

译文

TMPM375FSDMG

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：“TMPM375FSDMG” 2014-11-26

翻译日: 2015-03-04

译文

TOSHIBA

32位RISC微控制器
TX03系列

TMPM375FSDMG

东芝公司

半导体&存储产品公司

译文

ARM Cortex以及Thumb为ARM Limited(或其子公司)在EU或其它各地区注册的商标。版权所有。



东芝MCU使用一般注意事项

本页阐明了东芝MCU使用的一般注意事项。

注意当一般注意事项与本文件主体说明之间有差异时，应以文件主体中的说明为准。

1. MCU上电时的操作

上电时，MCU的内部状态不稳定。因此，引脚的状态直至复位操作完成才确定。

由外部复位引脚执行复位时，使用复位引脚的MCU引脚直至外部引脚复位完成后才得以确定。

同样，当由内部上电复位执行复位时，使用内部上电复位的MCU引脚直至供电电压达到上电复位有效时的电压才得以确定。

2. 不用的引脚

MCU不用的输入/输出端口禁止使用。这些引脚为高阻抗引脚。一般情况下，如MCU在高阻抗引脚保持断开时操作，则在内部LSI中，可能因外部噪声影响而产生的感应电压造成静电破坏或闭锁。

东芝建议，所有不用的引脚应通过电阻器连接到电源引脚或接地引脚。

3. 时钟振荡稳定性

复位状态必须在时钟振荡稳定后解除。当时钟在程序进行中转换到另一个时钟时，应等待，直至时钟稳定下来。

简介：关于本规范中SFR(特殊功能寄存器)说明的注释

SFR(特殊功能寄存器)是一种外围电路控制寄存器(IP)。

IP的SFR地址见存储器映射章节，SFR详见各IP章节。

本规范中采用的SFR的定义遵照下述规则。

a. 各IP SFR表举例

- IP各章中的SFR表提供有寄存器名称，地址以及简要说明。
- 所有寄存器均具有一个32位唯一地址，并且这些寄存器的地址除了一些异常情况之外，均规定如下："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寄存器地址离地址0x0000_0004(基址(0x00000000)+唯一地址(0x0004))32位宽。

注) 以上寄存器为说明性举例，并非实例。在该微控制器中，该寄存器并不存在。

b. SFR(寄存器)

- 每个寄存器基本上由一个32位寄存器组成(但有异常情况)。
- 每个寄存器的说明介绍了位，比特符号，复位后的初始值以及各个功能。

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	型号	功能
31-10	□	R	可读取"0"。
9-7	MODE[2:0]	R/W	操作模式设置 000:采样模式 0 001:采样模式 1 010:采样模式 2 011:采样模式 3 非上述设置:保留
6-0	TDATA[6:0]	W	传输的数据

注) 分为如下三类型。

R/W	READ WRITE
R	READ
W	WRITE

c. 数据说明

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. 寄存器说明

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第 2 位 ~ 第 0 位(3 位宽)。
- 寄存器名称 [位]
示例: SAMCR[9:7] = "000"
表示寄存器SAMCR的第 9 位 ~ 第 7 位(32 位宽)。

译文

修订记录

日期	修订版	备注
2013/6/4	1	第一次发布
2013/8/30	2	修订内容
2014/11/26	3	修订内容

译文

目录

简介：关于本规范中SFR(特殊功能寄存器)说明的注释

TMPM375FSDMG	1
1.1 特点	1
1.2 方块图	4
1.3 引脚分配(顶视图)	5
1.4 引脚名称与功能	6
1.4.1 按端口分类	6
1.5 引脚编号与电源引脚	8
2. 处理器内核	9
2.1 处理器内核上的信息	9
2.2 可配置选项	9
2.3 异常/中断	10
2.3.1 中断输入的数目	10
2.3.2 优先级中断位的数目	10
2.3.3 SysTick	10
2.3.4 SYSRESETREQ	10
2.3.5 LOCKUP	10
2.3.6 辅助故障状态寄存器	10
2.4 事件	11
2.5 电源管理	11
2.6 独占通道	11
3. 存储器映象	13
3.1 存储器映象	13
3.1.1 TMPM375FSDMG 内存映射	13
3.2 SFR 区详情	14
4. 复位操作	17
4.1 冷复位	18
4.1.1 由上电复位电路(不用 RESET 引脚)进行复位	18
4.1.2 由 RESET 引脚复位	19
4.2 预热	20
4.2.1 复位时长	20
4.3 复位后	20

5. 时钟/模式控制	21
5.1 特点	21
5.2 寄存器	22
5.2.1 寄存器列表	22
5.2.2 CGSYSCR (系统控制寄存器)	23
5.2.3 CGOSCCR (振荡控制寄存器)	24
5.2.4 CGSTBYCR (待机控制寄存器)	26
5.2.5 CGPLLSEL (PLL 选择寄存器)	27
5.3 时钟控制	28
5.3.1 时钟类型	28
5.3.2 复位之后的初始值	28
5.3.3 时钟系统图	29
5.3.4 时钟乘法电路(PLL)	30
5.3.4.1 稳定时间	30
5.3.4.2 PLL 设置顺序	31
5.3.5 预热功能	32
5.3.6 系统时钟	34
5.3.7 预分频器时钟控制	35
5.4 模式与模式推移	36
5.4.1 模式推移	36
5.5 运行模式	36
5.5.1 NORMAL 模式	36
5.6 低功耗模式	37
5.6.1 IDLE 模式	37
5.6.2 STOP 模式	38
5.6.3 低功耗模式设置	39
5.6.4 各模式下的操作状态	39
5.6.5 释放低功耗模式	40
5.6.6 预热	42
5.6.7 模式转换中的时钟运行情况	43
5.6.7.1 运行模式的转换: NORMAL → STOP → NORMAL	43
6. 内部高速振荡调节功能	45
6.1 结构	45
6.2 寄存器	46
6.2.1 寄存器列表	46
6.2.2 TRMOSCPRO(保护寄存器)	46
6.2.3 TRMOSCEN(启用寄存器)	47
6.2.4 TRMOSCINIT(初始修正值监视寄存器)	48
6.2.5 TRMOSCSET (修正值设置寄存器)	49
6.3 操作说明	50
6.3.1 概要	50
6.3.2 调节范围	50
6.3.3 使用 TMRB 进行内部振荡频率测量	50
7. 异常	51
7.1 概述	51
7.1.1 异常类型	51
7.1.2 处理流程图	52
7.1.2.1 异常请求与检测	53
7.1.2.2 异常处理与转入该中断服务程序(先占)	55
7.1.2.3 执行 ISR	56
7.1.2.4 异常出口	57
7.2 复位异常	58
7.3 非屏蔽中断(NMI)	58

7.4 SysTick	58
7.5 中断	59
7.5.1 中断源.....	59
7.5.1.1 中断路径.....	59
7.5.1.2 生成.....	60
7.5.1.3 传输.....	60
7.5.1.4 使用外部中断引脚时的预防措施.....	60
7.5.1.5 中断源列表.....	61
7.5.1.6 作用电平.....	62
7.5.2 中断处理.....	63
7.5.2.1 流程图.....	63
7.5.2.2 准备工作.....	65
7.5.2.3 由时钟发生器进行检测.....	67
7.5.2.4 由 CPU 进行检测.....	68
7.5.2.5 CPU 处理.....	68
7.5.2.6 中断服务程序(ISR).....	68
7.6 异常/中断相关寄存器	69
7.6.1 寄存器列表.....	69
7.6.2 NVIC 寄存器.....	70
7.6.2.1 SysTick 控制器与状态寄存器.....	70
7.6.2.2 SysTick 重新加载值寄存器.....	71
7.6.2.3 SysTick 修正值寄存器.....	71
7.6.2.4 SysTick 校准值寄存器.....	72
7.6.2.5 中断设置-启用寄存器 1.....	73
7.6.2.6 中断清除-启用寄存器 1.....	74
7.6.2.7 中断设置-悬挂寄存器 1.....	75
7.6.2.8 中断清除挂起寄存器 1.....	76
7.6.2.9 中断优先级寄存器.....	77
7.6.2.10 矢量表位移寄存器.....	78
7.6.2.11 应用中断与复位控制寄存器.....	79
7.6.2.12 系统处理器优先寄存器.....	80
7.6.2.13 系统处理器控制器与状态寄存器.....	81
7.6.3 时钟发生器寄存器.....	83
7.6.3.1 CGIMCGA (CG 中断模式控制寄存器 A).....	83
7.6.3.2 CGICRCG (CG 中断请求清除寄存器).....	85
7.6.3.3 CGNMIFLG (NMI 标志寄存器).....	86
7.6.3.4 CGRSTFLG (复位标志寄存器).....	87
8. 数字噪声滤波电路(DNF)	89
8.1 配置	89
8.2 寄存器	90
8.2.1 寄存器列表.....	90
8.2.1.1 NFCKCR (噪声滤波器控制寄存器).....	91
8.2.1.2 NFENCR (噪声滤波器启用寄存器).....	92
8.3 操作说明	93
8.3.1 配置.....	93
8.3.2 操作.....	93
8.3.3 噪声滤波器可用操作模式.....	93
8.3.4 STOP 模式使用注意事项.....	93
8.3.5 最短噪声滤波时间.....	93
9. 输入/输出端口	95
9.1 端口功能	95
9.1.1 功能列表.....	95
9.1.2 端口寄存器概述.....	96
9.1.3 STOP 模式下端口状态.....	97
9.2 端口功能	98
9.2.1 Port B (PB3 ~ PB6).....	98
9.2.1.1 端口 B 寄存器.....	98
9.2.1.2 PBDATA (端口 B 数据寄存器).....	99
9.2.1.3 PBCR (端口 B 输出控制寄存器).....	99
9.2.1.4 PBFRI (端口 B 功能寄存器 1).....	100

9.2.1.5	PBFR2 (端口 B 功能寄存器 2)	101
9.2.1.6	PBFR3 (端口 B 功能寄存器 3)	102
9.2.1.7	PBFR4 (端口 B 功能寄存器 4)	103
9.2.1.8	PBFR5 (端口 B 功能寄存器 5)	104
9.2.1.9	PBOD (端口 B 开漏控制寄存器)	105
9.2.1.10	PBPUP (端口 B 上拉控制寄存器)	105
9.2.1.11	PBPDN (端口 B 下拉控制寄存器)	106
9.2.1.12	PBIE (端口 B 输入控制寄存器)	107
9.2.2	端口 E (PE0 ~ PE2)	108
9.2.2.1	端口 E 寄存器	108
9.2.2.2	PEDATA (端口 E 数据寄存器)	109
9.2.2.3	PECR (端口 E 输出控制寄存器)	109
9.2.2.4	PEFR1 (端口 E 功能寄存器 1)	110
9.2.2.5	PEFR2 (端口 E 功能寄存器 2)	111
9.2.2.6	PEFR3 (端口 E 功能寄存器 3)	112
9.2.2.7	PEFR4 (端口 E 功能寄存器 4)	113
9.2.2.8	PEFR5 (端口 E 功能寄存器 5)	114
9.2.2.9	PEOD (端口 E 开漏控制寄存器)	115
9.2.2.10	PEPUP (端口 E 上拉控制寄存器)	115
9.2.2.11	PEPDN (端口 E 下拉控制寄存器)	116
9.2.2.12	PEIE (端口 E 输入控制寄存器)	116
9.2.3	端口 F (PF0)	117
9.2.3.1	端口 F 寄存器	117
9.2.3.2	PFDATA (端口 F 数据寄存器)	118
9.2.3.3	PFCR (端口 F 输出控制寄存器)	118
9.2.3.4	PFFR1 (端口 F 功能寄存器 1)	119
9.2.3.5	PFFR2 (端口 F 功能寄存器 2)	120
9.2.3.6	PFFR4 (端口 F 功能寄存器 4)	121
9.2.3.7	PFFR5 (端口 F 功能寄存器 5)	122
9.2.3.8	PFOD (端口 F 开漏控制寄存器)	123
9.2.3.9	PFPUP (端口 F 上拉控制寄存器)	124
9.2.3.10	PFPDN (端口 F 下拉控制寄存器)	124
9.2.3.11	PFIE (端口 F 输入控制寄存器)	125
9.2.4	端口 G (PG0 ~ PG6)	126
9.2.4.1	端口 G 寄存器	126
9.2.4.2	PGDATA (端口 G 数据寄存器)	127
9.2.4.3	PGCR (端口 G 输出控制寄存器)	127
9.2.4.4	PGFR1 (端口 G 功能寄存器 1)	128
9.2.4.5	PGFR2 (端口 G 功能寄存器 2)	129
9.2.4.6	PGOD (端口 G 开漏控制寄存器)	130
9.2.4.7	PGPUP (端口 G 上拉控制寄存器)	130
9.2.4.8	PGPDN (端口 G 下拉控制寄存器)	131
9.2.4.9	PGIE (端口 G 输入控制寄存器)	131
9.2.5	端口 J (PJ6 ~ PJ7)	132
9.2.5.1	端口 J 寄存器	132
9.2.5.2	PJDATA (端口 J 数据寄存器)	133
9.2.5.3	PJCR (端口 J 输出控制寄存器)	133
9.2.5.4	PJOD (端口 J 开漏控制寄存器)	134
9.2.5.5	PJPUP (端口 J 上拉控制寄存器)	134
9.2.5.6	PJPDN (端口 J 下拉控制寄存器)	135
9.2.5.7	PJIE (端口 J 输入控制寄存器)	135
9.2.6	端口 K (PK0 ~ PK1)	136
9.2.6.1	端口 K 寄存器	136
9.2.6.2	PKDATA (端口 K 数据寄存器)	137
9.2.6.3	PKCR (端口 K 输出控制寄存器)	137
9.2.6.4	PKOD (端口 K 开漏控制寄存器)	138
9.2.6.5	PKPUP (端口 K 上拉控制寄存器)	138
9.2.6.6	PKPDN (端口 K 下拉控制寄存器)	139
9.2.6.7	PKIE (端口 K 输入控制寄存器)	139
9.2.7	端口 M (PM0 ~ PM1)	140
9.2.7.1	端口 M 寄存器	140
9.2.7.2	PMDATA (端口 M 数据寄存器)	141
9.2.7.3	PMCR (端口 M 输出控制寄存器)	141
9.2.7.4	PMOD (端口 M 开漏控制寄存器)	142
9.2.7.5	PMPUP (端口 M 上拉控制寄存器)	142
9.2.7.6	PMPDN (端口 M 下拉控制寄存器)	143
9.2.7.7	PMIE (端口 M 输入控制寄存器)	143
9.3	端口方块图	144
9.3.1	端口类型	144
9.3.2	类型 FT 1	145

9.3.3	类型 FT2	146
9.3.4	类型 FT3	147
9.3.5	类型 FT4	148
9.3.6	类型 FT5	149
9.3.7	类型 FT6	150
9.4	附录端口设置列表	151
9.4.1	端口 B 设置	151
9.4.2	端口 E 设置	152
9.4.3	端口 F 设置	152
9.4.4	端口 G 设置	153
9.4.5	端口 J 设置	154
9.4.6	端口 K 设置	154
9.4.7	端口 M 设置	154
<hr/>		
10.16	位定时器 / 事件计数器(TMRB)	155
<hr/>		
10.1	概述	155
10.2	规格差异	156
10.3	配置	157
10.4	寄存器	158
10.4.1	寄存器列表(按通道).....	158
10.4.2	TBxEN(启用寄存器).....	159
10.4.3	TBxRUN(RUN 寄存器)	160
10.4.4	TBxCR(控制寄存器).....	161
10.4.5	TBxMOD(模式寄存器)	162
10.4.6	TBxFFCR(触发电路控制寄存器)	163
10.4.7	TBxST(状态寄存器)	164
10.4.8	TBxIM(中断屏蔽寄存器)	165
10.4.9	TBxUC (递增计数器捕捉寄存器)	166
10.4.10	TBxRG0(计时器寄存器 0)	167
10.4.11	TBxRG1 (计时器寄存器 1)	167
10.4.12	TBxCP0 (捕捉寄存器 0).....	168
10.4.13	TBxCP1 (捕捉寄存器 1).....	168
10.5	各电路的运行说明	169
10.5.1	预分频器	169
10.5.2	升计数器(UC).....	173
10.5.3	计时器寄存器(TBxRG0, TBxRG1)	173
10.5.4	捕获	173
10.5.5	捕获寄存器(TBxCP0, TBxCP1)	174
10.5.6	向上计数器捕获寄存器(TBxUC)	174
10.5.7	比较器(CP0, CP1)	174
10.5.8	计时器触发电路(TBxFF0).....	174
10.5.9	捕获中断(INTCAPx0, INTCAPx1)	174
10.6	各模式的运行说明	175
10.6.1	16 位间隔计时器模式	175
10.6.2	16 位事件计数器模式	175
10.6.3	16 位 PPG(可编程脉冲发生)输出模式	176
10.6.4	外部触发器可编程脉冲发生输出模式(PPG)	178
10.7	基于捕获功能的应用	179
10.7.1	由外部脉冲触发的单次脉冲输出	179
10.7.2	脉冲宽度测量	181
<hr/>		
11	串行通道(SIO/UART)	183
<hr/>		
11.1	概述	183
11.2	SIO 模块规格差异.....	184
11.3	配置	185
11.4	寄存器说明	186

11.4.1	寄存器清单.....	186
11.4.2	SCxEN(启用寄存器).....	187
11.4.3	SCxBUF(缓存寄存器).....	187
11.4.4	SCxCR(控制寄存器).....	189
11.4.5	SCxMOD0(模态控制寄存器 0).....	191
11.4.6	SCxMOD1(模态控制寄存器 1).....	192
11.4.7	SCxMOD2(模态控制寄存器 2).....	193
11.4.8	SCxBRCR(波特率发生器控制寄存器).....	195
11.4.9	SCxBRADD(波特率发生器控制寄存器 2).....	196
11.4.10	SCxFCNF(FIFO 配置寄存器).....	197
11.4.11	SCxRFC(接收 FIFO 配置寄存器).....	198
11.4.12	SCxTFC(发送 FIFO 配置寄存器).....	199
11.4.13	SCxRST(接收 FIFO 状态寄存器).....	200
11.4.14	SCxTST(发送 FIFO 状态寄存器).....	201
11.5	在各模式下的操作	202
11.6	数据格式	203
11.6.1	数据格式列表.....	203
11.6.2	奇偶校验控制器.....	204
11.6.2.1	传送.....	204
11.6.2.2	接收数据.....	204
11.6.3	STOP 位长度.....	204
11.7	时钟控制	205
11.7.1	预分频器.....	205
11.7.2	串行时钟脉冲振荡电路.....	206
11.7.2.1	波特率生成器.....	206
11.7.2.2	时钟选择电路.....	208
11.7.3	发送/接收缓存器和 FIFO.....	210
11.7.3.1	配置.....	210
11.7.3.2	发送/接收缓存.....	210
11.7.3.3	初始化发送缓存.....	211
11.7.3.4	FIFO.....	211
11.8	状态标志	212
11.9	错误标记	212
11.9.1	OERR 标志.....	212
11.9.2	PERR 标志.....	212
11.9.3	FERR 标志.....	213
11.10	接收	214
11.10.1	接收计数器.....	214
11.10.2	接收控制单元.....	214
11.10.2.1	I/O 接口模式.....	214
11.10.2.2	UART 模式.....	214
11.10.3	接收操作.....	214
11.10.3.1	接收缓存器.....	214
11.10.3.2	接收 FIFO 操作.....	215
11.10.3.3	带 SCLK 输出的 I/O 接口模式.....	216
11.10.3.4	读取接收的数据.....	216
11.10.3.5	唤醒功能.....	216
11.10.3.6	超限操作错误.....	216
11.11	传送	218
11.11.1	发送计数器.....	218
11.11.2	发送控制器.....	218
11.11.2.1	I/O 接口模式.....	218
11.11.2.2	UART 模式.....	218
11.11.3	发送操作.....	219
11.11.3.1	传送缓存操作.....	219
11.11.3.2	发送 FIFO 操作.....	219
11.11.3.3	带 SCLK 输出的 I/O 接口模式/传送.....	221
11.11.3.4	I/O 接口模式下最后位输出后的 TXDx 引脚电平.....	221
11.11.3.5	在运行错误.....	222
11.11.3.6	I/O 接口 SCLK 输入模式下的数据保持时间.....	222
11.12	握手功能	223
11.13	中断/错误生成时序	224

11.13.1	接收中断.....	224
11.13.1.1	单缓存/双缓存.....	224
11.13.1.2	FIFO.....	224
11.13.2	发送中断.....	225
11.13.2.1	单缓存/双缓存.....	225
11.13.2.2	FIFO.....	225
11.13.3	错误生成.....	226
11.13.3.1	UART 模式.....	226
11.13.3.2	I/O 接口模式.....	226
11.14	软件复位.....	226
11.15	在各模式下的操作.....	227
11.15.1	模式 0 (I/O 接口模式).....	227
11.15.1.1	传输数据.....	227
11.15.1.2	接收.....	231
11.15.1.3	传输和接收(全双工).....	234
11.15.2	模式 1 (7-位 UART 模式).....	238
11.15.3	模式 2 (8-位 UART 模式).....	238
11.15.4	模式 3(9-位 UART 模式).....	239
11.15.4.1	唤醒功能.....	239
11.15.4.2	协议.....	239
12.	串行总线接口(I2C/SIO):.....	241
12.1	配置.....	242
12.2	寄存器.....	243
12.2.1	各通道寄存器.....	243
12.3	I2C 总线模式数据格式.....	244
12.4	I2C 总线模式控制寄存器.....	245
12.4.1	SBICR0(控制寄存器 0).....	245
12.4.2	SBICR1(控制寄存器 1).....	246
12.4.3	SBICR2(控制寄存器 2).....	248
12.4.4	SBISR(状态寄存器).....	249
12.4.5	SBIBR0(串行总线接口波特率寄存器 0).....	250
12.4.6	SBIDBR(串行总线接口数据缓存寄存器).....	250
12.4.7	SBII2CAR(I2C 总线地址寄存器).....	251
12.5	I2C 总线模式下控制.....	252
12.5.1	串行时钟.....	252
12.5.1.1	时钟源.....	252
12.5.1.2	时钟同步.....	252
12.5.2	设置应答模式.....	253
12.5.3	设置每次传输的位数.....	253
12.5.4	从地址和地址识别模式.....	253
12.5.5	操作模式.....	253
12.5.6	配置 SBI 为传输器或接收器.....	254
12.5.7	将 SBI 作为主机模式或从机模式进行配置.....	254
12.5.8	生成启动和停止条件.....	254
12.5.9	中断服务请求和释放.....	255
12.5.10	仲裁丢失检测监视器.....	255
12.5.11	从地址匹配检测监视器.....	257
12.5.12	一般调用检测监视器.....	257
12.5.13	最后接收位监视器.....	257
12.5.14	数据缓存寄存器(SBIDBR).....	257
12.5.15	波特率寄存器(SBIBR0).....	258
12.5.16	软件复位.....	258
12.6	在 I2C 总线模式下数据传输程序.....	259
12.6.1	设备初始化.....	259
12.6.2	生成启动条件和从地址.....	259
12.6.2.1	主机模式.....	259
12.6.2.2	从机模式.....	260
12.6.3	传输数据字.....	261
12.6.3.1	主机模式(<MST> = "1").....	261
12.6.3.2	从机模式(<MST> = "0").....	264

12.6.4	生成停止条件	266
12.6.5	重启程序	266
12.7	多主机使用注意事项	268
12.8	SIO 模式控制寄存器.....	269
12.8.1	SBICR0(控制寄存器 0).....	269
12.8.2	SBICR1(控制寄存器 1).....	270
12.8.3	SBIDBR(数据缓存寄存器).....	271
12.8.4	SBICR2(控制寄存器 2).....	272
12.8.5	SBISR(状态寄存器).....	273
12.8.6	SBIBR0(波特率寄存器 0).....	274
12.9	SIO 模式控制.....	275
12.9.1	串行时钟.....	275
12.9.1.1	时钟源	275
12.9.1.2	移位沿	276
12.9.2	传输模式	277
12.9.2.1	8-位传输模式	277
12.9.2.2	8-位接收模式	279
12.9.2.3	8 位传输/接收模式.....	280
12.9.2.4	传输结束时最后位的数据保留时间.....	281
13.	12 位模拟数字转换器.....	283
13.1	功能和特点	283
13.2	方块图	283
13.3	寄存器列表	284
13.4	寄存器描述	286
13.4.1	ADxCLK(时钟设置寄存器)	286
13.4.2	ADxMOD0 (模式设置寄存器 0).....	287
13.4.3	ADxMOD1 (模式设置寄存器 1).....	288
13.4.4	ADxMOD2 (模式设置寄存器 2).....	289
13.4.5	ADxMOD3 (模式设置寄存器 3).....	290
13.4.6	ADxCMPCR0(监控设置寄存器 0)	291
13.4.7	ADxCMPCR1(监视设置寄存器 1)	292
13.4.8	ADxCMP0(转换结果比较寄存器 0)	293
13.4.9	ADxCMP1(转换结果比较寄存器 1)	293
13.4.10	ADxREG0(转换结果寄存器 0)	294
13.4.11	ADxREG1(转换结果寄存器 1)	295
13.4.12	ADxREG2(转换结果寄存器 2)	296
13.4.13	ADxREG3(转换结果寄存器 3)	297
13.4.14	ADxREG4(转换结果寄存器 4)	298
13.4.15	ADxREG5(转换结果寄存器 5)	299
13.4.16	ADxREG6(转换结果寄存器 6)	300
13.4.17	ADxREG7(转换结果寄存器 7)	301
13.4.18	ADxREG8(转换结果寄存器 8)	302
13.4.19	ADxREG9(转换结果寄存器 9)	303
13.4.20	ADxREG10(转换结果寄存器 10).....	304
13.4.21	ADxREG11(转换结果寄存器 11).....	305
13.4.22	PMD 触发程序寄存器	306
13.4.22.1	ADxPSEL6~ADxPSEL11(PMD 触发程序编号选择寄存器 6~11)	307
13.4.22.2	ADxPINTS0 ~5 (PMD 触发中断选择寄存器 0~5).....	311
13.4.22.3	ADxPSET0~5(PMD 触发程序寄存器 0~5)	314
13.4.23	ADxTSET03 / ADxTSET47 / ADxTSET811 (定时器触发程序寄存器).....	322
13.4.24	ADxSSET03 / ADxSSET47 / ADxSSET811(软件触发程序寄存器).....	326
13.4.25	ADxASET03 / ADxASET47 / ADxASET811(常量转换程序寄存器).....	330
13.5	操作说明	334
13.5.1	模拟标准电压	334
13.5.2	启动 AD 转换.....	334
13.5.3	AD 转换监控功能.....	335
13.6	AD 转换时序图.....	336
13.6.1	软件触发转换	336

13.6.2	常量转换.....	337
13.6.3	通过触发进行 AD 转换.....	338
13.7	使用举例.....	340
13.7.1	利用一个 PMD1(三个分流器)和一个 ADC 进行的连续转换.....	340
13.7.2	利用一个 PMD1(一条分路)与一个 ADC 进行连续转换.....	341

14. 马达控制电路(PMD: 可编程马达驱动器).....343

14.1	PMD 输入/输出信号.....	345
14.2	PDM 电路配置.....	346
14.3	PMD 寄存器.....	347
14.3.1	PMD1MDEN(PMD 启用寄存器).....	348
14.3.2	PMD1PORTMD(端口输出模式寄存器).....	349
14.3.3	PMD1MODESEL (模式选择寄存器).....	350
14.3.4	脉冲宽度调制电路.....	351
14.3.4.1	PMD1MDCR(PMD 控制寄存器).....	354
14.3.4.2	PMD1CNTSTA(PWM 计数器状态寄存器).....	357
14.3.4.3	PMD1MDCNT(PWM 计数器寄存器).....	358
14.3.4.4	PMD1MDPRD(PWM 周期寄存器).....	359
14.3.4.5	PMD1CMPU(U 相 PWM 比较寄存器).....	360
14.3.4.6	PMD1CMPV(V 相 PWM 比较寄存器).....	361
14.3.4.7	PMD1CMPW(W 相 PWM 比较寄存器).....	362
14.3.5	导通控制电路.....	363
14.3.5.1	PMD1MDPOT(PMD 输出设置寄存器).....	365
14.3.5.2	PMD1MDOUT(PMD 导电控制寄存器).....	366
14.3.6	保护控制电路.....	368
14.3.6.1	EMG 保护电路.....	369
14.3.6.2	PMD1EMGREL(EMG 解除寄存器).....	370
14.3.6.3	PMD1EMGCR(EMG 控制寄存器).....	371
14.3.6.4	PMD1EMGSTA(EMG 状态寄存器).....	372
14.3.6.5	OVV 保护控制电路(OVV 输入字块).....	373
14.3.6.6	PMD1OVVCR(OVV 控制寄存器).....	374
14.3.6.7	PMD1OVVSTA(OVV 状态寄存器).....	376
14.3.7	停滞时间电路.....	377
14.3.7.1	PMD1DTR(停滞时间寄存器).....	378
14.3.8	Sync 触发生成电路.....	379
14.3.8.1	PMD1TRGCMP0(触发比较寄存器 0).....	381
14.3.8.2	PMD1TRGCMP1(触发比较寄存器 1).....	382
14.3.8.3	PMD1TRGCMP2(触发比较寄存器 2).....	383
14.3.8.4	PMD1TRGCMP3(触发比较寄存器 3).....	384
14.3.8.5	PMD1TRGCR(触发控制寄存器).....	385
14.3.8.6	PMD1TRGSYNCR(触发更新时序设置寄存器).....	388
14.3.8.7	PMD1TRGMD(触发输出模式设置寄存器).....	389
14.3.8.8	PMD1TRGSEL(触发输出选择寄存器).....	390

15. 矢量引擎(VE+).....391

15.1	概述.....	391
15.1.1	特点.....	391
15.1.2	主要规格.....	392
15.2	配置.....	392
15.2.1	矢量引擎, 马达控制电路和 A/D 转换器之间的相互作用.....	393
15.3	寄存器列表.....	394
15.3.1	寄存器列表.....	394
15.3.2	VE 控制寄存器.....	396
15.3.2.1	VEEN(VE 启用/禁用寄存器).....	396
15.3.2.2	VECPURUNTRG(CPU 开始触发选择寄存器).....	397
15.3.2.3	VETASKAPP(任务选择寄存器).....	398
15.3.2.4	VEACTSCH(操作进程计划选择寄存器).....	399
15.3.2.5	VEREPTIME(进程重复计数).....	400
15.3.2.6	VETRGMODE(启动触发模式).....	401
15.3.2.7	VEERRINTEN(错误中断启用/禁用).....	402
15.3.2.8	VECOMPEND(VE 强制终止).....	403
15.3.2.9	VEERRDET(错误检测).....	404

15.3.2.10	VESCHTASKRUN(进程执行标志/执行中的任务).....	405
15.3.2.11	VETMPREG0(暂存寄存器 0).....	406
15.3.2.12	VETMPREG1(暂存寄存器 1).....	406
15.3.2.13	VETMPREG2(暂存寄存器 2).....	407
15.3.2.14	VETMPREG3(暂存寄存器 3).....	407
15.3.2.15	VETMPREG4(暂存寄存器 4).....	408
15.3.2.16	VETMPREG5(暂存寄存器 5).....	408
15.3.3	公用寄存器.....	409
15.3.3.1	VETADC(ADC 启动等待设置寄存器).....	409
15.3.4	特定通道寄存器.....	410
15.3.4.1	VE 模式 1(任务控制模式寄存器).....	410
15.3.4.2	VEFMODE1(流量控制寄存器).....	412
15.3.4.3	VETPWM1(PWM 周期速率寄存器).....	414
15.3.4.4	VEOMEGA1(转速寄存器).....	415
15.3.4.5	VETHETA1(马达相寄存器).....	416
15.3.4.6	VECOS1/VEVIN1/VECOSM1/VEVINM1(SIN/COS 寄存器).....	417
15.3.4.7	VEIDREF1/VEIQREF1(d-轴/q-轴电流标准寄存器).....	421
15.3.4.8	VEVD1/VEVQ1(d-轴/q-轴电压寄存器).....	423
15.3.4.9	VECIDK1/VECIDKP1/VEVCIQK1/VEVCIQKP1(PI 控制系数寄存器).....	425
15.3.4.10	VEVDIH1/VEVDILH1/VEVQIH1/VEVQILH1(PI 控制积分项寄存器).....	427
15.3.4.11	VEMCTLF1(状态标志寄存器).....	429
15.3.4.12	VEFPWMCHG1(PWM 切换速度设置寄存器).....	430
15.3.4.13	VEMDPRD1(PWM 周期寄存器).....	431
15.3.4.14	VEMINPLS1(最小脉冲宽度设置寄存器).....	432
15.3.4.15	VESECTOR1/VESECTORM1(扇区信息寄存器).....	433
15.3.4.16	VEIAO1/VEIBO1/VEICO1(零电流寄存器).....	435
15.3.4.17	VEIAADC1/VEIBADC1/VEICADC1(电流 ADC 结果寄存器).....	438
15.3.4.18	VEVDC1/VEVDCL1(供给电压寄存器).....	441
15.3.4.19	VEID1/VEIQ1(d-轴/q-轴电流寄存器).....	443
15.3.4.20	VECMPI1 / VECMPV1/ VECMPW1(PWM 工作寄存器).....	445
15.3.4.21	VEOUTCR1(6 相输出控制寄存器).....	448
15.3.4.22	VETRGCRC1(同步触发校正寄存器).....	450
15.3.4.23	VETRGCMP01/VETRGCMP11(触发时间设置寄存器).....	451
15.3.4.24	VETRGSSEL1(同步触发选择寄存器).....	453
15.3.4.25	VEEMGRS1(EMG 返回控制寄存器).....	454
15.4	操作说明.....	455
15.4.1	进程管理.....	455
15.4.1.1	进程计划控制.....	456
15.4.1.2	启动控制.....	458
15.4.2	任务一览.....	458
15.4.2.1	电流控制(任务 5).....	460
15.4.2.2	SIN/COS 计算(任务 6).....	461
15.4.2.3	输出电压转换(坐标轴转换/相转换).....	462
15.4.2.4	输出控制.....	467
15.4.2.5	触发生成.....	469
15.4.2.6	输入处理.....	470
15.4.2.7	输入电流变换(相位变换/坐标轴变换).....	472
15.4.2.8	其它任务.....	473
15.5	VE 通道通道, ADC 单元和 PMD 通道通道的组合.....	474
16.	运算放大器(AMP).....	475
16.1	配置.....	475
16.2	寄存器列表.....	476
16.2.1	运算放大器.....	476
16.2.1.1	AMPCTLD(Amp D 控制寄存器).....	476
16.3	运行.....	477
16.3.1	基本运行.....	477
17.	编码器输入电路(ENC).....	479
17.1	概述.....	479
17.2	各通道之间的差值.....	479
17.3	方块图.....	479

17.4 寄存器	480
17.4.1 寄存器列表	480
17.4.2 编码器输入控制寄存器(ENTNCR)	481
17.4.3 编码器计数器重载寄存器(ENRELOAD)	484
17.4.4 ENINT(编码器比较寄存器)	485
17.4.5 ENCNT(编码器计数器)	486
17.5 操作说明	487
17.5.1 编码器模式	487
17.5.2 传感器模式	487
17.5.2.1 事件计数模式	487
17.5.2.2 定时器计数模式	487
17.5.3 定时器模式	487
17.6 功能	488
17.6.1 模式运算要点	488
17.6.1.1 编码器模式	488
17.6.1.2 传感器模式(事件计数)	489
17.6.1.3 传感器模式(定时器计数)	490
17.6.1.4 定时器模式	491
17.6.2 $\langle \text{CMPEN} \rangle = 1$ 时的计数器和中断生成运算	493
17.6.2.1 编码器模式	493
17.6.2.2 传感器模式(事件计数)	493
17.6.2.3 传感器模式(定时器计数)	493
17.6.2.4 定时器模式	494
17.6.3 $\langle \text{CMPEN} \rangle = 0$ 时的计数器和中断生成运算	495
17.6.3.1 编码器模式	495
17.6.3.2 传感器模式(事件计数)	495
17.6.3.3 传感器模式(定时器计数)	495
17.6.3.4 定时器模式	496
17.6.4 旋转方向检测	497
17.6.5 计数器电路	498
17.6.5.1 操作说明	498
17.6.6 中断	499
17.6.6.1 操作性说明	499
<hr/>	
18. 上电复位电路 (POR)	500
18.1 结构	500
18.2 功能	501
<hr/>	
19. 低压检测电路 (VLTD)	502
19.1 结构	502
19.2 寄存器	503
19.2.1 寄存器列表	503
19.2.2 电压检测控制寄存器(VDCR)	503
19.3 操作说明	504
19.3.1 控制	504
19.3.2 功能	504
19.3.2.1 启用/禁用电压检测	504
19.3.2.2 选择检测电压电平	504
<hr/>	
20. 振荡频率检测器(OFD)	506
20.1 方块图	507
20.2 寄存器	508
20.2.1 寄存器列表	508
20.2.1.1 OFDCR1(振荡频率检测控制寄存器)	509
20.2.1.2 OFDCR2(振荡频率检测控制寄存器 2)	510
20.2.1.3 OFDMNPLLOFF(较低检测频率设置寄存器(PLL OFF))	511
20.2.1.4 OFDMNPLLON(较低检测频率设置寄存器(PLL ON))	512
20.2.1.5 OFDMXPLLOFF(高检测频率设置寄存器(PLL OFF))	513

20.2.1.6	OFDMXPLLON(高检测频率设置寄存器(PLLON)).....	514
20.3	操作说明	515
20.3.1	设置	515
20.3.2	操作	516
20.3.3	检测频率	517
20.3.4	可用的运行模式	518
20.3.5	运算程序示例	518
<hr/>		
21.	看门狗定时器(WDT).....	520
<hr/>		
21.1	配置	520
21.2	寄存器	521
21.2.1	WDMOD (看门狗定时器模式寄存器).....	521
21.2.2	WDCR(看门狗定时器控制寄存器).....	523
21.3	运行	524
21.3.1	基本运行	524
21.3.2	运行模式和状态	524
21.4	在故障(失控)被检测时运行.....	525
21.4.1	INTWDT 中断生成.....	525
21.4.2	内部复位生成.....	526
21.5	控制寄存器	527
21.5.1	看门狗定时器寄存器地址(WDMOD).....	527
21.5.2	看门狗定时器控制寄存器(WDCR).....	527
21.5.3	设置例子	528
21.5.3.1	禁用控制	528
21.5.3.2	启用控制	528
21.5.3.3	看门狗定时器清除控制	528
21.5.3.4	看门狗定时器的检测时间	528
<hr/>		
22.	闪存操作.....	530
<hr/>		
22.1	特点	530
22.1.1	存储器容量和配置	530
22.1.2	功能.....	531
22.1.3	运行模式.....	531
22.1.3.1	模式说明	531
22.1.3.2	模式判定	532
22.1.4	存储器映像	533
22.1.5	保护/安全功能.....	533
22.1.5.1	保护功能	534
22.1.5.2	安全功能	534
22.1.6	寄存器.....	535
22.1.6.1	寄存器列表.....	535
22.1.6.2	FCCR(闪存接口控制寄存器)	535
22.1.6.3	FCSR(闪存状态寄存器).....	536
22.1.6.4	FCSECBIT (安全位寄存器).....	536
22.1.6.5	FCPSRA(闪存保护状态寄存器).....	537
22.2	闪存详细情况	538
22.2.1	功能	538
22.2.2	闪存运行模式	538
22.2.3	硬件复位.....	538
22.2.4	如何执行指令	539
22.2.5	指令说明	539
22.2.5.1	自动页面程序.....	540
22.2.5.2	自动芯片擦除.....	540
22.2.5.3	自动块擦除	541
22.2.5.4	自动保护位程序	541
22.2.5.5	自动保护位擦除.....	541
22.2.5.6	ID-读取	542
22.2.5.7	读取指令和读取/复位指令(软件复位).....	542
22.2.6	指令序列	543
22.2.6.1	指令序列列表.....	543

22.2.6.2	总线周期中的地址位配置	544
22.2.6.3	块地址(BA)	544
22.2.6.4	如何规定保护位(PBA)	545
22.2.6.5	ID-读取代码 (IA, ID)	545
22.2.6.6	指令序列举例	546
22.2.7	流程图	547
22.2.7.1	自动程序	547
22.2.7.2	自动擦除	548
22.3	如何利用单模式为闪存编程	549
22.3.1	模式设置	549
22.3.2	接口规格	549
22.3.3	对内存存储器的限值	550
22.3.4	操作指令	550
22.3.4.1	RAM 传输	550
22.3.4.2	闪存芯片擦除和保护位擦除	551
22.3.5	无关指令的公共操作	551
22.3.5.1	串行运行模式判定	551
22.3.5.2	应答数据	553
22.3.5.3	密码判定	554
22.3.5.4	CHECK SUM 计算	555
22.3.6	在 RAM 传输下的传输格式	557
22.3.7	闪存芯片擦除和保护位擦除的转换格式	559
22.3.8	启动程序 WHOLE 流程图	561
22.3.9	片上 BOOT ROM 中使用重编程算法重编闪存程序	563
22.3.9.1	步骤-1	563
22.3.9.2	步骤-2	563
22.3.9.3	步骤-3	564
22.3.9.4	步骤-4	564
22.3.9.5	步骤-5	565
22.3.9.6	步骤-6	565
22.4	用户启动模式下编程	566
22.4.1	(1-A)编程程序存储于闪存的步骤	566
22.4.1.1	步骤-1	566
22.4.1.2	步骤-2	567
22.4.1.3	步骤-3	567
22.4.1.4	步骤-4	568
22.4.1.5	步骤-5	568
22.4.1.6	步骤-6	569
22.4.2	(1-B)编程程序自外部主机进行传输的步骤	570
22.4.2.1	步骤-1	570
22.4.2.2	步骤-2	571
22.4.2.3	步骤-3	571
22.4.2.4	步骤-4	572
22.4.2.5	步骤-5	572
22.4.2.6	步骤-6	573

23. 调试接口 **574**

23.1	规范概述	574
23.2	SWJ-DP 的特点	574
23.3	ETM 的特点	574
23.4	引脚功能	575
23.5	与调试工具的连接	576
23.5.1	连接方式	576
23.5.2	在使用通用端口时	576
23.6	HALT 模式期间的外部设备运行	576

24. 端口部分等效电路示意图 **578**

24.1	PB3~6, PE0~2, PG0~6	578
24.2	PJ6~7, PK0~1	578
24.3	PF0	579

24.4 PM0~1	579
24.5 X1, X2.....	580
24.6 RESET	580
24.7 MODE	580
24.8 VREFHB VREFLB.....	580
24.9 VOUT15, VOUT3	581
<hr/>	
25. 电气特性.....	582
<hr/>	
25.1 绝对最大额定值	582
25.2 直流电电气特性(1/2).....	583
25.3 DC 电气特性(2/2).....	584
25.4 12 位 ADC 电气特性.....	584
25.5 Op-Amps 电气特性.....	584
25.6 AC 电气特性.....	585
25.6.1 AC 测量条件	585
25.6.2 串行信道时序(SIO).....	585
25.6.2.1 I/O 接口模式	585
25.6.3 串行总线接口(I2C/SIO)	586
25.6.3.1 I2C 模式	586
25.6.3.2 时钟同步 8-位 SIO 模式	588
25.6.4 事件计数器	589
25.6.5 捕捉.....	589
25.6.6 外部中断.....	590
25.6.7 调试通讯.....	591
25.6.7.1 AC 测量条件	591
25.6.7.2 SWD 接口.....	591
25.6.7.3 JTAG 接口	591
25.6.8 闪存特性.....	592
25.6.9 芯片上振荡器.....	592
25.6.10 外部振荡器举例	592
25.7 振荡电路	593
25.7.1 建议采用的陶瓷振荡器	593
<hr/>	
26. 封装尺寸.....	594
<hr/>	

CMOS 32 位微控制器

TMPM375FSDMG

TMPM375FSDMG为32 位RISC微型处理器序列，带有 ARM®Cortex®-M3 显微处理器内核。其特性如下所述：

1.1 特点

1. ARM Cortex-M3微处理器内核
 - a. 通过使用拇指@-2指令，提高了编码效率。
 - 新 16 位Thumb指令改进了程序流程
 - 新 32 位Thumb指令改进了性能
 - 新Thumb混合的 16 / 32 位指令集能产生更快，更高效的编码
 - b. 已实现高性能，低功耗。
 - [高性能]
 - 32位倍增($32 \times 32 = 32$ 位)可以由一个时钟执行完成
 - 根据被除数和除数，除法执行 2 ~ 12 次
 - [低功耗]
 - 用低功耗库优化了设计
 - 备用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
 - 由硬件自动处理入栈
2. 单片级程序存储器与数据内存
 - 芯片内RAM: 4千字节
 - 芯片内闪存ROM: 64千字节
3. 16 位定时器 (TMRB): 4 通道
 - 16 位区间计时器模式
 - 16 位事件计数器模式
 - 输入捕捉功能
 - 16 位PPG输出
 - 外部触发器PPG输出
4. 看门狗定时器 (WDT) : 1 通道
看门狗定时器(WDT)生成复位
5. 上电复位功能 (POR)
6. 电压监测功能 (VLTD)

7. 振荡频率检测功能(OFD)
8. 矢量发动机(VE+): 1单元
 - 马达控制计算电路
 - 对应1个发动机
9. 可编程马达驱动 (PMD) : 1通道
 - 3相补充PWM生成器
 - 同步AD转换开始触发器生成器
 - 应急保护功能(EMG)
10. 编码器输入电路 (ENC): 1通道
 - 对应增量编码器(AB / ABZ)
 - 旋转方向检测
 - 绝对位置检测计数器
 - 位置检测比较器
 - 噪声滤波器
 - 3相传感器输入
11. 通用串行接口(SIO/UART): 2通道
 - 可以选择UART模式或同步模式(配置4字节FIFO)(1通道)
 - UART模式(1通道)
12. 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。
13. 12 位 AD 变流器(ADC): 1单元(模拟输入: 4通道)
 - 由内部触发器触发开始: TMRB 中断/ PMD触发器
 - 恒定转换模式
 - AD监测 2 个通道
 - 转换速率为2 μ s(@ADC转换时钟= 40兆赫兹)
14. OP-Amp(AMP) : 1通道
 - 8个增益模式可以选择。
15. 输入/ 输出端口(PORT) : 21个引脚
 - 输入/输出引脚: 21个引脚
16. 中断源
 - 29个内部因子: 优先顺序设置为 7 级(看门狗定时器中断除外)
 - 3个外部因子: 优先顺序设置为 7 级
17. 待机模式

18. 时钟发生器 (CG)
 - 片上内PLL(4或5倍)
 - 时钟变速器功能: 高速时钟分为 1/1, 1/2, 1/4, 1/8 或 1/16

19. 字节序
 - 低位优先

20. 内部高速振荡电路: 10MHz

21. 最高工作频率:40MHz

22. 工作电压范围
 - DVDD5B = 3.9V~5.5V($f_{\text{sys}}=40\text{MHz}$)
 - (4.5V~5.5V:全功能操作)
 - (3.9V~4.5V:12位ADC, FLASH写入等的电气特性限制)

23. 温度范围
 - $-40\text{ }^{\circ}\text{C}\sim 105\text{ }^{\circ}\text{C}$ (闪存写入/擦除数据时除外)
 - $0\text{ }^{\circ}\text{C}\sim 70\text{ }^{\circ}\text{C}$ (FLASH写入/擦除数据时)

24. 封装
 - SSOP30(5.6mm x 9.7mm, 0.65mm节距)

1.2 方块图

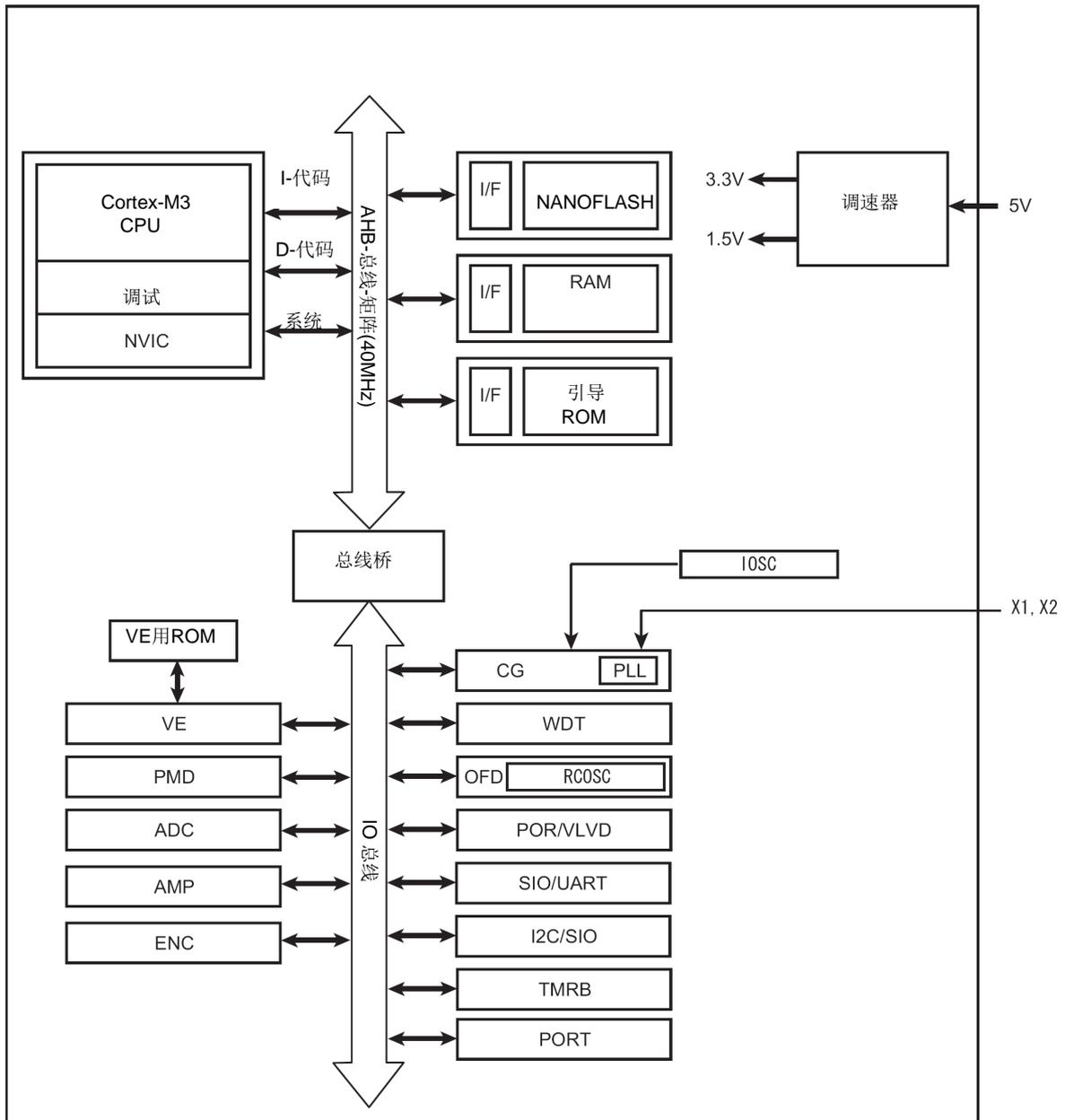


图1-1 TMPM375FSDMG 方块图

1.3 引脚分配(顶视图)

TMPM375FSDMG的引脚布局图如下图所示。

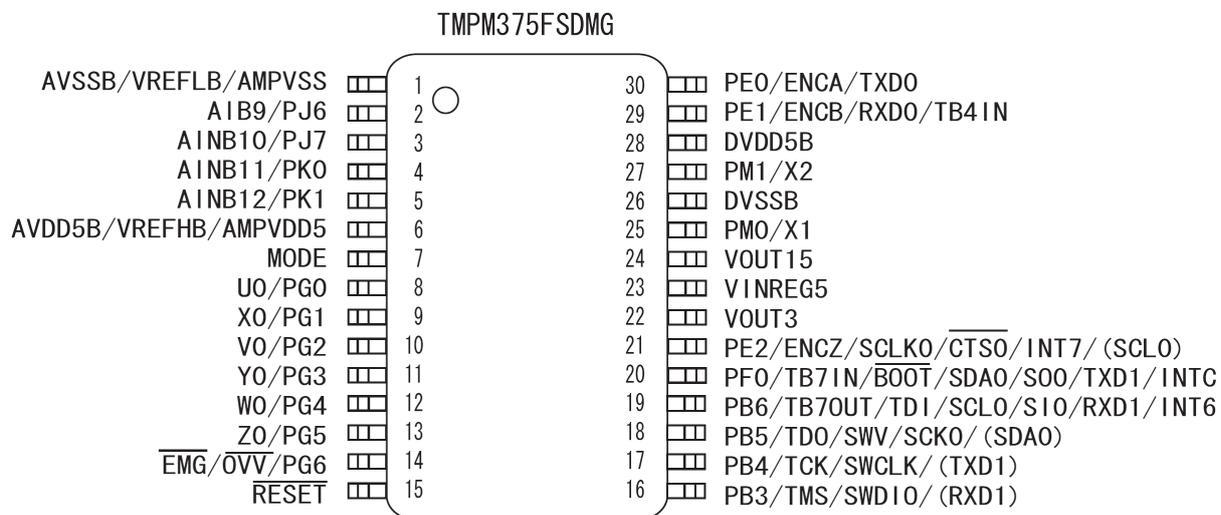


图1-2 引脚布局图(SSOP30)

1.4 引脚名称与功能

表 1-1 中，TMPM375FSDMG的输入和输出引脚按引脚或端口分类。每个表格中包含多功能引脚交替出现的引脚名称和功能。

1.4.1 按端口分类

表 1-1 引脚名称与功能按端口分类(1/2)

PORT	型号	引脚编号	引脚名称	输入/输出	功能
PORT B	功能/调试	16	PB3	I/O	I/O 端口
			TMS/SWDIO	I/O	调试引脚
			(RXD1)	I	接收串行数据
PORT B	功能/调试	17	PB4	I/O	I/O 端口
			TCK/SWCLK	I/O	调试引脚
			(TXD1)	O	发送串行数据
PORT B	功能/调试	18	PB5	I/O	I/O端口
			TDO/SWV	O	调试引脚
			SCK0	I/O	串行时钟
			(SDA0)	I/O	I2C数据输入/输出
PORT B	功能/调试	19	PB6	I/O	I/O 端口
			TDI	I	调试引脚
			SI0/SCL0	I/O	串行数据输入/ I2C时钟输入/输出
			TB7OUT	O	16 位定时器输出
			INT6	I	外部中断输入引脚
			RXD1	I	接收串行数据
PORT E	功能	30	PE0	I/O	I/O 端口
			TXD0	O	发送串行数据
			ENCA	I	A相输入引脚
PORT E	功能	29	PE1	I/O	I/O 端口
			RXD0	I	接收串行数据
			ENCB	I	B相输入引脚
			TB4IN	I	输入16 位定时器捕捉触发器
PORT E	功能	21	PE2	I/O	I/O 端口
			SCLK0	I/O	串行时钟输入/输出
			CTS0	I	握手输入引脚
			ENCZ	I	Z相输入引脚
			INT7 (SCL0)	I/O	外部中断输入引脚 I2C时钟输入/输出
PORT F	功能/控制	20	PF0	I/O	I/O 端口
			TB7IN	I	输入16 位定时器捕捉触发器
			SO0/SDA0	I/O	串行数据输出/ I2C数据输入/输出
			BOOT	I	BOOT模式引脚。(注)在复位信号升高时对“低”采样，该引脚可进入单启动模式。
			TXD1	O	发送串行数据
INTC	I	中断输入引脚			
PORT G	功能	8	PG0 U0	I/O O	I/O 端口 U相输出引脚
PORT G	功能	9	PG1 X0	I/O O	I/O 端口 X相输出引脚

表 1-1 引脚名称与功能按端口分类(2/2)

PORT	型号	引脚编号	引脚名称	输入/输出	功能
PORT G	功能	10	PG2 V0	I/O O	I/O 端口 V相输出引脚
PORT G	功能	11	PG3 Y0	I/O O	I/O 端口 Y相输出引脚
PORT G	功能	12	PG4 W0	I/O O	I/O 端口 W相输出引脚
PORT G	功能	13	PG5 Z0	I/O O	I/O 端口 Z相输出引脚
PORT G	功能	14	PG6 $\overline{\text{EMGI}}$ $\overline{\text{OVV}}$	I/O I I	I/O 端口 应急状态检测输入 过电压检测输入
PORT J	功能	2	PJ6 AINB9	I/O I	I/O 端口 模拟输入
PORT J	功能	3	PJ7 AINB10	I/O I	I/O 端口 模拟输入
PORT K	功能	.	PK0 AINB11	I/O I	I/O 端口 模拟输入
PORT K	功能	5	PK1 AINB12	I/O I	I/O 端口 模拟输入
PORT M	功能/时钟	25	PM0 x1	I/O I	I/O 端口 连接到高速振荡器。
PORT M	功能/时钟	27	PM1 x2	I/O O	I/O 端口 连接到高速振荡器。
-	控制	7	MODE	I	模式引脚 (注) 必须将MODE引脚连接至GND。
-	功能	15	$\overline{\text{RESET}}$	I	复位输入引脚 (注) 带有一个上拉与一个噪声滤波器(约30 ns (典型值))
-	压力开关	26	DVSSB	-	GND引脚
-	压力开关	28	DVDD5B	-	电源引脚
-	压力开关	23	VINREG 5	-	电源引脚
-	压力开关	24	VOUT15	-	电源引脚
-	压力开关	22	VOUT3	-	电源引脚
-	压力开关	1	AVSSB VREFLB AMPVSS	-	AD转换器: 接地引脚(注 1) 为AD转换器提供基准电源。Op-Amp接地引脚
-	压力开关	6	AVDD5B VREFHB AMPVDD 5	-	为AD转换器提供电源。(注2) 为AD转换器提供基准电源。为Op-Amp提供电源。

注1: 即使不使用AD转换器的情况下, AVSS必须始终与接地连接。

注2: 即使不使用AD转换器的情况下, 必须始终与电源连接。

1.5 引脚编号与电源引脚

表 1-2 引脚编号与电源

电源	电压范围	引脚编号	引脚名称
DVDD5B	4.5~ 5.5 V	28	PB, PE, PF, PG, PM, RESET, MODE
AVDD5B		6	PJ, PK
VINREG5	3.9 ~4.5 V(注)	23	-

注：功能操作有一定限制。具体情况参见"电气特性"。

表 1-3 片上内稳压器输出引脚

引脚名称	引脚编号	备注
VOUT15	19	VOUT15必须通过3.3~4.7微法的电容器与DVSS连接，向内部电路供电。
VOUT3	17	VOUT3必须通过3.3~4.7微法的电容器与DVSS连接，向内部电路供电。

注：VOUT15和VOUT3必须与相同容量的电容器连接。IC外部无法从VOUT15和VOUT3获得供电。

2. 处理器内核

该TX03系列有一个高性能的 32 位处理器内核(ARM Cortex-M3处理器内核)。本处理器内核的信息请参阅ARM Limited公司的"Cortex-M3 技术参考手册"。本章介绍只有 TX03 序列才具有的功能，该系列在本文件中没有详述。

2.1 处理器内核上的信息

TMPM375FSDMG中对处理器核心做了某些修订，如下表所示。

在以下URL链接中参阅有关CPU内核与构造的详细信息，参阅ARM公司手册"Cortex-M 系列处理器"。

<http://infocenter.arm.com/help/index.jsp>

产品名称	内核版次
TMPM375FSDMG	r2p1

2.2 可配置选项

Cortex-M3内核具备可选块。在r2p1修订版中，可选程序块包括ETM™和MPU。下表显示TMPM 375FSDMG中的可配置选项。

特点	配置选项
FPB	两个文字比较器 六个指令比较器
DWT	四个比较器
ITM	有
MPU	无
ETM	有
AHB-AP	有
AHB跟踪宏单元接口	无
TPIU	有
WIC	无
调试端口	JTAG/串行线
位区	有
恒定 AHB 控制	无

2.3 异常/中断

以下将对异常与中断进行说明。

2.3.1 中断输入的数目

可在Cortex-M3内核中，将中断输入的数目选择性地定义为 1 ~ 240 范围内的某个数值。

TMPM375FSDMG有 32 个中断输入。中断输入的数目可在NVIC寄存器的<INTLINESNUM[4:0]>位中反映出来。对于本产品，读取 <INTLINESNUM[4:0]>位时会读出0x00。

2.3.2 优先级中断位的数目

Cortex-M3内核可选择性地配置优先级中断位的数目，其配置范围为 3 位 ~ 8 位。

TMPM375FSDMG有3个优先级中断位。优先级中断位的数目，用于指定中断优先级寄存器与系统处理器优先寄存器中的优先级。

2.3.3 SysTick

Cortex-M3内核具备一个可生成SysTick异常的SysTick计时器。

SysTick异常，详见异常中的"SysTick"一节及NVIC寄存器中的SysTick寄存器。

2.3.4 SYSRESETREQ

当设置应用中断和复位控制寄存器的<SYSRESETREQ>位时，Cortex-M3内核会输出SYSRESETREQ信号。

当SYSRESETREQ信号输出时TMPM375FSDMG提供相同操作。

2.3.5 LOCKUP

当产生不能补救的异常时，Cortex-M3内核会输出LOCKUP信号，显示软件中包括的严重错误。

TMPM375FSDMG不使用该信号。为了从LOCKUP状态中返回，必须使用不可屏蔽中断(NMI)或复位。

2.3.6 辅助故障状态寄存器

Cortex-M3内核配备辅助故障状态寄存器，用以向软件提供附加系统故障信息。

然而，TMPM375FSDMG并未定义该功能。如读取辅助故障状态寄存器，则始终读出"0x0000_0000"。

2.4 事件

Cortex-M3内核具有事件输出信号与事件输入信号。事件输出信号由SEV指令执行来输出。如事件已输入，则内核从WFE指令所引起的低功耗模式返回。

TMPM375FSDMG不使用事件输出信号和事件输入信号。请不要使用SEV指令与WFE指令。

2.5 电源管理

Cortex-M3 内核配备有电源管理系统，该系统使用SLEEPING信号与SLEEPDEEP信号。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
- 如在设置系统控制寄存器的<SLEEPONEXIT>位情况下中断服务程序(ISR)退出时，则进行定时。

TMPM375FSDMG不使用SLEEPDEEP信号，因此请勿设置<SLEEPDEEP>位。其也不使用事件信号，因此请不要使用WFE指令。

有关电源管理的详细资料，请参看"时钟/模式控制"一节。

2.6 独占通道

在Cortex-M3内核中，该DCode总线系统支持独占通道。然而 TMPM375FSDMG 不使用该功能。



3. 存储器映象

3.1 存储器映象

TMPM375FSDMG的内存映射功能基于ARM Cortex-M3处理器核心内存映射。TMPM375FSDMG的内部ROM, 内部RAM和特殊功能寄存器(SFR)分别对应Cortex-M3的代码, SRAM和外设区。特殊功能寄存器(SFR)指的是所有输入/输出端口与外设功能的控制寄存器。SRAM和SFR区都包含在位带区内。

CPU寄存器区是处理器核心的内部寄存器区。

有关各区域的详细资料, 见Cortex-M3 技术参考手册。

注意访问各区时如出现"故障"信息, 如果启用了内存故障功能则会导致内存故障, 如果禁用了内存故障功能则导致硬件故障。同样, 请勿访问厂商专用区。

3.1.1 TMPM375FSDMG 内存映射

TMPM375FSDMG的内存映射如图 3-1 所示。

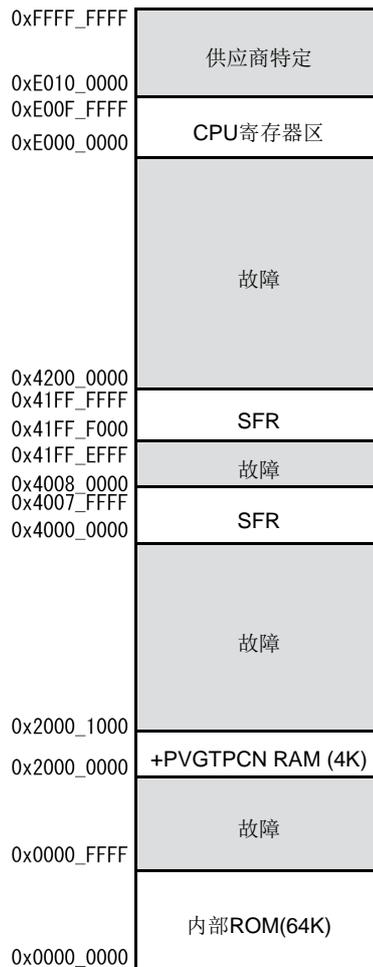


图 3-1 内存映射

3.2 SFR 区详情

SFR区的详细信息如表 3-1 所示。

不要访问表 3-1 中的保留区。有关详细资料见各外设功能相关节次。

表 3-1 SFR区详情

起始地址	结束地址	外设
0x4000_0000	0x4000_033F	PORT
0x4000_0340	0x4000_FFFF	保留
0x4001_0000	0x4001_01FF	TMRB
0x4001_0200	0x4001_03FF	保留
0x4001_0400	0x4001_043F	ENC
0x4001_0440	0x4001_FFFF	保留
0x4002_0000	0x4002_007F	I2C/SIO
0x4002_0080	0x4002_00FF	SIO / UART
0x4002_0100	0x4003_01FF	保留
0x4003_0200	0x4003_02FF	ADC
0x4003_0300	0x4003_0417	保留
0x4003_0418	0x4003_041F	AMP
0x4003_0420	0x4003_FFFF	保留
0x4004_0000	0x4004_003F	WDT
0x4004_0040	0x4004_01FF	保留
0x4004_0200	0x4004_022F	CG
0x4004_0230	0x4004_02FF	保留
0x4004_0300	0x4004_030F	TRM
0x4004_0310	0x4004_07FF	保留
0x4004_0800	0x4004_083F	OFD
0x4004_0840	0x4004_08FF	保留
0x4004_0900	0x4004_093F	VLTD
0x4004_0940	0x4004_FFFF	保留
0x4005_0000	0x4005_01FF	VE
0x4005_0200	0x4005_047F	保留
0x4005_0480	0x4005_04FF	PMD
0x4005_0500	0x4005_FFFF	保留
0x4006_0000	0x4006_0007	DNF
0x4006_0008	0x4007_FFFF	保留
0x4008_0000	0x41FF_EFFF	硬故障
0x41FF_F000	0x41FF_F03F	FLASH
0x41FF_F040	0x41FF_FFFF	保留

表 3-2 基地址列表

外设名称	基址	SFR
PORT B	0x4000_0040	PORT
PORT E	0x4000_0100	PORT
PORT F	0x4000_0140	PORT
PORT G	0x4000_0180	PORT

表 3-2 基地址列表

PORT J	0x4000_0240	PORT
PORT K	0x4000_0280	PORT
PORT M	0x4000_0300	PORT

TMRB 0	0x4001_0000	TMRB
TMRB 4	0x4001_0100	TMRB
TMRB 5	0x4001_0140	TMRB
TMRB 7	0x4001_01C0	TMRB

ENC 0	0x4001_0400	ENC
-------	-------------	-----

SBI 0	0x4002_0000	SBI
-------	-------------	-----

SIO UART 0	0x4002_0080	SIO / UART
SIO UART 1	0x4002_00C0	SIO / UART

ADC	0x4003_0200	ADC
-----	-------------	-----

AMP D	0x4003_0418	AMP
-------	-------------	-----

WDT	0x4004_0000	WDT
-----	-------------	-----

CG	0x4004_0200	CG
----	-------------	----

OSCTRM	0x4004_0300	TRM
--------	-------------	-----

OFD	0x4004_0800	OFD
-----	-------------	-----

LVD	0x4004_0900	LVD
-----	-------------	-----

VE	0x4005_0000	VE
----	-------------	----

PMD 1	0x4005_0480	PMD
-------	-------------	-----

DNF	0x4006_0000	DNF
-----	-------------	-----



4. 复位操作

复位操作源如下几种。

- 上电复位电路(POR)
- 电压检测电路(VLTD)
- RESET引脚 ($\overline{\text{RESET}}$)
- 看门狗定时器(WDT)
- 振荡频率电路(OFD)
- CPU和复位寄存器<SYSRESETREQ>位发出的信号导致的应用中断

辨别复位源时，检查"异常"一章中所述的时钟发生器寄存器中的CGRSTFLG。

有关上电复位电路，电源检测电路，看门狗定时器与振荡频率检测电路的详情，参阅各个章节。

关于<SYSRESETREQ>复位的详细信息，参见*Cortex-M3技术参考手册*。

注：复位操作一旦执行，内部RAM数据将无法保证。

4.1 冷复位

打开电源时，应考虑内置调节器，内置闪存存储器与内部高速振荡器的稳定时间。TMPM375FSDMG具有自动插入稳定时间的功能。

4.1.1 由上电复位电路(不用 $\overline{\text{RESET}}$ 引脚)进行复位

一旦电源电压超出上电复位的释放电压，电源计数器开始运行，大约3.2 毫秒之后释放内部复位信号。关于上电复位电路的运行信息，参见"上电复位电路(POR)"一节。

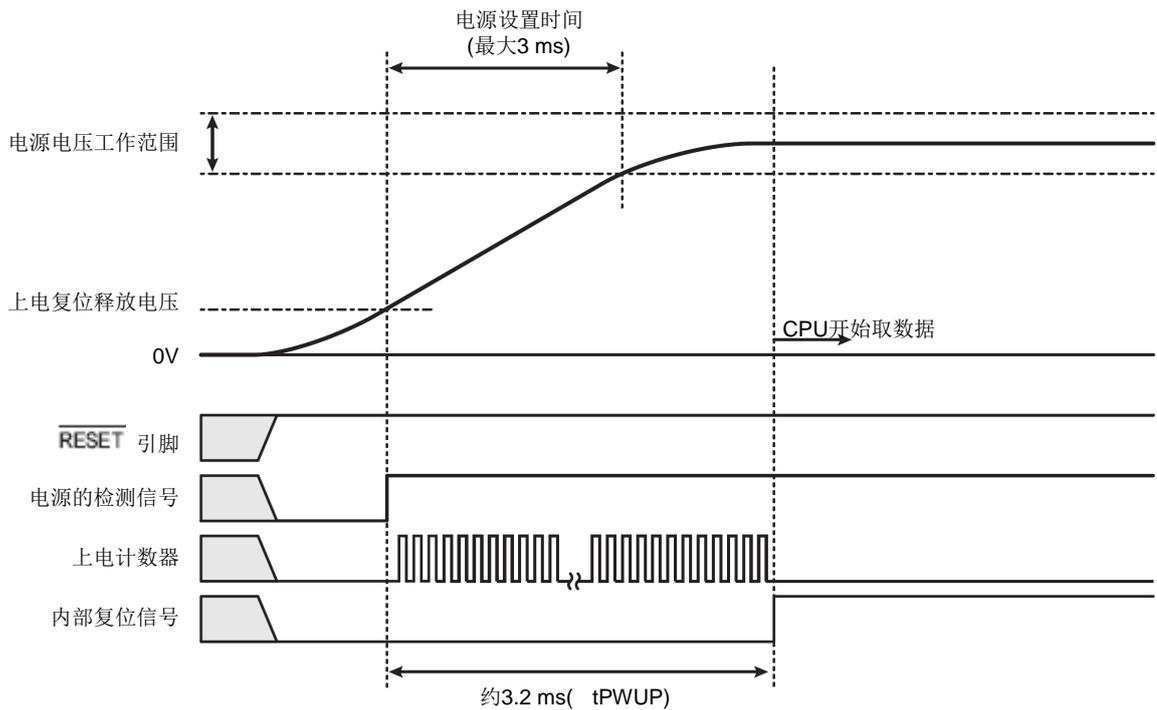


图 4-1 利用上电电路进行复位操作

4.1.2 由 $\overline{\text{RESET}}$ 引脚复位

利用 $\overline{\text{RESET}}$ 引脚进行复位在上电计数器完成后有效。如在上电复位信号变“高”后的3.2 ms 之内将引脚设置为“高”，则复位过程与 4.1.1 中所述的上电相同。

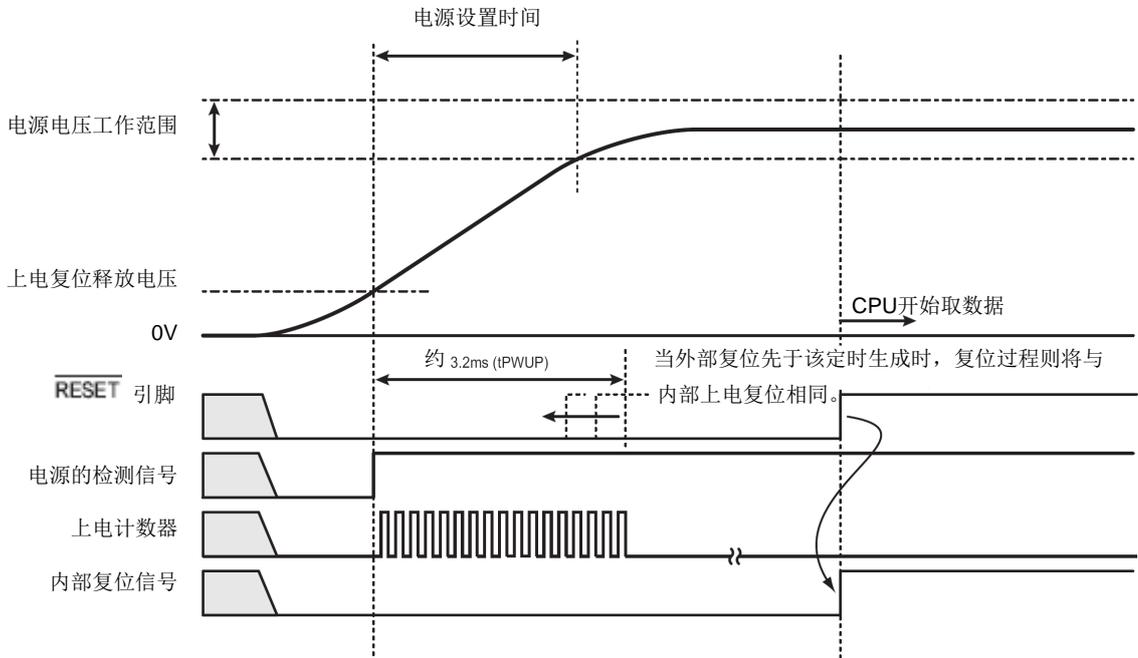


图 4-2 通过引脚进行复位操作

4.2 预热

4.2.1 复位时长

对TMPM375FSDMG进行复位时，需要满足以下条件；供电电压处于操作范围。至少12个系统时钟的复位 $\overline{\text{RESET}}$ 引脚通过内部高频振荡器保持在"低"位。引脚变"高"后，内部复位将被解除。

4.3 复位后

复位后，Cortex-M3的控制寄存器和外设功能控制寄存器(SFR)被初始化。内核的系统调试部件寄存器(FPB, DWT, 与ITM)，时钟发生器中的CGRSTFLG以及闪存相关寄存器中的FCSECBIT仅通过冷复位进行初始化。复位释放后，MCU根据内部高速振荡器的时钟开始运行。外部时钟和PLL多条电路可以按需设置。

5. 时钟/模式控制

5.1 特点

时钟/模式控制块用于选择时钟齿轮，预分频器时钟及PLL时钟乘法电路和振荡器的预热。

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。

该时钟/模式控制程序块具备以下功能：

- 控制系统时钟
- 控制预分频器时钟
- 控制PLL倍频电路；
- 控制预热定时器。

除NORMAL模式外，TMPM375FSDMG可根据使用条件，以低功率的六种类型进行操作，以减少功耗。

5.2 寄存器

5.2.1 寄存器列表

下表给出了与CG相关的各寄存器与地址。

基地址= 0x4004_0200

寄存器名称		地址(基+)
系统控制寄存器	CGSYSCR	0x0000
振荡控制寄存器	CGOSCCR	0x0004
待机控制寄存器	CGSTBYCR	0x0008
PLL选择寄存器	CGPLLSEL	0x000C
保留	-	0x0010

5.2.2 CGSYSCR (系统控制寄存器)

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

位	比特符号	型号	功能
31-18	-	R	读作"0",
17-16	-	R/W	写入"01".
15-13	-	R	读作"0",
12	FPSEL	R/W	选择fperiph源时钟 0: fgear 1: fc
11	-	R	读作"0",
10- 8	PRCK[2:0]	R/W	预分频时钟 000: fperiph 001: fperiph/2 010: fperiph/4 011: fperiph/8 100: fperiph/16 101: fperiph/32 110: 保留 111: 保留 将预分频器时钟指定给外设I/O。
7-3	-	R	读作"0",
2-0	GEAR[2:0]	R/W	高速时钟(fc)变速器 000: fc 001: 保留 010: 保留 011: 保留 100: (fc/2) 101: (fc/4) 110: (fc/8) 111: (fc/16)

5.2.3 CGOSCCR (振荡控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	WUODR							
复位后	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	WUODR				WUPSEL2	HOSCON	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	-	-	-	-	-	-	-	XEN1
复位后	0	0	0	0	0	0	0	0
	7	6	5	.	3	2	1	0
比特符号	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-20	WUODR[11:0]	R/W	设置预热定时器的计时时间。
19	WUPSEL2	R/W	预热定时器(WUP)的时钟源 0: 内部(OSC2) 1: 外部(OSC1) 预热定时器的时钟源可以选择外部振荡器(OSC1)或内部振荡器(OSC2)。
18	HOSCON	R/W	端口M或外部振荡器(X1/X2)(注) 0: 端口M 1: 外部振荡器(X1/X2)设置端口M或X1/X2。 当使用外部振荡器(DSC1)时, 端口M寄存器(PMCR/PMPUP/PMPDN/PMIE)应被禁用复位后, 端口M寄存器会被禁用。
17	OSCSEL	R/W	选择高速振荡器 0: 内部(OSC2) 1: 外部(OSC1) 高速振荡器可以选择外部振荡器(OSC1)或内部振荡器(OSC2)。确认<OSCSEL>变成"1", 然后在切换到OSC1后立即停止OSC2。切换到OSC1后请勿再次选择OSC2。
16	XEN2	R/W	高速振荡器2(内部) 0: 停止 1: 振荡 设置高速振荡器2(OSC2)的运行情况。
15-12	-	R/W	写作"0"。
11-10	-	R	读作"0",
9	-	R/W	写作"0"。
8	XEN1	R/W	高速振荡器1(外部) 0: 停止 1: 振荡 设置高速振荡器1(OSC1)的运行情况。
7-4	-	R/W	读作"0",
3	WUPSEL1	R/W	预热定时器的时钟源 写作"0"。
2	PLLON	R/W	PLL运行情况 0: 停止 1: 振动 设置的PLL运行情况。 复位后即停止, 需要对该为进行设置。

位	比特符号	型号	功能
1	WUEF	R	预热定时器(WUP)的 0: 预热完成 1: 预热进行 启用后监控预热定时器的状态。
0	WUEON	W	预热定时器的运行情况。 0: 忽略 1: 开始预热 启用以启动该预热计时器。 (注) 在采用自动预热进行从停止模式返回的设置时, 不要将"1"写入<WUEON>。通过软件(<WUOEN > = "1")进行预热时, 请监控 <WUEF>并确认预热是否完成。当<WUEN>变成 "0"时, 运行模式可以切换到停止模式。

注: 在将<HOSCON>设置为"1"时, 所有端口M的寄存器均不能进入, 从这些寄存器中读取的数据始终是"0"。如果端口M的某个寄存器(除PMDATA和PMOD)不等于初始值, 则<HOSCON>无法设置为 "1"。

5.2.4 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	-	-	-	-	-	-	-	RXEN
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	型号	功能
31-18	-	R	读作"0",
17	-	R/W	写作"0"。
16	DRVE	R/W	停止模式下的引脚状态 0: 无效 1: 有效
15-10	-	R	读作"0",
9	-	R/W	写作"0"。
8	RXEN	R/W	释放停止模式后的高速振荡器运行情况。 写作"1"。
7-3	-	R	读作"0",
2-0	STBY[2:0]	R/W	低功耗模式 000: 保留 001: 停止 010: 保留 011: IDLE 100: 保留 101: 保留 110: 保留 111: 保留 如需进入STOP模式, 必须先禁用不用于系统时钟的振荡 (OSC1或OSC2) 。

5.2.5 CGPLLSEL (PLL 选择寄存器)

	31	30	29	28	27	26	25	24	
比特符号	-	-	-	-	-	-	-	-	
复位后	0	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	16	
比特符号	-	-	-	-	-	-	-	-	
复位后	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	
比特符号	PLLSET1				-	PLLSET0			
复位后	1	1	0	1	0	0	0	0	
	7	6	5	.	3	2	1	0	
比特符号	PLLSET0							PLLSEL	
复位后	0	0	0	1	1	1	1	0	

位	比特符号	型号	功能
31-16	-	R	读作"0",
15-12	PLLSET1[3:0]	R/W	写入 "1101".
11	-	R	读作"0",
10-1	PLLSET0[9:0]	R/W	PLL多项设置 使用8MHz外部时钟时, 按如下设置写入。 5 PLL: "01_0001_0011". 使用10 MHz外部时钟时, 按如下设置写入。 4 PLL: "01_0000_1111". 注:除了以上设置外, 请勿设置其他值。
0	PLLSEL	R/W	PLL的使用 0: fosc使用 1: PLL的使用 规定使用或不用已被乘以PLL的时钟。 "fosc" 可在复位后自动进行设置。使用PLL时必须复位。

5.3 时钟控制

5.3.1 时钟类型

各时钟定义如下：

fosc1	:来自外部高速振荡器(X1和X2) 的时钟输入
fosc2	:来自内部高速振荡器的时钟输入
fosc	:CGOSCCR<OSCSEL>指定的高速时钟
f _{PLL}	:PLL(x 5或4) 增加8倍的时钟
f _c	:CGPLLSEL<PLLSEL>所规定的时钟 (高速时钟)
fgear	:CGSYSCR<GEAR[2:0]>指定的时钟
fsys	:与f变速器相同的时钟(系统时钟)
fperiph	:CGSYSCR<FPSEL>所规定的时钟
φT0	:CGSYSCR<PRCK[2:0]> 所规定的时钟(预分频器时钟)

高速时钟f_c和预定标时钟φT0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	:fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16, fperiph/32

5.3.2 复位之后的初始值

复位操作可初始化该时钟配置(如以下所述)。

高速振荡器 1(外部)	:停止
高速振荡器 2(内部)	:振荡
PLL(锁相环电路)	:停止
高速时钟齿轮	:f _c (无分频)

复位操作会导致所有的时钟配置与fosc2相同。

$$f_c = f_{osc2}$$

$$f_{sys} = f_c (= f_{osc2})$$

$$f_{periph} = f_c (= f_{osc2})$$

$$\phi T0 = f_{periph} (= f_{osc2})$$

5.3.3 时钟系统图

时钟系统如图5-1 所示。

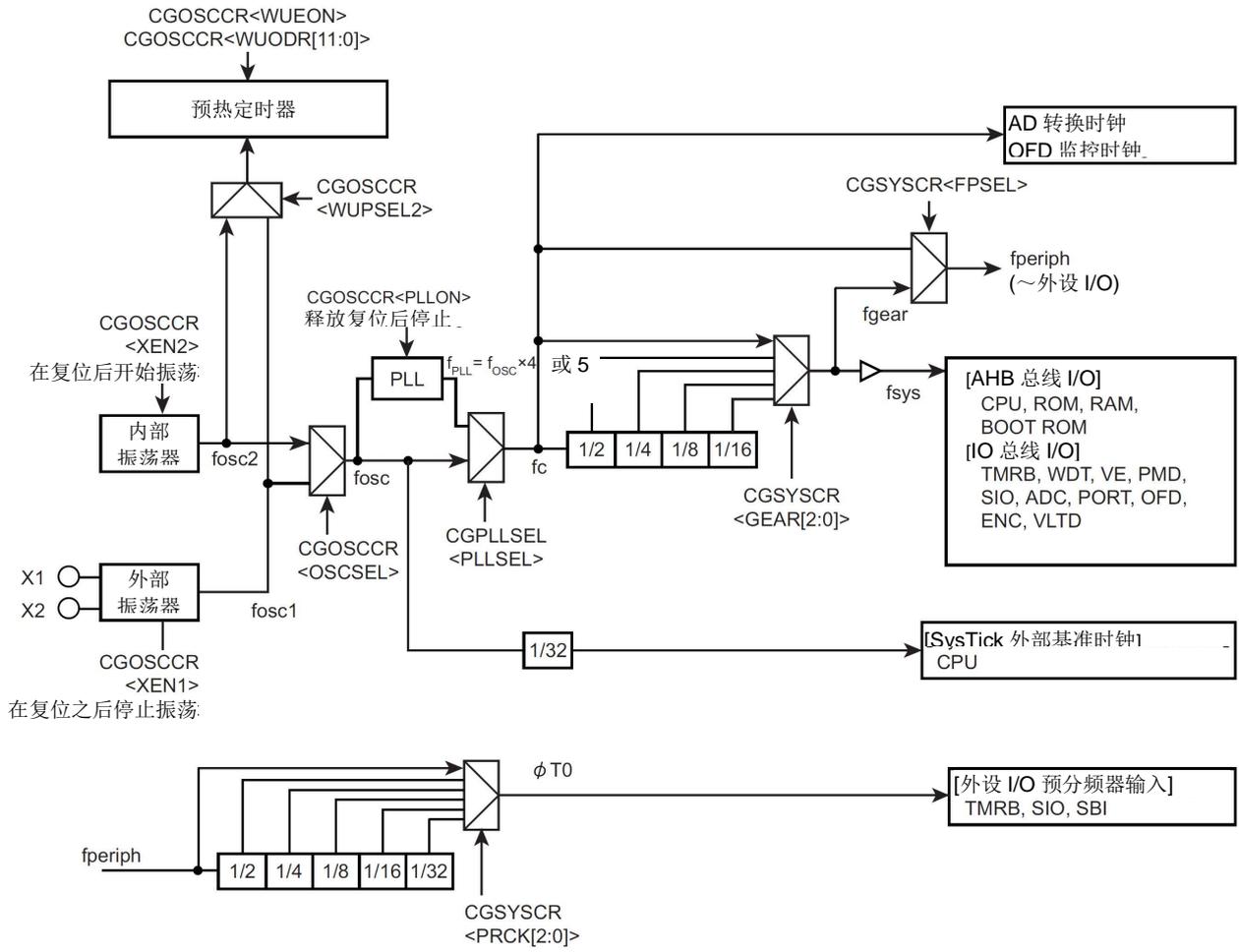


图5-1 时钟方块图

复位后图中箭头所示的输入时钟选择器设置为默认值。

5.3.4 时钟乘法电路(PLL)

电路能够输出 f_{PLL} 时钟，是高速振荡器输出时钟(f_{osc})的4倍或5倍。这样，振荡器的输入频率变低，同时内部时钟变为高速时钟。

在复位之后该PLL被禁用。如需启用PLL功能，将CGOSCCR<PLLON>位设置为"1"，并将CGPLLSEL<PLLSEL>位也设置为"1"。然后 f_{PLL} 时钟输出变成高速振荡器(f_{osc})的4倍或5倍。PLL需要一定的时间稳定下来，并应利用预热功能或其它方法予以确保。

5.3.4.1 稳定时间

PLL功能需要一定时间稳定，可通过预热功能或其他方法保证稳定时间。

当<PLLON>设置为"1"且开始运行后，需要大约200 μ s的锁定时间。

当乘数改变且PLL停止时<PLLON>首先被设置为"0"。当乘数<PLLSEL>改变时，大约需要100 μ s的PLL初始化时间，之后<PLLON>设置为"1"，进入PLL 状态。此时需要确保锁定时间作为PLL的稳定时间。

5.3.4.2 PLL 设置顺序

复位后的PLL设置顺序如下。

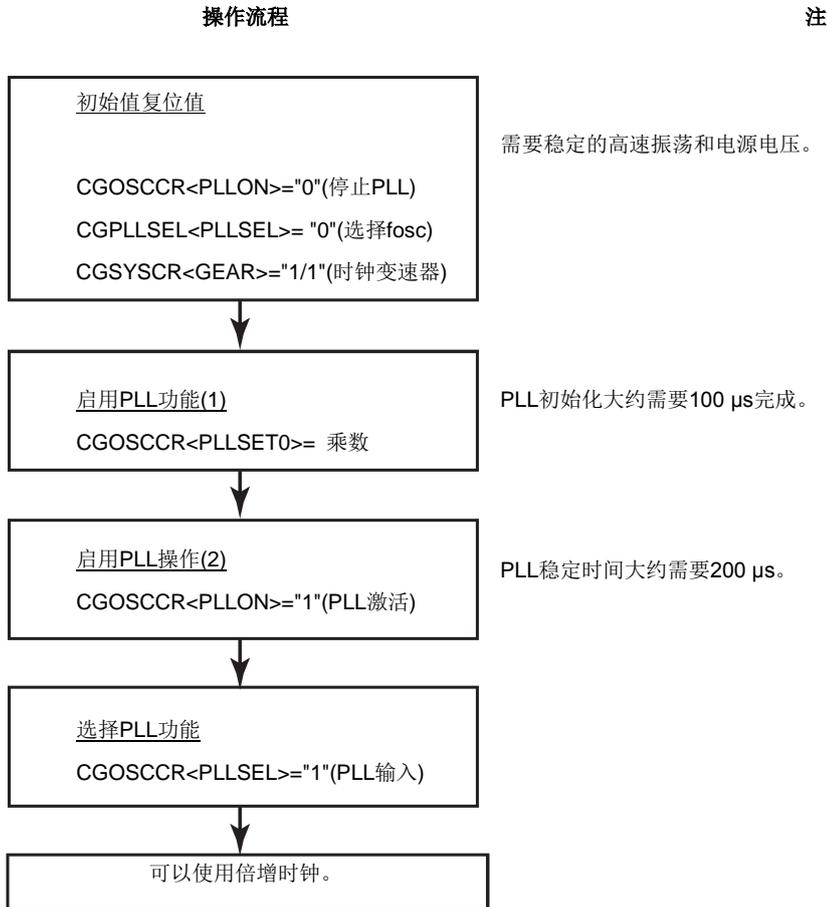


图5-2 复位后的PLL设置顺序

注：停止PLL功能时，请检查设置CGPLLSEL<PLLSEL>="0"之后的寄存器CGPLLSEL<PLLSEL>="0"。然后，请设置CGOSCCR<PLLON>="0"(停止PLL功能)。

注：设置PLL乘值后，需要将CGOSCCR <PLLON>="0"(PLL停止)保持100 μs以上，作为PLL初始化稳定时间。

5.3.5 预热功能

该预热功能可利用该预热计时器，固定该振荡器稳定时间与PLL。

从STOP模式返回时可以使用预热功能.关于此功能的详细信息，参见第5.6.6节"预热"。

注：在预热定时器运行的过程中请勿切换到STOP模式。

此时，从低功耗模式返回导致的中断会触发自动定时器计时。达到指定时间后，会输出系统时钟，同时CPU开始运行。

在STOP模式下，PLL被禁用。当从这些模式返回时，配置预热时间，并考虑到PLL与内振荡器的稳定时间。

如何配置该预热功能。

1. 规定递增计数时钟

在CGOSCCR<WUPSEL2>内为预热计数器设置递增计数时钟。

在CGOSCCR<WUPSEL1>和<WUPSEL2>位中为预热计数器设置递增计数时钟。将"0"写入<WUPSEL1>，将"0"或"1"写入<WUPSEL2>。"0"规定内部振荡器，"1"规定外部振荡器。

2. 规定预热计数器值

通过设置CGOSCCR<WUODR[11:0]>，可选择预热时间。

以下是预热设置的方法和举例。

$$\text{预热周期} = \frac{\text{设置预热时间}}{\text{输入周期(频率)}}$$

<例1>使用8 MHz振荡器设置5 ms的预热时间

设置预热时间	5 ms	= 40,000 个循环 = 0x9C40
输入周期(频率)	1/8 MHz	

丢弃最后4位，将0x9C4写入CGOSCCR<WUPT[11:0]>。

3. 确认预热开始与结束

通过软件(指令)，利用CGOSCCR<WUEON><WUEF>确认预热的开始与完成。

注：预热定时器按照振荡时钟运行。若振荡频率波动，它可能会包含错误。因此，该预热时间应被视为近似时间。

预热设置如下。

<例>确保PLL稳定时间($f_c = f_{osc1}$)

CGOSCCR<WUPSEL1> = "0" :将"0"写入CGOSCCR<WUPSEL1>

CGOSCCR<WUPSEL2> = "1" :设置预热定时器的时钟源

CGOSCCR<WUODR[11:0]> = "0x9C4" :预热时间设置

关于如何从内部振荡器切换到外部振荡器的步骤，参见第5.3.6节。

CGOSCCR<WUEON> = "1" :启用预热计数(WUP)

读取 CGOSCCR<WUEF> :保持等待，直至状态变为"0"(预热完成)

5.3.6 系统时钟

TMPM375FSDMG提供了高速时钟作为系统时钟，系统时钟可从内振荡器或外部振荡器中选择。复位后，内部振荡器被启用，而外部振荡器被禁用。高速时钟是可分割的时钟。

- 来自X1和X2的输入频率:8 MHz~10 MHz
- 内部振荡器频率:10 MHz
- 时钟变速器: 1/1, 1/2, 1/4, 1/8, 1/16(复位后: 1/1)

表5-1 高频范围(单位:MHz)

输入频率	最小工作频率	最大工作频率	复位后 (PLL = OFF, CG =1/1)	时钟变速器(CG): PLL = 开					时钟变速器(CG): PLL =关					
				1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16	
OSC1	8	1	40	8	40	20	10	5	2.5	8	4	2	1	-
	10			10	40	20	10	5	2.5	10	5	2.5	1.25	-
OSC2	10			10	40	20	10	5	2.5	10	5	2.5	1.25	-

注 1: PLL=ON/OFF设置:仅在CGOSCCR<PLLON>内可用。

注 2: 当某值写入CGSYSCR<GEAR[2:0]>寄存器时，会发生时钟变速器切换。实际的切换发生在短暂的延迟之后。

注 3: "-" :保留

注 4: 当使用SysTick时，请勿使用1/16。

从内部振荡器切换到外部振荡器的步骤如下：

1. 禁用端口M寄存器(PMCR /PMPUP /PMPDN /PMIE)。复位后，这些寄存器被禁用。
2. CGOSCCR<WUODR[11:0]> = "预热时间" :设置预热时间。
3. CGOSCCR<HOSCON> = "1" :从端口M切换到振荡器连接引脚。
4. CGOSCCR<XEN1> = "1" 启用外部振荡器。
5. CGOSCCR<WUPSEL2> = "1" :将外部振荡器时钟规定为预热计数器的源时钟。
6. CGOSCCR<WUEON>="1" :启用预热计数(WUP)
读取 CGOSCCR<WUEF> :保持等待，直至状态变为"0"(预热完成)
7. CGOSCCR<OSCSEL> = "1" :将系统时钟切换到外部振荡器。
8. 读取CGOSCCR<OSCSEL> :确认CGOSCCR[17]<OSCSEL>是否变为"1"。
(外部振荡器已选择。)
9. CGOSCCR<XEN2> = "0" :内部振荡器已禁用。

通过将CGOSCCR<HOSCON>设置为"1"，禁止对端口M的寄存器 (PMDATA/PMCR/PMOD/PMPUP/PMPDN/PMIE) 进行重写。

5.3.7 预分频器时钟控制

每个外设功能都有一个用于分割时钟的预分频器。由于时钟 $\phi T0$ 要输入至每个预分频器，CGSYSCR <FPSEL>中规定的"fperiph"时钟可根据CGSYSCR <PRCK [2:0]>中的设置进行划分。在控制器复位之后，"fperiph/1"即被选作 $\phi T0$ 。

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定标输出 ϕ 比fsys慢($\phi Tn < fsys$)。不要在定时器计数器或其它外设功能工作时切换时钟档。

5.4 模式与模式推移

5.4.1 模式推移

正常模式使用高速时钟作为系统时钟。

IDLE与STOP模式可用作低功耗模式，能通过暂停处理器内核运行来降低能耗。

模式转换如图 5-3 所示。

有关"退出时睡眠"的详细资料，请参看*Cortex-M3 技术参考手册*。

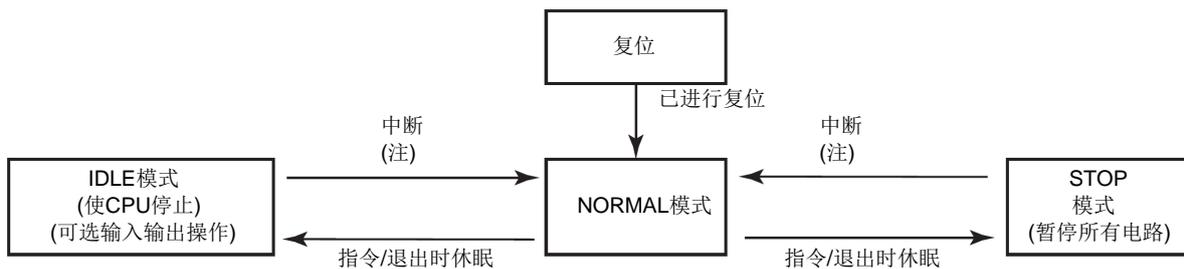


图 5-3 模式转换图

注：需进行预热。预热时间必须在切换到STOP模式之前在NORMAL模式下设置。关于预热时间的详细信息，参见第5.6.6节“预热”。

5.5 运行模式

NORMAL模式作为一种操作模式是可用的。NORMAL模式的特点在下节中叙述。

5.5.1 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。

5.6 低功耗模式

TMPM375FSDMG提供两种低功耗模式:IDLE和STOP。要转入低功耗模式,则规定系统控制寄存器CGSTBYCR<STBY [2:0]>中的模式,并执行WFI(等待中断)指令。在这种情况下,执行复位或生成中断,以解除该模式。通过中断来释放该模式需要提前设置,有关详细资料见"异常"一节。

注1: TMPM375FSDMG不提供释放低功耗模式的任何事件,并禁止通过执行WFE(等待事件)指令切换到低功耗模式。

注2: TMPM375FSDMG不支持Cortex-M3内核中配置有SLEEPDEEP位的低功耗模式.禁止设置该系统控制寄存器的<SLEEPDEEP>位。

各种模式的特性如下所述。

5.6.1 IDLE 模式

在该模式下,仅CPU被停止。各外设功能在其控制寄存器中有一位,用于在IDLE模式时启用或禁用操作。在进入IDLE模式时,在IDLE模式下其运行被禁止的各外设功能停止运行,并保持当时的状态。以下外设功能可在该IDLE模式下被启用或被禁用。有关设置的详细资料,见各外设功能相关节次。

- 16 位定时器/事件计数器(TMRB)
- 串行通道(SIO/UART)
- 串行总线接口(I2C/SIO)
- 看门狗定时器(WDT)
- 矢量引擎(VE+)

注:在进入IDLE模式之前应停止WDT。

5.6.2 STOP 模式

在该模式下，所有内部电路(包括内部振荡器)进入停止状态。

通过解除STOP模式，设备返回至STOP模式之前的模式，并开始操作。

STOP模式允许通过设置 CGSTBYCR<DRVE>选择引脚状态。停止模式下的引脚状态如表5-2 所示。

表 5-2 停止模式下的引脚状态

	引脚名称	I/O	<DRVE> = 0	<DRVE> = 1
非端口	$\overline{\text{RESET}}$ 模式	模式 仅输入	o	
	VOUT15, VOUT3	仅输出	o	
端口	x1	仅输入	x	
	x2	仅输出	"高"电平输出	
	TMS TCK TDI	输入	o	
	TDO/SWV	输出	被启用(在数据有效时)。 被禁用(在数据无效时)。	
	(SWCLK)	输入	o	
	SWDIO	输入	o	
		输出	被启用(在数据有效时)。 被禁用(在数据无效时)。	
	U0, V0, W0, X0, Y0, Z0	输出	被启用(在数据有效时)。 被禁用(在数据无效时)。	
	INT6, INT7, INTC	输入	o	
	除上述引脚或用作 通用端口的端口以 外的其它功能引脚	输入	x	o
输出		x	o	

o:输入或输出启用

x:输入或输出禁用

5.6.3 低功耗模式设置

通过设置备用控制寄存器 CGSTBYCR<STBY[2:0]>确定低功耗模式。
<STBY[2:0]>内的模式设置如表 5-3 所示。

表 5-3 低功耗模式设置

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

注：除上述<STBY[2:0]>中的各值外，不要设置其它任何值。

5.6.4 各模式下的操作状态

各种模式下的运行状态如表 5-4 所示。

表 5-4 各模式下的运行状态

模块	NORMAL	IDLE	停止
处理器内核	o	x	x
I/O 端口	o	o	*(注 1)
PMD	o	o	x
ENC	o	o	x
OFD	o	o	x
ADC	o	o	x
VE	o	各模块均可 选择 ON/OFF	x
SIO	o		x
SBI	o		x
TMRB	o		x
WDT	o		x
AMP	o	o	o(注 2)
VLTD	o	o	o(注 2)
POR	o	o	o(注 2)
DNF	o	o	x
CG	o	o	x
PLL	o	o	x
高速振荡器(fc)	o	o	x

o :操作

x :停止

注 1：由CGSTBYCR<DRVE>确定。

注 2：即使时钟停止，这些程序块也不会停止。

5.6.5 释放低功耗模式

低功耗模式可以通过中断请求，不可屏蔽中断(NMI)或复位释放。
或复位。可通过所选择的低功耗模式，确定可使用的释放源。
详细信息参见表 5-5。

表 5-5 各种模式下的释放源

低功耗模式		IDLE (可编程)	停止
释放源	中断	INT6, 7, C(注 1)	o
		INTRX0, 1, INTTX0, 1	o
		INTVCNB	o
		INTEMG1	o
		INTOVV1	o
		INTADBPDB	o
		INTTB00, 40, 50, 70	o
		INTTB01, 41, 51, 71	o
		INTPMD1	o
		INTCAP00, 50, 70	o
		INTCAP01, 51, 71	o
		INTADBCPA, INTADBCPB	o
		INTADBSFT	o
		INTADBTMR	o
		INTENC0	o
	INTSBIO	o	
	NMI(INTWDT)	o	
RESET ($\overline{\text{RESET}}$ 引脚)	o		

o :在模式被释放后，启动中断处理。(复位使LSI初始化)

x :不可用

注 1: 要利用等级模式中断解除低功耗模式，则保持该等级，直至中断处理开始。如果等级改变，则中断处理无法正常开始。

注 2: 如拟转入该低功耗模式，可设置CPU为禁止除该释放源以外的所有中断。否则会导致不明原因的唤醒释放。

注 3: 关于预热时间的信息，参见第5.6.6节"预热"。

通过中断请求释放

通过中断来释放低功耗模式时，必须事先设置CPU，以检测中断。除在CPU中设置外，必须设置时钟发生器，以检测用于解除STOP模式的中断。并禁用数字噪声过滤电路。

通过不可屏蔽中断(NMI)释放

有一个看门狗定时器中断(INTWDT)可以作为不可屏蔽中断源，INTWDT

仅可以在IDLE模式下使用。

注：注意在IDLE模式下不能通过CPU操作对WDT清除。

通过复位释放

低功耗模式可通过引脚的复位实现释放此后，模式切换到NORMAL模式，所有寄存器按正常复位的情况初始化。

通过SysTick中断释放

SysTick中断仅可在IDLE模式下使用。

详细信息参见"中断"一章。

5.6.6 预热

在进行模式推移时，可能会要求进行预热，以使得内部振荡器可提供稳定振荡。

在从STOP到NORMAL的模式转换时，预热计数器会被自动激活。然后经过配置的预热时间后开始系统时钟输出。需要通过CGOSCCR<WUPSEL1><WUPSEL2>设置一个振荡器用来预热(注 1)，并在执行指令进入STOP模式前通过CGOSCCR<WUODR>设置预热时间。

注1: CGOSCCR<WUPSEL1>应始终设置为 "0"。

注2: 在STOP模式下，PLL被禁用。当从这些模式返回后，配置预热时间需考虑PLL和内部振荡器所需的稳定时间。PLL稳定时间大约需要200 μ s。

注3: 在设置通过自动预热从低功耗模式返回时，请勿将"1"写入CGOSCCR<WUEON>位。

各个模式转换是否需要预热设置参见表 5-6 。

表 5-6 模式转换的预热设置

模式推移	预热设置
NORMAL → IDLE	不需要
NORMAL → STOP	不需要
IDLE → NORMAL	不需要
STOP → NORMAL	自动预热

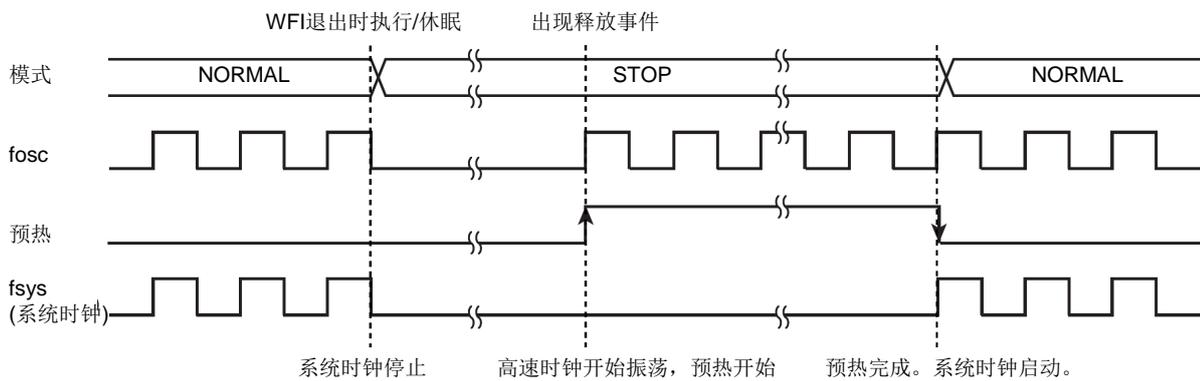
5.6.7 模式转换中的时钟运行情况

关于模式转换中的时钟运行情况，参见第 5.6.7.1 节。

5.6.7.1 运行模式的转换: NORMAL → STOP → NORMAL

当从STOP模式返回到NORMAL模式时，预热自动开启。在进入STOP模式前需要提前设置预热时间。

在通过复位返回到NORMAL模式时，不会引发自动预热。保持认定的复位信号，直至振荡器运行稳定下来。





6. 内部高速振荡调节功能

TMPM375FSDMG提供内部高速振荡调节的功能,

注: 该调节功能不适用于OFD的基准时钟。

6.1 结构

内部振荡调节功能使用16位定时器/事件计数器(TMRB)的脉冲宽度测量功能实现。
功能配置如图 6-1 所示。

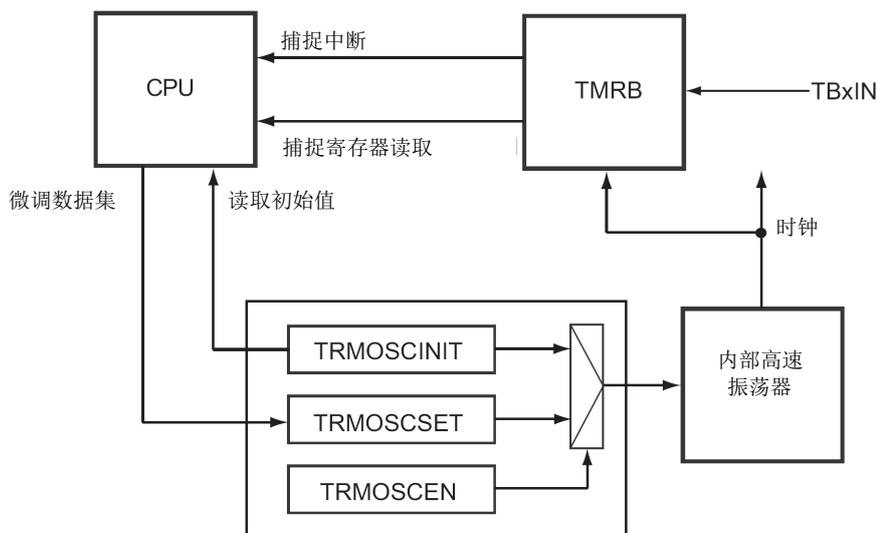


图 6-1 功能方块图

6.2 寄存器

6.2.1 寄存器列表

控制寄存器和相关地址如下。

基地址= 0x4004_0300

寄存器名称		地址(基+)
保护寄存器	TRMOSCPRO	0x0000
启用寄存器	TRMOSCEN	0x0004
初始微调值监控寄存器	TRMOSCINIT	0x0008
微调值设置寄存器	TRMOSCSET	0x000C

6.2.2 TRMOSCPRO(保护寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PROTECT							
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	PROTECT[7:0]	R/W	写寄存器控制 0xC1:启用 除0xC1以外:禁用 设置"0xC1"后, TRMOSCEN, TRMOSCINIT和TRMOSC设置允许写操作。

6.2.3 TRMOSCEN(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	-	-	-	-	-	-	-	TRIMEN
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	TRIMEN	R/W	微调控制 0: 禁用 1: 启用 当设置了"1"时, 内振荡器的调节值则从TRIMOSCINIT的某个值转换到TRMOSCSET的某个值。

6.2.4 TRMOSCINIT(初始修正值监视寄存器)

	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	-	-	TRIMINITC					
复位后	0	0	未定义					
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	TRIMINITF			
复位后	0	0	0	0	未定义			

位	比特符号	类型	功能
31-14	-	R	读作"0",
13-8	TRIMINITC [5:0]	R/W	初始粗调值 启用后监控初始粗调值。
7-4	-	R	读作"0",
3-0	TRIMINITF[3:0]	R/W	初始精调值 启用后监控初始精调值。

关于粗和精调的设置和调节值，参见表6-1 "调节范围"。

6.2.5 TRMOSCSET (修正值设置寄存器)

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

位	比特符号	类型	功能
31-14	-	R	读作"0",
13-8	TRIMSETC [5:0]	R/W	粗调值设置 设置粗调值。
7-4	-	R	读作"0",
3-0	TRIMSETF[4: 0]	R/W	精调值设置 设置精调值。

关于粗和精调的设置和调节值，参见表6-1 "调节范围"。

6.3 操作说明

6.3.1 概要

通过使用粗调值和精调值调节振荡值。

发货前的设置可以通过TRMOSCINIT<TRIMINITC>和<TRIMINITF>查看。当设置值改变时，将新值设置为TRMOSCSET<TRIMSETC>和<TRIMSETF>中。将"1"设置为TRMOSCEN <TRIMEN>，则内振荡器的某个设置值即改变。

注：复位后，禁止向TRMOSCSET和TRMOSCEN写入数据。当向这些位写入数据时，TRMOSCPRO<PROTECT>必须设置为"0xC1"。

6.3.2 调节范围

粗调时，可以1.8%的步长在-57.6% ~ +55.8%的范围内进行调节。精调时，可以0.3%的步长在-2.4% ~ +2.1% 的范围内进行调节。具体的调节范围如表6-1所示。

注：每个步长值是基于典型条件而假设的。在粗调的情况下，存在±0.2%的误差，在精调的情况下，存在±0.1%的误差。

表 6-1 调节范围

粗调		精调	
<TRIMSETC>	频率变化 (典型值)	<TRIMSETF>	频率变化 (典型值)
011111	+55.8%	0111	+2.1%
.	.	.	.
000001	+1.8%	0001	+0.3%
000000	±0%	0000	±0%
111111	-1.8%	1111	-0.3%
111110	-3.6%	1110	-0.6%
.	.	.	.
100000	-57.6%	1000	-2.4%

6.3.3 使用 TMRB 进行内部振荡频率测量

测量高速振荡器的频率时，可以使用TMRB的脉冲宽度测量功能。首先，选择一个内部振荡器作为TMRB的预分频器时钟ΦT0。其次，从TBxIN输入一个脉冲。然后，使用捕捉功能在脉冲的上升边捕捉一个上升计数器值。最后，通过比较捕捉值计算得出的TBxIN频率和真实频率之间的差值确定调节值。

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。如有必要，请参看Cortex-M3 技术参考手册。

7.1 概述

有两类异常：当某一错误条件发生时或者当产生异常的指令被执行时产生的异常；硬件产生的异常，例如外部引脚或外设功能的中断请求信号。

所有例外由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。

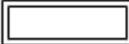
7.1.1 异常类型

Cortex-M3 存在以下异常类型。

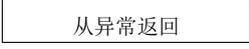
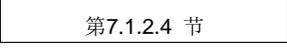
有关各异常的详细说明，请参看Cortex-M3 技术参考手册。

- 复位
- 不可屏蔽中断(NMI)
- 硬故障
- 存储器管理
- 总线故障
- 使用故障
- SVCALL(管理程序调用)
- 调试监视器
- 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。以下说明中，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。	 第7.1.2.1 节
		
 由CPU进行处理	CPU处理该异常请求。	 第7.1.2.2 节
		
 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。	 第7.1.2.4 节
		
 从异常返回	CPU转入到另一ISR，或返回到前一程序。	 第7.1.2.4 节

7.1.2.1 异常请求与检测

(1) 异常发生

异常的来源包括CPU的指令执行，存储器访问和外部中断引脚或外设功能的中断请求。

CPU在执行引起异常的指令时，或指令执行期间发生故障条件时发生异常。

由于从不执行(XN)区域的指令取出或故障区域的访问破坏，异常也会发生。

中断请求从外部中断引脚或外设功能中生成。对于用于解除待机模式的中断，必须在时钟生成器中进行相关设置。详情参见第 7.5 节"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。内存管理，总线故障和使用故障异常可以根据需要启用或禁用。如果发生某被禁用异常，其将被作为硬故障接受处理。

表 7-1 异常类型和优先级

序号	异常类型	优先级	说明
1	复位	-3 (最高)	复位引脚, WDT, POR, VLTD, OFD或SYSRETRREQ
2	非屏蔽中断	-2	WDT
3	硬故障	-1	由于较高优先级故障正被处理或者它被禁用, 无法激活的故障
4	存储器管理	可配置	存储器保护单元(MPU)异常 (注 1) 从不执行(XN)区域的指令取出
5	总线故障	可配置	对存储器映像硬故障区域的访问破坏
6	使用故障	可配置	未定义的指令执行或其他与指令执行有关的故障
7~10	保留	-	
11	SVCcall	可配置	用SVC指令进行系统服务调用
12	调试监视器	可配置	在CPU无故障时调试监视器
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 1: 本产品不含该MPU。

注 2: 各产品外部中断的源与数目各不相同。相关的详细信息, 参见第 7.5.1.5 节"中断源列表"。

(3) 优先级设置

优先级

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

可改变配置<PRI_n>，且优先级设置所需位的数目范围在 3 位到 8 位的范围内，具体范围视产品而定。因此，操作员可指定优先值的范围均视产品而定。

如果是 8 位配置，则优先级的可配置范围为 0~255。最高优先级为"0"。若存在具有相同优先级的多个要素，编号越小，优先级越高。

注：<PRI_n> 位被定义为本产品的 3 位配置。

优先级分组

优先级组可被分为若干组。通过设置应用中中断和复位控制寄存器的<PRIGROUP>，<PRI_n>可分成占先式优先级和次级优先级。

优先级与先占优先级类似。若优先级与占先式优先级相同，则它与次级优先级比较。如果子优先级与优先级相同，则异常数目越小，优先级就越高。

优先级分组的设置如表 7-2 所示。表中的占先式优先级和次级优先级是<PRI_n>被定义为 8 位配置情况下的编号。

表 7-2 优先级分组设置

<PRIGROUP[2:0]> 设置	<PRI_n[7:0]>		先占优先级的数目	子优先级的数目
	先占字段	子优先级字段		
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
111	无	[7:0]	1	256

注：如果<PRI_n>的配置小于 8 位，则较低的位为 "0"。例如，如果<PRI_n>采用 3 位配置，则优先级设置为 <PRI_n[7:5]>，<PRI_n[4:0]>为 "00000"。

