TOSHIBA

译文

TMPM370FYDFG / TMPM370FYFG

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: “TMPM370FYDFG / TMPM370FYFG”
2013-04-15

翻译日: 2014-07-08

TOSHIBA CORPORATION

译文

TOSHIBA

32 位 RISC 微控制器

TX03 系列

TMPM370FYDFG / TMPM370FYFG

东芝公司
半导体&存储产品公司

译文

译文

修订记录

日期	版次	说明
2010/10/8	1.0	首次发布
2011/3/7	2.0	对内容进行了修订
2013/4/15	3.0	对内容进行了修订

译文

译文

译文

TMPM370FYDFG/FYFG

TMPM370FYDFG/FYFG 是配备 ARM Cortex™-M3 微处理器内核的 32-位 RISC 微处理器系列。

产品名称	ROM (FLASH)	RAM	封装
TMPM370FYDFG	256 K 字节	10 K 字节	P-QFP100-1420-0.65Q
TMPM370FYFG	256 K 字节	10 K 字节	P-LQFP100-1414-0.50H

TMPM370FYDFG/FYFG 的特征如下：

1.1 特征

1. ARM Cortex-M3 微处理器内核

   a. 通过使用 Thumb®-2 指令，改善了编码效率。

   - 全新的 16-位 Thumb 指令改善了程序流
   - 全新的 32-位 Thumb 指令提高了性能
   - 全新的 Thumb 混合 16-/32-位指令集可生成更快，更高效的代码

   b. 同时实现了高性能与低功耗。

   [高性能]

   - 一次时钟可执行 32-位乘法(32 × 32 = 32 位)
   - 根据被除数与除数，在 2 个与 12 个周期之间生成除法

   [低功耗]

   - 采用低功耗库的优化设计
   - 可停止微控制器内核运行的待机功能

   c. 适合于实时控制的高速度中断响应

   - 可中断的长指令
   - 硬件可自动处理堆栈

2. 片内程序存储器与数据存储器

   - 片内 RAM: 10 K 字节
   - 片内 FlashROM: 256 K 字节

3. 16-位计时器 (TMRB): 8 通道

   - 16-位间隔计时器模式
   - 16-位事件计数器模式
   - 输入捕捉功能
   - 外触发器 PPG 输出

4. 看门狗计时器(WDT)：1 通道

看门狗计时器(WDT) 可生成复位或非可屏蔽中断 (NMI)。

5. 上电复位功能(POR)

6. 电压检测功能(VLTD)

7. 振荡频率检测功能(OFD)

8. 矢量引擎(VE)：1 单元
   - 电机控制用计算电路
   - 对应 2 台电机

9. 可编程电机驱动器(PMD)：2 通道
   - 3 相互补 PWM 发生器
   - 同步 AD 转换起动触发器发生器
   - 应急防护功能(EMG / 比较器输出)

10. 编码器输入电路(ENC)：2 通道
    - 对应增量编码器(AB / ABZ)
    - 旋转方向检测
    - 绝对位置检测用计数器
    - 位置检测用比较器
    - 噪声滤波器
    - 3 相传感器输入

11. 通用串行接口(SIO/UART)：4 通道
    可选择 UART 模式或同步模式(配备 4 字节 FIFO)

12. 12 位 AD 转换器(ADC)：2 单元 (模拟输入：22 通道)
    - 从内部触发开始：TMRB 中断 / PMD 触发
    - 恒定转换方式
    - AD 监控 2ch
    - 转换速度 2 μsec (@ADC 转换时钟 = 40 MHz)

13. OP-Amp(AMP)：4 通道
    可选择 8 增益

14. 比较器(CMP)：3+1 通道
    - 电机紧急停止用保护
    - 2 种输入类型(OP-Amp 输出/模拟输入)

15. 输入/输出端 (PORT)：76 引脚
    I/O 引脚：74 引脚
    输入引脚：2 引脚

16. 中断源

- 内部 62 个因数：可按 7 种级别设置优先顺序(看门狗计时器中断除外)
- 外部 16 个因数：可按 7 种级别设置优先顺序

17. 待机模式

待机模式：IDLE, STOP

18. 时钟发生器(CG)

- 片内 PLL (8 倍)
- 时钟齿轮功能：该高速时钟可被分成 1/1, 1/2, 1/4, 1/8 或 1/16。

19. 字节存储次序

低端在前格式

20. 最高工作频率：80 MHz

21. 工作电压范围

4.5 V ~ 5.5 V (带片内调压器)

22. 温度范围

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

23. 封装

- P-QFP100-1420-0.65Q (14 mm × 20 mm, 0.65 mm 间距)
- P-LQFP100-1414-0.50H (14 mm × 14 mm, 0.5 mm 间距)

1.2 方块图

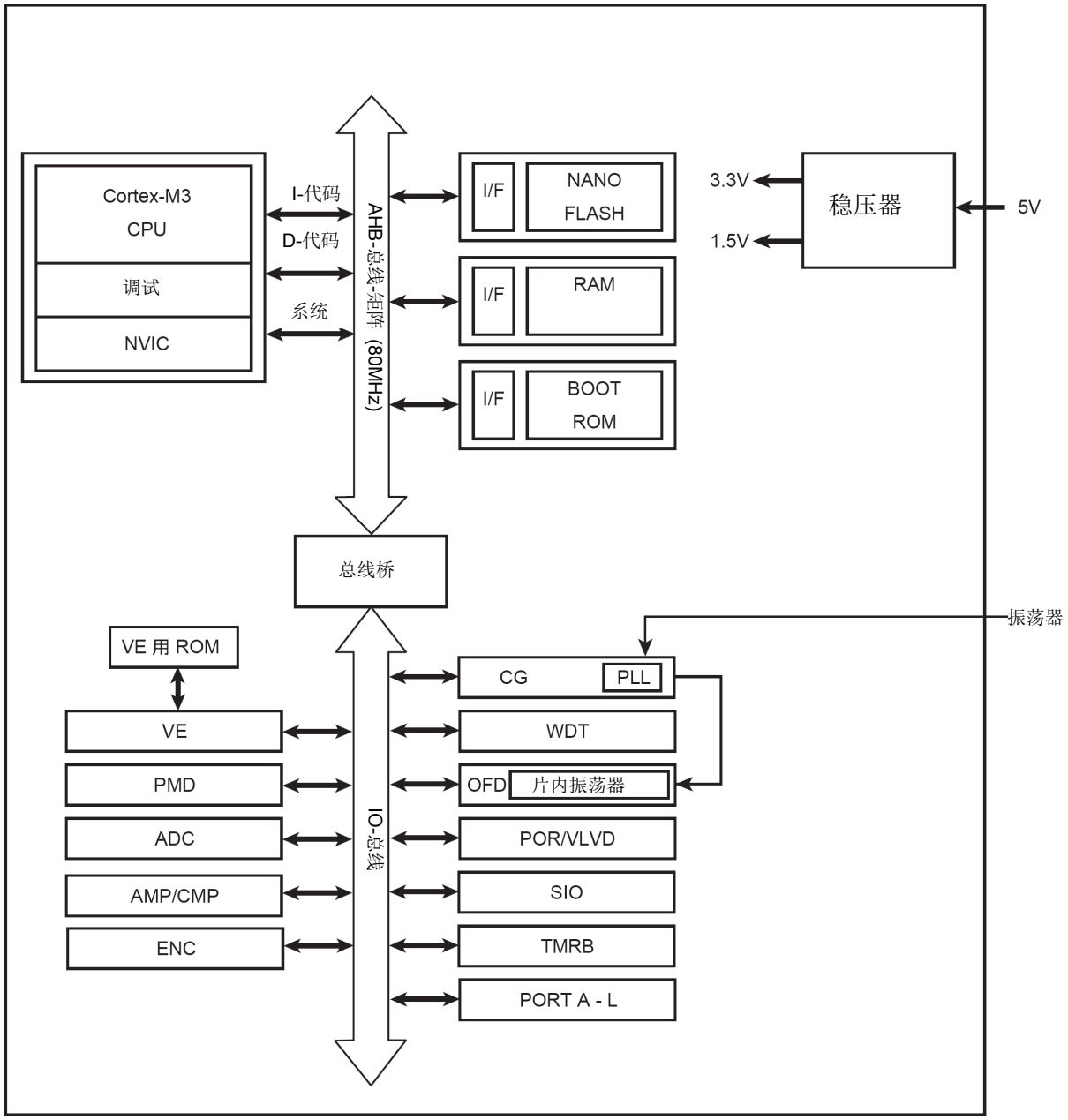


图 1-1 TMPM370FYDFG/FYFG 方块图

1.3 引脚布局(顶视图)

TMPM370FYDFG/FYFG 的引脚布局如下图所示。

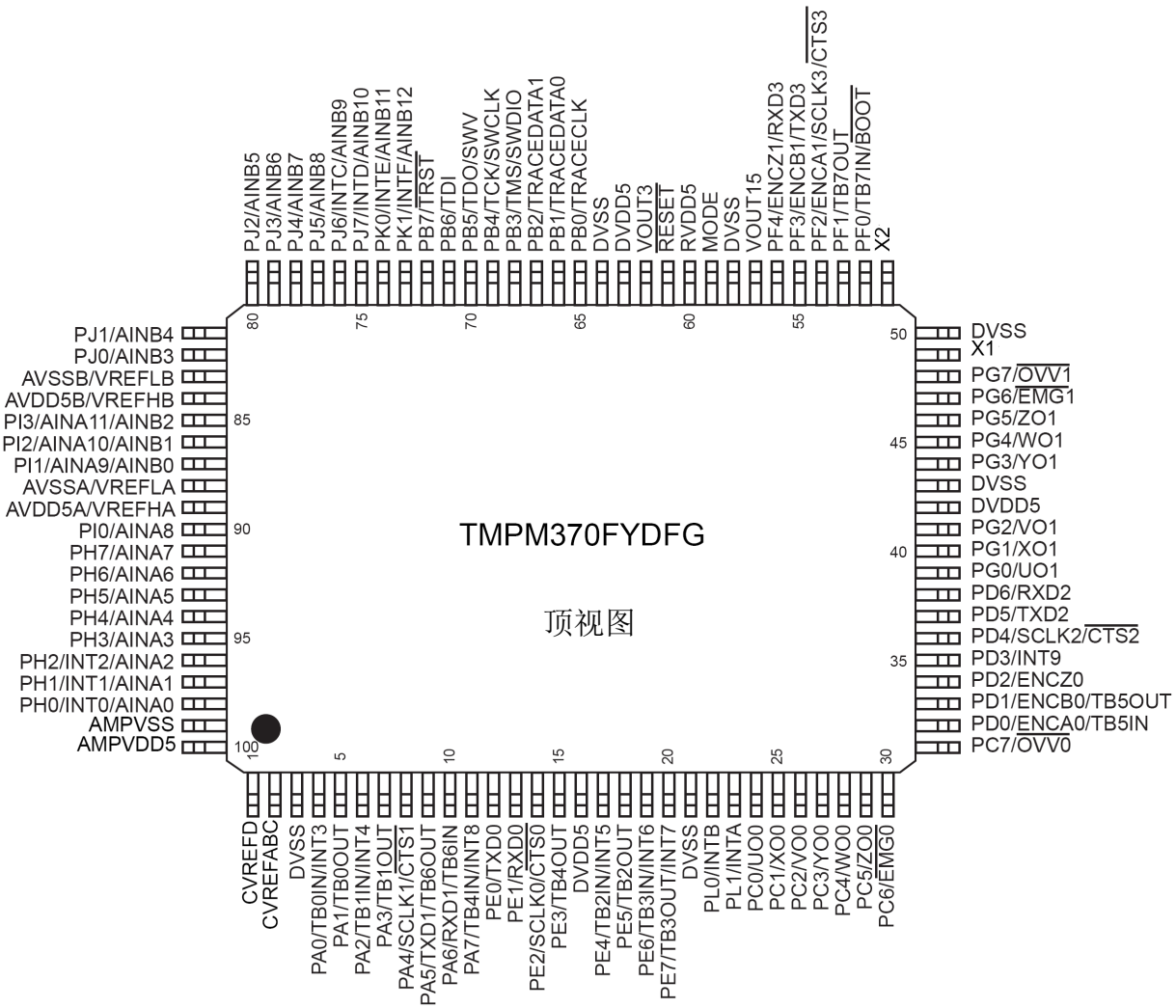


图 1-2 引脚布局 (QFP100)

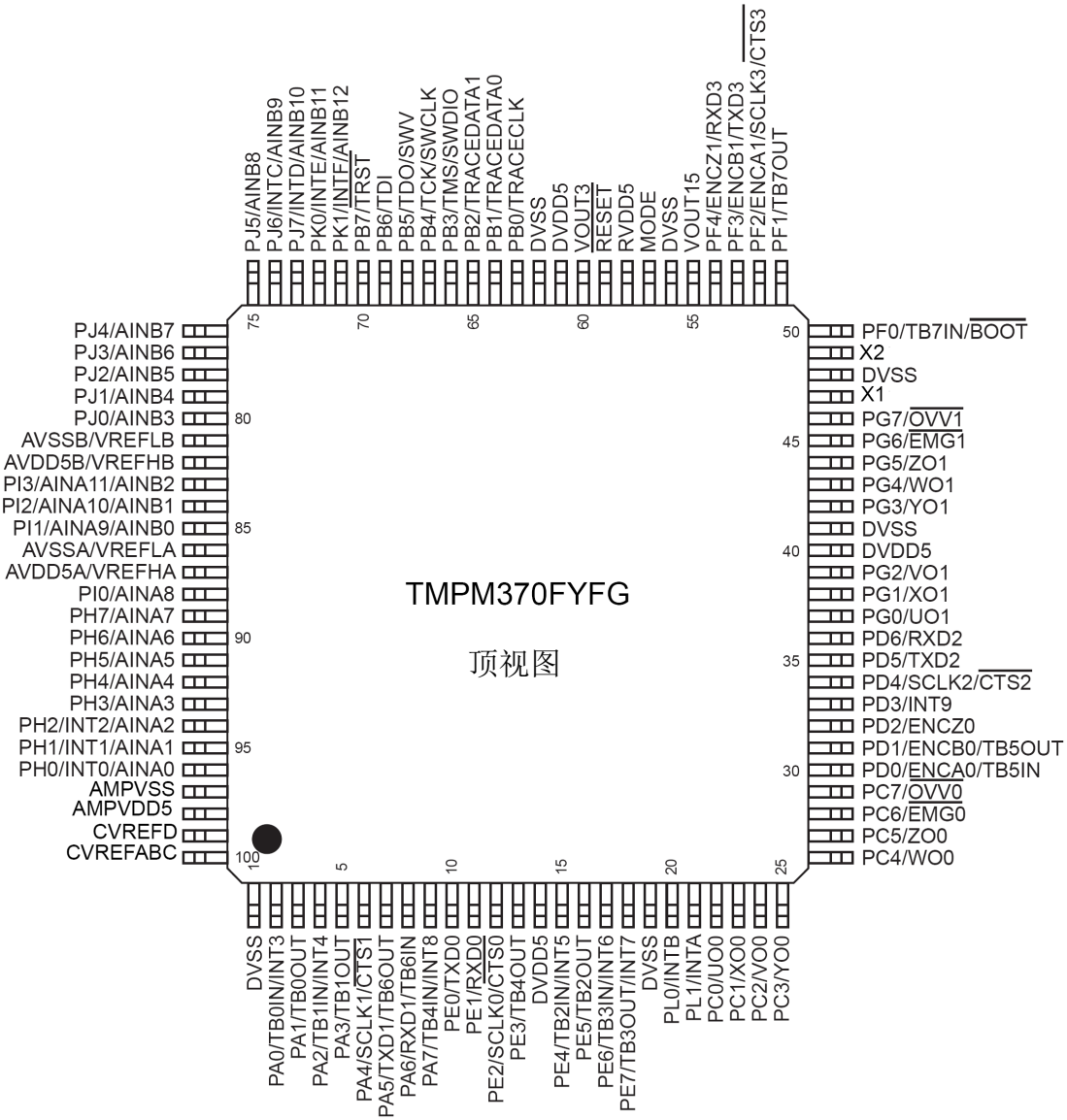


图 1-3 引脚布局(LQFP100)

1.4 引脚名称与功能

在表 1-1 中，按引脚或端口对 TMPM370FYDFG/FYFG 的输入与输出引脚进行分类。各表均给出了各多功能引脚的替代引脚名称与功能。

1.4.1 按端口分类

表 1-1 按端口分类的引脚名称与功能(1/5)

PORT	型号	引脚编号 (DFG/ FG)	引脚名称	输入/输出	功能
PORT A	功能	4 / 2	PA0 TB0IN INT3	I/O I I	I/O 端口 输入计时器 B 捕捉触发脉冲 外部中断引脚
PORT A	功能	5 / 3	PA1 TB0OUT	I/O O	I/O 端口 计时器 B 输出
PORT A	功能	6 / 4	PA2 TB1IN INT4	I/O I I	I/O 端口 输入计时器 B 捕捉触发脉冲 外部中断引脚
PORT A	功能	7 / 5	PA3 TB1OUT	I/O O	I/O 端口 计时器 B 输出
PORT A	功能	8 / 6	PA4 SCLK1 $\overline{\text{CTS1}}$	I/O I/O I	I/O 端口 串行时钟输入/输出 握手输入引脚
PORT A	功能	9 / 7	PA5 TXD1 TB6OUT	I/O O O	I/O 端口 发送串行数据 计时器 B 输出
PORT A	功能	10 / 8	PA6 RXD1 TB6IN	I/O I I	I/O 端口 接收串行数据 输入计时器 B 捕捉触发脉冲
PORT A	功能	11 / 9	PA7 TB4IN INT8	I/O I I	I/O 端口 输入计时器 B 捕捉触发脉冲 外部中断引脚
PORT B	功能/调试	65 / 63	PB0 TRACECLK	I/O O	I/O 端口 调试引脚
PORT B	功能/调试	66 / 64	PB1 TRACEDATA0	I/O O	I/O 端口 调试引脚
PORT B	功能/调试	67 / 65	PB2 TRACEDATA1	I/O O	I/O 端口 调试引脚
PORT B	功能/调试	68 / 66	PB3 TMS / SWDIO	I/O I/O	I/O 端口 调试引脚
PORT B	功能/调试	69 / 67	PB4 TCK / SWCLK	I/O I	I/O 端口 调试引脚
PORT B	功能/调试	70 / 68	PB5 TDO / SWV	I/O O	I/O 端口 调试引脚
PORT B	功能/调试	71 / 69	PB6 TDI	I/O I	I/O 端口 调试引脚
PORT B	功能/调试	72 / 70	PB7 $\overline{\text{TRST}}$	I/O I	I/O 端口 调试引脚
PORT C	功能	24 / 22	PC0 UO0	I/O O	I/O 端口 U-相输出引脚

表 1-1 按端口分类的引脚名称与功能(2/5)

PORT	型号	引脚编号 (DFG/ FG)	引脚名称	输入/输出	功能
PORT C	功能	25 / 23	PC1 XO0	I/O O	I/O 端口 X-相输出引脚
PORT C	功能	26 / 24	PC2 VO0	I/O O	I/O 端口 V-相输出引脚
PORT C	功能	27 / 25	PC3 YO0	I/O O	I/O 端口 Y-相输出引脚
PORT C	功能	28 / 26	PC4 WO0	I/O O	I/O 端口 W-相输出引脚
PORT C	功能	29 / 27	PC5 ZO0	I/O O	I/O 端口 Z-相输出引脚
PORT C	功能	30 / 28	PC6 $\overline{\text{EMG0}}$	I/O ^I	I/O 端口 应急状态检测输入
PORT C	功能	31 / 29	PC7 $\overline{\text{OVV0}}$	I/O I	I/O 端口 过电压检测输入
PORT D	功能	32 / 30	PD0 ENCA0 TB5IN	I/O I I	I/O 端口 A-相输入引脚 输入计时器 B 捕捉触发脉冲
PORT D	功能	33 / 31	PD1 ENCB0 TB5OUT	I/O I O	I/O 端口 B-相输入引脚 计时器 B 输出
PORT D	功能	34 / 32	PD2 ENCZ0	I/O I	I/O 端口 Z-相输入引脚
PORT D	功能	35 / 33	PD3 INT9	I/O I	I/O 端口 外部中断引脚
PORT D	功能	36 / 34	PD4 SCLK2 $\overline{\text{CTS2}}$	I/O I/O I	I/O 端口 串行时钟输入/输出 握手输入引脚
PORT D	功能	37 / 35	PD5 TXD2	I/O O	I/O 端口 发送串行数据
PORT D	功能	38 / 36	PD6 RXD2	I/O I	I/O 端口 接收串行数据
PORT E	功能	12 / 10	PE0 TXD0	I/O O	I/O 端口 发送串行数据
PORT E	功能	13 / 11	PE1 RXD0	I/O I	I/O 端口 接收串行数据
PORT E	功能	14 / 12	PE2 SCLK0 $\overline{\text{CTS0}}$	I/O I/O I	I/O 端口 串行时钟输入/输出 握手输入引脚
PORT E	功能	15 / 13	PE3 TB4OUT	I/O O	I/O 端口 计时器 B 输出
PORT E	功能	17 / 15	PE4 TB2IN INT5	I/O I I	I/O 端口 输入计时器 B 捕捉触发脉冲 外部中断引脚
PORT E	功能	18 / 16	PE5 TB2OUT	I/O O	I/O 端口 计时器 B 输出
PORT E	功能	19 / 17	PE6 TB3IN INT6	I/O I I	I/O 端口 输入计时器 B 捕捉触发脉冲 外部中断引脚

表 1-1 按端口分类的引脚名称与功能(3/5)

PORT	型号	引脚编号 (DFG/ FG)	引脚名称	输入/输出	功能
PORT E	功能	20 / 18	PE7 TB3OUT INT7	I/O O I	I/O 端口 计时器 B 输出 外部中断引脚
PORT F	功能/控制	52 / 50	PF0 TB7IN $\overline{\text{BOOT}}$	I/O I I	I/O 端口 输入计时器 B 捕捉触发脉冲 BOOT 模式引脚。注)该引脚可在 RESET 信号上升时进入单一引导模式"低"电平。
PORT F	功能	53 / 51	PF1 TB7OUT	I/O O	I/O 端口 计时器 B 输出
PORT F	功能	54 / 52	PF2 ENCA1 SCLK3 $\overline{\text{CTS3}}$	I/O I I/O I	I/O 端口 编码器输入 串行时钟输入/输出 握手输入引脚
PORT F	功能	55 / 53	PF3 ENCB1 TXD3	I/O I O	I/O 端口 编码器输入 发送串行数据
PORT F	功能	56 / 54	PF4 ENCZ1 RXD3	I/O I I	I/O 端口 编码器输入 接收串行数据
PORT G	功能	39 / 37	PG0 UO1	I/O O	I/O 端口 U-相输出引脚
PORT G	功能	40 / 38	PG1 XO1	I/O O	I/O 端口 X-相输出引脚
PORT G	功能	41 / 39	PG2 VO1	I/O O	I/O 端口 V-相输出引脚
PORT G	功能	44 / 42	PG3 YO1	I/O O	I/O 端口 Y-相输出引脚
PORT G	功能	45 / 43	PG4 WO1	I/O O	I/O 端口 W-相输出引脚
PORT G	功能	46 / 44	PG5 ZO1	I/O O	I/O 端口 Z-相输出引脚
PORT G	功能	47 / 45	PG6 $\overline{\text{EMG1}}$	I/O I	I/O 端口 应急状态检测输入
PORT G	功能	48 / 46	PG7 $\overline{\text{OVV1}}$	I/O I	I/O 端口 过电压检测输入
PORT H	功能	98 / 96	PH0 INT0 AINA0	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT H	功能	97 / 95	PH1 INT1 AINA1	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT H	功能	96 / 94	PH2 INT2 AINA2	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT H	功能	95 / 93	PH3 AINA3	I/O I	I/O 端口 模拟输入
PORT H	功能	94 / 92	PH4 AINA4	I/O I	I/O 端口 模拟输入

表 1-1 按端口分类的引脚名称与功能(4/5)

PORT	型号	引脚编号 (DFG/ FG)	引脚名称	输入/输出	功能
PORT H	功能	93 / 91	PH5 AINA5	I/O I	I/O 端口 模拟输入
PORT H	功能	92 / 90	PH6 AINA6	I/O I	I/O 端口 模拟输入
PORT H	功能	91 / 89	PH7 AINA7	I/O I	I/O 端口 模拟输入
PORT I	功能	90 / 88	PI0 AINA8	I/O I	I/O 端口 模拟输入
PORT I	功能	87 / 85	PI1 AINA9 / AINB0	I/O I	I/O 端口 模拟输入
PORT I	功能	86 / 84	PI2 AINA10 / AINB1	I/O I	I/O 端口 模拟输入
PORT I	功能	85 / 83	PI3 AINA11 / AINB2	I/O I	I/O 端口 模拟输入
PORT J	功能	82 / 80	PJ0 AINB3	I/O I	I/O 端口 模拟输入
PORT J	功能	81 / 79	PJ1 AINB4	I/O I	I/O 端口 模拟输入
PORT J	功能	80 / 78	PJ2 AINB5	I/O I	I/O 端口 模拟输入
PORT J	功能	79 / 77	PJ3 AINB6	I/O I	I/O 端口 模拟输入
PORT J	功能	78 / 76	PJ4 AINB7	I/O I	I/O 端口 模拟输入
PORT J	功能	77 / 75	PJ5 AINB8	I/O I	I/O 端口 模拟输入
PORT J	功能	76 / 74	PJ6 INTC AINB9	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT J	功能	75 / 73	PJ7 INTD AINB10	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT K	功能	74 / 72	PK0 INTE AINB11	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT K	功能	73 / 71	PK1 INTF AINB12	I/O I I	I/O 端口 外部中断引脚 模拟输入
PORT L	功能	22 / 20	PL0 INTB	I I	输入端口 外部中断引脚
PORT L	功能	23 / 21	PL1 INTA	I I	输入端口 外部中断引脚
-	时钟	49 / 47	X1	I	已连接至高速振荡器
-	时钟	51 / 49	X2	O	已连接至高速振荡器
-	控制		模式	I	模式引脚 (注) 必须将 MODE 引脚连接至 GND。

表 1-1 按端口分类的引脚名称与功能(5/5)

PORT	型号	引脚编号 (DFG/ FG)	引脚名称	输入/输出	功能
-	功能		$\overline{\text{RESET}}$	I	复位输入引脚 (注) 带有一个上拉与一个噪声滤波器(约 30ns (典型值))
		1 / 99	CVREFD		
		2 / 100	CVREFABC		
		99 / 97	AMPVSS		
		100 / 98	AMPVDD5		
-	PS		DVSS	–	GND 引脚
-	PS		DVSS	–	GND 引脚
-	PS	43 / 41	DVSS	–	GND 引脚
-	PS		DVSS	–	GND 引脚
-	PS		DVSS	–	GND 引脚
-	PS		DVSS	–	GND 引脚
-	PS		DVDD5	–	电源引脚
-	PS		DVDD5	–	电源引脚
-	PS		DVDD5	–	电源引脚
-	PS		RVDD5	–	电源引脚
-	PS		VOUT15	–	电源引脚
-	PS		VOUT3	–	电源引脚
-	PS		AVSSB VREFLB	–	AD 转换器：GND 引脚(注 1) 为该 AD 转换器提供基准电源
-	PS		AVDD5B VREFHB	–	为该 AD 转换器提供电源(注 2) 为该 AD 转换器提供基准电源
-	PS		AVSSA VREFLA	–	AD 转换器：GND 引脚(注 1) 为该 AD 转换器提供基准电源
-	PS		AVDD5A VREFHA	–	为该 AD 转换器提供电源(注 2) 为该 AD 转换器提供基准电源

注 1：即使未使用 AD 转换器，也必须将 AVSS 连接至 GND。

注 2：即使未使用 AD 转换器，也必须将其连接至电源。

1.5 引脚编号与电源引脚

表 1-2 引脚编号与电源

电源	电压范围	引脚编号	引脚名称
DVDD5	4.5 ~ 5.5V	, ,	PA, PB, PC, PD, PE, PF, PG, PL, PM PN, $\overline{\text{RESET}}$, MODE
AVDD5A			PH, PI
AVDD5B			PJ
RVDD5			–
VOUT15	1.35 ~ 1.65V		必须通过 3.3 ~ 4.7 μF 电容器将 VOUT15 连接至 DVSS，以向内部电路供电。
VOUT3	2.7 ~ 3.6V		必须通过 3.3 ~ 4.7 μF 电容器将 VOUT3 连接至 DVSS，以向内部电路供电。

注：VOUT15 与 VOUT3 所连接电容器的值必须相同。

2. 处理器内核

该 TX03 系列配备有一个高性能的 32-位处理器内核(ARM Cortex-M3 处理器内核)。有关该处理器内核运行方面的信息，请参阅 ARM 有限公司所发布的《Cortex-M3 技术参考手册》。本章主要对该文件中未述及的 TX03 系列所独有的功能进行说明。

2.1 有关该处理器内核的信息

下表给出了 TMPM370FYDFG/FYFG 中的处理器内核的版次情况。

请参看该 CPU 内核与 CPU 体系结构的详细信息，并请参看以下 URL 中的 ARM 手册《Cortex-M 系列处理器》：

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

产品名称	内核版次
TMPM370FYDFG/FYFG	r2p0

2.2 可配置的选项

Cortex-M3 内核具备可选块。版次 r2p0 的可选块为 ETM™ 与 MPU。下表给出了 TMPM370FYDFG/FYFG 中的可配置选项。

可配置的选项	执行
FPB	两个文字比较器 六个指令比较器
DWT	四个比较器
ITM	可执行
MPU	不可执行
ETM	可执行
AHB-AP	可执行
AHB 跟踪宏单元接口	可执行
TPIU	可执行
WIC	不可执行

2.3 异常/中断

以下将对异常与中断进行说明。

2.3.1 中断输入的数目

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

TMPM370FYDFG/FYFG 具备 78 次中断输入。中断输入的数目可在 NVIC 寄存器的<INTLINESNUM[4:0]>位中反映出来。在该产品中，如果读取<INTLINESNUM[4:0]> 位，即可读出 0x00。

2.3.2 优先级中断位的数目

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

TMPM370FYDFG/FYFG 具备 3 个优先级中断位。优先级中断位的数目，用于指定中断优先权寄存器与系统处理器优先寄存器中的优先级。

2.3.3 SysTick

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

有关 SysTick 异常的详细资料，请参看异常中的"SysTick"一节，以及 NVIC 寄存器中的 SysTick 的寄存器。

2.3.4 SYSRESETREQ

在应用中断与复位控制寄存器"的<SYSRESETREQ>位完成设置时，该 Cortex-M3 内核输出 SYSRESETREQ 信号。

在 SYSRESETREQ 信号输出时，TMPM370FYDFG/FYFG 进行相同的操作。

2.3.5 LOCKUP

在产生无可挽救的异常时，该 Cortex-M3 内核可输出 LOCKUP 信号，以给出软件中所包含的严重错误。

TMPM370FYDFG/FYFG 不使用该信号。必须使用非屏蔽中断指令(NMI)或复位，才能从 LOCKUP 状态返回。

2.3.6 辅助故障状态寄存器

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

不过，TMPM370FYDFG/FYFG 并未被定义该功能。如果读取了辅助故障状态寄存器，则所读出的始终会是"0x0000_0000"。

2.4 事件

该 Cortex-M3 内核具有事件输出信号与事件输入信号。事件输出信号由 SEV 指令执行输出。如果输入了某事件，则该内核会从 WFE 指令所导致的低功耗模式返回。

TMPM370FYDFG/FYFG 不使用事件输出信号与事件输入信号。请不要使用 SEV 指令与 WFE 指令。

2.5 电源管理

该 Cortex-M3 内核配备有使用 SLEEPING 信号与 SLEEPDEEP 信号的功率调节系统。在设置系统控制寄存器的<SLEEPDEEP>位时，可输出 SLEEPDEEP 信号。

在以下情况下会输出这些信号：

- 等待中断 (WFI) 指令执行
- 等待事件 (WFE) 指令执行
- 在设置了系统控制寄存器的<SLEEPONEXIT> 位的情况下，中断服务程序(ISR)时的时序。

TMPM370FYDFG/FYFG 不使用 SLEEPDEEP 信号，因此不必设置<SLEEPDEEP>位。其也不使用事件信号，因此请不要使用 WFE 指令。

有关电源管理的详细资料，请参看"时钟/模式控制"一章。

2.6 独占通道

在 Cortex-M3 内核中了，该 DCode 总线系统支持独占通道。不过，TMPM370FYDFG/FYFG 不使用该功能。

3. 存储器地址

3.1 存储器地址

TMPM370FYDFG/FYFG 的存储器地址基于 ARM Cortex-M3 处理器内核存储器地址。TMPM370FYDFG/FYFG 的内部 ROM，内部 RAM 与专用功能寄存器(SFR)分别被映射到该 Cortex-M3 的代码，SRAM 与周围区。专用功能寄存器(SFR)指的是所有输入输出口与外围功能的控制寄存器。该 SRAM 与 SFR 区域全被纳入该位-带区。

该 CPU 寄存器区是该处理器内核的内部寄存器区域。

有关各区域的详细资料，见《Cortex-M3 技术参考手册》。

注意被指示为"故障"的区域入口可导致存储器故障(如果存储器故障被启用)或硬故障(如果存储器故障被禁用)。另外，不要访问该卖方特定区域。

3.1.1 TMPM370FYDFG/FYFG 存储器地址

图 3-1 给出了 TMPM370FYDFG/FYFG 的存储器地址。

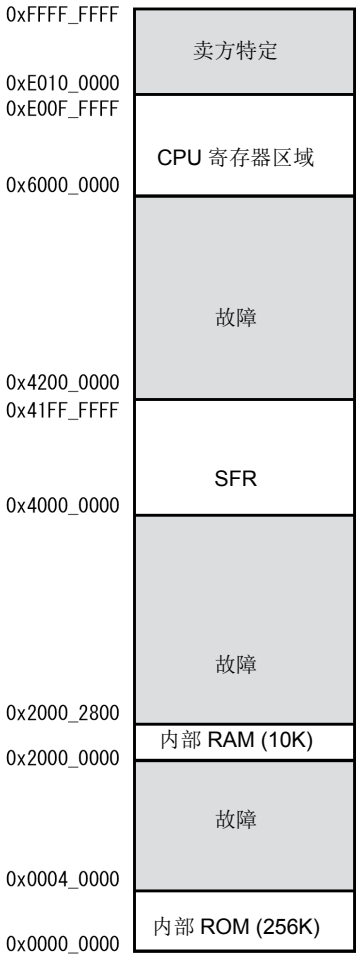


图 3-1 存储器地址

3.2 SFR 区域的详细

表 3-1 给出了该 SFR 区域的有关详细。

不要访问表 3-1 中的保留区。有关详细资料见各外围功能相关章节。

表 3-1 SFR 的详细资料

起始地址	结束地址	外围
0x4000_0000	0x4000_037F	PORT
0x4000_0380	0x4000_FFFF	保留
0x4001_0000	0x4001_01FF	TMRB
0x4001_0200	0x4001_03FF	保留
0x4001_0400	0x4001_053F	ENC
0x4001_0540	0x4002_007F	保留
0x4002_0080	0x4002_017F	SIO/UART
0x4002_0180	0x4002_FFFF	保留
0x4003_0000	0x4003_02FF	ADC
0x4003_0300	0x4003_FFFF	保留
0x4004_0000	0x4004_003F	WDT
0x4004_0040	0x4004_01FF	保留
0x4004_0200	0x4004_023F	CG
0x4004_0240	0x4004_07FF	保留
0x4004_0800	0x4004_083F	OFD
0x4004_0840	0x4004_08FF	保留
0x4004_0900	0x4004_093F	VLTD
0x4004_0940	0x4004_FFFF	保留
0x4005_0000	0x4005_023F	VE
0x4005_0240	0x4005_03FF	保留
0x4005_0400	0x4005_04FF	PMD
0x4005_0500	0x4007_FFFF	保留
0x4008_0000	0x41FF_EFFF	硬故障
0x41FF_F000	0x41FF_F03F	闪存
0x41FF_F040	0x41FF_FFFF	保留

4. 复位操作

4.1 初始状态

刚上电时，内部电路，寄存器设置与引脚状态均未定义。在施加了所有供电电压之后，该状态保持延续，直至 RESET 引脚收到"低"电平信号。

4.2 复位操作

TMPM370FYDFG/FYFG 具备上电复位电路，在电源合上时即生成上电复位信号。在通过外部复位引脚复位时，需向处于"低"电平状态的复位引脚输入复位信号至少持续 1.2 μsec，且供电电压应处于工作范围内。

4.3 复位之后

在复位解除时，Cortex-M3 处理器内核的系统控制寄存器与内部 I/O 寄存器即被初始化。注意，在复位解除之后，该 PLL 乘法电路即停止工作。因此，应设置 CGOSCCR 寄存器与 CGPLLSEL 寄存器，使之使用 PLL 乘法电路。

在执行了复位异常处理之后，该程序会转移至该中断服务程序。该中断服务程序启动时的地址存储在 0x0000_0004 中。

注 1：在 $\overline{\text{RESET}}$ 引脚被设置为"低"之后，即可打开电源。

注 2：该复位操作可改变内部 RAM 状态。

5. 时钟/模式控制

5.1 特征

该时钟/模式控制程序块允许选择时钟齿轮，预分频时钟，以及该 PLL 时钟乘法电路与振荡器的预热。

还有一个低功耗模式，该模式可通过模式推移减少功耗。

本章主要对时钟运行模式与模式推移进行了说明。

该时钟/模式控制程序块具备以下功能：

- 控制该系统时钟
- 控制该预分频时钟
- 控制该 PLL 乘法电路
- 控制该预热计时器

除 NORMAL 模式之外，TMPM370FYDFG/FYFG 还可在六种低功率模式下运行，以按照其使用条件减少功耗。

5.2 寄存器

5.2.1 寄存器列表

下表给出了与 CG 相关的各寄存器与地址。

基址 = 0x4004_0200

寄存器名称		地址(基本+)
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
系统时钟选择寄存器	CGCKSEL	0x0010

5.2.2 CGSYSCR (系统控制寄存器)

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

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

5.2.3 CGOSCCR (振荡控制寄存器)

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

位	比特符号	型号	功能
31-20	WUODR[11:0]	R/W	规定预热计时器的计数时间。
19-16	−	R/W	写作 "0y1110"。
15-12	−	R/W	写作 "0"。
11-10	−	R	读作 "0"。
9	−	R/W	写作 "0"。
8	XEN1	R/W	高速振荡器(外部) 0：停止 1：振荡 规定该高速振荡器(OSC)的运行。
7-4	−	R/W	读作 "0"。
3	WUPSEL1	R/W	预热计时器的时钟源 写作 "0"。
2	PLLON	R/W	PLL 运行 0：停止 1：振动 规定该 PLL 的运行。 其在复位之后停止。必须设置该位。
1	WUEF	R	预热计时器(WUP)的状态 0：预热已完成。 1：预热运行 启用以监视该预热计时器的状态。
0	WUEON	W	预热计时器的运行 0：忽略 1：开始预热 启用以启动该预热计时器。

5.2.4 CGSTBYCR (待机控制寄存器)

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

位	比特符号	型号	功能
31-18	−	R	读作 "0"。
17	−	R/W	写作 "0"。
16	DRVE	R/W	STOP 模式下的引脚状态 0：非激活 1：激活
15-10	−	R	读作 "0"。
9	−	R/W	写作 "0"。
8	RXEN	R/W	该 STOP 模式解除之后的高速振荡器运行。 写作 "1"。
7-3	−	R	读作 "0"。
2-0	STBY[2:0]	R/W	低功率模式 000：保留 001：STOP 010：保留 011：IDLE 100：保留 101：保留 110：保留 111：保留

5.2.5 CGPLLSEL (PLL 选择寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作 "0"。
15-12	−	R/W	写作 "1010"。
11	−	R	读作 "0"。
10-1	−	R/W	写作 "00_1001_1111"。
0	PLLSEL	R/W	PLL 的使用 0：fosc 使用 1：PLL 使用 规定使用或不用已被乘以 PLL 的时钟。 在复位之后，"fosc" 可被自动设置。在使用 PLL 时，必须进行复位。

5.3 时钟控制

5.3.1 时钟类型

各时钟被定义如下：

f_{osc} ：源自外部高速振荡器的时钟输入(X1 与 X2)

f_{PLL} ：被 PLL 八倍化的时钟

f_c ：CGPLLSEL<PLLSEL>所规定的时钟 (高速时钟)

f_{gear} ：CGSYSCR<GEAR[2:0]>所规定的时钟

f_{sys} ：与 fgear 相同的时钟(系统时钟)

f_{periph} ：CGSYSCR<FPSEL>所规定的时钟

$\phi T0$ ：CGSYSCR<PRCK[2:0]> 所规定的时钟(预分频时钟)

该高速时钟 fc 与该预分频时钟 $\phi T0$ 的可分性如以下所述。

高速时钟 ：fc, fc/2, fc/4, fc/8, fc/16

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

5.3.2 复位之后的初始值

复位操作可初始化该时钟配置(如以下所述)。

高速振荡器(外部) ：振荡

PLL (锁相环电路) ：停止

高速时钟齿轮 ：fc (无频率划分)

复位操作可导致所有时钟配置均变为与 f_{OSC} 的相同。

$f_C = f_{OSC}$

$f_{SYS} = f_C (= f_{OSC})$

$f_{periph} = f_C (= f_{OSC})$

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

5.3.3 时钟系统示意图

图 5-1 给出了该时钟系统示意图。

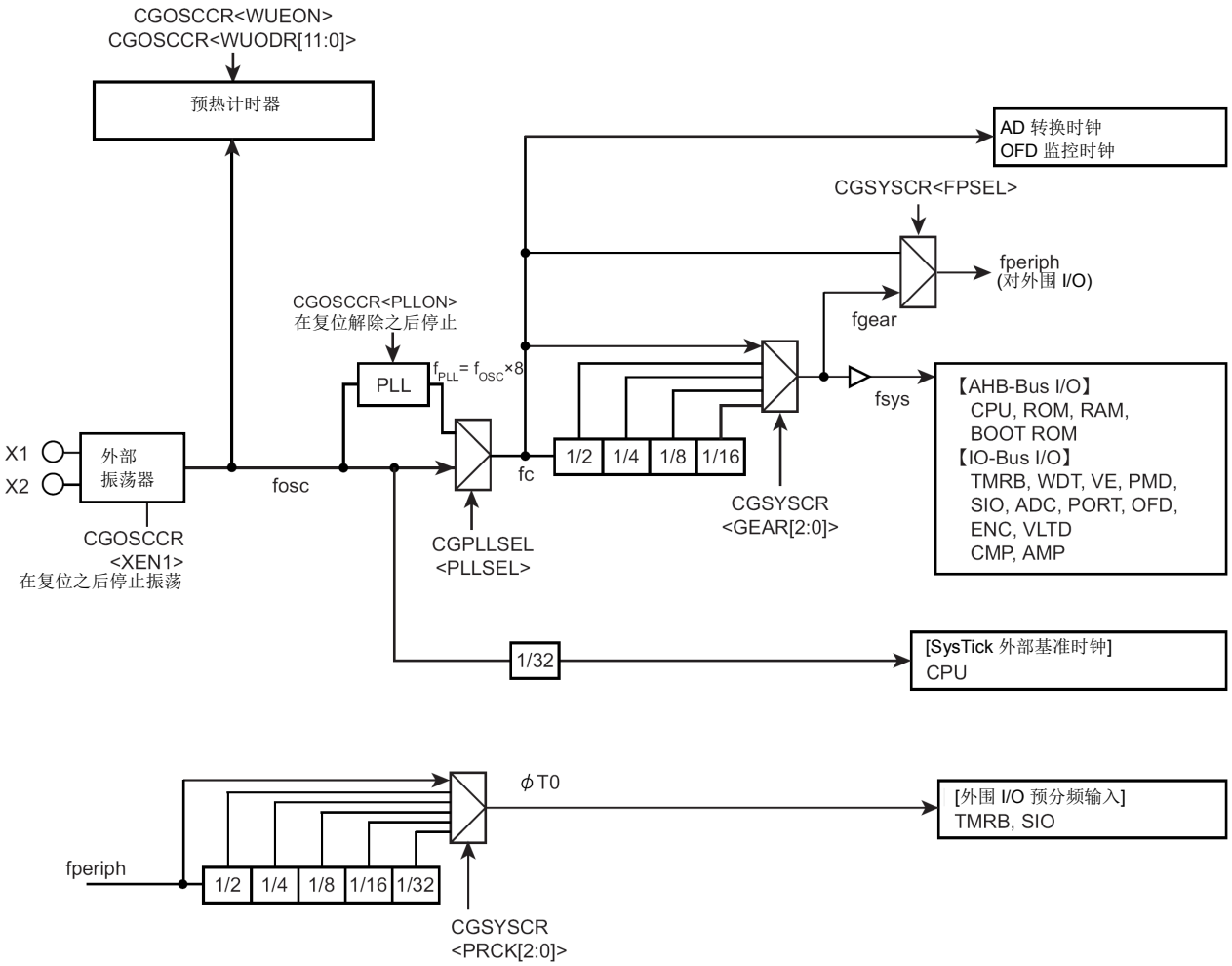


图 5-1 时钟方块图

在复位之后，该输入时钟选择器(显示时带有箭头记号)即被设置为默认值。

5.3.4 时钟乘法电路(PLL)

该电路可输出等于高速振荡输出时钟(fosc.)八倍的 f_{PLL} 时钟。因此，可将指向振荡器的输入频率设置为低，而降该内部时钟设置为高速。

在复位之后该 PLL 被禁用。将"1"设置到 CGOSCCR<PLLON>位，并将"1"设置到 CGPLLSEL<PLLSEL>，即可启用该 PLL。此时，f_{PLL} 时钟输出即变为高速振荡器(fosc)的八倍。

该 PLL 需要一定时间才能稳定，应用预热功能或其它方法使之固定。

注：PLL 的稳定需费时约 200 μs。

5.3.4.1 PLL 设置的顺序

以下给出了复位之后的 PLL 设置顺序。

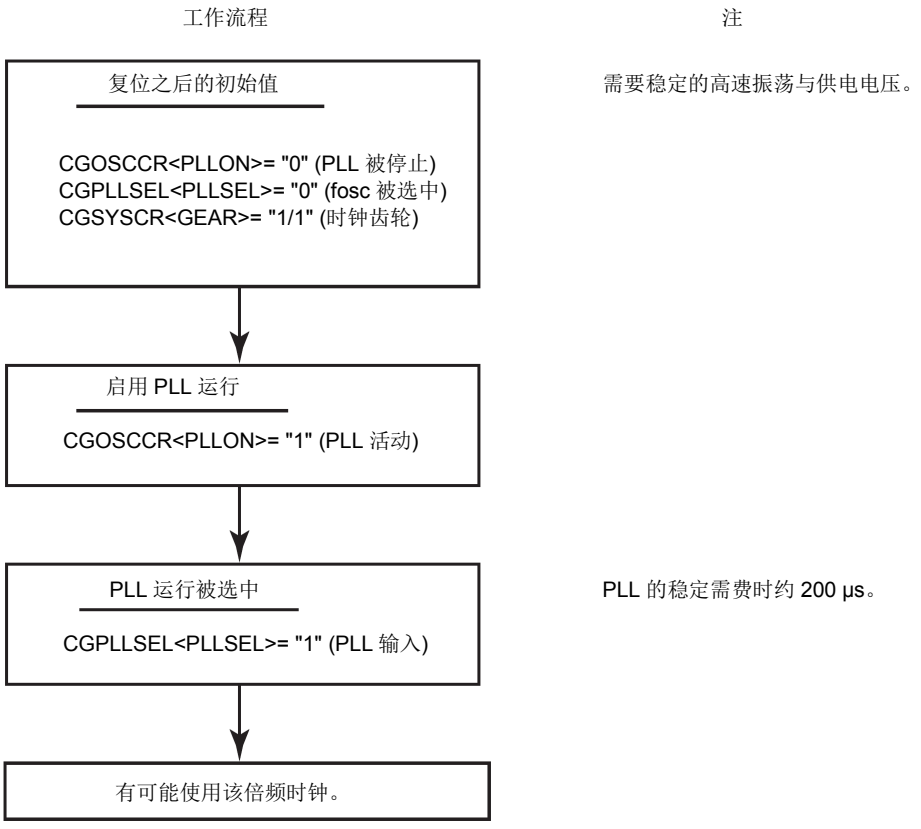


图 5-2 复位之后的 PLL 设置顺序

注：在操作员使 PLL 停止工作时，请检查并确认在设定了该 CGPLLSEL<PLLSEL> = "0"之后，其为寄存器 CGPLLSEL<PLLSEL> = "0"。然后，请设定 CGOSCCR<PLLON> = "0" (PLL 被停止)。

5.3.5 预热功能

该预热功能可利用该预热计时器，固定该振荡器稳定时间与 PLL。在从 STOP 模式返回时，需使用该预热功能。有关详细功能见"5.6.6 预热"。

注：在运行预热计时器期间，不要转入 STOP 模式。

在这种情况下，从低功耗模式返回的中断可触发该自动定时器计数。在达到给定时间之后，系统时钟被输出，CPU 开始运行。

在 STOP 模式下，PLL 被禁用。在从这些模式返回时，应参照该 PLL 与内部振荡器的稳定时间来配置该预热时间。

如何配置该预热功能。

1. 指定该向上计数时钟

   指定 CGOSCCR<WUPSEL1>与<WUPSEL2>位中的预热计数器的该向上计数时钟(将"0"写入 <WUPSEL1>，并将 "0"或"1"写入 <WUPSEL2>。"0"用于指定内部振荡器，而"1"则用于指定外部振荡器)。

2. 指定该预热计数器值

   通过设置 CGOSCCR<WUODR[11:0]>，可选择预热时间。

   以下给出了该预热设置及示例。

$$\text{预热周期} = \frac{\text{预热时间的给定值}}{\text{输入周期按频率(s)}}$$

<示例 1>在具备 8MHz 振荡器时，将预热时间设置为 5 ms

$$\frac{\text{预热时间的给定值}}{\text{输入周期按频率(s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40{,}000\text{个周期} = 0\text{x}9\text{C}40$$

最后 4 位下降，并将 0x9C4 设置到 CGOSCCR<WUODR[11:0]>中。

3. 确认预热的开始与完成

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

注：预热计时器按振荡时钟运行，且如果振荡频率中存在任何波动现象，则其可能包含误差。因此，该预热时间应被视为近似时间。

以下给出了该预热设置。

<示例> 固定 PLL 放入稳定时间 (fc = fosc1)

CGOSCCR<WUPSEL1> = "0"	：将"0" 写入 CGOSCCR<WUPSEL1>
CGOSCCR<WUODR[11:0]> = "0x9C4"	：预热时间设置
CGOSCCR<WUEON>="1"	：启动预热计数(WUP)
读取 CGOSCCR<WUEF>	：保持等待，直至状态变为"0" (预热完成)

5.3.6 系统时钟

该 TMPM370FYDFG/FYFG 提供高速时钟作为系统时钟。该高速时钟是可分的。

- 源自 X1 与 X2 的输入频率: 8 MHz ~ 10 MHz
- 时钟齿轮: 1/1, 1/2, 1/4, 1/8, 1/16 (复位之后：1/1)

表 5-1 高速频率的范围 (单位：MHz)

输入频率		最低运行频率	最高运行频率	复位之后 (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
OSC	8	1	80	8	64	32	16	8	4	8	4	2	1	-
	10			10	80	40	20	10	5	10	5	2.5	1.25	-

注 1：PLL=ON/OFF 设置：见 CGOSCCR<PLLON>。

注 2：将某值写入到 CGSYSCR<GEAR[2:0]>寄存器时，即执行时钟齿轮切换。在略微延迟之后，即发生实际切换。

注 3：."-"：保留

注 4：在使用"PLL =OFF"时，不要使用 1/16。

注 5：在使用 SysTick 时，不要使用 1/16。

5.3.7 预分频时钟控制

各外围功能均具备时钟划分用预分频。按照拟输入到各预分频的时钟 ϕT0，可按照 CGSYSCR<PRCK[2:0]>中的设置，划分 CGSYSCR<FPSEL>中所规定的"fperiph"时钟。在控制器复位之后，"fperiph/1"即被选作 ϕT0。

注：在使用时钟齿轮时，操作员所做的时间设置应能确保源自各外围功能的预分频输出 ϕ Tn 低于 fsys(ϕ Tn < fsys)。在定时计数器或其它外围功能运行期间，不要切换该时钟齿轮。

5.4 模式与模式推移

5.4.1 模式推移

NORMAL 模式采用高速时钟作为系统时钟。

该 IDLE 与 STOP 模式可用作低功耗模式，从而通过让处理器内核停止运行来降低功耗。

图 5-3 给出了模式推移图。

有关退出时睡眠的详细资料，请参看"Cortex-M3 技术参考手册"。

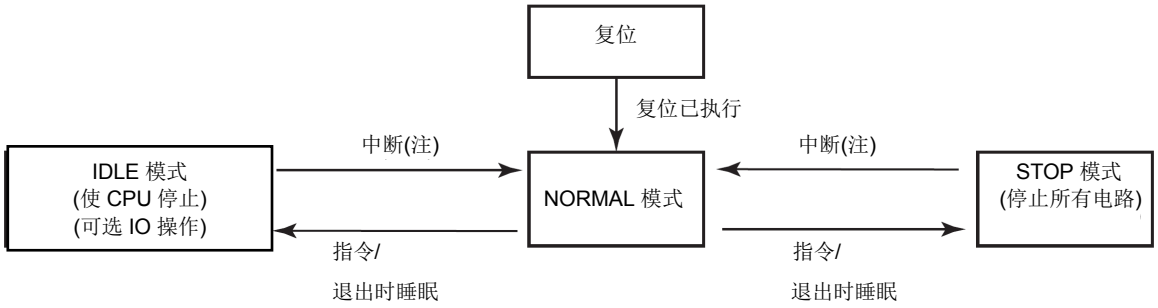


图 5-3 模式推移图

注：需进行预热。在转换到 STOP 模式之前，必须在 NORMAL 模式下设置该预热时间。关于预热时间，请参看"5.6.6 预热"。

5.5 运行模式

作为一种运行模式，NORMAL 可用。NORMAL 模式的特征见下一节。

5.5.1 NORMAL 模式

该模式可利用高速时钟运行该 CPU 内核与外围硬件。

在复位之后，其即被转入 NORMAL 模式。

5.6 低功率模式

TMPM370FYDFG/FYFG 有两种低功耗模式：IDLE 与 STOP。如拟转入某种低功耗模式，需在 CGSTBYCR<STBY[2:0]>中指定该模式，并执行该 WFI (等待中断)指令。在这种情况下，执行复位或生成中断即可解除该模式。如拟通过中断进行解除，需预先进行设置。有关详细资料见"异常"一章。

注 1：TMPM370FYDFG/FYFG 不提供任何低功耗模式解除用事件。禁止通过执行该 WFE(等待事件)指令来推移至低功耗模式。

注 2：TMPM370FYDFG/FYFG 不支持用 Cortex-M3 内核中的 SLEEPDEEP 位配置的低功耗模式。禁止设置该系统控制寄存器的 <SLEEPDEEP> 位。

各模式的特征如以下所述。

5.6.1 IDLE 模式

在该模式下，仅 CPU 被停止。各外围功能在其控制寄存器中均有一位，用于启动或禁止在该 IDLE 模式下运行。在进入 IDLE 模式时，在 IDLE 模式下其运行被禁止的各外围功能停止运行，并保持在当时的状态。

以下外围功能可在该 IDLE 模式下被启用或被禁用。有关设置的详细资料，见各外围功能相关章节。

- 16-位计时器/事件计数器(TMRB)
- 串行通道(SIO/UART)
- 看门狗计时器(WDT)
- 矢量引擎(VE)

注：在进入 IDLE 模式之前，应使 WDT 停止运行。

5.6.2 STOP 模式

在 STOP 模式下，所有内部电路包括内部振荡器均停止运行。

通过解除该 STOP 模式，该装置即可返回该 STOP 模式之前的模式，并开始运行。

该 STOP 模式允许通过设置 CGSTBYCR<DRVE>来选择引脚状态。表 5-2 给出了 STOP 模式下的引脚状态。

表 5-2 STOP 模式下的引脚状态

	引脚名称	I/O	<DRVE> = 0	<DRVE> = 1
非端口	X1	仅输入	×	
	X2	仅输出	"高"电平输出	
	$\overline{\text{RESET}}$, MODE	仅输入	o	
端口	TMS TCK TDI $\overline{\text{TRST}}$	输入	o	
	TDO	输出	被启用在数据有效时。 被禁用在数据无效时。	
	SWCLK	输入	o	
	SWDIO	输入	o	
		输出	被启用在数据有效时。 被禁用在数据无效时。	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	输出	o	
	UO0,1 VO0,1 WO0,1 XO0,1 YO0,1 ZO0,1	输出	被启用在数据有效时。 被禁用在数据无效时。	
	INT0, INT1, INT2 INT3, INT4, INT5 INT6, INT7, INT8 INT9, INTA, INTB INTC, INTD, INTE INTF	输入	o	
	除以上外的其它功能引脚，以及被用作通用端口的各端口	输入	×	o
		输出	×	o

o: 允许输入或输出。

×: 禁止输入或输出。

5.6.3 低功耗模式设置

该低功耗模式由待机控制寄存器 CGSTBYCR<STBY[2:0]>的设置规定。

表 5-3 给出了该模式在<STBY[2:0]>中的设置。

表 5-3 低功耗模式设置

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

注：除上述<STBY[2:0]>中的各值外，不要设置其它任何值。

5.6.4 各模式下的运行状态

表 5-4 给出了各模式下的运行状态。

对于 I/O 端口，"o" 与 "×"分别表示输入/输出被启用或被禁用。对于其它功能，"o" 与"×" 分别表示已提供时钟或未提供时钟。

表 5-4 各模式下的运行状态

模块	NORMAL	IDLE	STOP
处理器内核	o	×	×
I/O 端口	o	o	* (注 1)
PMD	o	o	×
ENC	o	o	×
OFD	o	o	×
ADC	o	o	×
VE	o		×
SIO	o	ON/OFF	×
SBI	o	可根据各模块	×
TMRB	o	进行选择	×
WDT	o		×
AMP/CMP	o	o	o (注 2)
VLTD	o	o	o (注 2)
POR	o	o	o (注 2)
CG	o	o	×
PLL	o	o	×
高速振荡器(fc)	o	o	×

o: 操作

×: 被停止

注 1：其取决于 CGSTBYCR<DRVE>。

注 2：即使该时钟被停止，该程序块也不会停止运行。

5.6.5 释放低功耗模式

可通过中断请求，非屏蔽中断 (NMI)或复位释放低功耗模式。可通过所选择的低功耗模式，确定可使用的释放释放源。

有关详细资料见表 5-5。

表 5-5 各模式下的释放释放源

低功率模式			IDLE (可编程)	STOP
释放源	中断	INT0 ~ F (注 1)	o	o
		INTRX0 ~ 3, INTTX0 ~ 3	o	×
		INTVCNA, INTVCNB	o	×
		INTEMG0 ~ 1	o	×
		INTOVV0 ~ 1	o	×
		INTADAPDA, INTADBPDA, INTADAPDB, INTADBPDB	o	×
		INTTB00, 10, 20, 30, 40, 50, 60, 70 INTTB01, 11, 21, 31, 41, 51, 61, 71	o	×
		INTPMD0, 1	o	×
		INTCAP00, 10, 20, 30, 40, 50, 60, 70 INTCAP01, 11, 21, 31, 41, 51, 61, 71	o	×
		INTADACPA, INTADBCPA, INTADACPB, INTADBCPB	o	×
		INTADASFT, INTADBSFT	o	×
		INTADATMR, INTADBTMR	o	×
		INTENC0, INTENC1	o	×
	SysTick		o	×
	NMI (INTWDT)		o	×
	RESET ($\overline{\text{RESET}}$ 引脚)		o	o

o: 在该模式被释放之后，开始中断处理 (复位初始化该 LSI)

×: 不可用

注 1：如拟通过采用电平模式中断来释放该低功耗模式，则需保持该电平，直至中断处理开始进行。如在此之前改变该电平，则会对中断处理的正确开始造成妨碍。

注 2：如拟转入该低功耗模式，可设置 CPU 以禁止除该释放释放源以外的所有中断。否则，可通过由未指定对象执行释放来实现唤醒。

注 3：关于预热时间，请参看"5.6.6 预热"。

- 通过中断请求实现释放

  如拟通过一次中断释放低功耗模式，则必须预先设置 CPU，以检测该中断。除 CPU 中的设置之外，还必须设置该时钟脉冲发生器，则必须预先设置 CPU，以检测拟用于释放该 STOP 模式的该中断。

- 通过非屏蔽中断实现释放 (NMI)

  有个看门狗计时器中断(INTWDT) 可被用作非屏蔽中断指令源。INTWDT 仅可用于该 IDLE 模式。

- 释放通过复位释放

  通过从 $\overline{\text{RESET}}$ 引脚进行复位，可释放任何低功耗模式。然后，该模式会切换到NORMAL模式，且所有寄存器均被初始化(与正常复位时相同)。

- 通过 SysTick 中断实现释放

SysTick 中断仅可用于该 IDLE 模式。

有关详细资料请参看"中断"一节。

5.6.6 预热

在进行模式推移时，可能会要求进行预热，以使得内部振荡器可提供稳定振荡。

在从 STOP 模式到 NORMAL 模式的模式推移过程中，该预热计数器会被自动激活。在所配置的预热时间消逝之后，系统时钟输出即被启动。在执行该指令以进入 STOP 模式之前，需在 CGOSCCR<WUPSEL1>(注 1)中设置拟用于预热的振荡器，并在 CGOSCCR<WUODR> 中设置预热时间。

注 1：始终将 CGOSCCR<WUPSEL1>设置为 "0"。

注 2：在 STOP 模式下，PLL 被禁用。在从这些模式返回时，应针对该 PLL 与内部振荡器的稳定时间，配置该预热时间。PLL 的稳定需费时约 200μs。

注 3：在设置从具备自动预热功能的低消耗模式返回时，不要将"1" 写入到 CGOSCCR<WUEON>位。

表 5-6 列出了各模式推移预热设置的必要性。

表 5-6 模式推移时的预热设置

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

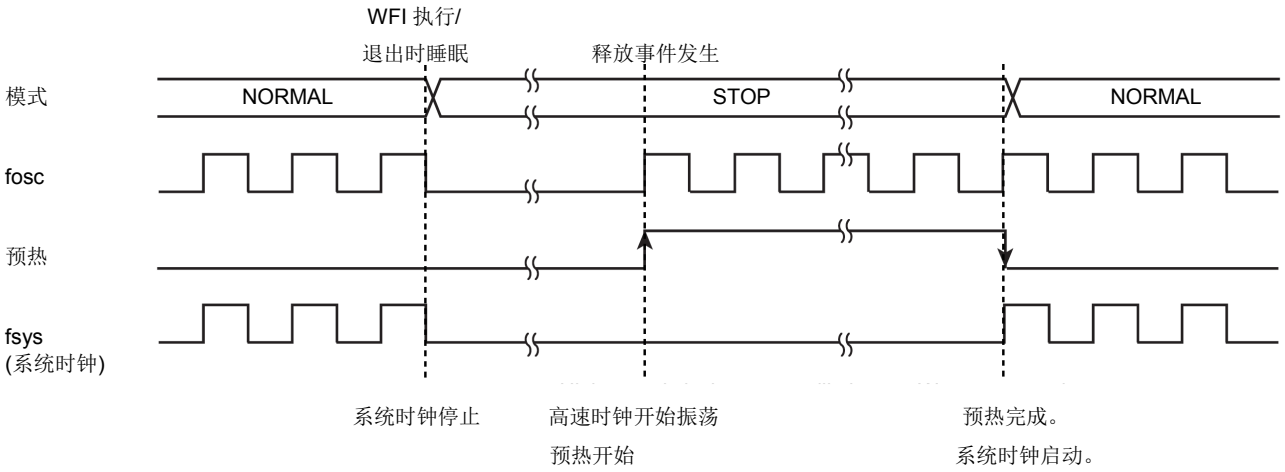
5.6.7 模式推移时的时钟运行

模式推移时的时钟运行如 5.6.7.1 一章所述。

5.6.7.1 运行模式的推移: NORMAL → STOP → NORMAL

在从 STOP 模式返回到 NORMAL 模式时，预热会被自动激活。在进入该 STOP 模式之前，需设置预热时间。

在通过复位返回到 NORMAL 模式时，不会引发自动预热。让该复位信号保持在已断言状态，直至振荡器运行变得稳定。



6. 异常

本章对各异常的特征，类型与处理进行说明。

异常与 CPU 内核之间有着密切的关系。如有必要，请参看"Cortex-M3 技术参考手册"。

6.1 概述

异常与 CPU 内核之间有着密切的关系。如有必要，请参看"Cortex-M3 技术参考手册"。

有两类异常：在某些误差条件发生或某异常生成指令被执行时所生成的异常，以及由硬件生成的异常，例如源自外部引脚或外围功能的中断请求信号。

所有异常均由 CPU 中的嵌套矢量中断控制器(NVIC)按照相应的优先级进行处理。在发生异常时，CPU 会将当前状态存储到该堆栈并转到相应的中断服务程序(ISR)。一旦该 ISR 完成，此前被存储该堆栈的信息即自动恢复。

6.1.1 异常类型

Cortex-M3 存在以下异常类型。

有关各异常的详细说明，请参看《Cortex-M3 技术参考手册》。

- 复位
- 非屏蔽中断(NMI)
- 硬故障
- 存储器管理
- 总线故障
- 使用故障
- SVCall (管理程序调用)
- 调试监督
- PendSV
- SysTick
- 外部中断

6.1.2 处理流程图

以下给出了异常/中断的处理方式。在以下说明中，[ ]表示硬件处理。[ ]表示软件处理。

各步骤将在本章后面的内容中进行说明。

6.1.2.1 异常请求与检测

(1) 异常发生

异常源包括 CPU 的指令执行，存取，以及源自外部中断引脚或外围功能的中断请求。

在 CPU 执行可引发异常的某指令，或在指令执行期间出现误差条件时，即发生异常。

在永不执行(XN)区进行指令取出，或对故障区的访问破坏，也可导致发生异常。

外部中断引脚或外围功能可导致生成中断请求。对于用于释放待机模式的中断而言，必须在时钟发生器中进行相关设置。有关详细资料，请参看"6.5 中断"。

(2) 异常检测

如果同时发生多个异常，则 CPU 会取优先级最高的异常。

表 6-1 给出了各异常的优先级。"可配置"指的是操作员可将某个优先级分频给该异常。存储器管理，总线故障与使用故障等异常可被启用或被禁用。如果发生某被禁用异常，其将被作为硬故障接受处理。

表 6-1 异常类型与优先级

编号	异常类型	优先级	描述
1	复位	−3 (最高)	复位引脚，WDT, POR, VLTD, OFD 或 SYSRETREQ
2	非屏蔽中断	−2	WDT
3	硬故障	−1	因某优先级较高的故障正在接受处理，或其被禁用而无法激活的故障
4	存储器管理	可配置	源自存储器保护装置(MPU) 的异常(注 1) 从永不执行(XN)区域进行指令取出时
5	总线故障	可配置	对存储器地址硬故障区域的访问破坏
6	使用故障	可配置	未定义指令执行，或与指令执行相关的其它故障
7~10	保留	−	
11	SVCall	可配置	用 SVC 指令进行系统服务调用
12	调试监督	可配置	在 CPU 无故障时调试监督程序
13	保留	−	
14	PendSV	可配置	可挂起系统服务申请
15	SysTick	可配置	源自系统记时器的通知
16~	外部中断	可配置	外部中断引脚或外围功能(注 2)

注 1：**本产品不含该 MPU。**

注 2：**各产品外部中断的源与数目各不相同。有关详细资料，见"6.5.1.5 中断源列表"。**

(3) 优先权设置

• 优先级

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

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

如果是 8-位配置，则优先级的配置范围为 0～255。最高优先级为"0"。如果存在具备相同优先级的多个元件，则数目越小，优先级应越高。

注：**<PRI_n>位被定义为本产品的位 3-位配置。**

• 优先级分组

优先级组可被分为若干组。通过设置该应用中断与复位控制寄存器的<PRIGROUP> ，可将<PRI_n>分成先占优先级与子优先级。

优先级与先占优先级类似。如果优先级与先占优先级相同，则其就比得上子优先级。如果子优先级与优先级相同，则异常数目越小，优先级就越高。

表 6-2 给出了优先级分组设置。该表中所列的该先占优先级与子优先级，均为〈PRI_n>被定义为 8-位配置时的数字。

表 6-2 优先级分组设置

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

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

6.1.2.2 异常处理与转入该中断服务程序(先占)

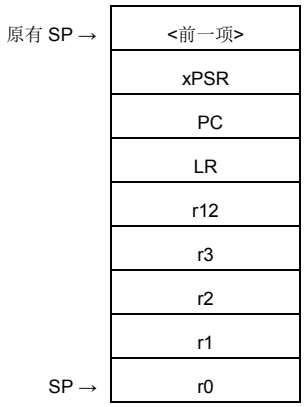
在异常发生时，CPU 会中止当前正在执行的处理，并转到中断服务程序。这就是所谓的"先占"。

(1) 堆栈

在 CPU 检测到异常时，其会按以下顺序，将以下八个寄存器的内容压入到该堆栈:

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

在压栈完成时，SP 即被减去八个字。以下给出了寄存器内容被压栈后该堆栈的状态。



(2) 取出 ISR

CPU 启用指令以取出该中断处理，并将数据存储至寄存器。

编制一个内含各异常 ISRs 前列地址的矢量表。复位后，该矢量表位于该代码区中的地址 0x0000_0000。通过设置该矢量表偏移寄存器，操作员可将该矢量表放置在该代码 SRAM 空间中的任何地址。

该矢量表还应包含该主堆栈的初始值。

(3) 迟来

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

迟来异常可导致 CPU 取出在转移到相应 ISR 时所需的新的向量地址，但 CPU 不会将寄存器内容重新压入该堆栈。

(4) 矢量表

该矢量表的配置如以下所示。

操作员必须始终设置前四个字(栈顶地址，复位 ISR 地址，NMI ISR 地址，以及硬故障 ISR 地址)。如有必要，可设置其它异常的 ISR 地址。

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

6.1.2.3 执行 ISR

ISR 可执行相应异常的必要处理。ISRs 必须由用户编制。

ISR 可能需要包含中断请求清除所需的代码，以确保在返回到正常程序执行时同一中断不会再次发生。

有关中断处理的详细资料，见"6.5 中断"。

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

6.1.2.4 异常出口

(1) 从 ISR 返回后的执行

在从某 ISR 返回时，CPU 会采取以下动作的其中之一：

- 尾链

  如果存在挂起异常且不存在堆栈异常，或该挂起异常的优先级高于所有的已堆栈异常，则 CPU 会返回到该挂起异常的 ISR。

  在这种情况下，CPU 在退出一个 ISR 并进入另一 ISR 时，会跳过八个寄存器的弹出，以及八个寄存器的压入。这就是所谓的"尾链"。

- 返回到上次堆栈的 ISR

  如果不存在挂起异常，或该最高优先级的已堆栈异常的优先级高于最高优先级的挂起异常，则 CPU 会返回到上次堆栈的 ISR。

- 返回到前一程序

  如果不存在挂起或已堆栈异常，则 CPU 会返回前一程序。

(2) 异常出口顺序

在从某 ISR 返回时，CPU 会执行以下操作：

- 弹出八个寄存器

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

- 加载当前活动中断号

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

- 选择 SP

  如果返回到某异常(处理器模式)，则 SP 是 SP_main。如果返回到线程模式，则 SP 可为 SP_main 或 SP_process。

6.2 复位异常

以下六个源可生成复位异常。

使用时钟发生器的该复位标志(CGRSTFLG)寄存器标识某个复位的源。

- 外部复位引脚

  在某外部复位引脚从"低"变为"高"时，会发生复位异常。

- 由 POR 导致的复位异常

  有关详细资料请参看"POR 上电复位电路"一章。

- 由 VLTD 导致的复位异常

  有关详细资料请参看"VLTD 电压检测电路"一章。

- 由 OFD 导致的复位异常

  有关详细资料请参看"OFD 振荡频率检测器"一章。

- 由 WDT 导致的复位异常

  看门狗计时器(WDT) 具备复位生成功能。有关详细资料见 WDT 相关章节。

- 由 SYSRESETREQ 导致的复位异常

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

6.3 非屏蔽中断(NMI)

看门狗计时器(WDT)具备屏蔽中断生成功能。有关详细资料见 WDT 相关章节。

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

6.4 SysTick

SysTick 具备中断功能(采用 CPU 的系统记时器)。

在操作员设置 SysTick 重新加载值寄存器中的某个值， 并启用 SysTick 控制器与状态寄存器中的 SysTick 功能时，该计数器会加载在该重新加载值寄存器中加载的该值，并开始递减计数。在该计数器达到"0"时，会发生 SysTick 异常。操作员可将其视为挂起异常，并利用某标志以掌握其何时达到 "0"。

该 SysTick 校准值寄存器可保持某个重新加载值，且保持的持续时间为 10ms(由系统记时器计时)。各产品的计数时钟频率各不相同，因此，各产品的 SysTick 校准值寄存器中的设置值也各不相同。

注：在本产品中，系统记时器可根据将源自 X1 引脚的时钟脉冲输入除以 32 所获取的时钟进行计时。

6.5 中断

本章对中断的路径，源与必要设置进行说明。

源自各中断源的中断信号可将中断请求通知 CPU。

其可设置各中断的优先级，并处理具备最高优先级的中断请求。

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

6.5.1 中断源

6.5.1.1 中断路径

图 6-1 给出了某中断请求路径。

并非用于释放待机的外围功能所发布的中断，即被直接输入到 CPU(路径 1)。

用于释放待机的外围功能(路径 2)，以及源自外部中断引脚(路径 3)的中断被输入到时钟发生器，并通过待机释放逻辑(路径 4 与 5)被输入到 CPU。

如果源自外部中断引脚的中断未被用于释放待机，则其会被直接输入到 CPU，不通过待机释放逻辑(路径 6)。

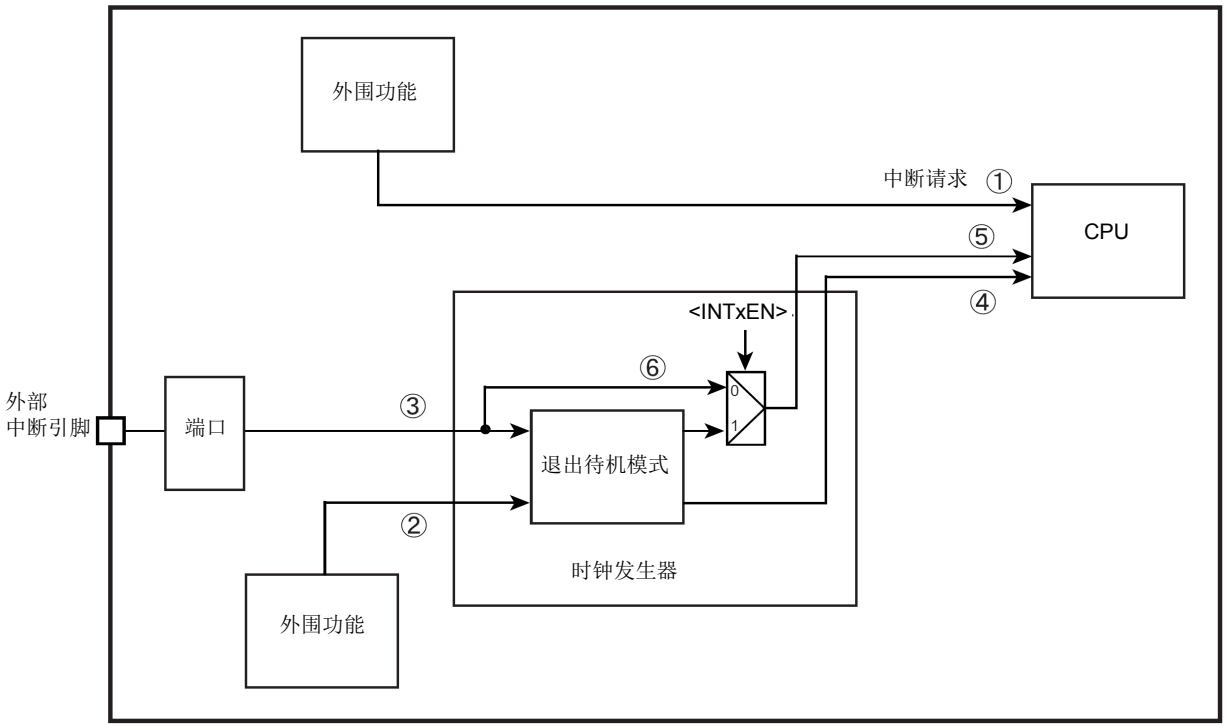


图 6-1 中断路径

6.5.1.2 生成

可从被指定为中断源的某外部引脚或外围功能，或通过设置 NVIC 的中断设置挂起寄存器，

生成一个中断请求。

- 从外部引脚

  设置该端口控制寄存，使得该外部引脚可用作中断功能引脚。

- 从外围功能

  设置该外围功能，使之可输出中断请求。

  有关详细资料见各外围功能相关章节。

- 通过设置中断设置挂起寄存器(强制挂起)

  通过设置该中断设置挂起寄存器的相关位，可生成一个中断请求。

6.5.1.3 传输

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

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

6.5.1.4 使用外部中断引脚时的预防措施

如果操作员使用外部中断，则应注意以下内容，以免生成意外中断。

如果输入被禁用(PxIE<PxmIE>="0")，则源自外部中断引脚的输入为"高"。此外，如果外部中断未被用作释放待机用触发信号(图 6-1 所示的路径 6)，则源自外部中断引脚的输入信号会被直接发送给 CPU。由于 CPU 会将"高"输入识别为中断，因此，如果 CPU 在输入被禁用时启用相应的中断，就会发生中断。

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

6.5.1.5 中断源列表

表 6-3 给出了中断源列表。

表 6-3 中断源列表

编号	中断源		激活电平 (清除待机)	CG 中断模式控制寄存器
0	INT0	中断引脚	高/低 缘/电平可选	CGIMCGA
1	INT1	中断引脚		
2	INT2	中断引脚		
3	INT3	中断引脚		
4	INT4	中断引脚	高/低 缘/电平可选	CGIMCGB
5	INT5	中断引脚		
6	INTRX0	串行接收(通道 0)		
7	INTTX0	串行传输(通道 0)		
8	INTRX1	串行接收(通道 1)		
9	INTTX1	串行传输(通道 1)		
10	INTVCNA	矢量引擎中断 A		
11	INTVCNB	矢量引擎中断 B		
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	PMD0 所触发的 ADCA 已完成		
17	INTADBPDA	PMD0 所触发的 ADCB 已完成		
18	INTADAPDB	PMD1 所触发的 ADCA 已完成		
19	INTADBPDB	PMD1 所触发的 ADCB 已完成		
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		

表 6-3 中断源列表

编号	中断源		激活电平 (清除待机)	CG 中断模式 控制寄存器
37	INTCAP51	16 位 TMRB5 输入捕捉 1		
38	INT6	中断引脚	高/低 缘/电平可选	CGIMCGB
39	INT7	中断引脚		
40	INTRX2	串行接收(通道 2)		
41	INTTX2	串行传输(通道 2)		
42	INTADACPA	ADCA 转换监控功能中断 A		
43	INTADBCPA	ADCB 转换监控功能中断 A		
44	INTADACPB	ADCA 转换监控功能中断 B		
45	INTADBCPB	ADCB 转换监控功能中断 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		
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	由软件启动的 ADC 单元 A 转换已完成		
55	INTADBSFT	由软件启动的 ADC 单元 B 转换已完成		
56	INTADATMR	由计时器触发的 ADC 单元 A 转换已完成		
57	INTADBTMR	由计时器触发的 ADC 单元 B 转换已完成		
58	INT8	中断引脚	高/低 缘/电平可选	CGIMCGC
59	INT9	中断引脚		
60	INTA	中断引脚		
61	INTB	中断引脚		
62	INTENC0	编码器输入 0 中断		
63	INTENC1	编码器输入 1 中断		
64	INTRX3	串行接收(通道 3)		
65	INTTX3	串行传输(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	中断引脚	高/低 缘/电平可选	CGIMCGD
75	INTD	中断引脚		
76	INTE	中断引脚		
77	INTF	中断引脚		

6.5.1.6 激活电平

激活电平可改变中断源的信号，并触发中断。CPU 会将"高"电平中断信号识别为中断。直接发自外围功能的中断信号，会被组态为输出"高"并用于指示中断请求。

激活电平可被设置为中断时钟发生器，其可被用作待机释放用触发信号。源自外围功能的中断请求，可被设置为上升沿或下降沿触发型。源自中断引脚的中断请求可被设置为电平感应型("高"或"低") 或沿触发型(上升或下降)。

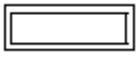
如果某中断源被用于清除待机模式，则必须设置相关时钟发生器寄存器。启用 CGIMCGx<INTxEN>位，并指定各 CGIMCGx<EMCGx>位中的作用电平。操作员必须按表 6-3 所示设置源自各外围功能中断请求的激活电平。

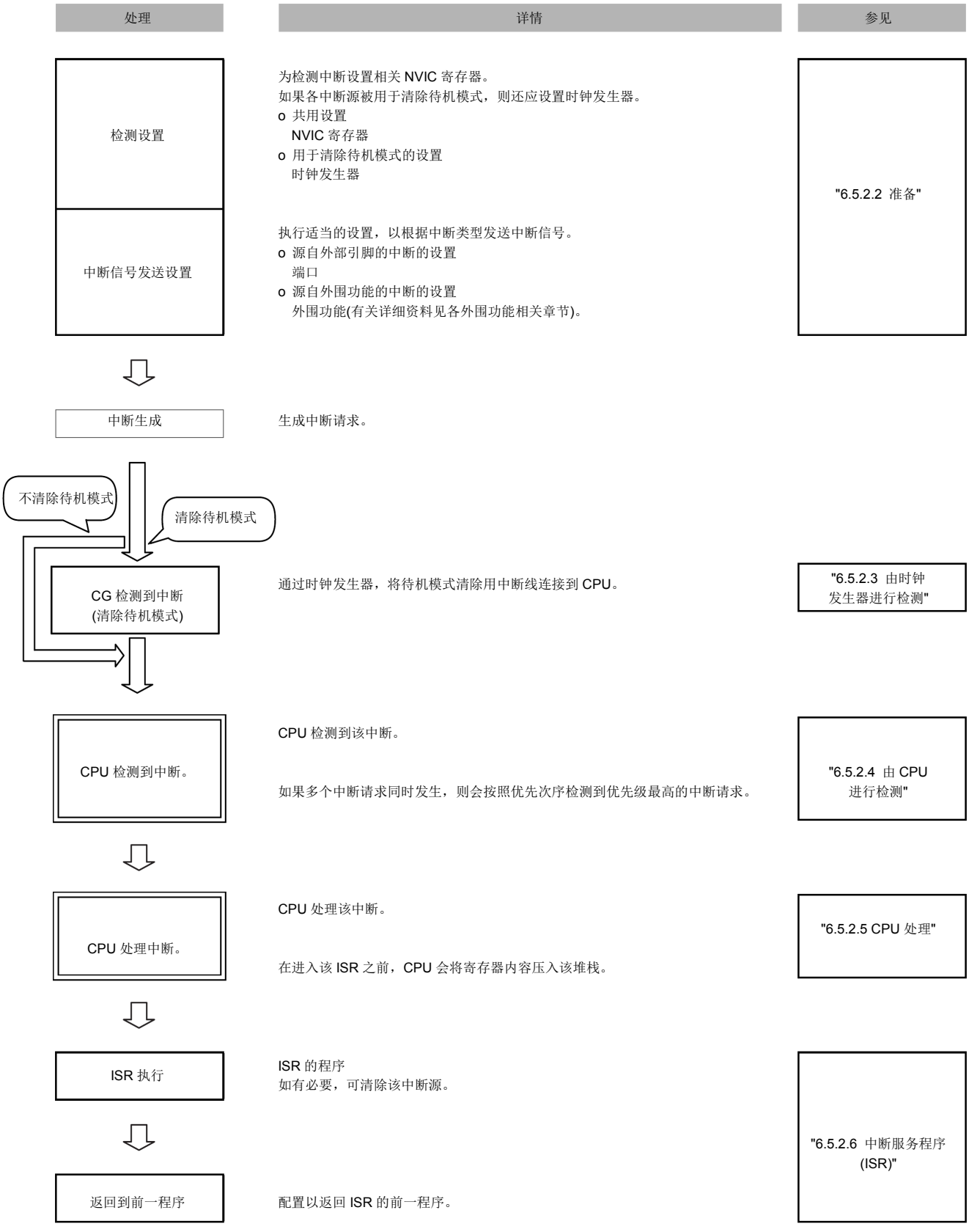
可用一个"高"电平信号将时钟发生器所检测到的中断请求通知 CPU。

6.5.2 中断处理

6.5.2.1 流程图

以下给出了中断的处理方式。

以下给出了异常/中断的处理方式。在以下说明中，  表示硬件处理。  表示软件处理。


处理
详情
参见
检测设置
为检测中断设置相关 NVIC 寄存器。
如果各中断源被用于清除待机模式，则还应设置时钟发生器。
o 共用设置
NVIC 寄存器
o 用于清除待机模式的设置
时钟发生器
中断信号发送设置
执行适当的设置，以根据中断类型发送中断信号。
o 源自外部引脚的中断的设置
端口
o 源自外围功能的中断的设置
外围功能(有关详细资料见各外围功能相关章节)。
"6.5.2.2 准备"
中断生成
生成中断请求。
不清除待机模式
清除待机模式
CG 检测到中断
(清除待机模式)
通过时钟发生器，将待机模式清除用中断线连接到 CPU。
"6.5.2.3 由时钟
发生器进行检测"
CPU 检测到中断。
CPU 检测到该中断。
如果多个中断请求同时发生，则会按照优先次序检测到优先级最高的中断请求。
"6.5.2.4 由 CPU
进行检测"
CPU 处理中断。
CPU 处理该中断。
在进入该 ISR 之前，CPU 会将寄存器内容压入该堆栈。
"6.5.2.5 CPU 处理"
ISR 执行
ISR 的程序
如有必要，可清除该中断源。
返回到前一程序
配置以返回 ISR 的前一程序。
"6.5.2.6 中断服务程序
(ISR)"

6.5.2.2 准备

在为某中断进行准备时，操作员需要注意配置顺序，以避免中途发生任何意外中断。

基本上应按以下顺序执行中断的初始化改变其配置。应用 CPU 禁用该中断。从 CPU 按最远的路径进行配置。此时用 CPU 启用该中断。

在配置该时钟发生器时，操作员必须遵循本文所指示的顺序，以避免出现任何意外中断。首先配置该前置条件。其次，清除时钟发生器中与该中断相关的数据，然后启用该中断。

以下各节均按中断处理的顺序列出，并将对其配制方法进行说明。

1. 通过 CPU 禁用中断
2. CPU 寄存器设置
3. 预配置(1) (源自外部引脚的中断)
4. 预配置(2) (源自外围功能的中断)
5. 预配置(3) (中断设置挂起寄存器)
6. 配置该时钟发生器
7. 通过 CPU 启用中断

(1) 通过 CPU 禁用中断

为使 CPU 不接受任何中断，可将"1"写入到 PRIMASK 寄存器的相应位。所有中断与异常非可屏蔽中断与硬故障除外均会被屏蔽。

用"MSR"指令设置该寄存器。

中断屏蔽寄存器		
PRIMASK	←	"1"(中断停用)

注 1：用户存取级不能修改 PRIMASK 寄存器。

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

(2) CPU 寄存器设置

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

各中断源均具备八个位用于指定从 0 ~ 255 的优先级，但各产品实际使用的位的数目各不相同。优先级 0 是最高优先级。如果多个源具备同一优先级，则编号最小的中断源具备最高的优先级。

通过使用应用中断与复位控制寄存器中的<PRIGROUP>，操作员可指定分组优先级。

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

注："n" 指相应的异常/中断。

本产品有三个位用于指定优先级。

(3) 预配置(1) (源自外部引脚的中断)

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

端口寄存器		
PxFRn<PxmFn>	←	"1"
PxIE<PxmIE>	←	"1"

注：x: 端口编号/ m: 相应的位/ n: 功能寄存器编号在各模式 STOP 模式除外下， 通过将 PxIE 设置为启动输入，即可启用相应的中断输入与 PxFR 设置无关。注意不得启用未使用的中断。此外，还应留意"6.5.1.4 使用外部中断引脚时的预防措施"所述内容。

(4) 预配置(2) (源自外围功能的中断)

该设置随所用外围功能的不同而不同。有关详细资料见各外围功能相关章节。

(5) 预配置(3) (中断设置挂起寄存器)

在利用该中断设置挂起寄存器生成中断时，需将"1"设置到该寄存器的相应位。

NVIC 寄存器		
中断设置挂起[m]	←	"1"

注：m: 相应的位

(6) 配置该时钟发生器

对于拟用于待机模式退出的中断源，操作员需在时钟发生器的 CGIMCG 寄存器中设置作用电平，并启用中断。CGIMCG 寄存器能够配置各源。

在启用某中断之前，清除已保留的相应中断请求。这样可避免发生意外中断。在清除相应的中断请求时，可将与拟使用中断对应的某值写入到该 CGICRCG 寄存器。有关各值见"6.6.3.5 CGICRCG (CG 中断请求清除寄存器)"。

在未设置时钟发生器的情况下即可使用源自外部引脚的中断请求，但前提是其未被用于待机模式的退出。不过，必须输入一个"高"脉冲或"高"电平信号，使得 CPU 可检测到它，并将其识别为中断请求。此外，还应留意"6.5.1.4 使用外部中断引脚时的预防措施"所述内容。

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

注: n: 寄存器编号 / m: 中断源的指定编号

(7) 通过 CPU 启用中断

通过 CPU 启用中断，如以下所述。

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

通过将"1"写入到该中断清除挂起寄存器的相应位，即可清除该已暂停的中断。通过将"1"写入到该中断设置启动寄存器的相应位，即可启用该预定中断。

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

最后，PRIMASK 寄存器被清零。

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

注 1：m: 相应的位

注 2：用户存取级不能修改 PRIMASK 寄存器。

6.5.2.3 由时钟发生器进行检测

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

一旦检测到沿触发中断请求，该中断请求即被保留在该时钟发生器中。必须将电平感应中断请求保持在该作用电平，直至其被检测到；否则，在该信号电平从活动变为非活动时，该中断请求就会不再存在。

在时钟发生器检测到某个中断请求时，它会始终按"高"电平将该中断信号发送给 CPU，直至该中断请求在 CG 中断请求清除(CGICRCG)寄存器中被清除。如果在未清除该中断请求的情况下退出待机模式，则在恢复正常运行时会再次检测到同一中断。务必清除 ISR 中的各中断请求。

6.5.2.4 由 CPU 进行检测

CPU 可检测到具备最高优先级的中断请求。

6.5.2.5 CPU 处理

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

6.5.2.6 中断服务程序(ISR)

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

(1) 在 ISR 运行期间进行压入

ISR 一般会按要求将 寄存器内容压入到堆栈，并处理某一中断。Cortex-M3 内核可将 PC，PSR，r0-r3，r12 与 LR 的内容自动压入到该堆栈。无需对其进行额外编程。

如果需要，还可压入其它寄存器的内容。

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

(2) 清除某个中断源

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

如某中断源被设置为电平感应型，则其会继续存在，直至其在某源时被清除。因此，必须清除该中断源。通过自动清除该中断源，可从时钟发生器清除该中断请求信号。

如果中断被设置为沿感应型，则通过在该 CGICRCG 寄存器中设置对应值，即可清除某个中断请求。在激活沿再次出现时，将检测到新的中断请求。

6.6 异常/中断相关寄存器

本章所述 CPU 的 NVIC 寄存器与时钟发生器寄存器如以下所示，并给出了其相应地址。

6.6.1 寄存器列表

NVIC 寄存器　　基址= 0xE000_E000

寄存器名称	地址
SysTick 控制器与状态寄存器	0x0010
SysTick 重新加载值寄存器	0x0014
SysTick 当前值寄存器	0x0018
SysTick 校准值寄存器	0x001C
中断设置启用寄存器 1	0x0100
中断设置启用寄存器 2	0x0104
中断设置启用寄存器 3	0x0108
中断清除启用寄存器 1	0x0180
中断清除启用寄存器 2	0x0184
中断清除启用寄存器 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
中断优先权寄存器	0x0400 ~ 0x0460
矢量表偏移寄存器	0x0D08
应用中断与复位控制寄存器	0x0D0C
系统处理器优先寄存器	0x0D18, 0x0D1C, 0x0D20
系统处理器控制器与状态寄存器	0x0D24

时钟发生器寄存器 基址 = 0x4004_0200

寄存器名称		地址
CG 中断请求清除寄存器	CGICRCG	0x0014
NMI 标志寄存器	CGNMIFLG	0x0018
复位标志寄存器	CGRSTFLG	0x001C
CG 中断模式控制寄存器 A	CGIMCGA	0x0020
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
保留	-	0x0030
保留	-	0x0034
保留	-	0x0038
保留	-	0x003C

注：禁止访问各"保留"区域。

6.6.2 NVIC 寄存器

6.6.2.1 SysTick 控制器与状态寄存器

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

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

注：在本产品中，系统记时器可根据将源自 X1 引脚的时钟脉冲输入除以 32 所获取的时钟进行计时。

6.6.2.2 SysTick 重新加载值寄存器

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

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

6.6.2.3 SysTick 校正值寄存器

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

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

6.6.2.4 SysTick 校准值寄存器

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

位	比特符号	型号	功能
31	NOREF	R	0：具备基准时钟 1：无基准时钟
30	SKEW	R	0：校准值是 10ms。 1：校准值不是 10ms。
29-24	–	R	读作 0。
23-0	TENMS	R	校准值 重新加载值，使得外部基准时钟可使用 10ms 时序(0xC35)。(注)

注：如果是重合多次，则请使用<TENMS>-1。

6.6.2.5 中断设置启用寄存器 1

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

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

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.6 中断设置启用寄存器 2

	31	30	29	28	27	26	25	24
比特符号	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
比特符号	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
比特符号	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
比特符号	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

位	比特符号	型号	功能
31-0	SETENA	R/W	中断号[63:32] [写入] 1：启用 [读取] 0：禁用 1：启用 每个位均与所指定的中断号对应。 通过将"1"写入到该寄存器中的某个位，即可启用相应的中断。写入"0"无任何影响。 通过读取各位，即可查看相应中断的启用/禁用条件。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.7 中断设置启用寄存器 3

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	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
比特符号	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

位	比特符号	型号	功能
31-14	−	R/W	读作 0。
13-0	SETENA	R/W	中断号[77:64] [写入] 1：启用 [读取] 0：禁用 1：启用 每个位均与所指定的中断号对应。 通过将"1"写入到该寄存器中的某个位，即可启用相应的中断。写入"0"无任何影响。 通过读取各位，即可查看相应中断的启用/禁用条件。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.8 中断清除启用寄存器 1

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

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

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.9 中断清除启用寄存器 2

	31	30	29	28	27	26	25	24
比特符号	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
比特符号	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
比特符号	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
比特符号	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

位	比特符号	型号	功能
31-0	CLRENA	R/W	中断号[63:32] [写入] 1：禁用 [读取] 0：禁用 1：启用 每个位均与所指定的中断号对应。其可用于启用中断，以及检测各中断是否被禁用。 通过将"1"写入到该寄存器中的某个位，即可启用相应的中断。写入"0"无任何影响。 通过读取各位，即可查看相应中断的启用/禁用条件。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.10 中断清除启用寄存器 3

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	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
比特符号	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

位	比特符号	型号	功能
31-14	−	R/W	读作 0。
13-0	CLRENA	R/W	中断号[77:64] [写入] 1：禁用 [读取] 0：禁用 1：启用 每个位均与所指定的中断号对应。其可用于启用中断，以及检测各中断是否被禁用。 通过将"1"写入到该寄存器中的某个位，即可启用相应的中断。写入"0"无任何影响。 通过读取各位，即可查看相应中断的启用/禁用条件。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.11 中断设置挂起寄存器 1

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

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

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.12 中断设置挂起寄存器 2

	31	30	29	28	27	26	25	24
比特符号	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
比特符号	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
比特符号	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
比特符号	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

位	比特符号	型号	功能
31-0	SETPEND	R/W	中断号[63:32] [写入] 1：挂起 [读取] 0：未挂起 1：挂起 各位对应于指定的编号，强迫各中断进入挂起状态，并可确定当前处于挂起状态的中断。 通过将"1"写入到该寄存器中的某个位，即可使相应的中断进入挂起状态。不过，对于已处于挂起状态或已被禁用的中断而言，写入"1"对其无任何影响。写入"0"无任何影响。 通过读取该位，即可返回相应中断的当前状态。 通过将"1"写入到中断清除挂起寄存器中的某个对应位，即可清除该寄存器中的该位。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.13 中断设置挂起寄存器 3

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	15	14	13	12	11	10	9	8
比特符号	-	-	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
比特符号	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

位	比特符号	型号	功能
31-14	–	R/W	读作 0。
13-0	SETPEND	R/W	中断号[77:64] [写入] 1：挂起 [读取] 0：未挂起 1：挂起 各位对应于指定的编号，强迫各中断进入挂起状态，并可确定当前处于挂起状态的中断。 通过将"1"写入到该寄存器中的某个位，即可使相应的中断进入挂起状态。 不过，对于已处于挂起状态或已被禁用的中断而言，写入"1"对其无任何影响。写入"0"无任何影响。 通过读取该位，即可返回相应中断的当前状态。 通过将"1"写入到中断清除挂起寄存器中的某个对应位，即可清除该寄存器中的该位。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.14 中断清除挂起寄存器 1

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

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

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.15 中断清除挂起寄存器 2

	31	30	29	28	27	26	25	24
比特符号	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
比特符号	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
比特符号	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
比特符号	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

位	比特符号	型号	功能
31-0	CLRPEND	R/W	中断号[63:32] [写入] 1：清除挂起中断 [读取] 0：未挂起 1：挂起 各位对应于指定的编号，强迫各中断进入挂起状态，并可确定当前处于挂起状态的中断。 通过将"1"写入到该寄存器中的某个位，即可清除相应的挂起中断。不过，对于已处于检修状态的中断而言，写入"1"对其无任何影响。写入"0"无任何影响。 通过读取该位，即可返回相应中断的当前状态。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.16 中断清除挂起寄存器 3

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	未定义	未定义	未定义	未定义	未定义	未定义	未定义	未定义
	15	14	13	12	11	10	9	8
比特符号	-	-	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
比特符号	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

位	比特符号	型号	功能
31-14	-	R/W	读作 0。
13-0	CLRPEND	R/W	中断号[77:64] [写入] 1：清除挂起中断 [读取] 0：未挂起 1：挂起 各位对应于指定的编号，强迫各中断进入挂起状态，并可确定当前处于挂起状态的中断。 通过将"1"写入到该寄存器中的某个位，即可清除相应的挂起中断。不过，对于已处于检修状态的中断而言，写入"1"对其无任何影响。写入"0"无任何影响。 通过读取该位，即可返回相应中断的当前状态。

注：有关中断与中断号的说明，见"6.5.1.5 中断源列表"一节。

6.6.2.17 中断优先权寄存器

各中断均具备中断优先权寄存器的八个位。

以下给出了各中断优先权寄存器的地址(与中断号对应)。

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28
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_77	PRI_76

各产品拟用于优先级指定的位数各不相同。本产品有三个位用于指定优先级。

以下给出了各中断优先权寄存器中断号 0～3 的字段。用于其它所有中断号的各中断优先权寄存器，均具备完全相同的字段。未使用的位在读取时会返回"0"，且写入未使用的位将不起任何作用。

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

位	比特符号	型号	功能
31-29	PRI_3	R/W	中断号 3 的优先级
28-24	−	R	读作 0。
23-21	PRI_2	R/W	中断号 2 的优先级
20-16	−	R	读作 0。
15-13	PRI_1	R/W	中断号 1 的优先级
12-8	−	R	读作 0。
7-5	PRI_0	R/W	中断号 0 的优先级
4-0	−	R	读作 0。

6.6.2.18 矢量表偏移寄存器

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

位	比特符号	型号	功能
31-30	–	R	读作 0。
29	TBLBASE	R/W	表基 本矢量表位于： 0：代码空间 1：SRAM 空间
28-7	TBLOFF	R/W	偏移值 将偏移值从空间顶部设置到 TBLBASE 中的指定空间。 必须根据该表中异常的数目校准该偏移。这意味着最小校准量为 32 字(操作员可将其用于 16 个中断)。如果中断数目较大，则操作员必须通过四舍五入调节该校准量，最高可为二的下一个幂。
6-0	–	R	读作 0。

6.6.2.19 应用中断与复位控制寄存器

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

位	比特符号	型号	功能
31-16	VECTKEY (已写入) / VECTKEYSTAT (已读取)	R/W	寄存器键 [写入]如拟写入到该寄存器，则 <VECTKEY>字段中必须有 0x5FA 字样。 [读取] 读作 0xFA05。
15	ENDIANESS	R/W	字节存储次序位: (注 1) 1：从大到小 0：从小到大
14-11	−	R	读作 0。
10-8	PRIGROUP	R/W	中断优先级分组 000：先占优先级七位，子优先权一位 001：先占优先级六位，子优先权二位 010：先占优先级五位，子优先权三位 011：先占优先级四位，子优先权四位 100：先占优先级三位，子优先权五位 101：先占优先级二位，子优先权六位 110：先占优先级一位，子优先权七位 111：先占优先级无位，子优先权八位 位组合旨在将该中断优先权寄存器<PRI_n>划分为先占优先级与子优先级。
7-3	−	R	读作 0。
2	SYSRESET REQ	R/W	系统复位请求 1=CPU 输出一个 SYSRESETREQ 信号(注 2)
1	VECTCLR ACTIVE	R/W	清除活动矢量位 1：清除活动 NMI，故障与中断的所有状态信息。 0：不清除。 该位可自行清除。 该应用负责重新初始化该堆栈。
0	VECTRESET	R/W	系统复位位 1：复位系统 0：不复位系统。 复位该系统，但各调试部件(FPB，DWT 与 ITM)除外；方法是设置"1"，并对该位进行清零。

注 1：**从小到大是本产品的默认存储格式。**

注 2：**在输出 SYSRESETREQ 时，本产品会执行热复位通过热复位即可清除<SYSRESETREQ>。**

6.6.2.20 系统处理器优先寄存器

各异常均具备系统处理器优先寄存器的八个位。

以下给出了各系统处理器优先寄存器的地址与各异常对应。

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

各产品拟用于优先级指定的位数各不相同。本产品有三个位用于指定优先级。

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

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

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

6.6.2.21 系统处理器控制器与状态寄存器

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	SVCALL 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
比特符号	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-19	-	R	读作 0。
18	USGFAULT ENA	R/W	使用故障 0：禁用 1：启用
17	BUSFAUL TENA	R/W	总线故障 0：禁用 1：启用
16	MEMFAULT ENA	R/W	存储器管理 0：禁用 1：启用
15	SVCALL PENDED	R/W	SVCall 0：未挂起 1：挂起
14	BUSFAULT PENDED	R/W	总线故障 0：未挂起 1：挂起
13	MEMFAULT PENDED	R/W	存储器管理 0：未挂起 1：挂起
12	USGFAULT PENDED	R/W	使用故障 0：未挂起 1：挂起
11	SYSTICKACT	R/W	SysTick 0：非激活 1：激活
10	PENDSVACT	R/W	PendSV 0：非激活 1：激活
9	-	R	读作 0。
8	MONITORACT	R/W	调试监督程序 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。
3	USGFAULT ACT	R/W	使用故障 0：非激活 1：激活
2	-	R	读作 0。
1	BUSFAULT ACT	R/W	总线故障 0：非激活 1：激活
0	MEMFAULT ACT	R/W	存储器管理 0：非激活 1：激活

注：操作员在清除或设置各激活位时必须非常小心，原因是这些位的清除与设置并不能修复栈内容。

6.6.3 时钟发生器寄存器

6.6.3.1 CGIMCGA (CG 中断模式控制寄存器 A)

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

位	比特符号	型号	功能
31	−	R	读作 0。
30-28	EMCG3[2:0]	R/W	INT3 待机清除请求的激活电平设置 (101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100: 两沿
27-26	EMST3[1:0]	R	INT3 待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
25	−	R	读作未定义。
24	INT3EN	R/W	INT3 清除输入 0：禁用 1：启用
23	−	R	读作 0。
22-20	EMCG2[2:0]	R/W	INT2 待机清除请求的激活电平设置(101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
19-18	EMST2[1:0]	R	INT2 待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
17	−	R	读作未定义。
16	INT2EN	R/W	INT2 清除输入 0：禁用 1：启用
15	−	R	读作 0。
14-12	EMCG1[2:0]	R/W	INT1 待机清除请求的激活电平设置。(101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
11-10	EMST1[1:0]	R	INT1 待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
9	−	R	读作未定义。
8	INT1EN	R/W	INT1 清除输入 0：禁用 1：启用
7	−	R	读作 0。
6-4	EMCG0[2:0]	R/W	INT0 待机清除请求的激活电平设置(101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
3-2	EMST0[1:0]	R	INT0 待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
1	−	R	读作未定义。

位	比特符号	型号	功能
0	INT0EN	R/W	INT0 清除输入 0：禁用 1：启用

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

注 2：请首先指定供该沿使用的位，然后再指定供<INTxEN>使用的位。禁止同时设置这两者。

6.6.3.2 CGIMCGB (CG 中断模式控制寄存器 B)

	31	30	29	28	27	26	25	24
比特符号	-	EMCG7			EMST7		-	INT7EN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
比特符号	-	EMCG6			EMST6		-	INT6EN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
比特符号	-	EMCG5			EMST5		-	INT5EN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
比特符号	-	EMCG4			EMST4		-	INT4EN
复位后	0	0	1	0	0	0	未定义	0

位	比特符号	型号	功能
31	-	R	读作 0。
30-28	EMCG7[2:0]	R/W	INT7 待机清除请求的激活电平设置 (101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
27-26	EMST7[1:0]	R	INT7 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
25	-	R	读作未定义。
24	INT7EN	R/W	INT7 清除输入 0：禁用 1：启用
23	-	R	读作 0。
22-20	EMCG6[2:0]	R/W	INT6 待机清除请求的激活电平设置 (101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
19-18	EMST6[1:0]	R	INT6 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
17	-	R	读作未定义。
16	INT6EN	R/W	INT6 清除输入 0：禁用 1：启用
15	-	R	读作 0。
14-12	EMCG5[2:0]	R/W	INT5 待机清除请求的激活电平设置 (101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
11-10	EMST5[1:0]	R	INT5 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
9	-	R	读作未定义。
8	INT5EN	R/W	INT5 清除输入 0：禁用 1：启用
7	-	R	读作 0。
6-4	EMCG4[2:0]	R/W	INT4 待机清除请求的激活电平设置(101~111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
3-2	EMST4[1:0]	R	INT4 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
1	-	R	读作未定义。

位	比特符号	型号	功能
0	INT4EN	R/W	INT4 清除输入 0：禁用 1：启用

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

注 2：请首先指定供该沿使用的位，然后再指定供<INTxEN>使用的位。禁止同时设置这两者。

6.6.3.3 CGIMCGC (CG 中断模式控制寄存器 C)

	31	30	29	28	27	26	25	24
比特符号	-	EMCGB			EMSTB		-	INTBEN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
比特符号	-	EMCGA			EMSTA		-	INTAEN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
比特符号	-	EMCG9			EMST9		-	INT9EN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
比特符号	-	EMCG8			EMST8		-	INT8EN
复位后	0	0	1	0	0	0	未定义	0

位	比特符号	型号	功能
31	−	R	读作 0。
30-28	EMCGB[2:0]	R/W	待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
27-26	EMSTB[1:0]	R	待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
25	−	R	读作未定义。
24	INTBEN	R/W	INTB 清除输入 0：禁用 1：启用
23	−	R	读作 0。
22-20	EMCGA[2:0]	R/W	INTA 待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
19-18	EMSTA[1:0]	R	INTA 待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
17	−	R	读作未定义。
16	INTAEN	R/W	INTA 清除输入 0：禁用 1：启用
15	−	R	读作 0。
14-12	EMCG9[2:0]	R/W	INT9 待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
11-10	EMST9[1:0]	R	待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
9	−	R	读作未定义。
8	INT9EN	R/W	INT9 清除输入 0：禁用 1：启用
7	−	R	读作 0。
6-4	EMCG8[2:0]	R/W	INT8 待机清除请求的激活电平设置。(101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
3-2	EMST8[1:0]	R	INT8 待机清除请求的激活电平 00：− 01：上升沿 10：下降沿 11：两沿
1	−	R	读作未定义。

位	比特符号	型号	功能
0	INT8EN	R/W	INT8 清除输入 0：禁用 1：启用

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

注 2： 请首先指定供该沿使用的位，然后再指定供<INTxEN>使用的位。禁止同时设置这两者。

6.6.3.4 CGIMCGD (CG 中断模式控制寄存器 D)

	31	30	29	28	27	26	25	24
比特符号	-	EMCGF			EMSTF		-	INTFEN
复位后	0	0	1	0	0	0	未定义	0
	23	22	21	20	19	18	17	16
比特符号	-	EMCGE			EMSTE		-	INTEEN
复位后	0	0	1	0	0	0	未定义	0
	15	14	13	12	11	10	9	8
比特符号	-	EMCGD			EMSTD		-	INTDEN
复位后	0	0	1	0	0	0	未定义	0
	7	6	5	4	3	2	1	0
比特符号	-	EMCGC			EMSTC		-	INTCEN
复位后	0	0	1	0	0	0	未定义	0

位	比特符号	型号	功能
31	-	R	读作 0。
30-28	EMCGF[2:0]	R/W	INTF 待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
27-26	EMSTF[1:0]	R	INTF 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
25	-	R	读作未定义。
24	INTFEN	R/W	INTF 清除输入 0：禁用 1：启用
23	-	R	读作 0。
22-20	EMCGE[2:0]	R/W	INTE 待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
19-18	EMSTE[1:0]	R	INTE 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
17	-	R	读作未定义。
16	INTEEN	R/W	INTE 清除输入 0：禁用 1：启用
15	-	R	读作 0。
14-12	EMCGD[2:0]	R/W	INTD 待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
11-10	EMSTD[1:0]	R	INTD 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
9	-	R	读作未定义。
8	INTDEN	R/W	INTD 清除输入 0：禁用 1：启用
7	-	R	读作 0。
6-4	EMCGC[2:0]	R/W	INTC 待机清除请求的激活电平设置。 (101 ~ 111：禁止设置) 000："低"电平 001："高"电平 010：下降沿 011：上升沿 100：两沿
3-2	EMSTC[1:0]	R	INTC 待机清除请求的激活电平 00：- 01：上升沿 10：下降沿 11：两沿
1	-	R	读作未定义。

位	比特符号	型号	功能
0	INTCEN	R/W	INTC 清除输入 0：禁用 1：启用

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

注 2： 请首先指定供该沿使用的位，然后再指定供<INTxEN>使用的位。禁止同时设置这两者。

6.6.3.5 CGICRCG (CG 中断请求清除寄存器)

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

位	比特符号	型号	功能
31-5	−	R	读作"0"。
4-0	ICRCG[4:0]	W	清除中断请求。 0_0000:INT0　0_1000: INT8 0_0001: INT1　0_1001: INT9 0_0010: INT2　0_1010:INTA 0_0011: INT3　0_1011: INTB 0_0100: INT4　0_1100:INTC 0_0101: INT5　0_1101: INTD 0_0110: INT6　0_1110: INTE 0_ 0111: INT7　0_1111: INTF　1_0000 ~ 1_1111: 保留 读作 0。

6.6.3.6 CGNMIFLG (NMI 标志寄存器)

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

位	比特符号	型号	功能
31-1	−	R	读作 0。
0	NMIFLG0	R	NMI 源生成标志 0：不适用 1：从 WDT 中生成

注：当读取<NMIFLG>时，它们被清除为" 0 "。

6.6.3.7 CGRSTFLG (复位标志寄存器)

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

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

注 1：该标志表示 CPU 的 NVIC 的应用中断和复位控制寄存器 SYSRESETREQ 位生成的复位。

注 2：该产品有上电复位电路，本寄存器只有通过上电复位才能初始化。因此，在上电后，"1"在初始复位状态下设置为<PONRSTF>位。注意该位不由第二复位及后面的复位设定，该寄存器不会自动清零。写入"0"，使寄存器清零。

7. 输入/ 输出端口

7.1 端口功能

7.1.1 功能表

TMPM370FYDFG/FYFG 有 76 个端口。除端口功能外，这些端口还能用作外设功能的 I/O 引脚。端口功能表如表 7-1 所示。

表 7-1 端口功能表

端口	引脚	输入/输出	上拉下拉	施密特输入	噪声滤波器	可编程开漏	功能引脚
PORTA							
	PA0	I/O	上拉/下拉	o	o	o	TB0IN , INT3
	PA1	I/O	上拉/下拉	o	-	o	TB0OUT
	PA2	I/O	上拉/下拉	o	o	o	TB1IN , INT4
	PA3	I/O	上拉/下拉	o	-	o	TB1OUT
	PA4	I/O	上拉/下拉	o	-	o	SCLK1 , $\overline{\text{CTS1}}$
	PA5	I/O	上拉/下拉	o	-	o	TXD1 , TB6OUT
	PA6	I/O	上拉/下拉	o	-	o	RXD1 , TB6IN
	PA7	I/O	上拉/下拉	o	o	o	TB4IN , INT8
PORTB							
	PB0	I/O	上拉/下拉	o	-	o	TRACECLK
	PB1	I/O	上拉/下拉	o	-	o	TRACEDATA0
	PB2	I/O	上拉/下拉	o	-	o	TRACEDATA1
	PB3	I/O	上拉/下拉	o	-	o	TMS / SWDIO
	PB4	I/O	上拉/下拉	o	-	o	TCK / SWCLK
	PB5	I/O	上拉/下拉	o	-	o	TDO / SWV
	PB6	I/O	上拉/下拉	o	-	o	TDI
	PB7	I/O	上拉/下拉	o	o	o	$\overline{\text{TRST}}$
PORTC							
	PC0	I/O	上拉/下拉	o	-	o	UO0
	PC1	I/O	上拉/下拉	o	-	o	XO0
	PC2	I/O	上拉/下拉	o	-	o	VO0
	PC3	I/O	上拉/下拉	o	-	o	YO0
	PC4	I/O	上拉/下拉	o	-	o	WO0
	PC5	I/O	上拉/下拉	o	-	o	ZO0
	PC6	I/O	上拉/下拉	o	-	o	$\overline{\text{EMG0}}$
	PC7	I/O	上拉/下拉	o	-	o	$\overline{\text{OVV0}}$
PORTD							
	PD0	I/O	上拉/下拉	o	-	o	ENCA0 , TB5IN

o：存在

-：不存在

表 7-1 端口功能表

端口	引脚	输入/输出	上拉 下拉	施密特 输入	噪声滤 波器	可编程 开漏	功能引脚
	PD1	I/O	上拉/下拉	o	-	o	ENCB0 , TB5OUT
	PD2	I/O	上拉/下拉	o	-	o	ENCZ0
	PD3	I/O	上拉/下拉	o	o	o	INT9
	PD4	I/O	上拉/下拉	o	-	o	SCLK2 , $\overline{\text{CTS2}}$
	PD5	I/O	上拉/下拉	o	-	o	TXD2
	PD6	I/O	上拉/下拉	o	-	o	RXD2
PORTE							
	PE0	I/O	上拉/下拉	o	-	o	TXD0
	PE1	I/O	上拉/下拉	o	-	o	RXD0
	PE2	I/O	上拉/下拉	o	-	o	SCLK0 , $\overline{\text{CTS0}}$
	PE3	I/O	上拉/下拉	o	-	o	TB4OUT
	PE4	I/O	上拉/下拉	o	o	o	TB2IN , INT5
	PE5	I/O	上拉/下拉	o	-	o	TB2OUT
	PE6	I/O	上拉/下拉	o	o	o	TB3IN , INT6
	PE7	I/O	上拉/下拉	o	o	o	TB3OUT , INT7
PORTF							
	PF0	I/O	上拉/下拉	o	-	o	TB7IN , $\overline{\text{BOOT}}$
	PF1	I/O	上拉/下拉	o	-	o	TB7OUT
	PF2	I/O	上拉/下拉	o	-	o	ENCA1 ,SCLK3 , $\overline{\text{CTS3}}$
	PF3	I/O	上拉/下拉	o	-	o	ENCB1 , TXD3
	PF4	I/O	上拉/下拉	o	-	o	ENCZ1 , RXD3
PORTG							
	PG0	I/O	上拉/下拉	o	-	o	UO1
	PG1	I/O	上拉/下拉	o	-	o	XO1
	PG2	I/O	上拉/下拉	o	-	o	VO1
	PG3	I/O	上拉/下拉	o	-	o	YO1
	PG4	I/O	上拉/下拉	o	-	o	WO1
	PG5	I/O	上拉/下拉	o	-	o	ZO1
	PG6	I/O	上拉/下拉	o	-	o	$\overline{\text{EMG1}}$
	PG7	I/O	上拉/下拉	o	-	o	$\overline{\text{OVV1}}$
PORTH							
	PH0	I/O	上拉/下拉	o	o	o	INT0 , AINA0
	PH1	I/O	上拉/下拉	o	o	o	INT1 , AINA1
	PH2	I/O	上拉/下拉	o	o	o	INT2 , AINA2
	PH3	I/O	上拉/下拉	o	-	o	AINA3
	PH4	I/O	上拉/下拉	o	-	o	AINA4
	PH5	I/O	上拉/下拉	o	-	o	AINA5
	PH6	I/O	上拉/下拉	o	-	o	AINA6
	PH7	I/O	上拉/下拉	o	-	o	AINA7

o : 存在

- : 不存在

表 7-1 端口功能表

端口	引脚	输入/输出	上拉 下拉	施密特 输入	噪声滤 波器	可编程 开漏	功能引脚
PORTI							
	PI0	I/O	上拉/下拉	o	-	o	AINA8
	PI1	I/O	上拉/下拉	o	-	o	AINA9 / AINB0
	PI2	I/O	上拉/下拉	o	-	o	AINA10 / AINB1
	PI3	I/O	上拉/下拉	o	-	o	AINA11 / AINB2
PORTJ							
	PJ0	I/O	上拉/下拉	o	-	o	AINB3
	PJ1	I/O	上拉/下拉	o	-	o	AINB4
	PJ2	I/O	上拉/下拉	o	-	o	AINB5
	PJ3	I/O	上拉/下拉	o	-	o	AINB6
	PJ4	I/O	上拉/下拉	o	-	o	AINB7
	PJ5	I/O	上拉/下拉	o	-	o	AINB8
	PJ6	I/O	上拉/下拉	o	o	o	INTC , AINB9
	PJ7	I/O	上拉/下拉	o	o	o	INTD , AINB10
PORTK							
	PK0	I/O	上拉/下拉	o	o	o	INTE , AINB11
	PK1	I/O	上拉/下拉	o	o	o	INTF , AINB12
PORTL							
	PL0	输入	-	o	o	-	INTB
	PL1	输入	-	o	o	-	INTA

o：存在

-：不存在

注：在常见条件下，噪声滤波器的噪声消除宽度约 30 ns。

7.1.2 端口寄存器概要

为了使用端口，需要配置下列寄存器。

• PxDATA: 端口 x 数据寄存器

读/写端口数据。

• PxCR: 端口 x 输出控制寄存器

控制输出。

为了控制输入，需要配置 PxIE。

• PxFRn: 端口 x 功能寄存器 n

设置功能。

通过设置"1"，就能激活指定的功能。

• PxOD: 端口 x 开漏控制寄存器

控制可编程开漏。

可编程开漏是通过设置 PxOD 而要实现的假开漏功能.

