400 除尽的任何年份均为闰年。本产品将可 4 除尽的年份视为闰年, 且不将以上异常考虑在内。在出现类异常时, 需对其进行调整。

16.3 寄存器详细描述

16.3.1 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“存储器地址”一章的“外部功能地址表”。实时时钟有两个功能，PAGE0 (时钟)与PAGE1 (报警)，它们共享寄存器的某些部分。

PAGE可通过设置RTCPAGER<PAGE>来选取。

寄存器名称		地址 (基+)
秒钟列寄存器(仅PAGE0)	RTCSECR	0x0000
分钟列寄存器	RTCMINR	0x0001
小时列寄存器	RTCHOURR	0x0002
-(注 1)	-	0x0003
周日列寄存器	RTCDAYR	0x0004
日列寄存器	RTCDATER	0x0005
月份列寄存器(PAGE0) 24-小时, 12-小时选择寄存器(PAGE1)	RTCMONTHR	0x0006
年份列寄存器(PAGE0) 闰年寄存器(PAGE 1)	RTCYEARR	0x0007
页面寄存器	RTCPAGER	0x0008
-(注 1)	-	0x0009
-(注 1)	-	0x000A
-(注 1)	-	0x000B
PAGE寄存器	RTCRESTR	0x000C
-(注 1)	-	0x000D
保护寄存器	RTCPROTECT	0x000E
校正功能控制寄存器	RTCADJCTL	0x000F
校正值寄存器	RTCADJDAT	0x0010, 0x0011

注：通过读取地址，可读取“0”。写入忽略不计。

16.3.2 控制寄存器

复位工作使以下寄存器初始化。

- RTCPAGER<PAGE>, <ADJUST>, <INTENA>
- RTCRESTR
- RTCPROTECT
- RTCADJCTL
- RTCADJDAT

其它时钟相关的寄存器不由复位工作来初始化。

使用实时时钟之前，在有关寄存器中设置时间，月份，日份，周日，年份与闰年。

设置时钟数据，调整秒钟或复位时钟时需要格外小心。

有关详细，请参看“16.4.3 进入低功耗模式”。

表 16-1 PAGE0(时钟功能)寄存器

符号	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	功能
RTCSECR		-	40秒	20秒	10秒	8秒	4秒	2秒	1秒	秒钟列
RTCMINR		-	40分	20分	10分	8分	4分	2分	1分	分钟列
RTCHOURR		-	-	20小时 PM/AM	10小时	8小时	4小时	2小时	1小时	小时列
RTCDAYR		-	-	-	-	-	周日			周日列
RTCDATER		-	-	20日	10日	8日	4日	2日	1日	日份列
RTCMONTHR		-	-	-	十月	八月	四月	二月	一月	月份列
RTCYEARR		年份80	年份40	年份20	年份10	年份8	年份4	年份2	年份1	年份列 (以下两列)
RTCPAGER		中断 启用	-	-	调整 功能	时钟 启用	报警 启用	-	PAGE 设置	PAGE 寄存器
RTCRESTR		1 Hz 启用	16 Hz 启用	时钟 复位	报警 复位	-	2 Hz 启用	4 Hz 启用	8 Hz 启用	复位 寄存器
RTCPROTECT		保护代码								时钟校正功能 寄存器保护
RTCADJCTL		-	-	-	-	校正参考时间			校正 启用	校正功能 控制
RTCADJDAT		校正值								校正值

注：通过读取PAGE0 的RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR捕获当前状态。

表16-2 PAGE1 (报警功能)寄存器

符号	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	功能
RTCSECR		-	-	-	-	-	-	-	-	-
RTCMINR		-	40分	20分	10分	8分	4分	2分	1分	分钟列
RTCHOURR		-	-	20小时 PM/AM	10小时	8小时	4小时	2小时	1小时	小时列
RTCDAYR		-	-	-	-	-	周日			周日列
RTCDATER		-	-	20日	10日	8日	4日	2日	1日	日份列
RTCMONTHR		-	-	-	-	-	-	-	24/12	24-小时时钟模式
RTCYEARR		-	-	-	-	-	-	闰年设置		闰年模式
RTCPAGER		中断 启用	-	-	调整 功能	时钟 启用	报警 启用	-	PAGE 设置	PAGE 寄存器
RTCRESTR		1 Hz 启用	16 Hz 启用	时钟 复位	报警 复位	-	2 Hz 启用	4 Hz 启用	8 Hz 启用	复位寄存器
RTCPROTECT		保护代码								时钟校正功能 寄存器保护
RTCADJCTL		-	-	-	-	校正参考时间			校正启 用	校正功能 控制
RTCADJDAT		校正值								校正值

注 1：读取PAGE1 的RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR可获取当前状态。

注 2：PAGE0的RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCDATER, RTCMONTHR, RTCYEARR与PAGE1 的RTCYEARR(闰年)必须读取两次，并比较获取的数据。

16.3.3 控制寄存器的详细描述

16.3.3.1 RTCSECR (秒钟列寄存器(仅适用于 PAGE0))

	7	6	5	4	3	2	1	0
比特符号	-	SE						
复位后	0	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
7	-	R	读作 0。
6-0	SE	R/W	设置秒钟的位数寄存器 000_0000: 00秒 001_0000: 10秒 010_0000: 20秒 000_0001: 01秒 001_0001: 11秒 · 000_0010: 02秒 001_0010: 12秒 011_0000: 30秒 000_0011: 03秒 001_0011: 13秒 · 000_0100: 04秒 001_0100: 14秒 100_0000: 40秒 000_0101: 05秒 001_0101: 15秒 · 000_0110: 06秒 001_0110: 16秒 101_0000: 50秒 000_0111: 07秒 001_0111: 17秒 · 000_1000: 08秒 001_1000: 18秒 · 000_1001: 09秒 001_1001: 19秒 101_1001: 59秒

注：上列以外的设置禁止。

16.3.3.2 RTCMINR (分钟列寄存器(PAGE0/1))

	7	6	5	4	3	2	1	0
比特符号	-	MI						
复位后	0	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
7	-	R	读作 0。
6-0	MI	R/W	设置分钟的位数寄存器。 000_0000: 00分 001_0000: 10分 010_0000: 20分 000_0001: 01分 001_0001: 11分 · 000_0010: 02分 001_0010: 12分 011_0000: 30分 000_0011: 03分 001_0011: 13分 · 000_0100: 04分 001_0100: 14分 100_0000: 40分 000_0101: 05分 001_0101: 15分 · 000_0110: 06分 001_0110: 16分 101_0000: 50分 000_0111: 07分 001_0111: 17分 · 000_1000: 08分 001_1000: 18分 · 000_1001: 09分 001_1001: 19分 101_1001: 59分 111_1111: 不要在报警功能状态比较分钟。

注：上列以外的设置禁止。

16.3.3.3 RTCHOURR (小时列寄存器(PAGE0/1))

(1) 24-小时时钟模式(RTCMONTHR<MO0>= "1")

	7	6	5	4	3	2	1	0
比特符号	-	-	HO					
复位后	0	0	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
7-6	-	R	读作 0。
5-0	HO	R/W	设置小时的位数寄存器 00_0000: 0 点钟 01_0000: 10 点钟 10_0000: 20 点钟 00_0001: 1 点钟 01_0001: 11 点钟 10_0001: 21 点钟 00_0010: 2 点钟 01_0010: 12 点钟 10_0010: 22 点钟 00_0011: 3 点钟 01_0011: 13 点钟 10_0011: 23 点钟 00_0100: 4 点钟 01_0100: 14 点钟 00_0101: 5 点钟 01_0101: 15 点钟 00_0110: 6 点钟 01_0110: 16 点钟 00_0111: 7 点钟 01_0111: 17 点钟 00_1000: 8 点钟 01_1000: 18 点钟 00_1001: 9 点钟 01_1001: 19 点钟 11_1111 : 不要在报警功能状态比较“小时”。

注：上列以外的设置禁止。

(2) 12-小时时钟模式 (RTCMONTHR<MO0> = "0")

	7	6	5	4	3	2	1	0
比特符号	-	-	HO					
复位后	0	0	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
7-6	-	R	读作"0"。
5-0	HO	R/W	设置小时的位数寄存器 (上午) (下午) 00_0000: 0 点钟 10_0000: 0 点钟 00_0001: 1 点钟 10_0001: 1 点钟 00_0010: 2 点钟 10_0010: 2 点钟 00_0011: 3 点钟 10_0011: 3 点钟 00_0100: 4 点钟 10_0100: 4 点钟 00_0101: 5 点钟 10_0101: 5 点钟 00_0110: 6 点钟 10_0110: 6 点钟 00_0111: 7 点钟 10_0111: 7 点钟 00_1000: 8 点钟 10_1000: 8 点钟 00_1001: 9 点钟 10_1001: 9 点钟 01_0000: 10 点钟 11_0000: 10 点钟 01_0001: 11 点钟 11_0001: 11 点钟 11_1111: 不要在报警功能状态比较小时。

注：上列以外的设置禁止。

16.3.3.4 RTCDAYR(周日列寄存器(PAGE0/1))

	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	WE		
复位后	0	0	0	0	0	未定义	未定义	未定义

位	比特符号	类型	功能
7-3	-	R	读作 0。
2-0	WE	R/W	设置周日的位数寄存器。 000: 星期日 001: 星期一 010: 星期二 011: 星期三 100: 星期四 101: 星期五 110: 星期六 111: 请勿在报警功能状态比较周的日。

注: 上列以外的设置禁止。

16.3.3.5 RTCDATER (日列寄存器(仅适用于 PAGE0/1))

	7	6	5	4	3	2	1	0
比特符号	-	-	DA					
复位后	0	0	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
7-6	-	R	读作 0。
5-0	DA	R/W	设置日的位数寄存器。 00_0000: 第10日 01_0000: 第20日 10_0000: 第30日 00_0001: 第1日 01_0001: 第11日 10_0001: 第21日 11_0001: 第31日 00_0010: 第2日 01_0010: 第12日 10_0010: 第22日 00_0011: 第3日 01_0011: 第13日 10_0011: 第23日 00_0100: 第4日 01_0100: 第14日 10_0100: 第24日 00_0101: 第5日 01_0101: 第15日 10_0101: 第25日 00_0110: 第6日 01_0110: 第16日 10_0110: 第26日 00_0111: 第7日 01_0111: 第17日 10_0111: 第27日 00_1000: 第8日 01_1000: 第18日 10_1000: 第28日 00_1001: 第9日 01_1001: 第19日 10_1001: 第29日 11_1111: 请勿在报警功能状态比较日

注 1: 上列以外的设置禁止。

注 2: 请勿设置不存在的日期(例如: 2 月 30 日)。

16.3.3.6 RTCMONTHR (月列寄存器(仅适用于 PAGE0))

	7	6	5	4	3	2	1	0
比特符号	-	-	-	MO				
复位后	0	0	0	未定义	未定义	未定义	未定义	未定义

位	比特符号	类型	功能
7-5	-	R	读作 0。
4-0	MO	R/W	设置月份位数寄存器。 0_0001: 一月 0_0111 : 七月 0_0010 : 二月 0_1000 : 八月 0_0011 : 三月 0_1001 : 九月 0_0100 : 四月 1_0000 : 十月 0_0101 : 五月 1_0001 : 十一月 0_0110 : 六月 1_0010 : 十二月

注：上列以外的设置禁止。

16.3.3.7 RTCMONTHR(24-小时时钟或 12-小时时钟的选择(仅适用于 PAGE1))

	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	MO0
复位后	0	0	0	0	0	0	0	未定义

位	比特符号	类型	功能
7-1	-	R	读作 0。
0	MO0	R/W	0: 12-小时 1: 24-小时

注：RTC在运行时，请勿更改RTCMONTHR<MO0>。

16.3.3.8 RTCYEARR(年份列寄存器(仅适用于 PAGE0))

	7	6	5	4	3	2	1	0
比特符号	YE							
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
7-0	YE	R/W	设置年份位数寄存器。 0000_0000 : 00 年 0001_0000 : 10 年 0110_0000 : 60 年 0000_0001 : 01 年 · · 0000_0010 : 02 年 0010_0000 : 20 年 0111_0000 : 70 年 0000_0011 : 03 年 · · 0000_0100 : 04 年 0011_0000 : 30 年 1000_0000 : 80 年 0000_0101 : 05 年 · · 0000_0110 : 06 年 0100_0000 : 40 年 1001_0000 : 90 年 0000_0111 : 07 年 · · 0000_1000 : 08 年 01001_0000 : 50 年 · 0000_1001 : 09 年 · 1001_1001 : 99 年

注：上列以外的设置禁止。

16.3.3.9 RTCYEARR(闰年寄存器(仅适用于 PAGE1))

	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	LEAP	
复位后	0	0	0	0	0	0	未定义	未定义

位	比特符号	类型	功能
7-2	-	R	读作 0。
1-0	LEAP	R/W	00: 闰年 01: 闰年后的一年 10: 闰年后的两年 11: 闰年后的三年

16.3.3.10 RTCPAGER(PAGE 寄存器(PAGE0/1))

	7	6	5	4	3	2	1	0
比特符号	INTENA	-	-	ADJUST	ENATMR	ENAALM	-	PAGE
复位后	0	0	0	0	未定义	未定义	0	0

位	比特符号	类型	功能
7	INTENA	R/W	INTRTC 0: 禁用 1: 启用
6-5	-	R	读作 0。
4	ADJUST	R/W	[写入]: 0: 忽略 1: 设置ADJUST 请求 调整秒钟。秒数计数器递增计数时该请求被取样。 如果经过的时间在 0 与 29 秒钟之间, 秒数计数器被清除至"0"。 如果经过的时间在 30 与 59 秒之间, 分钟数计数器进位, 秒数计数器被清除至"0"。 [读取]: 0: ADJUST未请求 1: ADJUST已请求 如果读取"1", 表明ADJUST正在执行。如果读取"0", 表明执行已完成。
3	ENATMR	R/W	时钟 0: 禁用 1: 启用
2	ENAALM	R/W	ALARM 0: 禁用 1: 启用
1	-	R	读作 0。
0	PAGE	R/W	PAGE选择 0: 选择页面 0 1: 选择页面 1

注 1: 无法执行读取-修改-写入工作。

注 2: 在将各中断启用位设置到<ENATMR>, <ENAALM>和<INTENA>时, 用户必须遵循本文所指定的顺序。不得同时对其进行设置(确认在中断启用和时钟/报警启用之间存在延时)。在改变<ENATMR>和<ENAALM>的设置之前, 必须禁用<INTENA>。

示例: 时钟设置/报警设置

		7	6	5	4	3	2	1	0	
RTCPAGER	←	0	0	0	0	1	1	0	0	启用时钟与报警
RTCPAGER	←	1	0	0	0	1	1	0	0	启用中断

16.3.3.11 RTCRESTR(复位寄存器(适用于 PAGE0/1))

	7	6	5	4	3	2	1	0
比特符号	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	-	DIS2HZ	DIS4HZ	DIS8HZ
复位后	1	1	0	0	0	1	1	1

位	比特符号	类型	功能
7	DIS1HZ	R/W	1 Hz 0: 启用 1: 禁用
6	DIS16HZ	R/W	16 Hz 0: 启用 1: 禁用
5	RSTTMR	R/W	[写入] 0: 忽略 1: 秒钟计数器复位 复位秒数计数器。该请求系用低速时钟采样。 [读取] 0: 无复位请求 1: RESET已请求 如果读取"1", 则表明RESET正在执行。如果读取"0", 表明执行已完成。
4	RSTALM	R/W	0: 忽略 1: 报警复位 报警寄存器(分钟列, 小时列, 日列, 周日列)初始化如下。 分钟: 00, 小时: 00, 日: 01, 周日: 星期天
3	-	R	读作 0。
2	DIS2HZ	R/W	2 Hz 0: 启用 1: 禁用
1	DIS4HZ	R/W	4 Hz 0: 启用 1: 禁用
0	DIS8HZ	R/W	8 Hz 0: 启用 1: 禁用

注: 无法执行读取-修改-写入工作。

以下给出了<DIS1HZ>, <DIW2HZ>, <DIS4HZ>和<DIS16MHZ>, 报警用 RTCPAGER<ENAALM>, 1 Hz, 2 Hz, 4 Hz, 8 Hz 和 16 Hz 中断的设置。

<DIS1HZ>	<DIS2HZ>	<DIS4HZ>	<DIS8HZ>	<DIS16HZ>	RTCPAGER <ENAALM>	中断源 信号
1	1	1	1	1	1	ALARM
0	1	1	1	1	0	1 Hz
1	0	1	1	1	0	2 Hz
1	1	0	1	1	0	4 Hz
1	1	1	0	1	0	8 Hz
1	1	1	1	0	0	16 Hz
其他						中断 未生成。

16.3.3.12 RTCPROTECT(保护寄存器)

	7	6	5	4	3	2	1	0
比特符号	RTCPROTECT							
复位后	1	1	0	0	0	0	0	1

位	比特符号	类型	機能
7-0	RTCPROTECT	R/W	时钟校正功能寄存器保护 0xC1: 写入启用。 0xC1: 写入禁用。 在初始状态时, RTCPROTECT为"0xC1"和写入启用。如果RTCPROTECT被设置为"0xC1"以外的某个值, 则RTCADJCTL和RTCADJDAT会被写入禁用。

16.3.3.13 RTCADJCTL(校正功能控制寄存器)

	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	AJSEL			AJEN
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
7-4	-	R	读作 "0"。
3-1	AJSEL	R/W	校准参考时间设置 000: 1 秒钟 001: 10 秒钟 010: 20 秒钟 011: 30 秒钟 100: 1 分钟 101 - 111: 保留 设置校正参考时间。
0	AJEN	R/W	校正功能控制 0: 禁用 1: 启用

16.3.3.14 RTCADJDAT(校正寄存器)

	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	ADJDAT
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADJDAT							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
15-9	-	R	读作 "0"。
8-0	ADJDAT	R/W	校正值 0_0000_0000 : 无校正 0_0000_0001 : 32768 + 1 0_0000_0010 : 32768 + 2 . 0_1111_1110 : 32768 + 254 0_1111_1111 : 32768 + 255 1_0000_0000 : 32768 - 256 1_0000_0001 : 32768 - 255 . 1_1111_1110 : 32768 - 2 1_1111_1111 : 32768 - 1 每秒设置一个校正值。第 8 位为符号位。如果第 8 位为"0", 则应实施正校正。如果该位为"1", 则应实施负校正。 用位 7 将某个校正值指定为 0。

16.4 工作说明

RTC结合了一个从 32.768 kHz信号生成 1 Hz信号的秒数计数器。
在使用RTC时，必须考虑秒数计数器的运行。

16.4.1 读取时钟数据

1. 采用 1 Hz 中断

1 Hz 中断的生成与秒数计数器的递增计数同步。如果读取数据在 1 Hz 中断之后发生，则数据可正确读取。

2. 采用成对读取

如果内部计数器在读取期间运行进位，则时钟数据的读取方式可能会错误。为保证正确的数据读取，应按以下所述两次读取时钟数据。连续读取的数据对需匹配。

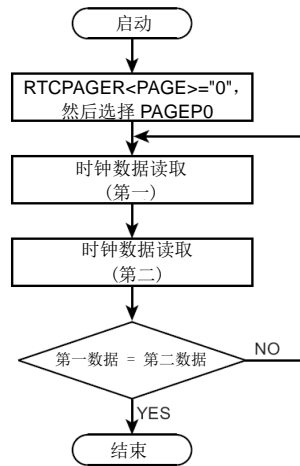


图 16-2 时钟数据读取流程图

16.4.2 写入时钟数据

若写入期间发生进位，则会破坏正确的数据写入。以下步骤可确保数据写入正确。

1. 使用 1 Hz 中断

通过与秒钟计数器的计数同步，可生成 1 Hz 中断。如果数据在 1 Hz 中断和后续的一秒钟计数之间被写入，则其完成正确。

2. 复位计数器

在复位秒数计数器后写入数据。

1 Hz 中断在紧接计数器复位后启用中断的一秒钟之后生成。

该时间必须在中断之后一秒钟之内设置。

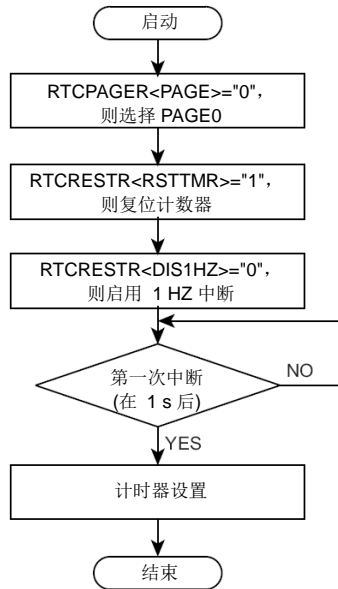


图 16-3 时钟数据写入流程图

3. 禁用时钟

写入"0"至 RTCPAGER<ENATMR>禁用时钟运行，包括进位。

在 1 Hz 中断之后，停止该时钟。秒钟计数器保持计数。

再次设置时钟并在下一个 1 Hz 中断之前的一秒钟之内启用时钟

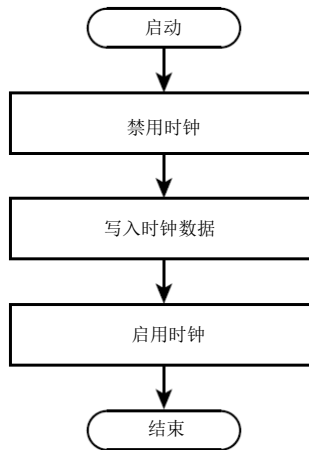


图16-4 时钟禁用流程图

16.4.3 进入低功耗模式

在进入 SLEEP 模式(在改变时钟数据, 调整秒钟或重新设置时钟之后, 其中的系统时钟会停止工作)时, 务必遵循以下步骤之一

1. 在更改时钟设置寄存器, 设置 RTCPAGER<ADJUST>位或设置 RTCRESTR<RSTTMR>位之后, 等待一秒钟以生成一个中断。
2. 在更改时钟设置寄存器, 设置 RTCPAGER<ADJUST>位或设置 RTCRESTR<RSTTMR>位之后, 读取相应时钟寄存器的值, <ADJUST>或<RSTTMR>, 确保所做的设置已反映出来。

16.5 报警功能

向RTCPAGER<PAGE>写入“1”，PAGE1寄存器的报警功能即被启用。如果该产品具备 $\overline{\text{ALARM}}$ 引脚，则以下信号之一会被输出到该引脚。

1. “低”脉冲(报警寄存器与时钟一致时)
2. 1, 2, 4, 8 或 16 Hz周期“低”脉冲

在上述的任何情况下，RTC都会输出低速时钟的一周期脉冲。其会同时输出该INTRTC中断请求。INTRTC中断信号由下降沿触发。可在CG中断模式控制寄存器中将该下降沿指定为活动状态。

16.5.1 报警功能的使用

在PAGE0时钟寄存器和PAGE1报警寄存器的值对应时，“低”脉冲就会被输出到 $\overline{\text{ALARM}}$ 引脚。INTRTC中断即被生成，报警即被触发。

报警设置

在已禁止报警的情况下，初始化报警。向RTCRESTR<RSTALM>写入“1”。

这就使报警设置为00分钟，00小时，01日与星期日。

写入数据至相关的PAGE1寄存器即完成报警的分钟，小时，日期与日的设置。

用RTCPAGER<ENAALM>位启用该报警。用RTCPAGER<INTENA>位启用该中断。

以下程序示例为“在五日(星期一)中午(12:00)从 $\overline{\text{ALARM}}$ 引脚输出一报警”。

	7	6	5	4	3	2	1	0	
RTCPAGER	← 0	0	0	0	1	0	0	1	禁用报警，设置PAGE1
RTCRESTR	← 1	1	0	1	0	0	0	0	报警初始化
RTCDAYR	← 0	0	0	0	0	0	0	1	星期一
RTCDATER	← 0	0	0	0	0	1	0	1	第5日
RTCHOURR	← 0	0	0	1	0	0	1	0	设置12点钟
RTCMINR	← 0	0	0	0	0	0	0	0	设置00分钟
RTCPAGER	← 0	0	0	0	1	1	0	0	启用报警
RTCPAGER	← 1	0	0	0	1	1	0	0	启用中断

如果某些报警寄存器被设置为“1”，则RTC不会比较该项。例如，如果RTCDATER被设置为“11_1111”，且RTCDAYR被设置为“111”，则每天中午(12:00)都会输出该报警。

以上报警的工作与低速时钟同步。当CPU按高频振荡运行时，会根据拟变为有效的寄存器设置，发生最长一次时钟的延迟 t_s 时(约30 μ s)。

16.5.2 1, 2, 4, 8 或 16 Hz 周期“低”脉冲

通过在设置 RTCPAGER<ENAALM>= "0"和 RTCRESTR 之后设置 RTCPAGER<INTENA>="1", RTC 就会向 $\overline{\text{ALARM}}$ 引脚输出一个“低”脉冲周期。必须将 RTCRESTR<DIS1HZ>, <DIS2HZ>, <DIS4HZ>, <DIS8HZ>或<DIS16HZ>的其中之一设置为"0"。

RTC 输出与 RTCRESTR 设置的低速时钟相符的一周期脉冲。其会同时生成一个 INTRTC 中断。

16.6 时钟校正功能

时钟校正功能可精确调整时钟的的偏置。

在图 16-5 中，T1 表示一秒钟。通过对fs (32768Hz)计数 32768 次，即可生成一秒钟。时钟校正功能可调整T2其为参考时间(Tall)的一秒钟计数的数目。可用RTCADJCTL<AJSEL>在1, 10, 20, 30 秒钟或 1 分钟中选择校正参考时间。可用RTCADJ DAT<ADJDAT>在 32768-255 ~ 32768+256 的范围内调整T2 的计数值。

符号	项目	说明
Tall	校正参考时间	可用RTCADJCTL<AJSEL>在1, 10, 20, 30 秒钟或 1 分钟中任选其一。
T1	1 秒	对fs计数 32768 次
T2	计数校正	用RTCADJDAT<ADJDAT>通过加/减 32768 计数，来调整计数值。

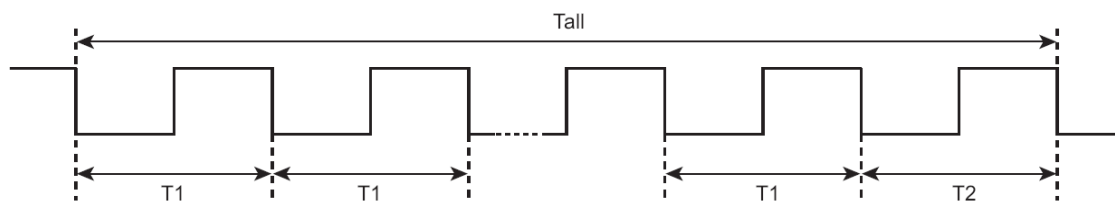


图 16-5 时钟校正

可用RTCPROTECT寄存器禁用与校正功能相关的寄存器RTCADJCTL和RTCADJDAT。在初始状态下，RTCPROTECT设置为“0xC1”并写入启用。如果RTCPROTECT被设置为“0xC1”以外的某个值，则RTCADJCTL和RTCADJDAT会被写入禁用。

16.7 1Hz 时钟输出功能

RTCOUT引脚可输出 1Hz时钟。时钟经调整后可按 50% 占空比率运行。如使用时钟校正功能，则占空比率可因错误校正而改变。

17. LCD 驱动器

TMPM061FWFG带有可直接驱动液晶显示(LCD)设备的驱动器和控制电路。
与液晶显示连接的引脚如下:

1. 分段输出引脚 : 40 引脚 (SEG39 ~ SEG0)
2. 共用输出引脚 : 4 引脚 (COM3 ~ COM0)

此外, VLC引脚为驱动电源引脚, LV1 和LV2 引脚则为外部泄放电阻连接引脚。

注: 当选择静态, 1/3 或 1/2 占空比时, 未使用的共用输出引脚应开路(其可输出偏置电压)。

LCD驱动器可直接驱动以下五类液晶显示:

1. 1/4 占空比(1/3 偏置) LCD 最大 160 个像素(8 个分段 × 20 个数字)
2. 1/3 占空比(1/3 偏置) LCD 最大 120 个像素(8 个分段 × 15 个数字)
3. 1/3 占空比(1/2 偏置) LCD 最大 120 个像素(8 个分段 × 15 个数字)
4. 1/2 占空比(1/2 偏置) LCD 最大 80 个像素(8 个分段 × 10 个数字)
5. 静态LCD 最大 40 个像素(8 个分段 × 5 个数字)