7.1.2.2 异常处理与转入该中断服务程序(先占)

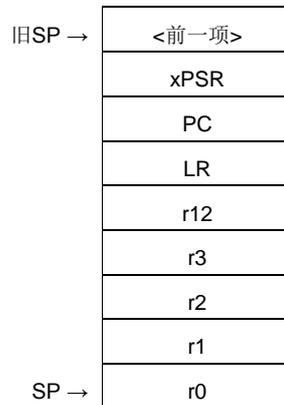
在异常发生时，CPU会中止当前正在执行的处理，并转到中断服务程序。这就是所谓的"先占"。

(1) 堆栈

在CPU检测到异常时，其会按以下顺序，将以下八个寄存器的内容压入到该堆栈：

程序计数器(PC)
 程序状态寄存器(xPSR)
 r0~r3
 r12
 链路寄存器 (LR)

到压栈完成时，SP渐减 8 个字。在寄存器内容已被压栈后，堆栈状态如下所示。



(2) 取出ISR

CPU启用指令以取出该中断处理，并将数据存储至寄存器。

为每个异常准备一个包含ISR顶部地址的向量表。复位后，该向量表指向代码区的地址0x0000_0000。通过设置向量表偏移寄存器，可以将向量表放置在代码或SRAM空间的任意地址。

该向量表还应包含该主堆栈的初始值。

(3) 迟来

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。

前4 个字(堆栈顶地址, 复位ISR地址, NMI ISR地址和硬件故障ISR地址)必须设置。对于其它异常, 可根据需要设置ISR地址。

偏移量	异常	内容	设置
0x00	复位	主堆栈的初始值	要求
0x04	复位	ISR地址	要求
0x08	非屏蔽中断	ISR地址	要求
0x0C	硬故障	ISR地址	要求
0x10	存储器管理	ISR地址	可选
0x14	总线故障	ISR地址	可选
0x18	使用故障	ISR地址	可选
0x1C~0x28	保留		
0x2C	SV调用	ISR地址	可选
0x30	调试监视器	ISR地址	可选
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

在从某ISR返回时，CPU会采取以下动作的其中之一：

尾链

如果存在一个挂起异常，且不存在任何堆栈异常，或挂起异常的优先级高于所有堆栈异常，则CPU会返回到挂起异常的ISR。

在这种情况下，CPU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。

返回到上一个程序

如果不存在待决或已堆栈异常，则CPU会返回前一程序。

(2) 异常出口顺序

在从某ISR返回时，CPU会执行以下操作：

读取8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SP_main或SP_process。

7.2 复位异常

复位异常的生成有以下6种来源。

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
在某外部复位引脚从"低"变为"高"时,会发生复位异常。
- POR复位异常
详细信息请参见"POR上电复位电路"一章。
- VLTD复位异常
详细信息请参见"VLTD电压检测电路"一章。
- OFD复位异常
详细信息请参见"OFD 振荡频率检测器"一章。
- 由WDT导致的复位异常
看门狗计时器(WDT)具备复位生成功能。有关详细资料见WDT相关节次。
- 由SYSRESETREQ导致的复位异常
通过在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)具备屏蔽中断生成功能。有关详细资料见WDT相关节次。

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

7.4 SysTick

SysTick具备中断功能(采用CPU的系统计时器)。

用户在SysTick重新加载值寄存器中设置某个值,并启用在SysTick控制和状态寄存器中的SysTick功能时,计数器会加载在重新加载值寄存器中设置的值,并开始倒数。在计数器达到"0"时,即发生SysTick异常。可选用挂起异常,并利用某个标志,以掌握计时器何时达到"0"。

SysTick校准值寄存器可保持某个重新加载值,且持续时间为10 ms(由系统计时器计时)。计数时钟频率随产品而变化,因此,在SysTick校准值寄存器中的设置值也会随产品而变化。

注:在本产品中,CGOSCCR <OSCSEL>基于32位选择的fosc用作外部基准时钟。

7.5 中断

本节对中断的路径，源与必要设置进行说明。

源自各中断源的中断信号可将中断请求通知CPU。

其可设置各中断的优先级，并处理具备最高优先级的中断请求。

可通过时钟发生器，将待机模式清除用中断请求通知CPU。因此，必须在时钟发生器中进行适当的设置。

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。

并非用于释放待机的外设功能所发布的中断，即被直接输入到CPU(路径 1)。

用于释放待机的外设功能(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，并通过待机释放逻辑(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释放待机，则其会被直接输入到CPU(不通过待机释放逻辑)(路径 6)。

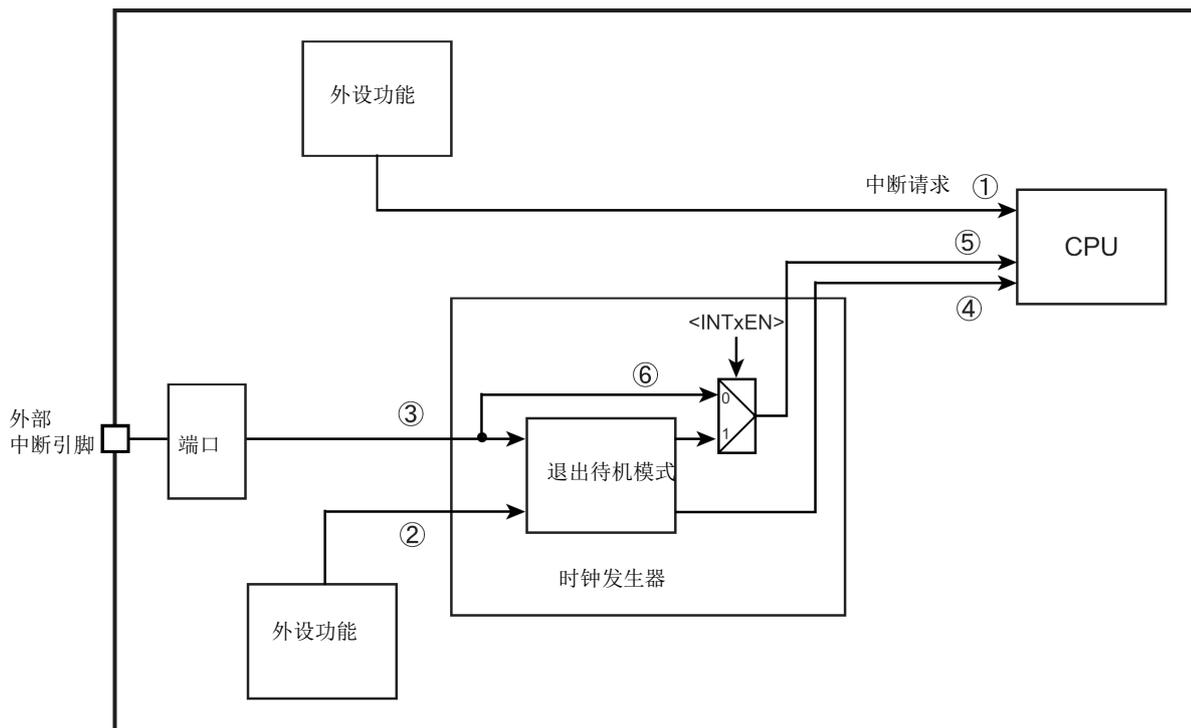


图 7-1 中断路径

7.5.1.2 生成

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

- 来自外部引脚
设置该端口控制寄存，使得该外部引脚可用作中断功能引脚。
- 来自外设功能
设置该外设功能，使之可输出中断请求。
有关详细资料见各外设功能相关节次。
- 通过设置中断挂起寄存器(强制挂起)
通过设置中断设置挂起寄存器的相关位，可生成一个中断请求。

7.5.1.3 传输

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

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

7.5.1.4 使用外部中断引脚时的预防措施

如果操作员使用外部中断，则应注意以下内容，以免生成意外中断。

如果输入被禁用($PxIE < PxIE < PxIE > = "0"$)，则源自外部中断引脚的输入为"高"。A同样，如外部中断不用作解除待机的触发器(图 7-1 的通路 6)，来自外部中断引脚的输入信号则被直接输送到CPU。由于CPU将"高"输入识别为中断，如果对应的中断因输入被禁用而被CPU启用，则就发生中断。将中断引脚输入设置为"低"并启用，即可在未将外部中断设置为待机触发信号的情况下使用该外部中断。即可启用CPU中断。

7.5.1.5 中断源列表

中断源列表如表 7-3 所示。

表 7-3 中断源列表

序号	中断源		作用电平 (清除备用模式)	CG中断模式 控制寄存器
0	INTRX0	串行接收(通道 0)		
1	INTTX0	串行发送(通道 0)		
2	INTRX1	串行接收(通道 1)		
3	INTTX1	串行发送(通道 1)		
4	INTVCNB	向量发动机中断B		
5	INTEMG1	PMD1 EMG中断		
6	INTOVV1	PMD1 OVV中断		
7	INTADBPDB	MD1触发的ADCB转换已结束		
8	INTTB00	16 位TMRB0比较匹配检测 0/溢出		
9	INTTB01	16 位TMRB0 比较匹配检测 1		
10	INTTB40	16 位TMRB4比较匹配检测 0/溢出		
11	INTTB41	16 位TMRB4 比较匹配检测 1		
12	INTTB50	16 位TMRB5比较匹配检测 0/溢出		
13	INTTB51	16 位TMRB5 比较匹配检测 1		
14	INTPMD1	PMD1 PWM中断		
15	INTCAP00	16 位TMRB0 输入捕捉 0		
16	INTCAP01	16 位TMRB0 输入捕捉 1		
17	INTCAP50	16 位TMRB5 输入捕捉 0		
18	INTCAP51	16 位TMRB5 输入捕捉 1		
19	INT6	中断引脚	高/低	CGIMCGA
20	INT7	中断引脚	边沿/电平可选	
21	INTADBCPA	ADCB转换监控功能中断A		
22	INTADBCPB	ADCB转换监控功能中断B		
23	INTADBSFT	软件开始的ADC单元B转换已结束		
24	INTADBTMR	定时器触发的ADC单元B转换已结束		
25	INTENC0	编码器输入0 中断		
26	INTTB70	16 位TMRB7 比较匹配检测 0/溢出		
27	INTTB71	16 位TMRB7 比较匹配检测 1		
28	INTCAP70	16 位TMRB7 输入捕捉 0		
29	INTCAP71	16 位TMRB7 输入捕捉 1		
30	INTC	中断引脚		
31	INTSBIO	串行总线接口		

7.5.1.6 作用电平

作用电平表明中断源信号的哪种改变会导致中断。CPU会再次组织"高"电平中的中断信号，将其作为中断信号直接从外设功能发送到CPU的中断信号被配置到输出"高"电平，指明中断请求。

作用电平，其可被用作待机释放用触发信号。源自外设功能的中断请求设置为上升沿或下降沿。源自中断引脚的中断请求可被设置为电平敏感型("高"或"低") 或边缘触发型(上升或下降)。

如果某中断源被用于清除待机模式，则必须设置相关时钟发生器寄存器。启用CGIMCGx<INTxEN> 位，并规定CGIMCGx<EMCGx> 位中的作用电平。必须为每个外设功能的中断请求设置作用电平，如表 7-3 所示。

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。

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。

处理	详情	参见
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">检测设置</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">发送中断信号的相关设置</p> </div>	<p>为检测中断设置相关NVIC寄存器。 若采用每个中断源来清除等待模式，则还应设置时钟发生器。</p> <ul style="list-style-type: none"> o 普通设置 <p>NVIC寄存器</p> <ul style="list-style-type: none"> o 设置清除备用模式 <p>时钟发生器</p> <p>执行适当的设置，以根据中断类型发送中断信号。</p> <ul style="list-style-type: none"> o 设置来自外部引脚的中断端口 o 设置来自外设功能的中断外设功能(有关详细资料见各外设功能相关节次)。 	<div style="border: 1px solid black; padding: 20px; width: 100%;"> <p>7.5.2.2 "准备"</p> </div>
<div style="border: 1px dashed black; padding: 2px; display: inline-block;">中断生成</div>	<p>生成中断请求。</p>	
<div style="display: flex; justify-content: space-around; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; border-radius: 10px; padding: 2px 5px;">不清除待机模式</div> <div style="font-size: 20px;">↓</div> <div style="border: 1px solid black; border-radius: 10px; padding: 2px 5px;">清除待机模式</div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">CG检测到中断 (清除待机模式)</p> </div>	<p>通过时钟发生器，将待机模式清除用中断线连接到CPU。</p>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>7.5.2.3 通过时钟生成器检测</p> </div>
<div style="border: 2px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">CPU检测到中断</p> </div>	<p>CPU检测到该中断。</p> <p>如果多个中断请求同时发生，则会按照优先次序检测到优先级最高的中断请求。</p>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>7.5.2.4 "CPU检测"</p> </div>
<div style="border: 2px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">CPU处理中断</p> </div>	<p>CPU处理该中断。</p> <p>在进入该ISR之前，CPU会将寄存器内容压入该堆栈。</p>	<div style="border: 1px solid black; padding: 10px; width: 100%;"> <p>"7.5.2.5 CPU 处理"</p> </div>

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 准备工作

当准备中断时，需要注意配置顺序，以避免意外中断的产生。

中断的执行及其配置的更改，总体上必须按照以下顺序进行。通过CPU禁用该中断。从CPU按最远的路径进行配置。然后通过CPU启用中断。

要对时钟发生器进行配置，必须依照本节所述的顺序，以免发生意外的中断。首先，配置预置条件。其次，清除时钟发生器中与该中断相关的数据，然后启用该中断。

以下各节均按中断处理的顺序列出，并将对其配置方法进行说明。

1. 通过CPU禁用中断
2. CPU寄存器设置
3. 配置 (1) (来自外部引脚的中断)
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PRI-MASK寄存器写入 "1"。所有中断与异常(非可屏蔽中断与硬故障除外)均会被屏蔽。

用"MSR"指令设置该寄存器。

中断屏蔽寄存器		
PRIMASK	←	"1" (中断停用)

注 1: 用户存取级不能修改PRIMASK寄存器。

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

(2) CPU寄存器设置

通过写入到NVIC寄存器的中断优先级寄存器中的<PRI_n>字段，操作员可指定某个优先级。每个中断源配有 8 个位，用于分配优先等级(从0 ~ 255)，但每种产品实际使用的位数会有差别。0 级是最优先的等级。若多个中断源的优先级相同，则最小编号的中断源的优先级最高。

通过应用中断和复位控制寄存器的<PRIGROUP>字段可以指定分组优先级。

NVIC寄存器		
<PRI_n>	←	"优先级"
<PRIGROUP>	←	"组优先级"(如有必要可配置)

注: "n" 指相应的异常/中断。

本产品有三个位用于指定优先级。

(3) 预配置 (1) (源自外部引脚的中断)

将 "1" 设置为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<PxmfFn>	←	"1"
PxIE<PxmlE>	←	"1"

注: x: 端口号 / m: 对应位 / n: 除STOP模式以外的其他模式下的功能寄存器号, 设置PxIE来启用输入, 这样无论PxFR设置如何都会开启对应的中断输入。注意不得启用未使用的中断。另外, 需注意 7.5.1.4 "外部中断引脚使用注意事项"。

(4) 预配置 (2) (源自外设功能的中断)

该设置随所用外设功能的不同而不同。有关外设功能的详细信息, 参见各外设功能对应的章节。

(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" 设置为该寄存器的对应位。

NVIC寄存器		
中断设置挂起[m]	←	"1"

注: m: 对应位

(6) 配置该时钟发生器

对于某个用于待机模式的中断源, 需要在时钟发生器的CGIMCGA寄存器中设置作用电平并启用中断。CGIMCGA寄存器能够配置每个中断源。

在启用某中断之前, 清除已保留的相应中断请求。这可避免意外的中断。要清除对应的中断请求, 写入与用于CGICRCG寄存器的中断相对应的值。各值见"7.6.3.2 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。另外，需注意 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>	←	作用电平
CGICRCG<ICRCG>	←	拟使用中断的对应值
CGIMCGn<INTmEN>	←	"1" (中断已启用)

注：n: 寄存器编号 / m: 中断源分配的编号

(7) 通过CPU启用中断

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。采用中断设置-启用寄存器启用预定的中断。将寄存器的各位指定到单个中断源。

将 "1" 写入到中断清除-挂起寄存器的对应位则清除该挂起中断。若在"中断设置—启用寄存器"的对应位中写入 "1"，则可启用相应的中断。

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

最后，PRIMASK寄存器被清0。

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

注 1: m: 对应位

注 2: 用户存取级不能修改PRIMASK寄存器。

7.5.2.3 由时钟发生器进行检测

如果用中断源退出待机模式，则会按照时钟发生器中所规定的作用电平进行检测，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。对于电平敏感型中断请求，在检测到该请求前，必须将其保持在作用电平，否则，当信号电平由有效变为无效时，中断请求将不存在。

当时钟发生器检测到中断请求时，会一直向CPU发生"高"电平的中断信号，直到CG中断请求(CGICRCG)寄存器中已将中断请求信号清除。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 由 CPU 进行检测

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运行期间进行压入

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核会自动将PC，PSR，r0-r3，r12和LR内容压倒堆栈中。无需对其进行额外编程。

如果需要，还可压入其它寄存器的内容。

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

(2) 清除某个中断源

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

如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。中断源的清除，会自动清除从时钟发生器发出的中断请求信号。

若某中断信号设置为边缘敏感型，可设置CGICRCG请求中的相应数值，从而清除中断请求。再次出现作用电平时，可检测到新的中断请求。

7.6 异常/中断相关寄存器

本章所述的CPU寄存器，NVIC寄存器和时钟发生寄存器如下所示(并附有相应的地址)。

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 1		0x0100
保留		0x0104
保留		0x0108
中断清除-启用寄存器 1		0x0180
保留		0x0184
保留		0x0188
中断设置-悬挂寄存器 1		0x0200
保留		0x0204
保留		0x0208
中断清除挂起寄存器 1		0x0280
保留		0x0284
保留		0x0288
中断优先级寄存器		0x0400 ~ 0x0460
矢量表位移寄存器		0x0D08
应用中断与复位控制寄存器		0x0D0C
系统处理器优先寄存器		0x0D18, 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= 0x4004_0200
寄存器名称		地址
CG中断请求清除寄存器	CGICRCG	0x0014
NMI标志寄存器	CGNMIFLG	0x0018
复位标志寄存器	CGRSTFLG	0x001C
CG中断模式控制寄存器A	CGIMCGA	0x0020

注：禁止访问各“保留”区域。

7.6.2 NVIC 寄存器

7.6.2.1 SysTick 控制器与状态寄存器

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

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

注: 在本产品中, CGOSCCR <OSCSEL>基于32 位选择的 f_{osc} 用作外部基准时钟。

7.6.2.2 SysTick 重新加载值寄存器

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

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

7.6.2.3 SysTick 修正值寄存器

	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	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TENMS							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TENMS							
复位后	0	0	0	0	1	0	0	1
	7	6	5	4	3	2	1	0
比特符号	TENMS							
复位后	1	1	0	0	0	1	0	0

位	比特符号	类型	功能
31	NOREF	R	0: 具备基准时钟 1: 无基准时钟
30	SKEW	R	0: 校准值是 10 ms。 1: 校准值不为10 ms。
29-24	-	R	读作0,
23-0	TENMS	R	校准值 外部基准时钟用于10 ms计时的重新加载值。(注)

注: 如果是重合多次, 则请使用<TENMS>-1。

7.6.2.5 中断设置-启用寄存器 1

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

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

注: 关于中断和中断编号的详细说明, 参见 7.5.1.5 "中断源列表"。

7.6.2.6 中断清除-启用寄存器 1

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

位	比特符号	类型	功能
31-0	CLRENA	R/W	中断号[31:0] [写入] 1: 禁用 [读取] 0: 禁用 1: 启用 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。 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。通过读取各位，即可查看相应中断的启用/禁用条件。

注：关于中断和中断编号的详细说明，参见 7.5.1.5 "中断源列表"。

7.6.2.7 中断设置-悬挂寄存器 1

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

位	比特符号	类型	功能
31-0	SETPEND	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：关于中断和中断编号的详细说明，参见 7.5.1.5 "中断源列表"。

7.6.2.8 中断清除挂起寄存器 1

	31	30	29	28	27	26	25	24
比特符号	CLRPEND (中断 31)	CLRPEND (中断 30)	CLRPEND (中断 29)	CLRPEND (中断 28)	CLRPEND (中断 27)	CLRPEND (中断 26)	CLRPEND (中断 25)	CLRPEND (中断 24)
复位后	未定义	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(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(中断 10)	CLRPEND(中(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。但写入 "1" 不会影响已使用的中断信号。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：关于中断和中断编号的详细说明，参见 7.5.1.5 "中断源列表"。

7.6.2.9 中断优先级寄存器

各中断均具备中断优先级寄存器的八个位。

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	

各产品拟用于优先级指定的位数各不相同。本产品有三个位用于指定优先级。

对于编号为0~3的中断信号，中断优先级寄存器的字段见下表。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
31-29	PRI_3	R/W	中断编号 3 的优先级
28-24	-	R	读作0,
23-21	PRI_2	R/W	中断编号 2 的优先级
20-16	-	R	读作0,
15-13	PRI_1	R/W	中断号 1 的优先级
12-8	-	R	读作0,
7-5	PRI_0	R/W	中断号 0 的优先级
4-0	-	R	读作0,

7.6.2.10 矢量表位移寄存器

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

位	比特符号	类型	功能
31-7	TBLOFF	R/W	偏移值 设置TBLOFF 中规定的间隔顶部偏移值。 该偏移必须基于表中异常数进行对准。即可用最小32 个字组的数列，用于多达16 个中断。对于更多的中断，必须通过查找到两者的下一个功率进行调准。
6-0	-	R	读作0。

7.6.2.11 应用中断与复位控制寄存器

	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	-	-	-	-	PRIGROUP		
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	SYSREST REQ	VECTCLR ACTIVE	VECTRES ET
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	VECTKEY (已写) / VECTKEYSTAT (已读)	R/W	寄存器键 [写入]写入到寄存器要求<VECTKEY>字段中的0x5FA[读取]读作"0xFA05"。
15	ENDIANESS	R/W	字节顺序: (注 1) 1: 从大到小 0: 小端字节顺序
14-11	-	R	读作0,
10-8	PRIGROUP	R/W	中断优先级分组 000: 先占优先级七位, 子优先级一位 001: 先占优先级六位, 子优先级二位 010: 先占优先级五位, 子优先级三位 011: 先占优先级四位, 子优先级四位 100: 先占优先级三位, 子优先级五位 101: 先占优先级二位, 子优先级六位 110: 先占优先级一位, 子优先级七位 111: 先占优先级无位, 子优先级八位 位组合旨在将该中断优先级寄存器<PRI_n>划分为先占优先级与子优先级。
7-3	-	R	读作0,
2	SYSREST REQ	R/W	系统复位请求 1=CPU输出一个 SYSRESETEQ 信号.(注 2)
1	VECTCLR ACTIVE	R/W	清除活动矢量位 1: 清除活动NMI, 故障与中断的所有状态信息。 0: 不清除。该位可自行清除。 应用程序负责重新启动堆叠。
0	VECTRESET	R/W	系统复位位 1: 复位系统 0: 不复位系统。 复位该系统, 但各调试部件(FPB, DWT与ITM)除外; 方法是设置"1", 并对该位进行清0。

注 1: 小端字节序是本产品的默认存储格式。

注 2: 在输出SYSRESETEQ时, 本产品会执行热复位<SYSRESETEQ>通过热复位清0。

7.6.2.12 系统处理器优先寄存器

各异常均具备系统处理器优先寄存器的八个位。

各例外情况对应的系统处理器优先级寄存器的地址，见下表。

	31	24 23	16 15	8 7	0
0xE000_ED18	PRI_7		PRI_6 (使用故障)	PRI_5 (BUSFAULT)	PRI_4 (存储器管理)
0xE000_ED1C	PRI_11 (SVCall)		PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)		PRI_14 (PendSV)	PRI_13	PRI_12 (调试监视器)

各产品拟用于优先级指定的位数各不相同。本产品有三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不起任何作用。

	31	30	29	28	27	26	25	24
比特符号	PRI_7			-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	PRI_6			-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	PRI_5			-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PRI_4			-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-29	PRI_7	R/W	保留
28-24	-	R	读作0,
23-21	PRI_6	R/W	使用故障的优先级
20-16	-	R	读作0,
15-13	PRI_5	R/W	总线故障的优先级
12-8	-	R	读作0,
7-5	PRI_4	R/W	存储器管理的优先级
4-0	-	R	读作0,

7.6.2.13 系统处理器控制器与状态寄存器

	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	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	SVCALL PENDE	BUSFAULT PENDE	MEMFAULT PENDE	USGFAULT PENDE	SYSTICKAC T	PENDSVAC T	-	MONITOR ACT
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-19	-	R	读作0,
18	USGFAULT ENA	R/W	使用故障 0: 禁用 1: 启用
17	BUSFAUL TENA	R/W	BUSFAULT 0: 禁用 1: 启用
16	MEMFAULT ENA	R/W	存储器管理 0: 禁用 1: 启用
15	SVCALL PENDE	R/W	SVCALL 0: 未挂起状态 1: 进入挂起状态
14	BUSFAULT PENDE	R/W	BUSFAULT 0: 未挂起状态 1: 进入挂起状态
13	MEMFAULT PENDE	R/W	存储器管理 0: 未挂起状态 1: 进入挂起状态
12	USGFAULT PENDE	R/W	使用故障 0: 未挂起状态 1: 进入挂起状态
11	SYSTICKACT	R/W	SysTick 0: 无效 1: 有效
10	PENDSVACT	R/W	PendSV 0: 无效 1: 有效
9	-	R	读作0,
8	MONITORACT	R/W	调试监视器 0: 无效 1: 有效
7	SVCALLACT	R/W	SVCALL 0: 无效 1: 有效
6-4	-	R	读作0,

位	比特符号	类型	功能
3	USGFAULT ACT	R/W	使用故障 0: 无效 1: 有效
2	-	R	读作0,
1	BUSFAULT ACT	R/W	BUSFAULT 0: 无效 1: 有效
0	MEMFAULT ACT	R/W	存储器管理 0: 无效 1: 有效

注：必须小心清除或设置激活位，因为清除与设置这些位并不修改堆栈内容。

7.6.3 时钟发生器寄存器

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	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	EMCG2			EMST2		-	INT2EN
复位后	0	0	1	0	0	0	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
31-23	-	R	读作0。
22-20	EMCG2[2:0]	R/W	对INTC备用件归零请求的作用电平设置。(101~111:禁止设置) 000: "低"级 001: "高"级 010: 下降沿 011: 上升沿 100: 两种边缘
19-18	EMST2[1:0]	R	INTC备用件归零请求的作用电平 00: - 01: 上升沿 10: 下降沿 11: 两种边缘
17	-	R	读作未定义。
16	INT2EN	R/W	INTC归零输入 0: 禁用 1: 启用
15	-	R	读作0。
14-12	EMCG1[2:0]	R/W	INT7待机清除请求的作用电平设置(101~111: 禁止设置) 000: "低"级 001: "高"级 010: 下降沿 011: 上升沿 100: 两种边缘
11-10	EMST1[1:0]	R	INT7待机清除请求的作用电平 00: - 01: 上升沿 10: 下降沿 11: 两种边缘
9	-	R	读作未定义。
8	INT1EN	R/W	INT7 归零输入 0: 禁用 1: 启用
7	-	R	读作0。

位	比特符号	类型	功能
6-4	EMCG0[2:0]	R/W	INT6 待机清除请求的作用电平设置(101~111:禁止设置) 000: "低"级 001: "高"级 010: 下降沿 011: 上升沿 100: 两种边缘
3-2	EMST0[1:0]	R	INT6 待机清除请求的作用电平 00: - 01: 上升沿 10: 下降沿 11: 两种边缘
1	-	R	读作未定义。
0	INT0EN	R/W	INT6 归零输入 0: 禁用 1: 启用

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

注2: 请首先指定供该边使用的位，然后再指定供<INTxEN>使用的位。自动禁止对其进行设置。

7.6.3.2 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: INT6 0_1000: 保留 0_0001: INT7 0_1001: 保留 0_0010: INTC 0_1010: 保留 0_0011: 保留 0_1011: 保留 0_0100: 保留 0_1100: 保留 0_0101: 保留 0_1101: 保留 0_0110: 保留 0_1110: 保留 0_0111: 保留 0_1111: 保留 0_0011 ~ 1_1111: 保留 读作0。

7.6.3.3 CGNMIFLG (NMI 标志寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	NMIFLG0
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	NMIFLG0	R	NMI源生成标志 0: 不适用 1: 从WDT中生成

注：当读取<NMIFLG>时，被清除到"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	-	-	-	-	-	-	-	-
上电复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
上电复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
上电复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	OFDRSTF	DBGRSTF	VLDRSTF	WDTRSTF	PINRSTF	PONRSTF
上电复位后	0	0	0	0	0	0	0	1

位	比特符号	类型	功能
31-6	-	R	读作0。
5	OFDRSTF	R/W	OFD复位标志 (注 1) 0: 写入"0" 1: 从OFD复位
4	DBGRSTF	R/W	调试复位标志 (注 1) 0: 写入"0" 1: 从SYSRESETREQ复位
3	VLDRSTF	R/W	VLTD复位标志 0: 写入"0" 1: 从VLTD复位
2	WDTRSTF	R/W	WDT复位标志 0: 写入"0" 1: 从WDT复位
1	PINRSTF	R/W	$\overline{\text{RESET}}$ 引脚标志 0: 写入"0" 1: 从 $\overline{\text{RESET}}$ 引脚复位
0	PONRSTF	R/W	上电标志 0: 写入"0" 1: 自上电复位复位

注1: 该标志表示由CPU的NVIC应用中与复位控制寄存器的SYSRESETREQ位所生成的复位。

注2: 此寄存器产品带有上电复位电路, 并且只能通过上电复位才能初始化。因此, 在上电后初始复位状态下即将"1"设置为被设置为<PONRSTF>位。注意该位并不由第二个以及随后的复位来设置, 而且该寄存器并不自动清除。写入"0"则寄存器清0。



8. 数字噪声滤波电路(DNF)

数字噪声消除器电路可消除来自外部中断引脚某些范围的输入信号噪音。

8.1 配置

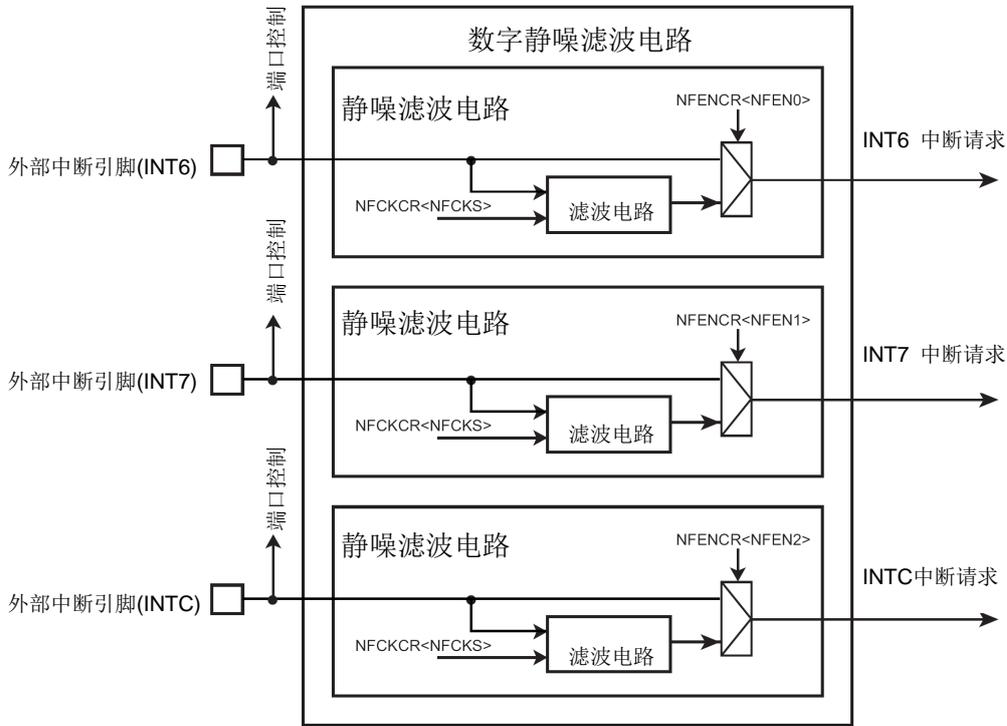


图 8-1 数字噪声滤波器电路图

8.2 寄存器

8.2.1 寄存器列表

基址 = 0x4006_0000

寄存器名称		地址(基+)
噪声滤波器控制寄存器	NFCKCR	0x0000
噪声滤波器启用寄存器	NFENCR	0x0004

8.2.1.1 NFCKCR (噪声滤波器控制寄存器)

	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	-	-	-	-	-	NFCKS		
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	NFCKS[2:0]	R/W	噪声滤波器时钟选择 000: 时钟控制电路停止 001: fsys/2 时钟输出 010: fsys/4 时钟输出 011: fsys/8 时钟输出 100: fsys/16 时钟输出 101: fsys/32 时钟输出 110: fsys/64 时钟输出 111: fsys/128时钟输出

注: NFENCR<NFEN[2:0]>="000"中规定了NFCKCR<NFCKS>设置值。

注: 当外部输入用于解除STOP模式时, 噪声滤波器电路不能使用。应确保禁用NFENCR寄存器的噪声滤波器启用位并通过NFCKCR寄存器停止该时钟。

8.2.1.2 NFENCR (噪声滤波器启用寄存器)

	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	-	-	-	-	-	NFEN2	NFEN1	NFEN0
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	NFENC	R/W	INTC 噪声滤波器已启用 0: 已禁用(预噪声滤波输出信号与噪声滤波器电路计数器在解除STOP模式时被清除。) 1: 启用 (后噪声滤波输出信号)
1	NFEN7	R/W	INT7 噪声滤波器已启用 0: 已禁用(预噪声滤波输出信号与噪声滤波器电路计数器在解除STOP模式时被清除。) 1: 启用 (后噪声滤波输出信号)
0	NFEN6	R/W	INT6 噪声滤波器已启用 0: 已禁用(预噪声滤波输出信号与噪声滤波器电路计数器在解除STOP模式时被清除。) 1: 启用 (后噪声滤波输出信号)

注: 有些短于 f_{sys} 的脉冲为不可能滤波的噪音。尤其是在 f_{sys} 频率低的情况下, 噪声滤波操作可能没有效。

注: 外部中断信号启用前, 清除中断事件, 然后设置要启用的NFENCR寄存器对应位。

注: 当外部输入用于解除STOP模式时, 噪声滤波器电路不能使用。应确保禁用NFENCR寄存器的噪声滤波器启用位并通过NFCKCR寄存器停止该时钟。

8.3 操作说明

8.3.1 配置

噪声滤波器电路由噪声滤波器电路与中断请求生成电路组成。

它消除来自外部输入的高电平或低电平噪声，然后CG检测上升/下降沿或信号电平(高或低)，以确定每个中断信号中的信号状态。

8.3.2 操作

噪声滤波器消除来自外部中断输入INTx的高与低电平噪声。

噪声滤波时间取决于NFCKCR<NFCKS>中规定的输入电平持续时间。当该时间不到 7 个时钟脉冲时，该输入被确定为噪音。当该时间超过 8 个时钟脉冲时，该输入被确定为无效信号。然而，确定某个 7~8 个时钟脉冲的输入信号因边沿定时而不同。

8.3.3 噪声滤波器可用操作模式

该噪声滤波器电路仅可用于NORMAL模式与IDLE模式。

8.3.4 STOP 模式使用注意事项

当使用STOP模式时，噪声滤波器电路因f_{sys}时钟的停止而不能使用。当外部输入用于解除STOP模式时，设置以下程序:设置要禁用的中断启用位；设置NFENCR寄存器的噪声滤波器启用/禁用位；停止NFCKCR寄存器的噪声滤波器时钟。

8.3.5 最短噪声滤波时间

当高电平或低电平输入继续输入超过NFCKCR寄存器中规定的 8 个时钟脉冲周期时，噪声滤波器电路决定发送外部中断信号的输入电平。

表 8-1 最短噪声滤波时间

NFCKCR<NFCKS>	f _{sys} [MHz]			单元
	20	32	40	
001	0.7	0.44	0.35	μs
010	1.4	0.88	0.7	
011	2.8	1.75	1.4	
100	5.6	3.5	2.8	
101	11.2	7.0	5.6	
110	22.4	14.0	11.2	
111	44.8	28.0	22.4	



9. 输入/输出端口

9.1 端口功能

9.1.1 功能列表

TMPM375FSDMG具有 21 个端口。除端口功能，这些端口还可被用作外设功能的I/O引脚。

表 9-1 显示端口功能表。

表 9-1 端口功能列表

端口	引脚	输入/输出 端口	上拉/ 下拉	Schmitt 输入	数字噪声滤 波器电路	可编程 开漏	功能引脚
PORTB							
	PB3	I/O	上拉/下拉	o	-	o	TMS / SWDIO, (RXD1)
	PB4	I/O	上拉/下拉	o	-	o	TCK / SWCLK, (TXD1)
	PB5	I/O	上拉/下拉	o	-	o	TDO / SWV, SCK0, (SDA0 / SO0)
	PB6	I/O	上拉/下拉	o	o	o	TDI, SCL0 / SI0, TB7OUT, INT6, RXD1
PORTE							
	PE0	I/O	上拉/下拉	o	-	o	TXD0, ENCA
	PE1	I/O	上拉/下拉	o	-	o	RXD0, TB4IN, ENCB
	PE2	I/O	上拉/下拉	o	o	o	SCLK0, $\overline{CTS0}$, ENCZ, INT7, (SCL0)
PORTF							
	PF0	I/O	上拉/下拉	o	o	o	TB7IN, SDA0 / SO0, \overline{BOOT} , INTC, TXD1
PORTG							
	PG0	I/O	上拉/下拉	o	-	o	U0
	PG1	I/O	上拉/下拉	o	-	o	X0
	PG2	I/O	上拉/下拉	o	-	o	V0
	PG3	I/O	上拉/下拉	o	-	o	Y0
	PG4	I/O	上拉/下拉	o	-	o	W0
	PG5	I/O	上拉/下拉	o	-	o	Z0
	PG6	I/O	上拉/下拉	o	-	o	\overline{EMG} , \overline{OVV}
PORTJ							
	PJ6	I/O	上拉/下拉	o	-	o	AINB9
	PJ7	I/O	上拉/下拉	o	-	o	AINB10
PORTK							
	PK0	I/O	上拉/下拉	o	-	o	AINB11
	PK1	I/O	上拉/下拉	o	-	o	AINB12
PORTM							
	PM0	I/O	上拉/下拉	o	-	o	x1
	PM1	I/O	上拉/下拉	o	-	o	x2

o: 存在

-: 不存在

9.1.2 端口寄存器概述

使用端口时，需配置下列寄存器。

- **PxDATA**: 端口x数据寄存器
读取/写入端口数据。
- **PxCR**: 端口x输出控制寄存器
控制输出。
控制输入时，需配置PxIE。
- **PxFRn**: 端口x功能寄存器n
设置功能。
通过设置"1"，就能激活指定的功能。
- **PxOD**: 端口x开漏控制寄存器
控制可编程开漏。
可编程开漏是通过设置PxOD而要实现假开漏功能。
当 PxOD 设置为"1"时，输出缓冲器被禁用，虚拟开漏极将实现。
- **PxPUP**: 端口x上拉控制寄存器
控制可编程上拉
- **PxPDN**: 端口x下拉控制寄存器
控制可编程下拉。
- **PxIE**: 端口x输入控制寄存器
控制输入。
为避免直通电流，默认设置禁止输入。

9.1.3 STOP 模式下端口状态

输入和输出在STOP模式时由CGSTBYCR<DRVE>位被启用。

如在停止模式下用<DRVE>=1启用PxIE或PxCR，则可相应启用输入或输出。若<DRVE> = 0，除一些端口外，即使PxIE或PxCR被启用，输入和输出在STOP模式时均禁用。

表 9-2 显示STOP模式中的引脚状态。

表 9-2 STOP模式中的端口状态

	引脚名称	I/O	<DRVE> = "0"	<DRVE> = "1"
非端口	RESET 模式	仅输入	o	
	VOUT15, VOUT3	仅输出	o	
端口	x1	仅输入	x	
	x2	仅输出	"高"电平输出	
	TMS TCK TDI	输入	o	
	TDO / SWV	输出	被启用(在数据有效时)。 被禁用(在数据无效时)。	
	(SWCLK)	输入	o	
	(SWDIO)	输入	o	
		输出	被启用(在数据有效时)。 被禁用(在数据无效时)。	
	U0 V0 W0 X0 Y0 Z0	输出	被启用(在数据有效时)。 被禁用(在数据无效时)。	
	INT6, INT7, INTC	输入	o	
	除上述引脚或用作通用端口的端口以外的其它功能引脚	输入	x	o
输出		x	o	

o :输入或输出启用

x :输入或输出禁用

9.2 端口功能

本章说明端口寄存器详细情况。

本章仅说明"电路类型"的读取电路配置。详细电路图参阅"端口段等效电路示意图"一章。

9.2.1 Port B (PB3 ~ PB6)

端口B为通用 4 位输入/输出端口。对此端口，可以位为单元确定输入与输出。除通用输入/输出功能以外，端口B还执行调试界面功能，串行接口功能(SIO / UART)，外信号中断输入与 16 位定时器输出功能。

复位使PB 3, PB 4, PB 5, PB 6 初始化，以执行调试接口功能。

当PB3起TMS或SWDIO的作用时，输入，输出和上拉被启用。当PB4起TCK或SWCLK的作用时，输入和下拉被启用。

当PB5起TDO或SWV的作用时，输出被启用。当PB 6起到TDI功能时，则输入，上拉均被启用。

使用外部中断输入解除STOP模式时，在PBFR 4 中选择该功能，并在PBIE寄存器中启用输入。这些设置将激活中断输入，即使在STOP模式中的时钟/模式控制板块中的CGSTBYCR<DRVE> 位被设置为停止控制引脚

注：在非STOP模式时，若输入在PxIE中被启用，则不管PxFR寄存器的设置，中断输入仍会被启用。当给装置编程时，保证禁用不用的中断。

9.2.1.1 端口 B 寄存器

基址 = 0x4000_0040

寄存器名称		地址(基+)
端口B数据寄存器	PBDATA	0x0000
端口B 输出控制寄存器	PBCR	0x0004
端口B功能寄存器 1	PBFR1	0x0008
端口B功能寄存器 2	PBFR2	0x000C
端口B功能寄存器 3	PBFR3	0x0010
端口B功能寄存器 4	PBFR4	0x0014
端口B功能寄存器 5	PBFR5	0x0018
端口B开漏控制寄存器	PBOD	0x0028
端口B上拉控制寄存器	PBPUP	0x002C
端口B下拉控制寄存器	PBPDN	0x0030
端口B输入控制寄存器	PBIE	0x0038

9.2.1.2 PBDATA (端口 B 数据寄存器)

	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	-	PB6	PB5	PB4	PB3	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-3	PB6 ~ PB3	R/W	端口B数据寄存器
2-0	-	R	读作0。

9.2.1.3 PBCR (端口 B 输出控制寄存器)

	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	-	PB6C	PB5C	PB4C	PB3C	-	-	-
复位后	0	0	1	0	1	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-3	PB6C ~ PB3C	R/W	输出 0: 禁用 1: 启用
2-0	-	R	读作0。

9.2.1.4 PBFR1 (端口 B 功能寄存器 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	-	PB6F1	PB5F1	PB4F1	PB3F1	-	-	-
复位后	0	1	1	1	1	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PB6F1	R/W	0: 端口 1: TDI
5	PB5F1	R/W	0: 端口 1: TDO / SWV
4	PB4F1	R/W	0: 端口 1: TCK / SWCLK
3	PB3F1	R/W	0: 端口 1: TMS / SWDIO
2-0	-	R	读作0。

9.2.1.5 PBFR2 (端口 B 功能寄存器 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	-	PB6F2	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PB6F2	R/W	0: 端口 1: SCL0 / SI0 (注)
5-0	-	R	读作0。

注: PB6 引脚与PE2 引脚均指定有SCL0 功能。使用该功能时, 应设置如下。

SCL0	<PB6F2>	<PE2F5>
PB6	1	0
PE2	0	1

如果<PB6F2>与<PE2F5>被设置为"1", 则PB6对该功能有效。

9.2.1.6 PBFR3 (端口 B 功能寄存器 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	-	PB6F3	PB5F3	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PB6F3	R/W	0: 端口 1: TB7OUT
5	PB5F3	R/W	0: 端口 1: SCK0
4-0	-	R	读作0。

9.2.1.7 PBFR4 (端口 B 功能寄存器 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	-	PB6F4	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PB6F4	R/W	0: 端口 1: INT6
5-0	-	R	读作0。

9.2.1.8 PBFR5 (端口 B 功能寄存器 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	-	PB6F5	PB5F5	PB4F5	PB3F5	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PB6F5	R/W	0: 端口 1: RXD1 (注)
5	PB5F5	R/W	0: 端口 1: SDA0/SO0 (注)
4	PB4F5	R/W	0: 端口 1: TXD1 (注)
3	PB3F5	R/W	0: 端口 1: RXD1 (注)
2-0	-	R	读作0。

注: PB6 引脚与PB3 引脚均指定RXD1 功能, PB5 引脚与PF0 引脚都指定SDA0/SO0 功能, 此外, PB4 引脚与PF0 引脚也都指定TXD1 功能。使用这些功能时, 应设置如下。

RXD1	<PB6F5>	<PB3F5>
PB6	1	0
PB3	0	1
如果<PB6F5>与<PB3F5>被设置为"1", 则PB6 对该功能有效。		

SDA0/SO0	<PF0F2>	<PB5F5>
PF0	1	0
PB5	0	1
如果<PF0F2>与<PB5F5>均设置为"1", 则PF0 对该功能有效。		

TXD1	<PF0F5>	<PB4F5>
PF0	1	0
PB4	0	1
如果<PF0F5>与<PB4F5>均设置为"1", 则PF0 对该功能有效。		

9.2.1.9 PBOD (端口 B 开漏控制寄存器)

	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	-	PB6OD	PB5OD	PB4OD	PB3OD	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-3	PB6OD ~ PB3OD	R/W	0: CMOS 1: 开漏
2-0	-	R	读作0。

9.2.1.10 PBPUP (端口 B 上拉控制寄存器)

	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	-	PB6UP	PB5UP	PB4UP	PB3UP	-	-	-
复位后	0	1	0	0	1	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-3	PB6UP ~ PB3UP	R/W	上拉 0: 禁用 1: 启用
2-0	-	R	读作0。

9.2.1.11 PBPDN (端口 B 下拉控制寄存器)

	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	-	PB6DN	PB5DN	PB4DN	PB3DN	-	-	-
复位后	0	0	0	1	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-3	PB6DN ~ PB3DN	R/W	下拉 0: 禁用 1: 启用
2-0	-	R	读作0。

9.2.1.12 PBIE (端口 B 输入控制寄存器)

	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	-	PB6IE	PB5IE	PB4IE	PB3IE	-	-	-
复位后	0	1	0	1	1	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-3	PB6IE ~ PB3IE	R/W	输入 0: 禁用 1: 启用
2-0	-	R	读作0。

9.2.2 端口 E (PE0 ~ PE2)

端口E为通用，3 位输入/输出端口。可规定该端口的输入和输出，单元为位。除通用输入/输出功能以外，端口E还执行串行接口功能(SIO / UART)，串行总线接口功能(I2C / SIO)，编码器输入功能，外部信号中断输入与 16 位定时器输入功能。

复位使端口E的所有位均初始化，成为输入，输出，上拉与下拉均被禁用的通用端口。

端口E有几种类型的功能寄存器。当将端口E用作通用端口时，将一些寄存器的对应位设置为"0"。若不将端口E作为通用端口，则应将功能寄存器的对应位设置为"1"。对于一些功能寄存器不要同时将其设置为"1"。

使用外部中断输入解除STOP模式时，在PEFR4 中选择该功能，并在PEIE寄存器中启用输入。这些设置将激活中断输入，即使在STOP模式中的时钟/模式控制板块中的CGSTBYCR<DRVE> 位被设置为停止控制引脚

注：在非STOP模式时，若输入在PxIE中被启用，则不管PxFR寄存器的设置，中断输入仍会被启用。当给装置编程时，保证禁用不用的中断。

9.2.2.1 端口 E 寄存器

基址 = 0x4000_0100

寄存器名称		地址(基+)
端口E数据寄存器	PEDATA	0x0000
端口E输出控制寄存器	PECR	0x0004
端口E功能寄存器 1	PEFR1	0x0008
端口E功能寄存器 2	PEFR2	0x000C
端口E功能寄存器 3	PEFR3	0x0010
端口E功能寄存器 4	PEFR4	0x0014
端口E功能寄存器 5	PEFR5	0x0018
端口E开漏控制寄存器	PEOD	0x0028
端口E上拉控制寄存器	PEPUP	0x002C
端口E下拉控制寄存器	PEPDN	0x0030
端口E输入控制寄存器	PEIE	0x0038

9.2.2.2 PEDATA (端口 E 数据寄存器)

	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	-	-	-	-	-	PE2	PE1	PE0
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	PE2 ~ PE0	R/W	端口E数据寄存器

9.2.2.3 PECCR (端口 E 输出控制寄存器)

	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	-	-	-	-	-	PE2C	PE1C	PE0C
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	PE2C ~ PE0C	R/W	输出 0: 禁用 1: 启用

9.2.2.4 PEFR1 (端口 E 功能寄存器 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	-	-	-	-	-	PE2F1	PE1F1	PE0F1
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	PE2F1	R/W	0: 端口 1: SCLK0
1	PE1F1	R/W	0: 端口 1: RXD0
0	PE0F1	R/W	0: 端口 1: TXD0

9.2.2.5 PEFR2 (端口 E 功能寄存器 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	-	-	-	-	-	PE2F2	PE1F2	-
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	PE2F2	R/W	0: 端口 1: $\overline{\text{CTS0}}$
1	PE1F2	R/W	0: 端口 1: TB4IN
0	-	R	读作0。

9.2.2.6 PEFR3 (端口 E 功能寄存器 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	-	-	-	-	-	PE2F3	PE1F3	PE0F3
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	PE2F3	R/W	0: 端口 1: ENCZ
1	PE1F3	R/W	0: 端口 1: ENCB
0	PE0F3	R/W	0: 端口 1: ENCA

9.2.2.7 PEF4 (端口 E 功能寄存器 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	-	-	-	-	-	PE2F4	-	-
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	PE2F4	R/W	0: 端口 1: INT7
1-0	-	R	读作0。

9.2.2.8 PEF5 (端口 E 功能寄存器 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	-	-	-	-	-	PE2F5	-	-
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	PE2F5	R/W	0: 端口 1: SCL0 / SI0 (注)
1-0	-	R	读作0。

注: PB6 引脚与PE2 引脚均指定有SCL0 功能。使用该功能时, 应设置如下。

SCL0	<PB6F2>	<PE2F5>
PB6	1	0
PE2	0	1
如果<PB6F2>与<PE2F5>被设置为"1", 则PB6对该功能有效。		

9.2.2.9 PEOD (端口 E 开漏控制寄存器)

	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	-	-	-	-	-	PE2OD	PE1OD	PE0OD
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	PE2OD ~ PE0OD	R/W	0: CMOS 1: 开漏

9.2.2.10 PEPUP (端口 E 上拉控制寄存器)

	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	-	-	-	-	-	PE2UP	PE1UP	PE0UP
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	PE2UP ~ PE0UP	R/W	上拉 0: 禁用 1: 启用

9.2.2.11 PEPDN (端口 E 下拉控制寄存器)

	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	-	-	-	-	-	PE2DN	PE1DN	PE0DN
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	PE2DN ~ PE0DN	R/W	下拉 0: 禁用 1: 启用

9.2.2.12 PEIE (端口 E 输入控制寄存器)

	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	-	-	-	-	-	PE2IE	PE1IE	PE0IE
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	PE2IE ~ PE0IE	R/W	输入 0: 禁用 1: 启用

9.2.3 端口 F (PF0)

端口F为通用，1 位输入/输出端口。可规定该端口的输入和输出，单元为位。除通用输入/输出功能外，端口F还执行串行总线接口功能(I2C / SIO)，串行接口功能(UART)，16 位定时器输入功能与操作模式设置。

在某一复位信号处于"0"状态期间，PF0输入与上拉即被启用。在复位信号的上升沿，如PF0 为"1"，则设备进入单模式并从单片闪存启动。如PF0 为"0"，则设备进入单启动模式并从内部启动程序启动。单核启动模式详见"闪速存储器操作"。

复位使端口F功能寄存器的某个位初始化，成为输入，输出，上拉与下拉均被禁用的通用端口。

复位排序后，PF0 被禁用，输入与上拉均基于PFIE与PFPUP寄存器的初始设置。

使用外部中断输入解除STOP模式时，则在PFFR 4 中选择该功能，并在PFIE寄存器中启用输入。这些设置将激活中断输入，即使在STOP模式中的时钟/模式控制板块中的CGSTBYCR<DRVE> 位被设置为停止控制引脚

注：在非STOP模式时，若输入在PxIE中被启用，则不管PxFR寄存器的设置，中断输入仍会被启用。当给装置编程时，保证禁用不用的中断。

9.2.3.1 端口 F 寄存器

基址 = 0x4000_0140

寄存器名称		地址(基+)
端口F数据寄存器	PFDATA	0x0000
端口F输出控制寄存器	PFCR	0x0004
端口F功能寄存器 1	PFFR1	0x0008
端口F功能寄存器 2	PFFR2	0x000C
端口F功能寄存器 4	PFFR4	0x0014
端口F功能寄存器 5	PFFR5	0x0018
端口F开漏控制寄存器	PFOD	0x0028
端口F上拉控制寄存器	PFPUP	0x002C
端口F下拉控制寄存器	PFPDN	0x0030
端口F输入控制寄存器	PFIE	0x0038

9.2.3.2 PFDATA (端口 F 数据寄存器)

	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	-	-	-	-	-	-	-	PF0
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	PF0	R/W	端口F数据寄存器

9.2.3.3 PFCR (端口 F 输出控制寄存器)

	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	-	-	-	-	-	-	-	PF0C
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	PF0C	R/W	输出 0: 禁用 1: 启用

9.2.3.4 PFFR1 (端口 F 功能寄存器 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	-	-	-	-	-	-	-	PF0F1
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	PF0F1	R/W	0: 端口 1: TB7IN

9.2.3.5 PFFR2 (端口 F 功能寄存器 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	-	-	-	-	-	-	-	PF0F2
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	PF0F2	R/W	0: 端口 1: SDA0/SO0 (注)

注：PB5 引脚与PF0 引脚均指定SDA0/SO0 功能。使用该功能时，应设置如下。

SDA0/SO0	<PF0F2>	<PB5F5>
PF0	1	0
PB5	0	1

如果<PF0F2>与<PB5F5>均设置为"1"，则PF0 对该功能有效。

9.2.3.6 PFFR4 (端口 F 功能寄存器 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	-	-	-	-	-	-	-	PF0F4
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	PF0F4	R/W	0: 端口 1: INTC

9.2.3.7 PFFR5 (端口 F 功能寄存器 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	-	-	-	-	-	-	-	PF0F5
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	PF0F5	R/W	0: 端口 1: TXD1 (注)

注：PB4 引脚与PF0 引脚均指定TXD1 功能。使用该功能时，应设置如下。

TXD1	<PF0F5>	<PB4F5>
PF0	1	0
PB4	0	1
如果<PF0F5>与<PB4F5>均设置为"1"，则PF0对该功能有效。		

9.2.3.8 PFOD (端口 F 开漏控制寄存器)

	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	-	-	-	-	-	-	-	PF0OD
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	PF0OD	R/W	0: CMOS 1: 开漏

9.2.3.9 PFPUP (端口 F 上拉控制寄存器)

	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	-	-	-	-	-	-	-	PF0UP
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	PF0UP	R/W	上拉 0: 禁用 1: 启用

9.2.3.10 PFPDN (端口 F 下拉控制寄存器)

	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	-	-	-	-	-	-	-	PF0DN
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	PF0DN	R/W	下拉 0: 禁用 1: 启用

9.2.3.11 PFIE (端口 F 输入控制寄存器)

	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	-	-	-	-	-	-	-	PF0IE
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	PF0IE	R/W	输入 0: 禁用 1: 启用

9.2.4 端口 G (PG0 ~ PG6)

端口G为通用，7 位输入/输出端口。可规定该端口的输入和输出，单元为位。除通用输入/输出功能以外，端口G还执行用于三相马达控制(PMD)功能的输入/输出端口。

复位使端口G的所有位均初始化，成为输入，输出，上拉与下拉均被禁用的通用端口。

9.2.4.1 端口 G 寄存器

基址 = 0x4000_0180

寄存器名称		地址(基+)
端口G数据寄存器	PGDATA	0x0000
端口G输出控制寄存器	PGCR	0x0004
端口G功能寄存器 1	PGFR1	0x0008
端口G功能寄存器 2	PGFR2	0x000C
端口G开漏控制寄存器	PGOD	0x0028
端口G上拉控制寄存器	PGPUP	0x002C
端口G下拉控制寄存器	PGPDN	0x0030
端口G输入控制寄存器	PGIE	0x0038

9.2.4.2 PGDATA (端口 G 数据寄存器)

	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	-	PG6	PG5	PG4	PG3	PG2	PG1	PG0
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-0	PG6 ~ PG0	R/W	端口G数据寄存器

9.2.4.3 PGCR (端口 G 输出控制寄存器)

	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	-	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-0	PG6C ~ PG0C	R/W	输出 0: 禁用 1: 启用

9.2.4.4 PGFR1 (端口 G 功能寄存器 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	-	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PG6F1	R/W	0: 端口 1: $\overline{\text{EMG}}$
5	PG5F1	R/W	0: 端口 1: Z0
4	PG4F1	R/W	0: 端口 1: W0
3	PG3F1	R/W	0: 端口 1: Y0
2	PG2F1	R/W	0: 端口 1: V0
1	PG1F1	R/W	0: 端口 1: X0
0	PG0F1	R/W	0: 端口 1: U0

9.2.4.5 PGFR2 (端口 G 功能寄存器 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	-	PG6F2	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6	PG6F1	R/W	0: 端口 1: \overline{OVV}
5-0	-	R	读作0。

9.2.4.6 PGOD (端口 G 开漏控制寄存器)

	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	-	PG6OD	PG5OD	PG4OD	PG3OD	PG2OD	PG1OD	PG0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-0	PG6OD ~ PG0OD	R/W	0: CMOS 1: 开漏

9.2.4.7 PGPUP (端口 G 上拉控制寄存器)

	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	-	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-0	PG6UP ~ PG0UP	R/W	上拉 0: 禁用 1: 启用

9.2.4.8 PGPDN (端口 G 下拉控制寄存器)

	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	-	PG6DN	PG5DN	PG4DN	PG3DN	PG2DN	PG1DN	PG0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-0	PG6DN ~ PG0DN	R/W	下拉 0: 禁用 1: 启用

9.2.4.9 PGIE (端口 G 输入控制寄存器)

	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	-	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作0。
6-0	PG6IE ~ PG0IE	R/W	输入 0: 禁用 1: 启用

9.2.5 端口 J (PJ6 ~ PJ7)

端口J为通用，2 位输入/输出端口。可规定该端口的输入和输出，单元为位。除通用输入/输出功能以外，端口J还执行AD变流器的模拟输入。

复位使端口J的所有位均初始化，成为输入，输出，上拉与下拉均被禁用的通用端口。

注：除非将端口J的所有位用作模拟输入引脚，则转换精度可能降低。务必验证这不会对你的系统带来问题。

9.2.5.1 端口 J 寄存器

基址 = 0x4000_0240

寄存器名称		地址(基+)
端口J数据寄存器	PJDATA	0x0000
端口J输出控制寄存器	PJCR	0x0004
端口J开漏控制寄存器	PJOD	0x0028
端口J上拉控制寄存器	PJPUP	0x002C
端口J下拉控制寄存器	PJPDN	0x0030
端口J输入控制寄存器	PJIE	0x0038

9.2.5.2 PJDATA (端口 J 数据寄存器)

	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	PJ7	PJ6	-	-	-	-	-	-
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。
7-6	PJ7 ~ PJ6	R/W	端口J数据寄存器
5-0	-	R	读作0。

9.2.5.3 PJCR (端口 J 输出控制寄存器)

	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	PJ7C	PJ6C	-	-	-	-	-	-
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。
7-6	PJ7C ~ PJ6C	R/W	输出 0: 禁用 1: 启用
5-0	-	R	读作0。

9.2.5.4 PJOD (端口 J 开漏控制寄存器)

	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	PJ7OD	PJ6OD	-	-	-	-	-	-
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。
7-6	PJ7OD~ PJ6OD	R/W	0: CMOS 1: 开漏
5-0	-	R	读作0。

9.2.5.5 PJPUP (端口 J 上拉控制寄存器)

	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	PJ7UP	PJ6UP	-	-	-	-	-	-
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。
7-6	PJ7UP~ PJ6UP	R/W	上拉 0: 禁用 1: 启用
5-0	-	R	读作0。

9.2.5.6 PJPDN (端口 J 下拉控制寄存器)

	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	PJ7DN	PJ6DN	-	-	-	-	-	-
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。
7-6	PJ7DN-PJ6DN	R/W	下拉 0: 禁用 1: 启用
5-0	-	R	读作0。

9.2.5.7 PJIE (端口 J 输入控制寄存器)

	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	PJ7IE	PJ6IE	-	-	-	-	-	-
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。
7-6	PJ7IE ~ PJ6IE	R/W	输入 0: 禁用 1: 启用
5-0	-	R	读作0。

9.2.6 端口 K (PK0 ~ PK1)

端口K是一个通用的 2 位输入/输出端口。可规定该端口的输入和输出，单元为位。除通用输入/输出功能以外，端口K还执行AD变流器的模拟输入。

复位使端口K的所有位均初始化，成为输入，输出，上拉与下拉均被禁用的通用端口。

注：除非将端口K的所有位数用作模拟输入引脚，否则可降低转换精确度。验证这样做不会对系统造成问题。

9.2.6.1 端口 K 寄存器

基址 = 0x4000_0280

寄存器名称		地址(基+)
端口K数据寄存器	PKDATA	0x0000
端口K输出控制寄存器	PKCR	0x0004
端口K开漏控制寄存器	PKOD	0x0028
端口K上拉控制寄存器	PKPUP	0x002C
端口K下拉控制寄存器	PKPDN	0x0030
端口K输入控制寄存器	PKIE	0x0038

9.2.6.2 PKDATA (端口 K 数据寄存器)

	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	-	-	-	-	-	-	PK1	PK0
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。
1-0	PK1~PK0	R/W	端口K数据寄存器

9.2.6.3 PKCR (端口 K 输出控制寄存器)

	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	-	-	-				PK1C	PK0C
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。
1-0	PK1C ~ PK0C	R/W	输出 0: 禁用 1: 启用

9.2.6.4 PKOD (端口 K 开漏控制寄存器)

	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	-	-	-	-	-	-	PK1OD	PK0OD
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。
1-0	PK1OD ~ PK0OD	R/W	0: CMOS 1: 开漏

9.2.6.5 PKPUP (端口 K 上拉控制寄存器)

	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	-	-	-	-	-	-	PK1UP	PK0UP
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。
1-0	PK1UP ~ PK0UP	R/W	上拉 0: 禁用 1: 启用

9.2.6.6 PKPDN (端口 K 下拉控制寄存器)

	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	-	-	-	-	-	-	PK1DN	PK0DN
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。
1-0	PK1DN-PK0DN	R/W	下拉 0: 禁用 1: 启用

9.2.6.7 PKIE (端口 K 输入控制寄存器)

	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	-	-	-	-	-	-	PK1IE	PK0IE
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。
1-0	PK1IE-PK0IE	R/W	输入 0: 禁用 1: 启用

9.2.7 端口 M (PM0 ~ PM1)

端口M为通用，2 位输入/输出端口。可规定该端口的输入和输出，单元为位。除通用输入/输出功能以外，端口M通过CGOSCCR <HOSCON> = 1还执行高速振荡器 1 (X1 与X2)功能。

在成为CGOSCCR <HOSCON> = 1 时，端口M的每个寄存器均不能转换到写入。有关其用作外部高速振荡器接线端子时的步骤，参阅"系统时钟"一章。(注 1)

复位则使端口M的所有位均初始化，成为输入，输出，上拉与下拉均被禁用的通用端口。(注 2)

注 1: 如除PMDATA与PMOD之外的端口M寄存器之一与初始值不同，则CGOSCCR <HO-SCON>不能被设置为"1"。

注 2: 复位取消后选择的高速时钟为内置高速时钟。因此，它在初始状态中变成端口M。

9.2.7.1 端口 M 寄存器

基址 = 0x4000_0300

寄存器名称		地址(基+)
端口M数据寄存器	PMDATA	0x0000
端口M输出控制寄存器	PMCR	0x0004
端口M开漏控制寄存器	PMOD	0x0028
端口M上拉控制寄存器	PMPUP	0x002C
端口M下拉控制寄存器	PMPDN	0x0030
端口M输入控制寄存器	PMIE	0x0038

9.2.7.2 PMDATA (端口 M 数据寄存器)

	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	-	-	-	-	-	-	PM1	PM0
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。
1-0	PM1 ~ PM0	R/W	端口M数据寄存器

9.2.7.3 PMCR (端口 M 输出控制寄存器)

	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	-	-	-	-	-	-	PM1C	PM0C
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。
1-0	PM1C ~ PM0C	R/W	输出 0: 禁用 1: 启用

9.2.7.4 PMOD (端口 M 开漏控制寄存器)

	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	-	-	-	-	-	-	PM1OD	PM0OD
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。
1-0	PM1OD ~ PM0OD	R/W	0: CMOS 1: 开漏

9.2.7.5 PMPUP (端口 M 上拉控制寄存器)

	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	-	-	-	-	-	-	PM1UP	PM0UP
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。
1-0	PM1UP ~ PM0UP	R/W	上拉 0: 禁用 1: 启用

9.2.7.6 PMPDN (端口 M 下拉控制寄存器)

	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	-	-	-	-	-	-	PM1DN	PM0DN
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。
1-0	PM1DN ~ PM0DN	R/W	下拉 0: 禁用 1: 启用

9.2.7.7 PMIE (端口 M 输入控制寄存器)

	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	-	-	-	-	-	-	PM11E	PM01E
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。
1-0	PM11E ~ PM01E	R/W	输入 0: 禁用 1: 启用

9.3 端口方块图

9.3.1 端口类型

端口分类如下。有关各端口类型所对应的方块图，请参见后续几页。
图中虚线指“端口方块图”所述等效电路部分。

表 9-3 功能列表

类型	GP端口	功能	模拟	上拉	下拉	可编程开漏	注
FT1	I/O	I/O	-	R	-	o	
FT2	I/O	I/O	-	R	-	o	功能输出通过启用信号触发
FT3	I/O	I/O	-	R	-	o	功能输出通过启用信号触发
FT4	I/O	输入(int)	-	R	-	o	配置有噪声滤波器
FT5	I/O	I/O	o	R	-	-	
FT6	输出	输出	-	NoR	-	o	$\overline{\text{BOOT}}$ 复位期间输入被启用

int: 阻断输入

-: 不存在

o: 存在

R复位时强制禁用。

NoR: 不受复位影响

9.3.2 类型 FT 1

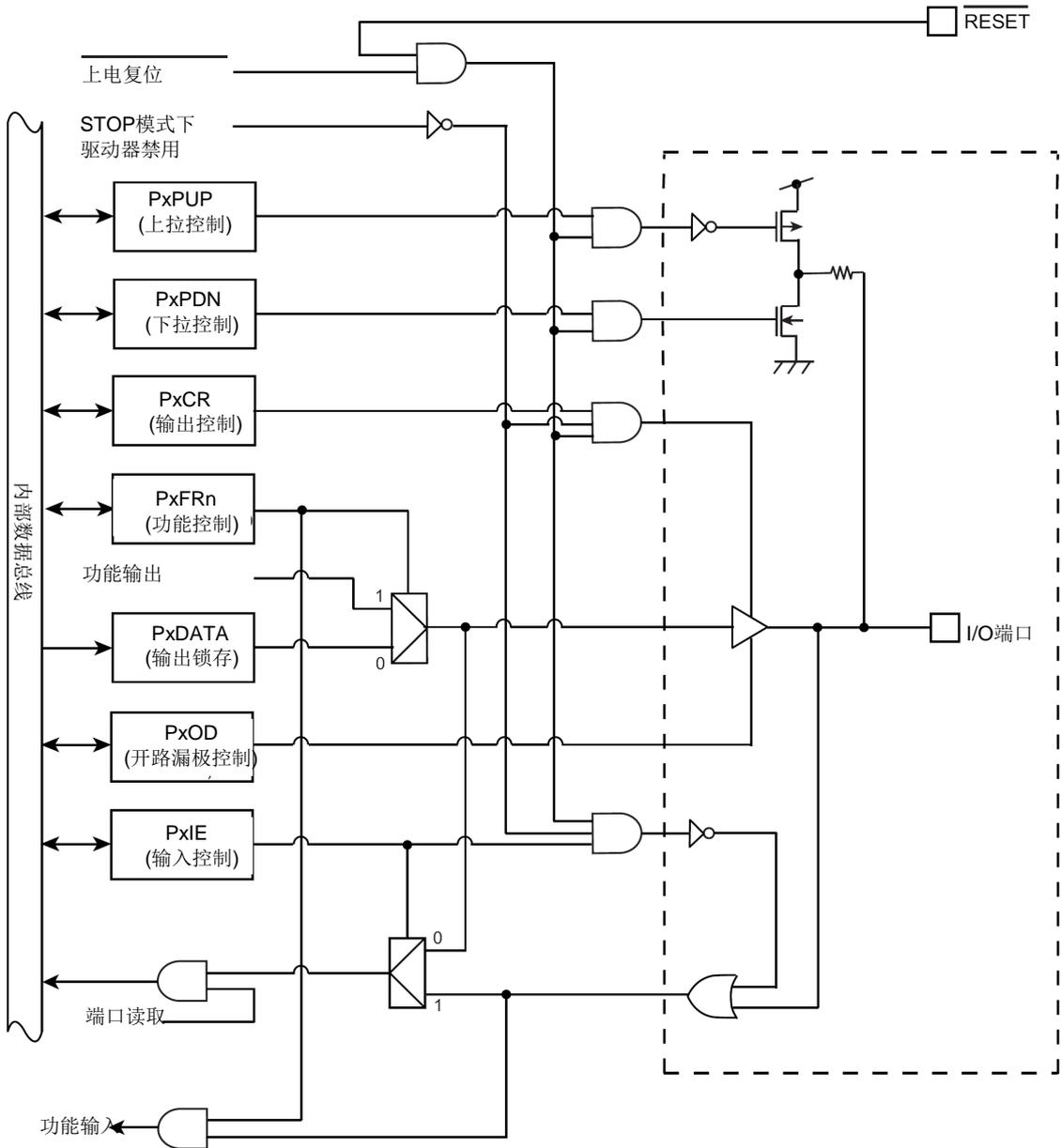


图 9-1 端口类型FT1

9.3.3 类型 FT2

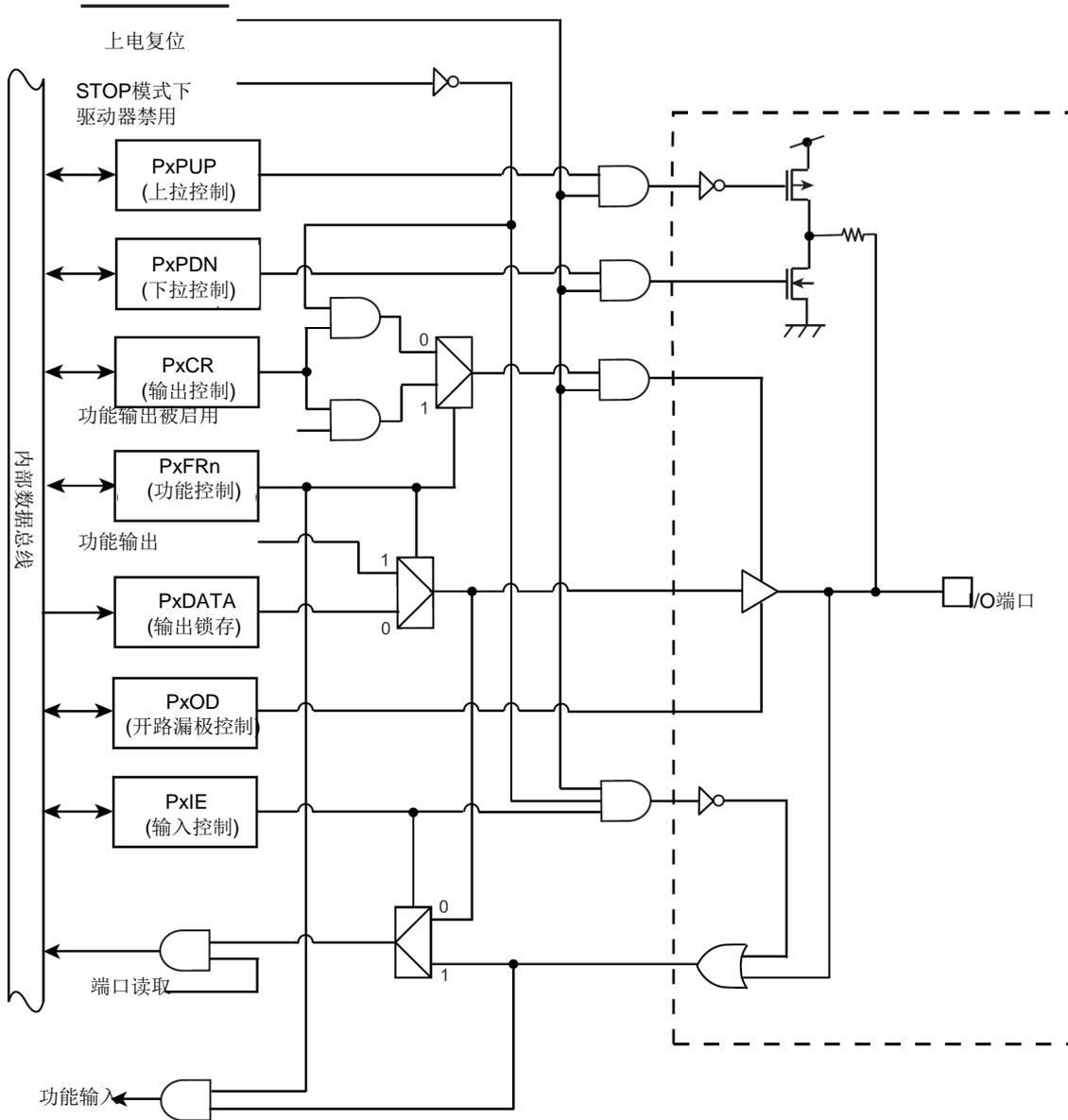


图 9-2 端口类型FT2

9.3.4 类型 FT3

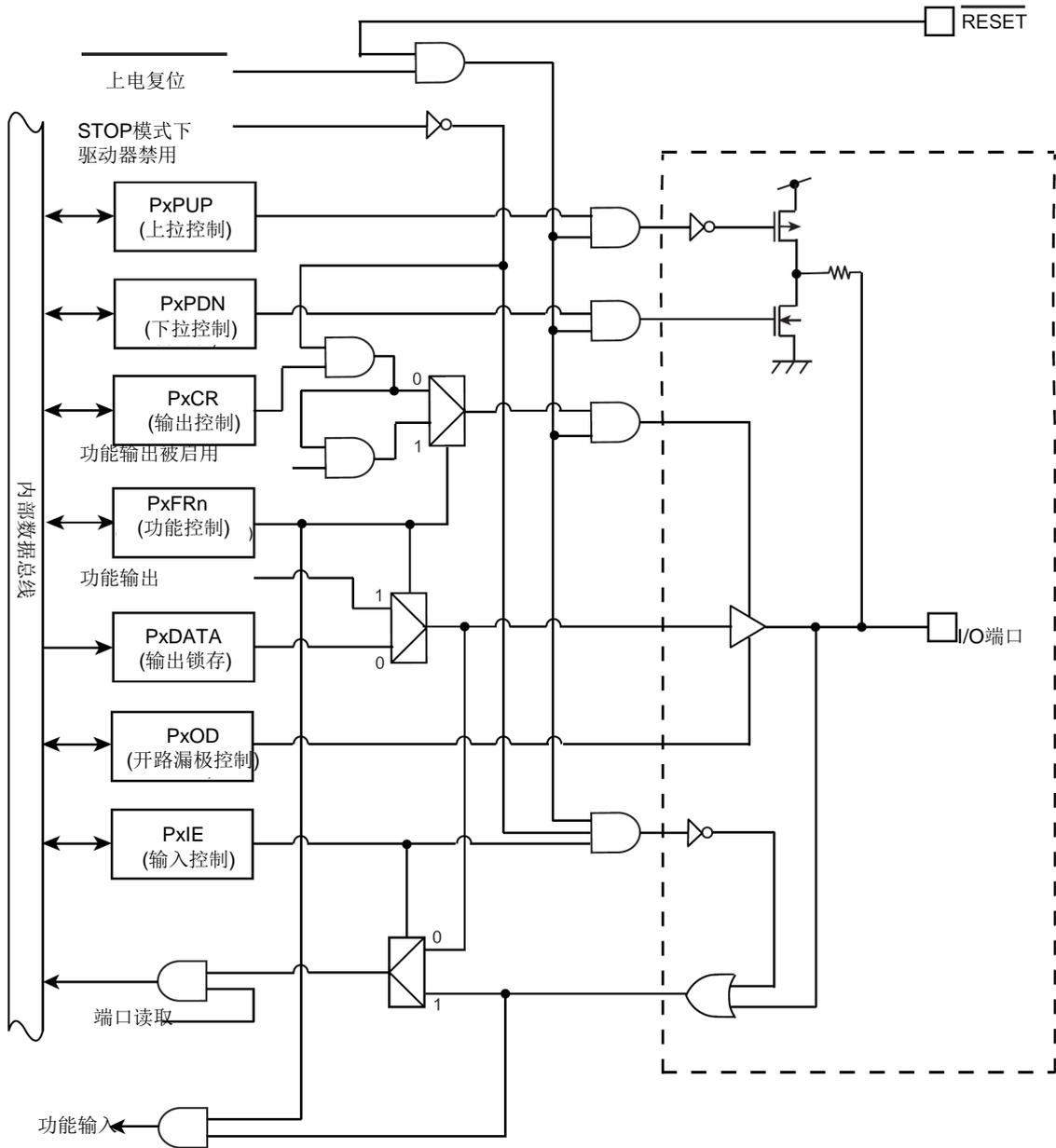


图 9-3 端口类型FT3

9.3.5 类型 FT4

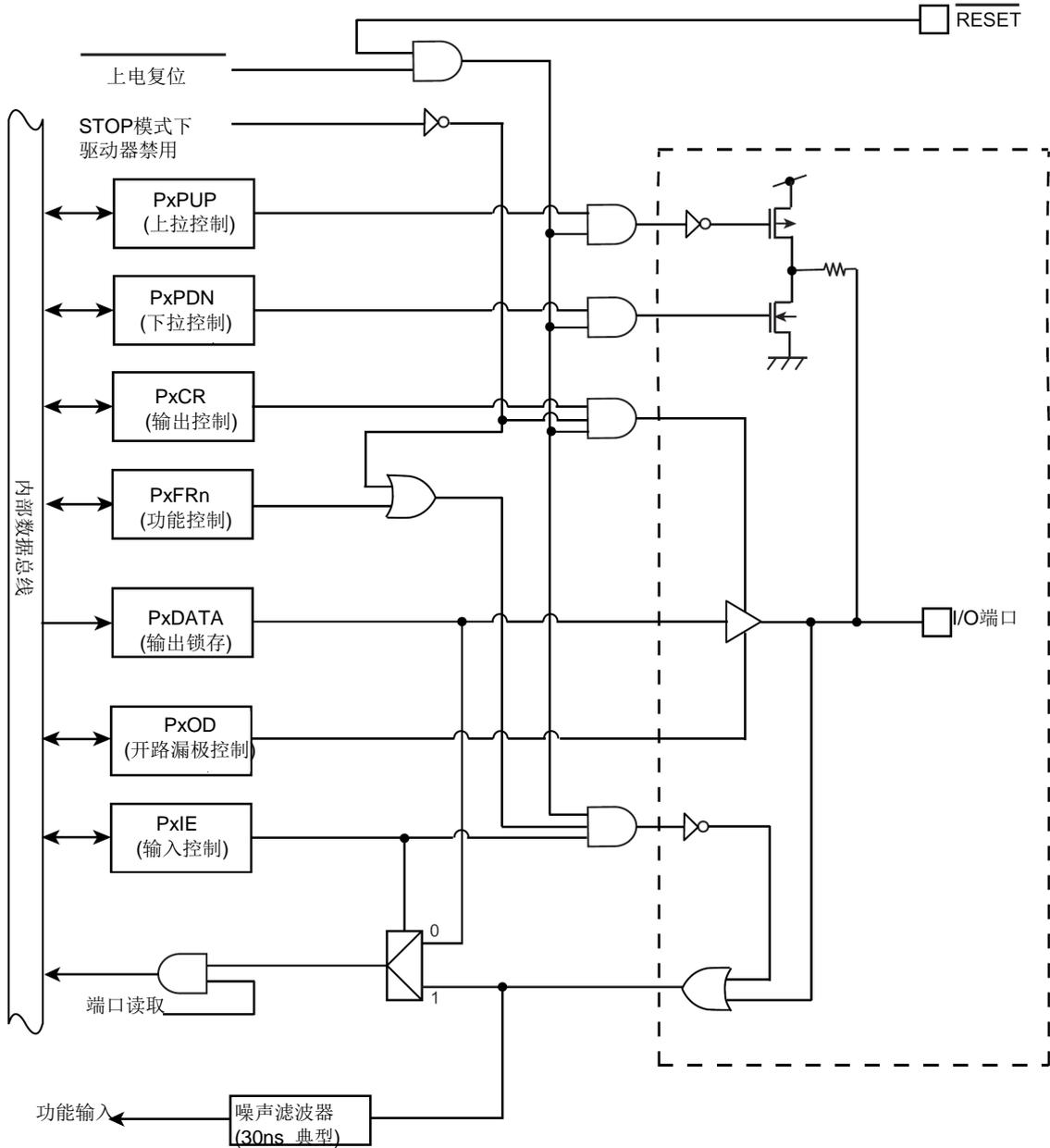


图 9-4 端口类型 FT4

9.3.6 类型 FT5

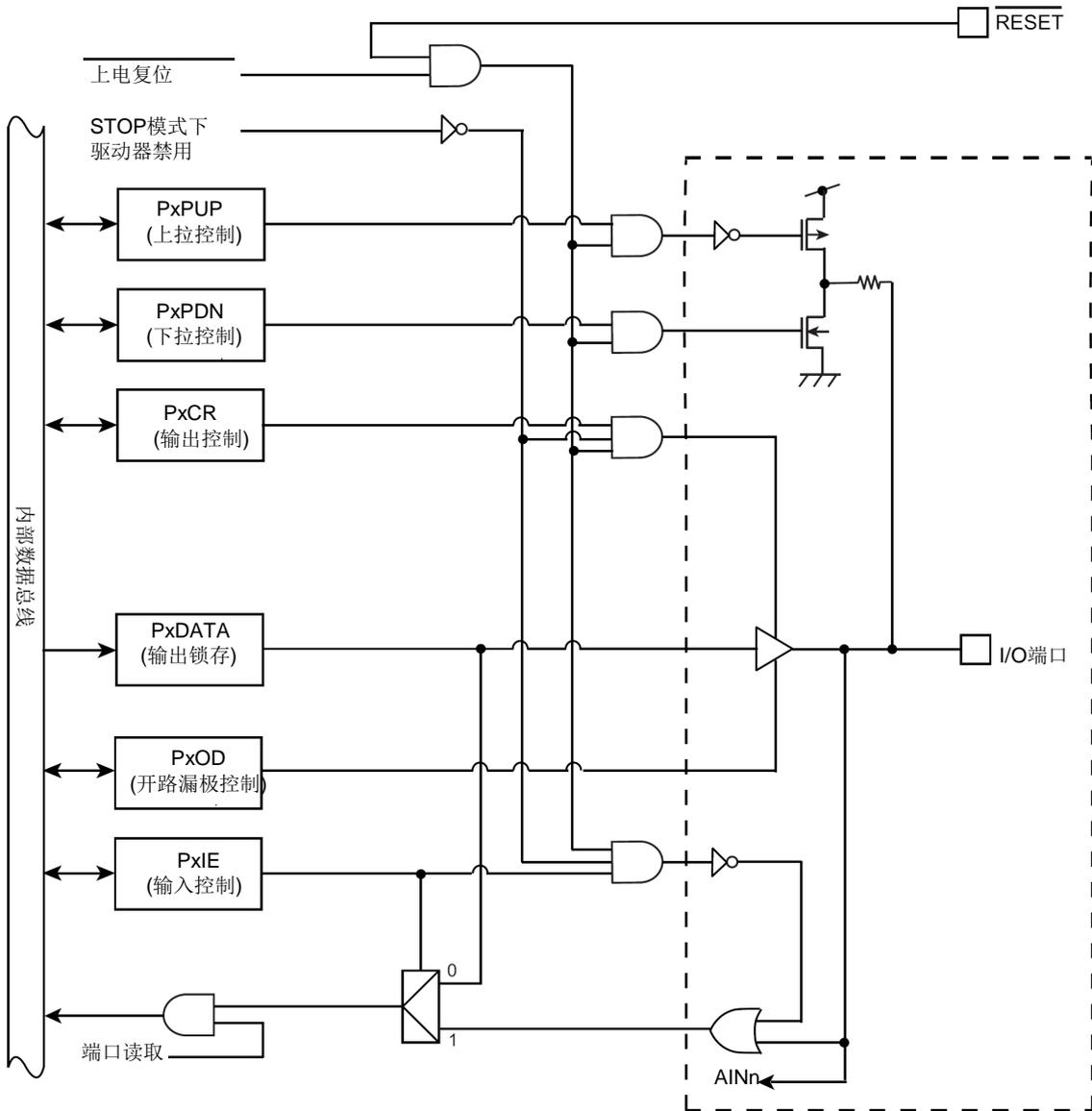


图 9-5 端口类型FT5

9.3.7 类型 FT6

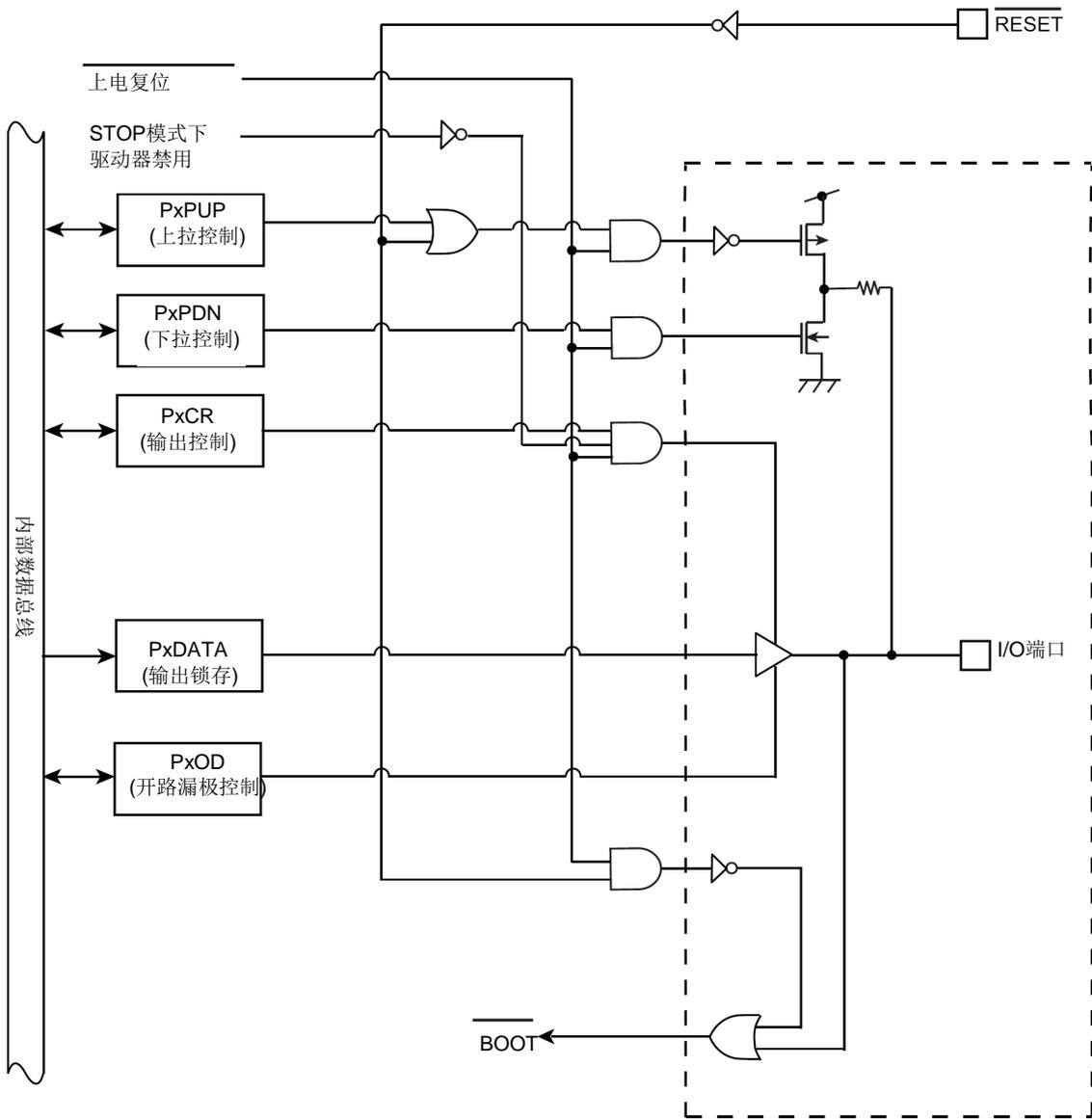


图 9-6 端口类型FT6

9.4 附录端口设置列表

下表显示了每个功能相应的寄存器设置。

对所有寄存器设置来说，"复位后"字段中不存在[·]的端口初始化均被设置为"0"。

"x"位的设置可任意指定。

9.4.1 端口 B 设置

表 9-4 端口设置列表(端口B)

引脚	端口类型	功能	复位后	PB CR	PB FR 1	PB FR 2	PB FR 3	PB FR 4	PB FR 5	PB OD	PB PUP	PB PDN	PB IE
PB3	-	输入端口		0	0	-	-	-	0	x	x	x	1
		输出端口		1	0	-	-	-	0	x	x	x	0
	FT2	TMS / SWDIO (I/O)	·	1	1	-	-	-	0	0	1	0	1
	FT1	RXD1(输入)		0	0	-	-	-	1	x	x	x	1
PB4	-	输入端口		0	0	-	-	-	0	x	x	x	1
		输出端口		1	0	-	-	-	0	x	x	x	0
	FT2	TCK / SWCLK (输入)	·	0	1	-	-	-	0	0	0	1	1
	FT1	TXD1(输出)		1	0	-	-	-	1	x	x	x	0
PB5	-	输入端口		0	0	-	0	-	0	x	x	x	1
		输出端口		1	0	-	0	-	0	x	x	x	0
	FT2	TDO / SWV (输出)	·	1	1	-	0	-	0	0	0	0	0
	FT1	SCK (I/O)		1	0	-	1	-	0	x	x	x	1
		SO0(输出)		1	0	-	0	-	1	x	x	x	0
	SDA0 (I/O)		1	0	-	0	-	1	x	x	x	1	
PB6	-	输入端口		0	0	0	0	0	0	x	x	x	1
		输出端口		1	0	0	0	0	0	x	x	x	0
	FT2	TDI (输入)	·	0	1	0	0	0	0	0	1	0	1
	FT1	SI0(输入)		0	0	1	0	0	0	x	x	x	1
		SCL0 (I/O)		1	0	1	0	0	0	x	x	x	1
		TB7OUT(输出)		1	0	0	1	0	0	x	x	x	0
	FT4	INT6(输入)		0	0	0	0	1	0	x	x	x	1
FT1	RXD1(输入)		0	0	0	0	0	1	x	x	x	1	

9.4.2 端口 E 设置

表 9-5 端口设置列表(端口E)

引脚	端口类型	功能	复位后	PE CR	PE FR 1	PE FR 2	PE FR 3	PE FR 4	PE FR 5	PE OD	PE PUP	PE PDN	PE IE
PE0	-	输入端口		0	0	-	0	-	-	x	x	x	1
		输出端口		1	0	-	0	-	-	x	x	x	0
	FT1	TXD0 (输出)		1	1	-	0	-	-	x	x	x	0
		ENCA(输入)		0	0	-	1	-	-	x	x	x	1
PE1	-	输入端口		0	0	0	0	-	-	x	x	x	1
		输出端口		1	0	0	0	-	-	x	x	x	0
	FT1	RXD0 (输入)		0	1	0	0	-	-	x	x	x	1
		TB4IN(输入)		0	0	1	0	-	-	x	x	x	1
		ENCB(输入)		0	0	0	1	-	-	x	x	x	1
PE2	-	输入端口		0	0	0	0	0	0	x	x	x	1
		输出端口		1	0	0	0	0	0	x	x	x	0
	FT1	SCLK0 (I / O)		1	1	0	0	0	0	x	x	x	1
		$\overline{\text{CTS0}}$ (输入)		0	0	1	0	0	0	x	x	x	1
		ENCZ(输入)		0	0	0	1	0	0	x	x	x	1
	FT4	INT7 (输入)		0	0	0	0	1	0	x	x	x	1
	FT1	SCL0 (I / O)		1	0	0	0	0	1	x	x	x	1

9.4.3 端口 F 设置

表 9-6 端口设置列表(端口F)

引脚	端口类型	功能	复位后	PF CR	PF FR 1	PF FR 2	PF FR 4	PF FR 5	PF OD	PF PUP	PF PDN	PF IE
PF0	-	输入端口		0	0	0	0	0	x	x	x	1
		输出端口		1	0	0	0	0	x	x	x	0
	FT1	TB7IN (输入)		0	1	0	0	0	x	x	x	1
		SO0(输出)		1	0	1	0	0	x	x	x	0
		SDA0 (I / O)		1	0	1	0	0	x	x	x	1
	FT4	INTC(输入)		0	0	0	1	0	x	x	x	1
	FT1	TXD1 (输出)		1	0	0	0	1	x	x	x	0

注：PF0 输入与上拉被启用，并在某个引脚处于“低”状态时充当输入引脚。在这种情况下，端口类型为FT6。

9.4.4 端口 G 设置

表 9-7 端口设置列表(端口G)

引脚	端口类型	功能	复位后	PG CR	PG FR 1	PG FR 2	PG OD	PG PUP	PG PDN	PG IE
PG0	-	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
	FT2	U0(输出)		1	1	-	x	x	x	0
PG1	-	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
	FT2	X0(输出)		1	1	-	x	x	x	0
PG2	-	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
	FT2	V0(输出)		1	1	-	x	x	x	0
PG3	-	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
	FT2	Y0(输出)		1	1	-	x	x	x	0
PG4	-	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
	FT2	W0(输出)		1	1	-	x	x	x	0
PG5	-	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
	FT2	Z0(输出)		1	1	-	x	x	x	0
PG6	-	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
	FT1	$\overline{\text{EMG}}$ (输入)		0	1	0	x	x	x	1
		$\overline{\text{OVV}}$ (输入)		0	0	1	x	x	x	1

9.4.5 端口 J 设置

表 9-8 端口设置列表(端口J)

引脚	端口类型	功能	复位后	PJ CR	PJ OD	PJ PUP	PJ PDN	PJ IE
PJ6	-	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
	FT5	AINB9(输入)	·	0	0	0	0	0
PJ7	-	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
	FT5	AINB10(输入)	·	0	0	0	0	0

9.4.6 端口 K 设置

表 9-9 端口设置列表(端口K)

引脚	端口类型	功能	复位后	PK CR	PK OD	PK PUP	PK PDN	PK IE
PK0	-	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
	FT5	AINB11(输入)	·	0	0	0	0	0
PK1	-	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
	FT5	AINB12(输入)	·	0	0	0	0	0

9.4.7 端口 M 设置

表 9-10 端口设置列表(端口M)

引脚	端口类型	功能	复位后	PM CR	PM OD	PM PUP	PM PDN	PM IE
PM0	-	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
PM1	-	输入端口		0	x	x	x	1
		输入端口		1	x	x	x	0

注：X1 与X2 功能可用。

10.16 位定时器 / 事件计数器(TMRB)

10.1 概述

TMRB在以下四个操作模式下操作：

- 16 位区间计时器模式
- 16 位事件计数器模式
- 16 位可编程脉冲生成模式 (PPG)
- 外触发器可编程脉冲生成模式(PPG)

捕捉功能的使用，允许TMRB执行以下两种测量。

- 一个通过外部触发器的触发脉冲输出
- 脉冲宽度测量

在本节的以下说明中，"x"表示通道编号。

10.2 规格差异

TMPM375FSDMG包含TMRB的 4 通道。

每个通道单独发挥功能，并且通道以相同的方式操作，表 10-1 中所示的规范差异除外。

表 10-1 TMRB模块的规范差异

规格通道	外部引脚		中断		内部连接
	外部时钟/ 捕获触发器输入引脚	计时触发器 输出引脚	捕捉中断	TMRB 中断	
	信号名称	信号名称			
TMRB0	-	-	INTCAP00 INTCAP01-	INTTB00 INTTB01	TIMPLS --> (捕捉输入)
TMRB4	TB4IN	-	-	INTTB40 INTTB41	
TMRB5	-	-	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51 --> (ADC转换启动)
TMRB7	TB7IN	TB7OUT	INTCAP70 INTCAP71	INTTB70 INTTB71	

10.3 配置

每个通道包含一个 16 位递增计数器，两个 16 位定时器寄存器(双重缓冲)，两个 16 位捕捉寄存器，两个比较器，一个捕捉输入控制，一个定时器flip-flop及其配套控制电路。定时器操作模式与定时器 flip-flop触发器均受某个寄存器控制。

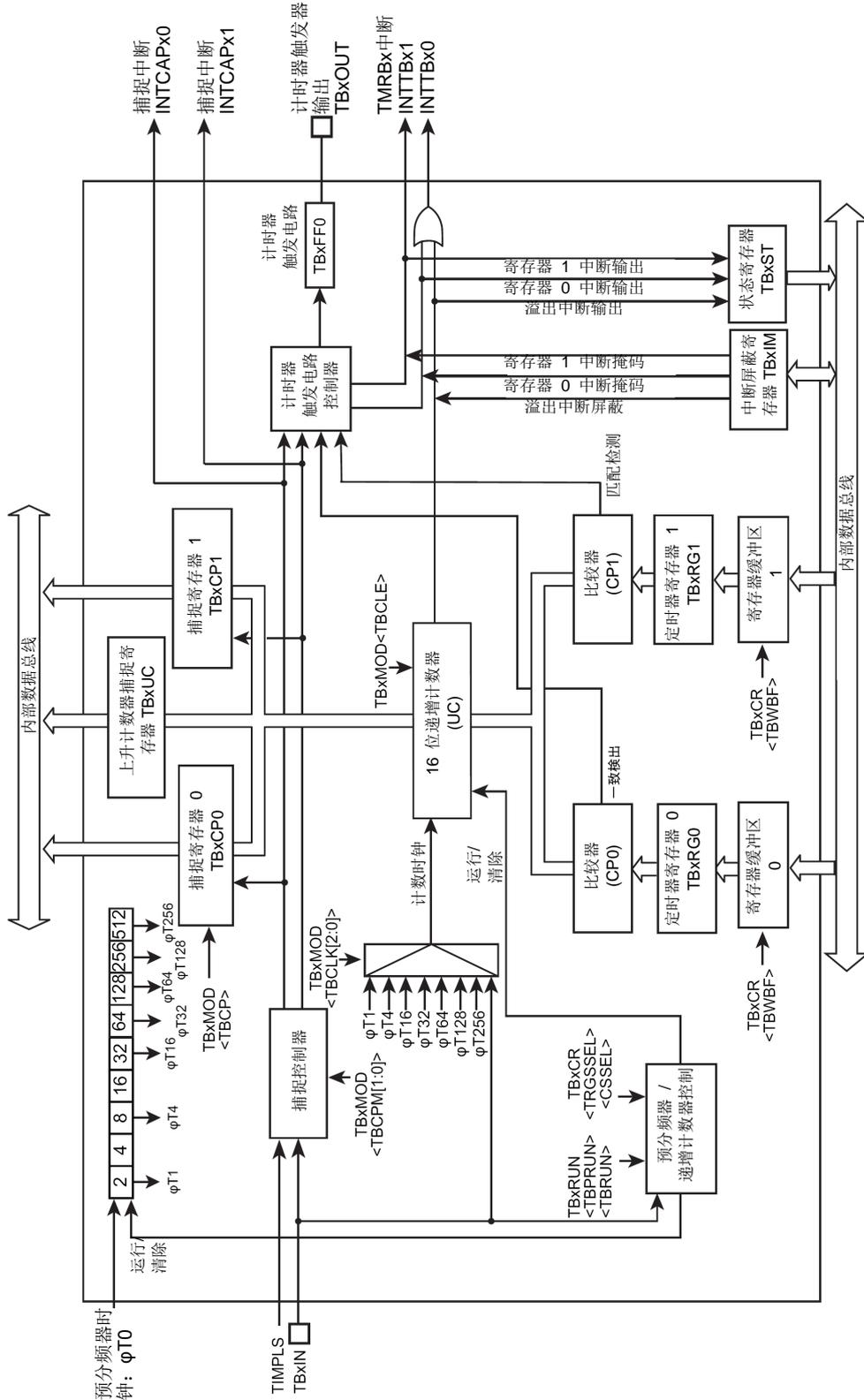


图 10-1 TMRBx方块图 (x= 0, 4, 5, 7)

10.4 寄存器

10.4.1 寄存器列表(按通道)

下表给出了各通道的寄存器名称与地址。

通道 x	基址
通道 0	0x4001_0000
通道 4	0x4001_0100
通道 5	0x4001_0140
通道 7	0x4001_01C0

寄存器名称 (x=0, 4, 5, 7)		地址(基+)
启用寄存器	TBxEN	0x0000
RUN寄存器	TBxRUN	0x0004
控制寄存器	TBxCR	0x0008
模式寄存器	TBxMOD	0x000C
触发电路控制寄存器	TBxFFCR	0x0010
状态寄存器	TBxST	0x0014
中断屏蔽寄存器	TBxIM	0x0018
递增计数器捕捉寄存器	TBxUC	0x001C
定时器寄存器 0	TBxRG0	0x0020
定时器寄存器 1	TBxRG1	0x0024
捕捉寄存器 0	TBxCP0	0x0028
捕捉寄存器 1	TBxCP1	0x002C

10.4.2 TBxEN(启用寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TBEN	R/W	<p>TMRBx操作</p> <p>0: 禁用 1: 启用</p> <p>规定TMRB运行。当禁用TMRB操作时, 无法向TMRB中的其他寄存器提供时钟, 可降低功耗。这使从TBxEN寄存器除外的其它寄存器中读取及写入被禁用。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")。如果TMRB操作执行后被禁用, 则该设置仍会保留在每个寄存器中。</p>
6	TBHALT	R/W	<p>调试停止时的时钟操作。</p> <p>0: 运行 1: 停止</p>
5-0	-	R	读作"0"。

10.4.3 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	预分频器操作 0: 停止&清除 1: 计数
1	-	R	读作"0"。
0	TBRUN	R/W	计数操作 0: 停止&清除 1: 计数

注 1: 在使用了外触发器启动时(<SSEL>=1), 应在<TBRUN>=<TBPRUN>=1 设置前选择 <CSSEL> 与 <TRGSEL>。

注 2: 在该计数器被停止(<TBRUN>="0"), 且TBxUC<TBUC[15:0]>被读取时, 会读取该计数器运行期间所捕捉的那个值。

10.4.4 TBxCR(控制寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TBWBFB	R/W	双缓冲器 0: 禁用 1: 启用
6-5	-	R/W	写作"0"。
4	-	R	读作"0"。
3	I2TB	R/W	在IDLE模式下运行 0: 停止 1: 操作
2	-	R/W	写作"0"。
1	TRGSEL	R/W	外触发器选择 0: 上升沿 1: 下降沿
0	CSSEL	R/W	计数器启动选择 0: 软件启动 1: 外触发

注 1: 操作TMRB时勿改变TBxCR。

注 2: 使用外触发器启动时 (<CSSEL> = 1), 则在设置<TBRUN> = <TBPRUN> = 1 之前, 选择<CSSEL>与<TRGSEL>。

10.4.5 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	TBRSWR	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	TBRSWR	R/W	写入到定时器寄存器0与1(在双缓冲器被启用时) 0: 数据传输至定时器寄存器 0 与 1 由相应的递增计数器(UC)来完成, 而不考虑重新写入缓冲寄存器 0 与 1。 1: 要将缓冲寄存器数据传输到定时器寄存器, 需要一起写入定时器寄存器 0 与 1。
6	TBCP	W	通过软件捕获控制 0: 通过软件TB捕捉 1: 忽略 在写入"0"时, 捕捉寄存器 0 (TBxCP0)取计数值。读作"1"。
5-4	TBCPM[1:0]	R/W	捕捉定时 00: 禁用捕捉定时 01: TBxIN↑ 一旦TBxIN引脚输入上升, 即将各计数值纳入捕捉寄存器 0 (TBxCP0)。 10: TBxIN↑ TBxIN↓ 一旦TBxIN引脚输入上升, 即将各计数值纳入捕捉寄存器 0 (TBxCP0)。 一旦TBxIN引脚输入下降, 即将各计数值纳入捕捉寄存器 1 (TBxCP1)。 11: TIMPLS↑ TIMPLS↓ 一旦TIMPLS输入上升, 即将计数值放入捕捉寄存器 0 (TBxCP0)中。 一旦TIMPLS输入下降, 即将计数值放入捕捉寄存器 1 (TBxCP1)中。
3	TBCLE	R/W	递增计数器控制器 0: 禁用递增计数器清除 1: 启用递增计数器清除。 清除并控制该递增计数器。 当写入"0"时上升计数器清0功能被禁用。在已写入"1"时, 其可在存在寄存器 1 (TBxRG1)匹配的情况下清除递增计数器。
2-0	TBCLK[2:0]	R/W	选择该TMRBx源时钟。 00: TBxIN引脚输入 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

注: 定时器在操作时不要转换TBxMOD寄存器。

10.4.6 TBxFFCR(触发电路控制寄存器)

	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	<p>在该递增计数器值被纳入TBxCP1 时, TBxFF0 反转触发。</p> <p>0: 禁用触发器 1: 启用触发</p> <p>通过设置"1", 该计时器触发电路会在递增计数器被纳入该捕捉寄存器1(TBxCP1)时反转。</p>
4	TBC0T1	R/W	<p>在该递增计数器值被纳入TBxCP0 时, TBxFF0 反转触发。</p> <p>0: 禁用触发器 1: 启用触发</p> <p>通过设置"1", 该计时器触发电路会在递增计数器被纳入该捕捉寄存器0(TBxCP0)时反转。</p>
3	TBE1T1	R/W	<p>在该递增计数器值与TBxRG1 匹配时, TBxFF0 反转触发。</p> <p>0: 禁用触发器 1: 启用触发</p> <p>通过设置"1", 该计时器触发电路会在递增计数器与计时器寄存器1(TBxRG1)匹配时反转。</p>
2	TBE0T1	R/W	<p>在该递增计数器值与TBxRG0 匹配时, TBxFF0 反转触发。</p> <p>0: 禁用触发器 1: 启用触发</p> <p>通过设置"1", 该计时器触发电路会在递增计数器与计时器寄存器0(TBxRG0)匹配时反转。</p>
1-0	TBFF0C[1:0]	R/W	<p>TBxFF0控制</p> <p>00: 反转 逆转TBxFF0 的值(使用软件逆转)。</p> <p>01: 设置 把TBxFF0 设置为"1"。</p> <p>10: 清除 把TBxFF0清0。</p> <p>11: 忽略</p> <p>*始终读作"11"。</p>

注: 注: 不要在该计时器运行期间改变TBxFFCR寄存器。

10.4.7 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	溢出标志 0: 未发生溢出。 1: 发生溢出。 在某递增计数器溢出时, 即设置"1"。
1	INTTB1	R	匹配标志(TBxRG1) 0: 未检测到匹配。 1: 检测到与TBxRG1 匹配。 当检测到计时寄存器1 (TBxRG1)的匹配时, 设置"1"。
0	INTTB0	R	匹配标志(TBxRG0) 0: 未检测到匹配。 1: 检测到与TBxRG0匹配。 在检测到计时器寄存器0(TBxRG0)的匹配时, 即设置"1"。

注 1: 仅没有被TBxIM屏蔽的因子向CPU输出中断请求。即使屏蔽设置完成, 标志还是被设置。

注 2: 通过读取TBxST寄存器, 该标志被清除。要清除此标志, 应读取TBxST寄存器。

10.4.8 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	溢出中断屏蔽 0: 禁用 1: 启用 将该递增计数器溢出中断设置为禁用或启用。
1	TBIM1	R/W	匹配中断屏蔽(TBxRG1) 0: 禁用 1: 启用 将计时器寄存器1 (TBxRG1)的匹配中断屏蔽设置为启用或禁用。
0	TBIM0	R/W	匹配中断屏蔽(TBxRG0) 0: 禁用 1: 启用 将计时器寄存器0 (TBxRG0)的匹配中断屏蔽设置为启用或禁用。

注：即使TBxIM寄存器的屏蔽配置是有效的，但要把状态设置为TBxST寄存器。

10.4.9 TBxUC (递增计数器捕捉寄存器)

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

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-0	TBUC[15:0]	R	通过读取递增计数器输出，捕捉某个值。 如果已读取TBxUC，则可捕捉当前的递增计数器值。

注：当操作计数器且读取TBxUC时，上上升计数器的数值被捕获并读取。

10.4.10 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	将某比较值设置为该递增计数器。

10.4.11 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	将某比较值设置为该递增计数器。

10.4.12 TBxCP0 (捕捉寄存器 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	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TBCP0							
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	TBCP0[15:0]	R	已读取从该递增计数器捕捉到的某个值。

10.4.13 TBxCP1 (捕捉寄存器 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	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TBCP1							
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	TBCP1[15:0]	R	已读取从该递增计数器捕捉到的某个值。

10.5 各电路的运行说明

除表 10-1 所示规格不同外，所有通道以相同的方式运行。

10.5.1 预分频器

利用现有的 4 位预分频器，可生成向上计数器 UC 的源时钟。

预分频器输入时钟 $\phi T0$ 为 CG 中的 CGSYSCR<PRCK[2:0]> 选择的 fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 或 fperiph/32。外围时钟 fperiph 可以是 fgear (通过 CG 中的 CGSYSCR<FPSEL> 选中的时钟)，也可以是 fc (被时钟齿轮划分之前的时钟)。

预分频器的运行或停止用 TBxRUN<TBPRUN> 设置，其中写入 "1"，则开始计数，写入 "0"，则清除，并停止计数。预分频器输出时钟分辨率如表 10-2 所示。

表 10-2 预分频器输出时钟分辨率 (fc = 40 MHz)

选择外围时钟 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	选择预分频器时 钟CGSYSCR <PRCK[2:0]>	预分频器输出时钟功能		
			$\phi T1$	$\phi T4$	$\phi T16$
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2 ¹ (0.05 μ s)	fc/2 ³ (0.2 μ s)	fc/2 ⁵ (0.8 μ s)
		001 (fperiph/2)	fc/2 ² (0.10 μ s)	fc/2 ⁴ (0.4 μ s)	fc/2 ⁶ (1.6 μ s)
		010 (fperiph/4)	fc/2 ³ (0.2 μ s)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)
		011 (fperiph/8)	fc/2 ⁴ (0.4 μ s)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)
		100 (fperiph/16)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)
		101 (fperiph/32)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)
	100 (fc/2)	000 (fperiph/1)	fc/2 ² (0.1 μ s)	fc/2 ⁴ (0.4 μ s)	fc/2 ⁶ (1.6 μ s)
		001 (fperiph/2)	fc/2 ³ (0.2 μ s)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)
		010 (fperiph/4)	fc/2 ⁴ (0.4 μ s)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)
		011 (fperiph/8)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)
		100 (fperiph/16)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)
		101 (fperiph/32)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)	fc/2 ¹¹ (51.2 μ s)
	101 (fc/4)	000 (fperiph/1)	fc/2 ³ (0.2 μ s)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)
		001 (fperiph/2)	fc/2 ⁴ (0.4 μ s)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)
		010 (fperiph/4)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)
		011 (fperiph/8)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)
		100 (fperiph/16)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)	fc/2 ¹¹ (51.2 μ s)
		101 (fperiph/32)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)	fc/2 ¹² (102.4 μ s)
	110 (fc/8)	000 (fperiph/1)	fc/2 ⁴ (0.4 μ s)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)
		001 (fperiph/2)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)
		010 (fperiph/4)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)
		011 (fperiph/8)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)	fc/2 ¹¹ (51.2 μ s)
		100 (fperiph/16)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)	fc/2 ¹² (102.4 μ s)
		101 (fperiph/32)	fc/2 ⁹ (12.8 μ s)	fc/2 ¹¹ (51.2 μ s)	fc/2 ¹³ (204.8 μ s)
111 (fc/16)	000 (fperiph/1)	fc/2 ⁵ (0.8 μ s)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)	
	001 (fperiph/2)	fc/2 ⁶ (1.6 μ s)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)	
	010 (fperiph/4)	fc/2 ⁷ (3.2 μ s)	fc/2 ⁹ (12.8 μ s)	fc/2 ¹¹ (51.2 μ s)	
	011 (fperiph/8)	fc/2 ⁸ (6.4 μ s)	fc/2 ¹⁰ (25.6 μ s)	fc/2 ¹² (102.4 μ s)	
	100 (fperiph/16)	fc/2 ⁹ (12.8 μ s)	fc/2 ¹¹ (51.2 μ s)	fc/2 ¹³ (204.8 μ s)	
	101 (fperiph/32)	fc/2 ¹⁰ (25.6 μ s)	fc/2 ¹² (102.4 μ s)	fc/2 ¹⁴ (409.6 μ s)	

表10-2 预分频器输出时钟分辨率($f_c = 40 \text{ MHz}$)

选择外围时钟 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	选择预分频器时 钟CGSYSCR <PRCK[2:0]>	预分频器输出时钟功能		
			$\phi T1$	$\phi T4$	$\phi T16$
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.05 μs)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.10 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	100 (fc/2)	000 (fperiph/1)	-	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.10 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	-	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	-	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	110 (fc/8)	000 (fperiph/1)	-	-	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	-	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	-	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
111 (fc/16)	000 (fperiph/1)	-	-	$fc/2^5$ (0.8 μs)	
	001 (fperiph/2)	-	-	$fc/2^6$ (1.6 μs)	
	010 (fperiph/4)	-	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	
	011 (fperiph/8)	-	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	
	100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)	
	101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	

注 1: 必须选择预分频器输出时钟 ϕTn , 以满足 $\phi Tn < f_{sys}$ 的条件(使得 ϕTn 慢于 f_{sys})。

注 2: 不要在该计时器运行期间改变时钟齿轮。

注 3: 符号“-”表示某设置被禁止。

表10-3 预分频器输出时钟分辨率($f_c = 40 \text{ MHz}$)