当 PxOD 设置为" 1 "时，输出缓冲器禁用而实现假开漏。

• PxPUP:端口 x 上拉控制寄存器

控制可编程上拉。

• PxPDN: 端口 x 下拉控制寄存器

控制可编程下拉。

• PxIE: 端口 x 输入控制寄存器

控制输入。

为了避免直通电流，默认设置禁止输入。

7.1.3 在 STOP 模式时端口状态

输入和输出在 STOP 模式时由 CGSTBYCR<DRVE>位启用/禁用。

若 PxIE 或 PxCR 在<DRVE> = 1 时启用，则输入或输入分别在 STOP 模式时启用。若<DRVE> = 0，除一些端口外，即使 PxIE 或 PxCR 启用，输入和输出在 STOP 模式时均禁用。

在 STOP 模式时引脚情况如表 7-2 所示。

表 7-2 在 STOP 模式时端口情况

	引脚名称	I/O	<DRVE> = 0	<DRVE> = 1
非端口	$\overline{\text{RESET}}$, MODE	仅输入	o	
端口	X1	仅输入	×	
	X2	仅输出	"高"电平输出	
	TMS TCK TDI TRST	输入	o	
	TDO	输出	被启用在数据有效时。 被禁用在数据无效时。	
	SWCLK	输入	o	
	SWDIO	输入	o	
		输出	被启用在数据有效时。 被禁用在数据无效时。	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	输出	o	
	UO0,1 VO0,1 WO0,1 XO0,1 YO0,1 ZO0,1	输出	被启用在数据有效时。 被禁用在数据无效时。	
	INT0, INT1, INT2 INT3, INT4, INT5 INT6, INT7, INT8 INTB INTC, INTD, INTE INTF	输入	o	
	除以上外的其它功能引脚，以及被用作通用端口的各端口	输入	×	o
		输出	×	o

o：允许输入或输出。

×：禁止输入或输出。

7.2 端口功能

本章描述端口寄存器详细情况。

本章仅描述"电路类型"读出电路配置。电路图见"7.3 端口方块图"。

7.2.1 端口 A (PA0 ~ PA7)

端口 A 是一个通用的 8-位输入/输出端口。可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 A 还执行串行接口功能(SIO/UART)，外部信号中断输入，16-位定时器输入/输出功能。

复位使端口 A 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

端口 A 有两类功能寄存器。若您将端口 A 用作通用端口，将两个寄存器的对应位设置为"0"。若您将端口 A 用作非通用端口，将功能寄存器的对应位设置为"1"。不要将一些功能寄存器同时设为"1"。

为了用外部中断输入释放 STOP 模式，在 PAFR 中选择该功能，并在 PAIE 寄存器中启用输入。即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

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

7.2.1.1 端口 A 电路类型

	7	6	5	4	3	2	1	0
型号	T12	T11	T13	T9	T2	T12	T2	T12

7.2.1.2 端口 A 寄存器

基址 = 0x4000_0000

寄存器名称		地址(基本+)
端口 A 数据寄存器	PADATA	0x0000
端口 A 输出控制寄存器	PACR	0x0004
端口 A 功能寄存器 1	PAFR1	0x0008
端口 A 功能寄存器 2	PAFR2	0x000C
端口 A 开漏控制寄存器	PAOD	0x0028
端口 A 上拉控制寄存器	PAPUP	0x002C
端口 A 下拉控制寄存器	PAPDN	0x0030
端口 A 输入控制寄存器	PAIE	0x0038

7.2.1.3 PADATA (端口 A 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号		PA6	PA5	PA4	PA3	PA2	PA1	PA0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PA7 ~ PA0	R/W	端口 A 数据寄存器

7.2.1.4 PACR (端口 A 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PA7C ~ PA0C	R/W	输出 0：禁用 1：启用

7.2.1.5 PAFR1 (端口 A 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7F1	PA6F1	PA5F1	PA4F1	PA3F1	PA2F1	PA1F1	PA0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PA7F1	R/W	0：端口 1：TB4IN
6	PA6F1	R/W	0：端口 1：RXD1
5	PA5F1	R/W	0：端口 1：TXD1
4	PA4F1	R/W	0：端口 1：SCLK1
3	PA3F1	R/W	0：端口 1：TB1OUT
2	PA2F1	R/W	0：端口 1：TB1IN
1	PA1F1	R/W	0：端口 1：TB0OUT
0	PA0F1	R/W	0：端口 1：TB0IN

7.2.1.6 PAFR2 (端口 A 功能寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7F2	PA6F2	PA5F2	PA4F2	-	PA2F2	-	PA0F2
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PA7F2	R/W	0：端口 1：INT8
6	PA6F2	R/W	0：端口 1：TB6IN
5	PA5F2	R/W	0：端口 1：TB6OUT
4	PA4F2	R/W	0：端口 1：$\overline{\text{CTS1}}$
3	−	R	读作 0。
2	PA2F2	R/W	0：端口 1：INT4
1	−	R	读作 0。
0	PA0F2	R/W	0：端口 1：INT3

7.2.1.7 PAOD (端口 A 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7OD	PA6OD	PA5OD	PA4OD	PA3OD	PA2OD	PA1OD	PA0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-0	PA7OD ~ PA0OD	R/W	0：CMOS 1：开漏

7.2.1.8 PAPUP (端口 A 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7UP	PA6UP	PA5UP	PA4UP	PA3UP	PA2UP	PA1UP	PA0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-0	PA7UP ~ PA0UP	R/W	上拉 0：禁用 1：启用

7.2.1.9 PAPDN (端口 A 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7DN	PA6DN	PA5DN	PA4DN	PA3DN	PA2DN	PA1DN	PA0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PA7DN ~ PA0DN	R/W	下拉 0：禁用 1：启用

7.2.1.10 PAIE (端口 A 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PA7IE ~ PA0IE	R/W	输入 0：禁用 1：启用

7.2.2 端口 B (PB0 ~ PB7)

端口 B 是一个通用的 8-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除了通用的输入/输出功能外，端口 B 还执行调试接口功能和调试跟踪输出功能。

为了执行调试接口功能，复位使 PB3，PB4，PB5，PB6，PB7 初始化。

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

当 PB5 起 TDO 或 SWV 的作用时，输出启用。当 PB6 起 TDI 的作用时，上拉启用。当 PB7 起 $\overline{\text{TRST}}$ 输入的作用时，上拉启用。

PB0，PB1，PB2 用作通用端口，输入，输出，上拉和下拉禁用。

注：若 PB3 被配置为 TMS/SWDIO 引脚，则不管 CGSTBYCR<DRVE>位的设置，输出甚至在 STOP 模式也启用。

7.2.2.1 端口 B 电路类型

	7	6	5	4	3	2	1	0
型号	T7	T7	T19	T8	T6	T18	T18	T18

7.2.2.2 端口 B 寄存器

基址 = 0x4000_0040

寄存器名称		地址(基本+)
端口 B 数据寄存器	PBDATA	0x0000
端口 B 输出控制寄存器	PBCR	0x0004
端口 B 功能寄存器 1	PBFR1	0x0008
端口 B 开漏控制寄存器	PBOD	0x0028
端口 B 上拉控制寄存器	PBPUP	0x002C
端口 B 下拉控制寄存器	PBPDN	0x0030
端口 B 输入控制寄存器	PBIE	0x0038

7.2.2.3 PBDATA (端口 B 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PB7 ~ PB0	R/W	端口 B 数据寄存器

7.2.2.4 PBCR (端口 B 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
复位后	0	0	1	0	1	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PB7C ~ PB0C	R/W	输出 0：禁用 1：启用

7.2.2.5 PBFR1 (端口 B 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7F1	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
复位后	1	1	1	1	1	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PB7F1	R/W	0：端口 1：$\overline{\text{TRST}}$
6	PB6F1	R/W	0：端口 1：TDI
5	PB5F1	R/W	0：端口 1：TDO / SWV
4	PB4F1	R/W	0：端口 1：TCK / SWCLK
3	PB3F1	R/W	0：端口 1：TMS / SWDIO
2	PB2F1	R/W	0：端口 1：TRACEDATA1
1	PB1F1	R/W	0：端口 1：TRACEDATA0
0	PB0F1	R/W	0：端口 1：TRACECLK

7.2.2.6 PBOD (端口 B 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7OD	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-0	PB7OD ~ PB0OD	R/W	0：CMOS 1：开漏

7.2.2.7 PBPUP (端口 B 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7UP	PB6UP	PB5UP	PB4UP	PB3UP	PB2UP	PB1UP	PB0UP
复位后	1	1	0	0	1	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-0	PB7UP ~ PB0UP	R/W	上拉 0：禁用 1：启用

7.2.2.8 PBPDN (端口 B 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7DN	PB6DN	PB5DN	PB4DN	PB3DN	PB2DN	PB1DN	PB0DN
复位后	0	0	0	1	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PB7DN ~ PBDN	R/W	下拉 0：禁用 1：启用

7.2.2.9 PBIE (端口 B 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PB7IE	PB6IE	PB5IE	PB4IE	PB3IE	PB2IE	PB1IE	PB0IE
复位后	1	1	0	1	1	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PB7IE ~ PB0IE	R/W	输入 0：禁用 1：启用

7.2.3 端口 C (PC0 ~ PC7)

端口 C 是一个通用的 8 位输入/输出端口。可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 C 还作为三相马达控制(PMD)的输入/输出端口。

复位使端口 C 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

7.2.3.1 端口 C 电路类型

	7	6	5	4	3	2	1	0
型号	T3	T3	T1	T1	T1	T1	T1	T1

7.2.3.2 端口 C 寄存器

基址 = 0x4000_0080

寄存器名称		地址(基本+)
端口 C 数据寄存器	PCDATA	0x0000
端口 C 输出控制寄存器	PCCR	0x0004
端口 C 功能寄存器 1	PCFR1	0x0008
端口 C 开漏控制寄存器	PCOD	0x0028
端口 C 上拉控制寄存器	PCPUP	0x002C
端口 C 下拉控制寄存器	PCPDN	0x0030
端口 C 输入控制寄存器	PCIE	0x0038

7.2.3.3 PCDATA (端口 C 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PC7 ~ PC0	R/W	端口 C 数据寄存器

7.2.3.4 PCCR (端口 C 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7C	PC6C	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PC7C ~ PC0C	R/W	输出 0：禁用 1：启用

7.2.3.5 PCFR1 (端口 C 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7F1	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PC7F1	R/W	0：端口 1：$\overline{\text{OVV0}}$
6	PC6F1	R/W	0：端口 1：$\overline{\text{EMG0}}$
5	PC5F1	R/W	0：端口 1：ZO0
4	PC4F1	R/W	0：端口 1：WO0
3	PC3F1	R/W	0：端口 1：YO0
2	PC2F1	R/W	0：端口 1：VO0
1	PC1F1	R/W	0：端口 1：XO0
0	PC0F1	R/W	0：端口 1：UO0

7.2.3.6 PCOD (端口 C 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7OD	PC6OD	PC5OD	PC4OD	PC3OD	PC2OD	PC1OD	PC0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PC7OD ~ PC0OD	R/W	0：CMOS 1：开漏

7.2.3.7 PCPUP (端口 C 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7UP	PC6UP	PC5UP	PC4UP	PC3UP	PC2UP	PC1UP	PC0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PC7UP ~ PC0UP	R/W	上拉 0：禁用 1：启用

7.2.3.8 PCPDN (端口 C 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7DN	PC6DN	PC5DN	PC4DN	PC3DN	PC2DN	PC1DN	PC0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PC7DN ~ PC0DN	R/W	下拉 0：禁用 1：启用

7.2.3.9 PCIE (端口 C 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE	PC0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PC7IE ~ PC0IE	R/W	输入 0：禁用 1：启用

7.2.4 端口 D (PD0 ~ PD6)

端口 D 是一个通用的 7-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 D 还执行串行接口功能(SIO/UART)，外部信号中断输入，16-位定时器输入/输出功能和编码器输入功能。

复位使端口 D 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

端口 D 有两类功能寄存器。当端口 D 用作通用端口时，将两个寄存器的对应位设置为"0"。当将端口 D 用作非通用端口时，将功能寄存器的对应位设置为"1"。不要将一些功能寄存器同时设置为"1"。

使用外部中断输入释放 STOP 模式时，在 PDFR1 中选择该功能，并在 PDIE 寄存器中启用输入。 即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

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

7.2.4.1 端口 D 电路类型

	7	6	5	4	3	2	1	0
型号	-	T3	T2	T9	T4	T3	T10	T11

7.2.4.2 端口 D 寄存器

基址= 0x4000_00C0

寄存器名称		地址(基本+)
端口 D 数据寄存器	PDDATA	0x0000
端口 D 输出控制寄存器	PDCR	0x0004
端口 D 功能寄存器 1	PDFR1	0x0008
端口 D 功能寄存器 2	PDFR2	0x000C
端口 D 开漏控制寄存器	PDOD	0x0028
端口 D 上拉控制寄存器	PDPUP	0x002C
端口 D 下拉控制寄存器	PDPDN	0x0030
端口 D 输入控制寄存器	PDIE	0x0038

7.2.4.3 PDDATA (端口 D 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6	PD5	PD4	PD3	PD2	PD1	PD0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	−	R	读作 0。
6-0	PD6 ~ PD0	R/W	端口 D 数据寄存器

7.2.4.4 PDCR (端口 D 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	−	R	读作 0。
6-0	PD6C ~ PD0C	R/W	输出 0：禁用 1：启用

7.2.4.5 PDFR1 (端口 D 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6F1	PD5F1	PD4F1	PD3F1	PD2F1	PD1F1	PD0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	–	R	读作 0。
6	PD6F1	R/W	0：PORT 1：RXD2
5	PD5F1	R/W	0：PORT 1：TXD2
4	PD4F1	R/W	0：PORT 1：SCLK2
3	PD3F1	R/W	0：PORT 1： INT9
2	PD2F1	R/W	0：PORT 1：ENCZ0
1	PD1F1	R/W	0：PORT 1：ENCB0
0	PD0F1	R/W	0：PORT 1：ENCA0

7.2.4.6 PDFR2 (端口 D 功能寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PD4F2	-	-	PD1F2	PD0F2
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4	PD4F2	R/W	0：PORT 1：$\overline{\text{CTS2}}$
3-2	−	R	读作 0。
1	PD1F2	R/W	0：PORT 1：TB5OUT
0	PD0F2	R/W	0：PORT 1：TB5IN

7.2.4.7 PDOD (端口 D 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6OD	PD5OD	PD4OD	PD3OD	PD2OD	PD1OD	PD0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	–	R	读作 0。
6-0	PD6OD ~ PD0OD	R/W	0：CMOS 1：开漏

7.2.4.8 PDPUP (端口 D 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6UP	PD5UP	PD4UP	PD3UP	PD2UP	PD1UP	PD0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	−	R	读作 0。
6-0	PD6UP ~ PD0UP	R/W	上拉 0：禁用 1：启用

7.2.4.9 PDPDN (端口 D 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6DN	PD5DN	PD4DN	PD3DN	PD2DN	PD1DN	PD0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	−	R	读作 0。
6-0	PD6DN ~ PD0DN	R/W	下拉 0：禁用 1：启用

7.2.4.10 PDIE (端口 D 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	PD6IE	PD5IE	PD4IE	PD3IE	PD2IE	PD1IE	PD0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-7	–	R	读作 0。
6-0	PD6IE ~ PD0IE	R/W	输入 0：禁用 1：启用

7.2.5 端口 E (PE0 ~ PE7)

端口 E 是一个通用的 8-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 E 还执行串行接口功能(SIO/UART)，外部信号中断输入，16-位定时器输入/输出功能。

复位使端口 E 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

端口 E 有两类功能寄存器。当将端口 E 用作通用端口时，将两个寄存器的对应位设置为"0"。当将端口 E 用作非通用端口，将功能寄存器的对应位设置为"1"。不要将一些功能寄存器同时设置为"1"。

当使用外部中断输入释放 STOP 模式时，在 PEFR2 中选择该功能，并在 PEIE 寄存器中启用输入。即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

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

7.2.5.1 端口 E 电路类型

	7	6	5	4	3	2	1	0
型号	T14	T12	T2	T12	T2	T9	T3	T2

7.2.5.2 端口 E 寄存器

基址= 0x4000_0100

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

7.2.5.3 PEDATA (端口 E 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7	PE6				PE2	PE1	PE0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PE7 ~ PE0	R/W	端口 E 数据寄存器

7.2.5.4 PECR (端口 E 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PE7C ~ PE0C	R/W	输出 0：禁用 1：启用

7.2.5.5 PEFR1 (端口 E 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7F1	PE6F1	PE5F1	PE4F1	PE3F1	PE2F1	PE1F1	PE0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PE7F1	R/W	0：PORT 1：TB3OUT
6	PE6F1	R/W	0：PORT 1:TB3IN
5	PE5F1	R/W	0：PORT 1：TB2OUT
4	PE4F1	R/W	0：PORT 1：TB2IN
3	PE3F1	R/W	0：PORT 1：TB4OUT
2	PE2F1	R/W	0：PORT 1：SCLK0
1	PE1F1	R/W	0：PORT 1：RXD0
0	PE0F1	R/W	0：PORT 1：TXD0

7.2.5.6 PEFR2 (端口 E 功能寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7F2	PE6F2	-	PE4F2	-	PE2F2	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PE7F2	R/W	0：PORT 1：INT7
6	PE6F2	R/W	0：PORT 1：INT6
5	−	R	读作 0。
4	PE4F2	R/W	0：PORT 1：INT5
3	−	R	读作 0。
2	PE2F2	R/W	0：PORT 1：$\overline{\text{CTS0}}$
1-0	−	R	读作 0。

7.2.5.7 PEOD (端口 E 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7OD	PE6OD	PE5OD	PE4OD	PE3OD	PE2OD	PE1OD	PE0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PE7OD ~ PE0OD	R/W	0：CMOS 1：开漏

7.2.5.8 PEPUP (端口 E 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7UP	PE6UP	PE5UP	PE4UP	PE3UP	PE2UP	PE1UP	PE0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PE7UP ~ PE0UP	R/W	上拉 0：禁用 1：启用

7.2.5.9 PEPDN (端口 E 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7DN	PE6DN	PE5DN	PE4DN	PE3DN	PE2DN	PE1DN	PE0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PE7DN ~ PE0DN	R/W	下拉 0：禁用 1：启用

7.2.5.10 PEIE (端口 E 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PE7IE	PE6IE	PE5IE	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PE7IE ~ PE0IE	R/W	输入 0：禁用 1：启用

7.2.6 端口 F (PF0 ~ PF4)

端口 F 是一个通用的 5-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 E 还执行串行接口功能(SIO/UART)，16-位定时器输入/输出功能，编码器输入功能和运行模式设置。

当复位信号在" 0 "状态时,，启用 PF0 输入和上拉启用。在复位信号上升沿，若 PF0 为"1"，则装置进入单核模式，并从芯片上的闪存存储器启动。若 PF0 为"0"，则装置进入单核启动模式，并从内部启动程序启动。单核启动模式详见"闪存存储器操作"。

复位使端口 F 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

7.2.6.1 端口 F 电路类型

	7	6	5	4	3	2	1	0
型号	-	-	-	T11	T10	T15	T2	T20

7.2.6.2 端口 F 寄存器

基址= 0x4000_0140

寄存器名称		地址(基本+)
端口 F 数据寄存器	PFDATA	0x0000
端口 F 输出控制寄存器	PFCR	0x0004
端口 F 功能寄存器 1	PFFR1	0x0008
端口 F 功能寄存器 2	PFFR2	0x000C
端口 F 功能寄存器 3	PFFR3	0x0010
端口 F 开漏控制寄存器	PFOD	0x0028
端口 F 上拉控制寄存器	PFPUP	0x002C
端口 F 下拉控制寄存器	PFPDN	0x0030
端口 F 输入控制寄存器	PFIE	0x0038

7.2.6.3 PFDATA (端口 F 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4	PF3	PF2	PF1	PF0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4-0	PF4 ~ PF0	R/W	端口 F 数据寄存器

7.2.6.4 PFCR (端口 F 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4C	PF3C	PF2C	PF1C	PF0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4-0	PF4C ~ PF0C	R/W	输出 0：禁用 1：启用

7.2.6.5 PFFR1 (端口 F 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4F1	PF3F1	PF2F1	PF1F1	PF0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4	PF4F1	R/W	0：PORT 1：ENCZ1
3	PF3F1	R/W	0：PORT 1：ENCB1
2	PF2F1	R/W	0：PORT 1：ENCA1
1	PF1F1	R/W	0：PORT 1：TB7OUT
0	PF0F1	R/W	0：PORT 1：TB7IN

7.2.6.6 PFFR2 (端口 F 功能寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4F2	PF3F2	PF2F2	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4	PF4F2	R/W	0：PORT 1：RXD3
3	PF3F2	R/W	0：PORT 1：TXD3
2	PF2F2	R/W	0：PORT 1：SCLK3
1-0	−	R	读作 0。

7.2.6.7 PFFR3 (端口 F 功能寄存器 3)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	PF2F3	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-3	−	R	读作 0。
2	PF2F3	R/W	0：PORT 1：$\overline{\text{CTS3}}$
1-0	−	R	读作 0。

7.2.6.8 PFOD (端口 F 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4OD	PF3OD	PF2OD	PF1OD	PF0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4-0	PF4OD ~ PF0OD	R/W	0：CMOS 1：开漏

7.2.6.9 PFPUP (端口 F 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4UP	PF3UP	PF2UP	PF1UP	PF0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4-0	PF4UP ~ PF0UP	R/W	上拉 0：禁用 1：启用

7.2.6.10 PFPDN (端口 F 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4DN	PF3DN	PF2DN	PF1DN	PF0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4-0	PF4DN ~ PF0DN	R/W	下拉 0：禁用 1：启用

7.2.6.11 PFIE (端口 F 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-5	−	R	读作 0。
4-0	PF4IE~PF0IE	R/W	输入 0：禁用 1：启用

7.2.7 端口 G (PG0 ~ PG7)

端口 G 是一个通用的 8-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 G 还作为三相马达控制 (PMD)功能的输入/输出端口。

复位使端口 G 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

7.2.7.1 端口 G 电路类型

	7	6	5	4	3	2	1	0
型号	T3	T3	T1	T1	T1	T1	T1	T1

7.2.7.2 端口 G 寄存器

基址= 0x4000_0180

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

7.2.7.3 PGDATA (端口 G 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PG7 ~ PG0	R/W	端口 G 数据寄存器

7.2.7.4 PGCR (端口 G 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7C	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PG7C ~ PG0C	R/W	输出 0：禁用 1：启用

7.2.7.5 PGFR1 (端口 G 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7F1	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PG7F1	R/W	0：PORT 1: $\overline{\text{OVV1}}$
6	PG6F1	R/W	0：PORT 1：$\overline{\text{EMG1}}$
5	PG5F1	R/W	0：PORT 1：ZO1
4	PG4F1	R/W	0：PORT 1：WO1
3	PG3F1	R/W	0：PORT 1：YO1
2	PG2F1	R/W	0：PORT 1：VO1
1	PG1F1	R/W	0：PORT 1：XO1
0	PG0F1	R/W	0：PORT 1：UO1

7.2.7.6 PGOD (端口 G 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7OD	PG6OD	PG5OD	PG4OD	PG3OD	PG2OD	PG1OD	PG0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PG7OD ~ PG0OD	R/W	0：CMOS 1：开漏

7.2.7.7 PGPUP (端口 G 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7UP	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PG7UP ~ PG0UP	R/W	上拉 0：禁用 1：启用

7.2.7.8 PGPDN (端口 G 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7DN	PG6DN	PG5DN	PG4DN	PG3DN	PG2DN	PG1DN	PG0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PG7DN ~ PG0DN	R/W	下拉 0：禁用 1：启用

7.2.7.9 PGIE (端口 G 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PG7IE	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PG7IE ~ PG0IE	R/W	输入 0：禁用 1：启用

7.2.8 端口 H (PH0 ~ PH7)

端口 H 是一个通用的 8-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 H 还执行 AD 转换器的模拟输入和外部信号中断输入。

复位使端口 H 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

当使用外部中断输入释放 STOP 模式时，在 PHFR1 中选择该功能，并在 PHIE 寄存器中启用输入。即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

注 1：除非将端口 H 的所有位数用作模拟输入引脚，否则可降低转换精确度。应验证这样做不会对系统造成问题。

注 2：在非 STOP 模式时，若输入在 PxIE 中启用，则不管 PxFR 寄存器的设置，中断输入仍会启用。当给装置编程时，保证禁用不用的中断。

7.2.8.1 端口 H 电路类型

	7	6	5	4	3	2	1	0
型号	T16	T16	T16	T16	T16	T17	T17	T17

7.2.8.2 端口 H 寄存器

基址= 0x4000_01C0

寄存器名称		地址(基本+)
端口 H 数据寄存器	PHDATA	0x0000
端口 H 输出控制寄存器	PHCR	0x0004
端口 H 功能寄存器 1	PHFR1	0x0008
端口 H 开漏控制寄存器	PHOD	0x0028
端口 H 上拉控制寄存器	PHPUP	0x002C
端口 H 下拉控制寄存器	PHPDN	0x0030
端口 H 输入控制寄存器	PHIE	0x0038

7.2.8.3 PHDATA (端口 H 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PH7 ~ PH0	R/W	端口 H 数据寄存器

7.2.8.4 PHCR (端口 H 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PH7C ~ PH0C	R/W	输出 0：禁用 1：启用

7.2.8.5 PHFR1 (端口 H 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	PH2F1	PH1F1	PH0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-3	–	R	读作 0。
2	PH2F1	R/W	0：PORT 1：INT2
1	PH1F1	R/W	0：PORT 1：INT1
0	PH0F1	R/W	0：PORT 1：INT0

7.2.8.6 PHOD (端口 H 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PH7OD	PH6OD	PH5OD	PH4OD	PH3OD	PH2OD	PH1OD	PH0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PH7OD ~ PH0OD	R/W	0：CMOS 1：开漏

7.2.8.7 PHPUP (端口 H 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PH7UP	PH6UP	PH5UP	PH4UP	PH3UP	PH2UP	PH1UP	PH0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PH7UP ~ PH0UP	R/W	上拉 0：禁用 1：启用

7.2.8.8 PHPDN (端口 H 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PH7DN	PH6DN	PH5DN	PH4DN	PH3DN	PH2DN	PH1DN	PH0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PH7DN ~ PH0DN	R/W	下拉 0：禁用 1：启用

7.2.8.9 PHIE (端口 H 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PH7IE	PH6IE	PH5IE	PH4IE	PH3IE	PH2IE	PH1IE	PH0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PH7IE ~ PH0IE	R/W	输入 0：禁用 1：启用

7.2.9 端口 I (PI0 ~ PI3)

端口 I 是一个通用的 4-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 I 还执行 AD 转换器的模拟输入。

复位使端口 I 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

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

7.2.9.1 端口 I 电路类型

	7	6	5	4	3	2	1	0
型号	-	-	-	-	T16	T16	T16	T16

7.2.9.2 端口 I 寄存器

基址= 0x4000_0200

寄存器名称		地址(基本+)
端口 I 数据寄存器	PIDATA	0x0000
端口 I 输出控制寄存器	PICR	0x0004
端口 I 开漏控制寄存器	PIOD	0x0028
端口 I 上拉控制寄存器	PIPUP	0x002C
端口 I 下拉控制寄存器	PIPDN	0x0030
端口 I 输入控制寄存器	PIIE	0x0038

7.2.9.3 PIDATA (端口 I 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	PI3	PI2	PI1	PI0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	PI3 ~ PI0	R/W	端口 I 数据寄存器

7.2.9.4 PICR (端口 I 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	PI3C	PI2C-	PI1C-	PI0C-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	PI3C ~ PI0C	R/W	输出 0：禁用 1：启用

7.2.9.5 PIOD (端口 I 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	PI3OD	PI2OD	PI1OD	PI0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	PI3OD-PI0OD	R/W	0：CMOS 1：开漏

7.2.9.6 PIPUP (端口 I 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	PI3UP	PI2UP	PI1UP	PI0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	PI3UP ~ PI0UP	R/W	上拉 0：禁用 1：启用

7.2.9.7 PIPDN (端口 I 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	PI3DN	PI2DN	PI1DN	PI0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	PI3DN ~ PI0DN	R/W	下拉 0：禁用 1：启用

7.2.9.8 PIIE (端口 I 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	PI3IE	PI2IE	PI1IE	PI0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	PI3IE ~ PI0IE	R/W	输入 0：禁用 1：启用

7.2.10 端口 J (PJ0 ~ PJ7)

端口 J 是一个通用的 8-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 J 还执行 AD 转换器的模拟输入和外部信号中断输入。

复位使端口 J 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

使用外部中断输入释放 STOP 模式时，在 PJFR1 中选择该功能，并在 PJIE 寄存器中启用输入。

即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

注 1：除非将端口 J 的所有位数用作模拟输入引脚，否则可降低转换精确度。 验证这样做不会对系统造成问题。

注 2：在非 STOP 模式时，若输入在 PxIE 中启用，则不管 PxFR 寄存器的设置，中断输入仍会启用。当给装置编程时，保证禁用不用的中断。

7.2.10.1 端口 J 电路类型

	7	6	5	4	3	2	1	0
型号	T17	T17	T16	T16	T16	T16	T16	T16

7.2.10.2 端口 J 寄存器

基址= 0x4000_0240

寄存器名称		地址(基本+)
端口 J 数据寄存器	PJDATA	0x0000
端口 J 输出控制寄存器	PJCR	0x0004
端口 J 功能寄存器 1	PJFR1	0x0008
端口 J 开漏控制寄存器	PJOD	0x0028
端口 J 上拉控制寄存器	PJPUP	0x002C
端口 J 下拉控制寄存器	PJPDN	0x0030
端口 J 输入控制寄存器	PJIE	0x0038

7.2.10.3 PJDATA (端口 J 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PJ7 ~ PJ0	R/W	端口 J 数据寄存器

7.2.10.4 PJCR (端口 J 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PJ7C ~ PJ0C	R/W	输出 0：禁用 1：启用

7.2.10.5 PJFR1 (端口 J 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7F1	PJ6F1	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	PJ7F1	R/W	0：PORT 1：INTD
6	PJ6F1	R/W	0：PORT 1：INTC
5-0	−	R	读作 0。

7.2.10.6 PJOD (端口 J 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7OD	PJ6OD	PJ5OD	PJ4OD	PJ3OD	PJ2OD	PJ1OD	PJ0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PJ7OD ~ PJ0OD	R/W	0：CMOS 1：开漏

7.2.10.7 PJPUP (端口 J 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7UP	PJ6UP	PJ5UP	PJ4UP	PJ3UP	PJ2UP	PJ1UP	PJ0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PJ7UP ~ PJ0UP	R/W	上拉 0：禁用 1：启用

7.2.10.8 PJPDN (端口 J 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7DN	PJ6DN	PJ5DN	PJ4DN	PJ3DN	PJ2DN	PJ1DN	PJ0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PJ7DN-PJ0DN	R/W	下拉 0：禁用 1：启用

7.2.10.9 PJIE (端口 J 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PJ7IE	PJ6IE	PJ5IE	PJ4IE	PJ3IE	PJ2IE	PJ1IE	PJ0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	PJ7IE ~ PJ0IE	R/W	输入 0：禁用 1：启用

7.2.11 端口 K (PK0 ~ PK1)

端口 K 是一个通用的 2-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 K 还执行 AD 转换器的模拟输入和外部信号中断输入。

复位使端口 K 的所有位数初始化为通用端口，输入，输出，上拉和下拉禁用。

使用外部中断输入释放 STOP 模式时，在 PKFR1 中选择该功能，并在 PKIE 寄存器中启用输入。即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

注 1：除非将端口 K 的所有位数用作模拟输入引脚，否则可降低转换精确度。验证这样做不会对系统造成问题。

注 2：在非 STOP 模式时，若输入在 PxIE 中启用，则不管 PxFR 寄存器的设置，中断输入仍会启用。当给装置编程时，保证禁用不用的中断。

7.2.11.1 端口 K 电路类型

	7	6	5	4	3	2	1	0
型号	-	-	-	-	-	-	T17	T17

7.2.11.2 端口 K 寄存器

基址= 0x4000_0280

寄存器名称		地址(基本+)
端口 K 数据寄存器	PKDATA	0x0000
端口 K 输出控制寄存器	PKCR	0x0004
端口 K 功能寄存器 1	PKFR1	0x0008
端口 K 开漏控制寄存器	PKOD	0x0028
端口 K 上拉控制寄存器	PKPUP	0x002C
端口 K 下拉控制寄存器	PKPDN	0x0030
端口 K 输入控制寄存器	PKIE	0x0038

7.2.11.3 PKDATA (端口 K 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PK1	PK0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PK1 ~ PK0	R/W	端口 K 数据寄存器

7.2.11.4 PKCR (端口 K 输出控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-				PK1C	PK0C
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PK1C ~ PK0C	R/W	输出 0：禁用 1：启用

7.2.11.5 PKFR1 (端口 K 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PK1F1	PK0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	PK1F1	R/W	0：PORT 1：INTF
0	PK0F1	R/W	0：PORT 1：INTE

7.2.11.6 PKOD (端口 K 开漏控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PK1OD	PK0OD
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PK1OD ~ PK0OD	R/W	0：CMOS 1：开漏

7.2.11.7 PKPUP (端口 K 上拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PK1UP	PK0UP
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PK1UP ~ PK0UP	R/W	上拉 0：禁用 1：启用

7.2.11.8 PKPDN (端口 K 下拉控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PK1DN	PK0DN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PK1DN- PK0DN	R/W	下拉 0：禁用 1：启用

7.2.11.9 PKIE (端口 K 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PK1IE	PK0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	–	R	读作 0。
1-0	PK1IE-PK0IE	R/W	输入 0：禁用 1：启用

7.2.12 端口 L (PL0 ~ PL1)

端口 L 是一个通用的 2-位输入/输出端口。 可规定该端口的输入和输出，单位为位。除通用的输入/输出功能外，端口 L 还执行外部信号中断输入。

复位使端口 L 的所有位数初始化为通用端口，输入禁用。

使用外部中断输入释放 STOP 模式时，在 PLFR1 中选择该功能，并在 PLIE 寄存器中启用输入。 即使在 STOP 模式时在时钟/模式控制块中设置了 CGSTBYCR<DRVE>位，以停止引脚的驱动，这些设置仍启用中断输入。

注 1：在非 STOP 模式时，若输入在 PxIE 中启用，则不管 PxFR 寄存器的设置，中断输入仍会启用。当给装置编程时，保证禁用不用的中断。

注 2：当打开电源时，请保持端口 L 的"低"电平，恒定时间(包括在复位时间中)。详见"电气特性"的"电源注意事项"。

7.2.12.1 端口 L 电路类型

	7	6	5	4	3	2	1	0
型号	-	-	-	-	-	-	T5	T5

7.2.12.2 端口 L 寄存器

基址= 0x4000_02C0

寄存器名称		地址(基本+)
端口 L 数据寄存器	PLDATA	0x0000
端口 L 功能寄存器 1	PLFR1	0x0008
端口 L 输入控制寄存器	PLIE	0x0038

7.2.12.3 PLDATA (端口 L 数据寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PL1	PL0
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PL1 ~ PL0	R/W	端口 L 数据寄存器

7.2.12.4 PLFR1 (端口 L 功能寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PL1F1	PL0F1
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	PL1F1	R/W	0：PORT 1：INTA
0	PL0F1	R/W	0：PORT 1：INTB

7.2.12.5 PLIE (端口 L 输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	PL1IE	PL0IE
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-2	−	R	读作 0。
1-0	PL1IE ~ PL0IE	R/W	输入 0：禁用 1：启用

7.3 端口方块图

7.3.1 端口类型

端口分类如下。请参考下列各页的端口类型方块图。

图中虚线指"端口方块图"所述等效电路部分。

表 7-3 功能表

型号	GP 端口	功能 1	功能 2	功能 3	模拟	上拉	下拉	可编程开漏	注
T1	I/O	输出	-	-	-	R	R	o	启用信号触发的功能输出
T2	I/O	输出	-	-	-	R	R	o	
T3	I/O	输入	-	-	-	R	R	o	
T4	I/O	输入(int)	-	-	-	R	R	o	
T5	输入	输入(int)	-	-	-	-	-	-	
T6	I/O	I/O	-	-	-	NoR	-	-	启用信号触发的功能输出
T7	I/O	输入	-	-	-	NoR	-	-	
T8	I/O	输入	-	-	-	-	NoR	-	
T9	I/O	I/O	输入	-	-	R	R	o	
T10	I/O	输入	输出	-	-	R	R	o	
T11	I/O	输入	输入	-	-	R	R	o	
T12	I/O	输入	输入(int)	-	-	R	R	o	
T13	I/O	输出	输出	-	-	R	R	o	
T14	I/O	输出	I/O	-	-	R	R	o	
T15	I/O	输入	I/O	输入	-	R	R	o	
T16	I/O	-	-	-	o	R	R	o	
T17	I/O	输入(int)	-	-	o	R	R	o	
T18	I/O	输出	-	-	-	R	-	-	
T19	I/O	输出	-	-	-	NoR	-	-	启用信号触发的功能输出
T20	I/O	输入	-	-	-	NoR	NoR	o	在复位时启用的 $\overline{\text{BOOT}}$ 输入
T21	I/O	-(OSC1)	-	-	-	R	R	o	(外部)高速振荡器

