



# CMS32M67xx用户手册

**基于ARM® Cortex®-M0+的超低功耗32位微控制器**

**V0.5.1**

请注意以下有关CMS知识产权政策

\* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

\* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

\* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 [www.mcu.com.cn](http://www.mcu.com.cn)

## 文档使用说明

本手册是CMS32M67xx微控制器产品的技术参考手册，技术参考手册是有关如何使用本系列产品的应用说明资料，包含各个功能模块的结构、功能描述、工作模式以及寄存器配置等详细信息,并对每种功能模块都有专门的章节进行介绍。

技术参考手册是针对这一系列产品所有功能模块的说明，若要了解特定型号产品的特征说明（即功能搭载情况），可参考相应的数据手册。

数据手册信息如下：

CMS32M67xx: CMS32M67xx\_datasheet\_vx.x.x.pdf

通常在芯片选型的初期，首先要看数据手册，以评估该产品是否能够满足设计上的功能需求；在基本选定所需产品后，需要查看技术参考手册，以确定各功能模块的工作模式是否符合要求；在确定选型进入编程设计阶段时，需要详细阅读技术参考手册，以获知各项功能的具体实现方式和寄存器配置。在设计硬件时可参考数据手册以获得电压，电流，驱动能力以及管脚分配等信息。

关于Cortex-M0+核心、SysTick定时器和NVIC的详细说明，请参照对应ARM的文档。

## 目录

文档使用说明.....	2
<b>第1章 CPU .....</b>	<b>17</b>
1.1 概述 .....	17
1.2 Cortex-M0+内核特性.....	17
1.3 调试特性 .....	17
1.4 SWD 接口引脚 .....	19
1.5 ARM 参考文档.....	20
<b>第2章 引脚功能.....</b>	<b>21</b>
2.1 端口功能 .....	21
2.2 端口复用功能 .....	21
2.3 寄存器映射.....	24
2.3.1 控制功能寄存器映射 .....	24
2.3.2 输出输入复用功能寄存器映射 .....	26
2.3.3 特殊功能端口 RESINB 控制寄存器映射.....	28
2.4 寄存器说明.....	28
2.4.1 端口寄存器 (Px) .....	29
2.4.2 端口模式寄存器 (PMx) .....	30
2.4.3 上拉电阻选择寄存器 (PUx) .....	31
2.4.4 下拉电阻选择寄存器 (PDx) .....	32
2.4.5 端口输出模式寄存器 (POMx) .....	33
2.4.6 端口模式控制寄存器 (PMCx) .....	34
2.4.7 端口置位控制寄存器 (PSETx) .....	35
2.4.8 端口清零控制寄存器 (PCLR <sub>x</sub> ) .....	36
2.4.9 端口输出复用功能配置寄存器 (P <sub>mn</sub> CFG) .....	37
2.4.10 端口输入复用功能配置寄存器 (PS <sub>xx</sub> _CFG) .....	38
2.4.11 TTL 和施密特输入选择(P0TTLCFG).....	43
2.4.12 特殊功能端口 RESINB 说明(RSTM).....	43
<b>第3章 系统结构.....</b>	<b>44</b>
3.1 概述 .....	44
3.2 系统地址划分 .....	45
<b>第4章 时钟发生电路 .....</b>	<b>48</b>
4.1 时钟发生电路的功能.....	48
4.2 时钟发生电路的结构.....	49
4.3 寄存器映射.....	51
4.4 寄存器说明.....	52
4.4.1 系统时钟控制寄存器 (CKC) .....	52
4.4.2 时钟运行状态控制寄存器 (CSC) .....	53
4.4.3 PLL 控制寄存器(PLLCR).....	53
4.4.4 系统时钟主控制寄存器(MCKC).....	54

4.4.5	时钟运行模式控制寄存器(CMC)	54
4.4.6	振荡稳定时间选择寄存器(OSTS)	55
4.4.7	振荡稳定时间计数器的状态寄存器 (OSTC)	56
4.4.8	外围允许寄存器 (PER0、PER10、PER11、PER12、PER13)	57
4.4.9	12 位间隔定时器运行时钟选择寄存器 (OSMC)	60
4.4.10	高速内部振荡器的频率选择寄存器 (HOCODIV)	60
4.4.11	低速内部振荡器时钟选择寄存器 (SUBCKSEL)	61
4.4.12	供电模式控制保护寄存器 (PMUKEY)	61
4.4.13	供电模式控制寄存器 (PMUCTL)	61
4.5	系统时钟振荡电路	62
4.5.1	X1 振荡电路	62
4.5.2	高速内部振荡器	62
4.5.3	低速内部振荡器	63
4.6	时钟发生电路的运行	64
4.7	时钟控制	65
4.7.1	高速内部振荡器的设置例子	65
4.7.2	X1 振荡电路的设置例子	66
4.7.3	CPU 时钟的状态转移图	67
4.7.4	CPU 时钟转移前的条件和转移后的处理	70
4.7.5	CPU 时钟和主系统时钟的切换所需时间	71
4.7.6	时钟振荡停止前的条件	71
<b>第5章</b>	<b>通用定时器单元的功能</b>	<b>72</b>
5.1	通用定时器的功能	72
5.1.1	独立通道运行功能	72
5.1.2	多通道联动运行功能	74
5.1.3	8 位定时器运行功能 (只限于通道 1 和通道3)	74
5.2	通用定时器单元的结构	75
5.2.1	寄存器映射	77
5.2.2	定时器计数寄存器 mn (TCRmn)	78
5.2.3	定时器数据寄存器 mn (TDRmn)	79
5.2.4	外围允许寄存器 0 (PER0)	80
5.2.5	定时器时钟选择寄存器 m (TPSm)	81
5.2.6	定时器模式寄存器 mn (TMRmn)	83
5.2.7	定时器状态寄存器 mn (TSRmn)	91
5.2.8	定时器通道允许状态寄存器 m (TEm)	92
5.2.9	定时器通道开始寄存器 m (TSM)	93
5.2.10	定时器通道停止寄存器 m (TTm)	94
5.2.11	定时器输入输出选择寄存器 (TIOS0)	95
5.2.12	定时器输出允许寄存器 m (TOEm)	96
5.2.13	定时器输出寄存器 m (TOM)	97
5.2.14	定时器输出电平寄存器 m (TOLm)	98
5.2.15	定时器输出模式寄存器 m (TOMm)	99
5.2.16	控制定时器输入/输出引脚端口功能的寄存器	100



5.3 通用定时器单元的基本规则 .....	101
5.3.1 多通道联动运行功能的基本规则 .....	101
5.3.2 8位定时器运行功能的基本规则（只限于通道1和通道3） .....	103
5.4 计数器的运行 .....	104
5.4.1 计数时钟 ( $f_{TCLK}$ ) .....	104
5.4.2 计数器的开始时序 .....	106
5.4.3 计数器的运行 .....	107
5.5 通道输出 (TOmn 引脚) 的控制 .....	112
5.5.1 TOmn 引脚输出电路的结构 .....	112
5.5.2 TOmn 引脚的输出设定 .....	113
5.5.3 通道输出运行的注意事项 .....	114
5.5.4 TOmn 位的一次性操作 .....	118
5.5.5 有关开始计数时的定时器中断和 TOmn 引脚输出 .....	119
5.6 定时器输入 (TI mn) 的控制 .....	120
5.6.1 TI mn 引脚输入电路的结构 .....	120
5.6.2 操作通道输入时的注意事项 .....	120
5.7 通用定时器单元的独立通道运行功能 .....	121
5.7.1 作为间隔定时器/方波输出的运行 .....	121
5.7.2 作为外部事件计数器的运行 .....	125
5.7.3 作为分频器的运行 .....	128
5.7.4 作为输入脉冲间隔测量的运行 .....	131
5.7.5 作为输入信号高低电平宽度测量的运行 .....	134
5.7.6 作为延迟计数器的运行 .....	138
5.8 通用定时器单元的多通道联动运行功能 .....	141
5.8.1 作为单触发脉冲输出功能的运行 .....	141
5.8.2 作为 PWM 功能的运行 .....	147
5.8.3 作为多重 PWM 输出功能的运行 .....	154
<b>第6章 LSITIMER 12位间隔定时器 .....</b>	<b>162</b>
6.1 12位间隔定时器的功能 .....	162
6.2 12位间隔定时器的结构 .....	162
6.3 寄存器映射 .....	162
6.4 12位间隔定时器的控制寄存器 (CON0) .....	162
6.5 12位间隔定时器的运行 .....	164
6.5.1 12位间隔定时器的运行时序 .....	164
6.5.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移 .....	165
<b>第7章 时钟输出控制电路 .....</b>	<b>166</b>
7.1 时钟输控制电路的功能 .....	166
7.2 时钟输出控制电路的结构 .....	167
7.3 寄存器映射 .....	167
7.3.1 时钟输出选择寄存器 (CKS0) .....	168
7.4 配置时钟输出端口功能的寄存器 .....	169

7.5 时钟输出控制电路的运行 .....	170
7.5.1 输出引脚的运行 .....	170
7.6 时钟输出控制电路的注意事项 .....	170
<b>第8章 看门狗定时器 .....</b>	<b>171</b>
8.1 看门狗定时器的功能 .....	171
8.2 看门狗定时器的结构 .....	171
8.3 寄存器映射 .....	173
8.3.1 看门狗定时器的允许寄存器 (WDTE) .....	173
8.3.2 LOCKUP 控制寄存器 (LOCKCTL) .....	174
8.3.3 保护寄存器 (PRCR) .....	174
8.3.4 看门狗配置寄存器 WDTCFGx(x=0~3) .....	175
8.4 看门狗定时器的运行 .....	176
8.4.1 看门狗定时器的运行控制 .....	176
8.4.2 看门狗定时器上溢时间的设定 .....	177
8.4.3 看门狗定时器窗口打开期间的设定 .....	178
8.4.4 看门狗定时器间隔中断的设定 .....	179
8.4.5 LOCKUP 期间看门狗定时器的运行 .....	179
<b>第9章 除法与开方运算单元 (DIVSQRT) .....</b>	<b>180</b>
9.1 概述 .....	180
9.2 特性 .....	180
9.3 功能说明 .....	180
9.4 寄存器映射 .....	181
9.5 寄存器说明 .....	182
9.5.1 除法开方器控制寄存器(CON) .....	182
9.5.2 除法开方器数据 A 寄存器(ALUA) .....	182
9.5.3 除法开方器数据 B 寄存器(ALUB) .....	182
9.5.4 除法开方器结果 0 寄存器(RES0) .....	182
9.5.5 除法开方器结果 1 寄存器(RES1) .....	182
<b>第10章 除法运算单元 (DIV) .....</b>	<b>183</b>
10.1 概述 .....	183
10.2 特性 .....	183
10.3 功能说明 .....	183
10.4 寄存器映射 .....	183
10.5 寄存器说明 .....	184
10.5.1 除法控制寄存器(CON) .....	184
10.5.2 除法器数据 A 寄存器(ALUA) .....	184
10.5.3 除法器数据 B 寄存器(ALUB) .....	184
10.5.4 除法器结果 0 寄存器(RES0) .....	184
10.5.5 除法器结果 1 寄存器(RES1) .....	184
<b>第11章 定时器 (TIMER0/1) .....</b>	<b>185</b>
11.1 概述 .....	185

11.2 特性 .....	185
11.3 功能描述 .....	185
11.3.1 单次触发模式 .....	185
11.3.2 周期计数模式 .....	185
11.3.3 连续计数模式 .....	185
11.3.4 延迟加载功能 .....	185
11.4 寄存器映射 .....	186
11.5 寄存器说明 .....	186
11.5.1 定时器控制寄存器 (CON0/1) .....	186
11.5.2 定时器加载寄存器 (LOAD0/1) .....	187
11.5.3 定时器当前值寄存器 (VAL0/1) .....	187
11.5.4 定时器中断源状态寄存器 (RIS0/1) .....	187
11.5.5 定时器已使能中断状态寄存器 (MIS0/1) .....	187
11.5.6 定时器中断清零寄存器 (ICLR0/1) .....	187
11.5.7 定时器延迟加载寄存器 (BGLOAD0/1) .....	187
<b>第12章 捕捉/比较/脉宽调制模块(CCP0/1) .....</b>	<b>188</b>
12.1 概述 .....	188
12.2 特性 .....	188
12.3 功能描述 .....	189
12.3.1 脉宽调制模式 (PWM) .....	189
12.3.2 方波输出模式 .....	190
12.3.3 捕捉模式 0 .....	190
12.3.4 捕捉模式 1 .....	191
12.3.5 捕捉模式 2 .....	193
12.3.6 PWM 配置过程 .....	194
12.3.7 中断 .....	194
12.4 寄存器映射 .....	195
12.5 寄存器说明 .....	196
12.5.1 CCP0 控制寄存器 (CCP0CON) .....	196
12.5.2 CCP0 重加载寄存器 (CCP0LOAD) .....	197
12.5.3 CCP0A 数据寄存器 (CCP0DA) .....	197
12.5.4 CCP0B 数据寄存器 (CCP0DB) .....	197
12.5.5 CCP1 控制寄存器 (CCP1CON) .....	198
12.5.6 CCP1 重加载寄存器 (CCP1LOAD) .....	199
12.5.7 CCP1A 数据寄存器 (CCP1DA) .....	199
12.5.8 CCP1B 数据寄存器 (CCP1DB) .....	199
12.5.9 CCP 中断使能寄存器 (CCPIMSC) .....	199
12.5.10 CCP 中断源状态寄存器 (CCPRIS) .....	200
12.5.11 CCP 已使能中断状态寄存器 (CCPMIS) .....	201
12.5.12 CCP 中断清零寄存器 (CCPICLR) .....	202
12.5.13 CCP 运行寄存器 (CCPRUN) .....	202
12.5.14 CCP 写使能控制寄存器 (LOCK) .....	202

12.5.15	CCP CAP 控制寄存器 (CAPCON)	203
12.5.16	CCP CAP 通道选择寄存器 (CAPCHS)	205
12.5.17	CCP1 CAP 控制寄存器 2 (CAPCON2)	206
12.5.18	CCP1 CAP 控制寄存器 3 (CAPCON3)	206
12.5.19	CCP1 CAP0 数据寄存器 (CAP0DATA)	206
12.5.20	CCP1 CAPn 数据寄存器 (CAPnDATA) (n=1-3)	206
12.5.21	CCP1 捕获模式 2 下的捕获占空比寄存器 (CAPDUTY)	206
<b>第13章 HALL 信号处理模块</b>		<b>207</b>
13.1	概述	207
13.2	特性	207
13.3	功能描述	207
13.3.1	功能框图	207
13.3.2	时钟分频	207
13.3.3	信号输入	208
13.3.4	滤波器	208
13.3.5	捕获	208
13.4	寄存器映射	209
13.5	寄存器说明	210
13.5.1	HALL 模块配置寄存器 (CFG)	210
13.5.2	HALL 模块信息寄存器 (INFO)	210
13.5.3	HALL 信息清零寄存器 (CLRF)	211
13.5.4	HALL 模块宽度计数值寄存器 (WIDTH)	211
13.5.5	HALL 模块计数器门限值寄存器 (TH)	211
13.5.6	HALL 计数寄存器 (CNT)	211
13.5.7	HALL 写使能寄存器 (LOCK)	211
<b>第14章 增强型PWM(EPWM)</b>		<b>212</b>
14.1	概述	212
14.2	特性	212
14.3	功能描述	213
14.3.1	结构框图	214
14.3.2	时钟分频	214
14.3.3	独立输出模式	214
14.3.4	互补输出模式	215
14.3.5	同步输出模式	215
14.3.6	成组输出模式	215
14.3.7	加载更新模式	216
14.3.8	边沿对齐计数模式	218
14.3.9	中心对齐计数模式	219
14.3.10	独立计数器比较功能	222
14.3.11	可编程死区发生器	224
14.3.12	掩码及掩码预设功能	225
14.3.13	霍尔传感器接口功能	225

14.3.14 故障保护功能（刹车及恢复功能） .....	228
14.3.15 调试模式下的输出状态 .....	230
14.3.16 输出通道重映射功能 .....	230
14.3.17 EPWM 配置过程 .....	231
14.3.18 中断 .....	231
14.4 寄存器映射 .....	232
14.5 寄存器说明 .....	234
14.5.1 EPWM 预分频寄存器(CLKPSC) .....	234
14.5.2 EPWM 时钟选择寄存器(CLKDIV) .....	234
14.5.3 EPWM 控制寄存器(CON) .....	236
14.5.4 EPWM 控制寄存器(CON2) .....	238
14.5.5 EPWM 控制寄存器(CON3) .....	239
14.5.6 EPWM 控制寄存器(CON4) .....	241
14.5.7 EPWM 周期寄存器 0-7(PERIOD0-7) .....	241
14.5.8 EPWM 比较寄存器 0-7(CMPDAT0-7) .....	241
14.5.9 EPWM 输出控制寄存器(POEN) .....	242
14.5.10 EPWM 输出通道重映射使能寄存器 1(POREMAP_EN) .....	242
14.5.11 EPWM 输出通道重映射寄存器(POREMAP) .....	243
14.5.12 EPWM 故障保护控制寄存器(BRKCTL) .....	245
14.5.13 EPWM 死区长度寄存器(DTCTL01) .....	247
14.5.14 EPWM 死区长度寄存器(DTCTL23) .....	247
14.5.15 EPWM 死区长度寄存器(DTCTL45) .....	247
14.5.16 EPWM 死区长度寄存器(DTCTL67) .....	247
14.5.17 EPWM 掩码输出控制寄存器(MASK) .....	248
14.5.18 EPWM 掩码输出控制预设寄存器(MASKNXT) .....	250
14.5.19 EPWM 触发比较寄存器(CMPTGD0-1) .....	252
14.5.20 EPWM 中断使能寄存器(IMSC) .....	252
14.5.21 EPWM 中断使能寄存器 2(IMSC2) .....	252
14.5.22 EPWM 中断源状态寄存器(RIS) .....	253
14.5.23 EPWM 中断源状态寄存器 2(RIS2) .....	253
14.5.24 EPWM 已使能中断状态寄存器(MIS) .....	254
14.5.25 EPWM 已使能中断状态寄存器 2(MIS2) .....	254
14.5.26 EPWM 中断清零控制寄存器(ICLR) .....	254
14.5.27 EPWM 中断清零控制寄存器 2(ICLR2) .....	255
14.5.28 EPWM 中断累加控制寄存器(IFA) .....	255
14.5.29 EPWM 写使能控制寄存器(LOCK) .....	256
14.5.30 EPWM 故障保护恢复延时寄存器(BRKRDT) .....	256

## **第15章 通用异步收发器（UARTn n=0/1）概述 .....** **257**

15.1 特性 .....	257
15.2 功能描述 .....	257
15.2.1 UARTn 功能模式 .....	257
15.2.2 UARTn 中断和状态 .....	257
15.3 寄存器映射 .....	258

15.4 寄存器说明.....	258
15.4.1 接收缓存寄存器 (RBR) .....	258
15.4.2 发送缓存寄存器 (THR) .....	258
15.4.3 波特率分频寄存器 (DLR) .....	258
15.4.4 中断使能寄存器 (IER) .....	259
15.4.5 中断状态寄存器 (IIR) .....	259
15.4.6 线控制寄存器 (LCR) .....	260
15.4.7 Modem 控制寄存器 (MCR) .....	260
15.4.8 线状态寄存器 (LSR) .....	261
<b>第16章 串行接口IICA (IICA) .....</b>	<b>262</b>
16.1 串行接口 IICA 的功能.....	262
16.2 串行接口 IICA 的结构.....	265
16.3 寄存器映射.....	268
16.4 控制串行接口 IICA 的寄存器说明 .....	269
16.5 外围允许寄存器 0 (PER0) .....	270
16.5.1 IICA 控制寄存器 n0 (IICCTLn0) .....	271
16.5.2 IICA 状态寄存器 n (IICSn) .....	275
16.5.3 IICA 标志寄存器 n (IICFn) .....	278
16.5.4 IICA 控制寄存器 n1 (IICCTLn1) .....	280
16.5.5 IICA 低电平宽度设定寄存器 n (IICWLn) .....	282
16.5.6 IICA 高电平宽度设定寄存器 n (IICWHn) .....	282
16.5.7 控制 IICA 引脚端口功能的寄存器.....	282
16.6 I2C 总线模式的功能 .....	283
16.6.1 引脚结构.....	283
16.6.2 通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法 .....	284
16.7 I <sup>2</sup> C 总线的定义和控制方法 .....	286
16.7.1 开始条件.....	287
16.7.2 地址 .....	288
16.7.3 传送方向的指定.....	288
16.7.4 应答 (ACK) .....	289
16.7.5 停止条件.....	290
16.7.6 等待.....	291
16.7.7 等待的解除方法.....	293
16.7.8 中断请求 (INTIICAn) 的产生时序和等待控制.....	294
16.7.9 地址匹配的检测方法 .....	296
16.7.10 错误的检测 .....	296
16.7.11 扩展码 .....	296
16.7.12 仲裁.....	297
16.7.13 唤醒功能.....	299
16.7.14 通信预约.....	302
16.7.15 其他注意事项 .....	306
16.7.16 通信运行.....	307



16.7.17 I <sup>2</sup> C 中断请求 (INTIICAn) 的产生时序 .....	315
16.8 时序图.....	336
<b>第17章 串行外围接口控制器(SSP/SPI) .....</b>	<b>351</b>
17.1 概述 .....	351
17.2 特性 .....	351
17.3 寄存器映射.....	351
17.4 寄存器说明.....	352
17.4.1 SSP 控制寄存器 (CON) .....	352
17.4.2 SSP 状态寄存器 (STAT) .....	353
17.4.3 SSP 数据寄存器 (DAT) .....	353
17.4.4 SSP 时钟控制器 (CLK) .....	353
17.4.5 SSP 中断使能寄存器 (IMSC) .....	353
17.4.6 SSP 中断源状态寄存器 (RIS) .....	354
17.4.7 SSP 已使能中断状态寄存器 (MIS) .....	354
17.4.8 SSP 中断清零寄存器 (ICLR) .....	354
17.4.9 SSP 软件片选信号寄存器 (CSCR) .....	354
<b>第18章 增强型DMA .....</b>	<b>355</b>
18.1 DMA 的功能 .....	355
18.2 DMA 的结构 .....	357
18.3 寄存器映射.....	358
18.4 控制 DMA 的寄存器.....	359
18.4.1 DMA 控制数据区和 DMA 向量表区的分配 .....	359
18.4.2 控制数据的分配.....	360
18.4.3 向量表 .....	362
18.4.4 外围允许寄存器 10 (PER10) .....	364
18.4.5 DMA 触发源选择寄存器 (DMATGS) .....	364
18.4.6 触发源信号触发 DMA 使能寄存器 0 (TGSEN0) .....	365
18.4.7 触发源信号触发 DMA 使能寄存器 1 (TGSEN1) .....	365
18.4.8 触发源信号触发 DMA 使能寄存器 2 (TGSEN2) .....	366
18.4.9 DMA 控制寄存器 j (DMACRj) (j=0~7) .....	367
18.4.10 DMA 块大小寄存器 j (DMBLSj) (j=0~7) .....	368
18.4.11 DMA 传送次数寄存器 j (DMACTj) (j=0~7) .....	369
18.4.12 DMA 传送次数重加载寄存器 j (DMRLDj) (j=0~7) .....	370
18.4.13 DMA 源地址寄存器 j (DMSARj) (j=0~7) .....	371
18.4.14 DMA 目标地址寄存器 j (DMDARj) (j=0~7) .....	371
18.4.15 DMA 启动允许寄存器 i (DMAEN0) .....	372
18.4.16 DMA 基址寄存器 (DMABAR) .....	373
18.5 DMA 的运行 .....	374
18.5.1 启动源 .....	374
18.5.2 正常模式.....	375
18.5.3 重复模式.....	378
18.5.4 链传送 .....	381

18.6 使用 DMA 时的注意事项.....	383
18.6.1 DMA 控制数据和向量表的设置 .....	383
18.6.2 DMA 控制数据区和 DMA 向量表区的分配 .....	383
18.6.3 DMA 的执行时钟数.....	384
18.6.4 DMA 的响应时间 .....	385
18.6.5 DMA 的启动源.....	385
18.6.6 待机模式中的运行 .....	385
<b>第19章 快速模数转换 (ADC) .....</b>	<b>386</b>
19.1 概述 .....	386
19.2 特性 .....	386
19.3 功能描述 .....	387
19.3.1 ADC 的通道.....	387
19.3.2 ADC 结构框图 .....	388
19.3.3 ADC 的功耗模式.....	388
19.3.4 ADC 的转换模式.....	388
19.3.5 ADC 的时钟.....	389
19.3.6 ADC 的通道选择及中断产生 .....	389
19.3.7 ADC 软件启动 .....	389
19.3.8 ADC 硬件触发启动.....	390
19.4 寄存器映射.....	393
19.5 寄存器说明.....	395
19.5.1 ADC 控制寄存器(CON) .....	395
19.5.2 ADC 控制寄存器 2(CON2) .....	397
19.5.3 ADC 硬件触发控制寄存器(HWTG) .....	398
19.5.4 ADC EPWM 触发延时寄存器(EPWMTGDLY) .....	400
19.5.5 ADC 扫描寄存器(SCAN) .....	400
19.5.6 ADC EPWM 输出触发转换通道使能寄存器(CHEPWM) .....	400
19.5.7 ADC EPWM 周期点触发转换通道使能寄存器(CHPIFTG) .....	400
19.5.8 ADC EPWM 零点触发转换通道使能寄存器(CHZIFTG) .....	400
19.5.9 ADC EPWM 比较器 0 触发转换通道使能寄存器(CHPTG0) .....	401
19.5.10 ADC EPWM 比较器 1 触发转换通道使能寄存器(CHPTG1) .....	401
19.5.11 ADC 测试寄存器(TEST) .....	401
19.5.12 ADC 转换结果寄存器(RESULT).....	401
19.5.13 ADC 通道转换结果寄存器(DATAx) x=0~26 .....	401
19.5.14 ADC 比较控制寄存器 0(CMPx) x=0~1 .....	402
19.5.15 ADC 中断使能寄存器(IMSC).....	402
19.5.16 ADC 中断源状态寄存器(RIS) .....	403
19.5.17 ADC 已使能中断状态寄存器(MIS) .....	403
19.5.18 ADC 中断清零寄存器(ICLR).....	403
19.5.19 ADC 写使能控制寄存器(LOCK) .....	403
<b>第20章 可编程增益放大器 (PGA0/1/2/3) .....</b>	<b>404</b>
20.1 概述 .....	404



20.2 特性 .....	404
20.3 结构框图 .....	405
20.4 寄存器映射 .....	406
20.5 寄存器说明 .....	406
20.5.1 PGA0 控制寄存器 0 PGA0CON0 .....	406
20.5.2 PGA1/2/3 控制寄存器 0 PGANCON0 .....	407
20.5.3 PGA1/2/3 控制寄存器 1 PGANCON1 .....	407
20.5.4 PGA1/2/3 输入通道开关寄存器 1 PGANSW .....	408
20.5.5 PGA 访问寄存器使能 PGALOCK .....	408
<b>第21章 模拟比较器 (ACMP0/1) .....</b>	<b>409</b>
21.1 概述 .....	409
21.2 结构框图 .....	409
21.3 特性 .....	410
21.4 功能说明 .....	411
21.4.1 迟滞功能 .....	411
21.4.2 开窗控制 .....	412
21.4.3 滤波功能 .....	413
21.5 寄存器映射 .....	414
21.6 寄存器说明 .....	415
21.6.1 模拟比较器 0 控制寄存器 0(C0CON0) .....	415
21.6.2 模拟比较器 0 控制寄存器 1(C0CON1) .....	416
21.6.3 模拟比较器 1 控制寄存器 0(C1CON0) .....	417
21.6.4 模拟比较器 1 控制寄存器 1(C1CON1) .....	418
21.6.5 模拟比较器事件控制寄存器(CEVCON) .....	419
21.6.6 模拟比较器中断使能寄存器(IMSC) .....	419
21.6.7 模拟比较器中断源状态寄存器(RIS) .....	420
21.6.8 模拟比较器已使能中断源状态寄存器(MIS) .....	421
21.6.9 模拟比较器中断清零控制寄存器(ICLR) .....	421
21.6.10 模拟比较器开窗控制寄存器 (ACMPOW) .....	422
21.6.11 模拟比较器写使能控制寄存器(LOCK) .....	422
<b>第22章 DAC .....</b>	<b>423</b>
22.1 概述 .....	423
22.2 结构框图 .....	423
22.3 特性 .....	423
22.4 寄存器映射 .....	423
22.5 寄存器说明 .....	424
22.5.1 DAC 控制寄存器 0(CON0) .....	424
22.5.2 DAC 写使能控制寄存器(LOCK) .....	424
<b>第23章 ADCLDO概述 .....</b>	<b>425</b>
23.1 特性 .....	425
23.2 结构框图 .....	425

23.3 寄存器映射.....	425
23.4 寄存器说明.....	426
23.4.1 ADCLDO 控制寄存器 0(CON0).....	426
23.4.2 ADCLDO 写使能控制寄存器(LOCK).....	426
<b>第24章 嵌套向量中断控制器(NVIC).....</b>	<b>427</b>
24.1 特性.....	427
24.2 异常模式和系统中断映射.....	427
24.3 向量表.....	428
24.4 寄存器映射.....	429
24.5 寄存器说明.....	429
24.5.1 中断设置使能控制寄存器 (ISER).....	429
24.5.2 中断清使能控制寄存器 (ICER).....	430
24.5.3 中断设置挂起控制寄存器 (ISPR).....	430
24.5.4 中断清挂起控制寄存器 (ICPR).....	430
24.5.5 IRQ0~IRQ3 中断优先级寄存器 (IPR0).....	431
24.5.6 IRQ4~IRQ7 中断优先级寄存器 (IPR1).....	431
24.5.7 IRQ8~IRQ11 中断优先级寄存器 (IPR2).....	432
24.5.8 IRQ12~IRQ15 中断优先级寄存器 (IPR3).....	432
24.5.9 IRQ16~IRQ19 中断优先级寄存器 (IPR4).....	433
24.5.10 IRQ20~IRQ23 中断优先级寄存器 (IPR5).....	433
24.5.11 IRQ24~IRQ27 中断优先级寄存器 (IPR6).....	434
24.5.12 IRQ28~IRQ31 中断优先级寄存器 (IPR7).....	434
24.5.13 外部中断上升沿允许寄存器 (EGP0).....	435
24.5.14 外部中断下降沿允许寄存器 (EGN0).....	436
<b>第25章 待机功能.....</b>	<b>437</b>
25.1 待机功能.....	437
25.2 睡眠模式.....	438
25.2.1 睡眠模式的设定.....	438
25.2.2 睡眠模式的解除.....	440
25.3 深度睡眠模式.....	441
25.3.1 深度睡眠模式的设定.....	441
25.3.2 深度睡眠模式的解除.....	443
25.4 部分掉电的深度睡眠模式.....	444
25.4.1 部分掉电的深度睡眠模式的设定.....	444
25.4.2 部分掉电的深度睡眠模式的解除.....	446
<b>第26章 复位功能.....</b>	<b>447</b>
26.1 复位时序.....	449
26.2 确认复位源的寄存器.....	451
26.2.1 寄存器映射.....	451
26.2.2 复位控制标志寄存器 (RESF).....	451
<b>第27章 上电复位电路.....</b>	<b>454</b>

27.1 上电复位电路的功能.....	454
27.2 上电复位电路的结构.....	455
27.3 上电复位电路的运行.....	456
<b>第28章 电压检测电路 .....</b>	<b>459</b>
28.1 电压检测电路的功能.....	459
28.2 电压检测电路的结构.....	460
28.3 寄存器映射.....	460
28.4 控制电压检测电路的寄存器 .....	460
28.4.1 电压检测寄存器 (LVIM) .....	461
28.4.2 电压检测电平寄存器 (LVIS) .....	462
28.5 电压检测电路的运行.....	463
28.5.1 用作复位模式时的设定 .....	463
28.6 用作中断模式时的设定 .....	464
28.7 用作中断&复位模式时的设定.....	466
28.8 电压检测电路的注意事项 .....	471
<b>第29章 安全功能.....</b>	<b>473</b>
29.1 安全功能的概要.....	473
29.2 寄存器映射.....	474
29.3 安全功能的运行.....	475
29.3.1 闪存 CRC 运算功能 (高速 CRC) .....	475
29.3.2 SFR 保护功能.....	480
29.3.3 频率检测功能 .....	481
29.3.4 A/D 测试功能.....	482
29.3.5 输入/输出引脚的数字输出信号电平检测功能.....	483
29.3.6 产品唯一身份标识寄存器 .....	484
<b>第30章 温度传感器 .....</b>	<b>485</b>
30.1 温度传感器的功能 .....	485
30.2 寄存器映射.....	485
30.3 温度传感器的寄存器.....	485
30.3.1 温度传感器校准数据寄存器 TSN25 .....	485
30.4 温度传感器的使用说明 .....	486
<b>第31章 选项字节.....</b>	<b>487</b>
31.1 选项字节的功能.....	487
31.1.1 用户选项字节 (000C0H~000C2H) .....	487
31.2 闪存数据保护选项字节 (000C3H, 500004H) .....	488
31.3 寄存器映射.....	488
31.4 用户选项字节 .....	489
31.4.1 用户选项字节 (000C0H) .....	489
31.4.2 用户选项字节 (000C1H) .....	490
31.4.3 用户选项字节 (000C2H) .....	494

31.4.4 闪存数据保护选项字节 (000C3H) .....	495
31.4.5 闪存数据保护选项字节 (500004H) .....	495
<b>第32章 FLASH控制.....</b>	<b>496</b>
32.1 FLASH 控制功能描述.....	496
32.2 FLASH 存储器结构 .....	496
32.3 寄存器映射.....	497
32.4 寄存器说明.....	497
32.4.1 Flash 写保护寄存器(FLPROT).....	497
32.4.2 FLASH 操作控制寄存器(FLOPMD1).....	497
32.4.3 FLASH 操作控制寄存器(FLOPMD2).....	498
32.4.4 Flash 擦除控制寄存器(FLERMD).....	498
32.4.5 Flash 状态寄存器(FLSTS).....	498
32.4.6 Flash 全片擦除时间控制寄存器(FLCERCNT).....	499
32.4.7 Flash 页擦除时间控制寄存器 (FLSERCNT) .....	499
32.4.8 Flash 写入时间控制寄存器 (FLPROCNT) .....	499
32.5 FLASH 操作方法 .....	500
32.5.1 页擦除 (sector erase) .....	500
32.5.2 全片擦除 (chip erase) .....	501
32.5.3 编程 (word program) .....	501
32.6 FLASH 操作的注意事项 .....	501
<b>第33章 测试相关.....</b>	<b>502</b>
33.1 寄存器映射.....	502
33.2 寄存器说明.....	502
33.2.1 测试寄存器 1(TEST1).....	502
33.2.2 测试寄存器访问使能控制寄存器 .....	502
<b>第34章 附录 修订记录 .....</b>	<b>503</b>

# 第1章 CPU

## 1.1 概述

本章节简单介绍本制品搭载的ARM Cortex-M0+内核的特性及调试特性，详情请参考ARM相关文档。

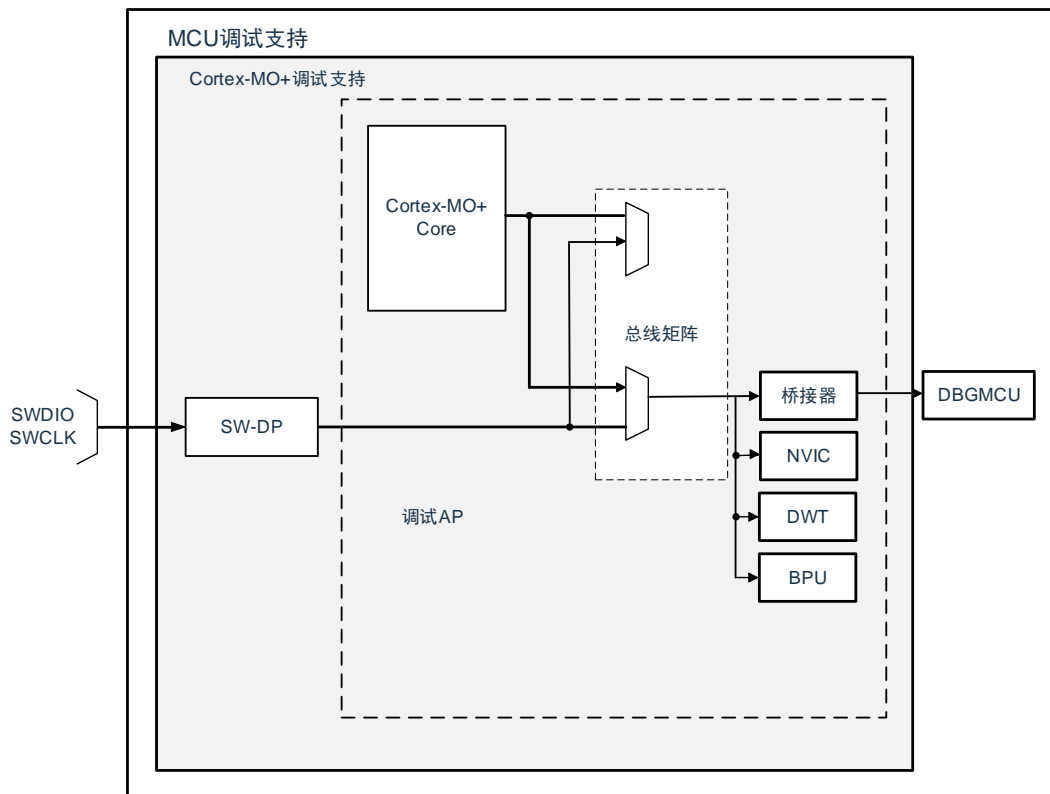
## 1.2 Cortex-M0+内核特性

- ARM Cortex-M0+处理器是32位RISC内核，采用2级流水线，仅支持特权模式
- 1周期硬件乘法器
- 嵌套向量中断控制器（NVIC）
  - 1个不可屏蔽中断(NMI)
  - 支持23个可屏蔽中断请求(IRQ)
  - 4个中断优先级
- 系统定时器SysTick是一个24位倒计时定时器，可选择 $f_{CLK}$ 或 $f_{IL}$ 计数时钟
- 向量表偏移寄存器(VTOR)
  - 软件可以写VTOR将向量表起始地址重新定位到不同的位置
  - 该寄存器的默认值为0x0000\_0000,低8位写忽略，读为零，也就是偏移量256字节对齐

## 1.3 调试特性

- 2线SWD调试接口
- 支持暂停、恢复和单步执行程序
- 访问处理器的内核寄存器和特殊功能寄存器
- 4个硬件断点(BPU)
- 无限个软件断点(BKPT指令)
- 2个数据观察点(DWT)
- 内核执行的时候访问存储器

图1-1 Cortex-M0+的调试框图



注：SWD在Deep Sleep模式下不能工作，请在active和sleep模式下进行调试操作。

## 1.4 SWD接口引脚

本产品的2个GPIO可用作SWD接口引脚，这些引脚在所有的封装里都存在。

表1-1 SWD调试端口引脚

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	P03
SWDIO	串行数据输入 /输出	P02

不使用SWD功能时，可以通过设置debug停止控制寄存器（DBGSTOPCR）来禁用SWD。

位	符号	描述	复位值
31:25	-	保留	-
24	SWDIS	SWD调试接口禁用 0: SWD调试接口使能。在连接调试器的状态下，P02/P03不能用作GPIO（因为此时该IOBUF的ENO和DOUT由调试器控制） 1: SWD调试接口禁用。P02/P03可用作GPIO	0
23:2	-	保留	0x0
1	FRZEN1	在调试器连接的状态下，并且CPU处于调试状态时（HALTED=1），通信系周边模块动作/停止 <sup>注2</sup> 0: 周边动作 1: 周边停止	0
0	FRZEN0	在调试器连接的状态下，并且CPU处于调试状态时（HALTED=1），定时器系周边模块动作/停止 <sup>注1</sup> 0: 周边动作 1: 周边停止	0

注1：本制品的定时器系周边模块包括：通用定时器单元Timer4。

注2：本制品的通信系周边模块包括：串行IICA。

## 1.5 ARM参考文档

Cortex®-M0+ 内核中内置的调试功能是ARM® CoreSight 设计套件的一部分。相关文档请参考：

- Cortex®-M0+技术参考手册(TRM)
- ARM®调试接口V5
- ARM® CoreSight设计套件版本r1p1 技术参考手册



## 第2章 引脚功能

### 2.1 端口功能

具体端口功能参见各产品系列的数据手册。

### 2.2 端口复用功能

具体端口复用功能参见各产品系列的数据手册。详细端口复用功能参见下表2-1。

表2-1 端口复用功能数字映射表

功能名称	输入	复用功能 PmnCFG			
		0	1	2	3
P00	HALL_IN0/RXD0/CCP0A_I	--	TXD0	CCP0A_O	ADC_TRIG
P01	NRST/RXD0/CCP0B_I		TXD0	CCP0B_O	ADC_TRIG
P02	CCP1A_I		-	CCP1A_O	-
P03	CCP1B_I		-	CCP1B_O	ADC_TRIG
P04	CCP1A_I		C0_O	CCP1A_O	SCL
P05	CCP1B_I		-	CCP1B_O	SDA
P06	CCP0A_I		NSS	CCP0A_O	SCL
P07	CCP0B_I		SCLK	CCP0B_O	SDA
P10	ADET		MOSI	-	-
P11	CCP1A_I		MISO	CCP1A_O	C0_O
P12	CCP1B_I		-	CCP1B_O	-
P13	BKIN		-	-	-
P14	-		MISO	PCUBZ0	SCL
P15	RXD0		MOSI	TXD0	SDA
P16	BKIN/RXD0		SCLK	TXD0	-
P17	-		NSS	-	ADC_TRIG
P20	HALL_IN0		-	-	-
P21	HALL_IN1		-	-	ADC_TRIG
P22	HALL_IN2		-	-	ADC_TRIG
P23	BKIN/RXD1/CCP0B_I		TXD1	CCP0B_O	C1_O
P24	RXD1/CCP0A_I		TXD1	CCP0A_O	ADC_TRIG
P25	RXD1/CCP1A_I		TXD1	CCP1A_O	C1_O
P30	-		-	-	-
P31	-		-	-	-
P32	-		-	-	-
P33	-	-	-	-	
P34	RXD0	TXD0	-	SDA	
P35	RXD0	TXD0	-	SCL	
P36	-	EPWM0	-	-	
P37	-	EPWM1	-	-	
P40	-	EPWM2	-	-	
P41	-	EPWM3	-	-	

P42	-		EPWM4	-	-
P43	-		EPWM5	-	-
P44	-		EPWM6	-	-
P45	-		EPWM7	-	-
P46	-		-	-	-
P47	-		-	-	-
P50	-		-	-	-
P51	-		-	-	-
P52	CCP0B_I		-	CCP0B_O	-
P53	RXD1/ CCP1B_I		TXD1	CCP1B_O	-
P54	RXD1		TXD1	-	SCL
P55	BKIN/RXD1		TXD1	-	SDA
P56	HALL_IN2/CCP1A_I		-	CCP1A_O	
P57	HALL_IN1/ CCP1B_I		-	CCP1B_O	

注1：本产品IO复用功能，需要用户单独配置PMC，PM等寄存器。

注2：当选择到IIC功能时，开漏功能自动开启。

注3：针对复用功能，即可做输入又可做输出功能，选择好PmnCFG后，输入通道自动开启。

表2-2: 模拟功能和特殊功能管脚

管脚	模拟			特殊功能管脚
	ADC	ACMP	PGA	
P00	AN7	C0P0	-	-
P01	-	-	-	NRST
P02	-	-	-	SWDDAT
P03	-	-	-	SWDCLK
P04	AN8	-	-	OSCOUT
P05	AN9	-	-	OSCIN
P06	AN10	-	-	-
P07	AN11	-	-	-
P10	-	-	--	-
P11	AN12	-	-	-
P12	AN13	-	-	DAC_O
P13	AN14	-	-	-
P14	AN15	-	-	-
P15	-	-	-	-
P16	-	-	-	-
P17	-	-	-	-
P20	AN16	C1P0	-	-
P21	AN17	C1P1	-	-
P22	AN18	C1P2	-	-
P23	-	C1P3	-	-
P24	AN19	CIN	-	-
P25	-	-	-	-
P30	-	-	PGA3_P	-
P31	-	-	PGA3_N	-
P32	-	-	PGA2_P	-
P33	-	-	PGA2_N	-
P34	AN20	-	-	-
P35	-	-	-	-
P36	-	-	-	-
P37	-	-	-	-
P40	-	-	-	-
P41	-	-	-	-
P42	-	-	-	-
P43	-	-	-	-
P44	-	-	-	-
P45	-	-	-	-
P46	-	-	PGA1_P	-
P47	-	-	PGA1_N	-
P50	-	-	PGA0_P	-
P51	-	-	PGA0_N	-
P52	AN21	-	PGA0_O	-
P53	AN22	-	PGA123_O	DAC_O
P54	-	C0P3	-	-
P55	-	C0N	-	-
P56	AN5	C0P2	-	-
P57	AN6	C0P1	-	-

## 2.3 寄存器映射

### 2.3.1 控制功能寄存器映射

(端口控制寄存器的基地址=0x40040000)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
P0	0x000	R/W	以位为单位, 设定端口输出锁存器值的寄存器; 输入模式中读此寄存器可以得到引脚电平, 在输出模式中读此寄存器可得到端口的输出锁存器的值。	0x00
P1	0x001	R/W		0x00
P2	0x002	R/W		0x00
P3	0x003	R/W		0x00
P4	0x004	R/W		0x00
P5	0x005	R/W		0x00
PM0	0x020	R/W	在端口作为数字通道使用时, 以位为单位, 设定端口输入或输出的寄存器。	0xFF
PM1	0x021	R/W		0xFF
PM2	0x022	R/W		0xFF
PM3	0x023	R/W		0xFF
PM4	0x024	R/W		0xFF
PM5	0x025	R/W		0xFF
PU0	0x030	R/W	端口内部上拉电阻选择寄存器, 只能在对应PMCmn=0时, 设置上拉电阻有效。P01、P02、P03的上拉功能默认打开。	0x0E
PU1	0x031	R/W		0x00
PU2	0x032	R/W		0x00
PU3	0x033	R/W		0x00
PU4	0x034	R/W		0x00
PU5	0x035	R/W		0x00
PD0	0x040	R/W	端口内部下拉电阻选择寄存器, 只能在对应PMCmn=0时, 设置下拉电阻有效。P01无下拉功能	0x00
PD1	0x041	R/W		0x00
PD2	0x042	R/W		0x00
PD3	0x043	R/W		0x00
PD4	0x044	R/W		0x00
PD5	0x045	R/W		0x00
POM0	0x050	R/W	开漏模式寄存器, 只有在端口配置为输出模式时, N沟道漏极开路才会开启。	0x00
POM1	0x051	R/W		0x00
POM2	0x052	R/W		0x00
POM3	0x053	R/W		0x00
POM4	0x054	R/W		0x00
POM5	0x055	R/W		0x00
PMC0	0x060	R/W	端口模式寄存器, 以位为单位设定端口作为数字或模拟通道使用; P01、P02、P03、P04、P05默认作为数字通道使用。	0xC1
PMC1	0x061	R/W		0xFF
PMC2	0x062	R/W		0xFF
PMC3	0x063	R/W		0xFF
PMC4	0x064	R/W		0xFF
PMC5	0x065	R/W		0xFF
PSET0	0x070	W	以位为单位来置位端口输出锁存器的寄存器。	0x00
PSET1	0x071	W		0x00
PSET2	0x072	W		0x00
PSET3	0x073	W		0x00
PSET4	0x074	W		0x00
PSET5	0x075	W		0x00

PCLR0	0x080	W	以位为单位来清零端口输出锁存器的寄存器。	0x00
PCLR1	0x081	W		0x00
PCLR2	0x082	W		0x00
PCLR3	0x083	W		0x00
PCLR4	0x084	W		0x00
PCLR5	0x085	W		0x00

## 2.3.2 输出输入复用功能寄存器映射

(输出输入复用功能寄存器的基地址=0x40040800)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
P00CFG	0x00	R/W	端口输出复用功能配置寄存器, 可实现将一部分外围模块的输出功能映射到对应的端口上, 具体管脚功能数字映射, 参见表2-1; 端口输出复用功能寄存器的复位值为0x00, 此时端口为默认的兼用功能和GPIO功能。	0x00
P01CFG	0x02	R/W		0x00
P02CFG	0x04	R/W		0x00
P03CFG	0x06	R/W		0x00
P04CFG	0x08	R/W		0x00
P05CFG	0x0A	R/W		0x00
P06CFG	0x0C	R/W		0x00
P07CFG	0x0E	R/W		0x00
P10CFG	0x10	R/W		0x00
P11CFG	0x12	R/W		0x00
P12CFG	0x14	R/W		0x00
P13CFG	0x16	R/W		0x00
P14CFG	0x18	R/W		0x00
P15CFG	0x1A	R/W		0x00
P16CFG	0x1C	R/W		0x00
P17CFG	0x1E	R/W		0x00
P20CFG	0x20	R/W		0x00
P21CFG	0x22	R/W		0x00
P22CFG	0x24	R/W		0x00
P23CFG	0x26	R/W		0x00
P24CFG	0x28	R/W		0x00
P25CFG	0x2A	R/W		0x00
P30CFG	0x30	R/W		0x00
P31CFG	0x32	R/W		0x00
P32CFG	0x34	R/W		0x00
P33CFG	0x36	R/W		0x00
P34CFG	0x38	R/W		0x00
P35CFG	0x3A	R/W		0x00
P36CFG	0x3C	R/W		0x00
P37CFG	0x3E	R/W		0x00
P40CFG	0x40	R/W		0x00
P41CFG	0x42	R/W		0x00
P42CFG	0x44	R/W	0x00	
P43CFG	0x46	R/W	0x00	
P44CFG	0x48	R/W	0x00	
P45CFG	0x4A	R/W	0x00	
P46CFG	0x4C	R/W	0x00	
P47CFG	0x4E	R/W	0x00	
P50CFG	0x50	R/W	0x00	
P51CFG	0x52	R/W	0x00	
P52CFG	0x54	R/W	0x00	
P53CFG	0x56	R/W	0x00	
P54CFG	0x58	R/W	0x00	

P55CFG	0x5A	R/W		0x00
P56CFG	0x5C	R/W		0x00
P57CFG	0x5E	R/W		0x00
PS <sub>int0</sub> _CFG	0x60	R/W	外部中断0输入端口选择寄存器	0x7f
PS <sub>int1</sub> _CFG	0x61	R/W	外部中断1输入端口选择寄存器	0x7f
PS <sub>int2</sub> _CFG	0x62	R/W	外部中断2输入端口选择寄存器	0x7f
PS <sub>int3</sub> _CFG	0x63	R/W	外部中断3输入端口选择寄存器	0x7f
PS <sub>tau0tin0</sub> _CFG	0x64	R/W	TAU0外部输入通道0输入端口选择寄存器	0x7f
PS <sub>tau0tin1</sub> _CFG	0x65	R/W	TAU0外部输入通道1输入端口选择寄存器	0x7f
PS <sub>tau0tin2</sub> _CFG	0x66	R/W	TAU0外部输入通道2输入端口选择寄存器	0x7f
PS <sub>tau0tin3</sub> _CFG	0x67	R/W	TAU0外部输入通道3输入端口选择寄存器	0x7f
PS <sub>uart0rx</sub> _CFG	0x68	R/W	UART0的RXD0信号输入端口选择寄存器	0x07
PS <sub>epwmnkin</sub> _CFG	0x69	R/W	EPWM外部刹车输入端口选择寄存器	0x07
PS <sub>ccp0ain</sub> _CFG	0x6A	R/W	CCP0A通道捕获输入端口选择寄存器	0x07
PS <sub>ccp0bin</sub> _CFG	0x6B	R/W	CCP0B通道捕获输入端口选择寄存器	0x07
PS <sub>ccp1ain</sub> _CFG	0x6C	R/W	CCP1A通道捕获输入端口选择寄存器	0x07
PS <sub>ccp1bin</sub> _CFG	0x6D	R/W	CCP1B通道捕获输入端口选择寄存器	0x07
PS <sub>uart1rx</sub> _CFG	0x6E	R/W	UART1的RXD1信号输入端口选择寄存器	0x07
PS <sub>hall_in0</sub> _CFG	0x6F	R/W	HALL传感器输入IN0通道端口选择寄存器	0x07
PS <sub>hall_in1</sub> _CFG	0x70	R/W	HALL传感器输入IN1通道端口选择寄存器	0x07
PS <sub>hall_in2</sub> _CFG	0x71	R/W	HALL传感器输入IN2通道端口选择寄存器	0x07
P0TTLCFG	0x74	R/W	P0端口输入电平选择寄存器。	0x00
PMS	0x7B	R/W	端口读取模式选择寄存器；在端口为输出模式时，该寄存器选择读端口锁存器的值或引脚输出电平。详见第29.3.5章	0x00

### 2.3.3 特殊功能端口RESINB控制寄存器映射

(寄存器的基地址=0x40020400)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
RSTM	0x0B	R/W	选择RESINB(P01)口作为外部复位口或GPIO口的寄存器	0x00

## 2.4 寄存器说明

通过以下寄存器控制端口：

- (1) 端口寄存器 (Px)
- (2) 端口模式寄存器 (PMx)
- (3) 上拉电阻选择寄存器 (PUx)
- (4) 下拉电阻选择寄存器 (PDx)
- (5) 端口输出模式寄存器 (POMx)
- (6) 端口模式控制寄存器 (PMCx)
- (7) 端口置位控制寄存器 (PSETx)
- (8) 端口清零控制寄存器 (PCLRx)
- (9) 端口输出复用功能配置寄存器 (PxxCFG)
- (10) 端口输入复用功能配置寄存器 (PSxx\_CFG)
- (11) 端口电平选择寄存器 (PxTTLCFG)
- (12) 特殊功能端口RESINB控制寄存器(RSTM)



## 2.4.1 端口寄存器 (Px)

这是以位为单位设定端口输出锁存器值的寄存器 Px (x=0~5)。在输入模式中读此寄存器可以得到引脚电平，而在输出模式中读可以得到端口的输出锁存器的值。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
7	Px7	端口x的模式寄存器第7位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
6	Px6	端口x的模式寄存器第6位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
5	Px5	端口x的模式寄存器第5位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
4	Px4	端口x的模式寄存器第4位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
3	Px3	端口x的模式寄存器第3位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
2	Px2	端口x的模式寄存器第2位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
1	Px1	端口x的模式寄存器第1位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
0	Px0	端口x的模式寄存器第0位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0

注：必须给未分配的位设定初始值。

## 2.4.2 端口模式寄存器 (PMx)

当端口作为数字通道使用时，这是以位为单位设定其输入/输出的寄存器PMx (x=0~5)。在产生复位信号后，所有端口默认为输入状态。具体寄存器描述如下：

位	符号	描述	复位值
7	PMx7	端口x的模式寄存器第7位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
6	PMx6	端口x的模式寄存器第6位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
5	PMx5	端口x的模式寄存器第5位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
4	PMx4	端口x的模式寄存器第4位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
3	PMx3	端口x的模式寄存器第3位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
2	PMx2	端口x的模式寄存器第2位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
1	PMx1	端口x的模式寄存器第1位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1
0	PMx0	端口x的模式寄存器第0位 0: 输出模式(用作输出端口 (输出缓冲器ON) ) 1: 输入模式(用作输入端口 (输出缓冲器OFF) )	1

注：P26, P27端口无效，PM2的bit7~6保持为1。

### 2.4.3 上拉电阻选择寄存器 (PUx)

内部上拉电阻的选择寄存器 PUx (x=0~5)。只能在对应的 PMCx 对应位等于 0 的时候, 设置上拉电阻有效。

在产生复位信号后, P01, P02, P03 端口上拉功能自动打开, 其他端口的上拉功能默认不打开。具体寄存器描述如下:

位	符号	描述	复位值
7	PUx7	Px引脚的内部上拉电阻选择第7位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
6	PUx6	Px引脚的内部上拉电阻选择第6位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
5	PUx5	Px引脚的内部上拉电阻选择第5位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
4	PUx4	Px引脚的内部上拉电阻选择第4位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
3	PUx3	Px引脚的内部上拉电阻选择第3位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	x=0,复位值为1 x=其他值,复位值为0
2	PUx2	Px引脚的内部上拉电阻选择第2位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	x=0,复位值为1 x=其他值,复位值为0
1	PUx1	Px引脚的内部上拉电阻选择第1位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	x=0,复位值为1 x=其他值,复位值为0
0	PUx0	Px引脚的内部上拉电阻选择第0位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0

注1: P26, P27端口无效, PU2的bit7~6保持为0。

注2: 内部上拉电阻为10K。

## 2.4.4 下拉电阻选择寄存器 (PDx)

内部下拉电阻选择寄存器PDx (x=0~5)。只能在对应的PMCx对应位等于0 的时候, 设置下拉电阻有效; RESINB(P01)端口无下拉功能。

在产生复位信号后, 所有端口的下拉功能默认不打开。具体寄存器描述如下:

位	符号	描述	复位值
7	PDx7	Px引脚的内部下拉电阻选择第7位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
6	PDx6	Px引脚的内部下拉电阻选择第6位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
5	PDx5	Px引脚的内部下拉电阻选择第5位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
4	PDx4	Px引脚的内部下拉电阻选择第4位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
3	PDx3	Px引脚的内部下拉电阻选择第3位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
2	PDx2	Px引脚的内部下拉电阻选择第2位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
1	PDx1	Px引脚的内部下拉电阻选择第1位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻 注: RESINB(P01)端口无下拉功能	0
0	PDx0	Px引脚的内部下拉电阻选择第0位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0

注: P26, P27端口无效, PD2的bit7~6保持为0。

## 2.4.5 端口输出模式寄存器 (POMx)

端口输出模式寄存器 POMx (x=0~5)，只有在配置成输出模式N沟道漏极开路才会开启。P04,P05,P06,P07,P14,P15,P34,P35,P54,P55端口复用为IIC功能时，都会强制开启开漏模式。

在产生复位信号后，这些寄存器的值变为“00H”。

注意：对于设定N沟道漏极开路输出模式 (POMmn=1) 的位，不连接内部上拉电阻。

具体寄存器描述如下：

位	符号	描述	复位值
7	POMx7	Px引脚输出模式选择第7位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
6	POMx6	Px引脚输出模式选择第6位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
5	POMx5	Px引脚输出模式选择第5位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
4	POMx4	Px引脚输出模式选择第4位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
3	POMx3	Px引脚输出模式选择第3位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
2	POMx2	Px引脚输出模式选择第2位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
1	POMx1	Px引脚输出模式选择第1位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
0	POMx0	Px引脚输出模式选择第0位 0: 通常输出模式 1: N沟道漏极开路输出模式	0

注：P26, P27端口无效，POM2的bit7~6保持为0。

## 2.4.6 端口模式控制寄存器 (PMCx)

端口模式寄存器 PMCx(x=0~5), 以位为单位设定端口作为数字 (输入/输出) 或者作为模拟 (输入) 通道。

在产生复位信号后, P01, P02, P03, P04, P05 默认作为数字通道使用 (PMC01, PMC02, PMC03, PMC04, PMC05 复位值为“0”), 其他端口默认作为模拟通道使用即 PMCx 对应位等于 1。具体寄存器描述如下:

位	符号	描述	复位值
7	PMCx7	Px引脚数字 (输入/输出) 或模拟 (输入) 第7位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	1
6	PMCx6	Px引脚数字 (输入/输出) 或模拟 (输入) 第6位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	1
5	PMCx5	Px引脚数字 (输入/输出) 或模拟 (输入) 第5位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	x=0时: 0 x=其他值: 1
4	PMCx4	Px引脚数字 (输入/输出) 或模拟 (输入) 第4位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	x=0时: 0 x=其他值: 1
3	PMCx3	Px引脚数字 (输入/输出) 或模拟 (输入) 第3位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	x=0时: 0 x=其他值: 1
2	PMCx2	Px引脚数字 (输入/输出) 或模拟 (输入) 第2位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	x=0时: 0 x=其他值: 1
1	PMCx1	Px引脚数字 (输入/输出) 或模拟 (输入) 第1位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	x=0时: 0 x=其他值: 1
0	PMCx0	Px引脚数字 (输入/输出) 或模拟 (输入) 第0位 0: 数字输入/输出 (模拟输入以外的复用功能) 1: 模拟输入	1

注: P26, P27端口无效, PMC2的bit7~6保持为1。

## 2.4.7 端口置位控制寄存器 (PSETx)

这是以位为单位来置位端口输出锁存器的寄存器PSETx (x=0~5)。在产生复位信号后, 这些寄存器的值变为“00H”。具体寄存器描述如下:

位	符号	描述	复位值
7	PSETx7	Px引脚的置位控制第7位 0: 无操作 1: 对应的Px7置1	0
6	PSETx6	Px引脚的置位控制第6位 0: 无操作 1: 对应的Px6置1	0
5	PSETx5	Px引脚的置位控制第5位 0: 无操作 1: 对应的Px5置1	0
4	PSETx4	Px引脚的置位控制第4位 0: 无操作 1: 对应的Px4置1	0
3	PSETx3	Px引脚的置位控制第3位 0: 无操作 1: 对应的Px3置1	0
2	PSETx2	Px引脚的置位控制第2位 0: 无操作 1: 对应的Px2置1	0
1	PSETx1	Px引脚的置位控制第1位 0: 无操作 1: 对应的Px1置1	0
0	PSETx0	Px引脚的置位控制第0位 0: 无操作 1: 对应的Px0置1	0

注: P26, P27端口无效, PSET2的bit7~6保持为0。

## 2.4.8 端口清零控制寄存器 (PCLR<sub>x</sub>)

这是以位为单位来清零端口输出锁存器的寄存器 PCLR<sub>x</sub> (x=0~5)。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
7	PCLR <sub>x</sub> 7	Px引脚的清零控制第7位 0: 无操作 1: 对应的Px7清零	0
6	PCLR <sub>x</sub> 6	Px引脚的清零控制第6位 0: 无操作 1: 对应的Px6清零	0
5	PCLR <sub>x</sub> 5	Px引脚的清零控制第5位 0: 无操作 1: 对应的Px5清零	0
4	PCLR <sub>x</sub> 4	Px引脚的清零控制第4位 0: 无操作 1: 对应的Px4清零	0
3	PCLR <sub>x</sub> 3	Px引脚的清零控制第3位 0: 无操作 1: 对应的Px3清零	0
2	PCLR <sub>x</sub> 2	Px引脚的清零控制第2位 0: 无操作 1: 对应的Px2清零	0
1	PCLR <sub>x</sub> 1	Px引脚的清零控制第1位 0: 无操作 1: 对应的Px1清零	0
0	PCLR <sub>x</sub> 0	Px引脚的清零控制第0位 0: 无操作 1: 对应的Px0清零	0

注1: P26, P27端口无效, PCLR2的bit7~6保持为0。



## 2.4.9 端口输出复用功能配置寄存器 (P<sub>mn</sub>CFG)

端口输出复用功能配置寄存器P<sub>mn</sub>CFG(m=0、1、3、4、5,n=0~7.m=2时,n=0~5), 可实现将一部分外围模块的输出功能映射到端口, 具体映射功能参见表2-1; 端口输出复用功能配置寄存器的复位值为“00H”, 此时端口为默认的兼用功能和GPIO功能。设置步骤如下:

- 1) 设置 PMCmn=0, 选择数字输入输出
- 2) 设置PMmn=0, 设置输出模式
- 3) 设置Pmn\_CFG选择对应的管脚的复位功能输出

注: IIC的SDA, SCL以及SPI所有功能脚既可以做输出也可做输入, 在设置PmnCFG后, 输入功能将自动开启, 除以上3个步骤外, 无需其他操作。

具体寄存器描述如下:

位	符号	描述	复位值
7:3	--	保留	--
2:0	PmnCFG[2:0]	Pmn引脚输出复用功能 0x00: Pmn对应输出映射表=0x00的功能 0x01: Pmn对应输出映射表=0x01的功能 0x02: Pmn对应输出映射表=0x02的功能 0x03: Pmn对应输出映射表=0x03的功能	0x0

## 2.4.10 端口输入复用功能配置寄存器 (PSxx\_CFG)

INTP0、INTP1、INTP2、INTP3、TI00、TI01、TI02、TI03 可映射到任意的 GPIO 输入；RXD0、RXD1、BKIN、CCP0AIN、CCP0BIN、CCP1AIN、CCP1BIN、HALL\_IN0、HALL\_IN1、HALL\_IN2 可映射到指定 GPIO 输入，具体输入功能映射参见表 2-1；由于每个功能都有多管脚输入，因此在用输入复用功能时需要选择具体的 GPIO 输入，设置步骤如下：

- 4) 设置 PMCmn = 0
- 5) 设置 PMmn=1
- 6) 设置 PSxx\_CFG 选择对应的管脚输入

PSintp0\_CFG 描述如下：

位	符号	描述	复位值
7	--	保留	--
6:0	PSintp0_CFG[6:0]	INTP0 选择 GPIO 输入 0x00: 选择 P00 作为 INTP0 输入 0x01: 选择 P01 作为 INTP0 输入 ... .. 0x56: 选择 P56 作为 INTP0 输入 0x57: 选择 P57 作为 INTP0 输入 其他: 输入低电平	0x7F

PSintp1\_CFG 描述如下：

位	符号	描述	复位值
7	--	保留	--
6:0	PSintp1_CFG[6:0]	INTP1 选择 GPIO 输入 0x00: 选择 P00 作为 INTP1 输入 0x01: 选择 P01 作为 INTP1 输入 ... .. 0x56: 选择 P56 作为 INTP1 输入 0x57: 选择 P57 作为 INTP1 输入 其他: 输入低电平	0x7F

PSintp2\_CFG 描述如下：

位	符号	描述	复位值
7	--	保留	--
6:0	PSintp2_CFG[6:0]	INTP2 选择 GPIO 输入 0x00: 选择 P00 作为 INTP2 输入 0x01: 选择 P01 作为 INTP2 输入 ... .. 0x56: 选择 P56 作为 INTP2 输入 0x57: 选择 P57 作为 INTP2 输入 其他: 输入低电平	0x7F

PSintp3\_CFG描述如下:

位	符号	描述	复位值
7	--	保留	--
6:0	PSintp3_CFG[6:0]	INTP3选择GPIO输入 0x00: 选择P00作为INTP3输入 0x01: 选择P01作为INTP3输入 ... .. 0x56: 选择P56作为INTP3输入 0x57: 选择P57作为INTP3输入 其他: 输入低电平	0x7F

PStau0tin0\_CFG描述如下:

位	符号	描述	复位值
7	--	保留	--
6:0	PStau0tin0_CFG[6:0]	TI00选择GPIO输入 0x00: 选择P00作为TI00输入 0x01: 选择P01作为TI00输入 ... .. 0x56: 选择P56作为TI00输入 0x57: 选择P57作为TI00输入 其他: TI00输入低电平	0x7F

PStau0tin1\_CFG描述如下:

位	符号	描述	复位值
7	--	保留	--
6:0	PStau0tin1_CFG[6:0]	TI01选择GPIO输入 0x00: 选择P00作为TI01输入 0x01: 选择P01作为TI01输入 ... .. 0x56: 选择P56作为TI01输入 0x57: 选择P57作为TI01输入 其他: TI01输入低电平	0x7F

PStau0tin2\_CFG描述如下:

位	符号	描述	复位值
7	--	保留	--
6:0	PStau0tin2_CFG[6:0]	TI02选择GPIO输入 0x00: 选择P00作为TI02输入 0x01: 选择P01作为TI02输入 ... .. 0x56: 选择P56作为TI02输入 0x57: 选择P57作为TI02输入 其他: TI02输入低电平	0x7F

PStau0tin3\_CFG描述如下:

位	符号	描述	复位值
7	--	保留	--
6:0	PStau0tin3_CFG [6:0]	TI03选择GPIO输入 0x00: 选择P00作为TI03输入 0x01: 选择P01作为TI03输入 ... .. 0x56: 选择P56作为TI03输入 0x57: 选择P57作为TI03输入 其他: TI03输入低电平	0x7F

PSuart0rx\_d\_CFG描述如下:

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSuart0rx_d_CFG[2:0]	UART0_RXD选择GPIO输入 0x00: 选择P00作为UART0_RXD输入 0x01: 选择P01作为UART0_RXD输入 0x02: 选择P15作为UART0_RXD输入 0x03: 选择P16作为UART0_RXD输入 0x04: 选择P34作为UART0_RXD输入 0x05: 选择P35作为UART0_RXD输入 其他: UART0_RXD输入高电平	0x7

PSuart1rx\_d\_CFG描述如下:

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSuart1rx_d_CFG[2:0]	UART1_RXD选择GPIO输入 0x00: 选择P23作为UART1_RXD输入 0x01: 选择P24作为UART1_RXD输入 0x02: 选择P25作为UART1_RXD输入 0x03: 选择P53作为UART1_RXD输入 0x04: 选择P54作为UART1_RXD输入 0x05: 选择P55作为UART1_RXD输入 其他: UART1_RXD输入高电平	0x7

PSepwmbkin\_CFG描述如下:

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSepwmbkin_CFG [2:0]	EPWM_BKIN选择GPIO输入 0x00: 选择P13作为EPWM_BKIN输入 0x01: 选择P16作为EPWM_BKIN输入 0x02: 选择P23作为EPWM_BKIN输入 0x03: 选择P55作为EPWM_BKIN输入 其他: EPWM_BKIN输入低电平	0x7

PSccp0ain\_CFG描述如下：

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSccp0a_i_CFG [2:0]	CCP0AIN选择GPIO输入 0x00: 选择P00作为CCP0A_I输入 0x01: 选择P06作为CCP0A_I输入 0x02: 选择P24作为CCP0A_I输入 其他: CCP0A_I输入低电平	0x7

PSccp0bin\_CFG描述如下：

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSccp0b_i_CFG [2:0]	CCP0BIN选择GPIO输入 0x00: 选择P01作为CCP0B_I输入 0x01: 选择P07作为CCP0B_I输入 0x02: 选择P23作为CCP0B_I输入 0x03: 选择P52作为CCP0B_I输入 其他: CCP0B_I输入低电平	0x7

PSccp1ain\_CFG描述如下：

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSccp1ain_CFG [2:0]	CCP1AIN选择GPIO输入 0x00: 选择P02作为CCP1A_I输入 0x01: 选择P04作为CCP1A_I输入 0x02: 选择P11作为CCP1A_I输入 0x03: 选择P25作为CCP1A_I输入 0x04: 选择P56作为CCP1A_I输入 其他: CCP1A_I输入低电平	0x7

PSccp1bin\_CFG描述如下：

位	符号	描述	复位值
7:3	--	保留	--
2:0	PSccp1bin_CFG [2:0]	CCP1BIN选择GPIO输入 0x00: 选择P03作为CCP1B_I输入 0x01: 选择P05作为CCP1B_I输入 0x02: 选择P12作为CCP1B_I输入 0x03: 选择P53作为CCP1B_I输入 0x04: 选择P57作为CCP1B_I输入 其他: CCP1B_I输入低电平	0x7

PShall\_in0\_CFG描述如下：

位	符号	描述	复位值
7:3	--	保留	--
2:0	PShall_in0_CFG [2:0]	HALL_IN0选择GPIO输入 0x00: 选择P00作为HALL_IN0输入 0x01: 选择P20作为HALL_IN0输入 其他: HALL_IN0输入低电平	0x7

PShall\_in1\_CFG描述如下:

位	符号	描述	复位值
7:3	--	保留	--
2:0	PShall_in1_CFG [2:0]	HALL_IN1选择GPIO输入 0x00: 选择P21作为HALL_IN1输入 0x01: 选择P57作为HALL_IN1输入 其他: HALL_IN1输入低电平	0x7

PShall\_in2\_CFG描述如下:

位	符号	描述	复位值
7:3	--	保留	--
2:0	PShall_in2_CFG [2:0]	HALL_IN2选择GPIO输入 0x00: 选择P22作为HALL_IN2输入 0x01: 选择P56作为HALL_IN2输入 其他: HALL_IN2输入低电平	0x7

### 2.4.11 TTL和施密特输入选择(P0TTLCFG)

P0TTLCFG选择寄存器描述如下：

位	符号	描述	复位值
7	P0TTL7	P07 输入电平选择 0: 施密特输入 1: TTL输入	0
6	P0TTL6	P06 输入电平选择 0: 施密特输入 1: TTL输入	0
5	P0TTL5	P05 输入电平选择 0: 施密特输入 1: TTL输入	0
4	P0TTL4	P04 输入电平选择 0: 施密特输入 1: TTL输入	0
3	P0TTL3	P03 输入电平选择 0: 施密特输入 1: TTL输入	0
2	P0TTL2	P02 输入电平选择 0: 施密特输入 1: TTL输入	0
1	P0TTL1	P01 输入电平选择 0: 施密特输入 1: TTL输入	0
0	P0TTL0	P00 输入电平选择 0: 施密特输入 1: TTL输入	0

### 2.4.12 特殊功能端口RESINB说明(RSTM)

本产品上电默认 RESINB（P01）有效，如果需要把该口做 GPIO 使用，那需要通过寄存器关掉复位功能，寄存器描述如下：

位	符号	描述	复位值
7:1	--	保留为0	0
0	RSTM	RESINB引脚的外部复位功能屏蔽 0: RESINB外部复位引脚 1: RESINB做GPIO引脚	0

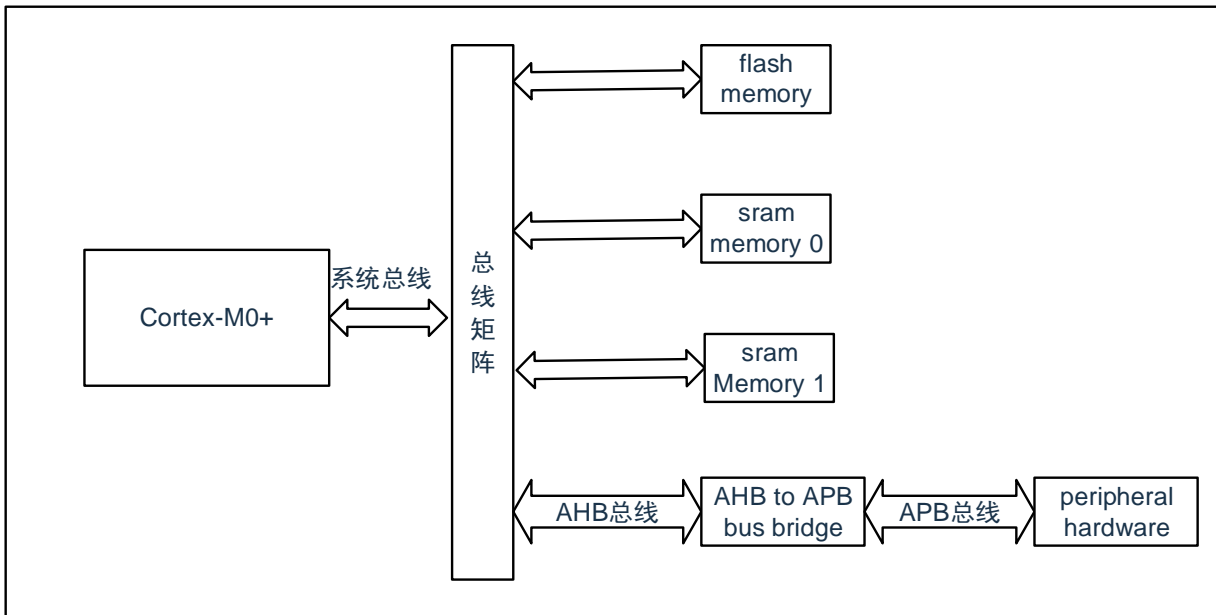
## 第3章 系统结构

### 3.1 概述

本产品系统由以下部分组成：

- 1个AHB总线Master：
  - Cortex-M0+
- 4个AHB总线Slaves：
  - - FLASH存储器
  - - SRAM存储器0
  - - SRAM存储器1
  - - AHB to APB Bridge，包含所有APB接口外设

图3-1 系统结构示意图



- 1) 系统总线: 此总线连接Cortex-M0+内核的系统总线(外设总线)到总线矩阵。
- 2) 总线矩阵: 总线矩阵协调内核系统总线其他总线的访问。
- 3) AHB to APB Bridge : AHB to APB Bridge 在AHB和APB总线间提供同步连接。有关连接到每个桥的不同外设的地址映射请参考表3-1。



### 3.2 系统地址划分

图3-2 地址区域划分示意图



## 外设地址分配

表3-1 外设的寄存器组起始地址

起始地址	外设	备注
0x4000_0000 - 0x4000_4FFF	保留	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	保留	
0x4000_7000 - 0x4001_FFFF	保留	
0x4002_0000 - 0x4002_03FF	FLASH 控制	
0x4002_0400 - 0x4002_0FFF	时钟控制	
0x4002_1000 - 0x4002_1001	看门狗定时器	
0x4002_1002 - 0x4002_17FF	保留	
0x4002_1800 - 0x4002_1BFF	高速 CRC	详见第 29 章 安全功能
0x4002_1C00 - 0x4002_1FFF	时钟控制	
0x4002_2000 - 0x4003_FFFF	保留	
0x4004_0000 - 0x4004_0F9F	GPIO	
0x4004_0FA0 - 0x4004_10FF	时钟输出	
0x4004_1100 - 0x4004_1A2F	保留	
0x4004_1A30 - 0x4004_1D7F	IICA	
0x4004_1D80 - 0x4004_217F	通用定时器单元	
0x4004_2180 - 0x4004_31FF	保留	
0x4004_3200 - 0x4004_33FF	通用 CRC	详见第 29 章 安全功能
0x4004_3400 - 0x4004_4B4F	保留	
0x4004_4B50 - 0x4004_4B50	LSITIMER	
0x4004_4B51 - 0x4004_5AFF	保留	
0x4004_5B00 - 0x4004_5BFF	外部中断控制	
0x4004_5C00 - 0x4006_0FFF	保留	
0x4006_1000 - 0x4006_1FFF	TIMER01	
0x4006_2000 - 0x4006_2FFF	保留	
0x4006_3000 - 0x4006_3FFF	SPI	
0x4006_4000 - 0x4006_40FF	UART0	
0x4006_4100 - 0x4006_41FF	UART1	
0x4006_4200 - 0x4006_42FF	EPWM	
0x4006_4300 - 0x4006_437F	CCP	
0x4006_4380 - 0x4006_447F	保留	
0x4006_4480 - 0x4006_44BF	DIVSQRT	
0x4006_44C0 - 0x4006_44FF	DIV	
0x4006_4500 - 0x4006_453F	HALL	
0x4006_4540 - 0x4006_7FFF	保留	
0x4006_8000 - 0x4006_80FF	ADC	
0x4006_8100 - 0x4006_81FF	保留	

0x4006_8200 – 0x4006_823F	ACMP0	
0x4006_8240 – 0x4006_82FF	保留	
0x4006_8300 – 0x4006_831F	PGA0/1/2/3	
0x4006_8320 – 0x4006_833F	保留	
0x4006_8340 – 0x4006_835F	ADCLDO	
0x4006_8360 – 0x4006_836F	DAC	
0x4006_8370 – 0x4006_8FFF	保留	
0x4006_9000 – 0x4006_901F	TEST	见 33 章节

## 第4章 时钟发生电路

### 4.1 时钟发生电路的功能

时钟发生电路是产生给CPU和外围硬件提供时钟的电路。有以下2种系统时钟和时钟振荡电路。

#### (1) 主系统时钟

**X1振荡电路**：能通过给X1引脚和X2引脚连接谐振器使 $f_x=4\sim 8\text{MHz}$ 的时钟振荡，并且能通过进入深度睡眠模式或者设置MSTOP位（CSC寄存器的bit7）使振荡停止。

**高速内部振荡器（高速OCO）**：能通过选项字节（000C2H）从 $f_{IH}=72\text{MHz}$ 、64MHz、36MHz、32MHz、18MHz、16MHz、9MHz、8MHz、4.5MHz、4MHz和2MHz (TYP.)中选择频率进行振荡。在解除复位后，CPU一定以 $f_{IH}$ 时钟开始运行。能通过进入深度睡眠模式或者设置HIOSSTOP位（CSC寄存器的bit0）使振荡停止。能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改选项字节设置的频率。有关频率设置，请参照“高内部振荡器的频率选择寄存器（HOCODIV）的设置”。

**PLL锁相环电路**：能通过倍频X1振荡电路或者分频后的高速内部振荡器，给系统时钟提供最高72MHz时钟，最小48MHz时钟。通过PLL控制器PLLCR来控制PLL的振荡和停止。

能通过设置 MCM0 位（系统时钟控制寄存器（CKC）的 bit4）进行 X1 时钟和高速内部振荡器时钟的切换。

#### (2) 副系统时钟

**低速内部振荡器（低速OCO）**：当选项字节（000C0H）的bit4（WDTON）或者12位间隔定时器运行时钟选择寄存器（OSMC）的bit4（WUTMMCK0）为“1”时，或者低速内部振荡器时钟选择寄存器（SUBCKSEL）的bit0（SELLOSC）为“1”时，低速内部振荡器振荡。

但是，在WDTON位为“1”并且WUTMMCK0位和SELLOSC位均为“0”，而且选项字节（000C0H）的bit0（WDSTBYON）为“0”，如果进入深度睡眠模式或睡眠模式，低速内部振荡器就停止振荡。

注： $f_{HOCO}$ ：高速内部振荡器的时钟频率

$f_{IH}$ ：高速内部振荡器分频后的时钟频率

$f_{IL}$ ：低速内部振荡器的时钟频率

$f_x$ ：高速外部晶振时钟振荡频率

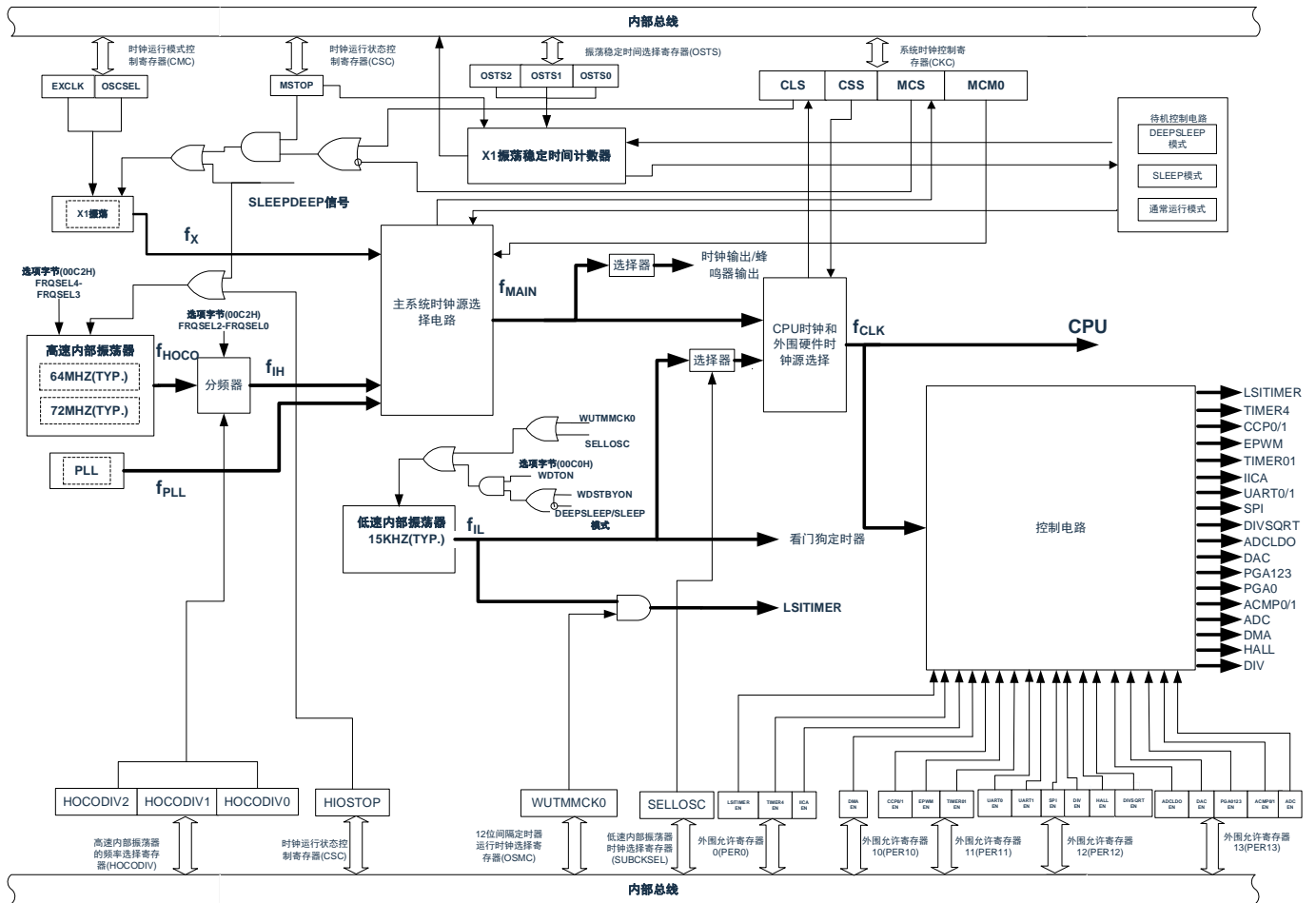
## 4.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表4-1 时钟发生电路的结构

项目	结构
控制寄存器	时钟运行模式控制寄存器 (CMC) 系统时钟控制寄存器 (CKC) 时钟运行状态控制寄存器 (CSC) 振荡稳定时间计数器的状态寄存器 (OSTC) 振荡稳定时间选择寄存器 (OSTS) 外围允许寄存器 (PER0、PER10、PER11、PER12、PER13) 高速内部振荡器的频率选择寄存器 (HOCODIV) 高速内部振荡器的微调寄存器 (HIOTRM)
振荡电路	高速内部振荡器 低速内部振荡器 X1振荡电路

图4-1 时钟发生电路的框图



- 注：  
 $f_{HOCO}$ ：高速内部振荡器的时钟频率  
 $f_{IH}$ ：高速内部振荡器分频后的时钟频率  
 $f_{MAIN}$ ：主系统时钟频率  
 $f_{CLK}$ ：CPU/外围硬件的时钟频率  
 $f_{IL}$ ：低速内部振荡器的时钟频率  
 $f_X$ ：高速外部晶振时钟振荡频率  
 $f_{PLL}$ ：PLL时钟频率

## 4.3 寄存器映射

(以下寄存器基地址 = 0x4002\_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CMC	0x000	R/W	时钟运行模式控制寄存器	0x00
CSC	0x001	R/W	时钟运行状态控制寄存器	0xC0
OSTC	0x002	RO	振荡稳定时间计数器的状态寄存器	0x00
OSTS	0x003	R/W	振荡稳定时间选择寄存器	0x07
CKC	0x004	R/W	系统时钟控制寄存器	0x00
SUBCKSEL	0x007	R/W	低速内部振荡器时钟选择寄存器	0x00
PMUKEY	0x008	WO	供电模式控制保护寄存器	0x00
PMUCTL	0x00A	R/W	供电模式控制寄存器	0x00
PER0	0x020	R/W	外围允许寄存器0	0x00
OSMC	0x023	R/W	12位间隔定时器运行时钟选择寄存器	0x00

(以下寄存器基地址 = 0x4002\_0810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER10	0x000	R/W	外围允许寄存器10	0x00
PER11	0x001	R/W	外围允许寄存器11	0x00
PER12	0x002	R/W	外围允许寄存器12	0x00
PER13	0x003	R/W	外围允许寄存器13	0x00

(以下寄存器基地址 = 0x4002\_1C00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
HOCODIV	0x020	R/W	高速内部振荡器的频率选择寄存器	选项字节 (000C2H) 的 FRQSEL[2:0]位的设置值

(以下寄存器基地址 = 0x4002\_0C00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
MCKC	0x000	R/W	系统时钟主控制寄存器	0x00
PLLCR	0x004	R/W	PLL控制寄存器	0x00

## 4.4 寄存器说明

### 4.4.1 系统时钟控制寄存器（CKC）

这是选择CPU/外围硬件时钟和主系统时钟的寄存器。

通过8位存储器操作指令设置CKC寄存器。

位	符号	描述	复位值
7	CLS <sup>注1</sup>	CPU/外围硬件时钟( $f_{CLK}$ )的状态 0: 主系统时钟( $f_{MAIN}$ ) 1: 低速内部振荡器时钟 ( $f_{IL}$ )	0
6	CSS	CPU/外围硬件时钟( $f_{CLK}$ )的选择 0: 主系统时钟( $f_{MAIN}$ ) 1: 低速内部振荡器时钟 ( $f_{IL}$ )	0
5	MCS <sup>注1</sup>	主系统时钟时钟( $f_{MAIN}$ )的状态 0: 高速内部振荡时钟或PLL作为主系统时钟 ( $f_{MAIN}$ ) 1: X1振荡时钟作为主系统时钟 ( $f_{MAIN}$ )	0
4	MCM0	主系统时钟时钟( $f_{MAIN}$ )的选择 0: 高速内部振荡时钟或PLL作为主系统时钟 ( $f_{MAIN}$ ) 1: X1振荡时钟作为主系统时钟 ( $f_{MAIN}$ )	0
3:0	--	保留	0x0