选择外围时钟 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	选择预分频器时 钟CGSYSCR <PRCK[2:0]>	预分频器输出时钟功能			
			$\phi T32$	$\phi T64$	$\phi T128$	$\phi T256$
0 (fgear)	000 (fc)	000 (fperiph/1)	$f_c/2^6$ (1.6 μs)	$f_c/2^7$ (3.2 μs)	$f_c/2^8$ (6.4 μs)	$f_c/2^9$ (12.8 μs)
		001 (fperiph/2)	$f_c/2^7$ (3.2 μs)	$f_c/2^8$ (6.4 μs)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$f_c/2^8$ (6.4 μs)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)
	100 (fc/2)	000 (fperiph/1)	$f_c/2^7$ (3.2 μs)	$f_c/2^8$ (6.4 μs)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)
		001 (fperiph/2)	$f_c/2^8$ (6.4 μs)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)
		010 (fperiph/4)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)
		011 (fperiph/8)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)
		100 (fperiph/16)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)
		101 (fperiph/32)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)
	101 (fc/4)	000 (fperiph/1)	$f_c/2^8$ (6.4 μs)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)
		001 (fperiph/2)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)
		010 (fperiph/4)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)
		011 (fperiph/8)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)
		100 (fperiph/16)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)
		101 (fperiph/32)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)	$f_c/2^{16}$ (1638.4 μs)
	110 (fc/8)	000 (fperiph/1)	$f_c/2^9$ (12.8 μs)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)
		001 (fperiph/2)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)
		010 (fperiph/4)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)
		011 (fperiph/8)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)
		100 (fperiph/16)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)	$f_c/2^{16}$ (1638.4 μs)
		101 (fperiph/32)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)	$f_c/2^{16}$ (1638.4 μs)	$f_c/2^{17}$ (3276.8 μs)
111 (fc/16)	000 (fperiph/1)	$f_c/2^{10}$ (25.6 μs)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	
	001 (fperiph/2)	$f_c/2^{11}$ (51.2 μs)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	
	010 (fperiph/4)	$f_c/2^{12}$ (102.4 μs)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)	
	011 (fperiph/8)	$f_c/2^{13}$ (204.8 μs)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)	$f_c/2^{16}$ (1638.4 μs)	
	100 (fperiph/16)	$f_c/2^{14}$ (409.6 μs)	$f_c/2^{15}$ (819.2 μs)	$f_c/2^{16}$ (1638.4 μs)	$f_c/2^{17}$ (3276.8 μs)	
	101 (fperiph/32)	$f_c/2^{15}$ (819.2 μs)	$f_c/2^{16}$ (1638.4 μs)	$f_c/2^{17}$ (3276.8 μs)	$f_c/2^{18}$ (6553.6 μs)	

表10-3 预分频器输出时钟分辨率($f_c = 40 \text{ MHz}$)

选择外围时钟 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	选择预分频器时 钟CGSYSCR <PRCK[2:0]>	预分频器输出时钟功能			
			$\phi T32$	$\phi T64$	$\phi T128$	$\phi T256$
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^6$ (1.6 μs)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)
		001 (fperiph/2)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{14}$ (409.6 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^6$ (1.6 μs)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)
		001 (fperiph/2)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{14}$ (409.6 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^6$ (1.6 μs)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)
		001 (fperiph/2)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{14}$ (409.6 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^6$ (1.6 μs)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)
		001 (fperiph/2)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{14}$ (409.6 μs)
111 (fc/16)	000 (fperiph/1)	$fc/2^6$ (1.6 μs)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	
	001 (fperiph/2)	$fc/2^7$ (3.2 μs)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	
	010 (fperiph/4)	$fc/2^8$ (6.4 μs)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	
	011 (fperiph/8)	$fc/2^9$ (12.8 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	
	100 (fperiph/16)	$fc/2^{10}$ (25.6 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)	
	101 (fperiph/32)	$fc/2^{11}$ (51.2 μs)	$fc/2^{12}$ (102.4 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{14}$ (409.6 μs)	

注 1: 必须选择预分频器输出时钟 ϕTn , 以满足 $\phi Tn < fsys$ 的条件(使得 ϕTn 慢于 $fsys$)。

注 2: 不要在该计时器运行期间改变时钟齿轮。

注 3: 符号"-"表示某设置被禁止。

10.5.2 升计数器(UC)

UC为 16 位二进制计数器。

- 源时钟

TBxMOD<TBCLK[2:0]>规定的UC源时钟可从预分频器输出时钟的三种类型 ϕ T1, ϕ T4, ϕ T16, ϕ T32, ϕ T64, ϕ T128, ϕ T256中选择, 或者从TBxIN引脚的外部时钟中选择。

- 计数开始/停止

计数器运行由TBxRUN<TBRUN>规定。若<TBRUN> = "1", 则UC开始计数; 若<TBRUN> = "0" 则其停止计数并清除计数器值。

- 清除UC计时

1. 当检测到一个匹配时

在比较器检测到计数器值与在TBxRG1中设置的值之间存在匹配时, 通过设置

TBxMOD<TBCLE> = "1", 即可清除UC。若TBxMOD<TBCLE> = "0", 则UC作为自由运行的计数器运行。

2. 当UC停止时

若TBxRUN<TBRUN>="0", 则UC停止计数并清除计数器值。

- UC 溢出

若发生UC溢位, 则会生成INTTBx0溢位中断。

10.5.3 计时器寄存器(TBxRG0, TBxRG1)

TBxRG0 与TBxRG1 均为寄存器, 用于设置进行向上计数器值比较所需的值, 且各通道均内建有两个寄存器。若比较器检测到在计时器寄存器中设置的值与某UC向上计数器中某值之间存在匹配关系, 则其会输出匹配检测信号。

TBxRG0和TBxRG1由双缓存配置组成, 与寄存器缓存器配对。在初始状态时, 该双缓存被禁用。控制用双缓存的禁用或启用, 由TBxCR<TBWBF>位指定。若<TBWBF> = "0", 则双缓存变为禁用。若<TBWBF> = "1", 则其变为启用。当双缓存使能时, 若UC与TBxRG1 匹配, 则数据从寄存器缓存器传送到定时器寄存器(TBxRG0/1)。当计数器停止时, 即使双缓存使能, 双缓存仍然作为单缓存运行, 直接数据能写入TBxRG0和TBxRG1。

10.5.4 捕获

该电路用于控制"从UC向上计数器传送到TBxCP0 与TBxCP1 捕获寄存器的"锁存值的时序。该时段用于锁存数据, 并由TBxMOD<TBCPM [1:0]>指定。

软件也可用于将数值从UC上升计数器导入捕获寄存器中; 尤其是每次当"0"被写入TBxMOD<TBCP>时, UC数值被计入TBxCP0 捕获寄存器。

10.5.5 捕获寄存器(TBxCP0, TBxCP1)

此寄存器可捕获一个向上计数器(UC)值。

10.5.6 向上计数器捕获寄存器(TBxUC)

除上述捕获功能外，通过读取TBxUC寄存器，也能捕获UC当前计数值。

10.5.7 比较器(CP0, CP1)

此寄存器可将该向上计数器(UC)与计时器寄存器(TBxRG0 and TBxRG1)的值设置进行比较，以检测是否存在匹配。若检测到匹配，则产生INTTBx0 和INTTBx1。

10.5.8 计时器触发电路(TBxFF0)

计时器触发器 (TBxFF0)由比较器的匹配信号及捕获寄存器的锁存信号反转。通过设置TBxFFCR<TBC1T1, TBC0T1, TBE1T1与TBE0T1>，即可启用或禁用其反转。

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。写入"01"即可将其设置为"1"，通过写入"10"即可清除该设置为"0"。

TBxFF0 的值可被输出到计时器输出引脚(TBxOUT)。如拟进行计时器输出，则必须事先对相应端口设置进行编程。

10.5.9 捕获中断(INTCAPx0, INTCAPx1)

在将UC向上计数器的数值锁存到TBxCP0 和TBxCP1 捕获寄存器时，产生中断INTCAPx0 和INTCAPx1。中断时序由CPU指定。

10.6 各模式的运行说明

10.6.1 16 位间隔计时器模式

在产生恒定周期中断的情况下，将间隔时间设到定时器寄存器(TBxRG1)中，产生INTTBx1 中断。

	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	停止计数运行。
中断设置赋能寄存器	← *	*	*	*	*	*	*	*	通过将相应位设置为"1"允许INTTBx1中断。
TBxFFCR	← X	X	0	0	0	0	1	1	禁用TBxFF0 反转触发信号。
TBxMOD	← 0	1	0	0	1	*	*	*	改为将预分频器输出时钟用作输入时钟。禁用捕获功能。
						(*** = 001, 010, 011, 100, 101, 110, 111)			
TBxRG1	← *	*	*	*	*	*	*	*	指定时间间隔。(16 位)
	← *	*	*	*	*	*	*	*	
TBxRUN	← *	*	*	*	*	1	X	1	启动TMRBx。

注: X; 忽略 -; 无改变

10.6.2 16 位事件计数器模式

通过将输入时钟用作外部时钟(TBxIN引脚输入)，有可能将其变为事件计数器。

该向上计数器可在TBxIN引脚输入的上升沿实现递增计数。通过用软件捕获数值，然后读取捕获的数值，就有可能读取计数取。

	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	停止计数运行。
设置PORT寄存器。									将相应的端口分配到TBxIN。
TBxFFCR	← X	X	0	0	0	0	1	1	禁用TBxFF0反转触发信号。
TBxMOD	← 0	1	0	0	0	0	0	0	将TBxIN改为输入时钟。
TBxRUN	← *	*	*	*	*	1	X	1	启动TMRBx。
TBxMOD	← 0	0	0	0	0	0	0	0	软件捕获已完成。

注: X; 忽略 -; 无改变

10.6.3 16 位 PPG(可编程脉冲发生)输出模式

可输出具备任何频率与任何占空的方波(可编程方波)。输出脉冲可为低活跃度或高活跃度
 在向上计数器(UC的设置值)可匹配各计时器寄存器的设置值时,通过触发计时器触发电路(TBxFF)进行反转,即可从TBxOUT引脚输出可编程方波。需要注意的是, TBxRG0和TBxRG1 的设置值必须满足如下要求:

TBxRG0 的设置值 < TBxRG1 的设置值

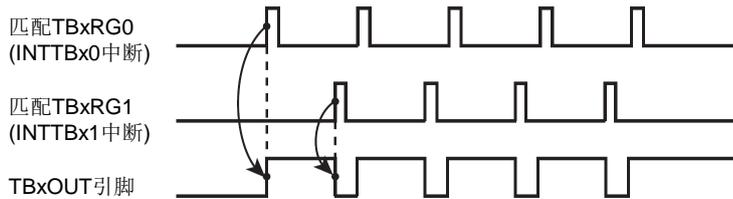


图 10-2 可编程脉冲发生(PPG)输出示例

在该模式下,在向上计数器的设置值匹配TBxRG1 的设置值时,通过启用TBxRG0 的双缓存,即可将寄存器缓存器 0 的值转入TBxRG0。这便于对低占空进行处理。

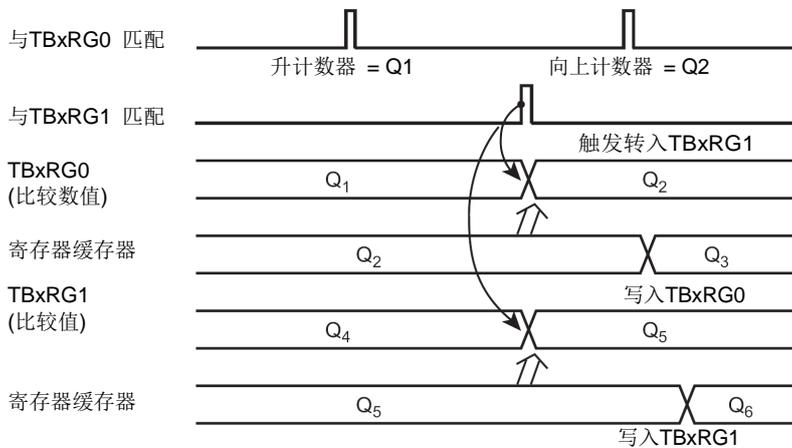


图 10-3 寄存器缓存操作

该模式方块图如下所示。

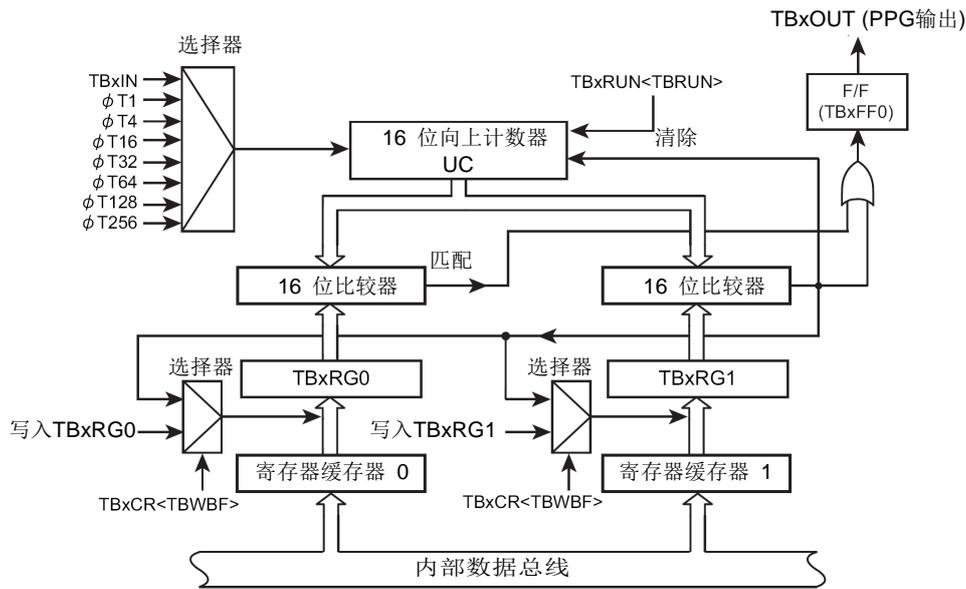


图 10-4 810ck 16 位PPG模式

	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	停止计数运行。
TBxCR	← 0	0	0	X	-	0	0	0	禁用双缓存。
TBxRG0	← *	*	*	*	*	*	*	*	指定某占空(16 位)
TBxRG1	← *	*	*	*	*	*	*	*	指定一个周期。(16 位)
TBxCR	← 1	0	0	X	-	X	0	0	启用TBxRG0双缓存。 (在生成INTTBx0中断时, 更改该占空/周期)
TBxFFCR	← X	X	0	0	1	1	1	0	当有对应TBxRG0 或 TBxRG1的匹配时, 做相应指定以触发TBxFF0进行逆转, 并将TBxFF0的初始值设置为"0"。
TBxMOD	← 0	1	0	0	1	*	*	*	指定该预分频器输出时钟用作输入时钟, 并禁用该捕获功能。 (* = 001, 010, 011, 100, 101, 110, 111)

设置PORT寄存器。

TBxRUN ← * *****1X1

注: X: 忽略

-: 无变化

UC即被清除, 以匹配TBxRG1。

将相应的端口分配到TBxOUT。

启动TMRBx。

10.6.4 外部触发器可编程脉冲发生输出模式(PPG)

利用外部计数启动触发器通过瞬间延迟激发单触发脉冲。

对16位上数计数器(UC)进行编程,对TBxIN引脚(TBxCR [1:0]="01")的上升沿进行上数计数。

TBxRG0加载有脉冲延迟(d),TBxRG1加载有TBxRG0值(d)与脉冲宽度(p)的和。当16位向上计数器停止时(TBxRUN<TBRUN>=0),必须进行上述设置。

为启用定时器触发器的触发脉冲,设置TBxFFCR<TBE1T1, TBET1>至"11"。采用这一设置,当16位上数计数器(UC)对应TBxRG TBxRG0或TBxRG1时,定时器触发器反转。

设置TBxRUN<TBRUN>至"1",通过外触发器启用上数计数。

在通过外触发器产生单触发脉冲之后,要通过TBxRUN<TBRUN>设置禁用定时器触发器的反转或停止16位计数器。

文本中使用的符号(d)与(p)相当于图10-5中的符号d与p。

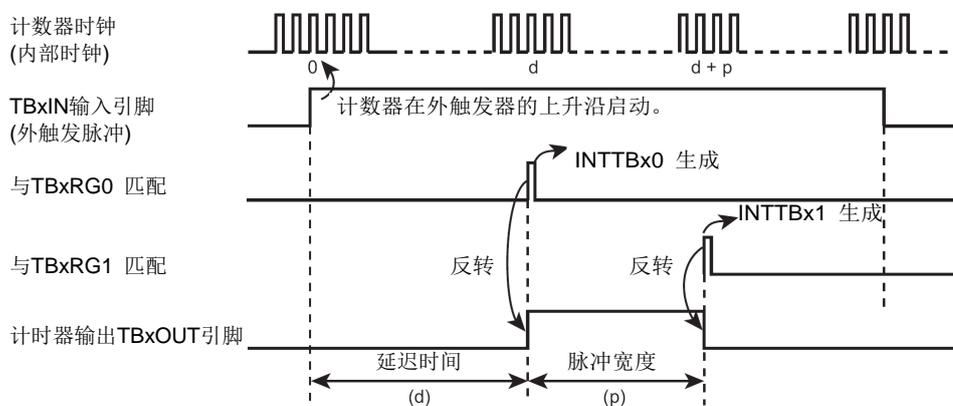


图10-5 利用外计数启动触发器的单触发脉冲(带延迟)

10.7 基于捕获功能的应用

捕获功能可用于改进多种应用(包括以下所述):

1. 外部脉冲触发的单次脉冲输出
2. 脉冲宽度测量

10.7.1 由外部脉冲触发的单次脉冲输出

由外部脉冲触发的单次脉冲输出的实现方式如下:

利用预分频器输出时钟,使 16 位向上计数器在自由运行状态下运行并计数。通过TBxIN引脚输入一个外部脉冲。利用该捕获功能,在该外部脉冲上升时生成一个触发信号,该向上计数器的值随即被纳入该捕获寄存器(TBxCP0)。

CPU必须编程,以便在外部触发脉冲上升沿产生中断INTCAPx0。该中断用于将该计时器寄存器(TBxRG0)设置为该TBxCP0值(c)与该延时时间(d)之和(c + d),并将该计时器寄存器(TBxRG1)设置为TBxRG值与单次脉冲的脉冲宽度之和(c + d + p)[必须在下一个匹配之前完成TBxRG1 的变换]。此外,定时器触发器控制寄存器(TBxFFCR<TBE1T1, TBE0T1>)必须设置为"11"。由此在TBxUC与TBxRG0和TBxRG1 匹配时,就能触发计时器触发器(TBxFF0)反转。在单次脉冲被输出之后,该触发信号即被INTTBx0 / INTTBx1 中断禁用。

文本中使用的符号(c), (d)与(p)相当于图 10-6 中的符号c, d与p。

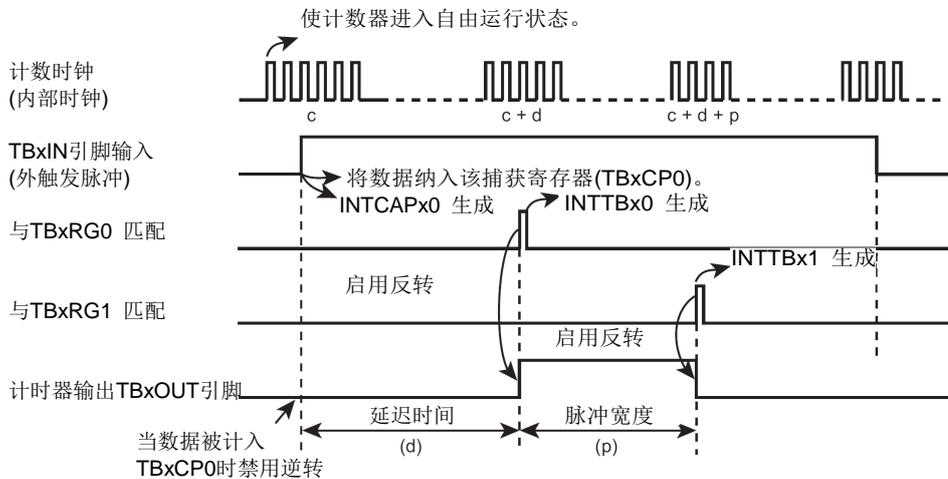


图 10-6 单触发脉冲输出(带延迟)

在上升沿触发TBxIN输入, 在 3 ms后输出 2 ms脉宽单次脉冲的情况下, 设置如下。(选择φT1 来计数。)

	7	6	5	4	3	2	1	0	
由TBxIN进行的[主处理]捕获设置 设置PORT寄存器。									
TBxEN	←	1	X	X	X	X	X	X	X
TBxRUN	←	X	X	X	X	X	0	X	0
TBxMOD	←	0	1	0	1	0	0	0	1
TBxFFCR 设置PORT寄存器。	←	X	X	0	0	0	0	1	0
中断设置赋能寄存器	←	*	*	*	*	*	*	*	*
TBxRUN INTCAPx0	←	*	*	*	*	*	1	X	1
TBxRG0	←	*	*	*	*	*	*	*	*
TBxRG1	←	*	*	*	*	*	*	*	*
TBxFFCR	←	*	*	*	*	*	*	*	*
TBxFFCR	←	X	X	-	-	1	1	-	-
TBxIM	←	X	X	X	X	X	1	0	1
中断设置赋能寄存器	←	*	*	*	*	*	*	*	*
INTTBx1 中断服务程序的[处理]输出禁用 TBxFFCR	←	X	X	-	-	0	0	-	-
	←	*	*	*	*	*	*	*	*

将相应的端口分配到TBxIN。
 启用TMRBx运行。
 停止计数运行。
 将源时钟变为φT1。将某计数值纳入TBxIN上升沿的TBxCP0。
 清除TBxFF0 反转触发脉冲并禁用。
 将相应的端口分配到TBxOUT。
 允许设置为"1"来产生INTCAPx0 中断对应位规定的中断。
 启动该TMRBx模块。
 中断服务程序的[处理]脉冲输出设置
 设置计数值。(TBxCAP0 + 3ms/φT1)。
 设置计数值。(TBxCAP0 + (3+2)ms/φT1)
 若UC与TBxRG0 以及TBxRG1 一致, 则使TBxFF0 反转。
 除TBxRG1对应中断信号之外均进行屏蔽。
 通过设置为"1"准许生成INTTBx1 中断信号对应位规定的中断信号。
 清除TBxFF0 反转触发器设置。
 禁止设置为"1"来产生INTCAPx0 中断对应位规定的中断。

注: X; 忽略

-; 无变化

当没有延迟要求时, 当数据纳入TBxCP0, 且TBxRG1 设置为TBxCP0 值(c)与单次脉宽(p)之和(c + p)时, 通过产生INTCAPx0中断, 可使TBxFF0 反转。[TBxRG1 更改必须在下一个匹配之前完成。]

当UC与TBxRG1 匹配时, 促使TBxFF0 反转, 而通过生成INTTBx1 中断, 被禁用。

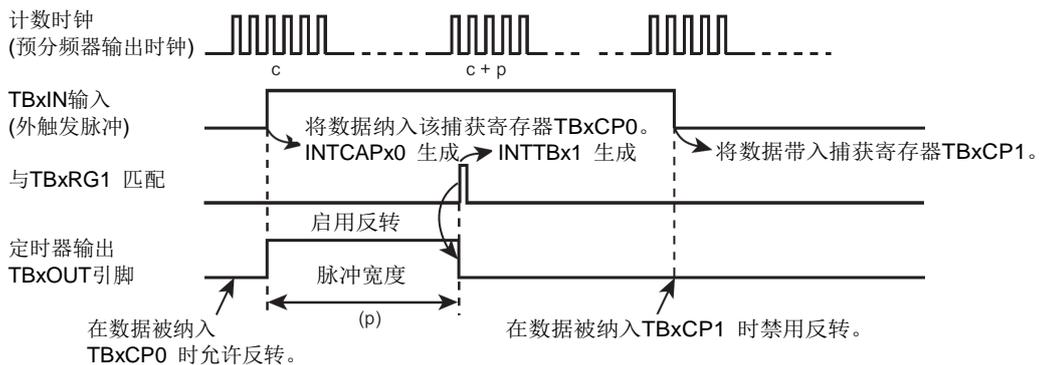


图 10-7 外部脉冲(无延时)触发的单触发脉冲输出

10.7.2 脉冲宽度测量

通过使用该捕获功能，可测量外部脉冲的“高”电平宽度。具体地说，利用该预分频器输出时钟使其进入自由运行状态，通过该TBxIN引脚输入一个外部脉冲，该向上计数器(UC)开始计数。用捕获功能在外部脉冲的各上升沿和下降沿产生触发器，向上计数器值被计入捕获寄存器(TBxCP0, TBxCP1)。必须对CPU进行编程，使通过TBxIN引脚在外部脉冲输入下降沿产生INTCAPx1。将TBxCP0与TBxCP1之差乘以内部时钟的时钟周期，即可计算得出该“高”电平脉冲宽度。例如，若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 $100 \times 0.5 \mu\text{s} = 50 \mu\text{s}$ 。

在测量超过UC最长计数时间的脉冲宽度时，必须小心操作(其取决于所使用的源时钟)。必须用软件测量这种脉冲宽度。

也可测量外部脉冲的“低”电平宽度。在此类情况下，执行图10-8所示第二阶INTCAPx0中断处理，可初步求得C2产生的第一时间与C1产生的第二时间之差，该差值乘以预分频器输出时钟周期，可求得“低”电平宽度。

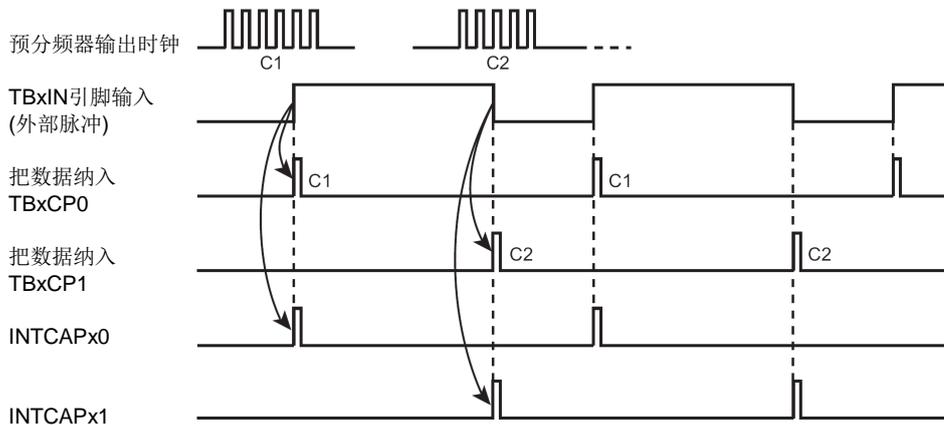


图 10-8 脉冲宽度测量



11 串行通道(SIO/UART)

11.1 概述

串行通道(SIO/UART) 具有如下所示的模式。

- 同步通信模式(输入/输出接口模式)
- 异步通信模式(UART模式), 有如下特点。
 - 传送时钟
 - 由预分频器从外部时钟($\phi T0$)频率分为1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128。
 - 使其从预分频器输出时钟频率分为 1 至 16 成为可能。
 - 使其从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)。
 - 缓存器
 - 可用的重复缓存器功能。
 - 使清除发送缓存器成为可能。
 - FIFO
 - 可用的 4 字节FIFO, 包括发送与接收。
 - 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。
 - 最后位输出之后的TXD_x引脚状态可选择如下:保持最后位的"高"电平/"低"电平/状态
 - 当输入SCLK情况下发生欠载运行出错时, TXD_x引脚的状态可选择如下:
保持"高"电平/"低"电平
 - TXD_x引脚的最后位保持时间可在输入SCLK的情况下进行设置。
 - UART模式
 - 数据长度:7 位, 8 位, 9 位
 - 增加奇偶校验位(应与 9 位数据长度对应)
 - 拟使用唤醒功能的串行链接
 - 握手功能 $\overline{CTS_x}$ (引脚)

在以下说明中, "x"表示通道编号。

11.2 SIO 模块规格差异

TMPM375FSDMG具有两个SIO通道。

各通道均独立工作。各通道中使用的引脚，中断及UART源时钟如下。

表 11-1 SIO模块规范差异

	引脚名称			中断		UART源时钟
	TXD	RXD	$\overline{\text{CTS}}_x/\text{SCLK}_x$	接收中断	传送中断	
通道 0	PE0	PE1	PE2	INTRX0	INTTX0	TB4OUT
通道 1	PF0	PB6	-	INTRX1	INTTX1	TB4OUT
	PB4		-			

11.3 配置

图 11-1 显示串行通道方块图。

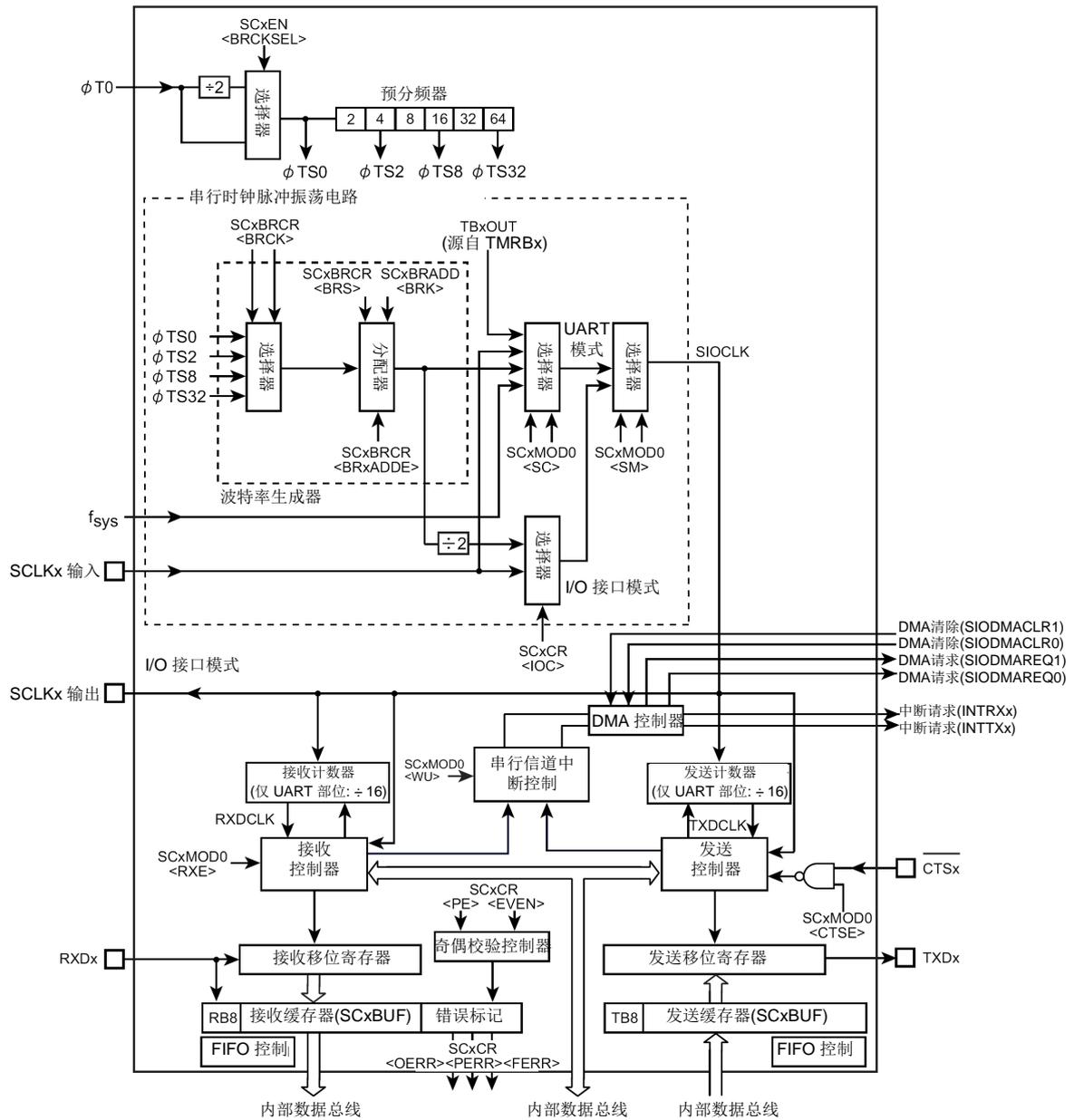


图 11-1 串行通道方块图

11.4 寄存器说明

11.4.1 寄存器清单

下表显示控制寄存器及其地址。

有关基地址的详细情况，参照“内存印象图”一章的“外部功能地址表”。

通道 x	基址
通道 0	0x4002_0080
通道 1	0x4002_00C0

寄存器名(x=0,1)		地址 (基+)
启用寄存器	SCxEN	0x0000:
缓存寄存器	SCxBUF	0x0004
控制寄存器	SCxCR	0x0008
模式控制寄存器 0	SCxMOD0	0x000C
波特率发生器控制寄存器	SCxBRCR	0x0010
波特率发生器控制寄存器 2	SCxBRADD	0x0014
模式控制寄存器 1	SCxMOD1	0x0018
模式控制寄存器2	SCxMOD2	0x001C
接收FIFO配置寄存器	SCxRFC	0x0020
发送FIFO配置寄存器	SCxTFC	0x0024
接收FIFO状态寄存器	SCxRST	0x0028
发送FIFO状态寄存器	SCxTST	0x002C
FIFO配置寄存器	SCxFCNF	0x0030

注：不要在数据发送或接收期间修改任何控制寄存器

11.4.2 SCxEN(启用寄存器)

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

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

11.4.3 SCxBUF(缓存寄存器)

写入操作时, SCxBUF用作传送缓存器或FIFO, 读取操作时, 用作接收缓存器或FIFO。

	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: 传送缓存器或FIFO [读取] RB: 接收缓存器或FIFO

位	比特符号	类型	功能
2	FERR	R	成帧错误标志(注) 0: 正常操作 1: 错误
1	SCLKS	R/W	选择输入时钟沿(适用于I/O接口模式) 在时钟脉冲输出模式下, 设置为"0"。 0: 发送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从高电平开始。 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从低电平开始。
0	IOC	R/W	选择时钟(适用于I/O接口模式) 0: 波特率生成器 1: SCLKx引脚输入

注: <OERR>, <PERR> 与<FERR>在读出时清除至"0"。

11.4.5 SCxMOD0 (模态控制寄存器 0)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	TB8	R/W	发送数据位 8 (适用于UART) 以 9 位UART模式写入发送数据的第 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。 设置值"1"可使能握手功能(利用引脚)
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 位长度的UART模式 10: 8 位长度的UART模式 11: 9 位长度的UART模式
1-0	SC[1:0]	R/W	串行传送时钟(适用于UART) 00: 定时器输出 01: 波特率生成器 10: 内部时钟f _{sys} 11: 外部时钟(SCLKx引脚输入) (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脉冲)

注 1: 首先确定所有模态控制寄存器, 然后<RXE>。

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

11.4.6 SCxMOD1 (模态控制寄存器 1)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	I2SC	R/W	IDLE 0: 停止 1: 操作 指定IDLE模式操作。
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(传送) 11: 全双工 可在I/O接口模式下配置该传送模式。当FIFO使能时, 指定FIFO的配置。在UART模式下, 仅指定FIFO的配置。
4	TXE	R/W	传送控制器(注1)(注2) 0: 禁用 1: 启用 该位可启用传送, 且在所有传送模式下均有效。
3-1	SINT[2:0]	R/W	连续传送间隔时间(适用于I/O接口模式) 000: 无 001: 1 x SCLK周期 010: 2 x SCLK周期 011: 4 x SCLK周期 100: 8 x SCLK周期 101: 16 x SCLK周期 110: 32 x SCLK周期 111: 64 x SCLK周期 在已选择SCLK引脚的去情况下, 该参数仅对I/O接口模式有效。在其它模式中, 该参数没有意义。 当双缓存以I/O接口模式启用时, 确定连续传送的间隔时间。
0	-	R/W	写作"0"。

注 1: 首先确定所有模态控制寄存器, 然后启用<TXE>。

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

11.4.7 SCxMOD2(模态控制寄存器 2)

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

位	比特符号	类型	功能											
31-8	-	R	读作"0"。											
7	TBEMP	R	<p>传送缓存器为空标志</p> <p>0: 满</p> <p>1: 空</p> <p>若双缓存被禁用, 则该标志可忽略。</p> <p>该标志表示该传送双缓存器为空。在该传送双缓存器中的数据被移动到传送移位寄存器, 且该双缓存器为空时, 可将该位设置为"1"。</p> <p>通过将数据重新写入到该双缓存器, 即可将该位设置为"0"。</p>											
6	RBFL	R	<p>接收缓存器已满标志。</p> <p>0: 空</p> <p>1: 满</p> <p>若双缓存被禁用, 则该标志可忽略。</p> <p>该标志表示接收双缓存器已满。</p> <p>当接收操作完成, 并且接收的数据从接收移位寄存器移到接收双缓存器时, 该位变为"1", 而读取该位, 就可将它变为"0"。</p>											
5	TXRUN	R	<p>在传送标志中</p> <p>0: 停止</p> <p>1: 操作</p> <p>该状态标志表示数据传送正在进行中。</p> <p><TXRUN>与<TBEMP>位可指示以下状态。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><TXRUN></th> <th><TBEMP></th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>-</td> <td>传送中</td> </tr> <tr> <td rowspan="2">0</td> <td>1</td> <td>传送完成</td> </tr> <tr> <td>0</td> <td>等待状态, 数据在发送缓存器中。</td> </tr> </tbody> </table>	<TXRUN>	<TBEMP>	状态	1	-	传送中	0	1	传送完成	0	等待状态, 数据在发送缓存器中。
<TXRUN>	<TBEMP>	状态												
1	-	传送中												
0	1	传送完成												
	0	等待状态, 数据在发送缓存器中。												
4	SBLN	R/W	<p>停止位长度(适用于UART模式)</p> <p>0: 1 位</p> <p>1: 2 位</p> <p>这样可在UART模式下指定传送停止位的长度。</p> <p>对于接收侧而言, 仅利用与<SBLN>设置无关的单个位即可做出该判定。</p>											
3	DRCHG	R/W	<p>设置传送方向</p> <p>0: LSB先</p> <p>1: MSB先</p> <p>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。</p>											
2	WBUF	R/W	<p>启用双缓存器</p> <p>0: 禁用</p> <p>1: 启用</p> <p>本参数使能或禁用发送/接收双缓存器在I/O接口模式下发送(在SCLK输出/输入两种模式)和接收(SCLK输出模式)数据及在UART模式下发送数据。</p> <p>在接口模式(SCLK输入)与UART模式下接收数据时, 双缓存存在 0 或 1 被设置为<WBUF>位时均会启用。</p>											

位	比特符号	类型	功能										
1-0	SWRST [1:0]	R/W	<p>软件复位</p> <p>用"01"盖写"10"即可触发软件复位。当执行该软件复位时，下列位元被初始化，发送/接收电路变成初始状态(注1)(注2)。</p> <table border="1"> <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: 在数据传送进行期间，任何软件复位操作均必须连续执行两次。

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

11.4.8 SCxBRCR(波特率发生器控制寄存器)

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

位	比特符号	类型	功能
31-8	-	R	读作"0"。
7	-	R/W	写入"0"。
6	BRADDE	R/W	$N + (16 - K)/16$ 分配器功能(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: ϕ TS0 01: ϕ TS2 10: ϕ TS8 11: ϕ TS32
3-0	BRS[3:0]	R/W	分频比"N" 0000: 16 0001: 1 0010: 2 ... 1111:15

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

注 2: 波特率发生器分频比"1"只有在I/O接口模式下使用双缓存时才能加以规定。

11.4.9 SCxBRADD(波特率发生器控制寄存器 2)

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

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

表 11-2 列出了波特率发生器分频比率的设置。

表 11-2 设置分频比率

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

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

注2: 当在UART模式下使用" $N + (16 - K) / 16$ "分频功能时, 1 ("0001")或 16 ("0000")作为分频比, 不能应用于N。

注3: 波特率发生器分频比"1"只有在I/O接口模式下使用双缓存时才能指定。

注4: 禁止规定"K = 0"。

11.4.10 SCxFCNF(FIFO 配置寄存器)

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

位	比特符号	类型	功能						
31-8	-	R	读作"0"。						
7-5	-	R/W	务必写入"000"。						
4	RFST	R/W	<p>在接收FIFO中使用的字节数。</p> <p>0: 最高</p> <p>1: 与接收FIFO的FILL电平相同</p> <p>当接收FIFO启用时, 即选择要使用的接收FIFO字节数。(注1)</p> <p>0: 已配置的FIFO最大字节数(亦见<CNFG>)。</p> <p>1: 与SC0RFC <RIL[1:0]>规定的产生接收中断的填充电平相同。</p>						
3	TFIE	R/W	<p>为发送FIFO发送中断信号。</p> <p>0: 禁用</p> <p>1: 启用</p> <p>当启用发送FIFO时, 则由该参数启用发送中断信号或禁用发送中断信号。</p>						
2	RFIE	R/W	<p>为接收FIFO接收中断信号。</p> <p>0禁用</p> <p>1: 启用</p> <p>当启用接收FIFO时, 则由该参数启用接收中断信号或禁用接收中断信号。</p>						
1	RXTXCNT	R/W	<p>RXE/TXE的自动禁用</p> <p>0: 无</p> <p>1: 自动禁用</p> <p>控制发送和接收的自动禁用。设置"1"则启用如下操作。</p> <table border="1" data-bbox="523 1464 1406 1659"> <tr> <td>半双工接收</td> <td>当接收FIFO填入达到确定的有效字节数时, SCxMOD0<RXE>则自动地设置为"0", 以禁止进一步的接收。</td> </tr> <tr> <td>半双工 传送</td> <td>当发送FIFO为空时, SCxMOD1<TXE>则自动地设置为"0", 以禁止更进一步的发送。</td> </tr> <tr> <td>全双工</td> <td>当满足上述两个条件之一时, TXE/RXE自动设置为"0", 以抑制进一步的发送和接收。</td> </tr> </table>	半双工接收	当接收FIFO填入达到确定的有效字节数时, SCxMOD0<RXE>则自动地设置为"0", 以禁止进一步的接收。	半双工 传送	当发送FIFO为空时, SCxMOD1<TXE>则自动地设置为"0", 以禁止更进一步的发送。	全双工	当满足上述两个条件之一时, TXE/RXE自动设置为"0", 以抑制进一步的发送和接收。
半双工接收	当接收FIFO填入达到确定的有效字节数时, SCxMOD0<RXE>则自动地设置为"0", 以禁止进一步的接收。								
半双工 传送	当发送FIFO为空时, SCxMOD1<TXE>则自动地设置为"0", 以禁止更进一步的发送。								
全双工	当满足上述两个条件之一时, TXE/RXE自动设置为"0", 以抑制进一步的发送和接收。								
0	CNFG	R/W	<p>FIFO启用。</p> <p>0: 禁用</p> <p>1: 启用</p> <p>使能FIFO。(注 2) 当<CNFG>设置为"1"时, FIFO使能。</p> <p>若得到启用, 该SCOMOD1<FDPX[1:0]>设置则自动地配置FIFO, 具体如下:</p> <table border="1" data-bbox="523 1823 1406 1960"> <tr> <td>半双工接收</td> <td>接收FIFO 4 字节</td> </tr> <tr> <td>半双工 传送</td> <td>发送FIFO 4 字节</td> </tr> <tr> <td>全双工接收</td> <td>接收FIFO 2 字节 + 发送FIFO 2 字节</td> </tr> </table>	半双工接收	接收FIFO 4 字节	半双工 传送	发送FIFO 4 字节	全双工接收	接收FIFO 2 字节 + 发送FIFO 2 字节
半双工接收	接收FIFO 4 字节								
半双工 传送	发送FIFO 4 字节								
全双工接收	接收FIFO 2 字节 + 发送FIFO 2 字节								

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

注 2: FIFO不能用于 9 位UART模式。

11.4.11 SCxRFC(接收 FIFO 配置寄存器)

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

位	比特符号	类型	功能															
31-8	-	R	读作"0"。															
7	RFCS	W	接收FIFO清除(注 1) 1: 清除 当SCxRFC<RFCS>设置为"1"时, 接收FIFO得以清除, 且SCxRST<RLVL[2:0]>为"000"。并且读指针也被初始化。读作"0"。															
6	RFIS	R/W	选择中断生成条件 0: 当FIFO填充电平(SCxRST<RLVL[2:0]>) = 产生接收中断<RIL [1:0]>的接收FIFO填充电平时 1: 当FIFO填充电平(SCxRST<RLVL[2:0]>) ≥ 产生接收中断<RIL [1:0]>的接收FIFO填充电平时															
5-2	-	R	读作"0"。															
1-0	RIL[1:0]	R/W	产生接收中断的FIFO填充电平 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>半双工</th> <th>全双工</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>4 字节</td> <td>2 字节</td> </tr> <tr> <td>01</td> <td>1 字节</td> <td>1 字节</td> </tr> <tr> <td>10</td> <td>2 字节</td> <td>2 字节</td> </tr> <tr> <td>11</td> <td>3 字节</td> <td>1 字节</td> </tr> </tbody> </table>		半双工	全双工	00	4 字节	2 字节	01	1 字节	1 字节	10	2 字节	2 字节	11	3 字节	1 字节
	半双工	全双工																
00	4 字节	2 字节																
01	1 字节	1 字节																
10	2 字节	2 字节																
11	3 字节	1 字节																

注1: 使用发送/接收FIFO缓存器时, 发送/接收FIFO必须在设置SIO传送模式(半双工/全双工)以及启用FIFO(SCxFCNF<CNFG>="1")之后再清除

11.4.12 SCxTFC(发送 FIFO 配置寄存器)

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

位	比特符号	类型	功能															
31-9	-	R	读作"0"。															
8	TBCLR	W	发送缓存器清除 1: 清除 当SCxTST<TBCLR>设置为"1"时, 该发送缓存器则被清除。读作"0"。															
7	TFCS	W	发送FIFO清除(注 1) 1: 清除 当SCxTST<TFCS>设置为"1"时, 则发送FIFO被清除, 且SCxTST <TLVL[2:0]>为"000"。并且写指针也被初始化。读作"0"。															
6	TFIS	R/W	选择中断生成条件。 0: 当FIFO填充电平(SCxTST<TLVL[2:0]>) = 产生发送中断<TIL [1:0]>的发送FIFO填充电平时 1: 当FIFO填充电平(SCxTST<TLVL[2:0]>) ≤ 产生发送中断<TIL [1:0]>的发送FIFO填充电平时															
5-2	-	R	读作"0"。															
1-0	TIL[1:0]	R/W	引起发生发送中断信号的填充电平。 <table border="1" style="margin-left: 20px;"> <tr> <td></td> <td>半双工</td> <td>全双工</td> </tr> <tr> <td>00</td> <td>空</td> <td>空</td> </tr> <tr> <td>01</td> <td>1 字节</td> <td>1 字节</td> </tr> <tr> <td>10</td> <td>2 字节</td> <td>空</td> </tr> <tr> <td>11</td> <td>3 字节</td> <td>1 字节</td> </tr> </table>		半双工	全双工	00	空	空	01	1 字节	1 字节	10	2 字节	空	11	3 字节	1 字节
	半双工	全双工																
00	空	空																
01	1 字节	1 字节																
10	2 字节	空																
11	3 字节	1 字节																

注1: 要使用发送/接收FIFO缓存器, 发送/接收FIFO必须在设置SIO传送模式(半双工/全双工)以及启用FIFO(SCxFCNF<CNFG> = "1")之后再清除。

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

11.4.13 SCxRST(接收 FIFO 状态寄存器)

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

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

注: 当从SCxBUF读取接收数据时, <ROR>被清除到"0"。

11.4.14 SCxTST(发送 FIFO 状态寄存器)

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

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

注1: 当发送数据写入SCxBUF时, <TUR>被清除到"0"。

11.5 在各模式下的操作

表 11-3 显示各个模式与数据格式。

表 11-3 模式与数据格式

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

模式 0 为同步通信，能用于扩展I/O。该模式与SCLKx同步发送和接收数据。SCLKx既可用于输入，也可用于输出。

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。

奇偶校验位能在模式 1 和模式 2 时添加。模式 3 具有唤醒功能，在该功能中，主控制器经串行链路启动从机控制器(多控制器系统)。

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

11.6 数据格式

11.6.1 数据格式列表

图 11-2 显示数据格式。



图 11-2 数据格式

11.6.2 奇偶校验控制器

奇偶检验位仅可在 7 位或 8 位UART模式中采用发送的数据进行添加。

将SCxCR<PE>设置为"1"就可使能奇偶性。

SCxCR<EVEN>选择偶数奇偶校验或奇数奇偶校验。

11.6.2.1 传送

在数据传送后，奇偶控制电路自动生成奇偶性，数据位于发送缓存器。

奇偶校验位在 7 位UART模式下储存于SCxBUF<TB7>中，在 8 位UART模式下储存于SCxMOD<TB8>中。

<PE>和<EVEN> 的设置必须在数据写入发送缓存器之前完成。

11.6.2.2 接收数据

若接收的数据从接收移位寄存器移到接收缓存器，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>设置为"1"。

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

11.6.3 STOP 位长度

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

11.7 时钟控制

串行时钟(SIOCLK)产生电路如下图所示。在改变串行时钟设置前，请检查设置是否满足AC电气特性。

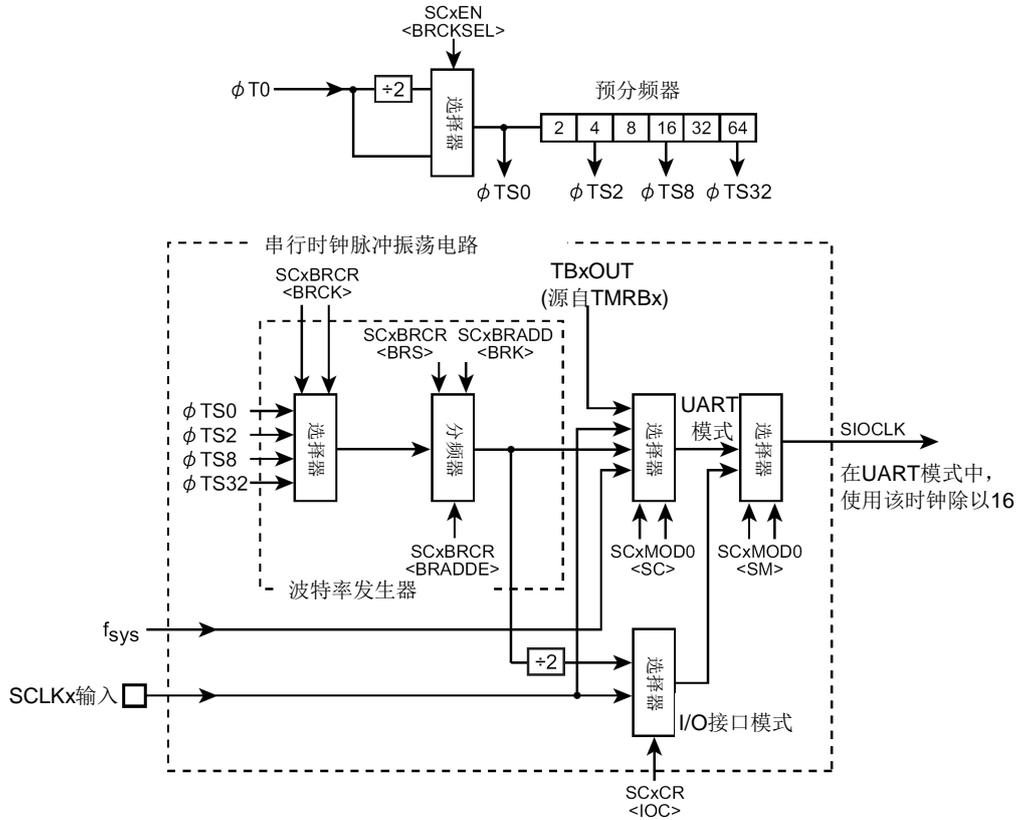


图 11-3 串行时钟生成电路

11.7.1 预分频器

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

11.7.2 串行时钟脉冲振荡电路

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

11.7.2.1 波特率生成器

波特率发生器生成发送和接收时钟，以确定串行通道传送率。

(1) 波特率发生器输入时钟

波特率发生器的输入时钟从预分频器输出中选取，这些输出除以 1, 4, 16 与 64。

该输入时钟通过设置SCxEN<BRCKSEL>与SCxBRCR<BRCK>来选取。

SCxEN<BRCKSEL>		SCxBRCR<BRCK>		波特率发生器 输入时钟 ϕ_{Tx}
0	$\phi_{T0}/2$	00	ϕ_{TS0}	$\phi_{T0}/2$
0		01	ϕ_{TS2}	$\phi_{T0}/8$
0		10	ϕ_{TS8}	$\phi_{T0}/32$
0		11	ϕ_{TS32}	$\phi_{T0}/128$
1	ϕ_{T0}	00	ϕ_{TS0}	ϕ_{T0}
1		01	ϕ_{TS2}	$\phi_{T0}/4$
1		10	ϕ_{TS8}	$\phi_{T0}/16$
1		11	ϕ_{TS32}	$\phi_{T0}/64$

(2) 波特率发生器输出时钟

波特率发生器中的输出时钟的分频比由SCxBRCR和SCxBRADD设置。

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

能选择的分频比如下表所示。

模式	分频功能设置 SCxBRCR<BRADDE>	由N SCxBRCR<BRS>分 频	由K SCxBRADD<BRK> 分频
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：1/N (N=1)频率标度比仅在启用双缓存器的情况下使用。

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

通过N进行分频

$$\text{波特率} = \frac{\phi_{Tx}}{N}$$

N+(16-K)/16 分频

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

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

(1) 在I/O接口模式时的传送时钟

表 11-4 显示I/O接口模式中的时钟选择。

表 11-4 I/O接口模式中的时钟选择

模式 SCxMOD0<SM>	输入/输出选择 SCxCR<IOC>	时钟边沿选择 SCxCR<SCLKS>	使用的时钟
I/O接口模式	SCLK输出	设置为"0"。 固定至上升沿	波特率发生器输出的 2 分频。
	SCLK输入	上升沿	SCLKx引脚输入上升沿
		下降沿	SCLKx引脚输入下降沿

使用SCLKx输入时，以下条件必须满足。

当使用双缓存器时

- SCLK周期 > 6 / fsys

当未使用双缓存器时

- SCLK周期 > 8 / fsys

(2) UART模式中的传送时钟

表 11-5 显示UART模式中的时钟选择在UART模式下,所选时钟在使用前在接收计数器或发送计数器中用16进行分频。

表 11-5 UART模式中的时钟选择

模式 SCxMOD0<SM>	时钟选择 SCxMOD0<SC>
UART模式	定时器输出
	波特率生成器
	f _{sys}
	SCLKx 输入

为了采用SCLK输入,必须满足下列条件。

- SCLK周期 > 2/f_{sys}

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

波特率由下述公式计算所得。

波特率的计算

由CGSYSCR<PRCK[1:0]>选择的时钟脉冲频率

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

↑ 在选择定时器预分频器时钟
 ΦT 1 (2 分频比)的情况下。
 一个时钟周期为定时器触发器
 反向两次的时间。

11.7.3 发送/接收缓存器和 FIFO

11.7.3.1 配置

图 11-4 显示发送缓存器，接收缓存器及FIFO的配置。
使用缓存器和FIFO，必须进行适当的设置。配置可根据模式预先定义。

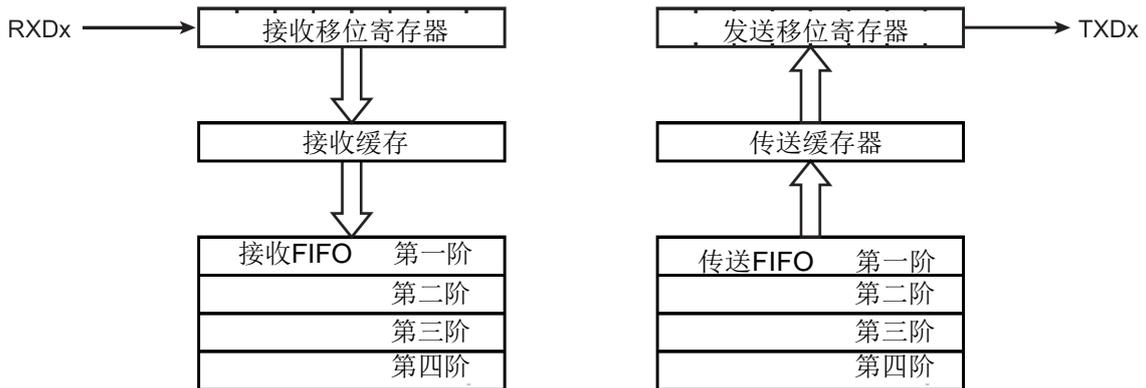


图 11-4 缓存器和FIFO的配置

11.7.3.2 发送/接收缓存

对发送缓存器和接收缓存器进行双缓存。缓存器配置由SCxMOD2<WBUF>指定。
在接收模式的情况下，若在I/O接口模式下设置SCLKx输入或者选择UART模式，不理睬<WBUF>设置进行双缓存。在其他模式时，它按照<WBUF>的设置。

表 11-6 显示各模式和缓存器之间的相关性

表 11-6 模式和缓存器构成

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

11.7.3.3 初始化发送缓存

当因发送缓存器内的某一数据而停止发送时，必须在新传送数据写入发送缓存器前将发送缓存器初始化。

传送操作停止后必须将发送缓存器进行初始化操作。通过读取SCxMOD2<TXRUN>，可确认发送操作的停止情况。在确认发送操作停止后，将SCxTFC<TBCLR>设置为"1"，并初始化发送缓存器。

当启动发送FIFO时，根据发送FIFO中的数据进行初始化操作。若发送FIFO有数据，则数据从发送FIFO传送到发送缓存器。若它没有数据，则SCxMOD2<RBEMP>设置为"1"。

注：在I/O接口模式中的SCLKx输入设置，时钟为异步输入。当发送操作停止时，勿输入时钟。

11.7.3.4 FIFO

除上述双缓存功能外，还能使用 4 字节FIFO。

为了使能FIFO，应通过将SCxMOD2<WBUF>设置为"1"及将SCxFCNF<CNFG>设置为"1"，使能双缓存器。FIFO缓存器配置由SCxMOD1<FDPX[1:0]>指定。

注：使用发送/接收FIFO缓存器时，在设置SIO传送模式(半双工/全双工)并启动FIFO (SCxFCNF<CNFG> = "1")后，发送/接收FIFO必须清除清0。

表 11-7显示各模式和FIFO之间的修正

表 11-7 模式和FIFO构成

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

11.8 状态标志

SCxMOD2寄存器有两类标志。只有在双缓存器使能时，该位才有效。

<RBFLl>是一个显示接收缓存器已满的标志。当收到一帧数据，并且数据从接收移位寄存器移到接收缓存器时，该位变为"1"，而读取该位会使其变为"0"。

<TBEMP>表示发送缓存器为空。当发送缓存器中的数据移到发送移位寄存器时，该位设置为"1"。当数据设置在双缓存器中，则该位被清除到"0"。

11.9 错误标记

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	溢位错误	奇偶错误	成帧错误
I/O接口 (SCLKx输入)	溢位错误	在运行错误 (当使用双缓存器和 FIFO时)	固定到"0"
		固定到"0" (当使用双缓存器和 FIFO时)	
I/O接口 (SCLKx输出)	未定义	未定义	固定到"0"

11.9.1 OERR 标志

在UART和I/O接口模式下，在已读取接收缓存器前，完成下一帧的接收，当产生错误时，该位设置为"1"。若接收FIFO使能，则接收数据自动移到接收FIFO，不会产生溢位错误，直到接收FIFO满(或者直到可用字节被完全占用)。

带SCLK输出的I/O接口模式下，SCLK输出在设置标志后停止。

注：转换I/O接口SCLK输出模式至其它模式，读取SCxCR寄存器并清除过运行标志。

11.9.2 PERR 标志

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

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>设置为"1"。

在I/O接口模式下，当启用双缓存器时，在下列条件下<PERR>设置为"1"。

在SCLK输入模式中，当完成发送缓存器无数据的发送移位寄存器的数据输出后，并输入SCLKx时，将<PERR>设置为"1"。

在SCLK输出模式中，在完成所有数据输出并停止SCLKx输出后，将<PERR>设置为"1"。

注：将I/O接口SCLK输出模式转换至其它模式，读取SCxCR寄存器并清除在运行标志。

11.9.3 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"，则会产生成帧错误。不管SCxMOD2<SBLEN>中停止位长度设置如何，停止位状态只能由 1 来确定。I/O接口模式下，该位固定至"0"。

11.10 接收

11.10.1 接收计数器

接收计数器为 4 位二进制计数器，并由SIOCLK向上计数。

在UART模式下，用 16 个SIOCLK时钟脉冲接收单一数据位，在第 8 个脉冲对数据符号进行采样。接收信号由SIOCLK进行采样，在进行 3 次过滤处理后（该过滤处理采用重合度理论中的延拓定理，因而有效），它被当作接收数据。

11.10.2 接收控制单元

11.10.2.1 I/O 接口模式

在将<IOC> 设置为"0"的SCxCR的SCLK输出模式中，在SCLKx引脚的上升沿进行抽取RXDx引脚样品。

在SCxCR <IOC>设置为"1"的SCLK输入模式下，根据SCxCR <SCLKS>，在SCLKx引脚上升沿或下降沿对RXDx引脚采样。

11.10.2.2 UART 模式

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

11.10.3 接收操作

11.10.3.1 接收缓存器

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

当启用双缓存器时，数据移到接收缓存器(SCxBUF)，接收缓存器满标志(SCxMOD2<RBFL>)设置为"1"。通过读取接收缓存器，接收缓存器全满标志被清除到"0"。当双缓存中止后，接收缓存器标志则没有任何意义。

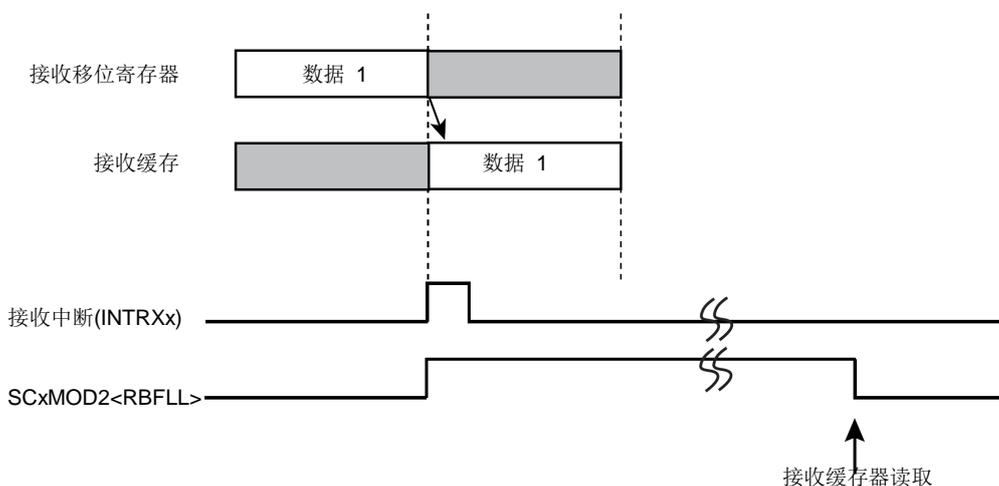


图 11-5 接收缓存器操作

11.10.3.2接收 FIFO 操作

启用FIFO时，接收的数据从接收缓存器移到接收FIFO，接收缓存器全满标志被立即清除。根据 SCxRFC<RIL[1:0]>生成一个中断。

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

在半双工接收模式中的配置和操作说明如下。

- SCxMOD1<FDPX[1:0]> = "01" :传送模式设置为半双工模式
- SCxFCNF<RFST><TFIE><RFIE> :在达到填充电平后，自动禁止连续接收。
- <RXTCNT><CNFG> = "10111" :在接收FIFO中使用的字节数与中断生成填充电平相同。
- SCxRFC<RIL[1:0]> = "00" :FIFO的填充电平生成接收中断设置为 4 字节。
- SCxRFC<RFCS><RFIS> = "01" :清除接收FIFO，并设置中断生成的条件。

在上述FIFO配置设置后，数据接收通过将"1"写入SCxMOD0<RXE>来启动。当所有数据存储在接收移位寄存器，接收缓存器及接收FIFO时，SCxMOD0<RXE>自动清0，接收操作完成。

在上述条件下，连续接收达到填充电平后，可通过在FIFO中连续读取数据。

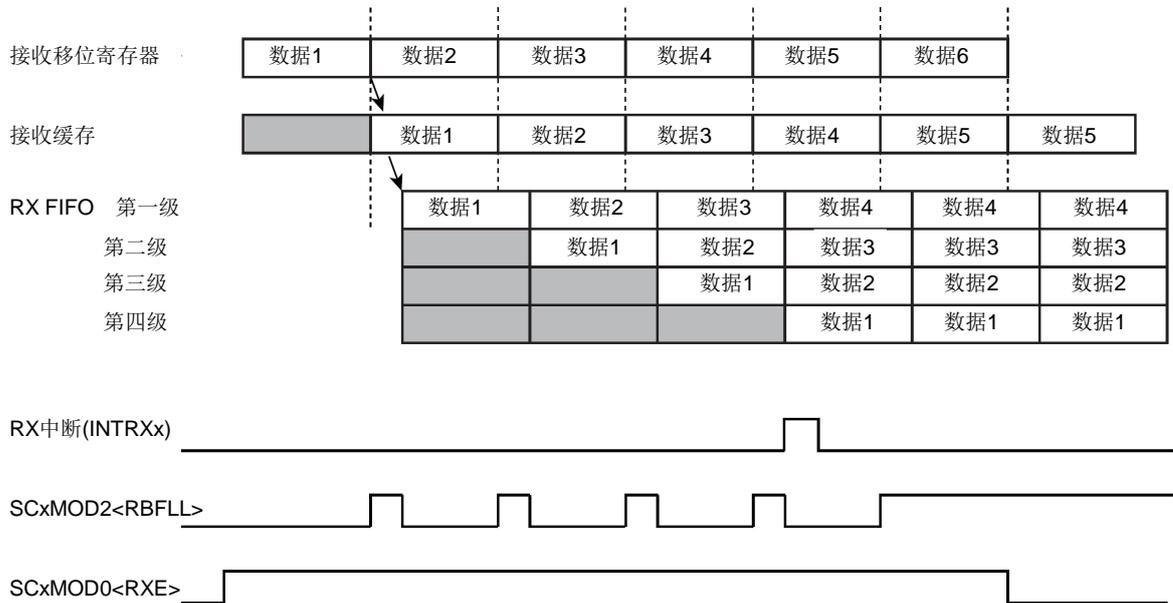


图 11-6 接收FIFO操作

11.10.3.3 带 SCLK 输出的 I/O 接口模式

在I/O接口模式及SCLK输出设置中，当所有接收数据存储在接受缓冲器和FIFO中时，SCLK输出停止。所以，在这种模式下，过运行错误标志没有意义。

SCLK输出停止和再输出的时间取决于接收缓冲器和FIFO。

(1) 单缓冲器的情况

在接收数据后，停止SCLK输出。此模式下，I/O接口通过握手可用传送装置传送各数据。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输出重启。

(3) FIFO的情况

在数据被接收至移位寄存器，接收缓冲器和FIFO后，停止SCLK输出。

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

当SCxFCNF<RXTXCNT>设置为"1"时，SCLK停止，接收操作停止，且清除SCxMOD0<RXE>。

11.10.3.4 读取接收的数据

尽管使能或禁用FIFO，仍会读取从接收缓冲器(SCxBUF)接收的数据。

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

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

11.10.3.5 唤醒功能

在9位UART模式下，将唤醒功能SCxMOD0<WU>设置为"1"，从机控制器就能在唤醒模式下运行。在这种情况下，只有当SCxCR<RB8>设置为"1"位时，会产生干扰INTRX。

11.10.3.6 超限操作错误

当中止FIFO时，在接收下一个数据前未完成数据读取则会发生过运行错误。当发生溢位错误时，虽然接收缓冲器和SCxCR<RB8>中的内容不会丢失，但接收移位寄存器中的内容会丢失。

当FIFO使能时，发生溢位错误，并且当FIFO满时，在下一数据移到接收缓存器前，不读取FIFO，由此设置溢位标志。在这种情况下，FIFO未丢失。

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

注：当模式从I/O接口SCLK输出模式变为其它模式时，读取SCxCR，并清除溢位标志。

11.11 传送

11.11.1 发送计数器

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

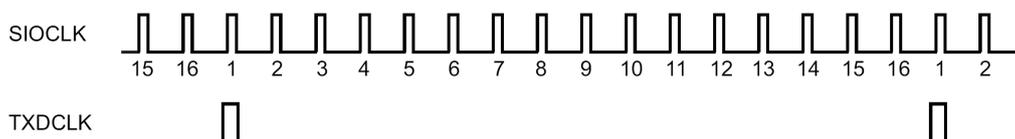


图 11-7 传送时钟的生成

11.11.2 发送控制器

11.11.2.1 I/O 接口模式

在SCxCR<IOC>设置为"0"的SCLK输出模式下，发送缓存器中各位数据在各SCLKx引脚下降沿被输出到TXDx引脚。

在SCxCR<IOC>设置为"1"的SCLK输入模式下，根据SCxCR<SCLKS>，发送缓存器中各位数据在SCLKx引脚下降沿被输出到TXDx引脚。

11.11.2.2 UART 模式

当发送数据写入发送缓存器时，数据传送在下一TXDCLK的上升沿启动，并且也生成发送移位时钟信号。

11.11.3 发送操作

11.11.3.1 传送缓存操作

当双缓存禁用时，CPU将数据仅写入发送移位寄存器，并在数据发送完成后，产生发送中断INTTXx。

当双缓存使能(包括发送FIFO使能的情况)时，发送缓存器中写入的数据移到发送移位寄存器。同时，产生INTTXx中断，发送缓存器空标志(SCxMOD2<TBEMP>)设置为"1"。该标志表示能写入下一发送数据。当下一数据写入发送缓存器时，<TBEMP>标志被清除到"0"。

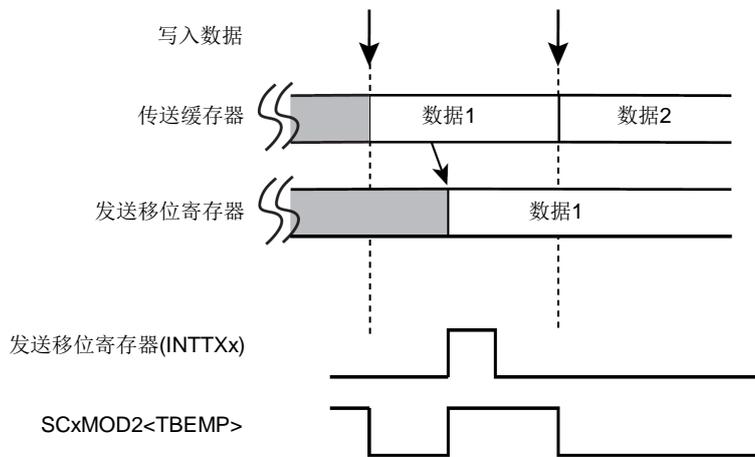


图 11-8 发送缓存器操作(启用双缓存)

11.11.3.2 发送 FIFO 操作

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

注：使用发送FIFO缓存器，在设置SIO传送模式（半双工/全双工）并启用FIFO(SCxFCNF<CNFG>="1")后发送FIFO必须清除。

通过将传送模式设置为半双工而发送 4 字节数据流的设置和操作如下所示。

SCxMOD1<FDPX[1:0]> = " 10 " : 传送模式设置为半双工。

SCxFCNF<RFST><TFIE><RFIE> : 若FIFO为空，传送自动禁用。

<RXTXCNT><CNFG> = " 11011 " : 用于RX FIFO的位数与中断的相同

: 生成填充电平

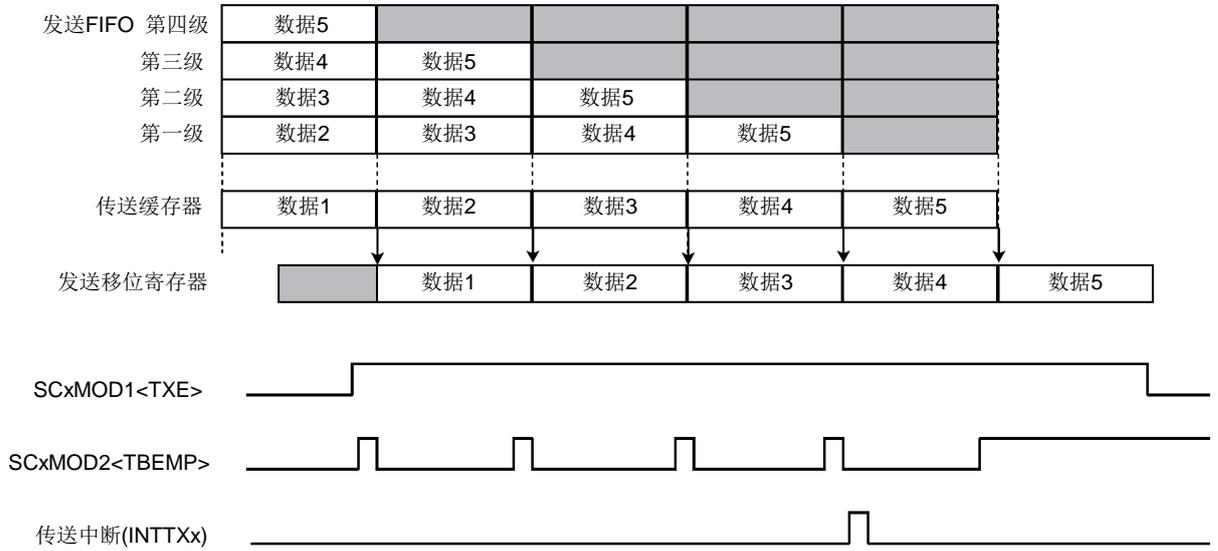
SCxTFC<TIL[1:0]> = " 00 " : 将中断生成填充电平设置为"0"。

SCxTFC<TFCS[1:0]> = " 11 " : 清除接收FIFO，并设置中断生成的条件。

SCxFCNF<CNFG> = " 1 " : 启用FIFO

在配置上述设置后，可以通过将 5 字节数据写入发送缓存器和FIFO，并将SCxMOD1<TXE>设置为"1"位来进行数据传送初始化。当最后一个传送数据移动至发送缓存器，产生传送中断。当最后一个数据完成发送时，时钟停止，发送顺序终止。

一旦配置上述设置，若发送不设置为自动禁用，则发送应持续写入发送数据。



11.11.3.3 带 SCLK 输出的 I/O 接口模式/传送

在I/O接口模式及SCLK输出设置中，当数据传送完成并且发生在运行错误时，SCLK 输出自动停止。

SCLK输出中止和恢复时序随缓存器和FIFO使用情况不同。

(1) 单缓存器

每当转移一帧数据时，SCLK输出停止。可使能各数据与通信另一侧的握手。当下一数据写入缓存器时，SCLK输出恢复。

(2) 双缓存器

当发送移位寄存器和发送缓存器内的数据传送完成后，SCLK输出停止。当下一数据写入缓存器时，SCLK输出恢复。

(3) FIFO

发送移位寄存器，发送缓存器和FIFO中存储的所有数据完成发送，SCLK输出停止。写入下一数据，SCLK输出恢复。

若配置SCxFCNF<RXTXCNT>，则在SCLK停止的同时，清除SCxMOD0<TXE>位，并且传送停止。

11.11.3.4 I/O 接口模式下最后位输出后的 TXD_x 引脚电平

最后位输出后超过时间保持时间的TXD_x引脚电平由SCxCR<TIDLE>指定。

当SCxCR<TIDLE> 为"00"，TXD_x引脚电平为输出 "低"电平。当SCxCR<TIDLE> 为"01"，TXD_x引脚电平为输出 "高"电平当SCxCR<TIDLE> 为"10"，TXD_x引脚电平为最后位的电平。

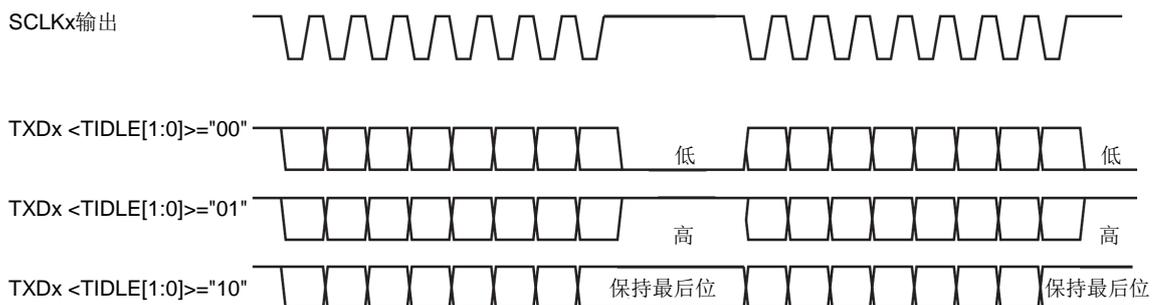


图 11-9 输出最后位的TXD_x 引脚电平

11.11.3.5 在运行错误

在I/O接口SCLK输入模式下，及若FIFO无数据及若在下一帧时钟输入前在发送缓存器中未设置任何数据，并且这在发送移位寄存器的数据完成发送后发生，则发生欠载错误，且SCxCR<PERR>设置为"1"。

TXDx引脚电平可通过SCxCR<TXDEMP>指定。当SCxCR<TXDEMP>为"0"，TXDx引脚在设计输出期间输出"低"电平。当SCxCR<TXDEMP>为"1"时，TXDx引脚输出"高"电平。

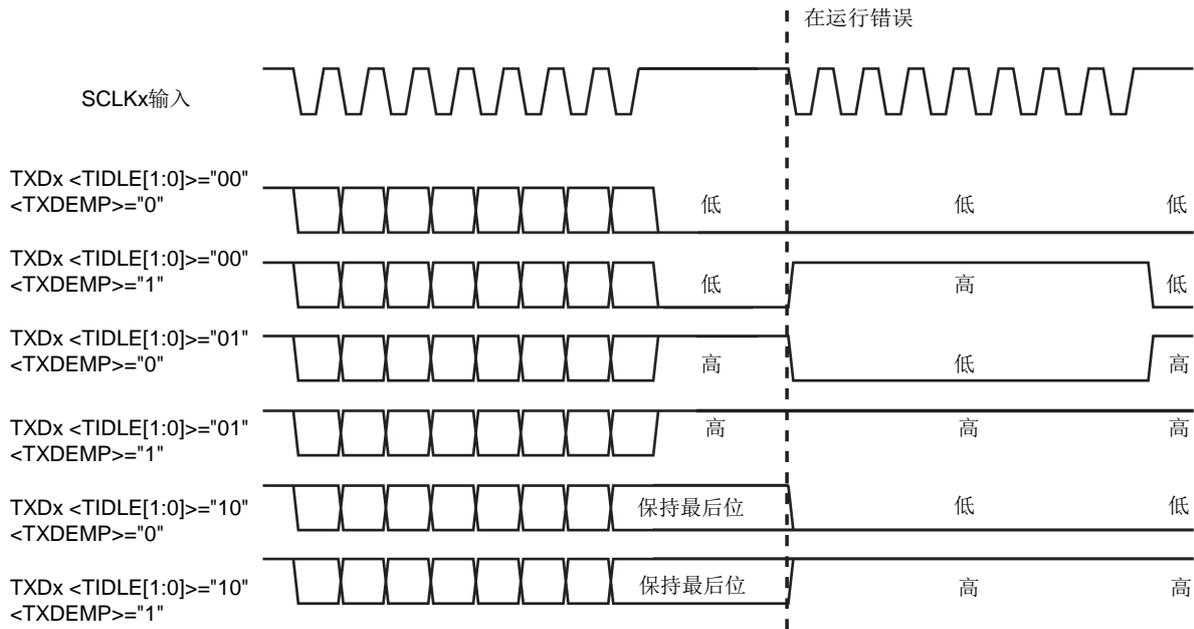


图 11-10 发生在运行错误时的TXDx引脚电平

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

注：在将I/O接口SCLK输出模式转换为其它模式前，读取SCxCR 寄存器并清除在运行标志。

11.11.3.6 I/O 接口 SCLK 输入模式下的数据保持时间

在I/O接口SCLK输入模式中，可通过SCxCR<EHOLD [2:0]>调整最后位的数据保持时间。确定一个数据保持时间和SCLK周期来满足下列公式。

最后位的数据保持时间 ≤ 周期SCLK / 2

11.12 握手功能

握手功能使用CTS (清除发送)引脚启用逐帧数据发送, 并防止溢位错误。功能由SCxMOD0<CTSE>启用或禁用。

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

注1: 传送期间当将 CTS信号设置为 "H" 时, 在完成电流传输后, 下一个数据传输挂起。
 注2: 将CTS 设置为 "L"后, 数据传送在TXDCLK时钟的第一个下降沿开始。

虽然无 $\overline{\text{RTS}}$ 引脚, 但可通过分配 $\overline{\text{RTS}}$ 功能端口的一个位即容易执行握手控制功能。在完成数据接收时(在接收终端路径下)将端口设置为 "高"电平, 可通过请求传送侧将数据传送挂起。

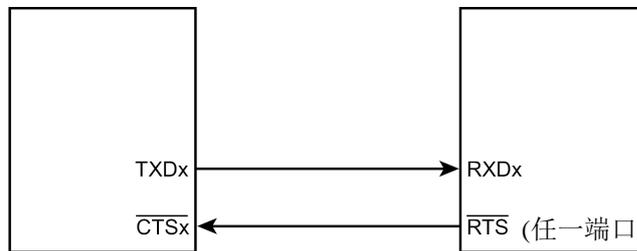


图 11-11 握手功能

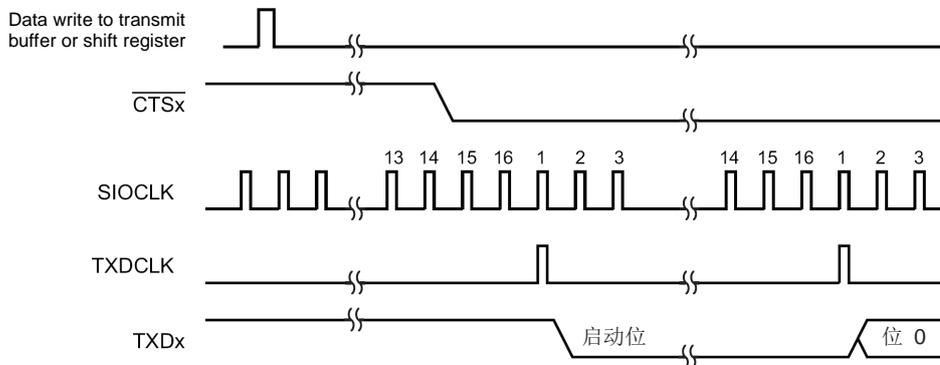


图 11-12 $\overline{\text{CTS}}_x$ 信号时序

11.13 中断/错误生成时序

11.13.1 接收中断

接收操作数据流及读取路径如图 11-13 所示。

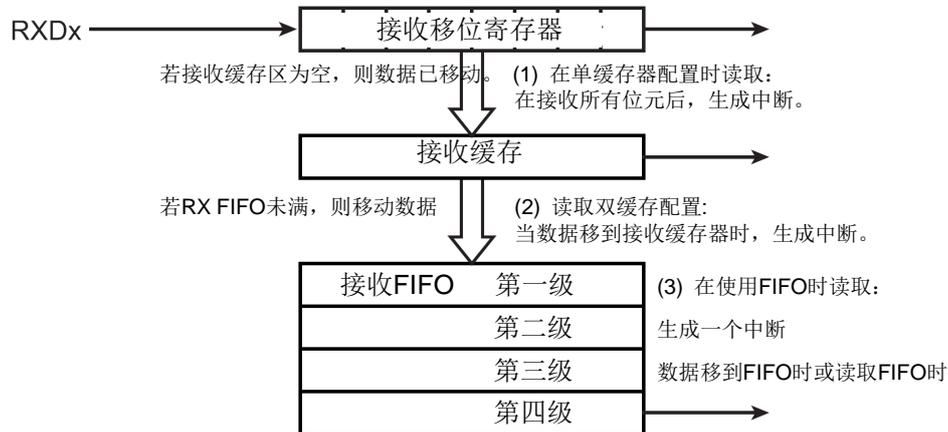


图 11-13 接收缓存/FIFO 配置方块图

11.13.1.1 单缓存/双缓存

根据传输模式和缓存器配置时间生成接收中断，如下表所示。

缓存器配置	UART模式	IO接口模式
单缓存器	-	紧跟SCLKx引脚上升/下降沿之后(上升或者下降根据SCxCR<SCLKS>的设置来确定)
双缓存器	沿第一个停止位中心	紧跟最后一个SCLKx引脚上升/下降沿后(上升或下降根据SCxCR<SCLKS>的设置确定。) 通过读取缓存器，在数据从移位寄存器传输到缓存器后。

注：当发生溢位错误时，不会生成中断。

11.13.1.2 FIFO

使用FIFO时，在下列任一操作及SCxRFC<RFIS>设置已确定的条件下，生成接收中断。

当从接收缓存传输一个接收数据值接收FIFO时。

当从接收FIFO读取一个接收数据时。

中断条件由表 11-8 中所描述的SCxRFC<RFIS> 设置来确定。

表 11-8 使用FIFO时的接收中断的条件

SCxRFC<RFIS>	中断条件
"0"	当FIFO填充电平(SCxRST<RLVL[2:0]>) = 生成接收中断的接收FIFO填充电平 <RIL[1:0]>
"1"	当FIFO填充电平(SCxRST<RLVL[2:0]>) ≥ 生成接收中断的接收FIFO填充电平 <RIL[1:0]>

11.13.2 发送中断

图 11-14 传输操作及读取路径的数据流。

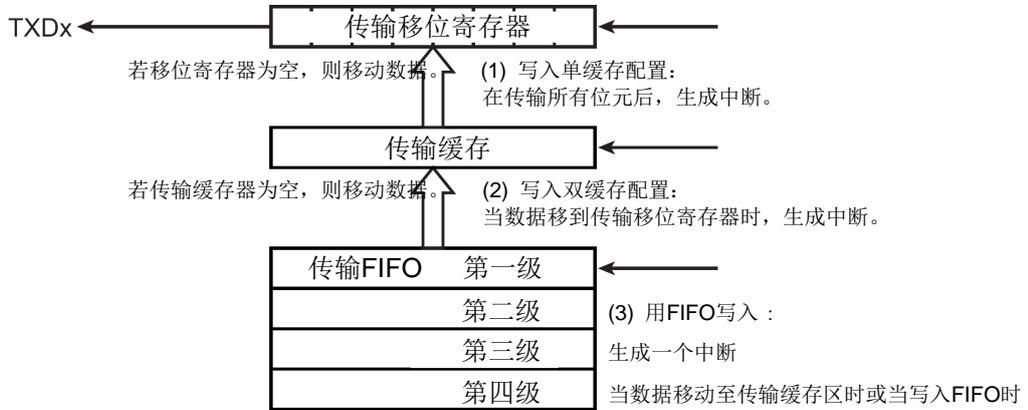


图 11-14 传输缓存/FIFO配置方块图

11.13.2.1 单缓存/双缓存

根据传输模式和缓存器配置时间生成接收中断，如下表所示。缓存生成传输

缓存器配置	UART模式	IO接口模式
单缓存器	恰好在传输停止位前	紧跟最后一个SCLKx引脚上升/下降沿后 (上升或下降按照SCxCR<SCLKS>的设置确定。)
双缓存器	当数据从传输缓存器移到传输移位寄存器时。 当传输移位寄存器为空时，传输中断的生成不取决于SCxMOD1<TXE> 因为写入传输缓存区的数据已经从传输缓存区移动至传输移位寄存器。	

11.13.2.2 FIFO

在使用FIFO时，在下列任一操作及SCxTFC<TFIS>设置已确定的条件下，生成传输中断。

当传输数据从传输FIFO传输至传输缓存区时。

当传输数据写入传输FIFO时

中断条件由表 11-9 中描述的SCxTFC<TFIS>设置确定。

表 11-9 使用FIFO的传输中断条件

SCxTFC<TFIS>	中断条件
"0"	当FIFO填充电平(SCxTST<TLVL[2:0]>) = 生成传输中断<TIL[1:0]>的传输FIFO填充电平
"1"	当FIFO填充电平(SCxTST<TLVL[2:0]>) ≤ 生成传输中断<TIL[1:0]>的传输FIFO填充电平

11.13.3 错误生成

11.13.3.1 UART 模式

模式	9 位	7 位 8 位 7 位+奇偶校验位 8 位 + 奇偶校验位
帧错误溢位错误	沿停止位中心	
奇偶错误	-	沿奇偶校验位中心

11.13.3.2 I/O 接口模式

溢位错误	紧跟最后一个SCLK的上升/下降缘后 (上升或下降按照SCxCR<SCLKS>的设置确定。)
欠载错误	紧跟下一个SCLK上升缘或下降缘后。 (上升或下降按照SCxCR<SCLKS>的设置确定。)

注：在SCLK输出模式中溢位错误和欠载错误没有任何意义

11.14 软件复位

将SCxMOD2<SWRST[1:0]>写作"01"，跟着再写作"10"，即生成软件复位。

结果是，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变为初始状态。其它状态保持不变。

11.15 在各模式下的操作

11.15.1 模式 0 (I/O 接口模式)

模式 0 由两种模式组成，即SCLK输出模式是输出同步时钟，SCLK输入模式是接收同步时钟。各模式下带禁用FIFO的操作说明如下。FIFO见上文所述接收FIFO和传输FIFO。

11.15.1.1 传输数据

(1) SCLK输出模式

- 若传输双缓存器已被禁用($SCxMOD2<WBUF> = "0"$)

数据从TXD_x引脚输出，每次CPU将数据传输缓存区时，时钟从SCLK_x引脚输出。当所有数据被输出时，生成中断INTTX_x。

- 若传输双缓存器已被启用($SCxMOD2<WBUF> = "1"$)

当CPU将数据写入传输缓存器，而数据传输停止时，或当传输缓存器(移位寄存器)的数据传输完成时，数据从传输缓存器移到传输移位寄存器。同时，传输缓存器空标志SCxMOD2<TBEMP>设置为 "1"，并生成INTTX_x中断。

当数据从传输缓存器移到传输移位寄存器时，若传输缓存器无数据移到传输移位寄存器，则不生成INTTX_x中断，SCLK输出停止。

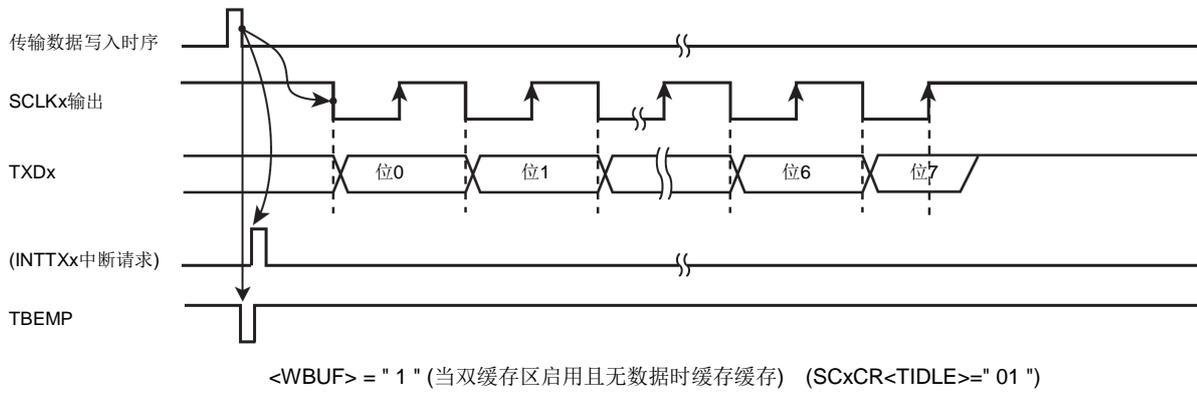
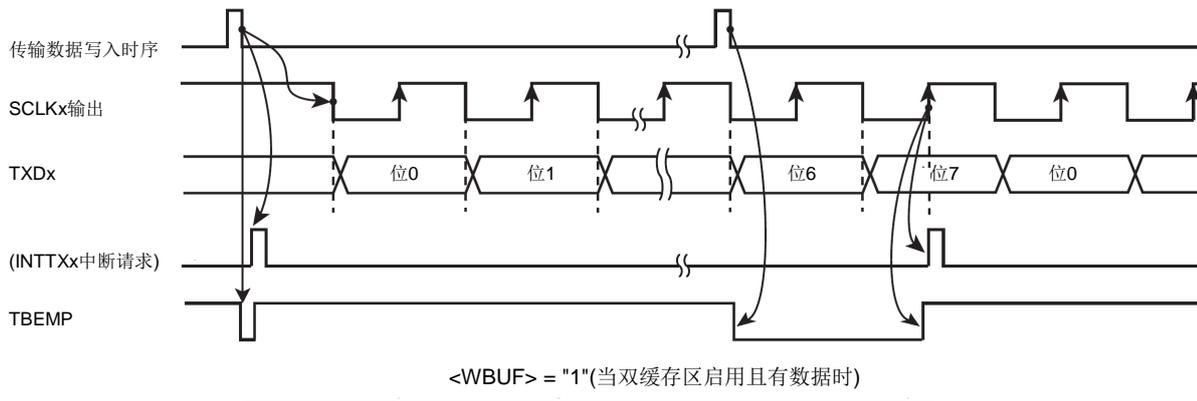
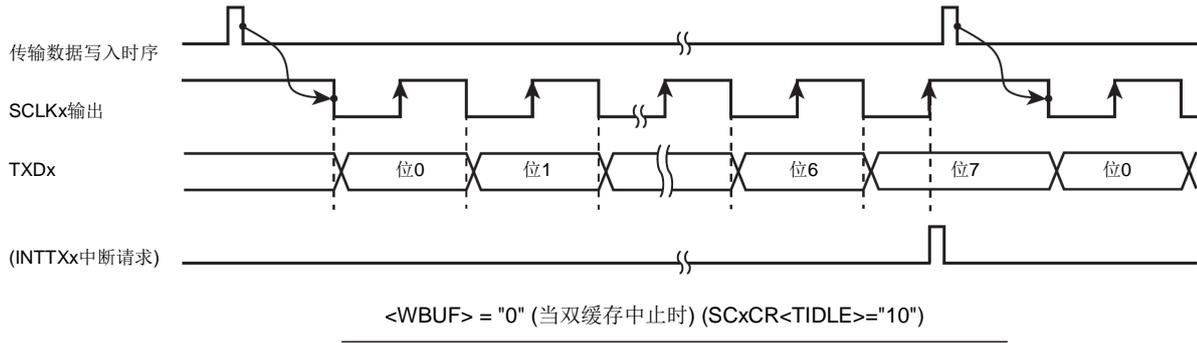


图 11-15 I/O接口模式中的传输操作(SCLK输出模式)

(2) SCLK输入模式

- 若双缓存已被禁用($SCxMOD2<WBUF> = "0"$)

在数据写入传输缓存器的条件下输入SCLK时，从TXD_x引脚输出 8 位数据。当输出所有数据时，生成中断INTTX_x。在图 11-16 所示时序点"A"前，必须写入下一传输数据。

- 若双缓存器已启用($SCxMOD2<WBUF> = "1"$)

当CPU在SCLK输入激活前将数据写入传输缓存器时，或当传输移位寄存器的数据传输完成时，数据从传输缓存器移到传输移位寄存器。同时，传输缓存器空标志 $SCxMOD2<TBEMP>$ 设置为 "1"，生成INTTX_x中断。

若SCLK中断激活，而传输缓存器中无数据，则虽然内部计数器启动，但仍会发生欠载错误，并且将 $SCxCR<TXDEMP>$ 指定的电平输出到TXD_x引脚。

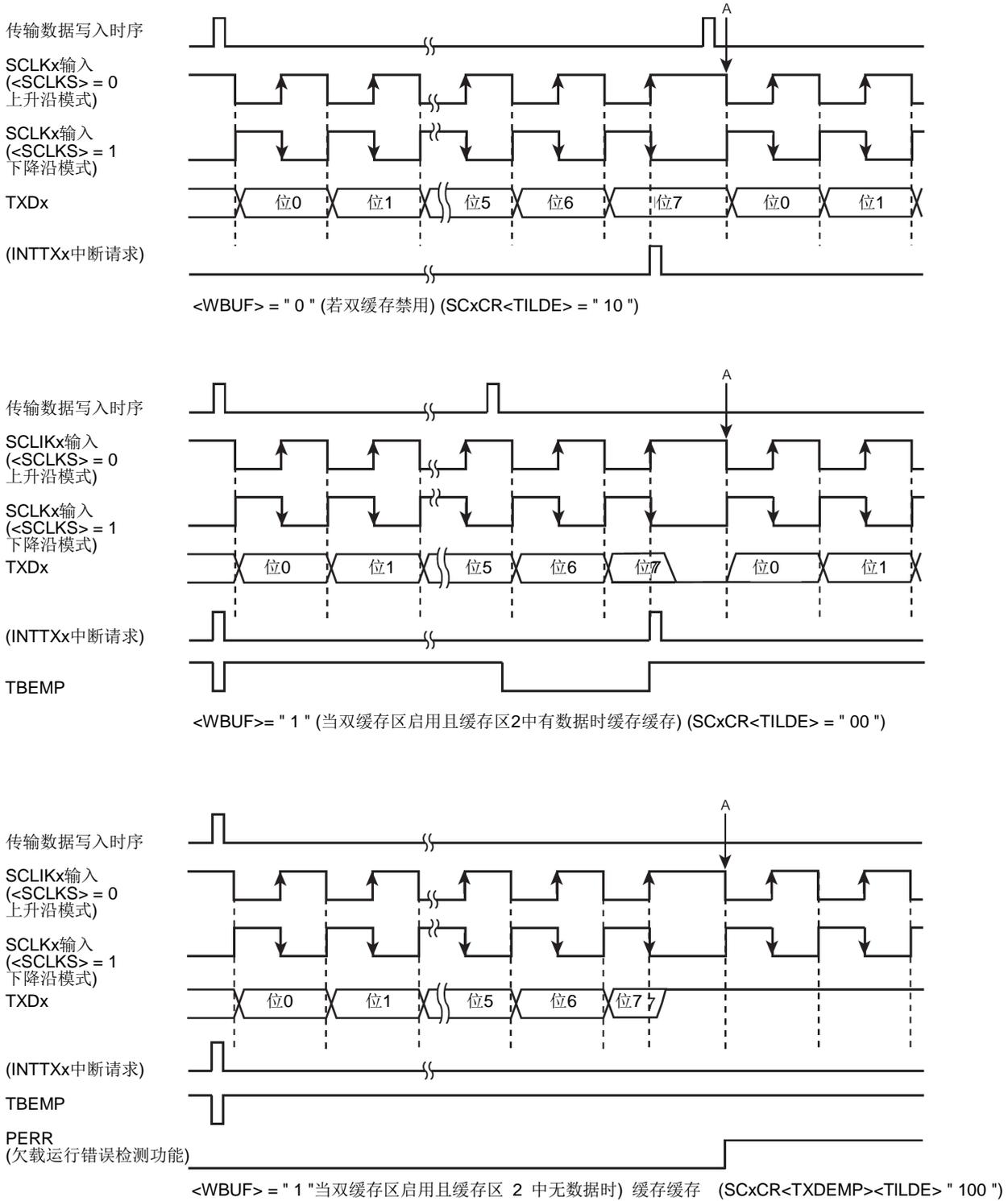


图 11-16 在I/O接口模式(SCLK输入模式)下的传输操作

11.15.1.2接收

(1) SCLK输出模式

将接收启用位SCxMOD0<RXE>设置为 "1", 可启用SCLK输出。

- 若双缓存器已被禁用(SCxMOD2<WBUF> = "0")

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

- 若双缓存器已被启用(SCxMOD2<WBUF> = "1")

移位寄存器中存储的数据移到接收缓存器, 移位寄存器可接收下一帧。数据从移位寄存器移到接收缓存器, 接收缓存器满标志SCxMOD2<RBFL>设置为 "1", 生成INTRXx中断生成。当接收缓存器中有数据时, 若在接下来的 8 位的接收完成前无法读取接收缓存器的数据, 则不生成INTRXx中断, SCLK输出停止。在这种状态下, 读取接收缓存器的数据可使移位寄存器中的数据移到接收缓存器, 因此产生INTRXx中断, 数据接收恢复。

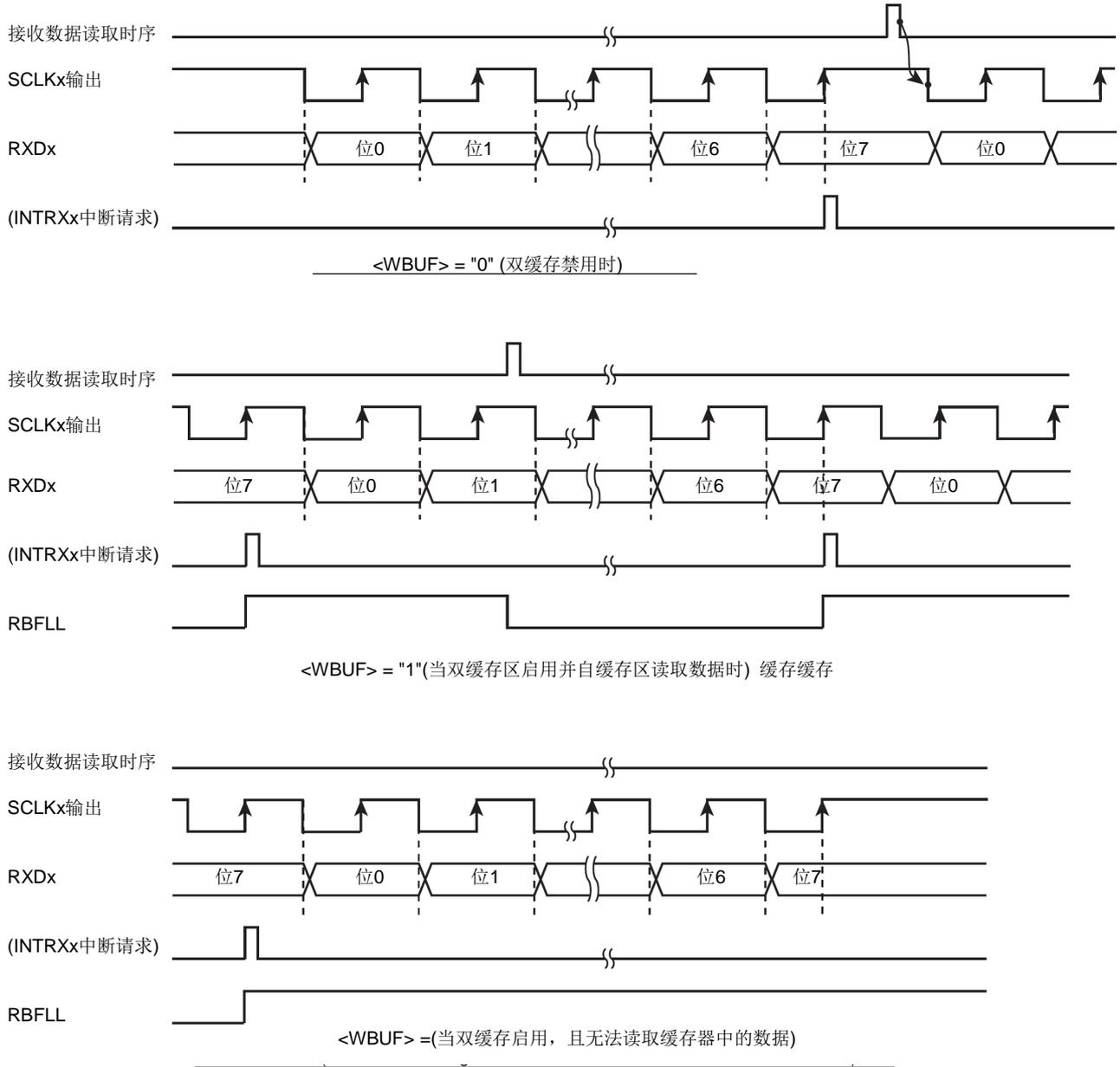


图 11-17 I/O接口模式 (SCLK输出模式)中的接收操作

(2) SCLK输入模式

在SCLK输入模式下，接收双缓存始终启用，接收的帧可从移位寄存器移到接收缓存器，并且接收缓存器能相继接收下一帧。

每一次将接收数据移动到接收缓存区时都会生成INTRXx中断。

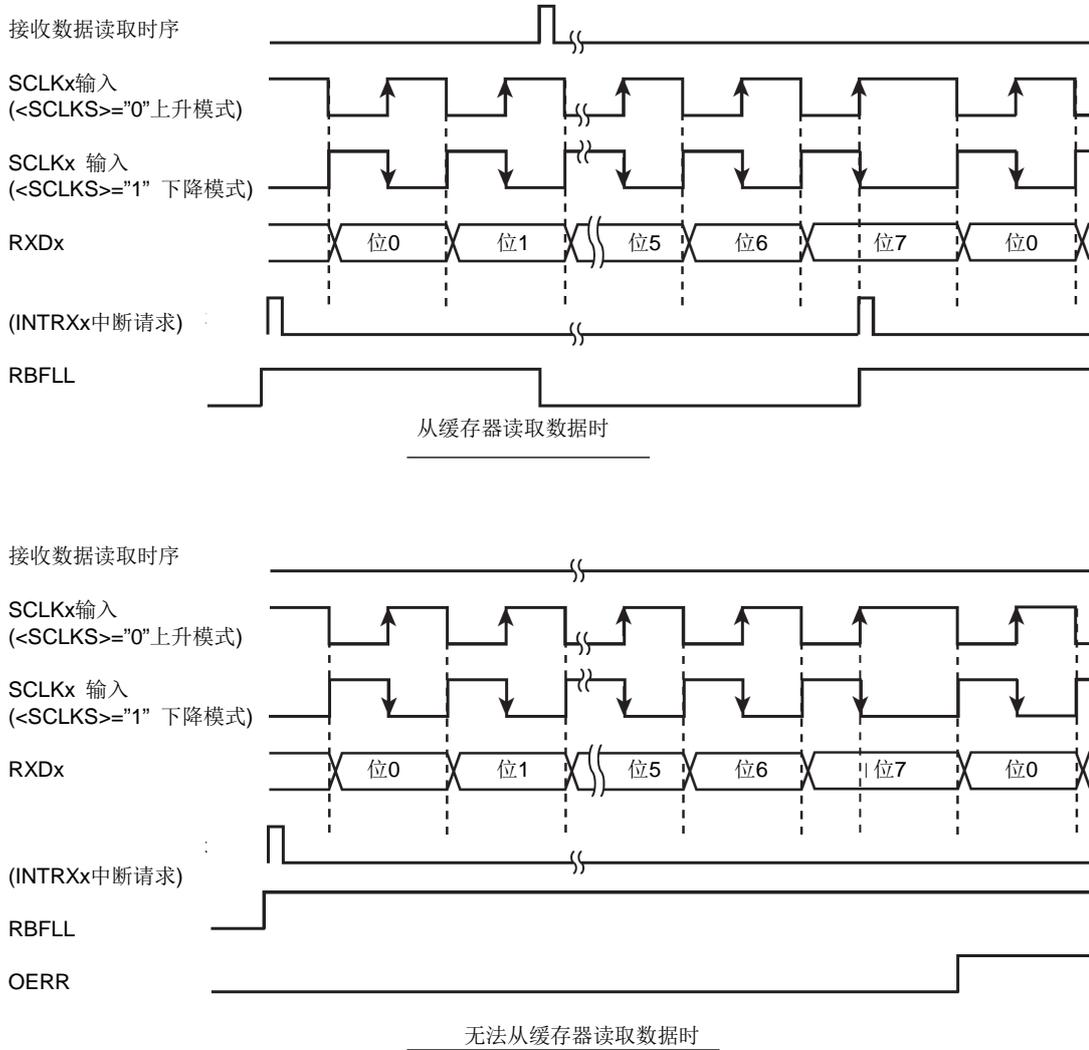


图 11-18 I/O接口模式中的接收操作(SCLK输入模式)

11.15.1.3 传输和接收(全双工)

(1) SCLK输出模式

双缓存中止时 (SCxMOD2<WBUF> = "0")

SCLK当CPU把数据写入传输缓存器时, 输出SCLK。

随后, 8 位数据移入接收缓存器, 生成INTRXx接收中断。同时, 当所有数据完成传输时, 从TXDx引脚输出 8 位写入传输缓存器的数据, 生成INTTXx传输中断。然后, SCLK输出停止。

当接收缓存器的数据被读取, 并且下一传输数据由CPU写入传输缓存器时, 就会开始下一轮数据传输和接收。能自由确定读取接收缓存器和写入传输缓存器的顺序。只有在满足这两个条件时, 数据传输才能恢复。

当双缓存启用(SCxMOD2<WBUF> = "1")时

SCLK当CPU把数据写入传输缓存器时, 输出SCLK。

8 位数据被转移到接收移位寄存器, 移动到接收缓存器, 并生成INTRXx中断。当接收到 8 位数据后, 从TXDx 引脚输出 8 位传输数据。所有数据传输后, 生成INTTXx中断生成, 下一数据从传输缓存器移到传输移位寄存器。

若传输缓存区无数据要移动至传输缓存区(SCxMOD2<TBEMP> = "1")或当接收缓存区已满时(SCxMOD2<RBFL> = "1"), SCLK输出停止。当满足接收数据读取和传输数据写入两种条件时, SCLK输出恢复, 开始下一轮数据传输和接收。

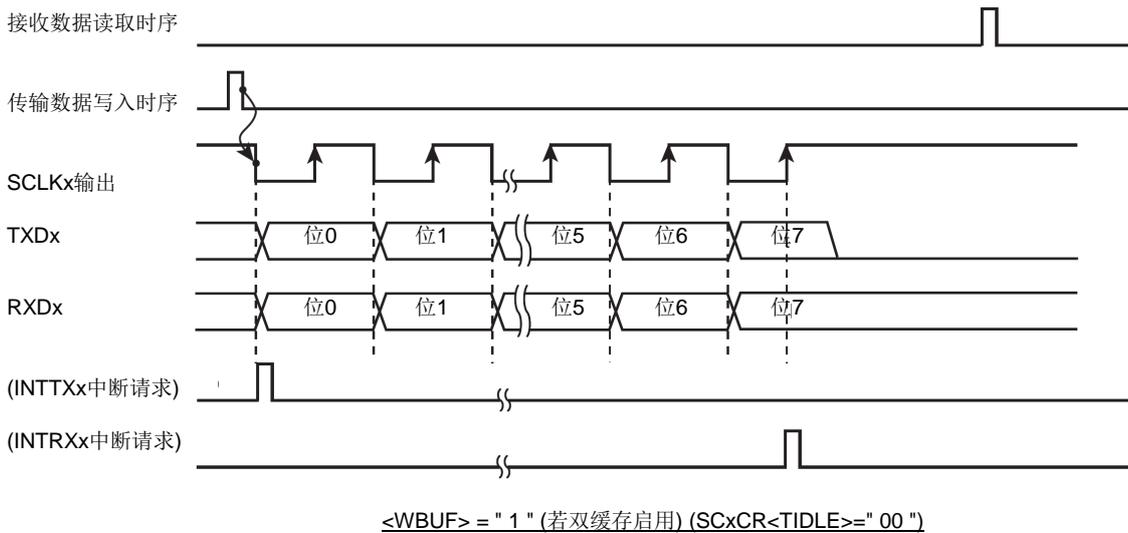
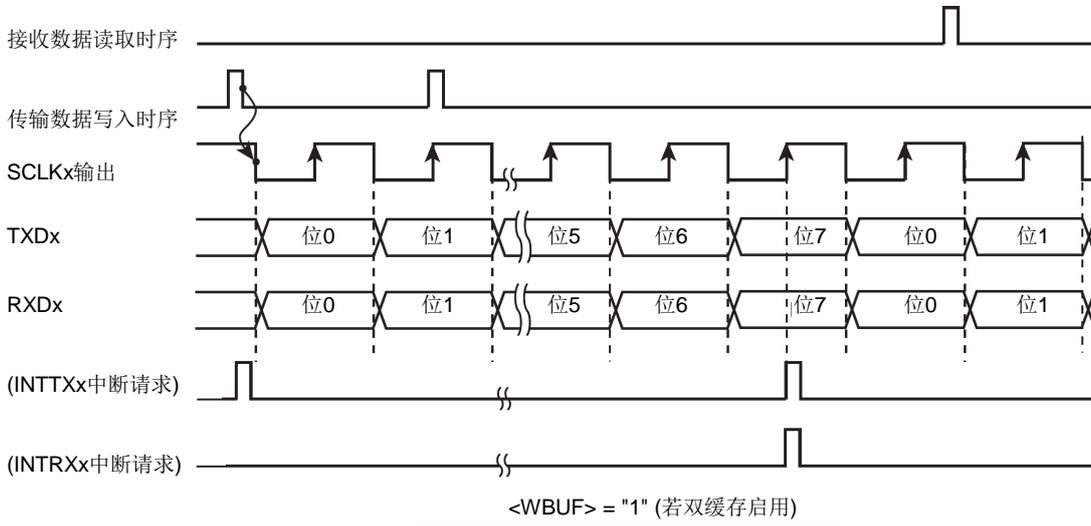
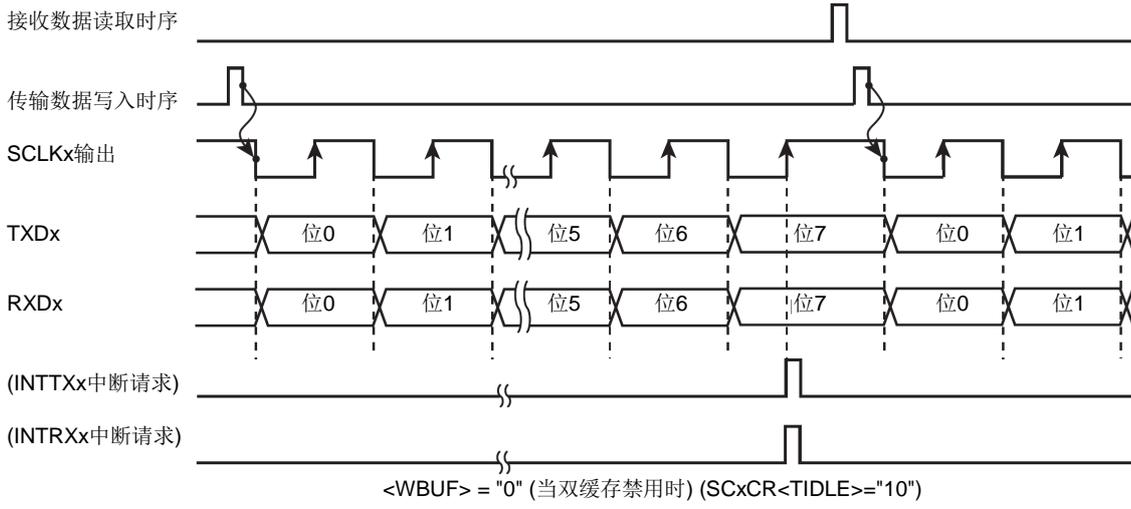


图11-19 I/O接口模式中的传输/接收操作(SCLK输出模式)

(2) SCLK输入模式

当双缓存禁用时。(SCxMOD2<WBUF> = "0")

当接收数据时，在不考虑SCxMOD2 <WBUF>的设置的情况下，总是启用双缓存。

当SCLK输入激活时，写入传输缓存器的8-位数据自TXD_x引脚输出，并且8-位数据移至接收缓存器传输缓存器。数据传输完成后，生成INTTX_x中断即生成。在数据接收完成后，当数据从移位寄存器移到接收缓存器时，生成INTRX_x中断。

请注意在SCLK输入下一帧 (在图 11-20 中A点前必须写入数据)前，必须将传输数据写入传输缓存区。在下一帧数据的接收完成前，必须读取数据。

当启用双缓存时(SCxMOD2<WBUF> = "1")

在完成从传输移位寄存器的数据传输后，当传输缓存区数据移动至传输移位寄存器时，生成中断INTTX_x。同时，接收的数据移入移位寄存器，移到接收缓存器时，生成INTRX_x中断。