int: 中断输入　　R: 在复位时强制的禁用

-: 不存在　　NoR: 不受复位的影响

o: 存在

7.3.2 T1 类

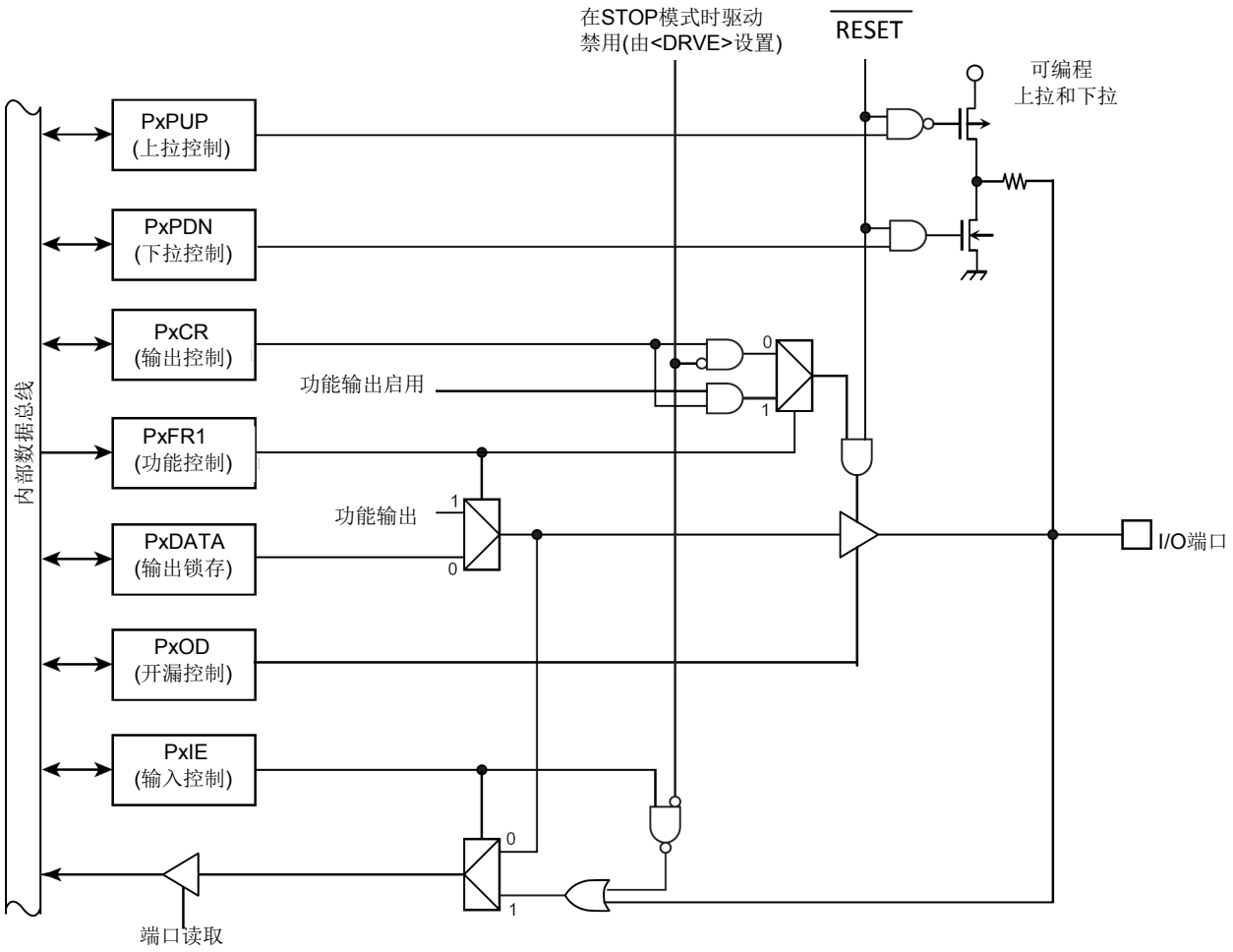


图 7-1 T1 端口类型

7.3.3 T2 类

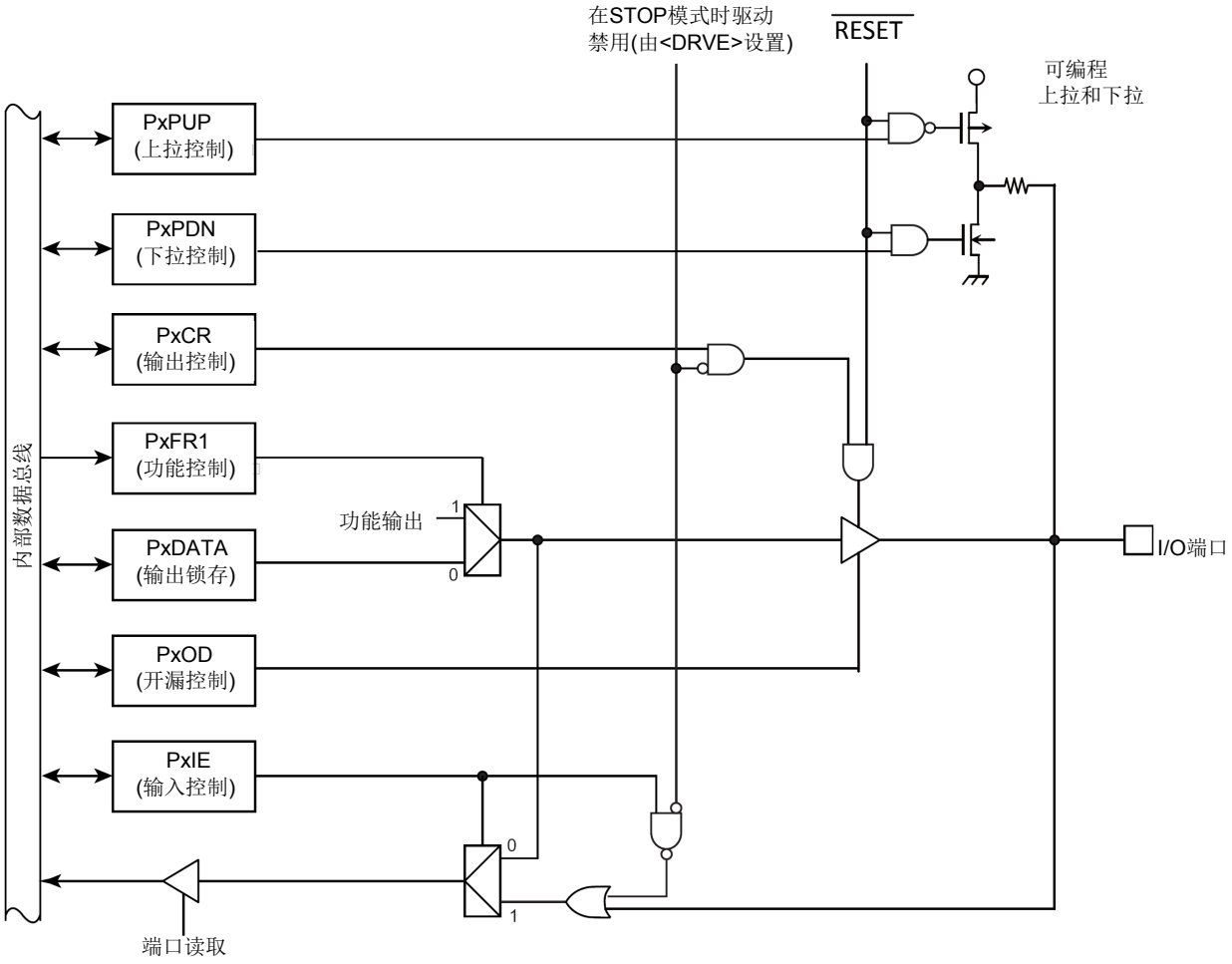


图 7-2 T2 端口类型

7.3.4 T3 类

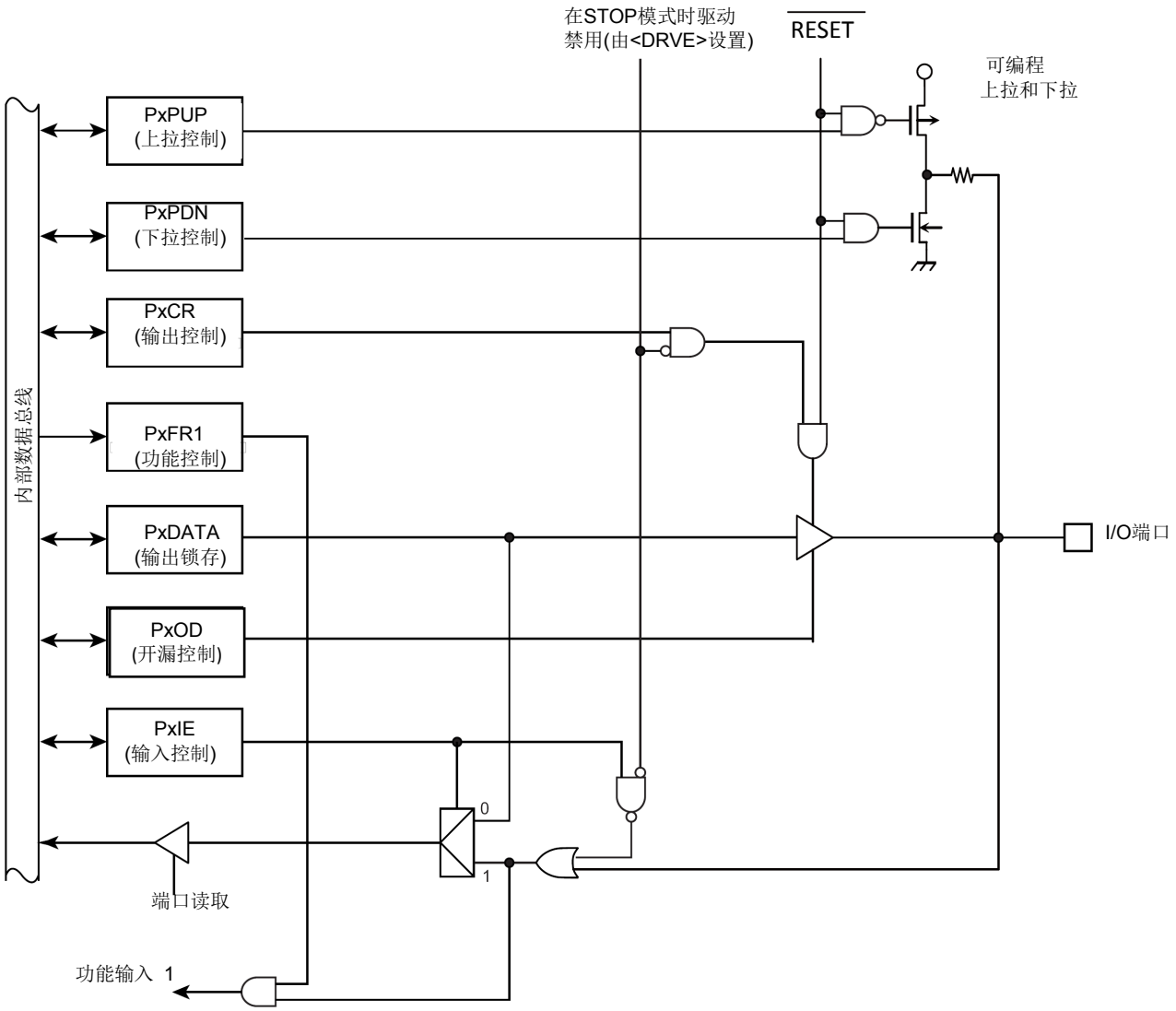


图 7-3 T3 端口类型

7.3.5 T4 类

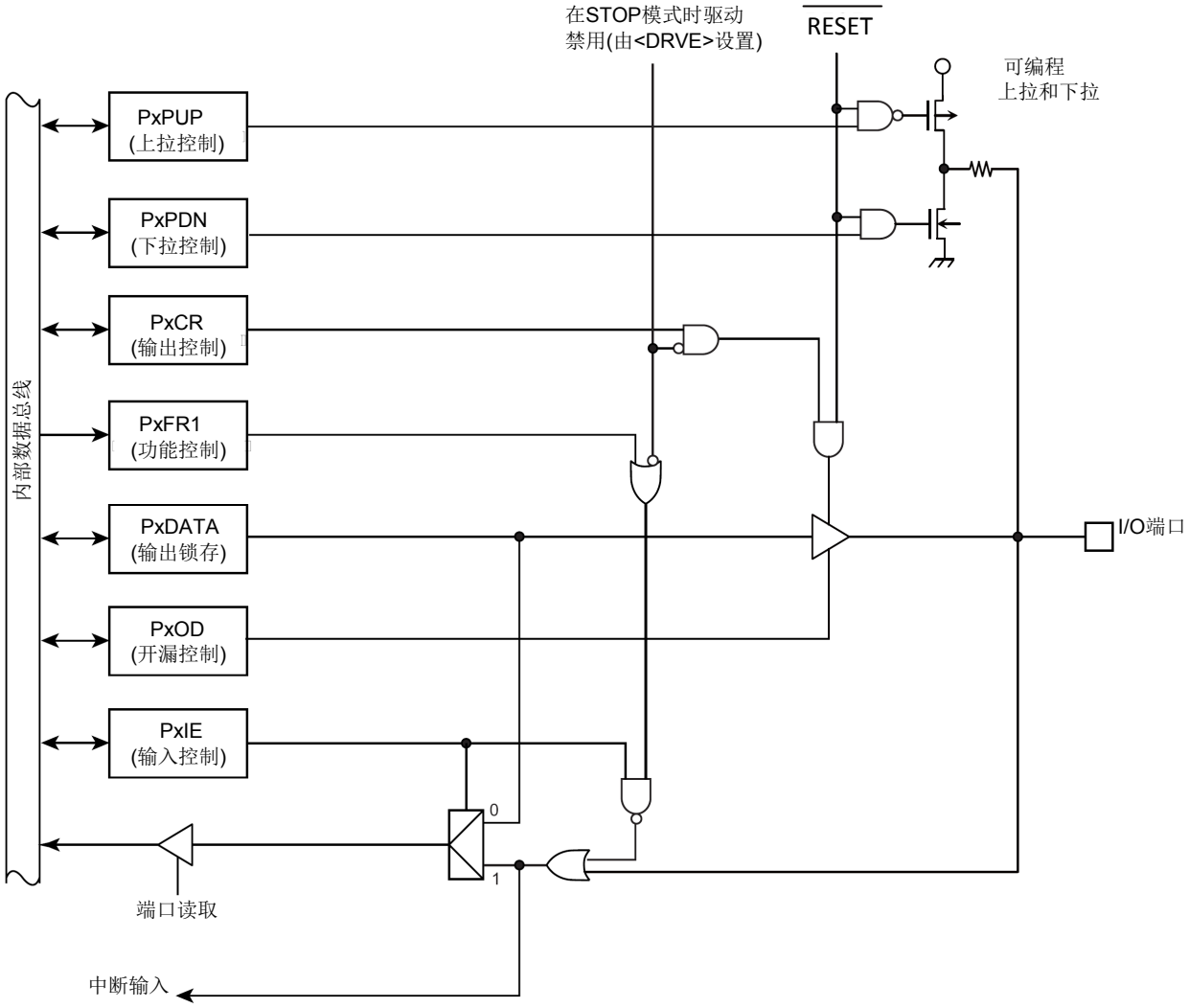


图 7-4 T4 端口类型

7.3.6 T5 类

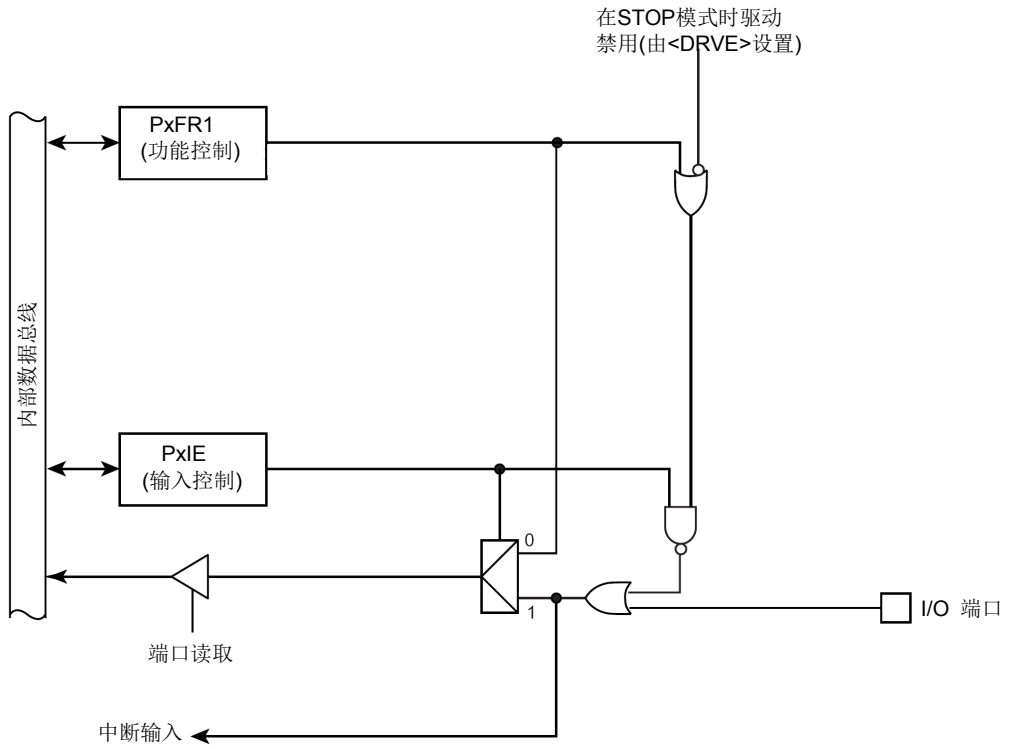


图 7-5 T5 端口类型

7.3.7 T6 类

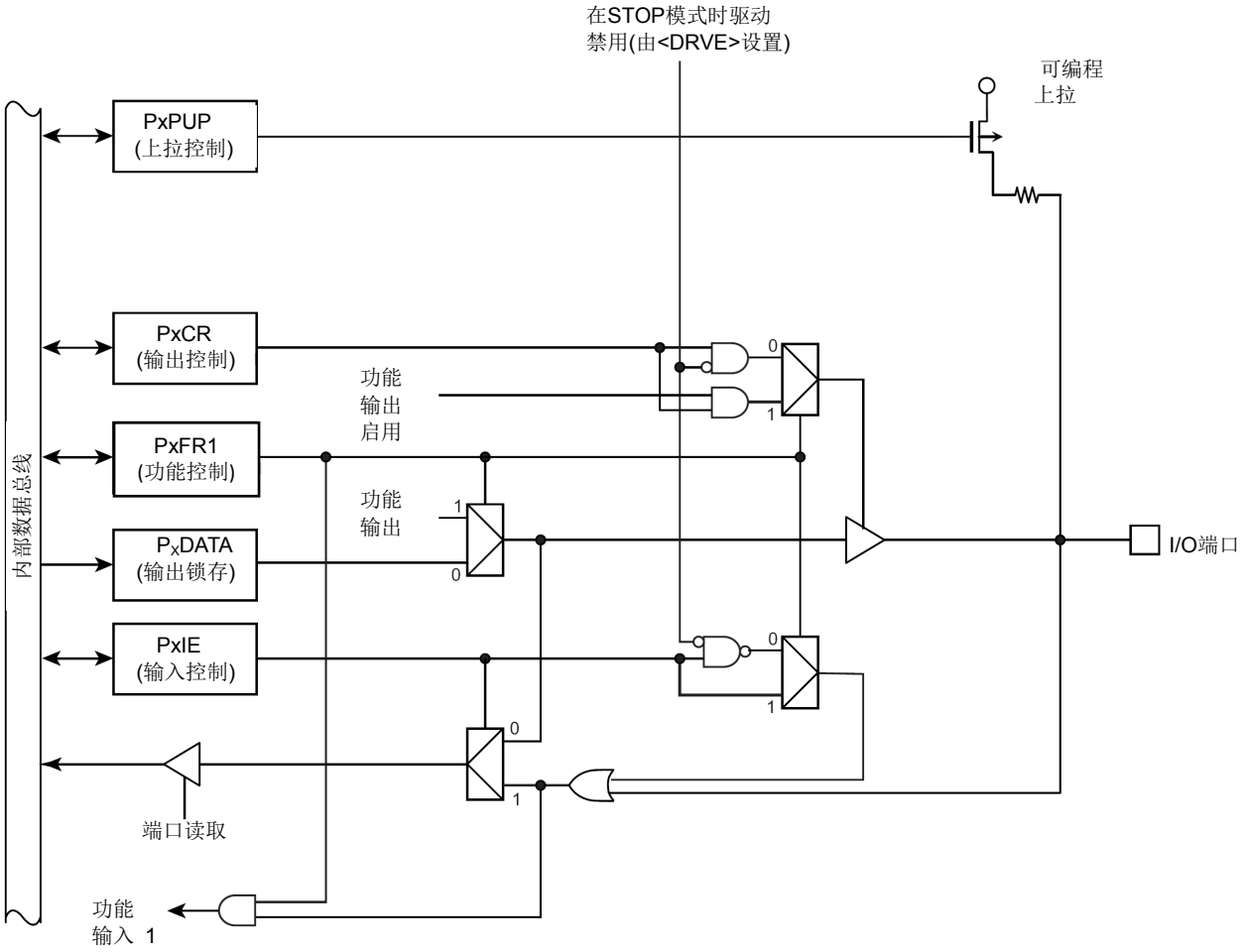


图 7-6 T6 端口类型

7.3.8 T7 类

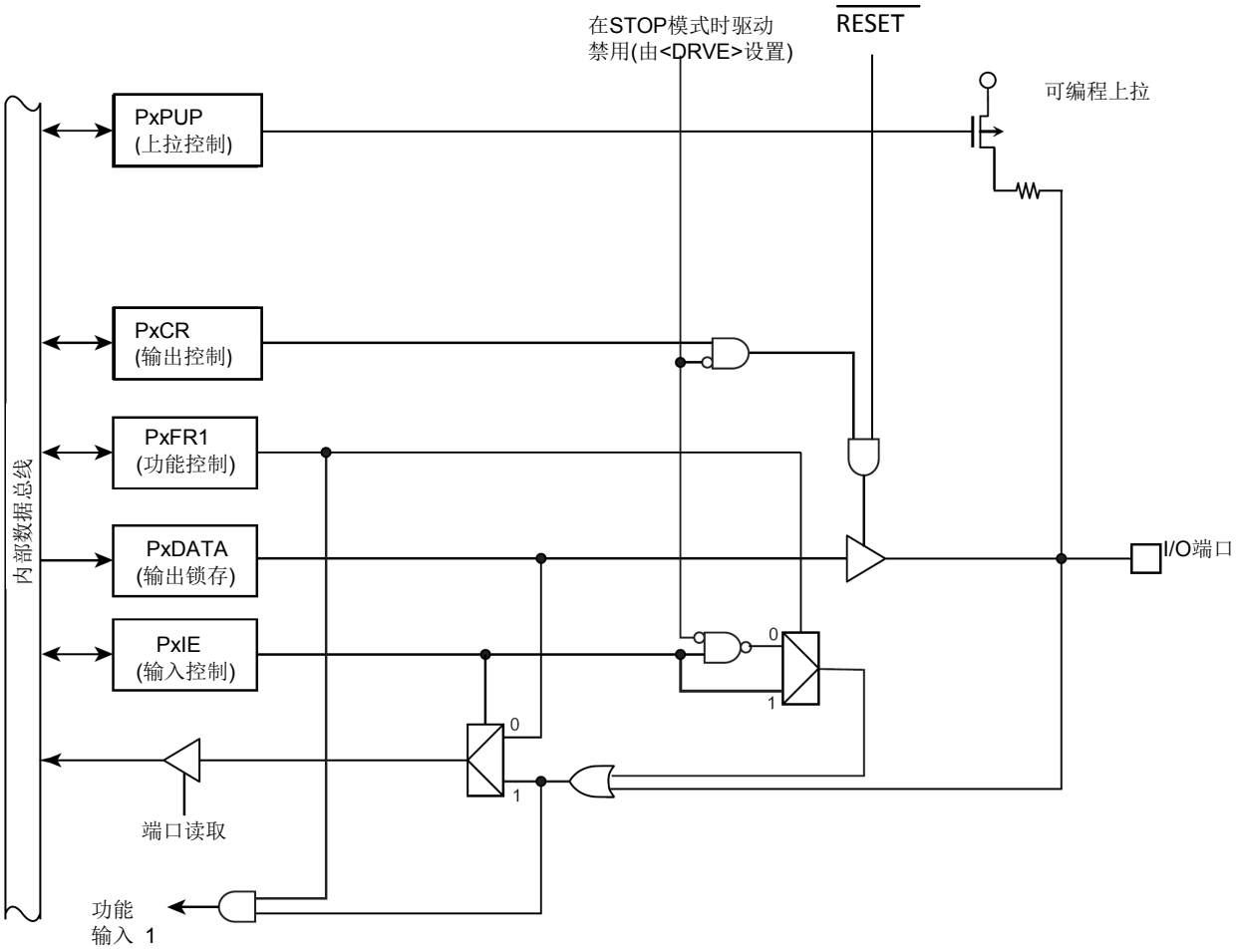


图 7-7 T7 端口类型

7.3.9 T8 类

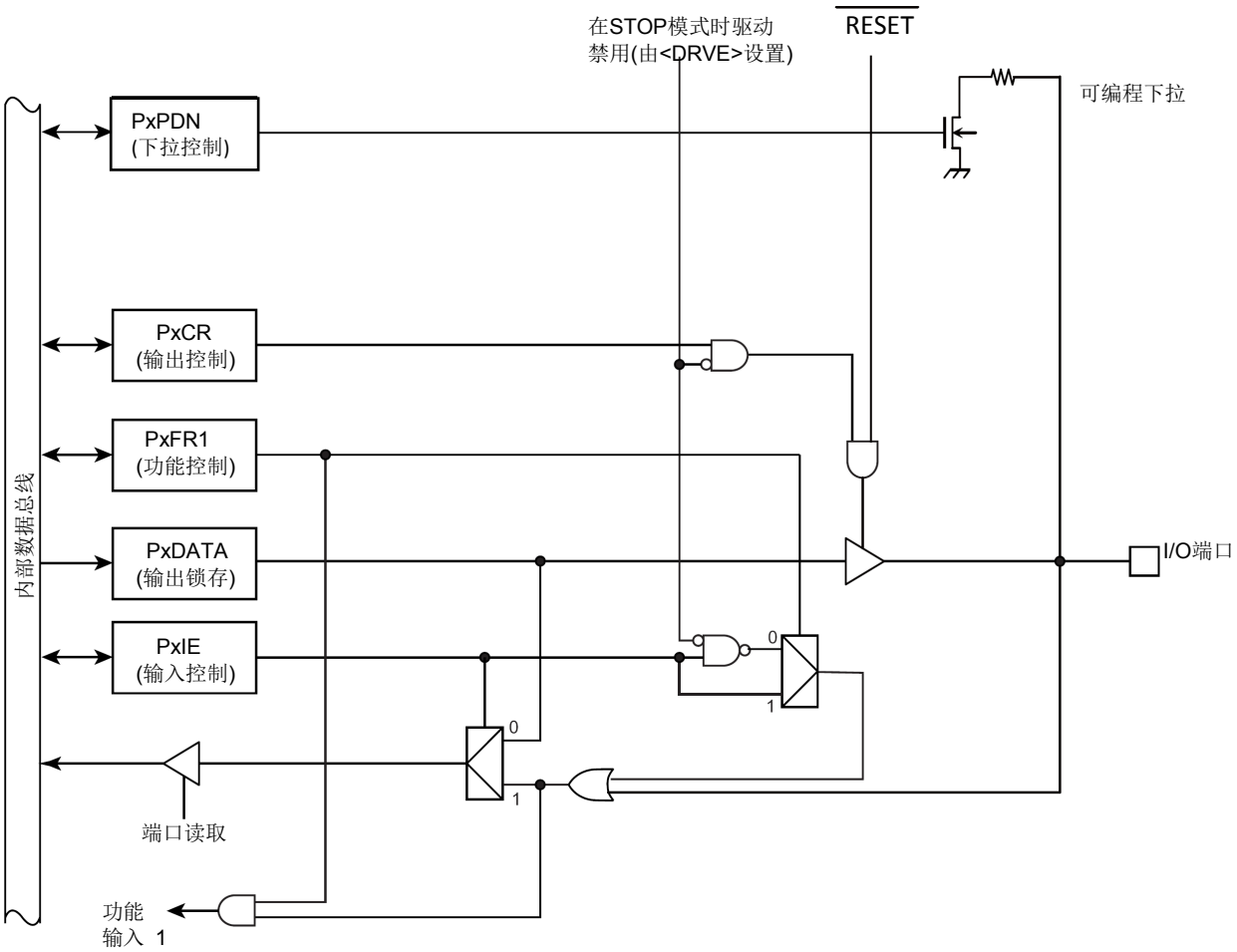


图 7-8 T8 端口类型

7.3.10 T9 类

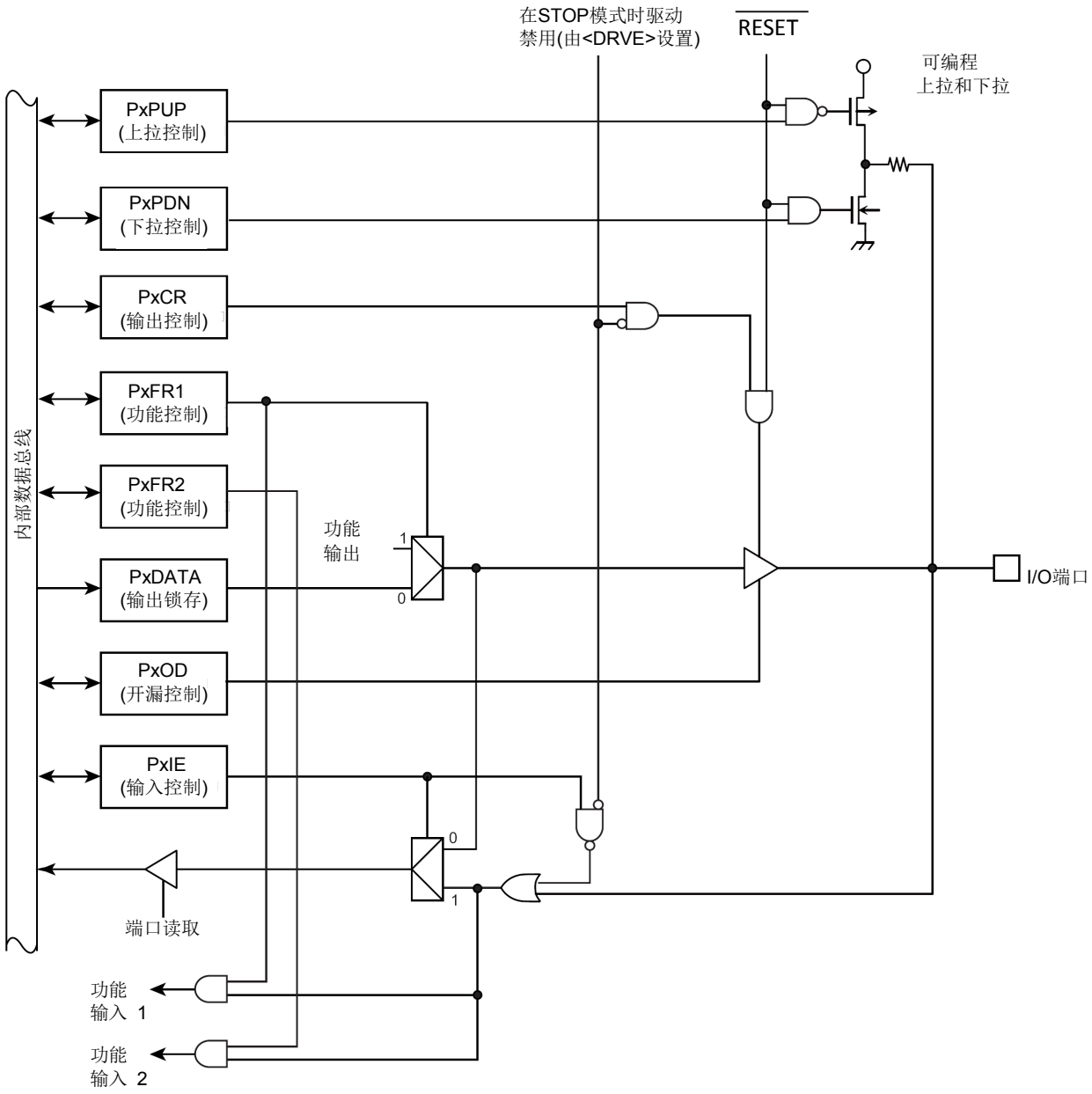


图 7-9 T9 端口类型

7.3.11 T10 类

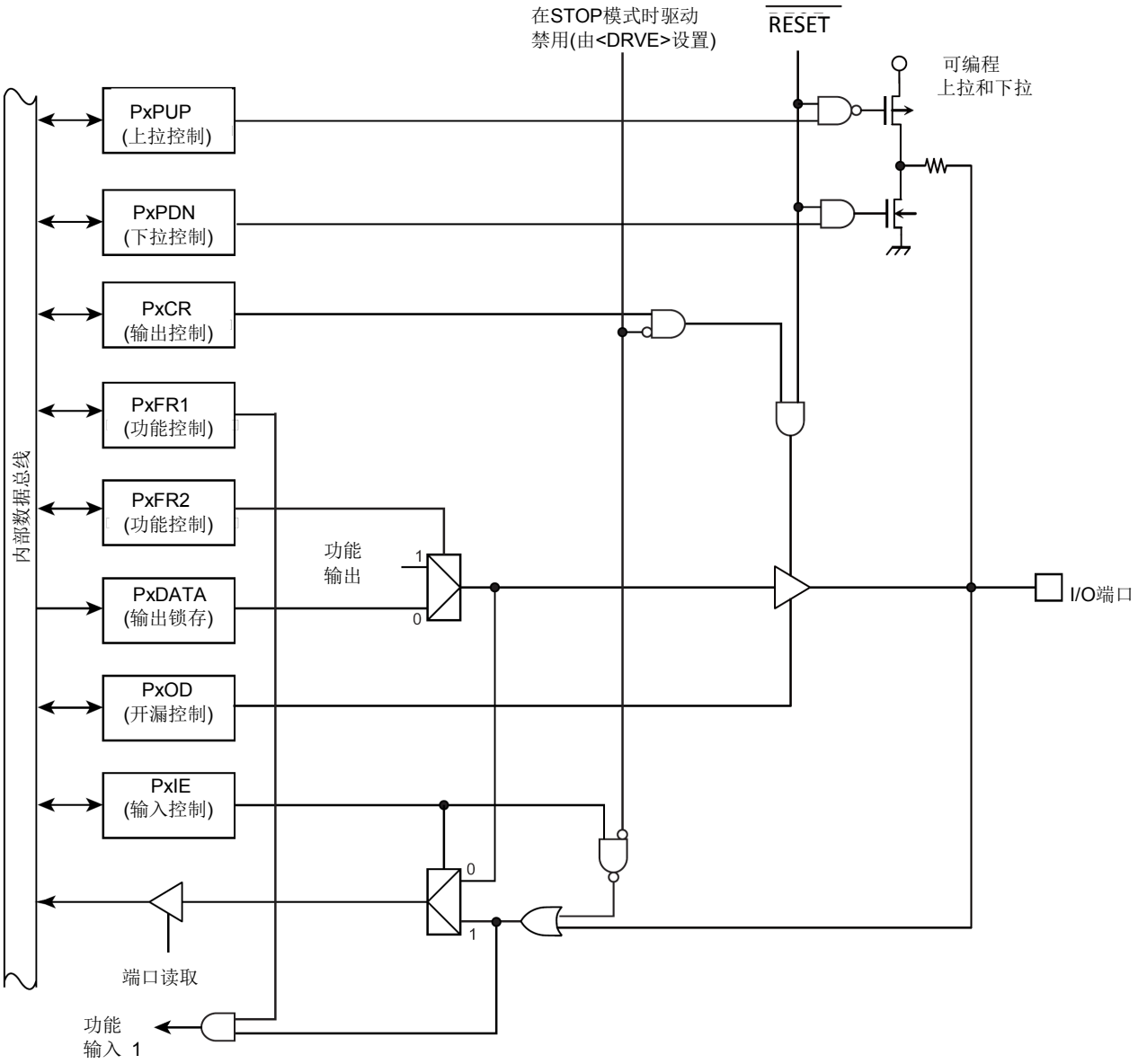


图 7-10 T10 端口类型

7.3.12 T11 类

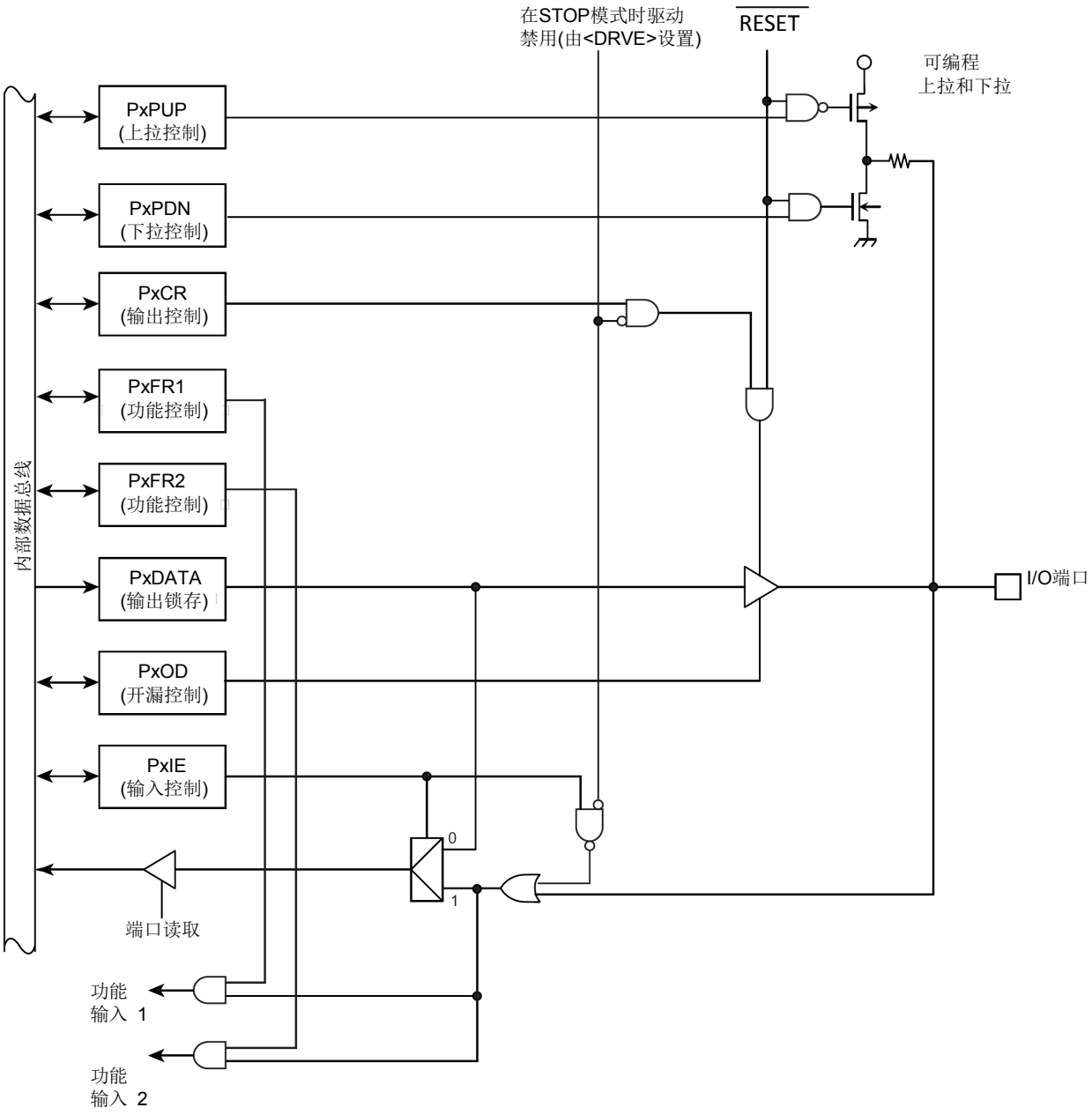


图 7-11 T11 端口类型

7.3.13 T12 类

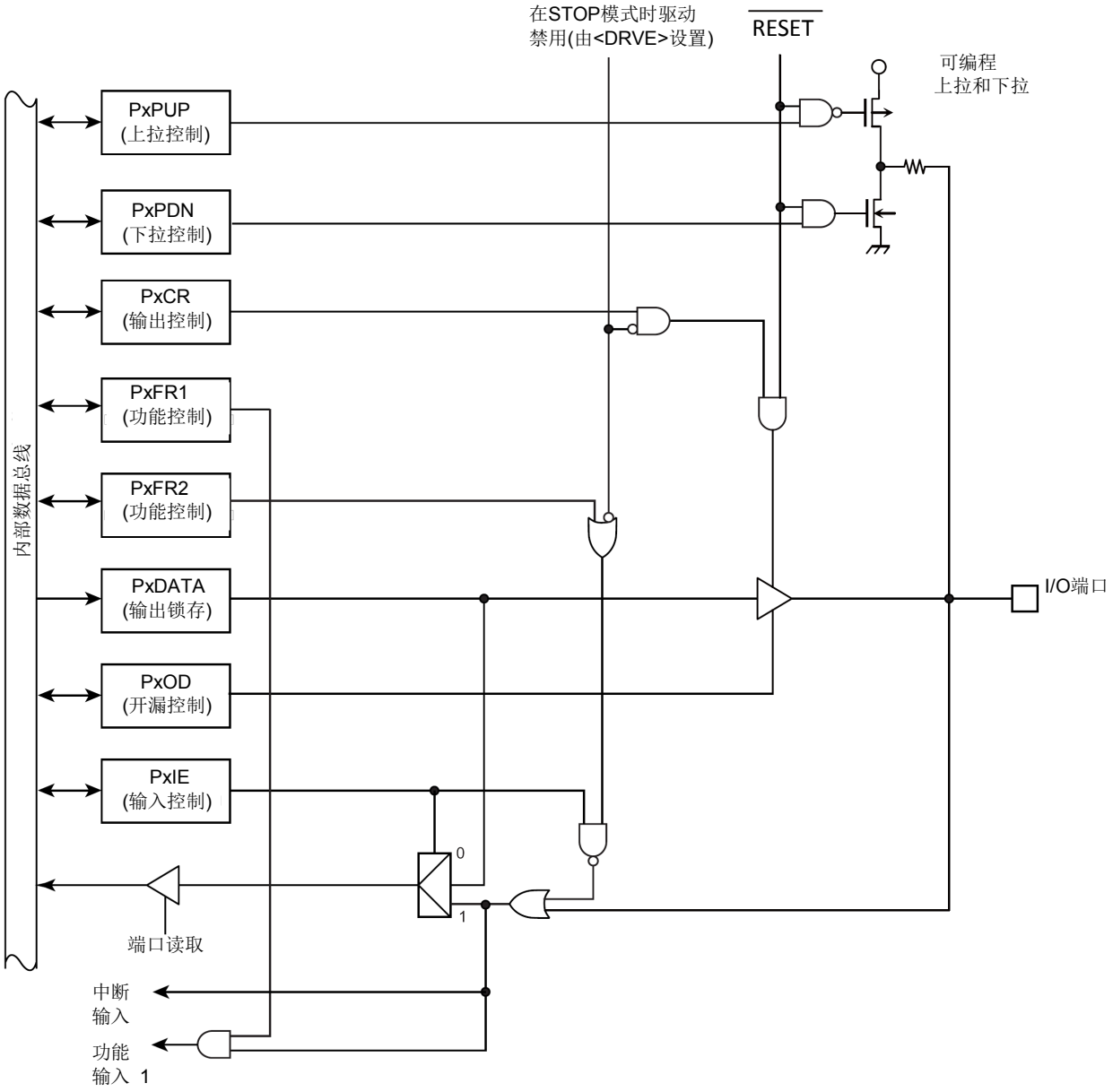


图 7-12 T12 端口类型

7.3.14 T13 类

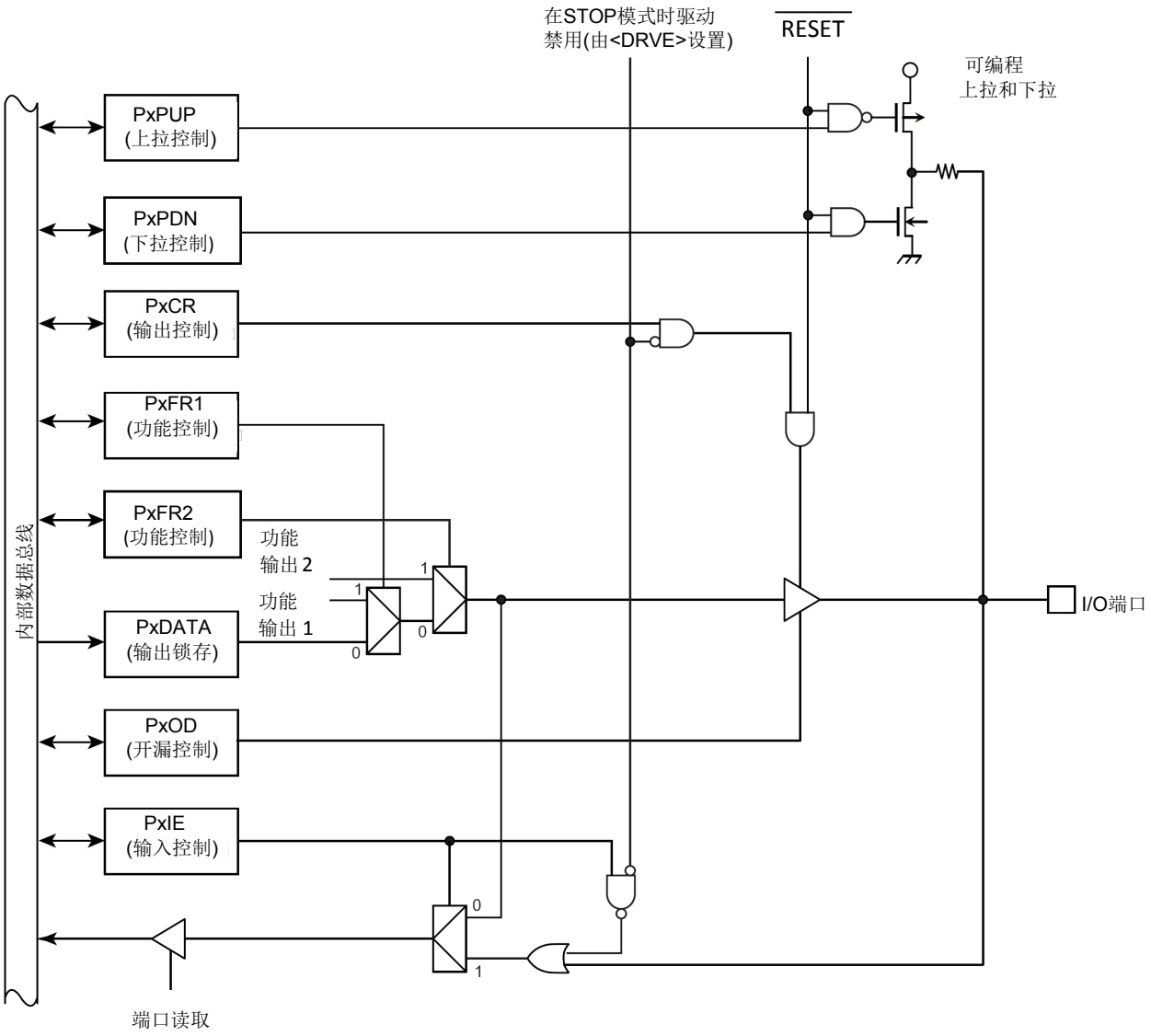


图 7-13 T13 端口类型

7.3.15 T14 类

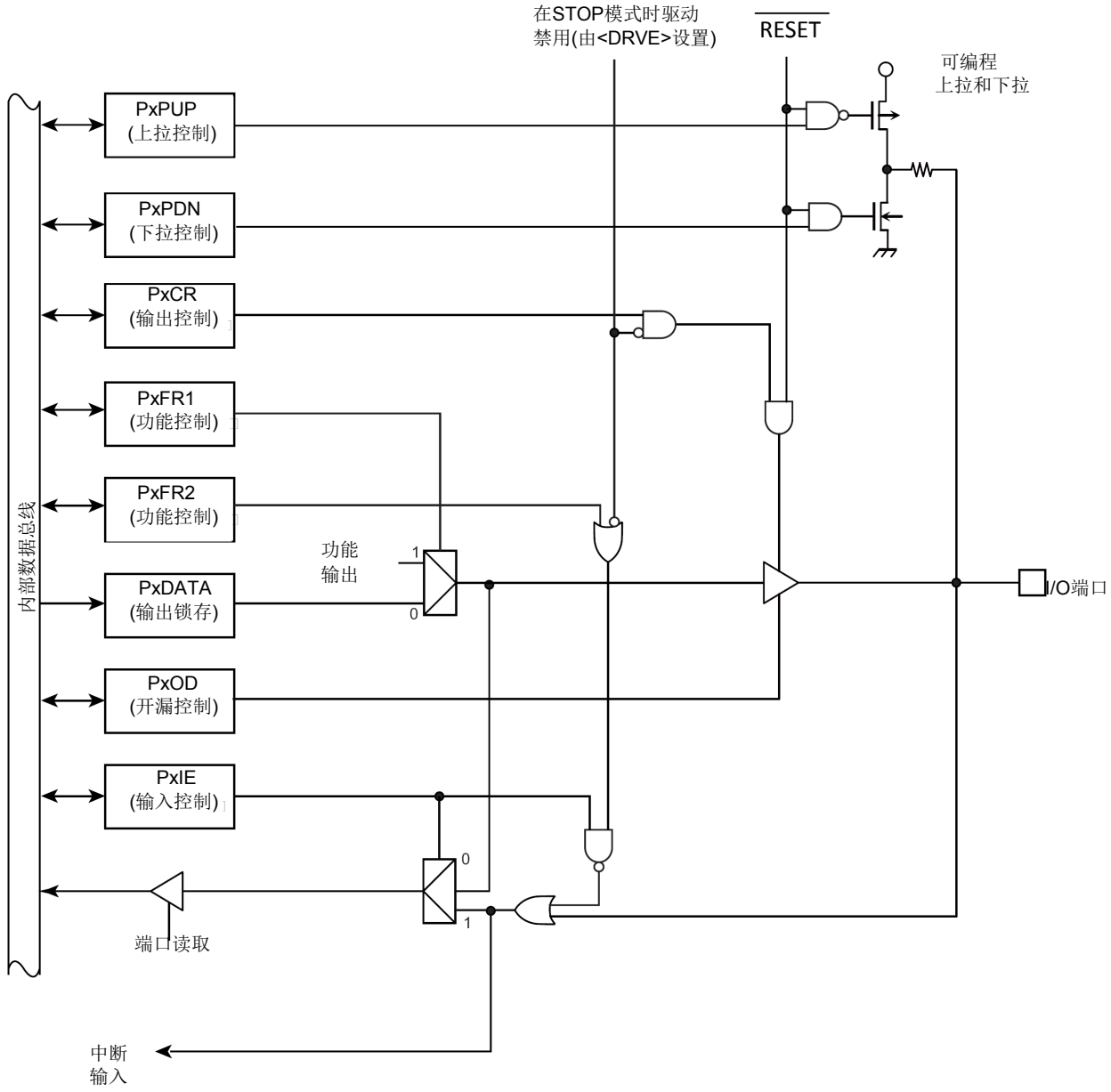


图 7-14 T14 端口类型

7.3.16 T15 类

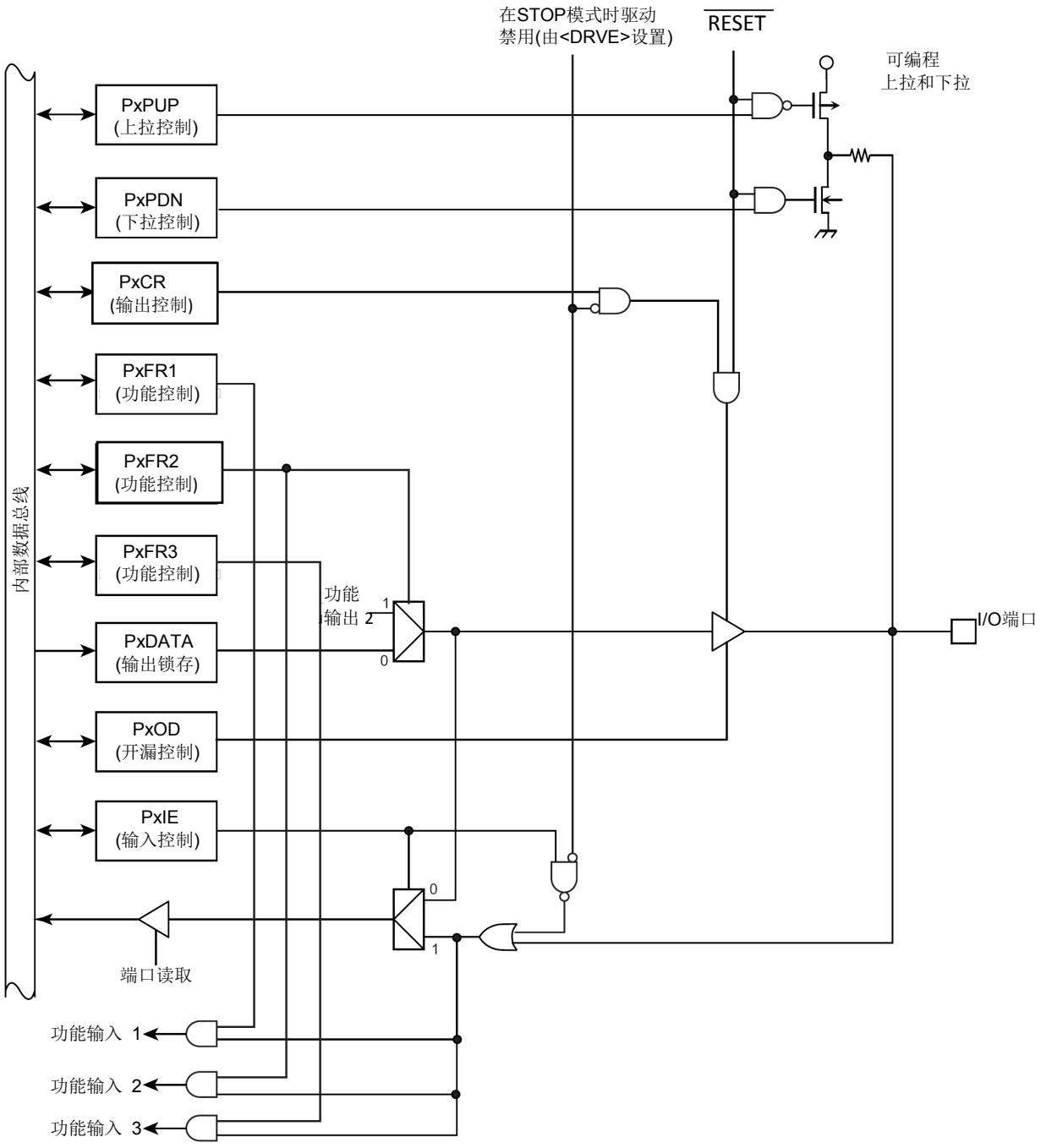


图 7-15 T15 端口类型

7.3.17 T16 类

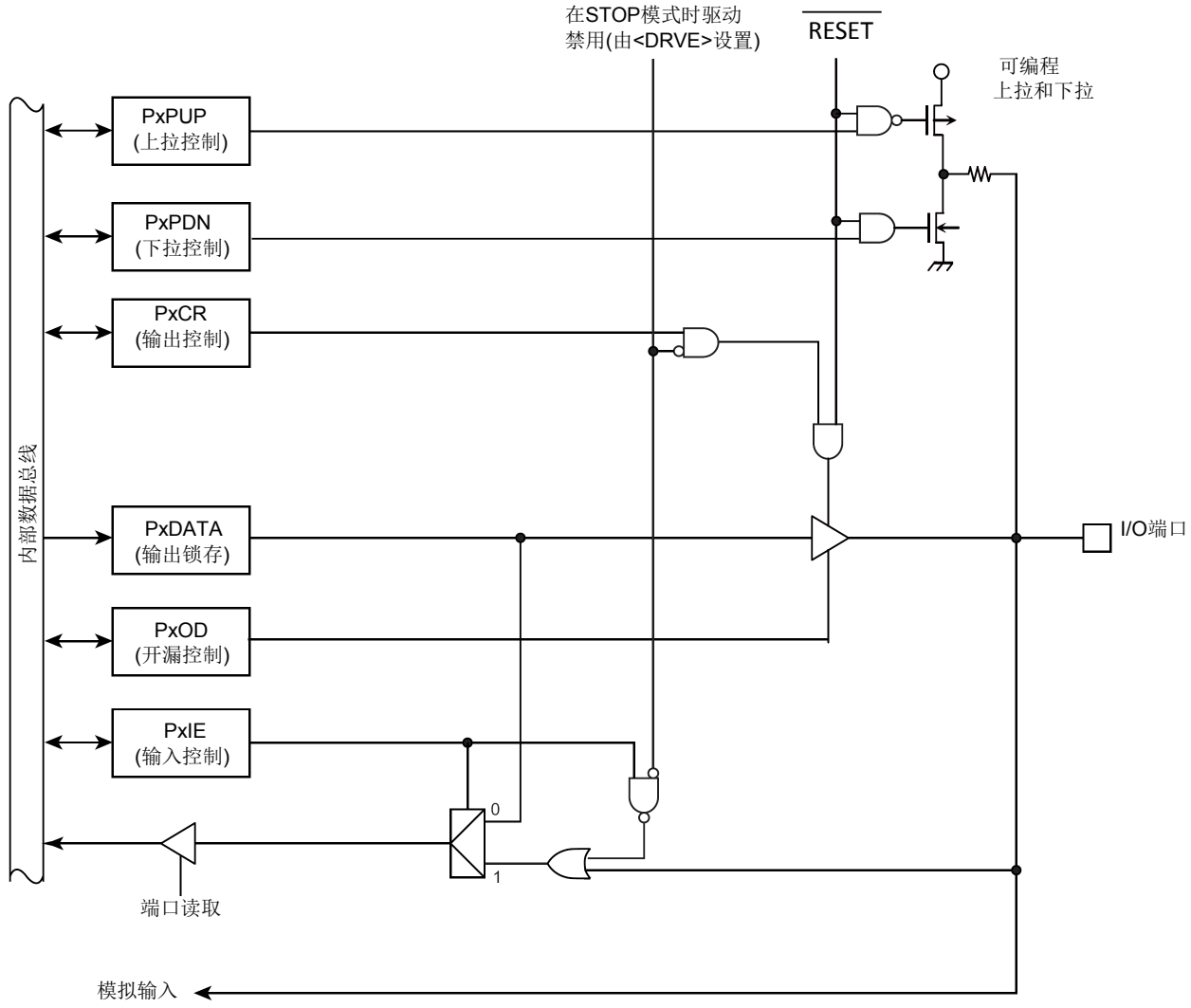


图 7-16 T16 端口类型

7.3.18 T17 类

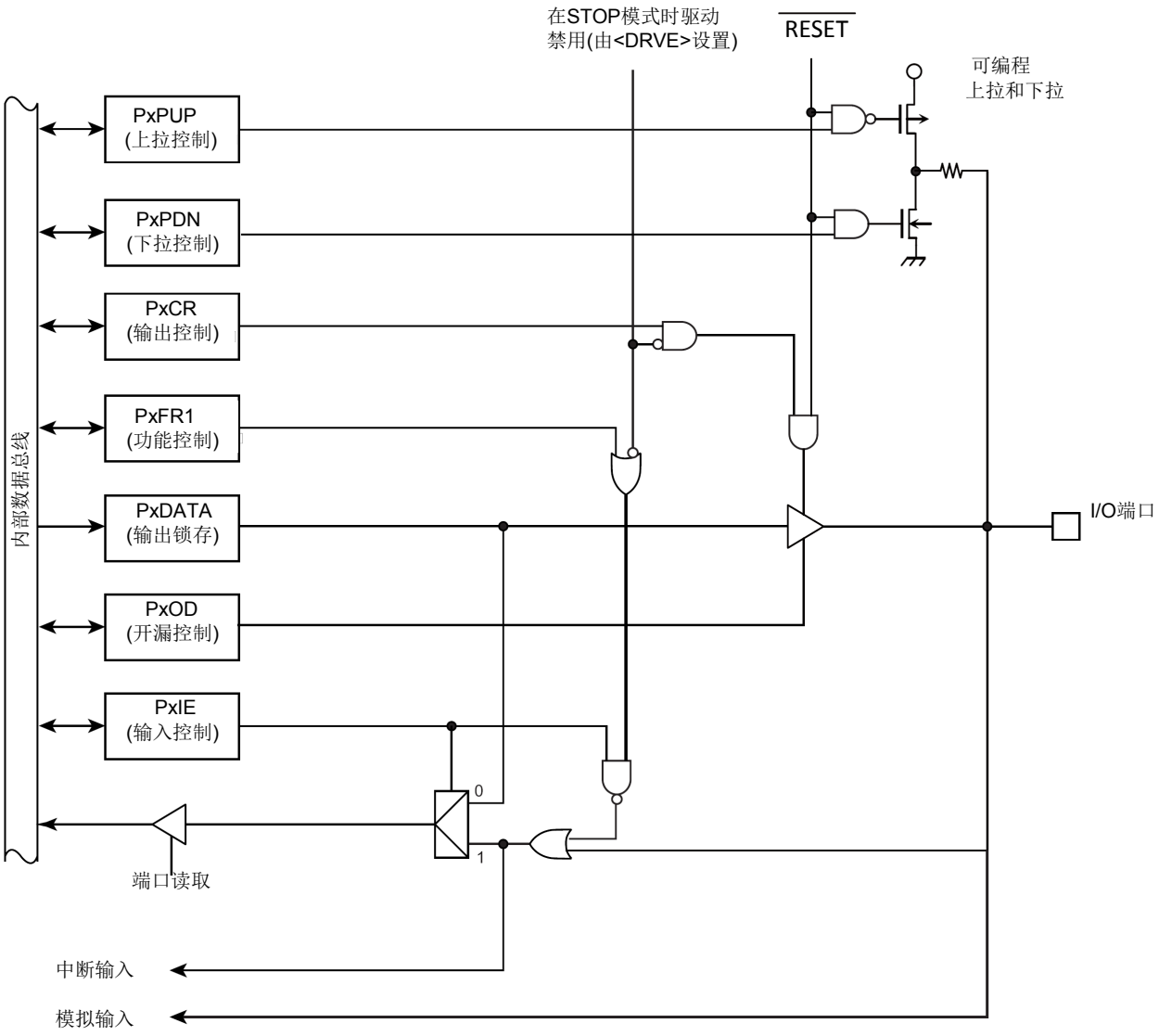


图 7-17 T17 端口类型

7.3.19 T18 类

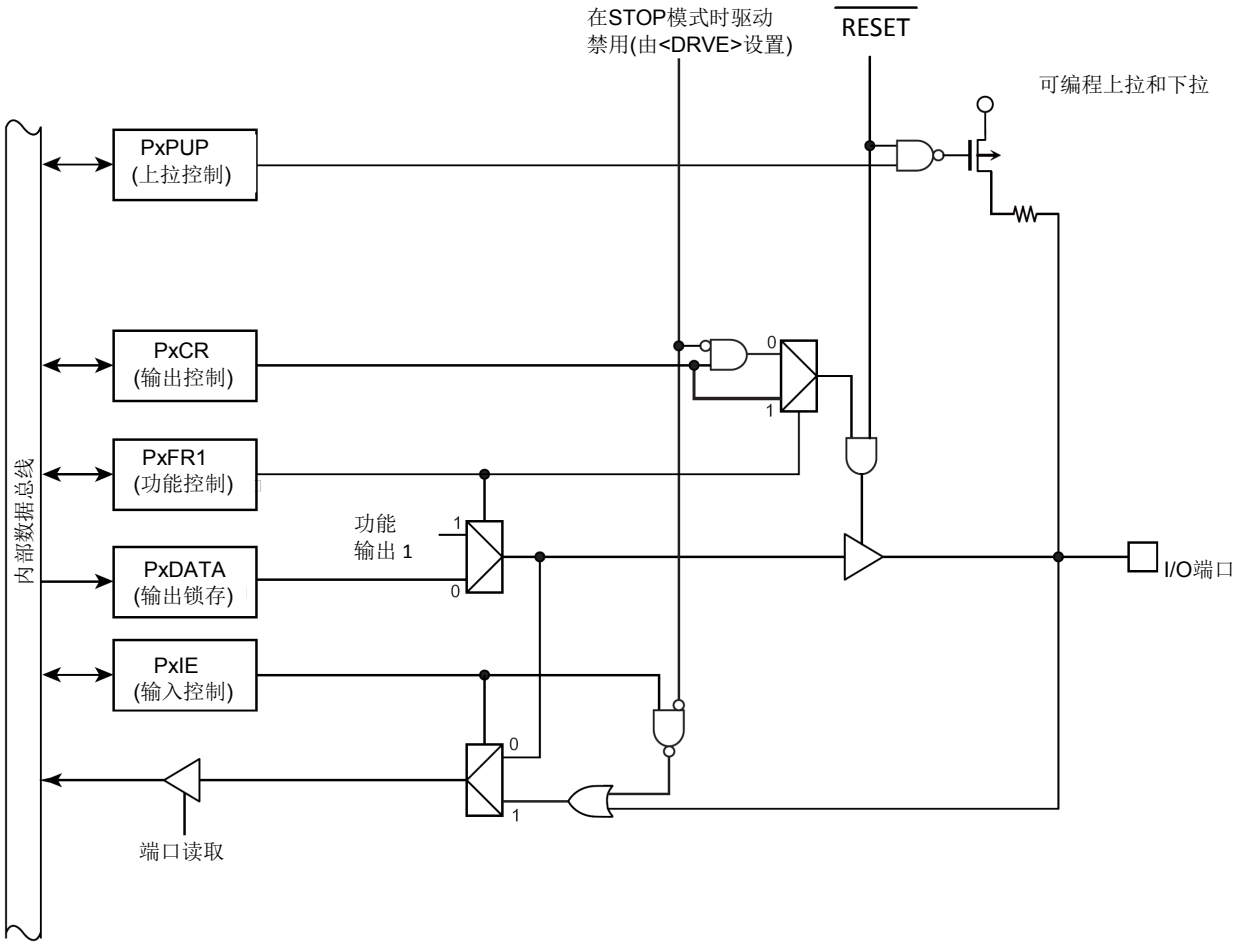


图 7-18 T18 端口类型

7.3.20 T19 类

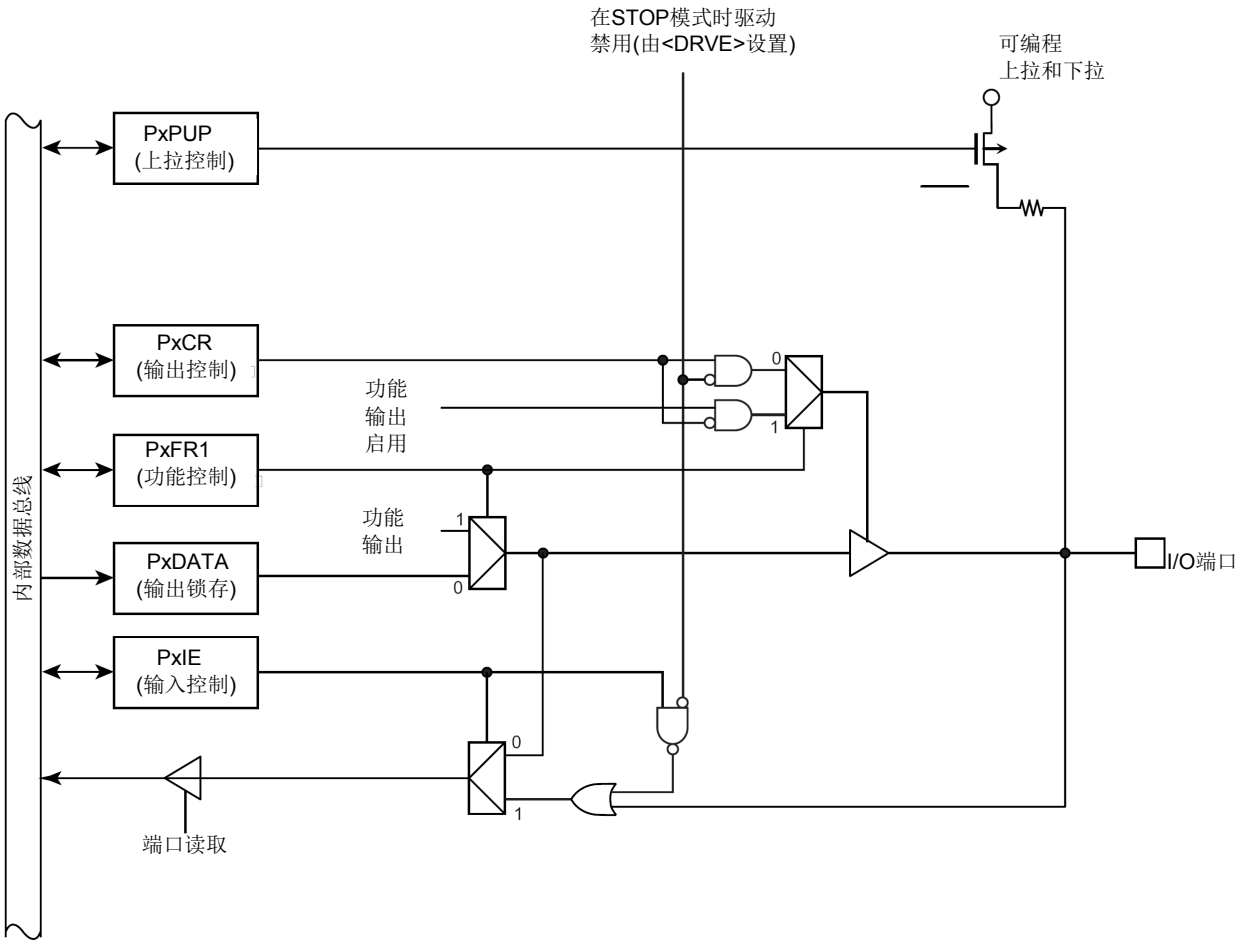
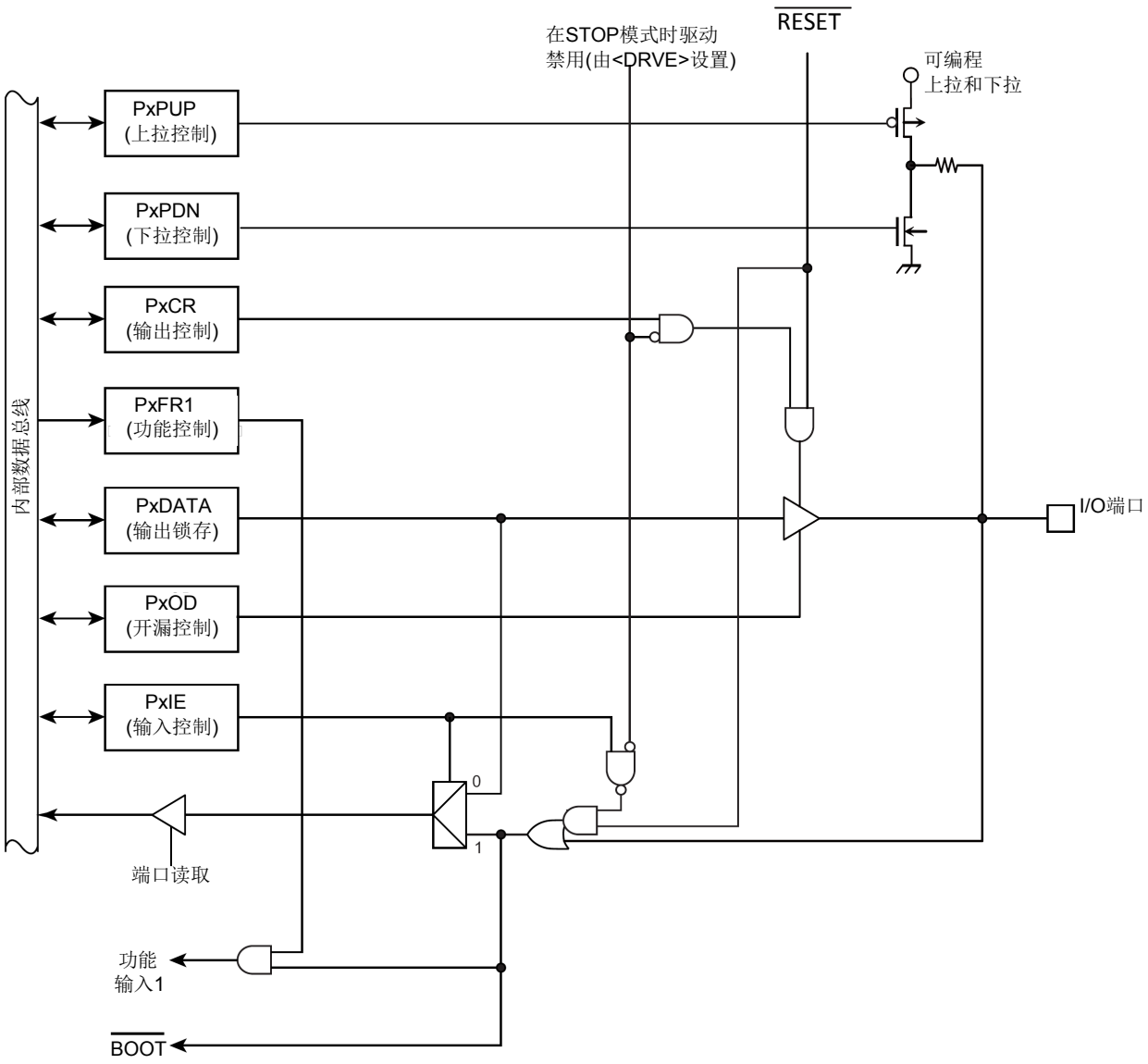
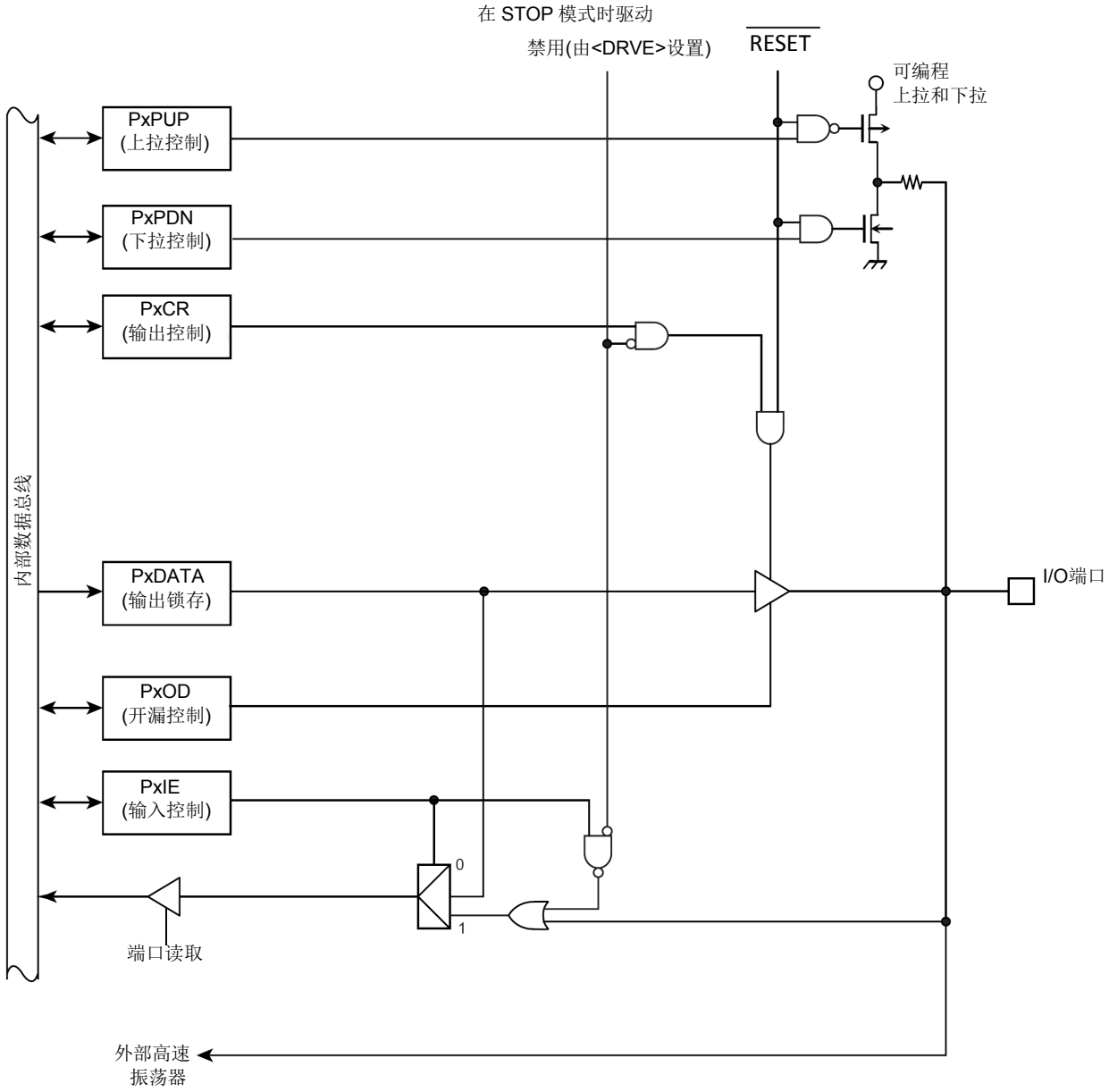


图 7-19 T19 端口类型

7.3.21 T20 类


RESET
在STOP模式时驱动
禁用(由<DRVE>设置)
可编程
上拉和下拉
PxPUP
(上拉控制)
PxPDN
(下拉控制)
PxCR
(输出控制)
PxFR1
(功能控制)
PxDATA
(输出锁存)
PxOD
(开漏控制)
PxIE
(输入控制)
内部数据总线
I/O端口
0
1
端口读取
功能
输入1
BOOT

7.3.22 T21 类


在 STOP 模式时驱动
禁用(由<DRVE>设置)
RESET
可编程
上拉和下拉
PxPUP
(上拉控制)
PxPDN
(下拉控制)
PxCR
(输出控制)
内部数据总线
PxDATA
(输出锁存)
PxOD
(开漏控制)
PxIE
(输入控制)
I/O端口
0
1
端口读取
外部高速
振荡器

7.4 附录端口设置列表

下表给出了各功能的寄存器设置。

将"复位后"字段中无[•]存在的端口的初始化设置为"0"用于所有寄存器设置。"x"位的设置可任意指定。

7.4.1 端口 A 设置

表 7-4 端口设置列表(端口 A)

引脚	端口类型	功能	复位后	PACR	PAFR1	PAFR2	PAOD	PAPUP	PAPDN	PAIE
PA0	T12	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TB0IN (输入)		0	1	0	x	x	x	1
		INT7 (输入)		0	0	1	x	x	x	1
PA1	T2	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		TB0OUT(输出)		1	1	-	x	x	x	0
PA2	T12	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TB1IN (输入)		0	1	0	x	x	x	1
		INT4 (输入)		0	0	1	x	x	x	1
PA3	T2	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		TB1OUT(输出)		1	1	-	x	x	x	0
PA4	T9	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		SCLK1 (I / O)		1	1	0	x	x	x	1
		$\overline{\text{CTS1}}$ (输入)		0	0	1	x	x	x	1
PA5	T13	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TXD1 (输出)		1	1	0	x	x	x	0
		TB6OUT(输出)		1	0	1	x	x	x	0
PA6	T11	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		RXD1 (输入)		0	1	0	x	x	x	1
		TB6IN (输入)		0	0	1	x	x	x	1
PA7	T12	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TB4IN (输入)		0	1	0	x	x	x	1
		INT8 (输入)		0	0	1	x	x	x	1

7.4.2 端口 B 设置

表 7-5 端口设置列表(端口 B)

引脚	端口类型	功能	复位后	PBCR	PBFR1	PBOD	PBPUP	PBPDN	PBIE
PB0	T18	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TRACECLK (输出)		1	1	0	0	0	0
PB1	T18	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TRACEDATA0 (输出)		1	1	0	0	0	0
PB2	T18	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TRACEDATA1 (输出)		1	1	0	0	0	0
PB3	T6	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TMS / SWDIO (I / O)		1	1	0	1	0	1
PB4	T8	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TCK / SWCLK (输入)		0	1	0	0	1	1
PB5	T19	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TDO / SWV (输出)		1	1	0	0	0	0
PB6	T7	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		TDI (输入)		0	1	0	1	0	1
PB7	T7	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		$\overline{\text{TRST}}$ (输入)		0	1	0	1	1	1

7.4.3 端口 C 设置

表 7-6 端口设置列表(端口 C)

引脚	端口类型	功能	复位后	PCCR	PCFR1	PCOD	PCPUP	PCPDN	PCIE
PC0	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		UO0 (输出)		1	1	x	x	x	0
PC1	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		XO0 (输出)		1	1	x	x	x	0
PC2	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		VO0 (输出)		1	1	x	x	x	0
PC3	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		YO0 (输出)		1	1	x	x	x	0
PC4	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		WO0 (输出)		1	1	x	x	x	0
PC5	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		ZO0 (输出)		1	1	x	x	x	0
PC6	T3	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		$\overline{\text{EMG0}}$ (输入)		0	1	x	x	x	1
PC7	T3	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		$\overline{\text{OVV0}}$ (输入)		0	1	x	x	x	1

7.4.4 端口 D 设置

表 7-7 端口设置列表(端口 D)

引脚	端口类型	功能	复位后	PDCR	PDFR1	PDFR2	PDOD	PDPUP	PDPDN	PDIE
PD0	T11	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		ENCA0 (输入)		0	1	0	x	x	x	1
		TB5IN (输入)		0	0	1	x	x	x	1
PD1	T10	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		ENCB0 (输入)		0	1	0	x	x	x	1
		TB5OUT (输出)		1	0	1	x	x	x	0
PD2	T3	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		ENCZ0(输入)		0	1	-	x	x	x	1
PD3	T4	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		INT9 (输入)		0	1	-	x	x	x	1
PD4	T9	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		SCLK2 (I / O)		1	1	0	x	x	x	1
		$\overline{\text{CTS2}}$ (输入)		0	0	1	x	x	x	1
PD5	T2	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		TXD2 (输出)		1	1	-	x	x	x	0
PD6	T3	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		RXD2 (输入)		0	1	-	x	x	x	1

7.4.5 端口 E 设置

表 7-8 端口设置列表(端口 E)

引脚	端口类型	功能	复位后	PECR	PEFR1	PEFR2	PEOD	PEPUP	PEPDN	PEIE
PE0	T2	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		TXD0 (输出)		1	1	-	x	x	x	0
PE1	T3	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		RXD0 (输入)		0	1	-	x	x	x	1
PE2	T9	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		SCLK0 (I / O)		1	1	0	x	x	x	1
		$\overline{\text{CTS0}}$ (输入)		0	0	1	x	x	x	1
PE3	T2	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		TB4OUT (输出)		1	1	-	x	x	x	0
PE4	T12	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TB2IN (输入)		0	1	0	x	x	x	1
		INT5 (输入)		0	0	1	x	x	x	1
PE5	T2	输入端口		0	0	-	x	x	x	1
		输出端口		1	0	-	x	x	x	0
		TB2OUT (输出)		1	1	-	x	x	x	0
PE6	T12	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TB3IN (输入)		0	1	0	x	x	x	1
		INT6 (输入)		0	0	1	x	x	x	1
PE7	T14	输入端口		0	0	0	x	x	x	1
		输出端口		1	0	0	x	x	x	0
		TB3OUT (输出)		1	1	0	x	x	x	0
		INT7 (输入)		0	0	1	x	x	x	1

7.4.6 端口 F 设置

表 7-9 端口设置列表(端口 F)

引脚	端口类型	功能	复位后	PFCR	PFFR1	PFFR2	PFFR3	PFOD	PFPUP	PFPDN	PFIE
PF0	T20	输入端口		0	0	-	-	x	x	x	1
		输出端口		1	0	-	-	x	x	x	0
		TB7IN (输入)		0	1	-	-	x	x	x	1
PF1	T2	输入端口		0	0	-	-	x	x	x	1
		输出端口		1	0	-	-	x	x	x	0
		TB7OUT (输出)		1	1	-	-	x	x	x	0
PF2	T15	输入端口		0	0	0	0	x	x	x	1
		输出端口		1	0	0	0	x	x	x	0
		ENCA1 (输入)		0	1	0	0	x	x	x	1
		SCLK3 (I / O)		1	0	1	0	x	x	x	1
		$\overline{\text{CTS3}}$ (输入)		0	0	0	1	x	x	x	1
PF3	T10	输入端口		0	0	0	-	x	x	x	1
		输出端口		1	0	0	-	x	x	x	0
		ENCB1 (输入)		0	1	0	-	x	x	x	1
		TXD3 (输出)		1	0	1	-	x	x	x	0
PF4	T11	输入端口		0	0	0	-	x	x	x	1
		输出端口		1	0	0	-	x	x	x	0
		ENCZ1 (输入)		0	1	0	-	x	x	x	1
		RXD3 (输入)		0	0	1	-	x	x	x	1

注：PF0 输入与上拉被启用并充当 $\overline{\text{BOOT}}$ 输入引脚，而 $\overline{\text{RESET}}$ 则处于"低"电平状态

7.4.7 端口 G 设置

表 7-10 端口设置列表(端口 G)

引脚	端口类型	功能	复位后	PGCR	PGFR1	PGOD	PGPUP	PGPDN	PGIE
PG0	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		UO1 (输出)		1	1	x	x	x	0
PG1	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		XO1 (输出)		1	1	x	x	x	0
PG2	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		VO1 (输出)		1	1	x	x	x	0
PG3	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		YO1 (输出)		1	1	x	x	x	0
PG4	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		WO1 (输出)		1	1	x	x	x	0
PG5	T1	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		ZO1 (输出)		1	1	x	x	x	0
PG6	T3	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		$\overline{\text{EMG1}}$ (输入)		0	1	x	x	x	1
PG7	T3	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		$\overline{\text{OVV1}}$ (输入)		0	1	x	x	x	1

7.4.8 端口 H 设置

表 7-11 端口设置列表(端口 H)

引脚	端口类型	功能	复位后	PHCR	PHFR1	PHOD	PHPUP	PHPDN	PHIE
PH0	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INT0 (输入)		0	1	x	x	x	1
PH1	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INT1 (输入)		0	1	x	x	x	1
PH2	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INT2 (输入)		0	1	x	x	x	1
PH3	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PH4	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PH5	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PH6	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PH7	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0

7.4.9 端口 I 设置

表 7-12 端口设置列表(端口 I)

引脚	端口类型	功能	复位后	PICR	PIOD	PIPUP	PIPDN	PIIE
PI0	T16	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
		模拟输入		0	0	0	0	0
PI1	T16	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
		模拟输入		0	0	0	0	0
PI2	T16	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
		模拟输入		0	0	0	0	0
PI3	T16	输入端口		0	x	x	x	1
		输出端口		1	x	x	x	0
		模拟输入		0	0	0	0	0

7.4.10 端口 J 设置

表 7-13 端口设置列表(端口 J)

引脚	端口类型	功能	复位后	PJCR	PJFR1	PJOD	PJPUP	PJPDN	PJIE
PJ0	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PJ1	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PJ2	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PJ3	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PJ4	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PJ5	T16	输入端口		0	-	x	x	x	1
		输出端口		1	-	x	x	x	0
		模拟输入		0	-	0	0	0	0
PJ6	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INTC (输入)		0	1	x	x	x	1
PJ7	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INTD (输入)		0	1	x	x	x	1

7.4.11 端口 K 设置

表 7-14 端口设置列表(端口 K)

引脚	端口类型	功能	复位后	PKCR	PKFR1	PKOD	PKPUP	PKPDN	PKIE
PK0	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INTE (输入)		0	1	x	x	x	1
PK1	T17	输入端口		0	0	x	x	x	1
		输出端口		1	0	x	x	x	0
		模拟输入		0	0	0	0	0	0
		INTF (输入)		0	1	x	x	x	1

7.4.12 端口 L 设置

表 7-15 端口设置列表(端口 L)

引脚	端口类型	功能	复位后	PLFR1	PLIE
PL0	T5	输入端口		0	1
		输出端口		0	0
		INTB (输入)		1	1
PL1	T5	输入端口		0	1
		输出端口		0	0
		INTA (输入)		1	1

8. 16-位计时器/事件计数器(TMRB)

8.1 概述

TMRB 可按以下四种运行模式运行：

- 16-位间隔计时器模式
- 16-位事件计数器模式
- 16-位可编程脉冲发生模式(PPG)
- 外触发器可编程脉冲发生模式(PPG)

捕捉功能的使用，允许 TMRB 执行以下两种测量。

- 由外触发器进行单触发脉冲输出
- 脉冲宽度测量

在本节的以下说明中，"x"表示通道编号。

8.2 规格差异

TMPM370FYDFG/FYFG 包含 TMRB 的 8-通道。

各通道均可独立工作，且除表 8-1 所示的规格差异之外，各通道均以同样的方式运行。

表 8-1 各 TMRB 模块的规格差异

通道规格	外部引脚		中断		内部连接	
	外部时钟/捕捉触发信号输入引脚	计时器触发电路输出引脚	捕捉中断	TMRB 中断	ADC 转换启动	计时器触发电路输出 TBxOUT 从 SIO/UART (TXTRG：传输时钟)
	信号名称	信号名称				
TMRB0	TB0IN	TB0OUT	INTCAP00 INTCAP01	INTTB00 INTTB01		
TMRB1	TB1IN	TB1OUT	INTCAP10 INTCAP11	INTTB10 INTTB11		
TMRB2	TB2IN	TB2OUT	INTCAP20 INTCAP21	INTTB20 INTTB21		
TMRB3	TB3IN	TB3OUT	INTCAP30 INTCAP31	INTTB30 INTTB31		
TMRB4	TB4IN	TB4OUT	INTCAP40 INTCAP41	INTTB40 INTTB41		SIO0,SIO1
TMRB5	TB5IN	TB5OUT	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51	
TMRB6	TB6IN	TB6OUT	INTCAP60 INTCAP61	INTTB60 INTTB61		
TMRB7	TB7IN	TB7OUT	INTCAP70 INTCAP71	INTTB70 INTTB71		SIO2,SIO3

8.3 配置

各通道均由一个 16-位递增计数器，两个 16-位计时器寄存器(双缓冲)，两个 16-位捕捉寄存器，两个比较器，一个捕捉输入控制器，一个计时器触发电路及其相关控制电路构成。计时器运行模式与计时器触发电路均受控于某个寄存器。

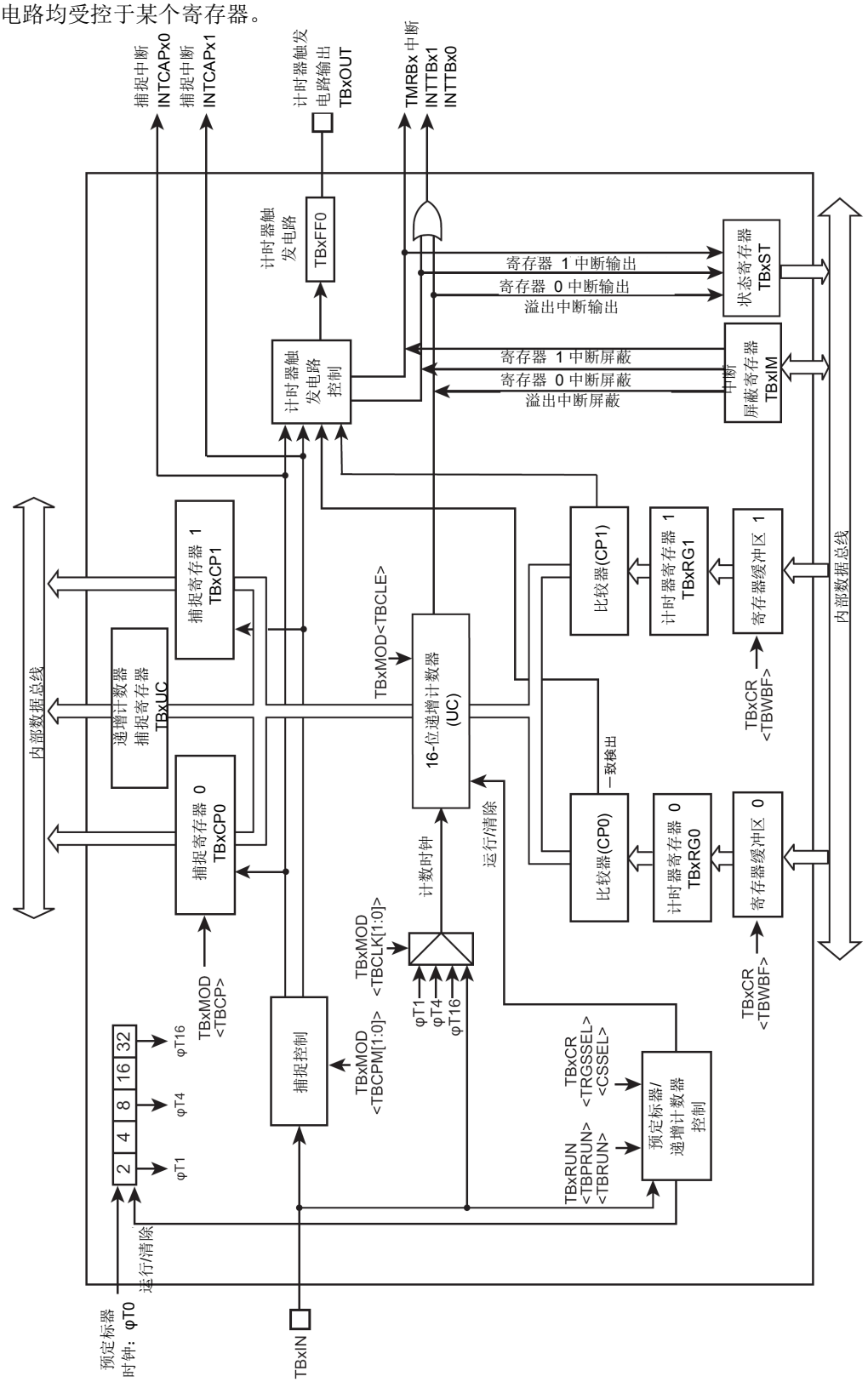


图 8-1 TMRBx 方块图(x= 0 ~ 7)

8.4 寄存器

8.4.1 寄存器列表按通道

下表给出了各通道的寄存器名称与地址。

通道 x	基址
通道 0	0x4001_0000
通道 1	0x4001_0040
通道 2	0x4001_0080
通道 3	0x4001_00C0
通道 4	0x4001_0100
通道 5	0x4001_0140
通道 6	0x4001_0180
通道 7	0x4001_01C0

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

8.4.2 TBxEN(启用寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 "0"。
7	TBEN	R/W	TMRBx 运行 0：禁用 1：启用 规定 TMRB 运行。在该运行被禁用时，不会向 TMRB 模块中的其它寄存器提供任何时钟脉冲。这样可降低功耗(禁止对 TBxEN 寄存器以外的其它所有寄存器进行读取写入操作)。 如需使用 TMRB，可在对该 TMRB 模块中的各寄存器进行编程之前，启动该 TMRB 运行(即将其设置为 "1")。如 TMRB 在被禁用之前执行了运行，则各寄存器中仍将保留该设置。
6-0	−	R	读作 "0"。

8.4.3 TBxRUN(RUN 寄存器)

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

位	比特符号	型号	功能
31-3	−	R	读作 "0"。
2	TBPRUN	R/W	预分频操作 0：停止&清除 1：计数
1	−	R	读作 "0"。
0	TBRUN	R/W	计数操作 0：停止&清除 1：计数

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

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

8.4.4 TBxCR(控制寄存器)

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

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

注：1：在运行 TMRB 期间，不要修改 TBxCR。

注 2：在使用了外触发器启动时(<CSSEL>=1)，应在<TBRUN>=<TBPRUN>=1 设置前选择 <CSSEL> 与 <TRGSEL>。

8.4.5 TBxMOD(模式寄存器)

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

位	比特符号	型号	功能
31-7	−	R	读作 "0"。
6	TBRSWR	R/W	写入计时器寄存器 0 与 1 (在双缓冲器被启用时) 0：目标为计时器寄存器 0 与 1 的数据传送，是通过对应该递增计数器(UC)实现的，与缓冲寄存器 0 与 1 的重写无关。 1：在将该缓冲寄存器数据传送到计时器寄存器时，需同时进行计时器寄存器 0 与 1 的写入。
5	TBCP	W	由软件进行捕捉控制 0：由软件进行捕捉 1：忽略 在已写入"0"时，由捕捉寄存器 0 (TBxCP0)取计数值。 读作"1"。
4-3	TBCPM[1:0]	R/W	捕捉时序 00：禁用捕捉时序 01：TBxIN↑ 一旦 TBxIN 引脚输入上升，即将各计数值纳入捕捉寄存器 0 (TBxCP0)。 10：TBxIN↑ TBxIN↓ 一旦 TBxIN 引脚输入上升，即将各计数值纳入捕捉寄存器 0 (TBxCP0)。 一旦 TBxIN 引脚输入下降，即将各计数值纳入捕捉寄存器 1(TBxCP1)。 11：禁用捕捉时序
2	TBCLE	R/W	递增计数器控制器 0：禁用递增计数器的清除 1：启用递增计数器的清除。 清除并控制该递增计数器。 在已写入"0"时，其可禁用递增计数器的清除。在已写入"1"时，其可在存在寄存器 1(TBxRG1)匹配的情况下清除递增计数器。
1-0	TBCLK[1:0]	R/W	选择该 TMRBx 源时钟。 00：TBxIN 引脚输入 01：ϕT1 10：ϕT4 11：ϕT16

注：不得在该计时器运行期间改变 TBxMOD 寄存器。

8.4.6 TBxFFCR(触发电路控制寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 "0"。
7-6	−	R	读作 "1"。
5	TBC1T1	R/W	在该递增计数器值被纳入 TBxCP1 时，TBxFF0 反转触发。 0：禁用触发 1：启用触发 通过设置"1"，该计时器触发电路会在递增计数器被纳入该捕捉寄存器 1(TBxCP1)时反转。
4	TBC0T1	R/W	在该递增计数器值被纳入 TBxCP0 时，TBxFF0 反转触发。 0：禁用触发 1：启用触发 通过设置"1"，该计时器触发电路会在递增计数器被纳入该捕捉寄存器 0(TBxCP0)时反转。
3	TBE1T1	R/W	在该递增计数器值与 TBxRG1 匹配时，TBxFF0 反转触发。 0：禁用触发 1：启用触发 通过设置"1"，该计时器触发电路会在递增计数器与计时器寄存器 1(TBxRG1)匹配时反转。
2	TBE0T1	R/W	在该递增计数器值与 TBxRG0 匹配时，TBxFF0 反转触发。 0：禁用触发 1：启用触发 通过设置"1"，该计时器触发电路会在递增计数器与计时器寄存器 0(TBxRG0)匹配时反转。
1-0	TBFF0C[1:0]	R/W	TBxFF0 控制器 00：转换 反转 TBxFF0 的值 (利用软件反转)。 01：设置 将 TBxFF0 设置为"1"。 10：清除 将 TBxFF0 清为"0"。 11：忽略 * 始终读作"11"。

注： 不要在该计时器运行期间改变 TBxFFCR 寄存器。

8.4.7 TBxST(状态寄存器)

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

位	比特符号	型号	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 在某递增计数器溢出时，即设置"1"。
1	INTTB1	R	匹配标志(TBxRG1) 0：未检测到任何匹配 1：检测到 TBxRG1 的匹配 在检测到计时器寄存器 1(TBxRG1)的匹配时，即设置"1"。
0	INTTB0	R	匹配标志(TBxRG0) 0：未检测到任何匹配 1：检测到 TBxRG0 的匹配 在检测到计时器寄存器 0(TBxRG0)的匹配时，即设置"1"。

注 1：仅未被 TBxIM 屏蔽的各因数会向 CPU 输出中断请求。即使屏蔽设置已完成，仍会设置该标志。

注 2：通过读取 TBxST 寄存器，即可清除该标志。在清除该标志时，应读取 TBxST 寄存器。

8.4.8 TBxIM(中断屏蔽寄存器)

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

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

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

8.4.9 TBxUC (递增计数器捕捉寄存器)

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

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

注：在该计数器处于运行状态，且已读取 TBxUC 时，会捕捉并读取递增计数器的值。

8.4.10 TBxRG0(计时器寄存器 0)

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

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

8.4.11 TBxRG1(计时器寄存器 1)

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

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

8.4.12 TBxCP0(捕捉寄存器 0)

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

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

8.4.13 TBxCP1(捕捉寄存器 1)

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

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

8.5 各电路的运行说明

且除表 8-1 所示的规格差异之外，各通道均以同样的方式运行。

8.5.1 预分频

利用现有的 4-位预分频，可生成递增计数器 UC 的源时钟。

预分频输入时钟 ϕT0 是通过 CG 中的 CGSYSCR<PRCK[2:0]>选中的 fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 或 fperiph/32。外围时钟 fperiph 可以是 fgear(通过 CG 中的 CGSYSCR<FPSEL>选中的时钟)，也可以是 fc(被时钟齿轮划分之前的时钟)。

用 TBxRUN<TBPRUN>设置预分频的运行或停止，其中，通过写入"1"可开始计数，写入"0"即可清除并停止计数。表 8-2 给出了预分频输出时钟分辨率。

表 8-2 预分频输出时钟分辨率 (fc = 80MHz)

选择 外围时钟 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	选择 预分频时钟 CGSYSCR <PRCK[2:0]>	预分频输出时钟功能		
			ϕT1	ϕT4	ϕT16
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025 μs)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		001 (fperiph/2)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		010 (fperiph/4)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		011 (fperiph/8)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		100 (fperiph/16)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		101 (fperiph/32)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		001 (fperiph/2)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		010 (fperiph/4)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		011 (fperiph/8)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
	111 (fc/16)	000 (fperiph/1)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		001 (fperiph/2)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		010 (fperiph/4)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		011 (fperiph/8)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		100 (fperiph/16)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
		101 (fperiph/32)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)	$fc/2^{14}$ (204.8 μs)

表 8-2 预分频输出时钟分辨率 (fc = 80MHz)

选择 外围时钟 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	选择 预分频时钟 CGSYSCR <PRCK[2:0]>	预分频输出时钟功能		
			ϕT1	ϕT4	ϕT16
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025 μs)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	–	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	101 (fc/4)	000 (fperiph/1)	–	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	–	$fc/2^4$ (0.2 μs)	fc/26 (0.8 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	–	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		010 (fperiph/4)	–	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	–	–	$fc/2^6$ (0.8 μs)
		010 (fperiph/4)	–	$fc/2^5$ (0.4 μs)	fc/27 (1.6 μs)
		011 (fperiph/8)	–	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)

注 1：必须选择预分频输出时钟 ϕTn，以满足 ϕTn < fsys 的条件(使得 ϕTn 慢于 fsys)。

注 2：注：不要在该计时器运行期间改变时钟齿轮。

注 3：符号"–"表示某设置被禁止。

8.5.2 递增计数器(UC)

UC 为 16-位二进制计数器。

• 源时钟

可从预分频输出时钟或 TBxIN 引脚外部时钟的三类 ϕT1, ϕT4, ϕT16 中，选择 UC 源时钟由 TBxMOD<TBCLK [1:0]>指定。

• 计数开始/停止

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

• UC 清除用时序

1. 在检测到某匹配时

在比较器检测到计数器值与在 TBxRG1 中设置的值之间存在匹配时，通过设置 TBxMOD<TBCLE> = "1"，即可清除 UC。如果 TBxMOD<TBCLE> = "0"，则 UC 可作为自由运行计数器运行。

2. 在 UC 停止时

如果 TBxRUN<TBRUN>="0"，则 UC 停止计数并清除计数器值。

• UC 溢出

如果发生 UC 溢出，则会生成 INTTBx0 溢出中断。

8.5.3 计时器寄存器(TBxRG0, TBxRG1)

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

TBxRG0 与 TBxRG1 均采用与各寄存器缓冲配套的双缓冲配置结构。在初始状态时，该双缓冲被禁用。

控制用双缓冲的禁用或启用，由 TBxCR<TBWBF>位指定。如果<TBWBF> = "0"，则双缓冲变为禁用。如果<TBWBF> = "1"，则其变为启用。在双缓冲被启用时，可在 UC 可与 TBxRG1 匹配的情况下进行从寄存器缓冲区到计时器寄存器(TBxRG0/1)的数据传送。在计数器停止时(即使双缓冲已被启用)，该双缓冲可作为单缓冲区运行，且直接数据可被写入到 TBxRG0 与 TBxRG1。

8.5.4 捕捉

该电路用于控制从 UC 递增计数器传送到 TBxCP0 与 TBxCP1 捕捉寄存器的锁存值的时序。该时序用于锁存数据，并由 TBxMOD<TBCPM [1:0]>指定。

也可利用软件将数值从 UC 递增计数器导入到该捕捉寄存器；具体地说，每将"0"写入到 TBxMOD<TBCP>一次，UC 值就会被纳入该 TBxCP0 捕捉寄存器一次。

8.5.5 捕捉寄存器(TBxCP0, TBxCP1)

该寄存器可捕捉一个递增计数器(UC)值。

8.5.6 递增计数器捕捉寄存器(TBxUC)

除了上述的捕捉外，还可通过读取 TBxUC 寄存器捕捉该 UC 的当前计数值。

8.5.7 比较器(CP0, CP1)

该寄存器可将该递增计数器(UC)与计时器寄存器(TBxRG0 与 TBxRG1)的值设置进行比较，以检测是否存在匹配。如果检测到某一匹配，即生成 INTTBx0 与 INTTBx1。

8.5.8 计时器触发电路(TBxFF0)

源自比较器的单个匹配信号，以及到达各捕捉寄存器的锁存信号，均可导致该计时器触发电路(TBxFF0) 反转。通过设置 TBxFFCR<TBC1T1，TBC0T1，TBE1T1 与 TBE0T1>，即可启用或禁用其反转。

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

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

8.5.9 捕捉中断(INTCAPx0, INTCAPx1)

在从 UC 递增计数器传送到 TBxCP0 与 TBxCP1 捕捉寄存器的锁存值的时序处，可生成中断 INTCAPx0 与 INTCAPx1。中断时序由 CPU 指定。

8.6 各模式的运行说明

8.6.1 16-位间隔计时器模式

如果生成恒定周期中断，则可将间隔时间设置到计时器寄存器(TBxRG0)，以生成 INTTBx0 中断。与 TBxRG0 相同，也可通过将不同的间隔时间值设置到 TBxRG1 计时器电阻器，从而生成 INTTBx1 中断。

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx 运行。
TBxRUN	←	X	X	X	X	X	0	X	0	停止计数运行。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	通过将相应的位设置为"1"，从而允许 INTTBx1 中断。
TBxFFCR	←	X	X	0	0	0	0	1	1	禁用 TBxFF0 反转触发信号。
TBxMOD	←	X	0	1	0	0	1	*	*	改为将预分频输出时钟用作输入时钟。 指定拟禁用的捕捉功能。
							(** = 01, 10, 11)			
TBxRG1	←	*	*	*	*	*	*	*	*	指定一个时间间隔(16 位)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	启动 TMRBx。

注：X; 忽略
−; 无变化

8.6.2 16-位事件计数器模式

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

该递增计数器可在 TBxIN 引脚输入的上升沿实现递增计数。通过用软件进行值捕捉，并读取所捕捉的值，即有可能读取该计数值。

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx 运行。
TBxRUN	←	X	X	X	X	X	0	X	0	停止计数运行。
设置各端口寄存器										将相应的端口分频到 TBxIN。
TBxFFCR	←	X	X	0	0	0	0	1	1	禁用 TBxFF0 反转触发信号。
TBxMOD	←	X	0	1	0	0	0	0	0	将 TBxIN 改为输入时钟。
TBxRUN	←	*	*	*	*	*	1	X	1	启动 TMRBx。
TBxMOD	←	X	0	0	0	0	0	0	0	软件捕捉已完成。

注：X; 忽略
−; 无变化

8.6.3 16-位 PPG (可编程脉冲发生)输出模式

可输出具备任何频率与任何占空比比的方波(可编程方波)。输出脉冲可高激活或低激活。

在递增计数器(UC)的设定值可匹配(TBxRG0 与 TBxRG1)计时器寄存器的设定值时，通过触发计时器触发电路(TBxFF)进行反转，即可从 TBxOUT 引脚输出可编程方波。注意，TBxRG0 与 TBxRG1 的设定值均必须满足以下要求：

TBxRG0 的设定值 ＜TBxRG1 的设定值

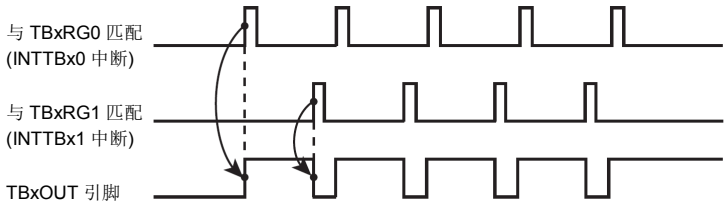


图 8-2 可编程脉冲发生(PPG)输出示例

在该模式下，在递增计数器的设定值匹配 TBxRG1 的设定值时，通过启用 TBxRG0 的双缓冲，即可将寄存器缓冲区 0 的值转入 TBxRG0。这便于对低占空比进行处理。

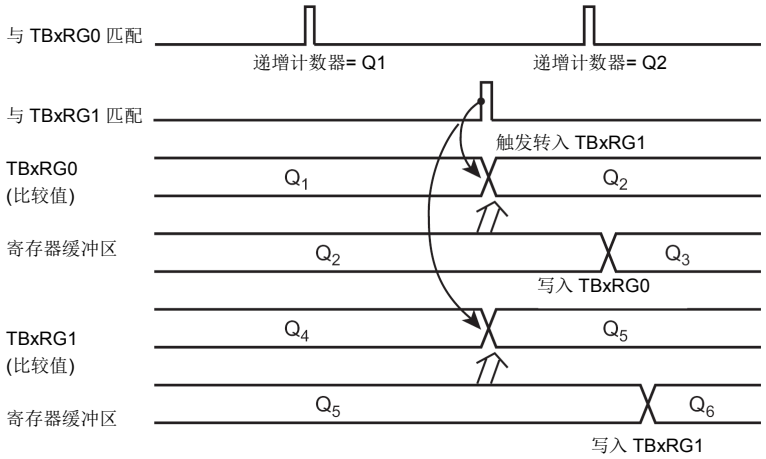


图 8-3 寄存器缓冲区运行

该模式的方块图如以下所示。

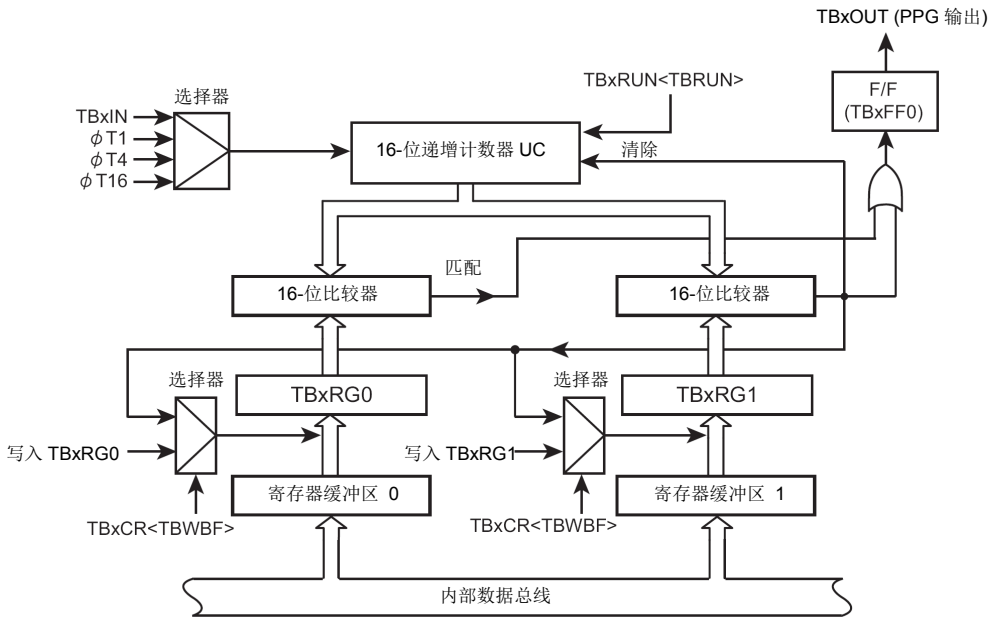


图 8-4 16-位 PPG 模式的方块图

在该 16-位 PPG 输出模式下，各寄存器均必须按以下所列内容接受编程。

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx 运行。
TBxRUN	←	X	X	X	X	X	0	X	0	停止计数运行。
TBxCR	←	0	0	–	X	–	X	0	0	禁用双缓冲。
TBxRG0	←	*	*	*	*	*	*	*	*	指定某占空比(16 位)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	指定某周期(16 位)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	0	X	–	X	0	0	启用 TBxRG0 双缓冲。
										(在生成 INTTBx0 中断时，更改该占空比/周期)
TBxFFCR	←	X	X	0	0	1	1	1	0	指定在检测到与 TBxRG0 或 TBxRG1 之间存在匹配时即触发 TBxFF0 反转，并将 TBxFF0 的初始值设置为"0"。
TBxMOD	←	X	0	1	0	0	1	*	*	指定该预分频输出时钟用作输入时钟，并禁用该捕捉功能。
							(** = 01, 10, 11)			UC 即被清除，以匹配 TBxRG1。
设置各端口寄存器。										将相应的端口分频到 TBxOUT。
TBxRUN	←	*	*	*	*	*	1	X	1	启动 TMRB

注：X：无关
　　–：无变化

8.6.4 外触发器可编程脉冲发生输出模式(PPG)

利用外部计数起动触发器，启用单次脉冲发生短延时。

该 16-位递增计数器(UC)经编程后，可在 TBxIN 引脚(TBxCR [1:0] = "01")的上升沿进行递增计数。TBxRG0 已加载脉冲延时(d)，TBxRG1 已加载 TBxRG0 值(d)与脉冲宽度(p)之和。必须在该 16 位递增计数器被停止期间(TBxRUN<TBRUN> = 0)进行以上设置。

将 TBxFFCR<TBE1T1, TBE0T1>设置为"11"，即可启用计时器触发电路的触发信号。在具备该设置的情况下，该计时器触发电路可在 16-位递增计数器(UC)对应于 TBxRG0 或 TBxRG1 时反转。

将 TBxRUN<TBRUN>设置为"1"，以通过外触发器启用递增计数。

在通过外触发器生成单次脉冲之后，通过 TBxRUN<TBRUN>设置，即可禁用计时器触发电路反转或让 16 位计数器停止运行。

本文此处的符号(d)与(p)，对应于图 8-5 中的符号 d 与 p。

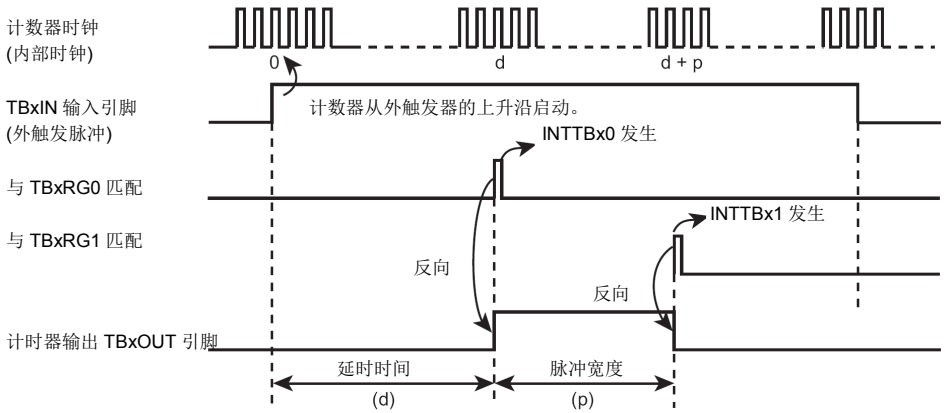


图 8-5 利用外部计数起动触发器实现单次脉冲发生(有延时)

8.7 基于该捕捉功能的应用

该捕捉功能可用于改进多种应用包括以下所述：

1. 由外部脉冲触发的单次脉冲输出
2. 脉冲宽度测量

8.7.1 由外部脉冲触发的单次脉冲输出

由外部脉冲触发的单次脉冲输出的实现方式如下：

利用预分频输出时钟，使 16 位递增计数器在自由运行状态下运行并计数。通过 TBxIN 引脚输入一个外部脉冲。利用该捕捉功能，在该外部脉冲上升时生成一个触发信号，该递增计数器的值随即被纳入该捕捉寄存器(TBxCP0)。

CPU 的编程必须能确保在外触发脉冲上升时生成中断 INTCAPx0。该中断用于将该计时器寄存器(TBxRG0)设置为该 TBxCP0 值(c)与该延时时间(d)， (c + d) 之和，并将该计时器寄存器(TBxRG1) 设置为 TBxRG0 值与单次脉冲的脉冲宽度(p)，(c + d + p) 之和。[必须在下一个匹配之前完成 TBxRG1 的变换]。

此外，必须将该计时器触发电路控制寄存器(TBxFFCR<TBE1T1, TBE0T1>)设置为"11"。这样就可以在 TBxUC 匹配 TBxRG0 与 TBxRG1 时，允许触发该计时器触发电路(TBxFF0)的反转。在单次脉冲被输出之后，该触发信号即被 INTTBx0 / INTTBx1 中断禁用。

本文此处的符号(c)，(d)与(p)，对应于图 8-6 中的符号 c，d 与 p。

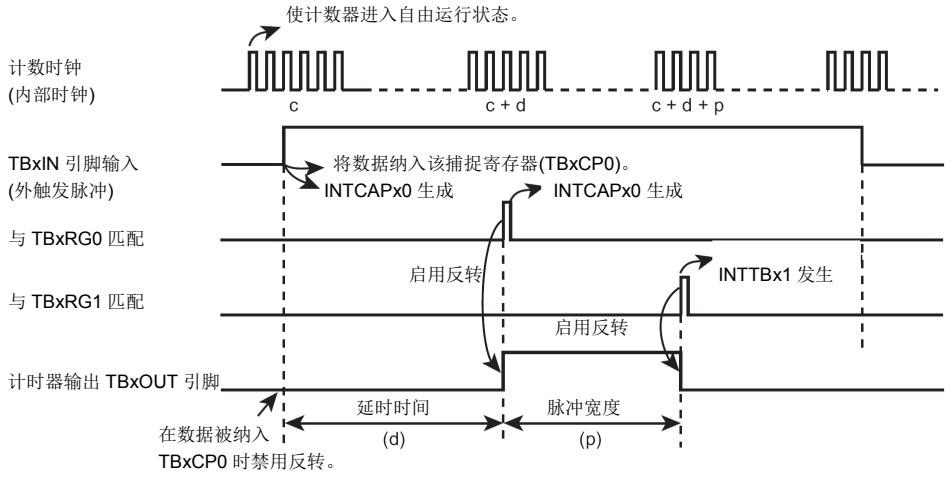


图 8-6 单次脉冲输出(有延时)

以下给出了在通过在上升沿触发 TBxIN 输入，3ms 后输出 2ms 宽度单次脉冲情况下的设置 (ΦT1 被选中用于计数)。

		7	6	5	4	3	2	1	0	
通过 TBxIN 进行的[主处理]捕捉设置										
设置端口寄存器。										将相应的端口分频到 TBxIN。
TBxEN	←	1	X	X	X	X	X	X	X	启用 TMRBx 运行。
TBxRUN	←	X	X	X	X	X	0	X	0	停止计数运行。
TBxMOD	←	X	0	1	0	1	0	0	1	将源时钟改为 ΦT1。将某计数值纳入 TBxIN 上升沿的 TBxCP0。
TBxFFCR	←	X	X	0	0	0	0	1	0	清除 TBxFF0 反转触发信号并禁用。
设置端口寄存器。										将相应的端口分频到 TBxOUT。
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	通过设置为"1"，允许生成 INTCAPx0 中断对应位所指定的中断。
TBxRUN	←	*	*	*	*	*	1	X	1	启动该 TMRBx 模块
[INTCAPx0 中断服务程序的处理]脉冲输出设置										
TBxRG0	←	*	*	*	*	*	*	*	*	设置计数值。(TBxCAP0 + 3ms/ΦT1)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	设置计数值。(TBxCAP0 + (3+2)ms/ΦT1)
	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	–	–	1	1	–	–	如果 UC 与 TBxRG0 andTBxRG1 一致，则反转 TBxFF0。
TBxIM	←	X	X	X	X	X	1	0	1	屏蔽 TBxRG1 通信中断除外
中断设置启用寄存器	←	*	*	*	*	*	*	*	*	通过将相应的位设置为"1"，从而允许生成 INTTBx 中断所指定的中断。
[INTTBx 中断服务程序的处理]禁止输出										
TBxFFCR	←	X	X	–	–	0	0	–	–	清除 TBxFF0 反转触发信号设置。
	←	*	*	*	*	*	*	*	*	通过将相应的位设置为"1"，从而禁止 INTTBx 中断所指定的中断。

注：X; 忽略
–; 无变化

如果不要求延时，则通过生成 INTCAPx0 中断，TBxFF0 会在数据被纳入 TBxCP0 时反转，且 TBxRG1 会被设置为该 TBxCP0 值(c)与该单次脉冲宽度(p)之和 (c + p)。必须在下一个匹配之前完成 TBxRG1 变换。

TBxFF0 在 UC 与 TBxRG1 匹配时被允许反转，并可通过生成 INTTBx1 中断而被禁止反转。

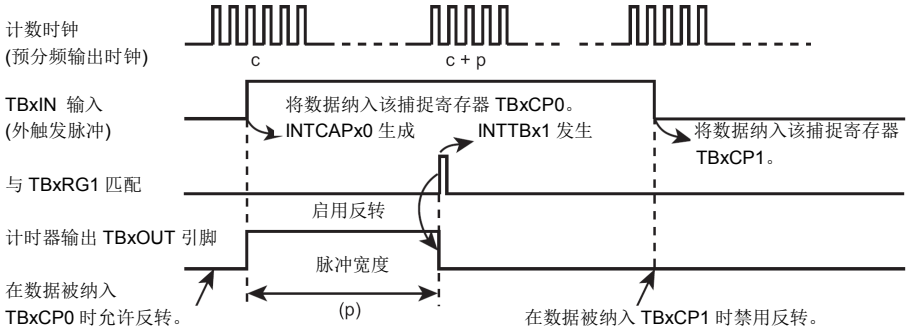


图 8-7 由外部脉冲触发的单次脉冲输出(无延时)

8.7.2 脉冲宽度测量

通过使用该捕捉功能，可测量外部脉冲的"高"电平宽度。具体地说，利用该预分频输出时钟使其进入自由运行状态，通过该 TBxIN 引脚输入一个外部脉冲，该递增计数器(UC)开始计数。利用该捕捉功能，在该外部脉冲的各上升沿与下降沿生成一个触发信号，该递增计数器的值随即被纳入各该捕捉寄存器(TBxCP0，TBxCP1)。必须对 CPU 进行编程以确保在通过 TBxIN 引脚外部脉冲输入的下降沿生成 INTCAPx1。

将 TBxCP0 与 TBxCP1 之差乘以内部时钟的时钟周期，即可计算得出该"高"电平脉冲宽度。

例如，如果 TBxCP0 与 TBxCP1 之差为 100，且预分频输出时钟的周期为 0.5μs，则脉冲宽度为 100 × 0.5 μs = 50 μs。

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

也可测量外部脉冲的"低"电平宽度。在这种情况下，通过执行图 8-8 如所示 INTCAPx0 中断处理的第二电平段，即可得出第一次时所生成 C2 与第二次时所生成 C1 之差，将该差乘以预分频输出时钟的周期即得出该"低"电平宽度。

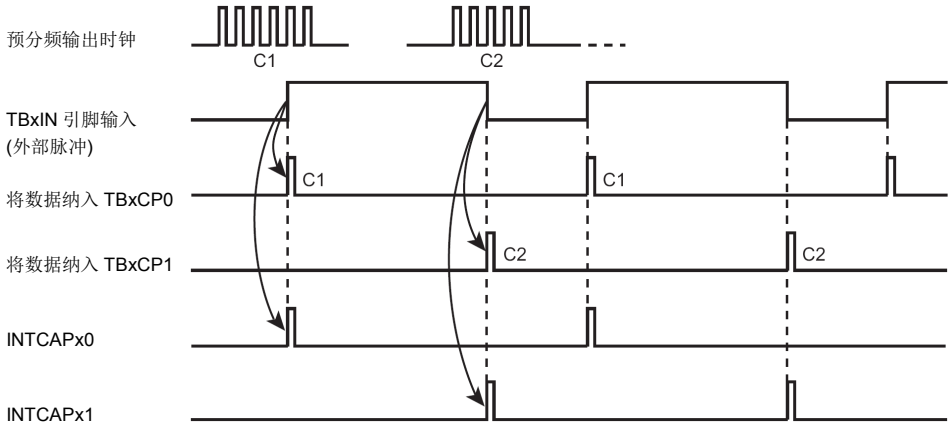


图 8-8 脉冲宽度测量

9. 串行通道(SIO/UART)

9.1 概述

该装置有两种模式可用于该串行通道，其中之一为同步通信模式(I/O 接口模式)，另一种为异步通信模式(UART 模式)。

其特点给出如下。

- 传送时钟
  - 由该预分频从外围时钟 (ϕT0) 频率划分为 1/2, 1/8, 1/32, 1/128。
  - 使得将预分频输出时钟脉冲频率划分成 1-16 成为可能。
  - 使得将预分频输出时钟脉冲频率划分成 1，N+m/16 (N=2-15, m=1-15)成为可能 (仅 UART 模式)。
  - 可用系统时钟(仅 UART 模式)。
- 双缓冲区 /FIFO

  发送与接收的可用双缓冲区功能，以及可用 FIFO 缓冲区，总计最多 4-字节。
- I/O 接口模式
  - 传送模式：半双工(传输/接收)，全双工
  - 时钟：输出(固定上升沿)/输入(可选上升/下降沿)
  - 使指定连续传输的间隔时间成为可能。
- UART 模式
  - 数据长度：7 位，8 位，9 位
  - 加奇偶位(依照 9 位数据长度)
  - 串行链路使用唤醒功能
  - 握手功能(带有 $\overline{\text{CTS}}$ 引脚)

在以下说明中，"x"表示通道编号。

9.2 SIO 模块规格的差异

TMPM370FYDFG/FYFG 具备四个 SIO 通道。

各通道均独立工作。以下列出了所使用的引脚，中断，DMA 请求与各通道中的 UART 源时钟。

表 9-1 SIO 模块规格的差异

	引脚名称			中断		UART 源时钟
	TXD	RXD	$\overline{\text{CTSx}}$ / SCLKx	接收中断	传输中断	
通道 0	PE0	PE1	PE2	INTRX0	INTTX0	TB4OUT
通道 1	PA5	PA6	PA4	INTRX1	INTTX1	TB4OUT
通道 2	PD5	PD6	PD4	INTRX2	INTTX2	TB7OUT
通道 3	PF3	PF4	PF2	INTRX3	INTTX3	TB7OUT

9.3 配置

图 9-1 给出了 SIO 方块图。

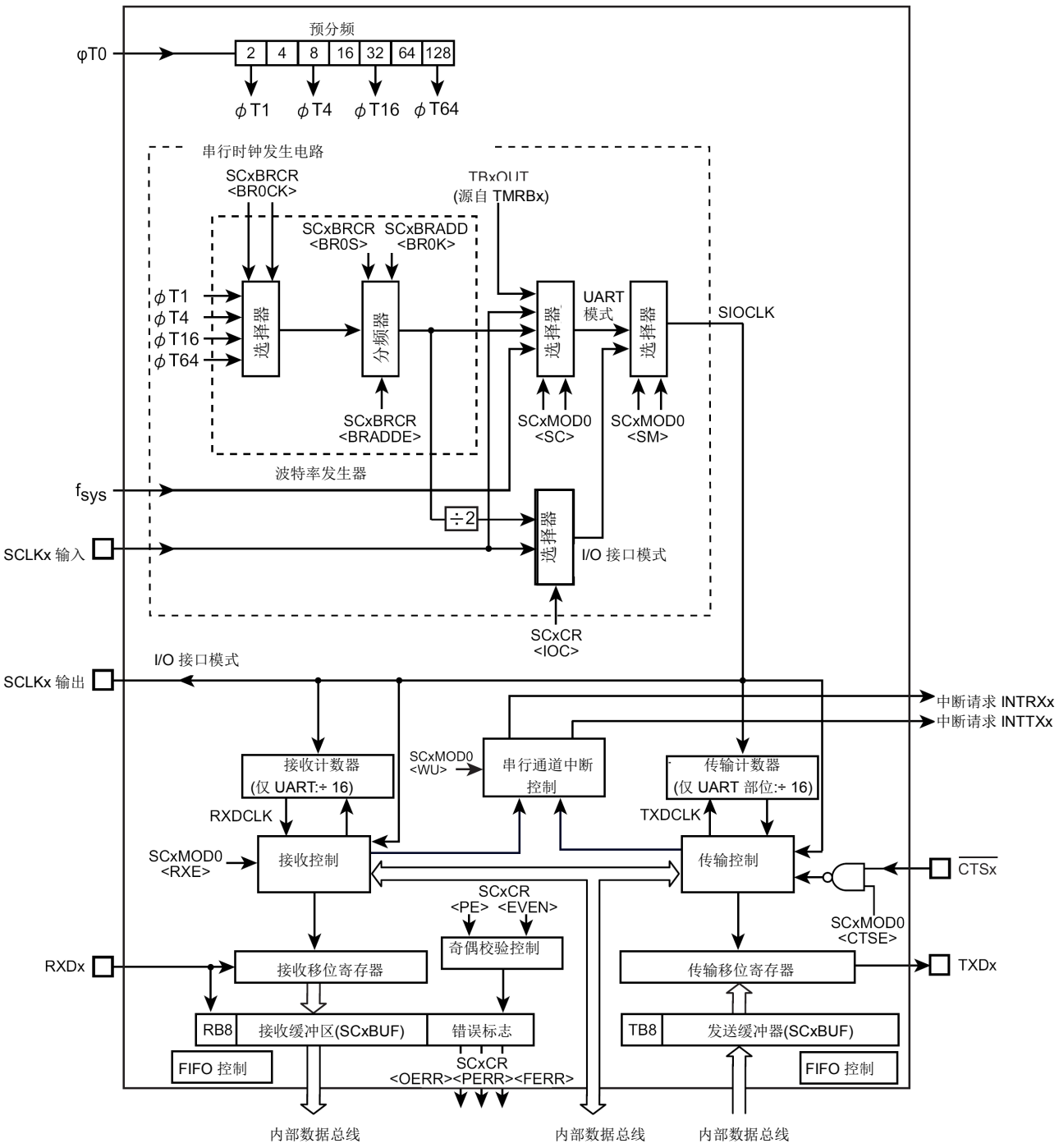


图 9-1 SIO 方块图

9.4 寄存器描述

9.4.1 各通道中的寄存器列表

各通道寄存器与地址给出如下：

通道 x	基址
通道 0	0x4002_0080
通道 1	0x4002_00C0
通道 2	0x4002_0100
通道 3	0x4002_0140

寄存器名称(x=0,1,2,3)		地址(基本+)
启动寄存器	SCxEN	0x0000
缓冲寄存器	SCxBUF	0x0004
控制寄存器	SCxCR	0x0008
模式控制寄存器 0	SCxMOD0	0x000C
波特率发生器控制寄存器	SCxBRCR	0x0010
波特率发生器控制寄存器 2	SCxBRADD	0x0014
模式控制寄存器 1	SCxMOD1	0x0018
模式控制寄存器 2	SCxMOD2	0x001C
RX FIFO 配置寄存器	SCxRFC	0x0020
TX FIFO 配置寄存器	SCxTFC	0x0024
RX FIFO 状态寄存器	SCxRST	0x0028
TX FIFO 状态寄存器	SCxTST	0x002C
FIFO 配置寄存器	SCxFCNF	0x0030

注：不要在数据传输或接收期间修改任何控制寄存器

9.4.2 SCxEN (启用寄存器)

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

位	比特符号	型号	功能
31-1	−	R	读作 0。
0	SIOE	R/W	SIO 运行 0：禁用 1：启用 指定 SIO 运行。 设置<SIOE> = "1"，即可使用该 SIO。 在该运行被禁用时，不会向 SIO 模块中的其它寄存器提供任何时钟脉冲。这样可降低功耗。 如果在执行 SIO 运行后禁止其运行，则各寄存器中均将保持该设置(SCxTFC<TIL[1:0]>除外)。

注： 在 SCxEN<SIOE>被清零(禁用 SIO 运行)，或通过将 SCxMOD1<I2S0>设置为"0"而使运行模式转变为 IDLE 模式时，必须复位 SCxTFC。

9.4.3 SCxBUF (缓冲寄存器)

SCxBUF 可用作写入操作发送缓冲器或 FIFO，以及读出操作的接收缓冲器或 FIFO。

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

位	比特符号	型号:	功能
31-8	−	R	读作 0。
7-0	TB[7:0] / RB[7:0]	R/W	[写入] TB：发送缓冲器/ FIFO [读取] RB：接收缓冲器 / FIFO

9.4.4 SCxCR (控制寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	RB8	R	接收数据位 8(适用于 UART) 9 位 UART 模式下的所接收数据的第 9 位
6	EVEN	R/W	奇偶校验(适用于 UART) 0：奇 1：偶 选择偶或奇校验。 "0"：奇校验, "1" ：偶校验 校验位仅可用于 7-位或 8-位 UART 模式。
5	PE	R/W	加奇偶校验(适用于 UART) 0：禁用 1：启用 可控制启用/禁用奇偶校验。 校验位仅可用于 7-位或 8-位 UART 模式。
4	OERR	R	溢出错误标志(注) 0：正常运行 1：错误
3	PERR	R	奇偶校验/欠载运行错误标志(注) 0：正常运行 1：错误
2	FERR	R	成帧错误标志(注) 0：正常运行 1：错误
1	SCLKS	R/W	选择数据传输与接收的输入时钟脉冲边沿(适用于 I/O 接口) 0：该传输缓冲区中的数据被发送到 SCLKx 下降沿上的 TXDx 引脚一次一位。由该接收缓冲区接收源自 RXDx 引脚的数据(位于 SCLKx 上升沿，一次一位。在这种情况下，SCLK 从高电平启动。 1：该传输缓冲区中的数据被发送到 SCLKx 上升沿上的 TXDx 引脚一次一位。由该接收缓冲区接收源自 RXDx 引脚的数据位于 SCLKx 下降，一次一位。在这种情况下，SCLK 从低电平启动。 在时钟脉冲输出模式下，设置为"0"。
0	IOC	R/W	选择时钟脉冲(适用于 I/O 接口) 0：波特率发生器 1：SCLK 引脚输入

注：**任何错误标志 (OERR, PERR, FERR) 在读取时均被清除为"0"。**

9.4.5 SCxMOD0 (模式控制寄存器 0)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	TB8	R/W	发送数据位 8(适用于 UART) 在 9 位 UART 模式下，写入发送数据的第 9 位。
6	CTSE	R/W	握手功能控制(适用于 UART) 0：CTS 被禁用 1：CTS 被启用 可控制握手功能。 设置"1"可启用握手功能(利用 $\overline{\text{CTS}}$ 引脚)。
5	RXE	R/W	接收控制(注 1)(注 2) 0：禁用 1：启用
4	WU	R/W	唤醒功能(适用于 UART) 0：禁用 1：启用 该功能仅在 9-位 UART 模式下可用。在其它模式下，该功能无意义。 在其处于被启用状态时，"中断"仅适用于 9-位 UART 模式下 RB9="1"时的情况。
3-2	SM[1:0]	R/W	指定传送模式。 00：I/O 接口模式 01：7-位长度 UART 模式 10：8-位长度 UART 模式 11：9-位长度 UART 模式
1-0	SC[1:0]	R/W	串行传送时钟脉冲(适用于 UART) 00：计时器 TBxOUT (请参看表 9-1) 01：波特率发生器 10：内部时钟 fsys 11：外部时钟(SCLK 输入) (至于 I/O 接口模式，可在控制寄存器(SCxCR)中设置串行传送时钟脉冲)

注 1：在<RXE> 被设置为"0"时，设置各模式寄存器(SCxMOD0, SCxMOD1 与 SCxMOD2)。然后将<RXE>设置为"1"。

注 2：不要在数据接收期间停止接收操作(通过设置 SCxMOD0<RXE>="0")。

9.4.6 SCxMOD1 (模式控制寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	I2S0	FDPX		TXE	SINT			-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	I2S0	R/W	IDLE 0：停止 1：操作并指定 IDLE 模式运行。
6-5	FDPX[1:0]	R/W	传送模式设置 00：传送被禁止 01：半双工(接收) 10：半双工(传输) 11：全双工 可在 I/O 接口模式下配置该传送模式。此外还可用于配置 FIFO 如其已被启用。 在 UART 模式下，其仅用于指定 FIFO 配置。
4	TXE	R/W	传输控制器(注 1)(注 2) 0：被禁用 1：启用 该位可启用传输，且在所有传送模式下均有效。
3-1	SINT[2:0]	R/W	连续传输的间隔时间(适用于 I/O 接口) 000：无 001：1SCLK 010：2SCLK 011：4SCLK 100：8SCLK 101：16SCLK 110：32SCLK 111：64SCLK 在已选择 SCLK 引脚的去情况下，该参数仅对 I/O 接口模式有效。在其它模式下，该功能无意义。 在 I/O 接口模式已启用双缓冲或 FIFO 的情况下，可指定连续传输的间隔时间。
0	−	R/W	写作 "0"。

注 1：**首先指定所有模式，然后启用该<TXE>位。**

注 2：**不要在数据传输期间停止传输操作(通过设置<TXE>="0")。**

注 3: 在 SCxEN<SIOE>被清为"0"(禁用 SIO 运行)，或通过将 SCxMOD1<I2S0>设置为"0"而使运行模式转变为 IDLE 模式时，必须复位 SCxTFC。

9.4.7 SCxMOD2 (模式控制寄存器 2)

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

<table>
<tr><th>位</th><th>比特符号</th><th>型号</th><th>功能</th></tr>
<tr><td>31-8</td><td>－</td><td>R</td><td>读作 0。</td></tr>
<tr><td>7</td><td>TBEMP</td><td>R</td><td>传输缓冲区为空标志
0：满
1：空
如果双缓冲被禁用，则该标志可忽略。
该标志表示该传输双缓冲区为空。在该传输双缓冲区中的数据被移动到传输移位寄存器，且该双缓冲区为空时，可将该位设置为"1"。
通过将数据重新写入到该双缓冲区，即可将该位设置为"0"。</td></tr>
<tr><td>6</td><td>RBFLL</td><td>R</td><td>接收缓冲器已满标志。
0：空
1：满
如果双缓冲被禁用，则该标志可忽略。
该标志表示接收双缓冲区已满。
在接收操作已完成，且所接收的数据被从接收移位寄存器移到该接收双缓冲区时，该位会变为"1"，而读取该位即可将该位更改为"0"。</td></tr>
<tr><td>5</td><td>TXRUN</td><td>R</td><td>在传输标志中
0：停止
1：操作
该状态标志表示数据传输正在进行中。
<TXRUN>与<TBEMP>位可指示以下状态。
<table>
<tr><th><TXRUN></th><th><TBEMP></th><th>状态</th></tr>
<tr><td>1</td><td>－</td><td>传输正在进行中</td></tr>
<tr><td rowspan="2">0</td><td>1</td><td>传输已完成</td></tr>
<tr><td>0</td><td>等待状态(数据在传输缓冲区中)</td></tr>
</table></td></tr>
<tr><td>4</td><td>SBLEN</td><td>R/W</td><td>停止位(适用于 UART)
0：1-位
1：2-位
这样可在 UART 模式下指定传输停止位的长度。
对于接收侧而言，仅利用与<SBLEN>设置无关的单个位即可做出该判定。</td></tr>
<tr><td>3</td><td>DRCHG</td><td>R/W</td><td>设置传送方向
0：首先 LSB
1：首先 MSB
指定 I/O 接口模式下的数据传送方向。
在 UART 模式下，首先将该位设置为 LSB。</td></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>双缓冲区
0：禁用
1：启用
该参数可允许或禁止该传输/接收双缓冲区在 I/O 接口模式下传输(在 SCLK 输出/输入模式下)与接收(在 SCLK 输出模式下)数据，以及在 UART 模式下发送数据。
在接口模式(SCLK 输入)与 UART 模式下接收数据时，双缓冲在 0 或 1 被设置到为<WBUF>位时均会被启用。</td></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>软件复位
用"01"盖写"10"即可触发软件复位。在执行该软件复位时，以下位即被初始化：
<table>
<tr><th>寄存器</th><th>位</th></tr>
<tr><td>SCxMOD0</td><td><RXE></td></tr>
<tr><td>SCxMOD1</td><td><TXE></td></tr>
<tr><td>SCxMOD2</td><td><TBEMP>, <RBFLL>, <TXRUN></td></tr>
<tr><td>SCxCR</td><td><OERR>, <PERR>, <FERR></td></tr>
</table>
该传输/接收电路与 FIFO 变为初始状态(见注 1 与注 2)。</td></tr>
</table>

注 1：在数据传输进行期间，任何软件复位操作均必须连续执行两次。

注 2：在进行软件复位时，软件复位指令的识别结束与执行开始之间的持续时间必须达到 2 个时钟脉冲。

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

可在以下所列的寄存器中指定波特率发生器的分频比。

SCxBRCR

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	BRADDE	BR0CK		BR0S			
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	−	R/W	写入"0"。
6	BRADDE	R/W	N + (16 − K)/16 分频器功能(适用于 UART) 0：禁用 1：启用 该分频功能仅可用于 UART 模式。
5-4	BR0CK[1:0]	R/W	选择输入时钟到该波特率发生器 00：ϕT1 01：ϕT4 10：ϕT16 11：ϕT64
3-0	BR0S[3:0]	R/W	分频比"N" 0000：16 0001：1 0010：2 ... 1111：15

SCxBRADD

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	BR0K			
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-0	BR0K[3:0]	R/W	指定划分"N + (16 − K)/16"的 K(适用于 UART) 0000：被禁止 0001：K = 1 0010：K = 2 ... 1111：K = 15

表 9-2 列出了波特率发生器分频比的设置情况。

表 9-2 设置分频比

	<BRADDE> = "0"	<BRADDE> = "1" (注 1) (仅 UART 模式)
<BR0S>	指定"N" (注 2) (注 3)	
<BR0K>	无设置要求	指定"K" (注 4)
分频比	除以 N	除式 $N + \frac{(16-k)}{16}$

注 1：**在使用 "N + (16 - K)/16" 分频功能时，务必在将 K 值设置为<BR0K>之后将<BRADDE>设置为 "1"。该"N + (16 - K)/16" 分频功能仅可用于 UART 模式。**

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

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

注 4：**禁止指定"K = 0"。**

9.4.9 SCxFCNF (FIFO 配置寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作 0
7-5	-	R/W	保证写入"000"
4	RFST	R/W	在 RX FIFO 中所用的字节 0：最大值 1：与 RX FIFO 的 FILL 率相同 当 RX FIFO 启用时，选择使用的 RX FIFO 字节数(注 1) 0：配置的 FIFO 最大字节数(另见<CNFG>)。 1：与 SCxRFC <RIL[1:0]>规定的接收中断生成的充满率相同
3	TFIE	R/W	TX FIFO 的 TX 中断 0：禁用 1：启用 当 TX FIFO 启用时，发送中断由该参数启用或禁用。
2	RFIE	R/W	RX FIFO 的 RX 中断 0：禁用 1：启用 当 RX FIFO 启用时，接收中断由该参数启用或禁用。
1	RXTXCNT	R/W	<RXE>/<TXE>自动禁用 0：无 1：自动禁用 控制发送和接收的自动禁用。 设置"1"能实现下列操作 半双工 RX：当接收移位寄存器，接收缓冲区和 RX FIFO 被充满时，SCxMOD0<RXE>自动设置为"0"，以禁止进一步的接收。 半双工 TX：当 TX FIFO，发送缓冲区和发送移位寄存器为空时，SCxMOD1<TXE>自动设置为"0"，以禁止进一步的发送。 全双工：当上述两个条件中的任何一个条件得到满足时，<TXE>/<RXE>自动设置为"0"，以禁止进一步的发送和接收。
0	CNFG	R/W	启用 FIFO (注 2) 0：禁用 1：启用 启用时，SCxMOD1 <FDPX[1: 0]> 设置项自动配置 FIFO 如下： (TX/RX 的类型能在模式控制寄存器 1 SCxMOD1<FDPX[1: 0]>中作出规定)。 半双工 RX：RX FIFO 4 字节 半双工 TX：TX FIFO 4 字节 全双工：RX FIFO 2 字节 + TX FIFO 2 字节

注 1：**关于 TX FIFO，配置的最大字节数总是可用的。可用字节数指已写入 TX FIFO 的字节。**

注 2：**FIFO 无法在 9 位 UART 模式中使用。**

9.4.10 SCxRFC (RX FIFO 配置寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	RFCS	W	RX FIFO 清除(注) 1：清除 RX FIFO 当 SCxRFC<RFCS>设置为"1"时，FIFO 被清除，SCxRST<RLVL>为"000"。 读指针也被初始化。
6	RFIS	R/W	选择中断生成条件 0：当数据达到规定的充满率时，生成中断。 1：当数据达到规定的充满率或者在数据读取时，数据超过规定的充满率时，生成中断。
5-2	−	R	读作 0。
1-0	RIL[1:0]	R/W	生成 RX 中断的 FIFO 充满率

	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： 使用 TX/RX FIFO 缓冲区时，在设置 SIO 传输模式 (半双工/全双工)并启用 FIFO (SCxFCNF<CNFG> = "1")后，必须清除 TX/RX FIFO。

9.4.11 SCxTFC (TX FIFO 配置寄存器) (注 2)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	TFCS	W	TX FIFO 清除(注 1) 1：清除 TX FIFO。 当 SCxTST<TFCS>设置为"1"时，发送 FIFO 被清除，SCxRST<TLVL>为"000"。 写指针也被初始化。
6	TFIS	R/W	选择中断生成条件。 0：当数据达到规定的充满率时，生成中断。 1：当数据达到规定的充满率或者在新数据读取之际，数据无法达到规定的充满率时，生成中断。
5-2	−	R	读作 0。
1-0	TIL[1:0]	R/W	生成 TX 中断的 FIFO 充满率。

	非全双工	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：**使用 TX/RX FIFO 缓冲区时，在设置 SIO 传输模式 (半双工/全双工)并启用 FIFO (SCxFCNF<CNFG> = "1")后，必须清除 TX/RX FIFO。**

注 2：**在进行下列操作后，再次配置 SCxTFC 寄存器。SCxEN<SIOE> = "0" (SIO 操作停止) 条件如下: SCxMOD1<I2S0> = "0" (在 IDLE 模式时禁止操作)，并释放 WFI (等待中断)指令所启动的低功耗模式。**

9.4.12 SCxRST (RX FIFO 状态寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	ROR	R	RX FIFO 溢出(注) 0：未生成 1：生成
6-3	−	R	读作 0。
2-0	RLVL[2:0]	R	RX FIFO 充满率状态。 000：空 001：1 字节 010：2 字节 011：3 字节 100：4 字节

注：**当从 SCxBUF 寄存器读取接收数据时，<ROR>位被清除为"0"。**

9.4.13 SCxTST (TX FIFO 状态寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	TUR	R	TX FIFO 欠载(注) 0：未生成 1：生成。
6-3	−	R	读作 0。
2-0	TLVL[2:0]	R	TX FIFO 充满率状态。 000：空 001：1 字节 010：2 字节 011：3 字节 100：4 字节

注： **当发送数据被写入 SCxBUF 寄存器时，<TUR>位被清除为"0"。**

9.5 在各模式时的操作

模式和数据格式如表 9-3 所示。

表 9-3 模式和数据格式

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

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

可从 LSB 先和 MSB 先中选择数据传输方向。 该模式不得使用奇偶校验位或 STOP 位。

模式 1 ，模式 2 和模式 3 为异步模式，传输方向固定为 LSB 先。

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

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

9.6 数据格式

9.6.1 数据格式表

数据格式如图 9-2 所示。

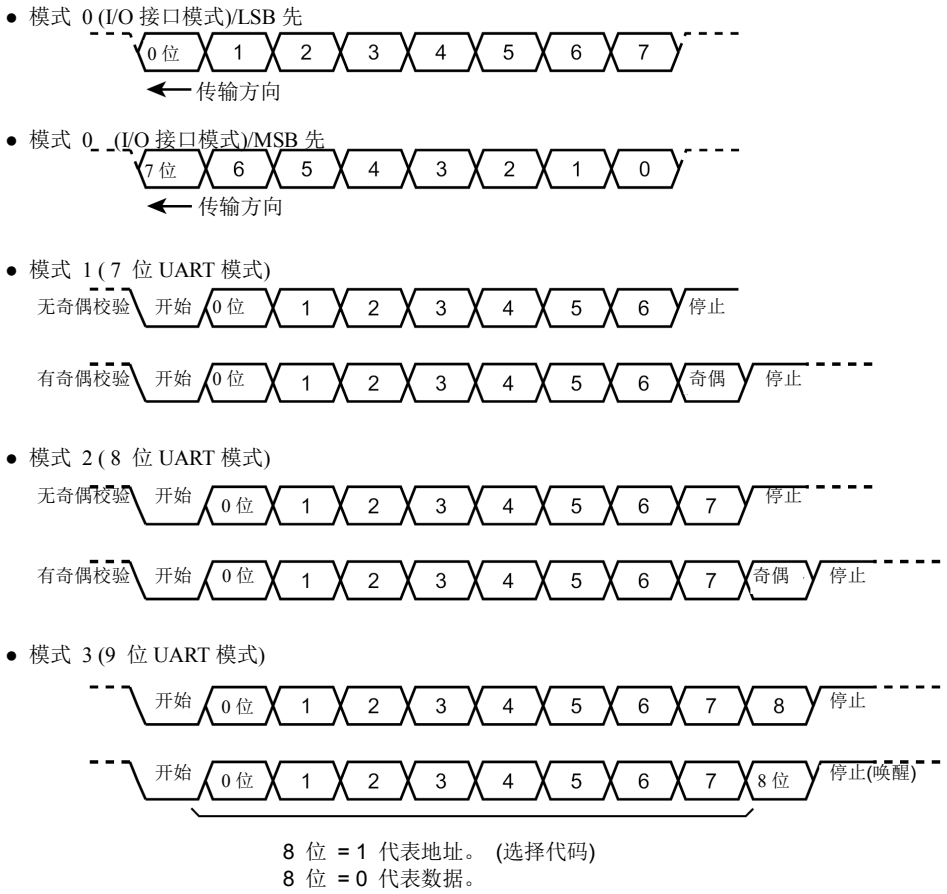


图 9-2 数据格式

9.6.2 奇偶校验控制

奇偶校验位仅能在 7 位或 8-位 UART 模式下添加。

将 SCxCR<PE>设置为"1"就可启用奇偶校验。

SCxCR 的<EVEN> 位选择偶校验或奇校验。

9.6.2.1 传输

在数据传输后，奇偶控制电路自动生成奇偶校验，数据位于发送缓冲区。

在数据传输完成后，奇偶校验位在 7-位 UART 模式时储存在 SCxBUF<TB7>中，在 8-位 UART 模式时储存在 SCxMOD0<TB8>中。

<PE>和<EVEN> 的设置必须在数据写入发送缓冲区之前完成。

9.6.2.2 接收数据

若接收的数据从接收移位寄存器移到接收缓冲区，则会生成奇偶校验。

在 7-位 UART 模式时，生成的奇偶校验与 SCxBUF<RB7>中储存的奇偶校验比较，而在 8-位 UART 模式时，它与 SCxCR<RB8>中储存的奇偶校验比较。

若有任何差异，就会发生奇偶校验错误，SCxCR 寄存器中的<PERR>设置为"1"。

在使用 FIFO 时，<PERR>表示在接收的数据之一中生成奇偶校验错误。

9.6.3 STOP 位长度

通过设置 SCxMOD2<SBLEN>，可从一位或两位中选择 UART 传输模式时的 STOP 位长度。 不管该位的设置，停止位数据的长度在它被接收时被确定为一位。

9.7 时钟控制

9.7.1 预分频器

有一个把预分频器输入时钟 ΦT0 除以 2, 8, 32 和 128 的 7-位预分频器。

用时钟/模式控制块中的 CGSYSCR 寄存器选择预分频器的输入时钟 ΦT0。

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

波特率发生器输入时钟分辨率如表 9-4 所示。

表 9-4 波特率发生器输入时钟分辨率 fc = 80 MHz

外围 时钟选择 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR [2:0]>	预分频器时钟选择 CGSYSCR <PRCK[2:0]>	预分频器输出时钟分辨率			
			ϕT1	ϕT4	ϕT16	ϕT64
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025 μs)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		001 (fperiph/2)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		010 (fperiph/4)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		011 (fperiph/8)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		001 (fperiph/2)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		010 (fperiph/4)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		011 (fperiph/8)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		100 (fperiph/16)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
		101 (fperiph/32)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)	$fc/2^{14}$ (204.8 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		001 (fperiph/2)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)	$fc/2^{14}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)	$fc/2^{15}$ (409.6 μs)
	111 (fc/16)	000 (fperiph/1)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		001 (fperiph/2)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		010 (fperiph/4)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
		011 (fperiph/8)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)	$fc/2^{14}$ (204.8 μs)
		100 (fperiph/16)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)	$fc/2^{15}$ (409.6 μs)
		101 (fperiph/32)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)	$fc/2^{14}$ (204.8 μs)	$fc/2^{16}$ (819.2 μs)

表 9-4 波特率发生器输入时钟分辨率 fc = 80 MHz

外围 时钟选择 CGSYSCR <FPSEL>	时钟齿轮值 CGSYSCR <GEAR[2:0]>	预分频器时钟 选择 CGSYSCR <PRCK[2:0]>	预分频器输出时钟分辨率			
			ϕT1	ϕT4	ϕT16	ϕT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.025 μs)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		001 (fperiph/2)	fc/2^2 (0.05 μs)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		001 (fperiph/2)	fc/2^2 (0.05 μs)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		001 (fperiph/2)	–	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		001 (fperiph/2)	–	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		010 (fperiph/4)	–	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		001 (fperiph/2)	–	–	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		010 (fperiph/4)	–	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		011 (fperiph/8)	–	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)

注 1：**必须选择预分频器输出时钟 ϕTn，以便满足关系式"ϕTn ≤ fsys / 2" (以便 ϕTn 慢于 fsys)。**

注 2：**当 SIO 正运行时，不得改变时钟齿轮。**

注 3：**上表中的破折号表示设置被禁止。**

9.7.2 串行时钟发生电路

串行时钟电路是一个发送和接收时钟(SIOCLK)发生模块，由通过设置波特率发生器和模式就可选择时钟的电路组成。

9.7.2.1 波特率发生器

波特率发生器生成发送和接收时钟，以确定串行通道传输率。

(1) 波特率发生器输入时钟

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

通过设置 SCxBRCR<BRCK>，便可选择该输入时钟。

(2) 波特率发生器输出时钟

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

可采用下列分频比；在 I/O 接口模式时 1/N 分频，在 UART 模式时 1/N 或 N+(16 – K)/16。能选择的分频比如下表所示。

模式	分频功能设置 SCxBRCR<BRADDE>	除以 N SCxBRCR<BR0S>	除以 K SCxBRADD<BR0K>
I/O 接口	除以 N	1 ~ 16 (注)	-
UART	除以 N	1 ~ 16	-
	N + (16 – K) / 16 分频	2 ~ 15	1 ~ 15

注：只有在双缓冲区启用时，才能采用 1 / N (N = 1)分频比。

9.7.2.2 时钟选择电路

通过设置模式和寄存器选择时钟。

通过设置 SCxMOD0<SM>规定模式。

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

(1) 在 I/O 接口模式时的传输时钟

在 I/O 接口模式时的时钟选择如表 9-5 所示。

表 9-5 在 I/O 接口模式时的时钟选择

模式 SCxMOD0<SM>	输入/输出选择 SCxCR<IOC>	时钟沿选择 SCxCR<SCLKS>	使用的时钟
I/O 接口模式	SCLK 输出	设置为"0"。 (固定为上升沿)	波特率发生器输出除以 2。
	SCLK 输入	上升沿	SCLK 输入上升沿
		下降沿	SCLK 输入下降沿

为了获得最高波特率，波特率发生器必须设置如下。

注：**当决定时钟设置时，确保 AC 电气特征得到满足。**

• 时钟/模式控制块设置

- fc = 80 MHz
- fgear = 80 MHz (CGSYSCR<GEAR[2:0]> = "000" : fc 被选中)
- ϕT0 = 80 MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 分频比)

• SIO 设置 (若采用双缓冲区)

- 时钟(SCxBRCR<BRCK[1:0]> = "00": ϕT1 被选中) = 40 MHz
- 时钟分频(SCxBRCR<BRS[3:0]> = "0001" : 1 分频比) = 40 MHz

若采用双缓冲区，则可选择 1 分频比。在这种情况下，因为 40 MHz 除以 2 ，所以波特率为 20 Mbps。

• SIO 设置(若未采用双缓冲区)

- 时钟 (SCxBRCR<BRCK[1:0]> = "00" : ϕT1 被选中) = 40MHz
- 时钟分频 (SCxBRCR<BRS[3:0]> = "0010" : 2 分频比) = 20 MHz

若未采用双缓冲区，2 分频比为最高值。在这种情况下，因为 20 MHz 除以 2，所以波特率为 10 Mbps。

为了采用 SCLK 输入，必须满足下列条件。

• 当使用双缓冲区时

- SCLK 周期 > 6 / fsys

最高波特率小于 80 ÷ 6 = 13.3 Mbps。

• 若没有使用双缓冲器

- SCLK 周期 > 8 / fsys

最高波特率小于 80 ÷ 8 = 10 Mbps。

(2) 在 UART 模式下的传输时钟

在 UART 模式时的时钟选择如表 9-6 所示。在 UART 模式下，在使用前在接收计数器或发送计数器中将所选时钟除以 16 。

表 9-6 在 UART 模式下的时钟选择

模式 SCxMOD0<SM>	时钟选择 SCxMOD0<SC>
UART 模式	定时器输出
	波特率发生器
	fsys
	SCLK 输入

各时钟设置中波特率的举例。

• 若使用波特率发生器

- fc = 80MHz

-fgear = 80MHz (CGSYSCR<GEAR[2:0]> = "000" : fc 被选中)

- ϕT0 = 80MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 分频比)

- 时钟 = ΦT1 = 40 MHz (SCxBRCR<BRCK[1:0]> = "00" : ϕT1 被选中)

40 MHz 除以 16 ，最高波特率为 2.5 Mbps。

当波特率发生器与下列时钟设置一起使用时，波特率的示例如表 9-7 所示。

• fc = 9.8304 MHz

• fgear = 9.8304 MHz (CGSYSCR<GEAR[2:0]> = "000" : fc 被选中)

• ϕT0 = 4.9152 MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 分频比)

表 9-7 UART 模式时下波特率的例子(使用波特率发生器)

fc [MHz]	分频比 N (SCxBRCR<BRS[3:0]>)	ϕT1 (fc/4)	ϕT4 (fc/16)	ϕT16 (fc/64)	ϕT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	16	9.600	2.400	0.600	0.150

单位：kbps

• 若使用 SCLK 输入

为了采用 SCLK 输入，必须满足下列条件。

- SCLK 周期 > 2 / fsys

最高波特率必须小于 80 ÷ 2 ÷ 16 = 2.5 Mbps。

• 若使用 fsys

因为 fsys 的最高值为 80 MHz，所以最高波特率为 80 ÷ 16 = 5 Mbps。

• 若使用定时器输出

启用定时器输出时，必须设置下列条件：当计数器的值与 TBxRG1 的值匹配时定时器触发器的输出反向。SIOCLK 时钟频率为"TBxRG1 设置值 × 2 "。

波特率由下述公式计算所得。

波特率的计算

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

在选择定时器预分频器时钟 ϕT1 (2 分频比)的情况下。

一个时钟周期为定时器触发器反向两次的时间。

当定时器输出与下列时钟设置一起使用时，波特率的示例如表 9-8 所示。

- fc = 80 MHz / 9.8304 MHz / 8 MHz
- fgear = 80 MHz / 9.8304 MHz / 8 MHz (CGSYSCR<GEAR[2:0]> = "000" : fc 被选中)
- ϕT0 = 40 MHz / 4.9152 MHz / 4 MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 分频比)
- 定时器计数时钟 = 4 MHz / 1.2287 MHz / 1 MHz (TBxMOD<TBCLK[1: 0]> = "01" : ϕT1 被选中)

表 9-8 UART 模式时下波特率的例子(使用定时器输出)

TBxRG0 的设置	fc		
	80MHz	9.8304MHz	8MHz
0x0001	625	76.8	62.5
0x0002	312.5	38.4	31.25
0x0003	-	25.6	-
0x0004	156.25	19.2	15.625
0x0005	125	15.36	12.5
0x0006	-	12.8	-
0x0008	78.125	9.6	-
0x000A	62.5	7.68	6.25
0x0010	39.025	4.8	-
0x0014	31.25	3.84	3.125

单位：kbps

9.8 发送/接收缓冲器和 FIFO

9.8.1 配置

发送缓冲器，接收缓冲器和 FIFO 的配置如图 9-3 所示。

使用缓冲器和 FIFO，必须进行适当的设置。可按照模式对配置进行预定义。

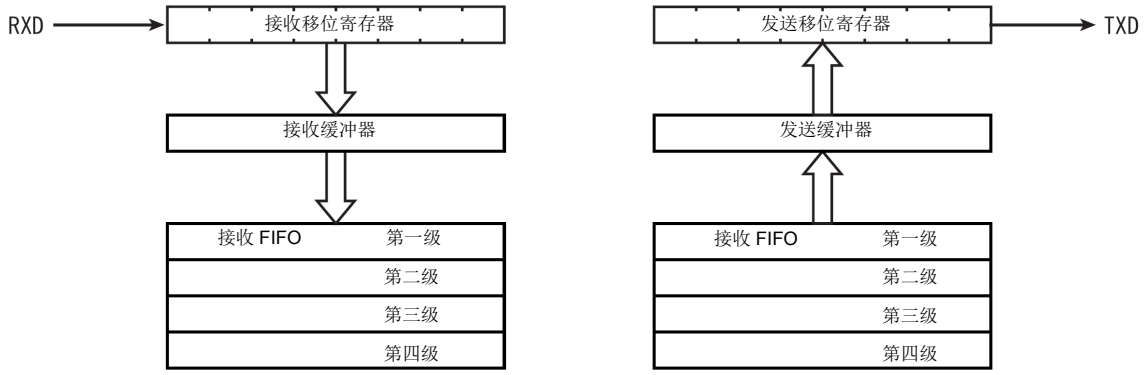


图 9-3 缓冲器和 FIFO 的配置

9.8.2 发送/接收缓冲器

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

在使用接收缓冲器的情况下，若在 I/O 接口模式时为了生成时钟输出而设置了 SCLK 输入，或者若选择了 UART 模式，则尽管有 <WBUF> 设置，仍会对其进行双缓冲。在其他模式时，它按照<WBUF>的设置。模式和缓冲器之间的相关性如表 9-9 所示。

