

译文

TMPM475FDFG/FZFG/FYFG

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。

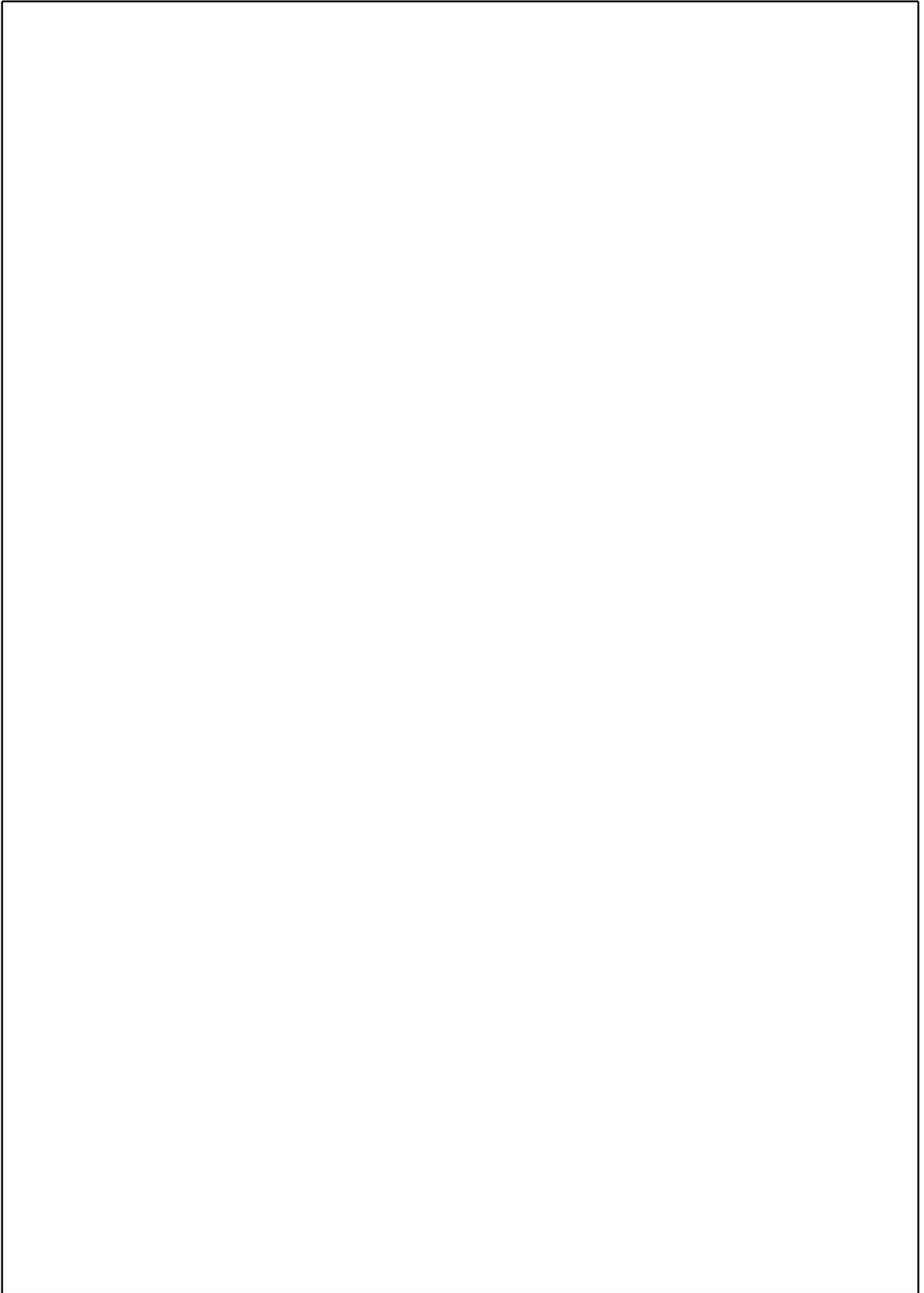
原本: “TMPM475FDFG/FZFG/FYFG” 2016-03-10

翻译日: 2016-09-08

TOSHIBA

32 位 RISC 微控制器
TX04 系列

TMPM475FDFG/FZFG/FYFG





ARM, Cortex and Thumb are registered trademarks of ARM Limited (or its subsidiaries) in the EU
and/or elsewhere. All rights reserved.



Toshiba MCU关于使用的注意事项概要

此页说明Toshiba MCU 关于使用的注意事项概要。
注意如果注意事项概要和文档本体描述之间有差异, 文档本体描述的优先级更高。

1. MCU 上电时工作

上电时, MCU内部状态不稳定。因此, 引脚的状态不确定直到复位工作完成。

当由一个外部复位引脚执行复位, MCU的使用复位引脚的引脚不确定直到复位工作由外部引脚完成。

另外,当复位由内部上电复位执行,MCU使用内部上电复位的引脚不确定直到电源电压达到上电复位有效的电压。

2. 未使用引脚

MCU未使用的输入/输出端口被禁止使用。引脚为高-阻抗。

通常,如果MCUs 在高-阻抗引脚保持开路时工作, 电气损坏或锁存会在内部 LSI发生由于受外部噪声影响的诱导电压。

Toshiba 推荐各未使用引脚应该通过电阻连接到电源引脚或GND引脚。

3. 时钟振荡稳定性

在时钟振荡变稳定后复位状态必须被释放。当程序处理中如果时钟变为另外时钟, 等到时钟稳定。

介绍: 关于在此规格下SFR (特殊 功能 寄存器) 描述的注释

SFR (特殊 功能 寄存器) 为外围电路 (IP)控制寄存器。

IPs 的 SFR 地址在关于内存映射章节, SFR 详细在各 IP 章节描述。

此规格中使用的 SFR 定义与下面规则一致。

a. 各 IP 的 SFR 表之一例

- IP 各章节 SFR 表提供寄存器名, 地址和简单描述。
- 所有寄存器有 32-位 唯一 地址且寄存器地址如下定义, 有一些异常: "基地址 + (唯一) 地址"

基 地址= 0x0000_0000

寄存器 名		地址(基+)
控制 寄存器	SAMCR	0x0004
		0x000C

注: **SAMCR** 寄存器地址为32 位宽从地址 **0x0000_0004** (基 地址(0x00000000) + 唯一 地址 (0x0004))。

注: 上述寄存器是为了解释目的一例且不是为了演示目的。
此寄存器不存在于此微控制器。

b. SFR(寄存器)

- 各寄存器基本由 32-位 寄存器组成(一些 异常)。
- 各寄存器描述提供位, 位符, 类型, 复位后初始值和功能。

1.2.2 SAMCR(控制 寄存器)

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

位	位 符	类型	功能
31-10	-	R	"0" 为可读。
9-7	MODE [2:0]	R/W	工作模式设置 000: 样本模式 0 001: 样本模式 1 010: 样本模式 2 011: 样本模式 3 上述设置以外: 保留
6-0	TDATA [6:0]	W	传输数据

注: 类型分为以下三种。

R / W	读 写
R	读
W	写

c. 数据描述

SFR 中使用的符号方法如下描述。

- x: 通道 数字/端口
- n,m: 位 数字

d. 寄存器 描述

寄存器如下描述。

- 寄存器 名 <位 符>

例: SAMCR<MODE >="000" 或 SAMCR<MODE [2:0]>="000"

<MODE [2:0]> 指位符模式中位 2 到位 0 (3 位 宽)。

- 寄存器 名 <位 >

例: SAMCR[9:7]="000"

指寄存器 SAMCR 位 9 到位 7 (32 位 宽)。

译文

修改 历史

日期	修改	备注
2015/1/9	暂定 1	首版
2015/5/20	暂定 2	内容 修改
-	-	-
2015/9/16	1	首版
2015/12/24	2.1	内容 修改
2016/2/18	2.2	内容 修改
2016/3/10	2.3	内容 修改

内容表

关于Toshiba MCU使用注意事项概要

TMPM475FDFG/FZFG/FYFG

1.1 特征	1
1.2 方块图	5
1.3 引脚布局 (顶视)	6
1.4 引脚名和功能	7
1.4.1 引脚名和功能各外设功能, 控制引脚和电源引脚.....	7
1.4.1.1 外设功能	
1.4.1.2 调试功能	
1.4.1.3 控制功能	
1.4.1.4 电源引脚	
1.4.1.5 电源引脚间电容	
1.4.2 TMPM475FDFG/FZFG/FYFG 引脚名和功能.....	11
1.4.2.1 引脚名和功能 表详细	
1.4.2.2 端口 / 调试 引脚	
1.4.2.3 控制 引脚	
1.4.2.4 电源引脚	

2. 产品信息

2.1 各外设功能信息	18
2.1.1 DMA 控制器 (DMAC).....	18
2.1.2 16-位 定时器 / 事件 计数器 (TMRB).....	18
2.1.3 串行 通道 (SIO/UART).....	19
2.1.4 串行 总线 接口 (I2C).....	20
2.1.5 CAN 控制器 (CAN).....	20
2.1.6 向量 引擎 (A-VE).....	21
2.1.7 电机 控制 电路 (PMD : 可编程 电机 驱动器).....	21
2.1.8 编码器 输入 电路 (A-ENC).....	22
2.1.9 模拟/数字 转换器 (ADC).....	22
2.1.10 看门狗 定时器(WDT).....	23
2.1.11 调试 接口.....	23

3. 处理器核

3.1 关于处理器核信息	25
3.2 可配置选项	25
3.3 异常/ 中断	26
3.3.1 中断输入数.....	26
3.3.2 优先级中断位数.....	26
3.3.3 SysTick.....	26
3.3.4 SYSRESETREQ.....	26
3.3.5 LOCKUP.....	26
3.3.6 备用 错误 状态 寄存器.....	26
3.4 事件	27
3.5 电源 管理	27

3.6 专用进入	27
3.7 浮点单元 (FPU)	27
<hr/>	
4. 内存映射	
<hr/>	
4.1 内存映射	29
4.2 总线矩阵	33
4.2.1 架构	34
4.2.1.1 单芯片模式	
4.2.1.2 单启动模式	
4.2.2 连接表	36
4.2.2.1 代码区域 / SRAM 区域	
4.2.2.2 外设区域 / 外部总线区域	
4.2.3 外设功能地址表	38
<hr/>	
5. 复位工作	
<hr/>	
5.1 冷复位	42
5.1.1 通过上电-复位电路复位(不使用 RESET 引脚)	42
5.1.2 通过低-压-检测电路复位(不使用 RESET 引脚)	43
5.1.3 通过 RESET 引脚复位(POR 复位工作有效)	44
5.1.4 通过 RESET 引脚复位(VLTD 复位工作有效)	45
5.2 预热	46
5.2.1 复位期间	46
5.3 复位后	46
<hr/>	
6. 时钟 / 模式控制	
<hr/>	
6.1 特征	47
6.2 寄存器	48
6.2.1 寄存器表	48
6.2.2 CGSYSCR (系统控制寄存器)	49
6.2.3 CGOSCCR (振荡控制寄存器)	50
6.2.4 CGSTBYCR (待机控制寄存器)	52
6.2.5 CGPLLSEL (PLL 选择寄存器)	53
6.2.6 CGCKSTP (外设时钟停止寄存器)	54
6.3 时钟控制	55
6.3.1 时钟类型	55
6.3.2 复位后初始值	55
6.3.3 时钟系统图	56
6.3.4 时钟乘法电路 (PLL)	57
6.3.4.1 稳定时间	
6.3.4.2 PLL 设置顺序	
6.3.5 预热功能	59
6.3.6 系统时钟	61
6.3.7 预分频时钟控制	62
6.4 模式和模式过渡	63
6.4.1 模式过渡	63
6.5 工作模式	63
6.5.1 正常模式	63
6.6 低功耗模式	64
6.6.1 IDLE 模式	64
6.6.2 STOP 模式	65
6.6.3 低功耗模式设置	66
6.6.4 各模式下工作状态	66
6.6.5 释放低功耗模式	67
6.6.6 预热	69

6.6.7 模式切换时的时钟操作	70
6.6.7.1 工作模式的切换 : NORMAL → STOP → NORMAL	

7. 内部高-速振荡调节 功能

7.1 架构	71
7.2 寄存器	72
7.2.1 寄存器 表.....	72
7.2.2 TRMOSCPRO (保护 寄存器).....	72
7.2.3 TRMOSCEN (使能 寄存器).....	73
7.2.4 TRMOSCINIT (初始微调值 监视器 寄存器).....	74
7.2.5 TRMOSCSET (微调值 设置 寄存器).....	75
7.3 工作 描述	76
7.3.1 概要.....	76
7.3.2 调节 范围.....	76

8. 异常

8.1 总览	77
8.1.1 异常 类型.....	77
8.1.2 处理流程图.....	78
8.1.2.1 异常 请求和检测	
8.1.2.2 异常 处理和分支到中断服务程序 (占先)	
8.1.2.3 执行 ISR	
8.1.2.4 异常 退出	
8.2 复位 异常	84
8.3 非-可屏蔽 中断 (NMI)	85
8.4 SysTick	85
8.5 中断	86
8.5.1 中断 源	86
8.5.1.1 中断 路径	
8.5.1.2 生成 中断 请求	
8.5.1.3 传输	
8.5.1.4 使用外部中断引脚 时注意事项	
8.5.1.5 中断源表	
8.5.1.6 激活 电平	
8.5.2 中断 处理.....	91
8.5.2.1 流程图	
8.5.2.2 准备	
8.5.2.3 通过 时钟 生成器检测	
8.5.2.4 通过 CPU 检测	
8.5.2.5 CPU 处理	
8.5.2.6 中断 服务 程序 (ISR)	
8.6 异常/中断-相关 寄存器	96
8.6.1 寄存器 表.....	96
8.6.2 NVIC 寄存器.....	97
8.6.2.1 SysTick 控制和状态 寄存器	
8.6.2.2 SysTick Reload 值 寄存器	
8.6.2.3 SysTick Current 值 寄存器	
8.6.2.4 SysTick 刻度值 寄存器	
8.6.2.5 中断 控制 寄存器	
8.6.2.6 中断 优先级 寄存器	
8.6.2.7 向量表 偏移 寄存器	
8.6.2.8 应用中断和复位 控制 寄存器	
8.6.2.9 系统 处理器 优先级 寄存器	
8.6.2.10 系统 处理器 控制和状态 寄存器	
8.6.3 时钟 生成器 寄存器.....	115
8.6.3.1 CG 中断 模式 控制 寄存器	
8.6.3.2 CGICRCG(CG 中断 请求 清除 寄存器)	
8.6.3.3 CGRSTFLG (复位 标志 寄存器)	

9. 数字噪声过滤电路 (DNF)

9.1 配置	123
9.2 寄存器	124
9.2.1 寄存器表	124
9.2.1.1 NFCKCR (噪声过滤控制寄存器)	
9.2.1.2 NFENCR (噪声过滤使能寄存器)	
9.3 工作描述	128
9.3.1 配置	128
9.3.2 工作	128
9.3.3 噪声过滤可用工作模式	128
9.3.4 STOP 模式使用注意事项	128
9.3.5 最小噪声过滤时间	128

10. μ DMA 控制器 (μ DMAC)

10.1 总览	129
10.1.1 功能表	129
10.2 方块图	130
10.3 寄存器	131
10.3.1 寄存器表	131
10.3.2 DMAxStatus (DMAC 状态寄存器)	132
10.3.3 DMAxCfg (DMAC 配置寄存器)	133
10.3.4 DMAxCtrlBasePtr (通道控制数据基-指针寄存器)	134
10.3.5 DMAxAltCtrlBasePtr (通道交替控制数据基-指针寄存器)	134
10.3.6 DMAxChnlSwRequest (通道软件请求寄存器)	135
10.3.7 DMAxChnlUseburstSet (通道 useburst 设置寄存器)	136
10.3.8 DMAxChnlUseburstClr (通道 useburst 清除寄存器)	137
10.3.9 DMAxChnlReqMaskSet (通道请求屏蔽设置寄存器)	138
10.3.10 DMAxChnlReqMaskClr (通道请求屏蔽清除寄存器)	139
10.3.11 DMAxChnlEnableSet (通道使能设置寄存器)	140
10.3.12 DMAxChnlEnableClr (通道使能清除寄存器)	141
10.3.13 DMAxChnlPriAltSet (通道主要-交替设置寄存器)	142
10.3.14 DMAxChnlPriAltClr (通道主要-交替清除寄存器)	143
10.3.15 DMAxChnlPrioritySet (通道优先级设置寄存器)	144
10.3.16 DMAxChnlPriorityClr (通道优先级清除寄存器)	145
10.3.17 DMAxErrClr (总线错误清除寄存器)	146
10.3.18 DMAIFFLGx (DMA 标志寄存器)	147
10.4 工作	148
10.4.1 通道控制数据内存映射	148
10.4.2 通道控制数据架构	149
10.4.2.1 传输源数据最终地址	
10.4.2.2 传输目的地地址最终地址	
10.4.2.3 控制数据设置	
10.4.3 工作模式	151
10.4.3.1 无效设置	
10.4.3.2 基础模式	
10.4.3.3 自动请求模式	
10.4.3.4 Ping-pong 模式	
10.4.3.5 内存分散/聚集模式	
10.4.3.6 外设分散/聚集模式	
10.5 注意事项	158
10.5.1 使用 SIO/UART, TMRB, ADC	158

11. 输入 / 输出 端口

11.1 寄存器	159
11.1.1 寄存器表	160

11.1.2 端口功能和设置表	161
11.1.2.1 端口 A	
11.1.2.2 端口 B	
11.1.2.3 端口 C	
11.1.2.4 端口 D	
11.1.2.5 端口 E	
11.1.2.6 端口 F	
11.1.2.7 端口 G	
11.1.2.8 端口 H	
11.1.2.9 端口 J	
11.1.2.10 端口 K	
11.1.2.11 端口 L	
11.1.2.12 端口 N	
11.1.2.13 端口 P	
11.2 端口方块图	175
11.2.1 类型 FT1	175
11.2.2 类型 FT2	176
11.2.3 类型 FT3	177
11.2.4 类型 FT4	178
11.2.5 类型 FT5	179
11.2.6 类型 FT6	180
11.2.7 类型 FT7	181
11.2.8 类型 FT8	182
<hr/>	
12. 16-位定时器 / 事件计数器 (TMRB)	183
12.1 概要	183
12.2 配置	184
12.3 寄存器	185
12.3.1 根据通道寄存器表	185
12.3.2 TBxEN(使能寄存器)	186
12.3.3 TBxRUN(RUN寄存器)	187
12.3.4 TBxCR(控制寄存器)	188
12.3.5 TBxMOD(模式寄存器)	189
12.3.6 TBxFFCR(Flip-flop控制寄存器)	190
12.3.7 TBxST(状态寄存器)	191
12.3.8 TBxIM(中断屏蔽寄存器)	192
12.3.9 TBxUC(递增计数器捕捉寄存器)	193
12.3.10 TBxRG0(定时器寄存器 0)	194
12.3.11 TBxRG1(定时器寄存器 1)	194
12.3.12 TBxCP0(捕捉寄存器 0)	195
12.3.13 TBxCP1(捕捉寄存器 1)	195
12.4 各电路工作描述	196
12.4.1 预分频	196
12.4.2 递增-计数器 (UC)	196
12.4.2.1 源时钟	
12.4.2.2 计数器开始 / 停止	
12.4.2.3 计数器清除	
12.4.2.4 递增-计数器溢出	
12.4.3 定时器寄存器 (TBxRG0, TBxRG1)	197
12.4.4 捕捉	197
12.4.5 捕捉寄存器 (TBxCP0, TBxCP1)	197
12.4.6 递增计数器捕捉寄存器 (TBxUC)	198
12.4.7 比较器 (CP0, CP1)	198
12.4.8 定时器 Flip-flop (TBxFF0)	198
12.4.9 捕捉中断 (INTCAPx0, INTCAPx1)	198
12.5 各模式工作描述	199
12.5.1 16-位间隔定时器模式	199
12.5.2 16-位事件计数器模式	199
12.5.3 16-位 PPG (可编程脉冲生成) 输出模式	200
12.5.4 外部触发器可编程脉冲生成输出模式 (PPG)	202
12.6 使用捕捉功能的应用	203
12.6.1 通过外部脉冲单-发脉冲输出触发器	203
12.6.2 脉冲宽测量	205

13. 串行通道带4字节FIFO (SIO/UART)

13.1 总览	207
13.2 配置	208
13.3 寄存器描述	209
13.3.1 寄存器表.....	209
13.3.2 SCxEN (使能寄存器).....	210
13.3.3 SCxBUF (缓冲寄存器).....	211
13.3.4 SCxCR (控制寄存器).....	212
13.3.5 SCxMOD0 (模式控制寄存器0).....	214
13.3.6 SCxMOD1 (模式控制寄存器1).....	215
13.3.7 SCxMOD2 (模式控制寄存器2).....	216
13.3.8 SCxBRCR (波特率生成器控制寄存器).....	218
13.3.9 SCxBRADD (波特率生成器控制寄存器2).....	219
13.3.10 SCxFCNF (FIFO配置寄存器).....	220
13.3.11 SCxRFC (接收FIFO配置寄存器).....	222
13.3.12 SCxTFC (传输FIFO配置寄存器).....	223
13.3.13 SCxRST (接收FIFO状态寄存器).....	224
13.3.14 SCxTST (传输FIFO状态寄存器).....	225
13.4 各模式下工作	226
13.5 数据格式	227
13.5.1 数据格式表.....	227
13.5.2 奇偶控制.....	228
13.5.2.1 传输	
13.5.2.2 接收	
13.5.3 STOP位长.....	228
13.6 时钟控制	229
13.6.1 预分频.....	229
13.6.2 串行时钟生成电路.....	229
13.6.2.1 波特率生成器	
13.6.2.2 时钟选择电路	
13.7 传输/接收缓冲和FIFO	233
13.7.1 配置.....	233
13.7.2 传输/接收缓冲.....	233
13.7.3 初始化传输缓冲.....	234
13.7.4 FIFO.....	234
13.8 状态标志	235
13.9 错误标志	235
13.9.1 OERR标志.....	235
13.9.2 PERR标志.....	236
13.9.3 FERR标志.....	236
13.10接收	237
13.10.1 接收计数器.....	237
13.10.2 接收控制单元.....	237
13.10.2.1 I/O接口模式	
13.10.2.2 UART模式	
13.10.3 接收工作.....	237
13.10.3.1 接收缓冲	
13.10.3.2 接收FIFO工作	
13.10.3.3 I/O接口模式带时钟输出模式	
13.10.3.4 读接收数据	
13.10.3.5 唤醒功能	
13.10.3.6 过运行错误	
13.11传输	241
13.11.1 传输计数器.....	241
13.11.2 传输控制.....	241
13.11.2.1 I/O接口模式下	
13.11.2.2 UART模式下	
13.11.3 传输工作.....	242
13.11.3.1 传输缓冲工作	
13.11.3.2 传输FIFO工作	
13.11.3.3 I/O接口模式带时钟输出模式下传输	

13.11.3.4	I/O 接口模式下最后一位输出后 SCxTXD 引脚电平	
13.11.3.5	欠-运行 错误	
13.11.3.6	带时钟输入模式的 I/O 接口模式下数据保持时间	
13.12	握手 功能	246
13.13	中断/错误 生成 时序	247
13.13.1	接收 中断	247
13.13.1.1	单 缓冲 / 双 缓冲	
13.13.1.2	FIFO	
13.13.2	传输 中断	248
13.13.2.1	单 缓冲 / 双 缓冲	
13.13.2.2	FIFO	
13.13.3	错误 生成	249
13.13.3.1	UART 模式	
13.13.3.2	I/O 接口 模式	
13.14	DMA 请求	250
13.15	软件 复位	251
13.16	各模式下工作	252
13.16.1	模式 0 (I/O 接口 模式)	252
13.16.1.1	传输	
13.16.1.2	接收	
13.16.1.3	传输和接收 (全-双工)	
13.16.2	模式 1 (7-位 UART 模式)	263
13.16.3	模式 2 (8-位 UART 模式)	263
13.16.4	模式 3 (9-位 UART 模式)	264
13.16.4.1	唤醒 功能	
13.16.4.2	协议	
<hr/>		
14.	串行 总线 接口 (I2C/SIO)	
14.1	配置	267
14.2	寄存器	268
14.2.1	各通道寄存器	268
14.3	I2C 总线 模式	269
14.3.1	控制 寄存器 I2C 总线 模式	269
14.3.1.1	SBIxCR0(控制 寄存器 0)	
14.3.1.2	SBIxCR1(控制 寄存器 1)	
14.3.1.3	SBIxCR2(控制 寄存器 2)	
14.3.1.4	SBIxSR (状态 寄存器)	
14.3.1.5	SBIxBR0(串行 总线 接口 波特率 寄存器 0)	
14.3.1.6	SBIxDBR (串行 总线 接口 数据 缓冲 寄存器)	
14.3.1.7	SBIxI2CAR (I2C 总线 地址 寄存器)	
14.3.2	控制	277
14.3.2.1	工作 模式	
14.3.2.2	串行 时钟	
14.3.2.3	设置 获取 模式	
14.3.2.4	设置 每传输位数	
14.3.2.5	从 地址和 地址 识别 模式	
14.3.2.6	配置 SBI 为 主机或从	
14.3.2.7	配置 SBI 为 传输器或接收器	
14.3.2.8	总线 忙 监视器	
14.3.2.9	中断 服务 请求和释放	
14.3.2.10	仲裁丢失检测 监视器	
14.3.2.11	从 地址 匹配 检测 监视器	
14.3.2.12	总-呼叫 检测 监视器	
14.3.2.13	最后 接收位 监视器	
14.3.2.14	数据 缓冲 寄存器 (SBIxDBR)	
14.3.2.15	波特率 寄存器 (SBIxBR0)	
14.3.2.16	软件 复位	
14.3.3	数据 传输 处理	283
14.3.3.1	设备 初始化	
14.3.3.2	生成开始 条件和从 地址	
14.3.3.3	传输数据 字	
14.3.3.4	生成停止 条件	
14.3.3.5	重新开始 处理	
14.3.4	数据 格式	292
14.3.5	关于多主机使用注意事项	292
14.4	SIO 模式	293

14.4.1 SIO 模式控制 寄存器	293
14.4.1.1 SBIxCR0(控制 寄存器 0)	
14.4.1.2 SBIxCRI(控制 寄存器 1)	
14.4.1.3 SBIxDBR (数据 缓冲 寄存器)	
14.4.1.4 SBIxCR2(控制 寄存器 2)	
14.4.1.5 SBIxSR (状态 寄存器)	
14.4.1.6 SBIxBR0 (波特率寄存器 0)	
14.4.2 控制	299
14.4.2.1 串行 时钟	
14.4.2.2 传输 模式	

15. CAN 控制器 (CAN)

15.1 总览	307
15.2 方块图	308
15.3 CAN 接口	308
15.4 寄存器	309
15.4.1 寄存器 表	309
15.4.2 CANMBxID (信息 ID 域 寄存器)	310
15.4.3 CANMBxTSVMCF (时间 标志值 / 信息 控制 域 寄存器)	311
15.4.4 CANMBxDL/CANMBxDH (数据 域 寄存器)	312
15.4.5 CANMC (邮箱 配置 寄存器)	314
15.4.6 CANMD (邮箱 方向 寄存器)	315
15.4.7 CANTRS (传输 请求 设置 寄存器)	316
15.4.8 CANTRR (传输 请求 复位 寄存器)	317
15.4.9 CANTA (传输 获取 寄存器)	318
15.4.10 CANAA (中止 获取 寄存器)	319
15.4.11 CANRMP (接收 信息 挂起 寄存器)	320
15.4.12 CANRML (接收 信息 丢失 寄存器)	321
15.4.13 CANLAM (当地 接受 屏蔽 寄存器)	322
15.4.14 CANGAM (全球 接受 屏蔽 寄存器)	323
15.4.15 CANMCR (主机 控制 寄存器)	324
15.4.16 CANGSR (全球 状态 寄存器)	326
15.4.17 CANBCR1 (位 配置 寄存器 1)	328
15.4.18 CANBCR2 (位 配置 寄存器 2)	329
15.4.19 CANGIF (全球 中断 标志 寄存器)	330
15.4.20 CANGIM (全球 中断 屏蔽 寄存器)	331
15.4.21 CANMBTIF (邮箱 传输 中断 标志 寄存器)	332
15.4.22 CANMBRIF (邮箱 接收 中断 标志 寄存器)	333
15.4.23 CANMBIM (邮箱 中断 屏蔽 寄存器)	334
15.4.24 CANCDR (改变 数据 请求 寄存器)	335
15.4.25 CANRFP (遥控 帧 挂起 寄存器)	336
15.4.26 CANCEC (错误 计数器 寄存器)	337
15.4.27 CANTSP (时间 标志 计数器 预分频 寄存器)	338
15.4.28 CANTSC (时间 标志 计数器 寄存器)	339
15.5 各电路工作说明	340
15.5.1 邮箱	340
15.5.2 传输 控制 寄存器	341
15.5.3 接收 控制 寄存器	342
15.5.4 遥控 帧 控制 寄存器	343
15.5.5 接收 过滤	344
15.5.6 时间 标志 功能	345
15.5.7 中断 控制	346
15.6 工作模式	348
15.6.1 配置 模式	348
15.6.2 睡眠 模式	350
15.6.3 挂起 模式	350
15.6.4 测试 循环 后 模式	351
15.6.5 测试 错误 模式	351
15.7 工作描述	352
15.7.1 接收 信息	352
15.7.2 传输 信息	353
15.7.3 遥控 帧 处理	354

15.8 位配置	355
16. 12-位 模拟-到-数字 转换器	
16.1 功能和特征	357
16.2 方块图	358
16.3 寄存器表	359
16.4 寄存器 描述	361
16.4.1 ADxCCLK (时钟 设置 寄存器)	361
16.4.2 ADxMOD0 (模式 设置 寄存器 0)	363
16.4.3 ADxMOD1 (模式 设置 寄存器 1)	364
16.4.4 ADxMOD2 (模式 设置 寄存器 2)	365
16.4.5 ADxMOD3 (模式 设置 寄存器 3)	366
16.4.6 ADxMOD4 (模式 设置 寄存器 4)	367
16.4.7 ADxMOD5 (模式 设置 寄存器 5)	368
16.4.8 ADxCMPCR0(监视器 设置 寄存器 0)	369
16.4.9 ADxCMPCR1(监视器 设置 寄存器 1)	370
16.4.10 ADxCMP0(转换 结果 比较 寄存器 0)	371
16.4.11 ADxCMP1(转换 结果 比较 寄存器 1)	371
16.4.12 ADxREG0(转换 结果 寄存器 0)	372
16.4.13 ADxREG1(转换 结果 寄存器 1)	373
16.4.14 ADxREG2(转换 结果 寄存器 2)	374
16.4.15 ADxREG3(转换 结果 寄存器 3)	375
16.4.16 ADxREG4(转换 结果 寄存器 4)	376
16.4.17 ADxREG5(转换 结果 寄存器 5)	377
16.4.18 ADxREG6(转换 结果 寄存器 6)	378
16.4.19 ADxREG7(转换 结果 寄存器 7)	379
16.4.20 ADxREG8(转换 结果 寄存器 8)	380
16.4.21 ADxREG9(转换 结果 寄存器 9)	381
16.4.22 ADxREG10(转换 结果 寄存器 10)	382
16.4.23 ADxREG11(转换 结果 寄存器 11)	383
16.4.24 PMD 触发器 程序 寄存器	384
16.4.24.1 ADxPSEL0 到 ADxPSEL11(PMD 触发器 程序 数 选择 寄存器 0 到 11)	
16.4.24.2 ADxPINTS0 到 5(PMD 触发器 中断 选择 寄存器 0 到 5)	
16.4.24.3 ADxPSET0 到 5(PMD 触发器 程序 寄存器 0 到 5)	
16.4.25 ADxTSET03 / ADxTSET47 / ADxTSET811 (定时器 触发器 程序 寄存器)	402
16.4.26 ADxSSET03 / ADxSSET47 / ADxSSET811 (软件 触发器 程序 寄存器)	406
16.4.27 ADxASET03 / ADxASET47 / ADxASET811 (恒定 转换 程序 寄存器)	410
16.5 工作 描述	414
16.5.1 模拟 参考电压	414
16.5.2 开始 AD 转换	414
16.5.3 AD 转换 监视器 功能	415
16.6 AD 转换时序图	416
16.6.1 软件 触发器 转换	416
16.6.2 恒定 转换	418
16.6.3 通过触发器 AD 转换	419
16.7 使用 例	421
16.7.1 逐次 转换 使用 一个PMD (三电阻)和一个 ADC	421
16.7.2 模拟 转换 使用 一个PMD (三电阻)和两个ADC	422
16.7.3 模拟 转换 使用 PMD0 (三电阻), PMD1 (一个 电阻)和两个 ADC	423
16.7.4 逐次 转换 使用 一个PMD (一个 电阻)和一个ADC	425
16.8 AD 转换器使用注意事项	426
17. 电机 控制 电路 (PMD: 可编程 电机 驱动器)	
17.1 PMD 电路 配置	429
17.2 PMD 寄存器	430
17.2.1 PMDxDEN(PMD 使能 寄存器)	431
17.2.2 PMDxPORTMD(端口 输出 模式 寄存器)	432

17.2.3	PMDxMODESEL (模式选择寄存器)	433
17.2.4	脉宽调制电路	434
17.2.4.1	PMDxMDCR (PMD控制寄存器)	
17.2.4.2	PMDxCNTSTA (PWM计数器状态寄存器)	
17.2.4.3	PMDxMDCNT (PWM计数器寄存器)	
17.2.4.4	PMDxMDPRD (PWM周期寄存器)	
17.2.4.5	PMDxCMPU (PWM比较寄存器U相)	
17.2.4.6	PMDxCMPV (PWM比较寄存器V相)	
17.2.4.7	PMDxCMPW (PWM比较寄存器W相)	
17.2.5	传导控制电路	446
17.2.5.1	PMDxMDPOT (PMD输出设置寄存器)	
17.2.5.2	PMDxMDOUT (PMD传导控制寄存器)	
17.2.6	保护控制电路	451
17.2.6.1	EMG保护电路	
17.2.6.2	PMDxEMGREL (EMG释放寄存器)	
17.2.6.3	PMDxEMGCR (EMG控制寄存器)	
17.2.6.4	PMDxEMGSTA (EMG状态寄存器)	
17.2.6.5	OVV保护控制电路 (OVV输入方块)	
17.2.6.6	PMDxOVVCR (OVV控制寄存器)	
17.2.6.7	PMDxOVVSTA (OVV状态寄存器)	
17.2.7	死区时间控制电路	460
17.2.7.1	PMDxDTR (死区时间寄存器)	
17.2.8	同步触发器生成电路	463
17.2.8.1	PMDxTRGCMP0 (触发器比较寄存器0)	
17.2.8.2	PMDxTRGCMP1 (触发器比较寄存器1)	
17.2.8.3	PMDxTRGCMP2 (触发器比较寄存器2)	
17.2.8.4	PMDxTRGCMP3 (触发器比较寄存器3)	
17.2.8.5	PMDxTRGCR (触发器控制寄存器)	
17.2.8.6	PMDxTRGSYNCR (触发器更新时序设置寄存器)	
17.2.8.7	PMDxTRGMID (触发器输出模式设置寄存器)	
17.2.8.8	PMDxTRGSEL (触发器输出选择寄存器)	

18. 向量引擎 (A-VE)

18.1	概要	475
18.1.1	特征	475
18.1.2	主要功能	476
18.2	配置	476
18.2.1	向量引擎, 电机控制电路, 和AD转换器间的配合	478
18.3	寄存器	480
18.3.1	寄存器表	480
18.3.2	VE控制寄存器	483
18.3.2.1	VExEN (向量引擎工作使能/禁止寄存器)	
18.3.2.2	VExCPURUNTRG (CPU开始触发器选择寄存器)	
18.3.2.3	VExTASKAPP (任务选择寄存器)	
18.3.2.4	VExACTSCH (工作调度程序选择寄存器)	
18.3.2.5	VExREPTIME (工作调度程序重复数选择寄存器)	
18.3.2.6	VExTRGMODE (开始触发器模式设置寄存器)	
18.3.2.7	VExERRINTEN (错误中断使能/禁止设置寄存器)	
18.3.2.8	VExCOMPEND (VE强制中止寄存器)	
18.3.2.9	VExERRDET (错误检测寄存器)	
18.3.2.10	VExSCHTASKRUN (调度程序工作状态/处理中任务数寄存器)	
18.3.2.11	VExTMPREG0到5 (暂时寄存器)	
18.3.3	专用寄存器	496
18.3.3.1	VExMODE (任务控制模式寄存器)	
18.3.3.2	VExFMODE (流动控制寄存器)	
18.3.3.3	VExTPWM (PWM周期率设置寄存器)	
18.3.3.4	VExOMEGA (转速设置寄存器)	
18.3.3.5	VExTHETA (电机相设置寄存器)	
18.3.3.6	VExCOS/VExSIN/VExCOSM/VExSINM (SIN/COS寄存器)	
18.3.3.7	VExIDREF/VExIQREF (d-轴/q-轴电流参考设置寄存器)	
18.3.3.8	VExVD/VExVQ (d-轴/q-轴电压设置寄存器)	
18.3.3.9	VExCIDKI/VExCIDKP/VExVICQKI/VExCICQP (PI控制系数寄存器)	
18.3.3.10	VExVDIH/VExVDILH/VExVQIH/VExVQILH (PI控制集成组成储存寄存器)	
18.3.3.11	VExMCTLF (异常决定结果储存寄存器)	
18.3.3.12	VExFPWMCHG (PWM开关速设置寄存器)	
18.3.3.13	VExMDPRD (PWM周期设置寄存器)	
18.3.3.14	VExMINPLS (最小脉宽差分设置寄存器)	
18.3.3.15	VExSECTOR/VExSECTORM (扇形信息寄存器)	
18.3.3.16	VExIAO/VExIBO/VExICO (零电流寄存器)	
18.3.3.17	VExIAADC/VExIBADC/VExICADC (ADC电流结果寄存器)	

18.3.3.18	VExVDC/VExVDCL (电源电压 寄存器)	
18.3.3.19	VExID/VExIQ (d-轴/q-轴 电流 寄存器)	
18.3.3.20	VExTADC (ADC 转换时间 设置 寄存器)	
18.3.3.21	VExCMPU/ VExCMPV/ VExCMPW (PWM 占空比 寄存器)	
18.3.3.22	VExOUTCR (6-相 输出 控制 寄存器)	
18.3.3.23	VExTRGCRC (同步 触发器 纠正 设置 寄存器)	
18.3.3.24	VExTRGCMP0/VExTRGCMP1 (触发器 时序 设置 寄存器)	
18.3.3.25	VExTRGSEL (同步 触发器 选择 寄存器)	
18.3.3.26	VExEMGRS (EMG 保护 释放 寄存器)	
18.3.3.27	VExPIOLIM (PI 控制 输出 限制 寄存器)	
18.3.3.28	VExCIDKG/VExCIQKG (PI 控制的系数 范围 设置 寄存器 d-轴/q-轴)	
18.3.3.29	VExVSLIM (电压 标量 限制 寄存器)	
18.3.3.30	VExVDQ (电压 标量 寄存器)	
18.3.3.31	VExDELTA (倾角 寄存器)	
18.3.3.32	VExCPHI/VExCLD/VExCLQ/VExCR/VExCPHIG/VExCLG/VExCRG (电机 恒定 寄存器)	
18.3.3.33	VExVDE/VExVQE (非-干扰 控制 电压 寄存器 d-轴/q-轴)	
18.3.3.34	VExDTC (死区 时间 补偿 寄存器)	
18.3.3.35	VExHYS (电流识别迟滞 宽)	
18.3.3.36	VExDTCS (死区 时间 补偿 控制/状态 寄存器)	
18.3.3.37	VExPWMMAX/VExPWMMIN (PWM 输出 限制 寄存器)	
18.3.3.38	VExTHTCLP (裁剪 相值 设置 寄存器)	
18.4	工作描述	547
18.4.1	调度程序 管理	547
18.4.1.1	调度程序 控制	
18.4.1.2	开始 控制	
18.4.1.3	中断 控制	
18.4.2	任务描述	553
18.4.2.1	电流 控制 (任务 5)	
18.4.2.2	SIN/COS 计算 (任务 6)	
18.4.2.3	输出 电压 转换 (协调轴 转换/相转换)	
18.4.2.4	输出 控制	
18.4.2.5	触发器 生成 (任务 1)	
18.4.2.6	输入 处理	
18.4.2.7	输入 电流转换 (相 转换/协调轴 转换)	
18.4.2.8	别的任务	
18.5	VE 通道, PMD 和 ADC 间的组合	585
<hr/>		
19.	编码器 输入 电路 (A-ENC)	
19.1	概要	587
19.2	方块 图	587
19.3	寄存器	588
19.3.1	寄存器 表	588
19.3.2	ENxTNCR (ENC 控制 寄存器)	589
19.3.3	ENxRELOAD (RELOAD 比较 寄存器)	592
19.3.4	ENxINT (INT 比较 寄存器)	593
19.3.5	ENxCNT (计数器 寄存器)	594
19.3.6	ENxMCMP (MCMP 计数器 寄存器)	595
19.3.7	ENxRATE (相 计数 率 寄存器)	596
19.3.8	ENxSTS (状态 寄存器)	597
19.3.9	ENxINPCR (输入 处理 控制 寄存器)	598
19.3.10	ENxSMPDLY (采样 延迟 寄存器)	599
19.3.11	ENxINPMON (输入 监视器 寄存器)	600
19.3.12	ENxCLKCR (采样 时钟 控制 寄存器)	601
19.3.13	ENxINTCR (中断 控制 寄存器)	602
19.3.14	ENxINTF (中断 事件 标志 寄存器)	603
19.4	工作 描述	604
19.4.1	编码器 模式	604
19.4.2	传感器 模式	605
19.4.2.1	事件 计数	
19.4.2.2	定时器 计数	
19.4.2.3	相 计数	
19.4.3	定时器 模式	610
19.4.4	相 计数器 模式	611
19.4.4.1	相 差分 计数 模式	
19.5	各电路工作 描述	614
19.5.1	输入 电路	614

19.5.1.1	采样时钟	
19.5.1.2	采样模式	
19.5.1.3	噪声消除	
19.5.2	解码器	617
19.5.2.1	转缘检测和方向信号生成	
19.5.2.2	Z-输入检测电路	
19.5.2.3	跳跃检测和异常输入检测	
19.5.2.4	缘检测错误检测	
19.5.2.5	缓冲更新控制	
19.5.2.6	BEMF检测控制	
19.5.3	计数器	620
19.5.3.1	编码器模式和传感器模式(事件计数)	
19.5.3.2	传感器模式(定时器计数)或定时器模式	
19.5.3.3	传感器模式(相计数)或相计数器模式	
19.5.4	中断请求控制寄存器	623
19.6	无刷DC电机控制例	624
<hr/>		
20.	上电复位电路(POR)	
20.1	架构	625
20.2	功能	626
<hr/>		
21.	低压检测电路(VLTD)	
21.1	架构	627
21.2	寄存器	628
21.2.1	寄存器表	628
21.2.2	VDCR(电压检测控制寄存器)	628
21.3	工作描述	629
21.3.1	控制	629
21.3.2	功能	629
21.3.2.1	使能/禁止电压检测工作	
21.3.2.2	选择检测电压电平	
<hr/>		
22.	振荡频率检测器(OFD)	
22.1	方块图	631
22.2	寄存器	632
22.2.1	寄存器表	632
22.2.1.1	OFDCR1(振荡频率检测控制寄存器)	
22.2.1.2	OFDCR2(振荡频率检测控制寄存器2)	
22.2.1.3	OFDMNPLLOFF(检测频率最小值的设置寄存器(PLL OFF时))	
22.2.1.4	OFDMNPLLON(检测频率最小值的设置寄存器(PLL ON时))	
22.2.1.5	OFDMXPLOFF(检测频率最大值的设置寄存器(PLL OFF时))	
22.2.1.6	OFDMXPLLON(检测频率最大值的设置寄存器(PLL ON时))	
22.3	工作描述	639
22.3.1	设置	639
22.3.2	可用工作模式	639
22.3.3	工作	640
22.3.4	检测频率	641
22.3.5	工作处理例	642
<hr/>		
23.	看门狗定时器(WDT)	
23.1	配置	643
23.2	寄存器	644
23.2.1	寄存器表	644

23.2.2	WDMOD (看门狗 定时器 模式 寄存器).....	644
23.2.3	WDCR (看门狗 定时器 控制 寄存器).....	645
23.3	工作描述	646
23.3.1	基础 工作	646
23.3.2	工作 模式和状态	646
23.3.3	误操作 (过运行)被检测时的工作	646
23.3.3.1	INTWDT 中断 生成	
23.3.3.2	内部 复位 生成	
23.4	看门狗 定时器的控制	647
23.4.1	禁止 控制	647
23.4.2	使能 控制	647
23.4.3	看门狗 定时器清除控制.....	647
23.4.4	看门狗 定时器检测时间.....	647
24.	内存 (FLASH)	
24.1	特征	649
24.1.1	内存大小和配置.....	649
24.1.2	功能	651
24.1.3	工作 模式	651
24.1.3.1	模式 描述	
24.1.3.2	模式 决定	
24.1.4	内存 映射	654
24.1.5	保护/安全 功能	656
24.1.5.1	保护 功能	
24.1.5.2	安全 功能	
24.1.6	内存 交换 功能	657
24.1.6.1	概要	
24.1.6.2	工作 描述	
24.1.6.3	内存交换的使用	
24.1.7	寄存器	659
24.1.7.1	寄存器 表	
24.1.7.2	FCSECBIT (安全 位 寄存器)	
24.1.7.3	FCPSR0 (保护 状态 寄存器 0)	
24.1.7.4	FCSR (状态 寄存器)	
24.1.7.5	FCSWPSR (交换 状态 寄存器)	
24.1.7.6	FCAREASEL (区域 选择 寄存器 0)	
24.1.7.7	FCCR (控制 寄存器)	
24.1.7.8	FCSTSCLR (状态 清除 寄存器)	
24.1.7.9	FCWCLKCR (WCLK 设置 寄存器)	
24.1.7.10	FCPROGCR (程序计数 设置 寄存器)	
24.1.7.11	FCERASECR (擦除计数 设置 寄存器)	
24.2	闪存 详细	671
24.2.1	功能	671
24.2.2	内存的工作模式.....	671
24.2.3	如何执行 命令	671
24.2.4	自动 工作中止	672
24.2.5	自动 工作完成通知	673
24.2.5.1	处理	
24.2.6	命令 描述	673
24.2.6.1	自动 页 程序	
24.2.6.2	自动 芯片 擦除	
24.2.6.3	自动 区域 擦除	
24.2.6.4	自动 方块 擦除	
24.2.6.5	自动 页 擦除	
24.2.6.6	自动 保护 位 程序	
24.2.6.7	自动 保护 位 擦除	
24.2.6.8	ID-读	
24.2.6.9	读/复位 命令 (软件 复位)	
24.2.6.10	自动 内存 交换	
24.2.7	命令 顺序	678
24.2.7.1	命令 顺序 表	
24.2.7.2	地址 位配置 在总线周期	
24.2.7.3	区域 地址 (AA) 和 方块 地址 (BA)	
24.2.7.4	如何指定 保护 位 (PBA)	
24.2.7.5	ID-读 命令 (IA, ID)	
24.2.7.6	内存 交换 位 分配 (MSA)	
24.2.7.7	命令 顺序例	

24.2.8	流程图	685
24.2.8.1	自动程序	
24.2.8.2	自动擦除	
24.3	如何在单启动模式下重编程闪存	687
24.3.1	模式设置	687
24.3.2	接口规格	687
24.3.3	内置内存的限制	688
24.3.4	工作命令	688
24.3.4.1	RAM 传输	
24.3.4.2	闪存芯片擦除和保护位擦除	
24.3.5	无关命令的共同工作	689
24.3.5.1	串行工作模式决定	
24.3.5.2	获取响应数据	
24.3.5.3	密码决定	
24.3.5.4	CHECK SUM 计算	
24.3.6	串行工作模式决定的通信规则	694
24.3.7	RAM 传输时通信规则	695
24.3.8	闪存芯片擦除和保护位擦除的通信规则	697
24.3.9	启动程序整体流程图	698
24.3.10	使用 BOOT ROM 中的程序对闪存进行重编程的步骤	699
24.3.10.1	步-1	
24.3.10.2	步-2	
24.3.10.3	步-3	
24.3.10.4	步-4	
24.3.10.5	步-5	
24.3.10.6	步-6	
24.4	用户启动模式下重新编程	702
24.4.1	(1-A) 储存在内存的编程程序处理	702
24.4.1.1	步-1	
24.4.1.2	步-2	
24.4.1.3	步-3	
24.4.1.4	步-4	
24.4.1.5	步-5	
24.4.1.6	步-6	
24.4.2	(1-B) 传输自外部主机的编程程序处理	706
24.4.2.1	步-1	
24.4.2.2	步-2	
24.4.2.3	步-3	
24.4.2.4	步-4	
24.4.2.5	步-5	
24.4.2.6	步-6	
24.5	如何重新编程使用用户启动模式的闪存	710
24.5.1	闪存重新编程处理例	710
24.5.1.1	步-1	
24.5.1.2	步-2	
24.5.1.3	步-3	
24.5.1.4	步-4	
24.5.1.5	步-5	
24.5.1.6	步-6	
24.5.1.7	步-7	
24.5.1.8	步-8	
24.5.1.9	步-9	

25. 调试接口

25.1	规格总览	715
25.2	SWJ-DP	715
25.3	ETM	715
25.4	外设功能停止模式	715
25.5	带调试工具连接	716
25.5.1	关于带调试工具连接	716
25.5.2	使用用作通用目的的端口的调试接口的引脚要点	716

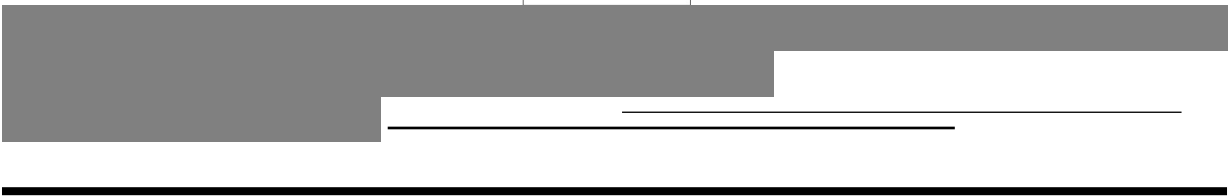
26. 端口部分等效电路图

26.1 端口	717
26.2 模拟 端子	718
26.3 控制 端子	718
26.4 时钟 端子	718

27. 电气 特征

27.1 绝对 最大 额定值	719
27.2 DC 电气 特征 (1/2)	720
27.3 DC 电气 特征 (2/2)	721
27.4 12-位 ADC 电气 特征	722
27.5 AC 电气 特征	723
27.5.1 AC 测量 条件	723
27.5.2 串行 通道 时序 (SIO/UART)	723
27.5.2.1 I/O 接口 模式 (VDD = 4.5 ~ 5.5V)	
27.5.3 串行 总线 接口 (I2C)	725
27.5.3.1 I2C 模式	
27.5.4 16-位 定时器 / 事件 计数器 (TMRB)	726
27.5.4.1 事件 计数器	
27.5.4.2 捕捉	
27.5.5 外部 中断	726
27.5.6 调试 通信	727
27.5.6.1 AC 测量 条件	
27.5.6.2 SWD 接口	
27.5.6.3 JTAG 接口	
27.5.7 EMT 跟踪	728
27.5.8 闪存 特征	729
27.5.9 芯片 上 振荡器	729
27.5.10 外部 振荡器	729
27.6 振荡 电路	730
27.6.1 陶瓷 振荡器	730

28. 封装 尺寸



CMOS 32-位 微控制器

TMPM475FDFG/FZFG/FYFG

TMPM475FDFG/FZFG/FYFG 为 32-位 RISC 微处理器系列带 ARM®Cortex®-M4F 微处理器核。
TMPM475FDFG/FZFG/FYFG 特征如下:

1.1 特征

1. ARM®Cortex®-M4F 微处理器核

a. 通过使用 Thumb®-2 指令, 实现了代码效率的提高。

新 16-位 Thumb 指令来提高程序流

新 32-位 Thumb 指令来提高性能

新 Thumb 混合 16-/32-位 指令组可制作更快,更高效代码。

b. 能实现高性能和低功耗。

[高性能]

- 32-位 乘法 ($32 \times 32 = 32$ 位) 和 乘积工作 ($32 \times 32 = 32$ 位) 可以用单时钟执行。

- SIMD (单指令 乘法数据) 工作可用一个单时钟执行。

- 除法需要 2 到 12 周期以内。

[低功耗]

- 使用低功耗库的优化设计

- 停止微控制器核工作的待机功能

c. 适合于实时控制的高-速中断响应

- 可中断的长指令

- 由硬件自动处理堆栈

2. 单-精度 浮-点 执行 单元 (FPU)

- 符合 IEEE754 规范

- 加法/减法/乘法可以用单时钟执行。乘积工作可以用 3 时钟执行。

- CPU 以外专用数据寄存器可执行并列处理。

3. 片上程序内存和数据内存

- 片上 RAM :

TMPM475FDFG : 34 K 字节

TMPM475FZFG : 34 K 字节

TMPM475FYFG : 18 K 字节

- 片上 闪存 ROM :

TMPM475FDFG : 512 K 字节

TMPM475FZFG : 384 K 字节

TMPM475FYFG : 256 K 字节

4. μ DMA 控制器 (μ DMAC) : 32 通道 / 1 单元
 - 传输 模式 : 内置映射和外设 功能
5. 时钟 生成器 (CG)
 - 配备 1 单元内置 PLL。 (8x, 10x, 12x)
 - 时钟 齿轮 功能 : 高-速 时钟可分割为 1/1, 1/2, 1/4, 1/8 或 1/16.
6. 低 功耗 功能 IDLE, STOP
7. 输入/ 输出 端口 (端口) : 79 引脚
8. 16-位 定时器 (TMRB) : 10 通道
 - 16-位 间隔 定时器 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
9. 看门狗 定时器 (WDT) : 1 通道
 - 看门狗 定时器 (WDT) 生成 复位或非-可屏蔽 中断 (NMI) .
10. 串行 通道 (SIO/UART) : 4 通道
 - UART 或同步 模式可选
 - 传输 FIFO: 4-级 8-位 宽, 接收 FIFO: 4-级 8-位 宽
11. 串行 总线 接口 (I2C) : 1 通道
 - 通信 率 100kbps / 400kbps
12. CAN 控制器 (CAN) : 1 单元
 - 符合 CAN 版本 2.0 B (有效)
 - 32 邮箱
 - CAN 总线波特率达 1 Mbps
13. 上电复位 功能 (POR) : 1 单元
14. 电压 检测 功能 (VLTD) : 1 单元
15. 振荡 频率 检测 功能 (OFD) : 1 单元
16. 向量 引擎 (A-VE) : 2 通道
 - 电机 控制计算 电路
 - 对应 2 电机

17. 可编程 电机 驱动器 (PMD) : 2 通道

- 3 相 补充 PWM 生成器
- 同步 AD 转换 开始 触发器 生成器
- 紧急 保护 功能 (EMG)

18. 编码器输入 电路 (A-ENC) : 2 通道

- 位置 检测 功能

支持增量 类型 编码器 (AB 编码器 和 ABZ 编码器).

支持霍尔传感器 IC (1 到 3 相 输入)。

支持 120-度传导时 3-相 BLDC 电机的诱导电压过零检测。

- 输入电路中包含数字-噪声过滤器。
- 采样输入可以用 PMD 的 PWM 信号执行同步。
- 定时器 计数器 功能

16-位 增/减 计数器根据使用编码和霍尔传感器 IC 的位置检测。

16-位 定时器 计数器/捕捉计数工作可以在规定的周期里设置。

32-位 定时器 计数器/捕捉

19. 12-位 AD 转换器 (ADC) : 2 单元

模拟 输入到单元 A : 12 通道

模拟 输入到单元 B : 11 通道

外部 模拟 输入 引脚到单元 A : 9 引脚

外部 模拟 输入 引脚到单元 A 和 B : 3 引脚

外部 模拟 输入 引脚到单元 B : 8 引脚

- 通过内部 触发器开始: TMRB 中断 / PMD 触发器
- 恒定 转换 模式
- AD 监视器 2 通道
- 转换 速 1.0 μ sec (@ADC 转换 时钟 = 120 MHz)

20. 中断 源 : 优先级顺序可以设置为 7 级。

- 内部 80 因素 (除看门狗 定时器 中断)
- 外部 16 因素

21. 输入 数字 噪声 过滤 (DNF) : 外部 中断 源 16 因素

22. 字顺

小 字顺

23. 内部 高-速 振荡 电路 : 10 MHz

24. 内部 高-速 振荡 调节 功能 (TRMOSC) : 1 单元

25. 最大 工作 频率 : 120 MHz

26. 工作 电压 范围

- DVDD5 = 4.5V 到 5.5V (fsys=120MHz)
所有功能 工作
- DVDD5 = 3.9V 到 4.5V (fsys=120MHz)
12-位 ADC 和 AC 电气 特征, 和闪存写的限制。

27. 温度 范围

-40 °C 到 85 °C

28. 封装

LQFP100 (14mm×14mm, 0.5mm 间距)

1.2 方块图

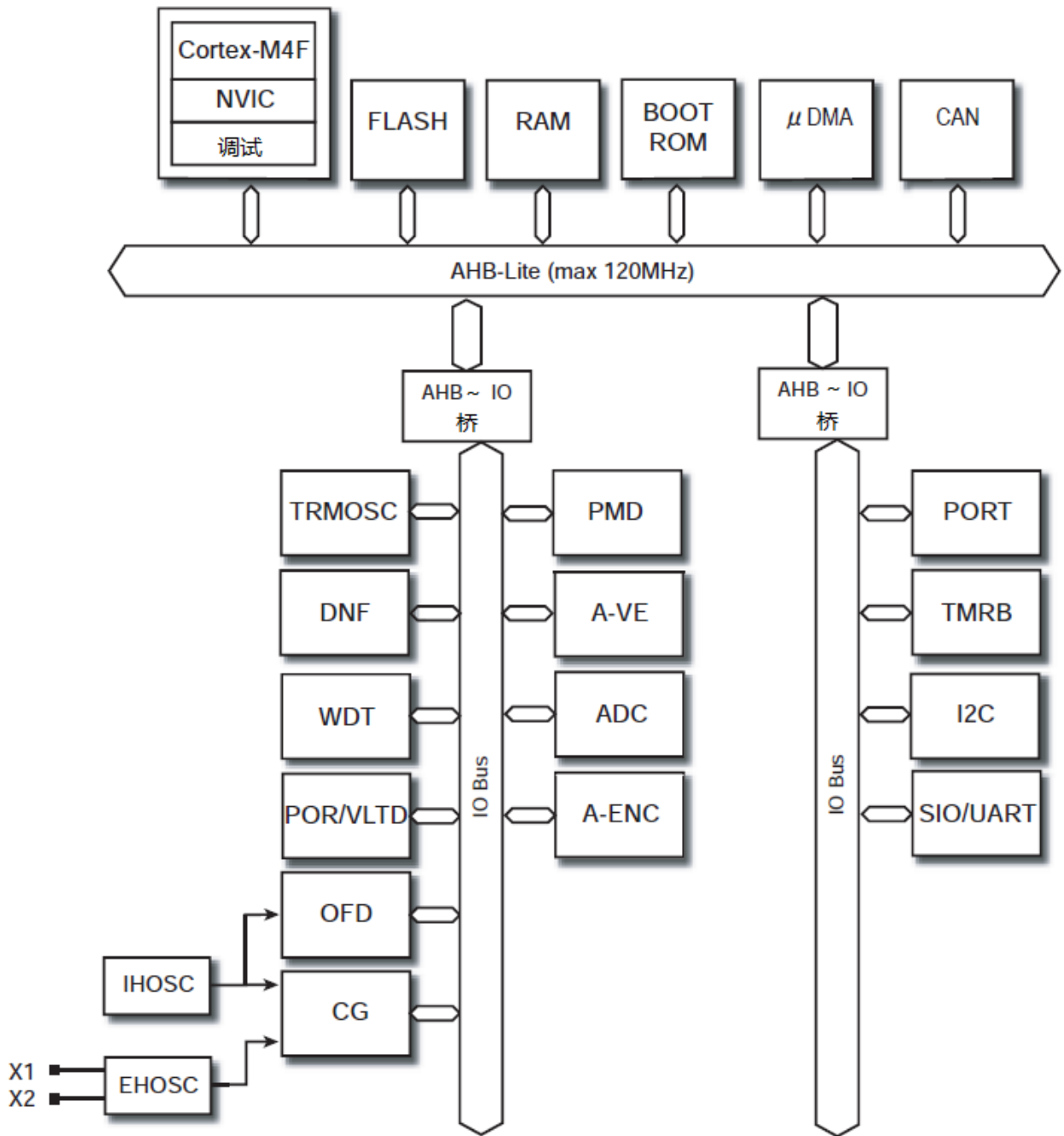


图 1-1 TPM475FDFG/FZFG/FYFG 方块图

1.3 引脚 布局 (顶视)

TMPM475FDFG/FZFG/FYFG 引脚布局为下图。

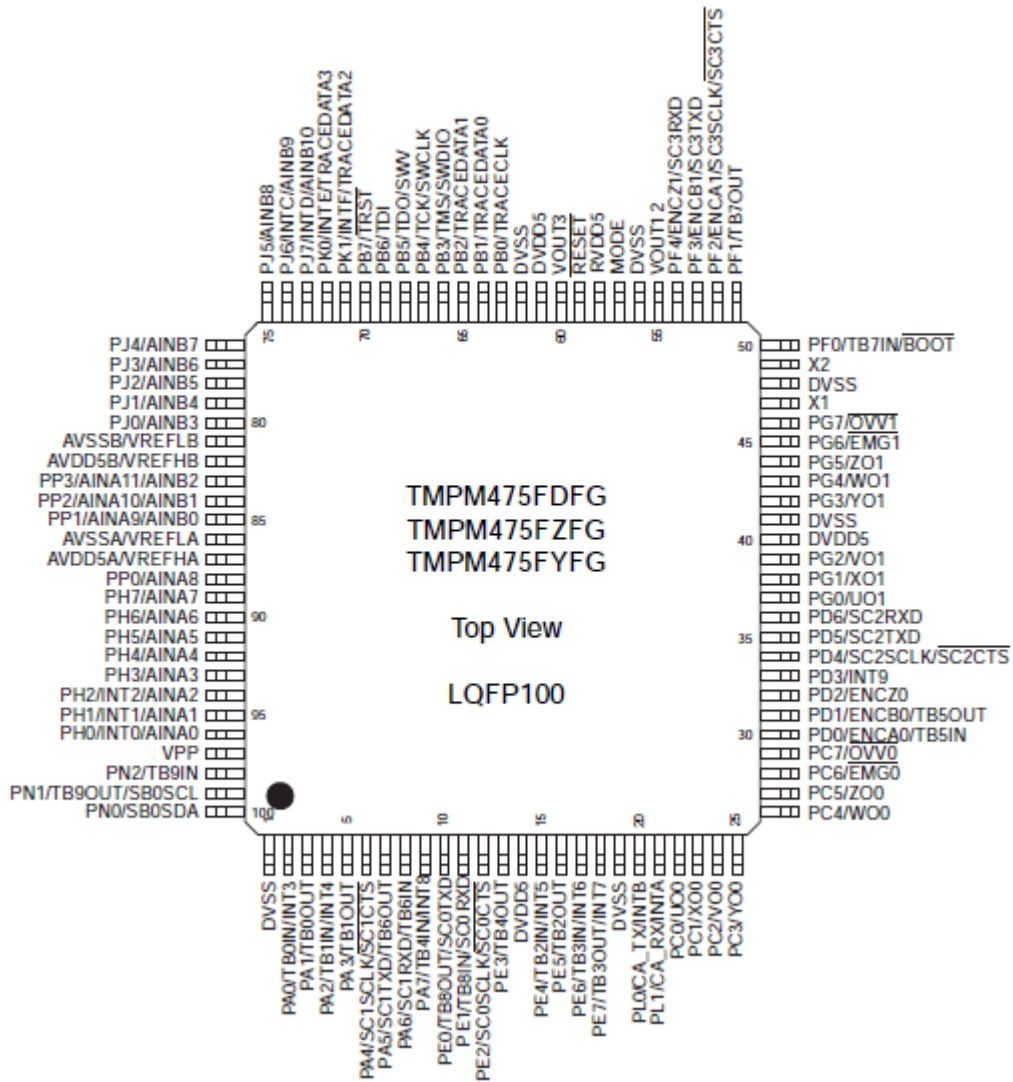


图 1-2 引脚 布局 (LQFP100)

1.4 引脚名和功能

1.4.1 引脚名和功能各外设功能，控制引脚和电源引脚

1.4.1.1 外设功能

表 1-1 引脚名和功能

外设功能	引脚名	输入或输出	功能
外部中断	INTx	输入	外部中断输入引脚 x 外部中断输入引脚 x 有噪声过滤 (过滤宽 30ns typ.).
16-位定时器 / 偶数计数器	TBxIN	输入	输入捕捉输入引脚
	TBxOUT	输出	输出引脚
SIO/UART	SCxTXD	输出	数据输出引脚
	SCxRXD	输入	数据输入引脚
	SCxSCLK	I/O	时钟输入 / 输出引脚
	$\overline{\text{SCxCTS}}$	输入	握手输入引脚
I2C	SBxSDA	I/O	数据输入 / 输出引脚
	SBxSCL	I/O	时钟输入 / 输出引脚
CAN	CA_TX	输出	传输数据输出引脚
	CA_RX	输入	接收数据输入引脚
模拟数字转换器	AINAx	输入	模拟输入引脚
	AINBx	输入	模拟输入引脚
编码器	ENCAx	输入	编码器输入引脚
	ENCBx	输入	编码器输入引脚
	ENCZx	输入	编码器输入引脚
PMD	XOx	输出	X-相输出引脚
	YOx	输出	Y-相输出引脚
	ZOx	输出	Z-相输出引脚
	UOx	输出	U-相输出引脚
	VOx	输出	V-相输出引脚
	WOx	输出	W-相输出引脚
	$\overline{\text{EMGx}}$	输入	紧急状态检测输入
	$\overline{\text{OVVx}}$	输入	过压检测输入

1.4.1.2 调试 功能

表 1-2 引脚名和功能

引脚名	输入或输出	功能
TMS	输入	JTAG 测试模式选择输入引脚
TCK	输入	JTAG 串行数据输入引脚
TDO	输出	JTAG 串行数据输出引脚
TDI	输入	JTAG 测试复位输入引脚
TRST	输入	串行线复位输入引脚
SWDIO	I/O	串行线数据输入/输出引脚
SWCLK	输入	串行线时钟输入引脚
SWV	输出	串行线观察输出引脚
TRACECLK	输出	跟踪时钟输出引脚
TRACEDATA0	输出	跟踪数据输出引脚 0
TRACEDATA1	输出	跟踪数据输出引脚 1
TRACEDATA2	输出	跟踪数据输出引脚 2
TRACEDATA3	输出	跟踪数据输出引脚 3

1.4.1.3 控制 功能

表 1-3 引脚名和功能

引脚名	输入或输出	功能
X1	输入	高频率共振器连接引脚
X2	输出	高频率共振器连接引脚
MODE	输入	模式引脚 此引脚必须固定到低电平。
$\overline{\text{RESET}}$	输入	复位信号输入引脚
$\overline{\text{BOOT}}$	输入	启动模式控制引脚 启动模式控制引脚在复位信号输入引脚上升缘采样。 TMPM475FDG/FZFG/FYFG 改变单启动模式时启动模式控制引脚为“低”。 TMPM475FDG/FZFG/FYFG 改变单芯片模式时启动模式控制引脚为“高”。 详细参考“闪存映射”部分。

1.4.1.4 电源 引脚

表 1-4 引脚名和功能

电源 引脚 名	功能
VOUT12	稳压器的用电容 (3.3 到 4.7 μ F) 连接的引脚
VOUT3	稳压器的用电容 (3.3 到 4.7 μ F) 连接的引脚
VPP	(此 引脚 必须开路.)
RVDD5	稳压器的电源引脚
DVDD5	数字电路 DVDD5 电源引脚提供 以下引 脚 . PA, PB, PC, PD, PE, PF, PG, PL, PK, PN, X1, X2, MODE, RESET
DVSS	数字 电路的 GND 引脚
AVDD5A AVDD5B	模拟 电路电源引脚 (注 1) AVDD5A 提供 以下引脚 . PH, PP0 AVDD5B 提供 以下引脚 . PJ, PP1 到 3
AVSSA AVSSB	ADC 的 GND 引脚 (注 2)
VREFHA VREFHB	ADC 的参考电源 引脚
VREFLA VREFLB	ADC 的参考电源 引脚

注 1: 必须连接电源即使 AD 转换器不被使用.

注 2: AVSSB 必须连接 GND 即使 AD 转换器不被使用.

1.4.1.5 电源引脚间电容

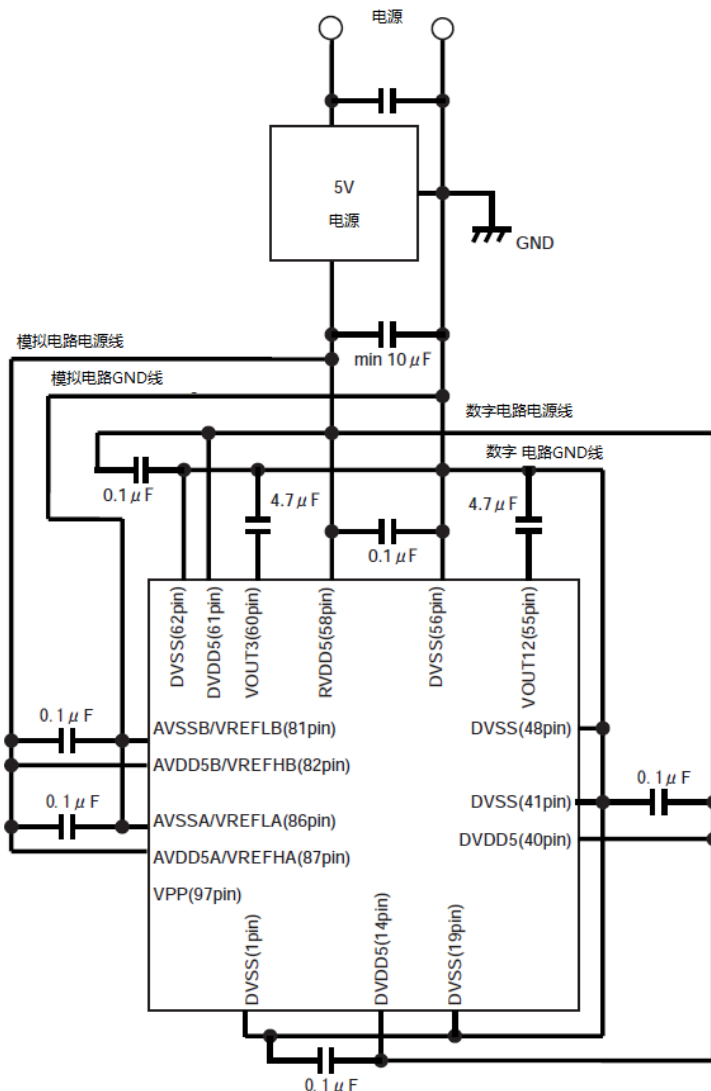


图 1-3 电源引脚连接电容电路

- 注 1: 5V 电源 输出 电容 ($\text{min } 10 \mu\text{F}$) 必须放在离58 引脚 RVDD5, 56 引脚 DVSS的最短距离。
- 注 2: 5V 电源和 GND 线必须分开给模拟电路和数字电路靠近在这个电容和 58 引脚 RVDD5, 56 引脚 DVSS之间的5V电源输出电容 ($\text{min } 10 \mu\text{F}$)。
- 注 3: 电源和 GND 线必须临近且布线。当远离时,电源周期通过电源电路的电容有利于电源和 GND 线,并且他们将是接收高频率噪声的天线。
- 注 4: 稳压器的 VOUT3 和 VOUT12电容必须相同容量,他们必须放在离56 引脚 DVSS的最短距离。
- 注 5: 14 引脚 和 19 引脚, 40 引脚 和 41 引脚, 58 引脚 和 56 引脚, 61 引脚 和 62 引脚之间的电容必须放在离引脚的最短距离。
- 注 6: 82 引脚 和 81 引脚, 87 引脚 和 86 引脚之间的电容必须放在离引脚的最短距离。

1.4.2 TMPM475FDFG/FZFG/FYFG 引脚名和功能

1.4.2.1 引脚名和功能 表详细

表中符号的意思如下所示。

1. 功能 A

没有功能寄存器设置的定义的功能在此单元显示。

2. 功能 B

有功能寄存器设置的定义的功能在此单元显示。此单元中的数字对应于功能寄存器的数。

3. 引脚 规格

表中符号的意思如下所示。

- SMT/CMOS :输入门类型
 - SMT : Schmitt 输入
 - CMOS : CMOS 输入
- 5V_T : 5V 精度支持
 - Yes : 支持
 - N/A : 不支持
- OD : 可编程开漏 输出支持
 - Yes : 支持
 - N/A : 不支持
- PU/PD: 支持可编程 上拉 / 下拉
 - PU : 支持可编程 上拉
 - PD : 支持可编程 下拉

1.4.2.2 端口 / 调试 引脚

表 1-5 引脚名和功能 <按端口归类>

Pin No.	PORT	功能 A	功能 B					端口规格		
			1	2	3	4	5	PU/PD	OD	SMT/CMOS
PORT A										
2	PA0	INT3	TB0IN					PU/PD	Yes	SMT
3	PA1		TB0OUT					PU/PD	Yes	SMT
4	PA2	INT4	TB1IN					PU/PD	Yes	SMT
5	PA3		TB1OUT					PU/PD	Yes	SMT
6	PA4		SC1SCLK	SC1CTST				PU/PD	Yes	SMT
7	PA5		SC1TXD	TB6OUT				PU/PD	Yes	SMT
8	PA6		SC1RXD	TB6IN				PU/PD	Yes	SMT
9	PA7	INT8	TB4IN					PU/PD	Yes	SMT
PORT B										
63	PB0		TRACECLK					PU/PD	Yes	SMT
64	PB1		TRACEDATA0					PU/PD	Yes	SMT
65	PB2		TRACEDATA1					PU/PD	Yes	SMT
66	PB3		TMS/SWDIO					PU/PD	Yes	SMT
67	PB4		TCK/SWCLK					PU/PD	Yes	SMT
68	PB5		TD0/SWV					PU/PD	Yes	SMT
69	PB6		TDI					PU/PD	Yes	SMT
70	PB7		TRST					PU/PD	Yes	SMT
PORT C										
22	PC0		U00					PU/PD	Yes	SMT
23	PC1		X00					PU/PD	Yes	SMT
24	PC2		V00					PU/PD	Yes	SMT
25	PC3		Y00					PU/PD	Yes	SMT
26	PC4		W00					PU/PD	Yes	SMT
27	PC5		Z00					PU/PD	Yes	SMT
28	PC6		EMG0					PU/PD	Yes	SMT
29	PC7		OVV0					PU/PD	Yes	SMT
PORT D										
30	PD0		ENCA0	TB5IN				PU/PD	Yes	SMT
31	PD1		ENCB0	TB5OUT				PU/PD	Yes	SMT
32	PD2		ENCZD					PU/PD	Yes	SMT
33	PD3	INT9						PU/PD	Yes	SMT
34	PD4		SC2SCLK	SC2CTS				PU/PD	Yes	SMT
35	PD5		SC2TXD					PU/PD	Yes	SMT
36	PD6		SC2RXD					PU/PD	Yes	SMT
PORTE										
10	PE0		SC0TXD	TB8OUT				PU/PD	Yes	SMT
11	PE1		SC0RXD	TB8IN				PU/PD	Yes	SMT
12	PE2		SC0SCLK	SC0CTS				PU/PD	Yes	SMT
13	PE3		TB4OUT					PU/PD	Yes	SMT
15	PE4	INT5	TB2IN					PU/PD	Yes	SMT
16	PE5		TB2OUT					PU/PD	Yes	SMT

表 1-5 引脚名和功能 <按端口归类>

Pin No.	PORT	功能 A	功能 B					端口规格		
			1	2	3	4	5	PUI/PD	OD	SMT/CMOS
17	PE6	INT6	TB3IN					PUI/PD	Yes	SMT
18	PE7	INT7	TB3OUT					PUI/PD	Yes	SMT
PORTF										
50	PF0	BOOT	TB7IN					PUI/PD	Yes	SMT
51	PF1		TB7OUT					PUI/PD	Yes	SMT
52	PF2		ENCA1	SC3SCLK	SC3CTS			PUI/PD	Yes	SMT
53	PF3		ENCB1	SC3TXD				PUI/PD	Yes	SMT
54	PF4		ENCZ1	SC3RXD				PUI/PD	Yes	SMT
PORTG										
37	PG0		UO1					PUI/PD	Yes	SMT
38	PG1		XO1					PUI/PD	Yes	SMT
39	PG2		VO1					PUI/PD	Yes	SMT
42	PG3		YO1					PUI/PD	Yes	SMT
43	PG4		WO1					PUI/PD	Yes	SMT
44	PG5		ZO1					PUI/PD	Yes	SMT
45	PG6		EMGT					PUI/PD	Yes	SMT
46	PG7		OVVT					PUI/PD	Yes	SMT
PORTH										
96	PH0	AINA0 INT0						PUI/PD	Yes	SMT
95	PH1	AINA1 INT1						PUI/PD	Yes	SMT
94	PH2	AINA2 INT2						PUI/PD	Yes	SMT
93	PH3	AINA3						PUI/PD	Yes	SMT
92	PH4	AINA4						PUI/PD	Yes	SMT
91	PH5	AINA5						PUI/PD	Yes	SMT
90	PH6	AINA6						PUI/PD	Yes	SMT
89	PH7	AINA7						PUI/PD	Yes	SMT
PORTJ										
80	PJ0	AINB3						PUI/PD	Yes	SMT
79	PJ1	AINB4						PUI/PD	Yes	SMT
78	PJ2	AINB5						PUI/PD	Yes	SMT
77	PJ3	AINB6						PUI/PD	Yes	SMT
76	PJ4	AINB7						PUI/PD	Yes	SMT
75	PJ5	AINB8						PUI/PD	Yes	SMT
74	PJ6	AINB9 INTC						PUI/PD	Yes	SMT
73	PJ7	AINB10 INTD						PUI/PD	Yes	SMT
PORTK										
72	PK0	INTE	TRACEDATA3					PUI/PD	Yes	SMT
71	PK1	INTF	TRACEDATA2					PUI/PD	Yes	SMT
PORTL										
20	PL0	INTB	CA_TX					PUI/PD	Yes	SMT
21	PL1	INTA	CA_RX					PUI/PD	Yes	SMT

表 1-5 引脚名和功能 <按端口归类>

Pin No.	PORT	功能 A	功能 B					端口规格			
			1	2	3	4	5	PU/PD	OD	SMT/ CMOS	
PORT N											
100	PN0			SB0SDA					PU/PD	Yes	SMT
99	PN1		TB9OUT	SB0SCL					PU/PD	Yes	SMT
98	PN2		TB9IN						PU/PD	Yes	SMT
PORT P											
88	PP0	AINA8							PU/PD	Yes	SMT
85	PP1	AINA9 AINB0							PU/PD	Yes	SMT
84	PP2	AINA10 AINB1							PU/PD	Yes	SMT
83	PP3	AINA11 AINB2							PU/PD	Yes	SMT

1.4.2.3 控制 引脚

表 1-6 引脚号和引脚 名

引脚 No.	控制 功能 引脚 名
47	X1
49	X2
57	MODE
59	$\overline{\text{RESET}}$
50	$\overline{\text{BOOT}}$

1.4.2.4 电源引脚

表 1-7 引脚号和引脚 名

引脚 No.	电源 引脚 名
55	VOUT12
60	VOUT3
58	RVDD5
97	VPP
14, 40, 61	DVDD5
1, 19, 41, 48, 56, 62	DVSS
87	AVDD5A VREFHA
82	AVDD5B VREFHB
86	VREFLB AVSSB
81	VREFLB AVSSB

2. 产品信息

此章节描述外设功能-相关通道或单元数, 引脚信息和产品规格功能信息. 使用此章节与外设功能章节结合.

- "2.1.1 DMA 控制器 (DMAC)"
- "2.1.2 16-位 定时器 / 事件 计数器 (TMRB)"
- "2.1.3 串行 通道 (SIO/UART)"
- "2.1.4 串行 总线 接口 (I2C)"
- "2.1.5 CAN 控制器 (CAN)"
- "2.1.6 向量 引擎 (A-VE)"
- "2.1.7 电机 控制 电路 (PMD : 可编程 电机 驱动器)"
- "2.1.8 编码器 输入 电路 (A-ENC)"
- "2.1.9 模拟/数字 转换器 (ADC)"
- "2.1.10 看门狗 定时器(WDT)"
- "2.1.11 调试 接口"

2.1 各外设功能信息

2.1.1 DMA 控制器 (DMAC)

TMPM475FDFG/FZFG/FYFG 含有 1 单元内置 DMA 控制器。

表 2-1 DMA 请求 表

通道	Burst	单
0	SIO/UART0 接收	-
1	SIO/UART0 传输	-
2	SIO/UART1 接收	-
3	SIO/UART1 传输	-
4	SIO/UART2 接收	-
5	SIO/UART2 传输	-
6	SIO/UART3 接收	-
7	SIO/UART3 传输	-
8	由定时器 A 触发的 AD 转换完成	-
9	由定时器 B 触发的 AD 转换完成	-
10	由软件 A 开始的 AD 转换完成	-
11	由软件 B 开始的 AD 转换完成	-
12	由 PMD0 A 触发的 AD 转换完成	-
13	由 PMD0 B 触发的 AD 转换完成	-
14	由 PMD1A 触发的 AD 转换完成	-
15	由 PMD1B 触发的 AD 转换完成	-
16	PMD0 PWM 中断	-
17	PMD1 PWM 中断	-
18	-	-
19	-	-
20	TMRB0 输入 捕捉 0	-
21	TMRB1 输入 捕捉 0	-
22	TMRB2 输入 捕捉 0	-
23	TMRB3 输入 捕捉 0	-
24	VE 中断 0	-
25	VE 中断 1	-
26	TMRB0 比较 匹配 (注)	-
27	TMRB1 比较 匹配 (注)	-
28	TMRB2 比较 匹配 (注)	-
29	TMRB3 比较 匹配 (注)	-
30	由任务的 VE0 中断完成	-
31	由任务的 VE1 中断完成	-

注: 在 TMRB 中断的相同条件下 TMRB 的 DMA 传输请求产生。当递增-计数器 匹配定时器寄存器 0/1, 或递增-计数器溢出时 TMRB 中断产生。必要时用中断屏蔽 寄存器 (TBxIM)屏蔽不必要的因素。

2.1.2 16-位 定时器 / 事件 计数器 (TMRB)

TMPM475FDFG/FZFG/FYFG 含有 10 通道 TMRB。

表 2-2 引脚 规格

通道	TBxOUT	TBxIN
TMRB0	PA1	PA0
TMRB1	PA3	PA2
TMRB2	PE5	PE4
TMRB3	PE7	PE6
TMRB4	PE3	PA7
TMRB5	PD1	PD0
TMRB6	PA5	PA6
TMRB7	PF1	PF0
TMRB8	PE0	PE1
TMRB9	PN1	PN2

表 2-3 中断

通道	输入 捕捉	比较 匹配 检测 / 溢出
TMRB0	INTCAP00, INTCAP01	INTTB00, INTTB01
TMRB1	INTCAP10, INTCAP11	INTTB10, INTTB11
TMRB2	INTCAP20, INTCAP21	INTTB20, INTTB21
TMRB3	INTCAP30, INTCAP31	INTTB30, INTTB31
TMRB4	INTCAP40, INTCAP41	INTTB40, INTTB41
TMRB5	INTCAP50, INTCAP51	INTTB50, INTTB51
TMRB6	INTCAP60, INTCAP61	INTTB60, INTTB61
TMRB7	INTCAP70, INTCAP71	INTTB70, INTTB71
TMRB8	INTCAP80, INTCAP81	INTTB80, INTTB81
TMRB9	INTCAP90, INTCAP91	INTTB90, INTTB91

表 2-4 同步开始 规格

主机 通道	从 通道
TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6
TMRB7	TMRB8, TMRB9

表 2-5 捕捉 触发器 规格

触发器 输入 通道	触发器 输出
TMRB0	ENC0 分割 脉冲
TMRB1	ENC1 分割 脉冲

2.1.3 串行 通道 (SIO/UART)

TMPM475FDFG/FZFG/FYFG 含有 4 通道 SIO/UART.

表 2-6 引脚规格

通道	SCxTXD	SCxRXD	SCxSCLK	SCxCTS
SC0	PE0	PE1	PE2	PE2
SC1	PA5	PA6	PA4	PA4
SC2	PD5	PD6	PD4	PD4
SC3	PF3	PF4	PF2	PF2

表 2-7 中断

通道	串行接收	串行传输
SC0	INTRX0	INTTX0
SC1	INTRX1	INTTX1
SC2	INTRX2	INTTX2
SC3	INTRX3	INTTX3

表 2-8 内部连接规格

通道	UART 源时钟
SC0	TMRB4
SC1	TMRB4
SC2	TMRB7
SC3	TMRB7

2.1.4 串行总线接口 (I2C)

TMPM475FDFG/FZFG/FYFG 含有 1 通道 I2C 和不支持 SIO.

表 2-9 引脚规格

通道	SBxSDA	SBxSCL
I2C0	PN0	PN1

表 2-10 中断

通道	中断
I2C0	INTSBI0

2.1.5 CAN 控制器 (CAN)

TMPM475FDFG/FZFG/FYFG 含有 1 单元 内置 CAN.

表 2-11 引脚规格

单元	CA_TX	CA_RX
CAN	PL0	PL1

表 2-12 中断

单元	接收完成中断	传输完成中断	全球中断
CAN	INTCANRX	INTCANTX	INTCANGB

2.1.6 向量引擎 (A-VE)

TMPM475FDFG/FZFG/FYFG 含有 2 通道内置 A-VE.

表 2-13 引脚规格

通道	中断
VE0	INTVCN0
VE1	INTVCN1

表 2-14 内部连接规格

通道	PWM 中断信号输入	ADC 转换完成信号输入	ADC 转换结果输入
VE0	INTPMD0	INTADAPDA INTADBPDA	ADAREG0 到 3
VE1	INTPMD1	INTADAPDB INTADBPDB	ADBREG0 到 3

2.1.7 电机控制电路 (PMD : 可编程电机驱动器)

TMPM475FDFG/FZFG/FYFG 含有 2 通道 PMD.

表 2-15 引脚规格

通道	OVVx 过压状态 检测输入	EMGx 紧急状态 检测输入	ZOx Z-相输出 引脚	WOx W-相 输出引脚	YOx Y-相输出 引脚	VOx V-相输出 引脚	XOx X-相输出 引脚	UOx U-相输出 引脚
PMD0	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PMD1	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0

表 2-16 中断

通道	OVV 中断	EMG 中断	PWM 中断
PMD0	INTOVV0	INTEMG0	INTPMD0
PMD1	INTOVV1	INTEMG1	INTPMD1

表 2-17 内部连接规格 (1/2)

通道	EMG 保护 释放输入 (VE)	PWM 比较 输入 (VE)	输出控制 输入 (VE)	触发器比较 输入 (VE)	选择触发器 输入 (VE)	OVV 输入 (ADC)
PMD0	VEEMGRS0	VECMPU0, V0, W0	VEOUTCR0	VETRGCMP00,01	VETRGSEL0	ADCA 监视器 功能 比较 0, 1 输出
PMD1	VEEMGRS1	VECMPU1, V1, W1	VEOUTCR1	VETRGCMP10,11	VETRGSEL1	ADCB 监视器 功能 比较 0, 1 输出

表 2-18 内部连接规格 (2/2)

通道	MDOUT 传输时序 信号输入
PMD0	INTENC0, INTTB00, CTRGO(ENC0)
PMD1	INTENC1, INTTB10, CTRGO(ENC1)

2.1.8 编码器输入电路 (A-ENC)

TMPM475FDFG/FZFG/FYFG 含有 2 通道 A-ENC.

表 2-19 引脚规格

通道	ENCAx	ENCBx	ENCZx
ENC0	PD0	PD1	PD2
ENC1	PF2	PF3	PF4

表 2-20 中断

通道	中断
ENC0	INTENC0
ENC1	INTENC1

表 2-21 内部连接规格

通道	定时器 脉冲输入	PWM 信号输入
ENC0	TB0OUT	PWMON0
ENC1	TB1OUT	PWMON1

2.1.9 模拟/数字转换器 (ADC)

TMPM475FDFG/FZFG/FYFG 含有 2 单元 ADC (12-位 逐次-近似 模拟-到-数字 转换器).

表 2-22 引脚 规格

通道	AINA0 到 7	AINA8	AINA9 到 11 AINB0 到 2	AINB3 到 10
ADCA	PH0 到 7	PP0	PP1 到 3	-
ADCB	-	-	PP1 到 3	PJ0 到 7

表 2-23 中断

单元	PMD 触发的 AD 转换完成	定时器触发的 AD 转换完成	通过软件开始的 AD 转换完成	AD 转换 监视器功能
ADCA	INTADAPDB	INTADATMR	INTAD 作 FT	INTADACPA INTADACPB
ADCB	INTADBPDB	INTADBTMR	INTADBSFT	INTADBCPA INTADBCPB

表 2-24 内部 连接 规格

单元	PMD 触发器 输入	TMRTRG
ADCA	PMD0TRG0 到 5	INTTB50
ADCB	PMD1TRG0 到 5	INTTB60

2.1.10 看门狗 定时器(WDT)

TMPM475FDFG/FZFG/FYFG, WDMOD(看门狗 定时器 模式 寄存器) 位 2 <I2WDT>不支持, 所以写 "0".

2.1.11 调试 接口

TMPM475FDFG/FZFG/FYFG 支持串行线 调试 端口, JTAG 调试 端口和跟踪输出.

表 2-25 引脚 规格

	TMS SWDIO	TCK SWCLK	TDO SWV	TDI	$\overline{\text{TRST}}$
JTAG 串行线	PB3	PB4	PB5	PB6	PB7

	TRACECLK	TRACEDATA0	TRACEDATA1	TRACEDATA2	TRACEDATA3
跟踪输出	PB0	PB1	PB2	PK1	PK0

3. 处理器核

TX04 系列有高-性能 32-位 处理器核 (ARM Cortex-M4F 处理器核)。关于处理器核工作的信息, 请参考 ARM 公司发布的文档。本章节描述文档中未被解释的 TX04 系列独特功能。

3.1 关于处理器核信息

下表表示 TMPM475FDFG/FZFG/FYFG 处理器核版本。

参考 CPU 核和结构的详细信息, 关于"Cortex-M4 系列处理器"参考 ARM 发布文档, 通过以下 URL:

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

产品名	核版本
TMPM475FDFG/FZFG/ FYFG	r0p1

3.2 可配置选项

Cortex-M4F 核 有选择方块。

下表表示 TMPM475FDFG/FZFG/FYFG 可配置选项。

可配置选项	执行
MPU (映射 保护 单元)	缺席
FPB (闪存 补丁和断点)	两个逐字比较器 六个指令比较器
DWT (数据 观察点和跟踪)	四个 比较器
ITM (Instrumentation Trace Macrocell)	当前
ETM (Embedded Trace Macrocell)	当前
AHB-AP (AHB 进入 端口)	当前
HTM 接口 (AHB Trace Macrocell 接口)	缺席
TPIU (跟踪 端口 接口 单元)	当前
WIC (唤醒 中断 控制器)	缺席
调试 端口 (串行-线 或 JTAG 调试 端口)	当前
FPU (浮 点 单元)	当前
位 绑定	当前
恒定 AHB 控制	禁止

3.3 异常/中断

以下部分描述异常和中断。

3.3.1 中断输入数

中断输入数可以选择性定义于Cortex-M4F 核1 到 240。

TMPM475FDFG/FZFG/FYFG 有 96 中断输入。中断输入数反映在NVIC 寄存器 <INTLINESNUM[3:0]> 位。本产品, 如果读 <INTLINENUM[3:0]> 位, "0x02" 被读出。

3.3.2 优先级中断位数

Cortex-M4F 核可以选择优先级中断位从 3 位到 8 位配置数。

TMPM475FDFG/FZFG/FYFG 有优先级中断位。优先级中断位数被用来分配中断优先级 寄存器和系统处理器优先级寄存器中的优先级。

3.3.3 SysTick

Cortex-M4F 核 有可以生成 SysTick 异常的 SysTick 定时器。

关于 SysTick 异常详细 , 参考"SysTick" 异常和 NVIC 寄存器中 SysTick 寄存器的部分。

3.3.4 SYSRESETREQ

应用中断和复位控制寄存器<SYSSRESETREQ> 位设置时, Cortex-M4F 核输出 SYSSRESETREQ 信号。

输出 SYSRESETREQ 信号时, TMPM475FDFG/FZFG/FYFG 提供与预热复位相同工作。

3.3.5 LOCKUP

当产生不可修复的异常, Cortex-M4F 核输出 LOCKUP 信号来表示包括软件的严重错误。

TMPM475FDFG/FZFG/FYFG 不使用此信号.从 LOCKUP 状态返回,需要使用非-可屏蔽中断 (NMI) 或复位。

3.3.6 备用 错误 状态 寄存器

Cortex-M4F 核提供备用错误状态寄存器 提供 额外的系统错误信息给软件。

然而, TMPM475FDFG/FZFG/FYFG 不定义此功能. 如果备用错误状态寄存器被读, 始终读出 "0x0000_0000".

3.4 事件

Cortex-M4F 核有事件输出信号和事件输入信号. 通过 SEV 指令执行, 输出事件输出信号. 如果事件输入, 核从通过 WFE 指令引起的低-功耗 模式返回。

TMPM475FDFG/FZFG/FYFG 不使用事件输出信号和事件输入信号. 请不要用 SEV 指令和 WFE 指令。

3.5 电源 管理

Cortex-M4F 核 提供使用 SLEEPING 信号和 SLEEPDEEP 信号的电源管理系统。当系统控制寄存器 <SLEEPDEEP> 位被设置时输出 SLEEPDEEP 信号。

以下 环境输出这些 信号:

-等待-给-中断 (WFI) 指令 执行

-等待-给-事件 (WFE) 指令 执行

-当中断-服务-程序 (ISR)退出以防系统控制寄存器 <SLEEPONEXIT> 位设置的时序。

TMPM475FDFG/FZFG/FYFG 不使用SLEEPDEEP 信号所以<SLEEPDEEP> 位必须不被设置。
另外事件信号也不使用所以请不要使用 WFE 指令。

关于电源管理详细, 参考章节 "时钟/模式 控制."

3.6 专用 进入

Cortex-M4F 核中, 编码总线系统 递增端口专用进入。然而 TMPM475FDFG/FZFG/ FYFG 不使用此功能。

3.7 浮 点 单元 (FPU)

此本产品执行ARMv7-M 浮-点扩展 (FPv4-SP)的单精度变异的Cortex-M4F FPU。提供符合ANSI/ IEEE Std 754-2008, IEEE 二进制浮-点算法规范, 以及IEEE 754 规范的浮-点计算功能。

FPU 用 Cortex-M4F 核分享地址总线 and 数据总线, 且协同工作。以 1 时钟执行加,减,和乘, 以 3时钟执行乘和积。使用专用数据寄存器可以做与CPU不同的并行处理。

FPU 支持所有 ARM 结构 参考手册中描述的单-精度数据-处理指令和数据类型。

4. 内存映射

4.1 内存映射

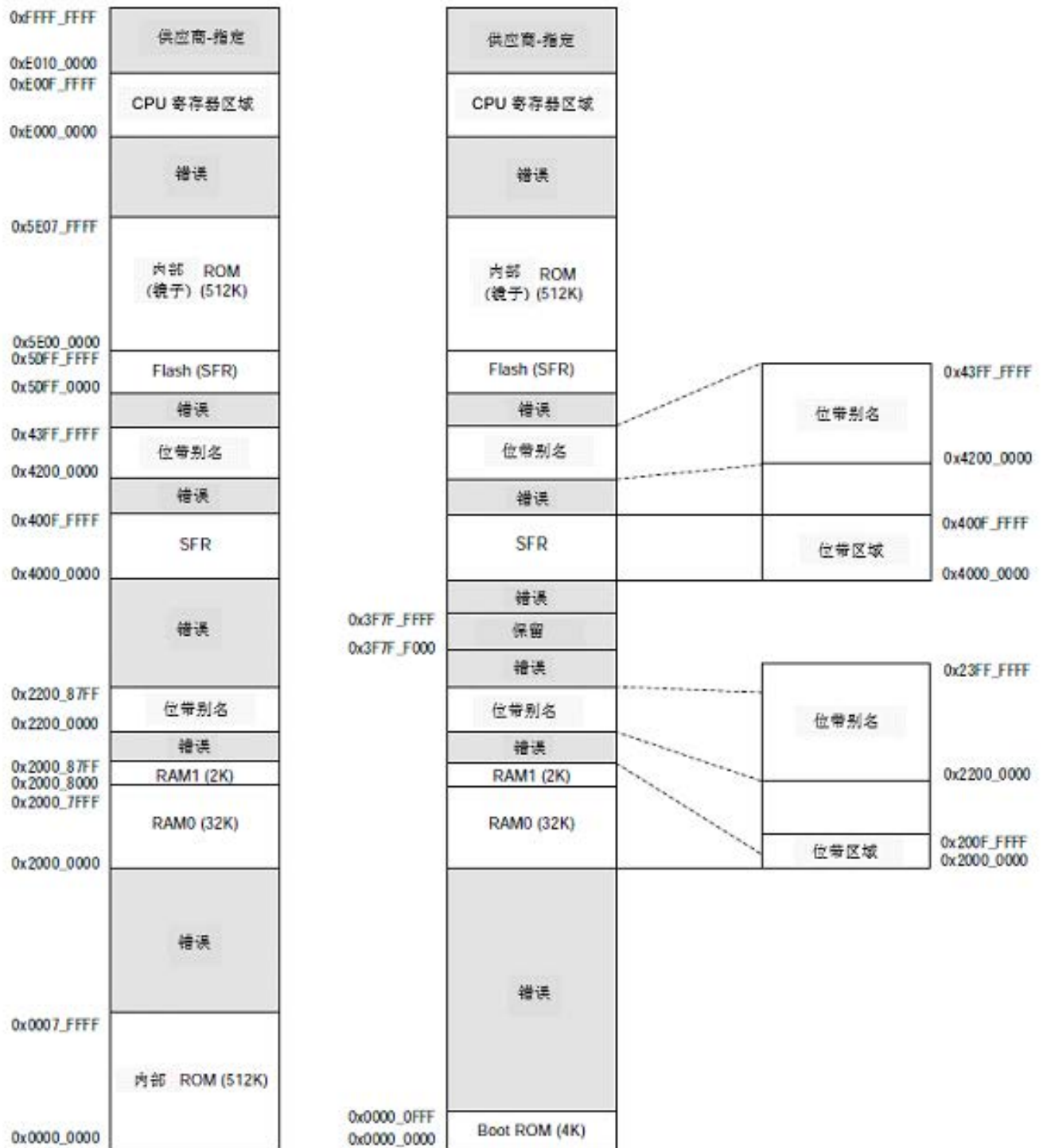
TMPM475FDFG/FZFG/FYFG 内存映射是基于在 ARM Cortex-M4F 处理器核内存映射基础上。TMPM475FDFG/FZFG/FYFG 内部 ROM, 内部 RAM 和特殊功能寄存器(SFR) 各自地映射到代码, SRAM 和 Cortex-M4F 外围区域。特殊功能寄存器(SFR) 指所有输入/输出端口和外围功能控制寄存器。

CPU 寄存器区域是处理器核的内部寄存器区域。

更多有关各区域的信息, 见 "ARM Cortex-M4的ARM 制定文档"。

注意如果内存错误被使能, 进入区域被指导致内存错误的"错误", 或如果内存错误禁用导致的硬错误。另外, 不要进入供应商-指定区域。

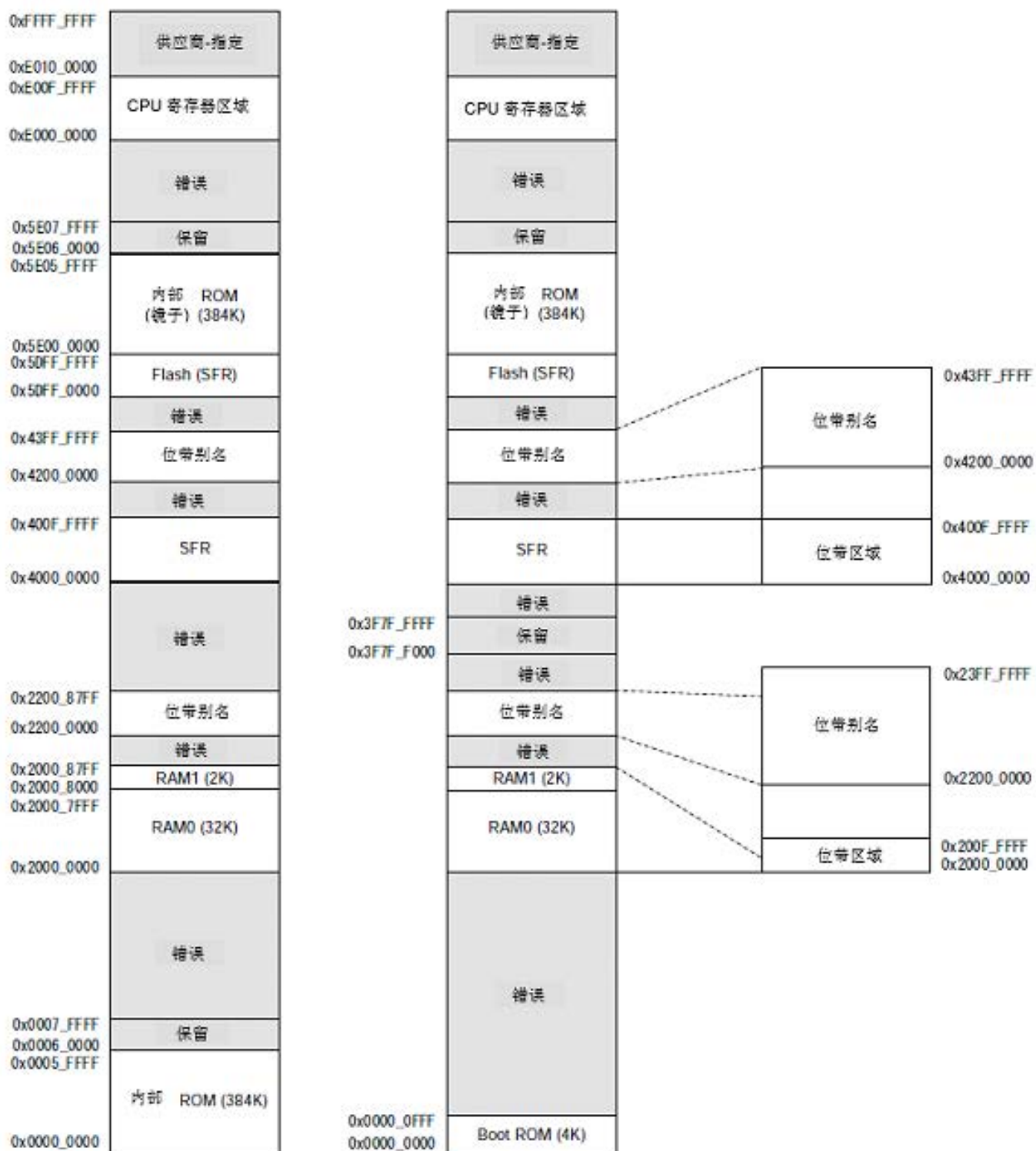
TMPM475FDFG/FZFG/FYFG 内存映射如下。



单芯片模式

单启动模式

图 4-1 内存映射(512KB)



单芯片模式

单启动模式

图 4-2 内存映射(384KB)

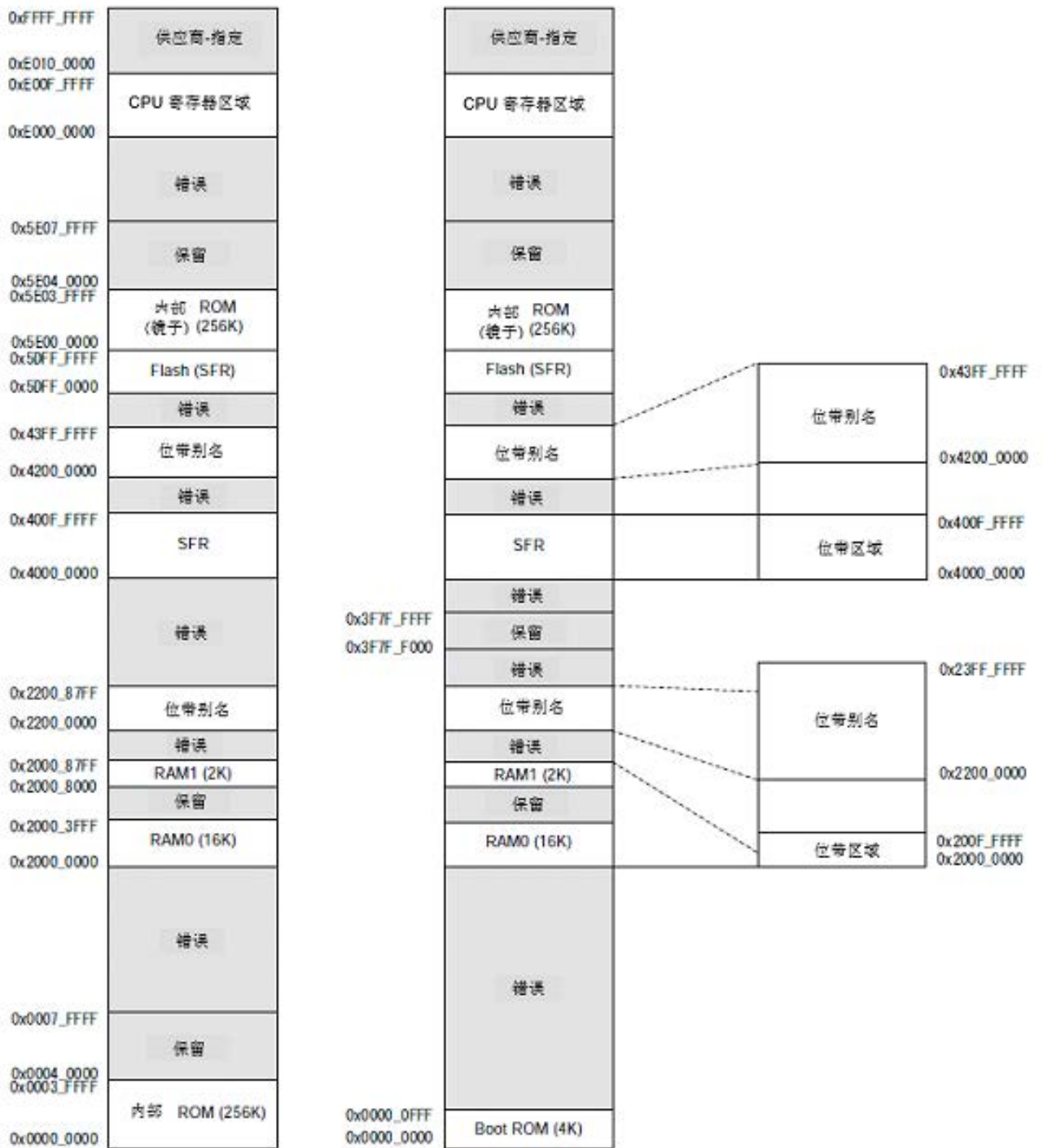


图 4-3 内存映射(256KB)

4.2 总线 矩阵

MCU 包含两个总线主机诸如 CPU 核和 μ DMA 控制器。

总线 主机连接到总线 矩阵从机端口 (S0 到 S5)。总线矩阵中, 主机端口 (M0 到 M15) 通过下图中描述为 (o) 或 (·) 的连接连接到外围功能。(·) 表示连接到镜像区域。

总线矩阵中当乘法从机连接到相同的总线主机线, 如果乘法从机进入被同时生成, 用最小从机数从主机优先进入。

4.2.1 架构

4.2.1.1 单芯片模式

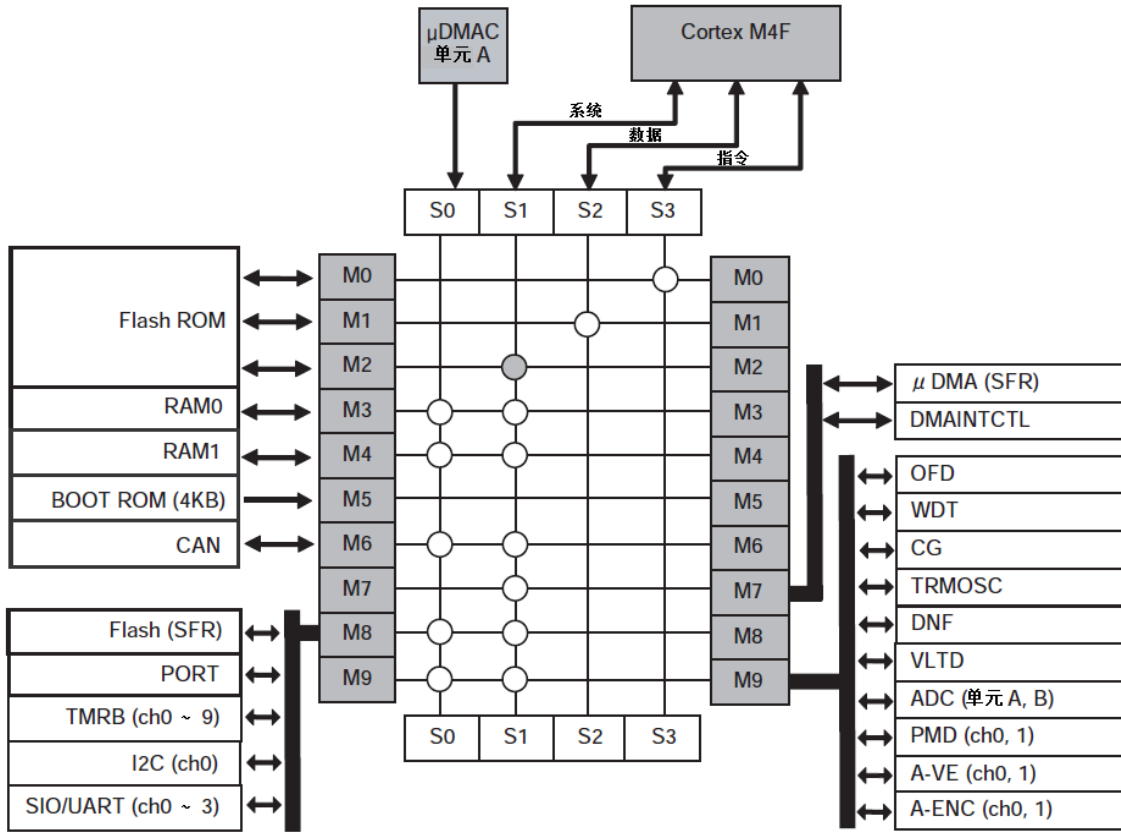


图 4-4 TMPM475DFDG/FZFG/FYFG 总线 矩阵

4.2.1.2 单启动模式

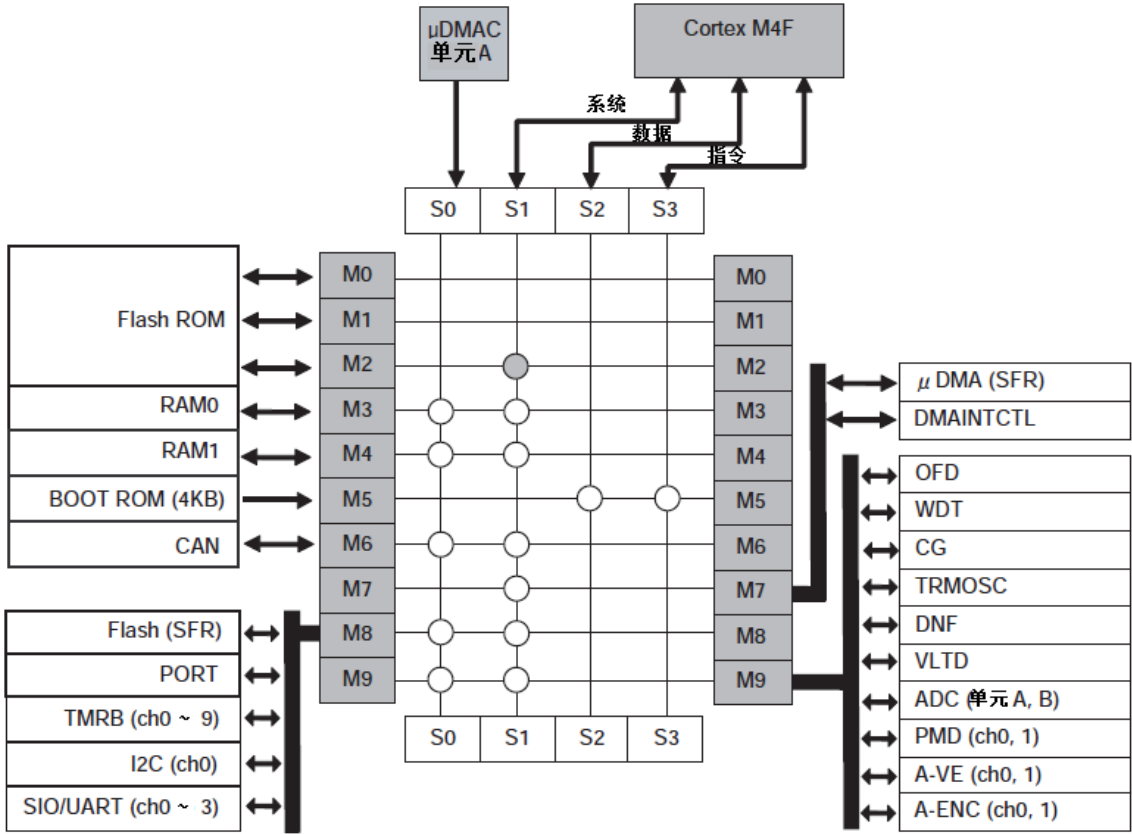


图 4-5 TPM475FDFG/FZFG/FYFG 总线 矩阵

4.2.2 连接表

4.2.2.1 代码区域 / SRAM 区域

(1) 单芯片模式

开始地址	主机		μDMAC 单元A	核 S-总线	核 D-总线	核 I-总线
	从机		S0	S1	S2	S3
0x0000_0000	Flash ROM	M0 M1	错误	错误	o	o
0x0010_0000	错误	-	错误	错误	错误	错误
0x2000_0000	RAM0	M3	o	o	错误	错误
0x2000_8000	RAM1	M4	o	o	错误	错误
0x2000_8800	错误	-	错误	错误	错误	错误
0x2200_0000	位带别名	-	错误	o	错误	错误
0x2400_0000	错误	-	错误	错误	错误	错误

(2) 单启动模式

开始地址	主机		μDMAC 单元A	核 S-总线	核 D-总线	核 I-总线
	从机		S0	S1	S2	S3
0x0000_0000	启动 ROM	M5	错误	错误	o	o
0x0000_1000	错误	-	错误	错误	错误	错误
0x2000_0000	RAM0	M3	o	o	错误	错误
0x2000_8000	RAM1	M4	o	o	错误	错误
0x2003_0400	错误	-	错误	错误	错误	错误
0x2200_0000	位带别名	-	错误	o	错误	错误
0x2400_0000	错误	-	错误	错误	错误	错误
0x3F7F_F000	保留	-	错误	保留	错误	错误
0x3F80_0000	错误	-	错误	错误	错误	错误

注:请不要进入保留地址范围.

4.2.2.2 外设区域 / 外部总线区域

开始地址	主机		μDMAC 单元A	核 S-总线	核 D-总线	核 I-总线
	从机		S0	S1	S2	S3
0x4000_0000	错误	-	错误	错误	错误	错误
0x4000_5000	CAN	M6	0	0	错误	错误
0x4004_C000	μDMAC 单元A (SFR)	M7	-	0	错误	错误
0x4005_F000	DMAIF	M7	-	0	错误	错误
0x400C_0000	PORT	M8	0	0	错误	错误
0x400C_4000	TMRB (ch0 到 -9)	M8	0	0	错误	错误
0x400E_0000	I2C (ch0)	M8	0	0	错误	错误
0x400E_1000	SIO/UART(ch0 到 3)	M8	0	0	错误	错误
0x400F_1000	OFD	M9	0	0	错误	错误
0x400F_2000	WDT	M9	0	0	错误	错误
0x400F_3000	CG	M9	0	0	错误	错误
0x400F_3200	TRMOSC	M9	0	0	错误	错误
0x400F_3400	DNF	M9	0	0	错误	错误
0x400F_4000	LVD	M9	0	0	错误	错误
0x400F_6000	PMD (ch0,1)	M9	0	0	错误	错误
0x400F_7000	A-ENC (ch0,1)	M9	0	0	错误	错误
0x400F_8000	A-VE (ch0, 1)	M9	0	0	错误	错误
0x400F_D000	ADC (单元 A, B)	M9	0	0	错误	错误
0x4010_0000	错误	-	错误	错误	错误	错误
0x4200_0000	位带别名	-	错误	0	错误	错误
0x4400_0000	错误	-	错误	错误	错误	错误
0x5DFF_0000	Flash (SFR)	M8	-	0	错误	错误
0x5E00_0000	Flash (镜像)	M2	0	0	错误	错误
0x5E08_0000	错误	-	错误	错误	错误	错误

4.2.3 外设 功能地址 表

不要进入外设区域控制寄存器以外的地址. 关于控制寄存器详细, 参考各外设功能章节.

外设 功能		基 地址	区 域
CAN 邮箱	ch0	0x4000_5000	SFR0
	ch1	0x4000_5020	SFR0
	ch2	0x4000_5040	SFR0
	ch3	0x4000_5060	SFR0
	ch4	0x4000_5080	SFR0
	ch5	0x4000_50A0	SFR0
	ch6	0x4000_50C0	SFR0
	ch7	0x4000_50E0	SFR0
	ch8	0x4000_5100	SFR0
	ch9	0x4000_5120	SFR0
	ch10	0x4000_5140	SFR0
	ch11	0x4000_5160	SFR0
	ch12	0x4000_5180	SFR0
	ch13	0x4000_51A0	SFR0
	ch14	0x4000_51C0	SFR0
	ch15	0x4000_51E0	SFR0
	ch16	0x4000_5200	SFR0
	ch17	0x4000_5220	SFR0
	ch18	0x4000_5240	SFR0
	ch19	0x4000_5260	SFR0
	ch20	0x4000_5280	SFR0
	ch21	0x4000_52A0	SFR0
	ch22	0x4000_52C0	SFR0
	ch23	0x4000_52E0	SFR0
	ch24	0x4000_5300	SFR0
	ch25	0x4000_5320	SFR0
	ch26	0x4000_5340	SFR0
	ch27	0x4000_5360	SFR0
	ch28	0x4000_5380	SFR0
	ch29	0x4000_53A0	SFR0
	ch30	0x4000_53C0	SFR0
	ch31	0x4000_53E0	SFR0
CAN 控制器 (CAN)	单元 0	0x4000_5400	SFR0
μDMA 控制器 (μDMAC)	单元 A	0x4004_C000	SFR0
	DMAIF	0x4005_F000	SFR0

外设 功能		基 址 址	区 域
输入/输出 端口	端口 A	0x400C_0000	SFR0
	端口 B	0x400C_0100	SFR0
	端口 C	0x400C_0200	SFR0
	端口 D	0x400C_0300	SFR0
	端口 E	0x400C_0400	SFR0
	端口 F	0x400C_0500	SFR0
	端口 G	0x400C_0600	SFR0
	端口 H	0x400C_0700	SFR0
	端口 J	0x400C_0800	SFR0
	端口 K	0x400C_0900	SFR0
	端口 L	0x400C_0A00	SFR0
	端口 N	0x400C_0C00	SFR0
	端口 P	0x400C_0D00	SFR0
16-位 定时器/事件 计数器 (TMRB)	ch0	0x400C_4000	SFR0
	ch1	0x400C_4100	SFR0
	ch2	0x400C_4200	SFR0
	ch3	0x400C_4300	SFR0
	ch4	0x400C_4400	SFR0
	ch5	0x400C_4500	SFR0
	ch6	0x400C_4600	SFR0
	ch7	0x400C_4700	SFR0
	ch8	0x400C_4800	SFR0
	ch9	0x400C_4900	SFR0
串行总线 接口 (I2C)	ch0	0x400E_0000	SFR0
串行 通道 (SIO/UART)	ch0	0x400E_1000	SFR0
	ch1	0x400E_1100	SFR0
	ch2	0x400E_1200	SFR0
	ch3	0x400E_1300	SFR0
振荡频率检测功能 (OFD)	单元 0	0x400F_1000	SFR0
看门狗 定时器 (WDT)	ch0	0x400F_2000	SFR0
时钟/模式 控制 (CG)		0x400F_3000	SFR0
内置高-速振荡调节功能 (TRMOSC)	单元 0	0x400F_3200	SFR0
数字噪声过滤 电路 (DNF)	单元 0	0x400F_3400	SFR0
电压检测功能 (VLTD)	单元 0	0x400F_4000	SFR0
可编程电机驱动器 (PMD)	ch0	0x400F_6000	SFR0
	ch1	0x400F_6100	SFR0
编码器输入电路 (A-ENC)	ch0	0x400F_7000	SFR0
	ch1	0x400F_7100	SFR0
向量 引擎 (A-VE)	ch0	0x400F_8000	SFR0
	ch1	0x400F_8400	SFR0
模拟/数字 转换器 (ADC)	单元 A	0x400F_D000	SFR0
	单元 B	0x400F_D100	SFR0
闪存 / 调试 (FC)	ch0	0x5DFF_0000	SFR0

5. 复位 工作

以下复位 工作源.

- 上电-复位 电路 (POR)
- 电压 检测 电路 (VLTD)
- 复位 引脚 (RESET)
- 看门狗 定时器 (WDT)
- 振荡 频率 电路 (OFD)
- 应用 中断 通过 CPU 和来自复位寄存器位 <SYSRESETREQ>的信号

识别复位源, 检查时钟生成器寄存器中 CGRSTFLG " 异常 " 章节描述.

关于上电复位 电路, 电源检测电路, 看门狗定时器和振荡频率检测 电路的详细, 参考各章节.

通过 <SYSRESETREQ>的复位参考" Cortex-M3 技术参考手册 " .

注:一旦复位工作完成, 内部 RAM 数据不确保.

5.1 冷复位

当上电时,需要内置稳压器,内置闪存和内部高-速振荡器的考虑稳定时间。TMPM475FDFG/FZFG/FYFG有自动插入稳定时间的功能。一旦电压上电复位释放电压范围外,电源计数器开始工作,然后 tPWUP (约 0.9ms fosc=10MHz中) 释放内部复位信。

TMPM475FDFG/FZFG/FYFG 有使能低电压检测电路 (VLTD) 工作功能。tPWUP内,如果供电电压变成大于检测电压 <VDLVL[1:0]>,通过计数器停止工作释放内部复位信号。

上电复位电路工作参考"上电复位电路 (POR) "部分。且低压检测电路工作参考"低压检测电路 (VLTD) "部分。

5.1.1 通过上电-复位电路复位(不使用RESET引脚)

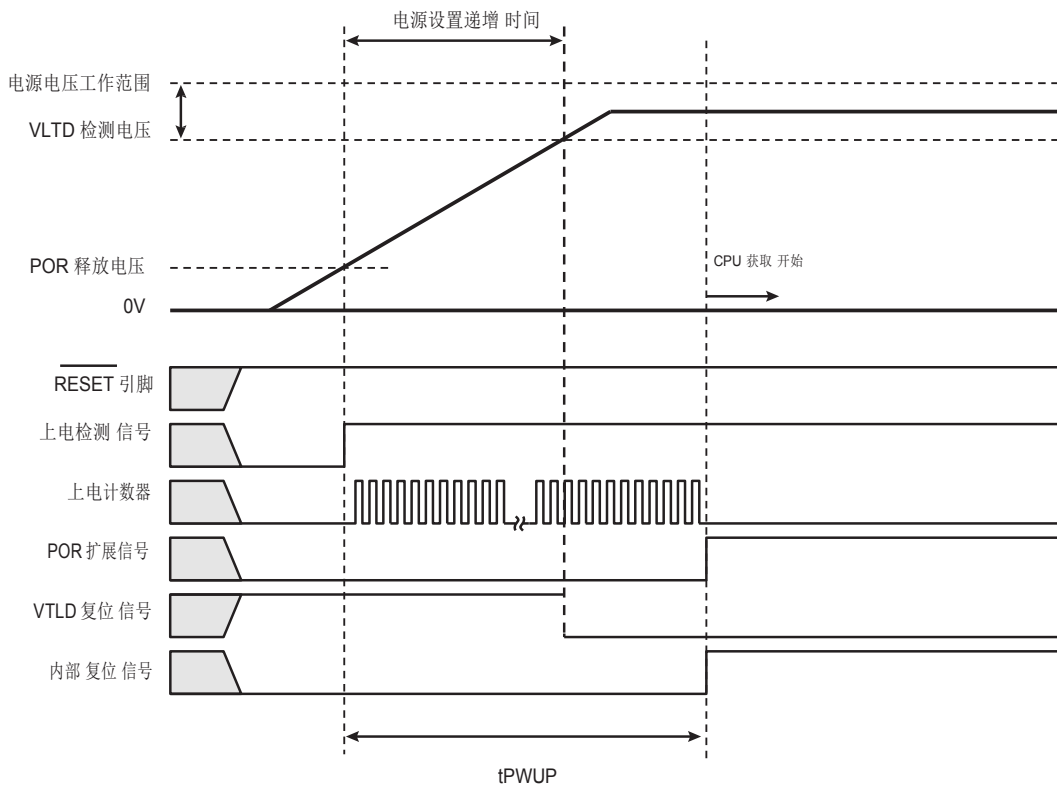


图 5-1 通过上电-复位电路的复位工作(不使用RESET引脚)

当 POR 扩展信号为"低"电平时,当电源电压上升超过 VLTD 检测电压,POR 扩展信号变"高"电平且这导致内部复位信号为"高"电平。然后,复位状态被释放。

5.1.2 通过低-压-检测电路复位(不使用RESET引脚)

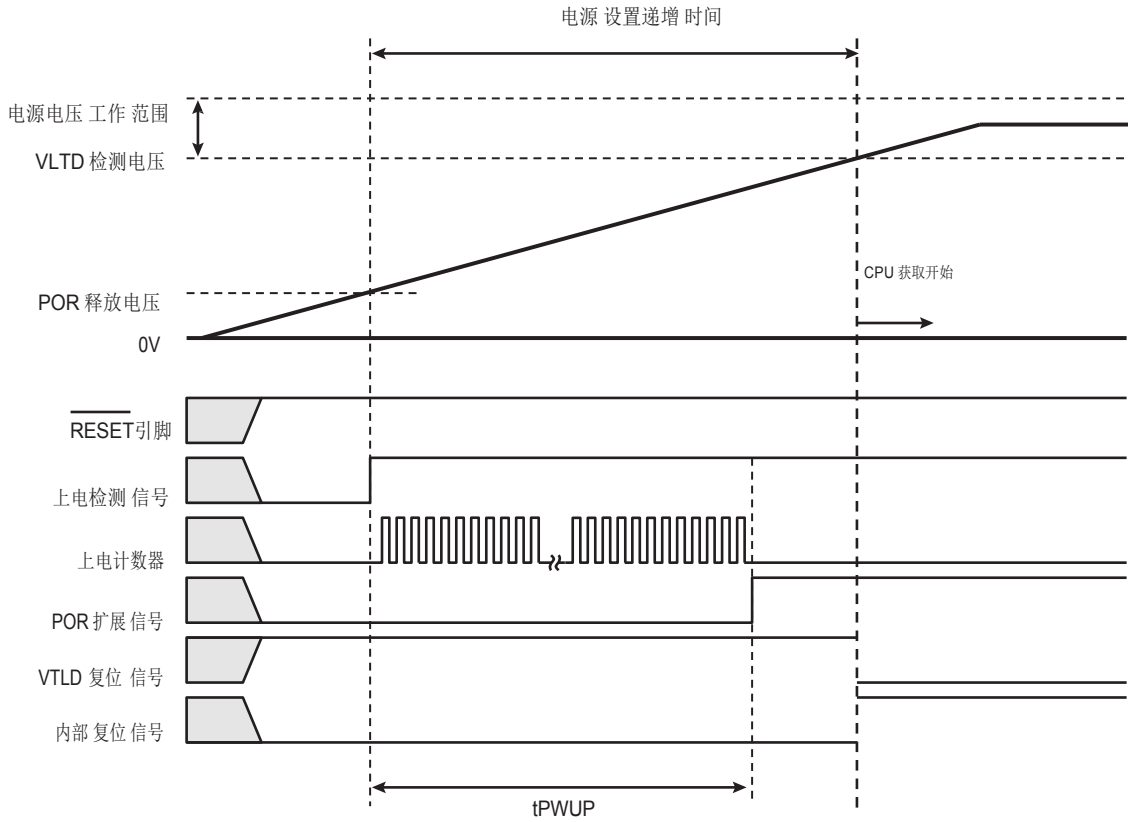


图 5-2 通过低-压-检测电路的复位工作(不使用RESET引脚)

当POR 扩展信号为 "高" 电平时, 当电源电压上升超过VLTD 检测电压, 当VLTD 复位信号变 "低" 电平且这导致内部复位信号为 "高" 电平。然后, 复位状态被释放。

5.1.3 通过 $\overline{\text{RESET}}$ 引脚复位(POR 复位工作有效.)

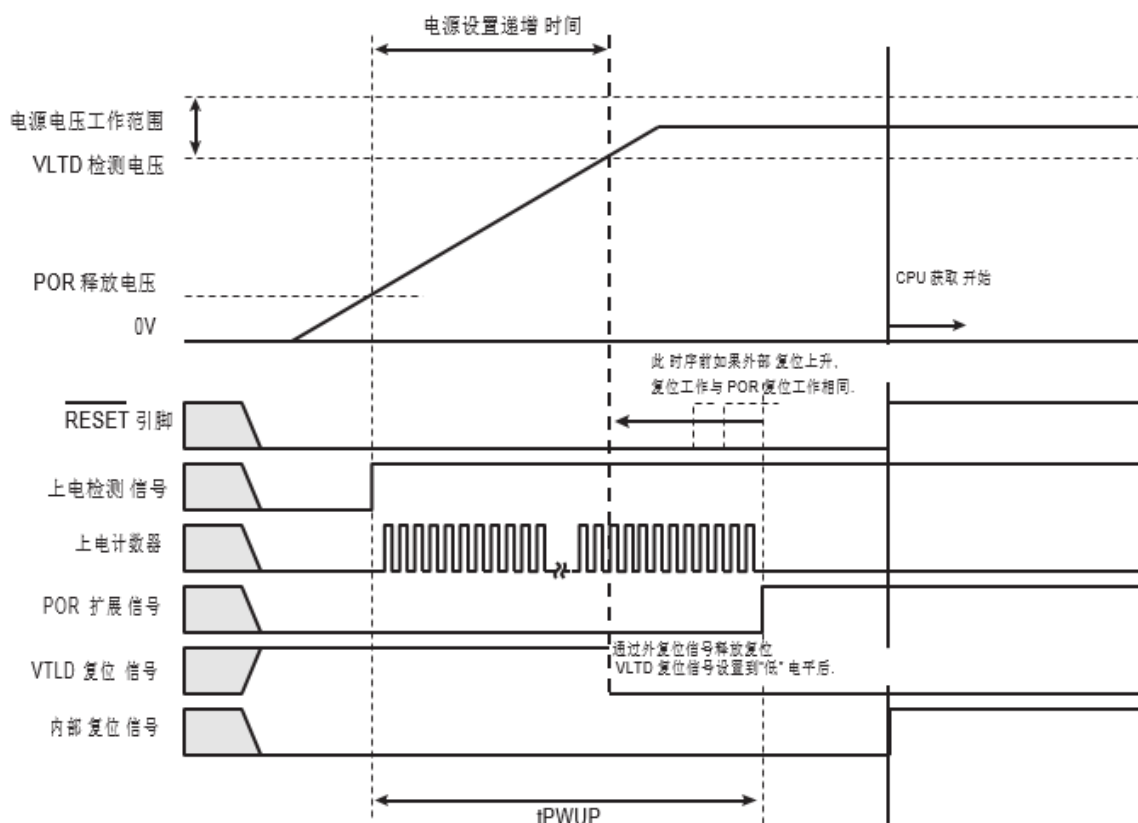


图 5-3 通过复位引脚 $\overline{\text{RESET}}$ 工作(POR 复位工作有效.)

当 POR 复位工作有效时, 当 POR 扩展信号已经变"高" 电平后, $\overline{\text{RESET}}$ 引脚变"高" 电平, 这导致内部复位信号为"高"。然后, 复位状态被释放。

如果 POR 扩展信号变"高"电平之前, VLTD-复位-信号已经变"低"电平后, $\overline{\text{RESET}}$ 信号立即变"高"电平, 复位工作将为 5.1.1 POR 复位工作描述中所述相同工作。

注意只要 VLTD 复位信号为 "高" , $\overline{\text{RESET}}$ 信号应该为 "低" (如: 电源电压比工作电压低)。

5.1.4 通过RESET引脚的复位(VLTD复位工作有效.)

有效是在使用RESET引脚复位上电计数结束后。复位处理与5.1.1上电描述同，若RESET引脚设为"高"，且tPWUP内上电复位信号变"高"后。

TMPM475FDFG/FZFG/FYFG 有一个可以去使能低压检测电路 (VLTD) 工作的功能。在RESET引脚被设为"高"之前, 如果电源电压变为在检测电压 <VDLVL[1:0]>之上的话, 内部的复位信号将会被释放，而且在RESET引脚设为"低"后。

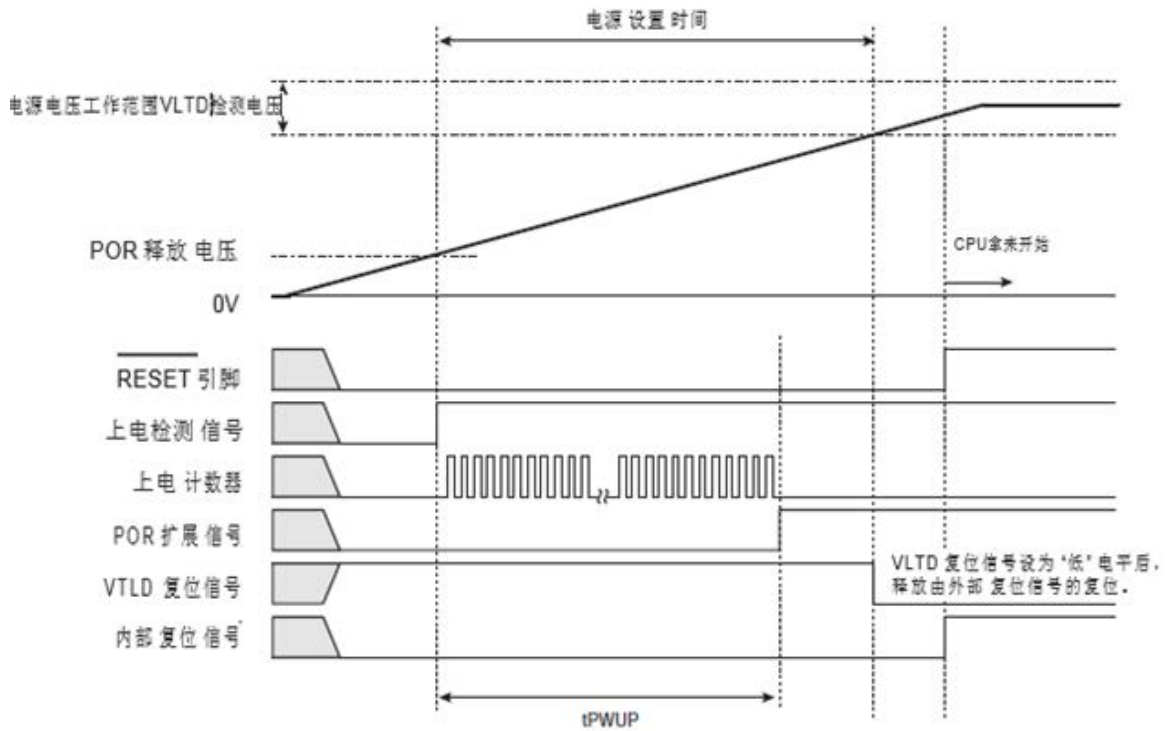


图 5-4 RESET 引脚复位工作 (VLTD 复位工作有效.)

当 VLTD复位工作有效, VLTD 复位信号已经变为"低" 电平后，当RESET引脚和内部复位信号变 "高" 电平, 复位状态被释放。

注意只要 VLTD 复位信号为 "高" ， RESET信号应该为 "低" (如：电源电压比工作电压低)。

5.2 预热

5.2.1 复位 期间

为了去做复位 TMPM475FDFG/FZFG/FYFG, 需要有以下条件; 电源电压必须在工作范围以内; 通过已经稳定的内部高频率振荡器, RESET引脚保持为 "低" 且至少需要 12 个系统时钟 (当 $f_{osc} = 10\text{MHz}$ 的时候, $1.2\ \mu\text{sec}$ 的最小期间)。内部的复位会被释放, 在 $\overline{\text{RESET}}$ 引脚变"高"以后。

5.3 复位后

复位后, 处理器核和外围功能控制寄存器 (SFR) 的控制寄存器几乎初始化。内部核的系统调试组成寄存器 (FPB, DWT, 和 ITM), 时钟生成器的CGRSTFLG和闪存相关寄存器 的FCSECBIT只通过冷复位被初始化。

当复位释放, MCU通过内部高-速 振荡器时钟开始工作。如果需要, 外部时钟和PLL 乘法电路应该被设置。

6. 时钟 / 模式 控制

6.1 特征

时钟/模式控制电路控制内部/外部振荡器, 时钟齿轮, 预分频时钟和 PLL 时钟乘法电路预热和振荡器.

也有通过模式过渡可以减少功耗的低功耗模式.

此章节描述如何控制时钟工作模式和模式过渡.

6.2 寄存器

6.2.1 寄存器 表

下表 寄存器和其时钟 控制相关地址.

关于基 地址, 参考 "内存映射"的" 外设功能基地址表"章节

外设 功能 :CG

寄存器 名		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
外设时钟 停止 寄存器	CGCKSTP	0x0040

6.2.2 CGSYSCR (系统控制寄存器)

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

位	位符	类型	功能
31-18	-	R	读作 "0".
17-16	-	R/W	写作 "01".
15-13	-	R	读作 "0".
12	FPSEL	R/W	选择 fperiph 源时钟 0: fgear 1: fc 指定源时钟 到 fperiph. 无论时钟齿轮模式, 选择 fc 固定 fperiph.
11	-	R	读作 "0".
10- 8	PRCK[2:0]	R/W	分频器 时钟 ($\phi T0$) 分割比例选择 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: 保留 011: fperiph/8 111: 保留 给提供到外围功能的分频器时钟($\phi T0$) 选择分割比例。 当外围功能工作时, 不要变化分频器时钟的分割比例。注意 当分频器 时钟的分割比例变化时, $\phi T0$ 的频率必须低于 fsys 的频率。
7-3	-	R	读作 "0".
2-0	GEAR[2:0]	R/W	时钟 齿轮 (fgear)分割比例选择 000: fc 100: fc/2 001: 保留 101: fc/4 010: 保留 110: fc/8 011: 保留 111: fc/16 选择时钟齿轮的分割比例。 当外围功能工作时, 不要变化时钟 齿轮的分割比例。注意 当时钟齿轮功能使用时, $\phi T0$ 的频率必须低于 fsys 的频率。

6.2.3 CGOSCCR (振荡 控制 寄存器)

	31	30	29	28	27	26	25	24
位 符	WUODR							
复位后	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	WUODR				WUPSEL2	-	OSCSEL	XEN2
复位后	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	XEN1
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-20	WUODR[11:0]	R/W	计数器比较高-速 振荡器预热定时器的设置值 设置预热定时器的计数器的比较值。 设置预热时间的 16 位 计算值的上 12 位值。
19	WUPSEL2	R/W	选择高-速 振荡器预热定时器源 时钟 (注 1) 0: 内部 (fosc) 1: 外部 (feosc)
18	-	R/W	保留 (即使"0" 或 "1" 写入,工作也没问题。)
17	OSCSEL	R/W	高-速 振荡器 (fosc) 选择 0: 内部 (fosc) 1: 外部 (feosc) 选择高-速 振荡器源时钟时, 寄存器内容更新, 确认是否目的地时钟稳定振荡。 另外, 如果源 时钟改变, 检查是否改变的值反映到 <OSCSEL>.
16	XEN2	R/W	内部 高-速 振荡器 (IOSC) 控制 0: 禁止 1: 使能 控制内部 高-速 振荡器 (IOSC). 此位使能时, 需要使用预热计数器的振荡稳定时间. 详细, 参考 "6.3.5 预热 功能".
15-12	-	R/W	写作 "0".
11-10	-	R	读作 "0".
9	-	R/W	写作 "0".
8	XEN1	R/W	外部 高-速 振荡器 (EOSC) 控制 0: 禁止 1: 使能 控制外部 高-速 振荡器。 此位使能时, 需要使用预热计数器振荡稳定时间. 详细, 参考 "6.3.5 预热 功能".
7-4	-	R/W	读作 "0".
3	WUPSEL1	R/W	预热 定时器时钟 源 写作 "0".
2	PLLON	R/W	PLL 工作 控制 0: 停止 1: 振荡 控制 PLL 工作。 此位使能时, 需要使用预热计数器的振荡稳定时间. 详细, 参考 "6.3.5 预热 功能".

位	位符	类型	功能
1	WUEF	R	高-振荡器预热定时器状态 0: 预热完成. 1: 预热工作 使能监视预热定时器状态.
0	WUEON	W	高-速 振荡器预热定时器控制 1: 工作 开始. 预热定时器 工作开始 前(设置 "1"), 检查是否 <WUEF> 为"0" . 写入 "0" 到 <WUON> 无意义 和读"0".

注:改变到 STOP 模式前, 通过 <OSCSEL>.设置相同时钟选择。

6.2.4 CGSTBYCR (待机 控制 寄存器)

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

位	位 符	类型	功能
31-18	-	R	读作 "0".
17	-	R/W	写作 "0".
16-10	-	R	读作 "0".
9	-	R/W	写作 "0".
8	RXEN	R/W	释放 STOP 模式后, 高-速 振荡器 工作。 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: STOP 010: 保留 011: IDLE 100: 保留 101: 保留 110: 保留 111: 保留 为了进入 STOP 模式, 禁止未被使用作系统时钟的振荡 (IOSC 或 EOSC) .

6.2.5 CGPLLSEL (PLL 选择 寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-1	PLLSET	R/W	PLL 乘法 值 (除以下外不用) 0x591E: 输入 时钟 10MHz, 输出 时钟 80MHz 0x5926: 输入 时钟 10MHz, 输出 时钟 100MHz 0x59AE: 输入 时钟 10MHz, 输出 时钟 120MHz
0	PLLSEL	R/W	选择高-速 时钟 (fc) 源 时钟 0: fosc 用 1: fPLL 用 此 被用来选择高-速 时钟 (fc) 源 时钟.

6.2.6 CGCKSTP (外设时钟停止 寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2	CANSTP	R/W	CAN 时钟 控制 0 : 工作 1 : 时钟 停止 使能 / 禁止 CAN 时钟
1-0	-	R	读作 "0".

注:从 "0"设置 CGCKLSTP<CANSTP> 到 "1"后, 确认停止电路工作.

6.3 时钟控制

6.3.1 时钟类型

以下定义各时钟：

f_{EOSC}	: 由外部高速振荡器 (EOSC) 时钟输入
f_{IOSC}	: 由内部高速振荡器 (IOSC) 时钟输入
f_{OSC}	: 由高速振荡器时钟输入
f_{PLL}	: PLL 电路 (x8, x10, x12) 时钟输入
f_{C}	: 由 f_{OSC} 或 f_{PLL} (高速时钟) 时钟选择
f_{gear}	: 由高速时钟时钟分割
f_{SYS}	: 相同时钟为 f_{gear} (系统时钟)
f_{periph}	: 由 CGSYSCR<FPSEL>指定的时钟
$\phi T0$: 由 CGSYSCR<PRCK[2:0]> 指定的时钟 (预分频时钟)

高速时钟 f_{C} 和预分频时钟 $\phi T0$ 可做以下分割.

高速时钟	: $f_{\text{C}}, f_{\text{C}}/2, f_{\text{C}}/4, f_{\text{C}}/8, f_{\text{C}}/16$
预分频时钟	: $f_{\text{periph}}, f_{\text{periph}}/2, f_{\text{periph}}/4, f_{\text{periph}}/8, f_{\text{periph}}/16, f_{\text{periph}}/32$

6.3.2 复位后初始值

以下复位工作初始化时钟配置.

高速振荡器 (EOSC)	: 停止
高速振荡器 (IOSC)	: 振荡相
锁循环电路 (PLL)	: 停止
高速时钟齿轮	: f_{C} (无频率分割)

复位工作引起所有时钟配置与 f_{OSC} 相同.

$$f_{\text{C}} = f_{\text{OSC}}$$

$$f_{\text{SYS}} = f_{\text{C}} (= f_{\text{OSC}})$$

$$f_{\text{periph}} = f_{\text{C}} (= f_{\text{OSC}})$$

$$\phi T0 = f_{\text{periph}} (= f_{\text{OSC}})$$

6.3.3 时钟 系统 图

图 6-1 显示时钟系统图。

复位后, 在提供到选择器的输入时钟中箭头指的时钟被选择。

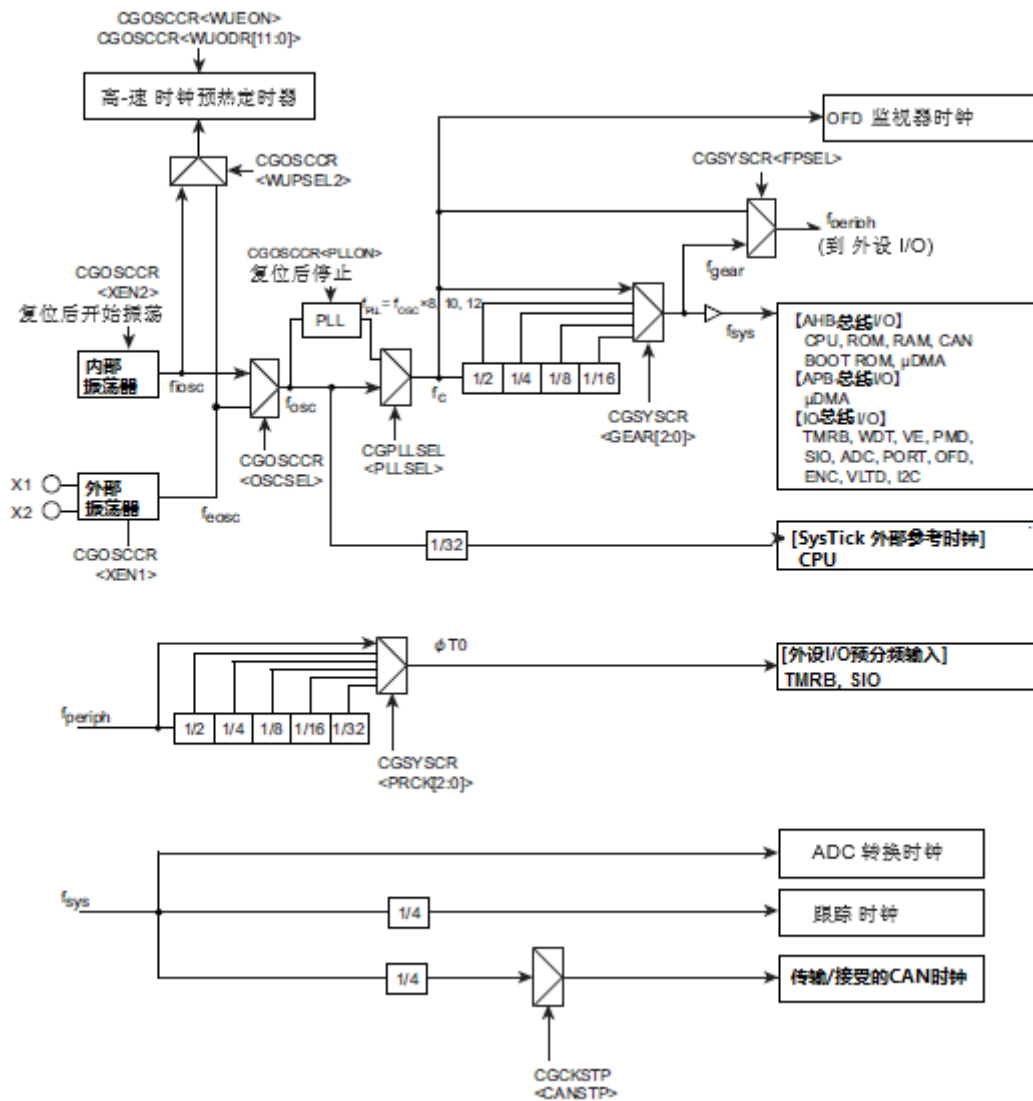


图 6-1 时钟 方块图

6.3.4 时钟乘法电路 (PLL)

此电路输出高-速振荡器输出时钟 (f_{osc})的乘法($\times 8$, $\times 10$, $\times 16$) f_{PLL} 时钟。因此, 振荡器输入频率可以为低, 且内部时钟为高-速。

PLL复位后禁用。为了使能PLL, 设置 "1" 到 CGOSCCR<PLLON> 位且设置 "1" 到 CGPLLSEL<PLLSEL>。然后 f_{PLL} 时钟 输出为高-速振荡器(f_{osc})的乘法 ($\times 8$, $\times 10$, $\times 16$)。

PLL 需要一定时间的总数来稳定, 它应该确保使用预热功能或别的方法。

6.3.4.1 稳定时间

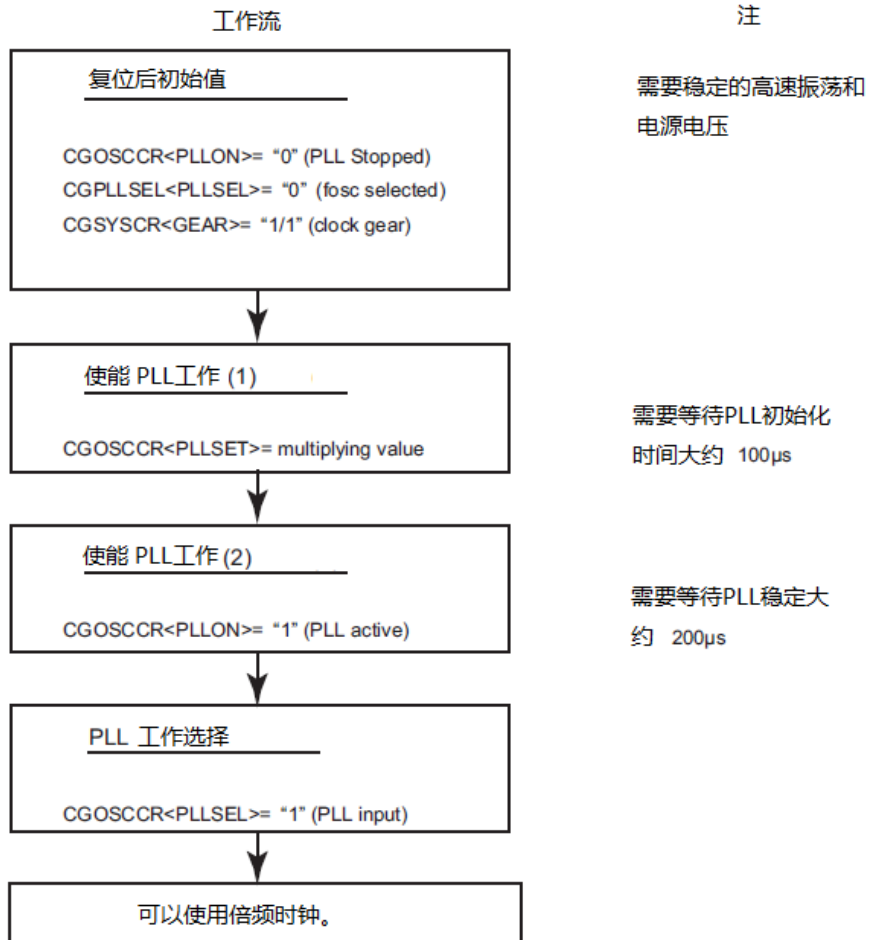
PLL 需要一定时间的总数来稳定, 它应该确保使用预热功能或别的方法。

当 <PLLON> 设为 "1" 且工作开始, 它需要花大约 200 μ s 作为Lock-up 时间。

<PLLON> 先作为 "0" 当乘法值变化和 PLL停止时。当乘法 <PLLSEL> 值变化时, <PLLON> 设为 "1"约100 μ s PLL初始化时间过去后, 然后PLL状态开始。今后, 请确保 Lock-up 时间为 PLL 稳定性时间。

6.3.4.2 PLL 设置顺序

以下表示复位后 PLL 设置 顺序.



6.3.5 预热 功能

预热功能 确保振荡器和 PLL 用预热定时器稳定时间.从 STOP 模式返回时预热功能 被使用. 关于详细功能, 在"6.6.6 预热"描述.

注:不要转移到 STOP 模式, 工作预热定时器期间.

此时,从低功耗模式返回的中断触发自动定时器计数。到达指定时间后, 系统时钟输出和 CPU 开始工作。

如何 配置预热功能.

1. 指定 计数 递增 时钟

指定计数递增时钟 CGOSCCR<WUPSEL1> 和 <WUPSEL2>位中预热计数器. 写 "0" 到 <WUPSEL1> 和 写 "0" 或 "1" 到 <WUPSEL2>. "0" 指定内部振荡器和 "1" 指定外部振荡器.

2. 指定 预热 计数器 值

预热时间可以通过设置 CGOSCCR<WUODR[11:0]> 选择。因此,下面等式的计算值低 4 位被四舍五入。上 12 位设置到 CGxWUHCR<WUPT[11:0]>。

以下表示预热设置和 例.

$$\text{预热周期} = \frac{\text{预热时间设置值}}{\text{输入周期 由频率(s)}}$$

<例 1>用 8MHz 振荡器设置 5 ms 预热 时间

$$\frac{\text{预热时间设置值}}{\text{输入周期 由频率(s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40,000 \text{ 周期} = 0x9C40$$

丢掉最后 4 位, 设置 0x9C4 到 CGOSCCR<WUPT[11:0]>.

3. 确认预热开始和完成

通过 软件 (指令)CGOSCCR<WUEON><WUEF> 被用来确认预热开始和完成.

注: 预热定时器根据振荡 时钟工作, 如果振荡 频率中有误动作也许包含错误。 因此, 预热时间应该 作为大约时间。

以下表示预热设置.

<例> 确保 PLL 稳定 时间($fc = feosc$)

CGOSCCR<WUPSEL1> = "0"	: 写 "0" 到 CGOSCCR<WUPSEL1>
CGOSCCR<WUPSEL2> = "1"	: 指定预热定时器时钟 源 (1: 外部 (feosc))
CGOSCCR<WUODR[11:0]> = "0x9C4"	: 预热时间设置
参考 6.3.6 内部 振荡器到外部 振荡器开关处理.	
CGOSCCR<WUEON>="1"	: 使能 预热计数 (WUP)
读 CGOSCCR<WUEF>	: 等待 到状态 变为 "0" (预热完成)

6.3.6 系统时钟

TMPM475FDFG/FZFG/FYFG 提供高-速时钟作系统时钟. 系统时钟从内部振荡器或外部振荡器可选. 复位后, 内部振荡器使能和外部振荡器禁止. 高-速时钟可分割.

- 由 X1 和 X2 输入 频率: 10MHz
- 内部 振荡器 频率: 10MHz
- 时钟 齿轮: 1/1, 1/2, 1/4, 1/8, 1/16 (复位后: 1/1)

表 6-1 高-速 频率范围(单元: MHz)

输入 freq. feosc fiosc	倍	Min 工作 freq.	Max 工作 freq.	复位后 (PLL = OFF, CG = 1/1)	时钟 齿轮 (CG): PLL = ON					时钟 齿轮 (CG): PLL = OFF				
					1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
10	8	1.25	80	10	80	40	20	10	5	10	5	2.5	1.25	-
10	10	1.25	100	10	100	50	25	12.5	6.25	10	5	2.5	1.25	-
10	12	1.25	120	10	120	60	30	15	7.5	10	5	2.5	1.25	-

注 1: PLL=ON/OFF 设置: CGOSCCR<PLLON>中可用.