注1: bit5、bit7是只读位。

注2: 给CPU和外围硬件提供CSS位设置的时钟。如果更改CPU时钟，就同时更改外围硬件的时钟（时钟输出和看门狗定时器除外）。因此，如果要更改CPU/外围硬件的时钟，就必须停止各外围功能。



## 4.4.2 时钟运行状态控制寄存器 (CSC)

这是控制高速内部振荡器时钟运行的寄存器。通过8位存储器操作指令设置CSC寄存器。  
在产生复位信号后，此寄存器的值变为“C0H”。

位	符号	描述	复位值
7	MSTOP	X1振荡时钟运行控制 0: X1振荡电路运行 1: X1振荡电路停止	1
6:1	--	保留	0x20
0	HIOSTOP	高速内部振荡器时钟的运行控制 0: 高速内部振荡器运行 1: 高速内部振荡器停止运行	0

注1: 不能通过CSC寄存器停止被选择为CPU/外围硬件时钟 (fCLK) 的时钟;

注2: 使用X1振荡时钟, 还需将CMC寄存器配置为X1振荡模式;

注3: 有关用于停止时钟振荡的寄存器标志设置和停止前的条件, 请参照表4-2。

表4-2 时钟停止方法

时钟	时钟停止前的条件	设置CSC寄存器的标志
高速内部振荡器时钟	CPU/外围硬件时钟以高速内部振荡器时钟或PLL时钟以外的时钟运行。 (CLS=1)	HIOSTOP=1

## 4.4.3 PLL控制寄存器(PLLCR)

位	符号	描述	复位值
31:8	-	保留	-
7	PLLCKSEL	PLL输入时钟选择: 0: 高速内部振荡器时钟分频后的时钟 1: X1振荡时钟	0
6:4	-	保留	-
3:1	PLL P	PLL时钟倍频选择位: 000: 6倍频 001: 9倍频 010: 12倍频 011: 18倍频 100: 8倍频 101: 12倍频 110: 16倍频 111: 24倍频	0x0
0	PLLON	PLL模块使能: 0: 禁止 1: 使能	0

### 4.4.4 系统时钟主控制寄存器(MCKC)

位	符号	描述	复位值
31:8	-	保留	-
7	PLLCKSTR	系统时钟选择PLL切换成功状态位： 0: 系统时钟选择 $f_{IH}$ 输出 1: 系统时钟选择PLL输出	0
6:3	-	保留	-
2:1	RDIV	PLL输出时钟的分频系数选择位： 00: 不分频 01: 2分频 10: 4分频 11: 8分频	0x0
0	MCKSEL	系统时钟选择位： 0: 选择 $F_{IH}$ 输出 1: 选择PLL输出	0

注：PLL的时钟输出范围48MHz~72MHz。

### 4.4.5 时钟运行模式控制寄存器(CMC)

这是设置 X1、X2 引脚的运行模式的寄存器。

在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。能通过 8 位存储器操作指令读此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述					复位值
		X1、X2引脚运行模式控制					
		EXCLK	OSCSEL	高速系统始终引脚的运行模式	X1 引脚	X2/ECLK引脚	
7	EXCLK	0	0	端口模式	输入/输出端口		0
		0	1	X1振荡模式	连接晶体或者陶瓷谐振器。		
6	OSCSEL	1	0	端口模式	输入/输出端口		0
		1	1	禁止设置			
3:0	-	保留					0x0

注1：在解除复位后，只能通过8位存储器操作指令写1次CMC寄存器。当以初始值（“00H”）使用CMC寄存器时，为了防止程序失控时的误动作（如果误写“00H”以外的值就不能恢复），必须在解除复位后将CMC寄存器置“00H”。

注2：在解除复位后并且在通过设置时钟运行状态控制寄存器（CSC）开始X1振荡前，必须设置CMC寄存器。

注3：系统时钟的频率上限为72MHz，但是X1振荡电路的频率上限为8MHz。

### 4.4.6 振荡稳定时间选择寄存器(OSTS)

这是选择 X1 时钟的振荡稳定时间的寄存器。

如果使 X1 时钟振荡，就在 X1 振荡电路运行 (MSTOP=0) 后自动等待 OSTS 寄存器设置的时间。

如果将 CPU 时钟从高速内部振荡器时钟或者低速内部振荡器时钟切换到 X1 时钟，或者如果 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式，就必须通过振荡稳定时间计数器的状态寄存器 (OSTC) 确认是否经过振荡稳定时间。

能通过 OSTC 寄存器确认 OSTS 寄存器事先设置的时间。

在产生复位信号后，此寄存器的值变为“07H”。

位	符号	描述						复位值
7:3	-	保留						0x0
2:0	OSTS[2:0]	振荡稳定时间的选择						0x7
		OSTS2	OSTS1	OSTS0		$f_x=4\text{MHz}$	$f_x=8\text{MHz}$	
		0	0	0	$2^8/f_x$	64us	32us	
		0	0	1	$2^9/f_x$	128us	64us	
		0	1	0	$2^{10}/f_x$	256us	128us	
		0	1	1	$2^{11}/f_x$	512us	256us	
		1	0	0	$2^{13}/f_x$	2.04ms	1.02ms	
		1	0	1	$2^{15}/f_x$	8.17ms	4.07ms	
		1	1	0	$2^{17}/f_x$	32.7ms	16.3ms	
1	1	1	$2^{18}/f_x$	65.5ms	32.7ms			

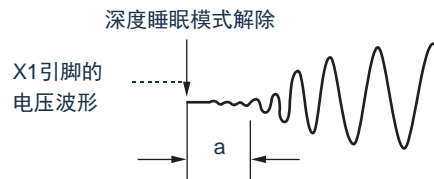
注1：要更改OSTS寄存器的设置时，必须在将时钟运行状态控制寄存器 (CSC) 的MSTOP位置“0”前进行更改。

注2：振荡稳定时间计数器只在OSTS寄存器所设振荡稳定时间内进行计数。

在以下情况下，OSTS寄存器的振荡稳定时间的设置值必须大于在开始振荡后通过OSTC寄存器确认的计数值。

- 当CPU时钟为高速内部振荡器时钟或者低速内部振荡器时钟并且要开始X1时钟的振荡时；
- 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式时（因此必须注意，解除深度睡眠模式后的OSTC寄存器只设置OSTS寄存器所设振荡稳定时间内的状态）。

注3：X1时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图a）。



### 4.4.7 振荡稳定时间计数器的状态寄存器 (OSTC)

这是表示 X1 时钟的振荡稳定时间计数器计数状态的状态寄存器。能在以下情况下确认 X1 时钟的振荡稳定时间：

当 CPU 时钟为高速内部振荡器时钟或者低速内部振荡器时钟并且开始 X1 时钟的振荡时。

当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到深度睡眠模式后解除睡眠模式时。

能通过 8 位存储器操作指令读 OSTC 寄存器。

通过复位信号的产生、进入深度睡眠模式或者 MSTOP 位（时钟运行状态控制寄存器（CSC）的 bit7）为“1”，此寄存器的值变为“00H”。

备注：在以下情况下，振荡稳定时间计数器开始计数：

- 1) 当X1时钟开始振荡（EXCLK、OSCSEL=0、1 MSTOP=0）时。
- 2) 当解除深度睡眠模式时。

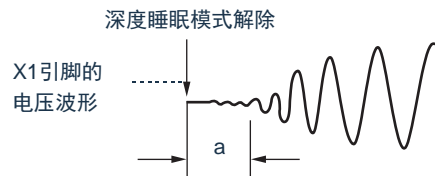
位	符号	描述											复位值			
		振荡稳定时间状态												$f_x=4\text{MHz}$	$f_x=8\text{MHz}$	
		MOST8	MOST9	MOST10	MOST11	MOST12	MOST13	MOST14	MOST15	MOST16	MOST17	MOST18				
7	MOST8	0	0	0	0	0	0	0	0	0	0	0	小于 $2^8/f_x$	小于64us	$f_x=8\text{MHz}$	0
6	MOST9	1	0	0	0	0	0	0	0	0	0	0	至少 $2^8/f_x$	至少64us	小于32us	0
5	MOST10	1	1	0	0	0	0	0	0	0	0	0	至少 $2^9/f_x$	至少128us	至少32us	0
4	MOST11	1	1	1	0	0	0	0	0	0	0	0	至少 $2^{10}/f_x$	至少256us	至少64us	0
3	MOST12	1	1	1	1	0	0	0	0	0	0	0	至少 $2^{11}/f_x$	至少512us	至少128us	0
2	MOST13	1	1	1	1	1	0	0	0	0	0	0	至少 $2^{13}/f_x$	至少2.04ms	至少256us	0
1	MOST14	1	1	1	1	1	1	0	0	0	0	0	至少 $2^{15}/f_x$	至少8.17ms	至少1.02ms	0
0	MOST15	1	1	1	1	1	1	1	1	0	0	0	至少 $2^{17}/f_x$	至少32.7ms	至少4.07ms	0
		1	1	1	1	1	1	1	1	1	1	1	至少 $2^{18}/f_x$	至少65.5ms	至少16.3ms	

注1：在经过上述时间后，各位从MOST8位开始依次变为“1”并且保持“1”的状态。

注2：振荡稳定时间计数器只在振荡稳定时间选择寄存器（OSTS）所设振荡稳定时间内进行计数。在以下情况下，OSTS寄存器的振荡稳定时间的设置值必须大于在开始振荡后通过OSTC寄存器确认的计数值。

- 当CPU时钟为高速内部振荡器时钟或者低速内部振荡器时钟并且要开始X1时钟的振荡时；
- 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式时（因此必须注意，解除深度睡眠模式后的OSTC寄存器只设置OSTS寄存器所设振荡稳定时间内的状态）。

注3：X1时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图a）。



## 4.4.8 外围允许寄存器（PER0、PER10、PER11、PER12、PER13）

这是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。当使用由这些寄存器控制的以下外围功能时，必须在进行外围功能的初始设置前将对应位置“1”。寄存器功能汇总：

PER0	LSITIMER, IICA, TIMER4
PER10	DMA
PER11	CCP0/1, EPWM, TMER01
PER12	HALL, DIV, UART1, UART0, SPI, DIVSQRT
PER13	ADCLDO, DAC, PGA, ACMP, ADC

通过8位存储器操作指令设置PER0寄存器和PER10、PER11、PER12、PER13寄存器。  
在产生复位信号后，这些寄存器的值变为“00H”。

### 外围允许寄存器0（PER0）

位	符号	描述	复位值
7	LSITIMEREN	提供LSITIMER输入时钟的控制(掉电睡眠可行) 0: 停止提供输入时钟，不能写LSITIMER使用的SFR 1: 提供输入时钟，能写LSITIMER使用的SFR	0
6:5	--	保留	0x0
4	IICAEN	提供IICA模块输入时钟的控制 0: 停止提供输入时钟，不能写IICA使用的SFR 1: 提供输入时钟，能写IICA使用的SFR	0
3:1	--	保留	0x0
0	TM40EN	提供通用定时器单元0输入时钟的控制 0: 停止提供输入时钟，不能写通用定时器单元使用的SFR 1: 提供输入时钟，能写通用定时器单元0使用的SFR	0

### 外围允许寄存器10（PER10）

位	符号	描述	复位值
7:4	--	保留	0x0
3	DMAEN	提供DMA模块输入时钟的控制 0: 停止提供输入时钟，DMA不能运行 1: 提供输入时钟，DMA能运行	0
2:0	--	保留	0x0

外围允许寄存器11 (PER11)

位	符号	描述	复位值
7:5	--	保留	0x0
4	CCPEN	提供CCP模块输入时钟的控制 0: 停止提供输入时钟, CCP不能运行 1: 提供输入时钟, CCP能运行	0
3	EPWMEN	提供EPWM模块输入时钟的控制 0: 停止提供输入时钟, EPWM不能运行 1: 提供输入时钟, EPWM能运行	0
2:1	--	保留	0x0
0	TIMER01EN	提供TIMER01的输入时钟的控制 0: 停止提供输入时钟, TIMER01不能运行 1: 提供输入时钟, TIMER01能运行	0

外围允许寄存器12 (PER12)

位	符号	描述	复位值
7	TESTEN	测试寄存器时钟使能 0: 停止提供输入时钟, TEST模块不能运行 1: 提供输入时钟, TEST模块能运行	0
6	HALLEN	提供HALL模块输入时钟的控制 0: 停止提供输入时钟, HALL模块不能运行 1: 提供输入时钟, HALL模块能运行	0
5	DIVEN	提供除法单元DIV的输入时钟的控制 0: 停止提供输入时钟, DIV不能运行 1: 提供输入时钟, DIV能运行	0
4	--	保留	--
3	UART1EN	提供UART模块输入时钟的控制 0: 停止提供输入时钟, UART不能运行 1: 提供输入时钟, UART能运行	0
2	UART0EN	提供UART模块输入时钟的控制 0: 停止提供输入时钟, UART不能运行 1: 提供输入时钟, UART能运行	0
1	SPIEN	提供SPI模块输入时钟的控制 0: 停止提供输入时钟, SPI不能运行 1: 提供输入时钟, SPI能运行	0
0	DIVSQRTEN	提供除法与开方运算单元DIVSQRT的输入时钟的控制 0: 停止提供输入时钟, DIVSQRT不能运行 1: 提供输入时钟, DIVSQRT能运行	0

## 外围允许寄存器13 (PER13)

位	符号	描述	复位值
7	-	保留	-
6	ADCLDOEN	提供ADCLDO模块输入时钟的控制 0: 停止提供输入时钟, ADCLDO不能运行 1: 提供输入时钟, ADCLDO能运行	0
5	DACEN	提供DAC模块输入时钟的控制 0: 停止提供输入时钟, DAC不能运行 1: 提供输入时钟, DAC能运行	0
4	-	保留	-
3	PGAEN	提供PGA0/1/2/3模块输入时钟的控制 0: 停止提供输入时钟, PGA0/1/2/3不能运行 1: 提供输入时钟, PGA0/1/2/3能运行	0
2	-	保留	-
1	ACMPEN	提供ACMP0/1模块输入时钟的控制 0: 停止提供输入时钟, ACMP0/1不能运行 1: 提供输入时钟, ACMP0/1能运行	0
0	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟, ADC不能运行 1: 提供输入时钟, ADC能运行	0

### 4.4.9 12位间隔定时器运行时钟选择寄存器 (OSMC)

通过OSMC寄存器选择12位间隔定时器LSITIMER的运行时钟。

通过8位存储器操作指令设置OSMC寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:5	--	保留	0x0
4	WUTMMCK0	12位间隔定时器的运行时钟的选择：  0: 低速内部振荡时钟停止给12位间隔定时器LSITIMER提供时钟  1: 低速内部振荡时钟为12位间隔定时器LSITIMER提供时钟	0
3:0	--	保留	0x0

### 4.4.10 高速内部振荡器的频率选择寄存器 (HOCODIV)

这是更改选项字节 (000C2H) 设置的高速内部振荡器频率的寄存器。但是，能选择的频率因选项字节 (000C2H) 的FRQSEL[4:3]位值而不同。

通过8位存储器操作指令设置HOCODIV寄存器。

在产生复位信号后，此寄存器的值变为选项字节 (000C2H) 的FRQSEL[2:0]位的设置值。

位	符号	描述	复位值		
7:3	--	保留	0		
2:0	HOCODIV[2:0]	高速内部振荡器时钟频率的选择	选项字节 (000C2H) 的FRQSEL[2:0]位的设置值		
				FSQSEL[4:3]=11/10	FSQSEL[4:3]=01/00
		000		$f_{IH}=72\text{MHZ}$ $f_{HOCO}=72\text{MHZ}$	$f_{IH}=64\text{MHZ}$ $f_{HOCO}=64\text{MHZ}$
		001		$f_{IH}=36\text{MHZ}$ $f_{HOCO}=72\text{MHZ}$	$f_{IH}=32\text{MHZ}$ $f_{HOCO}=64\text{MHZ}$
		010		$f_{IH}=18\text{MHZ}$ $f_{HOCO}=72\text{MHZ}$	$f_{IH}=16\text{MHZ}$ $f_{HOCO}=64\text{MHZ}$
		011		$f_{IH}=9\text{MHZ}$ $f_{HOCO}=72\text{MHZ}$	$f_{IH}=8\text{MHZ}$ $f_{HOCO}=64\text{MHZ}$
		100		$f_{IH}=4.5\text{MHZ}$ $f_{HOCO}=72\text{MHZ}$	$f_{IH}=4\text{MHZ}$ $f_{HOCO}=64\text{MHZ}$
		101		禁止设置	$f_{IH}=2\text{MHZ}$ $f_{HOCO}=64\text{MHZ}$
	上述以外	禁止设置			

注1：必须在选择高速内部振荡器时钟 ( $f_{IH}$ ) 作为CPU/外围硬件时钟 ( $f_{CLK}$ ) 的状态下设置HOCODIV寄存器。

注2：在通过HOCODIV寄存器更改频率后，经过以下转移时间之后进行频率切换：

- 以更改前的频率，最多进行3个时钟的运行。
- 以更改后的频率，最多等待3个CPU/外围硬件的时钟。



### 4.4.11 低速内部振荡器时钟选择寄存器 (SUBCKSEL)

SUBCKSEL寄存器是选择低速内部振荡器时钟 $f_{IL}$ 的寄存器。

通过8位存储器操作指令设置SUBCKSEL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	--	保留	0x0
0	SELLOCO	低速内部振荡器时钟选择 0: 禁止选择低速内部振荡器时钟 1: 选择低速内部振荡器时钟	0

### 4.4.12 供电模式控制保护寄存器 (PMUKEY)

PMUKEY寄存器是供电模式控制PMUCTL的保护的寄存器。

通过16位存储器操作指令设置PMUKEY寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15:0	PMUKEY	供电模式控制保护寄存器的选择 •解除 PMUCTL写入保护。通过先后写入192AH、3E4FH到PMUKEY使能PMUCTL的PVDNEN位的写入控制 •其他。PMUCTL写入设置无效	0x0000

### 4.4.13 供电模式控制寄存器 (PMUCTL)

PMUCTL寄存器是控制使能供电控制模式的寄存器。

通过8位存储器操作指令设置PMUCTL寄存器。

在产生复位信号后，此寄存器的值变为“00H”并且写入保护打开，通过PMUKEY解除写入控制。

位	符号	描述	复位值
7:1	--	保留	0x0
0	PVDNEN	供电模式控制寄存器的选择 0: 部分掉电模式禁止 1: 部分掉电模式使能	0

注：通过PMUKEY解除PMUCTL的写入保护。

## 4.5 系统时钟振荡电路

### 4.5.1 X1振荡电路

X1 振荡电路通过连接 X1 引脚和 X2 引脚的晶体谐振器或者陶瓷谐振器（4~8MHz）进行振荡。

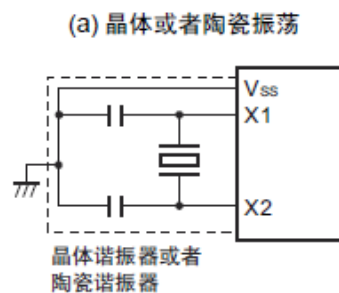
当使用 X1 振荡电路时，必须对时钟运行模式控制寄存器（CMC）的 bit7 和 bit6（EXCLK、OSCSEL）进行以下的设置：

晶体或者陶瓷振荡：EXCLK、OSCSEL=0、1

当不使用 X1 振荡电路时，必须设置为端口模式（EXCLK、OSCSEL=0、0）。

X1 振荡电路的外接电路例子如下图所示。

图4-2 X1振荡电路的外接电路例子



注：当使用X1振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图4-2中的虚线部分进行布线：

- a) 必须尽量缩短布线。
- b) 不能和其他的信号线交叉，并且不能接近有变化的大电流流过的布线
- c) 必须始终保持振荡电路的电容器接地点和VSS同电位，而且不能给大电流流过的接地图形接地
- d) 不能从振荡电路取出信号

### 4.5.2 高速内部振荡器

CMS32M67xx内置高速内部振荡器。能通过选项字节（000C2H）从 $f_{IH}$ =72MHz、64MHz、36MHz、32MHz、18MHz、16MHz、9MHz、8MHz、4.5MHz、4MHz和2MHz中选择频率。能通过时钟运行状态控制寄存器(CSC)的bit0(HIOSTOP)控制振荡。

在解除上电复位后，高速内部振荡器自动开始振荡。

### 4.5.3 低速内部振荡器

CMS32M67xx内置低速内部振荡器。

低速内部振荡器时钟用作看门狗定时器、LSITIMER，以及SysTick定时器的外部参考时钟，也可用作CPU时钟和外围模块时钟。

当选项字节（000C0H）的bit4（WDTON）或者12位间隔定时器运行时钟选择寄存器（OSMC）的bit4（WUTMMCK0）为“1”时，或者低速内部振荡器时钟选择寄存器(SUBCKSEL)的bit0(SELLOSC)为“1”时，低速内部振荡器振荡。

当看门狗定时器停止运行并且WUTMMCK0位不为“0”时，低速内部振荡器继续振荡。但是，在WUTMMCK0为0且SELLOSC为0，WDTON位为“1”并且选项字节（000C0H）的bit0（WDSTBYON）为“0”时，如果进入深度睡眠模式或睡眠模式，低速内部振荡器就停止振荡。在看门狗定时器运行时，即使程序失控，低速内部振荡器时钟也不停止运行。

## 4.6 时钟发生电路的运行

时钟发生电路产生以下所示各种时钟，并且控制待机模式等CPU的运行模式（参照图4-1）。

$F_{MAIN}$ : 主系统时钟

$F_{IH}$ : 高速内部振荡器时钟

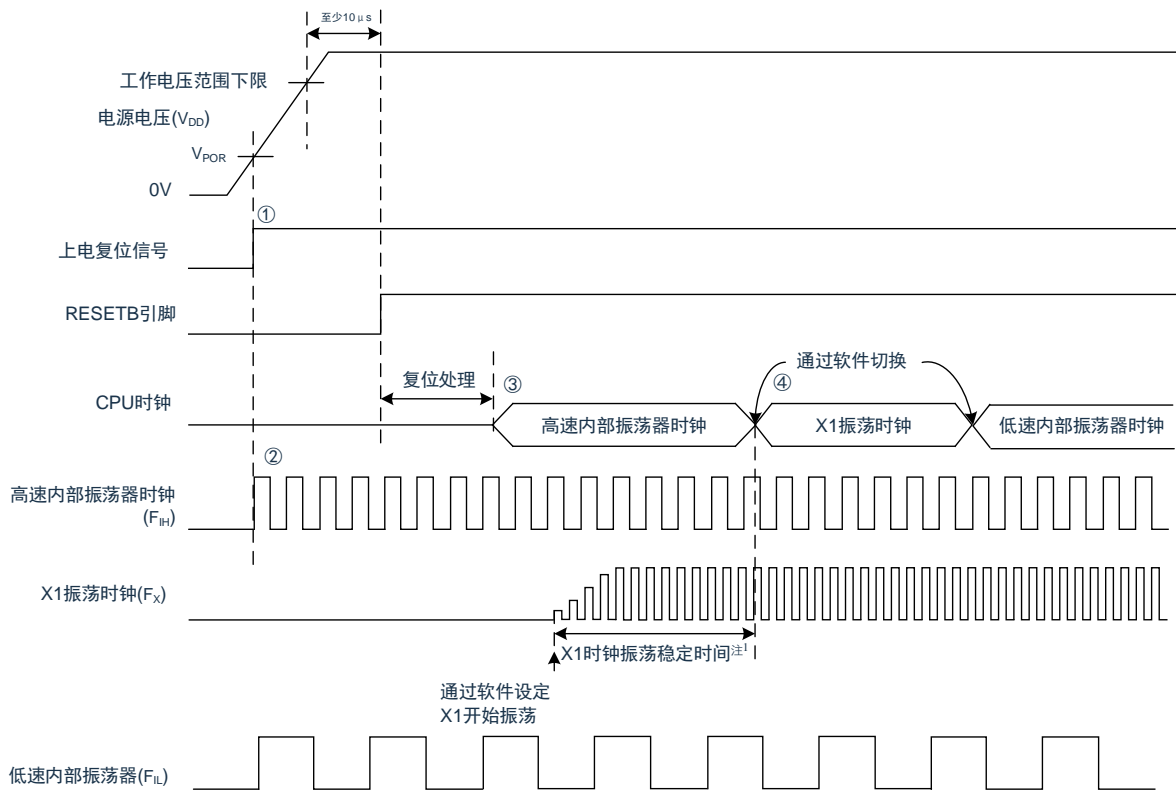
$F_{IL}$ : 低速内部振荡器时钟

$F_{CLK}$ : CPU/外围硬件时钟

$F_X$ : 高速外部晶振时钟振荡频率

CMS32M67xx 在解除复位后，CPU 通过高速内部振荡器的输出开始运行。接通电源时的时钟发生电路的运行如 4-3 所示。

图4-3 接通电源时的时钟发生电路的运行



- 1) 在接通电源后，通过上电复位(POR)电路产生内部复位信号。
- 2) 但是，在达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态（上图是使用外部复位时的例子）。
- 3) 如果解除复位，高速内部振荡器就自动开始振荡。
- 4) 在解除复位后，进行电压稳定等待和复位处理，然后 CPU 以高速内部振荡器时钟开始运行。
- 5) 必须通过软件设置 X1 时钟的开始振荡（参照“4.7.2 X1 振荡电路的设置例子”）。
- 6) 如果要将 CPU 时钟切换到 X1 时钟，就必须在等待时钟振荡稳定后通过软件设置切换（参照“4.7.2 X1 振荡电路的设置例子”）。

注：当解除复位时，必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认X1时钟的振荡稳定时间。

## 4.7 时钟控制

### 4.7.1 高速内部振荡器的设置例子

在解除复位后，CPU/外围硬件时钟 ( $f_{CLK}$ ) 一定以高速内部振荡器时钟运行。能通过选项字节 (000C2H) 的 FRQSEL0~FRQSEL4位，从72MHz、64MHz、36MHz、32MHz、18MHz、16MHz、9MHz、8MHz、4.5MHz、4MHz和2MHz中选择高速内部振荡器的频率。另外，能通过高速内部振荡器的频率选择寄存器 (HOCODIV) 更改频率。

选项字节000C2的设置

位	符号	描述			复位值
7:5	--	保留			--
4:0	FRQSEL[4:0]	高速内部振荡器时钟频率的选择			--
		FRQSEL[4:0]的值	$f_{HOCO}$	$f_{IH}$	
		5'b1x000	72MHZ	72MHZ	
		5'b1x001	72MHZ	36MHZ	
		5'b1x010	72MHZ	18MHZ	
		5'b1x011	72MHZ	9MHZ	
		5'b1x100	72MHZ	4.5MHZ	
		5'b0x000	64MHZ	64MHZ	
		5'b0x001	64MHZ	32MHZ	
		5'b0x010	64MHZ	16MHZ	
		5'b0x011	64MHZ	8MHZ	
		5'b0x100	64MHZ	4MHZ	
		5'b0x101	64MHZ	2MHZ	
		上述以外	禁止设置		

#### 【高速内部振荡器的频率选择寄存器 (HOCODIV) 的设置】

位	符号	描述			复位值
7:3	--	保留			0
2:0	HOCODIV[2:0]	高速内部振荡器时钟频率的选择			选项字节 (000C2H) 的 FRQSEL[2:0]位的设置值
		HOCODIV[2:0]的值	FSQSEL[4,3]=11/10	FSQSEL[4,3]=01/00	
		3'b000	$f_{IH}=72MHZ$ $f_{HOCO}=72MHZ$	$f_{IH}=64MHZ$ $f_{HOCO}=64MHZ$	
		3'b001	$f_{IH}=36MHZ$ $f_{HOCO}=72MHZ$	$f_{IH}=32MHZ$ $f_{HOCO}=64MHZ$	
		3'b010	$f_{IH}=18MHZ$ $f_{HOCO}=72MHZ$	$f_{IH}=16MHZ$ $f_{HOCO}=64MHZ$	
		3'b011	$f_{IH}=9MHZ$ $f_{HOCO}=72MHZ$	$f_{IH}=8MHZ$ $f_{HOCO}=64MHZ$	
		3'b100	$f_{IH}=4.5MHZ$ $f_{HOCO}=72MHZ$	$f_{IH}=4MHZ$ $f_{HOCO}=64MHZ$	
		3'b101	禁止设置	$f_{IH}=2MHZ$ $f_{HOCO}=64MHZ$	
		上述以外	禁止设置		

## 4.7.2 X1振荡电路的设置例子

在解除复位后，CPU/外围硬件时钟（ $f_{CLK}$ ）一定以高速内部振荡器时钟运行。此后，如果改为 X1 振荡时钟，就通过振荡稳定时间选择寄存器（OSTS）、时钟运行模式控制寄存器（CMC）和时钟运行状态控制寄存器（CSC）进行振荡电路的设置和振荡开始的控制，并且通过振荡稳定时间计数器的状态寄存器（OSTC）等待振荡稳定。在等待振荡稳定后通过系统时钟控制寄存器（CKC）将 X1 振荡时钟设置为  $f_{CLK}$ 。

**【寄存器的设置】**必须按照①~⑤的顺序设置寄存器。

① 将 CMC 寄存器的 EXCLK、OSCSEL 配置为 2'b01，将对应时钟输入选择为 X1 振荡模式。

bit	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	0	0	0	0	0

② 通过 OSTS 寄存器选择解除深度睡眠模式时的 X1 振荡电路的振荡稳定时间。

例) 要通过 8MHz 谐振器至少等待 128us 时，必须设置 OSTS 寄存器的值为 0x02。

bit	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

③ 将 CSC 寄存器的 MSTOP 位清“0”，使 X1 振荡电路开始振荡。

bit	7	6	5	4	3	2	1	0
CSC	MSTOP	0	0	0	0	0	0	HIOSTOP

④ 通过 OSTC 寄存器等待 X1 振荡电路的振荡稳定。

例) 要通过 8MHz 谐振器至少等待 128us 时，必须等到 OSTC 寄存器的值变为 0xE0。

bit	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18

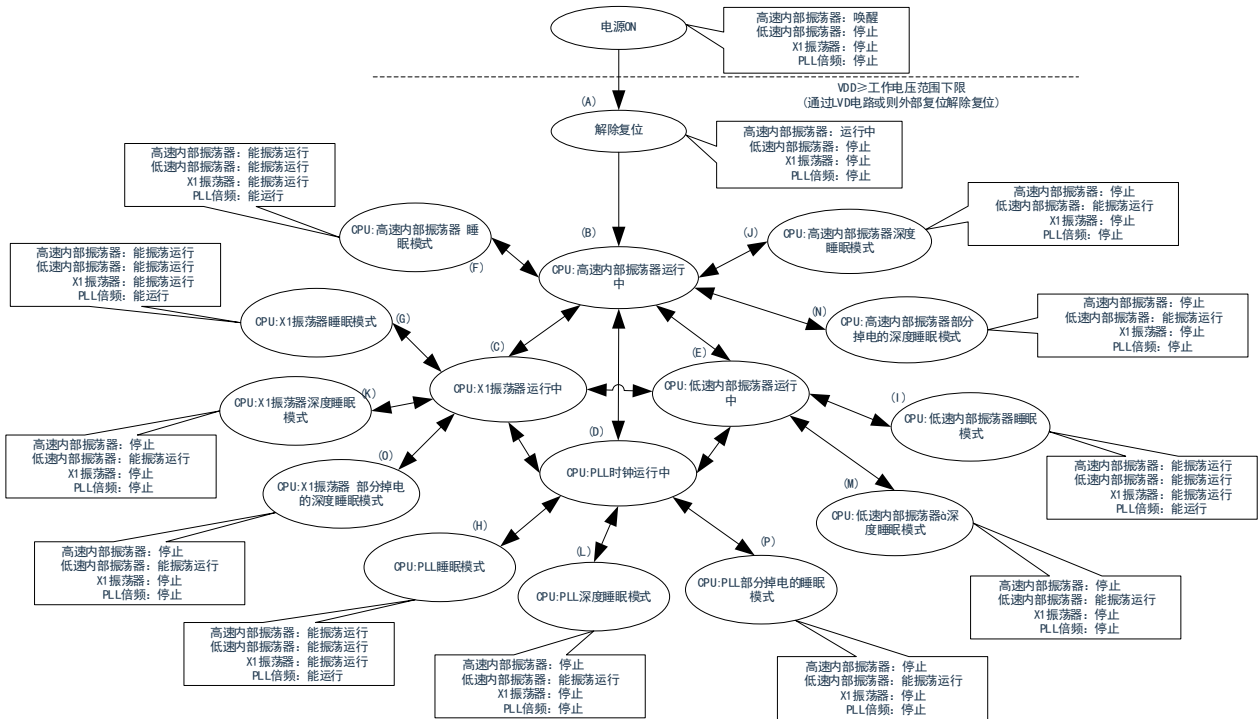
⑤ 通过 CKC 寄存器的 MCM0 位将 X1 振荡时钟设置为 CPU/外围硬件时钟。

bit	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0

### 4.7.3 CPU时钟的状态转移图

本产品的CPU时钟状态转移图如图4-4所示。

图4-4 CPU时钟的状态转移图



CPU时钟的转移和SFR寄存器的设置例子等如表4-3所示。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(1/5)

(1)在解除复位(A)后，CPU转移到高速内部振荡器时钟运行(B)。

状态转移	SFR寄存器的设置
(A)→(B)	不需要设置SFR寄存器（解除复位后的初始状态）。

(2)在解除复位(A)后，CPU转移到X1振荡器时钟运行(C)。

状态转移	SFR寄存器的设置
(B)→(C) (D)→(C) (E)→(C)	寄存器设置顺序：
	① 配置CKC寄存器的MCM0为1，选择X1振荡时钟作为主时钟输出
	② 判断CKC寄存器的MCS，等待该位为1
	③ 配置CKC寄存器的CSS为0，选择主时钟作为CPU和外围电路的时钟
	④ 判断CKC寄存器的CLS，等待该位为0

表4-3 CPU时钟的转移和SFR寄存器的设置例子(2/5)

(3) 在解除复位(A)后, CPU转移到PLL时钟运行(D)。

状态转移	SFR寄存器的设置
(B)→(D) (D)→(D) (E)→(D)	寄存器设置顺序:
	① 配置PLLCR寄存器的PLLCKSEL, PLLP, PLLON, 选择PLL的输入时钟源, 设置倍频系数, 使能PLL
	② 配置MCKC寄存器的MCKSEL位为1, 选择PLL时钟作为输出
	③ 判断MCKC寄存器的PLLCKSTR, 等待该位为1
	④ 配置CKC寄存器的MCM0为0, 选择PLL作为主时钟输出
	⑤ 判断CKC寄存器的MCS, 等待该位为0
	⑥ 配置CKC寄存器的CSS为0, 选择主时钟作为CPU和外围电路的时钟
	⑦ 判断CKC寄存器的CLS, 等待该位为0

(4) 在解除复位(A)后, CPU转移到低速内部振荡时钟运行(E)。

状态转移	SFR寄存器的设置
(B)→(E) (D)→(E) (C)→(E)	寄存器设置顺序:
	① SUBCKSEL寄存器SELLOCO为1
	② 配置CKC寄存器的CSS为1, 选择低速内部振荡时钟为CPU和外围电路的时钟
	③ 判断CKC寄存器的CLS, 等待该位为1

(5) CPU由其他时钟转移到高速内部振荡时钟运行(B)。

状态转移	SFR寄存器的设置
(C)→(B) (D)→(B) (E)→(B)	寄存器设置顺序:
	① 配置MCKC寄存器的MCKSEL位为0, 选择内部高速振荡器时钟作为输出
	② 判断MCKC寄存器的PLLCKSTR, 等待该位为0
	③ 配置CKC寄存器的MCM0为0, 选择高HOCO作为主时钟输出
	④ 判断CKC寄存器的MCS, 等待该位为0
	⑤ 配置CKC寄存器的CSS为0, 选择主时钟作为CPU和外围电路的时钟
⑥ 判断CKC寄存器的CLS, 等待该位为0	

注1: 表4-3的(A)~(P)对应图4-4的(A)~(P)。

注2: 高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。



表4-3 CPU时钟的转移和SFR寄存器的设置例子(3/5)

- (6) CPU从高速内部时钟运行(B)转移到睡眠模式(F)。  
 CPU从高速外部时钟运行(C)转移到睡眠模式(G)。  
 CPU从PLL时钟运行(D)转移到睡眠模式(H)。  
 CPU从低速内部时钟运行(E)转移到睡眠模式(I)。

状态转移	设置内容
(B)→(F) (C)→(G) (D)→(H)	执行WFI指令。

注：表4-3的(A)~(P)对应图4-4的(A)~(P)。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(4/5)

- (7) CPU从高速内部时钟运行(B)转移到深度睡眠模式(J)。  
 CPU从高速外部时钟运行(C)转移到深度睡眠模式(K)。  
 CPU从PLL时钟运行(D)转移到深度睡眠模式(L)。  
 CPU从低速内部时钟运行(E)转移到深度睡眠模式(M)。

(设置顺序)

状态转移	设置内容	
(B)→(J) (C)→(K) (D)→(L) (E)→(M)	停止不能在深度睡眠模式中运行的外围功能。	SCR寄存器bit2 (SLEEPDEEP) 置为1, 并执行WFI指令。

注：表4-3的(A)~(P)对应图4-4的(A)~(P)。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(5/5)

- (7) CPU从高速内部时钟运行(B)转移到部分掉电的深度睡眠模式(N)。  
 CPU从高速外部时钟运行(C)转移到部分掉电的深度睡眠模式(O)。  
 CPU从PLL时钟运行(D)转移到部分掉电的深度睡眠模式(P)。

(设置顺序)

状态转移	设置内容		
(B)→(N) (C)→(O) (D)→(P)	停止不能在深度睡眠模式中运行的外围功能。	PMUKEY=0x192A; PMUKEY=0x3E4F; PMUCTL=0x01;	SCR寄存器bit2 (SLEEPDEEP) 置为1, 并执行WFI指令。

注：表4-3的(A)~(P)对应图4-4的(A)~(P)。

## 4.7.4 CPU时钟转移前的条件和转移后的处理

CPU时钟转移前的条件和转移后的处理如下所示。

表4-4 有关CPU时钟的转移

CPU时钟		转移前的条件	转移后的处理
转移前	转移后		
高速内部振荡时钟	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	如果停止高速内部振荡器的振荡（HIOSTOP=1），就能减小工作电流。当PLL时钟源为高速内部振荡器时钟，CPU时钟切换到PLL时，不能停止高速内部振荡器
	PLL时钟	允许PLL运行 • PLLON=1 • PLL输入时钟源振荡稳定	
	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	
低速内部振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	若看门狗和LSITIMER没有运行，可关闭低速内部时钟振荡(SELLOCO=0) 就能减小工作电流。
	PLL时钟	允许PLL运行 • PLLON=1 • PLL输入时钟源振荡稳定	
	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	
PLL时钟	高速内部振荡时钟	允许高速内部振荡器振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	可以关闭PLL使能(PLLON=0)，以减少工作电流
	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	
	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	
X1振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	如果停止X1振荡时钟(MSTOP=1)，可以减少工作电流。当PLL时钟源为X1振荡时钟，CPU时钟切换到PLL时钟时，不能停止X1时钟振荡。
	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	
	PLL时钟	允许PLL运行 • PLLON=1 • PLL输入时钟源振荡稳定	

## 4.7.5 CPU时钟和主系统时钟的切换所需时间

能通过设置系统时钟控制寄存器(CKC)的bit6 (CSS)进行CPU时钟的切换(主系统时钟和低速内部振荡时钟, 即高速内部振荡时钟和低速内部振荡时钟)

在改写CKC寄存器后不立即进行实际的切换, 而是在更改CKC寄存器后仍然以切换前的时钟继续运行数个时钟(参照表4-5)。

能通过CKC寄存器的bit7(CLS)、bit5(MCS)和MCKC寄存器的bit7(PLLCKSTR)来判断CPU是运行时钟。

如果切换CPU时钟, 就同时切换外围硬件时钟。

表4-5 时钟切换所需要时钟数

切换前时钟 \ 切换后时钟	高速内部振荡器时钟( $f_{IH}$ )	X1振荡时钟( $f_X$ )	PLL( $f_{PLL}$ )	低速内部振荡器时钟( $f_{IL}$ )
高速内部振荡器时钟( $f_{IH}$ )		2.5个 $f_{IH} / f_X$ 时钟	2.5个 $f_{IH} / f_{PLL}$ 时钟	2.5个 $f_{IH} / f_{IL}$ 时钟
X1振荡时钟( $f_X$ )	2个时钟		2.5个 $f_X / f_{PLL}$ 时钟	2.5个 $f_X / f_{IL}$ 时钟
PLL( $f_{PLL}$ )	2个时钟	2.5个 $f_{PLL} / f_X$ 时钟		2.5个 $f_{PLL} / f_{IL}$ 时钟
低速内部振荡器时钟( $f_{IL}$ )	2个时钟	2个时钟	2个时钟	

注1: 表4-5中的时钟数是切换前的CPU时钟数。

注2: 表4-5中的时钟数是舍入小数部分的时钟数。

例: CPU从主系统时钟切换到副系统时钟的情况 (选择 $f_{MAIN}=f_{IH}=2\text{MHz}$ 、 $f_{SUB}=f_{IL}=15\text{KHz}$ 振荡的情况)

$$2.5f_{MAIN}/f_{SUB}=2.5(2000/15)=333.3\approx 334\text{个时钟}$$

## 4.7.6 时钟振荡停止前的条件

用于停止时钟振荡的寄存器标志设置和停止前的条件如下所示。

表4-6 时钟振荡停止前的条件和标志设置

时钟	时钟停止前的条件	SFR寄存器的标志
高速内部振荡器时钟	CLS=1 (CPU以高速内部振荡器时钟或PLL时钟以外的时钟运行)	HIOSTOP=1
X1振荡时钟	MCS=0或者CLS=1; 当以PLL时钟运行时PLLCKSEL=0 (CPU以高速系统时钟以外的时钟运行, 且当以PLL时钟运行时, 时钟源不为X1振荡时钟)	MSTOP=1
PLL时钟	MCS=0或者CLS=1或PLLCKSEL=0 (CPU以PLL时钟以外的时钟运行)	PLLON=0
低速内部振荡器时钟	CLS=0, WDTON=0, WUMMCK0=0 (CPU以低速内部振荡器时钟以外的时钟运行)	SELLOCO=0

# 第5章 通用定时器单元的功能

## 5.1 通用定时器的功能

通用定时器单元有以下功能：

### 5.1.1 独立通道运行功能

独立通道运行功能是不受其他通道运行模式的影响而能独立使用任意通道的功能。

#### (1) 间隔定时器

能用作以固定间隔产生中断（INTTmn）的基准定时器。



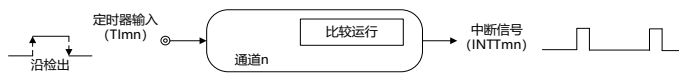
#### (2) 方波输出

每当产生INTTmn 中断时，就进行交替运行并且从定时器的输出引脚（TOmn）输出50% 占空比的方波。



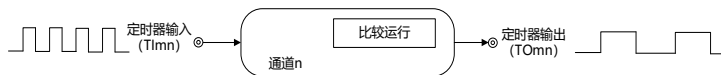
#### (3) 外部事件计数器

对定时器输入引脚（TImn）的输入信号的有效边沿进行计数，如果达到规定次数，就能用作产生中断的事件计数器。



#### (4) 分频器功能（只限于单元 0 的通道 0）

对定时器输入引脚（TI00）的输入时钟进行分频，然后从输出引脚（TO00）输出。



#### (5) 输入脉冲间隔的测量

在定时器输入引脚（TImn）的输入脉冲信号的有效边沿开始计数并且在下一个脉冲的有效边沿捕捉计数值，从而测量输入脉冲的间隔。



#### (6) 输入信号的高低电平宽度的测量

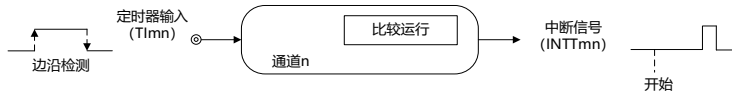
在定时器输入引脚（TImn）的输入信号的一个边沿开始计数并且在另一个边沿捕捉计数值，从而测量输入

信号的高低电平宽度。



(7) 延迟计数器

在定时器输入引脚 (TIMn) 的输入信号的有效边沿开始计数并且在经过任意延迟期间后产生中断。



注1: m: 单元号 (m=0) n: 通道号 (n=0~3)

注2: 通道0~3的定时器输入/输出引脚可配置请参照“第2章 引脚功能”。

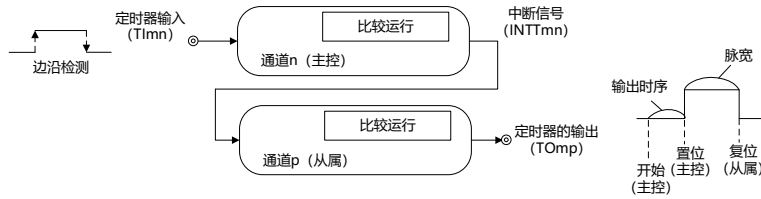
### 5.1.2 多通道联动运行功能

多通道联动运行功能是将主控通道（主要控制周期的基准定时器）和从属通道（遵从主控通道运行的定时器）组合实现的功能。

多通道联动运行功能可用作以下模式。

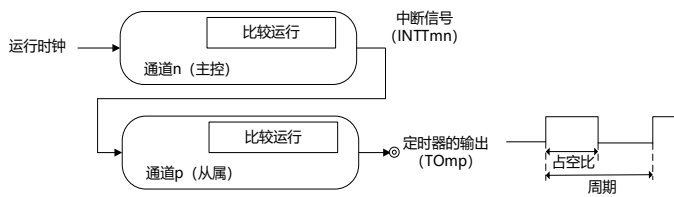
(1) 单触发脉冲输出

将2个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



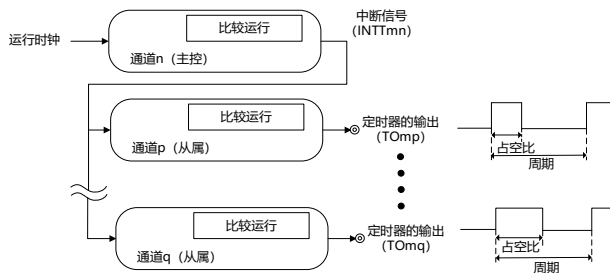
(2) PWM (Pulse Width Modulation) 输出

将2个通道成对使用，生成能任意设定周期和占空比的脉冲。



(3) 多重 PWM (Pulse Width Modulation) 输出

能通过扩展 PWM 功能并且使用 1 个主控通道和多个从属通道，以固定周期生成最多 3 种任意占空比的 PWM 信号。



注1：有关多通道联动运行功能规则的详细内容，请参照“5.3.1 多通道联动运行功能的基本规则”。

注2：m：单元号 (m=0) n：通道号 (n=0~3) p、q：从属通道号 (n < p < q ≤ 3)。

### 5.1.3 8 位定时器运行功能（只限于通道1和通道3）

8 位定时器运行功能是将 16 位定时器通道用作 2 个 8 位定时器通道的功能。只能使用通道 1 和通道 3。

注意：在使用 8 位定时器运行功能时，有几个规则。

详细内容请参照“5.3.2 8 位定时器运行功能的基本规则（只限于通道 1 和通道 3）”。

## 5.2 通用定时器单元的结构

通用定时器单元由以下硬件构成。

表5-1 通用定时器单元的结构

项目	结构
计数器	定时器计数寄存器mn (TCRmn)
寄存器	定时器数据寄存器mn (TDRmn)
定时器的输入	TI00~TI03 <sup>注1</sup>
定时器的输出	TO00~TO03 <sup>注1</sup> 、输出控制电路
控制寄存器	<单元设定部的寄存器> <ul style="list-style-type: none"> <li>• 外围允许寄存器0 (PER0)</li> <li>• 定时器时钟选择寄存器m (TPSm)</li> <li>• 定时器通道允许状态寄存器m (TEm)</li> <li>• 定时器通道开始寄存器m (TSM)</li> <li>• 定时器通道停止寄存器m (TTm)</li> <li>• 定时器输入选择寄存器0 (TIOS0) <sup>注2</sup></li> <li>• 定时器输出允许寄存器m (TOEm)</li> <li>• 定时器输出寄存器m (TOM)</li> <li>• 定时器输出电平寄存器m (TOLm)</li> <li>• 定时器输出模式寄存器m (TOMm)</li> </ul>
	<每个通道的寄存器> <ul style="list-style-type: none"> <li>• 定时器模式寄存器mn (TMRmn)</li> <li>• 定时器状态寄存器mn (TSRmn)</li> <li>• 端口模式控制寄存器 (PMCxx) <sup>注3</sup></li> <li>• 端口模式寄存器 (PMxx) <sup>注3</sup></li> <li>• 端口输出复用功能配置寄存器 (PxxCFG) <sup>注3</sup></li> <li>• 端口输入复用功能配置寄存器 (PStau0tinnCFG) <sup>注3</sup></li> </ul>

注1：通用定时器单元0的输入/输出引脚复用 to 固定端口。详细内容请参照“第2章 引脚功能”。

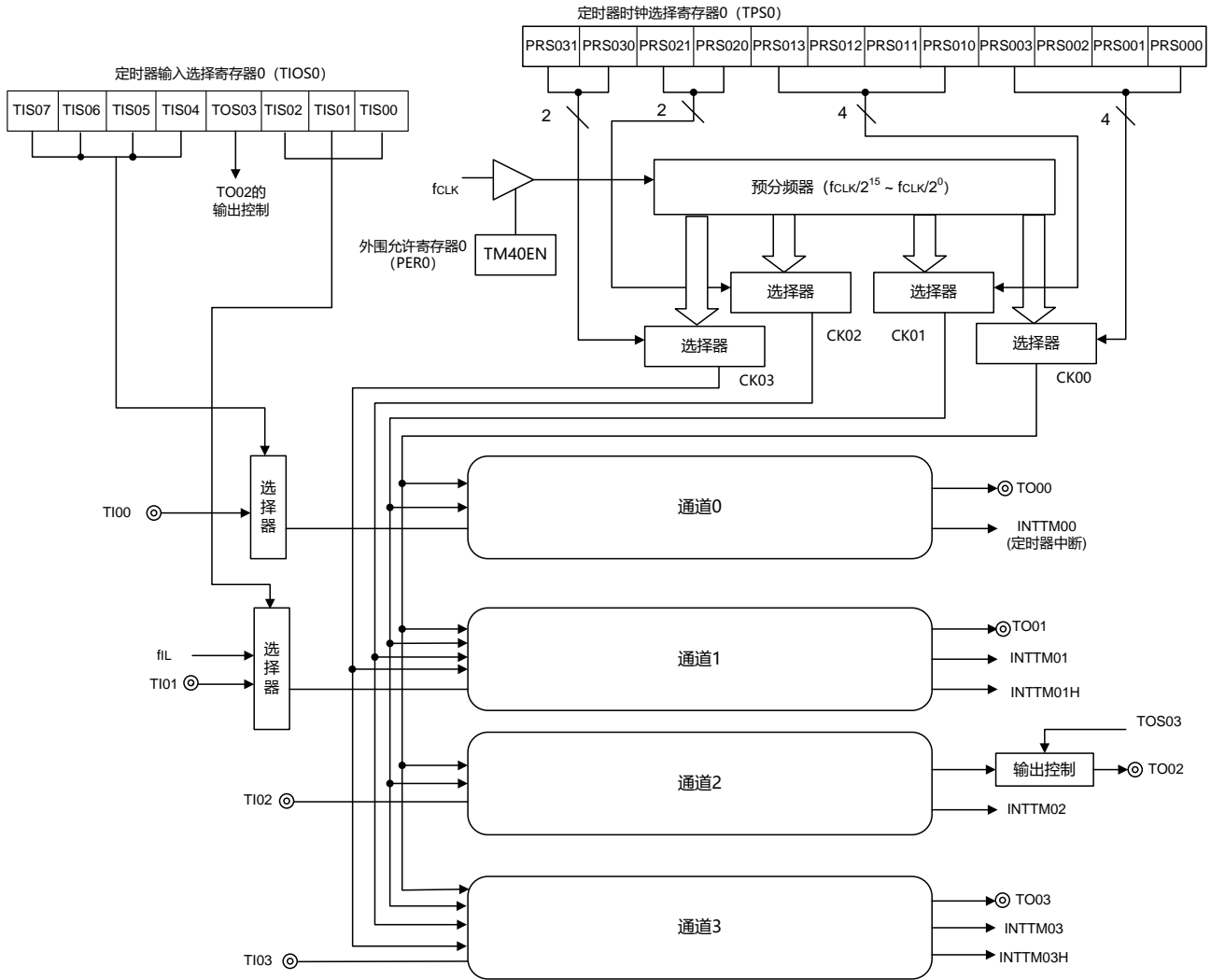
注2：仅用于单元0的通道选择。

注3：用于通道0~3的定时器输入/输出引脚配置。详细内容请参照“第2章 引脚功能”。

注4：m：单元号 (m=0) n：通道号 (n=0~3)。

通用定时器单元的框图如图 5-1 所示。

图5-1 通用定时器单元0的整体框图





## 5.2.1 寄存器映射

(以下寄存器基地址 = 0x4004\_1D80) RO: 只读, WO: 只写, R/W: 读写

寄存器名	偏移地址	读写属性	位宽	描述	复位值
TCR00	0x000	R	16	定时器通道 0 计数寄存器	0xFFFF
TCR01	0x002	R	16	定时器通道 1 计数寄存器	0xFFFF
TCR02	0x004	R	16	定时器通道 2 计数寄存器	0xFFFF
TCR03	0x006	R	16	定时器通道 3 计数寄存器	0xFFFF
TMR00	0x010	R/W	16	定时器通道 0 模式寄存器	0x0000
TMR01	0x012	R/W	16	定时器通道 1 模式寄存器	0x0000
TMR02	0x014	R/W	16	定时器通道 2 模式寄存器	0x0000
TMR03	0x016	R/W	16	定时器通道 3 模式寄存器	0x0000
TSR00	0x020	R	16	定时器通道 0 状态寄存器	0x0000
TSR01	0x022	R	16	定时器通道 1 状态寄存器	0x0000
TSR02	0x024	R	16	定时器通道 2 状态寄存器	0x0000
TSR03	0x026	R	16	定时器通道 3 状态寄存器	0x0000
TE0	0x030	R	16	定时器通道允许状态寄存器	0x0000
TS0	0x032	R/W	16	定时器通道开始寄存器	0x0000
TT0	0x034	R/W	16	定时器通道停止寄存器	0x0000
TPS0	0x036	R/W	16	定时器时钟选择寄存器	0x0000
TO0	0x038	R/W	16	定时器输出寄存器	0x0000
TOE0	0x03A	R/W	16	定时器输出允许寄存器	0x0000
TOL0	0x03C	R/W	16	定时器输出电平寄存器	0x0000
TOM0	0x03E	R/W	16	定时器输出模式寄存器	0x0000
TOM0L	0x03E	R/W	8	定时器输出模式寄存器低 8 位	0x00
TDR00	0x198	R/W	16	定时器通道 0 数据寄存器	0x0000
TDR01	0x19A	R/W	16	定时器通道 1 数据寄存器	0x0000
TDR02	0x1E4	R/W	16	定时器通道 2 数据寄存器	0x0000
TDR03	0x1E6	R/W	16	定时器通道 3 数据寄存器	0x0000

(以下寄存器基地址 = 0x4004\_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
TIOS0	0x004	R/W	定时器输入输出选择寄存器	0x0

## 5.2.2 定时器计数寄存器mn (TCRmn)

TCRmn寄存器是对计数时钟进行计数的16位只读寄存器。与计数时钟的上升沿同步进行递增或者递减计数。

通过定时器模式寄存器mn (TMRmn) 的MDmn3~MDmn0位来选择运行模式，进行递增和递减计数的切换（参照“5.2.6定时器模式寄存器mn (TMRmn)”）。

表5-2 定时器计数寄存器 mn (TCRmn) 的表格

位	符号	描述	复位值
15:0	TCRmn	计数时钟计数寄存器(只读)	0xFFFF

注：m：单元号 (m=0) n：通道号 (n=0~3)

能通过读定时器计数寄存器mn (TCRmn) 来读计数值。

在以下情况下，计数值变为“FFFFH”。

- 当产生复位信号时
- 当清除外围允许寄存器0 (PER0) 的TM40EN位时
- 在PWM输出模式中从属通道的计数结束时
- 在延迟计数模式中从属通道的计数结束时
- 在单触发脉冲输出模式中主控/从属通道的计数结束时
- 在多重PWM输出模式中从属通道的计数结束时

在以下情况下，计数值变为“0000H”。

- 在捕捉模式中输入开始触发时
- 在捕捉模式中捕捉结束时

注:即使读TCRmn寄存器，也不将计数值捕捉到定时器数据寄存器 mn (TDRmn)。

如下所示，TCRmn寄存器的读取值因运行模式和运行状态而不同。

表5-3 各运行模式中的定时器计数寄存器mn (TCRmn) 的读取值

运行模式	计数方式	定时器计数寄存器mn (TCRmn) 的读取值 <sup>注</sup>			
		解除复位后更改运行模式时的值	计数暂停 (TTmn=1) 时的值	计数暂停 (TTmn=1) 后更改运行模式时的值	单次计数后等待开始触发时的值
间隔定时器模式	递减计数	0xFFFF	停止时的值	不定值	—
捕捉模式	递增计数	0x0000	停止时的值	不定值	—
事件计数器模式	递减计数	0xFFFF	停止时的值	不定值	—
单次计数模式	递减计数	0xFFFF	停止时的值	不定值	0xFFFF
捕捉&单次计数模式	递增计数	0x0000	停止时的值	不定值	TDRmn寄存器的捕捉值+1

注1：m：单元号 (m=0) n：通道号 (n=0~3)

注2：表示通道n处于定时器运行停止状态 (TEmn=0) 和计数允许状态 (TSmn=1) 时的TCRmn寄存器的读取值。将此值保持在TCRmn寄存器，直到开始计数为止。

### 5.2.3 定时器数据寄存器mn (TDRmn)

这是能进行捕捉功能和比较功能切换使用的 16 位寄存器。通过定时器模式寄存器 mn (TMRmn) 的 MDmn3~MDmn0 位来选择运行模式，进行捕捉功能和比较功能的切换。

能随时改写 TDRmn 寄存器。

能以 16 位为单位读写此寄存器。

在 8 位定时器模式中 (定时器模式寄存器 m1、m3 (TMRm1、TMRm3) 的 SPLIT 位为“1”), 能以 8 位为单位读写 TDRm1 寄存器和 TDRm3 寄存器, 其中 TDRm1H 和 TDRm3H 用作高 8 位, TDRm1L 和 TDRm3L 用作低 8 位。

在产生复位信号后, TDRmn 寄存器的值变为“0000H”。

表5-4 通道0定时器数据寄存器TDR00

位	符号	描述	复位值
15:0	TDR00	定时器通道0数据寄存器	0x0000

表5-5 通道1定时器数据寄存器TDR01

位	符号	描述	复位值
15:8	TDR01H	定时器通道1数据寄存器bit15:8	0x00
7:0	TDR01L	定时器通道1数据寄存器bit7:0	0x00

表5-6 通道2定时器数据寄存器TDR02

位	符号	描述	复位值
15:0	TDR02	定时器通道2数据寄存器	0x00

表5-7 通道3定时器数据寄存器TDR03

位	符号	描述	复位值
15:8	TDR03H	定时器通道3数据寄存器bit15:8	0x00
7:0	TDR03L	定时器通道3数据寄存器bit7:0	0x00

(i) 定时器数据寄存器mn (TDRmn) 用作比较寄存器的情况

从TDRmn寄存器的设定值开始递减计数, 当计数值变为“0000H”时, 产生中断信号 (INTTMmn)。保持TDRmn寄存器的值, 直到被改写为止。

注: 即使输入捕捉触发信号, 设定为比较功能的TDRmn寄存器也不进行捕捉运行。

(ii) 定时器数据寄存器mn (TDRmn) 用作捕捉寄存器的情况

通过输入捕捉触发, 将定时器计数寄存器mn (TCRmn) 的计数值捕捉到TDRmn寄存器。

能选择TIMn引脚的有效边沿作为捕捉触发信号。通过定时器模式寄存器mn (TMRmn) 来设定捕捉触发的选择。

注: m: 单元号 (m=0) n: 通道号 (n=0~3)。

## 5.2.4 外围允许寄存器0 (PER0)

PER0 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用通用定时器单元 0 时，必须将 bit0 (TM40EN) 置“1”。通过 8 位存储器操作指令设定 PER0 寄存器。在产生复位信号后，PER0 寄存器的值变为“00H”。

表5-8 外围允许寄存器0 (PER0) 的表格

位	符号	描述	复位值
7	LSITIMEREN	提供LSITIMER输入时钟的控制(掉电睡眠可行) 0: 停止提供输入时钟，不能写LSITIMER使用的SFR 1: 提供输入时钟，能写LSITIMER使用的SFR	0
6:5	--	保留	0x0
4	IICAEN	提供IICA模块输入时钟的控制 0: 停止提供输入时钟，不能写IICA使用的SFR 1: 提供输入时钟，能写IICA使用的SFR	0
3:1	--	保留	0x0
0	TM40EN	提供通用定时器单元0输入时钟的控制 0: 停止提供输入时钟，不能写通用定时器单元0使用的SFR 1: 提供输入时钟，能写通用定时器单元0使用的SFR	0

注：要设定通用定时器单元时，必须先在TM40EN位为“1”的状态下设定以下的寄存器。当TM40EN位为“0”时，定时器阵列单元的控制寄存器的值为初始值，忽视写操作（定时器输入输出选择寄存器0 (TIOS0)、端口模式控制寄存器PMCx、端口模式寄存器PMx和端口复用功能配置寄存器PxxCFG除外）。

- 定时器状态寄存器mn (TSRmn)
- 定时器通道允许状态寄存器m (TEm)
- 定时器通道开始寄存器m (TSm)
- 定时器通道停止寄存器m (TTm)
- 定时器输出允许寄存器m (TOEm)
- 定时器输出寄存器m (TOM)
- 定时器输出电平寄存器m (TOLm)
- 定时器输出模式寄存器m (TOMm)

## 5.2.5 定时器时钟选择寄存器m (TPSm)

TPSm 寄存器是 16 位寄存器，选择提供给各通道的 2 种或者 4 种公共运行时钟 (CKm0、CKm1、CKm2、CKm3)。通过 TPSm 寄存器的 bit3~0 选择 CKm0，通过 TPSm 寄存器的 bit7~4 选择 CKm1。另外，只有通道 1 和通道 3 才能选择 CKm2 和 CKm3，通过 TPSm 寄存器的 bit9~8 选择 CKm2，通过 TPSm 寄存器的 bit13 和 bit12 选择 CKm3。

只有在以下情况下才能改写定时器运行中的 TPSm 寄存器：

能改写 PRSm00~PRSm03 位的情况 (n=0~3)：

选择 CKm0 作为运行时钟 (CKSmn1、CKSmn0=0、0) 的通道全部处于停止状态 (TEmn=0)。

能改写 PRSm10~PRSm13 位的情况 (n=0~3)：

选择 CKm2 作为运行时钟 (CKSmn1、CKSmn0=0、1) 的通道全部处于停止状态 (TEmn=0)。

能改写 PRSm20 位和 PRSm21 位的情况 (n=1、3)：

选择 CKm1 作为运行时钟 (CKSmn1、CKSmn0=1、0) 的通道全部处于停止状态 (TEmn=0)。

能改写 PRSm30 位和 PRSm31 位的情况 (n=1、3)：

选择 CKm3 作为运行时钟 (CKSmn1、CKSmn0=1、1) 的通道全部处于停止状态 (TEmn=0)。

通过 16 位存储器操作指令设定 TPSm 寄存器。在产生复位信号后，TPSm 寄存器的值变为“0000H”。

表5-9 定时器时钟选择寄存器m (TPSm) 的表格

位	符号	描述	复位值
15:14	-	必须为0	0x0
13:12	CKm3	定时器运行时钟选择CKm3: 00: $f_{clk}/2^8$ 01: $f_{clk}/2^{10}$ 10: $f_{clk}/2^{12}$ 11: $f_{clk}/2^{14}$	0x0
11:10	-	必须为0	0x0
9:8	CKm2	定时器运行时钟选择CKm2: 00: $f_{clk}/2$ 01: $f_{clk}/2^2$ 10: $f_{clk}/2^4$ 11: $f_{clk}/2^6$	0x0
7:4	CKm1	定时器运行时钟选择 $f_{clk}/2^{CKm1}$	0x0
3:0	CKm0	定时器运行时钟选择 $f_{clk}/2^{CKm0}$	0x0

注1:  $f_{CLK}$ : CPU/外围硬件的时钟频率。

注2: TPSm寄存器选择的时钟波形从上升沿开始只有1个 $F_{CLK}$ 周期为高电平。详细内容请参照“5.4.1 计数时钟 ( $f_{TCLK}$ )”。

注3: 必须将bit15、14、11、10置“0”。

注4: 在更改选择为 $f_{CLK}$ 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下，必须停止通用定时器单元 (TTm=000FH)。即使在选择运行时钟 ( $f_{MCK}$ ) 或者TImn引脚输入信号的有效边沿时，也需要停止通用定时器单元。

注5: 如果选择 $f_{CLK}$  (无分频) 作为运行时钟 (CKmk) 并且将TDRmn置“0000H” (m=0、1, n=0~3)，就不能使用通用定时器单元的中断请求。

如果在 8 位定时器模式中使用通道 1 和通道 3 并且将 CKm2 和 CKm3 作为运行时钟，就能通过间隔定时器

功能实现下表所示的间隔时间。

表5-10 运行时钟CKSm2和CKSm3能设定的间隔时间

时钟		间隔时间注 ( $f_{CLK}=32\text{MHz}$ )			
		10us	100us	1ms	10ms
CKm2	$f_{CLK}/2$	○	—	—	—
	$f_{CLK}/2^2$	○	—	—	—
	$f_{CLK}/2^4$	○	○	—	—
	$f_{CLK}/2^6$	○	○	—	—
CKm3	$f_{CLK}/2^8$	—	○	○	—
	$f_{CLK}/2^{10}$	—	○	○	—
	$f_{CLK}/2^{12}$	—	—	○	○
	$f_{CLK}/2^{14}$	—	—	○	○

注1：包含5%以内的误差。

注2： $f_{CLK}$ ：CPU/外围硬件的时钟频率。

注3：有关TPSm寄存器所选 $f_{CLK}/2^r$ 波形的详细内容，请参照“5.4.1 计数时钟 ( $f_{TCLK}$ )”。

## 5.2.6 定时器模式寄存器mn (TMRmn)

TMRmn 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟 ( $f_{MCK}$ ) 的选择、计数时钟的选择、主控/从属的选择、16 位/8 位定时器的选择（只限于通道 1 和通道 3）、开始触发和捕捉触发的设定、定时器输入有效边沿的选择以及运行模式（间隔、捕捉、事件计数器、单次计数、捕捉&单次计数）的设定。

禁止在运行中 (TEmn=1) 改写 TMRmn 寄存器。但是，能在一部分的功能运行中 (TEmn=1) 改写 bit7 和 bit6 (CISmn1、CISmn0)（详细内容请参照“5.7 通用定时器单元的独立通道运行功能”和“5.8 定时器阵列单元的多通道联动运行功能”）。

通过 16 位存储器操作指令设定 TMRmn 寄存器。在产生复位信号后，TMRmn 寄存器的值变为“0000H”。

注：TMRmn寄存器的bit11因通道而不同。

TMRm2: MASTERmn位 (n=2)

TMRm1、TMRm3: SPLITmn位 (n=1、3)

TMRm0: 固定为“0”。

表5-11 定时器通道0模式寄存器TMR00

位	符号	描述	复位值
15:14	CKS001- CKS000	通道n运行时钟 ( $f_{MCK}$ ) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 ( $f_{MCK}$ ) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 ( $f_{TCLK}$ )。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	-	必须为0	0
12	CCS00	通道n计数时钟 ( $f_{TCLK}$ ) 的选择 0: CKS000位和CKS001位指定的运行时钟 ( $f_{MCK}$ ) 1: TIOS0选择的输入信号的有效边沿 计数时钟 ( $f_{TCLK}$ ) 用于计数器、输出控制电路和中断控制电路	0
11	-	必须为0	0
10:8	STS002- STS000	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效（不选择其他触发源）。 001: 将TI00引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TI00引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 使用主控通道的中断信号（多通道联动运行功能的从属通道的情况） 其他 禁止设定	0x0
7:6	CIS001- CIS000	TI00引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿（测量低电平宽度时） 开始触发：下降沿，捕捉触发：上升沿	0x0



		11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	
5:4	0	保留	0x0
3:0	MD003- MD000	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。 1001: 单次计数模式, 计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式, 在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	0x0

MD003- MD000 详细说明见下表:

MD003	MD002	MD001	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出 (主控)	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出 (从属)	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MD000位而变 (参照下表)。					

运行模式 (MD003~MD001位的设定 (参照上表))	MD000	开始计数和中断的设定
<ul style="list-style-type: none"> <li>间隔定时器模式 (0、0、0)</li> <li>捕捉模式 (0、1、0)</li> </ul>	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断 (定时器的输出也发生变化)。
<ul style="list-style-type: none"> <li>事件计数器模式 (0、1、1)</li> </ul>	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
<ul style="list-style-type: none"> <li>单次计数模式<sup>注1</sup> (1、0、0)</li> </ul>	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效 <sup>注2</sup> 。此时不产生中断。
<ul style="list-style-type: none"> <li>捕捉&amp;单次计数模式 (1、1、0)</li> </ul>	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。计数运行中的开始触发无效。此时不产生中断。

注1: 在单次计数模式中, 不控制开始计数时的中断输出 (INTTM00) 和TO00输出。

注2: 如果在运行中产生开始触发 (TS00=1), 就对计数器进行初始化并且重新开始计数 (不产生中断请求)。



表 5-12定时器通道1模式寄存器TMR01

位	符号	描述	复位值
15:14	CKS011- CKS010	通道n运行时钟 ( $f_{MCK}$ ) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 ( $f_{MCK}$ ) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 ( $f_{TCLK}$ )。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	-	必须为0	0
12	CCS01	通道n计数时钟 ( $f_{TCLK}$ ) 的选择 0: CKS010位和CKS011位指定的运行时钟 ( $f_{MCK}$ ) 1: TIOS0选择的输入信号的有效边沿 计数时钟 ( $f_{TCLK}$ ) 用于计数器、输出控制电路和中断控制电路	0
11	SPLIT00	通道1的8位定时器/16位定时器的运行选择 0: 用作16位定时器。(用作独立通道功能或则多通道联动运行功能的从属通道) 1: 用作8位定时器。	0
10:8	STS012- STS010	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效 (不选择其他触发源)。 001: 将TI01引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TI01引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 使用主控通道的中断信号 (多通道联动运行功能的从属通道的情况) 其他 禁止设定	0x0
7:6	CIS011- CIS010	TI01引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿 11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	0x0
5:4	-	保留	0x0
3:0	MD013- MD010	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。 1001: 单次计数模式, 计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式, 在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	0x0

MD013- MD010 详细说明见下表

MD013	MD012	MD011	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出（主控）	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出（从属）	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MD010位而变（参照下表）。					

运行模式（MD013~MD011位的设定（参照上表））	MD010	开始计数和中断的设定
<ul style="list-style-type: none"> <li>间隔定时器模式（0、0、0）</li> <li>捕捉模式（0、1、0）</li> </ul>	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
	1	在开始计数时产生定时器中断（定时器的输出也发生变化）。
<ul style="list-style-type: none"> <li>事件计数器模式（0、1、1）</li> </ul>	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
<ul style="list-style-type: none"> <li>单次计数模式注1（1、0、0）</li> </ul>	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效注2。此时不产生中断。
<ul style="list-style-type: none"> <li>捕捉&amp;单次计数模式（1、1、0）</li> </ul>	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。计数运行中的开始触发无效。此时不产生中断。

注1：在单次计数模式中，不控制开始计数时的中断输出（INTTM01）和TO01输出。

注2：如果在运行中产生开始触发（TS01=1），就对计数器进行初始化并且重新开始计数（不产生中断请求）。

图5-13定时器通道2模式寄存器TMR02

位	符号	描述	复位值
15:14	CKS021- CKS020	通道n运行时钟 ( $f_{MCK}$ ) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 ( $f_{MCK}$ ) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 ( $f_{TCLK}$ )。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	-	必须为0	0
12	CCS02	通道n计数时钟 ( $f_{TCLK}$ ) 的选择 0: CKS020位和CKS021位指定的运行时钟 ( $f_{MCK}$ ) 1: TI02输入信号的有效边沿 计数时钟 ( $f_{TCLK}$ ) 用于计数器、输出控制电路和中断控制电路	
11	MASTER	通道2的独立通道运行/多通道联动运行 (从属或者主控) 的选择 0: 用作独立通道运行功能或则多通道联动运行功能的从属通道。 通道0固定为“0” (因为通道0为最高位的通道, 所以与此位的设定无关, 用作主控通道) 1: 用作多通道联动运行功能的主控通道。 通道0固定为“0” (因为通道0为最高位的通道, 所以与此位的设定无关, 用作主控通道) 只能将通道2设为主控通道 (MASTERmn=1)	0
10:8	STS022- STS020	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效 (不选择其他触发源)。 001: 将TI02引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TI02引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 使用主控通道的中断信号 (多通道联动运行功能的从属通道的情况) 其他: 禁止设定	0x0
7:6	CIS021- CIS020	TI02引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿 11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	0x0
5:4	-	保留	0x0
3:0	MD023- MD020	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。	0x0

		1001: 单次计数模式, 计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式, 在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	
--	--	---	--

## MD023- MD020 详细说明见下表

MD023	MD022	MD021	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出(主控)	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出(从属)	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MD020位而变(参照下表)。					

运行模式(MD023~MD021位的设定(参照上表))	MD020	开始计数和中断的设定
<ul style="list-style-type: none"> <li>间隔定时器模式(0、0、0)</li> <li>捕捉模式(0、1、0)</li> </ul>	0	在开始计数时不产生定时器中断(定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断(定时器的输出也发生变化)。
<ul style="list-style-type: none"> <li>事件计数器模式(0、1、1)</li> </ul>	0	在开始计数时不产生定时器中断(定时器的输出也不发生变化)。
<ul style="list-style-type: none"> <li>单次计数模式注1(1、0、0)</li> </ul>	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效注2。此时不产生中断。
<ul style="list-style-type: none"> <li>捕捉&amp;单次计数模式(1、1、0)</li> </ul>	0	在开始计数时不产生定时器中断(定时器的输出也不发生变化)。计数运行中的开始触发无效。

注1: 在单次计数模式中, 不控制开始计数时的中断输出(INTTM02)和TO02输出。

注2: 如果在运行中产生开始触发(TS02=1), 就对计数器进行初始化并且重新开始计数(不产生中断请求)。

表5-14 定时器通道3模式寄存器TMR03

位	符号	描述	复位值
15:14	CKS031- CKS030	通道n运行时钟 ( $f_{MCK}$ ) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 ( $f_{MCK}$ ) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 ( $f_{TCLK}$ )。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	-	必须为0	0
12	CCS03	通道n计数时钟 ( $f_{TCLK}$ ) 的选择 0: CKS030位和CKS031位指定的运行时钟 ( $f_{MCK}$ ) 1: TI03选择的输入信号的有效边沿 计数时钟 ( $f_{TCLK}$ ) 用于计数器、输出控制电路和中断控制电路	0
11	SPLIT03	通道1的8位定时器/16位定时器的运行选择 0: 用作16位定时器。(用作独立通道功能或则多通道联动运行功能的从属通道) 1: 用作8位定时器。	0
10:8	STS032- STS030	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效 (不选择其他触发源)。 001: 将TI03引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TI03引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 使用主控通道的中断信号 (多通道联动运行功能的从属通道的情况) 其他 禁止设定	0x0
7:6	CIS031- CIS030	TI03引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿 11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	0x0
5:4	-	保留	0x0
3:0	MD033- MD030	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。 1001: 单次计数模式, 计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式, 在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	0x0

MD033- MD030 详细说明见下表

MD033	MD322	MD031	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出（主控）	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出（从属）	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MD030位而变（参照下表）。					

运行模式（MD023~MD021位的设定（参照上表））	MD030	开始计数和中断的设定
<ul style="list-style-type: none"> <li>间隔定时器模式（0、0、0）</li> <li>捕捉模式（0、1、0）</li> </ul>	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
	1	在开始计数时产生定时器中断（定时器的输出也发生变化）。
<ul style="list-style-type: none"> <li>事件计数器模式（0、1、1）</li> </ul>	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
<ul style="list-style-type: none"> <li>单次计数模式注<sup>1</sup>（1、0、0）</li> </ul>	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效注 <sup>2</sup> 。此时不产生中断。
<ul style="list-style-type: none"> <li>捕捉&amp;单次计数模式（1、1、0）</li> </ul>	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。计数运行中的开始触发无效。

注1：在单次计数模式中，不控制开始计数时的中断输出（INTTM03）和TO03输出。

注2：如果在运行中产生开始触发（TS03=1），就对计数器进行初始化并且重新开始计数（不产生中断请求）。

## 5.2.7 定时器状态寄存器mn (TSRmn)

TSRmn 寄存器是表示通道 n 计数器的上溢状态的寄存器。

TSRmn 寄存器只在捕捉模式 (MDmn3 ~ MDmn1=010B) 和捕捉 & 单次计数模式 (MDmn3 ~ MDmn1=110B) 中有效。有关各运行模式中的 OVF 位的变化和置位/清除条件, 请参照表 5-16。

通过 16 位存储器操作指令读 TSRmn 寄存器。

在产生复位信号后, TSRmn 寄存器的值变为“0000H”。

表5-15 定时器状态寄存器mn (TSRmn) 的表格

位	符号	描述	复位值
15:1	-	保留	0x0
0	OVF	通道n的计数器上溢状态、 0: 没有发生上溢。 1: 发生上溢。 如果OVF位为“1”, 就在下一次计数不发生上溢并且捕捉到计数值时清除此标志 (OVF=0)。	0

注: m: 单元号 (m=0) n: 通道号 (n=0~3)

表5-16 各运行模式中的OVF位的变化和置位/清除条件

定时器运行模式	OVF位	置位/清除条件
<ul style="list-style-type: none"> <li>• 捕捉模式</li> <li>• 捕捉&amp;单次计数模式</li> </ul>	清除	在捕捉时没有发生上溢的情况
	置位	在捕捉时发生上溢的情况
<ul style="list-style-type: none"> <li>• 间隔定时器模式</li> <li>• 事件计数器模式</li> <li>• 单次计数模式</li> </ul>	清除	— (不能使用)
	置位	

注: m: 单元号 (m=0) n: 通道号 (n=0~3)

## 5.2.8 定时器通道允许状态寄存器m (TEM)

TEM 寄存器是表示各通道定时器运行的允许或者停止状态的寄存器。

TEM 寄存器的各位对应定时器通道开始寄存器 m (TSM) 和定时器通道停止寄存器 m (TTM) 的各位。如果将 TSM 寄存器的各位置“1”，就将 TEM 寄存器的对应位置“1”。如果将 TTM 寄存器的各位置“1”，就将其对应位清“0”。

通过 16 位存储器操作指令读 TEM 寄存器。

在产生复位信号后，TEM 寄存器的值变为“0000H”。

表5-17 定时器通道允许状态寄存器m (TEM) 的表格

位	符号	描述	复位值
15:12	-	保留	0x0
11	TEH03	通道3为8位定时器模式时的高8位定时器的运行允许或者停止状态的表示 0: 运行停止状态 1: 运行允许状态	0
10	-	保留	0
9	TEH01	通道1为8位定时器模式时的高8位定时器的运行允许或者停止状态的表示 0: 运行停止状态 1: 运行允许状态	0
8:4	-	保留	0x0
3:0	TE03-TE00	通道n的运行允许或者停止状态的表示: 0: 运行停止状态 1: 运行允许状态	0x0



## 5.2.9 定时器通道开始寄存器m (TSm)

TSm 寄存器是对定时器计数寄存器 mn (TCRmn) 进行初始化并且设定各通道计数运行开始的触发寄存器。如果将各位置“1”，定时器通道允许状态寄存器 m (TEm) 的对应位就被置“1”。因为 TSmn 位、TSHm1 位和 TSHm3 位是触发位，所以如果变为运行允许状态 (TEmn、TEHm1、TEHm3=1)，就立即清除 TSmn 位、TSHm1 位和 TSHm3 位。

通过 16 位存储器操作指令设定 TSm 寄存器。

在产生复位信号后，TSm 寄存器的值变为“0000H”。

表5-18 定时器通道开始寄存器m (TSm) 的表格

位	符号	描述	复位值
15:12	-	保留	0x0
11	TSHm3	通道3为8位定时器模式时的高8位定时器的运行允许（开始）触发 0: 没有触发 1: 将TEHm3位置“1”，进入计数允许状态。如果在计数允许状态下开始TCRm3寄存器的计数，就进入间隔定时器模式（参照“5.4.2 计数器的开始时序”的表5-25）	0
10	-	保留	0
9	TSHm1	通道1为8位定时器模式时的高8位定时器的运行允许（开始）触发 0: 没有触发 1: 将TEHm1位置“1”，进入计数允许状态。如果在计数允许状态下开始TCRm1寄存器的计数，就进入间隔定时器模式（参照“5.4.2 计数器的开始时序”的表5-25）	0
8:4	-	保留	0x0
3:0	TSm3-TSm0	通道n的运行允许（开始）触发： 0: 没有触发 1: 将TEmn位置“1”，进入计数允许状态。计数允许状态下的TCRmn寄存器的计数开始因各运行模式而不同（参照“5.4.2 计数器的开始时序”的表5-25）。在通道1和通道3为8位定时器模式时，TSm1和TSm3为低8位定时器的运行允许（开始）触发。	0x0

注1：必须将bit15~12、10、8~4置“0”。

注2：在从不使用TImn引脚输入的功能切换到使用TImn引脚输入的功能时，从设定定时器模式寄存器mn (TMRmn) 到将 TSmn (TSHm1、TSHm3) 位置“1”为止，需要2个运行时钟 ( $f_{MCK}$ ) 的等待。

注3：TSm寄存器的读取值总是“0”。

注4：m：单元号 (m=0)

## 5.2.10 定时器通道停止寄存器m (TTm)

TTm 寄存器是设定各通道计数停止的触发寄存器。

如果将各位置“1”，定时器通道允许状态寄存器 m (TEm) 的对应位就被清“0”。因为 TTmn 位、TTHm1 位和 TTHm3 位是触发位，所以如果变为运行停止状态 (TEmn、TEHm1、TEHm3=0)，就立即清除 TTmn 位、TTHm1 位和 TTHm3 位。

通过 16 位存储器操作指令设定 TTm 寄存器。

在产生复位信号后，TTm 寄存器的值变为“0000H”。

表5-19 定时器通道停止寄存器m (TTm) 的表格

位	符号	描述	复位值
15:12	-	保留	0x0
11	TTHm3	通道3为8位定时器模式时的高8位定时器的运行停止触发 0: 没有触发 1: 将TEHm3位清“0”，进入计数停止状态	0
10	-	保留	0
9	TTHm1	通道1为8位定时器模式时的高8位定时器的运行停止触发 0: 没有触发 1: 将TEHm1位清“0”，进入计数停止状态。	0
8:4	-	保留	0x0
3:0	TTm3-TTm0	通道n的运行允许（开始）触发： 0: 没有触发 1: 将TEmn位清“0”，进入计数停止状态。 在通道1和通道3为8位定时器模式时，TTm1和TTm3为低8位定时器的运行停止触发。	0x0

注1：必须将bit15~12、10、8~4置“0”。

注2：TTm寄存器的读取值总是“0”。

注3：m：单元号 (m=0)。

## 5.2.11 定时器输入输出选择寄存器 (TIOS0)

TIOS0 寄存器用于对单元 0 的输入输出进行选择。选择单元 0 的通道 0 和通道 1 的定时器输入以及通道 2 的定时器输出。通过 8 位存储器操作指令设定 TIOS0 寄存器。在产生复位信号后，TIOS0 寄存器的值变为“00H”。

表5-20 定时器输入输出选择寄存器0 (TIOS0) 的表格

位	符号	描述	复位值
7:5	TIS07- TIS05	通道0使用的定时器输入的选择 0: 定时器输入引脚 (TI00) 的输入信号 1: 禁止设定	0x0
4	TIS04	通道0使用的定时器输入的选择 0: 通过TIS07~TIS05选择的输入信号 1: 禁止设定	0
3	TOS03	通道2的定时器输出的使能 0: 允许输出 1: 禁止输出 (输出固定为0)。	0
2:0	TIS02- TIS00	通道1使用的定时器输入的选择 000: 定时器输入引脚 (TI01) 的输入信号 010: 定时器输入引脚 (TI01) 的输入信号。 011: 定时器输入引脚 (TI01) 的输入信号 100: 低速内部振荡器时钟 ( $f_{IL}$ ) 其他: 禁止设定	0x0

注：选择的定时器输入高低电平宽度需要大于等于 $1/f_{MCK}+10ns$ 。因此在选择 $f_{IL}$ 作为 $f_{CLK}$ 时 (CKC寄存器的CSS=1)，不能将TIS02位置“1”。

## 5.2.12 定时器输出允许寄存器m (TOEm)

TOEm 寄存器是设定允许或者禁止各通道定时器输出的寄存器。

对于允许定时器输出的通道 n，无法通过软件改写后述的定时器输出寄存器 m (TOm) 的 TOmn 位的值，并且由计数运行的定时器输出功能反映的值从定时器的输出引脚 (TOmn) 输出。

通过 16 位存储器操作指令设定 TOEm 寄存器。

在产生复位信号后，TOEm 寄存器的值变为“0000H”。

表5-21 定时器输出允许寄存器m (TOEm) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:0	TOEmn	通道n的定时器输出的允许/禁止: 0: 禁止定时器输出。 定时器的运行不反映到TOmn位，固定输出。能写TOmn位，并且从TOmn引脚输出TOmn位设定的电平。 1: 允许定时器输出 定时器的运行反映到TOmn位，产生输出波形。忽视TOmn位的写操作	0x0

注1：必须将bit15~4置“0”。

注2：m：单元号 (m=0) n：通道号 (n=0~3)

### 5.2.13 定时器输出寄存器m (TOM)

TOM 寄存器是各通道定时器输出的缓冲寄存器。

此寄存器各位的值从各通道定时器的输出引脚 (TOMn) 输出。

只有在禁止定时器输出 (TOEmn=0) 时才能通过软件改写此寄存器的 TOMn 位。当允许定时器输出时 (TOEmn=1), 忽视通过软件的改写操作, 而只通过定时器的运行更改其值。

要将 TI00/TO00、TI01/TO01、TI02/TO02、TI03/TO03 引脚用作端口功能时, 必须将相应的 TOMn 位置“0”。

通过 16 位存储器操作指令设定 TOM 寄存器。

在产生复位信号后, TOM 寄存器的值变为“0000H”。

表5-22 定时器输出寄存器m (TOM) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:0	TOMn	通道n的定时器输出: 0: 定时器的输出值为“0”。 1: 定时器的输出值为“1”	0x0

注1: m: 单元号 (m=0) n: 通道号 (n=0~3)

## 5.2.14 定时器输出电平寄存器m (TOLm)

TOLm 寄存器是控制各通道定时器输出电平的寄存器。

当允许定时器输出 (TOEmn=1) 并且使用多通道联动运行功能 (TOMmn=1) 时, 在定时器输出信号的置位和复位时序, 反映此寄存器进行的各通道 n 的反相设定。在主导通道输出模式 (TOMmn=0) 中, 此寄存器的设定无效。

通过 16 位存储器操作指令设定 TOLm 寄存器。

在产生复位信号后, TOLm 寄存器的值变为“0000H”。

表5-23 定时器输出电平寄存器m (TOLm) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:1	TOL03- TOL01	通道n的定时器输出电平的控制: 0: 正逻辑输出 (高电平有效) 1: 反相输出 (低电平有效)	0x0
0	-	保留为0	0

注1: 如果在定时器运行中改写此寄存器的值, 会在下一次定时器输出信号发生变化时反相定时器的输出逻辑, 而不是在改写后立即反相。

注2: m: 单元号 (m=0) n: 通道号 (n=0~3)。

## 5.2.15 定时器输出模式寄存器m (TOMm)

TOMm 寄存器是控制各通道定时器输出模式的寄存器。当用作独立通道运行功能时，将所用通道的对应位置“0”。

当用作多通道联动运行功能（PWM 输出、单触发脉冲输出和多重 PWM 输出）时，将主控通道的对应位置“0”并且将从属通道的对应位置“1”。

当允许定时器输出（TOEmn=1）时，在定时器输出信号的置位和复位时序，反映此寄存器进行的各通道 n 的设定。

通过 16 位存储器操作指令设定 TOMm 寄存器。

在产生复位信号后，TOMm 寄存器的值变为“0000H”。

表5-24 定时器输出模式寄存器m (TOMm) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:1	TOL03- TOL01	通道n的定时器输出模式的控制： 0: 主控通道输出模式（通过定时器中断请求信号（INTTMmn）进行交替输出） 1: 从属通道输出模式（通过主控通道的定时器中断请求信号（INTTMmn）将输出置位，并且通过从属通道的定时器中断请求信号（INTTMmp）对输出进行复位）	0x0
0	-	保留为0	0