请注意在SCLK输入下一帧 (在图 11-20 中A点前必须写入数据)前，必须将传输数据写入传输缓存区。在下一帧数据的接收完成前，必须读取数据。

在下一帧的SCLK输入后，开始传输移位寄存器 传输(在寄存器中，数据已从传输缓存器中移动)，同时接收数据移入接收移位寄存器。

当接收帧的最后一位时，若接收缓存器中的数据未被读取，则发生溢位错误。

当输入下一帧的SCLK时，若无任何数据写入传输缓存器，则发生欠载错误。

SCxCR<TXDEMP> 所指定的水平输出至TXD_x引脚。

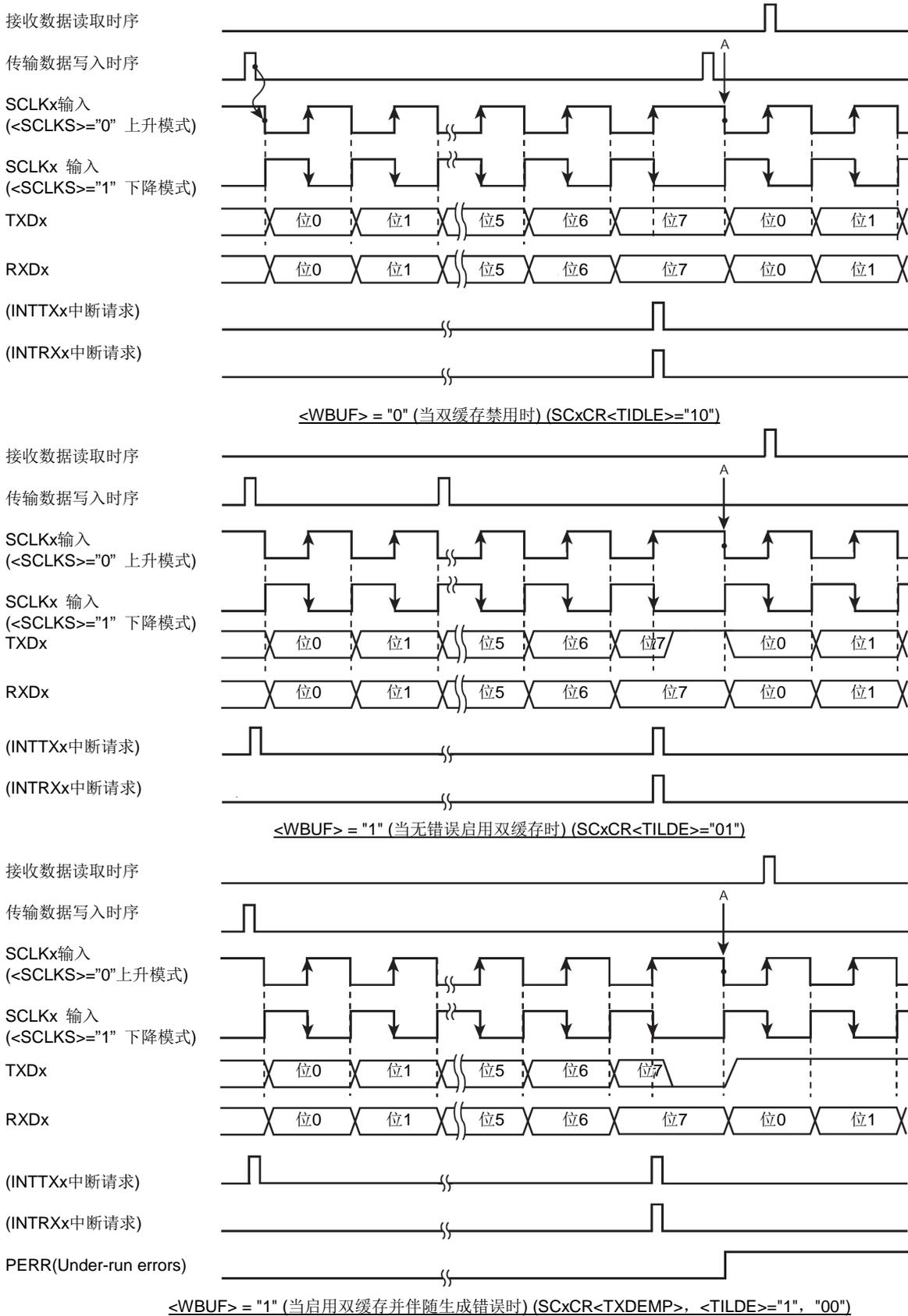
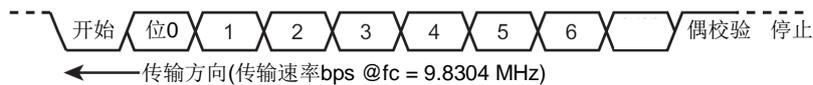


图 11-20 I/O接口模式中的传输/接收操作(SCLK输入模式)

11.15.2 模式 1 (7-位 UART 模式)

通过将串行模式控制寄存器 (SCxMOD<SM[1:0]>)设置为 "01", 即可选择 7-位UART模式。在此模式下, 奇偶校验位可加到传输数据流中; 串行模式控制寄存器 (SCxCR<PE>)控制奇偶校验的启用/禁用设置。当<PE>设置为 "1" (启用)时, 可以用SCxCR<EVEN>位选择偶校验或奇校验。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)

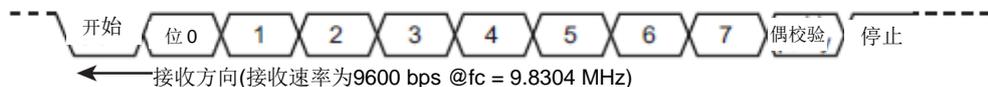
		7	6	5	4	3	2	1	0	
SCxMOD0	← x	0	-	0	0	1	0	1		设置 7-位UART模式
SCxCR	← x	1	1	x	x	x	0	0		偶校验启用
SCxBRCR	←	0	0	1	0	0	1	0	0	设置 2400 bps
SCxBUF	← *	*	*	*	*	*	*	*	*	设置传输数据

x:忽略-:无变化

11.15.3 模式 2 (8-位 UART 模式)

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

按以下格式接收数据的控制寄存器设置如下:



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")。

为了传输数据, 最高位(第 9 位)被写入到SCxMOD0<TB8>中。为了接收数据, 数据存储在SCxCR<RB8>中。

当将数据写入缓存器或从缓存器中读取数据时, 为了写入SCxBUF或从SCxBUF中读取, 必须首先写入或读取最高有效位。

可用SCxMOD2<SBLLEN>指定停止位长度。

11.15.4.1唤醒功能

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

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

注: 从机控制器的TXDx引脚必须用PxOD寄存器设置为开漏输出模式。

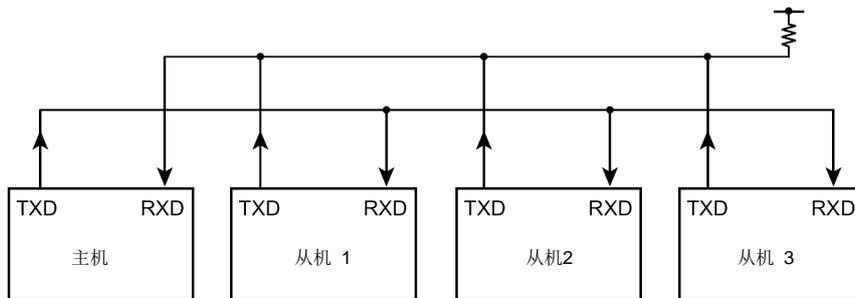
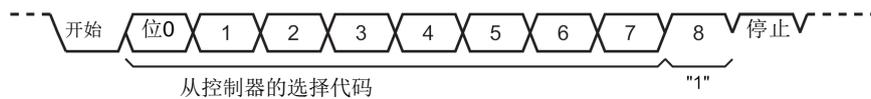


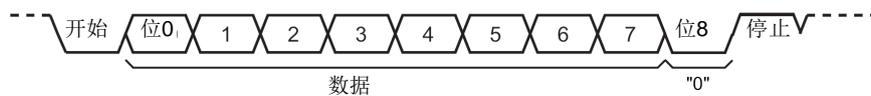
图 11-21 使用唤醒功能的串行链接

11.15.4.2协议

1. 选择主机和从机控制器的 9-位UART模式。
2. 将从机控制器的Set SCxMOD<WU>设置为 " 1 ", 使其准备接收数据。
3. 主机控制器传输包括从机控制器选择代码(8 位)在内的单帧数据。在这种情况下, 最重要的位元(位 8) <TB8> 必须设置为 "1"。



4. 各从机控制器接收上述数据帧；若接收的代码与控制器自己的选择代码匹配，则它将WU位清除到 "0"。
5. 主机控制器将数据传输到指定的从机控制器(SCxMOD<WU>位被清除到 "0" 的控制器)。在这种情况下，最重要的位元(位 8) <TB8> 必须设置为 "0"。



6. 因为最高有效位(第 8 位)<RB8>设置为 "0"，所以<WU>位设置为 "1" 的从机控制器忽略接收数据，因此不生成中断(INTRXx)。此外，<WU>位设置为 "0" 的从机控制器能将数据传输到主机控制器，通知数据已成功接收。

12. 串行总线接口(I2C/SIO):

TMPM375FSDMG包含 1 个串行总线接口(I2C/SIO)通道, 包括下列两种操作模式:

- I2C总线模式(具备多主机能力)
- 时钟同步 8-位SIO模式

在I2C总线模式下, I2C/SIO被通过SCL和SDA与外部设备连接。

在时钟同步 8-位SIO模式下, I2C/SIO通过SCK, SI 和SO与外部设备连接。下表给出了将I2C/SIO设置为各操作模式所需的编程。

表 12-1 使用串行总线接口所需的端口设置

通道	操作模式	引脚	端口功能寄存器	端口输出控制寄存器	端口输入控制寄存器	端口开漏输出控制寄存器
SBI	I2C 总线模式	SCL:PB6	PBFR2[6] = 1	PBCR[6] = 1	PBIE[6] = 1	PBOD[6] = 1
		SDA:PF0	PFFR2[0] = 1	PFCR[0] = 1	PFIE[0] = 1	PFOD[0] = 1
	SCL:PE2	PEFR5[2] = 1	PECR[2] = 1	PEIE[2] = 1	PEOD[2] = 1	
		SDA:PB5	PBFR5[5] = 1	PBCR[5] = 1	PBIE[5] = 1	PBOD[5] = 1
	SIO模式	SCK:PB5	PBFR3[5] = 1	PBCR[6:5] = 01 (SCK0输出)	PBIE[6:5]= 10 (SCK0输出)	PBOD[6:5] = xx
		SI:PB6	PBFR2[6]=1	PBCR[6:5] = 00 (SCK0输入)	出) PBIE[6:5]= 11	
		SO:PF0	PFFR2[0]=1	PFCR[0] = 1	(SCL0输入) PFIE[0] = 0	PFOD[0] = x

注: x:忽略

注: 在I2C总线模式下, 设置PB6&PF0 端口或PE2&PB5 端口。

12.1 配置

配置如图 12-1 所示。

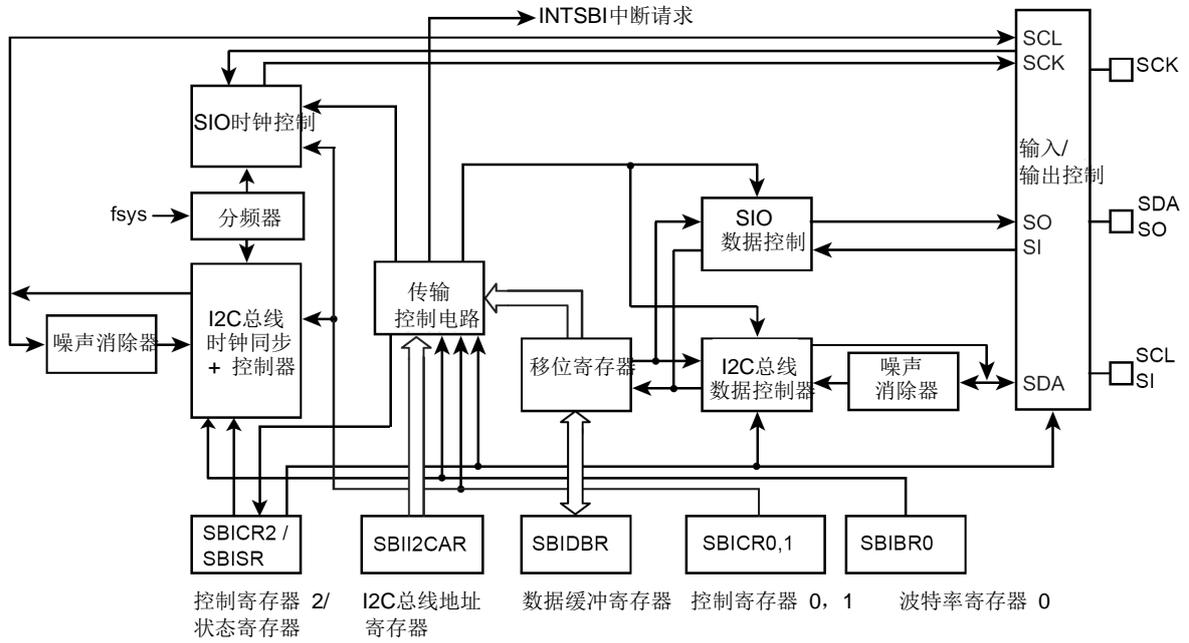


图 12-1 (I2C/SIO) 块接口

12.2 寄存器

以下寄存器控制串行总线接口，并提供监控状态信息。

以下寄存器在不同模式下可执行不同的功能。详见 "12.4 I2C总线模式控制寄存器"和 "12.8 SIO模式控制寄存器"说明。

12.2.1 各通道寄存器

下表给出了各通道寄存器和寄存器地址。

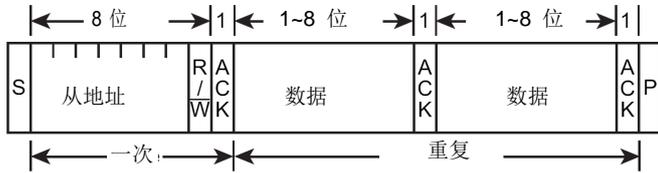
基址= 0x4002_0000

寄存器名称		地址(基+)
控制寄存器 0	SBICR0	0x0000
控制寄存器 1	SBICR1	0x0004
数据缓存寄存器	SBIDBR	0x0008
I2C总线地址寄存器	SBII2CAR	0x000C
控制寄存器 2	SBICR2(写入)	0x0010
状态寄存器	SBISR(读取)	
波特率寄存器 0	SBIBR0	0x0014

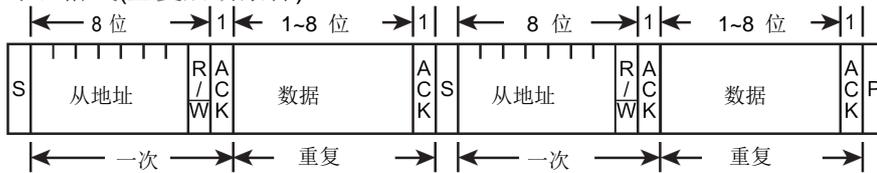
12.3 12C 总线模式数据格式

12C总线模式中使用的数据格式如图 12-2 所示。

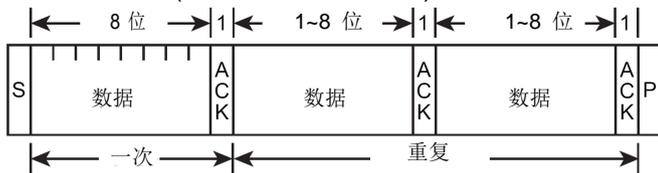
(a) 寻址格式



(b) 寻址格式(重复启动条件)



(c) 自由数据格式(主传输器~从接收器)



(注)启动条件
 R/W:方向位
 ACK:确认位
 P:停止条件

图 12-2 12C总线模式数据格式

12.4 I2C 总线模式控制寄存器

以下寄存器可在I2C总线模式下控制串行总线接口，并提供监控状态信息。

12.4.1 SBICR0(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	串行总线接口操作。 0: 禁用 1: 启用 在使用串行总线接口时，需启用第一个位。 首次设置为启用时，可读取或写入相关的SBI寄存器。 由于禁用此位时除SBICR0外的所有时钟均停止，因此可通过禁用此位直接降低功耗。 若此位在启用后已禁用，则各寄存器的设置均可保留。
6-0	-	R	读作0。

注：在使用串行总线接口时，需先启用此位。

12.4.2 SBICR1(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。																																																	
7-5	BC[2:0]	R/W	选择每次传输时的位数(注 1) <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2"><BC></th> <th colspan="2">当<ACK> = 0</th> <th colspan="2"><ACK> = 1 时</th> </tr> <tr> <th>时钟周期的数目</th> <th>数据长度</th> <th>时钟周期的数目</th> <th>数据长度</th> </tr> </thead> <tbody> <tr><td>000</td><td>8</td><td>8</td><td>9</td><td>8</td></tr> <tr><td>001</td><td>1</td><td>1</td><td>2</td><td>1</td></tr> <tr><td>010</td><td>2</td><td>2</td><td>3</td><td>2</td></tr> <tr><td>011</td><td>3</td><td>3</td><td>4</td><td>3</td></tr> <tr><td>100</td><td>4</td><td>4</td><td>5</td><td>4</td></tr> <tr><td>101</td><td>5</td><td>5</td><td>6</td><td>5</td></tr> <tr><td>110</td><td>6</td><td>6</td><td>7</td><td>6</td></tr> <tr><td>111</td><td>7</td><td>7</td><td>8</td><td>7</td></tr> </tbody> </table>	<BC>	当<ACK> = 0		<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
<BC>	当<ACK> = 0		<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	主机模式 0: 未生成应答时钟脉冲。 1: 生成了应答时钟脉冲。 从机模式 0: 未计数应答时钟脉冲。 1: 已计数应答时钟脉冲。																																																	
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	<table border="1" style="margin-left: 20px;"> <tbody> <tr><td>000</td><td>n = 5</td><td>385 kHz</td></tr> <tr><td>001</td><td>n = 6</td><td>294 kHz</td></tr> <tr><td>010</td><td>n = 7</td><td>200 kHz</td></tr> <tr><td>011</td><td>n = 8</td><td>122 kHz</td></tr> <tr><td>100</td><td>n = 9</td><td>68 kHz</td></tr> <tr><td>101</td><td>n = 10</td><td>36 kHz</td></tr> <tr><td>110</td><td>n = 11</td><td>19 kHz</td></tr> <tr><td>111</td><td></td><td>保留</td></tr> </tbody> </table> <div style="margin-left: 100px;"> } 系统时钟:fsys (= 40 MHz) { 时钟齿轮:fc/1 频率 = $\frac{fsys}{2^n + 72}$ [Hz] </div>	000	n = 5	385 kHz	001	n = 6	294 kHz	010	n = 7	200 kHz	011	n = 8	122 kHz	100	n = 9	68 kHz	101	n = 10	36 kHz	110	n = 11	19 kHz	111		保留																									
000	n = 5	385 kHz																																																		
001	n = 6	294 kHz																																																		
010	n = 7	200 kHz																																																		
011	n = 8	122 kHz																																																		
100	n = 9	68 kHz																																																		
101	n = 10	36 kHz																																																		
110	n = 11	19 kHz																																																		
111		保留																																																		
	SWRMON	R	读取<SWRMON>: 软件复位状态监控 0: 软件复位操作在进行中。 0: 软件复位操作未进行。																																																	

- 注1: 在将运行模式切换为SIO模式前, 将<BC[2:0]>清除至 "000"。
- 注2: 有关SCL线时钟频率的详细信息, 请参考"12.5.1 串行时钟"说明。
- 注3: 复位后, <SCK[0]/SWRMON>位即被读作"1"。然而, 若在SBICR2寄存器处选择SIO模式, 则<SCK[0]>位的初始值为 "0"。
- 注4: 所选频率的初始值为<SCK [2:0]>= 000 , 不受读取的初始值影响。
- 注5: 在主机模式下, 当<BC[2:0]> = "001" 和<ACK> = "0" 时, SCL线路可在生成STOP条件后在SCL线路下降沿固定至 "L", 其它主设备无法使用总线。对于与数个主设备连接的总线, 在生成STOP条件前, 每次传输的位数应设置为大于等于 2 。

12.4.3 SBICR2(控制寄存器 2)

此寄存器通过读取可作为SBISR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	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	选择主机/从机 0: 从机模式 1: 主机模式
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INTSBI中断请求 0: - 1: 清除中断请求
3-2	SBIM[1:0]	W	选择串行总线接口操作模式(注) 00: 端口模式(禁用某个串行总线接口输出) 01: SIO模式 10: I2C总线模式 11: 保留
1-0	SWRST [1:0]	W	软件复位生成 写入"10"再写入"01"以生成复位。详见"12.5.16 软件复位"。

注: 保应确保在通信会话期间, 所有模式不会改变。应确保在将操作模式切换到端口模式前, 总线空闲。应确保在将操作模式从端口模式切换到I2C总线或时钟同步 8-位SIO模式前, 端口处于"高"电平。

12.4.4 SBISR(状态寄存器)

通过向此寄存器写入，使其作为SBICR2使用。

	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	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	主机/从机选择监控程序 0: 从机模式 1: 主机模式
6	TRX	R	传输/接收选择监控程序 0: 接收 1: 传输
5	BB	R	I2C总线状态监视程序 0: 空闲 1: 忙碌
4	PIN	R	INTSBI中断请求监视器 0: 中断请求已生成 1: 中断请求已清除
3	AL	R	仲裁丢失检测 0: - 1: 已检测
2	AAS	R	从地址匹配检测 0: - 1: 已检测 (当检测到一般调用地址时，设置位。)
1	ADO	R	一般调用检测 0: - 1: 已检测
0	LRB	R	最后接收位监视器 0: 最后接收位 "0" 1: 最后接收的位"1"

12.4.5 SBIBR0(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	空闲模式下的操作 0: 停止 1: 操作
5-1	-	R	读作1。
0	-	R/W	务必写入 "0"。

12.4.6 SBIDBR(串行总线接口数据缓存寄存器)

	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)/ W(传输)	接收数据/传输数据

注1: 传输数据必须从MSB(位 7)写入寄存器。接收数据存于LSB中。

注2: 因为SBIDBR有独立的读写缓存区, 所以写入的数据无法读取。因此, 读-修改-写指令(如位处理)无法使用。

12.4.7 SBII2CAR(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作为从机运行时，设置从地址。
0	ALS	R/W	指定地址识别模式。 0: 识别其从地址。 1: 不识别其从地址予(自由数据格式)。

注1: 除了在使用自由数据格式时，请将I2C总线地址寄存器SBII2CAR第 0 位<ALS>设置为 " 0 "。当设置为 "1" 时，它作为自由数据格式运行。选择主机，可固定到传输上。选择拟被固定为接收功能的从机。

注2: 从机模式下，勿将SBII2CAR设置为 " 0x00 "。(若SBII2CAR设置为 " 0x00 "，则识别在从机模式下从地址与接收的I2C标准的START字节(" 0x01 ")匹配。)

12.5 I2C 总线模式下控制

12.5.1 串行时钟

12.5.1.1 时钟源

SBICR1<SCK [2:0]>规定在主机模式下从SCL引脚输出的串行时钟的最大频率。

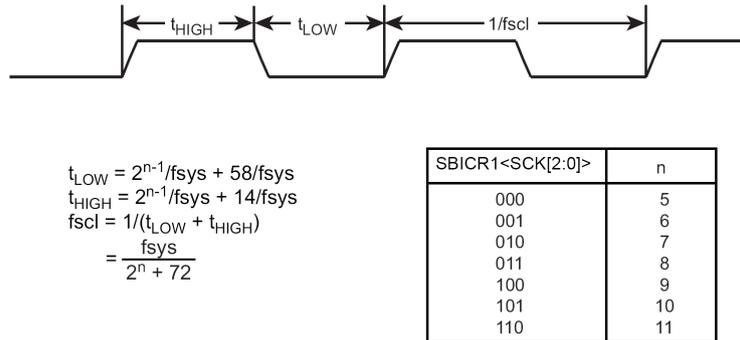


图 12-3 时钟源

注：遵循通信标准，在标准模式和高速模式下，最大速度分别规定为 100 kHz和 400 kHz。注意可根据所使用的fsys和上文给出的计算公式，求出内部SCL时钟频率。

12.5.1.2 时钟同步

因受其引脚结构影响，需采用配线AND连接驱动I2C总线。将其时钟线拖至“低”电平的首个主机，超越在其时钟线上生成“高”电平时其它主机。生成“高”电平的主机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机的总线的正确数据传输。

例如，以下给出了带有两个主机的总线的时钟同步程序。

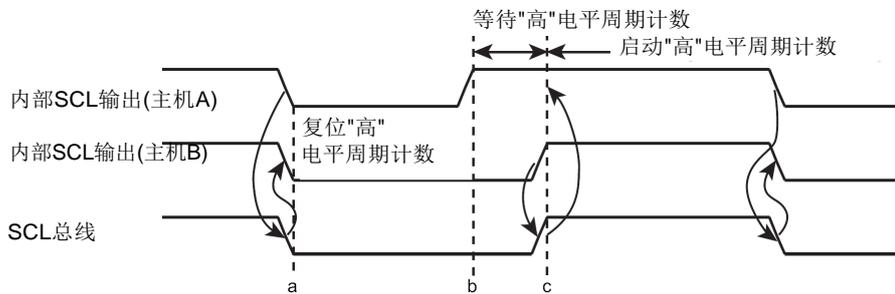


图 12-4 时钟同步示例

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

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

在主机完成"高"电平周期的计数后,主机将拖至"低"电平,SCL总线随之变为"低"电平。

通过这种方法,即可依据已被连接到母线的主机中"高"电平周期最短的主机,以及"低"电平周期最长的主机,确定总线的时钟。

12.5.2 设置应答模式

将SBICR1<ACK>设置为"1",可选择应答模式。作为主机运行时,SBI增加一个应答信号时钟。在从机模式下,可对应答信号的时钟进行计数。在传输器模式下,SBI会在时钟周期期间解除SDAx引脚,以接收来自接收器的应答信号。在接收器模式下,SBI会在时钟周期期间将SDAx引脚拖至"低"电平,并生成应答信号。此外,在从机模式下,若接收到一个一般调用地址,则SBI会在时钟周期期间将SDAx引脚拖至"低"电平,并生成应答信号。然而,根据第二个字节的内容,由软件进行控制生成应答信号,一般调用的第二个字节是必要的。

通过将<ACK>设置为"0",即可触发非应答模式。在运行某个主机时,SBI不会生成应答信号的时钟。在从机模式下,会计数应答信号的时钟。

12.5.3 设置每次传输的位数

SBICR1<BC[2:0]>规定了下一个将要传输或接收的数据的位数。

在启动条件下,<BC[2:0]>设置为"000",由此导致从地址和方向位在8位信息包中传输。在其它时间,<BC[2:0]>保持先前编程的值。

12.5.4 从地址和地址识别模式

将SBII2CAR<ALS>设置为"0",SBII2CAR<SA[6:0]>中的从地址可设置寻址格式,然后SBI识别主设备传输的从地址,并以寻址格式接收数据。

若<ALS>设置为"1",则SBI不识别从地址,以自由数据格式接收数据。在自由数据格式的情况下,不识别从地址和方向位;在生成启动条件后,它们立即被识别为数据。

12.5.5 操作模式

SBICR2<SBIM[1:0]>的设置控制操作模式。在I2C模式下操作时,在将<SBIM[1:0]>设置为"10"前,应确保串行总线接口引脚位于"高"电平。此外,还应确保在将操作模式切换到端口模式前,总线处于空闲状态。

12.5.6 配置 SBI 为传输器或接收器

通过将SBICR2<TRX>设置至 "1"，即可将SBI配置为传输器。通过将<TRX>设置为 "0"，即可将SBI配置为接收器。

在从机模式时：

在数据以寻址格式被传输时。

当接收的从地址与在SBI2CAR的指定值匹配。

在某个一般调用地址被收到时；亦即，启动条件后面的八位全部为零。

当方向位(R/ \bar{W})数值为"1"时，<TRX>通过硬件设置为"1"。若位为"0"，则<TRX>会被设置为"0"。作为主机，SBI可接收来自某个从机的应答。若传输方向位"1"，则硬件将<TRX>设置为"0"。若方向位为"0"，则<TRX>会变为"1"。若SBI不接收应答，则<TRX>保有先前的值。在其检测到总线上的停止条件，或仲裁丢失时，<TRX>即被硬件清除为"0"。若SBI被用于自由数据格式，则硬件不会改变<TRX>。

12.5.7 将 SBI 作为主机模式或从机模式进行配置

通过将SBICR2<MST>设置为"1"，可将SBI配置成主设备进行操作。

通过将<MST>设置为"0"，即可将SBI配置为从机。在其检测到总线上的停止条件，或判优丢失时，<MST>即被硬件清除为"0"。

12.5.8 生成启动和停止条件

当SBISR<BB>为"0"时，将"1"写入SBICR2<MST, TRX, BB, PIN>可导致SBI启动生成启动条件的顺序，并输出在数据缓存器寄存器可能写入的从地址和方向位。必须将<ACK>预先设置为"1"。

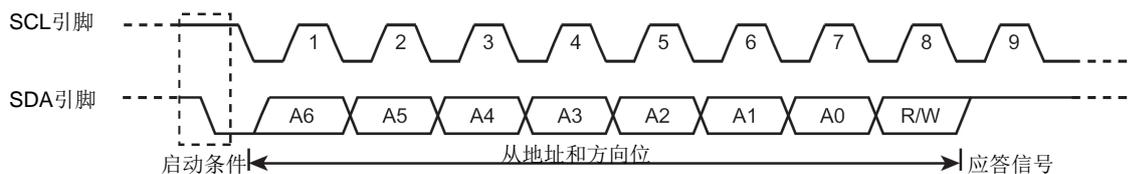


图 12-5 生成启动条件及从地址

当<BB>为 "1" 时，将 "1" 写入<MST, TRX, PIN>及将 "0" 写入<BB>可造成SBI启动在总线上生成停止条件的顺序。在总线上出现停止条件前，不应改变<MST, TRX, BB, PIN>的内容。

若SCL总线在停止条件生成时被其它设备拖至"低", 则停止条件会在SCL线路被解除时生成。

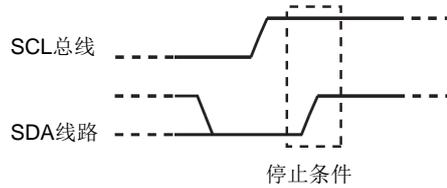


图 12-6 生成停止条件

可读取SBISR<BB>, 以检查总线状态。在总线上检测到启动条件时(总线忙), 将<BB>设置为"1"; 检测到停止条件时(总线空闲)将<BB>清"0"。

12.5.9 中断服务请求和释放

在主机模式下, 当完成由<BC>和<ACK>设置的时钟周期数的传输后, 生成串行总线接口请求(INTSBI)。

在从机模式下, 在以下条件下将生成INTSBI。

在输出应答信号后。应答信号是在接收的从地址与SBII2CAR<SA[6:0]>所设从地址匹配时生成。在接收到一般调用地址时应答信号被生成后。

在接收某个一般调用地址后从地址可匹配或某个数据传输完成时。

在从地址识别模式下(<ALS> = "0"), 当接收的从地址与SBII2CAR规定值匹配时或当接收一般调用(启动条件后的8位数据均为"0")时, 生成INTSBI。

当生成中断请求(INTSBI)时, SBICR2<PIN>被清除至"0"。在<PIN>被清除至"0"的同时, SBI会将SCL线路拖至"低"电平。

当数据写入SBIDBR或从SBIDBR中读取时, <PIN>设置为"1"。在<PIN>被设置为"1"后, SCL线路的解除需花费一段时期tLOW。在程序将"1"写入到<PIN>时, 其即被设置为"1"。然而, 写入"0"不会将该位清除至"0"。

注: 当从地址和方向位在主机模式下因传输而发生仲裁时, <PIN>被清除到"0", 并发生INTSBI。这与从地址是否与<SA>匹配无关。

12.5.10 仲裁丢失检测监视器

I2C总线具备带多主设备的能力(一条总线上有两台或多台主设备), 并且要求执行总线仲裁程序, 确保数据传输正确。

在总线繁忙时试图生成启动条件的主设备丢失总线仲裁, SDA总线和SCL总线上不出现启动条件。I2C总线在SDA总线上进行仲裁。

一条总线上的两个主机的仲裁程序如以下所述。

在到达点a前，主机A和主机B输出相同的数据。在点a时，主机A输出"低"电平，主机B输出"高"电平。然后，主机A将SDA总线拖至"低"电平，原因是线路具备配线AND连接。在SCL线路在点b变为高时，从机会读取SDA线路数据(即主机A所传输的数据)。此时，主机B所传输的数据变为无效。

主机B的这种情况被称为"仲裁丢失"。主机B释放其SDA引脚，使其不影响另一个主机发起的数据传输。若两个或更多主机已经传输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程序。

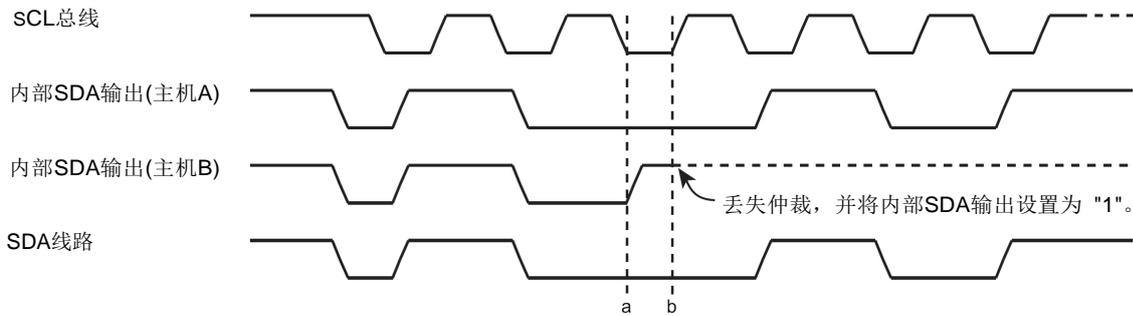


图 12-7 丢失的仲裁

主机会比较SDA总线电平和SCL线路上升部位的内部SDA输出电平。当这两个值不同时，仲裁丢失出现，将SBISR<AL>设置为"1"。

当出现仲裁丢失时，SBIxSR<MST>和<TRX>被清除至"0"，导致SBI作为从机接收器运行，并且在数据传输时，它停止时钟输出。若传输从地址和方向位的主设备出现仲裁丢失，则它接收作为从设备的其它主设备传输的从地址和方向位，不管接收的从地址是否与<SA>匹配。

<PIN>清至"0"，出现INTSBI。

当向/从SBIDBR写入/读取数据时或向SBICR2写入数据时，<AL>清"0"。

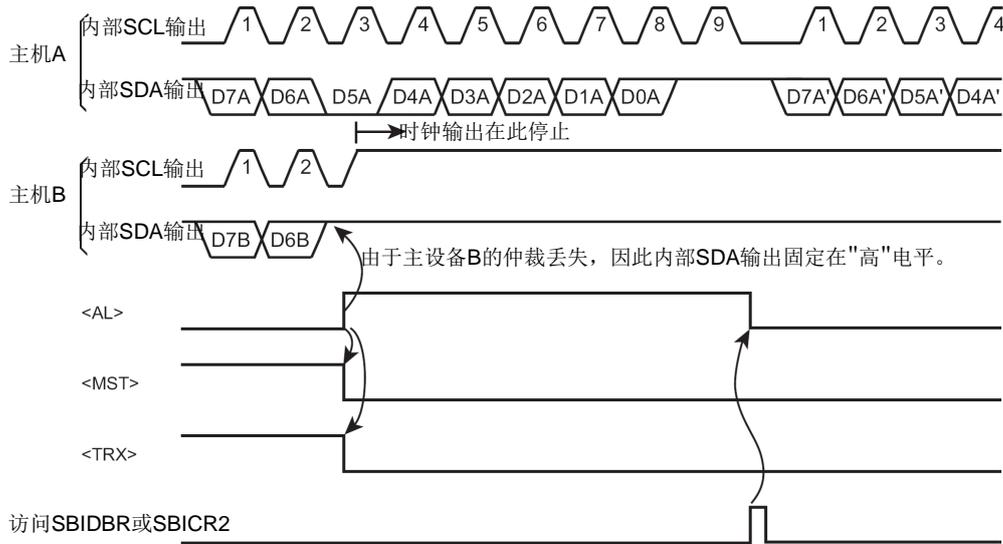


图 12-8 主机B仲裁丢失示例(D7A = D7B, D6A = D6B)

12.5.11 从地址匹配检测监视器

当SBI作为从设备运行时(SBII2CAR<ALS>="0")，在接收到一般呼叫地址或与在SBII2CAR的指定值匹配的从地址时，SBISR<AAS>设置为"1"。

在<ALS>为"1"时，<AAS>在首个数据字被接收时被设置为"1"。当数据写入SBIDBR或从SBIDBR中读取时，<AAS>清"0"。

12.5.12 一般调用检测监视器

当SBI作为从设备运行时，当SBISR<ADO>接收到一般呼叫地址时，SBISR<ADO>设置为"1"；即，在启动条件后的八位均为零。

在母线上的启/停条件被检测到时，<ADO>即清"0"。

12.5.13 最后接收位监视器

将SBISR<LRB>设置为在SCL线的上升沿读取到的SDA线值。

在应答模式下，若在INTSBI中断请求生成后立即读取SBISR<LRB>，导致ACK信号被读取。

12.5.14 数据缓存器寄存器(SBIDBR)

读取或写入SBIDBR启动读取接收到的数据或写入被传输数据。

在SBI作为主机运行时，通过将某个从地址和某个方向位设置为寄存器，可生成启动条件。

12.5.15 波特率寄存器(SBIBR0)

SBIBR0<I2SBI>寄存器判断SBI在进入IDLE模式时是否运行。
在执行一个指令以切换到待机模式前，必须对寄存器进行编程。

12.5.16 软件复位

若串行总线接口电路因外部噪声而锁定，则通过使用软件复位实现其初始化。
先将 "10"，再将 "01" 写入SBICR2<SWRST[1:0]>，就能生成使串行总线接口电路初始化的复位信号。对于I2C总线模式，当写入SBIxCR2<SWRST[1:0]>时，将SBIxCR2<MST><TRX><BB><PIN>设置为 "0000"，将SBIxCR2<SBIM[1:0]>设置为 "10"。在复位后，所有控制寄存器和状态标志均会被初始化为其复位值。当串行总线接口初始化时，<SWRST>被自动清 "0"

注：软件复位将导致SBI运行模式从I2C模式切换为端口模式。

12.6 在 I2C 总线模式下数据传输程序

12.6.1 设备初始化

首先，设置SBICR1<ACK><SCK[2:0]>。将<ACK>设置为 "1" 以规定应答模式。将SBICR1<BC[2:0]>设置为 "000"。

其次，将<SA[6:0]>(从地址)和<ALS>设置为SBII2CAR。(在寻址格式模式下，设置<ALS> = "0")。最后，将串行总线接口配置为从机接收器，保证串行总线接口引脚首先为"高"。然后将SBICR2<MST><TRX><BB>设置为 "000"，将<PIN>设置为 "1"，将<SBIM[1:0]>设置为 "10"，将<SWRST[1:0]>设置为 "00"。

注：串行总线接口电路的初始化必须在总线连接的所有装置初始化后任何装置均未生成启动条件的时间内完成。若不遵循该规则，则可能不会正确地接收数据，原因是其它设备可能在串行总线接口电路的初始化完成前已经开始传输。

	7	6	5	4	3	2	1	0	
SBICR1	← 0	0	0	1	0	x	x	x	指定ACK和SCL时钟。
SBII2CAR	← x	x	x	x	x	x	x	x	指定一个从地址和一种地址识别模式。
SBICR2	← 0	0	0	1	1	0	0	0	将SBI配置为从属接收器。

注:X: 忽略

12.6.2 生成启动条件和从地址

12.6.2.1 主机模式

在主机模式下，必须执行以下步骤，以生成启动条件和一个从地址。

首先，确保总线处于空闲状态h(<BB> = "0")。然后，将 "1" 写入SBICR1<ACK>，以选择应答模式。将传输的从地址和方向位写入SBIDBR。

当<BB> = "0"时，将 "1111" 写入SBICR2<MST, TRX, BB, PIN>，可在总线上生成启动条件。

在启动条件后，SBI生成SCL引脚的 9 个时钟。SBI用最先的 8 个时钟输出SBIDBR规定的从地址和方向位，在第 9 个时钟释放SDA线，以接收从机设备的应答信号。

INTSBI中断请求在第 9 个时钟的下降沿生成，<PIN>被清 "0"在主机模式下，SBI使SCL线路保持在"低"电平，同时<PIN> = "0"。若从机设备已返回应答信号，则在INTSBI中断请求生成时，<TRX>根据传输的方向位改变其值。

注：输出从地址时，应用软件检查总线在写入SBIDBR前是自由的。若不遵循此规则，则正在总线上输出的数据可能会被破坏。

主路径中的设置



SBICR1 ← X X X 1 0 X X X 选择应答模式。
 SBIDBR ← X X X X X X X X 指定所需的从地址和方向。
 SBICR2 ← 1 1 1 1 1 0 0 0 生成启动条件。
 INTSBI0中断程序示例
 解除中断请求。
 处理
 中断结束

12.6.2.2 从机模式

在从机模式下，SBI接收到该启动条件和一个从地址。
 在接收主设备传输的启动条件后，在SCL线上，SBI在前 8 个时钟接收主设备传输的从地址和方向位。
 若接收的地址与SBII2CAR指定的其从地址匹配或等于一般调用地址，则在第 9 个时钟，SBI将 SDA线路拉至"低"电平。
 在第 9 个时钟的下降沿，生成INTSBI中断请求，<PIN>被清 "0"。在从机模式下，SBI将SCL线路保持在"低"电平，同时<PIN>为 "0"。

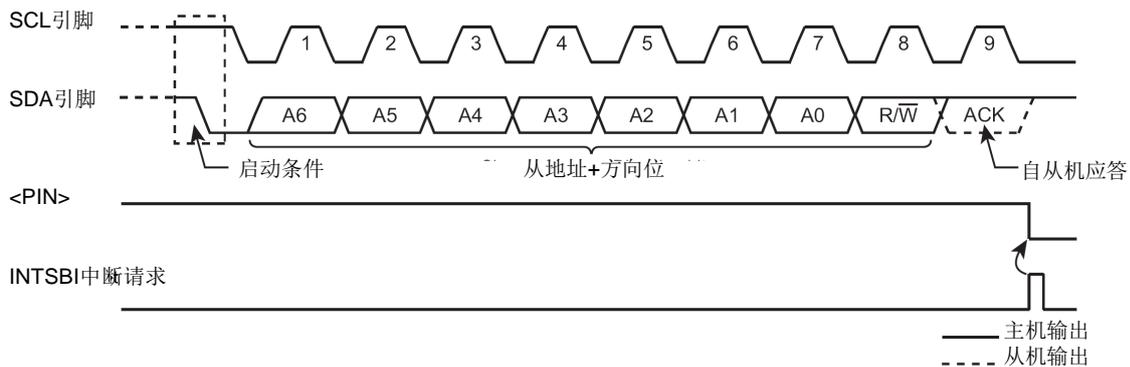


图 12-9 启动条件的生成及从地址

12.6.3 传输数据字

一个数据字传输结束时，INTSBI中断生成以测试<MST>，以判断SBI是在主机模式下还是在从机模式下。

12.6.3.1 主机模式(<MST> = "1")

测试<TRX>确定SBI是否被配置为传输器或接收器。

(1) 传输器模式(<TRX> = "1")

测试<LRB>。若<LRB>为"1"，则表示接收器不要求提供更多的数据。

然后，主机生成下文所述的停止条件，以停止传输。

若<LRB>为"0"，则意味着接收器需要进一步的数据。若下一传输数据有 8 位，则数据写入SBIDBR。若数据长度不同，则<BC[2:0]>和<ACK>被编程，传输数据写入SBIDBR。写入数据，可使<PIN>设置为"1"，导致SCL引脚生成用于传输下一数据字的串行时钟，并导致SDA引脚传输数据字。

传输完成后，生成INTSBI中断请求，<PIN>清"0"，且SCL引脚被拉至"低"电平。

在传输更多的数据字时，需重新测试<LRB>，并重复以上程序。

INTSBI中断

若MST = 0

则转到从机模式处理。

若TRX = 0

则转到接收器模式处理。

若LRB = 0

则转到停止条件生成处理。

SBICR1 ←XXXX0XXX 指定拟传输的位的数目，并指定是否需要ACK。

SBIDBR ←XXXXXXXX 写入传输数据。中断处理结束。

注：X；忽略

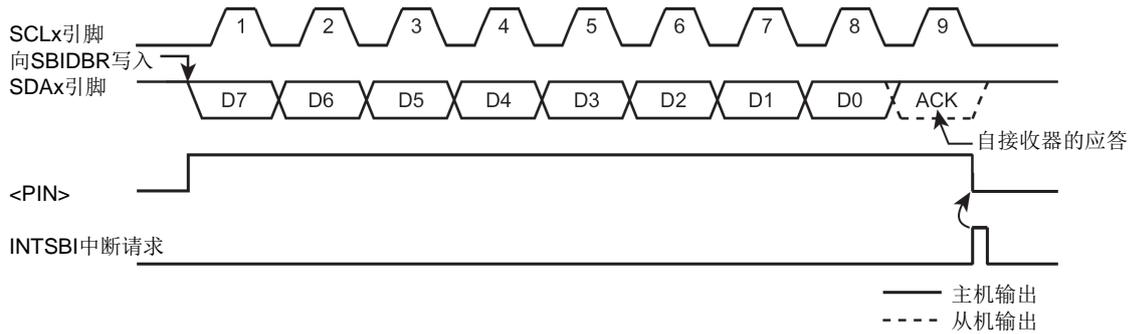


图 12-10 <BC[2:0]>= " 000 ", <ACK>= " 1 "(传输器模式)

(2) 接收器模式(<TRX> = " 0 ")

当下一个待传输数据拥有 8 位时，传输数据则被写入SBIDBR。
 若数据长度不同，则<BC [2:0]>和<ACK>被编程，接收的数据从SBIDBR中读取，以释放SCL线。(在从地址传输后立即读取的数据未定义。) 在读取数据后，<PIN>设置为 "1"，串行时钟被输出到SCL引脚，以传输下一数据字。在最后一位，当应答信号变成"低"电平时，"0" 被输出到SDA引脚。

此后，生成INTSBI中断请求，<PIN>被清 "0"，由此将SCL引脚拉到"低"电平。每次从SBIDBR读取接收数据时，可输出一字传输时钟及应答信号。

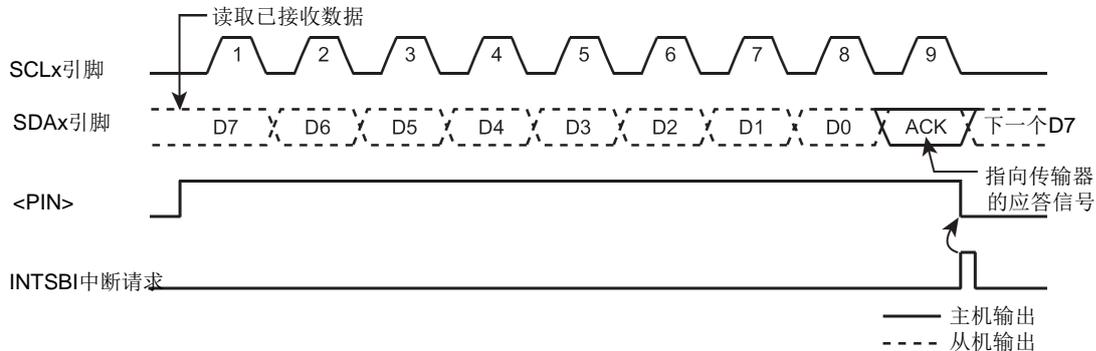


图 12-11 <BC[2:0]> = " 000 ", <ACK> = " 1 "(接收器模式)

终止传输器的数据传输时，在读取倒数第二个数据字前，<ACK>必须被立即清 "0"
 这样可禁止生成最后数据字的确认时钟。

在传输完成时，一个中断请求即被生成。在中断处理完成后，必须将<BC[2:0]>设置为 "001"，并必须读取该数据，以确保生成 1 位传输时钟。

此时，主接收器会将SDA总线保持在"高"电平，作为应答信号向传输器发送结束传输应答信号的信号。

在对终止 1-位数据的接收的中断处理时，生成的停止条件终止数据传输。

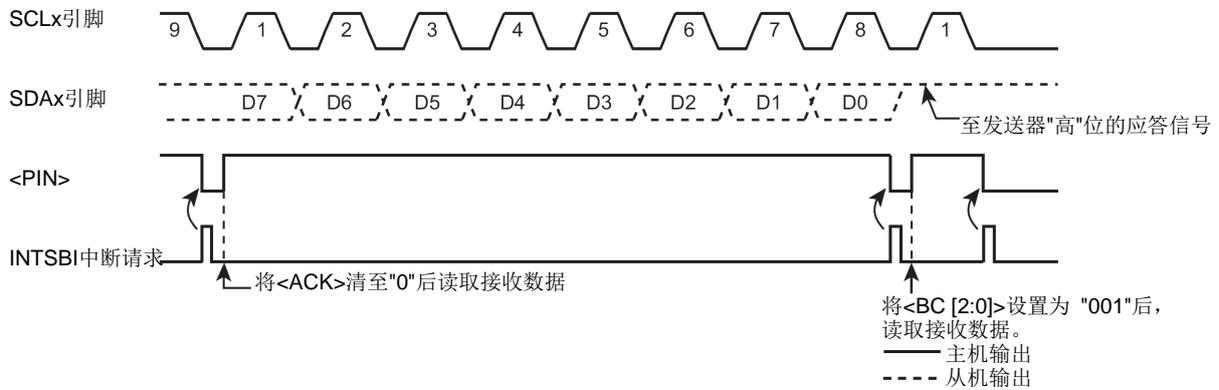


图 12-12 在主机接收器模式下终止数据传输

示例:在接收N数据字时

INTSBI中断(数据传输后)

	7	6	5	4	3	2	1	0	
SBICR1←	X	X	X	X	0	X	X	X	设置拟接收的数据位的数目，并规定是否需要ACK。
Reg. ←	SBIDBR								读取虚拟数据。
中断结束									

INTSBI中断(第1 至第(N-2)个数据的接收)

	7	6	5	4	3	2	1	0	
Reg.←	SBIDBR								读取第 1 个 ~ 第(N-2)个数据字。
中断结束									

INTSBI中断(第(N-1)个数据的接收)

	7	6	5	4	3	2	1	0	
SBICR1	←X	X	X	0	0	X	X	X	禁止应答时钟的生成。
Reg.	←SBIDBR								读取第(N-1)个数据字。
中断结束									

INTSBI中断(第N个数据的接收)

	7	6	5	4	3	2	1	0	
SBIxCR1	←0	0	1	0	0	X	X	X	禁止应答时钟的生成。
Reg.	←SBIDBR								读取第N个数据字。
中断结束									

INTSBI中断(完成数据接收后)

进行处理，以生成停止条件。 终止数据传输。

中断结束

注:X; 忽略

12.6.3.2 从机模式(<MST> = "0")

在从机模式下，SBI在四种情况下生成INTSBI中断请求：

- 1) 在SBI已从主机接收到任何从地址时。
- 2) 在SBI已接收到一个一般调用地址时。
- 3) 在所接收的从地址可匹配其地址时。
- 4) 在应一般调用的要求完成一次数据传输时。

此外，若在主模式SBI检测到仲裁丢失，则其会切换至从机模式。

若在数据字传输中检测到仲裁丢失，则在传输完成后，生成INTSBI中断请求，<PIN>被清 "0"，SCL引脚被拉到"低"电平。

当数据写入SBIDBR或从SBIDBR读取时，或当<PIN>设置为 "1" 时，SCLx引脚在tLOW时间后被释放。

然而，应根据第二字节的内容通过软件生成一应答信号来控制一般调用的第二个字节。

在从机模式下，执行正常从机模式处理或仲裁丢失的处理。

对SBISR<AL>，<TRX>，<AAS>和<ADO>进行测试以确定所需处理。"表 12-2 从机模式下处理"显示从机模式状态和所需的处理。

示例：在从机接收器模式下，所接收的从地址可匹配SBI的自身地址，且方向位为 "1" 时。

INTSBI中断

若TRX = 0

则转到其它处理。

若AL = 0

则转到其它处理。

若AAS = 0

则转到其它处理。

SBICR1	←	X	X	X	1	0	X	X	X	设置拟传输的位的数目。
SBIDBR	←	X	X	X	X	X	X	X	X	设置传输数据。

注:X; 忽略

表 12-2 在从机模式下的处理

<TRX>	<AL>	<AAS>	<ADO>	状态	处理
1	1	1	0	当从地址正在传输, SBI接收到另一主机传输的从地址及方向位 "1" 时, 检测到仲裁丢失。	将<BC[2:0]>设置为数据字位数, 并将传输数据写入SBIDBR。
	0	1	0	在从机接收器模式下, SBI接收到发自主机的一个从地址连同方向位 "1"。	
		0	0	0	在从机传输器模式下, SBI已完成了一个数据字的传输。
0	1	1	1/0	当从地址正在传输, SBI接收到另一主机传输的从地址及方向位 "0" 或一般调用地址时, 检测到仲裁丢失。	读取SBIDBR(假读), 以便将<PIN> 设置为1, 或将 "1" 写入<PIN>。
		0	0	当从地址或数据字正在传输, 传输终止时, 检测到仲裁丢失。	
	0	1	1/0	在从机接收器模式下, SBI接收主机传输的从地址及方向位 "0" 或一般调用地址。	
		0	1/0	在从机接收器模式下, SBI已完成一个数据字的接收。	为<BC[2:0]>设置数据字中的位数并从SBIDBR中读取接收的数据。

12.6.4 生成停止条件

当SBISR<BB>为"1"时，将"1"写入SBICR2<MST, TRX, PIN>，并将"0"写入<BB>会导致SBI开启一个可在总线上生成停止条件的序列。

总线上出现停止条件前，不得改变<MST, TRX, BB, PIN>的内容。

当另一设备限制了SCL总线时，SBI需等待至SCL线路被解除。

然后，SDA引脚走"高"，停止条件即生成。

```

      7  6  5  4  3  2  1  0
SBICR2 ← 1  1  0  1  1  0  0  0  生成停止条件。
  
```

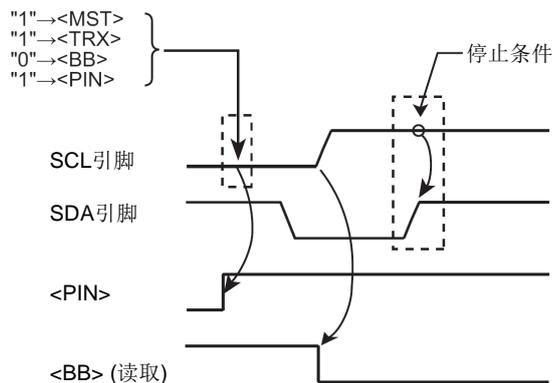


图 12-13 生成停止条件

12.6.5 重启程序

当主设备改变数据传输方向而不终止向从设备的传输时重新启动。在主机模式下的重新启动程序如下所述。

首先，将"0"写入SBICR2<MST, TRX, BB>，并将"1"写入<PIN>，以释放总线。此时，SDAx引脚保持在"高"位，SCLx引脚被释放。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SBISR<BB>并等待直到它变为"0"，以确保SCLx引脚被释放。

其次，测试<LRB>并等待至其变为"1"，以确保没有别的设备会将SCLx总线拖至"低"电平。

通过执行以上程序一旦确定总线空闲，则执行"12.6.2 生成启动条件和从地址"中的程序以生成启动条件。

为满足重启设置时间，在确定总线空闲后，必须通过软件创建至少 4.7 μs的等待期(在标准模式下)。

注1: 当其"0"时，不要将<MST>写至"0"。(重新启动无法启动)。

注2: 在主机作为接收器工作时，必须在重新启动生成前，完成从作为传输器工作的从机进行的数据传输。完成数据传输时，从设备必须接收"高"电平应答信号。因此，<LRB>在生成一重启前变为"1"。即使<LRB>=，也不会检测出SCL的上升缘。

通过以下重启程序后确定出"1"。在检查SCL线路的状态时，需读取端口。

```

      7 6 5 4 3 2 1 0
    ← SBICR2      ← 0 0 0 1 1 0 0 0 解除总线
    如SBISR<BB> ≠ 0      检查并确认SCL引脚已被解除。
    Then
    如SBISR<LRB> ≠ 1      检查并确认没有别的设备会将SCL引脚拖至"低"。
    Then
    4.7µs等待
    SBICR1      ← X X X 1 0 X X X 选择应答模式。
    SBIDBR      ← X X X X X X X X 设置所需的从地址和方向。
    SBICR2      ← 1 1 1 1 1 0 0 0 生成启动条件。

```

注: X; 忽略

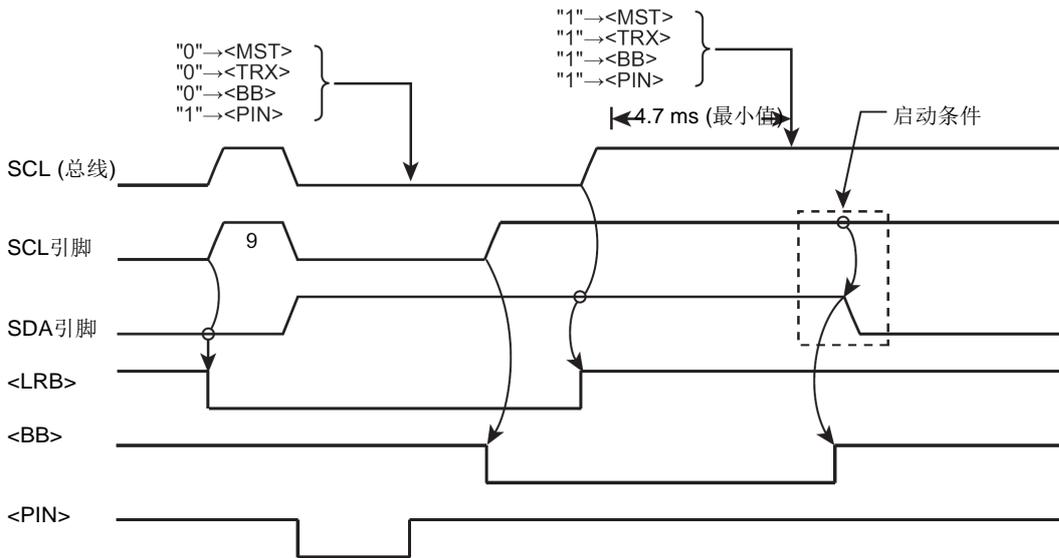


图 12-14 生成重启的时序图

12.7 多主机使用注意事项

在多主机模式时通信状态处于锁定状态的情况下，通过软件准备恢复过程。

恢复过程示例

1. 启动与启动通信同步的超时检测时序器。
2. 若在规时序间内未发生串行接口中断(INTSBIx)，则发生超时，并且MCU确定通信已被锁定。
3. 对串行总线接口进行软件复位，以释放通信锁定条件。
4. 调节传输时序。(注)
5. 重发传输数据。

注：调整MCU之间的传输时序，以免传输时序重叠。

12.8 SIO 模式控制寄存器

以下寄存器控制时钟同步 8 位SIO模式串行总线接口，具有状态信息，以便监视。

12.8.1 SBICR0(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	串行总线接口操作。 0: 禁用 1: 启用 使用串行总线接口前启用位。 若禁用此位，由于除了SBICR0之外的所有时钟均停止，因此会减少功耗。 若串行总线接口操作启用后再禁用，设置仍然保留于各寄存器。
6-0	-	R	读作0。

12.8.2 SBICR1(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	传输开始/停止 0: 停止 1: 开始																								
6	SIOINH	R/W	传输 0: 继续 1: 强制终止																								
5-4	SIOM[1:0]	R/W	选择传输模式 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	On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>2.5 MHz</td> </tr> <tr> <td>001</td> <td>n = 4</td> <td>1.25 MHz</td> </tr> <tr> <td>010</td> <td>n = 5</td> <td>625 kHz</td> </tr> <tr> <td>011</td> <td>n = 6</td> <td>313 kHz</td> </tr> <tr> <td>100</td> <td>n = 7</td> <td>156 kHz</td> </tr> <tr> <td>101</td> <td>n = 8</td> <td>78 kHz</td> </tr> <tr> <td>110</td> <td>n = 9</td> <td>39 kHz</td> </tr> <tr> <td>111</td> <td>-</td> <td>外部时钟</td> </tr> </table> <div style="margin-left: 100px;"> $\left. \begin{array}{l} \text{系统时钟: } f_{\text{sys}} \\ \text{时钟齿轮: } fc/1 \\ \text{频率} = \frac{f_{\text{sys}}/2}{2^n} \text{ [Hz]} \end{array} \right\} (= 40 \text{ MHz})$ </div>	000	n = 3	2.5 MHz	001	n = 4	1.25 MHz	010	n = 5	625 kHz	011	n = 6	313 kHz	100	n = 7	156 kHz	101	n = 8	78 kHz	110	n = 9	39 kHz	111	-	外部时钟
000	n = 3	2.5 MHz																									
001	n = 4	1.25 MHz																									
010	n = 5	625 kHz																									
011	n = 6	313 kHz																									
100	n = 7	156 kHz																									
101	n = 8	78 kHz																									
110	n = 9	39 kHz																									
111	-	外部时钟																									

注1: 复位后, <SCK[0]>位读作"1"。然而, 若在SBICR2寄存器处选择SIO模式, 则初始值读作"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。SBICR2寄存器和SBISR寄存器的说明相同。

注2: 传输模式和序列时钟程序编制前, 将<SIOS>设置为"0"且<SIOINH>至"1"。

12.8.3 SBIDBR(数据缓存寄存器)

	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	传送数据

注1: 传输数据必须从MSB(位 7)写入寄存器。接收数据存于LSB中。

注2: 由于SBIDBR拥有独立的写入和读取寄存器, 因此不能读取写入的数据。因此, 读-修改-写指令(如位处理)无法使用。

12.8.4 SBICR2(控制寄存器 2)

通过向此寄存器写入，可作为SBISR寄存器使用。

	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	-	-	-	-	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	选择串行总线接口操作模式 (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"列所写入数值为初始状态设置SIO模式后的数值。

注 2: 通信会话时不得改变模式。

12.8.5 SBISR(状态寄存器)

通过向此寄存器写入，可作为SBICR2使用。

	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	-	-	-	-	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 1)
3	SIOF	R	串行传送状态监控器。 0: 已完成。 1: 进行中
2	SEF	R	移位操作状态监视器 0: 已完成。 1: 进行中
1-0	-	R	读作1。(注 1)

注：本文中，“复位后”列所写入数值为初始状态设置SIO模式后的数值。

12.8.6 SBIBR0(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模式操作。 0: 停止 1: 操作
5-1	-	R	读作1。
0	-	R/W	务必写入 "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SBICR1<SCK[2:0]>进行编程可选择内部时钟或外部时钟。

(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。

传输开始时，SCKx引脚输出成为"高"电平。

若在写入传输数据或读取接收数据过程中，程序无法跟上串行时钟速率，则SBI自动输入等待时间。在时间段内，串行时钟自动停止，下一移位操作暂停，直至处理完成。

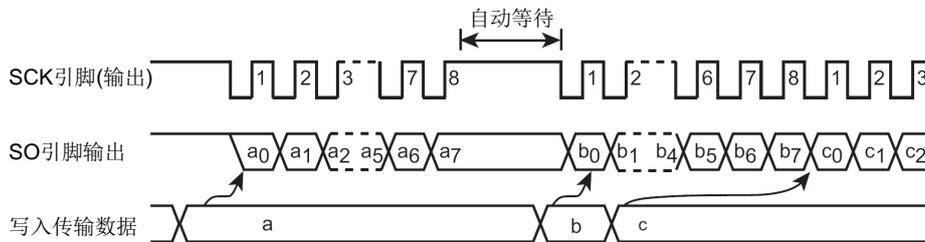


图 12-15 自动等待

(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。

正确移位操作，"高"和"低"电平的串行时钟须具备如下所示脉宽。

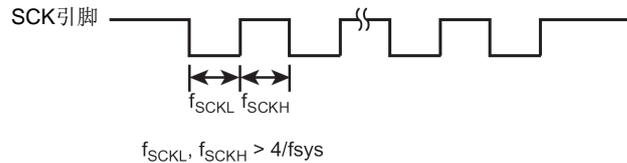


图 12-16 外部时钟输入最大传输频率

12.9.1.2 移位沿

前缘移位用于传送。后缘移位用于接收。

- 前边沿移位
数据于串行时钟前缘 (或SCKx引脚输入/输出下降缘) 移位。
- 后边沿移位
数据于串行时钟后缘 (或SCKx引脚输入/输出上升缘) 移位。

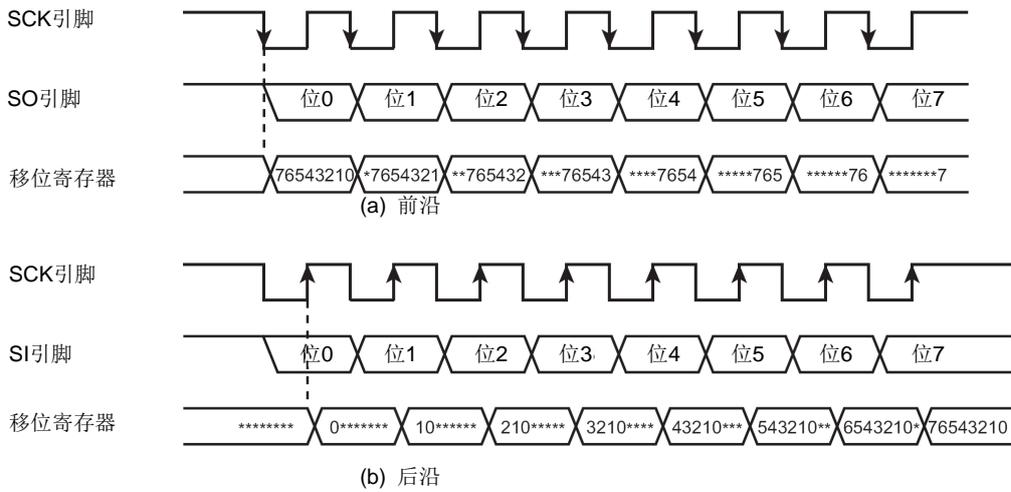


图 12-17 移位沿

12.9.2 传输模式

通过对SBICR1<SIOM[1:0]>进行编程，可以选择传输模式，接收模式或传输/接收模式。

12.9.2.1 8-位传输模式

将控制寄存器设置为传输模式并将传输数据写入SBIDBR。

在写入传输数据后，将"1"写入SBICR1<SIOS>，开始传输。在与串行时钟同步的情况下，传输数据从SBIDBR移到移位寄存器，并输出到SO引脚，最低有效位(LSB)在先。一旦传输数据被传输到移位寄存器，SBIDBR变为空，并生成INTSBI(缓存器空)中断，请求下一传输数据。

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。当SBIDBR加载下一传输数据时，将清除等待状态。

在外部时钟模式下，SBIDBR必须在下一数据移位操作开始前加载数据。因此，数据传输速率根据中断服务程序中中断请求生成时与SBIDBR加载数据时之间的最大等待时间而不同。

在传输开始时，与先前传输数据的最后一位相同的值被输出，输出时间从SBISR<SIOF>设置为"1"时起至SCK下降沿。

在INTSBI中断服务程序中，将<SIOS>清"0"或将<SIOINH>设置为"1"，可终止传输。若<SIOS>清除，传输结束前输出剩余数据。程序对SBISR<SIOF>进行检查，以确定传输是否结束。传输结束时，<SIOF>清至"0"。若<SIOINH>设置为"1"，立即放弃传输，且<SIOF>清"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		
SBICR1	←	0	1	0	0	0	X	X	X	选择传输模式。
SBIDBR	←	X	X	X	X	X	X	X	X	写入传输数据。
SBICR1	←	1	0	0	0	0	X	X	X	开始传输。

INTSBI中断

SBIDBR	←	X	X	X	X	X	X	X	X	写入传输数据。
--------	---	---	---	---	---	---	---	---	---	---------

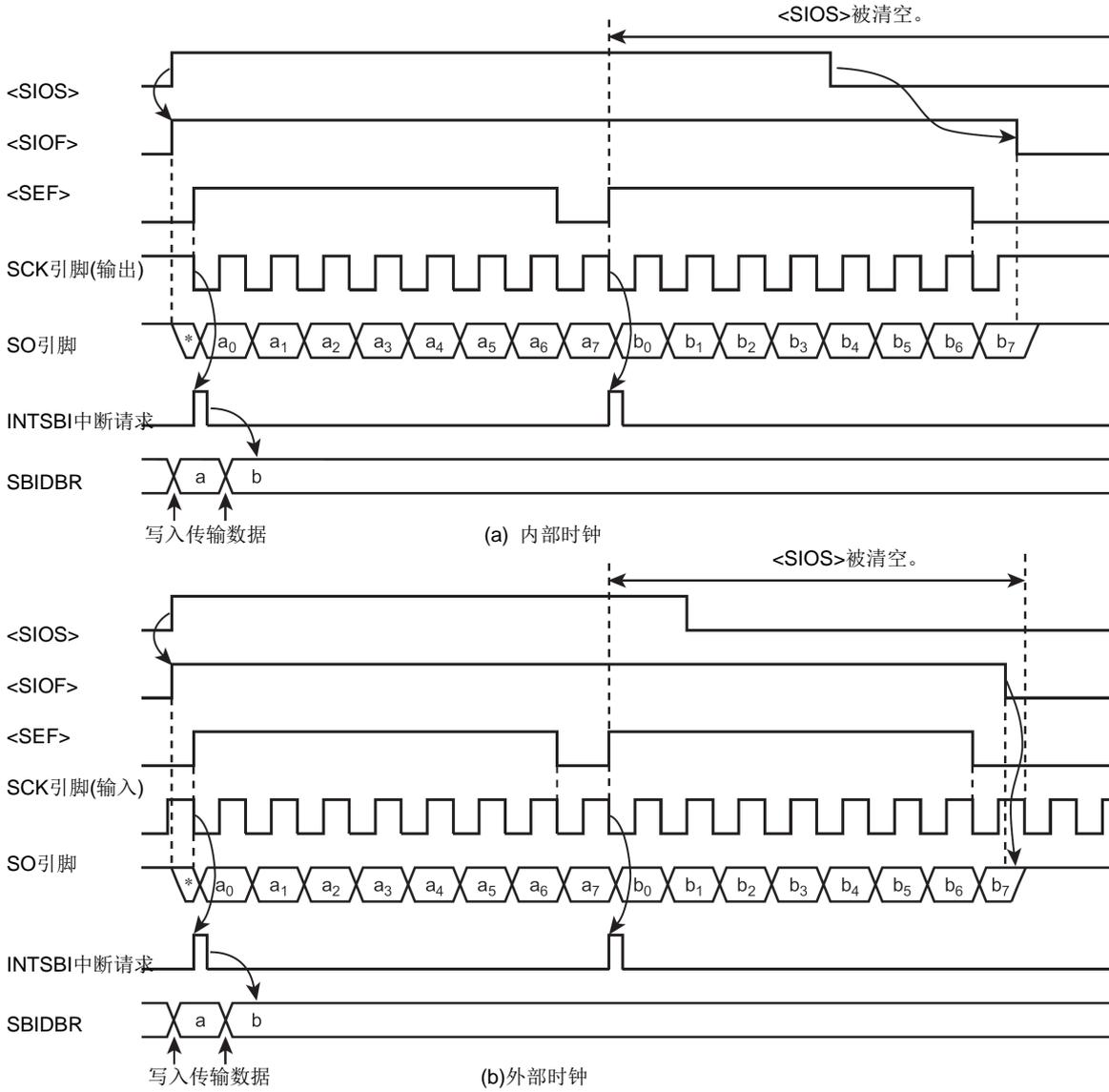


图 12-18 传输模式

示例：编程(外部时钟)，通过<SIO>终止传输

- | | | |
|----------------------------|-----------------|-------------------------|
| | 7 6 5 4 3 2 1 0 | |
| → 如SBISR< SIOF> ≠ 0 | | 确认传输完成。 |
| → 然后 | | |
| → 若SCK ≠ 1 | | 通过监视端口，确认 "1" 设置为SCK引脚。 |
| → 然后 | | |
| → SBICR1 ← 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

将控制寄存器设置为接收模式。然后将 "1" 写入SBICR1<SIOS>, 可进行接收。在与串行时钟同步的情况下, 数据从SI引脚纳入移位寄存器, 最低有效位(LSB)在先。一旦移位寄存器加载 8-位数据, 它可将接收数据传输给SBIDBR, 并生成INTSBI(缓存器满)中断请求, 以请求读取接收数据。然后中断服务程序将从SBIDBR读取接收到的数据。

在内部时钟模式下, 串行时钟将会停止并自动进入等待状态直到从SBIDBR读取接收到的数据。外部时钟模式中, 和外部时钟同步执行移位操作。视乎生成中断请求时和读取接收数据之间的最大延迟, 最大数据传输速率各不相同。

在INTSBI中断服务程序中, 将<SIOS>清 "0"或将<SIOINH>设置为 "1", 可终止接收。若<SIOS>被清除, 则接收继续, 直到所有接收数据位写入SBIDBR。程序对SBISR<SIOF>进行检查, 以确定接收是否结束。在接收结束时, <SIOF>被清 "0"确认接收完成后, 最后接收数据被读取。若<SIOINH>设置为 "1", 接收立即终止, <SIOF>被清至 "0"。(接收数据变为无效, 无需读出。)

注: 在传输模式改变后, SBIDBR的内容不保留。须将<SIOS>清至 "0" 才能完成进行中的接收, 最后接收数据须在传输模式改变前读取。

```

          7  6  5  4  3  2  1  0
SBICR1 ← 0  1  1  1  0  X  X  X  选择接收模式。

```

```

SBICR1 ← 1  0  1  1  0  X  X  X  开始接收。

```

INTSBI中断

```

Reg.    ←  SBIDBR                读取接收数据。

```

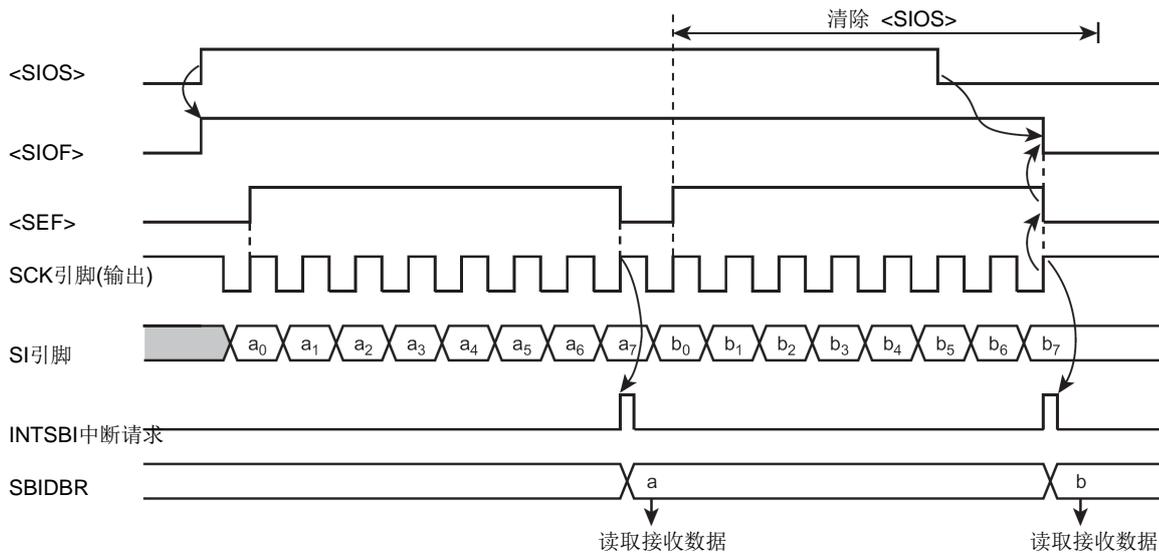


图 12-19 接收模式(示例:内部时钟)

12.9.2.3 8-位传输/接收模式

将控制寄存器设置为传输/接收模式。然后将传输数据写入SBIDBR，并将SBICR1<SIOS>设置为"1"，可进行传输和接收。传输数据在串行时钟下降沿经SOx引脚输出，而接收数据在串行时钟上升沿经SI引脚纳入，最低有效位(LSB)在先。一旦移位寄存器加载 8-位数据，它可将接收数据传输给SBIDBR，并生成INTSBI中断请求。中断服务程序读取数据缓存寄存器的接收数据，并写入下一传输数据。因为传输操作和接收操作共享SBIDBR，所以在写入下一传输数据前，必须读取接收数据。

在内部时钟操作中，串行时钟自动处于等待状态，直到读取接收数据及写入下一传输数据。

在外部时钟模式下，移位操作与外部串行时钟同步进行。因此，在下一移位操作开始前，必须读取接收数据并写入下一传输数据。根据中断请求生成时间与传输数据写入时之间的最大等待时间，外部时钟操作的最大数据传输速率不同。

在传输开始时，与先前传输的数据的最后一位相同的值被输出，输出时间从<SIOF>设置为"1"时起到SCK下降沿。

在INTSBI中断服务程序中，将<SIOS>清"0"或将SBICR1<SIOINH>设置为"1"，可终止传输和接收。若<SIOS>被清除，则传输和接收继续，直到接收数据被完全传输到SBIDBR。程序对SBISR<SIOF>进行检查，以确定传输和接收是否已结束。在传输和接收结束时，<SIOF>被清"0"。若<SIOINH>设置为"1"，则传输和接收立即中止，<SIOF>被清"0"。

注：在传输模式改变后，SBIDBR的内容不保留。须将<SIOS>清"0"才能完成进行中的传输和接收，最后接收数据须在传输模式改变前读取。

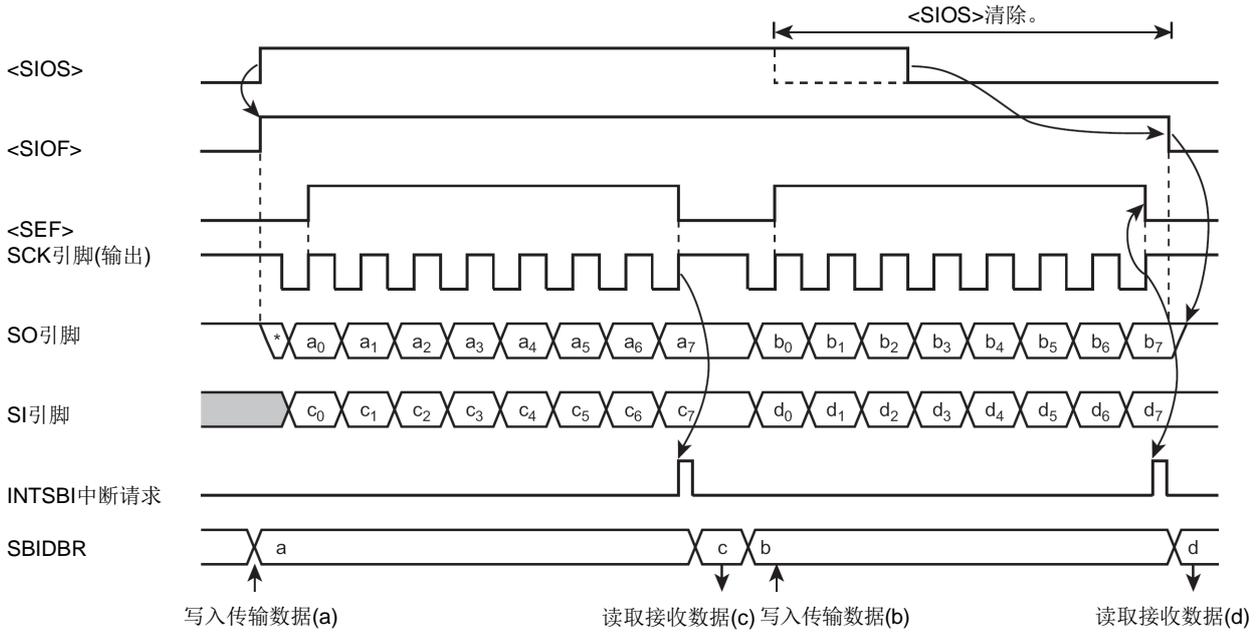


图 12-20 传输/接收模式(示例:(内部时钟))

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	1	0	0	X	X	X	选择传输模式。
SBIDBR	←	X	X	X	X	X	X	X	X	写入传输数据。
SBICR1	←	1	0	1	0	0	X	X	X	开始接收/传输。
INTSBI中断										
Reg.	←	SBIDBR								读取接收数据。
SBIDBR	←	X	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SBICR1<SIOS>= "0" 的条件下，传输数据最后一位保留SCK上升沿的数据，如下所示。传输模式和传输/接收模式相同。

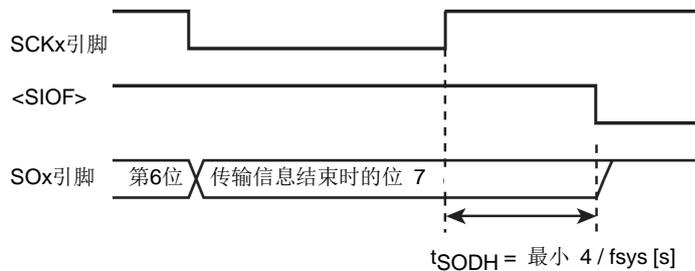


图12-21 在传输结束时最后一位的数据保留时间



13. 12 位模拟数字转换器

TMPM375FSDMG包含一个12 位的逐次逼近法模拟数字转换器(ADC)。

ADC单元B (ADC B)有5个模拟输入。其中四个输入可用于马达 0 的分流电阻电流，另一个输入与运算放大器的输出连接。而四个输入可用于外部输入。

四个外部模拟输入引脚(AINB9~AINB12)也可用作输入/输出端口。

13.1 功能和特点

1. 可在收到来自PMD或TMRB(中断)的触发脉冲信号时，选择模拟输入并开始AD转换。
2. 可在软件触发脉冲程序和常量触发脉冲程序中选择模拟输入。
3. ADC有十二个AD转换结果寄存器。
4. 在PMD触发脉冲和TMRB触发脉冲所启动的程序结束时，ADC生成中断信号。
5. 软件触发程序和常量触发程序结束时，ADC生成中断信号。
6. ADC具有AD转换监控功能。当此功能处于已启用状态时，当转换结果与规定的比较值相匹配时，生成一次中断。

13.2 方块图

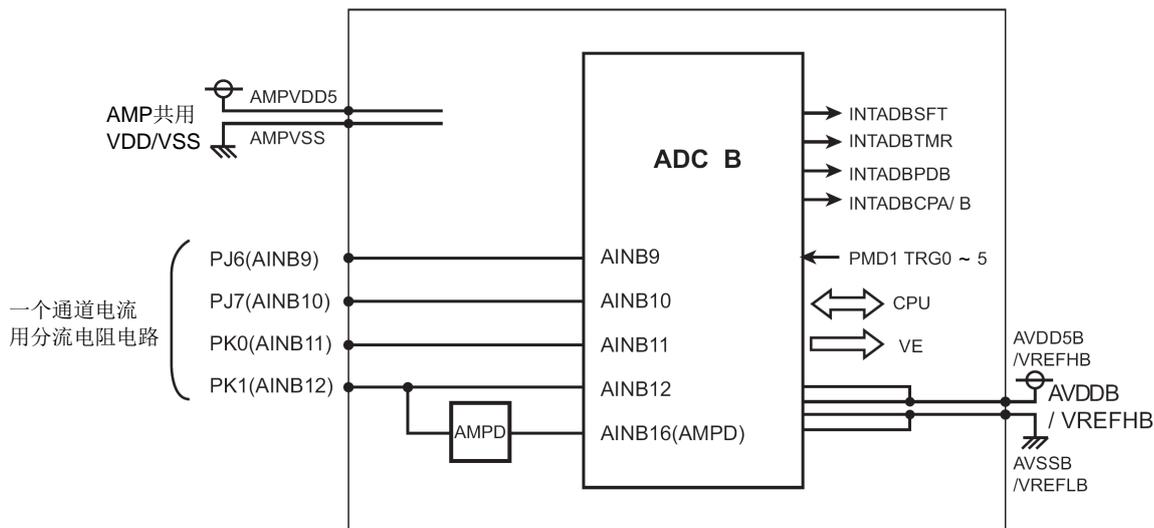


图13-1 AD转换器方块图

13.3 寄存器列表

单元	基址
单元B	0x4003_0200

寄存器名称(x=B)		地址(基+)
时钟设置寄存器	ADxCLK	0x0000
模式设置寄存器 0	ADxMOD0	0x0004
模式设置寄存器1	ADxMOD1	0x0008
模式设置寄存器 2	ADxMOD2	0x000C
监控设置寄存器 0	ADxCMPCR0	0x0010
监控设置寄存器1	ADxCMPCR1	0x0014
转换结果比较寄存器 0	ADxCMP0	0x0018
转换结果比较寄存器1	ADxCMP1	0x001C
转换结果寄存器 0	ADxREG0	0x0020
转换结果寄存器1	ADxREG1	0x0024
转换结果寄存器 2	ADxREG2	0x0028
转换结果寄存器 3	ADxREG3	0x002C
转换结果寄存器 4	ADxREG4	0x0030
转换结果寄存器 5	ADxREG5	0x0034
转换结果寄存器 6	ADxREG6	0x0038
转换结果寄存器 7	ADxREG7	0x003C
转换结果寄存器 8	ADxREG8	0x0040
转换结果寄存器 9	ADxREG9	0x0044
转换结果寄存器10	ADxREG10	0x0048
转换结果寄存器11	ADxREG11	0x004C
-	保留	0x0050
-	保留	0x0054
-	保留	0x0058
-	保留	0x005C
-	保留	0x0060
-	保留	0x0064
PMD触发程序号选择寄存器 6	ADxPSEL6	0x0068
PMD触发程序号选择寄存器 7	ADxPSEL7	0x006C
PMD触发程序号选择寄存器 8	ADxPSEL8	0x0070
PMD触发程序号选择寄存器 9	ADxPSEL9	0x0074
PMD触发程序号选择寄存器10	ADxPSEL10	0x0078
PMD触发程序号选择寄存器11	ADxPSEL11	0x007C
PMD触发中断选择寄存器 0	ADxPINTS0	0x0080
PMD触发中断选择寄存器1	ADxPINTS1	0x0084
PMD触发中断选择寄存器 2	ADxPINTS2	0x0088
PMD触发中断选择寄存器 3	ADxPINTS3	0x008C
PMD触发中断选择寄存器 4	ADxPINTS4	0x0090
PMD触发中断选择寄存器 5	ADxPINTS5	0x0094
PMD触发程序寄存器 0	ADxPSET0	0x0098
PMD触发程序寄存器1	ADxPSET1	0x009C
PMD触发脉序寄存器 2	ADxPSET2	0x00A0
PMD触发程序寄存器 3	ADxPSET3	0x00A4
PMD触发程序寄存器 4	ADxPSET4	0x00A8
PMD触发程序寄存器 5	ADxPSET5	0x00AC
定时器触发程序寄存器 0~3	ADxTSET03	0x00B0

寄存器名称(x=B)		地址(基+)
定时器触发程序寄存器 4 ~ 7	ADxTSET47	0x00B4
定时器触发程序寄存器 8~11	ADxTSET811	0x00B8
软件触发程序寄存器 0~3	ADxSSET03	0x00BC
软件触发程序寄存器 4 ~ 7	ADxSSET47	0x00C0
软件触发程序寄存器 8~11	ADxSSET811	0x00C4
常量转换程序寄存器 0~3	ADxASET03	0x00C8
常量转换程序寄存器 4~7	ADxASET47	0x00CC
常量转换程序寄存器 8~11	ADxASET811	0x00D0
模式设置寄存器3	ADxMOD3	0x00D4

注：禁止进入"保留"区域。

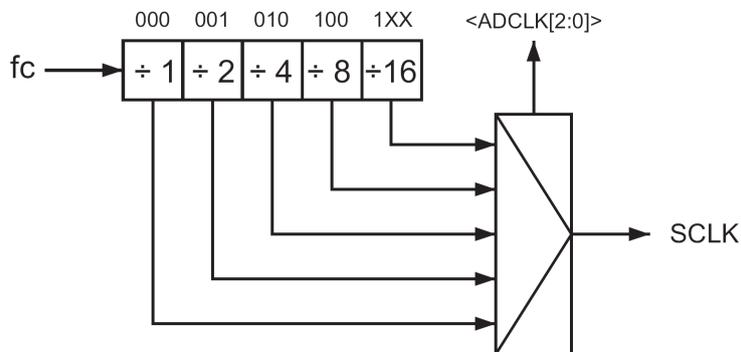
13.4 寄存器描述

按照在ADC时钟设置寄存器中选择的时钟频率执行AD转换。

13.4.1 ADxCLK(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	-	TSH				ADCLK		
复位后	0	1	0	0	1	0	0	0

位	比特符号	类型	功能
31-7	-	R	读作"0"。
6-3	TSH[3:0]	R/W	写作"1001"。
2-0	ADCLK[2:0]	R/W	AD预定标器输出(SCLK)选择 000: f_c (注1) 001: $f_c/2$ 010: $f_c/4$ 011: $f_c/8$ 1xx: $f_c/16$



注1: SCLK的频率可达40 MHz。

注2: 以寄存器所选时钟频率进行AD转换。为确保精度, 必须选择转换时钟频率。

注3: 当AD转换正在进行时, 不得改变转换时钟。

13.4.2 ADxMOD0 (模式设置寄存器 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	-	-	-	-	-	-	DACON	ADSS
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	DACON	R/W	DAC控制 0: OFF 1: ON 使用ADC时, 将<DACON>设置为"1"。
0	ADSS	W	软件触发转换 0: 忽略 1: 启动 将<ADSS>设置为 "1", 就能启动AD转换(软件触发转换)。接收PMD或MRB的触发信号(中断)启动AD转换。 详情见PMD和TMRB的章节。

13.4.3 ADxMOD1 (模式设置寄存器 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	ADEN	-	-	-	-	-	-	ADAS
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	ADEN	R/W	AD转换控制 0: 禁用 1: 启用 在使用ADC时, 将<ADEN>设置为"1"。将<ADEN>设为"1"后, 设置<ADAS>启动AD转换和重复转换。
6-1	-	R	读作"0"。
0	ADAS	R/W	常量AD转换控制 0: 禁用 1: 启用

13.4.4 ADxMOD2 (模式设置寄存器 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	-	-	-	-	-	-	ADSFN	ADBFN
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	ADSFN	R	软件转换占空标志 0: 转换完成 1: 转换中 <ADSFN>为软件AD转换占空标志。将<ADSS>设置为 "1" 后, AD转换实际开始时, 将<ADSFN>设置为 "1"。完成AD转换时, <ADSFN>清0。
0	ADBFN	R	AD转换占空标志 0: 转换未进行 1: 转换中 <ADBFN>为一AD转换占空标志。当AD转换启动时, 无论采用何种转换因数(PMD, 定时器, 软件, 常量), <ADBFN>均被设置为 "1"。完成AD转换时, <ADBFN>清0。

13.4.5 ADxMOD3 (模式设置寄存器 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	-	-	-	-	-	-	-	RCUT
复位后	0	0	0	0	0	1	0	1
	7	6	5	4	3	2	1	0
比特符号	-	-	PMODE			-	-	-
复位后	0	1	1	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-11	-	R/W	写作"0"。
10	-	R/W	写作"1"。
9	-	R/W	写作"0"。
8	RCUT	R/W	ADC操作控制 2 0: 启用 1: 禁用 ADC运行时, 先写至"0"。ADC停止运行时, 通过设置值"1", 可降低功耗。
7	-	R/W	写作"0"。
6	-	R/W	写作"1"。
5-3	PMODE[2:0]	R/W	写作"100"。
2-0	-	R/W	写作"0"。

注: 必须将ADxMOD3<PMODE[2: 0]>设置为 "100"。且不得改变ADxMOD3寄存器中的其它位。

13.4.6 ADxCMPCR0(监控设置寄存器 0)

转换结果固定后，中断信号(INTADxCPn)生成。

(x=B, n=A; A: 监视器 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	-	-	-	-	CMPCNT0			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMP0EN	-	-	ADBIG0	REGS0			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能												
31-12	-	R	读作"0"。												
11-8	CMPCNT0[3:0]	R/W	判定结果进行比较计数 0: 每次比较后 1: 两次比较后 15: 16 次比较后 ADxCMPCR0和ADxCMPCR1寄存器用于启用或禁用对AD转换结果和规定的比较值的比较，并选择与AD转换结果进行比较的寄存器，以及设置为判定结果应执行的比较次数。												
7	CMP0EN	R/W	监视功能 0: 禁用 1: 启用												
6-5	-	R	读作"0"。												
4	ADBIG0	R/W	比较条件 0: 大于或等于比较寄存器 1: 小于或等于比较寄存器												
3-0	REGS0[3:0]	R/W	用于比较的AD转换结果寄存器 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>0000: ADxREG0</td> <td>0100: ADxREG4</td> <td>1000: ADxREG8</td> </tr> <tr> <td>0001: ADxREG1</td> <td>0101: ADxREG5</td> <td>1001: ADxREG9</td> </tr> <tr> <td>0010: ADxREG2</td> <td>0110: ADxREG6</td> <td>1010: ADxREG10</td> </tr> <tr> <td>0011: ADxREG3</td> <td>0111: ADxREG7</td> <td>1011: ADxREG11</td> </tr> </table>	0000: ADxREG0	0100: ADxREG4	1000: ADxREG8	0001: ADxREG1	0101: ADxREG5	1001: ADxREG9	0010: ADxREG2	0110: ADxREG6	1010: ADxREG10	0011: ADxREG3	0111: ADxREG7	1011: ADxREG11
0000: ADxREG0	0100: ADxREG4	1000: ADxREG8													
0001: ADxREG1	0101: ADxREG5	1001: ADxREG9													
0010: ADxREG2	0110: ADxREG6	1010: ADxREG10													
0011: ADxREG3	0111: ADxREG7	1011: ADxREG11													

13.4.7 ADxCMPCR1(监视设置寄存器 1)

转换结果固定后，中断信号(INTADxCPn)生成。

(x=B, n=B ; B: 监视器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	-	-	-	-	CMPCNT1			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMP1EN	-	-	ADBIG1	REGS1			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能												
31-12	-	R	读作"0"。												
11-8	CMPCNT1[3:0]	R/W	判定结果进行比较计数 0: 每次比较后 1: 两次比较后 15: 16 次比较后 ADxCMPCR0和ADxCMPCR1寄存器用于启用或禁用对AD转换结果和规定的比较值的比较，并选择与AD转换结果进行比较的寄存器，以及设置为判定结果应执行的比较次数。												
7	CMP1EN	R/W	监视功能 0: 禁用 1: 启用												
6-5	-	R	读作"0"。												
4	ADBIG1	R/W	比较条件 0: 大于或等于比较寄存器 1: 小于或等于比较寄存器												
3-0	REGS1[3:0]	R/W	用于比较的AD转换结果寄存器 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>0000: ADxREG0</td> <td>0100: ADxREG4</td> <td>1000: ADxREG8</td> </tr> <tr> <td>0001: ADxREG1</td> <td>0101: ADxREG5</td> <td>1001: ADxREG9</td> </tr> <tr> <td>0010: ADxREG2</td> <td>0110: ADxREG6</td> <td>1010: ADxREG10</td> </tr> <tr> <td>0011: ADxREG3</td> <td>0111: ADxREG7</td> <td>1011: ADxREG11</td> </tr> </table>	0000: ADxREG0	0100: ADxREG4	1000: ADxREG8	0001: ADxREG1	0101: ADxREG5	1001: ADxREG9	0010: ADxREG2	0110: ADxREG6	1010: ADxREG10	0011: ADxREG3	0111: ADxREG7	1011: ADxREG11
0000: ADxREG0	0100: ADxREG4	1000: ADxREG8													
0001: ADxREG1	0101: ADxREG5	1001: ADxREG9													
0010: ADxREG2	0110: ADxREG6	1010: ADxREG10													
0011: ADxREG3	0111: ADxREG7	1011: ADxREG11													

13.4.8 ADxCMP0(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	AD0CMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	AD0CMP0				-	-	-	-
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"。
15-4	AD0CMP0[11:0]	R/W	与AD转换结果进行比较的值。 指定与AD转换结果进行比较的值。
3-0	-	R	读作"0"。

13.4.9 ADxCMP1(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	AD0CMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	AD0CMP1				-	-	-	-
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"。
15-4	AD0CMP1[11:0]	R/W	与AD转换结果进行比较的值。 指定与AD转换结果进行比较的值。
3-0	-	R	读作"0"。

13.4.10 ADxREG0(转换结果寄存器 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	ADR00							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR00				-	-	OVR0	ADR0RF
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"。
15-4	ADR00[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR0	R	溢位标志 0: 无溢位 1: 溢位 读取ADxREG0的值前,当存储新的AD转换结果时设置此标志,而在读取ADxREG0的低位字节时清空此标志。
0	ADR0RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR0RF>为一个标志,当向ADxREG0寄存器存储一个AD转换结果时设置此标志,而当读取ADxREG0 的低位字节时清空此标志。

13.4.11 ADxREG1(转换结果寄存器 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	ADR10							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR10				-	-	OVR1	ADR1RF
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"。
15-4	ADR10[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR1	R	溢位标志 0: 无溢位发生 1: 溢位发生 读取ADxREG1的值前,当存储新的AD转换结果时设置此标志,而在读取ADxREG1的低位字节时清空此标志。
0	ADR1RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR1RF>为一个标志,当向ADxREG1寄存器存储一个AD转换结果时设置此标志,而当读取ADxREG1 的低位字节时清空此标志。

13.4.12 ADxREG2(转换结果寄存器 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	ADR20							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR20				-	-	OVR2	ADR2RF
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"。
15-4	ADR20[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR2	R	溢位标志 0: 无溢位发生 1: 溢位发生 读取ADxREG2的值前,当存储新的AD转换结果时设置此标志,而在读取ADxREG2的低位字节时清空此标志。
0	ADR2RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR2RF>为一个标志,当向ADxREG2寄存器存储一个AD转换结果时设置此标志,而当读取ADxREG2 的低位字节时清空此标志。

13.4.13 ADxREG3(转换结果寄存器 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	ADR30							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR30				-	-	OVR3	ADR3RF
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"。
15-4	ADR30[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR3	R	溢位标志 0: 无溢位发生 1: 溢位发生 读取ADxREG3的值前, 当存储新的AD转换结果时设置此标志, 而在读取ADxREG3的低位字节时清空此标志。
0	ADR3RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR3RF>是一个标志, 当向ADxREG3寄存器存储一个AD转换结果时设置此标志, 而当读取ADxREG3的低位字节时清空此标志。

13.4.14 ADxREG4(转换结果寄存器 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	ADR40							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR40				-	-	OVR4	ADR4RF
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"。
15-4	ADR40[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR4	R	溢位标志 0: 无溢位发生 1: 溢位发生 读取ADxREG4的值前, 当存储新的AD转换结果时设置此标志, 而在读取ADxREG4的低位字节时清空此标志。
0	ADR4RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR4RF>是一个标志, 当向ADxREG4寄存器存储一个AD转换结果时设置此标志, 而当读取ADxREG4的低位字节时清空此标志。

13.4.15 ADxREG5(转换结果寄存器 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	ADR50							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR50				-	-	OVR5	ADR5RF
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"。
15-4	ADR50[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR5	R	溢位标志 0: 无溢位发生 1: 溢位发生 读取ADxREG5的值前, 当存储新的AD转换结果时设置此标志, 而在读取ADxREG5的低位字节时清空此标志。
0	ADR5RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR5RF>是一个标志, 当向ADxREG5寄存器存储一个AD转换结果时设置此标志, 而当读取ADxREG5的低位字节时清空此标志。

13.4.16 ADxREG6(转换结果寄存器 6)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR60							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR60				-	-	OVR6	ADR6RF
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"。
15-4	ADR60[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR6	R	溢位标志 0: 无溢位发生 1: 溢位发生 读取ADxREG6的值前, 当存储新的AD转换结果时设置此标志, 而在读取ADxREG6的低位字节时清空此标志。
0	ADR6RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR6RF>是一个标志, 当向ADxREG6寄存器存储一个AD转换结果时设置此标志, 而当读取ADxREG6的低位字节时清空此标志。

13.4.17 ADxREG7(转换结果寄存器 7)

	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	ADR70							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR70				-	-	OVR7	ADR7RF
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"。
15-4	ADR70[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR7	R	溢位标志 0: 无溢位发生 1: 溢位发生 该标志在当新的AD转换结果于ADxREG7的值储存之前储存时设置,在ADxREG7的低排序字节读取时清除。
0	ADR7RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR7RF>为一种标志,在AD转换结果储存在ADxREG7寄存器中时设置,在ADxREG7的低排序字节读取时清除。

13.4.18 ADxREG8(转换结果寄存器 8)

	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	ADR80							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR80				-	-	OVR8	ADR8RF
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"。
15-4	ADR80[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR8	R	溢位标志 0: 无溢位发生 1: 溢位发生 该标志在新的AD转换结果于ADxREG8的值储存之前储存时设置, 在ADxREG8的低排序字节读取时清除。
0	ADR8RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR8RF>为一种标志, 在AD转换结果储存于ADxREG8寄存器中时设置, 在ADxREG8的低排序字节读取时清除。

13.4.19 ADxREG9(转换结果寄存器 9)

	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	ADR90							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR90				-	-	OVR9	ADR9RF
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"。
15-4	ADR90[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR9	R	溢位标志 0: 无溢位发生 1: 溢位发生 该标志在新的AD转换结果于ADxREG9的值储存之前储存时设置, 在ADxREG9的低排序字节读取时清除。
0	ADR9RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR9RF>为一种标志, 在AD转换结果储存于ADxREG9寄存器中时设置, 在ADxREG9的低排序字节读取时清除。

13.4.20 ADxREG10(转换结果寄存器 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	ADR100							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR100				-	-	OVR10	ADR10RF
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"。
15-4	ADR100[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR10	R	溢位标志 0: 无溢位发生 1: 溢位发生 该标志在新的AD转换结果于ADxREG10的值储存之前储存时设置，在ADxREG10的低排序字节读取时清除。
0	ADR10RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR10RF>为一种标志，在AD转换结果储存于ADxREG10寄存器中时设置，在ADxREG10的低排序字节读取时清除。

13.4.21 ADxREG11(转换结果寄存器 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	ADR110							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR110				-	-	OVR11	ADR11RF
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"。
15-4	ADR110[11:0]	R	AD转换结果值
3-2	-	R	读作"0"。
1	OVR11	R	溢位标志 0: 无溢位发生 1: 溢位发生 该标志在新的AD转换结果于ADxREG11的值储存之前储存时设置,在ADxREG11的低排序字节读取时清除。
0	ADR11RF	R	AD转换结果存储标志 0: 无结果储存 1: 结果被存储 <ADR11RF>为一种标志,在AD转换结果储存于ADxREG11寄存器中时设置,在ADxREG11的低排序字节读取时清除。

13.4.22 PMD 触发程序寄存器

AD转换可由来自PMD(可编程马达驱动器)的触发脉冲启动。

PMD触发程序寄存器用于指定拟由该PMD所生成的六个触发各自启动的程序，以选择在程序完成时生成的中断，并选择拟使用的AIN输入端。

PMD触发程序寄存器包括三个类型的寄存器。(x=B : ADC单元)

PMD 触发程序号选择寄存器(ADxPSEL6~ADxPSEL11)

PMD程序号选择寄存器(ADxPSELn)，可规定拟由PMD所生成六个触发(PMD1TRG0~5)所对应的六个AD转换启动信号各自启动的程序。程序0~5可用。

"ADxPSEL6~ADxPSEL11"对应"PMD1TRG0~5"。

PMD触发中断信号选择寄存器(ADxPINTS0~ADxPINTS5)

PMD触发中断信号选择寄存器(ADxPINTS0~ADxPINTS5)选择每个程序完成时要生成的中断信号，并启用或禁用中断信号。

ADxPINTS0对应程序 0，它继续存在到ADxPINT5 (程序5)。

PMD触发程序寄存器(ADxPSET0~ADxPSET5)

PMD触发程序设置寄存器(ADxPSET0 ~ ADxPSET5)旨在规定程序0~5中每一个程序的设置。每一个PMD触发程序寄存器都由四个规定待转换的AIN输入。与ADx PSETn0~ADxPSETn3寄存器对应的转换结果，被储存于转换结果寄存器0~3 (ADxREG0~ADxREG3)中。

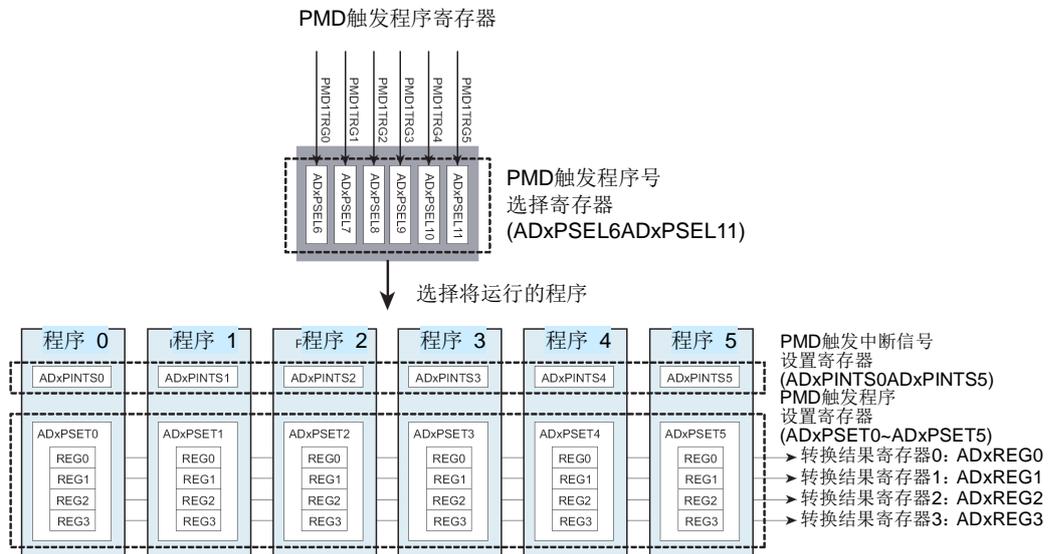


图13-2 PMD触发程序寄存器

13.4.22.1 ADxPSEL6~ADxPSEL11(PMD 触发程序编号选择寄存器 6~11)

ADxPSEL6: PMD触发程序号选择寄存器 6

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS6	-	-	-	-	PMDS6		
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	PENS6	R/W	PMD1TRG0触发控制 0: 禁用 1: 启用
6-3	-	R	读作"0"。
2-0	PMDS6[2:0]	R/W	程序号选择(见表13-1)

ADxPSEL7: PMD触发程序号选择寄存器 7

	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	PENS7	-	-	-	-	PMDS7		
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	PENS7	R/W	PMD1TRG1触发控制 0: 禁用 1: 启用
6-3	-	R	读作"0"。
2-0	PMDS7[2:0]	R/W	程序号选择(见表13-1)

ADxPSEL8: PMD触发程序号选择寄存器 8

	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	PENS8	-	-	-	-	PMDS8		
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	PENS8	R/W	PMD1TRG2触发控制 0: 禁用 1: 启用
6-3	-	R	读作"0"。
2-0	PMDS8[2:0]	R/W	程序号选择(见表13-1)

ADxPSEL9: PMD触发程序号选择寄存器 9

	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	PENS9	-	-	-	-	PMDS9		
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	PENS9	R/W	PMD1TRG3触发控制 0: 禁用 1: 启用
6-3	-	R	读作"0"。
2-0	PMDS9[2:0]	R/W	程序号选择(见表13-1)

ADxPSEL10: PMD触发程序号选择寄存器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	PENS10	-	-	-	-	PMDS10		
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	PENS10	R/W	PMD1TRG4触发控制 0: 禁用 1: 启用
6-3	-	R	读作"0"。
2-0	PMDS10[2:0]	R/W	程序号选择(见表13-1)

ADxPSEL11: PMD触发程序号选择寄存器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	PENS11	-	-	-	-	PMDS11		
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	PENS11	R/W	PMD1TRG5触发控制 0: 禁用 1: 启用
6-3	-	R	读作"0"。
2-0	PMDS11[2:0]	R/W	程序号选择(见表13-1)

表13-1 程序号选择

<PMDS6[2:0]>~ <PMDS11[2:0]>	
000	程序 0
001	程序 1
010	程序 2
011	程序 3
100	程序 4
101	程序 5
110	保留
111	保留

13.4.22.2 ADxPINTS0 ~5 (PMD 触发中断选择寄存器 0~5)

ADxPINTS0: PMD触发中断信号选择寄存器 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	-	-	-	-	-	-	INTSELO	
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"。
1-0	INTSELO[1:0]	R/W	中断信号选择 00: 无中断信号输出 01: 保留 10: INTADxPDB 11: 无中断信号输出 选择用于程序 0 的启动中断。

ADxPINTS1: PMD触发中断信号选择寄存器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	-	-	-	-	-	-	INTSEL1	
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"。
1-0	INTSEL1[1:0]	R/W	中断信号选择 00: 无中断信号输出 01: 保留 10: INTADxPDB 11: 无中断信号输出 选择用于程序1 的启动中断。

ADxPINTS2: PMD触发中断信号选择寄存器 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	-	-	-	-	-	-	INTSEL2	
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"。
1-0	INTSEL2[1:0]	R/W	中断信号选择 00: 无中断信号输出 01: 保留 10: INTADxPDB 11: 无中断信号输出 选择用于程序2的启动中断。

ADxPINTS3: PMD触发中断信号选择寄存器 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	-	-	-	-	-	-	INTSEL3	
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"。
1-0	INTSEL3[1:0]	R/W	中断信号选择 00: 无中断信号输出 01: 保留 10: INTADxPDB 11: 无中断信号输出 选择用于程序 3的启动中断。

ADxPINTS4: PMD触发中断信号选择寄存器 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	-	-	-	-	-	-	INTSEL4	
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"。
1-0	INTSEL4[1:0]	R/W	中断信号选择 00: 无中断信号输出 01: 保留 10: INTADxPDB 11: 无中断信号输出 选择用于程序 4 的启动中断。

ADxPINTS5: PMD触发脉冲中断信号选择寄存器 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	-	-	-	-	-	-	INTSEL5	
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"。
1-0	INTSEL5[1:0]	R/W	中断信号选择 00: 无中断信号输出 01: 保留 10: INTADxPDB 11: 无中断信号输出 选择用于程序 5 的启动中断。

13.4.22.3 ADxPSET0~5(PMD 触发程序寄存器 0~5)

各ADxPSETn (n=0~5: 程序号)由四个组(这些组假定<AINSPnm [4:0]>, <UVWISnm[1:0]>和<ENSPnm>在一个耦合中)组成。(m=0~3)(x=B : ADC单元)

将<ENSPnm>设置为"1", 即可启用UVWISnm[1:0]>, 各<AINSPnm[4:0]>位用于选择拟使用的AIN引脚。利用这些条件启动AD转换, 然后将结果存储到转换结果寄存器中。

ADxREGm ADxPSETn	m=0	m=1	m=2	m=3
n=0	<ENSP00> <UVWIS00> <AINSP00>	<ENSP01> <UVWIS01> <AINSP01>	<ENSP02> <UVWIS02> <AINSP02>	<ENSP03> <UVWIS03> <AINSP03>
n=1	<ENSP10> <UVWIS10> <AINSP10>	<ENSP11> <UVWIS11> <AINSP11>	<ENSP12> <UVWIS12> <AINSP12>	<ENSP13> <UVWIS13> <AINSP13>
n=2	<ENSP20> <UVWIS20> <AINSP20>	<ENSP21> <UVWIS21> <AINSP21>	<ENSP22> <UVWIS22> <AINSP22>	<ENSP23> <UVWIS23> <AINSP23>
n=3	<ENSP30> <UVWIS30> <AINSP30>	<ENSP31> <UVWIS31> <AINSP31>	<ENSP32> <UVWIS32> <AINSP32>	<ENSP33> <UVWIS33> <AINSP33>
n=4	<ENSP40> <UVWIS40> <AINSP40>	<ENSP41> <UVWIS41> <AINSP41>	<ENSP42> <UVWIS42> <AINSP42>	<ENSP43> <UVWIS43> <AINSP43>
n=5	<ENSP50> <UVWIS50> <AINSP50>	<ENSP51> <UVWIS51> <AINSP51>	<ENSP52> <UVWIS52> <AINSP52>	<ENSP53> <UVWIS53> <AINSP53>

表13-2 AIN引脚选择

<AINSP00 [4:0]> ~ <AINSP53 [4:0]>	ADC单元 B
0_0000	: 保留
0_0001	: 保留
0_0010	: 保留
0_0011	: 保留
0_0100	: 保留
0_0101	: 保留
0_0110	: 保留
0_0111	: 保留
0_1000	: 保留
0_1001	: AINB9
0_1010	: AINA10
0_1011	: AINB11
0_1100	: AINB12
0_1101	: 保留
0_1110	: 保留
0_1111	: 保留
1_0000	: AINB16
0_1101 ~1_1111	: 保留

ADxPSET0: PMD触发程序寄存器0

	31	30	29	28	27	26	25	24
比特符号	ENSP03	UVWIS03			AINSP03			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP02	UVWIS02			AINSP02			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP01	UVWIS01			AINSP01			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP00	UVWIS00			AINSP00			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSP03	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	UVWIS03[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
28-24	AINSP03[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
23	ENSP02	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	UVWIS02[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
20-16	AINSP02[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
15	ENSP01	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	UVWIS01[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
12-8	AINSP01[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
7	ENSP00	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	UVWIS00[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
4-0	AINSP00[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。

相位选择

00	未规定
01	U
10	V
11	W

ADxPSET1: PMD触发程序寄存器1

	31	30	29	28	27	26	25	24
比特符号	ENSP13	UVWIS13			AINSP13			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP12	UVWIS12			AINSP12			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP11	UVWIS11			AINSP11			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP10	UVWIS10			AINSP10			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSP13	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	UVWIS13[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
28-24	AINSP13[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
23	ENSP12	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	UVWIS12[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
20-16	AINSP12[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
15	ENSP11	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	UVWIS11[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
12-8	AINSP11[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
7	ENSP10	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	UVWIS10[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
4-0	AINSP10[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。

相位选择

00	未规定
01	U
10	V
11	W

ADxPSET2: PMD触发程序寄存器2

	31	30	29	28	27	26	25	24
比特符号	ENSP23	UVWIS23			AINSP23			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP22	UVWIS22			AINSP22			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP21	UVWIS21			AINSP21			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP20	UVWIS20			AINSP20			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSP23	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	UVWIS23[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
28-24	AINSP23[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
23	ENSP22	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	UVWIS22[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
20-16	AINSP22[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
15	ENSP21	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	UVWIS21[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
12-8	AINSP21[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
7	ENSP20	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	UVWIS20[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
4-0	AINSP20[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。

相位选择

00	未规定
01	U
10	V
11	W

ADxPSET3: PMD触发程序寄存器3

	31	30	29	28	27	26	25	24
比特符号	ENSP33	UVWIS33			AINSP33			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP32	UVWIS32			AINSP32			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP31	UVWIS31			AINSP31			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP30	UVWIS30			AINSP30			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSP33	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	UVWIS33[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
28-24	AINSP33[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
23	ENSP32	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	UVWIS32[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
20-16	AINSP32[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
15	ENSP31	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	UVWIS31[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
12-8	AINSP31[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
7	ENSP30	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	UVWIS30[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
4-0	AINSP30[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。

相位选择

00	未规定
01	U
10	V
11	W

ADxPSET4: PMD触发程序寄存器4

	31	30	29	28	27	26	25	24
比特符号	ENSP43	UVWIS43			AINSP43			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP42	UVWIS42			AINSP42			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP41	UVWIS41			AINSP41			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP40	UVWIS40			AINSP40			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSP43	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	UVWIS43[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
28-24	AINSP43[4:0]	R/W	AIN选择 见"表13-2 选择AIN引脚".
23	ENSP42	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	UVWIS42[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
20-16	AINSP42[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择".
15	ENSP41	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	UVWIS41[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
12-8	AINSP41[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择".
7	ENSP40	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	UVWIS40[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
4-0	AINSP40[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择".