注 2: 执行时钟 齿轮切换时, 值写入 CGSYSCR<GEAR[2:0]> 寄存器. 轻微延迟后实际切换发生.

注 3: PLL OFF 时, 不使用 1/16.

注 4: "-": 保留

注 5: SysTick 被使用时, 不使用 1/16.

以下为从内部 振荡器 到外部 振荡器切换的处理.

- CGOSCCR<WUODR[11:0]> = "预热 时间" : 设置 预热时间.
- CGOSCCR<XEN1> = "1" : 使能外部 振荡器.
- CGOSCCR<WUPSEL2> = "1" : 指定 外部 振荡器 时钟 为预热-计数器源 时钟
- SCCR<WUEON>="1" : 使能预热 计数 (WUP)
读 CGOSCCR<WUEF> : 等待 到状态 变为 "0" (预热 完成)
- CGOSCCR<OSCSEL> = "1" : 切换 系统 时钟到外部 振荡器.
- 读 CGOSCCR<OSCSEL> : 确认 CGOSCCR[17]<OSCSEL> 变为 "1".
(外部 振荡器 选择.)
- CGOSCCR<XEN2> = "0" : 内部 振荡器 禁止.

6.3.7 预分频 时钟 控制

各外围功能有分割时钟的预分频器。输入到各分频器的时钟 $\phi T0$, CGSYSCR<FPSEL>中指定的 "fperiph" 时钟可以根据 CGSYSCR<PRCK[2:0]>中的设置分割。控制器复位后, fperiph/1 被选择为 $\phi T0$ 。

注: 为使用时钟齿轮, 确保从各外围 功能的分频器输出 ϕTn 时间设置慢于 f_{sys} ($\phi Tn < f_{sys}$)。不要切换时钟齿轮, 当定时器计数器或 别的外围功能工作时。

6.4 模式和模式 过渡

6.4.1 模式 过渡

NORMAL 模式使用高-速时钟给系统时钟。

IDLE 和STOP模式可以用作通过停止处理器核工作达到减少功耗的低功耗模式。

图 6-2 显示模式过渡图。

关于 WFI 指令或 Sleep-on-exit 详细, 参考 ARM 公司发布的文档。

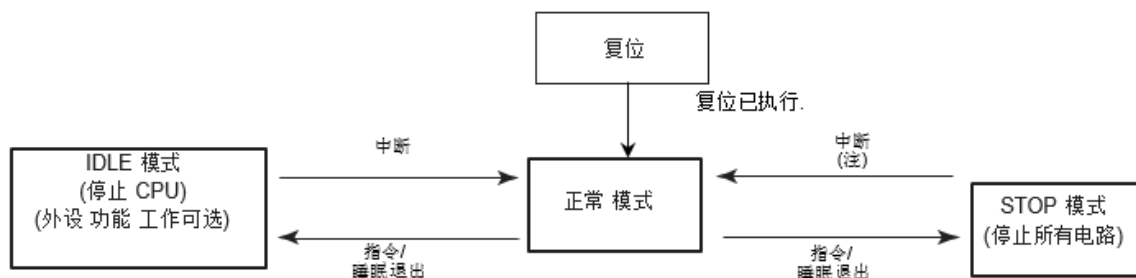


图 6-2 模式 过渡 图

注: 预热需要。预热时间必须在变为 STOP 模式前, 在 NORMAL 模式中设置。关于预热时间, 参考 "6.6.6 预热"。

6.5 工作 模式

6.5.1 正常 模式

此模式是通过使用高-速时钟来工作 CPU 核和外围硬件。

复位后换为 NORMAL 模式。

6.6 低功耗 模式

TMPM475FDFG/FZFG/FYFG 有两低功耗模式: IDLE和STOP。为转换低功耗模式, 在系统控制寄存器 CGSTBYCR<STBY[2:0]> 中规定模式且执行WFI (等待中断) 指令。此时, 执行复位或生成中断去释放模式。通过中断释放需要事先设置。详见章节 "异常"。

注 1: TMPM475FDFG/FZFG/FYFG 不提供任何释放低功耗模式的事件。禁止通过执行WFE (等待事件) 指令过渡到低功耗模式。

注 2: TMPM475FDFG/FZFG/FYFG 不支持Cortex-M3 核中用 SLEEPDEEP 位配置的低功耗模式。禁止设置系统控制寄存器 <SLEEPDEEP> 位。

注 3: 不要转换低功耗模式, 预热定时器工作期间。

以下描述各模式特征。

6.6.1 IDLE 模式

此模式下仅CPU 停止。IDLE 模式下, 各外围功能在控制寄存器中有使能或禁用工作的一位。当进入 IDLE 模式, IDLE 模式下, 禁用外围功能的工作停止工作且此时保持状态。

IDLE 模式下, 如下外围功能可以使能或禁用。关于设置详细, 见各外围功能章节。

- 16-位 定时器 / 事件 计数器 (TMRB)
- 串行 通道 (SIO/UART)
- 串行 总线 接口 (I2C)
- 向量 引擎 (A-VE)

6.6.2 STOP 模式

STOP模式下所有内部电路包括内部振荡器被停止。

通过释放 STOP 模式, 设备返回先前 STOP 模式的模式并开始工作。

表 6-2 表示 STOP 模式下引脚 状态 .

表 6-2 STOP 模式下引脚 状态

功能	引脚 名	I/O	STOP (注)
控制	RESET, MODE	输入	使能
振荡器	X1	输入	禁止
	X2	输出	"高" 电平 输出
端口	PAx 到 PPx	输入	取决于 PxIE[m]
		输出	取决于 PxCR[m]
调试	SWCLK, SWDIO TRST, TCK, TMS, TDI	输入	取决于 PxIE[m]
	SWDIO, SWV TDO, TRACECLK TRACEDATA0/1/2/3	输出	取决于 PxCR[m] 且使能 数据有效时
中断	INT0 到 INTF	输入	取决于 PxIE[m]
PMD	UO0, VO0, WO0, XO0, YO0, ZO0 UO1, VO1, WO1, XO1, YO1, ZO1	输出	取决于 PxCR[m]
以上 除外	以上除外	输入	取决于 PxIE[m]
		输出	取决于 PxCR[m]

(注) x: 端口 数 / m: 对应 位

6.6.3 低功耗模式 设置

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

表 6-3 表示<STBY[2:0]>中模式设置。

表 6-3 低功耗模式 设置

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

注:不要设置上示<STBY[2:0]>中以外任何值。

6.6.4 各模式下工作状态

表 6-4 表示各模式下工作状态。

表 6-4 各模式下工作状态

方块	正常	IDLE	STOP
处理器核	o	-	-
μDMAC	o	o	-
I/O 端口	o	o	- (注 1)
PMD	o	o	-
A-ENC	o	o	-
OFD	o	o	Δ
ADC	o	o	-
CAN	o	o	-
A-VE	o	*	-
SIO/UART	o	*	-
I2C	o	*	-
TMRB	o	*	-
WDT	o	Δ	Δ
VLTD	o	o	o (注 2)
POR	o	o	o (注 2)
DNF	o	o	-
CG	o	o	-
PLL	o	o	Δ
高速 振荡器 (fc)	o	o	-

o: 目标模式下可以工作。

*: 目标模式下可以工作(每个通道工作 / 停止可以选择)

-: 当过渡到目标模式时, 调制时钟自动停止。

Δ: 在目标模式下, 通过软件可以选择使能或禁用模组工作。

注 1: 取决于端口寄存器。

注 2: 即使时钟停止方块不停止。

6.6.5 释放低功耗 模式

低功耗模式可以由中断请求或复位释放。可以使用的释放源由选择低功耗模式决定。详细如表 6-5 所示。

表 6-5 各模式下释放 源

低功耗 模式		IDLE	STOP	
释放 源	中断	INT0 到 INTF (注 1)	o	o
		INTENC00, INTENC01 INTENC10, INTENC11	o	x
		INTVCN0, INTVCT0 INTVCN1, INTVCT1	o	x
		INTPWM0, INTEMG0, 到 VV0 INTPWM1, INTEMG1, 到 VV1	o	x
		INTRX0 到 3, INTTX0 到 3	o	x
		INTADACPA, INTADBCPA INTADACPB, INTADBCPB INTADATMR, INTADBTRM	o	x
		INTADASFT, INTADBSFT INTADAPDA, INTADBPDA INTADAPDB, INTADBPDB	o	x
		INTTB00 到 INTTB90 INTCAP00 到 INTCAP90 INTTB01 到 INTTB91 INTTCP01 到 INTCAP91	o	x
		INTDMACATC, INTDMACAERR	o	x
		INTCANRX, INTCANTX, INTCANGB	o	x
	SysTick	o	x	
	RESET (OFD)	o	x	
	RESET ($\overline{\text{RESET}}$ 引脚, POR, VLTD)	o	o	

o : 释放模式后, 开始中断处理. (复位初始化 LSI)

x : 不可用

注 1: 通过使用电平模式中断, 为了释放低功耗模式, 保持电平直到中断处理开始。此前变化电平可以从开始适当地防止中断处理。

注 2: 为转换低功耗模式, 设置 CPU 去禁止所有释放源以外的中断。否则, 释放可以通过未指定唤醒执行。

注 3: 关于预热时间参考"6.6.6 预热"。

- 通过中断 请求释放

为通过中断释放低功耗 模式, CPU 必须 事先设置去检测中断。CPU 设置以外, 必须设置时钟生成器去检测要用的中断去释放 STOP 模式。数字噪声过滤器电路也应该设置为禁用。

- 通过 复位释放

任何低功耗模式应该全部都可以来达到被释放, 也就是通过 $\overline{\text{RESET}}$ 引脚, POR 和 VLTD。

IDLE 模式可以通过复位 OFD 释放。

然后, 模式切换为 NORMAL 模式且所有寄存器用正常复位被初始化。

- 通过 SysTick 中断释放

IDLE 模式下, SysTick 中断可以被使用。

详细参考 "中断".

6.6.6 预热

模式过渡需要预热为了内部振荡器提供稳定振荡。

从停止到NORMAL模式过渡, 预热计数器自动激活。然后配置的预热时间过去后, 系统时钟输出开始。需要在CGOSCCR<WUPSEL1><WUPSEL2> (注1)设置预热要用的振荡器且执行进入STOP模式指令前, 在CGOSCCR<WUODR> (注2)设置预热时间。

注 1: 始终设置CGOSCCR<WUPSEL1> 为 "0"。

注 2: STOP模式下,PLL 禁用。当从这些模式返回时, 配置预热时间以考虑PLL 和内部振荡器的稳定性时间。大约需要 200 μ s PLL才能稳定化。

注 3: 不要写 "1" 到 CGOSCCR<WUEON> 位, 用自动预热从低耗模式返回的设置时。

表 6-6 表示各模式过渡是否需要预热设置。

表 6-6 模式 过渡下预热设置

模式 过渡	预热 设置
正常 → IDLE	不 需要
正常 → STOP	不 需要
IDLE → 正常	不 需要
STOP → 正常	自动-预热

注:当执行通过复位释放时, 自动预热不执行。 输入复位直到振荡器变稳定。

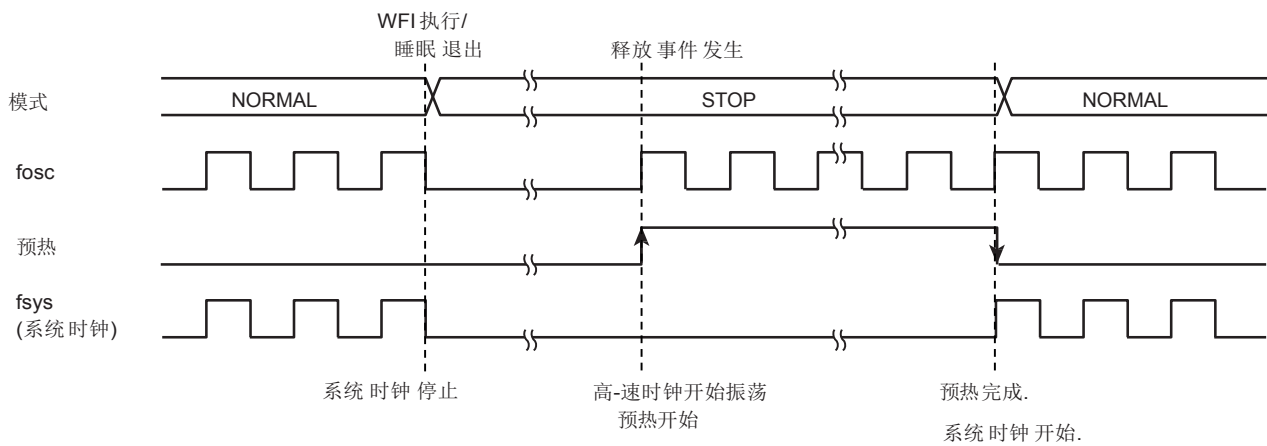
6.6.7 模式切换时的时钟操作

章节 6.6.7.1.描述模式过渡下时钟工作。

6.6.7.1 工作模式的切换：NORMAL → STOP → NORMAL

当从 STOP 模式返回NORMAL 模式时, 预热自动激活。进入STOP模式前, 需要设置预热时间。

通过复位返回 NORMAL 模式不引起自动预热。 保持这个复位信号直到振荡器工作变稳定。



7. 内部高-速振荡调节 功能

TMPM475FDFG/FZFG/FYFG 有内部高-速振荡调节功能.

7.1 架构

内部振荡调节功能 使用 16-位 定时器/事件 计数器(TMRB)的脉宽测量功能.

图 7-1 表示功能 配置.

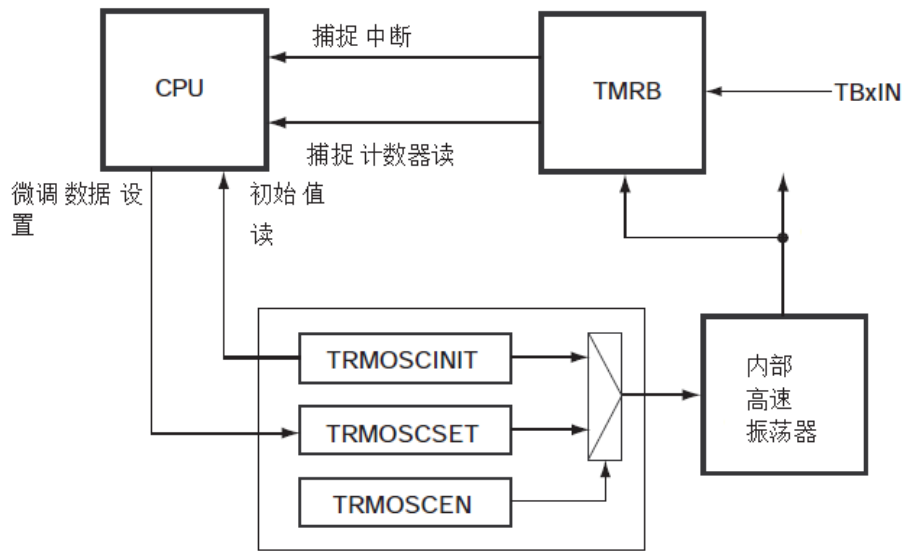


图 7-1 功能 方块 图

7.2 寄存器

7.2.1 寄存器 表

下表表示控制寄存器和其地址.

基 地址的详细, 参考"内存映射"的"外设功能地址表"章节 .

寄存器 名		地址(基+)
保护 寄存器	TRMOSCPRO	0x0000
使能 寄存器	TRMOSCEN	0x0004
初始 微调值监视器寄存器	TRMOSCINIT	0x0008
微调 值设置 寄存器	TRMOSCSET	0x000C

7.2.2 TRMOSCPRO (保护 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7-0	PROTECT [7:0]	R/W	写入 寄存器 控制 0xC1 : 使能 0xC1 以外: 禁止 "0xC1" 设置时, 允许写 TRMOSCxEN, TRMOSCxINIT 和 TRMOSCx 设置.

7.2.3 TRMOSCEN (使能 寄存器)

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

位	位 符	类型	功能
31-1	-	R	读作 "0".
0	TRIMEN	R/W	微调 控制 0: 禁用 1: 使能 当 "1" 被设置, 内部 振荡器的微调值从 TRIMOSCxINIT 的值切换到 TRMOSCxSET 的值。

7.2.4 TRMOSCINIT (初始微调值 监视器 寄存器)

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

位	位符	类型	功能
31-14	-	R	读作 "0".
13-8	TRMINITC [5:0]	R	初始粗微调值 使能监视器 初始粗微调值.
7-4	-	R	读作 "0".
3-0	TRMINITF[3:0]	R	初始 细微调值 使能监视器 初始细微调值.

关于指定设置和粗微调 and 细微调调节值的详细, 参考"表 7-1 调节 范围".

7.2.5 TRMOSCSET (微调值 设置 寄存器)

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

位	位 符	类型	功能
31-14	-	R	读作 "0".
13-8	TRIMSETC [5:0]	R/W	粗 微调值设置设置粗微调值.
7-4	-	R	读作 "0".
3-0	TRIMSETF[2:0]	R/W	细微调值设置设置细微调值.

关于指定设置和粗微调 和细 微调调节值的详细, 参考"表 7-1 调节 范围".

7.3 工作描述

7.3.1 概要

振荡使用粗微调值和细微调值调节。

运送前的设置值可以用TRMOSCINIT<TRIMINITC> 和 <TRIMINITF>检查。当值变化时, 设置新值到 TRMOSCSET<TRIMSETC> 和 <TRIMSETF>。通过设置 "1" 到 TRMOSCEN<TRIMEN>, 内部振荡器设置值会变化。

注:复位后, 禁止写到 TRMOSCSET 和 TRMOSCEN。当写到这些位, TRMOSCPRO<PROTECT> 必须设为 "0xC1"。

7.3.2 调节范围

粗微调中, 约 -19% 到 +32% 调节通过平均0.8%步可行。

细微调中, -0.8% 到 +0.7% 调节通过 0.1%-步可行。

表 7-1 显示调节范围。

注 1: 每个步值基于典型条件预估。粗微调中, 有 $\pm 0.6\%$ 误差余量。细微调中, 有 $\pm 0.1\%$ 误差余量。

注 2: 每个微调分辨率根据样本或条件有一些种类。

表 7-1 调节范围

粗微调	
<TRIMSETC[5:0]>	频率改变 (typ.)
011111	+32.0%
011110	+30.6%
011101	+29.2%
011100	+27.8%
011011	+26.5%
-	-
000011	+2.3%
000010	+1.5%
000001	+0.8%
000000	$\pm 0\%$
111111	-0.7%
111110	-1.5%
-	-
100100	-16.9%
100011	-17.4%
100010	-17.9%
100001	-18.3%
100000	-18.8%

细微调	
<TRIMSETF[3:0]>	频率改变 (typ.)
0111	+0.7%
-	-
0001	+0.1%
0000	$\pm 0\%$
1111	-0.1%
1110	-0.2%
-	-
1000	-0.8%

8. 异常

此章节描述异常的特征, 类型和处理.

异常与 CPU 核有密切关系。如果需要, 参考 "ARM 发布的 ARM Cortex-M4F 文档".

8.1 总览

异常导致CPU停止当前执行的过程且处理别的过程。

有两类型异常: 它们在当一些错误条件发生时或当执行异常指令时产生; 它们由硬件产生, 比如从外部引脚或外围功能的中断请求信号。

所有异常由 CPU 中嵌套向量中断控制器 (NVIC)根据各自的优先电平处理。当异常产生,CPU 存储当前状态到堆栈且 扩展到相应中断服务程序 (ISR)。ISR 完成后, 存储到堆栈的信息自动被存储。

8.1.1 异常 类型

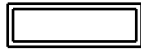
以下为存在于 Cortex-M4F 中的异常类型.

关于每个异常详细描述, 参考 "ARM 发布的 ARM Cortex-M4F 文档".

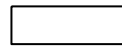
- 复位
- 非-可屏蔽 中断 (NMI)
- 硬 错误
- 映射 管理
- 总线 错误
- 使用 错误
- SVCcall (总监呼叫)
- 调试 监视器
- PendSV
- SysTick
- 外部 中断

8.1.2 处理流程图

以下表示 如何处理 异常/中断. 在以下描述中,



指 硬件处理。



指 软件 处理.

各步在此章节后面描述.

处理	描述	见
<div style="border: 3px double black; padding: 5px; width: fit-content; margin: 0 auto;">检测 通过 CG/CPU</div>	CG/CPU 检测异常 请求.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">部分 8.1.2.1</div>
<div style="border: 3px double black; padding: 5px; width: fit-content; margin: 0 auto;">处理 通过 CPU</div>	CPU 处理异常 请求.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">部分 8.1.2.2</div>
<div style="border: 3px double black; padding: 5px; width: fit-content; margin: 0 auto;">分支到 ISR</div>	CPU 分支到对应 中断 服务 程序 (ISR).	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">部分 8.1.2.3</div>
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">ISR 执行</div>	执行必要 处理.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">部分 8.1.2.4</div>
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">从异常返回</div>	CPU 分支到别的 ISR 或 返回到前面的程序.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">部分 8.1.2.4</div>

注:ISR : 中断 服务 程序

8.1.2.1 异常 请求和检测

(1) 异常 发生

异常源包括CPU执行的指令, 内存进入, 和中断请求从外部中断引脚或外围功能。

当CPU 执行导致异常的指令或当在指令执行期间, 产生错误条件时异常产生。

异常也通过从Execute Never (XN) 区域或取的指令或进入违反错误区域产生。

从外部中断引脚或外围功能产生中断请求。关于为释放待机模式使用的中断, 在时钟生成器必须做相关设置。关于详细, 参考"8.5 中断"。

(2) 异常 检测

如果多个异常同时发生, CPU以最高优先得到异常。

表 8-1显示异常的优先。"可配置" 意味着能分配优先级到异常。内存管理, 总线错误和使用错误异常可以使能或禁用。如果禁用异常产生, 作为硬错误处理。

表 8-1 异常 类型和优先级

No.	异常 类型	优先级	描述
1	复位	-3 (最高)	Reset 引脚, WDT, POR, VLTD, OFD 或 SYSRETRREQ
2	非-可屏蔽 中断	-2	WDT
3	硬 错误	-1	因为更高优先错误正被处理或禁用, 错误不能激活
4	映射 管理	可配置	从内存保护单元 (MPU) 的异常(注 1) 从 Execute Never (XN) 区域的指令获取
5	总线 错误	可配置	进入违反内存映射硬错误区域
6	使用 错误	可配置	未定义指令执行或别的相关指令执行的错误
7 到 10	保留	-	
11	SVCcall	可配置	带 SVC 指令的系统服务呼叫
12	调试 监视器	可配置	CPU 不错误时调试监视器
13	保留	-	
14	PendSV	可配置	挂起系统服务请求
15	SysTick	可配置	从系统定时器通知
达到 16	外部 中断	可配置	外部中断引脚或外设 功能 (注 2)

注 1: 此产品不包含 MPU.

注 2: 每个产品中外部中断有不同的源和数字。详细, 见 "8.5.1.5 中断源表"。

(3) 优先级 设置

- 优先级

外部中断优先设到中断优先寄存器且别的异常被设到系统处理器优先寄存器中<PRI_n> 位。

配置 <PRI_n> 可以变化, 设置优先需要的位数从 3 位到 8 位根据产品变化。因此, 根据产品能指定的优先值的范围不同。

TMPM475FDFG/FZFG/FYFG 有 3-位 配置.

8-位配置情况下,可以在范围 从 0 到 255 配置优先。最高优先为 "0"。有相同优先的多个元素存在, 数字越小, 优先越高。

- 优先级 组

优先级组可以分成组。通过设置应用中断和复位控制寄存器的 <PRIGROUP>, <PRI_n> 可以分成先占优先和子优先。

优先与占优优先比较。如果优先与占优优先相同, 那么与子优先比较。如果子优先与优先相同, 异常数越小, 优先越高。

表 8-2 显示优先级组设置。表中占优优先和子优先为 <PRI_n> 定义为 8-位 配置时的数。

表 8-2 优先级组设置

<PRIGROUP[2:0]> 设置	<PRI_n[7:0]>		占优优先级 数	子-优先级数
	占先 域	子-优先级 域		
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>的配置少于8位, 低位为 "0"。

例如, 3-位配置情况下, 优先设为 <PRI_n[7:5]> 且 <PRI_n[4:0]> 为 "00000"。

8.1.2.2 异常 处理和分支到中断服务程序 (占先)

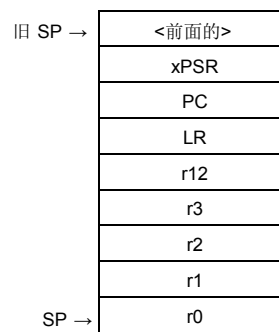
当异常产生, CPU 挂起当前执行的处理器且扩展到中断服务程序 (ISP)。这被叫做 "占先"。

(1) 堆栈

当 CPU 检测到异常, 用以下命令推送以下八个寄存器的内容到堆栈:

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

SP 用八个字减直到完成堆栈。以下显示寄存器内容被推送后堆栈的状态。



(2) 获取 ISR

CPU 使能用存储到寄存器数据获取中断处理指令。

为每个异常准备包含ISR的项地址的向量表。复位后,向量表位地址代码区域中0x0000_0000。通过设置向量表移位寄存器, 可以把向量表放在代码或SRAM 空间的任何地址。

向量表应该也包含主要堆栈的初始值。

(3) 后-到

在执行之前异常ISR以前, 如果CPU 检测到更高优先异常, CPU 处理更高优先异常。这被叫做 "后-到"。

后-到异常导致 CPU 去获取扩展相应 ISR 的新向量地址, 但 CPU 不重新推送寄存器内容到堆栈。

(4) 向量表

向量表如下配置。

必须始终设置开始四个字 (堆栈顶地址, 复位 ISR 地址, NMI ISR 地址, 和硬错误 ISR 地址)。

如果必要为别的异常设置 ISR 地址。

偏移	异常	内容	设置
0x00	复位	主堆栈初始值	需要
0x04	复位	ISR 地址	需要
0x08	非-可屏蔽 中断	ISR 地址	需要
0x0C	硬 错误	ISR 地址	需要
0x10	映射 管理	ISR 地址	可选
0x14	总线 错误	ISR 地址	可选
0x18	使用 错误	ISR 地址	可选
0x1C 到 0x28	保留	-	-
0x2C	SVCall	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

8.1.2.3 执行 ISR

ISR 执行必要相应异常的处理。ISRs 必须由用户准备。

ISR 可能需要包括清除中断请求的代码, 为了当返回到正常程序执行时, 相同中断不会再发生。

关于中断处理详细, 见 "8.5 中断"。

如果ISR执行当前异常期间, 更高优先异常产生, CPU 放弃执行当前 ISR 且服务新的检测异常。

8.1.2.4 异常退出

(1) 执行后从 ISR 返回

当从 ISR 返回, CPU 采取下列之一行为:

- 尾-链

如果挂起异常存在且没有堆栈的异常或中断异常比所有堆栈的异常有更高优先, CPU 返回到挂起的异常的ISR。

此时, 如果存在一个 ISR 和进入别的, CPU 跳过八个寄存器的弹出和八个寄存器的推送。这被叫做 "尾-链"。

- 返回到最后堆栈的 ISR

如果没有中断异常或如果最高优先堆栈的异常优先高于最高优先挂起的异常, CPU 返回到最后堆栈的 ISR。

- 返回到前面的程序

如果没有挂起 =或堆栈的异常, CPU 返回到前面的程序。

(2) 异常退出顺序

返回从 ISR 时, CPU 执行以下工作:

- Pop 八 寄存器

从堆栈弹出八个寄存器 (xPSR, PC, KR, r0 到 r3 和 r12)且调节 SP。

- Load 当前激活中断数

从堆栈的 xPSR 加载当前激活中断数。CPU 使用此追踪返回哪个中断。

- 选择 SP

如果返回到异常(处理器模式), SP为 SP_main。 如果返回到Thread 模式, SP可以为 SP_main 或 SP_process。

8.2 复位 异常

复位异常从以下源生成.

使用时钟生成器的复位标志 (CGRSTFLG) 寄存器识别复位源.

- 外部 $\overline{\text{RESET}}$ 引脚
外部 $\overline{\text{RESET}}$ 引脚从 "低" 到 "高" 改变时, 复位异常发生.
- 通过 POR 复位 异常
上电复位 (POR) 有复位生成 特征. 关于详细, 见 POR 章节.
- 通过 VLTD 复位 异常
低压检测电路 (VLTD) 有复位生成特征. 关于详细, 见 VLTD 章节.
- 通过 OFD 复位 异常
振荡 频率 检测器 (OFD) 有复位生成特征. 关于详细, 见 OFD 章节.
- 通过 WDT 复位 异常
看门狗 定时器 (WDT) 有复位生成特征. 关于详细, 见 WDT 章节.
- 通过 SYSRESETREQ 复位 异常
复位可以通过设置 NVIC 中的应用中断 SYSESETREQ 位和重设置控制寄存器生成.

8.3 非-可屏蔽 中断 (NMI)

看门狗 定时器 (WDT) 有非-可屏蔽中断生成特征. 关于详细, 见 WDT 章节.

8.4 SysTick

SysTick 提供使用 CPU 的系统定时器的中断特征.

当设置 SysTick Reload Value Register 中的值时, 且使能 SysTick Control 和状态寄存器中的 SysTick 特征, 计数器用设在 Reload Value Register 的值加载且开始向下计数。当计数器到达 "0", SysTick 异常产生。可以中断异常和使用标志去知道当定时器到达 "0"。

SysTick Calibration Value Register 保持重新加载值, 用系统定时器计数 10 ms。计数时钟频率随每个产品变化, 所以 SysTick Calibration Value Register 里设置的值也随每个产品变化。

注:此产品, 用 32 由 CGOSCCR <OSCSEL>选择的 fosc 被用作外部 参考 时钟.

8.5 中断

此章节描述路径, 源和需要的中断设置.

用来自每个中断源的中断信号通知CPU中断请求。用最高优先设置优先中断和处理中断请求。

通过时钟生成器通知CPU清除待机模式的中断请求。因此, 时钟生成器中必须做适当设置。

8.5.1 中断源

8.5.1.1 中断路径

图 8-1 表示中断 请求 路径.

由不用作释放待机的外围功能引起的中断直接输入到 CPU (路径 1)。

用作释放待机 (路径 2) 和来自外部中断引脚的中断 (路径 3) 的外围功能中断被输入到时钟生成器且通过释放待机逻辑输入到CPU (路径 4 和 5)。

如果来自外部中断引脚的中断不用作释放待机,它们直接输入到 CPU,不通过待机释放逻辑 (路径 6)。

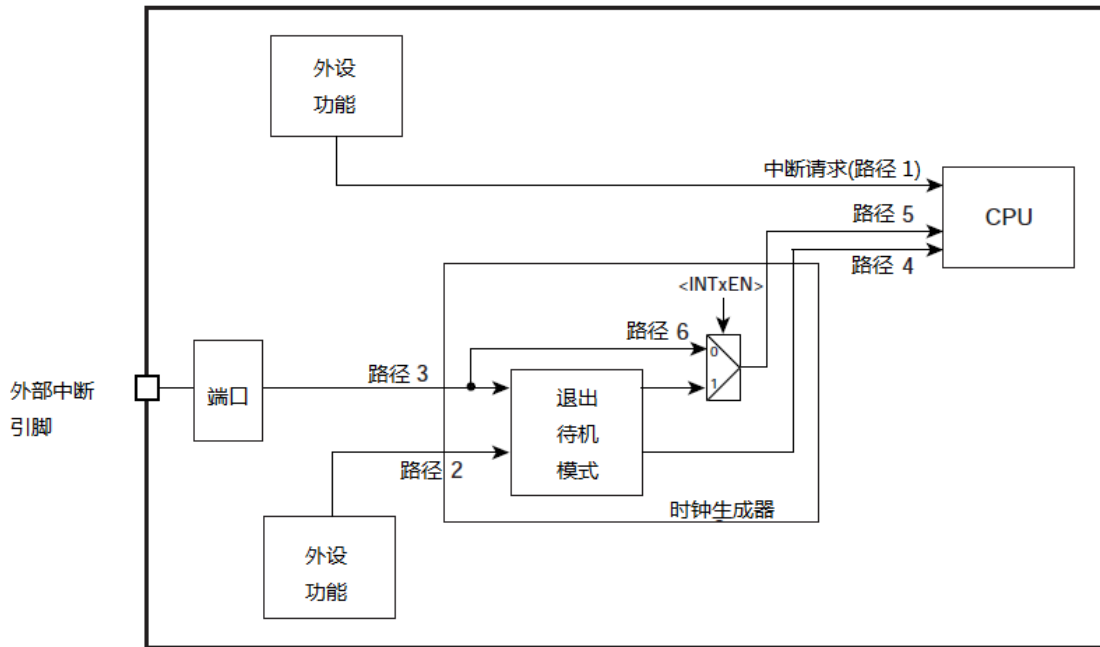


图 8-1 中断源路径

8.5.1.2 生成中断请求

通过设置分配给中断因素的外部中断引脚, 外围功能, 或中断 Set-Pending 寄存器生成中断请求。

- 通过外部中断引脚中断
- 通过外设功能中断

当使用外围功能中断时, 中断生成必须在使用的外围功能中使能。

详细设置, 参考外围功能每个章节。

- 中断的强制生成

强制生成中断请求, 设置 NVIC 中中断保持设置寄存器相应位。

CPU 识别中断请求信号的"高"电平为中断。

8.5.1.3 传输

来自外部引脚或外围功能的中断信号直接送到CPU 除非被用作退出待机模式。

通过时钟生成器把来自可以用作清除待机模式的中断源的中断请求传输到 CPU。这些中断源, 必须事先在时钟生成器做适当的设置。不用作退出待机模式的外部中断源不设置时钟生成器也可以使用。

8.5.1.4 使用外部中断引脚时注意事项

如果使用外部中断, 确保以下不产生不预期的中断。

如果输入禁用 ($PxIE < PxmIE = 0$), 从外部中断引脚的输入为"高"。另外, 如果外部中断不用作释放低功耗的触发器("图 8-1 中断路径" 路径 5), 从外部中断引脚的输入信号直接送到CPU。因为 CPU 识别"高"输入为中断, 如果CPU使能相应中断, 中断发生, 因为输入被禁用。

为了不设置低-功耗触发器使用外部中断, 设置中断引脚输入为"低"且使能它。然后, 使能 CPU 上的中断。

8.5.1.5 中断源表

表 8-3 表示中断源表。

表 8-3 中断源表

No.	中断源		激活电平 (清除低-功耗)	CG 中断模式 控制寄存器
0	INT0	外部中断引脚 0	高/低 缘/电平 可选	CGIMCGA
1	INT1	外部中断引脚 1		
2	INT2	外部中断引脚 2		
3	INT3	外部中断引脚 3	高/低 缘/电平 可选	CGIMCGB
4	INT4	外部中断引脚 4		
5	INT5	外部中断引脚 5		
6	INTRX0	串行接收中断 (通道 0)		

表 8-3 中断源表

No.	中断源		激活电平 (清除低-功耗)	CG 中断 模式 控制 寄存器
7	INTTX0	串行 传输 中断 (通道 0)		
8	INTRX1	串行 接收 中断 (通道 1)		
9	INTTX1	串行 传输 中断 (通道 1)		
10	INTVCN0	向量 引擎 中断 0		
11	INTVCN1	向量 引擎 中断 1		
12	INTEMG0	PMD0 EMG 中断		
13	INTEMG1	PMD1 EMG 中断		
14	INTOVV0	PMD0 OVV 中断		
15	INTOVV1	PMD1 OVV 中断		
16	INTADAPDA	通过 PMD 触发的 ADCA 转换 完成 选择 ADCA 中 INTADxPDA 时中断		
17	INTADBPDA	通过 PMD 触发的 ADCB 转换 完成 选择 ADCB 中 INTADxPDA 时中断		
18	INTADAPDB	通过 PMD 触发的 ADCA 转换 完成 选择 ADCA 中 INTADxPDB 时中断		
19	INTADBPDB	通过 PMD 触发的 ADCB 转换 完成 选择 ADCB 中 INTADxPDB 时中断		
20	INTTB00	16 位 TMRB0 比较 匹配 检测 0/溢出		
21	INTTB01	16 位 TMRB0 比较 匹配 检测 1		
22	INTTB10	16 位 TMRB1 比较 匹配 检测 0/溢出		
23	INTTB11	16 位 TMRB1 比较 匹配 检测 1		
24	INTTB40	16 位 TMRB4 比较 匹配 检测 0/溢出		
25	INTTB41	16 位 TMRB4 比较 匹配 检测 1		
26	INTTB50	16 位 TMRB5 比较 匹配 检测 0/溢出		
27	INTTB51	16 位 TMRB5 比较 匹配 检测 1		
28	INTPMD0	PMD0 PWM 中断		
29	INTPMD1	PMD1 PWM 中断		
30	INTCAP00	16 位 TMRB0 输入 捕捉 0		
31	INTCAP01	16 位 TMRB0 输入 捕捉 1		
32	INTCAP10	16 位 TMRB1 输入 捕捉 0		
33	INTCAP11	16 位 TMRB1 输入 捕捉 1		
34	INTCAP40	16 位 TMRB4 输入 捕捉 0		
35	INTCAP41	16 位 TMRB4 输入 捕捉 1		
36	INTCAP50	16 位 TMRB5 输入 捕捉 0		
37	INTCAP51	16 位 TMRB5 输入 捕捉 1		
38	INT6	外部 中断 引脚 6	高/低 缘/电平 可选	CGIMCGB
39	INT7	外部 中断 引脚 7		
40	INTRX2	串行 接收 中断 (通道 2)		
41	INTTX2	串行 传输 中断 (通道 2)		
42	INTADACPA	ADCA 转换 监视器功能 中断 A		
43	INTADBCPA	ADCB 转换 监视器功能 中断 A		
44	INTADACPB	ADCA 转换 监视器功能 中断 B		
45	INTADBCPB	ADCB 转换 监视器功能 中断 B		
46	INTTB20	16 位 TMRB2 比较 匹配 检测 0/ 溢出		
47	INTTB21	16 位 TMRB2 比较 匹配 检测 1		
48	INTTB30	16 位 TMRB3 比较 匹配 检测 0/ 溢出		

表 8-3 中断源表

No.	中断源		激活电平 (清除低-功耗)	CG 中断 模式 控制 寄存器
49	INTTB31	16 位 TMRB3 比较 匹配 检测 1		
50	INTCAP20	16 位 TMRB2 输入 捕捉 0		
51	INTCAP21	16 位 TMRB2 输入 捕捉 1		
52	INTCAP30	16 位 TMRB3 输入 捕捉 0		
53	INTCAP31	16 位 TMRB3 输入 捕捉 1		
54	INTADASFT	通过 软件开始的 ADCA 转换完成		
55	INTADBSFT	通过 软件开始的 ADCB 转换完成		
56	INTADATMR	通过定时器触发的 ADCA 转换完成		
57	INTADBTMR	通过定时器触发的 ADCB 转换 完成		
58	INT8	外部 中断 引脚 8	高/低 缘/电平 可选	CGIMCGC
59	INT9	外部 中断 引脚 9		
60	INTA	外部 中断 引脚 A		
61	INTB	外部 中断 引脚 B		
62	INTENC00	编码器 输入 0 中断 0		
63	INTENC01	编码器 输入 0 中断 1		
64	INTRX3	串行 接收 中断 (通道 3)		
65	INTTX3	串行 传输 中断 (通道 3)		
66	INTTB60	16 位 TMRB6 比较 匹配 检测 0/溢出		
67	INTTB61	16 位 TMRB6 比较 匹配 检测 1		
68	INTTB70	16 位 TMRB7 比较 匹配 检测 0/溢出		
69	INTTB71	16 位 TMRB7 比较 匹配 检测 1		
70	INTCAP60	16 位 TMRB6 输入 捕捉 0		
71	INTCAP61	16 位 TMRB6 输入 捕捉 1		
72	INTCAP70	16 位 TMRB7 输入 捕捉 0		
73	INTCAP71	16 位 TMRB7 输入 捕捉 1		
74	INTC	外部 中断 引脚 C	高/低 缘/电平 可选	CGIMCGD
75	INTD	外部 中断 引脚 D		
76	INTE	外部 中断 引脚 E		
77	INTF	外部 中断 引脚 F		
78	INTVCT0	向量 引擎 任务 中止 中断 0		
79	INTVCT1	向量 引擎 任务 中止 中断 1		
80	INTSBI0	I2C 中断 0		
81	INTCANRX	CAN 接收		
82	INTCANTX	CAN 传输		
83	INTCANGB	CAN 状态		
84	INTTB80	16 位 TMRB8 比较 匹配 检测 0/溢出		
85	INTTB81	16 位 TMRB8 比较 匹配 检测 1		
86	INTTB90	16 位 TMRB9 比较 匹配 检测 0/溢出		
87	INTTB91	16 位 TMRB9 比较 匹配 检测 1		
88	INTCAP80	16 位 TMRB8 输入 捕捉 0		
89	INTCAP81	16 位 TMRB8 输入 捕捉 1		
90	INTCAP90	16 位 TMRB9 输入 捕捉 0		
91	INTCAP91	16 位 TMRB9 输入 捕捉 1		
92	INTDMACATC	DMA 传输 中断		
93	INTDMACAERR	DMA 错误 中断		
94	INTENC10	编码器 输入 1 中断 0		

表 8-3 中断源表

No.	中断源		激活电平 (清除低-功耗)	CG 中断 模式 控制 寄存器
95	INTENC11	编码器 输入 1 中断 1		

8.5.1.6 激活 电平

激活电平表示 哪种中断源信号的变化触发中断。CPU识别"高" 电平中断信号为中断。直接从外围功能送入CPU的中断信号被配置为输出 "高"来表示中断请求。

设置激活电平到可以触发释放低-功耗的中断的时钟生成器。设置来自外围功能的中断请求为上升-缘 或下降-缘触发。来自中断引脚的中断请求可以设置为电平-感应 ("高" 或 "低") 或 缘-触发 (下降的上升)。

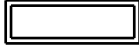
如果中断源用作清除低-功耗模式, 另外需要设置相关时钟生成器寄存器。使能 $CGIMCGx<INTxEN>$ 位和规定 $CGIMCGx<EMCGx>$ 位中的激活电平。必须设置来自每个外围功能的中断请求的激活电平如表 8-3所示。

时钟生成器的中断请求检测用"高" 电平信号通知 CPU。

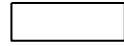
8.5.2 中断 处理

8.5.2.1 流程图



以下表示 如何处理 异常/中断. 在以下描述中,

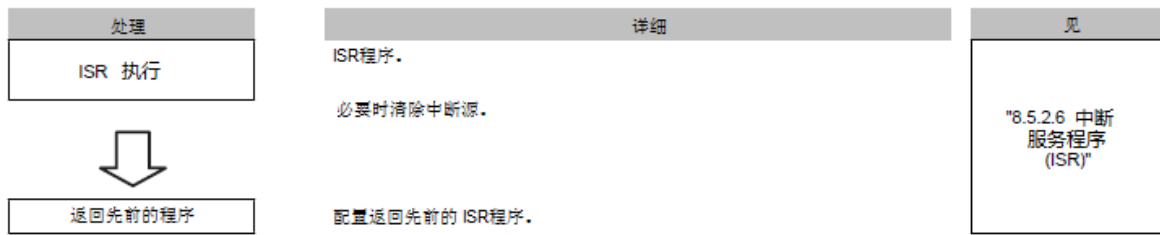


指 硬件 处理。



指 软件 处理。

处理	详细	见
<div style="border: 1px solid black; padding: 5px; text-align: center;">检测设置</div> <hr/> <div style="border: 1px solid black; padding: 5px; text-align: center;">发送中断信号设置</div>	<p>设置相关检测中断NVIC 寄存器。 设置时钟生成器 如果每个中断源用作清除待机模式。 o 共通 设置 NVIC 寄存器 o 设置清除 待机 模式 时钟 生成器 根据中断类型执行适当的设置发送中断信号。 o 外部引脚中断设置 端口 o 外围功能中断设置 外围 功能 (详见每个外围功能章节。)</p>	<div style="border: 1px solid black; padding: 5px; text-align: center;">"8.5.2.2 准备"</div>
 <div style="border: 1px dashed black; padding: 5px; text-align: center;">中断生成</div>	<p>中断请求生成。</p>	
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; font-size: small;">不清除 待机模式</div> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; font-size: small;">清除 待机模式</div> </div> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-top: 10px;">CG 检测 中断 (清除待机模式)</div>	<p>用作清除待机模式的中断线时钟生成器连接到 CPU</p>	<div style="border: 1px solid black; padding: 5px; text-align: center; font-size: small;">"8.5.2.3 由时钟 生成器检测"</div>
<div style="border: 1px solid black; padding: 5px; text-align: center;">CPU 检测中断</div>	<p>CPU 检测中断。 如果多个中断请求同时发生, 根据优先顺序检测高优先中断请求。</p>	<div style="border: 1px solid black; padding: 5px; text-align: center; font-size: small;">"8.5.2.4由CPU检测"</div>
<div style="border: 1px solid black; padding: 5px; text-align: center;">CPU 处理中断</div>	<p>CPU 处理中断。 进入ISR前, CPU 推送寄存器内容到堆栈。</p>	<div style="border: 1px solid black; padding: 5px; text-align: center; font-size: small;">"8.5.2.5 CPU处理"</div>
		



8.5.2.2 准备

准备中断时, 需要注意配置的顺序避免任何不预期的中断发生。

初始化中断或变化其配置必须基本执行以下顺序。禁用CPU中断。从CPU最远的路径配置。然后使能CPU中断。

配置时钟生成器, 必须遵循这里所描述的顺序不引起任何不预期的中断。首先, 配置预先条件。其次, 清除时钟生成器中相关中断的数据, 然后使能中断。

以下部分中断处理的顺序排列且描述如何配置它们。

1. 通过 CPU 禁止 中断
2. CPU 寄存器 设置
3. 预配置 (1) (从 外部 中断 引脚中断)
4. 预配置 (2) (从 外设 功能中断)
5. 预配置 (3) (中断 设置-挂起 寄存器)
6. 配置时钟 生成器
7. 通过 CPU 使能 中断

(1) 通过 CPU 禁止 中断

使CPU 不接受任何中断, 写 "1" 到PRIMASK 寄存器的相应位。所有非-可屏蔽中断和硬错误以外的中断和异常可以被屏蔽。

使用 "MSR" 指令设置此寄存器。

中断 屏蔽 寄存器	
PRIMASK	← "1" (中断 禁止)

注 1: PRIMASK 寄存器不能被入门级用户修改。

注 2: 当 "1" 被设到 PRIMASK 寄存器, 如果错误产生, 这被认为是硬错误。

(2) CPU 寄存器 设置

可以分配优先级通过写入NVIC 寄存器的中断优先寄存器中<PRI_n> 领域。

每个中断源提供八位分配优先电平从0到255,但实际使用位数随每个产品而变化。优先级0为最高优先级。如果多个源有相同优先,数字最小的中断源有最高优先。

可以通过使用应用中断和复位控制 寄存器中的 PRIGROUP 分配分组优先。

NVIC 寄存器		
<PRI_n>	←	"优先级"
<PRIGROUP >	←	"组 优先级" (如果需要此可配置.)

注:"n" 指对应异常/中断.

此产品用三位分配 优先级.

(3) 预 配置 (1) (从外部中断引脚中断)

设置 "1" 到对应 引脚的端口功能寄存器. 设置 PxIE[m] 允许引脚被用作输入端口.

端口 寄存器		
PxIE<PxIE>	←	"1"

注:x: 端口 数 / m: 对应 位

设置 PxIE 使能 输入会使能相应中断输入。当心不要使能不使用的中断。

(4) 预配置 (2) (从外设功能中断)

设置 变化取决于被使用外设 功能. 详细见各外设能章节.

(5) 预配置 (3) (中断 设置-挂起 寄存器)

通过使用中断设置-挂起 寄存器生成中断, 设置 "1" 到此寄存器对应 位.

NVIC 寄存器		
中断 设置-挂起 [m]	←	"1"

注:m: 对应 位

(6) 配置时钟 生成器

用作退出低-功耗模式的中断源,需要设置激活电平和使能时钟生成器的CGIMCG 寄存器中的中断。CGIMCG 寄存器可以配置每个源。

使能中断前, 清除已保持的相应中断请求。这可以避免不预期的中断。清除相应中断请求, 写入用到中断的相应值到CGICRCG 寄存器。每个值见 "8.6.3.2 CGICRCG(CG 中断请求清除寄存器)"。

可以不设置时钟生成器使用外部引脚的中断请求, 如果它们不被用作退出低-功耗模式。然而, "高" 脉冲或 "高"-电平信号必须输入为了 CPU 能检测它为一个中断请求。另外, 掌握 "8.5.1.4 使用外部中断引脚时注意事项"的描述。

时钟生成器 寄存器		
CGIMCGn<EMCGm>	←	激活 电平
CGICRCG<ICRCG>	←	对应被使用中中断的值
CGIMCGn<INTmEN>	←	"1" (中断 使能)

注:n: 寄存器 数 / m: 分配到中断 源数

(7) 通过 CPU 使能 中断

使能CPU中断如下所示

清除Interrupt Clear-Pending 寄存器中的挂起中断。用Interrupt Set-Enable 寄存器使能预期的中断。寄存器每个位分配到一个中断源。

写 "1" 到Interrupt Clear-Pending寄存器相应位清除挂起的中断。写入 "1"到Interrupt Set-Enable 寄存器相应位使能预期的中断。

为了生成Interrupt Set-Pending 寄存器设置中的中断, 如果清除挂起中断, 触发器中断因素丢失。因此, 此工作不需要。

最后, PRIMASK 寄存器清零。

NVIC 寄存器		
中断 清除-挂起 [m]	←	"1"
中断 设置-使能 [m]	←	"1"
中断 屏蔽 寄存器		
PRIMASK	←	"0"

注 1: m : 对应 位

注 2: PRIMASK 寄存器 不能被入门级用户修改.

8.5.2.3 通过 时钟 生成器检测

如果中断源用作退出待机模式,根据时钟生成器指定的激活电平, 检测出中断请求, 且通知CPU。

缘-触发中断请求,一旦检测, 保持在时钟生成器中。电平-感应中断请求必须保持在激活电平直到被检测, 否则当信号电平从激活到不激活变化, 中断请求停止存在。

当时钟生成器检测中断请求, 保持用"高"电平传输中断信号到 CPU 直到在 CG Interrupt Request Clear (CGICRCG) Register 中的中断请求清除。如果不清除中断请求退出待机模式, 当正常工作恢复时会再检测到相同的中断。务必清除 ISR 中每个中断请求。

8.5.2.4 通过 CPU 检测

CPU 用最高优先级检测中断请求。

8.5.2.5 CPU 处理

检测中断时, CPU 推送 xPSR, PC, LR, R12 和 r3 到 r0 的内容到堆栈 然后进入 ISR。

8.5.2.6 中断 服务 程序 (ISR)

ISR 需要根据用到的应用规定编程。此部分描述何为服务程序编程推荐的和如何清除源。

(1) ISR 期间处理

必要时 ISR 通常推送寄存器内容到堆栈和处理中断。Cortex-M4F 核自动推送 xPSR, PC, LR, R12 和 r3 到 r0 的内容到堆栈。它们不需要别的编程。

必要时推送别的寄存器的内容。

即使当 ISR 被执行时, 有更高优先和异常的诸如 NMI 的中断请求被接受。建议推送可以重写的通用-目的寄存器的内容。

(2) 清除中断 源

如果中断源用作清除待机 模式, 每个中断请求必须用CG Interrupt Request Clear (CGICRCG) 寄存器清除。

如果中断源设为电平-感应, 中断请求持续存在直到在它的源被清除。因此, 中断源必须清除。清除中断源自动清除来自时钟生成器的中断请求信号。

如果中断设为缘-感应,通过设置 CGICRCG 寄存器中相应值清除中断请求。当激活缘再产生时, 会检测到新中断请求。

8.6 异常/中断-相关 寄存器

此章节以下描述 CPU 的 NVIC 寄存器和时钟生成器寄存器用其独立的地址表示。

8.6.1 寄存器 表

关于基地址详细, 参考 "内存映射"的 "地址表外设 功能" 章节.

NVIC 寄存器

基地址 = 0xE000_E000

寄存器 名	地址
SysTick 控制 和 状态 寄存器	0x0010
SysTick Reload 值 寄存器	0x0014
SysTick Current 值 寄存器	0x0018
SysTick 标度值 寄存器	0x001C
中断 设置-使能 寄存器 1	0x0100
中断 设置-使能 寄存器 2	0x0104
中断 设置-使能 寄存器 3	0x0108
中断 清除-使能 寄存器 1	0x0180
中断 清除-使能 寄存器 2	0x0184
中断 清除-使能 寄存器 3	0x0188
中断 设置-挂起 寄存器 1	0x0200
中断 设置-挂起 寄存器 2	0x0204
中断 设置-挂起 寄存器 3	0x0208
中断 清除-挂起 寄存器 1	0x0280
中断 清除-挂起 寄存器 2	0x0284
中断 清除-挂起 寄存器 3	0x0288
中断 优先级 寄存器	0x0400 到 0x047F
向量表 偏移 寄存器	0x0D08
应用 中断 和 复位 控制 寄存器	0x0D0C
系统 处理器 优先级 寄存器	0x0D18, 0x0D1C, 0x0D20
系统 处理器 控制 和 状态 寄存器	0x0D24

外设 功能 名 : CG

寄存器 名	地址
CG 中断 请求 清除 寄存器	CGICRCG 0x0014
复位 标志 寄存器	CGRSTFLG 0x001C
CG 中断 模式 控制 寄存器 A	CGIMCGA 0x0020
CG 中断 模式 控制 寄存器 B	CGIMCGB 0x0024
CG 中断 模式 控制 寄存器 C	CGIMCGC 0x0028
CG 中断 模式 控制 寄存器 D	CGIMCGD 0x002C

8.6.2 NVIC 寄存器

8.6.2.1 SysTick 控制和状态 寄存器

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

位	位符	类型	功能
31-17	-	R	读作 0.
16	COUNTFLAG	R/W	0: 定时器不计数到 0 1: 定时器计数到 0 返回 "1" 如果定时器计数到 "0"因为最后时间已被读。 清除 SysTick 控制 和 状态 寄存器的所有部分的读。
15-3	-	R	读作 0.
2	CLKSOURCE	R/W	0: 外部参考时钟 (fosc/32) 1: CPU 时钟 (fsys)
1	TICKINT	R/W	0: 不挂起 SysTick 1: 挂起 SysTick
0	ENABLE	R/W	0: 禁止 1: 使能 如果 "1" 设置, 用 Reload 值寄存器的值重载和开始工作。

注:此 产品, 用 32 由 CGOSCCR <OSCSEL>选择的 fosc 被用作外部参考时钟.

8.6.2.2 SysTick Reload 值 寄存器

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

位	位 符	类型	功能
31-24	-	R	读作 0.
23-0	RELOAD	R/W	Reload 值 设置加载值到 SysTick 当前值寄存器， 当定时器到达 "0"时。

8.6.2.3 SysTick Current 值 寄存器

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

位	位 符	类型	功能
31-24	-	R	读作 0.
23-0	CURRENT	R/W	[读] Current SysTick 定时器 值 [写] 清除 用任何值写入此 寄存器清除它为 0. 清除此寄存器也清除<COUNTFLAG> SysTick 控制和状态寄存器的位.

8.6.2.4 SysTick 刻度值 寄存器

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

位	位 符	类型	功能
31	NOREF	R	0: 提供参考 时钟 1: 无参考 时钟
30	SKEW	R	0: 标度 值是 10 ms. 1: 标度 值不是 10 ms.
29-24	-	R	读 作 0.
23-0	TENMS	R	标度值 (注)

注:此 产品不准备仲裁值.

8.6.2.5 中断 控制 寄存器

各中断源有中断设置-使能 寄存器, 中断 清除-使能 寄存器, 中断设置-挂起寄存器和中断清除-挂起寄存器.

(1) 中断 设置-使能 寄存器

此寄存器指定使能中断和确认中断的使能/禁止 状态. 设置此寄存器到 "1"时, 对应中断使能.

写入"0" 无意义.

读 此寄存器, 可以确认对应中断的使能/禁止 状态 .

清除此寄存器位, 中断 清除-使能 寄存器对应位清为 "0".

位 符	类型	功能
SETENA	R/W	中断 No. [95:0] [写] 1: 使能 中断 [读] 0: 中断 状态 禁止 1: 中断 状态 使能

(a) 中断 设置-使能 寄存器 1

	31	30	29	28	27	26	25	24
位 符	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
位 符	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
位 符	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
位 符	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

(b) 中断 设置-使能 寄存器 2

	31	30	29	28	27	26	25	24
位 符	SETENA (中断 63)	SETENA (中断 62)	SETENA (中断 61)	SETENA (中断 60)	SETENA (中断 59)	SETENA (中断 58)	SETENA (中断 57)	SETENA (中断 56)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	SETENA (中断 55)	SETENA (中断 54)	SETENA (中断 53)	SETENA (中断 52)	SETENA (中断 51)	SETENA (中断 50)	SETENA (中断 49)	SETENA (中断 48)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	SETENA (中断 47)	SETENA (中断 46)	SETENA (中断 45)	SETENA (中断 44)	SETENA (中断 43)	SETENA (中断 42)	SETENA (中断 41)	SETENA (中断 40)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	SETENA (中断 39)	SETENA (中断 38)	SETENA (中断 37)	SETENA (中断 36)	SETENA (中断 35)	SETENA (中断 34)	SETENA (中断 33)	SETENA (中断 32)
复位后	0	0	0	0	0	0	0	0

(c) 中断 设置-使能 寄存器 3

	31	30	29	28	27	26	25	24
位 符	SETENA (中断 95)	SETENA (中断 94)	SETENA (中断 93)	SETENA (中断 92)	SETENA (中断 91)	SETENA (中断 90)	SETENA (中断 89)	SETENA (中断 88)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	SETENA (中断 87)	SETENA (中断 86)	SETENA (中断 85)	SETENA (中断 84)	SETENA (中断 83)	SETENA (中断 82)	SETENA (中断 81)	SETENA (中断 80)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	SETENA (中断 79)	SETENA (中断 78)	SETENA (中断 77)	SETENA (中断 76)	SETENA (中断 75)	SETENA (中断 74)	SETENA (中断 73)	SETENA (中断 72)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	SETENA (中断 71)	SETENA (中断 70)	SETENA (中断 69)	SETENA (中断 68)	SETENA (中断 67)	SETENA (中断 66)	SETENA (中断 65)	SETENA (中断 64)
复位后	0	0	0	0	0	0	0	0

(2) 中断 清除-使能 寄存器

此寄存器指定禁止中断和确认中断的使能/禁止 状态.设置 此寄存器到 "1"时, 对应中断禁止.

写入"0" 无意义.

读 此 寄存器, 可以确认对应 中断的使能/禁止 状态 .

位 符	类型	功能
CLRENA	R/W	中断 No. [95:0] [写] 1: 禁止 中断 [读] 0: 中断 状态 禁止 1: 中断 状态 使能

(a) 中断 清除-使能 寄存器 1

	31	30	29	28	27	26	25	24
位 符	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
位 符	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
位 符	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
位 符	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

(b) 中断清除-使能 寄存器 2

	31	30	29	28	27	26	25	24
位 符	CLRENA (中断 63)	CLRENA (中断 62)	CLRENA (中断 61)	CLRENA (中断 60)	CLRENA (中断 59)	CLRENA (中断 58)	CLRENA (中断 57)	CLRENA (中断 56)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	CLRENA (中断 55)	CLRENA (中断 54)	CLRENA (中断 53)	CLRENA (中断 52)	CLRENA (中断 51)	CLRENA (中断 50)	CLRENA (中断 49)	CLRENA (中断 48)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	CLRENA (中断 47)	CLRENA (中断 46)	CLRENA (中断 45)	CLRENA (中断 44)	CLRENA (中断 43)	CLRENA (中断 42)	CLRENA (中断 41)	CLRENA (中断 40)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CLRENA (中断 39)	CLRENA (中断 38)	CLRENA (中断 37)	CLRENA (中断 36)	CLRENA (中断 35)	CLRENA (中断 34)	CLRENA (中断 33)	CLRENA (中断 32)
复位后	0	0	0	0	0	0	0	0

(b) 中断清除-使能 寄存器 3

	31	30	29	28	27	26	25	24
位 符	CLRENA (中断 95)	CLRENA (中断 94)	CLRENA (中断 93)	CLRENA (中断 92)	CLRENA (中断 91)	CLRENA (中断 90)	CLRENA (中断 89)	CLRENA (中断 88)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	CLRENA (中断 87)	CLRENA (中断 86)	CLRENA (中断 85)	CLRENA (中断 84)	CLRENA (中断 83)	CLRENA (中断 82)	CLRENA (中断 81)	CLRENA (中断 80)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	CLRENA (中断 79)	CLRENA (中断 78)	CLRENA (中断 77)	CLRENA (中断 76)	CLRENA (中断 75)	CLRENA (中断 74)	CLRENA (中断 73)	CLRENA (中断 72)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CLRENA (中断 71)	CLRENA (中断 70)	CLRENA (中断 69)	CLRENA (中断 68)	CLRENA (中断 67)	CLRENA (中断 66)	CLRENA (中断 65)	CLRENA (中断 64)
复位后	0	0	0	0	0	0	0	0

(3) 中断 设置-挂起 寄存器

此寄存器指定挂起中断和确认中断挂起 状态。

设置此寄存器为 "1"时,对应中断挂起. 但此寄存器对已经挂起或禁止的中断无效。

写入 "0" 无 意义。

读此寄存器, 可以确认对应中断挂起状态 。

清除此寄存器位, 中断清除-挂起寄存器对应位清为 "0"。

位 符	类 型	功 能
SETPEND	R/W	中断 No. [95:0] [写] 1: 挂起 中断 [读] 0: 无挂起 1: 挂起

(a) 中断 设置-挂起 寄存器 1

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

(b) 中断 设置-挂起 寄存器 2

	31	30	29	28	27	26	25	24
位 符	SETPEND (中断 63)	SETPEND (中断 62)	SETPEND (中断 61)	SETPEND (中断 60)	SETPEND (中断 59)	SETPEND (中断 58)	SETPEND (中断 57)	SETPEND (中断 56)
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
位 符	SETPEND (中断 55)	SETPEND (中断 54)	SETPEND (中断 53)	SETPEND (中断 52)	SETPEND (中断 51)	SETPEND (中断 50)	SETPEND (中断 49)	SETPEND (中断 48)
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
位 符	SETPEND (中断 47)	SETPEND (中断 46)	SETPEND (中断 45)	SETPEND (中断 44)	SETPEND (中断 43)	SETPEND (中断 42)	SETPEND (中断 41)	SETPEND (中断 40)
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
位 符	SETPEND (中断 39)	SETPEND (中断 38)	SETPEND (中断 37)	SETPEND (中断 36)	SETPEND (中断 35)	SETPEND (中断 34)	SETPEND (中断 33)	SETPEND (中断 32)
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

(c) 中断设置-挂起 寄存器 3

	31	30	29	28	27	26	25	24
位 符	SETPEND (中断 95)	SETPEND (中断 94)	SETPEND (中断 93)	SETPEND (中断 92)	SETPEND (中断 91)	SETPEND (中断 90)	SETPEND (中断 89)	SETPEND (中断 88)
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
位 符	SETPEND (中断 87)	SETPEND (中断 86)	SETPEND (中断 85)	SETPEND (中断 84)	SETPEND (中断 83)	SETPEND (中断 82)	SETPEND (中断 81)	SETPEND (中断 80)
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
位 符	SETPEND (中断 79)	SETPEND (中断 78)	SETPEND (中断 77)	SETPEND (中断 76)	SETPEND (中断 75)	SETPEND (中断 74)	SETPEND (中断 73)	SETPEND (中断 72)
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
位 符	SETPEND (中断 71)	SETPEND (中断 70)	SETPEND (中断 69)	SETPEND (中断 68)	SETPEND (中断 67)	SETPEND (中断 66)	SETPEND (中断 65)	SETPEND (中断 64)
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

(4) 中断 清除-挂起 寄存器

此寄存器规定清除挂起中断和确认中断挂起状态。当设置此寄存器为 "1", 相应挂起中断清除。 但此已开始中断的寄存器无效。

写入 "0" 无意义。

读此寄存器,可以确认对应中断挂起状态。

位 符	类 型	功 能
SETPEND	R/W	中断 No. [95:0] [写] 1: 清除挂起中断 [读] 0: 无挂起 1: 挂起

(a) 中断 清除-挂起 寄存器 1

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

(b) 中断清除-挂起 寄存器 2

	31	30	29	28	27	26	25	24
位 符	CLRPEND (中断 63)	CLRPEND (中断 62)	CLRPEND (中断 61)	CLRPEND (中断 60)	CLRPEND (中断 59)	CLRPEND (中断 58)	CLRPEND (中断 57)	CLRPEND (中断 56)
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
位 符	CLRPEND (中断 55)	CLRPEND (中断 54)	CLRPEND (中断 53)	CLRPEND (中断 52)	CLRPEND (中断 51)	CLRPEND (中断 50)	CLRPEND (中断 49)	CLRPEND (中断 48)
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
位 符	CLRPEND (中断 47)	CLRPEND (中断 46)	CLRPEND (中断 45)	CLRPEND (中断 44)	CLRPEND (中断 43)	CLRPEND (中断 42)	CLRPEND (中断 41)	CLRPEND (中断 40)
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
位 符	CLRPEND (中断 39)	CLRPEND (中断 38)	CLRPEND (中断 37)	CLRPEND (中断 36)	CLRPEND (中断 35)	CLRPEND (中断 34)	CLRPEND (中断 33)	CLRPEND (中断 32)
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

(c) 中断清除-挂起 寄存器 3

	31	30	29	28	27	26	25	24
位 符	CLRPEND (中断 95)	CLRPEND (中断 94)	CLRPEND (中断 93)	CLRPEND (中断 92)	CLRPEND (中断 91)	CLRPEND (中断 90)	CLRPEND (中断 89)	CLRPEND (中断 88)
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
位 符	CLRPEND (中断 87)	CLRPEND (中断 86)	CLRPEND (中断 85)	CLRPEND (中断 84)	CLRPEND (中断 83)	CLRPEND (中断 82)	CLRPEND (中断 81)	CLRPEND (中断 80)
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
位 符	CLRPEND (中断 79)	CLRPEND (中断 78)	CLRPEND (中断 77)	CLRPEND (中断 76)	CLRPEND (中断 75)	CLRPEND (中断 74)	CLRPEND (中断 73)	CLRPEND (中断 72)
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
位 符	CLRPEND (中断 71)	CLRPEND (中断 70)	CLRPEND (中断 69)	CLRPEND (中断 68)	CLRPEND (中断 67)	CLRPEND (中断 66)	CLRPEND (中断 65)	CLRPEND (中断 64)
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

8.6.2.6 中断 优先级 寄存器

各中断 提供八位中断优先级寄存器.

以下 表示对应 中断数的中断 优先级寄存器的地址.

	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	
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32	
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36	
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40	
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44	
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48	
0xE000_E434	PRI_55	PRI_54	PRI_53	PRI_52	
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56	
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60	
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64	
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68	
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72	
0xE000_E44C	PRI_79	PRI_78	PRI_77	PRI_76	
0xE000_E450	PRI_83	PRI_82	PRI_81	PRI_80	
0xE000_E454	PRI_87	PRI_86	PRI_85	PRI_84	
0xE000_E458	PRI_91	PRI_90	PRI_89	PRI_88	
0xE000_E45C	PRI_95	PRI_94	PRI_93	PRI_92	

用作分配优先的位数随每个产品而变化。此产品使用三位来分配优先。

以下显示中断数字 0 到 3 的中断优先寄存器的领域。所有别的中断数字的中断优先寄存器有相等领域。当读时，未使用位返回 "0", 且写入未使用位无影响。

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

位	位符	类型	功能
31-29	PRI_3	R/W	中断 数优先级 3
28-24	-	R	读作 0.
23-21	PRI_2	R/W	中断 数优先级 2
20-16	-	R	读作 0.
15-13	PRI_1	R/W	中断 数优先级 1
12-8	-	R	读作 0.
7-5	PRI_0	R/W	中断优先级数 0
4-0	-	R	读作 0.

8.6.2.7 向量表 偏移 寄存器

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

位	位 符	类型	功能
31-7	TBLOFF	R/W	偏移 值 从TBLCASE中指定的顶端空间设置偏移值。 偏移必须基于此表异常数对齐。此意味着最小对齐为可以使用多达 16 中断的 32 字。更多中断,必须通过四舍五入后两个来调节对齐。
6-0	-	R	读作 0.

8.6.2.8 应用中断和复位 控制 寄存器

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

位	位符	类型	功能
31-16	VECTKEY (写)/ VECTKEY- STAT(读)	R/W	寄存器 键 [写] 写入此寄存器需要<VECTKEY> 中 0x5FA 域。 [读] 读作 0xFA05.
15	EDIANESS	R/W	字顺 位:(注 1) 1: 大 字顺 0: 小 字顺
14-11	-	R	读 作 0.
10-8	PRIGROUP	R/W	中断 优先级 组 000: 七 位 占 先 优 先 级, 一 位 子-优 先 级 001: 六 位 占 先 优 先 级, 两 位 子-优 先 级 010: 五 位 占 先 优 先 级, 三 位 子-优 先 级 011: 四 位 占 先 优 先 级, 四 位 子-优 先 级 100: 三 位 占 先 优 先 级, 五 位 子-优 先 级 101: 两 位 占 先 优 先 级, 六 位 子-优 先 级 110: 一 位 占 先 优 先 级, 七 位 子-优 先 级 111: 无 占 先 优 先 级, 八 位 子-优 先 级 分开中断 优先级 寄存器 <PRI_n> 为 占 先 优 先 级 和 子 优 先 级 的 配 置 位.
7-3	-	R	读 作 0.
2	SYSRESETRE Q	R/W	系统复位 请求. 1=CPU 输出 SYSRESETREQ 信号. (注 2)
1	VECTCLR ACTIVE	R/W	清除激活向量 位 1: 清除所有激活 NMI, 错误, 和中断状态信息 0: 不 清除. 此位自己-清除. 应用的任务是 重新初始化堆栈.
0	VECTRESET	R/W	系统复位 位 1: 复位 系统 0: 不 复位 系统 通过设置 "1", 用调试组成 (FPB, DWT 和 ITM) 异常复位系统且此位也清零.

注 1: 默认小-字顺为此产品映射格式.

注 2: SYSRESETREQ 输出时, 此产品执行预热复位. 通过预热复位, 清除<SYSRESETREQ>.

8.6.2.9 系统 处理器 优先级 寄存器

各异常提供八位系统 处理器 优先级 寄存器.

以下表示系统 处理器 优先级 寄存器 对应 各 异常的地址.

	31	24 23	16 15	8 7	0
0xE000_ED18	PRI_7	PRI_6 (使用 错误)	PRI_5 (总线 错误)	PRI_4 (内存 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)	

被用来分配优先级的位数随各产品变化. 此产品用三位分配优先级.

以下表示映射管理的系统处理器优先级寄存器域, 总线错误和使用错误. 返回 "0" 时读未被使用位, 和写入未被使用位无效.

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

位	位 符	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 错误优先级
20-16	-	R	读作 0.
15-13	PRI_5	R/W	总线 错误优先级
12-8	-	R	读作 0.
7-5	PRI_4	R/W	映射 管理优先级
4-0	-	R	读作 0.

8.6.2.10 系统 处理器 控制和状态 寄存器

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	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
位 符	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICKACT	PENDSVACT	-	MONITOR ACT
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-19	-	R	读作 0.
18	USGFAULT ENA	R/W	使用 错误 0: 禁止 1: 使能
17	BUSFAUL TENA	R/W	总线 错误 0: 禁止 1: 使能
16	MEMFAULT ENA	R/W	映射 管理 0: 禁止 1: 使能
15	SVCALL PENDED	R/W	SVCall 0: 不 挂起 1: 挂起
14	BUSFAULT PENDED	R/W	总线 错误 0: 不 挂起 1: 挂起
13	MEMFAULT PENDED	R/W	映射 管理 0: 不 挂起 1: 挂起
12	USGFAULT PENDED	R/W	使用 错误 0: 不 挂起 1: 挂起
11	SYSTICKACT	R/W	SysTick 0: 非激活 1: 激活
10	PENDSVACT	R/W	挂起 SV 0: 非激活 1: 激活
9	-	R	读作 0.
8	MONITORACT	R/W	调试 监视器 0: 非激活 1: 激活
7	SVCALLACT	R/W	SVCall 0: 非激活 1: 激活
6-4	-	R	读作 0.

位	位符	类型	功能
3	USGFAULT ACT	R/W	使用 错误 0: 非激活 1: 激活
2	-	R	读 作 0.
1	BUSFAULT ACT	R/W	总线 错误 0: 非激活 1: 激活
0	MEMFAULT ACT	R/W	映射 管理 0: 非激活 1: 激活

注:必须非常小心清除或设置激活位, 因为清除和设置这些位不修复堆栈内容。

8.6.3 时钟 生成器 寄存器

8.6.3.1 CG 中断 模式 控制 寄存器

此寄存器指定释放低功耗模式的激活电平和使能/禁止释放低功耗模式. 检测激活电平可以从此寄存器读.

位 符	类型	功能
EMCGx[2:0]	R/W	选择释放低功耗模式的激活电平 激活电平可以从表 8-4 选择. 000: "低" 电平 001: "高" 电平 010: 下降 缘 011: 上升 缘 100: 双缘 101 到 111: 设置禁止
EMSTx[1:0]	R	检测激活 电平 (此位 有效 EMCGx[2:0]="100".) 00: - 01: 上升 缘 10: 下降 缘 11: 双 缘
INTxEN	R/W	释放低 功 耗 模 式 0: 禁止 1: 使能

表 8-4 释放低功耗模式的激活电平

中断源		激活电平 控制寄存器	释放低功耗模式的激活电平				
			"低" 电平	"高" 电平	上升 缘	下降 缘	双缘
INT0	外部中断引脚 0	CGIMCGA <EMCG00[1:0]>	o	o	o	o	o
INT1	外部中断引脚 1	CGIMCGA <EMCG01[1:0]>	o	o	o	o	o
INT2	外部中断引脚 2	CGIMCGA <EMCG02[1:0]>	o	o	o	o	o
INT3	外部中断引脚 3	CGIMCGA <EMCG03[1:0]>	o	o	o	o	o
INT4	外部中断引脚 4	CGIMCGB <EMCG04[1:0]>	o	o	o	o	o
INT5	外部中断引脚 5	CGIMCGB <EMCG05[1:0]>	o	o	o	o	o
INT6	外部中断引脚 6	CGIMCGB <EMCG06[1:0]>	o	o	o	o	o
INT7	外部中断引脚 7	CGIMCGB <EMCG07[1:0]>	o	o	o	o	o
INT8	外部中断引脚 8	CGIMCGC <EMCG08[1:0]>	o	o	o	o	o
INT9	外部中断引脚 9	CGIMCGC <EMCG09[1:0]>	o	o	o	o	o
INTA	外部中断引脚 A	CGIMCGC <EMCG0A[1:0]>	o	o	o	o	o
INTB	外部中断引脚 B	CGIMCGC <EMCG0B[1:0]>	o	o	o	o	o
INTC	外部中断引脚 C	CGIMCGD <EMCG0C[1:0]>	o	o	o	o	o
INTD	外部中断引脚 D	CGIMCGD <EMCG0D[1:0]>	o	o	o	o	o
INTE	外部中断引脚 E	CGIMCGD <EMCG0E[1:0]>	o	o	o	o	o
INTF	外部中断引脚 F	CGIMCGD <EMCG0F[1:0]>	o	o	o	o	o

注: 被标为"o"释放低功耗模式激活电平可以被使用.
被标为"x"激活电平不可以被使用.

(1) CGIMCGA(CG 中断模式控制寄存器 A)

	31	30	29	28	27	26	25	24
位符	-	EMCG03			EMST03		-	INT03EN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
位符	-	EMCG02			EMST02		-	INT02EN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
位符	-	EMCG01			EMST01		-	INT01EN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
位符	-	EMCG00			EMST00		-	INT00EN
复位后	0	0	1	0	0	0	未定义	0

注 1: 由 <EMCGx[2:0]> 指定的激活电平取决于中断请求。为了正确设置, 参考表 8-4。

注 2: 上升和下降缘仅当 <EMCGx[2:0]> 设为 "100" 时, <EMSTx> 有效。用作待机复位的激活电平可以通过参考 <EMSTx> 检查。如果中断用 CGICRCG 寄存器清除, <EMSTx> 也被清除。

注 3: 请先规定缘的位, 然后规定 <INTxEN> 位。禁止同时设置它们。

注 4: "0" 从位 31, 23, 15 和 7 读。

注 5: 未定义值从位 25, 17, 9 和 1 读。

(2) CGIMCGB(CG 中断模式控制寄存器 B)

	31	30	29	28	27	26	25	24
位符	-	EMCG07			EMST07		-	INT07EN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
位符	-	EMCG06			EMST06		-	INT06EN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
位符	-	EMCG05			EMST05		-	INT05EN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
位符	-	EMCG04			EMST04		-	INT04EN
复位后	0	0	1	0	0	0	未定义	0

注 1: 由 <EMCGx[2:0]> 指定的激活电平取决于中断请求。为了正确设置, 参考表 8-4。

注 2: 上升和下降缘仅当 <EMCGx[2:0]> 设为 "100" 时, <EMSTx> 有效。用作待机复位的激活电平可以通过参考 <EMSTx> 检查。如果中断用 CGICRCG 寄存器清除, <EMSTx> 也被清除。

注 3: 请先规定缘的位, 然后规定 <INTxEN> 位。禁止同时设置它们。

注 4: "0" 从位 31, 23, 15 和 7 读。

注 5: 未定义值从位 25, 17, 9 和 1 读。

(3) CGIMCGC(CG 中断模式控制寄存器 C)

	31	30	29	28	27	26	25	24
位符	-	EMCG0B			EMST0B		-	INT0BEN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
位符	-	EMCG0A			EMSTA		-	INT0AEN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
位符	-	EMCG09			EMST9		-	INT09EN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
位符	-	EMCG08			EMST08		-	INT08EN
复位后	0	0	1	0	0	0	未定义	0

注 1: 由 <EMCGx[2:0]> 指定的激活电平取决于中断请求。为了正确设置, 参考表 8-4。

注 2: 上升和下降缘仅当 <EMCGx[2:0]> 设为 "100" 时, <EMSTx> 有效。用作待机复位的激活电平可以通过参考 <EMSTx> 检查。如果中断用 CGICRCG 寄存器清除, <EMSTx> 也被清除。

注 3: 请先规定缘的位, 然后规定 <INTxEN> 位。禁止同时设置它们。

注 4: "0" 从位 31, 23, 15 和 7 读。

注 5: 未定义值从位 25, 17, 9 和 1 读。

(4) CGIMCGD(CG 中断模式控制寄存器 D)

	31	30	29	28	27	26	25	24
位符	-	EMCG0F			EMST0F		-	INT0FEN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
位符	-	EMCG0E			EMST0E		-	INT0EEN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
位符	-	EMCG0D			EMST0D		-	INT0DEN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
位符	-	EMCG0C			EMST0C		-	INT0CEN
复位后	0	0	1	0	0	0	未定义	0

注 1: 由 <EMCGx[2:0]> 指定的激活电平取决于中断请求。为了正确设置, 参考表 8-4。

注 2: 上升和下降缘仅当 <EMCGx[2:0]> 设为 "100" 时, <EMSTx> 有效。用作待机复位的激活电平可以通过参考 <EMSTx> 检查。如果中断用 CGICRCG 寄存器清除, <EMSTx> 也被清除。

注 3: 请先规定缘的位, 然后规定 <INTxEN> 位。禁止同时设置它们。

注 4: "0" 从位 31, 23, 15 和 7 读。

注 5: 未定义值从位 25, 17, 9 和 1 读。

8.6.3.2 CGICRCG(CG 中断 请求 清除 寄存器)

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

位	位符	类型	功能
31-5	-	R	读作 0.
4-0	ICRCG[4:0]	W	清除 中断 请求. 0_0000: INT0 0_0100: INT4 0_1000: INT8 0_1100: INTC 0_0001: INT1 0_0101: INT5 0_1001: INT9 0_1101: INTD 0_0010: INT2 0_0110: INT6 0_1010: INTA 0_1110: INTE 0_0011: INT3 0_0111: INT7 0_1011: INTB 0_1111: INTF 1_0000 到 1_1111: 禁止 读作 0.

8.6.3.3 CGRSTFLG (复位 标志 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
上电 复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
上电复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
上电复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	OFDRSTF	DBGRSTF	VLDRSTF	WDTRSTF	PINRSTF	PONRSTF
上电复位后	0	0	0	0	0	0	0	1

位	位符	类型	功能
31-6	-	R	读作 0.
5	OFDRSTF	R/W	OFD 复位 标志 [读] 0: - 1: 从 OFD 复位 [写] 0: 清除 复位 标志 (注 2) 1: 忽略
4	DBGRSTF	R/W	调试 复位 标志 (注 2) [读] 0: - 1: 复位 从 SYSRESETREQ [写] 0: 清除 复位 标志 (注 2) 1: 忽略
3	VLDRSTF	R/W	VLTD 复位 标志 [读] 0: - 1: 从 VLTD 复位 [写] 0: 清除 复位 标志 (注 2) 1: 忽略
2	WDTRSTF	R/W	WDT 复位 标志 [读] 0: - 1: 从 WDT 复位 [写] 0: 清除 复位 标志 (注 2) 1: 忽略
1	PINRSTF	R/W	RESET 引脚 标志 [读] 0: - 1: 从 RESET 引脚 复位. [写] 0: 清除 复位 标志 (注 2) 1: 忽略
0	PONRSTF	R/W	上电 复位 标志 [读] 0: - 1: 从上电 复位 . [写] 0: 清除 复位 标志 (注 2) 1: 忽略

注 1: CPU 由NVIC中应用断产生的复位和设置复位控制寄存器 <SYSRESETREQ> 被显示。

注 2: 此产品有上电复位电路且此寄存器仅由上电复位初始化。因此, 在初始复位状态, 上电后马上把"1"设置到 <PONRSTF> 位。
注意此位不是由第二和后面的复位设置且此寄存器不自动清除。 写入 "0"去清除寄存器。

9. 数字 噪声 过滤 电路 (DNF)

数字 噪声消除电路可以在一定 范围从外部中断引脚输入信号消除噪声.

9.1 配置

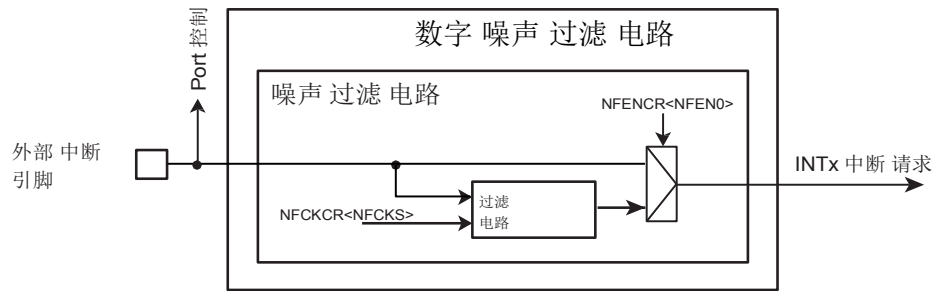


图 9-1 数字噪声过滤电路图

9.2 寄存器

9.2.1 寄存器 表

下表表示控制 寄存器和其地址.

关于基地址详细, 参考 "内存映射" 的"外设功能地址表 " 章节.

寄存器 名		地址(基+)
噪声 过滤 控制 寄存器	NFCKCR	0x0000
噪声 过滤 使能 寄存器	NFENCR	0x0004

9.2.1.1 NFCKCR (噪声 过滤 控制 寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2-0	NFCKS[2:0]	R/W	噪声 过滤 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 1: NFCKCR<NFCKS> 设置指定在 NFENCR<NFEN[15:0]>= 都"0".

注 2: 如果外部输入被使用释放 STOP 模式,噪声过滤电路不可用被使用. 确保禁止 NFENCR 寄存器噪声过滤使能位和通过 NFCKCR 寄存器停止时钟.

9.2.1.2 NFENCR (噪声 过滤 使能 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	NFEN15	NFEN14	NFEN13	NFEN12	NFEN11	NFEN10	NFEN9	NFEN8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	NFEN7	NFEN6	NFEN5	NFEN4	NFEN3	NFEN2	NFEN1	NFEN0
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 "0".
15	NFEN15	R/W	INTF 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
14	NFEN14	R/W	INTE 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
13	NFEN13	R/W	INTD 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
12	NFEN12	R/W	INTC 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
11	NFEN11	R/W	INTB 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
10	NFEN10	R/W	INTA 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
9	NFEN9	R/W	INT9 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
8	NFEN8	R/W	INT8 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
7	NFEN7	R/W	INT7 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
6	NFEN6	R/W	INT6 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)

位	位符	类型	功能
5	NFEN5	R/W	INT5 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
4	NFEN4	R/W	INT4 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
3	NFEN3	R/W	INT3 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
2	NFEN2	R/W	INT2 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
1	NFEN1	R/W	INT1 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)
0	NFEN0	R/W	INT0 噪声 过滤使能. 0: 禁止 (预-噪声过滤输出信号和噪声过滤电路计数器清除时释放 STOP 模式.) 1: 使能 (后-噪声过滤输出信号)

注 1: 一些短于 f_{sys} 的脉冲不能过滤噪声。 尤其, 在 f_{sys} 频率为低的情况下, 噪声过滤器也许没有效。

注 2: 外部中断信号使能前, 清除中断事件然后设置使能的NFENCR 寄存器相应位。

注 3: 如果用外部 输入释放停止模式, 噪声过滤器电路不能使用。 确保禁用 NFENCR 寄存器的噪声过滤器使能位且通过 NFCKCR 寄存器停止时钟。

9.3 工作描述

9.3.1 配置

噪声过滤电路由噪声过滤电路和中断请求生成电路组成。

消除从外部输入的高电平或低电平噪声和 然后 CG 检测上升/下降缘 或信号电平 (高或 低) 去决定各中断信号中信号状态。

9.3.2 工作

噪声过滤器消除外部中断输入 INTx的高和低电平噪声。

噪声过滤时间由 NFCKCR<NFCKS>中指定的输入电平配置时间决定。如果时间小于 7 时钟, 输入被认为噪声。如果时间超过 8 时钟, 输入被认为无效信号。然而, 7 到 8 时钟的输入信号的决定随缘时序变化。

9.3.3 噪声过滤可用工作 模式

仅在正常模式 和 IDLE 模式噪声过滤电路可以被使用。

9.3.4 STOP 模式使用注意事项

如果 STOP 模式被使用, 由于 fsys 时钟的停止, 噪声过滤器电路不能被使用。如果使用外部输入释放 STOP 模式, 设置以下流程: 设置中断使能位为禁用; 设置 NFENCR 寄存器的噪声过滤器使能/禁用位; 且停止 NFCKCR 寄存器的噪声过滤器时钟。

9.3.5 最小噪声过滤时间

噪声过滤器电路决定传输外部中断信号的输入电平, 如果高电平或低电平输入被持续输入超过 NFCKCR 寄存器中指定的 8 时钟周期。

表 9-1 最小 噪声 过滤时间

NFCKCR<NFCKS>	fsys [MHz]			单位
	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	

10. μ DMA 控制器 (μ DMAC)

10.1 总览

10.1.1 功能表

每单元主要功能如下所示:

关于外设 功能开始 触发器信息, 参考"产品信息" 章节.

表 10-1 μ DMA 概要 (每单元)

功能	特征		描述
通道	32 通道		-
开始 触发器	通过 硬件开始		从外设 功能 DMA 请求
	通过 软件开始		通过 DMAxChnlSwRequest 寄存器指定
优先级	通道间	ch0 (高 优先级) > ... > ch31 (高 优先级) > ch0 (正常 优先级) > ... > ch31 (正常 优先级)	高-优先级可以通过 DMAxChnlPrioritySet 寄存器配置
传输 数据 大小	8/16/32 位		-
传输数	1 到 1024 次		-
地址	传输 源 地址	增量 / 固定	传输源地址和目的地地址 可以从增量设置或固定设置选择.
	传输 目的 地 地址	增量 / 固定	
字顺	小-字顺		-
中断 功能	传输 完成 中断		各 单元输出
	错误 中断		
工作 模式	基础 模式 自动 请求 模式 ping-pong 模式 映射 分散/聚集 模式 外设 分散/聚集 模式		-

10.2 方块图

μ DMA 控制器包含以下方块:

- APB 方块
此方块控制控制寄存器进入.
- AHB 方块
此方块控制 DMA 传总线 周期输.
- DMA 控制 方块
此方块控制 DMA 整体工作.
- 中断 控制 方块
此方块集成中断和设置标志寄存器.

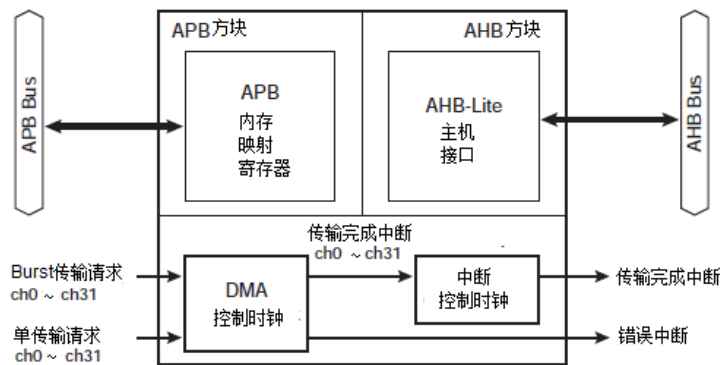


图 10-1 μ DMA 方块图

10.3 寄存器

10.3.1 寄存器表

下表表示控制寄存器和地址:

关于基地址,参考"内存映射章节"中"外设功能基地址表"。

然后外设名 : DMA

寄存器名		地址 (基+)
DMA 状态 寄存器	DMAxStatus	0x0000
DMA 配置 寄存器	DMAxCfg	0x0004
通道 控制 数据 基 指针 寄存器	DMAxCtrlBasePtr	0x0008
通道 交替 控制 数据 基 指针 寄存器	DMAxAltCtlBasePtr	0x000C
通道 软件 请求 状态 寄存器	DMAxChnlSwRequest	0x0014
通道 useburst 设置 寄存器	DMAxChnlUseburstSet	0x0018
通道 useburst 清除 寄存器	DMAxChnlUseburstClr	0x001C
通道 请求 屏蔽 设置 寄存器	DMAxChnlReqMaskSet	0x0020
通道 请求 屏蔽 清除 寄存器	DMAxChnlReqMaskClr	0x0024
通道 使能 设置 寄存器	DMAxChnlEnableSet	0x0028
通道 使能 清除 寄存器	DMAxChnlEnableClr	0x002C
通道 主要-交替 设置 寄存器	DMAxChnlPriAltSet	0x0030
通道 主要-交替 清除 寄存器	DMAxChnlPriAltClr	0x0034
通道 优先级 设置 寄存器	DMAxChnlPrioritySet	0x0038
通道 优先级 清除 寄存器	DMAxChnlPriorityClr	0x003C
总线 错误 清除 寄存器	DMAxErrClr	0x004C

然后外设名: DMAIF

寄存器名		地址 (基+)
标志 寄存器 A	DMAIFFLGA	0x0000

注:以字单元进入寄存器 (32 位).

10.3.2 DMAxStatus (DMAC 状态 寄存器)

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

位	位 符	类型	功能
31-29	-	R	读作 "0".
28	-	R	读作 "1".
27-21	-	R	读作 "0".
20-16	-	R	读作 "1".
15-8	-	R	读作 "0".
7-4	-	R	读作未定义值.
3-1	-	R	读作 "0".
0	master_enable	R	DMA 工作 0: 禁止 1: 使能

10.3.3 DMAxCfg (DMAC 配置 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
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
位符	-	-	-	-	-	-	-	-
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
位符	-	-	-	-	-	-	-	-
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
位符	-	-	-	-	-	-	-	master_ enable
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	位符	类型	功能
31-1	-	W	写作 "0".
0	master_ enable	W	DMA 工作 0: 禁止 1: 使能

注: 设置DMAxCfg = 0x00000001, DMAxChnlReqMaskSet = 0xFFFFFFFF 和 DMAxChnlEnableSet = 0xFFFFFFFF 后, 设置"1" 到 DMAxChnlReqMaskClr 相应位来释放使用的通道的屏蔽。

10.3.4 DMAxCtrlBasePtr (通道 控制 数据 基-指针 寄存器)

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

位	位 符	类型	功能
31-10	ctrl_base_ptr	R/W	主要数据基-指针 指定主要数据基 地址.
9-0	-	R	读作 "0".

10.3.5 DMAxAltCtrlBasePtr (通道 交替 控制 数据 基-指针 寄存器)

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

位	位 符	类型	功能
31-0	alt_ctrl_base_pt	R	交替数据基-指针. 读交替数据基 地址.

10.3.6 DMAxChnlSwRequest (通道 软件 请求 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_sw_re quest (ch31)	chnl_sw_re quest (ch30)	chnl_sw_re quest (ch29)	chnl_sw_re quest (ch28)	chnl_sw_re quest (ch27)	chnl_sw_re quest (ch26)	chnl_sw_re quest (ch25)	chnl_sw_re quest (ch24)
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
位 符	chnl_sw_re quest (ch23)	chnl_sw_re quest (ch22)	chnl_sw_re quest (ch21)	chnl_sw_re quest (ch20)	chnl_sw_re quest (ch19)	chnl_sw_re quest (ch18)	chnl_sw_re quest (ch17)	chnl_sw_re quest (ch16)
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
位 符	chnl_sw_re quest (ch15)	chnl_sw_re quest (ch14)	chnl_sw_re quest (ch13)	chnl_sw_re quest (ch12q)	chnl_sw_re quest (ch11)	chnl_sw_re quest (ch10)	chnl_sw_re quest (ch9)	chnl_sw_re quest (ch8)
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
位 符	chnl_sw_re quest (ch7)	chnl_sw_re quest (ch6)	chnl_sw_re quest (ch5)	chnl_sw_re quest (ch4)	chnl_sw_re quest (ch3)	chnl_sw_re quest (ch2)	chnl_sw_re quest (ch1)	chnl_sw_re quest (ch0)
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	位 符	类型	功能
31-0	chnl_sw_request	W	DMA 请求 0:传输 请求不 发生. 1: 传输 请求 发生. 指定传输请求到各通道.

10.3.7 DMAxChnlUseburstSet (通道 useburst 设置 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_useburst_set (ch31)	chnl_useburst_set (ch30)	chnl_useburst_set (ch29)	chnl_useburst_set (ch28)	chnl_useburst_set (ch27)	chnl_useburst_set (ch26)	chnl_useburst_set (ch25)	chnl_useburst_set (ch24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	chnl_useburst_set (ch23)	chnl_useburst_set (ch22)	chnl_useburst_set (ch21)	chnl_useburst_set (ch20)	chnl_useburst_set (ch19)	chnl_useburst_set (ch18)	chnl_useburst_set (ch17)	chnl_useburst_set (ch16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	chnl_useburst_set (ch15)	chnl_useburst_set (ch14)	chnl_useburst_set (ch13)	chnl_useburst_set (ch12)	chnl_useburst_set (ch11)	chnl_useburst_set (ch10)	chnl_useburst_set (ch9)	chnl_useburst_set (ch8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	chnl_useburst_set (ch7)	chnl_useburst_set (ch6)	chnl_useburst_set (ch5)	chnl_useburst_set (ch4)	chnl_useburst_set (ch3)	chnl_useburst_set (ch2)	chnl_useburst_set (ch1)	chnl_useburst_set (ch0)
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	chnl_useburst_set	R/W	<p>单-传输 禁用</p> <p>[写]</p> <p>1: 单-传输 禁用。</p> <p>[读]</p> <p>0: 单-传输 使能。</p> <p>1: 单-传输 禁用。</p> <p>每个位以指定号对应通道。</p> <p>写 "1" 禁用单-传输到相应 通道, 仅 burst 传输请求变为有效。写 "0" 无意义。设置 DMAxChnlUseburstClr 寄存器来取消禁用的单-传输。</p> <p>通过读此位, 可以检查相应位通道状态是否被使能或禁用。</p> <p>在以下条件, 位自动设置:</p> <ul style="list-style-type: none"> · 此位清为 "0", 如果2^R 次剩余传输的数小于在第二个来自末尾的 2^R 次("R" 由控制数据 channel_cfg<R_power>指定) 的传输。 · 如果在外围 分散/聚集 模式下, 控制数据的 channel_cfg<next_useburst> 被设为"1", 当 DMA 交替的数据的传输及结束时, 此位被设为"1"。

注:不要设置此位为 "1"如果在传输数小于 2^R 次条件下, 不使用 burst 传输请求。

10.3.8 DMAxChnlUseburstClr (通道 useburst 清除 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_useburst _clr (ch31)	chnl_useburst _clr (ch30)	chnl_useburst _clr (ch29)	chnl_useburst _clr (ch28)	chnl_useburst _clr (ch27)	chnl_useburst _clr (ch26)	chnl_useburst _clr (ch25)	chnl_useburst _clr (ch24)
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
位 符	chnl_useburst _clr (ch23)	chnl_useburst _clr (ch22)	chnl_useburst _clr (ch21)	chnl_useburst _clr (ch20)	chnl_useburst _clr (ch19)	chnl_useburst _clr (ch18)	chnl_useburst _clr (ch17)	chnl_useburst _clr (ch16)
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
位 符	chnl_useburst _clr (ch15)	chnl_useburst _clr (ch14)	chnl_useburst _clr (ch13)	chnl_useburst _clr (ch12)	chnl_useburst _clr (ch11)	chnl_useburst _clr (ch10)	chnl_useburst _clr (ch9)	chnl_useburst _clr (ch8)
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
位 符	chnl_useburst _clr (ch7)	chnl_useburst _clr (ch6)	chnl_useburst _clr (ch5)	chnl_useburst _clr (ch4)	chnl_useburst _clr (ch3)	chnl_useburst _clr (ch2)	chnl_useburst _clr (ch1)	chnl_useburst _clr (ch0)
位 符	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	位 符	类型	功能
31-0	chnl_useburst_clr	W	单-传输使能。 1: 使能单-传输。 以指定的数, 各位对应 通道。 写入 "1" 使能对应 通道单-传输. 写入 "0" 无意义.禁止 或确认信号-传输,配置 DMAxChnlUseburstSet 寄存器.

10.3.9 DMAxChnlReqMaskSet (通道 请求 屏蔽 设置 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_req_mas k_set (ch31)	chnl_req_mas k_set (ch30)	chnl_req_mas k_set (ch29)	chnl_req_mas k_set (ch28)	chnl_req_mas k_set (ch27)	chnl_req_mas k_set (ch26)	chnl_req_mas k_set (ch25)	chnl_req_mas k_set (ch24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	chnl_req_mas k_set (ch23)	chnl_req_mas k_set (ch22)	chnl_req_mas k_set (ch21)	chnl_req_mas k_set (ch20)	chnl_req_mas k_set (ch19)	chnl_req_mas k_set (ch18)	chnl_req_mas k_set (ch17)	chnl_req_mas k_set (ch16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	chnl_req_mas k_set (ch15)	chnl_req_mas k_set (ch14)	chnl_req_mas k_set (ch13)	chnl_req_mas k_set (ch12)	chnl_req_mas k_set (ch11)	chnl_req_mas k_set (ch10)	chnl_req_mas k_set (ch9)	chnl_req_mas k_set (ch8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	chnl_req_mas k_set (ch7)	chnl_req_mas k_set (ch6)	chnl_req_mas k_set (ch5)	chnl_req_mas k_set (ch4)	chnl_req_mas k_set (ch3)	chnl_req_mas k_set (ch2)	chnl_req_mas k_set (ch1)	chnl_req_mas k_set (ch0)
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	chnl_req_mask_set	R/W	<p>DMA 请求 屏蔽</p> <p>[写]</p> <p>1: 屏蔽 DMA 请求</p> <p>[读]</p> <p>0: DMA 请求有效.</p> <p>1: DMA 请求无效.</p> <p>以指定的数, 各位对应 通道.</p> <p>写入 "1" 禁止对应 通道单-传输. 写入 "0" 无意义. 禁止屏蔽, 配置 DMAxChnlReqMaskClr 寄存器.</p> <p>通过读此 位, 可以检查 DMA 请求 设置状态是否使能禁止.</p>

10.3.10 DMAxChnlReqMaskClr (通道 请求 屏蔽 清除 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_req_mas k_clr (ch31)	chnl_req_mas k_clr (ch30)	chnl_req_mas k_clr (ch29)	chnl_req_mas k_clr (ch28)	chnl_req_mas k_clr (ch27)	chnl_req_mas k_clr (ch26)	chnl_req_mas k_clr (ch25)	chnl_req_mas k_clr (ch24)
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
位 符	chnl_req_mas k_clr (ch23)	chnl_req_mas k_clr (ch22)	chnl_req_mas k_clr (ch21)	chnl_req_mas k_clr (ch20)	chnl_req_mas k_clr (ch19)	chnl_req_mas k_clr (ch18)	chnl_req_mas k_clr (ch17)	chnl_req_mas k_clr (ch16)
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
位 符	chnl_req_mas k_clr (ch15)	chnl_req_mas k_clr (ch14)	chnl_req_mas k_clr (ch13)	chnl_req_mas k_clr (ch12)	chnl_req_mas k_clr (ch11)	chnl_req_mas k_clr (ch10)	chnl_req_mas k_clr (ch9)	chnl_req_mas k_clr (ch8)
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
位 符	chnl_req_mas k_clr (ch7)	chnl_req_mas k_clr (ch6)	chnl_req_mas k_clr (ch5)	chnl_req_mas k_clr (ch4)	chnl_req_mas k_clr (ch3)	chnl_req_mas k_clr (ch2)	chnl_req_mas k_clr (ch1)	chnl_req_mas k_clr (ch0)
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	位 符	类型	功能
31-0	chnl_req_mask_clr	W	DMA 请求 屏蔽 清除 1: 清除 DMA 请求 屏蔽的对应 通道. 以指定的数, 各位对应 通道. 写入 "1" 禁止对应 通道 DMA 请求 屏蔽 设置. 写入 "0" 无 意义. 配置 DMAxChnlReqMaskSet 寄存器 使能确认设置.

10.3.11 DMAxChnlEnableSet (通道 使能 设置 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_enable_ set (ch31)	chnl_enable_ set (ch30)	chnl_enable_ set (ch29)	chnl_enable_ set (ch28)	chnl_enable_ set (ch27)	chnl_enable_ set (ch26)	chnl_enable_ set (ch25)	chnl_enable_ set (ch24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	chnl_enable_ set (ch23)	chnl_enable_ set (ch22)	chnl_enable_ set (ch21)	chnl_enable_ set (ch20)	chnl_enable_ set (ch19)	chnl_enable_ set (ch18)	chnl_enable_ set (ch17)	chnl_enable_ set (ch16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	chnl_enable_ set (ch15)	chnl_enable_ set (ch14)	chnl_enable_ set (ch13)	chnl_enable_ set (ch12)	chnl_enable_ set (ch11)	chnl_enable_ set (ch10)	chnl_enable_ set (ch9)	chnl_enable_ set (ch8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	chnl_enable_ set (ch7)	chnl_enable_ set (ch6)	chnl_enable_ set (ch5)	chnl_enable_ set (ch4)	chnl_enable_ set (ch3)	chnl_enable_ set (ch2)	chnl_enable_ set (ch1)	chnl_enable_ set (ch0)
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	chnl_enable_set	R/W	<p>DMA 工作</p> <p>[写]</p> <p>1: 使能对应通道.</p> <p>[读]</p> <p>0: 对应位无效.</p> <p>1: 对应位有效.</p> <p>以指定的数, 各位对应 通道.</p> <p>写入 "1" 使能对应通道. 写入 "0" 无意义. 要禁止设置, 配置 DMAxChnlEnableClr 寄存器.</p> <p>通过 读这位, 可以检查对应通道是否使能禁止. 以下条件, 功能自动变为无效.</p> <ul style="list-style-type: none"> • DMA 周期结束. • 如果 channel_cfg<cycle_ctrl> 读"000"控制 数据. • 总线错误发生.

10.3.12 DMAxChnlEnableClr (通道 使能 清除 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_enable_ clr (ch31)	chnl_enable_ clr (ch30)	chnl_enable_ clr (ch29)	chnl_enable_ clr (ch28)	chnl_enable_ clr (ch27)	chnl_enable_ clr (ch26)	chnl_enable_ clr (ch25)	chnl_enable_ clr (ch24)
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
位 符	chnl_enable_ clr (ch23)	chnl_enable_ clr (ch22)	chnl_enable_ clr (ch21)	chnl_enable_ clr (ch20)	chnl_enable_ clr (ch19)	chnl_enable_ clr (ch18)	chnl_enable_ clr (ch17)	chnl_enable_ clr (ch16)
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
位 符	chnl_enable_ clr (ch15)	chnl_enable_ clr (ch14)	chnl_enable_ clr (ch13)	chnl_enable_ clr (ch12)	chnl_enable_ clr (ch11)	chnl_enable_ clr (ch10)	chnl_enable_ clr (ch9)	chnl_enable_ clr (ch8)
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
位 符	chnl_enable_ clr (ch7)	chnl_enable_ clr (ch6)	chnl_enable_ clr (ch5)	chnl_enable_ clr (ch4)	chnl_enable_ clr (ch3)	chnl_enable_ clr (ch2)	chnl_enable_ clr (ch1)	chnl_enable_ clr (ch0)
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	位 符	类型	功能
31-0	chnl_enable_clr	W	<p>DMA 禁止</p> <p>1: 禁止对应通道。 以指定的数，各位对应通道。 写入 "1" 禁止对应通道，写入 "0" 无意义。 配置 DMAxChnlEnableSet 寄存器来使能和确认设置。 以下条件，功能自动变为无效。</p> <ul style="list-style-type: none"> • DMA 周期结束。 • 如果 channel_cfg<cycle_ctrl> 读"000"控制 数据。 • 总线错误发生。

10.3.13 DMAxChnlPriAltSet (通道 主要-交替 设置 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_pri_alt_set (ch31)	chnl_pri_alt_set (ch30)	chnl_pri_alt_set (ch29)	chnl_pri_alt_set (ch28)	chnl_pri_alt_set (ch27)	chnl_pri_alt_set (ch26)	chnl_pri_alt_set (ch25)	chnl_pri_alt_set (ch24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	chnl_pri_alt_set (ch23)	chnl_pri_alt_set (ch22)	chnl_pri_alt_set (ch21)	chnl_pri_alt_set (ch20)	chnl_pri_alt_set (ch19)	chnl_pri_alt_set (ch18)	chnl_pri_alt_set (ch17)	chnl_pri_alt_set (ch16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	chnl_pri_alt_set (ch15)	chnl_pri_alt_set (ch14)	chnl_pri_alt_set (ch13)	chnl_pri_alt_set (ch12)	chnl_pri_alt_set (ch11)	chnl_pri_alt_set (ch10)	chnl_pri_alt_set (ch9)	chnl_pri_alt_set (ch8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	chnl_pri_alt_set (ch7)	chnl_pri_alt_set (ch6)	chnl_pri_alt_set (ch5)	chnl_pri_alt_set (ch4)	chnl_pri_alt_set (ch3)	chnl_pri_alt_set (ch2)	chnl_pri_alt_set (ch1)	chnl_pri_alt_set (ch0)
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	chnl_pri_alt_set	R/W	<p>选择主要数据或交替数据</p> <p>[写]</p> <p>1: 使用交替数据</p> <p>[读]</p> <p>0: 主要数据</p> <p>1: 交替数据</p> <p>以指定的数, 各位对应通道.</p> <p>写入 "1" 指定对应 通道中先被使用的数据为 "交替数据". 写入 "0" 无意义. 禁止此位, 使用 DMAxChnlEnableClr 寄存器.</p> <p>仅在基础模式, 自动请求模式, 和 ping-pong 模式下, 最初数据可以被指定为交替数据.</p> <p>读此位时, 对可以检查应 通道数据是否数据为主要 数据或交替数据.</p> <p>以下条件, 设置自动改变.</p> <ul style="list-style-type: none"> - ping-pong 模式, 内存分散 / 聚集 模式或外设分散 / 聚集 模式下, 主要数据传输完成. - ping-pong 模式, 内存分散 / 聚集 模式或外设分散 / 聚集 模式下, 交替数据的数据传输完成.

10.3.14 DMAxChnlPriAltClr (通道 主要-交替 清除 寄存器)

	31	30	29	28	27	26	25	24
位 符	chn_pri_alt_clr (ch31)	chn_pri_alt_clr (ch30)	chn_pri_alt_clr (ch29)	chn_pri_alt_clr (ch28)	chn_pri_alt_clr (ch27)	chn_pri_alt_clr (ch26)	chn_pri_alt_clr (ch25)	chn_pri_alt_clr (ch24)
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
位 符	chn_pri_alt_clr (ch23)	chn_pri_alt_clr (ch22)	chn_pri_alt_clr (ch21)	chn_pri_alt_clr (ch20)	chn_pri_alt_clr (ch19)	chn_pri_alt_clr (ch18)	chn_pri_alt_clr (ch17)	chn_pri_alt_clr (ch16)
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
位 符	chn_pri_alt_clr (ch15)	chn_pri_alt_clr (ch14)	chn_pri_alt_clr (ch13)	chn_pri_alt_clr (ch12)	chn_pri_alt_clr (ch11)	chn_pri_alt_clr (ch10)	chn_pri_alt_clr (ch9)	chn_pri_alt_clr (ch8)
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
位 符	chn_pri_alt_clr (ch7)	chn_pri_alt_clr (ch6)	chn_pri_alt_clr (ch5)	chn_pri_alt_clr (ch4)	chn_pri_alt_clr (ch3)	chn_pri_alt_clr (ch2)	chn_pri_alt_clr (ch1)	chn_pri_alt_clr (ch0)
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	位 符	类型	功能
31-0	chnl_pri_alt_clr	W	<p>清除交替数据设置。</p> <p>1: 使用主要数据 以指定的数, 各位对应通道。 写入 "1" 设置对应 通道数据为主要数据. 设置 "0" 无效. 配置 DMAxChnlPriAlt 设置寄存器来设置主要数据或确认设置。 以下条件, 设置自动改变.</p> <ul style="list-style-type: none"> - 内存分散/聚集 模式或外设分散/聚集 模式下, 主要数据传输完成. - ping-pong 模式下, 主要数据传输完成. - ping-pong 模式, 内存分散/聚集模式, 或外设分散/聚集模式下, 交替传输完成.

10.3.15 DMAxChnlPrioritySet (通道 优先级 设置 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_priority_ set (ch31)	chnl_priority_ set (ch30)	chnl_priority_ set (ch29)	chnl_priority_ set (ch28)	chnl_priority_ set (ch27)	chnl_priority_ set (ch26)	chnl_priority_ set (ch25)	chnl_priority_ set (ch24)
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	chnl_priority_ set (ch23)	chnl_priority_ set (ch22)	chnl_priority_ set (ch21)	chnl_priority_ set (ch20)	chnl_priority_ set (ch19)	chnl_priority_ set (ch18)	chnl_priority_ set (ch17)	chnl_priority_ set (ch16)
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	chnl_priority_ set (ch15)	chnl_priority_ set (ch14)	chnl_priority_ set (ch13)	chnl_priority_ set (ch12)	chnl_priority_ set (ch11)	chnl_priority_ set (ch10)	chnl_priority_ set (ch9)	chnl_priority_ set (ch8)
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	chnl_priority_ set (ch7)	chnl_priority_ set (ch6)	chnl_priority_ set (ch5)	chnl_priority_ set (ch4)	chnl_priority_ set (ch3)	chnl_priority_ set (ch2)	chnl_priority_ set (ch1)	chnl_priority_ set (ch0)
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	chnl_priority_set	R/W	<p>优先级设置</p> <p>[写]</p> <p>1: 设置高-优先级</p> <p>[读]</p> <p>0: 正常优先级</p> <p>1: 高 优先级</p> <p>以指定的数, 各位对应通道.</p> <p>写入 "1" 设置对应通道优先级高. 写入 "0" 无意义. 再改变优先级为正常, 配置 DMAxChnlPriorityClr 寄存器.</p> <p>对应通道优先级, 高-优先级或正常优先级, 可以通过读此位确认.</p>

10.3.16 DMAxChnlPriorityClr (通道 优先级 清除 寄存器)

	31	30	29	28	27	26	25	24
位 符	chnl_priority_ clr (ch31)	chnl_priority_ clr (ch30)	chnl_priority_ clr (ch29)	chnl_priority_ clr (ch28)	chnl_priority_ clr (ch27)	chnl_priority_ clr (ch26)	chnl_priority_ clr (ch25)	chnl_priority_ clr (ch24)
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
位 符	chnl_priority_ clr (ch23)	chnl_priority_ clr (ch22)	chnl_priority_ clr (ch21)	chnl_priority_ clr (ch20)	chnl_priority_ clr (ch19)	chnl_priority_ clr (ch18)	chnl_priority_ clr (ch17)	chnl_priority_ clr (ch16)
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
位 符	chnl_priority_ clr (ch15)	chnl_priority_ clr (ch14)	chnl_priority_ clr (ch13)	chnl_priority_ clr (ch12)	chnl_priority_ clr (ch11)	chnl_priority_ clr (ch10)	chnl_priority_ clr (ch9)	chnl_priority_ clr (ch8)
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
位 符	chnl_priority_ clr (ch7)	chnl_priority_ clr (ch6)	chnl_priority_ clr (ch5)	chnl_priority_ clr (ch4)	chnl_priority_ clr (ch3)	chnl_priority_ clr (ch2)	chnl_priority_ clr (ch1)	chnl_priority_ clr (ch0)
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	位 符	类型	功能
31-0	chnl_priority_clr	W	清除高-优先级设置。 [写] 1:设置正常优先级设置 以指定的数, 各位对应通道。 写入 "1" 改变对应 通道优先级为正常优先级. 写入 "0" 无意义。配置 DMAxChnlPrioritySet 寄存器来设置高-优先级和确认设置。

10.3.17 DMAxErrClr (总线 错误 清除 寄存器)

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

位	位 符	类型	功能
31-1	-	R	读作 "0".
0	err_clr	R/W	总线错误 [写] 1: 清除总线错误. [读] 0: 无总线错误 1:总线 错误状态 可以通过读此位确认总线错误发生. 写入 "1" 清除总线错误. 写入 "0" 无意义.

10.3.18 DMAIFFLGx (DMA 标志 寄存器)

	31	30	29	28	27	26	25	24
位 符	FLG31	FLG30	FLG29	FLG28	FLG27	FLG26	FLG25	FLG24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	FLG23	FLG22	FLG21	FLG20	FLG19	FLG18	FLG17	FLG16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	FLG15	FLG14	FLG13	FLG12	FLG11	FLG10	FLG9	FLG8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	FLG7	FLG6	FLG5	FLG4	FLG3	FLG2	FLG1	FLG0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	FLG31 - FLG0	R	DMA 因素 标志 0:- 1: 完成中断发生. 用通道数对应位数. 如果传输完成中断发生, 对应位设置为 "1". 通过自动读此寄存器清除.

注: 使能传输完成中断前, 必须读和清除此寄存器.

10.4 工作

DMA 由通道控制数据控制, 位于内存。每个数据通道为四个字且安置于和通道数相同的邻近区域。

有两类通道控制数据: 主要数据和交替的数据。根据工作模式, 其中之一通过设置寄存器选择或数据都使用。

10.4.1 通道 控制 数据 内存映射

图 10-2 表示通道控制数据的内存映射例。

设置主要数据开始地址到 DMAxCtrlBasePtr 和交替数据开始地址到 DMAxAltCtrlBasePtr.

Alternate Ch31	0x3F0	Primary Ch31	0x1F0
Alternate Ch30	0x3E0	Primary Ch30	0x1E0
Alternate Ch29	0x3D0	Primary Ch29	0x1D0
Alternate Ch28	0x3C0	Primary Ch28	0x1C0
Alternate Ch27	0x3B0	Primary Ch27	0x1B0
Alternate Ch26	0x3A0	Primary Ch26	0x1A0
Alternate Ch25	0x390	Primary Ch25	0x190
Alternate Ch24	0x380	Primary Ch24	0x180
Alternate Ch23	0x370	Primary Ch23	0x170
Alternate Ch22	0x360	Primary Ch22	0x160
Alternate Ch21	0x350	Primary Ch21	0x150
Alternate Ch20	0x340	Primary Ch20	0x140
Alternate Ch19	0x330	Primary Ch19	0x130
Alternate Ch18	0x320	Primary Ch18	0x120
Alternate Ch17	0x310	Primary Ch17	0x110
Alternate Ch16	0x300	Primary Ch16	0x100
Alternate Ch15	0x2F0	Primary Ch15	0x0F0
Alternate Ch14	0x2E0	Primary Ch14	0x0E0
Alternate Ch13	0x2D0	Primary Ch13	0x0D0
Alternate Ch12	0x2C0	Primary Ch12	0x0C0
Alternate Ch11	0x2B0	Primary Ch11	0x0B0
Alternate Ch10	0x2A0	Primary Ch10	0x0A0
Alternate Ch9	0x290	Primary Ch9	0x090
Alternate Ch8	0x280	Primary Ch8	0x080
Alternate Ch7	0x270	Primary Ch7	0x070
Alternate Ch6	0x260	Primary Ch6	0x060
Alternate Ch5	0x250	Primary Ch5	0x050
Alternate Ch4	0x240	Primary Ch4	0x040
Alternate Ch3	0x230	Primary Ch3	0x030
Alternate Ch2	0x220	Primary Ch2	0x020
Alternate Ch1	0x210	Primary Ch1	0x010
Alternate Ch0	0x200	Primary Ch0	0x000

保留	0x00C
控制	0x008
目的地结束指针	0x004
源结束指针	0x000

图 10-2 控制 数据内存 映射

图 10-2 显示所有 32 通道可以使用的内存映射。需要区域由可用通道数决定。

表 10-2 显示通道数和地址间的关系。

表 10-2 通道 控制地址位设置

通道	地址						[3:0]	设置表基地址
	[9]	[8]	[7]	[6]	[5]	[4]		
0	-	-	-	-	-	A	通道控制 数据设置	0xFFFF_X00, 0xFFFF_X20, 0xFFFF_X40, 0xFFFF_X60, 0xFFFF_X80, 0xFFFF_XA0, 0xFFFF_XC0, 0xFFFF_XE0
0 到 1	-	-	-	-	A	C[0]		0xFFFF_X00, 0xFFFF_X40, 0xFFFF_X80, 0xFFFF_XC0
0 到 3	-	-	-	A	C[1:0]			0xFFFF_X00, 0xFFFF_X80
0 到 7	-	-	A	C[2:0]				0xFFFF_X00, 0xFFFF_X100, 0xFFFF_X200, 0xFFFF_X300, 0xFFFF_X400, 0xFFFF_X500, 0xFFFF_X600, 0xFFFF_X700, 0xFFFF_X800, 0xFFFF_X900, 0xFFFF_XA00, 0xFFFF_XB00, 0xFFFF_XC00, 0xFFFF_XD00, 0xFFFF_XE00, 0xFFFF_XF00
0 到 15	-	A	C[3:0]					0xFFFF_X00, 0xFFFF_X200, 0xFFFF_X400, 0xFFFF_X600, 0xFFFF_X800, 0xFFFF_XA00, 0xFFFF_XC00, 0xFFFF_XE00
0 到 31	A	C[4:0]						0xFFFF_X00, 0xFFFF_X400, 0xFFFF_X800, 0xFFFF_XC00

A: 主要/交替 设置 (0:主要, 1:交替)

C[x:0]: 通道 数 设置

10.4.2 通道 控制 数据 架构

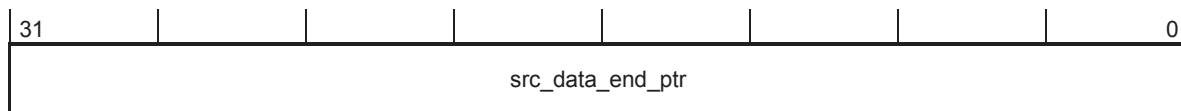
通道控制数据包含三 种数据如下所示:

- 传输 源地址最终地址
- 传输 目的地地址最终地址
- 控制 数据

以下部分描述各数据:

10.4.2.1 传输 源 数据最终 地址

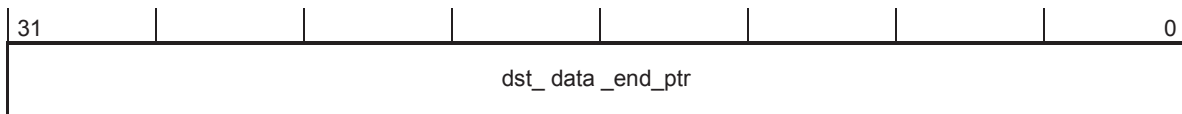
指定要传输的数据的最终地址。地址的对齐应该调节到传输数据大小。DMA 计算使用此数据的源地址开始地址。



位	位符	功能
[31:0]	src_data_end_ptr	源 传输数据最终地址

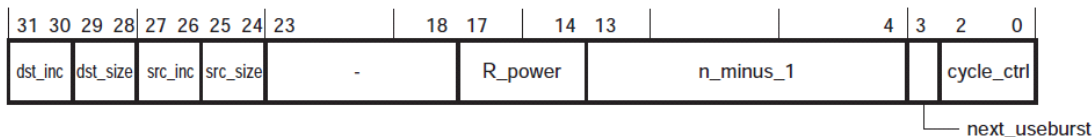
10.4.2.2 传输 目的地 地址最终 地址

指定目的地地址的最终地址。地址的对齐应该调节到传输数据大小。DMA 计算传输目的地地址的目的地地址的开始地址。



位	位符	功能
[31:0]	dst_data_end_ptr	传输 目的地地址最终地址

10.4.2.3 控制 数据 设置



位	位符	功能
[31:30]	dst_inc	增量传输 目的地 地址 (注 2) 00: 1 字节 01: 2 字节 10: 4 字节 11: 无增量
[29:28]	dst_size	传输 目的地数据 大小(注 1) 00: 1 字节 01: 2 字节 10: 4 字节 11: 保留
[27:26]	src_inc	增量传输源 地址 (注 2) 00: 1 字节 01: 2 字节 10: 4 字节 11: 无 增量
[25:24]	src_size	传输源数据 大小 (注 1) 00: 1 字节 01: 2 字节 10: 4 字节 11: 保留
[23:18]	-	设置 "000000".

位	位符	功能
[17:14]	R_power	仲裁 0000: 1 传输后 0001: 2 传输后 0010: 4 传输后 0011: 8 传输后 0100: 16 传输后 0101: 32 传输后 0110: 64 传输后 0111: 128 传输后 1000: 256 传输后 1001: 512 传输后 1010 - 1111: 无仲裁 指定数传输后, 检查传输请求。如果更高优先请求存在, 控制器仲裁 DMA 传输。
[13:4]	n_minus_1	传输数 0x000: 一次 0x001: 两次 0x002: 三次 : 0x3FF: 1024 次
[3]	next_useburst	变化单-传输的设置 0: 不要变化<chnl_useburst_set>值。 1: 设置 <chnl_useburst_set> 到 "1"。 在外围分散/聚集模式下, 规定使用交替的数据的DMA传输结束时, 是否设置"1"到 <chnl_useburst_set> 位。 注) 此位 <chnl_useburst_set> 清零, 如果从末尾在第二次 2 ^R 次传输主要的传输数小于 2 ^R 次 ("R" 由<R_power>指定)。设置此位到 "1" 设置 "1" 到 <chnl_userburst_set>。
[2:0]	cycle_ctrl	工作 模式 000: 无效. DMA 停止工作。 001: 基础 模式 010: 自动请求 模式 011: Ping-pong 模式 100: 内存分散 / 聚集 模式 (主要 数据) 101: 内存分散 / 聚集 模式 (交替 数据) 110: 外设内存分散 / 聚集 模式 (主要 数据) 111: 外设内存分散 / 聚集 模式 (交替 数据)

注 1: <dst_size >设置值必须与 <src_size >相同.

注 2: 根据<dst_size > 和 <src_size >设置, <dst_inc> 和 <src_inc>设置被如下限制:

<src_inc>/<dst_inc>	<src_size >/<dst_size >		
	00 (1 字节)	01 (2 字节)	10 (4 字节)
00 (1 字节)	o	-	-
01 (2 字节)	o	o	-
10 (4 字节)	o	o	o
无 增量	o	o	o

10.4.3 工作 模式

此部分描述通过通道 控制数据的 channel_cfg< cycle_ctrl>配置的工作 模式

10.4.3.1 无效 设置

传输结束后, DMA 设置工作模式无效。此工作防止传输被再次执行。另外, 在 ping-pong 模式, 内存分散 / 聚集模式或外围分散 / 聚集模式下, 如果读到无效数据, 工作完成。

10.4.3.2 基础 模式

基本模式下, 数据结构可用从主要数据或交替的数据选择。

通过收到传输请求传输开始。

对由<R_power>配置的每次传输执行仲裁。如果更高-优先请求存在, DMA切换通道。如果收到工作通道的传输请求, 传输继续。

执行由 <n_minus_1>指定的传输几次后, 传输完成中断产生。

10.4.3.3 自动 请求 模式

此模式下, 单-传输请求停止 DMA 传输。数据结构可以从 主要数据或交替的数据选择。

DMA 传输通过传输请求开始。

每个由<R_power>配置的传输中, 如果收到更高-优先请求, 通道被切换。否则, 传输继续。

执行由 <n_minus_1>指定的传输几次后, 传输完成中断产生。

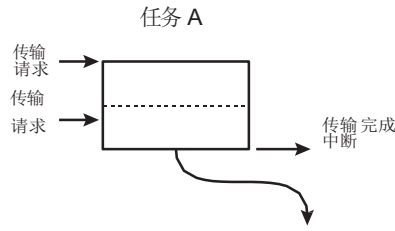
10.4.3.4 Ping-pong 模式

ping-pong 模式下, 交替执行使用主要数据和交替的数据持续的 DMA 传输。如果 <cycle_ctrl> 读指定为无效("000")数据, 或通道指定为无效, 传输停止。每次使用主要数据或交替的数据的DMA 传输 (任务)完成, 传输完成中断产生。

准备: 准备主要数据和交替的数据, 和设置 "1" 到 DMAxCfg<master_enable> 和 DMAxChnlEnable Set相应通道位。

任务 A: 主要 数据

```
<cycle_ctrl[2:0]> = "011"
(ping-pong 模式)
<R_power[3:0]> = "0010"
(4 次)
<n_minus_1[9:0]> =
"0x005" (6 次)
```

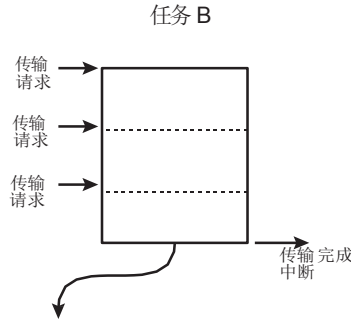


接收传输请求, DMA 执行传输 四次和 执行仲裁.
如果没有别的高-优先级请求, DMA 执行剩余传输 两次 到传输对应通道请求.
DMA 生成传输完成中断请求和执行仲裁.
完成任务 A 后, 任务 C 主要数据可以设置.

任务 B: 交替 数据

```
< cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"

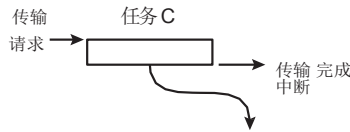
(4 次)
<n_minus_1[9:0]> =
"0x00B" (12 次)
```



接收传输请求, DMA 执行传输四次和执行仲裁.
如果没有别的高-优先级请求, DMA 执行传输 两次请求到 传输对应 通道.
DMA 生成传输完成中断请求和执行仲裁
完成任务后 B, 任务 D 交替数据可以设置.

任务 C: 主要 数据

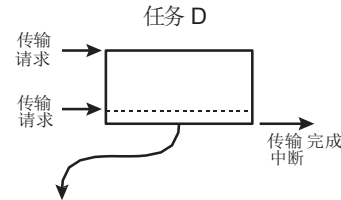
```
< cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0001"
(2 次)
<n_minus_1[9:0]> =
"0x001" (2 次)
```



接收 传输请求,DMA 执行传输 两次和执行仲裁.
DMA 生成 传输完成中断 请求和执行仲裁.
完成任务 C 后, 任务 E 交替数据可以设置.

任务 D: 交替 数据

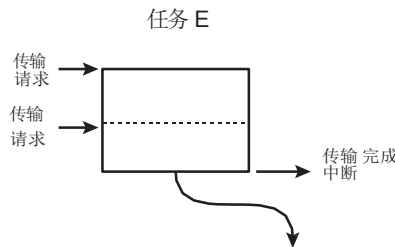
```
< cycle_ctrl[2:0]> = "011"
<R_power[4:0]> = "0010"
(4 次)
<n_minus_1[9:0]> =
"0x004" (5 次)
```



接收传输请求, DMA 执行传输四次和执行仲裁.
如果没有别的高-优先级请求,DMA 执行 传输 一次 请求到对应通道传输.
DMA 生成传输完成中断请求和执行仲裁.

任务 E: 主要 数据

```
< cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 次)
<n_minus_1[9:0]> =
"0x006" (7 次)
```



接收传输请求,DMA 执行传输四次 执行仲裁.
如果没有别的高-优先级请求,DMA 执行传输三次 请求 给对应 通道传输.
DMA 生成传输完成中断 请求和执行仲裁

最终: 交替 数据

```
< cycle_ctrl[2:0]> =
"000" (无效)
```



即使接收传输请求, 工作停止因为< cycle_ctrl[2:0]> 设置为无效.
(工作也可停止通过设置 任务 E < cycle_ctrl[2:0]> 到 正常 模式 "001".)

10.4.3.5 内存 分散/聚集 模式

内存分散/聚集 模式下, 主要数据用来交替数据的传输数据.

收到传输请求, DMA 传输 四次使用主要数据的交替数据。如无新请求, 开始使用交替数据的数据传输。然后, 保持传输使用主要数据的交替数据和传输使用交替的数据, 直到或者 `<cycle_ctrl [2:0]>` 无效设置 ("000") 或者基本模式的设置数据("001") 被读。在此周期期间, 新传输请求不需要。传输工作后, 中断产生。

主要数据 `channel_cfg` 设置必须如下配置:

表 10-3 内存分散/聚集 模式 (主要 数据) 设置值

位	位 符	设置 值	描述
[31:30]	dst_inc	10	指定 4-字节 增量作传输目的地地址.
[29:28]	dst_size	10	指定 4 字节作传输目的地 地址.
[27:26]	src_inc	10	指定 4-字节 增量作传输 源 地址.
[25:24]	src_size	10	指定 4 字节作传输源 地址.
[17:14]	R_power	0010	指定 4 作仲裁周期.
[13:4]	n_minus_1	N	指定准备交替任务数 $\times 4$.
[3]	next_useburst	0	内存分散/聚集 模式下指定"0" .
[2:0]	cycle_ctrl	100	指定内存分散/聚集 模式 (主要 数据). 注)

注: 如果`<n_minus_1>`中指定的传输完成, 无效数据 "000" 自动设置.

准备: 准备 主要 数据. 设置 "100" 到 < cycle_ctrl> 和设置四个任务数据 $4 \times 4 = 16$ 作传输 <n_minus_1>的数.
 设置任务 A,B,C 和 D 交替数据到设置为 <src_data_end_ptr>内存地方.
 设置 "1" 到对应 DMAxCfg <master_en- able> 和 DMAxChnlSet 通道位.

复制 A: 主要 数据

< cycle_ctrl[2:0]> = "100"
 (内存分散 / 聚集 模式)
 <R_power[3:0]> = "0010"
 (4 次)
 <n_minus_1[9:0]> = "0x00F" (16 次)

任务 A: 交替 数据
 < cycle_ctrl[2:0]> = "100"
 <R_power[3:0]> = "0010"
 (4 次)
 <n_minus_1[9:0]> = "0x002" (3 次)

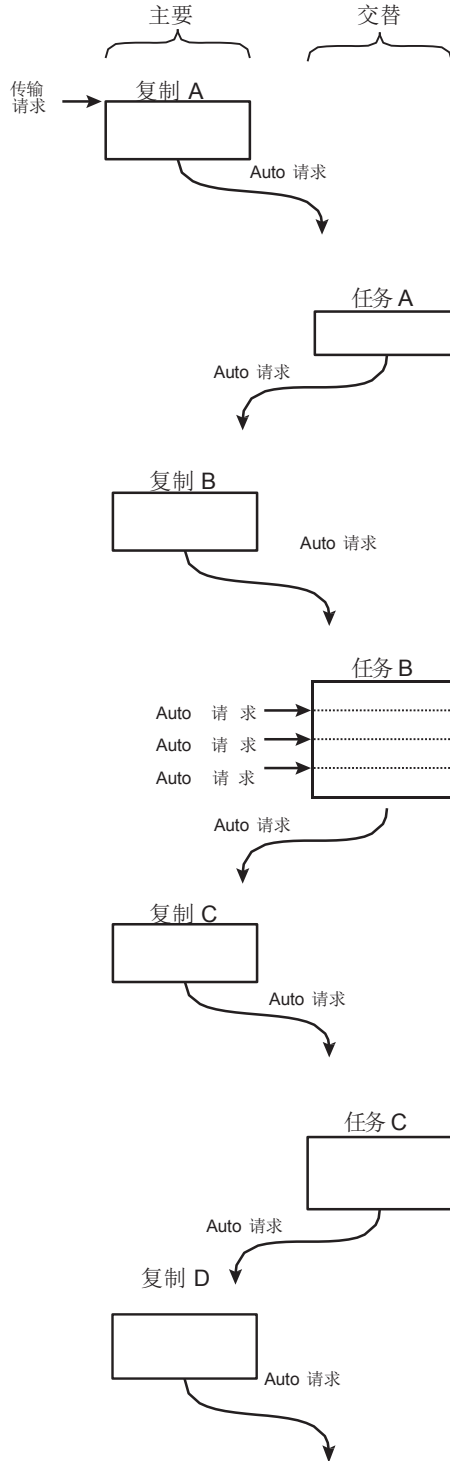
复制 B: 主要 数据

任务 B: 交替 数据
 < cycle_ctrl[2:0]> = "100"
 <R_power[3:0]> = "0001"
 (2 次)
 <n_minus_1[9:0]> = "0x007" (8 次)

复制 C: 主要 数据

任务 C: 交替 数据
 < cycle_ctrl[2:0]> = "100"
 <R_power[3:0]> = "0011"
 (8 次)
 <n_minus_1[9:0]> = "0x004" (5 次)

复制 D: 主要 数据



接收传输请求, DMA 执行任务 A 交替 数据传输四次.
 完成传输后, 传输请求自动生成和仲裁开始.

DMA 执行 任务 A.
 完成传输后, 传输请求自动生成和仲裁开始

DMA 执行任务 B 交替数据传输四次.
 完成传输后, 传输请求自动生成和仲裁开始

DMA 执行任务 B.
 完成传输后, 传输请求自动生成和仲裁开始

DMA 执行任务 C 交替数据传输四次.
 完成传输后, 传输请求自动生成和仲裁开始

DMA 执行 任务 C.
 完成传输后, 传输请求自动生成和仲裁开始.

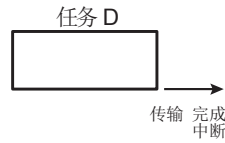
DMA 执行任务 D 交替数据传输四次. DMA 也设置 "000" 到主要数据 < cycle_ctrl> 来设置下一个主要数据无效. 传输 请求自动生成和仲裁开始.

任务 D: 交替 数据

< cycle_ctrl[2:0]> =
"001"

<R_power[3:0]> = "0010"
(4 次)

<n_minus_1[9:0]> =
"0x003" (4 次)



DMA 执行 任务 D.

因为 < cycle_ctrl[2:0]> 设置为 基础模式 "001",
DMA 生成传输完成中断请求, 在传输结束,和完成工
作后.

10.4.3.6 外设 分散/聚集 模式

外设分散/聚集模式下, 主要数据被使用作交替数据的传输数据.

收到传输请求, DMA传输四个使用主要数据的交替数据, 然后开始使用交替数据传输.

然后, 如果传输请求生成, 开始使用主要数据的交替数据传输. 然后,保持传输使用主要数据的交替数据和传输使用交替数据, 直到或者 <cycle_ctrl> 的无效设置 ("000") 或者 基本模式 ("001")的设置数据被读. 此周期期间不需要新传输请求. 传输工作后, 中断生成

主要数据 channel_cfg 设置必须如下配置:

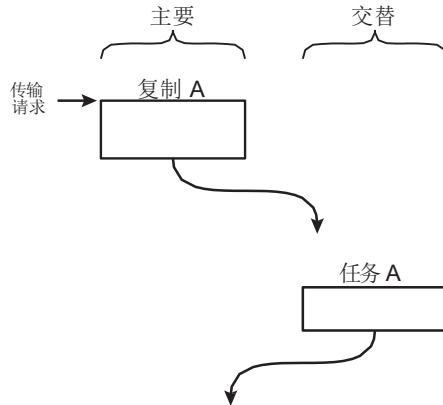
表 10-4 外设 分散 / 聚集 模式下 (主要 数据) 固定 值

位	位 符	设置 值	描述
[31:30]	dst_inc	10	指定 4-字节 增量作传输目的地地址.
[29:28]	dst_size	10	指定 4 字节作传输目的地地址.
[27:26]	src_inc	10	指定 4-字节 增量作传输源地址.
[25:24]	src_size	10	指定 4 字节作传输源地址.
[17:14]	R_power	0010	指定 4 作仲裁周期.
[13:4]	n_minus_1	N	指定准备交替任务数 $\times 4$.
[2:0]	cycle_ctrl	110	指定外设分散/聚集模式 (主要 数据).

注:如果<n_minus_1>中指定的传输完成, 无效数据 "000" 自动设置.

准备: 准备主要数据. 设置 "110" 到 < cycle_ctrl> 和 $4 \times 4 = 16$ 四个任务到传输<n_minus_1>的数.
 设置任务 A,B,C 和 D 交替数据 到设置为 <src_data_end_ptr>的内
 存地方.
 设置 "1" 到对应 DMAxCfg <master_enable> 和
 DMAxChnlEnableSet 通道位.

复制 A: 主要 数据
 < cycle_ctrl[2:0]> = "110"
 (外设 分散 / 聚集)
 <R_power[3:0]> = "0010"
 (4 次)
 <n_minus_1[9:0]> =
 "0x00F" (16 次)

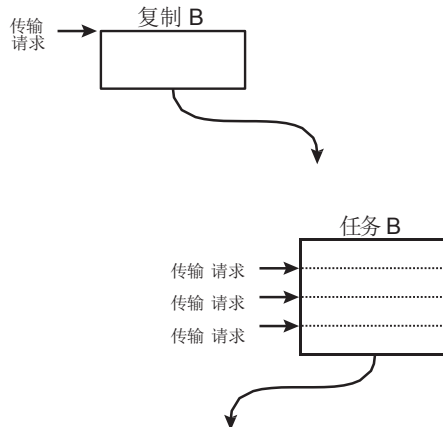


接收传输请求, DMA 执行
 任务 A 交替 数据传输 四次.
 完成传输后, 工作自动移到任务 A.

任务 A: 交替 数据
 < cycle_ctrl[2:0]> = "111"
 <R_power[3:0]> = "0010"
 (4 次)
 <n_minus_1[9:0]> =
 "0x002" (3 次)

DMA 执行任务 A.
 完成传输后, 如果传输请求从外设功能送入和如果是
 高-优先级请求, 下一个工作开始.

复制 B: 主要 数据

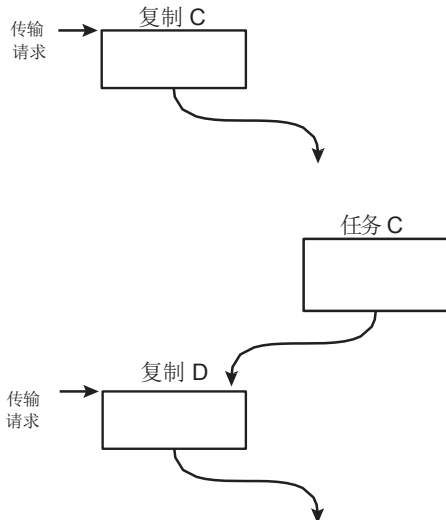


DMA 执行任务 B 交替 数据传输四次.
 完成传输后,任务 B 处理自动开始.

任务 B: 交替 数据
 < cycle_ctrl[2:0]> = "111"
 <R_power[3:0]> = "0001"
 (2 次)
 <n_minus_1[9:0]> =
 "0x007" (8 次)

DMA 执行任务 B. 因为仲裁每传输的 2^8 次发生, 至少
 需要三 次传输完成任务 B.
 完成传输后, 如果传输请求从外设功能被送入和如果
 它是高-优先级 请求, 下一个工作开始

复制 C: 主要 数据



DMA 执行任务 C 交替 数据传输四次.
 完成传输后, 工作自动移到任务 C.

任务 C: 交替 数据
 < cycle_ctrl[2:0]> = "111"
 <R_power[3:0]> = "0011"
 (8 次)
 <n_minus_1[9:0]> =
 "0x004" (5 次)

DMA 执行任务 C.
 完成传输后, 如果传输请求从外设功能送入和如果它
 是高-优先级 请求, 下一个工作开始.

复制 D: 主要 数据

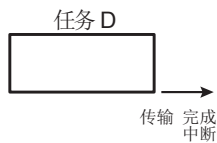
DMA 执行任务 D 交替 数据传输四次. 而且, DMA
 执行交替 数据传输四次. 设置 "000" 到主要数据的<
 cycle_ctrl>和 屏蔽下一个主要 数据 无效.
 工作 自动移到任务 D.

任务 D: 交替 数据

```
< cycle_ctrl[2:0]> = "001"
```

```
<R_power[3:0]> = "0010"
(4 次)
```

```
<n_minus_1[9:0]> = "0x003" (4 次)
```



DMA 执行任务 D.

因为 < cycle_ctrl> 设置为基础模式 "001", DMA 生成传输完成中断请求, 在传输结束, 和完成工作后.

10.5 注意事项

以下外设功能中使用 DMA 传输 请求时需要额外注意:

- 串行 通道 用 4-字节 FIFO (SIO/UART)
- 16-位 定时器/事件 计数器 (TMRB)
- 模拟-到-数字 转换器 (ADC)

10.5.1 使用 SIO/UART, TMRB, ADC

应该考虑以下点:

- 建议使用基础 模式作传输 模式.
- 设置 " 1 传输后"作 DMA 传输率.
指定 "0000" 作控制 数据<R_power>仲裁率.
- 不使用 SIO/UART 的 FIFO.
用单-缓冲或双-缓冲设置使用 SIO/UART.

DMA传输开始后, 新请求产生, 仅执行单传输。小心设计执行 DMA 传输程序。

万一传输不开始, 可以期待以下环境:

- 更高优先级传输请求发生在相同单元
- 别的更高总线主机和传输器间, 传输目的地冲突发生.

作为在指引, 在预-/后-处理中, 此 μ DMA 控制器需要 11 时钟。外围功能和内部 RAM 之间数据传输需要约 5 时钟。

11. 输入 / 输出 端口

此章节描述 端口-相关寄存器, 其设置和电路.

11.1 寄存器

使用端口 寄存器时, 以下寄存器 必须设置.

所有寄存器为 32-位. 配置不同取决于端口位数和功能分配.

以下描述"x" 表示端口名 和 "n" 表示功能数.

寄存器 名		设置 值	
PxDATA	数据 寄存器	0 或 1	此寄存器读 / 写 端口数据.
PxCR	输出 控制 寄存器	0 : 输出 禁止 1 : 输出 使能	此寄存器控制输出.
PxFRn	功能 寄存器 n	0 : 端口 1 : 功能	此寄存器设置功能。 可以通过设置 "1"使能分配功能。此寄存器存在于每个分配给端口的功能。如果有一些功能, 仅可以使能一个功能。
PxOD	开漏 控制 寄存器	0 : CMOS 1 : 开-漏	此寄存器控制可编程开-漏输出. 可编程开-漏输出用 PxOD 设置. 输出数据"1"时, 输出 缓冲禁止和变为假-开-漏输出.
PxPUP	上拉- 控制 寄存器	0 : 上拉禁止 1 : 上拉使能	此寄存器控制可编程上拉.
PxPDN	下拉 控制 寄存器	0 : 下拉禁止 1 : 下拉使能	此寄存器控制可编程下拉.
PxIE	输入 控制 寄存器	0 : 输入禁止 1 : 输入使能	此寄存器控制输入。 使能 PxIE 后需要一些时间直到外部数据反映到 PxDATA 中。

11.1.1 寄存器表

关于基地址详细, 参考"内存映射"章节"外设功能地址表".

寄存器名	地址 (基+)	端口 A	端口 B	端口 C	端口 D	端口 E
数据寄存器	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA
输出控制寄存器	0x0004	PACR	PBCR	PCCR	PDCR	PECR
功能寄存器 1	0x0008	PAFR1	PBFR1	PCFR1	PDFR1	PEFR1
功能寄存器 2	0x000C	PAFR2	-	-	PDFR2	PEFR2
开漏控制寄存器	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD
上拉控制寄存器	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP
下拉控制寄存器	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN
输入控制寄存器	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE

寄存器名	地址 (基+)	端口 F	端口 G	端口 H	端口 J	端口 K
数据寄存器	0x0000	PFDATA	PGDATA	PHDATA	PJDATA	PKDATA
输出控制寄存器	0x0004	PFCR	PGCR	PHCR	PJCR	PKCR
功能寄存器 1	0x0008	PFFR1	PGFR1	-	-	PKFR1
功能寄存器 2	0x000C	PFFR2	-	-	-	-
功能寄存器 3	0x0010	PFFR3	-	-	-	-
开漏控制寄存器	0x0028	PFOD	PGOD	PHOD	PJOD	PKOD
上拉控制寄存器	0x002C	PFPUP	PGPUP	PHPUP	PJPUP	PKPUP
下拉控制寄存器	0x0030	PFPDN	PGPDN	PHPDN	PJPDN	PKPDN
输入控制寄存器	0x0038	PFIE	PGIE	PHIE	PJIE	PKIE

寄存器名	地址 (基+)	端口 L	端口 N	端口 P
数据寄存器	0x0000	PLDATA	PNDATA	PPDATA
输出控制寄存器	0x0004	PLCR	PNCR	PPCR
功能寄存器 1	0x0008	PLFR1	PNFR1	-
功能寄存器 2	0x000C	-	PNFR2	-
开漏控制寄存器	0x0028	PLOD	PNOD	PPOD
上拉控制寄存器	0x002C	PLPUP	PNPUP	PPPUP
下拉控制寄存器	0x0030	PLPDN	PNPDN	PPPDN
输入控制寄存器	0x0038	PLIE	PNIE	PPIE

注: 表示为"-"的地址不进入.

11.1.2 端口 功能和设置 表

各端口功能和设置寄存器表如下所示.