表 9-9 模式和缓冲器的组成

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

9.8.3 FIFO

除上述双缓冲功能外，还能使用 4-字节 FIFO。

为了启用 FIFO，应通过将 SCxMOD2<WBUF>设置为"1"及将 SCxFCNF<CNFG>设置为"1"，启用双缓冲器。FIFO 缓冲器的配置由 SCxMOD1<FDPX[1:0]>规定。

注： 使用 TX/RX FIFO 缓冲区时，在设置 SIO 传输模式 (半双工/全双工)并启用 FIFO (SCxFCNF<CNFG> = "1")后，必须清除 TX/RX FIFO。

模式和 FIFO 之间的相关性如表 9-10 所示。

表 9-10 模式和 FIFO 的组成

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

9.9 状态标志

SCxMOD2 寄存器有两类标志。只有在启用双缓冲器时，该位才有效。

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

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

9.10 错误标志

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

在读取 SCxCR 寄存器后，这些标志被清除为"0"。

模式	标志		
	<OERR>	<PERR>	<FERR>
UART	溢出错误	奇偶校验错误	成帧错误
I/O 接口 (SCLK 输入)	溢出错误	欠载运行错误 (当使用双缓冲器或 FIFO 时)	固定为 0
		固定为 0 (当未使用双缓冲器和 FIFO 时)	
I/O 接口 (SCLK 输出)	未定义	未定义	固定为 0

9.10.1 OERR 标志

UART 和 I/O 接口模式下，在完成读取接收缓冲器前，当通过完成下一帧接收数据的接收而出错时，该位设置为"1"。启用接收 FIFO 时，接收的数据自动移至接收 FIFO，不会产生溢出错误，直到接收 FIFO 已满(或者直到可用的字节被完全占用)。

带 SCLK 输出的 I/O 接口模式下，SCLK 输出在设置标志后停止。

注：将 I/O 接口 SCLK 输出模式切换到其他模式时，应读取 SCxCR 寄存器，并清除溢出标志。

9.10.2 PERR 标志

UART 模式时下，该标志表示奇偶校验错误；I/O 接口模式下，该标志表示欠载运行错误。

UART 模式下，当接收的数据产生的奇偶校验不同于接收的奇偶校验时，将<PERR>设置为"1"。

I/O 接口模式下，启用双缓冲器，在下列条件下将<PERR>设置为"1"。

SCLK 输入模式下，在完成发送移位寄存器的数据输出，并且在发送缓冲器中无数据后，当输入 SCLK 时，将<PERR>设置为"1"。

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

注：为了将 I/O 接口 SCLK 输出模式切换到其他模式时，应读取 SCxCR 寄存器，并清除欠载运行标志。

9.10.3 FERR 标志

若在中心周围对相应的停止位取样，该位被确定为"0"，则会产生成帧错误。不管 SCxMOD2<SBLEN>寄存器中的停止位长度设置，停止位状态仅由 1 确定。

在 I/O 接口模式下，该位固定至"0"。

9.11 接收

9.11.1 接收计数器

接收计数器为 4-位二进制计数器，并由 SIOCLK 向上计数。在 UART 模式时， 16 个 SIOCLK 时钟脉冲用于接收单一数据位，在第七，第八和第九个脉冲对数据符号取样。从这三个样本中，多数逻辑用于决定接收的数据。

9.11.2 接收控制器

9.11.2.1 I/O 接口模式

在 SCLK 输出模式下且 SCxCR <IOC> 设置为"0"时，在向 SCLK 引脚输出的移位时钟的上升沿，对 RXD 引脚取样。

在 SCLK 输入模式下且 SCxCR <IOC> 设置为"1"时，按照 SCxCR <SCLKS>的设置在 SCLK 输入信号的上升沿或下降沿对串行接收数据 RXD 引脚取样。

9.11.2.2 UART 模式

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

9.11.3 接收操作

9.11.3.1 接收缓冲器

接收的数据按 1 位储存在接收移位寄存器中。当储存一整套位完成后，会产生中断 INTRXx。

启用双缓冲器时，数据移到接收缓冲器(SCxBUF) ，接收缓冲器全满标志(SCxMOD2<RBFLL>)设置为"1"。通过读取接收缓冲器，接收缓冲器全满标志被清除为"0"。关于单个缓冲器，接收缓冲器全满标志无数值。

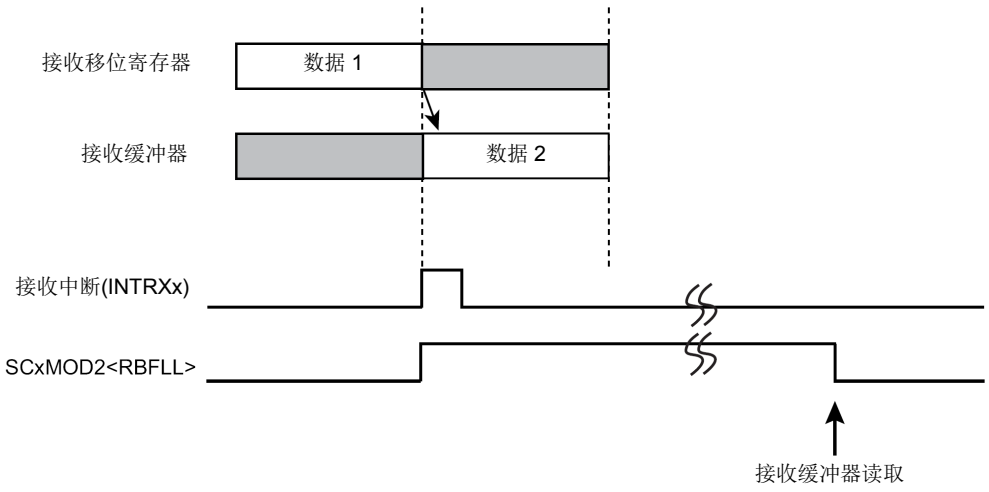


图 9-4 接收缓冲器操作

9.11.3.2 接收 FIFO 操作

启用 FIFO 时，接收的数据从接收缓冲器移到接收 FIFO，接收缓冲器全满标志被立即清除。按照 SCxRFC<RIL>的设置，会产生中断。

注： UART 模式下使用 FIFO 接收带奇偶校验位的数据时，会显示奇偶校验错误标志，表示在接收的数据中发生奇偶校验错误。

半双工 RX 模式下的配置和操作说明如下。

SCxMOD1[6:5] = 01	：传输模式设置为半双工模式
SCxFCNF[4:0] = 10111	：在达到充满率后，自动禁止连续接收。
	：在接收 FIFO 中使用的字节数与中断生成充满率相同。
SCxRFC[1:0] = 00	：生成的接收中断设置为 4-字节的 FIFO 的充满率。
SCxRFC[7:6] = 11	：清除接收 FIFO，并设置中断生成的条件。

在设置上述 FIFO 配置后，将"1"写入 SCxMOD0 <RXE>，可启动数据接收。当数据均被储存在接收移位寄存器，接收缓冲器和接收 FIFO 时，SCxMOD0<RXE>被自动清除，完成接收操作。

在上述条件下，若在达到充满率后启用连续接收，并且有可能连续接收数据，而数据位于 FIFO，并读取 FIFO 中的数据。

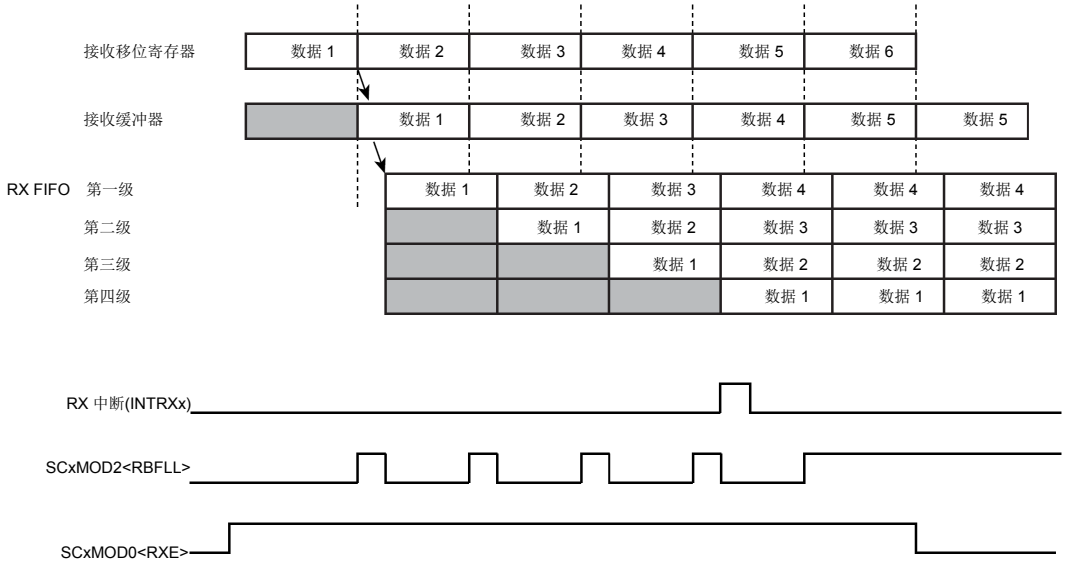


图 9-5 接收 FIFO 的操作

9.11.3.3 带 SCLK 输出的 I/O 接口模式

在 I/O 接口模式和 SCLK 输出设置中，当所有接收的数据储存在接收缓冲器和 FIFO 时，SCLK 输出停止。因此，此模式下，溢出错误标志无意义。

SCLK 输出停止和再输出的时间取决于接收缓冲器和 FIFO。

(1) 单缓冲器的情况

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

当读取缓冲器中的数据时，SCLK 输出被重启。

(2) 双缓冲器的情况

在数据被接收至接收移位寄存器和接收缓冲器后，停止 SCLK 输出。

当读取数据时，SCLK 输出被重启。

(3) FIFO 的情况

在数据被接收至移位寄存器，接收缓冲器和 FIFO 后，停止 SCLK 输出。

当读取一字节数据时，接收缓冲器中的数据被传输到 FIFO，接收移位寄存器中的数据被传输到接收缓冲器，SCLK 输出被重启。

若 SCxFCNF<RXTXCNT>设置为"1"，SCLK 停止，接收操作也停止，并清除 SCxMOD0<RXE>位。

9.11.3.4 读取接收的数据

尽管启用或禁用 FIFO，仍会读取从接收缓冲器(SCxBUF)接收的数据。

当接收 FIFO 禁用时，通过该次读取，缓冲器全满标志 SCxMOD2<RBFLL>被清除为"0"。在这种情况下，在读取接收缓冲器的数据前，在接收移位寄存器中能接收下一数据。8-位 UART 模式下要添加的奇偶校验位及在 9-位 UART 模式时最有效的位，将储存在 SCxCR<RB8>中。

当接收 FIFO 可用时，由于 8-位数据能储存在 FIFO 中，9-位 UART 模式禁用。8-位 UART 模式下，虽然奇偶校验位丢失，但是奇偶校验错误得到确定，并且结果被储存在 SCxCR<PERR>中。

9.11.3.5 唤醒功能

9-位 UART 模式下，通过将唤醒功能 SCxMOD0 <WU>设置为"1"，从机控制器就能在唤醒模式下运行。在这种情况下，只有当 SCxCR <RB8>设置为"1"时，才会产生 INTRXx。

9.11.3.6 溢出错误

当 FIFO 禁用时，发生溢出错误，在接收下一数据前，在没有完成数据读取的情况下设置溢出标志。当发生溢出错误时，虽然接收缓冲器和 SCxCR<RB8>的内容未丢失，但是接收移位寄存器的内容丢失。

启用 FIFO 时，发生溢出错误，在 FIFO 已满而将下一数据移入接收缓冲器前，通过不读取数据而设置溢出标志。在这种情况下，FIFO 的内容不会丢失。

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

注：当模式从 I/O 接口 SCLK 输出模式切换到其他模式时，读取 SCxCR，并清除溢出标志。

9.12 传输

9.12.1 传输计数器

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

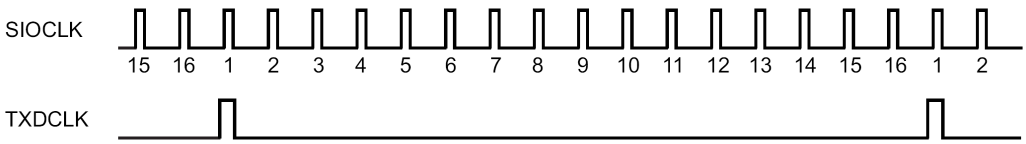


图 9-6 传输时钟的生成

9.12.2 传输控制

9.12.2.1 I/O 接口模式

在 SCLK 输出模式下且 SCxCR<IOC>设置为"0"时，发送缓冲器中的各位数据在从 SCLK 引脚输出的移位时钟的下降沿被输出到 TXD 引脚。

在 SCLK 输入模式下且 SCxCR<IOC> 设置为"1"时，发送缓冲器中的各位数据按照 SCxCR<SCLKS>的设置，在 SCLK 输入信号的上升沿或下降沿被输出到 TXD 引脚。

9.12.2.2 UART 模式

当发送数据写入发送缓冲器时，数据传输在下一 TXDCLK 的上升沿启动，并且也生成发送移位时钟信号。

9.12.3 发送操作

9.12.3.1 传输缓冲器的操作

若双缓冲禁用，CPU 将数据仅写入发送移位寄存器， 并在数据传输完成后产生发送中断 INTTXx。

若启用双缓冲(包括启用发送 FIFO 的情况)，写入发送缓冲器的数据移至发送移位寄存器。同时，产生 INTTXx 中断，发送缓冲器空标志(SCxMOD2<TBEMP>)设置为"1"。该标志表示能写入下一发送数据。当下一数据写入发送缓冲器时，<TBEMP>标志被清除为"0"。

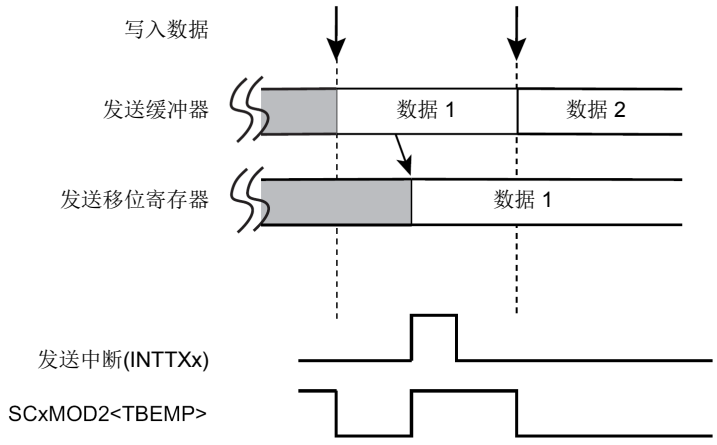


图 9-7 传输缓冲器的操作(启用双缓冲器)

9.12.3.2 发送 FIFO 的操作

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

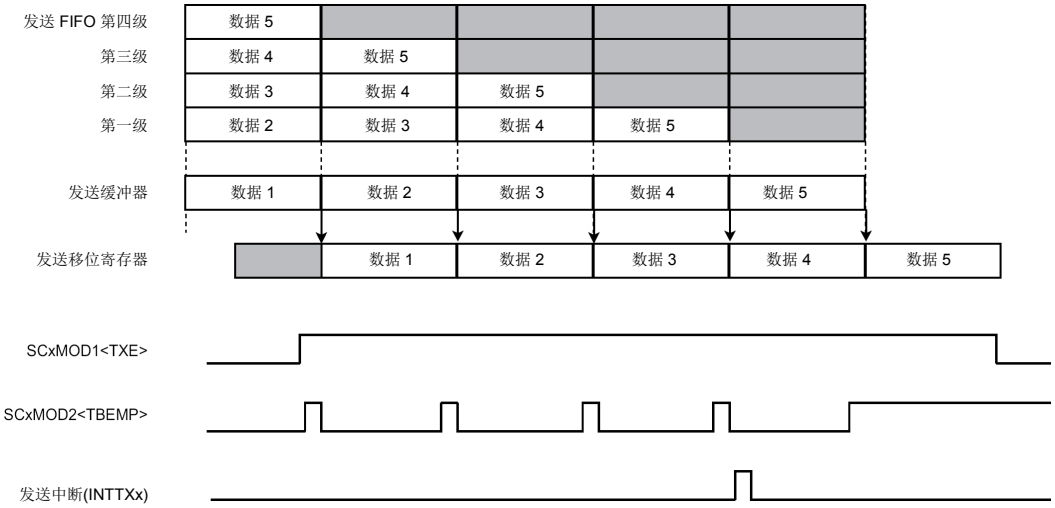
注：**使用 TX FIFO 缓冲器时，在设置 SIO 传输模式(半双工/全双工)并启用 FIFO (SCxFCNF<CNFG> = "1")后，必须清除 TX FIFO。**

通过将传输模式设置为半双工而发送 4-字节数据流的设置和操作如下所示。

SCxMOD1[6:5] = 10	: 传输模式设置为半双工。
SCxFCNF[4:0] = 11011	: 若 FIFO 为空，传输自动禁用。
	: 在接收 FIFO 中使用的字节数与中断生成充满率相同。
SCxTFC[1:0] = 00	: 将中断生成充满率 设置为"0"。
SCxTFC[7:6] = 11	: 清除接收 FIFO，并设置中断生成的条件。
SCxFCNF[0] = 1	:启用 FIFO。

在配置上述设置后，通过将 5-字节数据写入发送缓冲器和 FIFO，并将 SCxMOD1<TXE>位设置为"1"，便可启动数据传输。当最后的发送数据移到发送缓冲器时，就会生成发送 FIFO 中断。当最后的数据传输完成时，传输顺序终止。

一旦配置上述设置，若传输未设置为自动禁用，则应写入发送数据，使传输持续进行。



9.12.3.3 I/O 接口模式/SCLK 输出传输

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

SCLK 输出中止和恢复的时间随缓冲器和 FIFO 的使用而不同。

(1) 单缓冲器

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

(2) 双缓冲器

在发送移位寄存器和发送缓冲器的数据传输完成后，SCLK 输出停止。 当下一数据写入缓冲器时，SCLK 输出恢复。

(3) FIFO

发送移位寄存器，发送缓冲器和 FIFO 中储存的所有数据的传输完成，SCLK 输出停止。写入下一数据，SCLK 输出恢复。

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

9.12.3.4 欠载运行错误

若在 I/O 接口 SCLK 输入模式时，发送 FIFO 禁用，并且若在下一帧时钟输入前在发送缓冲器中未设置任何数据，其在发送移位寄存器的数据传输完成后发生，则会发生欠载运行错误，SCxCR<PERR>设置为"1"。

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

注：**在 I/O 接口 SCLK 输出模式切换到其他模式前，应读取 SCxCR 寄存器，并清除欠载运行标志。**

9.13 握手功能

握手功能是为了启用 CTS(清除发送)引脚逐帧数据传输，并防止溢出错误。该功能由 SCxMOD0<CTSE>启用或禁用。

当 $\overline{\text{CTS}}$ 引脚设置为"高"电平时，虽然能完成当前数据传输，但是下一数据传输中止，直到 $\overline{\text{CTS}}$ 引脚恢复"低"电平。然而在这种情况下，在正常时间生成 INTTXx 中断，下一发送数据写入发送缓冲器，并且它会等到准备发送数据时为止。

注 1：若 $\overline{\text{CTS}}$ 信号在传输时设置为"H"，则下一数据传输在当前传输完成后中止。(图 9-9 中的"a"点)

注 2：**在 $\overline{\text{CTS}}$ 设置为"L"后，数据传输在 TXDCLK 时钟的第一个下降沿开始。**(图 9-9 中的"b"点)

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

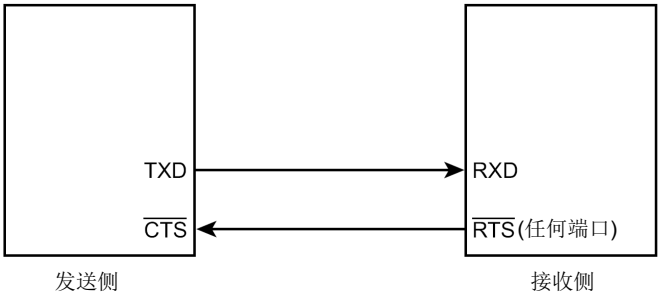


图 9-8 握手功能

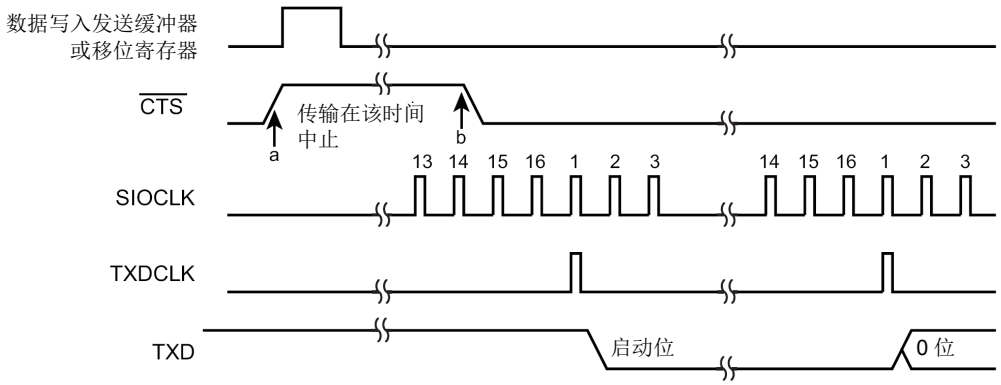


图 9-9 $\overline{\text{CTS}}$信号时序

9.14 中断/错误产生时间

9.14.1 RX 中断

接收操作数据流和读取路由如图 9-10 所示。

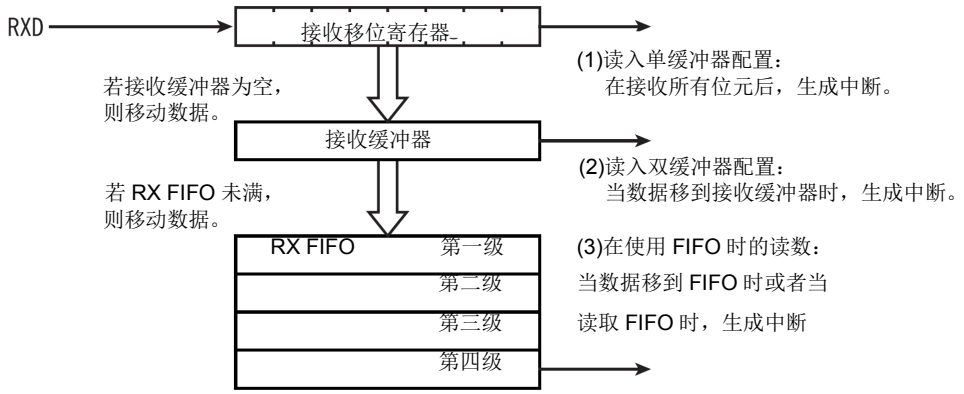


图 9-10 接收缓冲器/FIFO 配置图

9.14.1.1 单缓冲器/双缓冲器

如下表所示，在传输模式和缓冲器配置所确定的时间，生成 RX 中断。

缓冲器 配置	UART 模式	IO 接口模式
单缓冲器	-	恰在最后的 SCLK 的上升沿/下降沿后 (上升或下降按照 SCxCR<SCLKS>的设置确定。)
双缓冲器	在第一个停止位的中心周围	恰在最后的 SCLK 的上升沿/下降沿后 (上升或下降按照 SCxCR<SCLKS>的设置确定。) 通过读取缓冲器，数据从移位寄存器转移到缓冲器之后。

注：当发生溢出错误时，不生成中断。

9.14.1.2 FIFO

使用 FIFO 时，在操作和 SCxRFC<RFIS >设置得到设立的条件下,会生成接收中断。

- 一帧的所有位元的接收完成。
- 读取 FIFO

如表 9-11 所述，中断条件由 SCxRFC<RFIS>的设置决定。

表 9-11 在使用 FIFO 时接收中断条件

SCxRFC<RFIS>	中断条件
"0"	"FIFO 充满率"等于"FIFO 中断生成的充满率"。
"1"	"FIFO 充满率"大于等于"FIFO 中断生成的充满率"。

9.14.2 TX 中断

发送操作数据流和读取路由如图 9-11 所示。

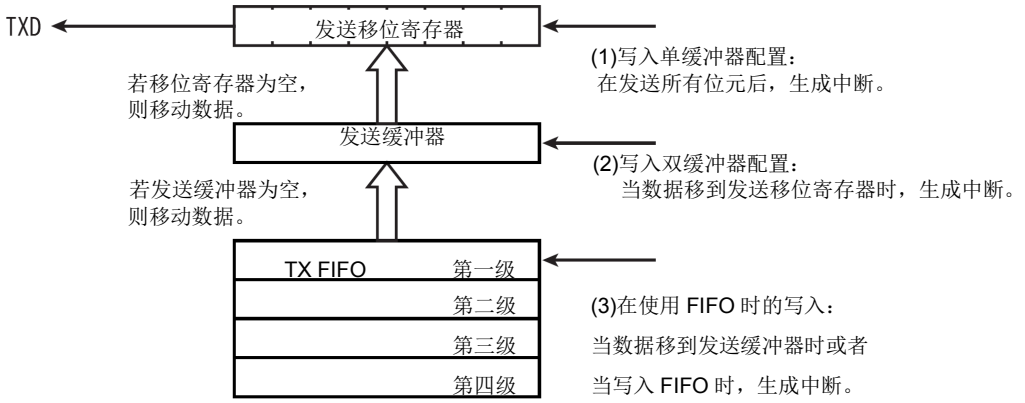


图 9-11 发送缓冲器/FIFO 配置图

9.14.2.1 单缓冲器/双缓冲器

如下表所示，在传输模式和缓冲器配置所确定的时间，生成 TX 中断。

缓冲器 配置	UART 模式	IO 接口模式
单缓冲器	恰在发送停止位前	恰在最后的 SCLK 的上升沿/下降沿后 (上升或下降按照 SCxCR<SCLKS> 的设置确定。)
双缓冲器	当数据从发送缓冲器移到发送移位寄存器时。	

注：若启用双缓冲器，当数据通过写入缓冲器而从缓冲器移到移位寄存器时，也生成中断。

9.14.2.2 FIFO

在使用 FIFO 时，在操作和 SCxTFC<TFIS>设置得到设立的条件下，会生成发送中断。

- 一帧的所有位元的传输完成。
- 写入 FIFO

如表 9-12 所述，中断条件由 SCxTFC<TFIS>的设置决定。

表 9-12 在使用 FIFO 时发送中断条件

SCxTFC<TFIS>	中断条件
"0"	"FIFO 充满率"等于"FIFO 中断生成的充满率"。
"1"	"FIFO 充满率"小于等于"FIFO 中断生成的充满率"。

9.14.3 错误产生

9.14.3.1 UART 模式

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

9.14.3.2 IO 接口模式

溢出错误	恰在最后的 SCLK 的上升沿/下降沿后 (上升或下降按照 SCxCR<SCLKS>的设置确定。)
欠载运行错误	恰在下一 SCLK 上升沿或下降沿后。 (上升或下降按照 SCxCR<SCLKS>的设置确定。)

注：在 SCLK 输出模式时，溢出错误和欠载运行错误无意义。

9.15 软件复位

将 SCxMOD2<SWRST[1:0]>写为"10" 然后"01"，便可产生软件复位。

结果，SCxMOD0<RXE>，SCxMOD1<TXE>，SCxMOD2<TBEMP><RBFLL><TXRUN>，SCxCR <OERR><PERR><FERR>被初始化。 接收电路，发送电路和 FIFO 变成初始状态。 其他状态保持不变。

9.16 各模式下的操作

9.16.1 模式 0 (I/O 接口模式)

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

下列操作说明是针对 FIFO 禁用的情况。FIFO 操作，详见先前描述接收/发送 FIFO 功能的章节。

9.16.1.1 发送数据

(1) SCLK 输出模式

- 若发送双缓冲器禁用(SCxMOD2<WBUF> = "0")

  每当 CPU 把数据写入发送缓冲器时，数据从 TXD 引脚输出，时钟从 SCLK 引脚输出。当所有数据被输出时，生成中断(INTTXx)。

- 若启用发送双缓冲器(SCxMOD2<WBUF> = "1")

  当 CPU 把数据写入发送缓冲器，而数据传输停止时，或者当发送缓冲器的数据传输完成时，数据从发送缓冲器移到发送移位寄存器。同时，发送缓冲器空标志 SCxMOD2<TBEMP>设置为"1"，生成 INTTXx 中断。

  当数据从发送缓冲器移到发送移位寄存器时，若发送缓冲器无任何数据要移到发送移位寄存器，则不会生成 INTTXx 中断，SCLK 输出停止。

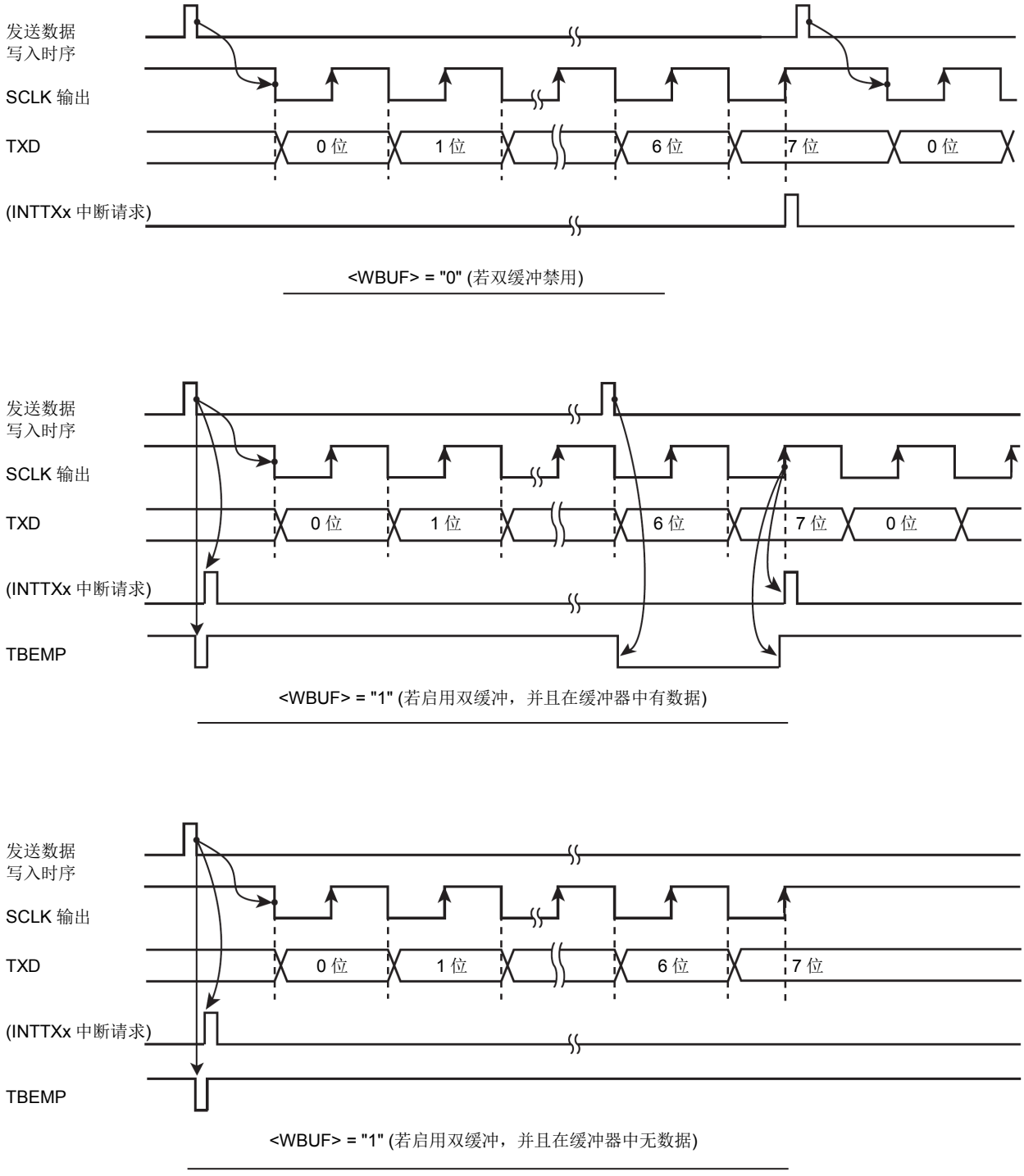


图 9-12　I/O 接口模式(SCLK 输出模式)下的发送操作

(2) SCLK 输入模式

• 若双缓冲禁用(SCxMOD2<WBUF> = "0")

若在数据写入发送缓冲器的条件下输入 SCLK，则 8 位数据从 TXD 引脚输出。当所有数据被输出时，生成中断 INTTXx。如图 9-13 所示，在时间点"A"之前，下一发送数据必须被写入。

• 启用双缓冲器 SCxMOD2<WBUF> = "1")

当 CPU 在 SCLK 输入被激活前把数据写入发送缓冲器时，或者当发送移位寄存器的数据传输完成时，数据从发送缓冲器移到发送移位寄存器。同时，发送缓冲器空标志 SCxMOD2<TBEMP>设置为"1"，生成 INTTXx 中断。

若 SCLK 输出被激活而在发送缓冲器中无数据，则虽然内部位元计数器被启动，但仍会发生欠载运行错误，并发送 8-位虚拟数据(0xFF)。

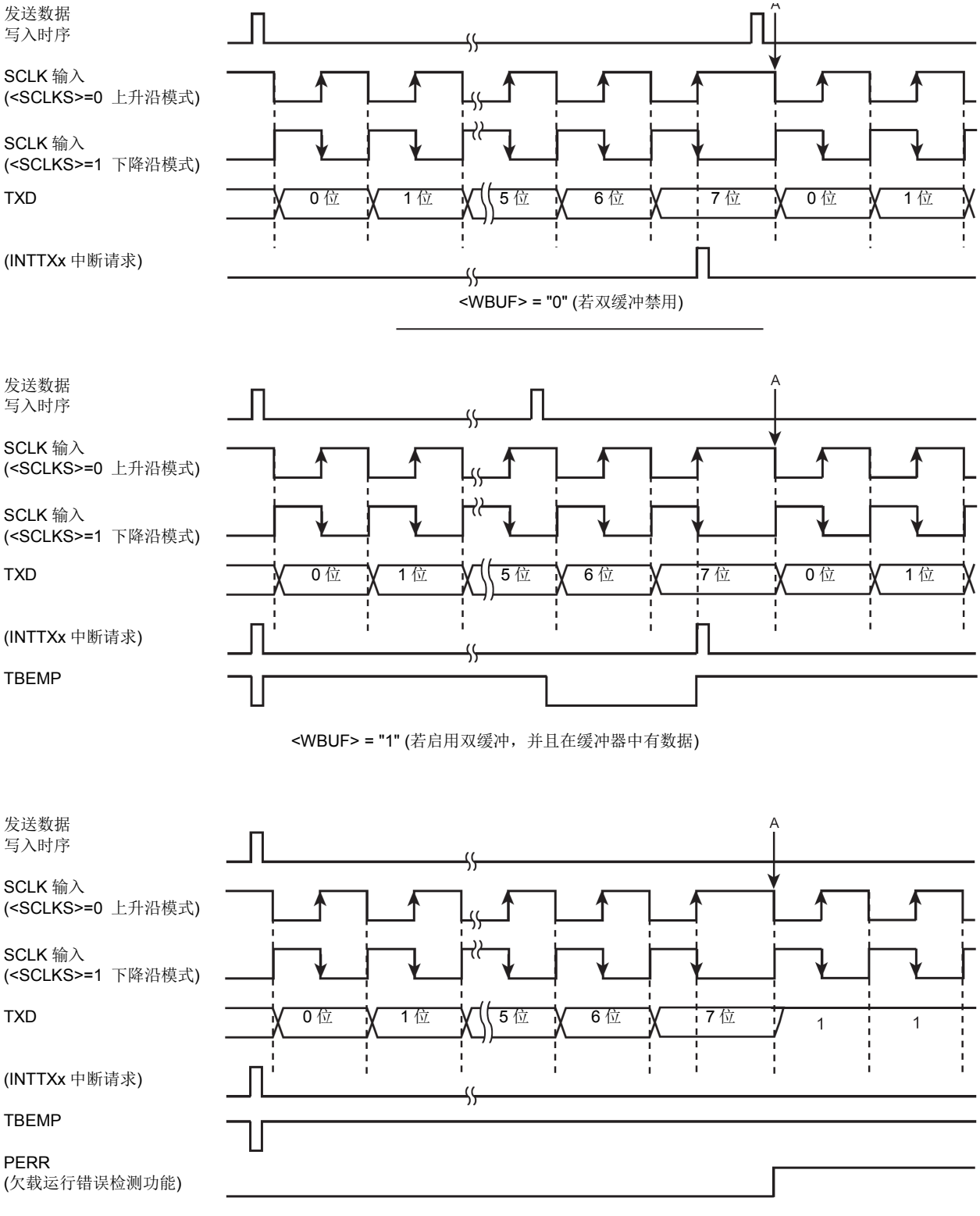


图 9-13 在 I/O 接口模式(SCLK 输入模式)时的发送操作

9.16.1.2 接收

(1) SCLK 输出模式

将接收启用位 SCxMOD0<RXE>设置为"1"，可启动 SCLK 输出。

- 若双缓冲器禁用(SCxMOD2<WBUF> = "0")

  每当 CPU 读取接收的数据时，时钟脉冲就会从 SCLK 引脚输出，下一数据被储存在移位寄存器中。当接收到所有 8-位时，生成 INTRXx 中断。

- 启用双缓冲器 SCxMOD2<WBUF> = "1")

  在移位寄存器中储存的数据移到接收缓冲器，并且接收缓冲器能接收下一帧。数据从移位寄存器移到接收缓冲器，接收缓冲器全满标志 SCxMOD2<RBFLL>设置为"1"，并且生成 INTRXx。

  当数据在接收缓冲器中时，若在完成下一 8 位的接收前无法从接收缓冲器读取数据，则不生成 INTRXx 中断，并且 SCLK 输出停止。在这种状态下，读取接收缓冲器的数据可使移位寄存器中的数据移到接收缓冲器，因此生成 INTRXx 中断，数据接收恢复。

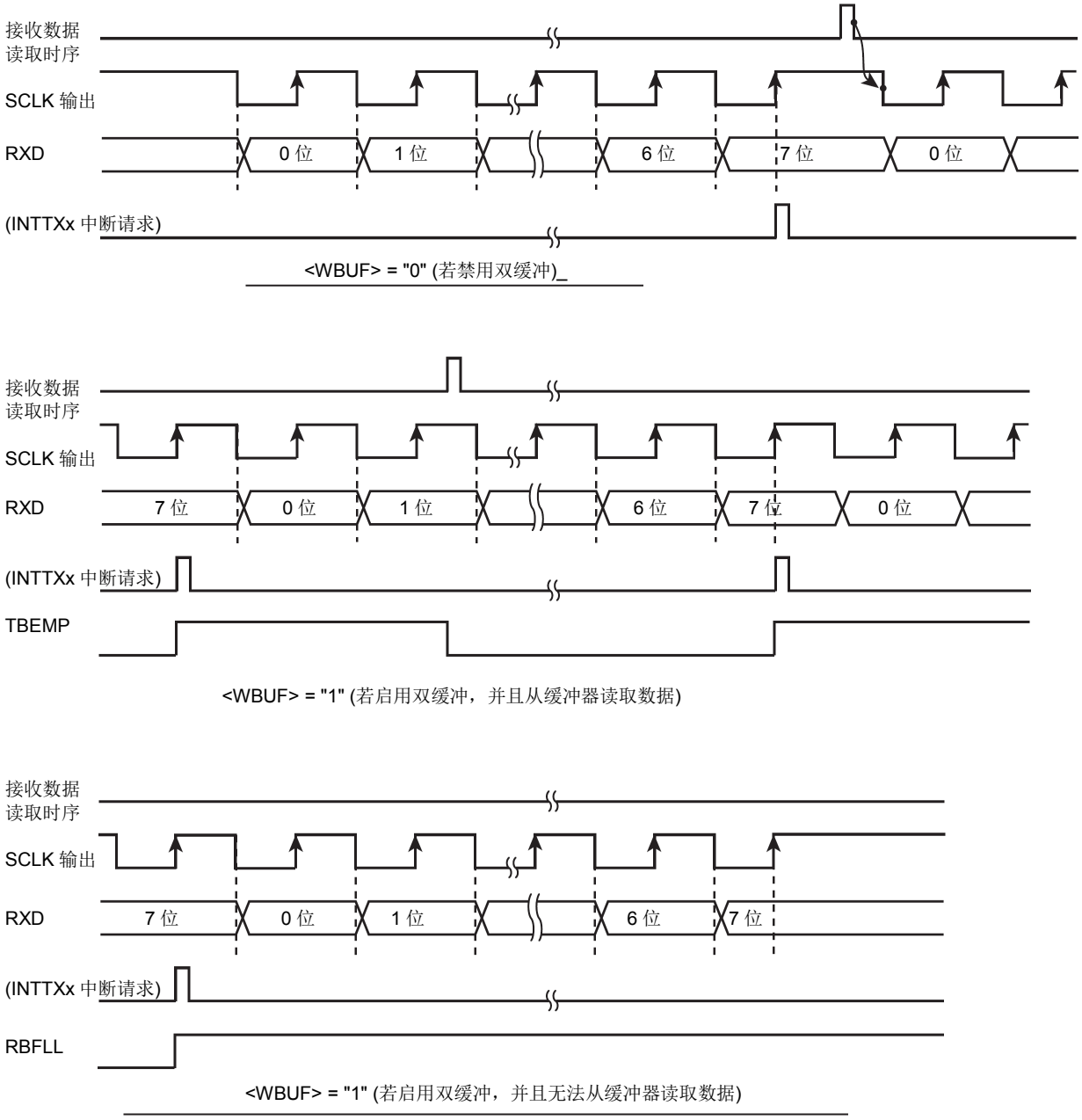


图 9-14　I/O 接口模式(SCLK 输出模式)下的接收操作

(2) SCLK 输入模式

在 SCLK 输入模式时，始终启用接收双缓冲，接收的帧能从移位寄存器移到接收缓冲器，并且接收缓冲器能相继接收下一帧。

每当接收的数据移到接收缓冲器时，生成 INTRXx 接收中断。

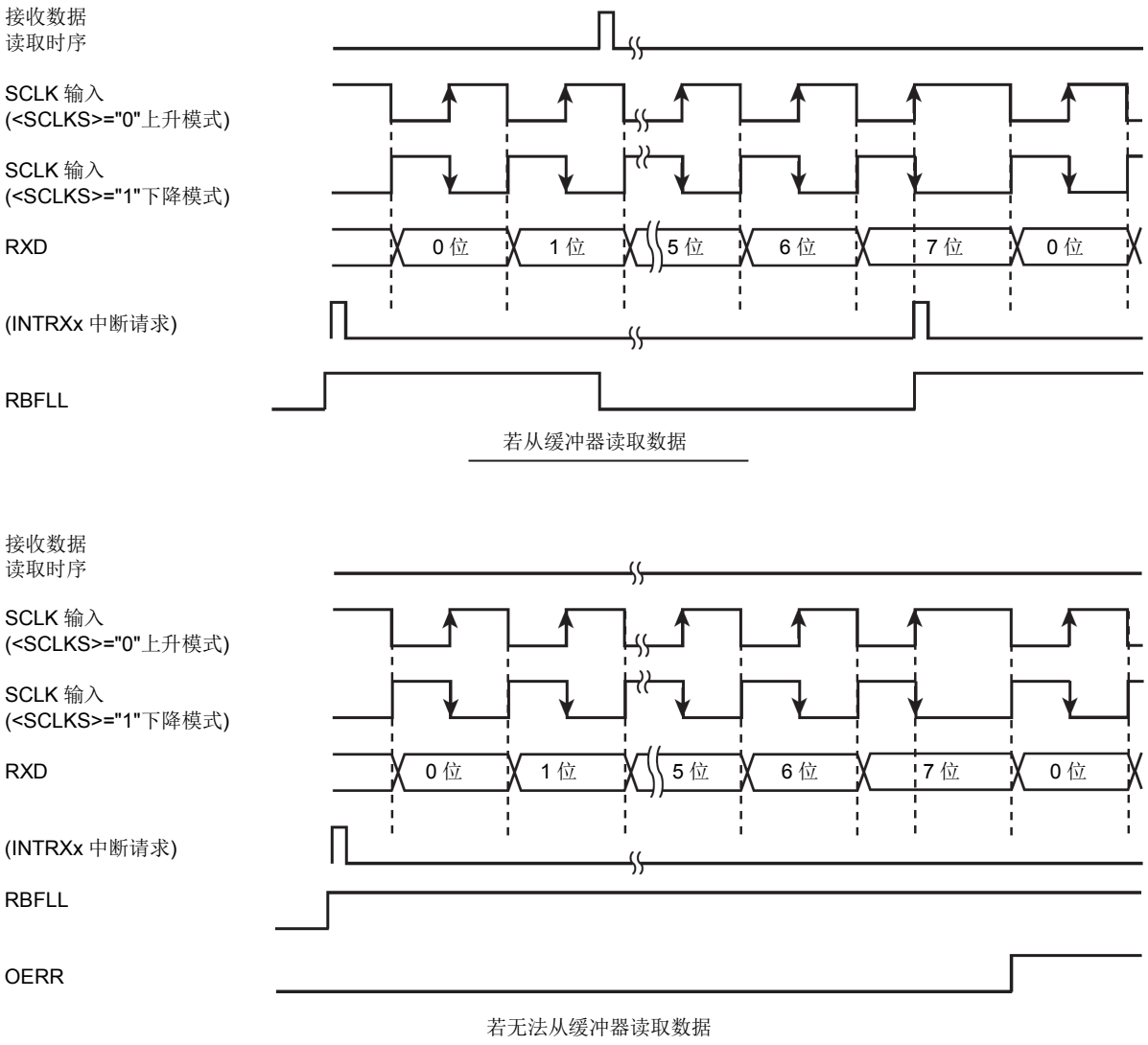


图 9-15 I/O 接口模式(SCLK 输入模式)下的接收操作

9.16.1.3 发送和接收(全双工)

(1) SCLK 输出模式

- 若 SCxMOD2<WBUF>设置为"0"，并且双缓冲器禁用

  SCLK 当 CPU 把数据写入发送缓冲器时，输出 SCLK。

  随后，8 位数据被转移到接收移位寄存器，并生成 INTRXx 接收中断。同时，写入发送缓冲器的 8 位数据从 TXD 引脚输出，当所有数据位的传输已完成时，生成 INTTXx 发送中断。然后，SCLK 输出停止。

  当从接收缓冲器读取数据并且下一发送数据由 CPU 写入发送缓冲器时，下一轮数据传输和接收开始。读取接收缓冲器和写入发送缓冲器的顺序可自由确定。只有在满足这两个条件时，数据传输才能恢复。

- 若 SCxMOD2<WBUF>设置为"1"，并且启用双缓冲器

  SCLK 当 CPU 把数据写入发送缓冲器时，输出 SCLK。

  8 位数据被转移到接收移位寄存器，移动到接收缓冲器，并生成 INTRXx 中断。当接收到 8 位数据时， 8 位发送数据从 TXD 引脚输出。当所有数据位已被发送时，生成 INTTXx 中断，下一数据从发送缓冲器移到发送移位寄存器。

  若发送缓冲器无数据要移到发送缓冲器时(SCxMOD2<TBEMP> = 1)或者当接收缓冲器全满时(SCxMOD2<RBFULL> = 1)，SCLK 输出停止。当满足这两个条件时即接收数据被读取和发送数据被写入，SCLK 输出恢复，下一轮数据传输和接收开始。

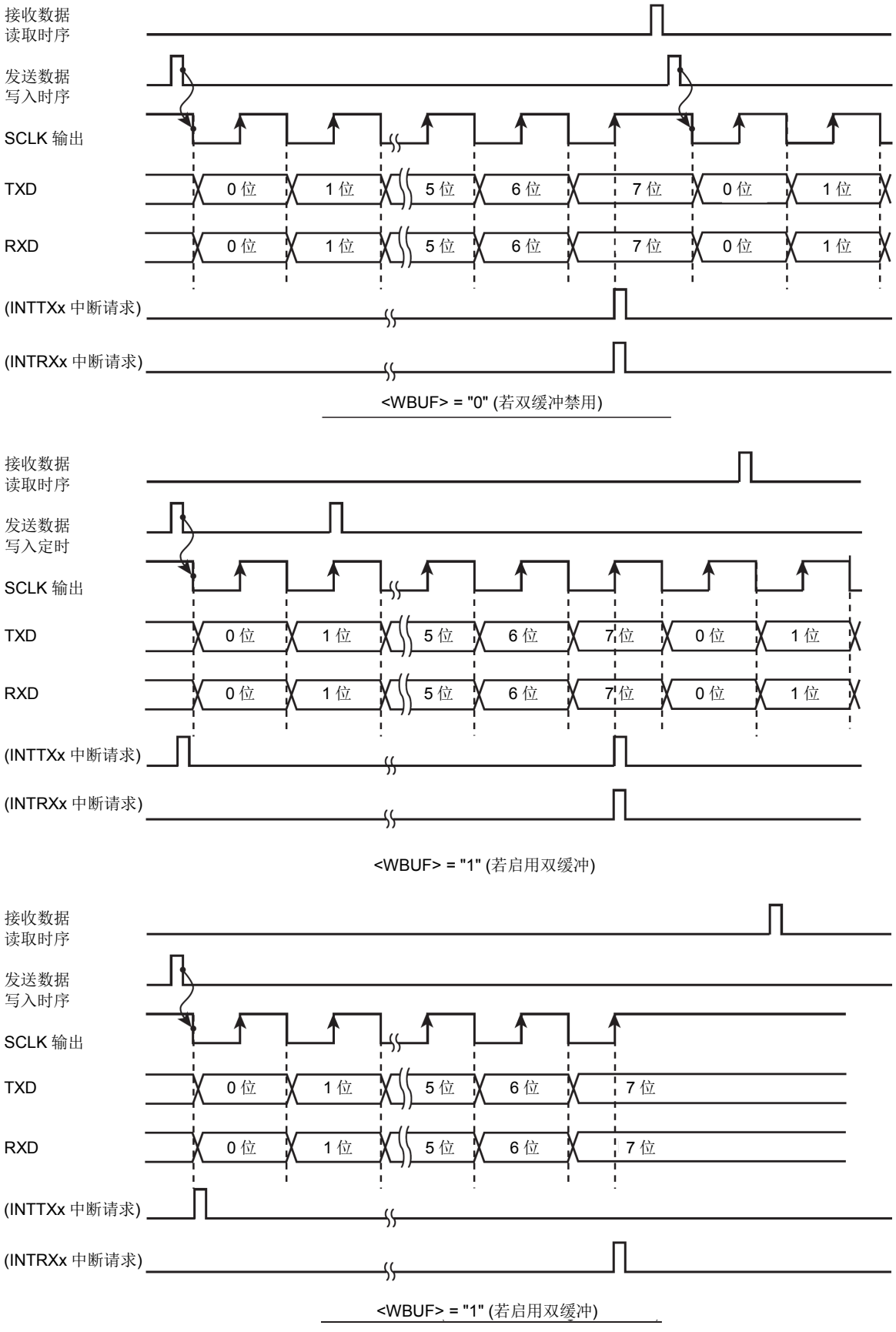


图 9-16 I/O 接口模式(SCLK 输出模式)下的发送/接收操作

(2) SCLK 输入模式

- 当 SCxMOD2<WBUF>设置为"0"时，并且发送双缓冲器禁用

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

  写入发送缓冲器的 8-位数据从 TXD 引脚输出，并且当 SCLK 输入被激活时，8 位数据被转移到接收缓冲器。在数据传输完成后，生成 INTTXx 中断。在数据接收完成后，当数据多接收移位寄存器移到接收缓冲器时，生成 INTRXx 中断。

  注意在下一帧的 SCLK 输入前，发送数据必须写入发送缓冲器(数据必须在图 9-17 中的 A 点前写入)。在下一帧数据的接收完成前，必须读取数据。

- 当 SCxMOD2<WBUF>设置为"1"时，并且启用双缓冲器。

  在发送移位寄存器的数据传输完成后，在发送缓冲器数据移到发送移位寄存器时，生成中断 INTTXx。同时，接收的数据被转移到移位寄存器，移动到接收缓冲器，并生成 INTRXx 中断。

  注意在下一帧的 SCLK 输入前，发送数据必须写入发送缓冲器(数据必须在图 9-17 中的 A 点前写入)。在下一帧数据的接收完成前，必须读取数据。

  在下一帧的 SCLK 输入后，发送移位寄存器(在该寄存器中，数据已从发送缓冲器中移动)的传输开始,同时接收数据被转移到接收移位寄存器。

  当收到帧的最后一位时，若接收缓冲器中的数据未被读取，则发生溢出错误。同样，当下一帧的 SCLK 被输入时，若无数据写入发送缓冲器，则发生欠载运行错误。

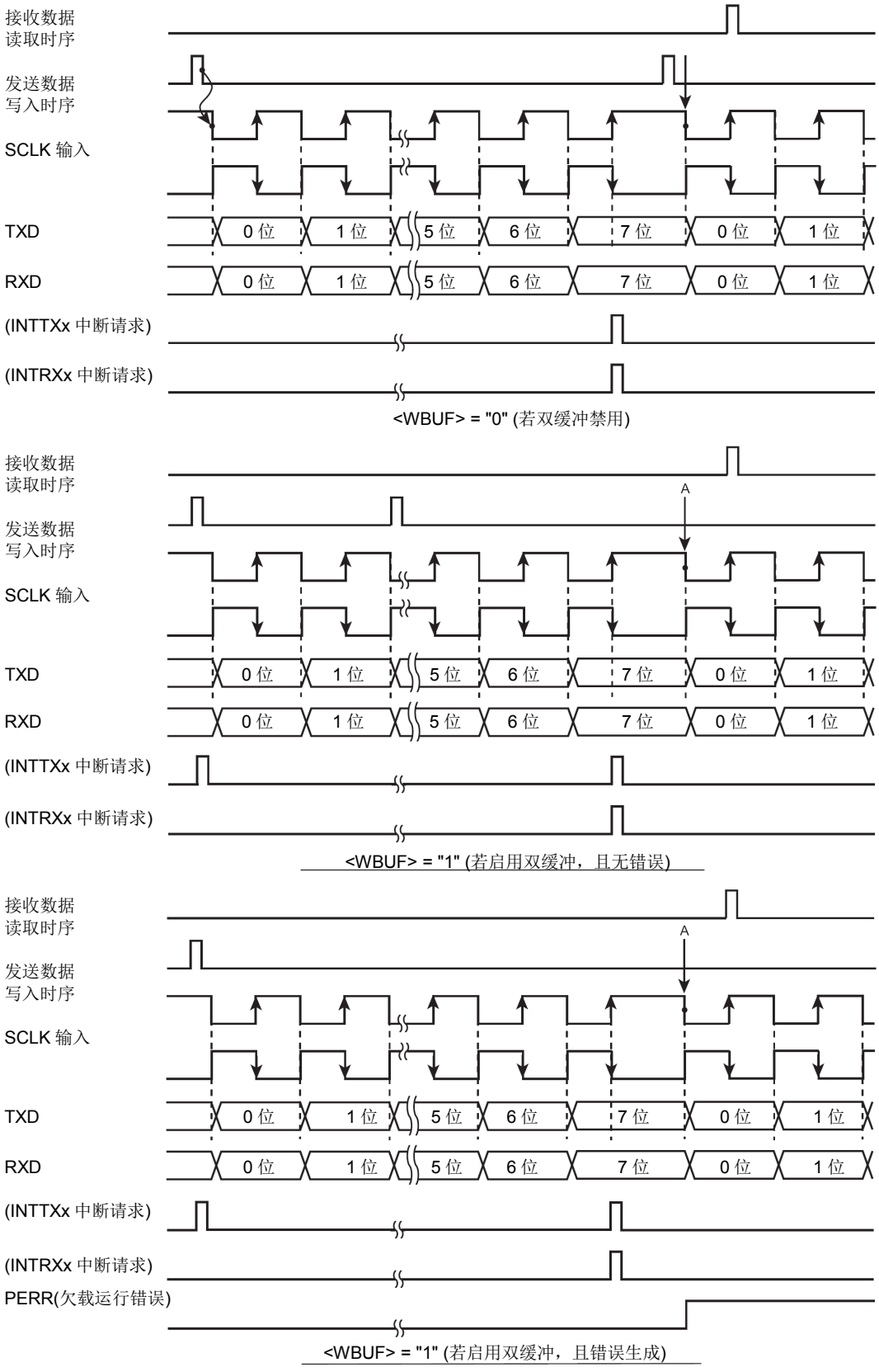


图 9-17 I/O 接口模式(SCLK 输入模式)下的发送/接收操作

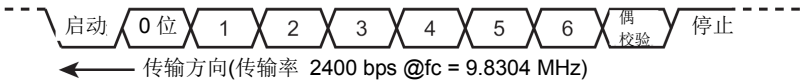
9.16.2 模式 1 (7-位 UART 模式)

将模式控制寄存器(SCxMOD<SM[1:0]>) 设置为"01"，可选择 7-位 UART 模式。

在该模式下，奇偶校验位可添加到发送数据流中；控制寄存器(SCxCR<PE>)控制奇偶校验启用/禁用设置。

当<PE>设置为"1" (启用)时，可以用 SCxCR<EVEN>位选择偶或奇校验。能用 SCxMOD2<SBLEN>规定停止位的长度。

以下列数据格式发送时，控制寄存器的设置如下表如示。



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

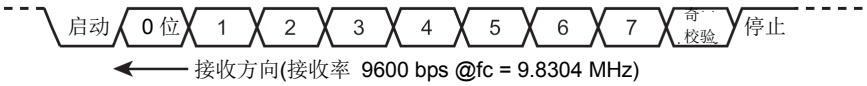
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	设置 7-位 UART 模式
SCxCR	←	x	1	1	x	x	x	0	0	启用偶校验
SCxBRCR	←	0	0	1	0	0	1	0	0	设置 2400 bps
SCxBUF	←	*	*	*	*	*	*	*	*	设置发送数据

X: 忽略 -: 无变化

9.16.3 模式 2 (8-位 UART 模式)

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

为了以下列格式接收数据，控制寄存器的设置如下：



时钟条件	系统时钟：	高速(fc)
	高速时钟齿轮：	× 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	启用奇校验
SCxBRCR	←	0	0	0	1	0	1	0	0	设置 9600 bps
SCxMOD0	←	-	-	1	-	-	-	-	-	启用接收

x: 忽略 -: 无变化

9.16.4 模式 3 (9-位 UART 模式)

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

发送数据时，最有效的位(第 9 位)写入 SCxMOD0<TB8>。数据储存在 SCxCR<RB8>中。

向/从缓冲器写入或读取数据时，最有效的位必须在向/从 SCxBUF 写入或读取前先写入或读取。

可用 SCxMOD2<SBLEN>规定停止位长度。

9.16.4.1 唤醒功能

在 9-位 UART 模式时，将唤醒功能控制位 SCxMOD0<WU>设置为"1"，从机控制器就能以唤醒模式运行。

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

注：从机控制器的 TXD 引脚必须用 PxOD 寄存器设置为开漏输出模式。

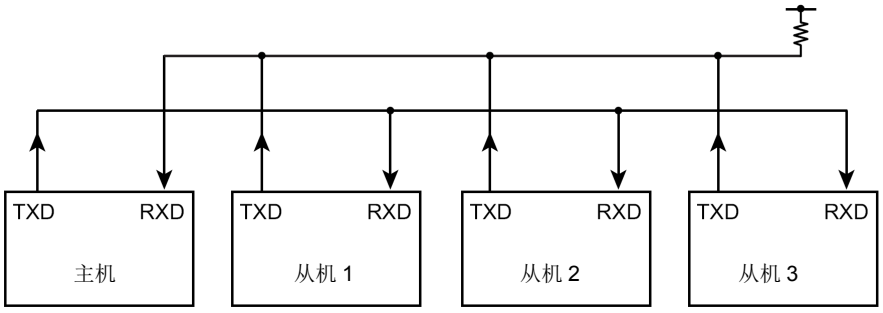


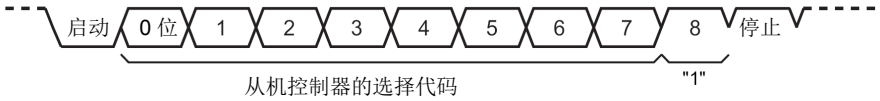
图 9-18 使用唤醒功能的串行链路

9.16.4.2 协议

1. 选择主从机控制器的 9-位 UART 模式。

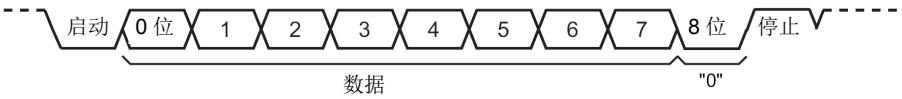
2. 关于从机控制器，将 SCxMOD0<WU>设置为"1"，使它们准备接收数据。

3. 主控制器发送包括从机控制器选择代码(8 位)在内的单帧数据。在这种情况下，最有效的位元(第 8 位) <TB8> 必须设置为"1"。



4. 各从机控制器接收上述数据帧；若接收的代码与控制器自己的选择代码匹配，则它将 WU 位清除为"0"。

5. 主控制器将数据发送到指定的从机控制器(其控制器 SCxMOD<WU>位被清除为"0")。在这种情况下，最有效的位(第 8 位) <TB8> 必须设置为"0"。



6. 因为最有效的位(第 8 位)<RB8>设置为"0"，所以<WU>位设置为"1"的从机控制器忽略接收数据，因此不会生成中断(INTRXx)。此外，<WU>位设置为"0"的从机控制器能将数据发送给主控制器，告知数据已成功接收。

10. 12-位模数转换器

TMPM370FYDFG/FYFG 包含两个 12-位逐次逼近模数转换器(ADCs)。

ADC 单元 A (ADC A)有 15 个模拟输入。其中 3 个输入能用于马达 0 的分流电阻器电流， 12 个输入能用于外部输入。

ADC 单元 B (ADC B)有 138 个模拟输入。其中 3 个输入能用于马达 0 的分流电阻电流， 1 个输入能用于马达 1 的分流电阻电流，13 个输入能用于外部输入。

外部模拟输入引脚(AINA0～AINA8，AINA9/AINB0，AINA10/AINB1，AINA11/AINB2，AINB3～AINB12)也能用作输入/输出端口。

10.1 功能和特征

1. 当接收 PMD 或 TMRB 的触发信号时(中断)，可选择模拟输入并启动 AD 转换。
2. 在软件触发器程序和恒定触发器程序中，可选择模拟输入。
3. ADCs 有 12 个 AD 转换结果寄存器。
4. 在 PMD 触发器和 TMRB 触发器启动的程序结束时，ADCs 生成中断信号。
5. 在软件触发器程序和恒定触发器程序结束时，ADCs 生成中断信号。
6. ADCs 具备 AD 转换监控功能。 在该功能被启用时，一旦转换结果可匹配指定的比较值，即发生一次中断。

10.2 方块图

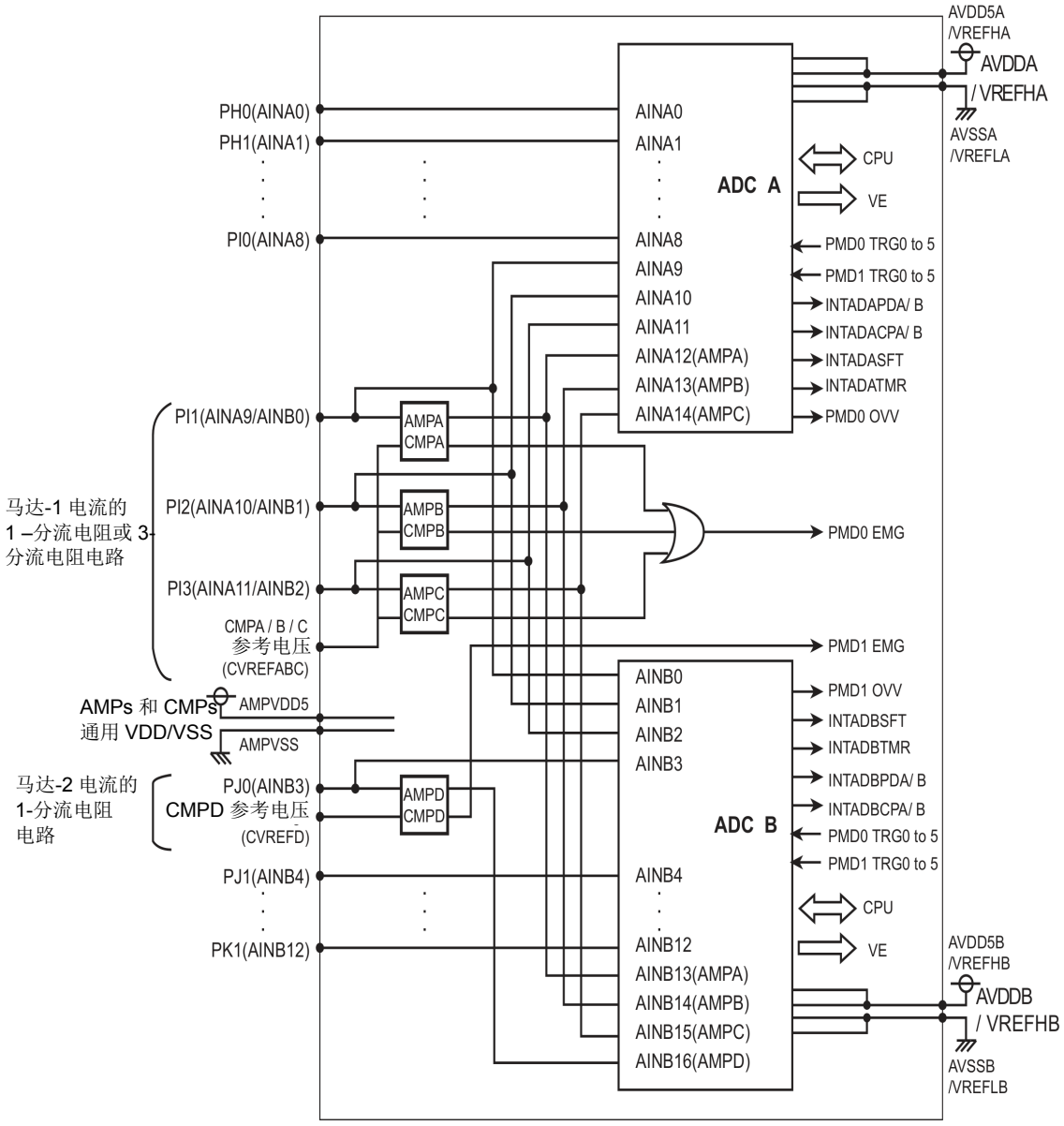


图 10-1 AD 转换器方块图

10.3 寄存器列表

单元 x	基址
单元 A	0x4003_0000
单元 B	0x4003_0200

寄存器名称 (x=A，B)		Address (Base+)
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
监测设置寄存器 0	ADxCMPCR0	0x0010
监测设置寄存器 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 触发器程序编号选择寄存器 0	ADxPSEL0	0x0050
PMD 触发器程序编号选择寄存器 1	ADxPSEL1	0x0054
PMD 触发器程序编号选择寄存器 2	ADxPSEL2	0x0058
PMD 触发器程序编号选择寄存器 3	ADxPSEL3	0x005C
PMD 触发器程序编号选择寄存器 4	ADxPSEL4	0x0060
PMD 触发器程序编号选择寄存器 5	ADxPSEL5	0x0064
PMD 触发器程序编号选择寄存器 6	ADxPSEL6	0x0068
PMD 触发器程序编号选择寄存器 7	ADxPSEL7	0x006C
PMD 触发器程序编号选择寄存器 8	ADxPSEL8	0x0070
PMD 触发器程序编号选择寄存器 9	ADxPSEL9	0x0074
PMD 触发器程序编号选择寄存器 10	ADxPSEL10	0x0078
PMD 触发器程序编号选择寄存器 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

寄存器名称 (x=A，B)		Address (Base+)
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
定时器触发器程序寄存器 0 ~ 3	ADxTSET03	0x00B0
定时器触发器程序寄存器 4 ~ 7	ADxTSET47	0x00B4
定时器触发器程序寄存器 8 ~ 11	ADxTSET811	0x00B8
软件触发器程序寄存器 0 ~ 3	ADxSSET03	0x00BC
软件触发器程序寄存器 4 ~ 7	ADxSSET47	0x00C0
软件触发器程序寄存器 8 ~ 11	ADxSSET811	0x00C4
恒定转换程序寄存器 0 ~ 3	ADxASET03	0x00C8
恒定转换程序寄存器 4 ~ 7	ADxASET47	0x00CC
恒定转换程序寄存器 8 ~ 11	ADxASET811	0x00D0
模式设置寄存器 3	ADxMOD3	0x00D4

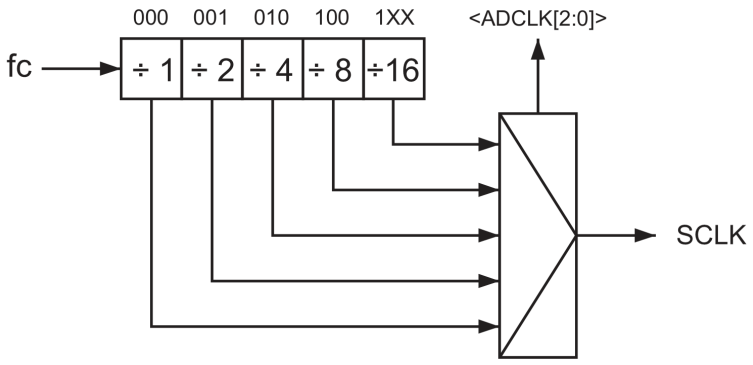
10.4 寄存器说明

以 ADC 时钟设置寄存器所选时钟频率进行 AD 转换。

10.4.1 ADxCLK (时钟设置寄存器)

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

位	比特符号	型号	功能
31-7	−	R	读作 "0"。
6-3	TSH[3: 0]	R/W	写为"1001"。
2-0	ADCLK[2: 0]	R/W	AD 预分频器输出(SCLK)选择 000：fc (注 1) 001：fc/2 010：fc/4 011：fc/8 1xx：fc/16



注 1：使用的 SCLK 频率最多 40MHz。当 fc > 40 MHz 时，不得将<ADCLK[2:0]>设置为"000"。

注 2：以该寄存器所选时钟频率进行 AD 转换。为了保证达到所担保的准确性，必须选择转换时钟频率。

注 3：当正进行 AD 转换时，不得变更转换时钟。

10.4.2 ADxMOD0 (模式设置寄存器 0)

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

位	比特符号	型号	功能
31-2	−	R	读作 "0"。
1	DACON	R/W	DAC 控制 0：OFF 1：ON 当使用 ADC 时，将<DACON>设置为"1"。
0	ADSS	W	软件触发的转换 0：忽略 1：启动 将<ADSS>设置为"1"，就能启动 AD 转换(软件触发的转换)。接收 PMD 或 TMRB 的触发信号(中断)也能启动 AD 转换。 关于详细设置，请阅读关于 PMD 和 TMRB 的章节。

10.4.3 ADxMOD1 (模式设定寄存器 1)

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

位	比特符号	型号	功能
31-8	−	R	读作 "0"。
7	ADEN	R/W	AD 转换控制 0：禁用 1：启用 使用 ADC 时，允许将<ADEN>设置为"1"。将<ADEN>设置为"1"后，设置<ADAS>"1"开始 AD 转换并且重复转换。
6-1	−	R	读作 "0"。
0	ADAS	R/W	常数 AD 转换控制 0：禁用 1：启用

10.4.4 ADxMOD2 (模式设定寄存器 2)

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

位	比特符号	型号	功能
31-2	−	R	读作 "0"。
1	ADSFN	R	软件转换忙碌标志 0：转换完成 1：转换正在进行 <ADSFN>为软件 AD 转换忙碌标志。将<ADSS>设置为"1"后，AD 转换实际开始时，将<ADSFN>设置为"1"。完成 AD 转换时，<ADSFN>将清零。
0	ADBFN	R	AD 转换忙碌标志 0：没有进行转换 1：转换正在进行 <ADBFN>为软件 AD 转换忙碌标志。AD 转换启用时，不论对于哪个转换因子(PMD，定时器，软件，常数)，<ADBFN>将设置为"1"。完成 AD 转换时，<ADBFN>将清零。

10.4.5 ADxMOD3(模式设定寄存器 3)

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

位	比特符号	型号	功能
31-16	−	R	读作 "0"。
15-11	−	R/W	写作 "0"。
10	−	R/W	写作 "1"。
9	−	R/W	写作 "0"。
8	−	R/W	写作 "0"。
7	−	R/W	写作 "0"。
6	−	R/W	写作 "1"。
5-3	PMODE[2:0]	R/W	写为"100"。
2-0	−	R/W	写作 "0"。

注：ADxMOD3 < PMODE [2: 0]> 必须设置为"100"。而且不得改变 ADxMOD3 寄存器中的其他位元。

10.4.6 ADxCMPCR0(监视设定寄存器 0)

在确定转换结果后，中断信号(INTADxCPn)将生成。

(n=A, B / A: 监视器 0/ B:监视器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	CMPCNT0			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMP0EN	-	-	ADBIG0	REGS0			
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-12	-	R	读作 "0"。
11-8	CMPCNT0[3:0]	R/W	确定结果的比较计数 0：每一项比较之后 1：两项比较之后 • • 15: 16 项比较之后
7	CMP0EN	R/W	监视功能 0：禁用 1：启用
6-5	-	R	读作 "0"。
4	ADBIG0	R/W	比较条件 0：大于或等于比较寄存器 1：小于或等于比较寄存器
3-0	REGS0[3:0]	R/W	有待比较的 AD 转换结果寄存器 0000: ADxREG0　0100: ADxREG4　1000: ADxREG8 0001: ADxREG1　0101: ADxREG5　1001: ADxREG9 0010: ADxREG2　0110: ADxREG6　1010: ADxREG10 0011: ADxREG3　0111: ADxREG7　1011: ADxREG11

注： 寄存器 ADxCMPCR0 和 ADxCMPCR1 主要用于启用或禁用 AD 转换结果与规定比较值之间的比较，选择与 AD 转换结果进行比较的寄存器，同时设定必须进行多少次比较才能确定最终结果。

10.4.7 ADxCMPCR1(监视设定寄存器 1)

在确定转换结果后，中断信号(INTADxCPn)将生成。

(n=A, B / A: 监视器 0/ B:监视器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	CMPCNT1			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	CMP1EN	-	-	ADBIG1	REGS1			
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-12	-	R	读作 "0"。
11-8	CMPCNT1[3:0]	R/W	确定结果的比较计数 0：每一项比较之后 1：两项比较之后 • • 15：16 项比较之后
7	CMP1EN	R/W	监视功能 0：禁用 1：启用
6-5	-	R	读作 "0"。
4	ADBIG1	R/W	比较条件 0：大于或等于比较寄存器 1：小于或等于比较寄存器
3-0	REGS1[3:0]	R/W	有待比较的 AD 转换结果寄存器 0000: ADxREG0 0100: ADxREG4 1000: ADxREG8 0001: ADxREG1 0101: ADxREG5 1001: ADxREG9 0010: ADxREG2 0110: ADxREG6 1010: ADxREG10 0011: ADxREG3 0111: ADxREG7 1011: ADxREG11

注： 寄存器 ADxCMPCR0 和 ADxCMPCR1 主要用于启用或禁用 AD 转换结果与规定比较值之间的比较，选择与 AD 转换结果进行比较的寄存器，同时设定必须进行多少次比较才能确定最终结果。

10.4.8 ADxCMP0 (转换结果比较寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	AD0CMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	AD0CMP0				-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	AD0CMP0[11:0]	R/W	与 AD 转换结果进行比较的数值。 规定与 AD 转换结果进行比较的数值。
3-0	-	R	读作 "0"。

10.4.9 ADxCMP1 (转换结果比较寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	AD0CMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	AD0CMP1				-	-	-	-
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	AD0CMP1[11:0]	R/W	与 AD 转换结果进行比较的数值。 规定与 AD 转换结果进行比较的数值。
3-0	-	R	读作 "0"。

10.4.10 ADxREG0 (转换结果比较寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR0				-	-	OVR0	ADR0RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR0[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR0	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG0 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG0 低位字节时清除。
0	ADR0RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR0RF>是一种在 ADxREG0 寄存器中存储 AD 转换结果时设定的标志，该标志在读取 ADxREG0 低位字节时清除。

10.4.11 ADxREG1 (转换结果寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR1				-	-	OVR1	ADR1RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR1[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR1	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG1 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG1 低位字节时清除。
0	ADR1RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR1RF>是一种在 ADxREG1 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG1 低位字节时清除。

10.4.12 ADxREG2 (转换结果寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR2				-	-	OVR2	ADR2RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR2[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR2	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG2 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG2 低位字节时清除。
0	ADR2RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR2RF>是一种在 ADxREG2 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG2 低位字节时清除。

10.4.13 ADxREG3 (转换结果寄存器 3)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR3				-	-	OVR3	ADR3RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR3[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR3	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG3 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG3 低位字节时清除。
0	ADR3RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR3RF>是一种在 ADxREG3 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG3 低位字节时清除。

10.4.14 ADxREG4 (转换结果比较寄存器 4)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR4							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR4				-	-	OVR4	ADR4RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR4[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR4	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG4 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG4 低位字节时清除。
0	ADR4RF	R	AD 转换结果保存标志 0：无转换结果保存 1：转换结果已保存 <ADR4RF>是一种在 ADxREG4 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG4 低位字节时清除。

10.4.15 ADxREG5 (转换结果比较寄存器 5)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR5							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR5				-	-	OVR5	ADR5RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR5[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR5	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG5 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG5 低位字节时清除。
0	ADR5RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR5RF>是一种在 ADxREG5 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG5 低位字节时清除。

10.4.16 ADxREG6 (转换结果比较寄存器 6)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR6							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR6				-	-	OVR6	ADR6RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR6[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR6	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG6 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG6 低位字节时清除。
0	ADR6RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR6RF>是一种在 ADxREG6 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG6 低位字节时清除。

10.4.17 ADxREG7 (转换结果比较寄存器 7)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR7							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR7				-	-	OVR7	ADR7RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR7[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR7	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG7 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG7 低位字节时清除。
0	ADR7RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR7RF>是一种在ADxREG7寄存器中存储AD转换结果时设定的标志，该标志将在读取ADxREG7低位字节时清除。

10.4.18 ADxREG8 (转换结果比较寄存器 8)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR8							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR8				-	-	OVR8	ADR8RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR8[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR8	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG8 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG8 低位字节时清除。
0	ADR8RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR8RF>是一种在 ADxREG8 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG8 低位字节时清除。

10.4.19 ADxREG9 (转换结果比较寄存器 9)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR9							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR9				-	-	OVR9	ADR9RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR9[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR9	R	溢出标志 0： 未出现溢出现象 1： 出现溢出现象 该标志在读取 ADxREG9 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG9 低位字节时清除。
0	ADR9RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR9RF>是一种在 ADxREG9 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG9 低位字节时清除。

10.4.20 ADxREG10 (转换结果比较寄存器 10)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR10							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR10				-	-	OVR10	ADR10RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR10[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR10	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG10 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG10 低位字节时清除。
0	ADR10RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR10RF>是一种在 ADxREG10 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG10 低位字节时清除。

10.4.21 ADxREG11 (转换结果比较寄存器 11)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ADR11							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADR11				-	-	OVR11	ADR11RF
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 "0"。
15-4	ADR11[11:0]	R	AD 转换结果值
3-2	-	R	读作 "0"。
1	OVR11	R	溢出标志 0：未出现溢出现象 1：出现溢出现象 该标志在读取 ADxREG11 数值之前，存储新的 AD 转换结果时设定，并且在读取 ADxREG11 低位字节时清除。
0	ADR11RF	R	AD 转换结果保存标志 0：无结果保存 1：结果已保存 <ADR11RF>是一种在 ADxREG11 寄存器中存储 AD 转换结果时设定的标志，该标志将在读取 ADxREG11 低位字节时清除。

10.4.22 PMD 触发程序寄存器

可通过 PMD (可编程电动机驱动器) 触发器开始 AD 转换。

PMD 触发程序寄存器主要用于规定通过 PMD 生成的十二个触发器中的每一个启用的程序，选择在该程序结束后生成的中断信号，同时选择即将使用的 AIN 输入信息。

PMD 触发程序寄存器分为三种类型：

(x=A, B：ADC 单元)

- PMD 触发程序编号选择寄存器(ADxPSEL0 ~ ADxPSEL11)

  PMD 触发程序编号选择寄存器(ADxPSELn)旨在规定与 PMD 生成的十二个触发器(PMD0TRG0 ~ 5, PMD1TRG0 ~ 5)对应的十二个 AD 转换启用信号启用的程序。程序 0 ~ 5 可用。

  "ADxPSEL0 ~ ADxPSEL5"与"PMD0TRG0 ~ 5"。"ADxPSEL6 ~ ADxPSEL11"与"PMD1TRG0 ~ 5"对应。

- PMD 触发器中断选择寄存器(ADxPINTS0 ~ ADxPINTS5)

  PMD 触发器中断选择寄存器(ADxPINTS0 ~ ADxPINTS5)选择每一个程序完成后生成的中断，并启用或禁用该中断。

  ADxPINTS0 与程序 0 对应，并且一直存续至 ADxPINT5 (程序 5)。

- PMD 触发程序寄存器(ADxPSET0 ~ ADxPSET5)

  PMD 触发程序设置寄存器(ADxPSET0 ~ ADxPSET5)旨在规定程序 0 ~ 5 中每一个程序的设置。每一个 PMD 触发程序寄存器都由四个旨在规定待转换的 AIN 输入的寄存器组成。与 ADxPSETn0 ~ ADxPSETn3 寄存器对应的转换结果将存入转换结果寄存器 0 ~ 3(ADxREG0 ~ ADxREG3)。

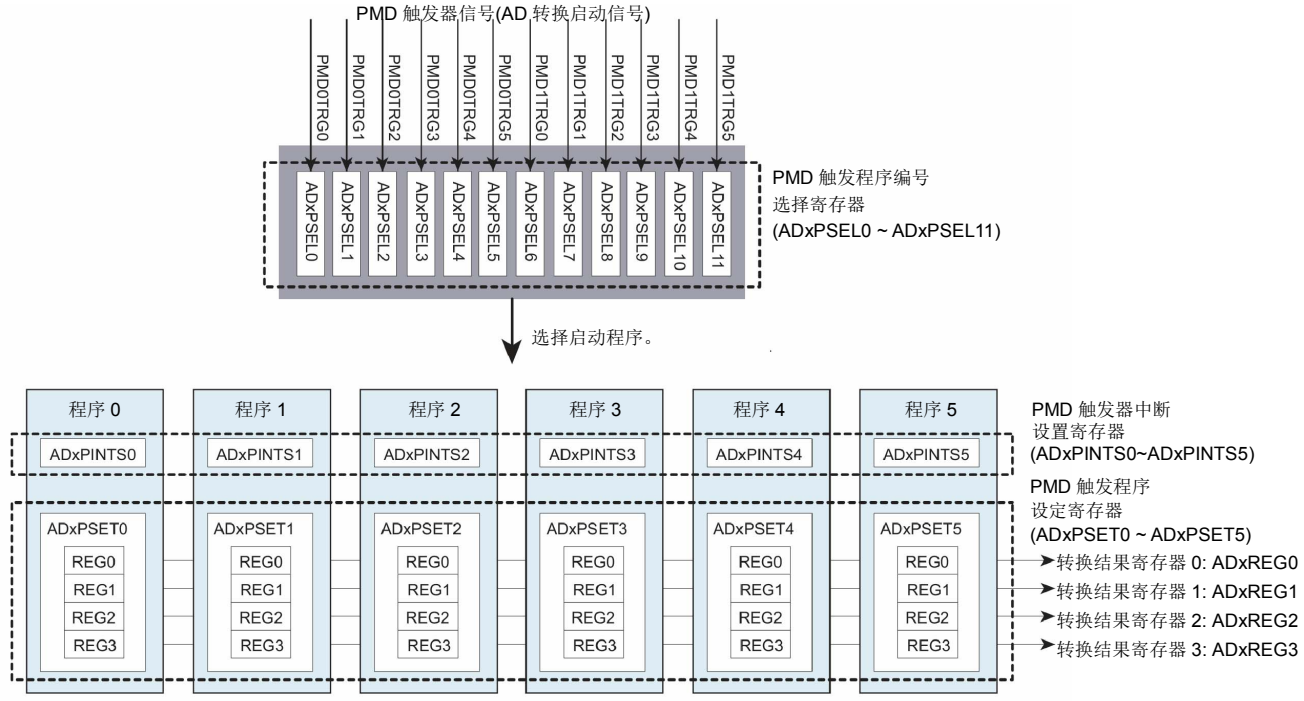


图 10-2 PMD 触发程序寄存器

10.4.22.1 ADxPSEL0 ~ ADxPSEL11 (PMD 触发程序编号选择寄存器 0 ~ 11)

ADxPSEL0:PMD 触发程序编号选择寄存器 0

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS0	-	-	-	-	PMDS0		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS0	R/W	PMD0TRG0 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS0[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL1:PMD 触发程序编号选择寄存器 1

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS1	-	-	-	-	PMDS1		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS1	R/W	PMD0TRG1 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS1[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL2:PMD 触发程序编号选择寄存器 2

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS2	-	-	-	-	PMDS2		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS2	R/W	PMD0TRG2 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS2[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL3:PMD 触发程序编号选择寄存器 3

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS3	-	-	-	-	PMDS3		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS3	R/W	PMD0TRG3 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS3[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL4:PMD 触发程序编号选择寄存器 4

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS4	-	-	-	-	PMDS4		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS4	R/W	PMD0TRG4 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS4[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL5:PMD 触发程序编号选择寄存器 5

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS5	-	-	-	-	PMDS5		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS5	R/W	PMD0TRG5 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS5[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL6:PMD 触发程序编号选择寄存器 6

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS6	-	-	-	-	PMDS6		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS6	R/W	PMD1TRG0 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS6[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL7:PMD 触发程序编号选择寄存器 7

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS7	-	-	-	-	PMDS7		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS7	R/W	PMD1TRG1 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS7[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL8:PMD 触发器程序编号选择寄存器 8

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS8	-	-	-	-	PMDS8		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS8	R/W	PMD1TRG2 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS8[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL9:PMD 触发程序编号选择寄存器 9

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS9	-	-	-	-	PMDS9		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS9	R/W	PMD1TRG3 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS9[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL10:PMD 触发程序编号选择寄存器 10

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS10	-	-	-	-	PMDS10		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS10	R/W	PMD1TRG4 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS10[2:0]	R/W	程序编号选择(参见表 10-1)

ADxPSEL11:PMD 触发程序编号选择寄存器 11

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	PENS11	-	-	-	-	PMDS11		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-8	-	R	读作 "0"。
7	PENS11	R/W	PMD1TRG5 触发控制 0：禁用 1：启用
6-3	-	R	读作 "0"。
2-0	PMDS11[2:0]	R/W	程序编号选择(参见表 10-1)

表 10-1 程序编号选择

<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

10.4.22.2 ADxPINTS0 ~ 5 (PMD 触发中断选择寄存器 0 ~ 5)

ADxPINTS0:PMD 触发中断选择寄存器 0

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	INTSEL0	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	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 程序 0 选择启动中断。

ADxPINTS1:PMD 触发中断选择寄存器 1

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	INTSEL1	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	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 程序 1 选择启动中断。

ADxPINTS2:PMD 触发中断选择寄存器 2

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	INTSEL2	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	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 程序 2 选择启动中断。

ADxPINTS3:PMD 触发中断选择寄存器 3

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	INTSEL3	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	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 程序 3 选择启动中断。

ADxPINTS4:PMD 触发中断选择寄存器 4

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	INTSEL4	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	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 程序 4 选择启动中断。

ADxPINTS5:PMD 触发中断选择寄存器 5

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	INTSEL5	
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	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 程序 5 选择启动中断。

10.4.22.3 ADxPSET0 ~ 5 (PMD 触发程序寄存器 0 ~ 5)

每一个 ADxPSETn (n=0 ~ 5: 程序编号)由假设<AINSPnm [4:0]>, <UVWISnm [1:0]>, 与<ENSPnm>为一对的四个集合组成。

(m=0 ~ 3)(x=A, B:ADC 单元)

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>

将<ENSPnm>设置为"1"启用 ADxPSETnm 寄存器。用<UVWISnm [1:0]>位元选择相位-U, 相位-V 或相位-W。用<AINSPnm[4:0]>位元选择准备使用的 AIN 引脚。

表 10-2 选择 AIN 引脚

<AINSP00 [4:0]> ~ <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	: AINA10
0_1011	: AINA11	: AINB11
0_1100	: AINA12	: AINB12
0_1101	: AINA13	: AINB13
0_1110	: AINA14	: AINB14
0_1111	: 保留	: AINB15
1_0000	: 保留	: AINB16
0_1101 ~ 1_1111	: 保留	: 保留

ADxPSET0:PMD 触发程序寄存器 0

	31	30	29	28	27	26	25	24
比特符号	ENSP03	UVWIS03		AINSP03				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP02	UVWIS02		AINSP02				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP01	UVWIS01		AINSP01				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP00	UVWIS00		AINSP00				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSP03	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	UVWIS03[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
28-24	AINSP03[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"。
23	ENSP02	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	UVWIS02[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
20-16	AINSP02[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
15	ENSP01	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	UVWIS01[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
12-8	AINSP01[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
7	ENSP00	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	UVWIS00[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
4-0	AINSP00[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"

相位选择

00	未指定
01	U
10	V
11	W

ADxPSET1:PMD 触发程序寄存器 1

	31	30	29	28	27	26	25	24
比特符号	ENSP13	UVWIS13		AINSP13				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP12	UVWIS12		AINSP12				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP11	UVWIS11		AINSP11				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP10	UVWIS10		AINSP10				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSP13	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	UVWIS13[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
28-24	AINSP13[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
23	ENSP12	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	UVWIS12[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
20-16	AINSP12[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
15	ENSP11	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	UVWIS11[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
12-8	AINSP11[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
7	ENSP10	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	UVWIS10[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
4-0	AINSP10[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"

相位选择

00	未指定
01	U
10	V
11	W

ADxPSET2:PMD 触发程序寄存器 2

	31	30	29	28	27	26	25	24
比特符号	ENSP23	UVWIS23		AINSP23				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP22	UVWIS22		AINSP22				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP21	UVWIS21		AINSP21				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP20	UVWIS20		AINSP20				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSP23	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	UVWIS23[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
28-24	AINSP23[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
23	ENSP22	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	UVWIS22[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
20-16	AINSP22[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
15	ENSP21	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	UVWIS21[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
12-8	AINSP21[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
7	ENSP20	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	UVWIS20[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
4-0	AINSP20[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"

相位选择

00	未指定
01	U
10	V
11	W

ADxPSET3:PMD 触发程序寄存器 3

	31	30	29	28	27	26	25	24
比特符号	ENSP33	UVWIS33		AINSP33				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP32	UVWIS32		AINSP32				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP31	UVWIS31		AINSP31				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP30	UVWIS30		AINSP30				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSP33	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	UVWIS33[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
28-24	AINSP33[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
23	ENSP32	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	UVWIS32[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
20-16	AINSP32[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
15	ENSP31	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	UVWIS31[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
12-8	AINSP31[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
7	ENSP30	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	UVWIS30[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
4-0	AINSP30[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"

相位选择

00	未指定
01	U
10	V
11	W

ADxPSET4:PMD 触发程序寄存器 4

	31	30	29	28	27	26	25	24
比特符号	ENSP43	UVWIS43		AINSP43				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP42	UVWIS42		AINSP42				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP41	UVWIS41		AINSP41				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP40	UVWIS40		AINSP40				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSP43	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	UVWIS43[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
28-24	AINSP43[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
23	ENSP42	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	UVWIS42[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
20-16	AINSP42[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
15	ENSP41	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	UVWIS41[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
12-8	AINSP41[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
7	ENSP40	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	UVWIS40[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
4-0	AINSP40[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"

相位选择

00	未指定
01	U
10	V
11	W

ADxPSET5:PMD 触发程序寄存器 5

	31	30	29	28	27	26	25	24
比特符号	ENSP53	UVWIS53		AINSP53				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSP52	UVWIS52		AINSP52				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSP51	UVWIS51		AINSP51				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSP50	UVWIS50		AINSP50				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSP53	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	UVWIS53[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
28-24	AINSP53[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
23	ENSP52	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	UVWIS52[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
20-16	AINSP52[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
15	ENSP51	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	UVWIS51[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
12-8	AINSP51[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"
7	ENSP50	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	UVWIS50[1:0]	R/W	相位选择(用于矢量引擎) 见下表。
4-0	AINSP50[4:0]	R/W	AIN 选择 参见"表 10-2 选择 AIN 引脚"

相位选择

00	未指定
01	U
10	V
11	W

10.4.23 ADxTSET03 / ADxTSET47 / ADxTSET811 (定时器触发程序寄存器)

AD 转换可通过作为触发器的定时器 5 (TMRB5)生成的 INTTB51 启动。目前，共有十二个 8-位寄存器用于定时器触发器编程。将<ENSTm>设置为"1"启用 ADxTSETm 寄存器。<AINSTm [4:0]>用于选择准备使用的 AIN 引脚。定时器触发程序寄存器的编号与 AD 转换结果寄存器编号对应。完成本次 AD 转换时，中断: INTAD x TMR 生成。

(m=0 ~ 11), (x=A, B : ADC 单元)

表 10-3 选择 AIN 引脚

<AINST0 [4:0]> ~ <AINST11 [4:0]>	ADC A 单元	ADC B 单元
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	: AINA10
0_1011	: AINA11	: AINB11
0_1100	: AINA12	: AINB12
0_1101	: AINA13	: AINB13
0_1110	: AINA14	: AINB14
0_1111	: 保留	: AINB15
1_0000	: 保留	: AINB16
0_1101 ~ 1_1111	: 保留	: 保留

ADxTSET03:定时器触发程序寄存器 03

	31	30	29	28	27	26	25	24
比特符号	ENST3	-	-	AINST3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENST2	-	-	AINST2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENST1	-	-	AINST1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENST0	-	-	AINST0				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENST3	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINST3[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
23	ENST2	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINST2[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
15	ENST1	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINST1[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
7	ENST0	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINST0[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"

ADxTSET47:定时器触发程序寄存器 47

	31	30	29	28	27	26	25	24
比特符号	ENST7	-	-	AINST7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENST6	-	-	AINST6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENST5	-	-	AINST5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENST4	-	-	AINST4				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENST7	R/W	ADxREG7 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINST7[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
23	ENST6	R/W	ADxREG6 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINST6[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
15	ENST5	R/W	ADxREG5 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINST5[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
7	ENST4	R/W	ADxREG4 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINST4[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"

ADxTSET811:定时器触发程序寄存器 811

	31	30	29	28	27	26	25	24
比特符号	ENST11	-	-	AINST11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENST10	-	-	AINST10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENST9	-	-	AINST9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENST8	-	-	AINST8				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENST11	R/W	ADxREG11 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINST11[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
23	ENST10	R/W	ADxREG10 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINST10[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
15	ENST9	R/W	ADxREG9 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINST9[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"
7	ENST8	R/W	ADxREG8 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINST8[4:0]	R/W	AIN 选择 参见"表 10-3 选择 AIN 引脚"

10.4.24 ADxSSET03/ADxSSET47/ADxSSET811 (软件触发程序寄存器)

AD 可由软件启动。目前，共有十二个 8-位寄存器用于软件触发器编程。将<ENSSm>设置为"1"启用 ADxSSETm 寄存器。The <AINSSm[4:0]> 用于选择准备使用的 AIN 引脚。软件触发程序寄存器的编号与转换结果寄存器编号对应。完成本次 AD 转换时，中断: NTADxSFT 生成。
(m=0 ~ 11), (x=A, B : ADC 单元)

表 10-4 选择 AIN 引脚

<AINSS0 [4: 0]> ~ <AINSS11 [4: 0]>	ADC A 单元	ADC B 单元
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	: AINA10
0_1011	: AINA11	: AINB11
0_1100	: AINA12	: AINB12
0_1101	: AINA13	: AINB13
0_1110	: AINA14	: AINB14
0_1111	: 保留	: AINB15
1_0000	: 保留	: AINB16
0_1101 ~ 1_1111	: 保留	: 保留

ADxSSET03:软件触发程序寄存器 03

	31	30	29	28	27	26	25	24
比特符号	ENSS3	-	-	AINSS3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSS2	-	-	AINSS2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSS1	-	-	AINSS1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSS0	-	-	AINSS0				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSS3	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINSS3[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"。
23	ENSS2	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINSS2[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
15	ENSS1	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINSS1[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
7	ENSS0	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINSS0[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"

ADxSSET47:软件触发程序寄存器 47

	31	30	29	28	27	26	25	24
比特符号	ENSS7	-	-	AINSS7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSS6	-	-	AINSS6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSS5	-	-	AINSS5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSS4	-	-	AINSS4				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSS7	R/W	ADxREG7 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINSS7[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
23	ENSS6	R/W	ADxREG6 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINSS6[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
15	ENSS5	R/W	ADxREG5 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINSS5[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
7	ENSS4	R/W	ADxREG4 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINSS4[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"

ADxSSET811:软件触发程序寄存器 811

	31	30	29	28	27	26	25	24
比特符号	ENSS11	-	-	AINSS11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSS10	-	-	AINSS10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSS9	-	-	AINSS9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSS8	-	-	AINSS8				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSS11	R/W	ADxREG11 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINSS11[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
23	ENSS10	R/W	ADxREG10 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINSS10[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
15	ENSS9	R/W	ADxREG9 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINSS9[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"
7	ENSS8	R/W	ADxREG8 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINSS8[4:0]	R/W	AIN 选择 参见"表 10-4 选择 AIN 引脚"

10.4.25 ADxASET03 / ADxASET47 / ADxASET811 (恒定转换程序寄存器)

ADC 可使转换触发器不断启用。目前，共有十二个 8-位寄存器用于恒定触发器编程。将 <ENSAm>设置为"1"启用 ADxASETm 寄存器。<AINSAm[4:0]> 用于选择准备使用的 AIN 引脚。恒定触发程序寄存器的编号与转换结果寄存器编号对应。

(m=0 ~ 11), (x=A, B : ADC 单元)

表 10-5 选择 AIN 引脚

<AINSA0 [4:0]> ~ <AINSA11 [4:0]>	ADC A 单元	ADC B 单元
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	: AINA10
0_1011	: AINA11	: AINB11
0_1100	: AINA12	: AINB12
0_1101	: AINA13	: AINB13
0_1110	: AINA14	: AINB14
0_1111	: 保留	: AINB15
1_0000	: 保留	: AINB16
0_1101 ~ 1_1111	: 保留	: 保留

ADxASET03:恒定转换程序寄存器 03

	31	30	29	28	27	26	25	24
比特符号	ENSA3	-	-	AINSA3				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSA2	-	-	AINSA2				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSA1	-	-	AINSA1				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSA0	-	-	AINSA0				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSA3	R/W	ADxREG3 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINSA3[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
23	ENSA2	R/W	ADxREG2 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINSA2[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
15	ENSA1	R/W	ADxREG1 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINSA1[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
7	ENSA0	R/W	ADxREG0 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINSA0[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"

ADxASET47:恒定转换程序寄存器 47

	31	30	29	28	27	26	25	24
比特符号	ENSA7	-	-	AINSA7				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSA6	-	-	AINSA6				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSA5	-	-	AINSA5				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSA4	-	-	AINSA4				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSA7	R/W	ADxREG7 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINSA7[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
23	ENSA6	R/W	ADxREG6 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINSA6[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
15	ENSA5	R/W	ADxREG5 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINSA5[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
7	ENSA4	R/W	ADxREG4 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINSA4[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"

ADxASET811:恒定转换程序寄存器 811

	31	30	29	28	27	26	25	24
比特符号	ENSA11	-	-	AINSA11				
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	ENSA10	-	-	AINSA10				
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	ENSA9	-	-	AINSA9				
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ENSA8	-	-	AINSA8				
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31	ENSA11	R/W	ADxREG11 启用 0：禁用 1：启用
30-29	-	R	读作 "0"。
28-24	AINSA11[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
23	ENSA10	R/W	ADxREG10 启用 0：禁用 1：启用
22-21	-	R	读作 "0"。
20-16	AINSA10[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
15	ENSA9	R/W	ADxREG9 启用 0：禁用 1：启用
14-13	-	R	读作 "0"。
12-8	AINSA9[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"
7	ENSA8	R/W	ADxREG8 启用 0：禁用 1：启用
6-5	-	R	读作 "0"。
4-0	AINSA8[4:0]	R/W	AIN 选择 参见"表 10-5 选择 AIN 引脚"

10.5 操作说明

10.5.1 模拟参考电压

对于高电平和低电平模拟参考电压，VREFHA 引脚和 VREFLA 引脚用于 ADC A；VREFHB 引脚与 VREFLB 引脚用于 ADC B。目前有两个寄存器控制 VREFHA 与 VREFLA(或 VREFHB 与 VREFLB)之间的电流。对这些引脚的输出值固定不变。

内部放大器与比较器共用电源线和地线，其分别与 AMPVDD5 和 AMPVSS 连接。

注 1：AD 转换期间，不得变更 H/I/J/K 端口输出数据，从而避免对转换结果构成影响。

注 2：AD 转换结果可能会因下列条件的存在而不稳定：

执行输入操作指令。

执行输出操作指令。

端口输出电流在不断变化。

通过求取多个输出结果平均值等应对措施获得精确数值。

10.5.2 启动 AD 转换

通过软件或下列三种触发器信号中的一种启动 AD 转换：

- PMD 触发器 (参见"10.4.22 PMD 触发程序寄存器")
- 定时器触发器 (TMRB5) (参见"10.4.23 定时器触发程序寄存器")
- 软件触发器(参见"10.4.24 软件触发程序寄存器")

这些启动触发器将被赋予下图所示优先级。

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 触发出现时正在进行 AD 转换，则应对该 PMD 触发现象进行处理，同时停止正在执行的程序，然后启动与该 PMD 触发器编号对应的 AD 转换。

如果高优先级触发出现时正在进行 AD 转换，则应在正在执行的程序完成后立即对该高优先级触发现象进行处理。

从触发器的生成到 AD 转换启动存在一定时延。该时延长短取决于触发器。 该时延如下述时序图表所示：

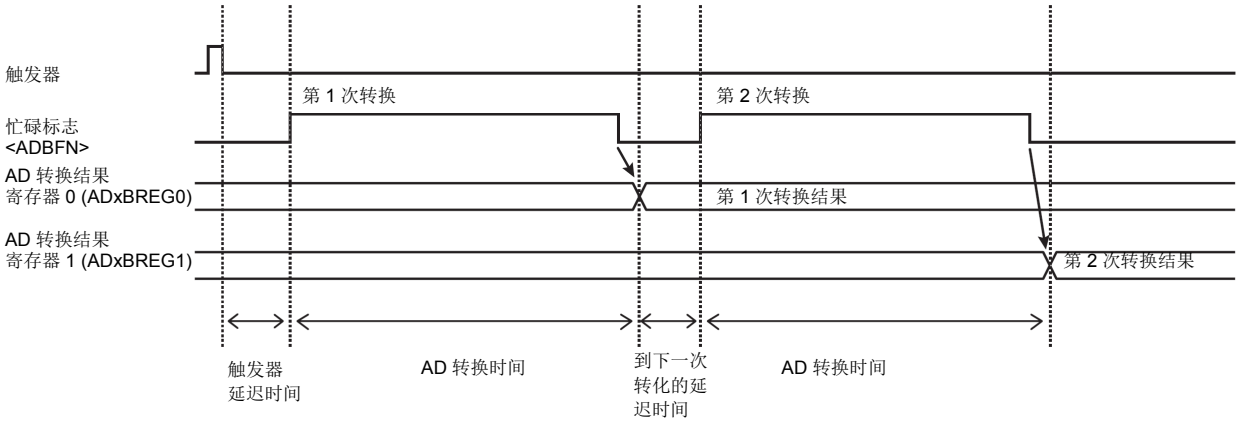


图 10-3 AD 转换时序图

表 10-6 AD 转换时间(SCLK = 40MHz)

		fsys = 80MHz		fsys = 40MHz	
	触发器	MIN	MAX	MIN	MAX
自触发器的延迟时间 [μs] (注 1)	PMD	0.125	0.163	0.225	0.3
	TMRB	0.125	0.263	0.225	0.5
	软件，恒定	0.138	0.275	0.25	0.525
AD 转换时间[μs]	-	1.85		1.85	
到下一次转换的延迟时间[μs] (注 2)	PMD	0.1	0.125	0.175	0.225
	TMRB，软件，恒定	0.1	0.238	0.175	0.425

注 1：从触发到 AD 转换开始时的延迟时间。

注 2：到第 2 次的延迟时间；或在一次触发多重转换的情况下，转换后的延迟时间。

10.5.3 AD 转换监控功能

ADC 具备 AD 转换监控功能。在该功能被启用时，一旦转换结果可匹配指定的比较值，即发生一次中断。

将 ADxCMPCR0<CMP0EN>或 ADxCMPCR1<CMP1EN>设置为"1"，即可启用该监控功能。在该监控功能运行期间，如果已被赋予该监控功能的 AD 转换结果寄存器的值符合 ADxCMCR0<ADBIG0>/ ADxCMPR1<ADBIG1>所规定的比较条件，则发生中断 (INTADxCPA 适用于 ADxCMPCR0，INTADxCPB 适用于 ADxCMPCR1)。可在转换结果存入寄存器的时序点执行该比较。

注 1：该比较功能不会清除 AD 转换结果存储标志(<ADRxRF>)。

注 2：该比较功能不同于通过软件读取转换结果。因此，如果在下一次转换完成时未读取前一次的结果，则会设置溢出标志 (<OVRx>)。

10.6 AD 转换的时序图

以下给出了软件触发转换，恒定转换与触发验收的时序图。

10.6.1 软件触发转换

在进行软件触发转换时，在经 ADxSSET03，ADxSSET47 与 ADxSSET811 编程的转换完成后，即可发生中断。(图 10-4)

如果 ADxMOD1<ADEN> 在 AD 转换期间被清”0”，则当前正在进行中的转换会停止，且不会存入该结果寄存器。(图 10-5)