相位选择

00	未规定
01	U
10	V
11	W

ADxPSET5: PMD触发程序寄存器5

	31	30	29	28	27	26	25	24
比特符号	ENSP53	UVWIS53			AINSP53			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP52	UVWIS52			AINSP52			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP51	UVWIS51			AINSP51			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP50	UVWIS50			AINSP50			
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSP53	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	UVWIS53[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
28-24	AINSP53[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
23	ENSP52	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	UVWIS52[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
20-16	AINSP52[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
15	ENSP51	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	UVWIS51[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
12-8	AINSP51[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。
7	ENSP50	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	UVWIS50[1:0]	R/W	相位选择(适用于矢量引擎)见下表。
4-0	AINSP50[4:0]	R/W	AIN选择 见"表13-2 AIN引脚选择"。

相位选择

00	未规定
01	U
10	V
11	W

13.4.23 ADxTSET03 / ADxTSET47 / ADxTSET811 (定时器触发程序寄存器)

AD转换可通过作为触发的定时器5 (TMRB5)生成的INTTB51启动。编程定时触发有十二个8 -位寄存器。将<ENSTm>设置为 "1", 即可启用该ADxTSETm寄存器。<AINSTm [4:0]>用于选择待用AIN引脚。定时器触发程序寄存器的数目, 与AD转换结果寄存器的数目相同。该AD转换完成时, 中断: INTADxTMR即生成。

(m=0 ~ 11), (x=B : ADC单元)

表13-3 AIN引脚的选择

<AINST0 [4:0]> ~ <AINST11 [4:0]>	ADC单元B
0_0000	: 保留
0_0001	: 保留
0_0010	: 保留
0_0011	: 保留
0_0100	: 保留
0_0101	: 保留
0_0110	: 保留
0_0111	: 保留
0_1000	: 保留
0_1001	: AINB9
0_1010	: AINA10
0_1011	: AINB11
0_1100	: AINB12
0_1101	: 保留
0_1110	: 保留
0_1111	: 保留
1_0000	: AINB16
0_1101 ~ 1_1111	: 保留

ADxTSET03定时器触发程序寄存器 03

	31	30	29	28	27	26	25	24
比特符号	ENST3	-	-	AINST3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENST2	-	-	AINST2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENST1	-	-	AINST1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENST0	-	-	AINST0				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENST3	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINST3[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。
23	ENST2	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINST2[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。
15	ENST1	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINST1[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。
7	ENST0	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINST0[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。

ADxTSET47定时器触发程序寄存器 47

	31	30	29	28	27	26	25	24
比特符号	ENST7	-	-	AINST7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENST6	-	-	AINST6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENST5	-	-	AINST5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENST4	-	-	AINST4				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENST7	R/W	ADxREG7启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINST7[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。
23	ENST6	R/W	ADxREG6启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINST6[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。
15	ENST5	R/W	ADxREG5启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINST5[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。
7	ENST4	R/W	ADxREG4启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINST4[4:0]	R/W	AIN选择 见"表13-3 AIN引脚选择"。

ADxTSET811 定时器触发程序寄存器 811

	31	30	29	28	27	26	25	24
比特符号	ENST11	-	-	AINST11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENST10	-	-	AINST10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENST9	-	-	AINST9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENST8	-	-	AINST8				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENST11	R/W	ADxREG11 启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINST11[4:0]	R/W	AIN 选择 见"表13-3 AIN 引脚选择"。
23	ENST10	R/W	ADxREG10 启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINST10[4:0]	R/W	AIN 选择 "表13-4 AIN 引脚选择"。
15	ENST9	R/W	ADxREG9 启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINST9[4:0]	R/W	AIN 选择 见"表13-3 AIN 引脚选择"。
7	ENST8	R/W	ADxREG8 启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINST8[4:0]	R/W	AIN 选择 见"表13-3 AIN 引脚选择"。

の

13.4.24 ADxSSET03 / ADxSSET47 / ADxSSET811(软件触发程序寄存器)

AD可通过软件启动。共有十二个8 -位寄存器供编程软件触发使用。将<ENSSm>设置为 "1" 启用 ADxSSETm 寄存器。<AINSSm[4:0]>用于选择待用AIN引脚。软件触发程序寄存器的数目，与转换寄存器的数目相同。AD转换完成时，中断：INTADxSFT即生成。

(m=0 ~ 11), (x=B : ADC单元)

表13-4 AIN引脚选择

<AINSS0 [4:0]> ~ <AINSS11 [4:0]>	ADC单元B
0_0000	: 保留
0_0001	: 保留
0_0010	: 保留
0_0011	: 保留
0_0100	: 保留
0_0101	: 保留
0_0110	: 保留
0_0111	: 保留
0_1000	: 保留
0_1001	: AINB9
0_1010	: AINA10
0_1011	: AINB11
0_1100	: AINB12
0_1101	: 保留
0_1110	: 保留
0_1111	: 保留
1_0000	: AINB16
0_1101 ~ 1_1111	: 保留

ADxSSET03软件触发程序寄存器 03

	31	30	29	28	27	26	25	24
比特符号	ENSS3	-	-	AINSS3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSS2	-	-	AINSS2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSS1	-	-	AINSS1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSS0	-	-	AINSS0				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSS3	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINSS3[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
23	ENSS2	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINSS2[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
15	ENSS1	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINSS1[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
7	ENSS0	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINSS0[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。

ADxSSET47软件触发程序寄存器 47

	31	30	29	28	27	26	25	24
比特符号	ENSS7	-	-	AINSS7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSS6	-	-	AINSS6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSS5	-	-	AINSS5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSS4	-	-	AINSS4				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSS7	R/W	ADxREG7启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINSS7[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
23	ENSS6	R/W	ADxREG6启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINSS6[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
15	ENSS5	R/W	ADxREG5启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINSS5[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
7	ENSS4	R/W	ADxREG4启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINSS4[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。

ADxSSET811软件触发程序寄存器 811

	31	30	29	28	27	26	25	24
比特符号	ENSS11	-	-	AINSS11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSS10	-	-	AINSS10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSS9	-	-	AINSS9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSS8	-	-	AINSS8				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSS11	R/W	ADxREG11启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINSS11[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
23	ENSS10	R/W	ADxREG10启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINSS10[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
15	ENSS9	R/W	ADxREG9启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINSS9[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。
7	ENSS8	R/W	ADxREG8启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINSS8[4:0]	R/W	AIN选择 见"表13-4 AIN引脚选择"。

13.4.25 ADxASET03 / ADxASET47 / ADxASET811(常量转换程序寄存器)

ADC使转换触发不断启用。目前，共有十二个8位寄存器用于常量触发编程。将<ENSA_m>设置为"1"启用ADxASET_m寄存器。<AINSA_m[4:0]>用于选择将使用的AIN引脚。常量触发程序寄存器的数目，与转换结果寄存器的数目相同。

(m=0 ~ 11), (x=B : ADC单元)

表13-5 AIN引脚选择

<AINSA0 [4:0]> ~ <AINSA11 [4:0]>	ADC单元B
0_0000	: 保留
0_0001	: 保留
0_0010	: 保留
0_0011	: 保留
0_0100	: 保留
0_0101	: 保留
0_0110	: 保留
0_0111	: 保留
0_1000	: 保留
0_1001	: AINB9
0_1010	: AINA10
0_1011	: AINB11
0_1100	: AINB12
0_1101	: 保留
0_1110	: 保留
0_1111	: 保留
1_0000	: AINB16
0_1101 ~ 1_1111	: 保留

ADxASET03常量转换程序寄存器 03

	31	30	29	28	27	26	25	24
比特符号	ENSA3	-	-	AINSA3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSA2	-	-	AINSA2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSA1	-	-	AINSA1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSA0	-	-	AINSA0				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSA3	R/W	ADxREG3启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINSA3[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
23	ENSA2	R/W	ADxREG2启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINSA2[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
15	ENSA1	R/W	ADxREG1启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINSA1[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
7	ENSA0	R/W	ADxREG0启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINSA0[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"

ADxASET47常量转换程序寄存器 47

	31	30	29	28	27	26	25	24
比特符号	ENSA7	-	-	AINSA7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSA6	-	-	AINSA6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSA5	-	-	AINSA5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSA4	-	-	AINSA4				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSA7	R/W	ADxREG7启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINSA7[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
23	ENSA6	R/W	ADxREG6启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINSA6[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
15	ENSA5	R/W	ADxREG5启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINSA5[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
7	ENSA4	R/W	ADxREG4启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINSA4[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"

ADxASET811常量转换程序寄存器 811

	31	30	29	28	27	26	25	24
比特符号	ENSA11	-	-	AINSA11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSA10	-	-	AINSA10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSA9	-	-	AINSA9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSA8	-	-	AINSA8				
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31	ENSA11	R/W	ADxREG11启用 0: 禁用 1: 启用
30-29	-	R	读作"0"。
28-24	AINSA11[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
23	ENSA10	R/W	ADxREG10启用 0: 禁用 1: 启用
22-21	-	R	读作"0"。
20-16	AINSA10[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
15	ENSA9	R/W	ADxREG9启用 0: 禁用 1: 启用
14-13	-	R	读作"0"。
12-8	AINSA9[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"
7	ENSA8	R/W	ADxREG8启用 0: 禁用 1: 启用
6-5	-	R	读作"0"。
4-0	AINSA8[4:0]	R/W	AIN选择 见"表13-5 AIN引脚选择"

13.5 操作说明

13.5.1 模拟标准电压

对于高电平与低电平模拟参标准电压，VREFHB与VREFLB引脚可用于ADC B。
内部放大器共用电源与地线(其分别与AMPVDD5和AMPVSS连接)。

注1: AD转换期间，不要改变端口J/K的输出数据，以免影响到转换结果。

注2: 以下情况可能导致AD转换结果不稳定。

输入操作完成。

输出操作完成。

端口变量输出电流。

可求出多个转换结果的平均值，从而获得精确值。

13.5.2 启动 AD 转换

可利用软件或以下三个触发信号的一个启动AD转换。

PMD 触发(见 "13.4.22 PMD触发程序寄存器”。)

定时器触发(TMRB5)(见"13.4.23 定时器触发程序寄存器”。)

软件触发(见 "13.4.24 软件触发程序寄存器”。)

这些启动触发的优先级如下。

PMD触发0 > ... > PMD触发5 > 定时器触发 > 软件触发 > 常量触发

如果在AD转换进行期间出现PMD触发，应处理PMD触发，使正在执行的程序停止，并启动与PMD触发编号对应的AD转换。

如果在AD转换期间发生优先级较高的触发，则应在正在执行的程序完成后，处理优先级较高的触发。

从触发的生成到AD转换的启动之间，会存在一些滞后现象。滞后取决于触发。以下时序图和列表格列出了这种时延。

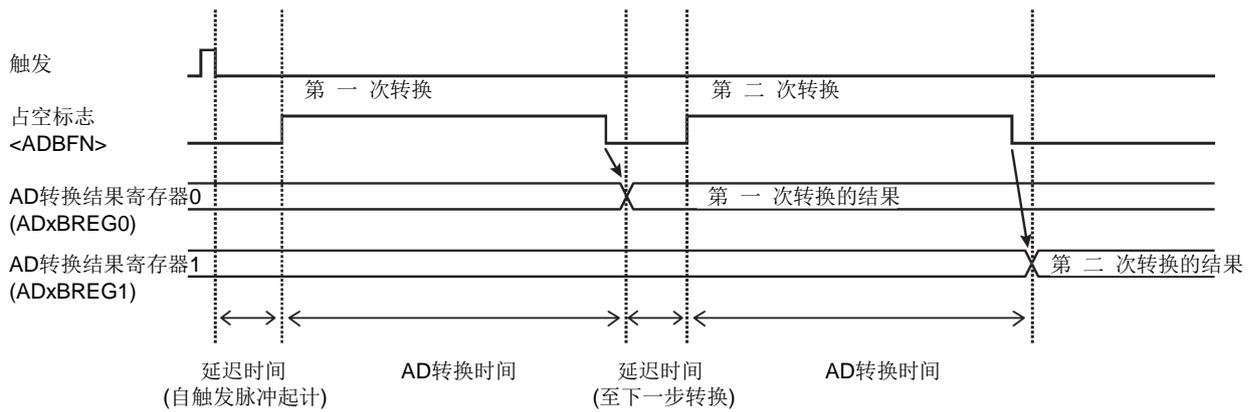


图13-3 AD转换的时序图

表13-6 AD转换时间(SCLK = 40 MHz)

		f _{sys} = 40 MHz	
		触发器	MIN
自触发脉冲起计的 延迟时间 (注 1)	PMD	0.225	0.3
	TMRB	0.225	0.5
	软件, 常量	0.25	0.525
AD 转换时间 [μs]	-	1.85	
延迟时间至下一步 转换[μs] (注 2)	PMD	0.175	0.225
	TMRB, 软件, 常量	0.175	0.425

注1: 从AD转换触发到启动之间的延迟时间。

注2: 用一个触发进行多重转换时, 至第2次转换时的延迟时间, 或转换后的延迟时间。

13.5.3 AD 转换监控功能

ADC具有AD转换监视功能。在该功能处于已启用状态时, 一旦某个转换结果与所指定的比较值匹配, 即生成一次中断。

将ADxCMPCR0<CMP0EN>或ADxCMPCR1<CMP1EN>设置为"1", 即可启用监控功能。在监视功能运行期间, 如果由监控功能负责监控的AD转换结果寄存器的值与ADxCMCR0<ADBIG0>/ADxCMCR1<ADBIG1>所指定的比较条件相符, 则可生成中断(INTADxCPA(适用于ADxCMPCR0), INTADxCPB(适用于ADxCMPCR1)。可在将转换结果存入到该寄存器时执行该比较。

注1: 比较功能不会清除AD转换结果存储标志(<ADR_xRF>)。

注2: 比较功能与通过软件读取转换结果的方式有所不同。因此, 如果在下一步转换完成时未读取先前的结果, 则会设置溢位标志(<OVR_x>)。

13.6 AD 转换时序图

下述为通过软件触发转换，常量转换和触发验收的时序图。

13.6.1 软件触发转换

在进行软件触发转换时，在通过ADxSSET03，ADxSSET47和ADxSSET811编程的转换完成后可生成中断(图13-4)。

如果ADxMOD1<ADEN>在AD转换期间被清"0"，则进行中的转换会停止，且不会存入到该结果寄存器中(图13-5)。

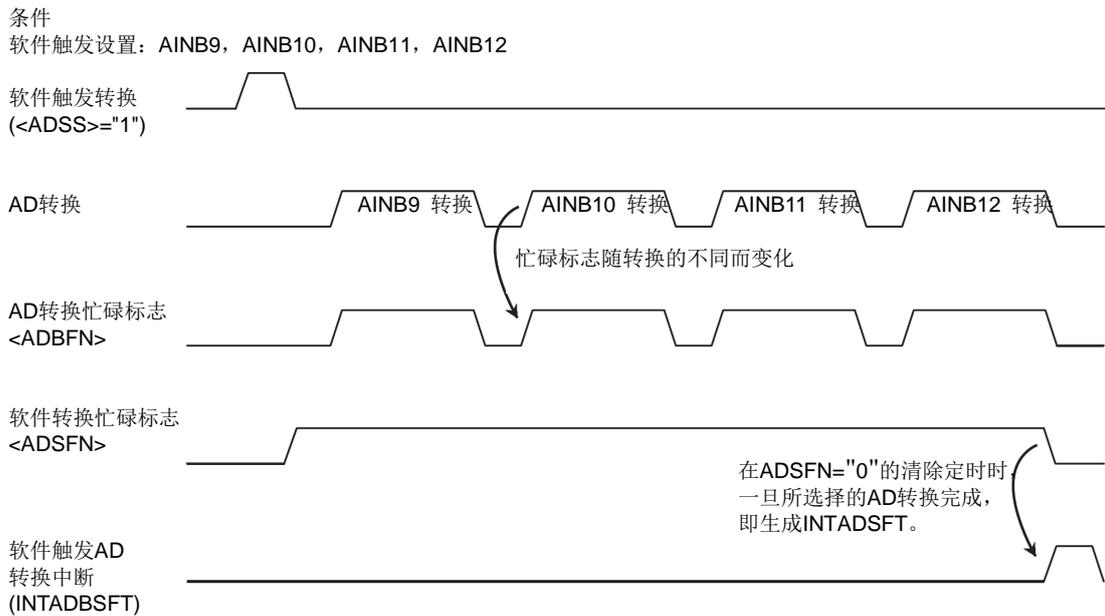


图13-4 软件触发AD转换

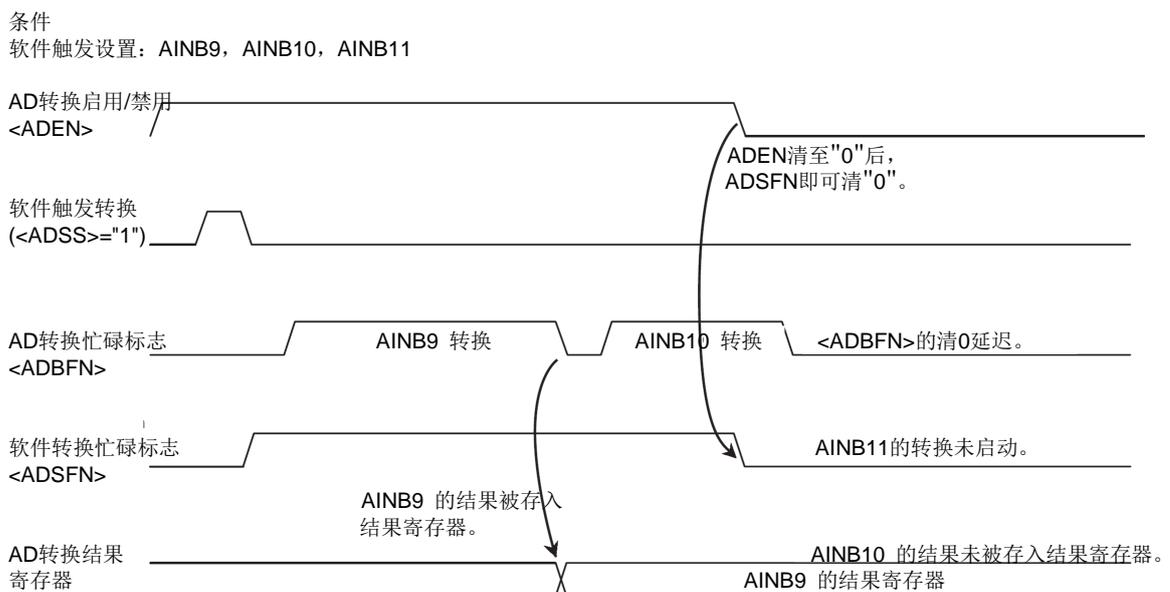


图13-5 在软件触发AD转换期间将"0"写到<ADEN>

13.6.2 常量转换

在常量转换过程中，如在下一转换完成时未从该转换结果寄存器读取前一结果，则溢位标记即被设置为 "1"。在这种情况下，下一个结果会覆盖该转换结果寄存器中的前一转换结果。通过读取该转换结果，溢位标记即被清除(图13-6)。

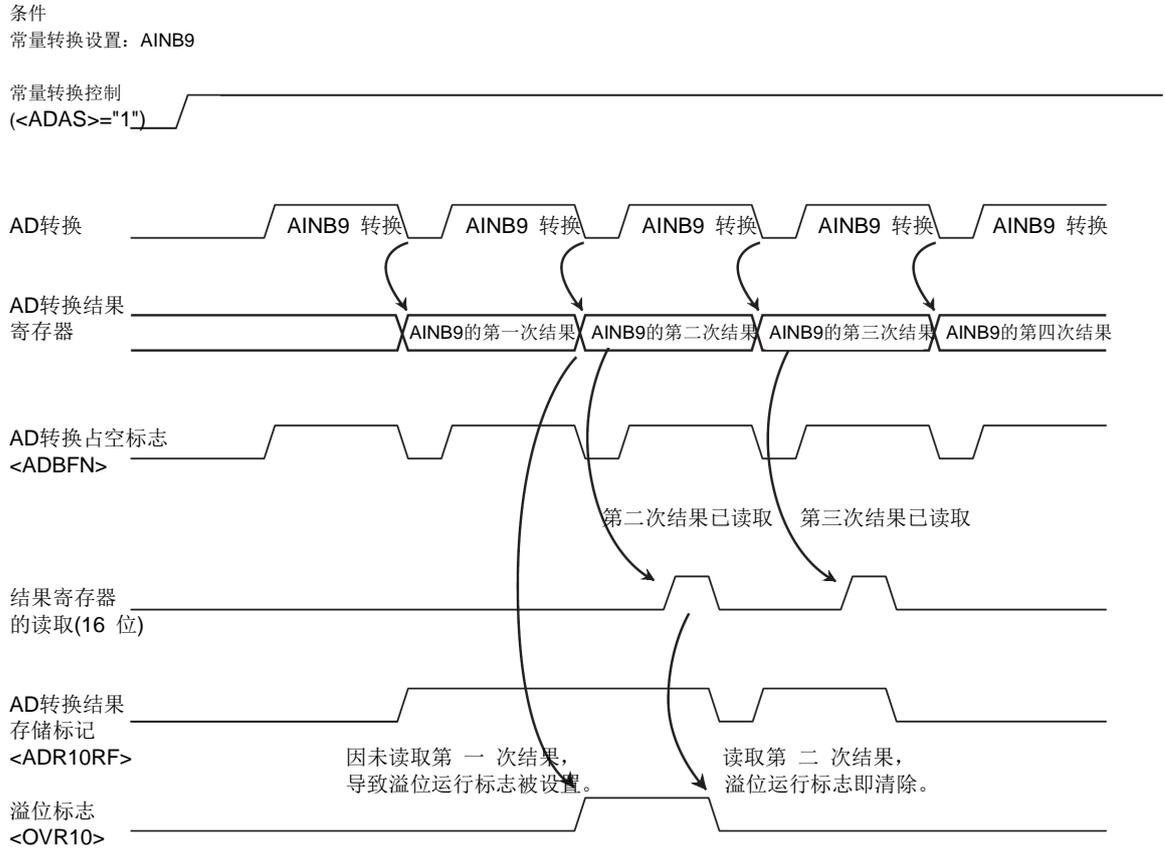


图13-6 常量转换

13.6.3 通过触发进行 AD 转换

如果在软件触发转换期间发生PMD触发，则进行中的转换会立即(图13-7)停止。如果在软件触发转换期间发生定时器触发，则进行中的转换会在该进行中的转换完成之后停止(图13-8)。在通过触发进行的转换完成之后，会从由ADxSSET03，ADxSSET47和ADxSSET811编程的开头部分开始软件触发转换(图13-9)。

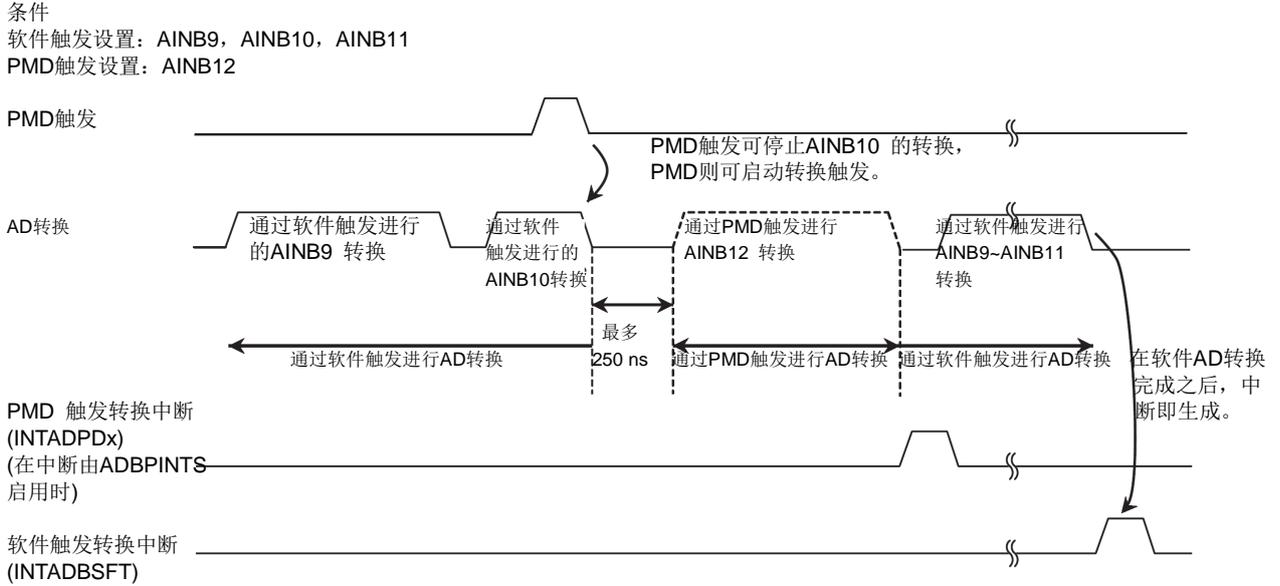


图13-7 通过PMD触发进行AD转换

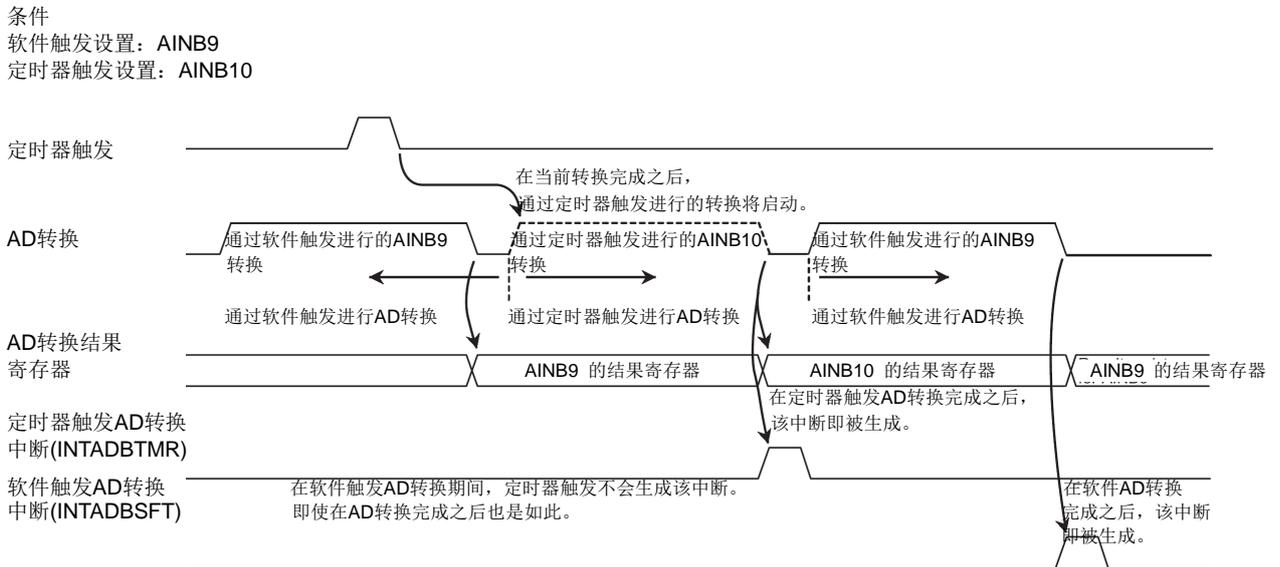


图13-8 通过定时器触发进行的AD转换(1)

条件
 软件触发设置: AINB9, AINB10, AINB11
 定时器 触发设置 : AINB12

定时器触发

AD转换

AD转换结果寄存器

定时器触发AD转换中断 (INTADBTMR)

软件触发AD转换中断 (INTADBSFT)

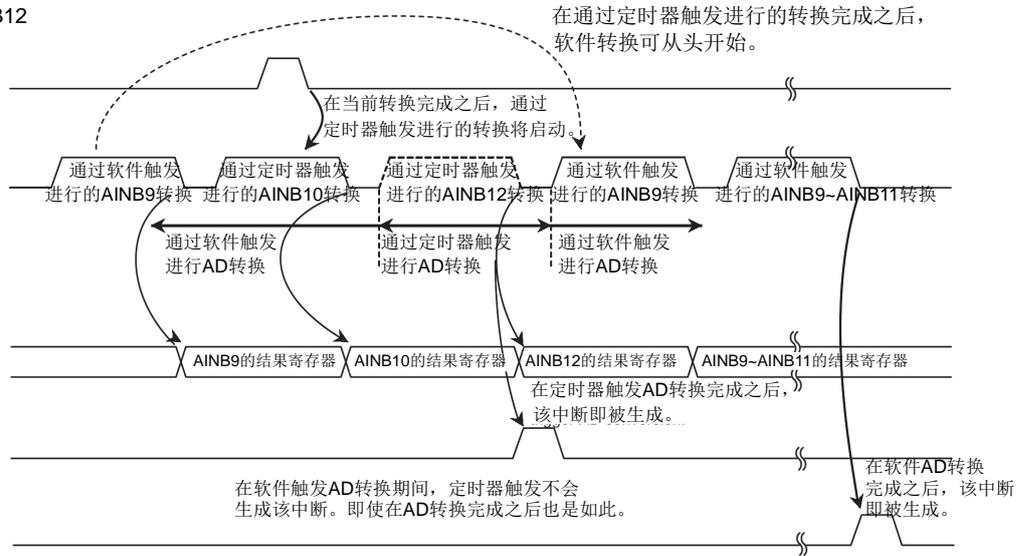
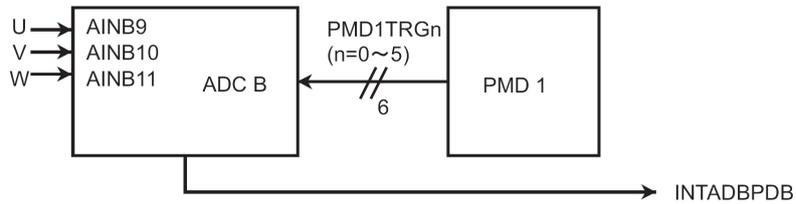


图13-9 通过定时器触发进行的AD转换 (2)

13.7 使用举例

13.7.1 利用一个 PMD1(三个分流器)和一个 ADC 进行的连续转换

下图为一个用于三分路和一个ADC的PMD1进行AD转换的电路图。



ADC设置举例如下所示。

ADC单元B

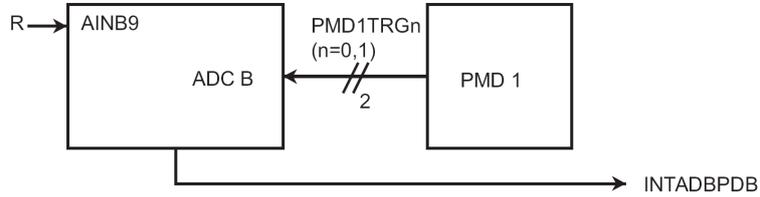
程序	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	B	B	B	B	B	B

程序 0~5 被分配到触发输入端PMD1TRG0 ~5 上。"reg0"和"reg1"表示PMD触发程序寄存器 ADBPSETn[7:0]和ADBPSETn[15:8]。"U", "V"和"W"表示马达的相位。选择AIN输入即可获得这些相位。

发生触发输入时, 根据reg0和reg1顺次进行AD转换, 然后即生成中断信号(INTADBPDB)。

13.7.2 利用一个PMD1(一条分路)与一个ADC进行连续转换

以下给出了利用一个PMD1 与一条分路和一个ADC配套，进行AD转换的电路图。



ADC设置示例如下所示。

ADC 单元B

触发脉冲	PMD1	PMD1
	0	1
程序	0	1
reg0	R	-
reg1	-	R
INT	-	B

将程序 0 与1 从PMD1 分配至两个触发信号上。

"reg0"和"reg1"表示PMD触发程序寄存器ADBPSETn[7:0]和ADBPSETn[15:8]。"R"表示一电阻器，并设置了被连接至该电阻器的AIN输入。

在一个触发输入出现时，ADC即启动，并连续执行程序 0 与1。在程序1完成时，中断(INTADBPDB)即被生成。



14. 马达控制电路(PMD: 可编程马达驱动器)

TMPM375FSDMG配备有1 个通道可编程马达驱动器(PMD)。本产品上的PMD，可控制与一 台矢量引擎 (VE+) 和一个模拟/数字转换器(ADC)配套使用的矢量马达。通过矢量引擎的命令，激活脉冲宽度调制电路，导通控制和同步触发脉冲发生器。同步触发生成的电路，可命令AD转换器启动ADC转换。

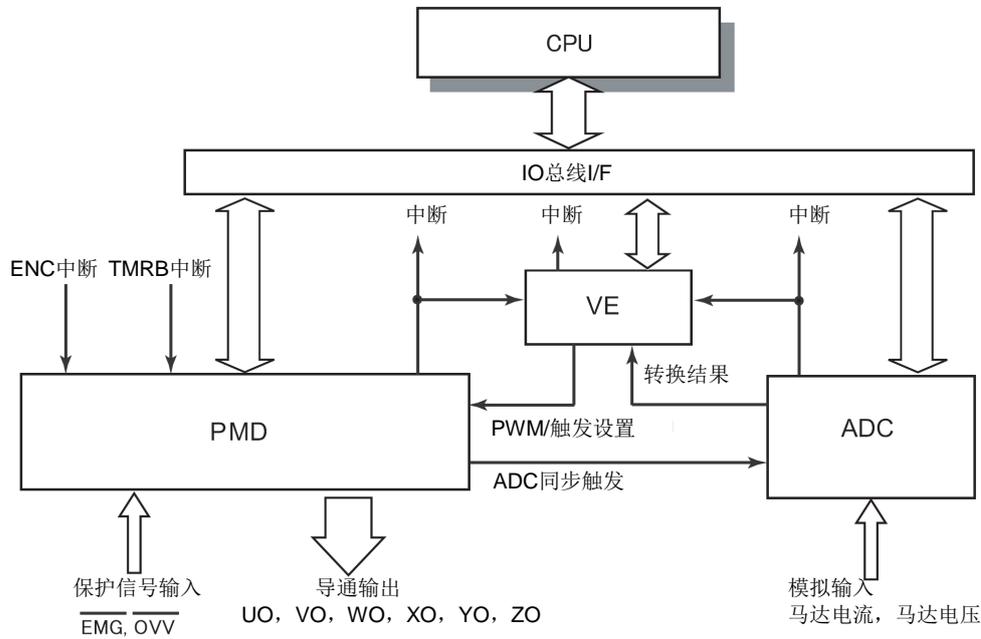


图14-1 与马达控制有关的功能方块图

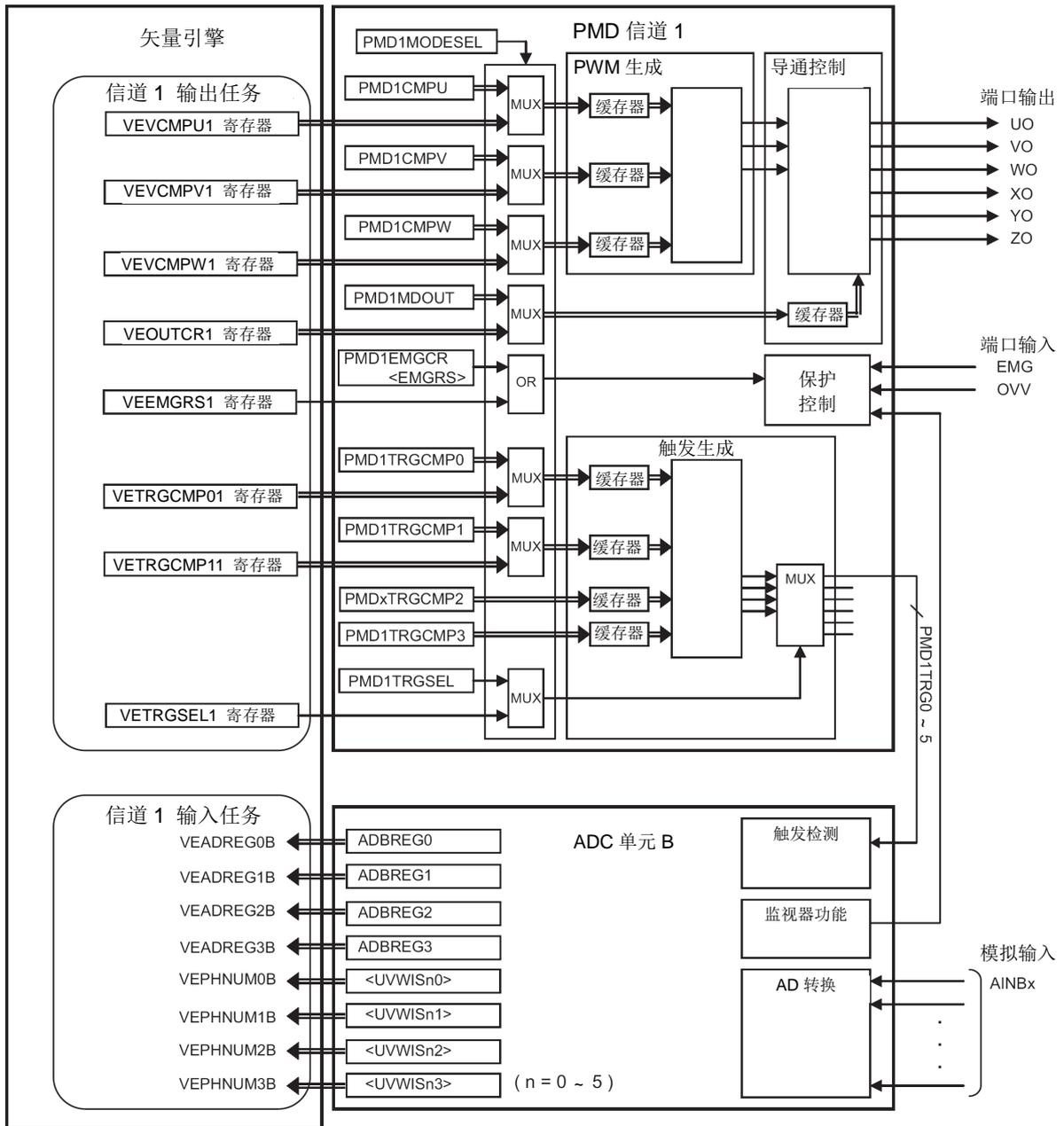


图14-2 马达控制电路，矢量引擎和A/D转换器的相关示意图

14.1 PMD 输入/输出信号

下表给出了输入为可编程马达驱动器，或从可编程马达驱动器输出的信号。

表14-1 输入/输出信号

通道	引脚名称	PMD信号名	说明
PMD1	PG6/ $\overline{\text{EMG}}/\overline{\text{OVV}}$	$\overline{\text{EMG}}/\overline{\text{OVV}}$	EMG状态信号/OVV状态信号
	PG5/UO	UO	U相输出
	PG4/XO	XO	X相输出
	PG3/VO	VO	V相输出
	PG2/YO	YO	Y相输出
	PG1/WO	WO	W相输出
	PG0/ZO	ZO	Z相输出

14.2 PDM 电路配置

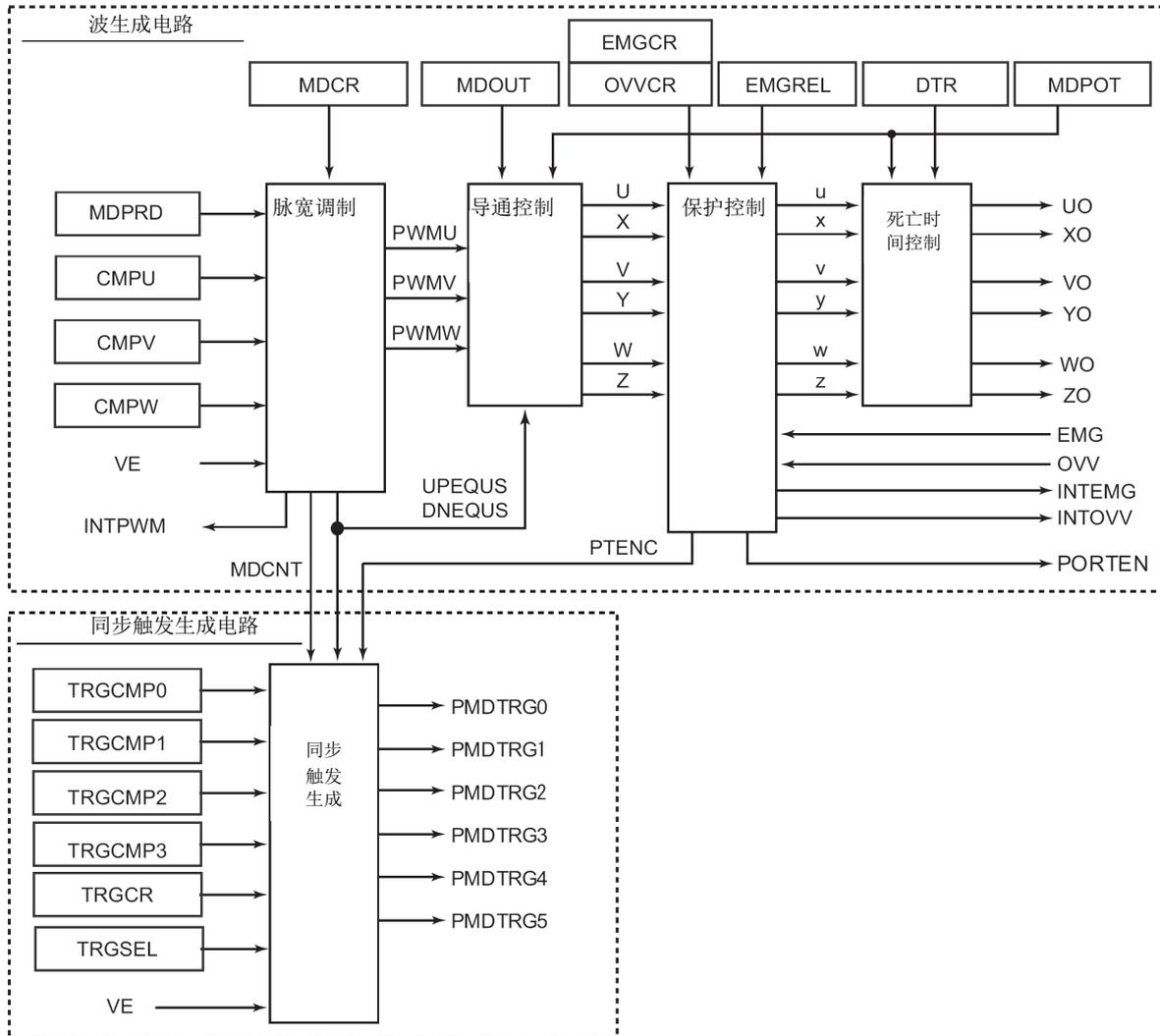


图14-3 PMD电路方块图

PMD电路由两个部分，即波发生电路和同步触发发生电路构成。波生成电路包括一个脉宽调制电路，一个导通控制电路，一个防护控制电路，以及一个停滞时间控制电路。

脉宽调制电路具备共用PWM载波波形，并可生成独立的 3-相PWM波形。

导通控制电路决定U相，V相和W相各上下侧的输出型式。

该防护控制电路可通过EMG输入和OVV输入控制应急输出停止。

停滞时间控制电路可防止在上侧和下侧切换时发生短路。

同步触发脉冲生成电路可生成同步触发信号并将其发送给该AD转换器。

14.3 PMD 寄存器

下表列出了与PMD相关的寄存器。

基本地址 = 0x4005_0480

寄存器名称		地址(基+)
PMD启用寄存器	PMD1MDEN	0x0000
端口输出模式寄存器	PMD1PORTMD	0x0004
PMD控制寄存器	PMD1MDCR	0x0008
PWM计数器状态寄存器	PMD1CNTSTA	0x000C
PWM计数寄存器	PMD1MDCNT	0x0010
PWM周期寄存器	PMD1MDPRD	0x0014
PMD比较U寄存器	PMD1CMPU	0x0018
PMD比较V寄存器	PMD1CMPV	0x001C
PMD比较W寄存器	PMD1CMPW	0x0020
模式选择寄存器	PMD1MODESEL	0x0024
PMD导通控制寄存器	PMD1MDOUT	0x0028
PMD输出设置寄存器	PMD1MDPOT	0x002C
EMG释放寄存器	PMD1EMGREL	0x0030
EMG控制寄存器	PMD1EMGCR	0x0034
EMG状态寄存器	PMD1EMGSTA	0x0038
OVV控制寄存器	PMD1OVVCR	0x003C
OVV状态寄存器	PMD1OVVSTA	0x0040
停滞时间寄存器	PMD1DTR	0x0044
触发脉冲比较 0 寄存器	PMD1TRGCMP0	0x0048
触发比较1 寄存器	PMD1TRGCMP1	0x004C
触发比较 2 寄存器	PMD1TRGCMP2	0x0050
触发比较 3 寄存器	PMD1TRGCMP3	0x0054
触发控制寄存器	PMD1TRGCR	0x0058
触发输出模式设置寄存器	PMD1TRGMD	0x005C
触发输出选择寄存器	PMD1TRGSEL	0x0060
触发更新时限设置寄存器	PMD1TRGSYNCR	0x0064
保留	-	0x0068
保留	-	0x006C
保留	-	0x0070
保留	-	0x0074
保留	-	0x0078
保留	-	0x007C

注：禁止访问各"保留"区域。

14.3.1 PMD1MDEN(PMD 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	-	-	-	-	-	-	-	PWMEN
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	PWMEN	R/W	启用或禁用波形合成 0: 禁用 1: 启用 注: 在该端口被设置为功能输出(PWM输出)时, 通过设置<PWMEN> = "0", 该端口即可禁止输出(高阻抗)。 注: 启用PMD之前, 设置 <PWMEN>="1"(启用)输出端口极性及其它相关的设置。

14.3.2 PMD1PORTMD(端口输出模式寄存器)

	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	-	-	-	-	-	-	PORTMD	
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。
1-0	PORTMD[1:0]	R/W R/W	<p>发生工具断开时的端口控制设置</p> <p>00: 上相位 = 高-z / 下相位 = 高-z</p> <p>01: 上相位 = 高-z / 下相位 = PMD输出</p> <p>10: 上相位 = PMD输出/下相位=高-z</p> <p>11: 上相位 = PMD输出/下相位=PMD输出</p> <p>在使用各端口进行功能输出(PWM输出)期间发生工具断开时, 可设置上相位(UO/VOWO)和下相位(XO/YO/ZO)的端口输出。</p> <p>在"高-Z"被选中期间发生工具断开时, 这些舱口即被禁止输出(高阻抗)。在其它情况下, 外部端口输出取决于PMD输出。</p>

注1: 在<PWMEN>=0 时, 无论PORTMD设置如何, 输出端口均被禁止输出(高阻抗)。

注2: 发生EMG输入时, 通过设置PMD1EMGCR<EMGMD[1:0]>, 即可对端口输出实施控制。

14.3.3 PMD1MODESEL (模式选择寄存器)

	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	DCMPEN	-	-	-	MDSEL3	MDSEL2	MDSEL1	MDSEL0
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	DCMPEN	R/W	在VE寄存器和PMD寄存器之间自动切换 0: 禁用这两个寄存器之间的切换控制(仅利用该寄存器设置<MDSEL0>。 1: 启用2个寄存器之间的切换控制(在PWM计数器的向上计数和向下计数状态)。 注: 在<MDSEL0>="1"时有效。 注: 在已选择三角载波(PMD1MDCR<PWMMD>="1")时有效。
6-4	-	R	读作0。
3	MDSEL3	R/W	模式选择 3 0: 总线模式(使用PMD寄存器: PMD1TRGSEL) 1: VE模式(使用VE寄存器: VETRGSEL)
2	MDSEL2	R/W	模式选择 2 0: 总线模式(使用PMD寄存器: PMD1TRGCMP0和PMD1TRGCMP1) 1: VE模式(使用VE寄存器: VETRGCMP01 和VETRGCMP11)
1	MDSEL1	R/W	模式选择1 0: 总线模式(使用PMD寄存器: PMD1MDOUT) 1: VE模式(使用VE寄存器: VEOUTCR1)
0	MDSEL0	R/W	模式选择 0 0: 总线模式(使用PMD寄存器: PMD1CMPU, PMD1CMPV1和PMD1CMPW) 1: VE模式(使用VE寄存器: VECMPU1, VECMPV1和VECMPW1 寄存器, 并将VEEMGRS1 寄存器设置已启用)

14.3.4 脉冲宽度调制电路

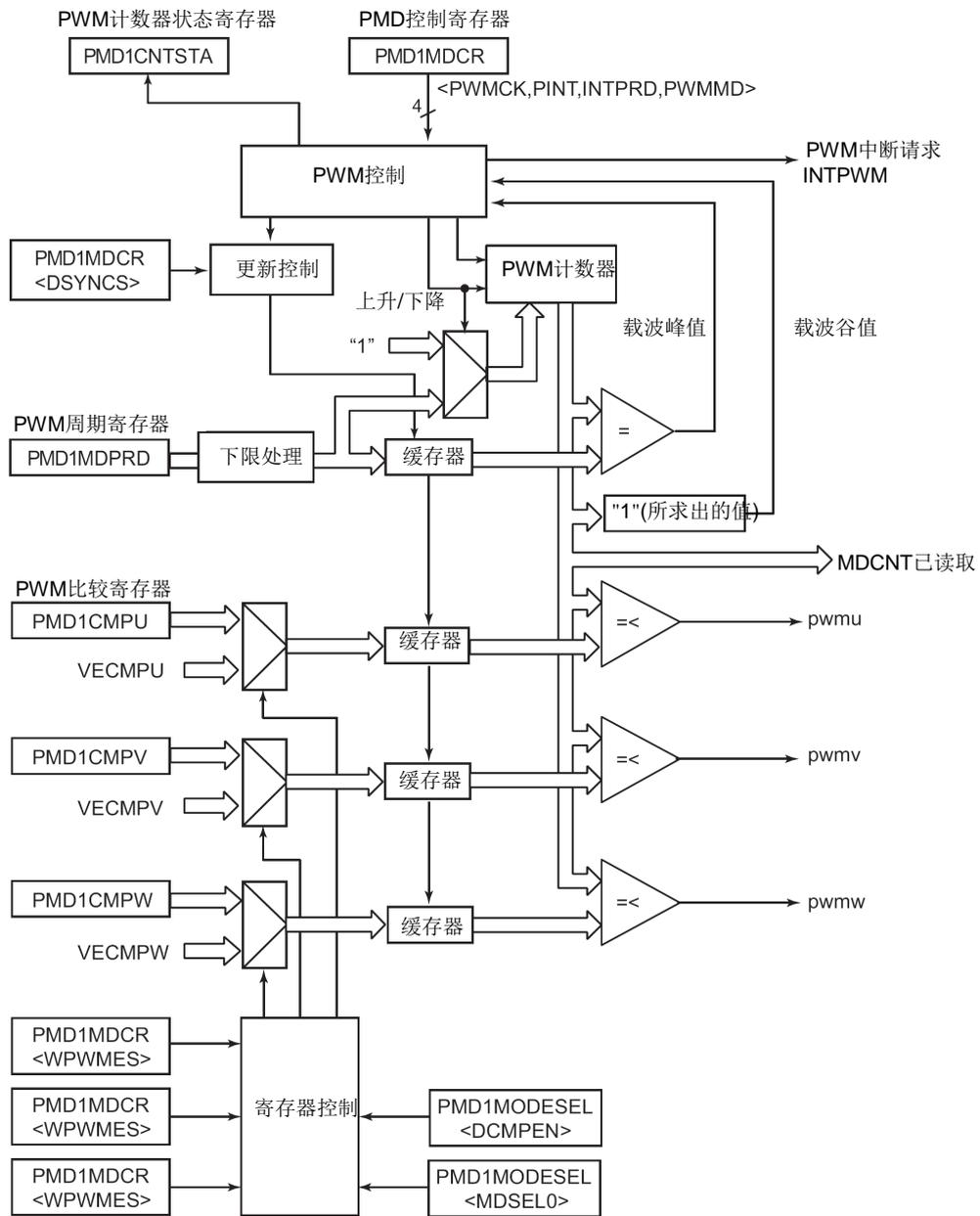


图14-4 脉冲宽度调制电路

脉宽调制电路有一个16位PWM向上/向下计数器，并可生成40 MHz时分辨率为25 nsec的PWM载波波形。PWM周期扩展模式(PMD1MDCR<PWMCK>="1")也可用。选择此模式时，PWM计数器可在40 MHz时生成分辨率为100 nsec的PWM载波波形。

可从模式0(沿对齐PWM，锯齿波调制)和模式1(中心对齐PWM，三角波调制)选择该PWM载波波形模式(见"图14-5 PWM 波形")。在三角波模式下，可从中心PWM，固定下降沿PWM和固定上升沿PWM中选择PWM波形(见"图14-6 使用固定沿的PWM三角波载波的波形")。

1. PWM周期设置

PWM周期取决于PMD1MDPRD寄存器。寄存器为双缓存型。后级缓存器在每个PWM周期更新一次。也可每半个PWM周期更新一次(见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMPU1/V1/W1缓存器更新定时")。

$$\text{锯齿波PWM: PMD1MDPRD寄存器值} = \frac{\text{振荡频率[Hz]}}{\text{PWM频率[Hz]}}$$

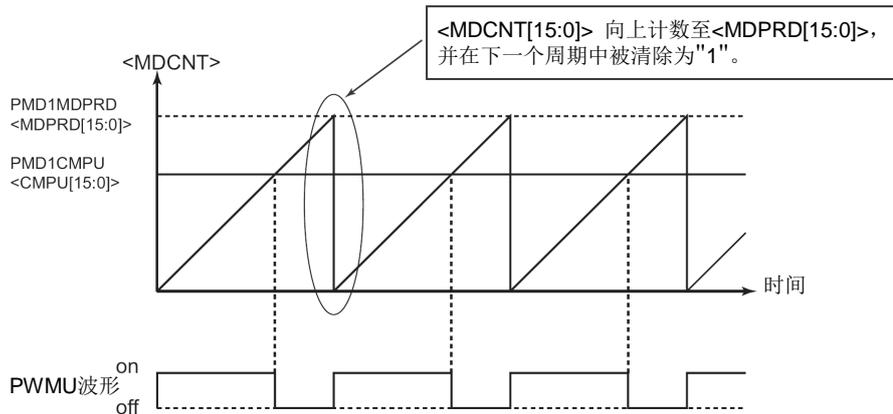
$$\text{三角波PWM: PMD1MDPRD寄存器值} = \frac{\text{振荡频率[Hz]}}{\text{PWM频率[Hz]} \times 2}$$

2. 比较功能

通过比较PWM比较寄存器(PMD1CMPU/V/W)和PWM计数器(PMD1MDCNT <MDCNT[15:0]>)所生成的PWM载波的大小, PWM电路可生成PWM波形。

各相位的PWM比较寄存器均有一个双缓存寄存器。在每个PWM周期, PWM比较寄存器值被加载到后级缓存器中。它也可每半个PWM周期更新一次(见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMPU1/V1/W1缓存器更新定时")。

[锯齿波]



[三角波]

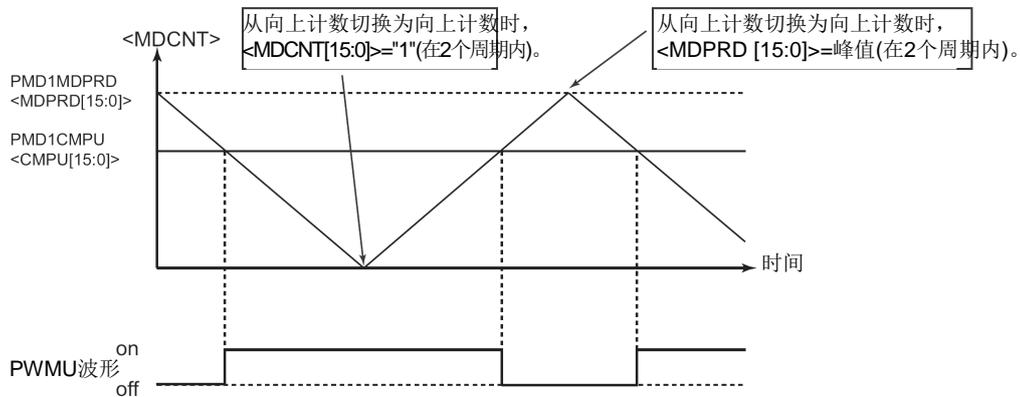


图14-5 PWM波形

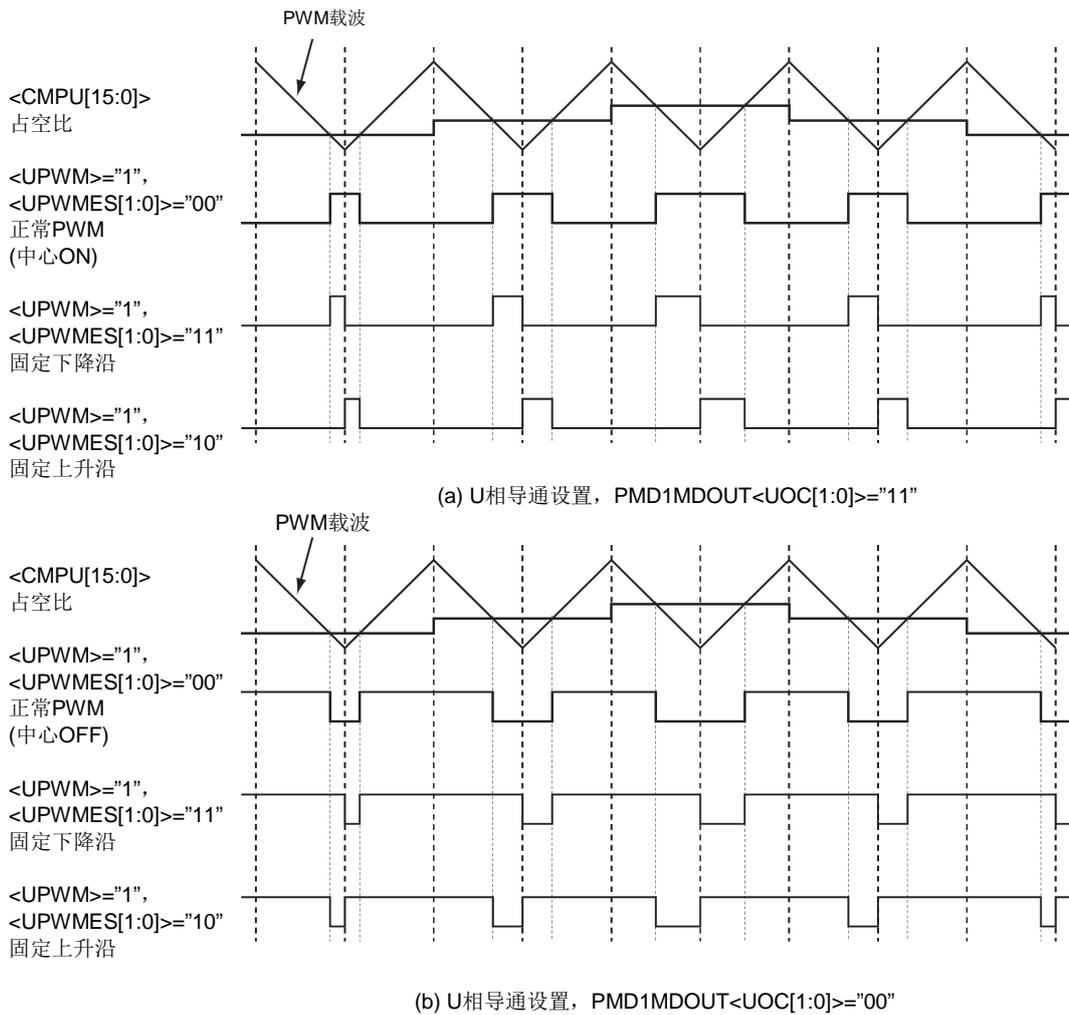


图14-6 PWM三角波载波的波形(采用固定缘)

3. 波形模式

在以下两种模式下, 可生成三相PWM波形:

1. 三相独立模式:

单独设置用于三相的每个PWM比较寄存器, 生成每个相的独立PWM波形。此模式用于生成正弦波等驱动器波形。

2. 三相共模:

仅U相PWM比较寄存器被设置为可为这三相生成完全相同的PWM波形。此模式用于无刷直流电动机的矩形波驱动。

4. 中断处理

PWM电路可生成与PWM波形同步的PWM中断请求。可在PWM 载波峰值或PWM载波谷值时选择中断请求定时。

可将PWM中断周期设置为半个PWM周期, 一个PWM 周期, 两个PWM周期或四个PWM周期。

14.3.4.1 PMD1MDCR(PMD 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	WPWMES		VPWMES		UPWMES		DSYNCS	
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
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。
15-14	WPWMES[1:0]	R/W	W相边缘设置 00: 边缘不固定(中心对齐PWM) 01: 保留 10: PWM上升缘固定(至PWM载波谷值) 11: PWM下降缘固定(至PWM载波谷值) 注: 在选择三角载波PWM时有效(<PWMMD>="1")。
13-12	VPWMES[1:0]	R/W	V相边缘设置 00: 边缘不固定(中心对齐PWM) 01: 保留 10: PWM上升缘固定(至PWM载波谷值) 11: PWM下降缘固定(至PWM载波谷值) 注: 在选择三角载波PWM时有效(<PWMMD>="1")。
11-10	UPWMES[1:0]	R/W	U相边缘设置 00: 边缘不固定(中心对齐PWM) 01: 保留 10: PWM上升缘固定(至PWM载波谷值) 11: PWM下降缘固定(至PWM载波谷值) 注: 在选择三角载波PWM时有效(<PWMMD>="1")。
9-8	DSYNCS[1:0]	R/W	占空比较寄存器和PWM周期寄存器的双缓存器更新时序。 00: 取决于中断周期设置(见表14-2) 在已选择0.5 PWM周期的情况下, 在载波峰值和载波谷值时进行更新(<INTPRD>="00")。否则在载波峰值时更新。 01: 在PWM载波谷值时进行更新 10: 在PWM载波峰值时进行更新 11: 在PWM载波峰值和谷值时进行更新 注1: 选择锯齿波载波(<PWMMD>="0")时, 无论设置如何, 均应在载波峰值时更新。 注 2: PMD1MDEN<PWMEN>="0" 时, 无论是设置如何, 均应采用非同步更新。
7	-	R	读作0。
6	PWMCK	R/W	PWM周期扩展模式 0: 正常周期 1: 4 × 周期 设置该PWM计数器的计数周期。 正常周期设置: 锯齿波25 ns/三角波50 ns(在fsys=40 MHz时) 四倍周期设置: 锯齿波100 ns/三角波200 ns(在fsys=40 MHz时)
5	SYNTMD	R/W	端口输出模式 端口输出受控于<nOC>, <nPWM>, <POLH>, <POLL>和<SYNTMD>的组合(见表14-5)。

位	比特符号	类型	功能
4	DTYMD	R/W	占空比模式 0: 三相共用模式: 1: 三相单独模式: 此位可选择是否为各相单独进行任务设置, 或将PMD1CMPU寄存器用作3相共用寄存器。
3	PINT	R/W	PWM中断请求时序 0: 在PWM载波谷值(PMD1MDCNT<MDCNT[15:0]> = 0x0001)时出现中断请求。 1: 在PWM载波峰值(PMD1MDCNT <MDCNT [15:0]>=<MDPRD [15:0]>)时出现中断请求。注1: PWM载波为锯齿波时, 在载波峰值时发生中断请求(<PWMMD>="0")。注 2: 中断周期为0.5 个周期(<INTPRD>="00")时, 可在峰值和载波谷值时出现中断请求。
2-1	INTPRD[1:0]	R/W	PWM中断请求周期 00: 在每0.5个PWM周期时发生中断请求 注1: 仅当PWM载波为三角波时, 才可配置PWM中断请求周期 (<PWMMD>="1") 注 2: (PMD1CMPU/V/W)和周期寄存器(PMD1MDPRD)的双缓存器, 由PWM载波的峰值和谷值进行更新。 01: 每个PWM周期发生一次中断请求 10: 每两个PWM周期发生一次中断请求 11: 每四个PWM周期发生一次中断请求 此字段可从0.5 个PWM周期, 一个PWM周期, 两个PWM周期和PWM个周期中, 选择PWM中断请求周期。
0	PWMMD	R/W	PWM载波波形 0: PWM模式 0 (边缘对齐PWM和锯齿波) 1: PWM模式1 (中心对齐PWM和三角波)

表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMCU1/V1/W1缓存器更新定时

设置		更新时序
<DSYNCS[1:0]>	<INTPRD[1:0]>	
00	1x	在PWM载波峰值时进行更新
	x1	在PWM载波峰值时进行更新
	00	在PWM载波峰值和PWM载波谷值时进行更新
01	xx	在PWM载波谷值时进行更新
10	xx	在PWM载波峰值时进行更新
11	xx	在 载波峰值 和 PWM 时进行更新 载波谷值

x: 忽略

表14-3 PMD1CMPU/V/W和VECMPU1/V1/W1的切换控制

普通设置		各相的设置		寄存器选择信号
<DSYNCS[1:0]>	<INTPRD[1:0]>	<nPWMMES[1]>		
01	xx	x		VE寄存器
10	xx	x		VE寄存器
11	xx	0		在向上计数期间：PMD寄存器 在向下计数期间：VE寄存器
		1		VE寄存器
00	00	0		在向上计数期间：PMD寄存器 在向下计数期间：VE寄存器
		1		VE寄存器

注：在<MSEL0>="1"，<DCMEN>="1"和<PWMMD>="1" 时有效。

x: 忽略

14.3.4.2 PMD1CNTSTA(PWM 计数器状态寄存器)

	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	-	-	-	-	-	-	-	UPDWN
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	UPDWN	R	PWM计数器标志 0: 向上计数 1: 下降计数 该位可指示该PWM计数器是向上计数上升计数还是向下计数下降计数。 注: 该PWM载波为锯齿波(PMD1MDCR<PWMMD>="0"), 且读数为零。

14.3.4.3 PMD1MDCNT(PWM 计数器寄存器)

	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	MDCNT							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	MDCNT							
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	MDCNT[15:0]	R	<p>PWM计数器</p> <p>可从生成PWM载波的该上升-下降计数器读取一个值。计数器分辨率：25 ns(当fsys = 40 MHz时)</p> <p>注1：在已选择四倍循环模式的情况下(PMD1MDCR<PWMCK>="1")，该计数器的时间分辨率为100 ns(在fsys=40 MHz时)。</p> <p>注 2：依据PWM载波(PMD1MDCR<PWMMD>)的设置，PMD被禁用(PMD1MDEN<PWMEN>="0")时的PWM计数器值如下：</p> <p>如果PMD1MDCR<PWMMD>="0":0x0001</p> <p>如果PMD1MDCR<PWMMD>="1":PMD1MDPRD<MDPRD[15:0]>的值</p>

14.3.4.4 PMD1MDPRD(PWM 周期寄存器)

	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	MDPRD							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	MDPRD							
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	MDPRD[15:0]	R/W	<p>PWM 周期</p> <p>$\langle \text{MDPRD}[15:0] \rangle \geq 0x010$</p> <p>利用PWM周期扩展模式(PMD1MDCR<PWMCK>)和PWM载波波形 <PWMMMD>的组合, PWM周期可计算如下:</p> <p>在 <PWMCK>="0", <PWMMMD>="0" : $\langle \text{MDPRD} \rangle \times 1/\text{fsys}$</p> <p><PWMMMD>="1" : $\langle \text{MDPRD} \rangle \times 2/\text{fsys}$</p> <p>在<PWMCK>="1", <PWMMMD>="0" : $\langle \text{MDPRD} \rangle \times 4/\text{fsys}$</p> <p><PWMMMD>="1" : $\langle \text{MDPRD} \rangle \times 8/\text{fsys}$</p> <p>注: 如果<MDPRD[15: 0]>设置为某个小于 0x0010 的值, 它会自动假定为 0x0010(该寄存器保留所写入的真值)。</p>

注1: 不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的, 则无法保证运算。

注2: 由于PMD1MDPRD寄存器为双缓存型, 因此, 在PWM计数器工作期间可改变周期。

注3: 有关后级缓存器的详细更新时序, 见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMCU1/V1/W1缓存器更新时序"。

注4: 所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.4.5 PMD1CMPU(U相PWM比较寄存器)

	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	CMPU							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMPU							
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	CMPU[15:0]	R/W	<p>U相的PWM脉冲宽度</p> <p>0x0000 ~ 0xFFFF</p> <p>注：在<CMPU> > <MDPRD>时，占空为100%。</p> <p>比较寄存器<CMPU[15:0]>用于确定U相的输出脉冲宽度。这些寄存器属于双缓存型。通过比较后级缓存器和PWM计数器的大小，来确定脉冲宽度。</p> <p>利用PWM周期扩展模式(PMD1MDCR)和PWM载波波形(<PWMMMD>)的组合，脉冲宽度可计算如下：</p> <p>When <PWMCK>="0", <PWMMMD>="0" : <CMPU> × 1/fsys</p> <p><PWMMMD>="1" : <CMPU> × 2/fsys</p> <p>When <PWMCK>="1", <PWMMMD>="0" : <CMPU> × 4/fsys</p> <p><PWMMMD>="1" : <CMPU> × 8/fsys</p>

注1：在将经过总线更新的比较寄存器中的值加载到后级缓存器时，可通过将PMD1MODESEL<MDSEL0>设置为"0"，选择总线模式(默认值)。

注2：不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的，则无法保证运算。

注3：由于PMD1CMPU寄存器属于双缓存型，因此，在PWM计数器运行期间可改变周期。

注4：有关后级缓存器的详细更新时序，见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMCU1/V1/W1缓存器更新时序"。

注5：所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.4.6 PMD1CMPV(V相PWM比较寄存器)

	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	CMPV							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMPV							
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	CMPV[15:0]	R/W	<p>V相的PWM脉冲宽度</p> <p>0x0000 ~ 0xFFFF</p> <p>注: <CMPV>><MDPRD>时, 占空比为100%。</p> <p>比较寄存器<CMPV[15:0]>用于确定V相的输出脉冲宽度。这些寄存器属于双缓存型。通过比较后级缓存器和PWM计数器的大小, 来确定脉冲宽度。</p> <p>利用PWM周期扩展模式(PMD1MDCR)和PWM载波波形(<PWMMMD>)的组合, 脉冲宽度可计算如下:</p> <p>When <PWMCK>="0", <PWMMMD>="0" : <CMPV> × 1/fsys</p> <p><PWMMMD>="1" : <CMPV> × 2/fsys</p> <p>When <PWMCK>="1", <PWMMMD>="0" : <CMPV> × 4/fsys</p> <p><PWMMMD>="1" : <CMPV> × 8/fsys</p>

注1: 在将经过总线更新的比较寄存器中的值加载到后级缓存器时, 可通过将PMD1MODESEL<MDSEL0>设置为"0", 选择总线模式(默认值)。

注2: 不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的, 则无法保证运算。

注3: 由于PMD1CMPV寄存器属于双缓存型, 因此, 可在PWM计数器运行期间改变该周期。

注4: 有关后级缓存器的详细更新时序, 见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMCU1/V/W1缓存器更新定时"。

注5: 所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.4.7 PMD1CMPW(W相PWM比较寄存器)

	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	CMPW							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMPW							
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	CMPW[15:0]	R/W	<p>W相的PWM脉冲宽度</p> <p>0x0000 ~ 0xFFFF</p> <p>注: <CMPW> > <MDPRD>时, 占空比为100%。</p> <p>比较寄存器<CMPW[15:0]>用于确定W相的输出脉冲宽度。这些寄存器属于双缓存型。通过比较后级缓存器和PWM计数器的大小, 来确定脉冲宽度。</p> <p>利用PWM周期扩展模式(PMD1MDCR)和PWM载波波形(<PWMMMD>)的组合, 脉冲宽度可计算如下:</p> <p>When <PWMCK>="0", <PWMMMD>="0" : <CMPW> x 1/fsys</p> <p><PWMMMD>="1" : <CMPW> x 2/fsys</p> <p>When <PWMCK>="1", <PWMMMD>="0" : <CMPW> x 4/fsys</p> <p><PWMMMD>="1" : <CMPW> x 8/fsys</p>

注1: 在将经过总线更新的比较寄存器中的值加载到后级缓存器时, 可通过将PMD1MODESEL<MDSEL0>设置为"0", 选择总线模式(默认值)。

注2: 不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的, 则无法保证运算。

注3: 由于PMD1CMPW寄存器属于双缓存型, 因此, 可在PWM计数器运行期间改变该周期。

注4: 有关后级缓存器的详细更新时序, 见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMCU1/V/W1缓存器更新定时"。

注5: 所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.5 导通控制电路

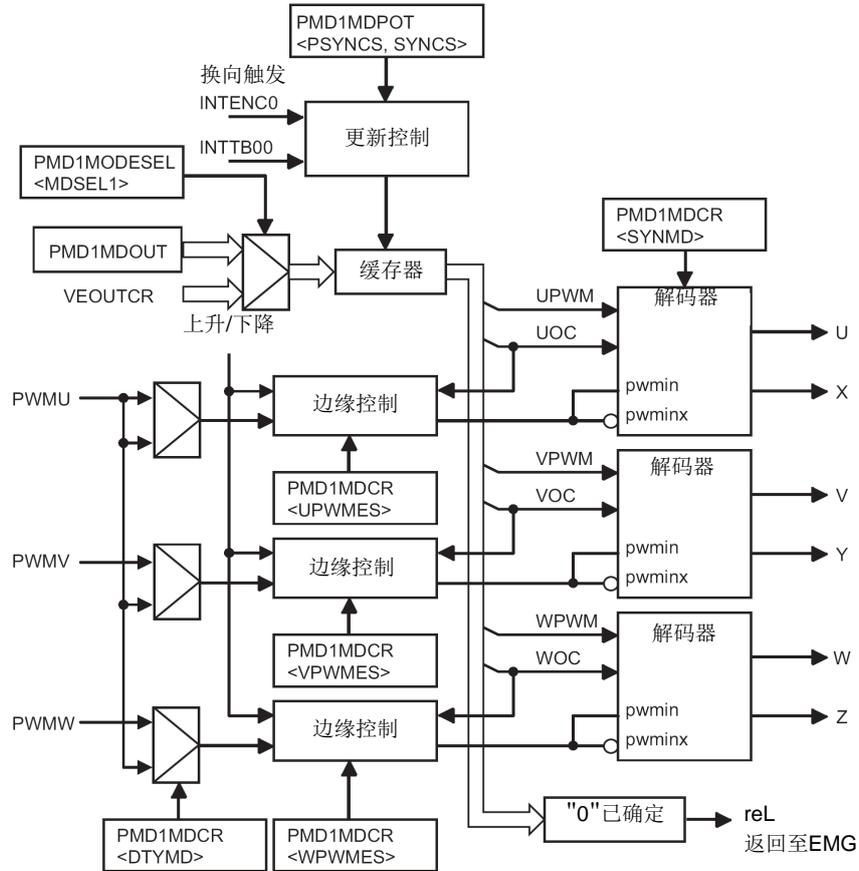


图14-7 导通控制电路

导通控制电路可按照在输出控制寄存器PMD1MDOUT(VEOUTCR)和输出设置寄存器PMD1MDPOT中的所做的设置执行输出端口控制。PMD1MDOUT (VEOUTCR)寄存器为双缓存型，且其更新定时可被设置为与PWM同步或非同步。也可选择更新定时与触发脉冲输入同步(有关更新定时的详细数据，见"表14-4 PMD1MDOUT(VEOUTCR)缓存器的更新定时")。

利用PMD1MDPOT<POLH>和<POLL>，可将六个输出端口分别设置为上相输出(UO, VO, WO)或下相输出(XO, YO, ZO)部位的低活动状态或高活动状态。此外，PMD1MDOUT(VEOUTCR)寄存器的<WPWM>，<VPWM>和<UPWM>还可为U, V和W相选择PWM或高/低输出。在已选择PWM输出的情况下，可输出PWM波形。在已选择高/低输出时，输出即被固定为高电平或低电平。通过寄存器PMD1MDOUT(VEOUTCR)的<WOC>，<VOC>和<UOC>，各输出被设置为高或低。

"按<UOC>，<VOC>，<WOC>，<UPWM>，<VPWM>和<WPWM>设置进行的端口输出表14-5"显示按照PMD1MDOUT(VEOUTCR)寄存器和PMD1MDPOT寄存器中的端口输出设置，以及PMD1MDCR寄存器的端口输出模式下的端口输出极性设置，给出了端口输出设置。

表14-4 PMD1MDOUT(VEOUTCR)缓存器的更新时序

		PSYNCS设置			
		00	01	10	11
SYNCS 设置	00	常量更新	PWM载波谷值	PWM载波峰值	PWM载波峰值和 PWM载波谷值
	01	在INTENC出现时。	INTENC每次出现时的 第一个PWM载波谷值	INTENC每次出现时的 第一个PWM载波峰值	INTENC每次出现时的 第一个PWM载波峰值 或第一个载波谷值。
	10	在INTTB00出现时。	INTTB00每次出现时 的第一个PWM载波谷 值	INTTB00每次出现时 的首个PWM载波峰值。	INTTB00每次出现时 的第一个PWM载波峰 值或第一个载波谷值。
	11	-	-	-	-

注：当PMD被禁用(PMD1MDCR<PMWEN>="0")时，所保留的触发状态即被清除。

14.3.5.1 PMD1MDPOT(PMD 输出设置寄存器)

	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	-	-	-	-	-	-	SYNCS	
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	POLH	POLL	PSYNCS	
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-10	-	R	读作0。
9-8	SYNCS[1:0]	R/W	选择PMD1MDOUT(VEOUTCR)传输时序(触发脉冲同步设置)。 00: 非同步 01: 在发生INTENC(ENC中断请求)时 10: 在发生INTTB00(TMRB中断请求)时 11: 保留 选择导通控制寄存器的后级缓存器更新时序。 注1: 通过<PSYNC>和 <SYNCS>的设置组合, 可确定缓存器更新时序(见"表14-4 PMD1MDOUT(VEOUTCR)缓存器的更新时序") 注 2: 在PMD被禁用时(PMD1MDEN<PWMEN>="0"), 无论设置如何, 时序为非同步型。
7-4	-	R	读作0。
3	POLH	R/W	选择上相输出(UO, VO, WO)的输出极性。 0: 低位激活 1: 高位激活
2	POLL	R/W	选择下相输出(XO, YO, ZO)的输出端极性。 0: 低位激活 1: 高位激活
1-0	PSYNCS[1:0]	R/W	选择PMD1MDOUT(VEOUTCR)传输时序(PWM同步设置)。 00: 与PWM非同步 该设置被同时应用于端口输出, 以及PMD1MDOUT/VEOUTCR寄存器。 01: 载波谷值 (在<MDCNT[15:0]>="1"时) 10: 载波峰值 (<MDCNT[15:0]>=<MDPRD[15:0]>) 11: 载波峰值和载波谷值 选择导通控制寄存器的后级缓存器更新时序。 注1: 在PWM载波为锯齿波时, 缓存器更新新定时是载波峰值, 但<PSYNCS>="00"除外。 注 2: 通过<PSYNC>和 <SYNCS>的设置组合, 可确定缓存器更新时序(见"表14-4 PMD1MDOUT(VEOUTCR)缓存器的更新时序") 注 3: PMD被禁用时(PMD1MDEN<PWMEN>="0"), 无论设置如何, 时序为非同步型。

注: 在PMD1MDEN<PWMEN>="0", 必须设置此字段。

14.3.5.2 PMD1MDOUT(PMD 导电控制寄存器)

	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	-	-	-	-	-	WPWM	VPWM	UPWM
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	WOC		VOC		UOC	
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-11	-	R	读作0。
10	WPWM	R/W	W相PWM输出设置 0: 高/低输出 1: PWM输出 端口输出受<WOC>, <WPWM>, <POLH>, <POLL>和<SYNTMD>的组和控制 (见表14-5)。
9	VPWM	R/W	V相PWM输出设置 0: 高/低输出 1: PWM输出 端口输出受<VOC>, <VPWM>, <POLH>, <POLL>和<SYNTMD>的组和控制 (见表14-5)。
8	UPWM	R/W	U相PWM输出设置 0: 高/低输出 1: PWM输出 端口输出受<UOC>, <UPWM>, <POLH>, <POLL>和<SYNTMD>的组和控制 (见表14-5)。
7-6	-	R	读作0。
5-4	WOC[1:0]	R/W	W相导通控制设置 端口输出受<WOC>, <WPWM>, <POLH>, <POLL>和<SYNTMD>的组和控制 (见表14-5)。
3-2	VOC[1:0]	R/W	V相导通控制设置 端口输出受<VOC>, <VPWM>, <POLH>, <POLL>和<SYNTMD>的组和控制 (见表14-5)。
1-0	UOC[1:0]	R/W	U相导通控制设置 端口输出受<UOC>, <UPWM>, <POLH>, <POLL>和<SYNTMD>的组和控制 (见表14-5)。

注1: 向后续阶段缓存器加载PMD1MDOUT(VEOUTCR)寄存器中通过总线更新的值时, 通过将PMD1MODESEL<MDESEL0>设置为"0", 即可选择总线模式(缺省)。

注2: 不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的, 则无法保证运算。

注3: 由于导通控制寄存器属于双缓冲型, 因此, 在PWM计数器运行期间可改变周期。

注4: 有关后续阶段缓存器的详细数据定时, 见"表14-4 PMD1MDOUT(VEOUTCR)缓存器的更新定时"。

注5: 所读取的值是第一个缓存值(通过总线获得的最新数据集)。

表14-5 按<UOC>, <VOC>, <WOC>, <UPWM>, <VPWM>和<WPWM>设置进行的端口输出

PMD1MDCR<SYNTMD>="0"

极性: 高位激活(PMD1MDPOT<POLH><POLL>="11")

PMD1MDOUT导通控制		<WPWM><VPWM><UPWM>			
(上相)	(下相)	PWM输出设置			
<WOC[1]>	<WOC[0]>	0:H/L输出		1:PWM输出	
		上相输出	下相输出	上相输出	下相输出
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

PMD1MDCR<SYNTMD>=0

极性: 低位激活(PMD1MDPOT<POLH><POLL>="00")

PMD1MDOUT导通控制		<WPWM><VPWM><UPWM>			
(上相)	(下相)	PWM输出设置			
<WOC[1]>	<WOC[0]>	0:H/L输出		1:PWM输出	
		上相输出	下相输出	上相输出	下相输出
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	$\overline{\text{PWM}}$
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

PMD1MDCR<SYNTMD>=1

极性: 高位激活(PMD1MDPOT<POLH><POLL>="11")

PMD1MDOUT导通控制		<WPWM><VPWM><UPWM>			
(上相)	(下相)	PWM输出设置			
<WOC[1]>	<WOC[0]>	0:H/L输出		1:PWM输出	
		上相输出	下相输出	上相输出	下相输出
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	$\overline{\text{PWM}}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

PMD1MDCR<SYNTMD>=1

极性: 低位激活(PMD1MDPOT<POLH><POLL>="00")

PMD1MDOUT导通控制		<WPWM><VPWM><UPWM>			
(上相)	(下相)	PWM输出设置			
<WOC[1]>	<WOC[0]>	0:H/L输出		1:PWM输出	
		上相输出	下相输出	上相输出	下相输出
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

14.3.6 保护控制电路

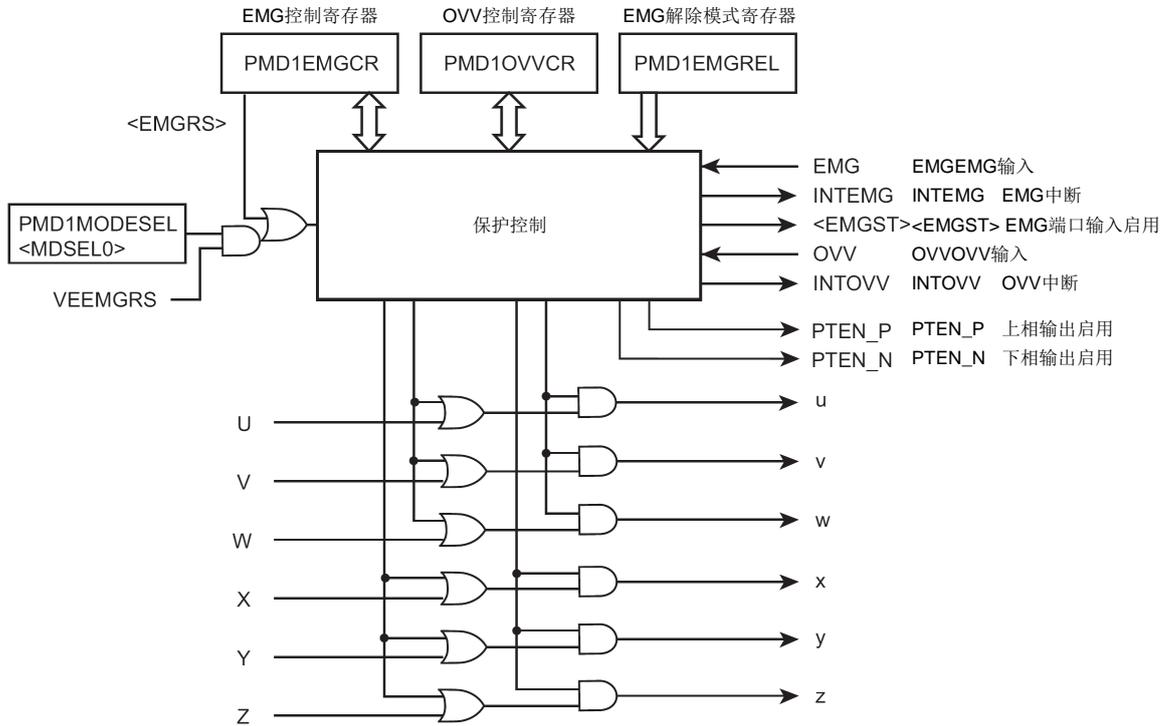


图14-8 保护控制电路

保护控制电路由一个EMG保护控制电路和一个OVV保护控制电路组成。

14.3.6.1 EMG 保护电路

EMG保护电路由一个EMG保护控制部件和一个端口输出禁用部件构成。EMG输入变低时，电路即被启动。

EMG保护电路可提供一个紧急停车机构：在EMG输入被断言为(H→L)时，所有六个端口的输出都会被立即禁用(视PMD1EMGCR<EMGMD>设置而定)，并生成一次EMG中断(INTEMG)。

如发生紧急状况，可对<EMGMD>进行设置，使之输出一个控制信号，并通过该信号将外部输出端口设置为高-z。

工具断开根据PMD1PORTMD<PORTMD>设置也可禁用所有六根PWM输出线。在发生工具断开时，可通过设置PMD1PORTMD<PORTMD>寄存器，将各外部输出端口设置为高-z。

EMGSTA<EMGST>中的已读取的值"1"表示该EMG保护电路处于活动状态。

可通过EMG控制寄存器(PMD1EMGCR)设置EMG保护。

在EMG保护状态下，也可通过将所有端口输出线设置为不活动而实现其解除(将"0"设置为PMD1MDOUT(VEOUTCR)<UPWM>，<VPWM>，<WPWM>，<UOC>，<VOC>，<WOC>)(注1)此时，可将PMD1EMGCR<EMGRS>或VEEMGRS<EMGRS>设置为"1"。如需禁用该EMG保护功能，可按该顺序将"0x5A"和"0xA5"写入到PMD1EMGREL寄存器，然后将PMD1EMGCR<EMGEN>清除为"0"(必须连续执行这三条指令)。在EMG保护输入为低期间，任何试图解除EMG保护状态的努力均会被忽略。确认PMD1EMGSTA<EMGI>的状态标志为"1"后，可解除EMG保护状态。

只有在所指定关键代码("0x5A"，"0xA5")被写入到PMD1EMGREL寄存器中，以防止其无意中被禁用之后，才可禁用该EMG保护电路。

注1: PMD1MDOUT(VEOUTCR)的数据需在后续阶段缓存器中反映出来。

注2: EMG功能的初始过程

在复位之后，EMG功能即被启用，但EMG引脚配置与正常端口相同。因此，在EMG保护可能有效时，可在初始序列时按以下程序解除EMG保护。

- 1: 通过PxFR寄存器选择EMG功能。
- 2: 读取PMD1EMGSTA<EMGI>并确认其为"1"。
- 3: 将PMD1MDOUT(VEOUTCR)<UPWM>，<VPWM>，<WPWM>，<UOC>，<VOC>与<WOC>设置为"0"，使所有端口进入不活动状态("L"输出)。
- 4: 通过将PMD1EMGCR(VEEMGRS)<EMGRS>设置为"1"，解除EMG保护。

如拟禁用EMG保护，则继续以下程序。

- 5: 将各关键代码写入到PMD1EMGREL(按"0x5A"和"0xA5"的顺序)。
- 6: 将PMD1EMGCR<EMGEN>设置为"0"，即可禁用EMG保护。

14.3.6.2 PMD1EMGREL(EMG 解除寄存器)

	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	EMGREL							
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	EMGREL[7:0]	W	EMG/OVV禁用代码 通过按该顺序将0x5A和0xA5设置为寄存器，即可禁用EMG和OVV保护功能。在写入禁用代码之后，立即设置PMD1EMGCR<EMGEN>="0"或PMD1OVVCR<OVVEN>="0"。 在禁用这些功能时，必须将<EMGEN>和<OVVEN>清0。

注：在禁用EMG和OVV，各需写入一个禁用代码。

14.3.6.3 PMD1EMGCR(EMG 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	-	-	-	-	EMGCNT			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	INHEN	EMGMD		-	EMGRS	EMGEN
复位后	0	0	1	1	1	0	0	1

位	比特符号	类型	功能
31-12	-	R	读作0。
11-8	EMGCNT[3:0]	R/W	EMG输入检测时间 0x0至0xF (在<EMGCNT[3:0]>="0"时, 噪声滤波器被旁路)。 将异态检测输入的静噪滤波长度设置为这些位上。并按以下公式计算该值。 <EMGCNT[3:0]> x16/fsys (分辨率:400[nsec] 在40 MHz时)
7-6	-	R	读作0。
5	INHEN	R/W	工具断开启用/禁用 0: 禁用 1: 启用 在PMD停止信号从工具输入时, 该位可选择是否停止该PMD。 注: 工具断开在初始状态即已启用。
4-3	EMGMD[1:0]	R/W	EMG保护模式选择 00: 所有相位高-Z 01: 所有上相开/所有下相高-Z 10: 所有上相高-Z/所有下相开 11: 所有相位高-Z 在发生EMG时, 可将该端口输出设置为上(UO, VO, WO)和下(XO, YO, ZO)。 注: "开"表示PWM输出继续。
2	-	R/W	始终写入"0"。
1	EMGRS	W	EMG保护解除 0: - 1: 解除保护 通过将PMD1MDOUT寄存器设置为 "0x000", 然后将 <EMGRS> 位设置为 "1", 即可解除EMG保护。 注: 该位始终读作0。 注: 如果PMD1MDOUT(VEOUTCR)的后续阶段缓存器未被更新至"0x000", 则无法解除EMG保护。 注: 在解除EMG保护之前, 确认PMD1EMGSTA <EMGI>已重新变回"1"。
0	EMGEN	R/W	EMG保护电路启用/禁用 0: 禁用 1: 启用 通过将"0x5A"和"0xA5"先后写入到EMG解除寄存器(PMD1EM GREL), 即可禁用该功能。然后, 将"0"设置为<EMGEN>(必须按顺序执行这三条指令)。 注: EMG保护电路是在初始状态启用的。

14.3.6.4 PMD1EMGSTA(EMG 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	-	-	-	-	-	-	EMGI	EMGST
复位后	0	0	0	0	0	0	未定义	0

位	比特符号	类型	功能
31-2	-	R	读作0。
1	EMGI	R	EMG输入 EMG保护状态 通过读取此位可辨别EMG输入状态
0	EMGST	R	EMG保护状态 0: 正常操作 1: 保护 通过读取此位, 即可辨别EMG保护状态。

14.3.6.5 OVV 保护控制电路(OVV 输入字块)

OVV保护控制电路由一个OVV保护控制单元一个端口输出禁用单元组成。此电路在OVV输入端口断言时激活。

在OVV输入信号(H→L)已接受过一个规定周期的断言(设置为PMD1OVVCR <OVVCNT>)的情况下, OVV保护电路可将该导通控制电路中的六根端口输出线固定在"高"或"低"位。此时, 即可生成一次OVV中断(INTOVV)。可仅选择上相, 下相或所有的相位。

可通过OVV控制寄存器的PMD1OVVCR设置OVV保护。PMD1OVVSTA<OVVST>中的读取值"1"表示OVV保护电路处于激活状态。

通过将PMD1OVVCR<OVVRS>设置为"1", 即可启用OVV保护状态的解除操作。保护输入被取消后, OVV保护即会在预设的定时自动解除(在OVV保护输入为低期间, OVV保护状态不会被解除。通过读取PMD1OVVSTA<OVVI>, 可检查端口输入的状态)。

OVV保护状态可与PWM周期同步解除(在PWM计数PMD1MDCNT可匹配PMD1MDPRD的时序点)。然而, 如果已设置了基于半个PWM周期的中断频次, 则一旦PWM计数为"1"或可匹配PMD1MDPRD时, 保护状态即被解除。禁用OVV保护功能时, 可按顺序将"0x5A"和"0xA5"写入到EMG解除寄存器的PMD1EMGREL, 然后将PMD1OVVCR<OVVEN>清0(必须按顺序执行这三条指令)(必须连续执行这三条指令)。

只有在所指定关键代码("0x5A", "0xA5")被写入到PMD1EMGREL寄存器中, 以防止其无意中被禁用之后, 才可禁用OVV保护电路。

14.3.6.6 PMD1OVVCR(OVV 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	-	-	-	-	OVVCNT			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-12	-	R	读作0。
11-8	OVVCNT[3:0]	R/W	<p>OVV输入检测时间</p> <p>值: 0x1~ 0xF(如果已设置"0", 则其可作为"1"处理。)</p> <p>将OVV输入集的静噪滤波长度设置为这些位上。且可按以下公式计算该值。</p> <p>$\langle \text{OVVCNT}[3:0] \rangle \times 16/\text{fsys}$ (分辨率: 400[nsec] 在40 MHz时)</p> <p>注: $\langle \text{OVVCNT}[3:0] \rangle$仅当端口输入被选中 ($\langle \text{OVVISEL} \rangle = "0"$)时有效。</p>
7	-	R	读作0。
6	ADIN1EN	R/W	<p>ADC监视功能1 输入启用</p> <p>0: 禁用输入</p> <p>1: 启用输入</p> <p>从AD转换器的ADC监控功能1选择各启用/禁用信号。如启用之, 且选择输入的ADC监控信号 ($\langle \text{OVVISEL} \rangle = "1"$), 则ADC监控功能1的结果即为OVV输入(如果OVV保护处于已启用状态)。</p> <p>注: 有关AD转换监控功能的详细信息, 见"12位模拟/数字转换器"中的"AD转换监控功能"。</p>
5	ADIN0EN	R/W	<p>ADC监视功能 0 输入启用</p> <p>0: 禁用输入</p> <p>1: 启用输入</p> <p>从ADC的ADC监控功能0选择启用/禁用信号。如启用之, 且选择输入的ADC监控信号 ($\langle \text{OVVISEL} \rangle = "1"$), 则ADC监控功能0的结果即为OVV输入(如果OVV保护处于已启用状态)。</p> <p>注: 有关AD转换监控功能的详细信息, 见"12位模拟/数字转换器"中的"AD转换监控功能"。</p>
4-3	OVVMD[1:0]	R/W	<p>选择OVV保护模式</p> <p>00: 无输出控制</p> <p>01: 所有上相开启, 所有下相关闭</p> <p>10: 所有上相关闭, 所有下相开启</p> <p>11: 所有相均关闭</p> <p>在出现OVV条件时, 该字段控制上部各相(UO, VO, WO)和下部各相(XO, YO, ZO)的输出</p> <p>注: "ON"表示其已被固定为活动输出。"OFF"表示其被固定为不活动输出。"活动"和"不活动"则取决于 $\langle \text{POLL} \rangle$ 和 $\langle \text{POLH} \rangle$ 的设置。</p> <p>注: 如果OVV和EMG条件同时出现, 则各$\langle \text{EMGMD}[1:0] \rangle$位中的保护模式设置生效。</p>
2	OVVISEL	R/W	<p>选择OVV输入</p> <p>0: 端口输入</p> <p>1: ADC监视信号</p> <p>该位会选择是否将来自ADC的端口输入或监视信号作为拟输入到该保护电路中的OVV信号。</p> <p>注: 在ADC监视信号被选中时, 输入检测时间$\langle \text{OVVCNT}[3:0] \rangle$的设置失效(直接输入)。</p>
1	OVVRS	R/W	<p>选择OVV保护状态解除</p> <p>0: 禁用OVV保护状态的自动解除</p> <p>1: 启用OVV保护状态的自动解除</p> <p>注: 如果已启用OVV保护的自动解除, 则在检测到异常后OVV保护的状态发生变化时(OVV输入引发一次从高到低的过渡, 在OVV输入过渡到"高"之后更新PWM周期寄存器PMD1MDPRD的缓存器时, OVV保护状态会被自动解除(见"表14-2 PMD1MDPRD, PMD1CMPU/V/W和VECMCU1/V1/W1缓存器更新定时")。</p>
0	OVVEN	R/W	<p>OVV保护电路启用/禁用</p> <p>0: 禁用</p> <p>1: 启用</p> <p>注: 如需禁用该功能, 可将"0x5A"和"0xA5"先后写入到EMG 解除寄存器(PMD1EM- GREL)。然后将 "0" 设置为$\langle \text{OWEN} \rangle$。(必须连续执行这三条指令)。</p>

14.3.6.7 PMD1OVVSTA(OVV 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	-	-	-	-	-	-	OVVI	OVVST
复位后	0	0	0	0	0	0	未定义	0

位	比特符号	类型	功能
31-2	-	R	读作0。
1	OVVI	R	OVVI输入 OVVI状态 通过读取该位，即可辨别OVV输入状态(通过PMD1OVVCR<OVVISEL>选取)。
0	OVVST	R	OVV保护状态 0: 正常操作 1: 保护中 通过读取此位，即可辨别OVV保护状态。

14.3.7 停滞时间电路

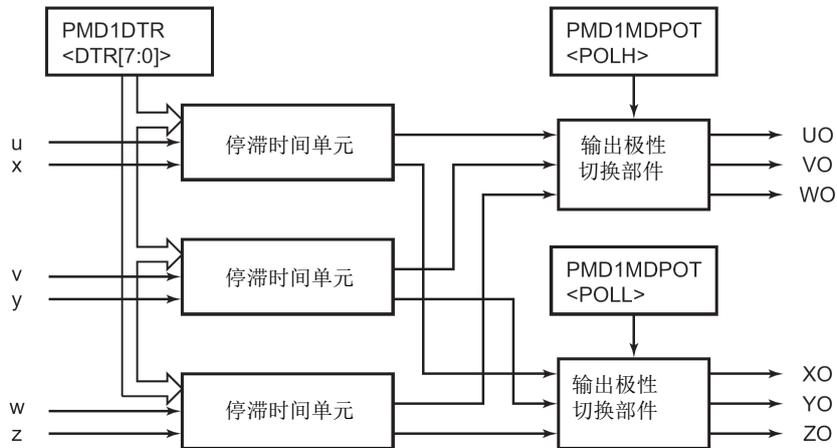


图14-9 停滞时间电路

停滞时间电路由一个停滞时间部件和一个输出极性切换部件构成。

对于各U相，V相和W相而言，在各上相和下相被切换以防止短路时，停滞时间单元可延迟各相的ON定时。可将停滞时间设置为停滞时间寄存器(PMD1DTR<DTR[7:0]>)，此时可设置一个40 MHz时的分辨率为200 [ns]的8 位值。

输出极性切换电路允许通过PMD输出PMD1MDPOT<POLH>和<POLL> 输出设置寄存器，单独设置上输出相(UO, VO, WO)和下输出相(XO, YO, ZO)(高位激活低位激活)的极性。

14.3.7.1 PMD1DTR(停滞时间寄存器)

	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	DTR							
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	DTR[7:0]	R/W	设置停滞时间 0x00~0xFF 通过以下的公式，可计算该停滞时间值。 $200 \text{ nsec} \times \langle \text{DTR}[7:0] \rangle$ (直至51 μsec (在fsys = 40 MHz时))

14.3.8 Sync 触发生成电路

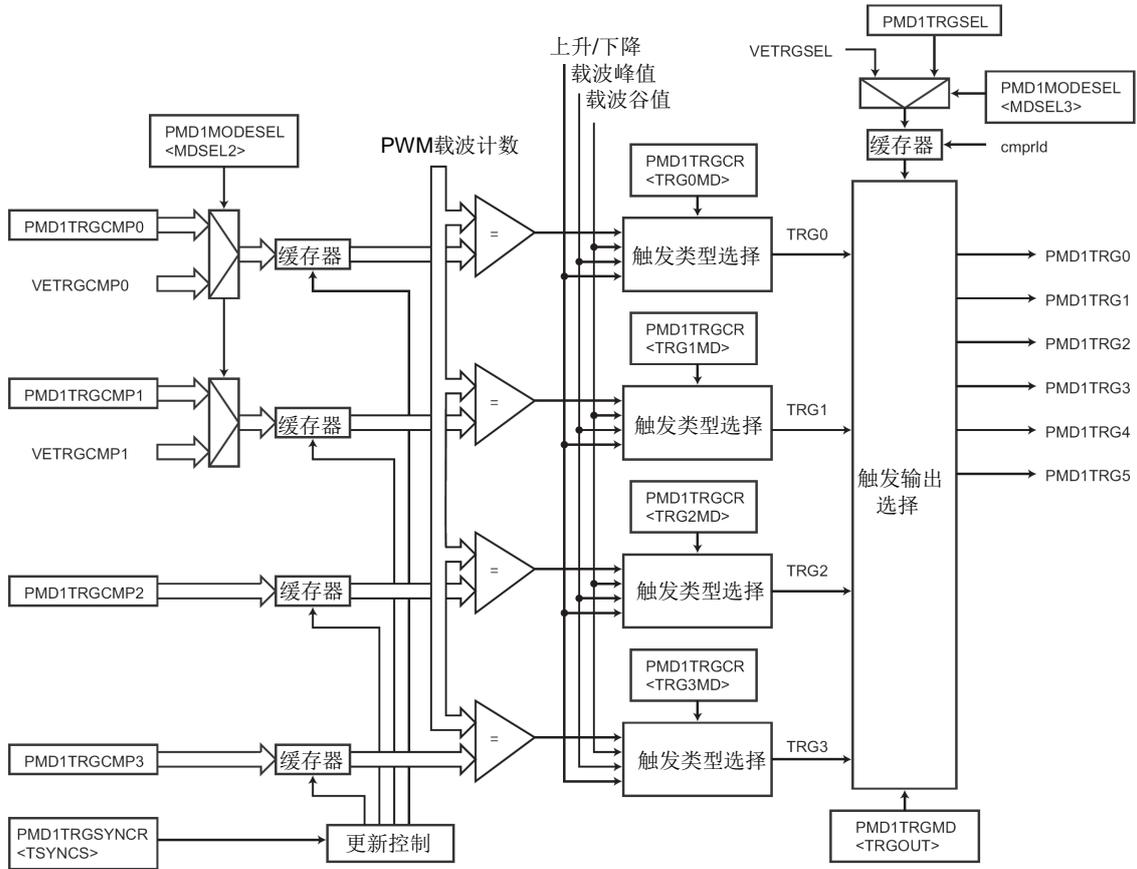


图14-10 Sync触发生成电路

同步触发生成电路可生成四个触发信号(TRG0至TRG3)，用于与PWM同步进行ADC采样。当通过PMD1MODESEL<MDESEL3>选择VE 模式上，PMD1TRGCMP0和PMD1TRGCMP1变为VE寄存器的VETRGCMP0和VETRGCMP1。

可选择以下六种类型的触发时序。

1. 在向上计数运行比较匹配时(注)
2. 在向下计数运行比较匹配时(注)
3. 在向下/向下计数运行比较匹配时(注)
4. PWM载波峰值
5. PWM载波谷值
6. PWM载波峰值和PWM载波谷值

注：比较匹配发生在PWM计数器 (PMD1MDCNT<MDCNT[15:0]>)和(PMD1TRGCMPn <TRGCMPn[15:0]>)之间。

在触发选择输出模式期间：PMD1TRGMD<TRGOUT>="1"。TRG0信号触发电路，是从通过触发电路选择寄存器PMD1TRGSEL(VETRGSSEL)选择的PMD1TRG0~5输出的。可通过PMD1TRGCMP0(VETRGCMP01)和PMD1TRGCR<TRG0MD>，来设置TRG0的设置。选择边缘模式(锯齿波载波模式)上，"比较-匹配"功能为上升计数。PMD1TRGMD<EMGTGE>="1"时，此电路也可在保护状态输出触发信号。

14.3.8.1 PMD1TRGCMP0 (触发比较寄存器 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	TRGCMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP0							
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	TRGCMP0 [15:0]	R/W	触发输出比较寄存器 应在1 ~ [<MDPRD[15:0]>设置值 -1]的范围内设置<TRGCMP0[15:0]>。 在PWM计数器值<MDCNT[15:0]> 可匹配TRGCMP0中的设置值时, 输出TRG0。 注: 禁止将<TRGCMP0>设置为 "0" 和 <TRGCMP0> ≥ <MDPRD[15:0]>值。

注1: 在将各比较寄存器中的数据加载到后续阶段缓存器中时, 可通过将PMD1MODESEL<MSEL2>设置为 "0", 来选择总线模式(缺省)。

注2: 不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的, 则无法保证运算。

注3: 该触发比较寄存器属于双缓存型, 因此可在PWM计数器运行期间改变该周期。

注4: 有关该后续阶段缓存器的详细更新定时, 见"表14-6 触发比较寄存器的缓存器更新时序"。

注5: 所读取的值是第一个缓存值(通过总线获得的最新数据集)。

表14-6 触发比较寄存器的缓存器更新时序

<TSYNCS>设置	<TRGnMD>设置	TRGCMPn寄存器 缓存器更新时序
00	000	立即更新
	001	在PWM载波峰值时更新
	010	在PWM载波谷值时更新
	011	在PWM载波峰值或PWM载波谷值时更新(注1)
	1xx	立即更新
01	xxx	在PWM载波谷值时更新
10	xxx	在PWM载波峰值时更新
11	xxx	在PWM载波峰值或PWM载波谷值时更新(注1)

注: x: 忽略

注: 无论设置如何, 非同步更新PMD1MDEN<PWMEN>="0"。

注1: 选择锯齿波载波时, 在载波峰值时进行更新(PMD1MDCR<PWMMD>="0")。

14.3.8.2 PMD1TRGCMP1 (触发比较寄存器 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	TRGCMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP1							
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	TRGCMP1 [15:0]	R/W	触发输出比较寄存器 应在1 ~[<MDPRD[15:0]>设置值 -1]的范围内设置<TRGCMP1[15:0]>。 在PWM计数器值<MDCNT[15:0]> 可匹配TRGCMP1中的设置值时，可输出TRG1。 注：禁止将<TRGCMP1>设置为 "0" 和<TRGCMP1> ≥ <MDPRD[15:0]>值。

注1：在将各比较寄存器中的数据加载到后续阶段缓存器中时，可通过将PMD1MODESEL<MSEL2>设置为 "0"，来选择总线模式(缺省)。

注2：不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的，则无法保证运算。

注3：该触发比较寄存器属于双缓存型，因此可在PWM计数器运行期间改变该周期。

注4：有关该后续阶段缓存器的详细更新定时，见"表14-6 触发比较寄存器的缓存器更新时序"。

注5：所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.8.3 PMD1TRGCMP2 (触发比较寄存器 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	TRGCMP2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP2							
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	TRGCMP2 [15:0]	R/W	触发输出比较寄存器 应在1 ~[<MDPRD[15:0]>设置值 -1]的范围内设置<TRGCMP2[15:0]>。 在PWM计数器值<MDCNT[15:0]> 可匹配TRGCMP2中的设置值时，可输出TRG2。 注：禁止将<TRGCMP2>设置为 "0" 和<TRGCMP2> ≥ <MDPRD[15:0]>值。

注1：不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的，则无法保证运算。

注2：该触发比较寄存器属于双缓存型，因此可在PWM计数器运行期间改变该周期。

注3：有关该后续阶段缓存器的详细更新定时，见"表14-6 触发比较寄存器的缓存器更新时序"。

注4：所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.8.4 PMD1TRGCMP3 (触发比较寄存器 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	TRGCMP3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP3							
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	TRGCMP3 [15:0]	R/W	触发输出比较寄存器 应在1 ~ [$\text{MDPRD}[15:0]$]设置值 -1]的范围内设置 $\text{TRGCMP3}[15:0]$ 。 在PWM计数器值 $\text{MDCNT}[15:0]$ 可匹配TRGCMP3中的设置值时，可输出TRG3。 注：禁止将 TRGCMP3 设置为"0"和 $\text{TRGCMP3} \geq \text{MDPRD}[15:0]$ 值。

注1：不要以字节为单元写入寄存器。如果上 8 位[15:8]和下 8 位[7:0]是单独写入的，则无法保证运算。

注2：该触发比较寄存器属于双缓存型，因此可在PWM计数器运行期间改变该周期。

注3：有关该后续阶段缓存器的详细更新时序，见"表14-6 触发比较寄存器的缓存器更新时序"。

注4：所读取的值是第一个缓存值(通过总线获得的最新数据集)。

14.3.8.5 PMD1TRGCR(触发控制寄存器)

	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	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
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	TRG3BE	R/W	<p>PMD1TRGCMP3<TRGCMP3[15:0]>缓存器的非同步更新 此位可从PMD1TRGCMP3启用后续阶段缓存器的非同步更新。 0: 同步更新 1: 非同步更新(可立即反映已写入PMD1TRGCMP3的值)。</p> <p>注: 有关详细更新时序, 见"表14-6 触发比较寄存器的缓存器更新时序"。 注: PMD1MDEN<PWMDEN>="0" 时, 无论是设置如何, 均采用非同步更新。</p>
14-12	TRG3MD[2:0]	R/W	<p>PMD1TRGCMP3模式设置 该寄存器选择触发电路的匹配模式。 000: 触发输出已被禁用 001: 向下计数下降计数匹配时的触发输出 010: 向上计数上升计数匹配时的触发输出 011: 上升-/下降计数匹配时的触发输出 100: PWM载波峰值时的触发输出 101: PWM载波谷值时的触发输出 110: PWM载波峰值/谷值时的触发输出 111: 触发输出已被禁用</p> <p>注: 在 "0" 被设置为MD1MDCR<PWMDMD> (锯齿波)时, 即使 "001" 被设置为<TRG3MD[2:0]>, 也会有一个触发被输出到向上计数器。并且, 即使 "101" 被设置为<TRG3MD[2:0]>, 也会在PWM载波峰值时输出一次触发。 注: 在<TRG3MD[2:0]>="011", PMD1TRGCMP3="0x0001"和PMD1MDCR<PWMDMD>="1" (三角波)时, 每个周期都会进行一次触发输出。</p>
11	TRG2BE	R/W	<p>PMD1TRGCMP2<TRGCMP2[15:0]>缓存器的非同步更新 该位可从PMD1TRGCMP2启用后续阶段缓存器的非同步更新。 0: 同步更新 1: 非同步更新(可立即反映已写入PMD1TRGCMP2的值)。</p> <p>注: 有关详细更新时序, 见"表14-6 触发比较寄存器的缓存器更新时序"。 注: PMD1MDEN<PWMDEN>="0" 时, 无论是设置如何, 均采用非同步更新。</p>

位	比特符号	类型	功能
10-8	TRG2MD[2:0]	R/W	<p>PMD1TRGCMP2模式设置</p> <p>该寄存器选择触发电路的匹配模式。</p> <p>000: 触发输出已被禁用</p> <p>001: 向下计数下降计数匹配时的触发输出</p> <p>010: 向上计数上升计数匹配时的触发输出</p> <p>011: 上升-/下降-计数匹配时的触发输出</p> <p>100: PWM载波峰值时的触发输出</p> <p>101: PWM载波谷值时的触发输出</p> <p>110: PWM载波峰值/谷值时的触发输出</p> <p>111: 触发输出已被禁用</p> <p>注: 在 "0" 被设置为PMD1MDCR<PWMMD> (锯齿波)时, 即使 "001" 被设置为 <TRG2MD[2:0]>, 也会有一个触发被输出到向上计数器。并且, 即使 "101" 被设置为 <TRG2MD[2:0]>, 也会在PWM载波峰值时输出一次触发。</p> <p>注: 在<TRG2MD[2:0]>="011", PMD1TRGCMP2="0x0001"和PMD1MDCR<PWMMD>="1" (三角波)时, 每个周期会进行一次触发输出。</p>
7	TRG1BE	R/W	<p>PMD1TRGCMP1<TRGCMP1[15:0]>缓存器的非同步更新</p> <p>该位可从PMD1TRGCMP1启用后续阶段缓存器的非同步更新。</p> <p>0: 同步更新</p> <p>1: 非同步更新(可立即反映已写入PMD1TRGCMP1的值)。</p> <p>注: 有关详细更新时序, 见"表14-6 触发比较寄存器的缓存器更新时序"。</p> <p>注: PMD1MDEN<PWMMEN>="0" 时, 无论是设置如何, 均应采用非同步更新。</p>
6-4	TRG1MD[2:0]	R/W	<p>PMD1TRGCMP1模式设置</p> <p>该寄存器选择触发电路的匹配模式。</p> <p>000: 触发输出已被禁用</p> <p>001: 向下计数下降计数匹配时的触发输出</p> <p>010: 向上计数上升计数匹配时的触发输出</p> <p>011: 上升-/下降-计数匹配时的触发输出</p> <p>100: PWM载波峰值时的触发输出</p> <p>101: PWM载波谷值时的触发输出</p> <p>110: PWM载波峰值/谷值时的触发输出</p> <p>111: 触发输出已被禁用</p> <p>注: 在 "0" 被设置为PMD1MDCR<PWMMD>(锯齿波)时, 即使 "001" 被设置为<TRG1MD[2:0]>, 也会有一个触发被输出到向上计数器。并且, 即使 "101" 被设置为<TRG1MD[2:0]>, 也会在PWM载波峰值时输出一次触发。</p> <p>注: 在<TRG1MD[2:0]>="011", PMD1TRGCMP1="0x0001"和PMD1MDCR<PWMMD>="1" (三角波)时, 每个周期会进行一次触发输出。</p>
3	TRG0BE	R/W	<p>PMD1TRGCMP0<TRGCMP0[15:0]>缓存器的非同步更新</p> <p>该位可从PMD1TRGCMP0启用后续阶段缓存器的非同步更新。</p> <p>0: 同步更新</p> <p>1: 非同步更新(可立即反映已写入PMD1TRGCMP0的值)。</p> <p>注: 有关详细更新时序, 见"表14-6 触发比较寄存器的缓存器更新时序"。</p> <p>注: PMD1MDEN<PWMMEN>="0" 时, 无论是设置如何, 均应采用非同步更新。</p>