- "表 11-1 端口 A 设置 表"
- "表 11-2 端口 B 设置 表"
- "表 11-3 端口 C 设置 表"
- "表 11-4 端口 D 设置 表"
- "表 11-5 端口 E 设置 表"
- "表 11-6 端口 F 设置 表"
- "表 11-7 端口 G 设置 表"
- "表 11-8 端口 H 设置 表"
- "表 11-9 端口 J 设置 表"
- "表 11-10 端口 K 设置 表"
- "表 11-11 端口 L 设置 表"
- "表 11-12 端口 N 设置 表"
- "表 11-13 端口 P 设置 表"

PxFRn单元显示必须去设置来选择功能的功能寄存器。 如果此寄存器设为 "1", 相应功能使能。

充满填充命令的单元中的一位读作 "0" 且写 数据到此位无效。

表"0" 或 "1" 显示设到寄存器的值。"0/1" 显示选择值可以设到寄存器。

一些功能输入 / 输出引脚分配到一些端口。 仅一个端口可以用功能寄存器分配。

Pxm: P (端口) + 端口名 "x" 和其寄存器位 "m"。 例如, PA0 代表端口名 A和其寄存器的位 0。

11.1.2.1 端口 A

表 11-1 端口 A 设置表

PORT	复位 状态	输入/输出	端口 类型	控制 寄存器						
				PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INT3	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB0IN	输入	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
PA1	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB0OUT	输出	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	PA2	复位后			0	0	0	0	0	0
输入 端口		输入		0/1	0	0	0/1	0/1	0/1	1
输出 端口		输出		0/1	1	0	0/1	0/1	0/1	0
INT4		输入	FT4	0/1	0	0	0/1	0/1	0/1	1
TB1IN		输入	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
PA3	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB1OUT	输出	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
PA4	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC1SCLK	输入	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
		输出		0/1	1	PAFR1	0/1	0/1	0/1	0
SCICTS	输入	FT2	0/1	0	PAFR2	0/1	0/1	0/1	1	
PA5	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC1TXD	输出	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	TB6OUT	输出	FT1	0/1	1	PAFR2	0/1	0/1	0/1	0
PA6	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC1RXD	输入	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	TB6IN	输入	FT1	0/1	0	PAFR2	0/1	0/1	0/1	1
PA7	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INT8	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB4IN	输入	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1

11.1.2.2 端口 B

表 11-2 端口 B 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TRACECLK0	输出	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB1	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TRACEDATA0	输出	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB2	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TRACEDATA1	输出	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB3	复位后 (TSM/SWDIO)			0	1	PBFR1	0	1	0	1
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TMS	I/O	FT3	0/1	1	PBFR1	0/1	0/1	0/1	1
	SWDIO	I/O	FT3	0/1	1	PBFR1	0/1	0/1	0/1	1
PB4	复位后 (TCK/SWCLK)			0	0	PBFR1	0	0	1	1
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TCK	输入	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
	SWCLK	输入	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
PB5	复位后 (TDO/SWV)			0	1	PBFR1	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TDO	输出	FT3	0/1	1	PBFR1	0/1	0/1	0/1	0
	SWV	输出	FT3	0/1	1	PBFR1	0/1	0/1	0/1	0
PB6	复位后 (TDI)			0	0	PBFR1	0	1	0	1
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TDI	输入	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
PB7	复位后 $\overline{\text{TRST}}$			0	0	PBFR1	0	1	0	1
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	$\overline{\text{TRST}}$	输入	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1

11.1.2.3 端口 C

表 11-3 端口 C 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE
PC0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	UO0	输出	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	XO0	输出	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	VO0	输出	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	YO0	输出	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	WO0	输出	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ZO0	输出	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC6	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	EMG0	输入	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1
PC7	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	OW0	输入	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1

11.1.2.4 端口 D

表 11-4 端口 D 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE
PD0	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ENCA0	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5IN	输入	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
PD1	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ENCB0	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5OUT	输出	FT1	0/1	1	PDFR2	0/1	0/1	0/1	0
PD2	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ENCZ0	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
PD3	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INT9	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
PD4	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC2SCLK	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
		输出		0/1	1	PDFR1	0/1	0/1	0/1	0
SC2CTS	输入	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1	
PD5	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC2TXD	输出	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
PD6	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC2RXD	输入	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1

11.1.2.5 端口 E

表 11-5 端口 E 设置表

PORT	复位 状态	输入/输出	端口 类型	控制 寄存器						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC0TXD	输出	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
	TB8OUT	输出	FT1	0/1	1	PEFR2	0/1	0/1	0/1	0
PE1	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC0RXD	输入	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
	TB8IN	输入	FT1	0/1	0	PEFR2	0/1	0/1	0/1	1
PE2	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SC0SCLK	输入	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
		输出		0/1	1	PEFR1	0/1	0/1	0/1	0
	SC0CTS	输入	FT1	0/1	0	PEFR2	0/1	0/1	0/1	1
PE3	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB4OUT	输出	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE4	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INT5	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB2IN	输入	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE5	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB2OUT	输出	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE6	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INT6	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB3IN	输入	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE7	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INT7	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB3OUT	输出	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0

11.1.2.6 端口 F

表 11-6 端口 F 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PFDATA	PF0CR	PFFRn	PF0D	PFPUP	PFPDN	PFIE
PF0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB7IN	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
PF1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB7OUT	输出	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ENCA1	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3SCLK	输入	FT1	0/1	0	PFFR2	0/1	0/1	0/1	1
		输出		0/1	1	PFFR2	0/1	0/1	0/1	0
SC3CTS	输入	FT1	0/1	0	PFFR3	0/1	0/1	0/1	1	
PF3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ENCB1	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3TXD	输出	FT1	0/1	1	PFFR2	0/1	0/1	0/1	0
PF4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ENCZ1	输入	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3RXD	输入	FT1	0/1	0	PFFR2	0/1	0/1	0/1	1

注: PF0 工作为BOOT功能。当复位引脚为 "低"时, 它使能去输入和上拉。在复位信号的上升缘, 如果 PF0为 "高", 设备进入单芯片模式且从片上闪存启动。如果PF0为 "低", 设备进入单BOOT模式和来自内部

11.1.2.7 端口 G

表 11-7 端口 G 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PG0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	UO1	输出	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	XO1	输出	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	VO1	输出	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG3	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	YO1	输出	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG4	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	WO1	输出	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG5	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	ZO1	输出	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG6	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	EMGi	输入	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
PG7	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	OW1	输入	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1

11.1.2.8 端口 H

表 11-8 端口 H 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PHDATA	PHCR	PHFRn	PHOD	PHPUP	PHPDN	PHIE
PH0	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA0	输入	FT5	0/1	0		0/1	0	0	0
	INT0	输入	FT4	0/1	0		0/1	0	0/1	1
PH1	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA1	输入	FT5	0/1	0		0/1	0	0	0
	INT1	输入	FT4	0/1	0		0/1	0	0/1	1
PH2	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA2	输入	FT5	0/1	0		0/1	0	0	0
	INT2	输入	FT4	0/1	0		0/1	0	0/1	1
PH3	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA3	输入	FT5	0/1	0		0/1	0	0	0
PH4	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA4	输入	FT5	0/1	0		0/1	0	0	0
PH5	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA5	输入	FT5	0/1	0		0/1	0	0	0
PH6	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA6	输入	FT5	0/1	0		0/1	0	0	0
PH7	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINA7	输入	FT5	0/1	0		0/1	0	0	0

11.1.2.9 端口 J

表 11-9 端口 J 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ0	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB3	输入	FT5	0/1	0		0/1	0	0	0
PJ1	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB4	输入	FT5	0/1	0		0/1	0	0	0
PJ2	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB5	输入	FT5	0/1	0		0/1	0	0	0
PJ3	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB6	输入	FT5	0/1	0		0/1	0	0	0
PJ4	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB7	输入	FT5	0/1	0		0/1	0	0	0
PJ5	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB8	输入	FT5	0/1	0		0/1	0	0	0
PJ6	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB9	输入	FT5	0/1	0		0/1	0	0	0
	INTC	输入	FT4	0/1	0		0/1	0/1	0/1	1
PJ7	复位后			0	0		0	0	0	0
	输入端口	输入		0	0		0/1	0/1	0/1	1
	输出端口	输出		0	1		0/1	0/1	0/1	0
	AINB10	输入	FT5	0/1	0		0/1	0	0	0
	INTD	输入	FT4	0/1	0		0/1	0/1	0/1	1

11.1.2.10 端口 K

表 11-10 端口 K 设置表

PO RT	复位状态	输入/输出	端口 类型	控制 寄存器						
				PKDATA	PKCR	PKFRn	PKOD	PKPUP	PKPDN	PKIE
PK0	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INTE	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TRACEDATA3	输出	FT1	0/1	1	PKFR1	0/1	0/1	0/1	0
PK1	复位后			0	0	0	0	0	0	0
	输入 端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出 端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INTF	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	TRACEDATA2	输出	FT1	0/1	1	PKFR1	0/1	0/1	0/1	0

11.1.2.11 端口 L

表 11-11 端口 L 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE
PL0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INTB	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	CA_TX	输出	FT1	0/1	1	PLFR1	0/1	0/1	0/1	0
PL1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	INTA	输入	FT4	0/1	0	0	0/1	0/1	0/1	1
	CA_RX	输入	FT1	0/1	0	PLFR1	0/1	0/1	0/1	1

11.1.2.12 端口 N

表 11-12 端口 N 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PNDATA	PNCR	PNFRn	PNOD	PNPUP	PNPDN	PNIE
PN0	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	SB0SDA	I/O	FT1	0/1	1	PNFR2	0/1	0/1	0/1	1
PN1	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB9OUT	输出		0/1	1	PNFR1	0/1	0/1	0/1	0
	SB0SCL	I/O	FT1	0/1	1	PNFR2	0/1	0/1	0/1	1
PN2	复位后			0	0	0	0	0	0	0
	输入端口	输入		0/1	0	0	0/1	0/1	0/1	1
	输出端口	输出		0/1	1	0	0/1	0/1	0/1	0
	TB9IN	输入	FT1	0/1	0	PNFR1	0/1	0/1	0/1	1

11.1.2.13 端口 P

表 11-13 端口 P 设置表

PORT	复位状态	输入/输出	端口类型	控制寄存器						
				PPDATA	PPCR	PPFRn	PPOD	PPPUP	PPPDN	PPIE
PP0	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AINA8	输入	FT5	0/1	0		0/1	0	0	0
PP1	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AINA9	输入	FT5	0/1	0		0/1	0	0	0
	AINB0	输入	FT5	0/1	0		0/1	0	0	0
PP2	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AINA10	输入	FT5	0/1	0		0/1	0	0	0
	AINB1	输入	FT5	0/1	0		0/1	0	0	0
PP3	复位后			0	0		0	0	0	0
	输入端口	输入		0/1	0		0/1	0/1	0/1	1
	输出端口	输出		0/1	1		0/1	0/1	0/1	0
	AINA11	输入	FT5	0/1	0		0/1	0	0	0
	AINB2	输入	FT5	0/1	0		0/1	0	0	0

11.2 端口方块图

端口如下归类。每个端口类型方块图请参考下页。

点线表示"端口方块图"中描述的等价电路部分。

当冷-复位产生或当STOP2 模式由复位引脚释放时，电路图中所示的"直接复位" 工作被使能。

11.2.1 类型 FT1

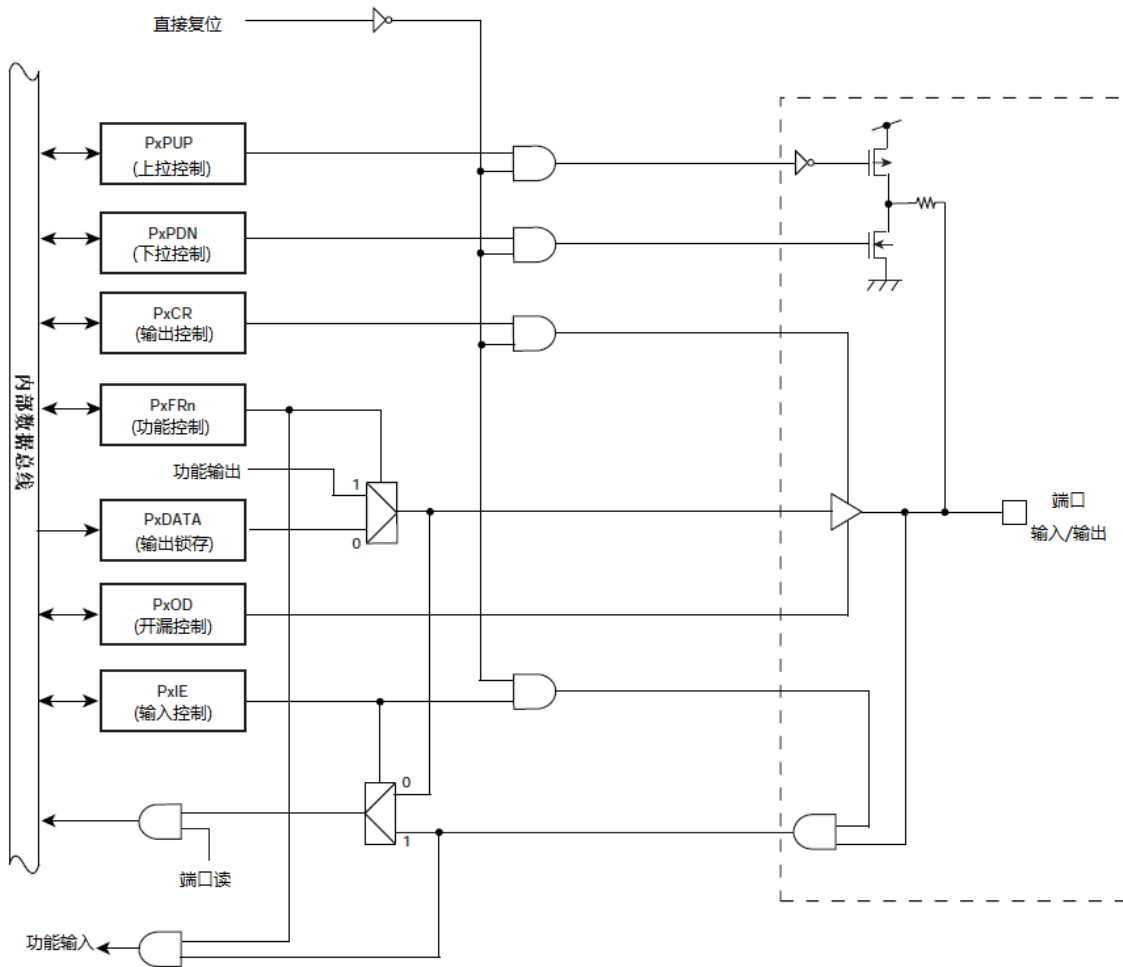


图 11-1 端口 类型 FT1

11.2.2 类型 FT2

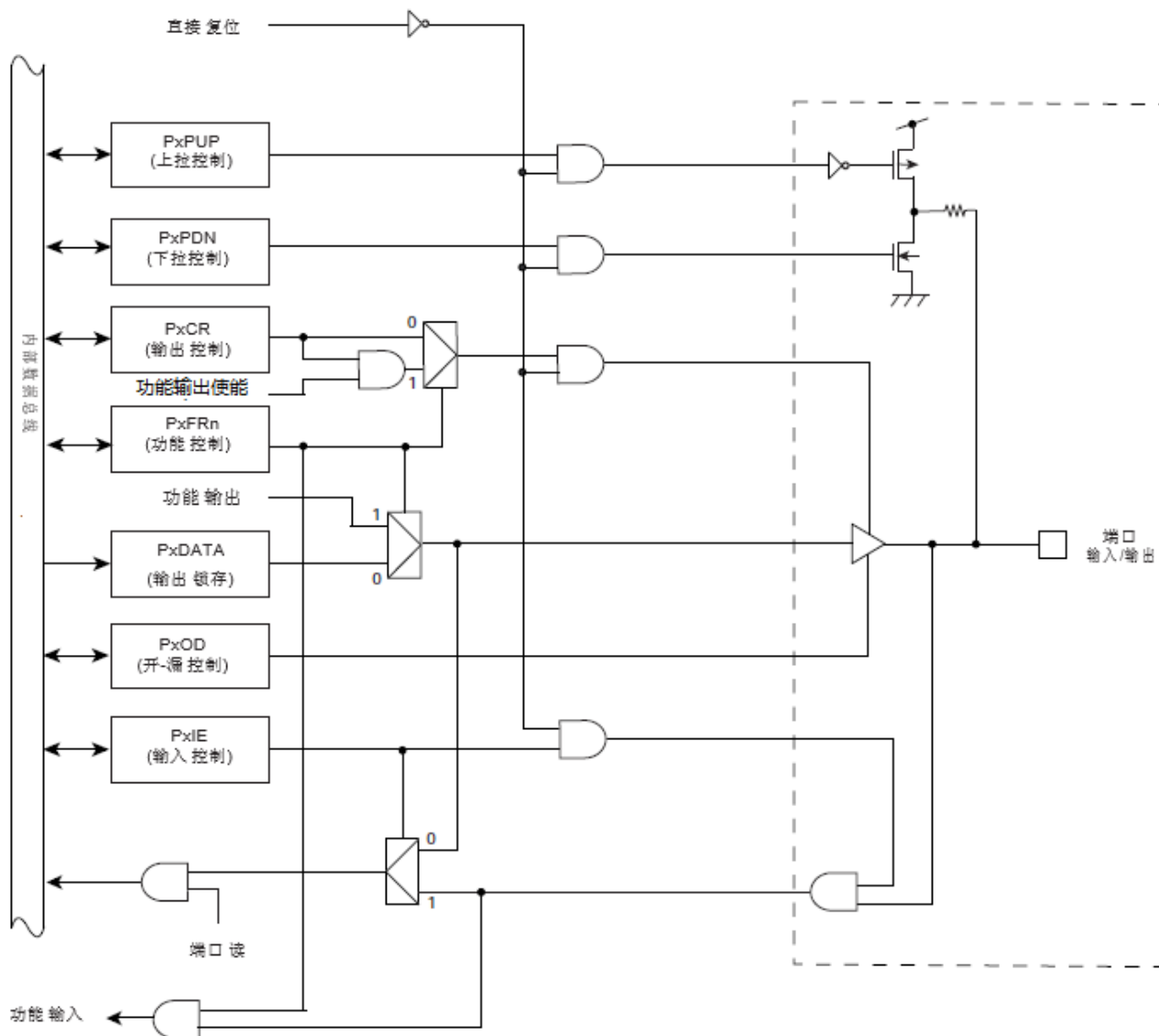


图 11-2 端口 类型 FT2

11.2.3 类型 FT3

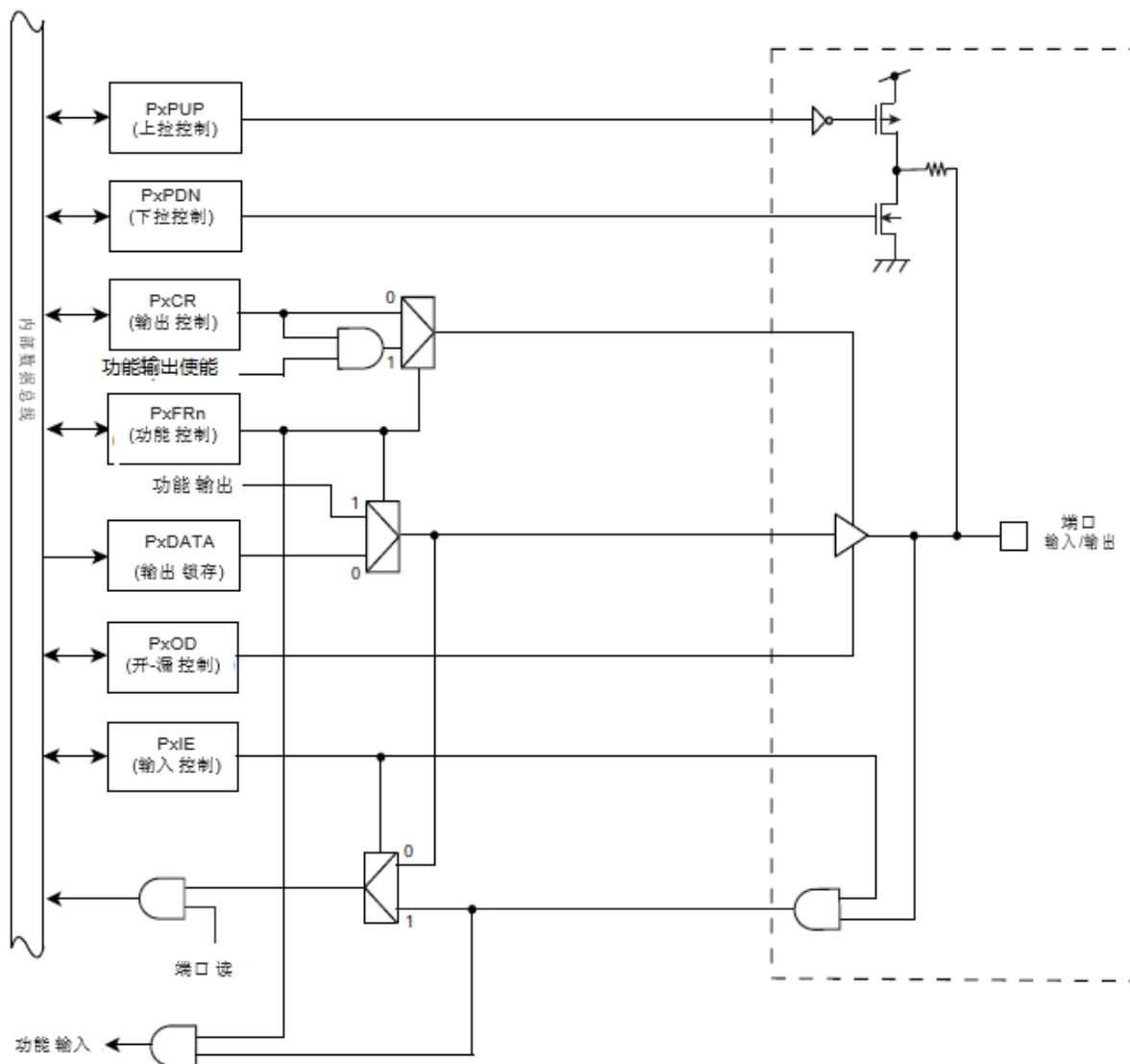


图 11-3 端口 类型 FT3

11.2.4 类型 FT4

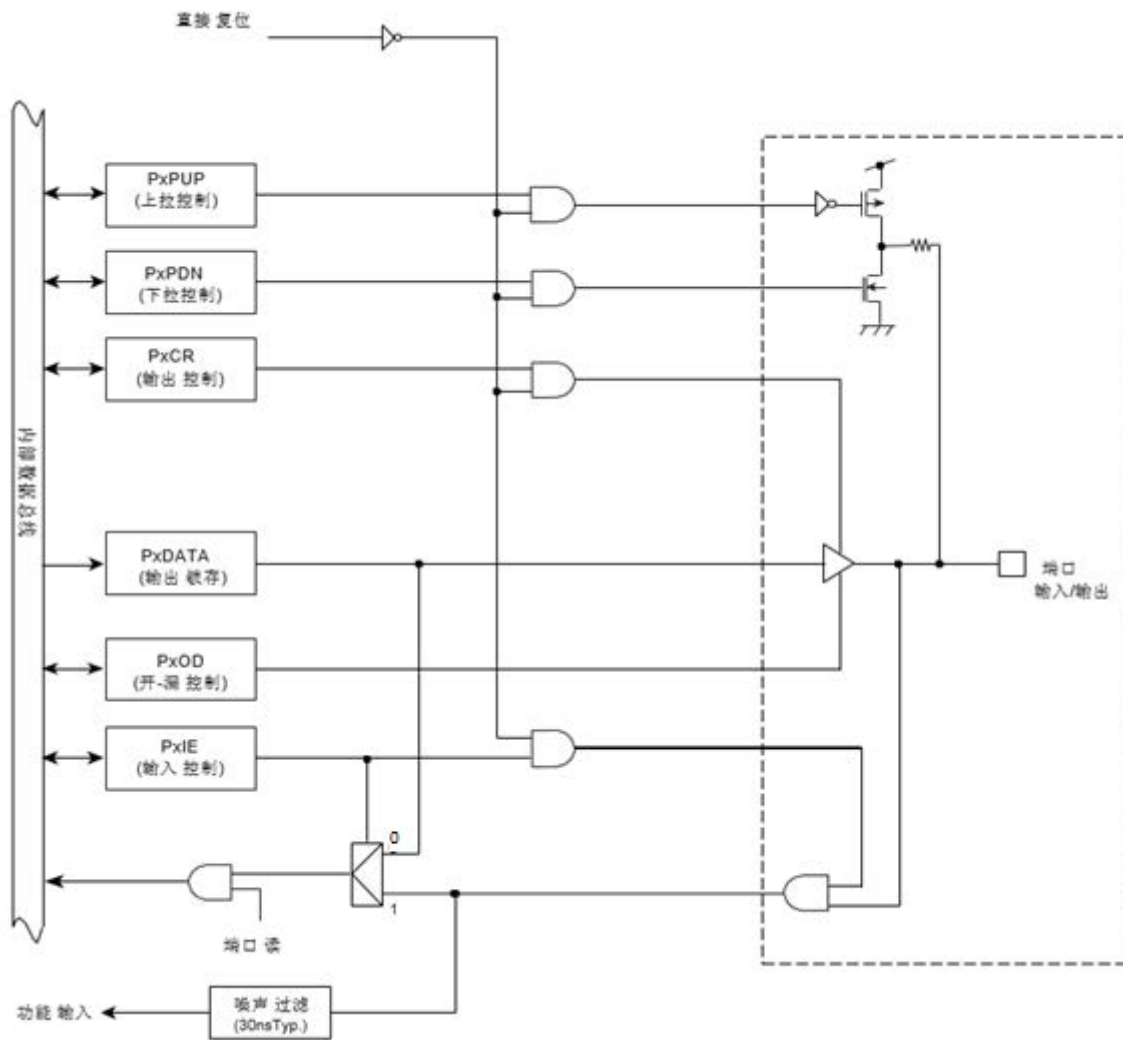


图 11-4 端口 类型 FT4

11.2.5 类型 FT5

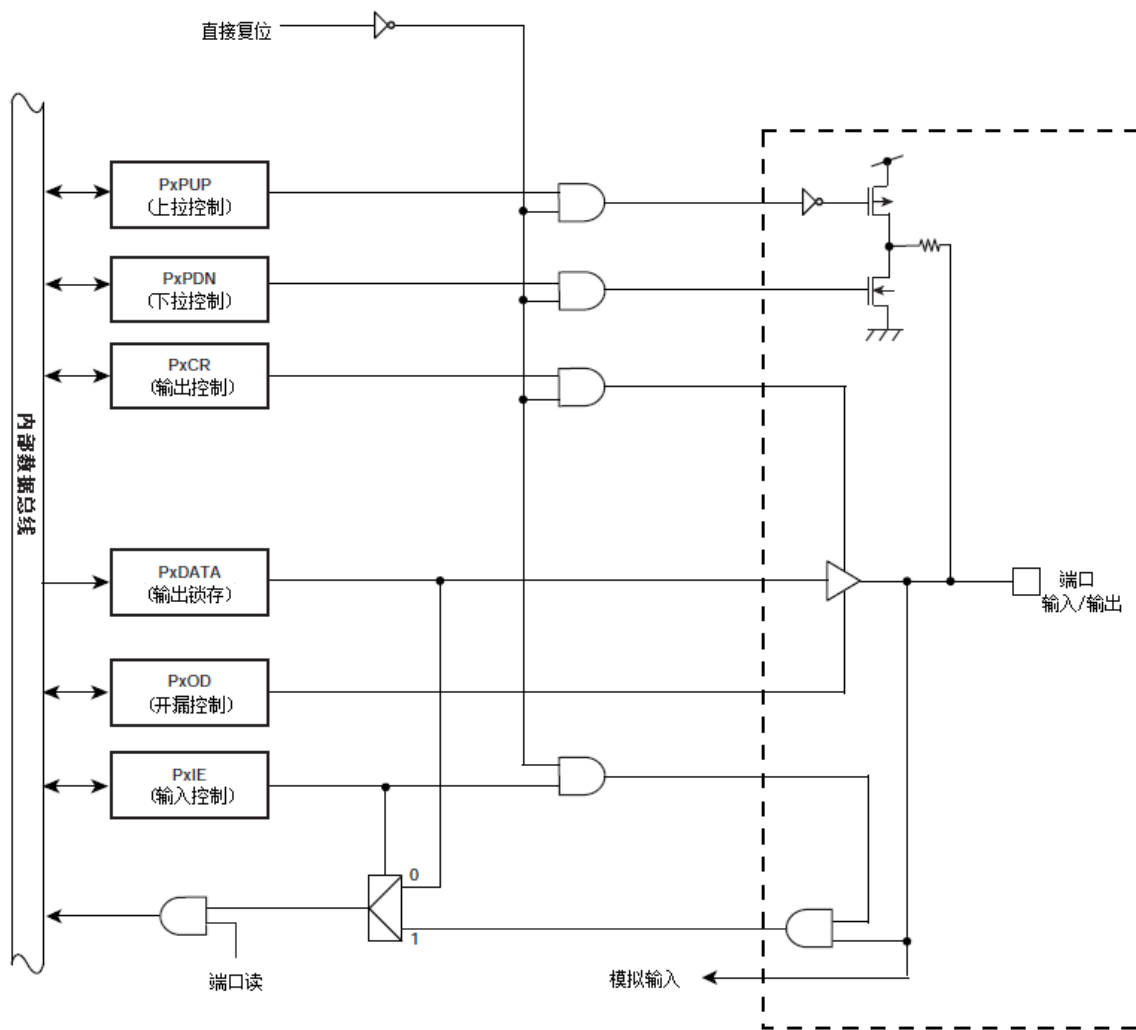


图 11-5 端口 类型 FT5

11.2.6 类型 FT6

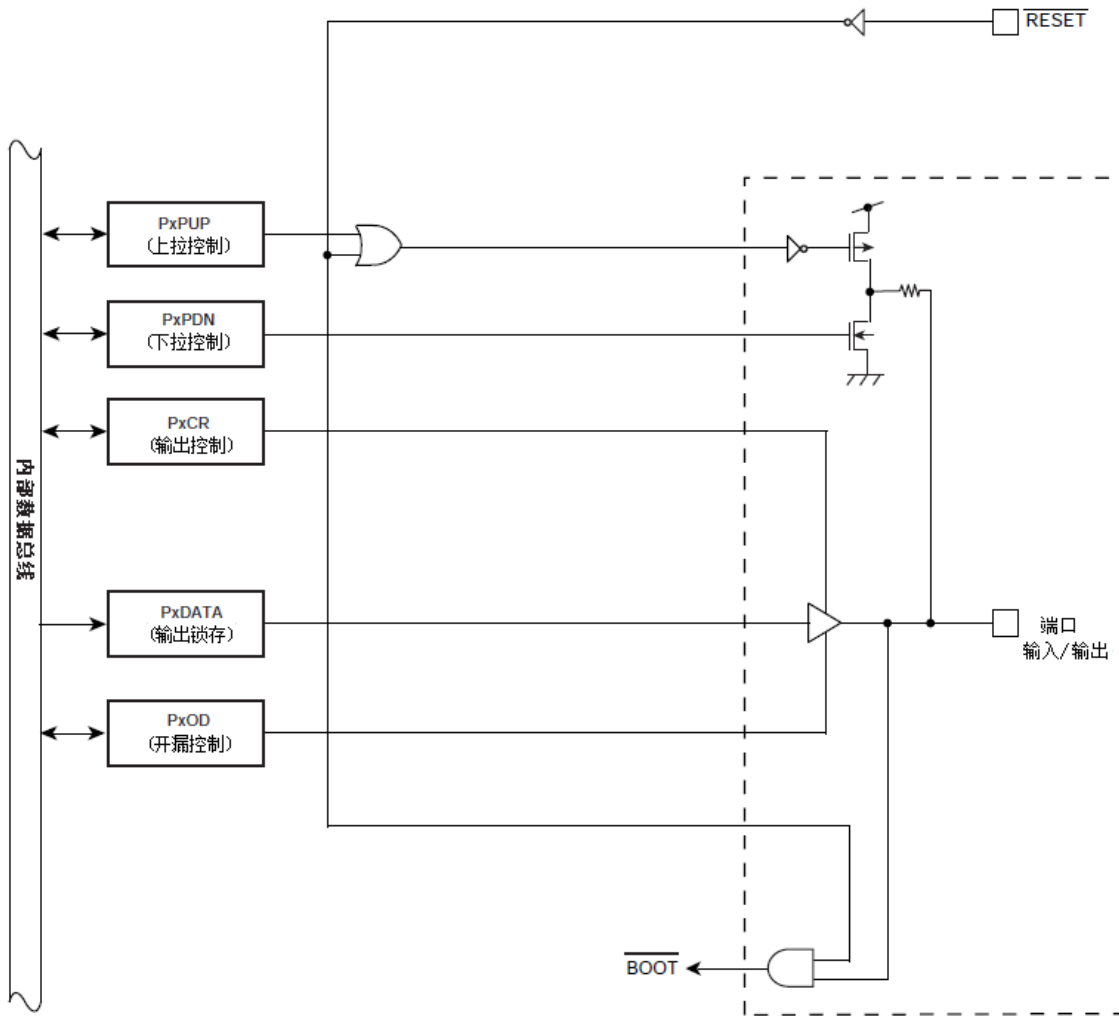


图 11-6 端口 类型 FT6

11.2.7 类型 FT7

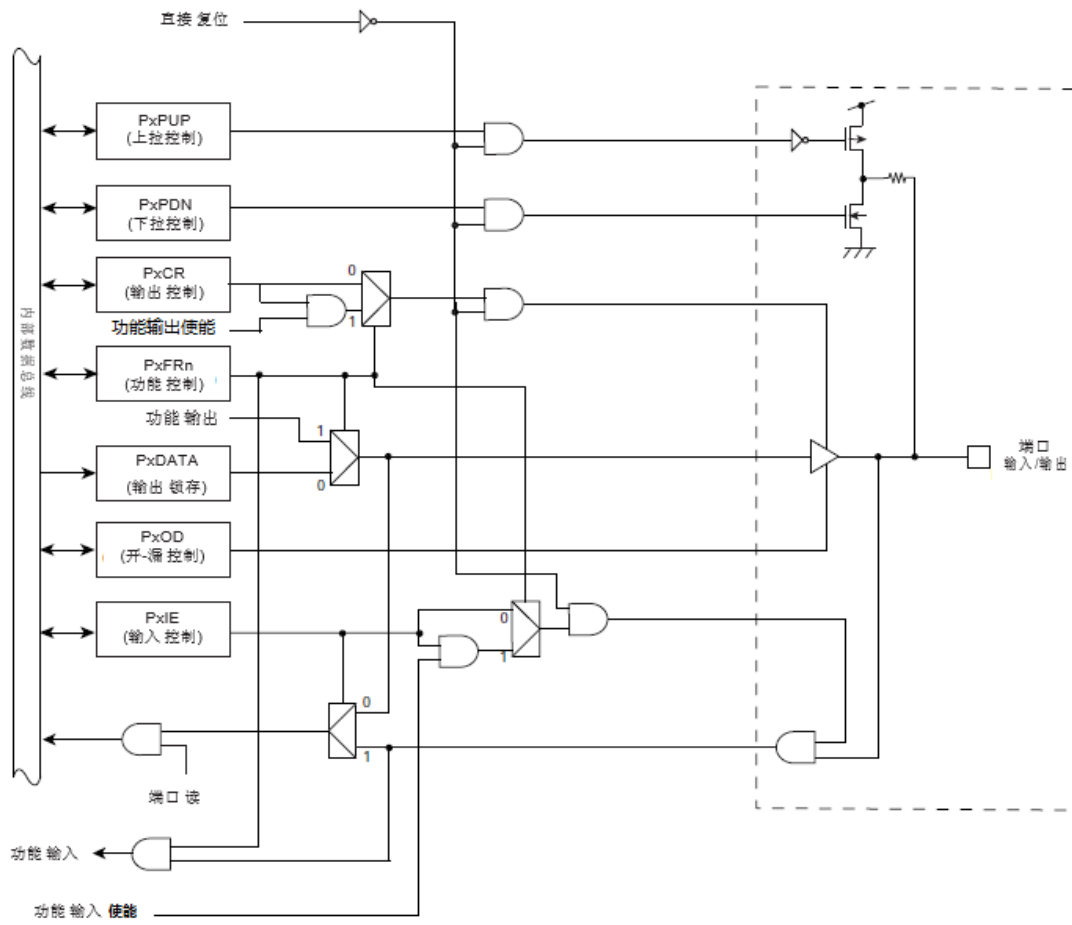


图 11-7 端口 类型 FT7

11.2.8 类型 FT8

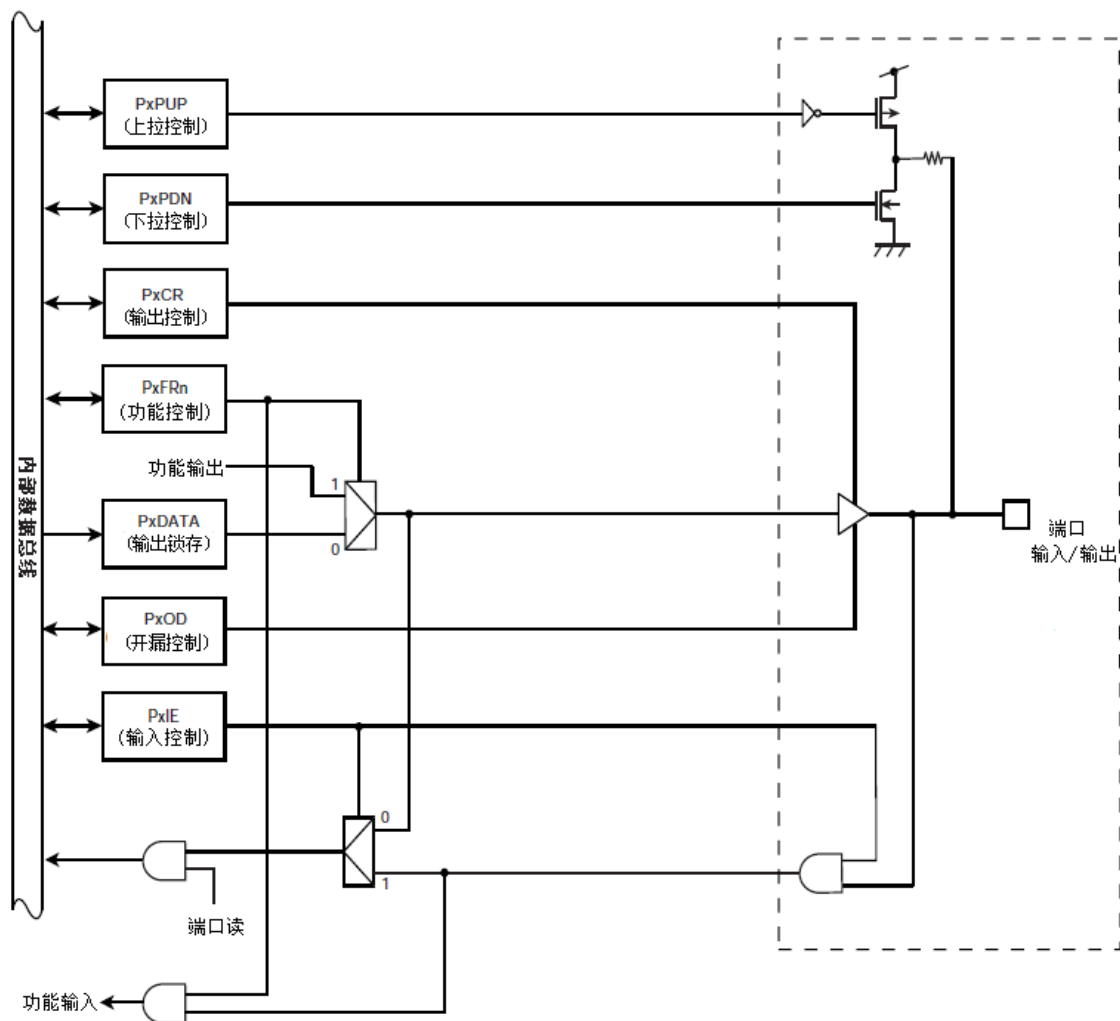


图 11-8 端口 类型 FT8

12. 16-位 定时器 / 事件 计数器 (TMRB)

12.1 概要

TMRB 以下四个工作模式中工作:

- 16-位 间隔 定时器 模式
- 16-位 事件 计数器 模式
- 16-位 可编程 脉冲 生成 模式 (PPG)
- 外部触发器可编程 脉冲 生成 模式 (PPG)

捕捉功能的使用允许 TMRB 执行以下两个测量.

- 一个发脉冲输出通过外部触发器
- 脉冲宽测量

此部分以下说明, "x" 指通道数.

12.2 配置

各通道由 16-位 递增-计数器组成, 两个 16-位 定时器 寄存器 (双-缓冲), 两个 16-位 捕捉寄存器, 两个比较器, 捕捉输入控制, 定时器 flip-flop 和其相关控制电路. 定时器工作模式和定时器 flip-flop 通过寄存器控制.

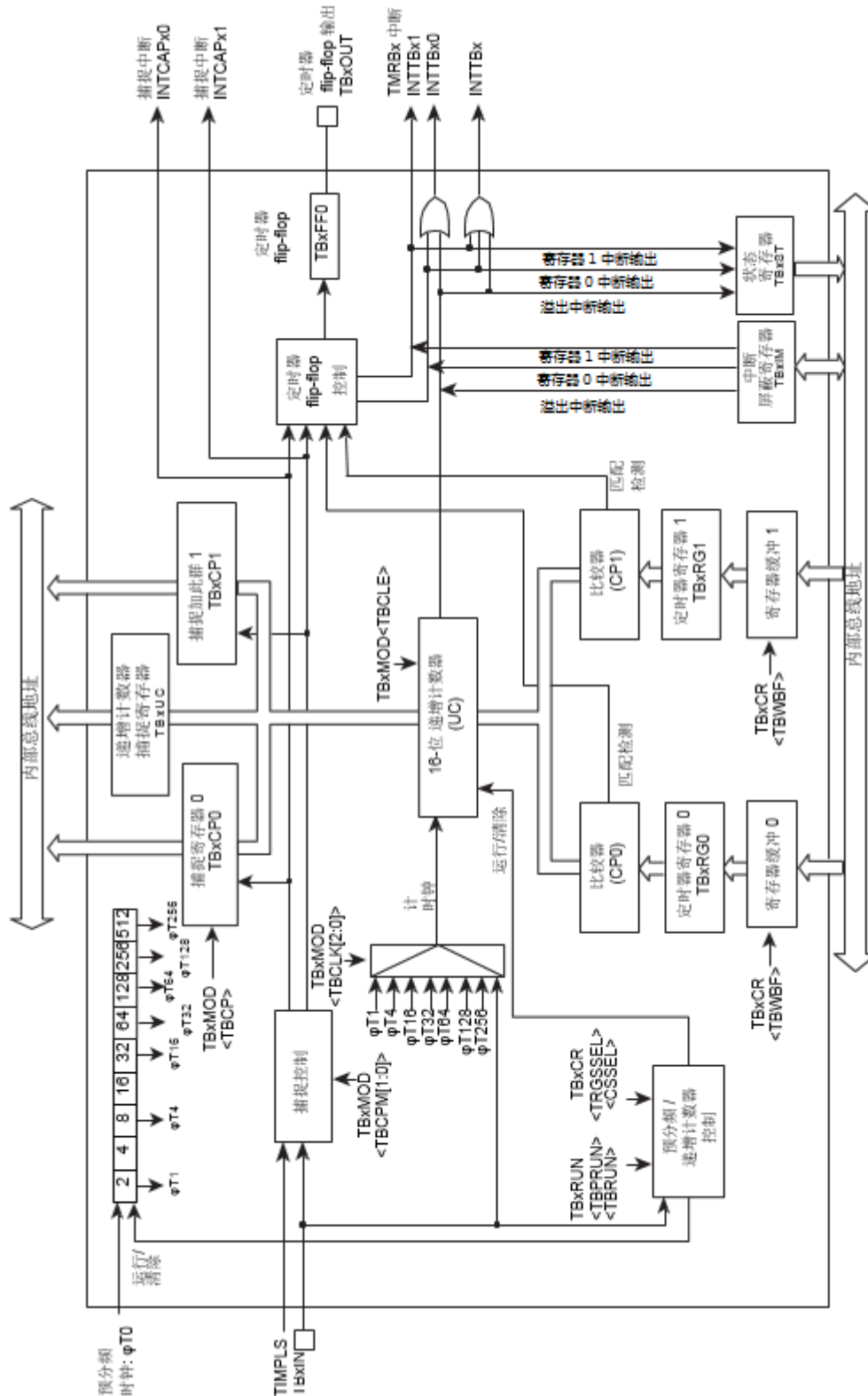


图 12-1 TMRBx 方块图

12.3 寄存器

12.3.1 根据通道寄存器 表

如下表示控制寄存器和其地址.

关于基地址详细, 参考"内存 映射"章节"外设 功能地址表".

寄存器名		地址(基+)
使能 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
Flip-flop 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

12.3.2 TBxEN(使能 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	TBEN	R/W	TMRBx 工作 0: 禁用 1: 使能 规定 TMRBx 工作。当工作禁用, 无时钟提高到TMRBx 模组里别的寄存器。此能减少功耗。(这禁止从其他寄存器读和写, 除TBxEN 寄存器外。) 要使用 TMRBx, 在编程 TMRBx 模组里每个寄存器前, 使能 TMRBx 工作 (设置"1")。如果 TMRBx 工作执行且后 禁用, 设置将维持在每个寄存器。
6	TBHALT	R/W	时钟工作期间调试 HALT. 0: 运行 1: 停止
5-0	-	R	读作 "0".

12.3.3 TBxRUN (RUN 寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2	TBPRUN	R/W	预分频 工作 0: 停止 & 清除 1: 计数
1	-	R	读作 "0".
0	TBRUN	R/W	计数 工作 0: 停止 & 清除 1: 计数

注 1: 当外部触发器开始使用 (<SSEL>=1), 在<TBRUN>=<TBPRUN>=1设置前, 选择 <CSSEL> 和 <TRGSEL>。

注 2: 当计数器停止 (<TBRUN>="0") 且 TBxUC<TBUC[15:0]> 被读, 读取当计数器工作时, 被捕捉的值。

12.3.4 TBxCR(控制 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	TBWBF	R/W	双 缓冲 0: 禁止 1: 使能
6	-	R/W	写作 "0".
5	TBSYNC	R/W	同步 模式 开关 0: 独立 (各 通道) 1: 同步
4	-	R	读作 "0".
3	I2TB	R/W	工作在 IDLE 模式 0: 停止 1: 工作
2	-	R/W	写作 "0".
1	TRGSEL	R/W	外部 触发器 选择 0: 上升 缘 1: 下降 缘
0	CSSEL	R/W	计数器 开始 选择 0: 软件 开始 1: 外部 触发器

注 1: TMRB 工作期间不要修改 TBxCR.

注 2: 外部触发器开始被使用 (<SSEL>=1) 时,设置<TBRUN>=<TBPRUN>=1 前, 选择 <CSSEL> 和 <TRGSEL>.

12.3.5 TBxMOD(模式 寄存器)

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7	TBRSWR	R/W	写入定时器 寄存器 0 和 1 (双 缓冲使能时) 0: 通过对应递增-计数器 (UC), 无论寄存器 0 和 1 的重写, 数据传输到定时器寄存器 0 和 1 完成. 1: 要传输缓冲 寄存器数据到定时器寄存器, 需要一起写入定时器 寄存器 0 和 1.
6	TBCP	W	捕捉 控制 通过 软件 0: 捕捉 通过 软件 TB 1: 忽略 写入 "0" 时, 捕捉寄存器 0 (TBxCP0) 接受计数值. 读作 "1".
5-4	TBCPM[1:0]	R/W	捕捉 时序 00: 禁止 捕捉 时序 01: TBxIN↑ 在 TBxIN 引脚 输入 上升时, 接受计数值到捕捉寄存器 0 (TBxCP0). 10: TBxIN↑ TBxIN↓ 在 TBxIN 引脚 输入 上升时, 接受计数值到捕捉寄存器 0 (TBxCP0). 在下降 TBxIN 引脚 输入时, 接受计数值到捕捉 寄存器 1 (TBxCP1). 11: TIMPLS↑ TIMPLS↓ 在 TIMPLS 输入 上升时, 接受计数值到捕捉 寄存器 0 (TBxCP0). 在 TIMPLS 输入 下降 时, 接受计数值到捕捉寄存器 1 (TBxCP1).
3	TBCLE	R/W	递增-计数器 控制 0: 禁止 清除递增-计数器 1: 使能 清除递增-计数器. 清除和控制递增-计数器. "0" 写入时, 禁止清除递增-计数器. "1" 写入时, 匹配定时器 寄存器 1 (TBxRG1) 时, 清除递增计数器.
2-0	TBCLK[2:0]	R/W	选择 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: 定时器工作时, 不要改变 TBxMOD 寄存器 .

12.3.6 TBxFFCR(Flip-flop 控制 寄存器)

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7-6	-	R	读作 "1".
5	TBC1T1	R/W	递增-计数器值接受到 TBxCP 1 时, TBxFF0 反转触发器. 0: 禁止 触发器 1: 使能 触发器 递增-计数器值接受到捕捉寄存器 1 (TBxCP1) 时, 通过设置 "1", 定时器-flip-flop 反转.
4	TBC0T1	R/W	递增-计数器值接受到 TBxCP0 时, TBxFF0 反转触发器. 0: 禁止 触发器 1: 使能 触发器 递增-计数器值接受到捕捉寄存器 0 (TBxCP0) 时, 通过设置 "1", 定时器-flip-flop 反转.
3	TBE1T1	R/W	递增-计数器 值匹配 TBxRG 1 时, TBxFF0 反转触发器. 0: 禁止 触发器 1: 使能 触发器 递增-计数器值匹配定时器寄存器 1 (TBxRG1) 时, 通过设置 "1", 定时器-flip-flop 反转.
2	TBE0T1	R/W	递增-计数器 值匹配 TBxRG0 时, TBxFF0 反转触发器. 0: 禁止 触发器 1: 使能 触发器 递增-计数器值匹配定时器 寄存器 0 (TBxRG0) 时, 通过 设置 "1", 定时器-flip-flop 反转.
1-0	TBFF0C[1:0]	R/W	TBxFF0 控制 00: 反转 反转 TBxFF0 值 (反转通过使用软件). 01: 设置 设置 TBxFF0 到 "1". 10: 清除 清除 TBxFF0 到 "0". 11: 忽略 * 此始终读作 "11".

注: 定时器工作时不要改变 TBxFFCR 寄存器.

12.3.7 TBxST(状态 寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2	INTTBOF	R	溢出标志 0: 无溢出发生 1: 溢出发生 递增-计数器溢出时, 设置"1".
1	INTTB1	R	匹配标志 (TBxRG1) 0: 无匹配检测 1: 用 TBxRG1 检测匹配 用定时器寄存器 1 (TBxRG1) 检测匹配时, 设置"1".
0	INTTB0	R	匹配标志 (TBxRG0) 0: 无匹配检测 1: 用 TBxRG0 检测匹配 用定时器寄存器 0 (TBxRG0) 检测匹配时, 设置"1".

注 1: 只有没有被TBxIM 的因素输出中断请求给 CPU。即使屏蔽设置已做, 标志被设置。

注 2: 通过读 TBxST 寄存器, 清除标志。要清除标志, 应该读 TBxST 寄存器。

12.3.8 TBxIM(中断 屏蔽 寄存器)

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

位	位 符	类型	功能
31-3	-	R	读作 "0".
2	TBIMOF	R/W	溢出 中断 屏蔽 0: 禁止 1: 使能 设置递增-计数器溢出中断来禁止或使能.
1	TBIM1	R/W	匹配 中断 屏蔽 (TBxRG1) 0: 禁止 1: 使能 用定时器寄存器 1 (TBxRG1) 设置匹配中断屏蔽来使能或禁止.
0	TBIM0	R/W	匹配 中断 屏蔽 (TBxRG0) 0: 禁止 1: 使能 用定时器寄存器 0 (TBxRG0)设置匹配中断屏蔽来使能或禁止.

注:即使屏蔽通过 TBxIM 寄存器配置有效,状态设置到 TBxST 寄存器.

12.3.9 TBxUC(递增 计数器 捕捉 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	TBUC[15:0]	R	通过读出递增-计数器, 捕捉值. 如果读 TBxUC, 可以捕捉当前递增-计数器值.

注:计数器工作且读 TBxUC 时, 读最后一次读到的 TBxUC 捕捉值, 可以捕捉当前递增-计数器值.

12.3.10 TBxRG0(定时器 寄存器 0)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	TBRG0[15:0]	R/W	设置比较递增-计数器的值.

12.3.11 TBxRG1(定时器 寄存器 1)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	TBRG1[15:0]	R/W	设置比较递增-计数器的值.

12.3.12 TBxCP0(捕捉 寄存器 0)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	TBCP0[15:0]	R	读从递增-计数器捕捉的值.

12.3.13 TBxCP1(捕捉 寄存器 1)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	TBCP1[15:0]	R	读从递增-计数器捕捉的值.

12.4 各电路工作描述

12.4.1 预分频

有4-位给递增-计数器 UC生成源时钟的预分频器。

预分频器输入时钟 ϕ_T 为由CG 中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, 或者为fgear, 一个由CG中CGSYSCR<FPSEL>选择的时钟, 或是fc, 一个由时钟齿轮分割前的时钟。

工作或停止预分频器 由TBxRUN<TBPRUN>设置, 写 "1" 开始计数和写 "0"清除和停止计数。

12.4.2 递增-计数器 (UC)

UC 为 16-位 二进制 计数器。

12.4.2.1 源 时钟

递增-计数器的源时钟通过 TBxMOD<TBCLK[2:0]> 指定。

可以从预分频 输出时钟 - ϕ_{T1} , ϕ_{T4} , ϕ_{T16} , ϕ_{T32} , ϕ_{T64} , ϕ_{T128} 和 ϕ_{T256} – 或 TBxIN 引脚的外部时钟选择。

12.4.2.2 计数器 开始 / 停止

有软件开始, 外部触发器开始和同步开始来开始计数器。

1. 软件开始

如果 TBxRUN<TBRUN> 被设置"1", 计数器将开始。如果 "0"被设置 <TBRUN>, 计数器将停止和递增-计数器同时将被清除。

2. 外部触发器开始

外部触发器模式下,计数器通过外部 信号开始。

如果 TBxCR<CSSEL>设置到"1",设置外部触发器开始模式. 此时, 如果 <TBRUN > 设置到 "1",计数器条件为触发器等待.计数器在 TBxIN0TBxIN 上升/下降缘开始.

TBxCR<TRGSEL> 位 指定切换外部触发缘.

- <TRGSEL> = "0":选择 TBxIN 上升缘.
- <TRGSEL> = "1":选择 TBxIN 下降缘 .

如果 <TBRUN>设置到 "0", 计数器停止且递增-计数器同时清除。

3. 同步 开始

定时器同步模式下, 同步开始定时器可能. PPG 输出模式下, 如果定时器同步模式被使用, 可以实现电机驱动应用。

根据产品, 已经决定主机通道和从通道的组合. 关于此产品主机通道和从通道的组合, 参考章节产品信息。

TBxCR<TBSYNC> 位 规定同步模式开关。如果从机通道<TBSYNC> 位被设置"1", 计数器将用软件或主机通道外部触发器开始来开始/停止同步。从机通道 TBxRUN<TBPRUN, TBRUN> 位不需要被设置。主机通道<TBSYNC> 位必须被设置"0"。

注意如果外部触发器计数器模式和定时器同步模式都设置, 定时器同步模式获得更高优先级。

12.4.2.3 计数器 清除

递增-计数器在以下时序清除:

1. 用 TBxRG1 检测匹配时
设置 TBxMOD<TBCLE> = "1", 如果比较器检测到计数器值 和设置在 TBxRG1 中值之间匹配时, UC 清除. 如果 TBxMOD<TBCLE> = "0", UC 工作为自由运行计数器.
2. 递增-计数器 停止时
如果 TBxRUN<TBRUN> = "0", UC 停止计数和清除计数器值。

12.4.2.4 递增-计数器 溢出

如果递增-计数器溢出, INTTBx 溢出中断生成。

12.4.3 定时器 寄存器 (TBxRG0, TBxRG1)

TBxRG0 和 TBxRG1 为比较递增-计数器值的设置值的寄存器且两个寄存器内置到各通道。如果比较器检测到此定时器中设置的寄存器值和 UC 递增-计数器中的值之间匹配, 输出匹配检测信号。

TBxRG0 和 TBxRG1 由与寄存器双缓冲一对的双-缓冲 配置组成. 初始状态下, 双缓冲禁止。

控制双缓冲禁用或使能由 TBxCR<TBWBF> 位指定。如果 <TBWBF> = "0", 双 缓冲变禁用。如果 <TBWBF> = "1", 变使能。当双缓冲使能, 在 UC 用 TBxRG1 匹配的情况下, 完成从寄存器缓冲到定时器寄存器 (TBxRG0/1)的数据传输。即使双缓冲使能, 当计数器停止时, 双缓冲工作为单缓冲, 和即时数据可以 写入到 TBxRG0 和 TBxRG1。

12.4.4 捕捉

此电路控制从 UC 递增-计数器 到 TBxCP0 和 TBxCP1 捕捉寄存器的锁存值时序。用锁存数据的时序通过 TBxMOD<TBxCPM[1:0]> 指定。

软件也可以被用来输入从 UC 递增-计数器 到捕捉 寄存器的值; 尤其, 每次 "0" 写入 TBxMOD<TBxCP>, UC 值接受到 TBxCP0 捕捉寄存器。

12.4.5 捕捉 寄存器 (TBxCP0, TBxCP1)

此寄存器 捕捉递增-计数器 (UC) 值。

12.4.6 递增计数器 捕捉 寄存器 (TBxUC)

除以上捕捉功能, UC 当前计数值可以通过读 TBxUC 寄存器捕捉.

12.4.7 比较器 (CP0, CP1)

此寄存器用递增-计数器 (UC) 和定时器 寄存器 (TBxRG0 和 TBxRG1) 设置的值比较来检测是否有无匹配. 如果检测到匹配, INTTBx0 和 INTTBx1 生成.

12.4.8 定时器 Flip-flop (TBxFF0)

定时器 flip-flop (TBxFF0)通过从比较器 和捕捉寄存器锁存信号的匹配信号反转. 可以使能或禁止反转通过设置 TBxFFCR<TBC1T1, TBC0T1,TBE1T1, TBE0T1>.

复位后 TBxFF0 值变为未定义. flip-flop 可以通过写入 "00" 到 TBxFFCR<TBFF0C[1:0]> 反转. 可以设置带"1" 通过 写入 "01," 且可以清除到 "0" 通过写入 "10."

TBxFF0 值可以输出到定时器输出引脚 (TBxOUT). 如果定时器执行输出, 对应端口设置必须提前编程.

12.4.9 捕捉 中断 (INTCAPx0, INTCAPx1)

中断 INTCAPx0 和 INTCAPx1 可以在从 UC 递增计数器到 TBxCP0 和 TBxCP1 捕捉寄存器的锁存值时序生成. 中断时序通过 CPU 指定.

12.5 各模式工作描述

12.5.1 16-位 间隔 定时器 模式

在生成恒定周期中断情况下, 设置生成 INTTBx1 中断定时器寄存器 (TBxRG1)间隔时间.

	7	6	5	4	3	2	1	0		
TBxEN	← 1	X	X	X	X	X	X	X	使能 TMRBx 工作。	
TBxRUN	← X	X	X	X	X	0	X	0	停止计数工作。	
中断设置-使能	← *	*	*	*	*	*	*	*	通过设置相应位到 "1", 允许 INTTBx1 中断。	
寄存器										
TBxFFCR	← X	X	0	0	0	0	1	1	禁用TBxFF0 反转触发器。	
TBxMOD	← 0	1	0	0	1	*	*	*	变化到预分频器输出时钟为输入时钟。规定捕捉功能禁用。	
			(***) = 001, 010, 011, 100, 101, 110, 111)							
TBxRG1	← *	*	*	*	*	*	*	*	规定时间间隔。(16 位)	
	← *	*	*	*	*	*	*	*		
TBxRUN	← *	*	*	*	*	1	X	1	开始 TMRBx。	

注:X;忽略 -; 无改变

12.5.2 16-位 事件 计数器 模式

通过使用输入时钟作外部时钟 (TBxIN 引脚 输入), 可以使之成为事件计数器. 在 TBxIN 引脚 输入上升 缘递增-计数器 计数. 可以使用软件捕捉的值和读捕捉的值读计数值.

	7	6	5	4	3	2	1	0	
TBxEN	← 1	X	X	X	X	X	X	X	使能 TMRBx 工作。
TBxRUN	← X	X	X	X	X	0	X	0	停止计数工作。
设置端口寄存器									安置相应端口到TBxIN。
TBxFFCR	← X	X	0	0	0	0	1	1	禁用TBxFF0 反转触发器。
TBxMOD	← 0	1	0	0	0	0	0	0	变化到TBxIN 为输入时钟。
TBxRUN	← *	*	*	*	*	1	X	1	开始 TMRBx。
TBxMOD	← 0	0	0	0	0	0	0	0	软件 捕捉完成。

注:X; 忽略 -; 无改变

12.5.3 16-位 PPG (可编程 脉冲 生成) 输出 模式

可以输出带任何频率和任何占空比 (可编程方波) 的方波。输出脉冲可以为或低-激活或高-激活。

当递增-计数器 (UC) 设置值匹配定时器 寄存器 (TBxRG0 和 TBxRG1) 设置值时, 可编程方波可以通过触发定时器 flip-flop (TBxFF) 从TBxOUT 引脚输出来反转。注意设置 TBxRG0 和 TBxRG1 值必须满足以下需要:

TBxRG0 设置 值 < TBxRG 设置 值 1

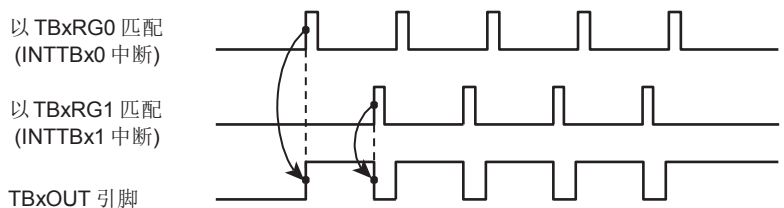


图 12-2 可编程 脉冲 生成 (PPG) 输出例

此模式, 通过使能 TBxRG0 双缓冲, 当递增-计数器设置值匹配 TBxRG1 的设置值时, 寄存器缓冲 0 的值被移到 TBxRG0。此帮助小占空比处理。

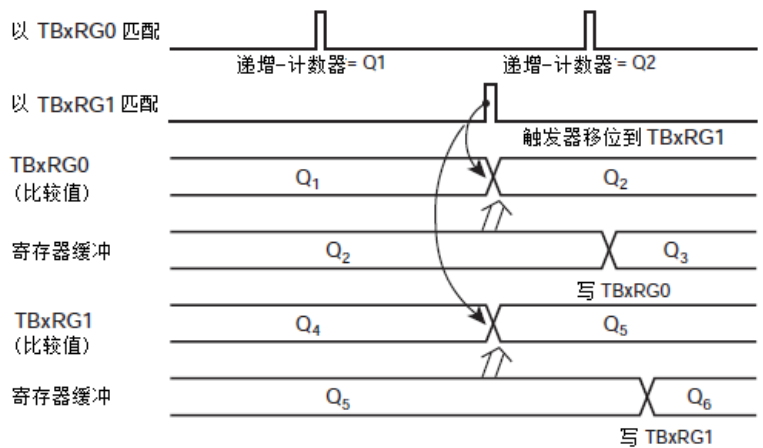


图 12-3 寄存器 缓冲 工作

此模式方块图如下所示。

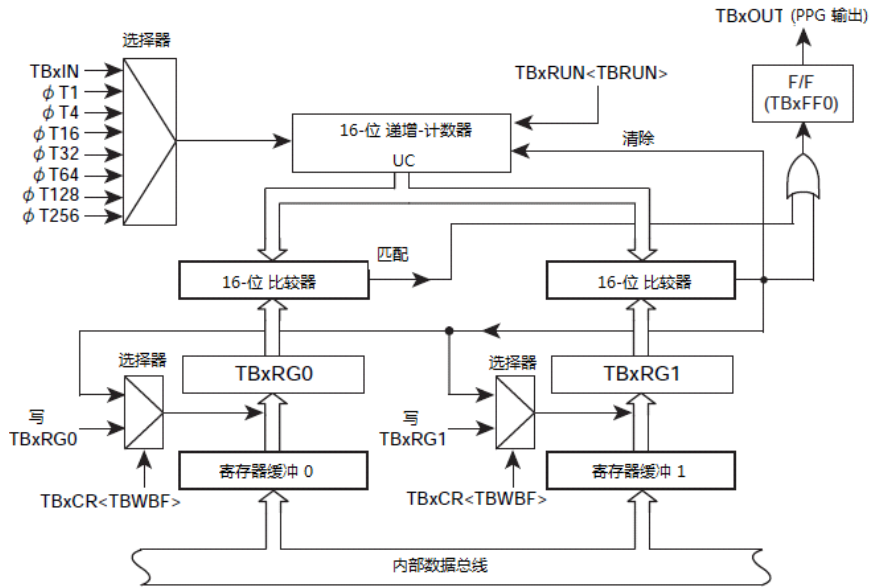


图 12-4 16-位 PPG 模式方块图

16-位 PPG 输出模式下各寄存器必须如下编程。

	7	6	5	4	3	2	1	0	
TBxEN	← 1	X	X	X	X	X	X	X	使能 TMRBx 工作。
TBxRUN	← X	X	X	X	X	0	X	0	停止计数工作。
TBxCR	← 0	0	0	X	-	0	0	0	禁用双缓冲。
TBxRG0	← *	*	*	*	*	*	*	*	规定占空比。(16 位)
TBxRG1	← *	*	*	*	*	*	*	*	规定周期。(16 位)
TBxCR	← 1	0	0	X	-	0	0	0	使能TBxRG0 双缓冲。 (当 INTTBx0 中断生成时, 变化占空比/周期)
TBxFFCR	← X	X	0	0	1	1	1	0	规定触发器 TBxFF0 来反转 当检测到带TBxRG0 或 TBxRG1匹配,且设置TBxFF0的初始值为 "0"。
TBxMOD	← 0	1	0	0	1	*	*	*	指派预分频器输出时钟为输入时钟,且禁用捕捉功能。
(** = 001, 010, 011,100,101,110,111)									
设置端口 寄存器。									UC 被清除来匹配 TBxRG1。 安置相应端口到 TBxOUT。
TBxRUN	← *	*	*	*	*	1	X	1	开始 TMRBx。

注:X; 忽略
-; 无改变

12.5.4 外部 触发器 可编程 脉冲 生成 输出 模式 (PPG)

用短延迟使用外部计数开始 触发器使能—发脉冲生成。

编程16-位递增-计数器 (UC)并在 TBxIN 引脚上升缘递增计数(TBxCR[1:0] = "01")。TBxRG0用脉冲延迟 (d) 加载, 且 TBxRG1用 TBxRG0 值 (d) 和 脉冲宽 (p)的总和加载。当 16-位递增-计数器停止时, 必须完成以上设置 (TBxRUN<TBRUN> = 0)。

要使能定时器 flip-flop 触发器, 设置 TBxFFCR<TBE1T1, TBE0T1> 为"11". 用此设置, 16-位递增-计数器 (UC) 对应 TBxRG0 或 TBxRG1 时, 定时器 flip-flop 反转。

通过外部 触发器设置 TBxRUN <TBRUN> 为 "1" 使能计数-递增。

通过外部触发器—发脉冲生成后, 禁止定时器 flip-flop 反转或停止 16 位计数器通过 TBxRUN<TBRUN> 设置。

图 12-5 中文本里使用的符 (d) 和 (p)对应 符 d 和 p.

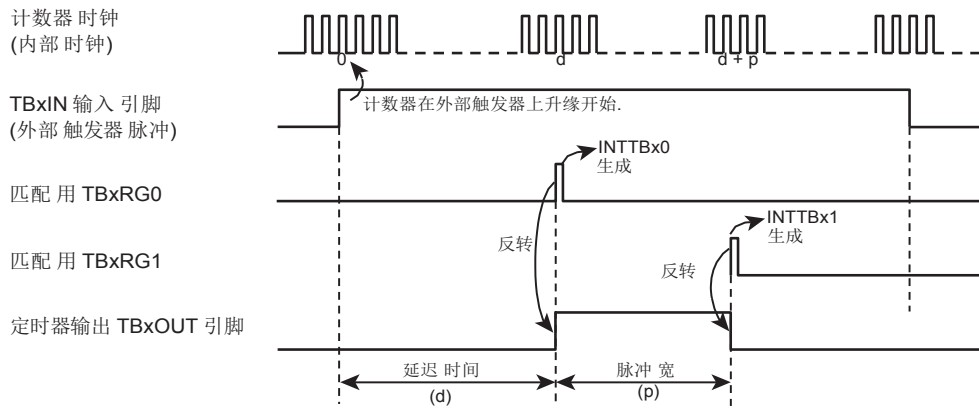


图 12-5 使用外部计数开始触发器—发脉冲生成 (有延迟)

12.6 使用捕捉功能的应用

捕捉功能可以被使用开发许多应用, 包括以下那些描述:

1. 通过外部脉冲输出触发的一发脉冲
2. 脉冲宽测量

12.6.1 通过外部脉冲单-发脉冲输出触发器

输出通过外部脉冲触发的一发脉冲被如下执行:

通过把它放入使用预分频器输出时钟的自由运行状态, 16-位递增-计数器去计数递增。通过 TBxIN 引脚输入外部脉冲。触发器通过使用捕捉功能在外部脉冲上升缘生成且递增-计数器的值被接受到捕捉寄存器 (TBxCP0)。

CPU 必须编程为了中断 INTCAPx0 在外部触发器脉冲上升缘生成。使用此中断来设置定时器寄存器 (TBxRG0) 到 TBxCP0 值 (c) 和延迟时间 (d), (c + d) 的总和, 且设置定时器寄存器 (TBxRG1) 到 TBxRG0 值和单-发脉冲的脉冲宽 (p), (c + d + p) 的总和。[TBxRG1 变化必须在下次匹配前完成。]

另外, 定时器 flip-flop 控制寄存器 (TBxFFCR < TBE1T1, TBE0T1 >) 必须设置为 "11"。当 TBxUC 匹配 TBxRG0 和 TBxRG1 时, 这会使能触发定时器 flip-flop (TBxFF0) 为反转。单-发脉冲输出后, 此触发器由 INTTBx0 / INTTBx1 中断禁用。

图 12-6 中文本里使用的符 (c), (d) 和 (p) 对应符 c, d 和 p.

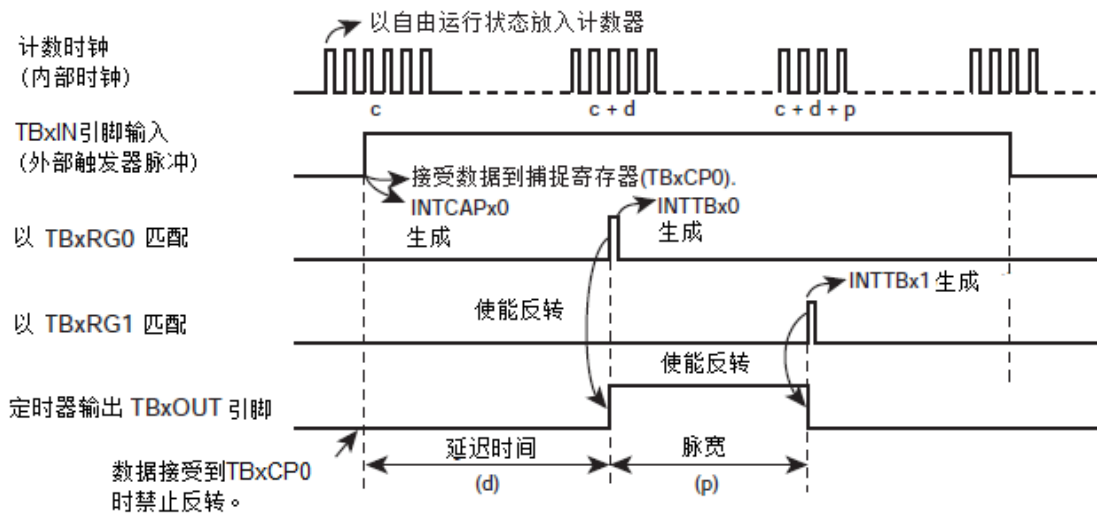


图 12-6 一发脉冲输出 (有延迟)

以下显示在上升缘触发 TBxIN 输入 3ms 后，输出 2 ms 宽单-发脉冲的情况的设置 ($\Phi T1$ 选择为计数。)

	7	6	5	4	3	2	1	0	
[[主要的处理] 由TBxIN 捕捉 设置									
设置PORT 寄存器									
TBxEN	← 1	X	X	X	X	X	X	X	安置相应端口到 TBxIN。
TBxRUN	← X	X	X	X	X	0	X	0	使能TMRBx 工作。 停止计数工作。
TBxMOD	← 0	1	0	1	0	0	0	1	改变源时钟到 $\Phi T1$ 。 在TBxIN上升缘获取一个值到TBxCP0。
TBxFFCR	← X	X	0	0	0	0	1	0	清除 TBxFF0 反转触发器和禁用。 安置相应端口到 TBxOUT。
设置 PORT 寄存器									
中断 Set-Enable 寄存器	← *	*	*	*	*	*	*	*	允许生成由 INTCAPx0 设为"1"的中断相应位指定的中断
TBxRUN	← *	*	*	*	*	1	X	1	开始TMRBx 模组。
[[INTCAPx0 中断服务程序的处理] 脉冲输出设置									
TBxRG0	← *	*	*	*	*	*	*	*	设置计数值.(TBxCAP0 + 3ms/ $\Phi T1$)
TBxRG1	← *	*	*	*	*	*	*	*	设置计数值.(TBxCAP0 + (3+2)ms/ $\Phi T1$)
TBxFFCR	← X	X	-	-	1	1	-	-	如果UC与TBxRG0和TBxRG1一致，反转TBxFF0
TBxIM	← X	X	X	X	X	1	0	1	除TBxRG1 相应中断以外
中断 Set-Enable 寄存器	← *	*	*	*	*	*	*	*	允许生成由INTTBx1 设为"1"的中断相应位指定的中断
[[INTTBx1中断服务程序的处理] 输出禁用									
TBxFFCR	← X	X	-	-	0	0	-	-	清除 TBxFF0 反转触发器设置。 通过设置"1"禁用由INTTBx中断相应位指定的中断。

注:X;忽略
-;无 改变

如果不需要延迟,当数据被接受到 TBxCP0时,通过产生 INTCAPx0 中断, TBxFF0 反转,且 TBxRG1 被设为TBxCP0 值 (c) 和 单-发脉冲 宽 (p), (c + p) 的总和。(TBxRG1 下一次匹配前,必须完成变化。)

当 UC 用 TBxRG1 匹配时, TBxFF0 使能反转, 且通过产生 INTTBx1 中断禁用。

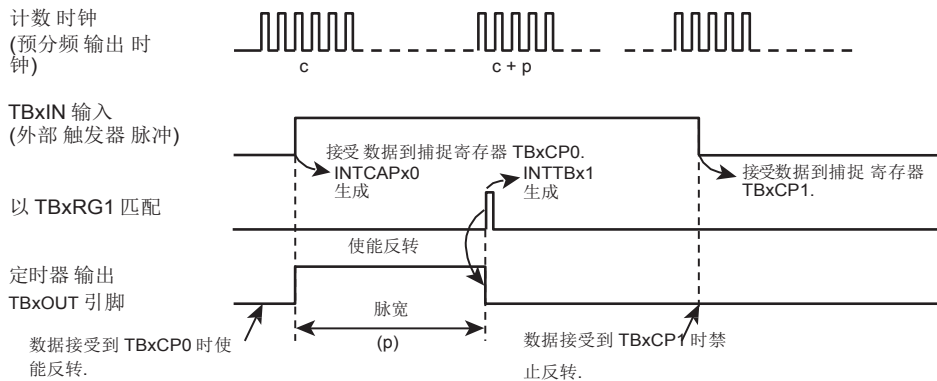


图 12-7 输出通过外部 脉冲触发的一发脉冲(无延迟)

12.6.2 脉冲宽测量

通过使用捕捉功能,可以测量外部脉冲的"高"电平宽。规定上,通过把它放入到一个使用预分频器输出时钟的自由-运行状态,通过 TBxIN 引脚,外部脉冲被输入且递增-计数器 (UC) 用来递增计数。通过使用捕捉功能和递增-计数器的值被接受到捕捉寄存器 (TBxCP0, TBxCP1), 触发器在每个外部脉冲上升和下降缘生成。CPU 必须编程为了通过 TBxIN 引脚, INTCAPx1 在外部脉冲输入的下降缘生成。

用内部时钟的时钟周期, 通过乘 TBxCP0 和 TBxCP1 之间的差分可以计算"高"电平脉冲宽。

比如, TBxCP0 和 TBxCP1 之间的差分为 100 且预分频器输出时钟周期为 $0.5 \mu\text{s}$, 脉冲宽为 $100 \times 0.5 \mu\text{s} = 50 \mu\text{s}$ 。

当测量时, 必须小心脉宽超过取决于使用的源时钟的 UC 最大计数时间。脉宽测量必须使用软件做。

也可以测量外部脉冲的"低"电平宽。此时, C2 生成首次时间和 C1 生成第二个时间之间的差分被初始获取通过执行 INTCAPx0 中断处理的第二级如图 12-8 中所示且此差分由预分频器输出时钟的周期相乘来获取"低"电平宽。

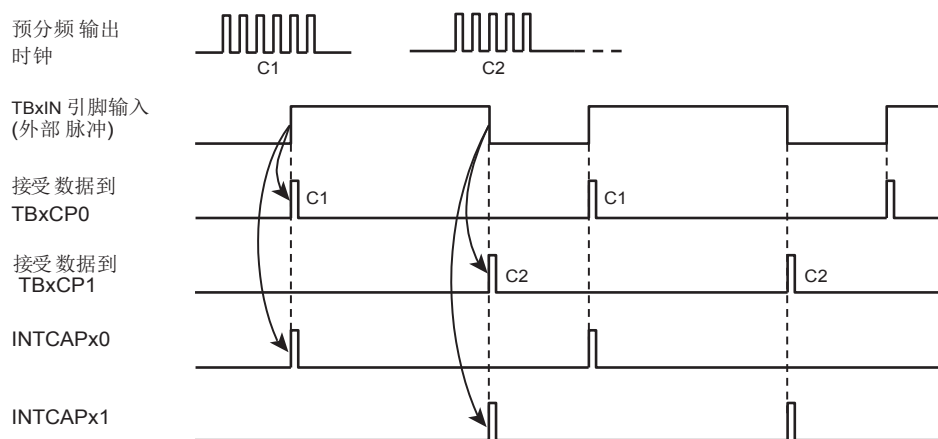


图 12-8 脉冲宽测量

13. 串行通道带 4 字节 FIFO (SIO/UART)

13.1 总览

串行通道 (SIO/UART) 有如下所示模式.

- 同步通信模式 (I/O 接口模式)
- 异步通信模式 (UART 模式)

其特征在以下给出.

- 传输时钟
 - 通过预分频分割, 从外设时钟 ($\phi T0$) 频率到 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128.
 - 使其可以到分割从预分频输出时钟频率到 1 到 16.
 - 使其可以到分割从预分频输出时钟频率到 $N+m/16$ ($N=2$ 到 15, $m=1$ 到 15). (仅 UART 模式)
 - 可用系统时钟 (fsys) (仅 UART 模式).
- 缓冲
 - 可用双缓冲功能.
 - 使其可以清除传输缓冲.
- FIFO
 - 可用 4 字节 FIFO 包括传输和接收.
- I/O 接口模式
 - 传输模式: 半双工 (传输/接收), 全双工
 - 时钟: 输出 / 输入 (可选 其上升或下降缘)
 - 使其可以指定持续传输间隔时间.
 - SCxTXD 引脚状态最后位输出后可以选择如下:
 - 保持 "高" 电平, "低" 电平或最后位状态
 - 时钟输入模式下, 欠载错误发生时, SCxTXD 引脚状态可以选择如下:
 - 保持 "高" 电平或 "低" 电平
 - SCxTXD 引脚最后位保持时间可以被指定在时钟输入模式中.
- UART 模式
 - 数据长: 7 位, 8 位, 9 位
 - 加奇偶位 (针对 9 位数据长)
 - 串行链接到使用唤醒功能
 - 带 \overline{SCxCTS} 引脚握手功能
 - SCxRXD 引脚噪声消除

以下解释, "x" 代表通道数.

13.2 配置

串行通道方块图和串行时钟生成器电路图所示如下。

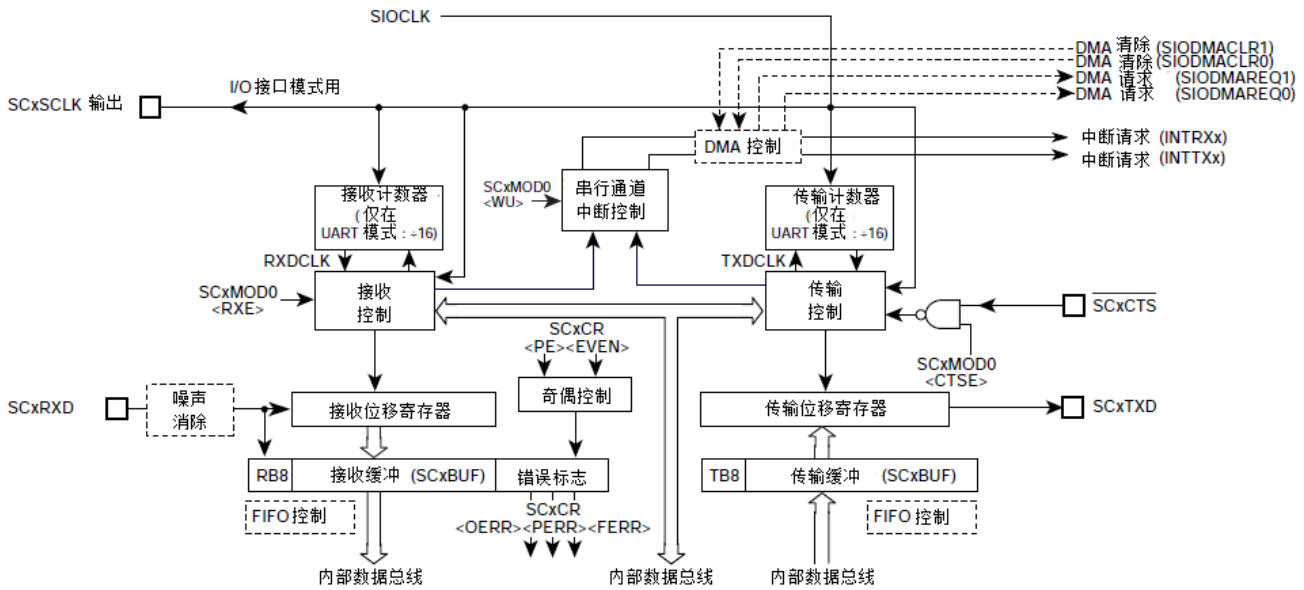


图 13-1 串行时钟生成电路

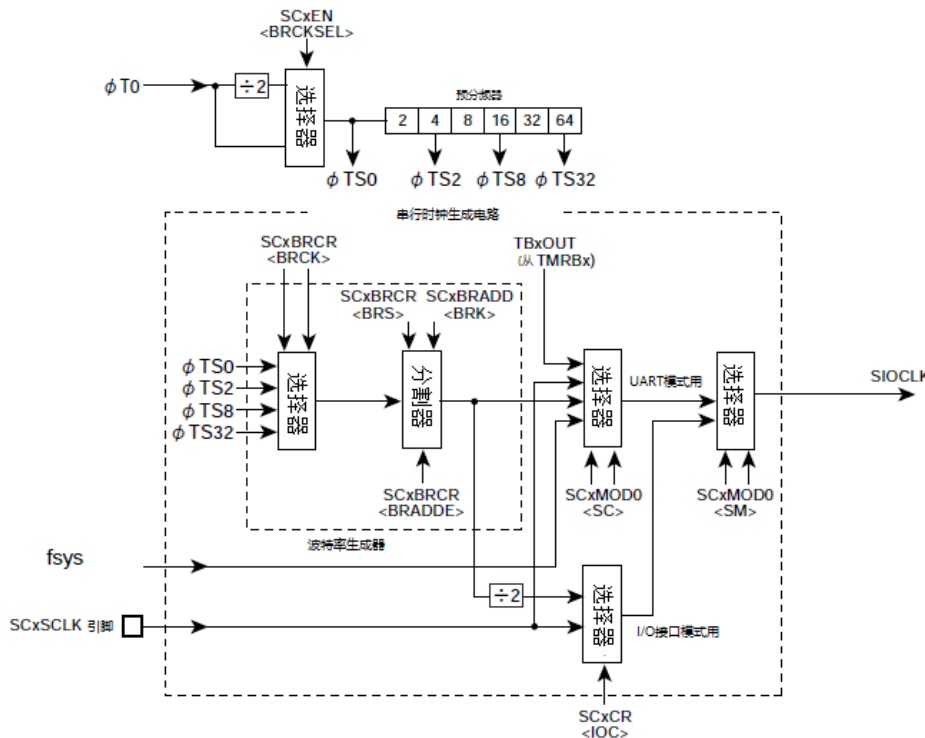


图 13-2 串行时钟生成电路方块图

13.3 寄存器 描述

13.3.1 寄存器 表

下表表示控制寄存器和其地址.

关于基地址详细, 参考"内存映射"章节的"外设功能地址表".

寄存器 名		地址 (基+)
使能 寄存器	SCxEN	0x0000
缓冲 寄存器	SCxBUF	0x0004
控制 寄存器	SCxCR	0x0008
模式 控制 寄存器 0	SCxMOD0	0x000C
波特 率 生成器 控制 寄存器	SCxBRCR	0x0010
波特 率 生成器 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
传输 FIFO 配置 寄存器	SCxTFC	0x0024
接收 FIFO 状态 寄存器	SCxRST	0x0028
传输 FIFO 状态 寄存器	SCxTST	0x002C
FIFO 配置 寄存器	SCxFCNF	0x0030

注: 不要修改任何控制寄存器, 数据被传输或接收时.

13.3.2 SCxEN (使能 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作"0".
1	BRCKSEL	R/W	选择输入预分频时钟. 0: $\phi T0/2$ 1: $\phi T0$
0	SIOE	R/W	串行通道工作 0: 禁止 1: 使能 被指定串行通道工作. 要使用串行通道, 设置 <SIOE> = "1". 工作禁止时, 无时钟提供到别的串行通道模组中寄存器. 此可以减少功耗. 如果串行通道工作被执行后禁止, 设置会维持在各寄存器.

13.3.3 SCxBUF (缓冲 寄存器)

SCxBUF 工作为传输缓冲或写入工作的 FIFO 且作为接收缓冲或读工作的 FIFO

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7-0	TB[7:0] / RB [7:0]	R/W	[写] TB: 传输 缓冲 或 FIFO [读] RB: 接收 缓冲 或 FIFO

13.3.4 SCxCR (控制 寄存器)

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

位	位 符	类型	功能
31-15	-	R	读作 "0".
14-12	EHOLD [2:0]	R/W	时钟输入模式下 SCxTXD 引脚最后位保持时间 (仅 I/O 接口模式用) 设置最后位保持时间且保持最后位保持时间的 SCLK 周期等于 或小于 SCLK 周期/2. 000: 2/fsys 100: 32/fsys 001: 4/fsys 101: 64/fsys 010: 8/fsys 110: 128/fsys 011: 16/fsys 111: 保留
11	-	R	读作 "0".
10	TXDEMP	R/W	欠运行错误发生时, 时钟输入模式下 SCxTXD 引脚状态. (仅 I/O 接口模式用) 0: "低" 电平 输出 1: "高" 电平 输出
9-8	TIDLE[1:0]	R/W	SCxTXD 引脚状态输出最后位后(仅 I/O 接口 模式用) <TIDLE[1:0]> 设置到"10"时,设置 "000" 到<EHOLD [2:0]>. 00: 保持"低" 电平 输出 01 :保持"高" 电平 输出 10: 保持最后 位 11: 保留
7	RB8	R	接收数据位 8 (仅 UART 模式用) 9-位 UART 模式下第 9 位接收数据 .
6	EVEN	R/W	奇偶 (仅 UART 模式用) 选择偶数或奇数奇偶. 仅 7-或 8-位 UART 模式下, 奇偶位可以被用. 0: 奇数 1: 偶数 选择偶数或奇数奇偶.
5	PE	R/W	加 奇偶 (仅 UART 模式用) 控制禁止或使能 奇偶. 仅 7- 或 8-位 UART 模式下, 奇偶位可以被用. 0: 禁止 1: 使能
4	OERR	R	过-运行 错误 标志 (注) 0: 正常工作 1: 错误
3	PERR	R	奇偶 / 欠-运行 错误 标志 (注) 0: 正常工作 1: 错误
2	FERR	R	成帧 错误 标志 (注) 0: 正常工作 1: 错误

位	位符	类型	功能
1	SCLKS	R/W	选择时钟 缘 (I/O 接口 模式用) 0: 在 SCxRXD 引脚下降缘, 传输缓冲数据被传输到 SCxTXD 引脚 每位. 在 SCxRXD 引脚上升缘, 接收从 SCxRXD 引脚的接收缓冲每位中的数据 . 此时, SCxRXD 引脚状态从"高"电平开始. (上升缘模式) 1: 在 SCxSCLK 引脚上升缘, 传输缓冲中的数据被传输到 SCxTXD 引脚中每位. 在 SCxSCLK 引脚下降缘接收从 SCxRXD 引脚的接收缓冲每位中的数据 . 此时, SCxSCLK 状态从 "低" 电平开始.
0	IOC	R/W	选择时钟 (I/O 接口 模式用) 0: 时钟 输出 模式 (传输 时钟从 SCxSCLK 引脚输出.) 1: 时钟 输入 模式 (传输 时钟输入 到 SCxSCLK 引脚.)

注:读时, <OERR>, <PERR> 和 <FERR>清除到"0".

13.3.5 SCxMOD0 (模式控制寄存器 0)

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

位	位符	类型	功能
31-8	-	R	读作"0".
7	TB8	R/W	传输数据位8 (仅 UART 模式用) 9位 UART 模式下, 写入第9位传输数据.
6	CTSE	R/W	握手功能控制 (仅 UART 模式用) 0: CTS 禁止 1: CTS 使能 控制握手功能. 设置"1"使能握手功能使用 SCxCTS 引脚.
5	RXE	R/W	接收控制 (注1)(注2) 0: 禁止 1: 使能
4	WU	R/W	唤醒功能 (仅 UART 模式用) 0: 禁止 1: 使能 仅9位 UART 模式下, 此功能可用. 其他模式, 此功能无意义. 使能时, 9位 UART 模式下, 仅 RB9 = "1"时中断发生.
3-2	SM[1:0]	R/W	指定传输模式. 00: I/O 接口模式 01: 7位 UART 模式 10: 8位 UART 模式 11: 9位 UART 模式
1-0	SC[1:0]	R/W	串行传输时钟 (仅 UART 模式用) 00: TMRB 输出 01: 波特率生成器 10: 系统时钟 (fsys) 11: 外部时钟 (SCxSCLK 引脚输入) (I/O 接口模式用, I/O 接口模式下, 传输时钟由 SCxCR<IOC>选择.)

注 1: 指定所有模式控制寄存器先然后<RXE>.

注 2: 不要停止接收工作 (通过设置 SCxMOD0<RXE> 到 "0") 数据接收时.

13.3.6 SCxMOD1 (模式 控制 寄存器 1)

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

位	位 符	类型	功能
31-8	-	R	读作"0".
7	I2SC	R/W	IDLE 0: 停止 1: 工作 指定 IDLE 模式下工作.
6-5	FDPX[1:0]	R/W	传输 模式 设置 00: 传输 禁止 01: 半 双工 (接收) 10: 半 双工 (传输) 11: 全 双工 I/O 接口 模式下, 配置传输模式. FIFO 使能时, 指定 FIFO 配置. UART 模式下, 仅指定配置 FIFO.
4	TXE	R/W	传输 控制 (注 1)(注 2) 0: 禁止 1: 使能 此位使能传输且 对所有传输 模式有效.
3-1	SINT[2:0]	R/W	持续传输间隔时间 (I/O 接口 模式用) 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 SCLK 输出模式被选择时, 此参数仅对 I/O 接口模式有效. 别的模式, 此参数无意义. 指定 I/O 接口模式下, 双缓冲或 FIFO 使能时, 持续传输间隔 时间.
0	-	R/W	写入 "0".

注 1: 指定所有模式控制寄存器先 然后<TXE>.

注 2: 不要停止传输工作 (通过设置 <TXE> 到 "0") 数据传输时.

13.3.7 SCxMOD2 (模式控制寄存器 2)

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

位	位符	类型	功能											
31-8	-	R	读作"0".											
7	TBEMP	R	<p>传输缓冲空标志</p> <p>0: 满</p> <p>1: 空</p> <p>如果双缓冲禁止, 此标志不重要.</p> <p>此标志表示传输双缓冲空.</p> <p>传输双缓冲数据被移到传输移位寄存器且双缓冲空时, 此位设置为 "1".</p> <p>再写入数据到双缓冲设置此位为 "0".</p>											
6	RBFL	R	<p>接收缓冲满标志</p> <p>0: 空</p> <p>1: 满</p> <p>如果双缓冲禁止, 此标志不重要.</p> <p>此标志表示接收双缓冲满.</p> <p>接收工作完成且接收数据从接收移寄存器被移到接收双缓冲时, 此位变为 "1". 读接收缓冲时, 此位清除为 "0".</p>											
5	TXRUN	R	<p>传输标志中</p> <p>0: 停止</p> <p>1: 工作</p> <p>此状态标志表示数据传输处理中.</p> <p><TXRUN> 和 <TBEMP> 位指以下状态.</p> <table border="1" data-bbox="523 1451 1249 1594"> <thead> <tr> <th><TXRUN></th><th><TBEMP></th><th>状态</th></tr> </thead> <tbody> <tr> <td>1</td><td>-</td><td>传输处理中</td></tr> <tr> <td rowspan="2">0</td><td>1</td><td>传输完成.</td></tr> <tr> <td>0</td><td>用传输缓冲数据等待状态</td></tr> </tbody> </table>	<TXRUN>	<TBEMP>	状态	1	-	传输处理中	0	1	传输完成.	0	用传输缓冲数据等待状态
<TXRUN>	<TBEMP>	状态												
1	-	传输处理中												
0	1	传输完成.												
	0	用传输缓冲数据等待状态												
4	SBLEN	R/W	<p>停止位长 (UART 模式用)</p> <p>0: 1-位</p> <p>1: 2-位</p> <p>此指定 UART 模式下传输停止位长.</p> <p>接收侧, 无论 <SBLEN> 仅使用单个位做决定.</p>											
3	DRCHG	R/W	<p>设置传输方向</p> <p>0: LSB 先</p> <p>1: MSB 先</p> <p>指定数据传输方向.</p> <p>UART 模式下, 设置此位为 LSB 先.</p>											
2	WBUF	R/W	<p>使能双-缓冲</p> <p>0: 禁止</p> <p>1: 使能</p> <p>此参数使能或禁止传输/接收双缓冲来传输 (SCLK 输出/输入模式下) 和接收 (SCLK 输出模式下) I/O 接口模式下的数据且 UART 模式下传输.</p> <p>接收 I/O 接口模式下 (时钟输入模式下) 和 UART 模式数据时, 无论 <WBUF> 双缓冲使能.</p>											

位	位符	类型	功能										
1-0	SWRST[1:0]	R/W	<p>软件 复位 覆盖 "01" 来取代"10" 生成软件复位. 软件复位被执行时, 以下位初始化且 传输/接收电路和 FIFO 变为初始状态 (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>, <RBFL>, <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>, <RBFL>, <TXRUN>	SCxCR	<OERR>, <PERR>, <FERR>
寄存器	位												
SCxMOD0	<RXE>												
SCxMOD1	<TXE>												
SCxMOD2	<TBEMP>, <RBFL>, <TXRUN>												
SCxCR	<OERR>, <PERR>, <FERR>												

注 1: 数据传输处理时, 任何软件复位工作必须连续被执行两次.

注 2: 在识别结束和软件复位指令执行开始之间的时间, 软件复位需要 2 时钟-期间 .

13.3.8 SCxBRCR (波特率生成器控制寄存器)

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7	-	R/W	写入 "0".
6	BRADDE	R/W	$N + (16 - K)/16$ 分割器功能 (仅 UART 模式用) 0: 禁止 1: 使能
5-4	BRCK[1:0]	R/W	波特率生成器选择输入时钟. 00: ϕ TS0 01: ϕ TS2 10: ϕ TS8 11: ϕ TS32
3-0	BRS[3:0]	R/W	分割比例 "N" 0000: N = 16 0001: N = 1 0010: N = 2 ... 1111: N = 15

注 1: UART 模式下使用 " $N + (16 - K)/16$ " 分割功能时, 作为分割比例, 1 ("0001") 或 16 ("0000") 可以不被应用到 N.

注 2: 仅 I/O 接口模式下双缓冲被用时, 波特率生成器分割比例 "1" 可以被指定.

13.3.9 SCxBRADD (波特率 生成器 控制 寄存器 2)

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

位	位 符	类	功能
31-4	-	R	读作 "0".
3-0	BRK[3:0]	R/W	指定 K 给 " $N + (16 - K)/16$ " 分割 (UART 模式用) 0000: 禁止 0001: K = 1 0010: K = 2 ... 1111: K = 15

表 13-1 波特率生成器分割比例设置表.

表 13-1 设置分割 比例

	<BRADDE> = "0"	<BRADDE> = "1" (注 1) (仅 UART 模式下)
<BRS>	指定 "N"	
<BRK>	无设置 需要	指定 "K" (注 2)
分割 比例	通过 N 分割	$N + \frac{(16 - K)}{16}$ 分割.

注 1: 要使用 " $N + (16 - K)/16$ " 分割功能, 确保设置 K 值 到<BRK> 后, 设置 <BRADDE>到"1"
." $N + (16 - K)/16$ " 分割功能可以仅被用在 UART 模式下.

注 2: 指定 "K = 0" 禁止.

13.3.10 SCxFCNF (FIFO 配置 寄存器)

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

位	位 符	类型	功能						
31-8	-	R	读作"0".						
7-5	-	R/W	确保写入 "000".						
4	RFST	R/W	字节被用在接收 FIFO. 0: 最大值 1: 相同于接收 FIFO 的充满电平 选择接收被用的 FIFO 字节数.(注 1) 0: 配置的 FIFO 字节数最大值(也见<CNFG>). 1: 相同于由 SC0RFC <RIL[1:0]> 指定的接收中断生成的充满电平.						
3	TFIE	R/W	指定传输 FIFO 的传输中断. 0: 禁止 1: 使能 传输 FIFO 使能时, 通过此参数, 传输中断使能或禁止.						
2	RFIE	R/W	指定接收 FIFO 的接收中断. 0: 禁止 1: 使能 接收 FIFO 使能时, 通过此参数, 接收中断使能或禁止.						
1	RXTXCNT	R/W	RXE/TXE 自动 禁止. 0: 无 1: 自动 禁止 控制传输和接收的自动禁止. 设置 "1" 使能工作如下. <table border="1" data-bbox="523 1550 1378 1749"> <tr> <td>半 双工 接收</td> <td>接收移位寄存器时, 接收缓冲且接收 FIFO 被装满到指定的有效字节数, SCxMOD0<RXE> 自动设置到 "0" 来禁止接收.</td> </tr> <tr> <td>半 双工 传输</td> <td>传输 移位寄存器时, 传输 缓冲且传输 FIFO 为空, SCxMOD1<TXE>自动设置到 "0" 来禁止传输.</td> </tr> <tr> <td>全 双工</td> <td>以上两个条件被满足其一时, <TXE> 且 <RXE> 自动设置到 "0" 来禁止传输和接收.</td> </tr> </table>	半 双工 接收	接收移位寄存器时, 接收缓冲且接收 FIFO 被装满到指定的有效字节数, SCxMOD0<RXE> 自动设置到 "0" 来禁止接收.	半 双工 传输	传输 移位寄存器时, 传输 缓冲且传输 FIFO 为空, SCxMOD1<TXE>自动设置到 "0" 来禁止传输.	全 双工	以上两个条件被满足其一时, <TXE> 且 <RXE> 自动设置到 "0" 来禁止传输和接收.
半 双工 接收	接收移位寄存器时, 接收缓冲且接收 FIFO 被装满到指定的有效字节数, SCxMOD0<RXE> 自动设置到 "0" 来禁止接收.								
半 双工 传输	传输 移位寄存器时, 传输 缓冲且传输 FIFO 为空, SCxMOD1<TXE>自动设置到 "0" 来禁止传输.								
全 双工	以上两个条件被满足其一时, <TXE> 且 <RXE> 自动设置到 "0" 来禁止传输和接收.								
0	CNFG	R/W	FIFO 使能. 0: 禁止 1: 使能 使能 FIFO.(注 2) <CNFG> 设置到 "1"时, FIFO 使能. 如果 FIFO 使能, SCOMOD1<FDPX[1:0]> 设置自动配置 FIFO 如下: <table border="1" data-bbox="523 1921 1378 2078"> <tr> <td>半 双工 接收</td> <td>接收 FIFO 4 字节</td> </tr> <tr> <td>半 双工 传输</td> <td>传输 FIFO 4 字节</td> </tr> <tr> <td>全 双工</td> <td>接收 FIFO 2 字节且传输 FIFO 2 字节</td> </tr> </table>	半 双工 接收	接收 FIFO 4 字节	半 双工 传输	传输 FIFO 4 字节	全 双工	接收 FIFO 2 字节且传输 FIFO 2 字节
半 双工 接收	接收 FIFO 4 字节								
半 双工 传输	传输 FIFO 4 字节								
全 双工	接收 FIFO 2 字节且传输 FIFO 2 字节								

注 1: 关于传输 FIFO, 配置的最大值字节数始终可用. (也见 <CNFG>.)

注 2: 9 位 UART 模式下, FIFO 可以不被用.

13.3.11 SCxRFC (接收 FIFO 配置 寄存器)

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

位	位 符	类型	功能															
31-8	-	R	读作 "0".															
7	RFCS	W	接收 FIFO 清除 (注) 1: 清除 SCxRFC<RFCS>设置为 "1" 时, 接收 FIFO 被清除且 SCxRST<RLVL[2:0]>为 "000". 而且读指针初始化. 读作 "0".															
6	RFIS	R/W	选择中断生成条件. 0: FIFO 装满 电平 (SCxRST<RLVL[2:0]>) = 接收 FIFO 装满 电平生成接收 中断 (<RIL [1:0]>) 时 1: FIFO 装满 电平 (SCxRST<RLVL[2:0]>) ≥ 接收 FIFO 装满 电平生成接收 中断 (<RIL [1:0]>) 时 关于中断条件详细, 参考 "13.13.1.2 FIFO"															
5-2	-	R	读作 "0".															
1-0	RIL[1:0]	R/W	FIFO 装满电平生成接收中断. <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>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> </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: 要使用传输/接收 FIFO 缓冲, 设置 SIO 传输模式 (半 双工/全 双工) 且使能 FIFO (SCxFCNF<CNFG> = "1") 后, 传输接收 FIFO 必须被清除。

13.3.12 SCxTFC (传输 FIFO 配置 寄存器)

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

位	位 符	类型	功能															
31-9	-	R	读作"0".															
8	TBCLR	W	传输缓冲 清除 0: 忽略 1: 清除 SCxTFC<TBCLR> 设置为"1" 时, 传输缓冲被清除. 读作"0".															
7	TFCS	W	传输 FIFO 清除 (注 1) 0: 忽略 1: 清除 SCxTFC<TFCS> 设置为"1"时, 传输 FIFO 被清除且 SCxTST<TLVL[2:0]> 为 "000". 且写入指针初始化. 读作 "0".															
6	TFIS	R/W	选择中断生成条件. 0: FIFO 装满 电平 (SCxTST<TLVL[2:0]>) = 传输 FIFO 装满电平生成传输 中断 (<TIL [1:0]>) 时 1: FIFO 装满 电平 (SCxTST<TLVL[2:0]>) ≤ 传输 FIFO 装满电平生成传输 中断 (<TIL [1:0]>) 时 关于中断条件详细, 参考 "13.13.2.2 FIFO"															
5-2	-	R	读作"0".															
1-0	TIL[1:0]	R/W	传输中断发生装满电平. <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>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> </tbody> </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: 要使用传输/接收 FIFO 缓冲, 设置 SIO 传输模式 (半 双工/全 双工) 且使能 FIFO (SCxFCNF<CNFG> = "1")后, 传输/接收 FIFO 必须被清除。

注 2: 万一 SCxEN<SIOE>="0" (停止 SIO/UART 工作) 或工作模式被改变为带 SCxMOD<I2SC>="0" (IDLE 模式下停止 SIO/UART 工作)的 IDLE 模式, SCxTFC 再初始化。执行以下工作后, 再配置 SCxTFC 寄存器。

13.3.13 SCxRST (接收 FIFO 状态 寄存器)

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

位	位符	类型	功能
31-8	-	R	读作"0".
7	ROR	R	接收 FIFO 过运行. (注) 0: 不生成 1: 生成
6-3	-	R	读作"0".
2-0	RLVL[2:0]	R	接收 FIFO 装满电平的状态. 000: 空 001: 1 字节 010: 2 字节 011: 3 字节 100: 4 字节

注:从 SCxBUF 读接收数据时, <ROR>清除到"0".

13.3.14 SCxTST (传输 FIFO 状态 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作"0".
7	TUR	R	传输 FIFO 欠运行.(注) 0: 不生成 1: 生成
6-3	-	R	读作"0".
2-0	TLVL[2:0]	R	传输 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:传输数据写入 SCxBUF 时, <TUR> 清除到"0".

13.4 各模式下工作

表 13-2 表示 模式.

表 13-2 模式

模式	类型	数据 长	传输 方向	指定是否使用 奇偶位.	停止位长(传输)
模式 0	同步通信模式 (I/O 接口 模式)	8 位	LSB 先/MSB 先	-	-
模式 1	异步通信模式 (UART 模式)	7 位	LSB 先	0	1 位 或 2 位
模式 2		8 位		0	
模式 3		9 位		×	

模式 0 为同步通信和可以被用来扩展 I/O. 此模式传输和接收用 SCLK 时钟同步的数据. SCLK 时钟可以被用作输入和输出模式. 数据传输方向可以从 LSB 先 或 MSB 先选择. 此模式不允许使用奇偶位或停止 位.

模式 1, 模式 2 和模式 3 为异步模式且传输方向可以仅选择为 LSB 先.

模式 1 和模式 2 中, 奇偶位可以加. 模式 3 有主机控制器可以开始从机控制器通过串行链接 (多-控制器系统)的唤醒功能. 传输中停止位可以从 1 位和 2 位选择. 接收中停止位长被固定到一位.

13.5 数据 格式

13.5.1 数据 格式 表

图 13-3 表示数据格式.

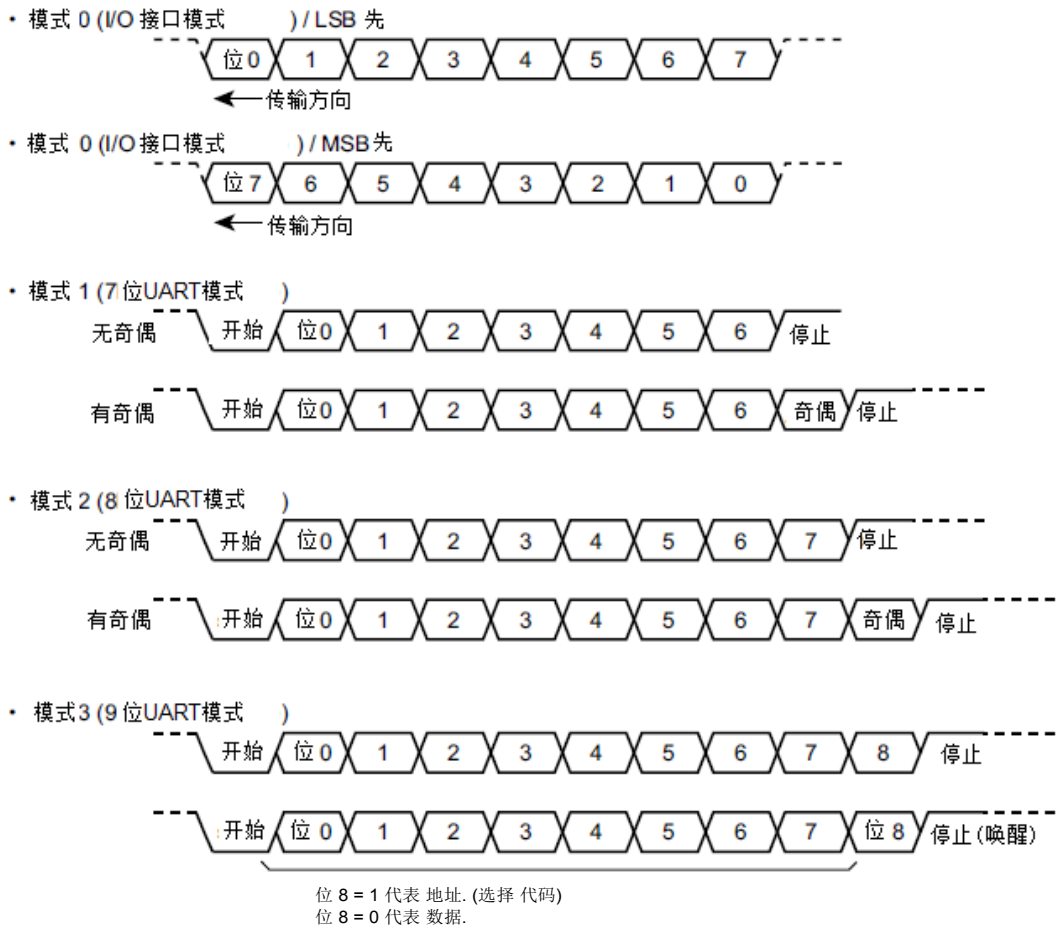


图 13-3 数据 格式

13.5.2 奇偶控制

仅 7- 或 8-位 UART 模式下, 奇偶位可以用传输数据加. 且接收奇偶位用生成一比较.

设置 "1" 到 SCxCR<PE> 使能奇偶. SCxCR<EVEN> 选择偶数或奇数奇偶.

13.5.2.1 传输

数据传输时, 奇偶控制电路用传输缓冲中数据自动生成奇偶. 奇偶位会储存在 SCxBUF<TB7> 7-位 UART 模式和 SCxMOD<TB8> 8-位 UART 模式下.

数据写入传输缓冲前<PE> 和 <EVEN> 设置必须完成.

13.5.2.2 接收

如果接收数据从接收移位寄存器被移到接收缓冲, 奇偶生成.

7-位 UART 模式下, 生成奇偶与储存在 SCxBUF<RB7>奇偶比较, 8-位 UART 模式下, 用 SCxCR<RB8>中的一个比较.

如果有任何区别, 奇偶错误发生和 SCxCR<PERR> 设置为"1".

FIFO 使用时, <PERR> 指奇偶错误生成在接收数据之一中.

13.5.3 STOP 位长

UART 传输 模式下, 停止位长可以通过设置 SCxMOD2<SBLLEN> 一位或两位选择. 被接收而无论此位设置时, 停止位数据长被决定为一位.

13.6 时钟 控制

13.6.1 预分频

有 7-位分割预分频输入时钟 $\phi T0$ 的预分频通过 1, 2, 4, 8, 16, 32, 64 和 128.

使用时钟/模式控制方块中 CGSYSCR 和 SCxEN<BRCKSEL>选择预分频输入时钟 .

仅 波特率生成器通过 SCxMOD0<SC[1:0]> = "01"选择为传输时钟时, 预分频变为激活.

13.6.2 串行时钟 生成 电路

串行时钟生成电路为生成传输和接收时钟 (SIOCLK)由可以通过设置波特率生成器和模式选择时钟的电路组成.

13.6.2.1 波特率 生成器

波特率生成器生成传输且接收决定串行通道传输率的时钟.

(1) 波特率生成器输入时钟

波特率生成器输入时钟从由 1, 4, 16 和 64 分割的预分频输出选择.

此输入时钟通过设置 SCxEN<BRCKSEL> 和 SCxBRCR<BRCK>选择.

SCxEN<BRCKSEL>	SCxBRCR<BRCK>	波特率生成器输入 时钟 ϕT_x
0	00	$\phi T0/2$
0	01	$\phi T0/8$
0	10	$\phi T0/32$
0	11	$\phi T0/128$
1	00	$\phi T0$
1	01	$\phi T0/4$
1	10	$\phi T0/16$
1	11	$\phi T0/64$

(2) 波特率生成器输出时钟

波特率生成器中输出时钟的频率分割比例通过 SCxBRCR 和 SCxBRADD 设置.

以下频率分割比例可以被用; I/O 接口 模式, 1/N 或 $1/(N + (16-K)/16)$ UART 模式下, 1/N 频率分割 .

下表表示可以选择的; 频率分割比例.

模式	分割 功能 设置 SCxBRCR<BRADDE>	通过 N 分割 SCxBRCR<BRS[3:0]>	通过 K 分割 SCxBRADD<BRK[3:0]>
I/O 接口	通过 N 分割	1 到 16 (注)	-
UART	通过 N 分割	1 到 16	-
	N + (16-K)/16 分割	2 到 15	1 到 15

注: 1/N (N=1) 频率分割比例仅双缓冲使能时可以被用.

波特率生成器的分割器输入时钟为 ϕ_{Tx} , 1/N 和 N + (16-K)/16 时波特率所示如下.

- 通过 N 分割

$$\text{波特率} = \frac{\phi_{Tx}}{N}$$

- N + (16-K)/16 分割

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

13.6.2.2 时钟 选择 电路

时钟可以通过设置模式和寄存器选择.

模式可以通过设置 SCxMOD0<SM[1:0]>指定

I/O 接口模式中时钟通过设置 SCxCR<IOC><SCLKS>选择.

UART 模式下时钟通过设置 SCxMOD0<SC[1:0]>选择.

(1) I/O 接口模式下传输时钟

表 13-3 表示 I/O 接口模式下时钟选择.

表 13-3 I/O 接口模式下时钟选择

模式 SCxMOD0<SM[1:0]>	输入/输出 选择 SCxCR<IOC>	时钟缘选择 SCxCR<SCLKS>	使用时钟
"00" (I/O 接口 模式)	"0" (时钟 输出 模式)	"0" (传输:下降 缘, 接收:上升 缘)	通过 波特率生成器 输出分割 2.
		"1" (传输:上升 缘, 接收:下降 缘)	通过 波特率生成器 输出分割 2
	"1" (时钟 输入 模式)	"0" (传输:下降 缘, 接收:上升 缘)	SCxSCLK 引脚 输入
		"1" (传输:上升 缘, 接收:下降 缘)	SCxSCLK 引脚 输入

要使用 SCxSCLK 输入, 以下条件必须被满足.

- 如果双缓冲被用
 - SCLK 周期 > 6/fsys
- 如果双缓冲不被用
 - SCLK 周期 > 8/fsys

(2) UART 模式下, 传输时钟

表 13-4 表示 UART 模式下, 时钟选择. UART 模式下, 使用前, 选择时钟接收计数器或传输计数器中通过 16 分割.

表 13-4 UART 模式下时钟 选择

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

要使用 SCxSCLK 引脚输入, 以下条件必须被满足.

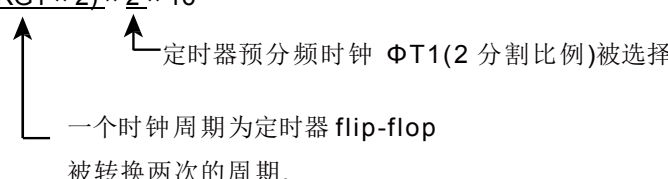
- SCLK 周期 > 2/fsys

要启用时间输出, 计数器值和 TBxRG1 的匹配时, 定时器 flip-flop 输出反转. SIOCLK 时钟频率为 "TBxRG1 × 2 的设置值".

波特率可以通过使用通过以下方程式获得.

波特率 计算

$$\text{传输率} = \frac{\text{时钟频率通过 CGSYSCR<PRCK[1:0]> 选择}}{(\text{TBxRG1} \times 2) \times 2 \times 16}$$



13.7 传输/接收 缓冲和 FIFO

13.7.1 配置

图 13-4 表示 传输 缓冲配置, 接收缓冲和 FIFO.

使用缓冲和 FIFO 需要适当设置. 配置可以根据模式被预定义.

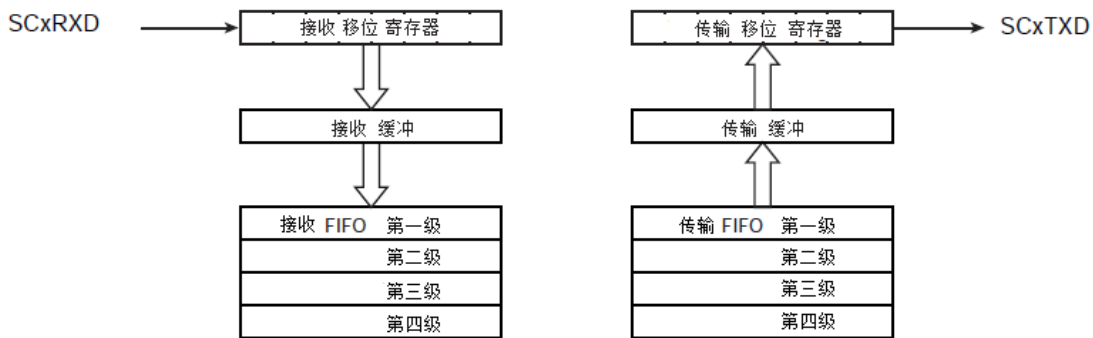


图 13-4 缓冲和 FIFO 配置

13.7.2 传输/接收 缓冲

传输缓冲和接收缓冲为双-缓冲. 缓冲配置被 SCxMOD2<WBUF>指定.

串行通道工作接收时, 如果 I/O 接口 模式下, 工作为时钟输入模式或工作为 UART 模式, 其被双缓冲, 无论 <WBUF> 设置.

别的模式下, 根据 <WBUF> 设置.

表 13-5 表示模式和缓冲之间关系.

表 13-5 模式和缓冲组成

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

13.7.3 初始化 传输 缓冲

传输用传输缓冲中数据停止时, 新传输数据写入传输缓冲前需要初始化传输缓冲器.

传输工作停止时, 传输缓冲必须初始化. 要停止传输工作可以通过读 SCxMOD2<TXUN>确认. 停止传输工作确认后, SCxTFC<TBCLR>设置为 "1" 且初始化传输缓冲.

传输 FIFO 使能时, 初始化工作取决于传输 FIFO 中的数据. 如果传输 FIFO 有数据, 数据从传输 FIFO 被传输到传输缓冲. 如果没有数据, SCxMOD2<RBEMP> 设置为 "1".

注: 带时钟输入模式的 I/O 接口模式中被异步输入. 传输工作停止时, 不要输入时钟.

13.7.4 FIFO

以上描述的双缓冲功能外, 4-字节 FIFO 可以被用.

要使能 FIFO, 通过设置 SCxMOD2<WBUF> 为 "1" 和 SCxFCNF<CNFG> 为 "1" 使能双缓冲. FIFO 缓冲配置被 SCxMOD1<FDPX[1:0]> 指定.

注: 要使用传输/接收 FIFO 缓冲, 设置 SIO 传输模式 (半双工/全双工) 且使能 FIFO (SCxFCNF<CNFG> = "1") 后, 传输/接收 FIFO 必须被清除.

表 13-6 表示模式和 FIFO 之间关系.

表 13-6 模式和 FIFO 组成

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

13.8 状态 标志

SCxMOD2 有两型标志. 此位仅双缓冲使能时重要.

<RBFLl> 为表示接收缓冲为满的标志. 一 帧数据被接收且数据从 接收移位寄存器被移到接收缓冲时, 此位变为 "1". 读到的接收缓冲器被读时, 此位清除为"0".

<TBEMP> 表示传输缓冲为空. 传输缓冲中数据被移到传输移位寄存器时, 此位设置为"1". 数据设置为传输缓冲时, 位清除为"0".

13.9 错误 标志

三个错误标志被提供在 SCxCR 中. 标志意义被改变取决于模式. 下表表示各模式下意义.

读 SCxCR 后, 这些标志被清除为 "0".

模式	标志		
	<OERR>	<PERR>	<FERR>
UART 模式	过-运行 错误	奇偶 错误	成帧 错误
I/O 接口 模式 (时钟 输入 模式)	过-运行 错误	欠-运行 错误 (双缓冲和 FIFO 被用时)	被固定到 0
		被固定到 0 (双缓冲和 FIFO 不被用时)	
I/O 接口 模式 (时钟 输出 模式)	未定义	未定义	被固定到 0

13.9.1 OERR 标志

UART 和 I/O 接口模式下, 接收缓冲已经被读前, 此位设置为 "1" 通过完成下一个帧接收生成错误时.

如果接收 FIFO 使能, 接收数据自动被移到接收 FIFO 且无过- 运行错误会生成直到接收 FIFO 满 (或直到可用字节全被占).

带时钟输出模式的 I/O 接口模式下, 设置标志时 SCxSCLK 引脚输出停止.

注:要从时钟输出模式的 I/O 接口模式切换到或模式, 读 SCxCR 且清除过运行标志.

13.9.2 PERR 标志

此标志指 UART 模式下, 奇偶错误和 I/O 接口模式下, 欠-运行错误或传输完成.

UART 模式下, " 从接收数据生成的奇偶与接收的奇偶位不同时, <PERR> 设置为"1".

I/O 接口模式下, 双缓冲使能时, 在以下条件<PERR>设置为"1" .

时钟 输入模式下,完成传输缓冲中没有数据的传输移位寄存器输出后, 时钟输入时, <PERR>设置为"1" .

时钟输出模式下,完成所有数据输出且时钟输出停止后, <PERR> 设置为"1".

注:要从带时钟输出模式的切换到或模式, 读 SCxCR 且 清除欠-运行标志.

13.9.3 FERR 标志

如果通过采样中心周围的位, 对应的停止位被决定为 "0", 成帧错误生成. 无论 SCxMOD2<SBLEN>中停止位长设置, 停止位状态仅通过第 1 停止位被决定.

I/O 接口模式下, 此位被固定到"0".

13.10 接收

13.10.1 接收 计数器

接收计数器为 4-位 二进制计数器且通过 SIOCLK 递增-计数.

UART 模式下,十六 SIOCLK 时钟脉冲在接收单数据位中被用且数据符号以八脉冲采样.

13.10.2 接收 控制 单元

13.10.2.1 I/O 接口 模式

带 SCxCR <IOC>时钟 输出模式下设置为"0", SCxRXD 引脚在 SCxSCLK 引脚的上升或下降缘采样取决于 SCxCR <SCLKS>.

带 SCxCR <IOC>时钟 输入模式下设置为"1", SCxRXD 引脚 在 SCxSCLK 引脚的上升或下降缘采样取决于 SCxCR <SCLKS>.

13.10.2.2 UART 模式

接收控制单元有开始位检测电路,被用来在正常开始位被检测时, 初始接收工作.

13.10.3 接收 工作

13.10.3.1 接收 缓冲

接收数据用 1 位 储存在接收 移位 寄存器中. 当完整的位设置被储存时, 中断 INTRXx 生成.

双缓冲使能时, 数据被移到接收缓冲 (SCxBUF) 且 接收双缓冲满标志 (SCxMOD2<RBFL>) 设置为"1". 通过读接收缓冲, 接收缓冲满标志清除为 "0". 双缓冲禁止时, 接收缓冲全标志无意义.

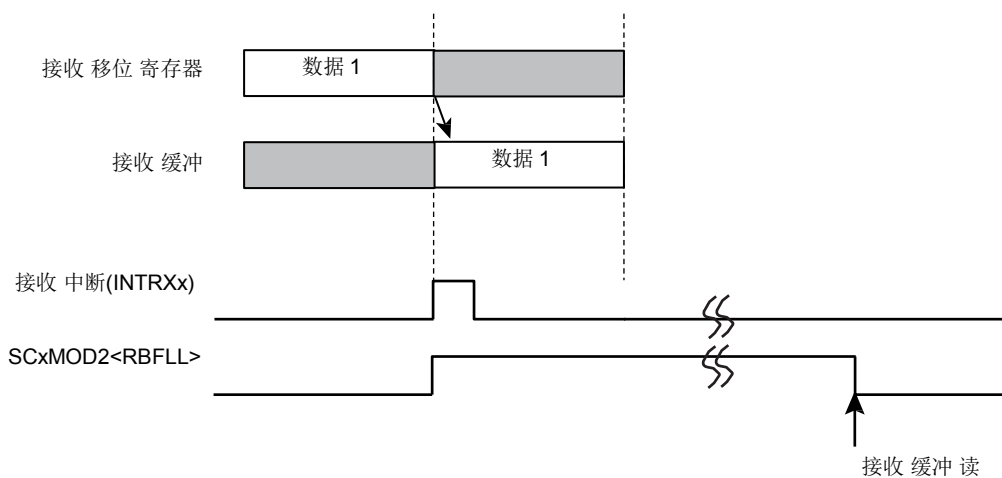


图 13-5 接收缓冲工作

13.10.3.2 接收 FIFO 工作

FIFO 使能时, 接收数据从接收缓冲被移到 接收 FIFO 且接收缓冲满标志被立即清除. 中断会生成根据 SCxRFC<RIL[1:0]>.

注: UART 模式下带奇偶位的数据通过使用 FIFO 被接收时, 奇偶错误标志表示接收数据中发生奇偶错误.

半双工接收模式下配置和工作描述如下.

- SCxMOD1<FDPX[1:0]> = "01" :传输模式设置为半双工模式
- SCxFCNF<RFST><TFIE><RFIE> : 到达装满电平后, 自动禁止持续接收.
- <RXTCNT><CNFG> = "10111" : 接收 FIFO 中被用的字节数相同于中断生成装满电平.
- SCxRFC<RIL[1:0]> = "00" : 生成接收中断 FIFO 的装满电平被设置为 4 字节
- SCxRFC<RFCS><RFIS> = "01" :清除接收 FIFO 且设置中断生成条件.

以上 FIFO 配置设置后, 通过写入 "1" SCxMOD0<RXE>数据接收开始. 数据全储存在接收移位寄存器中时, 接收缓冲且接收 FIFO, SCxMOD0<RXE>自动清除且接收工作完成.

以上条件下, 如果到达装满电平后使能持续接收,变为可以通过读 FIFO 中数据, 持续接收数据.

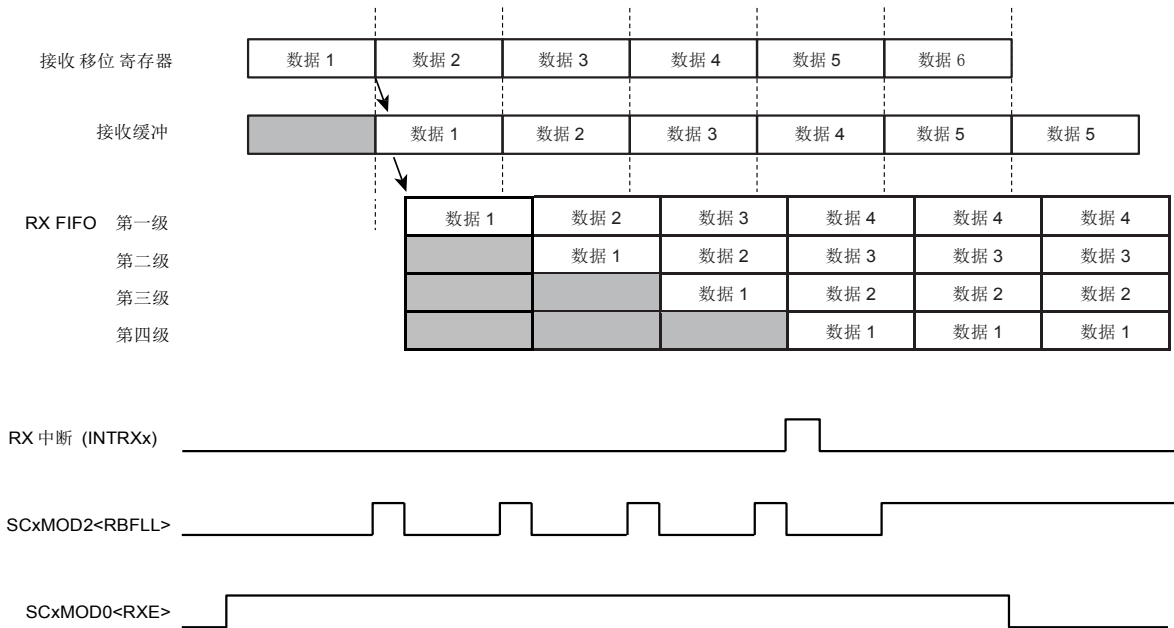


图 13-6 接收 FIFO 工作

13.10.3.3 I/O 接口 模式带时钟 输出 模式

带时钟输出模式的 I/O 接口模式下设置,所有接收数据储存在接收缓冲和 FIFO 时, 时钟停止. 所以, 此模式下, 过-运行错误标志无意义.

SCLK 输出时序停止且重输出取决于接收缓冲和 FIFO.

(1) 单 缓冲情况

接收数据后, 停止时钟输出.此模式下, I/O 接口可以用传输设备通过握手传输各数据.

读缓冲数据时, 时钟输出重新开始.

(2) 双 缓冲情况

接收到接收移位寄存器和接收缓冲中数据后, 停止时钟输出.

读数据时, 时钟输出重新开始.

(3) FIFO 情况

接收到移位寄存器, 接收的缓冲 和 FIFO 中数据后, 停止时钟输出.

读一字节数据时, 接收缓冲中数据被传输到 FIFO 且接收移位寄存器中数据被传输到接收缓冲且时钟输出重新开始.

如果 SCxFCNF<RXTXCNT> 设置为 "1",用清除 SCxMOD0<RXE>, 时钟停止且接收工作停止.

13.10.3.4 读接收数据

无论使能或禁止 FIFO,从接收缓冲 (SCxBUF) 读接收数据.

接收 FIFO 禁止时,缓冲满标志 SCxMOD2<RBFL> 清除为"0" 通过读此位. 读从接收缓冲器的数据前,下一个数据可以在接收移位寄存器中接收. 8-位 UART 模式下要加的奇偶位和 9-位 UART 模式下最重要的位会储存在 SCxCR<RB8>.

接收 FIFO 使能时, 9-位 UART 模式被禁止因为多大 8-位 数据可以储存在接收 FIFO. 8-位 UART 模式下, 奇偶位丢失但奇偶错误被决定且结果储存在 SCxCR<PERR>.

13.10.3.5 唤醒 功能

9-位 UART 模式下, 在唤醒模式下从机控制器可以通过设置唤醒功能 SCxMOD0 <WU> 为 "1" 工作. 此时, 仅当 SCxCR <RB8>设置为"1"时, 中断 INTRXx 会生成.

13.10.3.6 过运行 错误

接收 FIFO 禁止时, 过运行错误发生 而不在接收下一个数据前完成读数据. 过运行错误发生时, 接收缓冲和 SCxCR<RB8>内容不丢失, 但 接收移位寄存器内容丢失.

接收 FIFO 使能时, 过运行错误发生且接收 FIFO 为满时通过移下一个数据到接收缓冲前, 不读接收 FIFO, 设置过运行标志. 此时, 接收 FIFO 内容不丢失.

带时钟输出模式的 I/O 接口模式下, 时钟输出自动停止, 所以此标志无意义.

注: 模式从带时钟输出模式的 I/O 接口模式被改变为或模式时, 读 SCxCR 且清除过运行标志.

13.11 传输

13.11.1 传输 计数器

传输计数器为 4-位 二进制计数器且在接收计数器情况下通过 SIOCLK 计数。UART 模式下, 其以每第 16 时钟 脉冲生成传输时钟 (TXDCLK)。

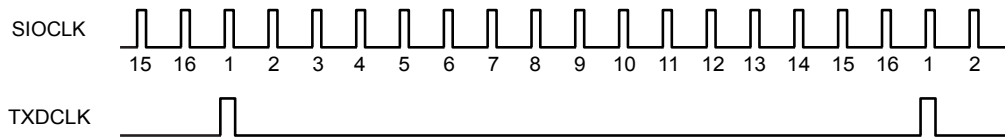


图 13-7 UART 模式下传输时钟生成

13.11.2 传输 控制

13.11.2.1 I/O 接口 模式下

带 SCxCR<IOC> 的时钟输出模式下设置为 "0", 传输缓冲中各位数据根据 SCxCR<SCLKS> 在 SCxSCLK 引脚上升或下降 缘输出到 SCxTXD 引脚。

带 SCxCR<IOC> 的时钟输入模式下设置为 "1", 传输缓冲中各位数据根据 SCxCR<SCLKS> 在 SCxSCLK 引脚上升或下降缘输出到 SCxTXD 引脚。

13.11.2.2 UART 模式下

传输数据写入到传输缓冲时, 数据传输被初始在下一个 TXDCLK 上升缘且传输移位时钟信号也生成。

13.11.3 传输工作

13.11.3.1 传输缓冲工作

如果双缓冲禁止, CPU 仅写入数据到传输移位寄存器且数据传输完成时传输中断 INTTXx 生成.

双缓冲使能 (包括传输 FIFO 使能情况)时, 如果 "1"设置到 SCxMOD1<TXE>, 传输缓冲中数据传输到传输移位寄存器. 同时 INTTXx 中断生成且传输缓冲空标志 (SCxMOD2<TBEMP>) 设置为 "1". 此标志指下一个传输数据可以写入. 下一个数据写入传输缓冲时, <TBEMP> 标志清除为 "0".

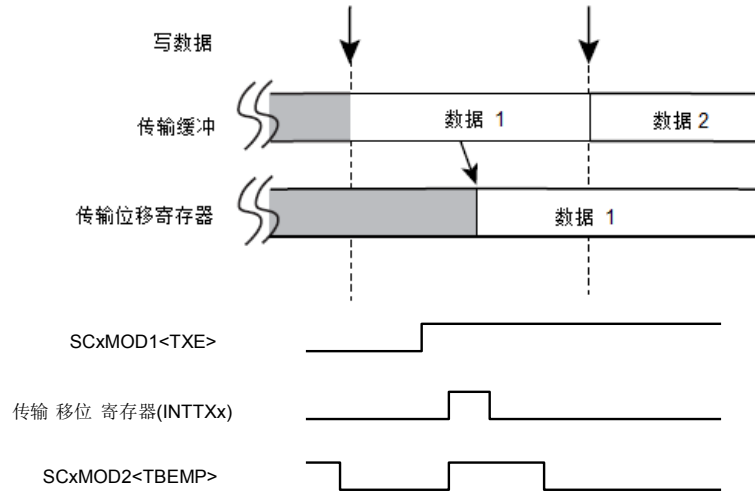


图 13-8 传输缓冲(双-缓冲使能)工作

13.11.3.2 传输 FIFO 工作

FIFO 使能时, 最大值 5-字节数据可以使用传输缓冲和 FIFO 储存. 一旦传输使能, 数据从传输缓冲传输到传输移位寄存器且开始传输. 如果数据存在 FIFO 中, 立即数据被移到传输缓冲, 且 <TBEMP> 标志清除为"0".

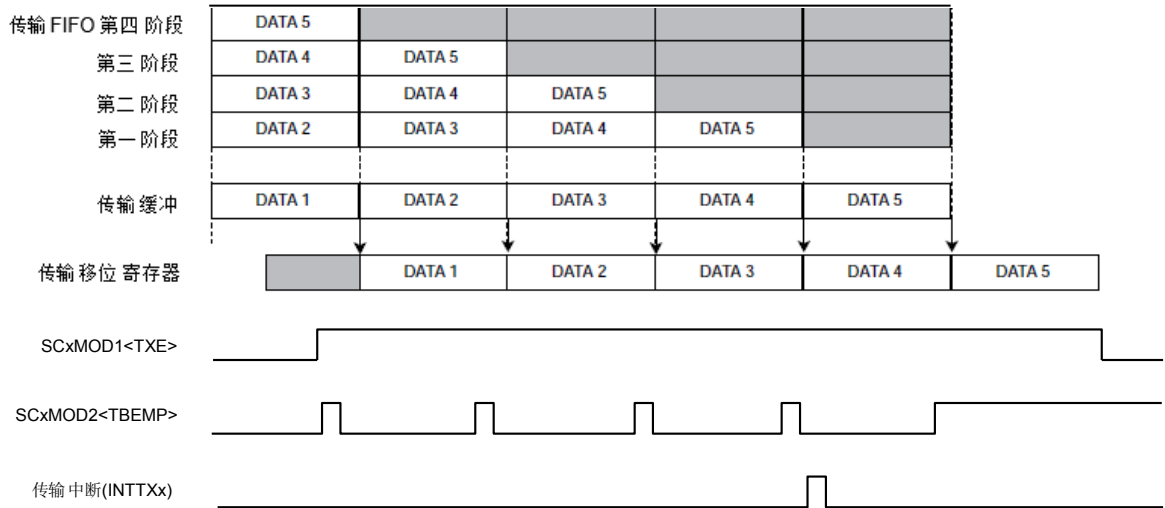
注: 要使用传输 FIFO 缓冲, 设置 SIO 传输模式 (半 双工/ 全 双工) 且使能 FIFO(SCxFCNF<CNFG>="1")后, 传输 FIFO 必须清除。

通过设置传输模式为半双工的传输 5 字节数据流设置和工作如下所示.

- SCxMOD1<FDPX[1:0]> = "10" :传输 模式设置为半双工.
- SCxFCNF<RFST><TFIE><RFIE> :如果 FIFO 变为空, 传输自动禁止.
- <RXTXCNT><CNFG> = "11011" :接收 FIFO 中被用字节数相同于中断
:生成装满电平.
- SCxTFC<TIL[1:0]> = "00" :设置 中断生成装满电平为 "0".
- SCxTFC<TFCS><TFIS> = "11" :清除接收 FIFO 且设置中断生成条件.
- SCxFCNF<CNFG> = "1" :使能 FIFO

配置以上设置后, 数据传输可以通过写入 5 字节数据到传输缓冲和 FIFO 初始, 且设置 SCxMOD1<TXE> 位为 "1". 最后传输数据被移到传输缓冲时, 传输中断生成. 最后数据传输完成时, 时钟停止且传输顺序被终止.

一旦以上设置被配置, 如果传输不设置为自动禁止, 传输应该最后写入传输数据.



13.11.3.3 I/O 接口模式带时钟输出模式下传输

带时钟输出模式的 I/O 接口模式下, 所有数据传输完成且欠运行错误不会发生时, 时钟输出自动停止.

时序挂起和时钟输出的恢复不同取决于缓冲和 FIFO 使用.

(1) 单缓冲

时钟输出停止每次输一帧数据传. 用通信另一侧各数据的握手可以使能. 下一个数据写入缓冲时, 时钟输出恢复.

(2) 双缓冲

在完成传输移位寄存器且传输缓冲中的数据传输时, 时钟输出停止. 下一个数据写入缓冲时, 时钟输出恢复.

(3) FIFO

储存在传输移位寄存器, 传输缓冲和 FIFO 中的所有数据的传输完成, SCLK 输出停止. 下一个数据写入, 时钟输出恢复.

如果配置 SCxFCNF<RXTXCNT>, SCxMOD0<TXE> 位在时钟停止和传输停止同时被清除.

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

数据保持时间通过最后位后, SCxTXD 引脚的电平被输出由 SCxCR<TIDLE>指定.

SCxCR<TIDLE>为"00"时, SCxTXD 引脚的电平被输出 "低" 电平. SCxCR<TIDLE>为 "01"时, SCxTXD 引脚电平输出 "高" 电平. SCxCR<TIDLE>为"10"时, SCxTXD 引脚电平输出最后位电平.

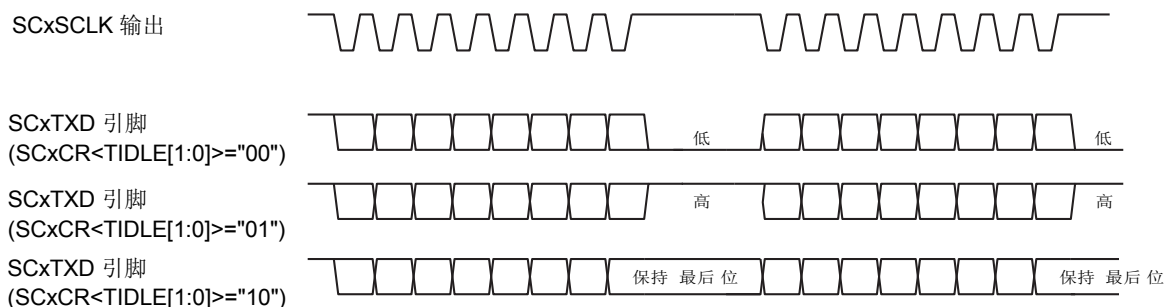


图 13-9 最后位输出后 SCxTXD 引脚电平

13.11.3.5 欠-运行 错误

带时钟输入模式的 I/O 接口模式下且如果 FIFO 为空且如果无数据设置在传输缓冲器中, 下一个帧时钟输入前, 这发生在数据从传输移位寄存器传输完成时, 欠-运行错误发生且 SCxCR<PERR> 设置为 "1".

SCxTXD 引脚电平可以被 SCxCR<TXDEMP> 指定. SCxCR<TXDEMP> 为 "0" 时, 在数据输出期间 SCxTXD 引脚输出 "低" 电平. SCxCR<TXDEMP> 为 "1" 时, SCxTXD 引脚输出 "高" 电平.

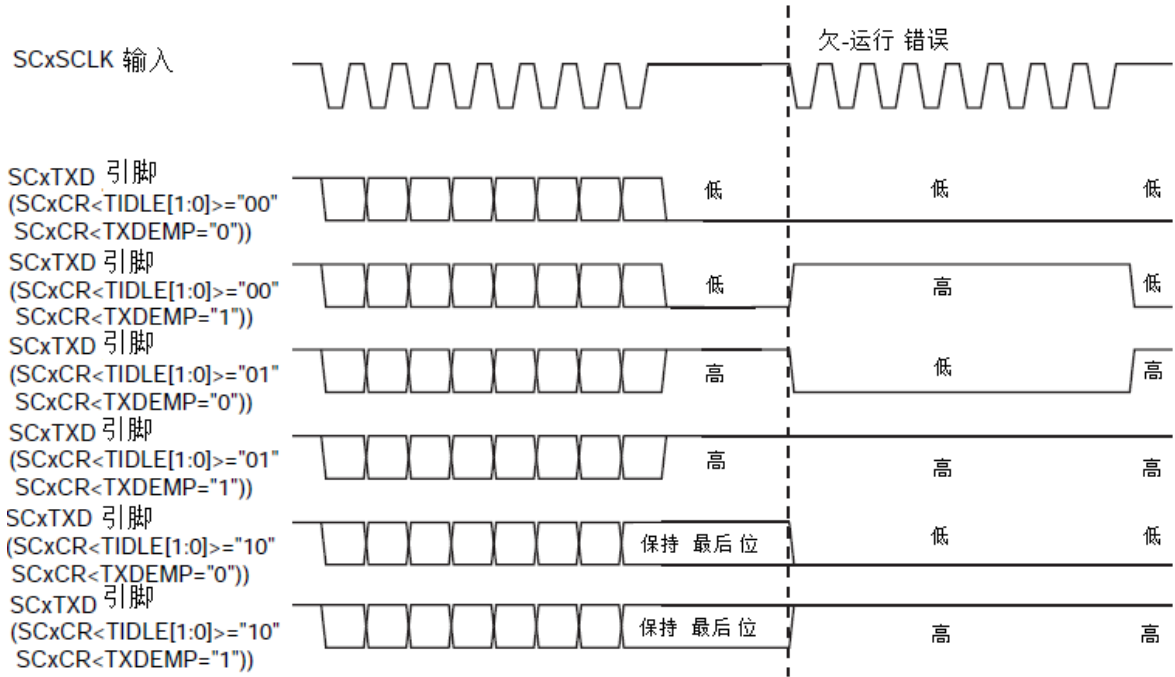


图 13-10 欠-运行错误发生时 SCxTXD 引脚电平

带 SCLK 输出设置的 I/O 接口模式下, 时钟输出自动停止, 所以 SCxCR<PERR> 无意义.

注: 切换带时钟输出模式的 I/O 接口模式到或模式前, 读 SCxCR 且清除欠-运行标志.

13.11.3.6 带时钟输入模式的 I/O 接口模式下数据保持时间

带时钟输入模式的 I/O 接口模式下, 最后位数据保持时间可以通过 SCxCR<EHOLD[2:0]> 被调节. 指定数据保持时间和 SCLK 期间满足以下方程式.

$$\text{最后位数据保持时间} \leq \text{SCLK} / 2 \text{ 期间}$$

13.12 握手功能

握手功能通过使用 $\overline{\text{SCxCTS}}$ (清除为传输) 引脚来使能帧-与-帧数据传输且防止过-运行错误. 此功能可以通过 $\text{SCxMOD0}<\text{CTSE}>$ 使能或禁止.

$\overline{\text{SCxCTS}}$ 引脚设置为"高"电平时, 当前数据传输可以完成但下一个数据传输被挂起直到 $\overline{\text{SCxCTS}}$ 引脚返回 "低" 电平. INTTXx 中断以正常时序生成, 下一个传输数据写入传输缓冲, 且等待直到准备好传输数据.

注 1: 如果传输期间 $\overline{\text{CTS}}$ 信号 设置为 "高" 电平, 当前传输完成后, 下一个数据传输被挂起.

注 2: $\overline{\text{CTS}}$ 设置为"低" 电平后, 数据传输在 TXDCLK 时钟第一个下降缘开始.

尽管无 $\overline{\text{RTS}}$ 引脚被提供, 握手控制功能可以通过分配一位 $\overline{\text{RTS}}$ 功能端口容易地被执行. 通过设置端口为 "高" 电平 在数据接收 (接收中断程序中) 完成时, 可以请求传输 侧挂起数据传输.

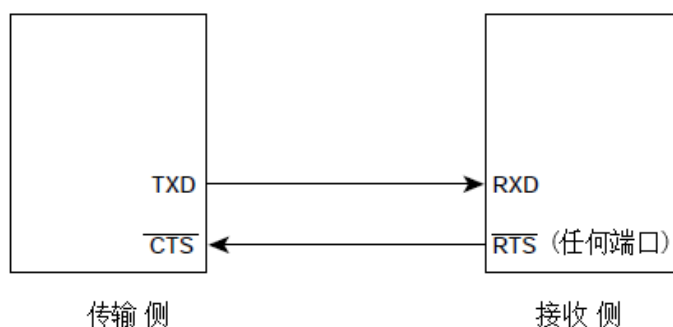


图 13-11 握手 功能

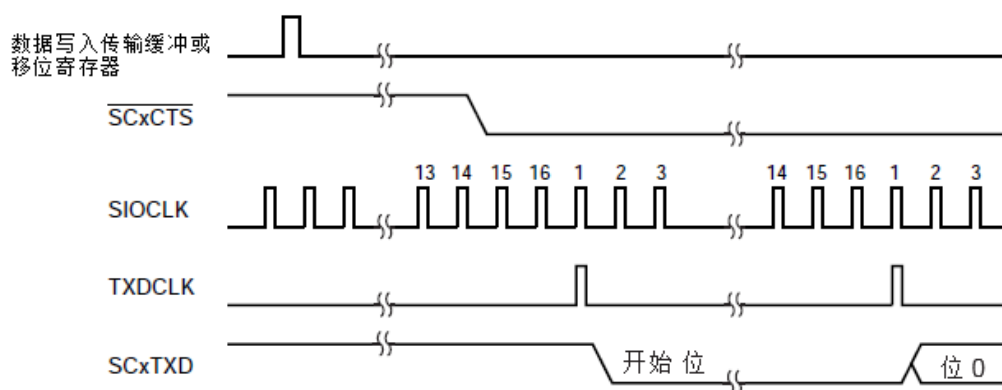


图 13-12 $\overline{\text{SCxCTS}}$ 信号 时序

13.13 中断/错误 生成 时序

13.13.1 接收 中断

图 13-13 表示接收工作和读的路径的数据流。

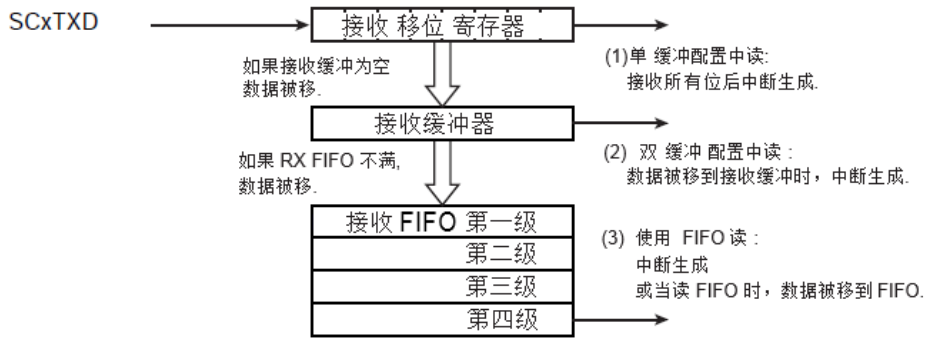


图 13-13 接收 缓冲/FIFO 配置 图

13.13.1.1 单 缓冲 / 双 缓冲

接收中断生成时间取决于传输模式和如下给出的缓冲 配置。

表 13-7 使用单 缓冲 / 双 缓冲的接收中断条件

缓冲配置	UART 模式	IO 接口 模式
单 缓冲	-	最后 SCxSCLK 引脚的上升 / 下降缘后立即 (根据 SCxCR<SCLKS> 设置, 上升或下降被决定.)
双 缓冲	数据从接收移位寄存器传输到接收缓冲时, 接收中断发生. 详细时序为: • 如果数据不存在于接收缓冲, 接收中断发生在第一停止位的中心区域. • 如果数据存在于接收移位寄存器和接收缓冲, 读缓冲时接收中断发生 .	数据从接收移位寄存器传输到接收缓冲时, 接收中断发生.详细时序为: • 如果数据不存在于接收缓冲,在 SCxSCLK 引脚最后位上升/下降缘后, 接收中断立即发生. (设置上升缘下降缘被 SCxCR<SCLKS>指定.) • 如果数据存在于接收移位寄存器和接收缓冲, 读缓冲时接收中断发生.

注:过-运行错误发生时中断不生成.

13.13.1.2 FIFO

FIFO 被用时, 接收中断发生取决于表 13-8 中描述的时序且条件被 SCxRFC<RFIS>指定.

表 13-8 使用 FIFO 的接收中断条件

SCxRFC<RFIS>	中断 条件	中断 生成 时序
"0"	FIFO 充满 电平 (SCxRST<RLVL[2:0]> =生成接收中断 <RIL[1:0]>的接收 FIFO 充满 电平时	<ul style="list-style-type: none"> 接收数据从接收缓冲传输到接收 FIFO 时 从接收 FIFO 读 接收数据时
"1"	FIFO 充满 电平 (SCxRST<RLVL[2:0]> ≥生成接收中断 <RIL[1:0]>的接收 FIFO 充满 电平时	<ul style="list-style-type: none"> 从接收 FIFO 读 接收数据时

13.13.2 传输 中断

图 13-14 表示 传输工作和读的路径的数据流。

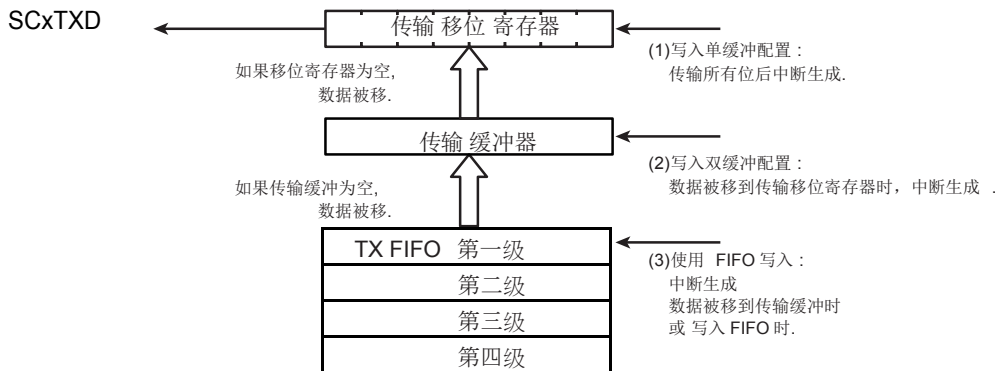


图 13-14 传输缓冲 / FIFO 配置图

13.13.2.1 单缓冲 / 双缓冲

传输中断生成时间取决于如下给出的传输模式和缓冲配置。

表 13-9 使用单缓冲/双缓冲传输中断的条件

缓冲配置	UART 模式	IO 接口 模式
单缓冲	停止位被传输前	最后 SCxSCLK 引脚的上升 / 下降缘后立即 (上升或下降根据 SCxCR<SCLKS> 设置被决定.)
双缓冲	数据从传输缓冲被移到传输移位寄存器时. 如果"1"设置到 SCxMOD1<TXE> 且 传输移位寄存器为空, 因为数据被立即从传输缓冲传输到传输移位寄存器, 传输中断发生.	

13.13.2.2 FIFO

FIFO 被用时,传输中断发生取决于表 13-10 中描述的时序且条件被 SCxTFC<TFIS>指定.

表 13-10 使用 FIFO 时传输中断条件

SCxTFC<TFIS>	中断条件	中断生成 时序
"0"	FIFO 充满 电平 (SCxTST<TLVL[2:0]>) =生成传输中断 <TIL[1:0]> 的传输 FIFO 充满 电平时	<ul style="list-style-type: none"> · 传输 数据 从传输 FIFO 被传输传输缓冲时 · 传输数据 写入到传输 FIFO 时
"1"	FIFO 充满 电平 (SCxTST<TLVL[2:0]>) ≤生成传输中断 <TIL[1:0]>的传输 FIFO 充满 电平时	<ul style="list-style-type: none"> · 传输数据 写入到传输 FIFO 时

13.13.3 错误 生成

13.13.3.1 UART 模式

错误	9 位	7 位 8 位 7 位 + 奇偶 8 位 + 奇偶
成帧 错误 过-运行 错误	停止位中心周围	
奇偶 错误	-	决定: 奇偶位中心周围 标志-改变: 停止位中心周围

13.13.3.2 I/O 接口 模式

过-运行 错误	最后 SCxSCLK 引脚上升 / 下降 缘后 立即 (上升或下降根据 SCxCR<SCLKS> 设置被决定.)
欠-运行 错误	最后 SCxSCLK 引脚上升 / 下降 缘后 立即 (上升或下降根据 SCxCR<SCLKS> 设置被决定.)

注: 时钟输出模式下过-运行错误和欠-运行错误无意义.

13.14 DMA 请求

DMA 传输可以在中断请求的时序开始.

请参考"产品信息"章节关于此产品可以被用作 DMA 请求的通道.

注 1: 使用通过传输或接收中断的 DMA 传输请求时, 使能 DMA 且生成由 SCxMOD<SWRST>的软件复位后, 设置传输和接收寄存器.

注 2: DMA 传输被用时, FIFO 不可以被用.

13.15 软件 复位

软件复位生成通过写入 SCxMOD2<SWRST[1:0]> 为 "10" 然后 "01".

因此, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFL><TXRUN>, SCxCR
<OERR><PERR><FERR> 被初始化. 且接收电路和传输电路变为初始状态. 别的状态维持.

13.16 各模式下工作

13.16.1 模式 0 (I/O 接口 模式)

I/O 接口模式通过设置 SCxMOD<SM[1:0]> 为 "00"选择.

模式 0 由两模式组成, 输出同步时钟 (SCLK)的时钟输出模式和从外部源接受同步时钟 (SCLK)的时钟输入模式.

各模式下禁止 FIFO 工作描述如下. 关于 FIFO, 参考之前描述接收 FIFO 和传输 FIFO.

13.16.1.1 传输

(1) 时钟 输出 模式

- 如果传输双缓冲禁止 (SCxMOD2<WBUF> = "0")

每次 CPU 写入数据到传输缓冲时, 从 SCxTXD 引脚输出数据且从 SCxSCLK 引脚输出时钟. 所有数据输出时,中断 (INTTXx) 生成.

- 如果传输双缓冲使能 (SCxMOD2<WBUF> = "1")

数据写入传输缓冲且移位寄存器为空时, 或数据传输从移位寄存器完成时, 数据从传输缓冲传输到移位寄存器. 同时, SCxMOD2<TBEMP>设置为 "1", 且 INTTXx 中断生成.

如果传输缓冲无被移到传输移位寄存器的数据, INTTXx 中断不生成且时钟输出停止.

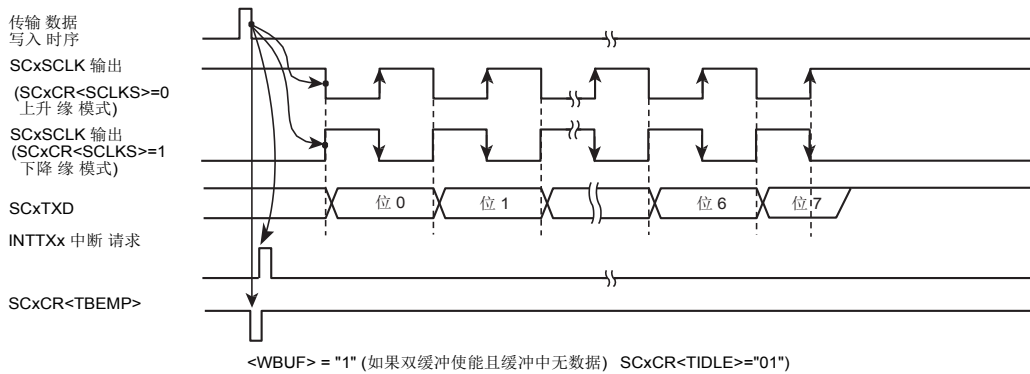
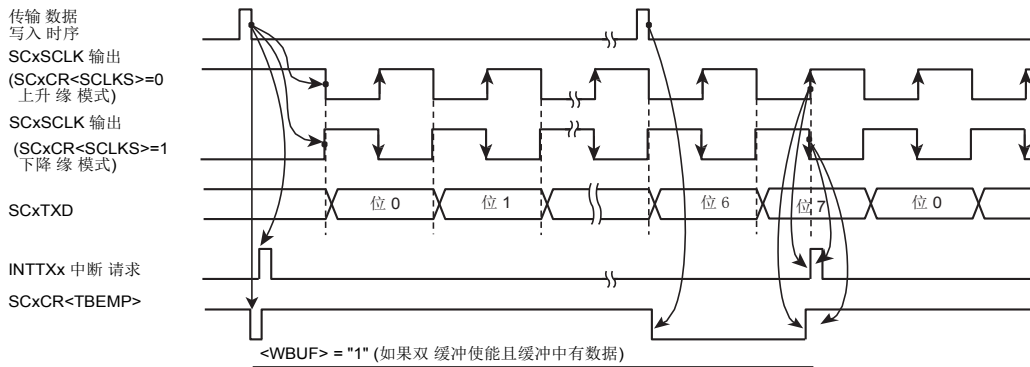
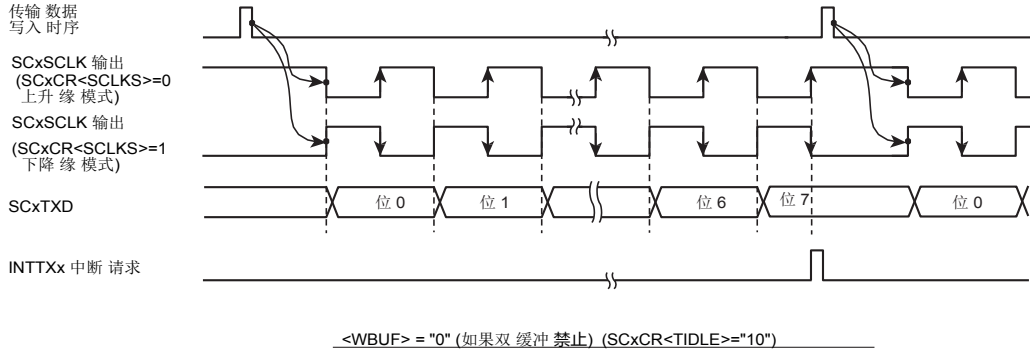


图 13-15 I/O 接口模式(时钟输出模式)下传输工作

(2) 时钟输入 模式

- 如果双缓冲禁止 (SCxMOD2<WBUF> = "0")

如果在写入传输缓冲数据的条件时钟输入, 8-位数据从 SCxTXD 引脚输出. 所有数据输出时, 中断 INTTXx 生成. 图 13-16 所示点 "A" 时序前, 下一个传输数据必须写入.

- 如果双缓冲使能 (SCxMOD2<WBUF> = "1")

时钟输入变为激活前, CPU 写入数据到传输缓冲时, 或数据从传输移位寄存器的传输完成时, 数据从传输缓冲被移到传输移位寄存器. 同时, SCxMOD2<TBEMP> 设置为 "1", 且 INTTXx 中断生成.

如果传输缓冲中无数据时, 时钟输入变为激活, 尽管内部位计数器开始, 欠-运行错误发生且被 SCxCR<TXDEMP>指定的电平输出到 SCxTXD 引脚.

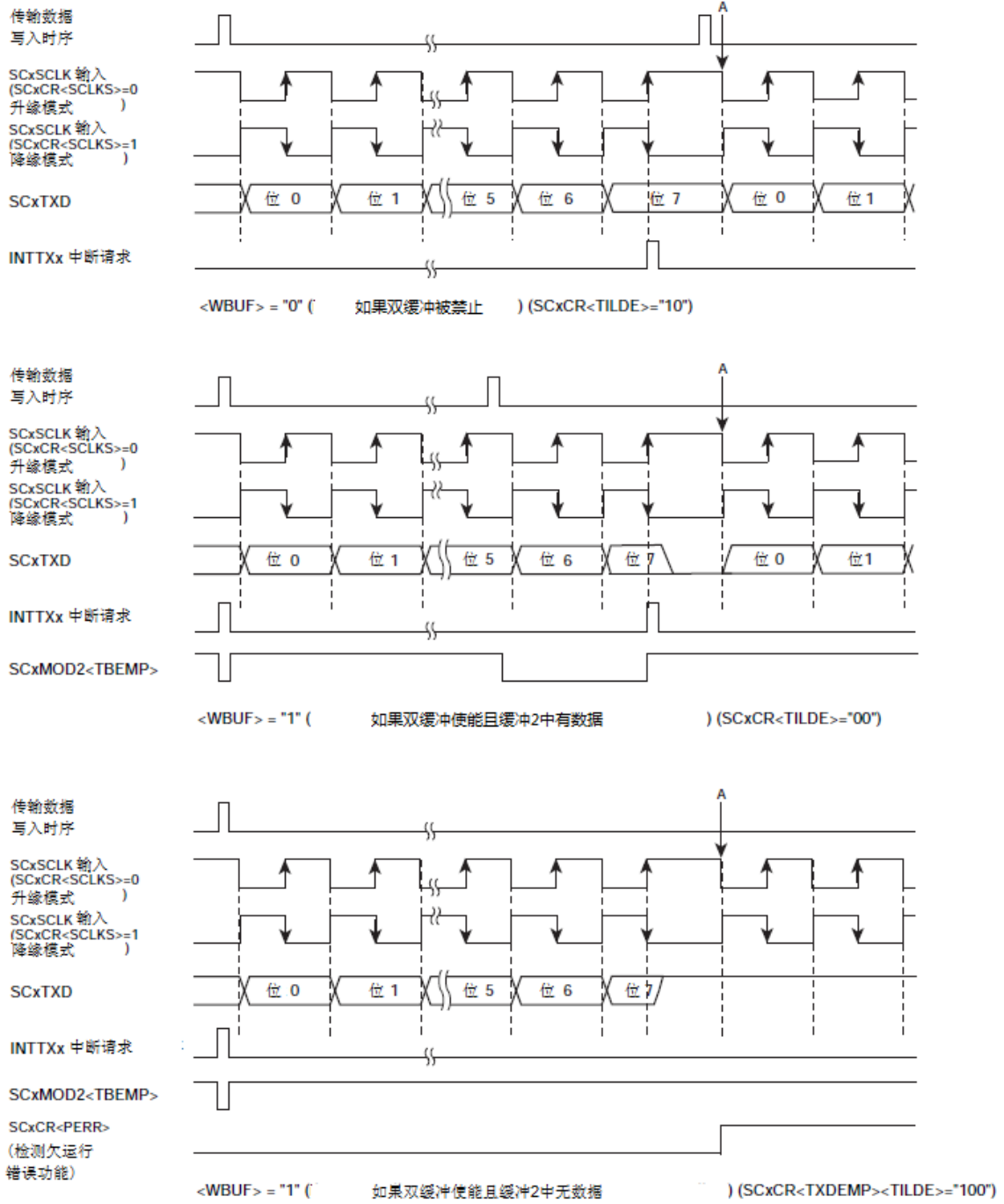


图 13-16 I/O 接口模式(时钟输入模式)下传输工作

13.16.1.2 接收

(1) 时钟输出 模式

通过设置接收使能位 SCxMOD0<RXE> 为"1"时钟输出开始.

- 如果双 缓冲禁止 (SCxMOD2<WBUF> = "0")

每次 CPU 读接收数据, 输出从 SCxSCLK 引脚时钟且下一个数据储存到移位寄存器. 所有 8 位接收时, INTRXx 中断生成.

- 如果双 缓冲使能 (SCxMOD2<WBUF> = "1")

储存在移位寄存器的数据被移到接收缓冲且接收缓冲可以接收下一个帧. 数据从移位寄存器被移到接收缓冲, SCxMOD2<RBFL>设置为"1" 且 INTRXx 生成.

数据在接收缓冲中时, 完成下一个 8 位接收前, 如果不从接收缓冲读数据, INTRXx 中断不生成且时钟输出停止. 此状态下, 从接收缓冲读数据允许移位寄存器的数据移到接收缓冲且这样 INTRXx 中断生成且数据接收恢复.