17.1 配置

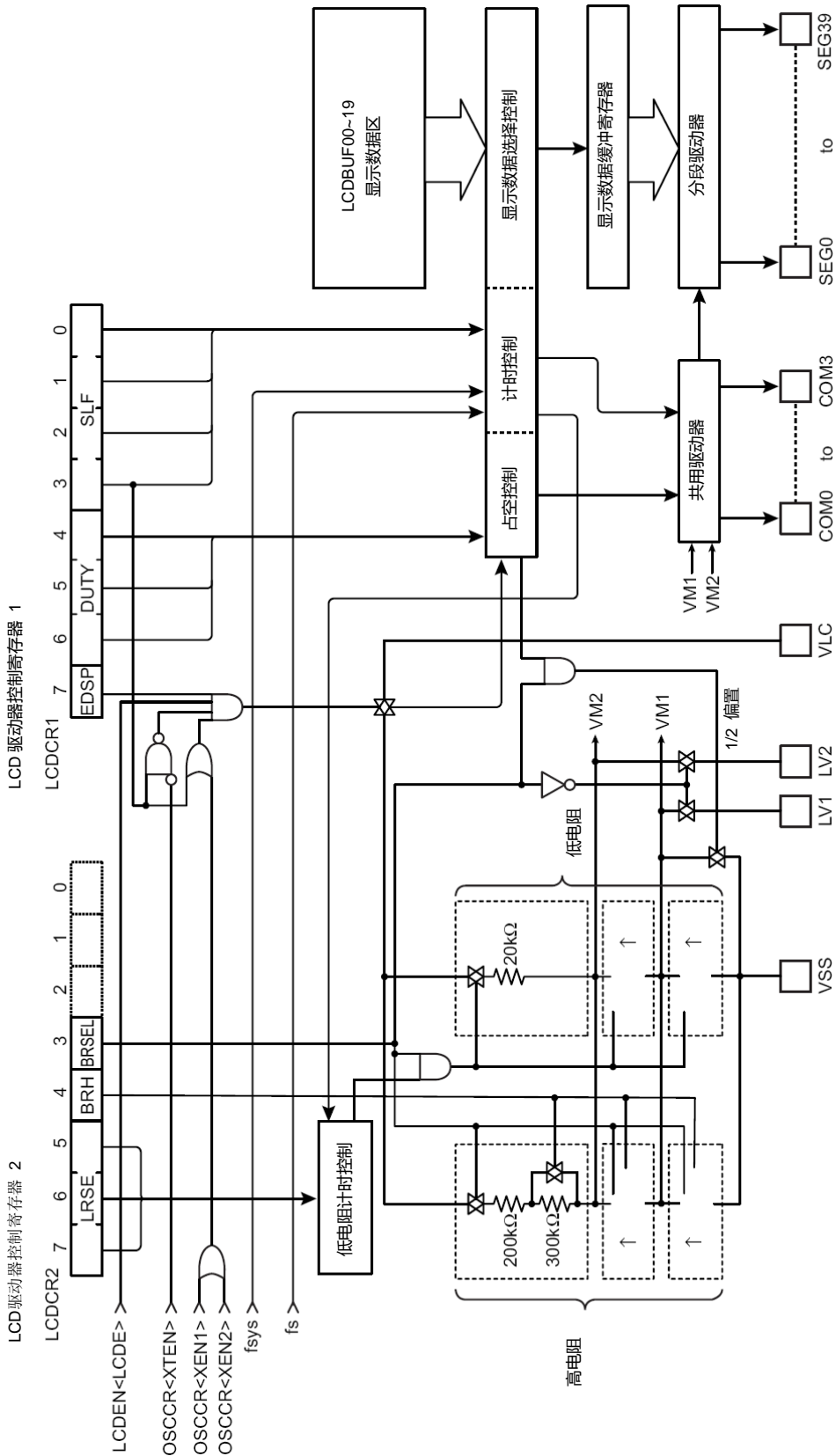


图 17-1 LCD驱动器

17.2 寄存器

17.2.1 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“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址 (基+)
启动寄存器	LCDEN	0x0000
控制寄存器 1	LCDCR1	0x0004
控制寄存器 2	LCDCR2	0x0008
缓冲寄存器 00	LCDBUF00	0x000C
缓冲寄存器 01	LCDBUF01	0x0010
缓冲寄存器 02	LCDBUF02	0x0014
缓冲寄存器 03	LCDBUF03	0x0018
缓冲寄存器 04	LCDBUF04	0x001C
缓冲寄存器 05	LCDBUF05	0x0020
缓冲寄存器 06	LCDBUF06	0x0024
缓冲寄存器 07	LCDBUF07	0x0028
缓冲寄存器 08	LCDBUF08	0x002C
缓冲寄存器 09	LCDBUF09	0x0030
缓冲寄存器 10	LCDBUF10	0x0034
缓冲寄存器 11	LCDBUF11	0x0038
缓冲寄存器 12	LCDBUF12	0x003C
缓冲寄存器 13	LCDBUF13	0x0040
缓冲寄存器 14	LCDBUF14	0x0044
缓冲寄存器 15	LCDBUF15	0x0048
缓冲寄存器 16	LCDBUF16	0x004C
缓冲寄存器 17	LCDBUF17	0x0050
缓冲寄存器 18	LCDBUF18	0x0054
缓冲寄存器 19	LCDBUF19	0x0058

17.2.2 寄存器详细

17.2.2.1 LCDEN (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	-	-	-	-	-	-	-	LCDE
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	LCDE	R/W	<p>LCD运行</p> <p>0: 禁用</p> <p>1: 启用</p> <p>所指定的LCD工作。设置<SIOE> = "1", 即可使用LCD。</p> <p>当工作被禁用时, 不会向LCD模块中的其它寄存器提供任何时钟, 从而降低功耗。</p> <p>如果执行LCD工作之后禁用, 则各寄存器中均会保有设置。</p>

17.2.2.2 LCDCR1 (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	EDSP	DUTY			SLF			
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	EDSP	R/W	LCD显示控制 0: LCD显示空白 1: 启用LCD显示
6-4	DUTY	R/W	LCD驱动方法选择 000: 1/4 占空比(1/3 偏置) 001: 1/3 占空比(1/3 偏置) 010: 1/3 占空比(1/2 偏置) 011: 1/2 占空比(1/2 偏置) 100: 静态 101: 保留 110: 保留 111: 保留
3-0	SLF	R/W	基频选择 0000: $f_{sys}/2^{18}$ 0001: $f_{sys}/2^{17}$ 0010: $f_{sys}/2^{16}$ 0011: $f_{sys}/2^{15}$ 0100: $f_{sys}/2^{14}$ 0101: 保留 0110: 保留 0111: 保留 1000: $f_s/2^9$ 1001: $f_s/2^8$ 1010~1111: 保留

注：在SLOW模式下，请勿将SLF设置为“0000”至“0100”(即：基于 f_{sys} 的频率)。如果SLF被设置为其中一种频率，则会从共用和分段输出引脚输出意外帧频的脉冲。

17.2.2.3 LCDCR2 (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	LRSE			BRH	BRSEL	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能																																																																																
31-8	-	R	读作 "0"。																																																																																
7-5	LRSE	R/W	<p>低内部泄放电阻连接时间选择</p> <p>应按以下所述，通过LCDCR1<SLF>的设置来选择时间。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="8">LCDCR1<SLF></th> </tr> <tr> <th></th> <th>0000</th> <th>0001</th> <th>0010</th> <th>0011</th> <th>0100</th> <th>1000</th> <th>1001</th> </tr> </thead> <tbody> <tr> <td>000:</td> <td colspan="7">未连接</td> </tr> <tr> <td>001:</td> <td>$2^{11}/fsys$</td> <td>$2^{10}/fsys$</td> <td>$2^9/fsys$</td> <td>$2^8/fsys$</td> <td>$2^7/fsys$</td> <td>$2^2/fs$</td> <td>$2/fs$</td> </tr> <tr> <td>010:</td> <td>$2^{12}/fsys$</td> <td>$2^{11}/fsys$</td> <td>$2^{10}/fsys$</td> <td>$2^9/fsys$</td> <td>$2^8/fsys$</td> <td>$2^3/fs$</td> <td>$2^2/fs$</td> </tr> <tr> <td>011:</td> <td>$2^{13}/fsys$</td> <td>$2^{12}/fsys$</td> <td>$2^{11}/fsys$</td> <td>$2^{10}/fsys$</td> <td>$2^9/fsys$</td> <td>$2^4/fs$</td> <td>$2^2/fs$</td> </tr> <tr> <td>100:</td> <td>$2^{14}/fsys$</td> <td>$2^{13}/fsys$</td> <td>$2^{12}/fsys$</td> <td>$2^{11}/fsys$</td> <td>$2^{10}/fsys$</td> <td>$2^5/fs$</td> <td>$2^4/fs$</td> </tr> <tr> <td>101:</td> <td>$2^{15}/fsys$</td> <td>$2^{14}/fsys$</td> <td>$2^{13}/fsys$</td> <td>$2^{12}/fsys$</td> <td>$2^{11}/fsys$</td> <td>$2^6/fs$</td> <td>$2^5/fs$</td> </tr> <tr> <td>110:</td> <td colspan="7">始终连接</td> </tr> <tr> <td>111:</td> <td colspan="7">保留</td> </tr> </tbody> </table>	LCDCR1<SLF>									0000	0001	0010	0011	0100	1000	1001	000:	未连接							001:	$2^{11}/fsys$	$2^{10}/fsys$	$2^9/fsys$	$2^8/fsys$	$2^7/fsys$	$2^2/fs$	$2/fs$	010:	$2^{12}/fsys$	$2^{11}/fsys$	$2^{10}/fsys$	$2^9/fsys$	$2^8/fsys$	$2^3/fs$	$2^2/fs$	011:	$2^{13}/fsys$	$2^{12}/fsys$	$2^{11}/fsys$	$2^{10}/fsys$	$2^9/fsys$	$2^4/fs$	$2^2/fs$	100:	$2^{14}/fsys$	$2^{13}/fsys$	$2^{12}/fsys$	$2^{11}/fsys$	$2^{10}/fsys$	$2^5/fs$	$2^4/fs$	101:	$2^{15}/fsys$	$2^{14}/fsys$	$2^{13}/fsys$	$2^{12}/fsys$	$2^{11}/fsys$	$2^6/fs$	$2^5/fs$	110:	始终连接							111:	保留						
LCDCR1<SLF>																																																																																			
	0000	0001	0010	0011	0100	1000	1001																																																																												
000:	未连接																																																																																		
001:	$2^{11}/fsys$	$2^{10}/fsys$	$2^9/fsys$	$2^8/fsys$	$2^7/fsys$	$2^2/fs$	$2/fs$																																																																												
010:	$2^{12}/fsys$	$2^{11}/fsys$	$2^{10}/fsys$	$2^9/fsys$	$2^8/fsys$	$2^3/fs$	$2^2/fs$																																																																												
011:	$2^{13}/fsys$	$2^{12}/fsys$	$2^{11}/fsys$	$2^{10}/fsys$	$2^9/fsys$	$2^4/fs$	$2^2/fs$																																																																												
100:	$2^{14}/fsys$	$2^{13}/fsys$	$2^{12}/fsys$	$2^{11}/fsys$	$2^{10}/fsys$	$2^5/fs$	$2^4/fs$																																																																												
101:	$2^{15}/fsys$	$2^{14}/fsys$	$2^{13}/fsys$	$2^{12}/fsys$	$2^{11}/fsys$	$2^6/fs$	$2^5/fs$																																																																												
110:	始终连接																																																																																		
111:	保留																																																																																		
4	BRH	R/W	<p>高内部泄放电阻选择</p> <p>0: 200 kΩ(典型值)</p> <p>1: 500 kΩ(典型值)</p>																																																																																
3	BRSEL	R/W	<p>内部/外部泄放电阻切换控制</p> <p>0: 使用外部泄放电阻</p> <p>1: 使用内部泄放电阻</p>																																																																																
2-0	-	R	读作"0"。																																																																																

注：LRSE和BRH设置仅当BRSEL经设置以选择内部泄放电阻时有效。

17.2.2.4 LCDBUF00 (缓冲寄存器 00)

	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	SEG1				SEG0			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG1	R/W	SEG1 的数据 指定SEG1 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG0	R/W	SEG0 的数据 指定SEG0 的COM3, COM2, COM1 和COM0 数据。

17.2.2.5 LCDBUF01 (缓冲寄存器 01)

	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	SEG3				SEG2			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG3	R/W	SEG3 的数据 指定SEG3 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG2	R/W	SEG2 的数据 指定SEG2 的COM3, COM2, COM1 和COM0 数据。

17.2.2.6 LCDBUF02 (缓冲寄存器 02)

	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	SEG5				SEG4			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG5	R/W	SEG5 的数据 指定SEG5 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG4	R/W	SEG4 的数据 指定SEG4 的COM3, COM2, COM1 和COM0 数据。

17.2.2.7 LCDBUF03 (缓冲寄存器 03)

	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	SEG7				SEG6			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG7	R/W	SEG7 的数据 指定SEG7 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG6	R/W	SEG6 的数据 指定SEG6 的COM3, COM2, COM1 和COM0 数据。

17.2.2.8 LCDBUF04 (缓冲寄存器 04)

	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	SEG9				SEG8			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG9	R/W	SEG9 的数据 指定SEG9 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG8	R/W	SEG8 的数据 指定SEG8 的COM3, COM2, COM1 和COM0 数据。

17.2.2.9 LCDBUF05 (缓冲寄存器 05)

	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	SEG11				SEG10			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG11	R/W	SEG11 的数据 指定SEG11的COM3, COM2, COM1 和COM0 数据。
3-0	SEG10	R/W	SEG10 的数据 指定SEG10 的COM3, COM2, COM1和COM0 数据。

17.2.2.10 LCDBUF06 (缓冲寄存器 06)

	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	SEG13				SEG12			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG13	R/W	SEG13 的数据 指定SEG13 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG12	R/W	SEG12 的数据 指定SEG12 的COM3, COM2, COM1 和COM0 数据。

17.2.2.11 LCDBUF07 (缓冲寄存器 07)

	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	SEG15				SEG14			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG15	R/W	SEG15 的数据 指定SEG15 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG14	R/W	SEG14 的数据 指定SEG14 的COM3, COM2, COM1 和COM0 数据。

17.2.2.12 LCDBUF08 (缓冲寄存器 08)

	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	SEG17				SEG16			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG17	R/W	SEG17 的数据 指定SEG17 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG16	R/W	SEG16 的数据 指定SEG16 的COM3, COM2, COM1 和COM0 数据。

17.2.2.13 LCDBUF09 (缓冲寄存器 09)

	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	SEG19				SEG18			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG19	R/W	SEG19 的数据 指定SEG19 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG18	R/W	SEG18 的数据 指定SEG18 的COM3, COM2, COM1 和COM0 数据。

17.2.2.14 LCDBUF10 (缓冲寄存器 10)

	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	SEG21				SEG20			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作"0"。
7-4	SEG21	R/W	SEG21 的数据 指定SEG21 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG20	R/W	SEG20 的数据 指定SEG20 的COM3, COM2, COM1 和COM0 数据。

17.2.2.15 LCDBUF11 (缓冲寄存器 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	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SEG23				SEG22			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG23	R/W	SEG23 的数据 指定SEG23 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG22	R/W	SEG22 的数据 指定SEG22 的COM3, COM2, COM1 和COM0 数据。

17.2.2.16 LCDBUF12 (缓冲寄存器 12)

	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	SEG25				SEG24			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG25	R/W	SEG25 的数据 指定SEG25 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG24	R/W	SEG24 的数据 指定SEG24 的COM3, COM2, COM1 和COM0 数据。

17.2.2.17 LCDBUF13 (缓冲寄存器 13)

	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	SEG27				SEG26			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作"0"。
7-4	SEG27	R/W	SEG27 的数据 指定SEG27 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG26	R/W	SEG26 的数据 指定SEG26 的COM3, COM2, COM1 和COM0 数据。

17.2.2.18 LCDBUF14 (缓冲寄存器 14)

	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	SEG29				SEG28			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG29	R/W	SEG29 的数据 指定SEG29 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG28	R/W	SEG28 的数据 指定SEG28 的COM3, COM2, COM1 和COM0 数据。

17.2.2.19 LCDBUF15 (缓冲寄存器 15)

	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	SEG31				SEG30			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG31	R/W	SEG31 的数据 指定SEG31 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG30	R/W	SEG30 的数据 指定SEG30 的COM3, COM2, COM1 和COM0 数据。

17.2.2.20 LCDBUF16 (缓冲寄存器 16)

	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	SEG33				SEG32			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG33	R/W	SEG33 的数据 指定SEG33 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG32	R/W	SEG32 的数据 指定SEG32 的COM3, COM2, COM1 和COM0 数据。

17.2.2.21 LCDBUF17 (缓冲寄存器 17)

	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	SEG35				SEG34			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG35	R/W	SEG35 的数据 指定SEG35 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG34	R/W	SEG34 的数据 指定SEG34 的COM3, COM2, COM1 和COM0 数据。

17.2.2.22 LCDBUF18 (缓冲寄存器 18)

	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	SEG37				SEG36			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG37	R/W	SEG37 的数据 指定SEG37 的COM3, COM2, COM1和COM0数据。
3-0	SEG36	R/W	SEG36 的数据 指定SEG36 的COM3, COM2, COM1 和COM0 数据。

17.2.2.23 LCDBUF19 (缓冲寄存器 19)

	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	SEG39				SEG38			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	機能
31-8	-	R	读作 "0"。
7-4	SEG39	R/W	SEG39 的数据 指定SEG39 的COM3, COM2, COM1 和COM0 数据。
3-0	SEG38	R/W	SEG38 的数据 指定SEG38 的COM3, COM2, COM1 和COM0 数据。

17.3 功能

17.3.1 LCD 显示控制

如果 LCDCR1<EDSP>被设置为"1", 则 LCD 驱动器的电源开关闭合, VLC 电压即被施加到 LCD 驱动器上。LCD 驱动器显示即被启用。如果<EDSP>被设置为"0", 则 LCD 驱动器的电源开关断开, VLC 电压即被切断。这将导致驱动器显示空白。

下表给出了 LCD 连接引脚在<EDSP>为"1"时的状态。

1. 共用输出引脚

仅可用 LCDCR1<EDSP>控制共用输出引脚。

LCDCR1 <EDSP>	共用输出引脚 COM0, COM1, COM2, COM3
0	"低"电平
1	共用输出

2. 外部泄放电阻器引脚

外部泄放电阻器引脚是一种带有通用输入/输出端口的两用引脚。当该引脚被用作外部泄放电阻器引脚时, 向 LCDCR2<BRSEL>设置"0"。无需对端口控制寄存器进行设置。

LCDCR2 <BRSEL>	外部泄放电阻器引脚 LV1, LV2
1	输入/输出端口
0	外部泄放电阻器

3. 分段输出引脚

分段输出引脚是一种带有通用输入/输出端口的两用引脚。在被用于分段输出时, 需向相应的功能寄存器 PxFRn 设置"1"。无需对其它端口控制寄存器进行设置 PxFRn 除外。

PxFRn	PxCR	外部泄放电阻器引脚 LV1, LV2
0	0	通用端口
0	1	
1	0	分段输出
1	1	-

注) 寄存器名称中的"x"指端口数, 而"n"指功能寄存器编号。

17.3.2 复位时的工作

在复位发生时, LCDCR1<EDSP>会被初始化为"0", LCD 驱动器的电源开关会被自动关闭, 从而断开 VLC 电压。此时, 各共用输出引脚会被固定为"低"电平。各多工引脚(输入/输出端口或分段输出)会被配置为端口输入引脚(高阻抗)。因此, 如果外部复位工作耗时较长, 则 LCD 显示可能会变得模糊。



17.3.3 SLEEP/STOP 模式下的工作

如果在 LCDCR1<EDSP>为"1"期间转换为 SLEEP/STOP 模式，则会出现以下状况。

注：在 SLEEP 模式下使用 LCD 时，应在 SLOW 模式下转换为 SLEEP 模式。如果发生从 NORMAL 模式向 SLEEP 模式的转换，则显示会变为空白。

- SLEEP 模式

共用引脚和分段引脚的状态，与其在发生转换和使用显示屏之前的状态相同。

- STOP 模式

如果转换为 STOP 模式，则 LCDCR1<EDSP>会被自动初始化为"0"，显示则变为空白。向 LCDCR1<EDSP>设置"1"，即可从 STOP 模式返回，并使 LCD 重新显示。

17.3.4 SLOW 模式下的工作

当 LCD 同时用于 NORMAL 模式 SLOW 模式时，建议将 LCDCR1<SLF>设置为基于 fs 的某个频率 ("1000"或"1001")(这样，每次在 NORMAL 模式和 SLOW 模式间进行工作模式切换时，就无需改变 LCDCR1<SLF>设置)。

如果基于 fsys 的某个频率被用于 NORMAL 模式，则需在切换为 SLOW 模式之前，将 LCDCR1<EDSP>清除为"0"。在进入 SLOW 模式之后，需将 LCDCR1<SLF>改为基于 fs 的某个频率，并将 LCDCR1<EDSP>设置为"1"。同样地，在从 SLOW 模式切换为 NORMAL 模式时，也需在切换为 NORMAL 模式之前，将 LCDCR1<EDSP>清除为"0"。在进入 NORMAL 模式之后，需将 LCDCR1<SLF>改为基于 fsys 的某个频率，并将 LCDCR1<EDSP>设置为"1"。

17.3.5 故障安全

在 LCD 无法显示时需注意以下事项。

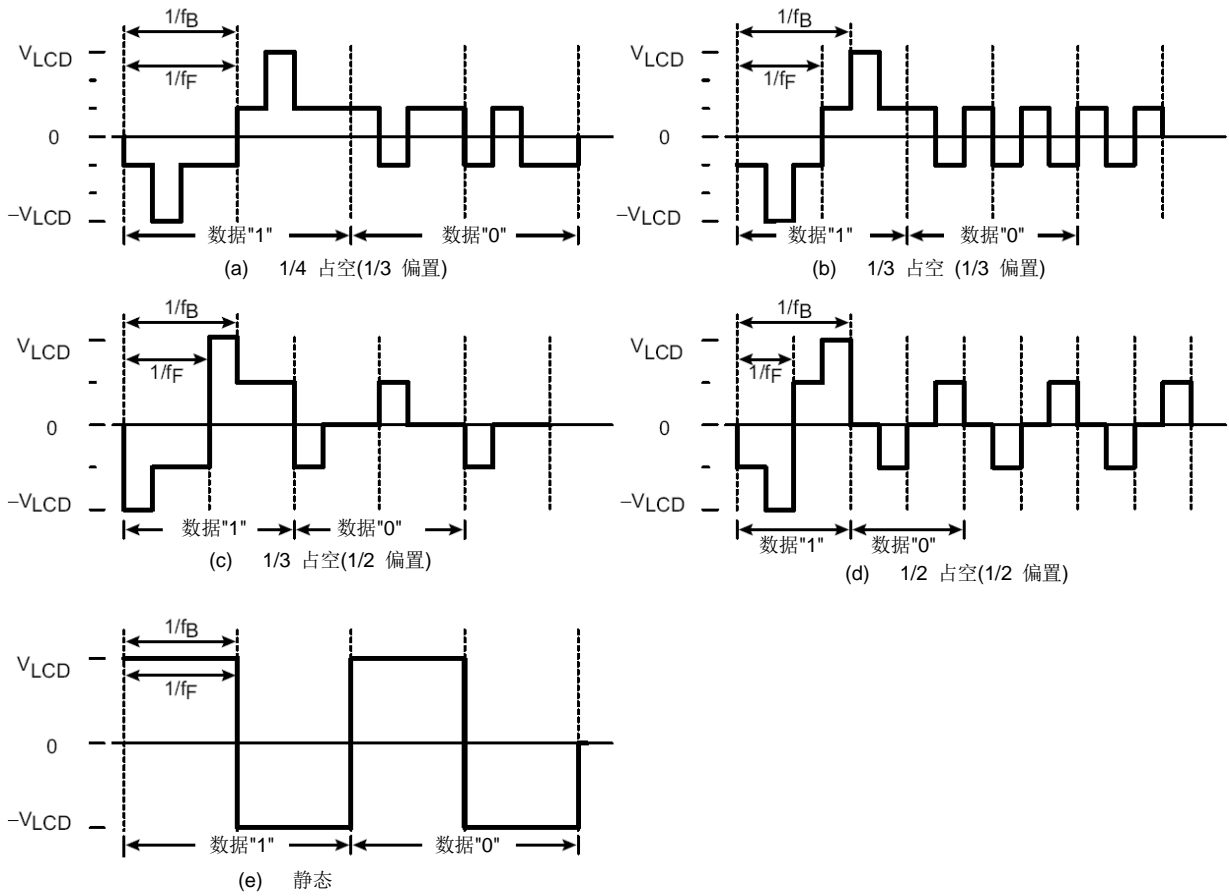
1. 当 LCDCR1<EDSP>="1"时，将 LCDEN<LCDE>设置为"0"会使 LCD 显示空白。通过将 LCDEN<LCDE>设置为"1"，即可重新启用显示。

2. 当 LCDCR1<SLF>被设置为"0000"至"0110"时，必须在 LCDCR1 <EDSP>可被设置为"1"之前激活高速时钟(OSCCR<XEN1> = "1"或 OSCCR<XEN> = "1")，并允许其实现稳定振荡。如果 LCDCR1<EDSP>被设置为"1"，而高速时钟已被停止，则 LCD 显示无法被启用(尽管 LCDCR1<EDSP>变为"1"，LCD 显示仍为空白)。

3. 当 LCDCR1<SLF>被设置为"1000"或"1001"时，必须在 LCDCR1 <EDSP>可被设置为"1"之前激活低速时钟(OSCCR <XEN1> = "1")，并允许其实现稳定振荡。如果 LCDCR1<EDSP>被设置为"1"，而低速时钟已被停止，则 LCD 显示无法被启用(尽管 LCDCR1<EDSP>变为"1"，LCD 显示仍为空白)。

17.3.6 LCD 驱动方法(LCDCR1<DUTY>)

通过设置 LCDCR1<DUTY>, 可从以下五种 LCD 驱动方法中进行选择。



注 1: f_F = 帧频

注 2: V_{LCD3} = LCD驱动电压(= $V_{LC} - V_{SS}$)

图 17-2 LCD驱动波形(COM与SEG引脚之间的电势差)

17.3.7 帧频(LCDCR1<SLF>)

依据表 17-1 所述的驱动方法和基频确定帧频(f_f)。基频通过 LCDCR1<SLF>进行选择。

表 17-1 帧频设置

SLF	基频[Hz]	帧频[Hz]			
		1/4 占空比	1/3 占空比	1/2 占空比	静态
0000	$f_{sys} / 2^{18}$	$f_{sys} / 2^{18}$	$(4/3) \times f_{sys} / 2^{18}$	$(4/2) \times f_{sys} / 2^{18}$	$f_{sys} / 2^{18}$
	($f_{sys} = 16 \text{ MHz}$)	61	81	122	61
0001	$f_{sys} / 2^{17}$	$f_{sys} / 2^{17}$	$(4/3) \times f_{sys} / 2^{17}$	$(4/2) \times f_{sys} / 2^{17}$	$f_{sys} / 2^{17}$
	($f_{sys} = 16 \text{ MHz}$)	122	163	244	122
	($f_{sys} = 8 \text{ MHz}$)	61	81	122	61
0010	$f_{sys} / 2^{16}$	$f_{sys} / 2^{16}$	$(4/3) \times f_{sys} / 2^{16}$	$(4/2) \times f_{sys} / 2^{16}$	$f_{sys} / 2^{16}$
	($f_{sys} = 8 \text{ MHz}$)	122	163	244	122
	($f_{sys} = 4 \text{ MHz}$)	61	81	122	61
0011	$f_{sys} / 2^{15}$	$f_{sys} / 2^{15}$	$(4/3) \times f_{sys} / 2^{15}$	$(4/2) \times f_{sys} / 2^{15}$	$f_{sys} / 2^{15}$
	($f_{sys} = 4 \text{ MHz}$)	122	163	244	122
	($f_{sys} = 2 \text{ MHz}$)	61	81	122	61
0100	$f_{sys} / 2^{14}$	$f_{sys} / 2^{14}$	$(4/3) \times f_{sys} / 2^{14}$	$(4/2) \times f_{sys} / 2^{14}$	$f_{sys} / 2^{14}$
	($f_{sys} = 2 \text{ MHz}$)	122	163	244	122
	($f_{sys} = 1 \text{ MHz}$)	61	81	122	61
1000	$f_s / 2^9$	$f_s / 2^9$	$(4/3) \times f_s / 2^9$	$(4/2) \times f_s / 2^9$	$f_s / 2^9$
	($f_s = 32.768 \text{ kHz}$)	64	85	128	64
1001	$f_s / 2^8$	$f_s / 2^8$	$(4/3) \times f_s / 2^8$	$(4/2) \times f_s / 2^8$	$f_s / 2^8$
	($f_s = 32.768 \text{ kHz}$)	128	171	256	128