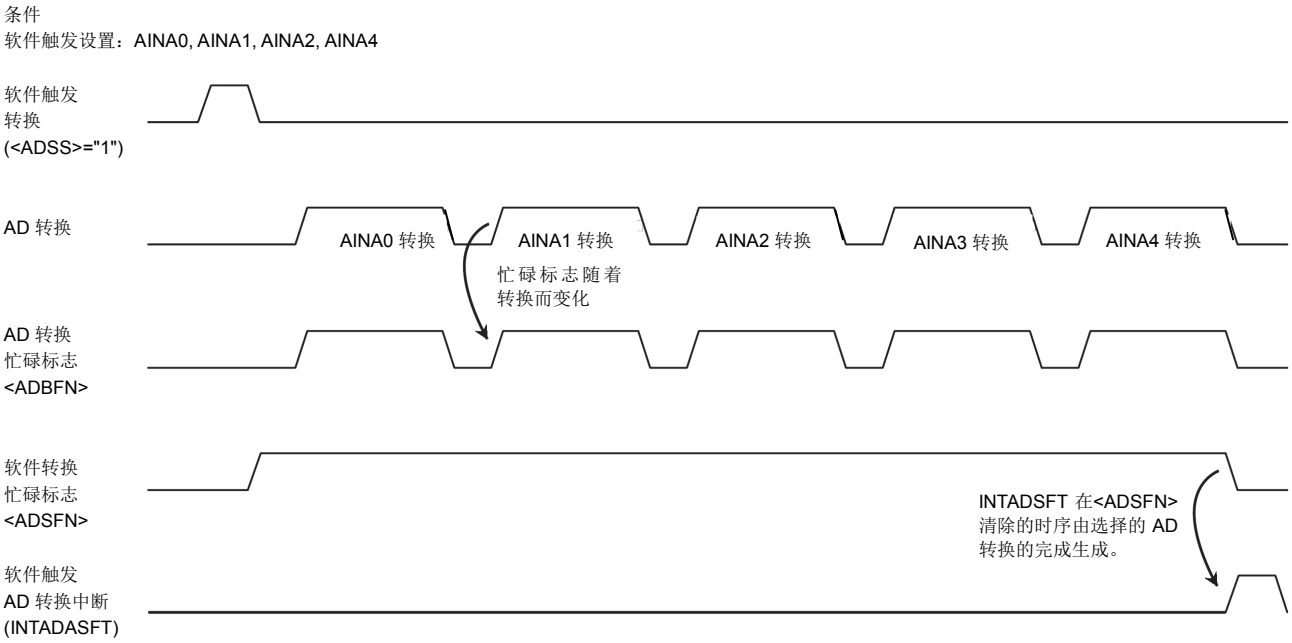


图 10-4 软件触发 AD 转换

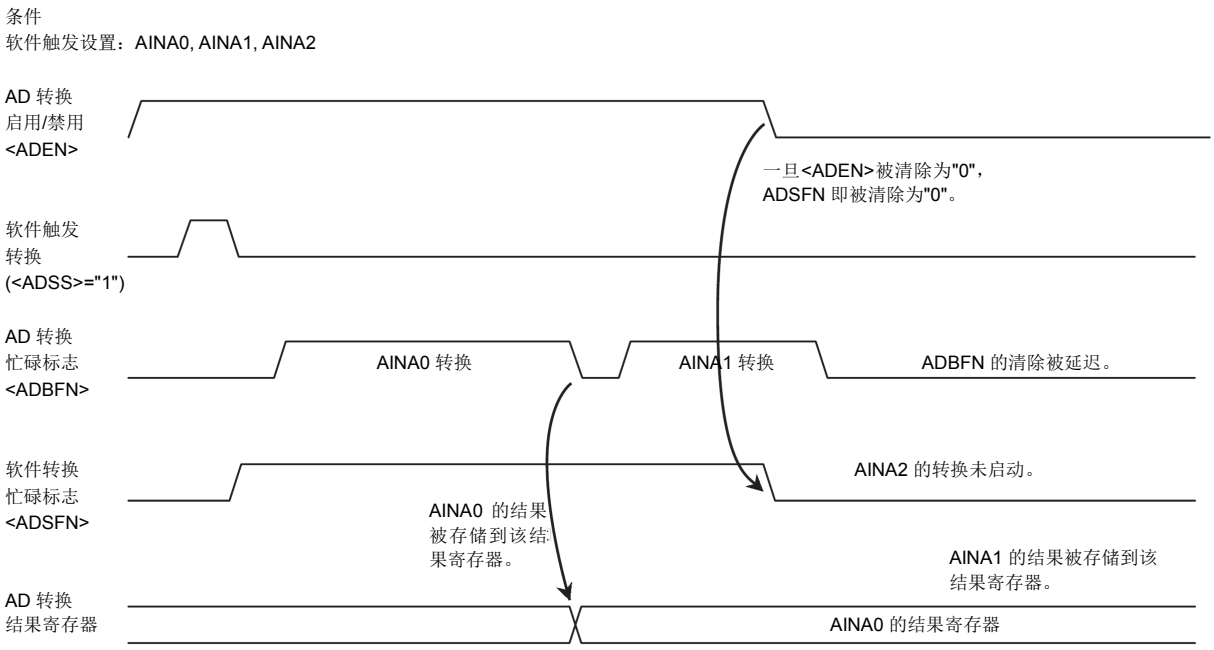


图 10-5 在软件触发 AD 转换期间将"0"写入到<ADEN>

10.6.2 恒定转换

在进行恒定转换时，如果在下一次转换完成时未从该转换结果寄存器读取前一次的结果，则溢出标志会被设置为"1"。在这种情况下，该转换结果寄存器中前一次的转换结果会被下一次的结果盖写掉。通过读取该转换结果，即可清除该溢出标志。(图 10-6)

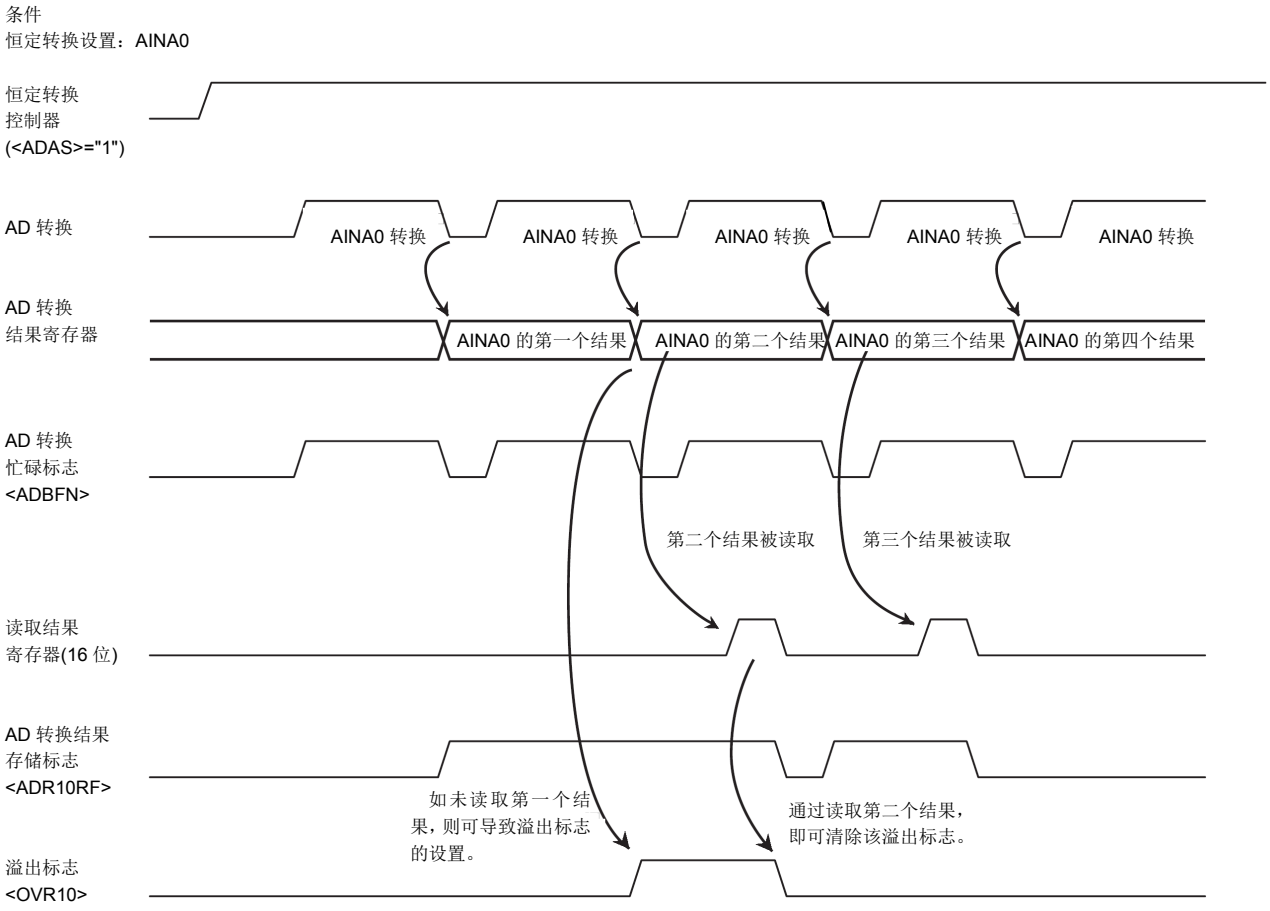


图 10-6 恒定转换

10.6.3 由触发器的 AD 转换

如果在软件触发转换期间发生 PMD 触发，则进行中的转换会立即停止。(图 10-7)如果在软件触发转换期间发生计时器触发，则进行中的转换会在进行中的转换完成之后停止。(图 10-8)在由触发信号启动的 AD 转换完成之后，经 ADxSSET03，ADxSSET47 与 ADxSSET811 编程的软件触发转换会从头开始。(图 10-9)

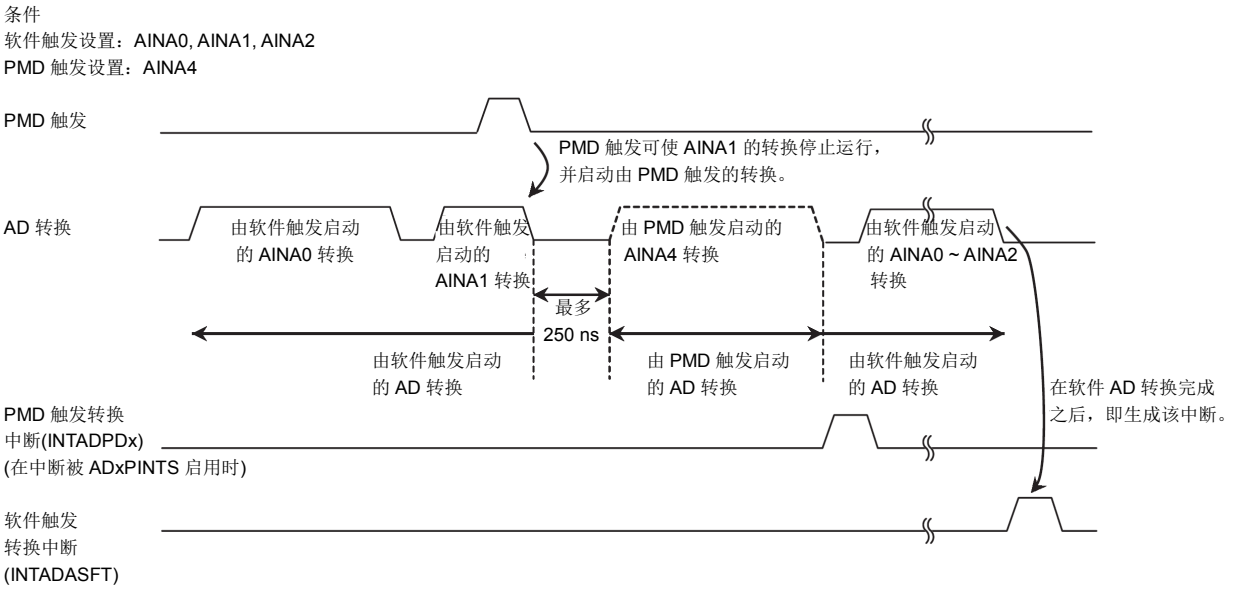


图 10-7 由 PMD 触发的 AD 转换

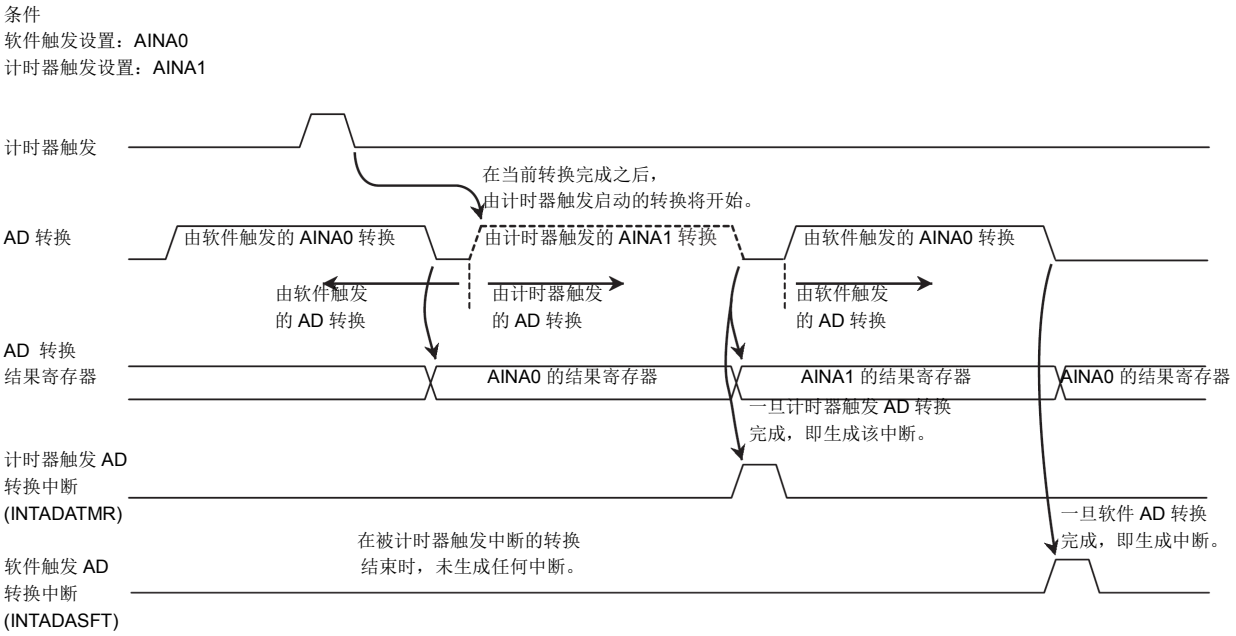


图 10-8 由计时器触发的 AD 转换(1)

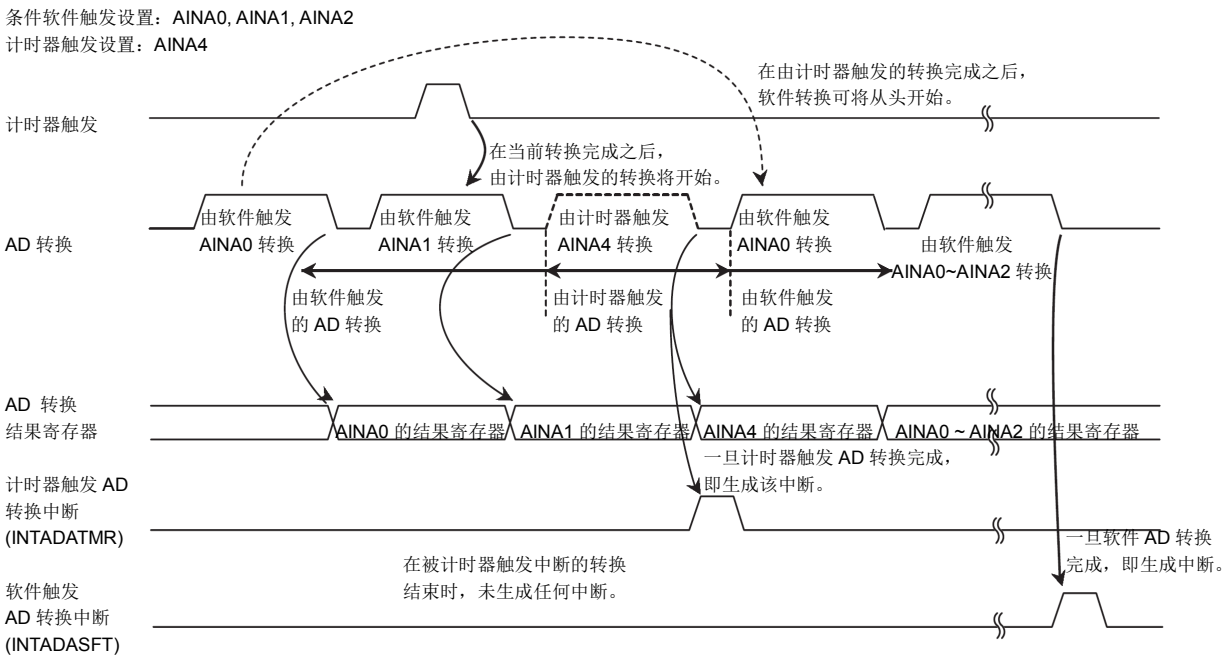
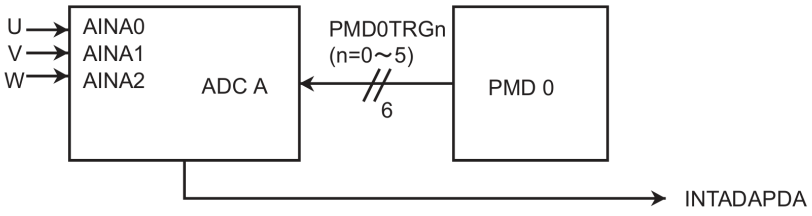


图 10-9 由计时器触发的 AD 转换(2)

10.7 使用示例

10.7.1 利用一个 PMD0 (三分路)与一个 ADC 进行的连续转换

以下给出了采用一个 PMD0 三分路与一个 ADC 的 AD 转换电路图。



ADC 设置示例给出如下。

ADC 装置 A

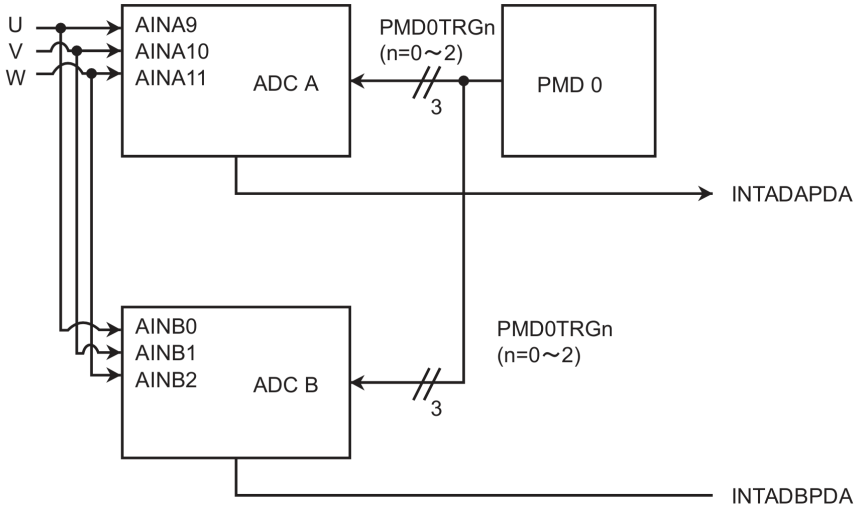
程序	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	A	A	A	A	A	A

程序 0 ~ 5 经指定，用于触发输入 PMD0TRG0 ~ 5。"reg0"与"reg1" 指 PMD 触发程序寄存器 ADAPSETn [7:0]与 ADAPSETn [15:8]。"U"，"V"与"W"指电机各相。通过选择 AIN 输入，即可获取这些相。

在触发脉冲输入发生时，可根据 reg0 与 reg1 顺次执行 AD 转换，然后生成该中断信号 (INTADAPDA)。

10.7.2 利用一个 PMD0 (三分路)与两个 ADC 进行的同步转换

以下给出了采用一个 PMD0(三分路)与两个 ADC 的 AD 转换方块图。



ADC 设置示例给出如下。

ADC 单元 A

程序	0	1	2
reg0	U	V	W
INT	A	A	A

ADC 单元 B

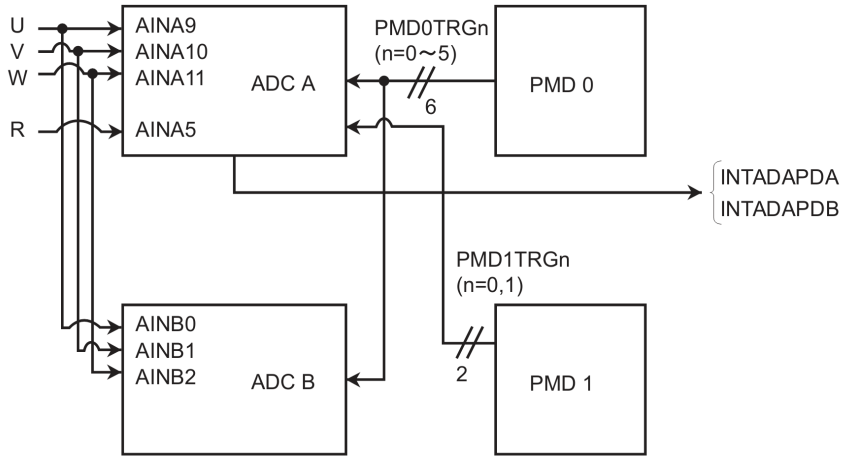
程序	0	1	2
reg0	U	V	W
INT	A	A	A

程序 0～2 已被分频给指向 ADC A 与 ADC B 的三个触发脉冲输入。"reg0"指 PMD 触发程序寄存器 ADAPSETn[7:0]与 ADBPSETn[7:0]。"U"，"V"与"W"指电机各相。通过选择 AIN 输入，即可获取这些相。

在触发脉冲输入发生时，ADC A 与 ADC B 即被同时启动以执行 AD 转换根据 reg 0，而中断信号(INTADAPDA, INTADBPDA)则被输出到 ADC A 与 ADC B。

10.7.3 利用 PMD0 (三分路)，PMD1 (一分路)与两个 ADC 进行的连续转换

以下给出了采用 PMD0 (三分路)，PMD1 (一分路)与两个 ADC 进行的 AD 转换的方块图。



ADC 设置示例给出如下。

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	PMD0 V	PMD0 W	PMD0 U
INT	-	-	-

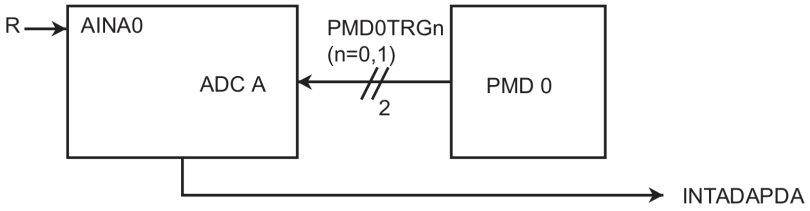
在 ADC A 中，程序 0～2 被指定给源自 PMD0 的六个触发信号，而程序 3 与 4 则被指定给源自 PMD1 的两个触发信号。在 ADC B 中，程序 0～2 被指定给源自 PMD0 的六个触发信号。

"reg0"，"reg1"与"reg2"指 PMD 触发程序寄存器 ADxPSETn[7:0]，ADxPSETn[15:8]与 ADxPSETn[23:16] (x=A,B : ADC 装置)。 "U"，"V"与"W"指电机各相。通过选择 AIN 输入，即可获取这些相。"R"指电阻器与该电阻器连接的 AIN 已设置。

在触发脉冲输入发生时，ADC A 或 ADC B 即被启动以执行 AD 转换。在 ADC A 中，所生成的中断 (INTADAPDA)用于源自 PMD0 的触发信号，而所生成的中断 (INTADAPDB)则用于源自 PMD1 的触发信号。在本例中，ADC B 中的中断发生被禁用。

10.7.4 利用一个 PMD0 (一分路)与一个 ADC 进行的连续转换

以下给出了采用 PMD0(一分路)与一个 ADC 的 AD 转换电路图。



ADC 设置示例给出如下。

ADC 单元 A

触发器	PMD0	PMD0
	0	1
程序	0	1
reg0	R	-
reg1	-	R
INT	-	A

程序 0 与 1 被指定给源自 PMD0 的两个触发信号。

"reg0"与"reg1" 指 PMD 触发程序寄存器 ADAPSETn [7:0]与 ADAPSETn [15:8]。"R"指电阻器与该电阻器连接的 AIN 输入已设置。

在触发脉冲输入发生时，ADC 即被启动以顺次执行程序 0 与 1。在程序 1 完成时，即生成中断(INTADAPDA)。

11. 电机控制电路(PMD：可编程电机驱动器)

该 TMPM370FYDFG/FYFG 包含 2 个通道可编程电机驱动器(PMD)。该产品的 PMD 具备导通输出控制与 DC 过电压检测等新增功能，可实现无传感器电机控制，并支持与 AD 转换器的互动。

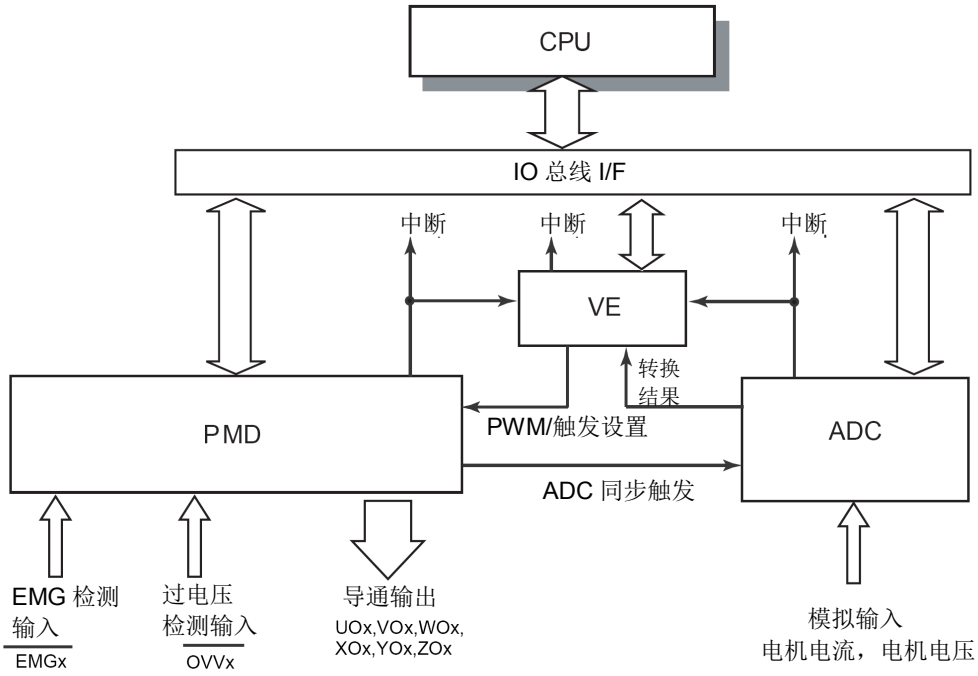


图 11-1 与电机控制相关的程序块结构

11.1 PMD 输入/输出信号

下表给出了输入到 PMD 的信号，以及从 PMD 输出的信号。

表 11-1 输入/输出信号

通道	引脚名称	PMD 信号名称	描述
PMD0	PC7/ $\overline{\text{OVV0}}$	OVV 0	OVV 状态信号
	PC6/ $\overline{\text{EMG0}}$	EMG 0	EMG 状态信号
	PC0/UO0	UO 0	U-相输出
	PC1/XO0	XO 0	X-相输出
	PC2/VO0	VO 0	V-相输出
	PC3/YO0	YO 0	Y-相输出
	PC4/WO0	WO 0	W-相输出
	PC5/ZO0	ZO 0	Z-相输出
PMD1	PG7/ $\overline{\text{OVV1}}$	OVV 1	OVV 状态信号
	PG6/ $\overline{\text{EMG1}}$	EMG 1	EMG 状态信号
	PG0/UO1	UO 1	U-相输出
	PG1/XO1	XO 1	X-相输出
	PG2/VO1	VO 1	V-相输出
	PG3/YO1	YO 1	Y-相输出
	PG4/WO1	WO 1	W-相输出
	PG5/ZO1	ZO 1	Z-相输出

11.2 PMD 电路

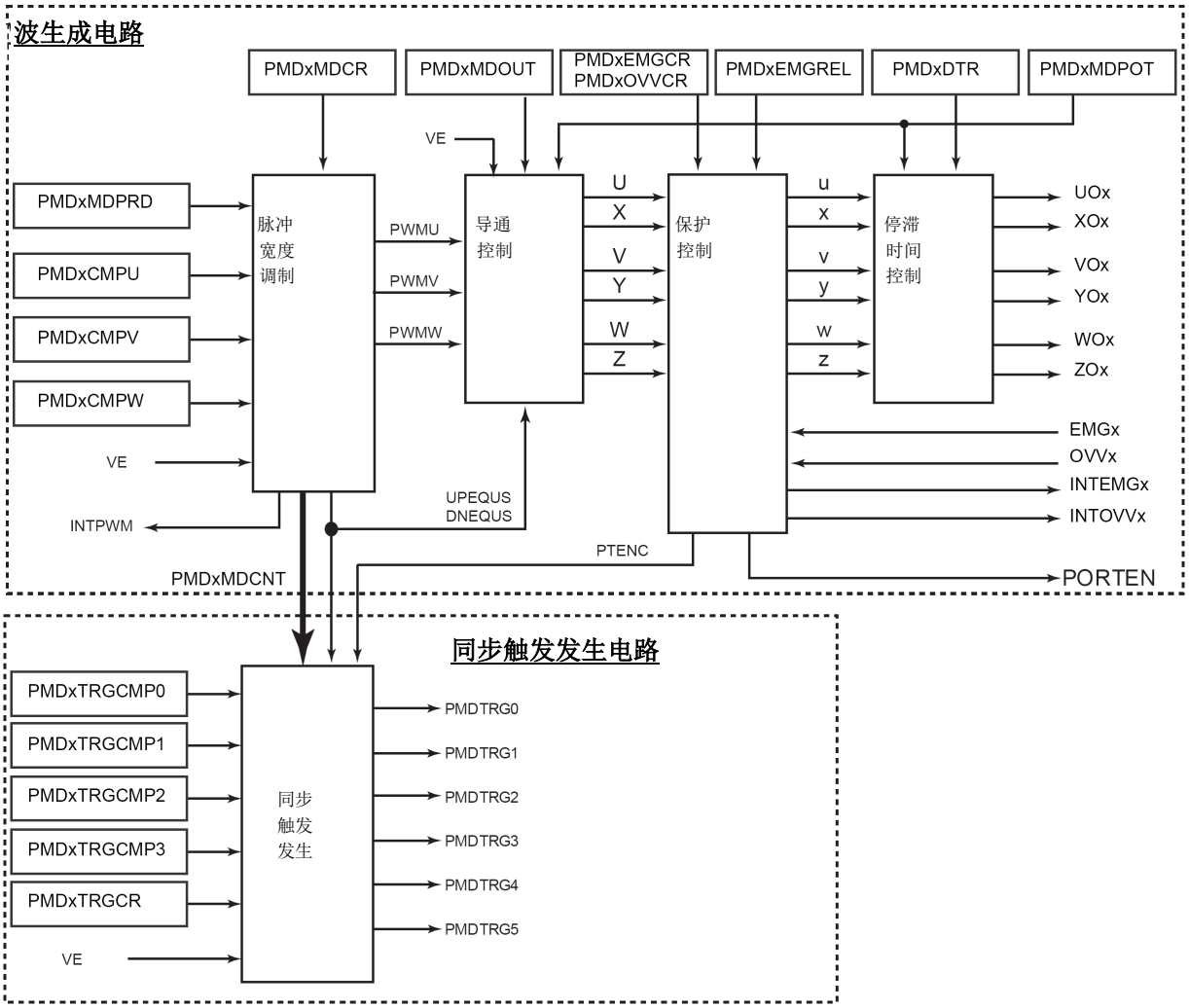


图 11-2 PMD 电路方块图

该 PMD 电路由两个程序块，即波发生电路与同步触发信号发生电路构成。

该波发生电路包括脉冲宽度调制电路，导通控制电路，保护控制电路，以及停滞时间控制电路。

- 该脉冲宽度调制电路可生成具备同一 PWM 频率的，独立的 3-相 PWM 波形。
- 该导通控制电路可决定 U，V 与 W 相各上下侧的输出模式。
- 该保护控制电路可通过 EMG 输入与 OVV 输入控制应急输出。
- 该停滞时间控制电路可防止在上下端切换时发生短路。
- 该同步触发信号发生电路可生成同步触发信号，并将其传送到 AD 转换器。

下表给出了与该 PMD 相关的各寄存器。

11.3 PMD 寄存器

通道 x	基址
通道 0	0x4005_0400
通道 1	0x4005_0480

寄存器名称		地址(基本+)
PMD 启动寄存器	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 计数寄存器	PMDxMDCNT	0x0010
PWM 周期寄存器	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 控输出控制寄存器	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
停滞时间寄存器	PMDxDTR	0x0044
触发比较 0 寄存器	PMDxTRGCMP0	0x0048
触发比较 1 寄存器	PMDxTRGCMP1	0x004C
触发比较 2 寄存器	PMDxTRGCMP2	0x0050
触发比较 3 寄存器	PMDxTRGCMP3	0x0054
触发控制寄存器	PMDxTRGCR	0x0058
触发输出模式设置寄存器	PMDxTRGMD	0x005C
触发输出选择寄存器	PMDxTRGSEL	0x0060
保留	-	0x007C

注：不要访问"保留"地址。

11.3.1 PMDxMDEN (PMD 启动寄存器)

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

位	比特符号	型号	功能
31-1	-	R	读作 0。
0	PWMEN	R/W	启用或禁用波形合成。 0：禁用 1：启用 用于使 PMD 在 PMD 被禁用时变成 High-z 的输出端口。 在启用 PMD 之前，设置<PWMEN>="1"(即可启用)输出端口极性等其它相关设置。

11.3.2 PMDxPORTMD (端口输出模式寄存器)

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

位	比特符号	型号	功能
31-2	-	R	读作 0。
1-0	PORTMD[1:0]	R/W	端口控制器设置 00：上相 = High-z /下相 = High-z 01：上相 = High-z / 下相 = PMD 输出 10：上相 = PMD 输出 / 下相 = High-z 11：上相 = PMD 输出 / 下相 = PMD 输出 该<PORTMD[1:0]>设置可控制各上相(U, V 与 W 相)和各下相(X, Y 与 Z 相)的外部端口输出。在发生工具中断，且已选择"High-Z"时， 各外部输出端口的上下相均被设置为 High-z。在其它情况下，各外部端口输出则取决于 PMD 输出。

注 1：在<PWMEN>=0 时，各输出端口会被设置为 High-z 不考虑输出端口设置。

注 2：在发生 EMG 输入时，各外部端口输出的控制视 PMDxEMGCR<EMGMD [1:0]>设置而定。

11.3.3 PMDxMODESEL (模式选择寄存器)

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

位	比特符号	型号	功能
31-1	-	R	读作 0。
0	MDSEL	R/W	模式选择寄存器 0：总线模式 1：VE 模式 该位可选择是否加载各双缓冲寄存器的第二缓冲区，且该寄存器值系通过该总线(总线模式)，或由矢量引擎(VE 模式)提供的值设置。各 PWM 比较寄存器(PMDxCMPU,PMDxCMPV, PMDxCMPW)，触发信号比较寄存器(PMDxTRGCMP0, PMDxTRGCMP1)，以及 PMDx MDOUT 寄存器为双缓冲型，且各第二缓冲区的加载与 PMD 的内部更新时序同步进行。

11.3.4 脉冲宽度调制电路

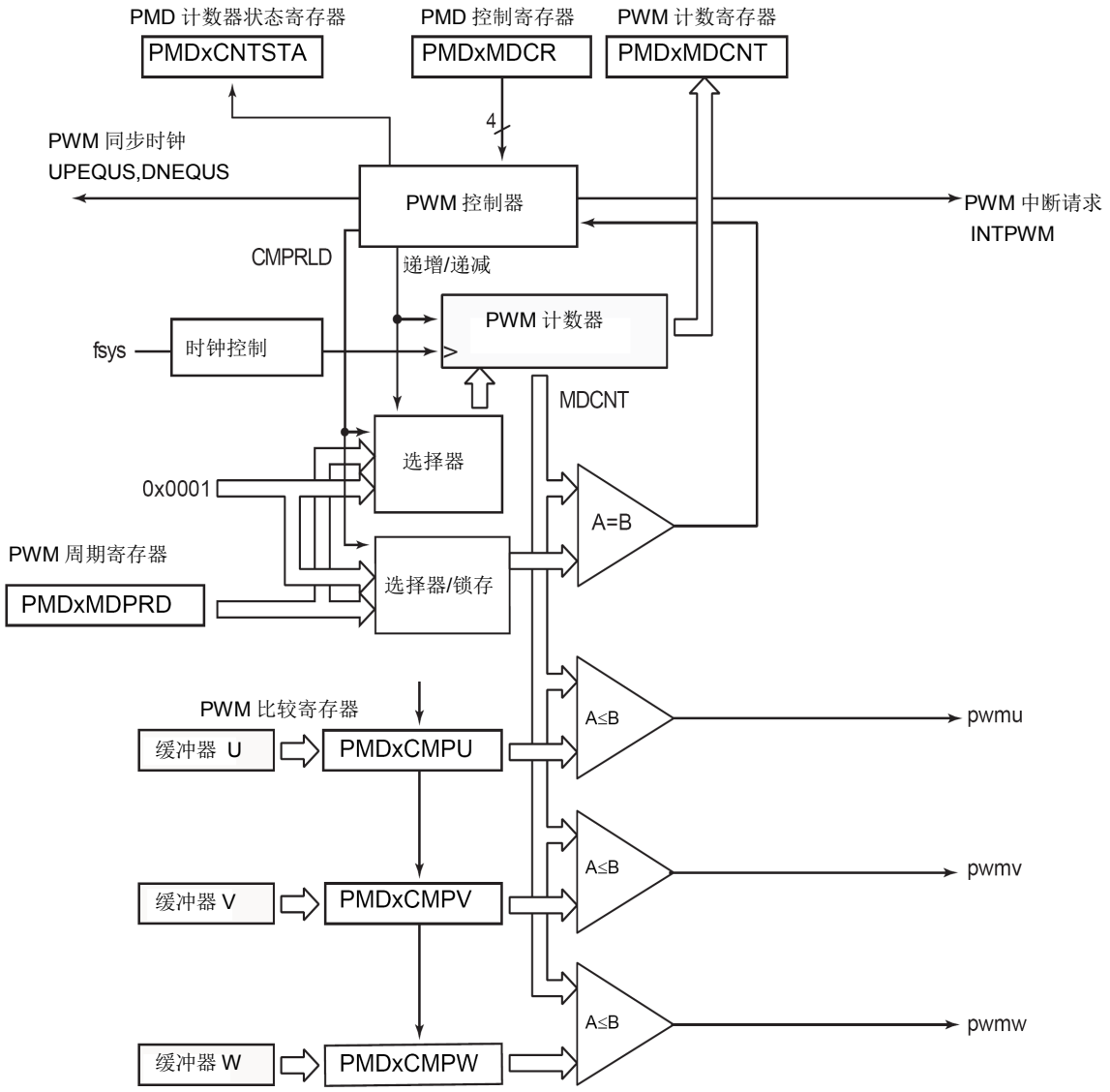


图 11-3 脉冲宽度调制电路

该脉冲宽度调制电路具备一个 16-位 PMD 递增/递减计数器，并可生成 80 MHz 时分辨率为 12.5 nsec 的 PWM 载波波形。可从模式 0(边缘对齐 PWM，锯齿波调制)与模式 1(中心对齐 PWM，三角波调制)选择该 PWM 载波波形模式。

该 PWM 周期扩展模式(PMDxMDCR<PWMCK> = 1) 也可用。在该模式被选中时，该 PWM 计数器可生成分辨率为 50 nsec 的 PWM 载波波形。

1. 设置 PWM 周期

PWM 周期由 PMDxMDPRD 寄存器决定。该寄存器为双缓冲型。比较器输入会在每个 PWM 周期接受更新。此外，还可每半个 PWM 周期更新一次比较器输入。

$$\text{锯齿波 PWM：PMDxMDPRD 寄存器值} = \frac{\text{振荡频率[Hz]}}{\text{PWM 频率[Hz]}}$$

$$\text{三角波 PWM: PMDxMDPRD 寄存器值} = \frac{\text{振荡频率[Hz]}}{\text{PWM 频率[Hz] x 2}}$$

2. 比较功能

该脉冲宽度调制电路可比较 3 相(PMDx-CMPU / V / W)的 PWM 比较寄存器与 PWM 计数器(PMDxMDCNT)所生成的载波了，以确定较大者，用于生成具备所需占空比的 PWM 波形。

各相的 PWM 比较寄存器均具备一个双缓冲型比较寄存器。每个 PWM 周期会加载一次 PWM 比较寄存器值(在内部计数器值可匹配 MDPRD<[15:0]>值时)。

此外，还可每 0.5 PWM 周期更新一次该比较寄存器。

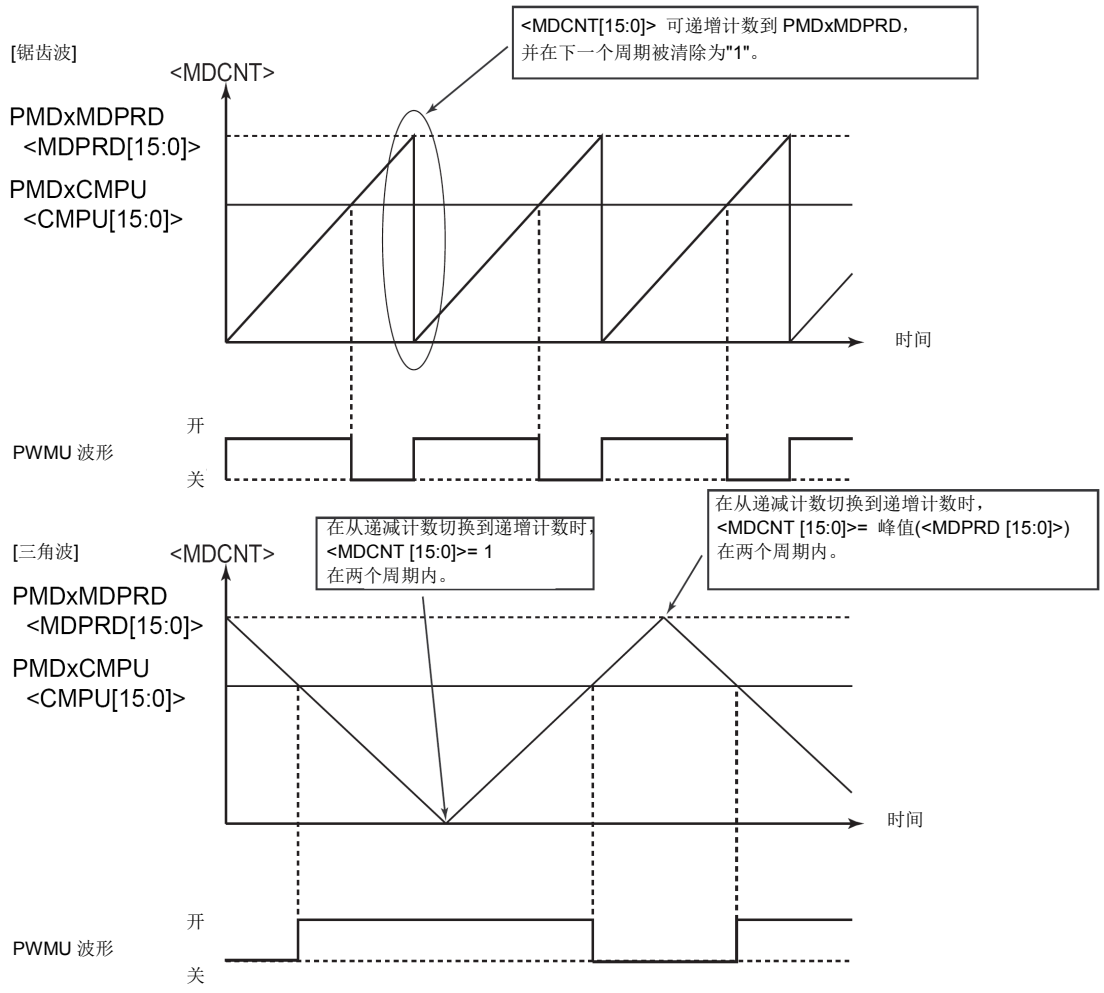


图 11-4 PWM 波形

3. 波形模式

在以下两种模式下，可生成三相 PWM 波形：

1. 3-相独立模式：

该三相的各 PWM 比较寄存器经单独设置后，可为各相生成单独的 PWM 波形。该模式用于生成正弦波等驱动波形。

2. 3-相共用模式：

仅 U-相 PWM 比较寄存器被设置成可为所有三相生成完全相同的 PWM 波形。该模式用于无刷直流电动机的矩形波驱动。

4. 中断处理

该脉冲宽度调制电路可生成与 PWM 波形同步的 PWM 中断请求。可将 PWM 中断周期设置为半个 PWM 周期，一个 PWM 周期，两个 PWM 周期或四个 PWM 周期。

11.3.4.1 PMDxMDCR (PMD 控制寄存器)

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

位	比特符号	型号	功能
31-7	-	R	读作 0。
6	PWMCK	R/W	PWM 周期扩展模式 0：标准周期 1：4x 周期 在<PWMCK>= "0"时，PWM 计数器按 12.5 ns 的分辨率运行 fsys = 80 MHz 时。 • 锯齿波：12.5 ns，三角波：25 ns 在<PWMCK>= "1"时，PWM 计数器按 50 ns 的分辨率运行 fsys = 80 MHz 时。 • 锯齿波：50 ns, 三角波：100 ns
5	SYNTMD	R/W	端口输出模式 该位可指定 U，V 与 W 相的端口输出设置(见表 11-2)。
4	DTYMD	R/W	占空比模式 0：3-相共用模式 1：3-相独立模式 该位可选择是否单独对各相进行占空比设置，或三相是否均采用 PMDxCMPU 寄存器。
3	PINT	R/W	PWM 中断定时 0：中断请求在 PWM 计数器 PMDxMDCNT<MDCNT[15:0]> = 0x0001 时 1：中断请求在 PWM 计数器 PMDxMDCNT<MDCNT[15:0]> = <MDPRD[15:0]>时 该位可选择是否在 PWM 计数器等于其最小值或最大值时生成中断请求。如边缘对齐 PWM 模式已被选中，则可在 PWM 计数器等于<MDPRD[15:0]>值时生成一个中断请求。如果 PWM 中断周期被设置为每 0.5PWM 周期一次，则在 PWM 计数器等于"1"或 <MDPRD[15:0]>时生成一个中断请求。
2-1	INTPRD[1:0]	R/W	PWM 中断周期 00：中断请求(每 0.5 PWM 周期一次(仅<PWMMD>= "1")) 01：中断请求(每个 PWM 周期一次) 10：中断请求(每 2 个 PWM 周期一次) 11：中断请求(每 4 个 PWM 周期一次) 该字段可选择的 PWM 中断周期为 0.5 个 PWM 周期，一个 PWM 周期，两个 PWM 周期，以及四个 PWM 周期。 •注) 在<INTPRD[1:0]>= "00"时，各比较寄存器(PMDxCMPU/V/W)与寄存器 (PMDxMDPRD)的内容，会在内部计数器等于 1 或 PMDxMDPRD 值时被更新到其相应的缓冲区内。
0	PWMMD	R/W	PWM 载波波形 0：PWM 模式 0 (边缘对齐 PWM，锯齿波) 1：PWM 模式 1(中心对齐 PWM，三角波) 该位可选择 PWM 模式。PWM 模式 0 为边缘对齐 PWM，而 PWM 模式 1 为中心对齐 PWM。

11.3.4.2 PMDxCNTSTA (PWM 计数器状态寄存器)

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

位	比特符号	型号	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 该位可指示该 PWM 计数器是为起床递增计数还是递减计数。 在边缘对齐 PWM 模式被选中时，该位始终被读作 0。

11.3.4.3 PMDxMDCNT(PWM 计数寄存器)

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	MDCNT[15:0]	R	PWM 计数器 PMD 计数器值 (分辨率：12.5 ns(fsys = 80 MHz 时)) 锯齿波 12.5 ns,三角波：25 ns • 在 PMDxMDCR<PWMCK> = 1 时，计数器分辨率变为 50 ns。 16-位计数器，用于读取周期计数值。其为只读型。 • 在 PMD 被禁用(<PWMEN>=0)时，PWM 计数器的值取决于<PWMMD> (PWM 载波波形)的设置。该值如以下所述。 如果 PMDxMDCR<PWMMD>= 0 : 0x0001 如果 PMDxMDCR<PWMMD>= 1 : PMDxMDPRD 的值<MDPRD[15:0]>

11.3.4.4 PMDxMDPRD(PWM 周期寄存器)

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	MDPRD[15:0]	R/W	PWM 周期 <MDPRD[15:0]> ≥ 0x010 16-位寄存器，用于指定 PWM 周期。该寄存器为双缓冲型，即使在 PWM 计数器运行期间也可对该寄存器进行更改。该缓冲区每个 PWM 周期加载一次。 (亦即在 PWM 计数器可匹配<MDPRD[15:0]>值时。如已选中 0.5 个 PWM 周期，则在 PWM 计数器可匹配 1 或<MDPRD[15:0]>时执行加载。必须将最小有效位设置为 0。) 如果<MDPRD[15:0]>被设置为小于 0x0010 的某个值，其会自动被假定为 0x0010(该寄存器可保留所写入的实际值。)

注： **不要以字节为单位写入到各寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

11.3.4.5 PMDxCMPU (U 相 PWM 比较寄存器)

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	CMPUx[15:0]	R/W	U 相的 PWM 脉冲宽度 比较寄存器(分辨率：12.5 ns(fsys = 80 MHz 时)) • 锯齿波 12.5 ns, 三角波：25 ns • 在 MDCR<PWMCK>="1"时，计数器分辨率变为 50 ns。 <CMPUx[15:0]> 为比较寄存器，可用于确定 U 相的输出脉冲宽度。这些寄存器均属于双缓冲型。通过比较该缓冲区与 PWM 计数器，判定哪一个较大，即可确定脉冲宽度(可在 PWM 计数器值可匹配 <MDPRD[15:0]>值加载。如已选中 0.5 个 PWM 周期，则在 PWM 计数器可匹配 1 或<MDPRD[15:0]>时执行加载)。 在读取该寄存器时，第一缓冲区的值(通过总线设置的数据)即被返回。

注 1：**通过将 PMDxMODESEL<MDSEL>设置为 0，选择总线模式(默认)，即可将通过该总线更新的比较寄存器中的值加载到第二缓冲区。**

注 2：**不要以字节为单位写入到这些寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

11.3.4.6 PMDxCMPV (V 相 PWM 比较寄存器)

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	CMPVx[15:0]	R/W	V 相 PWM 脉冲宽度 比较寄存器(分辨率：12.5 ns(fsys = 80 MHz 时)) • 锯齿波 12.5 ns, 三角波：: 25 ns • 在 MDCR<PWMCK>="1"时，计数器分辨率变为 50 ns。 <CMPVx [15:0]>为比较寄存器，可用于确定 V 相的输出脉冲宽度。 这些寄存器均属于双缓冲型。通过比较该缓冲区与 PWM 计数器，判定哪一个较大，即可确定脉冲宽度(可在 PWM 计数器值可匹配 <MDPRD[15:0]>值加载。如已选中 0.5 个 PWM 周期，则在 PWM 计数器可匹配 1 或<MDPRD[15:0]>时执行加载)。 在读取该寄存器时，第一缓冲区的值(通过总线设置的数据)即被返回。

注 1：**通过将 PMDxMODESEL 设置为 0，选择总线模式(默认)，即可将通过该总线更新的比较寄存器中的值加载到第二缓冲区。**

注 2：**不要以字节为单位写入到这些寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

11.3.4.7 PMDxCMPW (W 相 PWM 比较寄存器)

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	CMPWx[15:0]	R/W	W 相 PWM 脉冲宽度 比较寄存器(分辨率：12.5 ns(fsys = 80 MHz 时)) • 锯齿波：12.5 ns，三角波： 25 ns • 在 MDCR<PWMCK>="1"时，计数器分辨率变为 50 ns。 <CMPWx [15:0]>为比较寄存器，可用于确定 W 相输出脉冲宽度。这些寄存器均属于双缓冲型。通过比较该缓冲区与 PWM 计数器，判定哪一个较大，即可确定脉冲宽度(可在 PWM 计数器值可匹配 <MDPRD[15:0]>值加载。如已选中 0.5 个 PWM 周期，则在 PWM 计数器可匹配 1 或<MDPRD[15:0]>时执行加载)。 在读取该寄存器时，第一缓冲区的值(通过总线设置的数据)即被返回。

注 1：**通过将 PMDxMODESEL 设置为 0，选择总线模式(默认)，即可将通过该总线更新的比较寄存器中的值加载到第二缓冲区。**

注 2：**不要以字节为单位写入到这些寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

11.3.5 导通控制电路

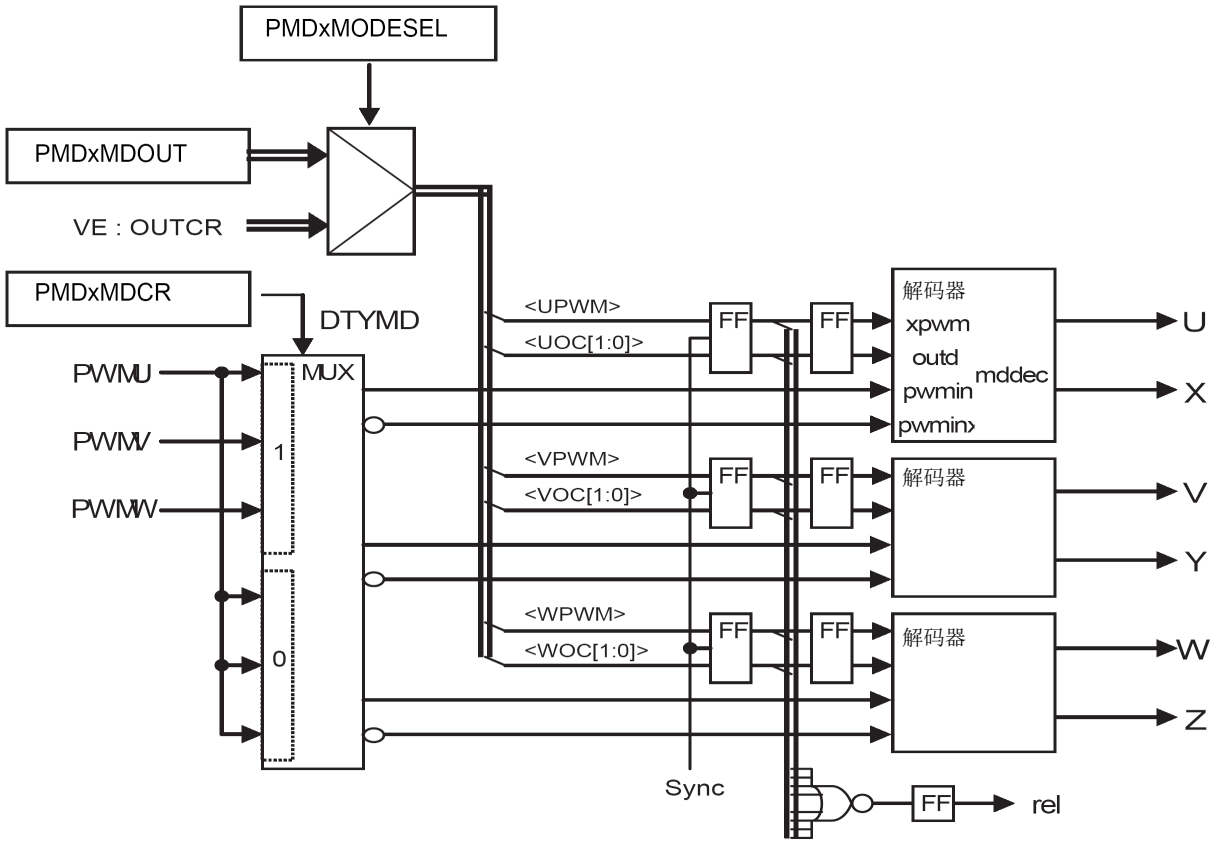


图 11-5 导通控制电路

该导通控制电路可按照"PMDxMDOUT"中所做的设置进行输出端口控制。该 PMDxMDOUT 寄存器位被分成两个部分：端口输出同步信号设置与端口输出设置。后一部分为双缓冲型，可同步或异步将更新时序设置到 PWM。

六个端口线路的输出设置，是通过 PMDxMDPOT<POLH><POLL>的位 10～位 8，以及 PMDxMDPOT 寄存器的位 3 与位 2，针对各上下相单独进行的。此外，PMDxMDOUT 寄存器的位 10～位 8 可针对 U 相，V 相与 W 相，选择 PWM 或高/低输出。如选中 PWM 输出，则输出 PWM 波形。如已选中高/低输出，则输出会被固定为高电平或低电平。表 11-2 按照 PMDxMDOUT 寄存器中的端口输出设置，以及 PMDxMDCR 寄存器中的极性设置，给出了各端口输出一览。

11.3.5.1 PMDxMDPOT (PMD 输出设置寄存器)

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

位	比特符号	型号	功能
31-4	-	R	读作 0。
3	POLH	R/W	上相端口极性(注) 0：低有效 1：高有效 POLH 可选择各上相的输出端口极性。
2	POLL	R/W	下相端口极性(注) 0：低有效 1：高有效 POLL 可选择各下相的输出端口极性。
1-0	PSYNCS[1:0]	R/W	MDOUT 传送时序(注) 00：与 PWM 异步 01：加载(在 PWM 计数器<MDCNT[15:0]> = 1 时) 10：加载(在 PWM 计数器<MOCNT[15:0]> = PMDxMDPRD<MDPRD[15:0]>时) 11：加载(在 PWM 计数器<MDCNT[15:0]> = 1 或 PMDxMDPRD<MDPRD[15:0]>时) 在端口输出可反映 U-， V-与 W-相输出设置(与 PWM 计数器峰值，下限或峰值/下限同步或异步)时，PSYNCS 可选择该定时。 如已选中"00" (与 PWM 异步)，则 MDOUT 寄存器的变更会被立即应用于 U-，V-与 W-相。在矢量引擎中，也有<PSYNCS>可用。

注：必须在 PMDxMDEN <PWMEN >=0 时设置该字段。

11.3.5.2 PMDxMDOUT(PMD 输出控制寄存器)

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

位	比特符号	型号	功能
31-11	-	R	读作 0。
10	WPWM	R/W	U-，V-与 W-相输出控制 0：高/低输出 1：PWM 输出 MDOUT 寄存器可控制 U 相，V 相与 W 相的端口输出(见下表 11-2)。
9	VPWM	R/W	
8	UPWM	R/W	
7-6	-	R	读作 0。
5-4	WOC[1:0]	R/W	U-，V-与 W-相输出控制 该 MDOUT 寄存器可控制 U 相，V 相与 W 相的端口输出(见下表 11-2)。
3-2	VOC[1:0]	R/W	
1-0	UOC[1:0]	R/W	

注 1：**通过将 PMDxMODESEL 设置为 0，选择总线模式(默认)，即可将通过该总线更新的值加载到 PWMxMDOUT 的第二缓冲区。**

注 2：**不要以字节为单位写入到各寄存器。 如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

表 11-2 端口输出(按照<UOC>, <VOC>, <WOC>, <UPWM>, <VPWM>与<WPWM>设置)

PMDxMDCR<SYNTMD>=0
极性：高有效(PMDxMDPOT<POLH><POLL>="11")

PMDxMDOUT 输出控制 (上相)	(下相)	<WPWM><VPWM><UPWM> 输出选择			
<WOC[1]>	<WOC[0]>	0：H/L 输出		1：PWM 输出	
<VOC[1]>	<VOC[0]>	上相	下相	上相	下相
<UOC[1]>	<UOC[0]>	输出	输出	输出	输出
0	0	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMDxMDCR<SYNTMD>=0
极性： 低有效(PMDxMDPOT<POLH><POLL>="00")

PMDxMDOUT 输出控制 (上相)	(下相)	<WPWM><VPWM><UPWM> 输出选择			
<WOC[1]>	<WOC[0]>	0：H/L 输出		1：PWM 输出	
<VOC[1]>	<VOC[0]>	上相	下相	上相	下相
<UOC[1]>	<UOC[0]>	输出	输出	输出	输出
0	0	H	H	PWM	$\overline{PWM}$
0	1	H	L	H	$\overline{PWM}$
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

PMDxMDCR<SYNTMD>=1
极性：高有效 (PMDxMDPOT<POLH><POLL>="11")

PMDxMDOUT 输出控制 (上相)	(下相)	<WPWM><VPWM><UPWM> 输出选择			
<WOC[1]>	<WOC[0]>	0：H/L 输出		1：PWM 输出	
<VOC[1]>	<VOC[0]>	上相	下相	上相	下相
<UOC[1]>	<UOC[0]>	输出	输出	输出	输出
0	0	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	$\overline{PWM}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMDxMDCR<SYNTMD>=0
极性：低有效 (PMDxMDPOT<POLH><POLL>="00")

PMDxMDOUT 输出控制 (上相)	(下相)	<WPWM><VPWM><UPWM> 输出选择			
<WOC[1]>	<WOC[0]>	0：H/L 输出		1：PWM 输出	
<VOC[1]>	<VOC[0]>	上相	下相	上相	下相
<UOC[1]>	<UOC[0]>	输出	输出	输出	输出
0	0	H	H	PWM	$\overline{PWM}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

• 输出设置适用于一分路模式

以下设置可支持一分路。

表 11-3 一个分路的寄存器设置

	标准 PWM 中心打开	**U-相 PWM 中心关闭**	**V-相 PWM 中心关闭**	**W-相 PWM 中心关闭**
CMPU	duty_U	<MDPRD[15:0]>-duty_U	duty_U	duty_U
CMPV	duty_V	duty_V	<MDPRD[15:0]>-duty_V	duty_V
CMPW	duty_W	duty_W	duty_W	<MDPRD[15:0]>-duty_W
<UOC[1:0]>	11	00	11	11
<VOC[1:0]>	11	11	00	11
<WOC[1:0]>	11	11	11	00

11.3.6 保护控制电路

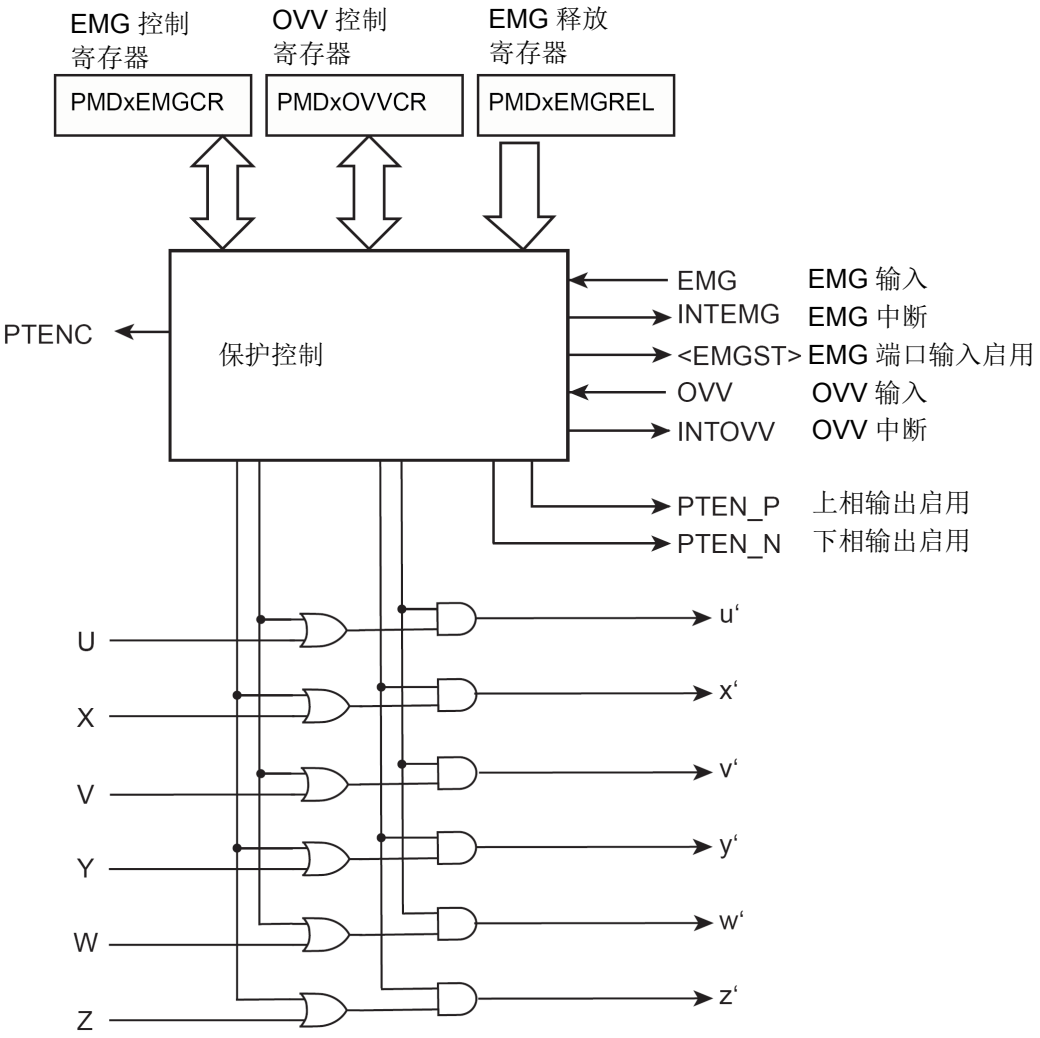


图 11-6 保护控制电路

该保护控制电路由一个 EMG 保护控制电路与一个 OVV 保护控制电路构成。

11.3.6.1 EMG 保护电路

该 EMG 保护电路由一个 EMG 保护控制单元与一个端口输出禁用单元构成。该电路在 EMG 输入变为低时即被激活。

该 EMG 保护电路可提供一个紧急停止机构：一旦 EMG 输入被断言 (H→L)时，所有六个端口输出即被禁用视 PMDxEMGCR<EMGMD>的设置而定，并生成一个 EMG(INTEMG)中断。<EMGMD>经过设置，可输出在紧急情况下将各外部输出端口设置为 High-z 的一个控制信号。

工具中断也可禁用所有六个 PWM 输出线视 PMDxPORTMD<PORTMD>设置而定。在发生工具中断时，通过 PMDxEMGSTA<EMGST >寄存器的设置，将各外部输出端口设为 High-z。

可通过 EMG 控制寄存器 (PMDxEMGCR)对 EMG 保护进行设置。

EMGSTA<EMGST>中的读取值 1 表示该 EMG 保护电路处于激活状态。在该状态下，通过将所有端口输出线均设置为非活动(PMDxMDOUT<[10:8]><[5:0]>)，并将 EMGCR<EMGRS>设置为 1，即可解除 EMG 保护。按该顺序将"0x5A" 与"0xA5"写入到 EMGREL 寄存器，并将 EMGCR<EMGEN>清除为 0，即可禁用 EMG 保护功能(必须连贯地执行这三条指令)。在 EMG 保护输入为低时，任何解除 EMG 保护状态的尝试均会被忽略。在将 PMDxMGCR<EMGRS>设置为 1 以解除 EMG 保护之前，确认 PMDxEMGST<EMGI>为高。

仅可在所指定的键控代码("0x5A", "0xA5")被写入到该<EMGREL>寄存器中以防止其无意中被禁用之后，该 EMG 保护电路才可被禁用。

注：EMG 功能的初始步骤

复位后，EMG 功能即被启用，但 EMG 引脚即被配置为标准端口。因此，在该 EMG 保护可能仍为有效时，应按初始序列的以下步骤解除 EMG 保护。

1：通过 PxFR 寄存器选择 EMG 功能。

2：读取 PMDxEMGSTA<EMGI>并确认其为 "1"。

3：将 PMDxMDOUT<[10:8]>, <[5:0]>设置为"0"，使得所有端口均进入非激活状态("L"输出)。

4：将 PMDxEMGCR<EMGRS>设置为"1"，从而解除 EMG 保护。

如拟禁用 EMG 保护，则应继续执行以下步骤。

5：将键控代码写入到 PMDxEMGREL (按 "0x5A"与"0xA5"的顺序)。

6：将 PMDxEMGCR<EMGEN> 设置为"0"，以禁用 EMG 保护。

11.3.6.2 PMDxEMGREL (EMG 解除寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-0	EMGREL[7:0]	W	EMG 禁用代码 按该顺序将 0x5A 与 0xA5 设置为 <EMGREL[7:0]>寄存器的位 7 ~ 0，即可禁用 EMG 与 OVV 保护功能。 在禁用这些功能时，<EMGEN>与<OVVEN>必须被清除为"0"。 • 该寄存器可用于 EMG 与 OVV 功能。

11.3.6.3 PMDxEMGCR (EMG 控制寄存器)

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

位	比特符号	型号	功能
31-12	-	R	读作 0。
11-8	EMGCNT[3:0]	R/W	EMG 输入检测时间 按以下公式计算出静噪消除时间值。 <EMGCNT[3:0]> × 16/fsys (分辨率：200[nsec]80 MHz 时) <EMGCNT[3:0]>= 0 ~ 15 (在<EMGCNT[3:0]> = 0 时，静噪滤波器即被旁路。)
7-6	-	R	读作 0。
5	INHEN	R/W	工具中断启用/禁用 0：禁用 1：启用 在 PMD 停止信号被从该工具输入时，该位可选择是否停止该 PMD。在初始状态下，工具中断即被启用。
4-3	EMGMD[1:0]	R/W	EMG 保护模式选择 00：PWM 输出控制被禁用 /端口输出 = 所有相为 High-z 01：所有上相 ON，所有下相 OFF /端口输出 = 下相为 High-z 10：所有上相 OFF ，所有下相 ON /端口输出 = 上相为 High-z 11：所有相 OFF/ 端口输出 = 所有相为 High-z • ON = PWM 输出(无输出控制)，OFF = 低[在<POLL>，<POLH> = 1 (高态有效)] 在紧急情况下，该字段可控制上下各相的 PWM 输出与端口输出。
2	EMGISEL	R/W	EMG 输入选择 0：输入端口 1：比较器输出 该位可选择是否将端口输入或比较器输出作为 EMG 信号输入该保护电路。
1	EMGRS	W	EMG 保护解除 0：- 1：解除保护 通过将 PMDxMDOUT 寄存器设置为 0，并将<EMGRS>位设置为 1，即可解除 EMG 保护。 该位始终被读作 0。 • PMDxMDOUT 寄存器务必将 0 写入上位[10:8]与下位[5:0]。 • 在解除 EMG 保护之前，确认 PMDxEMGSTA<EMGI>已恢复为高。
0	EMGEN	R/W	EMG 保护电路启用/禁用 0：禁用 1：启用 通过将该位设置为 1，即可启用 EMG 保护电路。在初始状态下，EMG 保护电路会被启用。 按该顺序将 0x5A 与 0xA5 写入到 PMDxEMGREL<EMGREL>寄存器，并将 EMGEN 位清除为 0，即可禁用该电路(必须连贯地执行这三条指令)。

11.3.6.4 PMDxEMGSTA (EMG 状态寄存器)

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

位	比特符号	型号	功能
31-2	-	R	读作 0。
1	EMGI	R	EMG 输入 EMG 保护状态 通过读取该位即可了解 EMG 输入状态
0	EMGST	R	EMG 保护状态 0：正常工作 1：保护 通过读取该位即可了解 EMG 保护状态。

11.3.6.5 OVV 保护控制电路(OVV 块)

OVV 保护控制电路由一个 OVV 保护控制单元与一个端口输出禁用单元构成。该电路在 OVV 输入端口被断言时即被激活。

在指定周期(在 OVVCR<OVVCNT>中设置)的 OVV 输入被断言(H→L)时，该 OVV 保护电路就会将导通控制电路中的六根端口输出线固定为高或低。此时即生成 OVV 中断(INTOVV)。

有可能仅关闭各上相或下相，或所有的相。

可通过"PMDxOVVCR"设置 OVV 保护。PMDxOVVSTA<OVVST>中的读取值"1" 表示 OVV 保护电路处于活动状态。

通过将 PMDxOVVCR<OVVRS>设置为"1"，即可解除 OVV 保护状态。此时，OVV 保护会在 OVV 保护电路完成其运行后被自动解除。

(在 OVV 保护输入为低时，OVV 保护状态不会被解除。通过读取 PMDxOVVSTA<OVVI>，即可检查该端口的状态。)

OVV 保护状态的解除可与 PWM 周期(在 PWM 计数可匹配 <MDPRD[15:0]>值时)值同步进行。如已选中 0.5 个 PWM 周期，则解除时序即为 PWM 计数器等于 1 或<MDPRD[15:0]>时的时间点。)按该顺序将"0x5A" 与"0xA5"写入到<EMGREL [7:0]>，并将 PMDxOVVCR<OVVEN>清除为 0，即可禁用 OVV 保护功能(必须连贯地执行这三条指令)。

仅可在所指定的键控代码("0x5A", "0xA5")被写入到该<EMGREL[7:0]>寄存器中以防止其无意中被禁用之后，该 OVV 保护电路才可被禁用。

11.3.6.6 PMDxOVVCR (OVV 控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	OVVCNT			
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-12	-	R	读作 0。
11-8	OVVCNT[3:0]	R/W	OVV 输入检测时间 OVVCNT = 1 ~ 15 (如果被设置为 0，则作为 1 处理)。 OVVCNT × 16/fsys (分辨率：200[nsec] 80 MHz 时) • OVVCNT 仅在端口输入被选为 OVV 信号时有效(<OVVISEL> = "1")。
7	-	R	读作 0。
6	ADIN1EN	R/W	ADC B 监控中断输入启用 0：禁用输入 1：启用输入 该位可选择是否启用或禁用源自 ADC B 的监控信号输入。 在该位被设置为启用，且<OVVISEL>="1"时，由 AD 转换结果与所指定比较值之间匹配生成的 ADC B 的中断信号，即可将 PMD 置于保护状态(如果 OVV 保护已被启用)。 • 有关详细资料见 ADC 相关节次。
5	ADIN0EN	R/W	ADC A 监控中断输入启用 0：禁用 1：启用 该位可选择是否启用或禁用源自 ADC A 的监控信号输入。 在该位被设置为启用，且<OVVISEL>="1"时，由 AD 转换结果与所指定比较值之间匹配生成的 ADC A 的中断信号，即可将 PMD 置于保护状态(如果 OVV 保护已被启用)。 • 有关详细资料见 ADC 相关节次。
4-3	OVVMD[1:0]	R/W	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(ON = 高，OFF = 低[在 <POLL>，<POLH> = 1 时 (高态有效)]) 在 OVV 条件发生时，该字段可控制各上下相的输出。 • 如果 OVV 与 EMG 条件同时发生，则<EMGMD[1:0]>寄存器中的保护模式设置生效。
2	OVVISEL	R/W	OVV 输入选择 0：端口输入 1：ADC 监控信号 该位可选择是否将端口输入或源自 ADC 的监控信号作为 OVV 信号输入该保护电路。 • 如已选中 ADC 监控信号，则<OVVCNT[3:0]>失效。
1	OVVRS	R/W	OVV 保护解除 0：禁用 OVV 保护的自动解除 1：启用 OVV 保护的自动解除 在过电压检测信号引起高 ~ 低推移时，即进入 OVV 保护状态。在过电压检测信号恢复为高之后，通过将该位设置为"1"，PWM 计数器与<MDPRD[15:0]>)寄存器之间的匹配可自动解除 OVV 保护状态。 • 如已选中 0.5 个 PWM 周期(PMDxMDCR<INTPRD[1:0]> = "00")，则在 PWM 计数器等于 "1" 或<MDPRD[15:0]>时，OVV 保护状态即被解除。
0	OVVEN	R/W	OVV 保护电路启用/禁用 0：禁用 1：启用 通过将该位设置为 1，即可启用 OVV 保护电路。在初始状态下，OVV 保护电路会被启用。 按该顺序将"0x5A" 与"0xA5"写入到<EMGREL[7:0]>寄存器，并将<OVVEN>位清除为"0"，即可禁用该电路(必须连贯地执行这三条指令)。

11.3.6.7 PMDxOVVSTA (OVV 状态寄存器)

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

位	比特符号	型号	功能
31-2	-	R	读作 0。
1	OVVI	R	OVVI 输入 OVVI 状态 通过读取该位，即可了解该 OVV 输入状态(如已被 OVVCR<OVVISEL>选中)。
0	OVVST	R	OVV 保护状态 0：正常工作 1：保护 通过读取该位即可了解该 OVV 状态。

11.3.7 停滞时间电路

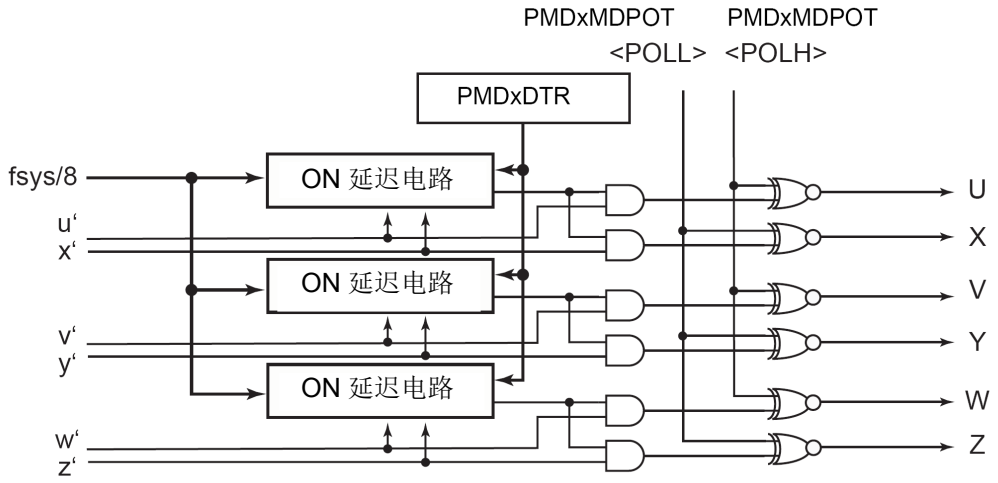


图 11-7 停滞时间电路

该停滞时间电路由一个停滞时间单元与一个输出端极性切换单元构成。

针对各 U，V 与 W 相，在各上下相被切换时，该 ON 延迟电路可引入一个延时(停滞时间) ，以防止发生短路。该停滞时间被作为一个 8-位值 80 MHz 的分辨率为 100 ns 设置到该停滞时间寄存器(PMDxDTR<DTR[7:0]>)。

该输出端极性切换电路允许通过 PMDxMDPOT<POLH>与<POLL>，单独设置各上下相的极性(高态有效或低态有效)。

11.3.7.1 PMDxDTR (停滞时间寄存器)

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

位	比特符号	型号	功能
31-8	-	R	读作 0。
7-0	DTR[7:0]	R/W	停滞时间 按以下公式计算出该停滞时间值。 100 ns × <DTR[7:0]> (最多 25.5 μsec fsys = 80 MHz 时)

注：**在 PMDxMDEN<PWMEN> = 1 时，不要改变<DTR[7:0]>寄存器。**

11.3.8 同步触发发生电路

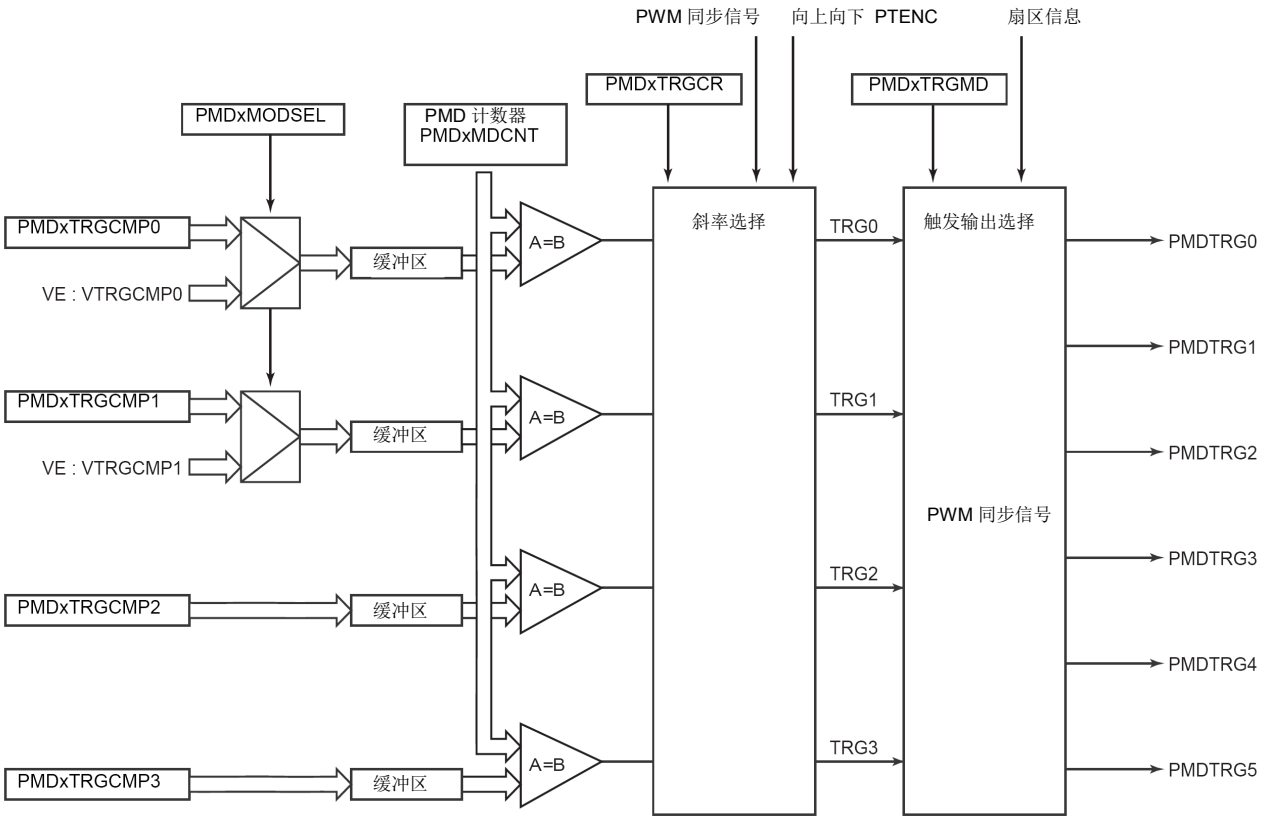


图 11-8 同步触发发生电路

该同步触发信号发生电路可生成用以启动 ADC 采样与 PWM 同步的触发信号。PMDxMDCNT 和 PMDxTRGCMP 之间的匹配可生成 ADC 触发信号(PMDTRG)。可从递增计数匹配，递减计数匹配与递增/递减计数匹配中，选择信号发生时序。如已选中该边缘对齐 PWM 模式，则会在存在递增计数匹配时生成 ADC 触发信号。如 PWM 输出被禁用(PMDxMDEN<PWMEN> = 0)，则触发输出随即被禁用。

如已选择该触发选择输出模式，则会按照 PMDxTRGSEL<TRGSEL>寄存器设置或源自矢量引擎的扇区信息，切换该触发输出端口。

11.3.8.1 PMDxTRGCMP0 (触发比较寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TRGCMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP0							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	TRGCMP0 [15:0]	R/W	触发输出比较寄存器 在 PMD 计数器值<MDCNT[15:0]>可匹配 TRGCMP0 中的设置值时，PMDTRG 即被输出。 在读取 TRGCMP0 时，双缓冲区中第一缓冲区的值(通过总线设置的数据)即被返回。 TRGCMP0 的设置范围应为 1 ~ [<MDPRD[15:0]>设定值− 1]。 禁止将<TRGCMP0>设置为 0 或<MDPRD[15:0]>值。

注 1：**通过将 PMDxMODESEL<MDSEL>设置为"0"，选择总线模式(默认)，即可将 TRGCMP0 与 TRGCMP1 中的数据加载到第二缓冲区。**

注 2：**不要以字节为单位写入到这些寄存器。 如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

注 3：**在<TRGCMP0>被设置为 0x0001 时，在 PWM 启动之后(<PWMEN> = 1)，仅在首个周期内无触发输出。**

11.3.8.2 PMDxTRGCMP1 (触发信号比较寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TRGCMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP1							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	TRGCMP1 [15:0]	R/W	触发输出比较寄存器 在 PMD 计数器值<MDCNT[15:0]>可匹配 TRGCMP1 中的设置值时，PMDTRG 即被输出。 在读取 TRGCMP1 时，双缓冲区中第一缓冲区的值(通过总线设置的数据)即被返回。 TRGCMP1 的设置范围应为 1 ~ [<MDPRD[15:0]>设定值 = 1]。 禁止将<TRGCMP1>设置为 0 或<MDPRD[15:0]>值。

注 1：**通过将 MODESELPMDxMODESEL<MDSEL>设置为"0"，选择总线模式(默认)，即可将 TRGCMP0 与 TRGCMP1 中的数据加载到第二缓冲区。**

注 2：**不要以字节为单位写入到这些寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

注 3：**在<TRGCMP1>被设置为 0x0001 时，在 PWM 启动之后(MDEN <PWMEN >=1)，仅在首个周期内无触发输出。**

11.3.8.3 PMDxTRGCMP2 (触发信号比较寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TRGCMP2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP2							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	TRGCMP2 [15:0]	R/W	触发输出比较寄存器 在 PMD 计数器值<MDCNT[15:0]>可匹配 TRGCMP2 中的设置值时，PMDTRG 即被输出。 在读取 TRGCMP2 时，双缓冲区中第一缓冲区的值(通过总线设置的数据)即被返回。 TRGCMP2 的设置范围应为 1 ~ [<MDPRD[15:0]>设定值 − 1]。 禁止将<TRGCMP2>设置为 0 或<MDPRD[15:0]>值。

注 1：**不要以字节为单位写入到这些寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

注 2：**在<TRGCMP2>被设置为"0x0001"时，在 PWM 启动之后(MDEN <PWMEN >="1")，仅在首个周期内无触发输出。**

11.3.8.4 PMDxTRGCMP3 (触发信号比较寄存器 3)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TRGCMP3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRGCMP3							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	TRGCMP3 [15:0]	R/W	触发输出比较寄存器 在 PMD 计数器值<MDCNT[15:0]>可匹配 TRGCMP3 中的设置值时，PMDTRG 即被输出。 在读取 TRGCMP3 时，双缓冲区中第一缓冲区的值(通过总线设置的数据)即被返回。 TRGCMP3 的设置范围应为 1 ~ [<MDPRD[15:0]>设定值 − 1]。 禁止将<TRGCMP3>设置为 0 或<MDPRD[15:0]>值。

注 1：**不要以字节为单位写入到这些寄存器。如果单独写入上 8 位[15:8]与下 8 位[7:0]，则无法保证运行。**

注 2：**在<TRGCMP3>被设置为"0x0001"时，在 PWM 启动之后(<PWMEN> = 1)，仅在首个周期内无触发输出。**

触发比较寄存器的更新时序(TRGCMPx)

该触发比较寄存器(TRGCMPx)属于双缓冲型。已写入到 TRGCMPx 的数据被加载到第二缓冲区的时序，取决于 PMDxTRGCR<TRGxMD[2:0]>的设置。在 PMDxTRGCR<TRGxBE>被设置为"1"时，已写入 TRGCMPx 的数据会被立即加载到第二缓冲区。

表 11-4 TRGCMPx 缓冲区更新时序按触发输出模式设置

<TRGxMD[2:0]>设置	**TBUFx 更新时序**
000：触发输出被禁用	始终更新
001：递减计数匹配时触发输出	在 PWM 计数器等于 MDPRD(PWM 载波峰值)时更新
010：递增计数匹配时触发输出	在 PWM 计数器等于"1"(PWM 载波下限)时更新
011：递增/递减计数匹配时触发输出	在 PWM 计数器等于"1"或 MDPRD 时更新 (PWM 载波峰值/下限)
100：PWM 载波峰值时触发输出	始终更新
101：PWM 载波下限时触发输出	
110：PWM 载波峰值/下限时触发输出	
111：触发输出被禁用	

11.3.8.5 PMDxTRGCR (触发控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	-	R	读作 0。
15	TRG3BE	R/W	PMDTRG3 缓冲区更新时序 0：同步 1：异步(直接反映被写入到 PMDTRG3 的值。) 该位可启用 PMDTRG3 缓冲区的异步更新。
14-12	TRG3MD[2:0]	R/W	PMDTRG3 模式设置 000：触发输出被禁用 001：递减计数匹配时触发输出 010：递增计数匹配时触发输出 011：递增/递减计数匹配时触发输出 100：PWM 载波峰值时触发输出 101：PWM 载波下限时触发输出 110：PWM 载波峰值/下限时触发输出 111：触发输出被禁用 该寄存器可选择触发输出时序。 在 PMDxMDCR<PMDMD> 被设置为边缘对齐模式时，即使已选择递减计数匹配或 PWM 载波下限，在出现递增计数匹配或 PWM 载波峰值时仍可进行触发输出。 • 在<TRG3MD[2:0]>="011"，PMDxTRGCMP3="0x0001"，且 PMDxMDCR<PWMMD>="1"(三角波)时，每个周期会进行一次触发输出。
11	TRG2BE	R/W	PMDTRG2 缓冲区更新时序 0：同步 1：异步(直接反映被写入到 PMDTRG2 的值。) 该位可启用 PMDTRG2 缓冲区的异步更新。
10-8	TRG2MD[2:0]	R/W	PMDTRG2 模式设置 000：触发输出被禁用 001：递减计数匹配时触发输出 010：递增计数匹配时触发输出 011：递增/递减计数匹配时触发输出 100：PWM 载波峰值时触发输出 101：PWM 载波下限时触发输出 110：PWM 载波峰值/下限时触发输出 111：触发输出被禁用 该寄存器可选择触发输出时序。 在 PMDxMDCR<PMDMD> 被设置为边缘对齐模式时，即使已选择递减计数匹配或 PWM 载波下限，在出现递增计数匹配或 PWM 载波峰值时仍可进行触发输出。 • 在<TRG2MD[2:0]>="011"，PMDxTRGCMP2="0x0001"，且 PMDxMDCR<PWMMD>="1"(三角波)时，每个周期会进行一次触发输出。
7	TRG1BE	R/W	PMDTRG1 缓冲区更新时序 0：同步 1：异步(直接反映被写入到 PMDTRG1 的值。) 该位可启用 PMDTRG1 缓冲区的异步更新。
6-4	TRG1MD[2:0]	R/W	PMDTRG1 模式设置 000：触发输出被禁用 001：递减计数匹配时触发输出 010：递增计数匹配时触发输出 011：递增/递减计数匹配时触发输出 100：PWM 载波峰值时触发输出 101：PWM 载波下限时触发输出 110：PWM 载波峰值/下限时触发输出 111：触发输出被禁用 该寄存器可选择触发输出时序。 在 PMDxMDCR<PMDMD> 被设置为边缘对齐模式时，即使已选择递减计数匹配或 PWM 载波下限，在出现递增计数匹配或 PWM 载波峰值时仍可进行触发输出。 • 在<TRG1MD[2:0]>="011"，PMDxTRGCMP1="0x0001"，且 PMDxMDCR<PWMMD>="1"(三角波)时，每个周期会进行一次触发输出。
3	TRG0BE	R/W	PMDTRG0 缓冲区更新时序 0：同步 1：异步(直接反映被写入到 PMDTRG0 的值。) 该位可启用 PMDTRG0 缓冲区的异步更新。

位	比特符号	型号	功能
2-0	TRG0MD[2:0]	R/W	PMDTRG0 模式设置 000：触发输出被禁用 001：递减计数匹配时触发输出 010：递增计数匹配时触发输出 011：递增/递减计数匹配时触发输出 100：PWM 载波峰值时触发输出 101：PWM 载波下限时触发输出 110：PWM 载波峰值/下限时触发输出 111：触发输出被禁用 该寄存器可选择触发输出时序。 在 PMDxMDCR<PMDMD> 被设置为边缘对齐模式时，即使已选择递减计数匹配或 PWM 载波下限，在出现递增计数匹配或 PWM 载波峰值时仍可进行触发输出。 • 在<TRG0MD[2:0]>="011"，PMDxTRGCMP0="0x0001"，且 PMDxMDCR<PWMMD>="1" (三角波)时，每个周期会进行一次触发输出。

11.3.8.6 PMDxTRGMD (触发输出模式设置寄存器)

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

位	比特符号	型号	功能
31-2	-	R	读作 0。
1	TRGOUT	R/W	触发输出模式 0：固定触发输出 1：可调触发输出 在<TRGOUT>="0"时，触发输出"PMDTRG0 ~ PMDTRG3"可输出分别由<TRGCMP0> ~ <TRGCMP3>的匹配所生成的触发信号。PMDTRG4 与 PMDTRG5 则被固定在某一低电平。在 <TRGOUT>="1"时，会按照<TRGSEL>设置或源自矢量引擎的扇区信息，切换 PMDxTRGCMP0 所引发的触发输出。有关详细资料见下表。
0	EMGTGE	R/W	EMG 保护状态下的输出启动 0：在该保护状态下禁用触发输出 1：在该保护状态下启用触发输出 该位可启用或禁用 EMG 保护状态下的触发输出。

表 11-5 触发信号输出模式

<TRGOUT>设置	比较寄存器	<TRGSEL[2:0]>设置	触发输出
<TRGOUT>=0	PMDxTRGCMP0	×	PMDTRG0
	PMDxTRGCMP1		PMDTRG1
	PMDxTRGCMP2		PMDTRG2
	PMDxTRGCMP3		PMDTRG3
<TRGOUT>=1	PMDxTRGCMP0	0	PMDTRG0
		1	PMDTRG1
		2	PMDTRG2
		3	PMDTRG3
		4	PMDTRG4
		5	PMDTRG5
	PMDxTRGCMP1	×	无触发输出
	PMDxTRGCMP2	×	无触发输出
	PMDxTRGCMP3	×	无触发输出

11.3.8.7 PMDxTRGSEL (触发输出选择寄存器)

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

位	比特符号	型号	功能
31-2	-	R	读作 0。
2-0	TRGSEL[2:0]	R/W	触发输出选择 000：从 PMDTRG0 输出 001：从 PMDTRG1 输出 010：从 PMDTRG2 输出 011：从 PMDTRG3 输出 100：从 PMDTRG4 输出 101：从 PMDTRG5 输出 110：无触发输出 111：无触发输出 在选择可调触发输出模式时(PMDxTRGMD<TRGOUT>="1")，该字段有效。 PMD 计数器与 PMDxTRGCMP0 值之间的匹配，可引发所选触发信号的输出(见表 11-5) 。

12. 矢量引擎(VE)

12.1. 概述

12.1.1 特征

矢量引擎具有下列特征：

1. 执行矢量控制基本任务(坐标转换，相位变换和 SIN/COS 计算)。

   使用固定点格式数据。

   → 软件无需控制小数点对齐。

2. 启用的接口(输出控制，触发器生成，输入处理)与电动机控制电路

   (PMD：可编程电动机驱动器)与 AD 转换器(ADC)。

- 将计算结果从固定点格式转换为 PMD 可用的数据格式。
- 生成与 PMD 和 ADC 交货操作的时序数据。
- 将 AD 转换结果转化为固定点格式。

3. 使用与固定点格式的最大值相对的标准化数值计算电流，电压和转速。
4. 在电流控制中执行 PI 控制。
5. 实施相位插值(转速集成)

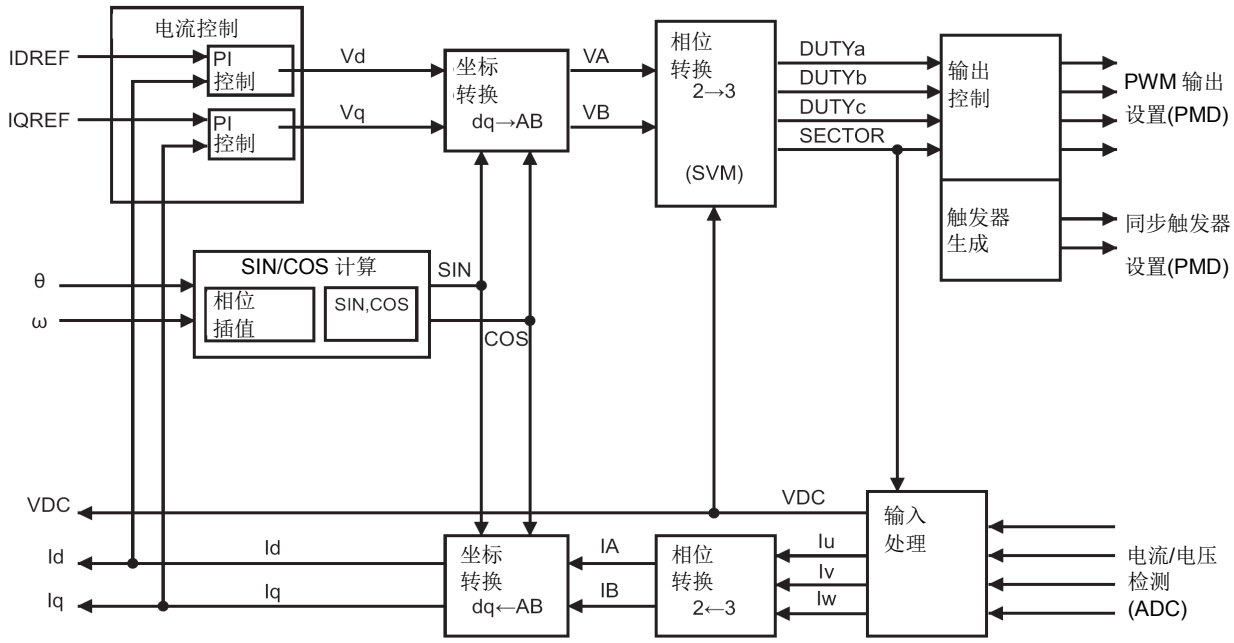


图 12-1 矢量控制方块图

12.1.2 关键规范

1. 空间矢量转换主要用于 2-相- ~ - 3-相的转换。同时支持 2-相调制与 3-相调制。
2. 可对无需传感器的电流检测生成 ADC 取样时序。可采用 1-分流器，3-分流器和 2-传感器方法进行电流检测。
3. 在电流控制中，可对 d-轴和 q-轴独立实施 PI 控制。无需使用电流控制也可直接提供参考电压信息。
4. 使用串联值，通过近似方法进行 SIN/COS 计算。
5. 使用相位插值，通过转速直接规定或计算相位信息。

注 1：使用矢量引擎时，必须通过模式选择寄存器(PMDxMODESEL)将 PMD 设置为 VE 模式。

注 2：此外，还有必要在 ADC 中对 PMD 的每一个同步触发器进行适当设置(启用触发器，选择 AIN 和准备使用的结果寄存器)。

12.2. 配置

图 12-2 所示为矢量引擎的配置。

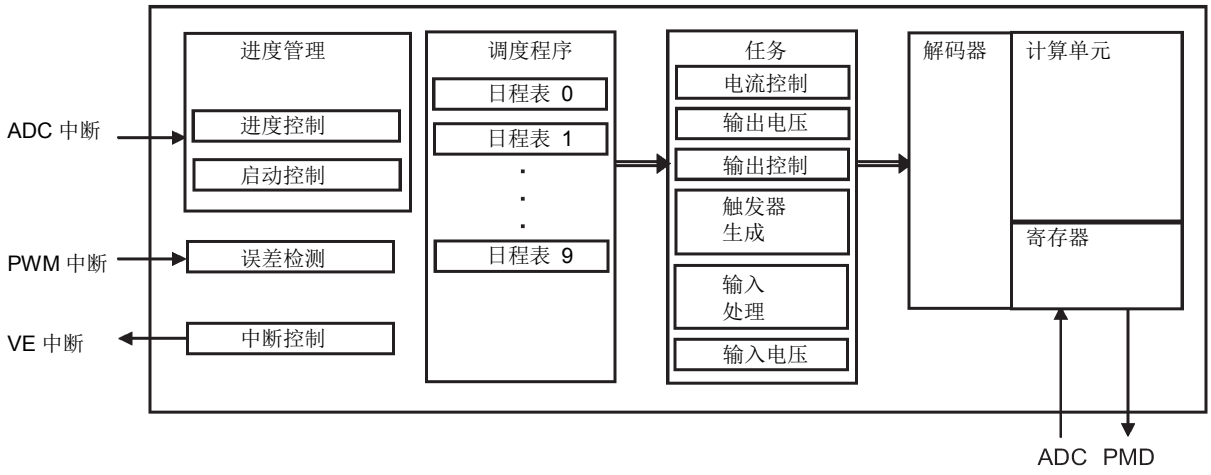


图 12-2 矢量引擎的配置

12.2.1 矢量引擎，电动机控制电路与 A/D 转换器之间的交互作用

矢量引擎可通过与电动机控制电路(PMD)和两台 AD 转换器(ADC)的交互作用控制两台电动机。矢量引擎通道 0 控制 PMD 通道 0；矢量引擎通道 1 控制 PMD 通道 1。

如图 12-3 所示，矢量引擎允许与 PMD 和 ADC 直接互动。

将 PMD0MODESEL 寄存器设置为 VE 模式时，PMD 通道 0 寄存器 PMD0CMPU，PMD0CMPV，PMD0CMPW， PMD0MDOUT，PMD0TRGCMP0，PMD0TRGCMP1 与 PMD0TRGSEL 将分别切换至矢量引擎寄存器 VECMPU0，VECMPV0，VECMPW0，VEOUTCR0，VETRGCMP00，VETRGCMP10 和 VETRGSEL0。 同样，PMD 通道 1 寄存器将切换至矢量引擎寄存器 VECMPU1，VECMPV1，VECMPW1，VEOUTCR1，VETRGCMP01，VETRGCMP11 与 VETRGSEL1。 在此情况下，这些寄存器只能通过矢量引擎控制，而不能从 PMD 写入。 其他 PMD 寄存器没有读/写限制。

ADC A 单元寄存器 ADAREG0 ， ADAREG1 ， ADAREG2 ， ADAREG3 和 ADABPSETn<UVWISn0[1:0]>， <UVWISn1[1:0]>，<UVWISn2[1:0]>， <UVWISn3[1:0]> 将被读入矢量引擎，分别作为矢量引擎寄存器 VEADREG0A，VEADREG1A，VEADREG2A，VEADREG3A，VEPHNUM0A，VEPNNUM1A，VEPHNUM2A 和 VEPHNUM3A。(这些寄存器不能从 CPU 存取。)同样，ADC B 单元寄存器将被读入矢量引擎，分别作为矢量引擎寄存器 VEADREG0B，VEADREG1B，VEADREG2B，VEADREG3B，VEPHNUM0B，VEPHNUM1B，VEPHNUM2B 与 VEPHNUM3B。 (这些寄存器不能从 CPU 存取。)这些 ADC 寄存器可从 ADC 写入和读取。

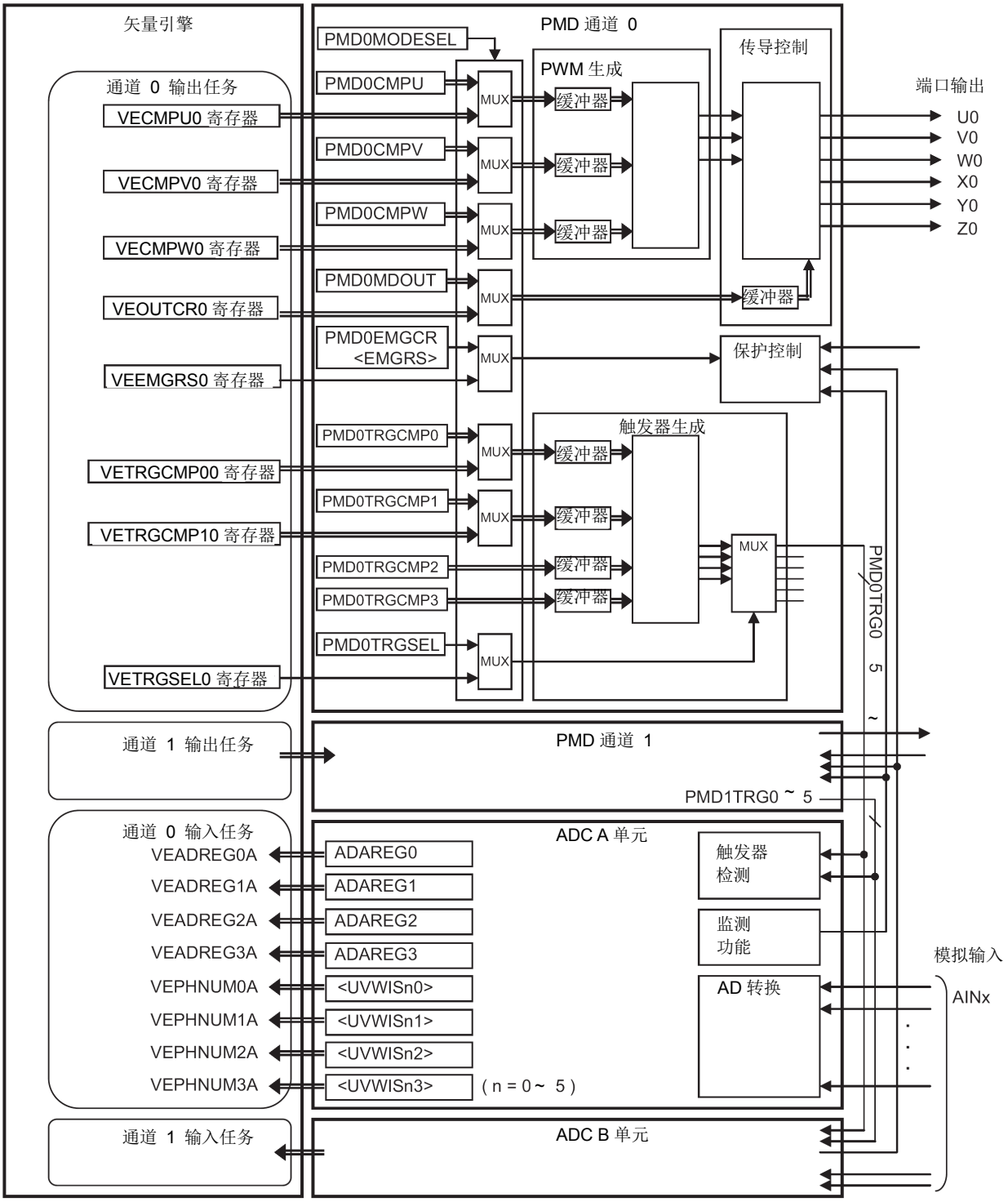


图 12-3 矢量引擎，PMD 与 ADC 之间的交互作用

12.3 寄存器列表

矢量引擎寄存器可分为下列三种类型：

• VE 控制寄存器

矢量引擎控制寄存器与临时寄存器

• 公用寄存器

两个通道公用的寄存器

• 特定通道专用寄存器

每一个通道的计算数据与控制寄存器

12.3.1 寄存器列表

VE 控制寄存器

寄存器名称			地址
VE 启用/禁用	VEEN	R/W	0x4005_0000
CPU 启动触发器选择	VECPURUNTRG	W	0x4005_0004
任务选择	VETASKAPP	R/W	0x4005_0008
运行计划表选择	VEACTSCH	R/W	0x4005_000C
计划表重复计数	VEREPTIME	R/W	0x4005_0010
启动触发器模式	VETRGMODE	R/W	0x4005_0014
错误中断允许/禁止	VEERRINTEN	R/W	0x4005_0018
VE 强行终止	VECOMPEND	W	0x4005_001C
误差检测	VEERRDET	R	0x4005_0020
计划表执行标志/执行任务	VESCHTASKRUN	R	0x4005_0024
保留	-	R	0x4005_0028
临时 0	VETMPREG0	R/W	0x4005_002C
临时 1	VETMPREG1	R/W	0x4005_0030
临时 2	VETMPREG2	R/W	0x4005_0034
临时 3	VETMPREG3	R/W	0x4005_0038
临时 4	VETMPREG4	R/W	0x4005_003C
临时 5	VETMPREG5	R/W	0x4005_0040
保留	-	R	0x4005_01BC