注1：m：单元号（m=0） n：通道号 n=0~3（主控通道时：n=0、2）

p：从属通道号

n=0：p=1、2、3

n=2：p=3。

注2：（有关主控通道和从属通道关系的详细内容，请参照“5.3.1 多通道联动运行功能的基本规则”）。

## 5.2.16 控制定时器输入/输出引脚端口功能的寄存器

在使用通用定时器单元时，定时器 0 的输出引脚复用到固定端口，定时器 0 的输入引脚可以配置到任意端口。详细内容请参照“第 2 章 引脚功能”。

在将定时器 0 的输出引脚复用到某端口时，必须将该端口对应的端口模式控制寄存器（PMCxx）的位、端口模式寄存器（PMxx）的位置“0”。并设置端口复用功能配置寄存器（PxxCFG）。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

（例）将 P21 配置为 TO00 用作定时器输出的情况

将端口模式控制寄存器 2 的 PMC21 位置“0”。

将端口模式寄存器 2 的 PM21 位置“0”。

将端口输出复用功能配置寄存器 P21CFG 置“0x01”。

在将定时器 0 输入引脚的复用端口用作定时器的输入时，必须将各端口对应的端口模式寄存器（PMxx）的位置“1”并且将端口模式控制寄存器（PMCxx）的位置“0”。并设置端口复用功能配置寄存器（PStau0tin0\_CFG）。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

（例）将 P20/TI00 用作定时器输入的情况

将端口模式控制寄存器 2 的 PMC20 位置“0”。

将端口模式寄存器 2 的 PM20 位置“1”。

将端口输入复用功能配置寄存器 PStau0tin0\_CFG 置“0x20”。



## 5.3 通用定时器单元的基本规则

### 5.3.1 多通道联动运行功能的基本规则

多通道联动运行功能是将主控通道（主要对周期进行计数的基准定时器）和从属通道（遵从主控通道运行的定时器）组合实现的功能，使用时需要遵守几个规则。

多通道联动运行功能的基本规则如下所示。

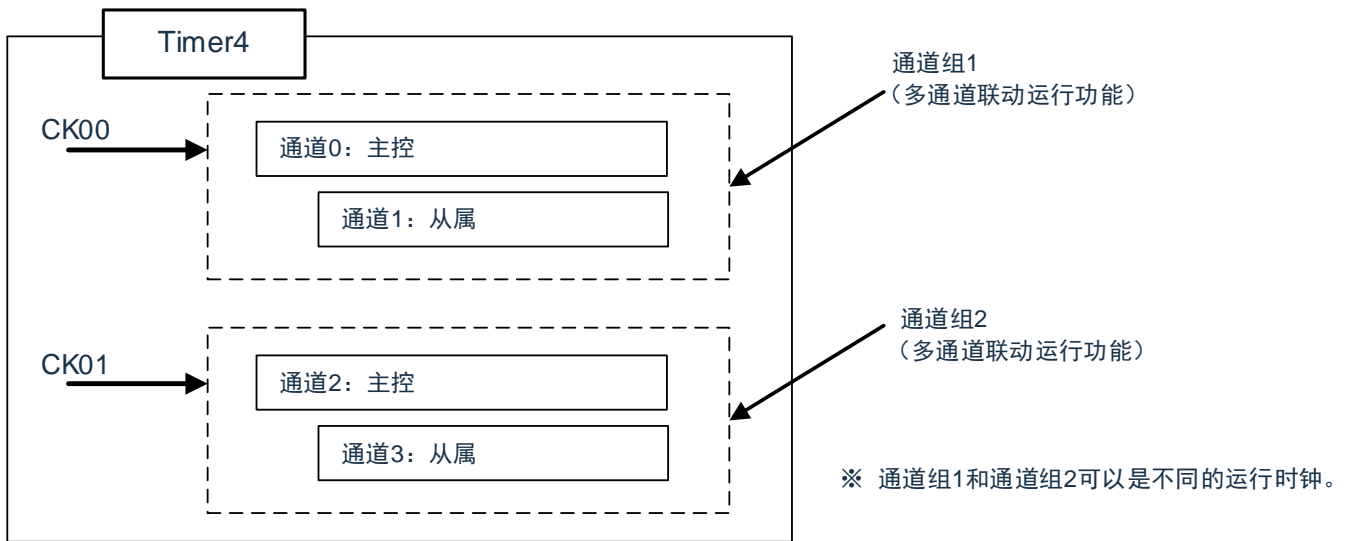
- 1) 只能将偶数通道（通道 0、通道 2）设定为主控通道。
- 2) 通道 0 以外的任何通道都能设定为从属通道。
- 3) 只能将主控通道的低位通道设定为从属通道。
- 4) 例在将通道 0 设定为主控通道时，能将通道 1 开始的通道（通道 1、通道 2、通道 3）设定为从属通道。
- 5) 能对 1 个主控通道设定多个从属通道。
- 6) 当使用多个主控通道时，不能设定跨越主控通道的从属通道。
- 7) 例在将通道 0 和通道 2 设定为主控通道时，能将通道 1 设定为主控通道 0 的从属通道，而不能将通道 3 设定为主控通道 0 的从属通道。
- 8) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的 CKSmn0 位和 CKSmn1 位（定时器模式寄存器 mn (TMRmn) 的 bit15 和 bit14) 的值需要是相同的设定值。
- 9) 主控通道能将 INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。
- 10) 从属通道能将主控通道的 INTTMmn（中断）、开始软件触发和计数时钟用作源时钟，但是不能将自己的 INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。
- 11) 主控通道不能将其他高位主控通道的 INTTMmn（中断）、开始软件触发和计数时钟用作源时钟。
- 12) 为了同时启动要联动的通道，需要同时设定联动通道的通道开始触发位 (TSmn)。
- 13) 只有联动的全部通道或者主控通道才能使用计数运行中的 TSmn 位的设定。不能只使用从属通道的 TSmn 位的设定。
- 14) 为了同时停止要联动的通道，需要同时设定联动通道的通道停止触发位 (TTmn)。
- 15) 在联动运行时，因为主控通道和从属通道需要相同的运行时钟，所以不能选择 CKm2/CKm3。
- 16) 定时器模式寄存器 m0 (TMRm0) 没有主控位而固定为“0”。但是，因为通道 0 是最高位的通道，所以在联动运行时能将通道 0 用作主控通道。

多通道联动运行功能的基本规则是适用于通道组群（形成 1 个多通道联动运行功能的主控通道和从属通道的集合）的规则。

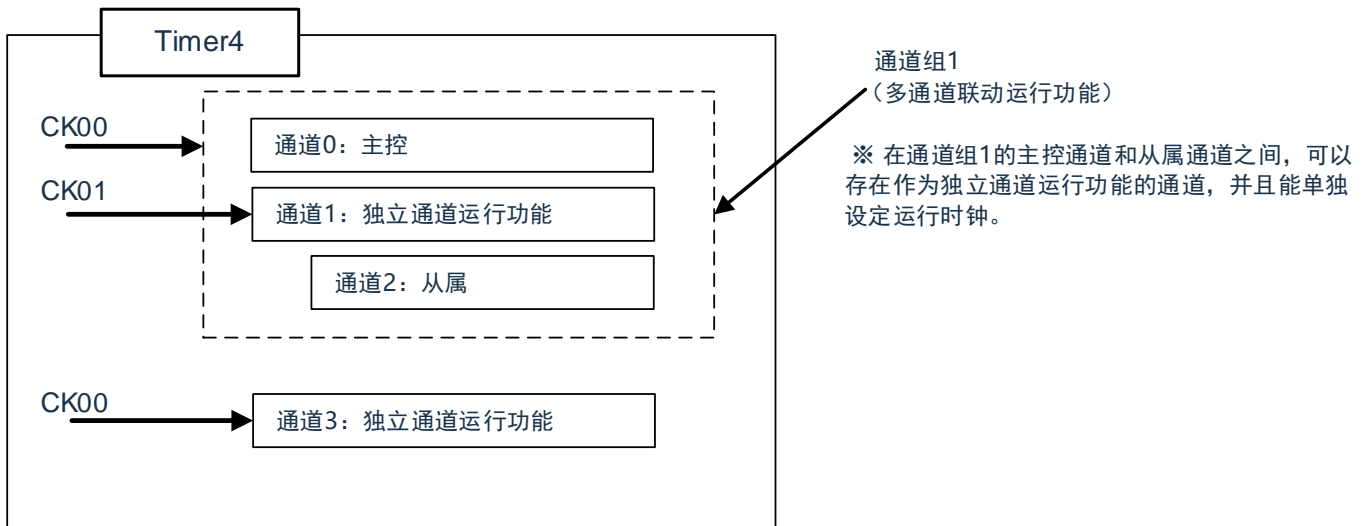
如果设定 2 个或者更多的相互不联动的通道群，通道群之间就不适用上述的基本规则。

注：m：单元号 (m=0) n：通道号 (n=0~3)。

例 1



例2



### 5.3.2 8 位定时器运行功能的基本规则（只限于通道1和通道3）

8 位定时器运行功能是将 16 位定时器的通道用作 2 个 8 位定时器的通道的功能。

只有通道 1 和通道 3 才能使用 8 位定时器运行功能，使用时需要遵守几个规则。

8 位定时器运行功能的基本规则如下所示。

- 1) 8 位定时器运行功能只适用于通道 1 和通道 3。
- 2) 当用作 8 位定时器时，将定时器模式寄存器 mn（TMRmn）的 SPLIT 位置“1”。
- 3) 高 8 位定时器能用作间隔定时器功能。
- 4) 在开始运行时，高 8 位定时器输出 INTTm1H（中断）（和 MDmn0 位为“1”的运行相同）。
- 5) 高 8 位定时器的运行时钟的选择取决于低位 TMRmn 寄存器的 CKSmn1 位和 CKSmn0 位的设定。
- 6) 对于高 8 位定时器，通过操作 TSHm1 位来开始通道的运行，并且通过操作 TTHm1 位来停止通道的运行。能通过 TEHm1 位确认通道的状态。
- 7) 低 8 位定时器的运行取决于 TMRmn 寄存器的设定，有以下 3 种支持低 8 位定时器运行的功能：
  - 间隔定时器功能
  - 外部事件计数器功能
  - 延迟计数功能
- 8) 对于低 8 位定时器，通过操作 TSm1/TSm3 位来开始通道的运行，并且通过操作 TTm1/TTm3 位来停止通道的运行。能通过 TEm1/TEm3 位确认通道的状态。
- 9) 在 16 位定时器运行时，TSHm1/TSHm3/TTHm1/TTHm3 位的操作无效。通过操作 TSm1/TSm3 位和 TTm1/TTm3 位使通道 1 和通道 3 运行。TEHm3 位和 TEHm1 位不变。
- 10) 8 位定时器功能不能使用联动运行功能（单触发脉冲、PWM 和多重 PWM）。

注：m：单元号（m=0） n：通道号（n=1、3）

## 5.4 计数器的运行

### 5.4.1 计数时钟 ( $f_{TCLK}$ )

通用定时器单元的计数时钟 ( $f_{TCLK}$ ) 能通过定时器模式寄存器 mn (TMRmn) 的 CCSmn 位选择以下任意一个时钟:

- CKSmn0 位和 CKSmn1 位指定的运行时钟 ( $f_{MCK}$ )
- TImn 引脚输入信号的有效边沿

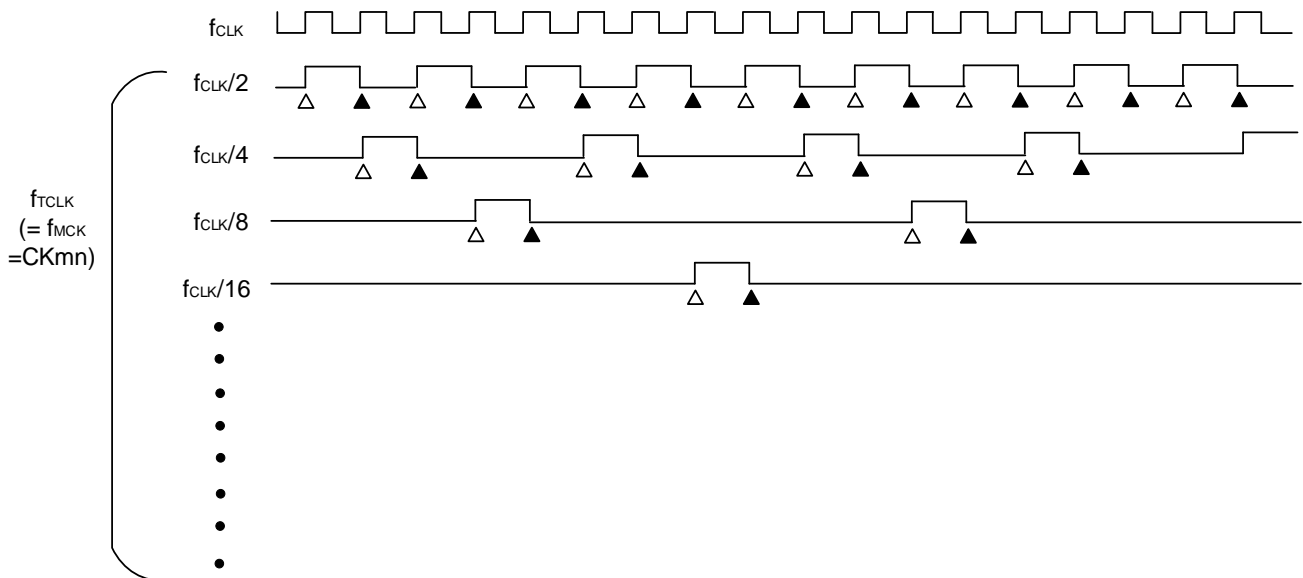
通用定时器单元被设计为与  $f_{CLK}$  同步运行, 因此计数时钟 ( $f_{TCLK}$ ) 的时序如下。

(1) 选择 CKSmn0 位和 CKSmn1 位指定的运行时钟 ( $f_{MCK}$ ) 的情况 (CCSmn=0)

根据定时器时钟选择寄存器 m (TPSm) 的设定, 计数时钟 ( $f_{TCLK}$ ) 为  $f_{CLK} \sim f_{CLK}/2^{15}$ 。但是, 当选择  $f_{CLK}$  的分频时, TPSm 寄存器选择的时钟是从上升沿开始只有 1 个  $f_{CLK}$  周期为高电平的信号。当选择  $f_{CLK}$  时, 固定为高电平。

为了取得与  $f_{CLK}$  的同步, 定时器计数寄存器 mn (TCRmn) 从计数时钟的上升沿开始延迟 1 个  $f_{CLK}$  时钟后进行计数, 出于方便而将其称为“在计数时钟的上升沿进行计数”。

图5-2  $f_{CLK}$ 和计数时钟 ( $f_{TCLK}$ ) 的时序 (CCSmn=0的情况)



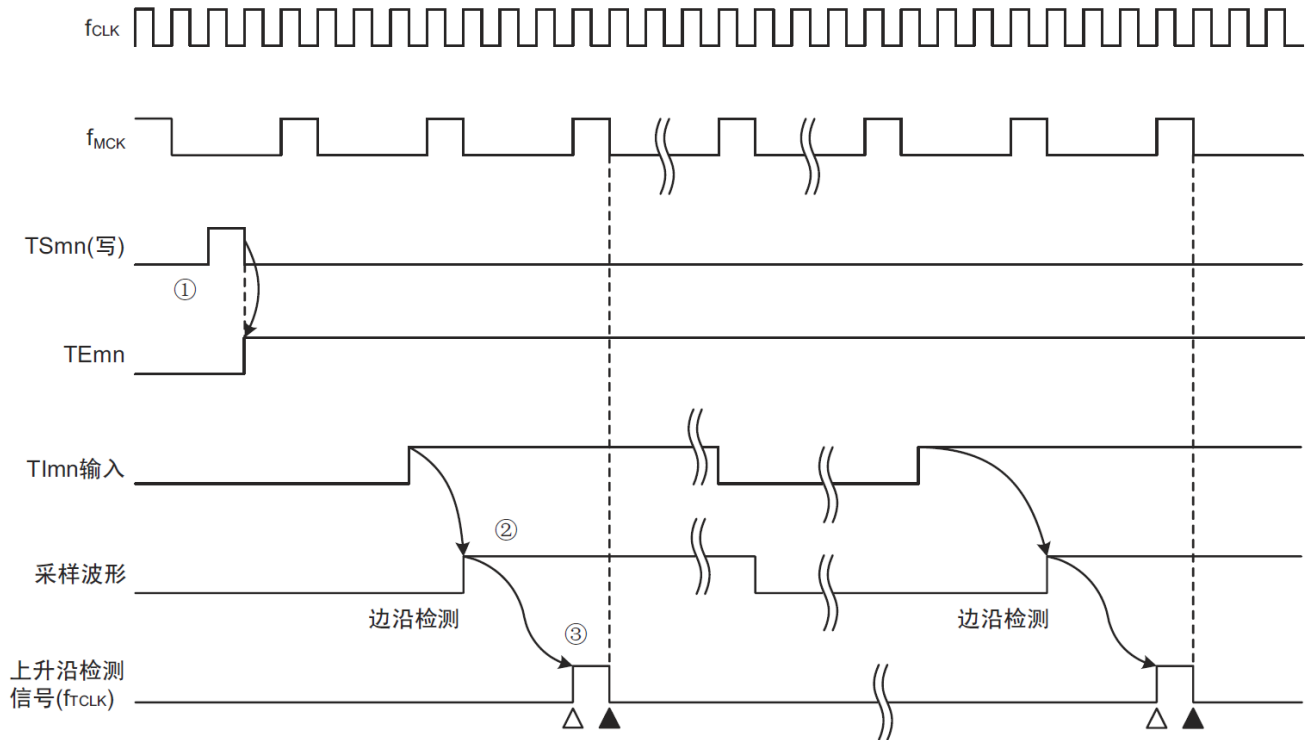
注1: Δ: 计数时钟的上升沿; ▲: 同步、计数器的递增/递减

注2:  $f_{CLK}$ : CPU/外围硬件的时钟

(2)选择 TImn 引脚输入信号的有效边沿的情况 (CCSmn=1)

计数时钟 ( $f_{TCLK}$ ) 是检测 TImn 引脚输入信号的有效边沿并且与下一个  $f_{MCK}$  上升沿同步的信号。实际上, 这是比 TImn 引脚的输入信号延迟了 1~2 个  $f_{MCK}$  时钟的信号。为了取得与  $f_{CLK}$  的同步, 定时器计数寄存器 mn (TCRmn) 从计数时钟的上升沿开始延迟 1 个  $f_{CLK}$  时后进行计数, 出于方便而将其称为“在 TImn 引脚输入信号的有效边沿进行计数”。

图5-3 计数时钟 ( $f_{TCLK}$ ) 的时序 (CCSmn=1)



- ① 通过将 TSmn 位置位来开始定时器的运行, 并且等待 TImn 输入的有效边沿。
- ② 通过  $f_{MCK}$  对 TImn 输入的上升沿进行采样。
- ③ 在采样信号的上升沿检测边沿, 并且输出检测信号 (计数时钟)。

注1: Δ: 计数时钟的上升沿; ▲: 同步、计数器的递增/递减

注2:  $f_{CLK}$ : CPU/外围硬件的时钟

$f_{MCK}$ : 通道n的运行时钟

注3: 输入脉冲间隔的测量、输入信号高低电平的测量、延迟计数器和单触发脉冲输出功能的TImn输入也是同样的波形。

## 5.4.2 计数器的开始时序

通过将定时器通道开始寄存器 m (TSMn) 的 TSMn 位置位, 定时器计数寄存器 mn (TCRmn) 进入运行允许状态。

从计数允许状态到定时器计数寄存器 mn (TCRmn) 开始计数为止的运行如表 5-25 所示。

表5-25 从计数允许状态到定时器计数寄存器mn (TCRmn) 开始计数为止的运行

定时器的运行模式	将TSMn位置“1”后的运行
间隔定时器模式	从检测到开始触发 (TSMn=1) 到产生计数时钟为止, 不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器, 并且通过后续的计数时钟进行递减计数 (参照“5.4.3(1) 间隔定时器模式的运行”)。
事件计数器模式	通过给TSMn位写“1”, 将TDRmn寄存器的值装入TCRmn寄存器。 如果检测到TIMn的输入边沿, 就通过后续的计数时钟进行递减计数。(参照“5.4.3 (2)事件计数器模式的运行”)。
捕捉模式	从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第1个计数时钟将“0000H”装入TCRmn寄存器, 并且通过后续的计数时钟进行递增计数 (参照“5.4.3(3)捕捉模式的运行 (输入脉冲的间隔测量)”)。
单次计数模式	通过在定时器停止运行 (TEmn=0) 的状态下给TSMn位写“1”, 进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器, 并且通过后续的计数时钟进行递减计数 (参照“5.4.3(4)单次计数模式的运行”)。
捕捉&单次计数模式	通过在定时器停止运行 (TEmn=0) 的状态下给TSMn位写“1”, 进入开始触发的等待状态 从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第1个计数时钟将“0000H”装入TCRmn寄存器, 并且通过后续的计数时钟进行递增计数 (参照“5.4.3(5)捕捉&单次计数模式的运行 (高电平宽度的测量)”)。

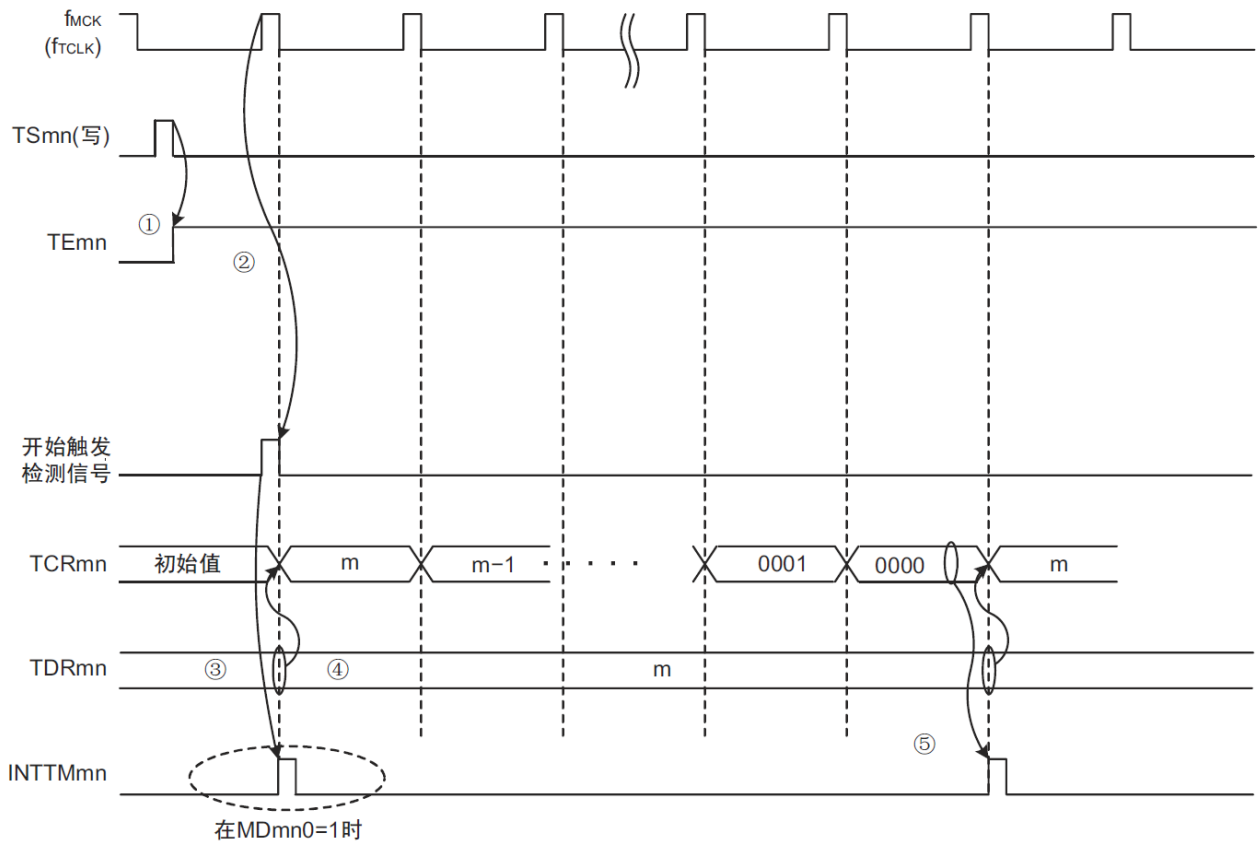
### 5.4.3 计数器的运行

以下说明各模式的计数器运行。

#### (1) 间隔定时器模式的运行

- ① 通过给 TSmn 位写“1”，进入运行允许状态 (TEmn=1)。定时器计数寄存器 mn (TCRmn) 保持初始值，直到产生计数时钟为止。
- ② 通过允许运行后的第 1 个计数时钟 (f<sub>MCK</sub>) 产生开始触发信号。
- ③ 当 MDmn0 位为“1”时，通过开始触发信号产生 INTTMmn。
- ④ 通过允许运行后的第 1 个计数时钟将定时器数据寄存器 mn (TDRmn) 的值装入 TCRmn 寄存器，并且以间隔定时器模式开始计数。
- ⑤ 如果 TCRmn 寄存器递减计数到“0000H”，就通过下一个计数时钟 (f<sub>MCK</sub>) 产生 INTTMmn 并且在将定时器数据寄存器 mn (TDRmn) 的值装入 TCRmn 寄存器后继续计数。

图5-4 运行时序 (间隔定时器模式)



注1: 因为第1个计数时钟周期的运行在写 TSmn 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MDmn0 位置“1”，以便能在开始计数时产生中断。

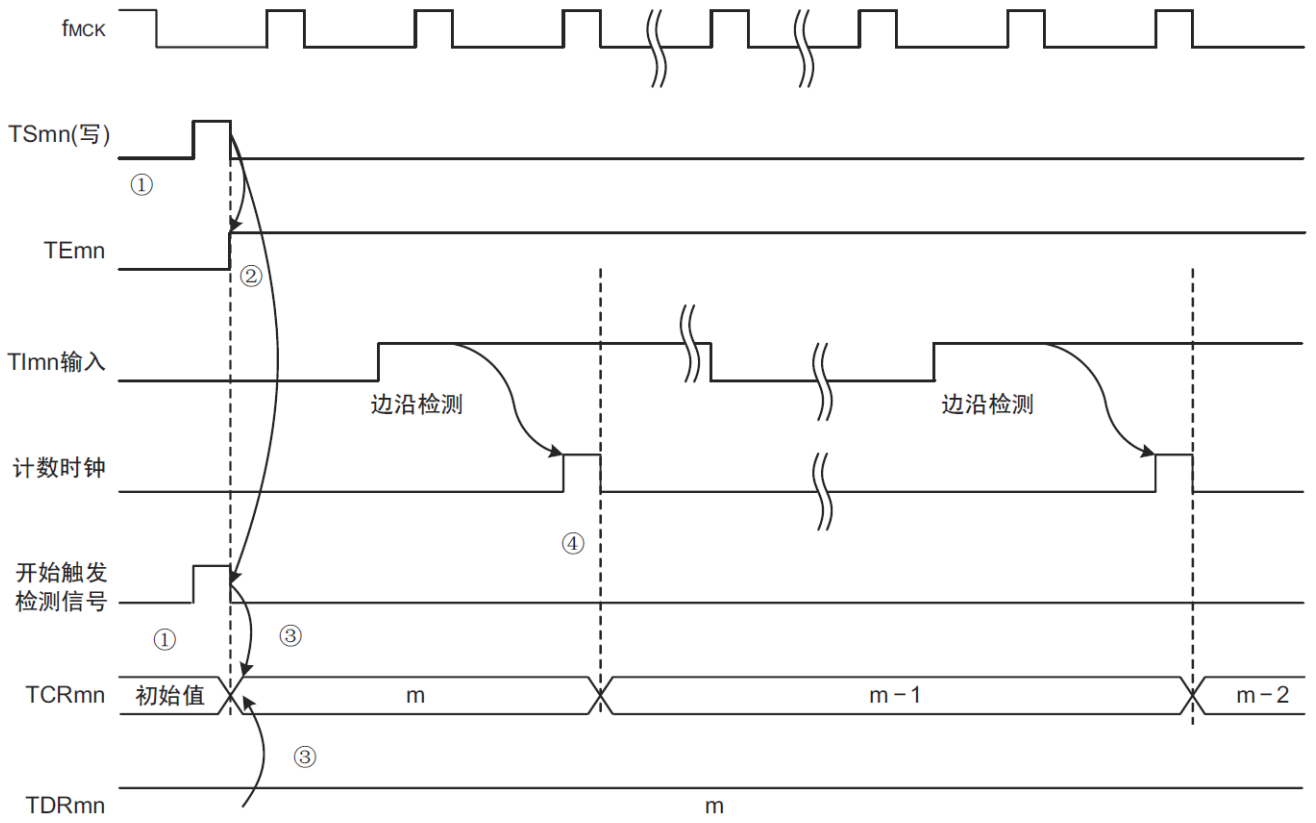
注2: f<sub>CLK</sub>、开始触发检测信号和 INTTMmn 与 f<sub>CLK</sub> 同步并且在1个时钟内有效。

(2) 事件计数器模式的运行

- ① 在运行停止状态 (TE<sub>mn</sub>=0) 的期间, 定时器计数寄存器 mn (TCR<sub>mn</sub>) 保持初始值。
- ② 通过给 TS<sub>mn</sub> 位写“1”, 进入运行允许状态 (TE<sub>mn</sub>=1)。
- ③ 在 TS<sub>mn</sub> 位和 TE<sub>mn</sub> 位都变为“1”的同时将定时器数据寄存器 mn (TDR<sub>mn</sub>) 的值装入 TCR<sub>mn</sub> 寄存器, 并且开始计数。

此后, 在 TI<sub>mn</sub> 输入的有效边沿, 通过计数时钟对 TCR<sub>mn</sub> 寄存器的值进行递减计数。

图5-5 运行时序 (事件计数器模式)



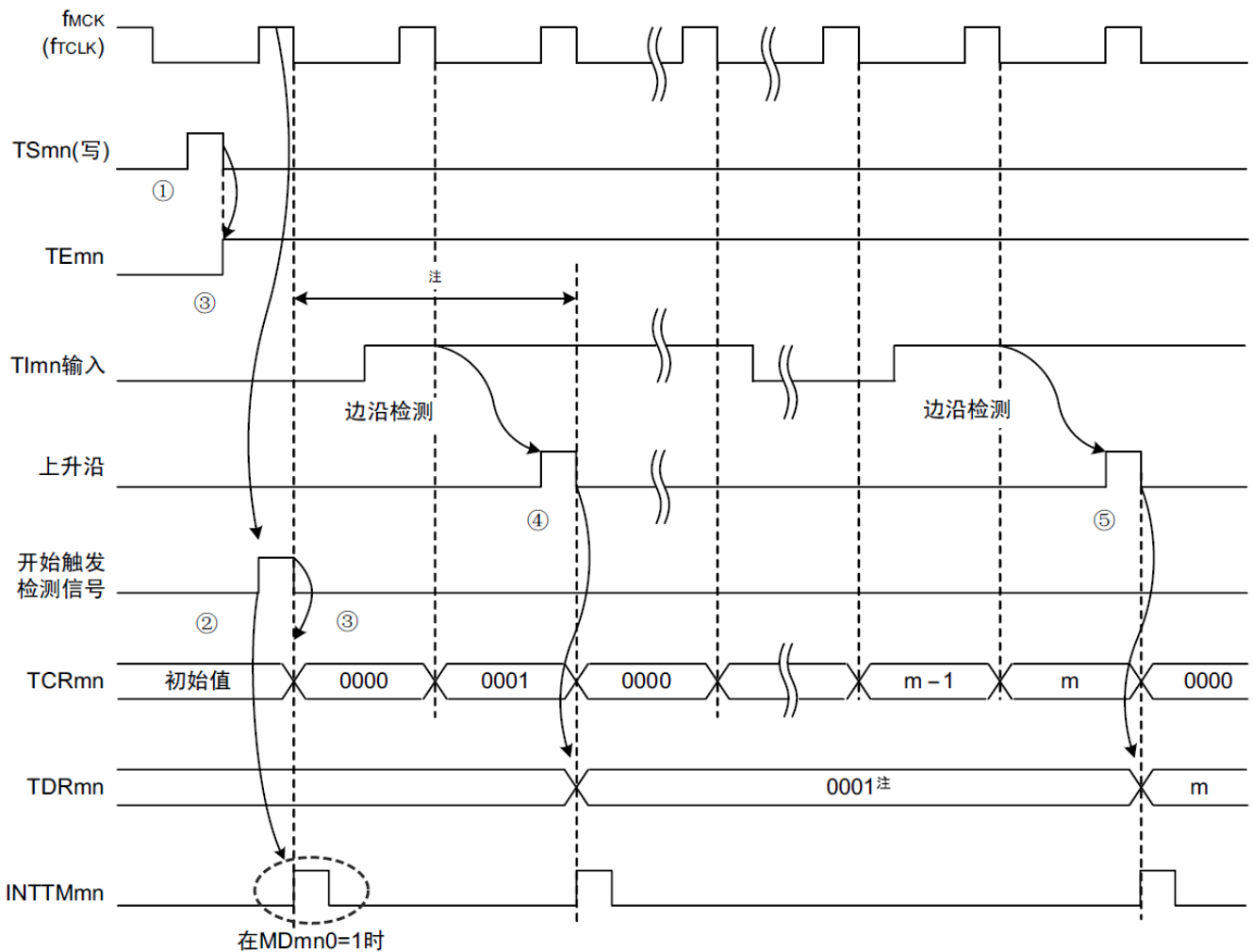
注: 1个周期的误差是因为TI<sub>mn</sub>输入与计数时钟 (f<sub>MCK</sub>) 不同步。



(3) 捕捉模式的运行（输入脉冲的间隔测量）

- ① 通过给 TSmn 位写“1”，进入运行允许状态（TEmn=1）。
- ② 定时器计数寄存器 mn（TCRmn）保持初始值，直到产生计数时钟为止。
- ③ 通过允许运行后的第 1 个计数时钟（f<sub>MCK</sub>）产生开始触发信号。然后，将“0000H”装入 TCRmn 寄存器并且以捕捉模式开始计数（当 MDmn0 位为“1”时，通过开始触发信号产生 INTTMmn）。
- ④ 如果检测到 TImn 输入的有效边沿，就将 TCRmn 寄存器的值捕捉到 TDRmn 寄存器，并且产生 INTTMmn 中断。此时的捕捉值没有意义。TCRmn 寄存器从“0000H”开始继续计数。
- ⑤ 如果检测到下一个 TImn 输入的有效边沿，就将 TCRmn 寄存器的值捕捉到 TDRmn 寄存器，并且产生 INTTMmn 中断。

图5-6 运行时序（捕捉模式：输入脉冲的间隔测量）



注1：在开始前将时钟输入到TImn（有触发）时，即使没有检测到边沿也通过检测触发来开始计数，因此第1次捕捉时（④）的捕捉值不是脉冲间隔（在此例子中，0001：2个时钟间隔），必须忽视。

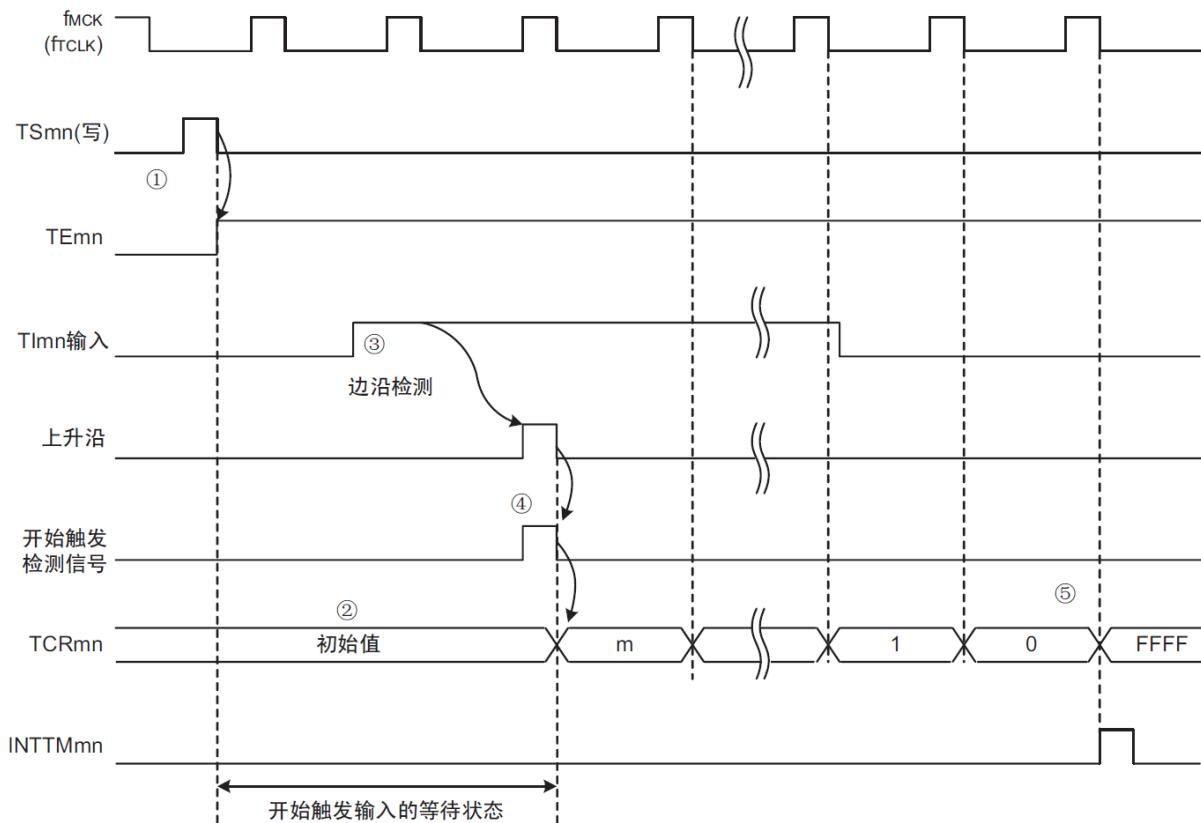
注2：因为第1个计数时钟周期的运行在写TSmn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

注3：1个周期的误差是因为TImn输入与计数时钟（f<sub>MCK</sub>）不同步。

单次计数模式的运行

- ① 通过给 TSmn 位写“1”，进入运行允许状态 (TEmn=1)。
- ② 定时器计数寄存器 mn (TCRmn) 保持初始值，直到产生开始触发信号为止。
- ③ 检测 TImn 输入的上升沿。
- ④ 在产生开始触发信号后将 TDRmn 寄存器的值 (m) 装入 TCRmn 寄存器，并且开始计数。
- ⑤ 当 TCRmn 寄存器递减计数到“0000H”时，产生 INTTMmn 中断，并且 TCRmn 寄存器的值变为“FFFFH”，停止计数。

图5-7 运行时序 (单次计数模式)

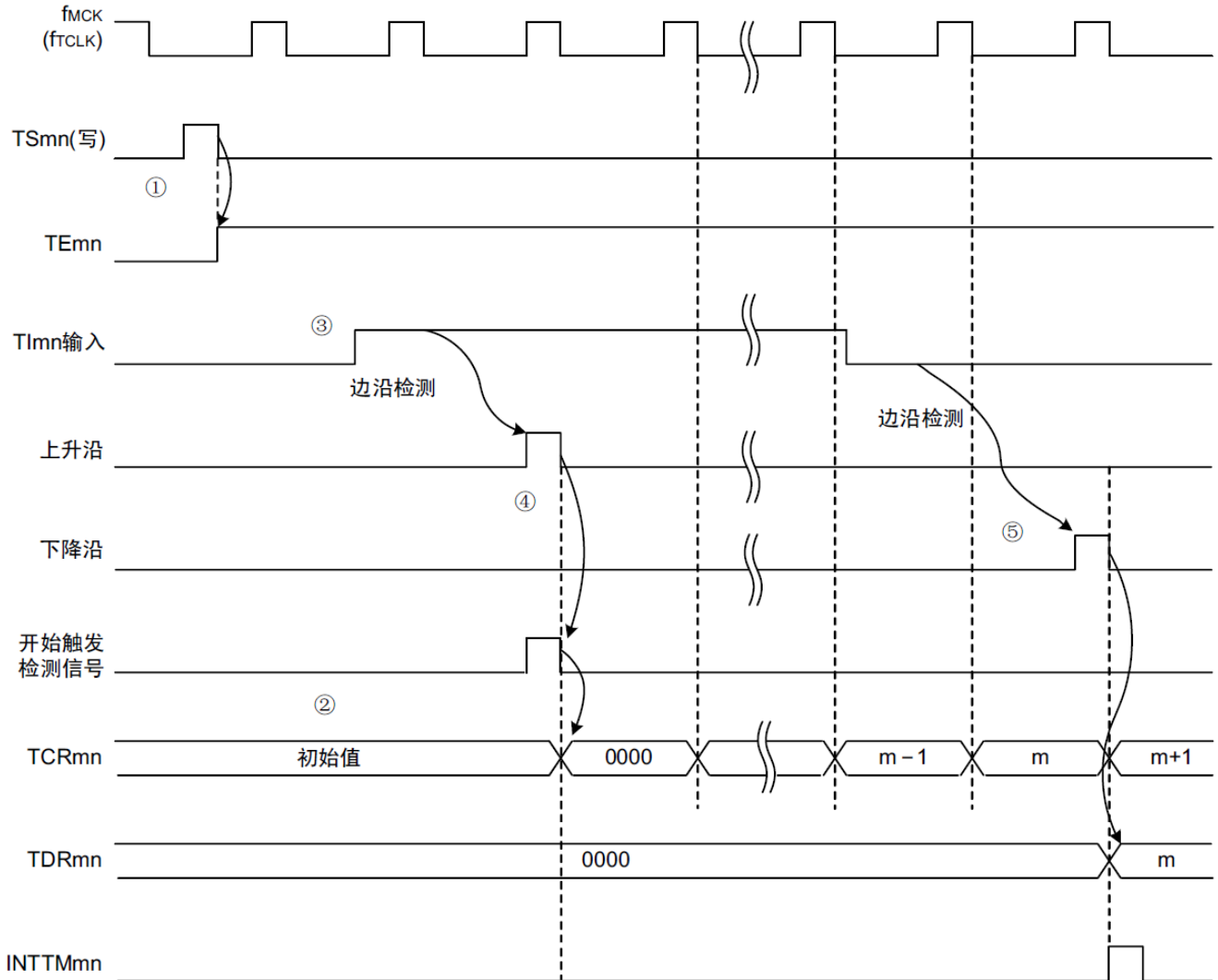


注：1个周期的误差是因为TImn输入与计数时钟 (f<sub>MCK</sub>) 不同步。

(4) 捕捉&单次计数模式的运行（高电平宽度的测量）

- ① 通过给定时器通道开始寄存器 m (TSMn) 的 TSMn 位写“1”，进入运行允许状态 (TEmn=1)。
- ② 定时器计数寄存器 mn (TCRmn) 保持初始值，直到产生开始触发信号为止。
- ③ 检测 TImn 输入的上升沿。
- ④ 在产生开始触发信号后将“0000H”装入 TCRmn 寄存器，并且开始计数。
- ⑤ 如果检测到 TImn 输入的下降沿，就将 TCRmn 寄存器的值捕捉到 TDRmn 寄存器，并且产生 INTTMmn 中断。

图5-8 运行时序（捕捉 & 单次计数模式：高电平宽度的测量）

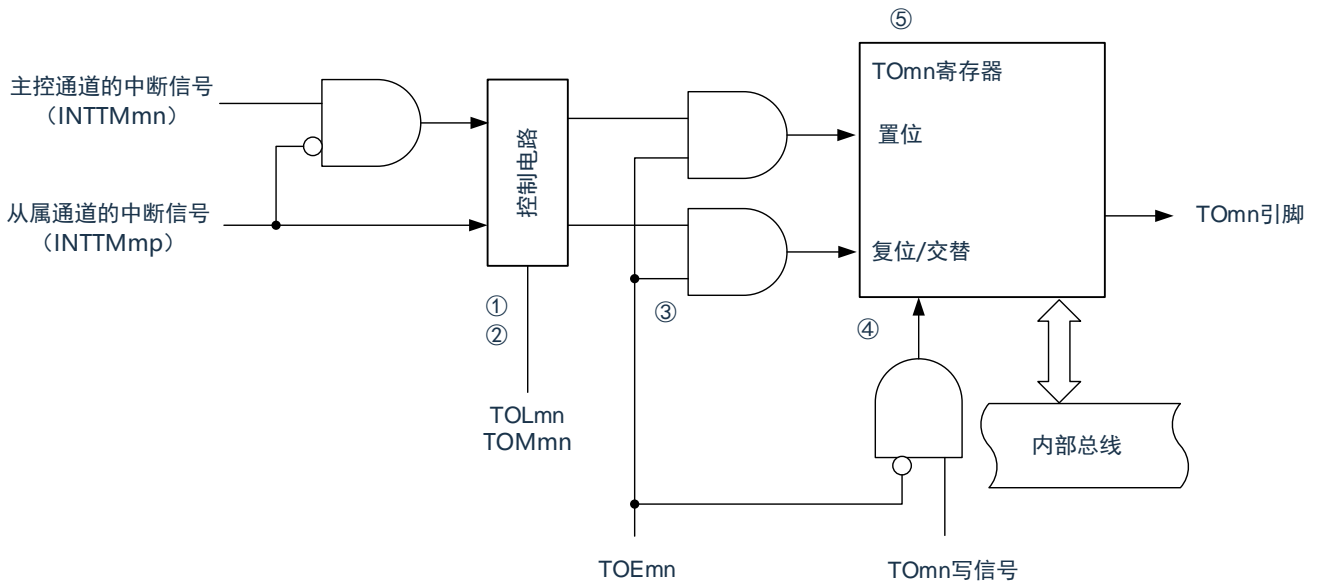


注：1个周期的误差是因为 $TImn$ 输入与计数时钟 ( $f_{MCK}$ ) 不同步。

## 5.5 通道输出（TOmn引脚）的控制

### 5.5.1 TOmn引脚输出电路的结构

图5-9 输出电路的结构



- ① 当 TOMmn 位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器 m（TOLm）的设定值，只将 INTTMmp（从属通道定时器中断）传给定时器输出寄存器 m（TOM）。
- ② 当 TOMmn 位为“1”（从属通道输出模式）时，将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。  
此时，TOLm 寄存器有效并且进行以下信号的控制：  
TOLmn=0 时：正相运行（INTTMmn 置位、INTTMmp 复位）  
TOLmn=1 时：反相运行（INTTMmn 复位、INTTMmp 置位）  
当同时产生 INTTMmn 和 INTTMmp 时（PWM 输出的 0%输出），优先 INTTMmp（复位信号）而屏蔽 INTTMmn（置位信号）。
- ③ 在允许定时器输出（TOEmn=1）的状态下，将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。TOM 寄存器的写操作（TOmn 写信号）无效。  
当 TOEmn 位为“1”时，除了中断信号以外，不改变 TOmn 引脚的输出。  
要对 TOmn 引脚的输出电平进行初始化时，需要在设定为禁止定时器输出（TOEmn=0）后给 TOM 寄存器写值。
- ④ 在禁止定时器输出（TOEmn=0）的状态下，对象通道的 TOmn 位的写操作（TOmn 写信号）有效。当定时器输出为禁止状态（TOEmn=0）时，不将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。
- ⑤ 能随时读 TOM 寄存器，并且能确认 TOmn 引脚的输出电平。

注：m：单元号（m=0） n：通道号 n=0~3（主控通道：n=0、2）

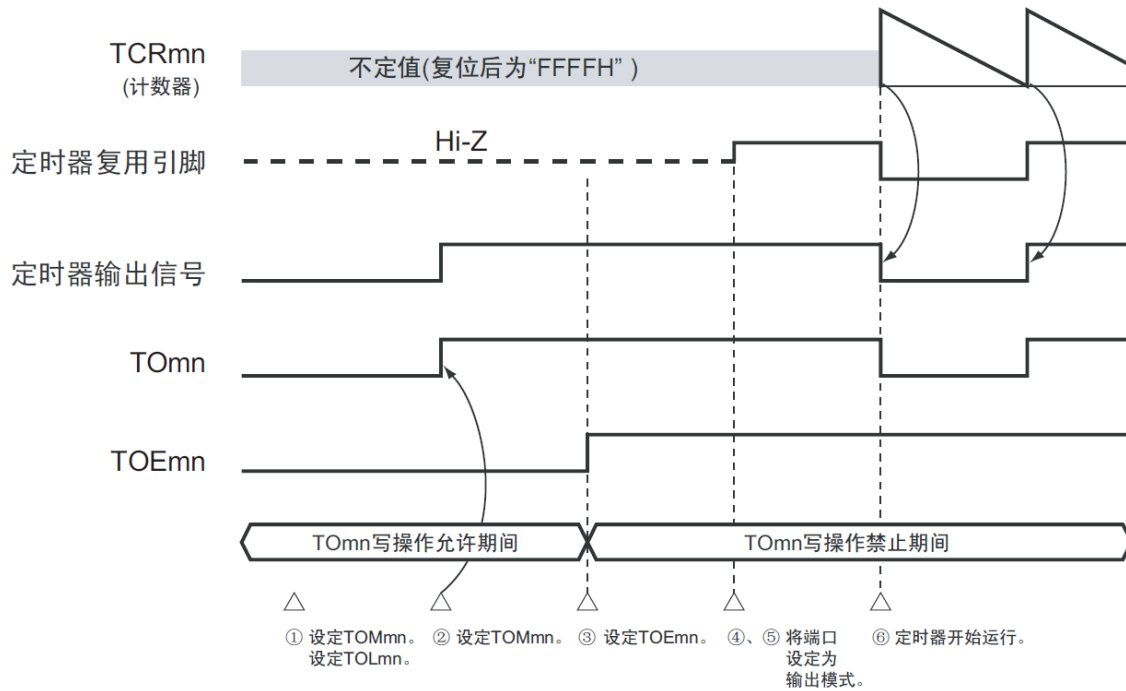
p：从属通道号

n=0：p=1、2、3 n=2：p=3

## 5.5.2 TOmn引脚的输出设定

从TOmn输出引脚的初始设定到定时器开始运行的步骤和状态变化如下所示。

图5-10 从设定定时器的输出到开始运行的状态变化



### ① 设定定时器输出的运行模式。

TOm<sub>mn</sub> 位 (0: 主控通道输出模式、1: 从属通道输出模式)

TOL<sub>mn</sub> 位 (0: 正逻辑输出、1: 负逻辑输出)

### ② 通过设定定时器输出寄存器 m (TOm)，将定时器输出信号设定为初始状态。

### ③ 给 TOEmn 位写“1”，允许定时器输出 (禁止写 TOm 寄存器)。

### ④ 通过端口模式控制寄存器 (PMCxx) 将端口设定为数字输入/输出。

### ⑤ 将端口的输入/输出设定为输出。

### ⑥ 允许定时器运行 (TS<sub>mn</sub>=1)。

注: m: 单元号 (m=0) n: 通道号 n=0~3

### 5.5.3 通道输出运行的注意事项

(1) 有关定时器运行中的  $TOm$ 、 $TOEm$ 、 $TOLm$ 、 $TOMm$  寄存器的设定值变更

定时器的运行（定时器计数寄存器  $mn$  ( $TCRmn$ ) 和定时器数据寄存器  $mn$  ( $TDRmn$ ) 的运行）和  $TOmn$  输出电路相互独立。因此，定时器输出寄存器  $m$  ( $TOm$ )、定时器输出允许寄存器  $m$  ( $TOEm$ ) 和定时器输出电平寄存器  $m$  ( $TOLm$ ) 的设定值的变更不会影响定时器的运行，能在定时器运行中更改设定值。但是，为了在各定时器的运行中从  $TOmn$  引脚输出期待的波形，必须设定为 5.7 和 5.8 所示的各运行的寄存器设定内容例子的值。

如果在产生各通道的定时器中断 ( $INTTMmn$ ) 信号前后更改除了  $TOm$  寄存器以外的  $TOEm$  寄存器和  $TOLm$  寄存器的设定值，就根据是在产生定时器中断 ( $INTTMmn$ ) 信号前更改还是在产生后更改， $TOmn$  引脚输出的波形可能不同。

注：m：单元号 (m=0) n：通道号 n=0~3

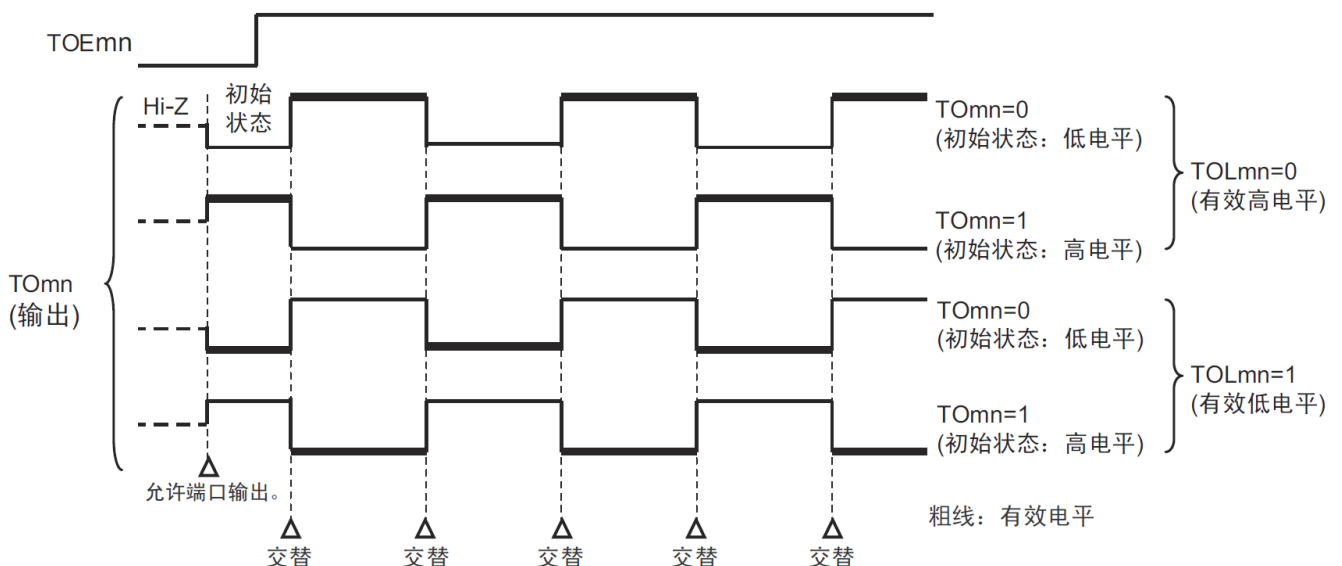
(2) 有关  $TOmn$  引脚的初始电平和定时器开始运行后的输出电平

在允许端口输出前并且在禁止定时器输出 ( $TOEmn=0$ ) 的状态下写定时器输出寄存器  $m$  ( $TOm$ )，在更改初始电平后设定为定时器输出允许状态 ( $TOEmn=1$ ) 时的  $TOmn$  引脚输出电平的变化如下所示。

(a) 在主导通道输出模式 ( $TOMmn=0$ ) 中开始运行的情况

在主导通道输出模式 ( $TOMmn=0$ ) 中，定时器输出电平寄存器  $m$  ( $TOLm$ ) 的设定无效。如果在设定初始电平后开始定时器的运行，就通过产生交替信号反相  $TOmn$  引脚的输出电平。

图5-11 交替输出时 ( $TOMmn=0$ ) 的  $TOmn$  引脚输出状态



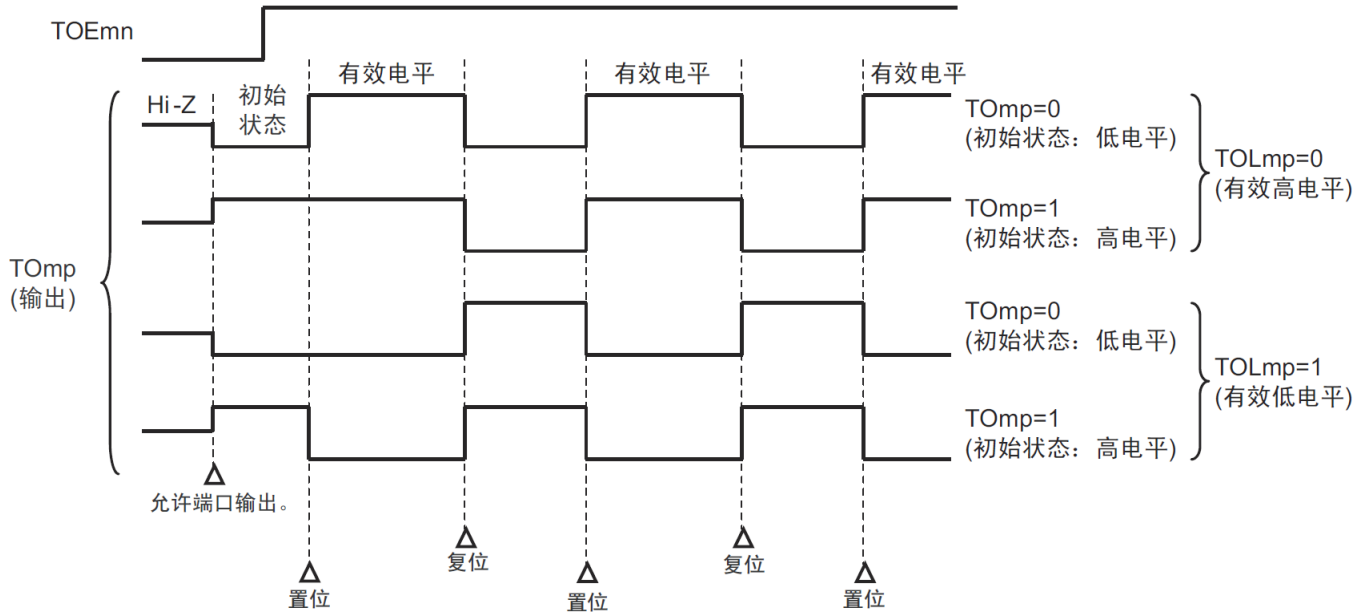
注1：交替：反相  $TOmn$  引脚的输出状态。

注2：m：单元号 (m=0) n：通道号 n=0~3

(b) 在从属通道输出模式 (TOMmn=1) 中开始运行的情况 (PWM 输出)

在从属通道输出模式 (TOMmn=1) 中, 有效电平取决于定时器输出电平寄存器 m (TOLmn) 的设置。

图5-12 PWM输出时 (TOMmn=1) 的TOMn引脚输出状态



注1: 置位: TOmp引脚的输出信号从无效电平变为有效电平。

注2: 复位: TOmp引脚的输出信号从有效电平变为无效电平。

注3: m: 单元号 (m=0) n: 通道号 (p=1~3)

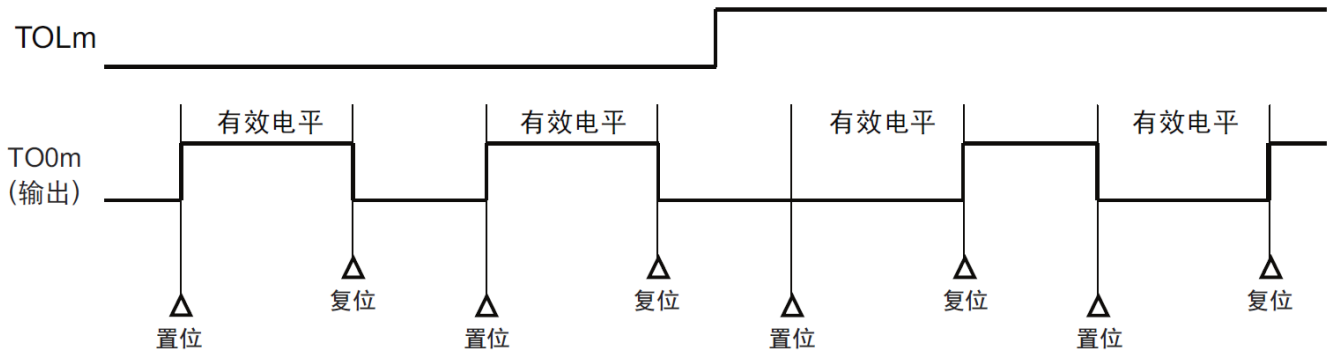
(3) 有关从属通道输出模式 (TOMmn=1) 的 TOmn 引脚变化

(a) 在定时器运行中更改定时器输出电平寄存器 m (TOLm) 的设定的情况

如果在定时器运行中更改 TOLm 寄存器的设定, 就在产生 TOmn 引脚变化条件时设定有效。无法通过改写 TOLm 寄存器来改变 TOmn 引脚的输出电平。

当 TOMmn 位为“1”时, 在定时器运行中 (TEmn=1) 更改 TOLm 寄存器的值时的运行如下所示。

图5-13 在定时器运行中更改TOLm寄存器的内容时的运行



注1: 置位: TOmp引脚的输出信号从无效电平变为有效电平。

注2: 复位: TOmp引脚的输出信号从有效电平变为无效电平。

注3: m: 单元号 (m=0) n: 通道号 (n=1~3)

(b) 置位/复位时序

(c) 为了在PWM输出时实现0%和100%的输出, 通过从属通道将产生主控通道定时器中断(INTTMnn)时的 TOmn引脚/TOmn位的置位时序延迟1个计数时钟。

当置位条件和复位条件同时产生时, 优先复位条件。

按照以下方法设定主控/从属通道时的置位/复位运行状态如图 5-14 所示。

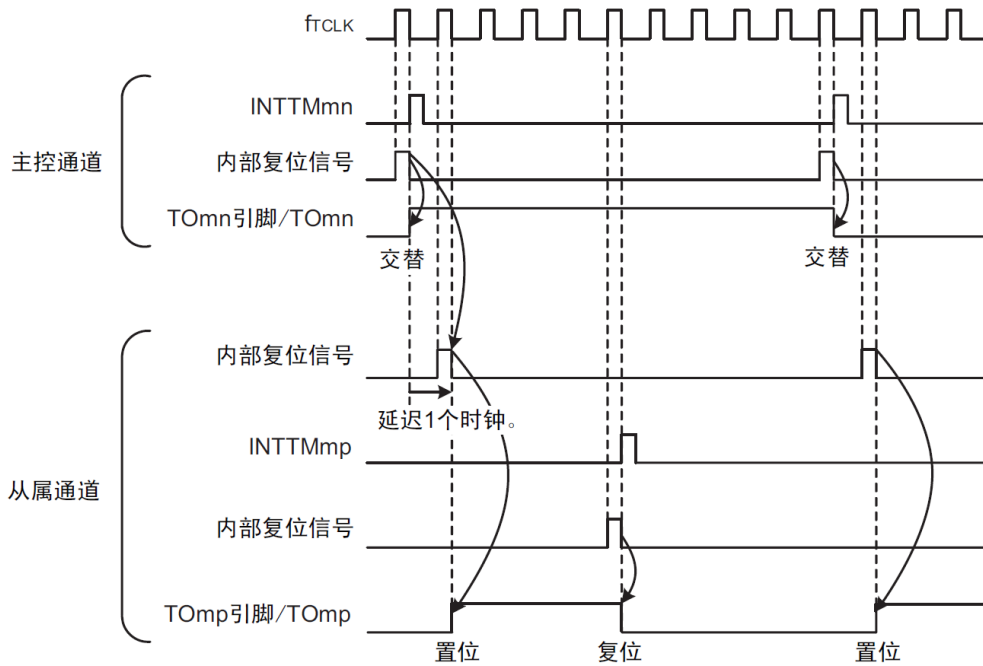
主控通道: TOEmn=1、TOMmn=0、TOLmn=0

从属通道: TOEmp=1、TOMmp=1、TOLmp=0

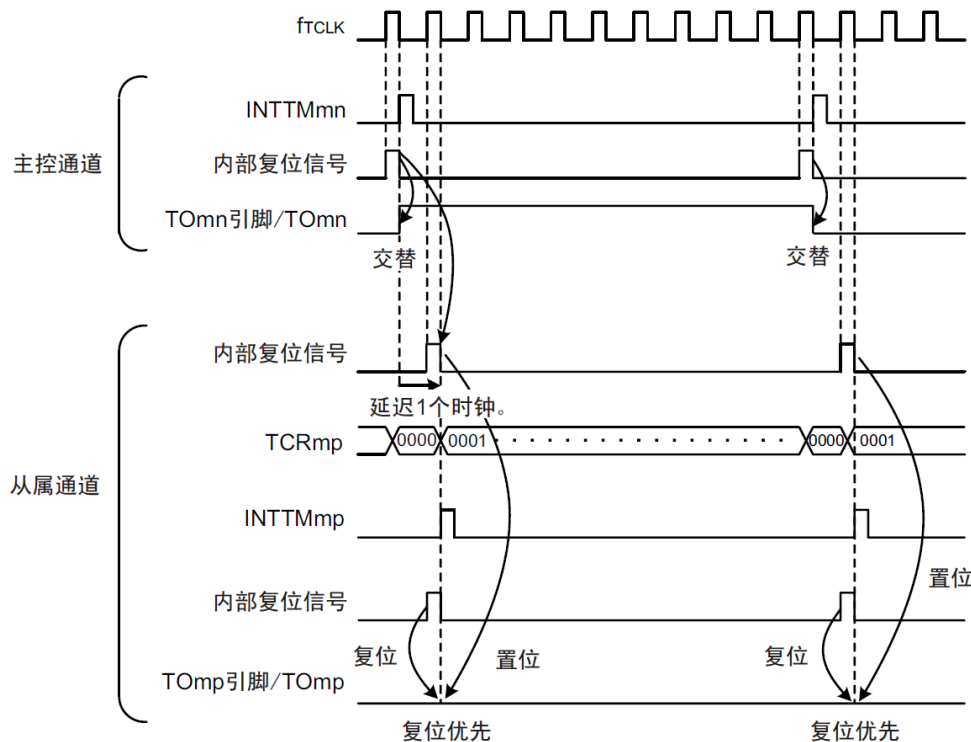


图5-14 置位/复位时序运行状态

(1) 基本运行时序



(2) 0%占空比的运行时序



注1: 内部复位信号: TOn引脚的复位/交替信号

注2: 内部置位信号: TOn引脚的置位信号

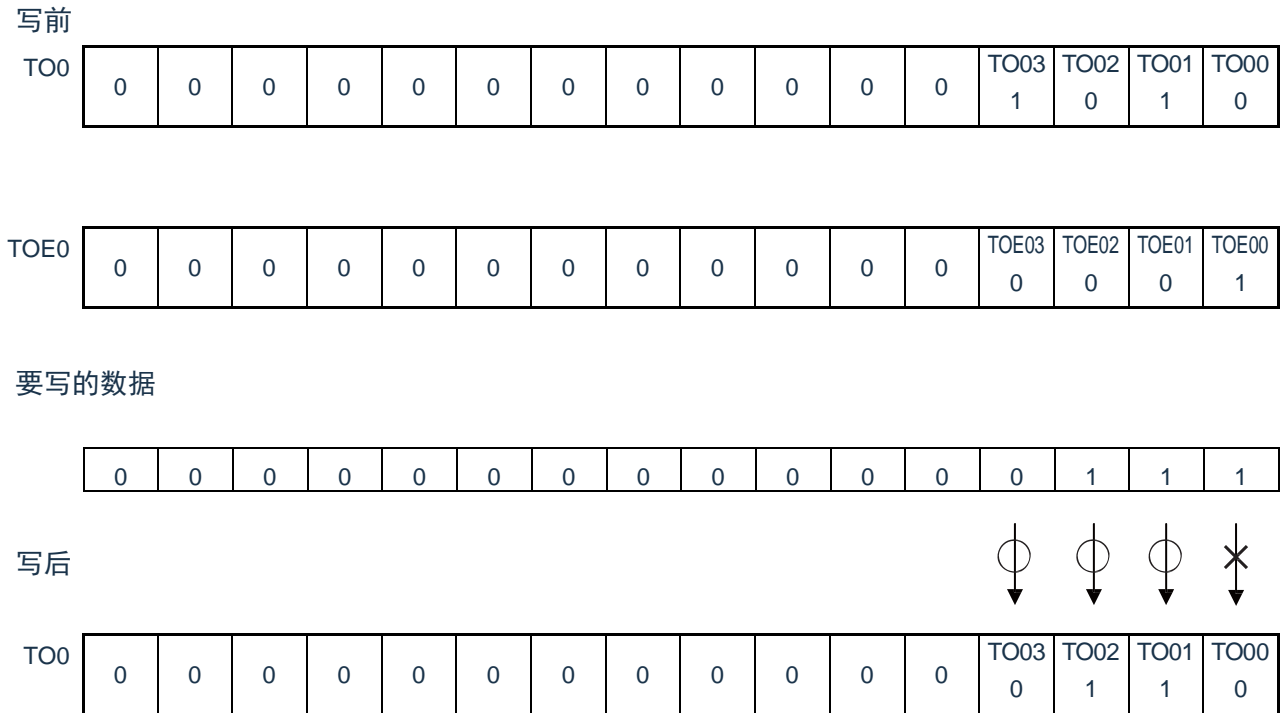
注3: m: 单元号 (m=0) n: 通道号 n=0~3 (主控通道: n=0、2)

p: 从属通道号 n=0: p=1、2、3 n=2: p=3

### 5.5.4 TOmn位的一次性操作

和定时器通道开始寄存器 m (TSm) 相同, 定时器输出寄存器 m (TOm) 有全部通道的设定位 (TOmn), 因此能一次性地操作全部通道的 TOmn 位。

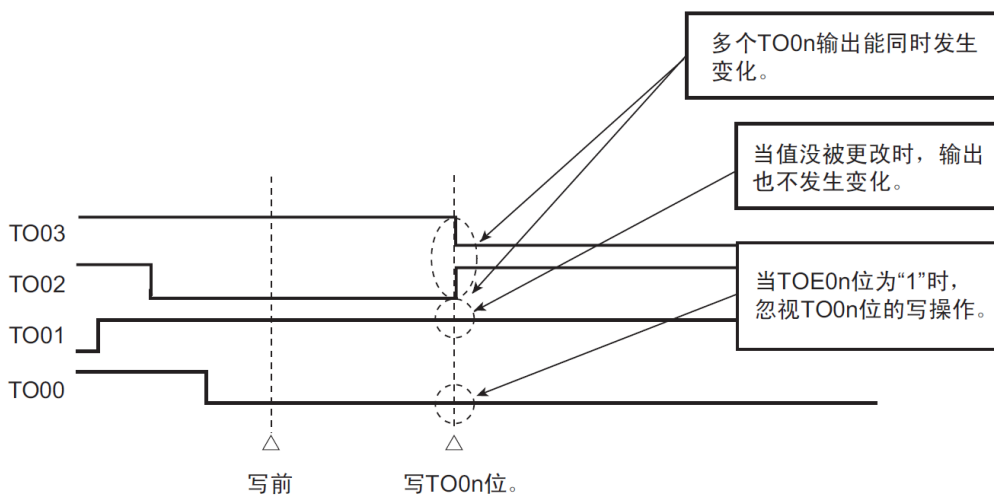
表5-26 TO0n位的一次性操作例子



只能写 TOEmn 位为“0”的 TOmn 位, 忽视 TOEmn 位为“1”的 TOmn 位的写操作。

TOEmn 位为“1”的 TOmn (通道输出) 不受写操作的影响, 即使写 TOmn 位也被忽视, 由定时器运行引起的输出变化正常进行。

图5-15 一次性操作TO0n位时的TO0n引脚状态



注: m: 单元号 (m=0) n: 通道号 (n=0~3)

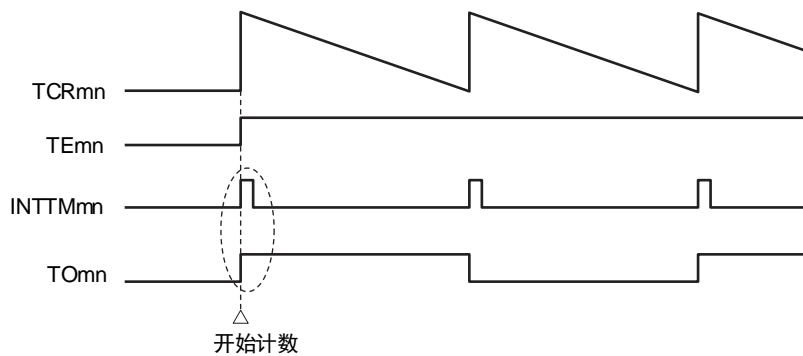
### 5.5.5 有关开始计数时的定时器中断和TOMn引脚输出

在间隔定时器模式或者捕捉模式中，定时器模式寄存器 mn (TMRmn) 的 MDmn0 位是设定是否在开始计数时产生定时器中断的位。

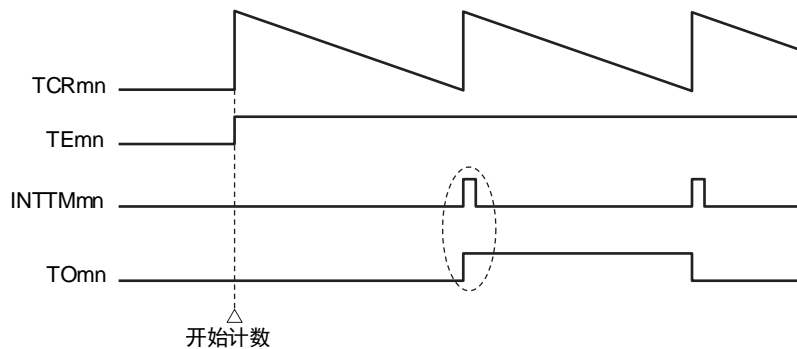
当 MDmn0 位为“1”时，能通过产生定时器中断 (INTTMmn) 得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和 TOMn 输出。设定为间隔定时器模式 (TOEmn=1、TOMmn=0) 时的运行例子如下所示。

图5-16 开始计数时的定时器中断和TOMn输出的运行例子

(a) MDmn0位为“1”的情况



(b) MDmn0位为“0”的情况



当 MDmn0 位为“1”时，在开始计数时输出定时器中断 (INTTMmn) 并且 TOMn 进行交替输出。

当 MDmn0 位为“0”时，在开始计数时不输出定时器中断 (INTTMmn) 并且 TOMn 也不发生变化，而在对 1 个周期进行计数后输出 INTTMmn 并且 TOMn 进行交替输出。

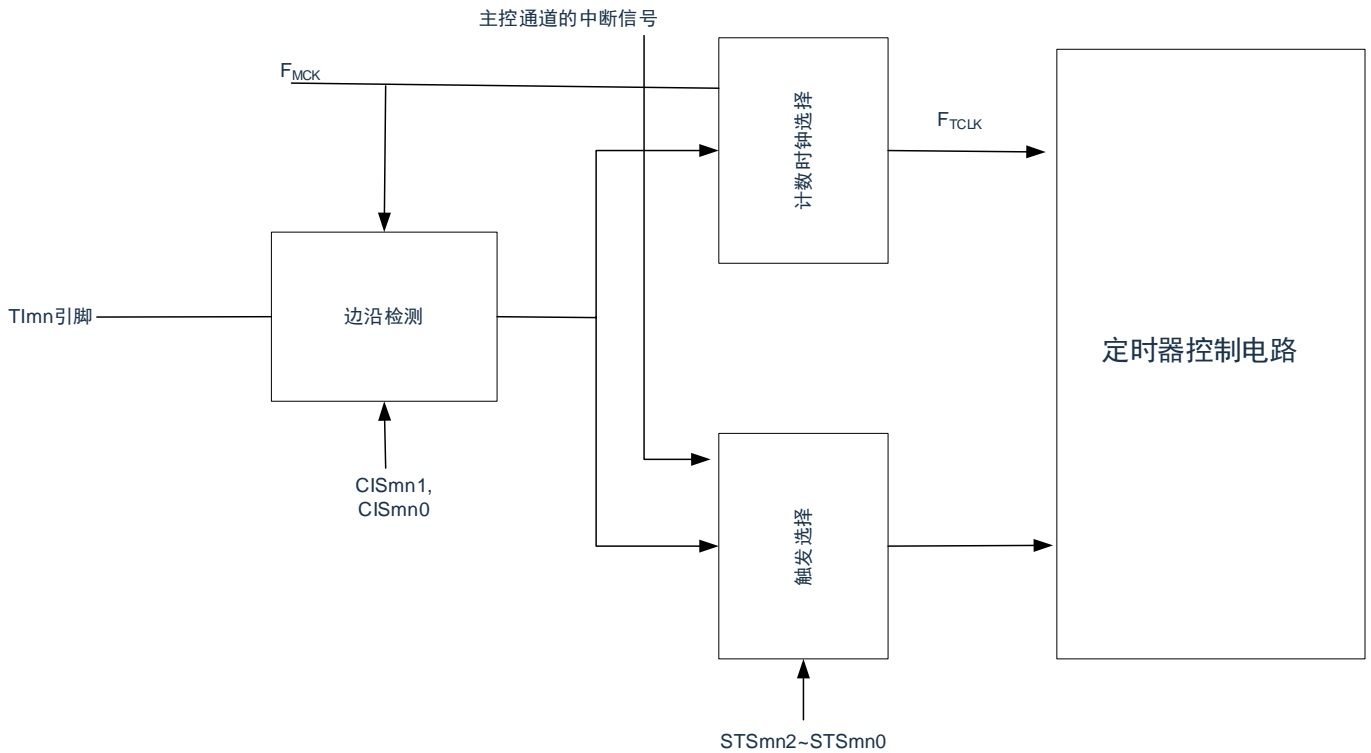
注：m：单元号 (m=0) n：通道号 (n=0~3)

## 5.6 定时器输入（TImn）的控制

### 5.6.1 TImn引脚输入电路的结构

定时器输入引脚的信号通过边沿检测电路输入到定时器控制电路。输入电路的结构如下所示。

图5-17 输入电路的结构



### 5.6.2 操作通道输入时的注意事项

从设定为使用定时器输入引脚到设定定时器输入引脚对应的通道运行允许触发，如果在定时器模式寄存器  $mn$  (TMR $mn$ ) 的 bit12 (CCS $mn$ )、bit9 (STS $mn$ 1) 和 bit8 (STS $mn$ 0) 全都为“0”的状态下将任意一位置位，就必须至少在经过 2 个运行时钟 ( $f_{MCK}$ ) 周期后将定时器通道开始寄存器 (TSM) 的运行允许触发置位。

## 5.7 通用定时器单元的独立通道运行功能

### 5.7.1 作为间隔定时器/方波输出的运行

#### (1) 间隔定时器

能用作以固定间隔产生 INTTMmn（定时器中断）的基准定时器。中断产生周期能用以下计算式进行计算：

$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟周期} \times (\text{TDRmn 的设定值} + 1)$$

#### (2) 作为方波输出的运行

TOmn 在产生INTTMmn 的同时进行交替输出，输出占空比为50% 的方波。

TOmn 输出方波的周期和频率能用以下计算式进行计算：

$$\text{TOmn 输出的方波周期} = \text{计数时钟周期} \times (\text{TDRmn 的设定值} + 1) \times 2$$

$$\text{TOmn 输出的方波频率} = \text{计数时钟频率} / \{(\text{TDRmn 的设定值} + 1) \times 2\}$$

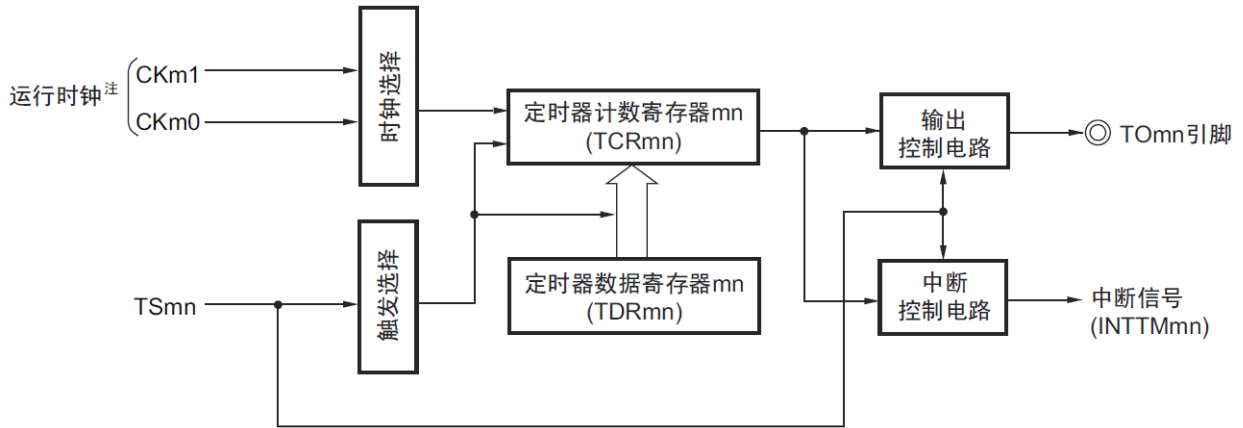
在间隔定时器模式中，定时器计数寄存器mn（TCRmn）用作递减计数器。

在将定时器通道开始寄存器m（TSM）的通道开始触发位（TSMn、TSHm1、TSHm3）置“1”后，通过第1个计数时钟将定时器数据寄存器mn（TDRmn）的值装入TCRmn寄存器。此时，如果定时器模式寄存器n（TMRmn）的MDmn0位为“0”，就不输出INTTMmn并且TOmn也不进行交替输出。如果TMRmn寄存器的MDmn0位为“1”，就输出INTTMmn并且TOmn进行交替输出。然后，TCRmn寄存器通过计数时钟进行递减计数。

如果TCRmn变为“0000H”，就通过下一个计数时钟输出INTTMmn并且TOmn进行交替输出。同时，再次将TDRmn寄存器的值装入TCRmn寄存器。此后，继续同样的运行。

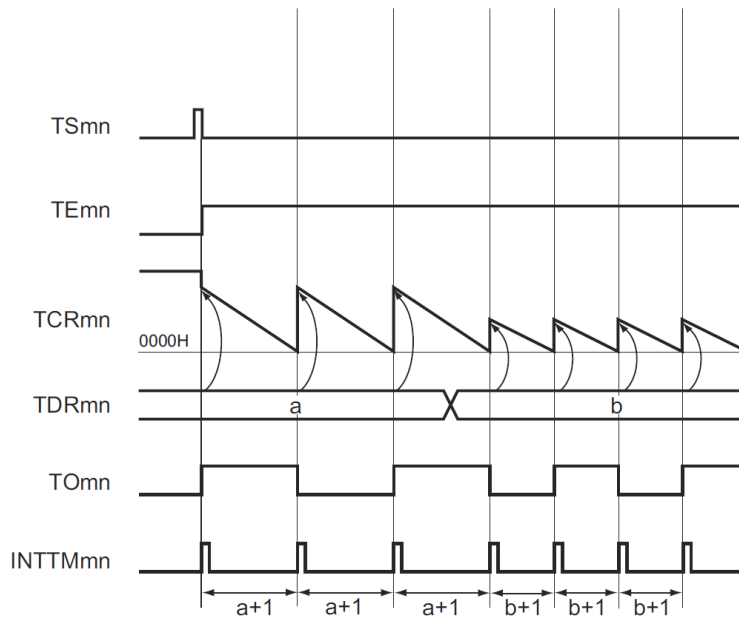
能随时改写TDRmn寄存器，改写的TDRmn寄存器的值从下一个周期开始有效。

图5-18 作为间隔定时器 / 方波输出运行的基本时序例子 (MDmn0=1)



注：在通道1 和通道3 时，能从CKm0、CKm1、CKm2 和CKm3 中选择时钟。

图5-19 作为间隔定时器 / 方波输出运行的基本时序例子 (MDmn0=1)



注1：在通道1 和通道3 时，能从CKm0、CKm1、CKm2 和CKm3 中选择时钟。

注2：TSmn：定时器通道开始寄存器m (TSm) 的bit n

TEMn：定时器通道允许状态寄存器m (TEm) 的bit n

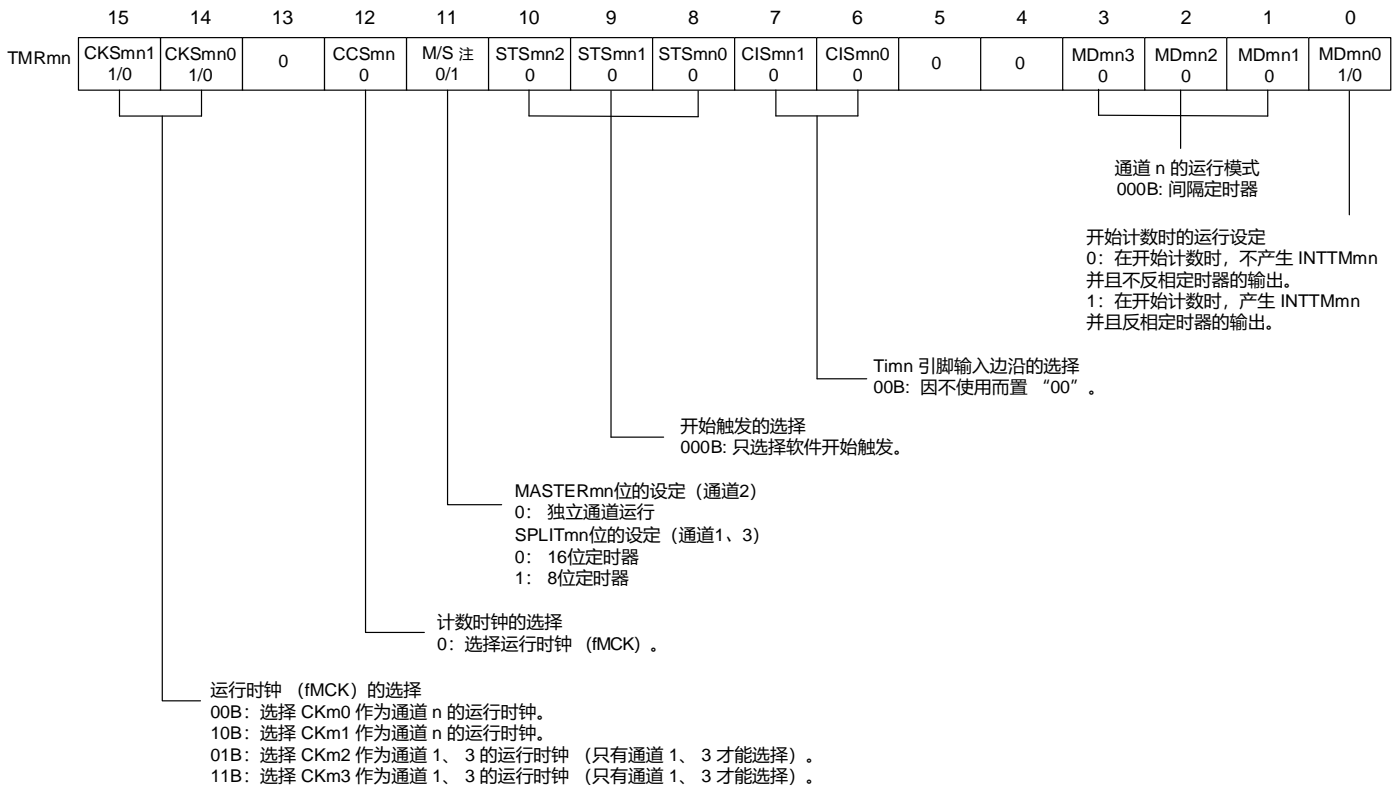
TCRmn：定时器计数寄存器 mn (TCRmn)

TDRmn：定时器数据寄存器 mn (TDRmn)

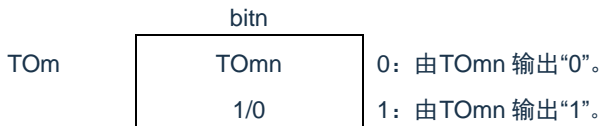
TOMn：TOMn 引脚输出信号

图5-20 间隔定时器 / 方波输出时的寄存器设定内容例子

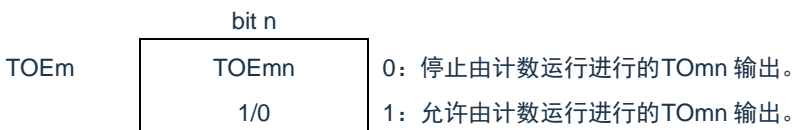
(a) 定时器模式寄存器mn (TMRmn)



(b) 定时器输出寄存器m (TOm)



(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注：TMRm2: MASTERmn 位

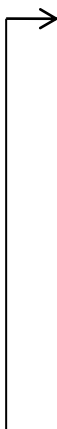
TMRm1、TMRm3: SPLITmn位

TMRm0: 固定为“0”。

m: 单元号 (m= 0) n: 通道号 (n=0~3)