注: f_{sys} = 齿轮时钟频率[Hz], f_s = 低频时钟频率[Hz]

17.3.8 内部/外部泄放电阻切换控制

LCD 偏置由泄放电阻生成。外部或内部泄放电阻均可使用。

使用内部泄放电阻时，将 LCDCR2<BRSEL> 设置为"1"。在这种情况下，将多工引脚(输入/输出端口或外部泄放电阻连接)用作输入输出。

使用外部泄放电阻时，将 LCDCR2<BRSEL>设置为"0"，并将外电阻连接到外部泄放电阻连接引脚(LV1, LV2)泄放。在这种情况下，仅可将多工引脚(输入/输出端口，外部泄放电阻连接，或节段输出)用作泄放电阻连接引脚。

有关泄放电阻的连接方式，请参看图 17-4。

17.3.9 低内部泄放电阻连接时间选择(LCDCR2<LRSE>)

内部泄放电阻由两部分组成：高电阻和低电阻。高电阻和低电阻部分并联后用于各偏置电压。低泄放电阻带有一个模拟开关，通过 LCDCR2<LRSE>调整低泄放电阻的开启时间。在模拟开关开启期间，低电阻与高电阻之间处于并联状态。其可减小电阻总量，从而增大 LCD 驱动器的驱动能力。

一般而言，低电阻连接周期越长，LCD 控制板的驱动能力越强，功耗越大。反之，低电阻连接周期越短，驱动能力越弱，功耗越小。不足的驱动能力将造成 LCD 显示模糊等不良影响。应为拟使用的 LCD 控制板选择最佳驱动能力。

表 17-2 给出了在各情况下每帧的低泄放电阻连接时间(百分比)，以及流经整个泄放电阻的估计电流。

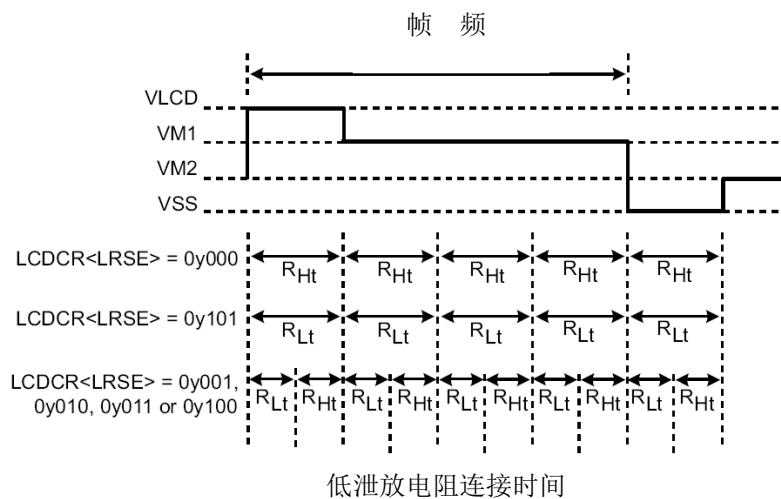
图 17-3 给出了 1/4 占空比和 1/3 偏置 LCD 的泄放电阻控制计时。

有关 LCDCR2<BRH>的设置，请参看“17.3.10 高内部泄放电阻选择(LCDCR2<BRH>)”。

表 17-2 低泄放电阻连接时间(%)和总泄放电流值(估计值)

LCDCR2 <LRSE>		1/4 占空比 (1/3 偏置)		1/3 占空比 (1/3 偏置)		1/3 占空比 (1/2 偏置)		1/2 占空比 (1/2 偏置)	
		BRH="1"	BRH="0"	BRH="1"	BRH="0"	BRH="1"	BRH="0"	BRH="1"	BRH="0"
000	低电阻 连接时间(%)	0% (始终高电阻)							
	泄放电流	2.00 μ A	5.00 μ A	2.00 μ A	5.00 μ A	3.00 μ A	7.50 μ A	3.00 μ A	7.50 μ A
001	低电阻 连接时间(%)	3.13%		2.34%		2.34%		1.56%	
	泄放电流	3.56 μ A	6.56 μ A	3.17 μ A	6.17 μ A	4.76 μ A	9.26 μ A	4.17 μ A	8.67 μ A
010	低电阻 连接时间(%)	6.25%		4.69%		4.69%		3.13%	
	泄放电流	5.13 μ A	8.13 μ A	4.34 μ A	7.34 μ A	6.52 μ A	11.02 μ A	5.34 μ A	9.84 μ A
011	低电阻 连接时间(%)	12.5%		9.38%		9.38%		6.25%	
	泄放电流	8.25 μ A	11.25 μ A	6.69 μ A	9.69 μ A	10.03 μ A	14.53 μ A	7.69 μ A	12.19 μ A
100	低电阻 连接时间(%)	25%		18.75%		18.75%		12.5%	
	泄放电流	14.50 μ A	17.50 μ A	11.38 μ A	14.38 μ A	17.06 μ A	21.56 μ A	12.38 μ A	16.88 μ A
101	低电阻 连接时间(%)	50%		37.5%		37.5%		25%	
	泄放电流	27.00 μ A	30.00 μ A	20.75 μ A	23.75 μ A	31.13 μ A	35.63 μ A	21.75 μ A	26.25 μ A
110	低电阻 连接时间(%)	100% (始终连接)							
	泄放电流	52.00 μ A	55.00 μ A	52.00 μ A	55.00 μ A	78.00 μ A	82.50 μ A	78.00 μ A	82.50 μ A

注：以上给出的泄放电阻电流值为估计值。实际电流值可随LCD负荷量和电阻值制造变化的不同而变化。



R_{Lt} : 低电阻连接的周期
(高电阻和低电阻处于并联状态)

R_{Ht} : 低电阻未连接的周期
(仅连接高电阻)

图17-3 通过LCDCR2<LRSE> (1/4 占空比, 1/3 偏置)实现的泄放电阻选择

17.3.10 高内部泄放电阻选择(LCDCR2<BRH>)

通过 LCDCR2<BRH>的设置，可按两种电平(500 kΩ (典型值)或 200 kΩ(典型值))选择电平高内部泄放电阻的值。一般而言，电阻值越小，LCD 控制板的驱动能力越强，功耗越大。反之，电阻值越大，LCD 控制板的驱动能力越弱，功耗越小。

低电阻的电阻值被固定至 20 kΩ (典型值)。低电阻与高电阻之间通过模拟开关并联，因此，可通过设置表 17-3 所给出的 LCDCR2<LRSE>调整电阻总量。

例如，将 LCDCR2<BRH>设置为"1"，可选择 19.23 kΩ(典型值)的合成电阻(当连接低电阻时)，也可选择 500 kΩ(典型值)的高电阻(当未连接低电阻时)。

表 17-3 泄放电阻值

LCDCR2<BRH>	当低电阻未被连接时	当低电阻已被连接时
1	500 kΩ (典型值)	19.23 kΩ (典型值)
0	200 kΩ (典型值)	18.18 kΩ (典型值)

17.3.11 LCD 显示工作

LCD 驱动电压 V_{LCD} 由 VLC 和 VSS 引脚($V_{LC} - V_{SS}$)之间的电势差给出。当节段和共用输出之间的电势差为 $\pm V_{LCD}$ 时，LCD 亮起。LCD 在其它时候处于熄灭状态。

应连接电源，以满足 $V_{LC} \leq V_{DD}$ 的条件。连接示例见图 17-4。

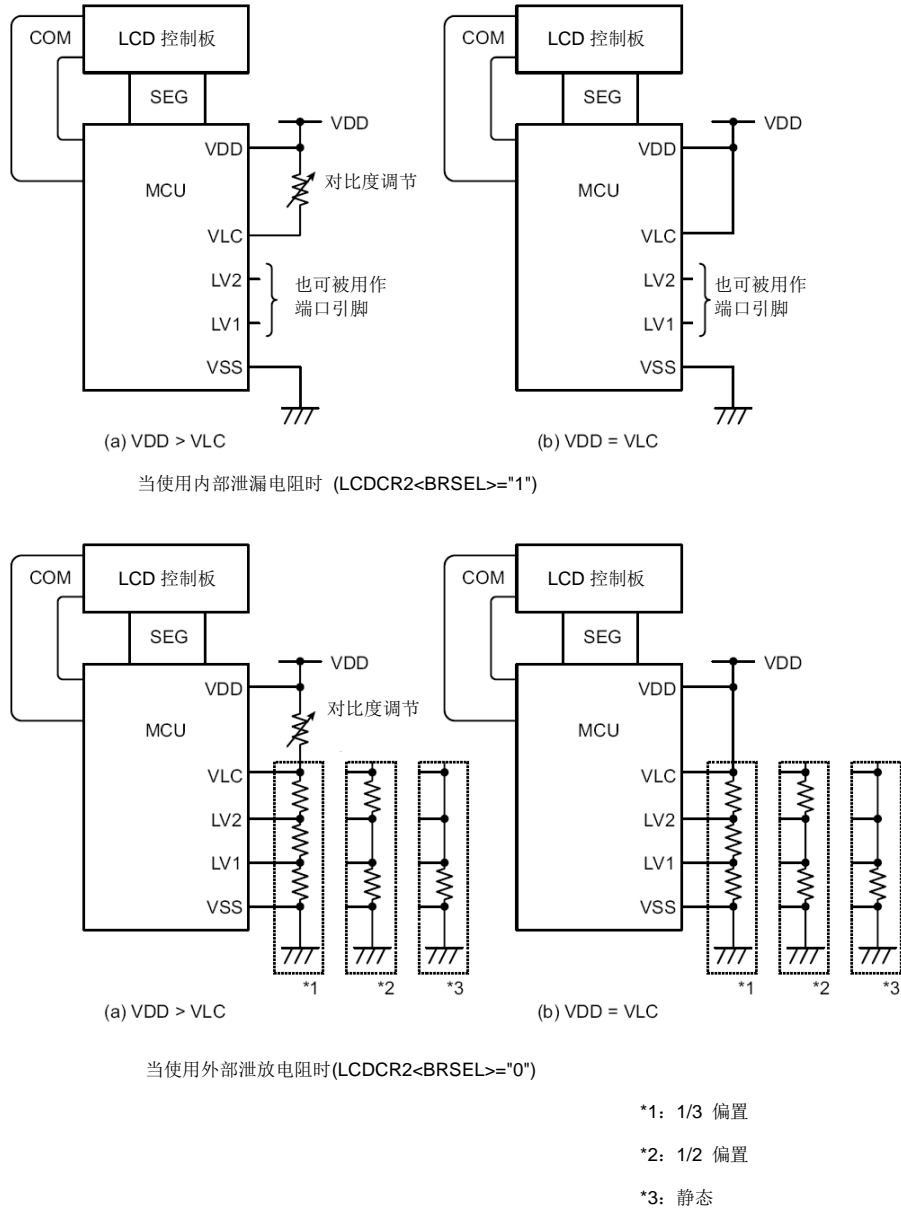


图17-4 连接示例

注 1: 当CPU 工作电压与LCD驱动电压相同时, 应将VLC引脚连接至VDD引脚。

注 2: 复位时, 共用输出引脚会变为低。但是, 各多工引脚(输入/输出端口或节段输出)可被配置为端口输入引脚(高阻抗)。因此, 如果多工引脚(输入/输出端口或阶段输出)被用作节段输出引脚, 且某外部复位信号被持续输入, 可能导致LCD显示受到模糊等不利影响。多工引脚(输入/输出端口或外部泄放电阻连接)可被配置为外部泄放电阻连接引脚。

17.3.12 显示数据设置

显示数据被存储在缓冲寄存器 00 至 19 中。

存储在显示数据区中的显示数据会被硬件自动读出, 并发送至 LCD 驱动器。LCD 驱动器可按照显示数据和驱动方法, 生成节段和共用信号。因此, 仅通过重写显示数据区的内容可改变显示模式。表 17-5 给出了显示数据区与各 SEG 和 COM 引脚之间的相应关系。

当显示数据为"1"时，LCD 亮起；当时显示数据为"0"时，LCD 熄灭。

复位时，缓冲寄存器中的数据会被初始化为"0"。

可被驱动的像素的数目会随着 LCD 驱动方法的不同而变化，显示数据存储用位数也会随之变动。因此，未被用于显示数据存储的位，以及与未接到 LCD 的各地址对应的数据存储位置，均可用于一般用户数据的存储(见表 17-4)。

表 17-4 拟用于显示数据存储的位

驱动方式	位 7/3	位 6/2	位 5/1	位 4/0
1/4 占空比	COM3	COM2	COM1	COM0
1/3 占空比	-	COM2	COM1	COM0
1/2 占空比	-	-	COM1	COM0
静态	-	-	-	COM0

注：“-”表示未用于显示数据存储的位。

表 17-5 LCD 显示数据区

寄存器名称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	读取/ 写入	初始值
LCDBUF00	SEG1			SEG0					R/W	(0000 0000)
LCDBUF01	SEG3			SEG2					R/W	(0000 0000)
LCDBUF02	SEG5			SEG4					R/W	(0000 0000)
LCDBUF03	SEG7			SEG6					R/W	(0000 0000)
LCDBUF04	SEG9			SEG8					R/W	(0000 0000)
LCDBUF05	SEG11			SEG10					R/W	(0000 0000)
LCDBUF06	SEG13			SEG12					R/W	(0000 0000)
LCDBUF07	SEG15			SEG14					R/W	(0000 0000)
LCDBUF08	SEG17			SEG16					R/W	(0000 0000)
LCDBUF09	SEG19			SEG18					R/W	(0000 0000)
LCDBUF10	SEG21			SEG20					R/W	(0000 0000)
LCDBUF11	SEG23			SEG22					R/W	(0000 0000)
LCDBUF12	SEG25			SEG24					R/W	(0000 0000)
LCDBUF13	SEG27			SEG26					R/W	(0000 0000)
LCDBUF14	SEG29			SEG28					R/W	(0000 0000)
LCDBUF15	SEG31			SEG30					R/W	(0000 0000)
LCDBUF16	SEG33			SEG32					R/W	(0000 0000)
LCDBUF17	SEG35			SEG34					R/W	(0000 0000)
LCDBUF18	SEG37			SEG36					R/W	(0000 0000)
LCDBUF19	SEG39			SEG38					R/W	(0000 0000)
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0		

17.4 LCD 驱动器控制方式示例

17.4.1 初始化

图 17-5 为 LCD 驱动器初始化进程流程图。

示例：当 LCD 驱动器拟在以下条件下运行时：

- 驱动方法：1/4 占空比，1/3 偏置
- LCD 帧频： $f_{sys}/2^{18}$ [Hz]
- 低泄放电阻(内部)的连接时间： $2^{15}/f_{sys}$
- 高泄放电阻(内部)：200 kΩ

```

LCDEN      ← 0x01    ; <LCDE> = "1"
LCDCR1     ← 0x00    ; 设置LCD驱动方法和基频
LCDCR2     ← 0x28    ; 设置低泄放电阻的连接时间，以及
                    ; 高泄放电阻值
PxFRn<PxmFn> ← 0x1    ; 设置PxFRn寄存器
                    ; (x: I/O端口编号, n: 功能寄存器编号, m; 位编号)
:          :
:          : ; 设置初始显示数据
LCDCR1     ← 0x80    ; 启用LCD显示
    
```



图 17-5 LCD驱动器初始化流程图

17.4.2 显示数据设置

一般在程序存储器(ROM)中将显示数据制作成固定数据，并通过指令将其传送到显示数据区。

示例：以下给出了利用 1/4 占空比和 1/3 偏置 LCD，对被存储在数据存储器地址 0x90 的 BCD 数据所对应编号的显示时所需显示数据进行设置的方式。图 17-6 给出了将 COM 和 SEG 引

脚连接到 LCD 的方式示例，表 17-6 给出了示例显示数据的设置方式。



图 17-6 COM和SEG引脚连接示例(1/4 占空比)

表 17-6 显示数据示例(1/4 占空比)

序号	显示	显示数据	序号	显示	显示数据
0	0.	11011111	5	5	10110101
1	1	00000110	6	6	11110101
2	2	11100011	7	7	00000111
3	3	10100111	8	8	11110111
4	4	00110110	9	9	10110111

示例：以下利用 1/2 占空比LCD，给出了示例 1 所述数字显示用显示数据的设置方式示例。图 17-7 给出了将SEG和COM引脚连接至LCD的方式示例，而表 17-7 给出了示例显示数据的设置方式。

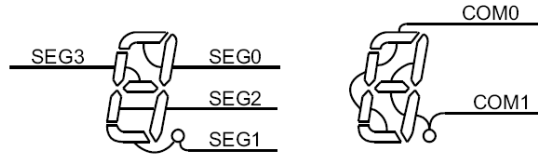


图 17-7 COM和SEG引脚连接示例

表 17-7 显示数据示例(1/2 占空比)

序号	显示数据		序号	显示数据	
	高阶地址	低阶地址		高阶地址	低阶地址
0	**01**11	**01**11	5	**11**10	**01**01
1	**00**10	**00**10	6	**11**11	**01**01
2	**10**01	**01**11	7	**01**10	**00**11
3	**10**10	**01**11	8	**11**11	**01**11
4	**11**10	**00**10	9	**11**10	**01**11

注：星号(*)表示“忽略”。

17.4.3 驱动输出示例

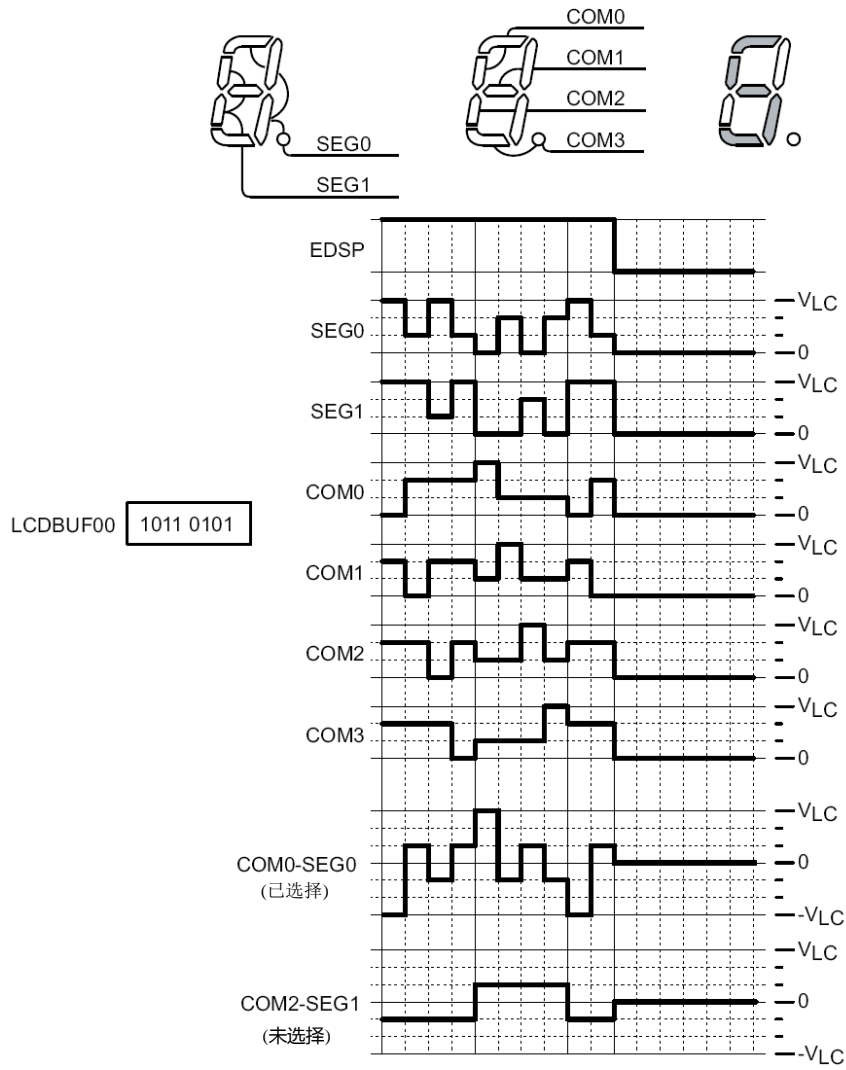


图 17-8 1/4 占空比(1/3偏置)驱动

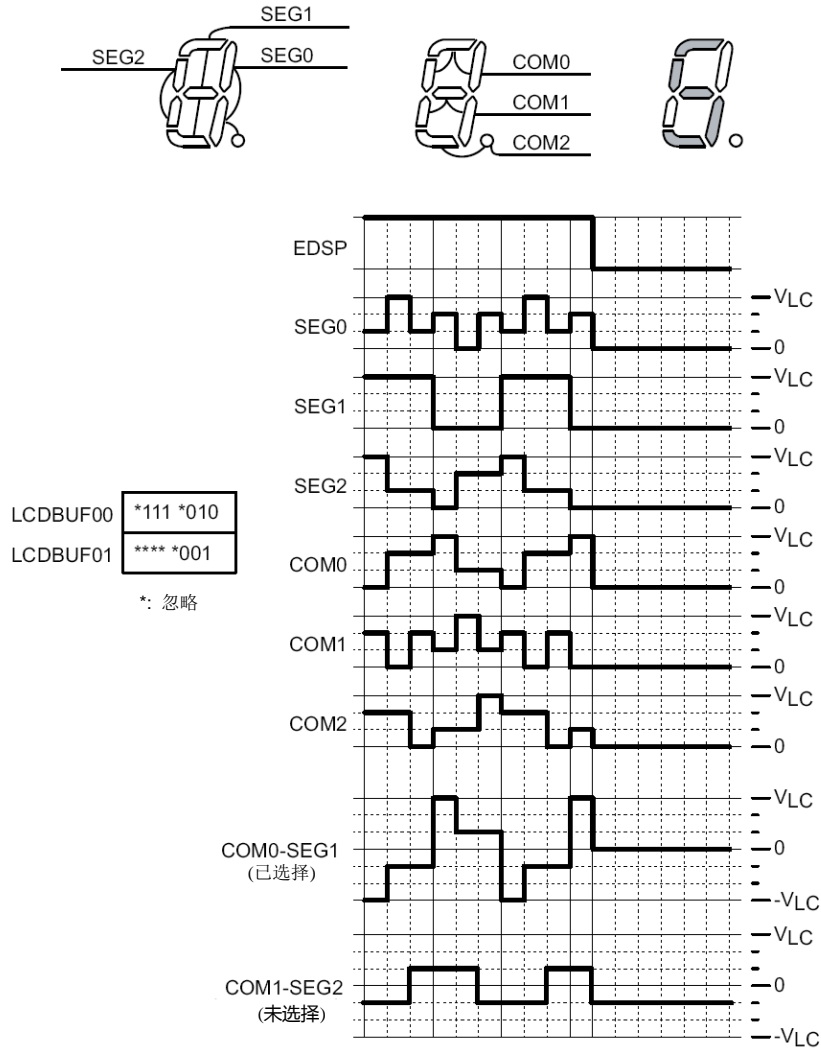


图 17-9 1/3 占空比(1/3偏置)驱动

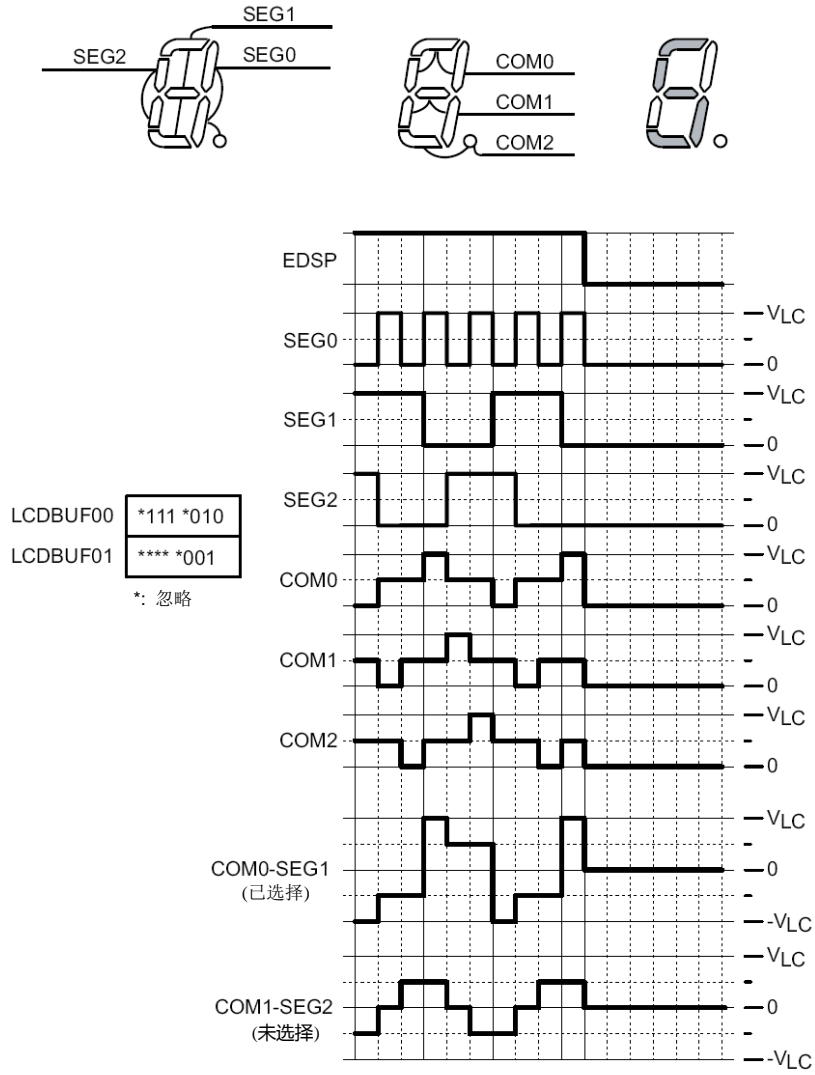


图 17-10 1/3 占空比(1/2 偏置)驱动

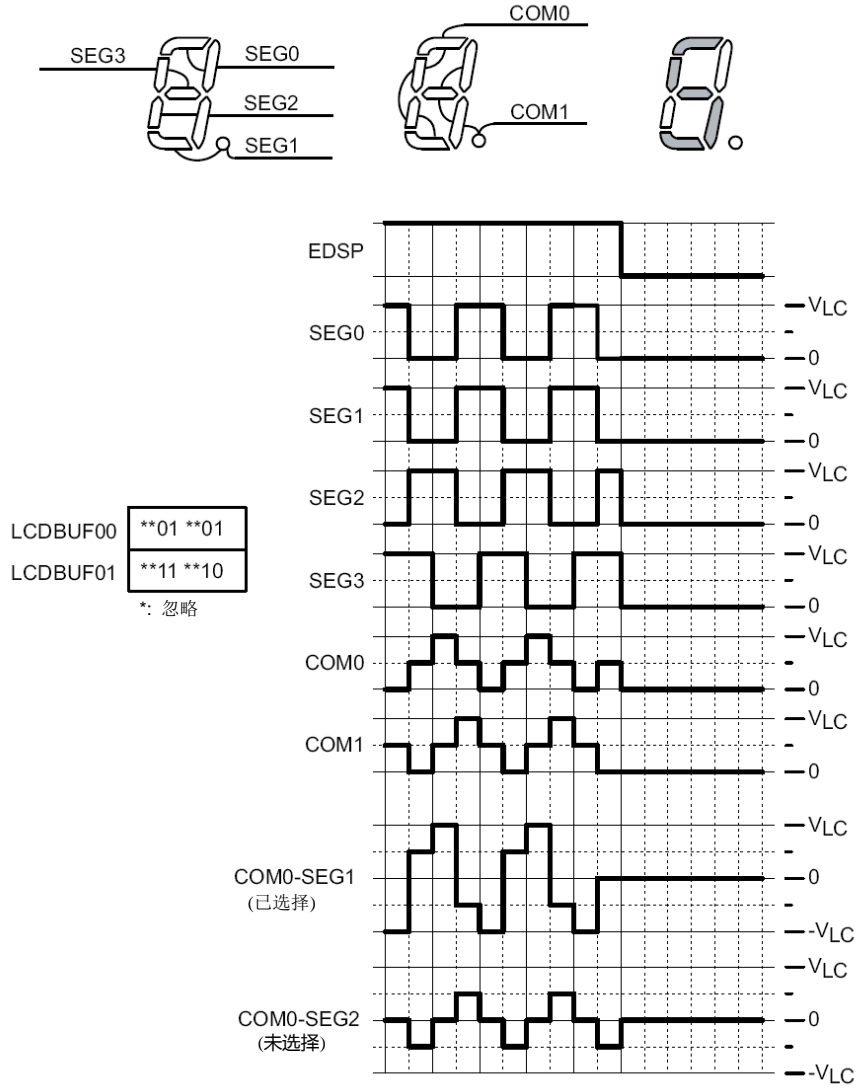


图 17-11 1/2 占空比(1/2 偏置)驱动

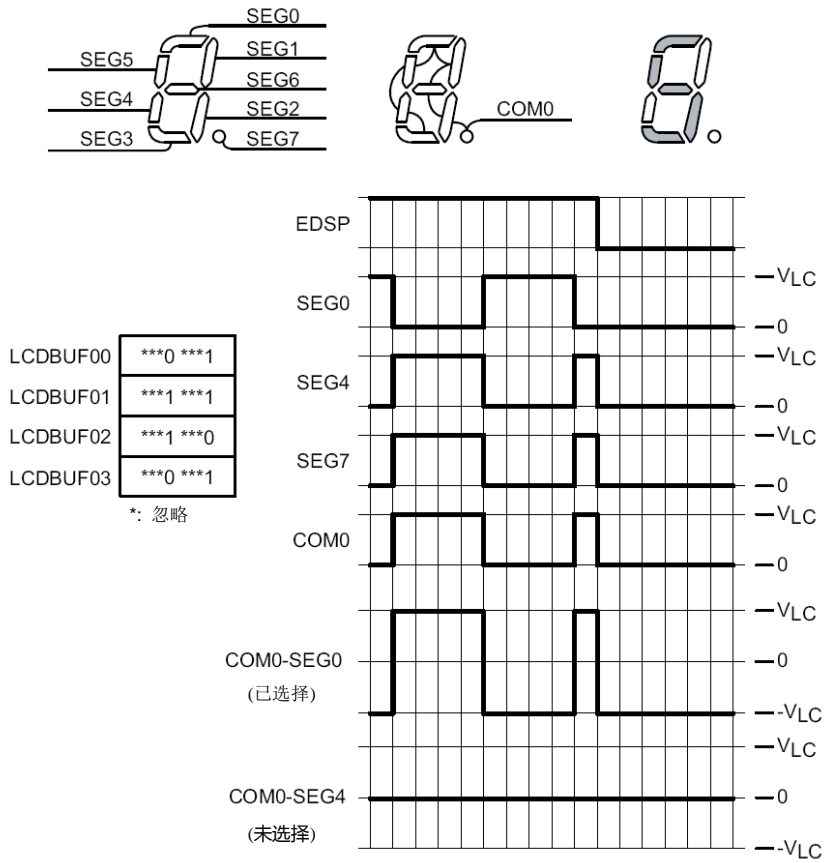


图 17-12 静态驱动

18. 低电压检测电路(LVD)