公用寄存器

寄存器名称			地址
保留	-	R/W	0x4005_0174
ADC 转换时间(基于 PWM 时钟)	VETADC	R/W	0x4005_0178

通道 0 专用寄存器

寄存器名称			地址
状态标志	VEMCTLF0	R/W	0x4005_0044
任务控制模式	VEMODE0	R/W	0x4005_0048
流量控制	VEFMODE0	R/W	0x4005_004C
PWM 周期率(PWM 时期[s] ×最大转速 (见注 1) × 2^{16})设置	VETPWM0	R/W	0x4005_0050
转速(转速[Hz] ÷最大转速(注 1)×2^{15}) 设置	VEOMEGA0	R/W	0x4005_0054
电动机相位(电动机相位[度]/360×2^{16})设置	VETHETA0	R/W	0x4005_0058
d-轴参考值(电流[A] ÷ 最大电流 (注 2)×2^{15})设置	VEIDREF0	R/W	0x4005_005C
q-轴参考值(电流[A] ÷ 最大电流 (注 2)×2^{15})设置	VEIQREF0	R/W	0x4005_0060
d-轴电压(电压[V] ÷ 最大电压(注 3)×2^{31})设置	VEVD0	R/W	0x4005_0064
q-轴电压(电压[V] ÷ 最大电压(注 3)×2^{31})设置	VEVQ0	R/W	0x4005_0068
d-轴 PI 控制整系数	VECIDKI0	R/W	0x4005_006C
d-轴 PI 控制比例系数	VECIDKP0	R/W	0x4005_0070
q-轴 PI 控制整系数	VECIQKI0	R/W	0x4005_0074
q-轴 PI 控制比例系数	VECIQKP0	R/W	0x4005_0078
d-轴电压积分项(VDI)上 32 位	VEVDIH0	R/W	0x4005_007C
d-轴电压积分项(VDI)下 32 位	VEVDILH0	R/W	0x4005_0080
q-轴电压积分项(VQI)下 32 位	VEVQIH0	R/W	0x4005_0084
q-轴电压积分项(VQI)下 32 位	VEVQILH0	R/W	0x4005_0088
切换速度(用于 2-相调制与 PWM 移位)	VEFPWMCHG0	R/W	0x4005_008C
PWM 周期(其设定值与 PMD 的 PWM 周期一致)	VEMDPRD0	R/W	0x4005_0090
最小脉冲宽度	VEMINPLS0	R/W	0x4005_0094
同步触发器校正值	VETRGCRC0	R/W	0x4005_0098
保留	-	R/W	0x4005_009C
输出转换 THETA 时间递耗余弦值(Q15 数据)	VECOS0	R/W	0x4005_00A0
输出转换 THETA 时间递耗正弦值(Q15 数据)	VESIN0	R/W	0x4005_00A4
输入处理以前的余弦值(Q15 数据)	VECOSM0	R/W	0x4005_00A8

通道 0 专用寄存器

寄存器名称			地址
输入处理以前的正弦值(Q15 数据)	VESINM0	R/W	0x4005_00AC
扇区信息	VESECTOR0	R/W	0x4005_00B0
输入处理以前的扇区信息	VESECTORM0	R/W	0x4005_00B4
a-相零电流 AD 模数转换结果(注 4)	VEIAO0	R/W	0x4005_00B8
b-相零电流 AD 模数转换结果(注 4)	VEIBO0	R/W	0x4005_00BC
c-相零电流 AD 模数转换结果(注 4)	VEICO0	R/W	0x4005_00C0
a-相电流 AD 模数转换结果(注 4)	VEIAADC0	R/W	0x4005_00C4
b-相电流 AD 模数转换结果(注 4)	VEIBADC0	R/W	0x4005_00C8
c-相电流 AD 模数转换结果(注 4)	VEICADC0	R/W	0x4005_00CC
DC 直流电源电压(电压 [V] ÷ 最大电压(注 3) × 2^{15})	VEVDC0	R/W	0x4005_00D0
d-轴电流(电流 [A] ÷ 最大电流(注 2) × 2^{31})	VEID0	R/W	0x4005_00D4
q-轴电流(电流 [A] ÷ 最大电流(注 2) × 2^{31})	VEIQ0	R/W	0x4005_00D8
PMD 控制：CMPU 设置	VECMPU0	R/W	0x4005_017C
PMD 控制：CMPV 设置	VECMPV0	R/W	0x4005_0180
PMD 控制：CMPW 设置	VECMPW0	R/W	0x4005_0184
PMD 控制：输出控制(MDOUT)	VEOUTCR0	R/W	0x4005_0188
PMD 控制：TRGCMP0 设置	VETRGCMP00	R/W	0x4005_018C
PMD 控制：TRGCMP1 设置	VETRGCMP10	R/W	0x4005_0190
PMD 控制：触发器选择	VETRGSEL0	R/W	0x4005_0194
PMD 控制：EMG 返回	VEEMGRS0	W	0x4005_0198

通道 1 专用寄存器

寄存器名称			地址
状态标志	VEMCTLF1	R/W	0x4005_00DC
任务控制模式	VEMODE1	R/W	0x4005_00E0
流量控制	VEFMODE1	R/W	0x4005_00E4
PWM 周期率(PWM 时期[s] ×最大转速 (注 1) × 2^{16})设置	VETPWM1	R/W	0x4005_00E8
转速(转速[Hz] ÷最大转速(注 1)×2^{15}) 设置	VEOMEGA1	R/W	0x4005_00EC
电动机相位(电动机相位[度]/360×2^{16})设置	VETHETA1	R/W	0x4005_00F0
d-轴参考值(电流[A] ÷ 最大电流 (注 2)×2^{15})设置	VEIDREF1	R/W	0x4005_00F4
q-轴参考值(电流[A] ÷ 最大电流 (注 2)×2^{15})设置	VEIQREF1	R/W	0x4005_00F8
d-轴电压(电压[V] ÷ 最大电压(注 3)×2^{31})设置	VEVD1	R/W	0x4005_00FC
q-轴电压(电压[V] ÷ 最大电压(注 3)×2^{31})设置	VEVQ1	R/W	0x4005_0100
d-轴 PI 控制整系数	VECIDKI1	R/W	0x4005_0104
d-轴 PI 控制比例系数	VECIDKP1	R/W	0x4005_0108
q-轴 PI 控制整系数	VECIQKI1	R/W	0x4005_010C
q-轴 PI 控制比例系数	VECIQKP1	R/W	0x4005_0110
d-轴电压积分项(VDI)上 32 位	VEVDIH1	R/W	0x4005_0114
d-轴电压积分项(VDI)下 32 位	VEVDILH1	R/W	0x4005_0118
q-轴电压积分项(VQI)下 32 位	VEVQIH1	R/W	0x4005_011C
q-轴电压积分项(VQI)下 32 位	VEVQILH1	R/W	0x4005_0120
切换速度(用于两相调制与 PWM 移位)	VEFPWMCHG1	R/W	0x4005_0124
PWM 周期(其设定值与 PMD 的 PWM 周期一致)	VEMDPRD1	R/W	0x4005_0128

通道 1 专用寄存器

寄存器名称			地址
最小脉冲宽度	VEMINPLS1	R/W	0x4005_012C
同步触发器校正值	VETRGCRC1	R/W	0x4005_0130
保留	-	R/W	0x4005_0134
输出转换 THETA 时间递耗余弦值(Q15 数据)	VECOS1	R/W	0x4005_0138
输出转换 THETA 时间递耗正弦值(Q15 数据)	VESIN1	R/W	0x4005_014C
输入处理以前的余弦值(Q15 数据)	VECOSM1	R/W	0x4005_0140
输入处理以前的正弦值(Q15 数据)	VESINM1	R/W	0x4005_0144
扇区信息	VESECTOR1	R/W	0x4005_0148
输入处理以前的扇区信息	VESECTORM1	R/W	0x4005_014C
a-相零电流 AD 转换结果(注 4)	VEIAO1	R/W	0x4005_0150
b-相零电流 AD 转换结果(注 4)	VEIBO1	R/W	0x4005_0154
c-相零电流 AD 转换结果(注 4)	VEICO1	R/W	0x4005_0158
a-相电流 AD 转换结果(注 4)	VEIAADC1	R/W	0x4005_015C
b-相电流 AD 转换结果(注 4)	VEIBADC1	R/W	0x4005_0160
c-相电流 AD 转换结果(注 4)	VEICADC1	R/W	0x4005_0164
DC 直流电源电压(电压 [V] ÷ 最大电压(注 3) × 2^{15})	VEVDC1	R/W	0x4005_0168
d-轴电流(电流 [A] ÷ 最大电流(注 2) × 2^{31})	VEID1	R/W	0x4005_016C
q-轴电流(电流 [A] ÷ 最大电流(注 2) × 2^{31})	VEIQ1	R/W	0x4005_0170
PMD 控制：CMPU 设置	VECMPU1	R/W	0x4005_019C
PMD 控制：CMPV 设置	VECMPV1	R/W	0x4005_01A0
PMD 控制：CMPW 设置	VECMPW1	R/W	0x4005_01A4
PMD 控制：输出控制(MDOUT)	VEOUTCR1	R/W	0x4005_01A8
PMD 控制：TRGCMP0 设置	VETRGCMP01	R/W	0x4005_01AC
PMD 控制：TRGCMP1 设置	VETRGCMP11	R/W	0x4005_01B0
PMD 控制：触发器选择	VETRGSEL1	R/W	0x4005_01B4
PMD 控制：EMG 返回	VEEMGRS1	W	0x4005_01B8

注 1：最大速度可控制或可操纵的最大转速 [Hz]。

注 2：最大电流：(与 AD 转换器的 1 个 LSB 最低有效位对应的相电流值 [A])× 2^{11}

注 3：最大电压：(与 AD 转换器的 1 个 LSB 最低有效位对应的电源电压(VDC)值 [V])× 2^{12}

注 4：AD 转换结果将保存在每一个 16-位寄存器的上位 12 位中。

12.3.2 VE 控制寄存器

12.3.2.1 VEEN(VE 启用/禁用寄存器)

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

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	VEIDLEN	R/W	对是否在 IDLE 闲置模式下向矢量引擎提供时钟进行控制。 0：非激活 1：激活
0	VEEN	R/W	禁用或启用矢量引擎。 0：禁用 1：启用

12.3.2.2 VECPURUNTRG(CPU 启动触发器选择寄存器)

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

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	VCPURTB	W	通过编程启动通道 1。 0：- 1：开始
0	VCPURTA	W	通过编程启动通道 0。 0：- 1：开始

注 1：将"1"写入这些位时，将在下一周期清除。这些位将始终改为 0。

注 2：待完成的任务将通过 VEACTSCH 寄存器和 VETASKAPP 寄存器的设置确定。

注 3：正在执行的通道重启时，必须在执行启动指令之前由 VECOMPEND 寄存器终止该通道。

12.3.2.3 VETASKAPP(任务选择寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-4	VTASKB[3:0]	R/W	通道 1 任务选择 0x0：输出控制 0x1：触发器生成 0x2：输入处理 0x3：输入相位转换 0x4：输入坐标轴转换 0x5：电流控制 0x6：SIN/COS 计算 0x7：输出坐标轴转换 0x8：输出相位转换 0x9 -0xF：保留 明确规定通过编程启动通道 1 时将要完成的任务。
3-0	VTASKA[3:0]	R/W	通道 0 任务选择 0x0：输出控制 0x1：触发器生成 0x2：输入处理 0x3：输入相位转换 0x4：输入坐标轴转换 0x5：电流控制 0x6：SIN/COS 计算 0x7：输出坐标轴转换 0x8：输出相位转换 0x9 -0xF：保留 指定通过编程启动通道 0 时将要完成的任务。

注：只可规定计划表中所包含的任务。

12.3.2.4 VEACTSCH(操作计划表选择寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-4	VACTB[3:0]	R/W	为通道 1 指定单个任务执行或计划表。 0x0：单个任务执行 0x1：计划表 1 0x4：计划表 4 0x9：计划表 9 其它：保留
3-0	VACTA[3:0]	R/W	为通道 0 指定单个任务执行或计划表。 0x0：单个任务执行 0x1：计划表 1 0x4：计划表 4 0x9：计划表 9 其它：保留

12.3.2.5 VEREPTIME(计划表重复计数)

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

位	比特符号	型号	功能
31-8	–	R	读作 0。
7-4	VREPB[3:0]	R/W	指定在通道 1 中执行一项计划的重复次数。 0：不执行计划 1~15：按规定次数重复执行计划。
3-0	VREPA[3:0]	R/W	指定在通道 0 中执行一项计划的重复次数。 0 不执行计划 1~15： 按规定次数重复执行计划。

注：设定"0"时，不执行任何计划。

12.3.2.6 VETRGMODE(启动触发模式)

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

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-2	VTRGB[1:0]	R/W	指定通道 1 中触发输入处理的 AD 转换端中断。 通道 1 触发模式 00：同时忽略 INTB0(A 单元)和 INTB1(B 单元) 01：通过 INTB0(A 单元)启动 10：通过 INTB1(B 单元)启动 11：当 INTB0(A 单元)和 INTB1(B 单元)都出现时启动
1-0	VTRGA[1:0]	R/W	指定通道 0 中触发输入处理的 AD 转换端中断。 通道 0 触发模式 00：同时忽略 INTA0(A 单元)和 INTA1(B 单元) 01：通过 INTA0(A 单元)启动 10：通过 INTA1(B 单元)启动 11：当 INTA0(A 单元)和 INTA1(B 单元)都出现时启动

12.3.2.7 VEERRINTEN(错误中断启用/禁用)

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

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	VERRENB	R/W	启用或禁用通道 1 中的错误检测中断。 0：禁用 1：启用
0	VERRENA	R/W	启用或禁用通道 0 中的错误检测中断。 0：禁用 1：启用

12.3.2.8 VECOMPEND(VE 强行终止)

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

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	VCENDB	W	强行终止通道 1 中正在执行的计划。 0：− 1：终止
0	VCENDA	W	强行终止通道 0 中正在执行的计划。 0：− 1：终止

注： 将"1"写入这些位时，将在下一周期清除。这些位始终读为"0"。

12.3.2.9 VEERRDET(错误检测)

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

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	VERRDB	R	通道 1 错误标志 0：未检测到错误 1：检测到错误
0	VERRDA	R	通道 0 错误标志 0：未检测到错误 1：检测到错误

注 1：一项计划执行期间(等待启动触发的待命时间除外)检测到 PWM 中断时设置。

注 2：通过读取该寄存器清除错误标志。

12.3.2.10 VESCHTASKRUN(计划执行标志/执行任务)

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

位	比特符号	型号	功能
31-10	−	R	读作 0。
9-6	VRTASKB[3:0]	R	通道 1 中当前正在执行的任务号 0x0：输出控制 0x1：触发器生成 0x2：输入处理 0x3：输入相位转换 0x4：输入坐标轴转换 0x5：电流控制 0x6：SIN/COS 计算 0x7：输出坐标轴转换 0x8：输出相位转换 0x9 ~ 0xF：保留
5	VRSCHB	R	通道 1 中的计划执行状态 0：不执行 1：执行
4-1	VRTASKA[3:0]	R	通道 0 中当前正在执行的任务号 0x0：输出控制 0x1：触发器生成 0x2：输入处理 0x3：输入相位转换 0x4：输入坐标轴转换 0x5：电流控制 0x6：SIN/COS 计算 0x7：输出坐标轴转换 0x8：输出相位转换 0x9 ~ 0xF：保留
0	VRSCHA	R	通道 0 中的计划执行状态 0：不执行 1：执行

12.3.2.11 VETMPREG0(临时寄存器 0)

	31	30	29	28	27	26	25	24
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG0							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-0	TMPREG0[31:0]	R/W	临时寄存器 0

12.3.2.12 VETMPREG1(临时寄存器 1)

	31	30	29	28	27	26	25	24
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG1							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-0	TMPREG1[31:0]	R/W	临时寄存器 1

12.3.2.13 VETMPREG2(临时寄存器 2)

	31	30	29	28	27	26	25	24
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG2							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-0	TMPREG2[31:0]	R/W	临时寄存器 2

12.3.2.14 VETMPREG3(临时寄存器 3)

	31	30	29	28	27	26	25	24
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG3							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-0	TMPREG3[31:0]	R/W	临时寄存器 3

12.3.2.15 VETMPREG4(临时寄存器 4)

	31	30	29	28	27	26	25	24
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG4							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-0	TMPREG4[31:0]	R/W	临时寄存器 4

12.3.2.16 VETMPREG5(临时寄存器 5)

	31	30	29	28	27	26	25	24
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	TMPREG5							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-0	TMPREG5[31:0]	R/W	临时寄存器 5

12.3.3 公用寄存器

12.3.3.1 VETADC(公用 ADC 转换时间)

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

位	比特符号	型号	功能
31-16	−	R	读作 "0"。
15-0	TADC[15:0]	R/W	ADC 转换时间(基于 PWM 时钟) 0x0000 ~ 0xFFFF：(ADC 模数转换时间[s] ÷ PWM 计数器时钟频率[s]) 注)选择 1-分流器电流检测模式并启用 PWM 移位时该寄存器生效。

12.3.4 特定通道寄存器(x = 0 ~ 1)

12.3.4.1 VEMODEx(任务控制模式寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-4	−	R/W	始终写为"0"。
3-2	OCRMD[1:0]	R/W	输出控制 00：输出 OFF 01：输出允许 10：保留 11：输出 OFF 与 EMG 返回
1	ZIEN	R/W	零电流检测 0：禁用 1：启用
0	PVIEN	R/W	相位插值 0：禁用 1：启用

12.3.4.2 VEFMODEx(流量控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	MREGDIS	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ADCSEL		-	PMDSEL	IDMODE		SPWMEN	C2PEN
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-10	−	R/W	始终写为"0"。
9	MREGDIS	R/W	保持 SIN/COS/SECTOR 之前的数值 0：有效 1：无效 如果无效，VESINMx = VESINx, VECOSMx = VECOSx, VESECTORMx = VESECTORx。
8	−	R/W	始终写为"0"。
7-6	ADCSEL[1:0]	R/W	选择准备使用的 ADC 转换单元 00：A 单元 01：B 单元 10：A 单元，B 11：A 单元，B 通过准备使用的矢量引擎通道，其设置如下： VE: 通道 0 → ADC 转换单元: A 单元或 A 单元/B 单元 VE: 通道 1 → ADC 转换单元: B 单元或 A 单元/B 单元
5	−	R/W	始终写为"0"。
4	PMDSEL	R/W	选择 PMD 通道 0：通道 0 1：通道 1 通过准备使用的矢量引擎通道，其设置如下。 VE: 通道 0 → PMD 单元: 通道 0 VE: 通道 1 → PMD 单元: 通道 1
3-2	IDMODE	R/W	电流检测模式 00：3-分流器 01：2-传感器 10：1-分流器(用于递增计数 PMDTRG) 11：1-分流器(用于递减计数 PMDTRG)
1	SPWMEN	R/W	允许或禁止 PWM 移位 0：禁用 1：启用
0	C2PEN	R/W	选择 3-相或 2-相调制。 0：3-相调制 1：2-相调制

注：使用 1-分流器模式时，可采用下列 PMDTRG：

VEFMODE0 <IDMODE[1:0]>	VEFMODE1 <IDMODE[1:0]>	PMD0TRGCR <TRG0MD[2:0]>	PMD0TRGCR <TRG1MD[2:0]>	PMD1TRGCR <TRG0MD[2:0]>	PMD1TRGCR <TRG1MD[2:0]>
10	-	010(递增计数)	010(递增计数)	-	-
10	-	101(载波底点)	010(递增计数)	-	-
11	-	001(递减计数)	001(递减计数)	-	-
11	-	001(递减计数)	101(载波底点)	-	-
-	10	-	-	010(递增计数)	010(递增计数)
-	10	-	-	101(载波底点)	010(递增计数)
-	11	-	-	001 (递减计数)	001 (递减计数)
-	11	-	-	001 (递减计数)	101(载波底点)

12.3.4.3 VETPWMx(PWM 周期率控制寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	TPWM[15:0]	R/W	设置 PWM 周期率(允许相位插值时有效，16-位固定点数据：0.0 ~ 1.0)如下： 0x0000 ~ 0xFFFF：PWM 周期 [s] × Max _ Hz × 2^{16} (Max _ Hz：最大转速[Hz]) (指 PWM 频率与最大转速之间的比率)

12.3.4.4 VEOMEGAx(转速控制寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	OMEGA[15:0]	R/W	设置转速(16-位固定点数据：-1.0 ~ 1.0)如下： 0x0000 ~ 0xFFFF :转速 [Hz] ÷ Max _ Hz × 2^{15} (Max_Hz : 最大转速[Hz])

12.3.4.5 VETHETAx(电动机相位控制寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	THETA[15:0]	R/W	设置相位数据(16-位固定点数据： 0.0 ~ 1.0)如下： 公式：相位[度] ÷ 360 × 2^{16}

12.3.4.6 VECOSx/VESINx/VECOSMx/VESINMx

VECOSx(THETA 时输出转换余弦值(Q15 数据))

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	COS[15:0]	R/W	基于 THETA 值的余弦值(16-位固定点数据：-1.0 ~ 1.0) 余弦值：0x0000~ 0xFFFF

VESINx(THETA 时输出转换正弦值(Q15 数据))

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	SIN[15:0]	R/W	基于 THETA 时间值的正弦值(16-位固定点参数： -1.0 ~ 1.0) 正弦值：0x0000 ~ 0xFFFF

VECOSMx(输入处理以前的余弦值(Q15 数据))

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	COSM[15:0]	R/W	COS 寄存器之前的数值 余弦值(之前的数值)： 0x0000 ~ 0xFFFF

VESINMx(输入处理以前的正弦值(Q15 数据))

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	SINM[15:0]	R/W	SIN 寄存器以前的数值 正弦值(之前的数值) 0x0000 ~ 0xFFFF

12.3.4.7 VEIDREFx/VEIQREFx(dq 电流基准寄存器)

VEIDREFx

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

位	比特符号	型号	功能
31-16	-	R	读作"0"。
15-0	IDREF[15:0]	R/W	d-轴电流参考值(16-位固定点数据：-1.0 ~ 1.0) 0x0000~0xFFFF(待设定的数值：d-轴电流参考值[A] ÷ Max _ I × 2^{15}) Max _ I：(与 ADC 一个 LSB 最低有效位对应的相电流值[A])×2^{11}

VEIQREFx

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

位	比特符号	型号	功能
31-16	-	R	读作"0"。
15-0	IQREF[15:0]	R/W	q-轴电流参考值(16-位固定点数据：-1.0 ~ 1.0) 0x0000 ~ 0xFFFF(待设定的数值：q-轴电流参考值[A] ÷ Max _ I × 2^{15}) Max_I:(与 ADC 一个 LSB 最低有效位对应的相电流值[A])×2^{11}

12.3.4.8 VEVDx/VEVQx(d-轴/q-轴电压寄存器)

VEVDx

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

位	比特符号	型号	功能
31-0	VD[31:0]	R/W	d-轴电压(32-位固定点数据：-1.0 ~ 1.0) 0x0000_0000 ~ 0xFFFF_FFFF(d-轴电压÷ Max _V×2^{31}) Max _ V：(与 ADC 一个 LSB 最低有效位对应的电源电压(VDC)值[V])×2^{12}

VEVQx

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

位	比特符号	型号	功能
31-0	VQ[31:0]	R/W	q-轴电压(32-位固定点数据：-1.0~ 1.0) 0x0000_0000 ~ 0xFFFF_FFFF(q-轴电压÷ Max _V×2^{31}) Max_V: (与 ADC 一个 LSB 最低有效位对应的电源电压(VDC)值[V])×2^{12}

12.3.4.9 VECIDKIx/VECIDKPx/VEVCIQKIx/VECIQKPx(PI 控制系数寄存器)

VECIDKIx

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

位	比特符号	型号	功能
31-16	-	R	读作"0"。
15-0	CIDKI[15:0]	R/W	d-轴 PI 控制整系数：0x0000 ~ 0xFFFF

VECIDKPx

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

位	比特符号	型号	功能
31-16	-	R	读作"0"。
15-0	CIDKP[15:0]	R/W	d-轴 PI 控制比例系数：0x0000 ~ 0xFFFF

VEVCIQKIx

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

位	比特符号	型号	功能
31-16	–	R	读作"0"。
15-0	CIQKI[15:0]	R/W	q-轴 PI 控制整系数：0x0000 ~ 0xFFFF

VECIQKPx

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

位	比特符号	型号	功能
31-16	–	R	读作"0"。
15-0	CIQKP[15:0]	R/W	q-轴 PI 控制比例系数：0x0000 ~ 0xFFFF

12.3.4.10 VEVDIHx/VEVDILHx/VEVQIHx/VEVQILHx(PI 控制积分项寄存器)

VEVDIHx

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

位	比特符号	型号	功能
31-0	VDIH[31:0]	R/W	d-轴 PI 控制积分项(VDI)上 32 位

VEVDILHx

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

位	比特符号	型号	功能
31-16	VDILH[15:0]	R/W	d-轴 PI 控制积分项(VDI)上 31 ~ 16 位
15-0	-	R	读作 0。

注：带 63 小数位的 64-位固定点数据(-1.0 ~ 1.0)

VEVQIHx

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

位	比特符号	型号	功能
31-0	VQIH[31:0]	R/W	q-轴 PI 控制积分项(VQI)上 32 位

VEVQILHx

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

位	比特符号	型号	功能
31-16	VQILH[15:0]	R/W	q-轴 PI 控制积分项(VQI)上 31~16 位
15-0	-	R	读作 0。

注： 带 63 小数位的 64-位固定点数据(-1.0 ~ 1.0)

12.3.4.11 VEMCTLFx(状态标志寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-6	−	R/W	始终写为"0"。
5	PLSLFM	R/W	<PLSLF>之前的数值
4	PLSLF	R/W	最小脉冲宽度差 最小脉冲宽度差 ≥ VEMINPLSx<MINPLS> 案例 = "0" 最小脉冲宽度差 < VEMINPLSx < MINPLS > 案例 = "1"
3	−	R/W	始终写为"0"。
2	LVTF	R/W	电源电压较低标志 VEVDCx<VDC> ≥ 0x0100 (1/128) 案例 = "0" VEVDCx<VDC> < 0x0100 (1/128) 案例 = "1"
1	LAVFM	R/W	之前的<LAVF>数值
0	LAVF	R/W	低速标志 0：高速 1：低速 VEOMEGAx<OMEGA> ≥ VEFPWMCHGx<FPWMCHG>案例 = "0" VEOMEGAx<OMEGA> < VEFPWMCHGx<FPWMCHG>案例 = "1"

12.3.4.12 VEFPWMCHGx(切换速度(对于 2-相调制与移位 PWM))

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	FPWMCHG[15:0]	R/W	启用 PWM 移位时的转速。 设置值为：转速[Hz] ÷ Max _ Hz × 2^{15} (Max_Hz :最大转速[Hz])

12.3.4.13 VEMDPRDx(PWM 周期控制寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	VMDPRD[15:0]	R/W	PWM 周期 设置 PMD 的 PMDxMDPRD 寄存器数值。

12.3.4.14 VEMINPLSx(最小脉冲宽度)

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	MINPLS[15:0]	R/W	设置 VECMPUx，VECMPVx 与 VECMPWx 负荷之间最小脉冲宽度差。 设置值为：脉冲宽度差[s] ÷ PWM 计数器时钟周期[s]

12.3.4.15 VESECTORx/VESECTORMx(扇区信息寄存器)

VESECTORx

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

位	比特符号	型号	功能
31-4	−	R	读作"0"。
3-0	SECTOR[3:0]	R/W	扇区信息 值：0x0 ~ 0xF 表示分 12 个扇区每个扇区 30 度输出时的转速。

VESECTORMx

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

位	比特符号	型号	功能
31-4	−	R	读作"0"。
3-0	SECTORM[3:0]	R/W	以前的扇区信息。 值：0x0 ~ 0xF 在输入处理时使用。

12.3.4.16 VEIAOx/VEIBOx/VEICOx(零电流寄存器)

VEIAOx

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	IAO[15:0]	R/W	a-相零电流 AD 转换结果。 (保存电机停机时 a-相电流的 AD 模数转换结果。)

VEIBOx

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	IBO[15:0]	R/W	b-相零电流 AD 模数转换结果。 (保存电机停机时 b-相电流的 AD 转换结果。)

VEICOx

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	ICO[15:0]	R/W	c-相零电流 AD 模数转换结果。 (保存电机停机时 c-相电流的 AD 转换结果)

注 1：启动零电流检测时，AD 转换结果将自动存入这些寄存器。

注 2：AD 模数转换结果将保存在 15-4 位，3-0 位始终为"0"。

12.3.4.17 VEIAADCx/VEIBADCx/VEICADCx(电流 AD 转换结果寄存器)

VEIAADCx

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	IAADC[15:0]	R/W	保存 a-相电流 AD 转换结果：0x0000 ~ 0xFFFF

VEIBADCx

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

位	比特符号	型号	功能
31-16	-	R	读作"0"。
15-0	IBADC[15:0]	R/W	保存 b-相电流 AD 转换结果：0x0000 ~ 0xFFFF

VEICADCx

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	ICADC[15:0]	R/W	保存 c-相电流 AD 模数转换结果：0x0000 ~ 0xFFFF

注： AD 模数转换结果将保存在 15-4 位，3-0 位始终为"0"。

12.3.4.18 VEVDCx(电源电压寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	VDC[15:0]	R/W	电源电压(16-位固定点数据：0 ~1.0) 值：0x0000 ~ 0xFFFF 实际电压值为：VDC 数值 × Max _ V 数值 ÷ 2^{15} Max_V: (与 ADC 一个 LSB 最低有效位对应的电源电压(VDC)值[V])×2^{12}

12.3.4.19 VEIDx/VEIQx(d-轴/q-轴电流寄存器)

VEIDx

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

位	比特符号	型号	功能
31-0	ID[31:0]	R/W	d-轴电流(32 位固定点数据： -1.0 ~ 1.0) d-轴电流：0x0000_0000 ~ 0xFFFF_FFFF 实际电流值为：ID 值× Max _ I 值 ÷ 2^{31} Max _ I：(与 ADC 一个 LSB 最低有效位对应的相电流值[A])×2^{11}

VEIQx

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

位	比特符号	型号	功能
31-0	IQ[31:0]	R/W	q-轴电流(32 位固定点数据： -1.0 ~ 1.0) q-轴电流：0x0000_0000 ~ 0xFFFF_FFFF 实际电流值为：IQ 值× Max _ I 值 ÷ 2^{31} Max_I：(与 ADC 一个 LSB 最低有效位对应的相电流值[A])×2^{11}

12.3.4.20 VECMPUx / VECMPVx/ VECMPWx(PWM 占空比寄存器)

VECMPUx

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	VCMPU[15:0]	R/W	U-相的 PWM 设置 U-相的 PWM 脉冲宽度：0x0000 ~ 0xFFFF

VECMPVx

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

位	比特符号	型号	功能
31-16	-	R	读作 0。
15-0	VCMPV[15:0]	R/W	V-相的 PWM 设置 V-相的 PWM 脉冲宽度：0x0000 ~ 0xFFFF

VECMPWx

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

位	比特符号	型号	功能
31-16	−	R	读作 0。
15-0	VCMPW[15:0]	R/W	W-相的 PWM 设置 W-相的 PWM 脉冲宽度：0x0000 ~ 0xFFFF

12.3.4.21 VEOUTCRx(6-相输出控制寄存器)

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

位	比特符号	型号	功能
31-9	−	R	读作"0"。
8	WPWM	R/W	W 相-PWM 0：ON/OFF 输出 1：PWM 输出
7	VPWM	R/W	V 相-PWM 0：ON/OFF 输出 1：PWM 输出
6	UPWM	R/W	U 相-PWM 0：ON/OFF 输出 1：PWM 输出
5-4	WOC[1:0]	R/W	W-相输出控制 00：WO OFF, ZO OFF 01：WO ON, ZO OFF 10：WO OFF, ZO ON 11：WO ON, ZO ON (注)<WPWM>=1 时，WO 与 ZO 均为"ON"。
3-2	VOC[1:0]	R/W	V-相输出控制 00：VO OFF, YO OFF 01：VO ON, YO OFF 10：VO OFF, YO ON 11：VO ON, YO ON (注)<VPWM>=1 时，VO 与 YO 均为"ON"。
1-0	UOC[1:0]	R/W	U-相输出控制 00：UO OFF, XO OFF 01：UO ON, XO OFF 10：UO OFF, XO ON 11：UO ON, XO ON (注)< UPWM>=1 时，UO 与 XO 均为"ON"。

PMD 的 U，V，W-相位输出控制如下表所示。(下表只列出 VE 矢量引擎中使用的组合信息。)

<UPWM>，<UOC> PMD 设置：U-相输出控制(UO，XO)

设置		输出	
<UPWM>	<UOC>	UO	XO
0	00	OFF 输出	OFF 输出
1	00	PWMU 反相输出	PWMU 输出
1	11	PWMU 输出	PWMU 反相输出

<VPWM>， <VOC> PMD 设置：V-相输出控制(VO，YO)

设置		输出	
<VPWM>	<VOC>	VO	YO
0	00	OFF 输出	OFF 输出
1	00	PWMV 反相输出	PWMV 输出
1	11	PWMV 输出	PWMV 反相输出

<WPWM>，<WOC>PMD 设置：W-相输出控制(WO，ZO)

设置		输出	
<WPWM>	<WOC>	WO	ZO
0	00	OFF 输出	OFF 输出
1	00	PWMW 反相输出	PWMW 输出
1	11	PWMW 输出	PWMW 反相输出

12.3.4.22 VETRGCRCx(同步触发校正值寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	TRGCRC[15:0]	R/W	用于校正同步触发时序。 设置值为：校正时间[s] ÷ PWM 计数器时钟频率[s]

12.3.4.23 VETRGCMP0x/VETRGCMP1x(触发时序设置寄存器)

VETRGCMP0x

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	VTRGCMP0							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VTRGCMP0							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	–	R	读作 0。
15-0	VTRGCMP0[15:0]	R/W	PMD 设置：规定与 PMD 同步的 ADC 取样触发时序。 0x0000：被禁止 0x0001 ~ (<MDPRD[15:0]> 值 – 1)：触发时序 <MDPRD[15:0]> 值 ~ 0xFFFF：被禁止

VETRGCMP1x

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	VTRGCMP1							
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	VTRGCMP1							
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-16	–	R	读作 0。
15-0	VTRGCMP1[15:0]	R/W	PMD 设置：规定与 PMD 同步的 ADC 取样触发时序。 0x0000：被禁止 0x0001 ~ (<MDPRD[15:0]> 值 – 1)： 触发时序 <MDPRD[15:0]> 值 ~ 0xFFFF: 被禁止

注 1：只有在选择下列 PMD 触发模式中的一种时，这些寄存器方可生效：计数递减匹配，计数递增匹配，计数递增/递减匹配。

注 2：将 PMD 触发输出模式设置为触发器选择输出(PMDxTRGMD < TRGOUT >= 1)时，这些寄存器将失效。

12.3.4.24 VETRGSELx(同步触发器选择寄存器)

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

位	比特符号	型号	功能
31-3	−	R	读作 0。
2-0	VTRGSEL[2:0]	R/W	PMD 设置：对在规定的时间要输出的同步触发器数作出规定 <VTRGCMP0[15:0]>。 0 ~ 5：输出触发器数 6 ~ 7：被禁止 注）这些寄存器有效，当 PMD 触发器输出模式设置为触发器选择输出时(PMDxTRGMD<TRGOUT>=1)。

12.3.4.25 VEEMGRSx(EMG 返回控制寄存器)

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

位	比特符号	型号	功能
31-1	−	R	读作"0"。
0	EMGRS	R/W	PMD 设置：EMG 返回命令，以便从 EMG 状态返回 0：空操作 1：EMG 返回命令

12.4 操作说明

12.4.1 进度管理

电机控制流程图如图 12-4 所示。矢量引擎按照经相关寄存器编程的进度和模式设置，进行状态转换。

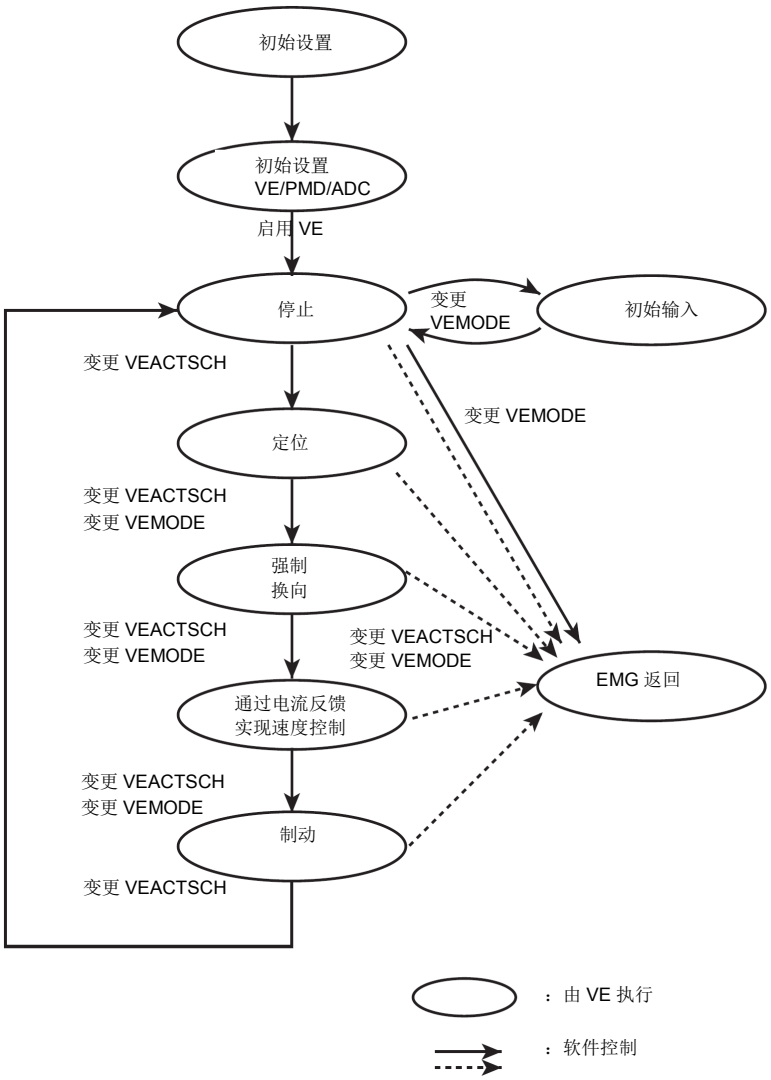


图 12-4 电机控制流程举例

RESET	：微控制器复位
初始设置	：由用户创建的程序进行初始设置
停止	：使电机停止。
初始输入	：当电机停止时，取样并储存零电流数据
定位	：确定电机初始位置。
强制换向	：启动电机。电机以规定的速度旋转规定的时间， 不受电流反馈控制。
实现速度控制通过 电流反馈	：用电流反馈控制电机旋转。
制动	：减速控制
EMG 返回	：从 EMG 状态返回。

12.4.1.1 进度控制

VEACTSCH 寄存器用于选择要执行的进度。

进度由处理输出相关的任务的输出进度和处理输入相关的任务的输入进度组成。在各进度中要执行的任务如表 12-1 所示。

VEMODE 寄存器用于启用或禁用相位插入，控制输出操作，并视情况而定，启用或禁用电机控制流程各步的零电流检测(见表 12-2)。

表 12-1 在各进度中应执行的任务

进度选择 VEACTSCH	输出进度						输入进度		
	电流控制	SIN/COS 计算	输出坐标轴转换	输出相位转换	输出控制	触发器生成	输入处理	输入相位转换	输入坐标轴转换
0：个别执行	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)	(注 1)
1：计划表 1	o	o (注 2)	o	o	o (注 3)	o	o(注 4)	o	o
4：计划表 4	-	o (注 2)	o	o	o(注 3)	o	o (注 4)	o	o
9：计划表 9	-	-	-	-	o (注 3)	o	o(注 4)	-	-

注 1：只有对各任务作出了规定时，它才会得到执行。

注 2：相位内插。

注 3：输出 OFF:<EMGRS>

注 4：由零电流检测切换的任务操作。

表 12-2 常见设置示例

寄存器设置	进度选择 VEACTSCH	任务说明 VETASKAPP	相位内插 VEMODE	输出控制 VEMODE	零电流检测 VEMODE
电机控制流程	<VACTn[3:0]>	<VTASKn[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
强制换向	1	5	1	01	0
实现速度控制通过电流反馈	1	5	1	01	0
制动	4	6	0	01	0
EMG 返回	9	0	x	11	0

输出进度用 VECPURUNTRG 命令开始执行。当所有输出相关的任务完成时，矢量引擎进入待机状态，等待输入相关的任务的启动触发器。此时，能执行其他通道的进度。

输入进度由启动触发器开始执行。当所有输入相关的任务完成时，矢量引擎向 CPU 生成中断，进入停止状态。然而，若进度的重复计数(VEREPTIME)设置为"2"或以上，则不会生成中断，直到进度执行了规定的次数。

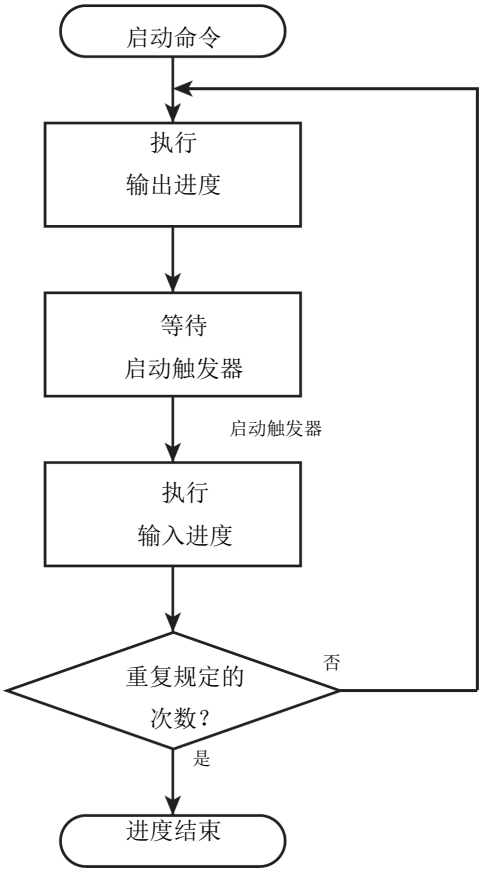


图 12-5 进度执行流程

12.4.1.2 启动控制

用 VEEN 寄存器启用矢量引擎。规定进度(VEACTSCH 寄存器)，要执行的任务(VETASKAPP 寄存器)和重复计数(VEREPTIME 寄存器)。

矢量引擎进度由输出进度和输入进度组成。矢量引擎通常先执行输出进度，进入待机状态，再由启动触发器开始执行输入进度。

- 输出进度启动：
  1. 用 VECPURUNTRG 命令启动。在这种情况下，执行 VETASKAPP 寄存器中规定的任务。
  2. 在相应的输入进度完成后重复启动(当 VEREPTIME ≥ 2 时)。

- 输入进度启动：
  1. 在相应的输出进度完成后由(VETRGMODE 寄存器中选择的)启动触发器启动。
  2. 用 VECPURUNTRG 命令启动。在这种情况下，执行 VETASKAPP 寄存器中规定的任务。

12.4.2 任务概要

在输出输入进度中执行的任务概要见表 12-3 。

当各任务被个别执行或者被规定为一个启动任务时，采用本表所示任务编号。

表 12-3 任务表

任务		任务说明	任务编号
输出进度	电流控制	控制 dq 电流	5
	SIN/COS 计算	进行正弦/余弦计算和相位内插。	6
	输出坐标轴转换	将 dq 坐标转换成 αβ 坐标。	7
	输出相位转换	将 2-相转换成 3-相。	8
	输出控制	将数据转换成 PMD 设置格式。 转换 PWM 移位。	0
	触发器生成	生成同步触发器时序。	1
输入进度	输入处理	获取 AD 转换结果， 并将它们转换成定点格式。	2
	输入相位转换	将 3-相转换成 2-相。	3
	输入坐标轴转换	将 αβ 坐标转换成 dq 坐标。	4

12.4.2.1 电流控制

电流控制单元由 d 轴 PI 控制单元和 q 轴 PI 控制单元组成，计算 d 轴和 q 轴电压。

1. d 轴电流 PI 控制

<等式>

$\Delta ID = VEIDREFx - <ID[31:0]>$ ：电流参考值和电流反馈之差

$VDIx = VECIDKIx \times \Delta ID + VDIx$ ：积分项计算

$VEVDx = VECIDKPx \times \Delta ID + VDIx$ ：用比例项进行电压计算

	寄存器名称	功能	
输入	VEIDx	d-轴电流	32-位定点数据（31 分位)
	VEIDREFx	d-轴电流参考值	16-位定点数据（15 分位)
	VECIDKPx	比例系数	16-位数据
	VECIDKIx	积分系数	16-位数据
输出	VEVDx	d-轴电压	32-位定点数据（31 分位)
内部	VDIx	d-轴电压积分项	64-位定点数据（63 分位)

2. q 轴电流 PI 控制

<等式>

$\Delta IQ = VEIQREFx - <IQ[31:0]>$ ：电流参考值和电流反馈之差

$VQIx = VECIQKIx \times \Delta IQ + VQIx$ ：积分项计算

$VEVQx = VECIQKPx \times \Delta IQ + VQIx$ ：用比例项进行电压计算

	寄存器名称	功能	
输入	VEIQx	q-轴电流	32- 位定点数据（31 分位)
	VEIQREFx	q-轴电流参考值	16- 位定点数据（15 分位)
	VECIQKPx	比例系数	16 -位数据
	VECIQKIx	积分系数	16 -位数据
输出	VEVQx	q-轴电压	32- 位定点数据（31 分位)
内部	VQIx	q-轴电压积分项	64- 位定点数据（63 分位)

12.4.2.2 SIN/COS 计算

SIN/COS 计算单元由相位内插单元和 SIN/COS 计算单元组成。

通过随 PWM 周期进行积分，相位内插计算转速。只有在相位内插启用时，它才会被执行。

1. 相位内插

<等式>

VETHETAx = VEOMEGAx × VETPWMx + VETHETAx ：转速积分。

仅当相位插入启用时。

	寄存器名称	功能	
输入	VETHETAx	θ 相	16-位定点数据 (0.0 ~1.0，16 分位)
	VEOMEGAx	旋转速度	16-位定点数据 (-1.0 ~ 1.0，15 分位)
	VETPWMx	PWM 周期率	16-位定点数据 (0.0 ~1.0，16 分位)
	VEMODEx	相位插入启用	模式设置
输出	VETHETAx	θ 相	16-位定点数据 (0.0 ~ 1.0，16 分位)

2. SIN/COS 计算

<等式>

VESINMx = VESINx ：(为了进行输入处理)保存先前的值。

VECOSMx = VECOSx ：(为了进行输入处理)保存先前的值。

VESINx = sin (VETHETAx × π) ：SIN/COS 计算

VECOSx = sin ((VETHETAx + 1/4) × π) ：SIN/COS 计算

	寄存器名称	功能	
输入	VETHETAx	θ 相	16-位定点数据 (0.0 ~ 1.0，16 分位)
输出	VESINx	在 θ 时的正弦值	16-位定点数据 (-1.0 ~ 1.0，15 分位)
	VECOSx	在 θ 时的余弦值	
	VESINMx	先前的正弦值	
	VECOSMx	先前的余弦值	

12.4.2.3 输出电压转换(坐标轴转换/相位转换)

输出电压转换由 dq ~ αβ 坐标轴转换和 2-相 ~ 3-相转换组成。

dq ~ αβ 坐标轴转换利用 SIN 和 COS 中的 Vd，Vq 计算 Vα，Vβ。

2-相 ~ 3-相转换用 Vα 和 Vβ 进行分频，并进行空间矢量转换，以计算 Va，Vb 和 Vc。

关于 2-相 ~ 3-相转换，可选择 2-相调制或 3-相调制。

1. dq ~ αβ 坐标转换

<等式>

VETMPREG3 = VECOSx × VEVDx − VESINx × VEVQx ：计算 Vα。

VETMPREG4 = VESINx × VEVDx + VECOSx × VEVQx ：计算 Vβ。

	寄存器名称	功能	
输入	VEVDx	d-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VEVQx	q-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VESINx	在 θ 时的正弦值	16-位定点数据 (-1.0 ~ 1.0，15 分位)
	VECOSx	在 θ 时的余弦值	16-位定点数据 (-1.0 ~ 1.0，15 分位)
输出	VETMPREG3	α-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG4	β-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)

2. 2-相 ~ 3-相转换(空间矢量转换)

a. 分频

<等式>

VESECTORMx = VESECTORx ：保存先前的矢量。

if(Vα ≥ 0 & Vβ ≥ 0)

  if(|Vα| ≥ |Vβ| ÷ SQR(3)

    if(|Vα| ÷ SQR(3) ≥ |Vβ|) <SECTOR[3:0]>=0

      else <SECTOR[3:0]>=1

    else <SECTOR[3:0]>=2

else if(Vα < 0 & Vβ ≥ 0)

    if(|Vα| < |Vβ| ÷SQR(3)) <SECTOR[3:0]>=3

    else if(|Vα| ÷SQR(3) < |Vβ|) <SECTOR[3:0]>=4

      else <SECTOR[3:0]>=5

else if(Vα < 0 & Vβ < 0)

    if(|Vα| ≥ |Vβ| ÷SQR(3))

      if(|Vα| ÷SQR(3) ≥ |Vβ|) <SECTOR[3:0]>=6

        else <SECTOR[3:0]>=7

  else <SECTOR[3:0]>=8

else if(Vα ≥ 0 & Vβ < 0)

    if(|Vα| < |Vβ| ÷SQR(3)) <SECTOR[3:0]>=9

    else if(|Vα| ÷SQR(3) < |Vβ|) <SECTOR[3:0]>=10

      else <SECTOR[3:0]>=11

	寄存器名称	功能	
输入	VETMPREG3	α-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG4	β-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
输出	VESECTORx	矢量	4-位数据
	VESECTORMx	先前的矢量	4-位数据

b. 3-相电压计算(当选择 3-相调制且<SECTOR[3:0]>= 0 时)

<等式>

t1 = ($\sqrt{3}$)／(VEVDC) × (($\sqrt{3}$)／2 × Vα – 1／2 × Vβ) ：计算 V1 周期。

t2 = ($\sqrt{3}$)／(VEVDC) × (Vβ) ：计算 V2 周期。

t3 = 1 − t1 − t2 ：计算 V0+V7 周期。

VETMPREG0 = t1 + t2 + t3 ÷ 2 ：计算 Va。

VETMPREG1 = t2 + t3 ÷ 2 ：计算 Vb。

VETMPREG2 = t3 ÷ 2 ：计算 Vc。

	寄存器 名称	功能	
输入	VETMPREG3	α-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG4	β-轴电压	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VEVDCx	电源电压	16-位定点数据(0.0 ~ 1.0，15 分位)
	VESECTORx	矢量	4-位数据
	VEFMODEx	调制方式	模式设置
输出	VETMPREG0	a-相电压	32-位定点数据 (0.0 ~ 1.0，31 分位)
	VETMPREG1	b-相电压	32-位定点数据 (0.0 ~ 1.0，31 分位)
	VETMPREG2	c-相电压	32-位定点数据 (0.0 ~ 1.0，31 分位)

12.4.2.4 输出控制

输出控制单元将 3-相电压值转换成 PWM 设置格式(VECMPUx，VECMPVx 和 VECMPWx)，并设置 VEOUTCRx 寄存器，以控制输出操作。

当选择 1-分流检测和 2-相调制，并启用 PWM 时，若转速慢于 PWM 移位开关参考值，则输出切换到移位 PWM 输出。

	寄存器名称	功能	
输入	VETMPREG0	a-相电压	32-位定点数据 (0.0 ~ 1.0，31 分位)
	VETMPREG1	b-相电压	32-位定点数据 (0.0 ~ 1.0，31 分位)
	VETMPREG2	c-相电压	32-位定点数据 (0.0 ~ 1.0，31 分位)
	VEMDPRDx	PWM 周期	16-位数据(PMD PWM 周期)
	VESECTORx	矢量	4-位数据
	VEOMEGAx	旋转速度	16-位定点数据 (-1.0 ~ 1.0，15 分位)
	VEFPWMCHGx	PWM 移位开关参考值	16-位定点数据 (-1.0 ~ 1.0，15 分位)
	VEMODEx	输出控制操作	模式设置
	VEFMODEx	PMD 通道/移位启用/ 调制方式/ 检测方式/	模式设置
输出	VECMPUx	PMD U-相 PMW 设置	16-位数据(0 ~ MDPRD 值)
	VECMPVx	PMD V-相 PWM 设置	16-位数据(0 ~ MDPRD 值)
	VECMPWx	PMD W-相 PWM 设置	16-位数据(0 ~ MDPRD 值)
	VEOUTCRx	PMD 输出控制设置	9-位设置
	VEEMGRSx	PMD EMG 返回	1-位设置
	VEMCTLFx	移位开关标志	状态

12.4.2.5 触发生成

视情况而定，触发生成单元按照电流检测方法利用 PWM 设置值(VECMPUx，VECMPV 和 VECMPW)计算触发时间，并设置 VETRGCMP0x 和 VETRGCMP1x 寄存器。

	寄存器 名称	功能	
输入	VECMPUx	PMD U-相 PWM 设置	16-位数据(0 ~ MDPRD 值)
	VECMPVx	PMD V-相 PWM 设置	16-位数据(0 ~ MDPRD 值)
	VECMPWx	PMD W-相 PWM 设置	16-位数据(0 ~ MDPRD 值)
	VEMDPRDx	PWM 周期设置	16-位数据(PMD PWM 周期)
	VETADC	AD 转换时间	16-位数据(0 ~ MDPRD 值)
	VETRGCRCx	触发器校正值	16-位数据(0 ~ MDPRD 值)
	VESECTORx	矢量	4-位数据
	VEMODEx	输出控制操作	模式设置
	VEFMODEx	PMD 通道/移位启用/ 调制方式/ 检测方式	模式设置
	VEMCTLFx	移位开关标志	状态
输出	VETRGCMP0	PMD 触发器 0 时序	16-位数据(0 ~ MDPRD 值)
	VETRGCMP1	PMD 触发器 1 时序	16-位数据(0 ~ MDPRD 值)
	VETRGSELx	PMD 触发器的选择	3-位数据

12.4.2.6 输入处理

输入处理单元保存分频的 3-相电流转换结果，并将电流和电压转换结果转换成定点数据。它将零电流转换结果保存在初始输入处理中。

	寄存器名称	功能	
输入	VEADREG0A	ADC 单元 A 的转换结果 0	16-位数据(采用 12 高位。)
	VEADREG1A	ADC 单元 A 的转换结果 1	
	VEADREG2A	ADC 单元 A 的转换结果 2	
	VEADREG3A	ADC 单元 A 的转换结果 3	
	VEADREG0B	ADC 单元 B 的转换结果 0	16-位数据(采用 12 高位。)
	VEADREG1B	ADC 单元 B 的转换结果 1	
	VEADREG2B	ADC 单元 B 的转换结果 2	
	VEADREG3B	ADC 单元 B 的转换结果 3	
	VEPHNUM0A	ADREG0A 检测到的相位信息	2-位数据
	VEPHNUM1A	ADREG1A 检测到的相位信息	
	VEPHNUM2A	ADREG2A 检测到的相位信息	
	VEPHNUM3A	ADREG3A 检测到的相位信息	
	VEPHNUM0B	ADREG0B 检测到的相位信息	2-位数据
	VEPHNUM1B	ADREG1B 检测到的相位信息	
	VEPHNUM2B	ADREG2B 检测到的相位信息	
	VEPHNUM3B	ADREG3B 检测到的相位信息	
	VESECTORMx	扇区信息	4 位数据
	VEMODEx	零电流检测	模式设置
	VEFMODEx	PMD 通道/ 电流检测方式/ ADC 单元/移位启用	模式设置
	VEMCTLFx	移位开关标志	状态
输出	VEVDCx	电源电压	16-位定点数据 (0.0 ~ 1.0，15 分位)
	VETMPREG0	a-相电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG1	b-相电流	
	VETMPREG2	c-相电流	
内部	VEIAOx	a-相零电流转换结果	16-位数据(采用 12 高位。)
	VEIBOx	b-相零电流转换结果	
	VEICOx	c-相零电流转换结果	
	VEIAADCx	a-相电流转换结果	16-位数据(采用 12 高位。)
	VEIBADCx	b-相电流转换结果	
	VEICADCx	c-相电流转换结果	

12.4.2.7 输入电流转换(相位转换/坐标轴转换)

输入电流转换由 3-相 ~2-相转换和 αβ~dq 坐标轴转换组成。

3-相 ~2-相转换利用 la，lb 和 lc 计算 Iα 和 Iβ。

αβ~dq 坐标轴转换利用得自 Iα，Iβ，VESINM 和 VECOSM 的 Id，Iq 计算 ld 和 lq。

1. 3-相 ~2-相转换

<等式>

VETMPREG3 = VETMPREG0 ：计算 Iα。

VETMPREG4 = 1 ÷ SQR(3) × VETMPREG1 − 1 ÷ SQR(3) × VETMPREG2 ：计算 Iβ。

	寄存器名称	功能	
输入	VETMPREG0	a-相电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG1	b-相电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG2	c-相电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
输出	VETMPREG3	α-轴电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG4	β-轴电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)

2. αβ~dq 坐标转换

<等式>

VEIDx = VECOSMx × VETMPREG3 + VESINMx × VETMPREG4 ：计算 Id。

VEIQx =− VESINMx × VETMPREG3 + VECOSMx × VETMPREG4 ：计算 Iq。

	寄存器名称	功能	
输入	VETMPREG3	α-轴电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VETMPREG4	β-轴电流	
	VESINMx	在 θ 时的正弦值	16-位定点数据 (-1.0 ~ 1.0，15 分位)
	VECOSMx	在 θ 时的余弦值	
输出	VEIDx	d-轴电流	32-位定点数据 (-1.0 ~ 1.0，31 分位)
	VEIQx	q-轴电流	

12.5 VE 通道，ADC 单元和 PMD 通道的组合

通过使用矢量引擎通道，可使用的 PMD 和 ADC 的组合有限制。另外电流检测选择和 ADC 单元选择所使用的组合会变化。

表 12-4 VE 和 PMD 的组合

矢量引擎	PMD
通道 0	通道 0
通道 1	通道 1

表 12-5 VE 和 ADC 的组合

矢量引擎			ADC 单位 A				ADC 单位 B			
通道	VEFMODE (注 2)		ADREG0	ADREG1	ADREG2	ADREG3	ADREG0	ADREG1	ADREG2	ADREG3
	电流检测 <IDMODE [1:0]>	ADC 选择 <ADCSEL [1:0]>								
0	0x	00	电流数据 1	电流数据 2	注 1	VDC 数据	-	-	-	-
		1x	电流数据 1	-	注 1	VDC 数据	电流数据 2	-	-	-
	1x	00	电流数据 1	电流数据 2	-	VDC 数据	-	-	-	-
1	0x	01	-	-	-	-	电流数据 1	电流数据 2	注 1	VDC 数据
		1x	-	电流数据 2	-	-	-	电流数据 1	注 1	VDC 数据
	1x	01	-	-	-	-	电流数据 1	电流数据 2	-	VDC 数据

注 1：必须规定寄存器的相位信息。然而，其寄存器的 AD 转换结果不用于计算。

注 2：不要使用本表所禁止的 VE 和 ADC 的组合。

13. 编码器输入电路 (ENC)

13.1 概述

编码器输入电路支持四种运行模式，包括编码器模式，传感器模式(两类)和定时器模式。

功能如下：

- 支持增量编码器和霍尔传感器 ICs。(霍尔传感器 IC 信号能直接输入)
- 24-位通用定时器模式
- 乘-4 (乘- 6)电路
- 转向检测电路
- 24 -位计数器
- 比较器启用/禁用
- 中断请求输出: 1
- 输入信号数字噪声滤波器

13.2 通道之间的差异

TMPM370FYDFG/FYFG 具有一个双通道增量编码器接口(ENC0 和 ENC1)，该接口能根据增量编码器的输入信号，获得电机的绝对位置。

除下列差异外，这些通道相同。

表 13-1 通道之间的差异

通道	输入引脚			编码器输入中断
	A-相	B-相	Z-相	
通道 0	PD0 / ENCA0	PD1 / ENCB0	PD2 / ENCZ0	INTENC0
通道 1	PF2 / ENCA1	PF3 / ENCB1	PF4 / ENCZ1	INTENC1

13.3 方块图

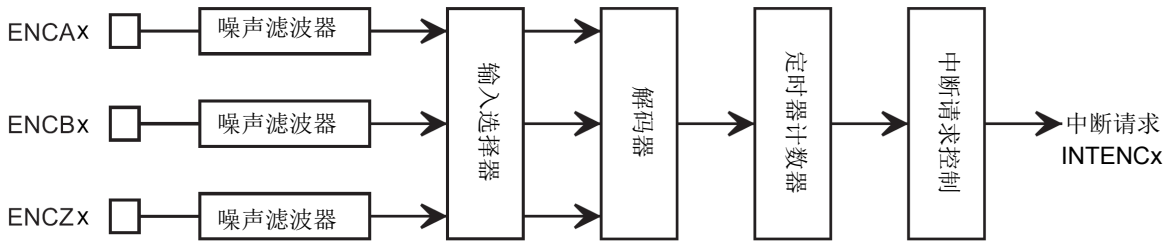


图 13-1 编码器输入电路方块图

13.4 寄存器

13.4.1 寄存器列表

控制寄存器及编码器输入电路的地置如下。

通道 x	基址
通道 0	0x4001_0400
通道 1	0x4001_0500

寄存器名称 (x=0,1)		地址(基本+)
编码器输入控制寄存器	ENxTNCR	0x0000
编码器计数器重新加载寄存器	ENxRELOAD	0x0004
编码器比较寄存器	ENxINT	0x0008
编码器计数器	ENxCNT	0x000C

13.4.2 ENxTNCR(编码器输入控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	MODE		P3EN
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	ZEN	ENRUN	NR		INTEN	ENDEV		
复位后	0	0	0	0	0	0	0	0

位	比特符号	型号	功能
31-19	−	R	读作"0"。
18-17	MODE[1:0]	R/W	编码器输入模式设置 00：编码器模式 01：传感器模式(事件计数) 10：传感器模式 (定时器计数) 11：定时器模式
16	P3EN	R/W	2-相/ 3-相输入选择(传感器模式) (注 1) 0：2-相输入 1：3-相输入 设置输入信号数。
15	CMP	R	比较标志 0：- 1：比较(由 RD 清除) 若执行比较，<CMP>设置为"1"。 通过读取数值，就可清除标志。当设置<ENRUN> = "0"时，应始终设置"0"。写入该位会无效。
14	REVERR	R	反向错误标志(传感器模式 (在定时器计数时)) (注 2) 0：- 1：错误(由 RD 清除) 在传感器模式下(在定时器计数时)，当发生反向错误时，<REVERR>设置为"1"。 通过读取数值，就可清除标志。当设置<ENRUN> = "0"时，应始终设置"0"。 写入该位会无效。 在编码器模式，传感器模式(事件计数)和定时器模式时，该位无意义。
13	UD	R	旋转方向 0：CCW (用增量编码器，A 相相对于 B-相具有 90 度相位超前) 1：CW (用增量编码器，A 相相对于 B-相具有 90 度相位滞后) 当<ENRUN> = "0"时，<UD>设置为"0"。
12	ZDET	R	Z-已检测 0：不选 1：Z-相已检测 在<ENRUN>从 0 写为 1 后，<ZDET>在 Z 输入信号(ENCZ)的第一沿设置为 1。该设置在 CW 旋转时发生在信号 Z 的上升沿或者在 CCW 旋转时在 Z 的下降沿。 当<ENRUN> = "0"时，<ZDET>设置为"0"。 <ZEN>对<ZDET>值无影响。 在传感器事件计数和传感器定时器计数模式时，<ZDET>设置为"0"。 在传感器模式(事件计数)和传感器模式(定时器计数)时，该位始终设置为"0"。
11	SFTCAP	W	执行软件捕捉(定时器模式/传感器模式(在定时器计数时)) 0：- 1：软件捕捉 若<SFTCAP>设置为 1，编码器计数器的值被捕捉到 ENxCNT 寄存器中。 将"0"写入<SFTCAP>会无效。读取<SFTCAP>会始终返回到"0"。 在编码器和传感器事件计数模式时，<SFTCAP>无效；将"1"写入该位会被忽略。
10	ENCLR	W	编码器脉冲计数器的清除 0：- 1：清除 将 1 写入<ENCLR>会将编码器计数器清除为"0"。一旦被清除，编码器计数器从"0"开始重新计数。 将"0"写入<ENCLR>会无效。读取<ENCLR>会始终返回到"0"。
9	ZESEL	R/W	ENCZ 边沿选择(定时器模式) 0：上升沿 1：下降沿 在定时器模式时，该位选择用作外触发器的 ENCZ 的输入边沿。 在其他模式时，该位无意义。
8	CMPEN	R/W	比较启用 0：禁用 1：启用 当"1"设置为<CMPEN>，该位比较编码器计数器的计数器值和 ENINT 的寄存器值。当"0"设置为<CMPEN>时，该比较禁用。

位	比特符号	型号	功能
7	ZEN	R/W	Z 相启用(编码器模式/定时器模式) 0：禁用 1：启用 在其他模式时，该位无意义。 | <编码器模式> 用 ENCZ 输入清除编码器计数器的设置 | 当设置<ZEN> = "1"时，若在顺时针旋转时检测到 ENCZ 的上升沿，则编码器计数器被清除为"0"。 若在逆时针旋转时检测到 ENCZ 的下降沿，则编码器计数器被清除为"0"。 若 ENCLK(源于解码的 A 和 B 信号的乘 4 时钟)和 ENCZ 边沿一致，则编码器计数器在无增量或减量的情况下被清除为"0"(即清除优先)。 | | <定时器模式> 设置 ENCZ 输入，以用作外触发器。 | 当<ZEN> = 1 时，编码器计数器值被捕捉到 EN0INT 寄存器中，并在<ZESEL>所选的 ENCZ 的边沿被清除为"0"。 |
6	ENRUN	R/W	编码器操作启用 0：禁用 1：启用 将<ENRUN>设置为 1 及将<ZDET>清除为"0"会启用编码器操作。 将<ENRUN> 清除为"0"会禁用编码器操作。 当<ENRUN>位被清除为 "0"时，有被清除的和未被清除的计数器和标志。
5-4	NR[1:0]	R/W	噪声滤波器 00：无滤波 01：滤除比 31 / fsys 窄的噪声脉冲(387.5 ns @ 80 MHz) 10：滤除比 63 / fsys 窄的噪声脉冲(787.5 ns @ 80 MHz) 11：滤除比 127 / fsys 窄的噪声脉冲(1587 ns @ 80 MHz) 数字噪声滤波器除去比<NR[1:0]>所选宽度更窄的脉冲。
3	INTEN	R/W	编码器中断启用 0：禁用 1：启用 <INTEN>启用或禁用 ENC 中断。 将<INTEN>设置为"1"会启用中断生成。将<INTEN>设置为"0"会禁用中断生成。
2-0	ENDEV[2:0]	R/W	编码器脉冲分频系数 000：除以 1　　100：除以 16 001：除以 2　　101：除以 32 010：除以 4　　110：除以 64 011：除以 8　　111：除以 128 设置编码器脉冲分频系数 编码器的频率除以<ENDEV[2:0]>规定的系数。分频的信号确定事件中断的间隔。

注 1：在编码器模式或定时器模式，<P3EN>必须设置为"0"。

注 2：若要改变模式，先应读取标志，以进行清除。

操作模式有<MODE[1:0]>，<P3EN>和<ZEN>规定的 8 种模式。

操作模式的设置如下：

<MODE[1:0]>	<ZEN>	<P3EN>	输入引脚	模式
00	0	0	A，B	编码器模式
	1		A,B,Z	编码器模式(Z 的使用)
01	0	0	U,V	传感器模式(事件计数，2-相输入)
		1	U,V,W	传感器模式(事件计数，3-相输入)
10	0	0	U,V	传感器模式 (定时器计数，2-相输入)
		1	U,V,W	传感器模式 (定时器计数，3-相输入)
11	0	0	-	定时器模式
	1		Z	定时器模式(Z 的使用)

<ENRUN>的状态及对应的信号如下。

计数器/标志	<ENRUN> = 0 (复位后)	<ENRUN> = 1 (操作)	<ENRUN> = 0 (停止)	<ENRUN> = 0 目标标志/计数器清除 程序
编码器计数器	0x000000	计数操作	当停止时保持数值	软件清除 (<ENCLR> = 1 WR)
噪声滤波器 计数器	0y0000000	向上计数操作	向上计数操作 (始终滤波)	仅复位
编码器脉冲 分频计数器	0x00	倒计数操作	停止并被清除	当<ENRUN> = "0"时清除
比较标志 <CMP>	0	当读取并比较清除时， 设置"1"。	清除	当<ENRUN> = "0"时清除
反向错误标志 <REVERR>	0	当出错时，设置"1"。 当读取时清除	清除	当<ENRUN> = "0"时清除
Z 检测标志 <ZDET>	0	当检测到 Z 时，设置"1"。	清除	当<ENRUN> = "0"时清除
转向位 <UD>	0	按照方向设置"0" / "1"	清除	当<ENRUN> = "0"时清除

13.4.3 ENxRELOAD(编码器计数器重新加载寄存器)

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

位	比特符号	型号	功能
31-16	−	R	读作"0"。
15-0	RELOAD[15:0]	R/W	(在乘以 4 或 6 后)设置编码器计数器周期 0x0000 ~ 0xFFFF 使用 Z-相：设置一次旋转的计数脉冲数 未使用 Z-相：设置一次旋转的计数脉冲数减 1 <RELOAD[15:0]>确定编码器计数器周期乘以 4。 若编码器计数器被配置为一个向上计数器，则它递增至<RELOAD[15:0]>中编程的数值，然后在下一 ENCLK 时折回至"0"。若编码器计数器被配置为倒计数器，则它递减到"0"，然后在下一 ENCLK 时，重新加载<RELOAD[15:0]> 值。

RELOAD 寄存器仅在编码器模式时使用。

13.4.4 ENxINT(编码器比较寄存器)

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

位	比特符号	型号	功能		
31-24	−	R	读作"0"。		
23-0	INT[23:0]	R/W	计数器比较值设置		
			编码器模式：	编码器脉冲位置的中断条件。	0x0000 ~ 0xFFFF
				当设置<CMPEN> = "1"，若编码器计数器值与<INT[15:0]>值匹配，则<CMP>设置为"1"。若设置<INTEN> = "1"，则发生中断请求(INTENC0)。 然而，若设置<ZEN> = "1"，则不会发生中断请求，直到 <ZDET> = "1"。	
			传感器模式： (事件计数)	编码器脉冲位置的中断条件。	0x0000 ~ 0xFFFF
				当设置<CMPEN> = "1"，若编码器计数器值与<INT[15:0]>值匹配，则<CMP>设置为"1"。若设置<INTEN> = "1"，则发生中断请求(INTENC0)。 该位对<ZEN>值无影响。	
			传感器模式： (定时器计数)	异常脉冲检测时间的中断条件	0x000000 ~ 0xFFFFFF
				当设置<CMPEN> = "1"时，内计数器值与<INT[23:0]>值匹配，异常脉冲检测时间错误得到确定，并且<CMP>设置为"1"。若设置<INTEN> = "1"，则发生中断请求(INTENC0)。 该位对<ZEN>值无影响。	
			定时器模式	定时器比较的中断条件	0x000000 ~ 0xFFFFFF
				当设置<CMPEN> = "1"时，内计数器值与<INT[23:0]>值匹配，异常脉冲检测时间错误得到确定，并且<CMP>设置为"1"。若设置<INTEN> = "1"，则发生中断请求(INTENC0)。 该位对<ZEN>值无影响。	

<INT[23:16]>仅在传感器模式(定时器计数)和定时器模式时使用。

13.4.5 ENxCNT (编码器计数器)

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

位	比特符号	型号	功能		
31-24	–	R	读作"0"。		
23-0	CNT[23:0]	R/W	编码器计数器/捕捉值		
			编码器模式：	编码器脉冲的计数器值	0x0000 ~ 0xFFFF
				可读取编码器计数值。 编码器模式下，编码器计数器向上计数或倒计数各编码器脉冲(ENCLK)。 在 CW 旋转时，编码器计数器向上计数；当它达到<RELOAD15:0>值时，在下一 ENCLK 时回到"0"。 在 CCW 旋转时，编码器计数器倒计数； 当达到"0"时，在下一 ENCLK 时重新加载<RELOAD15:0>值。	
			传感器模式： (事件计数)	编码器脉冲的计数器值	0x0000 ~ 0xFFFF
				可读取编码器计数值。 传感器事件计数模式下，编码器计数器向上计数或倒计数各编码器脉冲(ENCLK)。 在 CW 旋转时，编码器计数器向上计数；当它达到"0xFFFF"值时，它在下一 ENCLK 时回到"0"。 在 CCW 旋转时，编码器计数器倒计数； 当它达到"0"时，在下一 ENCLK 时回到"0xFFFF"。	
			传感器模式： (定时器计数)	脉冲检测时间或软件捕捉的值	0x000000 ~ 0xFFFFFF
				可读取编码器计时器的值。 传感器模式下，通过将"1"写入<SFTCAP>，可读取各编码器脉冲(ENCLK)的编码器计数器值，并由软件捕捉。 经系统复位，捕捉的值被清除为"0"。也可通过将<ENCLR>设置为 1 清除计数器然后将<SFTCAP>设置为 1 而清除。 传感器定时器计数模式下，编码器计数器被配置为一个用 fsys 向上计数的自由运行的计数器。当检测到编码器脉冲(ENCLK)时，编码器计数器被清除为"0"。当它已达到"0xFFFFFF"时，自动回到"0"。	
			定时器模式	内计数器的捕捉值或者软件捕捉的值	0x000000 ~ 0xFFFFFF
				通过将"1"写入<SFTCAP>，可读取编码器计数器值，并由软件捕捉。当<ZEN> = "1"时，编码器计数器值在<ZESEL>所选 Z 边沿也能被捕捉到<CNT23:0>中。 通过复位将捕捉的值清除为"0"。 通过将<ENCLR>设置为 1，再将<SFTCAP>设置为 1 而清除计数器，它也能得以清除。 在定时器模式时，编码器计数器被配置为一个用 fsys 向上计数的自由运行的计数器。 当它已达到"0xFFFFFF"时，自动回到"0"。	

<CNT[23:16]>仅在传感器模式 (定时器计数) 或定时器模式时使用。在编码器模式或传感器模式(事件计数)，始终读作"0"。

13.5 操作说明

13.5.1 编码器模式

高速定位传感器确定 AB 编码器和 ABZ 编码器的相位输入。

- 事件检测(旋转脉冲) →中断生成
- 事件计数→匹配检测中断生成 (测量传输量)
- 检测旋转方向
- 向上计数/倒计数(在使用中可变)
- 可设置的计数器周期

13.5.2 传感器模式

低速定位传感器确定(零交叉确定)UV 霍尔传感器和 UVW 霍尔传感器的相位输入。

有两类传感器模式，例如事件计数模式和定时器计数模式(用 fsys 计数)。

13.5.2.1 事件计数模式

- 事件检测(旋转脉冲) →中断生成
- 事件计数 → 发生匹配中断 (测量传输量)
- 旋转方向检测

13.5.2.2 定时器计数模式

- 事件检测(旋转脉冲) →中断生成
- 定时器计数
- 旋转方向检测
- 捕捉功能→事件捕捉(测量事件间隔) →中断生成

  软件捕捉
- 异常检测时间错误(定时器比较)→匹配检测中断生成
- 反向检测错误→旋转方向改变造成的错误标志

13.5.3 定时器模式

该模式能用作通用的 24-位定时器。

- 24-位向上计数器
- 计数器清除控制(软件清除，定时器清除，外触发器和自由运行计数)
- 比较功能→匹配检测中断生成
- 捕捉功能→外触发器捕捉→中断生成

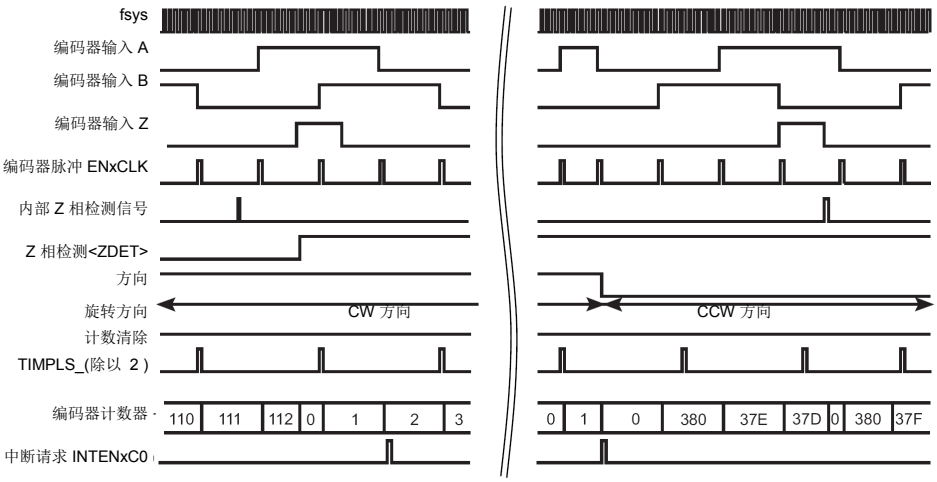
  软件捕捉

13.6 功能

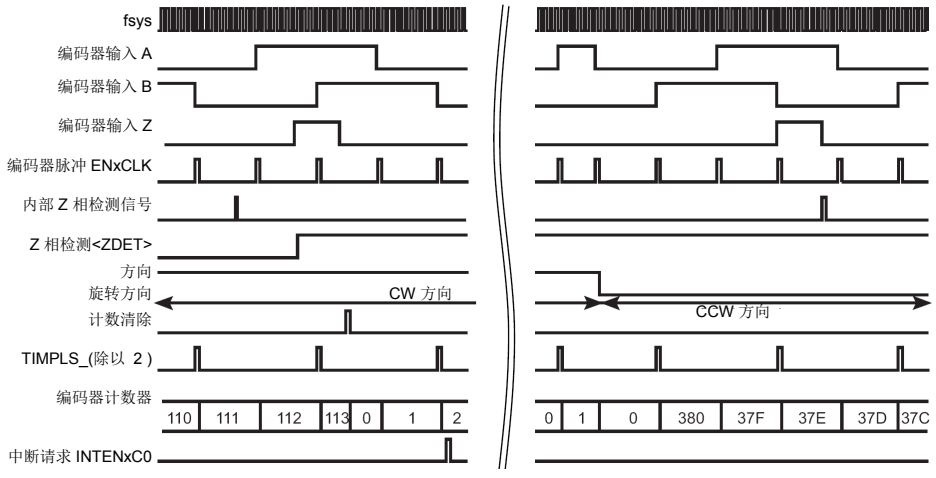
13.6.1 模式操作概述

13.6.1.1 编码器模式

1. 若<ZEN> = 1 (<RELOAD> = 0x0380，<EN0INT> = 0x0002)



2. 若<ZEN> = 0 (<RELOAD> = 0x0380，<EN0INT> = 0x0002)



- MCU 的增量编码器输入应与 A，B，Z 通道连接。编码器计数器对 ENCLK 脉冲计数，其乘以由解码的 A 和 B 正交信号推导的 4 时钟。
- 在 CW 旋转时(即 A 相对于 B 具有 90 度相位超前)，编码器计数器向上计数；当它达到<RELOAD>值时，它在下一 ENCLK 时回到"0"。
- 在 CCW 旋转时(即 A 相对于 B 具有 90 度相位滞后)，编码器计数器倒计数；当它达到"0x0000"时，它在下一 ENCLK 时重新加载<RELOAD>值。
- 此外，当<ZEN> = "1"时，编码器计数器在 CW 旋转时在 Z 的上升沿被清除为"0"，在 CCW 时在 Z 的下降沿 (在内部 Z_检测时间)。若 ENCLK 边沿与 Z 边沿匹配，则编码器计数器在无增量或减量的情况下被清除为"0"。

• 当<ENCLR>设置为 1 时，编码器计数器被清除为"0"。

• <UD>在 CW 旋转时设置为 1 ，在 CCW 旋转时被清除为"0"。

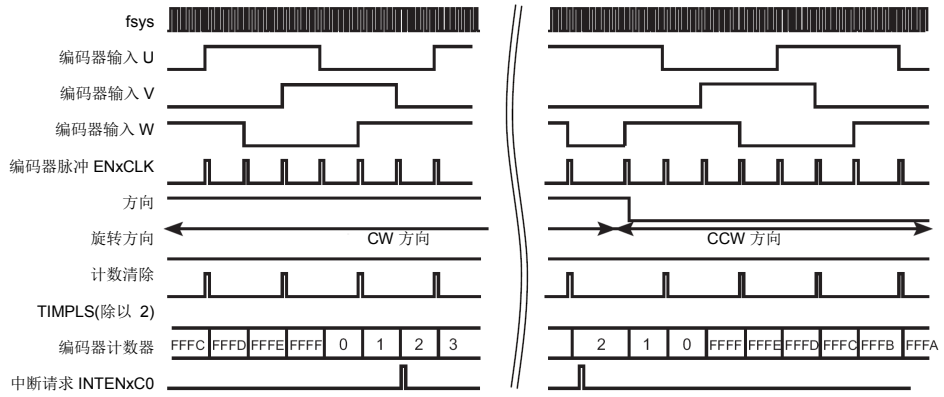
• TIMPLS,通过用程序性因数除 ENCLK 推导出,可被逐出外部。

• 若<CMPEN>设置为 1，当编码器计数器已达到<EN0INT>时，生成中断。然而，当<ZEN> = "1"及<ZDET> = "0"时，不会发生中断。

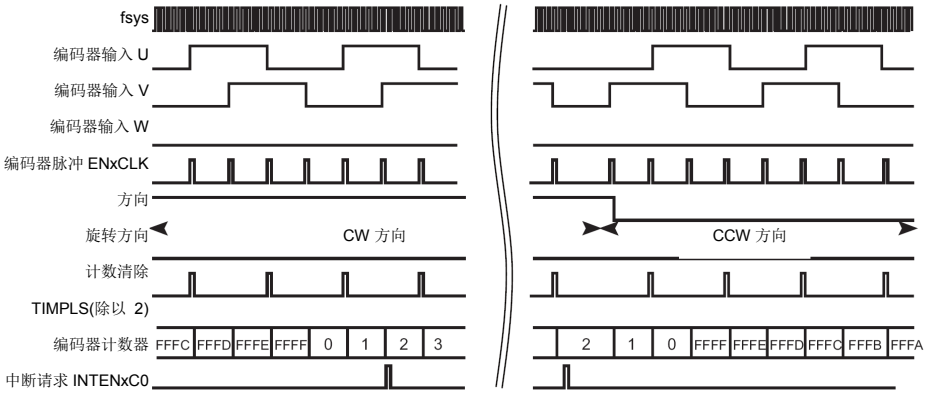
• 当<ZDET>和<UD>设置为"0"时，<ENRUN>被清除为"0"。

13.6.1.2 传感器模式(事件计数)

1. 若<P3EN> = 1 (<EN0INT> = 0x0002)



2. 若<P3EN> = 0 (<EN0INT> = 0x0002)



• MCU 的霍尔传感器输入应与 U，V，W 通道连接。编码器计数器对 ENCLK 脉冲计数，(当<P3EN> = "0"时)其乘以由解码的 U，V 信号推导的 4 时钟，或者(当<P3EN> = "1"时)其乘以由解码的 U，V，W 信号推导的 6 时钟。

• 在 CW 旋转时(即 U 通道相对于 V 通道具有 90 度相位超前；V 通道相对于 W 通道具有 90 度相位超前)，编码器计数器向上计数；当它达到"0xFFFF"时，在下一 ENCLK 时回到"0"。

• 在 CCW 旋转时(即 U 通道相对于 V 通道具有 90 度相位滞后；V 通道相对于 W 通道具有 90 度相位滞后)，编码器计数器倒计数；当它达到"0x0000"时，在下一 ENCLK 时回到"0xFFFF"。

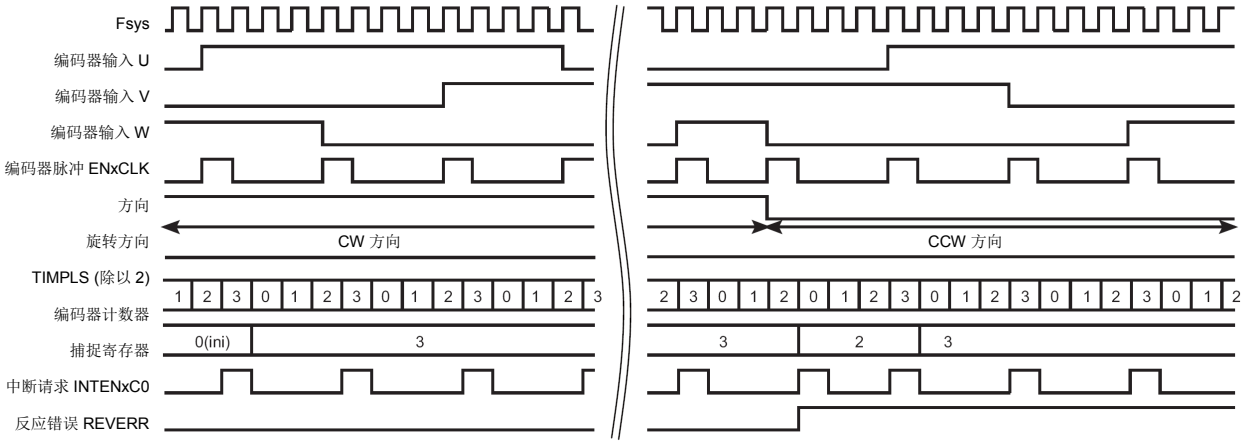
• 当<ENCLR>设置为 1 时，内部计数器被清除为"0"。

• <UD>在 CW 旋转时设置为 1，在 CCW 旋转时被清除为”0”。

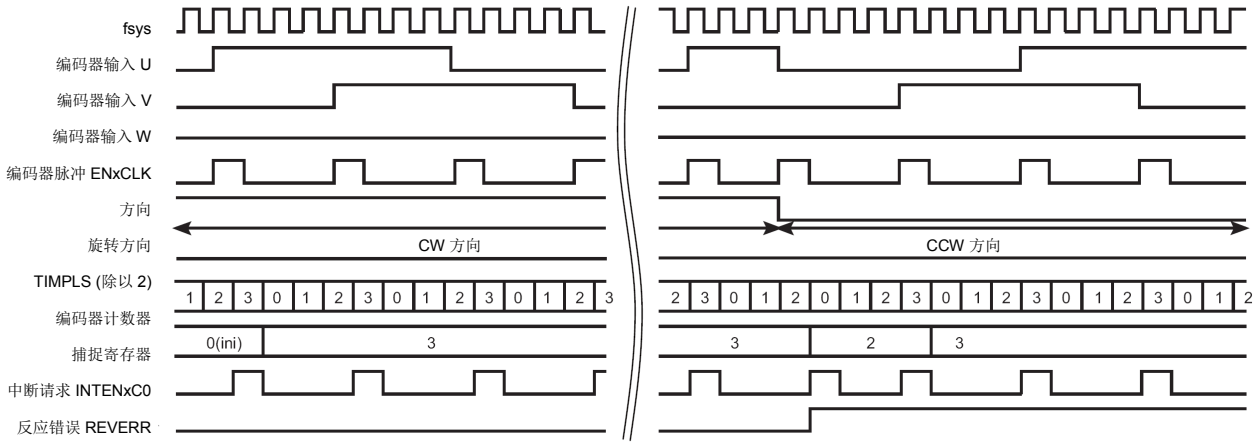
- TIMPLS,通过用程序性因数除 ENCLK 推导出,可被逐出外部。
- 若<CMPEN>设置为 1，当内计数器已达到<EN0INT>值时，生成中断。
- 当<UD>和<ENRUN>设置为"0"时，<UD>被清除为"0"。

13.6.1.3 传感器模式(定时器计数)

1. 若<P3EN> = 1 (<EN0INT> = 0x0002)



2. 若 <P3EN> = 0 (<EN0INT> = 0x0002)

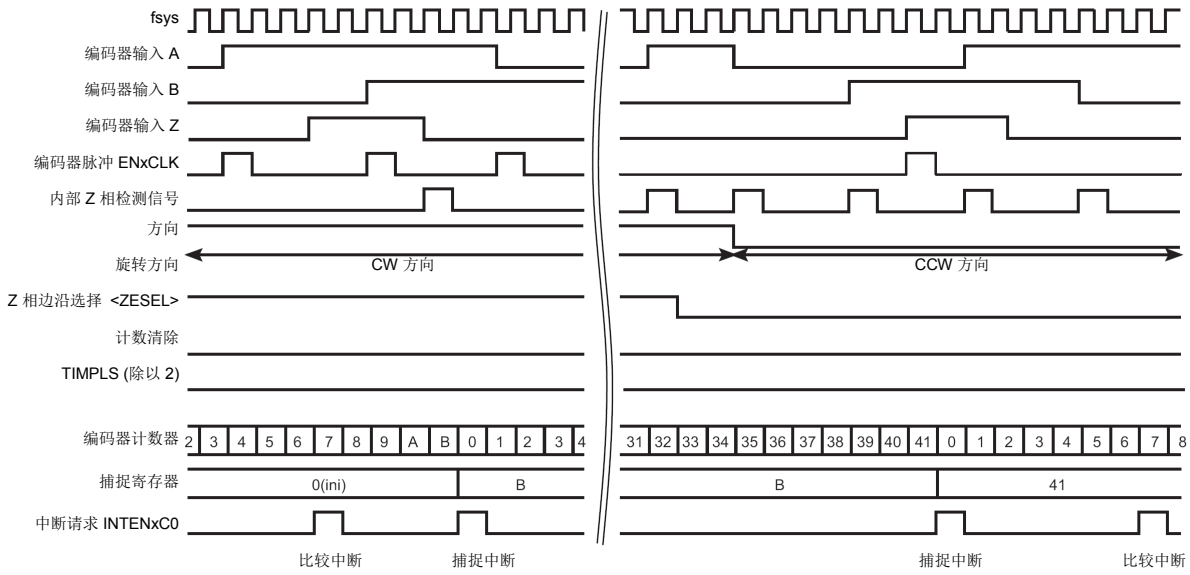


- 在传感器定时器计数模式时，MCU 的霍尔传感器应与 U，V，W 通道连接。编码器计数器测量两连续 ENCLK 脉冲之间的间隔，(当<P3EN> = "0"时)其乘以由解码的 U，V 信号推导的 4 时钟，或者(当<P3EN> = "1"时)其乘以由解码的 U，V，W 信号推导的 6 时钟。
- 编码器计数器始终向上计数；它在 ENCLK 时被清除为"0"。当编码器计数器已达到"0xFFFFFF"时，回到"0"。
- 当<ENCLR>设置为 1 时，编码器计数器被清除为”0”。
- ENCLK 将编码器计数器值捕捉到 EN0CNT 寄存器中。捕捉的计数器值能从 EN0CNT 中读取。
- 将软件捕捉位<SFTCAP>设置为 1 会造成编码器计数器值被捕捉到 ENCNT 寄存器中。该捕捉操作可在任何时间进行。捕捉的计数器值能从 ENCNT 中读取。

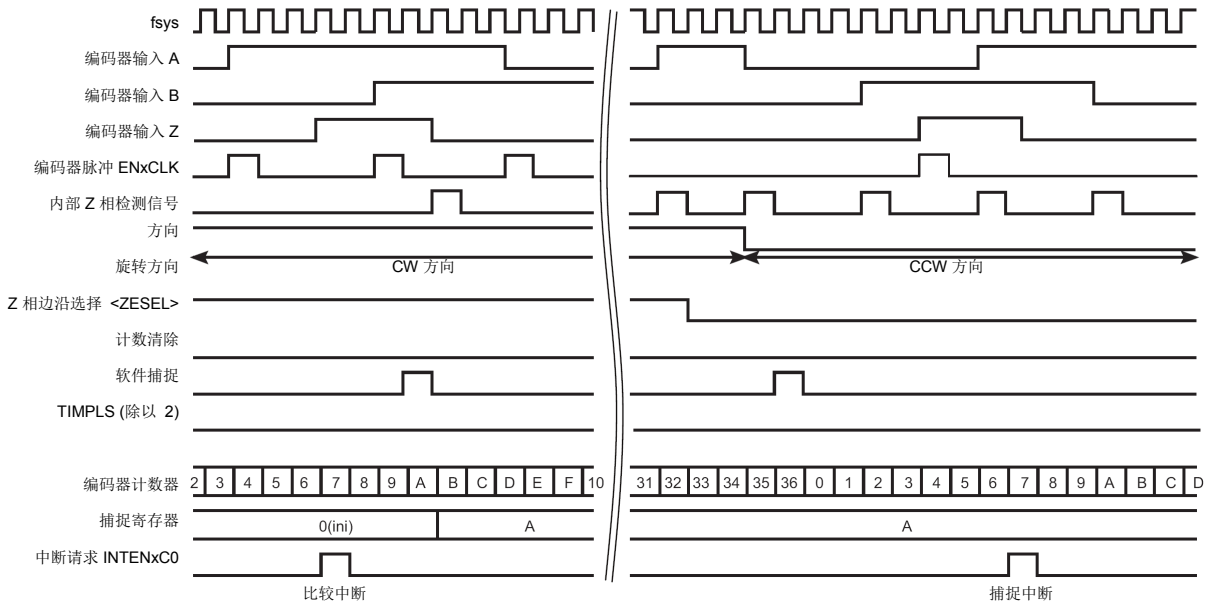
- <UD>在 CW 旋转时设置为 1 ，在 CCW 旋转时被清除为”0”。
- 若<CMPEN>设置为 1，当编码器计数器已达到<EN0INT>时，生成中断。
- 当<ENRUN>设置为"0"时，<UD>被清除为"0"。
- 当旋转方向已变化时，<REVERR>设置为 1 。在读取后，该位被清除为"0"。
- 不管<ENRUN>值，ENCNT 寄存器值(捕捉的值)得到保留。ENCNT 寄存器仅能通过复位得到清除。

13.6.1.4 定时器模式

1. 若<ZEN> = 1 (<EN0INT> = 0x0006)



2.若<ZEN> = 0 (<EN0INT> = 0x0006)

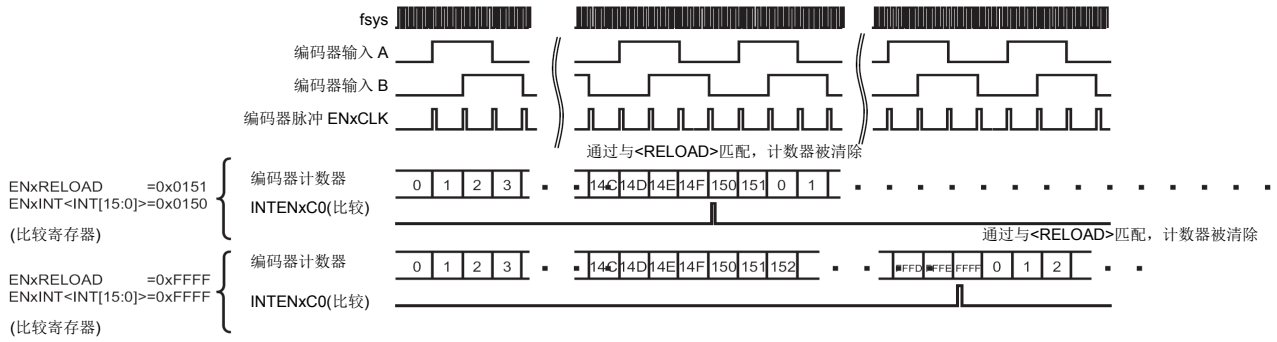


- 当<ZEN> = "1"时，Z 输入引脚用作触发器。当<ZEN> = "0"时，外部输入不用于触发定时器。

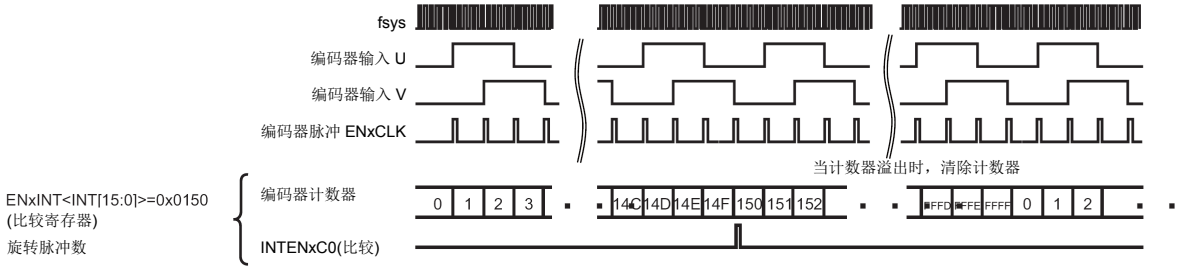
- 编码器计数器始终向上计数。若<ZEN> = "1"，当<ZESEL>设置为"0"时，计数器在 Z 的上升沿被清除为"0"，当<ZESEL>设置为"1"时，计数器在 Z 的下降沿被清除为"0"。 当编码器计数器已达到"0xFFFFFF"时，回到"0"。
- 当<ENCLR>设置为 1 时，编码器计数器被清除为"0"。
- Z-检测造成编码器计数器值被捕捉到 ENCNT 寄存器中。 捕捉的计数器值能从 ENCNT 中读取。
- 将软件捕捉位<SFTCAP>设置为 1 会造成编码器计数器值被捕捉到 ENCNT 寄存器中。该捕捉操作可在任何时间进行。捕捉的计数器值能从 ENCNT 中读取。
- <UD>在 CW 旋转时设置为 1，在 CCW 旋转时被清除为"0"。
- 若<CMPEN>设置为 1，当编码器计数器已达到<ENINT>值时，生成中断。
- 当<ENRUN>设置为"0"时，<UD>被清除为"0"。
- 不管<ENRUN>值，ENCNT 寄存器值(捕捉的值)得到保留。ENCNT 寄存器仅能通过复位得到清除。

13.6.2 当<CMPEN> = 1 时，计数器和中断生成操作

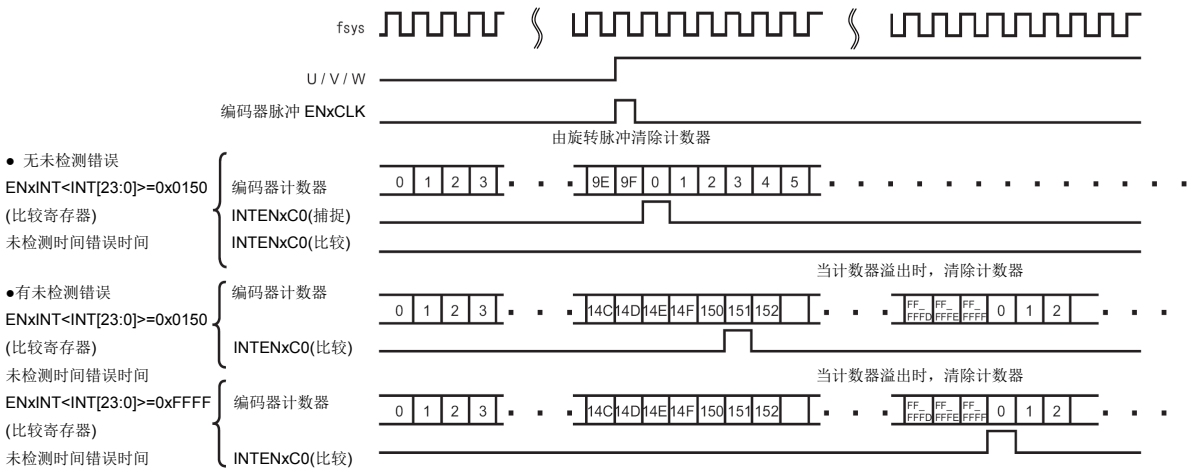
13.6.2.1 编码器模式



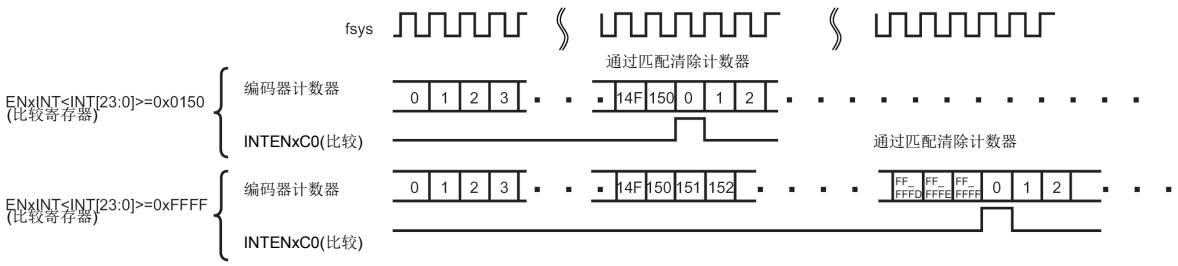
13.6.2.2 传感器模式(事件计数)



13.6.2.3 传感器模式(定时器计数)



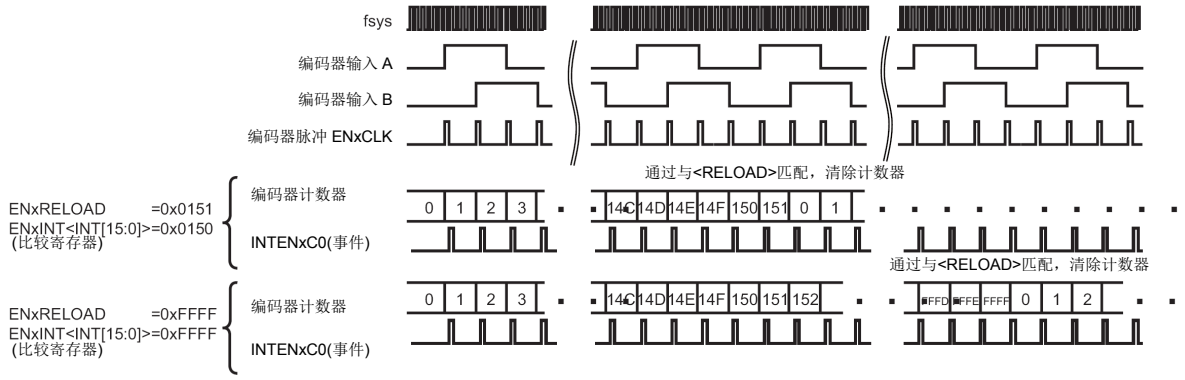
13.6.2.4 定时器模式


fsys
通过匹配清除计数器
ENxINT≤INT[23:0]>=0x0150
(比较寄存器)
编码器计数器
0 1 2 3 14F 150 0 1 2
INTENxC0(比较)
通过匹配清除计数器
ENxINT≤INT[23:0]>=0xFFFF
(比较寄存器)
编码器计数器
0 1 2 3 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2
INTENxC0(比较)

13.6.3 当<CMPEN> = 0 时，计数器和中断生成操作

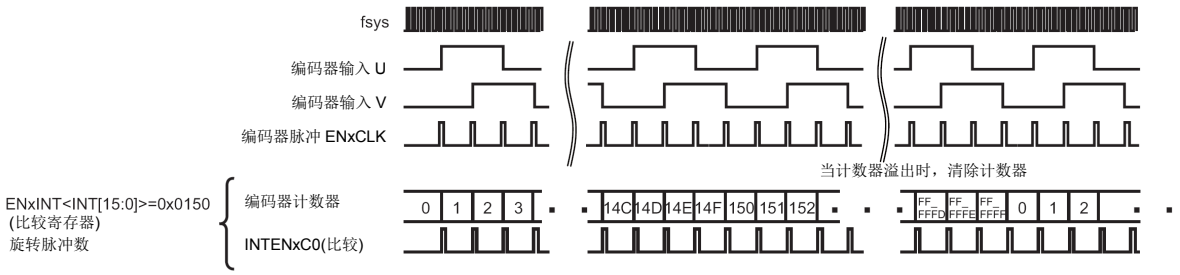
13.6.3.1 编码器模式

<ENDEV>="000"

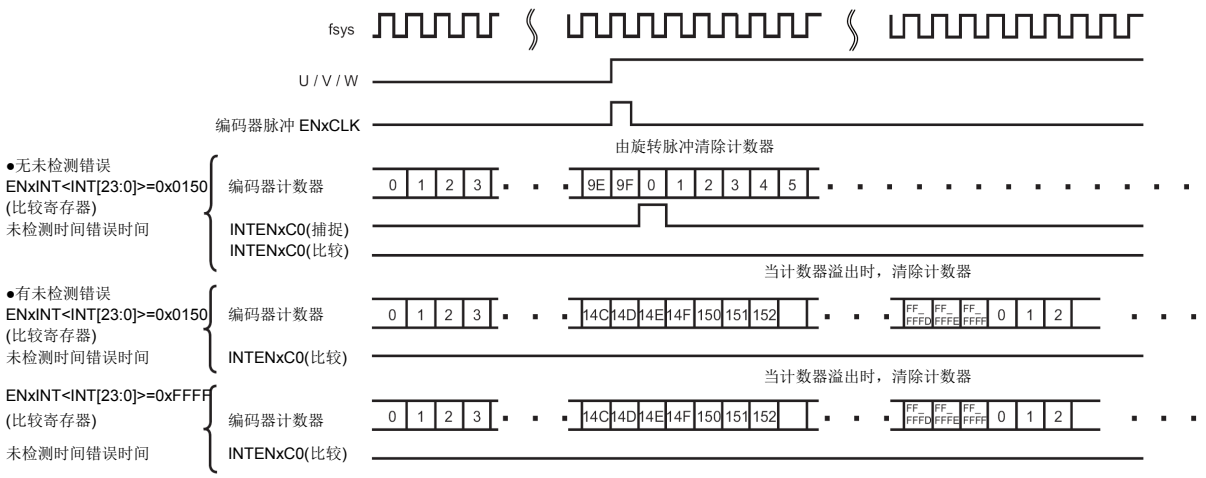


13.6.3.2 传感器模式(事件计数)