表5-27 间隔定时器 / 方波输出功能时的操作步骤

	软件操作	硬件状态
TIMER4 初始 设定		定时器单元m的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器(PER0)的TM4mEN位置“1”。 →	定时器单元m的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器mn (TMRmn) (确定通道的运行模式, 选择检测边沿)。 给定时器数据寄存mn(TDRmn)设定间隔(周期)值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	将定时器输出模式寄存器0mTOMm) 的TOMmn位置“1” (主控通道输出模式) 将TOLmn置“0”。	TOMn引脚处于Hi-Z输出状态
	设定TOMn位并确定TOMn输出的初始电平。 →	当端口模式寄存器为输出模式并且端口模式控制寄存器为“0”时, 输出TOMn初始设置的电平。
	将TOEmn位置“1”, 允许TOMn输出。 → 将端口模式寄存器和端口模式控制寄存器置“0” →	因为通道处于运行停止状态, 所以TOMn不变。 TOMn输出TOMn设定的电平。
开始运 行	(只在使用 TOMn 输出并且重新开始时, 将 TOEmn位置“1”) 将 TSmn (TSHm1、TSHm3) 位置“1”。 → 因为 TSmn (TSHm1、TSHm3) 位是触发位, 所以自动返回到“0”。	TEmn (TTHm1、TTHm3) 变为“1”并且开始计数。 将TDRmn寄存器的值装入定时计数器mn (TCRmn)。当TMRmn寄存器MDmn0位为“1”时, 产生INTTMmn并且TOMn交替输出。
	能任意更改TDRmn寄存器的设定值。 能随时读TCRmn寄存器。 不使用TSRmn寄存器。 能更改TOM寄存器和TOEm寄存器的设定值。 禁止更改TMRmn寄存器, TOMmn位和TOLmn位的设定值。	计数器 (TCRmn) 进行递减计数。如果计数到“0000H”, 就再次将TDRmn寄存器的值装入TCRmn寄存器并且继续计数。当检测到TCRmn为“0000H”时, 产生INTTMmn并且交替输出。 此后, 重复此运行。
停止运 行	将 TTmn (TTHm1、TTHm3) 位置 “1”。 → 因为 TTmn (TTHm1、TTHm3) 位是触发位, 所以自动返回到“0”。	TEmn (TTHm1、TTHm3) 位变为“0”并且停止计数。 TCRmn寄存器保持计数值而停止计数。 TOMn输出不被初始化而保持状态。
	将TOEmn位置“0”并且给TOMn位设定值。 →	TOMn引脚输出TOMn位设定的电平。
TIMER4 停止	要保持TOMn引脚输出电平的情况: 在给端口寄存器设定要保持的值后将TOMn位置“0”。 → 不需要保持TOMn引脚输出电平的情况: 不需要设定。	通过端口保持TOMn引脚的输出电平。
	将PER0寄存器的TM4mEN位置“0”。 →	定时器单元m的输入时钟处于停止提供状态。 对全部电路和各通道的SFR进行初始化。 (TOMn位变为“0”并且TOMn引脚变为端口功能)

 重新  
开始  
运行


注：m: 单元号 (m= 0) n: 通道号 (n=0~3)



### 5.7.2 作为外部事件计数器的运行

能用作事件计数器，对检测到的 TImn 引脚输入的有效边沿（外部事件）进行计数，如果达到规定的计数值，就产生中断。规定的计数值能用以下计算式进行计算：

$$\text{规定的计数值} = \text{TDRmn 的设定值} + 1$$

在事件计数器模式中，定时器计数寄存器mn（TCRmn）用作递减计数器。

通过将定时器通道开始寄存器m（TSM）的任意通道开始触发位（TSMn、TSHm1、TSHm3）置“1”，将定时器数据寄存器mn（TDRmn）的值装入TCRmn 寄存器。

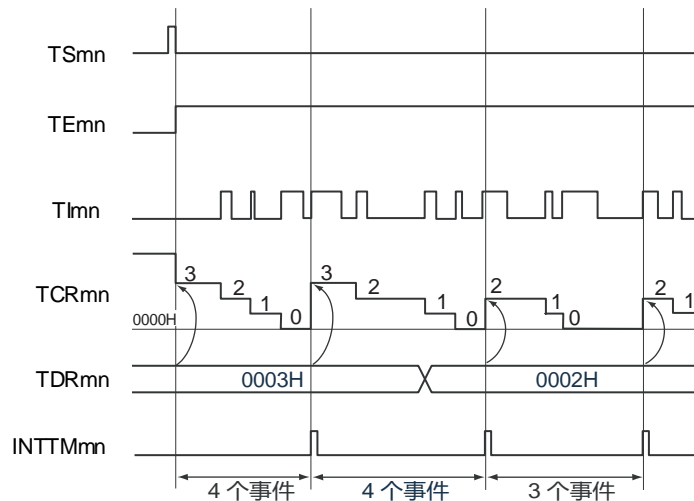
TCRmn 寄存器在检测到 TImn 引脚输入的有效边沿的同时进行递减计数。如果 TCRmn 变为“0000H”，就再次装入TDRmn 寄存器的值并且输出INTTMmn。

此后，继续同样的运行。

因为TOMn 引脚根据外部事件输出不规则的波形，所以必须将定时器输出允许寄存器m（TOEm）的TOEmn 位置“0”，停止输出。

能随时改写TDRmn 寄存器，改写的TDRmn 寄存器的值在下一个计数期间有效。

图5-21 作为外部事件计数器运行的基本时序例子



注：TSMn：定时器通道开始寄存器 m（TSM）的 bit n

TEmn：定时器通道允许状态寄存器 m（TEM）的 bit n

TImn：TImn 引脚输入信号

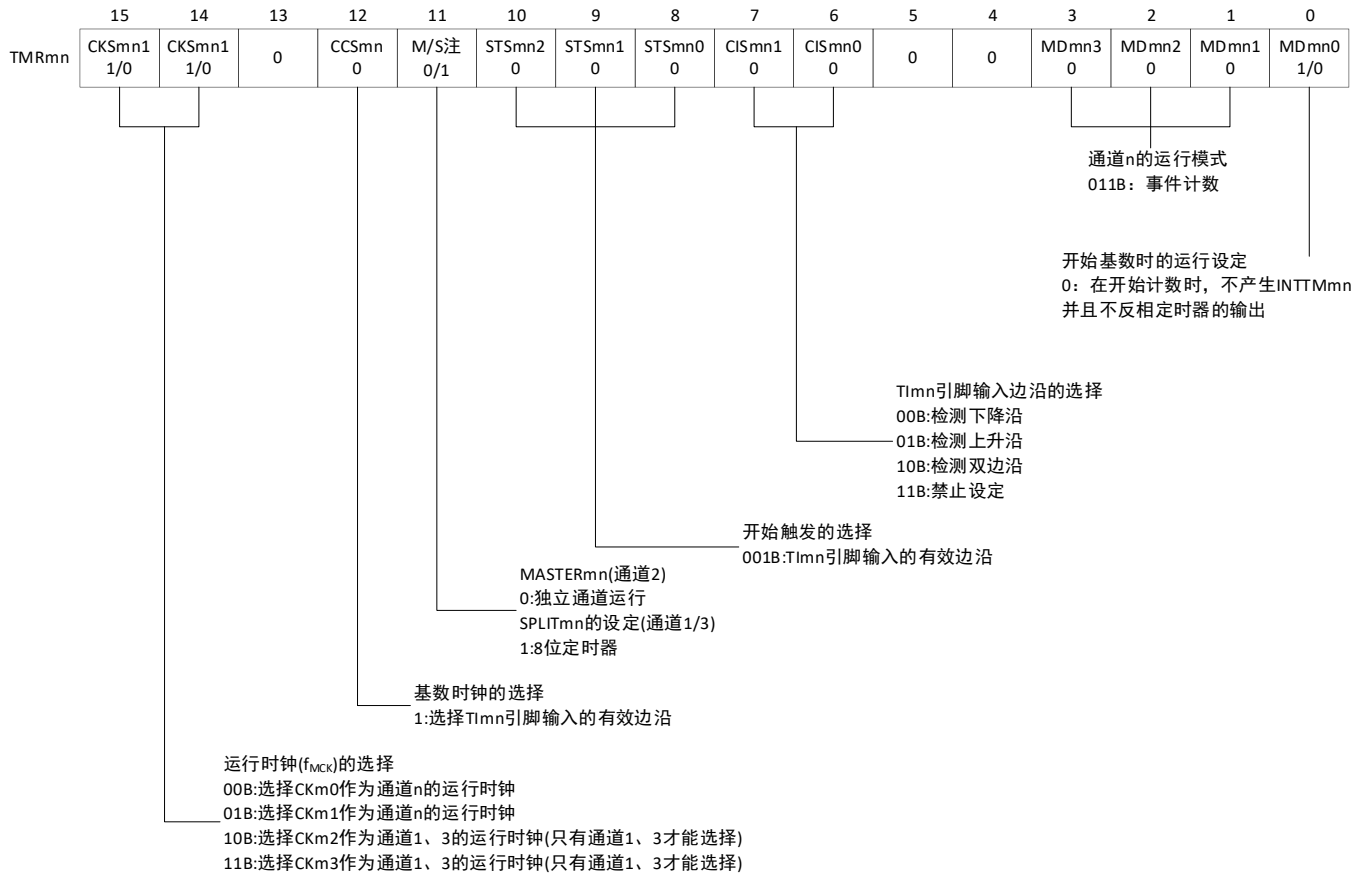
TCRmn：定时器计数寄存器mn（TCRmn）

TDRmn：定时器数据寄存器mn（TDRmn）

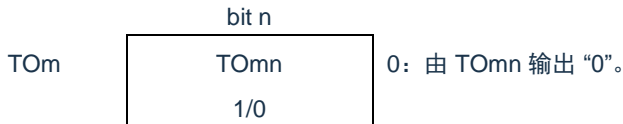
m：单元号（m=0） n：通道号（n=0~3）

图5-22 外部事件计数器模式时的寄存器设定内容例子

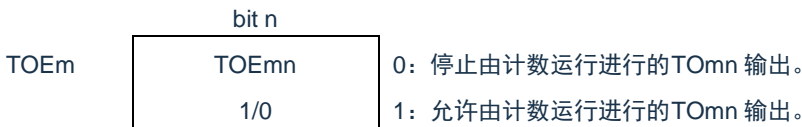
(a) 定时器模式寄存器mn (TMRmn)



(b) 定时器输出寄存器m (TOM)



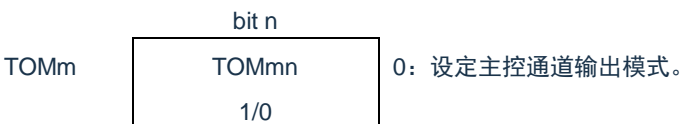
(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注: TMRm2: MASTERmn 位

TMRm1、TMRm3: SPLITmn位

TMRm0: 固定为“0”。

m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

表5-28 外部事件计数器功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元m的输入时钟处于停止提供的状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的TM4mEN 位置 “1”。	定时器单元m的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器mn (TMRmn) (确定通道的运行模式)。 给定定时器数据寄存器mn (TDRmn) 设定计数值。 将定时器输出允许寄存器m (TOEm) 的TOEmn 位置 “0”。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将TSmn 位置 “1”。 因为TSmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “1” 并且开始计数。 将TDRmn 寄存器的值装入定时器计数寄存器mn (TCRmn), 进入TImn 引脚输入边沿的检测等待状态。
重新 开始 运行	能任意更改TDRmn 寄存器的设定值。 能随时读TCRmn 寄存器。 不使用TSRmn 寄存器。 禁止更改TMRmn 寄存器、TOMmn位、TOLmn 位、TOMn 位和TOEmn 位的设定值。	每当检测到TImn 引脚的输入边沿时, 计数器 (TCRmn) 就进行递减计数。如果计数到 “0000H”, 就再次将TDRmn 寄存器的值装入TCRmn 寄存器并且继续计数。当检测到TCRmn 为 “0000H” 时, 产生INTTMmn。 此后, 重复此运行。
停止 运行	将TTmn 位置 “1”。 因为TTmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。
Timer4 停止	将PER0 寄存器的TM4mEN 位置 “0”。	定时器单元m 的输入时钟处于停止提供状态。 对全部电路和各通道的SFR 进行初始化。

注: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

### 5.7.3 作为分频器的运行

能对TI00 引脚输入的时钟进行分频并且用作TO00 引脚输出的分频器。

TO00 输出的分频时钟频率能用以下计算式进行计算：

- 选择上升沿或者下降沿的情况：  
分频时钟频率 = 输入时钟频率 / {(TDR00 的设定值 +1)\*2}
- 选择双边沿的情况：  
分频时钟频率≈输入时钟频率 / (TDR00 的设定值 +1)

在间隔定时器模式中，定时器计数寄存器 00（TCR00）用作递减计数器。

在将定时器通道开始寄存器 0（TS0）的通道开始触发位（TS00）置“1”后，通过检测到 TI00 的有效边沿将定时器数据寄存器 00（TDR00）的值装入 TCR00 寄存器。此时，如果定时器模式寄存器 00（TMR00）的 MD000 位为“0”，就不输出 INTTM00 并 TO00 不进行交替输出；如果 TMR00 寄存器的 MD000 位为“1”，就输出 INTTM00 并且 TO00 进行交替输出。

然后，TCR00 寄存器通过 TI00 引脚输入的有效边沿进行递减计数。如果 TCR00 变为“0000H”，TO00 就进行交替输出。同时，将 TDR00 寄存器的值装入 TCR00 寄存器并且继续计数。

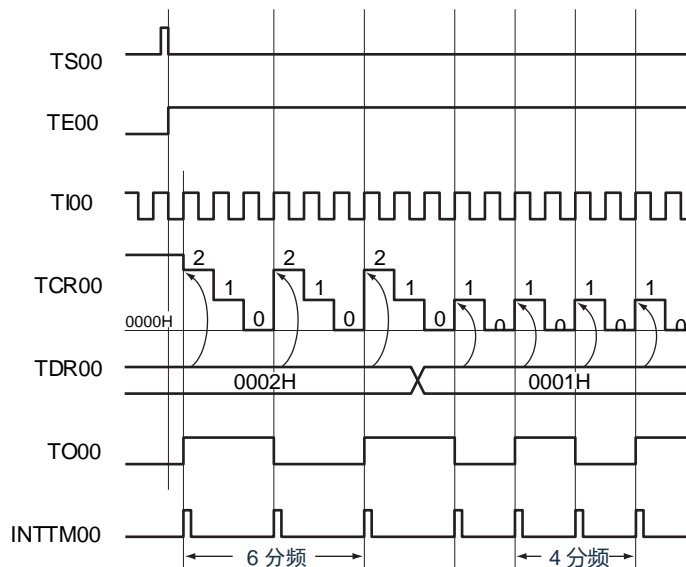
如果选择 TI0 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

TO00 输出的时钟周期包含 1 个运行时钟周期的采样误差。

TO00 输出的时钟周期= 理想的TO00 输出时钟周期± 运行时钟周期（误差）

能随时改写TDR00 寄存器，改写的TDR00 寄存器的值在下一个计数期间有效。

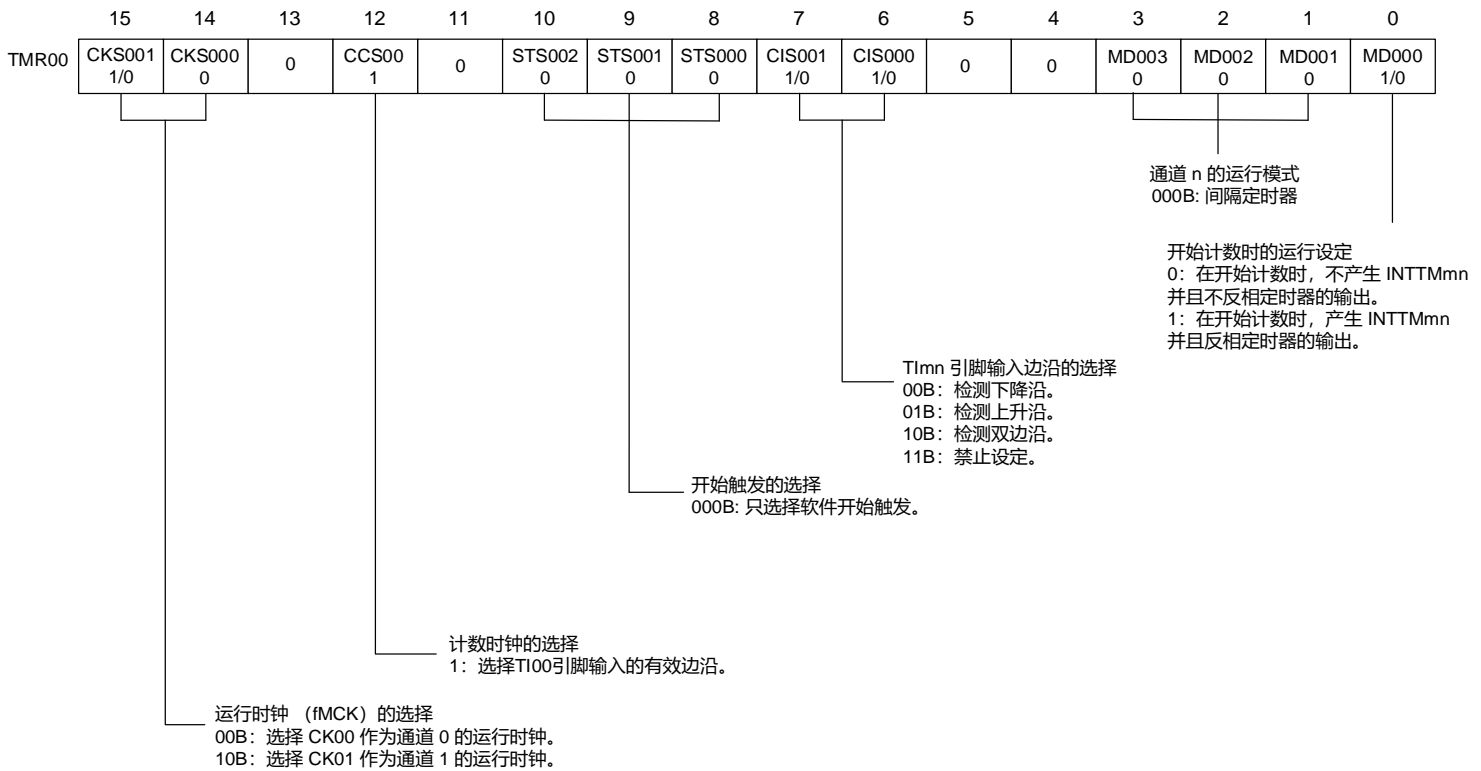
图5-23 作为分频器运行的基本时序例子（MD000=1）



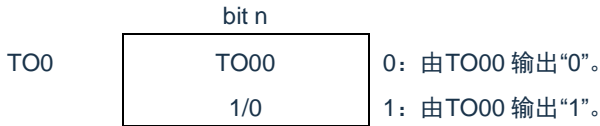
- 注：TS00：定时器通道开始寄存器0（TS0）的bit0  
 TE00：定时器通道允许状态寄存器0（TE0）的bit0  
 TI00：TI00引脚输入信号；TCR00：定时器计数寄存器00（TCR00）  
 TDR00：定时器数据寄存器00（TDR00）；TO00：TO00引脚输出信号

图5-24 作为分频器运行时的寄存器设定内容例子

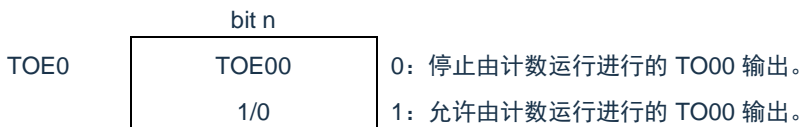
(a) 定时器模式寄存器00 (TMR00)



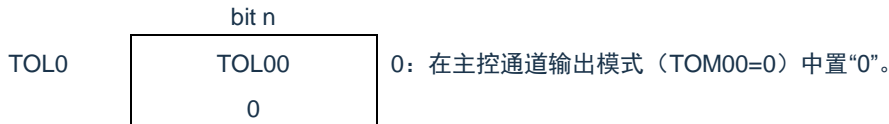
(b) 定时器输出寄存器0 (TO0)



(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器0 (TOL0)



(e) 定时器输出模式寄存器0 (TOM0)

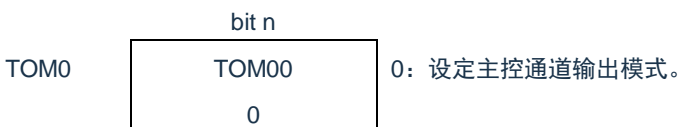


表5-29 分频器功能时的操作步骤

	软件操作	硬件状态
TIMER4 初始 设定		定时器单元0的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器(PER0)的TM4mEN位置“1”。	定时器单元0的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
通道初 始设定	设定定时器时钟选择寄存器0(TPS0)。 确定CK00~CK03的时钟频率。	
	设定定时器模式寄存器00(TMR00)(确定通道的运行模式, 选择检测边沿)。 给定时器数据寄存00(TDR00)设定间隔(周期)值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	将定时器输出模式寄存器0(TOM0)的TOM00位置“1” (主控通道输出模式) 将TOL00置“0”。 设定TO00位并确定TO00输出的初始电平。	TO00引脚处于Hi-Z输出状态  当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出TO00初始设置的电平。 因为通道处于运行停止状态, 所以TO00不变。 TO00输出TO00设定的电平。
	将TOE00位置“1”, 允许TO00输出。 将端口寄存器和端口模式寄存器置“0”	
开始运行	将TOE00位置“1”(只限于重新开始运行)。 将TS00位置“1” 因为TS00是触发位, 所以自动返回到“0”	TE00变为“1”并且开始计数。 将TDR00寄存器的值装入定时计数器00(TCR00)。当TMR00寄存器MD000位为“1”时, 产生INTTM00并且TO00交替输出。
运行中	能任意更改TDR00寄存器的设定值。 能随时读TCR00寄存器。 不使用TSR00寄存器。 能更改TO0寄存器和TOE0寄存器的设定值。 禁止更改TMR00寄存器, TOM00位和TOL00位的设定值。	计数器(TCR00)进行递减计数。如果计数到“0000H”, 就再次将TDR00寄存器的值装入TCR00寄存器并且继续计数。当检测到TCR00为“0000H”时, 产生INTTM00并且交替输出。 此后, 重复此运行。
停止运行	将TT0位置“1”。 因为TT00位是触发位, 所以自动返回到“0”。	TE00位变为“0”并且停止计数。 TCR00寄存器保持计数值而停止计数。 TO00输出不被初始化而保持状态。
	将TOE00位置“0”并且给TO00位设定值。	TO00引脚输出TO00位设定的电平。
TIMER4 停止	要保持TO00引脚输出电平的情况: 在给端口寄存器设定要保持的值后将TO00位置“0”。 不需要保持TO00引脚输出电平的情况: 不需要设定。	通过端口保持TO00引脚的输出电平。
	将PER0寄存器的TM4mEN位置“0”。	定时器单元0的输入时钟处于停止提供状态。 对全部电路和各通道的SFR进行初始化。 (TO00位变为“0”并且TO00引脚变为端口功能)

重新开始运行

注: m: 单元号 (m=0) n: 通道号 (n=0~3)

### 5.7.4 作为输入脉冲间隔测量的运行

能在 TImn 有效边沿捕捉计数值，测量 TImn 输入脉冲的间隔。在 TE<sub>mn</sub> 位为“1”的期间，也能将软件操作 (TS<sub>mn</sub>=1) 设定为捕捉触发，捕捉计数值。

脉冲间隔能用以下计算式进行计算：

$$\text{TImn 输入脉冲间隔} = \text{计数时钟的周期} \times ((10000\text{H} \times \text{TSRmn:OVF}) + (\text{TDRmn 的捕捉值} + 1))$$

注意 因为通过定时器模式寄存器 mn (TMR<sub>mn</sub>) 的 CKS<sub>mn</sub> 位选择的运行时钟对 TImn 引脚输入进行采样，所以产生 1 个运行时钟的误差。

在捕捉模式中，定时器计数寄存器 mn (TCR<sub>mn</sub>) 用作递增计数器。

如果将定时器通道开始寄存器 m (TS<sub>m</sub>) 的通道开始触发位 (TS<sub>mn</sub>) 置“1”，TCR<sub>mn</sub> 寄存器就通过计数时钟从“0000H”开始递增计数。

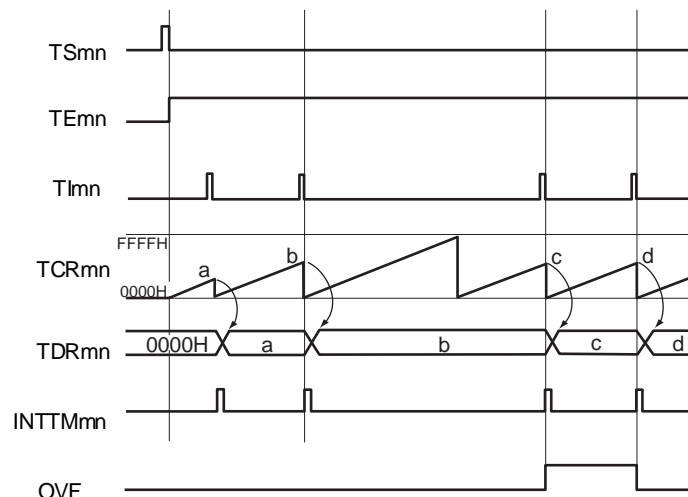
如果检测到 TImn 引脚输入的有效边沿，就将 TCR<sub>mn</sub> 寄存器的计数值传送 (捕捉) 到定时器数据寄存器 mn (TDR<sub>mn</sub>)，同时将 TCR<sub>mn</sub> 寄存器清“0000H”，然后输出 INTT<sub>Mmn</sub>。此时，如果计数器发生上溢，就将定时器状态寄存器 mn (TSR<sub>mn</sub>) 的 OVF 位置“1”。如果计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

在将计数值捕捉到 TDR<sub>mn</sub> 寄存器的同时，根据在测量期间是否发生上溢，更新 TSR<sub>mn</sub> 寄存器的 OVF 位，并且能确认捕捉值的上溢状态。

即使计数器进行了 2 个周期或者 2 个周期以上的完整计数，也认为发生上溢而将 TSR<sub>mn</sub> 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

将 TMR<sub>mn</sub> 寄存器的 STS<sub>mn2</sub> ~ STS<sub>mn0</sub> 位置“001B”，并且将 TImn 的有效边沿用于开始触发和捕捉触发。

图5-25 作为输入脉冲间隔测量的运行基本时序例子 (MD<sub>mn0</sub>=0)



注1: m: 单元号 (m=0) n: 通道号 (n=0~3)

注2: TS<sub>mn</sub>: 定时器通道开始寄存器 m (TS<sub>m</sub>) 的 bit n

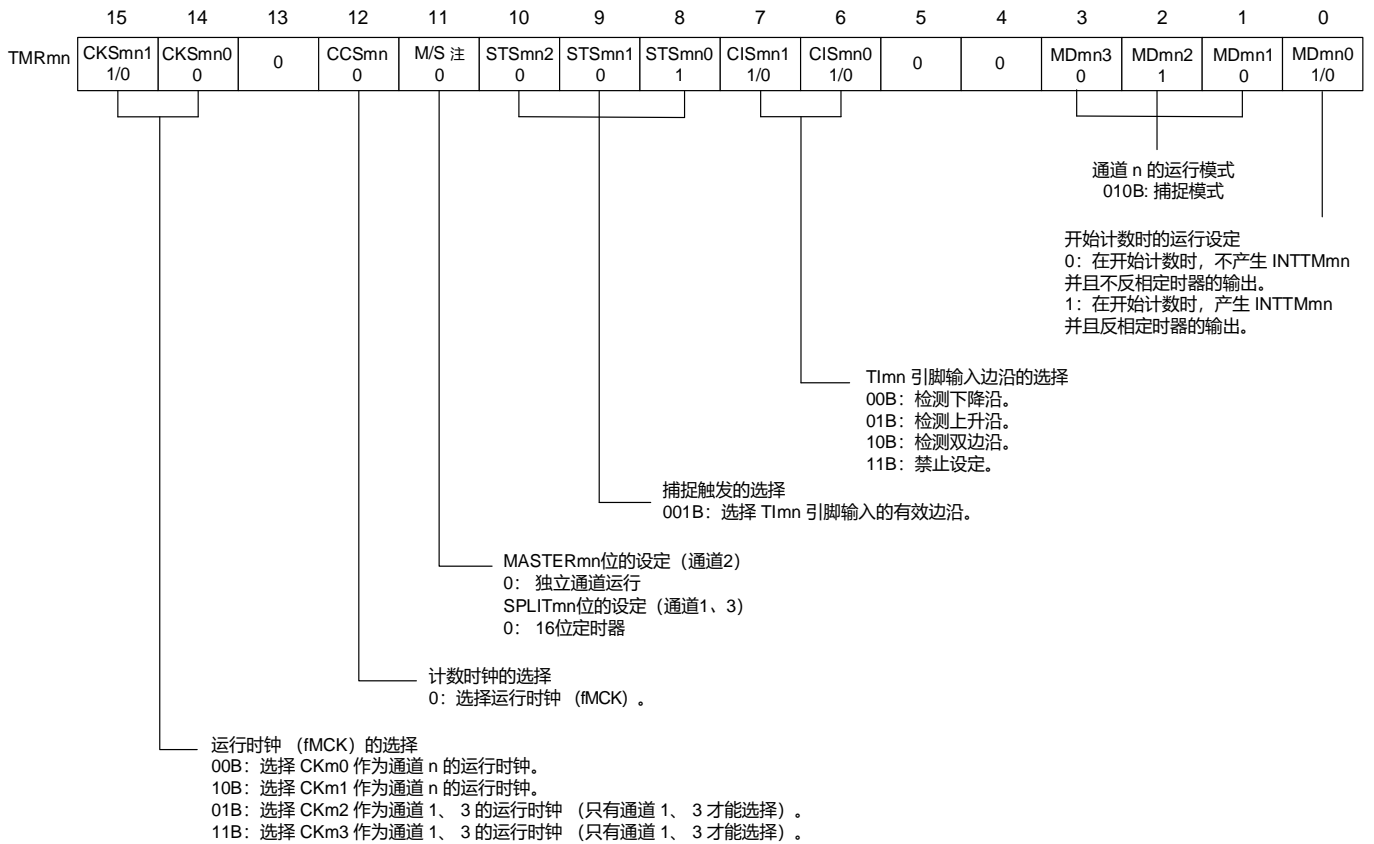
TE<sub>mn</sub>: 定时器通道允许状态寄存器 m (TE<sub>m</sub>) 的 bit n

TImn: TImn 引脚输入信号; TCR<sub>mn</sub>: 定时器计数寄存器 mn (TCR<sub>mn</sub>)

TDR<sub>mn</sub>: 定时器数据寄存器 mn (TDR<sub>mn</sub>); OVF: 定时器状态寄存器 mn (TSR<sub>mn</sub>) 的 bit0

图5-26 测量输入脉冲间隔时的寄存器设定内容例子

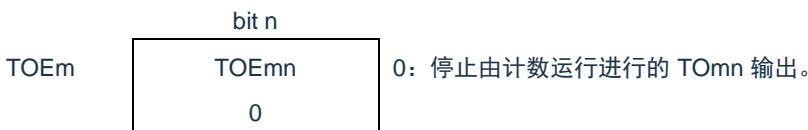
(a) 定时器模式寄存器mn (TMRmn)



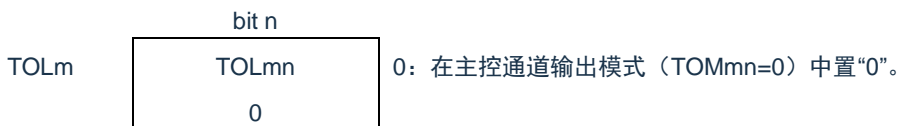
(b) 定时器输出寄存器m (TOM)



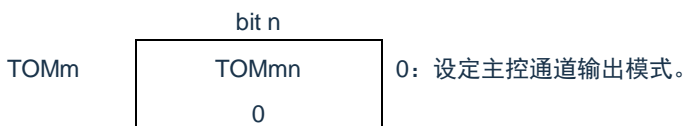
(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)





注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

表5-30 输入脉冲间隔测量功能时的操作步骤

	软件操作	硬件状态
TIMER4 初始 设定	将外围允许寄存器(PER0)的 TM40EN 位置“1”。 →	定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0~CKm3 的时钟频率。	定时器单元 0 的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
通道初 始设定	设定定时器模式寄存器 00 (TMRmn) (确定通道 n 的运行模式)。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
重新 开始 运行	开始 运行	将 TSmn 位置“1”。 → 因为 TSmn 是触发位, 所以自动返回到“0”
	运行中	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。 能随时读 TCRmn 寄存器。 能随时读 TDRmn 寄存器。 能随时读 TSRmn 寄存器。 禁止更改 TOMmn 位, TOLmn 位, TOmn 和 TOEmn 位的设定值。
	停止 运行	将 TTmn 位置“1”。 → 因为 TTmn 位是触发位, 所以自动返回到“0”。
TIMER4 停止	将 PER0 寄存器的 TM40EN 位置“0”。 →	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

注: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

## 5.7.5 作为输入信号高低电平宽度测量的运行

能通过通过 TImn 引脚输入的一个边沿开始计数并且在另一个边沿捕捉计数值，测量 TImn 的信号宽度（高低电平宽度）。TImn 的信号宽度能用以下计算式进行计算。

$$\text{TImn 输入的信号宽度} = \text{计数时钟的周期} \times ((10000H * \text{TSRmn:OVF}) + (\text{TDRmn 的捕捉值} + 1))$$

注意 因为通过定时器模式寄存器 mn (TMRmn) 的 CKSmn 位选择的运行时钟对 TImn 引脚输入进行采样，所以产生 1 个运行时钟的误差。

在捕捉 & 单次计数模式中，定时器计数寄存器 mn (TCRmn) 用作递增计数器。如果将定时器通道开始寄存器 m (TSM) 的通道开始触发位 (TSMn) 置“1”，TEmn 位就变为“1”，并且进入 TImn 引脚的开始边沿检测等待状态。

如果检测到 TImn 引脚输入的开始边沿（在测量高电平宽度时为 TImn 引脚输入的上升沿），就与计数时钟同步，从“0000H”开始递增计数。然后，如果检测到有效捕捉边沿（在测量高电平宽度时为 TImn 引脚输入的下降沿），就在将计数值传送到定时器数据寄存器 mn (TDRmn) 的同时，输出 INTTImn。此时，如果计数器发生上溢，就将定时器状态寄存器 mn (TSRmn) 的 OVF 位置位。如果计数器没有发生上溢，就清除 OVF 位。TCRmn 寄存器的值变为“传送到 TDRmn 寄存器的值 + 1”而停止计数，并且进入 TImn 引脚的开始边沿检测等待状态。此后，继续同样的运行。

在将计数值捕捉到 TDRmn 寄存器的同时，根据在测量期间是否发生上溢，更新 TSRmn 寄存器的 OVF 位，并且能确认捕捉值的上溢状态。

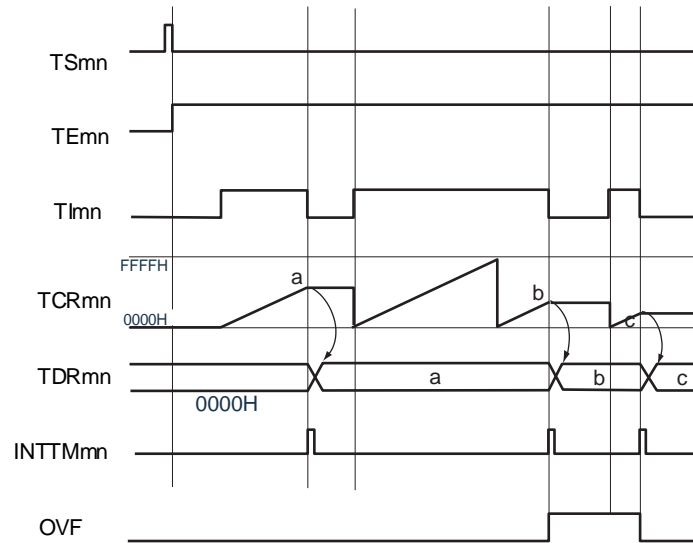
即使计数器进行了 2 个周期或者 2 个周期以上的完整计数，也认为发生上溢而将 TSRmn 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

能通过 TMRmn 寄存器的 CISmn1 位和 CISmn0 位来设定是测量 TImn 引脚的高电平宽度还是低电平宽度。此功能是以测量 TImn 引脚的输入信号宽度为目的，因此不能在 TE mn 位为“1”的期间将 TSMn 位置“1”。

TMRmn 寄存器的 CISmn1、CISmn0=10B：测量低电平宽度。

TMRmn 寄存器的 CISmn1、CISmn0=11B：测量高电平宽度。

图5-27 作为输入信号高低电平宽度测量的运行基本时序例子



注1: m: 单元号 (m=0) n: 通道号 (n=0~3)

注2: TSmn: 定时器通道开始寄存器m (TSm) 的bit n

TE mn: 定时器通道允许状态寄存器m (TEm) 的bit n

TI mn: TI mn 引脚输入信号

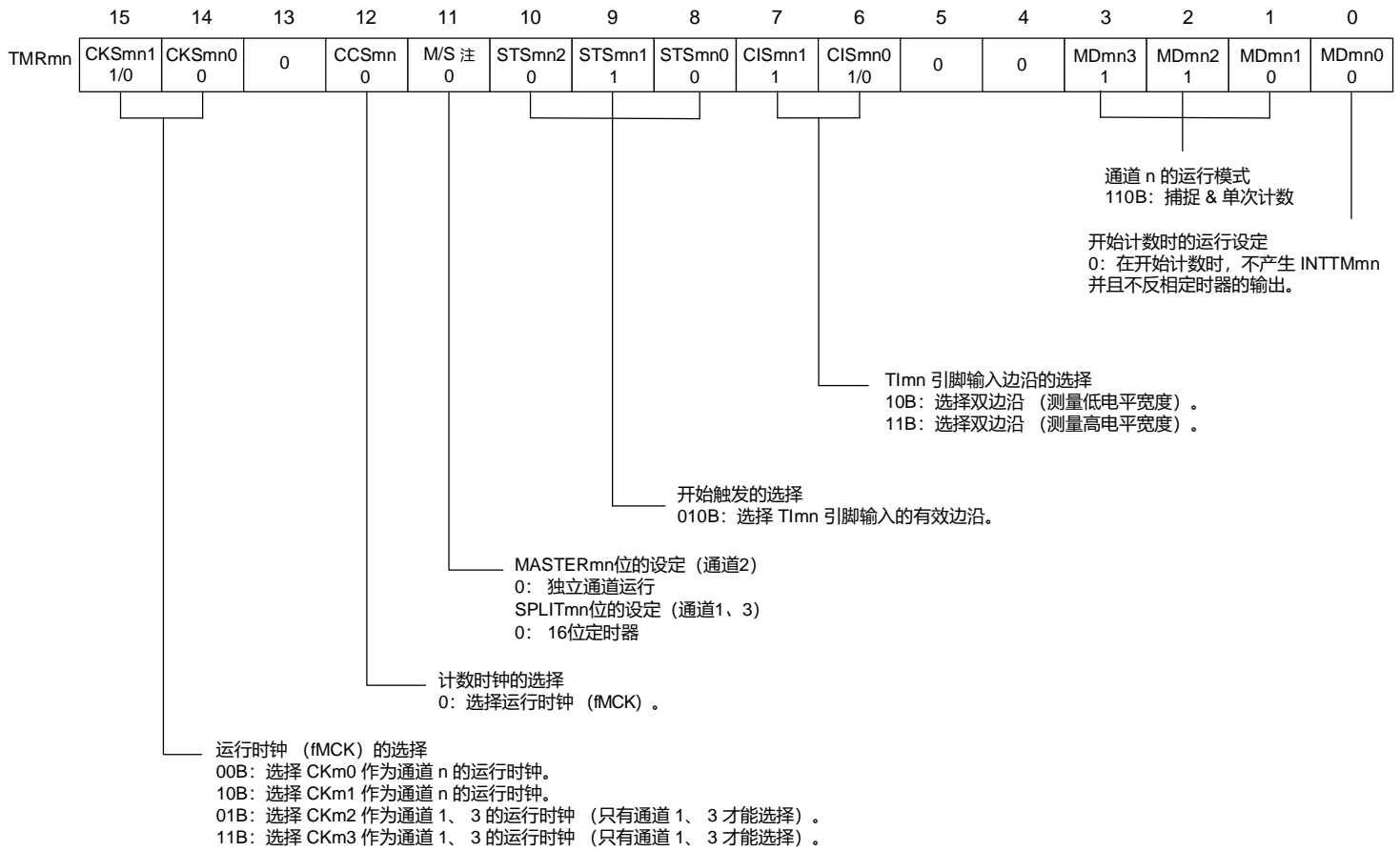
TCRmn : 定时器计数寄存器mn (TCRmn)

TDRmn : 定时器数据寄存器mn (TDRmn)

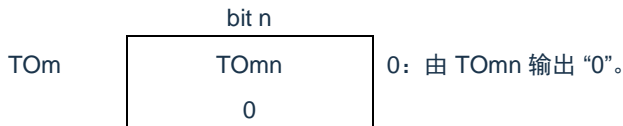
OVF: 定时器状态寄存器mn (TSRmn) 的bit0

图5-28 测量输入信号的高低电平宽度时的寄存器设定内容例子

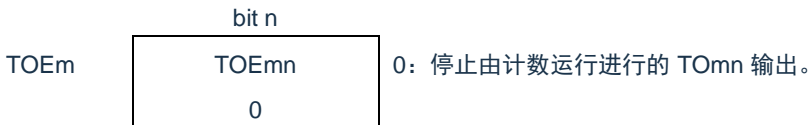
(a) 定时器模式寄存器mn (TMRmn)



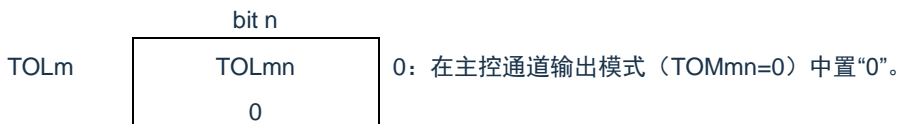
(b) 定时器输出寄存器m (TOM)



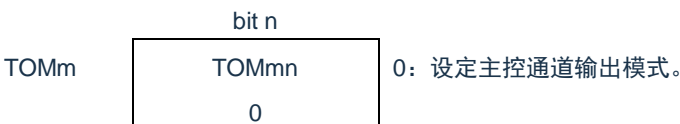
(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

表5-31 输入信号高低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
TIMER4 初始 设定		定时器单元0的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器(PER0)的TM40EN位置“1”。	定时器单元0的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0~CKm3的时钟频率。	
通道初始 设定	设定定时器模式寄存器00 (TMRmn) (确定通道n的运行模式)。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始运行	将TSmn位置“1”。 因为TSmn是触发位, 所以自动返回到“0”。	TEmn变为“1”并开始触发(检测TImn引脚输入的有效边沿或者将TSmn位置“1”)的检测等待状态。
	检测TImn引脚输入的计数开始边沿。	将定时计数器mn (TCRmn) 清“0000H”并且开始递增计数。
运行中	任意更改TDRmn寄存器的设定值。 能随时读TCRmn寄存器。 禁止更改TMRmn寄存器, TOMmn位, TOLmn位, TOmn位和TOEmn位的设定值。	在检测到TImn引脚的开始边沿后, 计数器 (TCRmn) 从“0000H”开始递增计数。如果检测到TImn引脚的捕捉边沿, 就将计数值传送到定时器数据寄存器mn(TDRmn), 并产生INTTmn。此时, 如果发生上溢, 就将定时器状态寄存器mn(TSRmn)的OVF位置位。如果没有发生上溢, 就清除OVF位。TCRmn寄存器在检测到下一个TImn引脚的开始边沿前停止计数。 此后, 重复此运行。
停止运行	将TTmn位置“1”。 因为TTmn位是触发位, 所以自动返回到“0”。	TEmn位变为“0”并且停止计数。 TCRmn寄存器保持计数值而停止计数。 保持TSRmn寄存器的OVF位。
TIMER4 停止	将PER0寄存器的TM40EN位置“0”。	定时器单元0的输入时钟处于停止提供状态。 对全部电路和各通道的SFR进行初始化。

重新开始运行

注: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

### 5.7.6 作为延迟计数器的运行

能通过 TImn 引脚输入的有效边沿检测（外部事件）开始递减计数，并且以任意的设定间隔产生 INTTMmn（定时器中断）。

在 TEmn 位为“1”的期间，能通过软件将 TSmn 位置“1”，开始递减计数，并且以任意的设定间隔产生 INTTMmn（定时器中断）。

中断产生周期能用以下计算式进行计算：

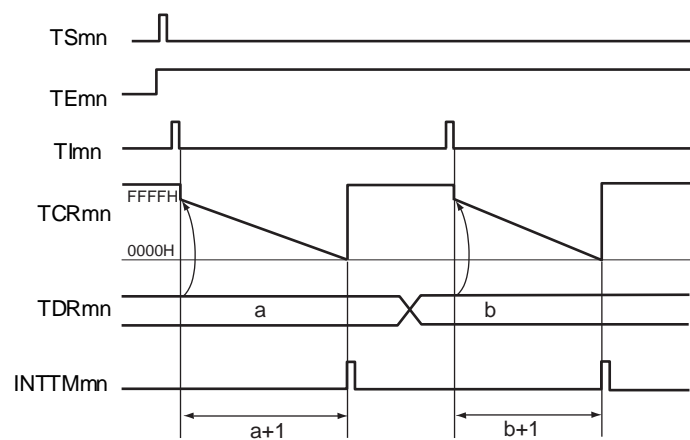
$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDRmn 的设定值} + 1)$$

在单次计数模式中，定时器计数寄存器mn（TCRmn）用作递减计数器。

如果将定时器通道开始寄存器m（TSm）的通道开始触发位TSmn、TSHm1、TSHm3）置“1”，TEmn 位、TEHm1 位和TEHm3 位就变为“1”，并且进入TImn 引脚的有效边沿检测等待状态。通过 TImn 引脚输入的有效边沿检测，开始 TCRmn 寄存器的运行，并且装入定时器数据寄存器 mn（TDRmn）的值。TCRmn 寄存器通过计数时钟，从装入的 TDRmn 寄存器的值开始递减计数。如果TCRmn 变为“0000H”，就输出INTTMmn，并且在检测到下一个TImn 引脚输入的有效边沿前停止计数。

能随时改写TDRmn 寄存器，改写的TDRmn 寄存器的值从下一个周期开始有效。

图5-29作为延迟计数器的运行基本时序例子



注1: m: 单元号 (m=0) n: 通道号 (n=0~3)

注2: TSmn: 定时器通道开始寄存器m (TSm) 的bit n

TEmn: 定时器通道允许状态寄存器m (TEm) 的bit n

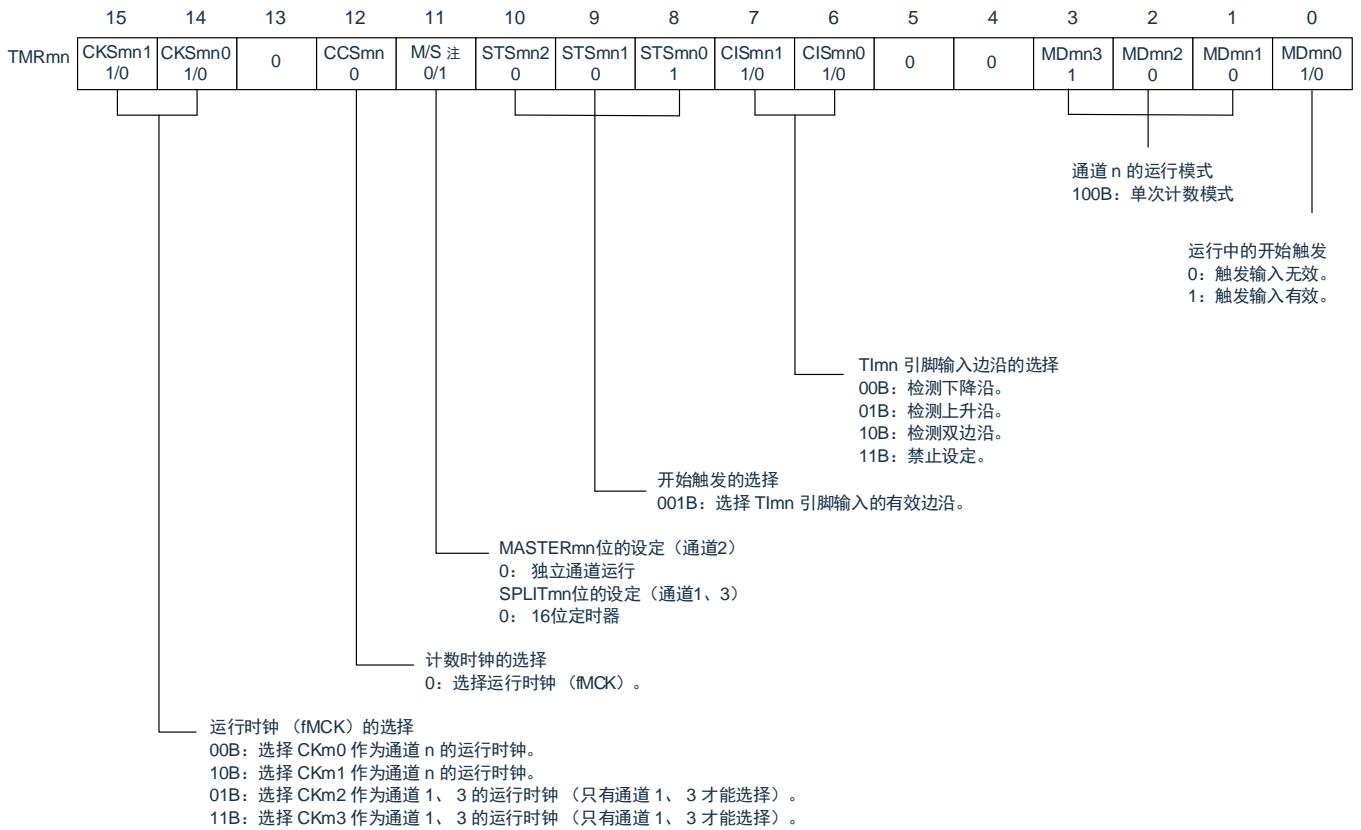
TImn: TImn 引脚输入信号

TCRmn: 定时器计数寄存器mn (TCRmn)

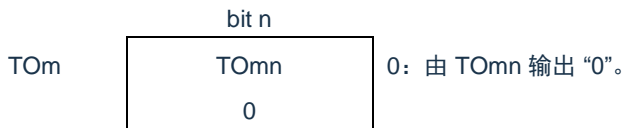
TDRmn: 定时器数据寄存器mn (TDRmn)

图5-30 延迟计数器功能时的寄存器设定内容例子

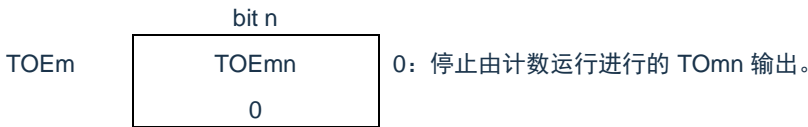
(a) 定时器模式寄存器mn (TMRmn)



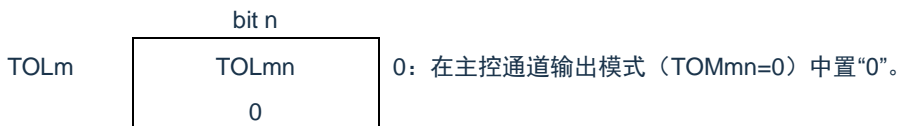
(b) 定时器输出寄存器m (TOM)



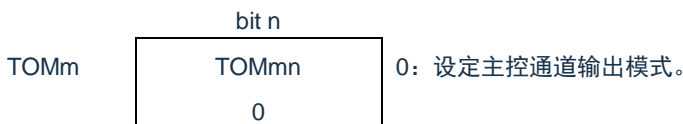
(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)

表5-32 延迟计数器功能时的操作步骤

	软件操作	硬件状态
TIMER4 初始 设定	将外围允许寄存器(PER0)的 TM40EN 位置“1”。 →	定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
		定时器单元 0 的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0~CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器 00 (TMRmn) (确定通道 n 的运行模式)。 给定时器数据寄存器 mn (TDRmn) 设定输出延时时间。 将 TOEmn 位置 “0” 并且停止 TOmn 的运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置“1”。 → 因为 TSmn 是触发位, 所以自动返回到“0”。	TEmn 变为“1”并开始触发 (检测 TIMn 引脚输入的有效边沿或者将 TSmn 位置 “1”) 的检测等待状态。
	通过检测到以下一个开始触发, 开始递减计数。 检测到 TIMn 引脚输入的有效边沿。 通过软件将 TSmn 位置 “1”。 →	将 TDR 寄存器的值装入定时计数寄存器 mn(TCRmn)。
运行中	任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。	计数器 (TCRmn) 开始递减计数。如果 TCRmn 计数到 “0000H”, 就产生 INTTMmn。并且在检测到下一次触发 (检测 TIMn 引脚输入的有效边沿或者将 TSmn 位置 “1”) 前, TCRmn 为 “0000H” 并停止计数。 此后, 重复此运行。
停止 运行	将 TTmn 位置“1”。 → 因为 TTmn 位是触发位, 所以自动返回到“0”。	TEmn 位变为“0”并且停止计数。 TCRmn 寄存器保持计数值而停止计数。
Timer4 停止	将 PER0 寄存器的 TM40EN 位置“0”。 →	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

重新开始运行

注: m: 单元号 (m= 0) n: 通道号 (n=0 ~3)



## 5.8 通用定时器单元的多通道联动运行功能

### 5.8.1 作为单触发脉冲输出功能的运行

将2个通道成对使用，能通过TImn 引脚的输入生成任意延迟脉宽的单触发脉冲。延迟和脉宽能用以下计算式进行计算：

$$\begin{aligned} \text{延迟} &= \{ \text{TDRmn (主控) 的设定值} + 2 \} \times \text{计数时钟周期} \\ \text{脉宽} &= \{ \text{TDRmp (从属) 的设定值} \} \times \text{计数时钟周期} \end{aligned}$$

在单次计数模式中，主控通道运行并且对延迟进行计数。通过检测开始触发，主控通道的定时器计数寄存器mn (TCRmn) 开始运行并且装入定时器数据寄存器mn (TDRmn) 的值。TCRmn 寄存器通过计数时钟，从装入的TDRmn 寄存器的值开始递减计数。如果TCRmn 变为“0000H”，就输出INTTMmn，并且在检测到下一个开始触发前停止计数。

在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的 INTTMmn 作为开始触发，从属通道的TCRmp 寄存器开始运行并且装入TDRmp 寄存器的值。TCRmp 寄存器通过计数时钟，从装入的TDRmp 寄存器值开始递减计数。如果计数值变为“0000H”，就输出INTTMmp，并且在检测到下一个开始触发（主控通道的INTTMmn）前停止计数。在从主控通道产生INTTMmn 并且经过1个计数时钟后，TOmp 的输出电平变为有效电平，如果TCRmp 变为“0000H”，就变为无效电平。

不使用TImn 引脚输入也能将软件操作 (TSmn=1) 作为开始触发来输出单触发脉冲。

注意 因为主控通道的TDRmn 寄存器和从属通道的TDRmp 寄存器的装入时序不同，所以如果在计数过程中改写TDRmn 寄存器和TDRmp 寄存器，就可能与装入时序发生竞争，输出不正常的波形。必须在产生INTTMmn 后改写TDRmn 寄存器，并且在产生INTTMmp 后改写TDRmp 寄存器。

注：m：单元号 (m=0) n：主控通道号 (n=0、2) p：从属通道号 (n=0：p=1、2、3, n=2：p=3)

图5-31 作为单触发脉冲输出功能运行的框图

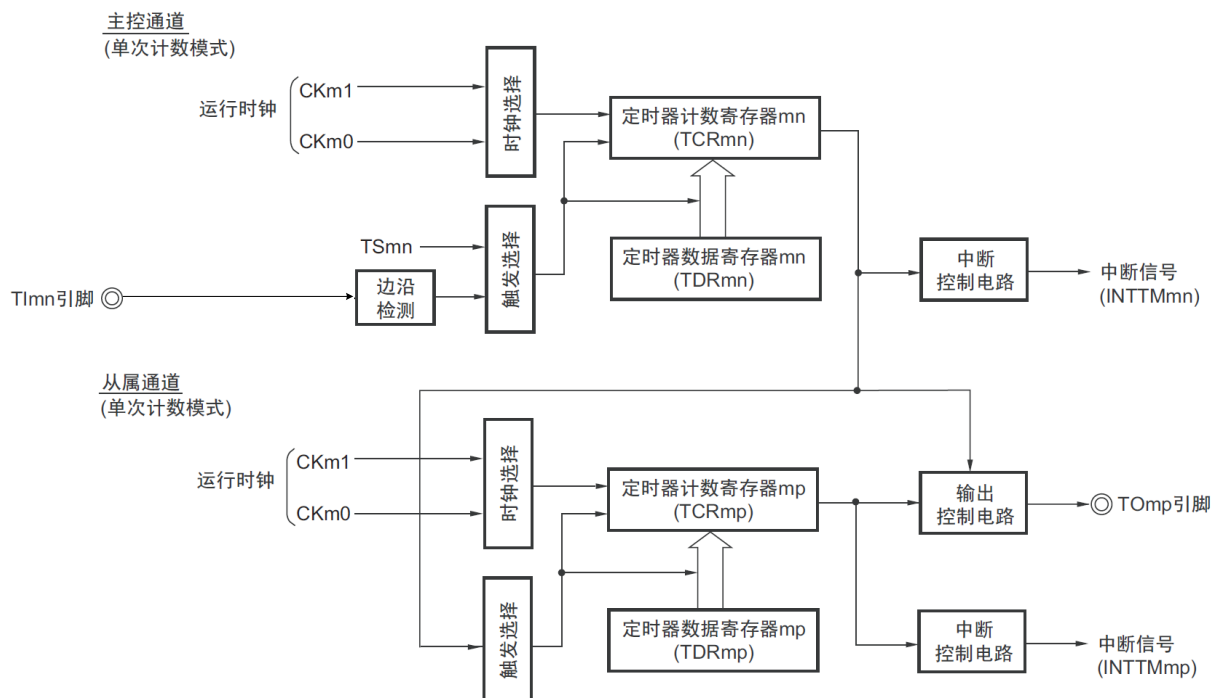
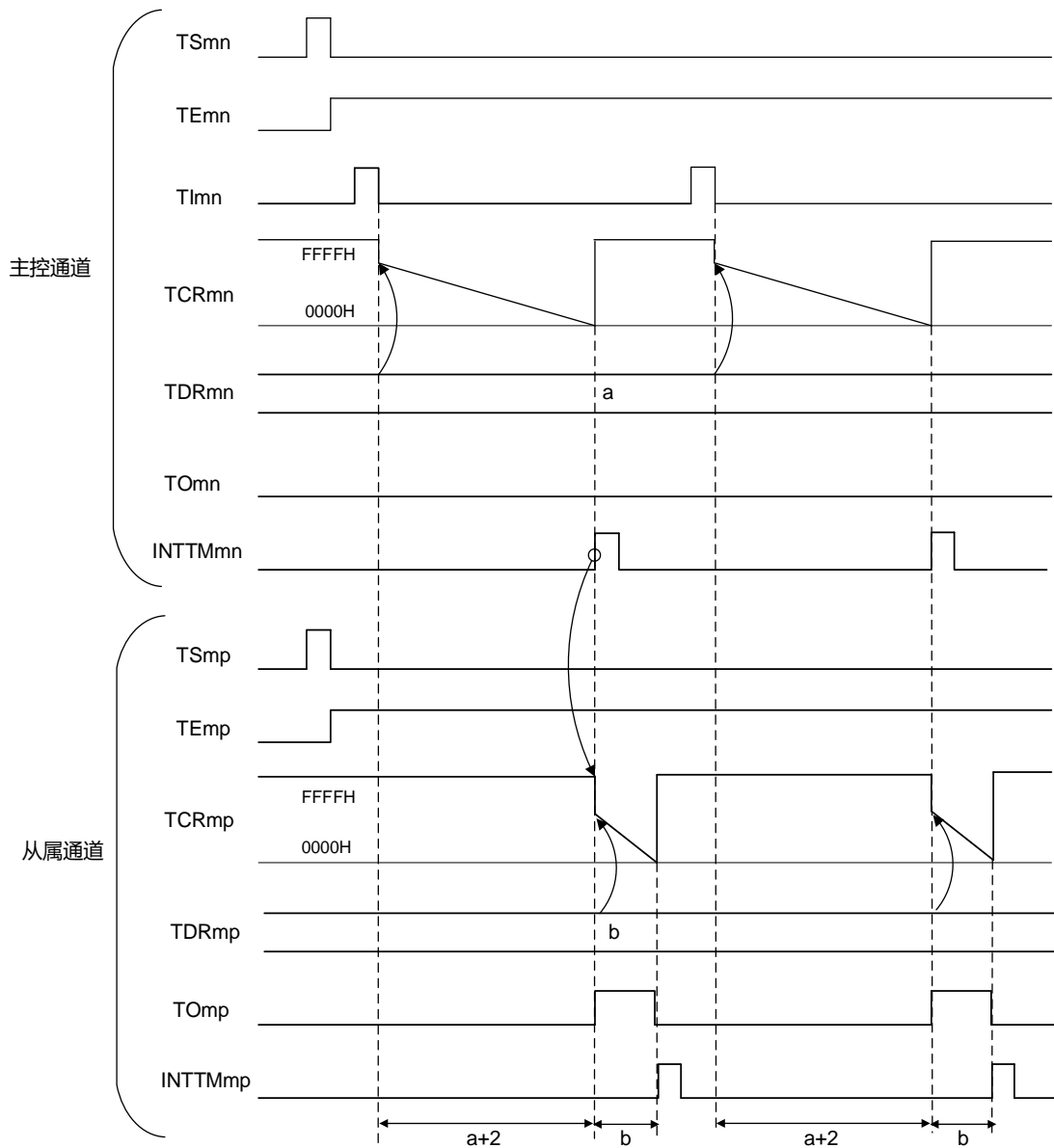


图5-32 作为单触发脉冲输出功能的运行基本时序例子



注1: m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

注2: TSmn、TSmp: 定时器通道开始寄存器m (TSm) 的bit n、p

TE mn、TE mp: 定时器通道允许状态寄存器m (TE m) 的bit n、p

TImn、TImp: TImn 引脚和TImp 引脚的输入信号

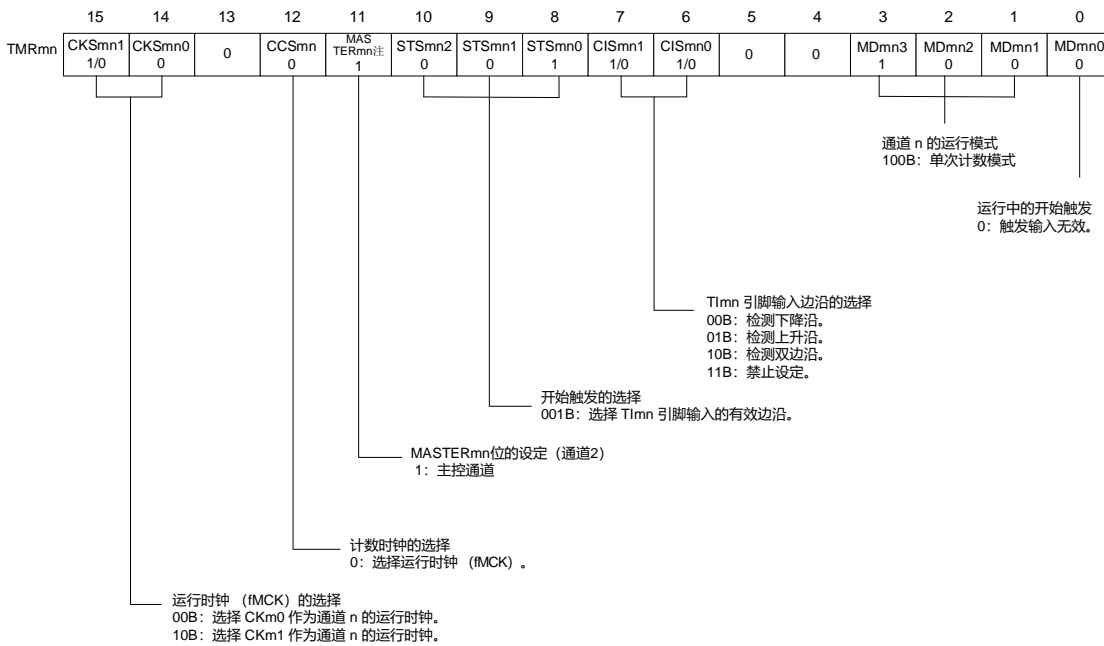
TCRmn、TCRmp : 定时器计数寄存器mn、mp (TCRmn、TCRmp)

TDRmn、TDRmp : 定时器数据寄存器mn、mp (TDRmn、TDRmp)

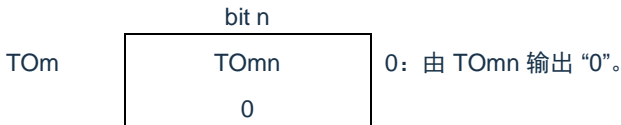
TOmn、TOmp: TOmn 引脚和TOmp 引脚的输出信号

图5-33 单触发脉冲输出功能时（主控通道）的寄存器设定内容例子

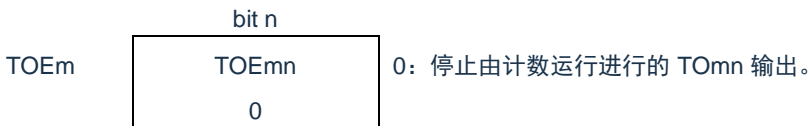
(a) 定时器模式寄存器mn（TMRmn）



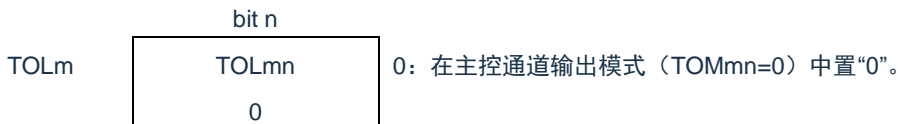
(b) 定时器输出寄存器m（TOMm）



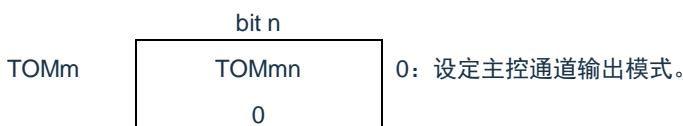
(c) 定时器输出允许寄存器m（TOEm）



(d) 定时器输出电平寄存器m（TOLm）



(e) 定时器输出模式寄存器m（TOMm）

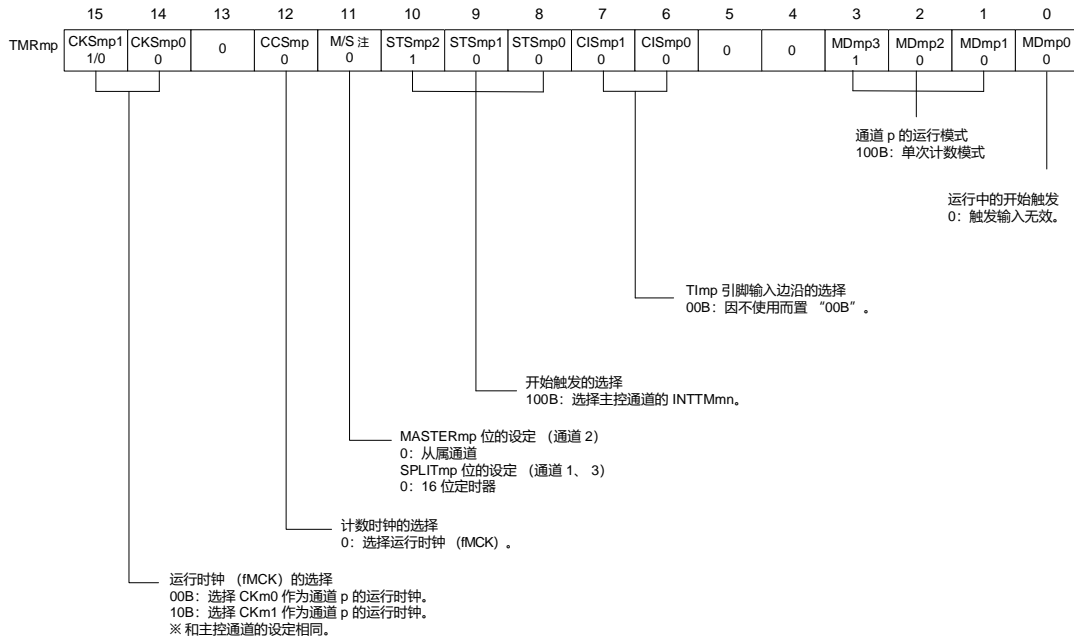


注1: TMRm2: MASTERmn 位  
 TMRm1、TMRm3 : SPLITmn位  
 TMRm0: 固定为“0”。

注2: m: 单元号 (m = 0) n: 通道号 (n=0 ~2)

图5-34 单触发脉冲输出功能时（从属通道）的寄存器设定内容例子

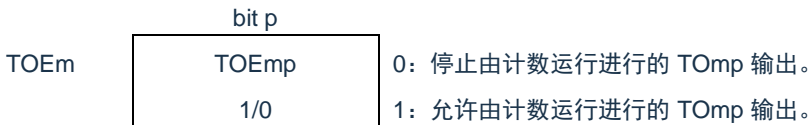
(a) 定时器模式寄存器mp（TMRmp）



(b) 定时器输出寄存器m（TOm）



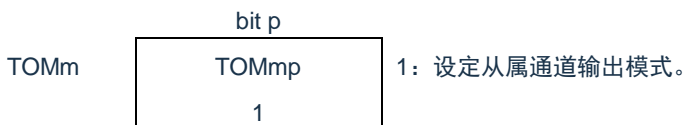
(c) 定时器输出允许寄存器m（TOEm）



(d) 定时器输出电平寄存器m（TOLm）



(e) 定时器输出模式寄存器m（TOMm）




注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m=0) n: 通道号 (n=0~2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

表5-33 单触发脉冲输出功能时的操作步骤(1/2)

	软件操作	硬件状态
TIMER4 初始 设定		定时器单元0的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器(PER0)的TM40EN位置“1” 	定时器单元0的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0~CKm3的时钟频率。	
通道初始 设定	设定使用的2个通道的定时器模式寄存器mn, mp (TMRmn, TMRmp) (确定通道的运行模式)。 给主控通道的定时器数据寄存器mn (TDRmn) 设定输出 延迟时间, 并且给从属通道的TDRmp寄存器设定脉宽。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器mn (TOMmn) 的TOMmp位置 “1” (从属通道输出模式)。 设定TOLmp位。 设定TOmp位并且确定TOmp输出的初始电平。	TOmp引脚处于Hi-Z输出状态。
	将TOEmp位置“1”, 允许TOmp输出。 将端口寄存器和端口模式寄存器置“0”。	当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出TOmp初始设定的电平。 因为通道处于运行停止状态, 所以TOmp不变。 TOmp引脚输出TOmp设定的电平。

5-33 单触发脉冲输出功能时的操作步骤(2/2)

重新开始运行	开始运行	将TOEmp位(从属)置“1”(只限于重新开始运行)。因为TSmn是触发位,所以自动返回到“0”。将定时器通道开始寄存器m(TSm)的TSmn(主控)道的TSmn位置“1”。 因为TSmn位和TSmp位是触发位,所以自动返回到“0”。 通过检测主控通道的开始触发,开始主控通道的计数。检测到Tlmn引脚输入的有效边沿。通过软件将TSmn位置“1”。	TEMn位和TEmp位都变为“1”,主控通道进入开始。触发(检测Tlmn引脚输入的有效边沿或者将主控通道的TSmn位置“1”)的检测等待状态。计数器还处于停止状态。
	运行中	只能更改TMRmn寄存器的CISmn1位和CISmn0位的设定值。 禁止更改TMRmp、TDRmn、TDRmp寄存器以及TOMmn位、TOMmp位、TOLmn位和TOLmp位的设定值。 能随时读TCRmn寄存器和TCRmp寄存器。 不使用TSRmn寄存器和TSRmp寄存器。 能更改从属通道的TOM寄存器 and TOEm寄存器的设定值。	主控通道通过检测开始触发(检测Tlmn引脚输入到的有效边沿或者将主控通道的TSmn位置“1”),将TDRmn寄存器的值装入定时器计数寄存器mn(TCRmn),并且进行递减计数。如果TCRmn计数到“0000H”,就产生INTTMmn,并且在下一次Tlmn引脚输入前停止计数。 从属通道以主控通道的INTTMmn为触发,将TDRmp寄存器的值装入TCRmp寄存器并且计数器开始递减计数。在主控通道输出INTTMmn并且经过一个计数时钟后,将TOmp的输出电平置为有效电平。然后,如果TCRmp计数到“0000H”,就在将TOmp的输出电平置为无效电平后停止计数。 此后,重复此运行。
	停止运行	将TTmn(主控)和TTmp位(从控)同时置“1”。因为TTmn位和TTmp位是触发位,所以自动返回到“0”。 将从属通道的TOEmp位置“0”并且给TOmps设定值。	TEMn位和TEmp位都变为“0”并且停止计数。TCRmn寄存器和TCRmp寄存器保持计数值而停止计数。TOmp输出不被初始化而保持状态。
	Timer4 停止	要保持TOmp引脚输出电平的情况: 在端口寄存器设置要保持的值后将TOmp位置“0”。 不需要保持TOmp引脚输出电平的情况: 不需要设定。 将PER0寄存器的TM40EN位置“0”。	TOmpy引脚输出TOmp设定的电平。 通过端口功能保持TOmp引脚的输出电平。 定时器单元0的输入时钟处于停止提供状态。对全部电路和各通道的SFR进行初始化。

注1: m: 单元号 (m=0) n: 主控通道号 (n=0)

p: 从属通道号 q: 从属通道号

 $n < p < q \leq 3$  (p和q是大于n的整数)

注2: 不能将从属通道的TSmn位置1

## 5.8.2 作为 PWM 功能的运行

将 2 个通道成对使用，能生成任意周期和占空比的脉冲。输出脉冲的周期和占空比能用以下计算式进行计算：

$$\begin{aligned} \text{脉冲周期} &= \{ \text{TDRmn (主控) 的设定值} + 1 \} * \text{计数时钟周期} \\ \text{占空比}[\%] &= \{ \text{TDRmp (从属) 的设定值} \} / \{ \text{TDRmn (主控) 的设定值} + 1 \} * 100 \\ 0\% \text{ 输出} &: \text{TDRmp (从属) 的设定值} = 0000\text{H} \\ 100\% \text{ 输出} &: \text{TDRmp (从属) 的设定值} \geq \{ \text{TDRmn (主控) 的设定值} + 1 \} \end{aligned}$$

备注 当 TDRmp（从属）的设定值 > {TDRmn（主控）的设定值+1} 时，占空比超过 100%，但是为 100% 输出。

主控通道用作间隔定时器模式。如果将定时器通道开始寄存器 m（TSm）的通道开始触发位（TSmn）置“1”，就输出中断（INTTMmn），然后将定时器数据寄存器 mn（TDRmn）的设定值装入定时器计数寄存器 mn（TCRmn），并且通过计数时钟进行递减计数。当计数到“0000H”时，在输出 INTTMmn 后再次将 TDRmn 寄存器的值装入 TCRmn 寄存器，并且进行递减计数。此后，在将定时器通道停止寄存器 m（TTm）的通道停止触发位（TTmn）置“1”前，重复此运行。

当用作 PWM 功能时，主控通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出（TOmp）周期。从属通道用作单次计数模式。以主控通道的 INTTMmn 为开始触发，将 TDRmp 寄存器的值装入 TCRmp 寄存器，并且进行递减计数，计数到“0000H”为止。当计数到“0000H”时，输出 INTTMmp，并且等待下一个开始触发（主控通道的 INTTMmn）。

当用作 PWM 功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出（TOmp）的占空比。

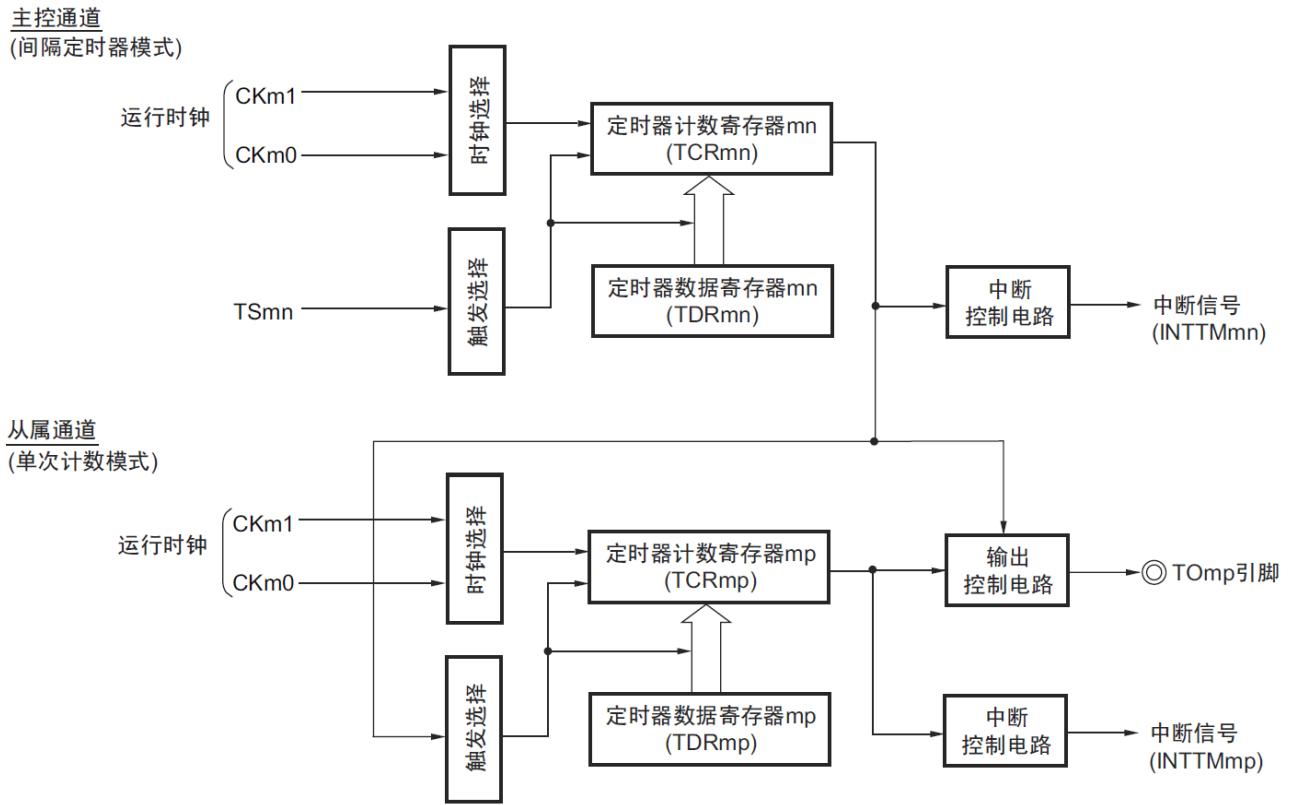
在从主控通道产生 INTTMmn 并且经过 1 个时钟后，PWM 输出（TOmp）变为有效电平，并且在从属通道的 TCRmp 寄存器的值为“0000H”时变为无效电平。

注意 要同时改写主控通道的定时器数据寄存器 mn（TDRmn）和从属通道的 TDRmp 寄存器时，需要 2 次写存取。因为在主控通道产生 INTTMmn 时将 TDRmn 寄存器和 TDRmp 寄存器的值装入 TCRmn 寄存器和 TCRmp 寄存器，所以如果分别在主控通道产生 INTTMmn 前后进行改写，TOmp 引脚就不能输出期待的波形。因此，要同时改写主控的 TDRmn 寄存器和从属的 TDRmp 寄存器时，必须在主控通道产生 INTTMmn 后立即改写这 2 个寄存器。

注：m：单元号（m=0），n：主控通道号（n=0、2）

p：从属通道号（n=0：p=1、2、3，n=2：p=3）

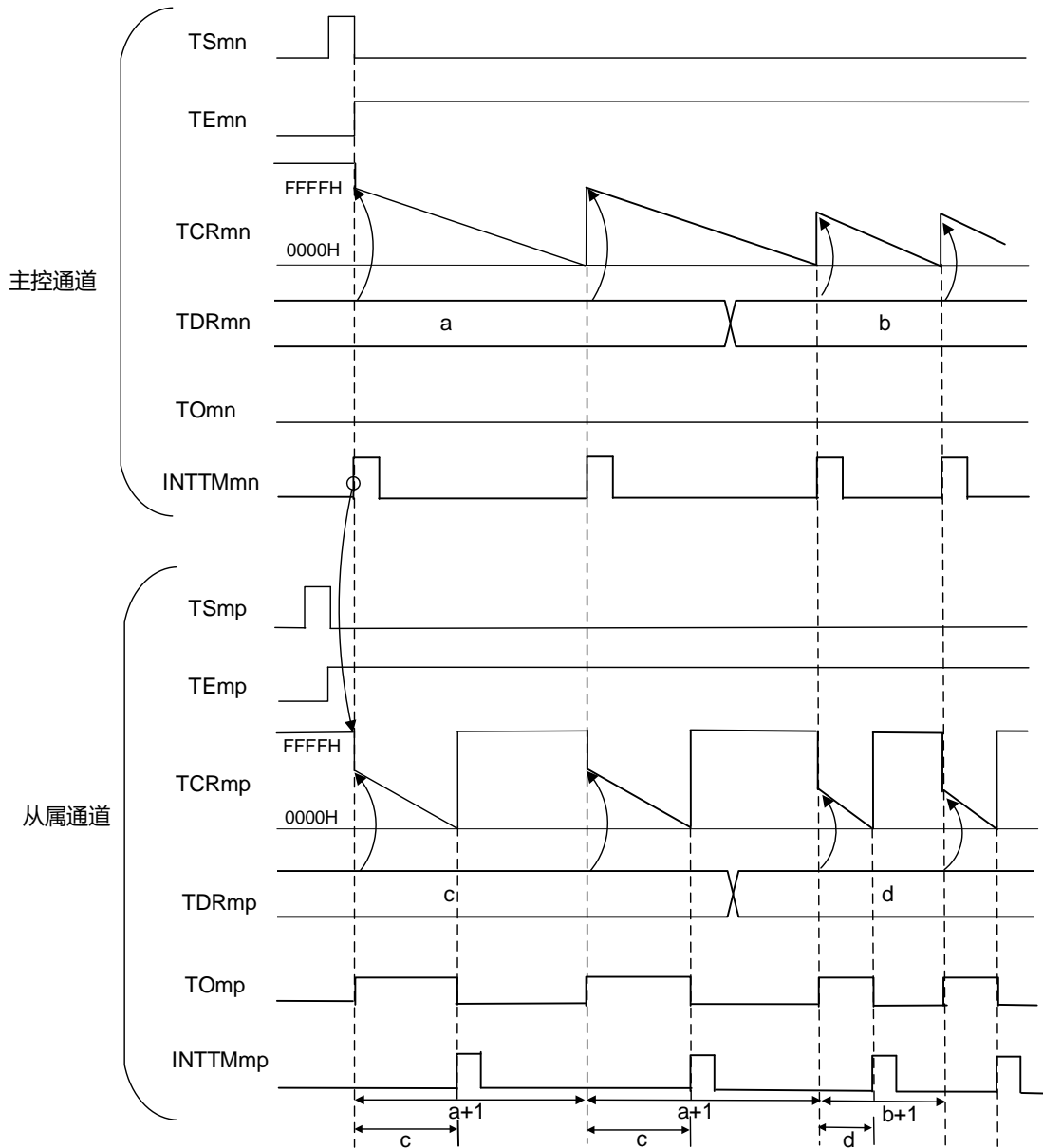
图5-35 作为PWM 功能运行的框图



注：m：单元号（m=0），n：主控通道号（n=0、2）  
 p：从属通道号（n=0：p=1、2、3，n=2：p=3）



图5-36 作为PWM功能的运行基本时序例子



注1: m: 单元号 (m=0) n: 主控通道号 (n=0, 2) p: 从属通道号 (n=0: p=1, 2, 3, n=2: p=3)

注2: TSmn、TSmp: 定时器通道开始寄存器m (TSM) 的bit n、p

TE mn、TE mp: 定时器通道允许状态寄存器m (TEM) 的bit n、p

TImn、TImp: TI mn 引脚和TI mp 引脚的输入信号

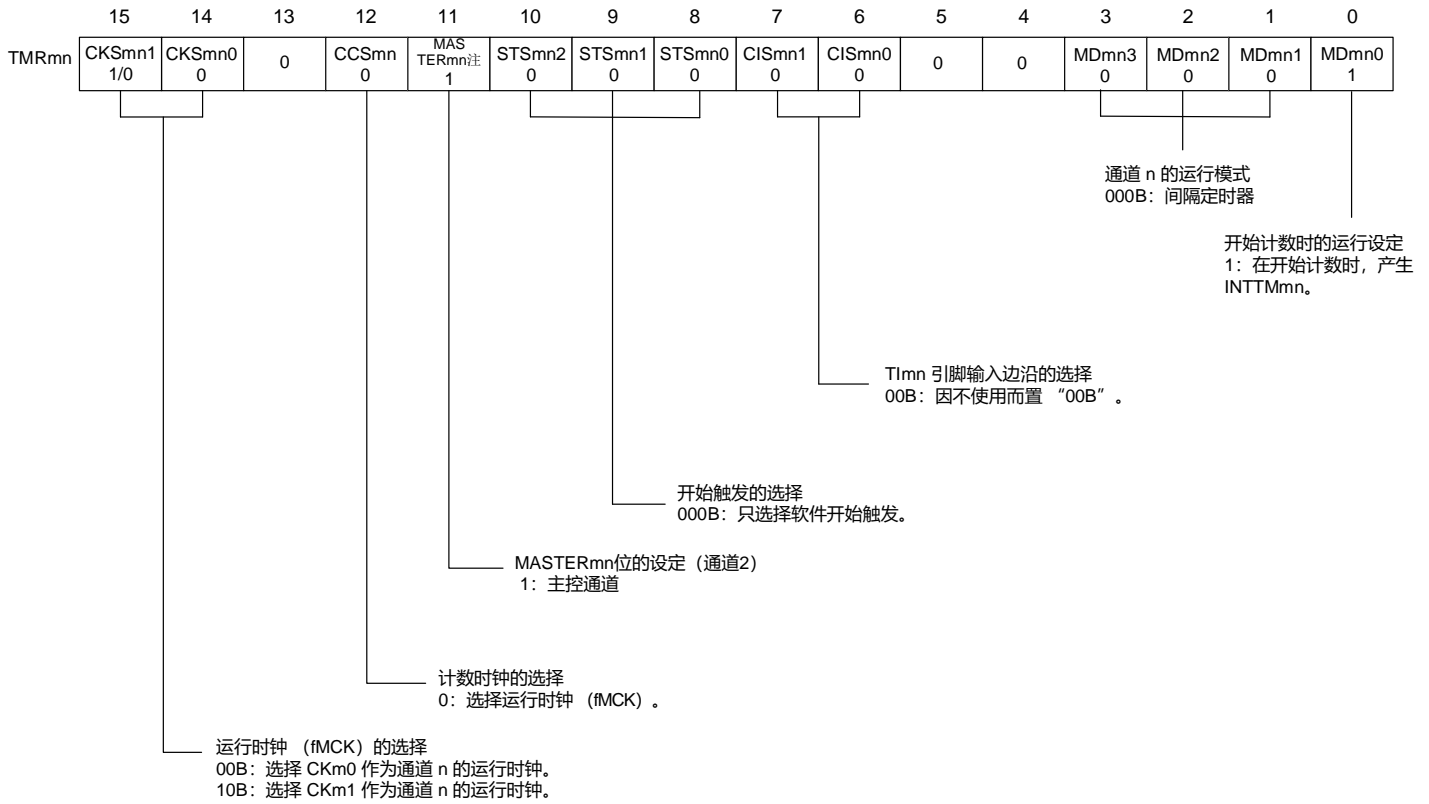
TCRmn、TCRmp : 定时器计数寄存器mn、mp (TCRmn、TCRmp)

TDRmn、TDRmp : 定时器数据寄存器mn、mp (TDRmn、TDRmp)