低电压检测电路通过检测渐降/渐升电压，生成一个中断信号。

电源电压表示为RVDD3。

注：上电复位电路因电源电压波动的影响而无法正确的工作。用户应根据装置设计的电气特性，予以适当考虑。

18.1 结构

低电压检测电路由参考电压生成电路，比较器和控制寄存器组成。

电源电压被梯形电阻器平分并输入到电压选择电路。在电压选择电路中，按照所检测到的电压选择某个电压，然后将其与比较器中的参考电压相比较。如果电源电压高于/低于所检测到的电压，出现一个中断信号。

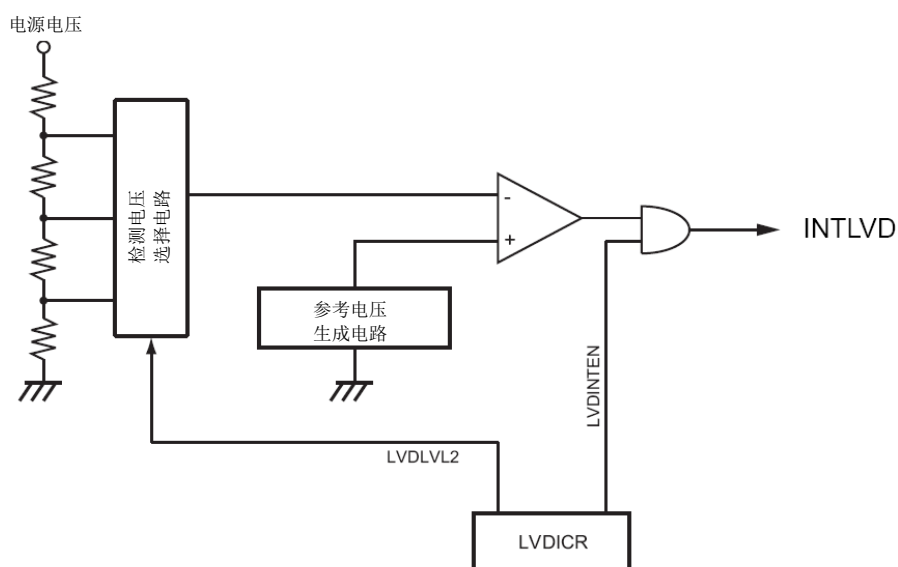


图 18-1 LVD方块图(仅涵盖LVD中断电路)

18.2 寄存器

18.2.1 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“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址 (基+)
LVD-INT控制寄存器	LVDICR	0x0004
LVD状态寄存器	LVDSR	0x0008

18.2.2 LVDICR (LVD-INT 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	-	-	LVDINTEN	INTSEL	LVDLVL2			LV DEN2
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31 - 6	-	R	读作 "0"。
5	LVDINTEN	R/W	控制INTLVD输出 0: 禁用 1: 启用
4	INTSEL	R/W	INT生成条件 0: 仅当电压渐降时低于设置电压。 1: 在电压渐降时低于设置电压，在电压渐升时高于设置电压。
3 - 1	LVDLVL2[2:0]	R/W	所检测到的电压 000: 2.80 ± 0.2 V 001: 2.85 ± 0.2 V 010: 2.90 ± 0.2 V 011: 2.95 ± 0.2 V 100: 3.00 ± 0.2 V 101: 3.05 ± 0.2 V 110: 3.10 ± 0.2 V 111: 3.15 ± 0.2 V
0	LV DEN2	R/W	低电压检测工作 0: 禁用 1: 启用

注：通过用 $\overline{\text{RESET}}$ 引脚复位，实现LVDICR的初始化。

18.2.3 LVDSR (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	-	-	-	-	-	-	LVDST2	-
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	LVDST2	R	表示LVDLVL2低电压检测状态 0: 电源电压高于检测电压。 1: 电源电压低于检测电压。
0	-	R	读作未定义。

注：通过用 $\overline{\text{RESET}}$ 引脚复位，实现LVDSR的初始化。

18.3 工作说明

18.3.1 检测电压选择和启用/禁用工作

LVDICR 寄存器进行以下设置：选择检测电压，设置工作启用/禁用，选择输出条件，以及设置输出启用/禁用。通过 RESET 引脚复位，实现 LVDICR 寄存器的初始化。

LVDICR<LVDLVL2[2: 0]>位选择检测电压。如果 LVDICR<LV DEN2>被设置为"1"，低电压检测工作被启用。

注：在电源电压低于检测电压期间，如果低电压检测工作被启用，INTLVD 就会在计时点生成。

18.3.2 低电压检测

如果电源电压低于检测电压电平，INTLVD 生成。当 LVDICR<IN- TSEL>被设置为"1"时，如果电源电压高于检测电压，INTLVD 生成。

在检测到较低的电压之后，INTLVD 的检测需要花费一定的时间。如果周期短于预期，INTLVD 未必生成。

如果电源电压低于 1.8 V，难以保证 MCU 工作。在这种情况下，必须将电源电压降低至 0 V，然后上电。

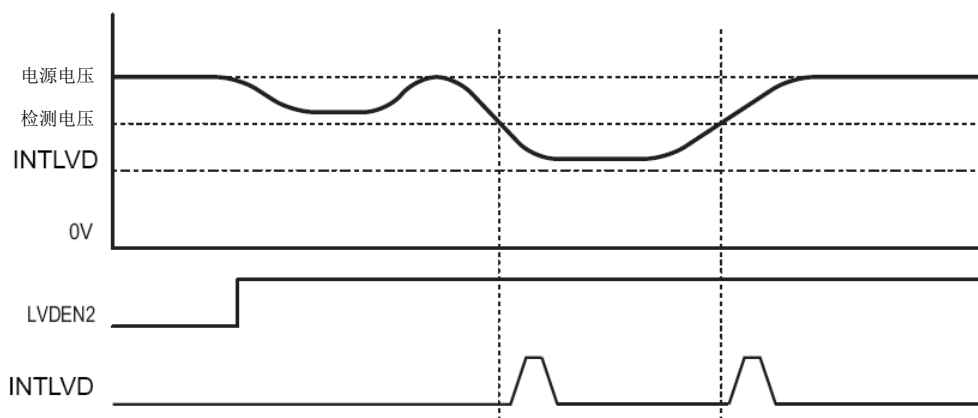


图 18-2 低电压检测时序

19.2 寄存器

19.2.1 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，参照“存储器地址”一章的“外部功能地址表”。

寄存器名称		地址 (基+)
看门狗计时器模式寄存器	WDMOD	0x0000
看门狗计时器控制寄存器	WDCR	0x0004

19.2.2 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	类型	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，以防止故障引发错误写入，首先将<WDTE>设置为"0"，然后必须将禁用代码(0xB1)写入到WDCR中。 在将看门狗计时器的状态从“禁用”改为“启用”，需将<WDTE>设置为"1"。
6-4	WDTP[2:0]	R/W	选择WDT检测时间 000: $2^{15}/f_{SYS}$ 100: $2^{23}/f_{SYS}$ 001: $2^{17}/f_{SYS}$ 101: $2^{25}/f_{SYS}$ 010: $2^{19}/f_{SYS}$ 110: 保留 011: $2^{21}/f_{SYS}$ 111: 保留
3	-	R	读作 "0"。
2	I2WDT	R/W	IDLE模式工作。 0: 停止 1: 工作
1	RESCR	R/W	在检测到故障后运行 0: INTWDT中断请求被生成。注) 1: 复位
0	-	R/W	[写入] "0"

注：INTWDT中断是非屏蔽中断的系数。

19.2.3 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	类型	功能
31-8	-	R	读作 "0"。
7-0	WDCR	W	禁用/清除代码 0xB1: 禁用代码 0x4E: 清除代码 其它: 保留

19.3 工作描述

19.3.1 基本运行

看门狗计时器由工作期间将系统时钟(fsys)用作输入的若干二进制计数器组成。

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。

在所指定的检测时间消逝后, 看门狗计时器中断(INTWDT)被生成, 且看门狗计时器输出引脚($\overline{\text{WDTOUT}}$)输出“低”。

在检测到由噪声或其它干扰所引起的CPU故障(失控)时, 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, INTWDT生成非屏蔽中断。因此, CPU会检测到故障(失控), 并会执行故障对策程序以返回正常运行状态。

此外, 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: TMPM061FWFG不具备看门狗计时器输出引脚($\overline{\text{WDTOUT}}$)。

19.3.2 运行模式和状态

一旦复位被解除, 看门狗计时器便开始运行。如看门狗计时器未被使用, 则应将其禁用。

在高速频率时钟被停止时, 无法使用看门狗计时器。在转换到以下各工作模式之前, 应禁用看门狗计时器。

在IDLE模式下, 其运行取决于WDMOD<I2WDT>设置。

- STOP 模式
- SLEEP 模式
- SLOW 模式

此外, 二进制计数器在调试模式时自动停止。

19.3.3 在检测到故障(失控)时的运行情况

19.3.3.1 INTWDT 中断生成

图 19-2 给出了INTWDT中断生成时的情况(WDMOD<RESCR>="0")。

当二进制计数器溢出发生时, INTWDT中断被生成。其为非屏蔽中断(NMI)的系数。因此, CPU会检测到非屏蔽中断, 并执行对策程序。

在INTWDT中断被生成的同时, 看门狗计时器输出($\overline{\text{WDTOUT}}$)输出“低”。

通过看门狗计时器清除将清除代码0x4E写入到WDCR, $\overline{\text{WDTOUT}}$ 就会变为“高”。

注: TMPM061FWFG不具备看门狗计时器输出引脚($\overline{\text{WDTOUT}}$)。

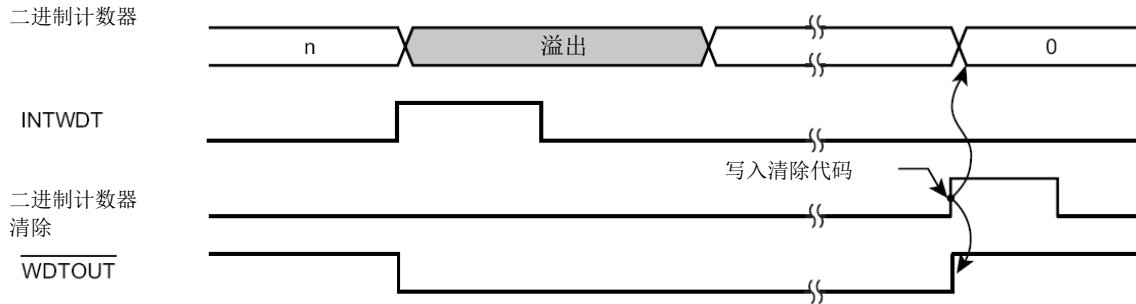


图 19-2 INTWDT中断产生

19.3.3.2 内部复位生成

图 19-3 显示内部复位 (WDMOD<RESCR>="1")。

MCU 通过二进制计数器的溢出而复位。在这种资料下，复位状态持续 32 种状态。

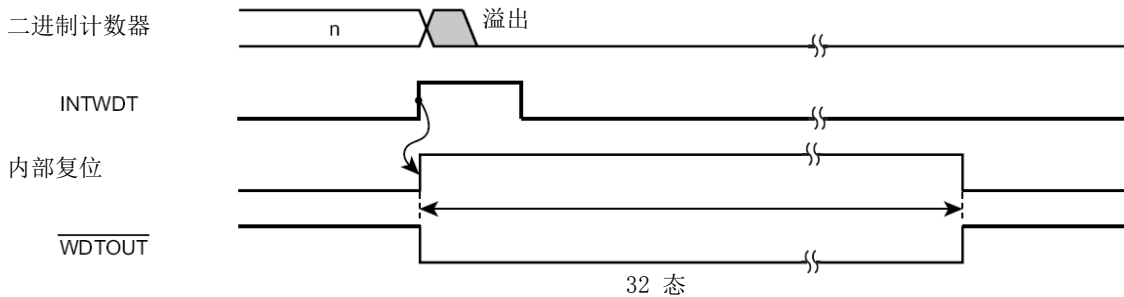


图 19-3 内部复位的生成

19.4 看门狗计时器的控制

19.4.1 禁用控制

通过在将 WDMOD<WDTE>设置为"0"后将禁用代码(0xB1)写入到 WDCR，禁用看门狗计时器，并清除二进制计数器。

19.4.2 启用控制

将WDMOD<WDTE>设置为"1"。

19.4.3 看门狗计时器清除控制

将清除代码(0x4E)写入 WDCR，清除二进制计数器，且其会重新开始计数。

19.4.4 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，如使用了 $2^{21}/f_{SYS}$ ，向WDMOD<WDTP[2:0]>设置"011"。



20. 闪存工作

本节描述了闪存的硬件配置和工作。本节"1-字"为 32 位。

20.1 特征

20.1.1 存储器容量和配置

表 20-1 和图 20-1 给出了内置存储器容量和 TMPM061FWFG 的配置情况。

表 20-1 存储器容量和配置

存储器容量	存储块配置				# 每页字 数	# 每页	写入时间		擦除时间	
	128 KB	64 KB	32 KB	16 KB			1 页	总面积	存储块擦除	芯片擦除
128 KB	-	-	4	-	32	1024	1.25 ms	1.28 秒	0.1 秒	0.2 秒

注：上述值为理论值不包括数据传送时间。每个芯片的写入时间取决于用户使用的写入方法。

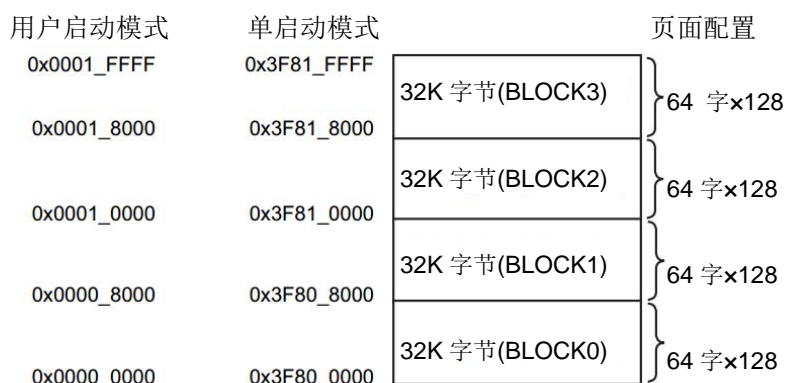


图 20-1 存储块配置

闪存配置单元被称作“存储块”和“页面”。

- 页面

一个页面是 32 个字。一个页面需使用相同的地址[31:7]。组的第 1 个地址为[6:0]=0，组的最后一个地址为[6:0]=0x7F。

- 存储块

一个存储块是 32 KB，闪存由四个存储块组成。

每页进行写工作。每页写入的时间为 1.25 ms。(典型值)

每存储块均进行擦除(使用自动存储块擦除命令)或整个闪存均进行擦除(使用自动片擦除命令)。不同擦除时间的命令。若使用自动存储块擦除命令，擦除时间将为 0.1 秒 每块(典型值)。若自动芯片擦除命令是用于擦除整个区域，时间将为 0.2 秒 (典型值)。

此外，保护功能可以为每存储块使用。有关保护功能的详细，请参照“20.1.5保护/安全功能”。

20.1.2 功能

除某些特定功能之外，该设备的内置闪存一般都符合 JEDEC 标准的要求。因此，如果某用户当前将闪存用作外部存储器，则容易将该类功能用于该设备。此外，为便于进行写入或擦除工作，该产品带有一个专用电路，负责自动执行写入或芯片擦除任务。

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 · 数据轮询/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>

20.1.3 运行模式

20.1.3.1 模式说明

该设备提供单芯片模式和单启动模式。该单芯片模式包含正常模式和用户启动模式。图 20-2 对模式转换进行了说明。

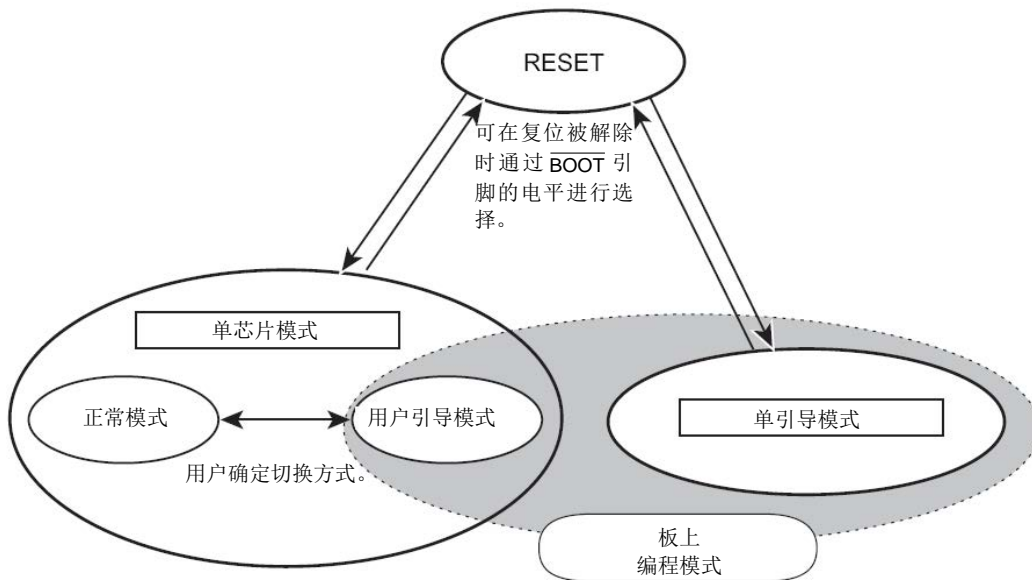


图 20-2 模式转换

(1) 单芯片模式

单芯片模式允许设备在复位后从闪存启动。该模式包含以下两种子模式。

- 正常模式
正常模式允许执行用户应用程序。
- 用户启动模式

该模式允许基于用户的设置对闪存进行重新编程。

用户可从该正常模式自由切换到用户启动模式。例如，用户可作如下设置：如果端口 A 的 PA0 为"1"，则模式为正常模式；如果端口 A 的 PA0 为"0"，则模式为用户启动模式。用户必须在该应用程序中预备一个用以确定该切换的例行程序。

(2) 单启动模式

该模式允许闪存在复位之后从内置 **BOOT ROM**(屏蔽 ROM)启动。**BOOT ROM** 包含通过用户设备串行端口重写闪存的算法。在串行端口已被连接到外部主机的情况下，可在上述协议和经重新编程的闪存中进行数据传送。

(3) 板上编程模式

用户启动模式和单启动模式，均允许闪存在用户设置的基础上进行重新编程。这两种模式被称作“板上编程模式”。

20.1.3.2 模式确定

单芯片或单个启动工作模式在复位释放时均依据 $\overline{\text{BOOT}}$ 引脚的电平来选择。

表 20-2 工作模式设置

运行模式	引脚	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
单芯片模式	0 → 1	1
单启动模式	0 → 1	0

20.1.4 存储器地址

图 20-3 给出了单芯片模式下，以及单启动模式下的存储器地址比较结果。在单启动模式下，内置闪存会被地址到 0x3F80_0000 和各后续地址，而该内置 BOOT ROM 则会被地址 0x0000_0000 至 0x0000_0FFF。

以下给出了闪存和 RAM 地址。

闪存容量	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(单启动模式)	0x2000_0000 ~ 0x2000_3FFF

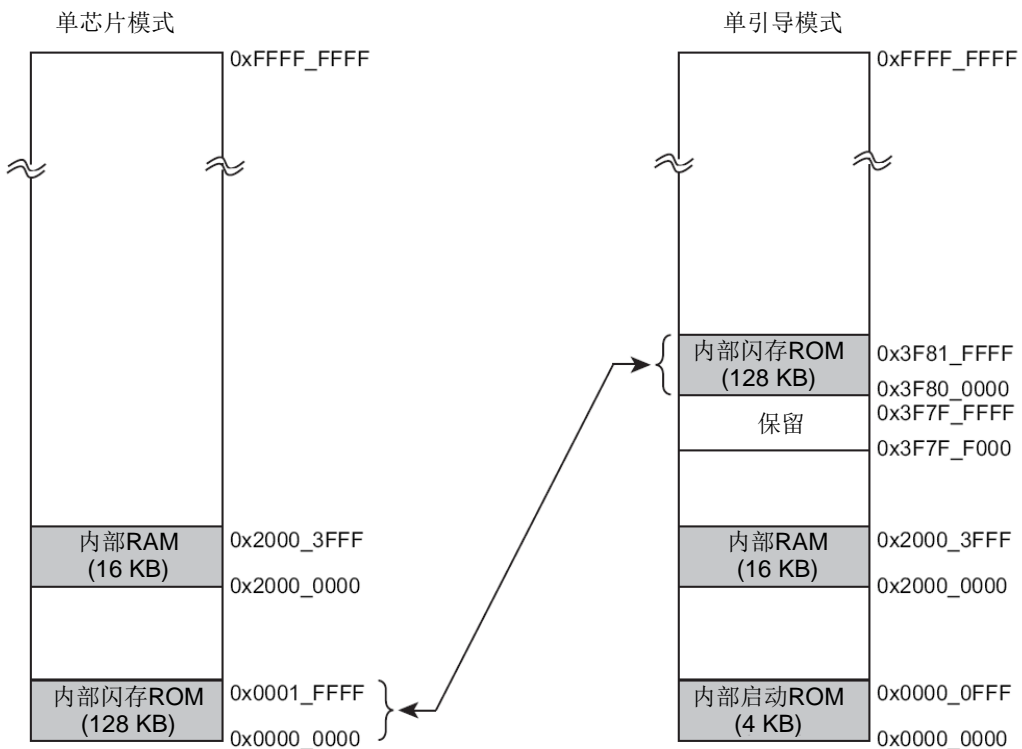


图 20-3 存储器地址的比较

20.1.5 保护/安全功能

该设备带有闪存保护和安全管理。

1. 保护功能
可按存储块禁止该写入/擦除工作。

2. 安全功能
可禁止针对闪存编写程序的读出工作。
对调试功能的使用限制



20.1.5.1 保护功能

该功能可按存储块禁止写入/擦除工作。

用该保护位程序命令将某存储块的对应保护位设置为"1",即可使能该保护功能。如果用保护位擦除命令将某个保护位设置为"0",取消存储块保护。可用 FCFLCS<BLPRO[3:0]>监视该保护位。

可通过 1-位单元对保护位程序进行编程,并可通过 4-位单元擦除保护位程序。有关保护位编程/擦除的详细,请参照"20.2.4 命令描述"。

20.1.5.2 安全功能

表 20-3 给出了安全功能被启用时的工作。

表 20-3 安全功能被启用时的工作

项目	说明
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	对闪存的命令写入未被接受。如果用户试图擦除某个保护位,则芯片擦除会被执行,且所有保护位均会被擦除。

在以下情况下,该安全功能会被启用:

1. FCSECBIT<SECBIT>被设置为"1"。
2. 所有保护位(FCFLCS<BLPRO>)均被设置为"1"。

FCSECBIT<SECBIT> 被引脚复位设置为"1"。以下对FCSECBIT <SECBIT>的重写进行说明。

注:在以下项目 1 和项目 2 写入工作时,可使用 32-位传送指令。

1. 将所指定的代码(0xa74a9d23)写入到FCSECBIT。
2. 在项目 1 运行后 16 次时钟内写入数据。

20.1.6 寄存器

20.1.6.1 寄存器列表

基址 = 0x41FF_F000

寄存器名称		地址 (基+)
安全位寄存器	FCSECBIT	0x0010
闪存控制寄存器	FCFLCS	0x0020

20.1.6.2 FCFLCS (闪存控制寄存器)

	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	-	-	-	-	BLPRO3	BLPRO2	BLPRO1	BLPRO0
复位后	0	0	0	0	(注 2)	(注 2)	(注 2)	(注 2)
	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	类型	功能
31-20	-	R	读作 "0"。
19-16	BLPRO3- BLPRO0	R	存储块 3 至 0 的保护状态 0: 无保护 1: 保护 保护位值符合每组的保护状态。如果相应的位指示"1", 则对应存储块即处于保护状态。 无法对处于保护状态的存储块进行重新编程。
15-1	-	R	读作 "0"。
0	RDY/BSY	R	准备/忙碌(注 1) 0: 忙碌(在自动工作期间) 1: 准备(自动工作结束) 该位属于功能位, 用于从CPU监视闪存。在闪存自动工作期间, 该位输出"0"表示闪存忙。一旦自动工作完成, 该位变为就绪状态"1"。然后接受下一条命令。 如果自动工作的结果为失效, 则该位连续输出"0"。该位会通过硬件复位返回为"1"。

注 1: 在发布命令之前, 应确认闪存已就绪。

注 2: 将出现该保护状态所对应的某值。

注 3: 在将<RDY/BSY> "0" 改为"1", 并等待 200 μs或以上之后, 访问闪存。

20.1.6.3 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	类型	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：通过引脚复位初始化寄存器。

20.2 闪存详细描述

在进行板上编程时，CPU会执行闪存重新编程或擦除命令。该重新编程/擦除控制程序应由用户预先准备好。在闪存被写入或擦除期间无法读取闪存的内容，因此，需在该内置RAM上运行该重新编程/擦除控制程序。请勿生成中断/故障，以免程序异常终止。