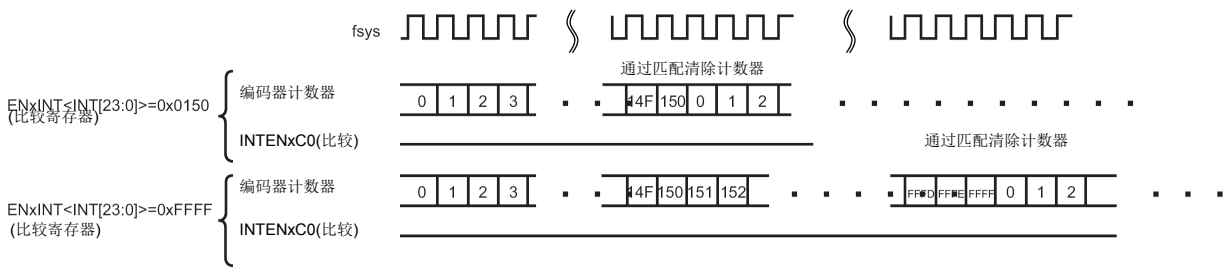
<ENDEV>="000"



13.6.3.3 传感器模式 (定时器计数)



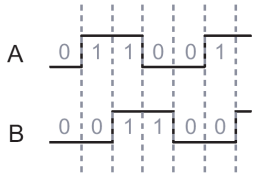
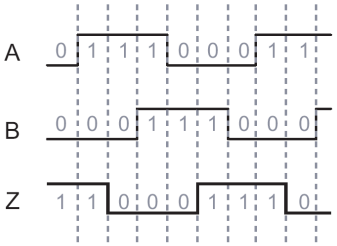
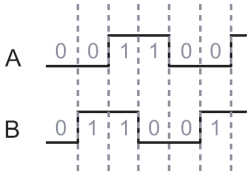
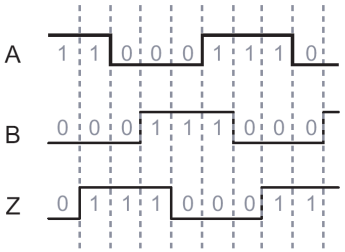
13.6.3.4 定时器模式


fsys
通过匹配清除计数器
ENxINT≤INT[23:0]>=0x0150
(比较寄存器)
编码器计数器
INTENxC0(比较)
0 1 2 3
14F 150 0 1 2
通过匹配清除计数器
ENxINT<INT[23:0]>=0xFFFF
(比较寄存器)
编码器计数器
INTENxC0(比较)
0 1 2 3
14F 150 151 152
FFFD FFFE FFFF 0 1 2

13.6.4 编码器旋转方向

该电路确定某一相位，即 A-相， B-相或 Z-相。

它被用作共同的 2-相输入(A，B) 和 3-相输入 (A，B，Z)。当使用 3-相输入时，设置<P3EN> = "1"。

	2-相输入	3-相输入
CW 方向		
CCW 方向		

13.6.5 计数器电路

计数器电路具有 24-位向上/向下计数器。

13.6.5.1 操作说明

根据操作模式，计数，清除和重新加载操作的控制如表 13-2 所述。

表 13-2 计数器控制

模式 <MODE[1:0]>	<ZEN>	<P3EN>	输入 引脚	计数	操作	计数器清除条件	计数器重新加载条件	计数器 操作范围 (重新加载值)
编码器模式 00	0	0	A，B	编码器 脉冲 (ENCLK)	向上	[1]<ENCLR> = 1 WR [2] 与<RELOAD>匹配	-	0x0000 ~ <RELOAD>
					向下	[1]<ENCLR> = 1 WR	[1] 与 0x0000 匹配	
	1		A,B,Z		向上	[1]<ENCLR> = 1 WR [2] 与<RELOAD>匹配 [3] Z-触发器	-	
					向下	[1]<ENCLR> = 1 WR	[1] 与 0x0000 匹配	
传感器模式 (事件计数) 01	0	0	U,V		向上	[1]<ENCLR> = 1 WR [2] 与 0xFFFF 匹配	-	0x0000 ~ 0xFFFF
					向下	[1]<ENCLR> = 1 WR	[1] 与 0x0000 匹配	
		1	U,V,W		向上	[1]<ENCLR> = 1 WR [2] 与 0xFFFF 匹配	-	
					向下	[1]<ENCLR> = 1 WR	[1] 与 0x0000 匹配	
传感器模式 (定时器计数) 10	0	0	U,V	fsys	向上	[1]<ENCLR> = 1 WR [2] 与 0xFFFFFF 匹配 [3] 编码器脉冲 (ENCLK)	-	0x000000 ~ 0xFFFFFF
		1	U,V,W		向上		-	
定时器模式 11	0	×	-		向上	[1]<ENCLR> = 1 WR [2] 与 0xFFFFFF 匹配 [3] 与<EN0INT>匹配	-	0x000000 ~ 0xFFFFFF
	1		Z		向上	[1]<ENCLR> = 1 WR [2] 与 0xFFFFFF 匹配 [3] 与<EN0INT>匹配 [4] Z-触发器	-	

注： 将"0"写入<ENRUN>，计数器值不会被清除。若再次设置<ENRUN> = "1"，则计数器从已停止的计数器值重新开始。若清除计数器值，则会将"1" 写入<ENCLR>，以执行软件清除。

13.6.6 中断

中断由四种中断组成，即事件中断(分频脉冲和捕捉)，异常检测时间中断，定时器比较中断，捕捉中断。

13.6.6.1 操作说明

当设置<INTEN> = "1"时，会发生计数器值和编码器脉冲产生的中断。

中断因数的设置由六类设置组成，即按操作模式进行的设置及<CMPEN>和<ZEN>的设置。中断因数如表 13-3 所示。

表 13-3 中断因数

	中断因数	说明	模式	中断输出	状态标志
1	事件计数中断	当<CMPEN> = 1 时，编码器计数器对事件(编码器脉冲)计数。当它已达到<EN0INT>中编程的值时，会发生中断。	编码器模式 和 传感器模式 (事件计数)	<INTEN> = 1 和 <CMPEN> = 1	<CMP>
2	事件中断 (分频脉冲)	在各分频的时钟脉冲(1 ~ 128 分频)(用<ENDEV>中编程的系数除编码器脉冲而得)，发生中断。		<INTEN> = 1	不适用
3	事件中断 (捕捉中断)	发生中断，表明已发生事件(编码器脉冲)，造成计数器值在旋转脉冲时间被捕捉。	传感器模式 (定时器计数)	<INTEN> = 1	不适用
4	异常检测时间 错误中断	当<CMPEN> = 1 时，ENC 使用以 fsys 向上计数的计数器，并由事件(编码器脉冲)清除。若在<EN0INT>中编程的时期内无事件发生，则发生中断。		<INTEN> = 1 和 <CMPEN> = 1	<CMP>
5	定时器比较中断	当<CMPEN> = 1 时，中断 在定时器已达到<EN0INT>中编程的值时发生。	定时器模式	<INTEN> = 1 和 <CMPEN> = 1	<CMP>
6	捕捉中断	当计数器值已在外触发器(Z 输入)上被捕捉时，会发生中断。		<INTEN> = 1	不适用

在传感器定时器计数模式和定时器模式时，编码器计数器值能被捕捉到 ENCNT 寄存器中。

捕捉的计数器值能从 ENCNT 寄存器中读取。

在传感器定时器计数模式下，在发生事件(编码器脉冲)后，编码器计数器值被捕捉到 ENCNT 寄存器中。用软件将 1 写入<SFTCAP>，计数器值也能被捕捉。

在定时器模式时， 用软件将 1 写入<SFTCAP>，计数器就能被捕捉。若<ZEN>设置为 1 ，计数器值也能由外触发器按照<ZESEL>选择的 Z 信号输入的边沿捕捉。

14. 上电复位电路(POR)

当打开电源时，上电复位电路产生复位。当电源电压低于上电复位电路检测电压时，会产生上电复位信号。

14.1 配置

上电复位电路由参考电压产生电路，比较器和上电计数器组成。

比较器对电阻梯分得的电源电压和参考电压产生电路产生的电压进行比较。

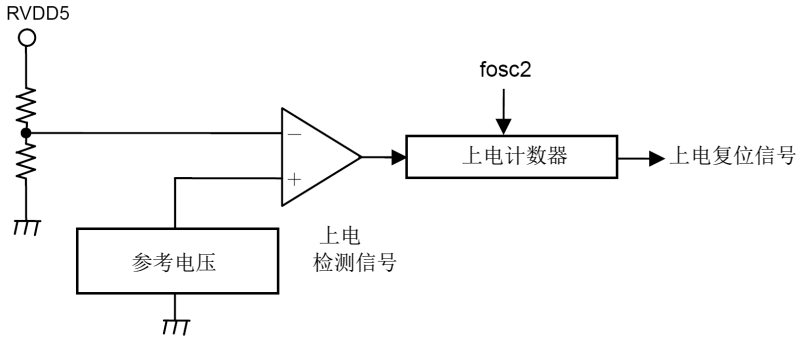


图 14-1 上电复位电路

14.2 功能

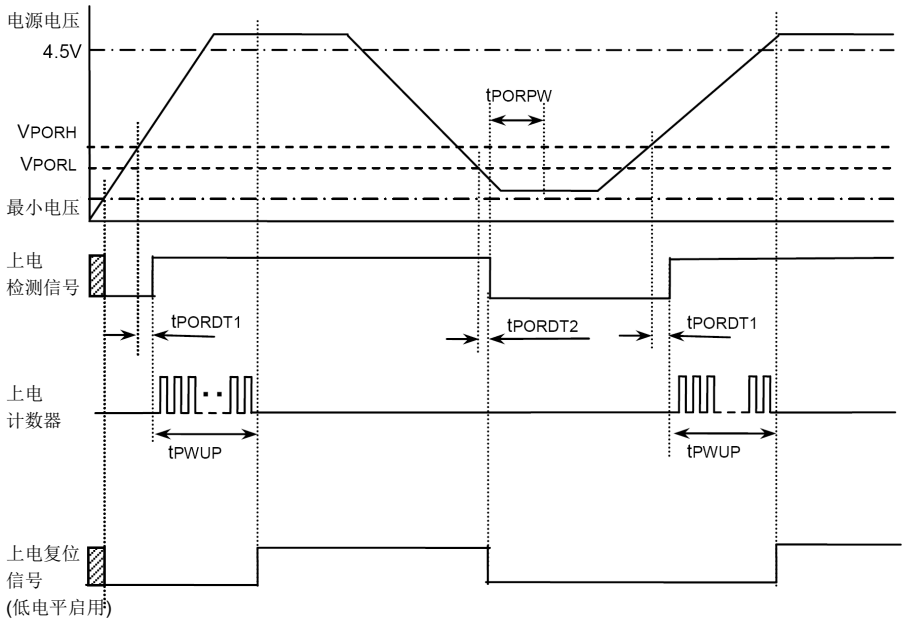
当电源电压上升时，若电源电压等于或小于上电复位电路的释放电压，则产生上电复位信号。若电源电压超过上电复位电路的释放电压，则先激活上电计数器，再激活 $2^{15}/f_{osc2}$ (s)，释放上电复位信号。

当电源电压下降时，若电源电压等于或小于上电复位电路的检测电压，则产生上电复位信号。

在上电复位产生时，上电计数器电路及 CPU 和外围电路复位。

当上电复位电路是在无外部复位输入信号的情况下激活时，电源电压应在检测到上电复位电路的释放电压后 3 ms 内增加到推荐的工作电压范围(注)。若电源电压未达到该范围，则 TMPM370 无法正常工作。

(注) 当电源电压上升时，直到(RVDD5 引脚)电源电压达到推荐的工作电压范围(4.5 V ~ 5.5 V)并且过去 200 μs，下列条件应得到满足；端口 L (PL0 和 PL1)打开或者输入电压在 0.5 V 内。



注 1：视电源电压的波动情况而定，上电复位电路可能工作不正常。当设计设备时，应参考并考虑电气特性。

注 2：若电源电压低于上电复位电路最小电压在该电压下，电路无法正常工作，则上电复位信号变成未定义的值。

图 14-2　上电复位的工作时间

符号	参数	最小	典型	最大	单位
VPORH	上电复位释放电压	2.8	3	3.2	V
VPORL	上电复位检测电压	2.6	2.8	3.0	V
tPORDT1	上电复位释放响应时间		30		μs
tPORDT2	上电复位检测响应时间		30		μs
tPORPW	上电复位最小脉冲宽度	45			μs

注 1：因为上电复位释放电压和上电复位检测电压相对变化，所以检测电压从不反向。

上电顺序，详见"电气特性"章节。

关于如何使用外部复位输入，详见"例外"章节中的"复位例外"。

15. 电压检测电路(VLTD)

电压检测电路检测电源电压的降低情况，并产生电压检测复位信号。

注：视电源电压(RVDD5)的波动情况而定，电路可能工作不正常。当设计设备时，应参考并考虑电气特性。

15.1 配置

电压检测电路由参考电压产生电路，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(RVDD5)由电阻梯分压，并被输入检测电压选择电路。检测电压选择电路按照规定的检测电压(VDLVL)选择电压，比较器将它与参考电压进行比较。

当电源电压(RVDD5)低于检测电压(VDLVL)时，产生电压检测复位信号。

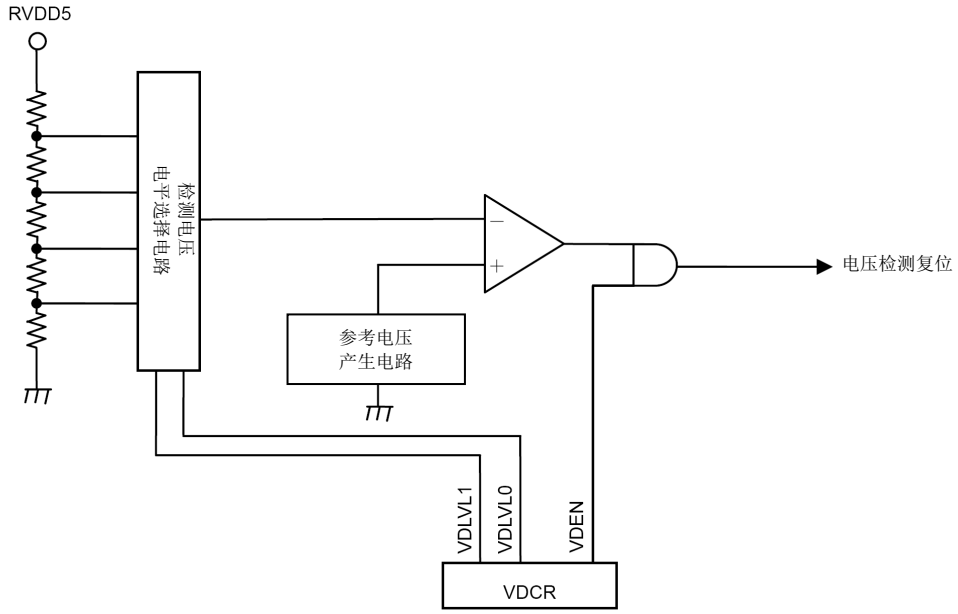


图 15-1 电压检测电路

15.2 控制

电压检测电路受电压检测控制寄存器控制。

电压检测控制寄存器

VDCR (0x4004_0900)

	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	VDLVL1	VDLVL0	VDEN
读取/写入	R	R	R	R	R	R/W		R/W
复位后	0	0	0	0	0	00		0

VDLVL[1:0]	检测电压的选择	00：保留 01：4.1 ± 0.2 V 10：4.4 ± 0.2 V 11：4.6 ± 0.2 V
VDEN	启用/禁用 电压检测操作	0：禁用电压检测操作 1：启用电压检测操作

注 1：VDCR 由上电复位或外部复位输入进行初始化。

15.3 功能

检测电压可由 VDCR<VDLVL[1:0]>选择。电压检测的启用/禁用可用 VDCR<VDEN>进行编程。

在电压检测操作启用后，当电源电压(RVDD5)低于检测电压 <VDLVL[1:0]> 时，会产生电压检测复位信号。

15.3.1 启用/禁用电压检测操作

将 VDCR<VDEN>设置为"1"，就会启用电压检测操作。.将其设置为"0"，禁用该操作。

恰在上电复位或由外部复位输入进行的复位被释放后，VDCR<VDEN>被清除为"0"。

注：当电源电压 (RVDD5) 低于检测电压(VDLVL)时，将 VDCR<VDEN>设置为"1"，此时产生复位信号。

15.3.2 选择检测电压电平

选择 VDCR<VDLVL[1:0]>时的检测电压。

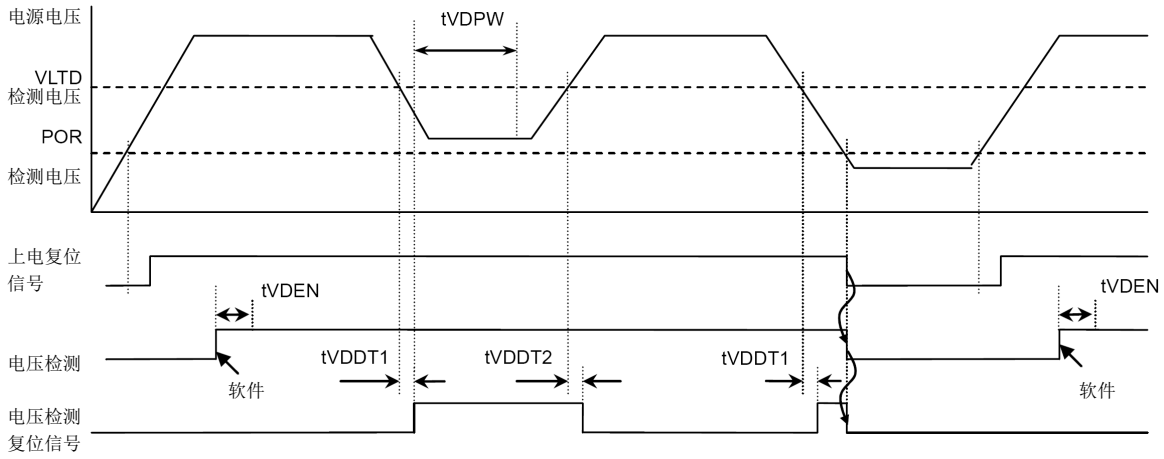


图 15-2 电压检测时间

符号	参数	最小	典型	最大	单位
tVDEN	在启用电压检测后的设置时间		40		μs
tVDDT1	电压检测响应时间		40		μs
tVDDT2	电压检测释放时间		40		μs
tVDPW	电压检测最小脉冲宽度	45			μs

16. 振荡频率检测器(OFD)

16.1 配置

若 CPU 时钟 OFDMNPLL 的高频振荡超过检测频率范围，则振荡频率检测器产生 I/O 复位。

振荡频率检测受 OFDCR1，OFDCR2 寄存器控制，检测频率范围由检测频率设置寄存器 OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON 规定。低检测频率由 OFDMNPLLOFF/OFDMNPLLON 寄存器规定，高检测频率由 OFDMXPLLOFF/OFDMXPLLON 寄存器规定。

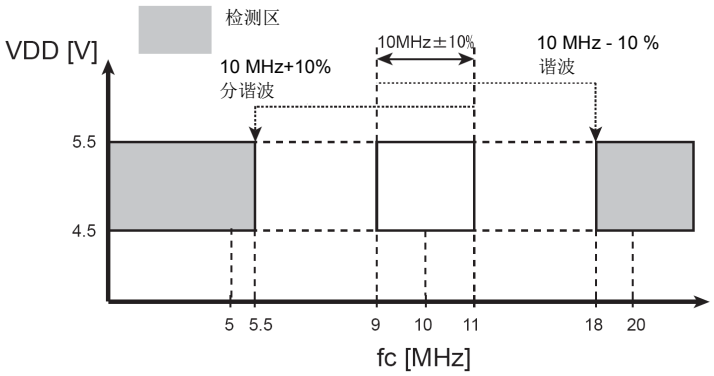
当振荡频率检测启用时，写入 OFDMNPLLOFF /OFDMNPLLON/ OFDMXPLLOFF/ OFDMXPLLON 寄存器的操作禁用。因此，当振荡频率检测禁用时，应设置这些寄存器的检测频率。写入 OFDCR2/OFDMNPLLOFF/ OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON 寄存器的操作受 OFDCR1 寄存器控制。为了写入 OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON 寄存器，应事先将写入启用代码"0xF9"设置为 OFDCR1。为了启用振荡频率检测器，在将"0xF9"设置为 OFDCR1 后，应将"0xE4"设置为 OFDCR2。因为振荡频率检测在外部复位输入，上电复位或 VLTD 复位后禁用，所以应将"0xF9"写入 OFDCR1，将"0xE4"写入 OFDCR2 寄存器，以启用它的功能。

当 TMPM370FYDFG/FYFG 用低检测频率设置寄存器和高检测频率设置寄存器检测到超出频率范围之外时，所有 I/O 经复位变成高阻抗。在 PLLOFF 下，OFDMNPLLOFF 和 OFDMXPLLOFF 寄存器对于检测是有效的，并忽略 OFDMNPLLON/OFDMXPLLON 寄存器的设置值。在 PLLON 情况下，OFDMNPLLON 和 OFDMXPLLON 寄存器对于检测是有效的，并忽略 OFDMNPLLOFF/OFDMXPLLOFF 寄存器的设置值。经振荡频率检测复位，除电源引脚，$\overline{\text{RESET}}$，X1 和 X2 外，所有 I/O 变成高阻抗。若振荡频率检测复位是由于检测到高频率的停止而产生，则寄存器等内部电路在振荡停止时保持条件。为了初始化这些内部电路，需要外部重启振荡。

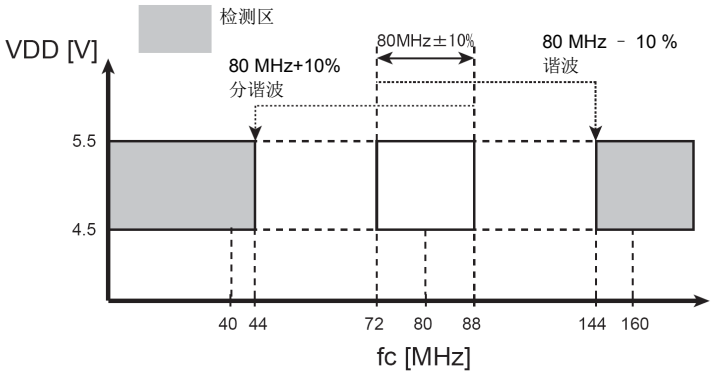
因为所有振荡检测器的所有寄存器(OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/ OFDMXPLLOFF/OFDMXPLLON)未因振荡频率检测器产生的复位而被初始化，所以振荡检测在振荡频率检测复位时保持其功能。因此，若发生振荡频率检测复位，则在 CPU 时钟恢复其正常频率前，复位不会被释放。

注 1：振荡频率检测复位仅在 NORMAL 和 IDLE 模式时可用。在 STOP 模式时，振荡频率检测复位自动禁用。

注 2：当 PLL 受 CGPLLSEL 寄存器控制(启用或禁用)时，OFD 必须事先禁用。若用 PLL-ON 产生 OFD 复位，则检测频率设置寄存器(OFDMNPLLON/OFDMXPLLON)自动切换到 OFDMNPLLOFF/OFDMXPLLOFF。



(a) 在 PLLOFF 情况下



(b) 在 PLLON 情况下

图 16-1 检测频率范围的例子

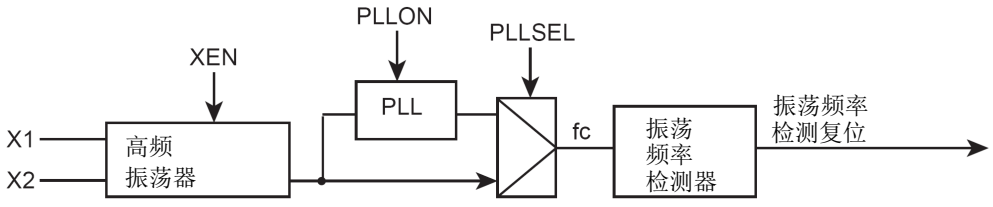


图 16-2 振荡频率检测器

16.2 控制

该振荡频率检测受控于振荡频率检测控制寄存器 2 (OFDCR2)。该检测频率是由较低/较高检测频率设置寄存器(OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF 与 OFDMXPLLON)指定的。对 OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON 的写入，受控于振荡频率控制寄存器 1 (OFDCR1)。

振荡频率检测控制寄存器 1

OFDCR1 (0x4004_0800)

	31-8							
比特符号	-							
读取/写入	R							
复位后	0							
	7	6	5	4	3	2	1	0
比特符号	OFDWEN7	OFDWEN6	OFDWEN5	OFDWEN4	COFDWEN3	OFDWEN2	OFDWEN1	OFDWEN0
读取/写入	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	1	1	0
功能	0x06：禁用向 OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/ OFDMXPLLON 写入(写入禁用代码) 0xF9：启用向 OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/ OFDMXPLLON 写入(写入启用代码) 其它：保留(注 1)							

注 1：仅"0x06"与"0xF9"对 OFDCR1 有效。如果向 OFDCR1 写入"0x06"与"0xF9"以外的其它值，"0x06"即被自动写入到 OFDCR1。
注 2：OFDCR1 的初始化可由该 $\overline{\text{RESET}}$ 引脚，上电复位或 VLTD 复位实现。

振荡频率检测控制寄存器 2

OFDCR2 (0x4004_0804)

	31-8							
比特符号	-							
读取/写入	R							
复位后	0							
	7	6	5	4	3	2	1	0
比特符号	OFDSEN7	OFDSEN6	OFDSEN5	OFDSEN4	OFDSEN3	OFDSEN2	OFDSEN1	OFDSEN0
读取/写入	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0
功能	0x00：振荡频率检测的禁用 0xE4：振荡频率检测的启用 其它：保留(注 1)							

注 1：仅"0x00"与"0xE4"对 OFDCR2 有效。向 OFDCR2 写入"0x00"与"0xE4"以外的其它值，会被忽略。
注 2：通过将"0x06"设置到 OFDCR1，可保护向 OFDCR2 写入；但无需设置 OFDCR1，从 OFDCR2 读取即始终处于已启用状态。
注 3：OFDCR2 的初始化可由该 $\overline{\text{RESET}}$ 引脚，上电复位或 VLTD 复位实现。

较低检测频率设置寄存器 (如果是 PLL OFF)

OFDMNPLLOFF (0x4004_0808)

	31-9						8	
比特符号	-						OFDMNPLLOFF	
读取/写入	R						R/W	
复位后	0						0	
	7	6	5	4	3	2	1	0
比特符号	OFDMNPLLOFF							
读取/写入	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	1	1	0	1	0

较低检测频率设置寄存器(如果 PLL ON)

OFDMNPLLON (0x4004_080C)

	31-9						8	
比特符号	-						OFDMNPLLON	
读取/写入	R						R/W	
复位后	0						0	
	7	6	5	4	3	2	1	0
比特符号	OFDMNPLLON							
读取/写入	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	1	1	0	1	0	0	0	1

较高检测频率设置寄存器(如果 PLL OFF)

OFDMXPLLOFF (0x4004_0810)

	31-9						8	
比特符号	-						OFDMXPLLOFF	
读取/写入	R						R/W	
复位后	0						0	
	7	6	5	4	3	2	1	0
比特符号	OFDMXPLLOFF							
读取/写入	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	1	0	1	0	0	0

较高检测频率设置寄存器(如果 PLL ON)

OFDMXPLLON (0x4004_0814)

	31-9						8	
比特符号	-						OFDMXPLLON	
读取/写入	R						R/W	
复位后	0						1	
	7	6	5	4	3	2	1	0
比特符号	OFDMXPLLON							
读取/写入	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	1	1	1	0	0	1

注 1： 该复位后的值为暂定值。

注 2： 在该振荡频率检测电路被启用(OFDCR2="0xE4")或已用 OFDCR1="0x06"禁用写入时，OFDMNPLLOFF，OFDMNPLLON，OFDMXPLLOFF 与 OFDMXPLLON 无法被写入。如试图写入 OFDMNPLLOFF，OFDMNPLLON，OFDMXPLLOFF 与 OFDMXPLLON，则写入操作无法完成。

注 3： 通过将"0x06"设置到 OFDCR1，可使得写入 OFDMNPLLOFF，OFDMNPLLON，OFDMXPLLOFF 与 OFDMXPLLON 处于受保护状态，而在未设置 OFDCR1 的情况下，从 OFDMNPLLOFF，OFDMNPLLON，OFDMXPLLOFF 与 OFDMXPLLON 读取则始终处于被启用状态。。

注 4： 根据拟在 OFDMNPLLOFF<OFDMXPLLOFF 的条件下使用的时钟脉冲频率，将适当的值指定给 OFDMNPLLOFF 与 OFDMXPLLOFF。有关该值的计算方式，请参看"16.3.2 设置较低与较高的检测用频率"。

注 5： 根据拟在 OFDMNPLLON<OFDMXPLLON 的条件下使用的时钟脉冲频率，将适当的值指定给 OFDMNPLLON 与 OFDMXPLLON。有关该值的计算方式，请参看"16.3.2 设置较低与较高的检测用频率"。

注 6： 通过 $\overline{\text{RESET}}$ 引脚，上电复位或 VLTD 复位，对 NPLLON，OFDMXPLLOFF 与 OFDMXPLLON 进行初始化。

注 7： 通过 PLLON 的设置，OFDMNPLLOFF/OFDMXPLLOFF 与 OFDMNPLLON/OFDMXPLLON 可实现自动切换。

16.3 功能

16.3.1 启用与禁用振荡频率检测

将"0xE4"写入到 OFDCR2 且 OFDCR1="0xF9"即可启用振荡频率检测，将"0x00"写入到 OFDCR2 且 OFDCR1="0xF9"即可禁用振荡频率检测。

通过 $\overline{\text{RESET}}$ 引脚，上电复位或 VLTD 复位，可实现 OFD 寄存器的初始化。

在用上述复位将 OFDCR1 初始化为"0x06"，将 OFDCR2 初始化为"0x00"之后，振荡频率检测写入到该寄存器即被禁用。在未对 OFDCR1 进行设置的情况下，"从 OFDCR2 读取"始终处于已启用状态。

注：在将数据写入到 OFDCR2 之后，将"0x06" 设置到 OFDCR1 以保护 OFDCR2 寄存器。

在执行 STOP 模式 OFDCR2=0xE4 时，振荡频率检测即自动被禁用。在解除 STOP 与预热周期之后，该振荡频率检测即被启用。振荡频率检测仅在 NORMAL 与 IDLE 模式下可用。表 16-1 给出了振荡频率检测器的适用性。

表 16-1 振荡频率检测器的适用性

操作模式	振荡频率检测 (OFDCR2=0xE4)	振荡频率检测 RESET 后的所有 I/Os 条件 (电源，$\overline{\text{RESET}}$，X1，X2 引脚除外)
NORMAL	适用	高阻抗
IDLE	适用	高阻抗
STOP (包括预热周期)	振荡频率检测即自动被禁用。	
通过振荡频率检测复位 实现复位	适用	高阻抗
看门狗定时器复位 SYSRESETREQ 复位	适用	高阻抗
RESET 通过外部复位 上电复位 VLTD 复位	禁用	-

图 16-3 振荡频率检测的适用性

16.3.2 设置较低与较高的检测频率

可根据目标时钟脉冲与基准的最大误差，计算出检测频率的上限与下限。基准时钟脉冲频率为9.5MHz，误差为±10%。

a)	目标时钟	最大
b)		最小
c)	基准时钟	最大(10.5MHz)
d)		最小(8.5MHz)

设定值的计算方式给出如下。

检测频率的上限值	$= 1 \div \{ (d \div 2^7) \div (a \div 4) \}$	(小数位后舍去)
检测频率的下限值	$= 1 \div \{ (c \div 2^7) \div (b \div 4) \}$	(小数位后进位舍入)

16.3.3 振荡频率检测复位

如果 TMPM370FYDFG/FYFG 检测到 OFDMNPLLOFF/OFDMNPLLON 指定的较低频率，或 OFDMXPLLOFF/OFDMXPLLON 指定的较高频率，则该振荡频率检测器会针对所有 I/O 输出一个复位信号。

a. 在高频振荡变得异常时

一旦某异常(较低或较高)频率振荡持续一段时间(T_{OFD})，该振荡频率检测复位即被生成。通过振荡频率检测，复位可初始化所有 I/O 电源引脚除外，$\overline{\text{RESET}}$，X1 与 X2 变为高阻抗。

b. 在高频振荡停止时

一旦高频振荡停止一段时间(T_{OFD})，该振荡频率检测复位即被生成。通过振荡频率检测，复位可初始化所有 I/O 电源引脚除外，$\overline{\text{RESET}}$，X1 与 X2 变为高阻抗。不过，由于 CPU 等内部线路的初始化是通过高频所锁存的复位信号实现的，因此，这些内部线路会保持在振荡频率检测状态。

一旦该振荡继续执行并持续一段时间(T_{OFD})，该振荡频率检测复位即被解除。

17. 看门狗定时器(WDT)

看门狗定时器(WDT)用于检测噪波或其它干扰所导致的 CPU 故障(超速)，并进行修复使 CPU 恢复正常运行。

如果该看门狗定时器检测到超速，其即可生成一个 INTWDT 中断或复位。

注：INTWDT 中断是非屏蔽中断(NMI)的一个因素。

此外，该看门狗定时器还可经由看门狗定时器引脚（$\overline{\text{WDTOUT}}$），通过输出"低"将所检测到的故障通知外部外围设备。

注：本产品不具备该看门狗定时器输出引脚($\overline{\text{WDTOUT}}$ **)。**

17.1 配置

图 17-1 给出了该看门狗定时器的方块图。

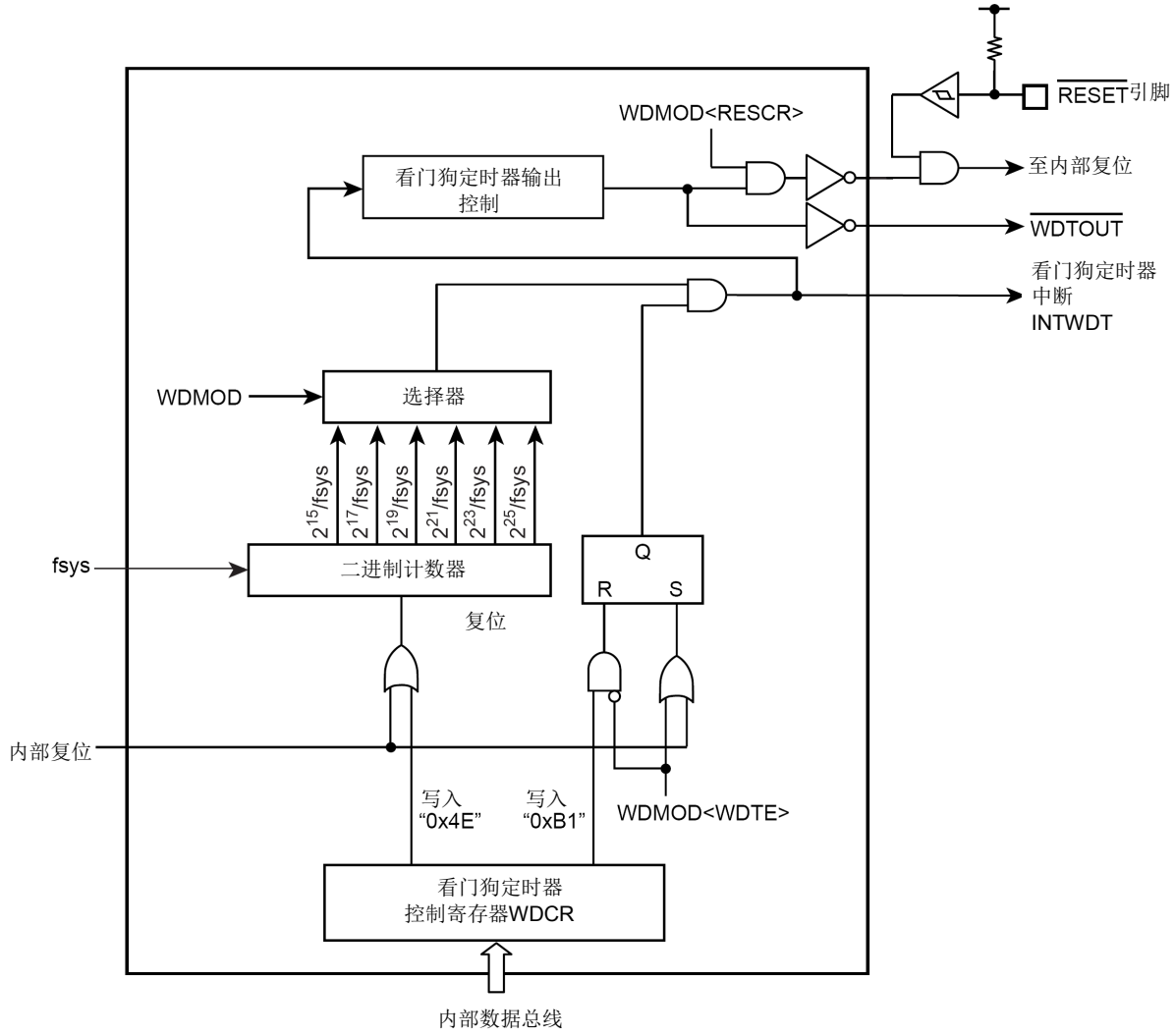


图 17-1 看门狗定时器的方块图

17.2 寄存器

以下给出了各看门狗定时器控制寄存器与地址。

基址 = 0x4004 _ 0000

寄存器名称		地址(基本+)
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

17.2.1 WDMOD(看门狗定时器模式寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	WDTE	WDTP			-	I2WDT	RESCR	-
复位后	1	0	0	0	0	0	1	0

位	比特符号	型号	功能
31-8	−	R	读作 0。
7	WDTE	R/W	启用/禁用控制器 0：禁用 1：启用
6-4	WDTP[2:0]	R/W	选择 WDT 检测时间(参看表 17-1) 000：2^{15}/fsys　100：2^{23}/fsys 001：2^{17}/fsys　101：2^{25}/fsys 010：2^{19}/fsys　110：设置被禁止。 011：2^{21}/fsys　111：设置被禁止。
3	−	R	读作 0。
2	I2WDT	R/W	在 IDLE 模式下运行 0：停止 1：运行中
1	RESCR	R/W	在检测到故障之后运行 0：INTWDT 中断请求生成。(注) 1：复位
0	−	R/W	写入 0。

注： **INTWDT 中断是非屏蔽中断(NMI)的一个因素。**

表 17-1 看门狗定时器的检测时间(fc = 80MHz)

时钟齿轮值 CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.41 ms	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms
100 (fc/2)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
101 (fc/4)	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
110 (fc/8)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
111 (fc/16)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s

17.2.2 WDCR (看门狗定时器控制寄存器)

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

位	比特符号	型号	功能
31-8	−	R	读作 0。
7-0	WDCR	W	禁用/清除代码 0xB1：禁用代码 0x4E：清除代码 其它：保留

17.3 运行

17.3.1 基本运行

看门狗定时器由二进制计数器构成，该二进制计数器在工作时将系统时钟(fsys)作为输入。可由WDMOD<WDTP[2:0]>在 2^{15}，2^{17}，2^{19}，2^{21}，2^{23} 与 2^{25} 之间选择检测时间。在所指定的检测时间到期后，看门狗定时器中断(INTWDT)生成，看门狗定时器输出引脚($\overline{\text{WDTOUT}}$) 输出"低"。

应在 INTWDT 中断生成之前清除该看门狗定时器的二进制计数器，从而可检测到静噪或其它干扰所导致的 CPU 故障(超速)。如果该二进制计数器未被清除，则 INTWDT 可生成非屏蔽中断指令。这样，CPU 检测到故障(超速)，故障对策程序随即被执行，从而恢复正常运行。

另外，通过将该看门狗定时器的输出引脚连接至外围设备的复位引脚，有可能解决 CPU 故障(超速)问题。

注：本产品不带看门狗定时器输出引脚（ $\overline{\text{WDTOUT}}$)。

17.3.2 运行模式与状态

在复位被清除之后，该看门狗定时器随即开始运行。

如未使用该看门狗定时器，则其应被禁用。

在高速频率时钟被停止后，无法使用该看门狗定时器。在推移到以下模式之前，该看门狗定时器应被禁用。在 IDLE 模式下，其运行取决于 WDMOD <I2WDT>设置。

- STOP 模式

此外，在调试期间，该二进制计数器会自动停止运行。

17.4 在检测到故障时的运行

17.4.1 INTWDT 中断发生

图 17-2 给出了 INTWDT 中断生成时的情况(WDMOD<RESCR>="0")。

在二进制计数器发生溢出时，INTWDT 中断生成。其为非屏蔽中断的一个因素。这样，CPU 检测到非屏蔽中断指令，并执行该对策程序。

非屏蔽中断的因素即为该复数。CGNMIFLG 可识别非可屏蔽中断的因素。如果是 INTWDT 中断，则会设置 CGNMIFLG<NMIFLG0>。

在 INTWDT 中断生成的同时，该看门狗定时器输出($\overline{\text{WDTOUT}}$)"低"。通过看门狗定时器清除(即将清除代码 0x4E 写入该 WDCR 寄存器)， $\overline{\text{WDTOUT}}$ 变为"高"。

注：本产品不具备该看门狗定时器输出引脚($\overline{\text{WDTOUT}}$)。

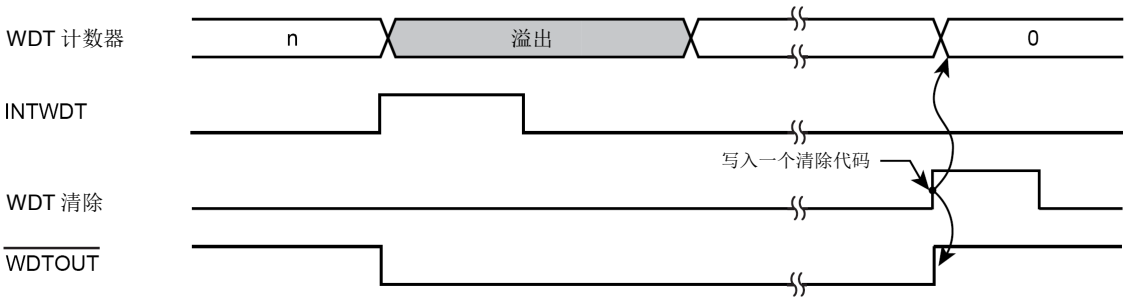


图 17-2 INTWDT 中断发生

17.4.2 内部复位发生

图 17-3 给出了内部复位发生时的情况(WDMOD<RESCR>="1")。

二进制计数器溢出可实现 MCU 的复位。在这种情况下，复位状态持续(适用于 32 状态)。某个时钟被初始化，使得输入时钟(fsys) 与内部高速频率时钟(fosc)相同。这就意味着 fsys = fosc。

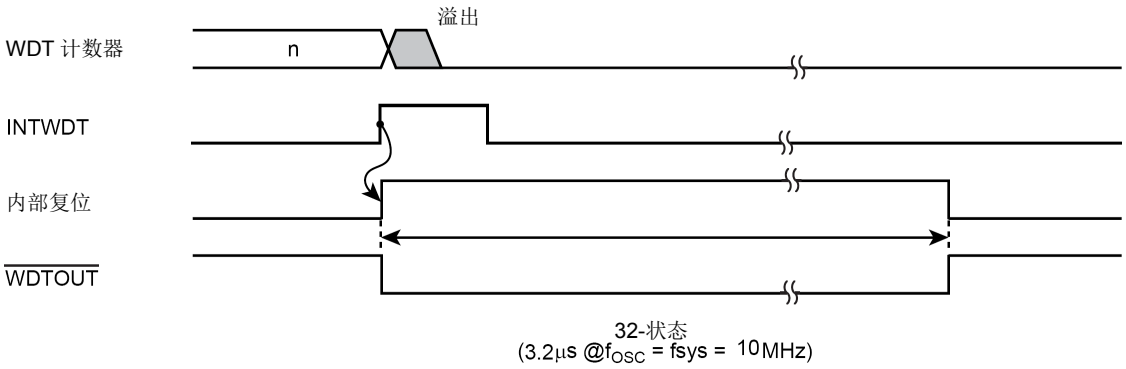


图 17-3 内部复位发生

17.5 控制寄存器

看门狗定时器(WDT) 受控于两个控制寄存器 WDMOD 与 WDCR。

17.5.1 看门狗定时器模式寄存器(WDMOD)

1. 指定看门狗定时器<WDTP[2:0]>的检测时间。

将看门狗定时器检测时间设置为 WDMOD<WDTP[2:0]>。复位后，其被初始化为 WDMOD<WDTP[2:0]> = "000"。

2. 启用/禁用该看门狗定时器<WDTE>。

在复位时，WDMOD <WDTE>被初始化为 "1"，看门狗定时器被启用。

为禁用该看门狗定时器以避免故障所导致的误差写入，可将首个 <WDTE>位设置为"0"，然后必须将该禁用代码(0xB1)写入到 WDCR 寄存器。

将<WDTE>设置为"1"，即可将看门狗定时器的状态从"禁用"改为"启用"。

3. 看门狗定时器输出复位连接<RESCR>

寄存器可指定是否将 WDTOUT 用于内部复位或中断。复位后，WDMOD<RESCR>即被初始化为"1"，并通过二进制计数器的溢出生成内部复位。

17.5.2 看门狗定时器控制寄存器(WDCR)

寄存器用于禁用该看门狗定时器功能，并控制该二进制计数器的清除功能。

17.5.3 设置示例

17.5.3.1 禁用控制

在将 WDMOD <WDTE>设置为"0"之后，通过将该禁用代码(0xB1)写入到该 WDCR 寄存器，该看门狗定时器即可被禁用，且该二进制计数器即被清除。

		7	6	5	4	3	2	1	0	
WDMOD	←	0	–	–	–	–	–	–	–	将<WDTE> 设置为"0"。
WDCR	←	1	0	1	1	0	0	0	1	写入该禁用代码(0xB1)。

17.5.3.2 启用控制

将 WDMOD <WDTE> 设置为"1"。

		7	6	5	4	3	2	1	0	
WDMOD	←	1	–	–	–	–	–	–	–	将<WDTE> 设置为"1"。

17.5.3.3 看门狗定时器清除控制

将清除代码(0x4E) 写入到 WDCR 寄存器，即可清除该二进制计数器，其重新开始计数。

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	写入该清除代码(0x4E)。

17.5.3.4 看门狗定时器的检测时间

如果使用了 2^{21}/fsys，则需将 "011" 设置为 WDMOD<WDTP[2:0]>。

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	1	1	–	–	–	–

18. 运算放大器/模拟比较器(AMP,CMP)

TMPM370FYDFG/FYFG 具备四个运算放大器与模拟比较器。各运算放大器均可放大经由输入端口接收的电压，并将其输出电压馈送到一个 12-位逐次逼近模数(A/D)转换器。这些运算放大器用于放大电机电流测量用分路电阻器之间的电压差。各运算放大器的输出也会被送入一个模拟比较器，并与采自外电路电阻的对应基准电压进行比较。该比较器可向 EMG 逻辑提供异常电流指示。

18.1 配置

图 18-1 给出了各运算放大器/模拟变换器的方块图。

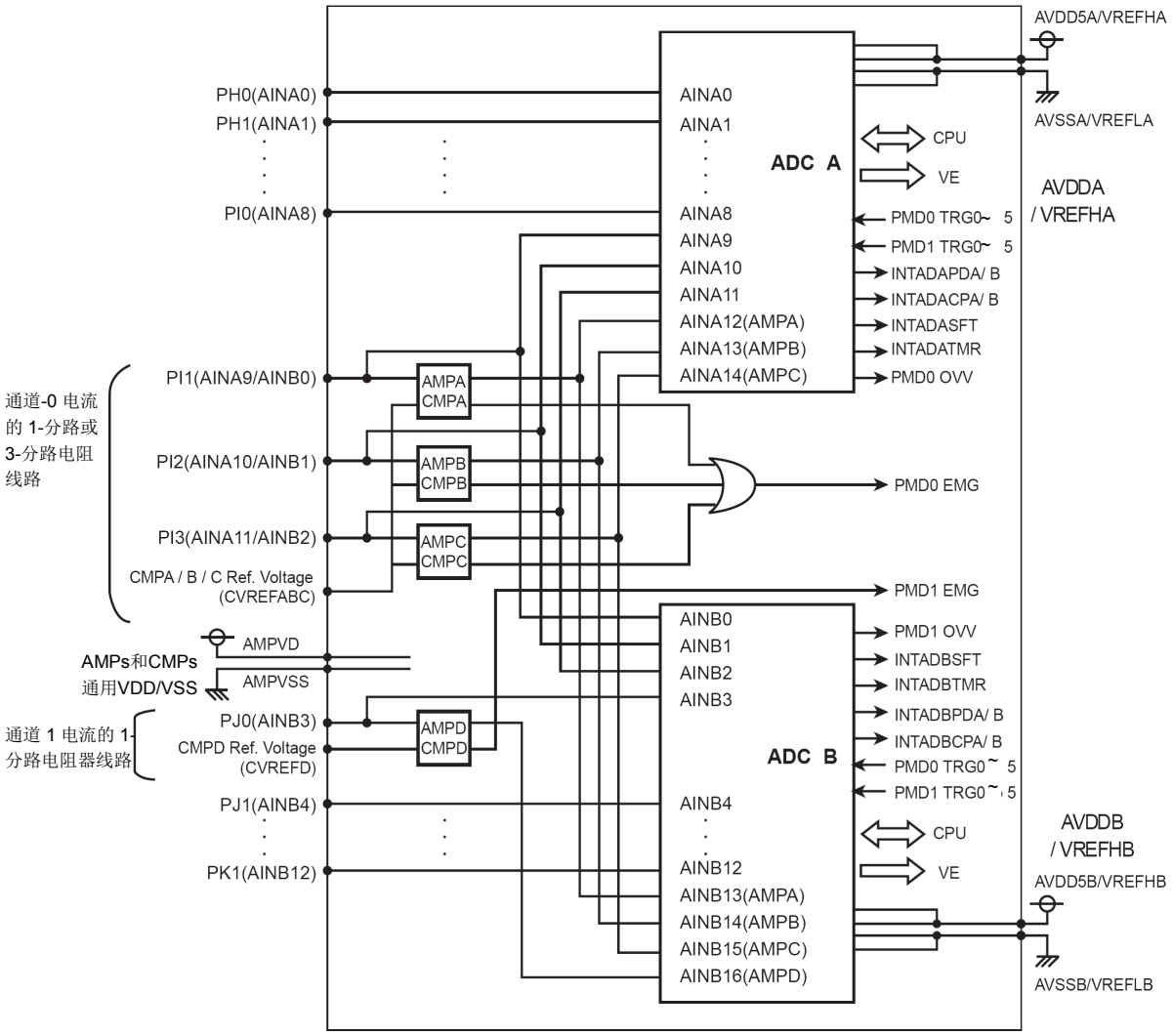


图 18-1 运算放大器/模拟变换器方块图

18.2 寄存器列表

各运算放大器均可通过 AMPCTLA，AMPCTLB，AMPCTLC 与 AMPCTLD 接受单独编程，其允许软件启用与禁用各运算放大器，并选择八种电压增益水平的其中之一。

各比较器均可通过 CMPCTLA，CMPCTLB，CMPCTLC 与 CMPCTLD 接受单独编程，其允许软件启用与禁用各比较器，并选择其输入源(端口输入或运算放大器输出)。

如果使用的是外部运算放大器而不是片上运算放大器，则该片上运算放大器应被禁用(<AMPEN> = 0)，且该比较器应接受组态，以撇开相关运算放大器接收输入端口电压(<CMPSEL> = 0)。

以下对各控制寄存器进行了说明。

18.2.1 运算放大器

基址 = 0x4003_0400

寄存器		地址(基本+)
放大器 A 控制寄存器	AMPCTLA	0x0000
放大器 B 控制寄存器	AMPCTLB	0x0008
放大器 C 控制寄存器	AMPCTLC	0x0010
放大器 D 控制寄存器	AMPCTLD	0x0018

18.2.1.1 AMPCTLA /AMPCTLB /AMPCTLC /AMPCTLD (放大器 A ~ D 控制寄存器)

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

位	比特符号	型号	功能
31-4	−	R	读作 0。
3-1	AMPGLIN[2:0]	R/W	增益选择 000：1.5x　100：4.0x 001：2.5x　101：6.0x 010：3.0x　110：8.0x 011：3.5x　111：10.0x
0	AMPEN	R/W	放大器启用 0：禁用 1：启用

注：在<AMPEN> 被设置为"1"时，该线路的安定约需费时 10μs。

18.2.2 模拟比较器

基址 = 0x4003_0420

寄存器		地址(基本+)
比较器 A 控制寄存器	CMPCTLA	0x0000
比较器 B 控制寄存器	CMPCTLB	0x0008
比较器 C 控制寄存器	CMPCTLC	0x0010
比较器 D 控制寄存器	CMPCTLD	0x0018

18.2.2.1 CMPCTLA /CMPCTLB /CMPCTLC /CMPCTLD (比较器 A ~ D 控制寄存器)

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

位	比特符号	型号	功能
31-2	−	R	读作 0。
1	CMPSEL	R/W	输入选择 0：输入端口 1：运算放大器输出
0	CMPEN	R/W	比较器启用 0：禁用 1：启用

注：在<CMPEN> 被设置为"1"时，该线路的安定约需费时 10 μs。

18.3 运行

18.3.1 基本运行

运算放大器 A，B 与 C (AMP A/B/C)拟用于 3-分路电流检测。来自放大器 A/B/C 的经放大电压被送入两个 A/D 转换器，以允许与电机 U 相，V 相与 W 相对应的三个分路电压的其中两个进行同步转换。

AMP A/B/C 的输入也与各模/数转换器(AINA 9/10/11, AINB13/14/15)直接相连接；因而，即使 AMP A/B/C 被禁用，也可每次将两个分路电压转换为数字值。

运算放大器 D(AMP D)仅支持 1-分路电流检测。来自 AMP D 的经放大电压被送入一个 A/D 转换器(AINB16)。

见图 18-2 所示的运算放大器/模拟变换器方块图。

模拟比较器 A/B/C/D 均与运算放大器 A/B/C/D 连接；因而，各比较器可将放大电压与基准电压进行比较。可通过软件单独禁用各运算放大器；如被禁用，则对应的比较器会将来自某输入端口的分路电压作为输入。

模拟比较器 A/B/C (其设计使之可被连接至一个 3-分路电阻器线路) 带有常用基准电压(CVREFABC)。模拟比较器的设计使之能连接至 1-分路电阻器线路；其带有独立的基准电压(CVREFD)。

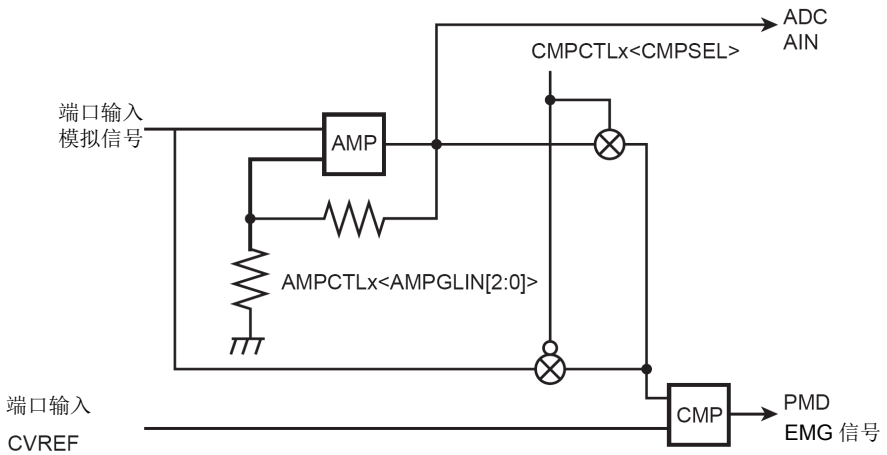


图 18-2 运算放大器/模拟比较器

19. 闪存

本节对闪存存储器的硬件配置与运行进行说明。

19.1 闪存存储器

19.1.1 特点

1. 存储容量

TMPM370FYDFG/FYFG 装有闪存存储器。存储容量与配置如下表所示。

可单独对各存储块进行写入存取。在 CPU 访问内部闪存存储器时，需使用 32-位数据总线宽度。

2. 写入/擦除时间

每个页面均需进行写入。TMPM370FYDFG/FYFG 内含 64 个字。

页面写入需耗时 1.25 ms(典型)不考虑字数。

单个存储块的擦除需耗时 0.1 秒(典型)。

下表给出了每片的写入与擦除时间。

产品名称	存储容量	存储块配置				字#	写入时间	擦除时间
		128 KB	64 KB	32 KB	16 KB			
TMPM370FYDFG / FYFG	256 KB	0	3	2	2	64	1.28 秒	0.4 秒

注：**以上各值均为理论值，未包括数据传送时间。每片的写入时间取决于用户拟采用的写入方式。**

3. 编程方式

有两种板上编程模式，可供用户在设备已安装在用户板上的情况下，对该设备进行编程(重写)：

a. 用户引导模式

可支持该应用的原重写方式。

b. 单一引导模式

可支持采用串行数据传送(东芝的独特方式)的重写方式。

4. 重写方式

除某些特定功能之外，本设备所带的闪存存储器一般可满足适用 JEDEC 标准的规定。因此，如果用户当前使用的是外部闪存存储器设备，则易于在该设备内部实现该功能。此外，用户无需构建自身程序即可实现复杂的写入与擦除功能，原因是可利用已内置到该闪存存储器芯片上的各电路自动执行该类功能。

符合 JEDEC 规定的功能	已修改，增加或删除的功能
•自动编程 •自动芯片擦除 •自动存储块擦除 •数据轮询/切换位	<已修改> 存储块保护 (仅支持软件保护) <已删除> 擦除继续执行 - 暂停功能

5. 保护/安全功能

也可对该设备执行读取保护功能，以禁止读取来自任何外部编写程序设备的闪存存储器数据。另一方面，重写保护仅在通过基于命令的软件进行编程时可用；不支持拟采用+12VDC 的任何硬件设置方式。有关 ROM 保护与安全功能的详细资料，见"ROM 保护"一节。

注： **如果密码被设置为 0xFF (已擦除数据)，则可因密码易于被猜出而导致难以妥善保护数据。即使未使用单一引导模式，也建议将某个唯一值设置为密码。**

19.1.2 闪存存储器部分方块图

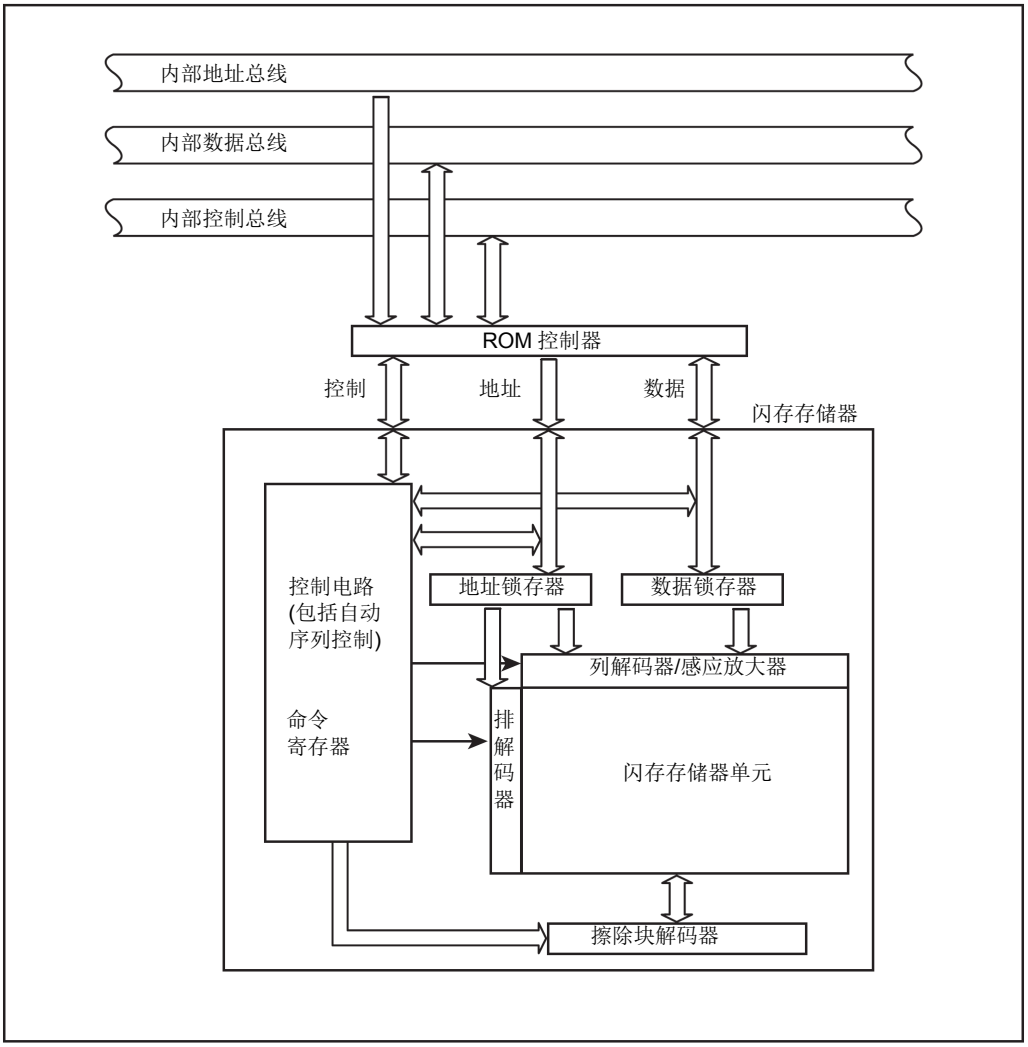


图 19-1 闪存存储器部分方块图

19.2 运行模式

该设备有三种运行模式包括不使用内部闪存存储器的模式。

表 19-1 运行模式

运行模式	运行详述
单片模式 标准模式	在复位被清除之后，其从内部闪存存储器启动。 在该运行模式下，可定义两种不同的模式即拟执行应用程序的模式，以及拟重写用户设置板上闪存存储器的模式。前者被称为"标准模式"，而后者则被称为"用户引导模式"。
用户引导模式	用户可以独特方式配置该系统，使之在这两种模式之间切换。例如，用户可自由设计该系统，使得在端口"a0"被设置为"1"时可选择该标准模式，而在其即被设置为"0"可选择用户引导模式。用户应编制例行程序并将其作为应用程序的一部分，然后就各模式的选择作出决定。
单一引导模式	在复位被清除之后，其从内部内部引导 ROM(蔽屏 ROM)启动。可在该引导 ROM 中编制一种算法，用于通过该设备的串行口启用对用户设置闪存存储器的重写。经由该串行口连接外部主计算机，通过按预先规定的协议输送数据，即可对该内部闪存存储器进行编程。

在上表所列的各闪存存储器运行模式中，该用户引导模式与该单一引导模式属于可编程模式。用户引导模式与单一引导模式这两种模式，被称为"板上编程"模式，其中，可对用户设置进行内部闪存存储器的板上重写。

在设备处于复位状态时，通过以外部方式设置该 $\overline{\text{BOOT}}$ (PF0)引脚的电平，即可选择单片或单一引导运行模式。

表 19-2 运行模式设置

运行模式	引脚	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PF0)
单片模式	0 → 1	1
单一引导模式	0 → 1	0

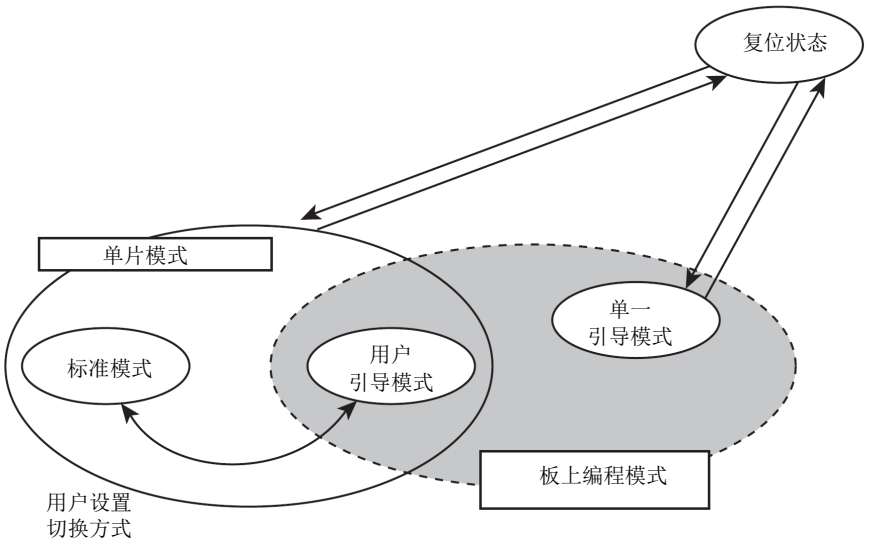


图 19-2 模式推移图

19.2.1 复位操作

在复位该设备时，应确保供电电压不超出工作电压范围，内部振荡器已经稳定，且该 $\overline{\text{RESET}}$ 输入被保持为"0"至少持续 12 系统时钟(0.15 μs，80 MHz 运行；复位后，该"1/1"时钟齿轮模式即被应用)。

注 1：**需在上电时即将"0"应用于该复位输入，且最小持续时间应为 700 μs 不考虑运行频率。**

注 2：**在闪存自动编程或擦除的进行期间，复位周期至少应为 0.5 μs 不考虑系统时钟频率。在这种情况下，复位后启用读取需耗时约 2 ms。**

19.2.2 用户引导模式(单片模式)

用户引导模式应使用用户所定义的闪存存储器编程例行程序。在旧应用上的闪存存储器程序代码数据传送总线与串行 I/O 的数据传送总线不同时，就会使用该程序。其在该单片模式下运行；因此，必须从标准模式在该模式下，用户应用按单片模式激活切换到闪存编程所需的用户引导模式。具体地说，就是将一个模式判断例行程序添加到该用户应用中的某个复位程序。

需依照用户的系统设定状态，利用 TMPM370FYDFG/FYFG 的 I/O 设置模式切换条件。至于用户以独特方式编制的闪存存储器编程例行程序，也需要在新的应用中进行设置。该例行程序用于在切换到用户引导模式后进行编程。必须在该编程例行程序被存储在闪存存储器以外的区域期间执行该编程例行程序，原因是在删除/写入模式期间，内部闪存存储器中的数据无法读出。建议一旦完成重新编程，即保护相关闪存块，已免其在随后的单片(标准模式)运行期间发生意外破坏。不得在用户引导模式期间引发任何异常包括非屏蔽中断。

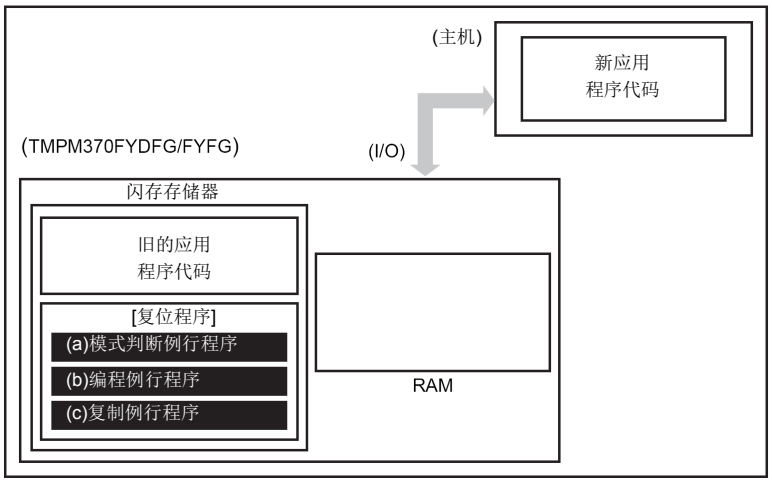
(1-A)与(1-B)为利用内部闪存存储器与外存储器中的例行程序进行编程的示例。有关擦除与程序序列的详细说明，请参看"19.3 闪存存储器的板上编程(重写/擦除)"。

19.2.2.1 (1-A)方法 1：将编程例行程序存储到闪存存储器中

(1) 步骤-1

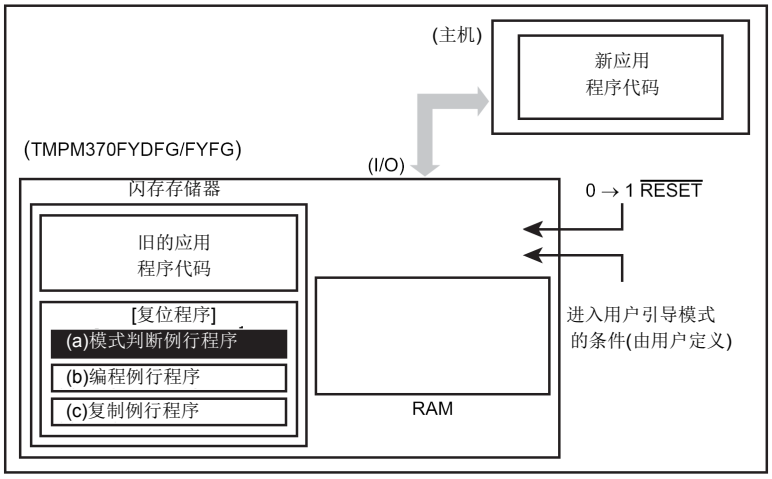
确定闪存存储器进入用户引导模式所需的条件(例如：引脚状态)，以及拟用于传送新的程序代码的 I/O 总线。相应地创建硬件和软件。在将 TMPM370FYDFG/FYFG 安装到印刷电路板上之前，应利用编程设备将以下例行程序写入到某随机闪存块。

(a)模式判断例行程序：	据以确定是否切换到用户引导模式的代码
(b)编程例行程序：	据以从主控制器下载新的程序代码，并重新编程该闪存存储器的代码
(c)复制例行程序：	据以将(b)中所述的数据从 TMPM370FYDFG/FYFG 闪存存储器复制到 TMPM370FYDFG/FYFG 片上 RAM 或外部存储器设备的代码。



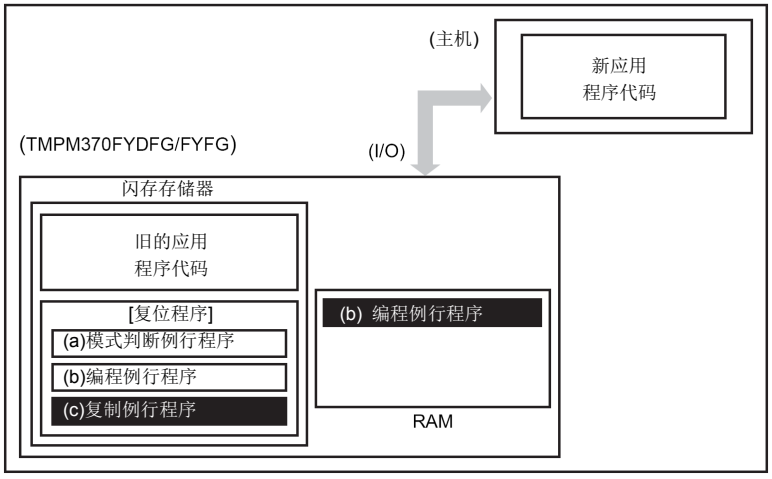
(2) 步骤-2

以下描述适用于编程例行程序被安装在复位处理程序中的情形。在 $\overline{\text{RESET}}$ 引脚被解除之后，该复位程序可确定是否将 TMPM370FYDFG/FYFG 置入用户引导模式。如果满足模式切换条件，则该闪存存储器进入用户引导模式(在用户引导模式期间，不得使用所有中断包括 NMI)。



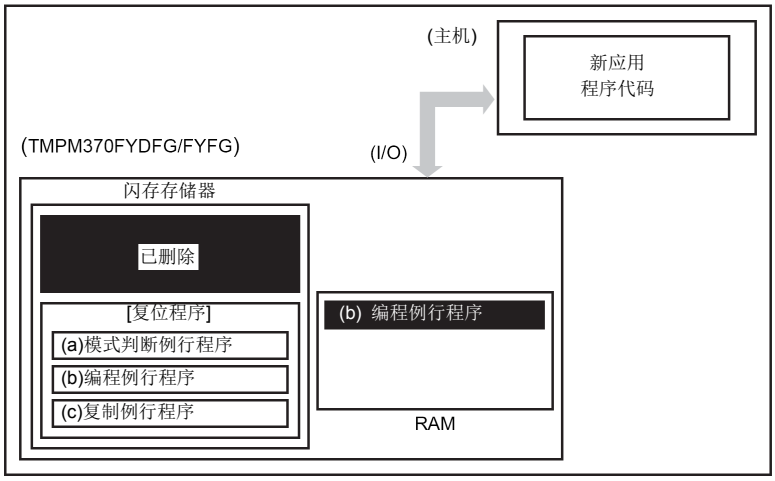
(3) 步骤-3

一旦发生向用户引导模式的推移，即执行该复制例行程序(c)，以将该闪存编程例行程序(b)复制到 TMPM370FYDFG/FYFG 片上 RAM。



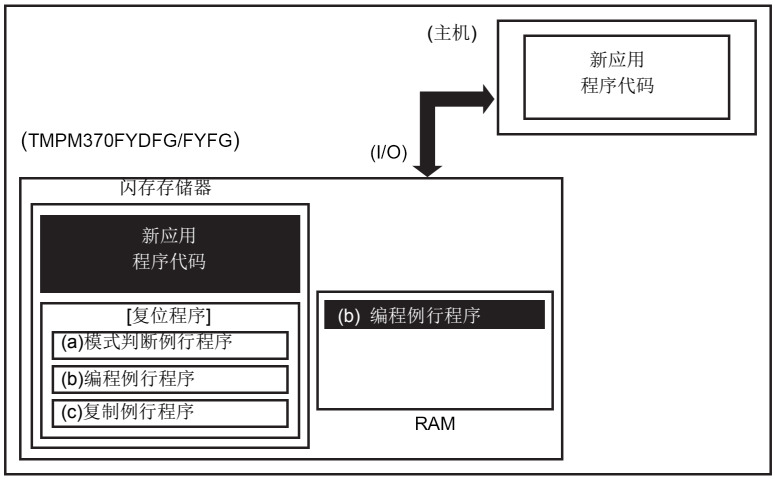
(4) 步骤-4

将程序执行转到片上 RAM 中的闪存编程例行程序，以清除写入或擦除保护，并擦除含有旧应用程序代码的闪存块。



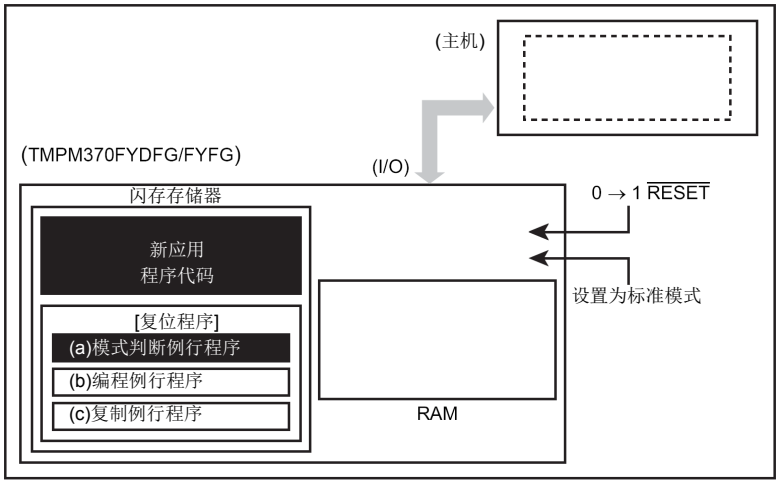
(5) 步骤-5

继续执行该闪存编程例行程序，以从主控制器下载新的程序代码，并将其编制到该已擦除的闪存块内。在编程完成时，必须设置用户程序区内闪存块的写入或擦除保护。



(6) 步骤-6

将 $\overline{\text{RESET}}$ 设置为 "0"，即可复位该 TMPM370FYDFG/FYFG。一旦复位，该片上闪存存储器即被设置为标准模式。在 $\overline{\text{RESET}}$ 被解除之后，CPU 可开始执行新的应用程序代码。



19.2.2.2 (1-B)方法 2：从外部主控制器传送编程例行程序

(1) 步骤-1

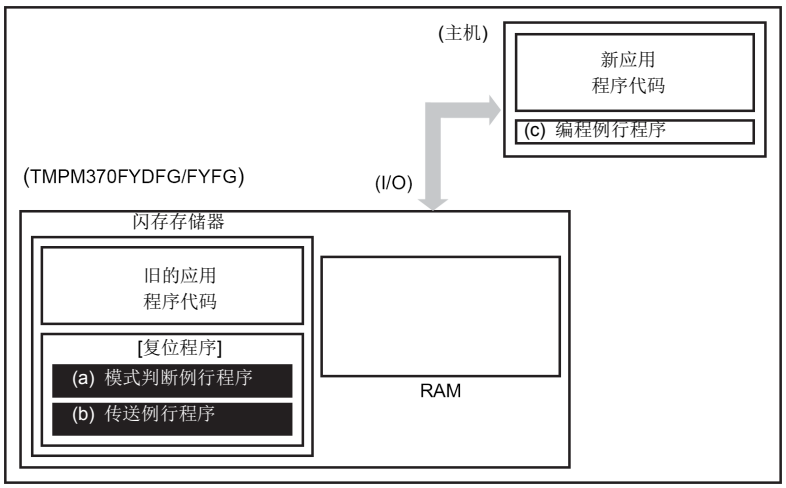
确定闪存存储器进入用户引导模式所需的条件(例如：引脚状态)，以及拟用于传送新的程序代码的 I/O 总线。相应地创建硬件和软件。在将 TMPM370FYDFG/FYFG 安装到印刷电路板上之前，应利用编程设备将以下例行程序写入到某随机闪存块。

(a) 模式判断例行程序： 据以确定是否切换到用户引导模式的代码

(b)传送例行程序： 据以从主控制器下载新的程序代码的代码

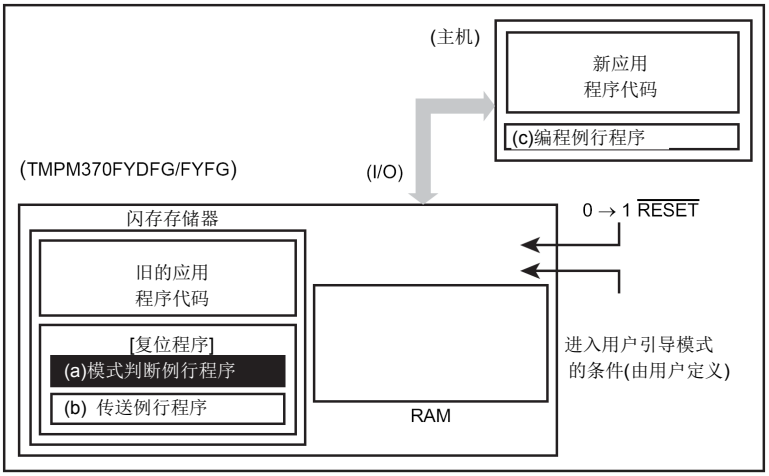
还需在主控制器上编制以下所示的编程例行程序：

(c)编程例行程序： 据以从外部主控制器下载新的程序代码，并重新编程该闪存存储器的代码



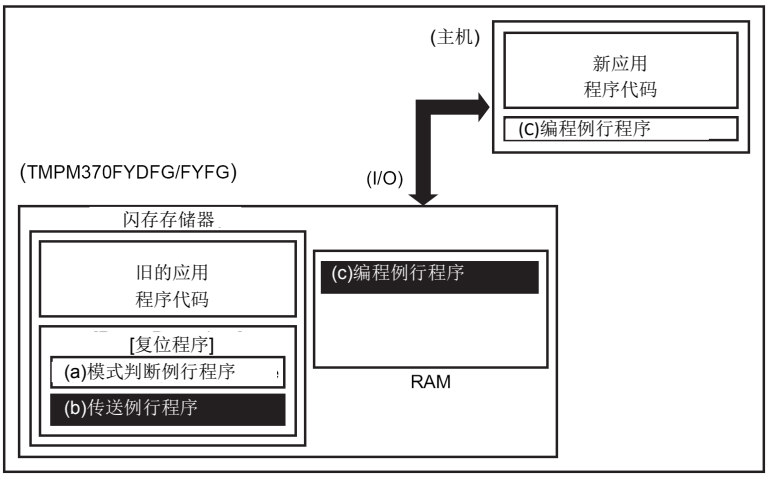
(2) 步骤-2

以下描述适用于编程例行程序被安装在复位处理程序中的情形。在 RESET 被解除之后，该复位程序可确定是否将 TMPM370FYDFG/FYFG 置入用户引导模式。如果满足模式切换条件，则该闪存存储器进入用户引导模式(在用户引导模式期间，不得使用所有中断包括 NMI)。



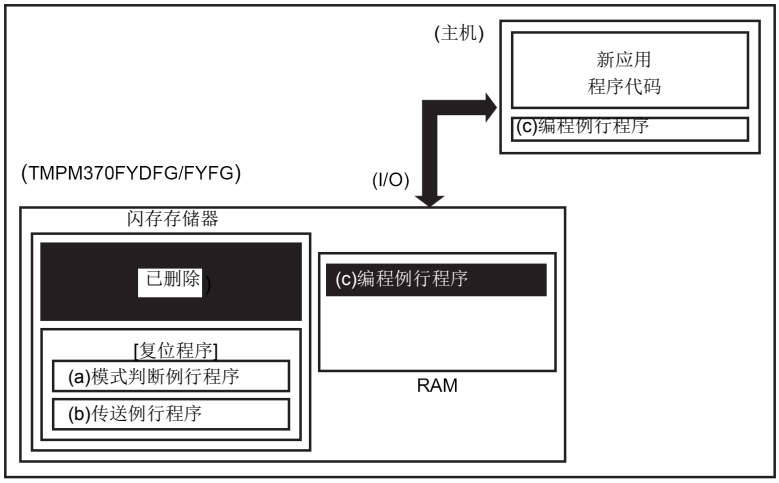
(3) 步骤-3

一旦进入用户引导模式，即执行该传送例行程序(b)，以将该闪存编程例行程序(c)从该主控制器下载到 TMPM370FYDFG/FYFG 片上 RAM。



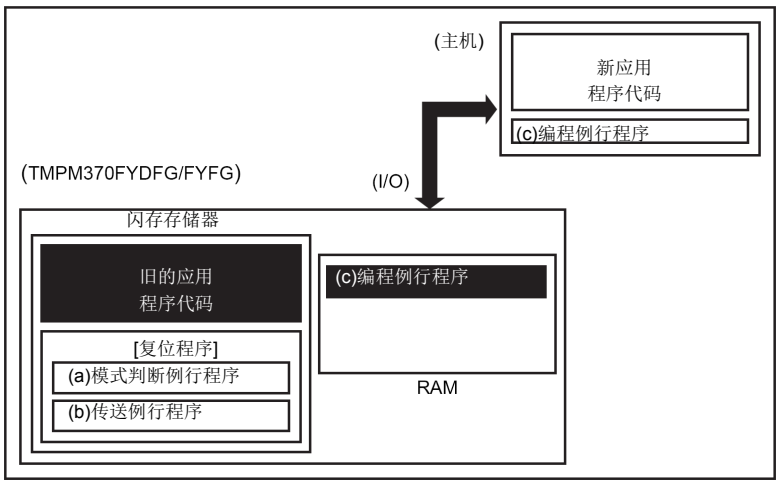
(4) 步骤-4

将程序执行转到片上 RAM 中的闪存编程例行程序，以清除写入或擦除保护，并擦除含有旧应用程序代码的闪存块。



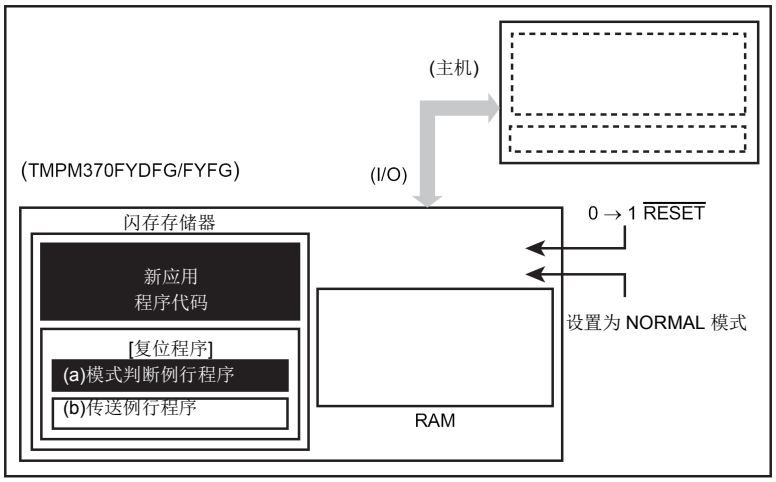
(5) 步骤-5

继续执行该闪存编程例行程序，以从主控制器下载新的程序代码，并将其编制到该已擦除的闪存块内。在编程完成时，必须设置用户程序区内闪存块的写入或擦除保护。



(6) 步骤-6

将 $\overline{\text{RESET}}$ 设置为"0"低，即可复位该 TMPM370FYDFG/FYFG。一旦复位，该片上闪存存储器即被设置为标准模式。 在 $\overline{\text{RESET}}$ 被解除之后，CPU 可开始执行新的应用程序代码。



19.2.3 单一引导模式

在单一引导模式下，利用 TMPM370FYDFG/FYFG 片上引导 ROM 中所包含的某程序，即可对该闪存存储器进行重新编程。该引导 ROM 是一个带屏蔽的 ROM。如果复位后即选择单一引导模式，该引导 ROM 即被映射到该地址区域包括中断向量表，而闪存存储器则会被映射到与其不同的某个地址区域。

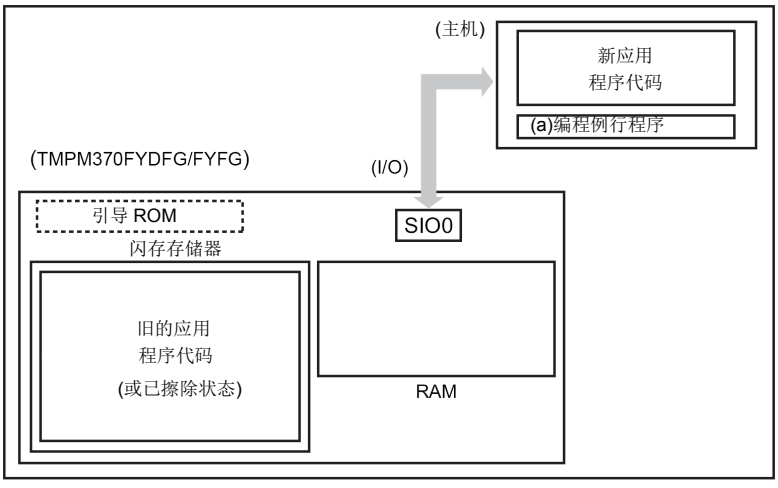
单一引导模式允许进行闪存存储器串行编程。TMPM370FYDFG/FYFG 的 SIO (SIO0) 的通道 0 被连接至某外部主控制器。通过该串行链路，即可将编程例行程序从该主控制器下载到 TMPM370FYDFG/FYFG 片上 RAM。然后，通过执行该编程例行程序，该闪存存储器即被重新编程。该主控制器会发送命令与编程数据，用于该闪存存储器的重新编程。SIO0 与该主控制器之间的通信必须遵循后文所述的协议。为保护该闪存存储器的内容，在编程例行程序被下载到该片上 RAM 中之前，会对该应用的密码的有效性进行验证。如果密码匹配失败，编程例行程序的传送即被异常终止。原因是在用户引导模式时，在闪存存储器正在被擦除或编程时，所有中断包括非屏蔽中断(NMI)在单一引导模式下均必须被禁用。在单一引导模式下，该引导 ROM 程序会按 NORMAL 模式执行。

建议一旦完成重新编程，即将该写入/擦除保护设置到相关闪存块，已免其在随后的单片(标准模式)运行期间发生意外讹误。

19.2.3.1 (2-A)使用片上引导 ROM 中的程序

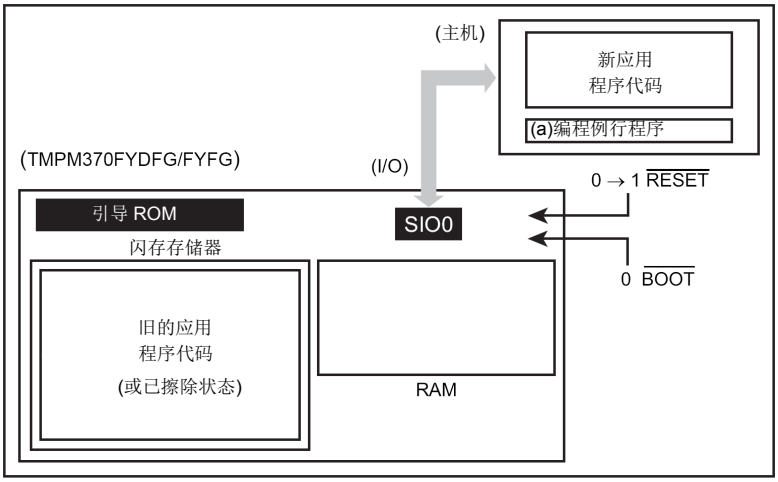
(1)步骤-1

在执行该编程例行程序之前，无需擦除内含该程序代码旧版本的闪存块。由于需经由该 SIO (SIO0)传送编程例行程序与编程数据，必须将该 SIO0 连接至某主控制器。需在该主控制器上编制一个编程例行程序(a)。



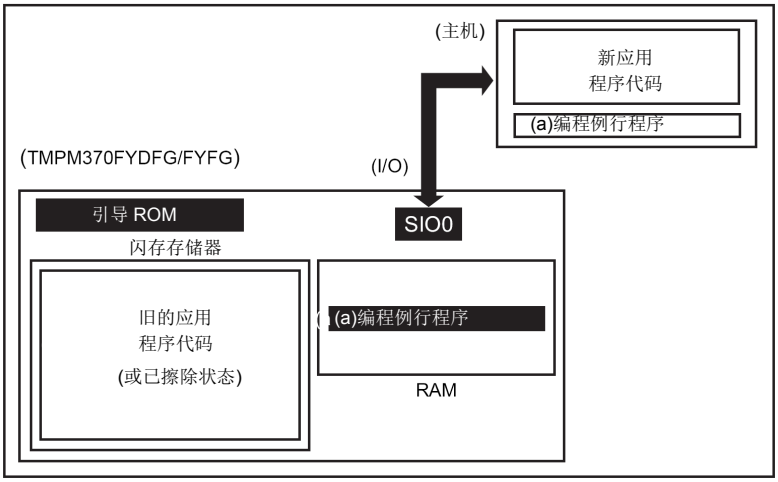
(2) 步骤-2

在 $\overline{\text{BOOT}}$ 引脚已被设置为"0"时，将该 $\overline{\text{RESET}}$ 引脚设置为 "1"，以取消 TMPM370FYDFG/FYFG 的复位。在复位之后，CPU 从片上引导 ROM 重新启动。首先会将经由 SIO0 从主控制器传送的该 12-字节密码，与该专用闪存存储器存储单元进行比较(如果该闪存块已被擦除，则该密码为 0xFF)。



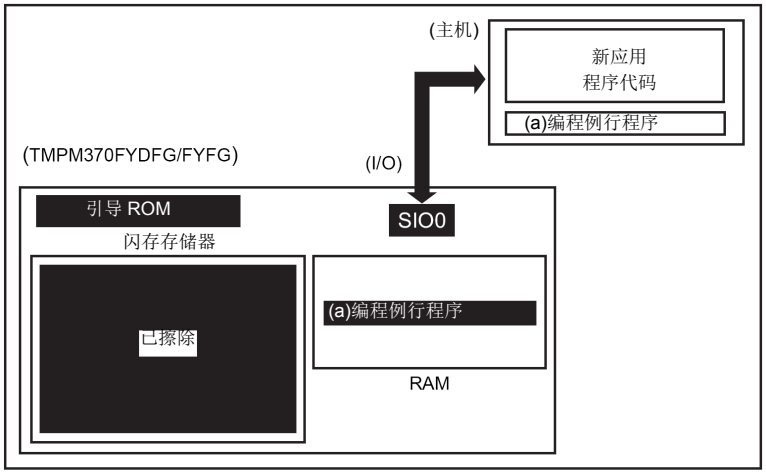
(3) 步骤-3

如果该密码是正确的，则该引导程序会将该编程例行程序(a)从主控制器下载到 TMPM370FYDFG/FYFG 的该片上 RAM。必须将该编程例行程序存储在从 0x2000_0400 到 RAM 结束地址之间的范围内。



(4) 步骤-4

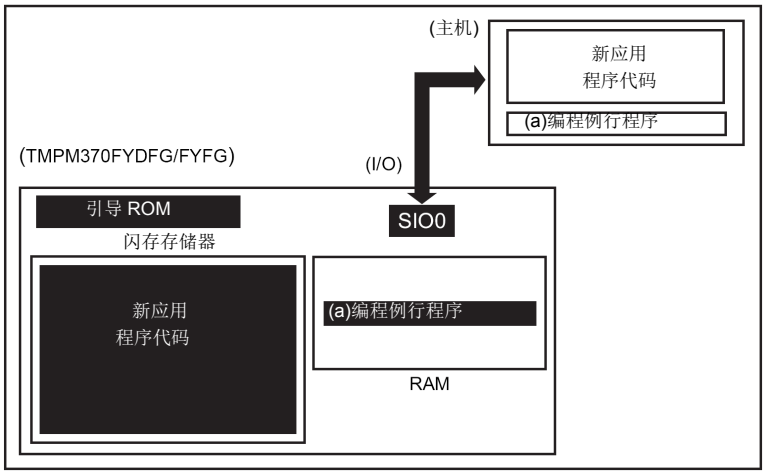
CPU 转到片上 RAM 中的闪存编程例行程序(a)，以擦除含有旧应用程序代码的闪存块。可使用该存储块擦除或片上擦除命令。



(5) 步骤-5

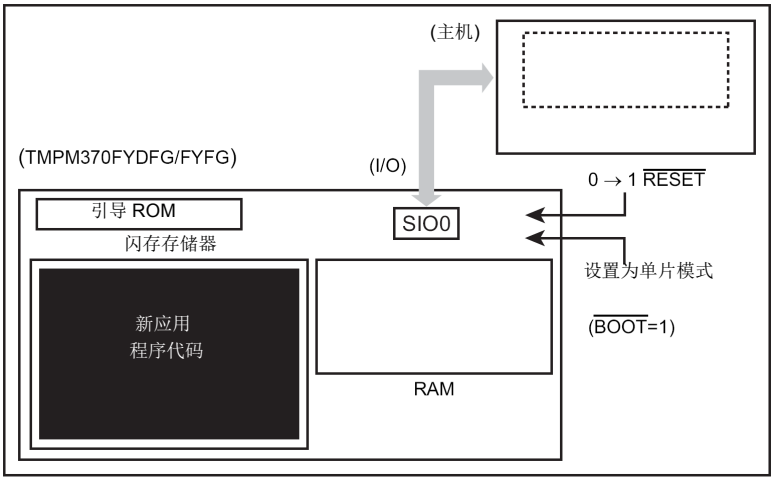
然后，该编程例行程序(a)从主控制器下载新的程序代码，并将其编制到该已擦除的闪存块内。在编程完成时，必须设置用户程序区内闪存块的写入或擦除保护。

在以下示例中，新的程序代码经由同一 SIO0 来自同一主控制器通道，用于该编程例行程序。不过，一旦该编程例行程序已开始在该片上 RAM 中执行，则可任意更改该传送路径与传送源。创建板硬件与编程例行程序，以满足特殊需求。



(6) 步骤-6

在该闪存存储器的编程完成时，断开该板电源，并断开主控制器与目标板之间的电缆。重新通电，使 TMPM370FYDFG/FYFG 在单片(标准)模式下重新引导以执行该新程序。



19.2.4 单一引导模式的配置

在执行该板上编程时，需按以下所示的配置，用单一引导模式引导该 TMPM370FYDFG/FYFG。

$\overline{BOOT}$(PF0) = 0
$\overline{RESET}$ = 0 → 1

将 $\overline{RESET}$ 输入设置为"0"，并将各 $\overline{BOOT}$ (PF0)引脚设置为以上所示的值，然后解除 $\overline{RESET}$ 引脚(高)。

19.2.5 存储器映射

图 19-3 给出了标准与单一引导模式下存储器映射的比较结果。在单一引导模式下，该内部闪存存储器会被映射到 0x3F80_0000 与后面的地址，而内部引导 ROM(屏蔽 ROM)则被映射到 0x0000_0000 ~ 0x0000_0FFF。

内部闪存存储器与各设备的 RAM 地址给出如下。

产品名称	闪存规格	RAM 规格	闪存地址 (单片/单一引导模式)	RAM 地址
TMPM370FYDFG/ FYFG	256 KB	10 KB	0x0000_0000 ~ 0x0003_FFFF 0x3F80_0000 ~ 0x3F83_FFFF	0x2000_0000 ~ 0x2000_27FF

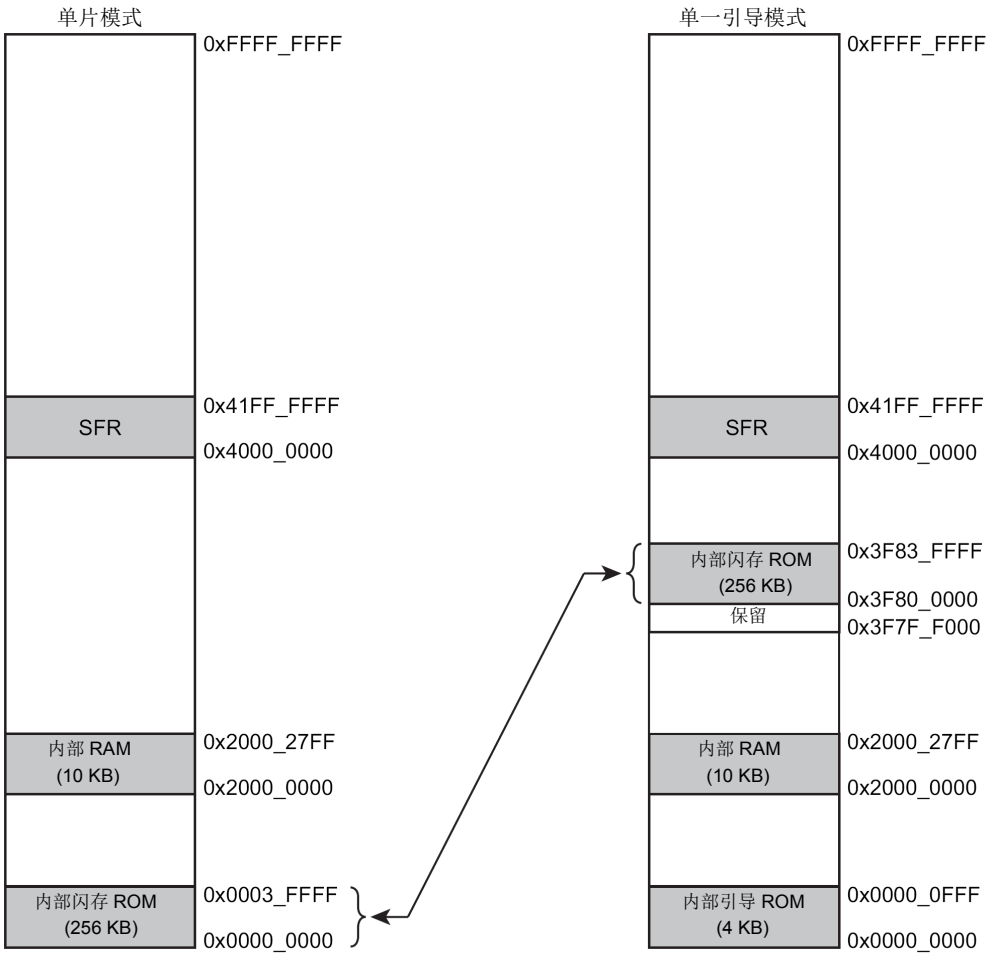


图 19-3 TMPM370FYDFG/FYFG 的存储器映射

19.2.6 接口规格

在单一引导模式下，SIO 通道用于与程序控制器通信。相同的配置被用于程序控制器的通信格式，用以进行板上编程。可同时支持 UART (异步)与 I/O 接口(同步)模式。各通信格式给出如下。

- UART 通信

  通信通道：SIO 通道 0

  串行传送模式：UART (异步)模式，半双工，先 LSB

  数据长度：8 位

  校验位：无

  STOP 位：1 位

  波特率：随机波特率

- I/O 接口模式

  通信通道：SIO 通道 0

  串行传送模式：I/O 接口模式，全双工，先 LSB

  同步时钟脉冲(SCLK0) ：输入模式

  握手信号：PE4 配置按输出模式

  波特率：随机波特率

表 19-3 所需引脚连接

引脚		接口	
		UART	I/O 接口模式
电源引脚	DVDD5	o	o
	DVDD5E	o	o
	DVSS	o	o
	AVDD5A	o	o
	AVSSA	o	o
	AVDD5B	o	o
	AVSSB	o	o
	VOUT3	o	o
	VOUT15	o	o
	RVDD5	o	o
模式设置引脚	$\overline{\text{BOOT}}$ (PF0)	o	o
复位引脚	$\overline{\text{RESET}}$	o	o
通信引脚	TXD0 (PE0)	o	o
	RXD0 (PE1)	o	o
	SCLK0 (PE2)	×	o (输入模式)
	PE4	×	o (输出模式)

19.2.7 数据传送格式

在表 19-4，表 19-6 到表 19-7 中，对各运行模式下的运行命令与数据传送格式进行了说明。除本节外，还请参看"19.2.10 引导程序的运行"一节。

表 19-4 单一引导模式命令

代码	命令
0x10	RAM 传送
0x40	芯片与保护位擦除

19.2.8 对内部存储器的限制

单一引导模式可设置针对内部 RAM 与 ROM 的限制性条件，如表 19-5 所示。

表 19-5 单一引导模式下的限制性条件

存储器	细节
内部 RAM	引导 ROM 中的程序将该区域从 0x2000_0000 ~ 0x2000_03FF 用作工作区。 将该 RAM 传送程序存储在从 0x2000_0400 至 RAM 结束地址的范围内。
内部 ROM	以下地址被指定用于存储软件 ID 信息与口令。 不建议将程序存储在这些地址内。 0x3F83_FFF0 ~0x3F83_FFFF

19.2.9 引导程序的传送格式

下表给出了各引导程序命令的传送格式。除本节外，还请参看"19.2.10 引导程序的运行"一节。

19.2.9.1 RAM 传送

表 19-6 RAM 传送命令的传送格式

	字节	被从控制器传送到 TMPM370FYDFG/FYFG 的数据	波特率	被从 TMPM370FYDFG/FYFG 传送到控制器的数据
引导 ROM	1 字节	串行操作模式与波特率 对于 UART 模式：0x86 对于 I/O 接口模式：0x30	所需的波特率 (注 1)	−
	2 字节	−		串行操作模式字节 ACK • 对于 UART 模式 - 标准确认：0x86 (如果不能正确设置波特率，则引导程序异常终止。) •对于 I/O 接口模式 - 标准确认：0x30
	3 字节	命令代码(0x10)		−
	4 字节	−		命令代码字节 ACK(注 2) - 标准确认：0x10 - 否定确认: 0xX1 - 通信错误：0xX8
	5 字节 ~ 16 字节	密码序列(12 字节) 0x3F83_FFF4~ 0x3F83_FFFF		−
	17 字节	字节 5 ~ 16 的校验和值		−
	18 字节	−		检验数字字节 ACK(注 2) - 标准确认：0x10 - 否定确认：0xX1 - 通信错误：0xX8
	19 字节	RAM 存储起始地址 31 ~ 24		−
	20 字节	RAM 存储起始地址 23 ~ 16		−
	21 字节	RAM 存储起始地址 15 ~ 8		−
	22 字节	RAM 存储起始地址 7 ~ 0		−
	23 字节	RAM 存储起始地址 15 ~ 8		−
	24 字节	RAM 存储起始地址 7 ~ 0		−
	25 字节	字节 19 ~ 24 的校验和值		−
	26 字节	−		检验数字字节 ACK(注 2) - 标准确认：0x10 - 否定确认：0xX1 - 通信错误：0xX8
	27 字节~ m 字节	RAM 存储数据		−
	m+ 1 字节	字节 27~ m 的校验和值		−
	m+ 2 字节	−		检验数字字节 ACK(注 2) - 标准确认：0x10 - 否定确认：0xX1 - 通信错误：0xX8
RAM	m+ 3 字节	−		转移至 RAM 存储起始地址

注 1：**在 I/O 接口模式下，第 1 字节与第 2 字节的传送波特率必须为所需波特率的 1/16。**

注 2：**如果是任何否定确认，则该引导程序会返回到等待命令代码的状态(第 3 字节)。在 I/O 接口模式下，如果发生通信错误，则不会出现否定确认。**

注 3：**第 19 字节~ 第 25 字节必须在从 0x2000_0400 到 RAM 结束地址之间的 RAM 地址范围内。**

19.2.9.2 芯片擦除与保护位擦除

表 19-7 芯片与保护位擦除命令的传送格式

	字节	被从控制器传送到 TMPM370FYDFG/FYFG 的数据	波特率	被从 TMPM370FYDFG/FYFG 传送到控制器的数据
引导 ROM	1 字节	串行操作模式与波特率 For UART mode : 0x86 对于 I/O 接口模式 0x30	所需的波特率 (注 1)	–
	2 字节	–		串行操作模式字节 ACK • 对于 UART 模式 - 标准确认：0x86 (如果不能正确设置波特率，则引导程序异常终止。) • 对于 I/O 接口模式 - 标准确认：0x30
	3 字节	命令代码(0x40)		–
	4 字节	–		命令代码字节 ACK(注 2) - 标准确认： 0x40 - 否定确认： 0xX1 - 通信错误：0xX8
	5 字节	芯片擦除命令代码 (0x54)		–
	6 字节	–		命令代码字节 ACK(注 2) - 标准确认： 0x40 - 否定确认： 0xX1 - 通信错误：0xX8
	7 字节	–		芯片擦除命令代码 ACK - 标准确认： 0x4F - 否定确认： 0x4C
	8 字节	(等待下一个命令代码)		–

注 1：**在 I/O 接口模式下，第 1 字节与第 2 字节的传送波特率必须为所需波特率的 1/16。**

注 2：**如果是任何否定确认，则该引导程序会返回到等待命令代码的状态(第 3 字节)。在 I/O 接口模式下，如果发生通信错误，则不会出现否定确认。**

19.2.10 引导程序的运行

在已选择单一引导模式的情况下，该引导程序会在起动时自动执行。该引导程序提供这四条命令，其详细说明见以下各小节。

1. RAM 传送命令

该 RAM 传送命令可将从主控制器传送过来的程序代码存储到片上 RAM，且一旦传送成功完成即执行该程序。用户程序 RAM 空间可被指定到从 0x2000_0400 至 RAM 结束地址之间的范围内，但是该引导程序区(0x2000_0000~ 0x2000_03FF)不可用。该用户程序可在指定 RAM 地址启动。

该 RAM 传送命令可用于下载自有的闪存编程例行程序；这样就能够以独特的方式控制闪存存储器的板上编程。该编程例行程序必须利用第 19.3 节所述的闪存存储器命令序列。在初始化某一传送之前，该 RAM 传送命令会对照存储在闪存存储器中的密码序列，对来自该控制器的密码序列进行校验。

注： **如果密码被设置为 0xFF (已擦除数据)，则可因密码易于被猜出而导致难以妥善保护数据。即使未使用单一引导模式，也建议将某个唯一值设置为密码。**

2. 显示闪存存储器 SUM 命令

利用该显示闪存存储器 SUM 命令，可同时添加该闪存存储器的全部内容。该引导程序未提供用以读出闪存存储器内容的命令。作为替代，闪存存储器 SUM 命令可用于软件版本管理。

3. 显示产品信息命令

该显示产品信息命令可提供产品名称，片上存储器配置等信息。该命令还可读出下述地址闪存存储器单元的内容。除显示闪存存储器求和命令之外，这些单元还可用于软件版本管理。

产品名称	区域
TMPM370FYDFG/ FYFG	0x3F83_FFF0 ~0x3F83_FFF3

4. 闪存存储器芯片擦除与保护位擦除命令

该命令可自动擦除闪存存储器整个区域。即使在任何存储块被禁止写入与擦除的情况下，该存储单元中的所有存储块及其保护条件仍然会被擦除。在该命令完成时，该 FCSECBIT <SECBIT> 位即被设置为"1"。该命令可在用户忘记密码时恢复引导编程运行。因此，不会执行密码验证。

19.2.10.1 RAM 传送命令

有关该命令的传送格式见表 19-6。

1. 第 1 字节可指定使用两个串行操作模式中的其中一个。有关串行操作模式确定方式的详细说明，见后文的"19.2.10.4 串行操作模式的确定"。如该模式被确定为 UART 模式，则该引导程序会检查波特率设置是否被执行。在第 1 字节处理期间，接收运行被禁止(SC0MOD0<RXE>=0)。

   • 在 UART 模式下进行通信

   第 1 字节被设置为"0x86" ，并通过设置 UART，被以规定的波特率从该控制器传输到目标板。如该串行操作模式被确定为 UART 模式，则该引导程序会检查波特率设置是否被执行。如果无法设置该波特率，则该引导程序异常终止，任何后续通信均无法进行。有关判定波特率的设置是否可进行的方法，请参看"波特率设置"。

   • 在 I/O 接口模式下进行通信

   第 1 字节被设置为"0x30" ，并通过同步设置，被以所需的波特率的 1/16 从该控制器传输到目标板。与第 1 字节一样，第 2 字节传输时也按所规定波特率的 1/16 进行。从第 3 字节(运行命令数据)开始，用户可按规定波特率传输数据。

   在 I/O 接口模式下，CPU 认为该接收端子应为输入端口，且其应负责监视 I/O 端口的电平。如果波特率过高或运行频率过高，则 CPU 不能辨别该 I/O 端口的电平。为避免出现这种情况，该波特率被设置为 I/O 接口所需波特率的 1/16。在串行操作模式被确定为 I/O 接口模式时，SCLK 输入模式即被设置。该控制器必须确保其 AC 时序限制性条件在选定波特率时得到满足。如果是 I/O 接口模式，则该引导程序不会检查接收错误标志；因而，不会出现错误确认响应(位 3, 0x08)。

2. 被从目标板传输到控制器的第 2 字节是对第 1 字节的确认响应，其中已设置了串行操作模式。在第 1 字节被确定为 UART，并可被设置在规定波特率时，即可传输数据"0x86"。在第 1 字节被确定为 I/O 接口时，即可传输数据"0x30"。

• UART 模式

第 2 字节用于判别是否可设置该波特率。如果可设置该波特率，则值 SC0BRCR 即被更新，且数据"0x86"即被发送给该控制器。如果无法设置该波特率，则传输操作即被停止，不会传输任何资料。在第 1 字节的传输完成之后，控制器允许暂停五秒。如其未能在该容许暂停周期内收到 0x86，则该控制器即放弃该次通信。通过在将 0x86 加载到 SIO 传输缓冲区之前设置 SC0MOD0<RXE>=1，接收操作即可获得容许。

• I/O 接口模式

该引导程序可设置 SC0MOD0 与 SC0CR 寄存器的值，以配置该 I/O 接口模式，并将 0x30 写入到该 SC0BUF。然后，SIO0 会等待来自该控制器的 SCLK0 信号。在第 1 字节的传输完成之后，控制器会在某一空闲时间(若干微秒)结束之后，将 SCLK 时钟脉冲发送目标板。该操作必须按所需波特率的 1/16 进行。如果被从目标板发送到控制器的第 2 字节是 0x30，则控制器会将其视为可允许通信。从第 3 个字节开始，用户可按规定波特率传输数据。通过在将 0x86 加载到 SIO 之前设置 SC0MOD0<RXE>=1，接收操作即可获得容许。

3. 从控制器传输到目标板的第 3 字节属于命令。RAM 传送命令的代码为 0x10。

4. 被从目标板传输到控制器的第 4 字节，是对第 3 字节的确认响应。在发回该确认响应之前，该引导程序会检查是否发生接收错误。如果存在接收错误，则该引导程序会传输 0xX 8 (位 3)，并重新返回到等待命令(第 3 字节)的状态。在这种情况下，该确认响应的上四位字节未被定义，且其值与此前所发命令的上四位字节的相同。在针对 I/O 接口模式对 SIO0 进行配置时，该引导程序不检查是否存在接收错误。

如果第 3 字节等于表 19-4 中所列的任何命令代码，则该引导程序会将其回送到控制器。在接收到 RAM 传送命令时，该引导程序会回送该值 0x10，然后转到 RAM 传送例行程序。一旦发生了该转入，密码验证即告完成。有关密码校验的详细说明，见后文的"密码"一节。如果第 3 字节不是有效命令，则该引导程序会将 0xX 1 (位 0)发回到该控制器，并重新返回到等待命令(第 3 字节)的状态。 在这种情况下，该确认响应的上四位字节未被定义，且其值与此前所发命令的上四位字节的相同。

5. 从控制器传输到目标板的第 5 ~ 第 16 字节，是 12-字节密码。各字节均需与闪存存储器中以下地址的内容进行比较。用第 5 字节启动该校验。如果密码校验失败，则 RAM 传送例行程序会设置该密码错误标志。

产品名称	区域
TMPM370FYDFG/ FYFG	0x3F83_FFF4 ~ 0x3F83_FFFF

6. 第 17 字节是该密码序列的检查和值(第 5～ 第 16 字节)。为计算该 12-字节密码的检查和值，同时添加该 12 个字节，忽略进位并利用下 8-位计算这两个 8 位项的余数，然后从控制器传输该检查和值。有关该检查和计算的详细说明，见后文"检查和计算"一节。

7. 被从目标板传输到控制器的第 18 字节，是对第 5 字节 ～ 第 17 字节的确认响应。首先，该 RAM 传送例行程序会检查第 5～ 第 17 字节中是否存在接收错误。如果存在接收错误，则该引导程序会发回 0x18(位 3)并重新返回到等待命令(即第 3 字节)的状态。在这种情况下，该确认响应的上四位字节与此前所发命令的相同(即 1)。在针对 I/O 接口模式对 SIO0 进行配置时，该 RAM 传送例行程序不检查是否存在接收错误。

   然后，RAM 传送例行程序进行检查和运算，以确保第 17 字节的数据完整性。第 5 ～ 第 16 字节系列的添加一定会产生 0x00(该进位被撤销)。如果是校验和错误，则该 RAM 传送例行程序会将 0x11 发回到该控制器，并重新返回到等待命令(即第 3 字节)的状态。

   最后，会对该密码校验结果进行检查。如果出现以下情况，则引导程序会传输一个密码错误确认响应(位 0，0x11)，并等待下一个运行命令(第 3 字节)。

   - 不管密码比较的结果如何，闪存存储器中某密码的所有 12 个字节均为相同的值，但 0xFF 除外。
   - 并非从控制器传输的全部密码字节均可匹配闪存存储器中所包含的那些对象。

   在以上所有校验均成功时，该 RAM 传送例行程序将标准确认响应(0x10)返回到该控制器。

8. 被从控制器传输到目标板的第 19~第 22 字节，可指示该 RAM 区域其中应存储有后续数据(例如：闪存编程例行程序))的起始地址。第 19 字节对应于该地址的位 31～24，第 22 字节则对应于该地址的位 7～0。

   所存储 RAM 的起始地址必须为偶地址。

9. 被从控制器传输至目标板的第 23 字节与第 24 字节，可指示拟被从控制器传送，并被存储在该 RAM 中的字节的数目。第 23 字节对应于拟传送字节数目的位 15～8，而第 24 字节则对应于该字节数目的位 7～0。

10. 第 25 字节是第 19 字节 ～ 第 24 字节的检查和值。为计算检查和值，应同时添加这些字节，忽略进位并利用下 8-位计算这两个 8 位项的余数，然后从控制器传输该检查和值。有关该检查和计算的详细说明，见后文"19.2.10.6 检查和计算"一节。

11. 被从目标板传输到控制器的第 26 字节，是对数据的第 19 字节 ~ 第 25 字节的确认响应。首先，该 RAM 传送例行程序会检查第 19 ~ 第 25 字节中是否存在接收错误。如果存在接收错误，则该 RAM 传送例行程序会发回 0x18，并重新返回到命令等待(即第 3 字节)的状态。在这种情况下，该确认响应的上四位字节与此前所发命令的相同(即 1)。在针对 I/O 接口模式对 SIO0 进行配置时，该 RAM 传送例行程序不检查是否存在接收错误。

    然后，该 RAM 传送例行程序进行检查和运算，以确保数据完整性。第 19~第 24 字节系列的添加一定会产生 0x00(该进位被撤销)。如果是校验和错误，则该 RAM 传送例行程序会将 0x11 发回到该控制器，并重新返回到等待命令(即第 3 字节)的状态。

    - 第 19 字节 ~ 第 25 字节数据必须在从 0x2000_0400 到 RAM 结束地址之间的范围内。

    在以上校验均成功时，该 RAM 传送例行程序会将一个标准确认响应(0x10)返回到该控制器。

12. 来自该控制器的第 27 字节 ~ 第 m 字节，即被存储在该 TMPM370FYDFG/FYFG 的片上 RAM 中。存储从第 19 ~ 第 22 字节所指定的地址开始，并会按第 23 ~ 第 24 字节所指定的字节数目持续。

13. 第(m+1)字节即为检查和值。为计算检查和值，应同时添加第 27 ~ 第 m 字节，忽略进位并利用下 8-位计算这两个 8 位项的余数，然后从控制器传输该检查和值。有关该检查和计算的详细说明，见后文"19.2.10.6 检查和计算"一节。

14. 第(m+2)字节是对第 27 ~ 第(m+1)字节的确认响应。首先，该 Transfer 例行程序会检查第 27 ~ 第(m+1)字节中是否存在接收错误。如果存在接收错误，则该 RAM 传送例行程序会发回 0x18(位 3)并重新返回到等待命令(即第 3 字节)的状态。在这种情况下，该确认响应的上四位字节与此前所发命令的相同(即 1)。在针对 I/O 接口模式对 SIO0 进行配置时，该 RAM 传送例行程序不检查是否存在接收错误。

    然后，该 RAM 传送例行程序进行检查和运算，以确保数据完整性。第 27 ~ 第(m+1)字节系列的添加一定会产生 0x00(该进位被撤销)。如果是校验和错误，则该 RAM 传送例行程序序会将 0x11(位 0)发回到该控制器，并重新返回到等待命令(即第 3 字节)的状态。在以上校验均成功完成时，该 RAM 传送例行程序会将一个标准确认响应(0x10)返回到该控制器。

15. 如果第(m+2)字节是标准确认响应，则转入第 19 ~ 第 22 字节所指定的地址。

.