TOmn、TOmp: TO mn 引脚和TO mp 引脚的输出信号

图5-37 PWM 功能时（主控通道）的寄存器设定内容例子

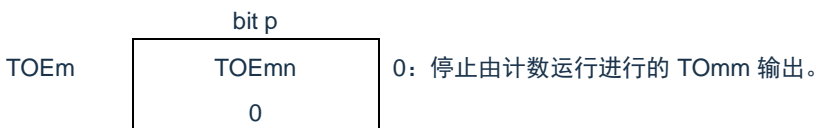
(a) 定时器模式寄存器mn（TMRmn）



(b) 定时器输出寄存器m（TOMm）



(c) 定时器输出允许寄存器m（TOEm）



(d) 定时器输出电平寄存器m（TOLm）



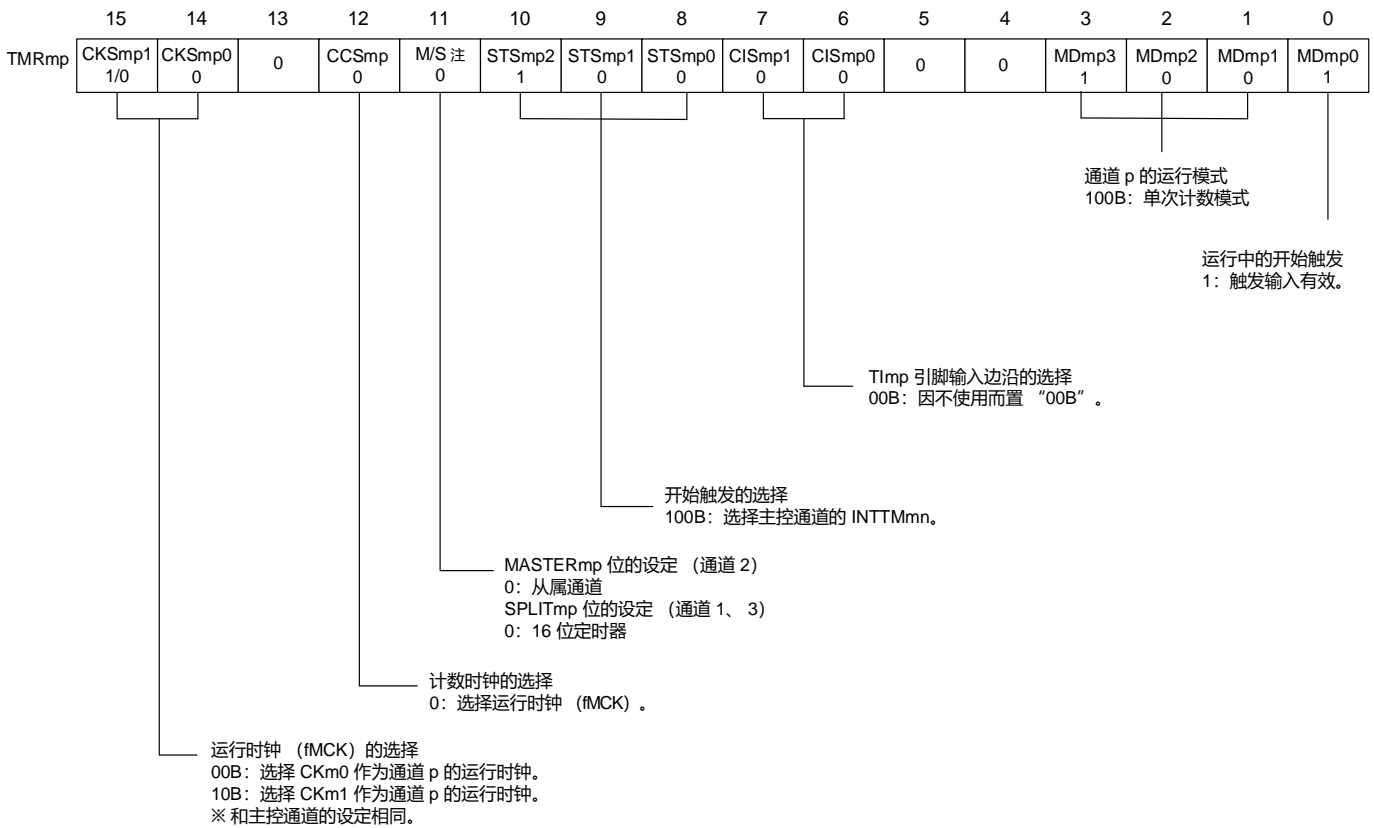
(e) 定时器输出模式寄存器m（TOMm）



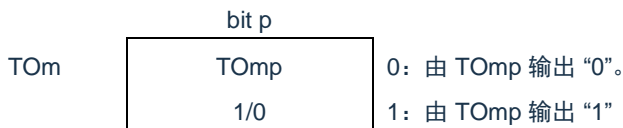
注: m: 单元号 (m=0) n: 主控通道号 (n=0、2)

图5-38 PWM 功能时（从属通道）的寄存器设定内容例子

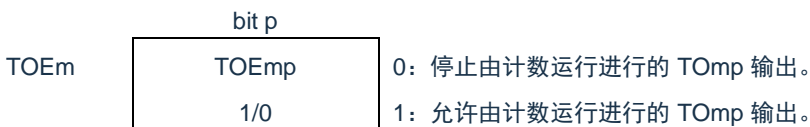
(a) 定时器模式寄存器mp（TMRmp）



(b) 定时器输出寄存器m（TOM）



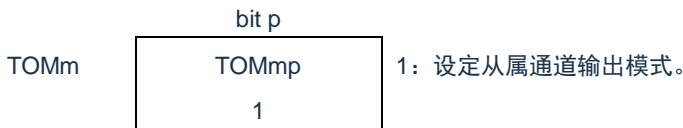
(c) 定时器输出允许寄存器m（TOEm）



(d) 定时器输出电平寄存器m（TOLm）



## (e) 定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m=0) n: 通道号 (n=0~2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

表5-34 PWM 功能时的操作步骤 (1/2)

	软件操作	硬件状态
TIMER4 初始 设定		定时器单元m的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器(PER0)的TM4mEN位置“1”。 →	定时器单元m的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0~CKm3的时钟频率。	
通道初始 设定	设定使用的2个通道的定时器模式寄存器mn, mp (TMRmn, TMRmp) (确定通道的运行模式)。 给主控通道的定时器数据寄存器mn (TDRmn) 设定间隔 (周期), 并且给从属通道的TDRmp寄存器设定占空比。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器mn (TOMmn) 的TOMmp位置 “1” (从属通道输出模式)。 设定TOLmp位。 设定TOmp位并且确定TOmp输出的初始电平。	TOmp引脚处于Hi-Z输出状态。
	将TOEmp位置“1”, 允许TOmp输出。 将端口寄模式控制寄存器和端口模式寄存器置“0”。	当端口模式寄存器为输出模式并且端口模式控制寄存器为 “0”时, 输出TOmp初始设定的电平。 因为通道处于运行停止状态, 所以TOmp不变。 TOmp引脚输出TOmp设定的电平。

表5-34 PWM 功能时的操作步骤 (2/2)

	软件操作	硬件状态
重新开始运行	开始运行 将TOEmp位 (从属)置“1”(只限于重新开始运行)。将定时器通道开始寄存器m (TSm)的TSmn(主控) 和TSmp(从属)位同时置“1”。 因为TSmn 位和TSmp位是触发位, 所以自动返回到“0”。	TTEmn位和TEmp位都变为“1”。主控通道开始计数并且产生 INTTMmn。以此为触发, 从属通道也开始计数。
	运行中 禁止更改TMRmn寄存器和TMRmp寄存器以及TOMmn位、TOMmp位、TOLmn位和TOLmp位的设定值。 能在主控通道产生 INTTMmn 后更改 TDRmn 寄存器和TDRmp寄存器的设定值。 能随时读TCRmn寄存器和TCRmp寄存器 不使用TSRmn寄存器和TSRmp寄存器。	主控通道通过检测开始触发(检测Timn引脚输入到的有效边沿或者将主控通道的TSmn位置“1”), 将TDRmn寄存器的值装入定时器计数寄存器mn(TCRmn), 并且进行递减计数。如果 TCRmn计数到“0000H”, 就产生INTTMmn, 并且在下一次Timn 引脚输入前停止计数。 从属通道以主控通道的INTTMmn为触发, 将TDRmp寄存器的值装入TCRmp寄存器并且计数器开始递减计数。在主控通道输出INTTMmn并且经过一个计数时钟后, 将TOmp的输出电平置为有效电平。然后, 如果TCRmp计数到“0000H”, 就在将TOmp的输出电平置为无效电平后停止计数。 此后, 重复此运行。
停止运行	将TTmn (主控) 和TTmp位 (从控) 同时置“1”。 因为TTmn位和TTmp位是触发位, 所以自动返回到“0”。	TEmn位和TEmp位都变为“0”并且停止计数。 TCRmn寄存器和TCRmp寄存器保持计数值而停止计数。 TOmp输出不被初始化而保持状态。
	将从属通道的TOEmp位置“0”并且给TOmp设定值。	TOmp引脚输出TOmp设定的电平。
Timer4 停止	要保持TOmp引脚输出电平的情况: 在端口寄存器设置要保持的值后将TOmp位置“0”。	通过端口功能保持TOmp引脚的输出电平。
	不需要保持TOmp引脚输出电平的情况: 不需要设定。 将PER0寄存器的TM4mEN位置“0”。	定时器单元m的输入时钟处于停止提供状态。 对全部电路和各通道的SFR进行初始化。

注1: m: 单元号 (m=0) n: 主控通道号 (n=0)

p: 从属通道号 q: 从属通道号

 $n < p < q \leq 3$  (p和q是大于n的整数)

注2: 不能将从属通道的TSmn位置1

### 5.8.3 作为多重 PWM 输出功能的运行

这是通过扩展 PWM 功能并且使用多个从属通道进行不同占空比的多个 PWM 输出的功能。例如，当将 2 个从属通道成对使用时，输出脉冲的周期和占空比能用以下计算式进行计算：

$$\begin{aligned} \text{脉冲周期} &= \{\text{TDRmn (主控) 的设定值} + 1\} \times \text{计数时钟周期} \\ \text{占空比 1}[\%] &= \{\text{TDRmp (从属 1) 的设定值}\} / \{\text{TDRmn (主控) 的设定值} + 1\} \times 100 \\ \text{占空比 2}[\%] &= \{\text{TDRmq (从属 2) 的设定值}\} / \{\text{TDRmn (主控) 的设定值} + 1\} \times 100 \end{aligned}$$

备注：当 TDRmp（从属 1）的设定值 > {TDRmn（主控）的设定值 + 1} 或者 {TDRmq（从属 2）的设定值} > {TDRmn（主控）的设定值 + 1} 时，占空比超过 100%，但是为 100% 输出。

在间隔定时器模式中，主控通道的定时器计数寄存器 mn（TCRmn）运行并且对周期进行计数。在单次计数模式中，从属通道 1 的 TCRmp 寄存器运行并且对占空比进行计数以及从 TOmp 引脚输出 PWM 波形。以主控通道的 INTTMmn 为开始触发，将定时器数据寄存器 mp（TDRmp）的值装入 TCRmp 寄存器并且进行递减计数。如果 TCRmp 变为“0000H”，就输出 INTTMmp，并且在输入下一个开始触发（主控通道的 INTTMmn）前停止计数。在从主控通道产生 INTTMmn 并且经过 1 个计数时钟后，TOmp 的输出电平变为有效电平，如果 TCRmp 变为“0000H”，就变为无效电平。

和从属通道 1 的 TCRmp 寄存器相同，在单次计数模式中，从属通道 2 的 TCRmq 寄存器运行并且对占空比进行计数以及从 TOMq 引脚输出 PWM 波形。以主控通道的 INTTMmn 为开始触发，将 TDRmq 寄存器的值装入 TCRmq 寄存器并且进行递减计数。如果 TCRmq 变为“0000H”，就输出 INTTMmq，并且在输入下一个开始触发（主控通道的 INTTMmn）前停止计数。在从主控通道产生 INTTMmn 并且经过 1 个计数时钟后，TOMq 的输出电平变为有效电平，如果 TCRmq 变为“0000H”，就变为无效电平。

当通过如此的运行将通道 0 用作主控通道时，最多能同时输出 3 种 PWM 信号。

注意：要同时改写主控通道的定时器数据寄存器 mn（TDRmn）和从属通道 1 的 TDRmp 寄存器时，至少需要 2 次写存取。因为在主控通道产生 INTTMmn 时将 TDRmn 寄存器和 TDRmp 寄存器的值装入 TCRmn 寄存器和 TCRmp 寄存器，所以如果分别在主控通道产生 INTTMmn 前和产生后进行改写，TOmp 引脚就不能输出期待的波形。因此，要同时改写主控的 TDRmn 寄存器和从属的 TDRmp 寄存器时，必须在主控通道产生 INTTMmn 后立即改写这 2 个寄存器（同样也适用于从属通道 2 的 TDRmq 寄存器）。

注：m：单元号（m=0），n：主控通道号（n=0）

p：从属通道号 q：从属通道号，n < p < q ≤ 3（p 和 q 是大于 n 的整数）

图5-39 作为多重 PWM 输出功能运行的框图（输出 2 种 PWM 的情况）

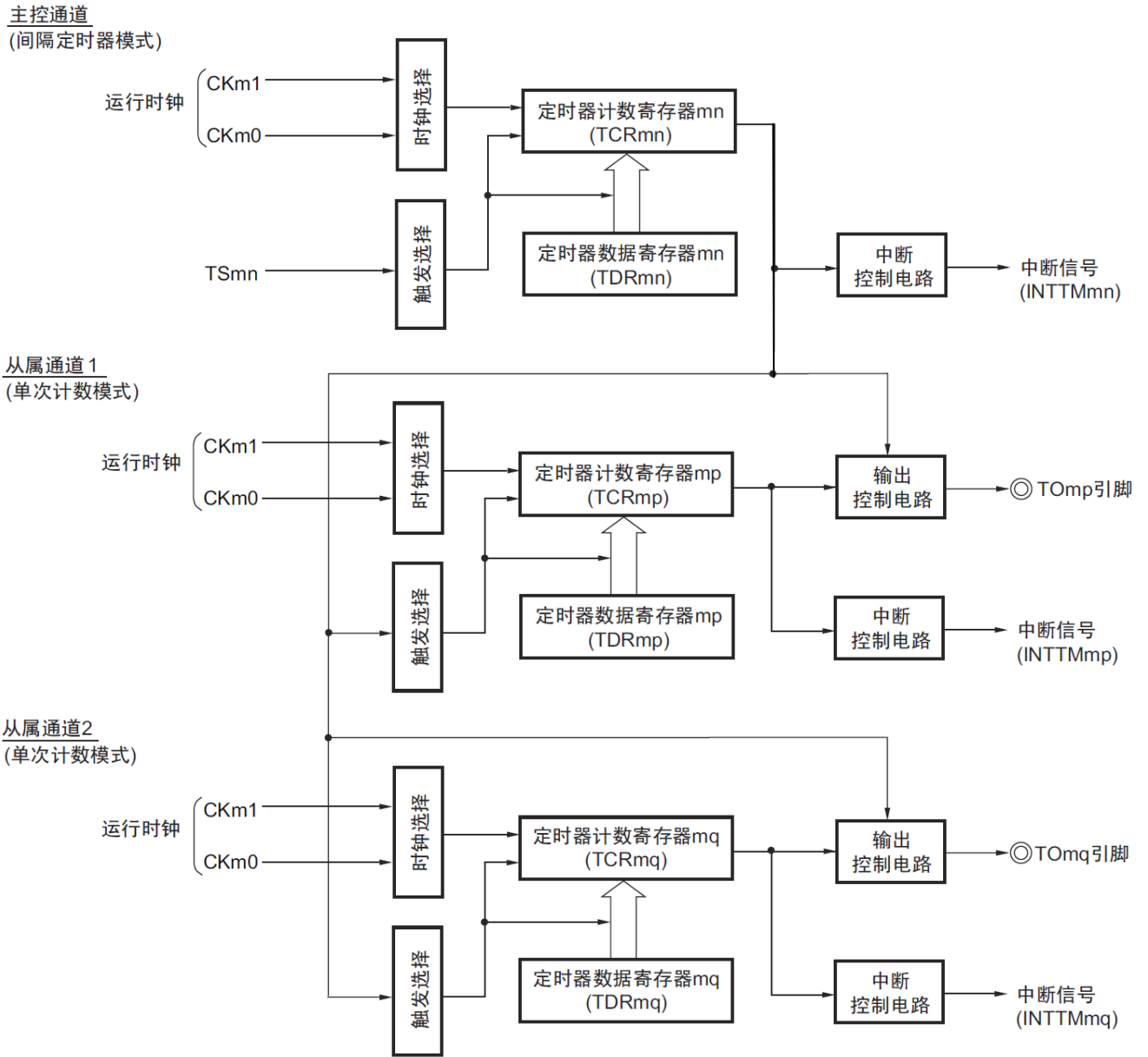
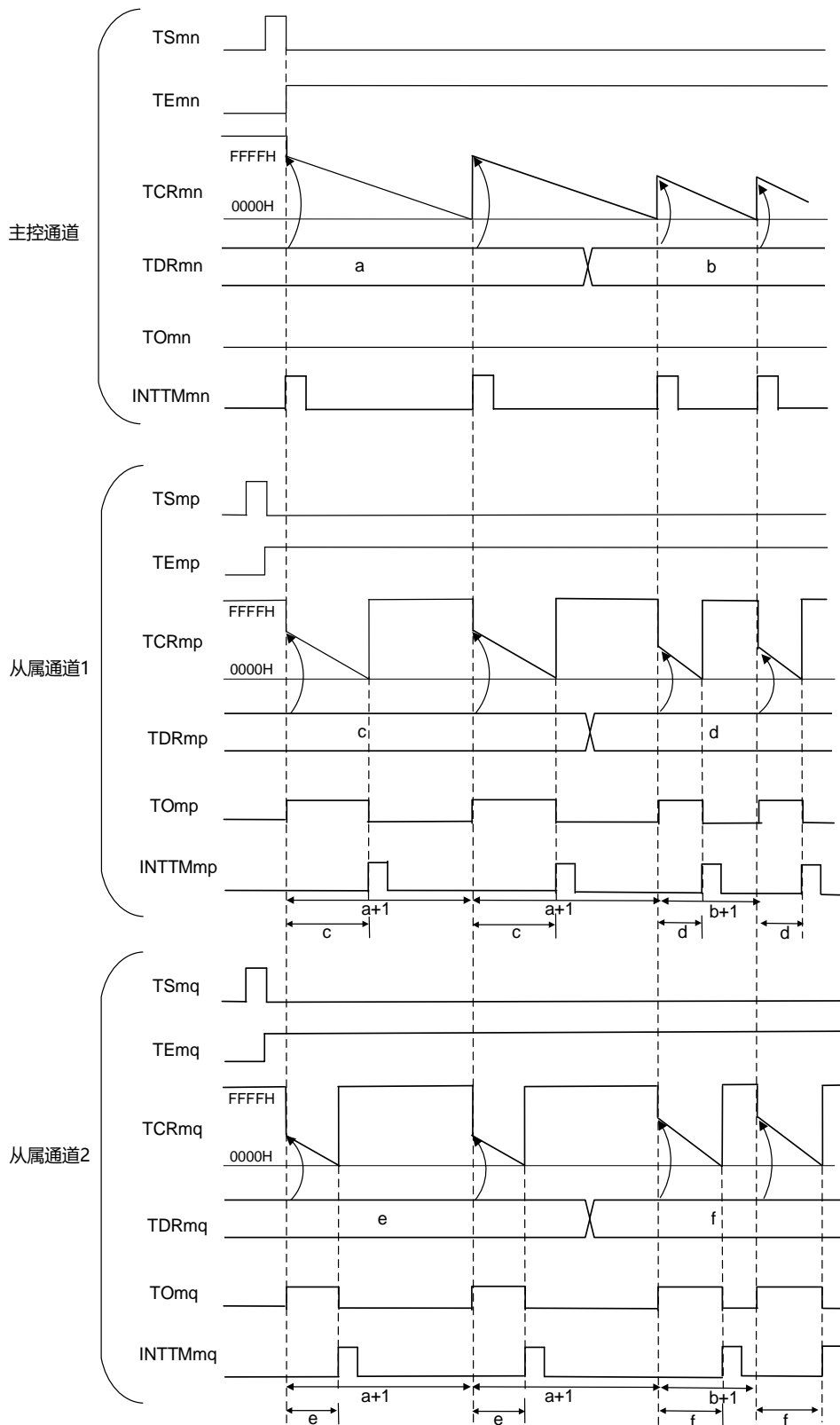


图5-40 作为多重 PWM 输出功能的运行基本时序例子（输出 2 种 PWM 的情况）



注1: m: 单元号 (m=0), n: 主控通道号 (n=0)

p: 从属通道号 q: 从属通道号,  $n < p < q \leq 3$  (p和q是大于n的整数)

注2: TSmn、TSmp、TSmq: 定时器通道开始寄存器m (TSM) 的bit n、p、q

TEmn、TEmp、TEmq: 定时器通道允许状态寄存器m (TEM) 的bit n、p、q



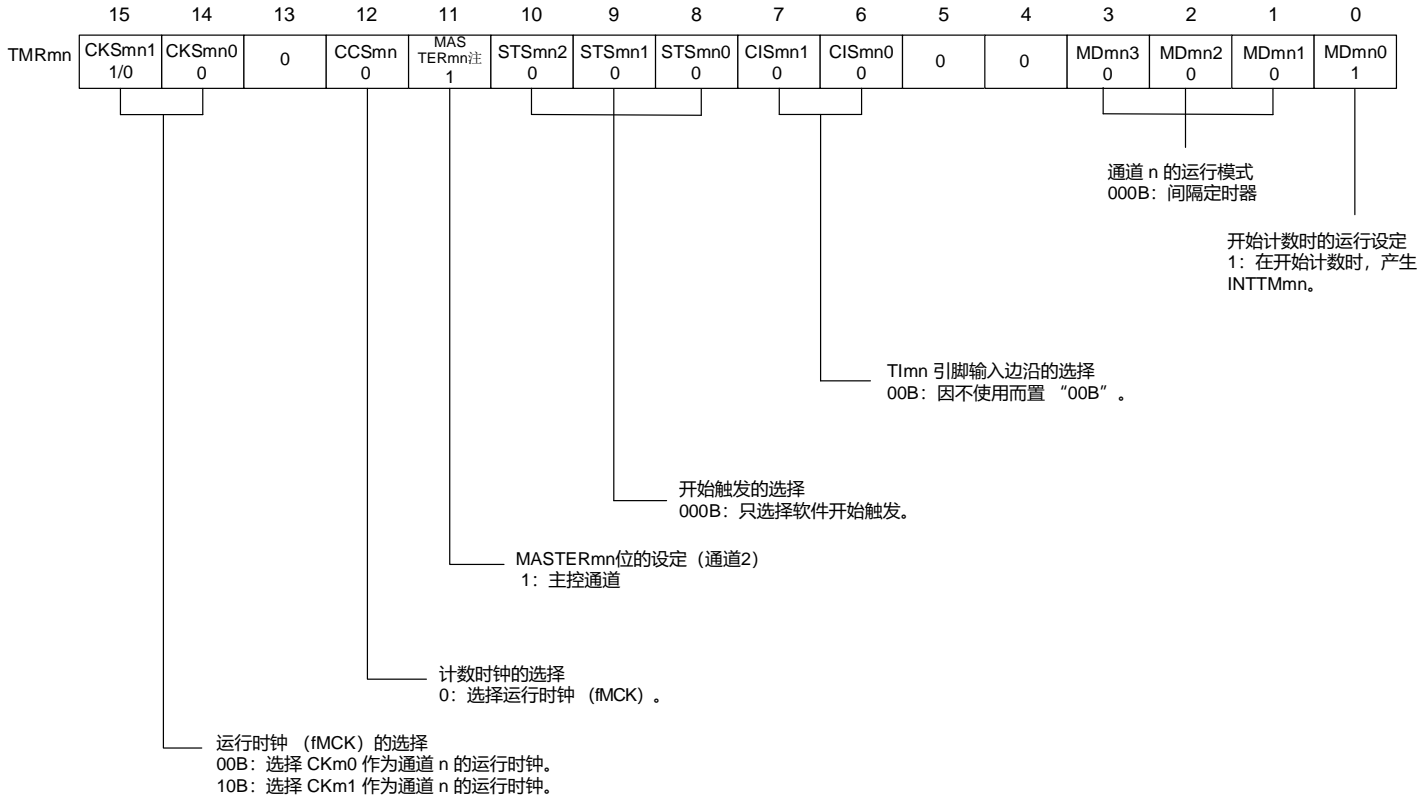
TCRmn、TCRmp、TCRmq：定时器计数寄存器mn、mp、mq（TCRmn、TCRmp、TCRmq）

TDRmn、TDRmp、TDRmq：定时器数据寄存器mn、mp、mq（TDRmn、TDRmp、TDRmq）

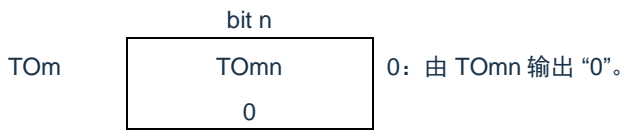
TOmn、TOmp、TOmq：TOmn、TOmp、TOmq 引脚的输出信号

图5-41 多重 PWM 输出功能时（主控通道）的寄存器设定内容例子

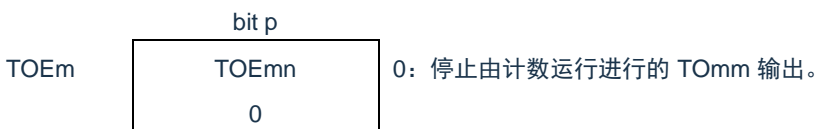
(a) 定时器模式寄存器mn（TMRmn）



(b) 定时器输出寄存器m（TOm）



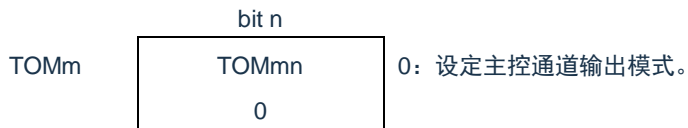
(c) 定时器输出允许寄存器m（TOEm）



(d) 定时器输出电平寄存器m（TOLm）



## (e) 定时器输出模式寄存器m (TOMm)



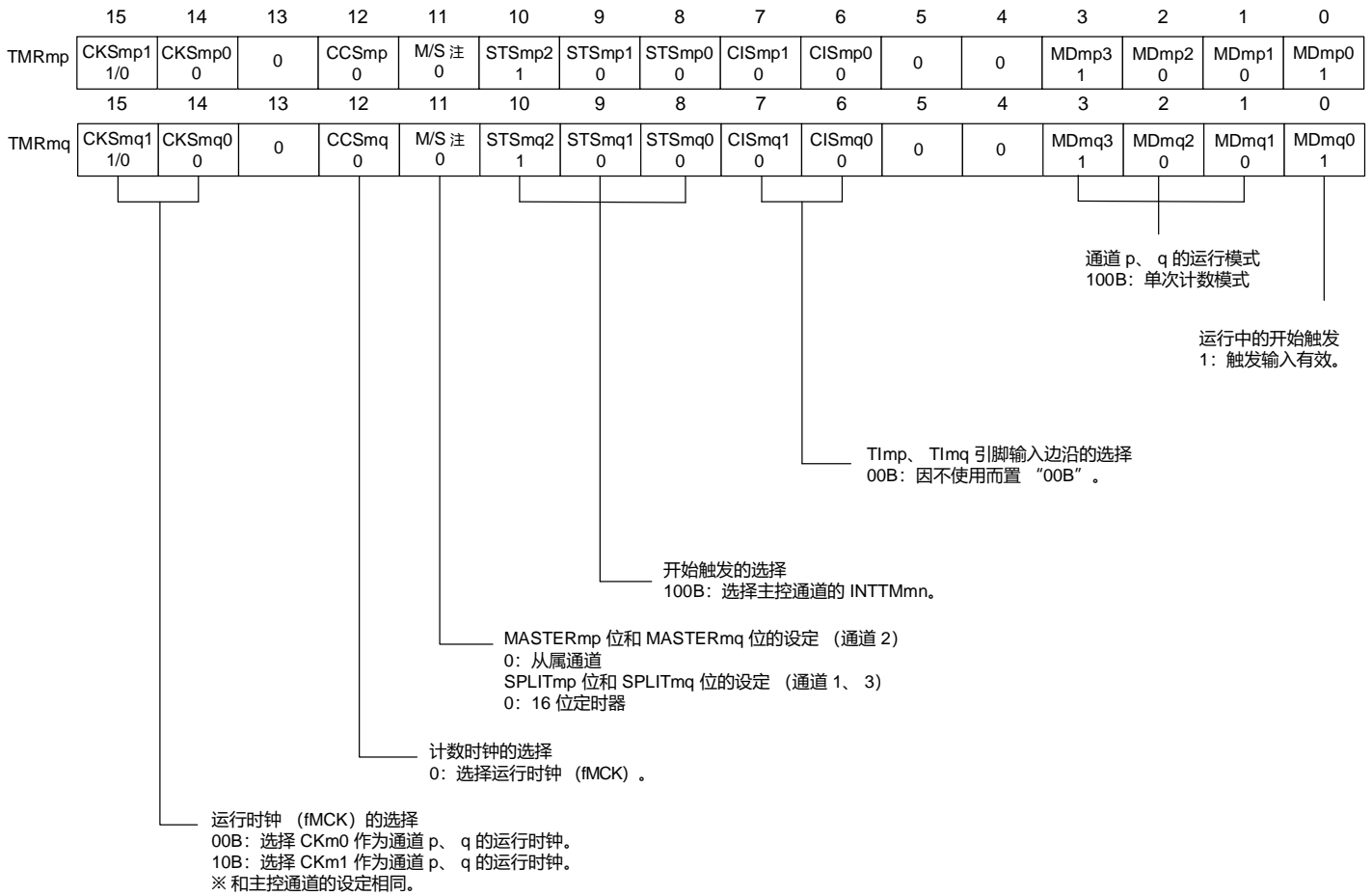
注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

注2: m: 单元号 (m= 0) n: 通道号 (n=0)

图5-42 多重 PWM 输出功能时（从属通道）的寄存器设定内容例子（输出 2 种 PWM 的情况）

(a) 定时器模式寄存器 mp、mq（TMRmp、TMRmq）



(b) 定时器输出寄存器 m（TOM）

	bit q	bit p	
TOM	TOMq	TOMp	0: 由 TOMp 和 TOMq 输出“0”。
	1/0	1/0	1: 由 TOMp 和 TOMq 输出“1”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit q	bit p	
TOEm	TOEq	TOEp	0: 停止由计数运行进行的 TOMp 和 TOMq 输出。
	1/0	1/0	1: 允许由计数运行进行的 TOMp 和 TOMq 输出。

(d) 定时器输出电平寄存器 m（TOLm）

	bit q	bit p	
TOLm	TOLmq	TOLmp	0: 正逻辑输出（高电平有效）
	1/0	1/0	1: 负逻辑输出（低电平有效）。

## (e) 定时器输出模式寄存器 m (TOMm)

	bit q	bit p	
TOMm	TOMmq 1	TOMmp 1	1: 设定从属通道输出模式。

注: m: 单元号 (m=0), n: 主控通道号 (n=0)

 p: 从属通道号 q: 从属通道号,  $n < p < q \leq 3$  (p和q是大于n的整数)

表5-35 多重 PWM 输出功能时的操作步骤 (输出 2 种 PWM 的情况) (1/2)

	软件操作	硬件状态
TIMER4 初始 设定		定时器单元m的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器(PERO)的TM4mEN位置“1”。	定时器单元m的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
通道初始 设定	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0~CKm3的时钟频率。	
	设定使用的2个通道的定时器模式寄存器mn, mp (TMRmn, TMRmp) (确定通道的运行模式)。 给主控通道的定时器数据寄存器mn (TDRmn) 设定间隔 (周期), 并且给从属通道的TDRmp寄存器设定占空比。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器mn (TOMmn) 的TOMmp位和 TOMmq位置“1” (从属通道输出模式)。 将TOLmp位和TOLmq位置0。 设定TOmp位和TOmq位, 并且确定TOmp和TOmq输出的 初始电平。	TOmp引脚处于Hi-Z输出状态。  当端口模式寄存器为输出模式并且端口模式控制寄存器为 “0”时, 输出TOmp和TOmq初始设定的电平。
	将TOEmp位和TOEmq位置“1”, 允许TOmp和TOmq输出。 将端口寄模式控制寄存器和端口模式寄存器置“0”。	因为通道处于运行停止状态, 所以TOmp和TOmq不变。  TOmp引脚和TOmq引脚输出TOmp和TOmq设定的电平。

表5-35 多重 PWM 输出功能时的操作步骤（输出 2 种 PWM 的情况）(2/2)

	软件操作	硬件状态
重新 开始 运行	开始运行 （只在重新开始运行时将TOEmp位和TOEmq位（从属）置“1”）。 将定时器通道开始寄存器m（TSm）的TSmn位（主控）、TSmp位和TSmq位（从属）同时置“1”。 因为TSmn位、TSmp位和TSmq位是触发位，所以自动返回到“0”。	TEmn位、TEmp位和TEmq位都变为“1”，主控通道开始计数并且产生INTTMmn以此为触发，从属通道也开始计数。
	运行中 禁止更改 TMRmn寄存器、TMRmp寄存器和TMRmq寄存器以及TOMmn位、TOMmp位、TOMmq位、TOLmn位、TOLmp位和TOLmq位的设定值。 能在主控通道产生 INTTMmn 后更改TDRmn、TDRmp和TDRmq寄存器的设定值。 能随时读TCRmn、TCRmp、TCRmq寄存器 不使用TSRmn、TSRmp、TSRmq寄存器。	主控通道将TDRmn寄存器的值装入定时器计数寄存器mn(TCRmn)，并且进行递减计数。如果 TCRmn计数到“0000H”，就产生INTTMmn，同时，将TDRmn寄存器的值装入TCRmn寄存器，并且重新开始递减计数。 从属通道1以主控通道的INTTMmn为触发，将TDRmp寄存器的值装入TCRmp寄存器并且计数器开始递减计数。在主控通道输出INTTMmn并且经过一个计数时钟后，将TOmp的输出电平置为有效电平。然后，如果TCRmp计数到“0000H”，就在将TOmp的输出电平置为无效电平后停止计数。 从属通道2以主控通道的INTTMmn为触发，将TDRmq寄存器的值装入TCRmq寄存器并且计数器开始递减计数。在主控通道输出INTTMmn并且经过一个计数时钟后，将TOMq的输出电平置为有效电平。然后，如果TCRmq计数到“0000H”，就在将TOMq的输出电平置为无效电平后停止计数。 此后，重复此运行。
	停止运行 将TTmn（主控）、TTmp位和TTmq位（从属）同时置“1”。 因为TTmn位、TTmp位和TTmq位是触发位，所以自动返回到“0”。 将从属通道的TOEmp位和TOEmq位置“0”并且给TOmp和TOMq设定值。	TEmn位、TEmp位和TEmq位都变为“0”并且停止计数。 TCRmn、TCRmp、TCRmq寄存器保持计数值而停止计数。 TOmp和TOMq输出不被初始化而保持状态。
	Timer4 停止 要保持TOmp引脚和TOMq引脚输出电平的情况： 在端口寄存器设置要保持的值后将TOmp位和TOMq位置“0”。 不需要保持TOmp引脚和TOMq引脚输出电平的情况： 不需要设定。	通过端口功能保持TOmp引脚和TOMq引脚的输出电平。
	将PER0寄存器的TM4mEN位置“0”。	定时器单元m的输入时钟处于停止提供状态。 对全部电路和各通道的SFR进行初始化。 （TOmp位和TOMq位变为“0”并且TOmp引脚和TOMq引脚变为端口功能）

注：m：单元号（m=0），n：主控通道号（n=0）

p：从属通道号 q：从属通道号，n<p<q≤3（p和q是大于n的整数）

## 第6章 LSITIMER 12位间隔定时器

### 6.1 12位间隔定时器的功能

以事先设定的任意时间间隔产生中断（INTIT），能用于从睡眠模式、深度睡眠模式和部分掉电模式的唤醒。

### 6.2 12位间隔定时器的结构

12 位间隔定时器由以下硬件构成。

表6-1 12位间隔定时器的结构

项目	结构
计数器	12位计数器
控制寄存器	12位间隔定时器的控制寄存器（CON0）

### 6.3 寄存器映射

(CON0基地址 = 0x4004\_4B50) RO: 只读, WO: 只写, RW: 读写

寄存器	偏移量	读/写	描述	复位值
CON0	0x000	R/W	12位间隔定时器的控制寄存器	0xFFFF

### 6.4 12位间隔定时器的控制寄存器（CON0）

这是设定 12 位间隔定时器的运行开始和停止以及比较值的寄存器。

通过 12 位存储器操作指令设定 CON0 寄存器。

在产生复位信号后，此寄存器的值变为“FFFH”。

位	符号	描述	复位值
15	RINTE	12位间隔定时器的运行控制 0: 停止计数器的运行（清除计数） 1: 开始计数器的运行	0
14:12	--	保留	--
11:0	ITCMP	12位间隔定时器比较值的设定	0xFFFF

有关ITCMP比较值的设定以及相关中断周期计算如下表：（仅供参考）

ITCMP[11]~ITCMP[0]	12位间隔定时器比较值的设定
001H	这些位产生周期为ITCMP设定值+1的固定周期中断。
•	
•	
FFFH	
000H	禁止设定。
ITCMP[11]~ITCMP[0]为“001H”或者“FFFH”时的中断周期例子	
ITCMP[11]~ITCMP[0]=001H, 计数时钟: $f_{clk}=15\text{kHz}$ $1/15[\text{kHz}]\cdot(1+1)=0.13333$ [ms]	
ITCMP[11]~ITCMP[0]=FFFH, 计数时钟: $f_{clk}=15\text{kHz}$ $1/15[\text{kHz}]\cdot(4095+1)=273.06667$ [ms]	

注1: 要将RINTE位从“1”改为“0”时, 必须在通过中断屏蔽标志寄存器将INTIT设定为禁止中断处理后进行改写。要重新开始运行（从“0”改为“1”）时, 必须在清除ITIF标志后设定为允许中断处理。

注2: RINTE位的读取值在设定RINTE位后的1个计数时钟之后被反映。

注3: 在从睡眠模式转移到通常运行模式后, 如果要设定CON0寄存器并且再次转移到睡眠模式, 就必须在确认CON0寄存器的写入值被反映后或者在设定CON0寄存器后至少经过1个计数时钟之后再转移到睡眠模式。

注4: 要更改ITCMP11~ITCMP0位的设定时, 必须在RINTE位为“0”的状态下进行。

注5: 允许在将RINTE位从“0”改为“1”或者从“1”改为“0”的同时更改ITCMP11~ITCMP0位的设定。

## 6.5 12位间隔定时器的运行

### 6.5.1 12位间隔定时器的运行时序

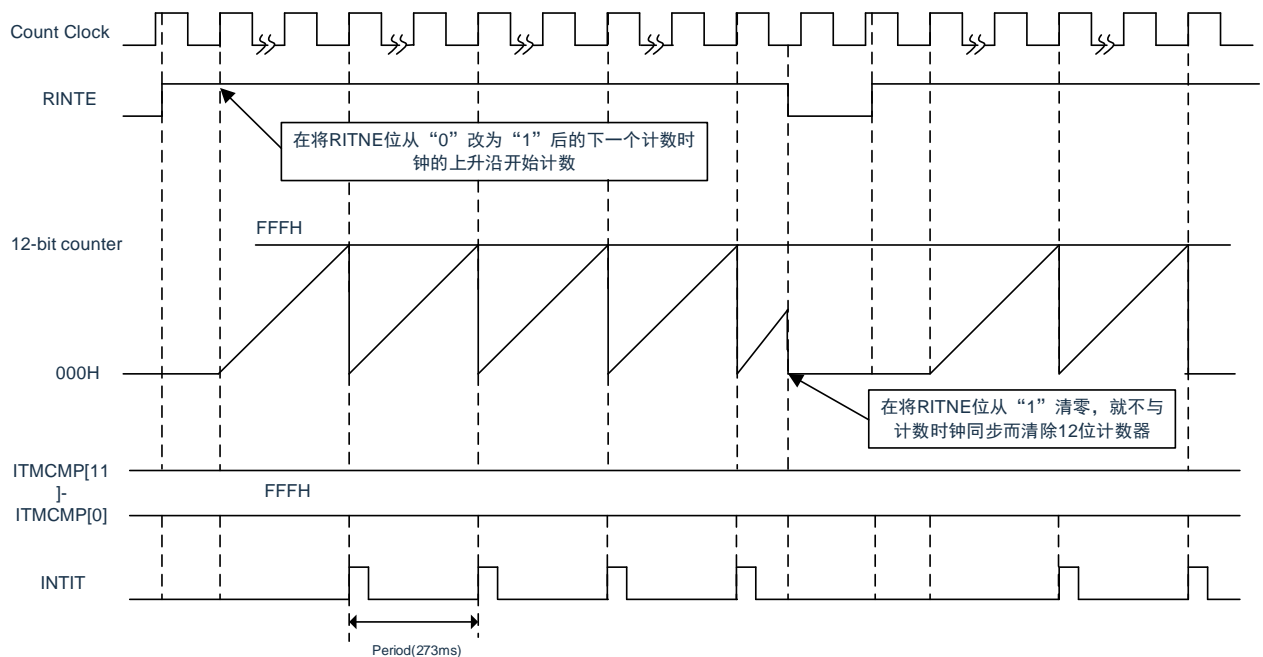
以ITCMP11~ITCMP0位设定的计数值为间隔，作为重复产生中断请求（INTIT）的12位间隔定时器运行。如果将RINTE位置“1”，12位计数器就开始计数。

当12位计数值和ITCMP11~ITCMP0位的设定值相同时，将12位计数值清“0”并且继续计数，同时产生中断请求信号（INTIT）。

12位间隔定时器的基本运行如图6-1所示。

图6-1 12位间隔定时器的运行时序

(ITCMP[11]~ITCMP[0]=FFFH, 计数时钟:  $f_{clk}=15kHz$ )

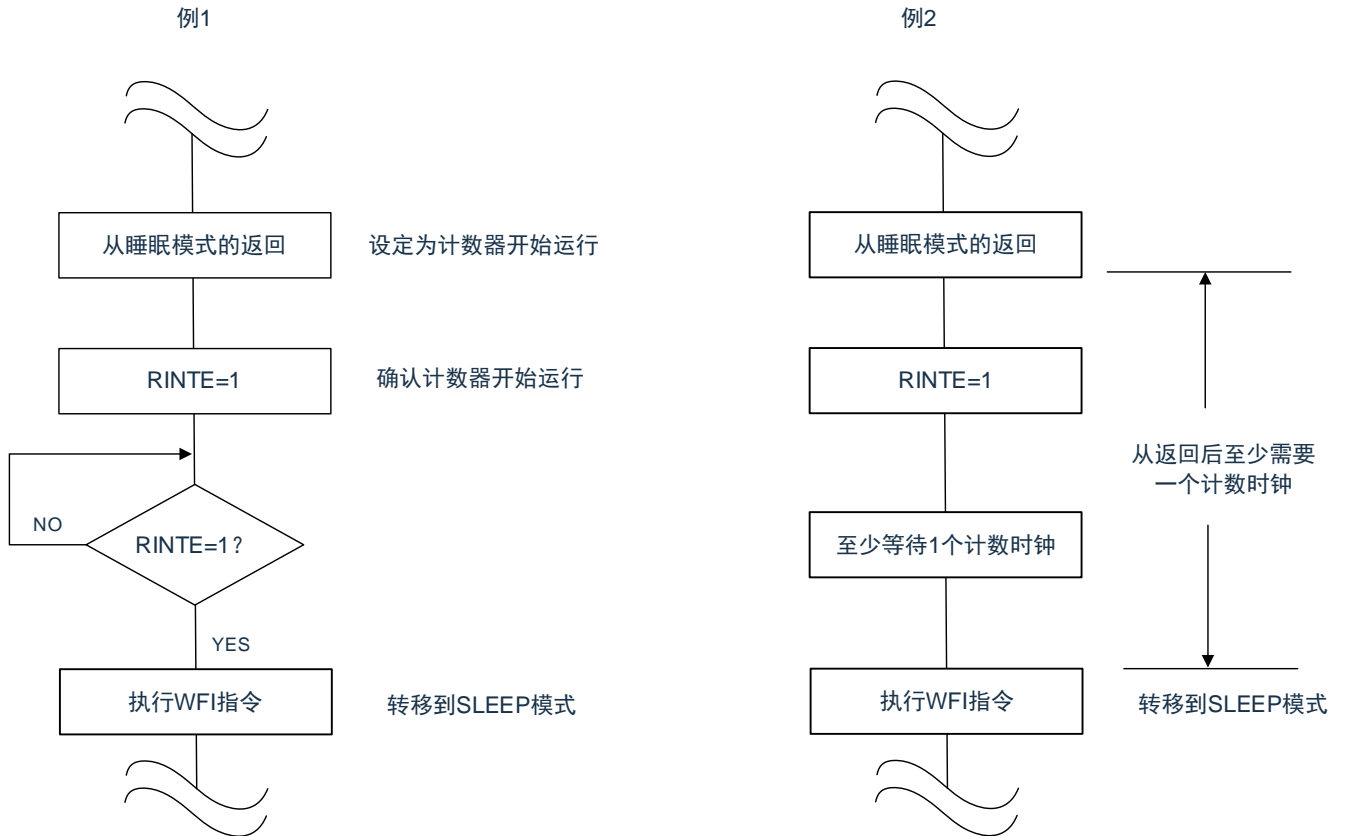




### 6.5.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移

在从睡眠模式返回后，如果要将 RINTE 位置“1”并且再次转移到睡眠模式，就必须在将 RINTE 位置“1”后确认 RINTE 位的写入值被反映，或者在返回后至少经过 1 个计数时钟的时间之后再转移到睡眠模式。

- 在将 RINTE 位置“1”后，通过轮询确认 RINTE 位变为“1”，然后转移到睡眠模式（参照下图的例 1）。
- 在将 RINTE 位置“1”后至少经过 1 个计数时钟的时间之后转移到睡眠模式（参照下图的例 2）。



# 第7章 时钟输出控制电路

## 7.1 时钟输控制电路的功能

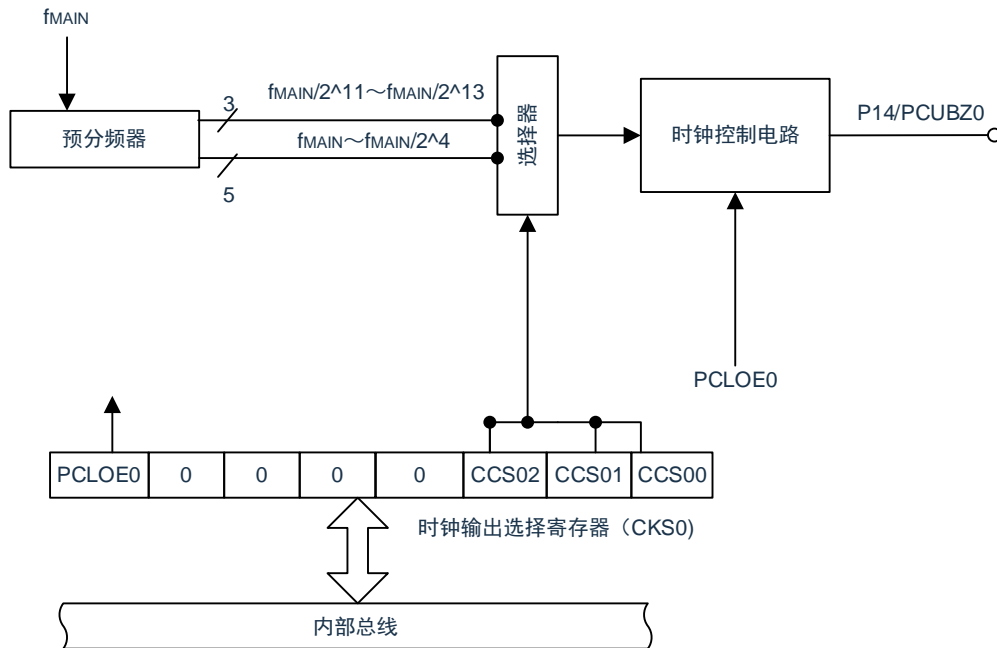
时钟输出是输出提供给外围 IC 时钟的功能。

本产品有一个时钟输出输出引脚 PCUBZ0，相对应的端口在复用后可做时钟输出。

PCUBZ0 引脚输出由时钟输出选择寄存器（CKS0）选择的时钟。

时钟输出控制电路的框图如图 7-1 所示。

图7-1 时钟输出控制电路的框图



注1：有关能从PCUBZ0引脚输出的频率，请参照“数据手册的AC特性”。

## 7.2 时钟输出控制电路的结构

时钟输出控制电路由以下硬件构成。

表7-1 时钟输出控制电路的结构

项目	结构
控制寄存器	时钟输出选择寄存器 (CKS0) 端口模式控制寄存器 (PMCxx)、端口模式寄存器 (PMxx)、端口复用控制寄存器 (PxxCFG)

## 7.3 寄存器映射

(CKS0 基地址 = 0x4004\_0FA5)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CKS0	0x000	R/W	时钟输出选择寄存器	0x0

### 7.3.1 时钟输出选择寄存器（CKS0）

这是允许或者禁止时钟输出引脚（PCUBZ0）的输出以及设定输出时钟的寄存器。通过 CKS0 寄存器选择 PCUBZ0 引脚输出的时钟。通过 8 位存储器操作指令设定 CKS0 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	PCLOE0	PCUBZ0引脚输出允许/禁止的指定 0: 禁止输出（默认值） 1: 允许输出	0x0
6:3	--	保留	--
2:0	CCS0	PCUBZ0引脚输出时钟的选择	0x0

具体PCUBZ0引脚输出时钟的选择如下表：

CCS0[2]	CCS0[1]	CCS0[0]	PCUBZ0引脚输出时钟的选择
0	0	0	$f_{MAIN}$
0	0	1	$f_{MAIN}/2$
0	1	0	$f_{MAIN}/2^2$
0	1	1	$f_{MAIN}/2^3$
1	0	0	$f_{MAIN}/2^4$
1	0	1	$f_{MAIN}/2^{11}$
1	1	0	$f_{MAIN}/2^{12}$
1	1	1	$f_{MAIN}/2^{13}$

注1：必须在16MHz以内的范围内使用输出时钟。详细内容请参照“数据手册的AC特性”。

注2：输出时钟的切换必须在设定为禁止输出（PCLOE0=0）后进行。

注3：如果要转移到深度睡眠模式，就必须在执行WFI指令前将PCLOE0置“0”。

注4： $f_{MAIN}$ ：主系统时钟频率

## 7.4 配置时钟输出端口功能的寄存器

本产品有一个时钟输出引脚 PCUBZ0，相对应的端口在复用后可做时钟输出。

使用时钟输出功能时，必须设定端口复用功能配置寄存器（PxxCFG），端口寄存器（Pxx），端口模式寄存器（PMxx）和端口模式控制寄存器（PMCxx）。详细内容请参照“第2章 引脚功能”。

被配置为时钟输出引脚的复用端口，其对应的端口寄存器（Pxx），端口模式寄存器（PMxx）的位和端口模式控制寄存器（PMCxx）的位必须置“0”。

（例）将 P14 用作时钟输出（PCUBZ0）的情况：

将端口寄存器 1 的 P14 位置 “0”。

将端口模式寄存器 1 的 PM14 位置 “0”。

将端口模式控制寄存器 1 的 PMC14 位置 “0”。

将端口复用功能配置寄存器 P14CFG 置 “0x02”。

## 7.5 时钟输出控制电路的运行

相对应的端口在复用后可做时钟输出。

PCUBZ0 引脚输出由时钟输出选择寄存器 (CKS0) 选择的时钟。

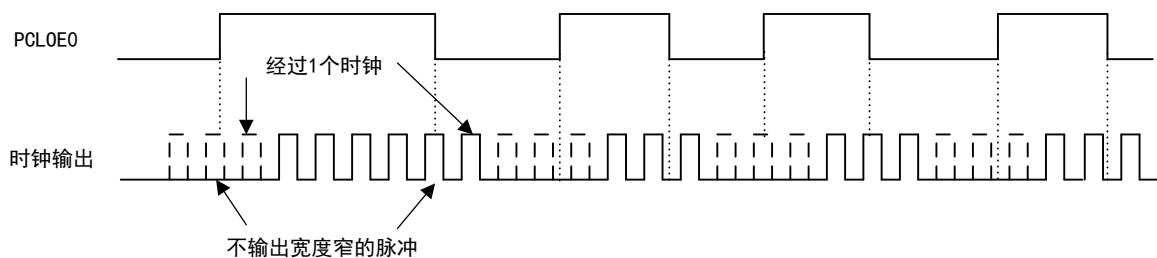
### 7.5.1 输出引脚的运行

相对应的端口在复用后可用作时钟输出。

- 1) 设定端口复用功能配置寄存器 (PmnCFG), 将用作 PCUBZ0 引脚的端口对应的端口寄存器 (Pxx), 端口模式寄存器 (PMxx) 和端口模式控制寄存器 (PMCxx) 的位置“0”。
- 2) 通过 PCUBZ0 引脚的时钟输出选择寄存器 (CKS0) 的 bit0~2 (CCS0~CCS2) 选择输出频率 (输出为禁止状态)。
- 3) 将 CKS0 寄存器的 bit7 (PCLOE0) 置“1”, 允许时钟输出。

注1: 用作时钟输出时的控制电路在允许或者禁止时钟输出 (PCLOE0位) 后的1个时钟之后, 开始或者停止时钟输出。此时不输出宽度窄的脉冲。通过PCLOE0位允许或者停止输出以及时钟输出的时序如图7-2所示。

图7-2 PCUBZ0引脚的时钟输出时序



## 7.6 时钟输出控制电路的注意事项

当选择主系统时钟作为 PCUBZ0 输出时, 如果在设定停止输出 (PCLOE0=0) 后的 1.5 个 PCUBZ0 引脚的输出时钟内转移到深度睡眠模式, PCUBZ0 的输出宽度就变窄。

## 第8章 看门狗定时器

### 8.1 看门狗定时器的功能

看门狗定时器通过选项字节（000C0H）设定计数运行。看门狗定时器以低速内部振荡器时钟（ $f_{LL}$ ）运行。看门狗定时器用于检测程序失控。在检测到程序失控时，产生内部复位信号。

下述情况判断为程序失控：

- 当看门狗定时器的计数器发生上溢时
- 当给 WDTE 寄存器写“ACH”以外的数据时
- 在窗口关闭期间给 WDTE 寄存器写数据时

当因看门狗定时器而发生复位时，将复位控制标志寄存器（RESF）的 bit4（WDTRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 26 章 复位功能”。当达到上溢时间的  $75\%+1/2f_{LL}$  时，能产生间隔中断。

### 8.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表8-1 看门狗定时器的结构

项目	结构
计数器	内部计数器（17 位）
控制寄存器	看门狗定时器的允许寄存器（WDTE）

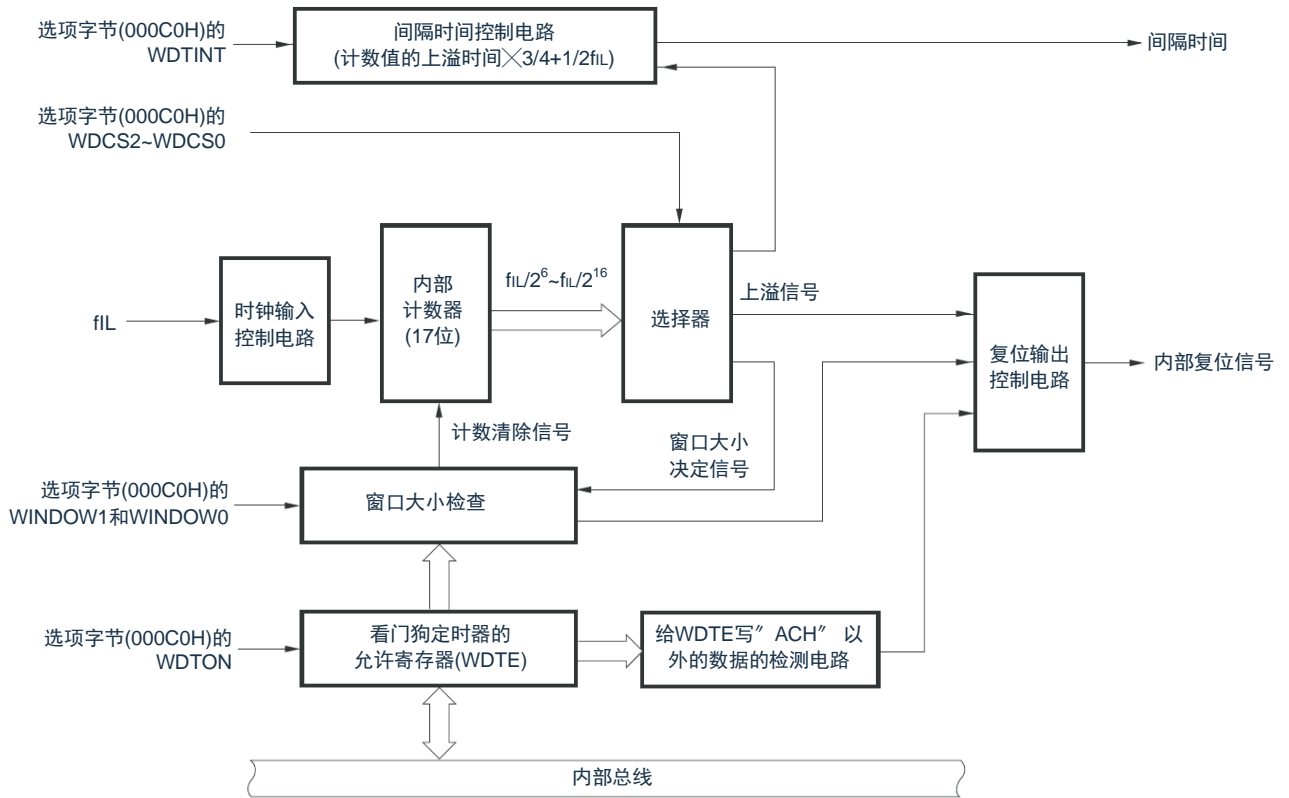
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表8-2 选项字节和看门狗定时器的设定内容

看门狗定时器的设定内容	选项字节（000C0H）
看门狗定时器的间隔中断的设定	bit7（WDTINT）
窗口打开期间的设定	bit6和bit5（WINDOW1、WINDOW0）
看门狗定时器的计数器运行控制	bit4（WDTON）
看门狗定时器的上溢时间的设定	bit3~1（WDCS2~WDCS0）
看门狗定时器的计数器运行控制（睡眠时）	bit0（WDSTBYON）

注：有关选项字节，请参照“第31章 选项字节”。

图8-1 看门狗定时器的框图



注:  $f_{IL}$ : 低速内部振荡器的时钟频率



## 8.3 寄存器映射

(WDTE 基地址 = 0x4002\_1001) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
WDTE	0x000	R/W	看门狗定时器的允许寄存器	0x1A/0x9A

(LOCKCTL基地址 = 0x4002\_0405) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LOCKCTL	0x000	R/W	控制寄存器	0x1

(PRCR基地址 = 0x4002\_0406) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PRCR	0x000	R/W	保护寄存器	0x0

### 8.3.1 看门狗定时器的允许寄存器 (WDTE)

通过给 WDTE 寄存器写“ACH”，清除看门狗定时器的计数器并且重新开始计数。通过 8 位存储器操作指令设定 WDTE 寄存器。在产生复位信号后，此寄存器的值变为“9AH”或者“1AH”<sup>注</sup>。

位	符号	描述	复位值
7:0	WDTE	写0xAC清除看门狗定时器的计数器并且重新开始计数	0x1A/0x9A

注：WDTE寄存器的复位值因选项字节（000C0H）的WDTON位的设定值而不同。要使看门狗定时器运行时，必须将WDTON位置“1”。

WDTON位的设定值	WDTE寄存器的复位值
0（禁止看门狗定时器的计数运行）	1AH
1（允许看门狗定时器的计数运行）	9AH

注1：当给WDTE寄存器写“ACH”以外的值时，产生内部复位信号。

注2：WDTE寄存器的读取值为“9AH/1AH”（和写入值（“ACH”）不同）。

### 8.3.2 LOCKUP控制寄存器（LOCKCTL）

LOCKCTL 寄存器是 Cortex-M0+ LockUp 功能是否引起看门狗定时器运行的配置寄存器，PRCR 是其写保护寄存器。

通过 8 位存储器操作指令设置 LOCKCTL 寄存器。

在产生复位信号后，LOCKCTL 寄存器的值变为“01H”。

位	符号	描述	复位值
7:1	-	保留	-
0	lockup_rst	LOCKUP功能的配置 0: LOCKUP不导致WDT复位 1: LOCKUP导致WDT复位	1

### 8.3.3 保护寄存器（PRCR）

LOCKCTL 寄存器是 Cortex-M0+ LockUp 功能是否引起看门狗定时器运行的配置寄存器，PRCR 是其写保护寄存器。

通过 8 位存储器操作指令设置 PRCR 寄存器。

在产生复位信号后 PRCR 寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	PRTKEY	PRCR的写保护 78H: PRCR可写 其他: PRCR不可写	0x0
0	PRCR	LOCKUP控制寄存器写保护 0: LOCKCTL寄存器不可写 1: LOCKCTL寄存器可写	0

### 8.3.4 看门狗配置寄存器WDTCFGx(x=0~3)

WDTCFGx 配置寄存器是是否强制看门狗定时器运行的寄存器。

通过 8 位寄存器操作指令设置 WDTCFG x 寄存器。

在产生复位信号后，WDTCFGx 寄存器的值变为“00H”

WDTCFGx 配置寄存器

位	符号	描述	复位值
7:0	WDTCFGx	看门狗配置寄存器设定为特定值时，可以强制看门狗定时器的运行，详见下表	0x0

WDTCFG0	WDTCFG1	WDTCFG2	WDTCFG3	看门狗定时器功能的配置
0x1A	0x2B	0x3C	0x4D	复位后看门狗定时器的运行由选项字节决定 <sup>注1</sup>
其他				复位后强制运行看门狗定时器

注1：详细配置参考31.4用户选项字节章节

## 8.4 看门狗定时器的运行

### 8.4.1 看门狗定时器的运行控制

1. 当使用看门狗定时器时，通过选项字节（000C0H）设定以下内容：

- 必须将选项字节（000C0H）的 bit4（WDTON）置“1”，允许看门狗定时器的计数运行（在解除复位后，计数器开始运行）（详细内容请参照第 31 章 选项字节）。

WDTON	看门狗定时器的计数器
0	禁止计数运行（解除复位后停止计数）。
1	允许计数运行（解除复位后开始计数）。

• 必须通过选项字节（000C0H）的 bit3~1（WDCS2~WDCS0）设定上溢时间（详细内容请参照 8.4.2 和第 31 章）。

• 必须通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照 8.4.3 和第 31 章）。

2. 在解除复位后，看门狗定时器开始计数。

3. 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。

4. 此后，解除复位后第 2 次以后的 WDTE 寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写 WDTE 寄存器，就产生内部复位信号。

5. 如果不给 WDTE 寄存器写“ACH”而超过上溢时间，就产生内部复位信号。以下情况会产生内部复位信号：

- 当给 WDTE 寄存器写“ACH”以外的数据时

注 1：只在解除复位后第 1 次写看门狗定时器的允许寄存器（WDTE）时，与窗口打开期间无关，只要在上溢时间前的任意时候写 WDTE，就清除看门狗定时器并且重新开始计数。

注 2：从给 WDTE 寄存器写“ACH”到清除看门狗定时器的计数器为止，有可能产生最大 2 个 fIL 时钟的误差。

注 3：在计数值发生上溢前，都能清除看门狗定时器。

注 4：如下所示，看门狗定时器在睡眠或者深度睡眠模式中的运行因选项字节（000C0H）的 bit0（WDSTBYON）的设定值而不同。

	WDSTBYON=0	WDSTBYON=1
睡眠模式	停止看门狗定时器运行。	继续看门狗定时器运行。
深度睡眠模式		

当 WDSTBYON 位为“0”时，在解除睡眠或者深度睡眠模式后重新开始看门狗定时器的计数。此时，将计数器清“0”，开始计数。

如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。

## 8.4.2 看门狗定时器上溢时间的设定

通过选项字节（000C0H）的 bit3~1（WDCS2~WDCS0）设定看门狗定时器的上溢时间。在发生上溢时，产生内部复位信号。如果在上溢时间前的窗口打开期间给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除计数并且重新开始计数。能设定的上溢时间如下所示。

表8-3看门狗定时器上溢时间的设定

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ( $f_{IL}=15\text{kHz}$ 的情况)
0	0	0	$2^6/f_{IL}$ (4.3ms)
0	0	1	$2^7/f_{IL}$ (8.5ms)
0	1	0	$2^8/f_{IL}$ (17.0ms)
0	1	1	$2^9/f_{IL}$ (34.0ms)
1	0	0	$2^{11}/f_{IL}$ (135.9ms)
1	0	1	$2^{13}/f_{IL}$ (543.5ms)
1	1	0	$2^{14}/f_{IL}$ (1086.9ms)
1	1	1	$2^{16}/f_{IL}$ (4347.8ms)

备注  $f_{IL}$ : 低速内部振荡器的时钟频率

### 8.4.3 看门狗定时器窗口打开期间的设定

通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定看门狗定时器的窗口打开期间。窗口概要如下：

- 如果在窗口打开期间给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
- 在窗口关闭期间，即使给 WDTE 寄存器写“ACH”，也会检测到异常并且产生内部复位信号。

注意：只在解除复位后第1次写WDTE寄存器时，与窗口打开期间无关，只要在上溢时间前的任意时候写WDTE，就清除看门狗定时器并且重新开始计数。

能设定的窗口打开期间如下所示。

表8-4 看门狗定时器窗口打开期间的设定

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间
0	-	禁止设定
1	0	75%
1	1	100%

注意：当选项字节（000C0H）的bit0（WDSTBYON）为“0”时，与WINDOW1位和WINDOW0位的值无关，窗口打开期间为100%。

当将上溢时间设定为 $2^9/f_{IL}$ 的情况时，窗口关闭时间和打开时间如下所示。

	窗口打开期间的设定	
	75%	100%
窗口关闭时间	0~12.8ms	无
窗口打开时间	12.8~25.6ms	0~25.6ms

<当窗口打开期间为75%时>

(1) 上溢时间：

$$2^9/F_{IL}(\text{MAX.})=2^9/20\text{kHz}(\text{MAX.})=25.6\text{ms}$$

(2) 窗口关闭时间：

$$0\sim 2^9/F_{IL}(\text{MIN.})\times(1-0.75)=0\sim 2^9/10\text{kHz}\times 0.25=0\sim 12.8\text{ms}$$

(3) 窗口打开时间：

$$2^9/F_{IL}(\text{MIN.})\times(1-0.75)\sim 2^9/F_{IL}(\text{MAX.})=12.8\sim 25.6\text{ms}$$

## 8.4.4 看门狗定时器间隔中断的设定

能通过设定选项字节（000C0H）的 bit7（WDTINT），在达到上溢时间的  $75\%+1/2f_{\text{L}}$  时产生间隔中断（INTWDTI）。

表8-5 看门狗定时器间隔中断的设定

WDTINT	看门狗定时器间隔中断的使用/不使用
0	不使用间隔中断。
1	在达到上溢时间的 $75\%+1/2f_{\text{L}}$ 时，产生间隔中断。

注意:当解除深度睡眠模式后以X1振荡时钟运行时，CPU在经过振荡稳定时间后开始运行。

如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。

备注:即使在产生看门狗定时器间隔中断后也继续计数（继续到给看门狗定时器的允许寄存器（WDTE）写“ACH”为止）。如果在上溢时间前不给WDTE寄存器写“ACH”，就产生内部复位信号。

## 8.4.5 LOCKUP期间看门狗定时器的运行

当 LOCKUP 控制寄存器 LOCKCTL 的 lockup\_rst 位设置为 1 时，一旦内核进入 LOCKUP 状态，低速内部振荡器开始振荡，看门狗定时器的计时器自动开始运行，并将上溢时间的控制位（WDCS2~WDCS0）设置为 3'b010，即设置上溢时间为 17ms。

## 第9章 除法与开方运算单元 (DIVSQRT)

### 9.1 概述

芯片包含一个 32bit/32bit 的硬件除法器与一个 32bit 的硬件开方器。

### 9.2 特性

- ◆ 支持有/无符号除法与开方运算。
- ◆ 商和余数均为 32 位宽度。
- ◆ 除法器除零标志指示位。
- ◆ 22 个 APB 时钟运算完成。
- ◆ 写 ALUB 寄存器启动运算。

### 9.3 功能说明

运算单元可以通过寄存器 DIVSQRT->CON[4]选择除法模式或者开根号模式，除法模式下寄存器 DIVSQRT->RES0 保存商，寄存器 DIVSQRT->RES1 保存余数；可以通过寄存器 DIVSQRT->CON[2]判别除数是否为 0，该位为只读位；开根号模式下 DIVSQRT->RES0 保存开根号结果，DIVSQRT->RES1 未用。

注意在开根号模式下，被开根号数最高位为 1，则被当成有符号数处理，先取绝对值，再进行开根号运算：

$$RSE0 = \sqrt[2]{absval(ALUB)}$$

同时可以通过寄存器 DIVSQRT->CON[3]判别是否运算完毕，该位为只读位，读取值为 0 表示正在运算，为 1 表示运算完毕，当除法器处于空闲状态时该位也为 1。

除法器可通过 DIVSQRT->CON[1]选择有符号还是无符号除法模式。

需要注意的是运算单元的时钟使能位在外围允许寄存器 PER12 中设置。

注意：在计算期间请不要写 ALUA 或 ALUB 寄存器，也不要读 RES0 或 RES1 寄存器，否则结果不可预知。

寄存器在不同模式下的定义：

运算单元模式	ALUA	ALUB	RES0	RES1
除法模式	被除数	除数	商	余数
开根号模式	-	被开根号数	开根号结果(低16位有效)	-



## 9.4 寄存器映射

(DIVSQRT 基地址 = 0x4006\_4480) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CON	0x000	R/W	运算单元控制寄存器	0x8
ALUA	0x004	R/W	运算单元数据A寄存器	0x0
ALUB	0x008	R/W	运算单元数据B寄存器	0x0
RES0	0x00C	RO	运算单元结果0寄存器	0x0
RES1	0x010	RO	运算单元结果1寄存器	0x0

## 9.5 寄存器说明

### 9.5.1 除法开方器控制寄存器(CON)

位	符号	描述	复位值
31:5	-	保留	-
4	MODE	运算模式选择位 0: 除法模式 1: 开根号模式	0
3	READY	运算完毕指示位 0: 运算正在进行 1: 运算完毕或空闲状态	1
2	DIVBY0	除法模式除零指示位 (写除数后自动更新该位) 0: 除数不为0 1: 除数为0	0
1	SIGN	除法模式符号选择位 0: 无符号模式 1: 有符号模式	0
0	-	保留	0

### 9.5.2 除法开方器数据A寄存器(ALUA)

位	符号	描述	复位值
31:0	ALUA	32位数据A	0x0

### 9.5.3 除法开方器数据B寄存器(ALUB)

位	符号	描述	复位值
31:0	ALUB	32位数据B	0x0

### 9.5.4 除法开方器结果0寄存器(RES0)

位	符号	描述	复位值
31:0	RES0	32位结果0	0x0

### 9.5.5 除法开方器结果1寄存器(RES1)

位	符号	描述	复位值
31:0	RES1	32位结果1	0x0

# 第10章 除法运算单元 (DIV)

## 10.1 概述

芯片包含一个 32bit/32bit 的硬件除法器。

## 10.2 特性

- ◆ 支持有/无符号除法运算。
- ◆ 商和余数均为 32 位宽度。
- ◆ 除法器除零标志指示位。
- ◆ 22 个 APB 时钟运算完成。
- ◆ 写 ALUB 寄存器启动运算。

## 10.3 功能说明

除法寄存器 DIV->RES0 保存商，寄存器 DIV->RES1 保存余数；可以通过寄存器 DIV->CON[2]判别除数是否为 0，该位为只读位；

同时可以通过寄存器 DIV->CON[3]判别是否运算完毕，该位为只读位，读取值为 0 表示正在运算，为 1 表示运算完毕，当除法器处于空闲状态时该位也为 1。

除法器可通过 DIV->CON[1]选择有符号还是无符号除法模式。

需要注意的是运算单元的时钟使能位在外围允许寄存器 PER12 中设置。

注意：在计算期间请不要写 ALUA 或 ALUB 寄存器，也不要读 RES0 或 RES1 寄存器，否则结果不可预知。

寄存器在除法模式下的定义：

运算单元模式	ALUA	ALUB	RES0	RES1
除法模式	被除数	除数	商	余数

## 10.4 寄存器映射

(DIV 基地址 = 0x4006\_44C0) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CON	0x000	R/W	运算单元控制寄存器	0x8
ALUA	0x004	R/W	运算单元数据A寄存器	0x0
ALUB	0x008	R/W	运算单元数据B寄存器	0x0
RES0	0x00C	RO	运算单元结果0寄存器	0x0
RES1	0x010	RO	运算单元结果1寄存器	0x0

## 10.5 寄存器说明

### 10.5.1 除法控制寄存器(CON)

位	符号	描述	复位值
31:4	-	保留	-
3	READY	运算完毕指示位 0: 运算正在进行 1: 运算完毕或空闲状态	1
2	DIVBY0	除法模式除零指示位 (写除数后自动更新该位) 0: 除数不为0 1: 除数为0	0
1	SIGN	除法模式符号选择位 0: 无符号模式 1: 有符号模式	0
0	-	保留	0

### 10.5.2 除法器数据A寄存器(ALUA)

位	符号	描述	复位值
31:0	ALUA	32位数据A	0x0

### 10.5.3 除法器数据B寄存器(ALUB)

位	符号	描述	复位值
31:0	ALUB	32位数据B	0x0

### 10.5.4 除法器结果0寄存器(RES0)

位	符号	描述	复位值
31:0	RES0	32位结果0	0x0

### 10.5.5 除法器结果1寄存器(RES1)

位	符号	描述	复位值
31:0	RES1	32位结果1	0x0

# 第11章 定时器 (TIMER0/1)

## 11.1 概述

包含2路可编程的32位/16位计数器，即TIMER0/TIMER1，为用户提供便捷的定时计数功能。

## 11.2 特性

- ◆ 可配置 32 位/16 位向下计数器。
- ◆ 每个定时器都有独立预分频器。
- ◆ 提供单次触发，周期计数，连续计数三种计数操作模式。
- ◆ 支持芯片从睡眠模式唤醒。

## 11.3 功能描述

### 11.3.1 单次触发模式

如果定时器工作在单次触发模式，使能定时器后，计数器从加载寄存器加载初值，向下计数，当计数器递减到0时，停止工作，同时产生中断。若要再次启动单次触发模式，需清零TMROS位，再置位TMROS位。

(再次启动单次触发模式时，需注意TMROS位清零时，保持为0的时间须大于一个定时器计数时钟周期)

### 11.3.2 周期计数模式

如果定时器工作在周期计数模式，使能定时器后，计数器从加载寄存器加载初值，向下计数，当计数器递减到0时，计数器从加载寄存器加载初值，并继续计数，同时产生中断。

### 11.3.3 连续计数模式

如果定时器工作在连续计数模式，使能定时器后，计数器从加载寄存器加载初值，向下计数，当计数器递减到0时，计数器加载最大值作为初值，并继续计数，同时产生中断。

### 11.3.4 延迟加载功能

当数据写入延迟加载寄存器时，数据在下一个TIMER\_CLK上升沿写入加载寄存器，若计数器已经开始计数，则会等待当前周期计数为0，再从加载寄存器中加载初值。

## 11.4 寄存器映射

(Timer0基地址= 0x4006\_1000, Timer1基地址= 0x4006\_1100) RO: 只读; WO: 只写; R/W: 读写;

寄存器	偏移量	读/写	描述	复位值
CON	0x000	R/W	定时器控制寄存器	0x0
LOAD	0x004	R/W	定时器加载寄存器	0x0
VAL	0x008	RO	定时器当前值寄存器	0xFFFFFFFF
RIS	0x00C	RO	定时器中断源状态寄存器	0x0
MIS	0x010	RO	定时器已使能中断状态寄存器	0x0
ICLR	0x014	WO	定时器中断清零寄存器	-
BGLOAD	0x018	R/W	定时器延迟加载寄存器	0x0

## 11.5 寄存器说明

### 11.5.1 定时器控制寄存器 (CON0/1)

位	符号	描述	复位值
31:8	-	保留	-
7	TMREN	定时器使能位 0: 禁止 1: 使能	0
6	TMRMS	定时器模式选择位 0: 连续计数模式 1: 周期计数模式	0
5	TMRIE	定时器中断使能位 0: 禁止中断 1: 使能中断	0
4	-	保留	-
3:2	TMRPRE	定时器预分频 00: 1分频 01: 16分频 10: 256分频 11: 保留	0x0
1	TMRSZ	定时器计数位数选择 0: 16位计数器 1: 32位计数器	0
0	TMROS	单次触发模式选择位 0: 模式由TMRMS位确定 1: 单次触发模式 (再次触发单次模式, 计数初始值由TMRMS位确定)	0

### 11.5.2 定时器加载寄存器 (LOAD0/1)

位	符号	描述	复位值
31:0	LOAD	定时器加载寄存器	0x0

### 11.5.3 定时器当前值寄存器 (VAL0/1)

位	符号	描述	复位值
31:0	VAL	定时器当前计数值	0xFFFFFFFF

### 11.5.4 定时器中断源状态寄存器 (RIS0/1)

位	符号	描述	复位值
31:1	-	保留	-
0	RIS	定时器中断源状态 1: 产生中断 0: 未产生中断	0

### 11.5.5 定时器已使能中断状态寄存器 (MIS0/1)

位	符号	描述	复位值
31:1	-	保留	-
0	MIS	定时器已使能中断状态位 1: 中断使能并产生中断 0: 未产生中断	0

### 11.5.6 定时器中断清零寄存器 (ICLR0/1)

位	符号	描述	复位值
31:0	ICLR	写入任意数, 清零定时器中断	-

### 11.5.7 定时器延迟加载寄存器 (BGLOAD0/1)

位	符号	描述	复位值
31:0	BGLOAD	定时器延迟加载寄存器 (读取值为最近一次写入LOAD或BGLOAD的值)	0x0

# 第12章 捕捉/比较/脉宽调制模块(CCP0/1)

## 12.1 概述

包含2组CCP模块CCP0/CCP1，每组CCP对应A，B两路通道。CCP0对应CCP0A/CCP0B，CCP1对应CCP1A/CCP1B。

## 12.2 特性

- ◆ 共两组 CCP，每组支持 2 路 PWM 输出。
- ◆ 每组 CCP 都可设置独立的周期。
- ◆ CCP0 内部有 16 位计数器，可产生比较/溢出中断。
- ◆ CCP1 内部有 32 位计数器，可产生比较/溢出中断。
- ◆ CCPn 具有独立捕捉功能，可选在 A 路或 B 路管脚输入信号。
- ◆ CCP1 具有 4 通道捕捉功能，可同时捕获 CCP0A/CCP0B/CCP1A/CCP1B 输入信号。
- ◆ 捕捉模式 1 下支持捕获操作重新加载 CCP0 计数器功能。
- ◆ 内部通道 CAP3 支持模拟比较器输出捕获功能。
- ◆ 内部通道 CAP0-CAP3 支持软件捕获功能。



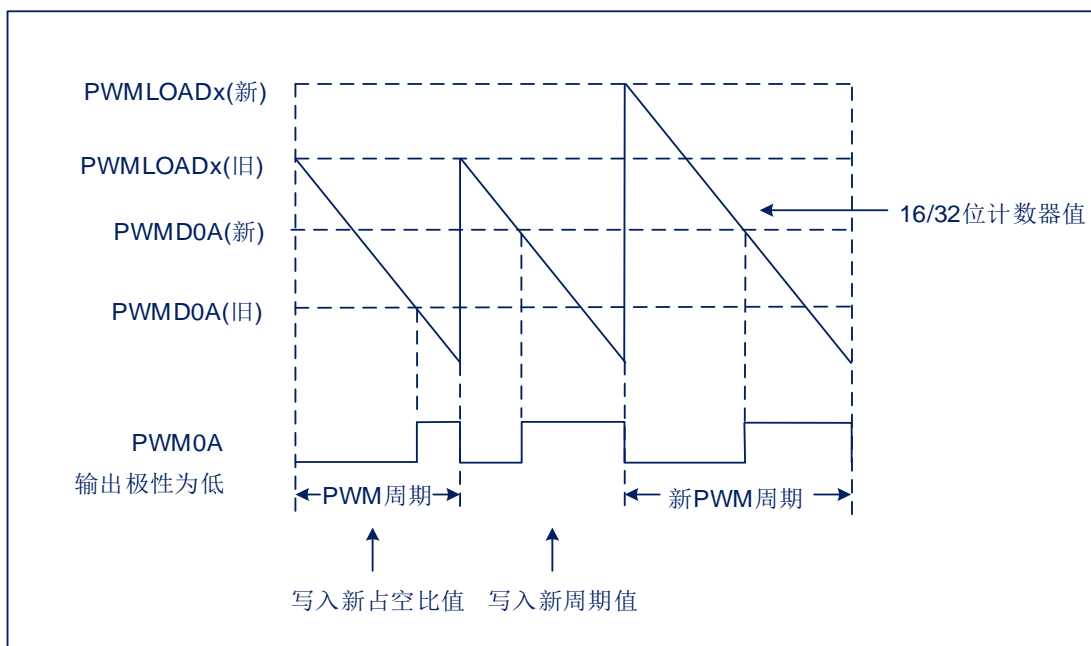
## 12.3 功能描述

### 12.3.1 脉宽调制模式 (PWM)

每组CCP可输出A、B两路PWM: PWMxA, PWMxB, 这两路共用一个周期, 输出占空比可以通过CCPxDA, CCPxDB独立设置。PWMxA/PWMxB输出极性可通过PWMxAOP/PWMxBOP位设置, 且分别对应CCPxA/CCPx B通道输出。

当CCP0RUN置1后, 16位计数器加载CCP0的计数初值, 向下计数; 当CCP1RUN置1后, 32位计数器加载CCP1的计数初值, 向下计数。当计数值等于CCPxDA/B的值或计数溢出时, PWMxA/PWMxB输出电平发生改变。

图 12-1: PWM 时序图



周期和占空比计算方式如下:

周期=CCPxLOAD×CCP时钟周期。

PWMxA占空比=CCPxDA/CCPxLOAD (支持0%~100%)。

PWMxB占空比=CCPxDB/CCPxLOAD (支持0%~100%)。

CCPxLOAD=0时, PWMxA, PWMxB占空比为0%。

CCPxDA/CCPxDA ≥ CCPxLOAD时, 占空比为100%。

## 12.3.2 方波输出模式

方波输出模式属于脉宽调制模式的一种，该模式周期自由可调，占空比固定为50%。

每组CCP均可设置A路或B路为方波输出模式。设置方波输出模式时需要配置PWM模式，且将CCPxCON.ZAEN或CCPxCON.ZBEN置位。周期由CCPx计数初值决定，占空比默认加载CCPxLOAD值的一半（最低位忽略）。

## 12.3.3 捕捉模式0

该捕捉模式为外部捕捉。

CCP0可设置A路或B路作为外部捕捉信号管脚，CCP0RUN置位后，16位计数从0xFFFF或CCP0LOAD[15:0]开始向下计数，当触发捕捉条件时，计数器停止计数，CCP0A或CCP0B返回当前计数器的值。若需要进行下一次捕捉，需将CCP0RUN清零，再置位。

捕捉时间计算方式为：

CCP0LOAD[15:0]=0或1，捕捉时间=  $(0xFFFF - CCP0DA/B) \times CCPx$ 时钟周期。

CCP0LOAD[15:0]=其他，捕捉时间=  $(CCP0LOAD[15:0] - CCPxDA/B) \times CCPx$ 时钟周期。

CCP1可设置A路或B路作为外部捕捉信号管脚，CCP1RUN置位后，32位计数从0xFFFFFFFF或CCP1LOAD[31:0]开始向下计数，当触发捕捉条件时，计数器停止计数，CCP1A或CCP1B返回当前计数器的值。若需要进行下一次捕捉，需将CCP1RUN清零，再置位。

捕捉时间计算方式为：

CCP1LOAD[31:0]=0或1，捕捉时间=  $(0xFFFFFFFF - CCP1DA/B) \times CCPx$ 时钟周期。

CCP1LOAD[31:0]=其他，捕捉时间=  $(CCP1LOAD[31:0] - CCP1DA/B) \times CCPx$ 时钟周期。

注：若设置重载使能，建议重载值不要设置为1。

### 12.3.4 捕捉模式1

CCP1包括4路内部通道：CAP0，CAP1，CAP2，CAP3。任意一路通道可选择外部通道中ECAP00-03或ECAP10-13中任意一路作为捕获通道。也可分别选择CCP0A/CCP0B/CCP1A/CCP1B作为捕获通道。

ECAP00-03对应模拟比较器0的正端输入C0P0-C0P3。

ECAP10-13对应模拟比较器1的正端输入C1P0-C1P3。

使用ECAP外部捕获时需要将相应的端口设置为GPIO功能。

使用CCP0A/CCP0B/CCP1A/CCP1B捕获时，需要将相应口设置为CCP口。

CAPn与外部通道的对应关系：

内部通道	外部通道
CAP0	CAP0CHS=n: 选择ECAP0n (n=0-3)@ECAPS=0 CAP0CHS=n: 选择ECAP1n (n=0-3)@ECAPS=1 CAP0CHS=F: 选择CAP0A CAP0CHS=其他值: 保留
CAP1	CAP1CHS=n: 选择ECAP0n (n=0-3)@ECAPS=0 CAP1CHS=n: 选择ECAP1n (n=0-3)@ECAPS=1 CAP1CHS=F: 选择CAP0B CAP1CHS=其他值: 保留
CAP2	CAP2CHS=n: 选择ECAP0n (n=0-3)@ECAPS=0 CAP2CHS=n: 选择ECAP1n (n=0-3)@ECAPS=1 CAP2CHS=F: 选择CAP1A CAP2CHS=其他值: 保留
CAP3	CAP3CHS=n: 选择ECAP0n (n=0-3)@ECAPS=0 CAP3CHS=n: 选择ECAP1n (n=0-3)@ECAPS=1 CAP3CHS=8: 选择ACMP0滤波选择后输出 CAP3CHS=9: 选择ACMP1滤波选择后输出 CAP3CHS=F: 选择CAP1B CAP3CHS=其他值: 保留

在捕获模式1下，CCP0与CCP1的PWM模式输出与外部捕捉模式0禁止。

该模式需要CCP1工作在计数模式下，捕获操作将CCP1计数值装入到相关寄存器中。

另外CCP0可选择工作在计数模式下，可分别设置CAP0-CAP3捕获触发加载功能。即设置的通道有捕获操作产生时，将重新加载CCP0的计数器初值。多个通道可同时设置该功能，软件触发捕捉则不会重新加载CCP0的初值。

在捕获模式1下，CCP0与CCP1的比较/溢出中断功能可正常使用。

该捕捉方式分为两种：一种外部信号触发捕捉，一种为软件触发捕捉。

1) 外部信号触发捕捉：

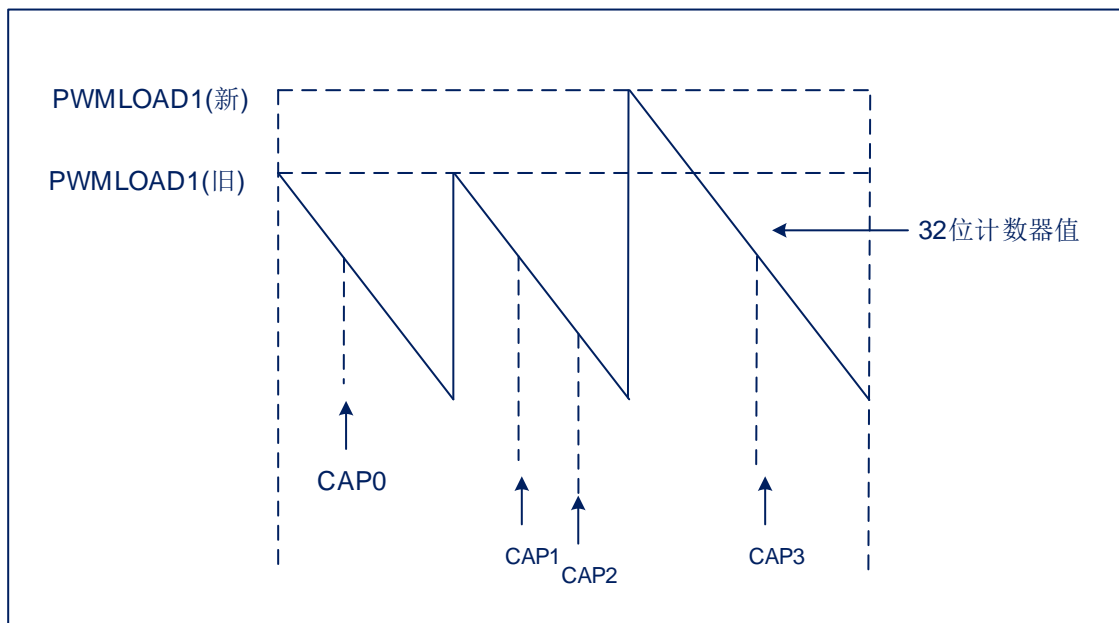
CAP0-CAP3均可选择上升沿/下降沿/双沿捕获。当产生信号时，将CCP1计数器的值捕获到相应的寄存器中，且产生中断标志。4个通道与捕获寄存器的对应关系如下：

CAP0/CAP1/CAP2/CAP3分别对应CAP0DATA/CAP1DATA/CAP2DATA/CAP3DATA寄存器。

2) 软件触发捕捉：

分别对CAPCON2[31:16]，CAPCON2[15:0]，CAPCON3[31:16]，CAPCON3[15:0]写0x55AA操作，则分别对CAP0-CAP3通道产生捕获操作，将CCP1计数器的值捕获到相应的寄存器中。软件触发捕捉不会产生中断标志。