20.2.1 功能

除某些特定功能之外，闪存一般均符合 JEDEC 标准。不过，工作命令地址命名的方法不同于标准命令。

如执行写入/擦除工作，需用 32-位(1-字)存储指令命令，将各命令输入到闪存。在命令输入之后，会在内部自动执行写入或擦除工作。

表 20-4 闪存功能

主要功能	说明
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域
自动存储块擦除	自动擦除所选定的某个存储块。
写入/擦除保护	可分别禁止各存储块的写入或擦除工作。

20.2.2 闪存的工作模式

闪存主要可提供两种工作模式：

- 存储器数据读取模式(读取模式)
- 存储器数据自动擦除或重写模式(自动工作模式)

在上电之后，在停止之后，或在自动工作模式正常完成之后，闪存会变为读取模式。在读取模式下，会对存储在闪存中的指令或数据执行读取工作。

如果在读取模式下输入命令，工作模式就会变为自动工作模式。

如果命令进程正常完成，工作模式会返回到读取模式ID读取命令除外。在自动工作期间，无法读取存储在闪存中的数据，也无法执行存储在闪存中的指令。

20.2.3 如何执行命令

通过用某条存储指令将指令顺序写入到闪存，完成命令执行动作。闪存按照输入地址和数据的组合，执行各自动工作命令。有关命令执行的详细，请参照“20.2.4 命令描述”。

对闪存执行存储指令，被称为“总线写入周期”。每条命令均由若干总线写入周期组成。在闪存内部，当总线写周期的地址和数据按所指定的顺序被执行时，会同时执行自动命令工作。当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。

如果用户在该命令顺序期间取消该命令，或输入一个不同的命令顺序，则执行该读取命令或读取/复位命令。然后，闪存停止命令执行，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”。

在写入命令顺序结束时，自动工作启动，且 FCFLCS<RDY/BSY>会被设置为“0”。在自动工作正常结束时，FCFLCS<RDY/BSY> = “1”即被设置，且闪存返回到读取模式。在将<RDY/BSY> “0” 改为“1”，并等待 200 μs 或以上之后，访问闪存。



在自动工作期间，不会接受新的指令顺序。

对命令执行的注释：

1. 在识别命令时，命令顺序发生器在命令启动之前需处于读取模式。在各命令的首个总线写入周期开始之前，确认FCFLCS<RDY/BSY> = 1已被设置。建议连续执行该读取命令。
2. 从闪存的外部执行各命令顺序。
3. 按一字(32-位)内的数据传送指令，连续执行各总线写入周期
4. 请勿在各命令顺序期间访问闪存。请勿生成任何中断或故障。
5. 在发布命令发布后，如果任何地址或数据的写入不正确，应通过使用软件复位返回到读取模式。

20.2.4 命令描述

本节对各命令的内容进行说明。有关特定指令顺序的详细，请参照“20.2.5 命令顺序”。

20.2.4.1 自动页面程序

(1) 工作说明

自动页面程序可按页面写入数据。在该程序将数据写入到多个页面时，需要按页面执行一条页面命令。无法对整个页面进行写入。

“写入到闪存”，表示“1”的数据单元变为“0”的数据。“1”的数据单元则不可能变为“0”的数据。需进行擦除工作，才能将数据单元从“0”变为“1”。

仅允许将该自动页面程序用于各已被擦除页面一次。无论“1”或“0”的数据单元均无法被写入数据两次或以上。如对已被写入过一次的页面重新写入，则需在自动存储块擦除或自动芯片擦除命令被执行之后，再次设置该自动页面程序。

注 1：在未进行擦除工作的情况下对同一页面执行两次或以上的页面程序，可损坏该设备。

注 2：无法对受保护的存储块进行写入。

(2) 如何设置

第 1 至第 3 总线写入周期可指示该自动页面程序命令。

在第 4 总线写入周期内，该页面的首个地址和数据会被写入。在第 5 个总线周期期间和之后，一个页面的数据会被连续写入。数据写入按一字单元(32-位)进行。

如果该页面的一部分被写入，则可将“0xFFFFFFFF”设置为数据，表示整个页面均不需要写入。

设备内部未进行任何自动验证工作。因此，务必读取已编程的数据，以确认其写入正确。

如果自动页面程序被异常终止，则该页面的写入即告失败。建议不要使用设备或存储块包括失效的地址。

20.2.4.2 自动芯片擦除

(1) 工作说明

可对所有地址的存储器单元执行自动芯片擦除。如果包含受保护的存储块，则这些存储块不会被擦除。如果所有存储块均受保护，则不会执行自动芯片擦除工作，并会在某命令顺序被输入之后返回到读取模式。

(2) 如何设置

第 1 至第 6 总线写入周期可指示该自动芯片擦除命令。在该命令顺序被输入之后，自动芯片擦除工作即启动。

设备内部未进行任何自动验证工作。因此，务必读取该数据，以确认其已被正确擦除。

20.2.4.3 自动存储块擦除

(1) 工作说明

自动擦除命令可执行对所指定存储块的擦除工作。所指定的存储块受保护，则不会执行擦除工作。

(2) 如何设置

第 1 至第 5 总线写入周期可指示该自动存储块擦除命令。在第 6 总线写入周期内，会指定拟被擦除的存储块。在命令顺序被输入之后，自动存储块擦除即启动。

设备内部未进行任何自动验证工作。因此，务必读取该数据，以确认其已被正确擦除。

20.2.4.4 自动保护位程序

(1) 工作说明

自动保护位程序可向单个保护位写入一次"1"。可使用自动保护位擦除命令将"0"设置到某保护位。

有关保护功能的详细，请参照“20.1.5 保护/安全功能”。

(2) 如何设置

第 1 至第 6 总线写入周期可指示该自动保护位程序命令。在第 7 总线写入周期内，可指定拟被写入的保护位。在该命令顺序被输入之后，该自动保护位程序即启动。用 FCFLCS<BLPRO>检查写入工作是否被正常终止。

20.2.4.5 自动保护位擦除

(1) 工作描述

自动保护位擦除命令工作取决于安全状态。有关安全状态的详细，请参照“20.1.5 保护/安全功能”。

- 非安全状态

将所指定的保护位清除为"0"。在 4-位单元中执行保护位擦除。

- 安全状态

在擦除了闪存的所有地址之后，擦除所有保护位。

(2) 如何设置

第 1 至第 6 总线写入周期可指示该自动保护位擦除命令。在第 7 总线写入周期内，可指定拟被擦除的保护位。在该命令顺序被输入之后，自动保护位擦除工作即启动。

在非安全状态下，会擦除所指定的保护位。用 FCFLCS<BLPRO>检查擦除工作是否被正常终止。

在安全状态下，闪存位的所有地址和所有保护均会被擦除。确认数据和保护位是否被正常擦除。如有必要，则执行自动保护位擦除，自动芯片擦除或自动存储块擦除。

所有情况都与其它命令的相同，且 FCFLCS<RDY/BSY>在自动保护位擦除命令工作期间变为"0"。在该工作完成之后，FCFLCS<RDY/BSY>变为"1"，且闪存会返回到读取模式。

20.2.4.6 ID-Read

(1) 工作描述

ID-Read 命令可读取信息，包括闪存类型和制造商代码，设备代码和宏代码等三类代码。

(2) 如何设置

第 1 至第 3 总线写入周期可指示该 ID-Read 命令。在第 4 总线写入周期内，可指定拟读取的代码。在第 4 总线写入周期结束之后，在任意闪存区进行读出工作均可获取代码。

可连续执行 ID-Read。可重复执行第 4 总线写入周期和 ID 值读取。

ID-Read 命令不会自动返回到读取模式。执行该读取命令或读取/复位命令，即可返回到读取模式。

20.2.4.7 读取命令和读取/复位命令(软件位)

(1) 工作描述

命令用于使闪存返回到读取模式。

在执行时 ID-Read 命令时，宏会停止在当前状态，且不会自动返回到读取模式。使用读取命令或读取/复位命令，即可返回到读取模式。在命令被输入到中间时，其也可用于取消该命令。

(2) 如何设置

第 1 总线周期可指示该读取命令。第 1 至第 3 总线写入周期可指示该读取/复位命令。在任一命令顺序被执行之后，闪存都会返回到读取模式。

20.2.5 命令顺序

20.2.5.1 命令顺序列表

表 20-5 给出了各命令中总线写入周期的地址和数据。

所有命令周期(ID-Read 命令的第 5 总线周期除外)均属于总线写入周期。可通过 32-位(1-字)数据传送指令, 执行总线写入周期(下表仅给出了数据的下 8 位)。

有关地址的详细, 请参照表 20-6。对于表 20-6 中 Addr[15:9]一栏所述的“命令”, 请使用以下的值。

注 1) 始终将地址位[1:0]设置为“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 1 MB区或以下, 则将该位设置为“0”。
 如果总线写入到 1 MB以上的区, 则将该位设置为“1”。

表 20-5 命令顺序

命令	第 1 总线周期	第 2 总线周期	第 3 总线周期	第 4 总线周期	第 5 总线周期	第 6 总线周期	第 7 总线周期
	地址	地址	地址	地址	地址	地址	地址
	数据	数据	数据	数据	数据	数据	数据
读取	0xXX	-	-	-	-	-	-
	0xF0	-	-	-	-	-	-
读取/复位	0xX55X	0xAAX	0xX55X	-	-	-	-
	0xAA	0x55	0xF0	-	-	-	-
ID-Read	0xX55X	0xAAX	0xX55X	IA	0xXX	-	-
	0xAA	0x55	0x90	0x00	ID	-	-
自动页面程序	0xX55X	0xAAX	0xX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
自动芯片擦除	0xX55X	0xAAX	0xX55X	0xX55X	0xAAX	0xX55X	-
	0xAA	0x55	0x80	0xAA	0x55	0x10	-
自动存储块擦除	0xX55X	0xAAX	0xX55X	0xX55X	0xAAX	BA	-
	0xAA	0x55	0x80	0xAA	0x55	0x30	-
自动保护位程序	0xX55X	0xAAX	0xX55X	0xX55X	0xAAX	0xX55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
自动保护位擦除	0xX55X	0xAAX	0xX55X	0xX55X	0xAAX	0xX55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

补充说明事项

- IA: ID 地址
- ID: ID 数据
- PA: 程序页地址
- PD: 程序数据(32-位数据)
 在第 4 总线周期结束之后, 按页面和地址顺序输入数据
- BA: 存储块地址 (见表 20-7)

- PBA: 保护位地址(见表 20-8)

20.2.5.2 总线周期内的地址位配置

表 20-6 连同“表 20-5 命令顺序”一起使用。

按首个总线周期的标准总线写入周期地址配置，来进行地址设置。

表 20-6 总线写入周期中的地址位配置

地址	地址 [31:15]	地址 [14]	地址 [13:12]	地址 [11:9]	地址 [8:7]	地址 [6:4]	地址 [3:0]
标准命令	标准总线写入周期地址配置						
	闪存区	建议设置为"0"。	命令	Addr[1:0] = "0" (固定) 其它位= "0" (建议)			
ID-READ	IA: ID地址(ID-READ用第 4 总线写入周期地址的设置)						
	闪存区	建议采用"0"。	ID地址	Addr[1:0] = "0" (固定) 其它位= "0" (建议)			
存储块擦除	BA: 存储块地址(存储块擦除用第 6 总线写入周期的设置)						
	存储块地址(表 20-7)	Addr[1:0] = "0" (固定) 其它位= "0" (建议)					
自动页面程序	PA: 程序页面地址(页面程序用第 4 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)	
保护位程序	PBA: 保护位地址(保护位程序用第7总线写入周期的设置)						
	闪存区	固定为 "0"			保护位选择 (表 20-8)	Addr[1:0] = "0" (固定) 其它位= "0" (建议)	

20.2.5.3 存储块地址(BA)

表 20-7 给出了存储块地址。在自动存储块擦除命令的第 6 总线写入周期内，指定拟被擦除的存储块中的任何地址。

表 20-7 存储块地址

存储块	地址 (用户启动模式)	地址 (单启动模式)	容量 (K字节)
2	0x0001_8000 ~ 0x0001_FFFF	0x3F81_0000 ~ 0x3F81_FFFF	32
3	0x0001_0000 ~ 0x0000_7FFF	0x3F81_0000 ~ 0x3F80_7FFF	32
1	0x0000_8000 ~ 0x0000_FFFF	0x3F80_8000 ~ 0x3F80_FFFF	32
0	0x0000_0000 ~ 0x0000_7FFF	0x3F80_0000 ~ 0x3F80_7FFF	32

20.2.5.4 如何指定保护位(PBA)

在 1-位单元(适用于编程时)或 4-位单元(适用于擦除时)中指定该保护位。

表 20-8 给出了自动保护位程序的保护位选择表。地址示例栏表明, 上侧中叙述的地址用于用途启动模式, 下侧的则用于单启动模式

由自动保护位擦除命令擦除, 共计四个保护位。

表 20-8 保护位程序地址

块	保护位	第7总线写入周期的地址			地址示例 [31:0]
		地址 [14:9]	地址 [8]	地址 [7]	
块 0	<BLPRO[0]>	固定为 "0"	0	0	0x0000_0000 0x3F80_0000
块 1	<BLPRO[1]>		0	1	0x0000_0080 0x3F80_0080
块 2	<BLPRO[2]>		1	0	0x0000_0100 0x3F80_0100
块 3	<BLPRO[3]>		1	1	0x0000_0180 0x3F80_0180

20.2.5.5 ID-读取代码(IA, ID)

表 20-9 给出了利用 ID 读取命令指定代码和内容的方式。

地址栏举例表明, 上侧中叙述的地址用于用户启动模式, 下侧的则用于单个启动模式

表 20-9 ID-Read 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	0y00	0x0000_0000 0x3F80_0000
设备代码	0x5A	0y01	0x0000_1000 0x3F80_1000
-	保留	0y10	-
宏代码	0x33	0y11	0x0000_3000 0x3F80_3000

20.2.5.6 命令顺序示例

(1) 用户启动模式

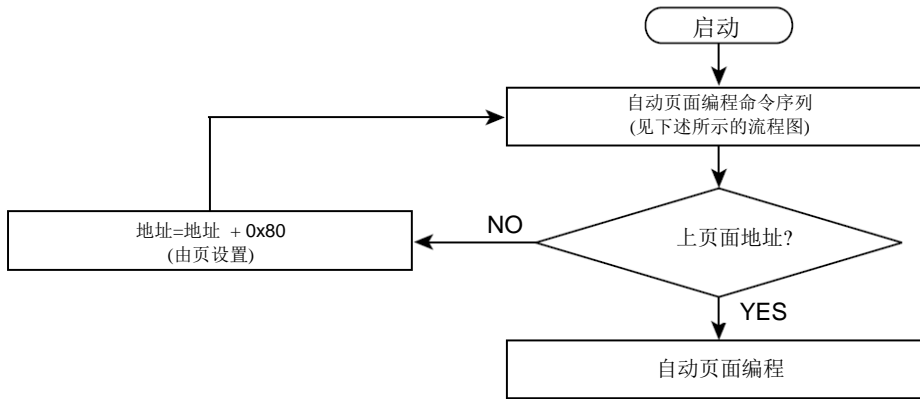
命令	总线周期							
		1	2	3	4	5	6	7
读取	地址	0x0000_0000	-	-	-	-	-	-
	数据	0x0000_00F0	-	-	-	-	-	-
读取/复位	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	-	-	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_00F0	-	-	-	-
ID-Read	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	-	-
自动页面程序	地址	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	在以下周期内，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) 数据单启动模式

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	在以下周期内，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

20.2.6 流程图

20.2.6.1 自动程序



自动页面编程命令顺序(地址/命令)

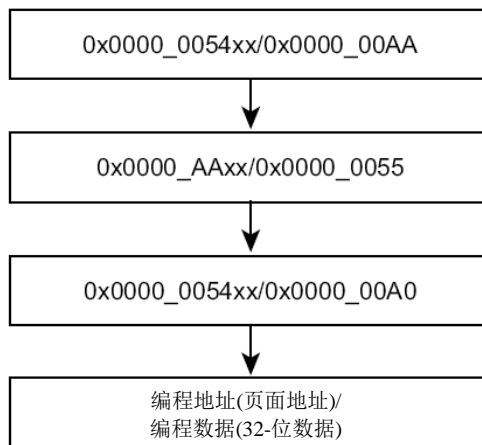


图 20-4 自动编程流程图

20.2.6.2 自动擦除

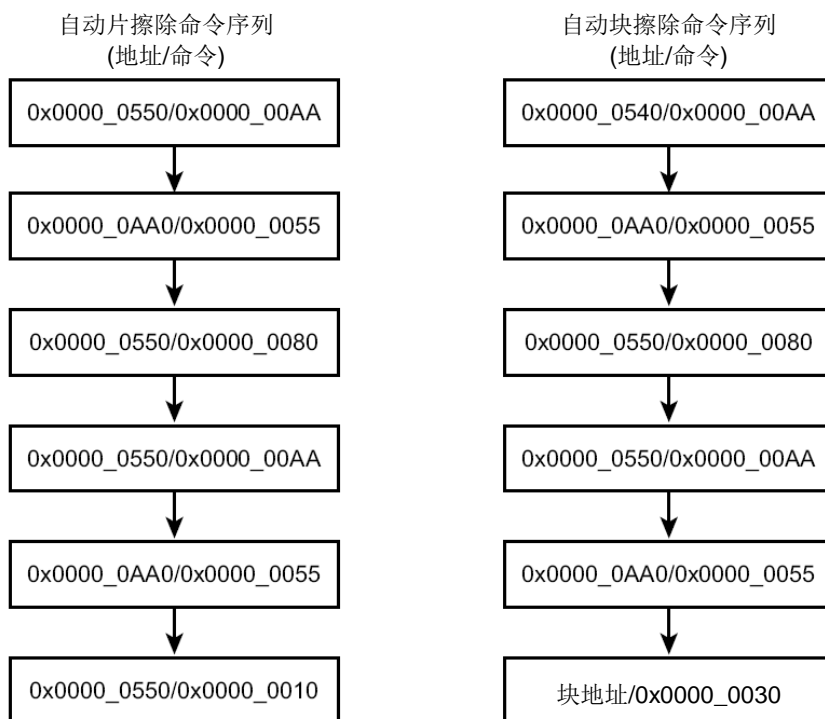
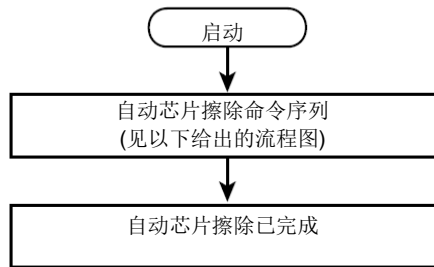


图 20-5 自动擦除流程图

20.3 如何利用单启动模式对闪存进行重新编程

该单启动模式可将内置BOOT ROM中所包含的某个程序用于闪存的重新编程。在该模式下，BOOT ROM会被地址至中断向量表所在区，而闪存则被地址至除BOOT ROM区以外的另一地址区。

在闪存模式下，闪存使用串行命令/数据传送被重新编程。在该设备的串行通道(SIO/UART)已被连接至外部主机的情况下，将某重新编程程序从外部主机复制到内置RAM。执行RAM中的重新编程例行程序，以重新编程闪存。有关与主机通信的详细，见后文所述的协议。

即使在单启动模式下也请勿生成中断/故障，以免发生程序异常终止。

为在单芯片模式(正常运行模式)下确保闪存的内容完好无损，建议在完成重新编程后的后续单芯片工作期间，防止相关闪存存储块发生意外擦除。

20.3.1 模式设置

为执行板上编程，应在单启动模式下启动该设备。以下设置适用于单模式设置。

$$\begin{aligned}\overline{\text{BOOT}} &= 0 \\ \overline{\text{RESET}} &= 0 \rightarrow 1\end{aligned}$$

在按上述要求提前设置 $\overline{\text{BOOT}}$ 引脚时，应将 $\overline{\text{RESET}}$ 引脚设置为"0"。然后解除 $\overline{\text{RESET}}$ 引脚，该装置即可在单启动模式下启动。

20.3.2 接口规格

本节对单启动模式下的SIO/UART通信格式进行说明。该串行工作同时支持UART(异步通信)和I/O接口模式。在执行板上编程时，还需设置编程控制器的通信格式。

- UART 通信

 通讯通道：通道 0

 串行传送模式：UART (非同步)，半双工，LSB 优先

 数据长度：8-位

 奇偶校验位：无

 STOP 位：1-位

 波特率：任意波特率

- I/O 接口模式

 通讯通道：通道 0

 串行传送模式 I/O 接口，全双工，LSB 优先

 同步信号(SCLK0)：输入模式，上升沿设置

 握手信号：PH3 (输出模式)

 波特率：任意波特率

时钟/模式控制存储块设置是该启动程序运行的初始条件。有关时钟初始设置的详细，请参照“时钟/模式控制”。



如“20.3.5.1 串行工作模式确定”所述，波特率通过 16-位计时器(TMRB)测定。在确定波特率时，按所需波特率的 1/16 执行通信。因此，通信波特率必须在可测量的范围之内。计时器计数时钟按 $\Phi T1 (fc/2)$ 运行。

I/O 接口模式的握手引脚可在等待接收状态输出“低”，并在发送状态输出“高”。在通信之前，应检查该握手引脚，并必须遵循该通讯协议。

表 20-10 给出了启动程序用引脚。该启动程序不拟使用除这些引脚外的其它引脚。

表 20-10 引脚连接

引脚		接口	
		UART	I/O接口模式
模式设置引脚	\overline{BOOT}	o	o
复位引脚	\overline{RESET}	o	o
通信 引脚	TXD0 (PH0)	o	o
	RXD0 (PH1)	o	o
	SCLK0(PH2)	x	O(输入模式)
	PH3	x	O(输出模式)

o: 已使用 x: 未使用

20.3.3 对内部存储器的限制

注意单启动模式对内置 RAM 和内置闪存设有表 20-11 所述的限制。

表 20-11 单启动模式下对存储器的限制

存储器	限制
内部RAM	启动程序将该存储器用作工作区0x2000_0000至0x2000_03FF。 通过RAM的结束地址存储该程序0x2000_0400。该程序的启动地址必须是偶校验地址。
内部闪存	以下地址被分配用于存储软件ID信息和密码。不建议将程序存储到以下地址。 0x3F81_FFF0 ~ 0x3F81_FFFF

注：如果密码是已被擦除的数据(0xFF)，则密码易于猜测，因此难以保护数据的安全。即使未使用单启动模式，仍建议将唯一值设置为密码。

20.3.4 工作命令

启动程序可提供工作命令。

表 20-12 工作命令数据

工作命令数据	运行模式
0x10	RAM传送
0x40	闪存芯片擦除和保护位擦除



20.3.4.1 RAM 传送

RAM 传送用于将数据从控制器储存到内置 RAM。在传送正常完成时，用户程序即启动。用户程序可将 0x2000_0400 或其后 0x2000_0000 至 0x2000_03FF 除外的存储器地址，用于该启动程序。CPU 会从 RAM 存储开始地址启动执行。该启动地址必须是偶校验地址。

RAM 传送功能可使能用用户特定的板上程序控制。在通过用户程序执行板上编程时，应采用 20.2.5 中所述的闪存命令顺序。

20.3.4.2 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，以及所有存储块的写入/擦除保护，且不考虑写入/擦除保护或安全状态。

注：即使命令完成正常，也可能返回ACK由于异常终止。在这种情况下，应再次检查擦除。

20.3.5 共用工作不考虑命令

本节对启动程序执行时的共用工作进行说明。

20.3.5.1 串行工作模式确定

在控制器通过 UART 进行通信时，需将第 1 字节设置为 0x86 在所需波特率时。在控制器通过 I/O 接口模式进行通信时，需将第 1 字节设置为 0x30 在所需波特率的 1/16 时。

图 20-6 给出了各种情况下的波形。

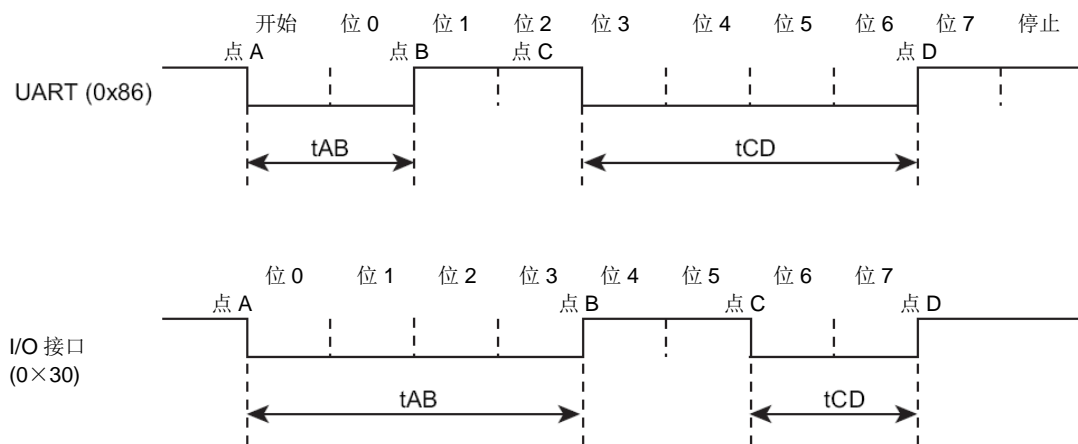


图 20-6 串行工作模式确定数据

图 20-7 给出了启动程序流程图。在使用 16-位计时器(TMRB)，且时间为 t_{AB} ， t_{AC} 和 t_{AD} 时，即可设置复位后的串行工作模式确定数据(0x86, 0x30)的第 1 字节。在图 20-7 中，CPU 可监控该接收引脚的电平，并可在接收引脚的电平被改变时获取计时器值。因此， t_{AB} ， t_{AC} 和 t_{AD} 的计时器值会存在误差容限。此外，还应注意的是，如果传送是在高波特率时进行的，则 CPU 不能测出该接收引脚的电平。尤其是 I/O 接口易于导致该问题，原因是其波特率一般远高于 UART 的波特率。为避免出现这种现象，控制器应在 I/O 接口模式下，按所要求波特率的 1/16 发送数据。

图 20-8 中的流程图给出了在串行工作模式下确定接收引脚持续时间长短的方式。如果该长度为 $t_{AB} \leq t_{CD}$ ，则串行工作模式会被确定为 UART 模式。无论自动波特率设置是否被启用，都会采用 t_{AD} 的时间。如果该长度为 $t_{AB} > t_{CD}$ ，则串行工作模式会被确定为 I/O 接口模式。注意， t_{AB} ， t_{AC} 和 t_{AD} 的计时器值有误差容限。如果该波特率为高，而工作频率为低，则各定时值都会变小。其可导致意外

确定发生(为防止发生该问题,可在编程例行程序内部重新设置 UART)。

例如,在预定模式为 UART 模式时,串行工作模式可能会被确定为 I/O 接口模式。为避免出现这种情况,在运用 UART 模式时,控制器应允许出现某个超时周期预计在该时间可从目标板接收一个回波信号(0x86)。如果控制器未能在该容许时间内获取该回波信号,则其应放弃该次通信。在运用 I/O 接口模式时,一旦发送了首个串行字节,控制器即应在某一空闲时间之后发送该 SCLK 时钟,以获取应答响应。如果所收到的应答响应不是 0x30,则控制器应放弃后续的通信。

在预定模式为 I/O 接口模式时,第一个字节不必是 0x30,只要 $t_{AB} > t_{CD}$ 如上所述即可。可将 0x91, 0xA1 或 0xB1 作为第一个字节代码发送,以确定点 A 和点 C 的下降沿,以及点 B 和点 D 的上升沿。如已确定 $t_{AB} > t_{CD}$,且该工作模式的分辨率已选择 SIO,则第二个字节代码即为 0x30,即使第一个字节上的已发送代码不是 0x30(用以确定 I/O 接口模式的第一个字节代码被描述为 0x30)。