19.2.10.2 芯片与保护位擦除命令

有关该命令的传送格式见表 19-7。

1. 第 1 与第 2 字节的处理同传送命令。
2. 从控制器到 TMPM370FYDFG/FYFG

目标板收自控制器的第 3 字节为命令。芯片与保护位擦除命令的代码是 0x40。

3. 从 TMPM370FYDFG/FYFG 到控制器

被从目标板传输到控制器的第 4 字节，是对第 3 字节的确认响应。

在发回该确认响应之前，该引导程序会检查是否存在接收错误。如果存在接收错误，则该引导程序会传输 0xX 8(位 3)，并重新返回到命令等待状态。在这种情况下，该确认响应的上四位字节未被定义，且其值与此前所发命令的上四位字节的相同。

如果第 3 字节等于表 19-4 中所列的任何命令代码，则该引导程序会将其回送到控制器。在接收到芯片与保护位擦除命令时，该引导程序回送值 0x40。如果第 3 字节不是有效命令，则该引导程序会将 0xX 1 (位 0)发回到该控制器，并重新返回到等待命令(第 3 字节)的状态。 在这种情况下，该确认响应的上四位字节未被定义，且其值与此前所发命令的上四位字节的相同。

4. 从控制器到 TMPM370FYDFG/FYFG

被从目标板传输控制器的第 5 字节，是芯片擦除启用命令代码(0x54)。

5. 从 TMPM370FYDFG/FYFG 到控制器

从目标板传输到控制器的第 6 字节，是对第 5 字节的确认响应。

在发回该确认响应之前，该引导程序会检查是否存在接收错误。如果存在接收错误，则该引导程序会传输 0xX 8(位 3)，并重新返回到命令等待状态。在这种情况下，该确认响应的上四位字节未被定义，且其值与此前所发命令的上四位字节的相同。

如果第 5 字节等于拟启用擦除的任何命令代码，则该引导程序会将其回送到控制器。在接收到芯片与保护位擦除命令时，该引导程序回送值 0x54，然后转到该芯片擦除例行程序。如果第 5 字节不是有效命令，则该引导程序会将 0xX 1(位 0)发回到该控制器，并重新返回到等待命令(第 3 字节)的状态。在这种情况下，该确认响应的上四位字节未被定义，且其值与此前所发命令的上四位字节的相同。

6. 从 TMPM370FYDFG/FYFG 到控制器

第 7 字节可指示芯片擦除命令是否已被正常完成。

在正常完成时，会发送完成码(0x4F)。

在检测到错误时，会发送错误代码(0x4C)。

7. 第 9 字节是下一个命令代码。

19.2.10.3 确认响应

该引导程序可显示处理状态及具体代码。表 19-8 给出了针对所接收数据的可能确认响应的值。该确认响应的上四位字节等于执行中命令的对应项。第 3 位可指示接收错误。第 0 位可指示无效命令错误，校验和错误或密码错误。第 1 位与第 2 位始终为"0"。接收误差检查并非在 I/O 接口模式下进行。

表 19-8 对串行操作模式字节的 ACK 响应

返回值	含义
0x86	该 SIO 经过配置，可在 UART 模式下运行. (参见注释)
0x30	该 SIO 经过配置，可在 I/O 模式下运行。

注：**在 UART 模式下，如果无法设定波特率设置，则通信停止，且无任何响应。**

表 19-9 对命令字节的 ACK 响应

返回值	含义
0x?8 (参见注释)	在命令代码接收期间发生的接收错误。
0x?1 (参见注释)	接收到未定义的命令代码(接收已正常完成)。
0x10	接收到 RAM 传送命令。
0x40	接收到芯片擦除命令。

注：**该 ACK 响应的上四位字节同前一命令代码。**

表 19-10 对检查和字节的 ACK 响应

返回值	含义
0xN8 (参见注释)	发生接收错误。
0xN1 (参见注释)	发生检查和密码错误。
0xN0 (参见注释)	检查和正确。

注：**该 ACK 响应的上四位字节同运算命令代码。例如，在发生密码错误时，其为 1 (N ; RAM 传送命令数据[7:4])。**

表 19-11 对芯片与保护位擦除字节的 ACK 响应

返回值	含义
0x54	接收到芯片擦除启用命令。
0x4F	芯片擦除命令已完成。
0x4C	芯片擦除命令完成异常。

19.2.10.4 串行操作模式的确定

来自控制器的第一个字节可决定串行操作模式。在利用 UART 模式进行控制器与目标板之间的通信时，控制器必须首先按所需的波特率将值 0x86 发送到该目标板。在采用 I/O 接口模式时，控制器必须按所需波特率的 1/16 发送值 0x30。图 19-4 给出了各模式下第一个字节的波形。

注：图 19-4 中 A/B/C/D 各点之间可表示为 tAB，tAC，tAD 与 tCD。

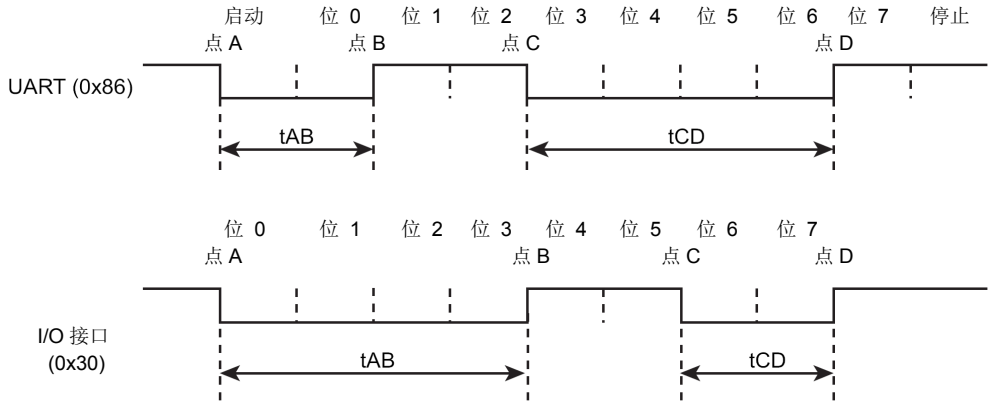


图 19-4 串行操作模式字节

在 $\overline{\text{RESET}}$ 被解除之后，该引导程序监视来自该控制器的第一个串行字节 SIO 接收被禁用，并计算 tAB ，tAC 与 tAD 的间距。图 19-5 给出了用于说明 tAB，tAC 与 tAD 间距的确定步骤的流程图。如该流程图所示，第一个串行字节中每发生一次逻辑推移，该引导程序就会捕捉一次计时器计数。因此，所计算出的 tAB，tAC 与 tAD 间距往往存在细微误差。如果传送时为高波特率，则 CPU 可能难以跟上该串行接收引脚部位的逻辑转换速度。尤其是 I/O 接口模式很可能会出现这种问题，原因是其波特率一般远高于 UART 模式下的波特率。为避免出现这种情况，控制器应按所需波特率的 1/16 发送第一个串行字节。

图 19-5 中的流程图给出了该引导程序判别 UART 与 I/O 接口方式的具体方法。如果 tAB 的长度等于或小于 tCD 的长度，则串行操作模式会被确定为 UART 模式。如果 tAB 的长度大于 tCD 的长度，则串行操作模式会被确定为 I/O 接口模式。注意，如果波特率过高或计时器运行频率过低，则各计时器值会变小。这样可导致控制器发生非故意的动作。为防止发生该问题，可在编程例行程序内部复位 UART 模式。

例如，在预定模式是 UART 模式时，该串行操作模式可被确定为 I/O 接口模式。为避免出现这种情况，在运用 UART 模式时，控制器允许出现(其有待从目标板接收某回送(0x86)的)暂停周期。如果其未能在该容许时间内收到该回送，则控制器应停止通信。在运用 I/O 接口模式时，一旦传输了第一个串行字节，控制器即应在某一空闲时间结束后发送 SCLK 时钟脉冲，以获得确认响应。如果所收到的确认响应不是 0x30，则控制器应在更大程度上停止通信。

在预定模式是 I/O 接口模式时，只要 tAB 大于上述的 tCD，第一个字节就无须是 0x30。0x91，0xA1 或 0xB1 可被作为第一个字节代码发送，用于确定点 A 与点 C 的下降沿，以及点 B 与点 D 的上升沿。如果 tAB 大于 tCD，且该运行模式确定的分辨率选择了 SIO，则即使第一个字节上的被传输代码不是 0x30(用于确定 I/O 接口模式第一个字节代码为 0x30)，第二字节代码仍然是 0x30。

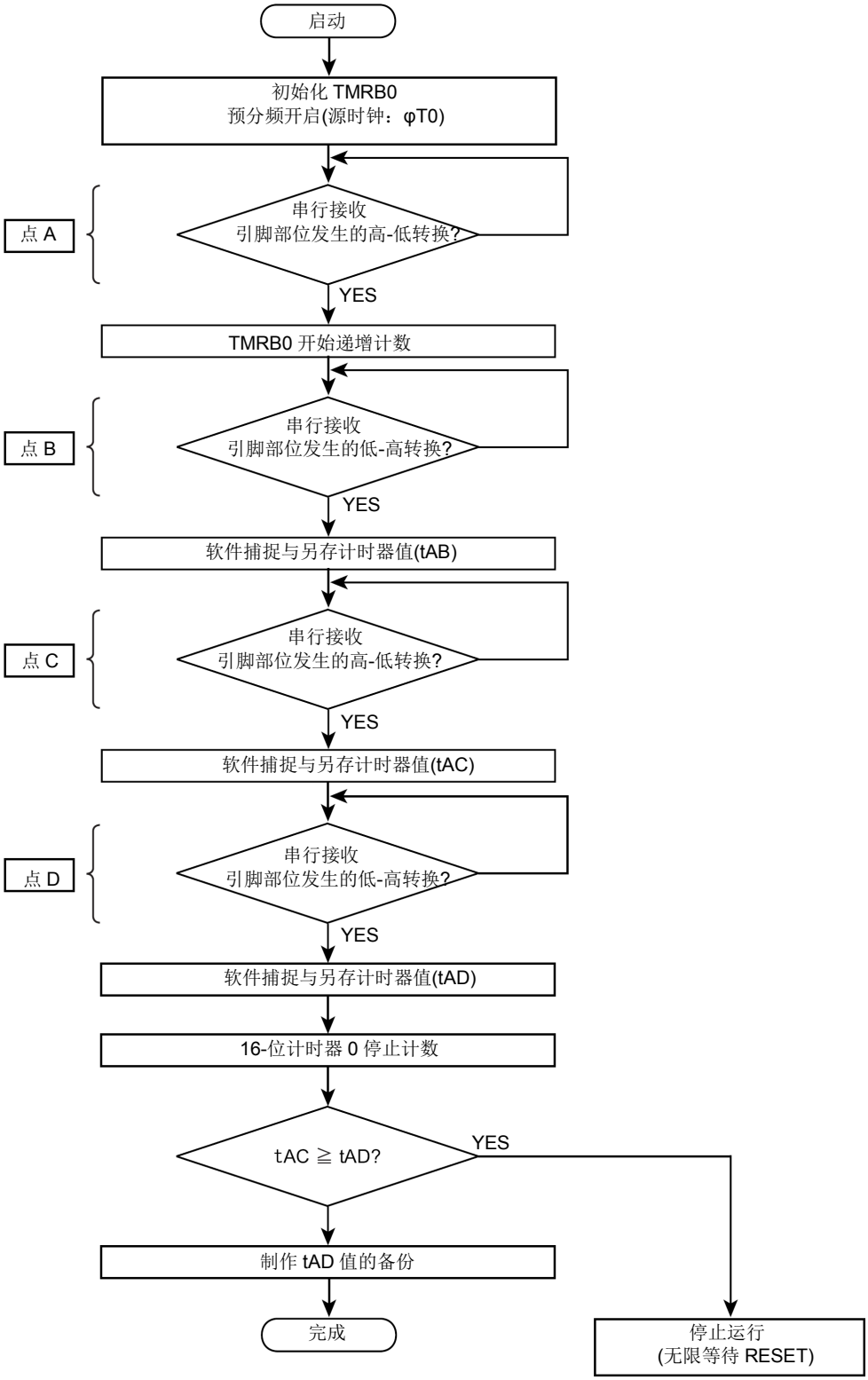


图 19-5 串行操作模式字节接收流程图

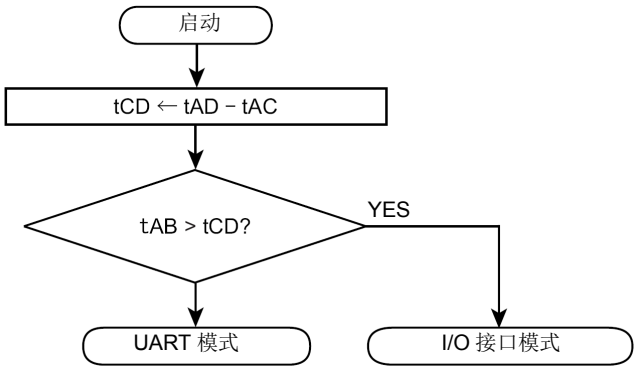


图 19-6 串行操作模式确定流程图

19.2.10.5 密码

RAM 传送命令(0x10)可导致该引导程序执行密码校验。在该命令代码的回送完成后，该引导程序会校验闪存存储器内部 12-字节密码区域的内容。下表给出了各产品的密码区域。

产品名称	区域
TMPM370FYDFG/ FYFG	0x3F83_FFF4 ~ 0x3F83_FFFF

注：**如果密码被设置为 0xFF (已擦除数据区域)，则可因密码易于被猜出而导致难以妥善保护数据。即使未使用单一引导模式，也建议将某个唯一值设置为密码。**

如果所有地址单元包含相同的数据字节 0xFF 除外,则会发生图 19-7 所示的密码区域错误。在这种情况下，无论发自该控制器的密码序列是否全部是 0xFFs，该引导程序都会根据该检查和字节(第 17 字节)返回一个错误确认(0x11)。

会将来自该控制器的接收数据(第 5～ 第 16 字节)，与闪存存储器中存储的密码进行比较。所有这 12 个字节均必须匹配才能通过该密码校验。否则会发生密码错误，导致该引导程序根据该检查和字节(第 17 字节)答复错误确认。

即使安全功能已被启用，也会执行该密码校验。

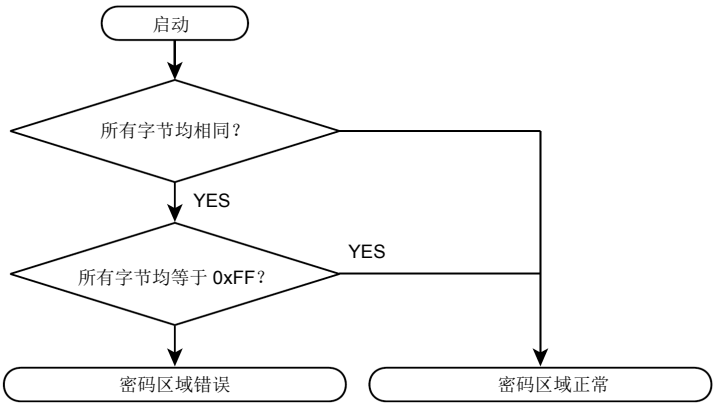


图 19-7 密码区域校验流程图

19.2.10.6 检查和计算

通过添加字节，忽略各进位并利用下 8 位计算这两个 8-位项的余数，即可计算出一系列数据字节的检查和字节。在传输检查和字节期间，控制器必须执行相同的检查和运算。

示例)计算一系列 0xE5 与 0xF6 的检查和：

添加各字节

且 0xE5 + 0xF6 = 0x1DB

利用下 8 位计算这二者的余数，其即为该检查和字节。然后将 0x25 发送到控制器。

0 - 0xDB = 0x25

19.2.11 一般引导程序流图

图 19-8 给出了该引导程序的总体流程图。

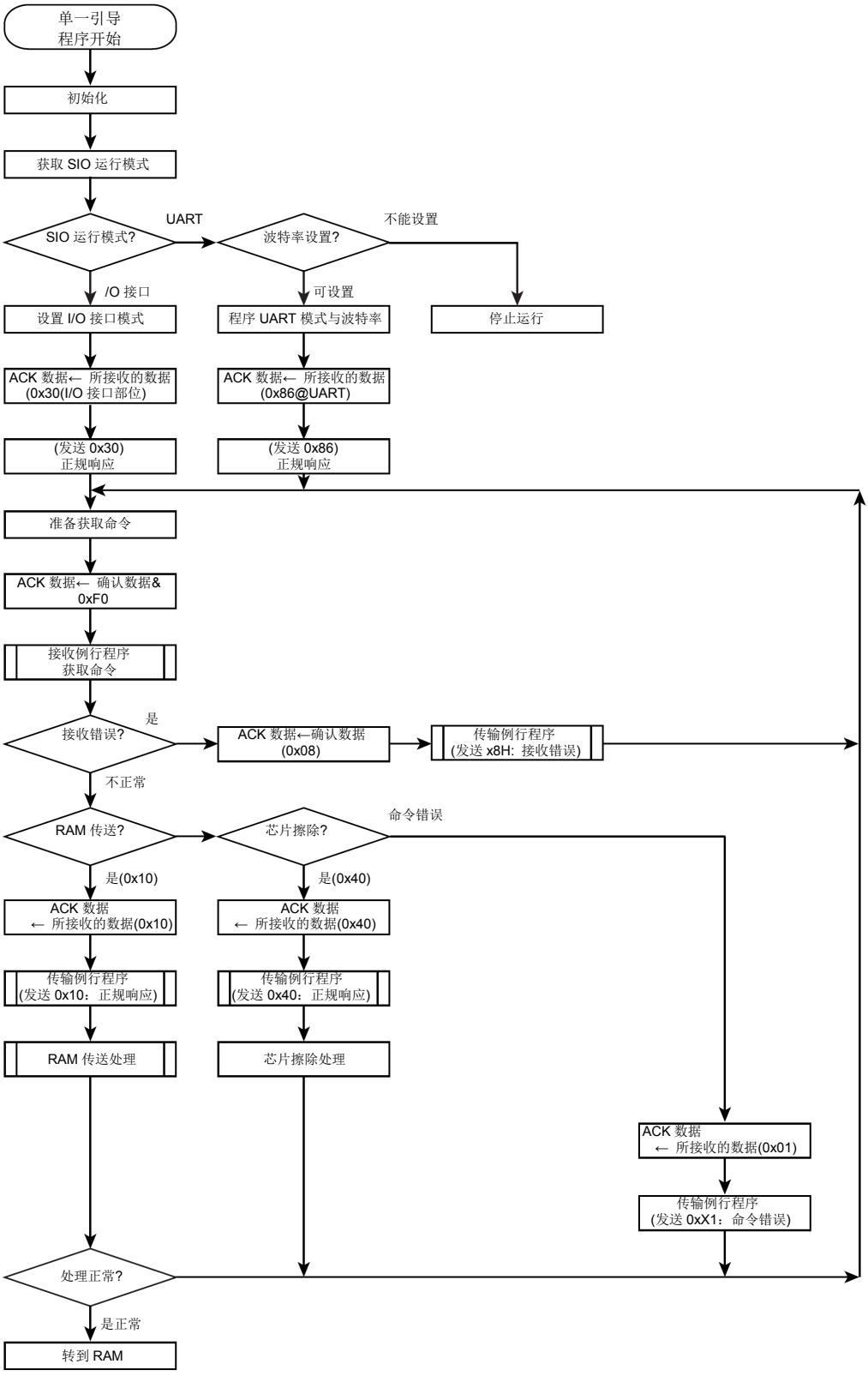


图 19-8 总体引导程序流图

19.3 闪存存储器的板上编程(重写/擦除)

在板上编程时，CPU 会执行闪存存储器重写或擦除软件指令。应由用户事先编制该重写 / 擦除控制程序。由于在被写入或擦除期间无法读取闪存存储器内容，必须在转入到用户引导模式之后，从内部 RAM 运行该重写/擦除程序。

19.3.1 闪存存储器

除某些功能之外，闪存存储器数据的写入与擦除应符合标准 JEDEC 命令的要求。

在写入或擦除时，应利用 CPU 的 32-位数据传送命令将命令输入到该闪存存储器。一旦该被输入，即以内部方式自动执行实际写入或擦除。

表 19-12 闪存存储器功能

主要功能	说明
自动页面程序	自动将数据写入每页。
自动芯片擦除	自动擦除该闪存存储器的整个区域。
自动存储块擦除	自动擦除某选定存储块。
保护功能	可单独禁止各存储块的写入或擦除操作。

19.3.1.1 块配置

(1) TMPM370FYDFG/ FYFG

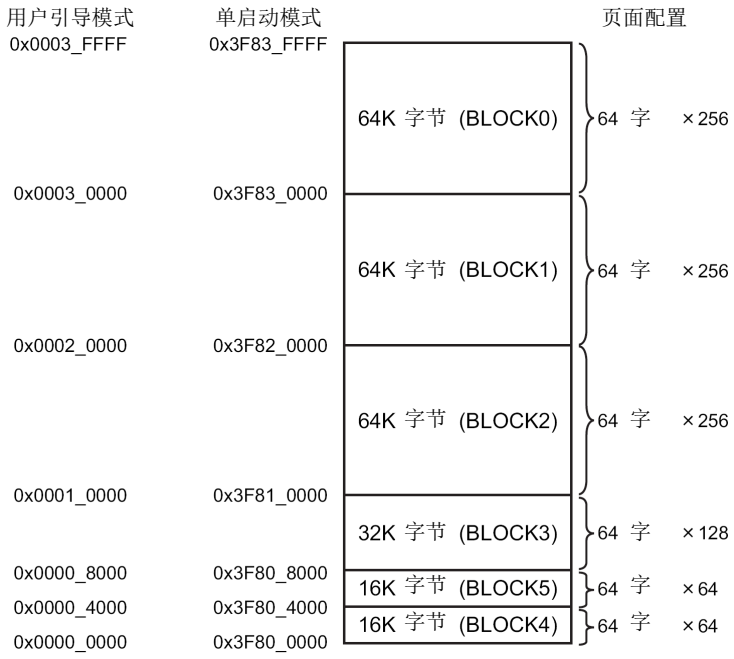


图 19-9 闪存存储器(TMPM370FYDFG / FYFG)的存储块配置

19.3.1.2 基本运行

闪存存储器设备具备以下双向运行模式：

- 读存储器数据模式(读取模式)
- 自动擦除或重写存储器数据(自动运行)的模式

通过在其处于存储器读出模式时执行某命令序列，即可转换到该自动模式。在自动运行模式下，无法读取速存储器数据，且无法执行存储在该闪存存储器中的任何命令。在该自动运行模式下，任何中断或异常发生均无法将该设备设置为读取模式，但硬件复位发生时除外。在自动运行期间，不得导致除复位与调试异常调试端口已连接外的任何异常。任何异常发生均可导致无法设置该设备，但发生硬件复位时的情形除外。

(1) 读取

在读取数据之前，必须将闪存存储器设置为读取模式。在 CPU 复位被清除，或自动运行被正常终止时，该闪存存储器在加电后就会被直接设置为读取模式。为了从其它模式返回到读取模式，或在自动运行被异常终止之后，可使用读取/复位命令(该软件命令将在后文述及)或硬件复位。在执行写在闪存存储器上的任何命令时，该设备也必须处于读取模式。

- 读取/复位命令与读取命令(软件复位)

  在使用 ID-读取命令时，该读取操作会被终止，而不是自动返回到该读取模式。在这种情况下，可利用该读取/复位命令使该闪存存储器返回到读取模式。此外，在必须取消某条尚未被完整写入的命令，必须使用该读取/复位命令。该读取命令用于在执行 32-位数据传送命令以将数据"0x0000_00F0" 写入闪存存储器的某个随机地址后，返回到读取模式。

- 利用该读取/复位，可让设备在完成第三总线写周期之后返回到读取模式。

(2) 命令写入

闪存存储器采用该命令控制方法。通过对该闪存存储器执行某命令序列，即可执行各命令。该闪存存储器可按照所应用的地址与数据组合执行自动运行命令(请参看命令序列)。

如需取消进行中的某命令写入操作，或在输入了任何不正确的命令序列时，可执行该读取/复位命令。然后，闪存存储器会终止命令执行并返回到读取状态。

命令一般由若干总线周期组成，且将该 32-位(字)数据传输命令应用于该闪存存储器的操作被称为"总线写周期"。该总线写周期具备特定的顺序，且是按照预定义的特定顺序操作该总线写周期数据与命令写入的地址时，该闪存存储器可执行自动运行。如果任何总线写周期不符合某个预定义的命令写入序列，该闪存存储器可终止该命令执行，并返回到读取模式。

注 1：**命令序列的执行是从闪存存储器区域的外部开始的。**

注 2：**必须通过 32 位数据传输命令，顺次执行各总线写周期。在正在执行某命令序列期间，禁止访问该闪存存储器。此外，不要生成任何中断(调试端口连接时的调试异常除外)。如果进行该操作，可导致对该闪存存储器进行意外的阅读存取，且该命令序列发生器不能正确识别该命令。在其可能导致该命令序列的异常终结的同时，还可能导致对该命令的不正确识别。**

注 3：**为便于该命令序列发生器识别命令，在执行该命令之前，该设备必须处于读取模式。在首个总线写周期开始之前，务必检查其中的 FCFLCS <RDY / BSY>是否已被设置为"1"。建议随后执行读取命令。**

注 4：**一旦发布了某命令，如果任何地址或数据的写入不正确，则务必执行软件复位以重新返回到读取模式。**

19.3.1.3 复位(硬件复位)

硬件复位用于在自动运行状态下进行自动编程/擦除或异常终结时需执行强制终止的情况下，取消通过命令写入操作设置的运行模式。

闪存存储器具备复位输入作为存储区，且其与 CPU 复位信号相连。因此，在该设备的 $\overline{\text{RESET}}$ 输入引脚被设置为 VIL，或 CPU 因监视时钟的任何溢出而被复位时，该闪存存储器可返回到读取模式，并终止可能正在进行中的任何自动运行。还应注意的是，在自动运行期间进行硬件复位可导致不正确的数据重写。在这种情况下，务必重新执行该重写。

有关 CPU 复位操作，请参看"19.2.1 复位操作"一节。在某个给定的复位输入完成后，CPU 可从闪存存储器读取复位向量资料，并在复位清除之后开始运行。

19.3.1.4 命令

(1) 自动页面程序

对闪存存储器设备进行写入，可导致"1"数据单元被改为"0"数据单元。任何"0"数据单元均无法被更改为"1"数据单元。在将"0"数据单元更改为"1"数据单元之前，必须执行擦除操作。

该设备的自动页面编程功能可用于写入各页面的数据。TMPM370FYDFG/FYFG 的每个页面内含 128 个字。由同一[31:9]地址定义 128 字存储块。其从地址[8:0] = 0x00 开始，并在地址[8:0] = 0x1FF 结束。在下文中，该编程单元被称为"页面"。

由内部序列发生器自动执行数据单元的写入，无需借助于 CPU 执行外部控制。可由 FCFLCS [0] <RDY/BSY>检查自动页面编程的状态(确认其是否处于写入操作状态)。

此外，在其处于自动页面编程模式期间，不会接收任何新的命令序列。如需中断中断自动页面编程，则可使用该硬件复位功能。如果硬件复位操作导致该操作停止，则必须一次性擦除该页面，然后重新执行自动页面编程，原因是页面写入尚未被正常终止。

只有在某页面已被一次性擦除的情况下，才允许进行自动页面编程操作。任何编程均不得执行两次或以上。注意，对于已写入页面的重写，需要在重新执行该自动页面编程命令之前执行自动存储块擦除或自动芯片擦除命令。注意，在未擦除其内容的情况下试图两次或两次以上重写某页面，可导致设备受损。

不得以内部方式对设备进行自动验证操作。因此，务必读取已编程的数据，以确认其写入正确。

一旦该命令周期的第三总线写周期结束，自动页面编程操作即开始。在第五个总线写周期结束之后，将从第四个总线写周期中所指定地址的下一地址开始(在第四总线写周期期间，可以命令形式写入页面顶端地址)(数据的 32 位被一次性输入)，顺次写入数据。在第四个总线周期结束之后，务必使用各写入命令中的该 32-位数据传送命令。此时，不应将任何 32-位数据传送命令置于跨字边界处。在第五个总线写周期结束之后，应以命令形式将数据写入到相同的页面区域。即使仅需部分写入该页面，仍需针对整个页面进行自动页面编程。在这种情况下，应将第四个总线写周期的地址输入设置为该页面的顶端地址。务必执行命令写入操作；对于不拟设置为"0"的数据单元，可将其输入数据设置为"1"。例如，如果不拟写入某页面的顶端地址，则将第四个总线写周期中的输入数据作为命令写入设置为 0xFFFFFFFF。

一旦第三个总线周期的执行完成，自动页面编程即处于运行状态。通过监视 FCFLCS<RDY / BSY>，即可检查其情况。在其处于自动页面编程模式期间，不会接收任何新的命令序列。如需停止运行，可使用该硬件复位功能。如果运行被中断，则数据无法被正常写入；因此，务必小心操作。在某单一页面已命令写入(正常终止自动页面写入过程)的情况下，FCFLCS<RDY / BSY>即被设置为"1"，然后其返回到读取模式。

在拟写入多个页面时，必须针对各页面执行该页面编程命令，原因是拟通过单次执行自动页面程序命令写入页面的数目仅限一个页面。不允许自动页面编程跨页处理输入数据。

数据无法被写入到受保护的存储块。在自动编程序完成时，其自动返回到读取模式。通过监视 FCFLCS<RDY / BSY>，即可检查其情况。如果自动编程序失败，则闪存存储器会被锁定在当前模式，且不会返回到读取模式。必须执行硬件复位以复位闪存存储器或设备之后，才能返回到读取模式。在这种情况下，在地址写入失败时，建议不使用该设备或不使用内含该失败地址的存储块。

注：**在自动页面编程命令的第四个总线写周期结束之后，软件复位无效。**

(2) 自动芯片擦除

一旦该命令周期的第六个总线写周期结束，自动芯片擦除操作即开始。

通过监视 FCFLCS<RDY / BSY>，即可检查其情况。由于未以内部方式对该设备进行自动验证，因此务必读取数据，以确认数据已被正确擦除。在其处于自动芯片擦除操作期间，不会接收任何新的命令序列。如需停止运行，可使用该硬件复位功能。如果不得不停止该操作，则必须执行重新执行该自动芯片擦除操作，原因是数据擦除操作未被正常终止。

此外，任何受保护的存储块均无法被擦除。如果所有存储块均处于受保护状态，则不会执行自动芯片擦除操作，且其会在完成该命令序列的第六个总线读周期之后返回到读取模式。在某自动芯片擦除操作被正常终止时，其会自动返回到读取模式。如果自动芯片擦除操作失败，则闪存存储器会被锁定在当前模式，且不会返回到读取模式。

必须执行硬件复位以复位该设备之后，才能返回到读取模式。在这种情况下，该失效存储块无法被检测到。建议不再使用该设备，也可利用存储块擦除功能标识该失效存储块，以确保不再使用该所标识的存储块。

(3) 自动存储块擦除(针对各存储块)

一旦该命令周期的第六个总线写周期结束，自动存储块擦除操作即开始。

通过监视 FCFLCS <RDY / BSY>，即可检查自动存储块擦除操作的状态。由于未以内部方式对该设备进行自动验证，因此务必读取数据，以确认数据已被正确擦除。在其处于自动存储块擦除操作期间，不会接收任何新的命令序列。如需停止运行，可使用该硬件复位功能。在这种情况下，必须重新执行自动存储块擦除操作，原因是数据擦除操作未被正常终止。

此外，任何受保护的存储块均无法被擦除。如果自动存储块擦除操作失败，则闪存存储器会被锁定在该模式，且不会返回到读取模式。在这种情况下，应执行硬件复位以复位该设备。

(4) 保护位的自动编程(针对各存储块)

该设备的实现基于保护位。可针对各存储块设置该保护。保护位地址见表 19-16。该设备会将 1 位作为保护位指定给 1 个存储块。适用保护位则由第七个总线写周期中的 PBA 指定。通过保护位自动编程，可针对各存储块，单独禁止写入和/或擦除功能。通过后文所述的 FCFLCS <BLPRO>，可检查各存储块的保护状态。通过监视 FCFLCS <RDY/BSY>，即可检查用于设置保护位的自动编程操作的状态。在保护位自动编程进行期间，不会接收任何新的命令序列。

如需停止该编程操作，可使用该硬件复位功能。在这种情况下，必须重新执行该编程操作，原因是各保护位可能未被正确编程。如果所有保护位均已编程，则所有 FCFLCS<BLPRO>会被设置为"1"表示其处于受保护状态。这样可禁用后续的写入，并擦除所有存储块。

注： **在自动保护位编程命令的第七个总线写周期内，软件复位无效。在进入第七个总线写周期之后，FCFLCS <RDY/BSY> 变成"0"。**

(5) 保护位的自动擦除

自动保护位擦除命令的执行可获得不同的结果，视各保护位与安全位的状态而定。其取决于在 FCSECBIT<FCSECBIT> 被设置为"1"时，是否 FCFLCS 寄存器中的所有<BLPRO>均已被设置为"1"。在执行自动保护位擦除命令之前，务必检查 FCFLCS <BLPRO>的值。有关详细说明见"保护/安全功能"一节。

- 在所有 FCFLCS <BLPRO> 均被设置为"1"时(所有保护位均已编程)：

  在自动保护位擦除命令已被命令写入时，闪存存储器即在设备内部被自动初始化。在第七个总线写周期完成时，各闪存存储器数据单元的整个区域均被擦除，然后各保护位也被擦除。通过监视 FCFLCS<RDY / BSY>，即可检查其操作情况。如果旨在擦除各保护位的自动运行被正常终止，则 FCFLCS 即被设置到"0x00000001"。由于未以内部方式对该设备进行自动验证，因此务必读取数据，以确认数据已被正确擦除。如需在第七总线周期后的自动操作正在进行期间返回读取模式，必须使用硬件复位方式复位该设备。如果成功，则必须在返回到读取模式之后，通过 FCFLCS<BLPRO>检查各保护位的状态，并根据需要执行自动保护位擦除，自动芯片擦除，自动存储块擦除操作。

- 在 FCFLCS <BLPRO> 包含 "0" (并非所有保护位均已编程)时：

  如果该自动保护位被清除，则保护状态即被取消。利用该设备，可将各保护位编程到单个存储块，并按四位字节单位执行位擦除操作(如表 19-16 所示)。应在第七总线写周期内指定各目标位。通过后文所述的 FCFLCS <BLPRO>，可检查各存储块的保护状态。通过监视 FCFLCS <RDY / BSY>，即可检查各自动保护位的编程操作的状态。在旨在擦除各保护位的自动操作被正常终止时，被选定用于擦除的 FCFLCS <BLPRO>的保护位即被设置为"0"。

无论如何，在其处于自保护擦除自动操作期间，不会接收任何新的命令序列。如需停止运行，可使用该硬件复位功能。在保护位擦除自动操作被正常终止时，其会自动返回到读取模式。

注： **在自动操作期间，FCFLCS <RDY / BSY>位为"0"，在自动操作终止时，其变为"1"。**

(6) ID-读取

利用该 ID 读取命令，操作员可获取该设备内装闪存存储器的型号及其他信息。拟加载的数据将有所不同，视第四个与后续各总线写周期(建议输入数据是 0x00)的地址[15:14]而定。在第四个总线写周期结束后，一旦读取某随机闪存存储器区域，该 ID 值即被加载。一旦某 ID 读取命令的第四总线写周期已过，该设备不会自动返回到读取模式。在这种情况下，可反复执行第四总线写周期与 ID 读取命令的设置。使用该读取/复位命令或硬件复位命令，即可返回到读取模式。

19.3.1.5 闪存控制器/状态寄存器

基址 = 0x41FF_F000

寄存器名称		地址(基本+)
保留	-	0x0000
保留	-	0x0004
安全位寄存器	FCSECBIT	0x0010
保留	-	0x0014
闪存控制寄存器	FCFLCS	0x0020
保留	-	0x0024 ~ 0x0FFF

注：**不要存取该保留地址。**

(1) FCFLCS (闪存控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
复位后	0	0	(注 2)	(注 2)	(注 2)	(注 2)	(注 2)	(注 2)
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	RDY_BSY
复位后	0	0	0	0	0	0	0	1

位	比特符号	型号	功能
31 ~ 22	−	R	读作 0。
21 ~16	BLPRO5 ~ BLPRO0	R	存储块 5~ 0 的保护 0：被禁用 1：被启用 各保护位可表示对应存储块的保护状态。在某位被设置为"1"时，表示该位所对应的存储块处于受保护状态。在该存储块受保护时，无法将数据写入到该存储块。
15~ 1	−	R	读作 0。
0	RDY/BSY	R	就绪/忙(注 1) 0：自动运行 1：自动运行被终止。 就绪/忙标志位 该 RDY/BSY 输出用于监视自动运行的状态。该位是 CPU 的功能位，用于监视该功能。在闪存存储器处于自动运行状态时，会输出"0"表示其忙。在自动运行被终止时，其返回就绪状态，并输出"1"以接受下一条命令。如果自动运行失败，则该位保持"0"输出。通过进行硬件复位，其返回到"1"。

注 1：必须在就绪状态发布该命令。在忙状态发布该命令，可禁用正确命令传输与进一步的命令输入。执行系统复位，即可退出该状态。无论系统时钟频率为何，系统复位均至少需要 0.5 μs。在这种情况下，复位后启用读取需耗时约 2 ms。

注 2：该值可随所应用保护的不同而变化。

(2) FCSECBIT (安全位寄存器)

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

位	比特符号	型号	功能
31-1	−	R	读作 0。
0	SECBIT	R/W	安全位 0:被禁用 1:被启用

注： **该寄存器已通过冷复位实现初始化。**

19.3.1.6 命令序列列表

表 19-13 给出了闪存存储器各命令的地址与数据。

除读取命令的第二总线周期，读取/复位命令的第四总线周期，以及 ID 读取命令的第五总线周期外，其它所有总线周期均为"总线写周期"。通过 32-位(字)数据传送命令执行总线写周期(在下表中，仅给出了下 8 位数据)。

有关的地址位配置的详细资料，见表 19-14。将表 19-13 中的值"Addr."用作表 19-14 中标准命令的地址[15:8]。

注：**始终将"0"设置到整个总线周期中的地址位[1:0]。**

表 19-13 该内部 CPU 进行闪存存储器访问

命令序列	第一个总线周期	第二个总线周期	第三个总线周期	第四个总线周期	第五个总线周期	第六个总线周期	第七个总线周期
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	数据	数据	数据	数据	数据	数据	数据
读取	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
读取/复位	0x54XX	0xAAXX	0x54XX	RA	–	–	–
	0xAA	0x55	0xF0	RD	–	–	–
ID-读取	0x54XX	0xAAXX	0x54XX	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
自动页面编程	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
自动芯片擦除	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
自动存储块擦除	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
保护位编程	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
保护位擦除	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

补充说明事项

- RA：读取地址
- RD：读取数据
- IA：ID 地址
- ID：ID 数据
- PA：程序页地址

  PD：程序数据 (32 位数据)

  在第四个总线周期结束之后，按某页面地址的顺序输入数据。
- BA：块地址
- PBA：保护位地址

19.3.2 总线写周期的地址位配置

表 19-14 应与"表 19-13 从内部 CPU 开始闪存存储器访问"配套使用。

可按照正常总线写周期地址配置，从第一个总线周期开始进行地址设置。可根据需要，更改表 19-14 总线写周期的地址位配置中的"建议采用 0"。

地址	Addr [31:19]	Addr [18]	Addr [17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10]	Addr [9]	Addr [8]	Addr [7:0]
标准命令	标准总线写周期地址配置										
	闪存区域	建议采用"0"			命令						Addr[1:0]="0" (固定) 其他：0 (建议)
ID-READ	IA：ID 地址(设置 ID-读取操作的第四个总线写周期地址)										
	闪存区域	建议采用"0"			ID 地址		Addr[1:0]="0" (固定)，其他：0 (建议)				
存储块擦除	BA：块地址(设置存储块擦除运行的第六个总线写周期地址)										
	块选择(表 19-14)						Addr[1:0]="0" (固定)，其他：0 (建议)				
自动页面编程	PA：程序页地址(设置页面编程操作的第四个总线写周期地址)										
	页面选择										Addr[1:0]="0" (固定) 其他：0 (建议)
保护位编程	PBA:保护位地址(设置保护位编程的第七个总线写周期地址)										
	闪存区域	保护位选择 (表 19-15)		固定为"0"。					保护位选择 (表 19-15)		Addr[1:0]="0" (固定) 其他：0 (建议)
保护位擦除	PBA:保护位地址(设置保护位擦除的第七个总线擦除周期地址)										
	闪存区域	保护位选择 (表 19-16)		固定为"0"。					Addr[1:0]="0" (固定) 其他：0 (建议)		

指定拟擦除该存储块中的任何地址作为存储块地址。

有关存储块配置请参看 19.3.1.1。

表 19-14 块地址表

块	地址 (用户引导模式)	地址 (单一引导模式)	规格 (k 字节)
4	0x0000_0000 ~0x0000_3FFF	0x3F80_0000 ~0x3F80_3FFF	16
5	0x0000_4000 ~0x0000_7FFF	0x3F80_4000 ~0x3F80_7FFF	16
3	0x0000_8000 ~0x0000_FFFF	0x3F80_8000 ~0x3F80_FFFF	32
2	0x0001_0000 ~0x0001_FFFF	0x3F81_0000 ~0x3F81_FFFF	64
1	0x0002_0000 ~0x0002_FFFF	0x3F82_0000 ~0x3F82_FFFF	64
0	0x0003_0000 ~ 0x0003_FFFF	0x3F83_0000 ~0x3F83_FFFF	64

注：**至于第一个总线周期到第五个总线周期的地址，可指定拟擦除存储块的上地址。**

表 19-15 保护位编程地址表

块	保护位	第七个总线写周期地址						
		地址 [18]	地址 [17]	地址 [16]	地址 [15:11]	地址 [10]	地址 [9]	地址 [9]
存储块 0	<BLPRO[0]>	0	0	固定为"0"			0	0
存储块 1	<BLPRO[1]>	0	0				0	1
存储块 2	<BLPRO[2]>	0	0				1	0
存储块 3	<BLPRO[3]>	0	0				1	1
存储块 4	<BLPRO[4]>	0	1				0	0
存储块 5	<BLPRO[5]>	0	1				0	1

表 19-16 保护位擦除地址表

块	保护位	第七个总线写周期地址 [18:17]	
		地址[18]	地址[17]
块 0 ~ 3	<BLPRO[3:0]>	0	0

注：**该保护位擦除命令无法按单个存储块进行擦除。**

表 19-17 ID-读取命令的第四个总线写周期 ID 地址(IA)
与拟由以下 32-位数据传送命令(ID)读取的数据

IA[15:14]	ID[7:0]	代码
0y00	0x98	制造厂商代码
0y01	0x5A	设备代码
0y10	保留	-
0y11	0x13	宏代码

19.3.2.1 流程图

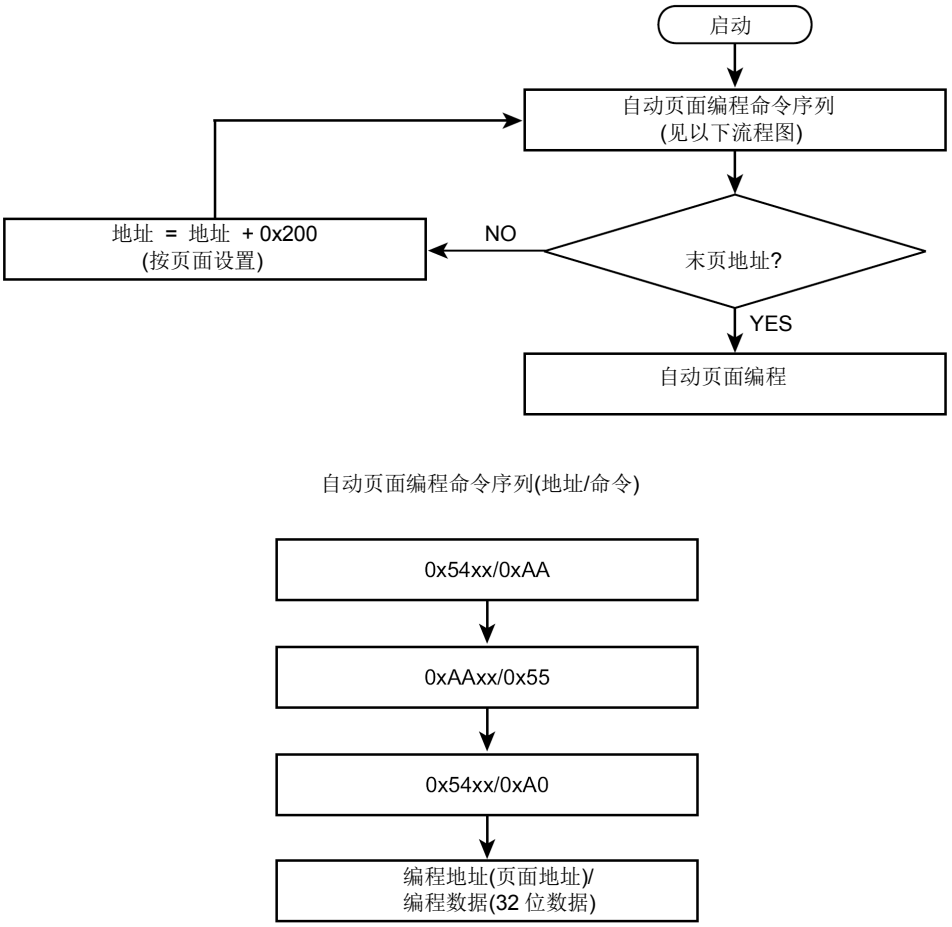


图 19-10 自动编程

注：**按 0x54xx 或 0x55xx 执行命令序列。**

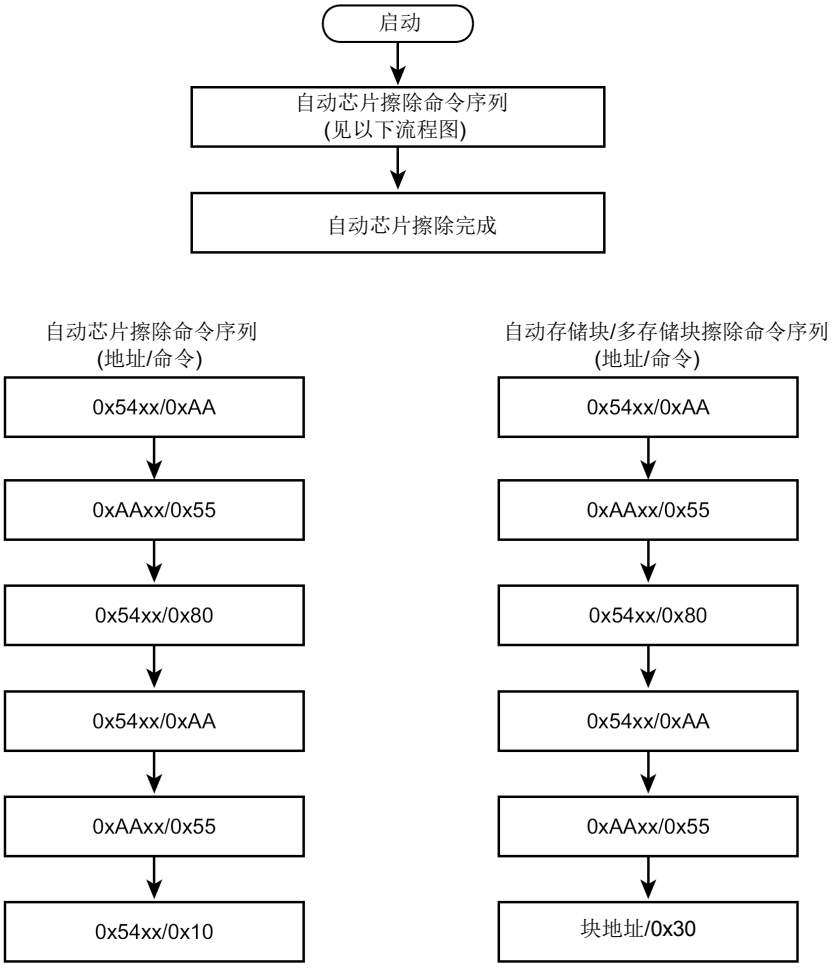


图 19-11 自动擦除

注： **按 0x54xx 或 0x55xx 执行命令序列。**

20. ROM 保护

20.1 概要

TMPM370FYDFG/FYFG 可提供两种 ROM 保护/安全功能。

其中一种是针对内部闪存 ROM 数据的写入/擦除保护功能。

另一种是用以限制内部闪存 ROM 数据读出与调试的安全功能。

20.2 未来

20.2.1 写入/擦除保护功能

该写入/擦除保护功能，可启用内部闪存禁止对各存储块进行写入与擦除操作。

将"1"写入到拟保护存储块的对应位，即可激活该功能。将"0"写入到该位，即可取消该保护。

由 FCFLCS <BLPRO[5:0]>位，可监视各位的保护设置。有关编程的详细说明，见"闪存"一节。

20.2.2 安全功能

安全功能可限制闪存 ROM 数据读出与调试。

该功能在下述条件下可用。

1. FCSECBIT <SECBIT>位被设置为"1"。
2. 用于写入/擦除保护功能的所有保护位(各 FCFLCS<BLPRO>位)均已被设置为"1"。

注：在上电后即进行的上电复位时，FCSECBIT <SECBIT>位即已被设置为"1"。

表 20-1 给出了该安全功能可提供限制的详细说明。

表 20-1 安全功能可提供的限制

项目	细节
1) ROM 数据读出	可从 CPU 读出数据。
2) 调试端口	JTAG/SW 的通信与跟踪被禁止
3) 针对闪存存储器的命令	禁止向闪存存储器写入命令。 旨在擦除各写入/擦除保护位内容的尝试，可擦除所有保护位。

20.3 寄存器

基址 = 0x41FF_F000

寄存器名称		地址(基本+)
保留	-	0x0000,0x0004
安全位寄存器	FCSECBIT	0x0010
保留	-	0x0014
闪存控制寄存器	FCFLCS	0x0020
保留	-	0x0024 ～ 0x0FFF

注：**禁止访问该"保留"区域。**

20.3.1 FCFLCS (闪存控制寄存器)

	31	30	29	28	27	26	25	24
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
比特符号	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
复位后	0	0	(注 2)	(注 2)	(注 2)	(注 2)	(注 2)	(注 2)
	15	14	13	12	11	10	9	8
比特符号	-	-	-	-	-	-	-	-
复位后	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
比特符号	-	-	-	-	-	-	-	RDY_BSY
复位后	0	0	0	0	0	0	0	1

位	比特符号	型号	功能
31-22	−	R	读作 0。
21-16	BLPRO53~ BLPRO0	R	块 5 ~ 0 的保护 0：被禁用 1：被启用 保护状态位 各保护位可表示对应存储块的保护状态。在某位被设置为"1"时，表示该位所对应的存储块处于受保护状态。在该存储块受保护时，无法将数据写入到该存储块。
17-1	−	R	读作 0。
0	RDY_BSY	R	就绪/忙(注 1) 0：自动运行 1：自动运行被终止。 就绪/忙标志位 该 RDY/BSY 输出用于监视自动运行的状态。该位是 CPU 的功能位，用于监视该功能。在闪存存储器处于自动运行状态时，会输出"0"表示其忙。在自动运行被终止时，其返回就绪状态，并输出"1"以接受下一条命令。如果自动运行失败，则该位保持"0"输出。通过进行硬件复位，其返回到"1"。

注 1：必须在就绪状态发布该命令。在忙状态发布该命令，可禁用正确命令传输与进一步的命令输入。执行系统复位，即可退出该状态。无论系统时钟频率为何，系统复位均至少需要 0.5 ms。在这种情况下，复位后启用读取耗时约 2 ms。

注 2：该值可随所应用保护的不同而变化。

20.3.2 FCSECBIT(安全位寄存器)

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

位	比特符号	型号	功能
31-1	−	R	读作 0。
0	SECBIT	R/W	安全位 0：被禁用 1：被启用

注：该寄存器已通过冷复位实现初始化。

20.4 写入与擦除

在单一芯片模式，单一引导模式与编写程序模式下，各写入与擦除保护位可用。

20.4.1 保护位

各保护位的写入应按存储块逐个进行。

在所有存储块的设置均为"1"时，必须在将 FCSECBIT <SECBIT> 位设置为"0"后再进行擦除。此时设置"1"可擦除所有保护位。应使用命令序列写入与擦除个保护位。

有关详细说明，见"闪存"一节。

20.4.2 安全位

上电后即进行的上电复位时，负责激活安全功能的 FCSECBIT <SECBIT>位即已被设置为"1"。应按以下程序重写该位。

1. 将代码 0xa74a9d23 写入到 FCSECBIT 寄存器。
2. 在 16 个时钟脉冲内写入数据根据第 1 条。

注：以上程序仅当使用 32-位数据传送命令时可被启用。

21. 调试接口

21.1 规格概述

TMPM370FYDFG/FYFG 带有串行线 JTAG 调试端口(SWJ-DP)单元，用于与该调试接口及跟踪输出用嵌入式跟踪宏单元™ (ETM)装置接口。经由片上跟踪端口接口单元(TPIU)，跟踪数据被输出到专用引脚(TRACEDATA[0] ~ [1], SWV)。

21.2 SWJ-DP 的特点

SWJ-DP 支持双引脚串行线调试端口 (SWDCK, SWDIO) 以及 JTAG 调试端口 (TDI, TDO, TMS, TCK, $\overline{\text{TRST}}$)。

21.3 ETM 的特点

ETM 支持两个数据信号引脚 (TRACEDATA[0] ~ [1])，一个时钟信号引脚 (TRACECLK)，以及来自 SWV 的跟踪输出。

21.4 引脚功能

该调试接口引脚也可用作通用端口。PB3 与 PB4 共用于 JTAG 调试端口功能与串行线调试端口功能。PB5 共用于 JTAG 调试端口功能与 SWV 跟踪输出功能。

表 21-1 SWJ-DP, ETM 功能

SWJ-DP 引脚名称	端口名称	JTAG 调试功能		SW 调试	
		I/O	说明	I/O	说明
TMS / SWDIO	PB3	输入	JTAG 测试模式选择	I/O	串行线数据输入/输出
TCK / SWCLK	PB4	输入	JTAG 测试检查	输入	串行线时钟脉冲
TDO / SWV	PB5	输出	JTAG 测试数据输出	(输入) (注 1)	(串行线查看器输出)
TDI	PB6	输入	JTAG 测试数据输入	-	-
$\overline{\text{TRST}}$	PB7	输入	JTAG 测试 RESET	-	-
TRACECLK	PB0	输出	跟踪时钟脉冲输出		
TRACEDATA0	PB1	输出	跟踪数据输出 0		
TRACEDATA1	PB2	输出	跟踪数据输出 1		

注：如果是启用 SWV 功能

复位后，PB3，PB4，PB5，PB6 与 PB7 即被配置为调试端口功能引脚。需按要求对其它调试接口引脚的功能进行编程。调试接口引脚可采用不使用调试接口的通用端口。

以下的表 21-2 汇总了复位后的各调试接口引脚功能，以及相关端口设置。

表 21-2 复位后的调试接口引脚与端口设置

初始设置	端口(位名称)	调试功能	复位后的端口设置(-;无寄存器)					
			功能 (PBFR)	输入 (PBIE)	输出 (PBCR)	开漏 (PBOD)	上拉 (PBPUP)	下拉 (PBPDN)
PORT	PB0	TRACECLK	0	0	0	0	0	0
PORT	PB1	TRACEDATA0	0	0	0	0	0	0
PORT	PB2	TRACEDATA1	0	0	0	0	0	0
DEBUG	PB3	TMS / SWDIO	1	1	1	0	1	0
DEBUG	PB4	TCK / SWCLK	1	1	0	0	0	1
DEBUG	PB5	TDO / SWV	1	0	1	0	0	0
DEBUG	PB6	TDI	1	1	0	0	1	0
DEBUG	PB7	$\overline{\text{TRST}}$	1	1	0	0	1	0

在采用低功耗模式时，需注意以下各点。

注 1：如果 PB3 与 PB5 被配置为调试功能引脚，则无论 CGSTBYCR<DRVE>的设置为何，即使在 STOP 模式下输出也仍可继续被启用。

注 2：如果 PB4 被配置为调试功能引脚，则其会阻止低功耗模式进入充分有效状态。如果未使用调试功能，则可配置 PB4 并使之用作通用端口。

21.5 与调试工具的连接

21.5.1 如何连接

有关调试工具的连接，请参看各制造厂商所推荐的采方法。各调试接口引脚均具备上拉或下拉寄存器。在与上拉或下拉寄存器连接时，必须注意其设置。

21.5.2 在使用通用端口时

在调试时，不要通过程序将调试接口设置更改为通用端口。然后，MCU 将不能控制来源于调试工具的信号，无法继续调试。按照调试接口引脚的使用情况，必须注意查看其设置。

表 21-3 调试接口

用法	使用调试接口(O:启用, -:禁用)							
	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA1	TRACE DATA0	TRACE CLK
JTAG+SW (RESET 后)	O	O	O	O	O	-	-	-
JTAG+SW (非 $\overline{\text{TRST}}$)	-	O	O	O	O	-	-	-
JTAG+TRACE	O	O	O	O	O	O	O	O
SW	-	-	-	O	O	-	-	-
SW+SWV	-	-	O	O	O	-	-	-
禁用调试功能	-	-	-	-	-	-	-	-

21.6 外围设备在 HALT 模式期间的运行

如在调试期间发生中断，则 Cortex-M3 CPU 核心进入 HALT 模式。看门狗定时器 (WDT) 自动停止计数。16 位计时器/计数器可指定 HALT 模式下的状态(继续运行或停止)。其它外围设备继续运行。

22. 电气特性

22.1 最大绝对额定值

参数		符号	额定值	单位
电源电压		DVDD5	-0.3 ~ 6	V
		DVDD5E	-0.3 ~ 6	
		RVDD5	-0.3 ~ 6	
		AVDD5A/B	-0.3 ~ 6	
		AMPVDD5	-0.3 ~ 6	
电容电压		VOUT15	−0.3 ~ 3	V
		VOUT3	−0.3 ~ 3.9	
输入电压		V_{IN}	−0.3 ~ VDD + 0.3 (注 2)	V
低电平输出电流	每个引脚	I_{OL}	5	mA
	合计	ΣI_{OL}	50	
高电平输出电流	每个引脚	I_{OH}	−5	
	合计	ΣI_{OH}	50	
功耗(Ta = 85 °C)		PD	600	mW
焊接温度(10 s)		T_{SOLDER}	260	°C
贮存温度		T_{STG}	−55 ~ 125	°C
工作温度	闪存 W/E 期间除外	T_{OPR}	−40 ~ 85	°C
	闪存 W/E 期间		0 ~ 70	

注 1：最大绝对额定值是在最坏情况下均不得超出的运行与环境条件的极限值。设备制造厂商所采用的设计，必须能确保不超过电流，电压，功耗，温度等方面的任何绝对最大额定值。如接触上列以外的条件，则可导致设备出现永久性损伤，或设备可靠性会受到影响，而这些因素均有可能加大因 IC 爆炸和/或燃烧而导致人身伤害的潜在风险。

注 2：**VDD = DVDD5E = DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5**

22.2 DC 电气特性(1/2)

DVSS = AVSSA = AVSSB = AMPVSS = 0V, Ta = −40 ～ 85 °C

参数		符号	额定值		最小	典型(注 1)	最大	单位
电源 电压 (注 2)	DVDD5 DVDD5E RVDD5 AVDD5A AVDD5B AMPVDD5	VDD	f_{OSC} = 8 ~ 10 MHz fsys = 1 ~ 80 MHz		4.5	−	5.5	V
电源 电压 (闪存 W/E 期间) (注 2)	DVDD5 DVDD5E RVDD5 AVDD5A AVDD5B AMPVDD5	VDD	f_{OSC} = 8 ~ 10 MHz fsys = 1 ~ 80 MHz (Ta (°C) = 0 ~ 70)		4.5	−	5.5	V
电源 电压 (上电或下电) (注 3)	DVDD5 DVDD5E RVDD5 AVDD5A AVDD5B AMPVDD5	VDD	f_{OSC} = 8 ~ 10 MHz fsys = 1 ~ 80 MHz		3.9	−	5.5	V
低电平输入 电压	施密特-输入	V_{IL1}	VDD = 4.5 V ~ 5.5 V (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 V ~ 5.5 V (注 4)		0.75VDD		VDD	V
VOUT15 与 VOUT3 的电容 (注 3)		C_{out}	RVDD5 = 4.5 V ~ 5.5 V VOUT15, VOUT3		3.3	−	4.7	μF
低电平输出电压		V_{OL}	I_{OL} = 1.6 mA	VDD ≥ 4.5 V (注 4)	−	−	0.4	V
高电平输出电压		V_{OH}	I_{OH} = −1.6 mA	VDD ≥ 4.5 V (注 4)	4.1	−	−	V
输入漏泄电流		I_{LI1}	0.0 ≤ VIN ≤ VDD (注 4)		−	0.02	±5	μA
输出泄漏电流		I_{LO}	0.2 ≤ VIN ≤ VDD -0.2 (注 4)		−	0.05	±10	
复位时的上拉电阻		R_{RST}	4.5 ≤ VDD ≤ 5.5 (注 4)		−	50	150	kΩ
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	−	μF
引脚电容 (电源引脚除外)		C_{IO}	fc = 1 MHz		−	−	10	pF

注 1：Ta = 25 °C, DVDD5 = DVDD5E = AVDD5A = AVDD5B = RVDD5 = AMPVDD5 = 5 V，但另有说明的情形除外。

注 2：DVDD5，DVDD5E，DVDD5A，DVDD5B，RVDD5 与 AMPVDD5 的电源电压必须相同。

注 3：其为上电或下电时的电压范围(在 VLTD 被禁用时)。在该范围内，电源线为 3.9 V ≤ VDD < 4,5 V，不保证 12-位 A/D 转换器，运算放大器/比较器，以及 AC 电气特性。有关详细说明，请参看图(上电序列 (仅使用上电复位))。

注 4：应将 VOUT15 与 VOUT3 引脚经由相同的电容值连接至 GND。IC 外部没有自 VOUT15 与 VOUT3 的电源。

注 5：VDD = DVDD5E = DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5

22.3 DC 电气特性(2/2)

DVDD5 = DVDD5E =RVDD5 = AVDD5A = AVDD5B = AMPVDD5 = 4.5 V ~ 5.5 V, Ta = −40 ~ 85 °C

参数	符号	额定值	最小	典型(注 1)	最大	单位
标准(注 2)齿轮 1/1	I_{DD}	fsys = 80 MHz	−	70	80	mA
IDLE (注 4) 齿轮 1/1			−	21	30	
停止		−	−	7	11	mA

注 1：Ta = 25 °C, DVDD5 = DVDD5E = AVDD5A = AVDD5B = RVDD5 = AMPVDD5 = 5 V，但另有说明除外。

注 2：I_{DD} NORMAL：

所有功能军运行不包括 A/D，运算放大器与比较器。

注 3：A/D 基准电压电源无法进入断路状态。

注 4：I_{DD} IDLE :

所有外围功能均已停止工作。

22.4 12-位 ADC 电气特性

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	最小	典型	最大	单位
模拟基准电压(+)		VREFHA VREFHB	−	−	AVDD	−	V
模拟输入电压		VAIN	−	AVSS	−	AVDD	V
模拟电源电流 (注 1),(注 2)		IREF	DVSS = AVSS	−	3.5	4.5	mA
电源电流 (注 1)	A/D 转换	−	IREF 除外	−	−	6.0	mA
INL 错误		−	AIN 电阻 ≤ 600 Ω AIN 负载电容 ≥ 0.1 μF 转换时间 ≥ 2 μs	−	−	± 6	LSB
DNL 错误				−	−	± 5	
补偿错误				−	−	± 5	
满标误差				−	−	± 5	
总误差				−	−	−10 ～ +5	

注 1：ADC 一个单元的电流。

注 2：A/D 基准电压电源无法进入断路状态。

注 3：1LSB = (AVDD − AVSS)/4096 [V]

注 4：AVDD = AVDD5A = AVDD5B, AVSS = AVSSA = AVSSB

注 5：该特性是在仅 ADC 工作的条件下测出的。

22.5 运算放大器放大器电气特性

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(注 3)		最小	典型	最大	单位
增益(注 1)		VGAIN	−		1.5	−	10	
输入电压范围		VAMPIN	−		(AVDD×0.1)/ VGAIN	−	(AVDD×0.9)/ VGAIN	V
补偿电压		VOFF1	VGAIN ≥ 3.5		−6 × VGAIN	−	+6 × VGAIN	mV
			VGAIN ≤ 3	Ta < 70 °C				
		VOFF2		Ta ≥ 70 °C	−20		+20	
增益误差		−	−		−	± 1	± 3	%
转换速率 (注 2)		Vthr	5pF, VGAIN = ×2.5		2	−	−	V /μs
电源电流	运算放大器 (每一个单元)	−	−		−	−	6	mA

注 1：可通过寄存器设置在×2.5，×3，×3.5，×4，×6 与×8 之中选择增益。

注 2：转换速率指的是放大器输出达到 AVDD-0.001×AVDD 时的倾斜度。

注 3：AVDD = AVDD5A = AVDD5B = 4.5 ~ 5.5V, AVSS = AVSSA = AVSSB = 0 V

22.6 比较器电气特性

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(注 3)	最小	典型	最大	单位
补偿电压		VOFF	−	±4	−	mV	
AIN 输入电压范围		VIN	−	AVSS	−	AVDD	V
基准电压范围		VREF		0.9	−	AVDD−0.2	V
响应时间 (注 1), (注 2)				−	−	1	μS
电源电流	比较器 (每一个单元)	−		−	−	0.75	mA

注 1：1.0 V ≤ VREF ≤ AVDD − 0.2 V

注 2：指 VIN 的变化范围 VREF−100 mV ~ VREF+100 mV，或 VREF+100 mV ~ VREF−100 mV。

注 3：AVDD = AVDD5A = AVDD5B = 4.5 ~ 5.5 V, AVSS = AVSSA = AVSSB = 0 V

22.7 AC 电气特性

22.7.1 AC 测量条件

AC 测量条件

- 输出电平：高= 0.8 × VDD / 低 = 0.2 × VDD
- 输入电平：请参看 DC 电气特性一节所给出的低电平输入电压与高电平输入电压。
- 负荷能力：CL = 30 pF

注：VDD = DVDD5E = DVDD5 = AVDD5A = AVDD5B = AMPVDD5

22.7.2 串行通道时序(SIO/UART)

22.7.2.1 I/O 接口模式(VDD=4.5 V ~ 5.5 V)

在下表中，字母 x 表示该系统时钟 (fsys) 的周期。其可随时钟脉冲齿轮功能编程的不同而变化。

(1) SCLK 输入模式 (Ta = −40 ~ 85 °C)

[输入]

参数	符号	方程		80 MHz		单位
		最小	最大	最小	最大	
SCLK 时钟高宽度(输入)	t_{SCH}	34x	−	37.5	−	ns
SCLK 时钟低宽度(输入)	t_{SCL}	3x	−	37.5	−	
SCLK 周期	t_{SCY}	$t_{SCH} + t_{SCL}$	−	75	−	
输入数据有效 SCLK 上升或下降(注 1)	t_{SRD}	30	−	30	−	
输入数据在 SCLK 上升后保持或下降(注 1)	t_{HSR}	x + 30	−	42.5	−	

[输出]

参数	符号	方程		80 MHz		单位
		最小	最大	最小	最大	
SCLK 时钟高宽度(输入)	t_{SCH}	3x	−	37.5 (注 3)	−	ns
SCLK 时钟低宽度(输入)	t_{SCL}	3x	−	37.5 (注 3)	−	
SCLK 周期	t_{SCY}	$t_{SCH} + t_{SCL}$	−	75	−	
输出数据至 SCLK 上升或下降(注 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$ (注 2)	−	0 (注 2)	−	
输入数据在 SCLK 上升后保持或下降(注 1)	t_{OHS}	$t_{SCY}/2$	−	37.5	−	

注 1：SCLK 上升或下降：

相对于 SCLK 的已编程活动边测出。

注 2：计算值应采用未被减去的范围的 SCLK 周期。

注 3：t_{OSS} 给出的是未被减去的最小值。

(2) SCLK 输出模式(Ta = −40 ~ 85°C)

[输出]

参数		符号	方程		80 MHz		单位
			最小	最大	最小	最大	
SCK 周期(可编程)		t_{SCY}	4x	−	60	−	ns
输出数据	←SCK 上升	t_{OSS}	t_{SCY} /2 − 30 (注 1)	−	0 (注 2)	−	
SCK 上升	→输出数据保持	t_{OHS}	t_{SCY} /2 − 30 (注 1)	−	0 (注 2)	−	
有效数据输入	←SCK 上升	t_{SRD}	45	−	45	−	
SCK 上升	→输入数据保持	t_{HSR}	0	−	0	−	

注 1：计算值应采用未被减去的范围的 SCLK 周期。

注 2：t_{OSS} 给出的是未被减去的最小值。

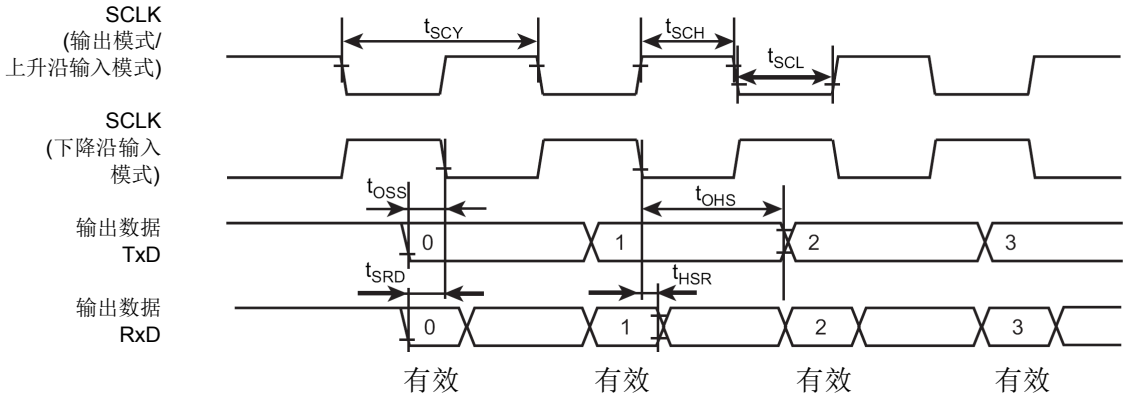


图 22-1 串行通道时序(SIO/UART)

22.7.3 事件计数器

字符 x 表示 TMRB 时钟脉冲的周期。TMRB 的时钟脉冲与系统时钟(fsys)的周期相同。其可随时钟脉冲齿轮功能编程的不同而变化。

Ta = −40 ~ 85 °C (1 ~ 80 MHz)

参数	符号	方程		80 MHz		单位
		最小	最大	最小	最大	
时钟低脉冲宽度	t_{VCKL}	2x + 100	−	125	−	ns
时钟高脉冲宽度	t_{VCKH}	2x + 100	−	125	−	ns

22.7.4 捕捉

字符 x 表示 TMRB 时钟脉冲的周期。TMRB 的时钟脉冲与系统时钟(fsys)的周期相同。其可随时钟脉冲齿轮功能编程的不同而变化。

Ta = −40 ~ 85 °C (1 ~ 80 MHz)

参数	符号	方程		80 MHz		单位
		最小	最大	最小	最大	
低脉冲宽度	t_{CPL}	2x + 100	−	125	−	ns
高脉冲宽度	t_{CPH}	2x + 100	−	125	−	ns

22.7.5 外部中断

在下表中，字母 x 表示该系统时钟(fsys)的周期。

Ta= −40 ~ 85 °C (1 ~ 80 MHz)

1. STOP 解除中断除外

参数	符号	方程		80 MHz		单位
		最小	最大	最小	最大	
INT0~ F 的低脉冲宽度	t_{INTAL}	x + 100	−	112.5	−	ns
INT0 ~ F 的高脉冲宽度	t_{INTAH}	x + 100	−	112.5	−	ns

2. STOP 解除中断

参数	符号	方程		80 MHz		单位
		最小	最大	最小	最大	
INT0~ F 的低脉冲宽度	t_{INTBL}	100	−	100	−	ns
INT0 ~ F 的高脉冲宽度	t_{INTBH}	100	−	100	−	ns

22.7.6 调试通信

22.7.6.1 AC 测量条件

- 输出电平：高 = 0.7 × DVDD5, 低 = 0.3 × DVDD5
- 负载电容：CL(TRACECLK) = 25pF, CL(TRACEDATA) = 20pF

22.7.6.2 SWD 接口

参数	符号	最小	最大	单位
CLK 周期	T_{dck}	100	-	ns
CLK 上升后的 DATA 保持	T_{d1}	4	-	
CLK 上升后的 DATA 有效	T_{d2}	-	37	
DATA 有效至 CLK 上升	T_{ds}	20	-	
CLK 下降后的 DATA 保持	T_{dh}	15	-	

22.7.6.3 JTAG 接口

参数	符号	最小	最大	单位
CLK 周期	T_{dck}	100	-	ns
CLK 下降后的 DATA 保持	T_{d3}	4	-	
CLK 下降后的 DATA 有效	T_{d4}	-	37	
DATA 有效至 CLK 上升	T_{ds}	20	-	
CLK 上升后的 DATA 保持	T_{dh}	15	-	

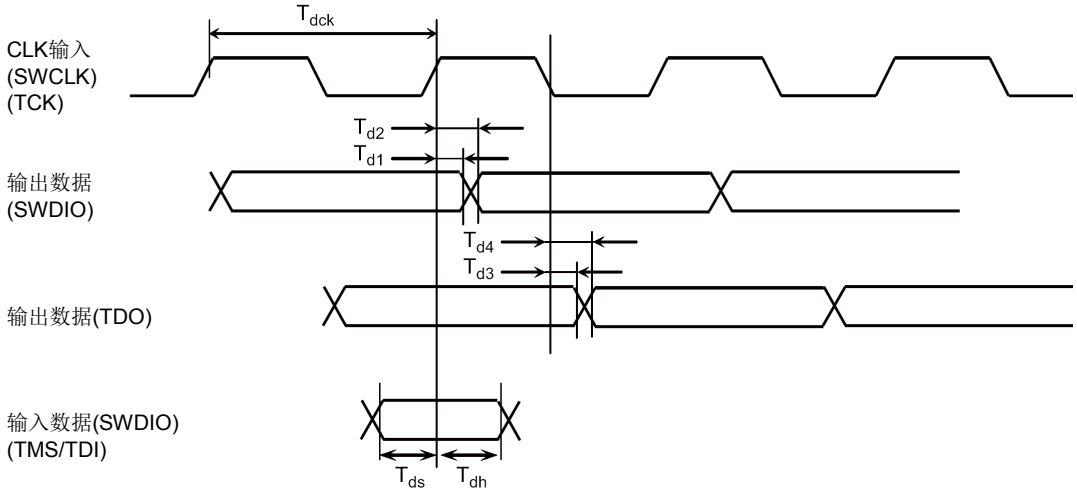


图 22-2 JTAG 与 SWD 通信时序

22.7.7 TRACE 输出

AC 测量条件

- 输出电平：高 = 0.7 × DVDD5, 低 = 0.3 × DVDD5
- 负载电容：CL(TRACECLK) = 25pF, CL(TRACEDATA) = 20pF

参数	符号	最小	最大	单位
TRACECLK 周期	t_{tclk}	25	−	ns
CLK 上升后的 DATA 有效	t_{setupr}	2	−	
CLK 上升后的 DATA 保持	t_{holdr}	1	−	
CLK 下降后的 DATA 有效	t_{setupf}	2	−	
CLK 下降后的 DATA 保持	t_{holdf}	1	−	

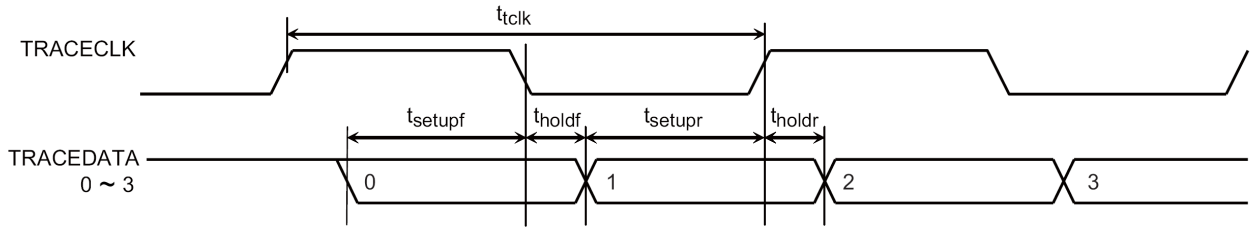


图 22-3 TRACE 通信时序

22.7.8 闪存特性

参数	额定值	最小	典型	最大	单位
闪存存储器 重写保证	Ta = 0 ~ 70 °C DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5 ~ 5.5 V	−	−	100	次

22.8 振荡电路

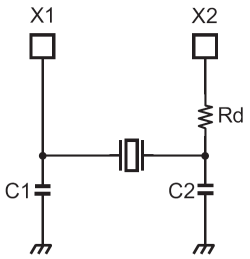


图 22-4 高频率振荡连接

注 1：该振荡器的负荷值，是各负荷(C1 与 C2)与实际装配板浮动负荷之和。在使用下表所列的 C1 与 C2 值时，有可能出现操作误差。在设计该板时，应确保该振荡器周围的最小长度模式。此外，我们建议采用实际板进行振荡器评价。

注 2：不要用外部驱动器驱动 X1/X2。

以下振荡器卖方已对 TX03 进行过评价。在选择外部零件时应使用该信息。

22.8.1 建议采用的陶瓷振荡器

TX03 建议采用村田制造有限公司出品的高频振荡器。

有关详细说明请参看以下 URL。

http://www.murata.co.jp

22.9 上电说明

上电时端口 L(PL0 与 PL1 引脚)使用说明

上电时，在 VDD 达到工作电压并 200 μs 消逝之前，端口 L(PL0 与 PL1 引脚)必须处于 OPEN 状态或提供"低"电平(低于 0.5 V)。

需同样测量并确定供电电压在运行期间下降，上电复位线路生成了复位信号，且供电线路重新上升。

注：VDD = DVDD5 = RVDD5 = DVDD5A = DVDD5B = DVDD5E = AMPVDD5

22.9.1 仅使用上电复位

注 1：利用内置上电复位启动电源时，DVDD5 与 RVDD5 端子部位的电源应 3ms 内达到建议工作电压范围 (3.9 ~ 5.5 V)。

注 2：在微电脑在电压检测器线路(VLTD)中开始运行之后，请选择任意忽略电平，并启用运行。

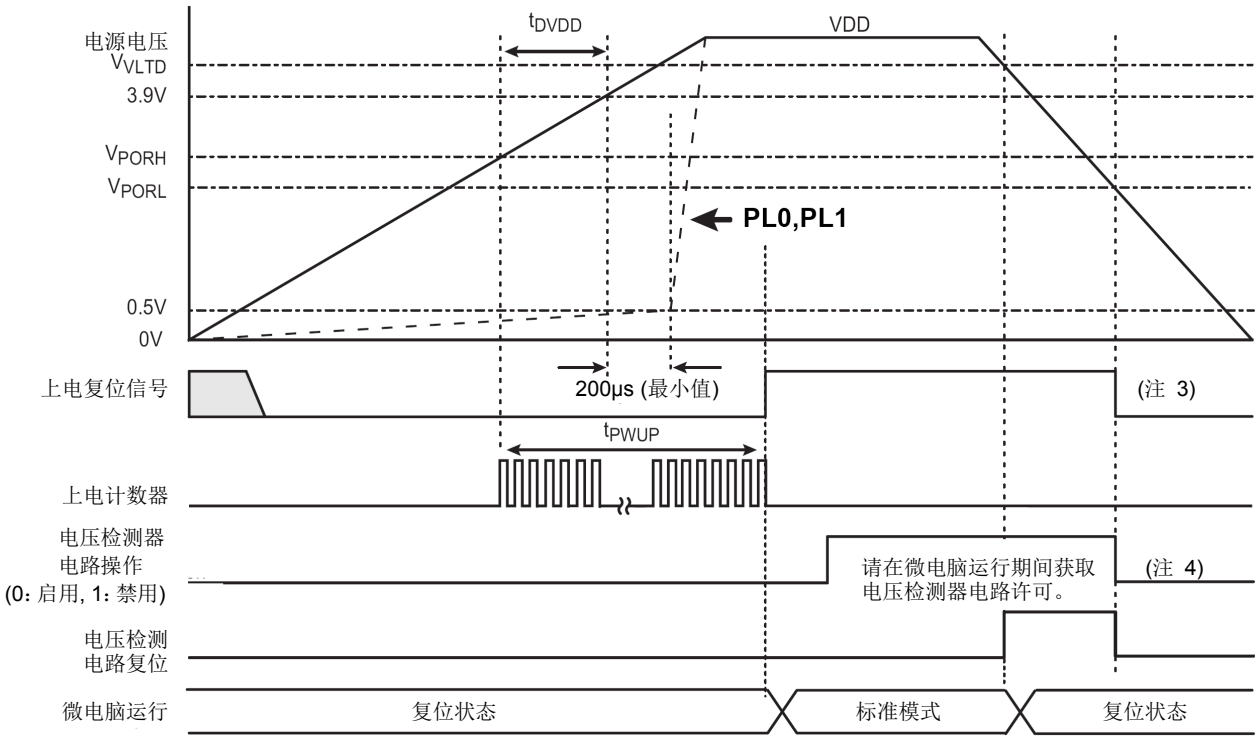


图 22-5 上电序列(仅使用上电复位)

注 1：VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5

注 2：由于上电复位解除电压(V_{PORH})与上电复位检测电压(V_{PORL})被相对改变，检测电压不会使其反向。

注 3：如果供电电压变成 V_{PORL} 或以下，则可开始上电复位。

注 4：通过上电复位生成，实现电压检测器电路(VLTD)的初始化。

22.9.2 使用外部复位

22.9.2.1 如果外部复位时间短于 POR

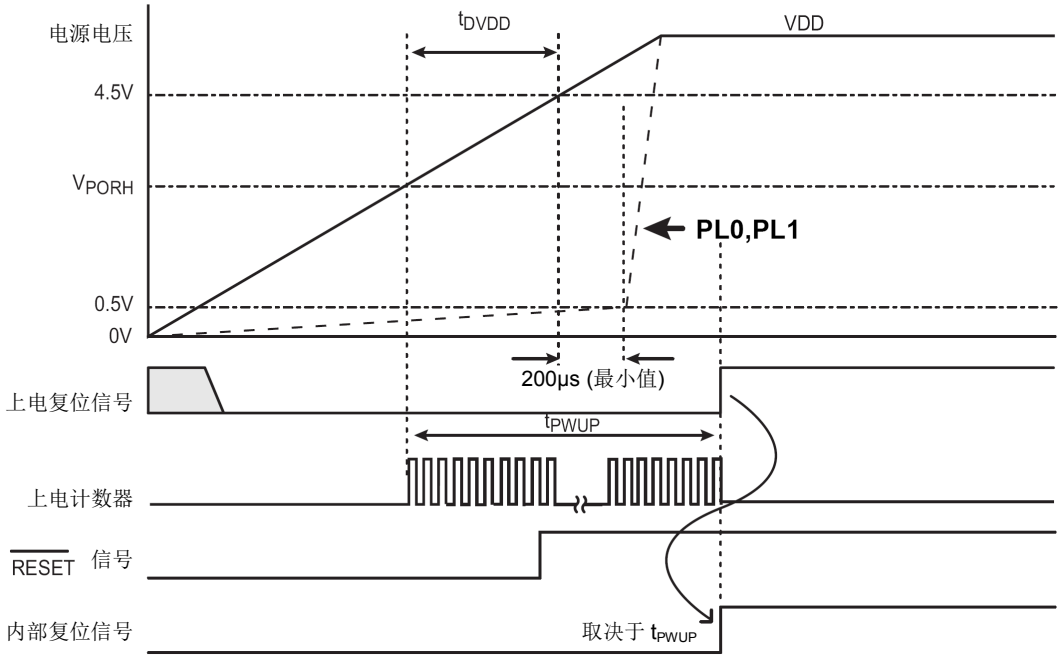


图 22-6 P 上电序列 (使用 POR 与外部复位) (1)

注：VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5

22.9.2.2 如果外部复位时间长于 t_{PWUP}

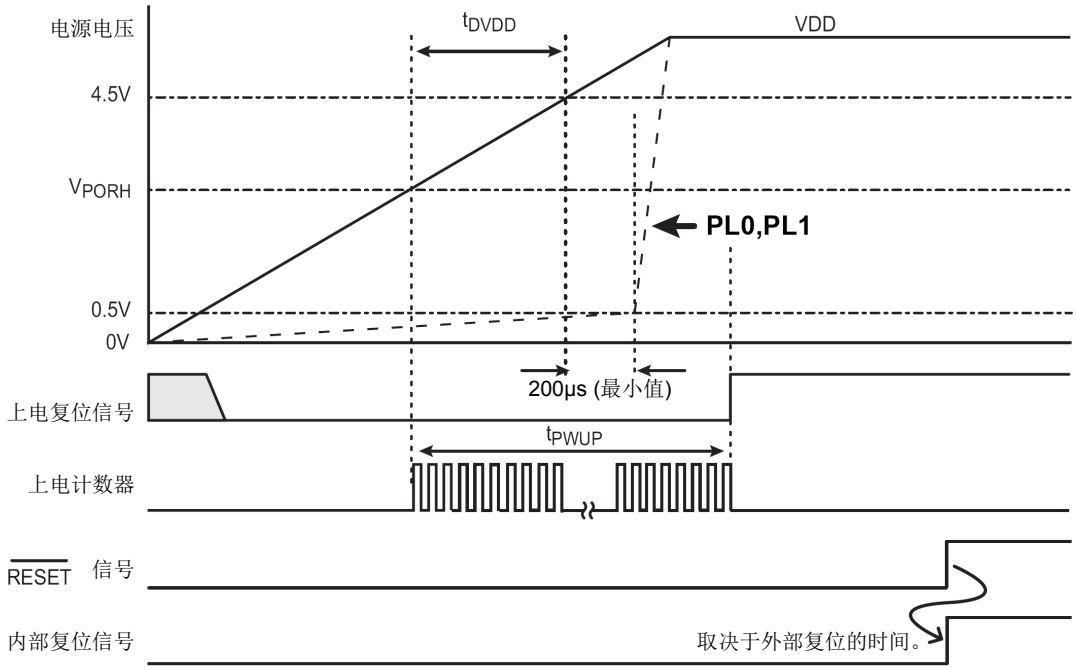


图 22-7 上电序列(使用 POR 与外部复位) (2)

注 1：VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5

22.9.2.3 如果电源线的上升时间长于 t_{PWUP}

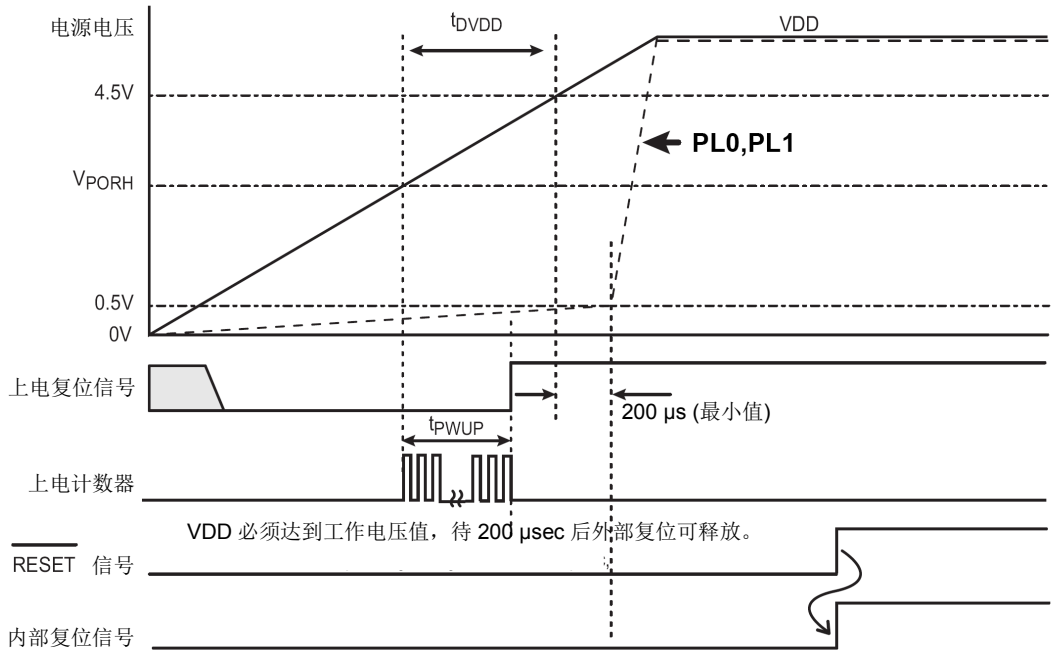


图 22-8 上电序列(t_{DVDD}> t_{PWUP})

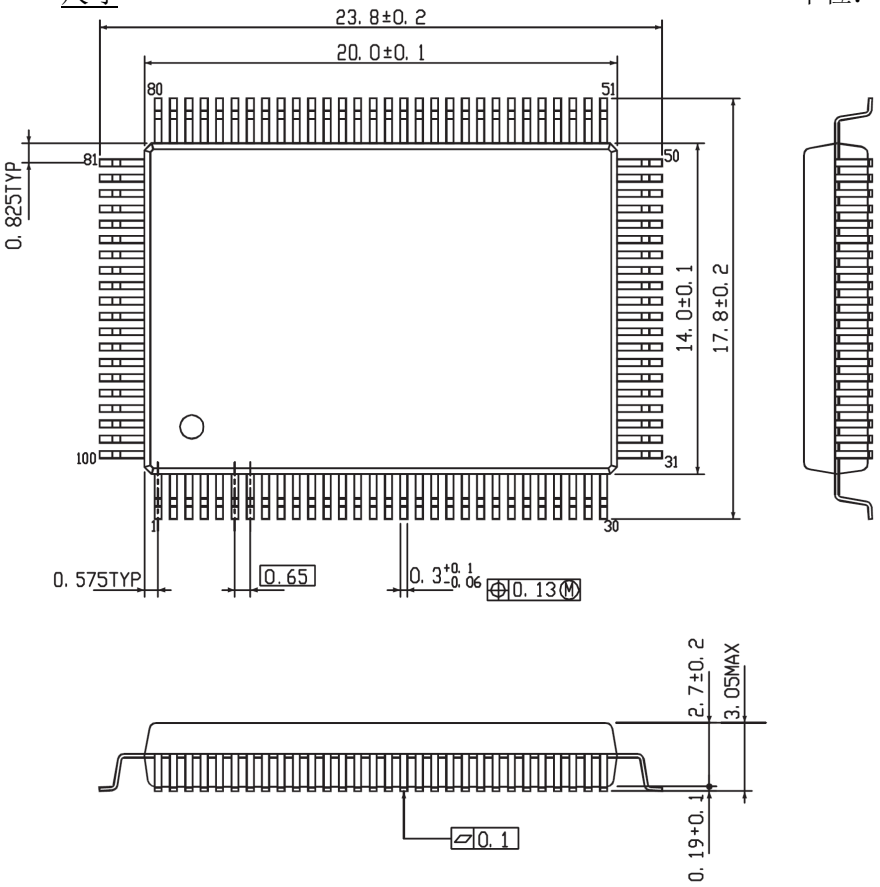
注 1：VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B = AMPVDD5

注 2：在这种情况下，必须从 $\overline{\text{RESET}}$ 引脚复位。

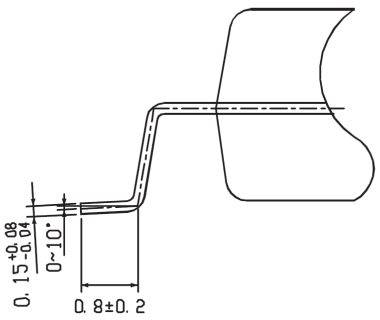
23. 封装尺寸

23.1 型号: P-QFP100-1420-0.65Q

尺寸 单位：mm

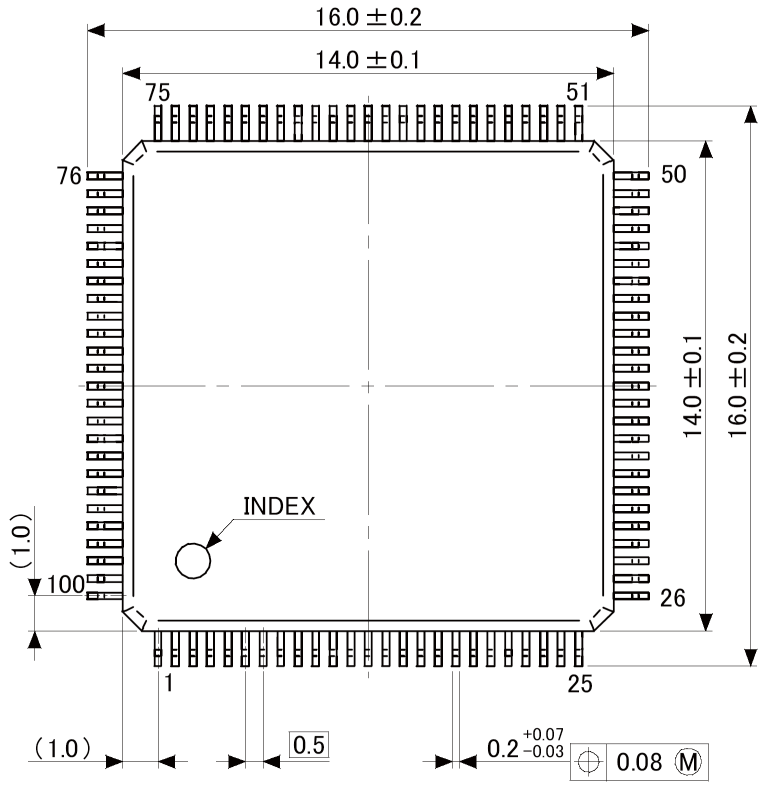


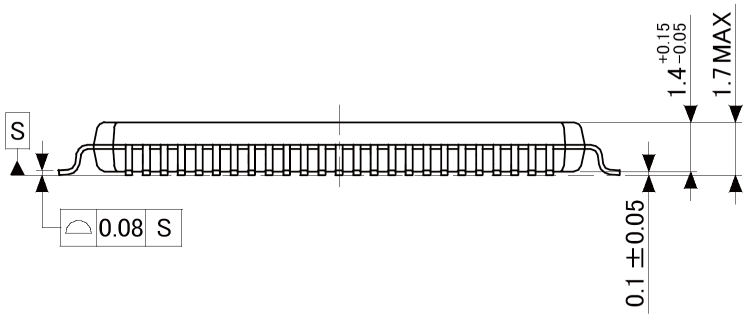
引脚详图



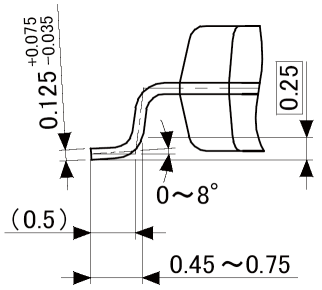
23.2 型号: P-LQFP100-1414-0.50H

尺寸 单位：mm





引脚详图



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.**