位	比特符号	类型	功能
2-0	TRG0MD[2:0]	R/W	<p>PMD1TRGCMP0模式设置</p> <p>该寄存器选择触发电路的匹配模式。</p> <p>000: 触发输出已被禁用</p> <p>001: 向下计数下降计数匹配时的触发输出</p> <p>010: 向上计数上升计数匹配时的触发输出</p> <p>011: 上升-/下降-计数匹配时的触发输出</p> <p>100: PWM载波峰值时的触发输出</p> <p>101: PWM载波谷值时的触发输出</p> <p>110: PWM载波峰值/谷值时的触发输出</p> <p>111: 触发输出已被禁用</p> <p>注: 在 "0" 被设置为PMD1MDCR<PWMMMD>(锯齿波)时, 即使 "001" 被设置为<TRG0MD[2:0]>, 也会有一个触发被输出到向上计数器。并且, 即使 "101" 被设置为<TRG0MD [2:0]>, 也会在PWM载波峰值时输出一次触发。</p> <p>注: 在<TRG0MD[2:0]>="011", PMD1TRGCMP0="0x0001"和PMD1MDCR<PWMMMD>="1" (三角波)时, 每个周期会进行一次触发输出。</p>

14.3.8.6 PMD1TRGSYNCR(触发更新时序设置寄存器)

	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	-	-	-	-	-	-	TSYNCS	
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。
1-0	TSYNCS	R/W	<p>触发比较寄存器缓存器的更新时序设置</p> <p>00: 已通过设置PMD1TRGCR<TRGxMD>, 为各触发设置了PWM载波谷值或PWM载波峰值。</p> <p>01: PWM载波谷值</p> <p>10: PWM载波峰值</p> <p>11: PWM载波峰值和PWM载波谷值</p> <p>注: 见"表14-6 触发比较寄存器的缓存器更新时序"。注: 无论设置如何, 非同步更新PMD1MDEN<PWMEN>="0"。</p>

14.3.8.7 PMD1TRGMD(触发输出模式设置寄存器)

	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	-	-	-	-	-	-	TRGOUT	EMGTGE
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	TRGOUT	R/W	触发输出模式 0: 固定触发输出 1: 变量触发输出 如已选择固定触发输出, 则从PMD1TRG0到PMD1TRG3的触发输出端可输出通过<TRGCMP0[15:0]>与<TRGCMP3[15:0]>间匹配而生成的触发信号。PMD1TRG4 和PMD1TRG5不输出这些触发信号。 如已选择可变触发输出, 则<TRGCMP0[15:0]>的输出信号会被输出到自PMD1TRG0至PMD1TRG5触发输出的其中之一。可通过触发输出选择寄存器, 选择该触发输出信号。 注: 如已选择可变触发输出, 则见"表 14-7 触发输出模式"。(<TRGOUT>="1")。
0	EMGTGE	R/W	EMG保护状态的输出启用 0: 在保护状态禁用触发输出 1: 在保护状态启用触发输出 该位可该EMG保护状态启用或停用触发输出。

表 14-7 触发输出型式

<TRGOUT>设置	(比较寄存器)	<TRGSEL[2:0]> 设置	触发输出
<TRGOUT>="0"	PMD1TRGCMP0	x	PMD1TRG0
	PMD1TRGCMP1		PMD1TRG1
	PMD1TRGCMP2		PMD1TRG2
	PMD1TRGCMP3		PMD1TRG3
<TRGOUT>="1"	PMD1TRGCMP0	0	PMD1TRG0
		1	PMD1TRG1
		2	PMD1TRG2
		3	PMD1TRG3
		4	PMD1TRG4
	5	PMD1TRG5	
	PMD1TRGCMP1	x	无触发输出
PMD1TRGCMP2	x	无触发输出	
PMD1TRGCMP3	x	无触发输出	

14.3.8.8 PMD1TRGSEL(触发输出选择寄存器)

	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	-	-	-	-	-	TRGSEL		
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。
2-0	TRGSEL[2:0]	R/W	触发输出选择 000: 从PMD1TRG0输出 001: 从PMD1TRG1输出 010: 从PMD1TRG2输出 011: 从PMD1TRG3输出 100: 从PMD1TRG4 输出 101: 从PMD1TRG5 输出 110: 无触发输出 111: 无触发输出 在可变触发输出模式被选中时, 该字段有效。 (PMD1TRGMD<TRGOUT>="1").通过设置PMD1TRGCMP0寄存器, 可选择一个输出触发(见表14-7)。

注1: 通过将"0"写入到PMD1MODESEL<MDESEL3>, 设置总线模式(默认), 即可将经由总线更新的比较寄存器的数据加载到后续阶段缓存器。

注2: 由于该触发输出选择寄存器为双缓存型, 因此, 可在PWM计数器运行期间改变该周期。

注3: 后续阶段缓存器的更新时序与比较寄存器(PMD 1CMPU/VW)的相同。

注4: 在PMD被禁用时(PMD1MDCR<PWMEN>="0"), 可进行非同步更新。

15. 矢量引擎(VE+)

15.1 概述

15.1.1 特点

矢量引擎有以下功能:

1. 执行矢量控制的基本任务(坐标变换, 相位变换和SIN/COS计算)。
 - 使用固定点格式数据。
 - 软件无需控制小数点对齐。
2. 用马达控制电路(PMD: 可编程马达驱动)和AD转换器(ADC)启用接口(输出控制, 触发生成, 输入处理)。
 - 将计算结果从固定点格式转换为PMD中可用的数据格式。
 - 用PMD和ADC生成交互作用时序数据。
 - 将AD转换结果转换为固定点格式。
3. 利用标准化的数值, 根据其固定点格式的最大值, 计算电流, 电压和转速。
4. 在电流控制中实现PI控制。
5. 实现相位插值(转速的积分)。

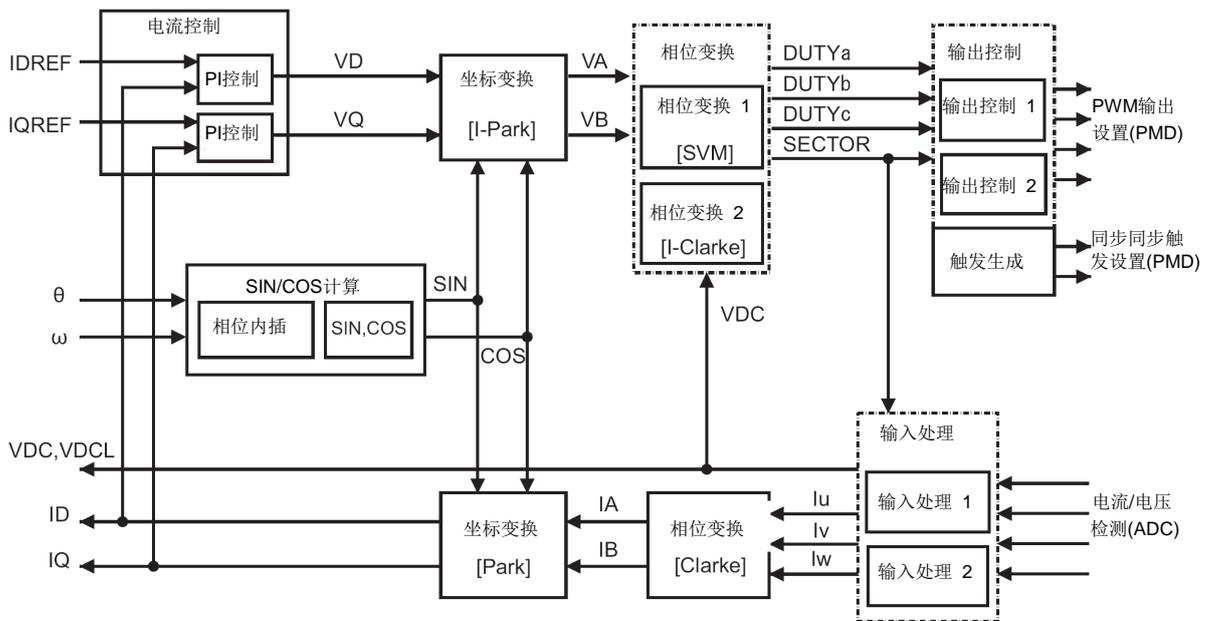


图 15-1 矢量控制方块图

15.1.2 主要规格

1. 空间矢量调制和逆克拉克变换用于2 相位到3相位变换。空间矢量调制可支持2-相调制和3-相调制。
2. 可生成无传感器电流检测用ADC采样时序。可按照1-分流，3-分流和2-传感器法进行电流检测。
3. 在进行电流控制时，d 轴和q 轴的PI控制需单独进行。也可直接设置d 轴和q 轴电压寄存器。
4. 利用级数展开式，按近似法进行SIN/COS计算。通过使用相位插值，可根据转速直接指定或计算出相位信息。

注：如拟使用矢量引擎，则需设置马达控制电路和ADC。

- 在使用矢量引擎时，必须通过模式选择寄存器 PMD1MODESEL将PMD设置为VE模式。
- 还需针对来自PMD的各同步触发，在ADC中进行适当的设置(启用触发，并选择拟使用的AIN和结果寄存器)。

15.2 配置

图 15-2给出了该矢量引擎的配置。

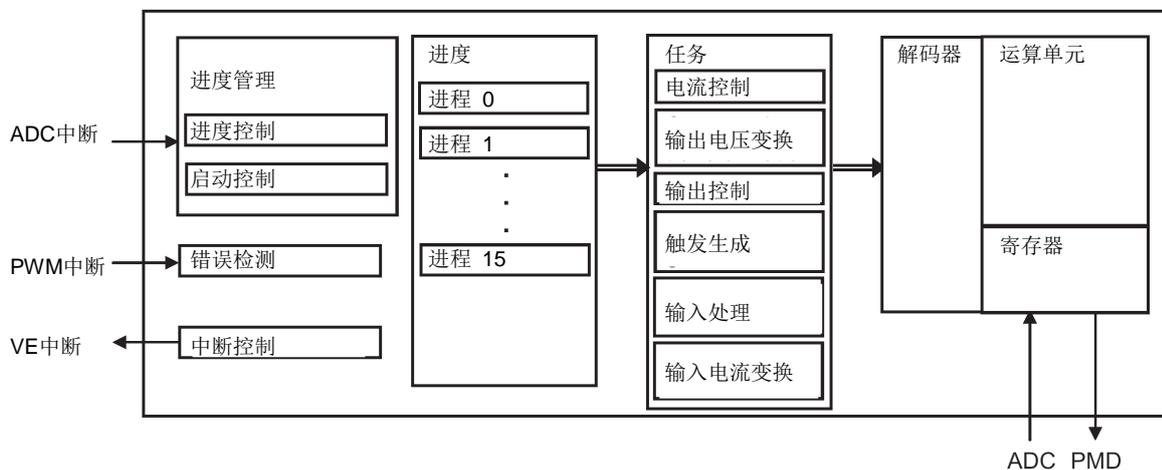


图 15-2 矢量引擎的配置

15.2.1 矢量引擎，马达控制电路和 A/D 转换器之间的相互作用

如图 15-3所示，该矢量引擎可与PMD和ADC直接发生相互作用。

在PMD1MODESEL寄存器被设置为VE模式时，PMD寄存器PMD1CMPU，PMD1CMPV，PMD1CMPW，PMD1MDOUT，PMD1TRGCMP0，PMD1TRGCMP1和PMD1TRGSEL即被分别切换到矢量引擎寄存器VEVCMPU1，VEVCMPV1，VEVCMPW1，VEOUTCR1，VETRGCMP01，VETRGCMP11和VETRGSSEL1。在这种情况下，仅能从该矢量引擎控制这些寄存器，且无法从PMD写入。其它PMD寄存器没有读取/写入限制性条件。

寄存器ADREG0，ADREG1，ADREG2，ADREG3，<UVWISn0>，<UVWISn1>，<UVWISn2>和<UVWISn3>被读取到该矢量引擎中。(n=0~5)

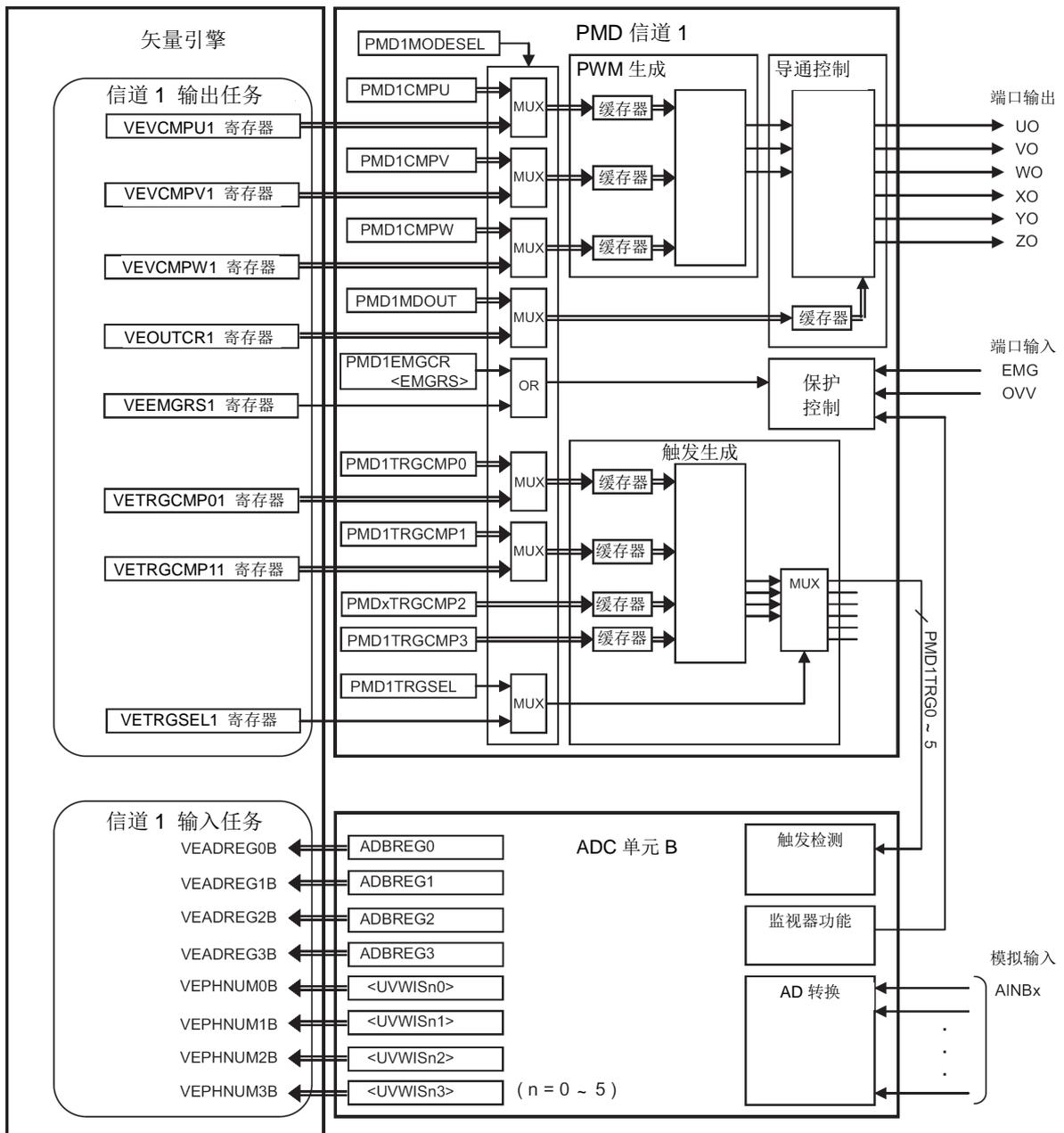


图 15-3 矢量引擎，PMD和ADC之间的相互作用

15.3 寄存器列表

矢量引擎寄存器可分为以下三种类型：

- VE控制寄存器

矢量引擎控制寄存器和暂存寄存器

- 公用寄存器

两通道共用寄存器

- 特定通道寄存器

各通道计算数据和控制寄存器

15.3.1 寄存器列表

VE控制寄存器

寄存器名称			地址
VE启用/禁用	VEEN	R/W	0x4005_0000
CPU启动触发选择	VECPURUNTRG	W	0x4005_0004
任务选择	VETASKAPP	R/W	0x4005_0008
作业进度选择选择	VEACTSCH	R/W	0x4005_000C
进度重复计数	VEREPTIME	R/W	0x4005_0010
启动触发模式	VETRGMODE	R/W	0x4005_0014
出错中断启用/禁用	VEERRINTEN	R/W	0x4005_0018
VE强制终止	VECOMPEND	W	0x4005_001C
错误检测	VEERRDET	R	0x4005_0020
进程执行标志/执行任务	VESCHTASKRUN	R	0x4005_0024
保留	-	R	0x4005_0028
暂存 0	VETMPREG0	R/W	0x4005_002C
暂存 1	VETMPREG1	R/W	0x4005_0030
暂存 2	VETMPREG2	R/W	0x4005_0034
暂存 3	VETMPREG3	R/W	0x4005_0038
暂存 4	VETMPREG4	R/W	0x4005_003C
暂存 5	VETMPREG5	R/W	0x4005_0040
保留	-	R	0x4005_01BC

公用寄存器

寄存器名称			地址
保留	-	R/W	0x4005_0174
ADC启动等待设置	VETADC	R/W	0x4005_0178

通道 1 特定通道寄存器

寄存器名称			地址
状态标志	VEMCTLF1	R/W	0x4005_00DC
任务控制模式	VEMODE1	R/W	0x4005_00E0
流量控制	VEFMODE1	R/W	0x4005_00E4
PWM周期率(PWM周期 × 最高速度(注 1) × 2 ¹⁶)设置	VETPWM1	R/W	0x4005_00E8
转速(速度[Hz] ÷ 最高转速(注 1) × 2 ¹⁵)设置	VEOMEGA1	R/W	0x4005_00EC
马达相位(马达相位[度]/360 × 2 ¹⁶)设置	VETHETA1	R/W	0x4005_00F0

通道 1 特定通道寄存器

寄存器名称			地址
D轴参考值(电流[A] ÷ 最大电流(注 2) × 2 ¹⁵) 设置	VEIDREF1	R/W	0x4005_00F4
Q轴参考值 (电流[A] ÷ 最大电流(注 2) × 2 ¹⁵) 设置	VEIQREF1	R/W	0x4005_00F8
d-轴电压(电压[V] ÷ 最大电压(注 3) × 2 ³¹)设置	VEVD1	R/W	0x4005_00FC
q-轴电压(电压[V] ÷ 最大电压(注 3) × 2 ³¹)设置	VEVQ1	R/W	0x4005_0100
d-轴PI 控制的积分系数:	VECIDK1	R/W	0x4005_0104
d-轴PI 控制比例系数:	VECIDKP1	R/W	0x4005_0108
q-轴PI 控制的积分系数:	VECIQK1	R/W	0x4005_010C
q-轴PI控制比例系数	VECIQKP1	R/W	0x4005_0110
d-轴电压积分项(VDI)上32 位	VEVDIH1	R/W	0x4005_0114
d-轴电压积分项(VDI)下32 位	VEVDILH1	R/W	0x4005_0118
q-轴电压积分项(VQI)下32 位	VEVQIH1	R/W	0x4005_011C
q-轴电压积分项(VQI)下32 位	VEVQILH1	R/W	0x4005_0120
PWM移相模式切换的标准速度	VEFPWMCHG1	R/W	0x4005_0124
PWM周期(按与PWM周期同样的方式进行设置)	VEMDPRD1	R/W	0x4005_0128
最小脉冲宽度	VEMINPLS1	R/W	0x4005_012C
同步触发校正	VETRGCR1	R/W	0x4005_0130
DC电源电压(电压[V] ÷ 最高电压(注 3) × 2 ¹⁵)	VEVDCL1	R/W	0x4005_0134
输出转换用THET余弦值(Q15 数据)	VECOS1	R/W	0x4005_0138
输出转换THETA正弦值(Q15 数据)	VESIN1	R/W	0x4005_013C
先前的余弦值(适用于输入处理)(Q15数据)	VECOSM1	R/W	0x4005_0140
先前的正弦值(适用于输入处理)(Q15数据)	VESINM1	R/W	0x4005_0144
扇区信息	VESECTOR1	R/W	0x4005_0148
先前的扇区信息(适用于输入处理)	VESECTORM1	R/W	0x4005_014C
a-相零电流AD转换结果(注 4)	VEIAO1	R/W	0x4005_0150
b-相零电流AD转换结果(注 4)	VEIBO1	R/W	0x4005_0154
c-相零电流AD转换结果(注 4)	VEICO1	R/W	0x4005_0158
a-相电流AD转换结果(注 4)	VEIAADC1	R/W	0x4005_015C
b-相电流AD转换结果(注 4)	VEIBADC1	R/W	0x4005_0160
c-相电流AD转换结果(注 4)	VEICADC1	R/W	0x4005_0164
直流电源电压(电压[V] ÷ 最高电压(注3) × 2 ¹⁵)	VEVDC1	R/W	0x4005_0168
d-轴电流(电流[A] ÷ 最大电流(注 2) × 2 ³¹)	VEID1	R/W	0x4005_016C
Q轴电流(电流[A] ÷ 最大电流(注 2) × 2 ³¹)	VEIQ1	R/W	0x4005_0170
PMD控制: CMPU设置	VECMU1	R/W	0x4005_019C
PMD控制: CMPV设置	VECMPV1	R/W	0x4005_01A0
PMD控制: CMPW设置	VECMPCV1	R/W	0x4005_01A4
PMD控制: 输出控制(MDOUT)	VEOUTCR1	R/W	0x4005_01A8
PMD控制: TRGCMP0 设置	VETRGCMP01	R/W	0x4005_01AC
PMD控制: TRGCMP1 设置	VETRGCMP11	R/W	0x4005_01B0
PMD控制: 触发选择	VETRGSEL1	R/W	0x4005_01B4
PMD控制: EMG 返回	VEEMGRS1	W	0x4005_01B8

注: 禁止访问各"保留"区域。

注1: 最大速度:可控制或可操纵的最大转速 [Hz]。

注2: 最大电流; (与AD转换器的1LSB对应的相电流值 [A]) × 2¹¹

注3: 最大电压:(与AD转换器的1LSB对应的电源电压 (VDC) 值[V]) × 2¹²

注4: AD转换结果存储于各16-位寄存器的上位12 位中。

15.3.2 VE 控制寄存器

15.3.2.1 VEEN(VE 启用/禁用寄存器)

	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	-	-	-	-	-	-	VEIDLLEN	VEEN
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	VEIDLLEN	R/W	控制该同步时钟在空闲模式是否被提供给该矢量引擎。 0: 无效 1: 有效
0	VEEN	R/W	禁用或启用该矢量引擎。 0: 禁用 1: 启用

注：在该矢量引擎被禁用时(VEEN = "0")，不允许访问该矢量引擎的其它寄存器。

15.3.2.2 VECPURUNTRG(CPU 开始触发选择寄存器)

	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	-	-	-	-	-	-	VCPURTB	-
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	VCPURTB	W	通过编程启动通道 1。 0: - 1: 开始 通过被配置到VETASKAPP<VTASKB>的任务实现的运行启动。
0	-	W	始终写入 "0"。

注1: "1"被写入此位时, 在下一周期即被清除。该位始终被读作"0"。

注2: 拟执行的任务取决于VEACTSCH和VETASKAPP寄存器的设置。

注3: 如果进度和执行中的任务重新启动, 则必须在启动命令执行之前, 通过VECOMPEND寄存器使其终止。

15.3.2.3 VETASKAPP(任务选择寄存器)

	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	VTASKB				-	-	-	-
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"。
7-4	VTASKB[3:0]	R/W	通道 1 任务选择 0x0: 输出控制 1 (任务 0) 0x1: 触发生成(任务 1) 0x2: 输入处理 1 (任务 2) 0x3: 输入相位变换(任务 3) 0x4: 输入坐标轴变换(任务 4) 0x5: 电流控制(任务5) 0x6: SIN/COS计算(任务 6) 0x7: 输出坐标轴转换(任务 7) 0x8: 输出相位变换 1 [SVM](任务 8) 0x9: 输出控制 2(任务 9) 0xA: 输入处理 2 (任务 10) 0xB: 输出相位变换 2 [I-Clarke](任务 11) 0xC: ATAN(任务 12) 0xD: SQRT(任务 13) 0xE: 保留 0xF: 保留 在通过编程启动通道 1 时, 指定拟执行的任务。
3-0	-	R/W	始终写入 "0"。

注: 仅可指定列入进程计划的任務。

15.3.2.4 VECTSCH(操作进程计划选择寄存器)

	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	VACTB				-	-	-	-
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"。
7-4	VACTB[3:0]	R/W	通道 1 规定的单个任务实施或进程。 0x0: 单个任务实施 0x1: 进程 1 0x2: 进程 2 0x3: 进程 3 0x4: 进程 4 0x5: 进程 5 0x6: 进程 6 0x7: 进程 7 0x8: 进程 8 0x9: 进程 9 0xA: 进程 10 0xB: 进程 11 0xC: 进程 12 0xD: 进程 13 0xE: 进程 14 0xF: 进程 15
3-0	-	R/W	始终写入 "0"。

注：详情见"表 15-4 各进程中拟执行的任务"。

15.3.2.5 VEREPTIME(进程重复计数)

	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	VREPB				-	-	-	-
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"。
7-4	VREPB[3:0]	R/W	指定某一进程拟在通道 1 中重复执行的次数。 0: 不执行进程 1 ~ 15: 按指定次数重复执行进程
3-0	-	R/W	始终写入"0"。

注：如已设置为"0"，则任何进程均不会执行。

15.3.2.6 VETRGMODE(启动触发模式)

	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	-	-	-	-	VTRGB		-	-
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"。
3-2	VTRGB[1:0]	R/W	规定可触发通道 1 中输入处理的AD转换端中断。通道 1 触发模式 00: 忽略中断 01: 保留 10: 通过INTADBPDB中断启动输入过程(ADCBPMD1触发同步转换已完成)。 11: 保留
1-0	-	R/W	始终写入 "0"。

15.3.2.7 VEERRINTEN(错误中断启用/禁用)

	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	-	-	-	-	-	-	VERRENB	-
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	VERRENB	R/W	启用或禁用通道 1 中的错误检测中断 0: 禁用 1: 启用
0	-	R/W	始终写入 "0"。

注：如果在执行运算进程时检测到一次PWM中断(不包括激活触发等待)，则 "1" 被作为错误标志进行设置。

15.3.2.8 VECOMPEND(VE 强制终止)

	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	-	-	-	-	-	-	VCENDB	-
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	VCENDB	W	强制终止当前在通道 1 中执行的进程。 0: - 1: 终止
0	-	W	始终写入 "0"。

注: "1"被写入该位时, 在下一周期即被清除。此位始终被读作"0"。

15.3.2.9 VEERDET(错误检测)

	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	-	-	-	-	-	-	VERRDB	-
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	VERRDB		通道 1 错误标记 0: 未检测到任何错误 1: 检测到错误
0	-	R	读作"0"。

注1: 在某一进程执行期间检测到PWM中断时, 即会设置该错误标记(不包括等待启动触发的备用周期)。

注2: 通过读取该寄存器清除该错误标记。

15.3.2.10VESCHTASKRUN(进程执行标志/执行中的任务)

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

位	比特符号	类型	功能
31-10	-	R	读作"0"。
9-6	VRTASKB[3:0]	R	当前在通道 1 中执行的任务的编号 0x0: 输出控制 1 (任务 0) 0x1: 触发生成(任务 1) 0x2: 输入处理1 (任务 2) 0x3: 输入相位变换(任务 3) 0x4: 输入坐标轴变换(任务 4) 0x5: 电流控制(任务5) 0x6: SIN/COS计算(任务 6) 0x7: 输出坐标轴转换(任务 7) 0x8: 输出相位变换 1 [SVM](任务 8) 0x9: 输出控制 2(任务 9) 0xA: 输入处理2 (任务 10) 0xB: 输出相位变换 2 [I-Clarke](任务 11) 0xC: ATAN(任务 12) 0xD: SQRT(任务 13) 0xE: 保留 0xF: 保留
5	VRSCHB	R	通道 1 中的进程执行状态 0: 未执行 1: 执行
4-0	-	R	读作"0"。

15.3.2.11 VETMPREG0 (暂存寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	TMPREG0[31:0]	R/W	暂存寄存器 0

15.3.2.12 VETMPREG1(暂存寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	TMPREG1[31:0]	R/W	暂存寄存器 1

15.3.2.13 VETMPREG2 (暂存寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	TMPREG2[31:0]	R/W	暂存寄存器 2

15.3.2.14 VETMPREG3(暂存寄存器 3)

	31	30	29	28	27	26	25	24
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	TMPREG3[31:0]	R/W	暂存寄存器 3

15.3.2.15 VETMPREG4 (暂存寄存器 4)

	31	30	29	28	27	26	25	24
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	TMPREG4[31:0]	R/W	暂存寄存器 4

15.3.2.16 VETMPREG5(暂存寄存器 5)

	31	30	29	28	27	26	25	24
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-0	TMPREG5[31:0]	R/W	暂存寄存器 5

15.3.3 公用寄存器

15.3.3.1 VETADC(ADC 启动等待设置寄存器)

	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	TADC							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TADC							
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	TADC[15:0]	R/W	设置一个ADC启动等待时间(见图 15-4)。 0x0000~0xFFFF:(ADCAD转换时间[s] ÷ PWM计数器时钟频率[s])

注1: 此寄存器仅当其处于一分路电流检测模式, 已启用PWM档, 且已选择档 1 时有效。

注2: 这些位通过任务 0来使用。

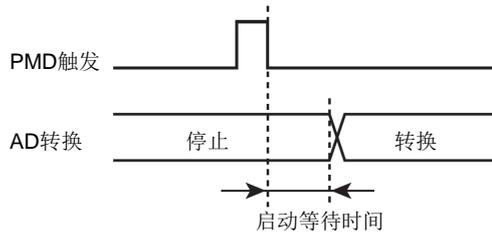


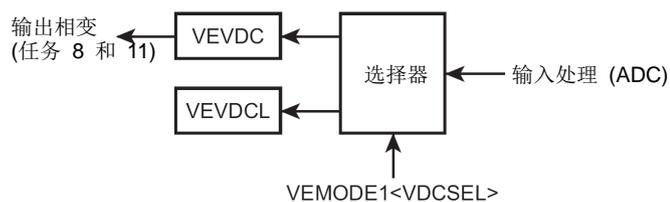
图 15-4 ADC启动等待

15.3.4 特定通道寄存器

15.3.4.1 VE 模式 1 (任务控制模式寄存器)

	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	T7SQRTEEN	T4ATANEN	-	VDCSEL	OCRMD		ZIEN	PVIEN
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	T7SQRTEEN	R/W	与任务 7 联锁的电压标量:算术控制(SQRT计算) 0: 禁用 1: 启用 见"15.4.2.3 输出电压变换(坐标轴变换/相位变换)"
6	T4ATANEN	R/W	与任务 4 联锁的电流矢量(dq):相位计算控制(ATAN计算) 0: 禁用 1: 启用 见"15.4.2.7 输入电流变换(相位变换/坐标轴变换)"
5	-	R/W	始终写入 "0"。
4	VDCSEL	R/W	选择寄存器, 用于存储任务 2 和任务 10 的电源电压 0: 存储VEVDC 1: 存储VEVDCL
3-2	OCRMD[1:0]	R/W	任务 0 和任务 9 的输出控制。 00: 输出OFF 01: 输出启用 10: 短路制动器(上相位输出OFF, 下相位的输出ON) 11: 输出OFF, EMG返回
1	ZIEN	R/W	任务 2 的零电流检测 0: 正常电流检测 1: 零位电流检测
0	PVIEN	R/W	任务 6 的相位插值控制 0: 禁用 1: 启用



注) 当修正VEVDC 寄存器控制的电源电压时，选择VEVDCL 寄存器作为存储位置，并在VEVDC 寄存器在设置修正值。

图 15-5 VEVDCL/VEVDC存储寄存器

15.3.4.2 VEFMODE1(流量控制寄存器)

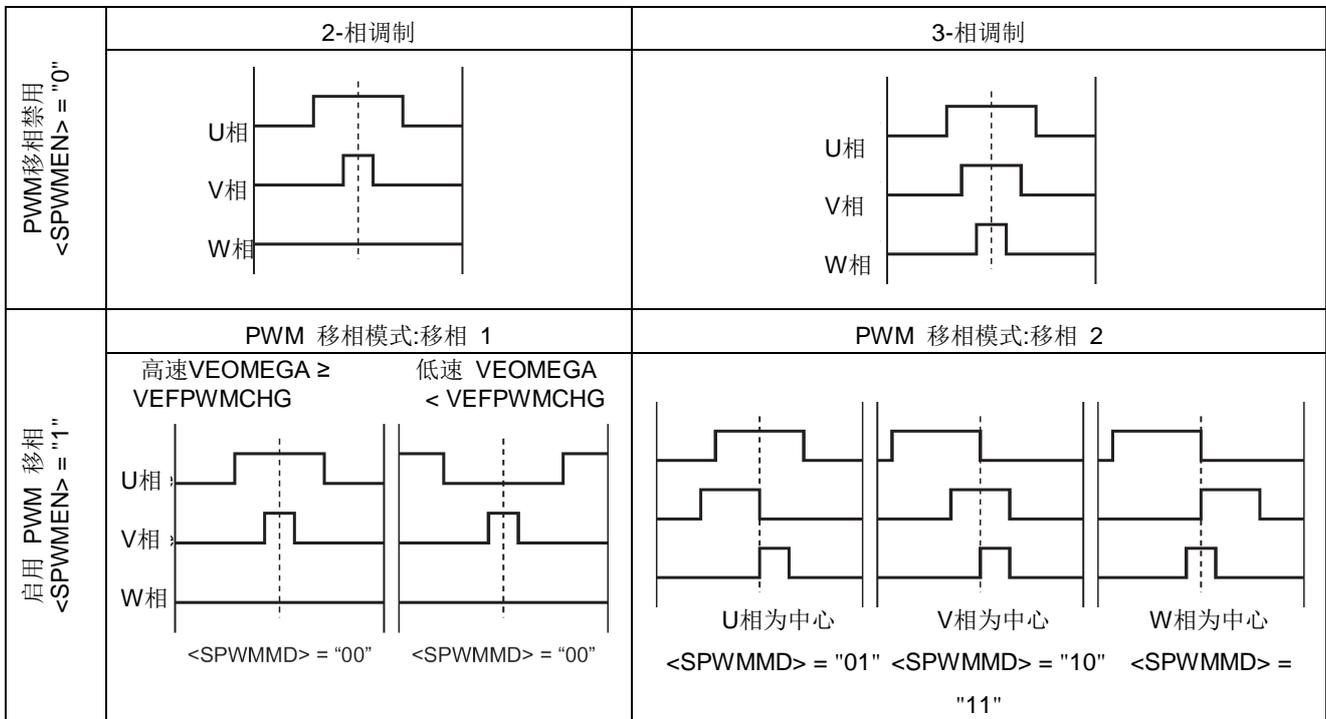
	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	SPWMMD		-	PHCVDIS	PIGSEL	-	MREGDIS	CRCEN
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	IDPLMD	-	IDMODE		SPWMEN	C2PEN
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-16	-	R	读作"0"。
15-14	SPWMMD	R/W	PWM 移相模式选择 00: 移相 1 01: 移相 2 (U-相中心) 10: 移相 2 (V-相中心) 11: 移相 2 (W-相中心) 详细内容请参考"图 15-6 与移相设置和PWM波形的关系"。 注: 当执行输出控制 1 (任务 0)时, 不可选择移相 2 (无效)。
13	-	R/W	始终写入 "0"。
12	PHCVDIS	R/W	相变禁用 0: 2-3 相变启用 (3-相 AC 输出) 1: 2-3 相变禁用(2-相 AC 输出) 注: 当空间矢量调制(任务 8)在运行时, 不能禁用相变 (无效)。
11	PIGSEL	R/W	转换电流控制增益 0: 使用PI系数寄存器的每一个数值作为PI 增益。 1: PI系数寄存器的每一个数值的1/256作为PI 增益。
10	-	R/W	始终写入 "0"。
9	MREGDIS	R/W	保持SIN/COS/象限的前值 0: 有效 1: 无效 在选择无效设置时, VESINM1=VESIN1, VECOSM1=VECOS1, VESECTORM1=VESECTOR1。
8	CRCEN	R/W	触发修正 0: 禁用 1: 启用 注: 其仅在1-分路电流检测模式下的触发(任务 1)和移相生成被禁用, 或移相 1 被选中时有效。
7-6	-	R/W	总是写入 "01"。
5	IDPLMD	R/W	电流极性检测 0: 分流器模式 (电流输入极性转换) 1: 传感器模式 (无电流输入极性转换) 注: 只有在输入处理 2 (任务 10)运行时才有效。
4	-	R/W	总是写入 "1"。

位	比特符号	类型	功能
3-2	IDMODE	R/W	电流检测模式 00: 3-分流器 (注 1) 01: 2-传感器 (注 2) 10: 1-分流器 (对于向上计数PMD TRG) (注释 3 及注释 4) 11: 1-分流器 (对于向下计数 PMD TRG) (注释 3 及注释 4) 注1:在输入处理 2期间的电流检测(任务 10) 为 3-相电流检测。注 2:输入处理 2 (任务 10)期间的电流检测, 属于 2-相电流检测。 注4: 如果已执行输出控制 2 (任务 9)及输入处理 2 (任务 10), 设置PWM 移相 2。
1	SPWMEN	R/W	启用或者禁用PWM 移相。 0: 禁用 1: 启用 注: 输出控制 1 (任务 0)及输入处理 1 (任务 2) 仅支持移相 1。输出控制 2 (任务 9)及输入处理 2 (任务 10) 仅支持移相 2。
0	C2PEN	R/W	选择3-相或者 2-相调制。 0: 3-相调制 1: 2-相调制

注 3: 当使用1-分流器模式时, 可接受PMDTRG 如下。

VEFMODE1 <IDMODE[1:0]>	PMD1TRGCR <TRG0MD[2:0]>	PMD1TRGCR <TRG1MD[2:0]>
10	010 (向上计数)	010 (向上计数)
10	101(载波谷值)	010 (向上计数)
11	001 (向下计数)	001 (向下计数)
11	001 (向下计数)	101(载波谷值)



- 注 1) 可选择移相-1, 仅 2-相调制。
- 注 2) 可选择移相-2, 仅 3-相调制。
- 注 3) 移相-2必须设置PMD寄存器。

图 15-6 与移相设置及PWM 波形的关系

15.3.4.3 VETPWM1(PWM 周期速率寄存器)

	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	TPWM							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TPWM							
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	TPWM[15:0]	R/W	设置一个PWM周期率(当启用相位内插时有效, 16-位定点数据:0.0~1.0)如下: $0x0000\sim 0xFFFF: PWM \text{ 周期 } [s] \div Max_Hz \times 2^{16}$ (Max _ Hz:最大旋转速度[Hz]) (指示PWM频率和最大旋转速度间的比率。)

注: 当启用相位内插时用于SIN/COS计算(任务 6)。

15.3.4.4 VEOMEGA1(转速寄存器)

	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	OMEGA							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	OMEGA							
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	OMEGA[15:0]	R/W	设置旋转速度 (16-位定点数据:-1.0~ 1.0) 如下: 0x8000~0x7FFF:旋转速度 [Hz] ÷Max_Hz × 2 ¹⁵ (Max_Hz: 最大旋转速度[Hz])

注1: 当启用相位内插时用于SIN/COS计算(任务 6)。

注2: 当1-分流器电流检测的PWM移相被选择作为输出控制 1 (任务 0)使用。

15.3.4.5 VETHETA1(马达相寄存器)

	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	THETA							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	THETA							
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	THETA[15:0]	R/W	设置相位数据(16-位定点数据:0.0~1.0)如下: 公式: 相 [deg] ÷ 360 × 2 ¹⁶

注1: 用于SIN/COS计算(任务6)。

注2: 当启用相位内插时, 对其进行更新以计算SIN/COS(任务 6)。

15.3.4.6 VECOS1/VESIN1/VECOSM1/VESINM1(SIN/COS 寄存器)

VECOS1

	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	COS							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	COS							
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	COS[15:0]	R/W	余弦值基于THETA值(16-位定点数据: -1.0~1.0) 余弦值: 0x8000~0x7FFF

注1: 当执行SIN/COS(任务 6)计算时更新。

注2: 在输出坐标轴转换(任务 7)中使用。

VESIN1

	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	SIN							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SIN							
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	SIN[15:0]	R/W	正弦值基于THETA值 (16-位定点数据: -1.0~1.0) 正弦值: 0x8000~0x7FFF

注1: 当执行SIN/COS(任务 6)计算时更新。

注2: 在输出坐标轴转换(任务 7)中使用。

VECOSM1

	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	COSM							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	COSM							
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	COSM[15:0]	R/W	VECOS1寄存器的前一数值 余弦值(之前的数值):0x8000 ~ 0x7FFF

注1: 当执行SIN/COS(任务 6)计算时更新。

注2: 在输入坐标轴转换(任务 4)中使用。

VESINM1

	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	SINM							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	SINM							
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	SINM[15:0]	R/W	VESIN1寄存器的前一数值 正弦值(之前的数值)0x8000 ~ 0x7FFF

注1: 当执行SIN/COS(任务 6)计算时更新。

注2: 在输入坐标轴转换(任务 4)中使用。

15.3.4.7 VEIDREF1/VEIQREF1(d-轴/q-轴电流标准寄存器)

VEIDREF1

	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	IDREF							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	IDREF							
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	IDREF[15:0]	R/W	d-轴电流参考值 (16-位定点数据:-1.0~1.0) 0x8000~0x7FFF(数值设置为:d-轴电流参考值 [A] ÷ Max_I × 2 ¹⁵) Max_I: (相电流值[A]对应ADC的1 LSB)×2 ¹¹

注：在电流控制中使用(任务 5)。

VEIQREF1

	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	IQREF							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	IQREF							
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	IQREF[15:0]	R/W	q-轴电流参考值 (16-位定点数据:-1.0~1.0) 0x8000~0x7FFF(数值设置为:q-轴电流参考[A]÷最大 _I ×2 ¹⁵) 最大 _I : (相电流值[A]对应ADC的1 LSB)×2 ¹¹

注：在电流控制中使用(任务 5)。

15.3.4.8 VEVD1/VEVQ1(d-轴/q-轴电压寄存器)

VEVD1

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

位	比特符号	类型	功能
31-0	VD[31:0]	R/W	d-轴电压 (32-位定点数据:-1.0~1.0) 0x8000_0000 ~ 0x7FFF_FFFF(d-轴电压÷最大_V×2 ³¹) 最大_V: (电源电压 (VDC)值[V] 对应ADC)×2 ¹² 的 LSB 。

注1: 当执行电流控制(任务 5)时更新。

注2: 在输出坐标轴转换(任务 7)中使用。

VEVQ1

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

位	比特符号	类型	功能
31-0	VQ[31:0]	R/W	q-轴电压 (32-位定点数据:-1.0~1.0) 0x8000_0000~ 0x7FFF_FFFF(q-轴电压÷最大_V×2 ³¹) 最大_V: (电源电压 (VDC)值[V] 对应ADC)×2 ¹² 的 LSB 。

注1: 当执行电流控制(任务 5)时更新。

注2: 在输出坐标轴转换(任务 7)中使用。

15.3.4.9 VECIDKI1/VECIDKP1/VEVCIQKI1/VEVCIQKP1(PI 控制系数寄存器)

VECIDKI1

	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	CIDKI							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CIDKI							
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	CIDKI[15:0]	R/W	d-轴PI 控制的积分系数: 0x8000 ~ 0x7FFF

VECIDKP1

	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	CIDKP							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CIDKP							
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	CIDKP[15:0]	R/W	d-轴PI 控制的比例系数:0x8000 ~ 0x7FFF

VEVCIQK1

	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	CIQKI							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CIQKI							
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	CIQKI[15:0]	R/W	q-轴PI 控制的积分系数: 0x8000 ~ 0x7FFF:

VECIQKP1

	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	CIQKP							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CIQKP							
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	CIQKP[15:0]	R/W	q-轴PI控制比例系数: 0x8000 ~ 0x7FFF

15.3.4.10VEVDIH1/VEVDILH1/VEVQIH1/VEVQILH1(PI 控制积分项寄存器)

VEVDIH1

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

位	比特符号	类型	功能
31-0	VDIH[31:0]	R/W	d-轴PI 控制的上32 位积分项 (VDI)

VEVDILH1

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

位	比特符号	类型	功能
31-16	VDILH[15:0]	R/W	d-轴PI控制的下16 位积分项 (VDI)
15-0	-	R	读作"0"。

注1: 64-位定点数据具有63 个小数位(-1.0~1.0)

注2: VDI数据由48 位组成。

VEVQIH1

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

位	比特符号	类型	功能
31-0	VQIH[31:0]	R/W	q-轴PI 控制的上32 位积分项 (VQI)

VEVQILH1

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

位	比特符号	类型	功能
31-16	VQILH[15:0]	R/W	Q轴PI控制用积分项(VQI)的下16 位
15-0	-	R	读作"0"。

注1: 64-位定点数据具有63 个小数位(-1.0~1.0)

注2: VQI数据为48 位。

15.3.4.11VEMCTLF1(状态标志寄存器)

	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	-	-	PLSLFM	PLSLF	-	LVTF	LAVFM	LAVF
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"。
7-6	-	R/W	始终写入"0"。
5	PLSLFM	R/W	<PLSLF>前值 注：当执行电流控制 (任务 0及任务 9)时更新。
4	PLSLF	R/W	脉冲宽度的最小差异 在选择了1-分流器电流检测来执行输出控制 1 (任务 0)的条件下 脉冲宽度的最小差异< VEMINPLS1<MINPLS> 案例="1" 如果在执行输出控制 2 (任务 9)。 最小ON 宽度或最小OFF宽度 < VEMINPLS1<MINPLS> 案例 ="1"
3	-	R/W	始终写入"0"。
2	LVTF	R/W	电源电压较低标志 VEVDC1<VDC> ≥ 0x0100 (1/128) 案例="0" VEVDC1<VDC> < 0x0100 (1/128) 案例="1" 注：当执行输出相位(任务 5)时更新。
1	LAVFM	R/W	前 <LAVF>值 当在1-分流器电流检测启用PWM 移相并且执行输出控制 1 (任务 0)时更新。
0	LAVF	R/W	低速标志 当在1-分流器电流检测启用PWM 移相并且执行输出控制 1 (任务 0)时更新。 0: 高速 1: 低速 VEOMEGA1<OMEGA> ≥ VEFPWMCHG1<FPWMCHG>案例="0" VEOMEGA1<OMEGA> < VEFPWMCHG1<FPWMCHG>案例="1"

15.3.4.12VEFPWMCHG1(PWM 切换速度设置寄存器)

	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	FPWMCHG							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	FPWMCHG							
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	FPWMCHG[15:0]	R/W	<p>当启用PWM 移相时设置转换速度。</p> <p>要设置为此寄存器的A 值按如下方法计算:</p> <p>转换速度 [Hz] \div Max _Hz $\times 2^{15}$</p> <p>(Max _Hz: 最大旋转速度[Hz])</p>

注：执行任务 0 时，它在当启用1-分路电流检测的PWM移相且选择移相 1 时的情况下使用。

15.3.4.13VMDPRD1(PWM 周期寄存器)

	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	VMDPRD							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VMDPRD							
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	VMDPRD[15:0]	R/W	PWM 周期 设置PMD的PMD1MDPRD 寄存器的数值。

注：其用于进行输出控制(任务 0 和任务 9)与触发生成(任务 1)。

15.3.4.14VEMINPLS1(最小脉冲宽度设置寄存器)

	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	MINPLS							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	MINPLS							
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	MINPLS[15:0]	R/W	<p>设置最小脉冲宽度差异的参考值(3-相 PWM (VECMPU1, VECMPV1 及 VECMPW1)的最小差异值) , 当在执行1-分流器电流检测及输出控制1(任务 0)时启用PWM 移相。</p> <p>计算公式如下 时钟宽度差异 [s] ÷ PWM 计数器时周期 [s]</p> <p>在执行输出控制 2(任务 9)时, 设置最小脉冲宽度(3-相PWM(VECMPU1, VECMPV1与 VECMPW1)的最小占空值)的参考值。</p> <p>计算公式如下 脉冲宽度 [s] ÷ PWM 计数器时钟周期 [s]</p>

15.3.4.15VESECTOR1/VESECTORM1(扇区信息寄存器)

VESECTOR1

	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	-	-	-	-	SECTOR			
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	SECTOR[3:0]	R/W	扇区信息 值: 0x0~0xB 表明每个为30度的12个象限输出时的旋转位置。

注1: 当执行输出相转换(任务 8 及任务 11)时更新。

注2: 在执行输出控制 1 (任务 0)时使用。

VESECTORM1

	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	-	-	-	-	SECTORM			
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	SECTORM[3:0]	R/W	先前扇区信息值:0x0 - 0xB 在输入处理时使用。

注1: 当执行输出相转换(任务 8 及任务 11)时更新。

注2: 在执行输入处理 1 (任务 2)时使用。

15.3.4.16VEIAO1/VEIBO1/VEICO1(零电流寄存器)

VEIAO1

	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	IAO							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	IAO							
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	IAO[15:0]	R/W	在零-电流处，一个-相的AD 转换结果。 (存储当马达停止时一个-相电流的AD转换结果。)

注1: 当选择零电流检测模式，在输入处理 1 (任务 2)中更新。

注2: AD转换结果存储在<IAO[15:4]>及<IAO[3:0]>，并且<IAO[3:0]>始终设置为 "0"。

VEIBO1

	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	IBO							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	IBO							
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	IBO[15:0]	R/W	在零电流处的b-相AD 转换结果。 (当马达停止时, 存储b-相电流的AD 转换结果。)

注1: 当选择零电流检测模式, 在输入处理 1 (任务 2)中更新。

注2: AD转换结果存储在<IBO[15:4]>并且<IBO[3:0]>总是设置为 "0"。

VEICO1

	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	ICO							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ICO							
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	ICO[15:0]	R/W	在零-电流处的c-相电流AD 转换结果。 (当马达停止时, 存储C-相的AD 转换结果。)

注1: 当选择零电流检测模式, 在输入处理 1 (任务 2)中更新。

注2: AD转换结果存储在<ICO[15:4]> , 并且<ICO[3:0]>总是设置为 "0"。

15.3.4.17VEIAADC1/VEIBADC1/VEICADC1(电流 ADC 结果寄存器)

VEIAADC1

	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	IAADC							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	IAADC							
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	IAADC[15:0]	R/W	存储a-相电流的AD转换结果:0x0000~0xFFFF:

注1: 在执行输入处理(任务 2及任务 10) 时更新。

注2: AD 转换结果存储在<IAADC[15:4]>, 且<IAADC[3:0]> 总是设置为 "0" 位。

VEIBADC1

	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	IBADC							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	IBADC							
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	IBADC[15:0]	R/W	存储b-相电流的AD转换结果:0x0000~0xFFFF:

注1: 在执行输入处理(任务 2及任务 10) 时更新。

注2: AD 转换结果存储在<IBADC[15:4]>, 且<IBADC[3:0]> 总是设置为 "0"。

VEICADC1

	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	ICADC							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ICADC							
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	ICADC[15:0]	R/W	存储c-相电流的AD转换结果:0x0000~0xFFFF:

注1: 在执行输入处理(任务2及任务10) 时更新。

注2: AD 转换结果存储在<ICADC[15:4]>, 且<ICADC[3:0]> 总是设置为 "0"。

15.3.4.18VEVDC1/VEVDCL1(供给电压寄存器)

VEVDC1

	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	VDC							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VDC							
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	VDC[15:0]	R/W	电源电压 (16-位定点数据:0~1.0)值:0x0000 ~ 0x7FFF 实际电压值为:VDC 值×最大_V 值÷2 ¹⁵ 最大_V:(电源电压值[V] 对应ADC)×2 ¹² 的1 LSB。

注1: 当寄存器VEMODE1 被选择作为"存储VEVDC"时, 在输入处理(任务 2 及 10)更新。

注2: 当执行输出相位转换(任务 8 及 11)时使用。

VEVDCL1

	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	VDCL							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VDCL							
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	VDCL[15:0]	R/W	电源电压 (16-位定点数据:0 ~1.0)值:0x0000 ~0x7FFF 实际电压值为:VDCL 值× 最大_V 值÷ 2 ¹⁵ 最大_V: (电源电压值[V] 对应ADC)×2 ¹² 的1 LSB。

注: 当寄存器VEMODE1被选择作为"存储VEVDCL"时, 在输入处理(任务 2 及 10)时更新。

15.3.4.19VEID1/VEIQ1(d-轴/q-轴电流寄存器)

VEID1

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

位	比特符号	类型	功能
31-0	ID[31:0]	R/W	d-轴电流 (32-位定点数据: -1.0~1.0) d-轴电流: 0x8000_0000~0x7FFF_FFFF 实际电流值为: ID 值× 最大_I 值÷ 2 ³¹ 最大_I: (相电流值[A]对应ADC的1 LSB)×2 ¹¹

注1: 当执行输入坐标轴转换 (任务 4)时更新。

注2: 在电流控制中使用(任务5)。

VEIQ1

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

位	比特符号	类型	功能
31-0	IQ[31:0]	R/W	q-轴电流 (32-位定点数据: -1.0~1.0) q-轴电流: 0x8000_0000~0x7FFF_FFFF 实际电流值为: IQ 值× 最大_I 值÷ 2 ³¹ 最大_I: (相电流值[A]对应ADC的1 LSB)×2 ¹¹

注1: 当执行输入坐标轴转换 (任务 4)时更新。

注2: 在电流控制中使用(任务 5)。

15.3.4.20 VECMPU1 / VECMPV1 / VECMPW1 (PWM 工作寄存器)

VECMPU1

	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	VCMPU							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VCMPU							
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	VCMPU[15:0]	R/W	U-相PWM设置 U-相PWM脉冲宽度: 0x0000~0xFFFF:

注1: 当执行输出控制(任务 0及 9) 时更新。

注2: 在触发生成时使用(任务 1)。

VECMPV1

	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	VCMPV							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VCMPV							
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	VCMPV[15:0]	R/W	V-相PWM 设置 V-相PWM 脉冲宽度: 0x0000~0xFFFF

注1: 当执行输出控制(任务 0 及 9) 时更新。

注2: 在触发生成时使用(任务 1)。

VECMPW1

	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	VCMPW							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VCMPW							
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	VCMPW[15:0]	R/W	W-相PWM 设置 W-相PWM 脉冲宽度: 0x0000~0xFFFF

注1: 当执行输出控制(任务 0 及 9) 时更新。

注2: 在触发生成时使用(任务 1)。

15.3.4.21VEOUTCR1(6 相输出控制寄存器)

	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	-	-	-	-	-	-	-	WPWM
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VPWM	UPWM	WOC		VOC		UOC	
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-9	-	R	读作"0"。
8	WPWM	R/W	W-相PWM 0: ON/OFF输出 1: PWM输出
7	VPWM	R/W	V-相PWM 0: ON/OFF输出 1: PWM输出
6	UPWM	R/W	U-相PWM 0: ON/OFF输出 1: PWM输出
5-4	WOC[1:0]	R/W	W-相输出控制 00: WO OFF, ZO OFF (注) 01: WO ON, ZO OFF 10: WO OFF, ZO ON 11: WO ON, ZO ON (注) 当 <WPWM>=1, WO及ZO均为ON。
3-2	VOC[1:0]	R/W	V-相输出控制 00: VO OFF, YO OFF (注) 01: VO ON, YO OFF 10: VO OFF, YO ON 11: VO ON, YO ON (注) 当 <VPWM>=1, VO及YO均为ON。
1-0	UOC[1:0]	R/W	U-相输出控制 00: UO OFF, XO OFF (注) 01: UO ON, XO OFF 10: UO OFF, XO ON 11: UO ON, XO ON (注) 当 <UPWM>=1, UO及XO均为ON。

注: 当执行输出控制(任务 0 及 9) 时更新。

PMD的U, V及W-相输出控制显示如下(该表仅给出了用于VE的那些组合)。

表 15-1 <UPWM>, <UOC> PMD设置: U-相输出控制(UO, XO)

设置		输出	
<UPWM>	<UOC>	UO	XO
0	00	OFF输出	OFF输出
1	00	PWMU反相输出	PWMU输出
1	11	PWMU输出	PWMU反相输出

表 15-2 <VPWM>, <VOC> PMD设置: V-相输出控制(VO, YO)

设置		输出	
<VPWM>	<VOC>	VO	YO
0	00	OFF输出	OFF输出
1	00	PWMV反相输出	PWMV输出
1	11	PWMV输出	PWMV反相输出

表 15-3 <WPWM>, <WOC> PMD设置: W-相输出控制(WO, ZO)

设置		输出	
<WPWM>	<WOC>	WO	ZO
0	00	OFF输出	OFF输出
1	00	PWMW反相输出	PWMW输出
1	11	PWMW输出	PWMW反相输出

15.3.4.22VETRGCR1 (同步触发校正寄存器)

	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	TRGCRC							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCRC							
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	TRGCRC[15:0]	R/W	用来修正同步出发时序。 设置值为：修正时间[s] ÷ PWM计数器时钟频率[s]

注 1：在触发生成使用(任务 1)。

注 2：当禁用 1 - 分路电流检测或者所选移相 1 的PWM移相，则寄存器有效。

15.3.4.23VETRGCMP01/VETRGCMP11 (触发时间设置寄存器)

VETRGCMP01

	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	VTRGCMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VTRGCMP0							
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	VTRGCMP0[15:0]	R/W	PMD设置：指定同步PMD中的采样ADC触发时序。 0x0000：禁止 0x0001 至(<MDPRD[15:0]> 值-1)：触发时序 <MDPRD[15:0]>值至 0xFFFF:禁止

注 1：当选择下列之一的PMD触发模式时，寄存器有效:倒计时匹配，顺计时匹配，或者向上计数/向下计数匹配。

注 2：当禁用 1 - 分路电流检测或者所选移相 1 (任务 1)的PWM移相时在触发产生(任务 1)中使用。

VTRGCMP11

	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	VTRGCMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VTRGCMP1							
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	VTRGCMP1[15:0]	R/W	PMD设置：规定同步PMD中的采样ADC触发时序。 0x0000：禁止 0x0001 ~ (<MDPRD[15:0]> 值-1)：触发时序 <MDPRD[15:0]> 值 ~ 0xFFFF:禁止

注 1：当选择下列之一的PMD触发模式时，寄存器有效:倒计时匹配，顺计时匹配，或者向上计数/向下计数匹配。

注 2：当设置PMD触发输出模式来触发选择输出(PMD1TRGMD<TRGOUT>="1")时，寄存器有效。

注 3：当禁用 1 - 分路电流检测或者所选移相 1 (任务 1)的PWM移相时在触发产生(任务 1)中使用。

15.3.4.24VETRGSEL1 (同步触发选择寄存器)

	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	-	-	-	-	-	VTRGSEL		
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	VTRGSEL[2:0]	R/W	PMD设置：指定在<VTRGCMP0[15:0]>中指时序输出的同步触发数字。 0 ~ 5:输出触发数字 6 ~ 7:禁止

注 1：当设置PMD触发输出模式来触发选择输出(PMD1TRGMD<TRGOUT>="1")时，该触发有效。

注 2：当执行触发产生(任务 1)时更新。

15.3.4.25VEEMGRS1 (EMG 返回控制寄存器)

	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	-	-	-	-	-	-	-	EMGRS
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	EMGRS	W	PMD设置：从EMG状态返回的EMG返回命令。 0: - 1: EMG返回命令

注 1：在"1"被写入该位时，其在下一周期即被清除。该位始终被读取为"0"。

注 2：当在EMG模式执行输出控制 (任务 0 及 9)，设置为"1"。

15.4 操作说明

15.4.1 进程管理

图 15-7 马达控制流程图。矢量引擎按通过相关寄存器编程的进程与模式设置进行状态转换。

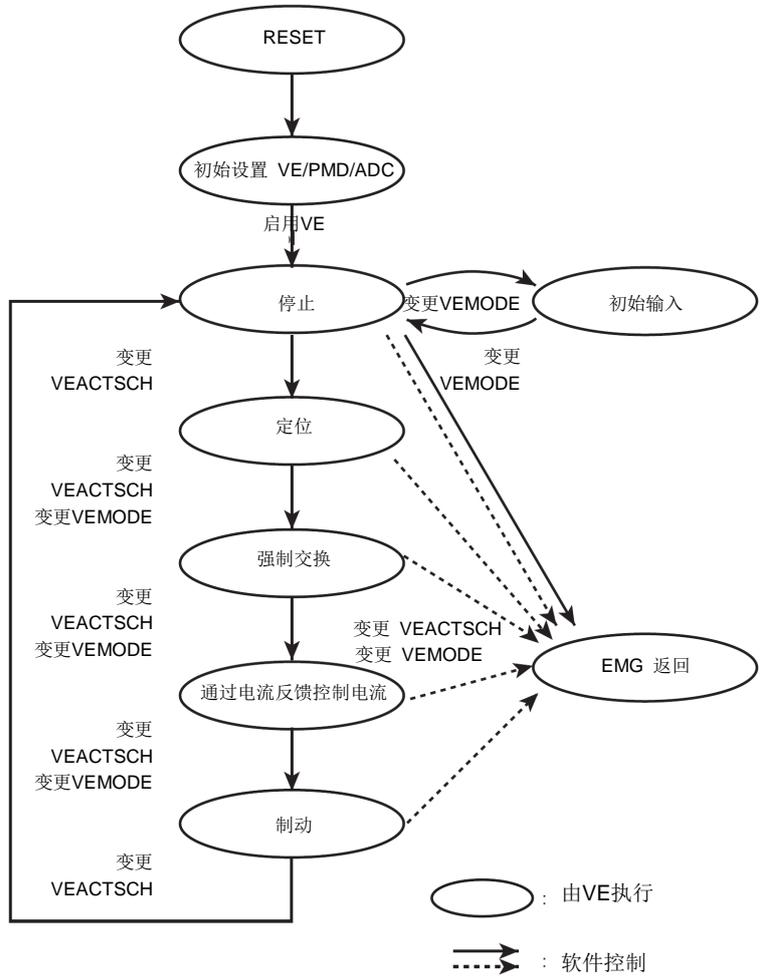


图15-7 马达控制流程示例

RESET	: 微控制器复位
初始设置	: 通过用户创建程序进行初始设置。
停止	: 停止马达
初始输入	: 当马达停止时, 采样并存储零电流数据。
定位	: 确定初始马达位置。
强制交换	: 启动马达.对于指定的周期, 马达在指定速度下旋转, 未通过电流反馈进行控制。
通过电流反馈控制电流。	: 通过电流反馈控制马达旋转。
制动	: 减速控制
EMG返回	: 从EMG状态返回。

15.4.1.1 进程计划控制

VEACTSCH 寄存器用来选择要执行的进程。

一个进程由一个与输出进程处理输出相关的任务，以及一个与输入进程处理输入相关的任务组成。表 15-4 显示每个进程所执行的任务。

可根据马达控制流程的各个步骤，选择VEMODE1 寄存器启用相位插值，控制输出运算，以及零电流检测。

表 15-4 每一进程应执行的任务

进程选择 VEACTSCH <VACTB[3:0]>	输出进程								输入进程			
	电流 控制	SIN/C OS 计算 (注 2)	输出坐 标轴变 换 (注 3)	输出 相位 变换 1	输出 相位 变换 2	输出 控制 1 (注 4)	输出 控制 2 (注 4)	触发 生成	输入 处理 (注 5)	输入处 理 2	输入相 位变换	输入坐 标轴变 换 (注 6)
	T5	T6	T7	T8	T11	T0	T9	T1	T2	T10	T3	T4
0: 单个任务实施	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)	(注1)
1: 进程 1	0	0	0	0	-	0	-	0	0	-	0	0
2: 进程 2	0	0	0	0	-	-	0	0	-	0	0	0
3: 进程 3	0	0	0	-	0	-	0	0	-	0	0	0
4: 进程 4	-	0	0	0	-	0	-	0	0	-	0	0
5: 进程 5	-	0	0	0	-	-	0	0	-	0	0	0
6: 进程 6	-	0	0	-	0	-	0	0	-	0	0	0
7: 进程 7	-	0	0	-	0	0	-	0	0	-	0	0
8: 进程 8	0	0	0	-	0	0	-	0	0	-	0	0
9: 进程 9	-	-	-	-	-	0	-	0	0	-	-	-
10: 进程 10	0	0	0	0	-	0	-	-	-	-	-	-
11: 进程 11	0	0	0	0	-	-	0	-	-	-	-	-
12: 进程 12	0	0	0	-	0	-	0	-	-	-	-	-
13: 进程 13	0	0	0	-	0	0	-	-	-	-	-	-
14: 进程 14	-	-	-	-	-	-	-	0	0	-	0	0
15: 进程 15	-	-	-	-	-	-	-	0	-	0	0	0

注 1: 只有在VETASKAPP指时序才执行每一任务。

注 2: 相位内插。

注 3: 当VEMODE1<T7QRTE>为"1", 同时执行电压标量计算 (SQRT计算)时。

注 4: 输出OFF: <EMGRS>

注 5: 通过零电流检测来转换任务操作。

注 6: 在VEMODE1<T4ATANEN>为"1"时, 同时进行电流矢量(dq)相位计算(ATAN计算)。

表 15-5 典型设置示例

寄存器设置	进程选择 VEACTSCH	任务说明 VETASKAPP	相位内插 VEMODE1	输出控制 VEMODE1	零电流检测 VEMODE1
马达控制流程	<VACTB[3:0]>	<VTASKB[3:0]>	<PVIEN>	<OCRMD[1:0]>	<ZIEN>
停止	9	0	x	00	0
初始输入	9	0	x	00	1
定位	1	5	0	01	0
强制交换	1	5	1	01	0
通过电流反馈实现					
速度控制	1	5	1	01	0
制动	4	6	0	01	0
EMG返回	9	0	x	11	0
短时制动	4	6	x	10	0

输出进程通过执行VECPURUNTRG命令起始执行。在与输出相关的所有任务都完成之后，矢量引擎进入备用状态，并等待与各输入相关的任务的启动触发。

输入进程从执行起始触发起始。在与输入相关的所有任务都完成之后，矢量引擎生成一个中断并将其发送到CPU，并进入暂停状态。不过，如果该进程的重复计数(VEREPTIME)被设置为"2"或以上，则在该进程被执行指定次数之前不会生成中断。

注：进程 10 ~ 15 不支持重复计数(甚至在VEREPTIME ≥ 2 时就立即退出。).

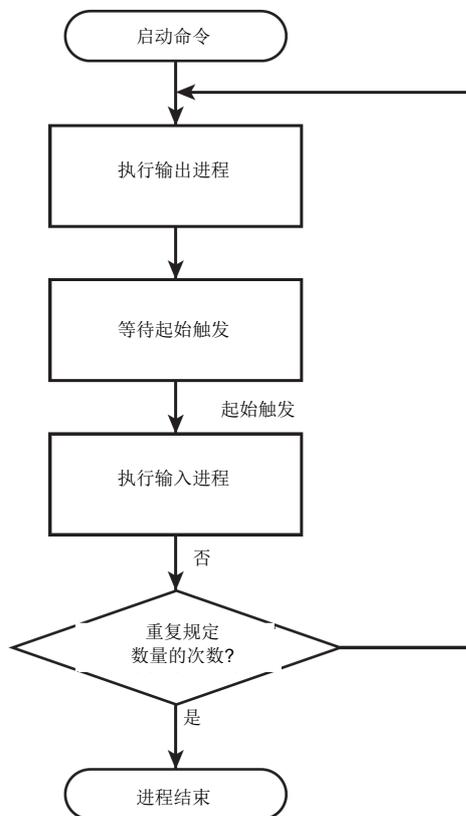


图 15-8 进程执行流程

15.4.1.2 启动控制

在执行该进程之前，将矢量引擎设置为启用(<VEEN>="1")，然后设置运算进程选择寄存器(VEACTSCH)，任务规定寄存器(VETASKAPP)和运算进程重复规定寄存器(VEREPTIME)。进程执行如下。

矢量引擎的进程包括输出进程及输入进程。该矢量引擎执行一般首先执行一个输出进程，进入备用状态，然后通过起动触发器开始执行输入进程。

输出进程及输入进程在如下条件下开始。

- 输出进程启动：
 1. 通过VECPURUNTRG命令。在这种情况下，执行VETASKAPP寄存器规定任务。
 2. 在相应输入进程完成之后重复启动(在VEREPTIME \geq 2时)时。

- 启动输入进程：
 1. 在相应输出进程完成之后，通过起动触发器(已在VETRGMODE寄存器中选择)。
 2. 通过VECPURUNTRG命令。在这种情况下，执行VETASKAPP寄存器规定任务。

15.4.2 任务一览

表 15-6 给出输出和输入进程要执行的任务概要。

当每一个任务要单独执行或者规定作为启动任务时，使用此表中显示的任务号码。

表 15-6 任务列表

任务	任务说明	任务数字	
输出进程	电流控制	控制dq电流	5
	SIN/COS计算	执行SIN/COS计算和相位的插值。	6
	输出坐标轴转换	电压标量计算	7
	输出相位转换 1	2 - 相转换为 3 - 相 [SVM]	8
	输出相位变换 2	2 - 相转换为 3 - 相 [I-Clarke]	11
	输出控制 1	将数据转换为PMD 设置格式。转换PWM 移相。	0
	输出控制 2	将数据转换为PMD 设置格式。转换PWM 移相。	9
	触发生成	产生同步触发时序。	1
输入进程	输入处理 1	获取AD转换结果并将其转换至定点格式。	2
	输入处理 2	获取AD转换结果并将其转换至定点格式。	10
	输入相位转换	3 - 相转换为2 - 相。	3
	输入坐标轴转换	电流矢量(dq)相位计算	4
ATAN 计算	计算反正切	12	
SQRT 计算	提取平方根。	13	

15.4.2.1 电流控制(任务 5)

电流控制任务由d轴电流的PI控制单元和q轴电流的PI控制单元组成，并可计算d轴和q轴电压。
如果电流控制增益选择(VEFMODE1<PIGSEL>)设置为"1"，则比例常数和积分常数乘以 1/256。

1. D轴电流的PI控制

<等式>

$$\Delta ID = VEIDREF1 - VEID1$$
 : 电流参考值和电流反馈间的差异。

$$VEVDI1 = VECIDKI1 \times \Delta ID + VEVDI1$$
 : 积分项计算

$$VEVD1 = VECIDKP1 \times \Delta ID + VEVDI1$$
 : 使用比例项计算电压。

	寄存器名称	功能	
输入	VEID1	d-轴电流:	32 位定点数据 (31 分位)
	VEIDREF1	d轴电流的参考值	16 位定点数据定点数据 (15 分位)
	VECIDKP1	比例系数	16 位数据
	VECIDKI1	积分系数	16 位数据
	VEFMODE1	转换电流控制增益	<PIGSEL>
输出	VEVD1	d-轴电压	32 位定点数据 (31 分位)
内部	VEVDI1	d-轴电压积分项	64 位定点数据 (63 分位)

注: VEVDI包括 64 位.VEVDIH1 为上寄存器, VEVDILH1 为下寄存器。

2. q轴电流的PI控制

<等式>

$$\Delta IQ = VEIQREF1 - VEIQ1$$
 : 电流参考值和电流反馈间的差异。

$$VEVQI1 = VECIQKI1 \times \Delta IQ + VEVQI1$$
 : 积分项计算

$$VEVQ1 = VECIQKP1 \times \Delta IQ + VEVQI1$$
 : 使用比例项计算电压。

	寄存器名称	功能	
输入	VEIQ1	q-轴电流	32 位定点数据(31 分位)
	VEIQREF1	q轴电流的参考值	16 位定点数据 (15 分位)
	VECIQKP1	比例系数	16 位数据
	VECIQKI1	积分系数	16 位数据
	VEFMODE1	转换电流控制增益	<PIGSEL>
输出	VEVQ1	q-轴电压	32 位定点数据(31 分位)
内部	VEVQI1	q-轴电压积分项	64 位定点数据 (63 分位)

注: VEIQ1 包括 64 位.VEVIQH1 为上寄存器, VEVQILH1 为下寄存器。

15.4.2.2 SIN/COS 计算(任务 6)

SIN/COS计算任务由一次相位插值计算和一次SIN/COS计算组成。

相位内插计算与PWM周期整合的旋转速度。其仅在相位插值已启用时被执行 (VEMODE1<PVIEN>="1")。

1. 相位插值

<等式>

$$VETHETA1 = VEOMEGA1 \times VETPWM1 + VETHETA1 \quad : \text{ 旋转速度整合。只有当相位内插启用时。}$$

	寄存器名称	功能	
输入	VETHETA1	相 θ	16 位定点数据 (0.0 ~ 1.0, 16 分位)
	VEOMEGA1	转速	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VETPWM1	PWM周期率	16 位数据
	VEMODE1	相位启用	<PVIEN>
输出	VETHETA1	相 θ	16 位定点数据 (0.0 ~ 1.0, 16 分位)

2. SIN/COS计算

<等式>

$$VESINM1 = VESIN1 \quad : \text{ 存储前值(输入处理)}$$

$$VECOSM1 = VECOS1 \quad : \text{ 存储前值(输入处理)}$$

$$VESIN1 = \sin(VETHETA1 \times \pi) \quad : \text{ SIN/COS计算}$$

$$VECOS1 = \sin((VETHETA1 + 1/4) \times \pi) \quad : \text{ SIN/COS计算}$$

	寄存器名称	功能	
输入	VETHETA1	相 θ	16 位定点数据 (0.0 ~ 1.0, 16 分位)
输出	VESIN1	正弦值在 θ	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VECOS1	余弦值在 θ	
	VESINM1	前正弦值	
	VECOSM1	前一余弦值	

15.4.2.3 输出电压转换(坐标轴转换/相转换)

输出电压变换分 2 步(即坐标轴变换和相位变换)进行。相位变换任务由2类组成, 即空间矢量调制和逆克拉克变换。

1. 输出坐标轴变换(任务7)

在执行输出坐标轴任务时, 利用d轴电压, q轴电压 $\sin \theta$ 和 $\cos \theta$ 计算 α 轴电压和 β 轴电压。

在执行输出坐标轴任务时, 进行坐标轴变换和电压标量计算。

a. 坐标轴变换

<等式>

$$\text{VETMPREG3} = \text{VECOS1} \times \text{VEVD1} - \text{VESIN1} \times \text{VEVQ1}; \quad \text{计算}V_{\alpha}$$

$$\text{VETMPREG4} = \text{VESIN1} \times \text{VEVD1} + \text{VECOS1} \times \text{VEVQ1}; \quad \text{计算}V_{\beta}$$

b. 电压标量计算

当 <T7SQRTEN>为"1"时, 计算电压标量。

<等式>

$$\text{VETMPREG5} = \text{SQRT} ((\text{VETMPREG3})^2 + (\text{VETMPREG4})^2)$$

	寄存器名称	功能	
输入	VEVD1	d-轴电压	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VEVQ1	q-轴电压	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VESIN1	正弦值在 θ	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VECOS1	余弦值在 θ	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VEMODE1	SQRT执行控制	<T7SQRTEN>
输出	VETMPREG3	α -轴电压	32 位 定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β -轴电压	32 位 定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG5	电压标量	32 位定点数据(0 ~ 2.0, 15 分位)

注: <VETMPREG5>寄存器用于存储利用标量运算计算电压时的结果。此寄存器用于多任务, 因此, 其计算结果可被其它任务覆盖。因此, 在启用按标量运算法进行电压计算时, 应在单任务执行级(<VACTB[3:0]>=0x0)执行标量运算。

2. 输出相位变换 1 (空间矢量调制) (任务 8)

输出相变 1 旨在确定一使用 α 轴电压和 β 轴的扇区。在该任务执行期间，需用空间矢量计算各任务的 a 相电压， b 相电压和 c 相电压的占空比。该任务可选择 2-相位或 3-相位调制作为调制类型。

a. 确定一个扇区

<等式>

VESECTORM1 = VESECTOR1

: 存储前一扇区。

如果($V_\alpha \geq 0$ & $V_\beta \geq 0$)

如果($|V_\alpha| \geq |V_\beta| \div \text{SQRT}(3)$)

如果($|V_\alpha| \div \text{SQRT}(3) \geq |V_\beta|$)

: SECTOR=0

否则

: SECTOR=1

否则

: SECTOR=2

否则如果 ($V_\alpha < 0$ & $V_\beta \geq 0$)

如果($|V_\alpha| < |V_\beta| \div \text{SQRT}(3)$)

: SECTOR=3

否则如果($|V_\alpha| \div \text{SQRT}(3) < |V_\beta|$)

: SECTOR=4

否则

: SECTOR=5

否则如果($V_\alpha < 0$ & $V_\beta < 0$)

如果 ($|V_\alpha| \geq |V_\beta| \div \text{SQRT}(3)$)

如果 ($|V_\alpha| \div \text{SQRT}(3) \geq |V_\beta|$)

: SECTOR=6

否则

: SECTOR=7

否则

: SECTOR=8

否则如果($V_\alpha \geq 0$ & $V_\beta < 0$)

如果($|V_\alpha| < |V_\beta| \div \text{SQRT}(3)$)

: SECTOR=9

否则如果 ($|V_\alpha| \div \text{SQRT}(3) < |V_\beta|$)

: SECTOR=10

否则

: SECTOR=11

	寄存器名称	功能	
输入	VETMPREG3	α -轴电压	32 位 定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β -轴电压	32 位 定点数据 (-1.0 ~ 1.0, 31 分位)
输出	VESECTOR1	扇区限	4 位数据
	VESECTORM1	前一扇区	4 位数据

b. 空间矢量调制 (表示仅当3-相位调制和<SECTOR[3:0]>="0"时)
<等式>

- $t1 = \text{SQRT}(3) \div \text{VEVDC1} \times (\text{SQRT}(3) \div 2 \times V\alpha - 1 \div 2 \times V\beta)$: 计算V1 周期。
- $t2 = \text{SQRT}(3) \div \text{VEVDC1} \times V\beta$: 计算V2 周期。
- $t3 = 1 - t1 - t2$: 计算V0 + V7 周期。
- $\text{VETMPREG0} = t1 + t2 + t3 \div 2$: 计算Va。
- $\text{VETMPREG1} = t2 + t3 \div 2$: 计算Vb。
- $\text{VETMPREG2} = t3 \div 2$: 计算Vc。

	寄存器名称	功能	
输入	VETMPREG3	α-轴电压	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β-轴电压	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VEVDC1	电源电压	16 位定点数据 (0.0 ~ 1.0, 15分位)
	VESECTOR1	扇区	4 位数据
	VEFMODE1	调制模式设置 SECTORM 存储设置	<C2PEN> <MREGDIS>
输出	VETMPREG0	a-相电压占空比	32 位定点数据(0.0 ~ 1.0, 31 分位)
	VETMPREG1	b-相电压占空比	32 位定点数据(0.0 ~ 1.0, 31 分位)
	VETMPREG2	c-相电压占空比	32 位定点数据(0.0 ~ 1.0, 31 分位)

3. 输出相位变换 2 (逆克拉克变换) (任务 11)

输出相位变换 2 旨在确定一个使用 α 轴和 β 轴电压的扇区。此项任务可利用逆向克拉克变换分别对A相电压占空比, B相电压占空比以及C相电压占空比进行计算。在此项任务中, 只能选择3-相调制作为调制类型。

此外, 当VEFMODE<PIGSEL>被设置为"1"时, 该任务会计算 2-相位电压的占空比。

- a. 确定一个扇区
<等式>

```

VESECTORM1 = VESECTOR1                                : 存储前一扇区。
如果(V $\alpha$   $\geq$  0 & V $\beta$   $\geq$  0)
如果(|V $\alpha$ |  $\geq$  |V $\beta$ | + SQRT(3))
如果 (|V $\alpha$ | + SQRT(3)  $\geq$  |V $\beta$ |)                        SECTOR=0
否则                                                    SECTOR=1
否则                                                    SECTOR=2
否则如果(V $\alpha$  < 0 & V $\beta$   $\geq$  0)
如果 (|V $\alpha$ | < |V $\beta$ | + SQRT(3))                        SECTOR=3
否则如果 (|V $\alpha$ | + SQRT(3) < |V $\beta$ |)                  SECTOR=4
否则                                                    SECTOR=5
否则如果 (V $\alpha$  < 0 & V $\beta$  < 0)
如果 (|V $\alpha$ |  $\geq$  |V $\beta$ | + SQRT(3))
如果 (|V $\alpha$ | + SQRT(3)  $\geq$  |V $\beta$ |)                        SECTOR=6
否则                                                    SECTOR=7
否则                                                    SECTOR=8
否则如果 (V $\alpha$   $\geq$  0 & V $\beta$  < 0)
如果 (|V $\alpha$ | < |V $\beta$ | + SQRT(3))                        SECTOR=9
否则如果 (|V $\alpha$ | + SQRT(3) < |V $\beta$ |)                  SECTOR=10
否则如果                                                    SECTOR=11
    
```

	寄存器名称	功能	
输入	VETMPREG3	α 轴电压	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β -轴电压	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
输出	VESECTOR1	扇区	4 位数据
	VESECTORM1	前一扇区	4 位数据

b. 逆克拉克变换

<等式>

(VEFMODE<PIGSEL> = "0")

$$\text{VETMPREG0} = 1 \div \text{VEVDC1} \times V\alpha + 1 \div 2 \quad : V\alpha \text{ 占空比}$$

$$\text{VETMPREG1} = 1 \div \text{VEVDC1} \times (-1 \div 2 \times V\alpha + \text{SQRT}(3) \div 2 \times V\beta) + 1 \div 2 \quad : V\beta \text{ 占空比}$$

$$\text{VETMPREG2} = 1 \div \text{VEVDC1} \times (-1 \div 2 \times V\alpha - \text{SQRT}(3) \div 2 \times V\beta) + 1 \div 2 \quad : Vc \text{ 占空比}$$

(VEFMODE<PIGSEL> = "1")

$$\text{VETMPREG0} = 1 \div \text{VEVDC1} \times V\alpha + 1 \div 2 \quad : V\alpha$$

$$\text{VETMPREG1} = 1 \div \text{VEVDC1} \times V\beta + 1 \div 2 \quad : V\beta$$

	寄存器名称	功能	
输入	VETMPREG3	α 轴电压	32 位定点数据(-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β -轴电压	32 位定点数据(-1.0 ~ 1.0, 31 分位)
	VEVDC1	电源电压	16 位定点数据 (0.0 ~ 1.0, 15 分位)
	VESECTOR1	扇区	4 位数据
	VEFMODE1	存储上一个数值设置 相变禁用	<MREGDIS> <PHCVDIS>
输出	VETMPREG0	a-相电压占空比	32 位定点数据(0.0 ~ 1.0, 31 位分位)
	VETMPREG1	b-相电压占空比	32 位定点数据(0.0 ~ 1.0, 31 位分位)
	VETMPREG2	c-相电压占空比	32 位定点数据(0.0 ~ 1.0, 31 位分位)

15.4.2.4 输出控制

输出控制单元可将 3-相电压占空比转换为 PMD 设置格式。将该转换结果设置为 VECMPU1, VECMPV1 和 VECMPW1, 然后按照输出控制设置设置 VEOUTCR1。

输出控制包含 2 种任务类型, 即输出控制 1 任务和输出控制 2 任务。每个任务支持不同的 PWM 输出。

1. 输出控制 1 (任务 0)

输出控制 1 任务既可支持正常的 PWM 输出, 也可支持移相 1 模式下的 PWM 输出。在设置了 "PWM 移相启用" 项后, 一旦转速 (VEOMEGA1) 低于 PWM 移相开关 (VEFPWMCHG1) 的参考值, PWM 输出则会变成 PWM 移相。

注: 只有在 1-分路电流检测模式下才能选择 PWM 移相。

	寄存器名称	功能	
输入	VETMPREG0	a-相电压	32 位定点数据(0.0 ~ 1.0, 31 位分位)
	VETMPREG1	b-相电压	32 位定点数据(0.0 ~ 1.0, 31 位分位)
	VETMPREG2	c-相电压	32 位定点数据(0.0 ~ 1.0, 31 位分位)
	VEMDPRD1	PWM周期	16 位数据(PMD PWM 周期)
	VESECTOR1	扇区	4 位数据
	VEOMEGA1	转速	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VEFPWMCHG1	移相模式切换时的参考速度	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VEMODE1	输出控制操作	<OCRMD[1:0]>
	VEMINPLS1	最小视差	16 位数据
	VEFMODE1	调制模式PWM 移相启用 电流检测模式	<C2PEN> <SPWMEN> <IDMODE[1:0]>
输出	VECMPU1	PMD U-相 PMW 设置	16 位数据(0 ~ MDPRD值)
	VECMPV1	PMD V-相 PMW 设置	16 位数据(0 ~ MDPRD值)
	VECMPW1	PMD W-相 PMW 设置	16 位数据(0 ~ MDPRD值)
	VEOUTCR1	PMD输出控制设置	9 位设置
	VEEMGRS1	PMD EMG 返回	1 位设置
	VEMCTLF1	转速过低标志 脉冲过小标志	<LAVF> <PLSLF>

2. 输出控制 2 (任务 9)

在移相 2 模式下，输出控制 2 任务支持标准PWM输出和PWM输出PWM。

如果PWM移相已被启用(VEFMODE1<SPWMEN>="1")，且PWM移相模式选择(VEFMODE1<SPWMMD>)已被设置为"00"以外的值，则PWM输出变为移相 2 输出。

注：只有在1-分路电流检测模式下才能选择PWM移相。

	寄存器名称	功能	
输入	VETMPREG0	a-相电压	32 位定点数据(0.0 ~ 1.0, 31 分位)
	VETMPREG1	b-相电压	32 位定点数据(0.0 ~ 1.0, 31 分位)
	VETMPREG2	c-相电压	32 位定点数据(0.0 ~ 1.0, 31 分位)
	VEMDPRD1	PWM周期	16 位数据(PMD PWM周期)
	VEFPWMCHG1	移相模式切换时的参考速度 PWM移相模式	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VEMODE1	输出控制操作	<OCRMD[1:0]>
	VEMINPLS1	最小脉冲宽度	16 位数据
	VEFMODE1	PWM移相启用 电流检测模式 PWM移相模式	<SPWMEN> <IDMODE[1:0]> <SPWMMD>
输出	VECMPU1	PMD U-相 PMW设置	16 位数据(0 ~ MDPRD值)
	VECMPV1	PMD V-相 PMW设置	16 位数据(0 ~ MDPRD值)
	VECMPW1	PMD W-相 PMW设置	16 位数据(0 ~ MDPRD值)
	VEOUTCR1	PMD输出控制设置	9 位设置
	VEEMGRS1	PMD EMG 返回	1 位设置
	VEMCTLF1	转速过低标志 脉冲过小标志	<LAVF> <PLSLF>

15.4.2.5 触发生成

触发生成单元可根据相应的电流检测方式，利用PWM设置值(VECMPI1, VECMPV1 和 VECMPW1)对触发时序进行计算，同时还可对VETRGCMP01 和 VETRGCMP11 寄存器进行设置。

注：VETRGCMP01 和VETRGCMP11 仅通过 1-分路电流检测进行更新。

注：选择PWM移相 2 模式时，不会更新VETRGCMP01 和VETRGCMP11。

	寄存器名称	功能	
输入	VECMPI1	PMD U-相 PWM设置	16 位数据(0 ~ MDPRD数值)
	VECMPV1	PMD V-相 PMW设置	16 位数据(0 ~ MDPRD数值)
	VECMPW1	PMD W-相 PMW设置	16 位数据(0 ~ MDPRD数值)
	VEMDPRD1	PWM周期设置	16 位数据(PMD PWM周期)
	VETADC	AD转换时间	16 位数据(0 ~ MDPRD数值)
	VETRGCRC1	触发修正值	16 位数据(0 ~ MDPRD数值)
	VESECTOR1	扇区	4 位数据
	VEMODE1	零电流检测 输出控制操作	<ZIEN> <OCRMD[1:0]>
	VEFMODE1	调制模式PWM移相启 用电流检测模式 启用触发修正 移相PWM模式	<C2PEN> <SPWMEN> <IDMODE[1:0]> <CRCEN> <SPWMMD>
	VEMCTLF1	转速过低标志	<LAVF>
输出	VETRGCMP01	PMD触发0 时序	16 位数据(0 ~ MDPRD数值)
	VETRGCMP11	PMD触发1时序	16 位数据(0 ~ MDPRD数值)
	VETRGSEL1	PMD触发选择	3 位数据

15.4.2.6 输入处理

在输入过程中，矢量引擎从A/D转换器读取转换结果和相位信息。根据电流检测的类型以及移相PWM模式的设置，矢量引擎可将转换结果中的3-相电流和电压转换成定点数据，并对其进行存储。在零电流检测模式下，电流检测结果被储存在零电流寄存器中。

输入过程包含2种任务类型，即输入过程1任务和输入过程2任务。各项任务支持不同的电流检测类型。

1. 输入过程1(任务2)

输入过程1任务支持3-分路电流检测(仅检测两相电流(注1))以及1-分路电流检测。

不过，在PWM移相2模式下，在PWM输出(注2)时不支持1-分路电流检测。

注1：只能检测3-相电流中的两相电流。其余的则用2-相电流的结果进行计算。

注2：只有在1-分路电流检测模式下才能选择PWM移相。

	寄存器名称	功能	
输入	VEADREG0B	ADC单元B转换结果 0	16 位数据(使用 12 位高位数据)
	VEADREG1B	ADC单元B转换结果 1	
	VEADREG2B	ADC单元B转换结果 2	
	VEADREG3B	ADC单元B转换结果 3	
	VEPHNUM0B	ADREG0B检测到的相位信息	2 位数据
	VEPHNUM1B	ADREG1B检测到的相位信息	
	VEPHNUM2B	ADREG2B检测到的相位信息	
	VEPHNUM3B	ADREG3B检测到的相位信息	
	VESECTORM1	扇区信息	4 位数据
	VEMODE1	零电流检测 VDC存储寄存器	<ZIEN> <VDCSEL>
	VEFMODE1	电流检测模式	<IDMODE[1:0]>
	VEMCTLF1	转速过低标志	<LAVFM>
输出	VEVDC1	DC电源电压	16 位定点数据 (0.0 ~ 1.0, 15 分位)
	VEVDCL1	DC电源电压	16 位定点数据 (0.0 ~ 1.0, 15 分位)
	VETMPREG0	a-相电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG1	b-相电流	
VETMPREG2	c-相电流		
内部	VEIAO1	a-相零电流转换结果	16 位数据(使用 12 位高位数据)
	VEIBO1	b-相零电流转换结果	
	VEICO1	c-相零电流转换结果	
	VEIADC1	a-相电流转换结果	16 位数据(使用 12 位高位数据)
	VEIBADC1	b-相电流转换结果	
	VEICADC1	c-相电流转换结果	

2. 输入过程 2 (任务 10)

输入过程 2 任务支持 3-分路电流检测(3-相位检测, 2-相位检测)和 2 传感器电流检测。它在 PWM在PWM移相 2 模式下进行输出时, 还支持 1-分路电流检测。(注)

输入过程 2 任务不支持零电流检测模式。

注: 只有在 1-分路电流检测模式下才能选择PWM移相。

	寄存器名称	功能	
输入	VEADREG0B	ADC单元B转换结果 0	16 位数据(使用 12 位高位数据)
	VEADREG1B	ADC单元B转换结果 1	
	VEADREG2B	ADC单元B转换结果 2	
	VEADREG3B	ADC单元B转换结果 3	
	VEPHNUM0B	ADREG0B检测到的相位信息	2 位数据
	VEPHNUM1B	ADREG1B检测到的相位信息	
	VEPHNUM2B	ADREG2B检测到的相位信息	
	VEPHNUM3B	ADREG3B检测到的相位信息	
	VEMODE1	VDC存储寄存器	<VDCSEL>
	VEFMODE1	电流检测模式 电流检测极性	<IDMODE[1:0]> <IDPLMD>
输出	VEVDC1	DC电源电压	16 位定点数据 (0.0 ~ 1.0, 15 分位)
	VEVDCL1	DC电源电压	16 位定点数据 (0.0 ~ 1.0, 15 分位)
	VETMPREG0	a-相电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG1	b-相电流	
	VETMPREG2	c-相电流	
内部	VEIAADC1	a-相电流转换结果	16 位数据(使用 12 位高位数据)
	VEIBADC1	b-相电流转换结果	
	VEICADC1	c-相电流转换结果	

15.4.2.7 输入电流变换(相位变换/坐标轴变换)

输入电流变换由两个任务(即相位变换和坐标轴变换)组成。

1. 输入相位变换(任务 3)

输入相位变换任务旨在利用 I_a , I_b 和 I_c 对 I_α 和 I_β 进行计算。

<等式>

$$VETMPREG3 = VETMPREG0 \quad : \text{计算 } I_\alpha.$$

$$VETMPREG4 = 1 \div \text{SQRT}(3) \times VETMPREG1 - 1 \div \text{SQRT}(3) \times VETMPREG2 \quad : \text{计算 } I_\beta.$$

	寄存器名称	功能	
输入	VETMPREG0	a-相电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG1	b-相电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG2	c-相电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VEFMODE1	相变禁用	<PHCVDIS>
输出	VETMPREG3	α -轴电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β -轴电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)

2. 输入坐标轴变换(任务 4)

输入坐标轴变换任务可根据 I_α , I_β , VESINM1 和 VECOSM1 计算 I_d 和 I_q 。

a. 坐标轴转换

<等式>

$$VEID1 = VECOSM1 \times VETMPREG3 + VESINM1 \times VETMPREG4 \quad : \text{计算 } I_d.$$

$$VEIQ1 = -VESINM1 \times VETMPREG3 + VECOSM1 \times VETMPREG4 \quad : \text{计算 } I_q.$$

b. 电流矢量相位计算

当 <T4ATANEN> 设置为"1"时, 则可进行相位计算。

<等式>

$$VETMPREG5 = \text{ATAN}(VEID1 / VEIQ1)$$

	寄存器名称	功能	
输入	VETMPREG3	α -轴电流	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VETMPREG4	β -轴电流	
	VESINM1	正弦值在 θ	16 位定点数据 (-1.0 ~ 1.0, 15 分位)
	VECOSM1	余弦值在 θ	
	VEMODE1	启用反正切(ATAN)执行	<T4ATANEN>
	VEMCTLF1	脉冲过小标志	<PLSLFM>
输出	VEID1	d-轴电流:	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
	VEIQ1	q-轴电流:	
	VETMPREG5	反正切(ATAN)结果	32 位数据(0xFFFFE000 至 0x00002000)

15.4.2.8 其它任务

1. ATAN (反正切) (任务 12)

反正切(ATAN)任务旨在输出一个介于 -45° ~ 45° 之间相位，该相位是利用一项介于 -1.0 ~ 1.0 之间的输入进行反正切计算所获得的。

<等式>

$$\text{VETMPREG5} = \text{ATAN}(\text{VETMPREG5})$$

: ATAN计算

	寄存器名称	功能	
输入	VETMPREG5	输入值	32 位定点数据 (-1.0 ~ 1.0, 31 分位)
输出	VETMPREG5	相位值	32 位数据(0xFFFFE000 ~ 0x00002000 (-45°至 45°))

2. SQRT (平方根函数) (任务 13)

SQRT任务输出利用 0.0 ~ 4.0 的输入，用平方根函数计算出的一个值(其范围为 0.0 ~ 2.0)。

<等式>

$$\text{VETMPREG5} = \text{SQRT}(\text{VETMPREG5})$$

: 平方根计算

	寄存器名称	功能	
输入	VETMPREG5	输入值	32 位定点数据 (0.0 ~ 4.0, 15 分位)
输出	VETMPREG5	平方值	32 位定点数据 (0.0 ~ 2.0, 15 分位)

15.5 VE 通道通道，ADC 单元和 PMD 通道通道的组合

在使用矢量引擎的通道时，会限制可使用的PMD和ADC组合。电流检测选择与ADC单元选择的可用组合会发生变化。

表 15-7 VE和PMD组合

矢量引擎	PMD
通道 1	通道 1

表 15-8 VE和ADC组合

矢量引擎		ADC 单元B			
通道	电流检测 VEMODE1 <IDMODE[1:0]>	ADREG0	ADREG1	ADREG2	ADREG3
1	0x	电流检测 1	电流检测 2	电流检测 3	VDC 检测
	1X	电流检测 1	电流检测 2	电流检测 3 (注)	VDC 检测

注：电流检测 3 只能在PWM移相 2 模式下使用。

16. 运算放大器(AMP)

TMPM375FSDMG有一个运算放大器。运算放大器可通过输入端口放大某个电压，并将其输出电压输送至一个12 位逐步近似模拟数字(A/D)转换器。运算放大器用于放大各马达电流测量用分流电阻器间的电压差。

16.1 配置

图 16-1 中显示了运算放大器/模拟转换器的方块图。

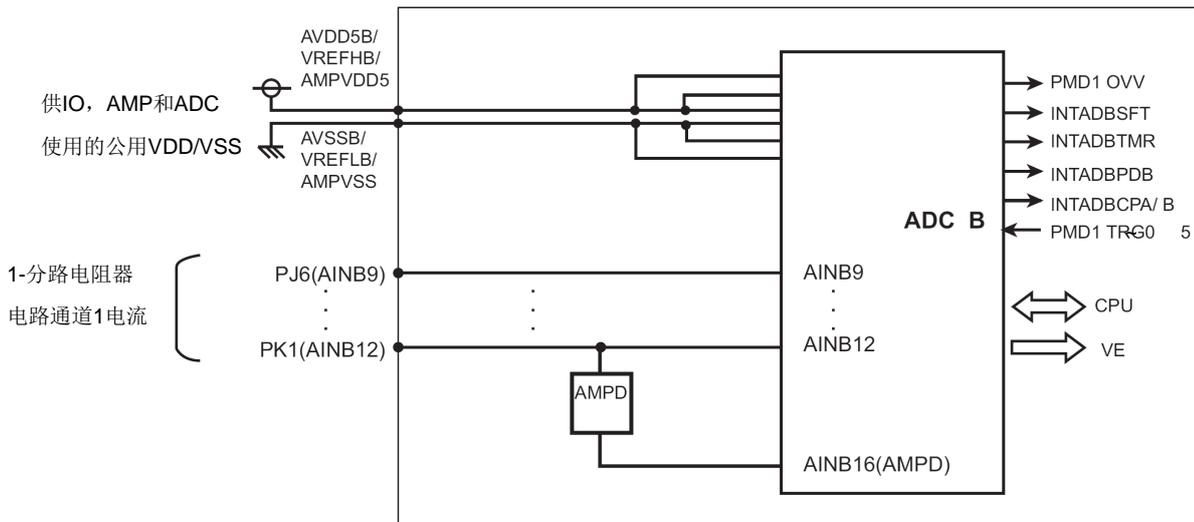


图 16-1 运算放大器/模拟转换器方块图

16.2 寄存器列表

运算放大器可通过 AMPCTLD 进行编程设置。AMPCTLD 允许软件将运算放大器功能设置为启用或禁用状态，并从八个等级中选择一项电压增益。

如使用的是外部运算放大器而不是板上运算放大器，则该板上运算放大器应被禁用(<AMPEN> = 0)。

有关控制寄存器的说明详见如下。

16.2.1 运算放大器

基址 = 0x4003_0400

寄存器		地址(基+)
Amp D 控制寄存器	AMPCTLD	0x0018

16.2.1.1 AMPCTLD(Amp D 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	-	-	-	-	AMPGLIN			AMPEN
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。
3-1	AMPGLIN[2:0]	R/W	增益选择 000: 1.5x 100: 4.0x 001: 2.5x 101: 6.0x 010: 3.0x 110: 8.0x 011: 3.5x 111: 10.0x
0	AMPEN	R/W	AMP启用 0: 禁用 1: 启用

注：在<AMPEN>被设置为"1"时，电路稳定需约 10 μs。

16.3 运行

16.3.1 基本运行

运算放大器 D(AMP D)只支持 1-分路电流检测。经放大的电压被从AMP D送入一个A/D转换器 (AINB16)。

详见图 16-2 中所示的运算放大器方块图。

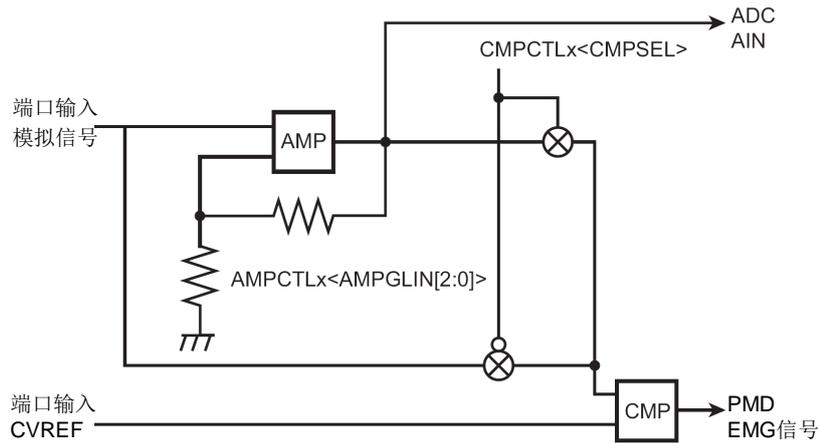


图 16-2 运算放大器



17. 编码器输入电路(ENC)

17.1 概述

编码器输入电路支持四种操作模式，包括编码模式，传感器模式(两种类型)以及定时器模式。其具体功能如下：

- 支持增量式编码器和霍尔传感器IC(可以直接输入霍尔传感器IC的信号)
- 24 位通用定时器模式
- 4 倍(6 倍)电路
- 旋转方向检测电路
- 24 位计数器
- 启用/禁用比较器
- 中断请求输出：1
- 输入信号数字噪声滤波器

17.2 各通道之间的差值

该TMPM375FSDMG有一个单通道增量编码器接口(ENC0)，其可依据增量编码器所提供的输入信号获取马达的绝对位置。

除下列差异外，这些通道的操作完全相同。

表 17-1 各通道间的差异

通道	输入引脚			编码器输入
	a-相电压	b-相	Z 相	中断
通道 0	ENCA0	ENCB0	ENCZ0	INTENC0

17.3 方块图

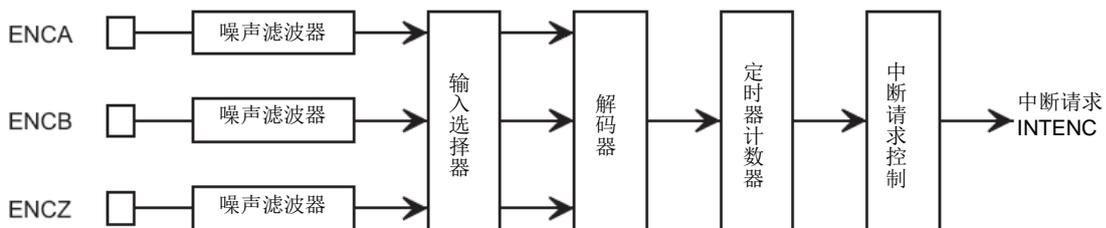


图 17-1 编码器输入电路方块图

17.4 寄存器

17.4.1 寄存器列表

以下为编码器输入电路的控制寄存器和地址。

通道 x	基址
通道 0	0x4001_0400

寄存器名称		地址(基+)
编码器输入控制寄存器	ENTNCR	0x0000
编码器计数器重载寄存器	ENRELOAD	0x0004
编码器比较寄存器	ENINT	0x0008
编码器计数器	ENCNT	0x000C

17.4.2 编码器输入控制寄存器(ENTNCR)

	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	-	-	-	-	-	MODE		P3EN
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ZEN	ENRUN	NR		INTEN	ENDEV		
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能
31-19	-	R	读作"0"。
18-17	MODE[1:0]	R/W	编码器输入模式设置 00: 编码器模式 01: 传感器模式(事件计数) 10: 传感器模式(定时器计数) 11: 定时器模式
16	P3EN	R/W	2 相/ 3-相输入选择(传感器模式)(注 1) 0: 2-相输入 1: 3-相输入 设置输入信号数。
15	CMP	R	比较标志 0:- 1: 比较(由RD清除) 若执行比较, <CMP>设置为"1"。 通过读取数值, 就可清除标志。当设置<ENRUN> = "0"时, 应始终设置"0"。写入该位会无效。
14	REVERR	R	反向错误标志(传感器模式(在定时器计数时))(注 2) 0:- 1: 错误(由RD清除) 在传感器模式下(在定时器计数时), 当发生反向错误时, <REVERR>设置为"1"。通过读取数值, 就可清除标志。当设置<ENRUN> = "0"时, 应始终设置"0"。写入该位会无效。 在编码器模式, 传感器模式(事件计数)和定时器模式时, 该位无意义。
13	UD	R	旋转方向 0:CCW (用增量编码器, A相相对于B相具有 90 度相位超前) 1: CW (用增量编码器, A-相相对于B-相具有 90 度相位滞后) 当<ENRUN> = "0"时, <UD>设置为"0"。
12	ZDET	R	Z-相检测 0: 不选 1: Z相被检测到 在<ENRUN>从 0 写为 1后, <ZDET>在Z输入信号(ENCZ)的第一沿设置为 1。该设置在CW旋转时发生在信号Z的上升沿或者在CCW旋转时在Z的下降沿。 当<ENRUN> = "0"时, <ZDET>设置为"0"。 <ZEN>对<ZDET>值无影响。 在传感器事件计数和传感器定时器计数模式时, <ZDET>设置为"0"。 在传感器模式(事件计数)和传感器模式(定时器计数)时, 该位始终设置为"0"。

位	比特符号	类型	功能	
11	SFTCAP	W	<p>执行软件捕获(定时器模式/传感器模式(在定时器计数时))</p> <p>0: - 1: 软件捕获</p> <p>如果<SFTCAP>被设置为 1, 则编码器计数器的值会被捕获到ENCNT寄存器中。将"0"写入<SFTCAP>会无效。读取<SFTCAP>会始终返回到"0"。 在编码器和传感器事件计数模式时, <SFTCAP>无效; 将"1"写入该位会被忽略。</p>	
10	ENCLR	W	<p>编码器脉冲计数器的清除</p> <p>0: - 1: 清0</p> <p>将 1 写入<ENCLR>会将编码器计数器清除到"0"。一旦被清除, 编码器计数器从"0"开始重新计数。 将"0"写入<ENCLR>会无效。读取<ENCLR>会始终返回到"0"。</p>	
9	ZESEL	R/W	<p>ENCZ边沿选择(定时器模式)</p> <p>0: 上升沿 1: 下降沿</p> <p>在定时器模式时, 该位选择用作外触发的ENCZ的输入边沿。在其他模式时, 该位无意义。</p>	
8	CMPEN	R/W	<p>比较启用</p> <p>0: 禁用 1: 启用</p> <p>当"1"设置为<CMPEN>, 该位比较编码器计数器的计数值和ENINT的寄存器值。当"0"设置为<CMPEN>时, 该比较禁用。</p>	
7	ZEN	R/W	<p>Z相启用(编码器模式/定时器模式)</p> <p>0: 禁用 1: 启用</p> <p>在其他模式时, 该位无意义。</p>	
			<table border="1"> <tr> <td><编码器模式> 用ENCZ输入清除编码器计数器的设置</td> <td> <p>在已设置<ZEN> = "1"的情况下, 如果在顺时针方向旋转期间检测到ENCZ的上升沿, 则编码器计数器即被清为"0"。 如果在期间顺时针方向旋转期间检测到ENCZ的一个下降沿, 则编码器计数器即被清为"0"。 如果ENCLK的沿(乘以根据已解码A和B信号所导出的4时钟)和ENCZ的沿重合, 则编码器计数器即被清为"0", 无递增或递减(即清0优先)。</p> </td> </tr> <tr> <td><定时器模式> 设置ENCZ输入, 以用作外触发。</td> <td> <p>在<ZEN> = 1 时, 编码器计数器的值被捕获到ENINT寄存器中, 并在<ZESEL>所选择ENCZ的沿上清为"0"。</p> </td> </tr> </table>	<编码器模式> 用ENCZ输入清除编码器计数器的设置
<编码器模式> 用ENCZ输入清除编码器计数器的设置	<p>在已设置<ZEN> = "1"的情况下, 如果在顺时针方向旋转期间检测到ENCZ的上升沿, 则编码器计数器即被清为"0"。 如果在期间顺时针方向旋转期间检测到ENCZ的一个下降沿, 则编码器计数器即被清为"0"。 如果ENCLK的沿(乘以根据已解码A和B信号所导出的4时钟)和ENCZ的沿重合, 则编码器计数器即被清为"0", 无递增或递减(即清0优先)。</p>			
<定时器模式> 设置ENCZ输入, 以用作外触发。	<p>在<ZEN> = 1 时, 编码器计数器的值被捕获到ENINT寄存器中, 并在<ZESEL>所选择ENCZ的沿上清为"0"。</p>			
6	ENRUN	R/W	<p>编码器操作启用</p> <p>0: 禁用 1: 启用</p> <p>将<ENRUN>设置为 1 及将<ZDET>清除到"0"会启用编码器操作。将<ENRUN> 清除到"0"会禁用编码器操作。 当<ENRUN>位被清除到"0"时, 有被清除的和未被清除的计数器和标志。</p>	
5-4	NR[1:0]	R/W	<p>噪声滤波器</p> <p>00: 无滤波 01: 滤除比 31 / fsys窄的噪声脉冲 10: 滤除比 63 / fsys窄的噪声脉冲 11: 将窄于127/fsys的脉冲作为杂波滤除 数字噪声滤波器除去比<NR[1:0]>所选宽度更窄的脉冲。</p>	
3	INTEN	R/W	<p>编码器中断启用</p> <p>0: 禁用 1: 启用</p> <p><INTEN>启用或禁用ENC中断。 将<INTEN>设置为"1"会启用中断生成。将<INTEN>设置为"0"会禁用中断生成。</p>	

位	比特符号	类型	功能
2-0	ENDEV[2:0]	R/W	编码器脉冲分频系数 000: 除以1 100: 除以16 001: 除以 2 101: 除以32 010: 除以4 110: 除以64 011: 除以8 111: 除以128 设置编码器脉冲分频系数 编码器的频率除以<ENDEV[2:0]>规定的系数。该被除信号确定事件中中断的间隔时间。

注 1: 在编码器模式或定时器模式, <P3EN>必须设置为"0"。

注 2: 若要改变模式, 应先读取标志, 以进行清除。

操作模式有<MODE[1:0]>, <P3EN>和<ZEN>规定的 8 种模式。

操作模式的设置如下:

<MODE[1:0]>	<ZEN>	<P3EN>	输入引脚	模式
00	0	0	A, B	编码器模式
	1		A, B和Z	编码器模式(Z的使用)
01	0	0	U, V	传感器模式(事件计数, 2 相输入)
		1	U, V, W	传感器模式(事件计数, 3-相输入)
10	0	0	U, V	传感器模式(定时器计数, 2-相输入)
		1	U, V, W	传感器模式(定时器计数, 3-相 输入)
11	0	0	-	定时器模式
	1		Z	定时器模式(Z的使用)

以下为<ENRUN>的状态以及相应的信号。

计数器/标记	<ENRUN> = 0 (在复位后)	<ENRUN> = 1 (操作)	<ENRUN> = 0 (停止)	<ENRUN> = 0 对象标记/计数器清零程序
编码器计数器	0x000000	计数操作	在停止保持某一数值	软件清零 (<ENCLR> = 1 WR)
噪声滤波器计数器	0y0000000	向上计数运算	向上计数运算 (始终进行滤波)	仅复位
编码器脉冲分频计数器	0x00	向下计数运算	已停止并清零	当 <ENRUN> = "0"时进行清零
比较标志 <CMP>	0	在比较时设置值为"1"。 读取后清零。	已清零	当 <ENRUN> = "0"时进行清零
反相错误标记 <REVERR>	0	在出现错误时设置值为 "1"。 读取后清零。	已清零	当 <ENRUN> = "0"时进行清零
Z检测标记 <ZDET>	0	在检测到Z时设置值为 "1"。	已清零	当 <ENRUN> = "0"时进行清零
旋转方向数位 <UD>	0	根据旋转方向设置为 "0"/"1"。	已清零	当 <ENRUN> = "0"时进行清零

17.4.3 编码器计数器重载寄存器(ENRELOAD)

	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	RELOAD							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	RELOAD							
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	RELOAD[15:0]	R/W	<p>设置该编码器计数器周期(在被乘以 4 或 6 之后) 0x0000 ~ 0xFFFF</p> <p>使用Z-相: 设置一次旋转的计数脉冲数 未使用Z-相: 设置一次旋转的计数脉冲数减 1</p> <p><RELOAD[15:0]>确定编码器计数器周期乘以4。 如果该编码器计数器被配置为向上计数器, 则其会递增至<RELOAD[15:0]>中编程的值, 然后回绕到下一个ENCLK上的"0"上。若编码器计数器被配置为倒计数器, 则它递减到"0", 然后在下一ENCLK时, 重新加载<RELOAD[15:0]> 值。</p>

注 1: RELOAD寄存器仅在编码器模式下使用。

注 2: 应用 32 位指令访问ENRELOAD寄存器。

17.4.4 ENINT(编码器比较寄存器)

	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	INT							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	INT							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	INT							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能												
31-24	-	R	读作"0"。												
23-0	INT[23:0]	R/W	<p>计数器比较数值设置</p> <table border="1"> <tr> <td>编码器模式</td> <td>编码器脉冲位置的中断条件。 当 <CMPEN> 等于"1"时, 如果编码器计数器的数值与<INT[15:0]> 的数值相匹配, 则 <CMP> 的设置值为"1"。如 <INTEN> 已设置为"1", 则会发出一项中断请求(INTENC0)。然而, 若设置<ZEN> = "1", 则不会发生中断请求, 直到<ZDET> = "1"。</td> <td>0x0000 ~ 0xFFFF</td> </tr> <tr> <td>传感器模式: (事件计数)</td> <td>编码器脉冲位置的中断条件。 当 <CMPEN> 等于"1"时, 如果编码器计数器的数值与<INT[15:0]> 的数值相匹配, 则 <CMP> 的设置值为"1"。如 <INTEN> 已设置为"1", 则会发出一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。</td> <td>0x0000 ~ 0xFFFF</td> </tr> <tr> <td>传感器模式: (定时器计数)</td> <td>异常脉冲检测时间的中断条件 在被设置为<CMPEN> = "1"时, 且内部计数器值可匹配<INT[23:0]>的一个值, 则可求出异常脉冲检测时间误差, 且<CMP>即被设置为"1"。如 <INTEN> 已设置为"1", 则会发生一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。</td> <td>0x000000 ~ 0xFFFFFF</td> </tr> <tr> <td>定时器模式</td> <td>定时器比较的中断条件 在被设置为<CMPEN> = "1"时, 且内部计数器值可匹配<INT[23:0]>的一个值, 则可求出异常脉冲检测时间误差, 且<CMP>即被设置为"1"。如 <INTEN> 已设置为"1", 则会发生一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。</td> <td>0x000000 至 0xFFFFFF</td> </tr> </table>	编码器模式	编码器脉冲位置的中断条件。 当 <CMPEN> 等于"1"时, 如果编码器计数器的数值与<INT[15:0]> 的数值相匹配, 则 <CMP> 的设置值为"1"。如 <INTEN> 已设置为"1", 则会发出一项中断请求(INTENC0)。然而, 若设置<ZEN> = "1", 则不会发生中断请求, 直到<ZDET> = "1"。	0x0000 ~ 0xFFFF	传感器模式: (事件计数)	编码器脉冲位置的中断条件。 当 <CMPEN> 等于"1"时, 如果编码器计数器的数值与<INT[15:0]> 的数值相匹配, 则 <CMP> 的设置值为"1"。如 <INTEN> 已设置为"1", 则会发出一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。	0x0000 ~ 0xFFFF	传感器模式: (定时器计数)	异常脉冲检测时间的中断条件 在被设置为<CMPEN> = "1"时, 且内部计数器值可匹配<INT[23:0]>的一个值, 则可求出异常脉冲检测时间误差, 且<CMP>即被设置为"1"。如 <INTEN> 已设置为"1", 则会发生一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。	0x000000 ~ 0xFFFFFF	定时器模式	定时器比较的中断条件 在被设置为<CMPEN> = "1"时, 且内部计数器值可匹配<INT[23:0]>的一个值, 则可求出异常脉冲检测时间误差, 且<CMP>即被设置为"1"。如 <INTEN> 已设置为"1", 则会发生一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。	0x000000 至 0xFFFFFF
编码器模式	编码器脉冲位置的中断条件。 当 <CMPEN> 等于"1"时, 如果编码器计数器的数值与<INT[15:0]> 的数值相匹配, 则 <CMP> 的设置值为"1"。如 <INTEN> 已设置为"1", 则会发出一项中断请求(INTENC0)。然而, 若设置<ZEN> = "1", 则不会发生中断请求, 直到<ZDET> = "1"。	0x0000 ~ 0xFFFF													
传感器模式: (事件计数)	编码器脉冲位置的中断条件。 当 <CMPEN> 等于"1"时, 如果编码器计数器的数值与<INT[15:0]> 的数值相匹配, 则 <CMP> 的设置值为"1"。如 <INTEN> 已设置为"1", 则会发出一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。	0x0000 ~ 0xFFFF													
传感器模式: (定时器计数)	异常脉冲检测时间的中断条件 在被设置为<CMPEN> = "1"时, 且内部计数器值可匹配<INT[23:0]>的一个值, 则可求出异常脉冲检测时间误差, 且<CMP>即被设置为"1"。如 <INTEN> 已设置为"1", 则会发生一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。	0x000000 ~ 0xFFFFFF													
定时器模式	定时器比较的中断条件 在被设置为<CMPEN> = "1"时, 且内部计数器值可匹配<INT[23:0]>的一个值, 则可求出异常脉冲检测时间误差, 且<CMP>即被设置为"1"。如 <INTEN> 已设置为"1", 则会发生一项中断请求(INTENC0)。此数位对 <ZEN> 的数值无影响。	0x000000 至 0xFFFFFF													