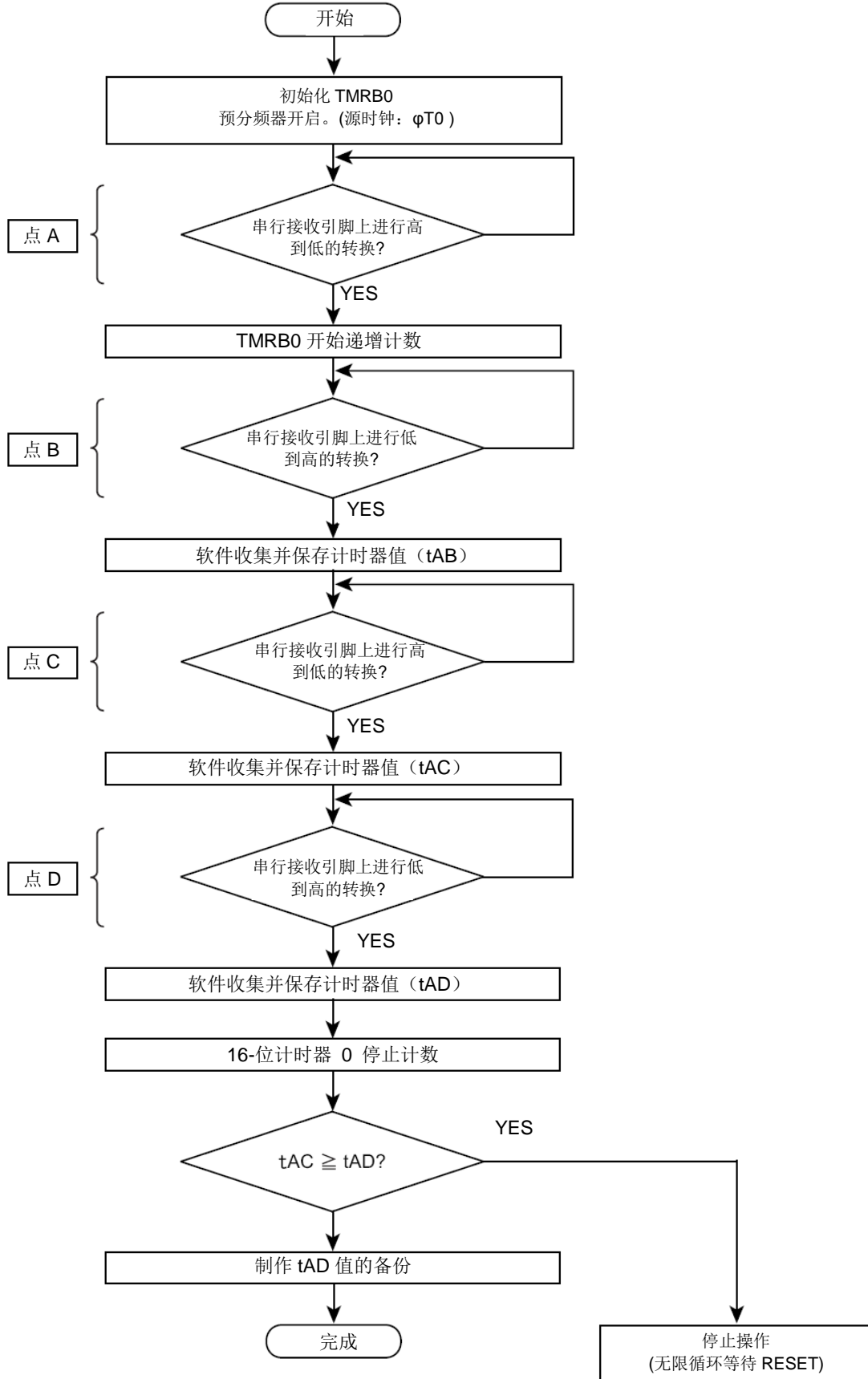


图20-7 串行工作模式接收流程图

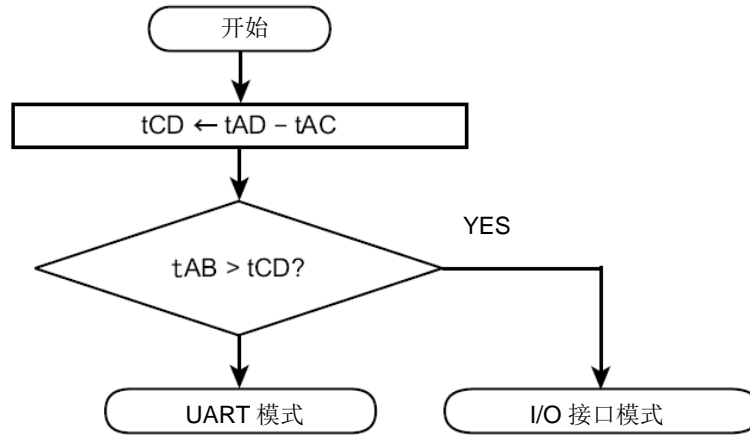


图 20-8 串行工作模式确定流程图

20.3.5.2 应答响应数据

该启动程序可用具体代码描述处理状态，并将其发送到控制器。表 20-13 至表 20-16 给出了对各接收数据应答响应值。

在 20-14 至表 20-16 中，该应答响应的上四位字节相当于工作命令数据的上四位字节。第 3 位可指示接收错误。第 0 位可指示无效的工作命令错误，校验和错误或密码错误。第 1 和第 2 位始终为 "0"。不会在 I/O 接口模式下执行接收错误检查。

表 20-13 对串行工作确定数据的 ACK 响应

传送数据	说明
0x86	已被确定为可实现UART通信。(注)
0x30	已被确定为可实现I/O接口通信。

注：在串行工作已被确定为UART时，如果波特率设置被确定为不合格，则启动程序就会放弃，且不会发回任何响应。

表 20-14 对工作命令数据的ACK响应

传送数据	说明
0x?8 (注)	在工作命令数据中出现接收错误
0x?1 (注)	已正常接收未定义工作命令数据。
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ACK响应数据的上 4 位与前一命令数据的相同。

表 20-15 对CHECK SUM数据的ACK响应

传送数据	说明
0xN8 (注)	发生接收错误。
0xN1 (注)	发生CHECK SUM或密码错误。
0xN0 (注)	CHECK SUM值正确。

注：该ACK响应数据的上 4 位与工作命令数据的相同。



表 20-16 对闪存芯片擦除和保护位擦除工作的ACK响应

传送数据	说明
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。

20.3.5.3 密码确定

启动程序可利用以下区来确定某密码是否有必要性，或是否可用作密码。

面积	地址
密码要求确定	0x3F81_FFF0 (1 字节)
密码区	0x3F81_FFF4 ~ 0x3F81_FFFF (12 字节)

RAM 传送命令可在不考虑必要性判定数据的情况下执行密码验证。闪存芯片擦除或保护位擦除命令仅可在必要性判定被确定为“必要”时执行密码验证。

密码要求设置	数据
需要密码	0xFF除外
无密码	0xFF

如果密码被设置为 0xFF(已被擦除的数据)，则密码易于猜测，因此难以保护数据的安全。即使未使用单启动模式，仍建议将唯一值设置为密码。

(1) 用 RAM 传送指令进行的密码验证

如果所有地址位置包含相同的数据字节(0xFF 除外)，则该状态会被确定为密码区错误(如图 20-9 所示)。在这种情况下，启动程序会根据校验和值第 17 字节返回一个错误应答(0x11)，且不考虑密码验证。

启动程序可验证接收数据(密码数据)的第 5 字节到第 16 字节。如果并非所有 12 个字节都匹配，则发生密码错误。如果密码错误已确定，则对第 17 个 CHECK SUM 数据的 ACK 响应数据即为密码错误。

即使安全功能已被启用，也可执行密码验证。

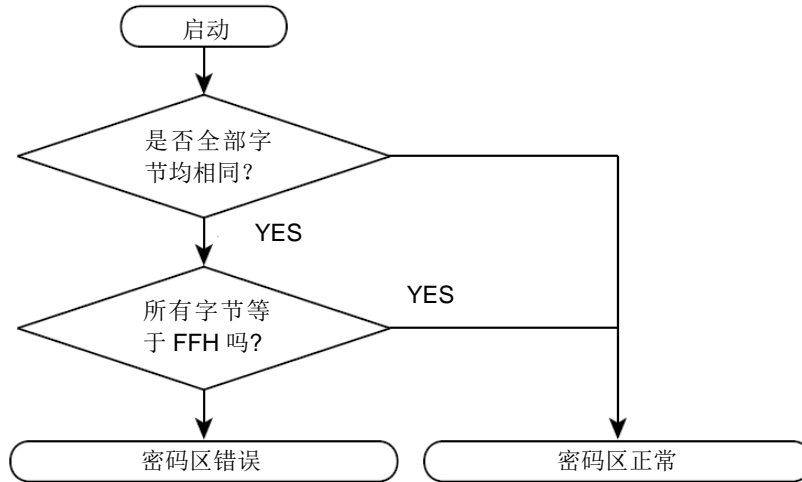


图 20-9 密码区检查流程图

(2) 对闪存芯片擦除和保护位擦除命令的密码验证

当在图 20-10 所示的擦除密码必要性确定区内启用某个密码，且各密码是完全相同的数据时，就会发生密码区错误。如已确定密码区错误，则对 CHECK SUM 第 17 字节的应答响应会发送 0x41，且不考虑密码验证。

该启动程序可验证接收数据(密码数据)的第 5 字节到第 16 字节。如果并非所有 12 个字节都匹配，则发生密码错误。如果密码错误已确定，则对第 17 个 CHECK SUM 数据的 ACK 响应数据即为密码错误。

即使安全功能已被启用，也可执行密码验证。

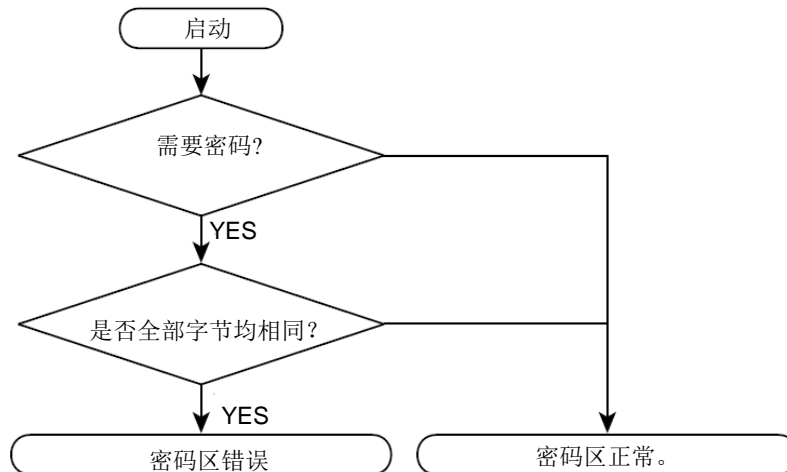


图 20-10 密码区检查流程图

20.3.5.4 CHECK SUM 计算

通过添加 8-位到发送数据，撤消各进位，并取这两者之和的余数，来计算该校验和。在发送各校验和字节时，控制器必须执行相同的校验和工作。

CHECK SUM 示例



执行 8-位加法，即可计算出数列 0xE5 和 0xF6 的校验和。

$$0xE5 + 0xF6 = 0x1DB$$

将该和的两者的余数置于下 8-位，其即为校验和值。因此，该启动程序可将 0x25 发送到控制器。

$$0 - 0xDB = 0x25$$

20.3.6 RAM 传送时的传送格式

本节对 RAM 传送命令格式进行说明。该表列出了以下传送方向：

传送方向(C→T)：控制器至 TMPM061FWFG

传送方向(C←T)：TMPM061FWFG 至控制器

传送字节数	传送方向	传送数据	说明
1	C→T	串行工作模式和波特率设置	发送用于确定串行工作模式的数据。有关模式确定的详细，请参照“20.3.5.1 串行工作模式确定”。
		[UART模式] 0x86	发送 0x86。如已确定为UART模式，则该程序可确定某波特率设置是否可实现。否则，程序会停止运行，通信随即被中止。
		[I/O接口模式] 0x30	按所需波特率的 1/6 发送 0x30。同时按所需波特率的 1/6 发送第 2 字节。从第 3 字节或后续字节开始，用户可按合乎需要的波特率发送该数据。
2	C←T	对串行工作模式的ACK响应	发送数据的第 2 字节，是对与串行工作设置模式数据对应的第 1 字节的ACK响应数据。如该设置可行，则设置SIO/UART。在发送缓冲器被写入到该数据之前，应设置接收使能计时。
		[UART 模式] 正常状态：(0x86)	如果该设置被确定为可行，则发送 0x86。否则，工作会被异常终止，且不会发回任何响应。 在控制器完成数据第 1 字节的发送时，会要求提供某超时时间(5 秒钟)。如果数据(0x86)并非是在超时时间内正常接收的，则通信不可实现。
		[I/O 接口模式] 正常状态：0x30	将数据(0x30)写入到发送缓冲器，并等待SCLK0时钟。在控制器完成数据第 1 字节的发送若干ms(空闲时间)之后，输出SCLK时钟。此时，可将波特率设置为所需波特率的 1/16。如果接收数据为 0x30，则通信可实现。在第 3 字节之后，应将所需波特率设置到通信。
3	C→T	工作命令数据(0x10)	发送RAM传送命令(0x10)。
4	C←T	对工作命令的ACK响应 正常状态：0x10 异常状态：0xX1 通讯错误：0xX8	对工作命令的ACK响应数据。 首先检查接收数据的第 3 字节是否有错误(仅适用UART模式)。如果存在接收错误，则发送表示异常通信的ACK响应数据 0xX8，并等待下一条工作命令(第 3 字节)。发送数据的上 4 位未定义。(与工作命令前的上 4 位相同)。注意，不会在I/O接口模式下进行接收错误检查。 然后，如果接收数据的第 3 字节对应表 20-12 中的任一工作命令数据，则接收数据会被反射回来。如果是RAM传送，则 0x10 会被返回，且传送数据会转至RAM传送服务程序。如果该数据不符合表 20-12 中的命令，则发送表示工作命令错误的ACK响应数据 0xX1，然后等待下一条工作命令(第 3 字节)。发送数据的上 4 位未定义(与工作命令数据前的上 4 位相同)。
5 ~ 16	C→T	密码数据 (12-字节) 0x3F81_FF04 ~ 0x3F81_FF0F	检查密码区中的数据。有关密码区检查的详细，请参照“20.3.5.3 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。如果该数据不匹配该地址，则会设置密码错误标志。

传送字节数	传送方向	传送数据	说明
17	C→T	CHECK SUM值的第 5 到第 16 字节	发送CHECK SUM值的第 5 到第 16 字节。 有关CHECK SUM计算的详细, 请参照 20.3.5.4。
18	C←T	对CHECK SUM数值的ACK响应 正常状态: 0x10 异常状态: 0x11 通讯错误: 0x18	首先, 检查是否接收数据的第 5 至第 17 字节存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK响应数据 0x18, 意为通讯异常, 等待下一工作指令 (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, 等待下一工作指令 (第 3 字节)。 最后, 检查密码验证结果。若存在密码错误, 发送ACK响应数据 0x11 意为密码错误, 等待下一工作指令 (第 3 字节)。 若所有步骤正常结束, 发送正常ACK响应数据 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 至第 24 位。第 22 字节对应地址的第 7 至第 0 位。 通过RAM的最后地址, 将该地址指定给地址 0x2000_0400。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	
24	C→T	RAM存储字节数 7 ~ 0	设定字节数, 以进行块传送。第 23 字节对应传送字节的第 15 至第 8 位。第 24 字节对应传送字节的第 7 至第 0 位。指定拟存储于自RAM地址 0x2000_0400 直至最后地址的数据。
25	C→T	CHECK SUM 数值的第 19 至第 24 字节	发送CHECK SUM值的第 19 至第 24 字节。
26	C←T	对CHECK SUM数值的ACK响应 正常状态: 0x10 异常状态: 0x11 通讯错误: 0x18	首先, 检查是否接收数据的第 19 至第 25 字节存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK响应数据 0x18, 意为通讯异常, 等待下一工作指令 (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, 等待下一工作指令 (第 3 字节)。 若所有步骤正常结束, 发送正常ACK响应数据 0x10。
27 to m	C→T	RAM已存数据	发送用于RAM已存储数据, 于第 23 至第 24 字节中指定数据的相同字节。
m+1	C→T	CHECK SUM 数值的 27 至第 m 字节	发送CHECK SUM值的第 27 到第 m 字节
m+2	C←T	对CHECK SUM数值的ACK响应 正常状态: 0x10 异常状态: 0x11 通讯错误: 0x18	首先, 检查是否接收数据的第 27 至第 m+1 字节存在错误。(仅限于UART模式)若存在接收错误, 发送ACK响应数据 0xX8, 意为通讯异常, 等待下一工作指令(第 3 字节)。然后, 检查CHECK SUM数据的m+1 字节。若错误存在, 发送 0x11, 等待下一工作指令 (第 3 字节)。若所有步骤正常结束, 发送正常ACK响应数据 0x10。
-	-	-	若ACK响应数据的第m+2 字节为正常ACK响应数据, 传送数据分至第 19 至第 22 字节指定的地址。

20.3.7 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。该表列出了以下传送方向：

传送方向 (C→T)：控制器至 TMPM061FWFG

传送方向 (C←T)：TMPM061FWFG 至控制器

传送字节数	传送方向	传送数据	说明
1	C→T	串行工作模式和波特率设置	发送用于确定串行工作模式的数据。有关模式确定的详细，请参照“20.3.5.1 串行工作模式确定”。
		[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，则进行停止通讯工作。
		[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	对串行工作模式的ACK响应	发送数据的第 2 字节，是对与串行工作设置模式数据对应的第 1 字节的ACK响应数据。如该设置可行，则设置SIO/UART。在发送缓冲器被写入到该数据之前，应设置接收使能计时。
		[UART 模式] 正常状态：0x86	如果该设置被确定为可行，则发送 0x86。否则，工作会被异常终止，且不会发回任何响应。
		[I/O 接口模式] 正常状态：0x30	在控制器完成数据第 1 字节的发送时，会要求提供某超时时间(5 秒钟)。如果数据(0x86)并非是在超时时间内正常接收的，则通信不可实现。
3	C→T	工作命令数据(0x40)	将数据(0x30)写入到发送缓冲器，并等待SCLK0时钟。在控制器完成数据第1字节的发送若干ms(空闲时间)之后，输出SCLK时钟。此时，可将波特率设置为所需波特率的1/16。如果接收数据为 0x30，则通信可实现。在第3字节之后，应将所需波特率设置到通信。
4	C←T	对工作命令的ACK响应 正常状态：0x40 异常状态：0xX1 通讯错误：0xX8	发送闪存芯片擦除和保护位擦除命令数据(0x40)。 对工作命令的ACK响应数据。 首先检查接收数据的第 3 字节是否有错误(仅适用UART模式)。如果存在接收错误，则发送表示异常通信的ACK响应数据 0xX8，并等待下一条工作命令(第 3 字节)。发送数据的上 4 位未定义。(与工作命令数据前的上 4 位相同)。注意，不会在I/O接口模式下进行接收错误检查。 然后，如果接收数据的第 3 字节对应表 20-12 中的任一工作命令数据，则接收数据会被反射回来。如果数据不对应表 20-12 中的命令，则发送表示工作命令错误的ACK响应数据 0xX1，并等待下一条工作命令。(第 3 字节) 传送数据高 4 位未定义。(使用了和紧接工作命令前的高 4 位。)
5 ~ 16	C→T	密码数据 (12-字节) 0x3F81_FF04 ~ 0x3F81_FF0F	若密码必要性设为“无”，该数据为虚拟数据。 若密码必要性设为“必要”，检查密码区数据。有关密码区数据检查的方法，请参看“20.3.5.3 密码确定”。 按顺序比较接收数据的第 5 至第 16 字节与闪存数据的 0x3F81_FFF0 至 0x3F81_FFFF。若数据不相匹配，设定密码错误标志。
17	C→T	CHECK SUM值的第 5 至第 16 字节	发送CHECK SUM值的第 5 至第 16 字节。 有关CHECK SUM计算的方法，请参看“20.3.5.4CHECK SUM计算”。

传送 字节数	传送 方向	传送数据	说明
18	C←T	对CHECK SUM值的ACK响应 正常状态: 0x40 异常状态: 0x41 通讯错误: 0x48	若密码必要性设为"无", 发送正常ACK响应数据 0x40。 若密码必要性设为"必要", 首先检查是否接收数据第 5 至第 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响应数据0x48意为通讯异常, 等待下一个工作命令。(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, 等待下一个工作命令(第 3 字节) 最后, 检查密码验证结果。若密码错误存在, 发送ACK响应数据 0x41, 意为密码错误, 等待下一个工作命令(第 3 字节) 若所有步骤正常结束, 发送正常ACK响应数据 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响应 正常状态: 0x54 异常状态: 0xX1 通讯错误: 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响应数据(第 3 位) 0x58, 意为通讯异常, 等待下一个工作命令(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, 接收数据获响应(正常ACK响应数据)。这种情况下, 0x54 获响应, 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响应数据 (第 0 位) 0xX1, 等待下一个工作命令。传输数据高 4 位未定义。(使用了和紧接工作命令前的高4位。)
21	C→T	对擦除命令的ACK响应 正常状态: 0x4F 异常状态: 0x4C	若工作正常完成, 结束码(0x4F)被返回。 若擦除错误发生, 错误码(0x4C) 被返回。
-	-	-	等待下一个工作命令。

20.3.8 启动程序完整流程图

本节显示启动程序的整个流程图。

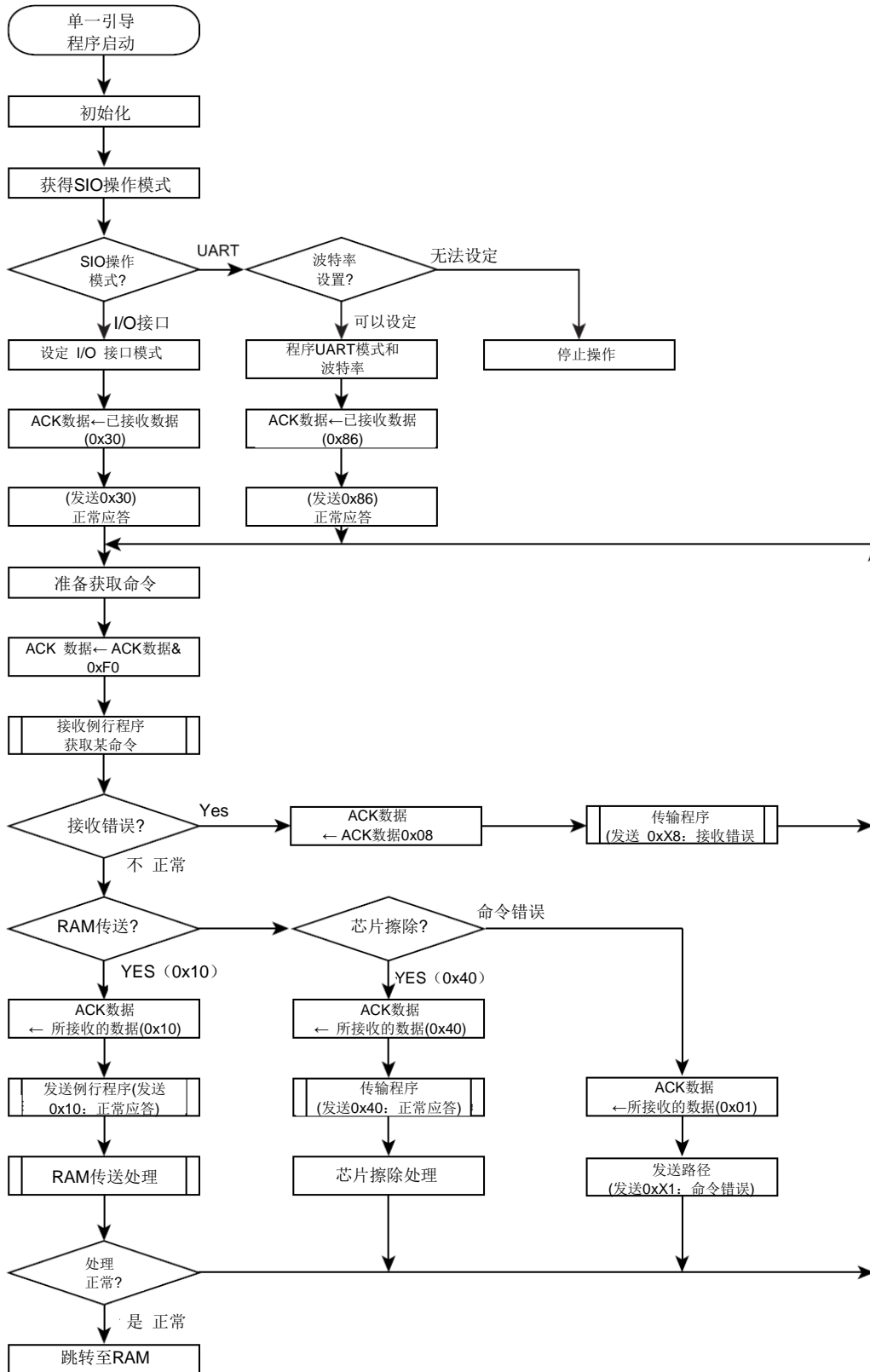


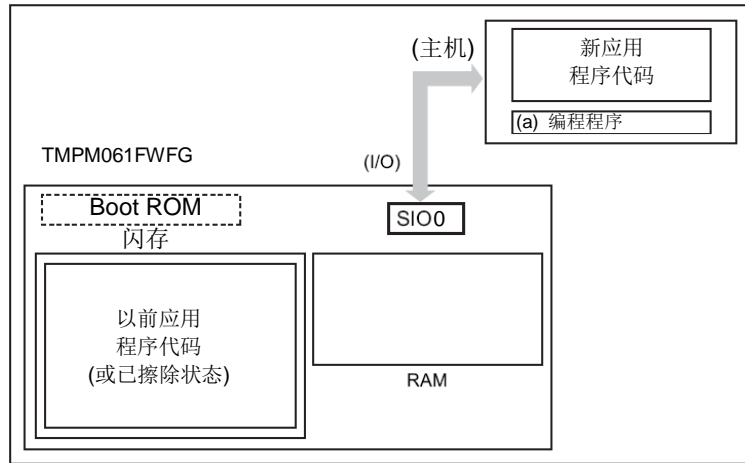
图 20-11 启动程序全流程图

20.3.9 芯片上 BOOT ROM 中使用重编程序算法重编闪存程序

本节所述为片上启动 ROM 中使用重编程序算法重编闪存程序。

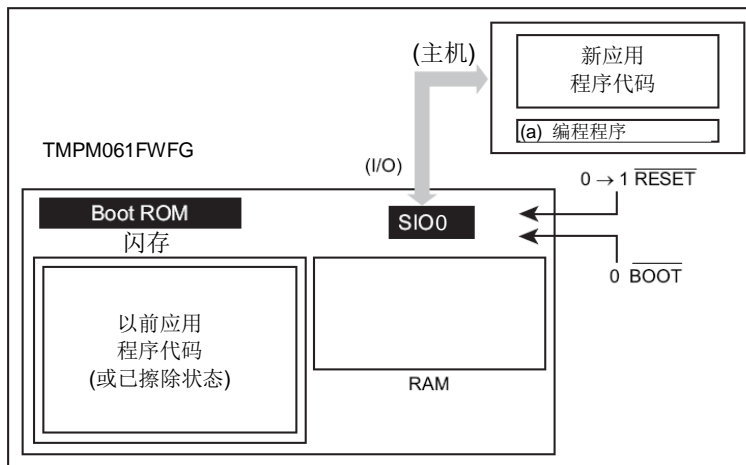
20.3.9.1 步骤-1

闪存条件不在乎是否前版构成的用户程序已写入或擦除。既然编程程序编程数据通过 SIO (SIO0) 被传送, SIO0 必定已连接至外部主机。编程程序(a)在主机上已准备好。



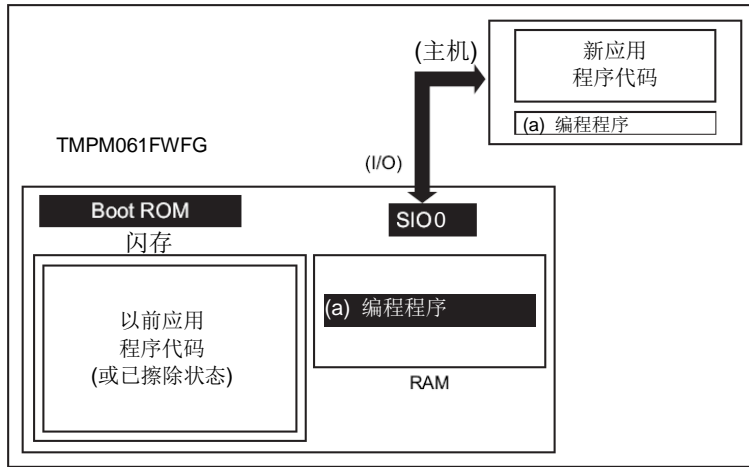
20.3.9.2 步骤-2

通过启动模式中的引脚条件设置释放复位, 启动 BOOT ROM。根据启动模式步骤, 将编程程序 (a) 通过 SIO0 自源(主机)进行传送。用户应用程序有密码的, 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。)