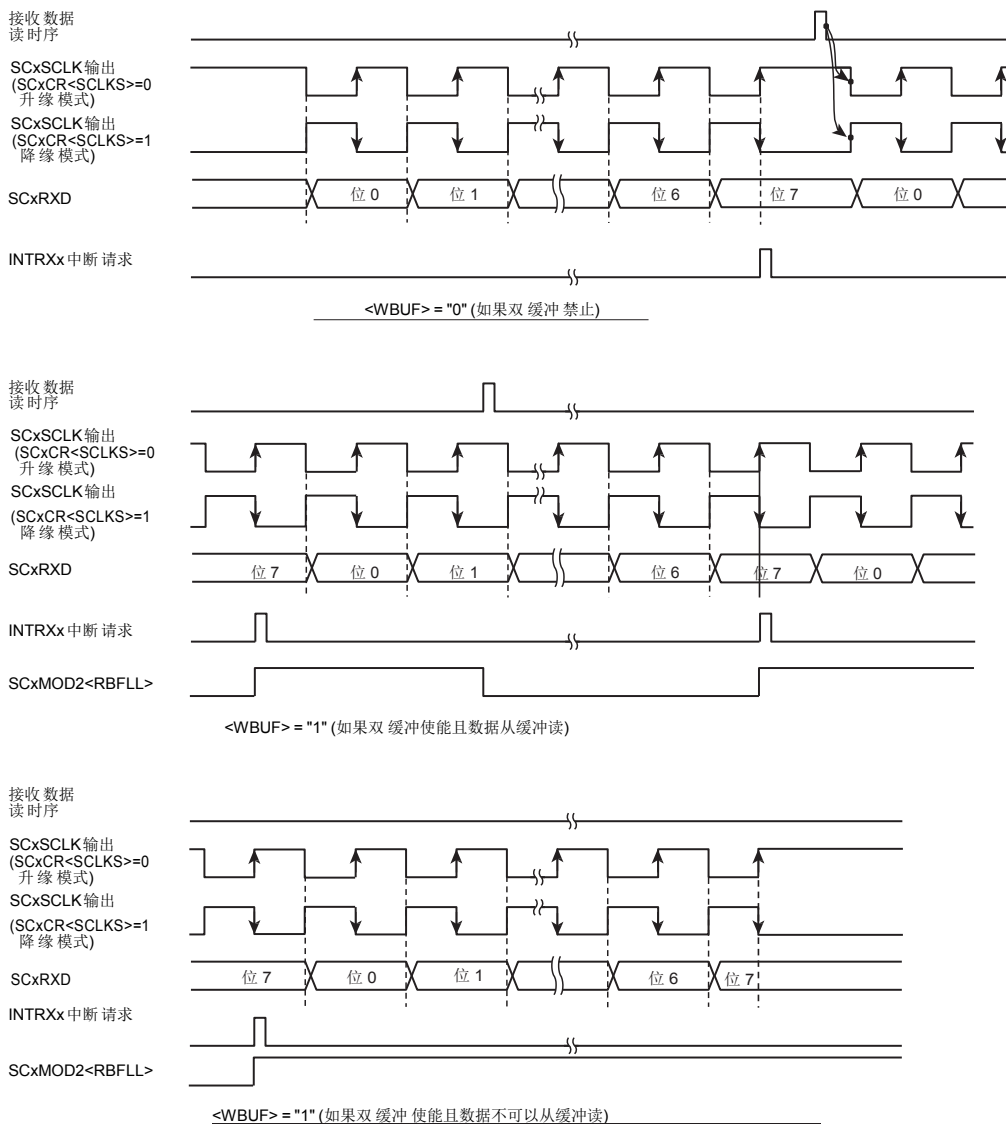


图 13-17 I/O 接口模式(时钟输出模式)下接收工作

(2) 时钟输入模式

时钟输入模式下,接收双缓冲始终使能,接收数据可以从移位寄存器被移到接收缓冲,且接收缓冲逐渐可以接收下一个数据.

每次接收数据被移到接收缓冲,INTRX_x接收中断生成.

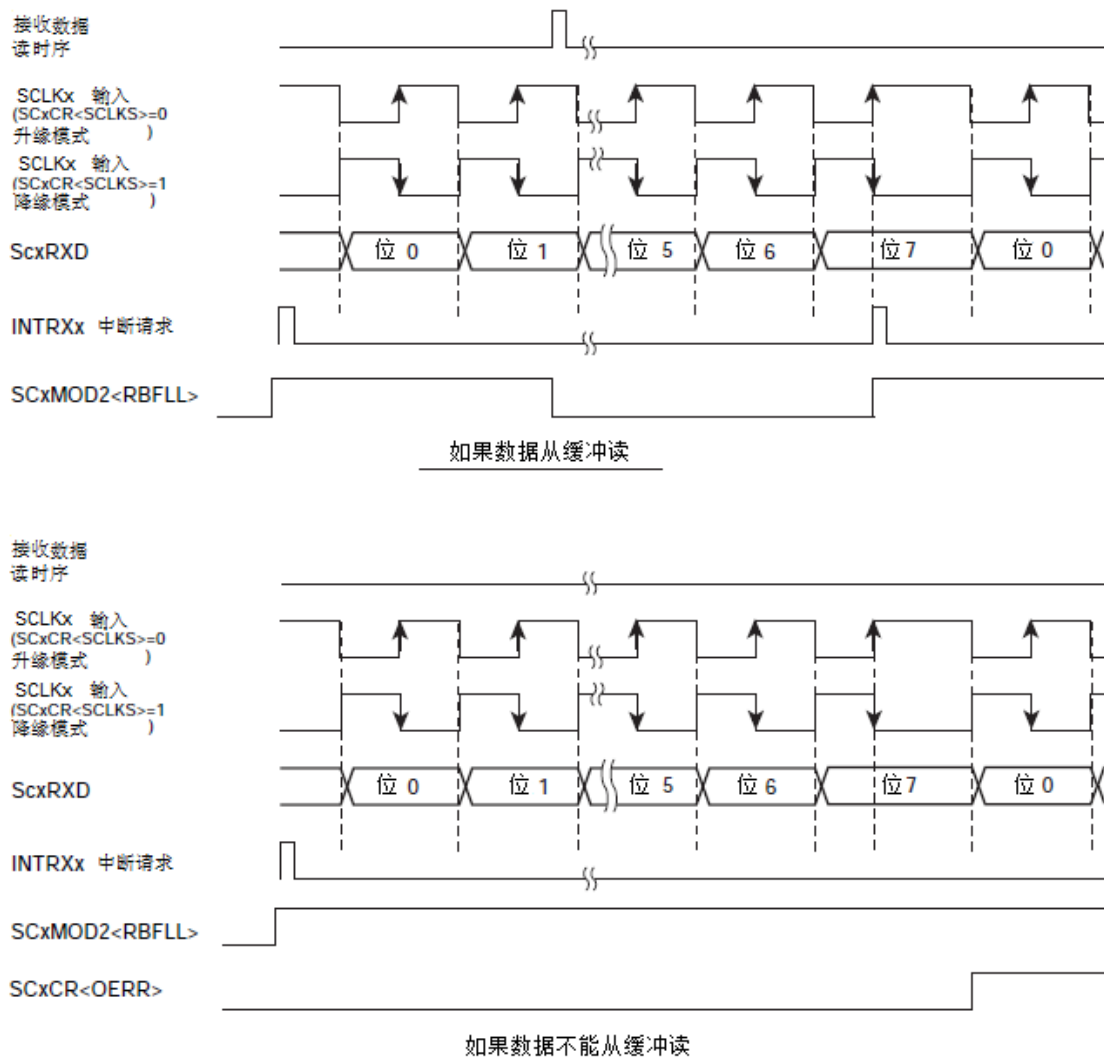


图 13-18 I/O 接口模式(时钟输入模式)下接收工作

13.16.1.3 传输和接收 (全-双工)

(1) 时钟输出模式

- 如果双缓冲禁止 ($SCxMOD2<WBUF> = "0"$)

CPU 写入数据到传输缓冲时, 时钟输出.

随后, 数据移位到接收缓冲且 $INTRXx$ 生成. 同时, 写入传输缓冲的数据从 $SCxTXD$ 引脚输出, 所有数据传输完成时, $INTTXx$ 生成. 然后, 时钟输出停止.

从接收缓冲读数据且下一个传输数据通过 CPU 写入传输缓冲时, 下轮数据传输和接收开始. 读接收缓冲和写入传输缓冲的命令可以自由决定. 仅条件都被满足时, 数据传输恢复.

- 如果双缓冲使能 ($SCxMOD2<WBUF> = "1"$)

CPU 写入数据到传输缓冲时, 时钟输出.

数据移位到接收移位寄存器, 移到接收缓冲, 且 $INTRXx$ 生成. 数据接收时, 传输数据从 $SCxTXD$ 引脚输出. 所有数据被送出时, $INTTXx$ 生成且下一个数据从传输缓冲被移到传输移位寄存器.

如果传输缓冲无被移到传输缓冲 ($SCxMOD2<TBEMP> = "1"$)的数据或接收缓冲为满 ($SCxMOD2<RBFL> = "1"$)时, 时钟输出停止. 两个条件, 读接收数据且写入传输数据, 被满足时, 时钟输出恢复且下轮数据传输且接收开始.

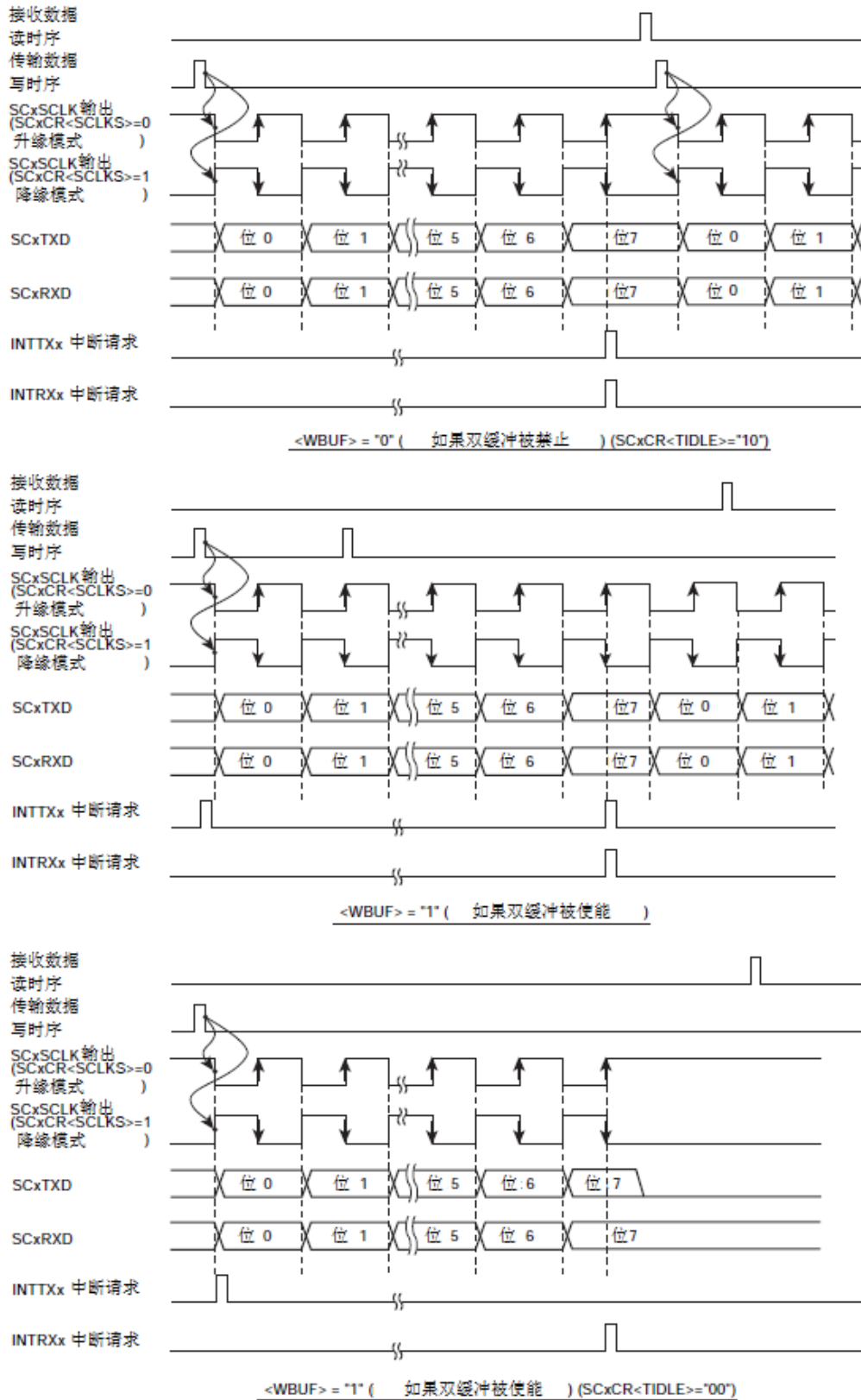


图 13-19 I/O 接口模式 (SCLK 输出模式) 下传输/接收工作

(2) 时钟输入模式

- 如果双缓冲禁止. ($SCxMOD2<WBUF> = "0"$)

接收数据时, 双缓冲始终使能无论 $SCxMOD2<WBUF>$ 设置.

时钟输入变为激活时, 写入传输缓冲中的数据从 $SCxTXD$ 引脚输出且数据被移到接收缓冲. 在数据传输完成时 $INTTx_x$ 生成. 数据接收完成后从移位寄存器被移到接收缓冲时, $INTRx_x$ 生成.

注意下一个数据 (数据必须写入图 13-20 点 A 前)时钟输入前, 传输数据必须写入到传输缓冲. 下一个数据接收完成前数据必须读.

- 如果双缓冲使能. ($SCxMOD2<WBUF> = "1"$)

$INTTx_x$ 生成在传输缓冲数据, 从传输移位寄存器完成数据传输后被移到传输移位寄存器的时序. 同时, 数据接收移位到移位寄存器, 其被移到接收缓冲, 且 $INTRx_x$ 生成.

注意下一个数据时钟输入(数据必须写入图 13-20 中点 A前)前, 传输数据必须写入到传输缓冲. 完成接收下一个数据前数据必须读.

在下一个数据时钟输入, 接收数据同时移位到接收移位寄存器, 从传输移位寄存器 (数据从传输缓冲被移)传输开始.

最后位数据接收时, 如果接收缓冲中数据没被读,过- 运行错误发生.

下一个数据的时钟输入时, 如果没有数据写入传输缓冲, 欠- 运行错误发生. 被 $SCxCR<TXDEMP>$ 指定的电平输出到 $SCxTXD$ 引脚.

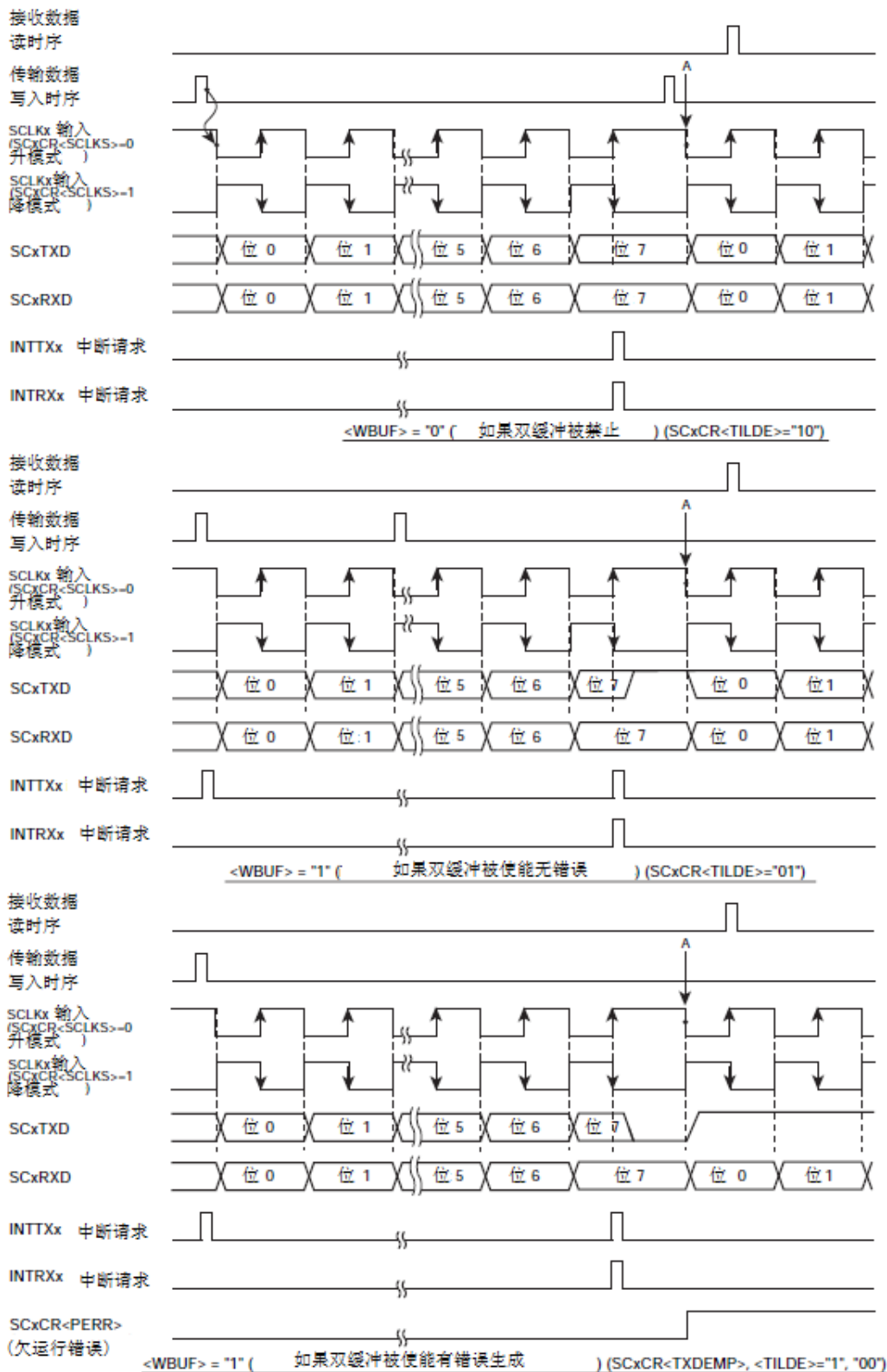


图 13-20 I/O 接口模式下(时钟输入模式)传输/接收工作

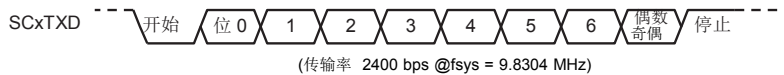
13.16.2 模式 1 (7-位 UART 模式)

7-位 UART 模式通过设置 SCxMOD<SM[1:0]> 为 "01"选择.

此模式下, 奇偶位可以加到传输数据流; SCxCR<PE> 控制奇偶使能/禁止设置.

<PE>设置为"1"(使能)时, 偶数或奇数奇偶可以使用 SCxCR<EVEN>选择. 停止位长可以使用 SCxMOD2<SBLEN>指定.

下表表示以下数据格式中传输控制寄存器设置.



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)

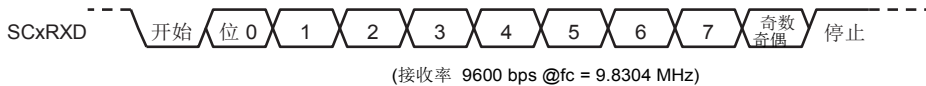
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	设置 7-位 UART 模式
SCxCR	←	x	1	1	x	x	x	0	0	偶数 奇偶 使能
SCxBRCR	←	0	0	1	0	0	1	0	0	设置 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	设置 传输 数据

x: 忽略 - : 不改变

13.16.3 模式 2 (8-位 UART 模式)

8-位 UART 模式通过设置 SCxMOD0<SM[1:0]> 为 "10"选择. 此模式下, 奇偶位可以被加且奇偶使能/禁止使用 SCxCR<PE>控制. 如果 <PE> = "1" (使能), 偶数或奇数奇偶可以使用 SCxCR<ENEN>选择.

以下格式中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	奇数 奇偶 使能
SCxBRCR	← 0	0	0	1	0	1	0	0	设置 9600bps
SCxMOD0	← -	-	1	-	-	-	-	-	接收 使能

x:忽略 - :不改变

13.16.4 模式 3 (9-位 UART 模式)

9-位 UART 模式通过设置 SCxMOD0<SM[1:0]> 为"11"选择.此模式下, 奇偶位必须禁止 (SCxCR<PE> = "0").

最重要位 (第 9 位) 写入到 SCxMOD0<TB8> 传输数据. 数据储存在 SCxCR<RB8> 接收数据.

到/从 缓冲写入或读数据时, 最重要位必须到/从 SCxBUF 写入或读前先写入或读.

停止位长可以使用 SCxMOD2<SBLEN>指定.

13.16.4.1 唤醒 功能

9-位 UART 模式下, 通过设置 SCxMOD0<WU> 为 "1 从机控制器可以在唤醒模式下工作".

此时, 仅 SCxCR<RB8> 设置为"1 "时, 中断 INTRXx 会生成.

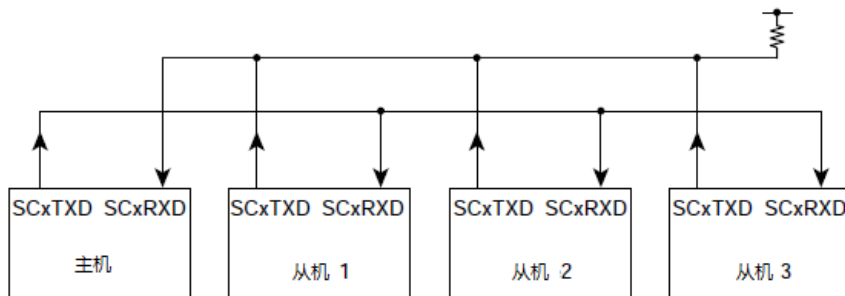
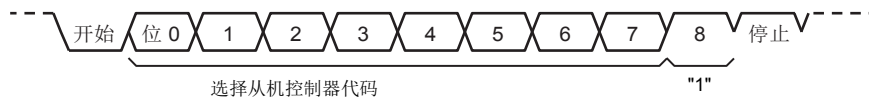


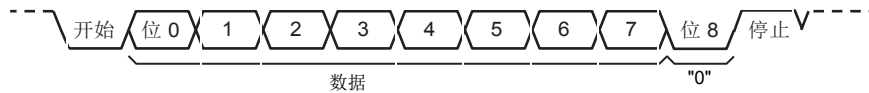
图 13-21 使用唤醒功能的串行链接

13.16.4.2 协议

1. 为主机和从机控制器选择 9-位 UART 模式。
2. 为从机控制器设置 $SCxMOD<WU>$ 为 "1" 使它们准备接收数据。
3. 主机控制器传输包括从机控制器选择代码 (8 位) 的单帧数据. 此时, 最重要位 (位 8) $<TB8>$ 必须设置为 "1".



4. 各从机控制器接收以上数据帧; 如果代码接收匹配控制器自己的选择代码, 清除 $<WU>$ 为 "0".
5. 主机控制器传输数据到指定的从机控制器 (控制器 $SCxMOD<WU>$ 清除为 "0"). 此时, 最重要位 (位 8) $<TB8>$ 必须设置为 "0".



6. 用 $<WU>$ 从机控制器设置为 "1" 忽略接收数据因为最重要位 (位 8) $<RB8>$ 设置为 "0" 且这样无中断 ($INTRXx$) 生成. 同样, 器用 $<WU>$ 设置从机控制为 "0" 可以传输数据到主机控制器来通知数据全部成功接收.

14. 串行 总线 接口 (I2C/SIO)

TMPM475FDFG/FZFG/FYFG 包含串行总线接口 (I2C/SIO), 其中包括以下两工作模式:

- I2C 总线 模式 (有多-主机 能力)
- 时钟-同步 8-位 SIO 模式

以下 解释中, "x" 代表通道数.

14.1 配置

串行总线接口配置图 14-1 所示.

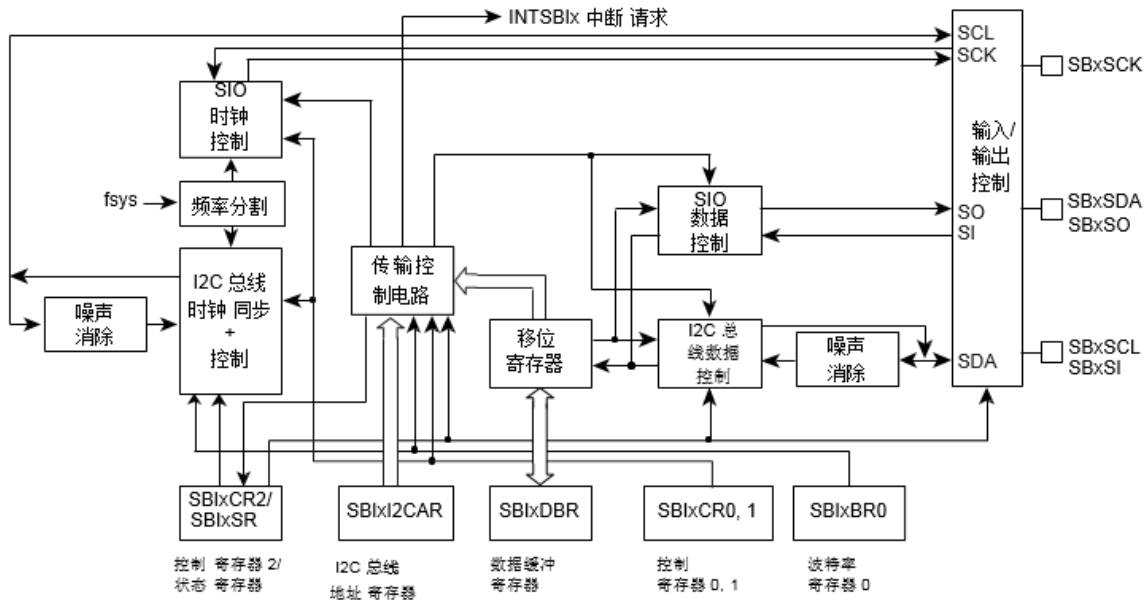


图 14-1 串行总线接口方块图

14.2 寄存器

以下寄存器控制行总线接口且提供其状态信息给监视器。

寄存器如下执行不同功能取决于模式.详细,参考"I2C 总线模式中 14.3.1 控制寄存器"和"14.4.1 SIO 模式控制寄存器".

14.2.1 各通道寄存器

下表表示控制寄存器和其地址.

关于基地址详细,参考"内存映射"章节的"外设功能地址表".

寄存器名		地址(基+)
控制寄存器 0	SBIXCR0	0x0000
控制寄存器 1	SBIXCR1	0x0004
数据缓冲寄存器	SBIXDBR	0x0008
I2C 总线地址寄存器	SBIXI2CAR	0x000C
控制寄存器 2	SBIXCR2 (写)	0x0010
状态寄存器	SBIXSR (读)	
波特率寄存器 0	SBIXBR0	0x0014

14.3 I2C 总线 模式

14.3.1 控制 寄存器 I2C 总线 模式

I2C 总线模式下以下寄存器控制串行总线接口且提供其状态通知给监视器。

14.3.1.1 SBIXCR0(控制 寄存器 0)

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

位	位 符	类型	功能
31-8	-	R	读作 0.
7	SBIEN	R/W	串行总线接口工作 0:禁止 1:使能 要使用串行总线接口, 使能此位先. 第一次设置使能时, 相关 SBI 寄存器可以读或写入. 因为如果此位禁止所有时钟停止除 SBIXCR0, 功耗可以减少通过禁止此位. 如果使能一次后此位禁止, 各寄存器设置被维持.
6-0	-	R	读作 0.

14.3.1.2 SBiXCR1(控制 寄存器 1)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
复位后	0	0	0	0	1	0	0	1(注3)

位	位 符	类型	功能																																																	
31-8	-	R	读作 0.																																																	
7-5	BC[2:0]	R/W	选择每传输位数(注 1) <table border="1"> <thead> <tr> <th rowspan="2"><BC></th> <th colspan="2"><ACK> = 0 时</th> <th colspan="2"><ACK> = 1 时</th> </tr> <tr> <th>时钟 周期数</th> <th>数据 长</th> <th>时钟 周期数</th> <th>数据 长</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>	<ACK> = 0 时		<ACK> = 1 时		时钟 周期数	数据 长	时钟 周期数	数据 长	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>	<ACK> = 0 时		<ACK> = 1 时																																																	
	时钟 周期数	数据 长	时钟 周期数	数据 长																																																
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: 承认时钟脉冲不生成. 1: 承认时钟 脉冲生成. 从机 模式 0: 承认时钟脉冲不计数. 1: 承认时钟 脉冲计数.																																																	
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"> <tbody> <tr> <td>000</td> <td>n = 5</td> <td rowspan="7"> 系统 时钟: fsys { 时钟 齿轮 : fc/1 频率 = $\frac{f_{sys}}{2^n + 72}$ [Hz] } </td> </tr> <tr> <td>001</td> <td>n = 6</td> </tr> <tr> <td>010</td> <td>n = 7</td> </tr> <tr> <td>011</td> <td>n = 8</td> </tr> <tr> <td>100</td> <td>n = 9</td> </tr> <tr> <td>101</td> <td>n = 10</td> </tr> <tr> <td>110</td> <td>n = 11</td> </tr> <tr> <td>111</td> <td>保留</td> <td></td> </tr> </tbody> </table>	000	n = 5	系统 时钟: fsys { 时钟 齿轮 : fc/1 频率 = $\frac{f_{sys}}{2^n + 72}$ [Hz] }	001	n = 6	010	n = 7	011	n = 8	100	n = 9	101	n = 10	110	n = 11	111	保留																																
000	n = 5	系统 时钟: fsys { 时钟 齿轮 : fc/1 频率 = $\frac{f_{sys}}{2^n + 72}$ [Hz] }																																																		
001	n = 6																																																			
010	n = 7																																																			
011	n = 8																																																			
100	n = 9																																																			
101	n = 10																																																			
110	n = 11																																																			
111	保留																																																			
	SWRMON	R	读 <SWRMON>时: 软件 复位 状态 监视器 0: 软件 复位工作处理中. 1: 软件 复位工作不在处理中.																																																	

- 注 1: 切换工作模式为 SIO 模式前清除 <BC[2:0]> 为 "000".
- 注 2: 关于 SCL 线时钟频率的详细, 参考"14.3.2.2 串行时钟"
- 注 3: 复位后, <SCK[0]/SWRMON> 位读作 "1". 然而, 如果 SIO 模式在 SBIXCR2 寄存器选择, <SCK[0]>位的初始值为 "0".
- 注 4: 选择频率的初始值为<SCK[2:0]>=000 且独立于读到的初始值.
- 注 5: <BC[2:0]>="001" 且 <ACK>="0" 主机模式时, 停止条件生成后且别的主机设备不可以使用总线 SCL 线可以通过 SCL 线下降缘被固定到 "L". 总线被连接到几个主机设备时, 停止条件生成前每传输的位数应该设置等于或大于 2.

14.3.1.3 SBIXCR2(控制 寄存器 2)

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

位	位 符	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	开始/停止 条件 生成 0: 停止 条件 生成 1: 开始 条件 生成
4	PIN	W	清除 INTSBIX 中断 请求 0: - 1: 清除 中断 请求
3-2	SBIM[1:0]	W	选择串行总线 接口工作模式 (注 1) 00: 端口模式 (禁止串行总线接口输出) (注 2) 01: SIO 模式 10: I2C 总线模式 (注 3) 11: 保留
1-0	SWRST[1:0]	W	软件 复位 生成 写入 "10" 然后 "01"生成复位. 关于详细, 参考 "14.3.2.16 软件复位".

注 1: 确保通信会话控制期间模式不被改变.

注 2: 确保切换工作模式到端口模式前总线自由.

注 3: 确保切换工作模式从端口模式到 I2C 总线或时钟-同步 8-位 SIO 模式前 SBxSDA 引脚 和 SBxSCL 引脚为"高"平.

注 4: SBIXCR2 用与 SBIXSR 相同的地址分配. 这样, 读-修改-写入工作不可以被用.

14.3.1.4 SBIXSR (状态 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 0.
7	MST	R	主机/从机 选择监视器 0: 从机 模式 1: 主机 模式
6	TRX	R	传输/接收 选择监视器 0: 接收 1: 传输
5	BB	R	I2C 总线 状态监视器 0: 自由 1: 忙
4	PIN	R	INTSBIX 中断 请求监视器 0: 中断 请求 生成 1: 中断 请求 清除
3	AL	R	仲裁丢失 检测 0: - 1: 被检测
2	AAS	R	从机地址匹配 检测 0: - 1: 被检测 (总-呼叫地址也被检测时, 此位设置.)
1	AD0	R	总呼叫检测 0: - 1: 被检测
0	LRB	R	最后接收位 监视器 0: 最后 接收 位 "0" 1: 最后 接收 位 "1"

14.3.1.5 SBiXBRO(串行 总线 接口 波特率 寄存器 0)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	I2SBI	-	-	-	-	-	-
复位后	1	0	1	1	1	1	1	0

位	位 符	类型	功能
31-8	-	R	读 作 0.
7	-	R	读 作 1.
6	I2SBI	R/W	工作在 IDLE 模式 0: 停止 1: 工作
5-1	-	R	读 作 1.
0	-	R/W	确保写入 "0".

14.3.1.6 SBiXDBR (串行 总线 接口 数据 缓冲 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读 作 0.
7-0	DB[7:0]	R	接收 数据
		W	传输 数据

注 1: 传输数据必须从 MSB (位 7)写入寄存器. 接收数据储存在 LSB.

注 2: 因为 SBiXBDR 有写入和读独立的缓冲, 写入数据不可以读. 这样, 读- 修改-写入指令, 诸如位操作, 不可以被用.

14.3.1.7 SBiI2CAR (I2C 总线 地址 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 0.
7-1	SA[6:0]	R/W	SBI 工作为从机设备时, 设置从机地址.
0	ALS	R/W	指定地址识别模式. 0: 识别其从机地址. 1: 不识别其从机地址 (自由-数据格式).

注 1: 请设置 I2C 总线地址寄存器 SBiI2CAR 的位 0 <ALS> 为 "0", 除使用自由数据格式时. 设置为 "1" 时, 工作为自由数

据格式. 选择主机固定到传输. 选择从机固定到接收.

注 2: 从机模式下不要设置 SBiI2CAR 为 "0x00". (如果 SBiI2CAR 设置为 "0x00", 从机模式下, 其识别为从机地址匹配 I2C 标准接收的开始字节 ("0x01").)

14.3.2 控制

14.3.2.1 工作模式

SBIxCR2<SBIM[1:0]> 控制工作模式的设置. 要在 I2C 模式下工作, 设置 <SBIM [1:0]>为"10".

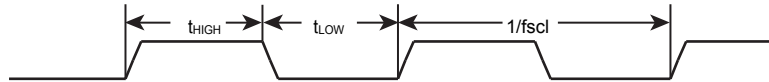
注 1: 确保 SBxSDA 引脚和 SBxSCL 引脚变为端口模式前为"高"电平.

注 2: 确保 SBxSDA 引脚和 SBxSCL 引脚切换工作模式从端口模式到 I2C 总线或时钟-同步 8-位 SIO 模式前为"高"电平.

14.3.2.2 串行 时钟

(1) 时钟 源

主机模式下 SBIxCR1<SCK[2:0]> 指定从 SBxSCL 引脚要输出的最大值串行时钟频率.



$$t_{LOW} = 2^{n-1}/f_{sys} + 58/f_{sys}$$

$$t_{HIGH} = 2^{n-1}/f_{sys} + 14/f_{sys}$$

$$f_{SCL} = \frac{1}{(t_{LOW} + t_{HIGH})} = \frac{f_{sys}}{2^n + 72}$$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

图 14-2 时钟 源

注: 标准和高-速模式下最大值速度根据通信标准分别被指定为 100kHz 和 400kHz. 注意内部 SCL 时钟 频率通过被使用的 f_{sys} 决定且计算方程式以上所示.

(2) 时钟 同步

I2C 总线由于其引脚架构通过使用 线-AND 连接被驱动. 首先拉其时钟线为"低"电平的主机覆盖别的主机生成 "高" 电平 在其时钟线上. 此必须通过主机生成 "高" 电平被检测且响应.

时钟同步确保正确数据传输在有两或更多主机总线上.

比如带两主机总线的, 时钟同步处理如下所示.

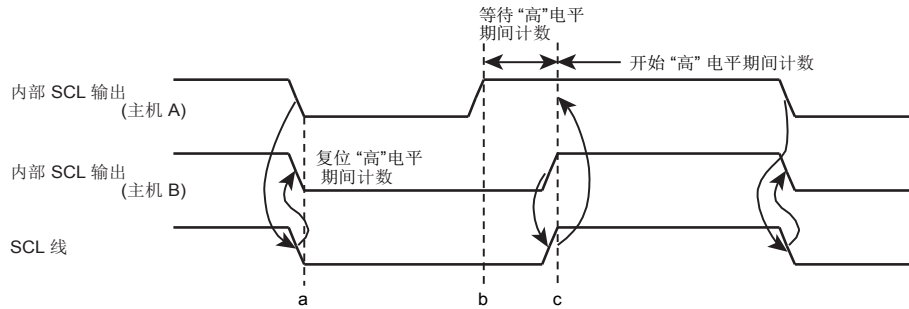


图 14-3 时钟 同步例

在点 a, 主机 A 拉其内部 SCL 输出为 "低" 电平, 带 SCL 总线线为 "低" 电平. 主机 B 检测此过渡, 复位其 "高" 电平期间计数器, 且拉其内部 SCL 输出 电平为 "低" 电平.

在点 b 期间主机 A 完成其 "低" 电平计数, 且带其内部 SCL 输出为 "高" 电平. 然而, 主机 B 仍然保持 SCL 总线线在 "低" 电平, 且计数期间 主机 A 停止其 "高" 电平计数. 主机 A 检测到主机 B 带其内部 SCL 输出为 "高" 电平且带 SCL 总线线为 "高" 电平在点 c 后, 其开始其 "高" 电平 期间计数.

主机完成计数 "高" 电平期后间, 主机拉 SBxSCL 引脚为 "低" 且 SCL 总线线变为 "低".

这样, 在那些被连接到总线之间, 总线上时钟被带最短 "高" 电平期间的主机和带最长 "低" 电平期间的主机决定.

14.3.2.3 设置 获取 模式

设置 SBIxCR1<ACK> 为 "1" 选择获取模式.

工作为主机时, SBI 加一时钟给获取信号.从机模式下, 计数获取信号的时钟.

传输模式下, 在时钟周期从接收接收获取信号期间 SBI 释放 SBxSDA 引脚.

接收模式下, 在时钟周期且生成获取信号期间 SBI 拉 SBxSDA 引脚为 "低" 电平. 同样从机模式下, 如果总-呼叫地址接收, 在电平时钟周期且生成获取信号期间 SBI 拉 SBxSDA 引脚为 "低". 然而, 需要通过软件控制总呼叫第二字节来生成取决于第二字节内容获取信号.

通过设置 <ACK> 为 "0", 非-获取模式被激活. 工作为主机时, SBI 不生成承认信号的时钟.从机模式下, 计数获取信号的时钟.

14.3.2.4 设置 每传输位数

SBIxCR1<BC[2:0]> 指定下一个数据传输或接收的位数.

在开始条件下, <BC[2:0]> 设置为 "000", 使从机地址和方向位以八位包传输. 其它时间, <BC[2:0]> 保持以前编程的值.

14.3.2.5 从地址和地址识别模式

设置 "0" 为 SBIxI2CAR<ALS> 且 SBIxI2CAR<SA[6:0]> 中从机地址设置地址格式, 然后 SBI 识别通过地址格式中主机设备和接收数据传输的从机地址.

如果 <ALS> 设置为 "1", SBI 不识别自由数据格式中的从机地址和接收数据. 自由数据格式情况下, 从机地址和方向位不识别; 开始条件生成后其立即被识别为数据.

14.3.2.6 配置 SBI 为主机或从

设置 SBIxCR2<MST> 为 "1" 配置 SBI 工作为主机设备.

设置 <MST> 为 "0" 配置 SBI 为从机设备.

其检测到总线上停止条件或仲裁丢失时, <MST> 通过硬件清除为 "0".

14.3.2.7 配置 SBI 为传输器或接收器

设置 SBIxCR2<TRX> 为 "1" 配置 SBI 为传输器.

设置 <TRX> 为 "0" 配置 SBI 为接收器.

自由数据格式中如果 SBI 被用, <TRX> 不通过硬件被改变.

地址格式中如果 SBI 被用, <TRX> 设置如下所示.

(1) 主机模式

作为主机模式, 如果 SBI 从从机设备接收承认, <TRX> 通过硬件设置如下所示.

如果 SBI 不承认, <TRX> 维持之前的值.

- 传输方向 位为 "1" 时, <TRX> 设置为 "0"
- 传输方向 位为 "0" 时, <TRX> 设置为 "1".

(2) 从机模式

作为从机模式, 地址格式下, 如果如下条件被满足, <TRX> 设置取决于通过主机设备被传输的方向位.

- 接收从机地址相同于设置在 SBIxI2CAR 中的值时.
- SBI 接收总-呼叫时

<TRX> 设置如下所示.

- 接收方向 位为 "1" 时, <TRX> 设置为 "0".
- 接收方向 位为 "0" 时, <TRX> 设置为 "1".

14.3.2.8 总线 忙 监视器

要确认总线状态, 读 SBIxSR<BB>.

SBI 检测到总线上开始条件时<BB>设置为"1"且 SBI 检测到总线上停止条件时清除为"0".

<BB> 为 "1" 时, 被叫作总线忙. <BB> 为 "0" 时, 被叫作总线自由.

仅总线自由中主机设备可以生成开始条件. 应该确认<BB> 为"0".

<BB> 为 "1" 时, SBI 生成开始条件, 开始条件不生成且仲裁丢失发生.

14.3.2.9 中断 服务 请求和释放

INTSBIx 生成时, SBIxCR2< PIN > 清除为"0" 且 SBI 在中断用请求状态. < PIN > 为 "0" 期间 SBI 拉 SBxSCL 引脚为 "低" 电平.

<PIN>设置为 "1" 时数据写入或从 SBIxDBR 读. 程序写入"1" 到 < PIN >时, 设置为 "1". 然而, 写入 "0" 不清除 < PIN > 为"0".

如果 < PIN >设置为"1", SBxSCL 引脚释放. 从设置 < PIN > 为"1" 到释放 SBxSCL 引脚需要 tLOW.

注: 从机 地址和方向在主机模式中传输时仲裁发生时,
< PIN >清除为 "0" 且 INTSBIx 发生. 此和从机地址是否匹配<SA>不相关.

14.3.2.10 仲裁丢失检测 监视器

I2C 总线有多-主机能力 (有两或更多主机在总线上), 且需要总线仲裁处理确保正确数据传输.

I2C-总线仲裁发生在 SDA 总线线上.

两主机总线上仲裁处理如下所示.

直到点 a, 主机 A 和主机 B 输出相同数据. 在点 a, 主机 A 输出 "低" 电平且主机 B 输出 "高" 电平. 然后主机 A 拉 SDA 总线线为 "低" 电平因为线有线-AND 连接.

SCL 线在点 b 走高时, 从机设备读 SDA 线数据, 比如, 数据通过主机 A 传输. 此时, 数据通过主机 B 传输变为无效.

主机 B 的此条件被叫作"仲裁丢失". 主机 B 释放其 SBxSDA 引脚, 为了其不影响由别的主机初始的数据传输.

如果两或更多主机已经准确传输相同的第一个数据字, 仲裁处理用第二数据字持续.

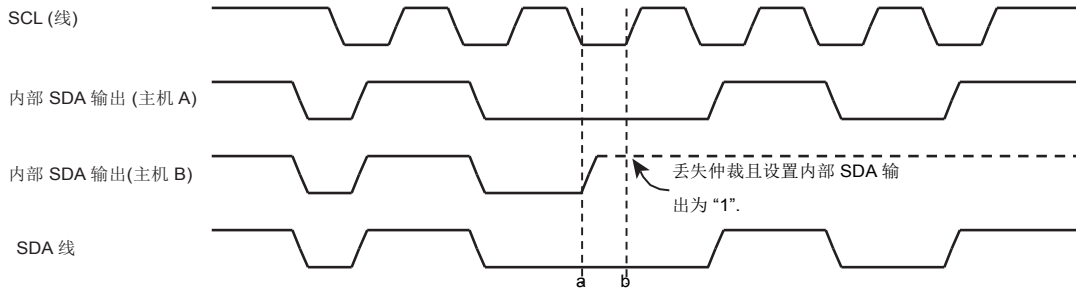


图 14-4 丢失仲裁

主机比较 SDA 总线电平且内部 SDA 在 SCL 线的上升输出电平. 如果这两值之间有区别, 仲裁丢失发生且 $SBIxSR<AL>$ 设置为 "1".

仲裁丢失发生, $SBIxSR<MST>$ 和 $<TRX>$ 清除为 "0", 使 SBI 工作为从机接收器且数据传输期间停止 SCL 时钟输出.

传输从机地址中生成仲裁丢失的设备接收通过别的像从机设备的主机设备传输的从机地址.

接收从机地址匹配 $SBIxI2CAR<SA>$ 时, $<PIN>$ 清除为 "0" 且不匹配时 $INTxSBI$ 发生, $<PIN>$ 维持 "1" 且 $INTxSBI$ 发生.

数据写入或从 $SBIxDBR$ 读或数据写入到 $SBIxCR2$ 时, $<AL>$ 清除为 "0".

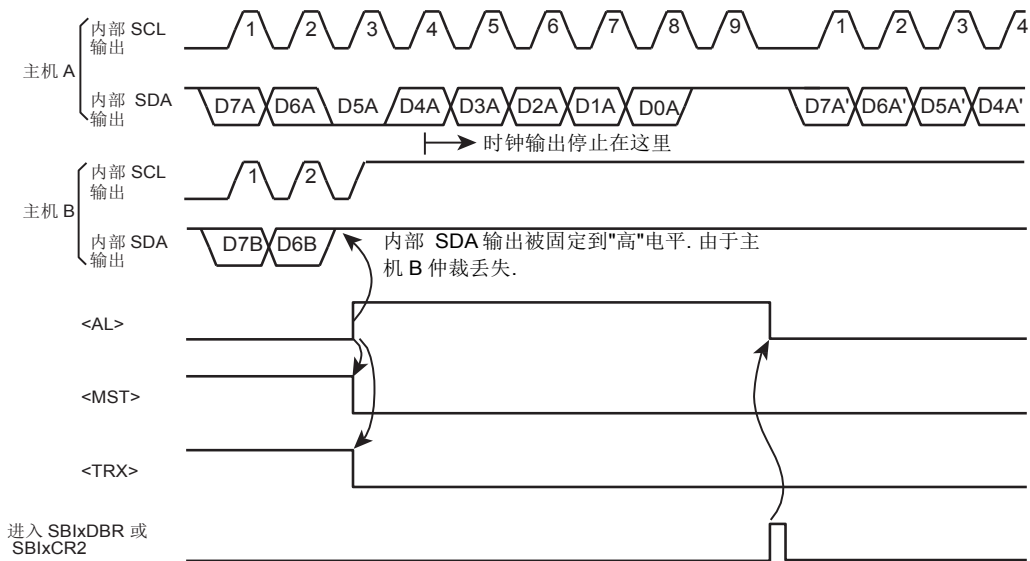


图 14-5 主机 B 丢失仲裁例($D7A = D7B, D6A = D6B$)

14.3.2.11 从 地址 匹配 检测 监视器

地址识别模式 (SBIxI2CAR<ALS>="0")中 SBI 工作为从机设备时, SBIxSR<AAS>设置为 "1" 在接收到匹配在 SBIxI2CAR 被指定的值的总-呼叫地址或从机地址时.

<ALS> 为 "1"时,第一个数据字被接收时, <AAS>设置为 "1".

<AAS>清除为"0" 时 数据写入或 读 从 SBIxDBR.

14.3.2.12 总-呼叫 检测 监视器

SBI 工作为从机设备时, 接收总-呼叫地址时 SBIxSR<AD0>设置为"1" ; 比如, 跟随开始条件的八位都为零.

总线上开始或停止条件被检测时<AD0>清除为 "0" .

14.3.2.13 最后 接收位 监视器

SBIxSR<LRB>设置到 SCL 线上升读到的 SDA 线值.

承认模式中, INTSBIx 中断请求生成后立即读 SBIxSR<LRB> 使 ACK 信号被读.

14.3.2.14 数据 缓冲 寄存器 (SBIxDBR)

读或写入 SBIxDBR 初始读接收的数据或写入传输的数据.

SBI 主机模式下时,写入从机地址和方向位到此 SBIxDBR 中寄存器后, 开始条件生成, SBI 传输从机地址和方向位到从机设备.

14.3.2.15 波特率 寄存器 (SBIxBR0)

进入 IDLE 模式时 SBIxBR0<I2SBI> 寄存器决定 SBI 是否工作.

执行切换 到待机模式指令前, 此寄存器必须编程.

14.3.2.16 软件 复位

如果 SBI 由于外部噪声锁定, 其可以通过使用软件复位初始化.

I2C 总线 模式下, 写入 "10" 然后"01"到 SBIxCR2<SWRST[1:0]> 生成初始化 SBI的复位信号. 写入 SBIxCR2<SWRST[1:0]> 时, 给 I2C 总线模式设置 SBIxCR2<SBIM[1:0]> 为 "10".复位后, 所有控制寄存器和状态标志初始化为其复位值. SBI 初始化时, <SWRST>自动清除为 "00".

注:软件复位使 SBI 工作模式从 I2C 模式切换到端口模式.

14.3.3 数据 传输 处理

14.3.3.1 设备 初始化

首先, 设置 $SBIxCR1<ACK><SCK[2:0]>$. 设置 "1"到 $<ACK>$ 指定承认模式. 设置 "000"到 $SBIxCR1<BC[2:0]>$.

第二, 设置 $<SA[6:0]>$ (从机 地址) 和 $<ALS>$ 到 $SBIxI2CAR$. (地址格式模式中, 设置 $<ALS>="0"$).

最后, 配置串行总线接口为从机接收器, 先确保串行总线接口引脚在 "高".然后写入 "000" 到 $SBIxCR2<MST><TRX><BB>$, "1" 到 $<PIN>$, "10"到 $<SBIM[1:0]>$ 和 "00"到 $<SWRST[1:0]>$.

注: 在所有设备被连接到初始化的总线后任何设备不生成开始条件期间内, 串行总线接口电路初始化必须完成. 如果此规则不遵循, 数据可能不正确接收因为串行总线接口电路完成前, 别的设备可能开始传输初始化.

	7	6	5	4	3	2	1	0	
$SBIxCR1$	← 0	0	0	X	0	X	X	X	指定 ACK 和 SCL 时钟.
$SBIxI2CAR$	← X	X	X	X	X	X	X	X	指定从机地址和地址识别模式.
$SBIxCR2$	← 0	0	0	1	1	0	0	0	配置 SBI 为从机 接收器.

注:X: 忽略

14.3.3.2 生成开始 条件和从 地址

生成开始条件和从机地址需要以下步.

首先, 确保总线自由 ($<BB>="0"$). 然后, 写入 "1" 到 $SBIxCR1<ACK>$ 选择获取模式. 写入 $SBIxDBR$ 从机地址和传输的方向位.

$<BB>="0"$ 时, 写入 "1111"到 $SBIxCR2<MST><TRX><BB><PIN>$ 生成开始条件总线.

伴随开始条件, 从 $SBxSCL$ 引脚 SBI 生成九个时钟.

SBI 用前八时钟,输出从机地址和被 $SBIxDBR$ 指定的方向位,且在第九时钟释放 SDA 线从从机设备接收获取信号.

在第九时钟下降 $INTSBIx$ 中断请求生成, 且 $<PIN>$ 清除为 "0".

$<PIN>="0"$ 时 SBI 保持 SCL 线在"低" 电平. $<TRX>$ 根据在 $INTSBIx$ 中断请求生成的传输的方向位改变其值, 提供从从机设备返回的获取信号.

注: 要输出从机 地址, 写入 $SBIxDBR$ 前用总线自由的软件检查. 如果此规则不遵循,输出在总线上的数据可能被损坏.

主程序中设置

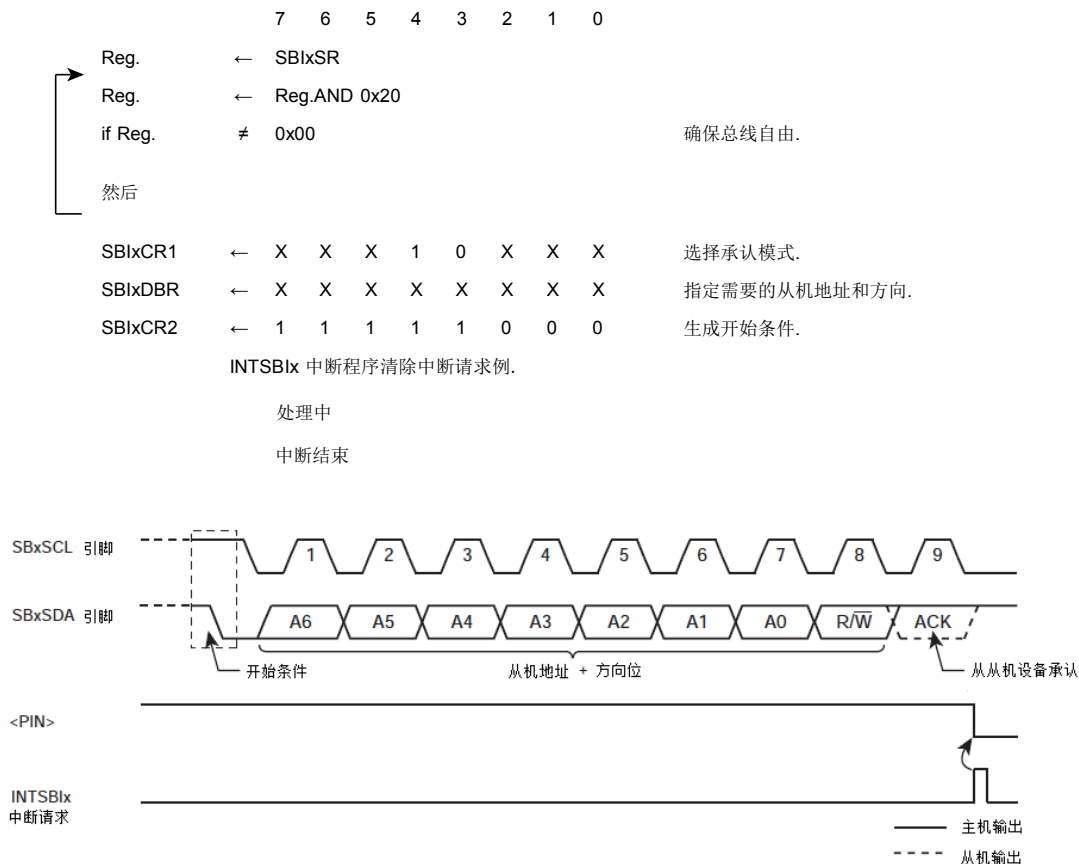


图 14-6 开始条件和从机地址生成

14.3.3.3 传输数据 字

在数据字传输结束, 生成 INTSBIx 中断来测试 <MST>决定 SBI 是否为主机从机模式.

(1) 主机 模式 (<MST> = "1")

测试 <TRX> 决定 SBI 被配置为传输器或接收器.

· 传输器 模式 (<TRX> = "1")

测试 <LRB>. 如果 <LRB> 为"1", 意味接收器不要需要进一步数据. 然后主机生成后面描述的停止条件来停止传输.

如果 <LRB> 为"0",意味接收器需要进一步数据.

如果 下一个传输的数据有八位, 数据写入到 SBIxDBR. 如果数据有不同长, <BC[2:0]> 和 <ACK> 被编程且 传输数据 写入到 SBIxDBR.

写入数据使 <PIN> 为 "1", 使 SBxSCL 引脚生成传输下一个数据字的串行时钟, 且 SBxSDA 引脚来传输数据字.

传输完成后, INTSBIx 中断请求生成, <PIN >清除为"0", 且 SBxSCL 引脚被拉到"低" 电平.

要传输更多数据字, 再测试 <LRB> 且重复以上处理.

INTSBIx 中断

```

if MST = 0
    然后进入从机-模式处理.
if TRX = 0
    然后进入接收器模式处理.
if LRB = 0
    然后进入生成停止条件处理.
    
```

```

SBIxCR1    ←  X  X  X  X  0  X  X  X           指定要传输的位数且
                                                    指定是否需要 ACK.
SBIxDBR    ←  X  X  X  X  X  X  X  X           写入传输数据.
    
```

中断处理结束.

注:X: 忽略

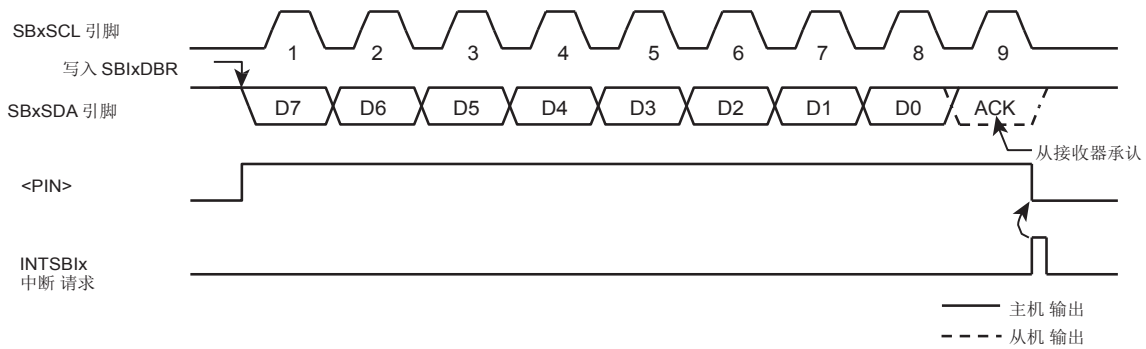


图 14-7 <BC[2:0]>= "000", <ACK>= "1" (传输器 模式)

· 接收器模式 (<TRX> = "0")

如果下一个传输的数据有八位, 传输数据写入到 SB_xDBR. 如果数据有不同长, <BC[2:0]> 被编程且接收数据从 SB_xDBR 释放 SCL 线. (未定义的从机地址传输后, 立即读数据.)

读数据时, <PIN> 设置为"1", 且串行时钟输出到 SB_xSCL 引脚传输下一个数据字. 最后位中, 承认信号变为 "低" 电平时, "0" 输出到 SB_xSDA 引脚.

然后, INTSB_x 中断请求生成, 且 <PIN> 清除为 "0", 拉 SB_xSCL 引脚到 "低" 电平. 每次从 SB_xDBR 读接收的数据, 一字传输时钟且输出承认信号.

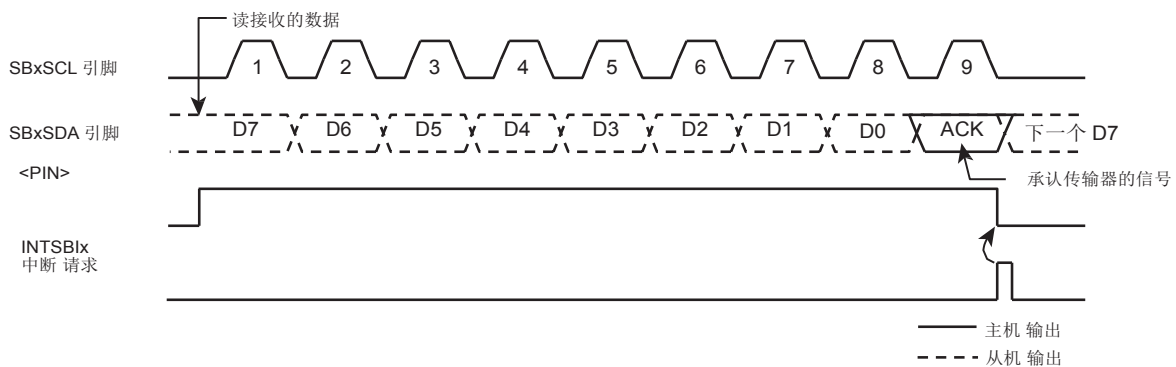


图 14-8 <BC[2:0]>= "000", <ACK>= "1" (接收器 模式)

要中止从传输器数据传输, 读倒数第二数据字前 <ACK> 必须立即清除为 "0". 此禁止最后数据字承认时钟生成. 传输完成时, 中断请求生成. 中断处理后, <BC[2:0]> 必须设置为 "001" 且 数据必须读为了时钟生成给 1-位 传输. 此时, 主机接收器保持 SDA 总线线在 "高" 电平, 其传输到传输器结束的信号为承认信号.

中断中止 1-位 数据接收的处理中, 停止条件生成中止数据传输.

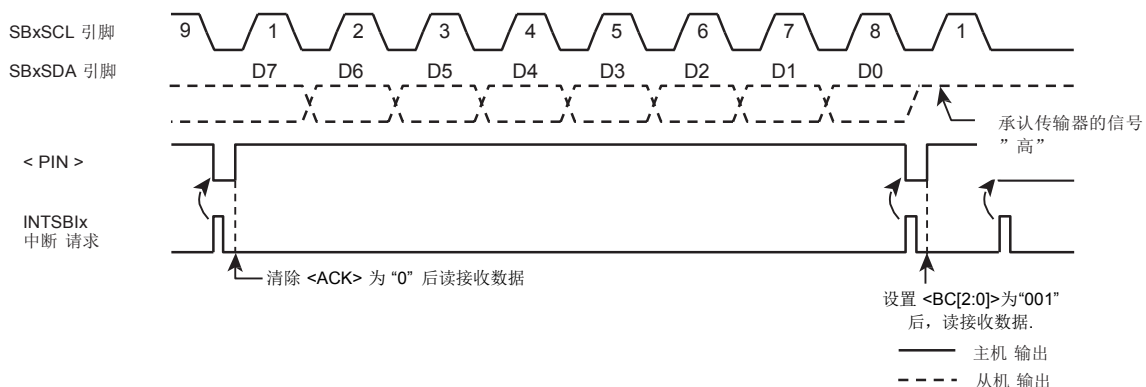


图 14-9 中止数据传输主机接收器模式

例: 接收 N 数据字时

INTSB_{ix} 中断 (数据 传输后)

		7	6	5	4	3	2	1	0	
SBI _x CR1	←	X	X	X	X	0	X	X	X	设置接收的数据位数且指定是否需要 ACK.
Reg.	←	SBI _x DBR								读虚拟数据.

中断结束

INTSB_{ix} 中断 (第一到第(N-2) 数据接收)

		7	6	5	4	3	2	1	0	
Reg.	←	SBI _x DBR								读第一到第(N-2) 数据字.

中断结束

INTSB_{ix} 中断 (第(N-1) 数据接收)

		7	6	5	4	3	2	1	0	
SBI _x CR1	←	X	X	X	0	0	X	X	X	禁止承认时钟生成.
Reg.	←	SBI _x DBR								读第 (N-1) 数据字.

中断结束

INTSB_{ix} 中断 (第 N 数据接收)

		7	6	5	4	3	2	1	0	
SBI _x CR1	←	0	0	1	0	0	X	X	X	禁止承认时钟生成.
Reg.	←	SBI _x DBR								读第 N 数据字.

中断结束

INTSB_{ix} 中断 (完成数据接收后)

生成停止条件的处理.	中止数据传输.
中断结束	

注:X: 忽略

(2) 从机 模式 (<MST> = "0")

从机模式下, SBI 接收任何从机地址或总-呼叫从主机设备时, SBI 接收其从机地址或总-呼叫后 SBI 完成传输数据时, SBI 生成 INTSBIx 中断请求.而且, 如果 SBI 检测仲裁丢失在主机模式中,其切换到从机模式. 数据字传输中仲裁丢失被检测完成时, INTSBIx 中断请求生成. INTSBIx 中断请求时, <PIN>清除为 "0", 且 SBxSCL 引脚被拉到"低" 电平. 数据写入或从 SBIxDBR 读时或< PIN >设置为 "1 "时, t_{LOW} 期间后 SBxSCL 引脚释放.

另外, 通过软件需要根据第二字节内容控制 ACK 信号.

从机模式下, 正常从机模式处理时或由仲裁丢失导致的处理被执行且从主机模式变为从机模式.

测试 SBIxSR<AL>, <TRX>, <AAS> 和 <AD0> 决定需要的处理.

"表 14-1 从机模式下处理"表示 从机模式状态和需要的处理.

例: 接收从机地址匹配 SBI 自己的地址且从机接收器模式下方向位为 "1"时 .

INTSBIx 中断

if TRX = 0
然后进入别的处理.

if AL = 0
然后进入别的处理.

if AAS = 0
然后进入别的处理.

SBIxCR1	←	X	X	X	1	0	X	X	X	设置传输的位数.
SBIxDBR	←	X	X	X	X	X	X	X	X	设置传输数据.

注:X:忽略

表 14-1 从机 模式中处理

<TRX>	<AL>	<AAS>	<AD0>	状态	处理
1	1	1	0	从机地址被传输且 SBI 接收通过另一个主机传输的带方向位 "1"从机地址时, 仲裁丢失被检测.	用数据字设置位数到 <BC[2:0]> 且写入传输数据到 SBIXDBR.
	0	1	0	从机接收器模式下, SBI 接收通过主机传输的带方向位 "1"从机地址.	
		0	0	0	从机传输器模式下, SBI 完成一数据字的传输.
0	1	1	1/0	从机地址被传输, 且 SBI 接收带方向位 "0" 从机地址或通过别的主机传输的总-呼叫地址时, 仲裁丢失被检测.	读 SBIXDBR (虚拟读) 来设置 <PIN> 为 1, 或写入 "1"到 <PIN>.
		0	0	从机地址 或数据字被传输时, 仲裁丢失被检测, 且传输被终止.	
	0	1	1/0	从机接收器模式下, SBI 接收带方向位 "0"从机地址或通过主机传输的总-呼叫地址.	
		0	1/0	从机接收器模式下, SBI 完成数据字接收.	

14.3.3.4 生成停止 条件

SBIXSR<BB> 为 "1"时, 写入 "1" 到 SBIXCR2<MST>, <TRX>, <PIN> 且 "0" 到 <BB> 使 SBI 开始总线上生成停止条件顺序.

不要改变<MST>, <TRX>, <BB>, <PIN> 内容直到停止条件出现在总线上.

如果别的设备沿着 SCL 总线线保持, SBI 等待直到 SCL 线释放.

然后, SBxSDA 引脚走 "高",使停止条件生成.

```

          7   6   5   4   3   2   1   0
SBIXCR2 ← 1   1   0   1   1   0   0   0   生成 停止 条件.

```

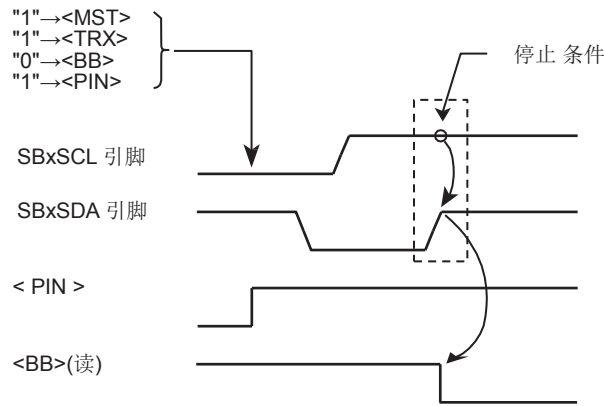


图 14-10 生成停止条件

14.3.3.5 重新开始 处理

主机设备改变数据传输方向而不中止从机设备的传输时，重启被用。主机模式下生成重启的处理如下描述。

首先，写入 $SBIxCR2<MST>$ 、 $<TRX>$ 、 $<BB>$ 为 "0" 且写入 "1" 为 $<PIN>$ 释放总线。此时， $SBxSDA$ 引脚被保持在 "高" 电平且 $SBxSCL$ 引脚释放。因为总线上没有停止条件生成，别的设备识别总线为忙。然后，测试 $SBIxSR<BB>$ 且等待直到其变为 "0" 来确保 $SBxSCL$ 引脚释放。接下来，测试 $<LRB>$ 且等待直到它变为 "1" 来确保没有别的设备拉 SCL 总线线到 "低" 电平。一旦通过遵循以上处理决定总线为自由，遵循 "14.3.3.2 生成开始条件和从机地址" 中描述的生成开始条件的处理。

要满足重启的设置时间，总线被决定为自由后，必须通过软件制作至少 $4.7\mu s$ 等待期间 (标准模式下)。

注 1: 其为 "0" 时不要写入 $<MST>$ 为 "0"。(重启不可以初始。)

注 2: 主机设备作为接收器时，生成重启前从用为传输器的从机设备的数据传输必须完成。要完成数据传输，从机设备必须接收 "高" 电平承认信号。因此，生成重启前 $<LBR>$ 变为 "1"， SCL 线上升缘不被检测即使通过以下重启处理确认 $<LBR>= "1"$ 。要检查 SCL 线状态，读端口。



注: X: 忽略

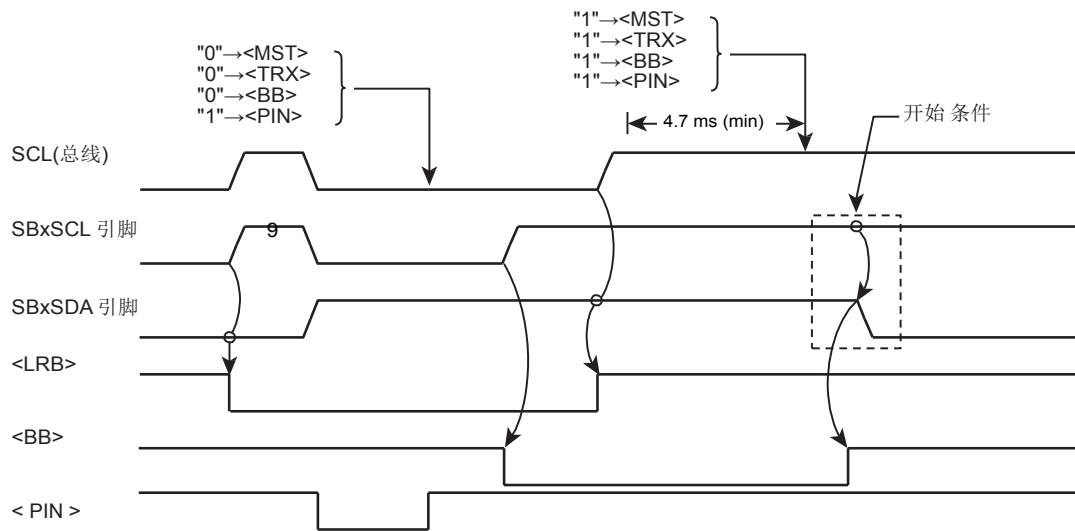
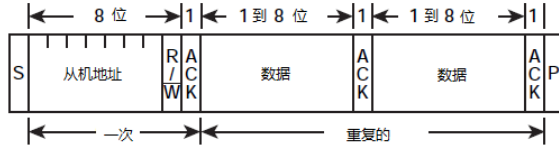


图 14-11 生成重开始时序图

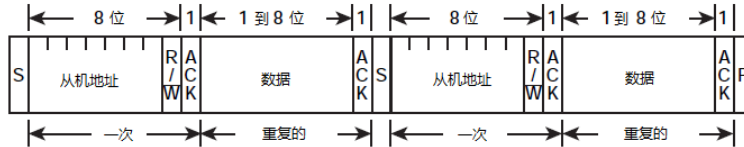
14.3.4 数据 格式

图 14-12 表示 I2C 总线模式下被用的数据格式.

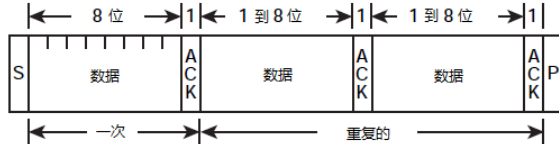
(a) 地址格式



(b) 地址格式 (带重复的开始条件)



(c) 自由数据格式 (主机-传输器到从机-接收器)



注) S: 开始 条件
R/W: 方向 位
ACK: 承认 位
P: 停止 条件

图 14-12 I2C 总线模式数据格式

14.3.5 关于多主机使用注意事项

通信在锁定状态下多-主机模式下通过软件准备恢复处理.

恢复处理例

1. 开始定时器时间结束检测与开始通信同步.
2. 如果被指定时间内串行接口中断 (INTSBIx)不发生, 时间结束发生且 MCU 决定通信被锁定.
3. 在串行总线接口做软件复位来释放通信锁定的条件.
4. 调节传输时序 (注)
5. 再传输传输数据.

注: 调节 MCU 之间传输时序避免重叠传输时序.

14.4 SIO 模式

14.4.1 SIO 模式控制 寄存器

时钟-同步 8-位 SIO 模式下，以下寄存器控制串行总线接口和提供其监视器状态信息。

14.4.1.1 SBIXCR0(控制 寄存器 0)

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

位	位符	类型	功能
31-8	-	R	读作 0.
7	SBIEN	R/W	串行总线接口工作. 0: 禁止 1: 使能 使用串行总线接口前，使能 此 位. 如果此位禁止, 功耗可以减少因为所有时钟停止除 SBIXCR0. 如果串行总线接口工作使能然后禁止, 设置会维持在各寄存器.
6-0	-	R	读作 0.

14.4.1.2 SBiXCR1(控制 寄存器 1)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	SIOS	SIOINH	SIOM		-	SCK		
复位后	0	0	0	0	1	0	0	0(注 1)

位	位符	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 中止																	
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	写入 <SCK[2:0]>时: 选择串行时钟频率. (注 1)																	
			<table border="1" style="margin-left: 20px;"> <tr> <td>000</td> <td>n = 3</td> <td rowspan="8" style="vertical-align: middle;"> $\left. \begin{array}{l} \text{系统 时钟: } f_{\text{sys}} \\ \text{时钟 齿轮: } fc/1 \\ \text{频率} = \frac{f_{\text{sys}}/2}{2^n} \text{ [Hz]} \end{array} \right\}$ </td> </tr> <tr> <td>001</td> <td>n = 4</td> </tr> <tr> <td>010</td> <td>n = 5</td> </tr> <tr> <td>011</td> <td>n = 6</td> </tr> <tr> <td>100</td> <td>n = 7</td> </tr> <tr> <td>101</td> <td>n = 8</td> </tr> <tr> <td>110</td> <td>n = 9</td> </tr> <tr> <td>111</td> <td>-</td> </tr> </table>	000	n = 3	$\left. \begin{array}{l} \text{系统 时钟: } f_{\text{sys}} \\ \text{时钟 齿轮: } fc/1 \\ \text{频率} = \frac{f_{\text{sys}}/2}{2^n} \text{ [Hz]} \end{array} \right\}$	001	n = 4	010	n = 5	011	n = 6	100	n = 7	101	n = 8	110	n = 9	111	-
000	n = 3	$\left. \begin{array}{l} \text{系统 时钟: } f_{\text{sys}} \\ \text{时钟 齿轮: } fc/1 \\ \text{频率} = \frac{f_{\text{sys}}/2}{2^n} \text{ [Hz]} \end{array} \right\}$																		
001	n = 4																			
010	n = 5																			
011	n = 6																			
100	n = 7																			
101	n = 8																			
110	n = 9																			
111	-																			

注 1: 复位后, <SCK[0]> 位读作 "1". 然而, 如果 SIO 模式在 SBiXCR2 寄存器选择, 初始值读作 "0". 此文档, 写入栏 "复位后"的值为初始状态下设置 SIO 模式后的值. SBiXCR2 寄存器和 SBiXSR 寄存器描述相同.

注 2: 编程传输模式和串行时钟前, 设置 <SIOS> 为 "0" 且 <SIOINH>描述"1".

14.4.1.3 SBixDBR (数据 缓冲 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读 作 0.
7-0	DB[7:0]	R	接收 数据
		W	传输 数据

注 1: 传输数据必须从 MSB (位 7) 写入寄存器. 接收数据储存在 LSB.

注 2: 因为 SBixDBR 有独立的写入和读的缓冲, 写入的数据不可以读. 这样, 读- 修改-写入指令, 诸如位操作, 不可以被用.

14.4.1.4 SBIXCR2(控制 寄存器 2)

此寄存器通过写入它用作 SBIXSR 寄存器.

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	-	-	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	位符	类型	功能
31-8	-	R	读作 "0".
7-4	-	R	读作 1. (注 1)
3-2	SBIM[1:0]	W	选择串行总线接口工作模式 (注 2) 00: 端口 模式 01: SIO 模式 10: I2C 总线 模式 11: 保留
1-0	-	R	读作 1. (注 1)

注 1: 此文档中, 写入栏 "复位后" 的值为初始状态下设置 SIO 模式后的值.

注 2: 确保通信会话控制期间, 模式不被改变.

14.4.1.5 SBIXSR (状态 寄存器)

此寄存器通过写入它用作 SBIxCR2.

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	-	-	SIOF	SEF	-	-
复位后	1(注)	1(注)	1(注)	1(注)	0	0	1(注)	1(注)

位	位 符	类型	功能
31-8	-	R	读 作 0.
7-4	-	R	读 作 1.(注)
3	SIOF	R	串行 传输 状态 监视器. 0: 完成 1: 处理中
2	SEF	R	移位 工作 状态 监视器 0: 完成. 1: 处理中
1-0	-	R	读 作 1.(注)

注:此文档中, 写入栏 " 复位后"的值为初始状态下设置 SIO 模式后的值.

14.4.1.6 SBiXBRO (波特率寄存器 0)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	I2SBI	-	-	-	-	-	-
复位后	1	0	1	1	1	1	1	0

位	位符	类型	功能
31-8	-	R	读作 0.
7	-	R	读作 1.
6	I2SBI	R/W	IDLE 模式下工作. 0: 停止 1: 工作
5-1	-	R	读作 1.
0	-	R/W	确保写入 "0".

14.4.2 控制

14.4.2.1 串行 时钟

(1) 时钟 源

内部或外部时钟可以通过编程 SBIxCR1<SCK[2:0]>选择.

(a) 内部 时钟

内部时钟模式中,七个频率之一可以选择作串行时钟, 其被输出到外面通过 SBxSCK 引脚.传输开始时, SBxSCK 引脚输出变为 "高" 电平.

写入传输数据或读 接收数据时, 如果程序不能与此串行时钟率保持一致, SBI 自动进入等待期间.在此期间, 串行时钟自动停止且下一个移位工作被挂起直到处理完成.

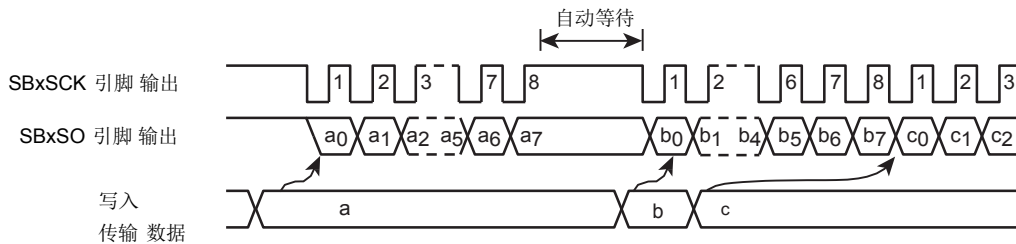


图 14-13 自动 等待

(b) 外部 时钟 (<SCK[2:0]> = "111")

SBI 使用从外面提供到 SBxSCK 引脚的外部时钟作串行时钟.

为了恰当的移位工作, 在 "高" 和 "低" 电平的串行时钟必须有如下所示脉宽.

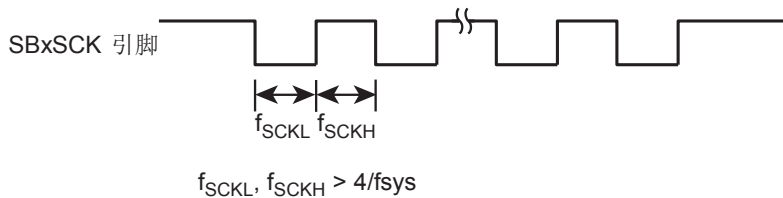


图 14-14 外部时钟输入的最大传输频率

(2) 移位 缘

前切-缘移位被用在传输中. 后切-缘移位被用在接收中.

- 前切-缘 移位

数据移位 在串行时钟 (或 SBxSCK 引脚 输入/输出的下降 缘)的前切 缘.

- 后切-缘 移位

数据移位 在串行时钟 (或 SBxSCK 引脚 输入/输出的上升 缘)的后切 缘.

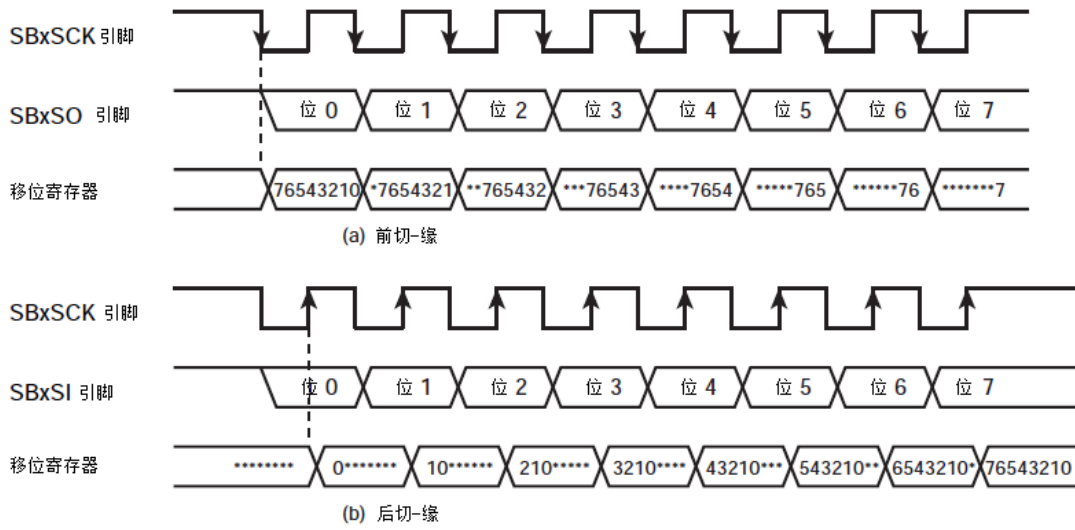


图 14-15 移位 缘

14.4.2.2 传输模式

传输模式,接收模式或传输/接收模式可以通过编程 SBIxCR1<SIOM[1:0]>选择.

(1) 8-位传输模式

设置控制寄存器到传输模式且写入传输数据到 SBIxDBR.

写入传输数据后,写入 "1" 到 SBIxCR1<SIOS> 开始传输. 传输数据从 SBIxDBR 被移到移位寄存器且输出到 SBxSO 引脚,先用最低有效位 (LSB),同步串行时钟. 一旦传输数据传输到移位寄存器, SBIxDBR 变为空,且 INTSBIx (缓冲-空) 中断生成,请求下一个传输数据.

内部时钟模式下,如果 8-位数据全部被传输后,下一个数据不被加载,串行时钟会停止且自动进入等待状态. SBIxDBR 用下一个传输数据被加载时,等待状态会清除.

外部时钟模式下, SBIxDBR 必须用下一个数据移位工作开始前的数据加载. 因此,数据传输率变化取决于中断请求生成时和 SBIxDBR 用中断服务程序中的数据加载时之间的最大延迟.

传输开始时,在从设置 SBIxSR<SIOF> 为 "1" 到 SCK 线的下降缘期间,输出与以前传输的数据的最后位中相同的值.

INTSBIx 中断服务程序中,传输可以通过清除 <SIOS> 为 "0" 或设置 <SIOINH> 为 "1" 被终止. 如果 <SIOS> 清除,传输结束前输出剩余数据. 程序检查 SBIxSR<SIOF> 决定传输是否结束. 传输结束时 <SIOF> 清除为 "0". 如果 <SIOINH> 设置为 "1",立即中断传输且 <SIOF> 清除为 "0".

在外部时钟模式下时,下一个数据移位前 <SIOS> 必须清除为 "0". 如果 <SIOS> 不清除为 "0" 下一个数据移位前, SBI 输出虚拟数据且停止.

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

INTSBIx 中断

SBIxDBR	← X	X	X	X	X	X	X	X	写入传输数据.
---------	-----	---	---	---	---	---	---	---	---------

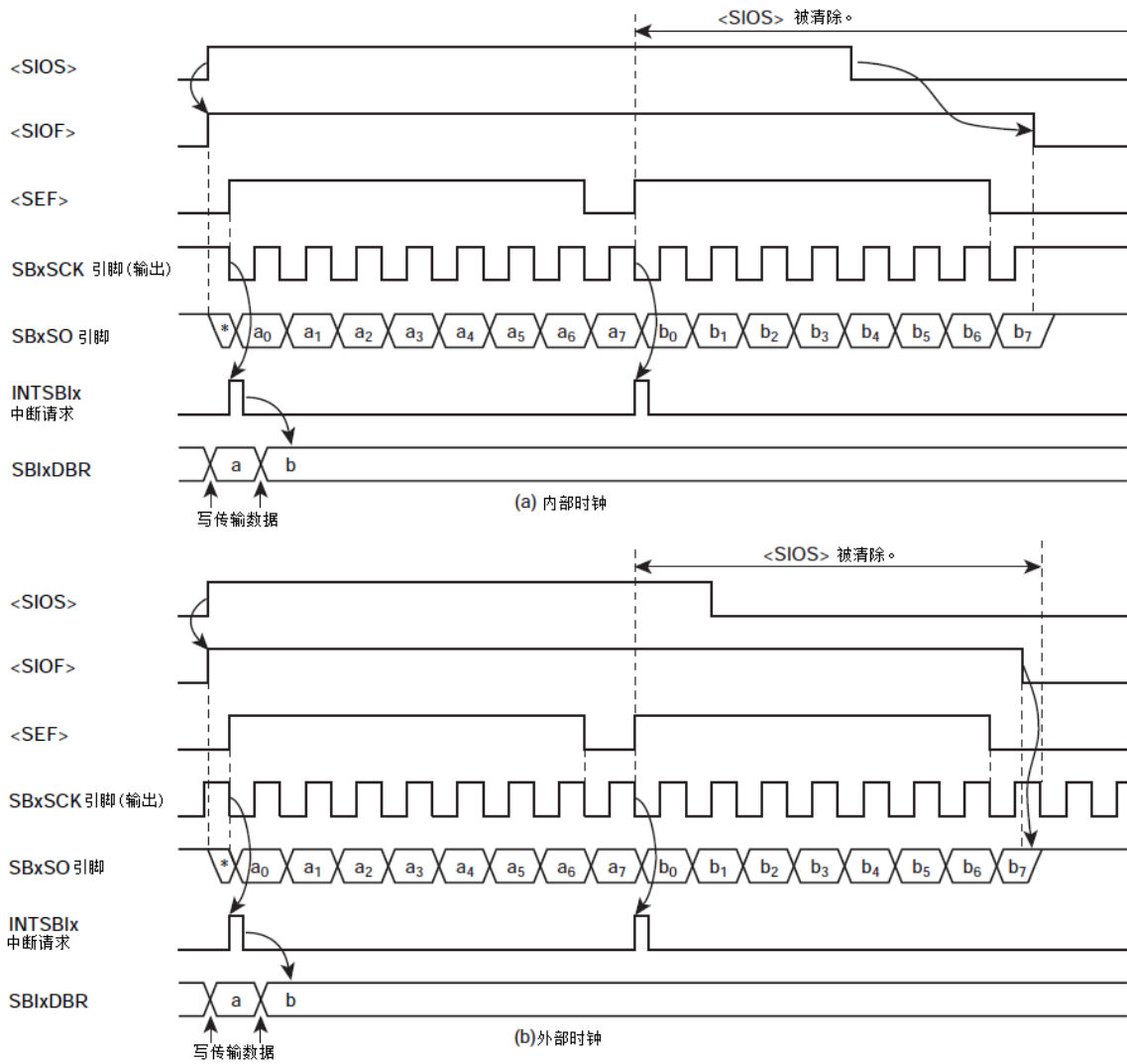


图 14-16 传输 模式

例: 编程(外部 时钟)通过<SIO>中止传输例

```

    7 6 5 4 3 2 1 0
    if SBxSR<SIOF> ≠ 0          识别完传输成.
    然后
    if SCK ≠ 1                  通过监视端口识别 "1" 设置到 SBxSCK 引脚.
    然后
    SBxCR1 ← 0 0 0 0 0 0 1 1 1  通过设置 <SIOS> = 0 完成传输.
    
```

(2) 8-位 接收 模式

设置接收模式的控制寄存器. 然后写入 "1" 到 SBIxCR1<SIOS> 使能接收. 数据从 SBxSI 引脚被带到移位寄存器, 先用最低有效位 (LSB), 同步串行时钟. 一旦移位寄存器用 8-位数据加载, 其传输接收数据到 SBIxDBR 且 INTSBIx (缓冲-全) 中断请求生成来请求读接收的数据. 中断服务程序从 SBIxDBR 读接收的数据.

内部时钟模式下, 串行时钟会停止且自动在等待状态直到从 SBIxDBR 读接收数据.

外部时钟模式下, 移位工作被执行同步外部 时钟. 最大数据传输率改变 取决于生成中断请求和读接收的数据之间最大延迟

通过 INTSBIx 中断服务程序中清除<SIOS> 为 "0" 或设置 <SIOINH> 为 "1", 接收可以被终止. 如果 <SIOS>清除, 接收继续直到接收的数据所有位写入 SBIxDBR. 程序检查 SBIxSR<SIOF> 决定是否接收结束. 接收结束时, <SIOF> 清除为"0". 确认接收完成后, 读最后接收的数据. 如果 <SIOINH>设置为"1", 立即中止接收且 <SIOF> 清除为"0". (接收的数据变为无效, 且没有必要读出它.)

注: 传输模式被改变后, SBIxDBR 内容不会被维持. 处理的接收必须通过清除 <SIOS>为"0" 完成且传输模式被改变前, 必须读最后接收的数据.

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

INTSBIx 中断

Reg.	←	SBIxDBR	读接收数据.
------	---	---------	--------

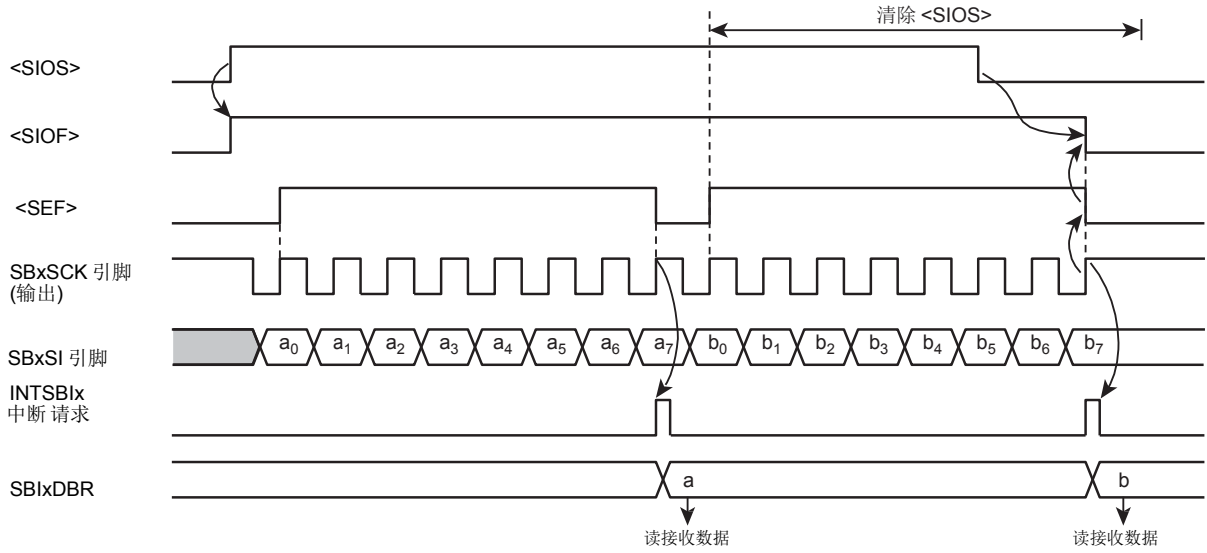


图 14-17 接收模式(例: 内部 时钟)

(3) 8-位 传输/接收 模式

设置传输/接收模式的控制寄存器.然后写入传输数据到 SBIxDBR 且设置 SBIxCR1<SIOS> 为"1" 使能传输和接收. 在串行时钟的下降通过 SBxSO 引脚输出传输数据, 且在串行时钟的上升通过 SBxSI 引脚接收的数据被导入, 先用最低有效位 (LSB). 一旦移位寄存器用 8-位数据加载, 其传输接收数据到 SBIxDBR 且 INTSBIx 中断请求生成. 中断服务程序从数据缓冲寄存器读接收的数据且写入下一个传输数据. 因为传输和接收工作之间 SBIxDBR 被分享, 下一个传输数据写入前必须读接收的数据.

内部时钟工作下, 串行时钟会自动在等待状态直到读接收的数据且写入下一个传输数据.

外部时钟模式下, 移位工作被执行同步外部串行时钟. 因此, 下一个移位工作开始前, 必须读接收的数据且必须写入下一个传输数据. 外部时钟最大数据传输率工作变化取决于中断请求生成时和传输数据写入时之间的最大延迟.

传输开始时, 在从设置 <SIOF> 为 "1" 到 SBxSCK 引脚的下降缘期间, 输出相同与以前传输的数据的最后位的值.

通过清除 INTSBIx 中断服务程序中的<SIOS> 为 "0" 或设置 SBIxCR1<SIOINH> 为"1", 传输和接收可以被终止. 如果 <SIOS> 清除, 传输和接收继续直到接收的数据被全部传输到 SBIxDBR. 程序检查 SBIxSR<SIOF> 决定传输和接收是否结束. 传输和接收结束时<SIOF> 清除为 "0". 如果 <SIOINH> 设置为 "1", 传输和接收被立即中止且 <SIOF> 清除为 "0".

注: 传输模式被改变后 SBIxDBR 的内容不会被维持. 处理中的传输和接收必须通过清除 <SIOS>为 "0" 完成且传输模式被改变前, 必须读最后接收的数据 .

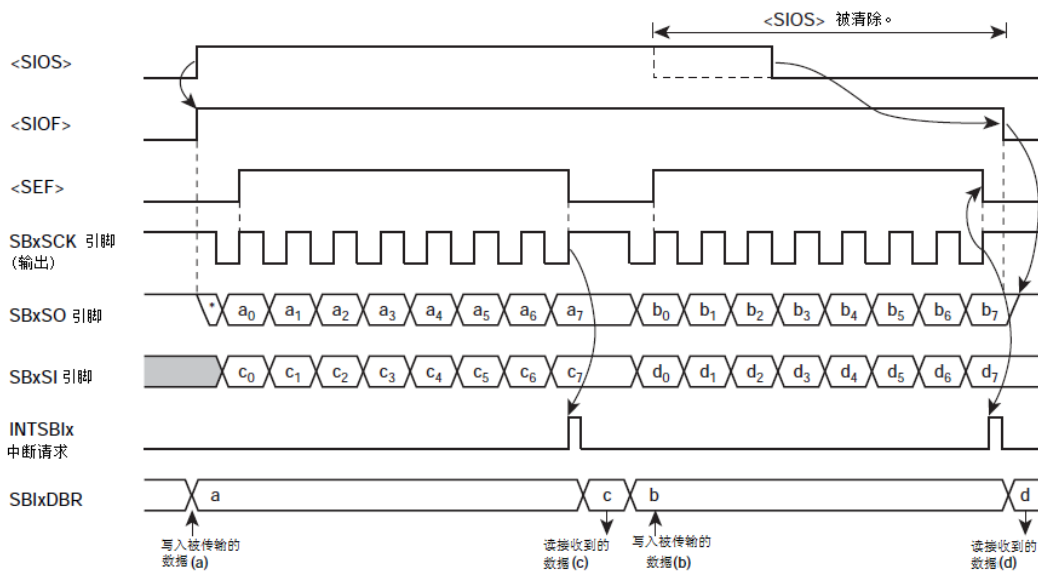


图 14-18 传输/接收模式 (例: 内部时钟)

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

INTSBIx 中断

Reg.	← SBIxDBR	读接收数据.
SBIxDBR	← X X X X X X X X	写入传输数据.

(4) 传输结束时最后位数据保留时间

在条件 $SBIxCR1<SIOS>= "0"$ 下, 传输的数据最后位维持 SBxSCK 引脚上升缘数据如下所示. 传输模式和传输/接收模式相同.

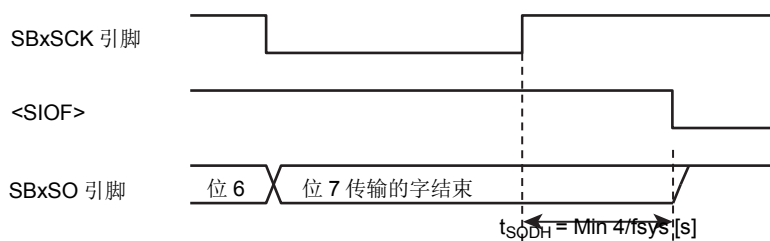


图 14-19 传输结束时最后位数据保留时间

15. CAN 控制器 (CAN)

此产品包括一 通道 CAN 控制器.

15.1 总览

- 符合 CAN 版本 2.0 B (激活)
- 支持标准和扩展格式
- 支持各格式数据帧和遥控帧
- 32 邮箱 (31 接收和传输, 仅 1 接收)
- CAN 总线波特率达 1 Mbps (带至少 48 MHz 系统时钟)
- 位时序参数 等同于 Intel 82527™
- 波特率预分频内置
- 被传输的信息的命令可以从 以下两类型内部仲裁选择 :
 - 带较低号邮箱会先被传输
 - 带较高优先级识别器邮箱会先被传输
- 接收和传输信息的时间标志功能
- 工作 模式

正常工作 模式	
配置 模式	
睡眠 模式	CAN walk-up 带总线激活状态检测 (CANMCR<WUBA> = "1"时) 或主机控制寄存器 MCR 的写的进入
挂起 模式	CAN 总线上非激活状态
测试 循环 后模式	自我承认
测试 错误 模式	可写错误计数器

- 两系统的信息接收屏蔽功能
 - 可编程全球接收屏蔽 (共通于邮箱 0 到 31)
 - 可编程地方接收屏蔽 (仅邮箱 31 用)
- ID 扩展位的接收屏蔽位
- 中断信号

INTCANRX	: CAN 接收完成中断
INTCANTX	: CAN 传输完成中断
INTCANGB	: CAN 全球中断 来自八原因包括警告电平, 错误被动和总线- 关闭中断) 的中断

15.2 方块图

图 15-1 所示 CAN 控制器方块图.

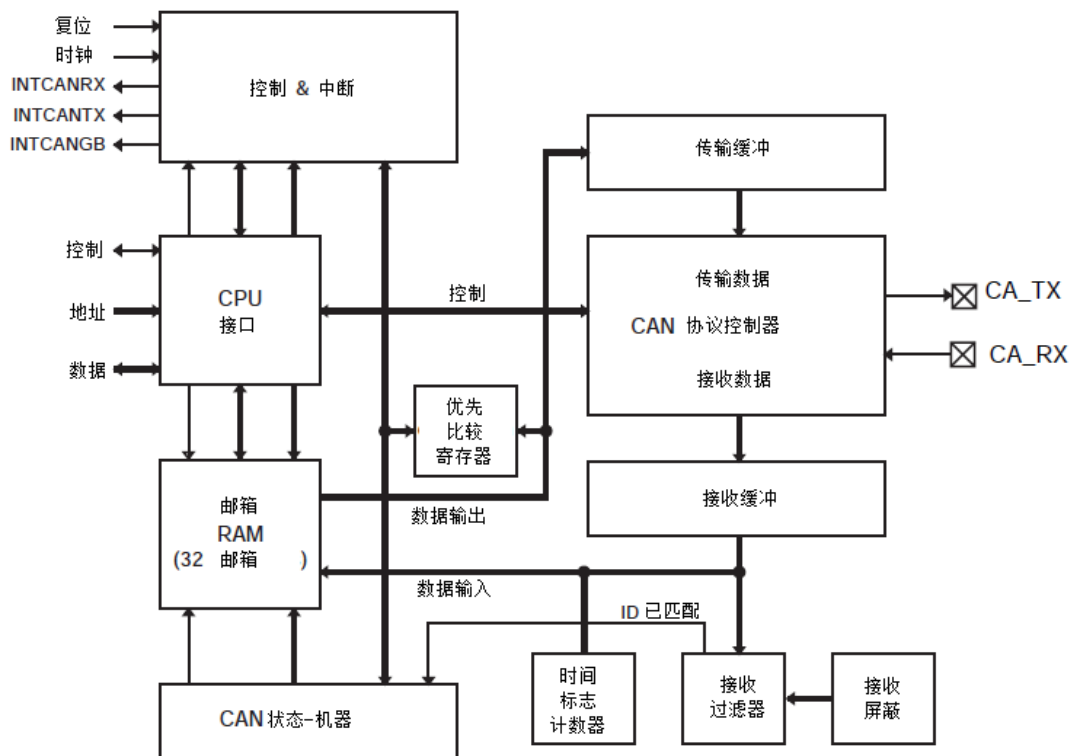


图 15-1 CAN 控制器方块图

15.3 CAN 接口

CAN 总线接口为输入引脚 CA_RX 和输出引脚 CA_TX. 连接这些引脚通过 CAN 总线接收器 (ISO / DIS 11898 符合).

高速和低速接收器不同. 此 IP 必须小心 (比如, 3.3 V 到 5 V) 这些芯片级引脚的电气特征满足接收器的需求.

15.4 寄存器

此 MCU 中, 复位后 CAN 时钟停止(CGCKSTP<CANSTP> = "0"). CAN 被用时, 设置 CAN 工作 (CGCKSTP<CANSTP> = "1").

关于时钟工作详细, 参考有关 "时钟/模式 控制" 章节.

15.4.1 寄存器 表

寄存器 名 (x=0 到 31)		地址(基+)
信息 ID 域寄存器	CANMBxID	0x0000
时间标志值 / 信息控制域寄存器	CANMBxTSMCF	0x0008
数据域寄存器	CANMBxDL	0x0010
数据域寄存器	CANMBxDH	0x0018

寄存器 名		地址(基+)
邮箱 配置 寄存器	CANMC	0x0000
邮箱 方向 寄存器	CANMD	0x0008
传输 请求 设置 寄存器	CANTRS	0x0010
传输 请求 复位 寄存器	CANTRR	0x0018
传输 承认 寄存器	CANTA	0x0020
中止 承认 寄存器	CANAA	0x0028
接收 信息 挂起寄存器	CANRMP	0x0030
接收 信息 丢失 寄存器	CANRML	0x0038
地方 接受屏蔽 寄存器	CANLAM	0x0040
全球 接受屏蔽 寄存器	CANGAM	0x0048
主机控制 寄存器	CANMCR	0x0050
全球 状态 寄存器	CANGSR	0x0058
位 配置 寄存器 1	CANBCR1	0x0060
位 配置 寄存器 2	CANBCR2	0x0068
全球 中断 标志 寄存器	CANGIF	0x0070
全球 中断 屏蔽 寄存器	CANGIM	0x0078
邮箱 传输 中断 标志 寄存器	CANMBTIF	0x0080
邮箱 接收 中断 标志 寄存器	CANMBRIF	0x0088
邮箱 中断 屏蔽 寄存器	CANMBIM	0x0090
改变 数据 请求 寄存器	CANCDR	0x0098
遥控 帧 挂起寄存器	CANRFP	0x00A0
CAN 错误 计数器 寄存器	CANCEC	0x00A8
定时器 标志 计数器 预分频 寄存器	CANTSP	0x00B0
定时器 标志 计数器 寄存器	CANTSC	0x00B8

15.4.2 CANMBxID (信息 ID 域 寄存器)

	31	30	29	28	27	26	25	24
位 符	IDE	GAME/LAME	RFH	ID				
复位后								
	23	22	21	20	19	18	17	16
位 符	ID							
复位后								
	15	14	13	12	11	10	9	8
位 符	ID							
复位后								
	7	6	5	4	3	2	1	0
位 符	ID							
复位后								

位	位 符	类型	功能
31	IDE	R/W	<p>ID 扩展 位</p> <p>0: 标准格式 (11-位 ID) 从 <ID28> 到 <ID18> 被用</p> <p>1: 扩展格式 (29-位 ID) 从 <ID28> 到 <ID0> 被用</p> <p>通过选择接收或传输扩展格式 (<IDE>="1") 或标准格式 (<IDE>="0")设置邮箱.</p>
30	GAME / LAME	R/W	<p>全球 (GAME) / 地方 (LAME) 接受屏蔽使能位</p> <p>0: 接收屏蔽不被用作接收过滤器.</p> <p>1: 接收屏蔽被用作接收过滤器.</p> <p><GAME> 为邮箱 0 到 30 中分享的全球接受屏蔽 GAM 使能位, 且 <LAME>为仅被用作邮箱 31 的地方接受屏蔽 LAM 使能位.</p> <p><GAME>=0 或 <LAME>=0 时, 仅当接收信息 ID 相同于邮箱 ID 时, 接收信息储存在邮箱.</p> <p>对于传输邮箱, 接受屏蔽功能不被应用. 此时, 始终设置 <GAME>到"0".</p>
29	RFH	R/W	<p>遥控 帧处理位 (仅给传输邮箱)</p> <p>0:传输 邮箱不响应遥控帧. 软件必须处理遥控帧.</p> <p>1:传输 邮箱响应到遥控帧. (<TRS> 位设置.)</p> <p><RFH> 决定是否邮箱配置为传输邮箱会自动响应遥控帧接收.</p> <p>接收遥控帧 ID 匹配 <RFH>="1" 且</p> <p><GAME>="1"传输邮箱 ID 时, 此邮箱 ID 用遥控 ID 覆盖, 且使用覆盖 ID 邮箱自动响应遥控帧.</p> <p>接收邮箱时, 处理为数据帧.(<RMP> 位和 <RFP> 位设置.)</p>
28-0	ID[28:0]	R/W	<p>信息 ID</p> <p>标准格式 (11-位 ID) : 从 <ID28>到<ID18> 被用.</p> <p>扩展格式 (29-位 ID) :从 <ID28>到<ID0> 被用.</p> <p>关于信息 ID 优先级, 连续从 ID 的最高位 (<ID28> 位) 开始有最多"0"的信息 ID 有更高优先级.</p>

在初始设置时寄存邮箱 ID. 邮箱使能后要改变信息 ID 域或邮箱, 写入新 ID 前, 清除对应邮箱到 "0" 的 CANMC 寄存器中<MCx> 位然后禁止 CAN 控制器的邮箱.

15.4.3 CANMBxTSMCF (时间标志值 / 信息控制域寄存器)

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

位	位符	类型	功能																														
31-16	TSV[15:0]	R/W	时间标志计数器值 信息完全成功接收或传输时读到的 16-位 时间标志计数器值被储存.信息接收或传输失败时, 没有值被设置. 关于整个时间标志计数器功能的详细, 参考 "15.5.6 时间标志功能".																														
15-5	-	R	读未定义. 写作 "0".																														
4	RTR	R/W	遥控帧传输请求位. 0:数据帧 1:遥控帧																														
3-0	DLC[3:0]	R/W	数据长代码 设置信息的数据长 (字节数) <table border="1" data-bbox="507 1240 999 1621"> <thead> <tr> <th><DLC[3:0]></th> <th>字节数</th> <th>对应数据</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>0 字节</td> <td>无</td> </tr> <tr> <td>0001</td> <td>1 字节</td> <td>D0</td> </tr> <tr> <td>0010</td> <td>2 字节</td> <td>D0,D1</td> </tr> <tr> <td>0011</td> <td>3 字节</td> <td>D0,D1,D2</td> </tr> <tr> <td>0100</td> <td>4 字节</td> <td>D0,D1,D2,D3</td> </tr> <tr> <td>0101</td> <td>5 字节</td> <td>D0,D1,D2,D3,D4</td> </tr> <tr> <td>0110</td> <td>6 字节</td> <td>D0,D1,D2,D3,D4,D5</td> </tr> <tr> <td>0111</td> <td>7 字节</td> <td>D0,D1,D2,D3,D4,D5,D6</td> </tr> <tr> <td>1000</td> <td>8 字节</td> <td>D0,D1,D2,D3,D4,D5,D6,D7</td> </tr> </tbody> </table> <DLC3:0>="1001" 或更多被设置时, 数据长被处理为 8 字节.	<DLC[3:0]>	字节数	对应数据	0000	0 字节	无	0001	1 字节	D0	0010	2 字节	D0,D1	0011	3 字节	D0,D1,D2	0100	4 字节	D0,D1,D2,D3	0101	5 字节	D0,D1,D2,D3,D4	0110	6 字节	D0,D1,D2,D3,D4,D5	0111	7 字节	D0,D1,D2,D3,D4,D5,D6	1000	8 字节	D0,D1,D2,D3,D4,D5,D6,D7
<DLC[3:0]>	字节数	对应数据																															
0000	0 字节	无																															
0001	1 字节	D0																															
0010	2 字节	D0,D1																															
0011	3 字节	D0,D1,D2																															
0100	4 字节	D0,D1,D2,D3																															
0101	5 字节	D0,D1,D2,D3,D4																															
0110	6 字节	D0,D1,D2,D3,D4,D5																															
0111	7 字节	D0,D1,D2,D3,D4,D5,D6																															
1000	8 字节	D0,D1,D2,D3,D4,D5,D6,D7																															

时间标志值不需要初始设置.

接收邮箱时, 信息控制域不需要初始程序. 接收信息储存在邮箱时, 同时<RTR> 和<DLC[3:0]> 也储存在信息控制域. 传输邮箱需要初始设置.

要改变传输邮箱 (设置到<RFH>="1") 信息控制域, 使能邮箱后, 清除 CANMC<MCx> 位为 "0" 然后写入新 <RTR> 和 <DLC[3:0]>前禁止 CAN 控制器邮箱. 设置为 <RFH>="0" 的传输邮箱信息控制域无论 CANMC<MCx> 位设置可以被改变, 但用户写入新<RTR>和 <DLC[3:0]>前需要检查 CANTRS<TRSx> 位为 "0".

15.4.4 CANMBxDL/CANMBxDH (数据域寄存器)

关于传输, 数据根据邮箱的<DLC[3:0]> 中数据字节计数设置被传输.

关于接收, 接收信息中数据长代码被复制到邮箱的<DLC[3:0]>, 且数据字节计数仅设置在<DLC[3:0]> 有效.

邮箱可读且可写, 但不要写入接收邮箱数据域. 如果数据域被写, 不匹配可能发生在接收数据.

要更新设置到<RFH>="1"的传输邮箱数据域, 写入新数据前在 CANCDR<CDRx> 中设置 "1" 且挂起暂时传输请求. 要更新设置到 <RFH>="0" 传输邮箱数据域, 写入新数据前检查 CANTRS<TRS> 位为 "0".

CANMBxDL

	31	30	29	28	27	26	25	24
位符	D3							
复位后								
	23	22	21	20	19	18	17	16
位符	D2							
复位后								
	15	14	13	12	11	10	9	8
位符	D1							
复位后								
	7	6	5	4	3	2	1	0
位符	D0							
复位后								

位	位符	类型	功能
31-24	D3[7:0]	R/W	传输和接收的数据被储存.
23-16	D2[7:0]	R/W	传输和接收的数据被储存.
15-8	D1[7:0]	R/W	传输和接收的数据被储存.
7-0	D0[7:0]	R/W	传输和接收的数据被储存.

CANMBxDH

	31	30	29	28	27	26	25	24
位符	D7							
复位后								
	23	22	21	20	19	18	17	16
位符	D6							
复位后								
	15	14	13	12	11	10	9	8
位符	D5							
复位后								
	7	6	5	4	3	2	1	0
位符	D4							
复位后								

位	位符	类型	功能
31-24	D7[7:0]	R/W	传输和接收的数据被储存.
23-16	D6[7:0]	R/W	传输和接收的数据被储存.
15-8	D5[7:0]	R/W	传输和接收的数据被储存.
7-0	D4[7:0]	R/W	传输和接收的数据被储存.

15.4.5 CANMC (邮箱 配置 寄存器)

	31	30	29	28	27	26	25	24
位 符	MC31	MC30	MC29	MC28	MC27	MC26	MC25	MC24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	MC23	MC22	MC21	MC20	MC19	MC18	MC17	MC16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能															
31-0	MC31 ~ MC0	R/W	邮箱的进入配置(各位对应邮箱 31 ~ 0) 0: 对应的邮箱 MBx 禁止给 CAN 控制器. 1: 对应的邮箱 MBx 使能给 CAN 控制器. 从 CPU 写入进入															
			<table border="1"> <thead> <tr> <th></th> <th>ID 域</th> <th>传输 邮箱 用 <RFH>="1"</th> <th>数据 域</th> <th>控制 域</th> </tr> </thead> <tbody> <tr> <td><MCx>=0</td> <td>使能</td> <td>使能</td> <td>使能</td> <td>使能</td> </tr> <tr> <td><MCx>=1</td> <td>禁止</td> <td>禁止</td> <td>使能</td> <td>使能</td> </tr> </tbody> </table>		ID 域	传输 邮箱 用 <RFH>="1"	数据 域	控制 域	<MCx>=0	使能	使能	使能	使能	<MCx>=1	禁止	禁止	使能	使能
	ID 域	传输 邮箱 用 <RFH>="1"	数据 域	控制 域														
<MCx>=0	使能	使能	使能	使能														
<MCx>=1	禁止	禁止	使能	使能														

注:工作中 CANMC 的重编程期间需要以下注意 .

接收: 对于接收邮箱需要确保此邮箱接收处理中邮箱不被禁止. 如果接收处理中期间邮箱禁止或重新配置, 当前帧 可以被接收.

传输: CAN 控制器传输数据 (CANTRS<TRSx>="1")时,传输完成后(CANTRS<TRSx>="0")清除 <MCx> 为 "0" .

15.4.6 CANMD (邮箱 方向 寄存器)

	31	30	29	28	27	26	25	24
位 符	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24
复位后	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	MD31	R	邮箱 方向：邮箱 31 邮箱 31 为仅接收邮箱. 此始终设置为 "1" 且不能被改变.
30-0	MD30 ~ MD0	R/W	邮箱 方向：邮箱 30 到 0 (各位对应邮箱 30 到 0.) 0:设置为传输 邮箱. 1:设置为接收 邮箱. 各邮箱可以设置为传输或接收邮箱.

在初始设置设置 CANMD 寄存器. 工作处理时邮箱方向不可以被改变. 要改变 CANMD 寄存器设置, 改变前设置对应的 CANMC<MCx> 位为 "0".

15.4.7 CANTRS (传输 请求 设置 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	TRS30	TRS29	TRS28	TRS27	TRS26	TRS25	TRS24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	TRS23	TRS22	TRS21	TRS20	TRS19	TRS18	TRS17	TRS16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	TRS15	TRS14	TRS13	TRS12	TRS11	TRS10	TRS9	TRS8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	TRS7	TRS6	TRS5	TRS4	TRS3	TRS2	TRS1	TRS0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	-	R	读：读作"0". 写：写作"0".
30-0	TRS30 ~ TRS0	R/W	传输 请求 设置 (各位对应邮箱 30~0.) 设置 <TRSx> 请求对应的邮箱 x 的信息传输. 传输请求为多邮箱时, 根据对应 MCR<MTOS> 位的优先级传输信息. 从 CPU 写入 "1"到配置为传输邮箱的邮箱 x 可以设置位. 从 CPU 写入"0" 无效.

注:邮箱 31 仅为接收邮箱.

传输请求设置寄存器可以通过写入 "1" 从 CPU 到配置为仅传输的邮箱 CANTRS<TRSx> 位设置. 不可以设置 配置为接收的邮箱的 CANTRS<TRSx> 位.

信息全部成功传输或传输请求通过设置 CANTRR<TRRx> 位为 "1" 复位时, CANTRS<TRSx> 位清除为 "0" .

传输失败时, 重复传输处理直到它成功或传输请求通过设置 CANTRR<TRRx> 位为 "1"复位.

CANTRS<TRSx> 位为 "1"时, 不要写入邮箱 x.

15.4.8 CANTRR (传输 请求 复位 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	TRR30	TRR29	TRR28	TRR27	TRR26	TRR25	TRR24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	TRR23	TRR22	TRR21	TRR20	TRR19	TRR18	TRR17	TRR16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	TRR15	TRR14	TRR13	TRR12	TRR11	TRR10	TRR9	TRR8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	-	R	读：读作 "0". 写：写作 "0".
30-0	TRR30 ~TRR0	R/W	传输 请求 复位 (各位对应邮箱 30 ~0.) 设置 <TRRx> 消除对应的邮箱 x 的信息传输。 从 CPU 写入 "1" 到配置为传输邮箱的邮箱 x 可以设置 位. 从 CPU 写入 "0" 无效.

注:邮箱 31 为仅接收邮箱.

传输请求复位寄存器可以通过从 CPU 写入 "1" 到仅配置为传输的邮箱的 CANTRR<TRRx> 位设置. 接收 CANTRR<TRRx> 位不可以设置.

通过内部逻辑 CANTRR<TRRx> 位清除为 "0" 信息全部成功传输或传输中止时. 从 CPU 写入 "0" 无效.

CANTRR<TRRx> 位为"1 时,"不要写入邮箱 x.

设置 CANTRR<TRRx> 位消除通过 CANTRS<TRSx> 位设置的邮箱 x 的信息传输,会执行任何以下三个顺序工作:

- a. 信息传输请求还没被传输.
信息传输请求会立即清除.
(CANTRS<TRSx> = 0, CANTRR<TRRx> = 0, CANAA<AAx> = 1)
- b. 信息传输 请求正被传输 仲裁丢失错误发生或在 CAN 总线上错误被检测.
信息传输请求会清除且传输会消除.
(CANTRS<TRSx> = 0, CANTRR<TRRx> = 0, CANAA<AAx> = 1)
- c. 信息传输 请求正被传输且仲裁丢失错误发生且无错误在 CAN 总线上被检测.
信息传输请求不会清除且传输会完成.
(CANTRS<TRSx> = 0, CANTRR<TRRx> = 0, CANTA<TAx> = 1)

15.4.9 CANTA (传输 获取 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	TA30	TA29	TA28	TA27	TA26	TA25	TA24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	TA23	TA22	TA21	TA20	TA19	TA18	TA17	TA16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	TA15	TA14	TA13	TA12	TA11	TA10	TA9	TA8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	-	R	读：读作"0". 写：写作"0".
30-0	TA30~TA0	R/W	传输 获取 (各位对应邮箱 30~0) 邮箱 x 中的信息全部成功传输时, <TAx> 位设置为 "1". <TAx> 位可以通过从 CPU 写入 "1"到 <TAx> 位 或 TRS<TRSx> 位清除.

注:邮箱 31 为仅接收邮箱.

邮箱 x 中信息全部成功传输时, CANTA<TAx> 位设置为"1".通过设置对应的邮箱中断屏蔽寄存器 CANMBIM 中 <MBIMx> 位为 "1"时, 邮箱中断使能, 邮箱传输中断标志寄存器 CANMBTIF 的 <MBTIFx> 位设置为 "1" 且可以传输完成中断 INTCANTX 发生.

从 CPU 写入 "1" 到 <TAx> 位或 CANTRS<TRSx> 位可以清除 <TAx> 位. 从 CPU 写入"0" 到 <TAx> 位或 CANTRS<TRSx> 位无效.

15.4.10 CANAA (中止 获取 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	AA30	AA29	AA28	AA27	AA26	AA25	AA24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	AA23	AA22	AA21	AA20	AA19	AA18	AA17	AA16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	AA15	AA14	AA13	AA12	AA11	AA10	AA9	AA8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	AA7	AA6	AA5	AA4	AA3	AA2	AA1	AA0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	-	R	读：读作 "0". 写：写作 "0".
30-0	AA30 ~ AA0	R/W	中止 获取 (各位对应邮箱 30 ~ 0.) 邮箱 x 中信息没有全部成功传输时, <AAx> 位设置为 "1". <AAx> 位可以从 CPU 通过写入 "1" 到 <AAx> 位或 CANTRS<TRSx> 位清除.

注: 邮箱 31 为仅接收邮箱.

邮箱 x 中信息没有被成功传输时, CANAA<AAx> 位设置为 "1". 全球中断标志寄存器中 CANGIF<TRMABF> 位也设置为 "1", 且传输中止中断使能通过设置全球中断屏蔽寄存器 CANGIM<TRAMABM> 位为 "1" 时, 可以全球中断 INTCANGB 发生.

从 CPU 写入 "1" 到 <AAx> 位或 CANTRS<TRSx> 位可以清除 <AAx> 位. 从 CPU 写入 "0" 到 <AAx> 位或 CANTRS<TRSx> 位无效.

15.4.11 CANRMP (接收 信息 挂起 寄存器)

	31	30	29	28	27	26	25	24
位 符	RMP31	RMP30	RMP29	RMP28	RMP27	RMP26	RMP25	RMP24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	RMP23	RMP22	RMP21	RMP20	RMP19	RMP18	RMP17	RMP16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	RMP15	RMP14	RMP13	RMP12	RMP11	RMP10	RMP9	RMP8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	RMP7	RMP6	RMP5	RMP4	RMP3	RMP2	RMP1	RMP0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	RMP31 ~ RMP0	R/W	接收信息中止(各位对应邮箱 31 ~ 0.) 接收的信息和接收信息的内容写入邮箱 x 后, <RMPx> 位设置为"1". 读接收的数据后, 写入"1" 到 <RMPx> 位可以清除 <RMPx> 位.

注:此寄存器不可以使用读-修改-写入指令.

邮箱 x 中信息成功接收时, CANRMP<RMPx> 位设置为"1". 邮箱中断使能通过设置对应的邮箱中断屏蔽寄存器 CANMBIM 中 <MBIMx> 位为 "1"时, 邮箱接收中断标志寄存器 CANMBRIF 的<MBRIFx> 位设置为 "1" 且可以接收完成中断 INTCANRX 发生.

要清除<RMPx> 位, 从 CPU 写入 "1"到 <RMPx> 位. 从 CPU 写入"0" 到 <RMPx> 位无效.

15.4.12 CANRML (接收 信息 丢失 寄存器)

	31	30	29	28	27	26	25	24
位 符	RML31	RML30	RML29	RML28	RML27	RML26	RML25	RML24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	RML23	RML22	RML21	RML20	RML19	RML18	RML17	RML16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	RML15	RML14	RML13	RML12	RML11	RML10	RML9	RML8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	RML7	RML6	RML5	RML4	RML3	RML2	RML1	RML0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	RML31~ RML0	R/W	接收 信息 丢失 (各位对应 31 ~ 0.) <RMPx> 位被设置为"1"的邮箱 x 接收下一个信息时, 接收的信息的内容覆盖邮箱 x,且 <RMLx> 位设置为"1". 写入"1" 到<RMPx> 位可以清除 <RMLx> 位.

CANRML<RMLx> 位通过内部逻辑从 CPU 设置且可以用写入 "1" 到 CANRMP<RMPx> 位清除.
<RMPx> 位同时也清除.从 CPU 写入"1" 或 "0"到<RMLx> 位无效.

用 CANRMP<RMPx> 位设置为 "1", 如果邮箱 x 接收下一个信息, 接收信息丢失寄存器 CANRML 中对应的<RMLx> 位设置为"1". 此时, 邮箱 x 用新接收的信息覆盖.

全球中断标志寄存器 CANGIF 中 <TRMABF> 位也设置为"1", 且传输中止中断使能通过设置全球中断屏蔽寄存器 CANGIM 中 <TRMABM> 位为"1"时, CAN 全球中断 INTCANGB 发生.

接收信息丢失中断使能 通过设置全球中断屏蔽寄存器 CANGIM 中 <RMLIM> 位为"1"时, CAN 中断 INTCANGB 发生.

表 15-1 表示 CANRMP 和 CANRML 寄存器信息接收前后的变化.

表 15-1 RMP 和 RML 寄存器信息被接收前 / 后的变化

ID	接收前		接收后		工作
	<RMPx>	<RMLx>	<RMPx>	<RMLx>	
无 匹配	忽略	忽略	忽略	忽略	接收的信息不储存在任何邮箱.
匹配	0	0	1	0	接收的信息用匹配 ID 储存在邮箱 x.
	1	0	1	1	接收的信息用匹配 ID 被覆盖在邮箱 x. 此表示之前的信息丢失.
	1	1	1	1	

15.4.13 CANLAM (当地 接受 屏蔽 寄存器)

当地接受屏蔽寄存器 CANLAM 仅会被用来过滤邮箱 31 的接收信息 ID。此特征允许当地屏蔽邮箱 31 的接收信息的任何 ID 位。

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

位	位 符	类型	功能
31	LAMI	R/W	邮箱 31 的 <IDE> 位屏蔽 0:不屏蔽 1:屏蔽 <LAMI>="0"时, 标准或扩展格式中的信息被接收, 根据邮箱 31 的<IDE> 位。 <LAMI>="1"时, 标准或扩展格式中的信息被接收, 无论邮箱 31 的 <IDE> 位。
30-29	-	R	读: 读作"0". 写: 写作"0".
28-0	LAM[28:0]	R/W	接收信息 ID 屏蔽 0:不屏蔽 对应的接收信息 ID 位相同于邮箱 ID 时, 接收信息被接收。 1:屏蔽 接收信息被接收无论接收信息对应的位的值。

扩展格式中, < ID[28:0] > 和 < LAM[28:0] > 被用来过滤。

标准格式中, < ID[28:18] > 和 < LAM[28:18] > 被用来过滤。

标准格式中信息被接收时, 部分扩展 ID (<ID[17:0]>) 会变为未定义值。因此, 标准和扩展格式不可以推荐在相同邮箱中交替接收。

初始 (在配置模式) 时请设置 CANLAM 且工作时不要改变设置。当接收信息时设置被改变时, 设置改变中的 CANLAM 值被用来过滤接收信息 ID。

15.4.14 CANGAM (全球 接受 屏蔽 寄存器)

全球 接受屏蔽 寄存器 CANGAM 会被用来过滤邮箱 0 到 30 的接收信息 ID. 此特征允许全球屏蔽邮箱 0 到 30 的接收信息的任何 ID 位.

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

位	位 符	类型	功能
31	GAMI	R/W	邮箱 0 ~30<IDE> 位的屏蔽 0:不屏蔽 1:屏蔽 如果 <GAMI> = "0", 标准或 扩展格式的信息被接收, 根据邮箱 0~30 的<IDE> 位 . <GAMI> = "1"时, 标准和扩展格式的信息被接收, 无论邮箱 0~30 的<IDE> 位 .
30-29	-	R	读 : 读作"0". 写 : 写作"0".
28-0	GAM[28:0]	R/W	接收信息 ID 的屏蔽 0:不屏蔽 接收信息 ID 对应的位相同于邮箱 ID 时, 接收信息被接收. 1:屏蔽 接收信息被接收无论接收信息对应的位的值.

扩展格式中, < ID[28:0] > 和 < GAM[28:0] > 被用来过滤.

标准格式中, < ID[28:18] > 和 < GAM[28:18] > 被用来过滤.

标准格式中信息被接收时, 部分扩展 ID (<ID[17:0]>) 会变为未定义值.因此,标准和扩展格式不可以被推荐在相同邮箱中交替接收.

初始(在 配置模式) 期间请设置 CANGAM 且工作期间不要改变设置.当接收信息时设置被改变时,设置改变处理中的 CANGAM 值被用来过滤接收信息 ID.

15.4.15 CANMCR (主机 控制 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	SUR	-	TSTLB	TSTERR
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CCR	SMR	-	WUBA	MTOS	-	TSCC	SRES
复位后	1	0	0	0	0	0	0	0

位	位 符	类型	功能
31-12	-	R	读：读作 "0". 写：写作 "0".
11	SUR	R/W	挂起 模式 请求 0: 消除挂起模式 (正常工作) 1: 请求挂起模式
10	-	R	读：读作 "0". 写：写作 "0".
9	TSTLB	R/W	测试 循环 后 0:消除测试 循环后 模式 (正常工作) 1:请求测试 循环后模式 (此模式支持独立工作.)
8	TSTERR	R/W	测试 错误 0:消除测试错误 模式 (正常工作) 1:请求测试错误 模式 (此模式,可以写入 CAN 错误计数器寄存器 (CANCEC)).
7	CCR	R/W	改变 配置 请求 0:消除配置模式 (正常工作) 1:请求配置模式 (此模式,可以写入位配置寄存器, CANBCR1 和 CANBCR2.)
6	SMR	R/W	睡眠 模式 请求 0:消除睡眠模式 (正常工作) 1:请求睡眠模式 (此模式, CAN 控制器时钟停止且错误计数且传输请求复位.)
5	-	R	读：读作 "0". 写：写作 "0".
4	WUBA	R/W	总线上唤醒激活 0: 仅通过写入进入到 CAMCR 寄存器唤醒. 1:通过检测总线激活状态或写入进入到 CAMCR 唤醒.
3	MTOS	R/W	邮箱传输顺序选择 0: 以邮箱数的升序传输信息. 1: 以信息 ID 优先级的降序传输邮箱中信息.
2	-	R	读：读作"0". 写：写作"0".
1	TSCC	R/W	时间标志计数器清除 0: 禁止 1:清除时间标志计数器为 "0". (注 1) 此位仅给写入且始终读作"0".

位	位符	类型	功能
0	SRES (注 2)	R/W	软件 复位 0:禁止 1:通过软件复位 CAN 控制器. 此位仅给写入且始终读作"0".

注 1: 时间标志计数器 也通过写入 CANTSP 寄存器和写入 "0" 到 CANTSC 寄存器清除.

注 2: 软件复位后, 所有 CAN 中寄存器以下时间后必须进入.

(1) 通过 CAN 总线通信不执行时, 请等待大于 16 CPU 时钟.

(2) 通过 CAN 总线通信执行时, 请等待大于 88 CPU 时钟.

注 3: 要消除 CAN 工作睡眠模式, 设置 CANMCR<SMR>为"0"前检查 CANGSR<SMA> 位为 "1".

15.4.16 CANGSR (全球 状态 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	MIS
复位后	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
位 符	MIS				RM	TM	-	SUA
复位后	1	1	1	1	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CCE	SMA	-	-	TSO	BO	EP	EW
复位后	1	0	0	0	0	0	0	0

位	位 符	类型	功能
31-17	-	R	读：读作"0". 写：写作"0".
16-12	MIS[4:0]	R	槽中信息 指传输缓冲中信息的邮箱数。 00000：邮箱 0 信息 01011：邮箱 11 信息 10110：邮箱 22 信息 00001：邮箱 1 信息 01100：邮箱 12 信息 10111：邮箱 23 信息 00010：邮箱 2 信息 01101：邮箱 13 信息 11000：邮箱 24 信息 00011：邮箱 3 信息 01110：邮箱 14 信息 11001：邮箱 25 信息 00100：邮箱 4 信息 01111：邮箱 15 信息 11010：邮箱 26 信息 00101：邮箱 5 信息 10000：邮箱 16 信息 11011：邮箱 27 信息 00110：邮箱 6 信息 10001：邮箱 17 信息 11100：邮箱 28 信息 00111：邮箱 7 信息 10010：邮箱 18 信息 11101：邮箱 29 信息 01000：邮箱 8 信息 10011：邮箱 19 信息 11110：邮箱 30 信息 01001：邮箱 9 信息 10100：邮箱 20 信息 11111：传输缓冲中无信息 01010：邮箱 10 信息 10101：邮箱 21 信息
11	RM	R	接收 模式 0: CAN 控制器不接收信息。 1: CAN 控制器接收信息。
10	TM	R	传输 模式 0: CAN 控制器不传输信息。 1: CAN 控制器传输信息。
9	-	R	读：读作 "0". 写：写作 "0".
8	SUA	R	挂起 模式 承认 0: CAN 控制器不在挂起模式。 1: CAN 控制器在挂起模式。
7	CCE	R	改变 配置 使能 0: CAN 控制器不在配置模式。 1: CAN 控制器在配置模式。 此模式下,可以写入位配置寄存器, CANBCR1 和 CANBCR2.
6	SMA	R	睡眠 模式 承认 0: CAN 控制器不在睡眠模式。 1: CAN 控制器在睡眠模式。 此模式下, CAN 控制器时钟停止且错误计数器和传输请求复位。
5-4	-	R	读：读作 "0". 写：写作 "0".
3	TSO	R	时间标志溢出 0: 时间标志计数器不溢出。 1: 时间标志计数器此位最后清除为 "0"后溢出至少一次. 要清除此位, 清除 CANGIF 寄存器中<TSOIF> 位为 "0".

位	位符	类型	功能
2	BO	R	总线关闭状态 0:总线 开启 状态 (正常工作) 中 1:总线关闭状态中 CAN 总线 错误经常发生异常且传输错误计数器 <TEC> 到达其 256 的极限时, CAN 控制器进入总线关闭状态. 没有 信息可以传输和接收, 错误计数器未定义. 总线关闭恢复顺序后, CAN 控制器自动进入总线开启状态.
1	EP	R	错误被动状态 0: CAN 控制器不在错误被动模式. 1: CAN 控制器在错误被动模式.
0	EW	R	警告 状态 0: <TEC> 和 <REC> 值 都为 96 或更低. 1:至少一个 <TEC> 和 <REC> 值大于 96 且到达警告电平.

15.4.17 CANBCR1 (位 配置 寄存器 1)

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

位	位 符	类型	功能
31-10	-	R	读：读作 "0". 写：写作 "0".
9-0	BRP[9:0]	R/W	波特率 预分频 设置值：0 ~ 1023

15.4.18 CANBCR2 (位 配置 寄存器 2)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	SJW	
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	SAM	TSEG2			TSEG1			
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-10	-	R	读：读作"0". 写：写作"0".
9-8	SJW[1:0]	R/W	再同步 跳跃宽. 00 : 1 × TQ 01 : 2 × TQ 10 : 3 × TQ 11 : 4 × TQ
7	SAM	R/W	设置采样 计数 0:单 采样 1 三 采样
6-4	TSEG2[2:0]	R/W	采样点后位时间设置 000 : 保留 100 : 5 × TQ 001 : 2 × TQ 101 : 6 × TQ 010 : 3 × TQ 110 : 7 × TQ 011 : 4 × TQ 111 : 8 × TQ
3-0	TSEG1[3:0]	R/W	采样点前 (除 SYNCSEG) 位时间设置. 0000 : 保留 1000 : 9 × TQ 0001 : 2 × TQ 1001 : 10 × TQ 0010 : 3 × TQ 1010 : 11 × TQ 0011 : 4 × TQ 1011 : 12 × TQ 0100 : 5 × TQ 1100 : 13 × TQ 0101 : 6 × TQ 1101 : 14 × TQ 0110 : 7 × TQ 1110 : 15 × TQ 0111 : 8 × TQ 1111 : 16 × TQ

15.4.19 CANGIF (全球 中断 标志 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读：读作"0". 写：写作 "0".
7	RFPF	R/W	遥控帧中止标志 0:无遥控帧已经接收. 1:遥控帧已经接收. (接收邮箱中) 匹配<RFH> 位为 "1" 的传输 邮箱时此位不会设置.
6	WUIF	R/W	唤醒中断标志 0:睡眠模式或正常工作模式中 1:睡眠模式已经消除.
5	RMLIF	R/W	接收信息丢失中断标志 0:无接收信息丢失错误发生. 1: 接收信息丢失错误发生在至少一个配置为接收邮箱的邮箱中.
4	TRMABF	R/W	传输中止标志 0:无传输中止发生. 1:传输中止发生. (至少 CANAA 寄存器中一 位被设置.)
3	TSOIF	R/W	时间标志计数器溢出中断标志 0: 此位最后清除后无溢出发生在时间标志计数器中. 1: 此位最后清除后至少时间标志计数器一个溢出.
2	BOIF	R/W	总线关闭中断标志 0: CAN 控制器 在总线开启模式. 1: CAN 控制器 在总线关闭模式.
1	EPIF	R/W	错误被动中断标志 0: CAN 控制器在错误激活模式. 1: CAN 控制器在错误被动模式.
0	WLIF	R/W	警告电平中断标志 0:无错误计数器到达警告电平. 1: 至少一个错误计数器到达警告电平.

如果满足对应的全球中断条件全球中断标志寄存器 (CANGIF)的各中断标志会设置为 "1". 全球中断标志设置为"1"时, 如果全球中断屏蔽寄存器 (CANGIM)中对应的位为"1" (中断使能), CAN 全球中断 (INTCANGB) 会为"高".

CANGIF 寄存器可以通过写入 "1" 到 CANGIF 寄存器中对应的位清除. 写入"0" 无效.

15.4.20 CANGIM (全球 中断 屏蔽 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	RFPFM	WUIM	RMLIM	TRMABF	TSOIM	BOIM	EPIM	WLIM
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读：读作 "0". 写：写作 "0".
7	RFPFM	R/W	遥控帧中止中断 屏蔽 0:中断 禁止 1:中断 使能
6	WUIM	R/W	唤醒中断 屏蔽 0:中断 禁止 1:中断 使能
5	RMLIM	R/W	接收信息丢失中断 屏蔽 0:中断 禁止 1:中断 使能
4	TRMABF	R/W	传输中止中断 屏蔽 0:中断 禁止 1:中断 使能
3	TSOIM	R/W	时间标志计数器溢出中断 屏蔽 0:中断 禁止 1:中断 使能
2	BOIM	R/W	总线关闭中断 屏蔽 0:中断 禁止 1:中断 使能
1	EPIM	R/W	错误被动中断 屏蔽 0:中断 禁止 1:中断 使能
0	WLIM	R/W	警告电平中断 屏蔽 0:中断 禁止 1:中断 使能

全球中断屏蔽寄存器 (CANGIM) 控制使能或禁止全球中断对应 CANGIF 寄存器的各中断条件。CANGIF 寄存器中的位为 "0" 时, 对应的 CAN 全球中断 (INTCANGB) 禁止。CANGIF 寄存器中的位为 "1" 时, 对应的 CAN 全球中断 (INTCANGB) 使能。

复位工作清除所有 CANGIM 寄存器中的位为 "0", 禁止全球中断。

15.4.21 CANMBTIF (邮箱 传输 中断 标志 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	MBTIF30	MBTIF29	MBTIF28	MBTIF27	MBTIF26	MBTIF25	MBTIF24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	MBTIF23	MBTIF22	MBTIF21	MBTIF20	MBTIF19	MBTIF18	MBTIF17	MBTIF16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	MBTIF15	MBTIF14	MBTIF13	MBTIF12	MBTIF11	MBTIF10	MBTIF9	MBTIF8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	MBTIF7	MBTIF6	MBTIF5	MBTIF4	MBTIF3	MBTIF2	MBTIF1	MBTIF0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	-	R	读：读作 "0". 写：写作 "0".
30-0	MBTIF30 ~ MBTIF0	R/W	邮箱传输中断标志 (各位对应邮箱 30~0.) 邮箱 x 中信息成功传输时且 CANMBIM 寄存器的中断屏蔽使能 (<MBIMx>="1"), <MBTIFx> 位设置为 "1" 且传输完成中断(INTCANTX) 变为 "高" 电平. CANMBIM<MBIMx> 位为 "0"时, <MBTIFx> 位不设置且 INTCANTX 保持在"低" 电平. 传输完成通过读 CANTA 寄存器检查. 如果哪怕 CANMBTIF 寄存器中的一 位为 "1", INTCANTX 为"高" 电平. 通过从 CPU 写入"1" 到 <MBTIFx> 位清除<MBTIFx> 位 . 写入"0"无效.

邮箱设置为接收时, CANMBTIF 寄存器中对应的位读作 "0". 邮箱设置为传输时, CANMBRIF 寄存器中对应的位读作 "0".

15.4.22 CANMBRIF (邮箱接收中断标志寄存器)

	31	30	29	28	27	26	25	24
位符	MBRIF31	MBRIF30	MBRIF29	MBRIF28	MBRIF27	MBRIF26	MBRIF25	MBRIF24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	MBRIF23	MBRIF22	MBRIF21	MBRIF20	MBRIF19	MBRIF18	MBRIF17	MBRIF16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	MBRIF15	MBRIF14	MBRIF13	MBRIF12	MBRIF11	MBRIF10	MBRIF9	MBRIF8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	MBRIF7	MBRIF6	MBRIF5	MBRIF4	MBRIF3	MBRIF2	MBRIF1	MBRIF0
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-0	MBRIF31 ~ MBRIF0	R/W	<p>邮箱接收中断标志 (各位对应邮箱 31 ~ 0.)</p> <p>邮箱 x 成功接收信息且 CANMBIM 寄存器的中断屏蔽使能 (<MBIMx> = "1")时, <MBRIFx> 位设置为 "1" 且接收完成中断 (INTRX) 变为"高"电平.</p> <p>MBIM 寄存器中<MBIMx> 位为 "0"时, <MBRIFx> 位不设置且 INTRX 保持在 "低" 电平. 通过读 CANRMP 寄存器检查接收完成.</p> <p>如果哪怕 CANMBRIF 寄存器中一位为 "1", INTCANRX 为"高" 电平. 通过从 CPU 写入"1" 到 <MBRIFx> 位清除<MBRIFx> 位.</p> <p>写入"0" 无效.</p>

15.4.23 CANMBIM (邮箱 中断 屏蔽 寄存器)

	31	30	29	28	27	26	25	24
位 符	MBIM31	MBIM30	MBIM29	MBIM28	MBIM27	MBIM26	MBIM25	MBIM24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	MBIM23	MBIM22	MBIM21	MBIM20	MBIM19	MBIM18	MBIM17	MBIM16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	MBIM15	MBIM14	MBIM13	MBIM12	MBIM11	MBIM10	MBIM9	MBIM8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	MBIM7	MBIM6	MBIM5	MBIM4	MBIM3	MBIM2	MBIM1	MBIM0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	MBIM31 到 MBIM0	R/W	邮箱中断屏蔽 0: 对应的邮箱中断 禁止 1: 对应的邮箱中断 使能

CANMBIM 决定中的设置, 使邮箱中断生成使能或禁止. 如果 CANMBIM 中的位为 "0", 对应的邮箱中断生成禁止且如果其为 "1", 中断生成使能. CANMBIM 的复位值为 "0".

15.4.24 CANCDR (改变 数据 请求 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	CDR30	CDR29	CDR28	CDR27	CDR26	CDR25	CDR24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	CDR23	CDR22	CDR21	CDR20	CDR19	CDR18	CDR17	CDR16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	CDR15	CDR14	CDR13	CDR12	CDR11	CDR10	CDR9	CDR8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CDR7	CDR6	CDR5	CDR4	CDR3	CDR2	CDR1	CDR0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	-	R	读：读作 "0". 写：写作 "0".
30-0	CDR30 ~CDR0	R/W	改变数据请求 (各位对应邮箱 30 ~0.) 传输 邮箱 x 的<CDRx> 位设置为 "1"时, 此邮箱 x 传输请求忽略. 意味着 CANTRS<TRSx> 位和 <CDRx> 位设置的邮箱 x 会从内部仲裁范围排除且如果传输没有开始不会传输. <CDRx> 位清除为"0"后, 邮箱 x 回到被包含在内部仲裁范围.

注: 邮箱 31 为仅接收邮箱.

更新遥控帧的自动承认的传输邮箱 x 的数据域使能 (CANMBnID<RFH>="1")时, 改变数据请求寄存器 CABCDR 有效. 邮箱 x 使能自动承认开始信息传输自动响应接收遥控帧且信息传输 (此时, 通过传输更新的数据中途被输出) 期间, 可以更新数据域. 数据域的更新可以通过设置 <CDRx> 位为"1" 避免且暂时挂起数据传输.

15.4.25 CANRFP (遥控 帧 挂起 寄存器)

	31	30	29	28	27	26	25	24
位 符	RFP31	RFP30	RFP29	RFP28	RFP27	RFP26	RFP25	RFP24
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	RFP23	RFP22	RFP21	RFP20	RFP19	RFP18	RFP17	RFP16
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	RFP31 ~ RFP0	R/W	遥控帧中止(各位对应邮箱 31~0.) 配置为接收邮箱的邮箱 x 接收遥控帧时, <RFPx> 位和 CANRMP<RMPx> 位设置为 "1". <RFPx> 位可以通过写入"1" 到 CANRMP<RMPx> 位清除.

CANRFP<RFPx> 位通过内部逻辑设置且可以从 CPU 用写入 "1" 到 CANRMP<RMPx> 位清除. <RMPx> 位也同时清除. 从 CPU 写入"0" 到 <RMPx> 位和写入"1"或"0" 到 <RFPx> 位无效. 即使带 <RFPx>="1"的邮箱 x 通过数据帧接收被覆盖时, <RFPx> 位清除.

遥控帧中止中断通过设置全球中断屏蔽寄存器 CANGIM 中的 <RFPM> 位中为 "1"使能时, CAN 全球中断 INTCANGB 发生.

15.4.26 CANCEC (错误 计数器 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读: 读作 "0". 写: 写作 "0".
15-8	TEC[7:0]	R	8-位 过渡 错误 计数器 (复位释放后)
		R/W	8-位 过渡 错误 计数器 (CANMCR<TSTERR>="1")
7-0	REC[7:0]	R	8-位 接收 错误 计数器 (复位释放后)
		R/W	8-位 接收 错误 计数器 (CANMCR<TSTERR>="1")

CAN 控制器包含两错误计数器：接收错误计数器 <REC> 和传输错误计数器 <TEC>. 计数器值都可以从 CPU 读. 仅可以在测试错误模式 (CANMCR 寄存器中<TSTERR> 位为"1")写入进入错误计数器. 写入 CANCEC 寄存器时, 低 8 位 <REC> 的写入数据也被写入高 8 位 (TEC).

根据 CAN 规格 2.0BCAN 错误计数器递增或递减计数.

超过错误被动极限 (128)后<REC>不增加. <REC>=128 时, 正确信息接收后, <REC>设置到值 119 和 127 之间. 到达 "总线关闭" 状态后, 错误计数器未定义.

如果状态 "总线关闭" 到达, 总线上 11 连续逆行位后, 接收错误计数器被增量. 如果计数器到达计数 128, 模组自动变为状态错误激活. 所有内部标志复位且错误计数器清除为 "0". 配置寄存器保持编程的值. "总线关闭" 状态期间计数器值未定义.

CAN 进入配置模式时, 错误计数器会清除.

15.4.27 CANTSP (时间 标志 计数器 预分频 寄存器)

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

位	位 符	类型	功能
31-4	-	R	读：读作 "0". 写：写作 "0".
3-0	TSP[3:0]	R/W	时间标志计数器 预分频 设置被加载值到 4-位 TSC 预分频.

确保写入邮箱周期期间 CANTSC 的值不会改变, 保持寄存器被执行. CANTSC 的值会被复制到保持寄存器然后如果信息成功接收或传输从保持寄存器写入邮箱. 如果除了帧的结束的最后位没有错误, 接收器接收成功. 如果直到帧的结束的最后位没有错误, 传输器传输成功. (参考 CAN 规格 2.0B).

15.4.28 CANTSC (时间标志计数器寄存器)

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

位	位符	类型	功能
31-16	-	R	读：读作 "0". 写：写作 "0".
15-0	TSC[15:0]	R/W	时间标志计数器 自由-运行 16-位 计数器

CANTSC 的溢出可以被全球中断标志寄存器 (CANGIF)的时间标志计数器溢出中断标志 <TSOIF> 检测, 和全球状态寄存器 (CANGSR)的时间标志计数器溢出标志 <TSO>. 标志都可以通过写入 "1" 到 CANGIF 寄存器中 <TSOIF> 清除.

有 4-位 CANTSC 预分频. 上电时间标志计数器 被直接从位时钟 (<TSP[3:0]>="0")驱动后. 时间标志计数器的 T_{TSC} 期间会被用 如下方程式计算 :

$$T_{TSC} = T_{BIT} \times (TSP + 1)$$

15.5 各电路工作说明

15.5.1 邮箱

邮箱由单端口 RAM (可从内部 CAN 核和 CPU 进入) 组成。CPU 控制 CAN 控制器通过改变邮箱和控制寄存器设置。邮箱和控制寄存器设置被用作诸如接收过滤, 信息传输, 和中断处理中处理。

要开始传输, 设置对应的邮箱的传输请求位传输信息。设置位以后, 所有传输处理和错误处理 (错误发生时) 被执行而没有 CPU 参与。邮箱设置为接收时, CPU 使用读指令读邮箱数据。用户也可以设置它为了每次信息成功接收或传输时, 中断会生成到 CPU。

整体上, 32 邮箱被提供, 各由 8 字节数据组成, 29 位 ID, 和几个控制位。邮箱 (除邮箱 31) 可以设置传输或接收。邮箱 31 仅为接收邮箱。设计邮箱 31 为了它可以接收不同使用别的接收屏蔽的信息 ID 组而不是邮箱 0 到 30。

图 15-2 表示邮箱配置。

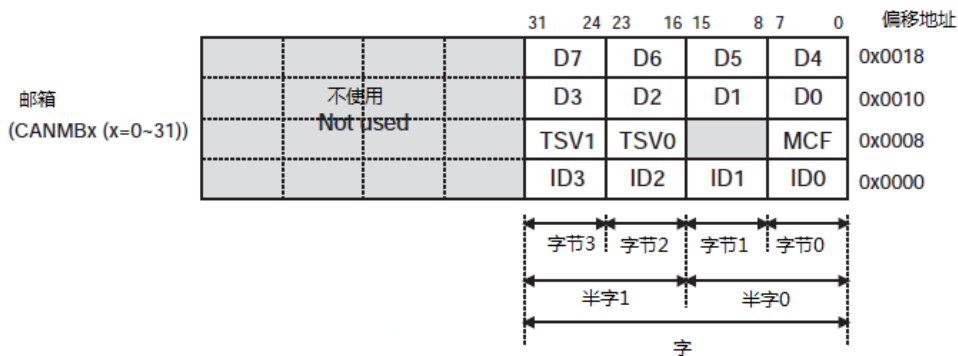


图 15-2 邮箱配置

1. 信息 ID 域 (ID3 到 ID0)
 - ID 扩展位 <IDE>
 - 全球 / 地方 接受屏蔽使能位 <GAME / LAME>
 - 遥控帧处理位 <RFH>
 - 29-位 信息 ID <ID[28:0]>
2. 信息 控制 域 (MCF)
 - 遥控帧传输请求位 <RTR>
 - 4 位数据长 <DLC[3:0]>
3. 时间标志 值 (TSV1, TSV0)

接收 / 传输 信息期间存储时间标志计数器值. <TSV[15:0]>
4. 数据 域 (D7 ~ D0)

8 字节数据 <D7[7:0]> ~ <D0[7:0]>

15.5.2 传输 控制 寄存器

传输控制由两寄存器组成. 一个是传输请求设置寄存器 CANTRS, 和另一个是传输请求复位寄存器 CANTRR. 因此它可以清除传输请求在处理状态-机器中传输邮箱而不生成冲突. 此架构也防止清除已经在处理中的传输邮箱的传输请求.

写入数据 ID 到配置为传输邮箱 (CANMD<MDx>="0")的邮箱 x 被执行且进入邮箱 x 使能 (CANMC<MCx>="1")时, 设置 CANTRS<TRSx> 位为 "1" 使邮箱 x 中信息被传输.

如果有超过一个配置为传输邮箱的邮箱且大于一个对应的 TRS 位被设置,那么信息会用选择的命令被传输. 传输命令取决于主机控制寄存器 CANMCR 中<MTOS> 位.

如果 CANMCR<MTOS> 位为 "0", 带较低数邮箱有更高优先级. 比如, 如果邮箱 CANMB0, CANMB2, 和 CANMB5 配置为传输邮箱且对应 CANTRS<TRSx> 位被设置为 "1",那么信息会用以下命令传输: CANMB0, CANMB2, 和 CANMB5. 如果 CANMB2 信息处理期间, 新传输请求设置给 CANMB0, 那么在下一个内部仲裁运行, CANMB0 被选择为下一个传输信息且 CANMB2 传输完成后 CANMB0 信息传输开始. 这也会发生在 CANMB2 信息被传输时, 仲裁丢失错误发生时. CANMB0 信息会被传输取代 CANMB2 仲裁中丢失.

如果 CANMCR<MTOS> 位为 "1", 在那些传输被请求的带最高优先级 ID 的邮箱间的邮箱会被传输. 在仲裁丢失错误发生后的传输中, 在那些传输被请求的带最高优先级 ID 的邮箱间的邮箱中信息会被传输.

15.5.3 接收控制寄存器

接收信息的 ID 与设置为接收邮箱的邮箱的 ID 比较. ID 比较取决于邮箱中全球/地方接受屏蔽使能位 MBnID<GAME> / <LAME> 值和被保持在全球/地方接受屏蔽寄存器 CANGAM/CANLAM 中的数据.

匹配被检测时, 接收信息的 ID, 控制位, 和数据字节写入匹配邮箱. 同时, 对应的接收信息中止位 CANRMP<RMPx> 设置为"1" 且邮箱中断使能 (CANMBIM<MBIMx>="1 ") 时, CAN 接收完成中断 INTCANRX 发生. 匹配被检测后, 没有进一步 ID 比较发生.

如果接收信息的 ID 不匹配任何 0 到 30 邮箱, ID 仅比较接收邮箱 31 的 ID. 匹配被检测时, 接收信息的设置被写入仅接收邮箱 31.

如果无匹配被检测, 接收信息不会储存在邮箱且邮箱中无改变发生.

读数据后<RMPx> 位必须通过 CPU 清除. 用 <RMPx> 位设置为 "1", 如果此邮箱 x 下一个信息被接收, 对应的接收信息丢失位 <RMLx> 设置为"1". 此时, 邮箱 x 用新信息覆盖.

图 15-3 表示接收信息丢失发生时时序.

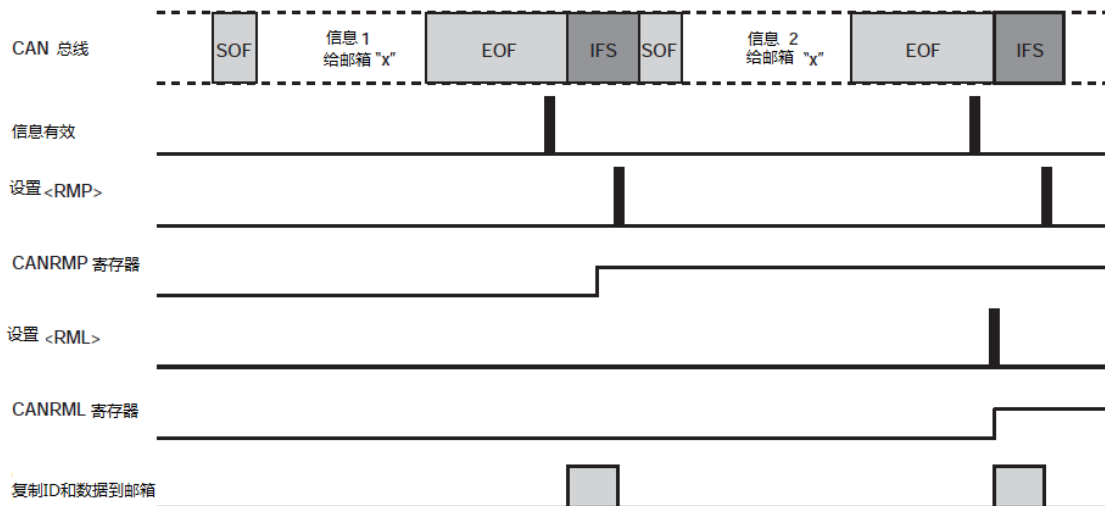


图 15-3 接收信息丢失发生时时序

15.5.4 遥控 帧 控制 寄存器

遥控帧接收后, 遥控帧 ID 比较邮箱 ID. ID 比较取决于邮箱中全球/地方接受屏蔽使能位 CANMBxID 的 <GAME> / <LAME> 值和被保持在全球/地方接受屏蔽寄存器 GAM/LAM 中的数据.

ID 匹配被检测后, 没有进一步比较发生.

遥控 帧 ID 匹配遥控帧处理位 CANMBxID<RFH> 设置为 "1" 的传输邮箱 n 的 ID 时, CANTRS<TRSx> 位会设置为 "1" 为了信息传输响应遥控帧. 关于 CANMBxID<RFH> 位设置为 "0" 的传输邮箱, 邮箱不响应遥控帧即使 ID 匹配.

如果 ID 匹配接收邮箱 n 的 ID, 接收信息与数据帧一样被处理, 且此设置 CANRMP<RMPx> 位和 CANRFP<RFPx> 位为 "1".

遥控帧 ID 匹配 CANMBxID<RFH> 位和 <GAME> 位都设置为 "1" 邮箱 n 的 ID 时, 邮箱的 ID 用遥控帧 ID 覆盖, 且邮箱会使用 ID (<TRSx> 位设置到传输数据帧) 自动响应遥控帧. 因此, 全球接受屏蔽寄存器 CANGAM 被用时, 一个邮箱 x 可能响应多个遥控帧 ID 取决于屏蔽值.