注1: <INT[23:16]> 只有在传感器模式(定时器计数)和定时器模式下才能使用。

注2: 应用 32 位指令访问各ENINT寄存器。

17.4.5 ENCNT(编码器计数器)

	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	CNT							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	CNT							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CNT							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	功能																								
31-24	-	R	读作"0"。																								
23-0	CNT[23:0]	R/W	<p>编码器计数器/捕获值</p> <table border="1"> <tr> <td>编码器模式:</td> <td>编码器脉冲的计数器值</td> <td>0x0000 ~ 0xFFFF</td> </tr> <tr> <td></td> <td colspan="2"> 可读取编码器计数值。 编码器模式下，编码器计数器向上计数或倒数各编码器脉冲(ENCLK)。 在CW旋转期间，编码器计数器向上计数；在其达到<RELOAD[15:0]>的值时，其回绕到下一个ENCLK上的"0"上。 在CCW旋转期间，编码器计数器向下计数；在其达到"0"时，其连同<重新加载[15:0]>的值被重新加载到下一个ENCLK上。 </td> </tr> <tr> <td>传感器模式: (事件计数)</td> <td>编码器脉冲的计数器值</td> <td>0x0000 ~ 0xFFFF</td> </tr> <tr> <td></td> <td colspan="2"> 可读取编码器计数值。 传感器事件计数模式下，编码器计数器向上计数或倒数各编码器脉冲(ENCLK)。 在CW旋转时，编码器计数器向上计数；当它达到"0xFFFF"值时，它在下一ENCLK时回到"0"。 在CCW旋转时，编码器计数器倒数；当它达到"0"时，在下一ENCLK时回到"0xFFFF"。 </td> </tr> <tr> <td>传感器模式: (定时器计数)</td> <td>脉冲检测时间或软件捕获数值</td> <td>0x000000 ~ 0xFFFFFF</td> </tr> <tr> <td></td> <td colspan="2"> The value of 编码器计数器 can be read. 传感器模式下，通过将"1"写入<SFTCAP>，可读取各编码器脉冲(ENCLK)的编码器计数值，并由软件捕获。 经系统复位，捕获的值被清除到"0"。通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在传感器定时器计数模式下，编码器计数器被配置为用fsys向上计数的自由运行计数器。当检测到编码器脉冲(ENCLK)时，编码器计数器被清除到"0"。当它已达到"0xFFFFFF"时，自动回到"0"。 </td> </tr> <tr> <td>定时器模式</td> <td>内部计数器的捕获值或被软件捕获的值</td> <td>0x000000 ~ 0xFFFFFF</td> </tr> <tr> <td></td> <td colspan="2"> 通过将"1"写入到<SFTCAP>，编码器计数器的值可被软件读取和捕获。在<ZEN> = "1"时，编码器计数器的值也可被捕获到<ZESEL>所选Z沿上的 <CNT23:0>中。 通过复位将捕获的值清除到"0"。 通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在定时器模式时，编码器计数器被配置为一个用fsys向上计数的自由运行的计数器。当它已达到"0xFFFFFF"时，自动回到"0"。 </td> </tr> </table>	编码器模式:	编码器脉冲的计数器值	0x0000 ~ 0xFFFF		可读取编码器计数值。 编码器模式下，编码器计数器向上计数或倒数各编码器脉冲(ENCLK)。 在CW旋转期间，编码器计数器向上计数；在其达到<RELOAD[15:0]>的值时，其回绕到下一个ENCLK上的"0"上。 在CCW旋转期间，编码器计数器向下计数；在其达到"0"时，其连同<重新加载[15:0]>的值被重新加载到下一个ENCLK上。		传感器模式: (事件计数)	编码器脉冲的计数器值	0x0000 ~ 0xFFFF		可读取编码器计数值。 传感器事件计数模式下，编码器计数器向上计数或倒数各编码器脉冲(ENCLK)。 在CW旋转时，编码器计数器向上计数；当它达到"0xFFFF"值时，它在下一ENCLK时回到"0"。 在CCW旋转时，编码器计数器倒数；当它达到"0"时，在下一ENCLK时回到"0xFFFF"。		传感器模式: (定时器计数)	脉冲检测时间或软件捕获数值	0x000000 ~ 0xFFFFFF		The value of 编码器计数器 can be read. 传感器模式下，通过将"1"写入<SFTCAP>，可读取各编码器脉冲(ENCLK)的编码器计数值，并由软件捕获。 经系统复位，捕获的值被清除到"0"。通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在传感器定时器计数模式下，编码器计数器被配置为用fsys向上计数的自由运行计数器。当检测到编码器脉冲(ENCLK)时，编码器计数器被清除到"0"。当它已达到"0xFFFFFF"时，自动回到"0"。		定时器模式	内部计数器的捕获值或被软件捕获的值	0x000000 ~ 0xFFFFFF		通过将"1"写入到<SFTCAP>，编码器计数器的值可被软件读取和捕获。在<ZEN> = "1"时，编码器计数器的值也可被捕获到<ZESEL>所选Z沿上的 <CNT23:0>中。 通过复位将捕获的值清除到"0"。 通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在定时器模式时，编码器计数器被配置为一个用fsys向上计数的自由运行的计数器。当它已达到"0xFFFFFF"时，自动回到"0"。	
编码器模式:	编码器脉冲的计数器值	0x0000 ~ 0xFFFF																									
	可读取编码器计数值。 编码器模式下，编码器计数器向上计数或倒数各编码器脉冲(ENCLK)。 在CW旋转期间，编码器计数器向上计数；在其达到<RELOAD[15:0]>的值时，其回绕到下一个ENCLK上的"0"上。 在CCW旋转期间，编码器计数器向下计数；在其达到"0"时，其连同<重新加载[15:0]>的值被重新加载到下一个ENCLK上。																										
传感器模式: (事件计数)	编码器脉冲的计数器值	0x0000 ~ 0xFFFF																									
	可读取编码器计数值。 传感器事件计数模式下，编码器计数器向上计数或倒数各编码器脉冲(ENCLK)。 在CW旋转时，编码器计数器向上计数；当它达到"0xFFFF"值时，它在下一ENCLK时回到"0"。 在CCW旋转时，编码器计数器倒数；当它达到"0"时，在下一ENCLK时回到"0xFFFF"。																										
传感器模式: (定时器计数)	脉冲检测时间或软件捕获数值	0x000000 ~ 0xFFFFFF																									
	The value of 编码器计数器 can be read. 传感器模式下，通过将"1"写入<SFTCAP>，可读取各编码器脉冲(ENCLK)的编码器计数值，并由软件捕获。 经系统复位，捕获的值被清除到"0"。通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在传感器定时器计数模式下，编码器计数器被配置为用fsys向上计数的自由运行计数器。当检测到编码器脉冲(ENCLK)时，编码器计数器被清除到"0"。当它已达到"0xFFFFFF"时，自动回到"0"。																										
定时器模式	内部计数器的捕获值或被软件捕获的值	0x000000 ~ 0xFFFFFF																									
	通过将"1"写入到<SFTCAP>，编码器计数器的值可被软件读取和捕获。在<ZEN> = "1"时，编码器计数器的值也可被捕获到<ZESEL>所选Z沿上的 <CNT23:0>中。 通过复位将捕获的值清除到"0"。 通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在定时器模式时，编码器计数器被配置为一个用fsys向上计数的自由运行的计数器。当它已达到"0xFFFFFF"时，自动回到"0"。																										

注 1: <CNT[23:16]> 只有在传感器模式(定时器计数)或定时器模式下才能使用。在编码器模式或传感器模式(事件计数)，始终读作"0"。

注 2: 应用 32 位指令访问ENCNT寄存器。

17.5 操作说明

17.5.1 编码器模式

高速定位传感器确定AB编码器和ABZ编码器的相位输入。

事件检测 (旋转脉冲) → 中断生成
事件计数 → 匹配检测中断生成 (测量传输量)
检测旋转方向
向上/向下计数(可在运行中变化)
可设置的计数器周期

17.5.2 传感器模式

低速定位传感器确定(零交叉确定)UV霍尔传感器和UVW 霍尔传感器的相位输入。
有两类传感器模式，例如事件计数模式和定时器计数模式(用f_{sys}计数)。

17.5.2.1 事件计数模式

- 事件检测 (旋转脉冲) → 中断生成
- 事件计数 → 匹配中断发生(测量输送量)
- 旋转方向检测

17.5.2.2 定时器计数模式

事件检测 (旋转脉冲) → 中断生成
定时器计数
旋转方向检测
捕获功能 → 事件捕获(测量事件时间间隔) → 中断生成软件捕获
异常检测时间错误(定时器比较) → 匹配检测中断生成
逆检测错误 → 旋转方向改变所引起的错误标志

17.5.3 定时器模式

该模式能用作通用的 24 位定时器。

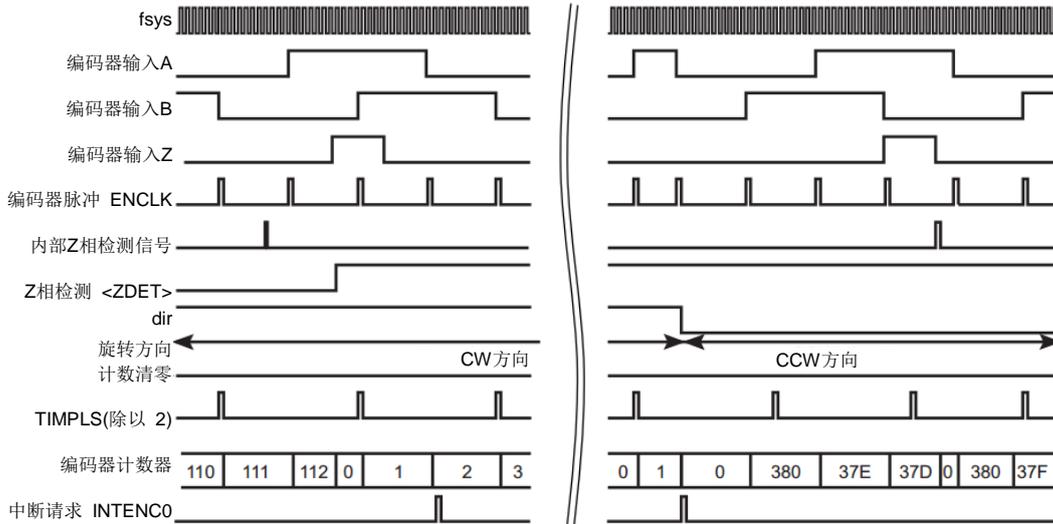
24 位向上计数器
计数器清除控制 (软件清除, 定时器清除, 外部触发和自由运行计数)
比较功能 → 匹配检测中断生成
捕获功能 → 外部触发捕获 → 中断生成软件捕获

17.6 功能

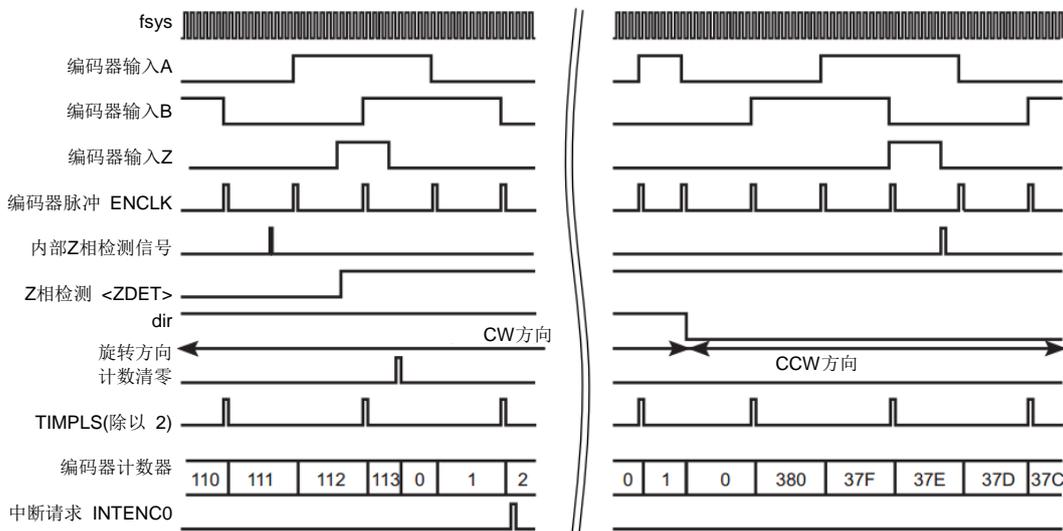
17.6.1 模式运算要点

17.6.1.1 编码器模式

1. If $\langle ZEN \rangle = 1$ ($\langle RELOAD \rangle = 0x0380$, $\langle ENINT \rangle = 0x0002$)



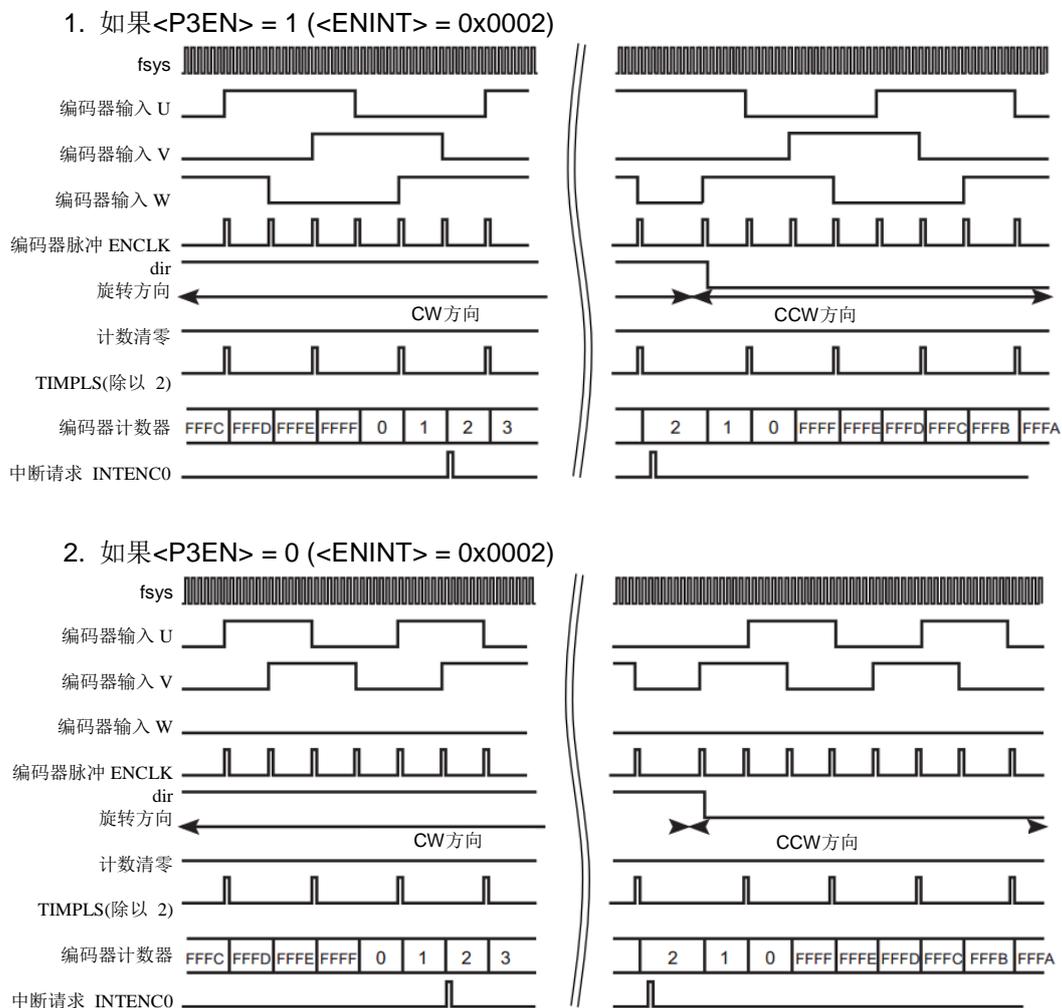
2. 如果 $\langle ZEN \rangle = 0$ ($\langle RELOAD \rangle = 0x0380$, $\langle ENINT \rangle = 0x0002$)



- MCU的增量式编码器输入应当连接至A通道，B通道以及Z通道。编码器计数器会对ENCLK脉冲进行计数，其数值为利用A相正交解码信号和B相正交解码信号所导出的时钟脉冲乘以4倍。
- 在CW旋转的过程中(即A相相对于B相存在 90 度相位超前时)，编码器计数器会进行向上计数。如果其已达到 $\langle RELOAD15:0 \rangle$ 所设置的数值，则会在下一ENCLK返回至"0"。
- 在CCW旋转的过程中(即A相相对于B相存在 90 度相位滞后时)，编码器计数器会进行向下计数。如果其已达到"0x0000"，则会在下一ENCLK重新载入 $\langle RELOAD \rangle$ 所设置的数值。

- 此外，在 $\langle ZEN \rangle = "1"$ 时，编码器计数器会在CW旋转期间在Z的上升沿上，以及在CCW旋转期间(在内部Z_检测到的时序时)在Z的下降沿上，被清除到"0"。若ENCLK边沿与Z边沿匹配，则编码器计数器在无增量或减量的情况下被清除到"0"。
- 当 $\langle ENCLR \rangle$ 设置为 1 时，编码器计数器将被清除到"0"。
- 在顺时针旋转过程中， $\langle UD \rangle$ 的设置值为 1；而在逆时针旋转过程中， $\langle UD \rangle$ 会被清除到"0"。
- 通过除以一个程序性因数而导出的TIMPLS，可被从外部清除。
- 如 $\langle CMPEN \rangle$ 已设置为"1"，则会在编码器计数器的数值达到 $\langle ENINT \rangle$ 所设置的数值时生成中断。但如 $\langle ZEN \rangle = "1"$ ，则不会在 $\langle ZDET \rangle = "0"$ 时生成中断。
- 当 $\langle ENRUN \rangle$ 和 $\langle UD \rangle$ 均设置为"0"时， $\langle ENRUN \rangle$ 会被清除到"0"。

17.6.1.2 传感器模式(事件计数)

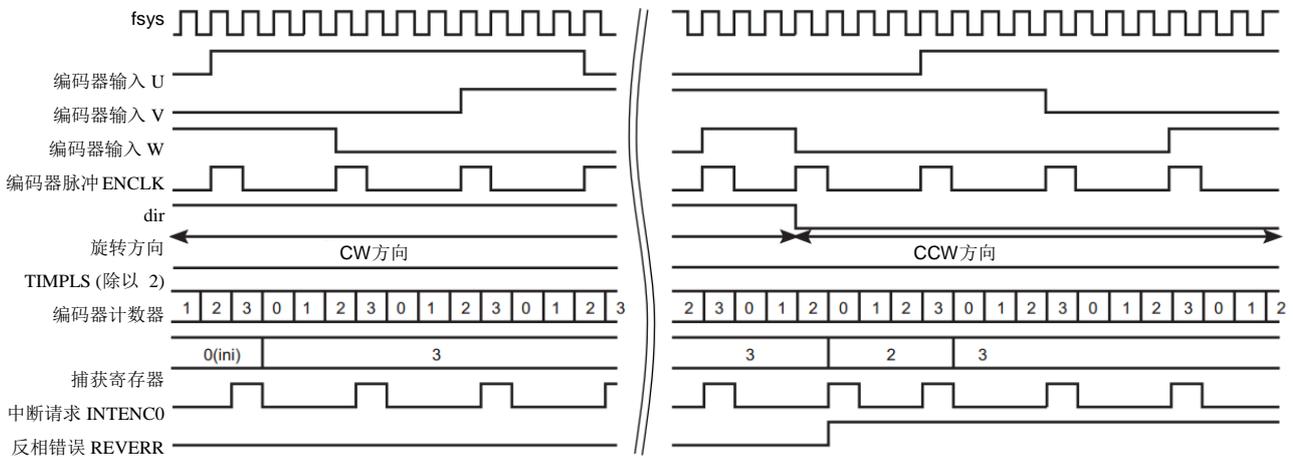


- 微处理器单元(MCU)的霍尔传感器输入应当连接至U通道，V通道以及W通道。编码器计数器对ENCLK脉冲计数，(当 $\langle P3EN \rangle = "0"$ 时)其乘以由解码的U，V信号推导的4个时钟脉冲，或者(当 $\langle P3EN \rangle = "1"$ 时)其乘以由解码的U，V，W信号推导的6个时钟脉冲。
- 在顺时针旋转的过程中(即U通道相对于V通道存在 90 度相位超前时；以及V通道相对于W通道存在 90 度相位超前时)，编码器计数器会进行向上计数。如果其已达到"0xFFFF"，则会在下一ENCLK返回至"0"。

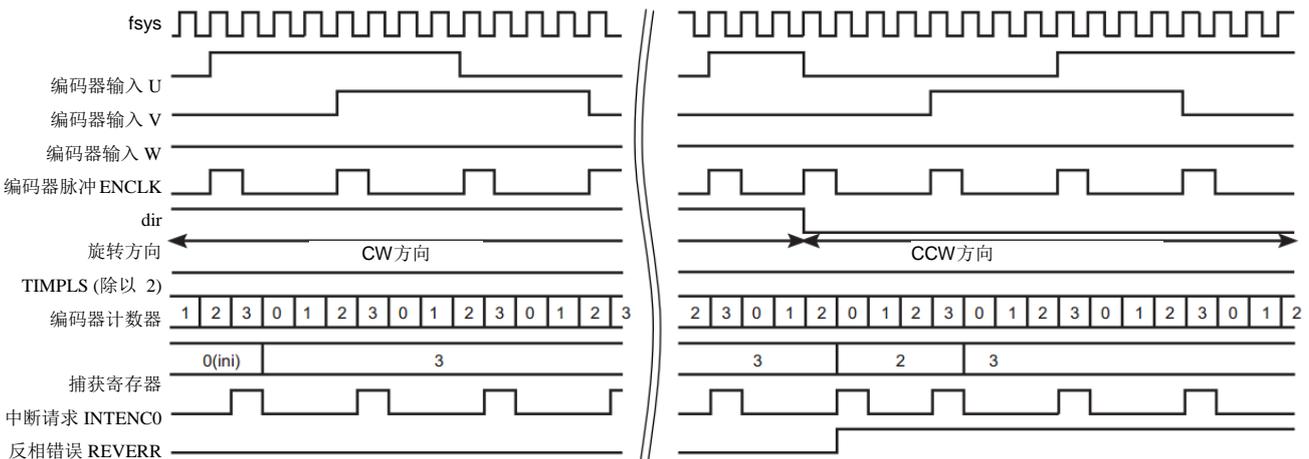
- 在逆时针旋转的过程中(即U通道相对于V通道存在 90 度相位滞后时; 以及V通道相对于W通道存在 90 度相位滞后时), 编码器计数器会进行向下计数。如果其已达到"0x0000", 则会在下一ENCLK返回至"0xFFFF"。
- 当 <ENCLR>设置为 1 时, 内部计数器将被清除到"0"。
- 在顺时针旋转过程中, <UD>的设置值为 1; 而在逆时针旋转过程中, <UD>会被清除到"0"。
- 通过除以一个程序性因数而导出的TIMPLS, 可被从外部清除。
- 如 <CMPEN> 已设置为 1, 则会在内部计数器的数值达到 <ENINT> 所设置的数值时生成中断。
- 当 <UD> 和 <ENRUN> 均设置为"0"时, <UD> 会被清除到"0"。

17.6.1.3 传感器模式(定时器计数)

1. 如果 <P3EN> = 1 (<ENINT> = 0x0002)



2. 如果 <P3EN> = 0 (<ENINT> = 0x0002)

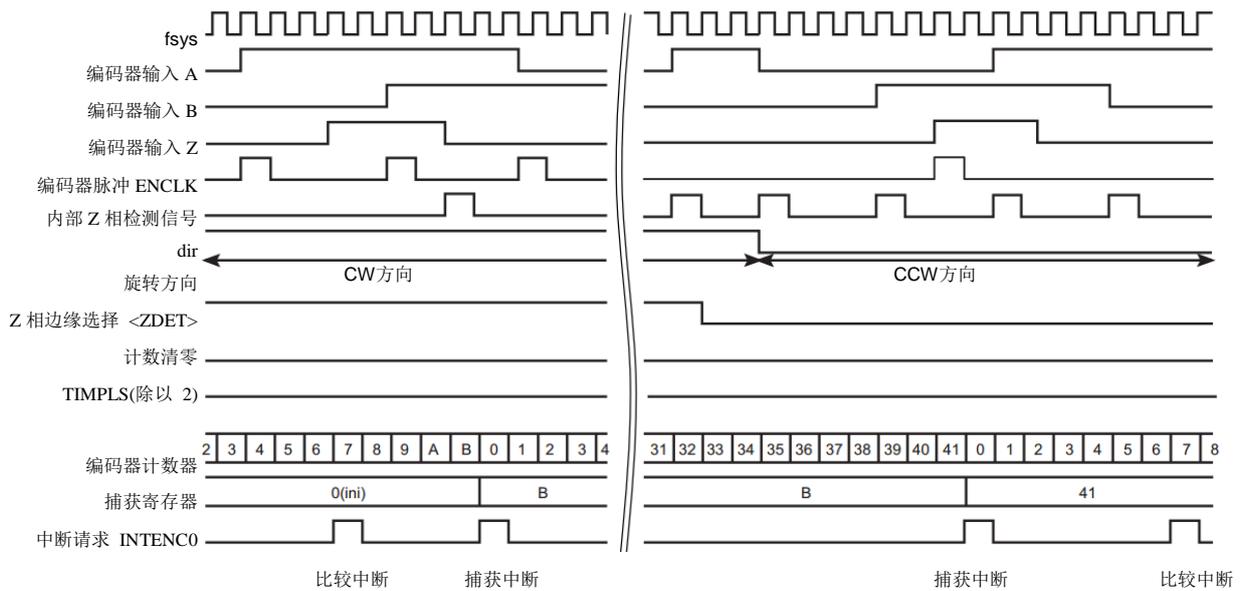


- 在传感器定时器计数模式下, 微处理器单元(MCU)的霍尔传感器输入应当连接至U通道, V通道以及W通道。编码器计数器测量ENCLK的两个相邻脉冲之间的间隔时间, 其被乘以从已被解码的U和V信号导出的 4 个时钟 (在<P3EN> = "0"时), 或被乘以从已被解码的U, V和W信号导出的 6 个时钟脉冲(在 <P3EN> = "1"时)。

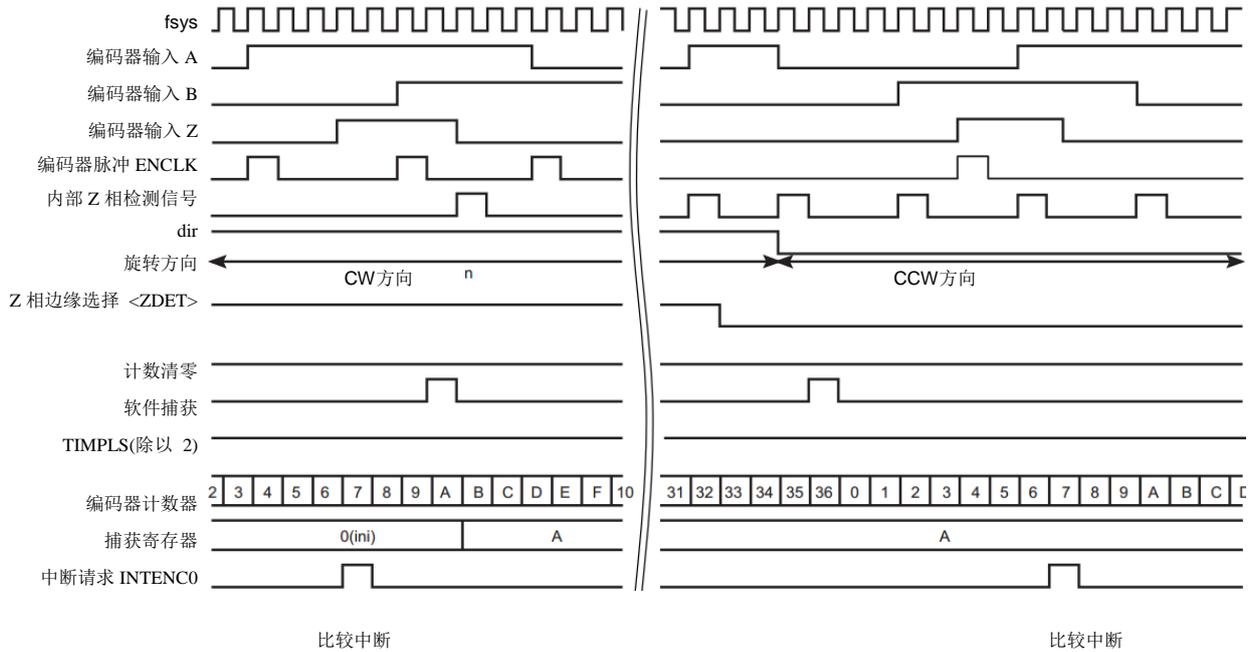
- 编码器计数器始终向上计数；它在ENCLK时被清除到"0"。如果此编码器计数器已达到"0xFFFFFFFF"，则会返回至"0"。
- 当 <ENCLR>设置为 1时，编码器计数器将被清除到"0"。
- ENCNT 可将编码器计数器的数值捕获至ENCNT寄存器中。捕获的计数器数值可以从ENCNT中进行读取。
- 将软件捕获数位 <SFTCAP> 设置为 1，则编码器计数器的数值将被捕获至ENCNT寄存器中。此捕获操作可以在任何时候进行。捕获的计数器数值可以从ENCNT中进行读取。
- 在顺时针旋转过程中，<UD>的设置值为 1；而在逆时针旋转过程中，<UD>会被清除到"0"。
- 如 <CMPEN> 已设置为 1，则会在编码器计数器的数值达到 <ENINT> 所设置的数值时生成中断。
- 当 <ENRUN> 设置为"0"时，<UD> 会被清除到"0"。
- 当旋转方向发生改变时，<REVERR> 的设置值将变为 1。读取后此数位将被清除到"0"。
- ENCNT寄存器的值 (捕获值)被保留，不考虑<ENRUN>的值。ENCNT寄存器只能在重设后才能清0。

17.6.1.4 定时器模式

1. 如果<ZEN> = 1 (<ENINT> = 0x0006)



2. 如果<ZEN> = 0 (<ENINT> = 0x0006)



在<ZEN> = "1"时，Z输入引脚被用作外部触发。当<ZEN> = "0"时，外部输入不用于触发定时器。此编码器计数器将始终进行向上计数。若<ZEN> = "1"，当<ZESEL>设置为"0"时，计数器在Z的上升沿被清除到"0"，当<ZESEL>设置为"1"时，计数器在Z的下降沿被清除到"0"。如果此编码器计数器已达到"0xFFFFF"，则会返回至"0"。

当 <ENCLR> 设置为 1 时，编码器计数器将被清除到"0"。

Z 相检测可将编码器计数器的数值捕获至ENCNT寄存器中。捕获的计数器数值可以从ENCNT中进行读取。

将软件捕获数位<SFTCAP>设置为"1"，则编码器计数器的数值将被捕获至ENCNT寄存器中。此捕获操作可以在任何时候进行。捕获的计数器数值可以从ENCNT中进行读取。

在CW旋转过程中，<UD>的设置值为 1；而在CCW旋转过程中，<UD>会被清除到"0"。

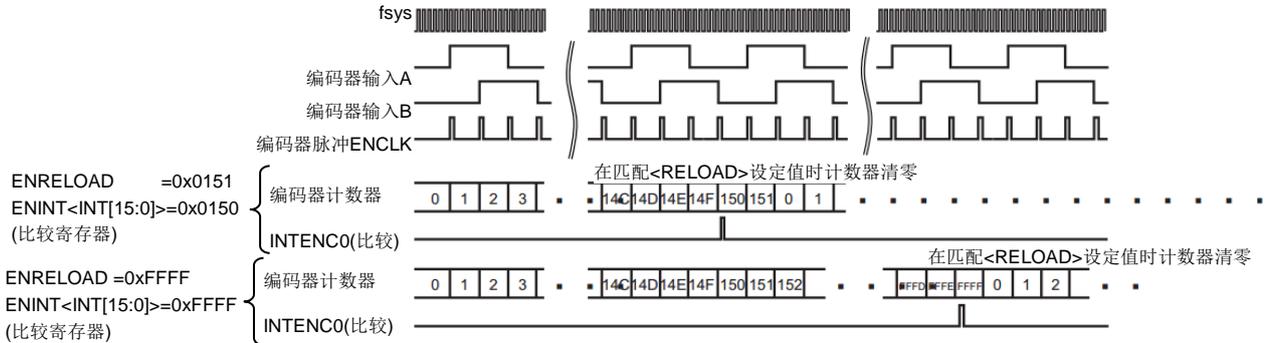
如 <CMPEN> 已设置为 1，则会在编码器计数器的数值达到<ENINT>所设置的数值时生成中断。

当 <ENRUN> 设置为"0"时，<UD> 会被清除到"0"。

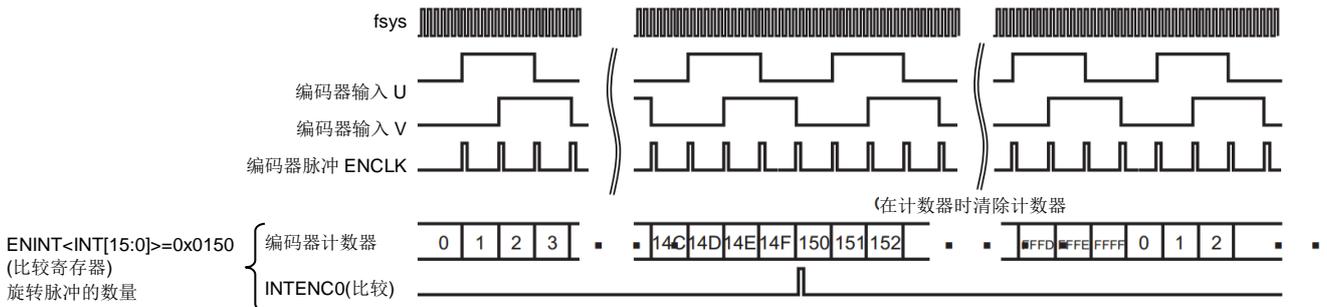
ENCNT寄存器的值 (捕获值)被保留，不考虑<ENRUN>的值。ENCNT寄存器只能在重设后才能清0。

17.6.2 <CMPEN >=1 时的计数器和中断生成运算

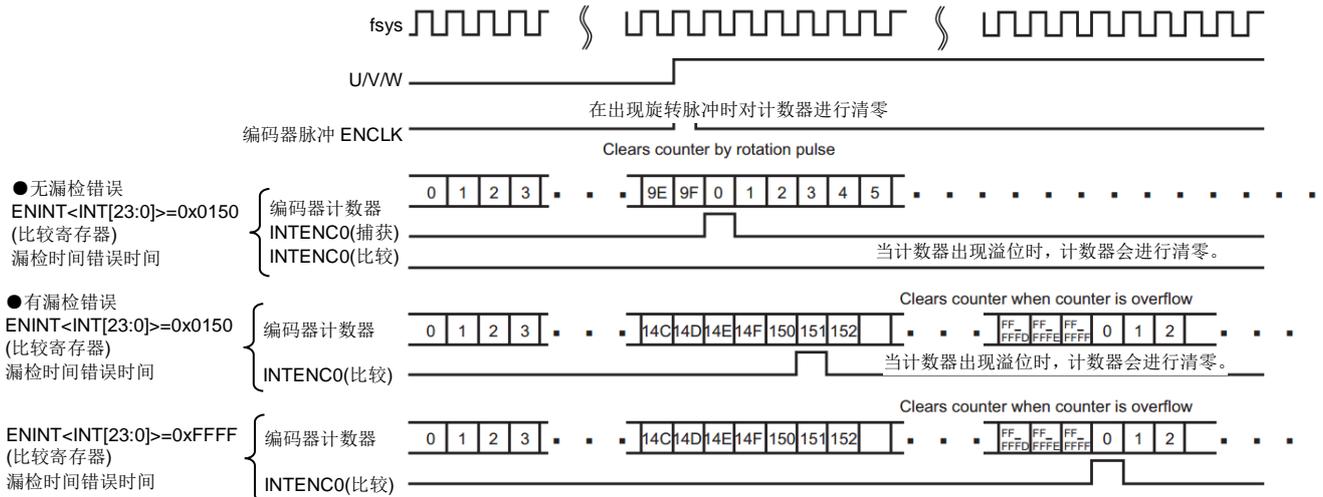
17.6.2.1 编码器模式



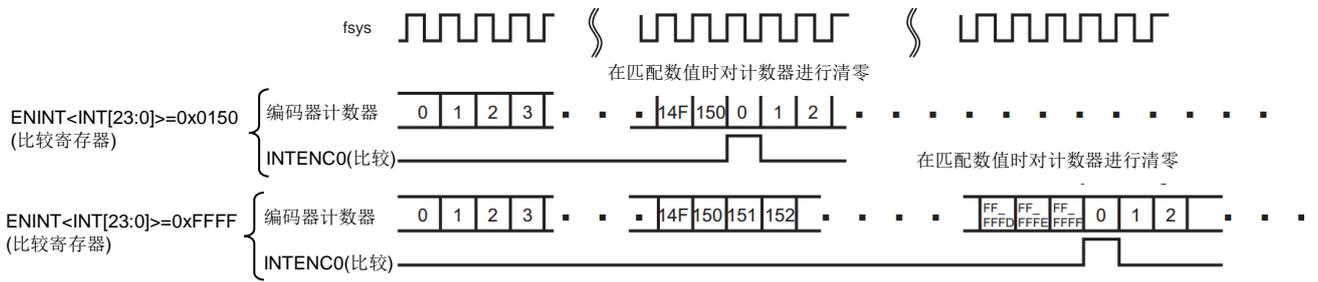
17.6.2.2 传感器模式(事件计数)



17.6.2.3 传感器模式(定时器计数)



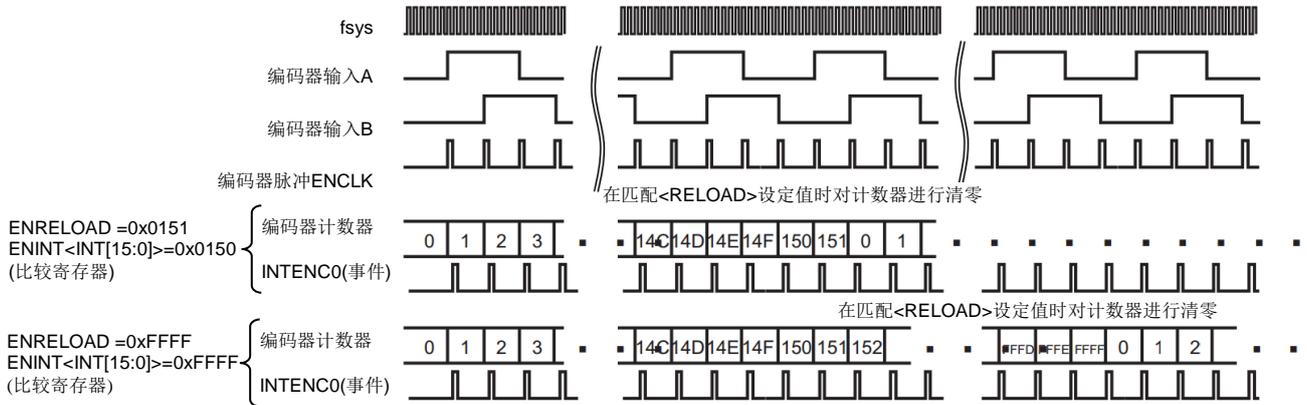
17.6.2.4 定时器模式



17.6.3 <CMPEN >= 0 时的计数器和中断生成运算

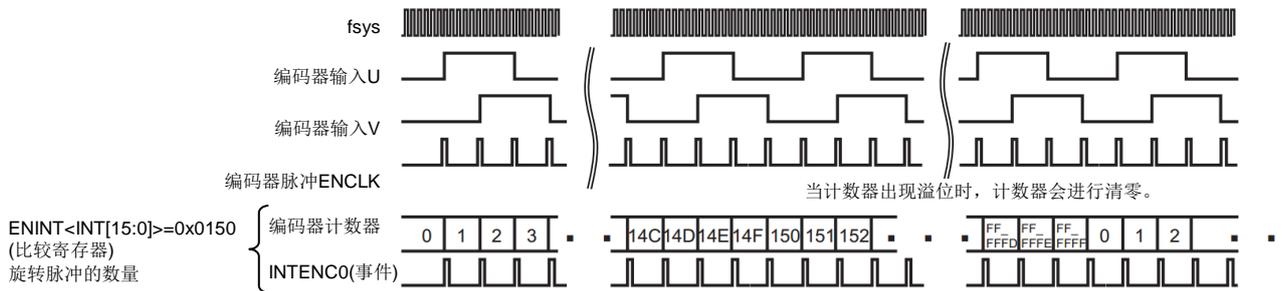
17.6.3.1 编码器模式

<ENDEV>="000"

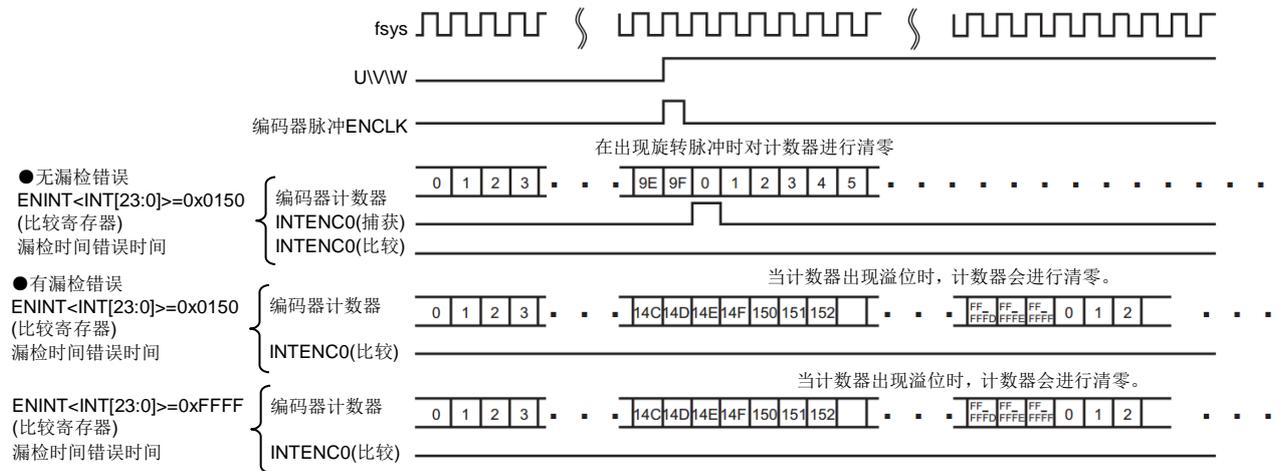


17.6.3.2 传感器模式(事件计数)

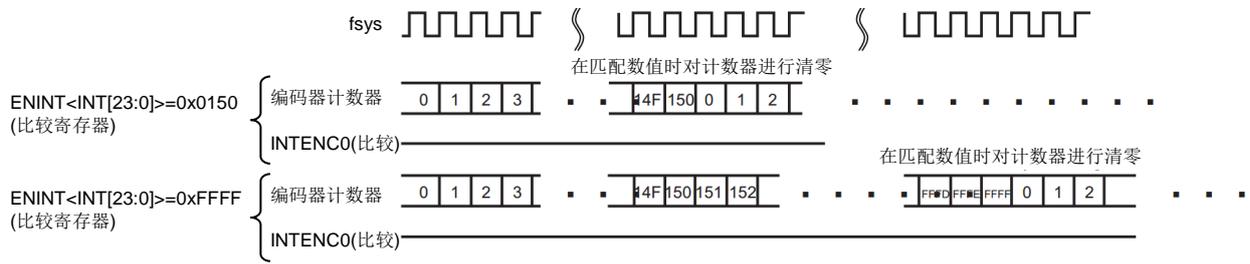
<ENDEV>="000"



17.6.3.3 传感器模式(定时器计数)



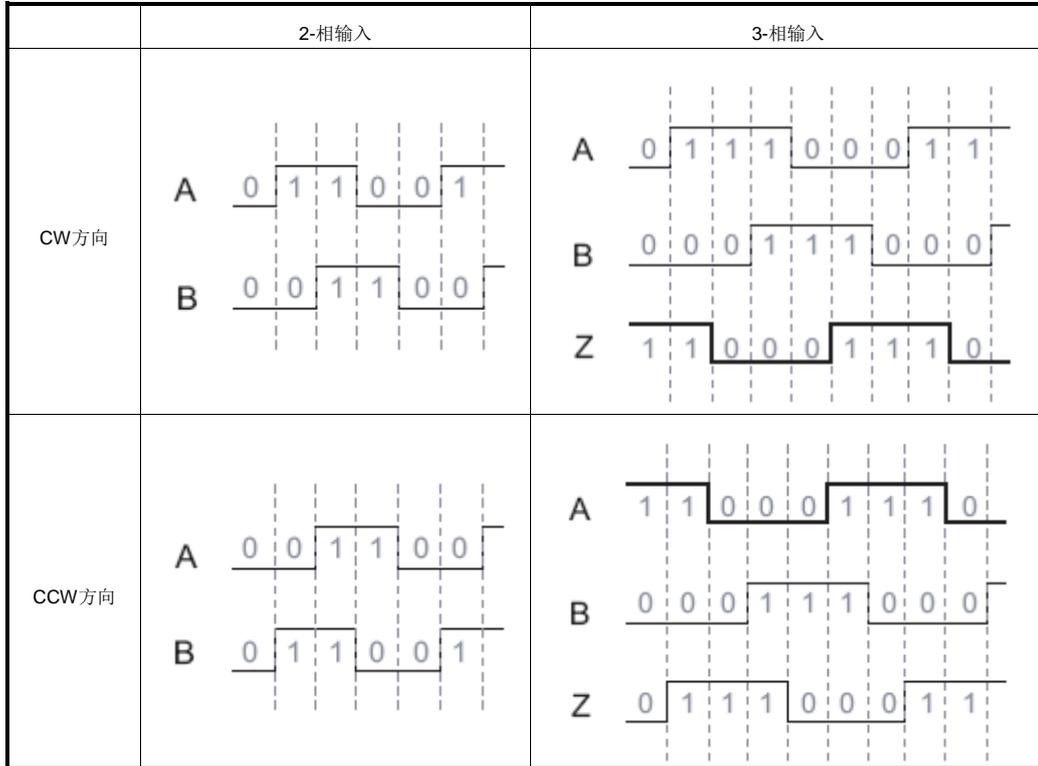
17.6.3.4 定时器模式



17.6.4 旋转方向检测

此电路旨在测定A-相，B-相以及Z-相中的其中一个相位。

其通常可用于两相输入(a-相和b-相)以及 3-相输入(a-相， b-相和z-相)。在使用 3-相输入时，应将 <P3EN>设置为"1"。



17.6.5 计数器电路

计数器电路拥有一个 24 位向上/向下计数器。

17.6.5.1 操作说明

见表 17-2 中的相关说明，可根据不同的操作模式对计数，清0和重载进行控制。

表 17-2 计数器控制

模式 <MODE[1:0]>	<ZEN>	<P3EN>	输入引脚	计数	运算	计数器清除条件	计数器重新加载条件	计数器的工 作范围 (重新加载值)	
编码器模式 00	0	0	A, B	编码器 脉冲 (ENCLK)	UP	[1]<ENCLR> = 1 WR [2] 与<RE-LOAD>匹配	-	0x0000 ~ <RE-LOAD>	
					DOWN	[1]<ENCLR> = 1 WR	[1] 与0x0000 匹配		
	1		A, B, Z		UP	[1]<ENCLR> = 1 WR [2] 与<RE-LOAD>匹配 [3] Z-相触发	-		
					DOWN	[1]<ENCLR> = 1 WR	[1] 与0x0000 匹配		
传感器模式 (事件计数) 01	0	0	U, V		UP	[1]<ENCLR> = 1 WR [2] 与 0xFFFF匹配	-		0x0000 ~ 0xFFFF
					DOWN	[1]<ENCLR> = 1 WR	[1] 与 0x0000 匹配		
	1	U, V, W	UP		[1]<ENCLR> = 1 WR [2] 与 0xFFFF匹配	-			
			DOWN		[1]<ENCLR> = 1 WR	[1] 与 0x0000 匹配			
传感器模式 (定时器计数) 10	0	0	U, V	fsys	UP	[1]<ENCLR> = 1 WR	-	0x000000 ~ 0xFFFFFFFF	
		1	U, V, W		UP	[2] 与 0xFFFFFFFF匹配 [3] 编码器脉冲(ENCLK)	-		
定时器模式 11	0	x	-	fsys	UP	[1]<ENCLR> = 1 WR [2] 与 0xFFFFFFFF匹配 [3]可匹配<ENINT>	-	0x000000 ~ 0xFFFFFFFF	
	1		Z		UP	[1]<ENCLR> = 1 WR [2] 与 0xFFFFFFFF匹配 [3] Matches with <ENINT> [4] Z-触发	-		

注：在将 <ENRUN> 设置为"0"时，重计数器的数值不会被清除到"0"。如此时再次将 <ENRUN> 设置为"1"，则计数器将重新从其此前停止的计数器数值开始计数。如需清除计数器数值，可将 <ENCLR> 设置为"1"以执行软件清零。

17.6.6 中断

中断由四种中断组成，即事件中断(分频脉冲和捕获)，异常检测时间中断，定时器比较中断，捕获中断。

17.6.6.1 操作性说明

当设置<INTEN> = "1"时，会发生计数器值和编码器脉冲产生的中断。

中断因数设置由带有运算模式和<CMPEN>及<ZEN>设置的六类设置组成。表 17-3 给出了这些中断因数。

表 17-3 中断因数

中断因数	说明	模式	中断输出	状态标志
1 事件计数中断	在<CMPEN> = 1 时，编码器计数器可对事件进行计数(编码器脉冲)。在其达到<ENINT>中的编程值时，发生一次中断。	编码器模式 和 传感器模式	<INTEN> = 1 和 <CMPEN> = 1	<CMP>
2 事件中断 (分频脉冲)	在各分频的时钟脉冲(1 ~ 128 分频)(用<ENDEV>中编程的系数除编码器脉冲而得)，发生中断。	传感器模式 (事件计数)	<INTEN> = 1	不可用
3 事件中断 捕获中断	发生一次中断，表示发生了一个事件(编码器脉冲)，导致计数器值在脉冲时序上被捕获到。	传感器模式 (定时器计数)	<INTEN> = 1	不可用
4 异常检测时间错误中断	在<CMPEN> = 1 时，ENC使用用fsys进行向上计数的计数器，并被某事件(编码器脉冲)清除如果在<ENINT>中所编程的一段时期内无事件发生，则发生一次中断。	传感器模式 (定时器计数)	<INTEN> = 1 和 <CMPEN> = 1	<CMP>
5 定时器比较中断	在<CMPEN> = 1 时，在定时器达到<ENINT>中所编程的值时，发生一次中断。	定时器模式	<INTEN> = 1 和 <CMPEN> = 1	<CMP>
6 捕获中断	在已在外部触发(Z输入)上捕获到计数器值时，发生一次中断。		<INTEN> = 1	不可用

在传感器定时器计数模式以及定时器模式下，可将编码器计数器的数值捕获至 ENCNT 寄存器中。

捕获的计数器数值可以从 ENCNT 寄存器中进行读取。

在传感器定时器计数模式下，可根据某一事件(编码器脉冲)的出现将编码器计数器的数值捕获至 ENCNT 寄存器中。用软件将 1 写入<SFTCAP>，计数器值也能被捕获。

在定时器模式下，将 <SFTCAP> 设置为 1 可以通过软件对计数器的数值进行捕获。如 <ZEN> 已设置为 1，则在出现根据 <ZESEL> 所选定的Z相信号输入的的边缘时，也可由外部触发对计数器的数值进行捕获。

18. 上电复位电路 (POR)

上电复位电路(POR)会在接通电源时产生一个上电复位信号。电源电压表示为DVDD5B。

18.1 结构

上电复位电路由基准电压生成电路，比较器，VLTD复位电路以及上电计数器组成。

此电路可对梯形电阻器所分的电压与在比较器的基准电压生成电路中所产生的基准电压进行比较。

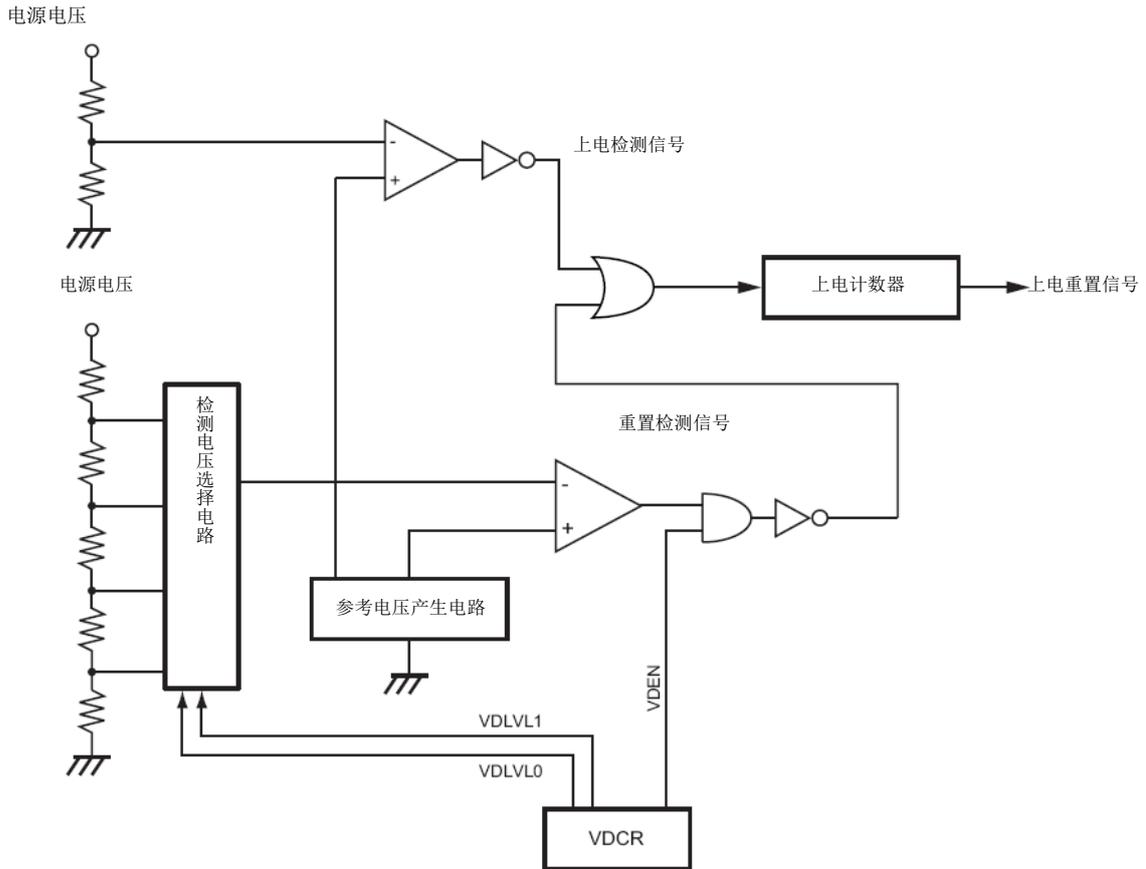


图 18-1 上电复位电路

有关VLTD复位电路中VDCR的详细数据，见"电压检测电路(VLTD)"一节。

18.2 功能

在接通电源时，如果电源电压低于释放电压，则会产生一个上电检测信号。当DVDD5B 超过 $3.0 \pm 0.2 V$ 时，即会释放上电检测信号。

上电计数器会在释放上电检测信号以及复位检测信号后开始工作。在等待时间(约 3.2 ms)过去后，系统则会释放上电复位信号。

在产生上电复位信号的同时，CPU以及外围功能将进行复位。

如果未使用复位输入，则必须将供电电压升高至建议的工作电压范围，直至完成上电复位释放。如果电源电压在这一时间内未能达到建议的工作电压范围，则TMPM375FSDMG将无法正常工作。

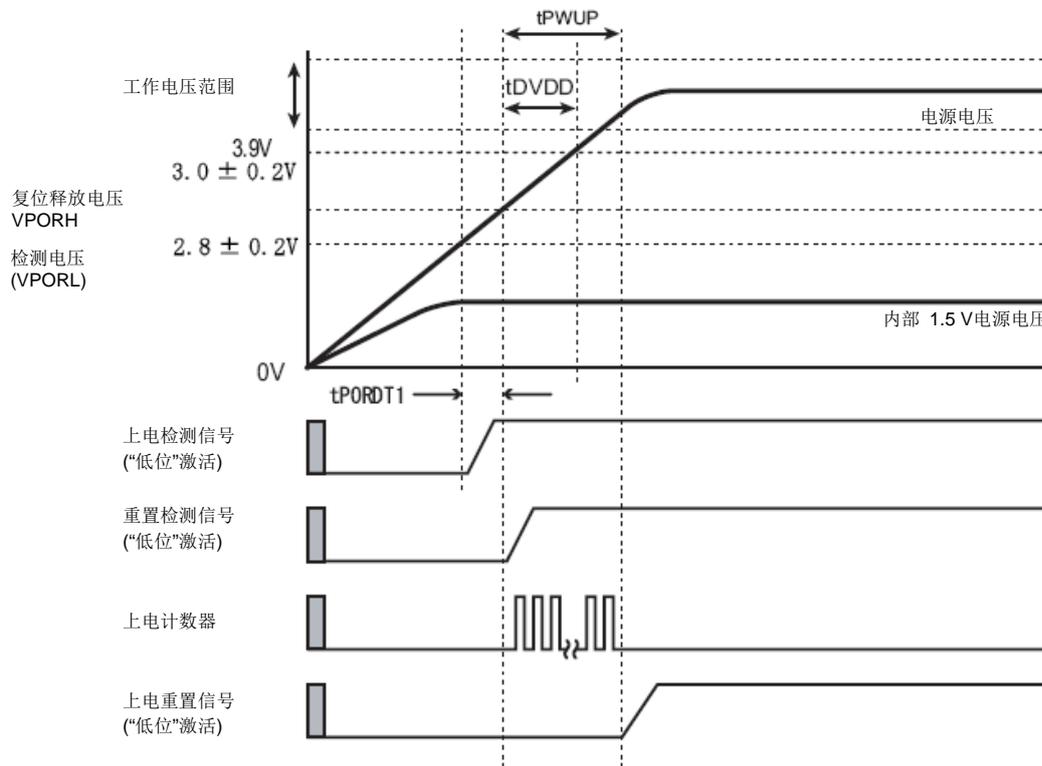


图 18-2 上电复位电路时序

符号	参数	最小值	典型值	最大值	单元
tPWUP	上电计数器	-	$2^{15}/f_{osc2}$	-	s
tDVDD	电源线的上升时间	-	-	3	ms
VPORH	上电复位释放电压	2.8	3	3.2	V
VPORL	上电复位检测电压	2.6	2.8	3.0	V
tPORDT1	上电复位解除响应时间		30		μs

注：由于上电释放电压以及上电复位检测电压会出现相对变化，因此检测电压永远不会发生逆转。

注：上电复位电路可因电源电压波动的影响而不能正确工作。用户应根据装置设计的电气特性，予以适当考虑。

19. 低压检测电路 (VLTD)

低电压检测电路通过检测下降电压，生成一个复位信号。

注：由于电源电压波动，电压检测电路可能不会正确运行。用户应根据装置设计的电气特性，予以适当考虑。

19.1 结构

低电压检测电路由一个参考电压产生电路，一个检测电压电平选择电路，一个比较器和若干控制寄存器组成。

电源电压(DVDD5B)被除以电阻梯，并被输入到检测电压选择电路中。检测电压选择电路按照规定的检测电压(VDLVL)选择电压，比较器将它与参考电压进行比较。在电源电压 (DVDD5B)低于检测电压(VDLVL)时，一个电压检测复位信号即被生成。

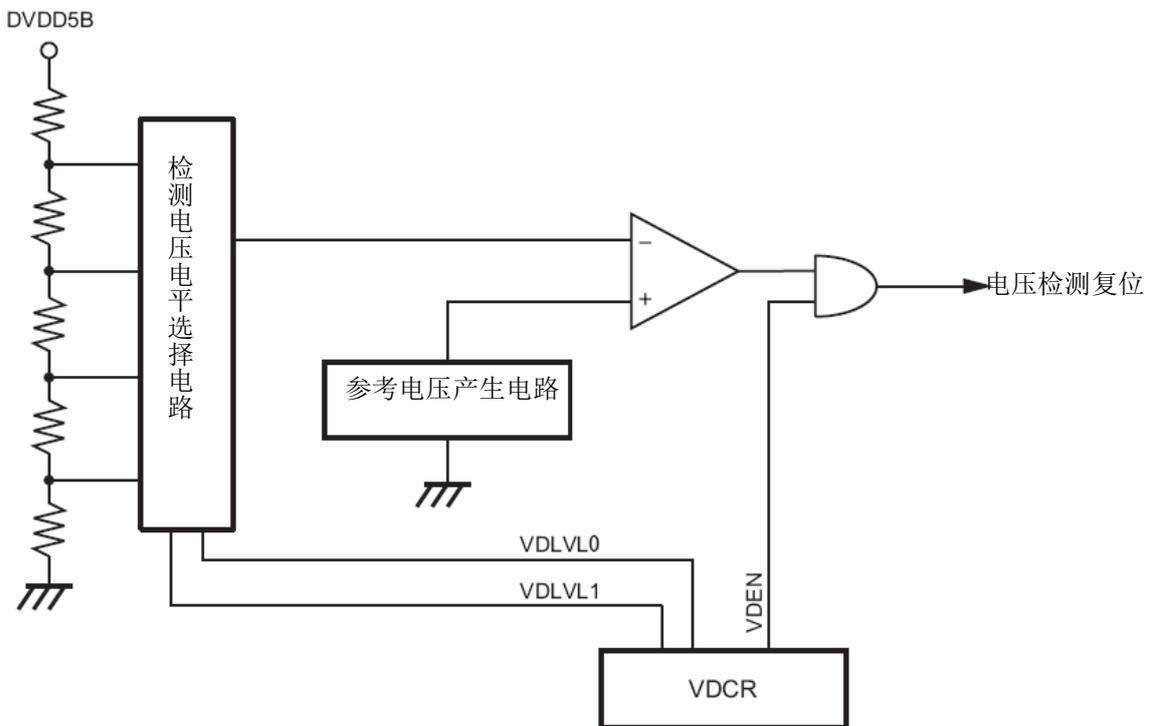


图 19-1 电压检测电路

19.2 寄存器

19.2.1 寄存器列表

基址 = 0x4004_0900

寄存器名称	地址(基+)
电压检测控制寄存器	VDCR 0x0000

19.2.2 电压检测控制寄存器(VDCR)

	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	-	-	-	-	-	VDLVL		VDEN
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"。
2-1	VDLVL[1:0]	R/W	检测电压的选择 00: 保留 01: $4.1 \pm 0.2 V$ 10: $4.4 \pm 0.2 V$ 11: $4.6 \pm 0.2 V$
0	VDEN	R/W	低电压检测运算 0: 禁用 1: 启用

注: VDCR 可通过上电复位或外部复位输入进行初始化。

19.3 操作说明

19.3.1 控制

电压检测电路受电压检测控制寄存器控制。

19.3.2 功能

检测电压可由VDCR<VDLVL[1:0]>选择。电压检测的启用/禁用可用VDCR<VDEN>进行编程。在电压检测运算被启用之后，在电源电压 (DVDD5B) 低于检测电压<VDLVL [1:0]>时，一个电压检测复位信号即被生成。

19.3.2.1 启用/禁用电压检测

将VDCR<VDEN> 设置为"1"，将启用电压检测运行操作功能。将其设置为"0"，将禁用电压检测操作。在通过外部复位输入实现的上电复位或复位被解除之后，VDCR<VDEN>即被清除为"0"。

注：在电源电压 (DVDD5B)低于检测电压VDCR<VDLVL[1:0]>时，将VDCR<VDEN>设置为"1"即可生成复位信号。

19.3.2.2 选择检测电压电平

选择VDCR<VDLVL[1:0]>时的检测电压。

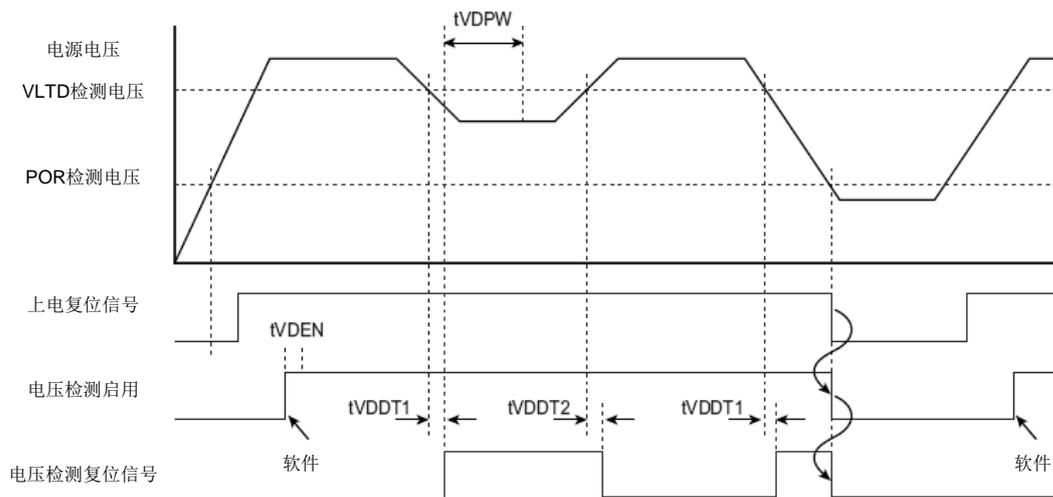


图 19-2 电压检测时序

符号	参数	最小值	典型值	最大值	单元
tVDEN	在启用电压检测后的设置时间	-	40	-	μs
tVDDT1	电压检测响应时间	-	40	-	
tVDDT2	电压检测解除时间	-	40	-	
tVDPW	电压检测最小脉冲宽度	45	-	-	



20. 振荡频率检测器(OFD)

如果CPU时钟脉冲的外部高频振荡超过检测频率范围，则该振荡频率检测器生成一次I/O复位。

振荡频率检测可通过OFDCR1和OFDCR2寄存器进行控制，检测频率范围则是由OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON等检测电频率设置寄存器进行设置的。较低检测频率可由OFDMNPLLOFF/OFDMNPLLON寄存器进行设置，而较高的检测频率则可由OFDMXPLLOFF/OFDMXPLLON寄存器进行设置。

当启用振荡频率检测时，将禁用OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON寄存器的写入。为此，应在禁用振荡频率检测的条件下对此类寄存器的检测频率进行设置。OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON寄存器写入可通过OFDCR1寄存器进行控制。如需对OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON寄存器进行写入，必须事先在OFDCR1中设置启用代码"0xF9"。在启用振荡频率检测器时，首先将OFDCR1设置为"0xF9"，然后再将OFDCR2设置为"0xE4"。由于振荡频率检测在外部复位输入，上电复位或VLTD复位之后被禁用，因此，需将"0xF9"写入到OFDCR1，并将"0xE4"写入到OFDCR2寄存器，以启用其功能。

当TMPM375FSDMG通过较低和较高检测频率设置寄存器检测到超出频率范围的现象时，所有的I/O都会通过复位变为高阻抗。在PLLOFF，OFDMNPLLOFF和OFDMXPLLOFF寄存器启用了检测功能的情况下，OFDMNPLLON/OFDMXPLLON寄存器的设置值将被忽略不计。在PLLON，OFDMNPLLON和OFDMXPLLON寄存器启用了检测功能的情况下，OFDMNPLLOFF/OFDMXPLLOFF寄存器的设置值则将被忽略不计。通过振荡频率检测复位，所有的I/Os(电源引脚除外)， $\overline{\text{RESET}}$ 引脚和MODE引脚均变为高阻抗，CPU也被初始化。由于CG寄存器进行了初始化，因此系统时钟的PLL运行操作功能将被禁用。只有将振荡器切换至片内振荡模式(fosc2)后才能重新启动时钟的操作。

因为振荡频率检测器的所有寄存器(OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON)均无法通过振荡频率检测器所产生的复位进行初始化。

为此，在产生振荡频率检测复位时，系统时钟将被切换至片内振荡器模式，并执行复位顺序，这是因为振荡频率检测功能此时已处于启用状态。

注：OFD无法保证能随时检测出所有的缺陷，其并不是一个用于测量错误频率的电路。

注：振荡频率检测的复位只有在NORMAL模式以及IDLE模式下才有效。STOP模式下会自动禁用振荡频率检测复位。

注：在通过CGPLSEL寄存器对PLL进行控制(启用或禁用)时，或通过CGOSCCR寄存器的<OSCSEL>对系统时钟时行更改(fosc1或fosc2)时，首先应禁用OFD。如果通过PLL-ON进行OFD复位，检测频率设置寄存器(OFDMNPLLON/OFDMXPLLON)将自动切换至OFDMNPLLOFF/OFDMXPLLOFF。

20.1 方块图

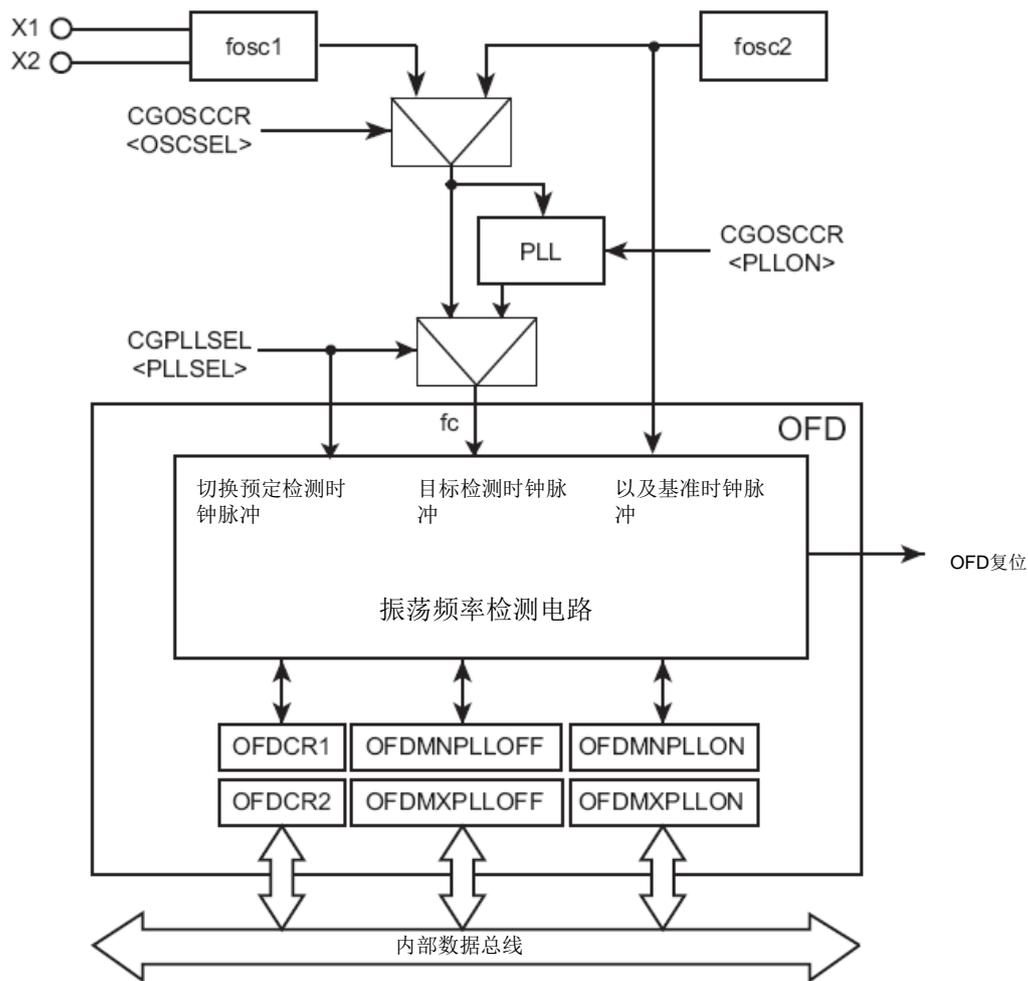


图 20-1 振荡频率检测器方块图

20.2 寄存器

20.2.1 寄存器列表

基址 = 0x4004_0800

寄存器名称		地址(基+)
振荡频率检测控制寄存器 1	OFDCR1	0x0000
振荡频率检测控制寄存器 2	OFDCR2	0x0004
较低检测频率设置寄存器(PLL OFF)	OFDMNPLLOFF	0x0008
较低检测频率设置寄存器(PLL ON)	OFDMNPLLON	0x000C
较高检测频率设置寄存器(PLL OFF)	OFDMXPLLOFF	0x0010
较高检测频率设置寄存器(PLL ON)	OFDMXPLLON	0x0014

注:禁止访问"保留"区。

20.2.1.1 OFDCR1(振荡频率检测控制寄存器)

	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	OFDWEN							
复位后	0	0	0	0	0	1	1	0

位	比特符号	类型	说明
31-8	-	R	读作0。
7-0	OFDWEN[7:0]	R/W	控制寄存器写入 0x06:禁用 0xF9:启用 设置0xF9即可启用写入到各寄存器(OFDCR 1 除外)。在写入0x06或0xF9以外的某个值时, 0x06 即被写入。 如果寄存器写入已被禁用, 则启用从各寄存器读取。

注:通过 $\overline{\text{RESET}}$ 引脚, 上电复位或VLTD复位初始化OFDCR1。

20.2.1.2 OFDCR2(振荡频率检测控制寄存器 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	OFDEN							
复位后	0	0	0	0	0	0	0	0

位	比特符号	类型	说明
31-8	-	R	读作0。
7-0	OFDEN[7:0]	R/W	控制频率检测。 0x00:禁用 0xE4:启用 写入 0x00 或 0xE4 以外的值无效，且不会改变某个值。

注:通过 $\overline{\text{RESET}}$ 引脚，上电复位或VLTD复位初始化OFDCR2。

20.2.1.3 OFDMNPLLOFF(较低检测频率设置寄存器(PLL OFF))

	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	-	-	-	-	-	-	-	OFDMNPLL- OFF
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	OFDMNPLLOFF							
复位后	0	0	0	1	1	1	0	1

位	比特符号	类型	说明
31-9	-	R	读作0。
8-0	OFDMNPLL- OFF[8:0]	R/W	设置内部较低检测频率。 针对基准时钟设置的复位后值为 $9.7\text{ MHz} \pm 5\%$ ，目标检测时钟脉冲为 $10\text{ MHz} \pm 5\%$ 。

注:在OFD电路操作的同时，OFDMNPLLOFF寄存器写入被保护。

注:通过RESET引脚，上电复位或VLTD复位初始化OFDMNPLLOFF。

20.2.1.4 OFDMNPLLON(较低检测频率设置寄存器(PLL ON))

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

位	比特符号	类型	说明
31-9	-	R	读作0。
8-0	OFDMNPLLON [8:0]	R/W	设置外部较低检测频率。 复位后，基准时钟的设置数值为 9.7 Hz 5%，目标检测时钟脉冲则为 40 MHz ± 5%。

注:在OFD电路操作的同时，OFDMNPLLON寄存器写入被保护。

注:通过RESET引脚，上电复位或VLTD复位初始化OFDMNPLLON。

20.2.1.5 OFDMXPLLOFF(高检测频率设置寄存器(PLL OFF))

	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	-	-	-	-	-	-	-	OFDMXPLL- OFF
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	OFDMXPLLOFF							
复位后	0	0	1	0	0	1	0	1

位	比特符号	类型	说明
31-9	-	R	读作0。
8-0	OFDMXPLLO FF[8:0]	R/W	设置内部较高检测频率。 复位后，基准时钟的设置数值为 $9.7 \text{ Hz} \pm 5\%$ ，目标检测时钟脉冲则为 $10 \text{ MHz} \pm 5\%$ 。

注:在OFD电路操作的同时，OFDMXPLLOFF寄存器写入被保护。

注:通过 $\overline{\text{RESET}}$ 引脚，上电复位或VLTD复位初始化OFDMXPLLOFF。

20.2.1.6 OFDMXPLLON(高检测频率设置寄存器(PLLON))

	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	-	-	-	-	-	-	-	OFDMXPL- LON
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	OFDMXPLLON							
复位后	1	0	0	0	1	1	1	1

位	比特符号	类型	说明
31-9	-	R	读作0。
8-0	OFDMXPLLON [8:0]	R/W	设置外部较高检测频率。 复位后，基准时钟的设置数值为 $9.7 \text{ Hz} \pm 5\%$ ，目标检测时钟脉冲则为 41 MHz 。

注:在OFD电路操作的同时，OFDMXPLLON寄存器写入被保护。

注:通过RESET引脚，上电复位或VLTD复位初始化OFDMXPLLON。

20.3 操作说明

20.3.1 设置

通过 $\overline{\text{RESET}}$ 引脚，上电复位或VLTD复位初始化OFD的各寄存器。除OFDCR1以外，所有的寄存器均无法通过复位进行写入操作。必须将OFDCR1设置为"0xF9"才能对这些寄存器进行写入操作。

每个目标时钟的检测频率范围均可通过 OFDMNPLLON/OFDMXPLLON or OFDMNPLLOFF/OFDMXPLLOFF 进行设置。在OFDCR1等于"0xF9"的条件下将OFDCR2设置为"0xE4"，可以启用振荡频率检测功能。

如需防止错误的写入，则应将OFDCR1设置为"0x06"。只有在OFD处于停机状态时才能对寄存器进行修改。

通过对CGPLLSEL<PLLSEL>进行设置，可以自动切换OFDMNPLLOFF/OFDMXPLLOFF和OFDMNPLLON/OFDMXPLLON。

如在OFDCR2等于"0xE4"的条件下运行STOP模式，其将自动禁用振荡频率检测的复位。只有在释放STOP模式以及结束暖机周期之后，才会启用振荡频率检测。振荡频率检测的只有在NORMAL模式以及IDLE模式下才能使用。表 20-1 中显示了振荡频率检测器的可用性。

表 20-1: 振荡频率检测器的可用性

操作模式	振荡频率检测 (OFDCR2=0xE4)	振荡频率检测后的所有I/Os条件(电源, $\overline{\text{RESET}}$, MODE, 引脚除外)
NORMAL	可用	高阻抗
空闲	可用	高阻抗
STOP (包括暖机时间)	振荡频率检测已被自动禁用。	
通过振荡频率检测复位实现复位	可用	高阻抗
看门狗时序器复位 SYSRESETREQ复位	可用	高阻抗
通过外部复位，上电复位，VLTD 复位实现复位	禁用	-

20.3.2 操作

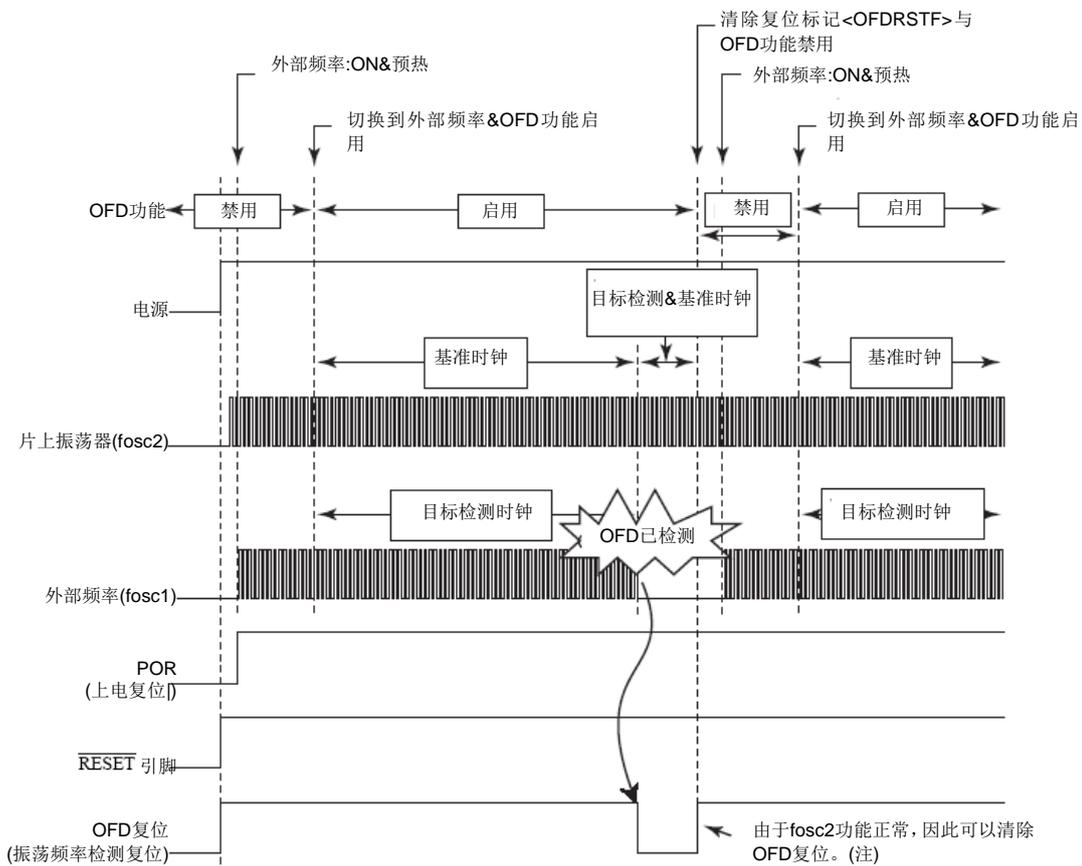
从操作启动到检测启动的过程中，其所需要的时间长度为检测时钟的两个周期。这一检测周期为128/基准时钟频率。

当生成复位功能处于启用状态时，一旦目标时钟频率超过了OFDMNPLLON/OFDMXPLLON和OFDMNPLLOFF/OFDMXPLLOFF中所设置的频率限值，则会进行重设。从检测异常到产生复位的过过程中，其所需要的时间长度为检测时钟的一个周期。OFD产生的复位并不能让其自身或OFD继续所检测的运行操作。

为此，当PLL处于OFF状态时，fosc会初始化为fosc2，而目标时钟(fc)也会变换为fosc2，且将继续进行所检测的运行操作。

注:复位具有多种因素。时钟生成寄存器CGRSTFLG可对这些因素进行确认。有关CGRSTFLG 的详情，请参阅"例外"章节。

注:目标时钟频率可以超出OFDMNPLLOFF和OFDMXPLLOFF的10 MHz频率限制进行设置。(例如:将寄存器的数值设置为 8 MHz)。此外，当目标时钟频率为"fosc1"，且已由OFD产生了重设时，有可能因出现异常检测而继续产生重设，直至OFD检测到正确为止。



注:目标时钟频率可以超出OFDMNPLLOFF和OFDMXPLLOFF的10 MHz频率限制进行设置。(例如，将8 MHz值设置为这些寄存器)。在目标时钟频率为fosc1，且由OFD生成复位时，可因异常检测而连续生成该复位，直至OFD可正常检测。

图 20-2:振荡频率检测操作示例

20.3.3 检测频率

由于振荡精确度的原因，检波频率有一个检测频率范围和一个不可检测的频率范围。因此，并未规定是否在检测频率范围和不可检测范围之间对其进行检测。

依据目标时钟和基准时钟的最大误差，计算出检测频率的上下限。

通过在决定OFDMNPLLON/OFDMNPLLOFF和OFDMXPLLON/OFDMXPLLOFF时四舍五入，可得出检测和不可检测范围的上下限，如以下所示。根据被检测时钟的非偶性，选择四舍五入的方式。

- 如果是往上舍入OFDMXPLLON/OFDMXPLLOFF和往下舍入OFDMNPLLON/OFDMNPLLOFF目标时钟高于不可检测范围的上限，并低于不可检测范围的下限。
- 如果是往下舍入OFDMXPLLON/OFDMXPLLOFF和往下舍入OFDMNPLLON/OFDMNPLLOFF目标时钟将低于漏检范围的上限，且高于漏检范围的下限。

以下给出了目标钟差为±5%(不可检测的范围)和基准时钟误差为±5%时，OFDMXPLLOFF/OFDMNPLLOFF设置值的计算方法。在该示例中，OFDMXPLLOFF被往上舍入，OFDMNPLLOFF被往下舍入(从 "a"到"h"，与"图 20-3 检测频率范围示例 (如果是10 MHz)"对应)。

目标时钟	10 MHz ± 5%	最大10.5 MHz	-----c
		最小9.5 MHz	-----b
基准时钟	9.7 MHz ± 5%	最大10.185 MHz	-----f
		最小9.215 MHz	-----e

$$OFDMXPLLOFF = c \div e \times 32 = 36.46... = 37 \text{ (往上舍入到最近的小数位)} = 0x25$$

$$OFDMNPLLOFF = b \div f \times 32 = 29.85... = 29 \text{ (往下舍入到最近的小数位)} = 0x1D$$

此时的检测范围计算按以下所述。

$$a = e \times OFDMNPLLOFF \div 32 = 8.35$$

$$d = f \times OFDMXPLLOFF \div 32 = 11.78$$

不可检测范围的计算按以下所述。

$$g = e \times OFDMXPLLOFF \div 32 = 10.65$$

$$h = f \times OFDMNPLLOFF \div 32 = 9.23$$

将"0x25"设置为寄存器OFDMXPLLOFF，并将"0x1D"设置为寄存器OFDMNPLLOFF，在检测到高于11.78 MHz或低于 8.35 MHz的目标时钟时，振荡频率检测器会输出一个复位信号。在检测到高于 9.23 MHz和低于 10.65 MHz的目标时钟时，振荡频率检测器不会输出复位信号。

图 20-3 给出了检测范围，或不可检测的和可检测的频率范围。

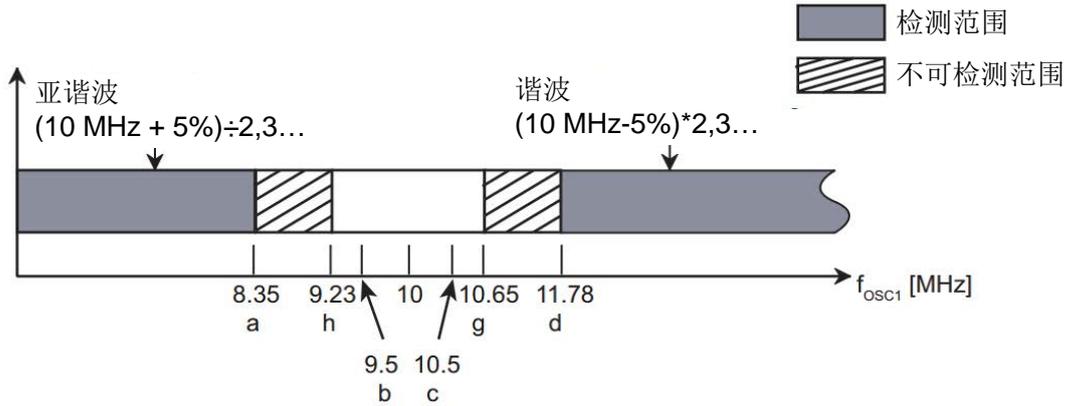


图 20-3 检测频率范围示例(如果是10 MHz)

20.3.4 可用的运行模式

振荡频率检测仪在外部振荡频率处于NORMAL和IDLE模式时可用。在转换到另一个模式或使用芯片振荡频率之前，需禁用振荡频率检测。

20.3.5 运算程序示例

运行程序示例如下所示。

在复位之后，需由CGRSTFLG认可各复位因数。如果复位因数并非按振荡频率检测确定，则启用外部振荡，将寄存器设置为使用OFD并启用运行。

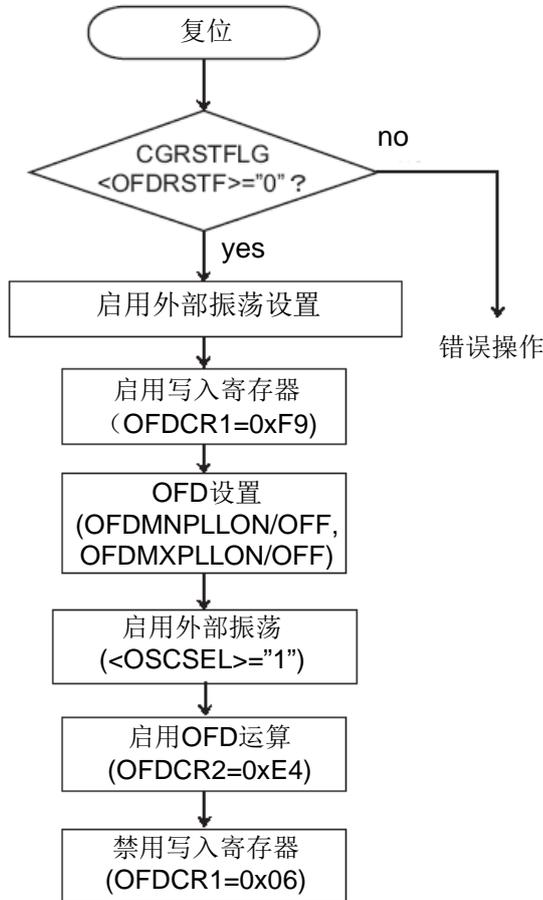


图 20-4 运算程序示例



21. 看门狗定时器(WDT)

看门狗定时器(WDT)用于检测噪声或其他干扰造成的故障(失控)，并修复它们，使CPU恢复正常。若看门狗定时器检测到失控现象，它会生成INTWDT中断或复位。

注:INTWDT中断是不可屏蔽中断(NMI)的一个因素。

此外，通过输出“低”，看门狗定时器将检测的故障从看门狗定时器引脚($\overline{\text{WDTOUT}}$)通知外设装置。

注:该产品没有看门狗定时器输出引脚($\overline{\text{WDTOUT}}$)。

21.1 配置

图 21-1 给出了看门狗定时器的方块图。

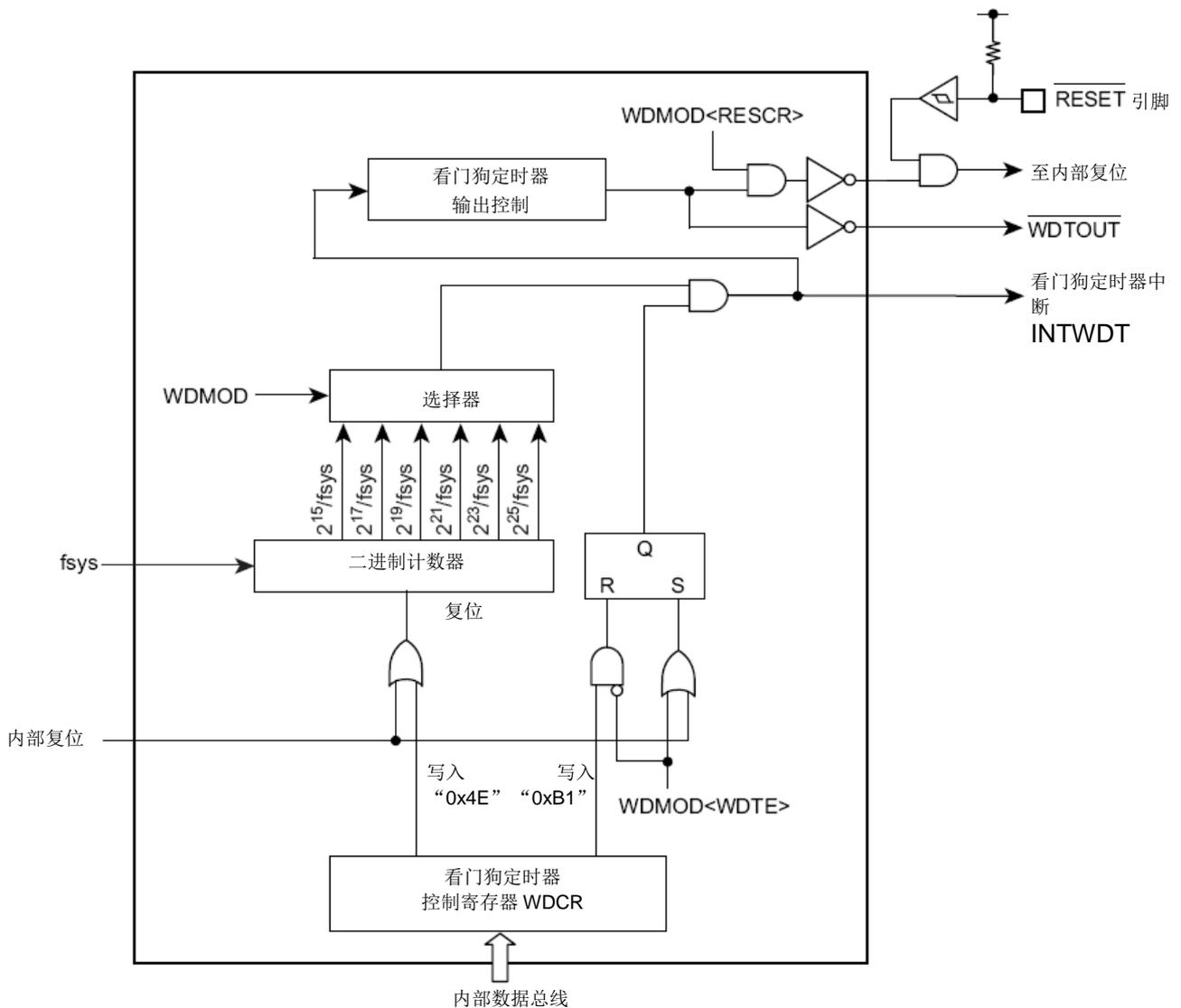


图 21-1:监视定时器方块图

21.2 寄存器

看门狗定时器控制寄存器和地址如下。

基址 = 0x4004_0000

寄存器名称		地址(基+)
看门狗定时器寄存器地址	WDMOD	0x0000
看门狗定时器控制寄存器	WDCR	0x0004

21.2.1 WDMOD (看门狗定时器模式寄存器)

	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
6-4	WDTP[2:0]	R/W	选择WDT检测时间(见表 21-1) 000:2 ¹⁵ /fsys 100: 2 ²³ /fsys 001:2 ¹⁷ /fsys 101: 2 ²⁵ /fsys 010:2 ¹⁹ /fsys 110:禁止设置。 011:2 ²¹ /fsys 111:设置被禁止。
3	-	R	读作0。
2	I2WDT	R/W	在IDLE模式时运行 0:停止 1:运行中
1	RESCR	R/W	在检测到故障后运行 0:INTWDT中断请求生成。(注) 1:复位
0	-	R/W	写入 0。

注:INTWDT中断是不可屏蔽中断(NMI)的一个因素。

表 21-1 看门狗定时器的检测时间 (fc = 40 MHz)

时钟齿轮值 CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
100 (fc/2)	1.63 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
101 (fc/4)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
110 (fc/8)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s
111 (fc/16)	13.12 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s	13.42 s

21.2.2 WDCR(看门狗定时器控制寄存器)

	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/清除编码 0xB1:禁用代码 0x4E:清除编码 其他:保留

21.3 运行

21.3.1 基本运行

看门狗定时器由二进制计数器组成，这些计数器将系统时钟(fsys) 用作输入。检测时间由 $WDMOD<WDTP[2:0]>$ 在 2^{15} ， 2^{17} ， 2^{19} ， 2^{21} ， 2^{23} 和 2^{25} 之间选择。规定的检测时间过去，看门狗定时器中断(INTWDT)生成，看门狗器输出引脚(\overline{WDTOUT}) 输出"低"。

为了检测噪声或其他干扰造成的CPU故障(失控)，在INTWDT中断生成前，应该用软件指令清除看门狗定时器二进制计数器。若二进制计数器未被清除，INTWDT不可屏蔽中断就会生成。因此，CPU进行故障检测，并运行故障对策程序，以恢复正常工作。

此外，可能通过连接看门狗定时器输出引脚和外设复位引脚而解决CPU故障(失控)问题。

注:本产品不包括看门狗定时器输出引脚(\overline{WDTOUT})。

21.3.2 运行模式和状态

在复位被清除后，看门狗定时器立即开始运行。

若不使用看门狗定时器，它应被禁用。

由于高速频率时钟停止，看门狗定时器无法使用。在切换到下列模式前，看门狗定时器应被禁用。在IDLE模式时，它的运行取决于 $WDMOD <I2WDT>$ 的设置。

而且，调试模式下，二进制计数器自动停止。

21.4 在故障(失控)被检测时运行

21.4.1 INTWDT 中断生成

图 21-2 中显示了INTWDT中断生成的情况(WDMOD<RESCR>="0")。

当二进制计数器发生溢位时，INTWDT中断就会产生。它是不可屏蔽中断(NMI)的一个因素。因此，CPU检测不可屏蔽中断，并运行对策程序。

不可屏蔽中断因素为复数。CGNMIFLG识别不可屏蔽中断因素。在INTWDT中断的情况下，CGNMIFLG<NMIFLG0>就会被设置。

当INTWDT中断生成时，看门狗定时器输出引脚($\overline{\text{WDTOUT}}$)同时输出"低"。通过看门狗定时器的清除行为(将清除编码0x4E写入WDCR寄存器)， $\overline{\text{WDTOUT}}$ 变"高"。

注:本产品无看门狗定时器输出引脚($\overline{\text{WDTOUT}}$)。

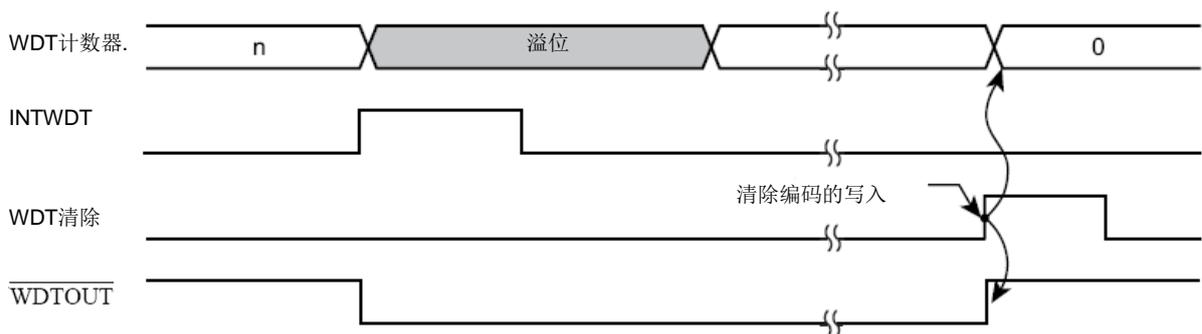


图 21-2:INTWDT 中断的生成

21.4.2 内部复位生成

图 21-3 给出了内部复位生成(WDMOD<RESCR>="1")。

MCU通过二进制计数器的溢位而复位。在这种情况下，对于 32 种状态，复位状态继续。时钟被初始化，以致输入时钟(fsys)与内部高速频率时钟(fosc)相同。这意味着 $f_{sys} = f_{osc}$ 。

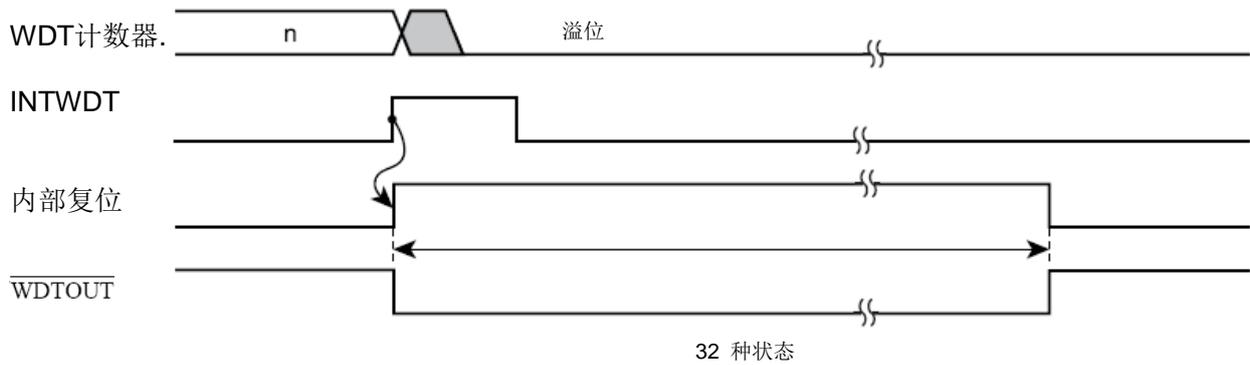


图 21-3:内部复位的生成

21.5 控制寄存器

看门狗定时器(WDT)受两个控制寄存器WDMOD和WDCR控制。

21.5.1 看门狗定时器寄存器地址(WDMOD)

1. 看门狗定时器<WDTP[2:0]>的检测时间的规定。

将看门狗定时器的检测时间设置为WDMOD<WDTP [2:0]>。复位后，初始化至WDMOD<WDTP [2:0]> = "000"。

2. 看门狗定时器<WDTE>的启用/禁用。

当复位时，WDMOD <WDTE>被初始化到 "1"，并启用看门狗定时器。

禁用看门狗定时器，防止因故障造成写入错误时，<WDTE>位元首先设置为 "0"，然后，禁用代码(0xB1)必须写入WDCR寄存器。

为了将看门狗定时器的状态从"禁用"变为"启用"，将<WDTE>位设置为 "1"。

3. 看门狗定时器关断复位连接<RESCR>

该寄存器规定WDTOUT是用于内部复位还是用于中断。在复位后，WDMOD<RESCR>被初始化到 "1"，内部复位通过二进制计数器的溢位而生成。

21.5.2 看门狗定时器控制寄存器(WDCR)

这是一个用于禁用定时器功能并控制二进制计数器的清除功能的寄存器。

21.5.3 设置例子

21.5.3.1 禁用控制

通过在将WDMOD <WDTE> 设置为 "0" 后将禁用代码(0xB1)写入本WDCR寄存器，可禁用看门狗定时器，而且二进制计数器可清零。

		7	6	5	4	3	2	1	0	
WDMOD	←	0	-	-	-	-	-	-	-	将<WDTE>设置为"0"。
WDCR	←	1	0	1	1	0	0	0	1	写入禁用编码(0xB1)。

21.5.3.2 启用控制

将WDMOD<WDTE>设置为 "1"。

		7	6	5	4	3	2	1	0	
WDMOD	←	1	-	-	-	-	-	-	-	将<WDTE>设置为 "1"。

21.5.3.3 看门狗定时器清除控制

将清除编码(0x4E)写入WDCR寄存器就能清除二进制计数器，并且开始计数。

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	写入清0代码(0x4E)。

21.5.3.4 看门狗定时器的检测时间

在使用 $2^{21}/f_{sys}$ 的情况下，将 "011" 设置为WDMOD<WDTP[2:0]>。

		7	6	5	4	3	2	1	0	
WDMOD	←	1	0	1	1	-	-	-	-	



22. 闪存操作

本章节对硬件配置和闪存操作进行了说明。在本章节中，“1-字”表示32 位。

22.1 特点

22.1.1 存储器容量和配置

表 22-1 与图 22-1 显示TMPM375FSDMG内置存储器容量与配置

表 22-1 存储器容量与配置

存储器容量	块配置				每页字数	# 页数	写入时间		擦除时间	
	128 KB	64 KB	32 KB	16 KB			1页	总面积	块擦除	芯片擦除
64 KB	-	-	2	-	32	512	1.25 ms	0.64 sec	0.1 sec	0.2 sec

注:上述数值为理论值, 不含数据传输时间。 每页写入时间根基用户使用的方法而定。



图 22-1 块配置

闪存配置单元按“块”和“页面”所述。

页

一个页面 32 个字。一个页面内使用同一个地址[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。

块

一个块为 32 KB, 闪存由两个块组成。

按页进行写入操作。每页写入时间为1.25 ms。(典型值)

每块均进行擦除(使用自动块擦除指令)或整个闪存均进行擦除(使用自动片擦除指令)。擦除时间因指令不同而不同。采用自动块指令时, 每块的擦除时间应为 0.1 s(典型值)。采用自动片擦除指令擦除整个区域时, 擦除时间应为 0.2 s(典型值)。

同时，每块可使用保护功能。有关该保护功能的详细说明，见"22.1.5 保护/安全功能"。

22.1.2 功能

内置于本装置内的闪存通常符合JEDEC标准，一些特定功能除外。所以，当用户正在使用闪存作为外部存储器时，很容易在本装置内执行这些功能。另外，当带有快写或擦除操作时，本产品配置有一固定的电路自动执行写入或芯片擦除操作。

JEDEC 符合功能	修改，增加或删除功能
<ul style="list-style-type: none"> · 自动编程 · 自动芯片擦除 · 自动块擦除 · 数据轮询/反转位 	<p><Modified> 块写入/擦除保护 (仅支持软件保护)</p> <p><Deleted> 擦除恢复暂停功能</p>

22.1.3 运行模式

22.1.3.1 模式说明

本装置具有单芯片模式和单启动模式。单芯片模式包括正常模式和用户启动模式。图 22-2 显示模式转换。

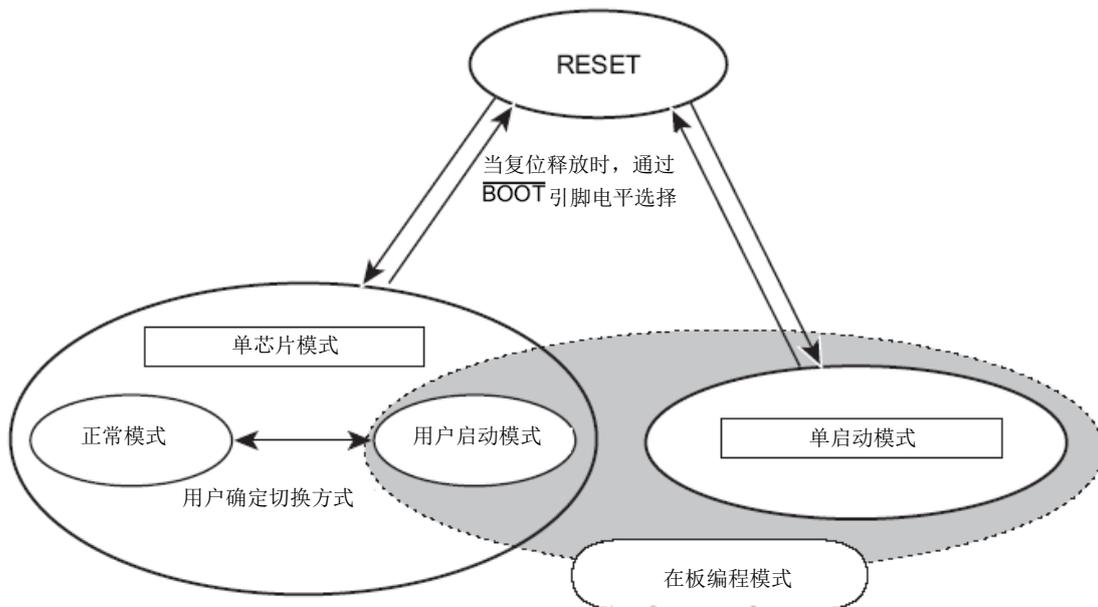


图 22-2 模式转换

(1) 单芯片模式

单芯片模式为一种复位后装置能够闪存启动的模式。它包括以下两种子模式。

正常模式

在该模式下，用户执行应用程序。

用户启动模式

在该模式下，在用户通过设置对闪存重新编程。

用户可将正常模式转换为用户启动模式。例如，当端口A的PA0设置为"1"时，用户可进行设置，使模式为正常模式。当端口A的PA0为"0"，模式为用户启动模式。用户必须在应用程序中编制一例程序来确定切换。

(2) 单启动模式

为复位后闪存可自内置BOOT ROM (Mask ROM)启动。

BOOT ROM 包括在用户设置条件下通过本装置串行端口重新写入闪存的算法。当将串行端口接入外部主机时，按照上述协议和重新编程的闪存进行数据传输。

(3) 在板编程模式

用户启动模式和单启动模式为在用户设置条件下，可进行编程的两种模式，被称之为"在板编程模式"。

22.1.3.2 模式判定

单芯片或单个启动运行模式在复位释放时均可依据BOOT引脚的电平来选择。

表 22-2 运行模式设置

运行模式	引脚	
	RESET	BOOT
单芯片模式	0 → 1	1
单启动模式	0 → 1	0

22.1.4 存储器映象

图 22-3 给出了单芯片模式和单启动模式下的存储器映象。在单启动模式下，内置闪存被映射到 0x3F80_0000和后续的地址，且内置BOOT ROM则被映射到0x0000_0000~0x0000_0FFF。

闪存和RAM地址如下。

闪存容量	RAM容量	FLASH 地址	RAM地址
64 KB	4 KB	0x0000_0000~0x0000_FFFF_ffff(单芯片模式)	0x2000_0000~0x2000_0FFF
		0x3F80_0000至0x3F80_FFFF(单启动模式)	

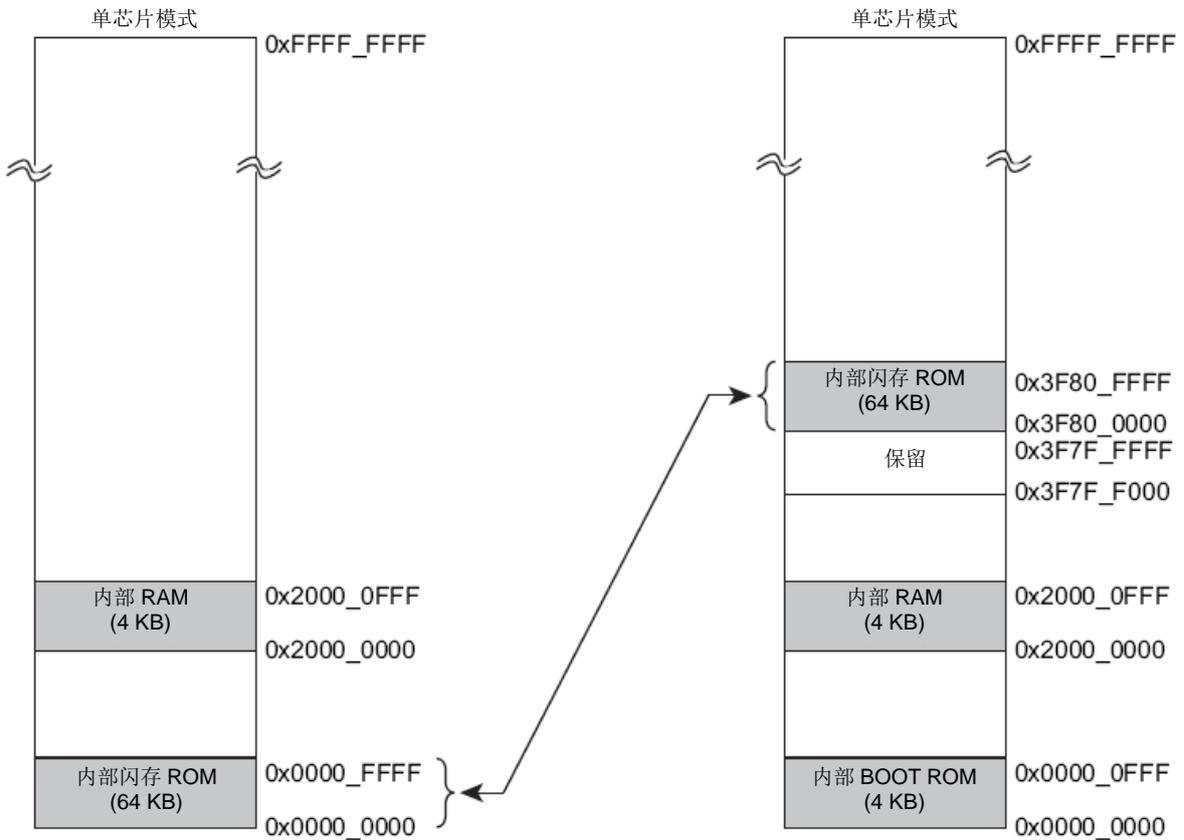


图 22-3 内存映象的比较

22.1.5 保护/安全功能

本装置对闪存带有保护和安全管理。

1. 保护功能
可禁止按块写入/擦除操作。
2. 安全功能
可禁止从闪存写入程序读取。
调试功能的使用限制

22.1.5.1 保护功能

该功能禁止按块进行的写入/擦除操作。

启用保护功能时，采用保护位程序指令将对应块的保护位设置为"1"。如果保护位被用保护位擦除指令设置为"0"，则可取消块保护。该保护位可通过FCPSRA<BLK[1:0]>监视。

保护位程序可通过 1-位单元进行编程，并通过 4-位单元擦除。有关保护位编程/擦除的详细说明，见"22.2.5 指令描述"。

22.1.5.2 安全功能

表 22-3 显示安全功能启用时的操作。

表 22-3 安全功能启用时的操作

项目	说明
读取闪存	CPU 可读取闪存。
调试端口	JTAG，串行线或跟踪通讯禁用。
向闪存执行指令	向闪存写入指令不被接受。当用户试图擦除某一保护位时，芯片被擦除，同时所有保护位被擦除。

下述条件下安全功能启用；

1. FCSECBIT<SECBIT>设置为 "1"。
2. 所有保护位(FCPSRA<BLK>)均设置为 "1"。

通过冷态复位(上电复位)将FCSECBIT<SECBIT>设置为 "1"。FCSECBIT <SECBIT>的覆写见以下所述。

注:当执行下述写入操作第 1 项和第 2 项时，采用 32-位传输指令。

1. 将规定的代码(0xa74a9d23) 写入到FCSECBIT
2. 在项目 1 运算结束之后 16 个时钟内写入数据。

22.1.6 寄存器

22.1.6.1 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
闪存接口控制寄存	FCCR 0x001C
闪存状态寄存器	FCSR 0x0020
闪存保护状态寄存器A	FCPSRA 0x0030

22.1.6.2 FCCR(闪存接口控制寄存器)

	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	-	-	-	-	-	-	-	FLBOFF
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	FLBOFF	R/W	与带指令缓存区的FLASH接口控制(注 1) 0:启用指令缓存器 1:禁用指令缓存器(用缓存器清除) 该位为控制FLASH接口的功能位。 如拟使用指令缓存器, 则设置为 "0"。如不拟使用指令缓存器, 则设置为 "1"。 在TMPM375FSDMG中, 必须将其设置为 "0" 已进行闪存存取。 b