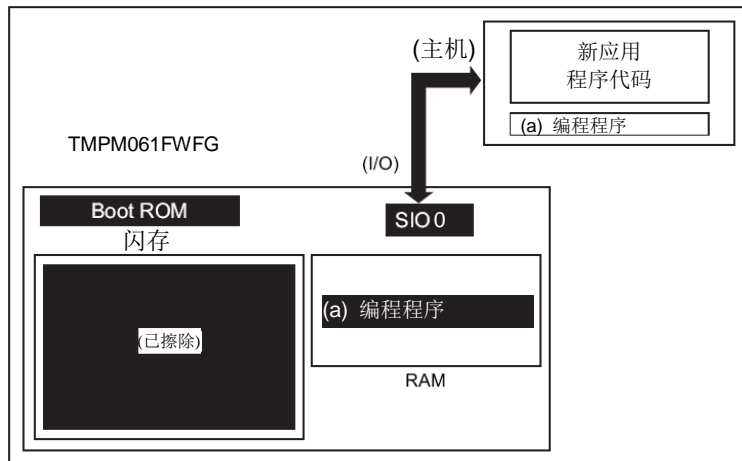
20.3.9.3 步骤-3

如果密码验证完成，则该启动程序会从主机传送一个编程例行程序(a)到片上 RAM。该编程程序必须存入 0x2000_0400 到 RAM 终端地址之间的范围内。



20.3.9.4 步骤-4

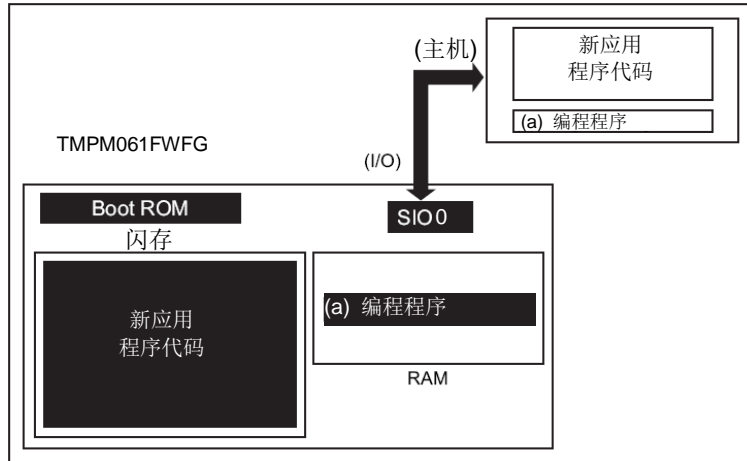
启动程序会跳转到该片上 RAM 中的该编程程序(a)，以擦除内含旧应用程序代码的闪存存储块。使用存储块擦除或芯片擦除命令



20.3.9.5 步骤-5

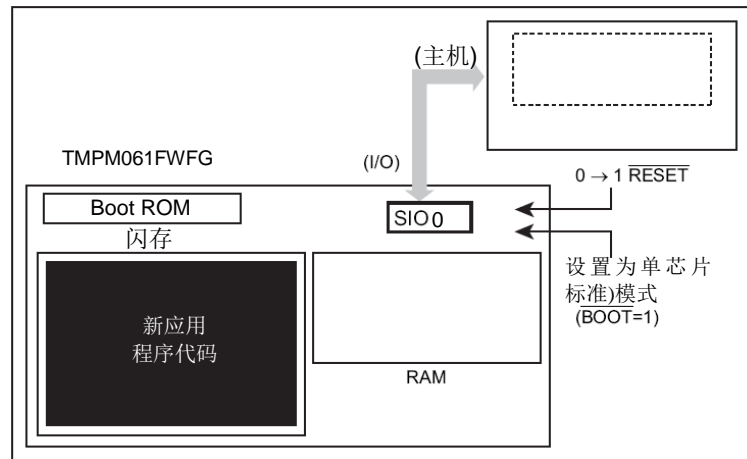
启动程序执行编程程序(a)自主机下载新应用程序代码，并将其编程至已擦除闪存区。编程完成时，须在用户程序中设定写入或擦除该闪存区保护。

以下示例中，新程序代码通过相同 SIO0 通道出自相同主机用于编程程序。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特定需求。



20.3.9.6 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，令装置以单片机(正常)模式重启以执行新程序。



20.4 用户启动模式下编程

用户启动模式为使用用户定义的闪存编程程序。当用户应用上的闪存程序代码的数据传送母线不同于串行I/O时，会使用用户启动模式。其需在单芯片模式下运行；因此，必须从正常模式用户应用程序在用途启动模式下被激活切换到闪存编程所需的用户启动模式。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。该程序在切换至用户启动模式后用于编程。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。确保不得生成中断/故障，以避免在用户启动模式期间发生异常终止。

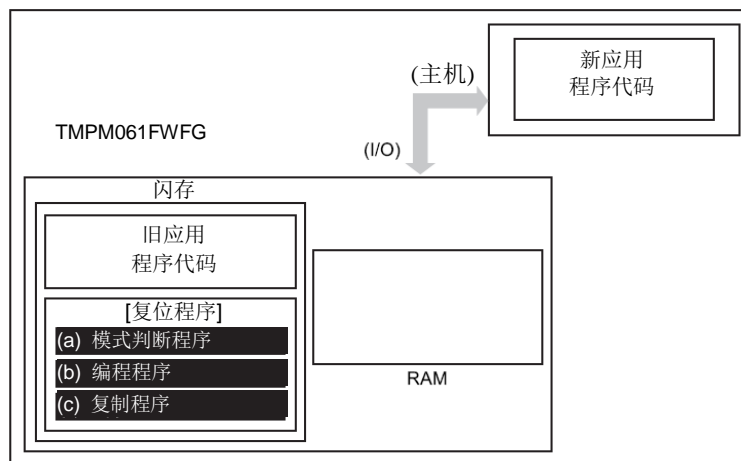
选取两种情况作为示例，如存储于闪存 (1-A)，传送自外部装置 (1-B)的重编程程序方法，下节对该步骤予以阐述。有关闪存程序/擦除的详细，请参照“20.2 闪存的详细”。

20.4.1 (1-A)编程程序存储于闪存的步骤

20.4.1.1 步骤-1

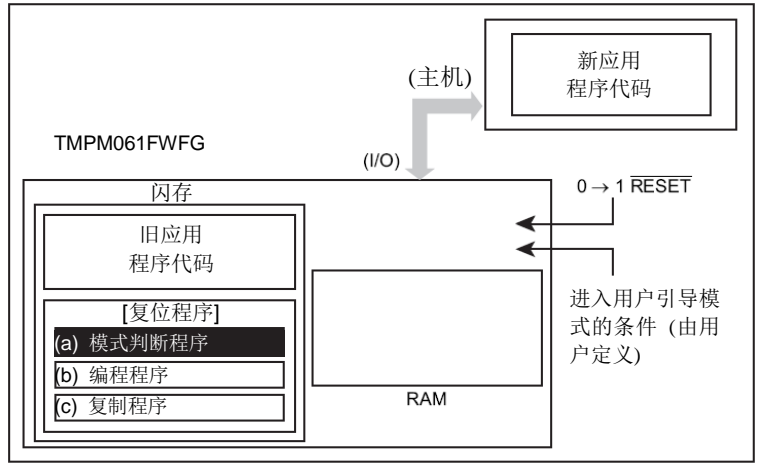
用户确定进入用户启动模式和拟用于传送数据的 I/O 总线的条件(如引脚状态)。然后创建适当的电路设计和程序。在将本设备安装到印刷电路板上之前，需利用闪存编写程序等编程设备将以下三个例行程序写入任意的闪存存储块中。

- (a) 模式确定程序： 确定是否切换至用户启动模式的程序
- (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： 用以将(a)所述的数据复制到内置 RAM 或外部存储器设备中的程序



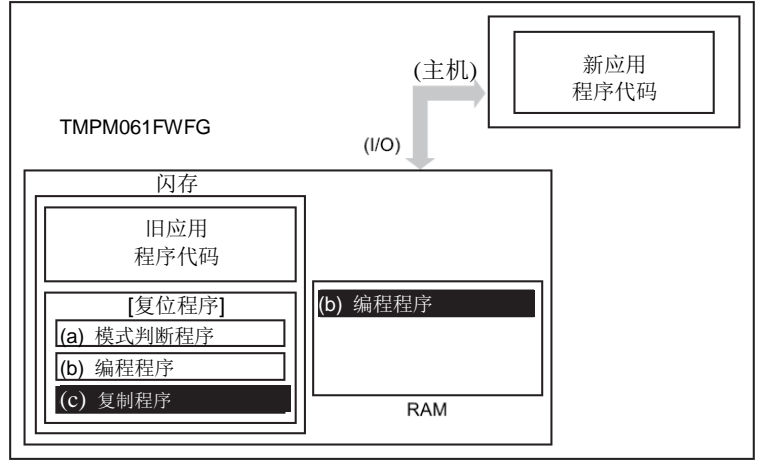
20.4.1.2 步骤-2

本节阐述了存储于复位程序的编程程序的情况。首先，该复位程序确定进入用户启动模式。若模式切换条件符合，装置进入用户启动模式对数据进行重编程。



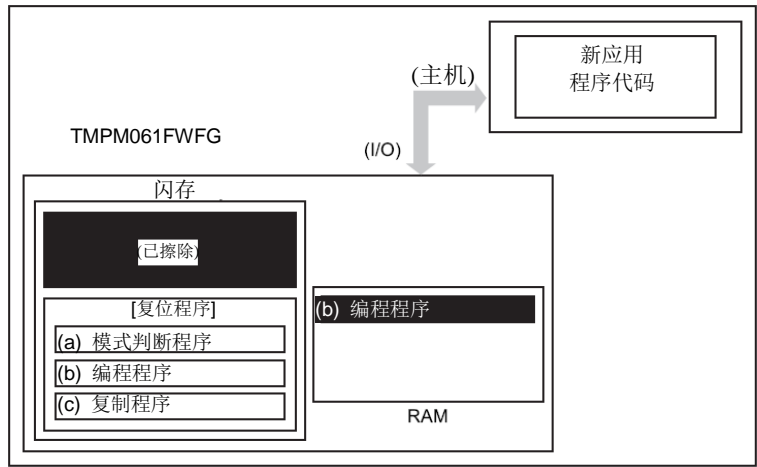
20.4.1.3 步骤-3

一旦该设备进入用户启动模式，就会执行该复制程序(C)，从主机控制器将闪存编程程序(b)下载到内置 RAM。



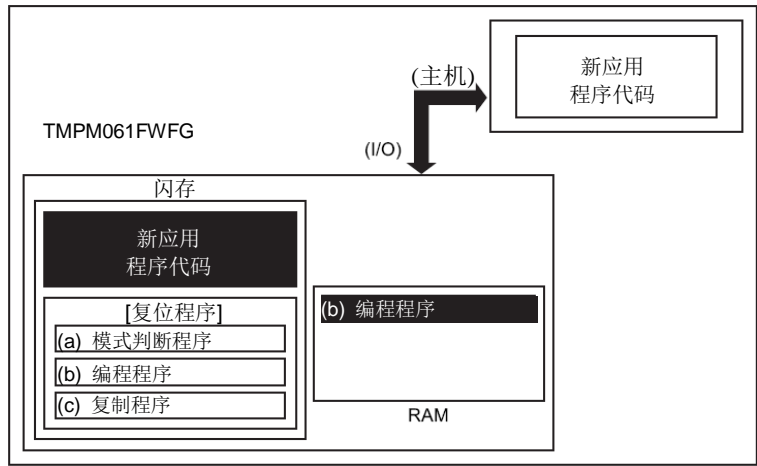
20.4.1.4 步骤-4

跳转至在内置 RAM 中的重编程程序，释放旧应用程序的写入/擦除保护，擦除块单元中的闪存。



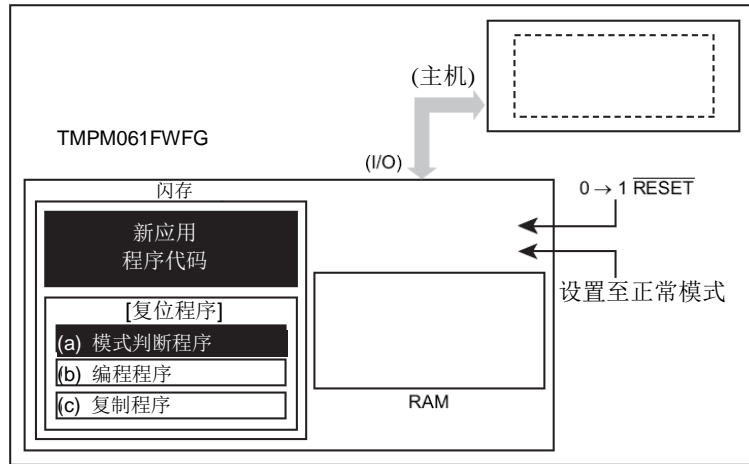
20.4.1.5 步骤-5

继续执行闪存编程程序，从主机控制器下载新的程序数据，并将其编入已被擦除的闪存存储块。当编程完成时，必须在用户程序区中设置闪存存储块的写入/擦除保护。



20.4.1.6 步骤-6

设 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。



20.4.2 (1-B)编程程序自外部主机进行传送的步骤

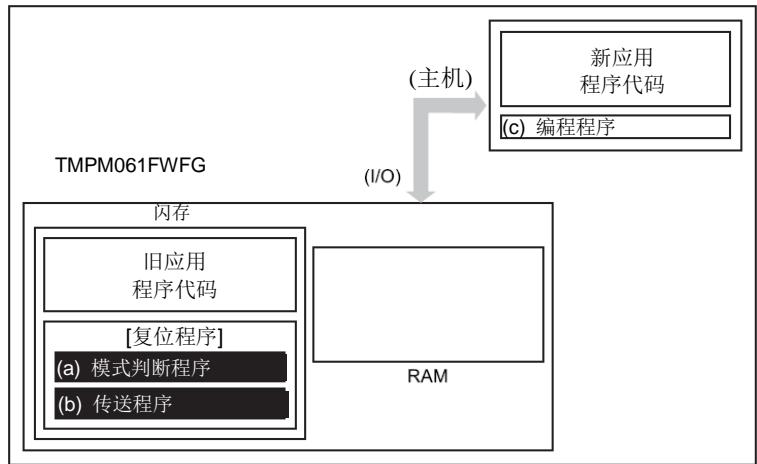
20.4.2.1 步骤-1

用户确定进入用户启动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。在将设备安装到印刷电路板上之前，需用闪存编写程序等编程设备将以下两个程序写入任意闪存存储块。

- (a) 模式确定例行程序： 确定是否切换至重编程工作的程序
- (b) 传送程序： 自外部装置获取重编程程序的程序。

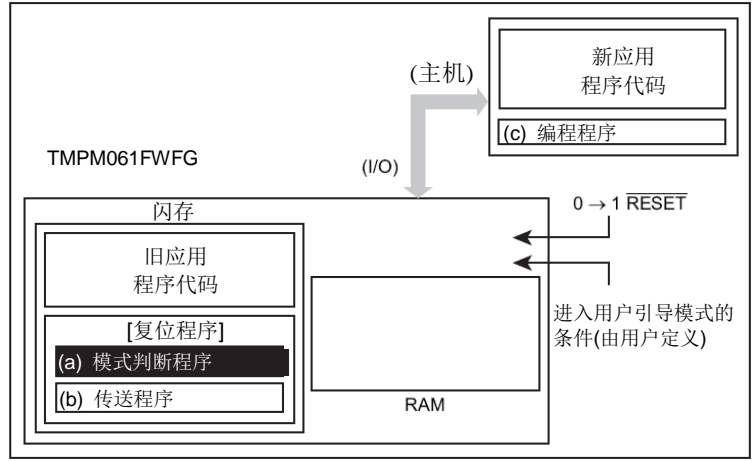
另外，准备须存储于主机控制器，如下所示的重编程程序。

- (c) 重编程程序： 对数据进行重编程的程序



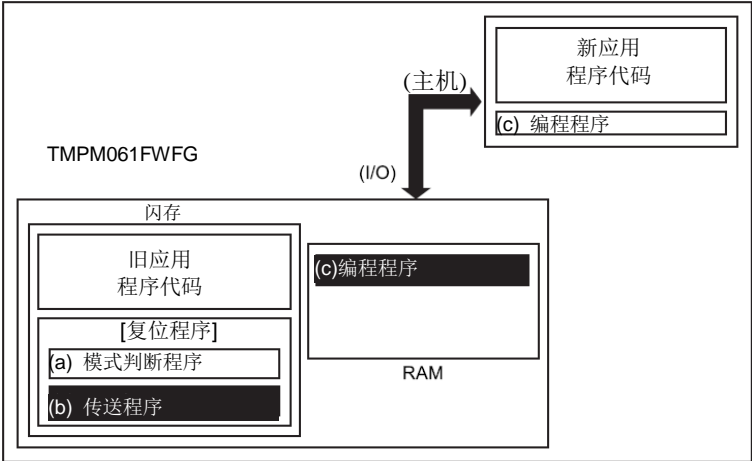
20.4.2.2 步骤-2

本节阐述了存储于复位程序的编程程序的情况。首先，该复位程序确定进入用户启动模式。若模式切换条件符合，装置进入用户启动模式对数据进行重编程。



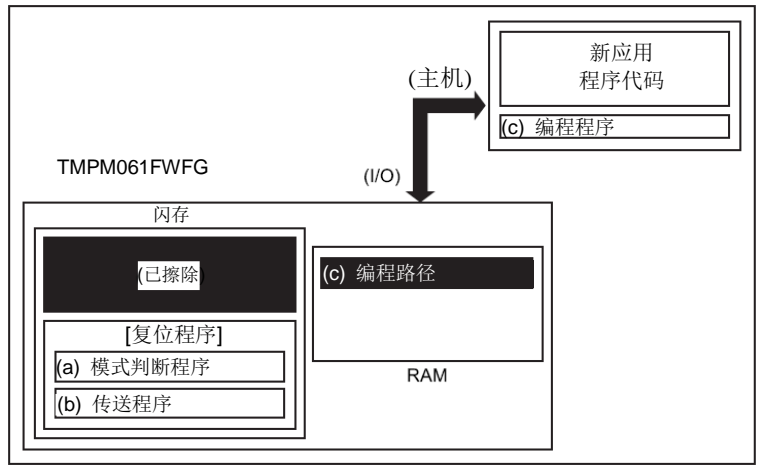
20.4.2.3 步骤-3

一旦该设备进入用户启动模式，就会执行该传送程序(b)从主机控制器将该编程程序(c)下载到内置RAM。



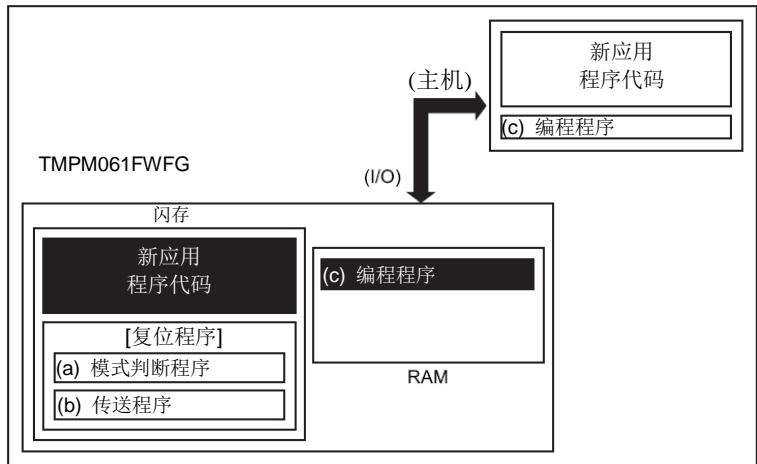
20.4.2.4 步骤-4

跳转至在内置 RAM 中的重编程程序，释放旧应用程序的写入/擦除保护，擦除块单元中的闪存。



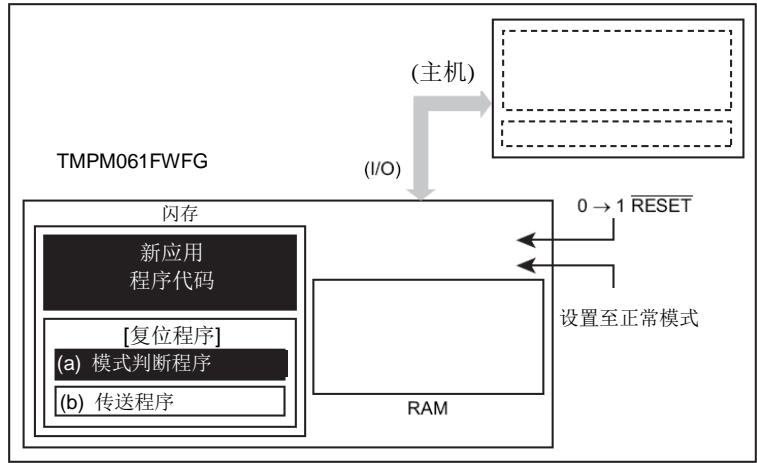
20.4.2.5 步骤-5

继续执行该闪存编程程序(c)，从主机控制器下载新的程序数据，并将其编入已被擦除的闪存存储块。当编程完成时，必须在用户程序区中设置闪存存储块的写入/擦除保护。



20.4.2.6 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。



21. 调试接口

21.1 规范概述

TMPM061FWFG包含可用作调试工具接口的串行电线调试端口(SW-DP)单元。SW-DP支持串行电线调试端口(SWCLK, SWDIO)。

有关SW-DP的详细，请参看“Cortex-M3技术参考手册”。

21.2 引脚功能

调试接口引脚也能用作通用端口。

在复位，各调试接口引脚可配置为调试端口功能引脚。在将各调试接口引脚用于其它功能之前，必须改变设置。

表 21-1 给出了复位之后的调试接口引脚设置。

表 21-1 复位之后的调试接口引脚及相关端口设置

调试 功能	复位后相关端口的设置值				
	功能 (PxFR)	输入 (PxIE)	输出 (PxCR)	上拉 (PxPUP)	下拉 (PxPDN)
SWCLK	1	1	0	0	1
SWDIO	1	1	1	1	0

21.3 复位向量中止

TMPM061FWFG被禁止在RESET引脚所导致的复位有效期间用调试工具进行发送。在用复位向量进行中止设置时，需在复位之后设置以下步骤：从调试工具设置拐点，然后设置应用程序中断和复位控制寄存器的<SYSRESETREQ>位，以再次复位。

21.4 调试接口引脚用于通用端口用途时的注意事项

如果在复位之后通过某个用户程序将某个调试接口引脚设置为通用端口，则调试工具无法控制MCU。为将某个调试工具再次连接到MCU，用户必须准备好其中可将通用端口改为调试接口设置的某种机构。

21.5 调试启用引脚

TMPM061FWFG准备好调试启用引脚(\overline{DBGEN})。这些调试启用引脚用于实现与调试工具之间的可靠通讯。

各调试启用引脚在复位期间均有效。如果在复位信号的上升沿对“低”信号进行采样，则调试启用状态已完成设置，且其功能已被固定为接口。该状态会保持到下一次引脚复位发生时为止。

在复位之后，调试工具可与处理器内核通讯。当用户程序在复位之后不久更改调试接口的设置时，这种现象有效。

以下寄存器可监控调试情况。

基址 = 0x41FF_F000

寄存器名称		地址 (基+)
调试启用监视器寄存器	FCDBGEN	0x005C

	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	-	-	-	-	-	-	-	EN0
复位后	1	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-1	-	R	读作 "0"。
0	EN0	R	可监控调试接口引脚。 0: 可改变SWCLK0 和SWDIO0 的功能。 1: 无法改变SWCLK0 和SWDIO0 的功能。 在复位之后，如果 $\overline{\text{DBGEN0}}$ 引脚为“低”，则设置“1”。

21.6 在停止模式时的外设功能

当Cortex-M0内核进入停止模式时，看门狗计时器(WDT)自动停止工作。可选择让 16-位计时器(TMRB和TMR16A)继续或停止计数。其它外部设备则继续运行。

21.7 关于调试工具连接

与调试工具的连接，见制造商的建议。

调试接口引脚包含上拉电阻器和下拉电阻器。当调试接口引脚与外部上拉或下拉连接时，请注意输入电平。

22. 电气特性

22.1 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
		AVDD3	-0.3 ~ 3.9	
		RVDD3	-0.3 ~ 3.9	
		DSRVDD3	-0.3 ~ 3.9	
		SRVDD3	-0.3 ~ 3.9	
		VLC	-0.3 ~ 3.9	
电容器电压		COUT	-0.3 ~ 3.0	V
输入电压	以下引脚除外	V_{IN}	-0.3 ~ DVDD3 + 0.3	V
	PI5, PI6, PJ2		-0.3 ~ 5.5	
	DAIN0+, DAIN0-, DAIN1+, DAIN1-, DAIN2+, DAIN3-		-0.375 ~ DSRVDD3 + 0.3	
低电平	每引脚	I_{OL}	5	mA
输出电流	合计	ΣI_{OL}	50	
	高电平	每引脚	I_{OH}	
输出电流	合计	ΣI_{OH}	50	
功耗 (Ta = 85 °C)		PD	600	mW
焊接温度(10 s)		T_{SOLDER}	260	°C
贮存温度		T_{STG}	-40 ~ 125	°C
运行温度	闪存W/E时除外	T_{OPR}	-40 ~ 85	°C
	在闪存写入/擦除期间		0 ~ 70	

注：绝对最大额定值是运行和环境条件的极限值，即使在最坏的情况下也不应超出这些绝对最大额定值。设备制造商所采用的设计，应能确保不会超出电流，电压，功耗，温度等的任何绝对最大额定值。如果让设备接触上列条件以外的其它条件，则可能导致永久性损伤设备或影响设备可靠性，从而加大IC 爆炸和/或燃烧造成人身伤害的潜在风险。

22.2 DC 电气特性(1/3)

Ta = -40 ~ 85 °C

参数		符号	条件	最小值	典型值(注1)	最大值	单位
电源电压	DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = RVDD3 = VLC (注 2) DVSS = AVSS = RVSS = DSRVSS = 0 V	DVDD3 AVDD3 RVDD3 DSRVDD3 VLC	f _{osc} = 8 ~ 16 MHz f _{sys} = 1 ~ 16 MHz f _s = 30 ~ 34 kHz	1.8	-	3.6	V
		SRVDD		2.9	-	3.6	V
低电平输入电压	PF0, PF1	V _{IL1}	1.8 V ≤ AVDD3 ≤ 3.6 V	-0.3	-	0.2 AVDD3	V
	PA0 ~ 7, PB0 ~ 7, PC0 ~ 7, PD0 ~ 7, PE0 ~ 7, PF0 ~ 1, PG0, PH0 ~ 5, PI0 ~ 6, PJ0 ~ 5, PK0 ~ 1 RESET, MODE	V _{IL2}	1.8 V ≤ DVDD3A ≤ 3.6 V			0.2 DVDD3	
高电平输入电压	PF0, PF1	V _{IH1}	1.8 V ≤ DVDD3B ≤ 3.6 V	0.8 AVDD3	-	AVDD3 + 0.3	V
	PA0 ~ 7, PB0 ~ 7, PC0 ~ 7, PD0 ~ 7, PE0 ~ 7, PF0 ~ 1, PG0, PH0 ~ 5, PI0 ~ 4, PJ0 ~ 1, PJ3 ~ 5, PK0 ~ 1 RESET, MODE	V _{IH2}	1.8 V ≤ DVDD3A ≤ 3.6 V	0.8 DVDD3		DVDD3 + 0.3	
	PI5, PI6, PJ2	V _{IH3}		0.9 DVDD3		5.5	
低电平输出电压	V _{OL}	I _{OL} = 2 mA	DVDD3 ≥ 2.7 V AVDD3 ≥ 2.7 V	-	-	0.4	V
			I _{OL} = 0.5 mA			DVDD3 ≥ 1.8 V AVDD3 ≥ 1.8 V	
高电平输出电压	V _{OH}	I _{OH} = -2 mA	DVDD3 ≥ 2.7 V	2.4	-	DVDD3	V
			AVDD3 ≥ 2.7 V			AVDD3	
		I _{OH} = -0.5 mA	DVDD3 ≥ 1.8 V	1.5		DVDD3	
			AVDD3 ≥ 1.8 V			AVDD3	
输入泄放电流	I _{L1}	0.0 ≤ V _{IN} ≤ DVDD3 0.0 ≤ V _{IN} ≤ AVDD3	-	0.02	±5	μA	
输出泄放电流	I _{LO}	0.2 ≤ V _{IN} ≤ DVDD3 - 0.2 0.2 ≤ V _{IN} ≤ AVDD3 - 0.2	-	0.05	±10		
复位上拉电阻器	RRST	DVDD3 = 1.8 V ~ 3.6 V	-	50	150	kΩ	
可编程上拉/下拉电阻器	PKH	DVDD3 = 1.8 V ~ 3.6 V AVDD3 = 1.8 V ~ 3.6 V	-	50	150	kΩ	
引脚电容(电源引脚除外)	C _{IO}	f _c = 1 MHz	-	-	10	pF	