图 12-2: CAP0-CAP3 通道捕获操作



### 12.3.5 捕捉模式2

该捕捉模式为外部捕捉。该模式主要用来捕捉外部输入的PWM波形信息。

在捕捉模式下2，CAP2，CAP3的通道映射到CAP1，即CAP1-3为同一捕获通道，CAP0禁止使用。

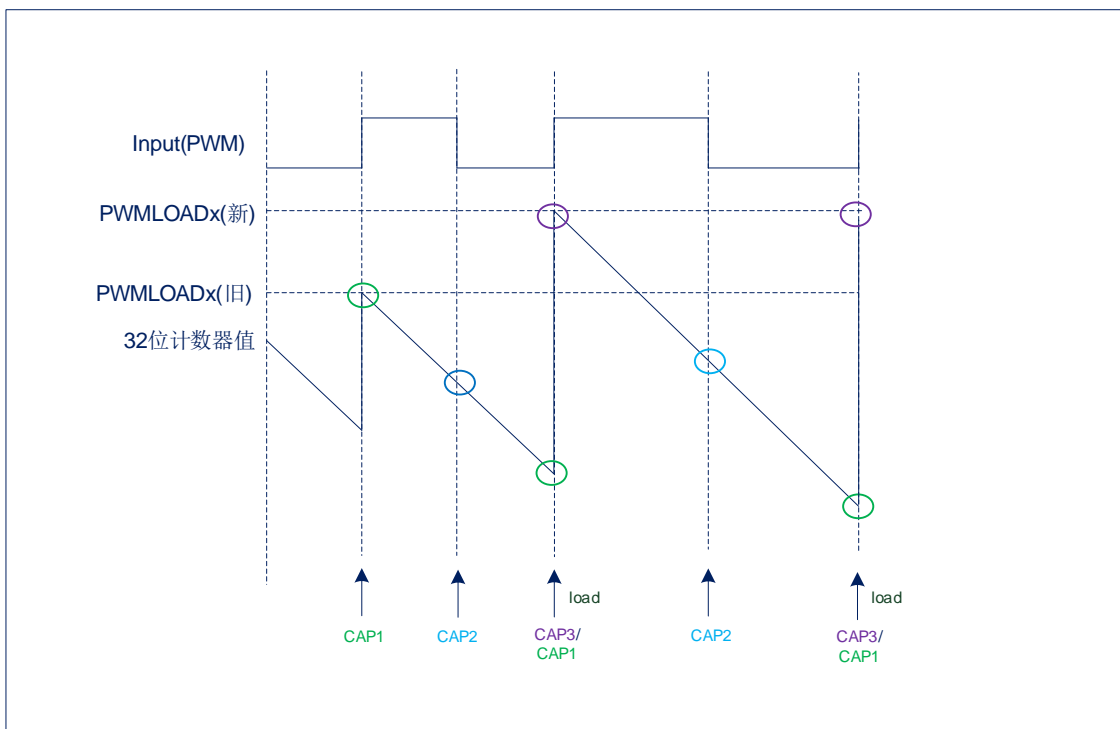
CCP0可自由设置，不受影响。

CCP1工作在计数模式下，CCP1RUN置位后，32位计数器从初始值开始向下计数。如果溢出且没有产生捕获操作，则计数器重新加载初值继续向下计数。

启动捕捉模式2的操作如下：

- (1) CAP1发生捕获之后，CCP1计数器重新加载初值，同时将该值装载到CAP1DATA，允许CAP2产生捕获；
- (2) CAP2发生捕获之后，将捕获的计数器的值装载到CAP2DATA，允许CAP3发生捕获，
- (3) CAP3发生捕获之后，将捕获的计数器的值装载到CAP3DATA，完成整次捕获。此时将CAP1DATA-CAP3DATA的值装入CAP0DATA[31:0]，将CAP1DATA-CAP2DATA的值装入CAPDUTY[31:0]。然后CAP1产生捕获动作，回到第(1)步。

图 12-3：捕捉模式 2 捕获操作



注1：如果完成CAP3产生的捕获的时间超过了CCP1计数器的一个周期，将会发生溢出，计算出的CAP0DATA 和CAPDUTY 的值是不准确的。建议设置计数器的周期远大于需要捕获的PWM周期。

注2：CAP1到CAP2，CAP2到CAP3捕获的间隔时间需要大于8个CCP1计数值。

注3：捕捉模式2支持CAP1-3的软件捕获动作。

注4：CAP1-CPA3共用同一个捕获通道，捕获操作会同时在CAP1-CAP3上产生相应的捕获标志位。

### 12.3.6 PWM配置过程

- 配置 PWM 控制寄存器，设置预分频，选择 PWM 模式，使能 PWM。
- 配置 PWM 周期，写入 CCPxLOAD 寄存器。
- 配置 PWM 占空比，写入 CCPxDA/CCPxDB 寄存器。
- 若需要中断，使能相关中断位，清零中断状态寄存器。
- 设置相应 I/O 口为 PWM 输出。
- 设置 PWM 运行寄存器，开始输出。

### 12.3.7 中断

在PWM模式下，CCPx可产生两种中断：

- 当计数器递减到 0 时，产生溢出中断。
- 当计数器的值与 CCPxDA 或 CCPxDB 的值相等时，产生比较中断。

在捕捉模式0/1/2下，可产生两种中断：

- 当计数器递减到 0 时，产生溢出中断。
- 触发捕捉条件时，产生捕捉中断。

## 12.4 寄存器映射

(CCP基地址 = 0x4006\_4300)

RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
CCP0CON <sub>(P1B)</sub>	0x000	R/W	CCP0控制寄存器	0x0
CCP0LOAD <sub>(P1A)</sub>	0x004	R/W	CCP0重加载寄存器	0x0
CCP0DA <sub>(P1A)</sub>	0x008	R/W	CCP0通道A数据寄存器	0x0
CCP0DB <sub>(P1A)</sub>	0x00c	R/W	CCP0通道B数据寄存器	0x0
CCP1CON <sub>(P1B)</sub>	0x010	R/W	CCP1控制寄存器	0x0
CCP1LOAD <sub>(P1A)</sub>	0x014	R/W	CCP1重加载寄存器	0x0
CCP1DA <sub>(P1A)</sub>	0x018	R/W	CCP1通道A数据寄存器	0x0
CCP1DB <sub>(P1A)</sub>	0x01C	R/W	CCP1通道B数据寄存器	0x0
CCPIMSC <sub>(P1B)</sub>	0x040	R/W	CCP中断使能寄存器	0x0
CCPRIS	0x044	RO	CCP中断源状态寄存器	0x0
CCPMIS	0x048	RO	CCP已使能中断状态寄存器	0x0
CCPICLR	0x04C	WO	CCP中断清零寄存器	0x0
CCPRUN <sub>(P1B)</sub>	0x050	R/W	CCP运行寄存器	0x0
CCPLOCK	0x054	R/W	CCP0/1写使能寄存器	0x0
CAPCON <sub>(P1B)</sub>	0x058	R/W	捕获控制寄存器	0x0
CAPCON2 <sub>(P1B)</sub>	0x05C	R/W	捕获控制寄存器2	0x0
CAPCON3 <sub>(P1B)</sub>	0x060	R/W	捕获控制寄存器3	0x0
CAPCHS <sub>(P1B)</sub>	0x064	R/W	捕获通道选择寄存器	0x0
CAP0DATA <sub>(P1A)</sub>	0x068	RO	捕获通道0数据寄存器	0x0
CAP1DATA <sub>(P1A)</sub>	0x06C	RO	捕获通道1数据寄存器	0x0
CAP2DATA <sub>(P1A)</sub>	0x070	RO	捕获通道2数据寄存器	0x0
CAP3DATA <sub>(P1A)</sub>	0x074	RO	捕获通道3数据寄存器	0x0
CAPDUTY <sub>(P1A)</sub>	0x078	RO	捕获模式2捕获占空比寄存器	0x0

注1: (P1A/P1B)标注的寄存器为被保护的寄存器。

注2: (P1A): LOCK=55H或AAH时, 标注的寄存器允许写入; =其他值, 禁止写入。

注3: (P1B): LOCK=55H时, 标注的寄存器允许写入; =其他值, 禁止写入。

## 12.5 寄存器说明

### 12.5.1 CCP0控制寄存器 (CCP0CON)

位	符号	描述	复位值
31:10	-	保留	-
9	CCP0ZBEN	通道B的方波模式使能位 (PWM模式有效) 0: 禁止 1: 使能, 占空比加载值为LOAD0/2	0
8	CCP0ZAEN	通道A的方波模式使能位 (PWM模式有效) 0: 禁止 1: 使能, 占空比加载值为LOAD0/2	0
7	-	保留	-
6	CCP0EN	CCP0使能位 0: 禁止 1: 使能	0
5:4	CCP0PS	CCP0预分频选择 0x0: PCLK 0x1: PCLK/4 0x2: PCLK/16 0x3: PCLK/64	0x0
3	CCP0MS	CCP0模式选择 0: 捕捉模式0 (CAPEN=0时生效) 1: PWM模式 (CAPEN=0时生效)	0
2	CCP0CM0CS	CCP0捕捉模式0捕捉通道选择 0: 通道CCP0A 1: 通道CCP0B	0
1:0	CCP0CM0ES	CCP0捕捉模式0捕捉方式选择 0x0: CCP0RUN=1开始计数, 上升沿捕捉并产生中断 0x1: CCP0RUN=1开始计数, 下降沿捕捉并产生中断 0x2: 上升沿开始计数, 下降沿捕捉并产生中断 0x3: 下降沿开始计数, 上升沿捕捉并产生中断	0x0



## 12.5.2 CCP0重加载寄存器 (CCP0LOAD)

位	符号	描述	复位值
31:17	-	保留	-
16	RELOAD	PWM模式下:重加载使能位 0: 计数器重加载值为0xFFFF 1: 计数器重加载值为CCP0LOAD 捕捉模式下: 0: 计数器重加载值为0xFFFF 1: 计数器重加载值为CCP0LOAD	0
15:0	CCP0LOAD	CCP0计数器的加载值 (建议加载值不为0)	0x0

## 12.5.3 CCP0A数据寄存器 (CCP0DA)

位	符号	描述	复位值
31:17	-	保留	-
16	PWM0AOP	PWM0A输出极性选择 0: 正常输出 1: 反相输出	0
15:0	CCP0ADATA	PWM模式时: PWM0A的占空比 捕捉模式0时: 捕捉结果	0x0

## 12.5.4 CCP0B数据寄存器 (CCP0DB)

位	符号	描述	复位值
31:17	-	保留	-
16	PWM0BOP	PWM0B输出极性选择 0: 正常输出 1: 反相输出	0
15:0	CCP0BDATA	PWM模式时: PWM0B的占空比 捕捉模式0时: 捕捉结果	0x0

## 12.5.5 CCP1控制寄存器 (CCP1CON)

位	符号	描述	复位值
31:15	-	保留	-
14	PWM1AOP	PWM1A输出极性选择 0: 正常输出 1: 反相输出	0
13	PWM1BOP	PWM1B输出极性选择 0: 正常输出 1: 反相输出	0
12	RELOAD	PWM模式下:重加载选择位 0: 计数器重加载值为0xFFFFFFFF 1: 计数器重加载值为CCP1LOAD 捕捉模式0下: 0: 计数器重加载值为0xFFFFFFFF 1: 计数器重加载值为CCP1LOAD	0
11:10	-	保留	-
9	CCP1ZBEN	通道B的方波模式使能位 (PWM模式有效) 0: 禁止 1: 使能, 占空比加载值为LOAD1/2	0
8	CCP1ZAEN	通道A的方波模式使能位 (PWM模式有效) 0: 禁止 1: 使能, 占空比加载值为LOAD1/2	0
7	-	保留	-
6	CCP1EN	CCP1使能位 0: 禁止 1: 使能	0
5:4	CCP1PS	CCP1预分频选择 0x0: PCLK 0x1: PCLK/4 0x2: PCLK/16 0x3: PCLK/64	0x0
3	CCP1MS	CCP1模式选择 0: 捕捉模式0 (CAPEN=0时生效) 1: PWM模式 (CAPEN=0时生效)	0
2	CCP1CM0CS	CCP1捕捉模式0捕捉通道选择 0: 通道CCP1A 1: 通道CCP1B	0
1:0	CCP1CM0ES	CCP1捕捉模式0捕捉方式选择 0x0: CCP1RUN=1开始计数, 上升沿捕捉并产生中断 0x1: CCP1RUN=1开始计数, 下降沿捕捉并产生中断 0x2: 上升沿开始计数, 下降沿捕捉并产生中断 0x3: 下降沿开始计数, 上升沿捕捉并产生中断	0x0

## 12.5.6 CCP1重加载寄存器 (CCP1LOAD)

位	符号	描述	复位值
31:0	CCP1LOAD	CCP1计数器的加载值 (建议加载值不为0)	0x0

## 12.5.7 CCP1A数据寄存器 (CCP1DA)

位	符号	描述	复位值
31:0	CCP1ADATA	PWM模式时: PWM1A的占空比 捕捉模式0时: 捕捉结果	0x0

## 12.5.8 CCP1B数据寄存器 (CCP1DB)

位	符号	描述	复位值
31:0	CCP1BDATA	PWM模式时: PWM1B的占空比 捕捉模式0时: 捕捉结果	0x0

## 12.5.9 CCP中断使能寄存器 (CCPIMSC)

位	符号	描述	复位值
31:12	-	保留	-
11	CAP3IMSC	CAP3捕获中断使能位 0: 禁止 1: 使能	0
10	CAP2IMSC	CAP2捕获中断使能位 0: 禁止 1: 使能	0
9	CAP1IMSC	CAP1捕获中断使能位 0: 禁止 1: 使能	0
8	CAP0IMSC	CAP0捕获中断使能位 0: 禁止 1: 使能	0
7:6	-	保留	0x0
5	CCPIMSC5	CCP1溢出中断使能位 0: 禁止 1: 使能	0
4	CCPIMSC4	CCP0溢出中断使能位 0: 禁止 1: 使能	0
3:2	-	保留	-
1	CCPIMSC1	CCP1比较/捕捉中断使能位 0: 禁止 1: 使能	0
0	CCPIMSC0	CCP0比较/捕捉中断使能位 0: 禁止 1: 使能	0

## 12.5.10 CCP中断源状态寄存器 (CCPRIS)

位	符号	描述	复位值
31:12	-	保留	-
11	CAP3RIS	CAP3捕获中断状态位 1: 产生中断 0: 未产生中断	0
10	CAP2RIS	CAP2捕获中断状态位 1: 产生中断 0: 未产生中断	0
9	CAP1RIS	CAP1捕获中断状态位 1: 产生中断 0: 未产生中断	0
8	CAP0RIS	CAP0捕获中断状态位 1: 产生中断 0: 未产生中断	0
7:6	-	保留	0x0
5	CCPRIS5	CCP1溢出中断状态位 1: 产生中断 0: 未产生中断	0
4	CCPMRIS4	CCP0溢出中断状态位 1: 产生中断 0: 未产生中断	0
3:2	-	保留	-
1	CCPRIS1	CCP1比较/捕捉中断状态位 1: 产生中断 0: 未产生中断	0
0	CCPRIS0	CCP0比较/捕捉中断状态位 1: 产生中断 0: 未产生中断	0

## 12.5.11 CCP已使能中断状态寄存器 (CCPMIS)

位	符号	描述	复位值
31:12	-	保留	-
11	CAP3MIS	CAP3已使能捕获中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
10	CAP2MIS	CAP2已使能捕获中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
9	CAP1MIS	CAP1已使能捕获中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
8	CAP0MIS	CAP0已使能捕获中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
7:6	-	保留	-
5	CCPMIS5	CCP1已使能溢出中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
4	CCPMIS4	CCP0已使能溢出中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
3:2	-	保留	-
1	CCPMIS1	CCP1已使能比较/捕捉中断状态位 1: 中断使能并产生中断 0: 未产生中断	0
0	CCPMIS0	CCP0已使能比较/捕捉中断状态位 1: 中断使能并产生中断 0: 未产生中断	0

### 12.5.12 CCP中断清零寄存器 (CCPICLR)

位	符号	描述	复位值
31:12	-	保留	-
11	CAP3ICLR	写1清除CAP3捕获中断状态位	0
10	CAP2ICLR	写1清除CAP2捕获中断状态位	0
9	CAP1ICLR	写1清除CAP1捕获中断状态位	0
8	CAP0ICLR	写1清除CAP0捕获中断状态位	0
7:6	-	保留	-
5	CCPICLR5	写1清除CCP1溢出中断状态位	0
4	CCPICLR4	写1清除CCP0溢出中断状态位	0
3:2	-	保留	-
1	CCPICLR1	写1清除CCP1比较/捕捉中断状态位	0
0	CCPICLR0	写1清除CCP0比较/捕捉中断状态位	0

### 12.5.13 CCP运行寄存器 (CCPRUN)

位	符号	描述	复位值
31:2	-	保留	-
1	CCP1RUN	CCP1运行控制位 0: 停止 1: 运行	0
0	CCP0RUN	CCP0运行控制位 0: 停止 1: 运行	0

### 12.5.14 CCP写使能控制寄存器(LOCK)

位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0xaa时, 使能操作保护级别为P1A的寄存器; 当LOCK=0x55时, 使能操作保护级别为P1B与P1A的寄存器; 当LOCK=其他值时, 禁止操作有保护级别的寄存器。	0x0

## 12.5.15 CCP CAP控制寄存器 (CAPCON)

位	符号	描述	复位值
31:14	-	保留	-
13	CAPEN2	捕捉模式2使能位 (仅对CCP1有效) 0: -- 1: 捕捉模式2使能位, 同时禁止捕捉模式1	0
12	CAPEN	捕捉模式1使能位 0: CCP0/CCP1为PWM模式或捕获模式0使能 1: 捕获模式1使能, 即全通道捕获模式 CCP0可设置为连续计数模式 CCP1可设置为连续计数模式	0
11	CAP3RLEN	捕捉模式1下CAP3捕获触发CCP0的计数器加载使能位 0: 禁止 1: 使能, (需要在捕获模式1, 且CCP0运行状态下生效) CAP3出现捕获触发信号, 则CCP0在计数器的运行过程中, 将重新加载CCP0计数初值。	0
10	CAP2RLEN	捕捉模式1下CAP2捕获触发CCP0的计数器加载使能位 0: 禁止 1: 使能, (需要在捕获模式1, 且CCP0运行状态下生效) CAP2出现捕获触发信号, 则CCP0在计数器的运行过程中, 将重新加载CCP0计数初值。	0
9	CAP1RLEN	捕捉模式1下CAP1捕获触发CCP0的计数器加载使能位 0: 禁止 1: 使能, (需要在捕获模式1, 且CCP0运行状态下生效) CAP1出现捕获触发信号, 则CCP0在计数器的运行过程中, 将重新加载CCP0计数初值。	0
8	CAP0RLEN	捕捉模式1下CAP0捕获触发CCP0的计数器加载使能位 0: 禁止 1: 使能, (需要在捕获模式1, 且CCP0运行状态下生效) CAP0出现捕获触发信号, 则CCP0在计数器的运行过程中, 将重新加载CCP0计数初值。	0
7:6	CAP3ES	CAP3捕捉模式选择 0x0: 禁止 0x1: 上升沿捕捉 0x2: 下降沿捕捉 0x3: 双沿	0x0
5:4	CAP2ES	CAP2捕捉模式选择 0x0: 禁止 0x1: 上升沿捕捉 0x2: 下降沿捕捉 0x3: 双沿	0x0
3:2	CAP1ES	CAP1捕捉模式选择 0x0: 禁止 0x1: 上升沿捕捉 0x2: 下降沿捕捉 0x3: 双沿	0x0
1:0	CAP0ES	CAP0捕捉模式选择	0x0

		0x0: 禁止 0x1: 上升沿捕捉 0x2: 下降沿捕捉 0x3: 双沿	
--	--	--	--



## 12.5.16 CCP CAP通道选择寄存器 (CAPCHS)

位	符号	描述	复位值
31:17	-	保留	-
16	ECAPS	ECAP捕获通道组选择 0: 选择ECAP00-ECAP03 1: 选择ECAP10-ECAP13	0
15:12	CAP3CHS	CAP3捕捉通道选择 (x=0或1, 由ECAPS决定) 0x0: ECAPx0 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: 禁止 0x5: 禁止 0x8: ACMP0的输出 (非事件输出) 0x9: ACMP1的输出 (非事件输出) 0xF: CCP1B 其他 值: 保留	0x0
11:8	CAP2CHS	CAP2捕捉通道选择 (x=0或1, 由ECAPS决定) 0x0: ECAPx0 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: 禁止 0x5: 禁止 0xF: CCP1A 其他 值: 保留	0x0
7:4	CAP1CHS	CAP1捕捉通道选择 (x=0或1, 由ECAPS决定) 0x0: ECAPx0 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: 禁止 0x5: 禁止 0xF: CCP0B 其他 值: 保留	0x0
3:0	CAP0CHS	CAP0捕捉通道选择 (x=0或1, 由ECAPS决定) 0x0: ECAPx0 0x1: ECAPx1 0x2: ECAPx2 0x3: ECAPx3 0x4: 禁止 0x5: 禁止 0xF: CCP0A 其他 值: 保留	0x0

### 12.5.17 CCP1 CAP控制寄存器2 (CAPCON2)

位	符号	描述	复位值
31:16	-	读: 写入值 写: 0x55aa, 产生CAP0的捕获操作 写: 其他值, 则无效	0x0
15:0	-	读: 写入值 写: 0x55aa, 产生CAP1的捕获操作 写: 其他值, 则无效	0x0

### 12.5.18 CCP1 CAP控制寄存器3 (CAPCON3)

位	符号	描述	复位值
31:16	-	读: 写入值 写: 0x55aa, 产生CAP2的捕获操作 写: 其他值, 则无效	0x0
15:0	-	读: 写入值 写: 0x55aa, 产生CAP3的捕获操作 写: 其他值, 则无效	0x0

### 12.5.19 CCP1 CAP0数据寄存器 (CAP0DATA)

位	符号	描述	复位值
31:0	CAP0DATA/ CAPYDATA	读: 捕获模式2: 捕获完成后, 存放CAPYDATA =CAP1的捕获值-CAP3的捕获值(周期) 其他: 为CAP0捕捉CCP1计数器的32bit值 写: 无效	0x0

### 12.5.20 CCP1 CAPn数据寄存器 (CAPnDATA) (n=1-3)

位	符号	描述	复位值
31:0	CAPnDATA	读: 为CAPn捕捉CCP1计数器的32bit值 写: 无效	0x0

### 12.5.21 CCP1 捕获模式2下的捕获占空比寄存器 (CAPDUTY)

位	符号	描述	复位值
31:0	CAPDUTY	读: 为CCP1捕获模式2下的占空比 写: 无效	0x0

# 第13章 HALL 信号处理模块

## 13.1 概述

HALL信号处理模块主要用于处理传感器的输入信号，支持3路HALL传感器信号输入处理。

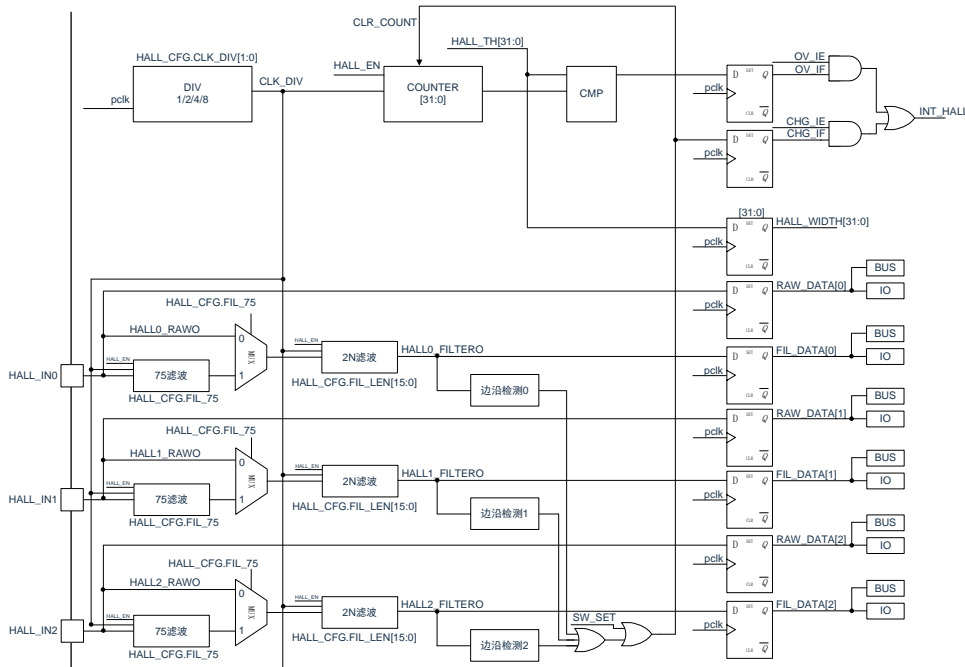
## 13.2 特性

- 支持3路HALL传感器输入信号
- 3路信号之间相互独立
- 每路信号均包含两级滤波器
- 支持HALL信号捕获，记录信号变化时间并输出中断
- 支持软件触发信号变化中断
- 32位独立计数器，支持溢出中断
- 支持实时输入信号与滤波结果输出

## 13.3 功能描述

### 13.3.1 功能框图

HALL模块的数据流程如下图所示，PCLK与系统主时钟保持一致。



### 13.3.2 时钟分频

HALL模块工作频率可以选择系统主时钟的1/2/4/8分频，其中滤波器和32位独立计数器均在此频率下工作。

### 13.3.3 信号输入

HALL传感器的输出信号通过GPIO输入到HALL信号处理模块，芯片对于每一路HALL信号输入通道均有2个GPIO可供选择，用户可以选择其中一个作为HALL信号的输入通道。

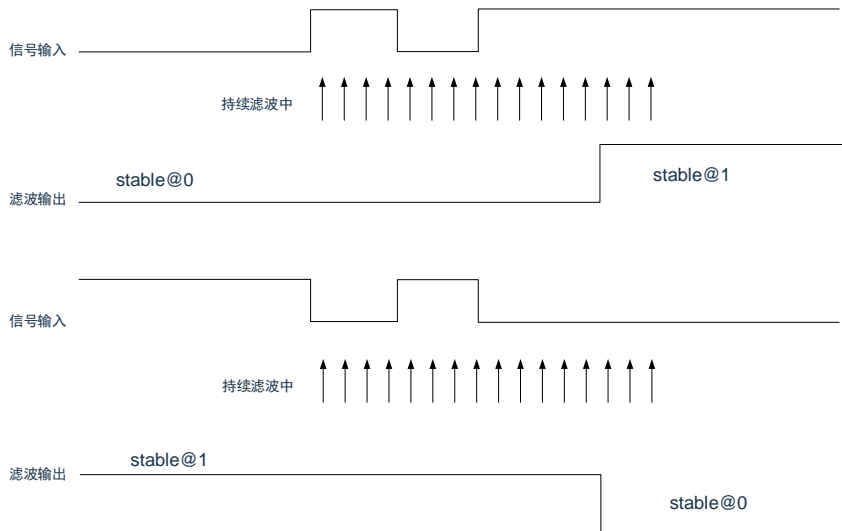
详细管脚位置说明见引脚功能章节。

### 13.3.4 滤波器

滤波器模块包含了两级滤波器，主要功能是去除HALL信号上的毛刺。

#### 第一级滤波器（7判5滤波）

连续 7 个采样点中，如果达到超过5个1则输出1，如果达到或超过5个0则输出0，否则输出保持上一次的滤波结果。具体如下图所示：



#### 第二级滤波器（连续滤波）

连续 N 个采样点中，如果全为 0 则输出 0，如果全为 1 则输出 1，否则输出保持上一次的滤波结果，滤波宽度可通过HALL\_CFG[15:0]设置。

### 13.3.5 捕获

独立的32位计数器可用于测量连续两次HALL信号变化之间的时间宽度。

32位计数器（HALL\_CNT）从0开始计数，当发生HALL信号变化时，将此刻的计数器的值保存到HALL\_WIDTH寄存器，将此刻的HALL信号保存在HALL\_INFO.FIT\_DATA，输出HALL信号变化中断，计数器重新从0开始计数。

向HALL\_CNT写入任意值，可清除计数值，计数器重新从0开始计数。

当计数器的计数值达到设置的计数门限值（HALL\_TH）时，产生计数器溢出中断，计数器重新从0开始计数。

## 13.4 寄存器映射

(HALL基地址 = 0x4006\_4500)

RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
CFG <sub>(P1B)</sub>	0x000	R/W	HALL模块配置寄存器	0x0
INFO <sub>(P1B)</sub>	0x004	RO	HALL模块信息寄存器	0x0
CLRF <sub>(P1B)</sub>	0x008	WO	HALL中断标志清0寄存器	0x0
WIDTH <sub>(P1B)</sub>	0x00C	RO	HALL宽度计数值寄存器	0x0
TH <sub>(P1B)</sub>	0x010	R/W	HALL模块计数器门限值寄存器	0x0
CNT <sub>(P1B)</sub>	0x014	R/W	HALL计数寄存器	0x0
LOCK	0x018	R/W	HALL写使能寄存器	0x0

注1: (P1B) 标注的寄存器为被保护的寄存器。

注2: LOCK==55H时, (P1B) 标注的寄存器允许写入; ==其他值, 禁止写入。

## 13.5 寄存器说明

### 13.5.1 HALL模块配置寄存器 (CFG)

位	符号	描述	复位值
31	-	保留	-
30	SW_IE	软件触发HALL信号变化中断使能 0: 禁止 1: 使能, CLR[F16]写 1, 将手动产生HALL 信号变化中断。	0
29	OV_IE	HALL 计数器溢出中断使能 0: 禁止 1: 使能	0
28	CHG_IE	HALL 信号变化中断使能 0: 禁止 1: 使能	0
27:25	-	保留	-
24	HALL_EN	HALL 模块使能开关 0: 禁止 1: 使能	0
23:21	-	保留	-
20	FIT_75	7/5 滤波使能(连续采样 7次, 5 次值一致) 0: 禁止 1: 使能	0
19:18	-	保留	-
17:16	CLK_DIV	HALL 时钟分频系数 00: 不分频 01: 2分频 10: 4分频 11: 8分频	0x0
15:0	FIT_LEN	滤波宽度, 低于对应脉冲宽度的信号将被硬件自动过滤掉, 滤波宽度的计算公式为[15:0]+1	0x0

### 13.5.2 HALL模块信息寄存器 (INFO)

位	符号	描述	复位值
31:29	-	保留	-
18:16	RAW_DATA	HALL[2:0]实时值	0x0
15:11	-	保留	-
10:8	FIT_DATA	HALL[2:0]滤波值	0x0
7:2	-	保留	-
1	OV_IF	HALL 计数器溢出事件标志 0: 没溢出 1: 溢出	0
0	CHG_IF	HALL事件变化标志 0: 无事件产生	0

		1: 有事件产生	
--	--	----------	--

### 13.5.3 HALL信息清零寄存器 (CLRf)

位	符号	描述	复位值
31:17	-	保留	-
16	SET_SWF	软件触发 HALL信号变化标志触发位 0: 无触发动作 1: 触发HALL产生事件, CHG_IF被置1, 该位自动清0	0
15:2	-	保留	-
1	CLR_OVF	HALL计数器溢出事件标志清零位 0: 无清除动作 1: 清除HALL计数器溢出标志OV_IF	0
0	CLR_CHGF	HALL信号变化事件标志清零位 0: 无清除动作 1: 清除HALL事件变化标志CHG_IF	0

### 13.5.4 HALL模块宽度计数值寄存器 (WIDTH)

位	符号	描述	复位值
31:0	CAP_CNT	HALL宽度计数器值	0x0

### 13.5.5 HALL模块计数器门限值寄存器 (TH)

位	符号	描述	复位值
31:0	TH	HALL计数器门限值	0x0

### 13.5.6 HALL计数寄存器 (CNT)

位	符号	描述	复位值
31:0	CNT	HALL计数值, 写入任意值可清零	0x0

### 13.5.7 HALL写使能寄存器 (LOCK)

位	符号	描述	复位值
31:0	LOCK	当LOCK=0x55, 使能操作HALL相关寄存器	0x0

## 第14章 增强型PWM(EPWM)

### 14.1 概述

增强型PWM模块支持8路PWM发生器，可以配置成相互独立的8路PWM输出（EPWM0-EPWM7），也可以配置成4对分别带有编程死区发生器的互补或同步的PWM（EPWM0-EPWM1，EPWM2-EPWM3，EPWM4-EPWM5，EPWM6-EPWM7）。

每一对PWM共用8位预分频器，有8组时钟分频器，提供5种分频系数（1、1/2、1/4、1/8、1/16）。每一路PWM输出有独立的16位计数器进行控制，另外16位的比较器用以调节占空比。8路PWM发生器提供36个中断标志，相关PWM通道的周期或占空比与计数器相符，将产生中断标志，每一路PWM有单独的使能位。

每路PWM可配置成单次模式（产生一个PWM信号周期）或者循环模式（连续输出PWM波形）。

### 14.2 特性

增强性PWM模块有如下特性：

- ◆ 8路独立的16位PWM控制模式。
  - 8路独立输出：EPWM0、EPWM1、EPWM2、EPWM3、EPWM4、EPWM5、EPWM6、EPWM7；
  - 4组互补PWM对：（EPWM0-EPWM1）、（EPWM2-EPWM3）、（EPWM4-EPWM5）、（EPWM6-EPWM7），可插入可编程死区时间；
  - 4组同步PWM对：（EPWM0-EPWM1）、（EPWM2-EPWM3）、（EPWM4-EPWM5）、（EPWM6-EPWM7），每组PWM对引脚同步。
- ◆ 支持组控制，EPWM0，EPWM2，EPWM4，EPWM6输出同步，EPWM1，EPWM3，EPWM5，EPWM7输出同步。
- ◆ 单次模式（仅支持边沿对齐）或者自动装载模式。
- ◆ 支持边沿对齐，中心对齐2种模式。
- ◆ 中心对齐模式支持对称计数和非对称计数。
- ◆ 互补的PWM中，支持可编程死区发生器。
- ◆ 每路PWM有独立的极性控制。
- ◆ 故障刹车保护以及恢复功能（软/硬件触发以及软/硬件恢复）。
- ◆ ACMP0/1输出和ADC比较器输出可触发硬件刹车保护。
- ◆ PWM边沿、周期点、零点、比较点0/1可触发启动AD转换。



## 14.3 功能描述

相关名称说明：

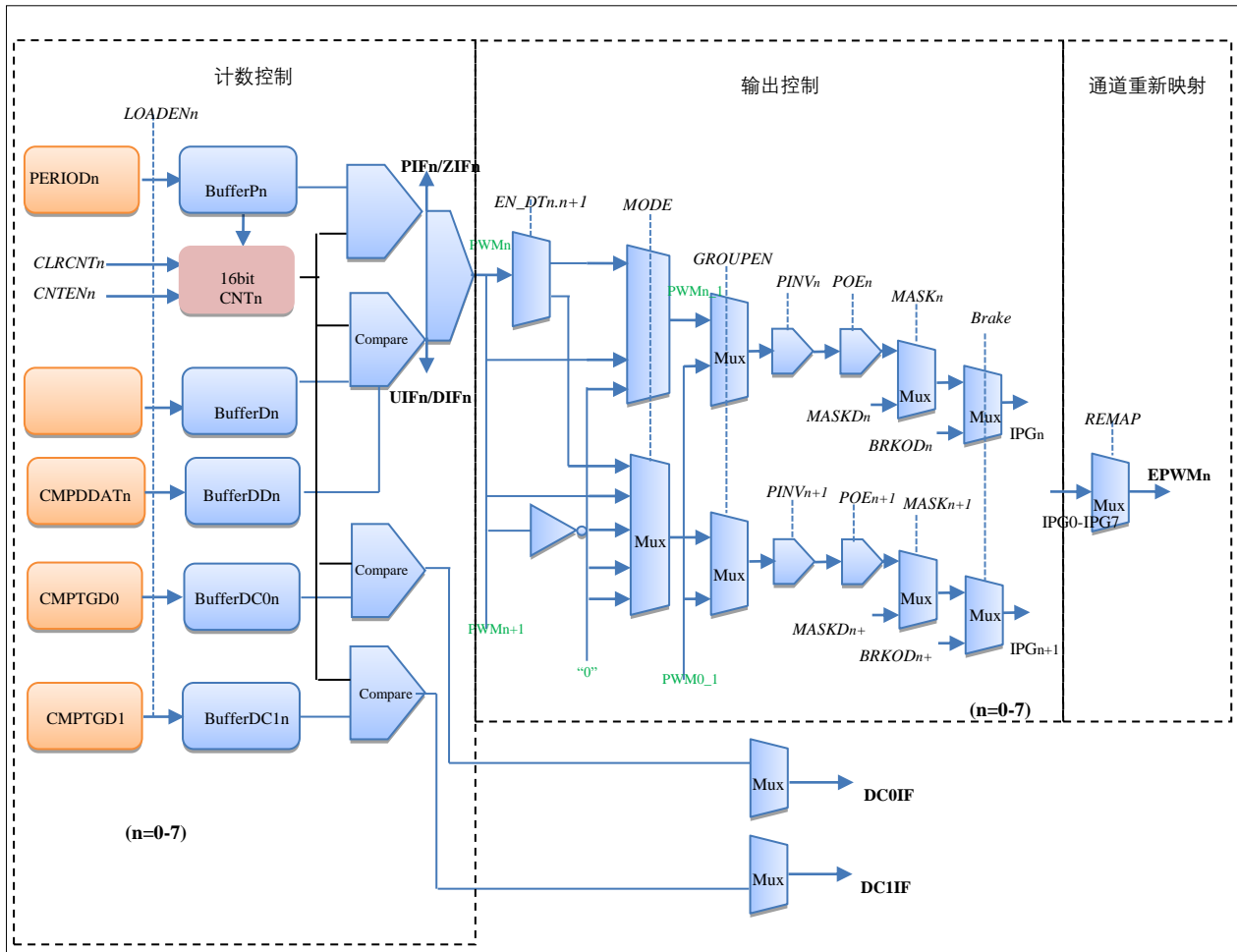
- 1) 周期点：计数器 CNTn 计数到与周期 PERIODn 相等时，称之为周期点。所产生的中断为 PIFn。
- 2) 零点：计数器 CNTn 计数到 0 时，称之为零点。所产生的中断为 ZIFn。
- 3) 向上比较点：计数器 CNTn 加计数到与 CMPDATn 相等时，称之为向上比较点。所产生的中断为 UIFn。边沿对齐计数方式无向上比较点。
- 4) 向下比较点：计数器 CNTn 减计数到与 CMPDATn 或 CMPDDATn 相等时，称之为向下比较点。所产生的中断为 DIFn。
- 5) 中点：中点即中间点，中点为中心对齐计数方式时 CNTn 计数到 PERIODn 相等的时刻，由于之后 CNTn 会减计数到 0，所以将该时刻称之为中点，也为周期点。边沿对齐计数方式无中点，但有周期点。

注意1：边沿对齐方式时，第一次计数开始时加载周期数据，将产生周期点；其他时刻由于计数器计数到0后，需要立即加载周期数据。所以之后的周期点与零点的位置是相同的。该对齐方式存在向下比较点，不存在向上比较点。

注意2：中心对齐时，第一次计数开始由0开始向上计数，将产生零点。在计数到周期数据时，将产生周期点(中点)。零点与中点交替存在。该对齐方式存在向上比较点与向下比较点，对称计数时，向上比较点与向下比较点均由CMPDATn决定；非对称计数时，向上比较点由CMPDATn决定，向下比较点由CMDDATn决定。

### 14.3.1 结构框图

图14-1：IPGn的信号为EPWMn重映射前的信号。



### 14.3.2 时钟分频

每一对PWM共用8位预分频器，在预分频之后，每一路PWM可以选择（1,1/2,1/4,1/8,1/16）5种分频比。  
 $PWM\_CLK = PCLK / (CLKPSC_{xx} + 1) / CLKDIV_n$ , 这里xx可以是01, 23, 45, 67, n=0-7。

### 14.3.3 独立输出模式

8路EPWM通道输出相互不影响，按照各自的周期/占空比数据运行。

### 14.3.4 互补输出模式

互补输出模式下，8路PWM分为4对，EPWM0与EPWM1成1对，EPWM2与EPWM3成1对，EPWM4与EPWM5成1对，EPWM6与EPWM7成1对。共有4对PWM。

EPWM0-EPWM1按EPWM0的周期/占空比数据运行，EPWM0与EPWM1波形反相。

EPWM2-EPWM3按EPWM2的周期/占空比数据运行，EPWM2与EPWM3波形反相。

EPWM4-EPWM5按EPWM4的周期/占空比数据运行，EPWM4与EPWM5波形反相。

EPWM6-EPWM7按EPWM6的周期/占空比数据运行，EPWM6与EPWM7波形反相。

在此模式下，EPWM1/EPWM3/EPWM5/EPWM7输出与自己的相关运行数据寄存器无关，但输出控制仍然有效。如输出使能，掩码，刹车等控制。

互补模式下支持死区延时控制。

### 14.3.5 同步输出模式

同步输出模式下，8路PWM分为4对，EPWM0与EPWM1成1对，EPWM2与EPWM3成1对，EPWM4与EPWM5成1对，EPWM6与EPWM7成1对。共有4对PWM。

EPWM0-EPWM1按EPWM0的周期/占空比数据运行，EPWM0与EPWM1波形同相。

EPWM2-EPWM3按EPWM2的周期/占空比数据运行，EPWM2与EPWM3波形同相。

EPWM4-EPWM5按EPWM4的周期/占空比数据运行，EPWM4与EPWM5波形同相。

EPWM6-EPWM7按EPWM6的周期/占空比数据运行，EPWM6与EPWM7波形同相。

在此模式下，EPWM1/EPWM3/EPWM5/EPWM7输出与自己的相关运行数据寄存器无关，但输出控制仍然有效。如输出使能，掩码，刹车等控制。

### 14.3.6 成组输出模式

GROUPEN=1使能成组功能，8路PWM分为2组，EPWM0，EPWM2，EPWM4，EPWM6为1组，EPWM1，EPWM3，EPWM5，EPWM7为1组。

EPWM0-EPWM2-EPWM4-EPWM6按EPWM0的周期/占空比数据运行，4个通道波形同相。

EPWM1-EPWM3-EPWM5-EPWM7按EPWM1的周期/占空比数据运行，4个通道波形同相。

成组功能打开时，EPWM2/EPWM4/EPWM6/EPWM3/EPWM5/EPWM7输出与自己的相关运行数据寄存器无关，但输出控制仍然有效。如输出使能，掩码，刹车等控制。

### 14.3.7 加载更新模式

计数器加载模式有两种：单次模式(One-shot)与连续模式(自动加载模式)

#### 单次模式：

周期占空比相关数据在计数器开始加载一次。

#### 连续模式：

周期占空比数据在PWM周期内零点与中点自动加载。中点加载只存在中心对齐计数模式下。

边沿对齐计数模式下，产生零点的同时也会产生周期点，此时计数比较电路会重新加载CMPDATn/PERIODn/CMPTGD0/CMPTGD1的值。

中心对齐计数模式下，中点和零点均会自动加载相关寄存器的值。这样的结构支持前半波形周期占空比与后半波形周期占空比设置不一致，当周期占空比相关寄存器的值未改变时，则保持一致。

由于EPWM存在双缓存结构，在EPWM运行的过程中，改变相关运行寄存器CMPDATn/CMPDDATn/PERIODn/CMPTGD0/CMPTGD1的值，PWM输出波形不会立即改变，只有在零点或周期点时这些寄存器的值才会加载到相应的缓存中。

这样的结构在改变周期占空比数据后，不会立即改变当前PWM周期或半周期内的输出波形，在下个周期或半周期内PWM波形才会做出相应的变化。即任何PWM相关数据的改变不会影响当前一个完整PWM周期或半周期。

在高速的应用中，有可能会出现加载点已经到来，但写入运行寄存器的操作还未完成。此时不希望出现部分运行数据已经加载，另外一部分运行数据没有加载的情况。

针对该高速应用情况。该EPWM模块提供了加载使能位，改变相关运行寄存器后，需要将加载使能位LOADENn置1，加载完毕后LOADENn位自动清零。另外可以读取该位来判断是否将相关寄存器的值加载到实际电路中。如果LOADENn=0，则表示已经加载，将影响正在输出的PWM波形；如果LOADENn=1，则表示还未加载，当前的PWM波形还未发生变化，将在下一个加载点才会加载之前改变的寄存器的值。如果再次改变相关运行寄存器的值，也需重新将LOADENn置1。

默认条件下，PWM在零点与周期点均会加载相关寄存器的运行数据，以及产生零点与周期点中断。为了适应更灵活应用需求。PWM支持不同方式的加载方式与零点/周期点中断产生方式。

在寄存器EPWMCON3中LOADTYPn(0-7)可设置加载方式与零点/周期点的中断方式：

LOADTYEn	中心对齐加载	边沿对齐加载
00	每个零点与周期点均加载与产生零点与周期点中断标志	每个零点或周期点均加载与产生零点与周期点中断标志
01	每个零点加载与产生零点中断标志	每2个零点加载与产生零点中断标志
10	第一个零点与下一个周期点交替加载与产生零点与周期点中断标志	每3个零点或周期点加载与产生零点与周期点中断标志
11	每两个零点加载与产生相关零点中断标志	每4零点加载与产生零点中断标志

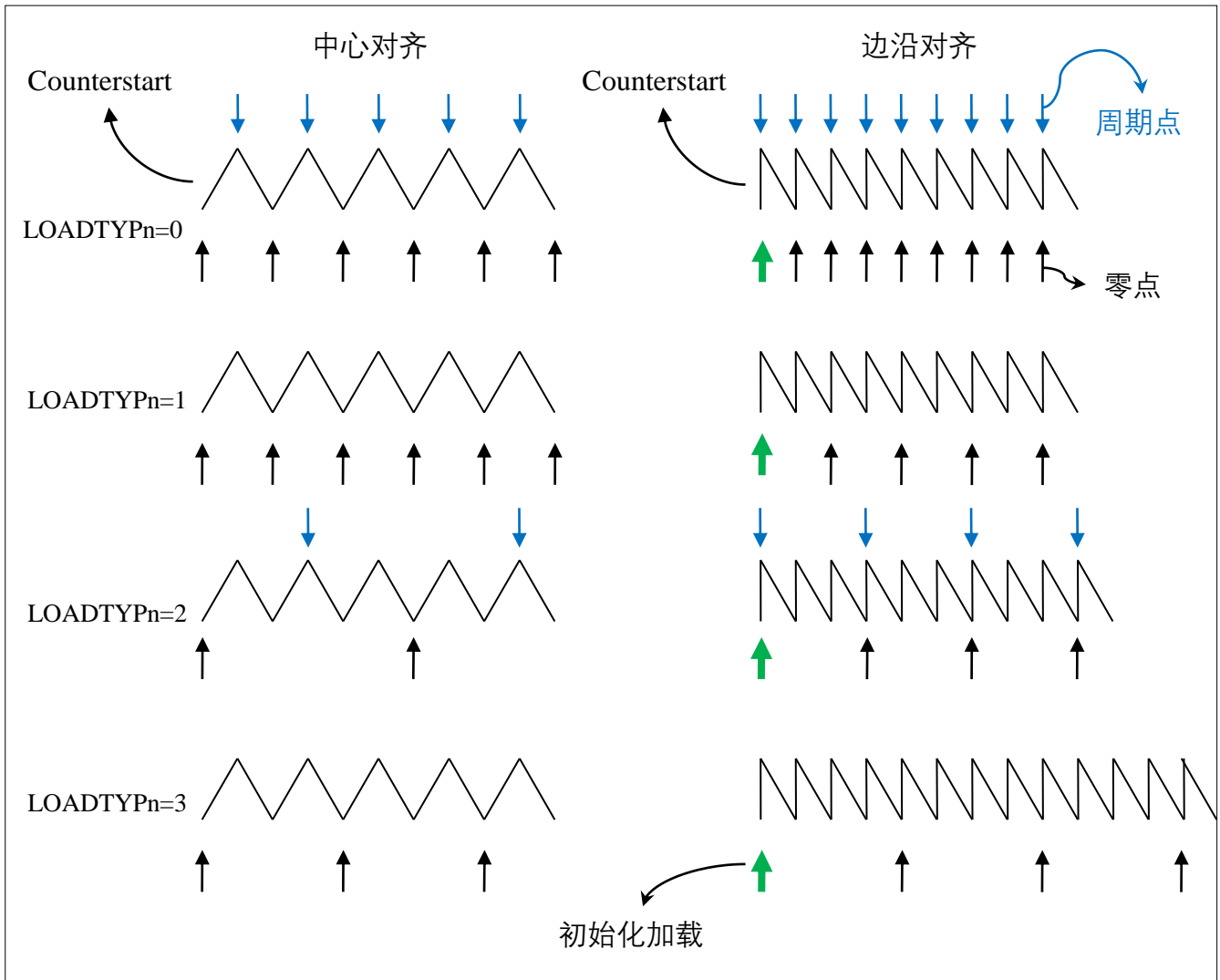
LOADTYPn=0时，边沿对齐为每1个周期加载一次，中心对齐为每0.5个周期加载一次。

LOADTYPn=1时，边沿对齐为每2个周期加载一次，中心对齐为每1个周期加载一次。

LOADTYPn=2时，边沿对齐为每3个周期加载一次，中心对齐为每1.5个周期加载一次。

LOADTYPn=3时，边沿对齐为每4个周期加载一次，中心对齐为每2个周期加载一次。

图14-2: PWM周期/占空比加载更新框图



### 14.3.8 边沿对齐计数模式

边沿对齐模式下，计数方式为向下计数，即减1计数。16位PWM计数器CNTn在每个周期开始向下计数，与锁存CMPDATn值进行比较，当CNTn=CMPDATn时EPWMn输出高电平，CMPnDIF置1。CNTn继续向下计数至0，此时EPWMn将输出低电平，当前CMPDATn和PERIODn在PWMnCNTM=1的情况会重新加载，PIF周期中断标志置位。

边沿对齐相关参数如下：

$$\text{高电平时间} = (\text{CMPDATn} + 1) \times T_{\text{pwm}}$$

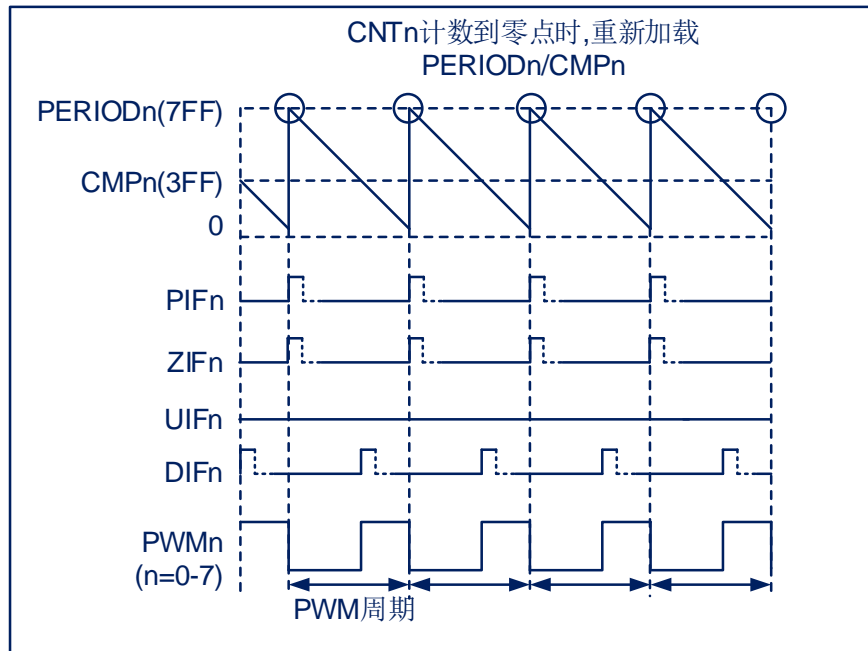
$$\text{周期} = (\text{PERIODn} + 1) \times T_{\text{pwm}}$$

$$\text{占空比} = \frac{\text{CMPDATn} + 1}{\text{PERIODn} + 1}$$

若CMPDATn>PERIODn，占空比为100%，EPWMn通道一直为高。且不会产生向下比较中断。

若CMPDATn=0，则占空比为0%。

图14-3：边沿对齐模式波形图



### 14.3.9 中心对齐计数模式

中心对齐模式下，计数方式为先向上计数再向下计数。

中心对齐模式按对称方式又分为两种：对称计数方式与非对称计数方式。

对称计数方式(ASYMEN=0)占空比由CMPDATn决定。

非对称计数方式(ASYMEN=1)占空比由CMPDATn与CMPDDATn共同决定。

中心对齐对称计数方式下，16位PWM计数器CNTn从0开始向上计数，当CNTn=CMPDATn时，EPWMn输出高电平，之后CNTn继续向上计数至与PERIODn相等，然后CNTn开始向下计数，在向下计数的过程中CNTn=CMPDATn时，EPWMn输出低电平，之后继续向下计数至0。

$$\text{高电平时间} = (\text{PERIODn} \times 2 - \text{CMPDATn} \times 2 - 1) \times T_{pwm}$$

$$\text{周期} = \text{PERIODn} \times 2 \times T_{pwm}$$

$$\text{占空比} = \frac{\text{PERIODn} \times 2 - \text{CMPDATn} \times 2 - 1}{\text{PERIODn} \times 2}$$

若CMPDATn>=PERIODn，占空比为0%，EPWMn通道一直为低，且不会产生向上比较中断与向下比较中断。

若PERIODn=0，占空比为0%，EPWMn通道一直为低，且CNTn使能时零点中断与周期点中断一直存在。

若CMPDATn=0，则占空比为100%。

图14-4：中心对齐模式对称计数波形图

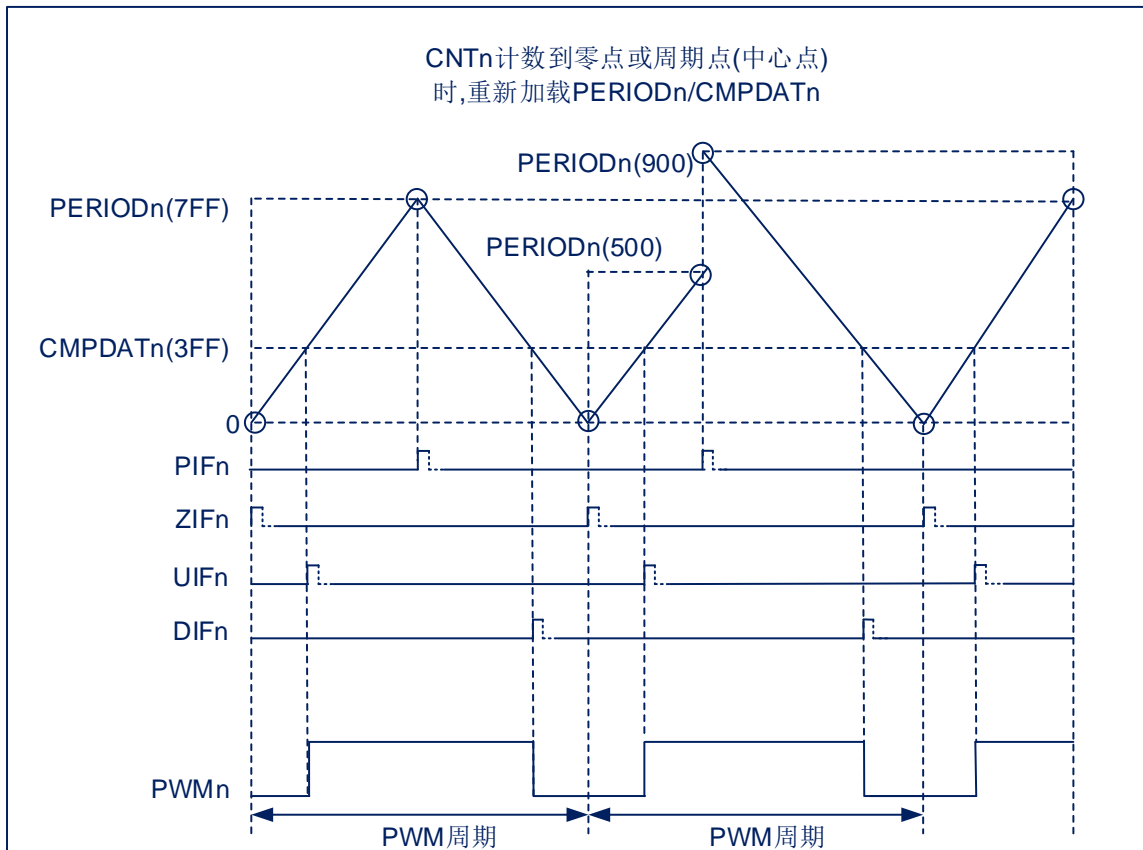
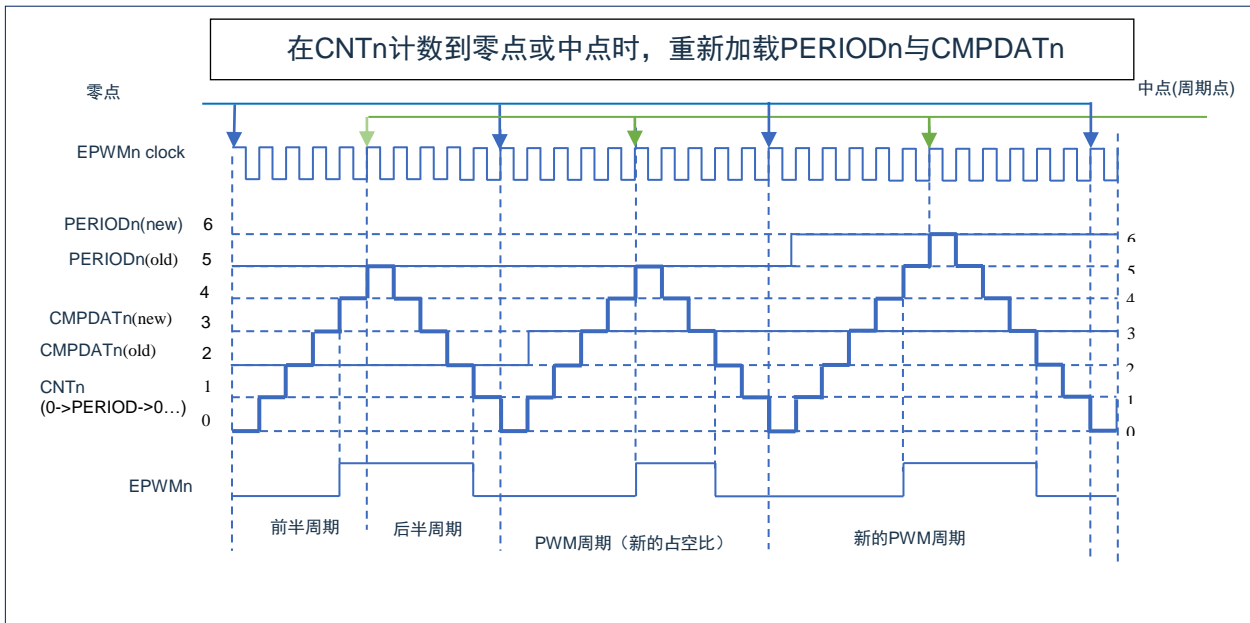


图14-5: 中心对齐计数器波形 (对称计数)



中心对齐非对称计数方式下, 16位PWM计数器CNTn从0开始向上计数, 当CNTn=CMPDATn时, EPWMn输出高电平, 之后CNTn继续向上计数至与PERIODn相等, 然后CNTn开始向下计数, 在向下计数的过程中CNTn=CMPDDATn时, EPWMn输出低电平, 之后继续向下计数至0。开启非对称计数方式需要将ASYMEN置1, 非对称计数方式下可实现精确的中心对齐波形。

中心对齐非对称计数方式相关参数如下:

$$\text{高电平时间} = (\text{PERIODn} \times 2 - \text{CMPDDATn} - \text{CMPDATn} - 1) \times T_{pwm}$$

$$\text{周期} = \text{PERIODn} \times 2 \times T_{pwm}$$

$$\text{占空比} = \frac{\text{PERIODn} \times 2 - \text{CMPDDATn} - \text{CMPDATn} - 1}{\text{PERIODn} \times 2}, \quad (\text{CMPDATn} < \text{PERIODn}, \text{CMPDDATn} < \text{PERIODn})$$

$$\text{占空比} = \frac{\text{PERIODn} - \text{CMPDDATn} - 1}{\text{PERIODn} \times 2}, \quad (\text{CMPDATn} \geq \text{PERIODn}, \text{CMPDDATn} < \text{PERIODn})$$

$$\text{占空比} = \frac{\text{PERIODn} - \text{CMPDATn}}{\text{PERIODn} \times 2}, \quad (\text{CMPDATn} < \text{PERIODn}, \text{CMPDDATn} \geq \text{PERIODn})$$

$$\text{占空比} = 0\%, \quad (\text{CMPDATn} \geq \text{PERIODn}, \text{CMPDDATn} \geq \text{PERIODn})$$

CMPDATn ≥ PERIODn时不会产生向上比较中断。

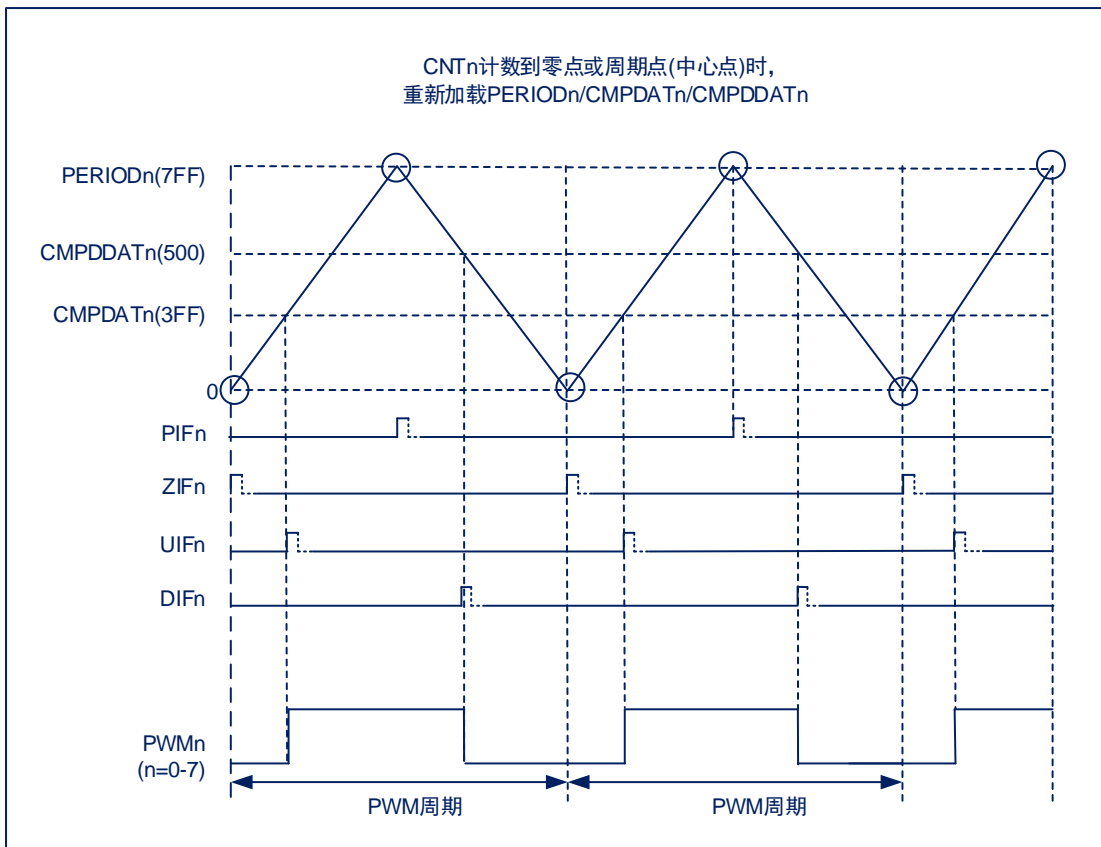
CMPDDATn ≥ PERIODn时不会产生向下比较中断。

若PERIODn=0, 占空比为0%, EPWMn通道一直为低, 且CNTn使能时零点中断与周期点中断一直存在。

若CMPDATn=0与CMPDDATn=0, 则占空比为100%。



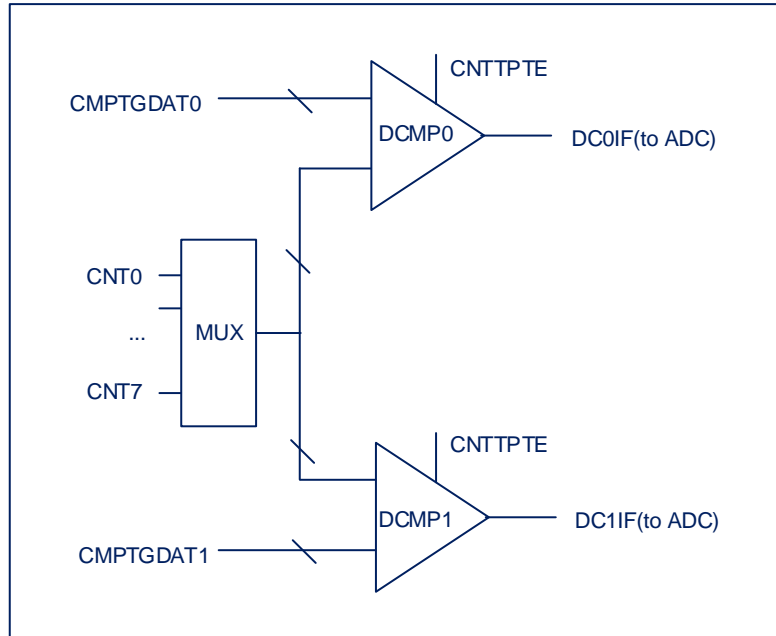
图14-6: 中心对齐模式非对称计数波形图



### 14.3.10 独立计数器比较功能

在PWMn通道计数器(CNTn)计数器期间，提供了两个数字比较器，计数器CNTn与预设的值进行比较，若计数器的值与预设值相等则可产生中断信号或者触发ADC启动。该功能不影响PWM的输出。

图14-7：独立计数器比较功能

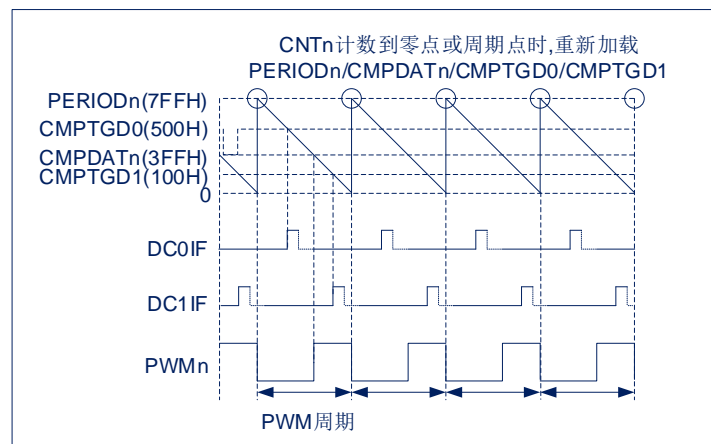


数字比较器0比较计数器CNTn的值与CMPTGDAT0的值。若相等则产生中断标志位DC0IF.CMPTGD0[18:16]选择PWM0-7通道计数器之一与CMPTGDAT0进行比较。

数字比较器1比较计数器CNTn的值与CMPTGDAT1的值。若相等则产生中断标志位DC1IF.CMPTGD1[18:16]选择PWM0-7通道计数器之一与CMPTGDAT1进行比较

1) 边沿对齐模式，数字比较器的工作方式：

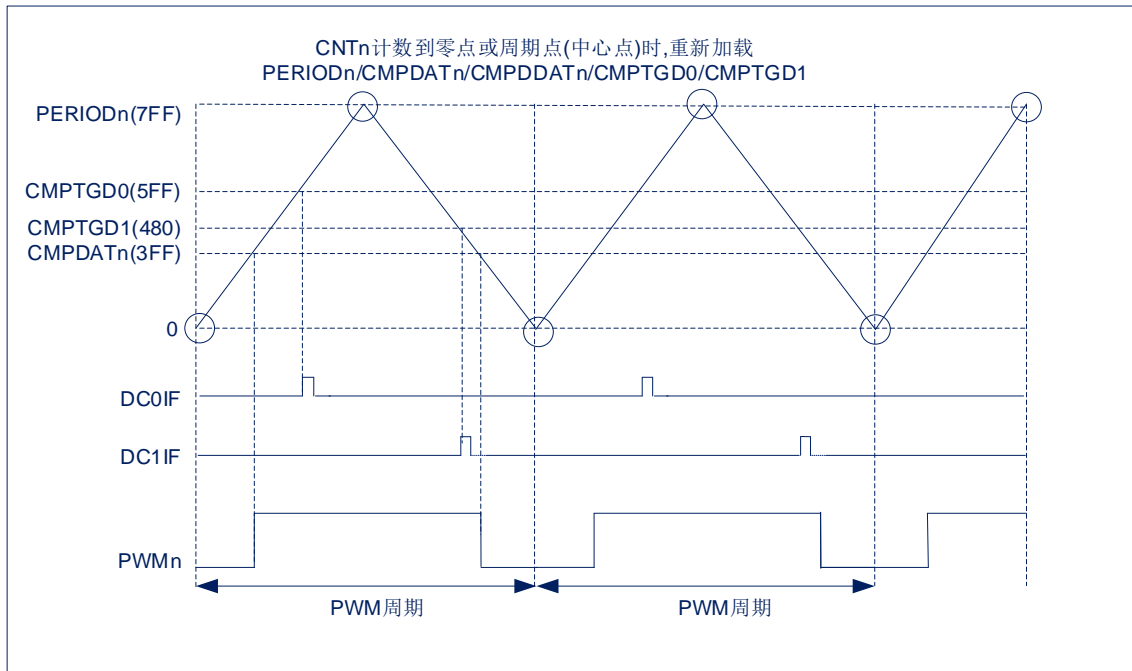
图14-8：边沿对齐模式，数字比较器的工作方式



2) 在边沿计数模式下，数字比较器0/1可设置为在任意计数时刻产生比较中断。

3) 中心对齐模式，数字比较器的工作方式：

图14-9: 中心对齐模式, 数字比较器的工作方式



4) 在中心对齐计数模式下, 数字比较器 0/1 可各自设置为在向上或向下计数模式比较触发。即都可在前半周期触发或后半周期触发, 也可一个在前半周期触发, 另一个在后半周期触发。由 CMPTGD0[19]位 CMPTGDSn 决定。

### 14.3.11 可编程死区发生器

8通道PWM可设置为4组互补对。在互补输出模式下，PWM1，PWM3，PWM5，PWM7的周期与占空比分别由PWM0，PWM2，PWM4，PWM6相关寄存器决定，同时死区延时寄存器也可影响PWM互补对的占空比。此时除了对应的输出使能控制位（PWMnOE）、通道重映射控制、掩码控制、刹车控制、成组控制和极性控制PWM1/PWM3/PWM5/PWM7输出波形不再受自己的寄存器控制。

在互补模式下，每组互补PWM对均支持插入死区延时，插入的死区时间如下：

PWM0/1死区时间：左死区（PWM01LDT[15:0]+1）\*TPWM0；右死区（PWM01RDT[15:0]+1）\*TPWM0

PWM2/3死区时间：左死区（PWM23LDT[15:0]+1）\*TPWM2；右死区（PWM23RDT[15:0]+1）\*TPWM2

PWM4/5死区时间：左死区（PWM45LDT[15:0]+1）\*TPWM4；右死区（PWM45RDT[15:0]+1）\*TPWM4

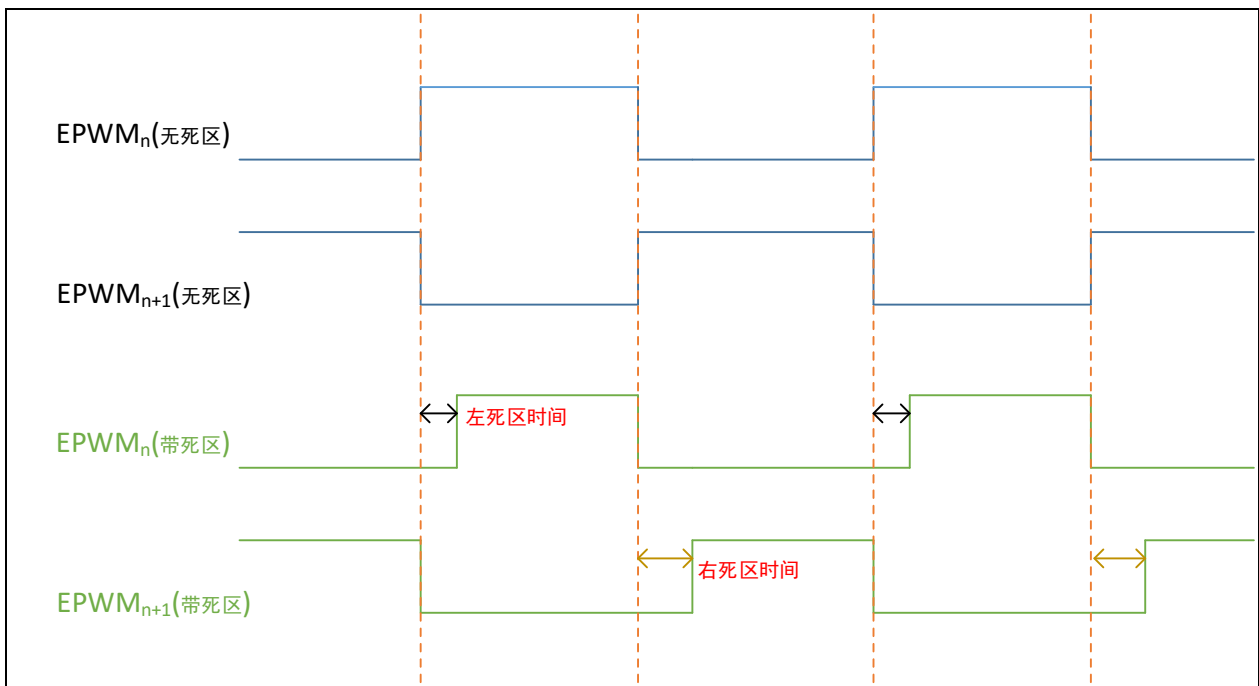
PWM6/7死区时间：左死区（PWM67LDT[15:0]+1）\*TPWM6；右死区（PWM67RDT[15:0]+1）\*TPWM6

TPWM0/TPWM2/TPWM4/TPWM6分别为PWM0/PWM2/PWM4/PWM6的时钟源周期。

死区时间可设置的范围：0.014us~904.40us（Fpwmn=72MHz）。

输出模式不影响计数器的模式，所以中心对齐与边沿对齐均支持互补输出模式。

图14-10：中心对齐与边沿对齐均支持互补输出模式



### 14.3.12掩码及掩码预设功能

EPWM支持掩码功能。EPWM0-EPWM7每个通道有单独的控制，EPWMn对应的控制位为MASKENn，MASKDn（在寄存器MASK中）。

当MASKENn=0时，EPWMn通道输出正常的PWM波形；

当MASKENn=1时，EPWMn通道输出MASKDn的数据；

掩码功能的控制寄存器MASK也支持自动加载预设值的功能。开启该功能需要将输出控制寄存器POEN的MASKLE位置1，允许MASK自动加载MASKNXT寄存器的值，同时禁止写MASK寄存器。

加载时刻在POEN中MASKLS<2:0>设置，可选择与EPWM0-EPWM7其中之一的周期/占空比加载时刻（加载点）相同。

### 14.3.13霍尔传感器接口功能

EPWM考虑与霍尔传感器的接口。内部包含一个HALL位置状态检测电路，该电路检测CCP0/1模块中内部捕获通道CAP0，CAP1，CAP2经过滤波后的电平。

检测电路内部处理后存在状态称之为HALLST：

HALLST存在8种状态，与HALL位置状态对应关系如下：

HALLST	对应的状态
000	HALL检测电路未启动或初始状态
001	{CAP2-CAP0}=001
010	{CAP2-CAP0}=010
011	{CAP2-CAP0}=011
100	{CAP2-CAP0}=100
101	{CAP2-CAP0}=101
110	{CAP2-CAP0}=110
111	{CAP2-CAP0}变化过程中出现错误的状态或者出现了错误的序列

HALLST的值可从MASKNXT寄存器中读出，在任何时间可判断HALL位置或序列状态，

HALL状态检测序列支持以下两种({CAP2，CAP1，CAP0}出现的顺序)：

- ◆ .....-6-2-3-1-5-4-6-.....
- ◆ .....-6-4-5-1-3-2-6-.....