15.5.5 接收过滤

关于邮箱 0 到 30, 如果邮箱中位 <GAME> 被设置, 全球接受屏蔽寄存器 CANGAM 会被用. 接收信息会用匹配的 ID 储存在第一个邮箱. 如果邮箱中没有匹配的 ID 0 到 30, 接收信息会比较仅接收邮箱 (邮箱 31). 如果邮箱 31 中 <LAME> 位设置, 地方接受屏蔽寄存器 CANLAM 会被用.

图 15-4 表示接收过滤.

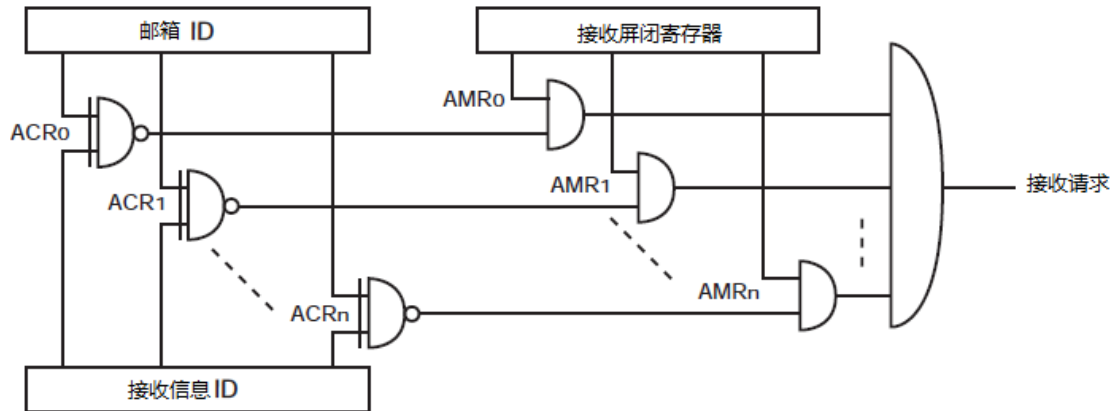


图 15-4 接收过滤器

15.5.6 时间 标志 功能

在 CAN 控制器中有自由-运行 16-位时间标志计数器 (CANTSC) 被执行表示信息接收和传输的时间. 接收信息已经储存或信息已经传输时, CANTSC 的内容写入到对应的邮箱的时间标志值(TSV).

CANTSC 被 CAN 总线线的位时钟驱动. CAN 工作 模式在配置模式或睡眠模式下, CANTSC 会停止. 上电复位后,写入时间标志计数器预分频寄存器 (CANTSP) 清除 CANTSC 为 "0". 在配置模式和正常工作模式下 CANTSC 从 CPU 可读且可写.

图 15-5 表示定时器标志计数器架构.

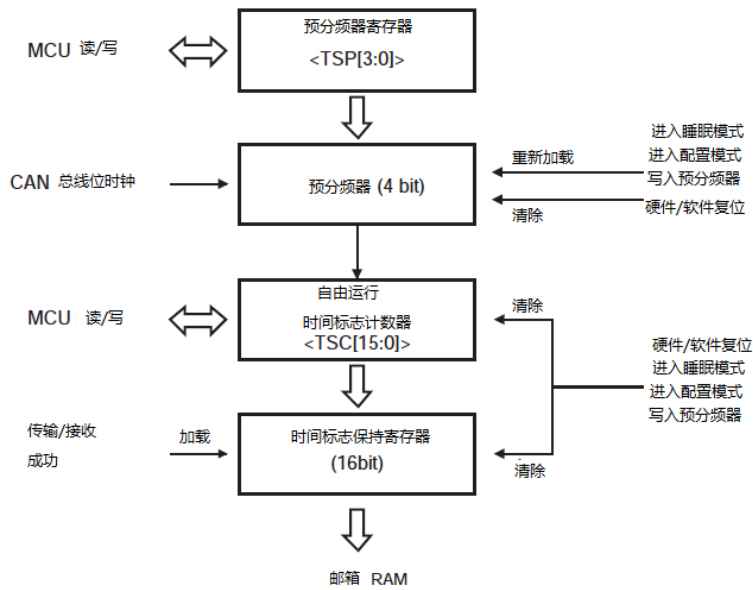


图 15-5 时间标志 计数器

以下情况自由运行时间标志计数器和时间标志保持寄存器会清除:

- 复位后(上电复位或软件复位)
- 控制器进入到配置模式时
- 控制器进入到睡眠模式时
- 写入进入执行到 CANTSP 寄存器时.

15.5.7 中断控制

CAN 控制器有以下中断源. 且这些中断源分割到三个组且各组有一中断输出.

- CAN 传输完成中断 (INTCANTX)
 发生在传输完成
- CAN 接收完成中断 (INTCANRX)
 发生在接收完成
- CAN 全球中断 (INTCANGB)
 通过八个源发生而非以上.

源		组
传输中断	:信息已经成功传输.	INTCANTX
接收中断	:信息已经成功接收.	INTCANRX
警告电平中断	:至少两错误计数器之一大于或等于 97.	INTCANGB
错误被动中断	:CAN 进入错误被动模式.	
总线关闭中断	:CAN 进入总线关闭模式.	
时间标志溢出中断		
传输中止中断		
接收信息丢失中断		
唤醒中断	:从睡眠模式唤醒后, 此中断会生成.	
遥控帧接收中断		

关于邮箱中断, 有两个从全球中断分离出来的中断输出线. 这些是邮箱接收完成中断 (INTCANRX) 和邮箱传输完成中断 (INTCANTX), 取决于邮箱设置.

有两个中断标志寄存器和一个中断屏蔽寄存器. 一个中断标志寄存器是给邮箱接收中断标志寄存器 (CANMBRIF) 且一个给邮箱传输中断标志寄存器 (CANMBTIF). 另外, 有设置使能或禁止各邮箱中断的邮箱中断屏蔽寄存器 (CANMBIM). CANMBIM 寄存器都被用来传输和接收邮箱.

图 15-6 表示 CAN 中断信号方块图.

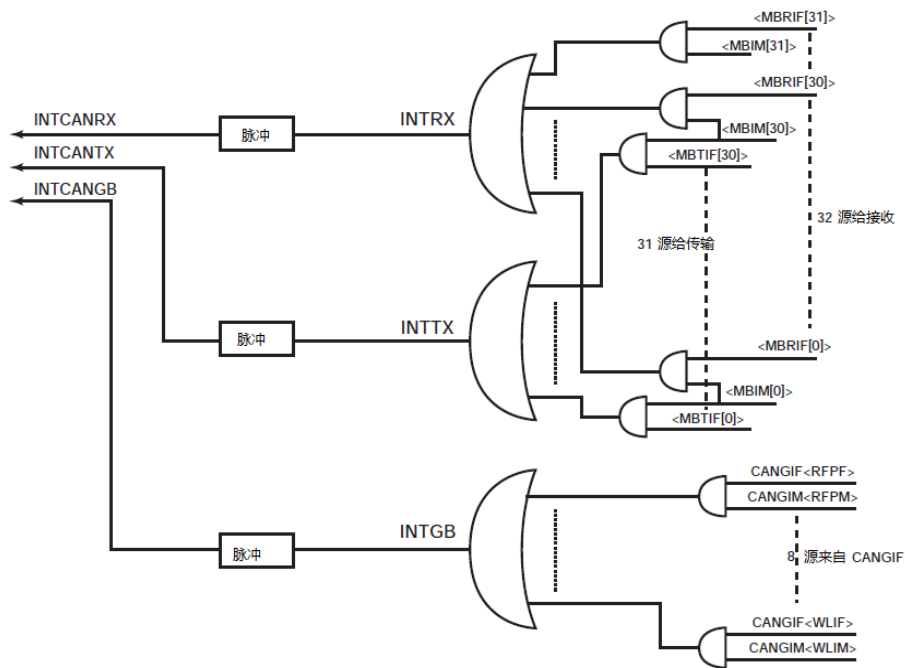


图 15-6 CAN 中断信号方块图

CAN 接收完成中断信号 INTRX 为通过用邮箱中断屏蔽寄存器 CANMBIM 各位取 AND 的邮箱接收中断标志寄存器 CANMBRIF 生成的 32 源的 OR 信号。

CAN 传输完成中断信号 INTTX 为通过用邮箱中断屏蔽寄存器 CANMBIM 各位取 AND 的邮箱传输中断标志寄存器 CANMBTIF 生成的 31 源的 OR 信号。

CAN 全球中断信号 INTGB 为通过用全球中断屏蔽寄存器 CANGIM 各位取 AND 的全球中断标志寄存器 CANGIF 生成的 8 源的 OR 信号。

15.6 工作模式

15.6.1 配置模式

开始工作前(位配置寄存器, CANBCR1 和 CANBCR2 设置) CAN 控制器需要初始设置. 仅当 CAN 控制器在配置模式时可以写入 CANBCR1 和 CANBCR2.

复位后, CANMCR<CCR> 和 CANGSR<CCE> 设置为 "1" 且配置模式设置. 写入 "0" 到<CCR> 位设置 CAN 控制器为正常工作模式. 离开配置模式后, <CCE> 位清除为"0" 且上电顺序开始. 在CAN 总线上上电顺序检测到 11 连续逆行位. 检测后, CAN 控制器在总线工作和准备中.

写入 "1" 到<CCR> 位设置 CAN 控制器从正常工作模式进入配置模式. CAN 控制器进入配置模式后, <CCE> 位设置为 "1".

图 15-7 表示 CAN 控制器初始设置的流程图

CAN 控制器进入到配置模式时, CAN 错误计数器 (CANCEC), 时间标志计数器 (CANTSC), 和时间标志保持寄存器会清除.

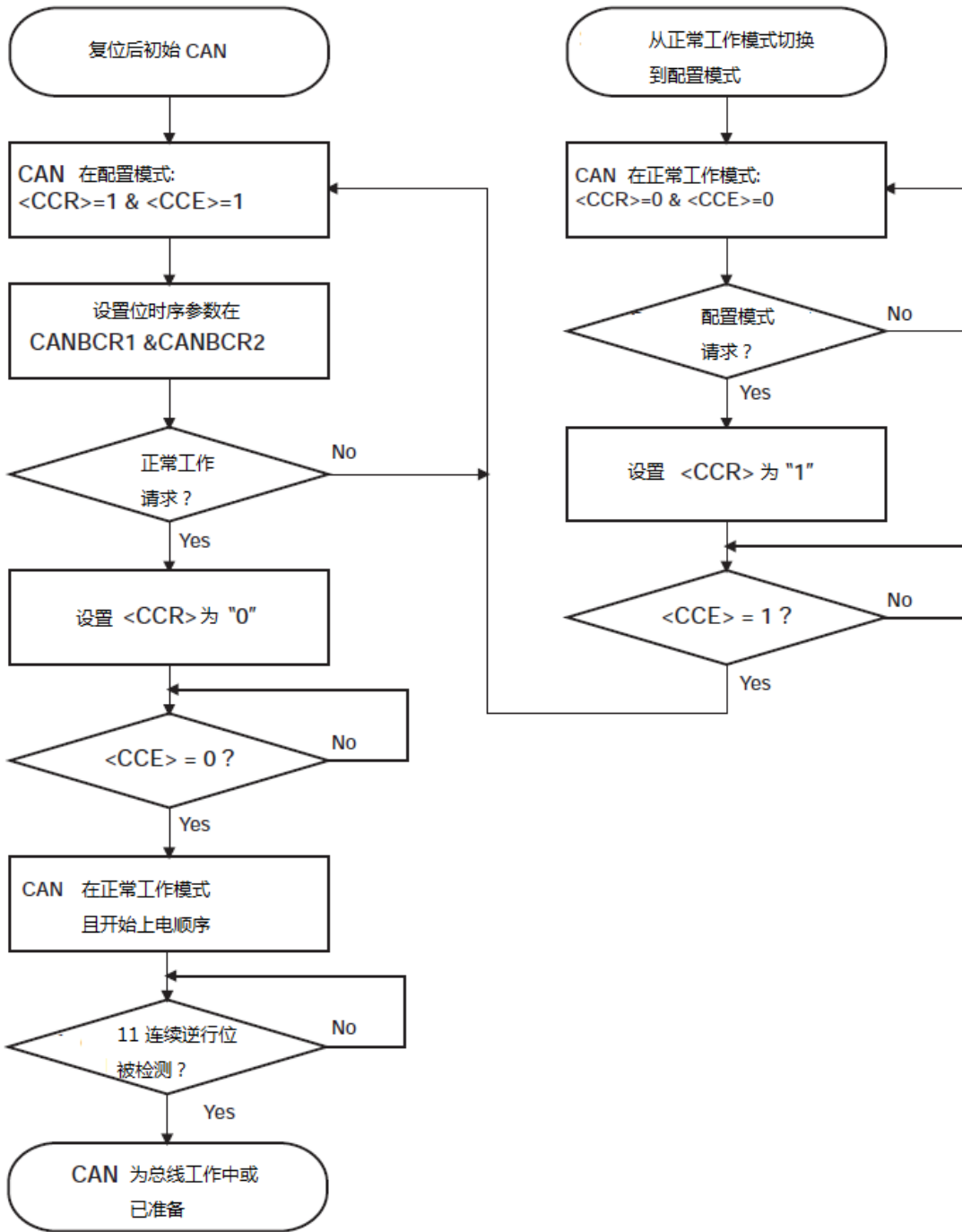


图 15-7 CAN 控制器初始设置流程图

15.6.2 睡眠模式

睡眠模式通过写入 "1" 到 CANMCR 寄存器中 <SMR> 位被请求。CAN 控制器进入到睡眠模式后, CANGSR<SMA> 位设置为 "1".

读 CANGSR 寄存器的值为 0xF040. 此意味着传输缓冲中没有信息且睡眠模式为激活 <SMA> 位为 "1". 读所有别的寄存器值递交值 0x0000. 写入进入到所有寄存器, 除 CANMCR 寄存器, 会被否认.

如果写入进入到 CANMCR 寄存器被检测, CAN 控制器消除睡眠模式 (唤醒) 且开始上电顺序, 或者在 CAN 总线上用 CANMCR 检测到有任何总线活动 <WUBA> 位设置为 "1". CAN 控制器等待直到在 CANRX 输入 端子检测 11 连续逆行位, 其走到总线激活状态后. 唤醒信息无效.

睡眠模式下, CAN 错误计数器和所有传输请求设置 CANTRS<TRSx> 位和传输请求复位 CANTRR<TRRx> 位清除. CAN 控制器离开睡眠模式后 <SMR> 位和 <SMA> 位清除.

CAN 控制器传输信息 (CANMCR<SMR>="1") 时, 如果睡眠模式被请求, 任何以下发生后 CAN 控制器进入睡眠模式:

- 信息已经成功传输.
- 仲裁丢失错误后信息已经成功传输.
- 仲裁丢失错误后信息已经成功接收.

15.6.3 挂起模式

挂起模式通过写入 "1" 到 CANMCR<SUR> 位被请求. 如果 CAN 总线 线不空闲, 挂起模式被激活前, 当前信息传输/接收完成. CAN 控制器进入挂起模式后, CANGSR<SUA> 位设置为 "1".

挂起模式下, CAN 总线线上 CAN 控制器不激活. 意味着错误帧或承认都不会被传输. 错误计数器和 CANGSR<EP> 位也不会清除.

总线关闭恢复顺序执行期间如果挂起模式被请求, 总线关闭恢复顺序结束后 CAN 控制器进入挂起模式.

要重启 CAN 控制器, <SUR> 位需要编程到 "0". 离开总线关闭状态或非激活状态后, CAN 控制器重启总线关闭恢复顺序.

用写入 "0" 到 <SUR> 位 CAN 控制器消除挂起模式.

15.6.4 测试循环后模式

测试循环后模式下, CAN 控制器可以接收其自己传输的信息和生成其自己的承认. 没有别的 CAN 节点需要工作.

仅 CAN 控制器在挂起模式时测试循环后模式可以使能或禁止. 测试循环后模式下, CAN 控制器可以从邮箱传输信息且在另一个邮箱接收它. 邮箱设置相同于正常工作模式.

15.6.5 测试错误模式

测试错误模式下, 可以写入 CAN 错误计数器 寄存器 (CANCEC). 较低 8 位值同时写入传输错误计数器 (CANTEC) 和接收错误计数器 (CANREC). 可以写入到 错误计数器最大值为 255. 强制 CAN 控制器总线关闭模式下的 256 错误计数器值不可以写入.

仅 CAN 控制器在挂起模式时, 测试错误模式可以使能或禁止.

图 15-8 表示测试循环后模式和测试错误模式设置流程图.

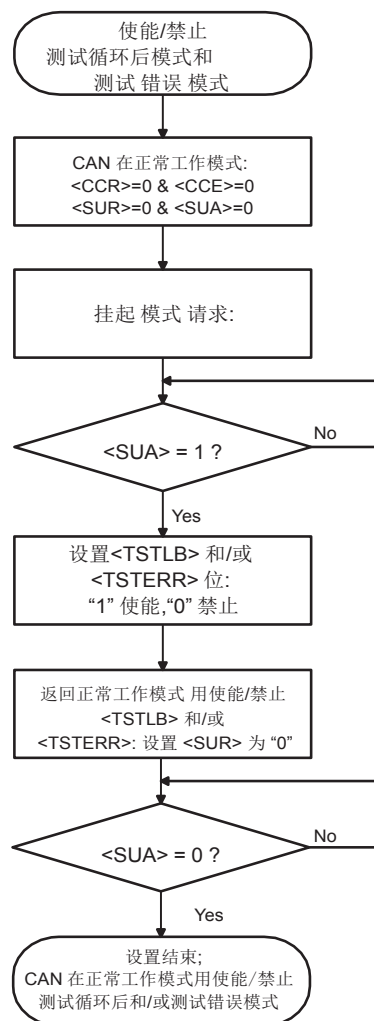


图 15-8 测试循环后模式和测试错误模式设置流程图

15.7 工作 描述

15.7.1 接收 信息

图 15-9 表示使用 CAN 接收完成中断(INTCANRX) 信息接收的流程图一例.

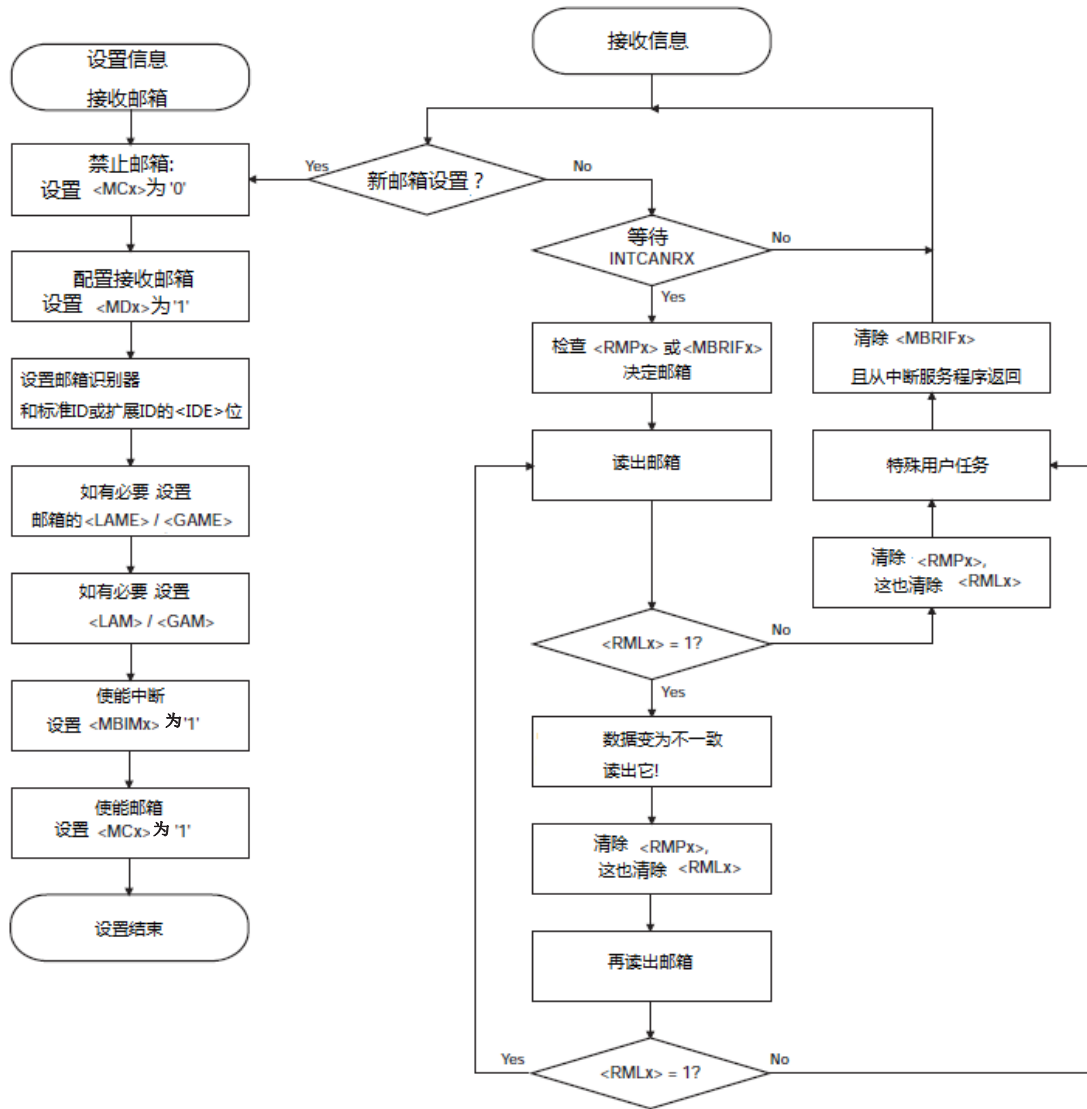


图 15-9 信息接收流程图

也可以使用调整取代接收中断.此时, 以上流程图"等待 INTCANRX" 必须用调整 CANRMP 替代. 再者, 使能中断和清除 CANMBRIF 必须从流程移除.

15.7.2 传输 信息

图 15-10 表示使用 CAN 传输完成中断(INTCANTX)的信息传输流程图的一例。

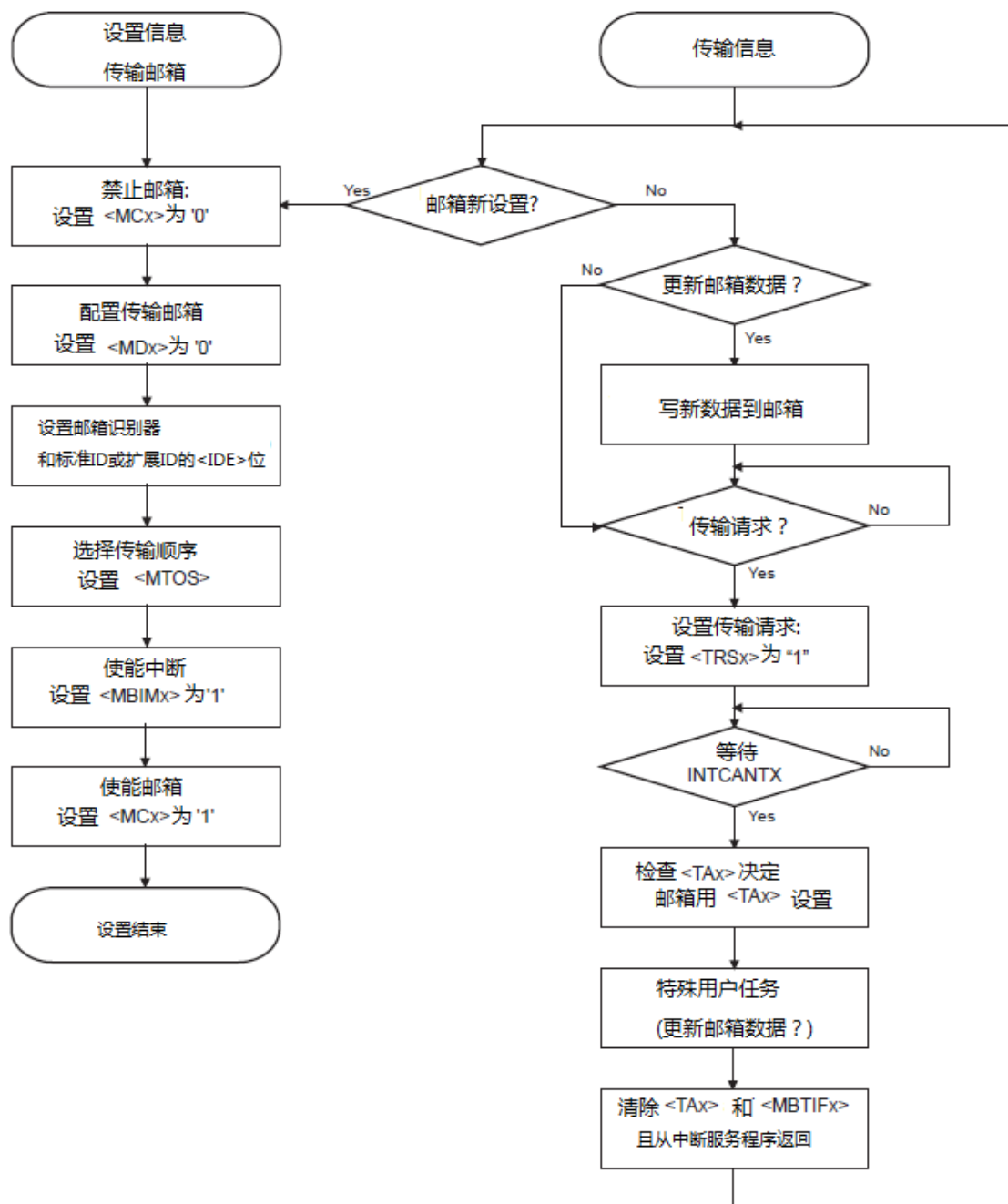


图 15-10 传输信息流程图

也可以使用调整 取代传输中断. 此时,以上流程图"等待 INTCANTX" 必须用轮询 TA 替代. 再者,使能中断和清除 CANMBTIF 必须从流程移除.

15.7.3 遥控 帧 处理

图 15-11 表示通过使用自动回复特征的遥控帧处理一例. 传输邮箱<RFH> 位设置为"1"时, 此特征可用. 更新邮箱数据时要避免数据不一致, 邮箱的数据更新期间 CANCDR 寄存器控制传输.

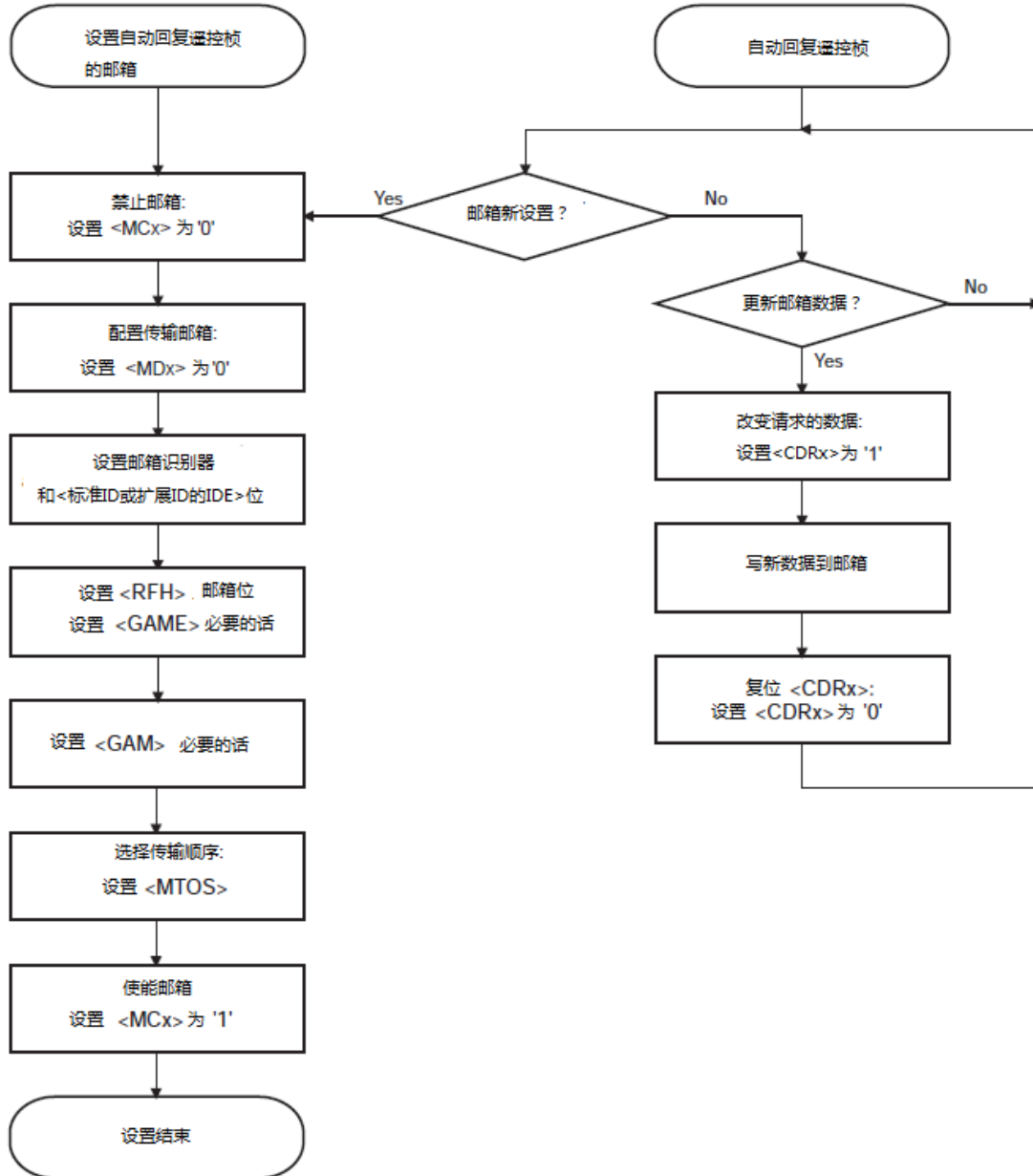


图 15-11 处理使用自动回复特征的遥控帧流程图

15.8 位配置

位长由参数 TSEG1, TSEG2, 和 BRP 决定. CAN 总线上所有控制器必须有相同波特率和位长. 独立控制器的不同时钟频率, 波特率由以上提到的参数调节. 位时序逻辑中, 需要的位时序的参数转换被执行. 配置寄存器 CANBCR1 和 CANBCR2 包含关于位时序的数据. 其定义对应 CAN 规格 2 (等同于 Intel 82527).

图 15-12 表示 CAN 位时序.

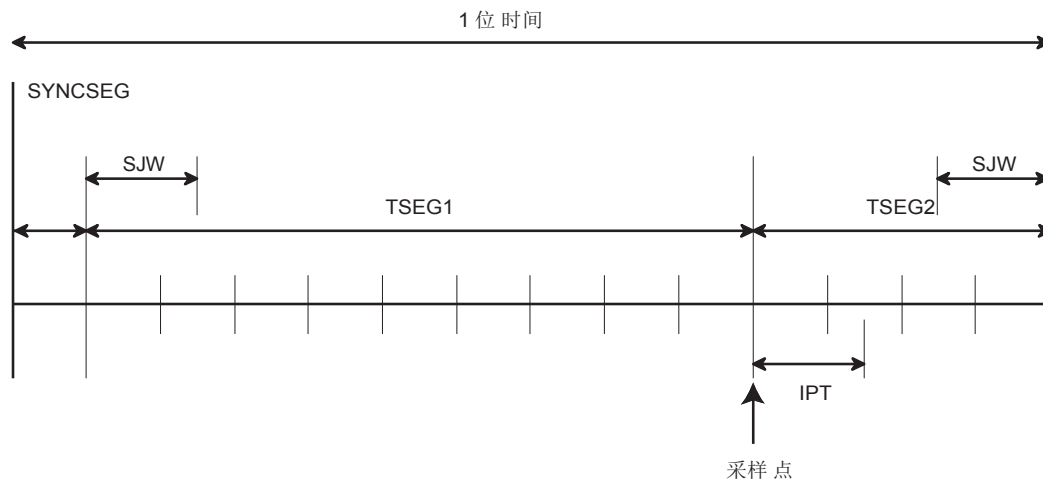


图 15-12 CAN 位 时序

T_{SCL} (CAN 系统 时钟) 通过以下定义:

$$T_{SCL} = \frac{\langle BRP[9:0] \rangle + 1}{f_{CANOSC}}$$

$1 \times T_{SCL} = 1 \times T_Q$ (T_Q : 时间 总量)

f_{CANOSC} 为 CAN 波特率生成时钟. 通过用 4 分割系统时钟 f_{SYS} 获得的时钟被供应为 CAN 波特率生成时钟. 如果 $f_{SYS} = 48\text{MHz}$ 那么 $f_{CANOSC} = 12\text{MHz}$.

同步部分 同步 SEG 始终有一 T_Q 长. 波特率通过以下定义:

$$BR = \frac{1}{((\langle TSEG1[3:0] \rangle + 1) + (\langle TSEG2[2:0] \rangle + 1) + 1) \times T_{SCL}}$$

注: $\langle TSEG1[3:0] \rangle$ 和 $\langle TSEG2[2:0] \rangle$ 为 CANBCR2 寄存器值. 不是 T_Q 单位值.

信息处理时间 (IPT) 为用采样位电平处理的保留的采样点开始的时间部分. 信息处理时间等于三个 CAN 系统时钟周期.

<SJW[1:0]>指再同步时位长中允许多少时间总量 (T_Q) 值被加长或减短. "1" (<SJW[1:0]> = 00) 和 "4" (<SJW[1:0]> = 11) 之间值可调. 总线线被采样且在各位格中总线信号下降缘执行同步. 关于 <SJW[1:0]>, 设置等于或小于<TSEG2[2:0]> 值.

设置 <SAM> 位使能总线线多采样. 电平由从三个采样值多数决定的结果决定. 在采样点且之前的最后两 CAN 系统时钟点采样. <BRP[9:0]> 小于 4 时, 始终执行一次采样无论 <SAM>位中设置的值.

表 15-2 表示波特率设置时限制.

表 15-2 波特率设置时限制

<BRP[9:0]>	T _Q 长 (CAN 时钟 周期数)	IPT 长 (CAN 时钟 周期数)	最小 TSEG2 长 (T _Q 单位)
0	1	3	3
1	2	3	2
> 1	<BRP[9:0]>+1	3	2

- TSEG1 限制

TSEG1 ≥ TSEG2 : TSEG1 长应该等于或大于 TSEG2 长.

- SJW 限制

SJW ≤ TSEG2 : 关于同步跳跃宽, 设置等于或小于 TSEG2 的值.

- SAM 限制

在 <BRP[9:0]>小于 4 的条件下不允许三-次采样. 关于 <BRP[9:0]> < 4 的条件,会始终执行一-次采样无论 SAM 的值.

例 : 关于 500 K 位/s

有 2μs 长的位. 如果 f_{CANOSC} = 12 MHz, 波特率预分频设置为 "1".意味着此数据传输率的位需要用 12T_Q 长编程. 根据以上方程式, 被编程值始终比被计算值小一:

<BRP[9:0]> = 00_0000_0001

<TSEG1[3:0]> = 0110 (7 T_Q)

<TSEG2[2:0]> = 011 (4 T_Q)

此时, 采样点为 8/12 = 66%.

可以其他 TSEG1 / TSEG2 的组合; 用 TSEG2 = 3 SJW 全范围.

SJW 应该始终设置 为可能的最高值. 不允许 SJW 大于 TSEG2.

不可以设置总线三-次采样因为 <BRP[9:0]> 的条件小于 4. 这样, 应该设置 SAM="0".

16. 12-位 模拟-到-数字 转换器

16.1 功能和特征

1. 可以选择模拟输入且从 PMD 或 TMRB(中断)接收 触发器 信号时开始 AD 转换.
2. 可以选择模拟输入, 软件触发器程序和恒定触发器程序中.
3. ADCs 有 十二个 AD 转换结果寄存器.
4. 在通过 PMD 触发器和 TMRB 触发器开始的程序结束时, ADCs 生成中断信号.
5. 在软件触发器程序结束时, ADCs 生成中断信号.
6. ADCs 有 AD 转换监视功能. 此功能使能时, 转换结果匹配被指定比较值时, 生成中断.

16.2 方块图

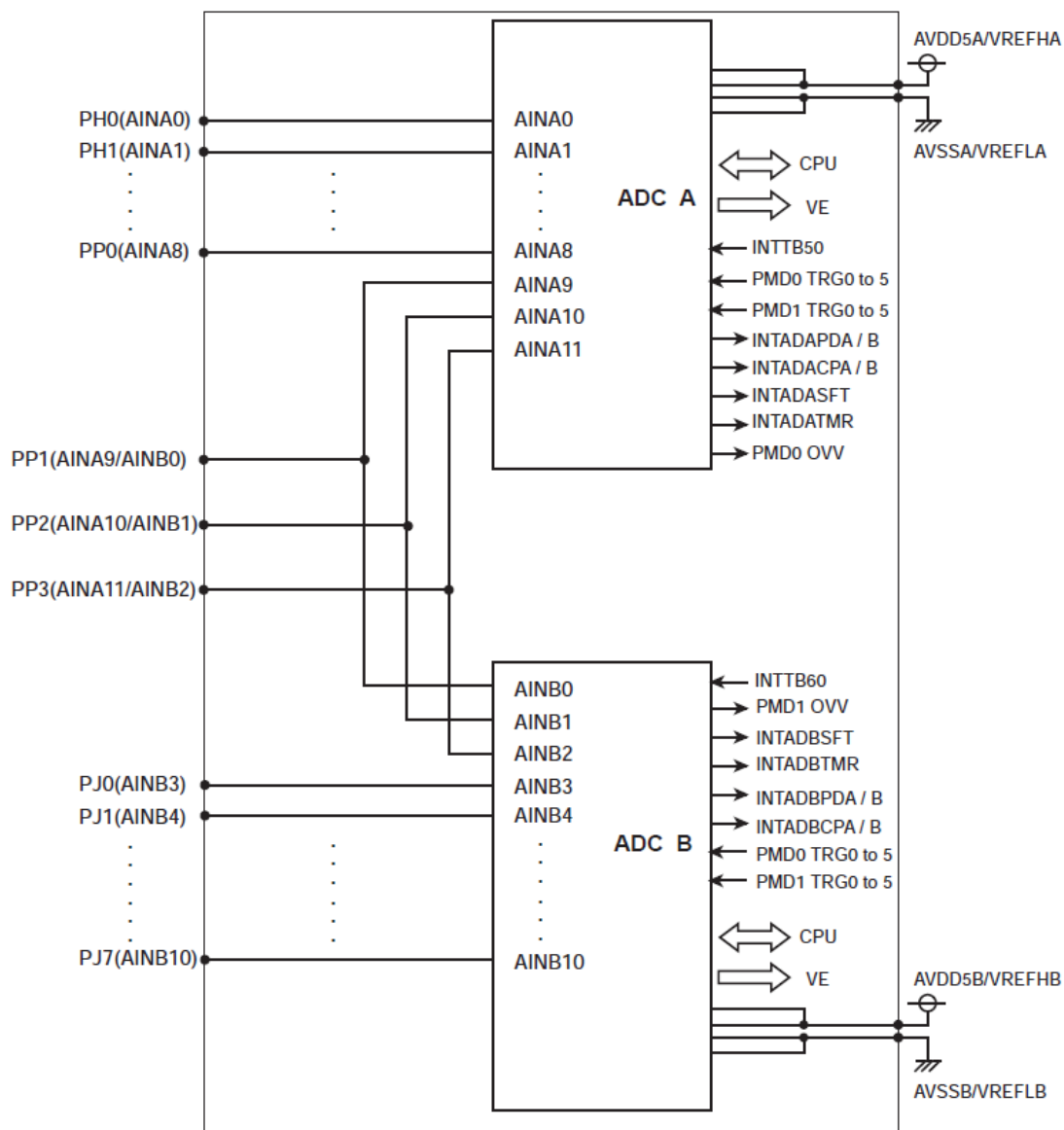


图 16-1 AD 转换器方块图

16.3 寄存器表

下表列出控制寄存器和其地址:

关于基地址, 参考关于"内存映射"章节中"外设功能基地址表".

寄存器名		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
监视器 设置 寄存器 0	ADxCMPCR0	0x0010
监视器 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
PMD 触发器 程序 数 选择 寄存器 0	ADxPSEL0	0x0050
PMD 触发器 程序 数 选择 寄存器 1	ADxPSEL1	0x0054
PMD 触发器 程序 数 选择 寄存器 2	ADxPSEL2	0x0058
PMD 触发器 程序 数 选择 寄存器 3	ADxPSEL3	0x005C
PMD 触发器 程序 数 选择 寄存器 4	ADxPSEL4	0x0060
PMD 触发器 程序 数 选择 寄存器 5	ADxPSEL5	0x0064
PMD 触发器 程序 数 选择 寄存器 6	ADxPSEL6	0x0068
PMD 触发器 程序 数 选择 寄存器 7	ADxPSEL7	0x006C
PMD 触发器 程序 数 选择 寄存器 8	ADxPSEL8	0x0070
PMD 触发器 程序 数 选择 寄存器 9	ADxPSEL9	0x0074
PMD 触发器 程序 数 选择 寄存器 10	ADxPSEL10	0x0078
PMD 触发器 程序 数 选择 寄存器 11	ADxPSEL11	0x007C
PMD 触发器 中断 选择 寄存器 0	ADxPINTS0	0x0080
PMD 触发器 中断 选择 寄存器 1	ADxPINTS1	0x0084
PMD 触发器 中断 选择 寄存器 2	ADxPINTS2	0x0088
PMD 触发器 中断 选择 寄存器 3	ADxPINTS3	0x008C
PMD 触发器 中断 选择 寄存器 4	ADxPINTS4	0x0090
PMD 触发器 中断 选择 寄存器 5	ADxPINTS5	0x0094
PMD 触发器 程序 寄存器 0	ADxPSET0	0x0098
PMD 触发器 程序 寄存器 1	ADxPSET1	0x009C
PMD 触发器 程序 寄存器 2	ADxPSET2	0x00A0
PMD 触发器 程序 寄存器 3	ADxPSET3	0x00A4
PMD 触发器 程序 寄存器 4	ADxPSET4	0x00A8
PMD 触发器 程序 寄存器 5	ADxPSET5	0x00AC

寄存器 名		地址(基+)
定时器 触发器 程序 寄存器 0 到 3	ADxTSET03	0x00B0
定时器 触发器 程序 寄存器 4 到 7	ADxTSET47	0x00B4
定时器 触发器 程序 寄存器 8 到 11	ADxTSET811	0x00B8
软件 触发器 程序 寄存器 0 到 3	ADxSSET03	0x00BC
软件 触发器 程序 寄存器 4 到 7	ADxSSET47	0x00C0
软件 触发器 程序 寄存器 8 到 11	ADxSSET811	0x00C4
恒定 转换 程序 寄存器 0 到 3	ADxASET03	0x00C8
恒定 转换 程序 寄存器 4 到 7	ADxASET47	0x00CC
恒定 转换 程序 寄存器 8 到 11	ADxASET811	0x00D0
模式 设置 寄存器 3	ADxMOD3	0x00D4
模式 设置 寄存器 4	ADxMOD4	0x00D8
模式 设置 寄存器 5	ADxMOD5	0x00DC

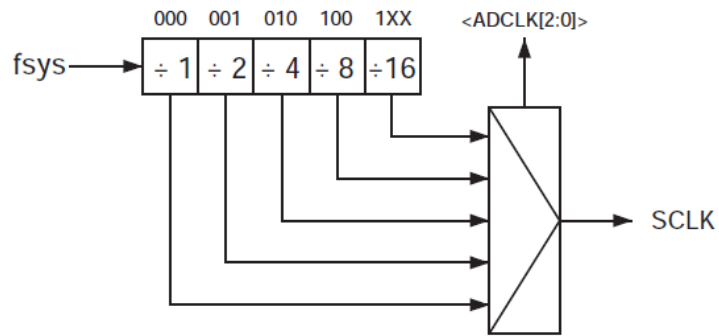
16.4 寄存器描述

在 ADC 时钟设置寄存器中选择的时钟频率执行 AD 转换。

16.4.1 ADxCLK (时钟 设置 寄存器)

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

位	位符	类型	功能
31-7	-	R	读作 "0".
6-3	EXAZ[3:0]	R/W	AIN 采样期间扩展. 0000: $([SCLK \text{ 周期}] \times 34) \times 1$ 0001: $([SCLK \text{ 周期}] \times 34) \times 2$ 0010: $([SCLK \text{ 周期}] \times 34) \times 3$ 0011: $([SCLK \text{ 周期}] \times 34) \times 4$ 0101: $([SCLK \text{ 周期}] \times 34) \times 16$ 0111: $([SCLK \text{ 周期}] \times 34) \times 64$ 1000: $([SCLK \text{ 周期}] \times 34) \times 128$ 1001: $([SCLK \text{ 周期}] \times 34) \times 256$ 1010: $([SCLK \text{ 周期}] \times 34) \times 512$ 1011: $([SCLK \text{ 周期}] \times 34) \times 1024$ 除以上外: 保留
2-0	VADCLK[2:0]	R/W	AD 预分频 输出 (SCLK) 选择 000: fsys (注 1) 001: fsys/2 010: fsys/4 011: fsys/8 1xx: fsys/16



注 1: SCLK 频率可以被用到 20MHz.

注 2: AD 转换在此寄存器中选择的时钟频率执行. 转换时钟频率必须选择确保精度.

注 3: AD 转换处理中转换时钟必须不被改变为.

16.4.2 ADxMOD0 (模式 设置 寄存器 0)

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

位	位 符	类型	功能
31-2	-	R	读作 "0".
1	DACON	R/W	DAC 控制 0: OFF 1: ON 设置 <DACON> 为 "1", 使用 ADC 时.
0	ADSS	W	软件 触发 转换 0: 忽略 1: 开始 设置 <ADSS> 为 "1" 开始 AD 转换 (软件 触发 转换). 从 PMD 或 TMRB(中断)接收触发器信号开始也 AD 转换. 关于详细设置, 请读关于 PMD 和 TMRB 章节.

注: 要开始 AD 转换, ADxMOD0<DACON> 设置为 "1" 后 ADxMOD1<ADEN> 设置为 "1".
然后, 执行软件触发转换开始或恒定 AD 转换控制使能. ADxMOD1<DACON> 设置为 "1" 后需要稳定时间 3 μ s.

16.4.3 ADxMOD1 (模式 设置 寄存器 1)

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	ADEN	R/W	AD 转换 控制 0: 禁止 1: 使能 设置 <ADEN> 为 "1", 使用 ADC 时.
6-1	-	R	读作 "0".
0	ADAS	R/W	恒定 AD 转换 控制 0: 禁止 1: 使能 设置 <ADEN> 为 "1"后,设置 <ADAS> 为 "1" 开始 AD 转换和重复转换.

16.4.4 ADxMOD2 (模式 设置 寄存器 2)

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

位	位 符	类型	功能
31-2	-	R	读作 "0".
1	ADSFN	R	软件转换忙标志 0: 转换 完成 1: 转换 处理中 <ADSFN> 为软件 AD 转换忙标志. <ADSS> 设置为 "1"后, AD 转换实际开始时, <ADSFN> 设置为 "1".所有通过软件 AD 转换触发器完成时, <ADSFN> 清除为 "0".
0	ADBFN	R	AD 转换忙标志 0: 转换 不在处理中 1: 转换 处理中 <ADBFN> 为 AD 转换忙标志. AD 转换开始时无论转换因素(PMD, 定时器, 软件, 恒定), <ADBFN>设置为 "1". AD 转换结束时, <ADBFN>清除为 "0".

16.4.5 ADxMOD3 (模式 设置 寄存器 3)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	PBSEL	-	BINMOD	BITS		-	RCUT
复位后	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
位 符	BIAS			-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15	-	R/W	写作 "0".
14	PBSEL	R/W	写作 "0".
13	-	R	读作 "0".
12	BINMOD	R/W	写作 "0".
11-10	BITS [1:0]	R/W	写作 "00".
9	-	R/W	写作 "0".
8	RCUT	R/W	减少 功耗 0: 正常 工作 1: Iref 切割 ADC 工作时, 事先写入 "0". ADC 停止工作时, 通过设置为 "1"可以减少功耗.
7-5	BIAS[2:0]	R/W	写作 "000".
4-0	-	R	读作 "0".

16.4.6 ADxMOD4 (模式 设置 寄存器 4)

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

位	位 符	类型	功能
31-12	-	R	读作 "0".
11-8	PHASEC[3:0]	R/W	写作 "0001".
7-4	PHASEB[3:0]	R/W	写作 "0001".
3	-	R	读作 "0".
2-0	PHASEA[2:0]	R/W	写作 "001".

16.4.7 ADxMOD5 (模式 设置 寄存器 5)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-13	PHASEF[2:0]	R/W	写作 "011".
12-8	PHASEE[4:0]	R/W	写作 "00110".
7-5	-	R	读作 "0".
4-0	PHASED[4:0]	R/W	写作 "01101".

16.4.8 ADxCMPCR0(监视器 设置 寄存器 0)

固定转换结果后, 中断信号 (INTADxCPA)生成.

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	CMPCNT0			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	CMP0EN	-	-	ADBIG0	REGS0			
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能												
31-12	-	R	读作 "0".												
11-8	CMPCNT0[3:0]	R/W	决定结果比较计数 0: 每个比较后 1: 两比较后 . . 15: 16 比较后												
7	CMP0EN	R/W	监视器 功能 0 使能 0:禁止 1:使能												
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	选择 AD 转换结果寄存器比较 ADxCMP0. <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													

16.4.9 ADxCMPCR1(监视器 设置 寄存器 1)

固定转换结果后, 中断信号 (INTADxCPB) 生成.

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	CMPCNT1			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CMP1EN	-	-	ADBIG1	REGS1			
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能												
31-12	-	R	读作 "0".												
11-8	CMPCNT1[3:0]	R/W	决定结果比较计数 0: 每个比较后 1: 两 比较后 . . 15: 16 比较后												
7	CMP1EN	R/W	监视器 功能 1 使能 0:禁止 1:使能												
6-5	-	R	读作 "0".												
4	ADBIG1	R/W	比较 条件 0:大于等于比较寄存器 1:小于或等于比较寄存器												
3-0	REGS1[3:0]	R/W	选择 AD 转换结果寄存器比较 ADxCMP1. <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													

16.4.10 ADxCMP0(转换 结果 比较 寄存器 0)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	AD0CMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	AD0CMP0				-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	AD0CMP0[11:0]	R/W	比较 AD 转换结果的值. 指定比较 AD 转换结果的值.
3-0	-	R	读作 "0".

16.4.11 ADxCMP1(转换 结果 比较 寄存器 1)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	AD0CMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	AD0CMP1				-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	AD0CMP1[11:0]	R/W	比较 AD 转换结果的值. 指定比较 AD 转换结果的值.
3-0	-	R	读作 "0".

16.4.12 ADxREG0(转换 结果 寄存器 0)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADRO							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADRO				-	-	OVR0	ADR0RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADRO[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR0	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG0 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG0 低-命令字节时清除.
0	ADR0RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR0RF> 为设置的标志时 AD 转换结果储存在 ADxREG0 寄存器且读 ADxREG0 低-命令字节时清除.

16.4.13 ADxREG1(转换 结果 寄存器 1)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR1				-	-	OVR1	ADR1RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR1[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR1	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG1 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG1 低-命令字节时清除.
0	ADR1RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR1RF> 为设置的标志时 AD 转换结果储存在 ADxREG1 寄存器且读 ADxREG1 低-命令字节时清除.

16.4.14 ADxREG2(转换 结果 寄存器 2)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR2				-	-	OVR2	ADR2RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR2[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR2	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG2 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG2 低-命令字节时清除.
0	ADR2RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR2RF> 为设置的标志时 AD 转换结果储存在 ADxREG2 寄存器且读 ADxREG2 低-命令字节时清除.

16.4.15 ADxREG3(转换 结果 寄存器 3)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR3				-	-	OVR3	ADR3RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR3[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR3	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG3 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG3 低-命令字节时清除.
0	ADR3RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR3RF> 为设置的标志时 AD 转换结果储存在 ADxREG3 寄存器且读 ADxREG3 低-命令字节时清除.

16.4.16 ADxREG4(转换 结果 寄存器 4)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR4							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR4				-	-	OVR4	ADR4RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR4[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR4	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG4 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG4 低-命令字节时清除.
0	ADR4RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR4RF> 为设置的标志时 AD 转换结果储存在 ADxREG4 寄存器且读 ADxREG4 低-命令字节时清除.

16.4.17 ADxREG5(转换 结果 寄存器 5)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR5							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR5				-	-	OVR5	ADR5RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR5[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR5	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG5 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG5 低-命令字节时清除.
0	ADR5RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR5RF> 为设置的标志时 AD 转换结果储存在 ADxREG5 寄存器且读 ADxREG5 低-命令字节时清除.

16.4.18 ADxREG6(转换 结果 寄存器 6)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR6							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR6				-	-	OVR6	ADR6RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR6[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR6	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG6 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG6 低-命令字节时清除.
0	ADR6RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR6RF> 为设置的标志时 AD 转换结果储存在 ADxREG6 寄存器且读 ADxREG6 低-命令字节时清除.

16.4.19 ADxREG7(转换 结果 寄存器 7)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR7							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR7				-	-	OVR7	ADR7RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR7[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR7	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG7 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG7 低-命令字节时清除.
0	ADR7RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR7RF> 为设置的标志时 AD 转换结果储存在 ADxREG7 寄存器且读 ADxREG7 低-命令字节时清除.

16.4.20 ADxREG8(转换 结果 寄存器 8)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR8							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR8				-	-	OVR8	ADR8RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR8[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR8	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG8 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG8 低-命令字节时清除.
0	ADR8RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR8RF> 为设置的标志时 AD 转换结果储存在 ADxREG8 寄存器且读 ADxREG8 低-命令字节时清除.

16.4.21 ADxREG9(转换 结果 寄存器 9)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR9							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR9				-	-	OVR9	ADR9RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR9[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR9	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG9 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG9 低-命令字节时清除.
0	ADR9RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR9RF> 为设置的标志时 AD 转换结果储存在 ADxREG9 寄存器且读 ADxREG9 低-命令字节时清除.

16.4.22 ADxREG10(转换 结果 寄存器 10)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR10							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR10				-	-	OVR10	ADR10RF
复位后	0	0	0	0	0	0	0	0

位	位 符	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	OVR10	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG10 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG10 低-命令字节时清除.
0	ADR10RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR10RF> 为设置的标志时 AD 转换结果储存在 ADxREG10 寄存器且读 ADxREG10 低-命令字节时清除.

16.4.23 ADxREG11(转换 结果 寄存器 11)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ADR11							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ADR11				-	-	OVR11	ADR11RF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-4	ADR11[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0".
1	OVR11	R	过运行 标志 0:无过运行 发生 1:过运行 发生 读 ADxREG11 值前新 AD 转换结果储存时, 设置此标志。且读 ADxREG11 低-命令字节时清除.
0	ADR11RF	R	AD 转换结果储存标志 0:无结果储存 1:结果储存 <ADR11RF> 为设置的标志时 AD 转换结果储存在 ADxREG11 寄存器且读 ADxREG11 低-命令字节时清除.

16.4.24 PMD 触发器 程序 寄存器

AD 转换可以通过从 PMD (程序可编程电机驱动器) 触发器开始。

PMD 触发器程序寄存器被用来指定开始程序通过由 PMD 生成的各十二个触发器,选择在完成程序时生成的中断且选择被用的 AIN 输入。

PMD 触发器程序寄存器包括三个类型寄存器。

- PMD 触发器程序数选择寄存器 (ADxPSEL0 到 ADxPSEL11)

PMD 触发器程序数选择寄存器 (ADxPSELn)通过 PMD 生成的对应十二个触发器 (PMD0TRG0 到 5, PMD1TRG0 到 5)的各十二个 AD 转换开始信号指定开始的程序. 程序 0 到 5 可用。

"ADxPSEL0 到 ADxPSEL5" 对应"PMD0TRG0 到 5". "ADxPSEL6 到 ADxPSEL11"对应 "PMD1TRG0 到 5".

- PMD 触发器中断选择寄存器 (ADxPINTS0 到 ADxPINTS5)

PMD 触发器中断选择寄存器 (ADxPINTS0 到 ADxPINTS5)在完成各程序, 和使能或禁止中断时选择生成的中断。

ADxPINTS0 对应程序 0, 且存在 ADxPINT5 (程序 5)。

- PMD 触发器程序寄存器 (ADxPSET0 到 ADxPSET5)

PMD 触发器程序设置寄存器 (ADxPSET0 到 ADxPSET5) 指定各程序 0 到 5 设置. 各 PMD 触发器程序寄存器由四个指定 AIN 转换输入的寄存器组成. 转换结果对应 ADxPSETn0 到 ADxPSETn3 寄存器储存在转换结果寄存器 0 到 3 (ADxREG0 到 ADxREG3)。

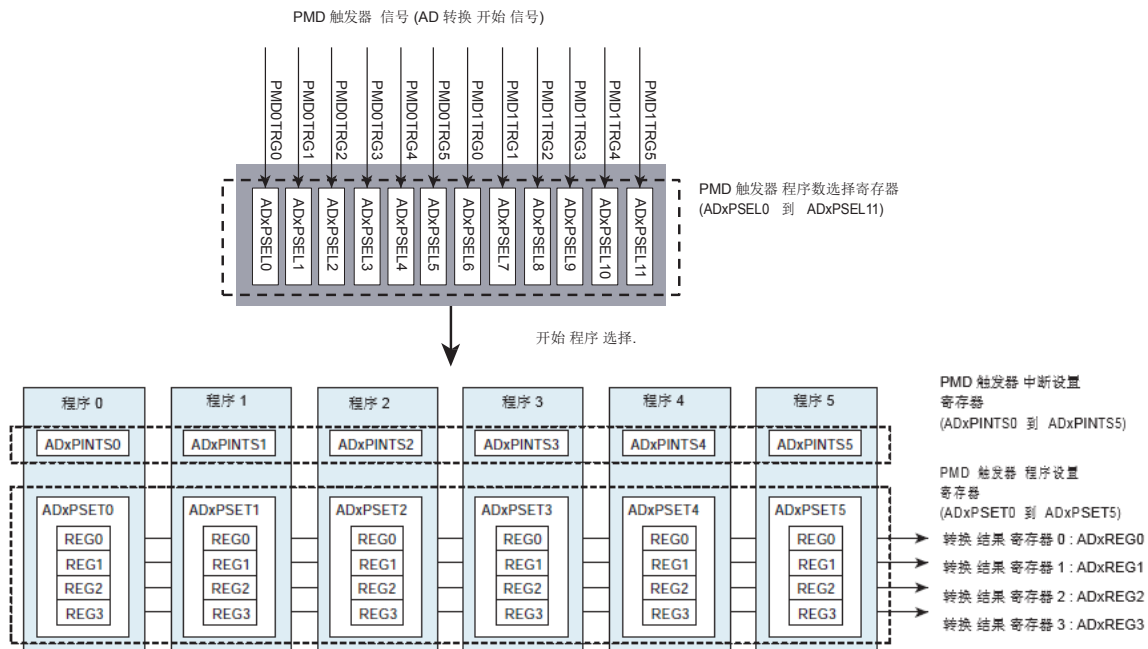


图 16-2 PMD 触发器程序 寄存器

16.4.24.1 ADxPSEL0 到 ADxPSEL11(PMD 触发器 程序 数 选择 寄存器 0 到 11)

ADxPSEL0:PMD 触发器程序 数选择 寄存器 0

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS0	-	-	-	-	PMDS0		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS0	R/W	PMD0TRG0 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS0[2:0]	R/W	程序数 选择 (参考表 16-1)

ADxPSEL1:PMD 触发器程序数选择 寄存器 1

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS1	-	-	-	-	PMDS1		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS1	R/W	PMD0TRG1 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS1[2:0]	R/W	程序数 选择 (参考表 16-1)

ADxPSEL2:PMD 触发器程序数选择 寄存器 2

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS2	-	-	-	-	PMDS2		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS2	R/W	PMD0TRG2 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS2[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL3:PMD 触发器程序数选择 寄存器 3

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS3	-	-	-	-	PMDS3		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS3	R/W	PMD0TRG3 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS3[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL4:PMD 触发器程序数选择 寄存器 4

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	PENS4	-	-	-	-	PMDS4		
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-8	-	R	读作 "0".
7	PENS4	R/W	PMD0TRG4 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS4[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL5:PMD 触发器 程序 数 选择 寄存器 5

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	PENS5	-	-	-	-	PMDS5		
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-8	-	R	读作 "0".
7	PENS5	R/W	PMD0TRG5 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS5[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL6:PMD 触发器程序数选择 寄存器 6

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS6	-	-	-	-	PMDS6		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS6	R/W	PMD1TRG0 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS6[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL7:PMD 触发器程序数选择 寄存器 7

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS7	-	-	-	-	PMDS7		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS7	R/W	PMD1TRG1 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS7[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL8:PMD 触发器程序数选择 寄存器 8

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	PENS8	-	-	-	-	PMDS8		
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-8	-	R	读作 "0".
7	PENS8	R/W	PMD1TRG2 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS8[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL9:PMD 触发器程序数选择 寄存器 9

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	PENS9	-	-	-	-	PMDS9		
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-8	-	R	读作 "0".
7	PENS9	R/W	PMD1TRG3 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS9[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL10:PMD 触发器程序数选择 寄存器 10

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS10	-	-	-	-	PMDS10		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS10	R/W	PMD1TRG4 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS10[2:0]	R/W	程序数选择 (参考表 16-1)

ADxPSEL11:PMD 触发器程序数选择 寄存器 11

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	PENS11	-	-	-	-	PMDS11		
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	PENS11	R/W	PMD1TRG5 触发器 控制 0:禁止 1:使能
6-3	-	R	读作 "0".
2-0	PMDS11[2:0]	R/W	程序数选择 (参考表 16-1)

表 16-1 程序数 选择

<PMDS0[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

16.4.24.2 ADxPINTS0 到 5(PMD 触发器 中断 选择 寄存器 0 到 5)

ADxPINTS0:PMD 触发器中断选择 寄存器 0

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	-	-	-	-	INTSEL0	
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-2	-	R	读作 "0".
1-0	INTSEL0[1:0]	R/W	中断 选择 00:无中断 输出 01:INTADxPDA 10:INTADxPDB 11:无中断 输出 开始中断选择给程序 0.

ADxPINTS1:PMD 触发器中断选择 寄存器 1

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	-	-	-	-	INTSEL1	
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-2	-	R	读作 "0".
1-0	INTSEL1[1:0]	R/W	中断 选择 00:无中断 输出 01:INTADxPDA 10:INTADxPDB 11:无中断 输出 开始中断选择给程序 1.

ADxPINTS2:PMD 触发器中断选择 寄存器 2

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	-	-	-	-	INTSEL2	
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-2	-	R	读作 "0".
1-0	INTSEL2[1:0]	R/W	中断选择 00:无中断 输出 01:INTADxPDA 10:INTADxPDB 11:无中断 输出 开始中断选择给程序 2.

ADxPINTS3:PMD 触发器中断选择 寄存器 3

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	-	-	-	-	INTSEL3	
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-2	-	R	读作 "0".
1-0	INTSEL3[1:0]	R/W	中断选择 00:无中断 输出 01:INTADxPDA 10:INTADxPDB 11:无中断 输出 开始中断选择给程序 3.

ADxPINTS4:PMD 触发器中断选择 寄存器 4

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	-	-	-	-	INTSEL4	
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-2	-	R	读作 "0".
1-0	INTSEL4[1:0]	R/W	中断 选择 00:无中断 输出 01:INTADxPDA 10:INTADxPDB 11:无中断 输出 开始中断选择给程序 4.

ADxPINTS5:PMD 触发器中断选择 寄存器 5

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	-	-	-	-	INTSEL5	
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-2	-	R	读作 "0".
1-0	INTSEL5[1:0]	R/W	中断 选择 00:无中断 输出 01:INTADxPDA 10:INTADxPDB 11:无中断 输出 开始中断选择给程序 5.

16.4.24.3 ADxPSET0 到 5(PMD 触发器 程序 寄存器 0 到 5)

各 ADxPSETn (n=0 到 5:程序 数) 由四个估计 <AINSPnm [4:0]>, <UVWISnm[1:0]>, 和 <ENSPnm> 一对(m=0 到 3) 设置组成.

设置 <ENSPnm> 为 "1" 使能 <UVWISnm[1:0]>, <AINSPnm[4:0]> 位被用来选择被用的 AIN 引脚. 用这些条件, AD 转换开始且储存到转换结果寄存器.

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 16-2 选择 AIN 引脚

<AINSP00 [4:0]> 到 <AINSP53 [4:0]>	ADC 单元 A	ADC 单元 B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:保留
0_1100 到 1_1111	:保留	

ADxPSET0:PMD 触发器程序 寄存器 0

	31	30	29	28	27	26	25	24
位 符	ENSP03	UVWIS03			AINSP03			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSP02	UVWIS02			AINSP02			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSP01	UVWIS01			AINSP01			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSP00	UVWIS00			AINSP00			
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSP03	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	UVWIS03[1:0]	R/W	相 选择 (向量 引擎用) 见 下表.
28-24	AINSP03[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
23	ENSP02	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	UVWIS02[1:0]	R/W	相 选择 (向量 引擎用) 见 下表.
20-16	AINSP02[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
15	ENSP01	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	UVWIS01[1:0]	R/W	相 选择 (向量 引擎用) 见 下表.
12-8	AINSP01[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
7	ENSP00	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	UVWIS00[1:0]	R/W	相 选择 (向量 引擎用) 见 下表.
4-0	AINSP00[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".

相选择

00	不被指定
01	U
10	V
11	W

ADxPSET1:PMD 触发器程序 寄存器 1

	31	30	29	28	27	26	25	24
位符	ENSP13	UVWIS13			AINSP13			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	ENSP12	UVWIS12			AINSP12			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	ENSP11	UVWIS11			AINSP11			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ENSP10	UVWIS10			AINSP10			
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31	ENSP13	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	UVWIS13[1:0]	R/W	相选择 (向量引擎用) 见下表 .
28-24	AINSP13[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
23	ENSP12	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	UVWIS12[1:0]	R/W	相选择 (向量引擎用) 见下表 .
20-16	AINSP12[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
15	ENSP11	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	UVWIS11[1:0]	R/W	相选择 (向量引擎用) 见下表 .
12-8	AINSP11[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
7	ENSP10	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	UVWIS10[1:0]	R/W	相选择 (向量引擎用) 见下表 .
4-0	AINSP10[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".

相选择

00	不被指定
01	U
10	V
11	W

ADxPSET2:PMD 触发器程序 寄存器 2

	31	30	29	28	27	26	25	24
位 符	ENSP23	UVWIS23			AINSP23			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSP22	UVWIS22			AINSP22			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSP21	UVWIS21			AINSP21			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSP20	UVWIS20			AINSP20			
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSP23	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	UVWIS23[1:0]	R/W	相 选择 (向量 引擎用) 见下表 .
28-24	AINSP23[4:0]	R/W	AIN 选择 参考"表 16-2 选择 AIN 引脚".
23	ENSP22	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	UVWIS22[1:0]	R/W	相 选择 (向量 引擎用) 见下表.
20-16	AINSP22[4:0]	R/W	AIN 选择 参考"表 16-2 选择 AIN 引脚".
15	ENSP21	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	UVWIS21[1:0]	R/W	相 选择 (向量 引擎用) 见下表 .
12-8	AINSP21[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
7	ENSP20	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	UVWIS20[1:0]	R/W	相 选择 (向量 引擎用) 见下表.
4-0	AINSP20[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".

相选择

00	不被指定
01	U
10	V
11	W

ADxPSET3:PMD 触发器程序 寄存器 3

	31	30	29	28	27	26	25	24
位符	ENSP33	UVWIS33			AINSP33			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	ENSP32	UVWIS32			AINSP32			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	ENSP31	UVWIS31			AINSP31			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ENSP30	UVWIS30			AINSP30			
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31	ENSP33	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	UVWIS33[1:0]	R/W	相选择 (向量引擎用) 见下表.
28-24	AINSP33[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
23	ENSP32	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	UVWIS32[1:0]	R/W	相选择 (向量引擎用) 见下表.
20-16	AINSP32[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
15	ENSP31	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	UVWIS31[1:0]	R/W	相选择 (向量引擎用) 见下表.
12-8	AINSP31[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
7	ENSP30	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	UVWIS30[1:0]	R/W	相选择 (向量引擎用)
4-0	AINSP30[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".

相选择

00	不被指定
01	U
10	V
11	W

ADxPSET4:PMD 触发器 程序 寄存器 4

	31	30	29	28	27	26	25	24
位 符	ENSP43	UVWIS43			AINSP43			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSP42	UVWIS42			AINSP42			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSP41	UVWIS41			AINSP41			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSP40	UVWIS40			AINSP40			
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSP43	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	UVWIS43[1:0]	R/W	相 选择 (向量 引擎用) 见下表.
28-24	AINSP43[4:0]	R/W	AIN 选择 参考"表 16-2 选择 AIN 引脚".
23	ENSP42	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	UVWIS42[1:0]	R/W	相 选择 (向量 引擎用) 见下表.
20-16	AINSP42[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".
15	ENSP41	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	UVWIS41[1:0]	R/W	相 选择 (向量 引擎用) 见下表.
12-8	AINSP41[4:0]	R/W	AIN 选择 参考"表 16-2 选择 AIN 引脚".
7	ENSP40	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	UVWIS40[1:0]	R/W	相 选择 (向量 引擎用) 见下表.
4-0	AINSP40[4:0]	R/W	AIN 选择 参考"表 16-2 选择 AIN 引脚".

相选择

00	不被指定
01	U
10	V
11	W

ADxPSET5:PMD 触发器程序 寄存器 5

	31	30	29	28	27	26	25	24
位符	ENSP53	UVWIS53			AINSP53			
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	ENSP52	UVWIS52			AINSP52			
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	ENSP51	UVWIS51			AINSP51			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ENSP50	UVWIS50			AINSP50			
复位后	0	0	0	0	0	0	0	0
位	位符	类型	功能					
31	ENSP53	R/W	ADxREG3 使能 0:禁止 1:使能					
30-29	UVWIS53[1:0]	R/W	相选择 (向量引擎用) 见下表.					
28-24	AINSP53[4:0]	R/W	AIN 选择 参考"表 16-2 选择 AIN 引脚".					
23	ENSP52	R/W	ADxREG2 使能 0:禁止 1:使能					
22-21	UVWIS52[1:0]	R/W	相选择 (向量引擎用) 见下表.					
20-16	AINSP52[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".					
15	ENSP51	R/W	ADxREG1 使能 0:禁止 1:使能					
14-13	UVWIS51[1:0]	R/W	相选择 (向量引擎用) 见下表.					
12-8	AINSP51[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".					
7	ENSP50	R/W	ADxREG0 使能 0:禁止 1:使能					
6-5	UVWIS50[1:0]	R/W	相选择 (向量引擎用) 见下表.					
4-0	AINSP50[4:0]	R/W	AIN 选择 参考 "表 16-2 选择 AIN 引脚".					

相选择

00	不被指定
01	U
10	V
11	W

16.4.25 ADxTSET03 / ADxTSET47 / ADxTSET811 (定时器 触发器 程序 寄存器)

AD 转换可以通过从 TMRBx 生成 INTTBx0 为触发器开始. 有十二个 8-位 寄存器编程定时器触发器. 设置 <ENSTm> 为"1" 使能 ADxTSETm 寄存器.

<AINSTm[4:0]>被用来选择被用的 AIN 引脚. 定时器触发器程序寄存器数对应 AD 转换结果寄存器. 此 AD 转换结束时, 中断 : INTADxTMR 生成.

(m=0 到 11)

表 16-3 选择 AIN 引脚

<AINST0 [4:0]> 到 <AINST11 [4:0]>	ADC 单元 A	ADC 单元 B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:保留
0_1100 到 1_1111	:保留	

ADxTSET03: 定时器触发器程序 寄存器 03

	31	30	29	28	27	26	25	24
位符	ENST3	-	-	AINST3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	ENST2	-	-	AINST2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	ENST1	-	-	AINST1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ENST0	-	-	AINST0				
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31	ENST3	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINST3[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
23	ENST2	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINST2[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
15	ENST1	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINST1[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
7	ENST0	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINST0[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".

ADxTSET47: 定时器 触发器 程序 寄存器 47

	31	30	29	28	27	26	25	24
位 符	ENST7	-	-	AINST7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENST6	-	-	AINST6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENST5	-	-	AINST5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENST4	-	-	AINST4				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENST7	R/W	ADxREG7 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINST7[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
23	ENST6	R/W	ADxREG6 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINST6[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
15	ENST5	R/W	ADxREG5 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINST5[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
7	ENST4	R/W	ADxREG4 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINST4[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".

ADxTSET811: 定时器触发器程序 寄存器 811

	31	30	29	28	27	26	25	24
位 符	ENST11	-	-	AINST11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENST10	-	-	AINST10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENST9	-	-	AINST9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENST8	-	-	AINST8				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENST11	R/W	ADxREG11 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINST11[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
23	ENST10	R/W	ADxREG10 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINST10[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
15	ENST9	R/W	ADxREG9 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINST9[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".
7	ENST8	R/W	ADxREG8 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINST8[4:0]	R/W	AIN 选择 参考 "表 16-3 选择 AIN 引脚".

16.4.26 ADxSSET03 / ADxSSET47 / ADxSSET811 (软件 触发器 程序 寄存器)

AD 转换可以通过软件开始. 有十二个 8-位编程软件触发器寄存器. 设置 <ENSSm>为"1" 使能 ADxSSETm 寄存器. <AINSSm[4:0]>被用来选择被用的 AIN 引脚. 软件触发器程序寄存器数对应转换结果寄存器. 此 AD 转换结束时, 中断 :INTADxSFT 生成.

(m=0 到 11)

表 16-4 选择 AIN 引脚

<AINSS0 [4:0]> 到 <AINSS11 [4:0]>	ADC 单元 A	ADC 单元 B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:保留
0_1100 到 1_1111	:保留	

ADxSSET03: 软件触发器程序 寄存器 03

	31	30	29	28	27	26	25	24
位 符	ENSS3	-	-	AINSS3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSS2	-	-	AINSS2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSS1	-	-	AINSS1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSS0	-	-	AINSS0				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSS3	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINSS3[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
23	ENSS2	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINSS2[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
15	ENSS1	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINSS1[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
7	ENSS0	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINSS0[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".

ADxSSET47: 软件触发器程序 寄存器 47

	31	30	29	28	27	26	25	24
位 符	ENSS7	-	-	AINSS7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSS6	-	-	AINSS6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSS5	-	-	AINSS5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSS4	-	-	AINSS4				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSS7	R/W	ADxREG7 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINSS7[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
23	ENSS6	R/W	ADxREG6 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINSS6[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
15	ENSS5	R/W	ADxREG5 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINSS5[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
7	ENSS4	R/W	ADxREG4 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINSS4[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".

ADxSSET811: 软件触发器程序 寄存器 811

	31	30	29	28	27	26	25	24
位符	ENSS11	-	-	AINSS11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	ENSS10	-	-	AINSS10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	ENSS9	-	-	AINSS9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ENSS8	-	-	AINSS8				
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31	ENSS11	R/W	ADxREG11 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINSS11[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
23	ENSS10	R/W	ADxREG10 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINSS10[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
15	ENSS9	R/W	ADxREG9 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINSS9[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".
7	ENSS8	R/W	ADxREG8 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINSS8[4:0]	R/W	AIN 选择 参考 "表 16-4 选择 AIN 引脚".

16.4.27 ADxASET03 / ADxASET47 / ADxASET811(恒定 转换 程序 寄存器)

ADCs 允许转换触发器恒定使能. 有十二个 8-位编程恒定触发器寄存器. 设置 <ENSAm>为"1" 使能 ADxASETm 寄存器. <AINSA_m[4:0]> 被用来选择被用的 AIN 引脚. 恒定触发器程序寄存器数对应转换结果寄存器.

(m=0 到 11)

表 16-5 选择 AIN 引脚

<AINSA0 [4:0]> 到 <AINSA11 [4:0]>	ADC 单元 A	ADC 单元 B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:保留
0_1100 到 1_1111	:保留	

ADxASET03: 恒定转换程序 寄存器 03

	31	30	29	28	27	26	25	24
位 符	ENSA3	-	-	AINSA3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSA2	-	-	AINSA2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSA1	-	-	AINSA1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSA0	-	-	AINSA0				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSA3	R/W	ADxREG3 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINSA3[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".
23	ENSA2	R/W	ADxREG2 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINSA2[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".
15	ENSA1	R/W	ADxREG1 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINSA1[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".
7	ENSA0	R/W	ADxREG0 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINSA0[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".

ADxASET47: 恒定转换程序 寄存器 47

	31	30	29	28	27	26	25	24
位 符	ENSA7	-	-	AINSA7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSA6	-	-	AINSA6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSA5	-	-	AINSA5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSA4	-	-	AINSA4				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSA7	R/W	ADxREG7 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINSA7[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".
23	ENSA6	R/W	ADxREG6 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINSA6[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".
15	ENSA5	R/W	ADxREG5 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINSA5[4:0]	R/W	AIN 选择 参考"表 16-5 选择 AIN 引脚".
7	ENSA4	R/W	ADxREG4 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINSA4[4:0]	R/W	AIN 选择 参考 "表 16-5 选择 AIN 引脚".

ADxASET811: 恒定转换程序 寄存器 811

	31	30	29	28	27	26	25	24
位 符	ENSA11	-	-	AINSA11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	ENSA10	-	-	AINSA10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	ENSA9	-	-	AINSA9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	ENSA8	-	-	AINSA8				
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31	ENSA11	R/W	ADxREG11 使能 0:禁止 1:使能
30-29	-	R	读作 "0".
28-24	AINSA11[4:0]	R/W	AIN 选择 参考 "表 16-5 选择 AIN 引脚".
23	ENSA10	R/W	ADxREG10 使能 0:禁止 1:使能
22-21	-	R	读作 "0".
20-16	AINSA10[4:0]	R/W	AIN 选择 参考 "表 16-5 选择 AIN 引脚".
15	ENSA9	R/W	ADxREG9 使能 0:禁止 1:使能
14-13	-	R	读作 "0".
12-8	AINSA9[4:0]	R/W	AIN 选择 参考 "表 16-5 选择 AIN 引脚".
7	ENSA8	R/W	ADxREG8 使能 0:禁止 1:使能
6-5	-	R	读作 "0".
4-0	AINSA8[4:0]	R/W	AIN 选择 参考 "表 16-5 选择 AIN 引脚".

16.5 工作描述

16.5.1 模拟参考电压

关于高-电平和低-电平模拟参考电压，VREFHA 和 VREFLA 引脚被用在 ADC A 且 VREFHB 和 VREFLB 引脚被用在 ADC B。ADxMOD3<RCUT>设置为 "1"时，VREFHx-VREFLx 切换从条件切换关闭。

16.5.2 开始 AD 转换

AD 转换通过软件或以下三个触发器信号的一个开始。

- PMD 触发器 (见 "16.4.24 PMD 触发器 程序 寄存器")
- 定时器触发器 (TMRB) (见 "16.4.25 定时器 触发器 程序 寄存器.")
- 软件 触发器 (见 "16.4.26 软件 触发器 程序 寄存器.")

这些开始触发器 给出所示如下优先级。

PMD 触发器 0 > ... > PMD 触发器 5 > 定时器触发器 > 软件触发器 > 恒定
触发器

如果 PMD 触发器发生在 AD 转换处理时，PMD 触发器处理停止处理中程序且开始对应 PMD 触发器数的 AD 转换。

如果高优先级触发器发生在 AD 转换处理时，处理中程序完成后高优先级触发器被处理。

从触发器生成到 AD 转换开始有延迟。延迟取决于触发器。以下时序图和表表示延迟。

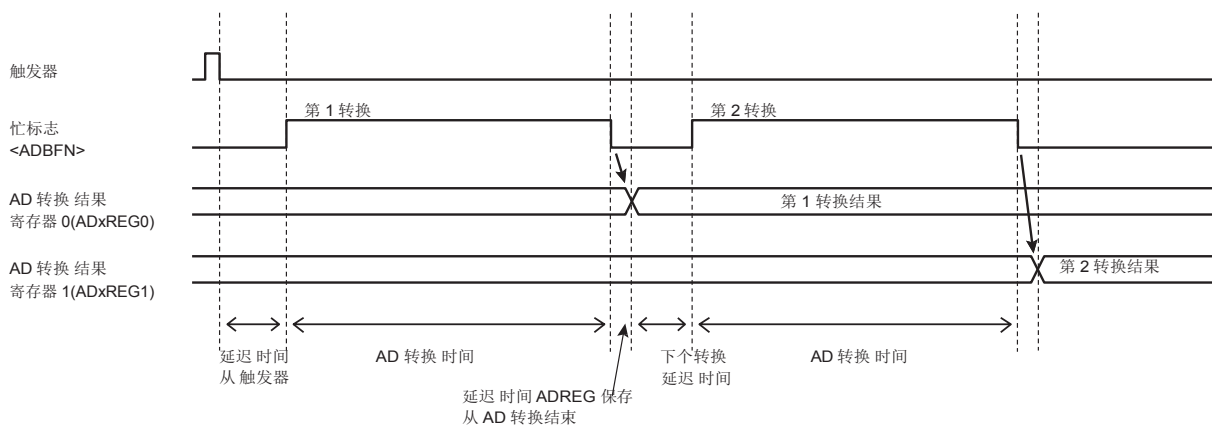


图 16-3 AD 转换时序图

表 16-6 AD 转换 时间

触发器	延迟 时间从触发器 (注 1)		转换 时间	延迟 时间 ADREG 保存 从 AD 转换结束	到下一个 转换延迟 时间(注 2)	
	Min	Max			Min	Max
PMD [SCLK]	5	8	120	2	3	5
TMRB [SCLK]		16				13
软件 [SCLK]	6	17				
恒定 [SCLK]						

注 1: 延迟时间从触发器到转换开始

注 2: 延迟时间从 ADREG 保存到第 2 转换开始和 大于 1 转换开始,用理想触发器

16.5.3 AD 转换 监视器 功能

ADCs 有 AD 转换监视器功能. 此功能使能时,转换结果匹配被指定的比较值时中断生成.

要使能监视器功能, 设置 ADxCMPCR0<CMP0EN> 或 ADxCMPCR1<CMP1EN> 为 "1". 监视器功能中, 如果监视器功能被分配的AD 转换结果寄存器值对应由 ADxCMCR0<ADBIG0>/ADxCMCR1<ADBIG1>指定的比较条件, 中断 (INTADxCPA 是监视器功能 0, INTADxCPB 是监视器功能 1)生成. 在储存转换结果到寄存器的时序执行比较.

注 1: AD 转换结果储存标志 (<ADR_xRF>)不是由比较功能清除.

注 2: 比较功能与通过软件读转换结果不同.因此, 如果下一个转换完成 而不读之前的结果, 过运行标志 (<OVR_x>) 被设置.

16.6 AD 转换时序图

以下表示软件触发器转换时序图, 恒定转换和触发器接受.

16.6.1 软件 触发器 转换

软件触发器转换中, 由 ADxSSET03, ADxSSET47 和 ADxSSET811 编程的转换完成后中断生成.(图 16-4)

如果 AD 转换期间 ADxMOD1<ADEN>清除为"0", 处理中转换停止而不储存到结果寄存器.(图 16-5)

条件

软件 触发器 设置 : AINA0, AINA1, AINA2, AINA4

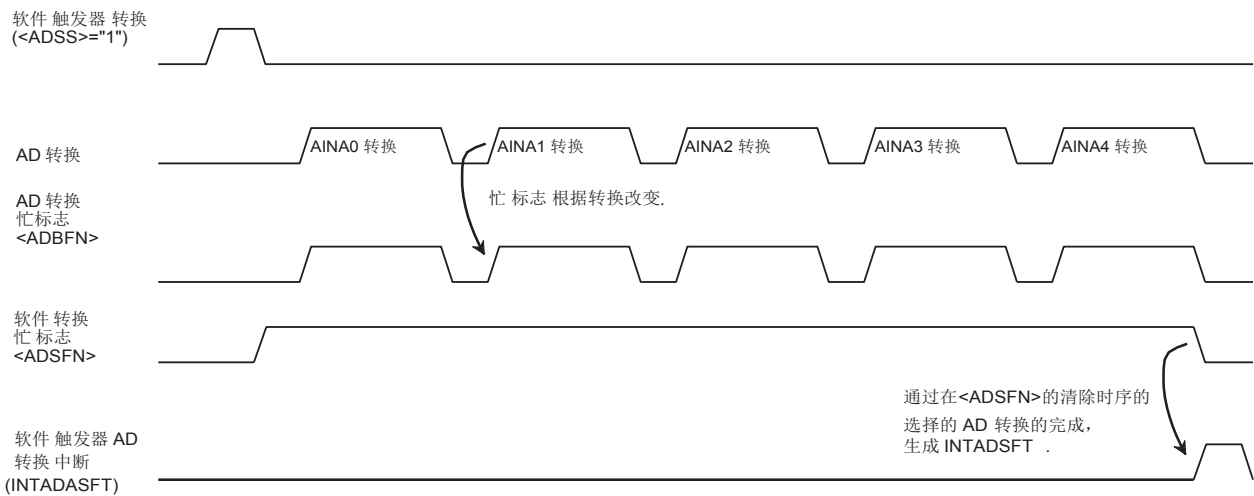


图 16-4 软件触发器 AD 转换

条件
软件触发器 设置 : AINA0, AINA1, AINA2

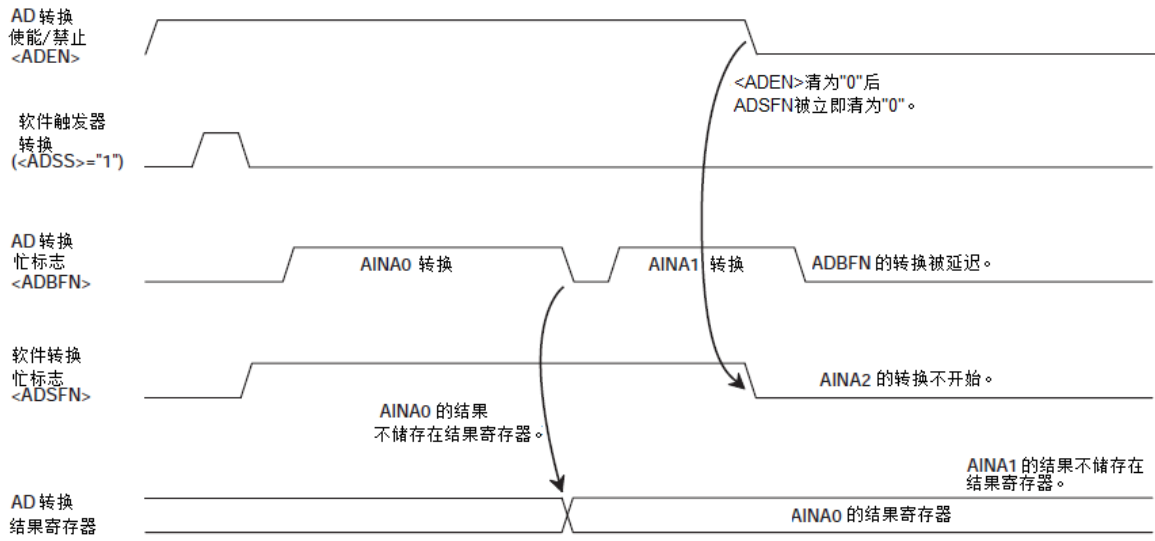


图 16-5 软件触发器 AD 转换期间写入 "0"到 <ADEN>

16.6.2 恒定 转换

恒定转换中, 如果下一个转换完成而不从转换结果寄存器读之前的结果, 过运行标志设置为"1". 此时, 变换结果寄存器中之前的转换结果通过下一个结果覆盖. 过运行标志通过读转换结果清除.(图 16-6)

条件

恒定转换 设置: AINA0

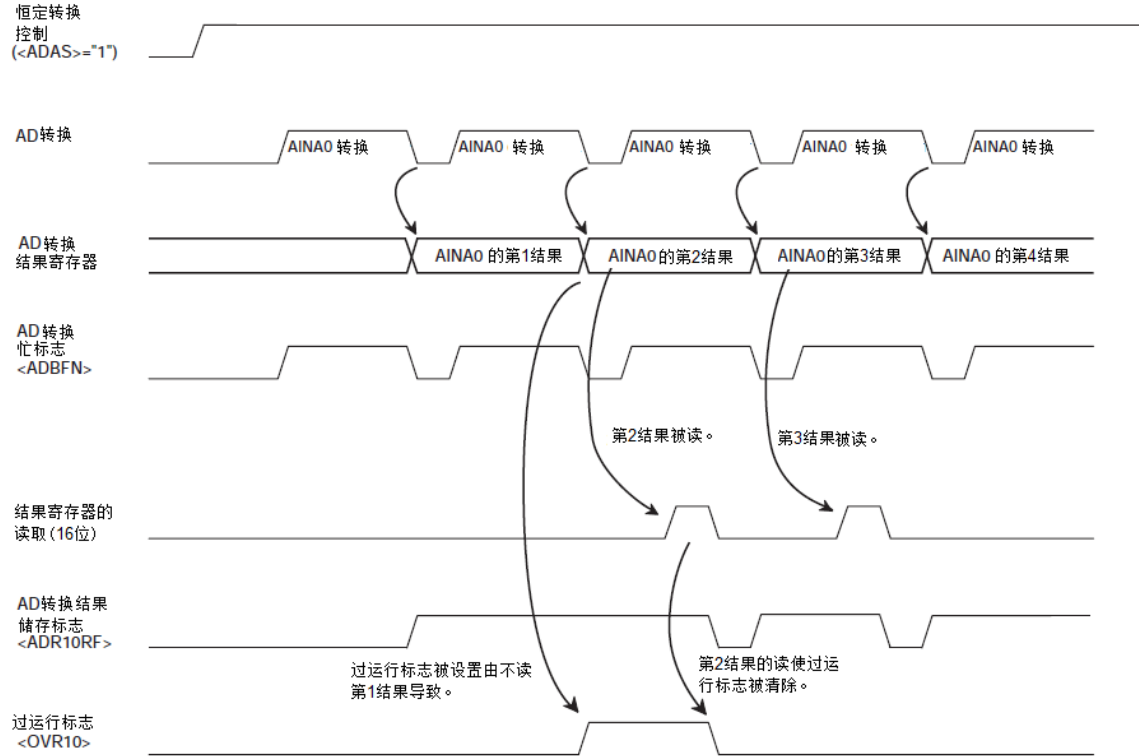


图 16-6 恒定 转换

16.6.3 通过触发器 AD 转换

软件触发器转换期间如果 PMD 触发器发生，处理中转换立即停止。(图 16-7) 软件触发器转换期间如果定时器触发器发生，处理中转换完成后处理中转换停止。(图 16-8) 通过触发器的转换完成后，软件触发器转换由 ADxSSET03, ADxSSET47 和 ADxSSET811 的编程开始处开始。(图 16-9)

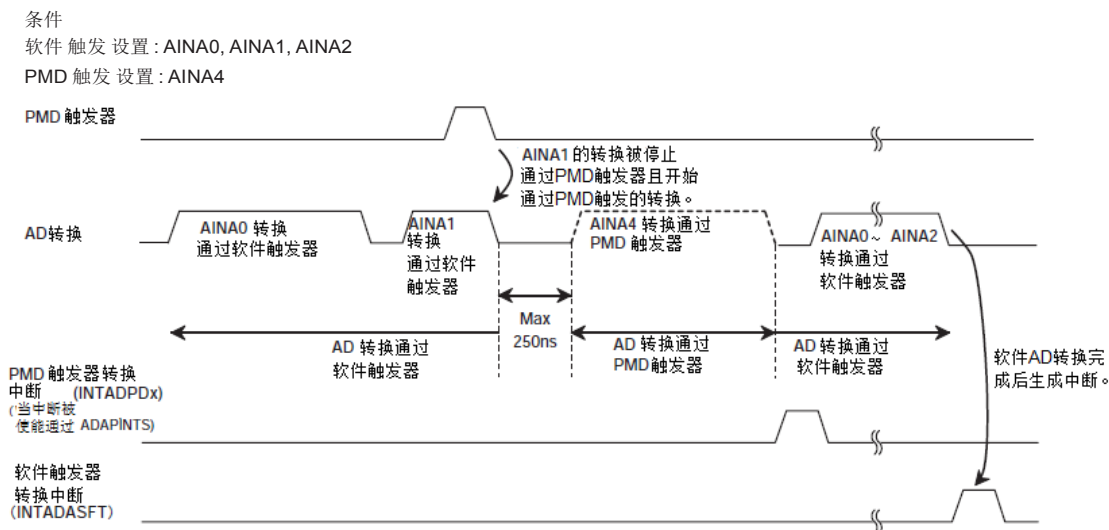


图 16-7 通过 PMD 触发 AD 转换

条件
软件触发设置: AINA0
定时器触发设置: AINA1

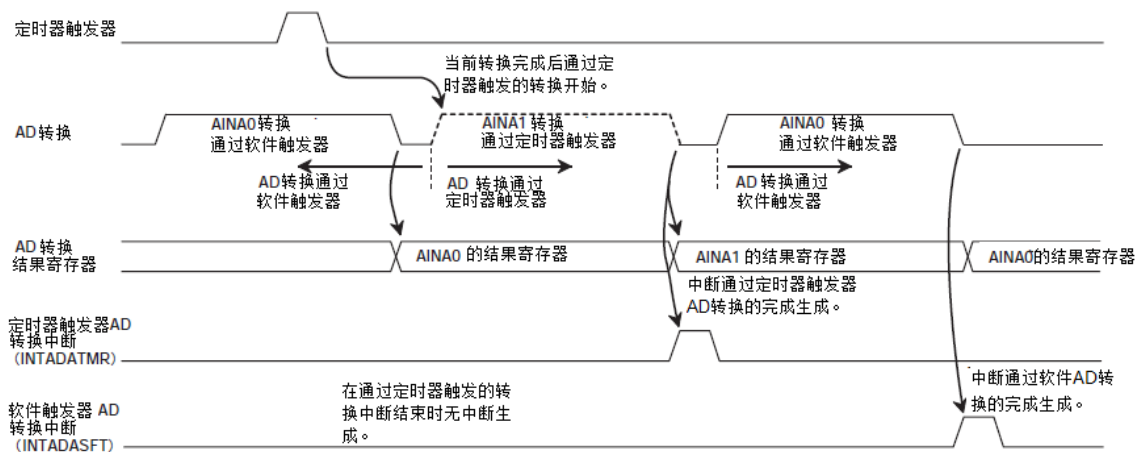


图 16-8 通过定时器触发 AD 转换(1)

16.6 AD 转换时序图

条件

软件触发 设置 : AINA0, AINA1, AINA2

定时器触发 设置 : AINA4

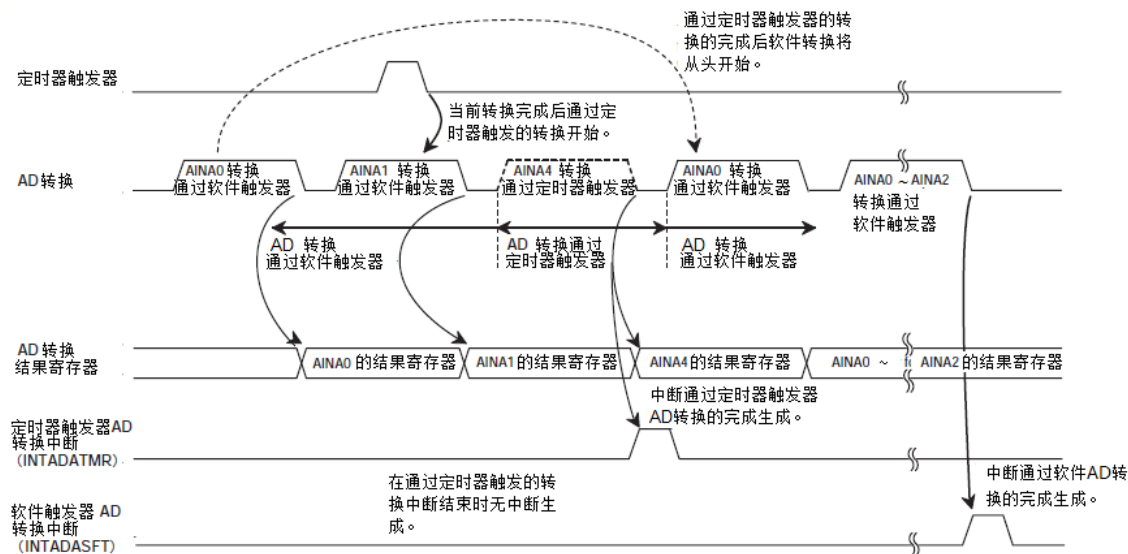
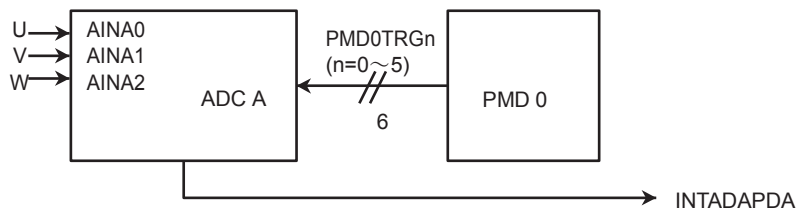


图 16-9 通过定时器触发器 AD 转换(2)

16.7 使用例

16.7.1 逐次转换 使用一个 PMD(三电阻)和一个 ADC

以下表示使用三个电阻的一个 PMD0 和一个 ADC 的 AD 转换电路图。



例 ADC 设置所示如下。

ADC 单元 A

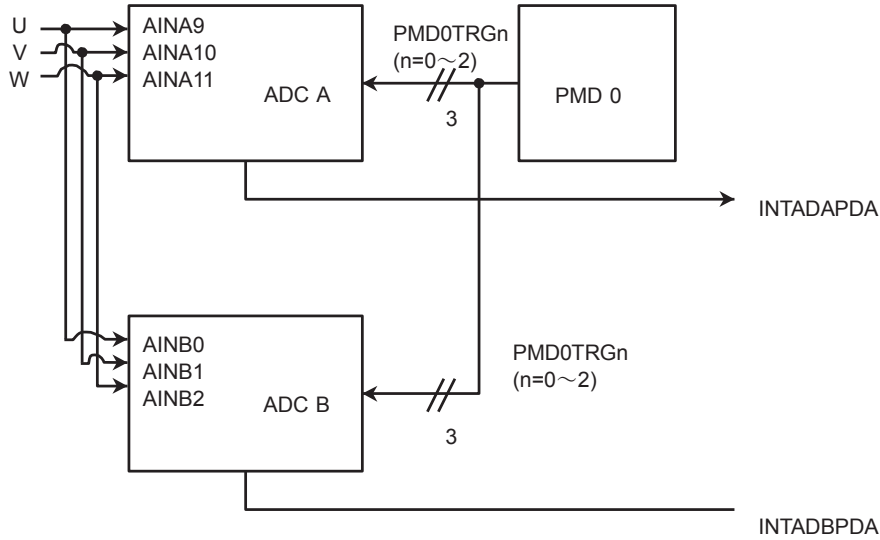
程序	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	A	A	A	A	A	A

程序 0 到 5 被分配给触发器输入 PMD0TRG0 到 5. "reg0" 和 "reg1" 指 PMD 触发器程序寄存器 ADAPSETn[7:0] 和 ADAPSETn[15:8]. "U", "V" 和 "W" 指电机的相. AIN 输入选择获取这些相.

触发器输入发生时, 逐个执行 AD 转换基于 reg0 和 reg1, 然后中断信号 (INTADAPDA)生成.

16.7.2 模拟 转换 使用一个 PMD (三电阻)和两个 ADC

以下表示使用三个电阻的 PMD0 和两个 ADCs 的 AD 转换方块图.



例 ADC 设置所示如下.

ADC 单元 A

程序	0	1	2
reg0	U	V	W
INT	A	A	A

ADC 单元 B

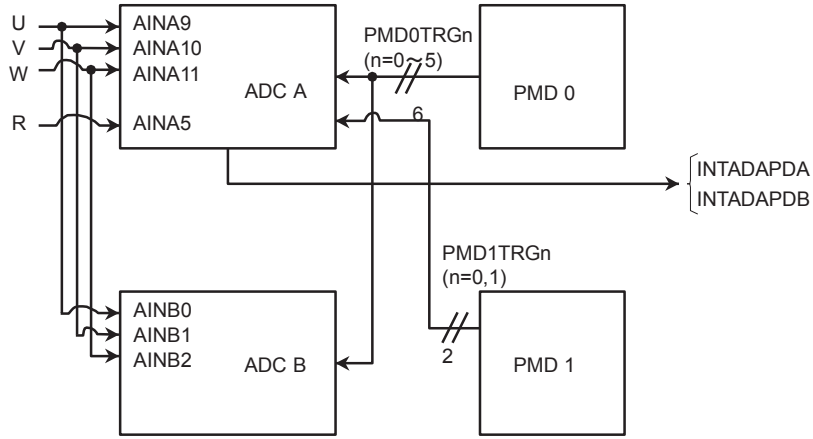
程序	0	1	2
reg0	V	W	U
INT	A	A	A

程序 0 到 2 被分配到 ADC A 和 ADC B 的三个触发器输入. "reg0" 指 PMD 触发器程序寄存器 ADAPSETn[7:0] 和 ADBPSETn[7:0]. "U", "V" 和 "W" 指电机相. AIN 输入被选择获取这些相.

触发器输入发生时, ADC A 和 ADC B 同时开始执行 AD 转换基于 reg0, 且中断信号 (INTADAPDA, INTADBPDA) 输出到 ADC A 和 ADC B.

16.7.3 模拟转换 使用 PMD0 (三电阻), PMD1 (一个 电阻)和两个 ADC

以下表示使用三个电阻的 PMD0, 一个电阻的 PMD1 和两个 ADCs的AD 转换电路图.



例 ADC 设置所示如下.

ADC 单元 A

触发器	PMD0	PMD0	PMD0	PMD1	PMD1
	0,3	1,4	2,5	6	7
程序	0	1	2	3	4
reg0	U	V	W	-	-
reg1	-	-	-	R	-
reg2	-	-	-	-	R
INT	A	A	A	-	B

ADC 单元 B

触发器	PMD0	PMD0	PMD0
	0,3	1,4	2,5
程序	0	1	2
reg0	V	W	U
INT	-	-	-

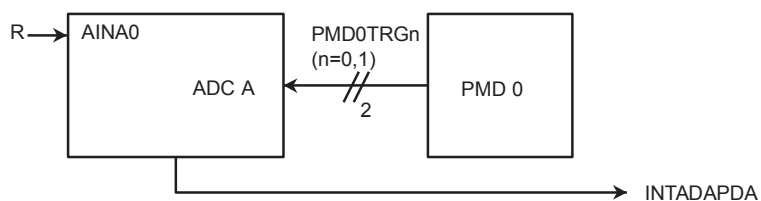
ADC A 中, 程序 0 到 2 被分配到 六个从 PMD0 的触发器信号且程序 3 和 4 被分配到从 PMD1 的两触发器信号. ADC B 中, 程序 0 到 2 被分配到从 PMD0 的六个触发器信号.

"reg0", "reg1" 和 "reg2" 指 PMD 触发器程序寄存器 ADxPSETn[7:0], ADxPSETn [15:8] 和 ADxPSETn[23:16] (x=A,B : ADC 单元). "U", "V" 和 "W" 指电机相. AIN 输入被选择获取这些相. "R" 指电阻, 在那里 AIN 被连接到电阻设置.

触发器输入发生时, ADC A 或 ADC B 开始执行 AD 转换. ADC A中, 从 PMD0的触发器中断 (INTADAPDA) 生成的和来自 PMD1 触发器的中断 (INTADAPDB)生成. ADC B中, 此例中禁止中断生成 .

16.7.4 逐次转换 使用一个 PMD (一个电阻) 和一个 ADC

以下表示使用一个电阻的 PMD0 和一个 ADC 的 AD 转换电路图。



例 ADC 设置所示如下。

ADC 单元 A

触发器	PMD0	PMD0
		0
程序	0	1
reg0	R	-
reg1	-	R
INT	-	A

程序 0 和 1 从 PMD0 被分配到两触发器信号。

"reg0" 和 "reg1" 指 PMD 触发器程序寄存器 ADAPSETn[7:0] 和 ADAPSETn[15:8]。"R" 指电阻, 在那里 AIN 被连接到电阻设置。

触发器输入发生时, ADC 逐个开始执行程序 0 和 1。程序 1 完成时, 中断 (INTADAPDA) 生成。

16.8 AD 转换器使用注意事项

AD 转换期间,不要改变端口 H/J/P 输出数据,避免影响转换结果.

AD 转换结果也许通过电源波动或环境噪声改变.

AD 转换处理时如果用 AD 输入分享的流向引脚的输入或输出被改变,或流向其他被指定为输出端口的引脚的输出电流波动,AD 转换精度会更低.

针对这些问题采取措施,通过用程序平均多次转换结果.

17. 电机控制电路 (PMD: 可编程电机驱动器)

此产品 PMD 可以控制三-相电机诸如用向量引擎(A-VE)和模拟/数字转换器 (ADC)联合的向量电机. 脉-宽调制电路, 传导控制且同步触发器生成器可以通过命令从向量引擎被激活. 同步触发器生成电路可以命令 AD 转换器开始 ADC 转换.

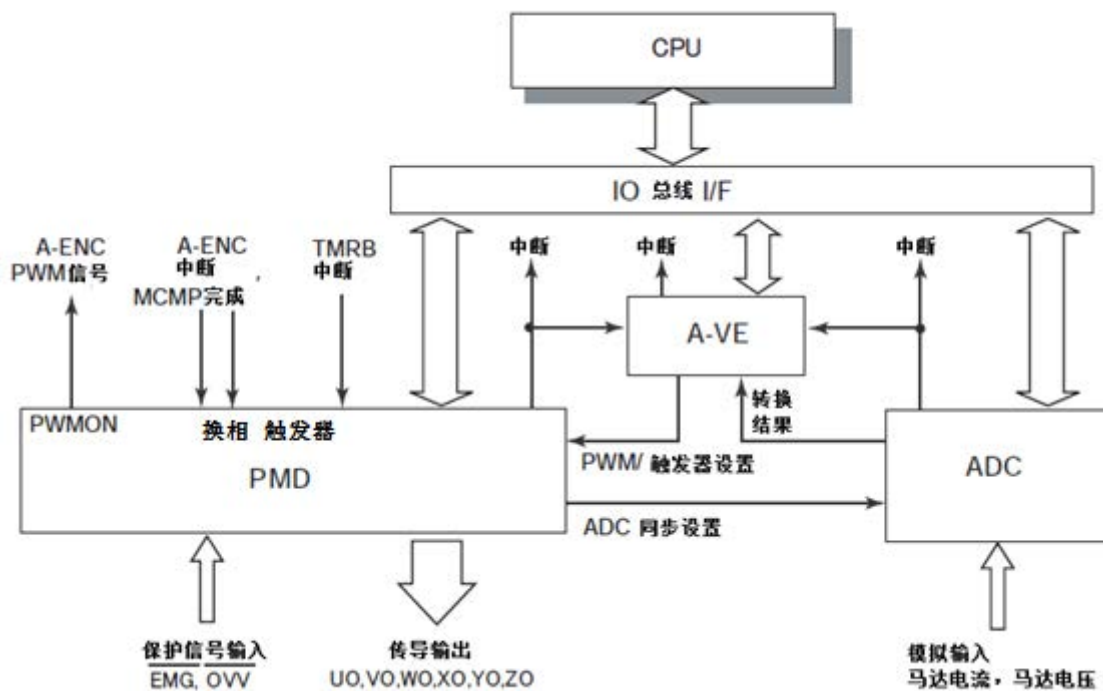


图 17-1 相关电机控制功能方块图

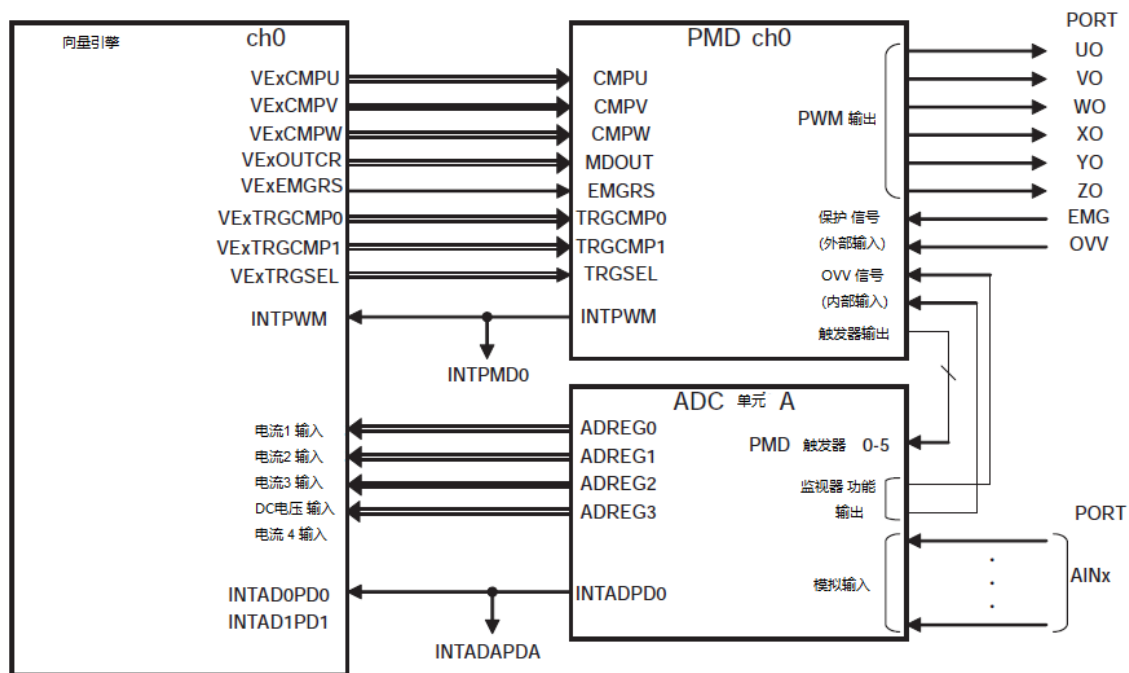


图 17-2 电机控制电路, 向量引擎和 A/D 转换器相关图

17.1 PMD 电路 配置

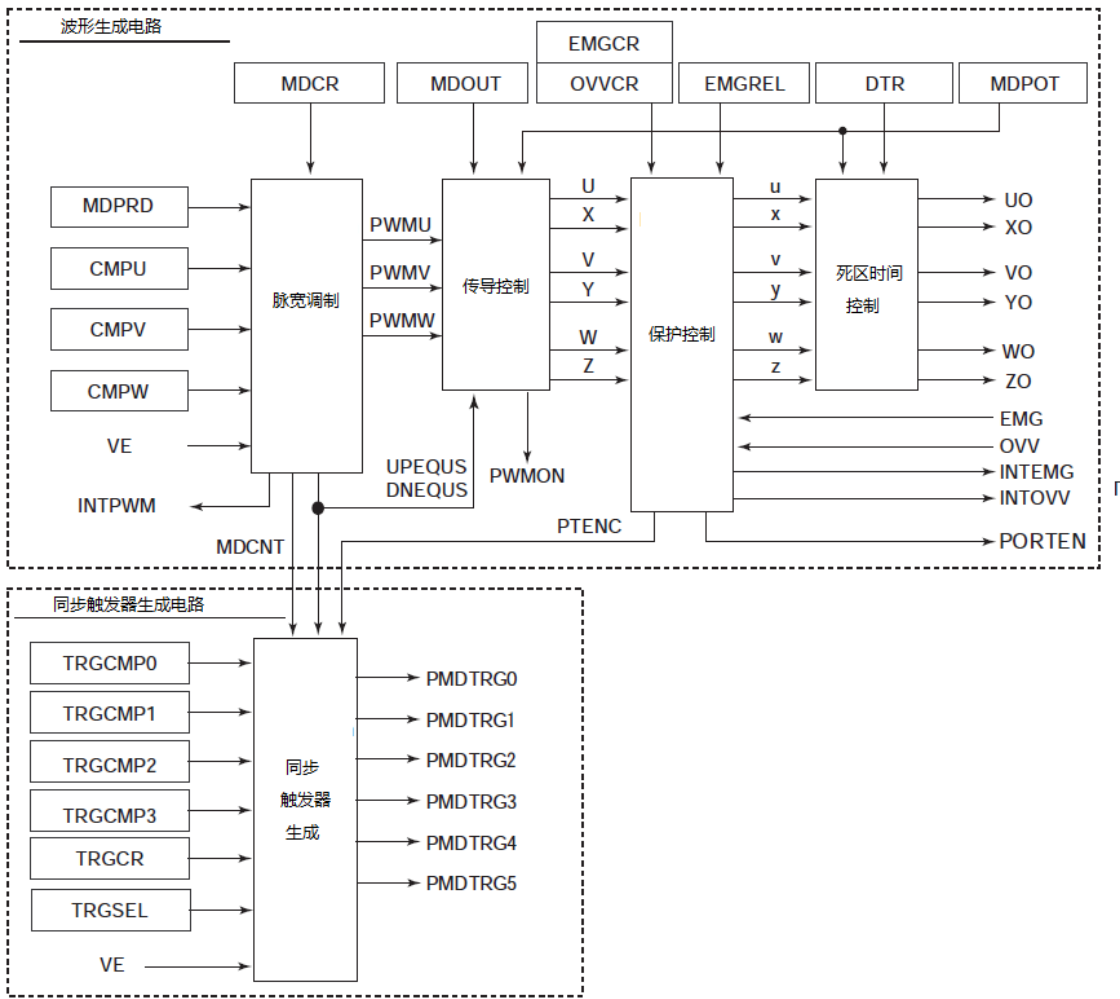


图 17-3 PMD 电路方块图

PMD 电路由波形生成电路和同步触发器生成电路两方块组成. 波形生成电路包括脉宽调制电路, 传导控制电路, 保护控制电路, 死区时间控制电路.

- 脉宽调制电路有共通 PWM 载波波形且生成独立 3-相 PWM 波形.
- 传导控制电路决定 U, V 和 W 相的各上 和低侧的输出类型 .
- 保护控制电路控制 紧急情况输出停止通过 EMG 输入和 OVV 输入.
- 死区时间控制电路防止上侧和低侧被切换时可能发生短路电路.
- 同步触发器生成电路生成同步触发器信号到 AD 转换器.

17.2 PMD 寄存器

下表列出控制寄存器和其地址:

关于基地址, 参考"内存映射"章节"中外设功能基地址表".

寄存器名		地址(基+)
PMD 使能 寄存器	PMDxMDEN	0x0000
端口 输出 模式 寄存器	PMDxPORTMD	0x0004
PMD 控制 寄存器	PMDxMDCR	0x0008
PWM 计数器 状态 寄存器	PMDxCNTSTA	0x000C
PWM 计数器 寄存器	PMDxMDCNT	0x0010
PWM 期间 寄存器	PMDxMDPRD	0x0014
PMD 比较 U 寄存器	PMDxCMPU	0x0018
PMD 比较 V 寄存器	PMDxCMPV	0x001C
PMD 比较 W 寄存器	PMDxCMPW	0x0020
模式 选择 寄存器	PMDxMODESEL	0x0024
PMD 传导 控制 寄存器	PMDxMDOUT	0x0028
PMD 输出 设置 寄存器	PMDxMDPOT	0x002C
EMG 释放 寄存器	PMDxEMGREL	0x0030
EMG 控制 寄存器	PMDxEMGCR	0x0034
EMG 状态 寄存器	PMDxEMGSTA	0x0038
OVV 控制 寄存器	PMDxOVVCR	0x003C
OVV 状态 寄存器	PMDxOVVSTA	0x0040
死区 时间 寄存器	PMDxDTR	0x0044
触发器 比较 0 寄存器	PMDxTRGCMP0	0x0048
触发器 比较 1 寄存器	PMDxTRGCMP1	0x004C
触发器 比较 2 寄存器	PMDxTRGCMP2	0x0050
触发器 比较 3 寄存器	PMDxTRGCMP3	0x0054
触发器 控制 寄存器	PMDxTRGCR	0x0058
触发器 输出 模式 设置 寄存器	PMDxTRGMD	0x005C
触发器 输出 选择 寄存器	PMDxTRGSEL	0x0060
触发器 更新 时序 设置 寄存器	PMDxTRGSYNR	0x0064

17.2.1 PMDxMDEN(PMD 使能 寄存器)

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

位	位 符	类型	功能
31-1	-	R	读 作 0.
0	PWMEN	R/W	使能或禁止波形综合. 0: 禁止 1: 使能 注: 端口设置为功能输出 (PWM 输出) 时, 端口禁止输出 (高 阻抗) 通过设置 <PWMEN> = "0". 注: 使能 PMD 前, 设置 <PWMEN>="1"(使能) 其他相关设置, 诸如输出端口极性.

17.2.2 PMDxPORTMD(端口 输出 模式 寄存器)

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

位	位 符	类型	功能
31-2	-	R	读作 0.
1-0	PORTMD[1:0]	R/W	<p>工具中断发生时端口控制设置</p> <p>00: 上相= 高-z / 下相= 高-z</p> <p>01: 上相= 高-z / 下相= PMD 输出</p> <p>10: 上相= PMD 输出 / 下相= 高-z</p> <p>11: 上相= PMD 输出 / 下相= PMD 输出</p> <p>设置端口输出给上相(UO/VOWO) 和下相(XO/YO/ZO)当使用功能 输出端口(PWM 输出) 工具中断发生时. "高-Z" 选择时, 工具中断发生时, 禁止端口输出 (高 阻抗). 其他时候, 外部端口输出取决于 PMD 输出.</p>

注 1: <PWMEN>=0 时, 禁止输出端口输出 (高 阻抗) 无论 PORTMD 设置.

注 2: EMG 输入发生时, 端口输出控制取决于设置 PMDxEMGCR<EMGMD[1:0]>.

17.2.3 PMDxMODESEL (模式 选择 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	DCMPEN	-	-	-	MDSEL3	MDSEL2	MDSEL1	MDSEL0
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-8	-	R	读作 0.
7	DCMPEN	R/W	VE 寄存器 和 PMD 寄存器之间自动切换 0: 2 寄存器之间禁止切换 控制(仅使用寄存器设置 <MDSEL0>). 1: 2 寄存器之间使能切换 控制(在 PWM 计数器递增递减计数状态). 注: 有效<MDSEL0> ="1"时. 注: 有效三角载波波形选择 (PMDxMDCR<PWMMD>="1")时.
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: PMDxTRGSEL) 1: VE 模式 (使用 VE 寄存器: VExTRGSEL)
2	MDSEL2	R/W	模式 选择 2 0: 总线 模式 (使用 PMD 寄存器: PMDxTRGCMP0 和 PMDxTRGCMP1) 1: VE 模式 (使用 VE 寄存器: VExTRGCMP0 和 VExTRGCMP1)
1	MDSEL1	R/W	模式 选择 1 0: 总线 模式 (使用 PMD 寄存器: PMDxMDOUT) 1: VE 模式 (使用 VE 寄存器: VExOUTCR)
0	MDSEL0	R/W	模式 选择 0 0: 总线 模式 (使用 PMD 寄存器: PMDxCMPU, PMDxCMPV1 和 PMDxCMPW) 1: VE 模式 (使用 VE 寄存器: VExCMPU, VExCMPV 和 VExCMPW 寄存器且设置 VExEMGRS 寄存器使能)

17.2.4 脉宽 调制 电路

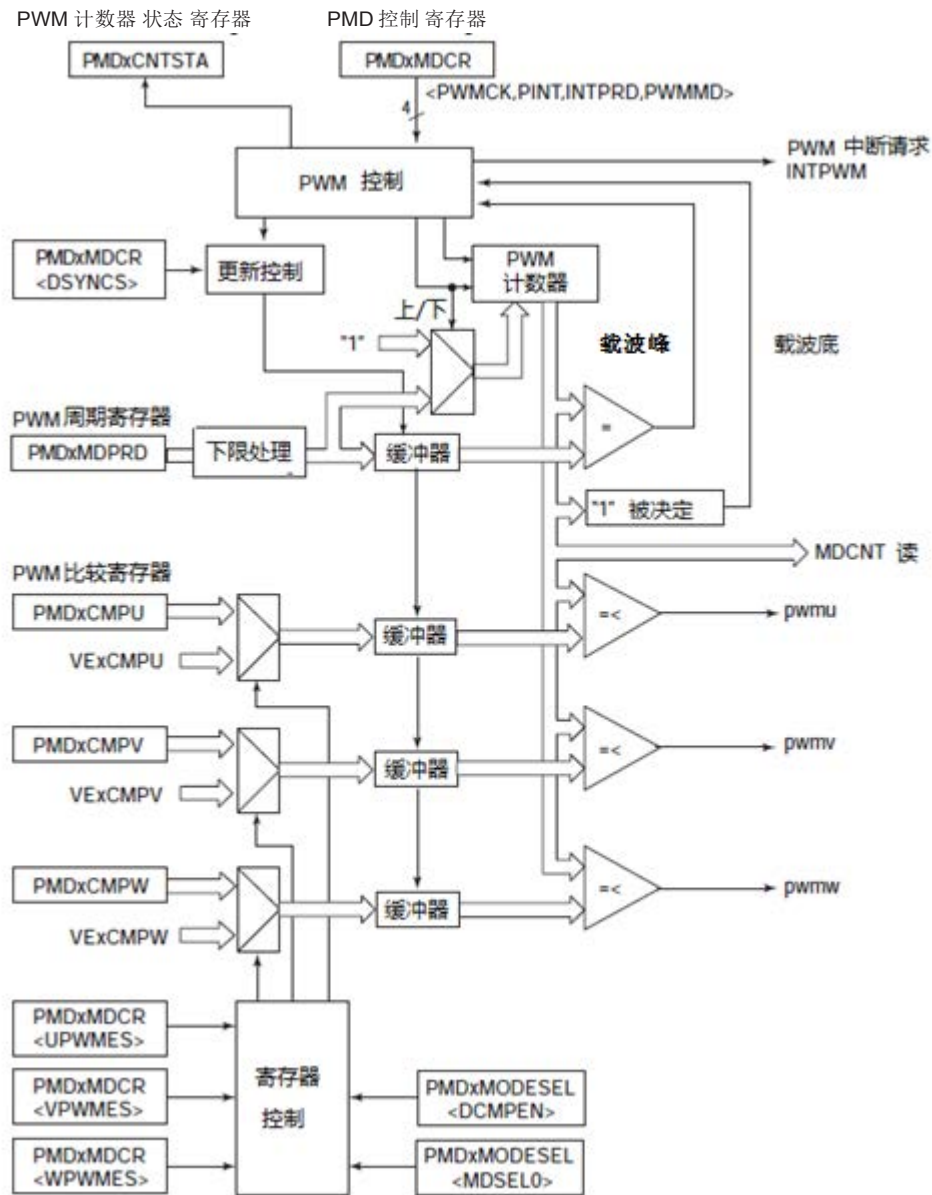


图 17-4 脉宽 调制 电路

脉宽调制电路有 16-位 PWM 递增-/递减-计数器且生成 PWM 载波波形用 $1/f_{sys}$ 分辨率(8.33[ns] 在 120[MHz])。PWM 期间扩展模式 (PMDxMDCR<PWMCK> = "1") 也可用。此模式选择时, PWM 计数器生成 PWM 载波波形用 $4/f_{sys}$ 分辨率 (33.3[ns] 在 120[MHz])。

PWM 载波波形模式可以从模式 0 (缘-排列 PWM, 锯齿波调制) 和模式 1 (中心-排列 PWM, 三角波调制) 选择。(参考"图 17-5 PWM 波形".) 三角波模式中, PWM 波形可以从中心 PWM 选择, 被固定下降 PWM 和被固定上升缘 PWM。(参考 "图 17-6 使用固定缘的 PWM 三角波载波波形".)

1. 设置 PWM 周期

PWM 周期被 PMD_xMDPRD 寄存器决定. 此寄存器双-缓冲. 随后阶段缓冲更新在每个 PWM 周期. 也可以更新在每半 PWM 周期. (参考 "表 17-1 PMD_xMDPRD, PMD_xCMPU/V/W 和 VExCMPU/V/W 缓冲更新时序".)

$$\text{锯齿波形 PWM : PMD}_x\text{MDPRD 寄存器值} = \frac{\text{振荡 频率[Hz]}}{\text{PWM 频率[Hz]}}$$

$$\text{三角波形 PWM : PMD}_x\text{MDPRD 寄存器值} = \frac{\text{振荡 频率[Hz]}}{\text{PWM 频率[Hz]} \times 2}$$

2. 比较 功能

脉宽调制电路通过比较 PWM 比较寄存器 (PMD_xCMPU/V/W) 和通过 PWM 计数器 (PMD_xMDCNT <MDCNT[15:0]>) 生成的 PWM 载波大小生成需要的占空比 PWM 波形 .

各相 PWM 比较寄存器有双-缓冲寄存器. PWM 比较寄存器值被加载到随后阶段缓冲在每个 PWM 周期. 也可以更新在每半 PWM 周期. (参考 "表 17-1 PMD_xMDPRD, PMD_xCMPU/V/W 和 VExCMPU/V/W 缓冲更新时序".)

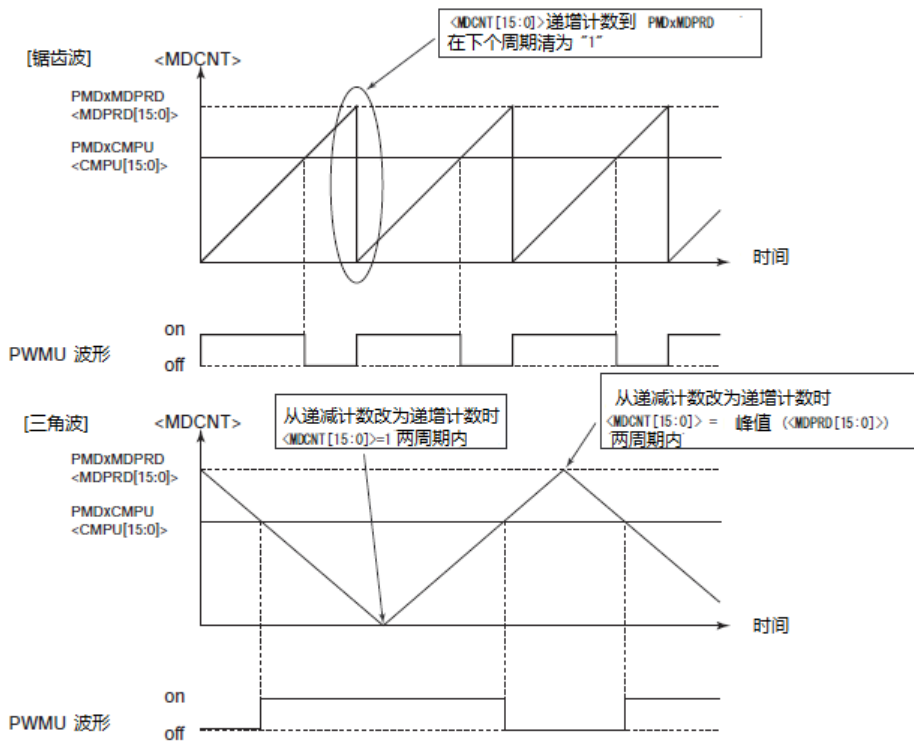


图 17-5 PWM 波形

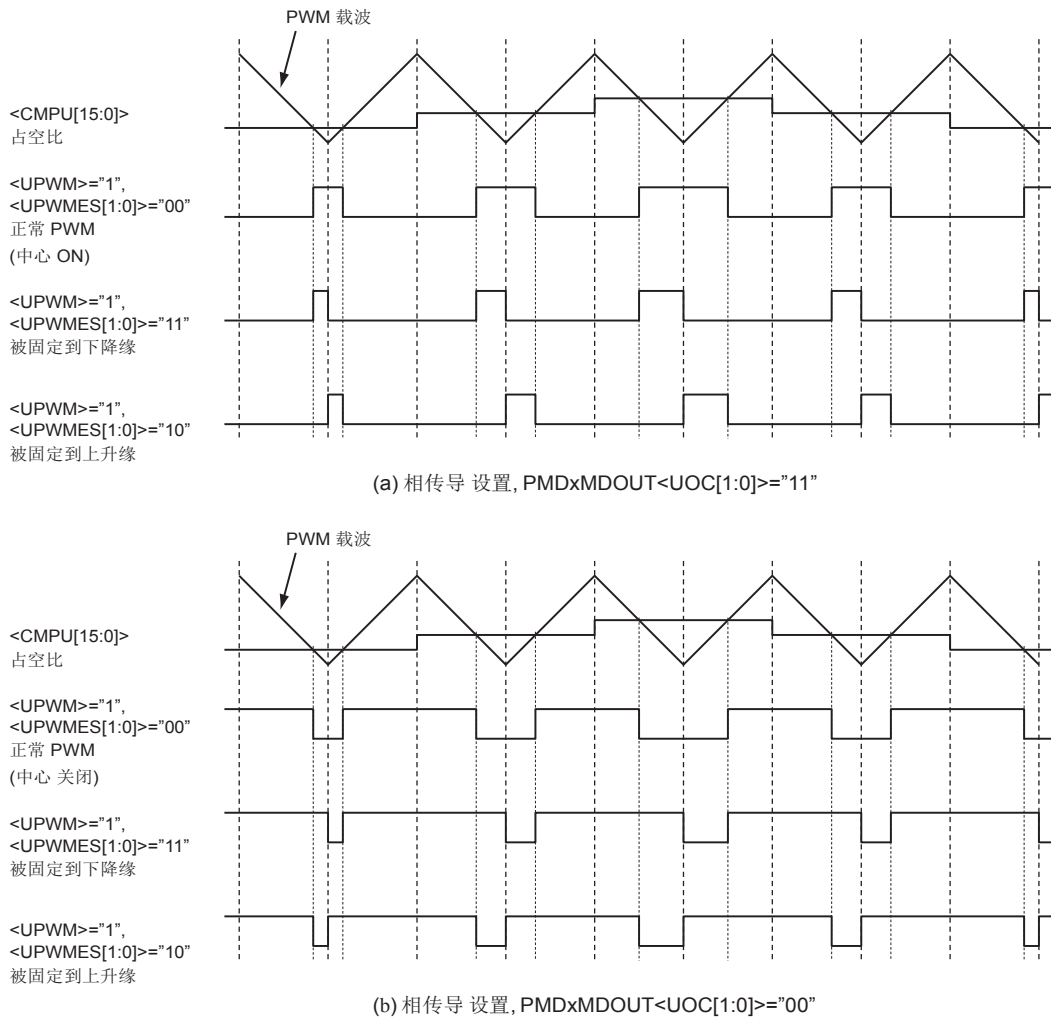


图 17-6 使用固定缘的 PWM 三角波载波波形

3. 波形模式

三相 PWM 波形可以生成在以下两模式:

1. 3-相 独立模式:

三相各 PWM 比较寄存器独立设置来生成独立的各相 PWM 波形. 此模式被用来生成驱动波形诸如正弦波形.

2. 3-相 共通模式:

仅 U-相 PWM 比较 寄存器设置来生成理想的所有三相 PWM 波形. 此模式被用来无刷 DC 电机矩形波形驱动.

4. 中断 处理中

脉宽调制电路生成 PWM 中断请求同步 PWM 波形. 中断请求时序可以在 PWM 载波峰或在 PWM 载波底选择.

PWM 中断期间可以设置半 PWM 周期, 一 PWM 周期, 两 PWM 周期或四 PWM 周期.

17.2.4.1 PMDxMDCR (PMD 控制 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	WPWMES		VPWMES		UPWMES		DSYNS	
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	DTCREN	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 0.
15-14	WPWMES[1:0]	R/W	W-相缘 设置 00: 缘未被固定到 (中心-排列 PWM) 01: 保留 10: PWM 上升-缘 被固定 (为 PWM 载波底) 11: PWM 下降-缘 被固定 (为 PWM 载波底) 注:三角 载波 PWM 选择时(<PMMMD> ="1")有效.
13-12	VPWMES[1:0]	R/W	V-相缘 设置 00: 缘 未被固定 (中心-排列 PWM) 01: 保留 10: PWM 上升-缘 被固定(为 PWM 载波底) 11: PWM 下降-缘 被固定 (为 PWM 载波底) 注: 三角载波 PWM 选择 (<PMMMD> ="1")时有效.
11-10	UPWMES[1:0]	R/W	U-相缘 设置 00: 缘 未被固定(中心-排列 PWM) 01: 保留 10: PWM 上升-缘 被固定到 (为 PWM 载波底) 11: PWM 下降-缘 被固定到 (为 PWM 载波底) 注:三角载波 PWM 选择 (<PMMMD> ="1")时有效.
9-8	DSYNS[1:0]	R/W	占空比较寄存器和 PWM 周期寄存器双缓冲更新时序. 00: 取决于中断周期设置 (参考表 17-1) 更新在载波峰和载波底 0.5 PWM 期间选择时(<INTPRD> ="00"). 否则, 更新在载波峰. 01: 更新在 PWM 载波底 10: 更新在 PWM 载波峰 11: 更新在 PWM 载波峰和底 注 1:锯齿波载波选择时(<PMMMD>="0") 无论设置更新在载波峰. 注 2: PMDxMDEN<PWMEN>="0"时, 更新异步无论设置.
7	DTCREN	R/W	设置死区时间修改. 0: 禁止 1: 使能
6	PWMCK	R/W	PWM 周期扩展 模式 0: 正常周期 1: 4 × 周期 设置 PWM 计数器计数周期. 正常周期设置: 锯齿波 1/fsys (8.33[ns] 在 120[MHz]) / 三角波 2/fsys (16.7[ns] 在 120[MHz]) 四倍 周期设置: 锯齿波 4/fsys (33.3[ns] 在 120[MHz]) / 三角波 8/fsys (66.7[ns] 在 120[MHz])
5	SYNTMD	R/W	端口 输出 模式 端口 输出通过<nOC>,<nPWM>,<POLH>,<POLL> 和 <SYNTMD>的组合控制(参考表 17-4).

位	位符	类型	功能
4	DTYMD	R/W	占空比 模式 0: 3-相共通模式 1: 3-相独立模式 此位选择是否使各相占空比设置独立或使用 PMDxCMPU 寄存器为 3-相共通。
3	PINT	R/W	PWM 中断请求时序 0: 中断请求发生在 PWM 载波 底 (PMDxMDCNT<MDCNT[15:0]> = 0x0001). 1: 中断请求发生在 PWM 载波 峰 (PMDxMDCNT<MDCNT[15:0]> = <MDPRD[15:0]>). 注 1: PWM 载波为锯齿波 (<PWMMMD>="0 时")中断请求发生在载波峰. 注 2: 中断周期为 0.5 周期 (<INTPRD>="00")时中断请求发生在载波峰和载波底.
2-1	INTPRD[1:0]	R/W	PWM 中断请求周期 00: 中断请求在每 0.5 PWM 周期 注 1: 仅 PWM 载波为三角波 (<PWMMMD>="1")时 PWM 中断请求周期可以配置 注 2: 通过 PWM 载波峰和底比较寄存器 (PMDxCMPU/V/W) 和周期寄存器 (PMDxMDPRD)的双缓冲更新. 01: 中断 请求在每 PWM 周期 10: 中断 请求在每两 PWM 周期 11: 中断 请求在每四 PWM 周期 此域选择 PWM 中断请求周期从 0.5 PWM 周期, 一 PWM 周期, 两 PWM 周期和四 PWM 周期.
0	PWMMMD	R/W	PWM 载波波形 0: PWM 模式 0 (缘-排列 PWM 和锯齿波) 1: PWM 模式 1(中心-排列 PWM 和三角波)

表 17-1 PMDxMDPRD, PMDxCMPU/V/W 和 VExCMPU/V/W 缓冲 更新 时序

设置		更新 时序
<DSYNCS[1:0]>	<INTPRD[1:0]>	
00	1x	更新在 PWM 载波峰
	x1	更新在 PWM 载波峰
	00	更新在 PWM 载波峰和 PWM 载波底
01	xx	更新在 PWM 载波底
10	xx	更新在 PWM 载波峰
11	xx	更新在 PWM 载波峰和 PWM 载波底

x :忽略

表 17-2 PMDxCMPU/V/W 和 VExCMPU/V/W 的切换控制

共通设置		各相设置	寄存器 选择信号
<DSYNS[1:0]>	<INTPRD[1:0]>	<UPWMES[1]> <VPWMES[1]> <WPWMES[1]>	
01	xx	x	VE 寄存器
10	xx	x	VE 寄存器
11	xx	0	递增-计数: PMD 寄存器期间 递减-计数: VE 寄存器期间
		1	VE 寄存器
00	00	0	递增-计数: PMD 寄存器期间 递减-计数: VE 寄存器期间
		1	VE 寄存器

注: <MSEL0>="1", <DCMEN>="1" 和 <PWMMD>="1"时有效.

x:忽略

17.2.4.2 PMDxCNTSTA (PWM 计数器 状态 寄存器)

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

位	位符	类型	功能
31-1	-	R	读作 0.
0	UPDWN	R	PWM 计数器 标志 0: 递增-计数 1: 递减-计数 此位指是否 PWM 计数器为递增-计数或递减-计数. 注: PWM 载波为锯齿波 (PMDxMDCR<PWMMD>="0"), 始终读零.

17.2.4.3 PMDxMDCNT (PWM 计数器 寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	MDCNT[15:0]	R	<p>PWM 计数器 可以从递增递减计数器读值生成 PWM 载波波形. 计数器分辨率: 1/fsys (8.33[ns] 在 120[MHz])</p> <p>注 1:四倍周期模式选择 (PMDxMDCR<PWMCK>="1")时, 计数器时间分辨率为 4/fsys (33.3[ns] 在 120[MHz]). 注 2: 取决于 PWM 载波 (PMDxMDCR<PMMMD>), PWM 计数器值的设置 PMD 禁止 (PMDxMDEN<PWMEN>="0")如下时: PMDxMDCR<PMMMD>="0" : 0x0001 时 PMDxMDCR<PMMMD>="1" 时: PMDxMDPRD<MDPRD[15:0]> 的值</p>

17.2.4.4 PMD_xMDPRD (PWM 周期 寄存器)

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

位	位符	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>通过 PWM 周期扩展模式 (PMD_xMDCR\langlePWMCK\rangle) 和 PWM 载波波形 \langlePWMMD\rangle 的组合, PWM 周期可以被计算为如下:</p> <p>$\langle \text{PWMCK} \rangle = "0", \langle \text{PWMMD} \rangle = "0"$ 时: $\langle \text{MDPRD} \rangle \times 1/\text{fsys}$</p> <p>$\langle \text{PWMMD} \rangle = "1"$: $\langle \text{MDPRD} \rangle \times 2/\text{fsys}$</p> <p>$\langle \text{PWMCK} \rangle = "1", \langle \text{PWMMD} \rangle = "0"$ 时: $\langle \text{MDPRD} \rangle \times 4/\text{fsys}$</p> <p>$\langle \text{PWMMD} \rangle = "1"$: $\langle \text{MDPRD} \rangle \times 8/\text{fsys}$</p> <p>注: 如果 $\langle \text{MDPRD}[15:0] \rangle$ 设置为小于 0x0010 的值, 自动被认为 0x0010. (寄存器维持实际写入值.)</p>

注 1: 不要写入此寄存器以字节单元. 如果上 8 位 [15:8] 和低 8 位 [7:0] 被分别写入, 工作不可以被确保.

注 2: 因为 PMD_xMDPRD 寄存器双缓冲, PWM 计数器工作期间周期可以被改变.

注 3: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-1 PMD_xMDPRD, PMD_xCMPU/V/W 和 VExCMPU/V/W 缓冲更新时序".

注 4: 读到的值 为第一个缓冲值 (通过总线最近的测试数据设置).

17.2.4.5 PMDxCMPU (PWM 比较 寄存器 U 相)

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

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	CMPU[15:0]	R/W	<p>U 相 PWM 脉宽 0x0000 到 0xFFFF 注: <CMPU> > <MDPRD> 时, 占空比为 100%.</p> <p><CMPU[15:0]> 为比较寄存器决定输出 U 相脉宽. 这些寄存器双-缓冲. 脉宽通过比较随后阶段缓冲和 PWM 计数器来评估哪个小或大. 通过 PWM 期间扩展模式 (PMDxMDCR<PWMCK>) 和 PWM 载波波形(<PWMMMD>)的组合, 脉宽可以被计算为如下: <PWMCK>="0", <PWMMMD>="0"时: <CMPU> × 1/fsys <PWMMMD>="1" : <CMPU> × 2/fsys <PWMCK>="1", <PWMMMD>="0"时: <CMPU> × 4/fsys <PWMMMD>="1" : <CMPU> × 8/fsys</p>

- 注 1: 要加载通过总线用比较寄存器中更新的随后阶段缓冲值, 选择总线模式 (默认) 通过设置 PMDxMODESEL<MDSEL0> 为"0".
- 注 2: 不要写入此寄存器以字节单元. 如果上 8 位 [15:8] 和低 8 位 [7:0] 被分别写入,工作不可以被确保.
- 注 3: 因为 PMDxCMPU 寄存器双缓冲, PWM 计数器工作期间周期可以被改变 .
- 注 4: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-1 PMDxMDPRD, PMDxCMPU/V/W 和 VExCMPU/V/W 缓冲更新时序".
- 注 5: 读到的值为第一个缓冲值 (最近的通过总线测试数据设置).

17.2.4.6 PMDxCMPV (PWM 比较 寄存器 V 相)

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

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	CMPV[15:0]	R/W	<p>V 相 PWM 脉宽 0x0000 到 0xFFFF 注: <CMPV> > <MDPRD> 时, 占空比为 100%.</p> <p><CMPV[15:0]> 为比较寄存器决定输出 V 相脉宽. 这些寄存器双-缓冲. 脉宽通过比较随后阶段缓冲和 PWM 计数器来评估哪个小或大. 通过 PWM 期间扩展模式 (PMDxMDCR<PWMCK>) 和 PWM 载波波形(<PWMMD>)的组合, 脉宽可以被计算为如下: <PWMCK>="0", <PWMMD>="0"时: <CMPU> × 1/fsys <PWMMD>="1" : <CMPU> × 2/fsys <PWMCK>="1", <PWMMD>="0"时: <CMPU> × 4/fsys <PWMMD>="1" : <CMPU> × 8/fsys</p>

注 1: 要加载通过总线用比较寄存器中更新的随后阶段缓冲值, 选择总线模式 (默认) 通过设置 PMDxMODESEL<MDSEL0> 为"0".

注 2: 不要写入此寄存器以字节单元. 如果上 8 位 [15:8] 和低 8 位 [7:0] 被分别写入, 工作不可以被确保.

注 3: 因为 PMDxCMPV 寄存器双缓冲, PWM 计数器工作期间周期可以被改变.

注 4: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-1 PMDxMDPRD, PMDxCMPU/V/W 和 VExCMPU/V/W 缓冲更新时序".

注 5: 读到的值为第一个缓冲值 (最近的通过总线测试数据设置).

17.2.4.7 PMDxCMPW (PWM 比较 寄存器 W 相)

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

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	CMPW[15:0]	R/W	<p>W 相 PWM 脉宽 0x0000 到 0xFFFF 注: <CMPW > > <MDPRD >时, 占空比为 100%.</p> <p><CMPW [15:0]> 为比较寄存器决定输出 W 相脉宽. 这些寄存器双-缓冲. 脉宽通过比较随后阶段缓冲和 PWM 计数器来评估哪个小或大. 通过 PWM 期间扩展模式 (PMDxMDCR<PWMCK>) 和 PWM 载波波形(<PWMMMD>)的组合, 脉宽可以被计算为如下: <PWMCK>="0", <PWMMMD>="0"时: <CMPU> × 1/fsys <PWMMMD>="1" : <CMPU> × 2/fsys <PWMCK>="1", <PWMMMD>="0"时: <CMPU> × 4/fsys <PWMMMD>="1" : <CMPU> × 8/fsys 时 <PWMCK>="1", <PWMMMD>="0" : <CMPW> × 4/fsys <PWMMMD>="1" : <CMPW> × 8/fsys</p>

注 1: 要加载通过总线用比较寄存器中更新的随后阶段缓冲值, 选择总线模式 (默认) 通过设置 PMDxMODESEL<MDSEL0> 为"0".

注 2: 不要写入此寄存器以字节单元. 如果上 8 位 [15:8] 和低 8 位 [7:0] 被分别写入,工作不可以被确保.

注 3: 因为 PMDxCMPW 寄存器双缓冲, PWM 计数器工作期间周期可以被改变 .

注 4: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-1 PMDxMDPRD, PMDxCMPU/V/W 和 VExCMPU/V/W 缓冲更新时序".

注 5: 读到的值为第一个缓冲值 (最近的通过总线测试数据设置).

17.2.5 传导控制 电路

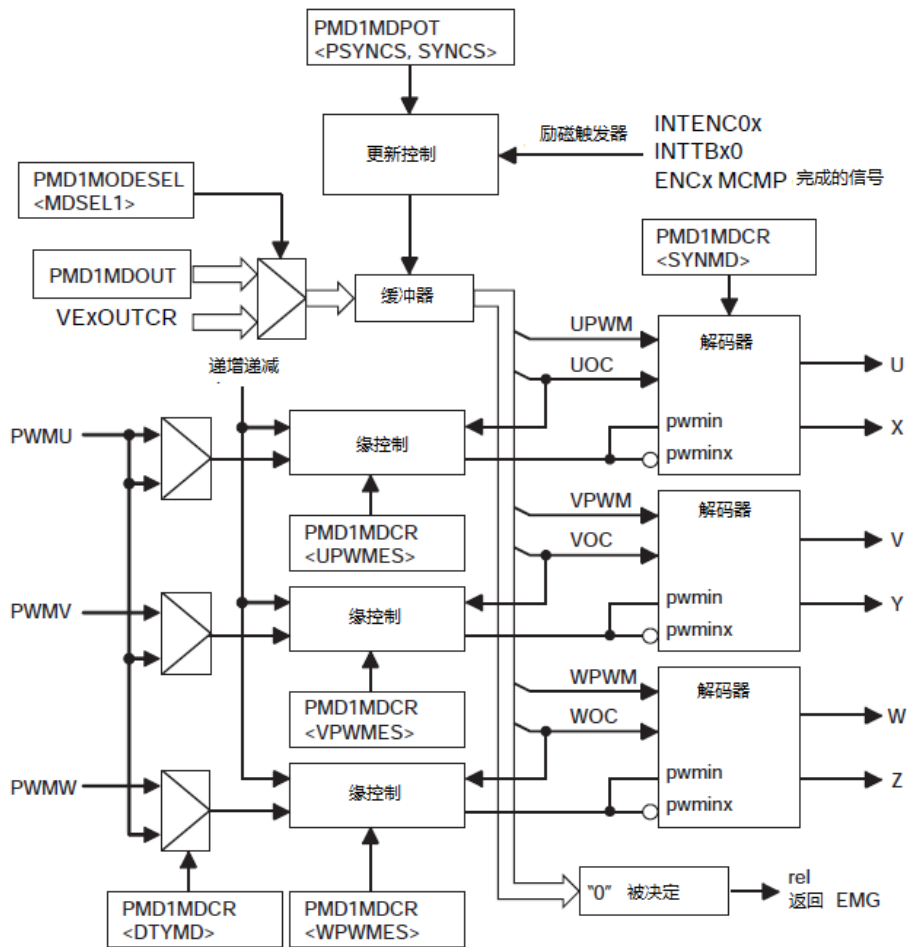


图 17-7 传导控制 电路

传导控制电路执行输出端口控制根据输出控制寄存器 $PMDxMDOUT(VExOUTCR)$ 和输出设置寄存器 $PMDxMDPOT$ 中的设置。 $PMDxMDOUT(VExOUTCR)$ 寄存器为双-缓冲且更新时序可以选择为同步或异步 PWM。也可以用触发器输入选择更新时序同步。(关于详细更新时序, 参考"表 17-3 $PMDxMDOUT(VExOUTCR)$ 缓冲的更新时序".)

使用 $PMDxMDPOT<POLH>$, $<POLL>$, 六个输出端口可以分别设置低-激活或高-激活在上-相输出 (UO, VO, WO) 或下-相输出 (XO, YO, ZO)。另外, $PMDxMDOUT(VExOUTCR)$ 寄存器的 $<WPWM>$, $<VPWM>$, $<UPWM>$ 选择各 U, V 和 W 相 PWM 或高/低 输出。PWM 输出选择时, 输出 PWM 波形。高/ 低输出选择时, 输出被固定到高或低电平。设置各输出高或低通过寄存器 $PMDxMDOUT(VExOUTCR)$ 的 $<WOC>$, $<VOC>$, $<UOC>$ 。

"表 17-4 端口输出根据 $<UOC>$, $<VOC>$, $<WOC>$, $<UPWM>$, $<VPWM>$ 和 $<WPWM>$ 设置" 表示根据 $PMDxMDOUT(VExOUTCR)$ 寄存器和 $PMDxMDPOT$ 寄存器中的端口输出设置的端口输出设置, 和 $PMDxMDCR$ 寄存器端口输出模式中的端口输出极性设置。

传导控制电路输出 PWM 信号 (PWMON) 到编码器输入电路 (AENC)。AENC 中外输入信号用 PWMON 同步采样。

表 17-3 PMD_xMDOUT(VE_xOUTCR) 缓冲的更新时序

		PSYNCS 设置			
		00	01	10	11
SYNCS 设置	00	恒定 更新	PWM 载波底	PWM 载波峰	PWM 载波峰和 PWM 载波底
	01	INTENC0x 上升时.	每次 IN- TENC0x 上升时 第一个 PWM 载波底.	每次 INTENC0x 上升时 第一个 PWM 载波峰.	每次 INTENC0x 上升时 第一个 PWM 载波峰或第 一个载波底.
	10	INTTBx0 上升时.	每次 INTTBx0 上升时 第一个 PWM 载波底.	每次 INTTBx0 上升时 第一个 PWM 载波峰.	每次 INTTBx0 上升时 第一个 PWM 载波峰或第 一个载波底.
	11	MCMP 完成信号 从 ENCx (CTRGO) 上升 时.	每次 CTRGO 上升时 第一个 PWM 载波底.	每次 CTRGO 上升时 第一个 PWM 载波峰.	每次 CTRGO 上升时 第一个 PWM 载波峰或第 一个载波底.

注: 如果 PMD 禁止 (PMDxMDCR<PMWEN>="0"), 清除被维持的触发器条件.

17.2.5.1 PMDxMDPOT (PMD 输出 设置 寄存器)

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

位	位符	类型	功能
31-10	-	R	读作 0.
9-8	SYNCS[1:0]	R/W	选择 PMDxMDOUT(VExOUTCR) 传输 时序 (触发器 同步 设置). 00: 异步 01: INTENCx (ENCx 中断 请求) 发生时 10: INTTBx0 (TMRBx 中断 请求) 发生时 11: CTRGO(ENCx MCMP 完成) 发生时 选择传导控制寄存器的随后阶段缓冲更新时序 . 注 1:通过<PSYNC> 和 <SYNCS>设置的组合, 缓冲更新时序可以被决定 (参考"表 17-3 PMDxMDOUT(VExOUTCR) 缓冲的更新时序"). 注 2: PMD 禁止 (PMDxMDEN<PWMEN>="0")时, 时序为异步无论设置.
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: 激活 低 1: 激活 高
2	POLL	R/W	选择低相输出 (XO, YO, ZO)的输出极性. 0: 激活 低 1: 激活 高
1-0	PSYNCS[1:0]	R/W	选择 PMDxMDOUT(VExOUTCR) 传输时序 (PWM 同步设置). 00: 异步 PWM 在 PMDxMDOUT/VExOUTCR 寄存器同时, 设置被应用到端口输出. 01: 载波底 (<MDCNT[15:0]>="1"时) 10: 载波峰 (<MDCNT[15:0]>=<MDPRD[15:0]>) 11: 载波峰和载波底 选择传导控制寄存器的随后阶段缓冲更新时序. 注 1: PWM 载波为锯齿波时, 缓冲更新时序为载波峰, 除 <PSYNCS>="00" . 注 2: 通过<PSYNC> 和 <SYNCS>设置的组合, 缓冲更新时序可以被决定 (参考 "表 17-3 PMDxMDOUT(VExOUTCR) 缓冲的更新时序"). 注 3: PMD 禁止 (PMDxMDEN<PWMEN>="0")时, 时序为异步无论设置.

注:PMDxMDEN<PWMEN>="0"时, 此域必须设置.

17.2.5.2 PMDxMDOUT (PMD 传导控制 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	WPWM	VPWM	UPWM
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	WOC		VOC		UOC	
复位后	0	0	0	0	0	0	0	0

位	位符	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 通过 <WOC>,<WPWM>,<POLH>,< POLL> 和 <SYNTMD>的组合控制(参考表 17-4).
9	VPWM	R/W	V-相 PWM 输出 设置 0: 高/低 输出 1: PWM 输出 端口输出通过 <VOC>,<VPWM>,<POLH>,< POLL> 和 <SYNTMD>的组合控制(参考表 17-4).
8	UPWM	R/W	U-相 PWM 输出 设置 0: 高/低 输出 1: PWM 输出 端口输出通过 <UOC>,<UPWM>,<POLH>,< POLL> 和 <SYNTMD>的组合控制(参考表 17-4).
7-6	-	R	读作 0.
5-4	WOC[1:0]	R/W	W-相传导控制 设置 端口输出通过<WOC>,<WPWM>,<POLH>,< POLL> 和 <SYNTMD> (参考表 17-4).
3-2	VOC[1:0]	R/W	V-相传导控制 设置 端口输出通过 <VOC>,<VPWM>,<POLH>,< POLL> 和 <SYNTMD>的组合控制(参考表 17-4).
1-0	UOC[1:0]	R/W	U-相传导控制 设置 端口输出通过<UOC>,<UPWM>,<POLH>,< POLL> 和 <SYNTMD>的组合控制(参考表 17-4).

注 1: 要加载通过总线用 PMDxMDOUT(VExOUT)寄存器中更新的随后阶段缓冲值, 选择总线模式 (默认) 通过设置 PMDxMODESEL<MDSEL0> 为"0".

注 2: 不要写入此寄存器以字节单元. 如果上 8 位 [15:8] 和低 8 位 [7:0] 被分别写入,工作不可以被确保.

注 3: 因为传导控制寄存器为双缓冲, PWM 计数器工作期间周期可以被改变

注 4: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-3 PMDxMDOUT(VExOUTCR) 缓冲更新 时序".

注 5: 读到的值为第一个缓冲值 (最近的通过总线测试数据设置).

表 17-4 端口输出根据 <UOC>,<VOC>,<WOC>,<UPWM>,<VPWM> 和 <WPWM> 设置

PMDxMDCR<SYNTMD>="0"

极性: 高激活(PMDxMDPOT<POLH>< POLL>="11")

PMDxMDOUT 传导控制		<WPWM><VPWM><UPWM> PWM 输出设置			
(上相)	(低相)	0: H/L 输出		1: PWM 输出	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	上相 输出	低相 输出	上相 输出	低相 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	$\overline{\text{PWM}}$

PMDxMDCR<SYNTMD>=0

极性: 低激活(PMDxMDPOT<POLH>< POLL>="00")

PMDxMDOUT 传导控制		<WPWM><VPWM><UPWM> PWM 输出设置			
(上相)	(低相)	0: H/L 输出		1: PWM 输出	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	上相 输出	低相 输出	上相 输出	低相 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

PMDxMDCR<SYNTMD>=1

极性: 高激活(PMDxMDPOT<POLH>< POLL>="11")

PMDxMDOUT 传导控制		<WPWM><VPWM><UPWM> PWM 输出设置			
(上相)	(低相)	0: H/L 输出		1: PWM 输出	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	上相 输出	低相 输出	上相 输出	低相 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	$\overline{\text{PWM}}$

PMDxMDCR<SYNTMD>=1

极性: 低激活(PMDxMDPOT<POLH>< POLL>="00")

PMDxMDOUT 传导控制		<WPWM><VPWM><UPWM> PWM 输出设置			
(上相)	(低相)	0: H/L 输出		1: PWM 输出	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	上相 输出	低相 输出	上相 输出	低相 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

17.2.6 保护 控制 电路

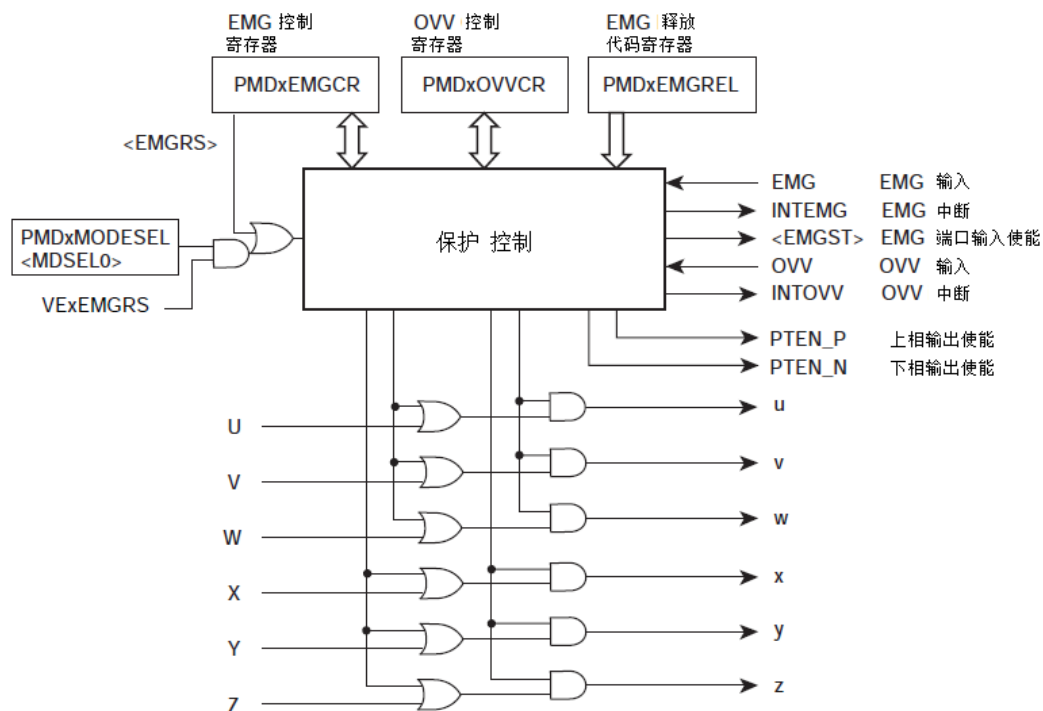


图 17-8 保护 控制 电路

保护控制电路由 EMG 保护控制电路和 OVV 保护控制电路组成.