注 1: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = DSRVDD = SRVDD = VLC = 3.3 V, 但另有说明的情形除外。

注 2: DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD和VLC的电源电压必须相同。

22.3 DC 电气特性(2/3)

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V ~ 3.6 V, Ta = -40 ~ 85 °C

参数	符号	条件	最小值	典型值	最大值	单位
低电平 输出电流	I_{OL}	每引脚	-	-	2	mA
	ΣI_{OL}	合计, 所有端口	-	-	35	mA
高电平 输出电流	I_{OH}	每引脚	-	-	-2	mA
	ΣI_{OH}	合计, 所有端口	-	-	-35	mA

注: DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD和VLC的电源电压必须相同。

22.4 DC 电气特性(3/3)

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V ~ 3.6 V, Ta = -40 ~ 85 °C

参数	符号	条件	最小值	典型值 (注 1)	最大值	单位
NORMAL (注 2)	I_{DD}	fsys = 16 MHz	-	6.5	9.45	mA
		fsys = 10 MHz	-	4	7.45	
IDLE (注 3)		fsys = 16 MHz	-	2.2	3.4	
		fsys = 10 MHz	-	1.5	2.6	
SLOW		fs = 32.768 kHz	-	300	1000	μ A
SLEEP			-	15	350	
STOP	-		-	14	300	

注 1: Ta = 25 °C, DVDD3 = RVDD3 = AVDD3 = DSRVDD3 = SRVDD = VLC = 3.3 V, 但另有说明的情形除外。

注 2: I_{DD} NORMAL: 用测试程序版本2.1运行于FLASH的测量。

所有功能均工作, 但不包括AD转换器的参考电压, $\Delta\Sigma$ AD转换器, 温度传感器, LCD和I/O端口的泄放电阻。

注 3: I_{DD} IDLE: 测量时所有功能停止。包括流经DVDD3, AVDD3, RVDD3, DSRVDD3, SRVDD和VLC的电流。

22.5 10-位 ADC 电气特性

 $DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8\text{ V} \sim 3.6\text{ V}$
 $AVSS = DVSS = 0\text{ V}, Ta = -40 \sim 85\text{ }^\circ\text{C}$

参数	符号	条件	最小值	典型值	最大值	单位
模拟参考电压(+)	AVREFH	-	1.8	2.7	3.6	V
模拟输入电压	VAIN	-	AVSS	-	AVREFH	V
模拟参考电压的 供电电流	AD转换	IREF DVSS = AVSS	-	0.4	0.55	mA
	非AD转换		-	0.05	10	μA
INL误差	-	AVDD = 1.8 V AVSS = 0.0 V VREFH = 1.8 V	-	± 3	± 4	LSB
DNL误差			-	± 3	± 4	
零刻度误差			-	± 3	± 4	
满刻度误差			-	± 3	± 4	
总误差			-	± 3	± 4	
INL误差	-	AVDD = 3.0 V AVSS = 0.0 V VREFH = 3.0 V	-	± 2	± 3	
DNL误差			-	± 2	± 3	
零刻度误差			-	± 2	± 3	
满刻度误差			-	± 2	± 3	
总误差			-	± 2	± 3	

注 1: $1\text{LSB} = (\text{AVREFH} - \text{AVREFL})/1024\text{ [V]}$

注 2: 外设功能被禁用。

22.6 24-位 $\Delta\Sigma$ ADC 电气特性

 $DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 2.9\text{ V} \sim 3.6\text{ V}, DSRVSS = DVSS = 0\text{ V}, Ta = -40 \sim 85\text{ }^\circ\text{C}$

参数	符号	条件	最小值	典型值	最大值	单位
模拟参考电压(+)	dsVREFIN	-	-	2.75	-	V
AMP的功耗电流 (3单元共用)	dsAMPICC	-	-	0.5	1.0	mA
转换的功耗电流 (每1单元)	dsADICC	-	-	1.4	2.5	mA
SNDR	dsSNDR	GAIN = x1 转换时间 $\geq 330\text{ }\mu\text{s}$	-	84	-	dB
输入范围	AINP AINN		-0.375	-	0.375	V
输入电阻抗	dsinp		-	49.5 (注 2)	-	kohm
转换时间	Tconvds	-	165	-	-	μs

注 1: 外设功能被禁用。

注 2: AINN=0 V时的值。



22.7 温度传感器特性

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V ~ 3.6 V, Ta = -40 ~ 85 °C

参数	符号	条件	最小值	典型值	最大值	单位
参考电压电路的功耗电流(注 1)	BGR _{ICC}	-	-	0.2	0.5	mA
温度传感器的功耗电流	TEMP _{ICC}	-	-	0.7	1.0	mA
相对误差(注 2)	-	Ta = -20 ~ 85 °C	-	-	±3	°C
		Ta = -40 ~ 85 °C	-	-	±5	

注 1: $\Delta\Sigma$ 模拟/数字转换器可共享该参考电压电路。

注 2: 其均为单个温度传感器的设计保证值, 且都是在 30 °C 和 60 °C 时通过基于实测值的直线接近法获取的。

22.8 LCD 特性

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V ~ 3.6 V, Ta = -40 ~ 85 °C

参数	符号	条件	最小值	典型值	最大值	单位
LCD驱动器的电源	VLC	LCD驱动器已被启用	2.2	-	3.6	V
		LCD驱动器未被启用	1.8	-	3.6	
内部泄放电阻	RH1	-	-	500	-	kohm
	RH2		-	200	-	
	RL		-	20	-	

22.9 AC 电气特性

22.9.1 AC 测量条件

除非另外注明，本节 AC 特性数据在以下条件下进行测量

- 输出电平：高 = $0.8 \times DVDD3A$ ， $0.8 \times DVDD3$ ，低 = $0.2 \times DVDD3A$ ， $0.2 \times DVDD3B$
- 输入电平：参照低电平输入电压和高电平输入电压 "DC 电气特性"。
- 负载容量：CL = 30 pF

22.9.2 串行通道(SIO/UART)

22.9.2.1 I/O 接口模式

下表中，字母 x 代表 SIO 运行时钟周期时间，和 fsys 周期时间相同。因时钟齿轮功能编程而异。

(1) SCLK 输入模式

[输入]

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK 时钟高宽度 (输入)	t_{SCH}	4x	-	250	-	ns
SCLK 时钟低宽度 (输入)	t_{SCL}	4x	-	250	-	
SCLK 周期	t_{SCY}	$t_{SCH} + t_{SCL}$	-	500	-	
有效数据输入 ← SCLK上升或下降 (注)	t_{SRD}	30	-	30	-	
SCLK上升 → 输入数据保持或下降(注)	t_{HSR}	x + 30	-	92.5	-	

[输出]

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK 时钟高宽度 (输入)	t_{SCH}	4x	-	250	-	ns
SCLK 时钟低宽度 (输入)	t_{SCL}	4x	-	250	-	
SCLK 周期	t_{SCY}	$t_{SCH} + t_{SCL}$	-	500	-	
输出数据 ← SCLK上升或下降(注)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	-	17.5	-	
SCLK上升 → 输出数据保持或下降(注)	t_{OHS}	$t_{SCY}/2$	-	250	-	

注：SCLK上升或下降：相对于SCLK的已编程激活沿测得。

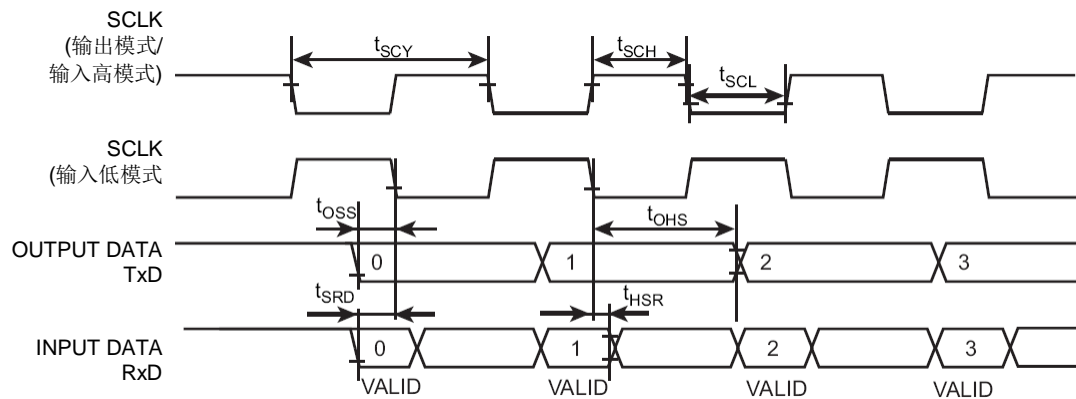
(2) SCLK 输出模式

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 2.7 ~ 3.6 V

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK 周期 (可编程)	t_{SCY}	4x	-	250	-	ns
输出数据 ← SCLK 上升	t_{OSS}	$t_{SCY}/2 - 30$	-	95	-	
SCLK上升→输出数据保持	t_{OHS}	$t_{SCY}/2 - 30$	-	95	-	
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 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCLK 周期 (可编程)	t_{SCY}	4x	-	250	-	ns
输出数据 ← SCLK 上升	t_{OSS}	$t_{SCY}/2 - 30$	-	95	-	
SCLK上升→输出数据保持	t_{OHS}	$t_{SCY}/2 - 70$	-	55	-	
有效数据输入 ← SCLK上升	t_{SRD}	90	-	90	-	
SCLK上升→ 输入数据保持	t_{HSR}	0	-	0	-	



22.9.3 串行总线接口(I2C/SIO)

22.9.3.1 I2C 模式

下表中, 字母 x 代表 I2C 运行时钟周期时间, 和 fsys 周期时间相同。因时钟齿轮功能编程而异。

n 表示编程入 SBIxCR 内 SCK 字段的 n 数值 (SCL 输出频率选择)。

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	t _{SCL}	0	-	0	100	0	400	kHz
START条件保持时间	t _{HD; STA}	-	-	4.0	-	0.6	-	μs
SCL低宽度(输入) (注 1)	t _{LOW}	-	-	4.7	-	1.3	-	μs
SCL高宽度(输入) (注 2)	t _{HIGH}	-	-	4.0	-	0.6	-	μs
重复START 条件的设置时间	t _{SU; STA}	(注 5)	-	4.7	-	0.6	-	μs
数据保持时间(输入) (注 3, 4)	t _{HD; DAT}	-	-	0.0	-	0.0	-	μs
数据设置时间	t _{SU; DAT}	-	-	250	-	100	-	ns
STOP条件的设置时间	t _{SU; STO}	-	-	4.0	-	0.6	-	μs
停止条件和启动条件之间的总线空闲时间	t _{BUF}	(注 5)	-	4.7	-	1.3	-	μs

注 1: SCL时钟低宽度 (输出) = $(2^{n-1} + 58)/x$

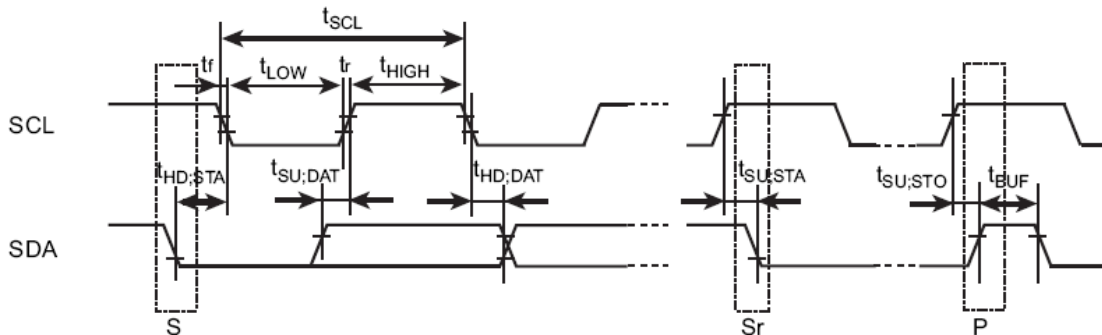
注 2: SCL时钟高宽度(输出) = $(2^{n-1} + 14)/x$ 就I2C-总线规格而言, 标准模式的最高速度为 100 kHz, 快速模式为 400 khz。内部SCL频率设置应与上述所示注 1 & 注 2相符。

注 3: 输出数据保持时间等于内部SCL的 4 倍4x。

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, 让SDA信号桥接SCL下降沿的未定义区域。但是, 该SBI无法满足本要求。此外, SCL的输出缓冲器不具备下降沿斜率控制装置; 因此, 该设备制造商所采用的设计应能确保满足表中所列的输入数据保持时间, 包括SCL和SDA线的tr/tf。

注 5: 视软件而定

注 6: 飞利浦I2C-总线规格表明, 如果快速模式设备的电源被断开, 则SDA和SCL I/O引脚必须为漂浮式, 从而不妨碍该总线。不过, 该SBI无法满足本要求。



S: 启动条件

Sr: 重复启动条件

P: 停止条件

22.9.3.2 时钟同步 8-位SIO模式

下表中，字母 x 代表 I2C 运行时钟周期时间，和 fsys 周期时间相同。因时钟齿轮功能编程而异。

(1) SCK 输入模式(以下电气规格适用于占空比为 50%的 SCK 信号)

[输入]

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCK时钟高宽度 (输入)	t_{SCH}	4x	-	250	-	ns
SCK时钟低宽度 (输入)	t_{SCL}	4x	-	250	-	
SCK 周期	t_{SCY}	$t_{SCH} + t_{SCL}$	-	500	-	
有效数据输入 ← SCK上升	t_{SRD}	30 - x	-	-32.5	-	
SCK 上升 → 输入数据保持	t_{HSR}	2x + 30	-	155	-	

[输出]

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCK时钟高宽度 (输入)	t_{SCH}	4x	-	250	-	ns
SCK时钟低宽度 (输入)	t_{SCL}	4x	-	250	-	
SCK 周期	t_{SCY}	$t_{SCH} + t_{SCL}$	-	500	-	
输出数据 ← SCK上升	t_{OSS}	$t_{SCY}/2 - 3x - 45$	-	17.5	-	
SCK 上升 → 输出数据保持	t_{OHS}	$t_{SCY}/2 + x$	-	312.5	-	

(2) SCK 输出模式(以下电气特性适用于 50% 占空比周期的 SCK 信号。)

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 2.7 ~ 3.6 V

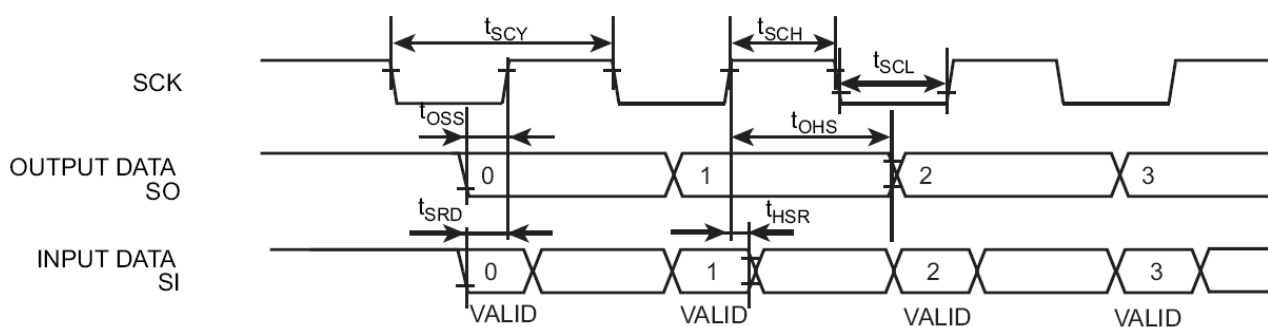
参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
SCK 周期 (可编程)	t_{SCY}	16x (注 1)	-	1000	-	ns
输出数据 ← SCK上升	t_{OSS}	$t_{SCY}/2 - 30$ (注 2)	-	470	-	
SCK 上升 → 输出数据保持	t_{OHS}	$t_{SCY}/2 - 30$	-	470	-	
有效数据输入 ← SCK上升	t_{SRD}	x + 45	-	107.5	-	
SCK 上升 → 输入数据保持	t_{HSR}	0	-	0	-	

DVDD3 = AVDD3 = RVDD3 = DSRVDD3 = SRVDD = VLC = 1.8 V

参数	符号	等式		40 MHz		单位
		最小值	最大值	最小值	最大值	
SCK 周期 (可编程)	t_{SCY}	16x (注 1)	-	1000	-	ns
输出数据 ← SCK上升	t_{OSS}	$t_{SCY}/2 - 30$ (注 2)	-	470	-	
SCK 上升 → 输出数据保持	t_{OHS}	$t_{SCY}/2 - 70$	-	430	-	
有效数据输入 ← SCK上升	t_{SRD}	x + 90	-	152.5	-	
SCK 上升 → 输入数据保持	t_{HSR}	0	-	0	-	

注 1: 自动等待后SCK周期变为 14x。

注 2: 自动等待后的SO数据输出可能会是 $t_{SCY}/2 - x - 20$ 。



22.9.4 事件计数器

下表中，字母 x 代表 TMRB 运行时钟周期时间，和 f_{sys} 周期时间相同。因时钟齿轮功能编程而异。

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
时钟低脉冲宽度	t_{VCKL}	$2x + 100$	-	225	-	ns
时钟高脉冲宽度	t_{VCKH}	$2x + 100$	-	225	-	ns

22.9.5 捕捉

下表中，字母 x 代表 TMRB 运行时钟周期时间，和 f_{sys} 周期时间相同。因时钟齿轮功能编程而异。

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
低脉冲宽度	t_{CPL}	$2x + 100$	-	225	-	ns
高脉冲宽度	t_{CPH}	$2x + 100$	-	225	-	ns

22.9.6 外部中断

下表中，字母 x 代表 f_{sys} 周期时间。

1. 除 STOP 释放中断信号之外

参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
INT0 ~ 3 低电平脉冲宽度	t_{INTAL}	$x + 100$	-	162.5	-	ns
INT0 ~ 3 高电平脉冲宽度	t_{INTAH}	$x + 100$	-	162.5	-	ns

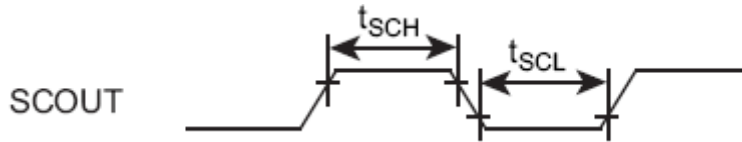
2. STOP 释放中断信号

参数	符号	最小值	最大值	单位
INT0 ~3 低电平脉冲宽度	t_{INTBL}	100	-	ns
INT0 ~ 3 高电平脉冲宽度	t_{INTBH}	100	-	ns

22.9.7 SCOUT 引脚 AC 特性

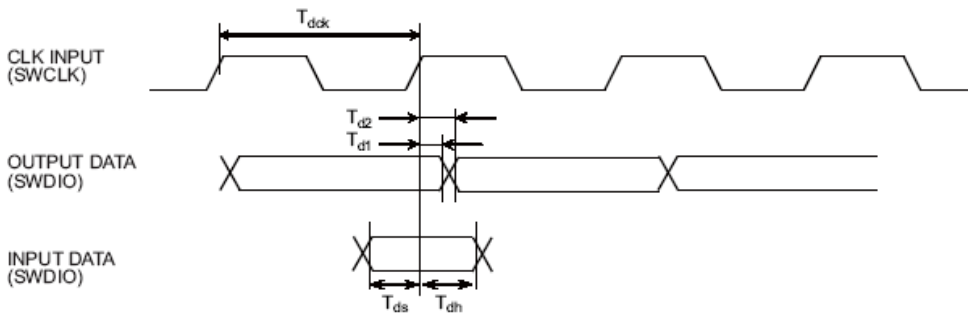
参数	符号	等式		16 MHz		单位
		最小值	最大值	最小值	最大值	
高电平脉冲宽度	t_{SCH}	$0.5T - 5$	-	26.25	-	ns
低电平脉冲宽度	t_{SCL}	$0.5T - 5$	-	26.25	-	ns

注：上表中，字母T代表SCOUT输出时钟的周期时间。



22.9.8 调试通信

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下降 → 输出数据有效	T_{d2}	-	30	
输入数据有效 → CLK上升	T_{ds}	20	-	
CLK 上升 → 输入数据保持	T_{dh}	15	-	



22.9.9 闪存特性

参数	条件	最小值	典型值	最大值	单位
闪存重新写入保证	DVDD3 = RVDD3 = 2.7 V ~ 3.6 V, Ta = 0 ~ 70 °C	-	-	100	次数

22.9.10 芯片上振荡器

参数	符号	最小值	典型值	最大值	单位
振荡频率	IHOSC	9	10	11	MHz

22.10 建议振荡电路

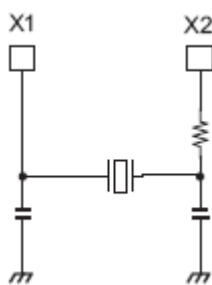


图 22-1 高频振荡连接

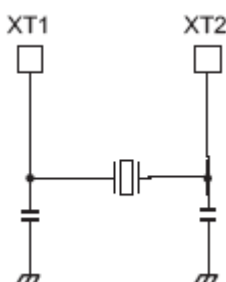


图 22-2 低频振荡连接

注：为获得稳定的振荡，振荡器的负荷量和位置必须正确配置。

由于这些因素受基板模式影响很大，请采用你所使用的基板评估振荡稳定性。

22.10.1 陶瓷振荡器

TMPM061FWFG 建议采用株式会社村田制作所出品的陶瓷振荡器。详情请参照以下 URL。

<http://www.murata.co.jp>

22.10.2 晶体振荡器

TMPM061FWFG 建议采用精工仪器公司出品的晶体振荡器。详情请参照以下网址。

<http://www.sii.co.jp>

23. 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。

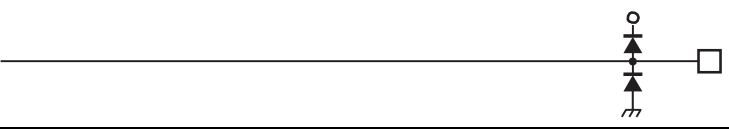

输入保护电阻有几十 Ω 到数百 Ω 不等。阻尼电阻器X2和XT2均标注有典型值。

注：图中未标值的各电阻器则为输入保护电阻器。



23.1 PORT 引脚

<p>PE0 ~ 5 (端口类型 FT1) PE6, 7 (端口类型 FT2) PF1 (端口类型 FT4) PG0 PH0 ~ 5 PIO ~ 4 PJ0 ~ 1, 3 ~ 4 PK0 (端口类型 FT4) PK1 (端口类型 FT1)</p>	
<p>PI5, PI6, PJ2</p>	<p>注: 只有在输入启用时, 这些引脚才能承受 5 V 输入。</p>
<p>AIN(输入) SEG(输出) PA0 ~ 7 PB0 ~ 7 PC0 ~ 7 PD0 ~ 7 PE0 ~ 7 (端口类型 FT10) PF0 PF1 (端口类型 FT5) PK0, 1 (端口类型 FT5)</p>	
<p>PJ5</p>	

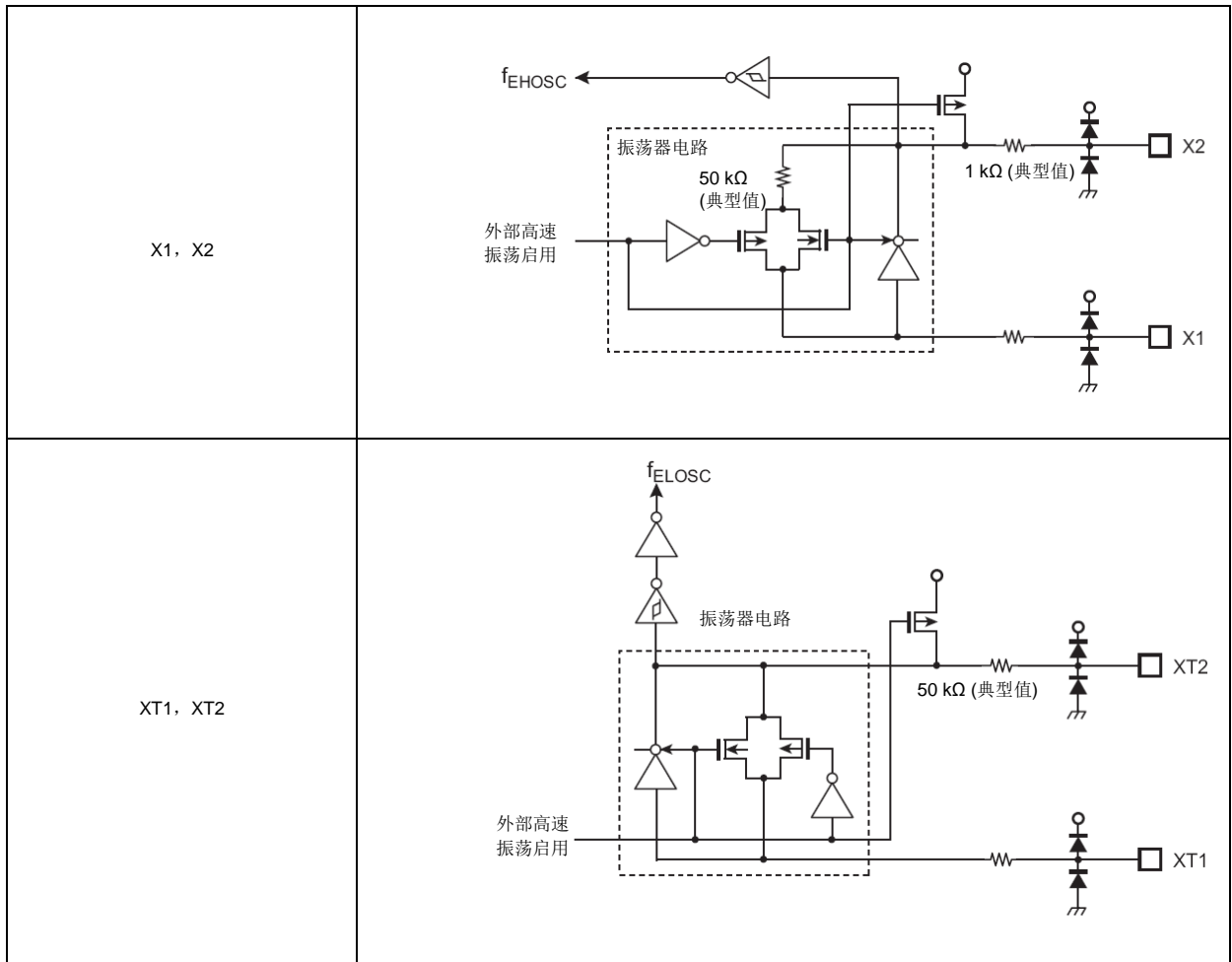
23.2 模拟引脚

COM0, COM1, COM2, COM3	 <p style="text-align: right;">输出端口</p>
DAIN0+, DAIN0-, DAIN1+, DAIN1-, DAIN2+, DAIN2-	 <p style="text-align: right;">输入端口</p>

23.3 控制引脚

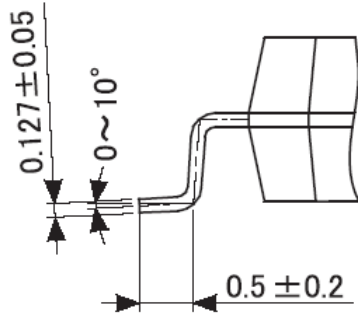
MODE	 <p style="text-align: right;">输入端口</p>
模式引脚必须接GND。	
RESET	 <p style="text-align: right;">输入端口</p>

23.4 时钟引脚



引脚细节

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 intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document.
Product is neither intended nor warranted for use in equipment 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 or serious public impact ("Unintended Use"). 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. Do not use Product for Unintended Use unless specifically permitted in this document.
- 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 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.