如出现其他序列则认为出现了错误，HALLST将进入111的状态后停止检测。同时将产生中断标志HALLIF。如果需要重新启动HALL检测电路，需要将MASKNXT寄存器中的HALLCLR位写1，HALLST即从111的状态进入000的初始状态重新启动检测电路。

HALL检测电路提供了可与掩码自动加载相关功能。该功能不需要软件介入便可以控制EPWM的输出通道波形。

HALLST每个有效状态对应一个掩码预设缓存，一共有7个掩码预设缓存：

HALLST(HALLEN=1)	对应的掩码预设缓存
000	掩码预设缓存7
001	掩码预设缓存1
010	掩码预设缓存2
011	掩码预设缓存3
100	掩码预设缓存4
101	掩码预设缓存5
110	掩码预设缓存6
111	掩码预设缓存7
HALLEN=0	掩码预设缓存0

如果开启掩码自动加载预设值的功能，则在相应的状态下，且在选择的加载点时刻，对应的掩码预设缓存中的数据将加载到MASK寄存器中。例如：

HALLST中位置状态从000改变到001时，在进入001状态第一个加载点，掩码预设缓存1的数据则加载到MASK寄存器中。

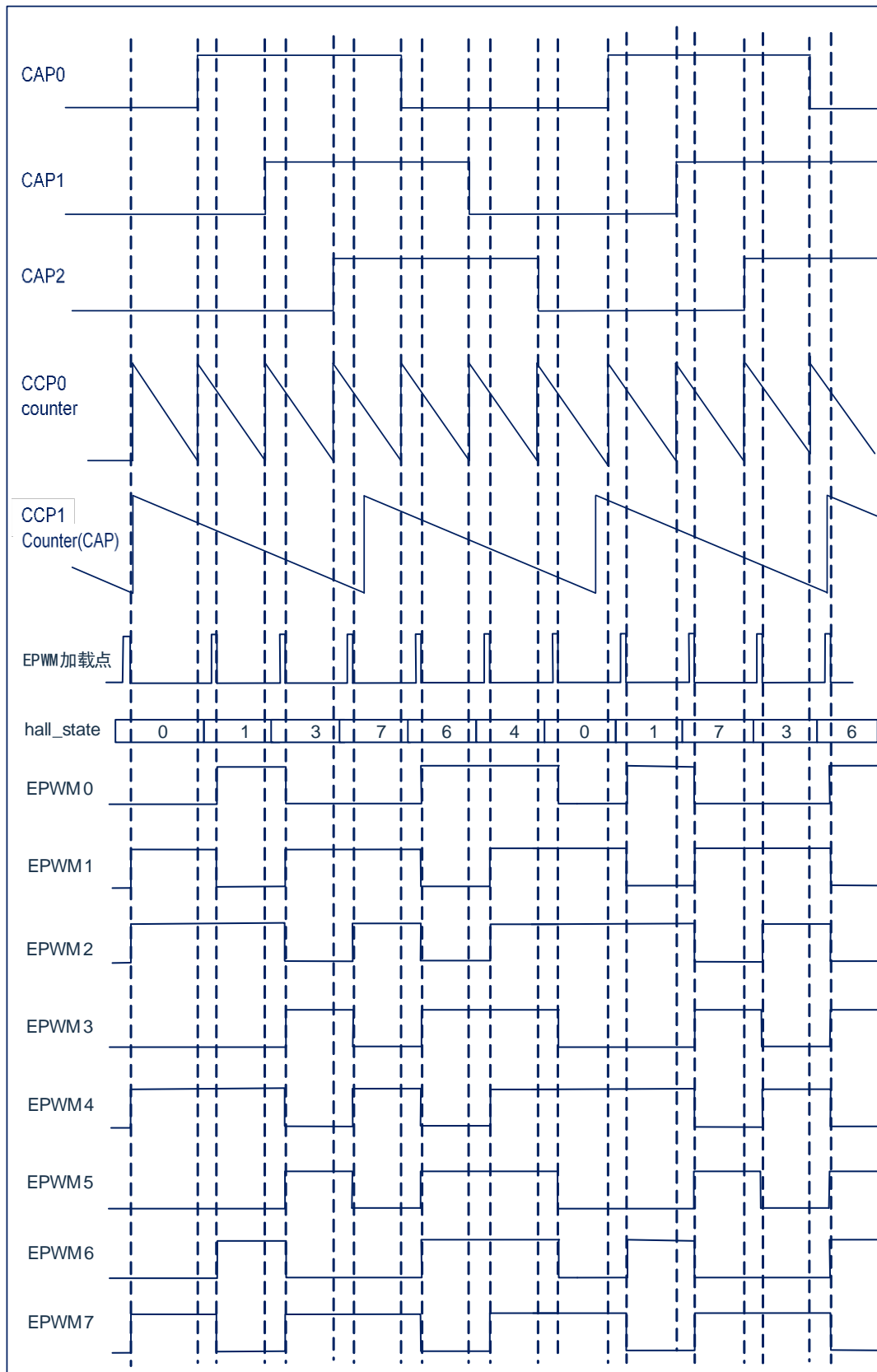
之后HALLST中位置状态从001改变到101时，在进入101状态第一个加载点，掩码预设缓存5的数据则加载到MASK寄存器中。

如果出现错误的序列，如CAP2-CAP0输入从101改变到010时，这不是正确的序列，HALLST中位置状态则从101改变到111，并将中断标志位HALLIF置1。在进入111状态第一个加载点，掩码预设缓存7的数据则加载到MASK寄存器中。

初始状态下，掩码预设缓存7的数据在加载点加载到MASK寄存器中。

下图为HALL检测与掩码预设功能的时序实例（仅作示例，不代表实际的波形），其中HALL检测功能使能，EPWM掩码控制预设数据加载使能，掩码预设缓存1设置为0xFF55，掩码预设缓存3设置为0xFFAA，掩码预设缓存4设置为0xFFFF，掩码预设缓存6设置为0xFF96，掩码预设缓存7设置为0xFF96。

图14-11: HALL检测及掩码预设时序示例



### 14.3.14 故障保护功能（刹车及恢复功能）

EPWM支持故障保护功能，BRKODn控制8个通道的刹车输出电平。故障保护功能由BRKCTL寄存器控制。EPWM故障保护触发来源有：

电平触发源：

- 1) 外部BKIN电平信号（高电平或低电平）
- 2) 软件刹车信号（SWBRK 位置为 1）
- 3) 模拟比较器 0 的输出（输出高或输出低）
- 4) 模拟比较器 1 的输出（输出高或输出低）

脉冲触发源：

- 1) 外部 BKIN 的边沿信号（上升沿或下降沿）
- 2) 模拟比较器 0 的输出事件（上升沿或下降沿或双沿）
- 3) 模拟比较器 1 的输出事件（上升沿或下降沿或双沿）
- 4) ADC 结果比较器 0 事件（结果比较事件）

故障中断标志位BRKIF（软件清0）：

检测到有效的刹车触发源信号后，故障中断标志BRKIF置1，需由软件清零。

故障信号标志位BRKAF（只读）：

故障信号标志位BRKAF置1，刹车信号撤销后，BRKAF自动清零0。BRKAF为只读位。

故障保护输出状态标志位BRKOSF（只读）：

BRKOSF=1，表示EPWMn通道输出BRKODn数据状态；

BRKOSF=0，表示EPWMn为正常输出状态。

指示EPWM输出在刹车状态还是正常状态。检测到有效的刹车信号时BRKOSF将置1。软件恢复模式下，执行刹车状态清除操作（BRKCLR=1）会影响到该位的状态。

故障保护模式可分为4种，以适应不同故障保护场合的需要。

BRKMS	故障保护模式
00	停止模式（软件恢复）
01	暂停模式（软件恢复）
10	恢复模式（硬件恢复）
11	延时恢复模式（硬件恢复）

注：故障中断标志（BRKIF）与恢复功能无关，仅代表产生了刹车信号。故障中断标志也支持累加功能。

#### 停止模式：

产生故障保护及故障中断标志，将CNTENn位清零，停止计数器运行。恢复输出需要刹车信号撤销，且执行故障状态清除操作（BRKCLR=1），然后重新将CNTENn置1。

#### 暂停模式：

产生故障保护及故障中断标志，但计数器继续运行。恢复输出需要刹车信号撤销，执行故障状态清除操作（BRKCLR=1）后，在最近一次的加载更新点恢复正常输出。

#### 恢复模式：

产生故障保护及故障中断标志，但计数器继续运行。刹车信号撤销后，自动在最近一次的加载更新点恢复正常输出。不需要执行故障状态清除操作。

需要注意区分刹车信号是脉冲信号还是电平信号：如果刹车源为电平信号，则需要等待刹车撤销后才能恢复输出；如果为脉冲信号，则EPWM输出在出发刹车之后最近一次加载更新点恢复输出，除非期间再次产



生刹车脉冲信号。

**延时恢复模式：**

产生故障保护及故障中断标志，但计数器继续运行。刹车信号撤销后延时一段时间EPWM在最近一次的加载更新点恢复正常输出。不需要执行故障状态清除操作。

延时时间可以自由设置，BRKRDT的低16位RDT控制延时时间。延时时间如下：

$$T_{delay} = (RDT+1) * T_{PCLK}$$

需要注意区分刹车信号是脉冲信号还是电平信号：如果刹车源为电平信号，则需要等待刹车撤销后才能恢复输出；如果为脉冲信号，则EPWM输出等待完成延时后最近一次加载更新点恢复输出，除非期间再次产生刹车脉冲信号。

产生刹车保护后，EPWMn通道输出BRKODn中数据，每个通道可单独设置输出高/低电平。

### 14.3.15 调试模式下的输出状态

在调试模式下，CPU 的状态有运行状态与暂停状态。一种为正常运行状态；一种状态为执行 STOP 命令/运行到断点/单步之后的暂停状态。

其中暂停状态下 EPWMn 工作时 (POEn=1) 输出状态可以通过寄存器 CON 中的 HALTMS 位来配置。

HALTMS=0 时，EPWMn 的输出状态在暂停时为正常输出。

HALTMS=1 时，EPWMn 的输出状态在暂停时输出刹车数据，但此时不会产生故障相关的标志位。EPWMn 的计数器将继续运行，在恢复运行状态后最近的加载更新点恢复 EPWMn 输出。

需要注意的是，在调试模式下暂停时，EPWMn 的相关运行数据寄存器的值不会自动改变，还会保持之前的状态。

### 14.3.16 输出通道重映射功能

输出通道重映射功能可满足应用中更加灵活的排版需求。芯片管脚分布图中 EPWM0-EPWM7 的管脚默认为对应的 PWM 通道输出。也可通过输出通道重映射功能来重新配置所需要的通道。

EPWM0-EPWM7 默认对应的内部通道分别为 IPG0-IPG7，通过 EPWM 输出通道重映射寄存器 POREMAP 可将 IPG0-IPG7 任意一路通道重新分配到 EPWMn(n=0-7)。输出通道重映射功能仅对端口输出通道重新分配，其内部控制及中断不会重映射。

### 14.3.17 EPWM配置过程

- ◆ 在 LOCK 寄存器中写入 0x55 使能 EPWM 寄存器操作
- ◆ 配置 EPWM 时钟分频，设置预分频比和独立分频比
- ◆ 选择模式，独立模式或互补模式
- ◆ 设置 EPWM 周期和占空比
- ◆ 设置 EPWM 输出极性
- ◆ 使能 EPWM 计数器
- ◆ 配置相关 IO 口为 EPWM 功能口
- ◆ 使能相关 EPWM 通道输出
- ◆ 在 LOCK 寄存器写入 0x00，避免 EPWM 相关寄存器被误操作，直至下一次需要操作 EPWM 相关寄存器时才重新使能

### 14.3.18 中断

EPWM单元有八个中断源：

- ZIFn—EPWM 计数器计数为零时产生的中断标志
- UIFn—EPWM 计数器向上计数到 CMPDATn 中断标志
- PIFn—EPWM 计数器边沿对齐计数周期点中断标志，中心对齐计数中点中断标志
- DIFn—EPWM 计数器向下计数到 CMPDATn/CMPDDATn 中断标志
- DC0IF—EPWM 计数器计数到与 CMPTGD0 相等的中断标志
- DC1IF—EPWM 计数器计数到与 CMPTGD1 相等的中断标志
- HALLIF—霍尔状态错误中断标志位
- BRKIF—故障中断标志位

所有的中断标志都由硬件置位，且必须通过软件清零。

## 14.4 寄存器映射

(EPWM 基地址 = 0x4006\_4200) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CLKPSC(P1B)	0x000	R/W	EPWM预分频寄存器	0x00000000
CLKDIV(P1B)	0x004	R/W	EPWM时钟选择寄存器	0x00000000
CON(P1B)	0x008	R/W	EPWM控制寄存器	0x00000000
CON2(P1B)	0x00C	R/W	EPWM控制寄存器2	0x00000000
CON3(P1B)	0x010	R/W	EPWM控制寄存器3	0x00000000
CON4(P1B)	0x014	R/W	EPWM控制寄存器4	0x00000000
PERIOD0(P1A)	0x018	R/W	EPWM周期寄存器0	0x00000000
PERIOD1(P1A)	0x01C	R/W	EPWM周期寄存器1	0x00000000
PERIOD2(P1A)	0x020	R/W	EPWM周期寄存器2	0x00000000
PERIOD3(P1A)	0x024	R/W	EPWM周期寄存器3	0x00000000
PERIOD4(P1A)	0x028	R/W	EPWM周期寄存器4	0x00000000
PERIOD5(P1A)	0x02C	R/W	EPWM周期寄存器5	0x00000000
PERIOD6(P1A)	0x030	R/W	EPWM周期寄存器6	0x00000000
PERIOD7(P1A)	0x034	R/W	EPWM周期寄存器7	0x00000000
CMPDAT0(P1A)	0x038	R/W	EPWM比较寄存器0	0x00000000
CMPDAT1(P1A)	0x03C	R/W	EPWM比较寄存器1	0x00000000
CMPDAT2(P1A)	0x040	R/W	EPWM比较寄存器2	0x00000000
CMPDAT3(P1A)	0x044	R/W	EPWM比较寄存器3	0x00000000
CMPDAT4(P1A)	0x048	R/W	EPWM比较寄存器4	0x00000000
CMPDAT5(P1A)	0x04C	R/W	EPWM比较寄存器5	0x00000000
CMPDAT6(P1A)	0x050	R/W	EPWM比较寄存器6	0x00000000
CMPDAT7(P1A)	0x054	R/W	EPWM比较寄存器7	0x00000000
POREMAP_EN(P1B)	0x058	R/W	EPWM输出通道重映射使能寄存器	0x0000
POREMAP(P1B)	0x05C	R/W	EPWM输出通道重映射寄存器	0x76543210
POEN(P1B)	0x060	R/W	EPWM输出控制寄存器	0x00000000
BRKCTL(P1B)	0x064	R/W	EPWM故障保护控制寄存器	0x00000000
MASK(P1B)	0x068	R/W	EPWM输出掩码寄存器	0x00000000
MASKNXT(P1B)	0x06C	R/W	EPWM输出掩码预设寄存器	0x00000000
CMPTGD0(P1B)	0x070	R/W	EPWM计数器比较寄存器0	0x00000000
CMPTGD1(P1B)	0x074	R/W	EPWM计数器比较寄存器1	0x00000000
IMSC (P1B)	0x078	R/W	EPWM中断使能寄存器	0x00000000
IMSC2(P1B)	0x07C	R/W	EPWM中断使能寄存器2	0x00000000
RIS	0x080	RO	EPWM中断源状态寄存器	0x00000000
RIS2	0x084	RO	EPWM中断源状态寄存器2	0x00000000
MIS	0x088	RO	EPWM已使能中断状态寄存器	0x00000000
MIS2	0x08C	RO	EPWM已使能中断状态寄存器2	0x00000000
ICLR	0x090	WO	EPWM中断清零寄存器	0x00000000
ICLR2	0x094	WO	EPWM中断清零寄存器2	0x00000000
IFA(P1B)	0x098	R/W	EPWM中断累加控制寄存器	0x00000000
LOCK	0x09C	R/W	EPWM写使能控制寄存器	0x00000000
BRKRDT(P1B)	0x0A0	R/W	EPWM故障保护恢复延时寄存器	0x00000000
DTCTL01(P1B)	0x0A4	R/W	EPWM01死区长度寄存器	0x00000000
DTCTL23(P1B)	0x0A8	R/W	EPWM23死区长度寄存器	0x00000000
DTCTL45(P1B)	0x0AC	R/W	EPWM45死区长度寄存器	0x00000000

DTCTL67(P1B)	0x0B0	R/W	EPWM67死区长度寄存器	0x00000000
--------------	-------	-----	---------------	------------

注1: (P1A/P1B)标注的寄存器为被保护的寄存器。

注2: (P1A): LOCK==55H或AAH时, 标注的寄存器允许写入; =其他值, 禁止写入。

注3: (P1B): LOCK==55H时, 标注的寄存器允许写入; =其他值, 禁止写入。

## 14.5 寄存器说明

### 14.5.1 EPWM预分频寄存器(CLKPSC)

位	符号	描述	复位值
31:24	CLKPSC67	EPWM计数器6和7 时钟预分频 $CLK\_PSC67 = PCLK/(CLKPSC67+1)$ 如果CLKPSC67=0, 预分频器没有时钟输出, CLKDIVn位若选择跟PSC相关的时钟时, 计数器不工作	0x0
23:16	CLKPSC45	EPWM计数器4和5 时钟预分频 $CLK\_PSC45 = PCLK/(CLKPSC45+1)$ 如果CLKPSC45=0, 预分频器没有时钟输出, CLKDIVn位若选择跟PSC相关的时钟时, 计数器不工作	0x0
15:8	CLKPSC23	EPWM计数器2和3 时钟预分频 $CLK\_PSC23 = PCLK/(CLKPSC23+1)$ 如果CLKPSC23=0, 预分频器没有时钟输出, CLKDIVn位若选择跟PSC相关的时钟时, 计数器不工作	0x0
7:0	CLKPSC01	EPWM计数器0和1 时钟预分频 $CLK\_PSC01 = PCLK/(CLKPSC01+1)$ 如果CLKPSC01=0, 预分频器没有时钟输出, CLKDIVn位若选择跟PSC相关的时钟时, 计数器不工作	0x0

### 14.5.2 EPWM时钟选择寄存器(CLKDIV)

位	符号	描述	复位值
31	-	保留	-
30:28	CLKDIV7	计数器7时钟分频选择 000: CLK_PSC67/2 001: CLK_PSC67/4 010: CLK_PSC67/8 011: CLK_PSC67/16 100: CLK_PSC67/1 其他值: PCLK	0x0
27	-	保留	-
26: 24	CLKDIV6	计数器6时钟分频选择 000: CLK_PSC67/2 001: CLK_PSC67/4 010: CLK_PSC67/8 011: CLK_PSC67/16 100: CLK_PSC67/1 其他值: PCLK	0x0
23	-	保留	-
22:20	CLKDIV5	计数器5时钟分频选择 000: CLK_PSC45/2 001: CLK_PSC45/4	0x0

		010: CLK_PSC45/8 011: CLK_PSC45/16 100: CLK_PSC45/1 其他值: PCLK	
19	-	保留	-
18:16	CLKDIV4	计数器4时钟分频选择 000: CLK_PSC45/2 001: CLK_PSC45/4 010: CLK_PSC45/8 011: CLK_PSC45/16 100: CLK_PSC45/1 其他值: PCLK	0x0
15	-	保留	-
14:12	CLKDIV3	计数器3时钟分频选择 000: CLK_PSC23/2 001: CLK_PSC23/4 010: CLK_PSC23/8 011: CLK_PSC23/16 100: CLK_PSC23/1 其他值: PCLK	0x0
11	-	保留	-
10:8	CLKDIV2	计数器2时钟分频选择 000: CLK_PSC23/2 001: CLK_PSC23/4 010: CLK_PSC23/8 011: CLK_PSC23/16 100: CLK_PSC23/1 其他值: PCLK	0x0
7	-	保留	-
6:4	CLKDIV1	计数器1时钟分频选择 000: CLK_PSC01/2 001: CLK_PSC01/4 010: CLK_PSC01/8 011: CLK_PSC01/16 100: CLK_PSC01/1 其他值: PCLK	0x0
3	-	保留	-
2:0	CLKDIV0	计数器0时钟分频选择 000: CLK_PSC01/2 001: CLK_PSC01/4 010: CLK_PSC01/8 011: CLK_PSC01/16 100: CLK_PSC01/1 其他值: PCLK	0x0

### 14.5.3 EPWM控制寄存器(CON)

位	符号	描述	复位值
31:27	-	保留	-
26	HALTMS	HALT（调试暂停）时EPWMn通道状态控制位 （若POENn=0，EPWMn的输出为高阻态） 0：所有通道正常输出（POENn=1） 1：所有通道输出刹车数据（POENn=1） （在调试态下，运行至断点/单步后或者操作STOP按钮后暂停时，EPWMn的输出为刹车数据。）	0
25:24	MODE	EPWM工作模式选择 00：独立模式 01：互补模式 10：同步模式 11：保留	0x0
23	GROUNPEN	EPWM成组功能使能位 0：所有的PWM通道相互独立 1：EPWM0控制EPWM2,EPWM4, EPWM6,EPWM1控制EPWM3,EPWM5,EPWM7	0
22	ASYMEN	EPWM中心对齐方式下非对称计数使能 0：对称计数使能 1：非对称计数使能	0
21	CNTTYPE	EPWM计数对齐方式选择 0：边沿对齐 1：中心对齐	0
20	-	保留	-
19	EN_DT67	EPWM计数器6和7死区使能位 0：禁止计数器6和7死区 1：使能计数器6和7死区	0
18	EN_DT45	EPWM计数器4和5死区使能位 0：禁止计数器4和5死区 1：使能计数器4和5死区	0
17	EN_DT23	EPWM计数器2和3死区使能位 0：禁止计数器2和3死区 1：使能计数器2和3死区	0
16	EN_DT01	EPWM计数器0和1死区使能位 0：禁止计数器0和1死区 1：使能计数器0和1死区	0
15	PINV7	EPWM7输出极性控制位 0：正常输出 1：反相输出	0
14	PINV6	EPWM6输出极性控制位 0：正常输出 1：反相输出	0
13	PINV5	EPWM5输出极性控制位 0：正常输出 1：反相输出	0
12	PINV4	EPWM4输出极性控制位 0：正常输出	0



		1: 反相输出	
11	PINV3	EPWM3输出极性控制位 0: 正常输出 1: 反相输出	0
10	PINV2	EPWM2输出极性控制位 0: 正常输出 1: 反相输出	0
9	PINV1	EPWM1输出极性控制位 0: 正常输出 1: 反相输出	0
8	PINV0	EPWM0输出极性控制位 0: 正常输出 1: 反相输出	0
7	CNTMODE7	EPWM7自动加载/单次模 0: 单次模式 1: 自动加载模式	0
6	CNTMODE6	EPWM6自动加载/单次模 0: 单次模式 1: 自动加载模式	0
5	CNTMODE5	EPWM5自动加载/单次模 0: 单次模式 1: 自动加载模式	0
4	CNTMODE4	EPWM4自动加载/单次模式 0: 单次模式 1: 自动加载模式	0
3	CNTMODE3	EPWM3自动加载/单次模式 0: 单次模式 1: 自动加载模式	0
2	CNTMODE2	EPWM2自动加载/单次模式 0: 单次模式 1: 自动加载模式	0
1	CNTMODE1	EPWM1自动加载/单次模式 0: 单次模式 1: 自动加载模式	0
0	CNTMODE0	EPWM0自动加载/单次模式 0: 单次模式 1: 自动加载模式	0

## 14.5.4 EPWM控制寄存器(CON2)

位	符号	描述	复位值
31:8	-	保留	-
7	CNTEN7	EPWM7计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
6	CNTEN6	EPWM6计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
5	CNTEN5	EPWM5计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
4	CNTEN4	EPWM4计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
3	CNTEN3	EPWM3计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
2	CNTEN2	EPWM2计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
1	CNTEN1	EPWM1计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0
0	CNTEN0	EPWM0 计数器使能位 0: 禁止 1: 使能 (单次模式完成后自动清除该位)	0

### 14.5.5 EPWM控制寄存器(CON3)

位	符号	描述	复位值
31:30	LOADTYP7	EPWM7加载/中断方式选择位 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
29:28	LOADTYP6	EPWM6加载/中断方式选择位 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
27:26	LOADTYP5	EPWM5加载/中断方式选择位 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
25:24	LOADTYP4	EPWM4加载/中断方式选择位 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
23:22	LOADTYP3	EPWM3加载/中断方式选择 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
21:20	LOADTYP2	EPWM2加载/中断方式选择 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
19:18	LOADTYP1	EPWM1加载/中断方式选择位 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
17:16	LOADTYP0	EPWM0加载/中断方式选择 00: 每个零点与周期点均加载与产生中断标志 01: 每个零点加载与产生中断标志 10: 第一个零点与下一个周期点交替加载与产生中断标志 11: 每两个零点加载与产生中断标志	0x0
15	LOADEN7	EPWM7周期/比较器加载使能 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
14	LOADEN6	EPWM6周期/比较器加载使能	0

		0: 禁止 1: 使能 (加载完毕硬件自动清零)	
13	LOADEN5	EPWM5周期/比较器加载使能 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
12	LOADEN4	EPWM4周期/比较器加载使能位 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
11	LOADEN3	EPWM3周期/比较器加载使能位 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
10	LOADEN2	EPWM2周期/比较器加载使能位 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
9	LOADEN1	EPWM1周期/比较器加载使能位 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
8	LOADEN0	EPWM0周期/比较器加载使能位 0: 禁止 1: 使能 (加载完毕硬件自动清零)	0
7	CNTCLR7	EPWM7 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
6	CNTCLR6	EPWM6 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
5	CNTCLR5	EPWM5 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
4	CNTCLR4	EPWM4 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
3	CNTCLR3	EPWM3计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
2	CNTCLR2	EPWM2 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
1	CNTCLR1	EPWM1 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0
0	CNTCLR0	EPWM0 计数器清零位 0: 禁止 1: 使能 (硬件自动清零)	0

注1: ADCCON3[31:16] 为中心对齐模式下EPWMn加载/中断方式选择的描述, 边沿对齐模式下的EPWMn加载/中断方式选择见章节14.3.7。

### 14.5.6 EPWM控制寄存器(CON4)

位	符号	描述	复位值
31:4	-	保留	-
3	LOADNWINT	EPWM加载与中断标志相关控制位 1: 加载控制与中断标志不相关 0: 加载控制与中断标志相关 加载动作产生后, 中断标志是否随加载产生, 不相关则每个周期和零点都会产生中断标志。	0
2	LETGHALL	HALL状态触发LOADENn使能位 0: 禁止 1: 使能HALL状态变化触发LOADENn=1 注: 如果HALL检测状态改变, 则将EPWM0-EPWM7的加载使能位置1。	0
1	LETGACMP1	ACMP1触发LOADENn使能位 0: 禁止 1: 使能ACMP1时触发LOADENn=1 注: 如果ACMP1事件产生, 则将EPWM0-EPWM7的加载使能位置1。	0
0	LETGACMP0	ACMP0触发LOADENn使能位 0: 禁止 1: 使能ACMP0时触发LOADENn=1 注: 如果ACMP0事件产生, 则将EPWM0-EPWM7的加载使能位置1。	0

### 14.5.7 EPWM周期寄存器0-7(PERIOD0-7)

位	符号	描述	复位值
31:16	-	保留	-
15:0	PERIODn	EPWMn计数器周期值	0x0

### 14.5.8 EPWM比较寄存器0-7(CMPDAT0-7)

位	符号	描述	复位值
31:16	CMPDDATn	EPWMn计数器向下比较值	0x0
15:0	CMPDATn	EPWMn计数器比较值	0x0

### 14.5.9 EPWM输出控制寄存器(POEN)

位	符号	描述	复位值
31:12	-	保留	-
11	MASKLE	EPWM掩码控制预设数据加载使能位 0: 禁止 1: 使能 (允许MASKNXT寄存器的数据加载到MASK寄存器中, 同时禁止写MASK寄存器。另外该位为1时并不立即加载掩码数据, 需要到相应加载点时才加载)	0
10:8	MASKLS	EPWM掩码控制数据加载时刻选择位 000: 在EPWM0的加载点加载 001: 在EPWM1的加载点加载 010: 在EPWM2的加载点加载 011: 在EPWM3的加载点加载 100: 在EPWM4的加载点加载 101: 在EPWM5的加载点加载 110: 在EPWM6的加载点加载 111: 在EPWM7的加载点加载	0x0
7:0	POENn	EPWMn输出使能位 0: EPWM通道n输出禁止 1: EPWM通道n输出使能	0x0

### 14.5.10 EPWM输出通道重映射使能寄存器1(POREMAP\_EN)

位	符号	描述	复位值
16:8	-	保留	-
7:0	PWMRMEN	EPWM通道重映射功能使能控制 AAH: 重映射功能使能 EPWMn由PWMnRM选择何种通道输出 其他: 重映射功能禁止 EPWMn固定通道输出如下: EPWM0<- IPG0 EPWM1<- IPG1 EPWM2<- IPG2 EPWM3<- IPG3 EPWM4<- IPG4 EPWM5<- IPG5 EPWM6<- IPG6 EPWM7<- IPG7	0x0

### 14.5.11 EPWM输出通道重映射寄存器(POREMAP)

位	符号	描述	复位值
31	-	保留	-
30:28	PWM7RM	EPWM通道7重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x7
27	-	保留	-
26:24	PWM6RM	EPWM通道6重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x6
23	-	保留	-
22:20	PWM5RM	EPWM通道5重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x5
19	-	保留	-
18:16	PWM4RM	EPWM通道4重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x4
15	-	保留	-
14:12	PWM3RM	EPWM通道3重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出	0x3

		011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	
11	-	保留	-
10:8	PWM2RM	EPWM通道2重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x2
7	-	保留	-
6:4	PWM1RM	EPWM通道1重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x1
3	-	保留	-
2:0	PWM0RM	EPWM通道0重映射选择位 000: 映射IPG0的输出 001: 映射IPG1的输出 010: 映射IPG2的输出 011: 映射IPG3的输出 100: 映射IPG4的输出 101: 映射IPG5的输出 110: 映射IPG6的输出 111: 映射IPG7的输出	0x0



## 14.5.12 EPWM故障保护控制寄存器(BRKCTL)

位	符号	描述	复位值
31	BRKEN	EPWM故障保护功能总使能位 0: 禁止 (复位故障保护电路) 1: 使能	0
30	BRKAF	EPWM故障信号标志位 (只读) 0: 没有产生故障 1: 产生了故障信号或刹车信号保持有效	0
29:28	BRKMS	EPWM故障保护模式选择位 00: 停止模式 01: 暂停模式 10: 恢复模式 11: 延时恢复模式 注: 切换故障保护模式时, 必须禁止故障保护总使能后, 然后切换故障保护模式, 最后打开故障保护总使能位。	0x0
27	BRKCLR	EPWM故障保护清除位 (只写) 0: -- 1: 清除故障保护状态 注: 仅当BRKAF=1时才能写1, 进行故障清除操作, 否则操作无效。	0
26:24	BRKRCS	EPWM故障恢复加载点选择位 000: EPWM0的加载点恢复 001: EPWM1的加载点恢复 010: EPWM2的加载点恢复 011: EPWM3的加载点恢复 100: EPWM4的加载点恢复 101: EPWM5的加载点恢复 110: EPWM6的加载点恢复 111: EPWM7的加载点恢复	0x0
23	ACMP1BKLE	模拟比较器1输出电平控制刹车使能位 0: 禁止 1: 使能	0
22	ACMP1BKLS	模拟比较器1输出电平控制刹车选择位 0: 低电平产生刹车 1: 高电平产生刹车	0
21	ACMP0BKLE	模拟比较器0输出电平控制刹车使能位 0: 禁止 1: 使能	0
20	ACMP0BKLS	模拟比较器0输出电平控制刹车选择位 0: 低电平产生刹车 1: 高电平产生刹车	0
19	ACMP1BKEN	模拟比较器1输出事件控制刹车使能位 0: 禁止 1: 使能 (比较器输出事件指产生上升沿/下降沿/双沿, 可在ACMP->CEVCON选择何种边沿)	0
18	ACMP0BKEN	模拟比较器0输出事件控制刹车使能位 0: 禁止	0

		1: 使能 (比较器输出事件指产生上升沿/下降沿/双沿, 可在ACMP->CEVCON选择何种边沿)	
17	ADCMP1BKEN	ADC比较器1输出刹车使能位 0: 禁止 1: 使能	0
16	ADCMP0BKEN	ADC比较器0输出刹车使能位 0: 禁止 1: 使能	0
15:14	-	保留	-
13	BRKOSF	EPWM故障保护输出状态标志位 (只读) 0: EPWMn通道为正常输出状态 1: EPWMn通道为输出BRKODn的数据状态	0
12	SWBRK	软件刹车使能位 0: 禁止软件刹车 1: 立即产生软件刹车	0
11	EXTBRKEE	外部硬件刹车边沿检测使能位 0: 禁止 1: 使能	0
10	EXTBRKES	外部硬件刹车边沿检测选择位 0: 下降沿触发刹车 1: 上升沿触发刹车	0
9	EXTBRKLE	外部硬件刹车电平检测使能位 0: 禁止 1: 使能	0
8	EXTBRKLS	外部硬件刹车电平检测选择位 0: 低电平产生刹车 1: 高电平产生刹车	0
7:0	BRKODn	EPWMn刹车输出电平选择位 0: 当故障刹车后, 通道n输出低电平 1: 当故障刹车后, 通道n输出高电平	0x0

### 14.5.13 EPWM死区长度寄存器(DTCTL01)

位	符号	描述	复位值
31:16	DTI01L	通道0和通道1左死区长度寄存器 死区时间 = PWM_CLK01 × DTI01L	0x0
15:0	DTI01R	通道0和通道1右死区长度寄存器 死区时间 = PWM_CLK01 × DTI01R	0x0

### 14.5.14 EPWM死区长度寄存器(DTCTL23)

位	符号	描述	复位值
31:16	DTI23L	通道2和通道3左死区长度寄存器 死区时间 = PWM_CLK23 × DTI23L	0x0
15:0	DTI23R	通道2和通道3右死区长度寄存器 死区时间 = PWM_CLK23 × DTI23R	0x0

### 14.5.15 EPWM死区长度寄存器(DTCTL45)

位	符号	描述	复位值
31:16	DTI45L	通道4和通道5左死区长度寄存器 死区时间 = PWM_CLK45 × DTI45L	0x0
15:0	DTI45R	通道4和通道5右死区长度寄存器 死区时间 = PWM_CLK45 × DTI45R	0x0

### 14.5.16 EPWM死区长度寄存器(DTCTL67)

位	符号	描述	复位值
31:16	DTI67L	通道6和通道7左死区长度寄存器 死区时间 = PWM_CLK67 × DTI67L	0x0
15:0	DTI67R	通道6和通道7右死区长度寄存器 死区时间 = PWM_CLK67 × DTI67R	0x0

### 14.5.17 EPWM掩码输出控制寄存器(MASK)

位	符号	描述	复位值
31:16	-	保留	-
15	MASKEN7	EPWM7掩码输出使能位 0: 禁止 1: 使能	0
14	MASKEN6	EPWM6掩码输出使能位 0: 禁止 1: 使能	0
13	MASKEN5	EPWM5掩码输出使能位 0: 禁止 1: 使能	0
12	MASKEN4	EPWM4掩码输出使能位 0: 禁止 1: 使能	0
11	MASKEN3	EPWM3掩码输出使能位 0: 禁止 1: 使能	0
10	MASKEN2	EPWM2掩码输出使能位 0: 禁止 1: 使能	0
9	MASKEN1	EPWM1掩码输出使能位 0: 禁止 1: 使能	0
8	MASKEN0	EPWM0掩码输出使能位 0: 禁止 1: 使能	0
7	MASKD7	EPWM7 掩码数据 0: 输出0 1: 输出1	0
6	MASKD6	EPWM6 掩码数据 0: 输出0 1: 输出1	0
5	MASKD5	EPWM5 掩码数据 0: 输出0 1: 输出1	0
4	MASKD4	EPWM4 掩码数据 0: 输出0 1: 输出1	0
3	MASKD3	EPWM3 掩码数据 0: 输出0 1: 输出1	0
2	MASKD2	EPWM2 掩码数据 0: 输出0 1: 输出1	0
1	MASKD1	EPWM1 掩码数据	0

		0: 输出0 1: 输出1	
0	MASKD0	EPWM0 掩码数据 0: 输出0 1: 输出1	0

### 14.5.18 EPWM掩码输出控制预设寄存器(MASKNXT)

位	符号	描述	复位值
31:25	-	保留	-
24	HALLEN	HALL检测模式使能位 0: 禁止 1: 使能	0
23	HALLCLR	HALL错误状态清除位 0: 写0无效 1: 写1清除HALLST的错误状态, 让其回到初始状态000。 读为0。 注1: 如果出现的错误的状态或者序列时, HALLST=111, HALL检测功能停止。再次检测开启HALL状态时, 需要写1清除111的状态。	0
22:20	HALLST	HALL接口的状态位 (只读) 检测对应{CAP2,CAP1,CAP0}的状态 000: 状态为0 (初始状态) 001: 状态为1 010: 状态为2 011: 状态为3 100: 状态为4 101: 状态为5 110: 状态为6 111: 错误状态 注1: 该状态为芯片内部检测到HALL接口的状态, 可由此判断是否进入了有效状态, 如果3个HALL传感器的状态出现错误或者状态的顺序出现错误, 则该状态位为111。 有效序列1: ...6-2-3-1-5-4-6-... 有效序列2: ...6-4-5-1-3-2-6-... 注2: 在有效的状态位下, 如果使能掩码预设数据加载功能, 则相应的掩码预设缓存数据在加载点装入到MASK寄存器中。如HALL检测改变为状态3后, 在进入状态3后第一个加载点将掩码预设缓存3的数据装入MASK寄存器中。 注3: 在初始状态000或错误状态111下输出掩码预设缓存7的数据。	0x0
19	-	保留	-
18:16	PMASKSEL	掩码预设缓存选择位; 000: 选择掩码预设缓存0 001: 选择掩码预设缓存1 010: 选择掩码预设缓存2 011: 选择掩码预设缓存3 100: 选择掩码预设缓存4 101: 选择掩码预设缓存5 110: 选择掩码预设缓存6 111: 选择掩码预设缓存7 注1: 该选择位影响读写低16位的数据, EPWM内部存在8个掩码预设缓存 如为000: 则该寄存器读写低16位为掩码缓存0中的数据, 如为001: 则该寄存器读写低16位为掩码缓存1中的数据 如为110: 则该寄存器读写低16位为掩码缓存6中的数据。	0x0

		注2: HALLEN=0时, 默认加载掩码预设缓存0中的数据。	
15	PMASKEN7	EPWM7掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
14	PMASKEN6	EPWM6掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
13	PMASKEN5	EPWM5掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
12	PMASKEN4	EPWM4掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
11	PMASKEN3	EPWM3掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
10	PMASKEN2	EPWM2掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
9	PMASKEN1	EPWM1掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
8	PMASKEN0	EPWM0掩码输出使能预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
7	PMASKD7	EPWM7掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
6	PMASKD6	EPWM6掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
5	PMASKD5	EPWM5 掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
4	PMASKD4	EPWM4 掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
3	PMASKD3	EPWM3 掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
2	PMASKD2	EPWM2 掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
1	PMASKD1	EPWM1 掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0
0	PMASKD0	EPWM0 掩码数据预设位 (该位可设置在EPWMn的加载点加载到MASK寄存器中)	0

### 14.5.19 EPWM触发比较寄存器(CMPTGD0-1)

位	符号	描述	复位值
31:20	-	保留	-
19	CMPTGDSn	EPWM计数比较器n触发模式 (在中心对齐计数方式下生效) 0: 向下计数时触发 1: 向上计数时触发	0
18:16	CMPPCHSn	EPWM 数字比较器n比较通道选择 000: PWM0的计数器 001: PWM1的计数器 010: PWM2的计数器 011: PWM3的计数器 100: PWM4的计数器 101: PWM5的计数器 110: PWM6的计数器 111: PWM7的计数器	0x0
15:0	CMPTGDn	EPWM计数比较器n触发比较值	0x0

### 14.5.20 EPWM中断使能寄存器(IMSC)

位	符号	描述	复位值
31:24	EN_DIFn n=7-0	EPWMn向下比较中断使能位 0: 禁止 1: 使能	0x0
23:16	EN_UIFn n=7-0	EPWMn向上比较中断使能位 0: 禁止 1: 使能	0x0
15:8	EN_PIFn n=7-0	EPWMn周期中断使能位 0: 禁止 1: 使能	0x0
7:0	EN_ZIFn n=7-0	EPWMn零点中断使能位 0: 禁止 1: 使能	0x0

### 14.5.21 EPWM中断使能寄存器2(IMSC2)

位	符号	描述	复位值
31:4	-	保留	-
3	EN_BRKIF	EPWM故障中断使能位 0: 禁止 1: 使能	0
2	EN_HALLIF	HALL状态错误中断使能位 0: 禁止 1: 使能	0
1	EN_DC1IF	计数比较器1中断使能位	0



		0: 禁止 1: 使能	
0	EN_DC0IF	计数比较器0中断使能位 0: 禁止 1: 使能	0

### 14.5.22 EPWM中断源状态寄存器(RIS)

位	符号	描述	复位值
31:24	RIS_DIFn n=7-0	EPWMn向下比较中断源状态位 0: 未产生中断 1: 已产生中断	0x0
23:16	RIS_UIFn n=7-0	EPWMn向上比较中断源状态位 0: 未产生中断 1: 已产生中断	0x0
15:8	RIS_PIFn n=7-0	EPWMn周期中断源状态位 0: 未产生中断 1: 已产生中断	0x0
7:0	RIS_ZIFn n=7-0	EPWMn零点中断源状态位 0: 未产生中断 1: 已产生中断	0x0

### 14.5.23 EPWM中断源状态寄存器2(RIS2)

位	符号	描述	复位值
31:4	-	保留	-
3	RIS_BRKIF	EPWM故障中断源状态位 0: 未产生中断 1: 已产生中断	0
2	RIS_HALLIF	HALL状态错误中断源状态位 0: 未产生中断 1: 已产生中断	0
1	RIS_DC1IF	计数比较器1中断状态位 0: 禁止 1: 使能	0
0	RIS_DC0IF	计数比较器0中断状态位 0: 禁止 1: 使能	0

### 14.5.24 EPWM已使能中断状态寄存器(MIS)

位	符号	描述	复位值
31:24	MIS_DIFn n=7-0	EPWMn向下比较已使能中断状态位 0: 未产生中断 1: 已使能并产生中断	0x0
23:16	MIS_UIFn n=7-0	EPWMn向上比较已使能中断状态位 0: 未产生中断 1: 已使能并产生中断	0x0
15:8	MIS_PIFn n=7-0	EPWMn周期已使能中断状态位 0: 未产生中断 1: 已使能并产生中断	0x0
7:0	MIS_ZIFn n=7-0	EPWMn零点已使能中断状态位 0: 未产生中断 1: 已使能并产生中断	0x0

### 14.5.25 EPWM已使能中断状态寄存器2(MIS2)

位	符号	描述	复位值
31:4	-	保留	-
3	MIS_BRKIF	EPWM故障已使能中断状态位 0: 未产生中断 1: 已使能并产生中断	0
2	MIS_HALLIF	HALL状态错误已使能中断状态位 0: 未产生中断 1: 已使能并产生中断	0
1	RIS_DC1IF	计数比较器1已使能中断状态位 0: 禁止 1: 使能	0
0	RIS_DC0IF	计数比较器0已使能中断状态位 0: 禁止 1: 使能	0

### 14.5.26 EPWM中断清零控制寄存器(ICLR)

位	符号	描述	复位值
31:24	ICLR_DIFn n=7-0	EPWMn向下比较中断清零控制位 写0: 不影响 写1: 清零RIS_DIFn标志位	0x0
23:16	ICLR_UIFn n=7-0	EPWMn向上比较中断清零控制位 写0: 不影响 写1: 清零RIS_UIFn标志位	0x0
15:8	ICLR_PIFn n=7-0	EPWMn周期中断清零控制位 写0: 不影响	0x0

		写1: 清零RIS_PIFn标志位	
7:0	ICLR_ZIFn n=7-0	EPWMn零点中断清零控制位 写0: 不影响 写1: 清零RIS_ZIFn标志位	0x0

### 14.5.27 EPWM中断清零控制寄存器2(ICLR2)

位	符号	描述	复位值
31:4	-	保留	-
3	ICLR_BRKIF	EPWM故障中断清零控制位 写0: 不影响 写1: 清零RIS_BRKIF标志位	0
2	ICLR_HALLIF	HALL状态错误中断清零控制位 写0: 不影响 写1: 清零RIS_HALLIF标志位 注: 如果HALLST=111, 则无法清除RIS_HALLIF标志位	0
1	ICLR_DC1IF	计数比较器1中断清零控制位 写0: 不影响 写1: 清零RIS_DC1IF标志位	0
0	ICLR_DC0IF	计数比较器0中断清零控制位 写0: 不影响 写1: 清零RIS_DC0IF标志位	0

### 14.5.28 EPWM中断累加控制寄存器(IFA)

位	符号	描述	复位值
31:16	-	保留	-
15:12	BRKIFCMP	故障保护中断累加比较值 当故障中断累加到(BRKIFCMP+1)时, BRKIF中断标志位置1	0x0
11:9	-	保留	-
8	BRKIFAEN	故障保护中断累加使能位 0: 禁止 1: 使能	0
7:4	ZIFCMP	零点中断累加比较值 当相应通道的零点中断累加到(ZIFCMP+1)时, ZIFn中断标志位置1 (所有的通道为同一个比较值)	0x0
3:1	-	保留	-
0	ZIFAEN	零点中断累加使能位 0: 禁止 1: 使能	0

### 14.5.29 EPWM写使能控制寄存器(LOCK)

位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0x55时, 使能操作EPWM其他寄存器; 当LOCK=0xaa时, 只使能操作EPWM周期寄存器和比较寄存器; 当LOCK=其他值时, 禁止操作EPWM相关寄存器。	0x0

### 14.5.30 EPWM故障保护恢复延时寄存器(BRKRDT)

位	符号	描述	复位值
31:20	-	必须为0	0x0
19:16	FILS	故障保护(刹车)信号滤波时间选择位 0000: (0~1)* TPCLK 0001: (1~2)* TPCLK 0010: (2~3)* TPCLK 0011: (4~5)* TPCLK 0100: (8~9)* TPCLK 0101: (16~17)* TPCLK 0110: (24~25)* TPCLK 0111: (32~33)* TPCLK 1000: (48~49)* TPCLK 1001: (64~65)* TPCLK 1010: (80~81)* TPCLK 1011: (96~97)* TPCLK 1100: (112~113)* TPCLK 其他: (0~1)* TPCLK	0x0
15:0	RDT	故障保护恢复延时(仅延时恢复模式有效) 延时时间 = (RDT+1)*T <sub>PCLK</sub>	0x0

# 第15章 通用异步收发器 (UARTn n=0/1) 概述

包含2路通用异步串行接口。

## 15.1 特性

- ◆ 全双工，异步通信。
- ◆ 可编程串行接口特性。
  - 数据位长度可设为 5~8 位。
  - 校验位可设为奇偶校验、无校验或固定校验位的产生和检测。
  - 停止位长度可设置为 1 位、1.5 位或 2 位。

## 15.2 功能描述

### 15.2.1 UARTn功能模式

UART为全双工异步通讯接口。UART收发器各包含一个缓冲区，可灵活设置发送字节长度和停止位长度。全双工串行接口通讯参数可设。

### 15.2.2 UARTn中断和状态

UART支持3种类型的中断，具体中断类型包括如下：

- ◆ Line 状态中断（奇偶校验错误、帧错误、打断中断）。
- ◆ 接收数据有效中断。
- ◆ 发送保持寄存器空中断。

## 15.3 寄存器映射

(UART0基地址 = 0x4006\_4000; UART1基地址 = 0x4006\_4100) RO: 只读; WO: 只写, RW: 读写;

寄存器	偏移量	读/写	描述	复位值
RBR	0x000	RO	接收缓存寄存器	-
THR	0x004	WO	发送缓存寄存器	-
DLR	0x008	R/W	波特率分频寄存器	0x00000001
IER	0x00c	R/W	中断使能寄存器	0x00000000
IIR	0x010	RO	中断状态寄存器	0x00000001
LCR	0x018	R/W	线控制寄存器	0x00000000
MCR	0x01C	R/W	Modem控制寄存器	0x00000000
LSR	0x020	RO	线状态寄存器	0x00000060

## 15.4 寄存器说明

### 15.4.1 接收缓存寄存器 (RBR)

位	符号	描述	复位值
31:8	-	保留	-
7:0	RBR	读操作, 返回接收到的来自接收Buffer的数据	-

### 15.4.2 发送缓存寄存器 (THR)

位	符号	描述	复位值
31:8	-	保留	-
7:0	THR	写数据到发送Buffer, UART模块随后会将Buffer中数据发送出去	-

### 15.4.3 波特率分频寄存器 (DLR)

位	符号	描述	复位值
31:16	-	保留	-
15:0	DLR	波特率 = PCLK/16×DLR	0x1

### 15.4.4 中断使能寄存器 (IER)

位	符号	描述	复位值
31:3	-	保留	-
2	RLSIE	接收线状态中断使能位 0: 禁止 1: 使能	0
1	THREIE	发送保持寄存器空中断使能位 0: 禁止 1: 使能	0
0	RBRIE	接收数据有效中断/接收定时器溢出中断使能位 0: 禁止 1: 使能	0

### 15.4.5 中断状态寄存器 (IIR)

位	符号	描述	复位值
31:4	-	保留	-
3:1	INTID	中断状态指示 0x0: Modem状态发生改变 0x1: 发送保持寄存器为空 0x2: 接收数据有效 0x3: 接收到线状态	0x0
0	INT STATUS	中断状态 0: 至少一个中断在队列中 1: 没有中断在队列中	1

### 15.4.6 线控制寄存器 (LCR)

位	符号	描述	复位值
31:7	-	保留	-
6	BCON	Break控制位 当该位写1, 使能Break传输, TXD口将会强制输出逻辑0	0
5:4	PSEL	奇偶校验位选择 0x0: 奇校验, 逻辑1的奇数数目在每个字节中被发送和检测 0x1: 偶校验, 逻辑1的偶数数目在每个字节中被发送和检测 0x2: 校验位强制为1 0x3: 校验位强制为0	0x0
3	PEN	奇偶校验位使能 0: 禁止校验位的产生的检测 1: 使能校验位的产生和检测	0
2	SBS	停止位选择 0: 1位停止位 1: 当发送字长为5位时, 停止位为1.5位; 当发送字长为其他时, 停止位为2位	0
1:0	WLS	字长度选择位 0x0: 5位字长 0x1: 6位字长 0x2: 7位字长 0x3: 8位字长	0x0

### 15.4.7 Modem控制寄存器 (MCR)

位	符号	描述	复位值
31:5	-	保留	-
4	MLBM	Modem回环模式 0: 禁止Modem回环模式 1: 使能Modem回环模式	0
3:0	-	保留	-



## 15.4.8 线状态寄存器 (LSR)

位	符号	描述	复位值
31:7	-	保留	-
6	TEMT	发送缓存空标志位 (只读) 0: 发送缓存有未发送完毕的数据 1: 发送缓存为空	0
5	THRE	表示为发送寄存器空标志位 (只读) 0: 发送寄存器有未发送的数据 1: 发送寄存器为空	0
4	BI	打断中断标志位 (只读) 0: 未检测到打断中断 1: 检测到打断中断 当UART数据输入口在一个传输过程中 (起始位, 数据, 校验位, 停止位) 保持为低电平, 则触发打断中断。UART保持空闲状态直到数据输入口为高电平。 可通过读LSR寄存器清零该位	0
3	FE	帧错误标志位 (只读) 0: 未检测到帧错误 1: 检测到帧错误 可通过读LSR寄存器清零该位	0
2	PE	校验位错误标志位 (只读) 0: 未检测到校验位错误 1: 检测到校验位错误 可通过读LSR寄存器清零该位	0
1	-	保留	0
0	RDR	接收数据有效标志位 (只读) 0: 接收区没有未读数据 1: 接收区有未读数据	0

## 第16章 串行接口IICA (IICA)

### 16.1 串行接口IICA的功能

串行接口IICA有以下3种模式。

#### 1) 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

#### 2) I<sup>2</sup>C总线模式（支持多主控）

此模式通过串行时钟（SCLAn）和串行数据总线（SDAAn）的2条线，与多个设备进行8位数据传送。符合I<sup>2</sup>C总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的I<sup>2</sup>C总线控制部分

因为串行接口IICA的SCLAn引脚和SDAAn引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

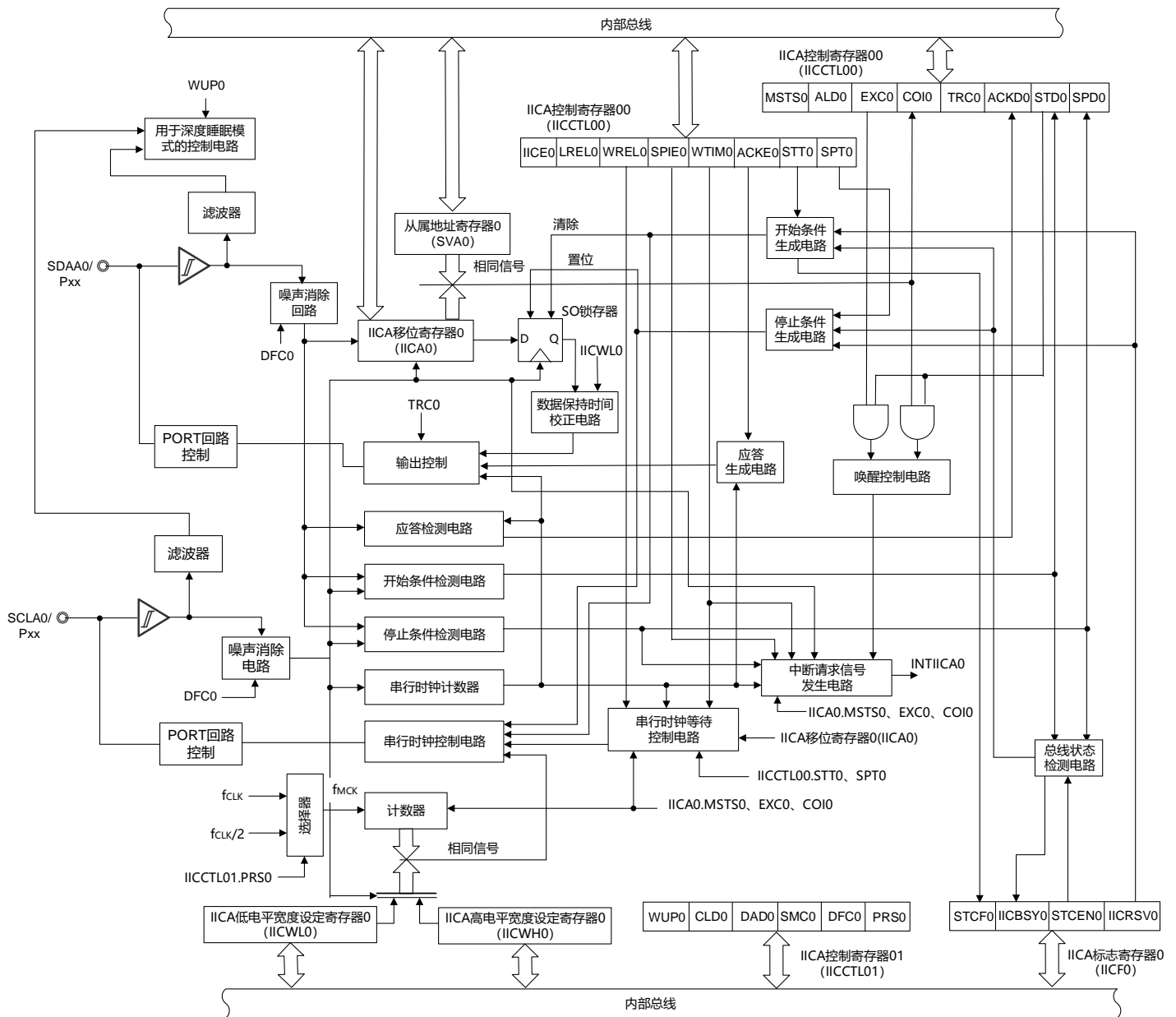
#### 3) 唤醒模式

在深度睡眠模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号（INTIICAn）解除深度睡眠模式。通过IICA控制寄存器n1（IICCTLn1）的WUPn位进行设定。

串行接口IICA的框图如图16-1所示。

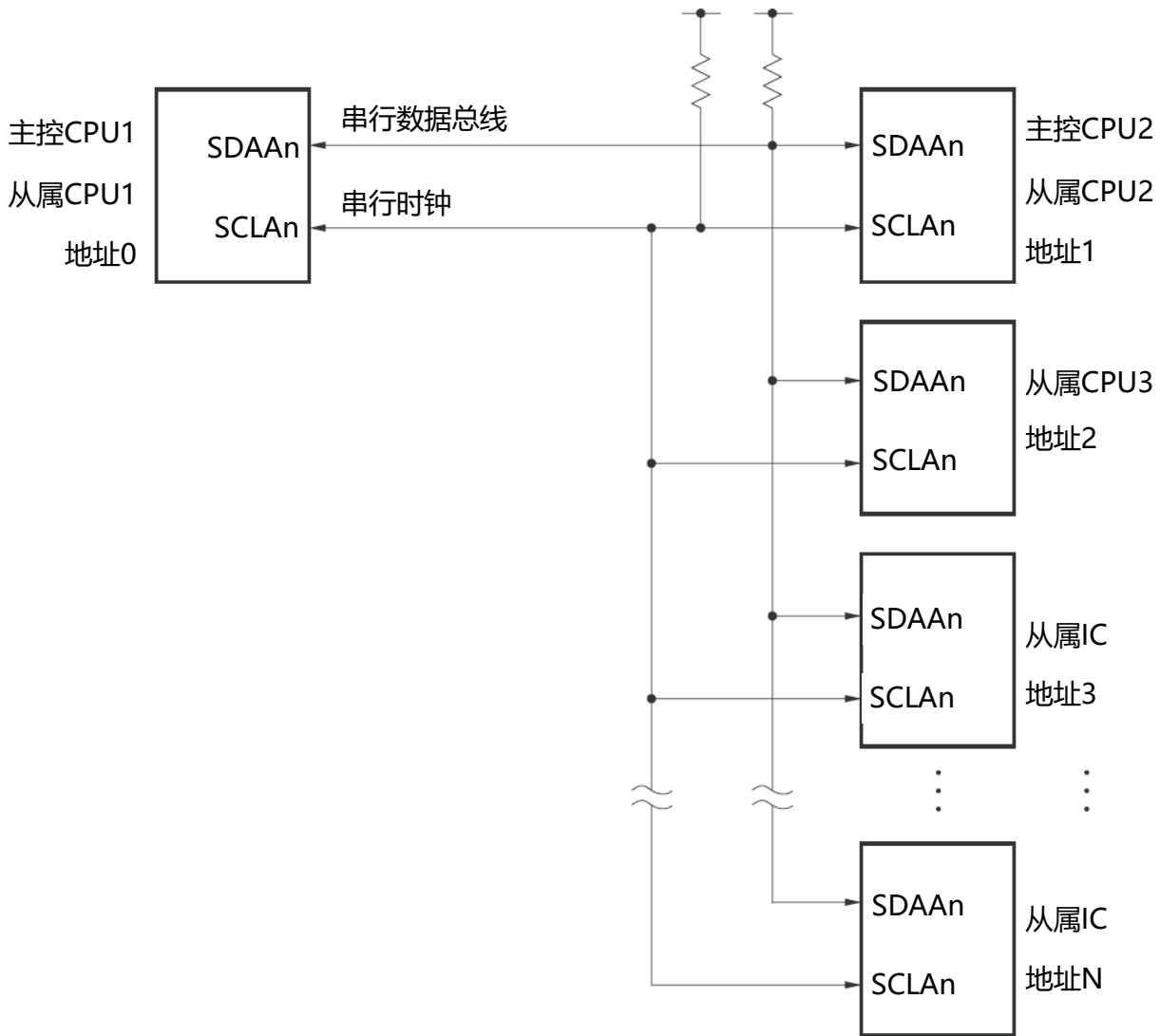
注：n=0。

图16-1 串行接口IICA的框图



串行总线的结构例子如图16-2所示。

图16-2 I<sup>2</sup>C总线的串行总线结构例子



注：n=0。

## 16.2 串行接口IICA的结构

串行接口IICA由以下硬件构成。

表16-1 串行接口IICA的结构

项目	结构
寄存器	IICA移位寄存器n (IICAn) 从属地址寄存器n (SVAn)
控制寄存器	外围允许寄存器0 (PER0) IICA控制寄存器n0 (IICCTLn0) IICA状态寄存器n (IICSn) IICA标志寄存器n (IICFn) IICA控制寄存器n1 (IICCTLn1) IICA低电平宽度设定寄存器n (IICWLn) IICA高电平宽度设定寄存器n (IICWHn) 端口模式寄存器 (PMxx) 端口模式控制寄存器 (PMCxx) 端口复用功能配置寄存器 (PxxCFG)

注1: n=0。

注2: 本产品可将IICA输入/输出引脚功能复用到多个端口。当某端口被配置为IICA引脚的复用功能后, 该端口的N沟道漏极开路输出 (VDD/EVDD耐压) 模式由设计保证自动打开, 即POMxx寄存器不需要用户设置。

### (1) IICA移位寄存器n (IICAn)

IICAn寄存器是与串行时钟同步进行8位串行数据和8位并行数据相互转换的寄存器, 用于发送和接收。能通过读写IICAn寄存器来控制实际的发送和接收。

在等待期间, 通过写IICAn寄存器来解除等待, 开始传送数据。通过8位存储器操作指令去设定IICAn寄存器。在产生复位信号后, 此寄存器的值变为“00H”。

表16-2 IICAn移位寄存器n (IICAn) 的格式

位	符号	描述	复位值
7:0	IICAn	IICAn移位寄存器n	0x0

注1: 在数据传送过程中, 不能给IICAn寄存器写数据。

注2: 只能在等待期间读写IICAn寄存器。除了等待期间以外, 禁止在通信状态下存取IICAn寄存器。但是在主控设备的情况下, 能在将通信触发位 (STTn) 置“1”后写一次IICAn寄存器。

注3: 当预约通信时, 必须在检测到由停止条件产生的中断后给IICAn寄存器写数据。

注4: n=0。

(2) 从属地址寄存器n (SVAn)

这是在用作从属设备时保存7位本地站地址{A6,A5,A4,A3,A2,A1,A0}的寄存器。

通过8位存储器操作指令设定SVAn寄存器。但是，在STDn位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表16-3 从属地址寄存器n (SVAn) 的格式

位	符号	描述	复位值
7:0	SVAn	从属地址寄存器n	0x0

注：bit0固定为“0”。

(3) SO锁存器

SO锁存器保持SDAAn引脚的输出电平。

(4) 唤醒控制电路

当设定在从属地址寄存器n (SVAn) 的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求 (INTIICAn)。

(5) 串行时钟计数器

在发送或者接收过程中，此计数器对输出或者输入的串行时钟进行计数，检查是否进行了8位数据的发送和接收。

(6) 中断请求信号发生电路

此电路控制产生中断请求信号 (INTIICAn)。由以下2种触发产生I2C中断请求。

第8个或者第9个串行时钟的下降 (通过WTIMn位进行设定)

因检测到停止条件而产生中断请求 (通过SPIEn位进行设定)。

注1：WTIMn位：IICA控制寄存器n0 (IICCTLn0) 的bit3

注2：SPIEn位：IICA控制寄存器n0 (IICCTLn0) 的bit4

(7) 串行时钟控制电路

在主机模式中，此电路从采样时钟生成输出到SCLAn引脚的时钟。

(8) 串行时钟等待控制电路

此电路控制等待时序。

(9) 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

(10) 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

(11) 开始条件生成电路

如果将STTn位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下 (IICRSVn位=1) 并且没有释放总线 (IICBSYn位=1) 时，忽视开始条件请求并且将STCFn位置“1”。

(12) 停止条件生成电路

如果将SPTn位置“1”，此电路就生成停止条件。

### (13) 总线状态检测电路

此电路通过检测开始条件和停止条件来检测总线是否被释放。但是，在刚运行时不能立即检测总线状态，因此必须通过STCENn位设定总线状态检测电路的初始状态。

注：TTn位：IICA控制寄存器n0 (IICCTLn0) 的bit1。

SPTn位：IICA控制寄存器n0 (IICCTLn0) 的bit0。

ICRSVn位：IICA标志寄存器n (IICFn) 的bit0。

IICBSYn位：IICA标志寄存器n (IICFn) 的bit6。

STCFn位：IICA标志寄存器n (IICFn) 的bit7。

STCENn位：IICA标志寄存器n (IICFn) 的bit1。

n=0。

## 16.3 寄存器映射

(以下寄存器基地址 = 0x4004\_1A30) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
IICCTLn0	0x000	R/W	IICA控制寄存器n0	0x00
IICCTLn1	0x001	R/W	IICA控制寄存器n1	0x00
IICWLn	0x002	R/W	IICA低电平宽度设定寄存器n	0xFF
IICWHn	0x003	R/W	IICA高电平宽度设定寄存器n	0xFF
SVA0	0x004	R/W	IICA从属地址寄存器	0x00
IICA0	0x120	R/W	IICA移位寄存器	0x00
IICSn	0x121	RO	IICA状态寄存器n	0x00
IICFn	0x122	R/W	IICA标志寄存器n	0x00

注: n=0。

(以下寄存器基地址 = 0x4002\_0420) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER0	0x020	R/W	外围允许寄存器0	0x00



## 16.4 控制串行接口IICA的寄存器说明

通过以下几种寄存器控制串行接口IICA。

外围允许寄存器0 (PER0)

IICA控制寄存器n0 (IICCTLn0)

IICA标志寄存器n (IICFn)

IICA状态寄存器n (IICSn)

IICA控制寄存器n1 (IICCTLn1)

IICA低电平宽度设定寄存器n (IICWLn)

IICA高电平宽度设定寄存器n (IICWHn)

端口模式寄存器 (PMxx)

端口模式控制寄存器 (PMCxx)

端口复用功能配置寄存器 (PxxCFG)

注：n=0。

## 16.5 外围允许寄存器0 (PER0)

PER0寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用串行接口IICAn时，必须将bit4 (IICAEN) 置“1”。

通过8位存储器操作指令设定PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	LSITIMEREN	提供LSITIMER输入时钟的控制(掉电睡眠可行) 0: 停止提供输入时钟，不能写LSITIMER使用的SFR 1: 提供输入时钟，能写LSITIMER使用的SFR	0
6:5	--	保留	0x0
4	IICAEN	提供IICA输入时钟的控制 0: 停止提供输入时钟，不能写IICA使用的SFR 1: 提供输入时钟，能写IICA使用的SFR	0
3:1	--	保留	0x0
0	TM40EN	提供通用定时器单元0输入时钟的控制 0: 停止提供输入时钟，不能写通用定时器单元0使用的SFR 1: 提供输入时钟，能写通用定时器单元0使用的SFR	0

注1：设定串行接口IICA时，必须先在IICAEN位为“1”的状态下设定以下的寄存器。当IICAEN位为“0”时，串行接口IICA的控制寄存器的值为初始值，忽视写操作（端口复用功能配置寄存器（PxxCFG）、端口模式寄存器（PMxx）和端口模式控制寄存器（PMCxx）除外）。

- (1) IICA控制寄存器n0 (IICCTLn0)
- (2) IICA标志寄存器n (IICFn)
- (3) IICA状态寄存器n (IICSn)
- (4) IICA控制寄存器n1 (IICCTLn1)
- (5) IICA低电平宽度设定寄存器n (IICWLn)
- (6) IICA高电平宽度设定寄存器n (IICWHn)

注2：n=0。

## 16.5.1 IICA控制寄存器n0 (IICCTLn0)

这是允许或者停止I<sup>2</sup>C运行、设定等待时序以及设定其他I<sup>2</sup>C运行的寄存器。

通过8位存储器操作指令设定IICCTLn0寄存器。但是必须在IICEn位为“0”时或者在等待期间设定SPIEn位、WTIMn位和ACKEn位，而且在将IICEn位从“0”置为“1”时能同时设定这些位。

在产生复位信号后，此寄存器的值变为“00H”。

注：n=0。

位	符号	描述	复位值
7	IICEn	I <sup>2</sup> C运行的允许 0: 停止运行。对IICA状态寄存器n (IICSn) 进行复位 <sup>注1</sup> ，并且停止内部运行。 1: 允许运行 注意：必须在SCLAn线和SDAAn线为高电平的状态下将此位置“1”。 置位条件 (IICEn=1) : •通过指令置位。 清除条件 (IICEn=0) : •通过指令清除。 •当复位时。	0
6	LRELn <sup>注2、3</sup>	通信的退出 0: 通常运行。 1: 退出当前的通信，进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况下使用。 SCLAn线和SDAAn线变为高阻状态。 IICA控制寄存器n0 (IICCTLn0) 和IICA状态寄存器n (IICSn) 中的以下标志被清“0”: •STTn•SPTn•MSTSn•EXCn•COLn•TRCn•ACKDn•STDn 变为退出通信的待机状态，保持到满足以下的通信参加条件为止。 •在检测到停止条件后作为主控设备启动。 •在检测到开始条件后地址匹配或者接收到扩展码。 置位条件 (LRELn=1) : •通过指令置位。 清除条件 (LRELn=0) : •在执行后自动清除。 •当复位时	0
5	WRELn <sup>注2、3</sup>	等待的解除 0: 不解除等待。 1: 解除等待。在解除等待后自动清除。 如果在发送状态下 (TRCn=1) 的第9个时钟等待期间将WRELn位 (解除等待) 置位，SDAAn线就变为高阻抗状态 (TRCn=0)。 置位条件 (WRELn=1) : •通过指令置位。 清除条件 (WRELn=0) : •在执行后自动清除。 •当复位时	0

4	SPIEn <sup>注4</sup>	<p>允许或者禁止停止条件检测产生的中断请求</p> <p>0: 禁止 1: 允许</p> <p>当IICA控制寄存器n1 (IICCTLn1) 的WUPn位为“1”时, 即使将SPIEn位置“1”也不产生停止条件中断。</p> <p>置位条件 (SPIEn=1) : •通过指令置位。</p> <p>清除条件 (SPIEn=0) : •在执行后自动清除。 •当复位时</p>	0
3	WTIMn <sup>注4</sup>	<p>等待和中断请求的控制</p> <p>0: 在第8个时钟的下降沿产生中断请求信号。 主控设备: 在输出8个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入8个时钟后, 将时钟置为低电平, 然后等待主控设备。</p> <p>1: 在第9个时钟的下降沿产生中断请求信号。 主控设备: 在输出9个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入9个时钟后, 将时钟置为低电平, 然后等待主控设备。</p> <p>在地址传送期间, 与此位的设定无关, 在第9个时钟的下降沿产生中断; 在地址传送结束后, 此位的设定有效。主控设备在地址传送期间的第9个时钟下降沿进入等待状态。接收到本站地址的从属设备在产生应答 (ACK) 后的第9个时钟下降沿进入等待状态, 但是接收到扩展码的从属设备在第8个时钟下降沿进入等待状态。</p> <p>置位条件 (WTIMn=1) : •通过指令置位。</p> <p>清除条件 (WTIMn=0) : •通过指令清除。 •当复位时</p>	0
2	ACKEn <sup>注4、5</sup>	<p>应答控制</p> <p>0: 禁止应答。 1: 允许应答。在第9个时钟期间将SDAAn线置为低电平。</p> <p>置位条件 (ACKEn=1) : •通过指令置位。</p> <p>清除条件 (ACKEn=0) : •通过指令清除。 •当复位时</p>	0
1	STTn <sup>注2、6</sup>	<p>开始条件的触发</p> <p>0: 不生成开始条件。 1: 当总线被释放时 (待机状态, IICBSYn位为“0”) : 如果将此位置“1”, 就生成开始条件 (作为主控设备的启动)。 当第三方正在通信时: •允许通信预约功能的情况 (IICRSVn=0) : 用开始条件预约标志。如果将此位置“1”, 就在释放总线后自动生成开始条件。 •禁止通信预约功能的情况 (IICRSVn=1) : 即使将此位置“1”, 也清除STTn位并且将</p>	0

		<p>STTn清除标志 (STCFn) 置“1”，不生成开始条件。</p> <ul style="list-style-type: none"> <li>•等待状态 (主控设备)：在解除等待后生成重新开始条件。</li> </ul> <p>有关置位时序的注意事项：</p> <ul style="list-style-type: none"> <li>•主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。</li> <li>•主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第9个时钟后的等待期间将此位置“1”。</li> <li>•禁止与停止条件的触发 (SPTn) 同时置“1”。</li> <li>•在将STTn位置“1”后，禁止在满足清除条件前再次将此位“1”。</li> </ul> <p>置位条件 (STTn=1)：</p> <ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul> <p>清除条件 (STTn=0)：</p> <ul style="list-style-type: none"> <li>•在禁止通信预约的状态下将STTn位置“1”。</li> <li>•在仲裁失败时</li> <li>•主控设备生成开始条件。</li> <li>•因LRELn位为“1” (退出通信) 而进行的清除</li> <li>•当IICEn位为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>	
0	SPTn <sup>注7</sup>	<p>停止条件的触发</p> <p>0: 不生成停止条件。</p> <p>1: 生成停止条件 (作为主控设备的传送结束)。</p> <p>有关置位时序的注意事项：</p> <ul style="list-style-type: none"> <li>•主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。</li> <li>•主控发送：在应答期间，可能无法正常生成停止条件。必须在输出第9个时钟后的等待期间将此位置“1”。</li> <li>•禁止与开始条件的触发 (STTn) 同时置“1”。</li> <li>•只有在主控设备的情况下才能将SPTn位置“1”。</li> <li>•在WTIMn位为“0”时，必须注意：如果在输出8个时钟后的等待期间将SPTn位置“1”，就在解除等待后的第9个时钟的高电平期间生成停止条件。必须在输出8个时钟后的等待期间将WTIMn位从“0”置为“1”并且在输出第9个时钟后的等待期间将SPTn位置“1”。</li> <li>•在将SPTn位置“1”后，禁止在满足清除条件前再次将此位置“1”。</li> </ul> <p>置位条件 (SPTn=1)：</p> <ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul> <p>清除条件 (SPTn=0)：</p> <ul style="list-style-type: none"> <li>•当仲裁失败时</li> <li>•在检测到停止条件后自动清除。</li> <li>•因LRELn位为“1” (退出通信) 而进行的清除</li> <li>•当IICEn位为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>	0

注 1：对 IICA 移位寄存器 n (IICAn)、IICA 标志寄存器 n (IICFn) 的 STCFn 位和 IICBSYn 位以及 IICA 控制寄存器 n1 (IICCTLn1) 的 CLDn 位和 DADn 位进行复位。

注 2：在 IICEn 位为“0”的状态下，此位的信号无效。

注 3：LRELn 位和 WRELn 位的读取值总是“0”。

注 4: 在 IICEn 位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。

注 5: 在地址传送过程中不是扩展码时, 设定值无效。当为从属设备并且地址匹配时, 与设定值无关而生成应答。

注 6: STTn 位的读取值总是“0”。

注 7: SPTn 位的读取值总是“0”。

注 8: 如果在 SCLAn 线为高电平、SDAAn 线为低电平并且数字滤波器为 ON (IICCTLn1 寄存器的 DFCn=1) 时允许 I2C 运行 (IICEn=1), 就立即检测开始条件。此时, 必须在允许 I2C 运行 (IICEn=1) 后连续通过位存储器操作指令将 LRELn 位置“1”。

注 9: n=0。在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICCTLn0 寄存器的 bit5 (WRELn) 置“1”来解除等待, 就在清除 TRCn 位 (接收状态) 后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1” (发送状态) 时的等待解除。

注 10: n=0

## 16.5.2 IICA状态寄存器n (IICSn)

这是表示I<sup>2</sup>C状态的寄存器。

只有在STTn位为“1”并且等待期间，才能8位存储器操作指令读IICSn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

注 1：在深度睡眠模式中允许地址匹配唤醒功能（WUPn=1）状态下，禁止读 IICSn 寄存器。在 WUPn 位为“1”的状态下，与 INTIICAn 中断请求无关，如果将 WUPn 位从“1”改为“0”（停止唤醒运行），就在检测到下一个开始条件或者停止条件后才会反映状态的变化。因此，要使用唤醒功能时，必须允许（SPIEn=1）因检测到停止条件而产生的中断，并且在检测到中断后读 IICSn 寄存器。

注 2：STTn：IICA 控制寄存器 n0 (IICCTLn0) 的 bit1, n=0。

注 3：WUPn：IICA 控制寄存器 n1 (IICCTLn1) 的 bit7, n=0。

位	符号	描述	复位值
7	MSTSn	主控状态的确认标志 0: 从属状态或者通信待机状态。 1: 主控通信状态。 置位条件 (MSTSn=1): •当生成开始条件时  清除条件 (MSTSn=0): •当检测到停止条件时 •当ALDn位为“1”（仲裁失败）时 •因LRELn位为“1”（退出通信）而进行的清除 •当IICEn位从“1”变为“0”（停止运行）时 •当复位时•当复位时。	0
6	ALDn	仲裁失败的检测 0: 表示未发生仲裁或者赢得仲裁。 1: 表示仲裁失败。清除MSTSn位。 置位条件 (ALDn=1): •当仲裁失败时  清除条件 (ALDn=0) •在读IICSn寄存器后自动清除 <sup>注1</sup> 。 •当IICEn位从“1”变为“0”（停止运行）时 •当复位时	0
5	EXCn	扩展码的接收检测 0: 未接收到扩展码。 1: 接收到扩展码。 置位条件 (EXCn=1): •当接收的地址数据的高4位为“0000”或者“1111”时 （在第8个时钟的上升沿置位）  清除条件 (EXCn=0): •当检测到开始条件时 •当检测到停止条件时 •因LRELn位为“1”（退出通信）而进行的清除 •当IICEn位从“1”变为“0”（停止运行）时 •当复位时	0
4	COIn	地址匹配的检测 0: 地址不同。 1: 地址相同。	0

		<p>置位条件 (COIn=1) :</p> <ul style="list-style-type: none"> <li>当接收地址和本地站地址 (从属地址寄存器n (SVAn)) 相同时 (在第8个时钟的上升沿置位)</li> </ul> <p>清除条件 (COIn=0) :</p> <ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>当检测到停止条件时</li> <li>因LRELn位为“1” (退出通信) 而进行的清除</li> <li>当IICEEn位从“1”变为“0” (停止运行) 时</li> <li>当复位时</li> </ul>	
3	TRCn	<p>发送/接收的状态检测</p> <p>0: 处于接收状态 (发送状态除外)。将SDAAn线置为高阻抗。</p> <p>1: 处于发送状态。设定为能将SOn锁存器的值输出到SDAAn线 (在第1字节的第9个时钟的下降沿以后有效)。</p> <p>置位条件 (TRCn=1) :</p> <p>&lt;主控设备&gt;</p> <ul style="list-style-type: none"> <li>当生成开始条件时</li> <li>当第1字节 (地址传送) 的LSB (传送方向指示位) 输出“0” (主控发送) 时</li> </ul> <p>&lt;从属设备&gt;</p> <ul style="list-style-type: none"> <li>当主控设备的第1字节 (地址传送) 的LSB (传送方向指示位) 输入“1” (从属发送) 时</li> </ul> <p>清除条件 (TRCn=0) :</p> <p>&lt;主控设备和从属设备&gt;</p> <ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>因LRELn位为“1” (退出通信) 而进行的清除</li> <li>当IICEEn位从“1”变为“0” (停止运行) 时</li> <li>因WRELn位为“1” (解除等待) 而进行的清除注2</li> <li>当ALDn位从“0”变为“1” (仲裁失败) 时</li> <li>当复位时</li> </ul> <p>&lt;主控设备&gt;</p> <ul style="list-style-type: none"> <li>当第1字节的LSB (传送方向指示位) 输出“1”时</li> </ul> <p>&lt;从属设备&gt;</p> <ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>当第1字节的LSB (传送方向指示位) 输入“0”时</li> </ul>	0
2	ACKDn	<p>应答 (ACK) 的检测</p> <p>0: 未检测到应答。</p> <p>1: 检测到应答。</p> <p>置位条件 (ACKDn=1) :</p> <ul style="list-style-type: none"> <li>在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平时</li> </ul> <p>清除条件 (ACKDn=0) :</p> <ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>当下一个字节的第1个时钟上升时</li> <li>因LRELn位为“1” (退出通信) 而进行的清除</li> <li>当IICEEn位从“1”变为“0” (停止运行) 时</li> <li>当复位时</li> </ul>	0
1	STDn	<p>开始条件的检测</p> <p>0: 未检测到开始条件。</p> <p>1: 检测到开始条件, 表示处于地址传送期间。</p> <p>置位条件 (STDn=1) :</p> <ul style="list-style-type: none"> <li>当检测到开始条件时</li> </ul> <p>清除条件 (STDn=0) :</p> <ul style="list-style-type: none"> <li>当检测到停止条件时</li> </ul>	0



		<ul style="list-style-type: none"> <li>•在地址传送后的下一个字节的第1个时钟上升时</li> <li>•因LRELn位为“1”（退出通信）而进行的清除</li> <li>•当IICEn位从“1”变为“0”（停止运行）时</li> <li>•当复位时</li> </ul>	
0	SPDn	停止条件的检测 0: 未检测到停止条件。 1: 检测到停止条件, 主控设备结束通信并且已释放总线。 置位条件 (SPDn=1) : <ul style="list-style-type: none"> <li>•当检测到停止条件时</li> </ul> 清除条件 (SPDn=0) : <ul style="list-style-type: none"> <li>•在将此位置位后, 在检测到开始条件后的地址传送字节的第1个时钟上升时</li> <li>•当WUPn位从“1”变为“0”时</li> <li>•当IICEn位从“1”变为“0”（停止运行）时</li> <li>•当复位时</li> </ul>	0

注 1: 即使对 IICSn 寄存器以外的位, 执行位存储器操作指令, 也清除此位。因此, 在使用 ALDn 位时, 必须在读其他位前先读 ALDn 位的数据。

注 2: 在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置“1”来解除等待, 就在清除 TRCn 位 (接收状态) 后将 SDAA<sub>n</sub> 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1” (发送状态) 时的等待解除。

注 3: LRELn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit6。

注 4: IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7。

注 5: n=0。

### 16.5.3 IICA标志寄存器n (IICFn)

这是设定I<sup>2</sup>C运行模式以及表示I<sup>2</sup>C总线状态的寄存器。

通过8位存储器操作指令设定IICFn寄存器。但是，只能读STTn清除标志 (STCFn) 和I<sup>2</sup>C总线状态标志 (IICBSYn)。

通过IICRSVn位设定允许或者禁止通信预约功能，并且通过STCENn位设定IICBSYn位的初始值。只有在禁止I<sup>2</sup>C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时才能写IICRSVn位和STCENn位。在允许运行后，只能读IICFn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	STCFn	STTn清除标志 0: 发行开始条件。 1: 无法发行开始条件而清除STTn标志 置位条件 (STCFn=1) : •在设定为禁止通信预约 (IICRSVn=1) 的状态下无法发行开始条件而将STTn位清“0”时 清除条件 (STCFn=0) : •因STTn位为“1”而进行的清除 •当IICEn位为“0” (停止运行) 时 •当复位时	0
6	IICBSYn	I <sup>2</sup> C总线状态标志 0: 总线释放状态 (STCENn=1时的通信初始状态) 1: 总线通信状态 (STCENn=0时的通信初始状态) 置位条件 (IICBSYn=1) : •当检测到开始条件时 •STCENn位为“0”时的IICEn位的置位 清除条件 (IICBSYn=0) : •当检测到停止条件时 •当IICEn位为“0” (停止运行) 时 •当复位时	0
5:2	-	保留	-
1	STCENn	初始开始允许触发 0: 在允许运行 (IICEn=1) 后，通过检测停止条件来允许生成开始条件。 1: 在允许运行 (IICEn=1) 后，不检测停止条件而允许生成开始条件。 置位条件 (STCENn=1) : •通过指令置位。 清除条件 (STCENn=0) : •通过指令清除。 •当检测到开始条件时 •当复位时	0
0	IICRSVn	通信预约功能禁止位 0: 允许通信预约。 1: 禁止通信预约。 置位条件 (IICRSVn=1) : •通过指令置位。 清除条件 (IICRSVn=0) : •通过指令清除。	0

注 1: bit6 和 bit7 是只读位。

注 2: 只有在停止运行 (IICEn=0) 时才能写 STCENn 位。

注 3: 如果 STCENn 位为“1”，就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0)，因此为了避免在发行第 1 个开始条件 (STTn=1) 时破坏其他通信，需要确认没有正在通信的第三方。

注 4: 只有在停止运行 (IICEn=0) 时才能写 IICRSVn。

注 5: STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1。

注 6: IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7。

## 16.5.4 IICA控制寄存器n1 (IICCTLn1)

这是用于设定I<sup>2</sup>C运行模式以及检测SCLAn引脚和SDAAn引脚状态的寄存器。

通过8位存储器操作指令设定IICCTLn1寄存器。但是，只能读CLDn位和DADn位。

除了WUPn位以外，必须在禁止I<sup>2</sup>C运行（IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设定IICCTLn1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	WUPn	<p>地址匹配唤醒的控制</p> <p>0: 在深度睡眠模式中，停止地址匹配唤醒功能的运行。</p> <p>1: 在深度睡眠模式中，允许地址匹配唤醒功能的运行。</p> <p>要通过将WUPn位置“1”来转移到深度睡眠模式时，必须在将WUPn位置“1”后至少经过3个fMCK时钟，然后执行深度睡眠指令。(参照图16-3 WUPn位置“1”的流程)</p> <p>在地址匹配或者接收到扩展码后，必须将WUPn位清“0”。能通过将WUPn位清“0”来参加后续的通信（需要在将WUPn位清“0”后解除等待以及写发送数据）。</p> <p>在WUPn位为“1”的状态下，地址匹配或者接收到扩展码时的中断时序与WUPn位为“0”时的中断时序相同（根据时钟产生采样误差的延迟差）。另外，当WUPn位为“1”时，即使将SPIEn位置“1”也不产生停止条件中断。</p> <p>置位条件 (WUPn=1) :</p> <ul style="list-style-type: none"> <li>•通过指令置位 (MSTSn=0、EXCn=0、COIn=0并且STDn=0 (不参加通信)) 注2。</li> </ul> <p>清除条件 (WUPn=0) :</p> <ul style="list-style-type: none"> <li>•通过指令清除 (在地址匹配或者接收到扩展码后)。</li> </ul>	0
6	-	保留	-
5	CLDn	<p>SCLAn引脚的电平检测（只在IICEn位为“1”时有效）</p> <p>0: 检测到SCLAn引脚为低电平。</p> <p>1: 检测到SCLAn引脚为高电平。</p> <p>置位条件 (CLDn=1) :</p> <ul style="list-style-type: none"> <li>•当SCLAn引脚为高电平时</li> </ul> <p>清除条件 (CLDn=0) :</p> <ul style="list-style-type: none"> <li>•当SCLAn引脚为低电平时</li> <li>•当IICEn位为“0”（停止运行）时</li> <li>•当复位时</li> </ul>	0
4	DADn	<p>SDAAn引脚的电平检测（只在IICEn位为“1”时有效）</p> <p>0: 检测到SDAAn引脚为低电平。</p> <p>1: 检测到SDAAn引脚为高电平。</p> <p>置位条件 (DADn=1) :</p> <ul style="list-style-type: none"> <li>•当SDAAn引脚为高电平时</li> </ul> <p>清除条件 (DADn=0) :</p> <ul style="list-style-type: none"> <li>•当SDAAn引脚为低电平时</li> <li>•当IICEn位为“0”（停止运行）时</li> <li>•当复位时</li> </ul>	0
3	SMCn	<p>运行模式的切换</p> <p>0: 在标准模式中运行（最大传送速率：100kbps）。</p>	0

		1: 在快速模式（最大传送速率：400kbps）或者增强型快速模式（最大传送速率：1Mbps）中运行。	
2	DFCn	数字滤波器的运行控制 0: 数字滤波器OFF 1: 数字滤波器ON 必须在快速模式或者增强型快速模式中使用数字滤波器。数字滤波器用于消除噪声 无论是将DFCn位置“1”还是清“0”，传送时钟都不变。	0
1	-	保留	-
0	PRSn	运行时钟（ $f_{MCK}$ ）的控制 0: 选择 $f_{CLK}$ （ $1MHz \leq f_{CLK} \leq 20MHz$ ）。 1: 选择 $f_{CLK}/2$ （ $20MHz < f_{CLK}$ ）。	0

注 1: bit4 和 bit5 是只读位。

注 2: 在以下所示的期间，需要确认 IICA 状态寄存器 n (IICSn) 的状态并且将其置位。

注 3: IICA 运行时钟 ( $f_{MCK}$ ) 的最大工作频率为 20MHz(Max.)。只有在  $f_{CLK}$  超过 20MHz 时才必须将 IICA 控制寄存器 n1 (IICCTLn1) 的 bit0 (PRSn) 置“1”。

注 4: 在设定传送时钟的情况下，必须注意  $f_{CLK}$  的最小工作频率。串行接口 IICA 的  $f_{CLK}$  最小工作频率取决于运行模式。

快速模式:  $f_{CLK}=3.5MHz(\text{Min.})$

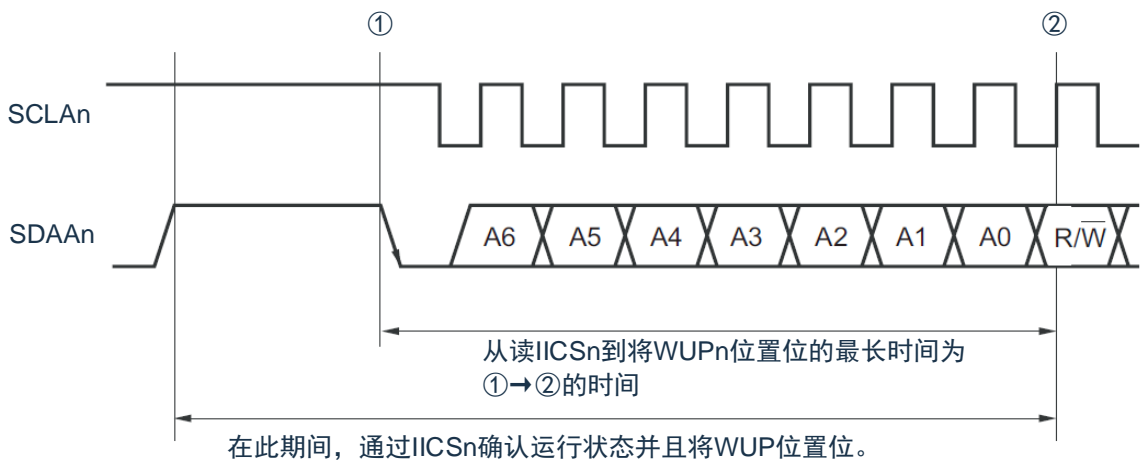
增强型快速模式:  $f_{CLK}=10MHz(\text{Min.})$

标准模式:  $f_{CLK}=1MHz(\text{Min.})$

注 5: IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

注 6: n=0。

图 16-3 WUPn 位置“1”的 I 流程



## 16.5.5 IICA低电平宽度设定寄存器n (IICWLn)

此寄存器控制串行接口IICA输出的SCLAn引脚信号低电平宽度 ( $t_{LOW}$ ) 和SDAAn引脚信号。

通过8位存储器操作指令设定IICWLn寄存器。

必须在禁止I<sup>2</sup>C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设定IICWLn寄存器。在产生复位信号后, 此寄存器的值变为“FFH”。

有关IICWLn寄存器的设定方法, 请参照“16.6.2通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法”。数据保持时间为IICWLn所设时间的1/4。

位	符号	描述	复位值
7:0	IICWLn	IICA低电平宽度设定寄存器	0xFF

## 16.5.6 IICA高电平宽度设定寄存器n (IICWHn)

此寄存器控制串行接口IICA输出的SCLAn引脚信号高电平宽度和SDAAn引脚信号。通过8位存储器操作指令设定IICWHn寄存器。

必须在禁止I<sup>2</sup>C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设定IICWHn寄存器。在产生复位信号后, 此寄存器的值变为“FFH”。

位	符号	描述	复位值
7:0	IICWHn	IICA高电平宽度设定寄存器	0xFF

注 1: 有关主控方传送时钟的设定方法, 请参照 16.6.2(1); 有关从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法, 请参照 16.6.2(2)。

注 2: n=0。

## 16.5.7 控制IICA引脚端口功能的寄存器

本产品可将IICA的引脚功能复用到多个端口。

将这两个端口对应的端口模式控制寄存器 (PMCxx) 的位和端口模式寄存器 (PMxx) 的位置“0”。

通过设置端口复用功能配置寄存器 (PmnCFG) 可将SCALn引脚和SDAAn引脚分别配置到两个端口。

这两个端口被配置为IICA引脚的复用功能后, 端口的N沟道漏极开路输出 (VDD/EVDD耐压) 模式由设计保证自动打开, 即POMxx寄存器不需要用户设置。

详细的设置方法参见“第2章引脚功能”

## 16.6 I2C总线模式的功能

### 16.6.1 引脚结构

串行时钟引脚 (SCLAn) 和串行数据总线引脚 (SDAAn) 的结构如下。

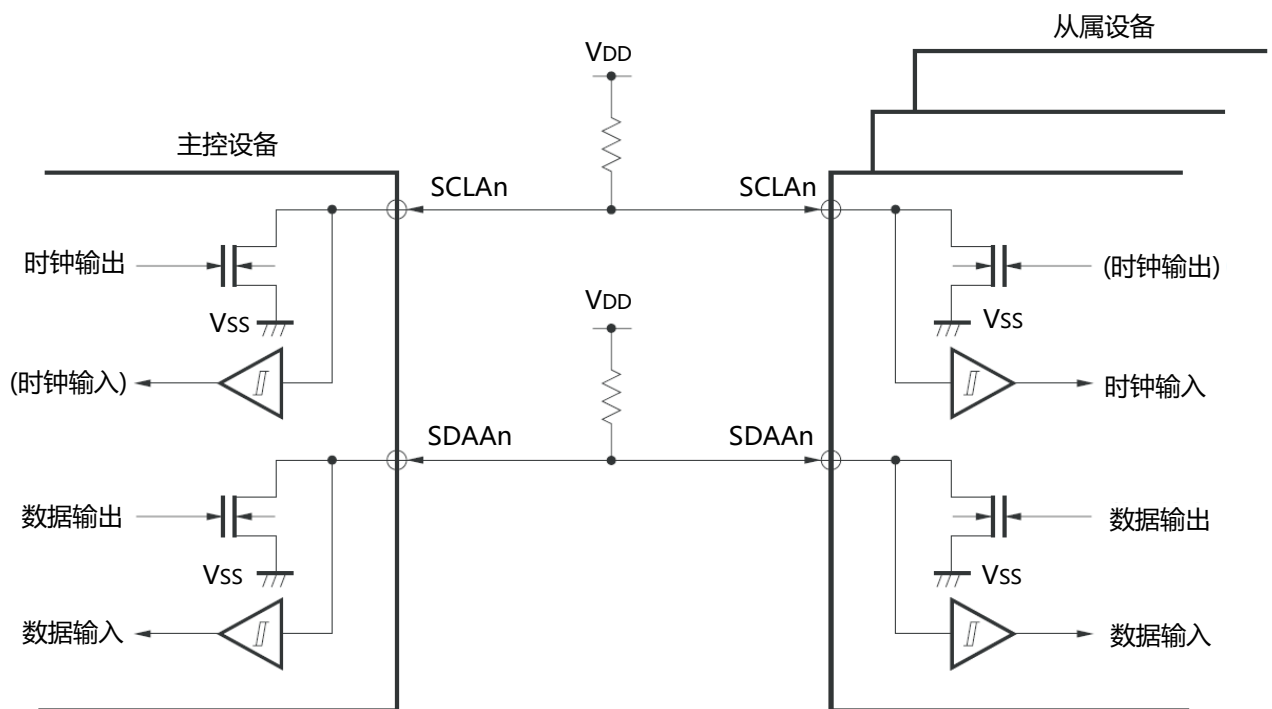
(1) SCLAn: 串行时钟的输入/输出引脚

主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

(2) SDAAn: 串行数据的输入/输出引脚主控设备和从属设备的输出都为 N 沟道漏极开路输出，输入都为施密特输入。

因为串行时钟线和串行数据总线的输出为 N 沟道漏极开路输出，所以需要外接上拉电阻。

图16-4 引脚结构图



注：n=0。

## 16.6.2 通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法

### (1) 主控方传送时钟的设定方法

$$\text{传送时钟} = \frac{f_{MCK}}{IICWL + IICWH + f_{MCK}(t_R + t_F)}$$

此时，IICWLn寄存器和IICWHn寄存器的最佳设定值如下：

(全部设定值的小数部分都舍入)

#### ① 快速模式

$$IICWLn = \frac{0.52}{\text{传送时钟}} \times f_{MCK}$$

$$IICWHn = \left( \frac{0.48}{\text{传送时钟}} - t_R - t_F \right) \times f_{MCK}$$

#### ② 标准模式

$$IICWLn = \frac{0.47}{\text{传送时钟}} \times f_{MCK}$$

$$IICWHn = \left( \frac{0.53}{\text{传送时钟}} - t_R - t_F \right) \times f_{MCK}$$

#### ③ 增强型快速模式

$$IICWLn = \frac{0.50}{\text{传送时钟}} \times f_{MCK}$$

$$IICWHn = \left( \frac{0.50}{\text{传送时钟}} - t_R - t_F \right) \times f_{MCK}$$

### (2) 从属方IICWLn寄存器和IICWHn寄存器的设定方法

(全部设定值的小数部分都舍入)

#### ① 快速模式

$$IICWLn = 1.3\mu s \times f_{MCK}$$

$$IICWHn = (1.2\mu s - t_R - t_F) \times f_{MCK}$$

#### ② 标准模式

$$IICWLn = 4.7\mu s \times f_{MCK}$$

$$IICWHn = (5.3\mu s - t_R - t_F) \times f_{MCK}$$

#### ③ 增强型快速模式

$$IICWLn = 0.50\mu s \times f_{MCK}$$

$$IICWHn = (0.50\mu s - t_R - t_F) \times f_{MCK}$$

注 1：IICA 运行时钟 ( $f_{MCK}$ ) 的最大工作频率为 20MHz(Max.)。只有在  $f_{CLK}$  超过 20MHz 时才必须将 IICA 控制寄存器 n1 (IICCTLn1) 的 bit0 (PRSn) 置“1”。

注 2：在设定传送时钟的情况下，必须注意  $f_{CLK}$  的最小工作频率。串行接口 IICA 的  $f_{CLK}$  最小工作频率取决于运行模式。

快速模式： $f_{CLK} = 3.5\text{MHz}(\text{Min.})$

增强型快速模式： $f_{CLK} = 10\text{MHz}(\text{Min.})$

标准模式： $f_{CLK} = 1\text{MHz}(\text{Min.})$

注 3：因为 SDAAn 信号和 SCLAn 信号的上升时间 ( $t_R$ ) 和下降时间 ( $t_F$ ) 因上拉电阻和布线电容而不同，所以必须各自计算。

注 4：IICWLn：IICA 低电平宽度设定寄存器 n。



IICWHn: IICA 高电平宽度设定寄存器 n。

t<sub>F</sub>: SDAAn 信号和 SCLAn 信号的下降时间。

t<sub>R</sub>: SDAAn 信号和 SCLAn 信号的上升时间。

f<sub>MCK</sub>: IICA 运行时钟频率

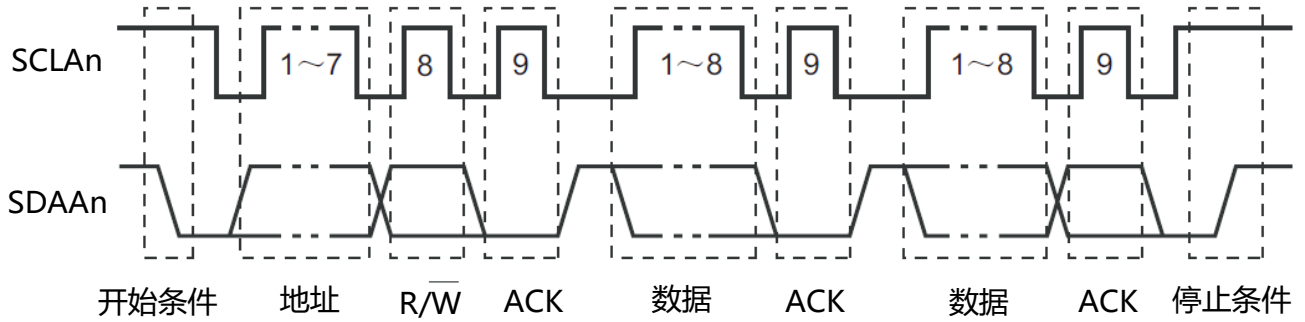
注 5: n=0。

## 16.7 I<sup>2</sup>C总线的定义和控制方法

以下说明I<sup>2</sup>C总线的串行数据通信格式和使用的信号。

I<sup>2</sup>C总线的串行数据总线上生成的“开始条件”、“地址”、“数据”和“停止条件”的各传送时序如下图所示。

图16-5 I<sup>2</sup>C总线的串行数据传送时序



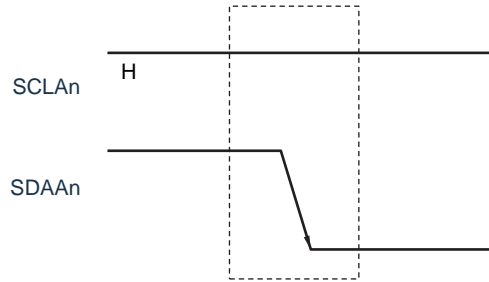
主控设备生成开始条件、从属地址和停止条件。

主控设备和从属设备都能生成应答 (ACK) (在一般情况下, 接收方输出8位数据)。主控设备连续输出串行时钟 (SCLAn)。但是, 从属设备能延长SCLAn引脚的低电平期间并且插入等待。