注 1:在TMPM375FSDMG中进行过闪存编程或闪存擦除之后, 其应为通过该功能位清除指令缓存器, 或插入一个复位信号。

指令缓存器清除操作如下。

在执行了FCCR<FLBOFF>="1" 之后, 再次在RAM上设置FCCR<FLBOFF>="0"。

22.1.6.3 FCSR(闪存状态寄存器)

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

位	比特符号	类型	功能
31-1	-	R	读作"0"。
0	RDY/BSY	R	准备/忙碌 (注 1) 0:忙碌 (自动操作时) 1:准备 (自动操作端) 此位为自CPU监视闪存的功能位。闪存自动操作时, 此位输出"0"表示闪存忙碌。自动操作完成后, 此位变为准备状态, 同时输出"1"。然后接收下一个指令。 当自动操作失败时, 此位继续输出"0"。此位通过硬件复位返回至"1"。

注 1:发出指令前应确保闪存处于准备状态。如果在忙碌期间发布一条指令, 则不仅该指令不会被发送, 而且后续的指令也不会被接收。在那样的情况下, 采用硬件复位方法返回。不管时钟设置如何, 硬件复位都需要0.5 μs 或以上的的时间。此时, 在复位后需耗时2 ms左右才能启用读取。

22.1.6.4 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 设置禁用。 1:安全功能 启用。

注:本寄存器通过冷态复位(上电复位)初始化。

22.1.6.5 FCPSRA(闪存保护状态寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	BLK1	BLK0
复位后	0	0	0	0	0	0	(注 1)	(注 1)

位	比特符号	类型	功能
31-2	-	R	读作"0"。
1-0	BLK1- BLK0	R	块 1~0 的保护状态 0:不受防护 1:受保护 与各块保护状态对应的保护位数值。 当对应位指示 "1" 时, 对应块处于保护状态。 处于保护状态的A块不可编程。

注 1: 保护状态应有A值对应。

22.2 闪存详细情况

在板编程时，CPU执行闪存重新编程或擦除指令。重新编程/擦除控制程序应由用户事先编制。闪存存在写入或擦除时不能同时读取，所以必须在内置RAM上运行重新编程/擦除控制程序。除复位外，不得生成中断/故障，以免出现异常程序终止。

22.2.1 功能

除一些特定功能外，闪存一般会符合JEDEC标准。不过，工作指令的地址规定方法不同于标准指令。如执行将写入/擦除操作，则需使用32-位(1-字)存储指令指令，将指令输入到闪存。输入指令后，内部自动执行写入或擦除操作。

表 22-4 闪存功能

主要的功能	说明
自动页面程序	自动写数据。
自动芯片擦除	自动擦除整个闪存区。
自动块擦除	自动擦除某个已选块。
写入/擦除保护	可分别禁止对每个块的写入或擦除。

注:在TMPM375FSDMG中进行闪存编程或闪存擦除之后，其应为清除指令缓存器。有关清除方法，见"22.1.6.2 FCCR (闪存接口控制控制寄存器)"。

注:检查FCSR<RDY/BSY>以确认闪存写入，闪存擦除，保护位程序，保护位擦除等各指令序列已结束，然后保持200 μs或以上，再从闪存读取数据或启动指令取出。

22.2.2 闪存运行模式

闪存主要提供两个运行模式；

- 读取寄存器数据模式(读取模式)
- 自动擦除或重写寄存器数据模式(自动运行模式)

上电后，正常复位或自动运行模式完成后，闪存变为读取模式。在读取模式下执行闪存指令存储或数据读取。

读取模式时输入的指令时，运行模式变为自动运行。当指令过程正常完成时，运算模式会返回到读取模式(ID读指令除外)。自动运行时，不能进行存储于闪存的数据读取和执行指令。

当完成指令处理出现异常时，应强制将运行模式返回至读取模式。在这种情况下，采用读取指令，读取/复位指令或硬件进行复位。

22.2.3 硬件复位

硬件复位指的是冷复位(通电复位)或热复位，在自动编程/擦除操作强制取消，或自动操作不正常结束时，使用返回到读取模式。

当自动操作过程中发生硬件复位时，闪存停止自动操作并返回读取模式。当闪存自动编程/擦除操作过程中生成硬件复位时，无论系统时钟如何，硬件复位都需要0.5 μs或更多的复位时间。这时，大约需要2 ms，直至复位后读取启用。注意当自动操作期间发生硬件复位时，数据写入操作不正。应再设置写入操作。

"复位操作的详细信息见"复位"说明。在给定复位输入后，CPU将读取复位矢量数据，然后在复位后启动例行程序。

22.2.4 如何执行指令

由指令序列用一条存储指令对闪存完成指令执行。闪存按照输入地址和数据的组合，执行各自动运算指令。有关指令执行的详细说明，见"22.2.5 指令描述"。

向闪存执行以存储指令被称为"总线写入周期"。每个指令都由一些总线构成。在闪存中按规定的顺序执行总线写周期的地址和数据时，可执行自动指令操作。在按非规定顺序执行该周期时，闪存停止指令执行，并返回到读取模式。

在指令序列或输入一个不同的指令序列过程中取消该指令时，应执行读取指令或读取/擦除指令。闪存停止执行指令并返回至读取模式。读取指令和读取/复位指令被称为"软件复位"。

在写指令序列结束时，自动运算启动，FCSR<RDY/BSY>即被设置为"0"。在自动运算正常结束时，FCSR<RDY/BSY>="1"即被设置，且闪存会返回到读取模式。

自动操作过程中不接受新的指令序列。可通过硬件复位来停止指令操作。如果该自动运算反常结束(FCSR<RDY/BSY>仍然是"0")，闪存仍处于锁定状态，且不会返回到读取模式。通过硬件复位返回至读取模式。当硬件复位停止了指令操作时，指令则不会正常执行。

指令执行注意事项：

1. 识别指令时，指令序列发生器需在指令启动之前处于读取模式。确认FCSR<RDY/BSY>=1已在各指令的首个总线写周期之前完成设置。建议连续执行读取指令。
2. 从闪存的外部执行各指令序列。
3. 通过一个字(32-位)数据传送指令，顺次执行各总线写入周期。
4. 在各指令序列执行期间，不要访问闪存。除复位外，请勿生成任何终端或错误。
5. 一旦发布一条指令，如果任何地址或数据写入错误，则通过使用软件复位确认并返回到读取模式。

22.2.5 指令说明

本章节对各种指令进行解释说明。有关特定指令序列的详细说明，见"22.2.6 指令序列"。

22.2.5.1 自动页面程序

(1)操作说明

按页面进行自动页面程序写入数据。当程序写入数据至多页时，需要每页执行一个页面指令。不能跨页写入。

向闪存写入是指数据单元"1"变为"0"。数据单元不能从"0"变为"1"。必须进行擦除操作才能成为从"0"变成"1"的数据单元。

对每个已经擦除的页面只允许自动页面程序一次。数据单元"1"或"0"不能写入两次或多次。当向已经写入一次的页面重新写入时，在执行自动块擦除或自动片擦除指令后需要重新设置自动页面程序。

注 1: 在没有擦除的情况下对同一个页面执行页面程序会损坏装置。

注 2: 不能向受保护的块写入。

(2)如何设置

第 1~第 3 个总线写入周期包括自动页面程序指令。

在第 4 个总线写入周期中，写入页面的第 1 个地址和数据。在第 5 个总线之后，按顺序写入一个页面。数据以一个字句单元(32-位)写入。

当部分页面写入后，将"0xFFFFFFFF"设置为数据，表示不需要整页写入。

装置内部不执行自动验证操作。因此，务必读取已编程的数据，以确认写入正确。

当自动页面程序异常终止时，该页面无法写入。建议不使用装置，或不使用块(包括已失效的地址)。

22.2.5.2 自动芯片擦除

(1)操作说明

向所有地址的存储单元执行自动芯片擦除操作，而其中所包含的受保护块不用擦除。但如果所有块均受保护，则不会执行自动芯片擦除操作，而且在输入指令序列后返回至读取模式。

(2)如何设置

第 1~6 个总线周期表示自动芯片擦除指令。输入指令序列后，开始自动芯片擦除操作。

装置内部不执行自动验证操作。所以，应确保读取数据符合已经正确擦除的数据。

22.2.5.3 自动块擦除

(1) 操作说明

自动擦除指令执行向规定块的擦除操作。如果规定的块受保护，则不执行擦除操作。

(2) 如何设置

第 1~5 个总线写入周期表示自动块擦除指令。在第 6 个总线周期中，规定了块的写入。输入指令序列后，启动自动块擦除操作。

装置内部不执行自动验证操作。所以，应确保读取数据符合已经正确擦除的数据。

22.2.5.4 自动保护位程序

(1) 操作说明

始终用自动保护位程序向受保护位写入"1"。向某一保护位写入"0"时，使用自动保护位擦除指令。有关该保护功能的详细说明，见"22.1.5 保护/安全功能"。

(2) 如何设置

第 1~6 个总线周期表示自动保护位程序指令。在第 7 个总线周期中，规定了待写入的保护位。输入指令序列后，启动自动保护位程序。检查写入操作是否是以FCPSRA<BLK>正常终止的。

22.2.5.5 自动保护位擦除

(1) 操作说明

自动保护位擦除指令根据安全状态进行操作。有关安全状态的详细说明，见"22.1.5 保护/安全功能"。

非安全状态

将规定的保护位设置为"0"。用 4-位单元执行保护位擦除。

安全状态

闪存的所有地址擦除后再擦除所有保护位。

(2) 如何设置

第 1~6 个总线写入周期表示自动保护位擦除指令。在第 7 个总线写入周期中，规定了待擦除的保护位。输入指令序列后，启动自动保护位擦除。

非安全状态下，将已经指定的保护位擦除。检查擦除操作是否是以FCPSRA<BLK>正常终止的。

在安全状态下，闪存位的所有地址和所有保护被清除。应确认数据和保护位是否正常擦除。必要时，执行自动保护位擦除，自动芯片擦除或自动块擦除。

如果所有情况都与其它指令相同，则FCSR<RDY/BSY>会在自动保护位擦除指令操作期间变为"0"。在该操作完成后FCSR<RDY/BSY>变为"1"，且闪存会返回到读取模式。终止该操作时，要求通过硬件复位。

22.2.5.6 ID-读取

(1) 操作说明

ID-读取指令可读取包含闪存类型和诸如制码，装置代码及微代码等信息。

(2) 如何设置

第 1~3 个总线写入周期表示ID-读取指令。在第 4 个总线写入周期中，规定了待读取代码。第 4 个总线周期后，在随机闪存区域要求有代码进行读取操作。

可依次执行ID-读取操作。可重复执行第 4 个总线周期和读取ID值。

ID读取指令不会自动返回读取模式。返回至读取模式时，执行读取指令，读取/复位指令或硬件复位指令。

22.2.5.7 读取指令和读取/复位指令(软件复位)

(1) 操作说明

返回闪存读取模式的指令。

执行ID-读取指令时，微处理器在当前状态停止，而不会自动返回至读取模式。从此状态返回至读取模式时，利用读取指令或读取/复位指令。当指令输入至中途时，它也看用于取消指令。

(2) 如何设置

第 1 个总线周期表示读取指令。第 1~3 个总线写入周期表示读取/复位指令。无论执行哪个指令序列后，闪存返回至读取模式。

22.2.6 指令序列

22.2.6.1 指令序列列表

表 22-5 显示了每个指令中的总线写入周期地址与数据。

除ID读取指令的第 5 个总线周期外，所有指令为总线写入周期。通过 32-位(1-字)数据传送指令执行一个总线写周期(下表仅给出了数据的较低 8 位)。

有关地址的详细说明，见表 22-6。针对表 22-6 Addr[15:9]栏中的"指令"，使用以下各值。

注 1) 始终将"0"设置为地址位[1:0]。

注 2) 根据闪存容量将以下值设置为地址位[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：当总线写至1 MB或以下区域时，将该位设置 "0"。

当总线写入至大于1MB的区域时，将该位设置为 "1"。

表 22-5 指令序列

指令	第一个总线周期	第二个总线周期	第三个总线周期	第四个总线周期	第五个总线周期	第六个总线周期	第七个总线周期
	Addr.						
	数据						
读取	0xXX	-	-	-	-	-	-
	0xF0	-	-	-	-	-	-
读取/复位	0xX55X	0xAAX	0x55X	-	-	-	-
	0xAA	0x55	0xF0	-	-	-	-
ID-读取	0xX55X	0xAAX	0x55X	IA	0xXX	-	-
	0xAA	0x55	0x90	0x00	ID	-	-
自动页面程序	0xX55X	0xAAX	0x55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
自动芯片擦除	0xX55X	0xAAX	0x55X	0x55X	0xAAX	0x55X	-
	0xAA	0x55	0x80	0xAA	0x55	0x10	-
自动块擦除	0xX55X	0xAAX	0x55X	0x55X	0xAAX	BA	-
	0xAA	0x55	0x80	0xAA	0x55	0x30	-
自动保护胃程序	0xX55X	0xAAX	0x55X	0x55X	0xAAX	0x55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
自动保护位擦除	0xX55X	0xAAX	0x55X	0x55X	0xAAX	0x55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

补充说明

IA:ID地址

ID:ID数据

PA:程序页面地址

PD:程序数据 (32-位数据)

第 4 个总线周期后，按页面输入地址顺序数据

BA:块地址 (见表 22-7)

PBA:保护位地址(见表 22-8)

22.2.6.2 总线周期中的地址位配置

表 22-6 应与"表 22-5 指令序列"配套使用。

根据正常总线写入周期地址配置，从第 1 个总线周期开始进行地址设置

表 22-6 总线写入周期中的地址位配置

地址	Addr. [31:15]	Addr. [14]	Addr. [13:12]	Addr. [11:9]	Addr. [8:7]	Addr. [6:4]	Addr. [3:0]
正常指令	正常总线写入周期地址配置						
	闪存区	推荐"0"	指令	Addr[1:0] = "0" (固定) 其它位= "0" (建议采用)			
ID-READ	IAID 地址 (ID-READ第 4 个总线写入周期设置)						
	闪存区	推荐"0"	ID地址	Addr[1:0] = "0" (固定)其它位= "0" (推荐)			
块擦除	BA块地址(块擦除第 6 个写入周期地址设置)						
	块地址(表 22-7)	Addr[1:0] = "0" (固定)其它位= "0" (推荐)					
自动页面程序	PA程序页面地址(页面程序第 4 总线写入周期设置)						
	页面地址					Addr[1:0] = "0" (固定) 其它位= "0" (推荐)	
保护位程序	PBA:保护位地址 (保护位程序第 7 总线写入周期地址设置)						
	闪存区	固定至 "0"			保护位选择 (表 22-8)	Addr[1:0] = "0" (固定) 其他位= "0" (推荐)	

22.2.6.3 块地址(BA)

表 22-7 给出了各块地址。 规定了包含在自动块擦除指令第 6 个总线写入周期中待擦除的块中的任意地址。

表 22-7 块地址

块	地址 (用户启动模式)	地址 (单启动模式)	容量 (千字节)
1	0x0000_8000~0x0000_FFFF	0x3F80_8000~0x3F80_FFFF	32
0	0x0000_0000~0x0000_7FFF	0x3F80_0000~0x3F80_7FFF	32

22.2.6.4 如何规定保护位(PBA)

在编程时在 1-位单元中和在擦除时 4-位单元中对保护位加以规定。

表 22-8 给出了自动保护位程序的保护位选择表。该地址栏表明，上面所述的地址用于该使用启动模式，而下面则用于单启动模式。

总共有四个保护位通过自动保护位擦除指令擦除。

表22-8 保护位程序地址

块	保护位	第七位总线地址写入周期			地址举例 [31:0]
		地址 [14:9]	地址 [8]	地址 [7]	
块 0	<BLK[0]>	固定为"0"	0	0	0x0000_0000 0x3F80_0000
块 1	<BLK[1]>		0	1	0x0000_0080 0x3F80_0080

22.2.6.5 ID-读取代码 (IA, ID)

表 22-9 显示如何使用ID-读取指令确定代码与内容。

地址栏举例表明，上侧中叙述的地址用于用户启动模式，下侧的则用于单启动模式

表 22-9 ID-读取指令代码与内容

代码	ID[7:0]	IA[13:12]	地址举例 [31:0]
制造商代码	0x98	0b00	0x0000_0000 0x3F80_0000
装置代码	0x5A	0b01	0x0000_1000 0x3F80_1000
-	保留	0b10	-
微处理器代码	0x33	0b11	0x0000_3000 0x3F80_3000

22.2.6.6 指令序列举例

(1)使用启动模式

指令	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-读取	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	在此后的周期中, 依次写入每页的地址和数据		
	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

指令	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-读取	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	在此后的周期中, 依次写入每页的地址和数据		
	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

22.2.7 流程图

22.2.7.1 自动程序

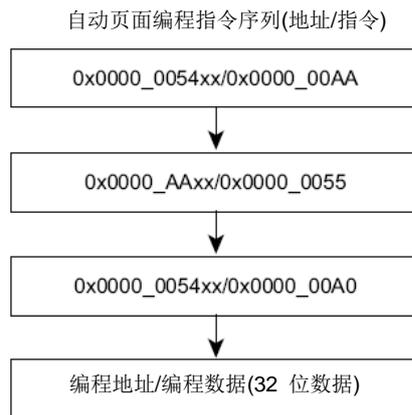
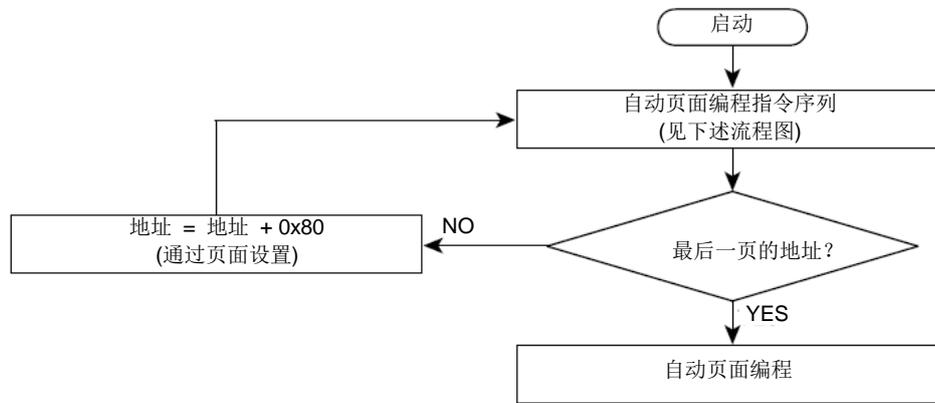


图 22-4 自动程序流程图

22.2.7.2 自动擦除

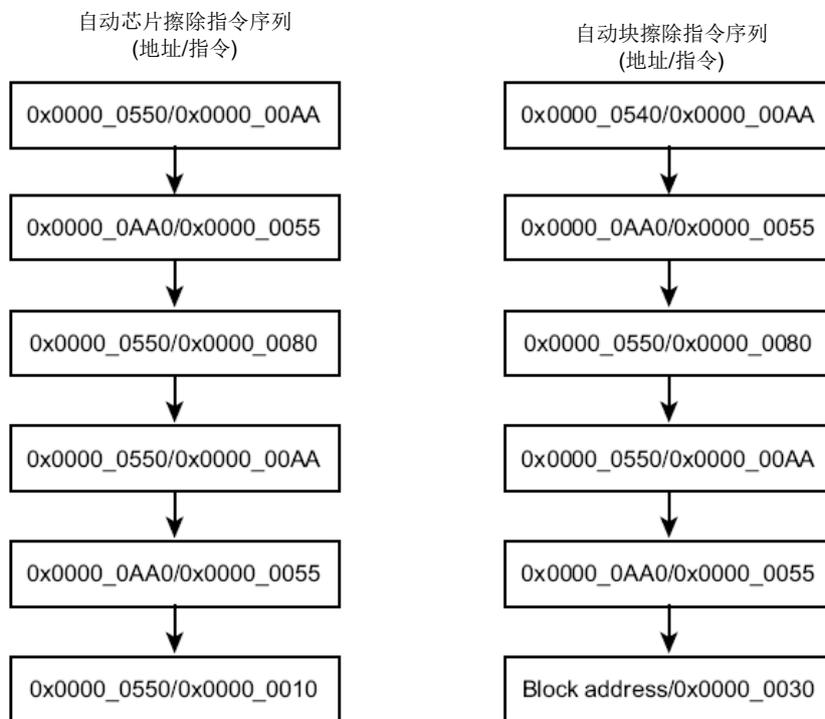
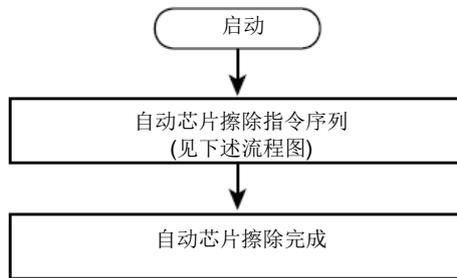


图 22-5 自动擦除流程图

22.3 如何利用单模式为闪存编程

单启动模式采用一包含闪存重新编程的内置BOOT ROM程序。此模式下，BOOT ROM映射到包含中断扇区表的区域，而闪存映射到BOOT ROM区域以外的其它地址区域。

在启动模式下，利用序列指令/数据传输对闪存进行编程。在该装置的串行信道(UART)已连接至外部主机的情况下，一个重新编程程序会被从外部主机复制到内置RAM。执行RAM中的重新编程路径对闪存重新编程。有关与主机通信的详细说明，请遵循下文所述的协议。

即使在单启动模式下，也不要生成复位以外的中断/错误，以免出现异常程序终止。

为确保单芯片模式(正常运行模式)中闪存内容的安全，一旦重新编程完成，建议保护相应的闪存块，防止在后续单芯片操作期间发生意外消磁。

22.3.1 模式设置

执行在板编程时，本装置在单启动模式下启动。以下设置用于单启动模式设置。

$\overline{\text{BOOT}} = 0$

$\overline{\text{RESET}} = 0 \rightarrow 1$

当 $\overline{\text{BOOT}}$ 引脚设置为上述地址时，将 $\overline{\text{RESET}}$ 引脚设置为"0"。然后是否 $\overline{\text{RESET}}$ 引脚，装置将在单启动模式下启动。

22.3.2 接口规格

本节对单引导模式下的UART通信格式进行说明。该串行操作支持UART(异步通讯)模式。在执行板上编程之前，还需设置程序控制器的通讯格式。

UART通讯

通信信道:信道 0

串行传输模式:UART(同步)，半双工，LSB优先

数据长度:8-位

奇偶检验位:无

停止位: 1-位

波特率:随机波特率

启动程序作为一初始条件操作时钟/模式控制块设置。块初始设置详情见"时钟/模式控制"。

如"22.3.5.1 串行运行模式测定"中所述，可通过该 16 位定时器(TMRB)测定波特率。在确定波特率时，通过1/16期望波特率值执行通讯。因此，通讯波特率必须在可测量范围内。定时器计数时钟按 $\Phi T1$ ($f_c/2$)运行。

I/O接口模式的握手引脚在接收状态下输出"低"等待，在传输状态下输出"高"电平。通讯前检查握手引脚，且必须遵循通讯协议。

表 22-10 给出了该启动程序所使用的引脚。不通过启动程序使用这些引脚以外的引脚。

表 22-10 引脚连接

引脚		接口
		UART
模式设置引脚	$\overline{\text{BOOT}}$	o
RESET引脚	$\overline{\text{RESET}}$	o
通讯引脚	TXD0 (PE0)	o
	RXD0 (PE1)	o

o: 已使用 x: 未使用

22.3.3 对内存存储器的限值

注意，单启动模式对内置RAM与内置闪存带来了一些限制，如表 22-11 所示。

表 22-11 单启动模式中对存储器的限制

存储器	限值
内部RAM	启动程序通过0x2000_0000~0x2000_03FF将寄存器作为工作区使用。通过RAM地址末端存储程序0x2000_0400。该程序的START地址必须为偶地址。
内部闪存	将下属地址进行分配已存储ID信息和密码。不建议按字啊书地址存储程序。 0x3F81_FFF0~0x3F81_FFFF

注:当某一密码为已擦除的数据(0xFF)时，由于某一容易猜到的密码而难以保护数据安全。即使单启动模式未使用，建议设置一独特的值作为密码。

22.3.4 操作指令

启动程序带有下述操作指令。

表 22-12 操作指令数据

操作指令数据	运行模式
0x10	RAM传输
0x40	闪存芯片擦除保护位擦除

22.3.4.1 RAM 传输

RAM传输为传输自控制器向内置RAM的数据。传输正常完成后，某一程序启动。用户程序和使用除启动程序用0x2000_0000 ~0x2000_03FF外0x2000_0400或后续寄存器地址。CPU将启动自RAM存储启动地址的操作。该起始地址必须是偶地址。

该RAM传输功能启用用户特定的在板编可编程制。在通过用户程序执行板上编程时，需使用 22.2.6 所述的闪存指令序列。

22.3.4.2 闪存芯片擦除和保护位擦除

无论写入/擦除保护或安全状态如何，闪存芯片擦除和保护位擦除指令擦除闪存寄存器的整个块，并写入/擦除所有块的保护。

22.3.5 无关指令的公共操作

本章节描述了执行启动程序时的公共操作。

22.3.5.1 串行运行模式判定

当控制器通过UART进行通讯时，以期望的波特率将第 1 个位设置为 0x86。图 22-6 给出了各种情况下的波形。

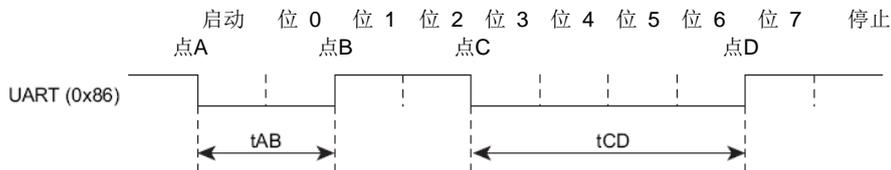


图 22-6 串行运行模式确定数据

图 22-7 给出了启动程序的流程图。利用 16-位时序器(TMRB)与 t_{AB} 、 t_{AC} 和 t_{AD} 的时间，复位后串行运行模式测定数据(0x86)的第一个字节可用。在图 22-7 中，CPU负责监视该接收引脚的电平，并可在接收引脚的电平被改变时，立即获取一个时序器值。因此， t_{AB} 、 t_{AC} 和 t_{AD} 的定时器值有误差容限。此外，注意如果该传输以高波特率进行，CPU可能无法确定接收引脚引脚的电平。

图 22-8 中的流程图表明，已确定串行运行模式，并已确定该接收引脚持续时间的长短。若长度为 $t_{AB} \leq t_{CD}$ ，串行运行模式判定为UART 模式。 t_{AD} 时间用于判定自动波特率设置是否启用。注意， t_{AB} 、 t_{AC} 和 t_{AD} 的定时器值有误差容限。如果波特率高而工作频率低，则各定时器值会变小。这样会生成非预期判定终止。(为防止此问题发生，在编程路径内重新设置UART。)

例如，在采用UART模式时，如该时间拟接收来自目标板的回波(0x86)，则控制器应允许存在某一超时周期。控制器如果不能在允许的时间内得到回应，应放弃通讯。

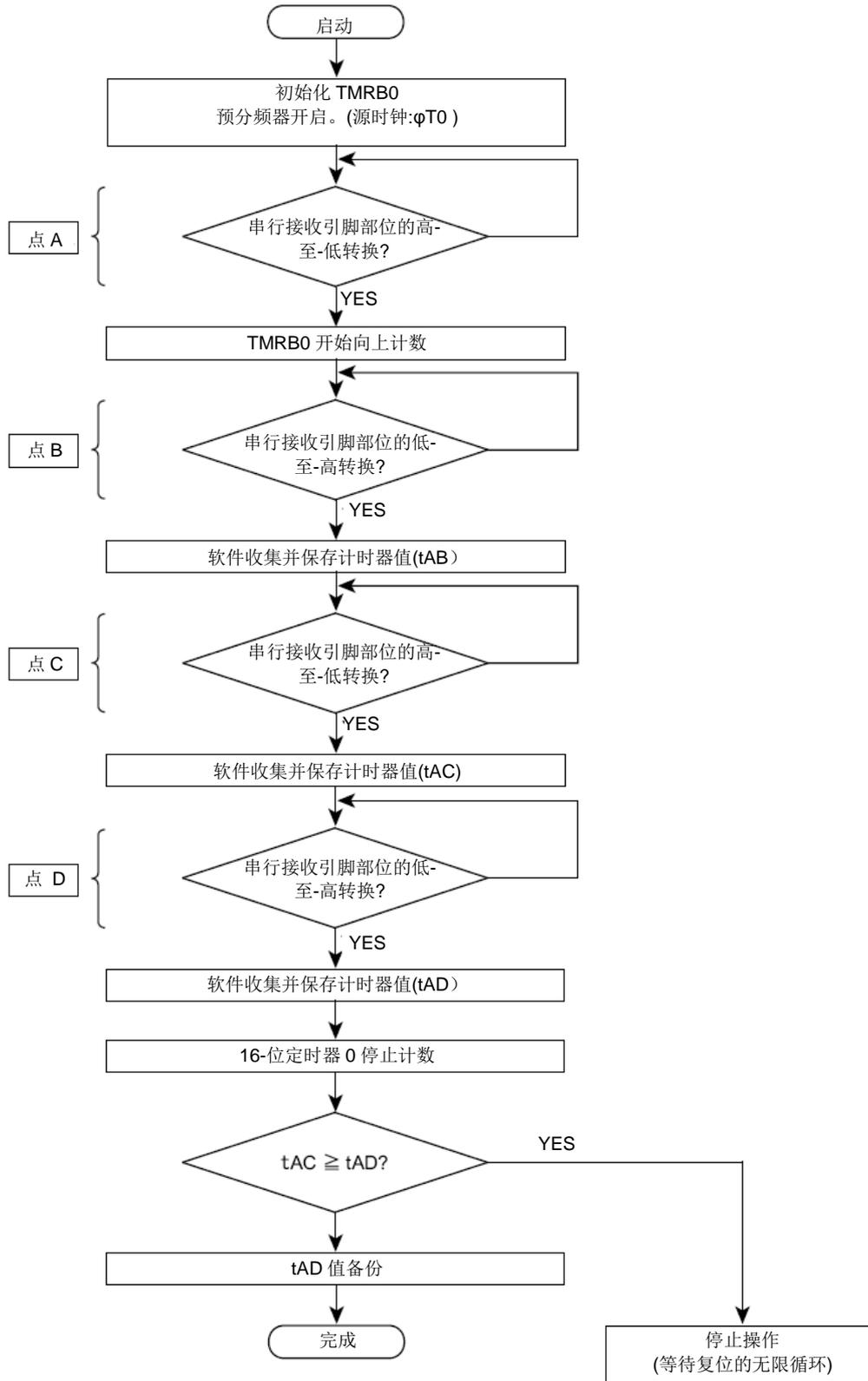


图 22-7 串行运行模式接收流程图

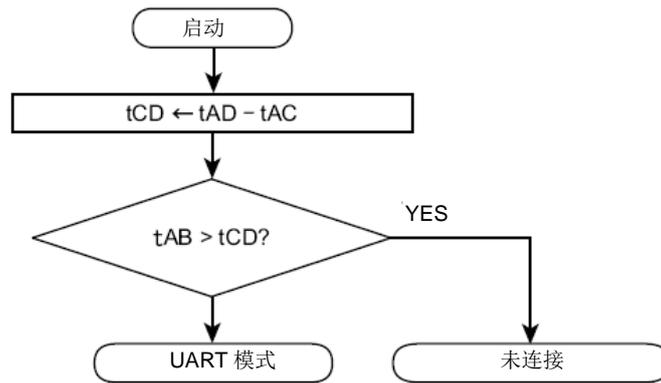


图 22-8 串行运行模式确定流程图

22.3.5.2 应答数据

启动程序指定的代码里代表在处理状态并将处理状态发生至控制器。表 22-13至表 22-16 给出了对各接收数据的确认响应值。

在表 22-14至表 22-16 中，确认响应的上部四位字节等于操作指令数据的上部四位字节。第3 个位表示一接收错误。指令第 0 位表示某一无效操作指令错误，一校验和错误或密码错误。第 1 位和第 2 位始终为 "0"。在I/O接口模式下不执行错误检查。

表 22-13 对串行操作确定数据的收到应答

传输数据	说明
0x86	判定可进行UART通讯(注)

注: 当串行操作判定为UART时，如果波特率设置判定为不接受，则启动程序终止而不会发回任何响应。

表 22-14 对操作指令数据的ACK应答

传输数据	说明
0x?8(注)	操作指令数据中发生接收错误
0x?1(注)	正常接收一未定义的操作指令数据。
0x10	判定为一个RAM传输指令
0x40	判定为一盒闪存芯片擦除指令

注: ACK应答数据上 4 位与前一个指令数据的上 4 位相同。

表 22-15 对检验和数据的收到应答

传输数据	说明
0xN8 (注)	发生接收错误。
0xN1 (注)	出现CHECK SUM 或密码错误。
0xN0 (注)	CHECK SUM 值正确。

注: ACK应答数据上 4 位与操作指令数据的上 4 位相同。

表 22-16 对闪存片擦除与保护位擦除操作的ACK应答

传输数据	说明
0x54	判定为擦除启用指令
0x4F	擦除指令完成。
0x4C	擦除指令异常终止。

注:即使在某条擦除指令被正常执行时,ACK响应也可能返回一条"否定确认"信息。检查该FCSR<RDY/BSY>,以确认该指令序列结束,然后保持200 μs或以上,然后再次确认该清除状态。

22.3.5.3 密码判定

启动程序利用一下区域来判定是否需要密码或作为密码使用。

区域	地址
密码要求判定	0x3F80_FFF0(1 字节)
密码区	0x3F80_FFF4~0x3F80_FFFF(12 字节)

无论是否需要判定数据,RAM传输指令均执行密码验证。闪存芯片擦除或保护位擦除指令仅在将需要判定判定为"要求"时才执行密码验证。

密码要求设置	数据
需要密码	除 0xFF外
无密码	0xFF

当将密码设置为0xFF(擦除数据)时,由于某一易猜出密码而难以保护数据安全。即使未使用单启动模式,建议设置一个独特的值作为密码。

(1)采用RAM传输指令的密码验证

如果所有这些地址位置均包含相同的数据字节(0xFF除外),则应将该状态确定为图 22-9 所示的密码区错误。在这种情况下,引导程序会根据检查和值的第 17 个字节(不考虑密码验证),返回到错误确认(0x11)。

启动程序 验证接收数据的第 5~16 位(密码数据)。当所有 12 位不匹配时,发生密码错误。当判定出密码错误时,至CHECK SUM数据的第 17 位为一密码错误。

即使安全功能启用,也可进行密码验证。

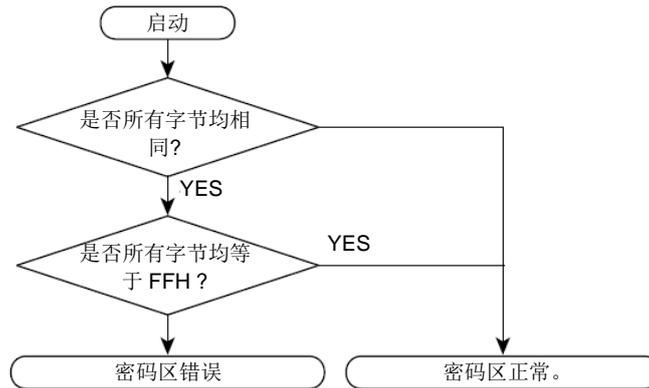


图 22-9 密码区检查流程图

(2) 对闪存芯片擦除与保护位擦除指令的密码验证

当某密码在图 22-10所示的擦除密码必要性确定区内被启用，且各密码均为完全相同的数据时，就会发生一个密码区错误。CHECK SUM0x41 当判定出密码区域错误时，无论是否进行密码验证，一至CHECK SUM第 17 位的ACK应答均发送0x41。

启动程序 验证接收数据的第 5~16 位 (密码数据)。当所有 12 位不匹配时，发生密码错误。当判定出密码错误时，至CHECK SUM数据的第 17 位为一密码错误。

即使安全功能启用，也可进行密码验证。

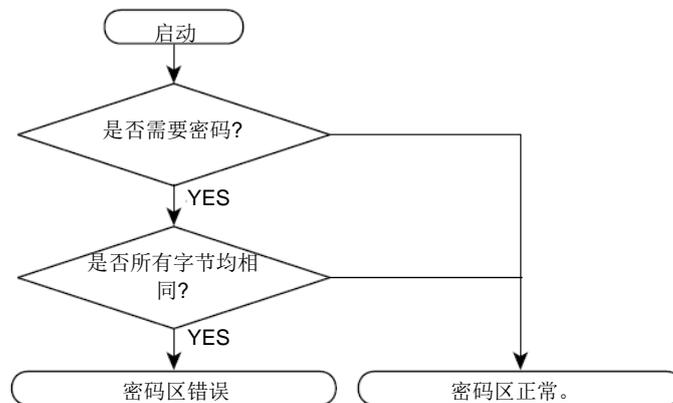


图 22-10 密码区检查流程图

22.3.5.4 CHECK SUM 计算

校验和用 8-位加至传输的数据，减去进位，再加上总和的两个的补数计算而得。在传输校验和位上，控制器必须执行相同的校验和操作。

CHECK SUM举例

计算0xE5和0xF6串行校验和时，执行 8-位加。

$$0xE5 + 0xF6 = 0x1DB$$

加上至下 8-位总和的各补数即为校验和值。因此，该启动程序会向控制器发送0x25。

$$0 - 0xDB = 0x25$$

22.3.6 在 RAM 传输下的传输格式

本章节说明了RAM传输指令格式。表中所示传输方向如下：

传送方向(C→T):控制器至TMPM375FSDMG

传送方向 (C←T):TMPM375FSDMG至控制器

传送字节数	传输方向	传输数据	说明
1	C→T	串行运行模式和波特率设置	发送数据以确定串行运行模式。有关模式确定的详细说明,见"22.3.5.1 串行运行模式的确定"。
		UART模式 0x86	发送0x86.如已确定为UART模式,则该程序会确定是否可采用波特设置。否则,程序停止且通讯关闭。
2	C←T	串行运行模式ACK应答	传输数据的第 2 位为对应串行操作设置模式数据对应的第 1 位ACK应答数据。当可进行设置时,设置SIO/UART。在传输缓存被写入数据前接收启用时序得到设置。
		[UART 模式] 正常状态:0x86	当可判断该设置时,发送0x86。否则,操作终止,不发回任何响应。当已完成的控制器发送数据的第 1 位时,要求以超时时间 (5 s)。当数据 (0x86)在超时内未正常接收时,不能进行通讯。
3	C→T	操作指令数据 (0x10)	发送RAM传输指令数据(0x10)。
4	C←T	操作指令ACK应答 正常状态:0x10 异常状态:0xX1 通讯错误:0xX8	操作指令ACK应答数据。 首先,检查接收数据的第 3 位是否错误。(仅UART 模式) 当存在接收错误时,发送ACK应答数据0xX8表示异常通讯并等待下一个操作指令(第 3 字节)。传输数据上 4 位未定义。(与操作指令之前的上 4 位相同)注: 在 I/O 接口中,不执行接收误差检查 此时,如接收数据的第三字节符合表 22-12 中的任一操作指令数据,则接收数据会被回应。在 RAM传输情况下,0x10 被回送,同时传输数据分至RAM传输 服务程序..如果该数据不符合表 22-12 中的指令指令,则发送一份ACK响应数据0xX1(其表示工作指令错误),并等待下一条工作指令(第三字节)。 传输数据的第 4 位未定义。(与操作指令之前的上 4 位相同。)
5~ 16	C→T	密码数据 (12-位) 0x3F80_FF04~0x3F80_FF0F	在密码区检查数据.有关密码区检查的详细说明,见 "22.3.5.3 密码确定"。 将闪存数据的0x3F81_FFF0~0x3F81_FFFF与接收数据的第 5~第 16 字节进行比较。当数据域地址不匹配时,以密码错误标志被设置。
17	C→T	CHECK SUM 值得第 5~16 字节	发送CHECK SUM值得第 5~16 位。 校验和计算详见 22.3.5.4。

传输字节数	传输方向	传输数据	说明
18	C←T	CHECK SUM值的ACK应答 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 ~ 第 17 字节存在错误。(仅UART模式) 若存在接收错误, 发送ACK应答数据0x18, 意为通讯异常, 等待下一操作指令 (第 3 字节)。 然后, 检查CHECK SUM数据的第 17 字节。若错误存在, 发送0x11, 等待下一操作指令 (第 3 字节)。 最后, 检查密码验证结果。若存在密码错误, 发送ACK应答数据 0x11 意为密码错误, 等待下一操作指令 (第 3 字节)。 若所有步骤正常结束, 发送正常ACK应答数据 0x10。
19	C→T	RAM存储开始地址 31 ~ 24	发送块传输开始地址, 用于RAM存储。第 19 字节对应地址的第 31 ~ 24 位。第 22 字节对应地址的第 7 ~ 0 位。 通过RAM的最后地址, 将该地址规定给以下地址即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	设置字节数, 以进行块传输。第 23 字节对应传输位的第 15 位~第 8 位。第 24 位对应传输位的第 7 位~第 0 位。规定拟存储于自RAM地址 0x2000_0400 直至最后地址的数据。
25	C→T	CHECK SUM 数值的第 19 ~ 24 字节数	发送CHECK SUM值的第 19 ~ 24 字节。
26	C←T	CHECK SUM值的ACK应答 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 ~ 25位存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK应答数据 0x18, 意为通讯异常, 等待下一操作指令 (第 3 字节)。 然后, 检查CHECK SUM数据的第 25 字节。若错误存在, 发送0x11, 等待下一操作指令 (第 3 字节)。 若所有步骤正常结束, 发送正常ACK应答数据 0x10。
27 ~ m	C→T	RAM已存数据	发送用于RAM已存储数据, 于第23字节~24位中规定数据的相同字节。
m+1	C→T	CHECK SUM 数值的 27 ~ m位	发送CHECK SUM值的第 27~ m 字节
m+2	C←T	对CHECK SUM数值的ACK应答 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 ~ m+1 字节存在错误。(仅限于UART模式) 若存在接收错误, 发送ACK应答数据0xX8, 意为通讯异常, 等待下一操作指令 (第 3 字节)。然后, 检查CHECK SUM数据的m+1位。若错误存在, 发送 0x11, 等待下一操作指令 (第 3字节)。若所有步骤正常结束, 发送正常ACK应答数据 0x10。
-	-	-	若ACK应答数据的第m+2 字节为正常ACK应答数据, 传输数据分至第19 字节~ 22 字节规定的地址。

22.3.7 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指令。表中所示传输方向如下：

传送方向(C→T):控制器至TMPM375FSDMG

传送方向 (C←T):TMPM375FSDMG至控制器

传输字节数	传输方向	传输数据	说明
1	C→T	串行运行模式和波特率设置	发送数据以确定串行运行模式.有关模式确定的详细说明,见"22.3.5.1 串行运行模式的确定".
		UART模式 0x86	发送0x86.若UART模式确定,检查是否可进行波特率设置.若否,则进行停止通讯操作。
2	C←T	串行运行模式ACK应答	传输数据的第 2 位为对应串行操作设置模式数据对应的第 1 位的ACK 应答数据。当可进行设置时,设置SIO/ UART。在传输缓存被写入数据前接收启用时序得到设置。
		[UART 模式] 正常状态:0x86	当可判断该设置时,发送0x86。否则,操作终止,不发回任何应答。当已完成的控制器发送数据的第 1 位时,要求一超时间(5 s)。当数据 (0x86) 在超时内未正常接收时,不能进行通讯。
3	C→T	操作指令指令数据(0x40)	发送闪存芯片擦除和保护位擦除指令数据(0x40)。
4	C←T	对操作指令的ACK应答 正常状态:0x40 异常状态:0xX1 通讯错误:0xX8	操作指令ACK应答数据。 首先,检查接收数据的第 3 位是否出错。(仅UART模式) 当存在接收错误时,发送ACK应答数据0xX8表示异常通讯兵等待下一个操作指令(第 3 字节)。传输数据的第 4 位未定义。(操作指令数据前立即执行与上 4 位相同的操作。)注意在I/O接口处,不执行接收错误检查。 此时,如接收数据的第三字节符合表 22-12 中的任一操作指令数据,则接收数据会被回应。如果该数据不符合表 22-12 中的指令,则发送一份ACK响应数据0xX1(其表示工作指令错误),并等待下一条操作指令。(第 3 位) 传输数据上 4 位未定义。(使用了和紧接操作指令前的上 4 位。)
5~16	C→T	密码数据 (12-位) 0x3F80_FF04~0x3F80_FF0F	若密码必要性设置为"无",该数据为虚拟数据。 若密码必要性设置为"必要",检查密码区数据。有关密码区数据检查的方法,见"22.3.5.3 密码确定"。 将闪存数据的0x3F80_FFF0~0x3F80_FFFF与接收数据的第 5~第 16 字节按顺序进行比较。若数据不相匹配,设置密码错误标志。
17	C→T	CHECK SUM值的第5~16 字节	发送CHECK SUM值的第 5~16 字节。 有关检查和计算的方法,见"22.3.5.4 CHECK SUM计算"。

传输字节数	传输方向	传输数据	说明
18	C←T	对CHECK SUM值的ACK应答 正常状态:0x40 异常状态:0x41 通讯错误:0x48	若密码必要性设置为"无", 发送正常ACK应答数据 0x40。 如果密码必要性被设置为"必要", 则首先应检查第五字节至第十七字节接收数据中是否存在接收错误(仅适用于UART模式)。如果存在接收错误, 则发送一份ACK响应数据0x48(其表示异常通讯), 并等待下一条操作指令(第 3 字节)。 然后, 检查CHECK SUM数据的第 17 字节。若错误发生, 发送 0x41, 等待下一个操作指令(第 3 字节) 最后, 检查密码验证结果。若密码错误存在, 发送ACK应答数据 0x41, 意为密码错误, 等待下一个操作指令(第 3 字节) 若所有步骤正常结束, 发送正常ACK应答数据 0x40。
19	C→T	擦除启用指令数据 (0x54)	发送启用指令数据(0x54)。
20	C←T	对擦除启用指令的ACK应答 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位存在错误。若接收错误存在, 发送ACK应答数据(第 3 位) 0x58, 意为通讯异常, 等待下一个操作指令(第 3 字节)。 然后, 若接收数据的第 19 字节和擦除启用指令不符, 接收数据获应答(正常ACK应答数据)。这种情况下, 0x54得到应答, 传输数据分至闪存芯片擦除例行程序。 若数据和擦除启用指令不符, 发送ACK应答数据 (第 0 位) 0xX1, 等待下一个操作指令。传输数据的第 4 位未定义。(使用了和紧接操作指令前的上 4 位。)
21	C→T	对擦除指令的ACK应答 注1: 正常状态:0x4F 异常状态:0x4C	若操作正常完成, 结束码(0x4F)被返回。 若擦除错误发生, 错误码(0x4C)返回。
-	-	-	等待下一个操作指令。

注 1:即使在某条擦除指令被正常执行时, ACK响应也可能返回一条"否定确认"。检查该FCSR<RDY/BSY>, 以确认该指令序列结束, 然后保持200 μs或以上, 然后再次确认该清除状态。

22.3.8 启动程序 WHOLE 流程图

本节显示启动程序的整个流程图。



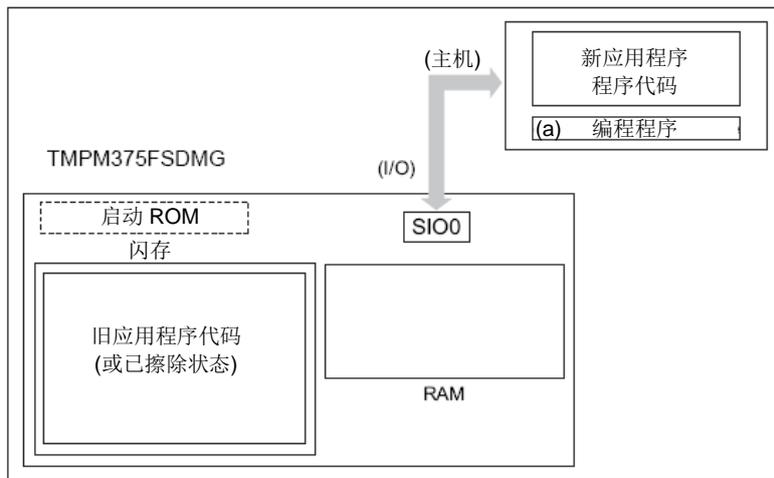
图 22-11 启动程序完整流程图

22.3.9 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。

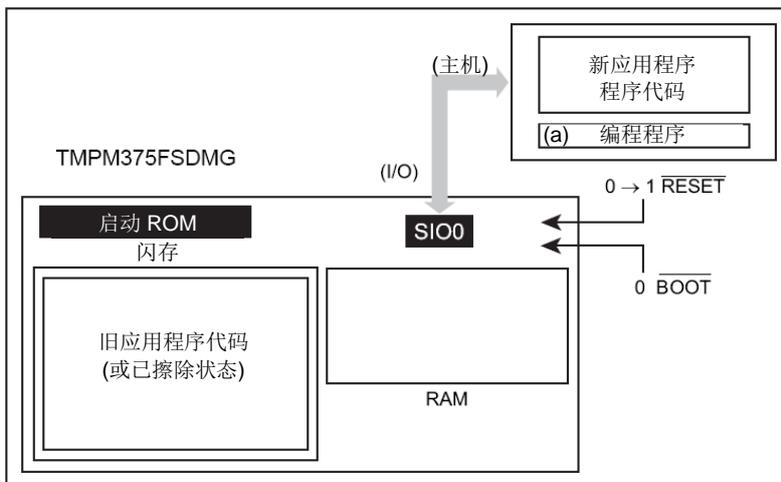
22.3.9.1 步骤-1

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) 被传输，SIO0必定已连接至外部主机。编程程序 (a) 在主机上已准备好。



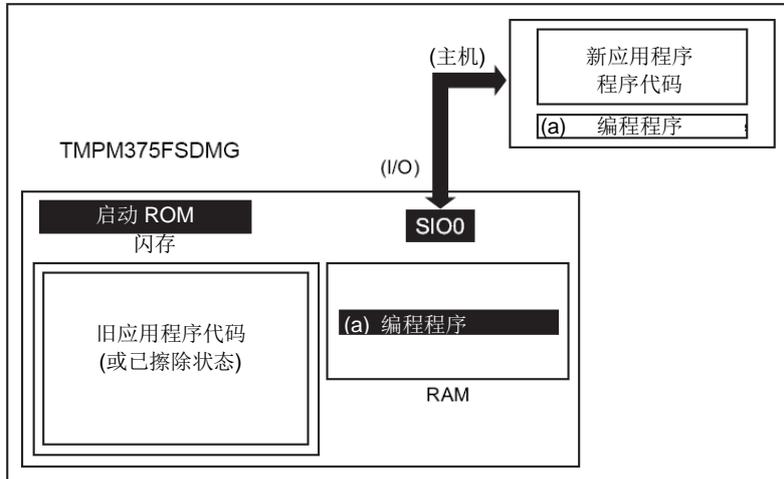
22.3.9.2 步骤-2

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)进行传输。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。)



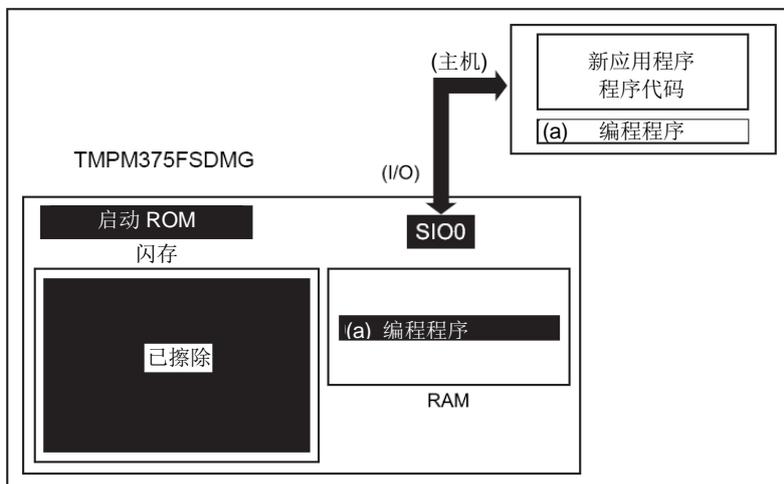
22.3.9.3 步骤-3

如果密码验证完成，则引导程序会将编程例行程序(a)从主机传送到片上RAM中。编程程序必须在0x2000_0400~RAM终端地址之间的范围内进行储存。



22.3.9.4 步骤-4

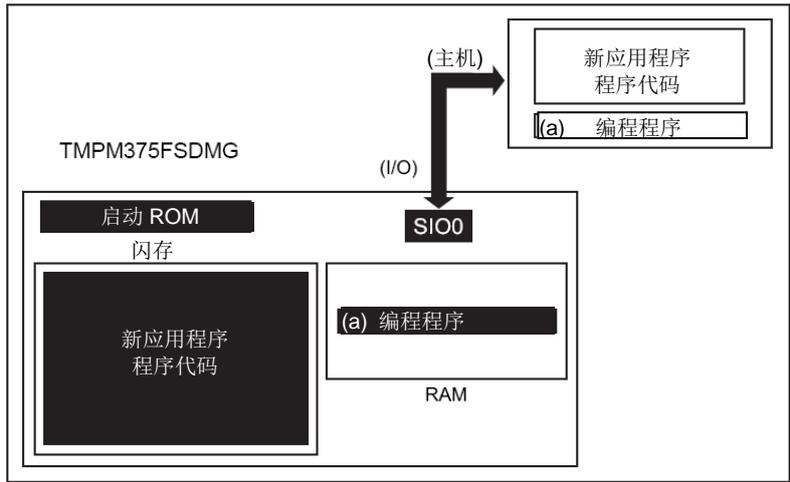
启动程序跳转至芯片上RAM内的编程程序(a)，以擦除含旧应用程序代码的闪存块。使用块擦除或芯片擦除指令



22.3.9.5 步骤-5

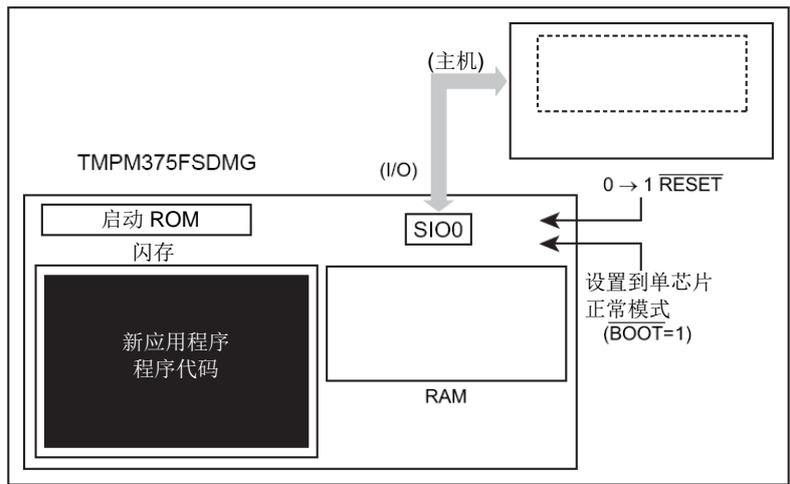
启动程序执行编程程序(a)自主机下载新应用程序代码，并将其编程至已擦除闪存区。编程完成时，须在用户程序中设置写入或擦除该闪存区保护。

以下示例中，新程序代码通过相同SIO0信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，可自由更换传输路径和传输源。创建硬件板卡并编程程序以满足你的特定需求。



22.3.9.6 步骤-6

闪存编程完成时，令板卡断电，并断开主机和目标板卡之间的连接线。再次通电，令装置以单芯片(正常)模式重启以执行新程序。



22.4 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。在用户应用程序上的闪存程序代码的数据传输总线不同于串行I/O时，就需要用到它。其以单芯片模式运行；因此必须从已在使用引导模式激活用户应用程序的标准模式，切换到编程闪存的用户引导模式。尤其是，在用户应用程序中增加了模式判断程序至复位服务程序。

需根据用户系统设置情况设置切换模式的条件。而且，用户独特编制的闪存编程程序需设置于新应用程序中。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。确保切勿生成复位外的中断/错误，以免用户启动模式中异常终止。

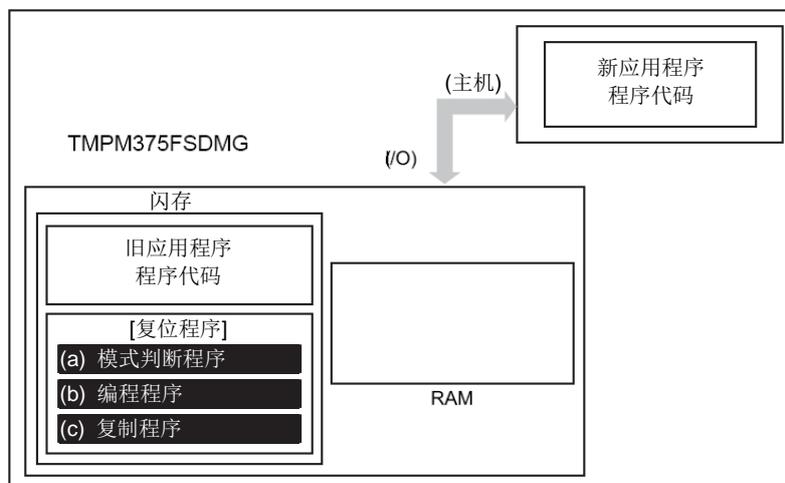
选取两种情况作为示例，如存储于闪存 (1-A)，传输自外部装置 (1-B)的重编程程序方法，下节对该步骤予以阐述。有关闪存程序/擦除的详细说明，见 "22.2 闪存的详细说明"。

22.4.1 (1-A)编程程序存储于闪存的步骤

22.4.1.1 步骤-1

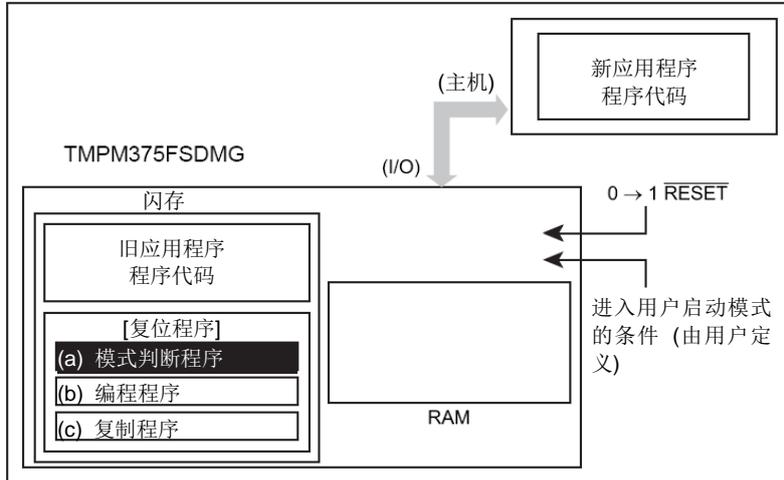
用户确定进入用户启动模式和拟用于传输数据的I/O总线的条件(如引脚状态)。然后创建适当的电路设计和程序。将装置安装于印刷电路板之前，用编程设备如闪存程序写入器将以下三个编程程序写入任意闪存块。

- (a) 模式确定例行程序: 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: 一个用于将(A)中所述数据复制到内置RAM或外部存储装置的程序



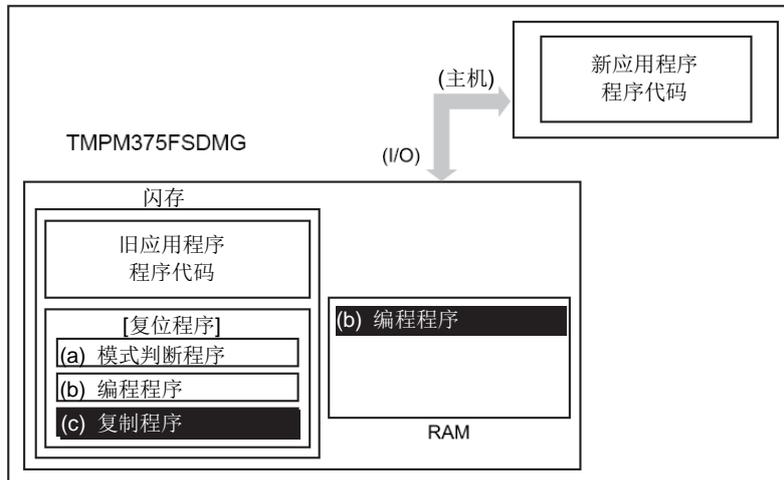
22.4.1.2 步骤-2

本节阐述了存储于复位程序的编程程序的情况。首先，复位例行程序确定并进入用户启动模式。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。



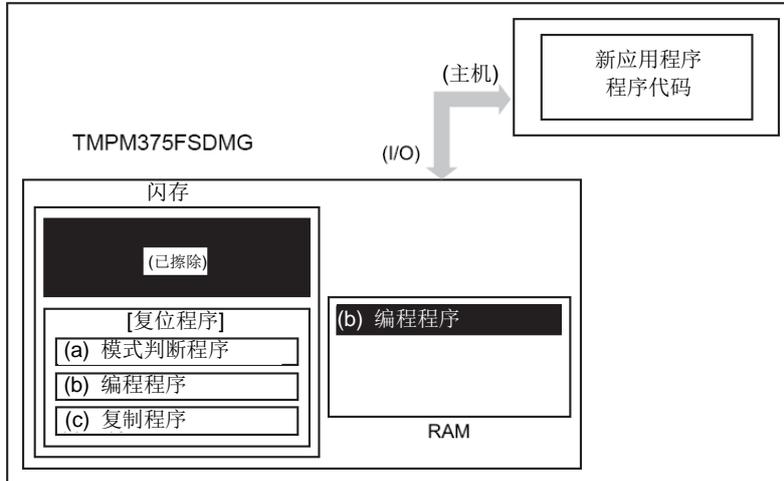
22.4.1.3 步骤-3

一旦该装置进入用户启动模式，即执行该复制例行程序(C)，将闪存编程例行程序(b)从主机控制器下载到内置RAM上。



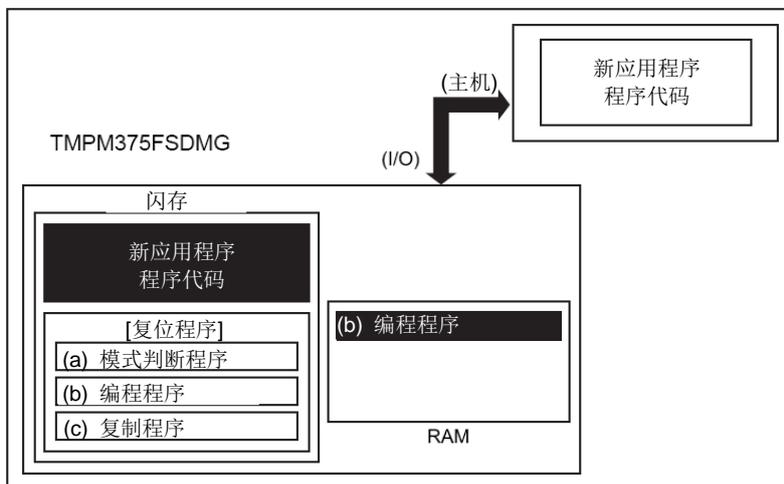
22.4.1.4 步骤.4

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。



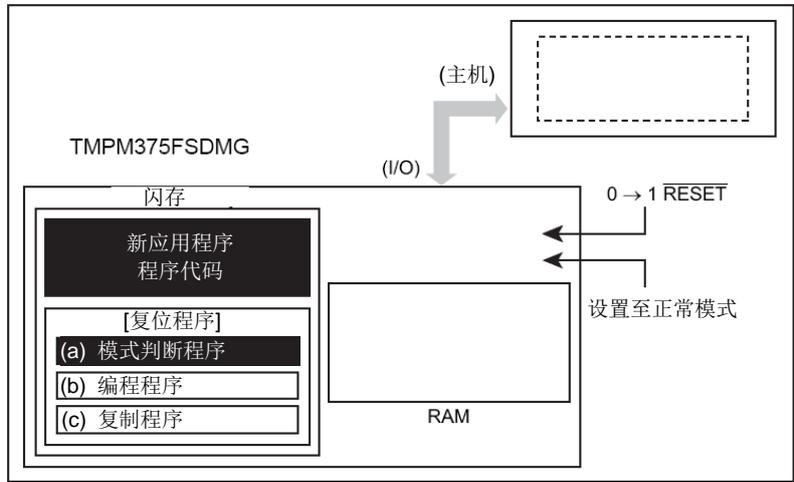
22.4.1.5 步骤-5

继续执行该闪存编程程序，从主机控制器下载新的程序数据，并将其编入已擦除的闪存块内。编程完成后，须设置用户程序区闪存块的写入/擦除保护。



22.4.1.6 步骤-6

设 $\overline{\text{RESET}}$ 为 "0"。一旦复位，闪存设置为正常模式。复位后，CPU 将随新应用程序启动。



22.4.2 (1-B)编程程序自外部主机进行传输的步骤

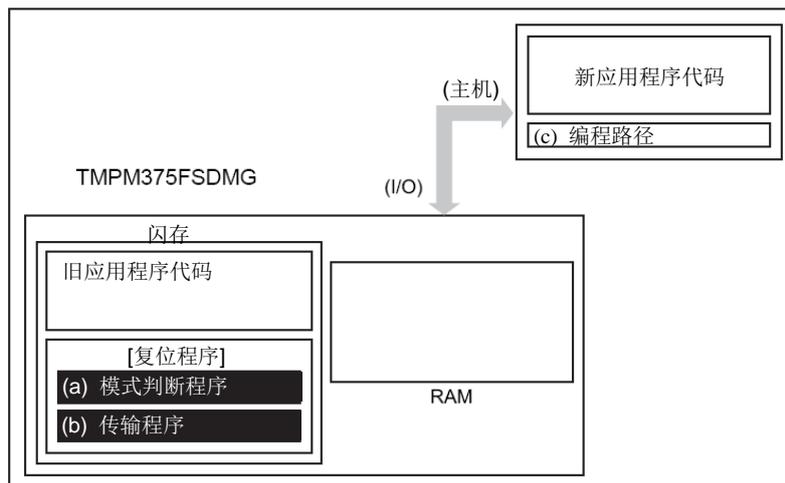
22.4.2.1 步骤-1

用户确定进入用户启动模式和拟用于传输数据的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。在将该装置安装到印刷电路板上之前,用闪存编写程序等编程设备,将以下两个程序的例程写入到任意闪存块。

- (a) 模式确定例行程序: 确定是否切换至重编程操作的程序
- (b) 传输程序自外部装置获取重编程程序的程序。

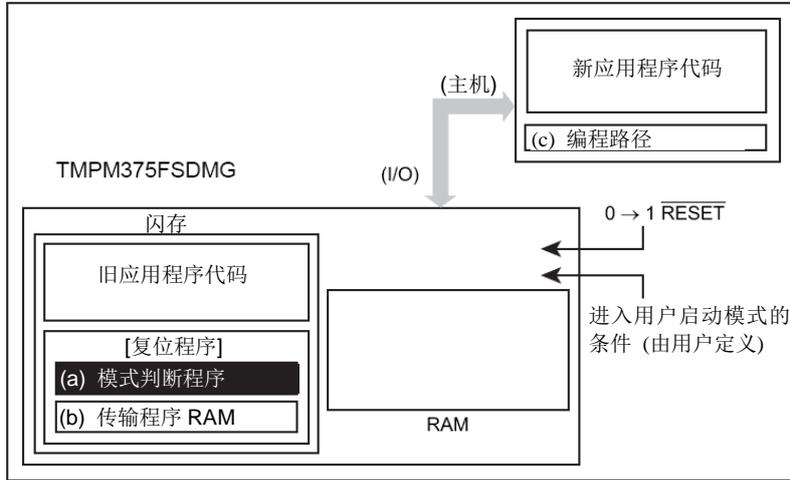
另外,准备须存储于主机控制器,如下所示的重编程程序。

- (c) 重编程程序对数据进行重编程的程序



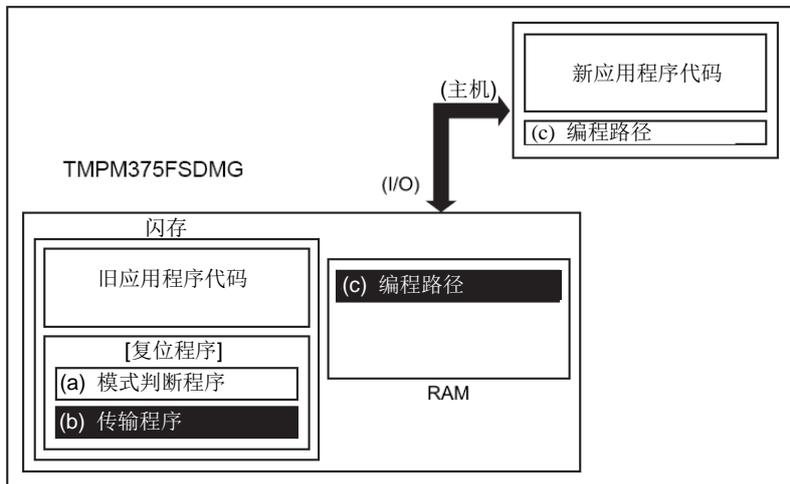
22.4.2.2 步骤--2

本节阐述了存储于复位程序的编程程序的情况。首先，复位例行程序确定并进入用户启动模式。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。



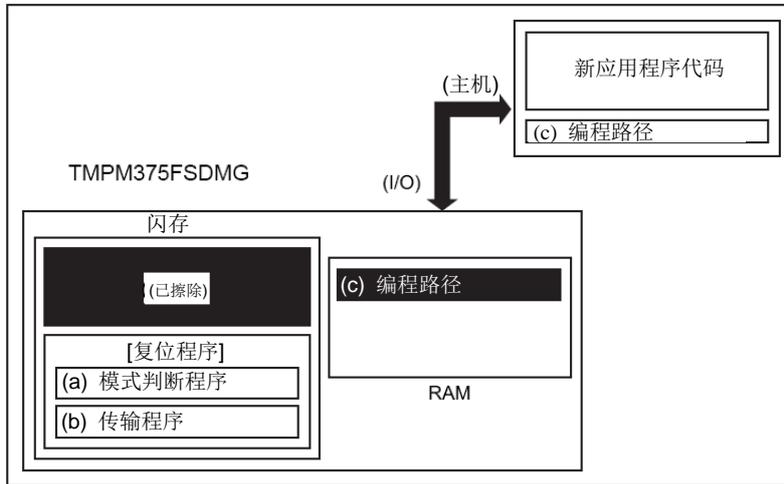
22.4.2.3 步骤-3

一旦该装置进入用户启动模式，即执行该输送例程(b)，将编程例程(c)从主机控制器下载到内置RAM上。



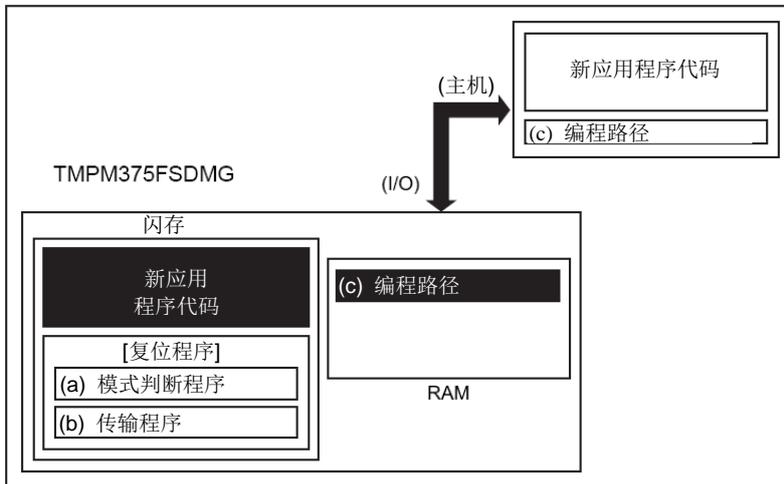
22.4.2.4 步骤.4

跳转到内置RAM中的该重新编程例程，以解除旧应用程序的写入/擦除保护，并擦除块单元中的闪存。



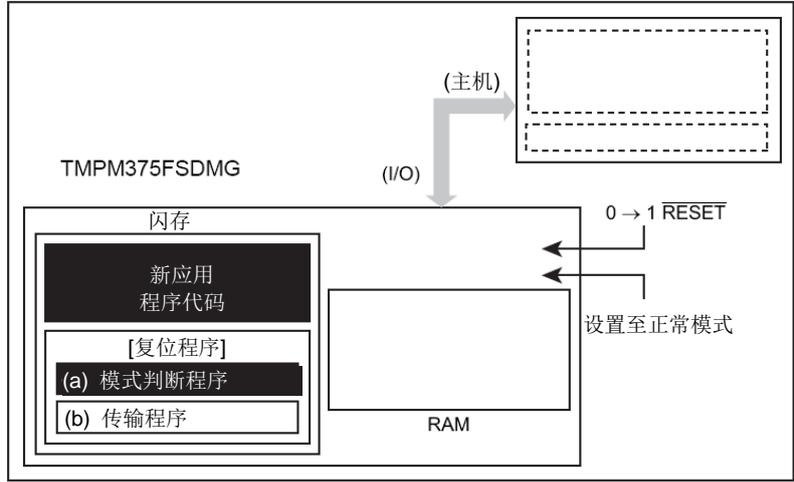
22.4.2.5 步骤.5

继续执行该闪存编程例程(c)，从主机控制器下载新的程序数据，并将其编入已擦除的闪存块内。在编程完成时，必须设置用户程序区中该闪存块的写入/擦除保护。



22.4.2.6 步骤-6

将 $\overline{\text{RESET}}$ 设置为 "0"。一旦复位，闪存设置为正常模式。复位后，CPU 将随新应用程序启动。



23. 调试接口

23.1 规范概述

该TMPM375FSDMG包含用于连接该调试接口的串行线JTAG调试端口(SWJ-DP)单元, 以及用于跟踪输出的嵌入式跟踪Macrocell™(ETM)单元。跟踪数据通过芯片上跟踪端口接口单元(TPIU)被输出到专用引脚(SWV)。有关SWJ-DP, ETM 和TPIU的详细资料, 见ARM公司发布的Cortex-M3技术参考手册。

23.2 SWJ-DP 的特点

SWJ-DP支持支持两针串行线调试端口(SWDCK, SWDIO)与JTAG调试端口(TDI, TDO, TMS, TCK)。

23.3 ETM 的特点

ETM支持SWV的跟踪输出。

23.4 引脚功能

调试接口引脚也能用作通用端口。PB3和PB4共用于JTAG调试端口功能和串行线调试端口功能。PB5在JTAG调试端口功能与SWV跟踪输出功能之间共用。

表 23-1 SWJ-DP, ETM功能

SWJ-DP 引脚名称	端口名 称	JTAG调试功能		SW调试	
		I/O	说明	I/O	说明
TMS/SWDIO	PB3	输入	JTAG试验模式 选项	I/O	串行线数据 输入/输出
TCK/SWCLK	PB4	输入	JTAG测试检查	输入	串行线时钟
TDO/SWV	PB5	输出	JTAG测试数据输出	(输入) (注 1)	(串行线查看器 (输出))
TDI	PB6	输入	JTAG测试数据输入	-	-

注:在启用SWV功能时

复位后, PB3, PB4, PB5和PB6被配置为调试端口功能引脚。其他调试接口引脚的功能需要按照要求进行编程。调试接口引脚可使用不用调试接口的通用端口。

下面的表 23-2 概括了调试接口引脚功能以及复位后涉及到的端口设置。

表 23-2 调试接口引脚与复位后的端口设置

初始 设置	端口 (位名称)	调试 功能	复位后的端口设置(-: 无寄存器)					
			功能 (PBFR)	输入 (PBIE)	输出 (PBCR)	开 漏 (PBOD)	上拉 (PBPUP)	下拉 (PBPDN)
DEBUG	PB3	TMS/SWDIO	1	1	1	0	1	0
DEBUG	PB4	TCK/SWCLK	1	1	0	0	0	1
DEBUG	PB5	TDO/SWV	1	0	1	0	0	0
DEBUG	PB6	TDI	1	1	0	0	1	0

使用低功耗模式时, 注意下列几点。

注 1:如果PB3和PB5被配置为调试功能引脚, 则继续启用输出(即使在STOP模式下), 而不考虑CGSTBYCR<DRVE>的设置。

注 2:如果PB4被配置为调试功能引脚, 则其会阻止低功耗模式完全生效。配置PB4, 使之能在该未使用该调试功能时用作通用端口。

23.5 与调试工具的连接

23.5.1 连接方式

有关调试工具的连接方式，见各制造商建议采用的方法。调试接口引脚带有上拉或下拉寄存器。在与上拉或下拉寄存器连接时，确信其正确设置。

23.5.2 在使用通用端口时

在调试时，不要通过程序将设置调试接口改为通用端口。此时，MCU不能控制收自各该调试工具的信号，且不能继续调试。应按照各调试接口引脚的用途，确认其设置。

表 23-3 调试接口

用法	使用调试接口(O:启用, -:停用)			
	TDI	TDO/SWV	TCK/ SWCLK	TMS/ SWDIO
JTAG+SW (RESET后)	O	O	O	O
JTAG+SW (非 $\overline{\text{TRST}}$)	O	O	O	O
JTAG + TRACE	O	O	O	O
SW	-	-	O	O
SW + SWV	-	O	O	O
停用调试功能	-	-	-	-

23.6 HALT 模式期间的外部设备运行

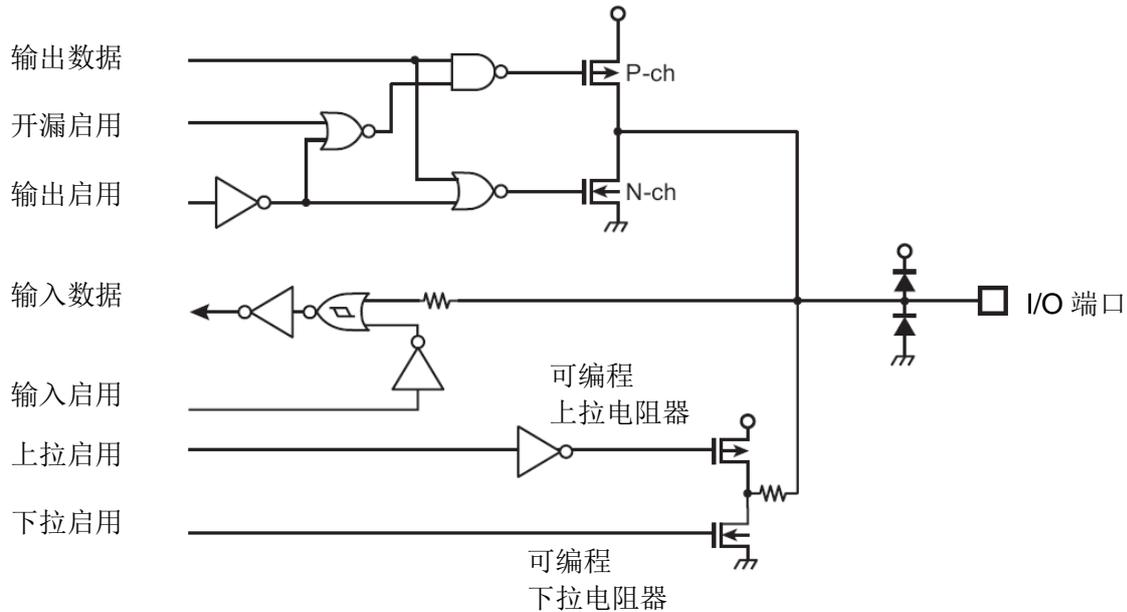
在调试期间断开时，Cortex-M3 CPU核心会进入HALT模式。看门狗定时器(WDT)自动停止计数。在HALT模式下，16 位定时器/计数器可规定该状态(继续运行或停止)。其它外部设备则继续运行。



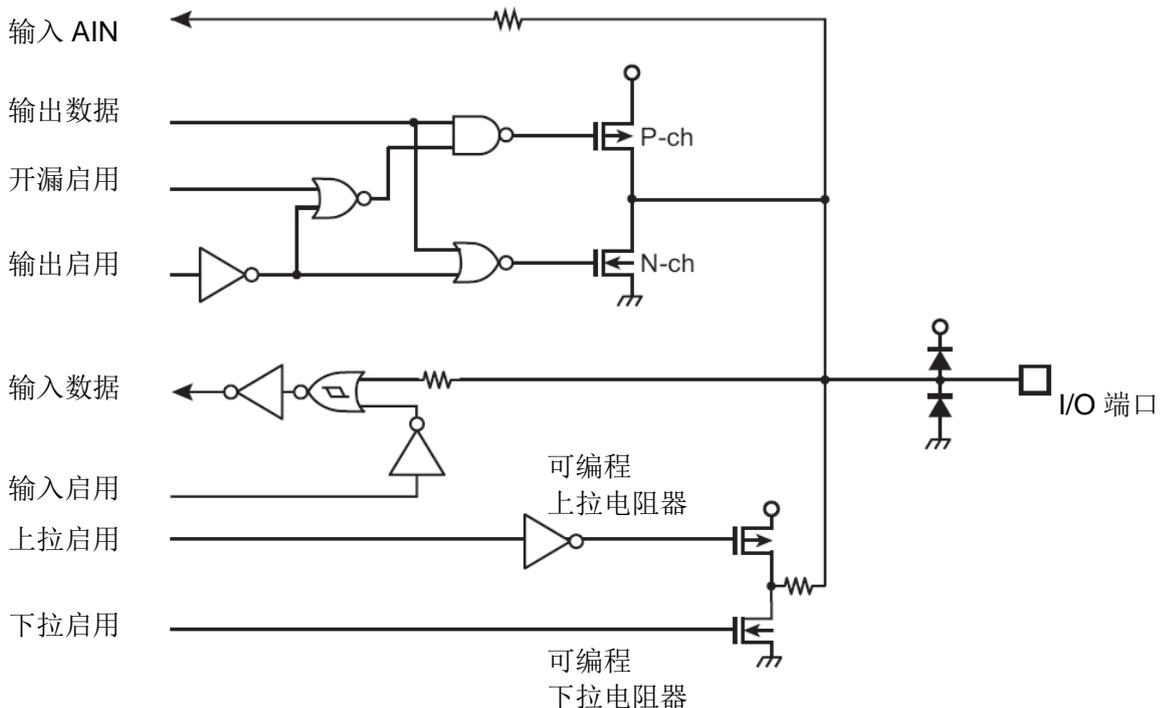
24. 端口部分等效电路示意图

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。其中已给出反馈电阻器和阻尼电阻器的典型值。

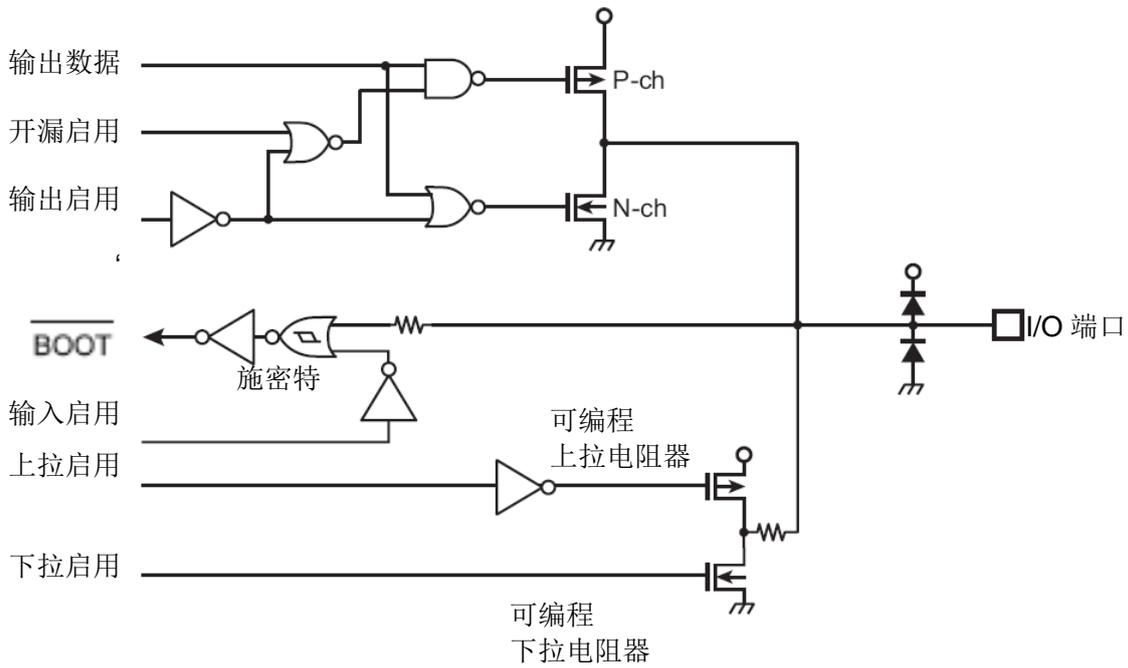
24.1 PB3~6, PE0~2, PG0~6



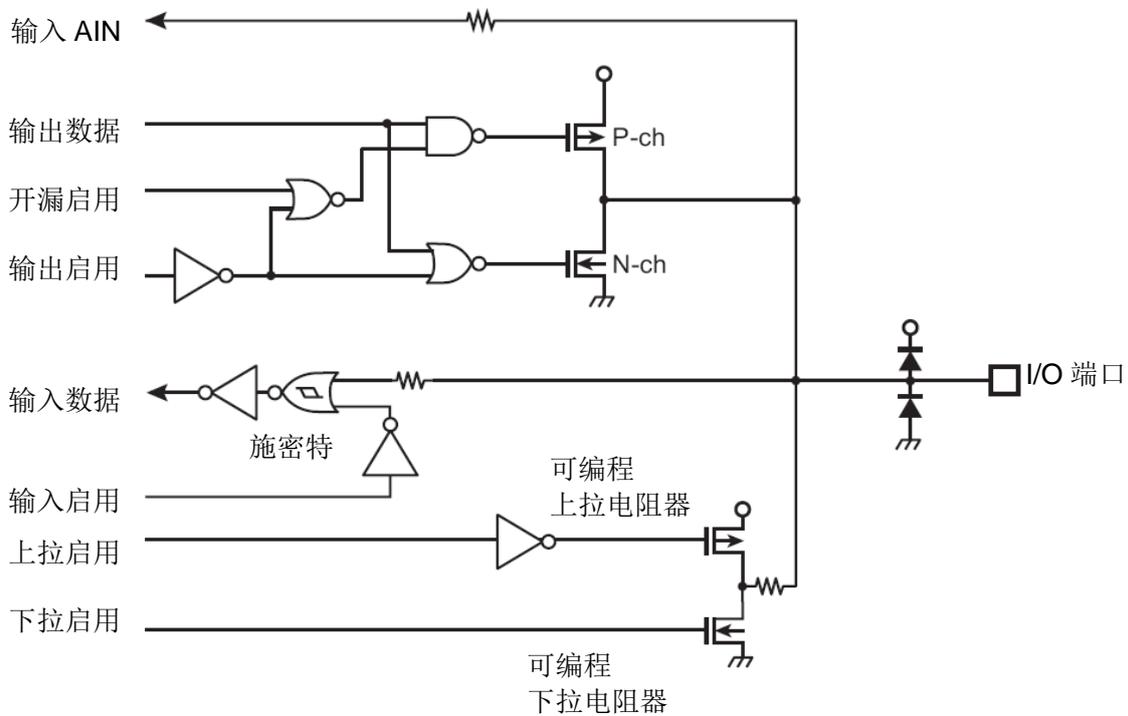
24.2 PJ6~7, PK0~1



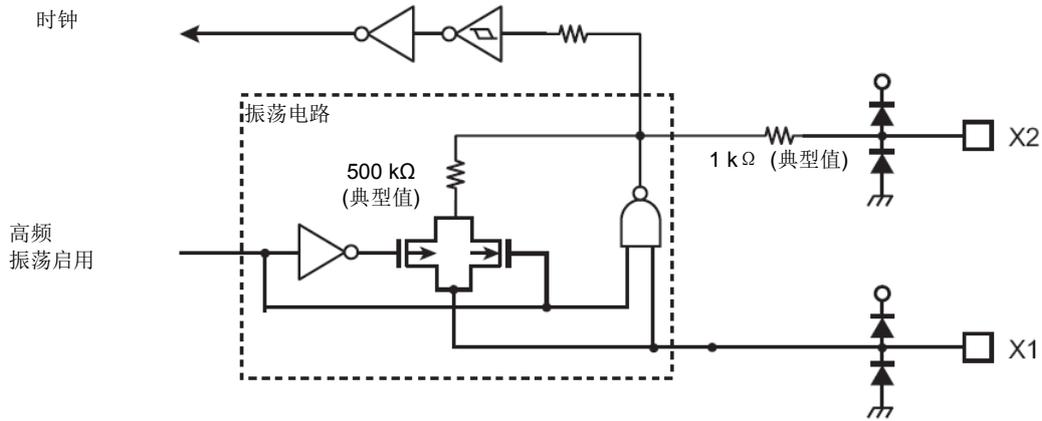
24.3 PF0



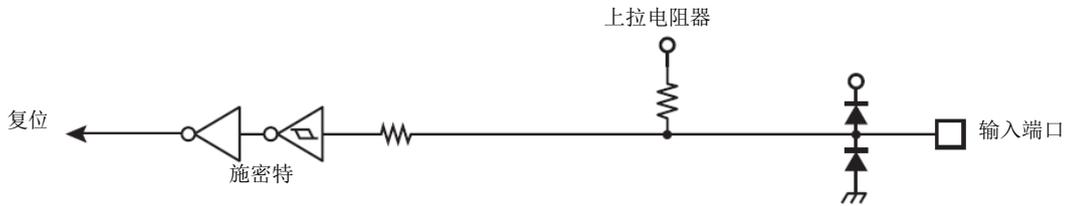
24.4 PM0~1



24.5 X1, X2



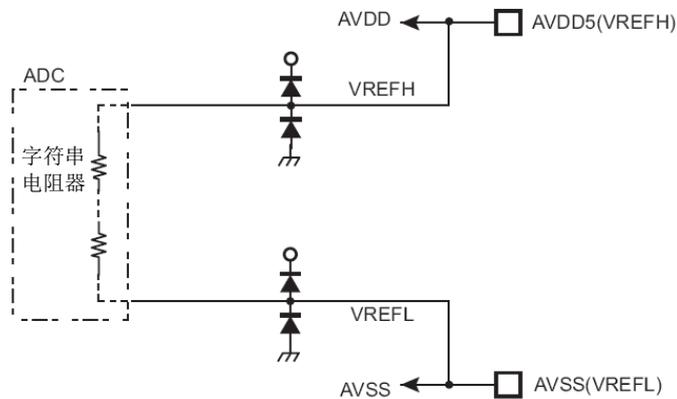
24.6 RESET



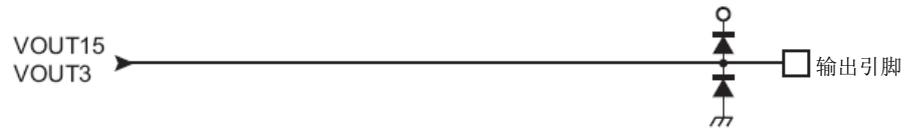
24.7 MODE



24.8 VREFHB VREFLB



24.9 VOUT15, VOUT3



25. 电气特性

25.1 绝对最大额定值

参数		符号	额定值	单元
电源电压		DVDD5B / AVDD5B / AMPVDD5	-0.3~6	V
电容器电压		VOUT15	-0.3~3	V
		VOUT3	-0.3~3.9	
输入电压		V _{IN}	-0.3~VDD+0.3(注 2)	V
低电平输出电流	每引脚	I _{OL}	5	mA
	合计	ΣI _{OL}	50	
高电平输出电流	每引脚	I _{OH}	-5	
	合计	ΣI _{OH}	50	
功耗		PD	350	mW
焊接温度(10 s)		T _{SOLDER}	260	°C
贮存温度		T _{STG}	-55~125	°C
工作温度	闪存W/E期间除外	T _{OPR}	-40~105	°C
	闪存 W/E时		0~70	

注:绝对最大额定值为最恶劣可能条件下不可超过的运行和环境条件极限值。装备制造商的设计,应在电流,电压,耗电,温度等方面不超过绝对最大额定值。暴露于以上所列情况下,可能导致装置受到永久性损坏或影响装置可靠性,可能增加由于IC爆炸和/或燃烧导致的人身伤害潜在风险。

注:VDD = DVDD5B / AVDD5B / AMPVDD5

25.2 直流电电气特性(1/2)

DVSS = DVSSB = AVSSB = 0V, Ta = -40 to 105 °C

参数		符号	额定值	最小	典型值(注1)	最大	单元
电源电压 (注 2)	DVDD5B	VDD	fOSC = 8或10 MHz fsys = 1~40 MHz	4.5	-	5.5	V
	AVDD5B						
	AMPVDD5						
电源电压 (在闪存写 入/擦除期 间)(注 2)	DVDD5B	VDD	fOSC = 8或10 MHz fsys = 1~40 MHz (Ta (°C) = 0~70)	4.5	-	5.5	V
	AVDD5B						
	AMPVDD5						
电源电压 (上电或下 电)(注 5)	DVDD5B	VDD	fOSC = 8或10 MHz fsys = 1~40 MHz	3.9	-	5.5	V
	AVDD5B						
	AMPVDD5						
低-电平输 入电压	施密特-输入	VIL1	VDD = 4.5V~5.5V (注 4)	-0.3	-	0.25 VDD	V
高-电平输 入电压	施密特-输入	VIH1	VDD = 4.5V~5.5V (注 4)	0.75V D	-	VDD+0. 3	V
VOUT15和VOUT3的电容 (注 3)		Cout	VOUT15, VOUT3	3.3	-	4.7	μF
低-电平输出电压		VOL	IOL = 1.6 mA VDD ≥ 4.5 V (注 4)	-	-	0.4	V
高-电平输出电压		VOH	I OH = -1.6 mA VDD ≥ 4.5V (注 4)	4.1	-	-	V
输入泄漏流		ILI1	0.0 ≤ VIN ≤ VDD (注 4)	-	0.02	±5	μA
输出泄漏流		ILO	0.2 ≤ VIN ≤ VDD -0.2 (注 4)	-	0.05	±10	
复位时的上拉寄存器		RRST	4.5 ≤ VDD ≤ 5.5 (注 4)	-	50	150	kΩ
可编程上拉/下拉电阻器		PKH	4.5 ≤ VDD ≤ 5.5 (注 4)	-	50	150	kΩ
施密特-触发端口		VTH	4.5 ≤ VDD ≤ 5.5 (注 4)	0.3	0.6	-	V
引脚 电容 (电源引脚 除外)	数字引脚	CI01	fc = 1 MHz	-	-	10	pF
	模拟引脚	CI02		-	-	30	
	PJ6, PJ7, PK0和 PK1						

注 1:除非另外注明, Ta =25°C, DVDD 5B=AVDD5B=5V。

注 2:必须给DVDD5B与AVDD5B提供相同电压。

注 3:应按照相同的电容值, 将VOUT15和VOUT3引脚连接到GND。IC外部无法从VOUT15和VOUT3获得供电。

注 4:VDD = DVDD5B = AVDD5B

注 5:其为上电或下电时的电压范围(在VLTD被禁用时)。在电源线为 3.9 V≤VDD <4.5 V的范围内, 不保证 12-位A/D转换器与交流电电气特性。

25.3 DC 电气特性(2/2)

DVDD5B=AVDD5B=4.5 V~5.5 V, Ta=-40~105°C

参数	符号	额定值	最小	典型值 (注 1)	最大	单元
NORMAL(注 2)齿轮 1/1	I _{DD}	f _{sys} = 40 MHz	-	15	25	mA
IDLE(注 3)齿轮1/1			-	5	15	
STOP			-	0.1	3	

注 1:除非另外注明, Ta =25°C, DVDD 5B=AVDD5B=5V。

注 2:IDD NORMAL:所有功能运行时均不带A/D和Op-amp。

注 3:IDD IDLE:所有外设功能均被停止。

25.4 12 位 ADC 电气特性

DVDD5B = AVDD5B / VREFHB = 4.5 V~5.5 V

DVSSB = AVSSB / VREFLB = 0V, Ta = -40~105 °C

参数	符号	额定值	最小	典型值	最大	单元
模拟参考电压(+)	VREFHB	-	-	AVDD	-	V
模拟输入电压	VAIN	-	AVSS	-	AVDD	V
电源电流	A/D转换	-	包括IREF	6.0	10.0	mA
INL错误	-	AIN 电阻 ≤ 600 Ω AIN 占空电容 ≥ 0.1 μF 转 换时间 ≥ 2 μs	-	-	± 8	LSB
DNL错误			-	-	± 5	
偏移错误			-	-	± 5	
满刻度错误			-	-	± 6	
总错误			-	-	-7~+11	

注:1LSB = (AVDD - AVSS)/4096 [V]

注:AVDD = AVDD5B, AVSS = AVSSB

注:该特性仅在ADC运行的条件下测量。

25.5 Op-Amps 电气特性

DVDD5B = AVDD5B / VREFHB = 4.5 V ~ 5.5 V

DVSSB = AVSSB / VREFLB = 0V, Ta = -40 ~ 105 °C

参数	符号	额定值(注 3)	最小	典型值	最大	单元
增量(注 1)	VGAIN	-	1.5	-	10	
输入电压范围	VAMPIN	-	(AVDD×0.1)/ VGAIN	-	(AVDD×0.9)/ VGAIN	V
放大器总输出错误	VAMP	-	-6 × VGAIN	-	+6 × VGAIN	mV
转换速率	V _{thr} (注 2)	5 pF, VGAIN = ×2.5	2	-	-	V / μs
	V _{thf}		1	-	-	
电源电 流	Op-Amp	-	-	4	6	mA

注:AVDD=AVDD5B=4.5~5.5 V, AVSS=AVSSB=0V

注 1:增量可通过寄存器设置在×2.5, ×3, ×3.5, ×4, ×6与×8中间选取。

注 2:信号响应速率表示放大器的输出达到AVDD-0.001×AVDD时的斜面。

25.6 AC 电气特性

25.6.1 AC 测量条件

AC测量条件

- 输出电平:高 = $0.8 \times VDD$ / 低 = $0.2 \times VDD$
- 输入电平:见DC电气特性中的低电平输入电压和高电平输入电压。
- 负荷能力: $CL=30 \text{ pF}$

注: $VDD = DVDD5B = AVDD5B$

25.6.2 串行信道时序(SIO)

25.6.2.1 I/O 接口模式

在下表中，字母x表示系统时钟的期限(fs_{sys})。因时钟齿轮功能编程而异。

(1)SCLK输入模式

[数据输入]

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
SCLK时钟高宽度(输入)	tSCH	4x	-	100	-	ns
SCLK时钟低宽度(输入)	tSCL	4x	-	100	-	
SCLK 周期	tSCY	8x	-	200	-	
有效数据输入← SCLK上升或下降(注 1)	tSRD	30	-	30	-	
SCLK上升或下降→ 输入数据保持(注 1)	tHSR	x + 30	-	55	-	

数据输出

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
SCLK时钟高宽度(输入)	tSCH	4x	-	120 (注 3)	-	ns
SCLK时钟低宽度(输入)	tSCL	4x	-	120 (注 3)	-	
SCLK 周期	tSCY	8x	-	240	-	
输出数据 ← SCLK上升或下降(注 1)	tOSS	$t_{scv}/2 - 3x - 45$	-	0 (注 2)	-	
SCLK上升或下降→ 输出数据保持(注1)	tOHS	$t_{scv}/2$	-	120	-	

注 1:SCLK上升或下降:关于SCLK的已编程主动沿测量

注 2:计算值应使用该范围没有扣减的SCLK周期。

注 3: t_{oss}给出了未经减去的最小值。

(2)SCLK输出模式

[数据输入]

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
SCLK 周期 (可编程)	t _{SCY}	4x	-	100	-	ns
输出数据 ← SCLK 上升	t _{OSS}	t _{SCY} /2 - 30	-	20	-	
SCLK上升→输出数据保持	t _{OHS}	t _{SCY} /2 - 30	-	20	-	
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	-	

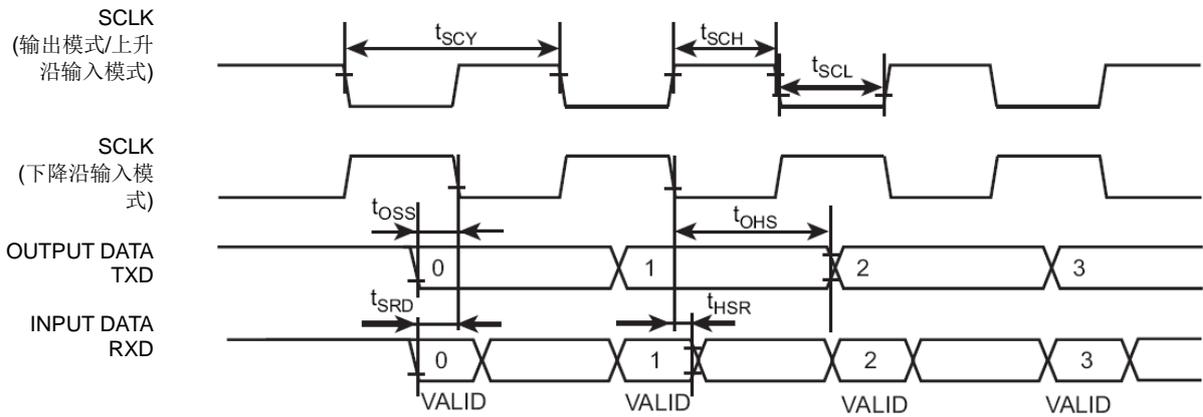


图 25-1 串行信道计时(串行输入输出(SIO))

25.6.3 串行总线接口(I2C/SIO)

25.6.3.1 I2C 模式

下表中，字母x代表I2C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。

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		单元
		最小	最大	最小	最大	最小	最大	
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)(注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
STOP条件和START条件之间的总线空闲时间。	t _{BUF}	(注 5)	-	4.7	-	1.3	-	μs

注 1:SCL时钟低幅宽(输出)=($2^{n-1}+58$)/x

注 2:SCL时钟高幅宽(输出)=($2^{n-1}+14$)/x

对于I2C-总线规格,标准模式的最高速度为 100 kHz,快速模式为 400 kHz。内部SCL频率设置应符合以上注 1与注 2的要求。

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,来让SDA信号桥接SCL下降沿的未定义区域。

不过,SBI仍然不符合该要求。此外,SCL的输出寄存器不含下降沿的斜率控制;因此,设备制造商所推出的设计应能满足该表所列输入数据保持时间的要求(包括SCL和SDA线路的tr/tf)。

注 5:取决于软件

注 6:飞利浦I2C-总线规范指示,若快速模式装置电源关闭,SDA 和SCL I/O引脚须为浮动,以免阻塞总线线路。不过,SBI仍然不符合该要求。

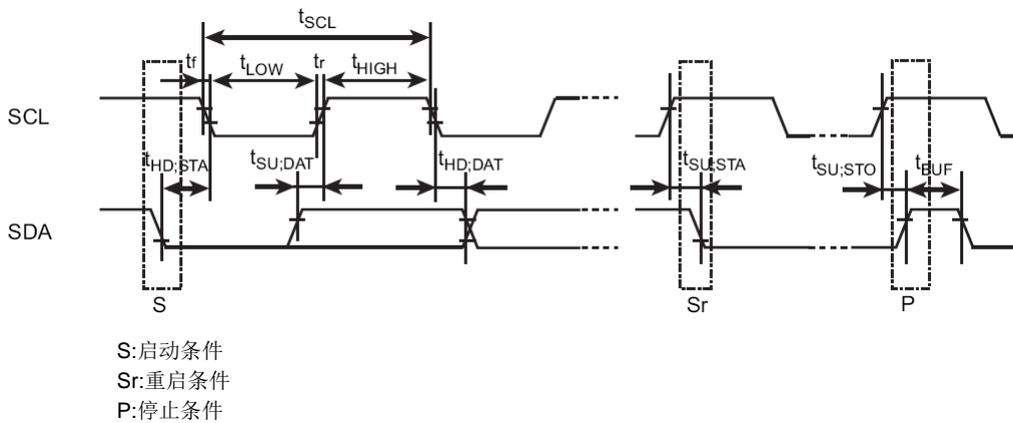


图 25-2 串行总线时序(I2C)

25.6.3.2 时钟同步 8-位 SIO 模式

在下表中，字母x表示I2C/SIO操作时钟周期时间，与f_{sys}周期时间一致。因时钟齿轮功能编程而异。

(1)SCK输入模式(占空度为 50%的SCK信号)

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
SCK时钟高宽度 (输入)	tSCH	4x	-	100	-	ns
SCK时钟低宽度 (输入)	tSCL	4x	-	100	-	
SCK 周期	tSCY	tSCH + tSCL	-	200	-	
输出数据 ←SCK上升	tOSS	tSCY/2 - 3x- 45	-	-20 (注)	-	
SCK上升 →输出数据保持	tOHS	tSCY/2 + x	-	125	-	
有效数据输入 ←SCK上升	tSRD	30 - x	-	5	-	
SCK上升 →输入数据保持	tHSR	30	-	30	-	

注:通过调整SCK周期来保持该值正值。

(2)SCK输出模式(占空度为 50%的SCK信号)

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
SCK 周期 (可编程)	tSCY	16x	-	400	-	ns
输出数据 ←SCK上升	tOSS	tSCY/2 - 30	-	170	-	
SCK上升 →输出数据保持	tOHS	tSCY/2 - 30	-	170	-	
有效数据输入 ←SCK上升	tSRD	45	-	45	-	
SCK上升 →输入数据保持	tOHS	0	-	0	-	

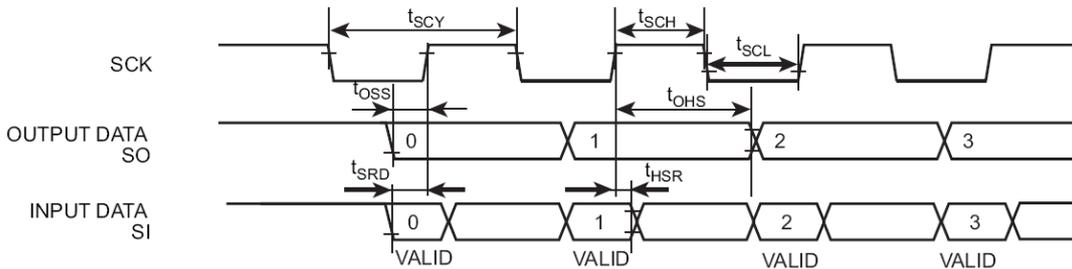


图 25-3 串行总线计时(串行输入输出)

25.6.4 事件计数器

下表中，字母x代表TMRB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。

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
时钟低脉宽	tVCKL	$2x + 100$	-	150	-	ns
时钟高脉宽	tVCKH	$2x + 100$	-	150	-	ns

25.6.5 捕捉

下表中，字母x代表TMRB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。

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
低脉冲宽度	tCPL	$2x + 100$	-	150	-	ns
高脉冲宽度	tCPH	$2x + 100$	-	150	-	ns

25.6.6 外部中断

下表中，字母x代表fsys周期时间。

1. 除STOP释放中断信号之外

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
INT6, 7与C的低脉宽	t _{INTAL}	x + 100	-	125	-	ns
INT6, 7与C的高脉宽	t _{INTAH}	x + 100	-	125	-	ns

2. STOP释放中断信号

参数	符号	等式		40 MHz		单元
		最小	最大	最小	最大	
INT6, 7与C的低脉宽	t _{INTBL}	100	-	100	-	ns
INT6, 7与C的高脉宽	t _{INTBH}	100	-	100	-	ns

25.6.7 调试通讯

25.6.7.1 AC 测量条件

- 输出电平:高 = $0.7 \times DVDD5$, 低 = $0.3 \times DVDD5$
- 占空电容:CL(TRACECLK) = 25 pF, CL(TRACEDATA) = 20 pF

25.6.7.2 SWD 接口

参数	符号	最小	最大	单元
CLK周期	T_{dck}	100	-	ns
DATA在CLK上升后保持	T_{d1}	4	-	
DATA在CLK上升后有效	T_{d2}	-	45	
DATA至CLK上升有效	T_{ds}	20	-	
DATA在CLK下降后保持	T_{dh}	15	-	

25.6.7.3 JTAG 接口

参数	符号	最小	最大	单元
CLK周期	T_{dck}	100	-	ns
DATA在CLK下降后保持	T_{d3}	4	-	
DATA在CLK下降后有效	T_{d4}	-	45	
DATA至CLK上升有效	T_{ds}	20	-	
DATA在CLK上升后保持	T_{dh}	15	-	

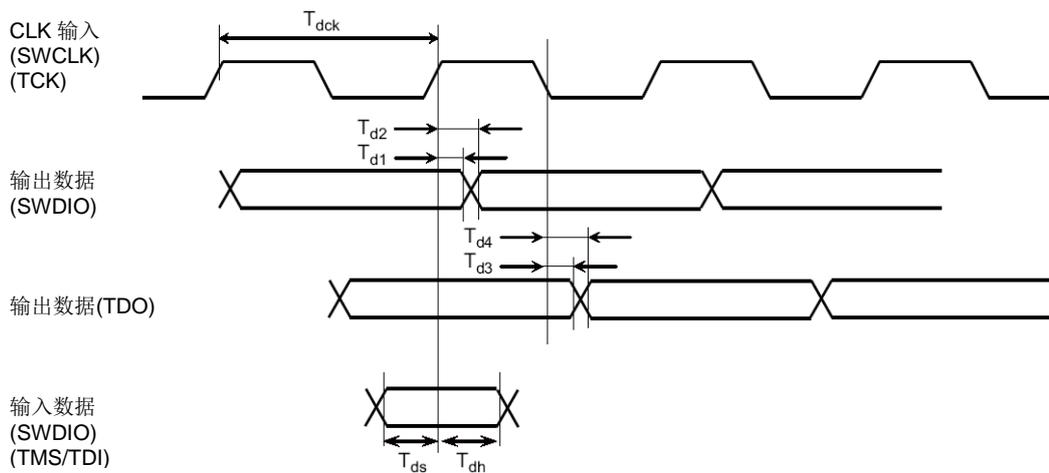


图 25-4 JTAG与SWD通讯计时

25.6.8 闪存特性

参数	额定值	最小	典型值	最大	单元
闪存存储器覆写保证	Ta = 0 to 70°C VDD5B=AVDD5B=4.5~5.0	-	-	100	次

25.6.9 芯片上振荡器

参数	符号	额定值	最小	典型值	最大	单元
振荡频率	fosc2	Ta=-40~105°C	9.4	9.7	10	MHz

注:工厂缺省值

25.6.10 外部振荡器举例

参数	符号	额定值	PLL	最小	典型值	最大	单元
高频振荡	fosc1	Ta=-40~105°C	5	7.92	8	8.08	MHz
			4	9.9	10	10.1	

25.7 振荡电路

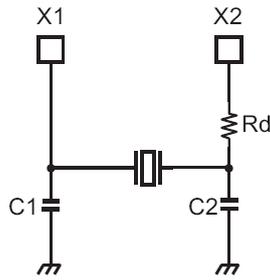


图 25-5 高频振荡连接

注 1:振荡器的负荷值为负荷(C1与C2)与实际组合版浮动负荷之和。在使用下表所列的各C1和C2值时,很可能出现操作错误。在进行板设计时,应设计振荡器的最小长度模式。此外,本公司建议用实际的板件进行振荡器评价。

注 2:不要用外部驱动器驱动X1/X2。

TX03已经以下振荡器供应商评估。应按本信息的要求选择外部件。

25.7.1 建议采用的陶瓷振荡器

Murata Manufacturing Co., Ltd(村田制造有限公司)推荐针对TX03使用高频率振荡器。

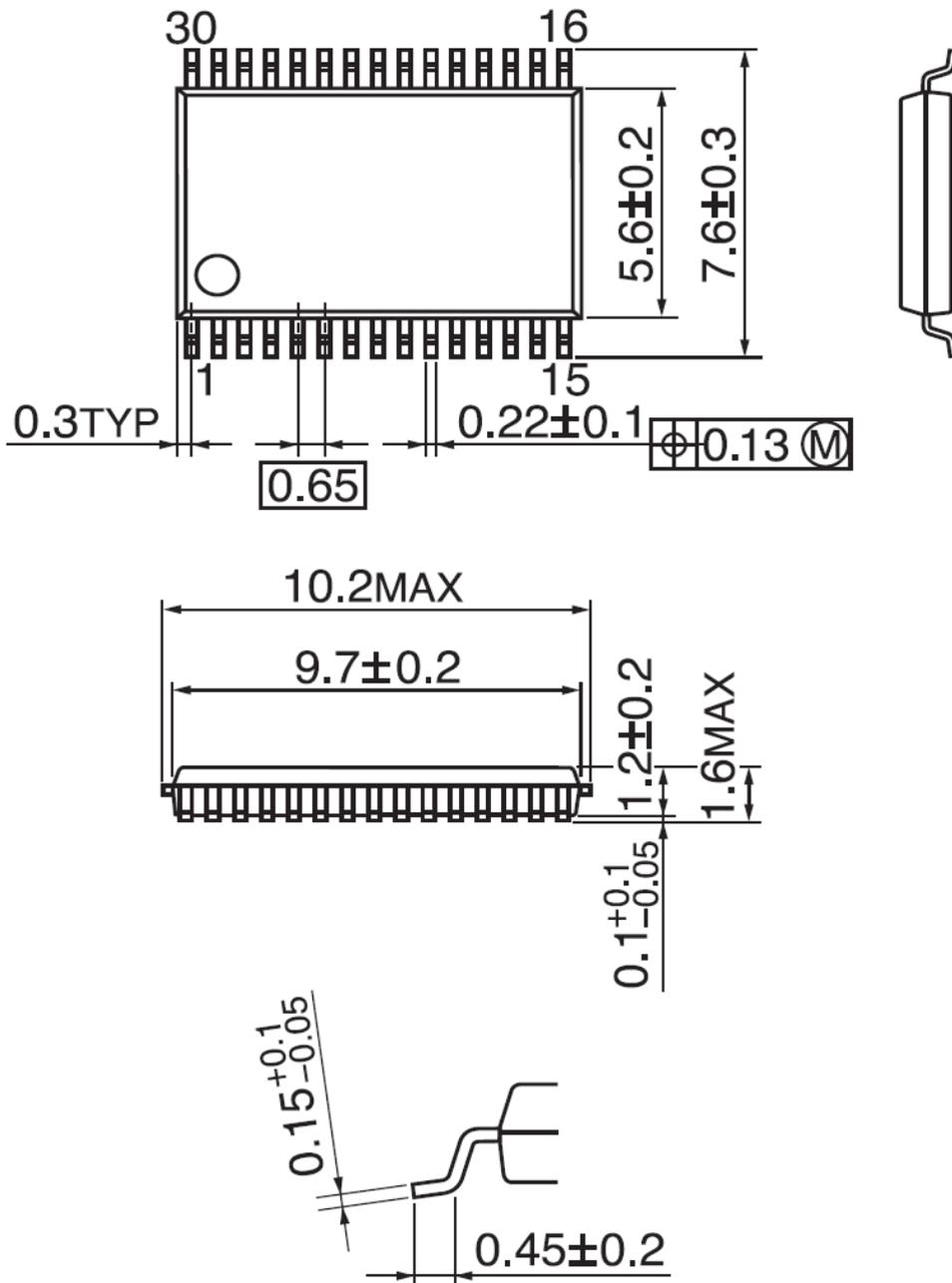
详情请参阅以下URL

<http://www.murata.co.jp>

26. 封装尺寸

类型SSOP30-P-300-0.65

单元:mm





RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

译文