17.2.6.1 EMG 保护 电路

EMG 保护电路由 EMG 保护控制单元和端口输出禁止单元组成. EMG 输入变为低时此电路被激活.

EMG 保护电路提供紧急情况停止机能: EMG 输入被启动 (H→L)时, 所有六个端口输出立即禁止 (取决于 PMDxEMGCR<EMGMD> 设置) 且 EMG 中断 (INTEMG) 生成.紧急情况时可以设置 <EMGMD> 输出设置外部输出端口为高-z 的控制信号.

工具中断也禁止所有六个 PWM 输出线取决于 PMDxPORTMD<PORTMD> 设置. 工具中断发生时, 外部输出端口可以设置为高-z 通过 PMDxPORTMD<PORTMD> 寄存器的设置. EMGSTA<EMGST> 中读到的值 1 指 EMG 保护电路激活.

通过 EMG 控制寄存器 (PMDxEMGCR) 设置 EMG 保护.

EMG 保护状态下,通过设置所有端口输出线非激活 (设置 "0" 到 PMDxMDOUT(VExOUTCR)<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC>.) (注 1) 然后设置 PMDxEMGCR<EMGRS> 或 VExEMGRS<EMGRS> 为"1"可以释放. 要禁止 EMG 保护功能, 以此命令写入 "0x5A" 和 "0xA5"到 PMDxEMGREL 寄存器然后清除 PMDxEMGCR<EMGEN> 为 "0". (这三个指令必须连续被执行.) EMG 保护输入为低时, 任何试图释放 EMG 保护状态被忽略. 确认 PMDxEMGSTA<EMGI>的状态标志为"1"后可以释放 EMG 保护状态.

仅被指定的关键代码 ("0x5A", "0xA5") 被写入 PMDxEMGREL 寄存器后可以禁止 EMG 保护电路防止无意禁止.

注 1: PMDxMDOUT(VExOUTCR)的数据需要被反映在随后阶段缓冲.

注 2: EMG 功能初始处理

复位后, EMG 功能使能但 EMG 引脚被配置为正常端口.因此, EMG 保护可能有效时, 在初始顺序通过以下处理释放 EMG 保护.

- 1: 选择 EMG 功能通过 PxFR 寄存器.
- 2: 读 PMDxEMGSTA<EMGI> 确认为 "1".
- 3: 设置 PMDxMDOUT(VExOUTCR)<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC> 为 "0" 使所有 端口非激活.
- 4: 释放 EMG 保护通过设置 PMDxEMGCR(VExEMGRS)<EMGRS> 为 "1".

如果要禁止 EMG 保护, 继续以下处理.

- 5: 写入关键代码到 PMDxEMGREL (以 "0x5A" 和 "0xA5"命令)
- 6: 设置 PMDxEMGCR<EMGEN>为 "0" 禁止 EMG 保护.

17.2.6.2 PMDxEMGREL (EMG 释放 寄存器)

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

位	位符	类型	功能
31-8	-	R	读作 0.
7-0	EMGREL[7:0]	W	EMG/OVV 禁止代码 EMG 和 OVV 保护功能可以禁止通过以此命令设置 0x5A 和 0xA5 到寄存器. 写入禁止代码后, 立即设置 PMDxEMGCR<EMGEN>="0" 或 PMDxOVVCR<OVVEN>="0". 禁止这些功能时, <EMGEN> 和 <OVVEN> 必须清除为 "0".

注:在禁止 EMG 和 OVV 时各写入禁止代码.

17.2.6.3 PMDxEMGCR (EMG 控制 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	EMGCNT			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	INHEN	EMGMD		-	EMGRS	EMGEN
复位后	0	0	1	1	1	0	0	1

位	位 符	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 到 0xF (时 <EMGCNT[3:0]>="0", 噪声过滤器被旁路.) 异常检测输入噪声过滤长设置到这些位. 且此值可以通过以下方程式计算. <EMGCNT[3:0]> × 16/fsys (分辨率: 133[ns] 在 120[MHz])
7-6	-	R	读作 0.
5	INHEN	R/W	工具中断使能/禁止 0: 禁止 1: 使能 PMD 停止信号从工具输入时, 此位选择是否停止 PMD. 注: 初始状态下使能工具中断.
4-3	EMGMD[1:0]	R/W	EMG 保护 模式 选择 00: 所有 相高-Z 01: 所有 上相 ON / 所有下相高-Z 10: 所有 上相 高-Z / 所有下相 ON 11: 所有相高-Z 设置 EMG 发生时端口输出上 (UO, VO, WO) 和下 (XO, YO, ZO). 注: "ON" 指 PWM 输出继续.
2	-	R/W	始终写入 "0".
1	EMGRS	W	EMG 保护 释放 0: - 1: 释放 保护 通过设置 PMDxMDOUT(VExOUTCR) 寄存器为 "0x000" 然后设置 <EMGRS> 位为 "1"可以释放 EMG 保护. 注: 此位始终读为 0. 注: 如果 PMDxMDOUT(VExOUTCR) 的随后阶段缓冲不更新为"0x000", EMG 保护不可以释放. 注: 释放 EMG 保护前, 确保 PMDxEMGSTA<EMGI> 返回为 "1".
0	EMGEN	R/W	EMG 保护电路使能/禁止 0: 禁止 1: 使能 要禁止此功能, 写入"0x5A" 然后写入 "0xA5" 到 EMG 释放寄存器(PMDxEMGREL).然后, 设置 "0" 到 <EMGEN> (这三个指令必须被连续执行.). 注: 初始状态下使能此 EMG 保护电路.

17.2.6.4 PMDxEMGSTA (EMG 状态 寄存器)

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

位	位 符	类型	功能
31-2	-	R	读作 0.
1	EMGI	R	EMG 输入 EMG 保护状态 EMG 输入状态可以通过读此位被区分
0	EMGST	R	EMG 保护 状态 0: 正常 工作 1: 被保护 EMG 保护状态可以通过读此位被区分.

17.2.6.5 OVV 保护 控制 电路 (OVV 输入 方块)

OVV 保护控制电路由 OVV 保护控制单元和端输出禁止单元组成. OVV 输入端口被启动时此电路被激活.

OVV 输入信号被启动 (H→L) 到指定的期间 (设置 PMDxOVVCR <OVVCNT>)时, 传导控制电路中 OVV 保护电路固定六个端口输出线为高或低. 此时, OVV 中断 (INTOVV)生成. 仅可以选择上相, 下相或所有相.

OVV 保护通过 OVV 控制寄存器的 PMDxOVVCR 设置. PMDxOVVSTA<OVVST> 中读到的值 "1"指 OVV 保护电路激活.

OVV 保护状态的释放通过设置 PMDxOVVCR<OVVRS>为"1"使能. 且保护输入消除后, 在预定的时序 OVV 保护自动释放. (OVV 保护输入为低时, OVV 保护状态不释放. 此端口输入的状态可以通过读 PMDxOVVSTA<OVVI>检查.)

OVV 保护状态释放同步 PWM 周期 (在 PWM 计数 PMDxMDCNT 匹配 PMDxMDPRD 时的时序. 然而如果在 PWM 半周期设置中断, PWM 计数为"1"或匹配 PMDxMDPRD 时释放保护状态.). 要禁止 OVV 保护功能, 以此命令写入 "0x5A" 和 "0xA5"到 EMG 释放寄存器的 PMDxEMGREL 然后清除 PMDxOVVCR<OVVEN> 为 "0". (这三个指令必须被连续执行.)

仅被指定的关键代码 ("0x5A", "0xA5") 被写入 PMDxEMGREL 寄存器后 OVV 保护电路可以禁止防止无意禁止.

17.2.6.6 PMDxOVVCR (OVV 控制 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	OVVCNT			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-12	-	R	读作 0.
11-8	OVVCNT[3:0]	R/W	OVV 输入 检测 时间 值: 0x1 到 0xF (如果 "0" 设置, 处理为"1".) OVV 输入噪声过滤长设置到这些位. 且此值可以通过以下方程式计算. <OVVCNT[3:0]> × 16/fsys (分辨率: 133[ns] 在 120[MHz]) 注:仅端口输入选择时(<OVVISEL>="0"), <OVVCNT[3:0]>有效.
7	-	R	读作 0.
6	ADIN1EN	R/W	ADC 监视器 功能 1 输入使能 0: 禁止 输入 1: 使能 输入 选择使能/禁止从 AD 转换器的 ADC 监视器功能 1 的信号. 如果使能它和选择输入 (<OVVISEL>="1")的 ADC 监视器信号, ADC 监视器功能 1 作为 OVV 输入的结果 (如果 OVV 保护使能). 注: 关于 AD 转换监视器功能详细信息参考"12-位 模拟/数字 转换器"中"AD 转换监视器功能"章节.
5	ADIN0EN	R/W	ADC 监视器 功能 0 输入使能 0: 禁止 输入 1: 使能 输入 选择使能/禁止从 AD 转换器的 ADC 监视器功能 0 的信号. 如果使能它和选择输入 (<OVVISEL>="1")的 ADC 监视器信号, ADC 监视器功能 0 作为 OVV 输入的结果 (如果 OVV 保护使能). 注: 关于 AD 转换监视器功能详细信息参考"12-位 模拟/数字 转换器"中"AD 转换监视器功能"章节.
4-3	OVVMD[1:0]	R/W	选择 OVV 保护 模式 00: 无输出 控制 01: 所有上相 ON, 所有下相 OFF 10: 所有上相 OFF, 所有下相 ON 11: 所有相 OFF OVV 条件发生时此域控制上 (UO,VO,WO) 和下(XO,YO,ZO) 相的输出. 注: "ON" 指被固定到激活输出. "OFF" 指被固定到非激活输出. 激活和非激活取决于 < POLL> 和<POLH>的设置. 注: 如果同时 OVV 和 EMG 条件发生, <EMGMD[1:0]>的此位中保护模式设置变为有效.
2	OVVISEL	R/W	选择 OVV 输入 0: 端口 输入 1: ADC 监视器 信号 此位选择是否从被输入到保护电路的作为 OVV 信号的 ADC 使用 端口输入或监视器信号. 注: ADC 监视器 信号选择时, OVV 输入检测时间的设置<OVVCNT[3:0]> 变为无效.(直接 输入)
1	OVVRS	R/W	选择 OVV 保护 状态 释放 0: 禁止自动释放 OVV 保护 状态 1: 使能自动释放 OVV 保护 状态 注: 如果自动释放 OVV 保护使能, 检测异常 (OVV 输入使高-到低过渡)后状态变为 OVV 保护时, OVV 输入过渡到"高"后更新 PWM 周期 寄存器 PMDxMDPRD 的缓冲时, 可以自动释放 OVV 保护状态.(参考 "表 17-1 PMDxMDPRD, PMDxCMPU/V/W 和 VExCMPU/V/W 缓冲 更新时序")
0	OVVEN	R/W	OVV 保护 电路 使能/禁止 0: 禁止 1: 使能 注: 要禁止此功能, 写入 "0x5A" 然后写入 "0xA5"到 EMG 释放寄存器 (PMDxEM- GREL). 然后, 设置 "0" 到 <OVVEN>. (此三个指令必须被连续执行.)

17.2.6.7 PMDxOVVSTA (OVV 状态 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作 0.
1	OVVI	R	OVVI 输入 OVVI 状态 OVV 输入状态 (通过 PMDxOVVCR<OVVISEL> 选择) 可以通过读此位被区分.
0	OVVST	R	OVV 保护 状态 0: 正常 工作 1: 被保护中 OVV 状态可以通过读此位被区分.

17.2.7 死区 时间 控制 电路

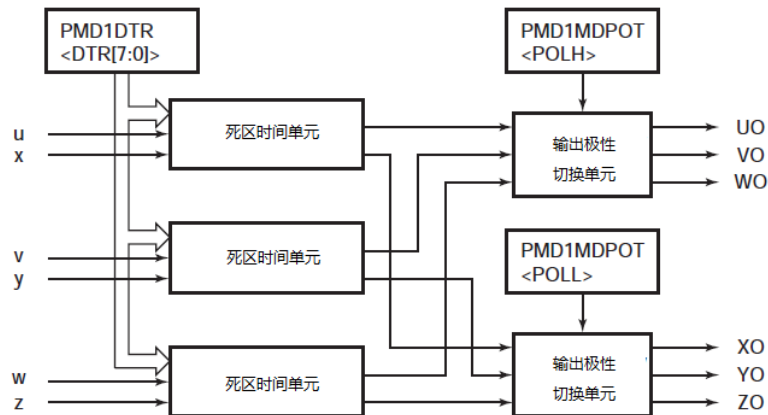


图 17-9 死区时间控制电路

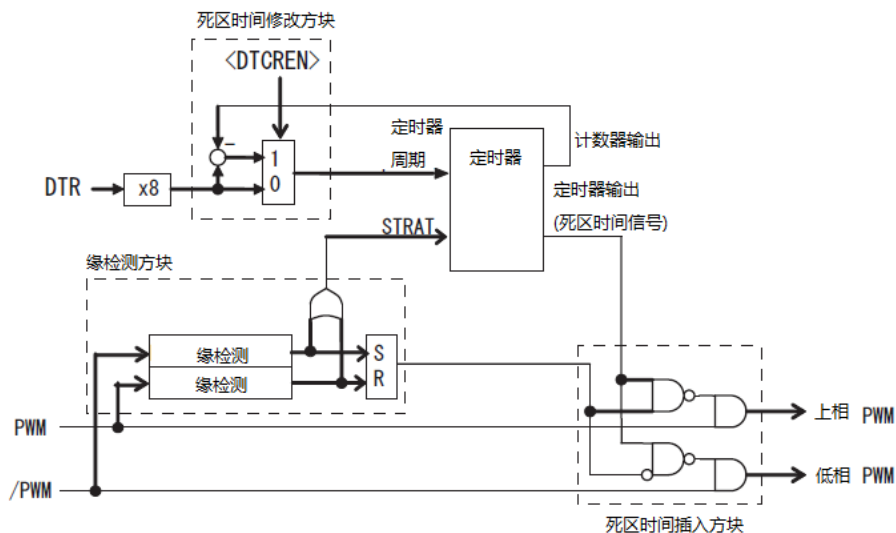


图 17-10 死区时间单元配置

死区时间控制电路由死区时间单元和输出极性切换单元组成。死区时间单元由缘检测方块, 定时器方块, 死区时间插入方块, 和死区时间修改方块组成。(参考 "图 17-10 死区时间单元配置")

关于各 U, V 和 W 相, 上和下相被切换防止短路电路时死区时间单元延迟各相的 ON-时序。死区时间设置到死区时间寄存器 (PMDxDTR<DTR[7:0]>) 因为可以设置 8-位带 $8/f_{sys}$ 分辨率的值(66.7[ns] 在 120[MHz])。

通过 PMD 输出设置寄存器 PMDxDMPOT<POLH> 和 <POLL>, 输出极性切换电路允许上-输出 (UO, VO, WO) 和下-输出 (XO, YO, ZO) 相的极性 (高激活或低激活) 独立设置。

死区时间修改方块中, 上 PWM 或低 PWM 的一个开启-期间为 "0" 时, 如果 PMDxDMCR<DTCREN> 设置为 "1", 或 PWM 延迟时间修改为短路。如果死区-时间周期期间 PWM 信号为关闭, 计数器相的延迟时间在余下的死区-时间 (死区-时间寄存器设置时间 - 开启时间) 为短路。死区时间周期期间上 PWM 变为关闭时, 低 PWM 的延迟时间应该修改为短路。

死区时间周期期间低 PWM 变为关闭时，上 PWM 延迟时间应该修改为短路. 图 17-11 表示死区时间修改.延迟时间在上 PWM 的 100%占空比附近修改且也在 0%占空比附近修改.

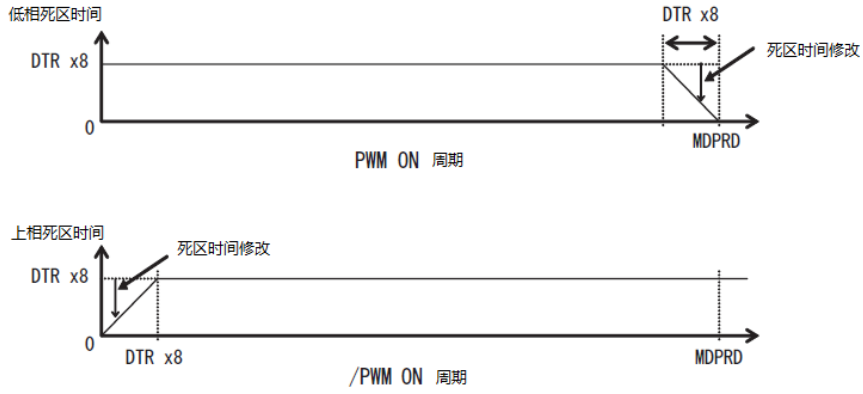


图 17-11 死区 时间修改

17.2.7.1 PMDxDTR (死区 时间 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 0.
7-0	DTR[7:0]	R/W	设置死区时间 0x00 到 0xFF 死区时间值可以通过以下方程式计算. <DTR[7:0]> × 8/fsys (达 17[μs] 在 120[MHz])

17.2.8 同步触发器 生成 电路

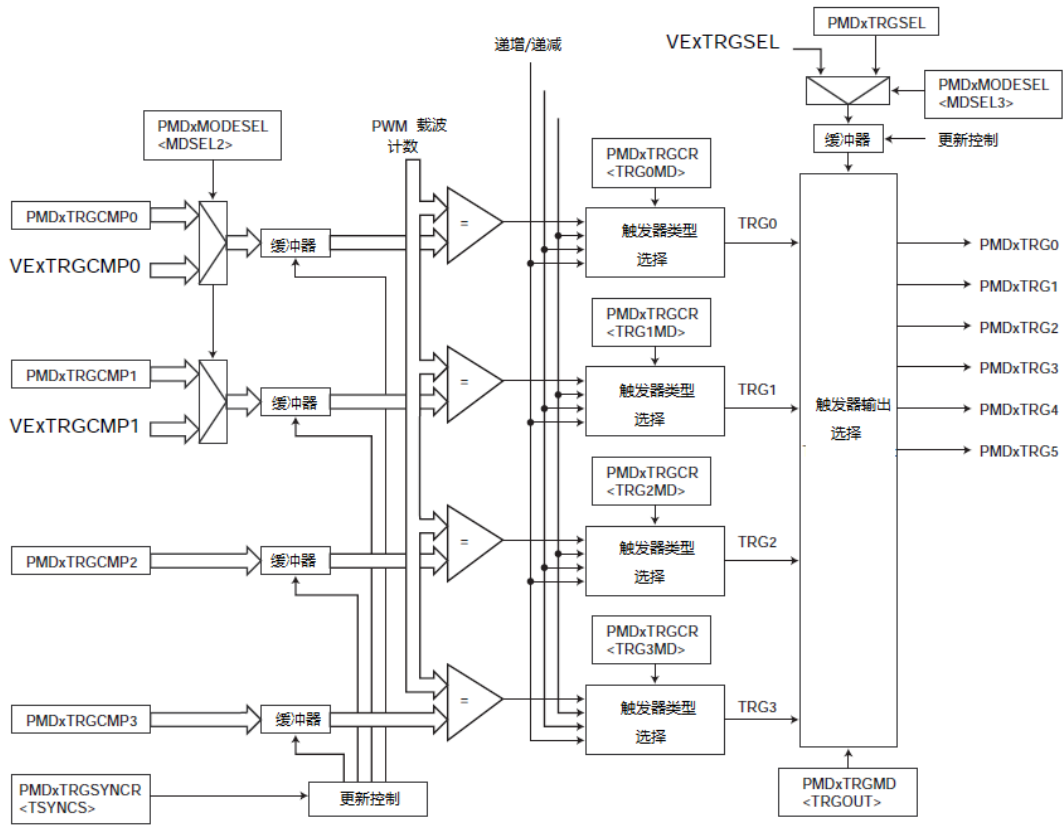


图 17-12 同步触发器生成 电路

同步触发器生成电路生成四触发器信号 (TRG0 到 TRG3) 开始同步 PWM 的 ADC 采样. 如果 VE 模式通过 PMDxMODESEL<MDESEL3>选择, PMDxTRGCMP0 和 PMDxTRGCMP1 变为 VE 寄存器的 VExTRGCMP0, VExTRGCMP1.

触发器时序可以用以下 6 类型选择.

1. 在递增计数 工作 比较-匹配 (注)
2. 在递减计数 工作 比较-匹配 (注)
3. 在递增递减计数 工作 比较-匹配 (注)
4. PWM 载波 峰
5. PWM 载波 底
6. PWM 载波 峰 和 PWM 载波 底

注: PWM 计数器 (PMDxMDCNT<MDCNT[15:0]>) 和 (PMDxTRGCMPn <TRGCMPn[15:0]>) 之间比较-匹配

触发器选择输出模式中期间: $\text{PMDxTRGMD}\langle\text{TRGOUT}\rangle="1"$. 从通过触发器输出选择寄存器 PMDxTRGSEL (VExTRGSEL) 选择的 $\text{PMDxTRG0}\sim 5$ 输出 TRG0 信号. TRG0 设置通过 PMDxTRGCMP0 (VExTRGCMP0) 和 $\text{PMDxTRGCR}\langle\text{TRG0MD}\rangle$ 设置.

缘模式 (锯齿波载波模式) 选择时, 比较-匹配功能 为递增计数. $\text{PMDxTRGMD}\langle\text{EMGTGE}\rangle="1"$ 时, EMG 保护状态下此电路也输出触发器信号.

17.2.8.1 PMDxTRGCMP0 (触发器 比较 寄存器 0)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TRGCMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TRGCMP0							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	TRGCMP0 [15:0]	R/W	触发器输出比较寄存器 <TRGCMP0[15:0]> 应该设置在 1 到 [<MDPRD[15:0]> 设置值 - 1]的范围. PWM 计数值 <MDCNT[15:0]> 匹配 TRGCMP0 中设置值时. 输出 TRG0 . 注: 禁止设置 <TRGCMP0> 为 "0" 且 <TRGCMP0> ≥ <MDPRD[15:0]> 值.

注 1: 要加载比较寄存器中的数据到随后阶段缓冲,通过设置 PMDxMODESEL<MDSEL2> 为"0"选择总线模式 (默认).

注 2: 不要以字节单元写入此寄存器. 如果上 8 位 [15:8] 和低 8 位 [7:0] 分别写入, 工作不可以被确保.

注 3: 因为触发器比较寄存器为双缓冲, PWM 计数器工作期间周期可以被改变.

注 4: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序".

注 5: 读到的值 为第一个缓冲值 (最近的通过总线的测试数据设置).

表 17-5 触发器比较寄存器的缓冲更新时序

<TSYNCS>设置	<TRGnMD>设置	TRGCMPn 寄存器 缓冲更新时序
00	000	立即更新
	001	PWM 载波 峰时更新
	010	PWM 载波 底时更新
	011	PWM 载波峰或 PWM 载波底时更新(注 1)
	1xx	立即更新
01	xxx	PWM 载波 底时更新
10	xxx	PWM 载波 峰时更新
11	xxx	PWM 载波峰或 PWM 载波底时更新 (注 1)

注: x : 忽略

注: 异步更新 PMDxMDEN<PWMEN>="0" 无论设置.

注 1: 锯齿波载波选择时(PMDxMDCR<PMMMD>="0")在载波峰更新.

17.2.8.2 PMDxTRGCMP1 (触发器 比较 寄存器 1)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TRGCMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TRGCMP1							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	TRGCMP1 [15:0]	R/W	触发器输出比较寄存器 <TRGCMP1[15:0]> 应该设置在 1 到 [<MDPRD[15:0]> 设置值 - 1]的范围. PWM 计数器值 <MDCNT[15:0]> 匹配 TRGCMP1 中设置值时. TRG1 输出. 注: 禁止设置 <TRGCMP1> 为 "0"且 <TRGCMP1> ≥ <MDPRD[15:0]> 值.

注 1: 要加载比较寄存器中的数据到随后阶段缓冲,通过设置 PMDxMODESEL<MSEL2> 为"0"选择总线模式 (默认).

注 2: 不要以字节单元写入此寄存器. 如果上 8 位 [15:8] 和低 8 位 [7:0] 分别写入, 工作不可以被确保.

注 3: 因为触发器比较寄存器为双缓冲, PWM 计数器工作期间周期可以被改变.

注 4: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序".

注 5: 读到的值 为第一个缓冲值 (最近的通过总线的测试数据设置).

17.2.8.3 PMDxTRGCMP2 (触发器 比较 寄存器 2)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TRGCMP2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TRGCMP2							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	TRGCMP2 [15:0]	R/W	触发器输出比较寄存器 <TRGCMP2[15:0]> 应该设置在 1 到 [<MDPRD[15:0]> 设置值 - 1]的范围. PWM 计数器值 <MDCNT[15:0]> 匹配 TRGCMP2 中设置值时, TRG2 输出. 注: 禁止设置 <TRGCMP2> 为 "0" 且 <TRGCMP2> ≥ <MDPRD[15:0]> 值.

- 注 1: 不要以字节单元写入此寄存器. 如果上 8 位 [15:8] 和低 8 位 [7:0] 分别写入, 工作不可以被确保.
- 注 2: 因为触发器比较寄存器为双缓冲, PWM 计数器工作期间周期可以被改变 .
- 注 3: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序".
- 注 4: 读到的值 为第一个缓冲值 (最近的通过总线的测试数据设置).

17.2.8.4 PMDxTRGCMP3 (触发器 比较 寄存器 3)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TRGCMP3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TRGCMP3							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 0.
15-0	TRGCMP3 [15:0]	R/W	触发器输出比较寄存器 <TRGCMP3[15:0]> 应该设置在 1 到 [<MDPRD[15:0]> 设置值 - 1]的范围. PWM 计数器值 <MDCNT[15:0]> 匹配 TRGCMP3 中设置值时. TRG3 输出. 注: 禁止设置 <TRGCMP3> 为 "0" 且 <TRGCMP3> ≥ <MDPRD[15:0]> 值.

注 1: 不要以字节单元写入此寄存器. 如果上 8 位 [15:8] 和低 8 位 [7:0] 分别写入, 工作不可以被确保.

注 2: 因为触发器比较寄存器为双缓冲, PWM 计数器工作期间周期可以被改变.

注 3: 关于详细随后阶段缓冲的更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序".

注 4: 读到的值 为第一个缓冲值 (最近的通过总线的测试数据设置).

17.2.8.5 PMDxTRGCR (触发器 控制 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 0.
15	TRG3BE	R/W	<p>PMDxTRGCMP3<TRGCMP3[15:0]> 缓冲的异步更新 此位使能异步更新从 PMDxTRGCMP3 到随后阶段缓冲。 0: 同步 更新 1: 异步 更新 (写入 PMDxTRGCMP3 的值立即被反映.)</p> <p>注:关于详细更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序". 注: PMDxMDEN<PWMEN>="0"时, 异步更新无论设置.</p>
14-12	TRG3MD[2:0]	R/W	<p>PMDxTRGCMP3 模式 设置 此寄存器选择触发器输出的匹配-模式. 000: 触发器 输出 禁止 001: 触发器 输出在 递减-计数 匹配 010: 触发器 输出在 递增-计数 匹配 011: 触发器 输出在 递增递减-计数 匹配 100: 触发器 输出在 PWM 载波 峰 101: 触发器 输出在 PWM 载波 底 110: 触发器 输出在 PWM 载波 峰/底 111: 触发器 输出 禁止</p> <p>注: "0"设置到 PMDxMDCR<PWMMD> (锯齿波)时, 在递增计数器匹配时输出触发器即使 "001" 被设置到 <TRG3MD[2:0]>. 同样, 在 PWM 载波峰输出触发器即使 "101" 设置到<TRG3MD[2:0]>. 注: <TRG3MD[2:0]>="011"时, PMDxTRGCMP3="0x0001" 和 PMDxMDCR<PWMMD>="1" (三角波), 每期间输出一个触发器.</p>
11	TRG2BE	R/W	<p>PMDxTRGCMP2<TRGCMP2[15:0]> 缓冲的异步更新 此位使能异步更新从 PMDxTRGCMP2 到随后阶段缓冲。 0: 同步 更新 1: 异步 更新 (写入 PMDxTRGCMP2 的值立即被反映.)</p> <p>注:关于详细更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序". 注: PMDxMDEN<PWMEN>="0"时, 异步更新无论设置.</p>

位	位符	类型	功能
10-8	TRG2MD[2:0]	R/W	<p>PMDxTRGCMP2 模式 设置</p> <p>此寄存器选择触发器 输出的匹配-模式.</p> <p>000: 触发器 输出 禁止</p> <p>001: 触发器 输出在递减-计数 匹配</p> <p>010: 触发器 输出在递增-计数 匹配</p> <p>011: 触发器 输出在递增递减-计数 匹配</p> <p>100: 触发器 输出在 PWM 载波 峰</p> <p>101: 触发器 输出在 PWM 载波 底</p> <p>110: 触发器 输出在 PWM 载波 峰/底</p> <p>111: 触发器 输出 禁止</p> <p>注: "0"设置到 PMDxMDCR<PWWMD> (锯齿波)时,在递增计数器匹配时输出触发器即使 "001" 被设置到 <TRG2MD[2:0]>. 同样,在 PWM 载波峰输出触发器即使 "101" 设置到<TRG2MD[2:0]>.</p> <p>注: <TRG2MD[2:0]>="011"时, PMDxTRGCMP2="0x0001" 和 PMDxMDCR<PWWMD>="1" (三角波), 每期间输出一个触发器.</p>
7	TRG1BE	R/W	<p>PMDxTRGCMP1<TRGCMP1[15:0]> 缓冲的异步 更新</p> <p>此位使能异步更新从 PMDxTRGCMP1 到随后阶段缓冲.</p> <p>0: 同步 更新</p> <p>1: 异步 更新 (写入 PMDxTRGCMP1 的值立即被反映.)</p> <p>注:关于详细更新时序,参考 "表 17-5 触发器比较寄存器的缓冲更新时序".</p> <p>注: PMDxMDEN<PWWMD>="0"时,异步更新无论设置.</p>
6-4	TRG1MD[2:0]	R/W	<p>PMDxTRGCMP1 模式 设置</p> <p>此寄存器选择触发器输出的匹配-模式 .</p> <p>000: 触发器 输出 禁止</p> <p>001: 触发器 输出在递减-计数 匹配</p> <p>010: 触发器 输出在递增-计数 匹配</p> <p>011: 触发器 输出在递增递减-计数 匹配</p> <p>100: 触发器 输出在 PWM 载波 峰</p> <p>101: 触发器 输出在 PWM 载波 底</p> <p>110: 触发器 输出在 PWM 载波 峰/底</p> <p>111: 触发器 输出 禁止</p> <p>注: "0"设置到 PMDxMDCR<PWWMD> (锯齿波)时,在递增计数器匹配时输出触发器即使 "001" 被设置到 <TRG1MD[2:0]>. 同样,在 PWM 载波峰输出触发器即使 "101" 设置到<TRG1MD[2:0]>.</p> <p>注: <TRG1MD[2:0]>="011"时, PMDxTRGCMP1="0x0001" 和 PMDxMDCR<PWWMD>="1" (三角波), 每期间输出一个触发器.</p>
3	TRG0BE	R/W	<p>PMDxTRGCMP0<TRGCMP0[15:0]> 缓冲的异步 更新</p> <p>此位使能异步更新从 PMDxTRGCMP0 到随后阶段缓冲.</p> <p>0: 同步 更新</p> <p>1: 异步 更新 (写入 PMDxTRGCMP0 的值立即被反映.)</p> <p>注:关于详细更新时序,参考 "表 17-5 触发器比较寄存器的缓冲更新时序".</p> <p>注: PMDxMDEN<PWWMD>="0"时,异步更新无论设置.</p>

位	位符	类型	功能
2-0	TRG0MD[2:0]	R/W	<p>PMDxTRGCMP0 模式 设置</p> <p>此寄存器选择触发器输出的匹配-模式 .</p> <p>000: 触发器 输出 禁止</p> <p>001: 触发器 输出在递减-计数 匹配</p> <p>010: 触发器 输出在递增-计数 匹配</p> <p>011: 触发器 输出在递增递减-计数 匹配</p> <p>100: 触发器 输出在 PWM 载波 峰</p> <p>101: 触发器 输出在 PWM 载波 底</p> <p>110: 触发器 输出在 PWM 载波 峰/底</p> <p>111: 触发器 输出 禁止</p> <p>注: "0"设置到 PMDxMDCR<PWWMD> (锯齿波)时, 在递增计数器匹配时输出触发器即使 "001" 被设置到 <TRG0MD[2:0]>. 同样, 在 PWM 载波峰输出触发器即使 "101" 设置到<TRG0MD[2:0]>.</p> <p>注: <TRG0MD[2:0]>="011"时, PMDxTRGCMP0="0x0001" 和 PMDxMDCR<PWWMD>="1" (三角波), 每期间输出一个触发器.</p>

17.2.8.6 PMDxTRGSYNCR (触发器 更新 时序 设置 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作 0.
1-0	TSYNCS	R/W	触发器比较寄存器的缓冲的更新时序设置. 00: PWM 载波 底 或 PWM 载波 峰被设置给各触发器通过设置 PMDxTRGCR<TRGxMD>. 01: PWM 载波 底 10: PWM 载波 峰 11: PWM 载波 峰和 PWM 载波 底 注:关于详细更新时序, 参考 "表 17-5 触发器比较寄存器的缓冲更新时序". 注: PMDxMDEN<PWMEN>="0"时, 异步更新无论设置.

17.2.8.7 PMDxTRGMD (触发器 输出 模式 设置 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作 0.
1	TRGOUT	R/W	触发器 输出 模式 0: 固定的触发器输出 1: 可变的触发器输出 固定的触发器 输出选择时, 从 PMDxTRG0 到 PMDxTRG3 的触发器输出输出通过分别匹配 <TRGCMP0[15:0]> 到 <TRGCMP3[15:0]> 生成的触发器信号. PMDxTRG4 和 PMDxTRG5 不输出触发器信号. 可变的 触发器 输出选择时, 输出<TRGCMP0[15:0]> 的输出信号到 从 PMDxTRG0 到 PMDxTRG5 触发器输出的一个. 触发器输出信号通过触发器输出选择寄存器选择. 注: 参考 "表 17-6 触发器输出类型" 可变的触发器输出选择时(<TRGOUT>="1").
0	EMGTGE	R/W	EMG 保护状态下输出使能 0: 保护 状态下禁止 触发器 输出 1: 保护 状态下使能 触发器 输出 EMG 保护状态下此位使能或禁止触发器输出.

表 17-6 触发器 输出 类型

<TRGOUT> 设置	比较 寄存器	<TRGSEL[2:0]> 设置	触发器 输出
<TRGOUT>="0"	PMDxTRGCMP0	x	PMDxTRG0
	PMDxTRGCMP1		PMDxTRG1
	PMDxTRGCMP2		PMDxTRG2
	PMDxTRGCMP3		PMDxTRG3
<TRGOUT>="1"	PMDxTRGCMP0	0	PMDxTRG0
		1	PMDxTRG1
		2	PMDxTRG2
		3	PMDxTRG3
		4	PMDxTRG4
	PMDxTRGCMP1	5	PMDxTRG5
		x	无触发器 输出
		x	无触发器 输出
PMDxTRGCMP2	x	无触发器 输出	
PMDxTRGCMP3	x	无触发器 输出	

17.2.8.8 PMDxTRGSEL (触发器 输出 选择 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作 0.
2-0	TRGSEL[2:0]	R/W	触发器 输出 选择 000: 输出 从 PMDxTRG0 001: 输出 从 PMDxTRG1 010: 输出 从 PMDxTRG2 011: 输出 从 PMDxTRG3 100: 输出 从 PMDxTRG4 101: 输出 从 PMDxTRG5 110: 无触发器 输出 111: 无触发器 输出 可变的触发器输出模式选择时(PMDxTRGMD<TRGOUT>="1"), 此域有效. 且输出触发器可以通过设置 PMDxTRGCMP0 寄存器选择. (参考表 17-6.)

注 1: 要通过总线加载比较寄存器更新的数据到随后阶段缓冲, 设置总线模式 (默认)通过写入 "0" 到 PMDxMODESEL<MSEL3>.

注 2: 因为触发器输出选择寄存器为双缓冲, PWM 计数器工作期间周期可以被改变.

注 3: 随后阶段缓冲的更新时序相同于比较寄存器 (PMDxCMPU/V/W).

注 4: PMD 禁止 (PMDxMDCR<PWMEN>="0")时, 异步更新.

18. 向量引擎 (A-VE)

18.1 概要

18.1.1 特征

1. 固定-点格式数据中，向量控制(协调转换, 相转换和 SIN/COS 计算) 基础任务被执行。
→不需要管理小数点对准的软件处理.
2. 电机控制电路(PMD)的接口处理 (输出控制, 触发器生成和输入处理) 和 AD 转换器 (ADC) 包含.
 - 在 PMD 中, 转换计算结果从固定-点格式到数据格式可用.
 - 用 PMD 和 ADC 生成相互工作的时序数据.
 - 转换 AD 转换 结果到固定-点 格式.
3. 电流, 电压和转速值用其最大值正常化且被计算.
小数点数被用在固定-点格式.
4. PI 控制包含电流控制.
5. 相插值被执行生成转速.

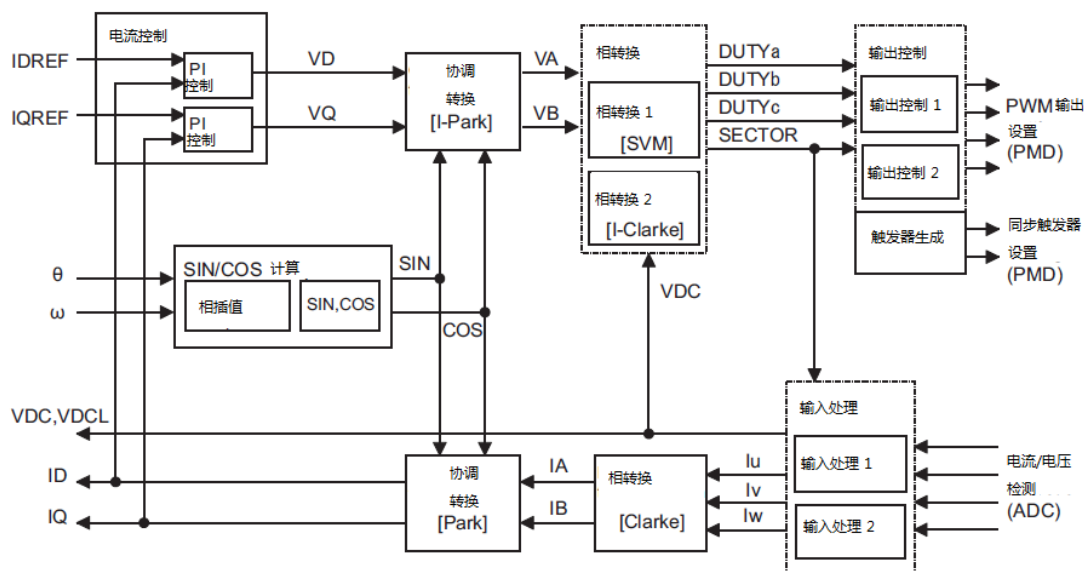


图 18-1 向量控制电路方块图

18.1.2 主要 功能

1. 空间向量调制和反转 Clarke 转换被用作 2-相-到-3-相转换. 空间向量调制支持 2-相调制和 3-相调制.
2. 触发器生成功能可以生成无传感器电流检测的 ADC 采样时序.
此采样时序可以被用在 1-电阻电流检测法.
3. 电流控制中, PI 控制独立对应 d-轴 和 q-轴. 也可以直接设置 d-轴和 q-轴电压寄存器.
控制 PI 控制的结果
独立控制 d-轴和 q-轴.
使用由 d-轴和 q-轴组成的电压标量值控制输出.
4. SIN/COS 计算用使用系列扩展的近似执行. 相信息可以直接从转速被指定或被计算通过使用相插值.
作为相信息, 直接设置或生成 PWM 周期上的转速的向插值可以被用.相信息执行时, 裁剪可以被用在指定的相.
输出控制下, 死区时间补偿和 PWM 输出控制可用.

注: 向量引擎被用时需要设置电机控制电路和 ADC.

- 向量引擎被用时,用电机控制电路 (PMD)的模式选择寄存器 (PMDxMODESEL) 设置 VE 模式.
- 向量引擎被用时, 根据同步触发器, AD 转换器选择电机控制电路 (PMD)的程序 (触发器使能, AIN 选择, 和结果寄存器选择).

18.2 配置

图 18-2 表示向量引擎配置.

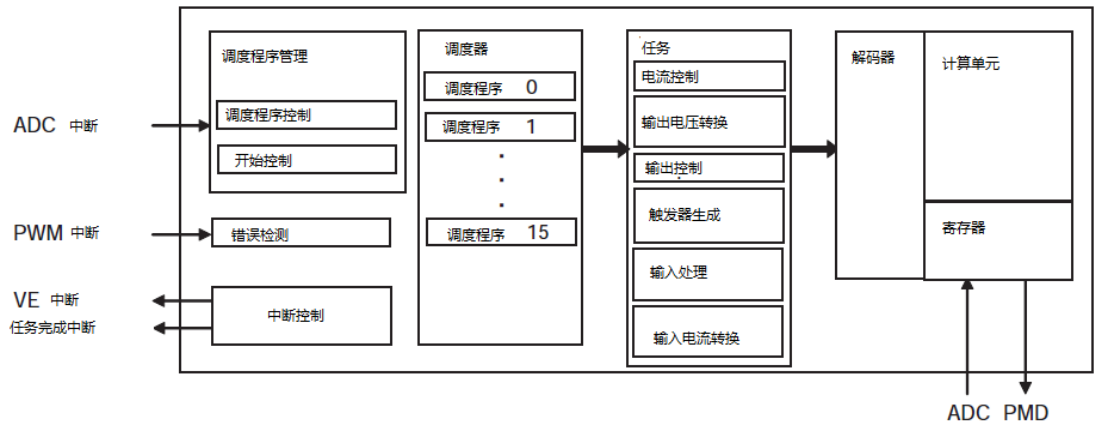


图 18-2 向量引擎配置

18.2.1 向量引擎, 电机控制电路, 和 AD 转换器间的配合

如图 18-3 和图 18-4 所示, 向量引擎允许直接交换电机控制电路 (PMD) 和 AD 转换器 (ADC) 之间数据.

用 PMD_xMODESEL 寄存器, 电机控制电路 (PMD) 的 PMD_xCMPU, PMD_xCMPV, PMD_xCMPW, PMD_xMDOU, PMD_xTRGCMP0, PMD_xTRGCMP1 和 PMD_xTRGSEL 寄存器设置为 VE 模式时, 这些寄存器切换到向量引擎寄存器的 VExCMPU, VExCMPV, VExCMPW, VExOUTCR, VExTRGCMP0, VExTRGCMP1 和 VExTRGSEL. 此时, 这些向量引擎寄存器不可以由 CPU 通过电机控制电路 (PMD) 的对应的寄存器控制. 这些寄存器可以通过向量引擎写入. 对其他 PMD 寄存器没有读/写入限制.

输入处理任务中向量引擎可以读 AD 转换器 (ADC) 的转换结果寄存器 (AD_xREG0, AD_xREG1, AD_xREG2, AD_xREG3) 的值. 也可以从两单元 ADCs 读转换结果且支持 2-相电流同步采样. 读转换结果时, 应该读根据 PMD 同步触发器编程的相信息.

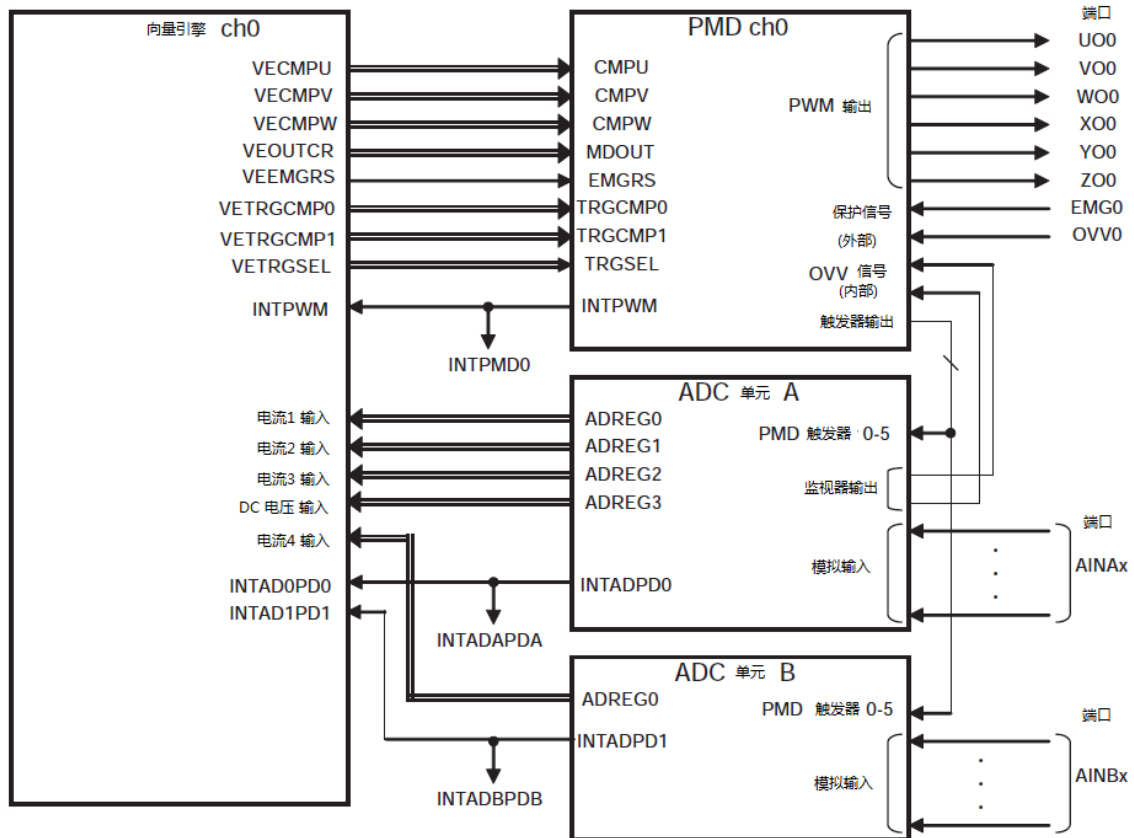


图 18-3 向量引擎, PMD 和 ADC (通道 0) 之间互相关系方块图

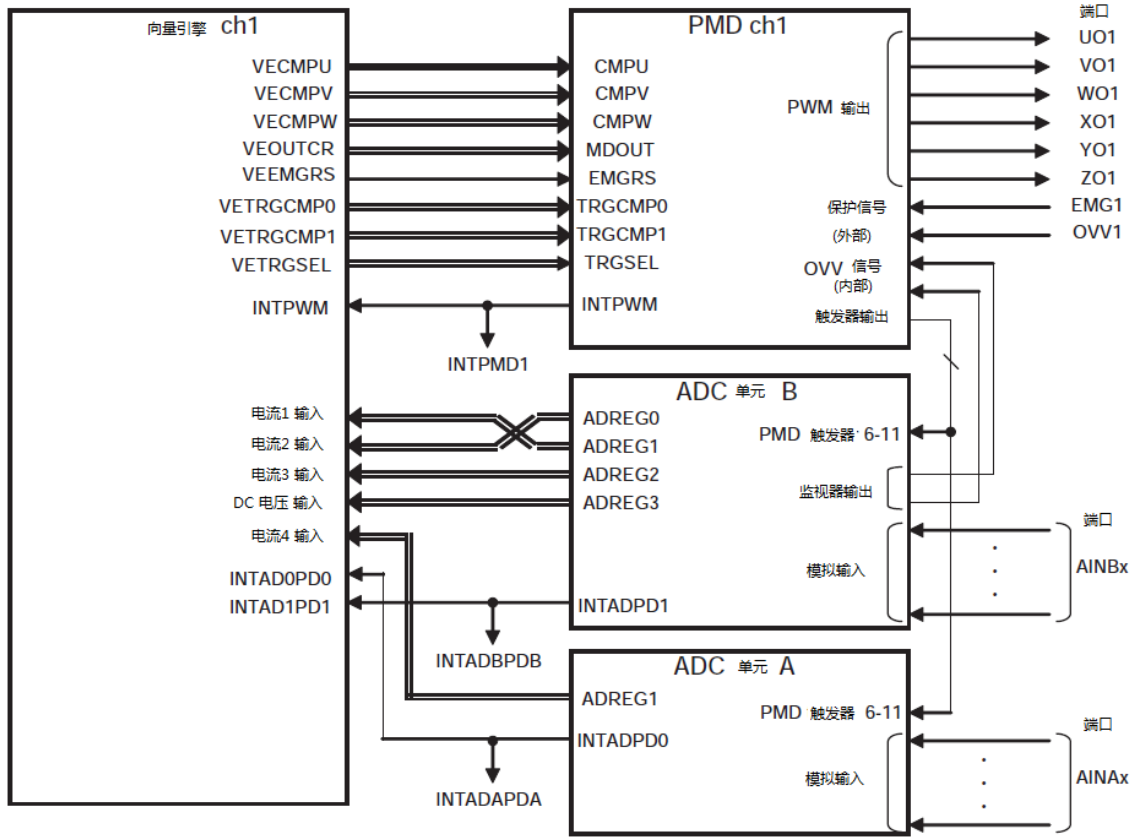


图 18-4 向量 引擎, PMD 和 ADC (通道 1) 之间互相关系方块图

18.3 寄存器

向量引擎有 VE 控制寄存器和专用寄存器.

- VE 控制 寄存器
 - 向量引擎控制寄存器和暂时寄存器.
- 专用 寄存器
 - 计算数据寄存器和计算控制寄存器.

18.3.1 寄存器表

VE 控制 寄存器

寄存器名		地址
向量引擎工作 使能/禁止	VExEN	R/W 0x0000
CPU 开始触发器 选择	VExCPURUNTRG	W 0x0004
任务 选择	VExTASKAPP	R/W 0x0008
工作调度程序 选择	VExACTSCH	R/W 0x000C
工作调度程序 重复数 选择	VExREPTIME	R/W 0x0010
开始触发器 模式 设置	VExTRGMODE	R/W 0x0014
错误中断 使能/禁止 设置	VExERRINTEN	R/W 0x0018
向量引擎强制中止	VExCOMPEND	W 0x001C
错误检测	VExERRDET	R 0x0020
处理中调度程序标志/处理中 任务	VExSCHTASKRUN	R 0x0024
保留	-	R 0x0028
暂时 0	VExTMPREG0	R/W 0x002C
暂时 1	VExTMPREG1	R/W 0x0030
暂时 2	VExTMPREG2	R/W 0x0034
暂时 3	VExTMPREG3	R/W 0x0038
暂时 4	VExTMPREG4	R/W 0x003C
暂时 5	VExTMPREG5	R/W 0x0040
保留	-	R 0x0044

注:不要进入描述为"保留"的地址.

专用 寄存器

寄存器名		地址
异常决定结果储存	VExMCTLF	R/W 0x0044
任务控制 模式	VExMODE	R/W 0x0048
流 控制	VExFMODE	R/W 0x004C
PWM 周期率 (PWM 期间 [s] × 最大速度 (注 1) × 2 ¹⁶) 设置	VExTPWM	R/W 0x0050
转速 (速度 [Hz] ÷ 最大速度 (注 1) × 2 ¹⁵) 设置	VExOMEGA	R/W 0x0054
电机相 (电机相 [deg]/360 × 2 ¹⁶) 设置	VExTHETA	R/W 0x0058
d-轴 参考值 (电流 [A] + 最大电流(注 2) × 2 ¹⁵) 设置	VExIDREF	R/W 0x005C
q-轴 参考值 (电流 [A] + 最大电流 (注 2) × 2 ¹⁵) 设置	VExIQREF	R/W 0x0060
d-轴 电压 (电压 [V] ÷ 最大电压 (注 3) × 2 ³¹) 设置	VExVD	R/W 0x0064
q-轴 电压 (电压 [V] ÷ 最大电压 (注 3) × 2 ³¹) 设置	VExVQ	R/W 0x0068

专用 寄存器

寄存器名			地址
d-轴 电流 PI 控制的积分系数控制 设置	VExCIDKI	R/W	0x006C
d-轴 电流 PI 控制的比例系数 设置	VExCIDKP	R/W	0x0070
q-轴 电流 PI 控制的积分系数 设置	VExCIQKI	R/W	0x0074
q-轴 电流 PI 控制的比例系数 设置	VExCIQKP	R/W	0x0078
d-轴 电压储存的积分组成 (VDI 的上 32 位)	VExVDIH	R/W	0x007C
d-轴 电压储存的积分组成 (VDI 的低 32 位)	VExVDILH	R/W	0x0080
q-轴 电压储存的积分组成 (VQI 的上 32 位)	VExVQIH	R/W	0x0084
q-轴 电压储存的积分组成 (VQI 的低 32 位)	VExVQILH	R/W	0x0088
切换 PWM 速度	VExFPWMCHG	R/W	0x008C
PWM 周期设置 (用 PMD 的 PWM 周期设置设置理想的值).	VExMDPRD	R/W	0x0090
最小脉宽 设置	VExMINPLS	R/W	0x0094
同步 触发器 修改值 设置	VExTRGCRC	R/W	0x0098
DC 电源 电压 (电压 [V] ÷ 最大值 电压 (注 3) × 2 ¹⁵)	VExVDCL	R/W	0x009C
输出 转换的 THETA 上 Cosine 值 (Q15 数据)	VExCOS	R/W	0x00A0
输出 转换的 THETA 上 Sine 值 (Q15 数据)	VExSIN	R/W	0x00A4
输入处理之前的 cosine 值 (Q15 数据)	VExCOSM	R/W	0x00A8
输入处理中之前的 sine 值 (Q15 数据)	VExSINM	R/W	0x00AC
扇形 信息	VExSECTOR	R/W	0x00B0
之前的扇形信息	VExSECTORM	R/W	0x00B4
a-相 零-电流 (注 4)	VExIAO	R/W	0x00B8
b-相 零-电流 (注 4)	VExIBO	R/W	0x00BC
c-相 零-电流 (注 4)	VExICO	R/W	0x00C0
a-相 电流的 AD 转换 结果(注 4)	VExIAADC	R/W	0x00C4
b-相 电流的 AD 转换 结果 (注 4)	VExIBADC	R/W	0x00C8
c-相 电流的 AD 转换 结果 (注 4)	VExICADC	R/W	0x00CC
电源 电压 (电压 [V] ÷ 最大值 电压 (注 3) × 2 ¹⁵)	VExVDC	R/W	0x00D0
d-轴 电流 (电流 [A] ÷ 最大值 电流 (注 2) × 2 ³¹)	VExID	R/W	0x00D4
q-轴 电流 (电流 [A] ÷ 最大值 电流 (注 2) × 2 ³¹)	VExIQ	R/W	0x00D8
保留	-	R/W	0x0174
AD 转换时间 设置	VExTADC	R/W	0x0178
PMD 控制: U-相 PWM 脉宽 设置	VExCMPU	R/W	0x017C
PMD 控制: V-相 PWM 脉宽 设置	VExCMPV	R/W	0x0180
PMD 控制: W-相 PWM 脉宽 设置	VExCMPV	R/W	0x0184
PMD 控制: 6-相 输出 控制	VExOUTCR	R/W	0x0188
PMD 控制: 触发器 时序 设置 (TRGCMP0)	VExTRGCMP0	R/W	0x018C
PMD 控制: 触发器 时序 设置 (TRGCMP1)	VExTRGCMP1	R/W	0x0190
PMD 控制: 同步 触发器 选择	VExTRGSEL	R/W	0x0194
PMD 控制: EMG 保护 释放	VExEMGRS	W	0x0198
PI 控制 输出 极限	VExPIOLIM	R/W	0x01BC
PI 控制 d-轴 系数范围 设置	VExCIDKG	R/W	0x01C0
PI 控制 q-轴 系数范围 设置	VExCIQKG	R/W	0x01C4
电压 标量极限	VExVSLIM	R/W	0x01C8
电压 标量	VExVDQ	R/W	0x01CC
偏向 角	VExDELTA	R/W	0x01D0
电机链接磁通	VExCPHI	R/W	0x01D4
电机 q-轴 电感	VExCLD	R/W	0x01D8
电机 d-轴 电感	VExCLQ	R/W	0x01DC
电机 电阻值	VExCR	R/W	0x01E0
电机 磁通范围 设置	VExCPHIG	R/W	0x01E4

专用 寄存器

寄存器名			地址
电机 电感范围 设置	VExCLG	R/W	0x01E8
电机 电阻范围 设置	VExCRG	R/W	0x01EC
d-轴 电压非-干扰 控制	VExVDE	R/W	0x01F0
q-轴 电压非-干扰 控制	VExVQE	R/W	0x01F4
死区时间补偿总和	VExDTC	R/W	0x01F8
电流识别滞后 宽	VExHYS	R/W	0x01FC
死区时间补偿 控制/状态	VExDTCS	R/W	0x0200
PWM 上-限 设置	VExPWMMAX	R/W	0x0204
PWM 下-限 设置	VExPWMMIN	R/W	0x0208
被裁剪相值 设置	VExTHTCLP	R/W	0x020C

注 1: 不要进入描述为"保留"的地址.

注 2: 最大速度: 可控或可工作最大转数[Hz]

注 3: 最大电流: AD 转换结果变为 1 LSB 时相电流的变化 ($[A] \times 2^{11}$).

注 4: 最大电压: AD 转换改变结果时电源电压 (VDC)变化 ($[V] \times 2^{12}$).

注 5: AD 转换结果储存在 16-位寄存器的上 12 位 .

18.3.2 VE 控制 寄存器

18.3.2.1 VExEN (向量引擎工作 使能/禁止 寄存器)

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

位	位 符	类型	功能
31-2	-	R	读作"0".
1	VEIDLEN	R/W	IDLE 模式下选择时钟工作. 0: 停止 1: 工作
0	VEEN	R/W	选择向量引擎工作. 0: 禁止 1: 使能

注: 向量引擎禁止 (VExEN = "0")时,不允许进入向量引擎其他寄存器.

18.3.2.2 VExCPURUNTRG (CPU 开始 触发器 选择 寄存器)

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

位	位符	类型	功能
31-1	-	R	读作"0".
0	VCPURT	W	通过软件开始向量引擎. 0: - 1: 开始工作 从配置为 VExTASKAPP<VTASKA>的任务工作开始. 开始工作前指定 VExTASKAPP, VExACTSCH, 和 VExREPTIME.

注 1: "1"写入此位时,在下一个周期被清除. 此位始终读作"0".

注 2: 另一个调度程序被执行时如果新调度程序和任开始工作, 再用 VExCPURUNTRG 寄存器开始前应该用 VExCOMPEND 寄存器终止向量引擎.

18.3.2.3 VExTASKAPP (任务选择寄存器)

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

位	位符	类型	功能
31-12	-	R	读作"0".
11-8	VITASK [3:0]	R/W	选择任务完成中断发生时的时序. 0x0: 输出控制 1 (任务数 0) 0x1: 触发器生成 (任务数 1) 0x2: 输入处理 1 (任务数 2) 0x3: 输入相转换(任务数 3) 0x4: 输入协调轴转换(任务数 4) 0x5: 电流控制 (任务数 5) 0x6: SIN/COS 计算 (任务数 6) 0x7: 输出协调轴转换(任务数 7) 0x8: 输出相转换 1[SVM] (任务数 8) 0x9: 输出控制 2 (任务数 9) 0xA: 输入处理 2 (任务数 10) 0xB: 输出相转换 2[I-Clarke] (任务数 11) 0xC: ATAN2 (任务数 12) 0xD: SQRT (任务数 13) 0xE,0xF: 保留
7-4	-	R	读作"0".
3-0	VTASK [3:0]	R/W	选择开始任务. 0x0: 输出控制 1 (任务数 0) 0x1: 触发器生成 (任务数 1) 0x2: 输入处理 1 (任务数 2) 0x3: 输入相转换(任务数 3) 0x4: 输入协调轴转换(任务数 4) 0x5: 电流控制 (任务数 5) 0x6: SIN/COS 计算 (任务数 6) 0x7: 输出协调轴转换(任务数 7) 0x8: 输出相转换 1[SVM] (任务数 8) 0x9: 输出控制 2 (任务数 9) 0xA: 输入处理 2 (任务数 10) 0xB: 输出相转换 2[I-Clarke] (任务数 11) 0xC: ATAN2 (任务数 12) 0xD: SQRT (任务数 13) 0xE,0xF: 保留 选择通过软件开始的任務.

注:仅可以指定那些包括调度程序中的任务.

18.3.2.4 VExACTSCH (工作 调度程序 选择 寄存器)

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

位	位 符	类型	功能
31-4	-	R	读作 "0".
3-0	VACT[3:0]	R/W	选择独立任务执行或工作调度程序. 0x0: 执行独立 任务. 0x1: 调度程序 1 0x2: 调度程序 2 0x3: 调度程序 3 0x4: 调度程序 4 0x5: 调度程序 5 0x6: 调度程序 6 0x7: 调度程序 7 0x8: 调度程序 8 0x9: 调度程序 9 0xA: 调度程序 10 0xB: 调度程序 11 0xC: 调度程序 12 0xD: 调度程序 13 0xE: 调度程序 14 0xF: 调度程序 15 关于详细, 参考 "表 18-4 各调度程序中执行任务".

18.3.2.5 VExREPTIME (工作 调度程序 重复 数 选择 寄存器)

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

位	位 符	类型	功能
31-4	-	R	读作"0".
3-0	VREP[3:0]	R/W	选择工作调度程序的重复数. 0: 无执行的调度程序 工作. 1-15: 重复被指定调度程序工作时间数.

注:此位设置为"0"时,不要开始调度程序工作.

18.3.2.6 VExTRGMODE (开始 触发器 模式 设置 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作 "0".
1-0	VTRG[1:0]	RW	选择使用 AD 转换完成中断的输入处理开始条件. 00: 中断忽略. 01: 输入处理通过 INTADAPDA (ADCA PMD0 触发器 同步 转换完成) 中断开始. 10: 输入处理通过 INTADBPDB (ADCB PMD1 触发器 同步 转换完成) 中断开始 11: 保留

18.3.2.7 VExERRINTEN (错误 中断 使能/禁止 设置 寄存器)

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

位	位 符	类型	功能
31-3	-	R	读作"0".
2	INTTEN	R/W	控制任务完成中断. 0: 禁止 1: 使能
1	-	R	读作"0".
0	VERREN	R/W	控制在错误检测的中断. 0: 禁止 1: 使能 工作调度程序被执行 (不包括激活触发器等待) 时如果 PWM 中断被检测, 设置"1" 为错误标志.

18.3.2.8 VExCOMPEND (VE 强制 中止 寄存器)

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

位	位符	类型	功能
31-1	-	R	读作 "0".
0	VCEND	W	处理中调度程序被强制终止. 0: - 1: 停止 如果 "1"写入此位, 此位在下一个周期清除. 始终读作"0".

18.3.2.9 VExERRDET (错误 检测 寄存器)

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

位	位 符	类型	功能
31-1	-	R	读作"0".
0	VERRD	R	此位指错误标志. 0:错误不被检测. 1:错误被检测. 工作调度程序被执行时(除开始触发器等待), 如果 PWM 中断被检测, 设置"1" 到此位. 此位在读时清除.

18.3.2.10 VExSCHTASKRUN (调度程序 工作状态/处理中任务数 寄存器)

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

位	位符	类型	功能
31-5	-	R	读作"0".
4-1	VRTASK [3:0]	R	指处理中任务数. 0x0: 输出 控制 1 (任务数 0) 0x1: 触发器 生成 (任务数 1) 0x2: 输入 处理 1 (任务数 2) 0x3: 输入 相转换(任务数 3) 0x4: 输入 协调 轴转换(任务数 4) 0x5: 电流 控制 (任务数 5) 0x6: SIN/COS 计算 (任务数 6) 0x7: 输出 协调 轴转换(任务数 7) 0x8: 输出 相转换 1 [SVM] (任务数 8) 0x9: 输出 控制 2 (任务数 9) 0xA: 输入 处理 2 (任务数 10) 0xB: 输出 相转换 2[I-Clarke] (任务数 11) 0xC: ATAN2 (任务数 12) 0xD: SQRT (任务数 13) 0xE: 保留 0xF: 保留
0	VRSCH	R	指调度程序执行状态. 0: 停止 1: 正被执行

18.3.2.11 VExTMPREG0 到 5 (暂时 寄存器)

VExTMPREG0

	31	30	29	28	27	26	25	24
位 符	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	TMPREG0							
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	TMPREG0[31:0]	R/W	暂时 寄存器 0

VExTMPREG1

	31	30	29	28	27	26	25	24
位 符	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	TMPREG1							
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-0	TMPREG1[31:0]	R/W	暂时 寄存器 1

VExTMPREG2

	31	30	29	28	27	26	25	24
位符	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TMPREG2							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-0	TMPREG2[31:0]	R/W	暂时寄存器 2

VExTMPREG3

	31	30	29	28	27	26	25	24
位符	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TMPREG3							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-0	TMPREG3[31:0]	R/W	暂时寄存器 3

VExTMPREG4

	31	30	29	28	27	26	25	24
位符	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TMPREG4							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-0	TMPREG4[31:0]	R/W	暂时寄存器 4

VExTMPREG5

	31	30	29	28	27	26	25	24
位符	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	TMPREG5							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-0	TMPREG5[31:0]	R/W	暂时寄存器 5

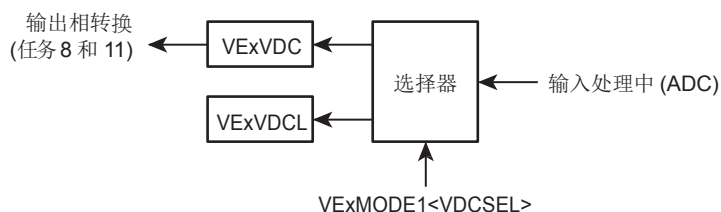
18.3.3 专用 寄存器

18.3.3.1 VExMODE (任务 控制 模式 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	IPDEN	PMDDTCCN	PWMFLEN	PWMBLEN	NICEN	T5ECEN	AWUMD	
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	CLPEN	ATANMD		VDCSEL	OCRMD		ZIEN	PVIEN
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15	IPDEN	R/W	禁止/使能任务 2 和任务 10 的电流极性决定. 0: 禁止 1: 使能 参考"18.4.2.6 输入处理".
14	PMDDTCCN	R/W	任务 0 和任务 9 的死区时间补偿执行时对应 PMD 死区时间修改控制的设置 . 0: PMD 死区 时间修改禁止. 1: PMD 死区 时间修改使能. 参考 "18.4.2.4 输出 控制".
13	PWMFLEN	R/W	任务 0 和任务 9 的上-极限设置时禁止/使能 100% 设置的占空比. 0: 100% 设置占空比禁止. 1: 100% 设置占空比使能. 参考"18.4.2.4 输出 控制".
12	PWMBLEN	R/W	任务 0 和任务 9 的下-限设置时禁止/使能 0% 设置的占空比. 0: 0% 设置占空比禁止. 1: 0% 设置占空比使能. 参考 "18.4.2.4 输出控制".
11	NICEN	R/W	禁止/使能任务 5 的非-干扰 控制. 0: 禁止 1: 使能 参考"18.4.2.1 电流控制 (任务 5)".
10	T5ECEN	R/W	禁止/使能任务 5 的扩展控制 (非-干扰控制和电压标量极限). 0: 禁止 1: 使能 参考 "18.4.2.1 电流控制 (任务 5)".
9-8	AWUMD[1:0]	R/W	PI 控制输出极限执行时指定任务 5 的抗-饱和 (AWU) 控制. 00: AWU 控制禁止. 01: 替代结果从极限总和 + 4 到 积分期限. 10: 替代结果从极限总和 + 2 到 积分期限. 11: 替代极限总和到 积分期限. 参考"18.4.2.1 电流控制 (任务 5)".
7	CLPEN	R/W	相插值执行时禁止/使能任务 6 的相裁剪控制. 0: 裁剪禁止. 1: 裁剪使能. 参考"18.4.2.2 SIN/COS 计算 (任务 6)".

位	位符	类型	功能
6-5	ATANMD[1:0]	R/W	指定任务 4 的 ATAN 计算控制。 0x: 计算禁止。 10: 计算电流 向量 d-q 协调的偏向角。 11: 计算诱导电压 向量 d-q 协调的偏向角。 参考"18.4.2.7 输入电流转换 (相转换/协调 轴转换)".
4	VDCSEL	R/W	选择任务 2 和 10 的电源电压储存寄存器。 0: 储存在 VExVDC。 1: 储存在 VExVDCL。 参考"图 18-5 VExVDC/VExVDCL 储存寄存器".
3-2	OCRMD[1:0]	R/W	指定任务 0 和任务 9 的输出控制。 00: 输出禁止。 01: 输出使能。 10: 短路 电路 刹车 (上-相输出关闭; 下-相输出 ON.) 11: EMG 恢复(输出关闭.) 参考"18.4.2.4 输出 控制".
1	ZIEN	R/W	指定任务 2 的零-电流检测控制。 0: 正常电流 检测 1: 零-电流 检测 参考"18.4.2.6 输入处理".
0	PVIEN	R/W	禁止/使能任务 6 的相插值控制。 0: 禁止 1: 使能 参考"18.4.2.2 SIN/COS 计算 (任务 6)".



注)修改通过 VExVDC 寄存器控制的电源电压时,选择 VExVDCL 寄存器为储存场所,且在 VExDVC 寄存器中设置修改的值.

图 18-5 VExVDC/VExVDCL 储存 寄存器

18.3.3.2 VExFMODE (流动 控制 寄存器)

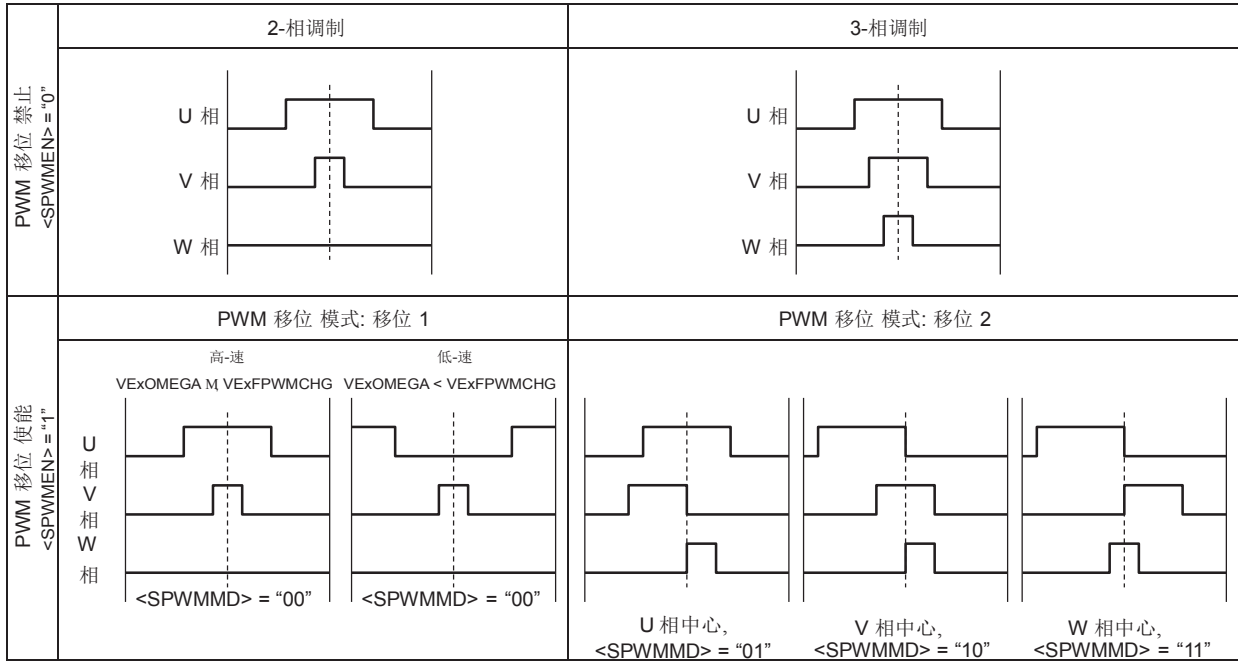
	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	SPWMMD		SADCEN	PHCVDIS	VSLIMMD		MREGDIS	CRCEN
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ICPLMD	IBPLMD	IAPLMD	-	IDMODE		SPWMEN	C2PEN
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 "0".
15-14	SPWMMD[1:0]	R/W	选择 PWM 移位模式. 00: 移位 1 01: 移位 2 (U-相 正常 PWM) 10: 移位 2 (V-相 正常 PWM) 11: 移位 2 (W-相 正常 PWM) 关于详细, 参考 "图 18-6 移位设置和 PWM 波之间关系". 注) 输出 控制 1 (任务 0)被执行时(无效), 移位 2 不可以选择.
13	SADCEN	R/W	使用两单元 ADC 禁止/使能同步电流采样控制. 0: 同步 采样禁止. 1: 同步 采样使能.
12	PHCVDIS	R/W	禁止/使能相转换. 0: 2-3 相转换使能. (3-相 AC 输出) 1: 2-3 相转换禁止. (2-相 AC 输出) 注) 不可以禁止相转换空间向量调制 (任务 8)运行(无效) 时.
11-10	VSLIMMD [1:0]	R/W	控制任务 5 的电压标量极限. 00: 标量 极限禁止. (各轴的极限使能.) 01: 限制 d-轴上电压. 10: 限制 q-轴上电压. 11: dq 比例极限
9	MREGDIS	R/W	存储 SIN/COS/SECTOR 之前的值 0: 使能 1: 禁止 此位禁止时, VExSINM = VExSIN, VExCOSM = VExCOS, 和 VExSECTORM = VExSECTOR 被设置.
8	CRCEN	R/W	禁止/使能触发器修改. 0: 禁止 1: 使能 注:触发器生成 (任务 1)被执行时此位使能; 1-电阻电流检测模式下 PWM 移位禁止时; 或移位 1 选择时.
7	ICPLMD	R/W	选择任务 10 电流方向 (Ic). 0: 电阻 模式 (Ic = VExICO - VExICADC) 1: 传感器 模式 (Ic = VExICADC - VExICO)
6	IBPLMD	R/W	选择任务 10 电流方向 (Ib). 0: 电阻 模式 (Ib = VExIBO - VExIBADC) 1: 传感器 模式 (Ib = VExIBADC - VExIBO)
5	IAPLMD	R/W	选择任务 10 电流方向 (Ia). 0: 电阻 模式 (Ia = VExIAO - VExIAADC) 1: 传感器 模式 (Ia = VExIAADC - VExIAO)

位	位符	类型	功能
4	-	R/W	写作 "0".
3-2	IDMODE [1:0]	R/W	选择电流检测 模式. 00: 3-电阻 (注 1) 01: 2-传感器 (注 2) 10: 1-电阻 (PMD TRG 递增 计数器 (注 3, 注 4)) 11: 1-电阻 (PMD TRG 递减 计数器 (注 3, 注 4)) 注 1: 3-相电流 检测被用在输入处理 2 (任务 10). 注 2: 2-相电流 检测被用在输入处理 2 (任务 10). 注 4) 输出 控制 2 (任务 9) 和输入处理 2 (任务 10)被执行时 PWM 移位 2 应该被用.
1	SPWMEN	R/W	禁止/使能 PWM 移位. 0: 禁止 1: 使能 注: 仅移位 1 模式下输出控制 1 (任务 0) 和输入处理 1 (任务 2)可用. 仅移位 2 模式下输出控制 2 (任务 9) 和输入处理 2 (任务 10)可用.
0	C2PEN	R/W	选择调制 模式. 0: 3-相 调制 1: 2-相 调制

注 3) 1-电阻模式被用时 PMDTRG 的设置.

VExFMODE <IDMODE[1:0]>	PMDxTRGCR <TRG0MD[2:0]>	PMDxTRGCR <TRG1MD[2:0]>
10	010 (递增-计数)	010 (递增-计数)
10	101 (载波 底)	010 (递增-计数)
11	001 (递减-计数)	001 (递减-计数)
11	001 (递减-计数)	101 (载波 底)



- 注 1) 仅 2-相调制移位-1 可以选择.
- 注 2) 仅 3-相调制移位-2 可以选择.
- 注 3) 移位-2 需要设置 PMD 寄存器.

图 18-6 移位设置和 PWM 波之间关系

18.3.3.3 VExTPWM (PWM 周期 率设置 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	TPWM[15:0]	R/W	设置相插值中积分的 PWM 周期率和单元(16-位 固定-点数据: 0.0 到 1.0). 0x0000 到 0xFFFF (PWM 周期[s] × Max_Hz × 2 ¹⁶) 此指 PWM 周期和最大值转速之间比例. (Max_Hz: 最大值转速) 相插值使能时此位被用来 SIN/COS 计算 (任务 6).

18.3.3.4 VExOMEGA (转速设置寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	OMEGA[15:0]	R/W	<p>设置转速 (16-位 固定-点数据: -1.0 到 1.0). 0x8000 到 0x7FFF (转速 [Hz] ÷ Max_Hz × 2¹⁵) (Max_Hz: 最大值转速)</p> <p>相插值使能时此位被用来 SIN/COS 计算 (任务 6). PWM 移位 1 选择给 1-电阻电流检测时此位被输出控制 1 (任务 0)用. 此位被电流控制 (任务 5) 和电机电压 等式中输入协调轴转换(任务 4)用.</p>

18.3.3.5 VExTHETA (电机 相 设置 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	THETA[15:0]	R/W	设置相(16-位 固定-点数据: 0.0 到 1.0). 设置值: $\text{相}[\text{deg}] \div 360 \times 2^{16}$ 相插值使能时此位被 SIN/COS 计算 (任务 6)用. 相插值使能时 SIN/COS 计算 (任务 6)被执行时此位更新.

18.3.3.6 VExCOS/VExSIN/VExCOSM/VExSINM (SIN/COS 寄存器)

VExCOS

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	COS[15:0]	R/W	设置 THETA 上 cosine 值 (16-位 固定-点数据: -1.0 到 1.0). 0x8000 到 0x7FFF SIN/COS 计算 (任务 6)被执行时此位更新. 此位被用作输出协调轴转换(任务 7).

VExSIN

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	SIN[15:0]	R/W	设置 THETA 上 sine 值 (16-位 固定-点数据: -1.0 到 1.0). 0x8000 到 0x7FFF SIN/COS 计算 (任务 6)被执行时此位更新. 此位被用作输出协调轴转换(任务 7).

VExCOSM

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	COSM[15:0]	R/W	存储之前的 VExCOS 寄存器值. 0x8000 到 0x7FFF SIN/COS 计算 (任务 6) 被执行时此位更新. 此位被用作输入协调轴转换 (任务 4).

VExSINM

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	SINM[15:0]	R/W	存储之前的 VExSIN 寄存器值. 0x8000 到 0x7FFF SIN/COS 计算 (任务 6) 被执行时此位更新. 此位被用作输入协调轴转换 (任务 4).

18.3.3.7 VExIDREF/VExIQREF (d-轴/q-轴 电流 参考 设置 寄存器)

VExIDREF

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	IDREF[15:0]	R/W	设置 d-轴电流参考值(16-位 固定-点数据: -1.0 到 1.0). 0x8000 到 0x7FFF (d-轴 电流 参考[A] + Max_I × 2 ¹⁵) Max_I: (AD 转换通过 1LSB [A]改变时相电流变化的总和) × 2 ¹¹ 此位被电流控制用 (任务 5).

VExIQREF

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	IQREF[15:0]	R/W	设置 q-轴电流参考值(16-位 固定-点数据: -1.0 到 1.0). 0x8000 到 0x7FFF (q 轴 电流 参考[A] + Max_I × 2 ¹⁵) Max_I: (AD 转换通过 1LSB [A]改变时相电流变化的总和) × 2 ¹¹ 此位被电流控制用 (任务 5).

18.3.3.8 VExVD/VExVQ (d-轴/q-轴 电压 设置 寄存器)

VExVD

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

位	位 符	类型	功能
31-0	VD[31:0]	R/W	设置 d-轴电压值(32-位 固定-点 数据: -1.0 到 1.0). 0x8000_0000 到 0x7FFF_FFFF: d-轴 电压[V] ÷ Max_V × 2 ³¹ Max_V: (AD 转换通过 1LSB [V] 改变时电源电压变化总和) × 2 ¹² 电流控制 (任务 5)被执行时此位更新. 此位被输出协调轴 变换用(任务 7). 电机电压等式中此位被输入协调轴变换用(任务 4).

VExVQ

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

位	位 符	类型	功能
31-0	VQ[31:0]	R/W	设置 q-轴电压值(32-位 固定-点 数据: -1.0 到 1.0). 0x8000_0000 到 0x7FFF_FFFF: q-轴 电压[V] ÷ Max_V × 2 ³¹ Max_V: (AD 转换通过 1LSB [V] 改变时电源电压变化总和) × 2 ¹² 电流控制 (任务 5)被执行时此位更新. 此位被输出协调轴 变换用(任务 7). 电机电压等式中此位被输入协调轴变换用(任务 4).

18.3.3.9 VExCIDKI/VExCIDKP/VExVCIQKI/VExCIQKP (PI 控制 系数 寄存器)

VExCIDKI

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

位	位符	类型	功能
31-16	-	R	读作"0".
15-0	CIDKI[15:0]	R/W	设置 d-轴 PI 控制 积分系数 (0x8000 到 0x7FFF).

VExCIDKP

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

位	位符	类型	功能
31-16	-	R	读作"0".
15-0	CIDKP[15:0]	R/W	设置 d-轴 PI 控制 比例系数 (0x8000 到 0x7FFF).

VExCIQKI

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	CIQKI[15:0]	R/W	设置 q-轴 PI 控制积分系数 (0x8000 到 0x7FFF).

VExCIQKP

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	CIQKP[15:0]	R/W	设置 q-轴 PI 控制积分系数 (0x8000 到 0x7FFF).

18.3.3.10 VExVDIH/VExVDILH/VExVQIH/VExVQILH (PI 控制 集成 组成 储存 寄存器)

VExVDIH

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

位	位 符	类型	功能
31-0	VDIH[31:0]	R/W	存储 PI 控制上 d-轴积分组成(VDI) 上 32 位.

VExVDILH

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

位	位 符	类型	功能
31-16	VDILH[15:0]	R/W	存储 PI 控制上 d-轴积分组成(VDI) 低 16 位.
15-0	-	R	读作"0".

注 1: VDI 数据为 64-位 固定-点 数据 (63 小数位从 -1.0 到 1.0).

注 2: VDI 数据由 48 位组成.

VExVQIH

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

位	位符	类型	功能
31-0	VQIH[31:0]	R/W	存储PI控制上q-轴积分组成(VDI)上32位.

VExVQILH

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

位	位符	类型	功能
31-16	VQILH[15:0]	R/W	存储PI控制上q-轴积分组成(VDI)低16位.
15-0	-	R	读作"0".

注 1: VDI 数据为 64-位 固定-点 数据 (63 小数位从 -1.0 到 1.0).

注 2: VDI 数据由 48 位组成.

18.3.3.11 VExMCTLF (异常 决定 结果 储存 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	PWMOVF	VSOVF	PIQOVF	PIDOVF
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	PLSLFM	PLSLF	-	LVTF	LAVFM	LAVF
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 "0".
15-12	-	R/W	写作"0".
11	PWMOVF	R/W	指 PWM 输出极限超出标志. 0: 所有相 PWM 输出为 PWMMIN 或更多, 且 PWMMAX 或更小. 1: 任何相 PWM 输出小于 PWMMIN, 或超过 PWMMAX. 输出控制 (任务 0 和任务 9)被执行时此位更新.
10	VSOVF	R/W	指电压标量极限超出标志. 0: 电压 标量 ≤ VSLIM 1: 电压 标量 > VSLIM 电流控制 (任务 5)被执行时此位更新.
9	PIQOVF	R/W	指 PI 控制上 q-轴 输出 极限超出 标志. 0: PI 控制上 q-轴 输出 ≤ PIOLIM 1: PI 控制上 q-轴 输出 > PIOLIM 电流控制 (任务 5) 被执行时此位更新.
8	PIDOVF	R/W	指 PI 控制上 d-轴 输出 极限超出标志. 0: PI 控制上 d-轴 输出 ≤ PIOLIM 1: PI 控制上 d-轴 输出 > PIOLIM 电流控制 (任务 5)被执行时此位更新.
7-6	-	R/W	写作 "0".
5	PLSLFM	R/W	指 <PLSLF> 寄存器之前的值. 输出控制 (任务 0 和任务 9)被执行时此位更新.
4	PLSLF	R/W	设置最小脉冲标志. 输出控制 1 (任务 0)被执行且设置 1-电阻电流检测时, 最小脉冲宽差分 < VExMINPLS<MINPLS>. 如果这样, 设置 "1". 输出控制 2 (任务 9) 被执行时, 最小开启-宽或最小关闭-宽 < VExMINPLS<MINPLS>. 如果这样, 设置 "1".
3	-	R/W	写作 "0".
2	LVTF	R/W	选择低电源电压标志. VExVDC<VDC> ≥ 1/128.如果这样, 设置 "0". VExVDC<VDC> < 1/128. 如果这样, 设置 "1". 输出相转换(任务 8 和 11)被执行时此位更新.
1	LAVFM	R/W	指<LAVF> 寄存器之前的值. 输出控制 1 (任务 0) 被执行且 1-电阻电流检测中 PWM 移位使能时此位更新.

位	位符	类型	功能
0	LAVF	R/W	选择低-速 标志. 输出控制 1 (任务 0) 被执行时, 且 1-电阻 电流 检测中 PWM 移位使能. 0: 高速 1: 低速 VExOMEGA<OMEGA> ≥ VExFPWMCHG<FPWMCHG>. 如果这样, 设置 "0". VExOMEGA<OMEGA> < VExFPWMCHG<FPWMCHG>. 如果这样, 设置 "1".

18.3.3.12 VExFPWMCHG (PWM 开关 速 设置 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	FPWMCHG[15:0]	R/W	PWM 移位使能时设置切换速度. 0x0000 到 0x7FFF (切换速度 [Hz] = Max_Hz × 2 ¹⁵) (Max_Hz: 最大值转速 [Hz]) 此位被用输出控制 1 (任务 0)被执行且 1-电阻电流检测中 PWM 移位 1 使能时.

18.3.3.13 VExMDPRD (PWM 周期 设置 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作"0".
15-0	VMDPRD[15:0]	R/W	设置 PWM 周期. 设置与 PMD 电路 PMDxMDPRD 寄存器相同值. 此位被输出控制用 (任务 0 和任务 9) 且触发器生成 (任务 1).

18.3.3.14 VExMINPLS (最小 脉宽 差分 设置 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	MINPLS[15:0]	R/W	<p>输出控制 1 (任务 0)被执行且 1-电阻电流 检测中 PWM 移位使能时设置最小脉宽差分参考值(三相 PWM (VExCMPU,VExCMPV, 和 VExCMPW) 占空比最小值).</p> <p>计算如下: 脉宽差分 [s] ÷ PWM 计数器 时钟 周期 [s]</p> <p>输出控制 2 (任务 9)被执行时设置最小脉宽参考值 (三相 PWM (VExCMPU,VExCMPV, 和 VExCMPW) 占空比最小值).</p> <p>计算如下: 脉宽 [s] ÷ PWM 计数器 时钟 周期 [s]</p>

18.3.3.15 VExSECTOR/VExSECTORM (扇形 信息 寄存器)

VExSECTOR

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

位	位 符	类型	功能
31-4	-	R	读作"0".
3-0	SECTOR[3:0]	R/W	指扇形信息. 设置值: 0x0 到 0xB 在扇形中通过 30 度用分割成 12 扇形表示输出时的转动位置. 此位更新输出相转换(任务 8 和任务 11)被执行时. 此位被输出控制 1 用 (任务 0).

VExSECTORM

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

位	位 符	类型	功能
31-4	-	R	读作"0".
3-0	SECTORM[3:0]	R/W	指之前的扇形信息. 设置值: 0x0 到 0xB 此位被用在输入处理中. 此位更新输出相转换(任务 8 和任务 11)被执行时. 此位被用在输入处理 1 中 (任务 2).

18.3.3.16 VExIAO/VExIBO/VExICO (零电流 寄存器)

VExIAO

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	IAO[15:0]	R/W	存储 a-相零电流时 AD 转换结果。(电机停止时存储 a-相电流 AD 转换结果.) 输入处理 1 中(任务 2) 零-电流检测模式选择时此位更新. AD 转换结果被捕捉时 a-相零电流时 AD 转换结果存储在<IAO[15:4]> . <IAO[3:0]> 存储 "0".

VExIBO

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	IBO[15:0]	R/W	存储 b-相零电流时 AD 转换结果.(电机停止时存储 b-相电流 AD 转换结果.) 输入处理 1 中(任务 2) 零-电流检测模式选择时此位更新. AD 转换结果被捕捉时 b-相零电流时 AD 转换结果存储在<IAO[15:4]> . <IAO[3:0]> 存储 "0".

VExICO

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	ICO[15:0]	R/W	<p>存储 c-相零电流时 AD 转换结果. (电机停止时存储 c-相电流 AD 转换结果.)</p> <p>输入处理 1 中(任务 2) 零-电流检测模式选择时此位更新.</p> <p>AD 转换结果被捕捉时 c-相零电流时 AD 转换结果储存在<IAO[15:4]> . <IAO[3:0]> 存储 "0".</p>

18.3.3.17 VExIAADC/VExIBADC/VExICADC (ADC 电流 结果 寄存器)

VExIAADC

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	IAADC[15:0]	R/W	存储 a-相电流 AD 转换结果(0x0000 到 0xFFFF). 此位更新输入处理(任务 2 和任务 10)被执行时. AD 转换结果被捕捉时 a-相 AD 转换结果储存在<IAADC[15:4]> . <IAADC[3:0]> 存储 "0".

VExIBADC

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	IBADC[15:0]	R/W	存储 b-相电流 AD 转换结果(0x0000 到 0xFFFF). 此位更新输入处理(任务 2 和任务 10)被执行时. AD 转换结果被捕捉时 b-相 AD 转换结果储存在<IAADC[15:4]> . <IAADC[3:0]> 存储 "0".

VExICADC

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	ICADC[15:0]	R/W	<p>存储 c-相电流 AD 转换结果(0x0000 到 0xFFFF).</p> <p>此位更新输入处理(任务 2 和任务 10)被执行时.</p> <p>AD 转换结果被捕捉时 c-相 AD 转换结果储存在<IAADC[15:4]> . <IAADC[3:0]> 存储 "0".</p>

18.3.3.18 VExVDC/VExVDCL (电源电压 寄存器)

VExVDCx

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	VDC[15:0]	R/W	设置电源电压 (16-位 固定-点 数据: 0 到 1.0). 设置值: 0x0000 到 0x7FFF 实际电压值可以被计算作: $VDC \text{ 值} \times \text{Max_V} \div 2^{15}$ 值 (Max_V: (AD 转换通过 1LSB [V] 改变时电源电压变化总和) $\times 2^{12}$) VExMODE<VDCSEL> 设置为 "0" 和输入处理(任务 2 和任务 10)被执行时此位更新. 此位被输出相转换用 (任务 8 和 11).

VExVDCL

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	VDCL[15:0]	R/W	设置电源电压 (16-位 固定-点 数据: 0 到 1.0). 设置值: 0x0000 到 0x7FFF 实际电压值可以被计算作: $VDC \text{ 值} \times \text{Max_V} \div 2^{15}$ 值 (Max_V: (AD 转换通过 1LSB [V] 改变时电源电压变化总和) $\times 2^{12}$) VexMODE<VDCSEL> 设置为 "1" 和输入处理(任务 2 和任务 10)被执行时此位更新.

18.3.3.19 VExID/VExIQ (d-轴/q-轴 电流 寄存器)

VExID

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

位	位符	类型	功能
31-0	ID[31:0]	R/W	<p>设置 d-轴 电流 (32-位 固定-点 数据: -1.0 到 1.0). 设置值: 0x8000_0000 到 0x7FFF_FFFF 实际电压值可以被计算作: ID 值 × Max_I ÷ 2³¹ 值 (Max_I: (AD 转换通过 1LSB [A]) × 2¹¹ 改变时的相电流变化总和)</p> <p>此位更新输入协调轴转换(任务 4)被执行时. 此位被电流控制用(任务 5).</p>

VExIQ

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

位	位符	类型	功能
31-0	IQ[31:0]	R/W	<p>设置 q-轴 电流 (32-位 固定-点 数据: -1.0 到 1.0). 设置值: 0x8000_0000 到 0x7FFF_FFFF 实际电压值可以被计算作: ID 值 × Max_I ÷ 2³¹ 值 (Max_I: (AD 转换通过 1LSB [A]) × 2¹¹ 改变时的相电流变化总和)</p> <p>此位更新输入协调轴转换(任务 4)被执行时. 此位被电流控制用(任务 5).</p>

18.3.3.20 VExTADC (ADC 转换时间 设置 寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	TADC[15:0]	R/W	设置 AD 转换时间 (参考图 18-7). 0x0000 到 0xFFFF: (AD 转换 时间 [s] ÷ PWM 计数器 时钟 周期 [s]) 执行载波峰触发器时序的向前修改, 1-电阻电流检测模式下 PWM 移位 1 输出 (移位 1 使能且低-速标志为 "1")被用时. 此位被用在任务 0.

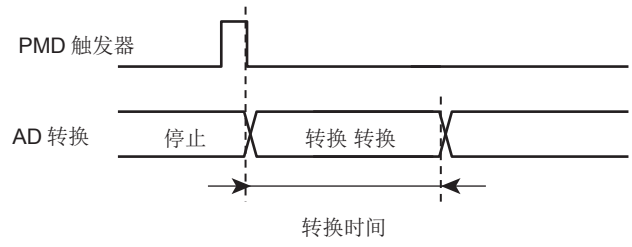


图 18-7 ADC 转换 时间

18.3.3.21 VExCMPU/ VExCMPV/ VExCMPW (PWM 占空比 寄存器)

VExCMPU

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	VCMPU[15:0]	R/W	设置 U-相 PWM 脉宽. 设置值: 0x0000 到 0xFFFF 此位更新输出控制 (任务 0 和任务 9)被执行时. 此位被触发器生成用(任务 1).

VExCMPV

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	VCMPV[15:0]	R/W	设置 V-相 PWM 脉宽. 设置值: 0x0000 到 0xFFFF 此位更新输出控制 (任务 0 和任务 9)被执行时. 此位被触发器生成用(任务 1).

VExCMPW

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	VCMPW[15:0]	R/W	设置 W-相 PWM 脉宽. 设置值: 0x0000 到 0xFFFF 此位更新输出控制 (任务 0 和任务 9) 被执行时. 此位被触发器生成 (任务 1).

18.3.3.22 VExOUTCR (6-相 输出 控制 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	WPWM
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	VPWM	UPWM	WOC		VOC		UOC	
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-9	-	R	读作 "0".
8	WPWM	R/W	选择 W-相 PWM. 0: 开启/关闭 输出 1: PWM 输出
7	VPWM	R/W	选择 V-相 PWM 0: 开启/关闭 输出 1: PWM 输出
6	UPWM	R/W	选择 U-相 PWM 0: 开启/关闭 输出 1: PWM 输出
5-4	WOC[1:0]	R/W	选择 W-相输出 控制. 00: WO 关闭. ZO 关闭. (注) 01: WO 开启. ZO 关闭. 10: WO 关闭. ZO 开启. 11: WO 开启. ZO 开启. (注) <WPWM> = 1 时 WO 和 ZO 开启.
3-2	VOC[1:0]	R/W	选择 V-相输出 控制. 00: VO 关闭. YO 关闭. (注) 01: VO 开启. YO 关闭. 10: VO 关闭. YO 开启. 11: VO 开启. YO 开启. (注) <VPWM> = 1 时 VO 和 YO 开启.
1-0	UOC[1:0]	R/W	选择 U-相输出 控制. 00: UO 关闭. XO 关闭. (注) 01: UO 开启. XO 关闭. 10: UO 关闭. XO 开启. 11: UO 开启. XO 开启. (注) <UPWM> = 1 时 UO 和 XO 开启.

注:输出控制 (任务 0 和任务 9)被执行时此设置更新.

PMD U,V 和 W-相输出控制所示 如下: (此表仅表示那些被用在 VE 的组合.)

表 18-1 <UPWM>, <UOC> PMD 设置: U-相(UO,XO) 输出 控制

设置		输出	
<UPWM>	<UOC>	UO	XO
0	00	关闭 输出	关闭 输出
1	00	PWMU 反向输出	PWMU 输出
1	11	PWMU 输出	PWMU 反向输出

表 18-2 <VPWM>,<VOC> PMD 设置: V-相(VO,YO) 输出 控制

设置		输出	
<VPWM>	<VOC>	VO	YO
0	00	关闭 输出	关闭 输出
1	00	PWMV 反向输出	PWMV 输出
1	11	PWMV 输出	PWMV 反向输出

表 18-3 <WPWM>,<WOC> PMD 设置: W-相(WO,ZO) 输出 控制

设置		输出	
<WPWM>	<WOC>	WO	ZO
0	00	关闭 输出	关闭 输出
1	00	PWMW 反向输出	PWMW 输出
1	11	PWMW 输出	PWMW 反向输出

18.3.3.23 VExTRGCRC (同步 触发器 纠正 设置 寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	TRGCRC[15:0]	R/W	修改同步触发器时序. 设置值: 修改时间 [s] ÷ PWM 计数器 时钟 周期 [s] 仅 1-电阻电流检测中 PWM 移位禁止时或移位 1 使能时此位使能. 此位被用来向后修改触发器时序. 此位被触发器生成用(任务 1).

18.3.3.24 VExTRGCMP0/VExTRGCMP1 (触发器 时序 设置 寄存器)

VExTRGCMP0

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	VTRGCMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	VTRGCMP0							
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	VTRGCMP0[15:0]	R/W	指定 PMD (PMD 设置) 触发器采样 ADC 时序同步. 0x0000: 禁止 0x0001 到 (VExMDPRD-1 的值): 触发器 时序 VExMDPRD 到 0xFFFF 的: 禁止 以下之一 PMD 触发器模式选择时此位使能: 用定时器递减计数的匹配; 用定时器递增计数的匹配; 用定时器递增和递减计数的匹配 在触发器生成中 (任务 1) 1-电阻电流检测中的 PWM 移位禁止时或移位 1 使能时更新此位.

VExTRGCMP1

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	VTRGCMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	VTRGCMP1							
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	VTRGCMP1[15:0]	R/W	<p>指定 PMD (PMD 设置) 触发器采样 ADC 时序同步.</p> <p>0x0000: 禁止</p> <p>0x0001 到 (VExMDPRD-1 的值): 触发器 时序</p> <p>VExMDPRD 到 0xFFFF 的: 禁止</p> <p>以下之一 PMD 触发器模式选择时此位使能:</p> <p>用定时器递减计数的匹配; 用定时器递增计数的匹配; 用定时器递增和递减计数的匹配</p> <p>在触发生成中 (任务 1) 1-电阻电流检测中的 PWM 移位禁止时或移位 1 使能时更新此位.</p>

18.3.3.25 VExTRGSEL (同步 触发器 选择 寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2-0	VTRGSEL[2:0]	R/W	<p>指定在 VExTRGCMP0 (PMD 设置) 中指定的时序被输出的同步触发器数.</p> <p>0 到 5: 输出触发器数</p> <p>6 到 7: 禁止</p> <p>触发器选择输出 (PMDxTRGMD<TRGOUT> = "1")被选作 PMD 触发器输出模式时此位使能.</p> <p>此位更新为触发器生成 中(任务 1)VExSECTOR ÷ 2 的值.</p>

18.3.3.26 VExEMGRS (EMG 保护 释放 寄存器)

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

位	位 符	类型	功能
31-1	-	R	读作 "0".
0	EMGRS	W	EMG 保护释放命令(PMD 设置) 0: - 1: EMG 保护释放 如果 "1"写入此位, 此位在下一个周期清除. 始终读作 "0". 恢复模式中输出控制 (任务 0 和任务 9) 被执行时此位设置为 "1".

18.3.3.27 VExPIOLIM (PI 控制 输出 限制 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	PIOLIM[15:0]	R/W	PI 控制输出极限值设置 设置值: 0x0000 到 0x7FFF 此位被电流控制用(任务 5).

18.3.3.28 VExCIDKG/VExCIQKG (PI 控制的系数 范围 设置 寄存器 d-轴/q-轴)

VExCIDKG

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7-0	CIDKG[7:0]	R/W	选择 PI 控制 d-轴系数范围. 0x00: 1/1 0x01: 1/2 ⁴ 0x02: 1/2 ⁸ 0x03: 1/2 ¹² 0x04: 1/2 ¹⁶ 0x05 到 0xFF: 保留 此位被电流控制用 (任务 5).

VExCIQKG

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7-0	CIQKG[7:0]	R/W	选择 PI 控制 q-轴系数范围 0x00: 1/1 0x01: 1/2 ⁴ 0x02: 1/2 ⁸ 0x03: 1/2 ¹² 0x04: 1/2 ¹⁶ 0x05 到 0xFF: 保留 此位被电流控制用 (任务 5).

18.3.3.29 VExVSLIM (电压 标量 限制 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	VSLIM[15:0]	R/W	设置 d-轴 电压 (VExVD) 和 q-轴 电压 (VExVQ) 电压标量极限值。 0x0000 到 0x7FFF: 电压 [V] ÷ Max_V × 2 ¹⁵ Max_V: (AD 转换通过 1LSB [V] 改变时电源电压变化总和) × 2 ¹² 0x0000 设置时标量电压极限无效。 扩展控制使能时此位被电流控制 (任务 5) 用。

18.3.3.30 VExVDQ (电压 标量 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	VDQ[15:0]	R/W	设置 d-轴 电压 (VExVD) 和 q-轴 电压 (VExVQ) 电压标量的值, 或设置方向标量被限时 d-轴 电压 (VExVD) 和 q-轴 电压 (VExVQ) 轴极限值。 0x0000 到 0x7FFF: 电压 [V] ÷ Max_V × 2 ¹⁵ Max_V: (AD 转换通过 1LSB [V] 改变时电源电压变化总和) × 2 ¹² 扩展控制使能时此位被电流控制用 (任务 5)。

18.3.3.31 VExDELTA (倾角寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	DELTA[15:0]	R/W	设置 d-q 协调上的偏向角. 范围: 0x8000 到 0x7FFF(-180 度 到 180 度) 偏向角: $[\text{deg}] \div 360 \times 2^{16}$ 协调轴转换(任务 4) 上 ATAN 计算使能或电流控制 (任务 5)中电压标量极限使能时此位更新.

18.3.3.32 VExCPHI/VExCLD/VExCLQ/VExCR/VExCPHIG/VExCLG/VExCRG (电机 恒定 寄存器)

VExCPHI

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	CPHI[15:0]	R/W	设置电机反电动势常数[V/rps] (链接磁通 [Wb/s]) 设置值: 0x0000 到 0x7FFF 反电动势常数 [V/rps] 值= Max_V × Max_Hz × 2 ¹¹ ÷ [CPHIG 设置] 此位被用在参考轴 转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时.

VExCLD

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	CLD[15:0]	R/W	<p>设置电机常数的 d-轴 电感. 设置值: 0x0000 到 0x7FFF 电感 [H] 值 $\times \text{Max}_I \div \text{Max}_V \times \text{Max}_{Hz} \times 2\pi \times 2^{11} \div [\text{CLG 设置}]$</p> <p>此位被用在参考轴 转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时.</p>

VExCLQ

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	CLQ[15:0]	R/W	<p>设置电机常数的 q-轴 电感. 设置值: 0x0000 到 0x7FFF 电感 [H] 值 $\times \text{Max}_I \div \text{Max}_V \times \text{Max}_{Hz} \times 2\pi \times 2^{11} \div [\text{CLG 设置}]$</p> <p>此位被用在参考轴 转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时</p>

VExCR

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	CR[15:0]	R/W	设置电机常数电阻值. 设置值: 0x0000 到 0x7FFF 电阻 [Ω] × Max_I ÷ Max_V × 2 ¹¹ + [CRG 设置] 值 此位被用在参考轴转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时

VExCPHIG

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7-0	CPHIG[7:0]	R/W	选择电机常数磁通范围. 0x00: 1/1 0x01: 1/2 ⁴ 0x02: 1/2 ⁸ 0x03: 1/2 ¹² 0x04: 1/2 ¹⁶ 0x05 到 0xFF: 保留 此位被用在参考轴 转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时

VExCLG

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7-0	CLG[7:0]	R/W	选择电机常数电感范围. 0x00: 1/1 0x01: 1/2 ⁴ 0x02: 1/2 ⁸ 0x03: 1/2 ¹² 0x04: 1/2 ¹⁶ 0x05 到 0xFF: 保留 此位被用在参考轴 转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时

VExCRG

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

位	位符	类型	功能
31-8	-	R	读作 "0".
7-0	CRG[7:0]	R/W	<p>选择电机常数电阻范围.</p> <p>0x00: 1/1 (CR 寄存器使用 [31:16] Q27 格式上.)</p> <p>0x01: 1/2⁴</p> <p>0x02: 1/2⁸</p> <p>0x03: 1/2¹²</p> <p>0x04: 1/2¹⁶</p> <p>0x05 到 0xFF: 保留</p> <p>此位被用在参考轴 转换(任务 4) 诱导电压偏向角被计算时或非-干扰在电流控制 (任务 5)中使能时</p>

18.3.3.33 VExVDE/VExVQE (非-干扰 控制 电压 寄存器 d-轴/q-轴)

VExVDE

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	VDE[15:0]	R/W	指非-干扰 d-轴计算值(16-位 固定-点 数据 从 -1.0 到 1.0). 0x8000 到 0x7FFF: 电压 [V] \div Max_V \times 2 ¹⁵ (Max_V: (AD 转换通过 1LSB [V]) \times 2 ¹² 改变时电压变化总和) 电流控制中非-干扰控制使能(任务 5) 时此位更新.

VExVQE

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	VQE[15:0]	R/W	指非-干扰 q-轴计算值(16-位 固定-点 数据 从 -1.0 到 1.0). 0x8000 到 0x7FFF: 电压 [V] \div Max_V \times 2 ¹⁵ (Max_V: (AD 转换通过 1LSB [V]) \times 2 ¹² 改变时电压变化总和) 电流控制中非-干扰控制使能(任务 5) 时此位更新.

18.3.3.34 VExDTC (死区 时间 补偿 寄存器)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	DTC	R/W	设置死区时间控制上补偿总和. 0x0000 到 0xFFFF: 死区 时间 [s] ÷ PWM 周期 [s] × VExMDPRD 值 输出控制中死区 时间补偿使能 (任务 0 和任务 9) 时此位被用.

18.3.3.35 VExHYS (电流识别迟滞 宽)

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	HYS[15:0]	R/W	电流极性被决定时(16-位 固定-点 数据 从 -1.0 到 1.0) 设置电流滞后宽. 0x8000 到 0x7FFF: 滞后 宽 [A] ÷ Max_I × 2 ¹⁵ (Max_I: (AD 转换通过 1LSB [A]) × 2 ¹¹ 改变时相电流变化总和) 电流极性决定使能时此位被用在输入处理(任务 2 和任务 10).

18.3.3.36 VExDTCS (死区 时间 补偿 控制/状态 寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-11	-	R/W	写作 "0".
10-8	ICSTS[1:0]	R/W	指 电流 (Ic) 极性决定的状态,或选择 CMPW 死区时间补偿控制设置. xx0: 极性 决定未定义. 死区时间补偿不执行. x01: 正极 电流被决定. 加 DTC 值死区时间补偿执行时. x11: 负极 电流被决定. 执行死区 时间补偿时减 DTC 值. 输入处理中 (任务 2 和任务 10) 电流极性决定使能时此位更新. 死区时间补偿使能时此位被用在输出 控制 (任务 0 和任务 9).
7	-	R/W	写作 "0".
6-4	IBSTS[1:0]	R/W	指 电流 (Ib) 极性决定的状态,或选择 CMPV 死区时间补偿控制设置. xx0: 极性 决定未定义. 死区时间补偿不执行. x01: 正极 电流被决定. 加 DTC 值死区时间补偿执行时. x11: 负极 电流被决定. 执行死区 时间补偿时减 DTC 值. 输入处理中 (任务 2 和任务 10) 电流极性决定使能时此位更新. 死区时间补偿使能时此位被用在输出 控制 (任务 0 和任务 9).
3	-	R/W	写作 "0".
2-0	IASTS[1:0]	R/W	指 电流 (Ia) 极性决定的状态,或选择 CMPU 死区时间补偿控制设置. xx0: 极性 决定未定义. 死区时间补偿不执行. x01: 正极 电流被决定. 加 DTC 值死区时间补偿执行时. x11: 负极 电流被决定. 执行死区 时间补偿时减 DTC 值. 输入处理中 (任务 2 和任务 10) 电流极性决定使能时此位更新. 死区时间补偿使能时此位被用在输出 控制 (任务 0 和任务 9).

18.3.3.37 VExPWMMAX/VExPWMMIN (PWM 输出 限制 寄存器)

VExPWMMAX

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	PWMMAX[15:0]	R/W	设置 PWM 输出上限值. 设置值: 0x0000 到 0xFFFF 此位被用在输出控制 (任务 0 和任务 9).

VExPWMMIN

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

位	位 符	类型	功能
31-16	-	R	读作 "0".
15-0	PWMMIN[15:0]	R/W	设置 PWM 输出下限值. 设置值: 0x0000 到 0xFFFF 此位被用在输出 控制 (任务 0 和任务 9).

18.3.3.38 VExTHTCLP (裁剪相值设置寄存器)

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

位	位符	类型	功能
31-16	-	R	读作 "0".
15-0	THTCLP[15:0]	RW	设置 THETA 角裁剪相插值执行时(16-位 固定-点 数据从 0.0 到 1.0). 设置值: 相 [deg] ÷ 360 × 2 ¹⁶ 相插值使能时此位被用在 SIN/COS 计算 (任务 5).

18.4 工作描述

18.4.1 调度程序 管理

图 18-8 表示电机控制流程图. 向量引擎变为工作状态根据调度程序设置 (VExACTSCH) 和模式设置 (VexMODE).

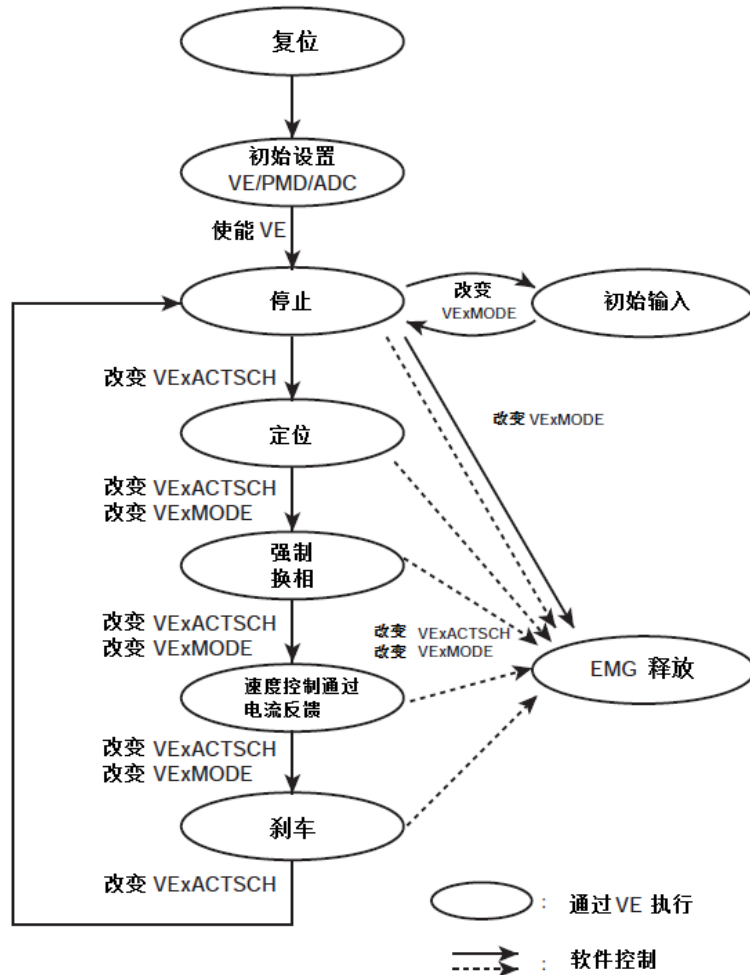


图 18-8 电机控制工作状态 流程图例

复位	: 复位 MCU.
初始 设置	: 通过用户-创造的程序指定初始设置.
停止	: 停止电机.
初始 输入	: 电机停止时采样和存储零-电流数据 .
定位	: 控制电机初始位置.
强制 换相	: 转动电机. 电机按指定的速度转动指定的期间没有电流反馈控制.
通过电流反馈速度 控制	: 控制电机转动通过电流反馈.
刹车	: 声明控制
EMG 恢复	: 释放 EMG 保护状态.

18.4.1.1 调度程序 控制

工作调度程序通过 VExACTSCH 寄存器选择.

调度程序由输出调度程序负责输出处理任务和输入调度程序负责输入处理任务. 表 18-4 表示调度程序和工作任务之间关系.

任务工作作用专用寄存器根据电机控制方法指定.

表 18-4 各调度程序中执行任务

调度程序 选择 VExACTSCH <VACT[3:0]>	输出 调度程序 执行 任务								输入 调度程序 执行 任务				仅由独立 执行提供	
	电流 控制	SIN/ COS	输出 协调 轴 转换	输出 相 转换 1	输出 相 转换 2	输出 控制 1	输出 控制 2	触发 器生成	输入 处理 1	输入 处理 2	输入 相 转换	输入 协调 轴 转换	ATAN2	平方根
	任务 5	任务 6	任务 7	任务 8	任务 11	任务 0	任务 9	任务 1	任务 2	任务 10	任务 3	任务 4	任务 12	任务 13
0:独立任务执行	(注 1)	(注 1)	(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: 调度程序 1	0	0	0	0	-	0	-	0	0	-	0	0	-	-
2: 调度程序 2	0	0	0	0	-	-	0	0	-	0	0	0	-	-
3: 调度程序 3	0	0	0	-	0	-	0	0	-	0	0	0	-	-
4: 调度程序 4	-	0	0	0	-	0	-	0	0	-	0	0	-	-
5: 调度程序 5	-	0	0	0	-	-	0	0	-	0	0	0	-	-
6: 调度程序 6	-	0	0	-	0	-	0	0	-	0	0	0	-	-
7: 调度程序 7	-	0	0	-	0	0	-	0	0	-	0	0	-	-
8: 调度程序 8	0	0	0	-	0	0	-	0	0	-	0	0	-	-
9: 调度程序 9	-	-	-	-	-	0	-	0	0	-	-	-	-	-
10: 调度程序 10	0	0	0	0	-	0	-	-	-	-	-	-	-	-
11: 调度程序 11	0	0	0	0	-	-	0	-	-	-	-	-	-	-
12: 调度程序 12	0	0	0	-	0	-	0	-	-	-	-	-	-	-
13: 调度程序 13	0	0	0	-	0	0	-	-	-	-	-	-	-	-
14: 调度程序 14	-	-	-	-	-	-	-	0	0	-	0	0	-	-
15: 调度程序 15	-	-	-	-	-	-	-	0	-	0	0	0	-	-

注 1: 仅执行用 VExTASKAPP 指定的任务.

注 2: 0: 执行 任务, -: 非 执行 任务

表 18-5 典型 workflow 设置例

设置	调度程序 设置 VExACTSCH	任务 选择 VExTASKAPP	相插值 使能 VExMODE	输出 控制 工作 VExMODE	零-电流 检测 VExMODE
电机 控制 流	<VACT[3:0]>	<VTASK[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 换相	1	5	1	01	0
电流 反馈 速度 控制	1	5	1	01	0
刹车	4	6	0	01	0
EMG 恢复	9	0	x	11	0
短路 电路 刹车	4	6	x	10	0

输出调度程序用命令 (VExCPURUNTRG) 开始工作. 所有输出-相关任务完成时, 向量引擎进入待机状态且等待开始触发器 (VExTRGMODE 设置).

输入调度程序通过开始触发器开始工作. 所有输入-相关任务完成时, 向量引擎生成中断给 CPU 且进入停止状态. 然而, 如果调度程序 (VExREPTIME)的重复数设置为"2" 或更多, 输出调度程序继续执行. 中断不发生直到调度程序已经被执行指定的次数.

注:调度程序 10 到调度程序 15 中重复设置不可用. (调度程序结束一次即使条件 VExREPTIME \geq 2.)

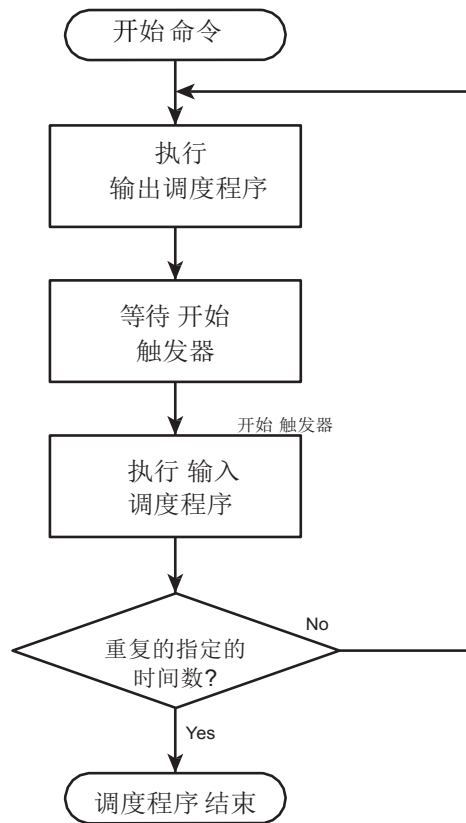


图 18-9 工作调度程序流程图

18.4.1.2 开始控制

调度程序开始前, 设置向量引擎使能 (<VEEN> ="1"), 然后设置工作调度程序选择寄存器 (VExACTSCH), 任务指定寄存器 (VExTASKAPP) 和工作调度程序重复指定寄存器 (VExREPTIME). 然后, 调度程序可以如下被执行:

向量引擎调度程序由输出调度程序和输入调度程序组成. 典型上, 输出调度程序完成后向量引擎进入等待状态. 此时, 如果开始触发器发生, 输入调度程序被执行.

输出调度程序和输入调度程序开始在如下不同条件:

- 输出调度程序的开始条件
 1. 通过命令开始. 任务 (VExTASKAPP) 被 VExCPURUNTRG 指定.
 2. 重复设置. 输入调度程序已经被执行数次后 ($VExREPTIME \geq 2$) 开始.
- 输入调度程序的开始条件
 1. 开始从通过触发器 (被 VExTRGMODE 指定) 的等待状态开始的输入处理任务. 此时, 输出调度程序完成后向量引擎处在等待状态.
 2. 通过命令开始. 任务 (VExTASKAPP) 被 VExCPURUNTRG 指定.

表 18-6 调度程序-相关任务

寄存器	功能	
VExACTSCH	工作 调度程序选择	0x0: 仅执行被 VExTASKAPP 指定的任务 0x1: 调度程序 1 被执行. 0x2: 调度程序 2 被执行. 0x3: 调度程序 3 被执行. 0x4: 调度程序 4 被执行. 0x5: 调度程序 5 被执行. 0x6: 调度程序 6 被执行. 0x7: 调度程序 7 被执行. 0x8: 调度程序 8 被执行. 0x9: 调度程序 9 被执行. 0xA: 调度程序 10 被执行. 0xB: 调度程序 11 被执行. 0xC: 调度程序 12 被执行. 0xD: 调度程序 13 被执行. 0xE: 调度程序 14 被执行. 0xF: 调度程序 15 被执行.
VExTASKAPP	开始任务设计	指定可以在选择工作调度程序中被执行的任务数.
VExREPTIME	调度程序的重复数	从 1 到 15 设置数. 注: 指定执行一次时, 设置 "1". 如果设置 "0", 调度程序不执行.
VExTRGMD	开始触发器模式选择	选择输入调度程序触发器. 设置 ADC 单元 A 中断或 ADC 单元 B 中断.

18.4.1.3 中断 控制

向量引擎有发生在调度程序完成的向量引擎中断 (INTVCNx), 和发生在指定的任务完成的任务完成中断 (INTVCTx).

- 向量 引擎 中断
 1. 用工作调度程序选择寄存器 (VExACTSCH) 指定调度程序.
设置命令(VExCPURUNTRG = "1").
 2. 调度程序已经被执行指定次(VExREPTIME) 时, INTVCNx 中断发生.
 3. 如果错误检测中断控制使能 (VExERRINTEN<VERREN> = "1"),输出调度程序被执行时 PMD 电路的 PWM 中断发生. 然后 INTVCNx 中断发生且错误标志 (VExERRDET<VERRD>)设置为 "1".
- 任务 完成 中断
 1. 指定由任务完成中断 (VExTASKAPP<VITASK>)触发的任务.
设置任务完成中断控制来使能 (VExERRINTEN<INTTEN> = "1").
 2. 通过命令(VExCPURUNTRG = "1")开始调度程序. 被 <VITASK>指定的任务完成时设置 INTVCTx 中断发生.

18.4.2 任务描述

此子章节描述调度程序中各任务工作概要。

表 18-7 表示被用来指定执行任务和开始任务的任務数。

表 18-7 任务表

任务		任务 功能	任务 数
输出 调度程序	电流 控制	d-轴/q-轴 PI 控制 (PI 控制 输出 极限 使能) d-轴 和 q-轴非-干扰 控制, 电压标量极限	5
	SIN/COS 计算	Sine/cosine 计算 相插值 (带 裁剪功能)	6
	输出 协调轴转换	反转 Park 转换	7
	输出 相转换 1	从 2-相到 3-相转换[SVM]	8
	输出 相转换 2	从 2-相到 3-相转换[反转 Clarke 转换]	11
	输出 控制 1	转换数据到 PMD 设置格式. 切换 PWM 移位 1. 限制 PWM 输出. 设置死区时间补偿控制.	0
	输出 控制 2	转换数据到 PMD 设置格式. 切换 PWM 移位 2. 限制 PWM 输出. 设置死区时间补偿控制.	9
	触发器 生成	生成同步触发器时序.	1
输入 调度程序	输入 处理 1	对应传感器, 3-电阻和 1-电阻电流检测.1- 电阻对应 PWM 移位禁止 或 PWM 移位 1 时. 捕捉 AD 转换结果和转换他们到固定- 点 格式. 指定滞后宽决定电流极性.	2
	输入 处理 2	对应传感器, 3-电阻和 1-电阻电流检测.1- 电阻对应 PWM 移位 2 时. 捕捉 AD 转换结果和转换他们到 固定- 点 格式. 指定滞后宽决定电流极性.	10
	输入 相转换	从 3-相到 2-相转换.	3
	输入 协调轴转换	执行 Park 转换 计算 d-q 协调上电流向量或诱导电压向量的偏向角.	4
ATAN2 计算		计算反正切.	12
SQRT 计算		计算平方根.	13

18.4.2.1 电流控制 (任务 5)

电流控制任务由 d-轴电流 PI 控制单元和 q-轴电流 PI 控制单元组成, 且分别计算 d-轴和 q-轴电压. 扩展控制使能非-干扰控制和一起控制 p-轴 和 q-轴的电压标量极限.

1. d-轴 电流 PI 控制

<等式>

[PI 控制]

$CIDKP = VExCIDKP \times [VExCIDKG \text{ 设置}]$: 比例系数
 $CIDKI = VExCIDKI \times [VExCIDKG \text{ 设置}]$: 积分系数
 $\Delta ID = VExIDREF - VExID$: 计算电流参考值和电流反馈之间.
 $VDI0 = CIDKI \times \Delta ID + VExVDI$: 计算积分组成.
 $VD0 = CIDKP \times \Delta ID + VExVDI0$: 通过加比例组成计算电压.

[PI 控制 输出 极限]

if ($VD0 > VExPIOLIM$) : 上-限 值
 $VExVD = VExPIOLIM$
 $VExMCTLF<PIDOVF> = 1$
 else if ($VD0 < -VExPIOLIM$) : 下-限 值
 $VExVD = -VExPIOLIM$
 $VExMCTLF<PIDOVF> = 1$
 else $VExVD = VD0$

[抗-饱和 (AWU)]

$\Delta VD = VExVD - VD0$: 计算 d-轴电压和极限之间差分.
 $VExVDI = VDI0 + \Delta VD \times [VExMODE<AWUMD> \text{ 设置}]$: 反映以上差分到比例组成.

	寄存器名	功能	
输入	VExID	d-轴 电流	32-位 固定-点 数据 (31 小数位)
	VExIDREF	d-轴 电流 参考值	16-位 固定-点 数据(15 小数位)
	VExCIDKP	比例系数	16-位 数据
	VExCIDKI	积分系数	16-位 数据
	VExCIDKG	d-轴 PI 控制系数范围 设置	000: 1/1, 001: 1/2 ⁴ , 010: 1/2 ⁸ , 011: 1/2 ¹² 100: 1/2 ¹⁶ , 101 到 111: 保留
	VExPIOLIM	PI 控制输出 极限	16-位 固定-点 数据 (15 小数位) 有效范围: 0x0 到 0x7FFF 注: VExPIOLIM = 0 时输出极限禁止.
	VExMODE[9:8]	输出受限时抗-饱和 比例设置.	<AWUMD> 00: 禁止, 01: 1/4, 10: 1/2, 11: 1
输出	VExVD	d-轴 电压	32-位 固定-点 数据 (31 小数位)
	VExMCTLF[8]	d-轴 输出 极限状态	<PIDOVF> 0: 无极限, 1: 受限
内部	VExVDI	d-轴 电压积分组成储存.	64-位 固定-点 数据 (63 小数位)

注: VExVDI 由 64 位组成. 上方被用在 VExVDIH 寄存器和下方被用在 VExVDILH 寄存器.

2. q-轴 电流 PI 控制

<等式>

[PI 控制]

CIQKP = VExCIQKP × [VExCIQKG 设置] : 比例系数
 CIQKI = VExCIQKI × [VExCIQKG 设置] : 积分系数
 $\Delta IQ = VExIQREF - VExIQ$: 计算电流参考值和电流反馈之间。
 $VQI0 = CIQKI \times \Delta IQ + VExVQI$: 积分组成计算
 $VQ0 = CIQKP \times \Delta IQ + VQI0$: 通过加比例组成计算电压。

[PI 控制 输出 极限]

if (VQ0 > VExPIOLIM) : 确认 上-限.

VExVQ = VExPIOLIM
 VExMCTLF<PIQOVF> = 1

else if (VQ0 < -VExPIOLIM) : 确认 下-限.

VExVQ = -VExPIOLIM
 VExMCTLF<PIQOVF> = 1

else VExVQ = VQ0

[抗-饱和 (AWU)]

$\Delta VQ = VExVQ - VQ0$: 计算 q-轴电压和极限之间差分.

$VExVQI = VQI0 + \Delta VQ \times [VExMODE<AWUMD> 设置]$: 反映以上差分到比例组成.

	寄存器名	功能	
输入	VExIQ	q-轴 电流	32-位 固定-点 数据 (31 小数 位)
	VExIQREF	q-轴 电流 参考值	16-位 固定-点 数据 (15 小数 位)
	VExCIQKP	比例系数	16-位 数据
	VExCIQKI	积分系数	16-位 数据
	VExCIQKG	q-轴 PI 控制系数范围 设置	000: 1/1, 001: 1/2 ⁴ , 010: 1/2 ⁸ , 011: 1/2 ¹² 100: 1/2 ¹⁶ , 101 到 111: 保留
	VExPIOLIM	PI 控制输出极限 值	16-位 固定-点 数据 (15 小数 位) 有效范围: 0x0 到 0x7FFF 注: VExPIOLIM = 0 时输出极限禁止.
	VExMODE [9:8]	输出受限时抗-饱和和比例设置.	<AWUMD> 00: 禁止, 01: 1/4, 10: 1/2, 11: 1
输出	VExVQ	q-轴 电压	32-位 固定-点 数据 (31 小数 位)
	VExMCTLF[9]	q-轴 输出极限 状态	<PIQOVF> 0: 无极限, 1: 受限
内部	VExVQI	q-轴 电压积分组成储存.	64-位 固定-点 数据 (63 小数 位)

注: VExVQI 由 64 位组成. 上方被用在 VExVQIH 寄存器和下方被用在 VExVQILH 寄存器.

3. 非-干扰 控制

PI 控制结果使用基于电机电压等式 d-轴和 q-轴干扰结果修改。
<等式>

if (VExMODE [10] = 1) : 扩展控制使能.
 LD = VExCLD × [VExCLG 设置] : d-轴 电感
 LQ = VExCLQ × [VExCLG 设置] : q-轴 电感
 PHI = VExCPHI × [VExCPHIG 设置] : 链接磁通
 VExVDE = -VExOMEGA × VExIQ × LQ : 计算 d-轴干扰.
 VExVQE = VExOMEGA × VExID × LD + VExOMEGA × PHI : 计算 q-轴干扰.
 if (VExMODE [11] = 1) : 非-干扰 控制使能.
 VExVD = VExVD + VExVDE
 VExVQ = VExVQ + VExVQE

	寄存器名	功能	
输入	VExVD	d-轴 电压	32-位 固定-点 数据 (31 小数位)
	VExVQ	q-轴 电压	
	VExID	d-轴 电流	
	VExIQ	q-轴 电流	
	VExCLD	d-轴 电感	16-位 固定-点 数据 (11 小数位)
	VExCLQ	q-轴 电感	
	VExCPHI	链接磁通	
	VExCLG	电感范围 设置	000: 1/1, 001: 1/2 ⁴ , 010: 1/2 ⁸ , 011: 1/2 ¹²
	VExCPHIG	链接磁通范围 设置	100: 1/2 ¹⁶ , 101 到 111: 保留
	VExOMEGA	转速	16-位 被固定到-点 数据 (15 小数位)
	VExMODE [11]	非-干扰控制 使能	<NICEN> 0: 非-干扰 控制禁止. 1: 非-干扰 控制使能.
	VExMODE [10]	扩展 控制 使能	<T5ECEN> 0: 扩展 控制禁止. (非-干扰控制禁止.) 1: 扩展 控制使能.
输出	VExVDE	d-轴非-干扰修改电压	16-位 固定-点 数据 (15 小数位)
	VExVQE	q-轴非-干扰修改电压	
	VExVD	d-轴 电压	32-位 固定-点 数据 (31 小数位)
	VExVQ	q-轴 电压	

4. 电压 标量 极限

电压标量极限控制 d-轴电压和 q-轴电压, 为了把由 d-轴 电压和 q- 轴电压组成的组成值 ($\sqrt{VD^2 + VQ^2}$ 平方根)降低在极限值以下.

<等式>

```

if (VExMODE [10] = 1 )                                     : 扩展控制使能.
  [VDQ 计算]
  if (VExVD2 + VExVQ2 > VExVSLIM2 )                   : 确认超出.
    if (VExFMODE [11:10] = 00)   VExVDQ = SQRT(VD2 + VQ2 )   : 电压标量极限禁止.
    else if (VExFMODE [11:10] =01) VExVDQ = SQRT(VExVSLIM2 - VQ2) : d-轴方向上标量极限
    else if (VExFMODE [11:10] =10) VExVDQ = SQRT(VExVSLIM2 - VD2) : q-轴方向上标量极限
    else if (VExFMODE [11:10] =11) VExVDQ = SQRT(VD2 + VQ2 )   : dq 比例标量极限

    注: SQRT 为平方根计算.

  [偏向角计算]
  X = | VExVQ |
  Y = | VExVD |
  VExDELTA = ATAN2(X, Y)
    注: ATAN2 为反正切计算.

  [各轴极限计算]
  if (VExFMODE [11:10] = 00)                               : 电压标量禁止.
    VDLIM = VExVSLIM
    VQLIM = VExVSLIM
  else if (VExFMODE [11:10] = 01)                           : d-轴 方向上标量极限
    VDLIM = VExVDQ
    VQLIM = VExVSLIM
  else if (VExFMODE [11:10] = 10)                           : q-轴 方向上标量极限
    VDLIM = VExVSLIM
    VQLIM = VExVDQ
  else if (VExFMODE [11:10] = 11)                           : dq 比例标量极限
    VDLIM = VExVSLIM × SIN(VExDELTA)
    VQLIM = VExVSLIM × COS(VExDELTA)

  [极限处理]
  if (VExVD > VDLIM)                                       VExVD = VDLIM           : d-轴上-限处理
                                                            VExMCTLF[10] = 1       : d-轴下-限
  else if (VExVD < -VDLIM)                                 VExVD = -VDLIM          : q-轴上-限
                                                            VExMCTLF[10] = 1
  if (VExVQ > VQLIM)                                       VExVQ = VQLIM           :
                                                            VExMCTLF[10] = 1

```

else if (VExVQ < -VQLIM)

VExVQ = -VQLIM

: q-轴下-限处理

VExMCTLF[10] = 1

	寄存器名	功能	
输入	VExVD	d-轴 电压	32-位 固定-点 数据 (31 小数 位)
	VExVQ	q-轴 电压	
	VExVSLIM	电压标量极限值	16-位 固定-点 数据 (15 小数 位) 0x0 到 0x7FFF 注: VExVSLIM = 0x0 时极限禁止.
	VExMODE [10]	扩展控制使能	<T5ECEN> 0: 扩展 控制禁止. (标量极限禁止.) 1: 扩展 控制使能.
	VExFMODE [11:10]	极限模式设置	<VSLIMMD> 00: 标量 极限禁止. (各轴极限使能.) 01: 标量 极限使能. d-轴 方向极限. 10: 标量 极限使能. q-轴 方向极限. 11: 标量 极限使能. dq 比例极限.
输出	VExVDQ	电压 标量值或轴方向极限值.	16-位 固定-点 数据 (15 小数 位)
	VExDELTA	偏向 角	16-位 数据 0x0000 到 0x4000 (0 到 90°)
	VExVD	d-轴 电压	32-位 固定-点 数据 (31 小数 位)
	VExVQ	q-轴 电压	
	VExMCTLF[10]	指电压标量极限超出标志.	<VSOVF> 0:非 超出 1:超出

18.4.2.2 SIN/COS 计算 (任务 6)

SIN/COS 计算任务执行相插值 计算和 SIN/COS 计算.

相插值计算转速通过 PWM 期间积分.仅相插值使能时 (VExMODE<PVIEN> = "1")被执行.

1. 相插值

<等式>

```

THETA0 = VExOMEGA × VExTPWM + VExTHETA           :计算相插值的值.
THETA0 = THETA0 & 0x0000FFFF
if (VExMODE [7] = 1)                               : 使能裁剪
    if ( VExOMEGA ≥ 0)                              : 正 转.
        if ( VExTHETA ≤ VExTHTCLP ≤ THETA0 )  THETA0 = THTCLP
        else if ( TA0 ≤ VExTHETA ≤ VExTHTCLP ) THETA0 = THTCLP
        else if ( VExTHTCLP ≤ THETA0 ≤ VExTHETA ) THETA0 = THTCLP
    else if ( VExOMEGA < 0)                         : 反 转.
        if (THETA0 ≤ VExTHTCLP ≤ VExTHETA )  THETA0 = THTCLP
        else if ( VExTHTCLP ≤ VExTA ≤ TA0 ) THETA0 = THTCLP
        else if ( VExTHETA ≤ THETA0 ≤ VExTHTCLP ) THETA0 = THTCLP
if ( VExMODE[0] = 1 ) VExTHETA = THETA0           : 相插值使能时更新 THETA 值.
    
```

	寄存器 名	功能	
输入	VExTHETA	相 θ	16-位 固定-点 数据 (0.0 到 1.0, 16 小数 位)
	VExOMEGA	转速	16-位 固定-点 数据 (-1.0 到 1.0, 15 小数 位)
	VExTPWM	PWM 周期率	16-位 数据
	VExTHTCLP	裁剪值	16-位 固定-点 数据 (0.0 到 1.0, 16 小数 位)
	VExMODE [0]	相插值 使能	<PVIEN> 0: 相插值禁止. 1: 相插值使能.
	VExMODE [7]	相裁剪控制	<CLPEN> 0: 裁剪禁止. 1: 裁剪使能.
输出	VExTHETA	相 θ	16-位 固定-点 数据 (0.0 到 1.0, 16 小数 位)

2. SIN/COS 计算

<等式>

VExSINM = VExSIN : 存储之前的值 (输入处理)

VExCOSM = VExCOS

VExSIN = SIN (VExTHETA)

: 计算 SIN 和 COS 值.

VExCOS = SIN (VExTHETA + 1/4)

if (VExFMODE [9] = 1)

: 确认之前的值不维持.

VExSINM = VExSIN

VExCOSM = VExCOS

注: SIN: Sine 计算

	寄存器名	功能	
输入	VExTHETA	相 θ	16-位 固定-点 数据 (0.0 到 1.0, 16 小数位)
	VExFMODE [9]	选择储存 SIN 和 COS 之前的值	<MREGDIS> 0: 之前的值维持. 1: 之前的值不维持.
输出	VExSIN	θ 的 sine 值	16-位 固定-点 数据 (-1.0 到 1.0, 15 小数位)
	VExCOS	θ 的 cosine 值	
	VExSINM	之前的 sine 值	
	VExCOSM	之前的 cosine 值	

18.4.2.3 输出 电压 转换 (协调轴 转换/相转换)

用两步执行输出电压转换: 协调轴转换和相转换. 有两类型相转换: 空间向量转换和反转 Clarke 转换.

1. 输出协调轴转换(任务 7)

输出协调轴任务中, α -轴和 β -轴电压基于 d-轴电压, q-轴电压, $\sin\theta$ 和 $\cos\theta$ 被计算.

<等式>

$$V_{\text{ExTMPREG3}} = V_{\text{ExCOS}} \times V_{\text{ExVD}} - V_{\text{ExSIN}} \times V_{\text{ExVQ}} \quad : \text{计算 } V_{\alpha}$$

$$V_{\text{ExTMPREG4}} = V_{\text{ExSIN}} \times V_{\text{ExVD}} + V_{\text{ExCOS}} \times V_{\text{ExVQ}} \quad : \text{计算 } V_{\beta}$$

	寄存器名	功能	
输入	VExVD	d-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExVQ	q-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExSIN	THETA Sine 值	16-位 固定-点 数据 (-1.0 到 1.0, 15 小数 位)
	VExCOS	THETA Cosine 值	16-位 固定-点 数据 (-1.0 到 1.0, 15 小数 位)
输出	VExTMPREG3	α -轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG4	β -轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)

2. 输出相转换 1 (空间向量转换) (任务 8)

输出相转换 1 决定使用 α -轴 电压 和 β -轴 电压的扇形. 此任务计算各扇形中基于空间向量转换的 a-相电压, b-相电压, 和 c-相电压占空比. 此任务可以选择 2-相或 3-相调制为调制类型.

a. 扇形决定

<等式>

```

VExSECTORM = VExSECTOR           : 存储之前的扇形.
Vα = VExTMPREG3
Vβ = VExTMPREG4
if ( Vα ≥ 0 & Vβ ≥ 0 )
    if ( |Vα| ≥ |Vβ| + sqrt(3) )
        if ( |Vα| + sqrt(3) ≥ |Vβ| )      VExSECTOR = 0
        else                               VExSECTOR = 1
    else                                   VExSECTOR = 2
else if ( Vα < 0 & Vβ ≥ 0 )
    if ( |Vα| < |Vβ| + sqrt(3) )          VExSECTOR = 3
    if ( |Vα| + sqrt(3) < |Vβ| )          VExSECTOR = 4
    else                                   VExSECTOR = 5
else if ( Vα < 0 & Vβ < 0 )
    if ( |Vα| ≥ |Vβ| + sqrt(3) )
        if ( |Vα| + sqrt(3) ≥ |Vβ| )      VExSECTOR = 6
        else                               VExSECTOR = 7
    else                                   VExSECTOR = 8
else if ( Vα ≥ 0 & Vβ < 0 )
    if ( |Vα| < |Vβ| + sqrt(3) )          VExSECTOR = 9
    else if ( |Vα| + sqrt(3) < |Vβ| )      VExSECTOR = 10
    else                                   VExSECTOR = 11
if ( VExFMODE[9] = 1 ) VExSECTORM = VExSECTOR : 检查之前的值是否无效.

```

	寄存器名	功能	
输入	VExTMPREG3	α-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG4	β-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExFMODE [9]	选择储存之前的值.	<MREGDIS> 0: 之前的扇形值维持. 1: 之前的扇形值不维持.
输出	VExSECTOR	扇形	4-位 数据
	VExSECTORM	之前的扇形	4-位 数据

b. 空间向量 转换(此例描述仅 3-相调制被用且 <SECTOR[3:0]> = "0" 和 "1"的时候.)

<等式>

```

if (SECTOR = 0,1)
    t1 = sqrt(3) ÷ VExVDC × (sqrt(3) ÷ 2 × Va - 1 ÷ 2 × Vβ)      : 计算 t1 周期.
    t2 = sqrt(3) ÷ VExVDC × Vβ                                  : 计算 t2 周期.
    t3 = 1 - t1 - t2                                           : 计算零-向量周期.
    if ( VExFMODE[0] = 0 )                                       : 3-相调制
        DUTYA = t1 + t2 + t3 ÷ 2
        DUTYB = t2 + t3 ÷ 2
        DUTYC = t3 ÷ 2
    else                                                         : 2-相调制
        DUTYA = t1 + t2
        DUTYB = t2
        DUTYC = 0

VExTMPREG0 = DUTYA
VExTMPREG1 = DUTYB
VExTMPREG2 = DUTYC
VExTMPREG5 = t3
    
```

	寄存器名	功能	
输入	VExTMPREG3	α-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数位)
	VExTMPREG4	β-轴 电压	
	VExVDC	电源 电压	16-位 固定-点 数据 (0.0 到 1.0, 15 小数位)
	VExSECTOR	扇形	4-位 数据
	VExFMODE [0]	调制 模式	<C2PEN> 0: 3-相调制 1: 2-相调制
输出	VExTMPREG0	a-相电压 占空比	32-位 固定-点 数据 (0.0 到 1.0, 31 小数位)
	VExTMPREG1	b-相电压 占空比	
	VExTMPREG2	c-相电压 占空比	
	VExTMPREG5	零-向量 占空比	

3. 输出相转换 2 (反转 Clarke 转换) (任务 11)

输出相转换 2 决定扇形使用 α -轴和 β -轴 电压. 此任务计算基于反转 Clarke 转换的 a-相, b-相和 c-相电压占空比. 此任务仅支持 3-相调制为调制类型.

另外, 如果 VEFMODE<PIGSEL> 设置为 "1", 此任务可以计算 2-相电压占空比.

a. 扇形 决定

<等式>

```

VExSECTORM = VExSECTOR           : 存储之前的扇形.
Va = VExTMPREG3
Vβ = VExTMPREG4
if ( Va ≥ 0 & Vβ ≥ 0 )
    if ( |Va| ≥ |Vβ| ÷ sqrt(3) )
        if ( |Va| + sqrt(3) ≥ |Vβ| )     VExSECTOR = 0
        else                             VExSECTOR = 1
    else                                 VExSECTOR = 2
else if ( Va < 0 & Vβ ≥ 0 )
    if ( |Va| < |Vβ| ÷ sqrt(3) )         VExSECTOR = 3
    if ( |Va| ÷ sqrt(3) < |Vβ| )         VExSECTOR = 4
    else                                 VExSECTOR = 5
else if ( Va < 0 & Vβ < 0 )
    if ( |Va| ≥ |Vβ| ÷ sqrt(3) )
        if ( |Va| + sqrt(3) ≥ |Vβ| )     VExSECTOR = 6
        else                             VExSECTOR = 7
    else                                 VExSECTOR = 8
else if ( Va ≥ 0 & Vβ < 0 )
    if ( |Va| < |Vβ| ÷ sqrt(3) )         VExSECTOR = 9
    else if ( |Va| + sqrt(3) < |Vβ| )     VExSECTOR = 10
    else                                 VExSECTOR = 11
if ( VExFMODE [9] = 1 ) VExSECTORM = VExSECTOR   : 确认之前的值禁止.

```

	寄存器名	功能	
输入	VExTMPREG3	α -轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数位)
	VExTMPREG4	β -轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数位)
	VExFMODE [9]	选择储存之前的扇形值	<MREGDIS> 0: 之前的值维持. 1: 之前的值不维持.
输出	VExSECTOR	扇形	4-位 数据
	VExSECTORM	之前的扇形	4-位 数据

b. 反转 Clarke 转换

<等式>

```

if ( VExFMODE [12] = 0 )                                     : 3-相转换使能.
    VExTMPREG0 = 1 ÷ VExVDC × Vα + 1/2                       : Va 占空
    比
    VExTMPREG1 = 1 ÷ VExVDC × ( -1 ÷ 2 × Vα + sqrt(3) ÷ 2 × Vβ ) + 1/2 : Vb 占空
    比 VExTMPREG2 = 1 ÷ VExVDC × ( -1 ÷ 2 × Vα - sqrt(3) ÷ 2 × Vβ ) + 1/2 : Vc 占
    空比
else                                                         : 相转换禁止.
    VExTMPREG0 = 1 ÷ VExVDC × Vα + 1/2                       : Va 占空
    比
    VExTMPREG1 = 1 ÷ VExVDC × Vβ + 1/2                       : Vb 占空
    比
    
```

	寄存器名	功能	
输入	VExTMPREG3	α-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数位)
	VExTMPREG4	β-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数位)
	VExVDC	电源 电压	16-位 固定-点 数据 (0.0 到 1.0, 15 小数位)
	VExFMODE [12]	相转换 设置	<PHCVDIS> 0: 相转换使能. 1: 相转换禁止.
输出	VExTMPREG0	a-相电压 占空比	32-位 固定到-点 数据 (0..0 到 1.0, 31 小数位)
	VExTMPREG1	b-相电压 占空比	32-位 固定到-点 数据 (0..0 到 1.0, 31 小数位)
	VExTMPREG2	c-相电压 占空比	32-位 固定到-点 数据 (0..0 到 1.0, 31 小数位)

18.4.2.4 输出控制

输出控制单元转换 3-相电压占空比到 PMD 设置格式. 转换结果设置到 VExCMPU, VExCMPV, 和 VExCMPW. 根据输出控制设置, 应该设置 VExOUTCR. 死区时间补偿控制和 PWM 输出极限也可以被执行.

有两类型输出控制任务: 输出控制 1 和输出控制 2. 各任务支持不同 PWM 输出.

1. 输出控制 1 (任务 0)

输出控制 1 任务支持正常 PWM 输出和移位 1 模式下 PWM 输出. PWM 移位使能时, 如果转速 (VExOMEGA) 低于 PWM 移位切换 (VExFPWMCHG) 的参考, PWM 输出变为 PWM 移位 1.

注: PWM 移位仅可以在 1-电阻电流检测模式下选择.

- 输出转换

<等式>

```

VExMCTLF[1] = VExMCTLF[0]           : 更新之前的值标志.
VExMCTLF[0] = 0                       : 清除电流标志.
if ((FMODE [3] = 1) & (FMODE [0] = 1) & (FMODE [1] = 1)) : 1-电阻电流和 2-相调制模式下移位 1 使能.
    if (|VExOMEGA| < VExFPWMCHG) VExMCTLF[0] = 1 : 设置低-速决定的低速标志.

DUTYA = VExTMPREG0
DUTYB = VExTMPREG1
DUTYC = VExTMPREG2
if (VExMCTLF[0] = 1)                  : 在低速 PWM 移位
    if (VExSECTOR = 0,3,4,7,8,11)     : 扇形决定
        DUTYA = DUTYA + VExTMPREG5    : 零-向量 V7 转换
        DUTYB = DUTYB + VExTMPREG5
        DUTYC = DUTYC + VExTMPREG5
PWMA = VExTMPREG0 × VExMDPRD          : 转换 PMD 设置 值.
PWMB = VExTMPREG1 × VExMDPRD
PWMC = VExTMPREG2 × VExMDPRD

```

	寄存器名	功能	
输入	VExTMPREG0	a-相电压	32-位 固定-点 数据 (0.0 到 1.0, 31 小数位)
	VExTMPREG1	b-相电压	
	VExTMPREG2	c-相电压	
	VExMDPRD	PWM 周期 设置	16-位 数据 (PMD 中 PWM 周期设置值.)
	VExSECTOR	扇形	4-位 数据
	VExOMEGA	转速	16-位 固定-点 数据 (-1.0 到 1.0, 15 小数位)
	VExFPWMCHG	PWM 移位切换参考速度	16-位 固定-点 数据 (0.0 到 1.0, 15 小数位)
	VExFMODE [0]	调制模式	<C2PEN> 0: 3-相调制, 1: 2-相调制
	VExFMODE [1]	PWM 移位使能	<SPWMEN> 0: 移位禁止. 1: 移位使能.
	VExFMODE [3:2]	电流 检测 模式	<IDMODE [1:0]> 00: 3-电阻. 01: 2 传感器, 1x: 1-电阻
输出	VExMCTLF[1:0]	低-速 标志	<LAVFM>,<LAVF>

- PWM 输出极限

<等式>

```

if (VExPWMMAX = 0)          MAX = VExMDPRD
else                          MAX = VExPWMMAX
if ((PWMA > MAX) & (VExMDPRD > MAX))          :检查 U-相 PWM 上-限.
    if ((VExMODE [13] = 0) | (PWMA < VExMDPRD)) :确认输出极限 100%占空比.
        PWMA = MAX
        VExMCTLF[11] = 1
MIN = VExPWMMIN
if ((PWMA < MIN) & (MIN > 0))          :检查 U-相 PWM 下-限.
    if ((VExMODE [12] = 0) | (PWMA > 0))      :确认输出极限 0%占空比.
        PWMA = MIN
        VExMCTLF[11] = 1
(用相同方法计算其他 2-相.)
    
```

	寄存器名	功能	
输入	VExMDPRD	PWM 周期设置	16-位 数据 (PMD 中 PWM 周期设置值.)
	VExPWMMAX	设置 PWM 上-限值.	16-位 数据 (设置值 从 0 到 VExMDPRD.)
	VExPWMMIN	设置 PWM 下-限值	
	VExMODE [12]	设置 PWM 受限时 0%输出占空比 .	<PWMBLEN> 0: 禁止,1: 使能
	VExMODE [13]	设置 P WM 受限时 100%输出占空比.	<PWFLEN> 0: 禁止, 1: 使能
输出	VExMCTLF[11]	指 PWM 输出极限的超出标志.	<PWMOVF>

- 死区时间补偿

<等式>

```

if (0 < PWMA < VExMDPRD)          DT = VExDTC
else                                DT = 0
if (VExDTCS<IASTS> = 01)           : 正电流上
    if (VExMODE [14] = 1)          : 设置 PMD 死区时间修改.
        if (PWMA > (VExMDPRD - 2× DT)) PWMA = (VExMDPRD + PWMA) ÷2
    else                            PWMA = PWMA + DT
if ((VExMODE [13] = 1) & (PWMMAX < VExMDPRD)) : 确认输出极限 100% 占空比
    if (PWMA > (VExMDPRD - 1))     PWMA = VExMDPRD - 1 : 修改后输出极限.
else
    if (PWMA > VExMDPRD)          PWMA = VExMDPRD : 修改后输出极限.
else if (VExDTCS<IASTS> = 11)     : 负电流上
    if (VExMODE [14] = 1)         : 设置 PMD 死区 时间修改.
        if (PWMA < (2 × DT))      PWMA = PWMA ÷2
    else                            PWMA = PWMA - DT
if ((VExMODE [12] = 1) & (PWMMIN > 0)) : 确认输出极限 0% 占空比.
    if (PWMA < 1)                 PWMA = 1 : 修改后输出极限.
else
    if (PWMA < 0)                 PWMA = 0 : 修改后输出极限.
(用相同方法计算其他 2-相.)

```

	寄存器名	功能	
输入	VExMDPRD	PWM 周期 设置	16-位 数据 (PMD PWM 周期设置值.)
	VExDTC	死区时间补偿总和	16-位 数据 (设置值从 0 到 VExMDPRD.)
	VExMODE[12]	极限使能时输出 0% 使能.	<PWMBLEN>
	VExMODE[13]	极限使能时输出 100% 使能.	<PWMFLEN>
	VExMODE[14]	PMD 电路死区时间修改控制.	<PMDDTCEN> 0: PMD 死区 时间修改禁止. 1: PMD 死区 时间修改使能.
	VExDCTS	死区时间补偿控制/状态	<ICSTS>, <IBSTS>, <IASTS> x0: 未定义, 01: 正电流, 11: 负电流

- 输出控制 /PWM 移位 1 转换

<等式>

```

OUTCR = 0x1FF                                     : 所有相中心开启
if (VExMCTLF[0] = 1)                               : 在低速 PWM 移位 1.
    if (VExSECTOR = 0,1,2,11)                       PWMB = MDPRD - PWMB      : V-相中心关闭
                                                OUTCR = 0x1F3
    else if (VExSECTOR = 3,4,5,6)                   PWMC = MDPRD - PWMC      : W-相中心关闭
                                                OUTCR = 0x1CF
    else if (VExSECTOR = 7,8,9,10)                  PWMA = MDPRD - PWMA      : U-相中心关闭
                                                OUTCR = 0x1FC
if (VExMODE[3:2] = 00,11)                           OUTCR = 0x000           : 输出关闭
else if (VExMODE[3:2] = 10)                         OUTCR = 0x015           : 短路电路刹车
VExCMPU = PWMA
VExCMPV = PWMB
VExCMPW = PWMC
VExOUTCR = OUTCR
    
```

	寄存器名	功能	
输入	VExSECTOR	扇形	4-位 数据
	VExMODE[3:2]	输出控制 工作	<OCRMD[1:0]>
	VExMINPLS	小脉宽差分	16-位 数据
输出	VExCMPU	PMD U-相 PWM 设置	16-位 数据 (值从 0 到 MDPRD.)
	VExCMPV	PMD V-相 PWM 设置	
	VExCMPW	PMD W-相 PWM 设置	
	VExOUTCR	PMD 输出控制 设置	9-位 设置
	VExEMGRS	PMD EMG 保护 释放	1-位 设置
	VExMCTLF	小脉冲 标志	<PLSLF>

2. 输出 控制 2 (任务 9)

输出控制 2 任务支持两模式输出: 正常 PWM 输出和移位 2 模式下 PWM 输出.

PWM 移位 2 模式下设置 PWM 输出时, 使能 PWM 移位 (VExFMODE <SPWMEN> = "1") 和 选择 "00"以外的值给 PWM 移位模式选择(VExFMODE<SPWMMD>).

注: 仅 1-电阻电流检测模式下 PWM 移位可以选择.

- 输出 转换

<等式>

VExMCTLF[1] = VExMCTLF[0] : 更新之前的值标志.
 VExMCTLF[0] = 0 : 清除电流标志.
 $PWMA = VExTMPREG0 \times VExMDPRD$: PMD 设置值被转换.
 $PWMB = VExTMPREG1 \times VExMDPRD$
 $PWMC = VExTMPREG2 \times VExMDPRD$

	寄存器 名	功能	
输入	VExTMPREG0	a-相电压	32-位 固定-点 数据 (0.0 到 1.0, 31 小数位)
	VExTMPREG1	b-相电压	
	VExTMPREG2	c-相电压	
	VExMDPRD	PWM 周期设置	16-位 数据 (PMD PWM 周期设置值)
输出	VExMCTLF[1:0]	低-速标志	<LAVFM>, <LAVF>

- PWM 输出 极限

<等式>

if (VExPWMMAX = 0) MAX = VExMDPRD
 else MAX = VExPWMMAX
 if ((PWMA > MAX) & (VExMDPRD > MAX)) :检查 U-相 PWM 上-限.
 if ((VExMODE[13] = 1) | (PWMA < VExMDPRD)) :确认输出极限 100% 占空比.
 PWMA = MAX
 VExMCTLF[11] = 1
 MIN = VExPWMMIN
 if ((PWMA < MIN) & (MIN > 0)) :检查 U-相 PWM 下-限.
 if ((VExMODE[12] = 1) | (PWMA > 0)) :确认输出 极限 0% 占空比.
 PWMA = MIN
 VExMCTLF[11] = 1
 (用相同方法计算其他 2-相.)

	寄存器 名	功能	
输入	VExMDPRD	PWM 周期 设置	16-位 数据 (PMD PWM 周期设置值.)
	VExPWMMAX	设置 PWM 上-限.	16-位 数据 (值从 0 到 VExMDPRD.)
	VExPWMMIN	设置 PWM 下-限.	
	VExMODE	输出 控制 工作	
	VExFMODE[12]	极限使能时输出 0%使能 .	<PWMBLEN> 0: 禁止, 1: 使能
	VExFMODE[13]	极限使能时输出 100%使能.	<PWFLEN> 0: 禁止, 1: 使能
输出	VExMCTLF[11]	PWM 输出极限超出标志	<PMMOVF>

- 死区 时间 补偿

<等式>

```

if (0 < PWMA < VExMDPRD)          DT = VExDTC
else                                 DT = 0
if (VExDTCS<IASTS> = 01)           : 正电流上
    if (VExMODE[14] = 1)           : 设置 PMD 死区时间修改.
        if (PWMA > (VExMDPRD - 2× DT)) PWMA =(VExMDPRD + PWMA)÷2
    else                             PWMA = PWMA + DT
if ((VExMODE[13] =1) & (PWMMAX < VExMDPRD)) : 确认输出极限 100% 占空比
    if (PWMA > (VExMDPRD - 1))     PWMA = VExMDPRD -1 : 修改后输出极限.
    else
        if (PWMA > VExMDPRD)       PWMA = VExMDPRD : 修改后输出极限.
        else if (VExDTCS<IASTS> =11) : 负电流上
if (VExMODE[14] =1)                 : 设置 PMD 死区时间修改 .
if (PWMA < (2× DT))                 PWMA = PWMA ÷ 2
else                                 PWMA = PWMA - DT
if ((VExMODE[12] =1) & (PWMMIN > 0)) : 确认输出极限 0% 占空比.
    if (PWMA < 1)                   PWMA = 1 : 修改后输出极限.
    else
        if (PWMA < 0)               PWMA = 0 : 修改后输出极限.
(用相同方法计算其他 2-相.)
    
```

	寄存器名	功能	
输入	VExMDPRD	PWM 周期 设置	16-位 数据 (PMD 中 PWM 周期设置值.)
	VExDTC	死区时间补偿总和	16-位 数据 (设置值 从 0 到 VExMDPRD.)
	VExMODE[12]	极限使能时输出 0%使能.	<PWMBLEN>
	VExMODE[13]	极限使能时输出 100%使能.	<PWFLEN>
	VExMODE[14]	PMD 电路死区时间控制.	<PMDDTCEN> 0: PMD 死区时间修改禁止. 1: PMD 死区时间修改使能.
	VExDCTS	死区时间补偿控制/状态	<ICSTS>, <IBSTS>, <IASTS> x0: 未定义, 01: 正电流, 11: 负电流

- 输出控制 /PWM 移位 2 转换

<等式>

```

OUTCR = 0x1FF : 所有相中心开启
if ((VExFMODE [3] = 1) & (VExFMODE [1] = 1)) : 1-电阻电流模式下 PWM 移位 2 使能.
    if (VExFMODE [15:14] = 01) : 移位 2, U-相参考
        if (PWMB > VExMDPRD ÷2) PWMB = VExMDPRD - PWMB : V-相 PWM 中心 关闭
        OUTCR = OUTCR & 0x1F3
        if (PWMC > VExMDPRD ÷2) PWMC = VExMDPRD - PWMC : W-相 PWM 中心 关闭
        OUTCR = OUTCR & 0x1CF
    else if (VExFMODE [15:14] = 10) : 移位 2, V-相参考
        if (PWMA > VExMDPRD ÷2) PWMA = VExMDPRD - PWMA : U-相 PWM 中心 关闭
        OUTCR = OUTCR & 0x1FC
        if (PWMC > VExMDPRD ÷2) PWMC = VExMDPRD - PWMC : W-相 PWM 中心 关闭
        OUTCR = OUTCR & 0x1CF
    else if (VExFMODE [15:14] = 11) : 移位 2, W-相参考
        if (PWMA > VExMDPRD ÷2) PWMA = VExMDPRD - PWMA : U-相 PWM 中心 关闭
        OUTCR = OUTCR & 0x1FC
        if (PWMB > VExMDPRD ÷2) PWMB = VExMDPRD - PWMB : V-相 PWM 中心 关闭
        OUTCR = OUTCR & 0x1F3
if (VExMODE [3:2] = 00,11) OUTCR = 0x000 : 输出 关闭
else if (VExMODE [3:2] = 10) OUTCR = 0x015 : 短路 电路 刹车
VExCMPU = PWMA
VExCMPV = PWMB
VExCMPW = PWMC
VExOUTCR = OUTCR

```

	寄存器名	功能	
输入	VExMDPRD	PWM 周期 设置	16-位 数据 (PMD 中 PWM 周期设置值.)
	VExFPWMCHG	PWM 电平 移位	16-位 固定-点 数据 (0.0 到 1.0, 15 小数 位)
	VExMODE[3:2]	输出 控制 工作	<OCRMD>
	VExMINPLS	小脉宽	16-位 数据
	VExFMODE[1]	PWM 移位 使能	<SPWMEN>
	VExFMODE[3:2]	电流 检测 模式	<IDMODE>
	VExFMODE[15:14]	PWM 移位 模式	<SPWMMD>
输出	VExCMPU	PMD U-相 PWM 设置	16-位 数据 (值 从 0 到 MDPRD.)
	VExCMPV	PMD V-相 PWM 设置	
	VExCMPW	PMD W-相 PWM 设置	
	VExOUTCR	PMD 输出 控制 设置	9-位 设置
	VExEMGRS	PMD EMG 保护 释放	1-位 设置
	VExMCTLF	小脉冲 标志	<PLSLF>

18.4.2.5 触发器生成 (任务 1)

触发器生成单元计算基于 电流检测方法的来自 PWM 设置值 (VExCMPU, VExCMPV, 和 VExCMPW) 的触发器时序, 且设置到 VExTRGCMP0 和 VExTRGCMP1 寄存器。

注: 仅 1-电阻电流检测中 VExTRGCMP0 和 VExTRGCMP1 更新。

注: PWM 移位 2 模式选择时 VExTRGCMP0, 和 VExTRGCMP1 不更新。

	寄存器名	功能	
输入	VExCMPU	PMD U-相 PWM 设置	16-位 数据 (值从 0 到 MDPRD)
	VExCMPV	PMD V-相 PWM 设置	16-位 数据 (值从 0 到 MDPRD)
	VExCMPW	PMD W-相 PWM 设置	16-位 数据 (值从 0 到 MDPRD)
	VExMDPRD	PWM 周期 设置	16-位 数据 (PMD PWM 周期设置值)
	VExTADC	AD 转换 时间	16-位 数据 (值从 0 到 MDPRD)
	VExTRGCRC	触发器 修改值	16-位 数据 (值从 0 到 MDPRD)
	VExSECTOR	扇形	4-位 数据
	VExMODE[0]	0-电流 检测	<ZIEN>
	VExMODE[3:2]	输出 控制 工作	<OCRMD>
	VExFMODE[0]	调制 模式	<C2PEN> 0: 3-调制, 1: 2-相调制
	VExFMODE[1]	PWM 移位 使能	<SPWMEN>
	VExFMODE[3:2]	电流 检测 模式	<IDMODE>
	VExFMODE[8]	触发器修改使能	<CRCEN>
	VExFMODE[15:14]	PWM 移位 模式	<SPWMMD>
	VExMCTLF[0]	低-速 标志	<LAVF>
输出	VExTRGCMP0	PMD 触发器 0 时序 设置	16-位 数据 (值从 0 到 MDPRD)
	VExTRGCMP1	PMD 触发器 1 时序 设置	16-位 数据 (值从 0 到 MDPRD)
	VExTRGSEL	PMD 触发器 选择	3-位 数据


```

else if (VExSECTORM = 0,5,6,11)    VExICADC = [电流 2]
else if (VExFMODE[3:2] = 00,01)    : 3-电阻, 2 传感器
    N = [电流 2 相信息]              :通过 1 AD 转换器被逐次检测.
    X = [电流 2]                      : 电流 4 被用在同步采样.
    if (VExFMODE[13] = 1)            N = [电流 4 相信息]
                                        X = [电流 4]
        if ( N = 1)                   VExIAADC = X
        else if ( N = 2)              VExIBADC = X
        else if ( N = 3)              VExICADC = X
[电流 3 读]
if (VExFMODE[3] ≠ 1)                : 除 1-电阻
    if ( [电流 3 相信息] = 1)        VExIAADC = [电流 3]
    else if ( [电流 3 相信息] = 2)   VExIBADC = [电流 3]
    else if ( [电流 3 相信息] = 3)   VExICADC = [电流 3]

[电流 固定-点 转换]
IA = VExIAO - VExIAADC
IB = VExIBO - VExIBADC
IC = VExICO - VExICADC
if (VExFMODE[3:2] = 10,11)          : 1-电阻
    if (VExMCTLF[1] = 0)            : 正常 PWM
        if (VExSECTORM = 0,1,10,11) IA = -IA
        else if (VExSECTORM = 6,7,8,9) IB = -IB
        else f (VExSECTORM = 2,3,4,5) IC = -IC
    else if (VExMCTLF[1] = 1)      : PWM 移位 1
        if (VExSECTORM = 1,2,5,6,9,10) IA = -IA
        IB = -IB
        IC = -IC

[电流 3 计算]
N = 6 - N - [电流 1 相信息]        :计算电流 3 相数.
if ( N = 1)                         IA = - IB - IC
else if ( N = 2)                     IB = - IC - IA
else if ( N = 3)                     IC = - IA - IB

[电流 储存]
VExTMPREG0 = IA
VExTMPREG1 = IB
VExTMPREG2 = IC

```

	寄存器名	功能	
输入	(DC 电压)	ADC 转换 结果输入	16-位 数据 (转换结果储存在上 12 位中.)
	(电流 1)		
	(电流 2)		
	(电流 3)		
	(电流 4)		
	VExSECTORM	扇形信息	4-位 数据
	VExMODE[1]	零-电流检测	<ZIEN>
	VExMODE[4]	VDC 储存寄存器	<VDCSEL>
	VExFMODE[3:2]	电流检测模式	<IDMODE>
	VExFMODE[13]	同步采样控制	<SADCEN> 0: 同步采样禁止., 1: 同步采样使能.
VExMCTLF[0]	低-速 标志	<LAVFM>	

	寄存器名	功能	
输出	VExVDC	DC 电源电压	16-位 固定-点 数据 (0.0 到 1.0, 15 小数位)
	VExVDCL	DC 电源电压	16-位 固定-点 数据 (0.0 到 1.0, 15 小数位)
	VExIAADC	a-相电流转换结果	16-位 数据 (结果储存在上 12 位.)
	VExIBADC	b-相电流转换结果	
	VExICADC	c-相电流转换结果	
	VExTMPREG0	a-相电流	32-位 被固定到-点 数据 (-1.0 到 1.0, 31 小数位)
	VExTMPREG1	b-相电流	
VExTMPREG2	c-相电流		
内部	VExIAO	a-相零-电流转换结果	16-位 数据 (结果储存在上 12 位.)
	VExIBO	b-相零-电流转换结果	
	VExICO	c-相零-电流转换结果	

- 电流 极性 决定

<等式>

```

if ( VExMODE [15] = 1)                                     : 电流极性决定使能.
    IA = VExTMPREG0
    if ( <IASTS> = xx0)                                     : 极性未定义.
        if ( IA ≥ |HYS| )      <IASTS> = 001             : 正极极性被决定.
        else if ( IA ≤ -|HYS| ) <IASTS> = 111           : 负极极性被决定.
    else if ( <IASTS> = 001)                                   : 之前的极性为正极.
        if ( IA ≤ -|HYS| )      <IASTS> = 111           : 极性被改变为负极.
        else if ( IA < -HYS )   <IASTS> = 011           : 极性被改变为负极.(滞后区域)
    else if ( <IASTS> = 101)
        if ( IA ≥ |HYS| )      <IASTS> = 001           : 极性正极.(滞后区域外)
    else if ( <IASTS> = 111)                                   : 之前的极性为负极.
        if ( IA ≥ |HYS| )      <IASTS> = 001           : 极性被改变为正极.
        else if ( IA > HYS )   <IASTS> = 101           : 极性被改变为正极.(滞后区域)
    else if ( <IASTS> = 011)
        if ( IA ≤ -|HYS| )      <IASTS> = 111           : 负极 (滞后区域外)
    
```

(用相同方法计算其他 2-相.)

	寄存器名	功能	
输入	VExHYS	电流极性决定滞后	16-位 被固定到-点 数据 (0.0 到 1.0, 15 小数位)
	VExDCTS	死区时间补偿控制/状态	<ICSTS>, <IBSTS>, <IASTS> xx0: 未定义, x01: 正电流, x11: 负电流
	VExMODE [15]	电流极性决定控制	<IPDEN> 0: 决定禁止. 1: 决定使能.
	VExTMPREG0	a-相电流	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数位)
	VExTMPREG1	b-相电流	
VExTMPREG2	c-相电流		
输出	VExDCTS	死区时间补偿控制/状态	<ICSTS>, <IBSTS>, <IASTS> x0: 未定义, 01: 正电流, 11: 负电流

2. 输入处理2 (任务10)

输入处理2 任务支持 3-电阻电流检测 (3-相检测和 2-相检测) 和 2-传感器电流检测. 也支持 PWM 移位 2 模式下 PWM 输出时 1-电阻电流检测. 各相中电流方向可以被指定.

注: 输入处理2 任务中零-电流 检测模式不可用.

注: 仅 1-电阻电流检测模式 PWM 移位可以选择.

- 输入转换

<等式>

[VDC 固定-点 转换/储存]

```
if (VExMODE[4] = 0          VExVDC = [DC 电压] >>1
else                          VExVDCL = [DC 电压] >>1
```

[电流 1 读]

```
if ( [电流 1 相信息] = 1)    VExIAADC = [电流 1]
else if ( [电流 1 相信息] = 2) VExIBADC = [电流 1]
else if ( [电流 1 相信息] = 3) VExICADC = [电流 1]
```

[电流 2 读]

N = [电流 2 相信息] : 通过 1 ADC 被逐次检测.

X = [电流 2]

if (VExFMODE[13] = 1 N = [电流 4 相信息] : 电流 4 被用在同步采样.

X = [电流 4]

if (N = 1)

VExIAADC = X

else if (N = 2)

VExIBADC = X

else if (N = 3)

VExICADC = X

[电流 3 读]

if (VExFMODE[3:2] = 00) : 仅 3-相检测执行时.

if ([电流 3 相信息] = 1) VExIAADC = [电流 3]

else if ([电流 3 相信息] = 2) VExIBADC = [电流 3]

else if ([电流 3 相信息] = 3) VExICADC = [电流 3]

else

N = 6 - N - [电流 1 相信息] : 电流 3 被计算除 3-相检测执行时.

[电流固定-点 转换/储存]

IA = VExIAO - VExIAADC

if (VExFMODE[5] = 1)

IA = -IA

: Ia 电流 检测 设置

IB = VExIBO - VExIBADC

if (VExFMODE[6] = 1)

IB = -IB

: Ib 电流 检测 设置

IC = VExICO - VExICADC

if (VExFMODE[7] = 1)

IC = -IC

: Ic 电流 检测 设置

if (VExFMODE[3:2] ≠ 00)

: 电流 3 被计算除 3-相检测执行时.

if (N = 1)

IA = - IB - IC

else if (N = 2)

IB = - IC - IA

else if (N = 3)

IC = - IA - IB

VExTMPREG0 = IA
 VExTMPREG1 = IB
 VExTMPREG2 = IC

	寄存器名	功能	
输入	(DC 电压)	输入 AD 转换结果.	16-位 数据 (转换 结果储存在上 12 位.)
	(电流 1)		
	(电流 2)		
	(电流 3)		
	(电流 4)		
	VExMODE[4]	VDC 储存 寄存器	<VDCSEL>
	VExFMODE[3:2]	电流检测 模式	<IDMODE>
	VExFMODE[7:5]	电流检测的极性	<ICPLMD>, <IBPLMD>, <IAPLMD> 0: 电阻 模式(In = VExInO - VExInADC) 1: 传感器 模式(In = VExInADC - VExInO) 注: n = A, B, 或 C
VExFMODE[13]	同步采样控制	<SADCEN> 0: 同步 采样禁止. 1: 同步 采样使能.	
输出	VExVDC	DC 电源 电压	16-位 固定-点 数据 (0.0 到 1.0, 15 小数 位)
	VExVDCL	DC 电源 电压	16-位 固定-点 数据 (0.0 到 1.0, 15 小数 位)
	VExIAADC	a-相电流转换结果	16-位 数据 (结果储存在上 12 位.)
	VExIBADC	b-相电流转换结果	
	VExICADC	c-相电流转换结果	
	VExTMPREG0	a-相电流	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG1	b-相电流	
VExTMPREG2	c-相电流		
内部	VExIAO	a-相零-电流转换结果	16-位 数据 (结果储存在上 12 位.)
	VExIBO	b-相零-电流转换结果	
	VExICO	c-相零-电流转换结果	

- 电流 极性 决定

<等式>

```

if ( VExMODE [15] = 1)                                     : 使能极性决定.
    IA = VExTMPREG0
    if ( <IASTS> = xx0)                                     : 极性未定义.
        if ( IA ≥ |HYS| )      <IASTS> = 001             : 极性正极.
        else if ( IA ≤ -|HYS| ) <IASTS> = 111           : 极性负极.
    else if ( <IASTS> = 001)                                 : 之前的极性正极.
        if ( IA ≤ -|HYS| )      <IASTS> = 111           : 极性被改变为负极.
        else if ( IA < -HYS )    <IASTS> = 011           : 极性被改变为负极.(滞后 区域)
    else if ( <IASTS> = 101)
        if ( IA ≥ |HYS| )      <IASTS> = 001           : 极性正极.(滞后区域外)
    else if ( <IASTS> = 111)                                 : 之前的极性负极.
        if ( IA ≥ |HYS| )      <IASTS> = 001           : 极性被改变为正极.
        else if ( IA > HYS )    <IASTS> = 101           : 极性被改变为正极.(滞后区域)
    else if ( <IASTS> =011)
        if ( IA ≤ -|HYS| )      <IASTS> = 111           : 极性负极.(滞后区域外)

```

(用相同方法计算其他 2-相.)

	寄存器 名	功能	
输入	VExHYS	电流极性决定滞后	16-位 固定-点 数据 (0.0 到 1.0, 15 小数 位)
	VExDCTS	死区时间补偿控制/状态	<ICSTS>, <IBSTS>, <IASTS> xx0: 未定义, 01: 正电流, 11: 负电流
	VExMODE[15]	电流极性决定控制	<IPDEN> 0: 决定禁止. 1: 决定使能.
	VExTMPREG0	a-相电流	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG1	b-相电流	
	VExTMPREG2	c-相电流	
输出	VExDCTS	死区时间补偿控制/状态	<ICSTS>, <IBSTS>, <IASTS> xx0: 未定义, x01: 正电流, x11: 负电流

18.4.2.7 输入 电流转换 (相 转换/协调轴 转换)

输入电流转换由两任务: 相转换和协调轴转换组成.

1. 输入相转换(任务 3)

输入相转换任务计算 I_α 和 I_β 基于 I_a , I_b , 和 I_c .

<等式>

```

if ( VExFMODE[12] = 0 )                                : 使能相转换.
    VExTMPREG3 = VExTMPREG0                             : 计算  $I_\alpha$ .
    VExTMPREG4 = 1 + sqrt(3) × VExTMPREG1 - 1 + sqrt(3) × VExTMPREG2 : 计算  $I_\beta$ .
else if ( VExFMODE[12] = 1 )                            : 禁止相转换.
    VExTMPREG3 = VExTMPREG0
    VExTMPREG4 = VExTMPREG1
    
```

	寄存器名	功能	
输入	VExTMPREG0	a-相电流	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG1	b-相电流	
	VExTMPREG2	c-相电流	
	VExFMODE[12]	相转换禁止.	<PHCVDIS>
输出	VExTMPREG3	α -轴 电流	32-位 固定到-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG4	β -轴 电流	

2. 输入协调轴转换(任务 4)

输入协调轴转换任务中, I_d 和 I_q 值从 I_α , I_β , VExSINM, 和 VExCOSM 被计算.

a. 协调轴转换

<等式>

```

if ( VExMCTLFM[5] = 0 )                                : 检查小脉冲标志.
    VExID = VExCOSM × VExTMPREG3 + VExSINM × VExTMPREG4 : 计算  $I_d$  值.
    VExIQ = - VExSINM × VExTMPREG3 + VExCOSM × VExTMPREG4 : 计算  $I_q$  值
    
```

	寄存器名	功能	
输入	VExTMPREG3	α -轴 电流	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExTMPREG4	β -轴 电流	
	VExSINM	theta 角 sine	16-位 固定-点 数据 (-1.0 到 1.0, 15 小数 位)
	VExCOSM	theta 角 cosine	
	VExMCTLF[5]	小脉冲 标志	<PLSLFM>
输出	VExID	d-轴 电流	32-位 固定-点 数据 (-1.0 到 1.0, 31 小数 位)
	VExIQ	q-轴 电流	

b. ATAN 计算

<ATANMD> = "10"时, ATAN 任务计算 d-轴电流偏向角.

<ATANMD> = "11 "时, ATAN 任务计算 d-轴和 q- 轴诱导电压. 然后使用基于这些结果的电机电压等式计算偏向角.

<等式>

```

if ( VExMODE [6] = 1 )                                :使能偏向角计算.
    LD = VExCLD × [VExCLG 设置]                       : d-轴 电感
    LQ = VExCLQ × [VExCLG 设置]                       : q-轴 电感
    R = VExCR × [VExCRG 设置]                         : 电阻
    VDIV = VExVD - (VExID × R - VExOMEGA × VExIQ × LQ) : d-轴诱导 电压
    VQIV = VExVQ - (VExIQ × R + VExOMEGA × VExID × LD) : q-轴诱导 电压
    if ( VExMODE [5] = 1 ) VExDELTA = ATAN2(VQIV, VDIV) : 计算诱导电压偏向角.
else                                                    VExDELTA = ATAN2(VExIQ, VExID) : 计算电流偏向角.

```

	寄存器 名	功能	
输入	VExCLD	电机 d-轴 电感	16-位 固定-点 数据 (11 小数 位)
	VExCLQ	电机 q-轴 电感	
	VExCR	电机 电阻 值	
	VExCLG	电感 范围 设置	000: 一 次, 001: 1/2 ⁴ -折叠, 010: 1/2 ⁸ -折叠, 011: 1/2 ¹² -折叠
	VExCRG	电阻 范围 设置	100: 1/2 ¹⁶ -折叠, 101 到 111: 保留
	VExOMEGA	转 速	16-位 固定-点 数据 (15 小数 位)
	VExVD	d-轴 电压	32-位 固定-点 数据 (-1.0 到 1.0 和 31 小数 位)
	VExVQ	q-轴 电压	
	VExMODE[6:5]	ATAN 工作 模式 设置	<ATANMD> 0x: 计算 禁止. 10: 计算 Id 和 Iq 偏向角. 11: 计算 d-轴和 q-轴诱导电压偏向角.
输出	VExDELTA	偏向 角输出	16-位 数据 (-180 到 180, 0x8000 到 0x7FFF)

18.4.2.8 别的任务

1. ATNA2 (反正切功能 2) (任务 12)

ATAN2 任务计算从基于开始点到 X-Y 协调上(X,Y)点画直线的 X-轴的起始点的角.

<等式>

```

X = VExTMPREG4
Y = VExTMPREG5
Z = ATAN ( |Y| ÷ |X|)           : 反正切计算, 0 到 90°
if ( X < 0 & Y ≥ 0 ) Z = 0x00008000 - Z       : 第二象限 (90 到 180°)
if ( X < 0 & Y < 0 ) Z = 0xFFFF8000 + Z      : 第三象限 (-90 到 -180°)
if ( X ≥ 0 & Y < 0 ) Z = - Z                  : 第四象限 (0 到 -90°)
if ( X = Y = 0 )   Z = 0x00000000           : 原点上输出( 0° )
VExTMPREG5 = Z

```

	寄存器名	功能	
输入	VExTMPREG4	输入 X	32-位 表示的数据
	VExTMPREG5	输入 Y	
输出	VExTMPREG5	相 值	32-位 数据 (0xFFFF8000 到 0x00008000 (-180 到 180°))

2. SQRT (平方根功能) (任务 13)

SQRT 任务输出基于使用 0.0 到 4.0 输入平方根的计算的 0.0 到 2.0 值.

<等式>

```

X = VExTMPREG5                                : 输入 ( 0 到 4.0 )
N = 0
if ( X < 0x2000 ) N = 1
if ( X < 0x0800 ) N = 2
if ( X < 0x0200 ) N = 3
if ( X < 0x0080 ) N = 4
if ( X < 0x0020 ) N = 5
if ( X < 0x0008 ) N = 6
if ( X < 0x0002 ) N = 7
if ( X ≥ 0x8000 ) N = -1
X = X × 2N                                    : 正常化 ( 0.25 到 1.0 )
if ( X > 0x7FFF ) X = 0x7FFF
Z = SQRT( X )                                  : 计算 X 的平方根.
if ( X = 0 ) Z = 0                             输出为 0.5 到 1.0 的范围.
Z = Z ÷ 2N                                    : 反向 转换( 0 到 2.0 )
VExTMPREG5 = Z

```

	寄存器 名	功能	
输入	VExTMPREG5	输入 值	指 32-位 固定-点格式上数据. (0.0 到 4.0, 15 小数 位) 0x0000_0000 到 0x0001_FFFF
输出	VExTMPREG5	平方根值	32-位固定-点 数据 (0.0 到 2.0, 15 小数 位) 0x0000_0000 到 0x0000_FFFF

18.5 VE 通道, PMD 和 ADC 间的组合

可用向量引擎, PMD, 和 ADC 之间组合受取决于向量引擎通道的限制.

注意此组合取决于电流检测方法改变.

3-电阻和 2-传感器电流检测中,可以使用两 ADCs 执行同步采样. 此时, 被用的电流输入为电流 1 和电流 4.

表 18-8 向量引擎和 PMD 之间组合

向量引擎	PMD 通道 0	PMD 通道 1
通道 0	o	-
通道 1	-	o

表 18-9 向量引擎和 ADC (无同步采样) 之间组合

向量引擎		ADC 单元				ADC 单元			
通道	电流检测 VExMODE <IDMODE[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	00	电流 1	电流 2	电流 3	DC 电压	-	-	-	-
	01	电流 1	电流 2	-	DC 电压	-	-	-	-
	1x	电流 1	电流 2	-	DC 电压	-	-	-	-
1	00	-	-	-	-	电流 2	电流 1	电流 3	DC 电压
	01	-	-	-	-	电流 2	电流 1	-	DC 电压
	1x	-	-	-	-	电流 2	电流 1	-	DC 电压

表 18-10 向量引擎和 ADC (有同步采样) 之间组合

向量引擎		ADC 单元				ADC 单元			
通道	电流检测 VExMODE <IDMODE[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	00	电流 1	-	电流 3	DC 电压	电流 4	-	-	-
	01	电流 1	-	-	DC 电压	电流 4	-	-	-
	1x	-	-	-	-	-	-	-	-
1	00	-	电流 4	-	-	-	电流 1	电流 3	DC 电压
	01	-	电流 4	-	-	-	电流 1	-	DC 电压
	1x	-	-	-	-	-	-	-	-

19. 编码器 输入 电路 (A-ENC)

19.1 概要

编码器输入电路 (A-ENC) 支持 6 工作模式: 编码器模式, 传感器模式 (3 类型), 定时器模式, 和相计数器模式.

A-ENC 有以下功能:

- 支持增量型编码器(Z-相可用) 和霍尔传感器 ICs.
- 转缘 检测 (2-相输入: 4-折叠, 3-相输入: 6-折叠)
- 旋转方向 检测 电路
- 计数器 功能 (事件计数, 定时器计数, 和相计数 功能)
- 2 类型中断输出
- 中断请求输出使能/禁止信号和事件标志
- 输入电路包含数字-噪声 过滤器
- 可以用 PMD PWM 信号同步执行采样.
- 过零检测 (PMD 电路被用) 在以矩形波驱动工作的 BLDC 电机的 BEMF (诱导电压).
- 可用作通用 32-位 定时器/捕捉.

19.2 方块 图

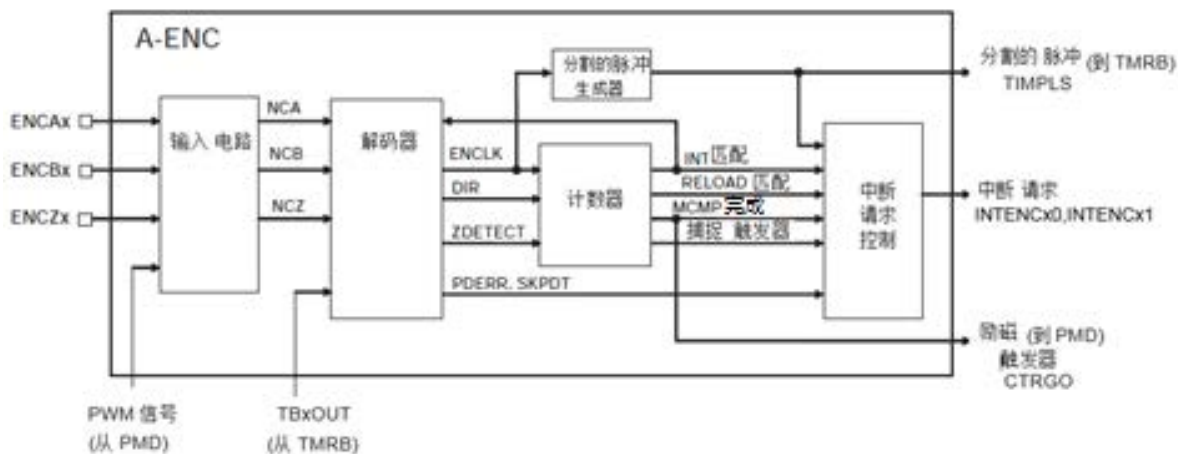


图 19-1 编码器输入电路方块图

19.3 寄存器

19.3.1 寄存器表

下表列出控制寄存器和其地址:

关于基地址, 参考"内存映射"章节中"外设功能基地址表".

寄存器名		地址 (基+)
ENC 控制 寄存器	ENxTNCR	0x0000
RELOAD 比较 寄存器	ENxRELOAD	0x0004
INT 比较 寄存器	ENxINT	0x0008
计数器/捕捉 寄存器	ENxCNT	0x000C
MCMP 比较 寄存器	ENxMCMP	0x0010
相计数率 寄存器	ENxRATE	0x0014
状态 寄存器	ENxSTS	0x0018
输入处理控制 寄存器	ENxINPCR	0x001C
采样 延迟 寄存器	ENxSMPDLY	0x0020
输入 监视器 寄存器	ENxINPMON	0x0024
采样 时钟 控制 寄存器	ENxCLKCR	0x0028
中断 请求 控制 寄存器	ENxINTCR	0x002C
中断 事件 标志 寄存器	ENxINTF	0x0030

19.3.2 ENxTNCR (ENC 控制 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	CMPSEL	UDMD		TOVMD	MCMPMD
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	DECMD		SDTEN	-	MODE			P3EN
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	TRGCAPMD	SFTCAP	ENCLR	ZESEL	
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	ZEN	ENRUN	-	-	-	ENDEV		
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-29	-	R	读作 "0".
28	CMPSEL	R/W	定时器模式中计数器清除条件. 0: ENxINT 寄存器匹配 1: ENxRELOAD 寄存器匹配
27-26	UDMD	R/W	传感器模式 (相计数) 或相计数模式中设置递增递减计数设置. 00: 递增 计数 01: 递减 计数 1x: 用 ENxRATE 寄存器设置递增或递减计数设置 "1x"设置到此时时, 如果 ENxRATE<RATE> < 0, 计数器设置为递减计数设置. 如果 < RATE > ≥ 0, 计数器设置为递增计数设置.
25	TOVMD	R/W	RE LOAD 寄存器匹配计数器值时(时间到工作设置) 设置计数器工作. [传感器 模式 (定时器 计数)] 0: 计数继续. 1: 计数停止. 计数器停止时, 应该通过软件清除匹配重启计数器. [定时器模式, 传感器 模式 (相计数), 相计数器 模式] 0: 计数器清除且计数继续. 1: 计数停止 计数器停止时, 应该通过软件清除匹配重启计数器. [编码器 模式] 无论 TOVMD 设置, 电机驱动在 CW 方向时, 计数器清除且计数继续, 电机驱动在 CCW 方向时, 计数继续. [传感器 模式 (事件 计数)] 此位不被用作匹配 RELOAD 寄存器.
24	MCMPMD	R/W	[传感器 模式 (定时器计数), 或定时器模式] ENxMCMP 寄存器比较模式. 0: 比较匹配 (ENxMCMP<MCMP> = 计数器 值 1: 尺寸比较 (ENxMCMP<MCMP> ≤ 计数器 值 除传感器模式 (定时器计数) 或定时器模式中, 设置为 <MCMPMD>=0.

位	位符	类型	功能
23-22	DECMD	R/W	[编码器模式或传感器模式] 设置解码器检测方向。 00: CW 或 CCW 缘检测 检测输入信号(ENCAx, ENCBx, 和 ENCZx)的变化。 01: CW 缘检测 检测转缘检测中之前的状态到当前状态的输入信号的变化。(维持检测结果) 10: CCW 缘检测 检测转缘检测中之前的状态到当前状态的输入信号的变化。(维持检测结果) CW 或 CCW 缘检测 11: 检测转缘检测中之前的状态到当前状态的输入信号的变化。(维持检测结果) 定时器模式或相计数模式下设置为<DECMD>=00.
21	SDTEN	R/W	设置传感器模式下 3-相 解码时的跳跃检测。 0: 检测禁止。 1: 检测使能。 跳跃被检测时, 设置错误标志 (ENxSTS<SKPDT>).
20	-	R	读作 "0".
19-17	MODE [2:0]	R/W	设置工作模式。 000: 编码器模式 001: 传感器模式 (事件计数) 010: 传感器模式 (定时器计数) 011: 定时器模式 100: 保留 101: 保留 110: 传感器模式 (相计数) 111: 相计数器模式
16	P3EN	R/W	[传感器模式](注 1) 设置解码模式 (2-相/3-相输入选择) 0: 2-相解码 1: 3-相解码
15-13	-	R	读作 "0".
12	TRGCAPMD	R/W	[传感器模式 (定时器计数 或 相计数), 定时器模式, 或 相计数器模式] 触发器捕捉工作选择 0: 捕捉且清除计数器 1: 仅捕捉 传感器模式(定时器计数 或 相计数)下选择转缘检测时捕捉工作.定时器模式 或 相计数器模式下 z-输入使能时选择捕捉工作。 软件捕捉被用时, 计数器不清除.
11	SFTCAP	W	[传感器模式 (定时器计数, 相计数), 定时器模式, 或 相计数器模式] 执行软件捕捉。 1: 捕捉计数器值。 此位为 "1"时, 捕捉计数器值. 要获取捕捉的值, 读 ENxCNT 寄存器。 写入 "0" 无意义. 读作 "0".
10	ENCLR	W	清除计数器。 1: 清除 此位为"1"时, 计数器清除时"0". 计数器清除后, 计数器重启工作。 写入 "0" 无意义. 读作 "0".
9-8	ZESEL	R/W	[定时器模式或相计数器模式] Z-相输入使能时选择检测缘。(Z-相输入/ PSGI 输入) 00: 保留 01: 上升缘被检测。 10: 下降缘被检测。 11: 双缘被检测。

位	位符	类型	功能
7	ZEN	R/W	[编码器模式, 定时器模式, 或相计数器模式] (注 2) 使能/禁止 Z-相输入. 0: ENCZ 输入禁止. 1: ENCZ 输入使能.
6	ENRUN	R/W	使能/禁止编码器输入电路. 0: 禁止 1: 使能 <ENRUN>="1", <ZDET> 清除为 "0"且编码器输入电路也使能时. <ENRUN>="0"时, 编码器输入电路禁止.
5-3	-	R	读作 "0".
2-0	ENDEV[2:0]	R/W	[编码器 模式, 或传感器 模式 (事件 计数)] 设置转缘脉冲分割的输出 (TIMPLS) 的比例. 根据此设置, 分割转缘脉冲且此脉冲被用作中断事件. 000: 分割通过 1 100: 分割通过 16 001: 分割通过 2 101: 分割通过 32 010: 分割通过 4 110: 分割通过 64 011: 分割通过 8 111: 分割通过 128

注 1: 设置 <P3EN>为"0" 除传感器 模式.

注 2: 设置 <ZEN>为"0" 除编码器 模式, 定时器模式, 和相计数器 模式.

工作模式被 <MODE[2:0]>, <P3EN>, 和 <ZEN>决定. 总共有 12 类型. 下表列出工作模式设置:

<MODE[2:0]>	<ZEN>	<P3EN>	输入 引脚	模式
000	0	0	A, B	编码器 模式
	1		A, B, Z	编码器 模式 (Z-输入的使用)
001	0	0	A, B	传感器 模式 (事件 计数)
		1	A, B, Z	传感器 模式 (事件 计数和 Z-输入的使用)
010	0	0	A, B	传感器 模式 (定时器计数和 2-相输入)
		1	A, B, Z	传感器 模式 (定时器计数和 3-相输入)
011	0	0	-	定时器模式
	1		Z	定时器模式 (Z-输入的使用)
110	0	0	A, B	传感器 模式 (相计数和 2-相输入)
		1	A, B, Z	传感器 模式 (相计数和 3-相输入)
111	0	0	-	相计数 模式
	1		Z	相计数 模式 (Z-输入的使用)

19.3.3 ENxRELOAD (RELOAD 比较 寄存器)

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

位	位符	类型	功能
31-16	RELOADH	R/W	<p>[编码器 模式] 设置 最大计数器值. 设置 输入脉冲每转 (后 $\times 4 - 1$) 数.</p> <p>[传感器 模式 (相计数) 或相计数器 模式] 设置计数器最大值 (每转计数范围) .</p> <p>[传感器 模式 (定时器计数) 或定时器模式] 此位被用作比较计数器的寄存器. 如果匹配发生, 中断发生. 设置 32 位的上 16 位作比较.</p> <p>[传感器 模式 (事件 计数)] 此位不被用.</p>
15-0	RELOADL	R/W	<p>[传感器 模式 (定时器计数) 或定时器模式] 设置 32-位的低 16 位作比较.</p> <p>[除 传感器 模式 (定时器计数) 或定时器模式] 此位不被用.</p>

19.3.4 ENxINT (INT 比较 寄存器)

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

位	位 符	类型	功能
31-16	INTH	R/W	<p>[编码器 模式或传感器 模式 (事件 计数)] 设置 16 位上比较计数器的值。 计数器 匹配<INTH>值时, INT 匹配信号发生.此时,中断可以生成.</p> <p>[传感器模式 (定时器计数) 或定时器模式] 设置 32 位上比较计数器的值. 设置 32 位的上 16 位到 <INTH>作比较. 计数器匹配 <INTH>值时, INT 匹配信号发生. 此时, 中断可以生成. PWM 同步采样使能时此匹配信号可以被用作位置检测开始信号.</p> <p>[传感器 模式 (相计数) 或相计数器 模式] 设置 16 位上比较计数器的值。 计数器匹配<INTH>值时, INT 匹配信号发生.此时, 中断可以生成.</p>
15-0	INTL	R/W	<p>[传感器 模式 (定时器计数) 或定时器模式] 设置 32-位 比较的低 16 位.</p> <p>[除传感器 模式 (定时器计数) 或你定时器模式] 此位不被用.</p>

19.3.5 ENxCNT (计数器 寄存器)

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

位	位符	类型	功能
31-16	CNTH	R	<p>[编码器模式或传感器模式 (事件 计数)] 读转缘脉冲的计数值.</p> <p>[传感器模式 (定时器计数或相计数)] 在转缘检测/在 ENCZx 输入缘检测读捕捉的值或软件捕捉的值.</p> <p>[定时器模式或相计数器模式] 在 ENCZ 输入缘检测读捕捉的值或软件捕捉的值.</p> <p>[传感器模式 (定时器计数) 或定时器模式] 读 32-位 捕捉时上 16 位的值.</p>
15-0	CNTL	R	<p>[编码器模式, 传感器模式 (事件 计数或相计数), 或相计数模式] 此位不被用.</p> <p>[传感器模式 (定时器计数) 或定时器模式] 读 32-位 捕捉时低 16 位的值.</p>

19.3.6 ENxMCMP (MCMP 计数器 寄存器)

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

位	位 符	类型	功能
31-16	MCMPL	R/W	<p>[传感器 模式 (定时器计数) 或定时器模式] 设置 32 位上比较计数器的值. 设置 32 位的上 16 位到 <MCMPL> 作比较. 中断可以生成.</p> <p><u>比较大小(<MCMPLD>=1)</u> ENxMCMP ≤ 计数器值建立时输出 MCMP 完成信号. 此模式下, 每次写入寄存器输出一 完成信号.</p> <p><u>比较匹配模式(<MCMPLD>=0)</u> ENxMCMP = 计数器值建立时输出 MCMP 完成信号.</p> <p>[除 传感器 模式 (定时器计数) 或定时器模式] 设置 16 位上比较计数器的值. 中断可以生成. ENxMCMP<MCMPL> = 计数器值时输出 MCMP 完成信号. ENxINTCR<MCMPIE>使能时, 输出换相触发器信号到 PMD 电路.</p>
15-0	MCMPL	R/W	<p>[传感器 模式 (定时器计数) 或定时器模式] 设置 32 位的低 16 位作比较. [除 传感器 模式 (定时器计数) 或定时器模式] 此位不被用.</p>

19.3.7 ENxRATE (相 计数 率 寄存器)

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

位	位 符	类型	功能
31-15	-	R	读作 "0".
15-0	RATE	R/W	<p>[传感器 模式 (相计数) 或相计数器 模式]</p> <p>设置计数器计数频率.</p> <p>时钟频率生成: $f_{sys} \times \langle \text{RATE} \rangle / 2^{16}$</p> <p>取决于 ENxTNCR<UDMD> 设置, < RATE > 值可以被指定 为表示位或非表示位. 如果 < RATE > 为负值, 计数器递减计数.</p> <p><UDMD>=0x: 非表示, 0 或大/小于 1.0 (0x0000 到 0xFFFF)</p> <p><UDMD>=1x: 表示, -0.5 或大/小于 0.5 (0x8000 到 0x7FFF)</p> <p><UDMD>=1x 时, 设置两的互补到< RATE >.</p>

19.3.8 ENxSTS (状态 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	REVERR	UD	ZDET	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	-	-	-	SKPDT	PDERR	INERR
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-15	-	R	读作 "0".
14	REVERR	R	[传感器 模式 (定时器计数或相计数)] 双缘被检测时设置<UD>反向标志. (注 1)(注 2) 0: - 1: <UD>反向被检测. 如果 ENxTNCr<ENRUN> = "0",始终设置"0".
13	UD	R	[编码器 模式或传感器 模式 (事件 计数, 定时器计数, 或相计数)] 设置电机转动方向. 0: CCW (逆顺时针) 1: CW (顺时针) 电机转动为 CW 方向时, 此位为"1". 电机转动为 CCW 方向时, 此位为"0". 如果 ENxTNCr<ENRUN> = "0",始终设置"0".
12	ZDET	R	检测 ENCZ 输入通过. 0:编码器输入已经使能后 Z-输入不被检测 . 1: Z-输入被检测. ENxTNCr<ENRUN>="0"时此位清除.
11-3	-	R	读作 "0".
2	SKPDT	R	跳跃 检测使能时检测跳跃检测标志. (注 1) 0: 未被检测 1: 跳跃被检测.
1	PDERR	R	[编码器 模式, 传感器 模式 (事件 计数, 定时器计数, 或相计数)] 检测解码检测错误标志. (注 1) 0: 未被检测 1: 错误被检测.
0	INERR	R	[传感器 模式 (事件 计数, 定时器计数, 或相计数)] 检测异常 输入 错误. (注 1) 0: 无异常 错误 1: 异常 错误 3-相解码中所有 3-相输入为 "低" 或"高"时此位设置为"1".

注 1: 通过读此寄存器清除标志.

注 2: 模式过渡后, 先确保读标志然后清除为"0".

19.3.9 ENxINPCR (输入 处理 控制 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	NCT						
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	PDSTP	PDSTT	-	-	-	SYNCNCZEN	SYNCSPLMD	SYNCSPLEN
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-15	-	R	读作 "0".
14-8	NCT	R/W	设置噪声消除时间. 设置范围: 0 到 127 (0x00 到 0x7F) 消除时间: 设置值 × 采样时钟 周期 (用 <SPLCKS> 设置) 此位设置为"0"时, 噪声消除不工作. PWM-关闭缘采样模式中的采样时钟为 PWM 信号.
7	PDSTP	W	[传感器 模式 (定时器计数或相计数)] 设置位置检测停止命令在 PWM 同步采样. (BFMF 检测 控制) 1: 停止位置检测 此位设置为 "1"时, 位置检测开始. 写入 "0" 无意义. 读作 "0".
6	PDSTT	W	[传感器 模式 (定时器计数或相计数)] 在 PWM 同步采样设置位置检测开始命令. (BEMF 检测控制) 1: 开始位置 检测 此位设置为"1 时", 位置检测开始. 写入 "0" 无意义. 读作 "0".
5-3	-	R	读作 "0".
2	SYNCNCZEN	R/W	采样执行在 PWM-开启期间时控制噪声消除计数器. 0: 计数器 停止在 PWM-关闭 期间. 1: 计数器 停止且清除在 PWM-关闭 期间. 此设置被用来使能 PWM 同步采样 (<SYNCSPLEN>=1) 和在 PWM-开启期间 (<SYNCSPLMD> =0)选择 PWM 采样.
1	SYNCSPLMD	R/W	选择 PWM 同步采样. 0: 在 PWM-开启期间采样 1: 在 PWM-关闭缘采样 PWM 同步采样使能时(<SYNCSPLEN>=1) 此设置使能.
0	SYNCSPLEN	R/W	使能 PWM 同步采样 0: 持续采样 1: PWM 同步采样 此位使能 PMD 同步采样. PWM 同步采样设置在传感器 模式 (定时器计数或相计数) 时, 如果 <SYNCSPLEN>设置为 "1", 解码工作中使能 BEMF 检测 控制.

19.3.10 ENxSMPDLY (采样 延迟 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7-0	SMPDLY	R/W	采样 禁止 期间 设置 范围: 0 到 255 (0x00 到 0xFF) 禁止 时间: <SMPDLY> 设置 值 × 采样 周期 (用 <SPLCKS> 设置) 在 PWM-开启期间 (ENxINPCR<SYNCSPLEN>=1, ENxINPCR<SYNCSPLMD>=0) 结束后设置采样禁止 期间 PWM 开启缘.

19.3.11 ENxINPMON (输入 监视器 寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	DETMONZ	DETMONB	DETMONA	-	SPLMONZ	SPLMONB	SPLMONA
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-7	-	R	读作 "0".
6	DETMONZ	R	监视 NCZ 位置检测状态. 存储位置检测时 NCZ 值.
5	DETMONB	R	监视 NCB 位置检测状态. 存储位置检测时 NCB 值.
4	DETMONA	R	监视 NCA 位置检测状态. 存储位置检测时 NCA 值.
3	-	R	读作 "0".
2	SPLMONZ	R	监视噪声消除后 ENCZx 状态. 存储噪声消除后 ENCZx 输入的 NCZ 状态.
1	SPLMONB	R	监视噪声消除后 ENCBx 状态. 存储噪声消除后 ENCBx 输入的 NCB 状态.
0	SPLMDNA	R	监视噪声消除后 ENCAx 状态. 存储噪声消除后 ENCAx 输入的 NCA 状态.

19.3.12 ENxCLKCR (采样 时钟 控制 寄存器)

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

位	位 符	类型	功能
31-2	-	R	读作 "0".
1-0	SPLCKS	R/W	设置采样频率. 00: fsys 01: fsys/2 10: fsys/4 11: fsys/8 设置 ENCAx 输入, ENCBx 输入, 和 ENCZx 输入的采样频率. PWM 同步采样时, 设置关闭缘采样(ENxINPCR<SYNCSPLEN>=1 和 ENxINPCR <SYNCSPLMD>=1) 时 此位禁止.

19.3.13 ENxINTCR (中断控制寄存器)

	31	30	29	28	27	26	25	24
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位符	-	-	MCMPIE	RLDIE	CMPIE	ERRIE	CAPIE	TPLSIE
复位后	0	0	0	0	0	0	0	0

位	位符	类型	功能
31-6	-	R	读作 "0".
5	MCMPIE	R/W	使能/禁止 MCMP 完成中断. 0: 禁止 1: 使能 此位为 "1"时, 在 MCMP 完成发生 INTENCx1.
4	RLDIE	R/W	使能/禁止 RELOAD 中断. 0: 禁止 1: 使能 此位为 "1"时, 计数器匹配 RELOAD 寄存器值时 INTENCx1 发生. 编码器模式或传感器模式 (事件 计数)中 RELOAD 中断不发生.
3	CMPIE	R/W	使能/禁止 INT 中断 0: 禁止 1: 使能 此位为 "1"时,计数器匹配 INT 寄存器值时 INTENCx1 发生.
2	ERRIE	R/W	使能/禁止检测错误中断 0: 禁止 1: 使能 此位为 "1"时, 缘检测 错误 (PDERR) 发生, 或跳跃被检测 (SKPDT) 时, INTENCx0 发生. 定时器模式或相计数器模式中中断不发生.
1	CAPIE	R/W	使能/禁止捕捉触发器中断. 0: 禁止 1: 使能 此位为 "1"时, 如果计数器值由于外部 触发器 (ENCZx 输入) 或转 缘 脉冲 (ENCLK)被捕捉, INTENCx0 发生. 编码器模式或传感器模式 (事件 计数)中断不发生.
0	TPLSIE	R/W	使能/禁止 转动的分割缘中断. 0: 禁止 1: 使能 此位为 "1"时,通过转缘分割的脉冲, INTENCx0 发生. 仅编码器模式和传感器模式(事件 计数)下中断发生.

19.3.14 ENxINTF (中断 事件 标志 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	-	-	MCMPF	RLDCPF	INTCPF	ERRF	CAPF	TPLSF
复位后	0	0	0	0	0	0	0	0

位	位 符	类型	功能
31-6	-	R	读作 "0".
5	MCMPF	R	MCMP 比较 完成 标志 0: 无标志 1: 标志生成.
4	RLDCPF	R	RELOAD 匹配 标志 0: 无标志 1: 标志生成. 编码器模式或传感器 模式 (事件 计数)下此位不设置.
3	INTCPF	R	INT 匹配 标志 0: 无标志 1: 标志生成.
2	ERRF	R	检测 错误 标志 0: 无标志 1: 标志生成. 定时器模式或相计数器模式下此位不设置.
1	CAPF	R	捕捉 标志 0: 无标志 1: 标志生成. 此位不通过软件捕捉设置. 编码器模式或传感器模式(事件 计数)下此位不设置.
0	TPLSF	R	转缘分割的脉冲标志 0: 无标志 1: 标志生成. 编码器模式或传感器模式(事件 计数)下此位使能.

注:各标志通过使能事件发生设置,且通过读 ENxINTF 寄存器清除.

19.4 工作描述

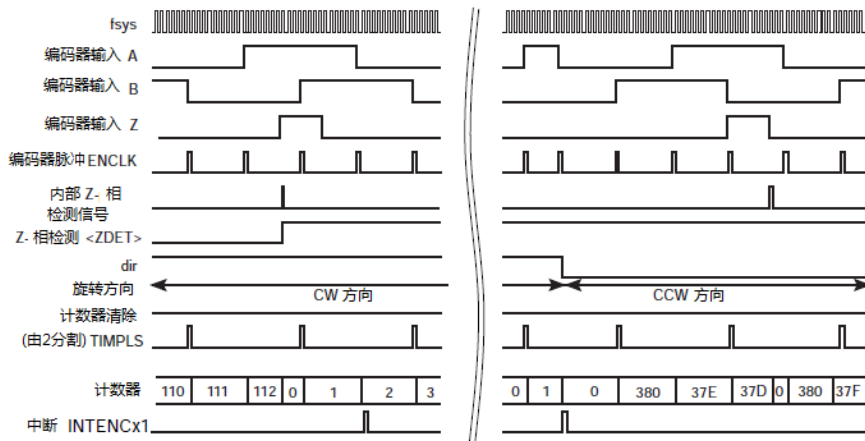
19.4.1 编码器模式

A-ENC 支持决定从增量编码器 (AB 编码器 或 ABZ 编码器) 相输入的高-速位置传感器 (相决定).

- 执行转缘检测, 然后输出分割的脉冲和中断请求.
- 转缘脉冲计数或指定的计数值时生成中断请求.
- 决定转动方向.
- 递增/递减 计数 (通过转动方向控制)
- 设置计数范围.
- 设置检测转动方向.
- 异常错误检测标志可用.

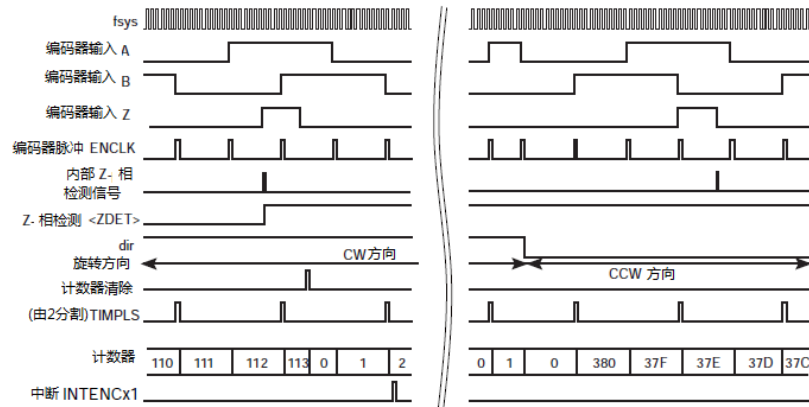
1. ENCZ 输入使能. ($EN_xTNCR<ZEN> = 1$)

$EN_xRELOAD<RELOADH[15:0]> = 0x0380$, $EN_xINT<INTH[15:0]> = 0x0002$



2. ENCZ 输入禁止. ($EN_xTNCR<ZEN> = 0$)

$EN_xRELOAD<RELOADH[15:0]> = 0x0380$, $EN_xINT<INTH[15:0]> = 0x0002$



编码器模式中，增量编码器输入应该被连接到 ENCA, ENCB, 和 ENCZ 引脚。编码器加倍 ENCA 和 ENCB 信号到 4 计数编码器脉冲。

CW 转动 (比如, A-相有 90-度相位提前于 B-相) 期间, 计数器递增计数;计数器值到达 <RELOAD>值时, 在下一个 ENCLK 计数器清除为 "0"。

CCW 转动 (比如, A-相有 90-度相位滞后于 B-相) 期间, 计数器递减计数;计数器值到达 "0x0000"时, 在下一个 ENCLK<RELOAD>值设置到计数器。

而且, <ZEN> = "1"时, CW 转动期间在 Z-相上升缘计数器清除为"0" 且 CCW 转动期间在 Z-相下降缘. 如果 ENCLK 匹配 Z-相检测时序, 编码器计数器清除为"0"无需增量或减量。

"1" 写入 ENxTNCR<ENCLR>时, 计数器清除为"0"。

CW 转动期间 ENxSTS<UD> 设置为 "1"且 CCW 转动期间清除为"0" 。

检测方向 (CW 方向或 CCW 方向) 可以用 ENxTNCR<DECMD[1:0]> 设置. 除<DECMD>="00"时, 转 缘被检测基于比较输入状态 ENxINPMON<DETMONA><DETMONB><DETMONZ> 在以前的被检测 缘和电流输入值之间。

然后输出 ENCLK 信号分割的信号 (TIMPLS)。

ENxINTCR<CMPIE> = "1"时, 如果 <ENINTH>值和计数器值变为等于, INTENCx1 中断发生。

ENxINTCR<MCMPIE> = "1"时, ENxMCMP<MCMPH[15:0]> 值和计数器值变为等于 INTENCx1 中 断发生。

然而, <ZEN> = "1"时, ENxSTS<ZDET> = "0"期间匹配中断不发生. 编码器输入使能后, 如果编码器计数器检测到第一个 Z-信号, <ZDET>设置为"1"。

ENxTNCR<ENRUN> = "0"时<ZDET> 和 ENxSTS<UD>清除为 "0"。

19.4.2 传感器 模式

计数器支持低-速位置传感 (过零决定) 处理 2-相霍尔传感器输入和 3-相霍尔传感器输入. 有三个模式: 事件计数模式, 定时器计数 模式, 和相计数模式。

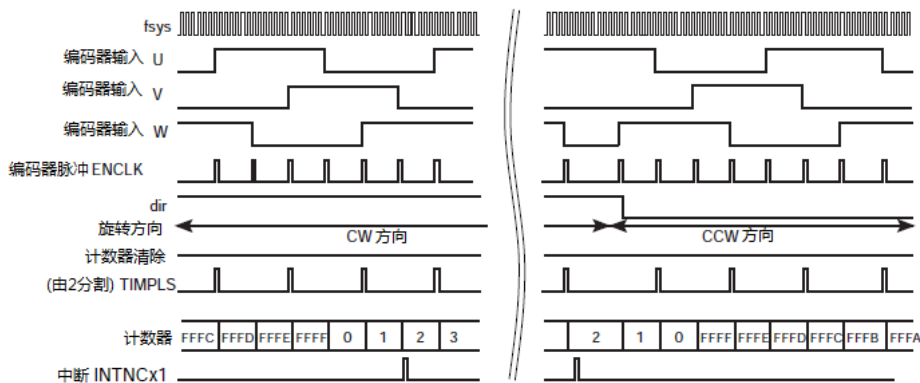
定时器计数模式或相计数模式中, 无刷 DC 电机以矩形波形从 PMD 电路被驱动时, A-ENC 可以执行使用 PWM 同步采样的诱导电压过零检测。

19.4.2.1 事件计数

使用转缘检测计数器计数.

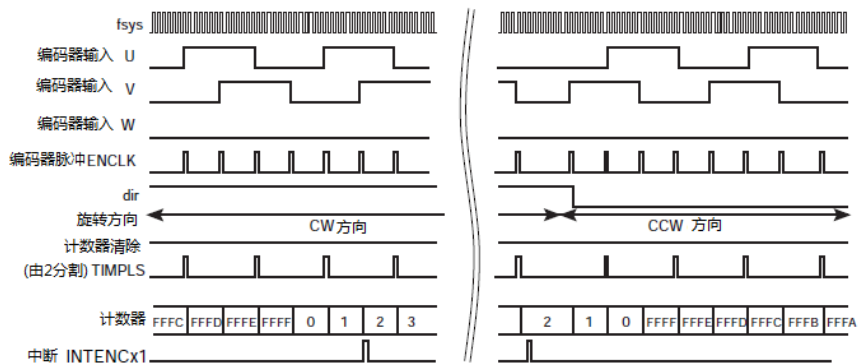
- 执行转缘检测且输出分割的脉冲和中断请求.
- 以转缘脉冲计数或指定的计数值生成中断请求.
- 决定转动方向.
- 设置递增递减计数 (由转动方向决定).
- 指定转到检测方向.
- 异常错误检测标志

1. 3相解码 (ENxTNCR<P3EN> = 1)



2. 2-相解码 (ENxTNCR<P3EN> = 0)

ENxINT<INTH[15:0]> = 0x0002)



霍尔传感器输入(U,V, 和 W) 应该被连接到 ENCAx, ENCBx, 和 ENCZx.

<P3EN> = "0"时, 计数器通过 4 倍 2-相输入 (ENCAx 或 ENCBx); <P3EN> = "1"时, 计数器通过 6 倍 3-相输入 (ENCAx, ENCBx, 或 ENCZx). 然后, 计数器计数霍尔传感器脉冲.

CW 转动 (比如, ENCA 有 90-相位提前于 ENCB 通道)期间, 计数器执行递增计数. 计数器值到达"0xFFFF"时, 在下一个 ENCLK 编码器计数器清除为 "0".

CCW 转动 (比如, ENCA 有 90-度相位滞后于 ENCB) 期间, 编码器计数器执行递减计数. 计数器值到达"0x0000"时, 在下一个 ENCLK 计数器设置为 "0xFFFF".

"1"写入 ENxTNCR<ENCLR>时, 计数器清除为"0".

CW 转动被检测时, ENxSTS<UD> 设置为 "1". CCW 转被检测时, ENxSTS<UD>清除为"0".

检测方向 (CW 方向或 CCW 方向) 可以用 ENxTNCR<DECMD[1:0]>设置.除<DECMD>="00"时, 转缘被检测基于比较输入状态 ENxINPMON <DETMONA><DETMONB><DETMONZ> 在以前的被检测缘和电流输入值之间.

然后输出 ENCLK 信号分割的信号 (TIMPLS) .

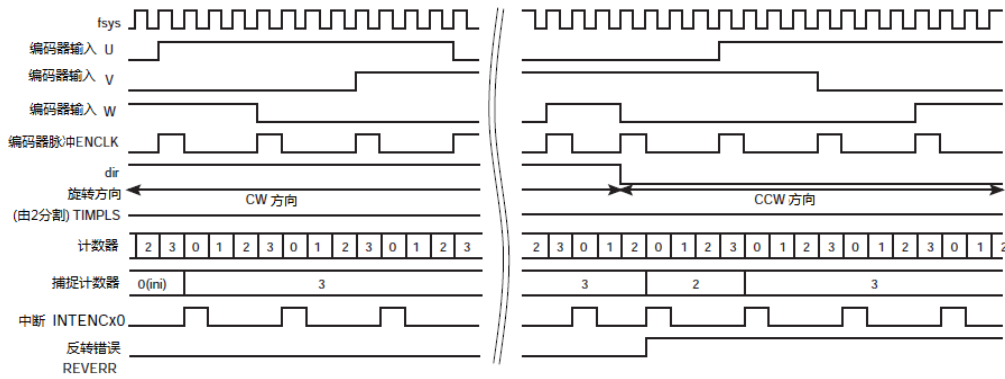
ENxINTCR<CMPIE> = "1"时, 如果 <ENINTH> 和计数器值变为等于, INTENCx1 中断发生.

ENxINTCR<MCMPIE> = "1"时, 如果 <ENMCMPH>值和计数器值变为等于, INTENCx1 中断发生.

19.4.2.2 定时器 计数

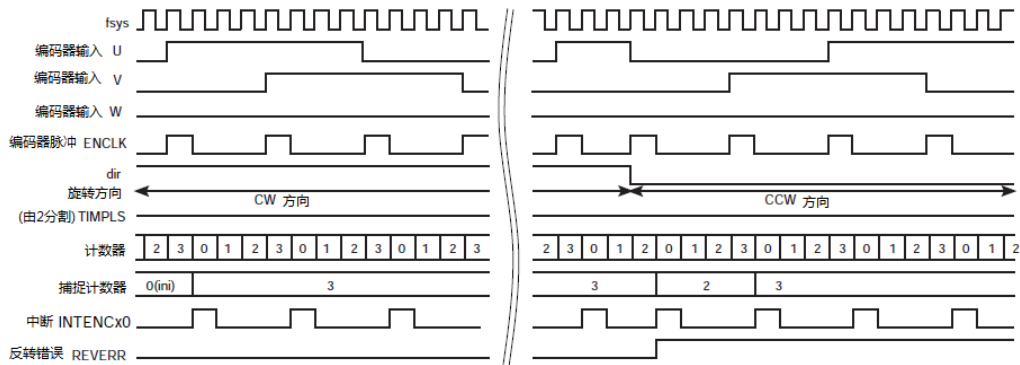
1. 3-相解码 (ENxTNCR<P3EN> = 1)

$$ENxINT<INTH[15:0]> = 0x0002$$



2. 2-相解码 (ENxTNCR<P3EN> = 0)

$$ENxINT<INTH[15:0]> = 0x0002$$



霍尔传感器输入(U,V, 和 W) 应该被连接到 ENCA_x, ENCB_x, 和 ENCZ_x.
 $\langle P3EN \rangle = "0"$ 时, 计数器通过 4倍 2-相输入 (ENCA_x 或 ENCB_x); $\langle P3EN \rangle = "1"$ 时, 计数器通过 6倍 3-相输入 (ENCA_x, ENCB_x, 或 ENCZ_x). 然后, 计数器生成 4 折叠或 6-折叠转缘脉冲 (ENCLK).

计数器在 ENCLK 始终执行递增计数且清除为"0".

"1"写入 EN_xTNCR \langle ENCLR \rangle 时, 计数器清除为 "0".

在 ENCLK 捕捉计数器值. 捕捉的值可以从 ENxCNT 寄存器读.

"1"写入 EN_xTNCR \langle SFTCAP \rangle 时, 捕捉计数器值.捕捉时序可以被用户指定. 捕捉的值可以从 ENxCNT 寄存器读.

维持 ENxCNT 寄存器 (捕捉的值) 值无论 EN_xTNCR \langle ENRUN \rangle 值.

CW 转动被检测时, EN_xSTS \langle UD \rangle 设置为"1". CCW 转动被检测时, EN_xSTS \langle UD \rangle 清除为"0".

\langle ENRUN $\rangle = "0"$ 时, \langle UD \rangle 清除为 "0". 如果转动方向反转, EN_xSTS \langle REVERR \rangle 设置为"1". 此标志通过读清除.

检测方向 (CW 方向或 CCW 方向) 可以用 EN_xTNCR \langle DECMD[1:0] \rangle 设置.

除 \langle DECMD $\rangle = "00"$ 时, 转缘被检测基于比较输入状态 EN_xINPMON

\langle DETMONA \rangle \langle DETMONB \rangle \langle DETMONZ \rangle 在以前的被检测缘和电流输入值之间.

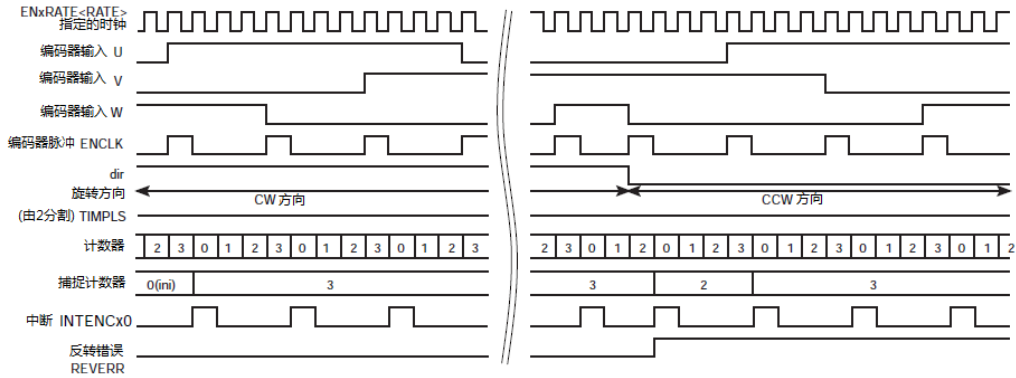
EN_xINTCR \langle RLDIE $\rangle = "1"$ 时, 如果 EN_xRELOAD \langle RELOADH[15:0] \rangle 值和计数器值变为等于, INTENC_{x1} 中断发生.

EN_xINTCR \langle CMPIE $\rangle = "1"$ 时, 如果 EN_xINT \langle INTH[15:0] \rangle 值和计数器值变为等于, INTENC_{x1} 中断发生.

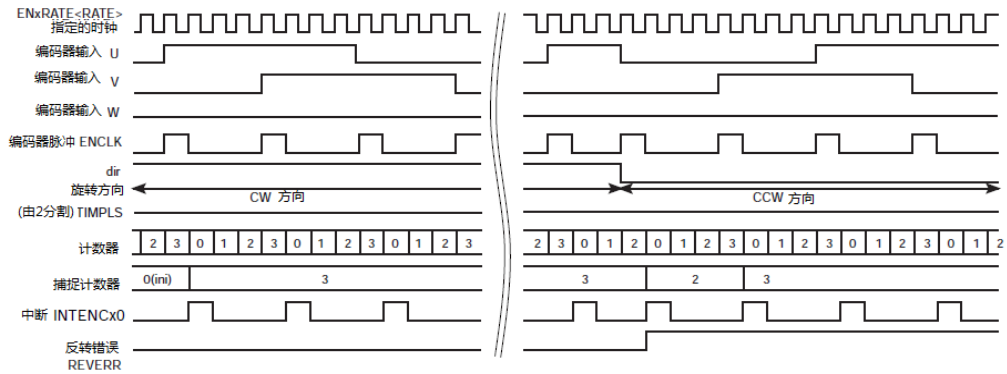
EN_xINTCR \langle MCMPIE $\rangle = "1"$ 时, 如果 EN_xMCMP \langle MCMPH[15:0] \rangle 值和计数器值变为等于, INTENC_{x1} 中断发生. EN_xINTCR \langle MCMPMD $\rangle = "1"$ 时, 如果计数器值为 \langle MCMPH \rangle 值或更多, INTENC_{x1} 中断发生.

19.4.2.3 相计数

1. 3-相解码 (EN_xTNCR \langle P3EN $\rangle = 1$)



2. 2-相解码 (ENxTNCR<P3EN> = 0)



霍尔传感器输入 (U, V, 和 W) 应该被连接到 ENCA_x, ENCB_x, 和 ENCZ_x.
 <P3EN> = "0"时, 计数器通过 4 倍 2-相输入 (ENCA_x 或 ENCB_x); <P3EN> = "1"时, 计数器通过 6 倍 3-相输入 (ENCA_x, ENCB_x, 或 ENCZ_x). 然后, 生成 4-折叠或 6-折叠转缘脉冲 (ENCLK).

计数器可以在指定的频率通过 <UDMD> 设置或 RATE 寄存器设置来设置 递增/递减计数. 计数器设置为递增计数时, 如果计数器值和 <RELOADH>值变为等于, 计数器清除为"0". 计数器设置为递减计数时, 如果计数器值到达 "0x0000", <RELOADH>到计数器.

"1" 写入 <ENCLR>时, 计数器清除为 "0".

<TOVMD> = "1"时, 计数器停止在<RELOADH>值.

通过 ENCLK 捕捉计数器值. 捕捉的值可以从 ENxCNT 寄存器读.

"1"写入 <SFTCAP>时, 捕捉计数器值. 捕捉时序可以由用户指定. 捕捉的值可以从 ENxCNT 寄存器读.

维持 ENxCNT 寄存器 (捕捉的值) 值无论值<ENRUN>.

CW 转动被检测时, <UD>设置为"1". CCW 转动被检测时, <UD>清除为"0". <ENRUN>= "0"时, <UD>清除为"0". 如果旋转方向反转, ENxSTS<REVERR> 设置为"1". 此标志通过读清除.

检测方向 (CW 方向或CCW 方向) 可以用 $EN_xTNCR<DECMD[1:0]>$ 设置. 除 $<DECMD>="00"$ 时, 转缘被检测基于比较输入状态 $EN_xINPMON <DETMONA><DETMONB><DETMONZ>$ 在以前的被检测缘和电流输入值之间.

$EN_xINTCR<CMPIE> = "1"$ 时, 如果 $<EN_xINT>$ 值和计数器值变为等于, $INTENC_x1$ 中断发生.

$EN_xINTCR<MCMPIE> = "1"$ 时, 如果 $<EN_xMCMP>$ 值和计数器值变为等于, $INTENC_x1$ 中断发生.

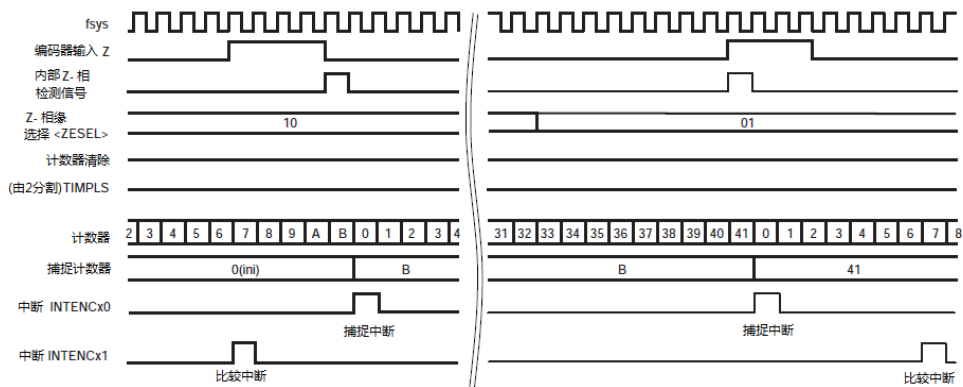
19.4.3 定时器 模式

此模式可以被用作通用 32-位 定时器.

- 32-位 递增计数器 (使用 f_{sys} 时钟计数)
- 计数器 清除 控制 (软件 清除, 比较 匹配 清除, 和外部 触发器)
- 使用比较功能匹配中断发生.
- 捕捉 功能 : 外部 触发器 捕捉 (中断可以生成), 软件 捕捉

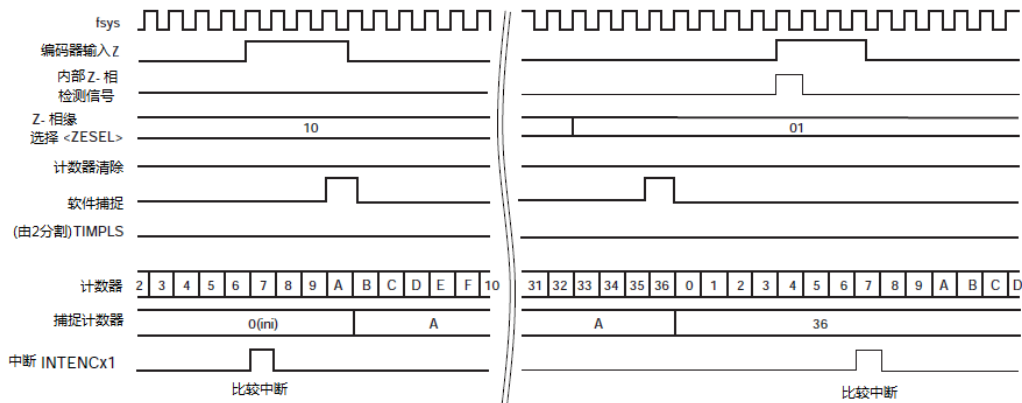
1. ENCZ 输入使能. ($EN_xTNCR<ZEN> = 1$)

$EN_xINT<INTH[15:0]><INTL[15:0]> = 0x0000_0006$



2. ENCZ 输入禁止. ($EN_xTNCR<ZEN> = 0$)

$EN_xINT<INTH[15:0]><INTL[15:0]> = 0x0000_0006$



<ZEN> = "1"时, Z-输入可以工作为外部触发器. <ZEN> = "0"时, 外部触发器不用.

计数器始终执行递增计数.

"1" 写入 ENxTNCR<ENCLR>时, 计数器清除为"0".

<ZEN> = "1"时, 如果 ENxTNCR<ZESEL>设置为"01", 计数器在 Z-相上升缘清除. <ZESEL> = "10"时, 计数器在 Z-相下降缘清除. <ZESEL> = "11"时, 计数器在 Z-相缘清除.

Z-相缘被检测时, 捕捉计数器值. 捕捉的值可以从 ENxCNT 寄存器读.

"1"写入 ENxTNCR<SFTCAP>时, 捕捉计数器值. 捕捉时序可以被用户指定. 捕捉的值可以从 ENxCNT 寄存器读.

维持 ENxCNT 寄存器值(捕捉的值) 无论 ENxTNCR<ENRUN>值. 捕捉的值清除事件仅为复位.

ENxINTCR<RLDIE>="1"时, ENxRELOAD<RELOADH[15:0]><RELOADL[15:0]> 值和计数器值变为等于, INTENCx1 中断发生.

ENxINTCR<CMPIE> = "1"时, ENxINT<INTH[15:0]><INTL[15:0]> 值和计数器值变为等于, INTENCx1 中断发生.

ENxINTCR<MCMPIE> = "1"时, ENxMCMP<MCMPH[15:0]><MCMPH[15:0]> 和计数器值变为等于, INTENCx1 中断发生.

ENxTNCR<MCMPLD> = "1"时, 如果计数器值为<MCMPH[15:0]><MCMPH[15:0]> 或更多, INTENCx1 中断发生.

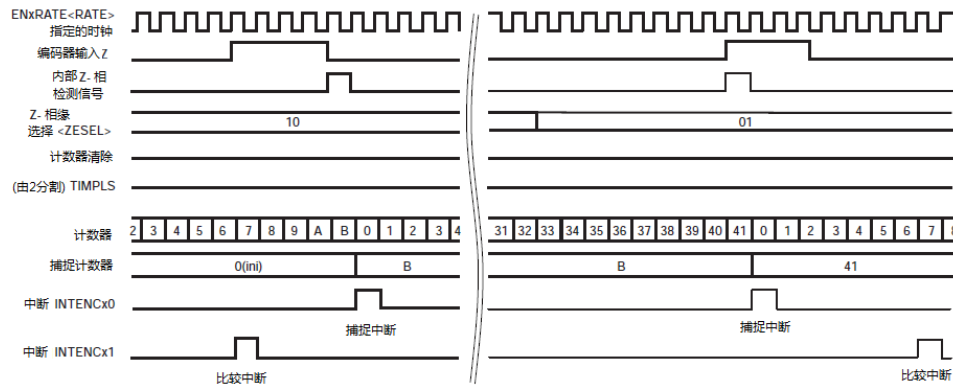
19.4.4 相 计数器 模式

此模式可以被用作可以由指定的频率控制的 16-位 计数器.

- 递增递减计数控制可用.
- 设置比较功能和匹配中断请求
- 清除使用 ENCZ 输入的计数器/捕捉功能, 然后发生中断请求

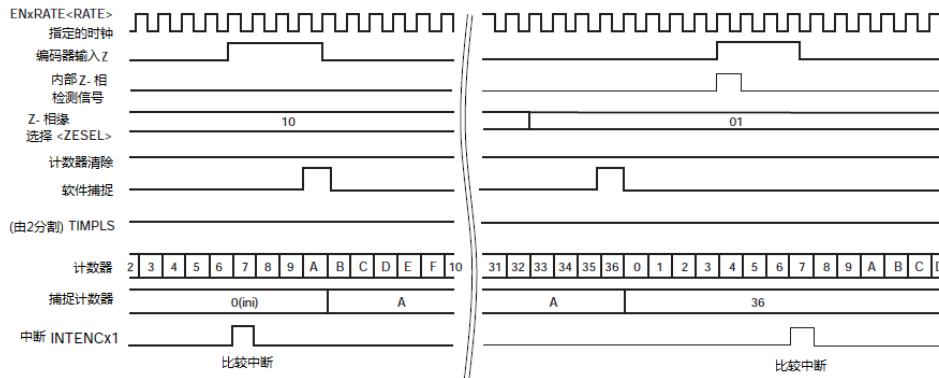
1. ENCZ 使能. (ENxTNCR<ZEN> = 1)

$$ENxINT<INTH[15:0]> = 0x0006$$



2. ENCZ 输入禁止.(ENxTNCR<ZEN> = 0)

$$ENxINT<INTH[15:0]> = 0x0006$$



<ZEN> = "1"时, Z-输入可以工作为外部触发器. <ZEN> = "0"时, 外部触发器不用.

计数器在指定的频率用 ENxTNCR<UDMD> 和 ENxRATE 寄存器控制递增/递减计数.

计数器执行递增计数时, 如果计数器值变为等于 ENxRELOAD <RELOADH[15:0]>值, 计数器清除为 "0".

计数器执行递减计数时, 如果计数器值到达 "0x0000", <RELOADH>值设置到计数器.

ENxTNCR<TOVMD> = "1"时, 计数器停止在 <RELOADH>值.

"1"写入 ENxTNCR<ENCLR>时, 计数器清除为 "0".

<ZEN> = "1"时, 如果 ENxTNCR<ZESEL> = "01", 在Z-相上升缘计数器清除为 "0"; 如果 <ZESEL> = "10", 在Z-相下降缘计数器清除为 "0"; 如果 <ZESEL> = "11", 在双缘计数器清除为 "0".

Z-相缘被检测时, 捕捉计数器值. 捕捉的值可以从 ENxCNT 寄存器读.

"1"写入 ENxTNCR<SFTCAP>时, 捕捉计数器值. 捕捉可以执行在指定的时序. 捕捉的值可以从 ENxCNT 寄存器读.

维持 ENxCNT 寄存器值(捕捉的值) 无论 ENxTNCR<ENRUN>值. 捕捉的值清除事件仅为复位.

ENxINTCR<CMPIE> = "1"时, 如果 ENxINT<INTH>值和计数器值变为等于, INTENCx1 中断发生.

ENxINTCR<MCMPIE> = "1"时, 如果 ENxMCMP<MCMPH>值和计数器值变为等于, INTENCx1 中断发生.

19.4.4.1 相差分 计数 模式

相计数器模式中, <P3EN> = <ZEN> = "1"时, 计数器进入相差分计数器模式. 此模式中, 递增/递减计数器用 TMRB 输出和 ENCZ 输入控制.

- TMRB 输出和 ENCZ 输入等于时, 计数器工作设置为递增计数.TMRB 输出和 ENCZ 输入不等于时, 计数器工作设置为递减计数.
- 在 TMRB 输出缘捕捉计数器值. 此时计数器可以清除且中断请求可以生成.

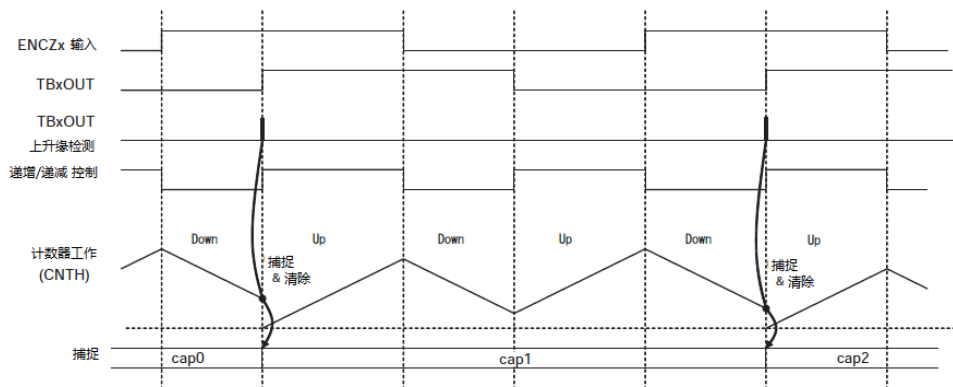


图 19-2 相计数器 模式下工作(相差分)

TBxOUT 信号缘被检测时, 捕捉的值和计数器清除.此时,检测缘用 ENxTNCR<ZESEL>设置.

捕捉的值指 ENCZ 输入和 TBxOUT 信号之间相差分. 标准值 (捕捉值为 "0") 被设为 ENCZ 输入和 TBxOUT 信号之间相差分为 1/4 周期.

19.5 各电路工作描述

19.5.1 输入电路

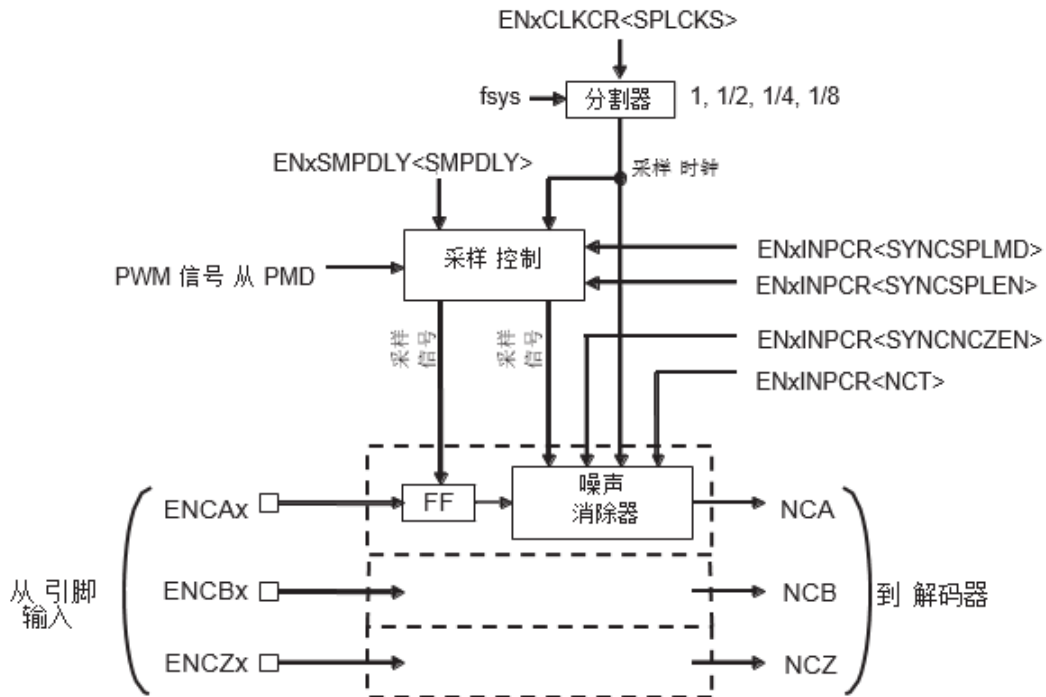


图 19-3 输入电路配置

输入电路用指定的采样信号从引脚 (ENA,ENB,ENZ) 采样输入来降低使用数字噪声-过滤器噪声。

19.5.1.1 采样时钟

采样时钟可以从 f_{sys} , $f_{sys}/2$, $f_{sys}/4$, 或 $f_{sys}/8$ 用 $\langle \text{SPLCKS} \rangle$ 选择。

19.5.1.2 采样模式

1. 持续采样 ($\text{ENxINPCR}\langle \text{SYNCSPLEN} \rangle = "0"$)

持续采样中，通过采样时钟输入电路采样输入时钟。

2. PWM 同步采样 ($\text{ENxINPCR}\langle \text{SYNCSPLEN} \rangle = "1"$)

PWM 同步采样中，输入电路从 PMD 电路用 PWM 信号同步采样输入时钟。

- PWM-开启期间采样 ($\text{ENxINPCR}\langle \text{SYNSPLMD} \rangle = "0"$)

PWM-开启期间采样中，输入电路采样输入信号通过用 $\text{ENxCLKCR}\langle \text{SPLCKS} \rangle$ 选择的指定的采样时钟给 PWM-开启期间。

- PWM 关闭-缘采样 ($\text{ENxINPCR}\langle \text{SYNSPLMD} \rangle = "1"$)

PWM 信号用作采样信号，且在 PWM 信号关闭-缘输入电路采样输入信号。

- PWM 开启-延迟设置

输入电路采样开启-延迟期间时 PWM 开启-延迟设置可以用 ENxSMPLDSY<SMPDLY> 设置.

$$\text{延迟时间} = \text{<SMPDLY>} \times \text{采样时钟周期}$$

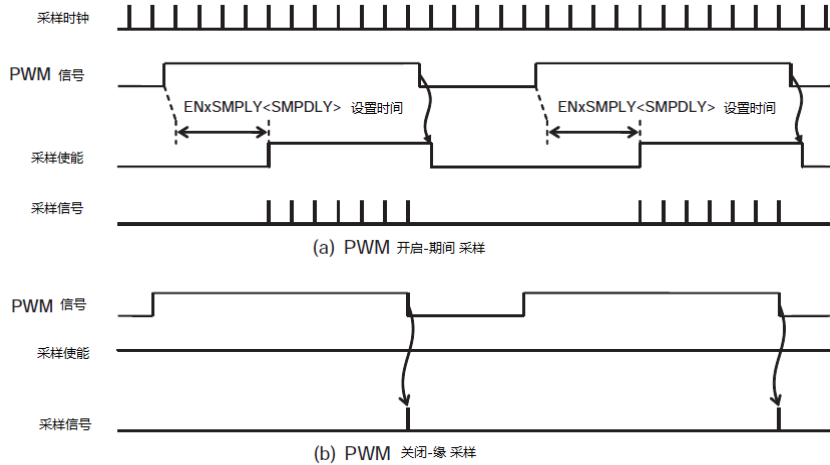


图 19-4 PWN 同步 采样

19.5.1.3 噪声 消除

1. 噪声 消除 时间

噪声消除时间可以用 ENxINPCR<NCT[6:0]> 设置. 噪声消除时间可以如下计算:

$$\text{噪声消除时间} = \text{<NCT>} \times \text{采样时钟周期}$$

注: "0" 设置到 <NCT>时, 噪声消除禁止.

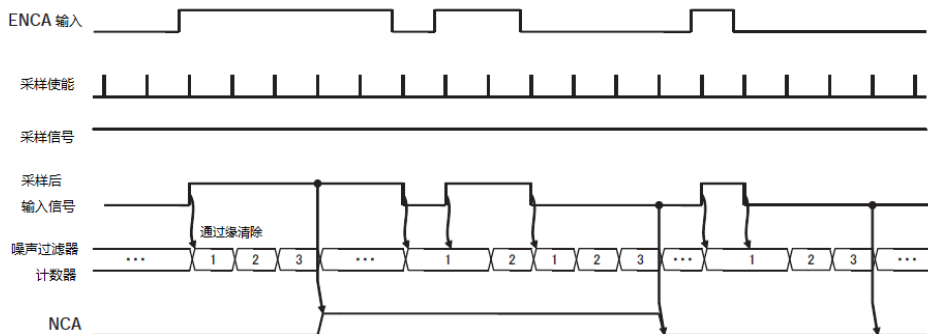
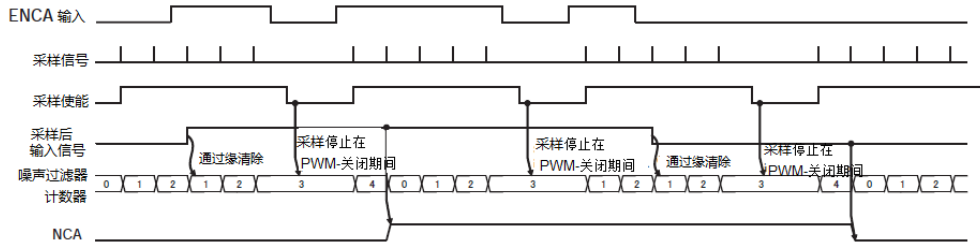
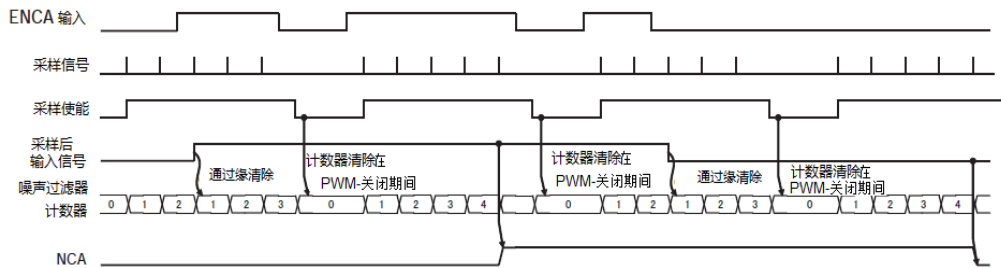


图 19-5 噪声消除 (持续, <NCT>=3)

2. 在 PWM-开启期间输入电路采样输入信号时噪声消除 .

- 噪声消除定时器停止在PWM-关闭期间 ($EN_xINPCR<SYNCCNCZEN>="0"$)
- 噪声消除定时器清除在PWM-关闭期间 ($EN_xINPCR<SYNCCNCZEN>="1"$)

图 19-6 噪声消除 (采样在 PWM-开启期间. 噪声消除定时器停止在 PWM-关闭期间. $<NCT>=4$)图 19-7 噪声消除 (采样在 PWM-开启期间. 噪声消除定时器清除在 PWM-关闭期间. $<NCT>=4$)

19.5.2 解码器

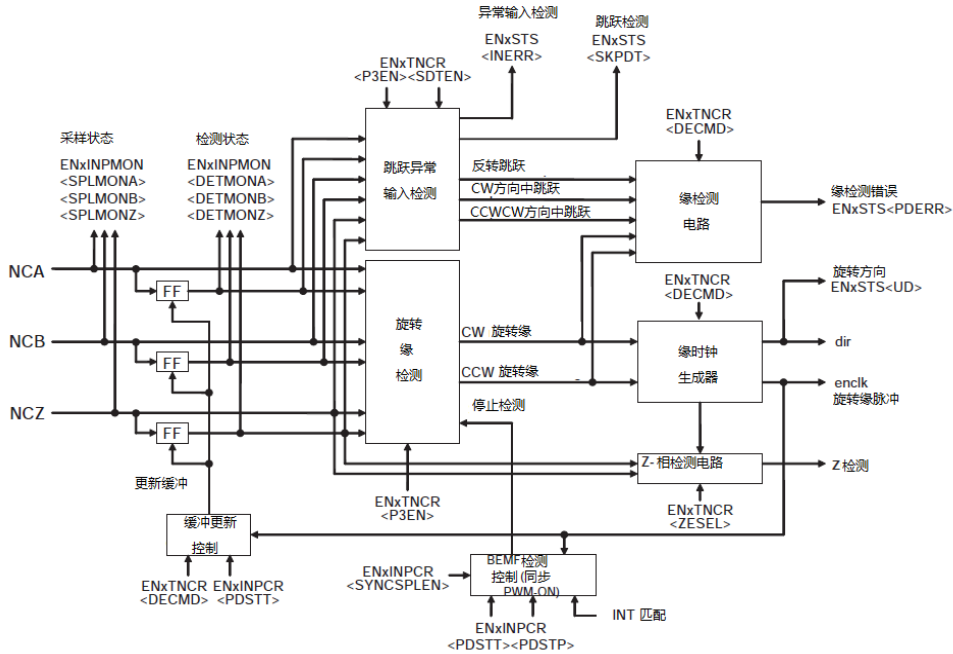


图 19-8 解码器配置

噪声消除后解码器检测转缘和转动方向从 2-相或 3-相输入.增量编码器被连接时也检测 Z-相; 单-相输入时检测 Z-信号缘.

19.5.2.1 转缘检测和方向信号生成

1. 2-相解码 (ENxTNCR<P3EN> = "0")

支持编码器模式和传感器模式 (通过设置 <P3EN>="0").

2-相解码中, 四输入类型变化 (转缘)被检测.

关于 CW 方向中输入, (1)→(2), (2)→(3), (3)→(4),或(4)→(1) 转缘被检测时, ENxSTS<UD>状态为 "1".

关于 CCW 方向中输入, (4)→(3), (3)→(2), (2)→(1), 或 (1)→(4)转缘被检测时, ENxSTS<UD>状态为 "0".

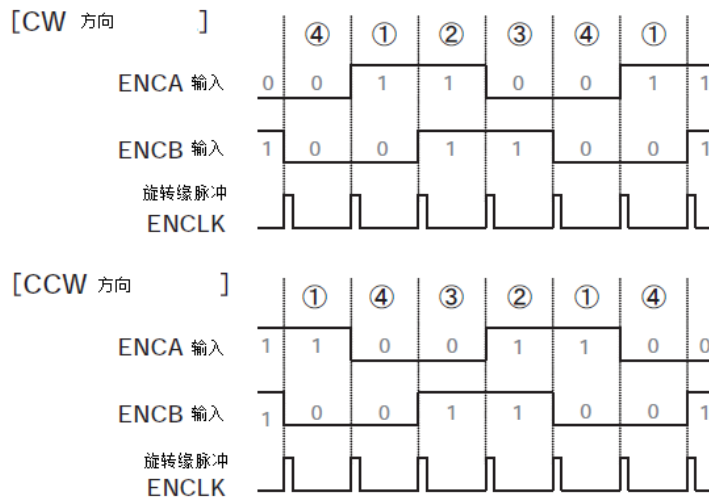


图 19-9 2-相 解码波形

2. 3-相解码 (ENxTNCR<P3EN> ="1") 可用

仅传感器模式中.

三-相解码检测 6 输入类型 (转缘) 变化.

关于 CW 方向中输入, (1)→(2), (2)→(3), (3)→(4), (4)→(5), (5)→(6), 或(6)→(1) 转缘时, ENxSTS<UD>状态设置为 "1".

关于 CCW 方向中输入, (6)→(5), (5)→(4), (4)→(3), (3)→(2), (2)→(1), 或(1)→(6) 转缘时, ENxSTS<UD>状态设置为"0".

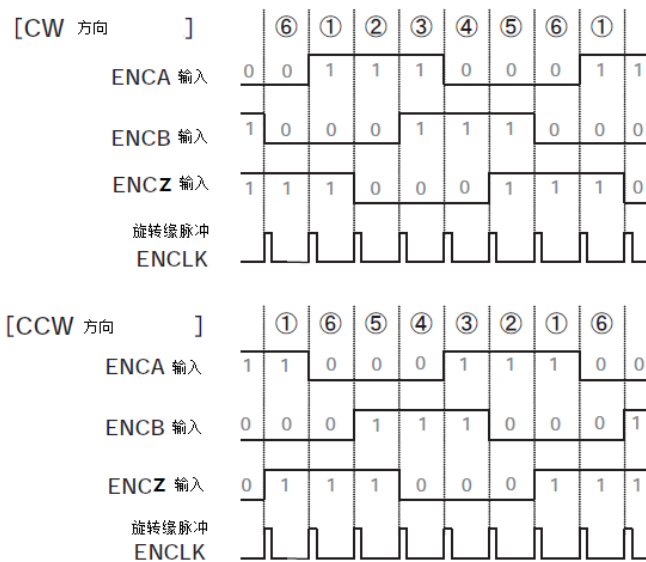


图 19-10 3-相解码波形

19.5.2.2 Z-输入 检测 电路

解码器检测 Z-输入缘.

1. 编码器 模式

解法器检测 CW 方向中上升缘和检测 CCW 方向中下降缘.

2. 定时器模式和相计数器 模式

解法器用 <ZESEL>从上升缘, 下降缘, 或双缘选择检测缘.

19.5.2.3 跳跃 检测和异常 输入 检测

1. 跳跃 检测

ENxTNCr<SDTEN> ="1"时跳跃检测使能.

如果跳跃被检测, <SKPDT>设置为 "1".

- 跳跃检测和 2-相解码 (ENxSTS<P3EN> ="0")组合

反转跳跃检测(1)→(3), (2)→(4), (3)→(1), (4)→(2)

- 跳跃检测和 3-相解码 (ENxSTS<P3EN> ="1")组合

CW 方向中跳跃检测: (1)→(3), (2)→(4), (3)→(5), (4)→(6), (5)→(1), (6)→(2)

CCW 方向中跳跃检测: (1)→(5), (2)→(6), (3)→(1), (4)→(2), (5)→(3), (6)→(4)

反转跳跃检测: (1)→(4), (4)→(1), (2)→(5), (5)→(2), (3)→(6), (6)→(3)

2. 异常输入 检测

传感器 模式 (事件 计数, 定时器计数, 或相计数)中, 3-相中的 三个输入都为 "0" 或都为 "1"的缘检测, 被检测, 编码器决定这些输入异常. 如果异常输入被检测, ENxSTS<INERR> 设置为"1".

19.5.2.4 缘 检测 错误 检测

不是用 ENxTNCr<DECMD>指定的方向的方向被检测时编码器决定错误. 缘检测错误检测为一个中断事件.

- 跳跃 检测禁止. (ENxTNCr<SDTEN> ="0")

CW 转动 (ENxTNCr<DECMD> ="01")期间: CCW 方向转缘被检测时错误发生.

CCW 转动 (ENxTNCr<DECMD> ="10")期间: CW 方向被检测时错误发生.

- 跳跃 检测使能. (ENxTNCr<SDETEN> ="1")

CW 转动 (ENxTNCr<DECMD> ="01")期间: CCW 方向中跳跃时错误发生, 反转跳跃, 或 CCW 方向转缘被检测.

CCW 转动 (ENxTNCr<DECMD> ="10")期间: CW 方向中跳跃时错误发生, 反转跳跃, 或 CW 方向转缘被检测.

19.5.2.5 缓冲 更新 控制

ENxTNCr<DECMD> ="00"时, 缓冲始终使能. 此时, 转缘检测和跳跃检测被输入信号变化决定.

如果 <DECMD> 设置到"00"以外,仅转缘检测时执行缓冲更新. 因此, 缘检测和跳跃检测结果储存在缓冲. 转缘被检测基于输入状态

(ENxINPMON<DETMONA><DETMONB><DETMONZ>) 在以前被检测缘和电流输入值 (ENxINPMON<SPLMONA><SPLMONB><SPLMONZ>)之间比较.

19.5.2.6 BEMF 检测 控制

传感器模式 (定时器计数和相计数)中, ENxINPCR<SYNCSPLEN> = "1" (PWM 同步 采样) 时 BEMF 检测控制使能. 可以停止 (挂起) 或开始 (重启) 转缘检测.

通过由电机控制电路 (PMD) 生成的矩形波形驱动的无刷 DC 电机诱导电压执行位置检测 (位置无传感器) 时此控制被用 .

1. 开始转缘 检测
 - 命令工作: 写入 "1" 到 ENxINPCR<PDSTT>
 - 事件 工作: INT 比较匹配从 计数器 电路
2. 停止旋转边缘 检测
 - 命令 工作: 写入 "1"到 ENxINPCR<PDSTP>
 - 事件 工作:转缘被检测.

19.5.3 计数器

计数器电路由时钟生成器, 计数器, 比较功能, 和捕捉功能组成. 可用功能取决于工作模式而变化.

19.5.3.1 编码器 模式和传感器 模式 (事件 计数)

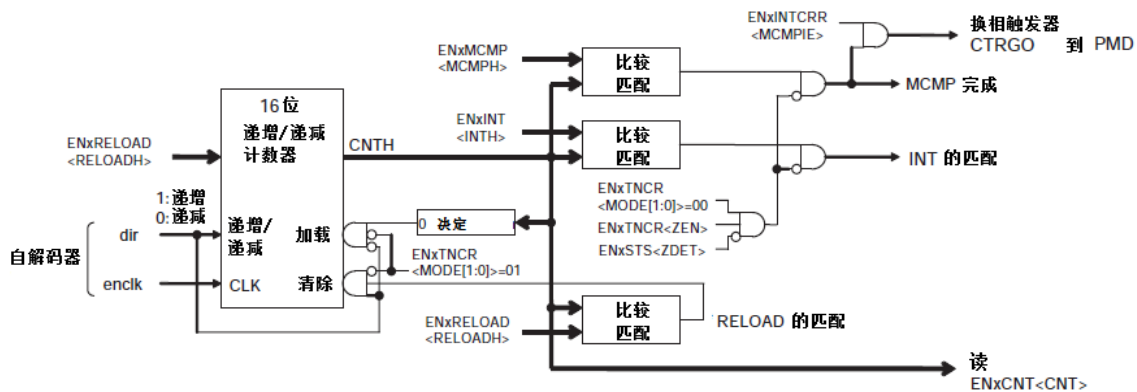


图 19-11 计数器 (编码器模式和传感器模式 (事件 计数)) 配置

计数器电路使用从 解码器的转缘脉冲 (enclk), 16-位递增/递减计数器工作在转动方向信号 (dir), 和三类型比较功能: <RELOADH>, <INTH>, 和 <MCMPH>.

编码器模式中, 通过 CW 转动期间 RELOAD 比较匹配清除计数器. CCW 转动检测被用时, 如果计数器被决定为 "0", <RELOADH>值被加载到计数器.

编码器模式中, Z-输入使能时(ENxTNCR<ZEN>="1"),编码器输入使能 (ENxTNCR<ENRUN>="1") 到 Z-相缘被检测时忽略 INT 比较和 MCMP 比较的匹配信号期间。

递增/递减计数器值可以通过计数器寄存器 (ENxCNT) 读. ENxINTCR<MCMPIE>="1"时, MCMP 比较匹配信号用作 PMD 电路换相触发器。

19.5.3.2 传感器模式 (定时器计数)或定时器模式

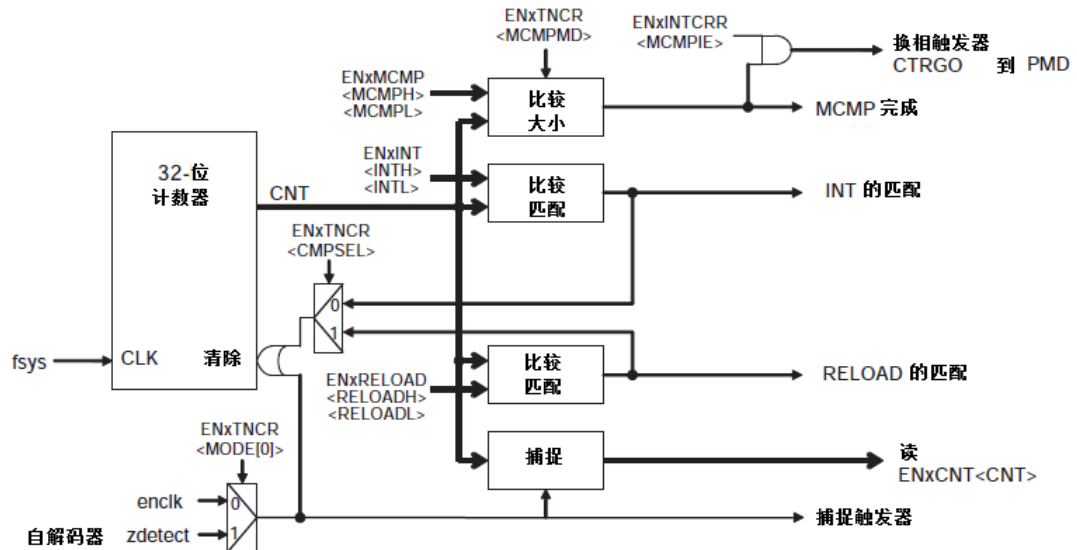


图 19-12 计数器 (传感器模式 (定时器计数) 或定时器模式) 配置

计数器由工作在系统时钟 (fsys) 32-位计数器, 三类型比较功能 (RELOAD, INT, 和 MCMP), 和捕捉功能组成。

MCMP 比较功能可以从匹配比较和比较尺寸选择选择类型. 比较尺寸选择 (ENxTNCR <MCMPMD>="1")时, 比较通过设置 ENxMCMP 寄存器开始. 如果满足条件, 输出 MCMP 完成信号且比较完成

定时器模式中计数器通过 INT 比较或RELOAD 比较匹配清除。

传感器模式 (定时器计数)中, 转缘被检测 (ENCLK) 时捕捉计数器且清除. 定时器模式中, Z-缘被检测 (Z 检测) 时捕捉计数器且清除. 捕捉值可以通过读计数器寄存器 (ENxCNT) 读到。

ENxINTCR<MCMPIE>="1"时, MCMP 比较匹配信号用作 PMD 电路换相触发器信号。

19.5.3.3 传感器模式(相计数)或相计数器模式

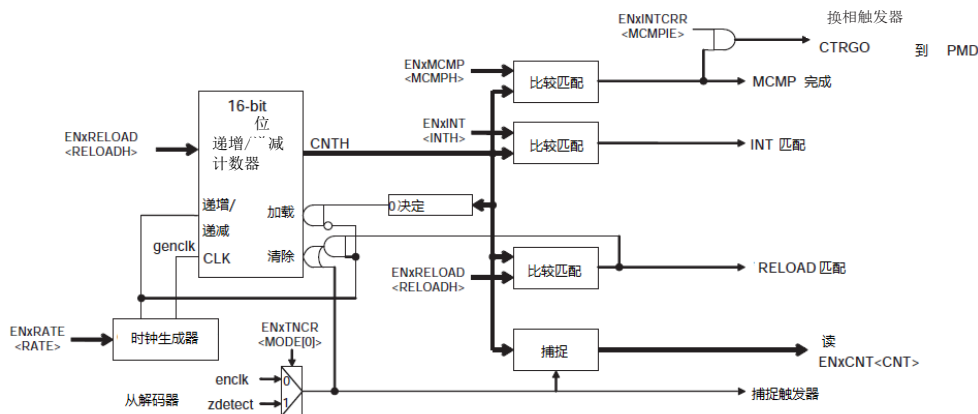


图 19-13 计数器(传感器模式(相计数),和相计数器模式)配置

计数器由时钟生成器被指定用 $ENxRATE<RATE>$, 16-位工作在时钟信号的递增/递减计数器和从时钟生成器的方向信号, 三-类型比较匹配比较器 ($<RELOADH>$, $<INTH>$, 和 $<MCMPH>$), 和捕捉功能组成.

计数器时钟在指定的率用 $ENxRATE$ 寄存器指定.

递增/递减计数器用 $ENxTNCR<UDMD>$ 指定.

计数器设置为递增计数时, 计数器通过匹配 $RELOAD$ 比较清除. 计数器设置为递减计数时, 如果计数器被决定为"0", $<RELOADH>$ 值被加载到计数器.

传感器模式(相计数)中, 在 $ENCLK$ 转缘被检测时捕捉计数器且清除. 相计数器模式中, Z -缘被检测 ($ZDETECT$) 时捕捉计数器且清除. 捕捉的值可以通过读计数器寄存器 ($ENxCNT$) 读到.

$ENxINTCRR<MCMPEN>="1"$ 时, $MCMP$ 比较匹配信号用作 PMD 电路换相触发器信号.

19.5.4 中断请求控制寄存器

计数器在 6 类型中断之间输出一/两 中断. 中断请求被使能来用中断控制寄存器 (ENxINTCR)输出根据中断事件. 中断事件可以用中断标志 (ENxINTF) 检查.

中断标志寄存器 (ENxINTF)通过中断事件的发生设置, 且标志通过读寄存器清除.

表 19-1 中断事件表

中断事件	描述	模式	中断请求输出使能 ENxINTCR	事件标志 ENxINTF	中断请求输出
分割的脉冲	转缘脉冲通过 1 到 128 用 ENxTNCR <ENDEV>分割时, 此中断请求被传输.	编码器 模式 传感器 模式(事件 计数)	<TPLSIE>	<TPLSF>	INTENCx0
捕捉	由于外部 触发器 (ENCZ 输入) 计数器值被捕捉时, 此中断请求被传输.	传感器 模式 (相计数) 相计数器 模式	<CAPIE>	<CAPF>	INTENCx0
	由于 ENCLK 计数器值被捕捉时, 此中断请求被传输.	传感器 模式 (定时器计数) 定时器模式			
检测 错误	缘检测错误(PDERR) 或 跳跃检测 (SKPDT) 发生时, 此中断请求被传输.	编码器 模式 传感器 模式	<ERRIE>	<ERRF>	INTENCx0
INT 匹配	ENxINT 寄存器值匹配计数器值时, 此中断请求被传输.	所有 模式	<CMPIE>	<INTCPF>	INTENCx1
RELOAD 匹配	ENxRELOAD 寄存器值匹配计数值时, 此中断请求被传输.	传感器 模式 (定时器计数相计数) 定时器 模式 相计数器 模式	<RLDIE>	<RLDCPF>	INTENCx1
MCMP 完成	ENxTNCR<MCMPMD>="0"时, 如果 ENxMCMP 寄存器匹配计数器值, 此中断请求被传输. <MCMPMD>="1"时, 如果计数器值超过 ENxMCMP 寄存器值或更多, 此中断请求被传输.	传感器 模式(定时器计数) 定时器 模式	<MCMPIE>	<MCMPF>	INTENCx1
	ENxMCMP 寄存器值匹配计数器值时, 此中断请求被传输.	编码器 模式 传感器 模式 (事件 计数 和相计数) 相计数器 模式			

19.6 无刷 DC 电机 控制例

此图表示通过工作于自 PMD 的矩形波形无刷 DC 电机 BEMF 过零被检测。

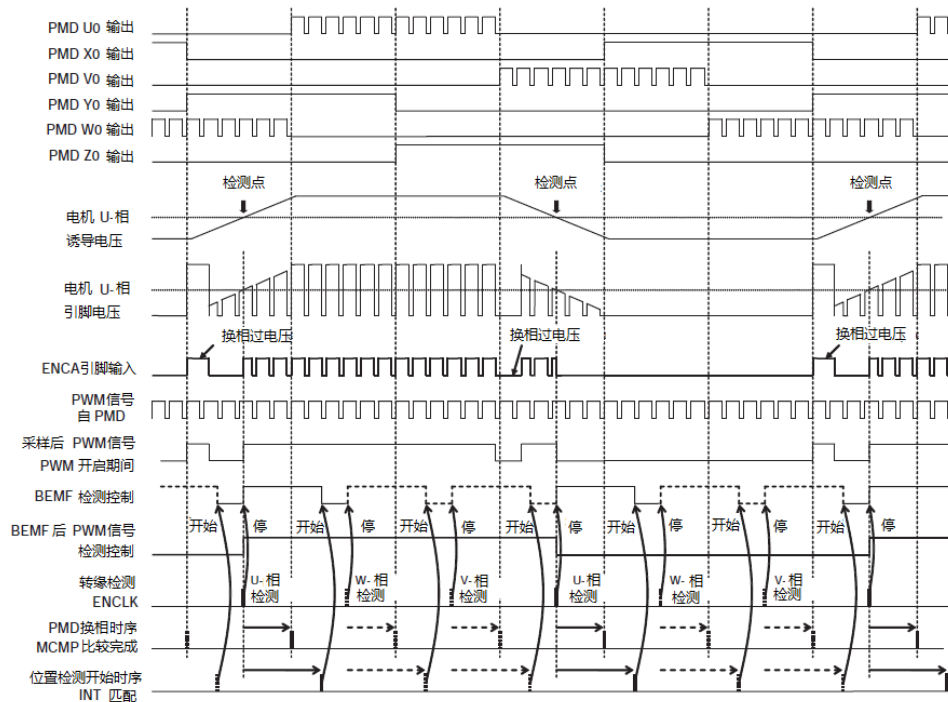


图 19-14 无传感器控制 120-度 激励时序波形

传感器模式(定时器计数和相计数)中, (EN_xINPCR <SYNCSPLEN> = "1")时 PWM 同步采样使能, 支持无刷 DC 电机以矩形波形工作的 PMD 中过零检测诱导电压 (BEMF). 此时, 输入电路使用 PWM 同步采样从输入信号移到 PWM 组成. 而且, 解码电路使用 BEMF 检测控制可以避免换相过电压. 基于以上功能, BEMF 过零可以从输入信号过滤为转缘脉冲 (ENCLK).

BEMF 检测控制由于转缘脉冲停止(挂起) 缘检测, 且由于 INT 比较匹配开始(重启) 缘检测. 也可通过命令 (EN_xINPCR <PDSTP> = "1") 停止, 且可以通过命令 (EN_xINPCR <PDSTT> = "1") 开始.

BEMF 过零功能与 PMD 电路连锁工作. 因此, 换相时序输出应该使能 (EN_xINTCR <MCMPIE> = "1").

要使用 BEMF 过零检测, 如下设置计数器捕捉功能和三个比较功能:

- 捕捉功能: 捕捉在转缘脉冲 (ENCLK) (以过零检测的间隔测量)
- MCMP 比较: PMD 换相时序 (设置例: 捕捉值 × 0.5)
- INT 比较: 位置检测开始时序 (设置例: 捕捉值 × 0.75)
- RELOAD 比较: 位置检测结束 (设置例: 捕捉值 × 2.0)

20. 上电复位 电路 (POR)

上电复位电路 (POR) 上电时生成上电复位信号。

电源电压指 DVDD5. 内部电源电压指 VOUT12.

注: 由于电源电压波动, 上电复位电路不能很好工作. 设备设计时用户应该基于电气特征做出考虑.

20.1 架构

上电复位电路由参考电压生成电路和比较器组成。

此电路用比较器中参考电压生成电路中生成的参考电压比较通过梯形电阻分割的电压。

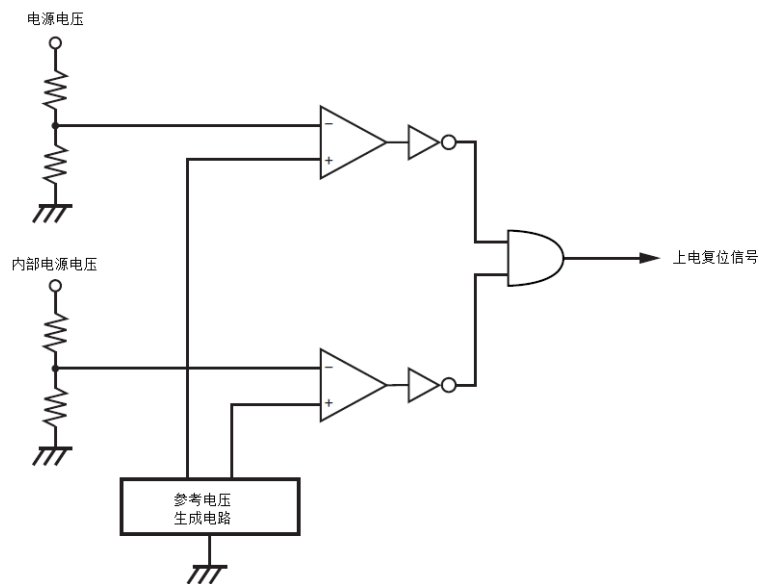


图 20-1 上电复位电路

20.2 功能

上电时, 电源电压低于复位释放电压时上电复位信号有效. 在电源电压超过 $3.0 \pm 0.2V$ 时序时上电复位信号无效.

电源 下降, 电源电压大于复位检测电压时上电复位信号无效. 在电源电压小于 $2.8 \pm 0.2V$ 时序时上电复位信号有效.

上电复位 信号有效期间, CPU 和 外设功能复位. 关于复位详细参考复位工作章节.

关于工作时序详细参考低电压检测电路 (VLTD) 章节.

注: 因为上电复位释放电压和上电复位检测电压相应改变, 检测电压从不反转.

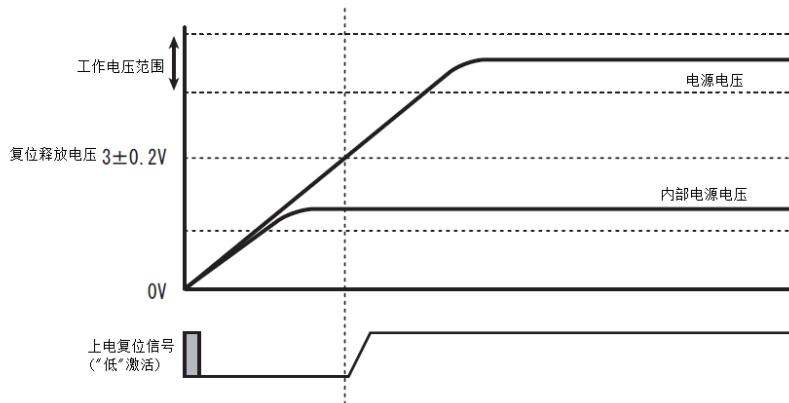


图 20-2 上电时上电复位工作时序

21. 低压检测电路 (VLTD)

低压检测电路 (VLTD) 通过检测降低的电源电压生成电压检测复位信号。

电源电压指 DVDD5。

注: 由于电源电压波动, 上电复位电路不能很好工作。设备设计时用户应该基于电气特征做出考虑。

21.1 架构

低压检测电路由参考电压生成电路, 检测电压电平选择电路, 比较器和控制寄存器组成。

电源电压通过电阻分割且输入检测电压选择电路。检测电压选择电路根据指定的检测电压 (VDLVL) 选择电压, 且比较器用参考电压比较它。

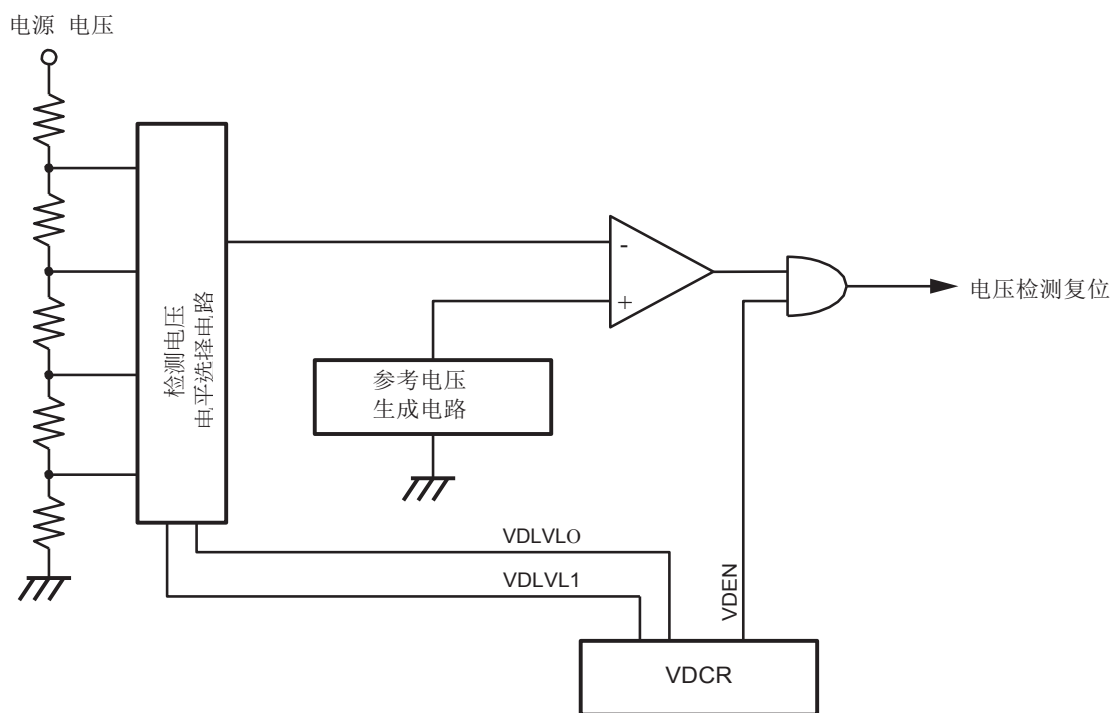


图 21-1 电压检测电路

21.2 寄存器

21.2.1 寄存器表

下表表示控制寄存器和其地址。

关于基地址详细, 参考"内存映射"章节的"外设功能地址表".

寄存器名	地址(基+)
电压检测控制寄存器	VDCR 0x0000

21.2.2 VDCR (电压检测控制 寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2-1	VDLVL[1:0]	R/W	检测电压选择 00: 保留 01: 4.1 ± 0.2V 10: 4.4 ± 0.2V 11: 4.6 ± 0.2V
0	VDEN	R/W	低电压检测工作 0: 禁止 1: 使能

注:通过上电复位或外部复位输入初始化 VDCR.

21.3 工作描述

21.3.1 控制

低电压检测电路通过电压检测控制寄存器控制。

21.3.2 功能

检测电压可以通过 $VDCR<VDLVL[1:0]>$ 。使能/禁止电压检测可以通过 $VDCR<VDEN>$ 编程。电压检测工作使能后, 电源电压变为低于检测电压 $<VDLVL[1:0]>$ 时, 电压检测复位信号有效。

复位参考"复位工作" 章节。

21.3.2.1 使能/禁止 电压 检测 工作

设置 $VDCR<VDEN>$ 为 "1" 使能电压检测工作。设置为 "0" 禁止工作。上电复位或通过外部复位输入的复位被释放后 $VDCR<VDEN>$ 立即设置为 "1"。

注: 电源电压低于检测电压 $VDCR<VDLVL[1:0]>$ 时, 设置 $VDCR<VDEN>$ 为 "1" 此时生成复位信号。

21.3.2.2 选择检测电压 电平

在 $VDCR<VDLVL[1:0]>$ 选择检测电压。

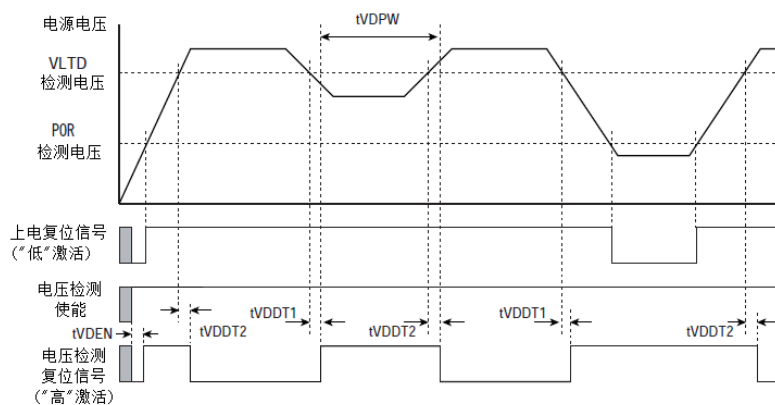


图 21-2 电压检测时序

注: 此时序指内部复位使能电压检测工作后电源电压大于检测电压 $VDCR<VDLVL[1:0]>$ 。

符号	参数	Min	Typ.	Max	单元
t_{VDEN}	使能 电压 检测后设置时间	-	40	-	μs
t_{VDDT1}	电压 检测 响应时间	-	40	-	
t_{VDDT2}	电压 检测 释放时间	-	40	-	
t_{VDPW}	电压 检测 最小脉宽	45	-	-	

22. 振荡频率检测器 (OFD)

振荡频率检测器电路 (OFD) 检测异常时钟频率. 要使用 OFD, 时钟异常状态诸如谐波, 子谐波或停止状态可以被检测.

OFD 监视使用参考频率的目标时钟频率和生成复位信号如果异常状态被检测. 此产品使用内部高-速振荡器时钟 (fiosc) 作参考且目标时钟为外部高-速振荡器时钟 (feosc). 它们由 CGOSCCR<OSCSSEL> 选择.

注: 不确保 OFD 可以在任何时间检测所有缺陷, 且它不是测量错误频率的电路.

22.1 方块图

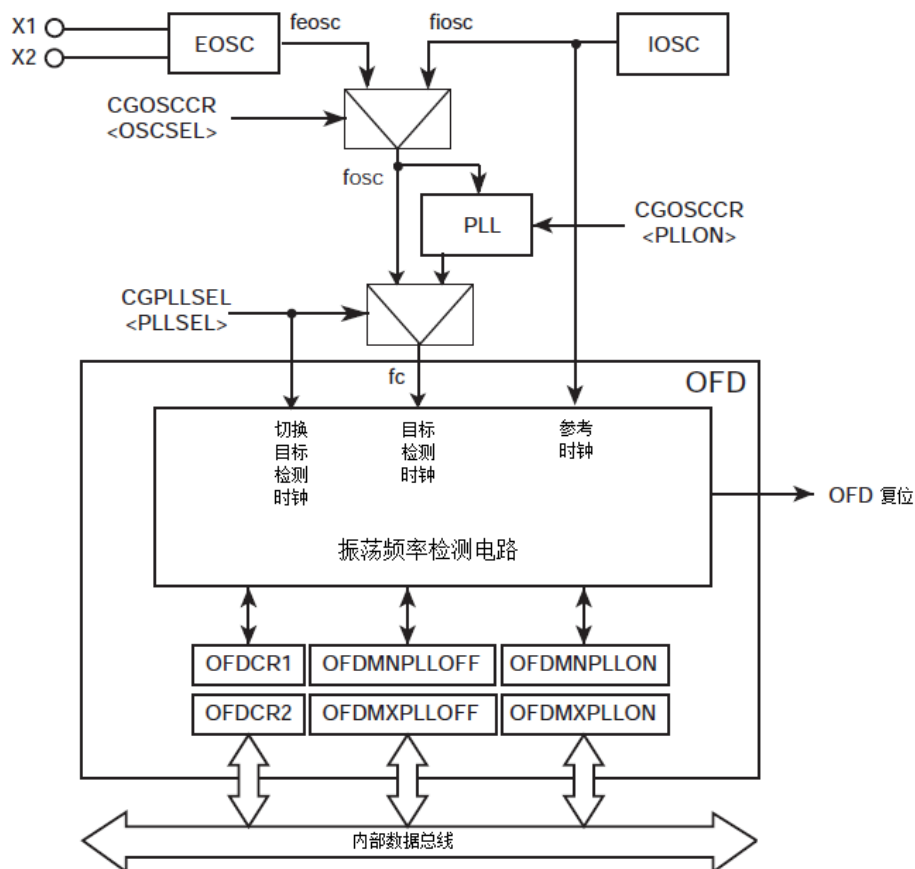


图 22-1 振荡频率检测器方块图

22.2 寄存器

22.2.1 寄存器表

下表表示控制寄存器和其地址。

关于基地址详细, 参考"内存映射"章节的"外设功能地址表".

寄存器名		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
低检测频率设置寄存器 (PLL OFF)	OFDMNPLLOFF	0x0008
低检测频率设置寄存器 (PLL ON)	OFDMNPLLON	0x000C
高检测频率设置寄存器 (PLL OFF)	OFDMXPLOFF	0x0010
高检测频率设置寄存器 (PLL ON)	OFDMXPLLON	0x0014

22.2.1.1 OFDCR1 (振荡频率检测控制寄存器)

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

位	位符	类型	描述
31-8	-	R	读作 0.
7-0	OFDWEN[7:0]	R/W	控制寄存器写入 0x06: 禁止 0xF9: 使能 设置 0xF9 使能 写入寄存器 除 OFDCR1. 写入 0x06 或 0xF9, 0x06 以外的值写入时, 如果写入寄存器禁止, 从各寄存器读使能.

注:通过 $\overline{\text{RESET}}$ 引脚, 上电复位或 LVD 复位初始化 OFDCR1.

22.2.1.2 OFDCR2 (振荡频率检测控制寄存器 2)

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

位	位符	类型	描述
31-8	-	R	读作 0.
7-0	OFDEN[7:0]	R/W	控制频率检测。 0x00: 禁止 0xE4: 使能 写入或 0xE4 以外的值无效且值不会被改变。

注:通过 $\overline{\text{RESET}}$ 引脚, 上电复位或 LVD 复位初始化 OFDCR2.

22.2.1.3 OFDMNPLLOFF (检测频率最小值的设置寄存器(PLL OFF 时))

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

位	位符	类型	描述
31-9	-	R	读作 0.
8-0	OFDMNPLL- OFF [8:0]	R/W	设置检测频率的最小值.

注: OFD 电路工作时忽略写入 OFDMNPLLOFF 寄存器.

注: 通过 $\overline{\text{RESET}}$ 引脚, 上电复位或 LVD 复位初始化 OFDMNPLLOFF.

22.2.1.4 OFDMNPLLON (检测频率最小值的设置寄存器(PLL ON 时))

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

位	位符	类型	描述
31-9	-	R	读作 0.
8-0	OFDMNPLLON [8:0]	R/W	设置检测频率的最小值.

注: OFD 电路工作时忽略写入 OFDMNPLLON 寄存器.

注: 通过 $\overline{\text{RESET}}$ 引脚, 上电复位或 LVD 复位初始化 OFDMNPLLON.

22.2.1.5 OFDMXPLLOFF (检测频率最大值的设置寄存器 (PLL OFF 时))

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

位	位符	类型	描述
31-9	-	R	读作 0.
8-0	OFDMXPLL- OFF[8:0]	R/W	设置检测频率的最大值.

注: OFD 电路工作时忽略写入 OFDMXPLLOFF 寄存器.

注: 通过 RESET 引脚, 上电复位或 LVD 复位初始化 OFDMXPLLOFF.

22.2.1.6 OFDMXPLLON (检测频率最大值的设置寄存器(PLL ON 时))

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

位	位符	类型	描述
31-9	-	R	读作0.
8-0	OFDMXPLLON [8:0]	R/W	设置检测频率的最大值.

注: OFD 电路工作时忽略写入 OFDMXPLLON 寄存器.

注: 通过 $\overline{\text{RESET}}$ 引脚, 上电复位或 LVD 复位初始化 OFDMXPLLON.

22.3 工作 描述

22.3.1 设置

振荡 频率 检测器 电路 (OFD) 寄存器通过 $\overline{\text{RESET}}$ 引脚, 上电复位或 VLTD 复位初始化. 所有寄存器除 OFDCR1 不可以通过复位写入. 可以通过写入 "0xF9" 到 OFDCR1 写入它们.

检测频率范围通过各目标时钟的 OFDMNPLLON/OFDMXPLLON 或 OFDMNPLLOFF/OFDMXPLLOFF 设置. 这些寄存器通过设置 CGPLLSEL<PLLSEL> 自动切换. 如果 OFD 复位用 PLLON (CGPLLSEL<PLLSEL> = "1") 生成, 检测频率设置寄存器 (OFDMNPLLON/OFDMXPLLON) 用 PLLOFF (CGPLLSEL<PLLSEL> = "0") 自动切换到 OFDMNPLLOFF/OFDMXPLLOFF.

用 OFDCR1="0xF9" 写入 "0xE4" 到 OFDCR2 使能振荡频率检测.

要防止防止错误写入, 应该写入 "0x06" 到 OFDCR1. 且 OFD 停止时寄存器应该修改

用 OFDCR2=0xE4 执行停止模式时, OFD 自动禁止. 释放停止模式和预热期间后, OFD 使能.

注: 通过 CGPLLSEL 寄存器控制 PLL (使能或禁止) 或通过 CGOSCCR 寄存器的 <OSCSSEL> 系统时钟被改变 (feosc 或 fiosc) 时, 处理前 OFD 必须禁止.

22.3.2 可用工作 模式

外部振荡频率仅在正常和 IDLE 模式下振荡频率检测可用. 移位到别的模式或使用片上振荡频率前, 禁止振荡频率检测.

表 22-1 振荡频率检测器有效性

工作 模式	振荡 频率 检测 (OFDCR2=0xE4)	振荡频率检测 RESET 后所有 I/Os 条件 (除电源, $\overline{\text{RESET}}$, MODE 引脚)
正常	可用	高 阻抗
IDLE	可用	高 阻抗
停止 (包括 预热期间)	振荡频率检测自动禁止.	
通过振荡频率 检测复位	可用	高 阻抗
看门狗 定时器复位 SYSRESETREQ 复位	可用	高 阻抗
通过外部 复位 RESET 上电复位 LVD 复位	禁止	-

22.3.3 工作

从工作开始到检测开始, 需要两周期时长的检测时钟.且检测周期为 128/参考时钟频率.

生成复位使能时,如果目标时钟频率超过通过 OFDMNPLLON/OFDMXPLLON 和 OFDMNPLLOFF/OFDMXPLLOFF 的频率极限设置 OFD 生成复位. 从异常频率检测到复位生成, 需要一周检测时钟时长. 通过 OFD 生成的复位使其自己和 OFD 不继续被检测工作.

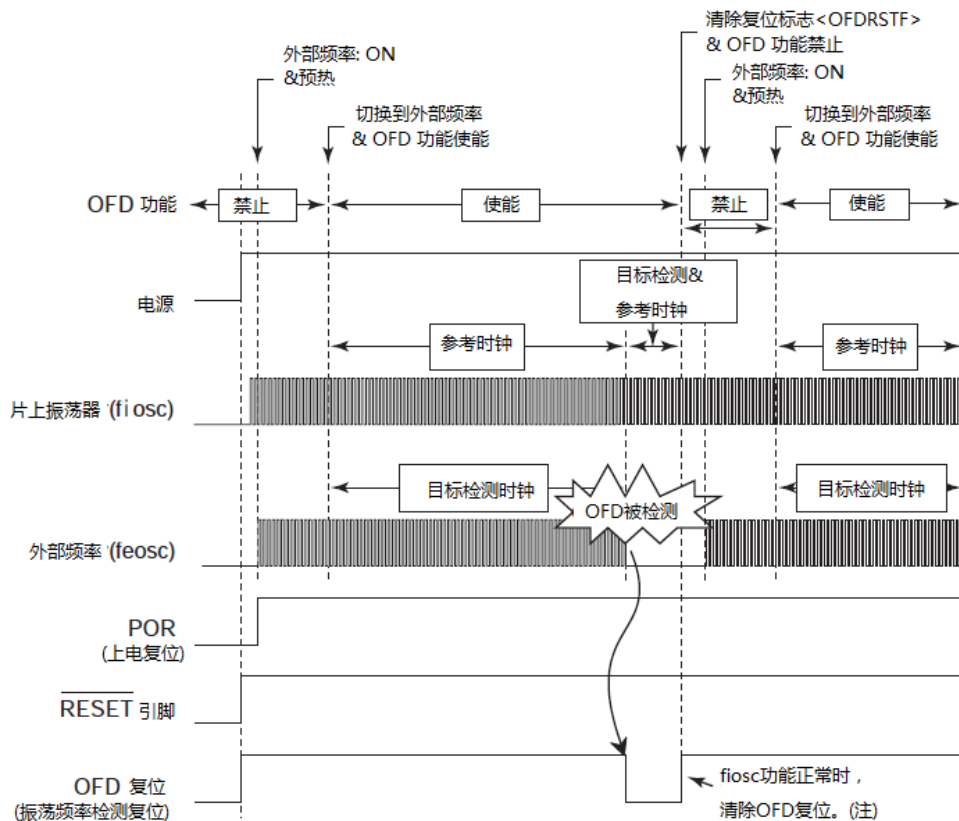
目标时钟(fc) 为 PLL ON 中 feosc 时如果 OFD 生成复位, fosc 初始化到 fiosc 且 PLL OFF 中 fc 自动改变为 fiosc,所以目标时钟变为相同于参考时钟. 频率范围在设置值之内且 OFD 复位释放.

内部高-速振荡器时钟 (fiosc) 被选择为系统时钟(fc) 时, OFD 不检测.

注 1: 有几个复位因素. 时钟生成器寄存器 CGRSTFLG 可以确认因素. CGRSTFLG 详细见章节 "异常."

注 2: 目标时钟不用 10MHz 设置到 OFDMNPLLOFF 和 OFDMXPLLOFF. (比如, 设置 8MHz 值到这些寄存器) 且目标

时钟频率为 fiosc 和 复位通过 OFD 生成时, 由于异常检测复位持续可能生成直到 OFD 正确检测.



注: 目标时钟不用 10MHz 设置到 OFDMNPLLOFF 和 OFDMXPLLOFF. (比如, 设置 8MHz 值到这些寄存器) 且目标时钟频率为 feosc 和 复位通过 OFD 生成时, 由于异常检测复位持续可能生成直到 OFD 正确检测.

图 22-2 振荡频率检测工作例

22.3.4 检测 频率

由于振荡精度，检测频率有检测频率范围和不可检测频率范围. 因此,未定义是否在检测频率范围和不可检测之间被检测.

检测频率的上和下限通过目标时钟和参考时钟的最大值误差计算.

OFDMNPLLON/OFDMNPLLOFF 和 OFDMXPLLON/ OFDMXPLLOFF 决定时通过四舍五入计算结果, 检测和未检测范围的上和下限所示如下. 四舍五入选择取决于被检测时钟的均匀性.

- 向上舍 OFDMXPLLON/OFDMXPLLOFF 和向下舍 OFDMNPLLON/ OFDMNPLLOFF 时
目标时钟高于未检测范围的上限且低于未检测范围的下限.
- 向下舍 OFDMXPLLON/OFDMXPLLOFF 和向下舍 FDMNPLLON/ OFDMNPLLOFF
目标时钟低于未检测范围的上限且高于未检测范围的下限.

目标 时钟误差±5% (未检测范围) 和参考时钟误差± 5%时如何计算 OFDMXPLLOFF/OFDMNPLLOFF 的设置值所示如下. 此例中, OFDMXPLLOFF 向上舍和 OFDMNPLLOFF 向下舍.(从 "a"到"h" 对应 "图 22-3 检测频率范围 (10MHz 时)例")

目标时钟	10MHz ± 5%	Max 10.5MHz	----- c
		Min 9.5MHz	----- b
参考时钟	9.7MHz ± 5%	Max 10.185MHz	----- f
		Min 9.215MHz	----- e

$$OFDMXPLLOFF = c \div e \times 32 = 36.46... = 37 \text{ (向上舍到最近小数)} = 0x25$$

$$OFDMNPLLOFF = b \div f \times 32 = 29.85... = 29 \text{ (向下舍到最近小数)} = 0x1D$$

此时，检测范围计算所示如下.

$$a = e \times OFDMNPLLOFF \div 32 = 8.35$$

$$d = f \times OFDMXPLLOFF \div 32 = 11.78$$

和未检测范围计算所示如下.

$$g = e \times OFDMXPLLOFF \div 32 = 10.65$$

$$h = f \times OFDMNPLLOFF \div 32 = 9.23$$

设置 "0x25" 到 寄存器 OFDMXPLLOFF 和 "0x1D" 到寄存器 OFDMNPLLOFF, 高于 11.78MHz 或低于 8.35MHz 的目标时钟被检测时, 振荡频率检测器输出复位信号. 且高于 9.23MHz 和低于 10.65MHz 的目标时钟被检测时, 振荡频率检测器不输出复位信号.

图 22-3 表示检测 或不可检测和可检测频率范围.

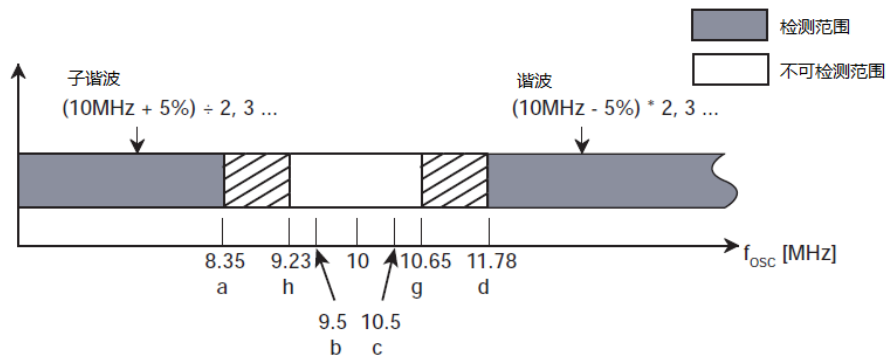


图 22-3 检测频率范围例(10MHz 时)

22.3.5 工作处理例

工作处理例所示如下。

复位后, 确认各种复位因素通过 CGRSTFLG。如果复位因素不通过振荡频率检测, 使能外部振荡, 设置寄存器使用 OFD 和使能工作。

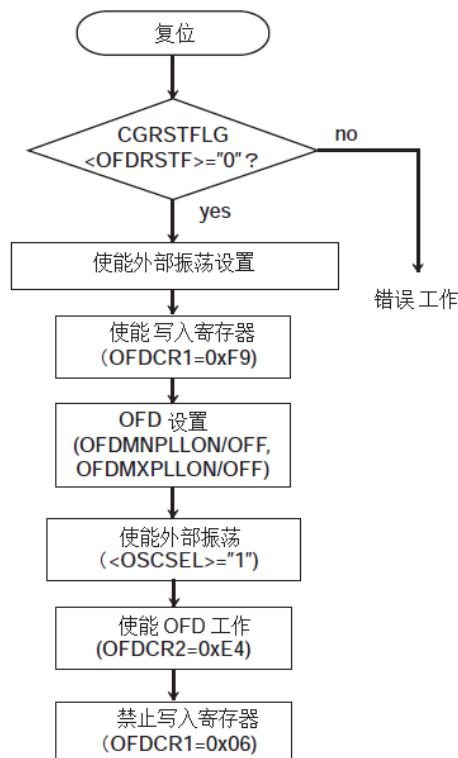


图 22-4 工作处理例

23.2 寄存器

23.2.1 寄存器 表

下表表示控制寄存器和其地址.

关于基地址详细, 参考"内存 映射" 章节"的外设功能地址表".

寄存器 名		地址 (基+)
看门狗 定时器 模式 寄存器	WDMOD	0x0000
看门狗 定时器 控制 寄存器	WDCR	0x0004

23.2.2 WDMOD (看门狗 定时器 模式 寄存器)

	31	30	29	28	27	26	25	24
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
位 符	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
位 符	WDTE	WDTP			-	I2WDT	RESCR	-
复位后	1	0	0	0	0	0	1	0

位	位 符	类型	功能
31-8	-	R	读作 "0".
7	WDTE	R/W	使能 / 禁止 控制 0: 禁止 1: 使能 要禁止看门狗定时器保护由误工作错误写入, 先 <WDTE>设置为"0", 如何禁止代码 (0xB1) 必须写入到 WDCR. 要改变看门狗定时器状态 从 "禁止" 到 "使能", 设置 <WDTE> 为 "1".
6-4	WDTP[2:0]	R/W	选择 WDT 检测时间 000: $2^{15}/f_{SYS}$ 100: $2^{23}/f_{SYS}$ 001: $2^{17}/f_{SYS}$ 101: $2^{25}/f_{SYS}$ 010: $2^{19}/f_{SYS}$ 110: 保留 011: $2^{21}/f_{SYS}$ 111: 保留
3	-	R	读作 "0"
2	I2WDT	R/W	IDLE 模式下工作 0: 停止 1: 工作
1	RESCR	R/W	检测 误工作后工作 0: INTWDT 中断请求生成. 注) 1: 复位
0	-	R/W	写入 "0".

注:INTWDT 中断为非-屏蔽中断因素.

23.2.3 WDCR (看门狗 定时器 控制 寄存器)

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

位	位 符	类型	功能
31-8	-	R	读作 "0".
7-0	WDCR	W	禁止 / 清除 代码 0xB1: 禁止 代码 0x4E: 清除 代码 其他: 保留

23.3 工作描述

23.3.1 基础 工作

看门狗定时器由使用系统时钟 (f_{sys}) 作输入工作的二进制计数器组成. 检测时间可以通过

WDMOD<WDTP[2:0]>在 2^{15} , 2^{17} , 2^{19} , 2^{21} , 2^{23} 和 2^{25} 之间选择.

指定的检测时间完毕, 看门狗定时器中断 (INTWDT) 生成, 且看门狗定时器引脚 ($\overline{\text{WDTOUT}}$) 输出 "低".

要检测由噪声或其他干扰导致的 CPU 误工作(失控), INTWDT 中断生成前应该通过软件指令清除二进制计数器看门狗定时器. 如果二进制计数器不清除, 非-可屏蔽中断通过 INTWDT 生成. 这样 CPU 检测误工作(失控), 执行误工作计数器测量程序返回正常工作.

另外, 通过连接看门狗定时器引脚到外设设备复位引脚可以解决 CPU 误工作(失控) 问题.

注:TMPM475FDFG/FZFG/FYFG 没有看门狗定时器引脚 ($\overline{\text{WDTOUT}}$).

23.3.2 工作 模式和状态

复位释放后看门狗定时器立即开始工作. 如果不使用看门狗定时器,应该禁止.

看门狗定时器不可以被用在停止高-速频率时钟. 过渡到以下工作模式前, 看门狗定时器应该禁止.

IDLE 模式下, 其工作取决于 WDMOD<I2WDT> 设置.

- 停止 模式

且, 调试模式期间二进制计数器自动停止.

23.3.3 误操作 (过运行)被检测时的工作.

23.3.3.1 INTWDT 中断 生成

图 23-2 表示 INTWDT 中断生成情况 (WDMOD<RESCR>="0").

二进制计数器溢出发生时, INTWDT 中断生成. 它是非-可屏蔽中断 (NMI)的因素. 这样 CPU 检测非-可屏蔽中断且执行计数器测量程序.

INTWDT 中断生成时, 同时看门狗定时器 ($\overline{\text{WDTOUT}}$) 输出 "低".

$\overline{\text{WDTOUT}}$ 变为 "高" 通过写入清除代码 0x4E 到 WDCR 的看门狗定时器清除.

注:TMPM475FDFG/FZFG/FYFG 没有看门狗定时器引脚 ($\overline{\text{WDTOUT}}$).

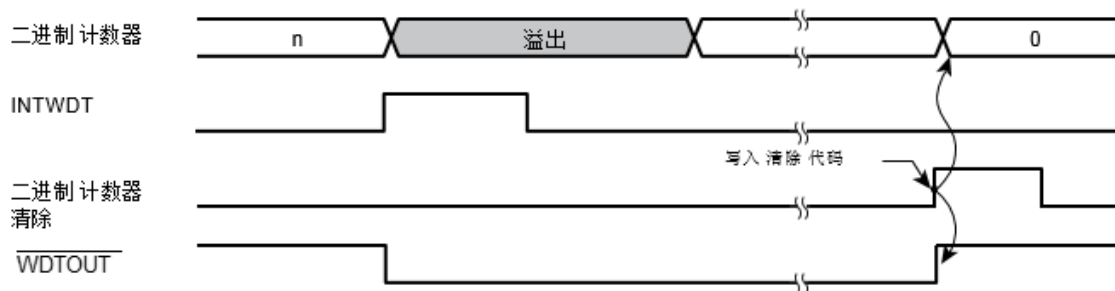


图 23-2 INTWDT 中断生成

23.3.3.2 内部复位生成

图 23-3 表示内部复位生成 ($WDMOD<RESCR>="1"$)。

MCU 通过二进制计数器溢出复位。此时，复位状态继续 32 状态。

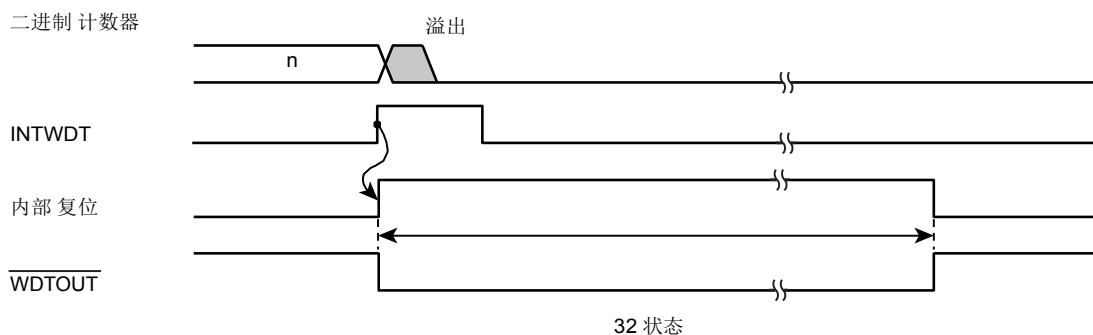


图 23-3 内部复位生成

23.4 看门狗 定时器的控制

23.4.1 禁止控制

设置 $WDMOD<WDTE>$ 为 "0" 后通过写入禁止代码 (0xB1) 到 WDCR，看门狗定时器可以禁止且二进制计数器可以清除。

23.4.2 使能控制

设置 $WDMOD<WDTE>$ 为 "1"。

23.4.3 看门狗 定时器清除控制

写入清除代码 (0x4E) WDCR 清除二进制计数器且重启计数。

23.4.4 看门狗 定时器检测时间

设置 $WDMOD<WDTP[2:0]>$ 取决于检测时间。

例如, $2^{21}/f_{SYS}$ 被用时, 设置 "011" 到 $WDMOD<WDTP[2:0]>$ 。

24. 内存 (FLASH)

此部分描述硬件配置和 Flash 内存工作.此部分中, "1-字" 表示 32 位.

24.1 特征

24.1.1 内存大小和配置

表 24-1 和表 24-2 表示内置内存大小和 TMPM475FDFG/FZFG/ FYFG 配置.

表 24-1 内存大小和配置

产品	内存大小(KB)	区域信息		方块信息		页信息		程序时间 (s) (注)	擦除时间 (ms) (注)				
		大小 (KB)	# 内存	大小 (KB)	# 内存	大小 (字节)	# 内存		1 页	1 方块	1 区域	芯片	
TMPM475FDFG	512	512	1	32	16	4096	128	8.0	115	920	115	115	
TMPM475FZFG	384	384	1		12		96						6.0
TMPM475FYFG	256	256	1		8		64						4.0

注:以上时间假设为复位后各寄存器初始值. 不包括数据传输时间.
每芯片 Flash 编程时间取决于用户'编程方法而变化.

表 24-2 方块 配置(1024KB)

区域 No.	方块- No.	地址 (单芯片模式)	地址 (单启动模式) (单芯片模式(镜像))	大小 (K 字节)	页数	可用区域		
						TMPM475DFG	TMPM475ZFG	TMPM475FYFG
0	0	0x0000_0000 到 0x0000_7FFF	0x5E00_0000 到 0x5E00_7FFF	32	8	*	*	*
	1	0x0000_8000 到 0x0000_FFFF	0x5E00_8000 到 0x5E00_FFFF	32	8	*	*	*
	2	0x0001_0000 到 0x0001_7FFF	0x5E01_0000 到 0x5E01_7FFF	32	8	*	*	*
	3	0x0001_8000 到 0x0001_FFFF	0x5E01_8000 到 0x5E01_FFFF	32	8	*	*	*
	4	0x0002_0000 到 0x0002_7FFF	0x5E02_0000 到 0x5E02_7FFF	32	8	*	*	*
	5	0x0002_8000 到 0x0002_FFFF	0x5E02_8000 到 0x5E02_FFFF	32	8	*	*	*
	6	0x0003_0000 到 0x0003_7FFF	0x5E03_0000 到 0x5E03_7FFF	32	8	*	*	*
	7	0x0003_8000 到 0x0003_FFFF	0x5E03_8000 到 0x5E03_FFFF	32	8	*	*	*
	8	0x0004_0000 到 0x0004_7FFF	0x5E04_0000 到 0x5E04_7FFF	32	8	*	*	
	9	0x0004_8000 到 0x0004_FFFF	0x5E04_8000 到 0x5E04_FFFF	32	8	*	*	
	10	0x0005_0000 到 0x0005_7FFF	0x5E05_0000 到 0x5E05_7FFF	32	8	*	*	
	11	0x0005_8000 到 0x0005_FFFF	0x5E05_8000 到 0x5E05_FFFF	32	8	*	*	
	12	0x0006_0000 到 0x0006_7FFF	0x5E06_0000 到 0x5E06_7FFF	32	8	*		
	13	0x0006_8000 到 0x0006_FFFF	0x5E06_8000 到 0x5E06_FFFF	32	8	*		
	14	0x0007_0000 到 0x0007_7FFF	0x5E07_0000 到 0x5E07_7FFF	32	8	*		
	15	0x0007_8000 到 0x0007_FFFF	0x5E07_8000 到 0x5E07_FFFF	32	8	*		

Flash 内存配置单元描述为"方块" 和 "页".

- 页
 - 被用在擦除功能和保护功能.
 - 一页被固定为 4096 字节.
- 方块
 - 被用在擦除功能和保护功能.
 - 一方块被固定为 32K 字节.
- 区域
 - 被用在擦除功能.
 - 有 512K 字节 区域.

以 16 字节 (4 字节 x 4 次) 单元执行写入工作. 每 16 字节写入时间为 163 μ s (Typ.).

执行每方块擦除或执行整个 flash 内存. 擦除时间随命令改变. 如果自动方块命令被用, 擦除时间会为 920 ms. 每方块 (Typ.). 如果自动芯片擦除命令被用来擦除整个区域, 时间会为 345 ms. (Typ.). 如果其他命令被用, 时间会为 115 ms (Typ.).

另外, 保护功能通过从 页 0 到 7 的页设置. 其他方块通过方块设置. 保护功能擦除时, 整个 Flash 内存擦除一次. 关于保护功能详细, 参考"24.1.5 保护/安全 功能".

24.1.2 功能

此设备内置 Flash 内存通常符合 JEDEC 标准除一些特定功能. 因此, 如果用户目前使用 Flash 内存为外部内存, 容易执行功能到此设备. 另外, 为了提供容易写入或擦除工作, 此产品包含执行写入或芯片自动擦除的专用 电路.

JEDEC 符合 功能	被修改, 被加, 或被删除功能
<ul style="list-style-type: none"> • 自动 编程 • 自动 芯片 擦除 • 自动 方块 擦除 • 数据 轮询/切换 位 	<p><加> 自动 区域 擦除, 自动 页 擦除, 自动 内存 交换</p> <p><被修改> 方块 写入/擦除 保护 (仅支持软件保护)</p> <p><被删除> 擦除 恢复 - 挂起 功能</p>

24.1.3 工作 模式

flash 内存 写入/擦除工作期间不要断电或复位 MCU.

24.1.3.1 模式 描述

此设备提供单芯片模式和单启动模式. 单芯片模式包含正常模式和用户启动模式. 图 24-1 表示模式过渡.

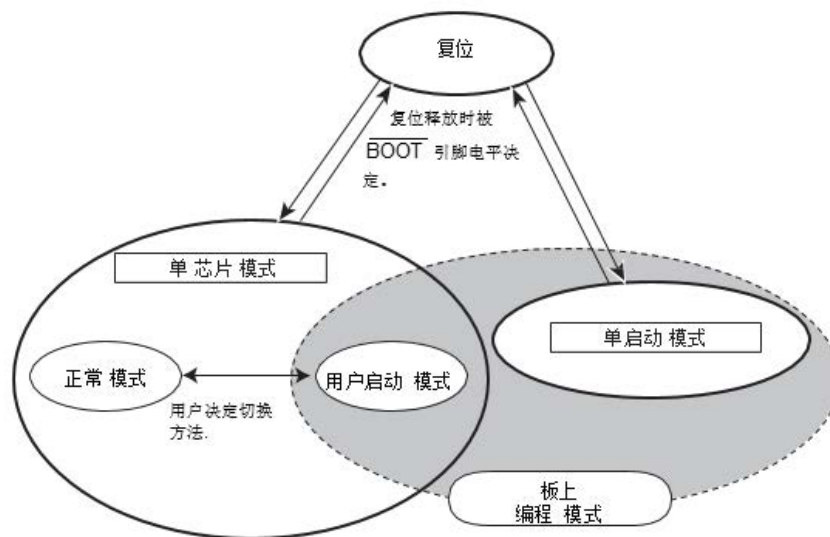


图 24-1 模式 过渡

(1) 单 芯片 模式

单芯片模式为复位后设备可以从 Flash 内存启动的模式. 模式 包含 3 子-模式如下.

- 正常 模式
 用户应用程序被执行的模式.
- 用户启动 模式

用户启动模式为 Flash 内存中位于 Flash 内存外部的可以重新-编程的模式。

关于 Flash 重新编程详细, 参考 "24.4 用户启动模式下重新-编程". 用户可以自由切换正常模式到用户启动模式. 例如, 如果端口 A 的 PA0 为 "1" 用户可以设置, 模式为正常模式. 如果端口 A 的 PA0 为 "0", 模式为用户启动模式.

用户必须准备应用程序中调度程序来决定切换.

切换各模式被为用户自由决定. 例如, 端口 A 的 PA0 设置为 "1" 时, 模式为正常模式; 端口 A 的 PA0 设置为 "0" 时, 模式为用户启动模式. 用户应该提供应用程序中部分调度程序来切换模式.

(2) 单启动模式

单启动模式为 Flash 内存中位于外部的 Flash 内存模式复位后可以从内置 BOOT ROM (屏蔽 ROM) 启动.

Flash 内存可以通过位于 Flash 内存外的 Flash 内存重新编程程序重新编程诸如单内部内存. 关于重新编程 Flash 内存详细, 参考 "24.3 单启动模式下如何重新编程 Flash".

BOOT ROM 包含可以根据用户的设置通过此设备串行端口重新写入 Flash 内存的算法. 用连接串行端口到外部主机, 用以上-提到的协议执行数据传输且重新编程 Flash 内存

(3) 板上重新编程模式

用户启动模式和单启动模式为根据用户设置 Flash 内存可以重新编程的模式. 这两模式叫做"板上编程模式".

24.1.3.2 模式决定

单芯片或单启动工作模式复位释放时可以通过 $\overline{\text{BOOT}}$ 引脚电平选择.

关于复位工作详细, 参考关于"复位工作"章节.

(1) 热复位时模式决定

表 24-3 热复位时工作模式设置

工作模式	引脚	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
单芯片模式	0 → 1	1
单启动模式	0 → 1	0

(2) 冷复位时模式决定

表 24-4 冷复位时工作 模式设置 (不使用 $\overline{\text{RESET}}$ 引脚)

工作 模式	POR 扩展 信号	引脚
		$\overline{\text{BOOT}}$
单 芯片 模式	0 → 1	1
单 启动 模式	0 → 1	0

- 模式决定 通过上电复位 电路 ($\overline{\text{RESET}}$ 引脚不被用).

维持 $\overline{\text{BOOT}}$ 引脚设置直到 POR 扩展信号从上电计数器从 "0" 被改变为 "1".

- 模式决定 通过 $\overline{\text{RESET}}$ 引脚

POR 扩展信号为 "1" 时, 使用 $\overline{\text{RESET}}$ 引脚复位工作可以被用. 因此, 执行表 24-3 中描述的同设置.

POR 扩展信号为 "0" 时, 执行表 24-4 中描述的同设置. 此时, 通过 POR 扩展信号的复位工作优先于通过 $\overline{\text{RESET}}$ 引脚的那些.

24.1.4 内存映射

图 24-2 和图 24-3 表示单芯片模式和单启动模式中内存映射的比较. 单启动模式中, 内置 Flash 内存映射到 0x5E00_0000 和随后地址, 且内置 BOOT ROM 映射到 0x0000_0000 到 0x0000_0FFF.

Flash 内存和 RAM 地址所示如下.

产品	Flash 大小	RAM 大小	Flash 地址	RAM 地址
TMPM475FDFG	512KB	34KB	0x0000_0000 ~ 0x0007_FFFF (单芯片模式) 0x5E00_0000 ~ 0x5E07_FFFF (单芯片模式(镜像)) 0x5E00_0000 ~ 0x5E07_FFFF (单启动模式)	0x2000_0000 ~ 0x2000_87FF
TMPM475FZFG	384KB	34KB	0x0000_0000 ~ 0x0005_FFFF (单芯片模式) 0x5E00_0000 ~ 0x5E05_FFFF (单芯片模式(镜像)) 0x5E00_0000 ~ 0x5E05_FFFF (单启动模式)	0x2000_0000 ~ 0x2000_87FF
TMPM475FYFG	256KB	34KB	0x0000_0000 ~ 0x0003_FFFF (单芯片模式) 0x5E00_0000 ~ 0x5E03_FFFF (单芯片模式(镜像)) 0x5E00_0000 ~ 0x5E03_FFFF (单启动模式)	0x2000_0000 ~ 0x2000_87FF

注: TMPM475FZFG 和 TMPM475FYFG 中, 有 ID 和密码的共通内存区域(0x5E07_FFF0 ~ 0x5E07_FFFF).

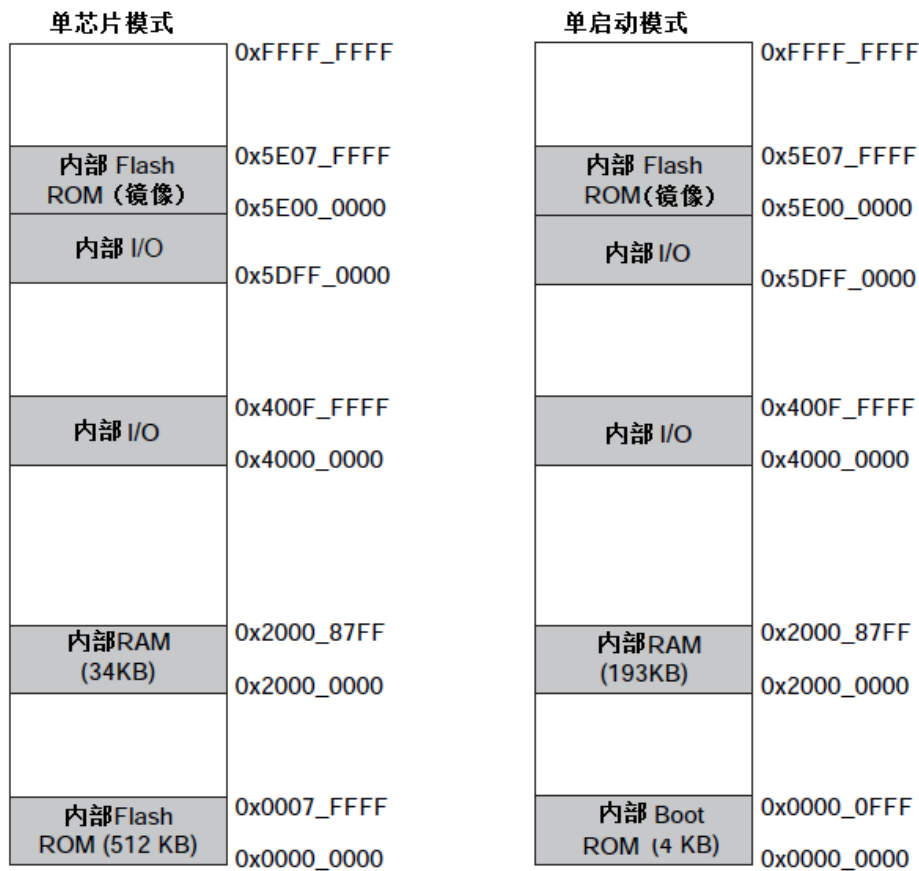


图 24-2 内存映射比较(TMPM475FDFG)

单芯片模式		单启动模式	
	0xFFFF_FFFF		0xFFFF_FFFF
内部 Flash ROM (镜像)	0x5E05_FFFF 0x5E00_0000	内部 Flash ROM(镜像)	0x5E05_FFFF 0x5E00_0000
内部 I/O	0x5DFF_0000	内部 I/O	0x5DFF_0000
内部 I/O	0x400F_FFFF 0x4000_0000	内部 I/O	0x400F_FFFF 0x4000_0000
内部RAM (34KB)	0x2000_87FF 0x2000_0000	内部RAM (193KB)	0x2000_87FF 0x2000_0000
内部Flash ROM (512 KB)	0x0005_FFFF 0x0000_0000	内部 Boot ROM (4 KB)	0x0000_0FFF 0x0000_0000

图 24-3 内存 映射比较(TMPM475FZFG)

单芯片模式		单启动模式	
	0xFFFF_FFFF		0xFFFF_FFFF
内部 Flash ROM (镜像)	0x5E03_FFFF 0x5E00_0000	内部 Flash ROM(镜像)	0x5E03_FFFF 0x5E00_0000
内部 I/O	0x5DFF_0000	内部 I/O	0x5DFF_0000
内部 I/O	0x400F_FFFF 0x4000_0000	内部 I/O	0x400F_FFFF 0x4000_0000
内部RAM (34KB)	0x2000_87FF 0x2000_0000	内部RAM (193KB)	0x2000_87FF 0x2000_0000
内部Flash ROM (512 KB)	0x0003_FFFF 0x0000_0000	内部 Boot ROM (4 KB)	0x0000_0FFF 0x0000_0000

图 24-4 内存 映射比较(TMPM475FYFG)

24.1.5 保护/安全 功能

此设备有 Flash 内存保护和安全管理功能。

1. 保护 功能

每方块写入/擦除工作可以禁止。

2. 安全 功能

从 flash 写入器读的工作可以禁止。

关于调试功能使用限制

24.1.5.1 保护 功能

方块 0 可以通过页使用保护功能。从方块 1 到最后方块, 这些方块可以通过方块使用保护功能。

要使用保护功能, 使用保护位程序命令对应方块的保护位设置为 "1"。如果使用保护位擦除命令保护位设置为"0", 方块保护可以取消。各 FCPSR 寄存器中可以监视保护位。

关于保护位的编程/擦除详细, 参考章节 "24.2.6 命令描述"。

24.1.5.2 安全 功能

表 24-5 表示安全功能使能时的工作。

表 24-5 安全功能使能时的工作。

项目	描述
读 flash 内存	CPU 可以读 flash 内存。
调试 端口	串行线或跟踪通信禁止。
Flash 内存命令执行	命令写入 flash 内存不接受。如果用户试图擦除保护位, 芯片擦除被执行且所有保护位被擦除。

在以下条件安全功能使能:

1. FCSECBIT<SECBIT>设置为 "1"。
2. 所有保护位 (各 FCPSR 寄存器中所有位)设置为 "1"。

通过冷复位 FCSECBIT<SECBIT> 设置为 "1"。FCSECBIT<SECBIT>的重新写入描述如下。

注:使用以下写入工作的 32-位传输指令, 项目 1 和 2。

1. 写入 被指定代码 (0xa74a9d23) 到 FCSECBIT
2. 项目 1 工作后 16 时钟内写入数据。

注: FCSECBIT<SECBIT> 为"0"时, 保护 位屏蔽为"0"且 写入/擦除工作使能。(安全和保护功能禁止。)然而, 原始保护位不改变。

表 24-6 <SECBIT> 和保护位

	<SECBIT>=1	<SECBIT>=0
保护 位 <所有>=1	安全 功能使能。	串行线 或跟踪 通信使能。 写入/擦除 工作使能。
保护 位 <任何>=1	串行线 或跟踪 通信使能。 每 方块写入/擦除 工作 禁止。	串行线 或跟踪 通信使能。 写入/擦除 工作使能。
保护 位 <所有>=0	串行线 或跟踪 通信使能。 写入/擦除 工作使能。	串行线 或跟踪 通信使能。 写入/擦除 工作使能。

24.1.6 内存 交换 功能

24.1.6.1 概要

如果 Flash 内存重新编程中断电, 编程不能被执行. 假设在程序擦除后变为断电的状态. 此时, 不可以完成编程. 要避免这种情况, 使用此内存交换功能保存你的程序.

24.1.6.2 工作 描述

交换区域通过从地址 0 和下一个区域开始的区域决定. 交换大小通过 FCSWPSR<SIZE>决定. 要改变此大小, 使用自动内存交换命令设置 "1" 到 FCSWPSR<SIZE> 位.

为了执行内存交换, 使用自动内存交换命令设置 "1" 到 FCSWPSR[0]. 要释放交换条件, 使用自动内存交换命令设置 "1" 到 FCSWPSR[1]. 交换条件可以通过 FCSWPSR<SWP>检查.

关于自动内存交换命令的详细, "24.2.6 命令描述".

24.1.6.3 内存交换的使用

此部分描述内存交换处理的基础流程. 关于内存交换例, 参考"24.5 使用用户启动模式如何重新编程 Flash".

1. 如果安全功能使能释放安全功能.

关于安全释放的详细, 参考"24.1.5.2 安全 功能".

如果安全功能不释放, 根据命令执行擦除 Flash 内存.

2. 如果安全功能使能, 擦除保护 位.

关于擦除保护位的详细, 参考 "24.1.5.1 保护 功能".

如果保护位不擦除, 根据处理命令执行不被执行.

3. 确认如果邻近从 地址 0 开始的区域的下一个区域为空白. (然后, 从地址 0 开始的区域被叫作页 0; 下一个区域叫作页 1) 如果此区域不空白, 擦除数据.

页 0: 旧原始数据

页 1: 空白

4. 写入原始的从地址 0 到下一个区域开始的数据 (区域都变为相同数据.)

页 0: 旧原始的数据

页 1: 复制数据 (旧原始数据)

5. 执行内存 交换

页 0: 复制数据 (旧原始数据)

页 1: 旧原始数据

6. 擦除旧原始的数据为空白

页 0:复制数据(旧原始数据)

页 1:空白

7. 写入新数据到空白区域

页 0:复制数据 (旧原始数据)

页 1:新原始数据

8. 释放交换条件

页 0:新原始数据

页 1:复制数据(旧原始数据)

9. 执行自动保护位擦除命令

10. 如果需要选项

- 擦除复制数据 (旧原始数据).
- 重新编程 Flash 内存数据除交换区域
- 激活保护功能
- 激活安全功能

处理		3	4	5	6	7	8
片上 RAM		擦除调度程序	编程调度程序	交换调度程序	擦除调度程序	编程调度程序	交换调度程序
Flash 内存	页 0	旧原始的	旧原始的	旧原始的 复制	旧原始的 复制	旧原始的 复制	新原始的
	页 1	空白	旧原始的 复制	旧原始的	空白	新原始的	旧原始的 复制

擦除调度程序:

擦除 Flash 内存的程序.

编程调度程序:

编程 Flash 内存的程序.

交换调度程序:

交换 Flash 内存的程序.

24.1.7 寄存器

24.1.7.1 寄存器表

下表表示控制寄存器和地址.

关于基地址的详细, 参考"内存映射"章节的"外设功能地址表".

外设功能: FC

寄存器名		地址 (基+)
安全位寄存器	FCSECBIT	0x0010
保护状态寄存器 0	FCPSR0	0x0020
状态寄存器	FCSR	0x0100
交换状态寄存器	FCSWPSR	0x0104
区域选择寄存器	FCAREASEL	0x0140
控制寄存器	FCCR	0x0148
状态清除寄存器	FCSTSCLR	0x014C
WCLK 设置寄存器	FCWCLKCR	0x0150
程序用计数设置寄存器	FCPROGCR	0x0154
擦除用计数设置寄存器	FCERASECR	0x0158

24.1.7.2 FCSECBIT (安全位寄存器)

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

位	位符	类型	功能
31-1	-	R	读作 "0".
0	SECBIT	R/W	安全位 0: 安全功能设置禁止. 1: 安全功能设置使能.

注:此寄存器通过冷复位初始化.

24.1.7.3 FCPSR0 (保护状态寄存器 0)

	31	30	29	28	27	26	25	24
位符	BLK15	BLK14	BLK13	BLK12	BLK11	BLK10	BLK9	BLK8
复位后	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)
	23	22	21	20	19	18	17	16
位符	BLK7	BLK6	BLK5	BLK4	BLK3	BLK2	BLK1	-
复位后	(注 1)	(注 1)	(注 1)	(注 1)	(注)	(注 1)	(注 1)	0
	15	14	13	12	11	10	9	8
位符	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
复位后	(注 1)	(注 1)	(注 1)	(注 1)	(注)	(注 1)	(注 1)	(注 1)
	7	6	5	4	3	2	1	0
位符	-	-	-	-	-	-	-	RDY_BSY
复位后	0	0	0	0	0	0	0	1

位	位符	类型	功能
31-17	BLK15 ~ BLK1	R	方块 1 ~ 15 保护状态 0: 无保护 1: 被保护 保护位值对应各方块保护状态. 如果对应位指 "1", 对应方块在保护状态中. 保护状态中方块不可以重新编程.
16	-	R	读作 "0".
15-8	PG7 ~ PG0	R	页 0 ~ 7 保护状态 0: 无保护 1: 被保护 保护位值对应各方块保护状态. 如果对应位指 "1", 对应方块在保护状态中. 保护状态中方块不可以重新编程.
7-1	-	R	读作 "0".
0	RDY_BSY	R	自动程序或自动芯片擦除命令执行时的准备/忙标志. (注 2) 0: 自动工作中 1: 自动工作结束 此位为从 CPU 监视 flash 内存的功能位. flash 内存自动工作中, 此位输出 "0" 指 flash 内存忙. 一旦自动工作完成, 此位变为准备状态且输出 "1". 然后接受下一个命令.

注 1: 取决于保护状态

注 2: 命令实施前确保 Flash 内存为准备.

24.1.7.4 FCSR (状态 寄存器)

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

位	位 符	类型	功能
31-25	-	R	读作 "0".
24	WEABORT	R	一旦自动工作中止, "1"设置到此位. 关于详细, 参考 "24.2.4 自动工作中止".
23-0	-	R	读作 "0x000001".

24.1.7.5 FCSWPSR (交换 状态 寄存器)

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

位	位 符	类型	功能
31-11	-	R	读作 "0".
10-8	SIZE[2:0]	R	交换大小 000: 4K 字节 001: 8K 字节 010: 16K 字节 011: 32K 字节 以上外: 设置禁止.
7-2	FLG	R	用作软件管理标志位(参考 "24.5 使用用户启动模式如何重新编程" 例.)
1-0	SWP[1:0]	R	交换 状态 11: 释放交换 10: 设置禁止. 01: 交换中 00: 释放交换 (初始 状态)

注:此寄存器通过自动保护位擦除命令初始化.

24.1.7.6 FCAREASEL (区域选择寄存器 0)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2-0	AREA0	R/W	指定通过 Flash 内存工作命令瞄准的 Flash 内存中的"区域". 111: 选择区域 0. 以上外: 非-选择区域 0.

注 1: 如果区域选择位 (<AREA n>) 写入, 等待写入数据通过轮询变为读.

注 2: 区域选择位为"111" (0x7) 以外时如果 Flash 内存工作命令被执行, 消除命令执行.

注 3: 自动芯片擦除命令被执行时, 设置 "111" (0x7) 到所有区域选择位.

24.1.7.7 FCCR (控制 寄存器)

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

位	位 符	类型	功能
31-3	-	R	读作 "0".
2-0	WEABORT	R/W	自动工作命令中止 111: 中止 读作设置值. 关于详细, 参考 "24.2.4 自动工作中止".

24.1.7.8 FCSTSCLR (状态清除寄存器)

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

位	位符	类型	功能
31-3	-	R	读作 "0".
2-0	WEABORT	R/W	清除 FCSR<WEABORT> 为 "0". 111: 清除 读作设置值. 关于详细, 参考"24.2.4 自动工作中止".

24.1.7.9 FCWCLKCR (WCLK 设置 寄存器)

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

位	位 符	类型	功能
31-5	-	R	读作 "0".
4-0	DIV	R/W	自动工作中改变时钟(WCLK: fsys/(DIV+1)) 8 到 12MHz 的频率分割比例 00000: 分割-通过-1 00001: 分割-通过-2 : 11110: 分割-通过-31 11111: 分割-通过-32

注 1: 设置 <DIV>前, 确保完成齿轮改变处理.

注 2: Flash 工作用改变的齿轮执行时, 重新设置 WCLK 为 8 到 12MHz 根据工作频率 (fsys). 表 24-6 表示总工作频率 (fsys) 和 <DIV>的例.

表 24-6 自动工作中对应工作频率的总时钟 (fsys) 和 <DIV>

	fsys	10MHz	20MHz	25MHz	30MHz	40MHz	80MHz	100MHz	120MHz
<DIV>	分割 比例	自动工作中时钟(MHz)							
00000	分割-通过-1	10	-	-	-	-	-	-	-
00001	分割-通过-2	-	10	-	-	-	-	-	-
00010	分割-通过-3	-	-	8.3	10	-	-	-	-
00011	分割-通过-4	-	-	-	-	10	-	-	-
00111	分割-通过-8	-	-	-	-	-	10	-	-
01001	分割-通过-10	-	-	-	-	-	8	10	12
01011	分割-通过-12	-	-	-	-	-	-	8.3	10
01110	分割-通过-15	-	-	-	-	-	-	-	8

-: 设置 禁止.

24.1.7.10 FCPROGCR (程序计数 设置 寄存器)

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

位	位符	类型	功能
31-2	-	R	读作 "0".
1-0	CNT	R/W	设置计数 使通过 自动程序执行命令的编程时间(CNT/WCLK) 在 20 到 40 μs 范围内. 00:计数 250 01:计数 300 以上外:计数 350

注：WCLKCR<DIV>重新设置时,如果需要在 20 到 40μs 范围内通过自动程序执行命令重新设置编程时间 . 表 24-7 表示总 WCLK 和 <CNT>值的例.

表 24-7 WCLK 和<CNT>总擦除时间

	WCLK	8.33MHz	10MHz	12MHz
<CNT>	计数	编程时间(μs)		
00	250	30.0	25.0	20.8
01	300	36.0	30.0	25.0
以上外	350	-	35.0	29.2

-: 设置 禁止.

24.1.7.11 FCERASECR (擦除计数 设置 寄存器)

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

位	位符	类型	功能
31-4	-	R	读作 "0".
3-0	CNT	R/W	计数直到使用各自动擦除命令擦除时间 (CNT/WCLK) 为 100 到 130ms 0000: 计数 850000 0001: 计数 900000 0010: 计数 950000 0011: 计数 1000000 0100: 计数 1050000 0101: 计数 1100000 0110: 计数 1150000 0111: 计数 1200000 1000: 计数 1250000 1001: 计数 1300000 1010: 计数 1350000 以上外: 计数 1400000

注:WCLKCR<DIV>重新设置时,如果需要在 100 到 130 范围内通过自动擦除命令重新设置擦除时间. 表 24-8 表示总 WCLK 和 <CNT>值的例.

表 24-8 WCLK 和 <CNT>总擦除时间

	WCLK	8.33MHz	10MHz	12MHz
<CNT>	计数	擦除时间 (ms)		
0000	850000	102.0	-	-
0001	900000	108.0	-	-
0010	950000	114.0	-	-
0011	1000000	120.0	-	-
0100	1050000	126.1	105.0	-
0101	1100000	-	110.0	-
0110	1150000	-	115.0	-
0111	1200000	-	120.0	-
1000	1250000	-	125.0	104.2
1001	1300000	-	-	108.3
1010	1350000	-	-	112.5
以上外	1400000	-	-	116.7

-: 设置禁止.

24.2 闪存 详细

板上编程中, CPU 执行重新编程或擦除 Flash 内存的命令. 此重新编程/擦除控制程序应该由用户提前准备.

而且, 例如在 Flash 内存的区域 0, 或区域 (诸如区域 1) 上执行程序时, 指令不被执行, 可以被写入/擦除. (相反情况可以.)

24.2.1 功能

Flash 内存通常符合 JEDEC 标准除一些指定的功能. 然而, 工作命令的地址指定方法不同于标准命令.

如果写入/擦除工作被执行, 使用 32-位 (1-字) 储存指令命令输入命令到 flash 内存. 命令输入后, 写入或擦除工作自动在里面被执行.

表 24-9 Flash 内存 功能

主要 功能	描述
自动 程序	自动以 4 字 (16 字节) 单元数据写入.
自动 芯片 擦除	自动擦除 Flash 内存整个区域.
自动 区域 擦除	自动擦除选择的区域.
自动 方块 擦除	自动擦除选择的方块.
自动 页 擦除	自动擦除选择的页.
写入/擦除 保护	写入或擦除工作可以禁止.
自动 内存 交换	自动执行交换/交换 释放/交换 大小指定.

24.2.2 内存的工作模式

Flash 内存提供主要两类工作模式:

- 读内存数据的模式(读 模式)
- 自动擦除或重新写入内存数据的模式(自动工作模式)

上电后, 复位后或自动工作模式完成后, Flash 内存变为读模式. 指令储存在 Flash 内存或在读模式下数据读被执行.

读模式期间 如果输入命令, 工作模式变为自动工作. 如果命令处理完成, 工作模式返回读模式除 ID-读命令. 命令执行的详细, 参考"24.2.3 如何执行命令". 自动工作期间, 储存在 Flash 内存的数据读和指令执行 可以不执行.

24.2.3 如何执行 命令

通过用储存指令写入命令顺序到 Flash 内存执行命令执行. 根据输入地址和数据的组合 Flash 内存执行各自动工作命令. 关于命令执行的详细, 参考"24.2.6 命令描述".

储存到 Flash 内存指令的执行被叫作"总线写入周期". 各命令由一些总线写入周期组成. Flash 内存中, 以指定的命令执行写入周期的总线地址和数据时, 执行自动命令工作. 非-被指定命令中执行周期时, Flash 内存停止命令执行且返回读模式.

在命令顺序或输入不同命令顺序期间如果消除命令, 执行读命令或读/复位命令. 然后 Flash 内存停止命令执行且返回读模式. 读命令和读/复位命令被叫做"软件 复位".

写入命令顺序结束时, 自动工作开始且 FCPSR0<RDY_BSY> = "0" 被设置. 自动工作结束时, FCPSR0<RDY_BSY> = "1" 被设置且 Flash 内存返回读模式.

自动工作期间不接受新命令顺序. 如果想要停止命令工作, 参考 "24.2.4 自动工作中止".

关于命令执行注意

1. 自动工作禁止以下工作.
 - 所有中断请求
2. 要识别命令, 命令开始前命令顺序需要在读模式中. 确认如果 FCPSR0<RDY_BSY> = "1" 被设置优先于各命令第一个总线写入周期. 建议连续执行读命令.
3. 命令顺序必须被执行在片上 RAM.
4. 各命令被执行前设置 FCAREASEL 寄存器中区域选择位 (写入 "111" (0x7) 到 <AREAn>).
5. 使用连续 1-字 (32-位) 数据传输指令设置各总线写入周期.
6. 各命令顺序中如果执行进入到目标 Flash 内存, 总线错误发生.
7. 实施命令时, 错误地址或数据被写入, 确保实施软件复位然后返回读模式.
8. 各命令完成后的确认处理如下.
 - 1) 执行最终总线写入周期.
 - 2) 通过轮询等待 FCPSR0<RDY_BSY>变为读 "0" (忙).
 - 3) 通过轮询等待 FCPSR0<RDY_BSY>变为读 "1" (准备).
9. 从 Flash 内存读数据时, 清除 FCAREASEL 寄存器中区域选择位. (写入 "000" (0x0) 到 <AREAn>)

24.2.4 自动 工作中止

以下处理表示如何中止自动程序/擦除:

1. 读 FCPSR0<RDY_BSY>.
2. 如果处理结果 1 为 "1"(准备), 在处理 9 结束整个处理. 如果结果为"0",在处理 3 执行以下处理.
3. 写入 "0x7" 到 FCCR<WEABORT>.
4. 写入 "0x0"到 FCCR<WEABORT>.
5. 轮询 直到 FCPSR0<RDY_BSY>设置为 "1" (准备).
6. 读 FCSR<WEABORT>.

7. 实施读/复位命令.
8. 如果处理 6 结果为 "0" (注), 在处理 9 结束整个处理. 如果结果为"1", 执行以下处理清除 FCPSR0<RDY_BSY>.
 - (1) 写入 "0x7" 到 FCSTSCLR<WEABORT>.
 - (2) 写入 "0x0" 到 FCSTSCLR<WEABORT>.
 - (3) 轮询直到 FCSR<WEABORT>="0".
9. 结束

注:假设 FCPSR0<RDY_BSY> = "1" 优先于处理 3.

24.2.5 自动 工作完成通知

TMPM475FDG/FZFG/FYFG 提供检测 Flash 编程/擦除工作的完成通知的中断功能.

24.2.5.1 处理

以下如何设置自动工作完成通知的处理. 关于中断服务程序详细, 参考章节 "异常" 中的"中断".

1. 编程/擦除命令实施到 Flash 后,通过 FCPSR0<RDY_BSY>检查自动工作状态 (BUSY 状态). 自动工作确认后, 使能 INTFLRDY 中断.
2. Flash 自动工作完成后, INTFLRDY 中断发生.
3. INTFLRDY 中断服务程序中, 禁止 INTFLRDY 中断事件.

24.2.6 命令 描述

此部分说明各命令内容. 关于指定的命令顺序的详细, 参考 "24.2.7 命令顺序".

24.2.6.1 自动 页 程序

(1) 工作 描述

自动页程序写入以 4-字 (16 字节) 单元的数据. 程序写入数据到多页时, 需要一页一页的执行页命令. 不可以跨页写入.

写入 Flash 内存意味 "1"的数据单元变为"0"的数据单元. 不可以从"0"数据变为"1" 的数据单元. 要从 "0"改变 "1"的数据单元, 需要擦除工作.

自动页程序仅允许已被擦的各页一次. 其 "1" 或 "0" 的数据单元不可以写入数据两次或更多. 如果重新写入已经写入一次的页, 自动方块擦除或自动芯片擦除命令被执行后需要再设置自动页程序.

自动程序期间不接受别的命令顺序. 参考 "24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能被正常执行. 这样, 擦除工作后自动程序必须再被执行.

注 1: 没有擦出工作相同页执行两次或以上的页程序可能损坏设备.

注 2: 不可以写入被保护方块.

(2) 如何设置

第 1 到第 3 总线写入周期指自动页程序命令.

第 4 总线写入周期中, 页第一个地址和数据被写入. 第 5 总线周期中或后, 一页数据会被持续写入. 数据以一字 (32-位)单元写入.

如果部分 16 字节 需要写入, 写入 16 字节"0xFFFFFFFF" 为不需要写入的数据.

设备不执行内部自动识别工作.因此,确保读编程的数据确认其已经正确写入.

24.2.6.2 自动 芯片 擦除

(1) 工作 描述

自动芯片擦除被执行到所有地址的内存单元. 如果包含被保护方块, 输入命令顺序后这些方块不会擦除且返回读模式.

自动程序期间不接受别的命令顺序. 参考"24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能被正常执行. 这样, 擦除工作后自动程序必须再被执行.

(2) 如何设置

第 1 到第 6 总线写入周期指自动芯片擦除命令. 输入命令顺序后, 自动芯片擦除工作开始.

设备不执行内部自动识别工作.因此,确保读数据确认其已经正确擦除.

24.2.6.3 自动 区域 擦除

(1) 工作 描述

自动擦除命令执行指定的区域的擦除工作. 如果指定的区域包含被保护方块, 输入命令顺序后擦除工作不被执行且返回读模式.

自动擦除工作期间不接受别的命令顺序. 参考"24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能被正常执行. 这样, 擦除工作后自动程序必须再被执行.

(2) 如何设置

第 1 到第 5 总线写入周期指自动方块擦除命令. 第 6 总线 写入周期中, 要擦除的方块被指定. 输入命令顺序后, 自动方块擦除工作开始.

设备不执行内部自动识别工作.因此,确保读数据确认其已经正确擦除.

24.2.6.4 自动 方块 擦除

(1) 工作 描述

自动擦除命令执行指定的方块的擦除工作. 如果被指定方块包含被保护页或方块, 命令顺序输入后擦除工作不被执行且返回读模式.

自动擦除工作期间不接受别的命令顺序. 参考"24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能正常执行. 这样, 擦除工作后自动程序必须再被执行.

(2) 如何设置

第 1 到第 5 总线写入周期指自动方块擦除命令. 第 6 总线 写入周期中, 要擦除的方块被指定. 命令顺序输入后, 自动方块 擦除工作开始.

设备不执行内部自动识别工作.因此,确保读数据确认其已经正确擦除.

24.2.6.5 自动 页 擦除

(1) 工作 描述

自动页命令执行指定的页的擦除工作. 如果被指定页被保护, 擦命令顺序输入后除工作不被执行且返回读模式.

自动擦除工作期间不接受别的命令顺序. 参考"24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能正常执行. 这样, 擦除工作后自动程序必须再被执行.

(2) 如何设置

第 1 到第 5 总线写入周期指自动页擦除命令顺序. 第 6 总线 写入周期中, 要擦除的页在总线写入周期被指定. 命令顺序输入后, 自动页擦除工作开始.

设备不执行内部自动识别工作.因此,确保读数据确认其已经正确擦除.

24.2.6.6 自动保护位程序

(1) 工作描述

在一个时间自动保护位程序写入 "1" 到保护位. 要设置 "0" 到保护位, 使用自动保护位擦除命令.

关于保护功能的详细, 参考"24.1.5 保护/安全 功能".

自动擦除工作期间不接受别的命令顺序. 参考"24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能正常执行. 这样, 擦除工作后自动程序必须再被执行.

(2) 如何设置

第 1 到第 3 总线写入周期指自动保护位程序命令顺序. 第 4 总线写入周期中, 要写入的保护位被指定. 输入命令顺序后, 自动保护位程序开始. 检查各 FCPSR 寄存器中保护位, 是否被正确写入.

24.2.6.7 自动保护位擦除

(1) 工作描述

自动保护位擦除命令工作取决于执行中安全状态 .

- 非-安全 状态

清除所有被指定保护位为 "0".

- 安全 状态

所有 Flash 内存的地址擦除后擦除所有保护位.

关于安全状态的详细, 参考"24.1.5 保护/安全 功能".

自动擦除工作期间不接受别的命令顺序. 参考"24.2.4 自动工作中止" 停止此工作. 此时, 数据写入不能正常执行. 这样, 擦除工作后自动程序必须再被执行.

(2) 如何设置

第 1 到第 3 总线写入周期指自动保护位擦除顺序. 第 4 总线写入周期中, 指定 0x5E000000. 输入命令顺序后, 自动保护位擦出工作开始.

非-安全状态中, 所有保护位被擦除. 检查各 FCPSR 寄存器中保护位是否被正确擦除.

安全状态中, Flash 内存位的所有地址和所有保护被擦除. 确认是否数据和保护位擦除正常, 如果需要然后再执行自动保护位擦除, 自动芯片擦除或自动方块擦除.

24.2.6.8 ID-读

(1) 工作 描述

ID-读命令可以读包括 Flash 内存类型和三 类型代码诸如用户代码, 设备代码和宏代码的信息.

(2) 如何设置

第 1 到第 3 总线写入周期指 ID-读命令顺序. 第 4 总线写入周期中, 要读的代码被指定. 第 4 总线写入周期后,通过读随机 Flash 区域中工作可以获取代码.

ID-读 可以依次被执行. 第 4 总线写入周期和读 ID 值可以被重复执行.

ID-读命令不自动返回读模式. 要返回读模式, 执行读/复位命令.

24.2.6.9 读/复位 命令(软件 复位)

(1) 工作 描述

读/复位命令为返回 Flash 内存到读模式的命令.

ID-读命令被执行时, Flash 内存停止在当前状态而不会自动返回读模式. 要从此状态返回读模式, 使用 读/复位命令.它也被用来取消命令被输入到当中时的命令.

(2) 如何 设置

第 1 总线周期指读时顺序.其时顺序被执行后, Flash 内存返回读模式.

24.2.6.10 自动 内存 交换

(1) 工作 描述

自动内存交换为以 1-位单元写入 "1" 到 FCSWPSR[10:0]的各位的命令. 位不可以独立设置为 "0". 所有位应该使用自动保护位擦除命令清除为"0".

自动内存交换工作期间不接受别的命令顺序. 参考 "24.2.4 自动工作中止" 停止此工作. 此时, 此工作不能被正常执行. 这样, 自动内存交换必须再被执行.

(2) 如何设置

第 1 到第 4 总线写入周期指自动内存交换命令顺序. 命令顺序输入后, "1" 写入到 FCSWPSR 寄存器被指定位 .

24.2.7 命令 顺序

24.2.7.1 命令 顺序 表

表 24-10 表示各命令中总线写入周期的地址和数据。

除 ID-读命令第 5 总线周期所有命令周期为总线写入周期. 通过 32-位 (1-字) 数据传输指令执行总线写入周期. (下表表示仅低 8 位数据.)

关于地址的详细, 参考表 24-11. 使用下表 24-11 中描述在 Addr[15:9] 栏的"命令"值.

注 1) 使用以下地址位 [20:19] 值取决于目标区域.

Area0 : "00"

注 2) MCU 从停止 2 模式被移到正常模式时, 为了 Flash 编程需要检查 CGRSTFLG<OSFLF> 哪个为"1".

表 24-10 通过内部 CPU Flash 内存进入

命令顺序	第 1 总线周期	第 2 总线周期	第 3 总线周期	第 4 总线周期	第 5 总线周期	第 6 总线周期	第 7 总线周期
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	数据	数据	数据	数据	数据	数据	数据
读/复位	0xFFFFXX	-	-	-	-	-	-
	0xF0	-	-	-	-	-	-
ID-读	0XX54XX	0XXAAXX	0XX54XX	IA	0XX	-	-
	0xAA	0x55	0x90	0x00	ID	-	-
自动 程序	0XX54XX	0XXAAXX	0XX54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
自动 芯片 擦除	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	0XX54XX	-
	0xAA	0x55	0x80	0xAA	0x55	0x10	-
自动 区域 擦除	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	AA	-
	0xAA	0x55	0x80	0xAA	0x55	0x20	-
自动 方块 擦除	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	BA	-
	0xAA	0x55	0x80	0xAA	0x55	0x30	-
自动 页 擦除	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	PGA	-
	0xAA	0x55	0x80	0xAA	0x55	0x40	-
自动 保护 位 程序	0XX54XX	0XXAAXX	0XX54XX	PBA	-	-	-
	0xAA	0x55	0x9A	0x9A	-	-	-
自动 保护 位 擦除	0XX54XX	0XXAAXX	0XX54XX	0x0000XX	-	-	-
	0xAA	0x55	0x6A	0x6A	-	-	-
自动 内存 交换	0XX54XX	0XXAAXX	0XX54XX	MSA	-	-	-
	0xAA	0x55	0x9A	0x9A	-	-	-

补充 解释

- IA: ID 地址
- ID: ID 数据
- PA: 程序 地址
- PD: 程序 数据 (32-位 数据)

第 4 总线周期后, 以地址命令输入 16 位数据

- AA: 区域 地址 (见 表 24-2)
- BA: 方块 地址 (见 表 24-2)
- PGA: 页 地址
- PBA: 保护 位 地址 (见 表 24-12)
- MSA: 内存 交换 地址 (见 表 24-14)

24.2.7.2 地址 位配置 在总线周期

使用表 24-11 连接"表 24-10 通过内部 CPU Flash 内存进入".

根据第一个总线周期中和后面的正常总线写入周期地址配置设置地址 .

表 24-11 总线写入周期中地址位配置

[正常 命令]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:16]	Addr [15:9]	Addr [8:0]
正常 命令	正常命令的总线写入周期 地址设置					
	0x5E	"000"	注	推荐"0"	命令	推荐"0".

[读/复位和 ID-读]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:16]	Addr [15:14]	Addr [13:0]
读/复位	读/复位 第 1 总线 写入周期地址设置					
	0x5E	"000"	被固定到 "00".	推荐"0".		
ID-读	IA: ID 地址 (ID-读 第 4 总线 写入周期地址设置)					
	0x5E	"000"	被固定到 "00".	推荐"0".	ID 地址 (表 24-13)	推荐"0".

[自动 区域 擦除]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:0]
区域 擦除	AA: 区域 地址 (区域 擦除 第 6 总线 写入周期地址设置)			
	0x5E	"000"	注	推荐"0".

[自动 方块 擦除]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:13]	Addr [12:0]
方块 擦除	BA: 方块 地址 (方块 擦除 第 6 总线 写入周期地址设置)				
	0x5E	"000"	注	方块 地址 (表 24-2)	推荐"0".

[自动 页 擦除]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:12]	Addr [11:0]
页 擦除	PGA: 页 地址 (页 擦除 第 6 总线 写入周期 地址设置)				
	0x5E	"000"	注	页 地址	推荐"0".

[自动 程序]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:3]	Addr [2:0]
程序	PA: 程序 页 地址 (页 程序 第 4 总线 写入周期地址设置)				
	0x5E	"000"	注	页 地址	推荐"0".

[保护位程序和内存交换]

地址	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:12]	Addr [11:8]	Addr [7:4]	Addr [3:0]
保护位 擦除	保护位 擦除 第 4 总线 写入周期地址设置						
	0x5E	"000"	被固定到 "0".			推荐"0".	
保护位 程序	PBA: 保护位 地址 (保护位 程序 第 4 总线 写入周期地址设置)						
	0x5E	"000"	被固定到 "00".	推荐"0".	保护位选择 (表 24-12)		推荐"0".
内存 交换	MSA: 内存 交换 地址 (内存 交换第 4 总线 写入周期地址设置)						
	0x5E	"000"	被固定到 "00".	推荐"0".	内存交换位选择 (表 24-14)		推荐"0".

注) 使用以下地址位 [20:19] 值取决于目标区域。

Area0 : "00"

24.2.7.3 区域地址 (AA) 和 方块地址 (BA)

表 24-2 表示区域和方块地址. 自动方块擦除命令的第 6 总线写入周期中指定任何包括在区域/方块要擦除的地址. 单芯片模式中, 使用镜像区域地址指定.

24.2.7.4 如何指定 保护位 (PBA)

工作中保护位以 1-位单元被指定.

表 24-12 表示自动保护位程序保护位选择表.

表 24-12 保护位程序地址

方块	页	寄存器	保护位	PBA[11:4]							地址例 [31:0]	
				地址 [11:10]	地址 [9]	地址 [8]	地址 [7]	地址 [6]	地址 [5]	地址 [4]		
0	0	FCPSR0	<PG0>	0	0	0	0	0	0	0	0	0x5E00_0000
	1		<PG1>	0	0	0	0	0	0	0	1	0x5E00_0010
	2		<PG2>	0	0	0	0	0	0	1	0	0x5E00_0020
	3		<PG3>	0	0	0	0	0	0	1	1	0x5E00_0030
	4		<PG4>	0	0	0	0	0	1	0	0	0x5E00_0040
	5		<PG5>	0	0	0	0	0	1	0	1	0x5E00_0050
	6		<PG6>	0	0	0	0	0	1	1	0	0x5E00_0060
	7		<PG7>	0	0	0	0	0	1	1	1	0x5E00_0070
1	8 ~ 15		<BLK1>	0	0	0	1	0	0	0	0	0x5E00_0080
2	16 ~ 23		<BLK2>	0	0	0	1	0	0	0	1	0x5E00_0090
3	24 ~ 31		<BLK3>	0	0	0	1	0	1	0	0	0x5E00_00A0
4	32 ~ 39		<BLK4>	0	0	0	1	0	1	1	1	0x5E00_00B0
5	40 ~ 47		<BLK5>	0	0	0	1	1	0	0	0	0x5E00_00C0
6	48 ~ 55		<BLK6>	0	0	0	1	1	0	0	1	0x5E00_00D0
7	56 ~ 63		<BLK7>	0	0	0	1	1	1	1	0	0x5E00_00E0
8	64 ~ 71	<BLK8>	0	0	0	1	1	1	1	1	0x5E00_00F0	
9	72 ~ 79	<BLK9>	0	0	1	0	0	0	0	0	0x5E00_0100	
10	80 ~ 87	<BLK10>	0	0	1	0	0	0	0	1	0x5E00_0110	
11	88 ~ 95	<BLK11>	0	0	1	0	0	0	1	0	0x5E00_0120	
12	96 ~ 103	<BLK12>	0	0	1	0	0	0	1	1	0x5E00_0130	
13	104 ~ 111	<BLK13>	0	0	1	0	0	1	0	0	0x5E00_0140	
14	112 ~ 119	<BLK14>	0	0	1	0	0	1	0	1	0x5E00_0150	
15	120 ~ 127	<BLK15>	0	0	1	0	0	1	1	0	0x5E00_0160	

24.2.7.5 ID-读 命令 (IA, ID)

表 24-13 表示使用 ID-读命令如何指定代码和内容.

表 24-13 ID-读命令代码和内容

代码	ID[7:0]	IA[15:14]	地址例 [31:0]
制造 代码	0x0098	00	0x5E00_0000
设备 代码	0x005A	01	0x5E00_4000
-	保留	10	-
宏 代码	0x012F	11	0x5E00_C000

24.2.7.6 内存 交换 位 分配 (MSA)

表 24-14 表示被分配在自动内存交换命令的第 4 总线写入周期中 FCSWPSR[10:0]的设置值。

表 24-14 通过内存交换命令和地址例的 FCSWPSR[10:0]的设置值

FCSWPSR[10:0]	MSA[11:4]							地址例 [31:0]
	地址 [11]	地址 [10:9]	地址 [8]	地址 [7]	地址 [6]	地址 [5]	地址 [4]	
FCSWPSR[0]	1	被固定到 "0".	0	1	0	0	0	0x5E00_0880
FCSWPSR[1]	1	被固定到 "0".	0	1	0	0	1	0x5E00_0890
FCSWPSR[2]	1	被固定到 "0".	0	1	0	1	0	0x5E00_08A0
FCSWPSR[3]	1	被固定到 "0".	0	1	0	1	1	0x5E00_08B0
FCSWPSR[4]	1	被固定到 "0".	0	1	1	0	0	0x5E00_08C0
FCSWPSR[5]	1	被固定到 "0".	0	1	1	0	1	0x5E00_08D0
FCSWPSR[6]	1	被固定到 "0".	0	1	1	1	0	0x5E00_08E0
FCSWPSR[7]	1	被固定到 "0".	0	1	1	1	1	0x5E00_08F0
FCSWPSR[8]	1	被固定到 "0".	1	0	0	0	0	0x5E00_0900
FCSWPSR[9]	1	被固定到 "0".	1	0	0	0	1	0x5E00_0910
FCSWPSR[10]	1	被固定到 "0".	1	0	0	1	0	0x5E00_0920

24.2.7.7 命令顺序例

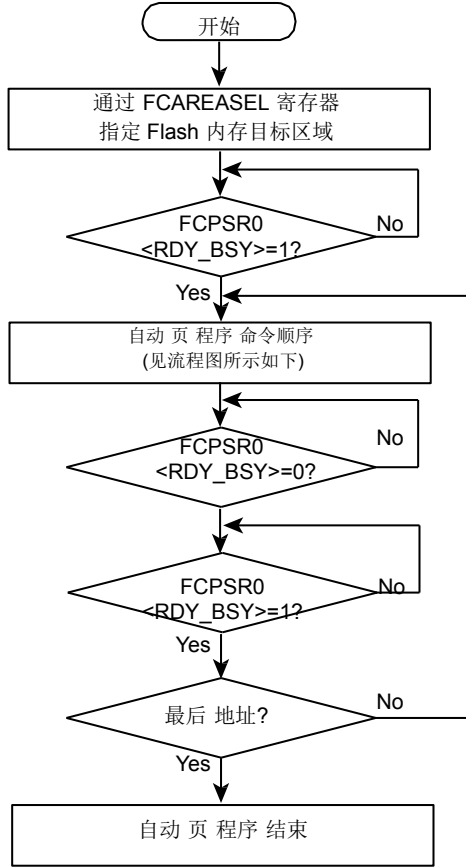
表 24-15 命令顺序例

命令	总线 周期							
		1	2	3	4	5	6	7
读/复位	地址	0x5E00_0000	-	-	-	-	-	-
	数据	0x0000_00F0	-	-	-	-	-	-
ID-读	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	IA	0x5E00_0000	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	-	-
自动芯片擦除	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	0x5E00_5400	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	-
自动保护位程序	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	PBA	-	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A	-	-	-
自动保护位擦出	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_0000	-	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A	-	-	-
自动内存交换	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	MSA	-	-	-
	数据	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_009A	-	-	-

命令	总线 写入							
		1	2	3	4	5	6	7
自动程序	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	PA	以后,16 位数据被连续写入.		
	数据	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
自动区域擦除	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	0x5E00_0000	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0020	-
自动方块擦除	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	BA	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	-
自动页擦除	地址	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	PGA	-
	数据	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0040	-

24.2.8 流程图

24.2.8.1 自动程序



自动页程序命令顺序 (地址/命令)

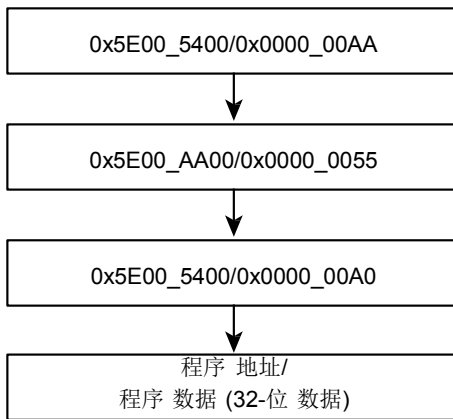


图 24-5 自动程序流程图

24.2.8.2 自动擦除

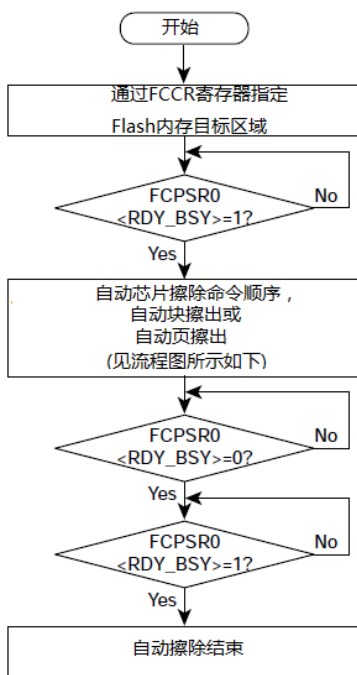


图 24-6 自动擦除流程图

24.3 如何在单启动模式下重编程闪存

单启动模式利用包含在内置编程 Flash 内存的 BOOT ROM 的程序。此模式下, BOOT ROM 映射到包含中断向量表的区域且 Flash 内存映射到 BOOT ROM 区域以外的别的地址区域。

单启动模式下, Flash 内存使用串行命令数据传输重新编程。通过连接此设备的串行通道 (SIO/UART) 到外部主机, 重新编程程序从外部主机被复制到内置 RAM。RAM 中重新编程程序被执行来重新编程 Flash 内存。关于用主机通信的详细, 遵照后面描述的协议。

即使在单启动模式下, 不要生成异常避免异常程序中止。

为了保证单芯片模式 (正常工作模式) 中 Flash 内存的内容, 建议保护相关 Flash 方块防止重新编程完成后随后单芯片工作期间无意擦出

24.3.1 模式 设置

为了执行板上编程, 此设备在单启动模式下启动。如下设置为单启动模式设置。

$$\overline{\text{BOOT}} = 0$$

$$\overline{\text{RESET}} = 0 \rightarrow 1$$

$\overline{\text{BOOT}}$ 引脚预先设置为以上条件时, 设置 $\overline{\text{RESET}}$ 引脚为 "0"。然后释放 $\overline{\text{RESET}}$ 引脚在单启动模式下启动。

24.3.2 接口 规格

此部分描述单启动模式下串行通信格式。串行工作支持 UART (异步通信) 模式。为了执行板上编程, 同时设置编程控制器通信格式。

通信 通道:	通道 0
串行 传输 模式:	UART (异步), 半-双工, LSB 先
数据 长:	8-位
奇偶 位:	注
停止 位:	1-位
波特 率:	随机波特率

内部启动程序工作在时钟/模式控制方块设置为初始条件。时钟初始设置的详细, 参考 "时钟/模式控制"。

正如"24.3.5.1 串行工作模式决定"说明, 波特率被 16- 位 定时器 (TMRB) 决定。因为决定波特率时通信需要的波特率的 1/16 执行; 因此通信波特率必须在可测量范围。定时器计数时钟工作在 $\Phi T1(fc/2)$ 。

表 24-16 表示 被用在启动程序中的引脚。这些引脚以外不被启动程序用。

表 24-16 被用 引脚

模式 设置 引脚	$\overline{\text{BOOT}}$ (PF0)
复位 引脚	$\overline{\text{RESET}}$
通信 引脚	SC0TXD (PE0)
	SC0RXD (PE1)

24.3.3 内置内存的限制

注意单启动模式有如表 24-17 中所示关于内置 RAM 和内置 flash 内存的限制。

表 24-17 单启动模式中有关内存限制

内存	限制
内部 RAM	启动程序使用内存为工作区域通过 0x2000_0000 ~ 0x2000_03FF. 储存程序 0x2000_0400 到 RAM 结束地址.
内部 Flash 内存	以下地址被分配储存软件 ID 信息和密码. 不应该使用以下程序储存地址. 0x5E07_FFF0 ~ 0x5E07_FFFF

注:如果密码为擦除的数据 (0xFF), 由于易猜的密码很难保护数据安全. 即使单启动模式不被用, 建议设置独特的值作密码

24.3.4 工作 命令

启动程序提供以下工作命令

表 24-18 工作命令数据

工作命令数据	工作 模式
0x10	RAM 传输
0x40	Flash 内存芯片擦除和保护位擦除

24.3.4.1 RAM 传输

RAM 传输为从控制器到内部 RAM 储存数据. 传输正常完成时, 用户程序开始. 用户程序可以使用启动程序内存地址 0x2000_0400 或后面 (除 0x2000_0000 ~ 0x2000_03FF). CPU 会在 RAM 储存开始地址开始.

此 RAM 传输功能使能用户指定板上编程控制. 为了通过用户程序执行板上编程使, 使用 24.2.7.中说明的 Flash 内存命令顺序.

24.3.4.2 闪存芯片擦除和保护位擦除

"Flash 内存芯片擦除和保护位擦除命令" 擦除 Flash 内存整个方块. 此命令擦除所有方块无论写入/擦除保护条件或安全状态.

24.3.5 无论命令的共同工作

此部分描述在启动程序执行下的共通工作。

24.3.5.1 串行工作模式 决定

控制器的在需要的波特率传输第 1 字节的 0x86。图 32-6 表示各情况下的波形。

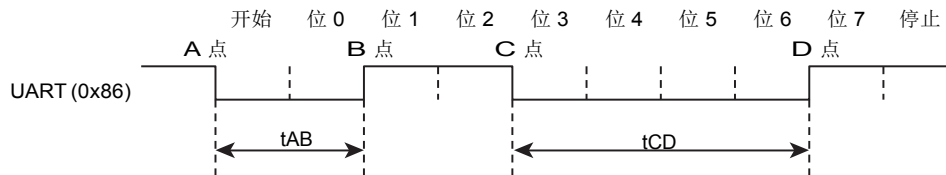


图 24-7 串行工作模式决定数据

图 24-8 表示内部启动程序流程图。使用 t_{AB} 、 t_{AC} 和 t_{AD} 时间的 16-位 定时器 (TMRB)，复位禁止后串行工作模式决定数据 (0x86) 第 1 字节。图 24-8 中，CPU 监视接收引脚电平，且接收引脚电平被改变时获取定时器值。因此 t_{AB} 、 t_{AC} 和 t_{AD} 定时器值有误差的余量。另外，注意如果传输走在高波特率，CPU 也许不能决定接收引脚电平。

图 24-9 中流程图表示串行工作模式由接收引脚时长的为长或短决定。如果长为 $t_{AB} \leq t_{CD}$ ，串行工作模式决定为 UART 模式。 t_{AD} 时间被用作自动波特率设置是否为使能。如果长为 $t_{AB} > t_{CD}$ ，串行工作模式不被决定为 UART 接口模式。注意 t_{AB} 、 t_{AC} 和 t_{AD} 定时器值有误差的余量。如果波特率为高但工作频率为低，各定时器值变小。此可能产生不预期的决定。(为了防止此问题，在编程调度程序中重新设置 UART。)

例如，控制器企图使用 UART 模式时串行工作模式可能不被决定为 UART 接口模式。为了避免这种情况，UART 模式被利用时，控制器应该允许一个从目标时间被期待接收后-回声 (0x86) 的结束周期。如果在允许的时间没能获取后-回声，控制器不能通信。

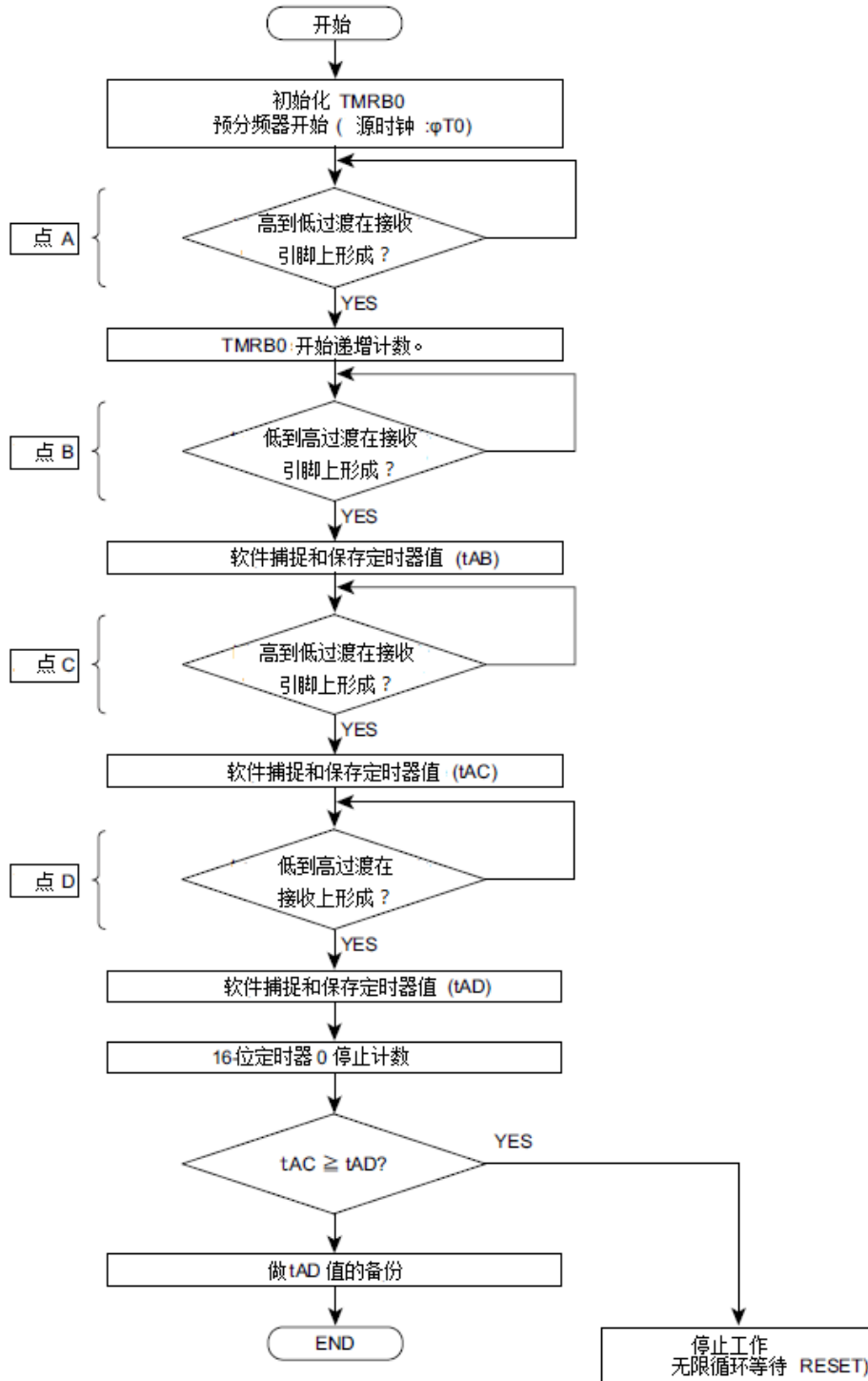


图 24-8 串行工作模式接收流程图

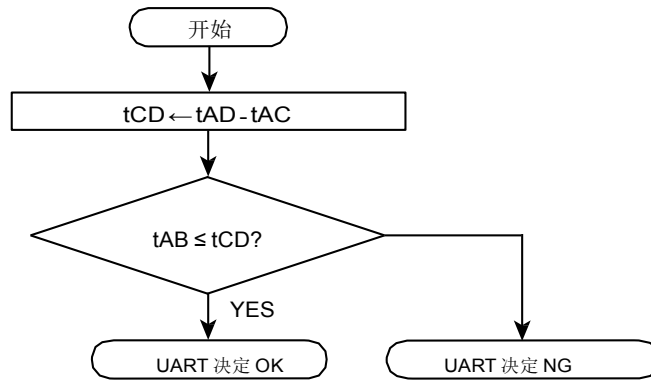


图 24-9 串行工作模式决定流程图

24.3.5.2 获取响应 数据

内部启动程序代表处理中状态以指定代码且传输到控制器. 表 24-19 到表 24-22 表示获取响应各接收数据的值.

表 24-20 到表 24-22 中, 获取响应的上四位等于那些工作命令数据的上 4 位. 第 3 位指接收错误. 第 0 位指无效工作命令, 校验和错误或密码错误. 第 1 位和第 2 位始终为 "0".

表 24-19 串行工作决定数据的 ACK 响应

传输 数据	描述
0x86	被决定为可以 UART 通信. (注)

注: 串行工作被决定为 UART 时, 如果波特率设置被决定为不可接受, 启动程序中止而不传输回任何响应.

表 24-20 工作命令数据的 ACK 响应

传输 数据	描述
0x?8 (注)	工作命令数据中接收错误发生.
0x?1 (注)	未定义工作命令数据正常接收.
0x10	被决定为 RAM 传输命令.
0x40	被决定为 flash 内存芯片擦除命令.

注: ACK 响应数据的上 4 位相同于之前的命令数据那些.

表 24-21 CHECK SUM 数据的 ACK 响应

传输 数据	描述
0xN8 (注)	接收错误发生.
0xN1 (注)	校验和或密码错误发生.
0xN0 (注)	校验和值被决定为正确.

注: ACK 响应数据的上 4 位相同于之前的命令数据那些.

表 24-22 Flash 内存芯片擦除和保护位擦除工作的 ACK 响应

传输 数据	描述
0x54	被决定为擦除使能命令.
0x4F	擦除命令结束
0x4C	擦除命令非法结束.
0x47	Flash 工作命令中止.

24.3.5.3 密码决定

启动程序使用如下区域作密码和密码数据区域必要性检查.

区域	地址
密码必要性检查检查	0x5E17_FFF0 (1 字节)
密码	0x5E17_FFF4 ~ 0x5E17_FFFF (12 字节)

RAM 传输命令执行密码认证无论必要性判断数据.

仅必要性判断被决定为 "需要"时 Flash 内存芯片擦除或保护位擦除命令执行密码认证.

密码需求 设置	数据
需要密码	0xFF 以外
无密码	0xFF

如果密码设置到0xFF (擦除的数据),由于易猜的密码很难安全保护数据. 即使单启动模式不被用, 建议设置独特的密码值.

(1) 使用 RAM 传输命令的密码认证

此项目描述"表 24-24 RAM 传输的通信规则 "的 No.5 中描述的密码决定.

如果所有这些地址场所包含 0xFF 以外的相同数据字节, 此条件被决定为密码区域错误如图 24-10 中所示. 此时, 启动程序返回错误承认 (0x11)响应校验值无论密码认证.

启动程序认证接收数据 (密码数据).如果所有 12 字节不匹配则密码错误发生. 如果密码错误被决定, ACK 响应变为密码错误.

即使安全功能使能时执行密码认证.

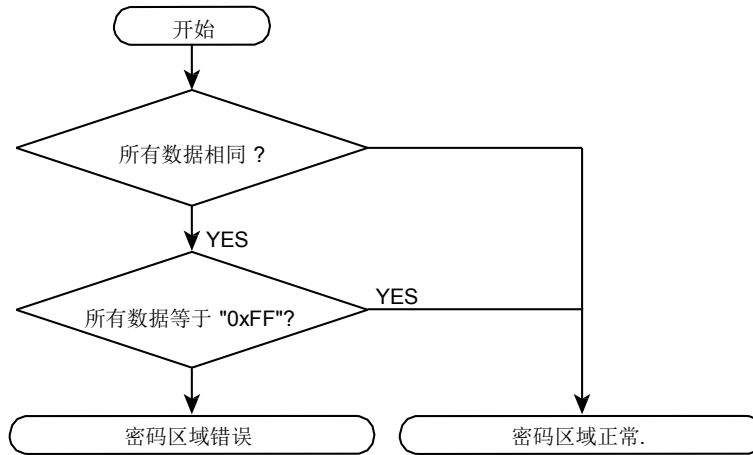


图 24-10 密码区域检查流程图

(2) Flash 内存芯片擦除和保护位擦除命令的密码认证

此项目描述"表 24-25 Flash 内存芯片擦除和保护位擦除的通信规则"的 No.5 中描述的密码决定。

擦除密码必要性决定区域中密码有效时如图 24-11 中所示。如果密码为理想的数据，密码区域错误发生。如果密码区域错误被决定，CHECK SUM 值的 ACK 响应传输 0x41 无论密码认证。

启动程序认证接收数据 (密码数据)。如果所有 12 字节不匹配则密码错误发生。如果密码错误被决定，CHECK SUM 数据的 ACK 响应数据为密码错误。即使安全功能使能时执行密码认证。

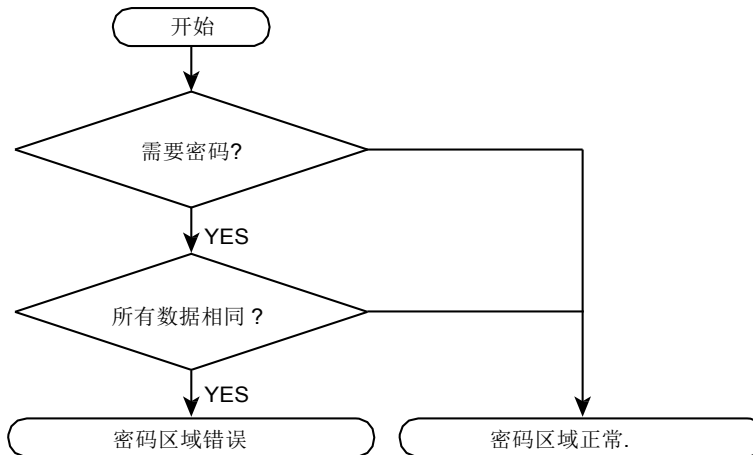


图 24-11 密码区域检查流程图

24.3.5.4 CHECK SUM 计算

CHECK SUM 通过 8-位附加(忽略溢出) 计算来传输数据且有两个低 8 位和的补充. 控制器传输 CHECK SUM 值时使用此计算.

CHECK SUM 计算例

为了计算 2 字节数据 (0xE5 和 0xF6)的 CHECK SUM, 执行 8-位附加.

$$0xE5 + 0xF6 = 0x1DB$$

有两个低 8 位和的补充, 且那是校验和值.所以启动程序传输 0x25 到控制器.

$$0 - 0xDB = 0x25$$

24.3.6 串行工作模式决定的通信规则

此部分描述串行工作模式决定的通信规则. 表中传输方向如下所指:

传输方向 (C→T): 控制器 到 TPM475

传输方向 (C←T): TPM475 到 控制器

表 24-23 串行工作模式决定的通信规则

No	传输方向	传输数据	描述
1	C→T	串行工作模式和波特率设置	控制器传输数据决定串行工作模式. 关于目标模式决定的详细, 参考 "24.3.5.1 串行工作模式决定".
		0x86	控制器传输 0x86. 如果目标决定 UART 模式为 OK, 目标依次决定波特-率设置是否可以. 如果不, 关闭程序停止和通信.
2	C←T	串行工作模式的 ACK 响应	从控制器接收数据为对应串行工作模式设置数据的第 1 字节的 ACK 响应数据. 如果目标决定设置为可以, 目标设置 UART. 传输缓冲写入数据前设置接收使能时序.
		正常 状态: 0x86	如果目标决定设置为可以, 目标传输 0x86. 如果目标决定设置为不可以, 目标不传输任何且停止工作. 控制器完成传输数据第 1 字节后设置结束时间 (5 sec.). 如果在结束时间内控制器不正常接收数据 (0x86), 应该被决定为通信失败. 如果在结束时间内数据 (0x86) 不正常接收, 不可以通信.
3	-	-	控制器传输工作命令数据. 关于各工作命令传输格式的详细, 参考 "24.3.7 RAM 传输时通信规则" 或 "24.3.8 Flash 内存芯片擦除和保护位擦除通信规则".

24.3.7 RAM 传输时通信规则

此部分表示 RAM 传输通信规则. 表中传输方向如下所指:

传输 方向 (C→T): 控制器 到 TMPM475

传输 方向 (C←T): TMPM475 到 控制器

表 24-24 RAM 传输通信规则

No	传输方向	传输数据	描述
1	C→T	工作命令数据 (0x10)	控制器传输 RAM 传输命令数据 (0x10).
2	C←T	工作命令的 ACK 响应 正常: 0x10 异常: 0x11 通信 错误: 0x18	目标检查接收数据和传输 ACK 响应数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x18 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 根据表 24-18 中描述的工作命令数据, 目标检查数据. 如果检查失败, 目标响应 ACK 响应数据 0x11 指异常状态, 然后返回初始状态等待工作命令数据. 如果检查成功, 目标响应 ACK 响应数据 0x10 指正常状态, 然后等待下一个数据.
3	C→T	密码数据 (12 字节)	控制器传输 Flash 内存密码数据相同区域的数据. 关于密码数据区域, 参考 "24.3.5.3 密码决定".
4	C→T	传输数据 (No.3)的 CHECK SUM 值	控制器传输传输数据 (No.3)的 CHECKSUM 值. 关于 CHECK SUM 计算的详细, 参考 "24.3.5.4 CHECK SUM 计算".
5	C←T	CHECK SUM 值的 ACK 响应 正常: 0x10 异常: 0x11 通信 错误: 0x18	目标检查接收数据和响应 ACK 响应数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x18 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标检查 CHECK SUM 值和密码.关于密码决定的详细, 参考 "24.3.5.3 密码决定". 如果密码决定失败, 目标响应 ACK 响应数据 0x11 指异常状态, 然后返回初始状态等待工作命令数据. 如果密码决定成功, 目标响应 ACK 响应数据 0x10 指正常状态, 然后等待下一个传输数据.
6	C→T	RAM 储存 开始 地址 31 ~ 24	通过从控制器分割 4 次为下一个传输数据, 传输储存在 RAM 储存数据的 RAM 开始地址. 传输命令如下: 第 1 字节对应 31 位 ~ 24 位和第 4 字节对应传输地址的第 7 位 ~ 第 0 位. 这些地址应该放在 0x2000_0400 到 RAM 地址的最后地址.
7	C→T	RAM 储存 开始 地址 23 ~ 16	
8	C→T	RAM 储存 开始 地址 15 ~ 8	
9	C→T	RAM 储存 开始 地址 7 ~ 0	目标检查接收数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x18 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标不传输任何, 等待下一个传输数据.
10	C→T	RAM 储存 字节 15 ~ 8 的数	从控制器传输要方块-传输的字节数. 传输顺序如下: 第 1 字节对应 15 位 ~ 8 位和第 2 字节对应传输地址的第 7 位 ~ 第 0 位. 这些地址应该放在 0x2000_0400 到 RAM 地址的最后地址. 目标检查接收数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x18 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标不传输任何, 等待下一个传输数据.
11	C→T	RAM 储存 字节 7 ~ 0 的数	
12	C→T	CHECK SUM 传输 数据 (No.6 ~ 11)的值	从控制器传输传输数据 (No.6 ~ 11)的 CHECK SUM 值.

No	传输方向	传输数据	描述
13	C←T	CHECK SUM 值的 ACK 响应 正常: 0x10 异常: 0x11 通信 错误: 0x18	目标检查接收数据和响应 ACK 响应数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x18 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标检查 CHECK SUM 值. 如果检查失败, 目标响应 ACK 响应数据 0x11 指异常状态, 然后返回初始状态等待工作命令数据. 如果检查成功, 目标响应 ACK 响应数据 0x10 指正常状态, 然后等待下一个数据.
14	C→T	RAM 储存的数据	从控制器传输储存在 RAM 的数据. 目标接收储存在 RAM 的数据.
15	C→T	传输数据 (No. 14)的 CHECK SUM 值	从控制器传输传输数据 (No.14)的 CHECK SUM 值.
16	C←T	CHECK SUM 值的 ACK 响应 正常:0x10 异常: 0x11 通信 错误: 0x18	目标检查接收数据和响应 ACK 响应数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x18 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标检查 CHECK SUM 值. 如果检查失败, 目标响应 ACK 响应数据 0x11 指异常状态, 然后返回初始状态等待工作命令数据. 如果检查成功, 目标响应 ACK 响应数据 0x10 指正常状态和跳跃到 RAM 储存开始地址 (No.6 ~ 9) 作分支地址.

24.3.8 闪存芯片擦除和保护位擦除的通信规则

此部分表示 Flash 内存芯片擦除和保护位擦除命令的通信格式. 表中传输方向如下所指:

传输方向 "C→T": 控制器 到 TMPM475

传输方向 "C←T": TMPM475 到 控制器

表 24-25 闪存芯片擦除和保护位擦除的通信规则

No	传输方向	传输数据	描述
1	C→T	工作命令数据 (0x40)	传输 Flash 内存芯片擦除和保护位擦除命令数据 (0x40).
2	C←T	工作命令的 ACK 响应 正常: 0x40 异常: 0x41 通信错误: 0x48	ACK 响应数据工作命令. 先, 检查接收数据第 3 字节是否有错误. (仅 UART 模式) 如果接收错误存在, 传输 ACK 响应数据 0xX8 意味异常通信且等待下一个工作命令 (第 3 字节). 传输数据的上 4 位未定义. (相同于工作命令数据直前的上 4 位.) 注意 I/O 接口中, 接收错误检查不执行. 然后如果接收数据第 3 字节对应表 24-13 中其工作命令数据, 接收数据为回声. 如果数据不对应表 24-13 中命令, 传输 ACK 响应数据 0xX1 意味工作命令错误, 且等待下一个工作命令. (3rd 字节) 传输数据上 4 位未定义. (工作命令数据被用直前的上 4 位.)
3	C→T	密码数据 (12 字节)	从控制器传输 Flash 内存密码数据相同区域的数据. 然而; 如果 Flash 内存的密码需要设置为 "无" (数据: 0xFF), 目标不执行密码验证, 为了虚拟数据可以被用作密码. 关于密码数据区域的详细, 参考 "24.3.5.3 密码决定".
4	C→T	传输数据 (No.3) 的 CHECK SUM 值	从控制器传输一个传输数据 (No.3) 的 CHECK SUM 值. 关于 CHECK SUM 计算的方法, 参考 "24.3.5.4 CHECK SUM 计算".
5	C←T	CHECK SUM 值的 ACK 响应 正常: 0x40 异常: 0x41 通信错误: 0x48	目标检查接收数据和响应 ACK 响应数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x48 指异常通信, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标检查 CHECK SUM 值. 如果检查失败, 目标响应 ACK 响应数据 0x41 指异常通信, 然后返回初始状态等待工作命令数据. 如果检查成功, 目标执行密码检查. 如果密码需要设置为 "无", 目标传输 ACK 响应数据 0x40 指正常. 如果密码需要设置为 "需要密码", 目标检查密码. 如果密码检查失败, 目标响应 ACK 响应数据 0x41 指异常通信, 然后返回初始状态等待工作命令数据. 如果密码检查成功, 目标响应 ACK 响应数据 0x40 指正常, 然后等待下一个数据.
6	C→T	擦除使能命令数据 (0x54)	从控制器传输擦除使能命令数据 (0x54).
7	C←T	擦除使能命令的 ACK 响应 正常: 0x54 异常: 0x51 通信错误: 0x58	目标检查接收数据和响应 ACK 响应数据. 如果接收错误存在, 目标响应 ACK 响应数据 0x58 指通信错误, 然后返回初始状态等待工作命令数据. 如果接收错误不存在, 目标检查擦除使能命令 (0x54). 如果检查失败, 目标响应 ACK 响应数据 0x51 指异常通信, 然后返回初始状态等待工作命令数据. 如果检查成功, 目标响应 ACK 响应数据 0x54 正常, 然后芯片擦除处理执行.
8	C←T	擦除命令的 ACK 响应 正常: 0x4F 异常: 0x4C 中止芯片擦除命令: 0x47	目标响应芯片擦除处理的结果. 如果任何问题发生, 目标响应 ACK 响应数据 (0x4F) 指正常. 如果空白检查错误发生, 目标响应 ACK 响应数据 (0x4C) 指异常. 如果芯片擦除命令中止, 目标响应 ACK 响应数据 (0x47) 指中止然后返回初始状态等待工作命令数据.

24.3.9 启动程序 整体流程图

此部分表示启动程序整体流程图.

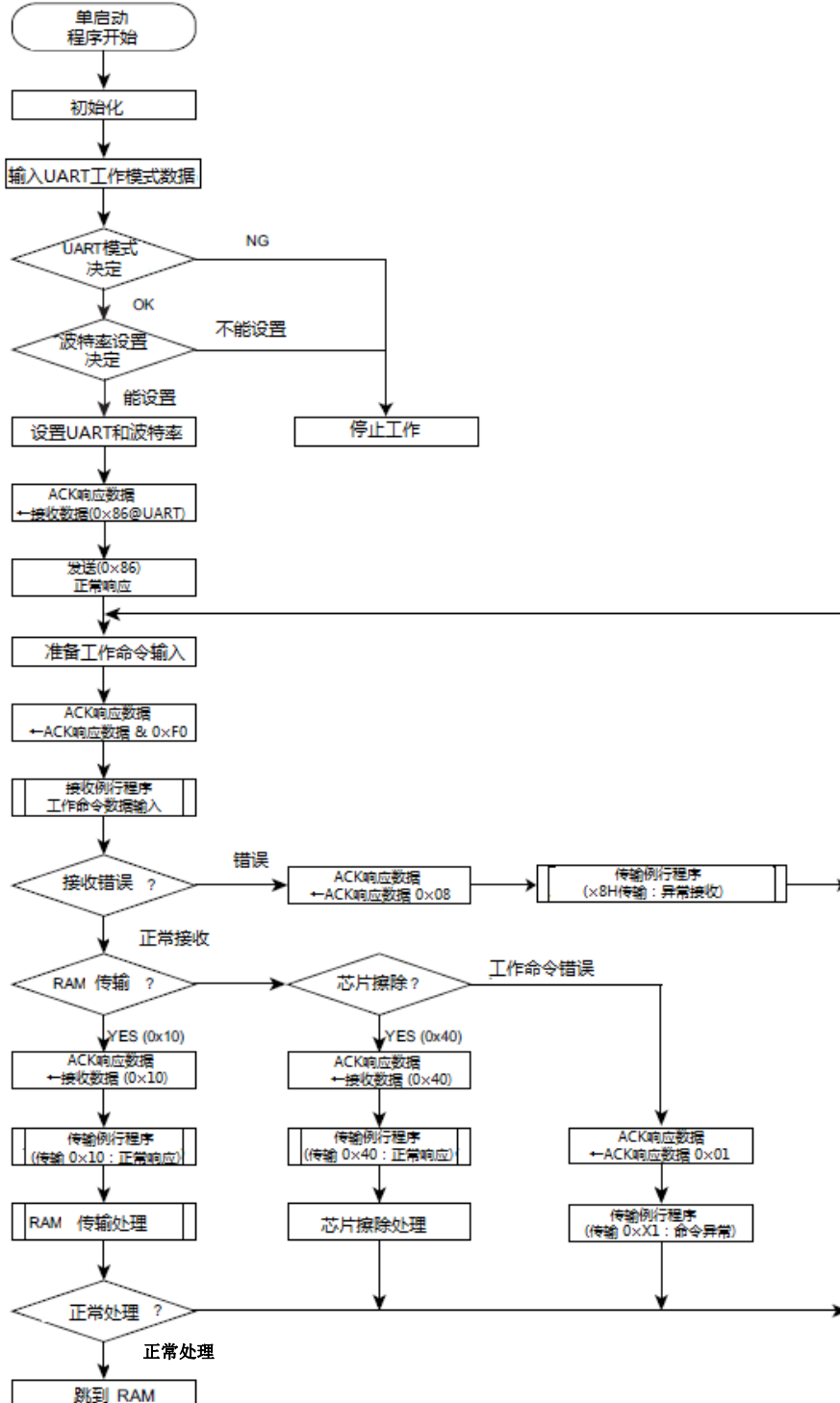


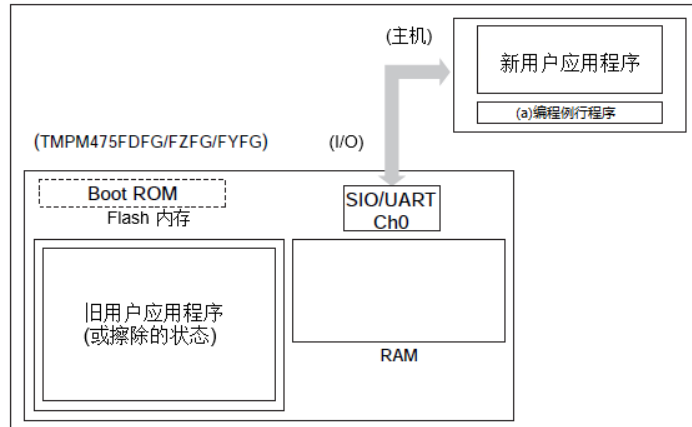
图 24-12 启动程序整体流程图

24.3.10 使用 BOOT ROM 中的程序对闪存进行重编程的步骤

此部分描述片上启动 ROM下使用重新编程算法的 flash重新编程处理.(下例使用 UART/SIO 通道 ch0)

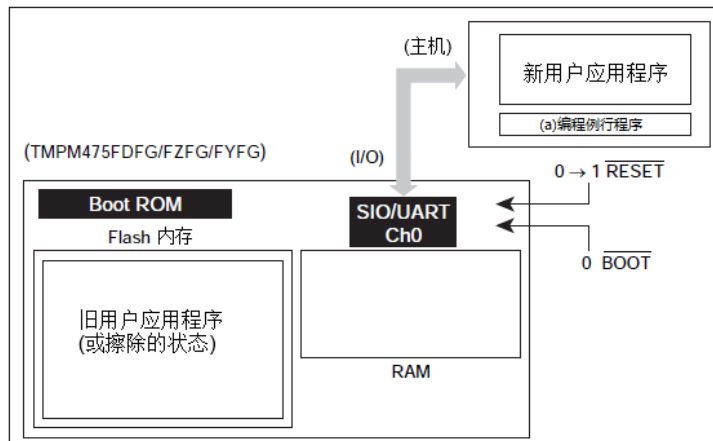
24.3.10.1 步-1

Flash 内存条件不需要关注以前的用户程序被写入或擦除. 因为编程序序和编程数据通过 SIO (SIO0) 传输, SIO0 必须被连接到外部主机. 编程序序 (a) 准备在主机上.



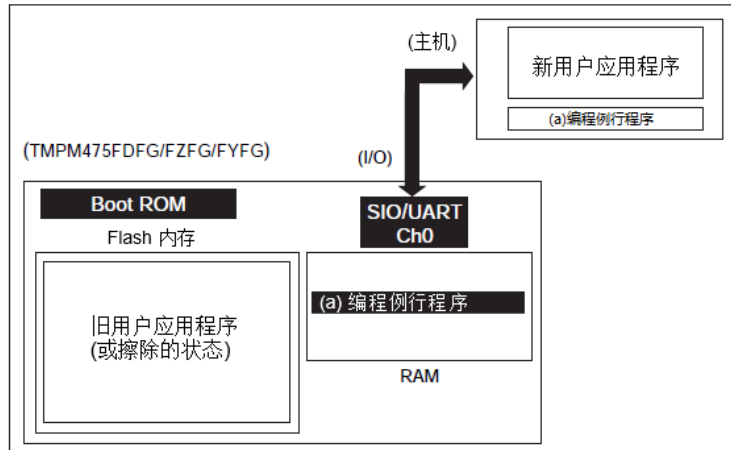
24.3.10.2 步-2

启动模式和 BOOT ROM 上启动中通过引脚条件设置释放复位. 根据启动模式的处理, 从源 (主机) 通过 SIO0 传输编程序序 (a). 先执行用户应用程序中密码验证. (如果 Flash 内存被擦除, 擦除数据 (0xFF) 被处理为密码.)



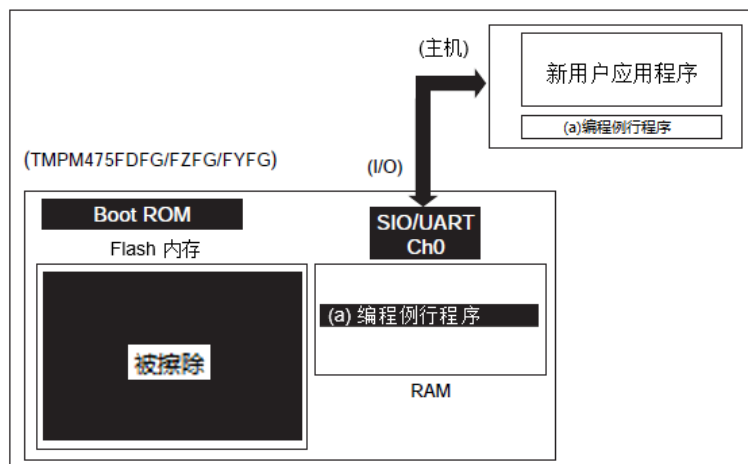
24.3.10.3 步-3

如果密码验证完成, 启动程序从主机传输编程程序 (a) 到片上 RAM. 编程程序必须储存在从 0x2000_0400 到 RAM 结束地址的范围.



24.3.10.4 步-4

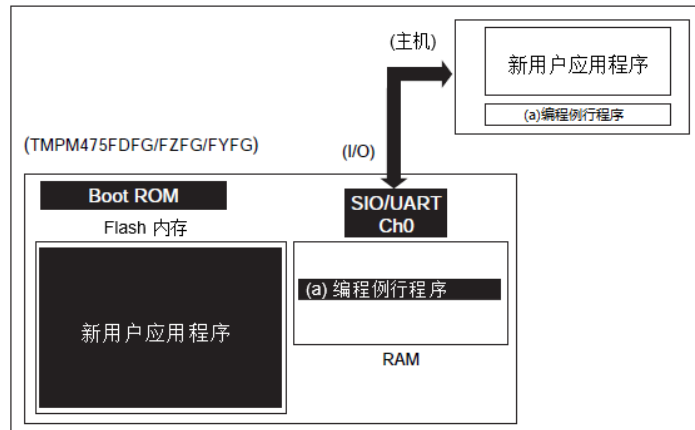
启动程序跳到片上 RAM 中编程程序 (a) 擦除含有旧应用程序代码 (擦除单元可以为任何) 的 Flash 方块.



24.3.10.5 步-5

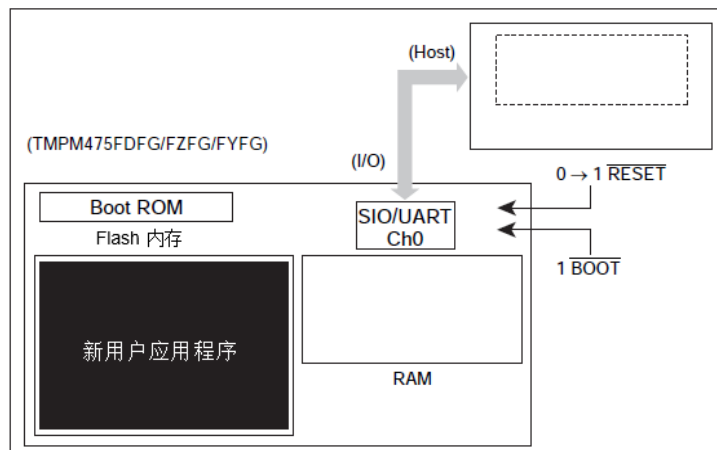
启动程序执行编程调度程序 (a) 从主机下载新应用程序代码且编程它到擦除的 flash 区域. 编程完成时, 用户程序中 Flash 区域的写入或擦除保护必须 ON.

下例中, 新程序代码通过相同 SIO0 通道来自相同主机作为编程程序. 然而, 一次编程程序已开始执行, 自由改变传输路径和传输源. 制造一个硬件板和编程程序来适合你特殊需要.



24.3.10.6 步-6

Flash 内存编程完成时, 断电板子且分离从主机到目标板的线. 再上电, 为了设备在单-芯片(正常)模式中重启来执行新程序.



24.4 用户启动模式下重新编程

用户启动模式在用户设置的内部 RAM 上使用 Flash 内存编程. 用户应用上的 Flash 内存程序代码的数据传输总线与串行 I/O 不同时这个被用. 工作在单芯片模式; 因此, 正常模式中, 用户应用被激活在单芯片模式, 为了编程 Flash 内存需要切换到用户启动模式. 因此, 用户应用程序中加一个模式调节调度程序来复位服务程序.

根据用户系统设置条件需要设置切换模式的条件. 同时, 新应用中需要设置用户独特完成的 Flash 内存编程程序. 被切换到用户启动模式后此调度程序被用作编程. 一旦重新编程完成, 单芯片模式 (正常工作模式) 中建议需要方块的写入/ 擦除保护来避免无意的修改. 确保即使用户启动模式中不要产生异常避免异常中止.

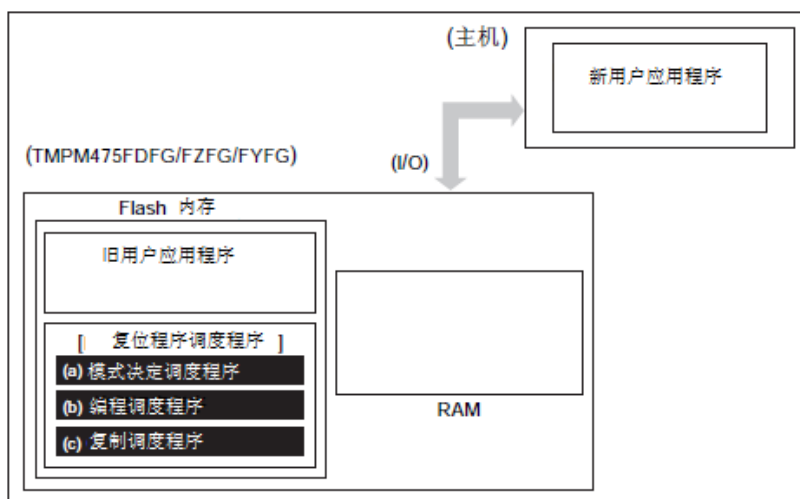
从两情况诸如重新编程储存在 Flash 内存 (1-A) 和从外部设备 (1-B) 传输的调度程序的方法举例, 以下部分说明处理. 关于程序/ 擦除 Flash 内存的详细, 参考 "24.2 Flash 内存的详细".

24.4.1 (1-A) 储存在内存的编程程序处理

24.4.1.1 步-1

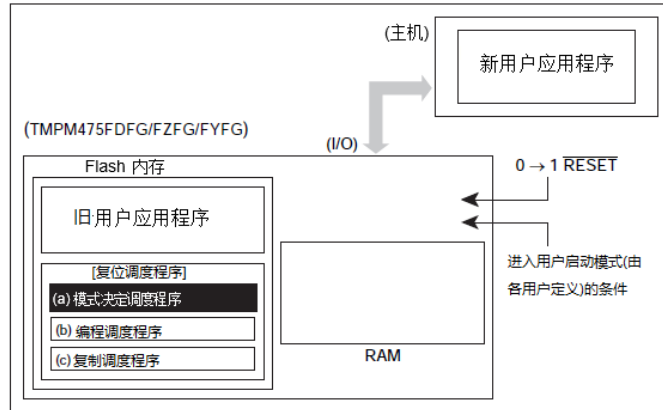
用户决定进入用户启动模式和被用来传输数据的 I/O 总线条件 (比如, 引脚状态). 然后制作适合的电路设计和程序. 安装设备到印制电路板上前, 使用诸如 Flash 写入器的编程设备写入以下三个程序调度程序到随意 Flash 方块.

- | | |
|-------------------|-----------------------------|
| (a) 模式决定调度程序: | 决定是否切换到用户启动模式的程序 |
| (b) Flash 编程调度程序: | 从外部设备下载新程序和重新编程 Flash 内存的程序 |
| (c) 复制调度程序: | 复制 (a) 中描述的数据到内置 RAM. |



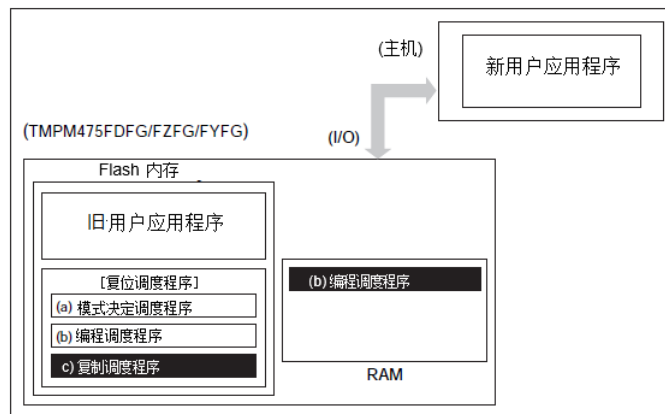
24.4.1.2 步-2

此部分说明储存在复位调度程序的编程调度程序情况. 首先, 复位调度程序决定进入用户启动模式. 如果满足模式切换条件, 设备进入用户启动模式重新编程数据. (进入用户启动模式后禁止生成所有异常)



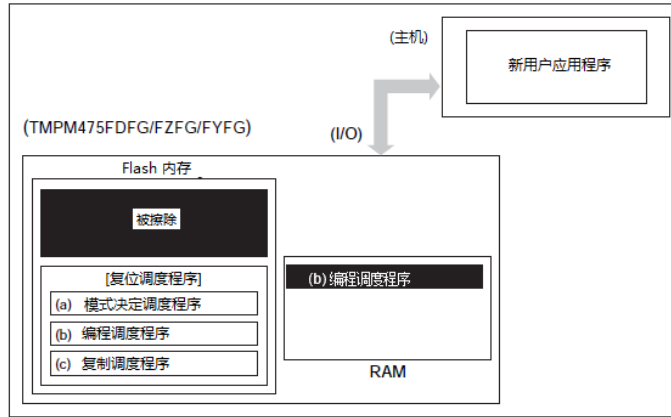
24.4.1.3 步-3

一旦设备进入用户启动模式, 执行复制调度程序 (C) 从主机控制器下载 Flash 编程调度程序 (b) 到内部 RAM.



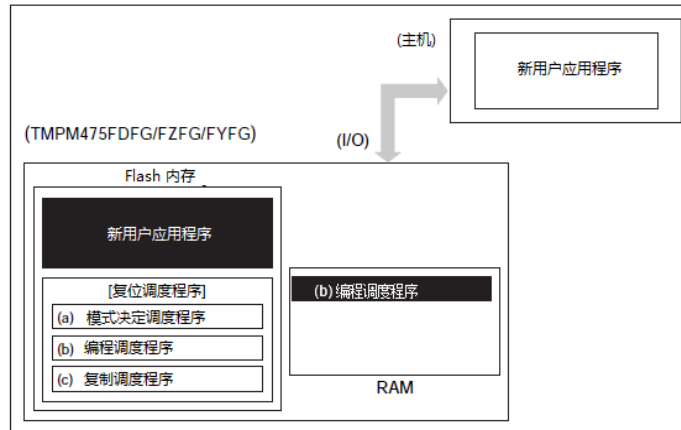
24.4.1.4 步-4

跳到 RAM 上重新编程例行程序释放旧应用程序写入/擦除保护, 且擦除 Flash.
(擦除单元可以为任何)



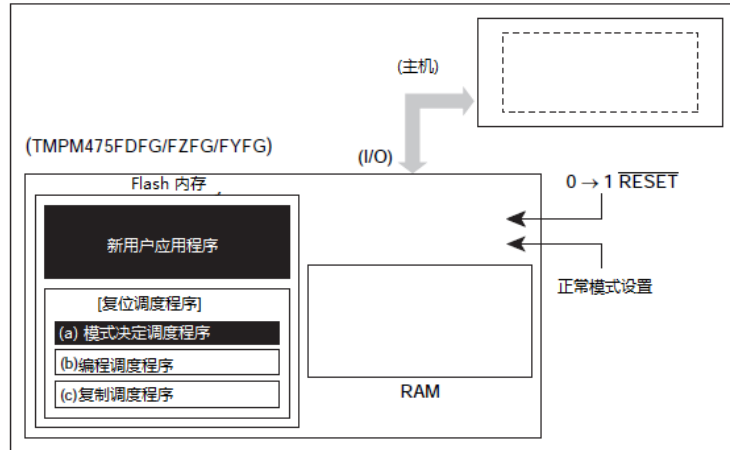
24.4.1.5 步-5

继续执行 flash 编程例行程序从主机控制器下载新程序数据且编程它到擦除的 Flash 方块.编程完成时, 用户程序区域中 flash 方块写入/擦除保护必须为 ON.



24.4.1.6 步-6

通过设置 "0" 到 $\overline{\text{RESET}}$. 复位中, Flash 内存设置为正常模式. 复位后, CPU 会开始新应用程序.



24.4.2 (1-B) 传输自外部主机的编程程序处理

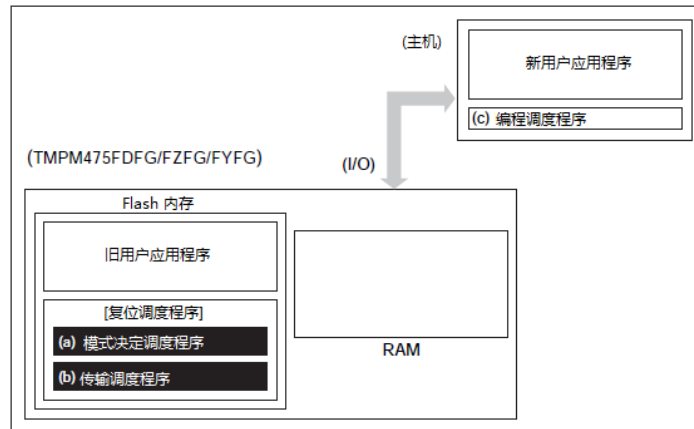
24.4.2.1 步-1

用户决定进入用户启动模式和被用来传输数据 I/O 总线的条件(比如, 引脚状态). 然后制作合适的电路设计和程序. 在印刷电路板上安装设备前, 使用编程设备诸如 Flash 写入器写入以下两例行程序到任意 Flash 方块.

- (a) 模式 决定 程序: 决定切换编程工作的程序
- (b) 传输 程序: 从外部设备获取编程程序的程序.

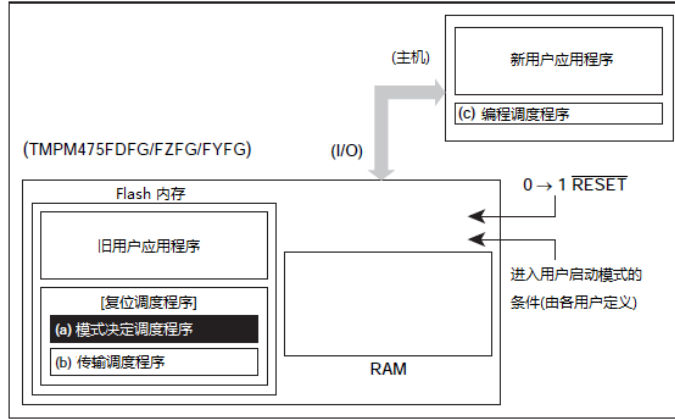
另外,准备如下所示重新编程例行程序必须储存在主机控制器.

- (c) 重新编程例行程序: 重新编程数据的程序



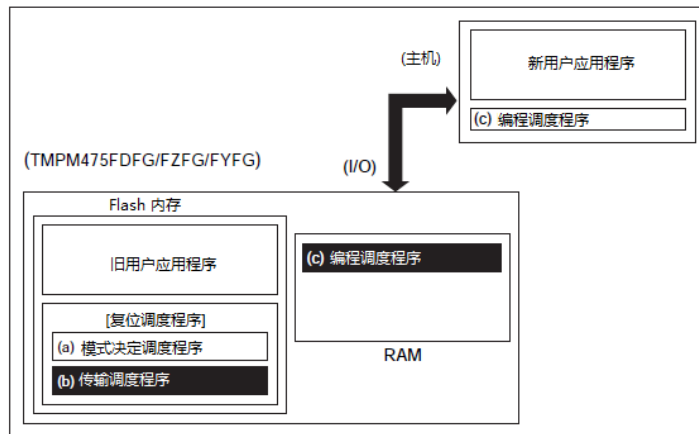
24.4.2.2 步-2

此部分说明储存在复位例程序的编程例行的情况. 首先, 复位程序程序决定进入用户启动模式. 如果满足模式切换条件, 设备进入用户启动模式重新编程数据. (进入用户启动模式后禁止产生所有异常)



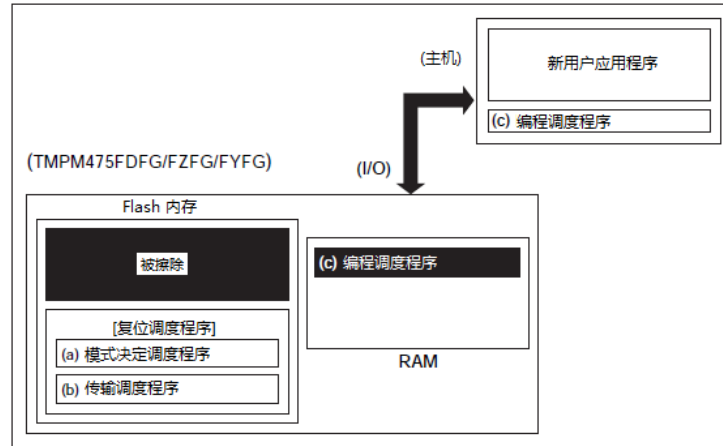
24.4.2.3 步-3

一旦设备进入用户启动模式, 执行传输例程序 (b) 从主机 控制器下载编程例程序 (c) 到内部 RAM.



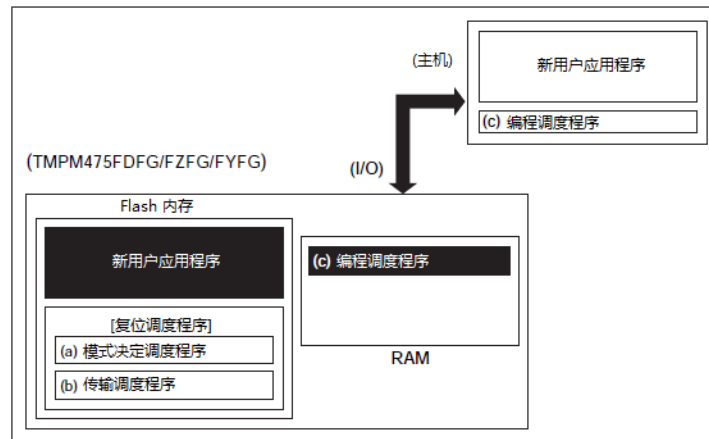
24.4.2.4 步-4

跳到内部 RAM 上重新编程例行程序释放旧应用程序写入/擦除保护, 且擦除 Flash. (擦除单元可以为任何)



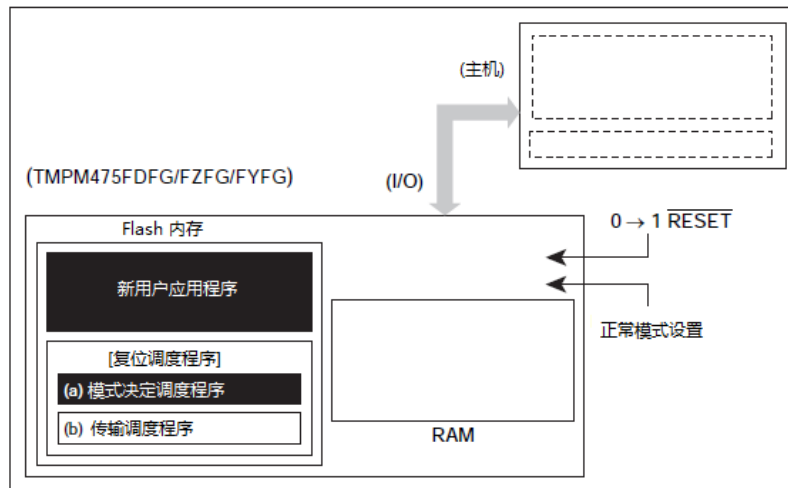
24.4.2.5 步-5

继续执行 Flash 编程例行程序 (c) 从主机控制器下载新程序数据且编程它到擦除的 Flash 方块. 编程完成时, 用户程序区域中 Flash 方块写入/擦除保护必须为 ON.



24.4.2.6 步-6

通过设置 "0" 到 $\overline{\text{RESET}}$.复位中, Flash 内存设置为正常模式.复位后, CPU 会开始新应用程序.



24.5 如何重新编程使用用户启动模式的闪存

Flash 内存重新编程时此方法切换页 0 区域到页 1 区域离开使用交换功能的用户启动程序。

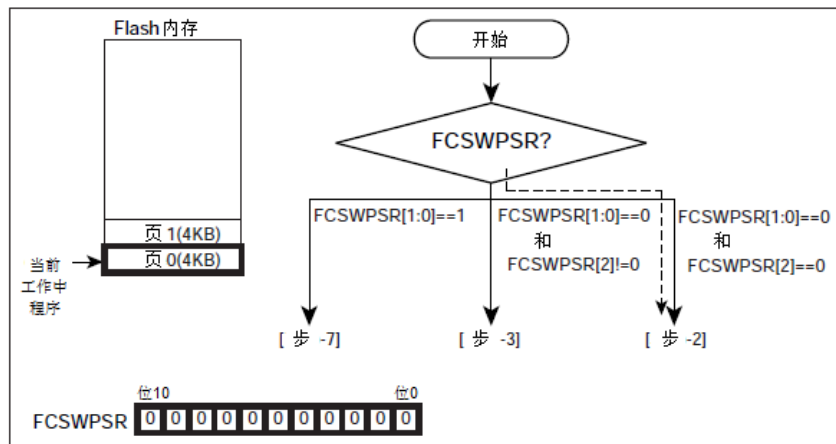
以下为用户启动程序重新编程处理例。

(假设条件作以下解释: 交换大小为 4K 字节. 从页 0 复制页 1 程序.)

24.5.1 闪存重新编程处理例

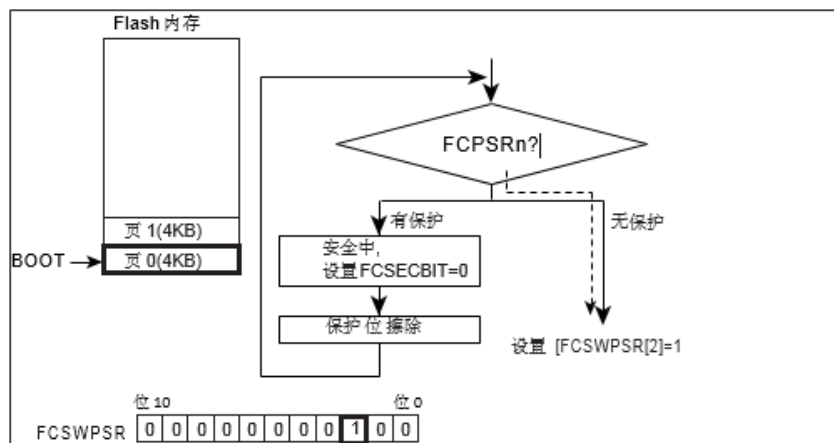
24.5.1.1 步-1

确认 0x0 是否从 FCSWPSR[2:0] 读。



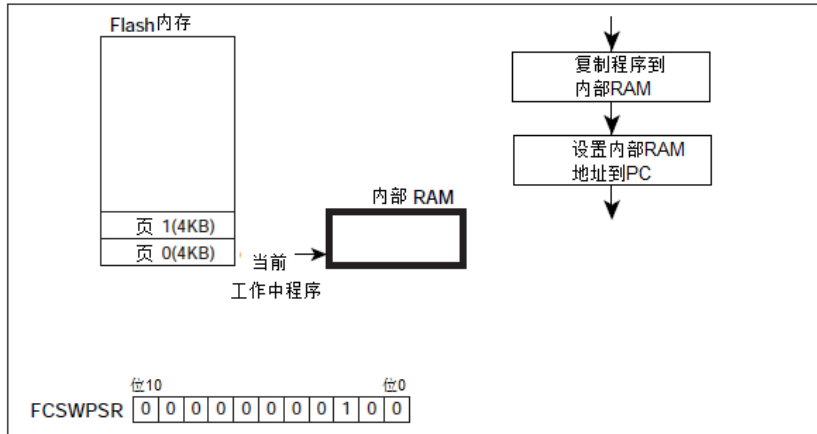
24.5.1.2 步-2

检查 FCPSR 各位确认保护状态是否释放. 然后使用自动内存交换命令设置"1" 到 FCSWPSR[2].



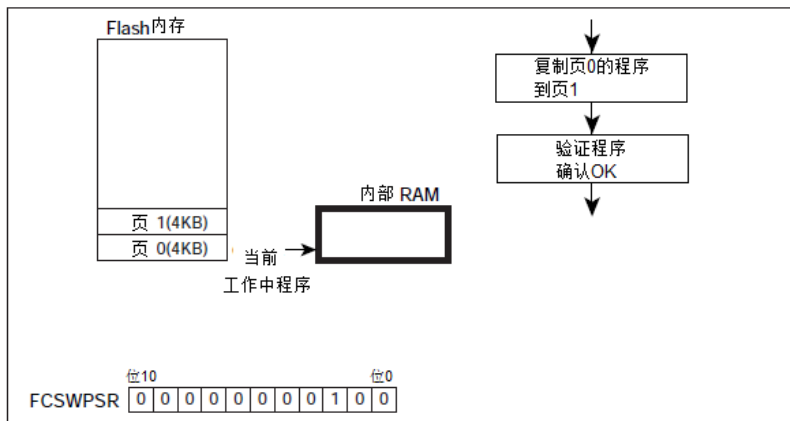
24.5.1.3 步-3

传输重新编程例程序到内部 RAM. 把 PC (程序计数器) 移到传输的程序.



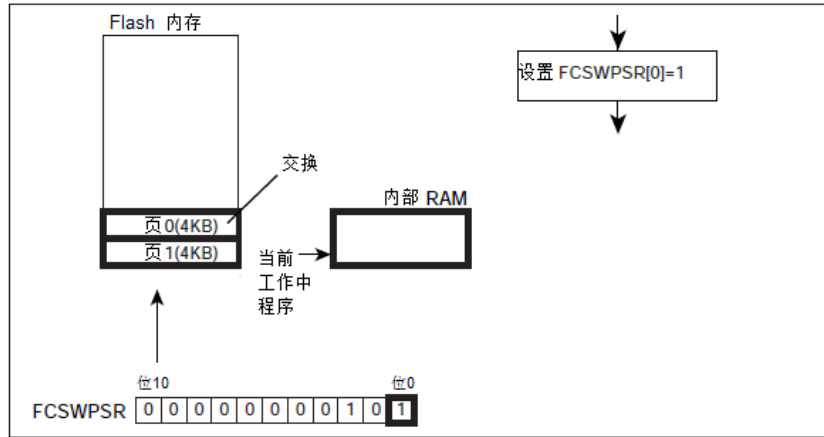
24.5.1.4 步-4

擦除 页 1. 然后写入页 0的程序到页 1 程序.



24.5.1.5 步-5

自动内存交换命令设置 "1" 到 FCSWPSR[0] 用 页 1 交换 页 0.

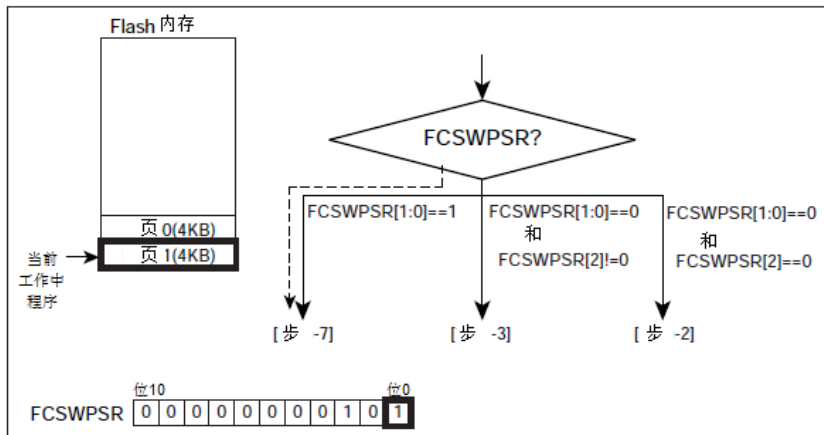


24.5.1.6 步-6

执行或释放 复位.

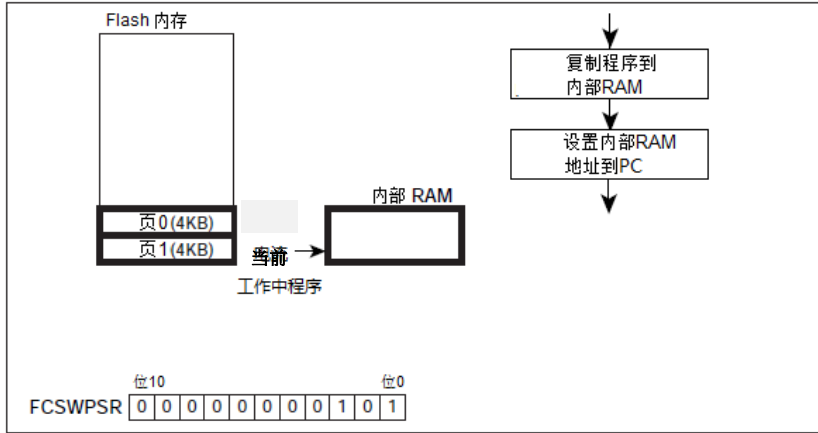
页 1 被分配到地址 0 且 Flash 内存在页 1 启动.

程序分支到 FCSWPSR[1:0]设置为"1"的条件例行程序.(到 [步-7])



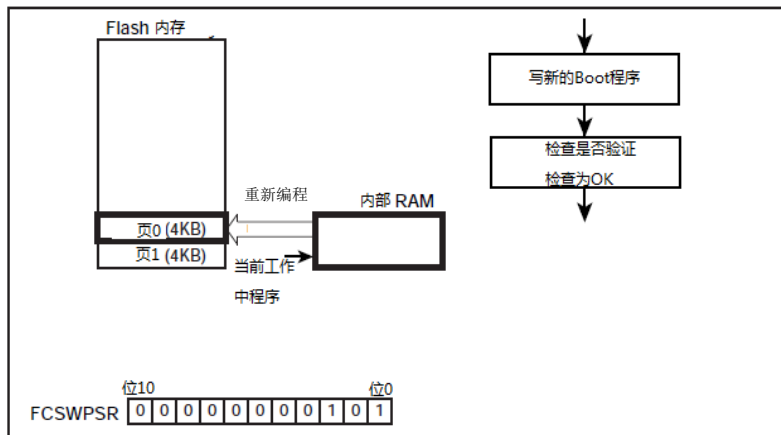
24.5.1.7 步-7

传输 Flash 重新编程例行程序到内部 RAM 然后设置 PC (程序计数器)的内部 RAM 地址.



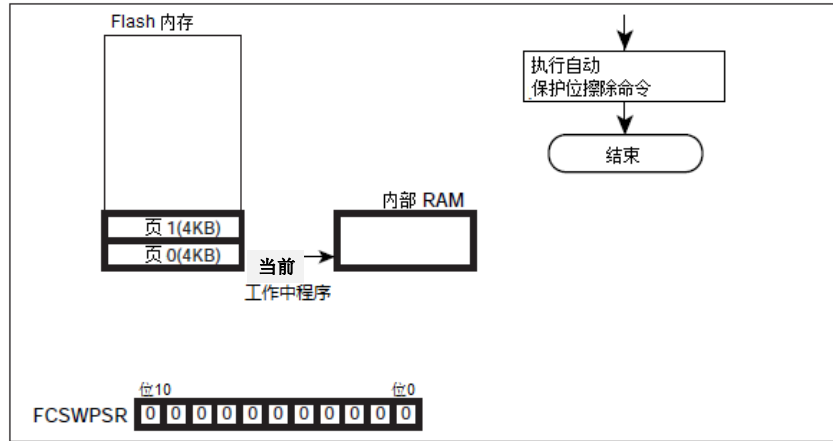
24.5.1.8 步-8

写入新启动程序到页 0.



24.5.1.9 步-9

执行自动保护位擦除命令。



25. 调试 接口

25.1 规格 总览

TMPM475DFDG/FZFG/FYFG 包含有调试工具的串行线 JTAG 调试端口 (SWJ-DP) 单元接口和指令跟踪输出的嵌入跟踪宏单元™(ETM) 单元.跟踪数据通过片上跟踪端口接口单元 (TPIU)输出到调试的专用引脚 (TRACEDATA[3:0], SWV) .

关于 SWJ-DP, ETM 和 TPIU 的详细, 参考 "ARM Cortex-M4F 设置文档".

25.2 SWJ-DP

SWJ-DP 支持串行线调试端口 (SWCLK, SWDIO) 和 JTAG 调试端口 (TDI, TDO, TMS, TCK, $\overline{\text{TRST}}$).

引脚名	功能	描述	I/O
TMS	JTAG	JTAG 测试 模式 选择	输入
SWDIO	SW	串行线 数据 输入/输出	I/O
TCK	JTAG	JTAG 测试 时钟	输入
SWCLK	SW	串行线 时钟	输入
TDO	JTAG	JTAG 测试 数据 输出	输出
SWV	SW	(串行线 观察 输出)	(输出) (注)
TDI	JTAG	JTAG 测试 数据 输入	输入
$\overline{\text{TRST}}$	JTAG	JTAG 测试 $\overline{\text{RESET}}$	输入

注:SWV 功能被用时, 此引脚被用作输出引脚.

25.3 ETM

ETM 支持四个数据信号引脚 (TRACEDATA[3:0]), 一个时钟信号引脚 (TRACECLK) 且从串行线观察器 (SWV)跟踪输出.

25.4 外设功能停止模式

Cortex-M4F 核进入停止模式时, 看门狗-定时器 (WDT) 自动停止.可选择 16-位定时器(TMRB 和 TMR16A) 继续或停止计数.其他外设功能继续工作.

25.5 带调试工具连接

25.5.1 关于带调试工具连接

关于用调试工具连接, 参考制造建议.

调试接口引脚包含上拉电阻和下拉电阻. 用外部上拉或下拉连接调试接口引脚时, 请注意输入电平.

注: 确保停止模式中用被连接的调试工具测量功耗被禁止.

25.5.2 使用用作通用目的的端口的调试接口的引脚要点

调试接口引脚也可以被用作通用端口.

释放复位后, 调试接口引脚的特殊引脚被初始化为调试接口引脚. 如果需要的话另一个调试接口引脚应该被改变为调试接口引脚.

如果调试接口引脚被用作总 I/O 端口, 请提前准备改变总 I/O 端口为调试接口引脚的方法.

表 25-1 使用调试接口引脚的例表

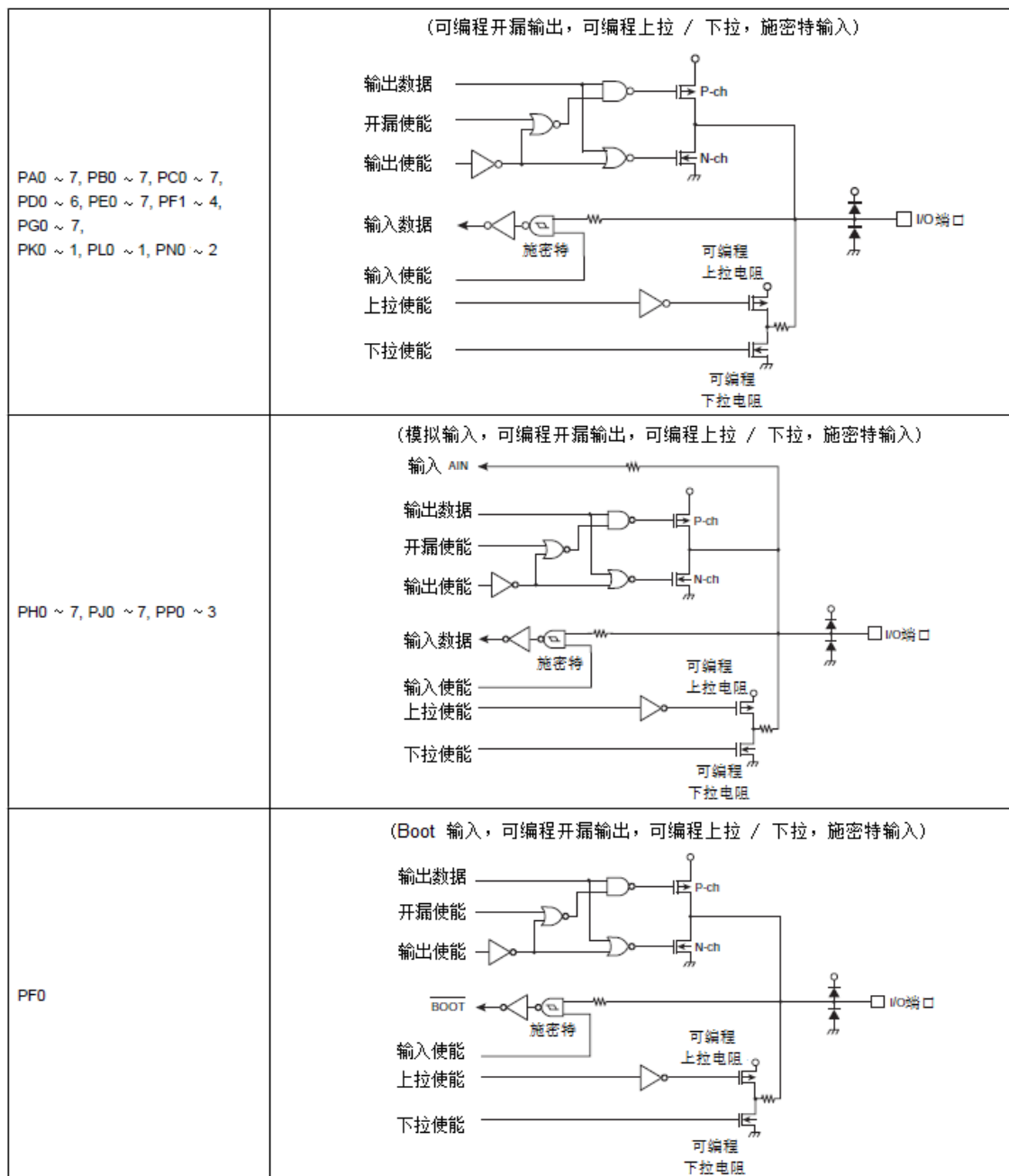
	调试接口引脚						
	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA[3:0]	TRACE CLK
JTAG+SW (复位后)	0	0	0	0	0	x	x
JTAG+SW (无 $\overline{\text{TRST}}$)	x	0	0	0	0	x	x
JTAG+TRACE	0	0	0	0	0	0	0
SW	x	x	x	0	0	x	x
SW+SWV	x	x	0	0	0	x	x
调试 禁止	x	x	x	x	x	x	x

0: 使能 x: 禁止 (可用作通用 端口)

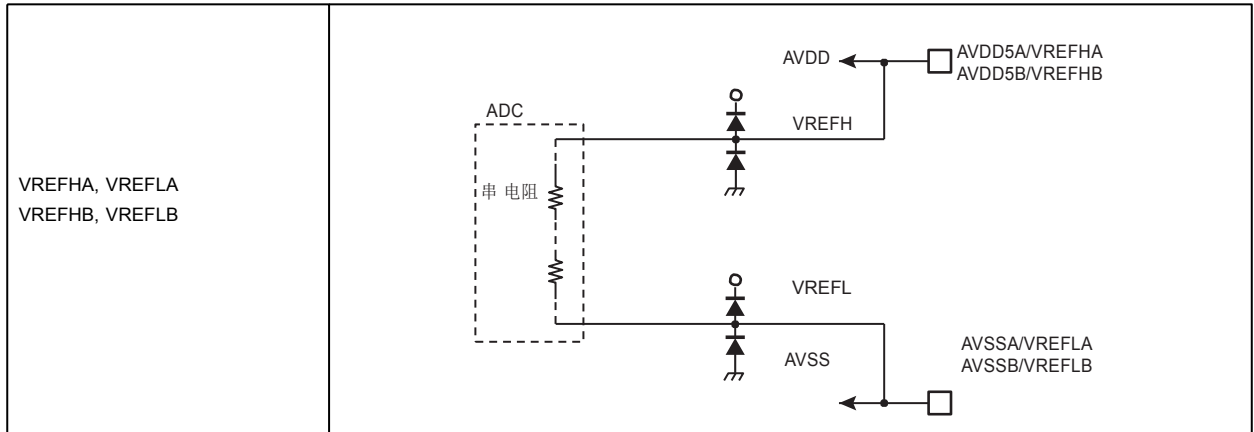
26. 端口部分等效电路图

基本上, 写入的门符和那些被用作标准 CMOS 逻辑 IC [74HCXX] 系列的相同。

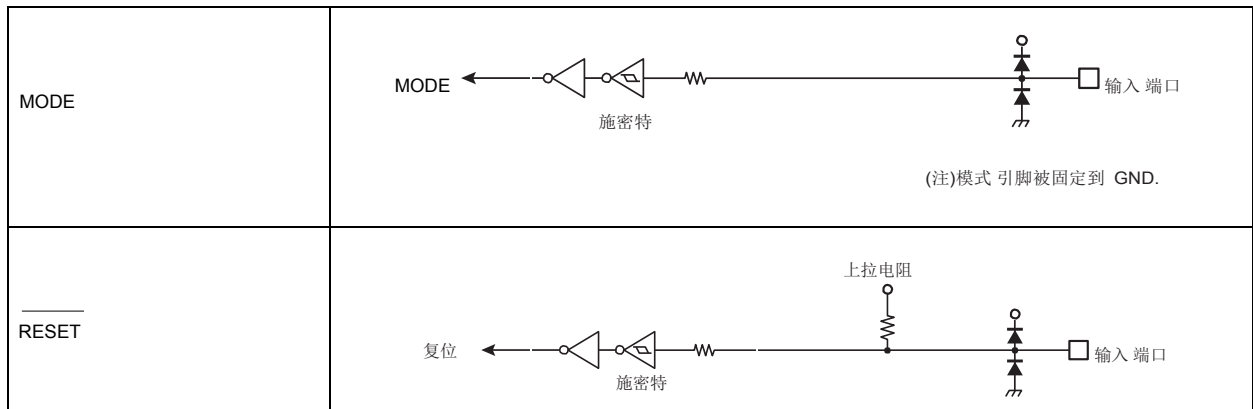
26.1 端口



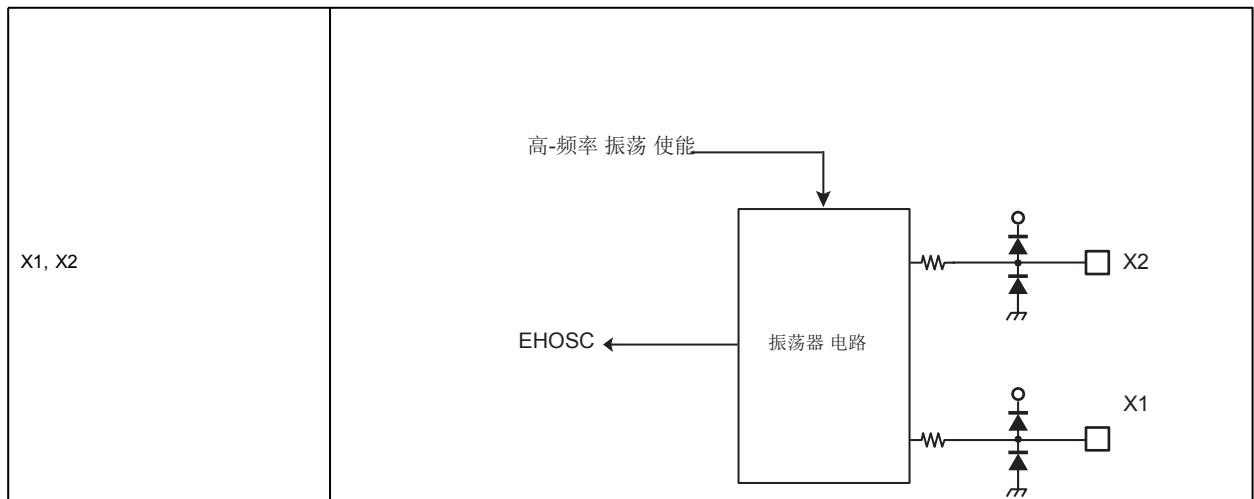
26.2 模拟 端子



26.3 控制 端子



26.4 时钟 端子



27. 电气 特征

27.1 绝对 最大 额定值

参数		符号	额定值	单位
电源 电压		DVDD5	-0.3 ~ 6	V
		RVDD5	-0.3 ~ 6	
		AVDD5A/B	-0.3 ~ 6	
电容 电压		VOUT12	-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
	总	ΣI_{OL}	50	
高-电平 输出 电流	每 引脚	I_{OH}	-5	
	总	ΣI_{OH}	-50	
功耗	$T_a = 85\text{ }^\circ\text{C}$	PD	600	mW
焊接 温度 (10 s)		T_{SOLDER}	260	$^\circ\text{C}$
储存 温度		T_{STG}	-55 ~ 125	$^\circ\text{C}$
工作 温度		T_{OPR}	-40 ~ 85	$^\circ\text{C}$

注 1: 绝对最大额定值为在最坏可能条件下不应该被超越的工作和环境条件极限值。设备制造商应该为了不超越与电流, 电压, 功耗, 温度, 等等有关的绝对最大额定值做好设计。以上那些表揭示的条件外可能 导致设备的永久损坏或影响设备稳定性, 它会增加个人受伤的潜在风险由于 IC 爆炸和/或燃烧。

注 2: VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B.

27.2 DC 电气特征 (1/2)

DVSS = AVSSA = AVSSB = 0V, Ta = -40 ~ 85 °C

参数		符号	额定值	Min	Typ. (注 1)	Max	单位
电源电压 (注 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f _{osc} = 8 ~ 10 MHz f _{sys} = 1 ~ 120 MHz	4.5	-	5.5	V
电源电压 (Flash W/E 期间) (注 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f _{osc} = 8 ~ 10 MHz f _{sys} = 1 ~ 120 MHz (Ta = 0 ~ 70 °C)	4.5	-	5.5	V
电源电压 (上电或断电) (注 3)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f _{osc} = 8 ~ 10 MHz f _{sys} = 1 ~ 120 MHz	3.9	-	5.5	V
低-电平输入电压	施密特输入	V _{IL1}	VDD = 4.5 ~ 5.5V (注 4)	-0.3	-	0.25 VDD	V
高-电平输入电压	施密特输入	V _{IH1}	VDD = 4.5 ~ 5.5V (注 4)	0.75 VDD	-	VDD+0.3	V
电容 (注 5)		C _{out}	RVDD5 = 4.5 ~ 5.5V VOUT12, VOUT3	3.3	-	4.7	μF
低-电平输出电压		V _{OL}	I _{OL} = 1.6 mA VDD ≥ 4.5V (注 4)	-	-	0.4	V
高-电平输出电压		V _{OH}	I _{OH} = -1.6 mA VDD ≥ 4.5V (注 4)	4.1	-	-	V
输入漏电流		I _{LI}	0.0 ≤ V _{IN} ≤ VDD (注 4)	-	0.02	±5	μA
输出漏电流		I _{LO}	0.2 ≤ V _{IN} ≤ VDD-0.2 (注 4)	-	0.05	±10	
复位时上拉寄存器		R _{RST}	4.5 ≤ VDD ≤ 5.5 (注 4)	-	50	150	kΩ
程序可编程上拉/下拉电阻		P _{KH}	4.5 ≤ VDD ≤ 5.5 (注 4)	-	50	150	kΩ
施密特-触发 端口		V _{TH}	4.5 ≤ VDD ≤ 5.5 (注 4)	0.3	0.6	-	V
引脚 电容 (除电源 引脚)		C _{IO}	f _c = 1 MHz	-	-	10	pF

注 1: Ta = 25 °C, DVDD5 = AVDD5A = AVDD5B = RVDD5 = 5V, 除非注明.

注 2: 必须提供相同电压到 DVDD5, AVDD5A, AVDD5B 和 RVDD5.

注 3: 为上电或断电时 (VLTD 禁止时) 电压范围. 电源线为
3.9V ≤ VDD < 4.5V 的范围中, 不保证 12-位 A/D 转换器和 AC 电气特征.

注 4: VDD = DVDD5 = AVDD5A = AVDD5B = RVDD5

注 5: 通过相同电容值 VOUT12 和 VOUT3 引脚应该被连接到 GND. IC 外侧不可以有来自 VOUT12 和 VOUT3 的电源.

27.3 DC 电气 特征 (2/2)

DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5V ~ 5.5V, Ta = -40 ~ 85 °C

参数	符	条件		Min	Typ. (注)	Max	单位
		系统 时钟 (fsys)	工作条件				
NORMAL	IDD	120MHz	参考表 27-1 表 27-2 关于工作 条件	-	45	100	mA
IDLE				-	13	50	mA
NORMAL		80MHz		-	34	75	mA
IDLE				-	9.8	38	mA
STOP		-		-	1.6	25	mA

注 1: Ta = 25 °C, DVDD5 = AVDDA5 = AVDD5B = RVDD5 = 5V, 除非注明.

注 2: 80MHz 时 IDD 为参考值.

表 27-1 IDD 测量条件 (引脚 条件, 振荡器)

		NORMAL	IDLE	STOP
引脚 条件	DVDD5 = AVDD5A = AVDD5B = RVDD5	4.5 ~ 5.5 V		
	X1, X2 引脚	被连接到振荡器		
	输入 引脚	被固定		
	输出 引脚	开路		
工作 条件 (振荡器)	外部 高速 振荡器 (EHOSC)	使能		禁止
	内部 高速 振荡器 (IHOSC)	禁止		
	fsys 的 PLL	使能 (x12)		禁止

表 27-2 IDD 测量条件 (CPU, 外设 电路)

电路	配备 电路数	NORMAL	IDLE	STOP
CPU	1	使能 Dhrystone 版本 2.1	禁止	禁止
A-VE	2	使能	禁止	禁止
µDMAC	1	禁止	禁止	禁止
ADC	2	使能	禁止	禁止
TMRB	10	使能	禁止	禁止
WDT	1	禁止	禁止	禁止
SIO/UART	4	使能	禁止	禁止
I2C	1	禁止	禁止	禁止
PMD	2	使能	禁止	禁止
A-ENC	2	使能	禁止	禁止
I/O 端口	-	使能	禁止	禁止
VLTD	1	使能	使能	禁止
OFD	1	使能	禁止	禁止
CAN	1	禁止	禁止	禁止

27.4 12-位 ADC 电气特征

DVDD5 = RVDD5 = AVDD5A / VREFHA = AVDD5B / VREFHB = 4.5 V ~ 5.5 V

DVSS = AVSSA / VREFLA = AVSSB / VREFLB = 0V, Ta = -40 ~ 85 °C

参数	符号	额定值	Min	Typ.	Max	单位
模拟参考电压 (+)	VREFHA VREFHB	-	-	AVDD	-	V
模拟输入电压	VAIN	-	AVSS	-	AVDD	V
模拟参考电压的电源电流 (注 1)	IREF	-	-	3.5	5.0	mA
供电电流 (注 1) A/D 转换	-	除了 IREF	-	-	6.0	mA
INL 误差	-	AIN 电阻 $\leq 600 \Omega$ AIN 负载 电容 $\geq 0.1 \mu\text{F}$ 转换时间 $\geq 1.00 \mu\text{s}$ 转换时钟 频率 = 120MHz (注 4)	-	-	± 6	LSB
DNL 误差			-	-	± 5	
偏移 误差			-	-	± 5	
全规模 误差			-	-	± 6	
总 误差			-	-	-10 ~ +6	

注 1: 在仅 ADC 工作时的条件下测量电流.

注 2: $1\text{LSB} = (\text{AVDD} - \text{AVSS}) / 4096 [\text{V}]$ 注 3: $\text{AVDD} = \text{AVDD5A} = \text{AVDD5B}$, $\text{AVSS} = \text{AVSSA} = \text{AVSSB}$

注 4: 在 2 单元 ADC 工作时的条件下测量特征.

27.5 AC 电气 特征

27.5.1 AC 测量 条件

AC 测量条件

- 输出 电平: 高 = $0.8 \times DVDD5$ / 低 = $0.2 \times DVDD5$
- 输入 电平: 参考 DC 电气 特征中低-电平 输入电压和高-电平输入电压 .
- 负载 电容:CL=30pF

注:VDD = DVDD5 = AVDD5A = AVDD5B

27.5.2 串行 通道 时序 (SIO/UART)

27.5.2.1 I/O 接口 模式 (VDD = 4.5 ~ 5.5V)

下表中, 字母x 代表系统时钟 (fsys)周期.取决于时钟齿轮功能的编程而变化.

(1) SCLK 输入 模式 (Ta = -40 ~ 85°C)

[数据 输入]

参数	符号	等式		80 MHz		120 MHz		单位
		Min	Max	Min	Max	Min	Max	
SCLK 时钟 高 宽 (输入)	t_{SCH}	4x	-	50	-	33.3	-	ns
SCLK 时钟 低 宽 (输入)	t_{SCL}	4x	-	50	-	33.3	-	
SCLK 周期	t_{SCY}	8x	-	100	-	66.6	-	
有效 数据 输入 ← SCLK 上升或下降(注 1)	t_{SRD}	30	-	30	-	30	-	
SCLK 上升或下降→ 输入 数据 保持 (注 1)	t_{HSR}	x + 30	-	42.5	-	38.3	-	

[数据 输出]

参数	符	等式		80 MHz		120M Hz		单位
		Min	Max	Min	Max	Min	Max	
SCLK 时钟 高 宽 (输入)	t_{SCH}	4x	-	82.5 (注 3)	-	70.0 (注 3)	-	ns
SCLK 时钟 低 宽 (输入)	t_{SCL}	4x	-	82.5 (注 3)	-	70.0 (注 3)	-	
SCLK 周期	t_{SCY}	8x	-	165	-	140	-	
输出 数据 ← SCLK 上升或下降(注 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$ (注 2)	-	0 (注 2)	-	0 (注 2)	-	
SCLK 上升或下降→ 输出 数据 保持 (注 1)	t_{OHS}	$t_{SCY}/2$	-	82.5	-	70.0	-	

注 1: SCLK 上升或下降: SCLK 上升模式时, 时序为 SCLK 上升. SCLK 下降模式时, 时序为 SCLK 上升.

注 2: 计算的值应该使用它没有被减的范围的 SCLK 周期.

注 3: t_{OSS} 表示没有被减的最小值. 值不被计算.

(2) SCLK 输出 模式 ($T_a = -40 \sim 85^\circ\text{C}$)

[数据 输入 / 输出]

参数	符	等式		80 ~ 120 MHz		单元
		Min	Max	Min	Max	
SCLK 周期 (程序可编程) (注 3)	t_{SCY}	4x	-	50	-	ns
输出 数据 ← SCLK 上升或下降	t_{OSS}	$t_{SCY}/2 - 25$ (注 1)	-	0 (注 2)	-	
SCLK 上升或下降 → 输出 数据 保持	t_{OHS}	$t_{SCY}/2 - 25$ (注 1)	-	0 (注 2)	-	
有效 数据 输入 ← SCLK 上升或下降	t_{SRD}	45	-	45	-	
SCLK 上升或下降 → 输入 数据 保持	t_{HSR}	0	-	0	-	

注 1: 计算的值应该使用它没有被减的范围的 SCLK 周期.

注 2: 表示没有被减的最小值. 值不被计算.

注 3: 请调节 SCLK 周期为 $t_{SCY} \geq 50\text{ns}$.

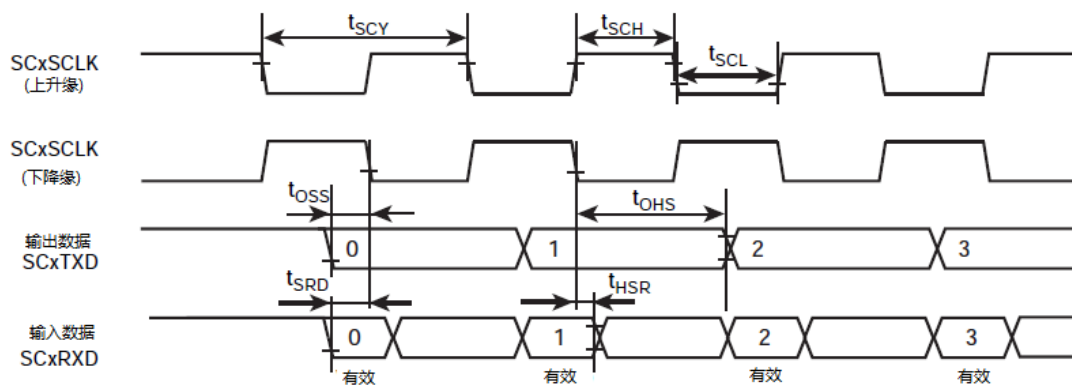


图 27-1 串行通道时序(SIO)

27.5.3 串行总线接口 (I2C)

27.5.3.1 I2C 模式

下表中, 字母x 代表对fsys 周期时间来说理想的 I2C 工作时钟周期时间.取决于时钟齿轮功能的编程而变化.

n 指 SBIxCR中编程到 SCK (SCL 输出 频率 选择) 域的值.

参数	符号	等式		标准 模式		快 模式		单元
		Min	Max	Min	Max	Min	Max	
SCL 时钟 频率	t _{SCL}	0	-	0	100	0	400	kHz
START 条件的保持时间	t _{HD; STA}	-	-	4.0	-	0.6	-	μs
SCL 低 宽 (输入) (注 1)	t _{LOW}	-	-	4.7	-	1.3	-	μs
SCL 高 宽 (输入) (注 2)	t _{HIGH}	-	-	4.0	-	0.6	-	μs
重复的 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 条件设置时间	t _{SU; STO}	-	-	4.0	-	0.6	-	μs
停止条件和开始条件之间总线 自由时间	t _{BUF}	(注 5)	-	4.7	-	1.3	-	μs

注 1: SCL 时钟 低 宽 (输出) = $(2^{n-1} + 58)/x$

注 2: SCL 时钟 高 宽 (输出) = $(2^{n-1} + 14)/x$

I2C-总线规格上, 标准模式最大速度为 100kHz, 快模式为 400kHz. 内部 SCL 频率设置应该对应以上所示注 1 & 注 2.

注 3: 输出数据保持时间等于 4x 内部 SCL.

注 4: Philips I2C-总线规格阐述设备必须内部提供保持时间至少 300 ns 给 SDA 信号来联系 SCL 下降缘未定义区域. 然而, 此 SBI 不满足此需要. 同时, SCL 的输出缓冲不包含下降缘的斜控制; 因此, 设备制造商应该为了]满足表中所示输入数据保持时间好好设计, 包括 SCL 和 SDA 线的 tr/td.

注 5: 软件 -取决

注 6: Philips I2C-总线 规格 描述如果快-模式设备电源关闭, 为了不妨碍总线线路 SDA 和 SCL I/O 引脚必须悬浮. 然而, 此 SBI 不满足此需要.

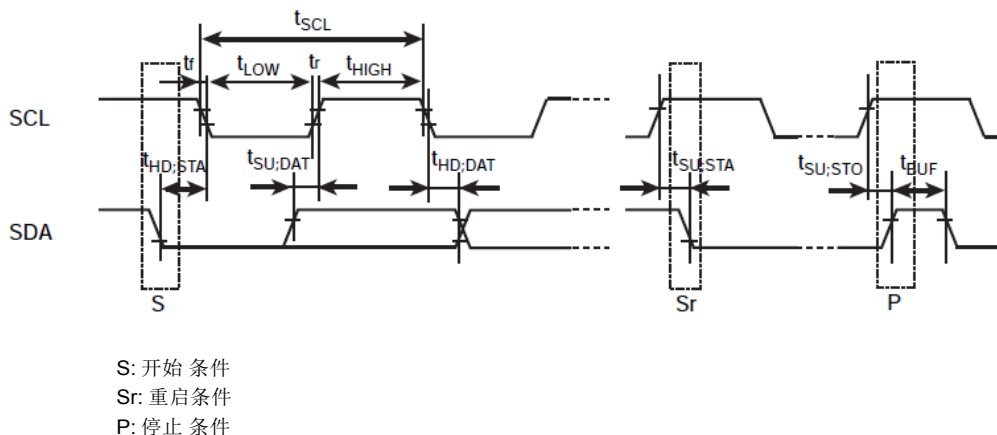


图 27-2 串行总线时序 (I2C)

27.5.4 16-位 定时器 / 事件 计数器 (TMRB)

27.5.4.1 事件 计数器

下表中，字母x 代表对 f_{sys} 周期时间来说理想的 TMRB 工作时钟周期时间.取决于时钟齿轮功能的编程而变化.

参数	符号	等式		40 MHz		120 MHz		单位
		Min	Max	Min	Max	Min	Max	
时钟 低脉冲宽	t_{VCKL}	$2x + 100$	-	125	-	117	-	ns
时钟 高脉冲宽	t_{VCKH}	$2x + 100$	-	125	-	117	-	ns

27.5.4.2 捕捉

下表中，字母x 代表对 f_{sys} 周期时间来说理想的 TMRB 工作时钟周期时间.取决于时钟齿轮功能的编程而变化.

参数	符号	等式		80 MHz		120 MHz		单位
		Min	Max	Min	Max	Min	Max	
低脉冲宽	t_{CPL}	$2x + 100$	-	125	-	117	-	ns
高脉冲宽	t_{CPH}	$2x + 100$	-	125	-	117	-	ns

27.5.5 外部 中断

下表中，字母x 代表 f_{sys} 周期时间.

1. 除停止释放中断

参数	符号	等式		80 MHz		120 MHz		单位
		Min	Max	Min	Max	Min	Max	
低脉冲宽	t_{NTAL}	$x + 100$	-	112.5	-	108.3	-	ns
高脉冲宽	t_{NTAH}	$x + 100$	-	112.5	-	108.3	-	ns

2. 停止释放中断

参数	符号	等式		80 MHz		120 MHz		单位
		Min	Max	Min	Max	Min	Max	
低脉冲宽	t_{NTAL}	100	-	100	-	100	-	ns
高脉冲宽	t_{NTAH}	100	-	100	-	100	-	ns

27.5.6 调试 通信

27.5.6.1 AC 测量 条件

- 输出 电平: 高 = $0.7 \times DVDD5$ / 低 = $0.3 \times DVDD5$
- 负载 电容 : TRACECLK CL=25pF, TRACEDATA CL=20pF

27.5.6.2 SWD 接口

参数	符号	Min	Max	单位
CLK 周期	T_{dck}	100	-	ns
CLK 上升后数据 保持	T_{d1}	4	-	
CLK 上升后数据 有效	T_{d2}	-	30	
CLK 上升的数据 有效	T_{ds}	20	-	
CLK 下降后数据 保持	T_{dh}	15	-	

27.5.6.3 JTAG 接口

参数	符号	Min	Max	单位
CLK 周期	T_{dck}	100	-	ns
CLK 上升后数据 保持	T_{d3}	4	-	
CLK 上升后数据 有效	T_{d4}	-	50	
CLK 上升的数据 有效	T_{ds}	20	-	
CLK 下降后数据 保持	T_{dh}	15	-	

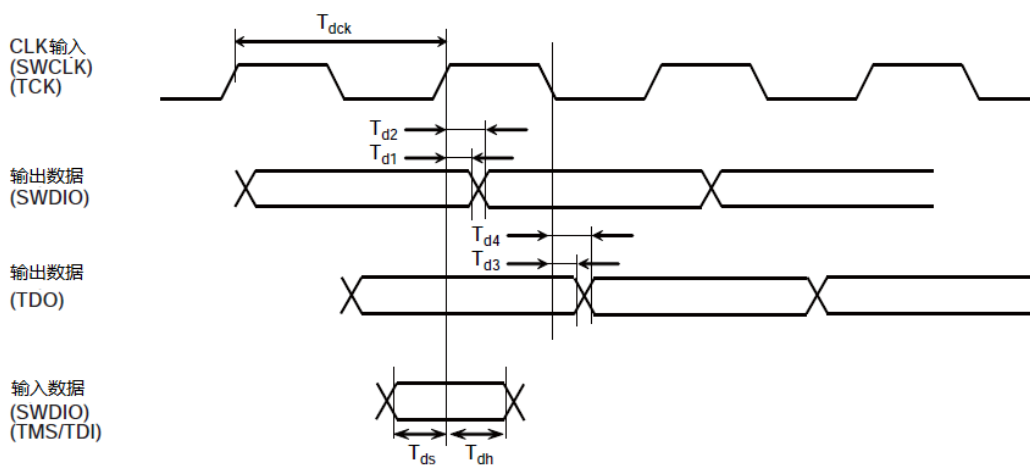


图 27-3 SWD / JTAG 通信 时序

27.5.7 EMT 跟踪

AC 测量条件如下.

- 输出电平: 高 = $0.7 \times DVDD5$ / 低 = $0.3 \times DVDD5$
- 负载电容: TRACECLK CL=25pF, TRACEDATA CL=20pF

参数	符号	Min	Max	单位
TRACECLK 周期	t_{clk}	33.3	-	ns
TRACEDATA 有效 ← TRACECLK 上升	t_{setupr}	2	-	
TRACECLK 上升 → TRACEDATA 保持	t_{holdr}	1	-	
TRACEDATA 有效 ← TRACECLK 下降	t_{setupf}	2	-	
TRACECLK 下降 → TRACEDATA 保持	t_{holdf}	1	-	

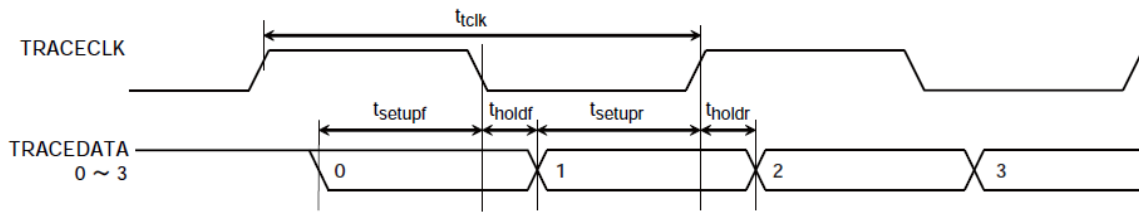


图 27-4 EMT 跟踪 时序

27.5.8 闪存 特征

参数	额定值	Min	Typ.	Max	单位
关于 Flash-内存重写的保证	Ta = -40 ~ 85°C DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5 ~ 5.5	-	-	1000	次

27.5.9 芯片上振荡器

参数	符号	额定值	Min	Typ.	Max	单位
振荡 频率 (注 1)	fosc	Ta = -40 ~ 85°C	-	-	10	MHz
fosc 精度 (注 1)(注 2)	-	Ta = 0 ~ 50°C	-	-	± 1	%
		Ta = -40 ~ 0°C, Ta = 50 ~ 85°C	-	-	± 2	%

注 1: 此为出厂测试值. 如果需要描述的振荡精度, 回流焊直后或稍后必须微调振荡频率不超越 120 MHz (PLL 设置为 12-折叠) 最大频率.

注 2: 此指 DVDD5 = RVDD5 = AVDD5A = AVDD5B = 5.0V 时微调的振荡频率的误差, 且 Ta=25°C.

27.5.10 外部 振荡器

参数	符号	额定值	Min	Typ.	Max	单位
高 频率 振荡	feosc	Ta = -40~ 85°C	-	10	-	MHz

27.6 振荡 电路

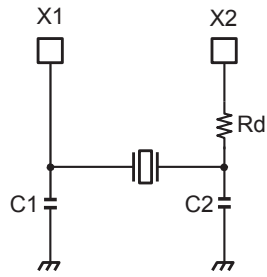


图 27-5 高-频率 振荡 连接

注:要获取稳定振荡, 负载电容和振荡器位置必须正确配置。
因为这些因素受底层图案严重影响, 请使用你使用的底层评估振荡稳定性。

27.6.1 陶瓷 振荡器

TMPM475FDFG/FZFG/FYFG 推荐 Murata Manufacturing Co., Ltd 公司的高-频率振荡器。

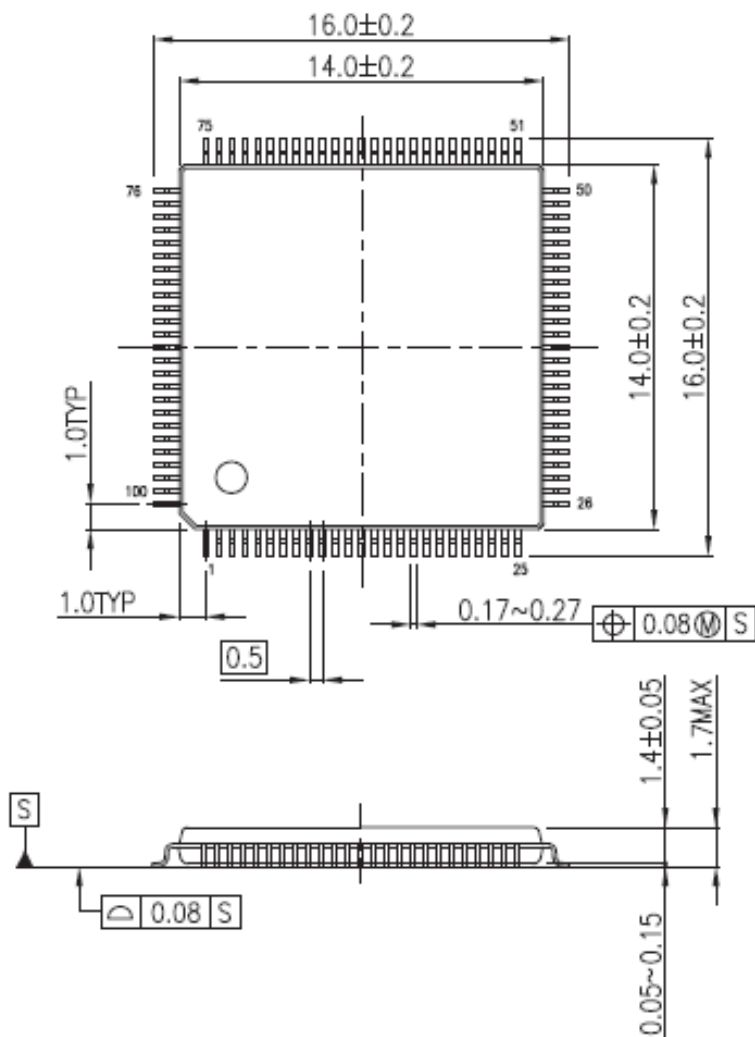
请参考以下 URL 的详细. <http://www.murata.co.jp/>

28. 封装尺寸

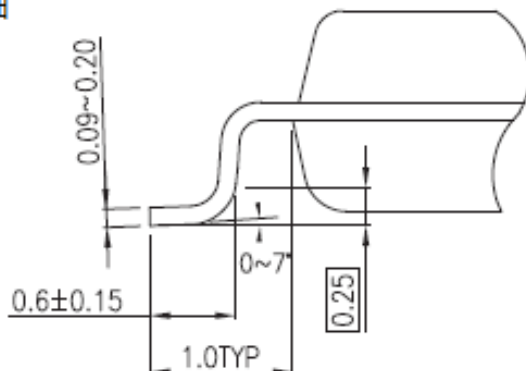
类型: P-LQFP100-1414-0.50-002

单位: mm

尺寸



引脚详细



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.