### 16.7.1 开始条件

在SCLAn引脚为高电平时，如果SDAAn引脚从高电平变为低电平，就生成开始条件。SCLAn引脚和SDAAn引脚的开始条件是在主控设备对从属设备开始串行传送时生成的信号。在用作从属设备时能检测到开始条件。

图16-6 开始条件



在检测到停止条件（SPDn: IICA状态寄存器n (IICSn) 的bit0=1）的状态下，如果将IICA控制寄存器n0 (IICCTLn0) 的bit1 (STTn) 置“1”，就输出开始条件。如果检测到开始条件，就将IICSn寄存器的bit1 (STDn) 置“1”。

注:n=0。

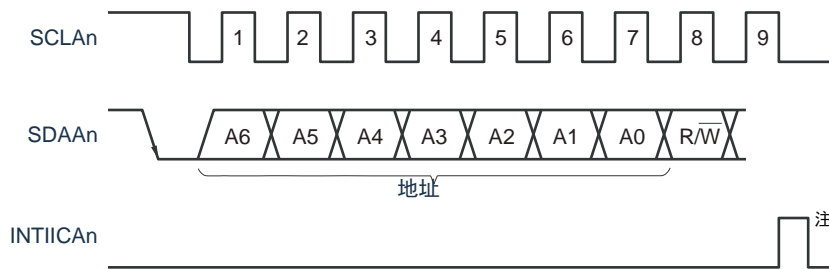
### 16.7.2 地址

开始条件的后续7位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的7位数据。因此，总线上的从属设备需要设定完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查7位数据是否和从属地址寄存器n (SVAn) 的内容相同。此时如果7位数据和SVAn寄存器的值相同，该从属设备就被选中，在 主控设备生成开始条件或者停止条件前，与主控设备进行通信。

图16-7 地址



注：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

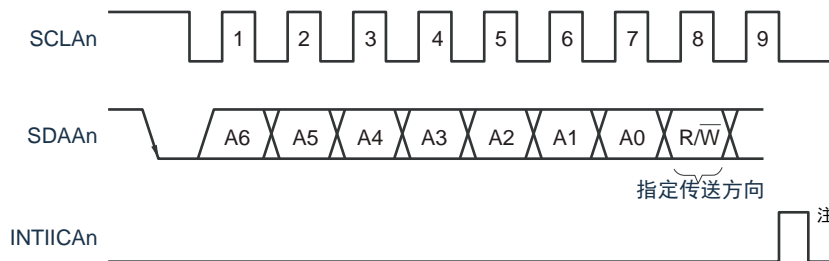
如果将从属地址和16.7.3传送方向的指定中说明的传送方向构成的8位数据写到IICA移位寄存器n (IICAn)，就输出地址。接收到的地址被写到IICAn寄存器。从属地址分配在IICAn寄存器的高7位。

### 16.7.3 传送方向的指定

主控设备在7位地址之后发送1位指定传送方向的数据。

当此传送方向指定位为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定位为“1”时，表示主控设备从从属设备接收数据。

图16-8 传送方向的指定



注1：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

注2：n=0。

### 16.7.4 应答 (ACK)

能通过应答 (ACK) 确认发送方和接收方的串行数据状态。接收方在每次接收到8位数据时返回应答。

通常，发送方在发送8位数据后接收应答。当接收方返回应答时，认为已正常接收，继续处理。能通过IICA状态寄存器n (IICSn) 的bit2 (ACKDn) 确认应答的检测。在主控设备为接收状态下接收到最后的数据时，不返回应答而生成停止条件。在从属设备接收数据后不返回应答时，主控设备输出停止条件或者重新开始条件，中止发送。不返回应答的原因如下：

- ①没有正常接收。
- ②已结束最后数据的接收。
- ③不存在地址指定的接收方。

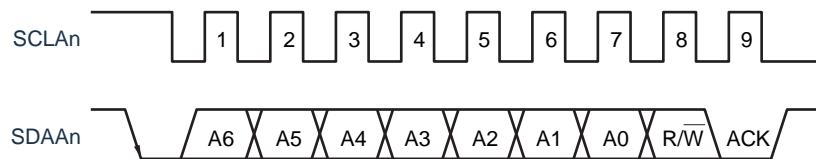
接收方在第9个时钟将SDAAn线置为低电平，生成应答 (正常接收)。

通过将IICA控制寄存器n0 (IICCTLn0) 的bit2 (ACKEn) 置“1”，变为能自动生成应答的状态。通过7位地址信息后续的第8位数据设定IICSn寄存器的bit3 (TRCn)。在接收 (TRCn=0) 的情况下，通常必须将ACKEn位置“1”。

在从属接收运行过程中 (TRCn=0) 不能接收数据或者不需要下一个数据时，必须将ACKEn位清“0”，通知主控方不能接收数据。

在接收运行过程中 (TRCn=0) 不需要下一个数据时，为了不生成应答，必须将ACKEn位清“0”，通知从属发送方数据的结束 (停止发送)。

图16-9 应答



当接收到本地站的地址时，与ACKEn位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答 (NACK)。

在接收到扩展码时，通过事先将ACKEn位置“1”，生成应答。接收数据时的应答生成方法因等待时序的设定而不同，如下所示。

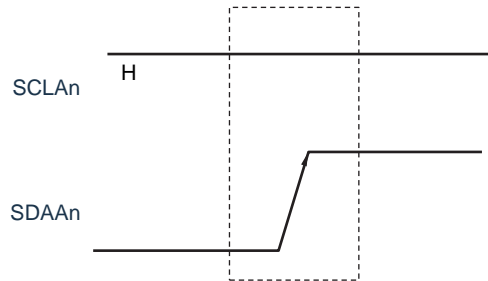
- (1) 当选择 8 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) =0)：通过在解除等待前将 ACKEn 位置“1”，与 SCLAn 引脚的第 8 个时钟下降沿同步生成应答。
- (2) 当选择 9 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) =1)：通过事先将 ACKEn 位置“1”，生成应答。

注：n=0。

### 16.7.5 停止条件

在SCLAn引脚为高电平时，如果SDAAn引脚从低电平变为高电平，就生成停止条件。停止条件是在主控设备结束对从属设备的串行传送时生成的信号。在用作从属设备时，能检测到停止条件。

图16-10 停止条件



如果将IICA控制寄存器n0 (IICCTLn0) 的bit0 (SPTn) 置“1”，就生成停止条件。如果检测到停止条件，就将IICA状态寄存器n (IICSn) 的bit0 (SPDn) 置“1”，并且在IICCTLn0寄存器的bit4 (SPIEn) 为“1”时产生INTIICAn。

注：n=0。

### 16.7.6 等待

通过等待来通知对方主控设备或者从属设备正在准备数据的发送/接收（等待状态）。

通过将SCLAn引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都被解除，就能开始下一次传送。

图16-11 等待(1/2)

(1) 主控设备为9个时钟等待，从属设备为8个时钟等待的情况

(主控设备：发送，从属设备：接收，ACKEn=1)

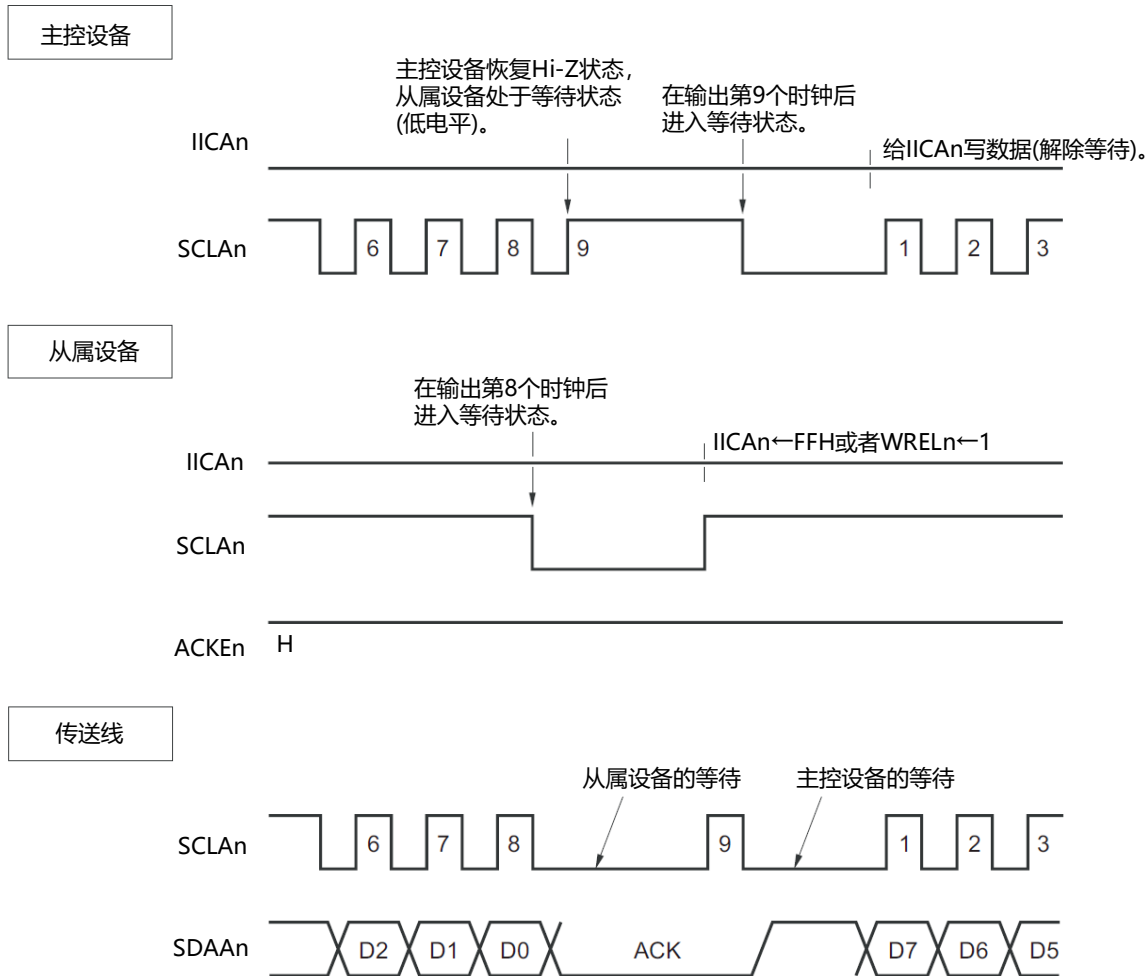
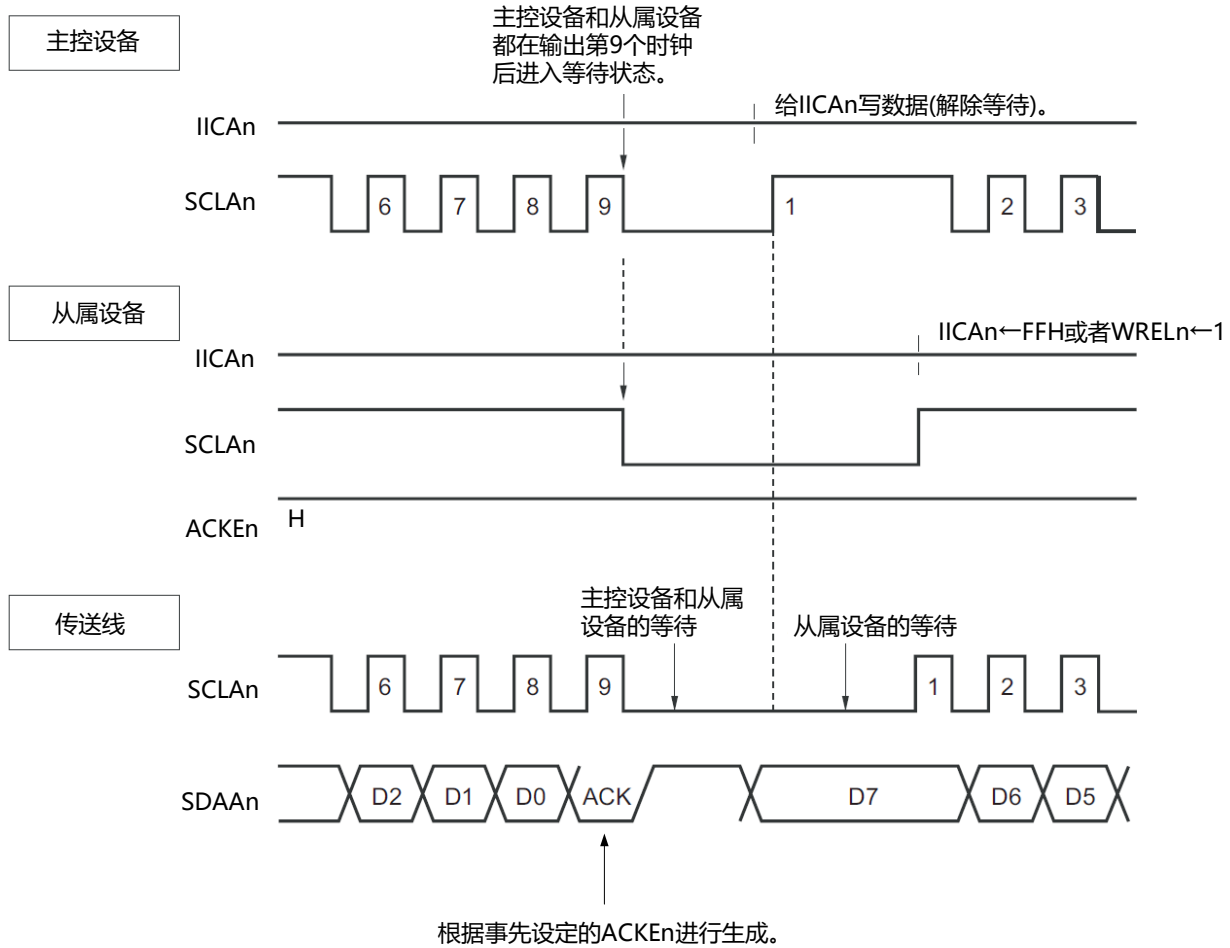


图16-11 等待(2/2)

(2) 主控设备和从属设备都为9个时钟等待的情况  
 (主控设备: 发送, 从属设备: 接收, ACKEn=1)



注1: ACKEn: IICA控制寄存器n0 (IICCTLn0) 的bit2。

注2: WRELn: IICA控制寄存器n0 (IICCTLn0) 的bit5。

通过设定IICA控制寄存器n0 (IICCTLn0) 的bit3 (WTIMn) 自动产生等待状态。通常, 在接收方, 如果IICCTLn0寄存器的bit5 (WRELn) 为“1”或者给IICA移位寄存器n (IICAn) 写“FFH”, 就解除等待; 在发送方, 如果给IICAn寄存器写数据, 就解除等待。主控设备还能通过以下方法解除等待:

将IICCTLn0寄存器的bit1 (STTn) 置“1”。

将IICCTLn0寄存器的bit0 (SPTn) 置“1”。

注: n=0。



## 16.7.7 等待的解除方法

在一般情况下，I<sup>2</sup>C能通过以下的处理来解除等待。

- (1) 给 IICA 移位寄存器 n (IICAn) 写数据。
- (2) 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置位 (解除等待)。
- (3) 将 IICCTLn0 寄存器的 bit1 (STTn) 置位 (生成开始条件)<sup>注</sup>
- (4) 将 IICCTLn0 寄存器的 bit0 (SPTn) 置位 (生成停止条件)<sup>注</sup>

注：只限于主控设备。

如果执行了这些等待的解除处理，I<sup>2</sup>C就解除等待，重新开始通信。要在解除等待后发送数据（包括地址）时，必须给IICAn寄存器写数据。

要在解除等待后接收数据或者结束发送数据时，必须将IICCTLn0寄存器的bit5 (WRELn) 置“1”。要在解除等待后生成重新开始条件时，必须将IICCTLn0寄存器的bit1 (STTn) 置“1”。要在解除等待后生成停止条件时，必须将IICCTLn0寄存器的bit0 (SPTn) 置“1”。对于一次等待只能执行一次解除处理。

例如，如果在通过将WRELn位置“1”来解除等待后给IICAn寄存器写数据，SDAAn线的变化时序与IICAn寄存器的写时序就可能发生冲突，导致将错误的值输出到SDAAn线。除了这些处理以外，在中途中止通信的情况下，如果将IICEn位清“0”，就停止通信，因此能解除等待。在I<sup>2</sup>C总线状态因噪声而被死锁的情况下，如果将IICCTLn0寄存器的bit6 (LRELn) 置“1”，就退出通信，因此能解除等待。

注1：如果在WUPn位为“1”时执行等待的解除处理，就不解除等待。

注2：n=0。

## 16.7.8 中断请求 (INTIICAn) 的产生时序和等待控制

通过设定IICA控制寄存器n0 (IICCTLn0) 的bit3 (WTIMn) , 在表16-3所示的时序产生INTIICAn并且进行等待控制。

表16-3 INTIICAn的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 <sup>注1、2</sup>	8 <sup>注2</sup>	8 <sup>注2</sup>	9	8	8
1	9 <sup>注1、2</sup>	9 <sup>注2</sup>	9 <sup>注2</sup>	9	9	9

注 1: 只有在接收的地址和从属地址寄存器 n (SVAn) 的设定地址相同时, 从属设备才在第 9 个时钟的下降沿产生 INTIICAn 信号并且进入等待状态。此时, 与 IICCTLn0 寄存器的 bit2 (ACKEn) 的设定无关, 生成应答。接收到扩展码的从属设备在第 8 个时钟的下降沿产生 INTIICAn。如果在重新开始后地址不同, 就在第 9 个时钟的下降沿产生 INTIICAn, 但是不进入等待状态。

注 2: 如果接收的地址和从属地址寄存器 n (SVAn) 的内容不同并且未接收到扩展码, 就不产生 INTIICAn 并且也不进入等待状态。

注 3: 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

### (1) 地址的发送和接收

- ① 从属运行: 与 WTIMn 位无关, 根据上述注 1 和注 2 的条件决定中断和等待的时序。
- ② 主控运行: 与 WTIMn 位无关, 在第 9 个时钟的下降沿产生中断和等待的时序。

### (2) 数据接收

主控运行/从属运行: 通过 WTIMn 位决定中断和等待的时序。

### (3) 数据发送

主控运行/从属运行: 通过 WTIMn 位决定中断和等待的时序。

注: n=0。

#### (4) 等待的解除方法

等待的解除方法有以下 4 种：

- ① 给 IICA 移位寄存器 n (IICAn) 写数据。
- ② 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置位 (解除等待)。
- ③ 将 IICCTLn0 寄存器的 bit1 (STTn) 置位 (生成开始条件)<sup>注</sup>。
- ④ 将 IICCTLn0 寄存器的 bit0 (SPTn) 置位 (生成停止条件)<sup>注</sup>。

注 1：只限于主控设备。

注 2：当选择 8 个时钟的等待 (WTIMn=0) 时，需要在解除等待前决定是否生成应答。

#### (5) 停止条件的检测

如果检测到停止条件，就产生 INTIICAn (只限于 SPIEn=1 的情况)。

## 16.7.9 地址匹配的检测方法

在I<sup>2</sup>C总线模式中，主控设备能通过发送从属地址来选择特定的从属设备。能通过硬件自动检测地址匹配。当主控设备发送的从属地址和从属地址寄存器n (SVAn) 的设定地址相同或者只接收到扩展码时，产生INTIICAn中断请求。

### 16.7.10 错误的检测

在I<sup>2</sup>C总线模式中，因为发送过程中的串行数据总线 (SDAAn) 的状态被取到发送器件的IICA移位寄存器n (IICAn)，所以能通过将开始发送前和发送结束后的IICA数据进行比较来检测发送错误。此时，如果2个数据不同，就判断为发生了发送错误。

注：n=0。

### 16.7.11 扩展码

(1)当接收地址的高4位为“0000”或者“1111”时，作为接收到扩展码，将扩展码接收标志 (EXCn) 置“1”，并且在第8个时钟的下降沿产生中断请求 (INTIICAn)。

不影响保存在从属地址寄存器n (SVAn) 的本地站地址。

(2)当SVAn寄存器的设定值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求 (INTIICAn)。

- ① 高4位数据相同：EXCn=1
- ② 7位数据相同：COIn=1

注1：EXCn：IICA状态寄存器n (IICSn) 的bit5。

注2：COIn：IICA状态寄存器n (IICSn) 的bit4。

(3)中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。如果在从属运行时接收到扩展码，即使地址不同也在参加通信。例如，在接收到扩展码后不想作为从属设备运行时，必须将IICA控制寄存器n0 (IICCTLn0) 的bit6 (LRELn) 置“1”，进入下一次通信的待机状态。

表16-4 主要扩展码的位定义

从属地址	R/W位	说明
0000000	0	全呼地址
11110xx	0	10位从属地址的指定 (地址认证时)
11110xx	1	10位从属地址的指定 (在地址相同后发行读命令时)

注1：有关上述以外的扩展码，请参照NXP公司发行的I2C总线规格书。

注2：n=0。

### 16.7.12 仲裁

当多个主控设备同时生成开始条件时（在STDn位变为“1”前将STTn位置“1”的情况），边调整时钟边进行主控设备的通信，直到数据不同为止。此运行称为仲裁。

在仲裁失败时，仲裁失败的主控设备将IICA状态寄存器n（IICSn）的仲裁失败标志（ALDn）置“1”，并且将SCLAn线和SDAAn线都置为高阻抗状态，释放总线。

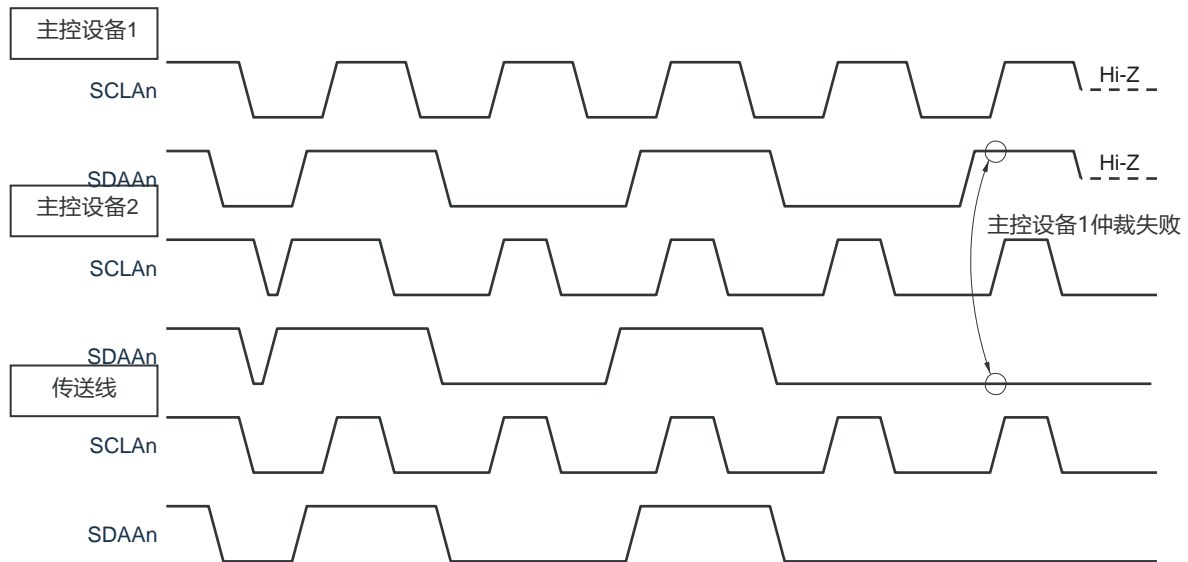
在发生下一次中断请求时（例如：在第8或者第9个时钟检测到停止条件），用软件通过ALDn位为“1”来检测仲裁的失败。

有关中断请求的产生时序，请参照“16.7.8中断请求（INTIICAn）的产生时序和等待控制”。

注1：STDn：IICA状态寄存器n（IICSn）的bit1。

注2：STTn：IICA控制寄存器n0（IICCTLn0）的bit1。

图16-12 仲裁时序例子



注：n=0。

表16-5 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第8或者第9个时钟的下降沿 <sup>注1</sup>
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时 (SPIEn=1) <sup>注2</sup>
想要生成重新开始条件, 但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 <sup>注1</sup>
想要生成重新开始条件, 但是检测到停止条件。	在生成停止条件时 (SPIEn=1) <sup>注2</sup>
想要生成停止条件, 但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 <sup>注1</sup>
想要生成重新开始条件, 但是SCLAn为低电平。	

注1: 当WTIMn位 (IICA控制寄存器n0 (IICCTLn0) 的bit3) 为“1”时, 在第9个时钟的下降沿产生中断请求; 当WTIMn位为“0”并且接收到扩展码的从属地址时, 在第8个时钟的下降沿产生中断请求。

注2: 当有可能发生仲裁时, 必须在主控运行时将SPIEn位置“1”。

注3: SPIEn: IICA控制寄存器n0 (IICCTLn0) 的bit4。

注4: n=0。

### 16.7.13 唤醒功能

这是I<sup>2</sup>C的从属功能，是在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）的功能。在地址不同的情况下不产生不需要的INTIICAn信号，能提高处理效率。如果检测到开始条件，就进入唤醒待机状态。因为主控设备（已经生成开始条件的情况）也有可能因仲裁失败而变为从属设备，所以在发送地址的同时进入唤醒待机状态。

要在深度睡眠模式中使用唤醒功能时，必须将WUPn位置“1”。与运行时钟无关而能接收地址。即使在这种情况下，也在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）。在产生此中断后，通过指令将WUPn位清“0”，返回到通常运行。

将WUPn位置“1”时的流程如图16-13所示，通过地址匹配将WUPn位置“0”时的流程如图16-14所示。

图16-13 将WUPn位置“1”时的流程

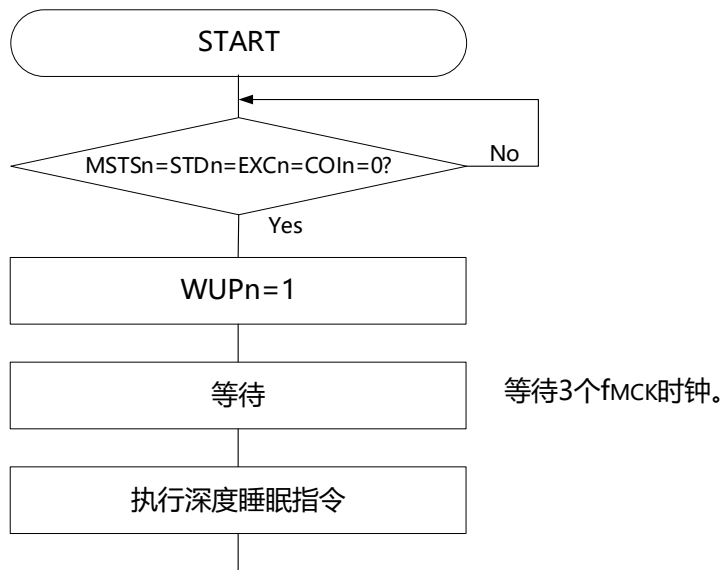
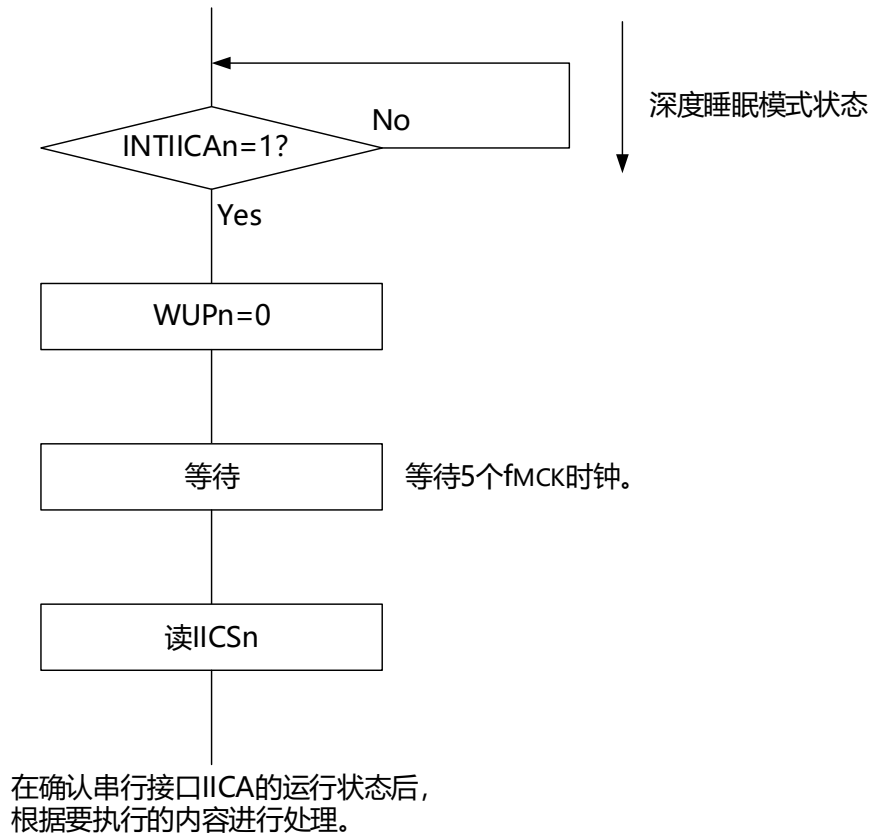


图16-14 通过地址匹配将WUPn位置“0”时的流程（包括接收扩展码）



除了串行接口IICA产生的中断请求（INTIICAn）以外，必须通过以下的流程解除深度睡眠模式。

(1) 下一次IIC通信为主控设备运行的情况：图16-14的流程

(2) 下一次IIC通信为从属设备运行的情况

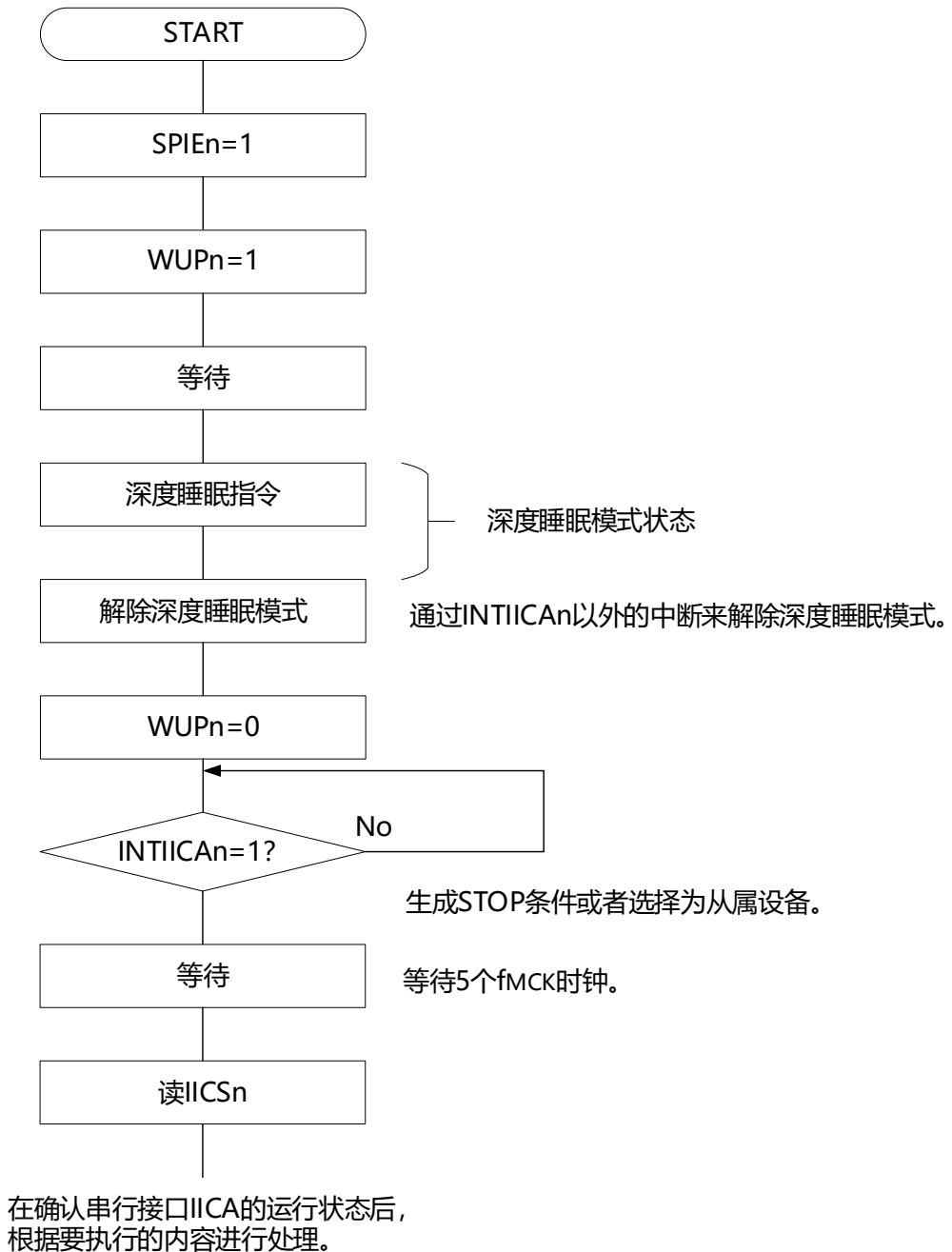
通过INTIICAn中断返回的情况：和图16-15的流程相同。

通过INTIICAn中断以外的中断返回的情况：必须在产生INTIICAn中断前保持WUPn位为“1”状态继续运行。

注：n=0。



图16-15 在通过INTIICAn以外的中断来解除深度睡眠模式后作为主控设备运行的情况



注：n=0。

## 16.7.14 通信预约

(1) 允许通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =0)

要在不加入总线的状态下进行下一次主控通信时, 能通过通信预约在释放总线时发送开始条件。此时的不加入总线包括以下 2 种状态:

- ① 在仲裁结果既不是主控设备也不是从属设备时
- ② 在接收到扩展码后不作为从属设备运行时 (不返回应答而将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit6 (LRELn) 置“1”, 退出通信后释放了总线)

如果在不加入总线的状态下将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”, 就在释放总线后 (检测到停止条件) 自动生成开始条件, 进入等待状态。

将 IICCTLn0 寄存器的 bit4 (SPIEn) 置“1”, 在通过产生的中断请求信号 (INTIICAn) 检测到总线的释放 (检测到停止条件) 后, 如果给 IICA 移位寄存器 n (IICAn) 写地址, 就自动作为主控设备开始通信。在检测到停止条件前, 给 IICAn 寄存器写的的数据无效。

当将 STTn 位置“1”时, 根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- ① 总线处于释放状态时生成开始条件
- ② 总线未处于释放状态 (待机状态) 时通信预约

在将 STTn 位置“1”并且经过等待时间后, 通过 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 确认是否作为通信预约运行。

必须通过软件确保以下计算式计算的等待时间:

$$\text{从将 STTn 位置“1”到确认 MSTSn 标志为止的等待时间:} \\ (\text{IICWLn 的设定值} + \text{IICWHn 的设定值} + 4) / f_{\text{MCK}} + t_{\text{F}} \times 2$$

注1: IICWLn: IICA 低电平宽度设定寄存器 n。

注2: IICWHn: IICA 高电平宽度设定寄存器 n。

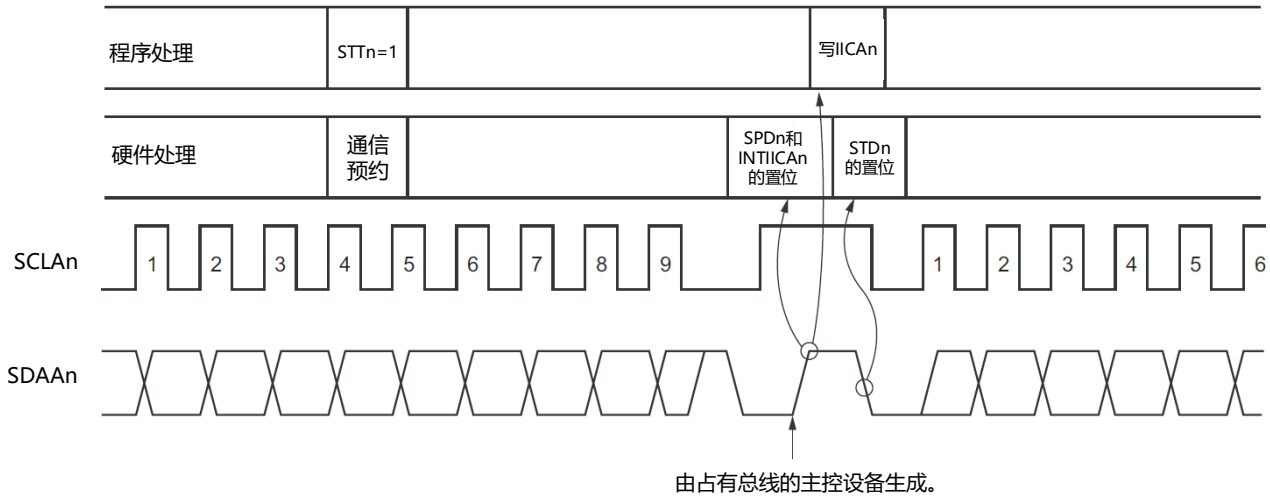
注3:  $t_{\text{F}}$ : SDAAn 信号和 SCLAn 信号的下降时间。

注4:  $f_{\text{MCK}}$ : IICA 运行时钟频率。

注5: n=0

通信预约的时序如下图所示。

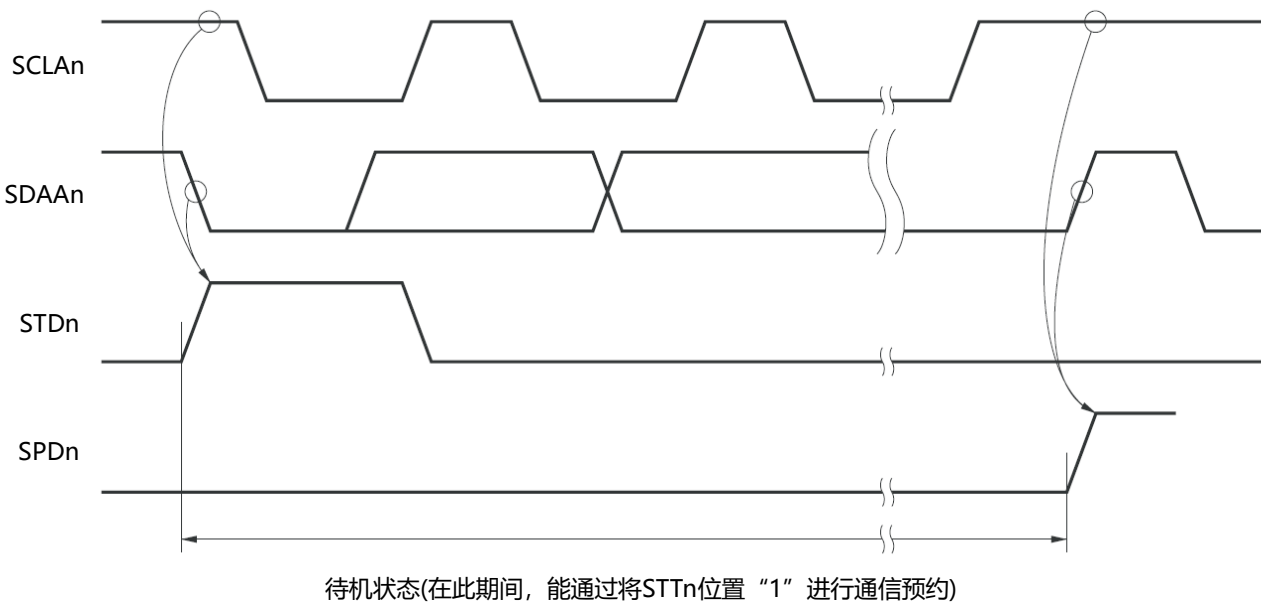
图16-16 通信预约的时序



- 注1: IICAn: IICA移位寄存器n。
- 注2: STTn: IICA控制寄存器n0 (IICCTLn0) 的bit1。
- 注3: STDn: IICA状态寄存器n (IICSn) 的bit1。
- 注4: SPDn: IICA状态寄存器n (IICSn) 的bit0。

通过图16-17所示的时序接受通信预约。在IICA状态寄存器n (IICSn) 的bit1 (STDn) 变为“1”后并且在检测到停止条件前, 将IICA控制寄存器n0 (IICCTLn0) 的bit1 (STTn) 置“1”进行通信预约。

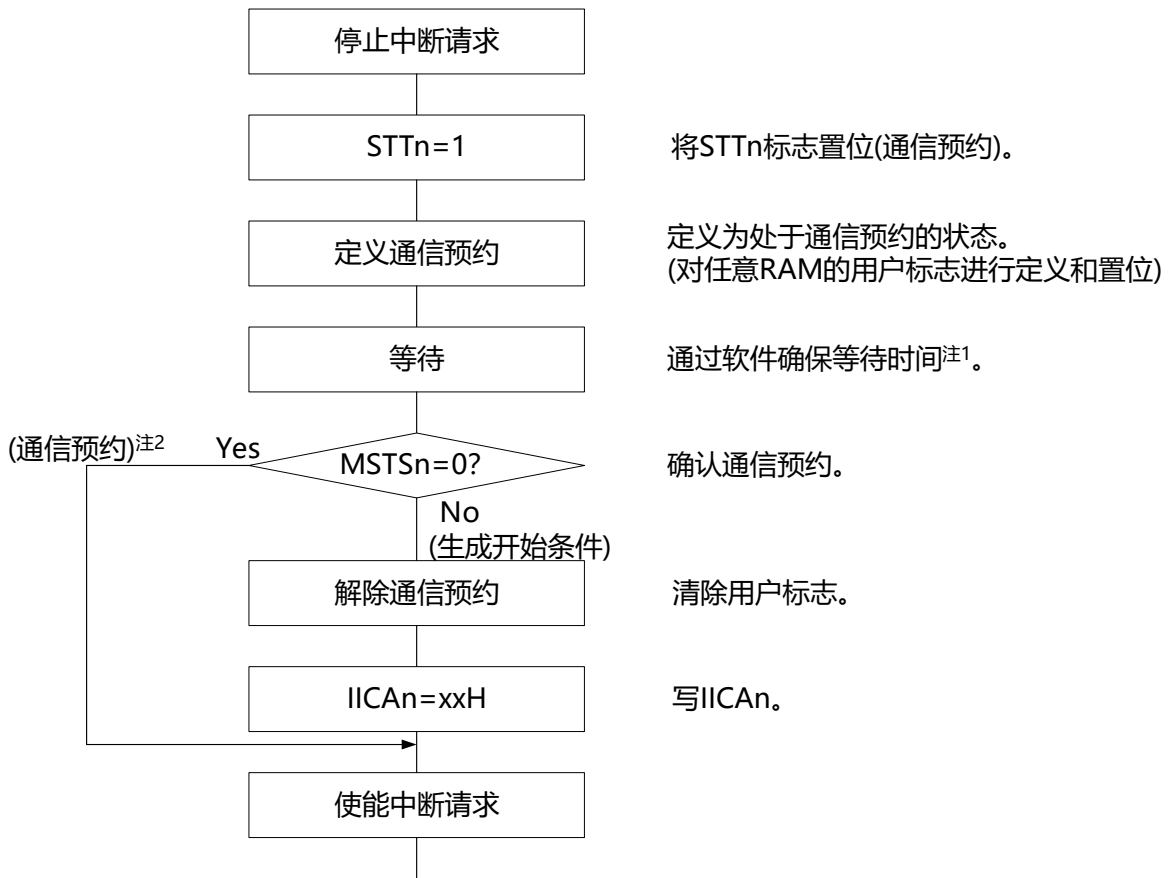
图16-17 通信预约的接受时序



注: n=0

通信预约的步骤如图16-18所示。

图16-18 通信预约的步骤



注1：等待时间如下： $(IICWLn$ 的设定值 $+IICWHn$ 的设定值 $+4)/f_{MCK}+t_F$ 。

注2：在通信预约运行时，通过停止条件中断请求来写IICA移位寄存器n（IICAn）。

注3：STTn：IICA控制寄存器n0（IICCTLn0）的bit1。

注4：MSTSn：IICA状态寄存器n（IICSn）的bit7。

注5：IICAn：IICA移位寄存器n。

注6：IICWLn：IICA低电平宽度设定寄存器n。

注7：IICWHn：IICA高电平宽度设定寄存器n。

注8： $t_F$ ：SDAAn信号和SCLAn信号的下降时间。

注9： $f_{MCK}$ ：IICA运行时钟频率。

注10： $n=0$ 。

(2) 禁止通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =1)

在总线通信过程中，如果不参加此通信的状态下将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”，就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下 2 种状态：

- ① 在仲裁结果既不是主控设备也不是从属设备时
- ② 在接收到扩展码后不作为从属设备运行时 (不返回应答而将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”，退出通信后释放了总线)

能通过 STCFn (IICFn 寄存器的 bit7) 来确认是生成了开始条件还是拒绝了请求。因为从 STTn 位为“1”到将 STCFn 位置“1”为止需要 5 个  $F_{MCK}$  时钟的时间，所以必须通过软件确保此时间。

注：n=0。

## 16.7.15 其他注意事项

### (1) STCENn 位为“0”的情况

在刚允许 I<sup>2</sup>C 运行 (IICEn=1) 后, 与实际的总线状态无关而视为通信状态 (IICBSYn=1)。要在没有检测到停止条件的状态下进行主控通信时, 必须先生成停止条件, 在释放总线后进行主控通信。对于多主控, 在总线未释放 (未检测到停止条件) 的状态下不能进行主控通信。按照以下顺序生成停止条件:

- ① 设定 IICA 控制寄存器 n1 (IICCTLn1)。
- ② 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) 置“1”。
- ③ 将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

### (2) STCENn 位为“1”的情况

在刚允许 I<sup>2</sup>C 运行 (IICEn=1) 后, 与实际的总线状态无关而视为释放状态 (IICBSYn=0)。因此在生成第 1 个开始条件 (STTn=1) 时, 为了不破坏其他的通信, 需要确认总线已被释放。

### (3) 正在和其他设备进行 I<sup>2</sup>C 通信的情况

在 SDAAn 引脚为低电平并且 SCLAn 引脚为高电平时, 如果允许 I<sup>2</sup>C 运行而且中途参加通信, I<sup>2</sup>C 的宏就视为 SDAAn 引脚从高电平变为低电平 (检测到开始条件)。如果此时总线上的值是能识别为扩展码的值, 就返回应答而妨碍和其他设备的 I<sup>2</sup>C 通信。为了避免这种情况, 必须按照以下顺序启动 I<sup>2</sup>C:

- ① 将 IICCTLn0 寄存器的 bit4 (SPIEn) 清“0”, 禁止在检测到停止条件时产生中断请求信号 (INTIICAn)。
- ② 将 IICCTLn0 寄存器的 bit7 (IICEn) 置“1”, 允许 I<sup>2</sup>C 运行。
- ③ 等待检测到开始条件。
- ④ 在返回应答前 (将 IICEn 位置“1”后的 4~72 个 f<sub>MCK</sub> 时钟内) 将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”, 强制将检测置为无效。

(4) 在将 STTn 位和 SPTn 位 (IICCTLn0 寄存器的 bit1 和 bit0) 置位后, 禁止清“0”前的再置位。

(5) 如果进行了通信预约, 就必须将 SPIEn 位 (IICCTLn0 寄存器的 bit4) 置“1”, 在检测到停止条件时产生中断请求。在产生中断请求后, 通过给 IICA 移位寄存器 n (IICAn) 写通信数据, 开始传送。如果在检测到停止条件时不发生中断, 就因在开始通信时不产生中断请求而停止在等待状态。但是, 当通过软件检测 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 时, 不需要将 SPIEn 位置“1”。

注: n=0。

## 16.7.16 通信运行

在此通过流程图表示以下3个运行步骤。

### (1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设定”和“通信处理”。在启动时执行“初始设定”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

### (2) 多主控系统的主控运行

在 I<sup>2</sup>C 总线的多主控系统中，只根据 I<sup>2</sup>C 总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1 帧）为高电平，就将总线作为释放状态而参加通信。此流程大体分为“初始设定”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设定”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

### (3) 从属运行

用作 I<sup>2</sup>C 总线从属设备的例子如下所示。

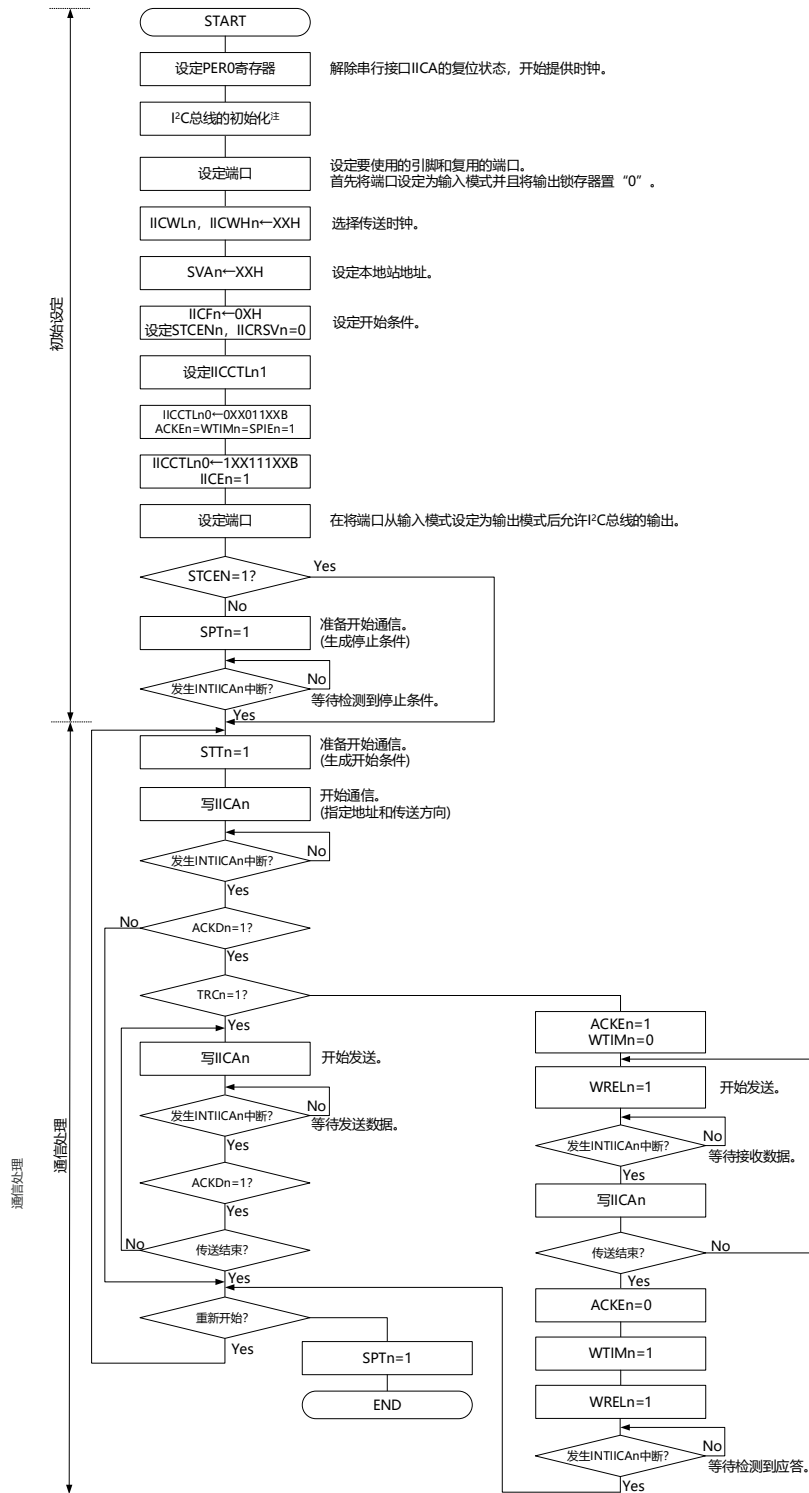
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICAn 中断的发生。如果发生 INTIICAn 中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

注：n=0。

(1) 单主控系统的主控运行

图16-19 单主控系统的主控运行



注1：必须根据通信中的产品的规格，释放I<sup>2</sup>C总线（SCLAn引脚和SDAAn引脚为高电平）。例如，如果EEPROM处于给SDAAn引脚输出低电平的状态，就必须将SCLAn引脚设定为输出端口，并且在SDAAn引脚固定为高电平前从输出端口输出时钟脉冲。

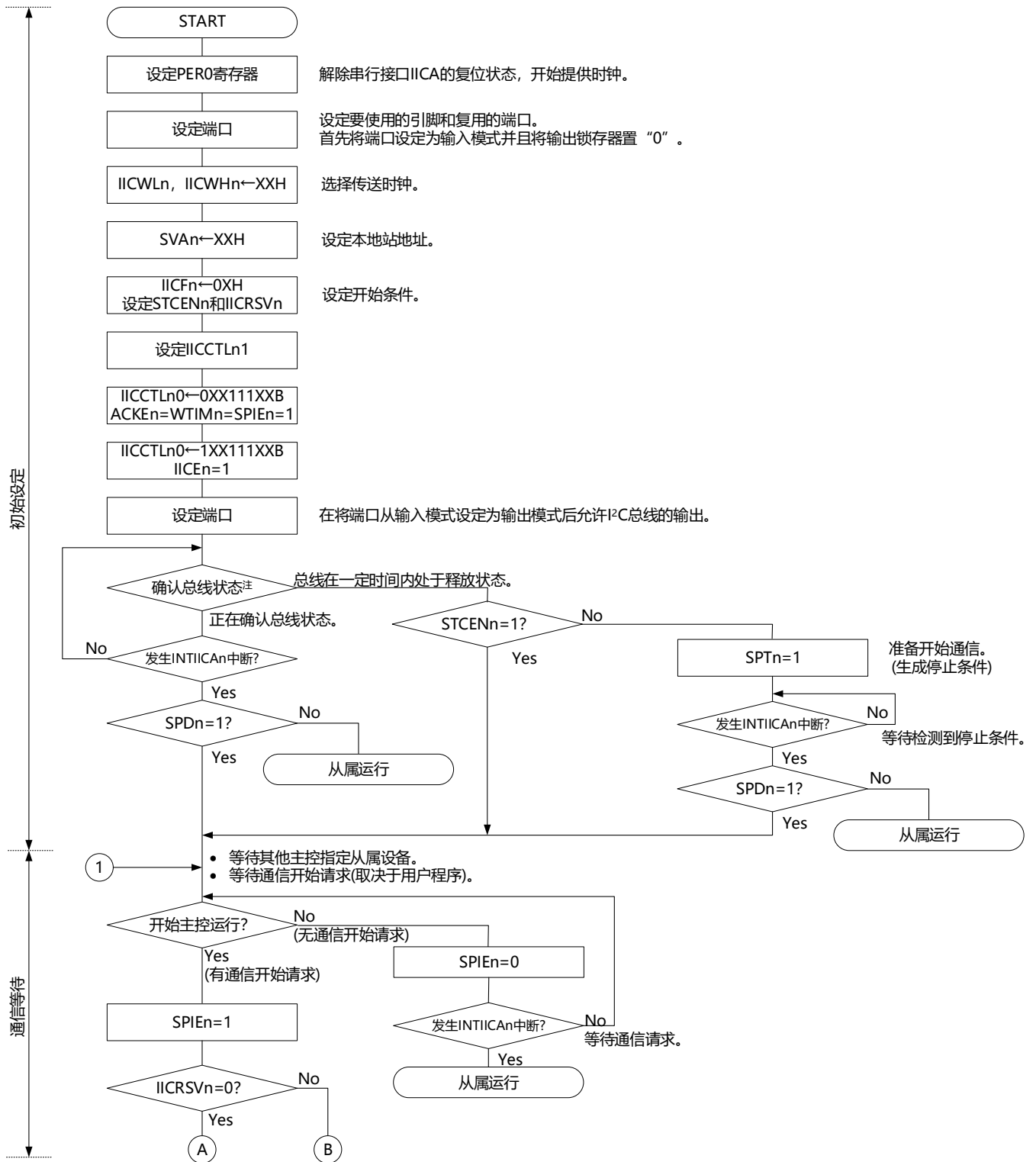
注2：发送和接收的格式必须符合通信中的产品的规格。

注3：n=0。



(2) 多主控系统的主控运行

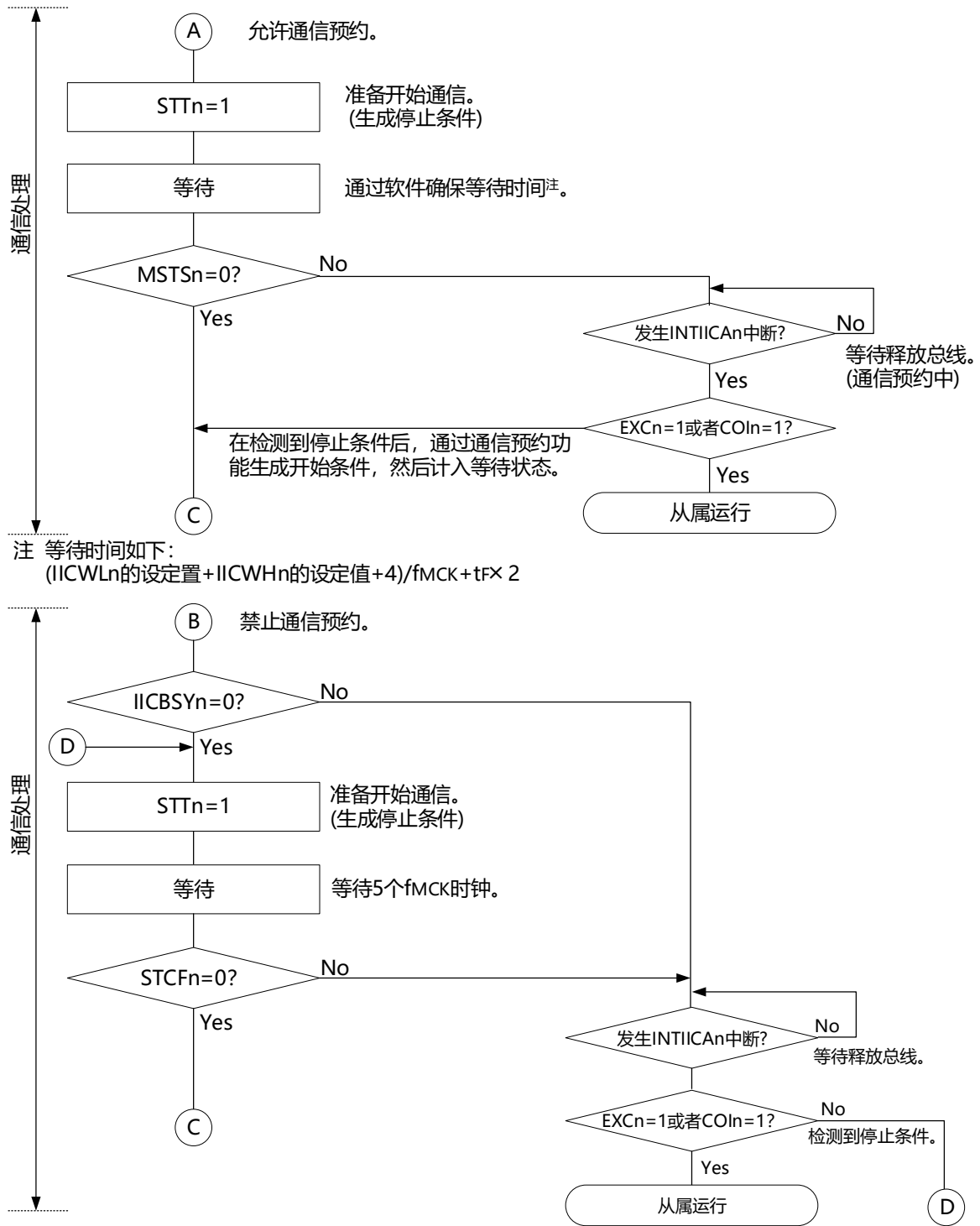
图16-20 多主控系统的主控运行(1/3)



注1: 必须确认总线在一定时间内(例如1帧)处于释放状态(CLDn位=1、DADn位=1)。当SDAAn引脚固定为低电平时, 必须根据通信中的产品的规格, 判断是否释放I2C总线(SCLAn引脚和SDAAn引脚为高电平)。

注2: n=0。

图16-20 多主控系统的主控运行(2/3)



注1: IICWLn: IICA低电平宽度设定寄存器n。

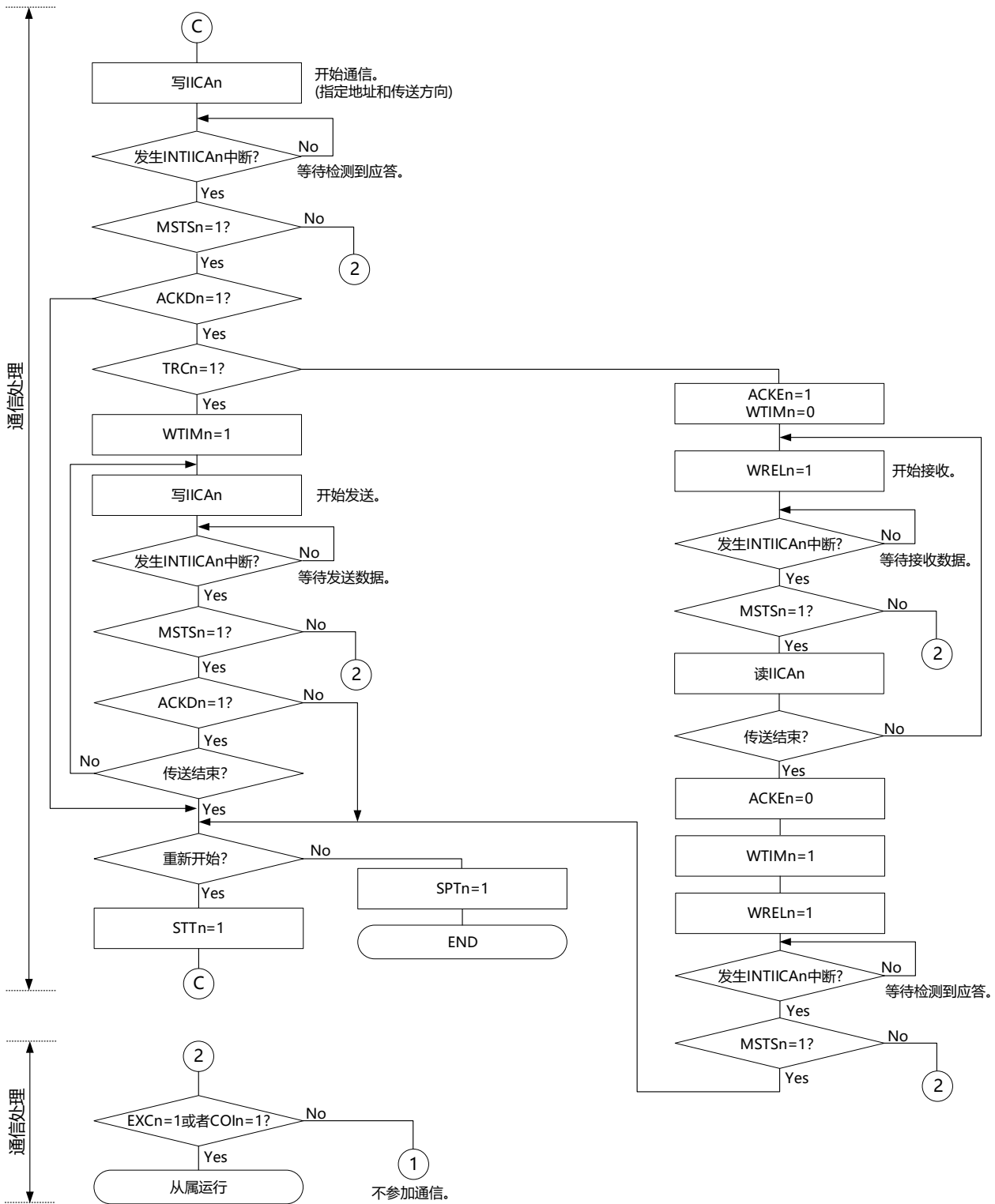
注2: IICWHn: IICA高电平宽度设定寄存器n。

注3: t<sub>F</sub>: SDAAn信号和SCLAn信号的下降时间。

注4: f<sub>MCK</sub>: IICA运行时钟频率。

注5: n=0。

图16-20 多主控系统的主控运行(3/3)



注1：传送和接收的格式必须符合通信中的产品的规格。

注2：在多主控系统中用作主控设备的情况下，必须在每次发生INTIICAn中断时读MSTSn位，确认仲裁结果。

注3：在多主控系统中用作从属设备的情况下，必须在每次发生INTIICAn中断时通过IICA状态寄存器n (IICSn) 和IICA标志寄存器n (IICFn) 确认状态，决定以后的处理。

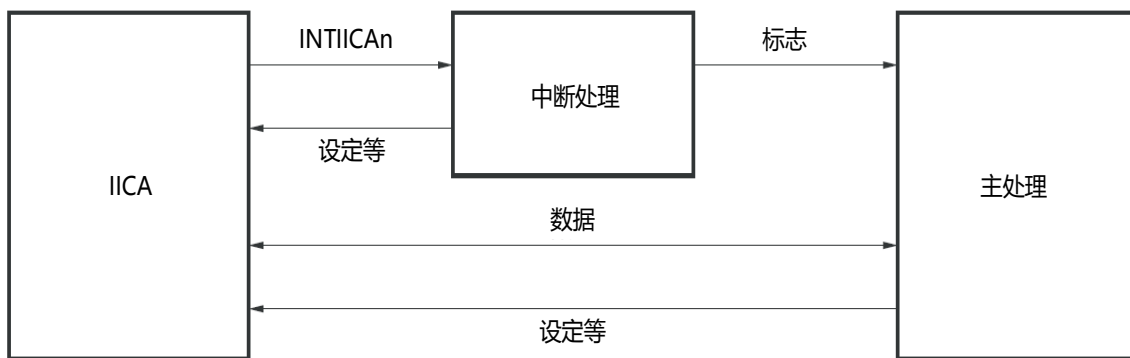
注4：n=0。

(3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过INTIICAn中断进行处理（需要对通信中的停止条件检测等的运行状态进行很大的变更处理）。

在此说明中，假设数据通信不支持扩展码，INTIICAn中断处理只进行状态转移处理并且实际的数据通信由主处理部进行。



因此，准备以下3个标志并且代替INTIICAn将标志传递给主处理部，进行数据通信处理。

① 通信模式标志

此标志表示以下 2 种通信状态：

清除模式：不在进行数据通信的状态

通信模式：正在进行数据通信的状态（有效地址的检测~停止条件的检测，未检测到主控设备的应答，地址不同）

② 就绪标志

此标志表示能进行数据通信。在通常的数据通信中，和 INTIICAn 中断相同，由中断处理部置位而由主处理部清除。在开始通信时，由中断处理部清除标志。但是，在发送第 1 个数据时，中断处理部不将就绪标志置位，因此在不清除标志的状态下发送第 1 个数据（地址匹配被解释为下一次数据请求）。

③ 通信方向标志

此标志表示通信方向，和 TRCn 位的值相同。

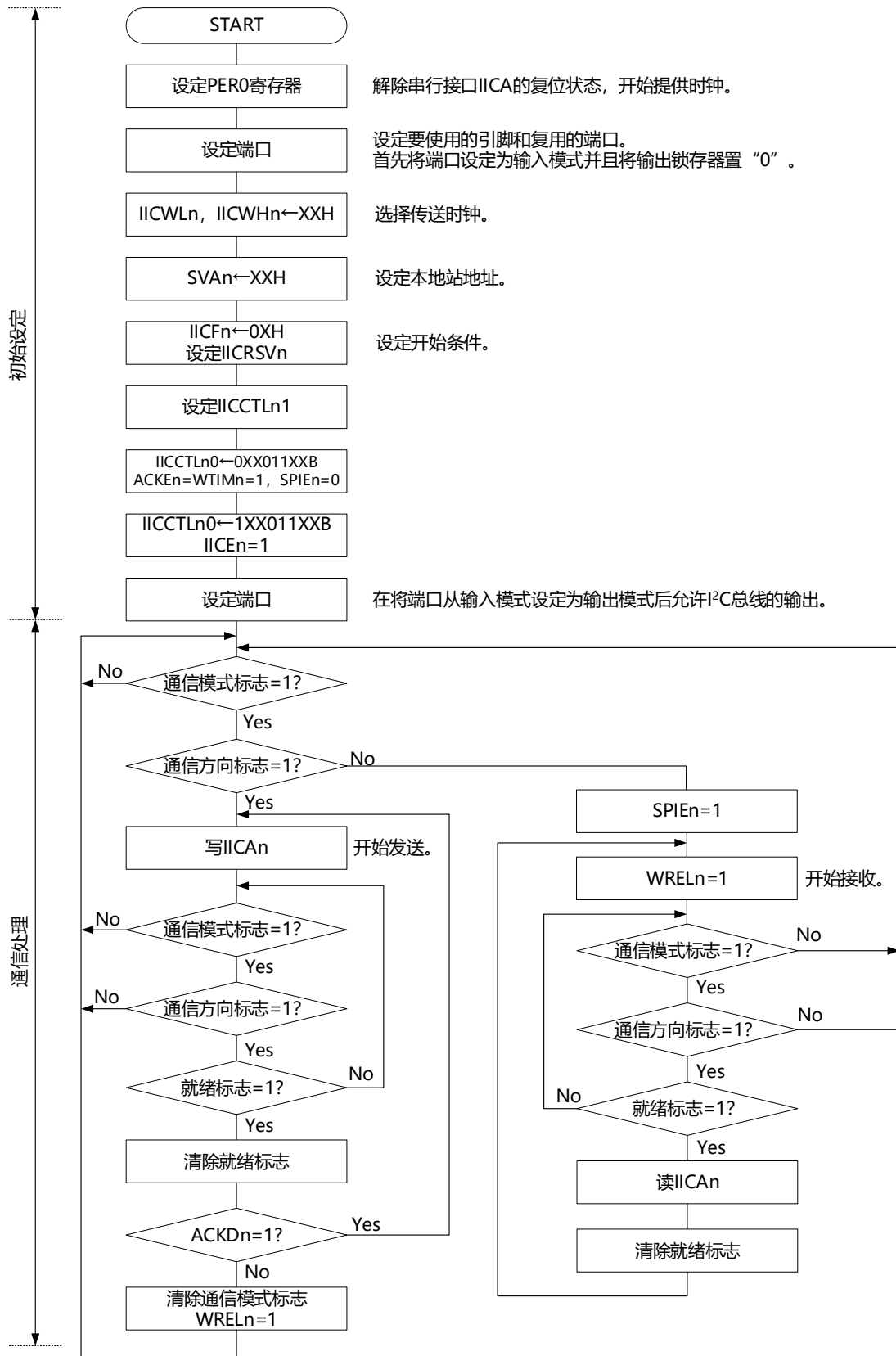
注：n=0。

从属运行的主处理部的运行如下所示。

启动串行接口IICA，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信（因为通过中断进行停止条件和开始条件的处理，所以在此通过标志确认状态）。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图16-21 从属运行步骤(1)



注1：传送和接收的格式必须符合通信中的产品的规格。

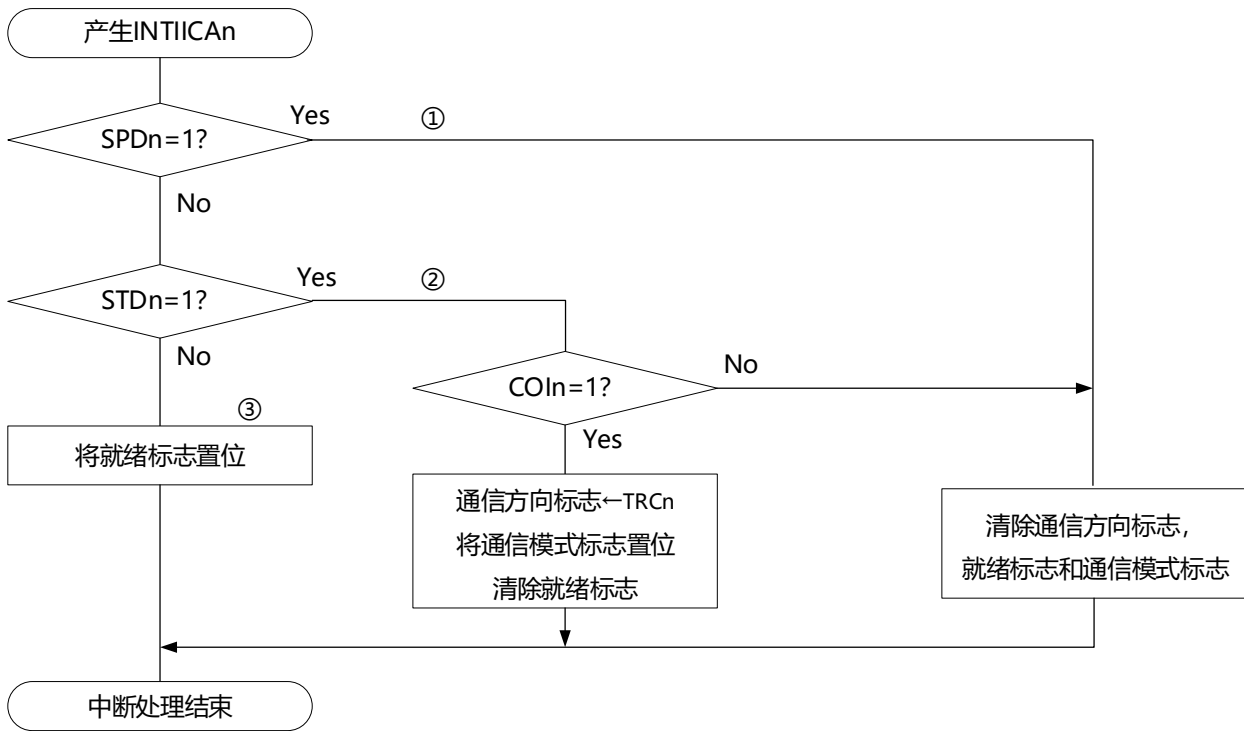
注2：n=0。

从属设备通过INTIICAn中断进行处理的步骤例子如下所示（在此假设没有用扩展码进行处理）。通过INTIICAn中断确认状态并且进行以下处理。

- ① 如果生成停止条件，就结束通信。
- ② 如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设定为通信模式并且解除等待，然后从中断返回（清除就绪标志）。
- ③ 当发送和接收数据时，只要将就绪标志置位，I<sup>2</sup>C总线就保持等待状态并且从中断返回。

注：上述的①~③对应“图16-21 从属运行步骤(2)”的①~③。

图16-21 从属运行步骤(2)



注：n=0。

### 16.7.17 I<sup>2</sup>C中断请求 (INTIICAn) 的产生时序

数据的发送和接收时序、INTIICAn中断请求信号的产生时序以及产生INTIICAn信号时的IICA状态寄存器n (IICSn) 的值如下所示。

注1: ST : 开始条件。

注2: AD6~AD0 : 地址。

注3: R/W : 传送方向的指定。

注4: ACK : 应答。

注5: D7~D0 : 数据。

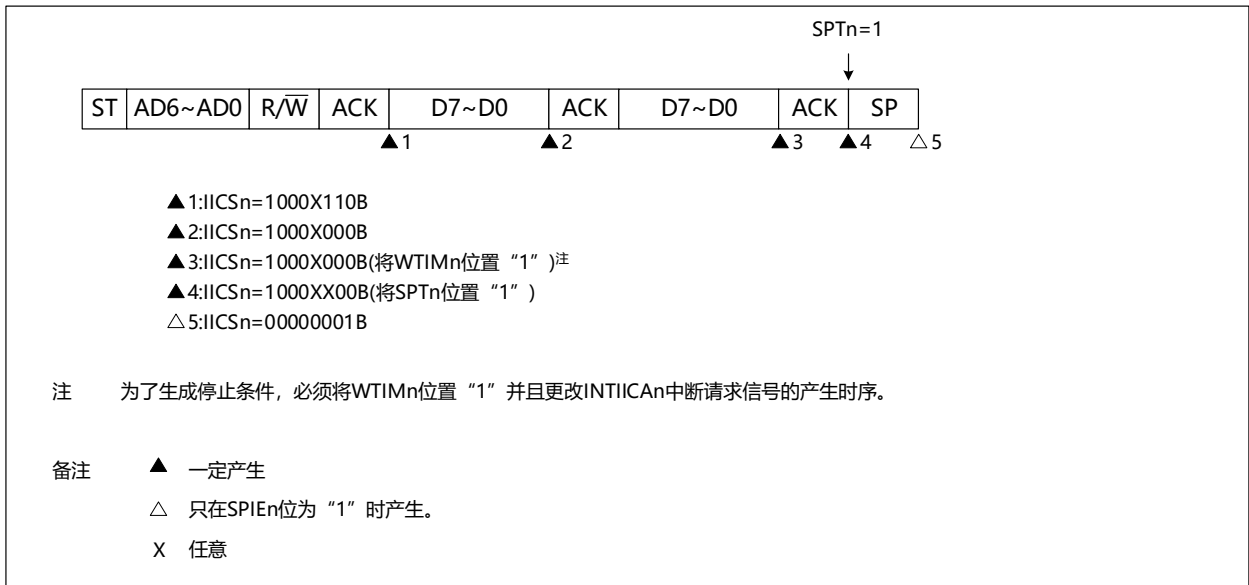
注6: SP : 停止条件。

注7: n=0。

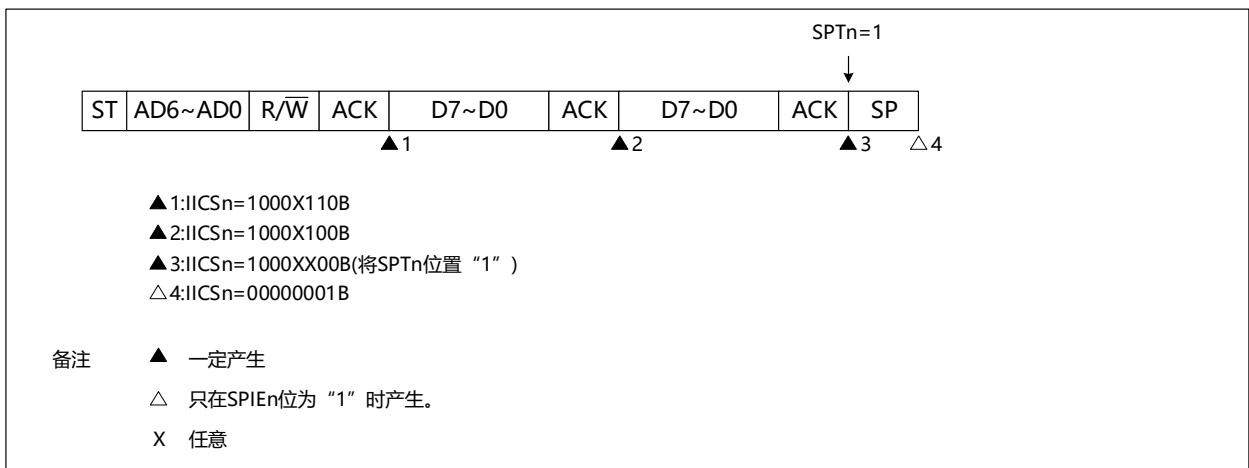
## (1) 主控运行

## ① Start~Address~Data~Data~Stop (发送和接收)

## a) WTIMn=0 的情况



## b) WTIM0=1 的情况

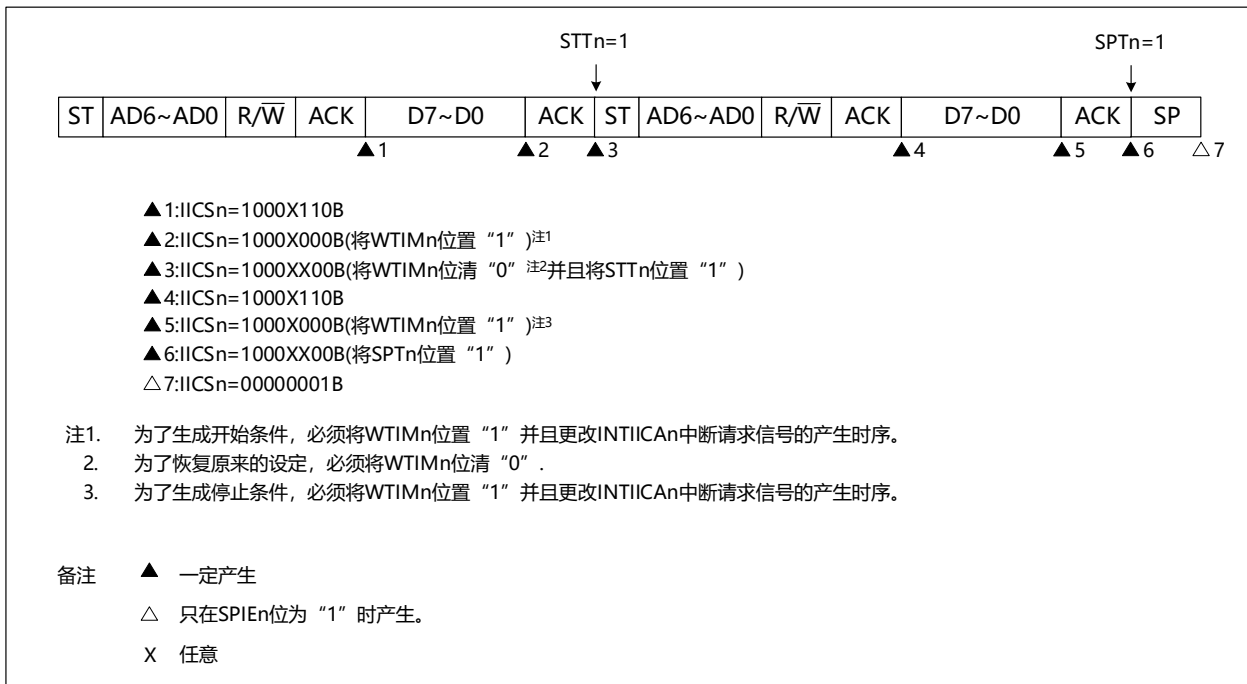


注: n=0。

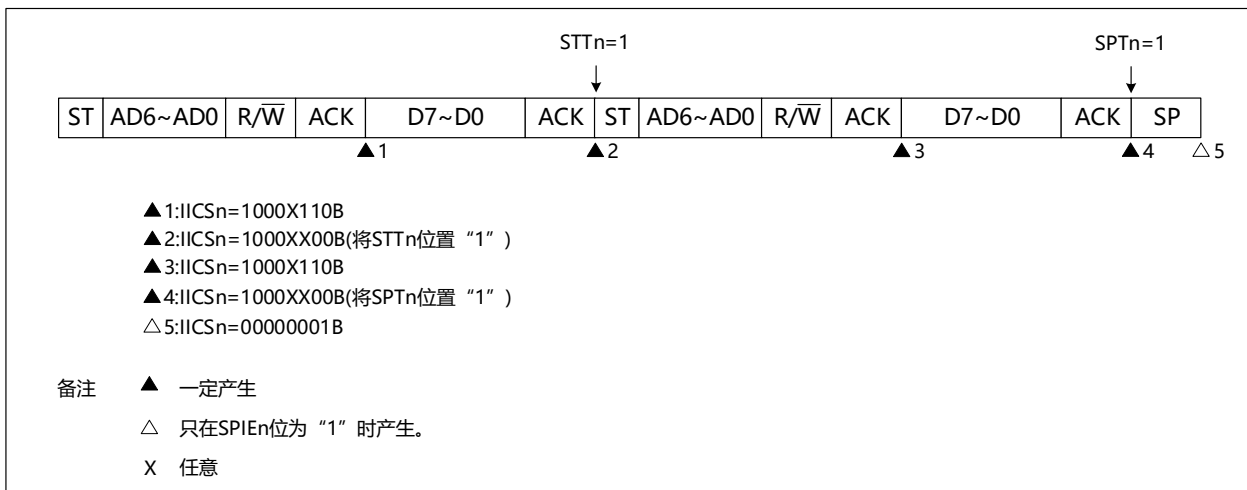


## ② Start~Address~Data~Start~Address~Data~Stop (重新开始)

## a) WTIMn=0 的情况



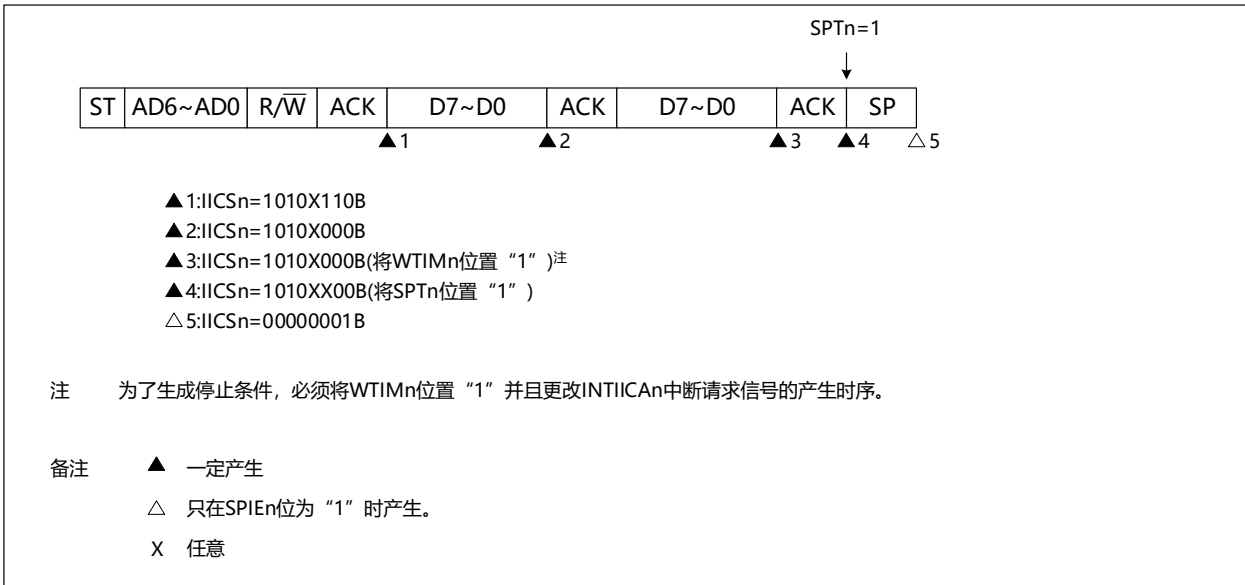
## b) WTIMn=1 的情况



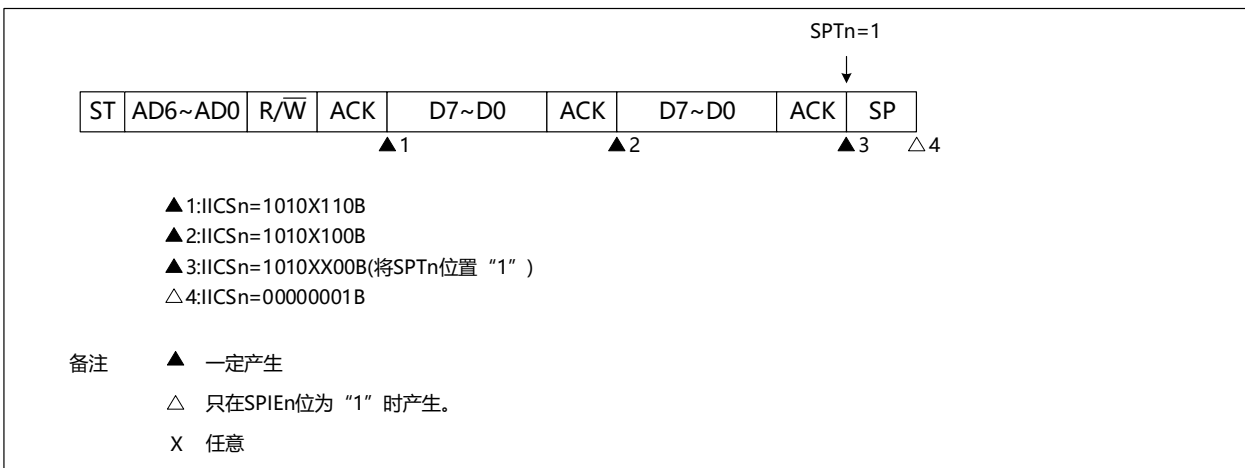
注: n=0。

## ③ Start~Code~Data~Data~Stop (发送扩展码)

## a) WTIMn=0 的情况



## b) WTIMn=1 的情况

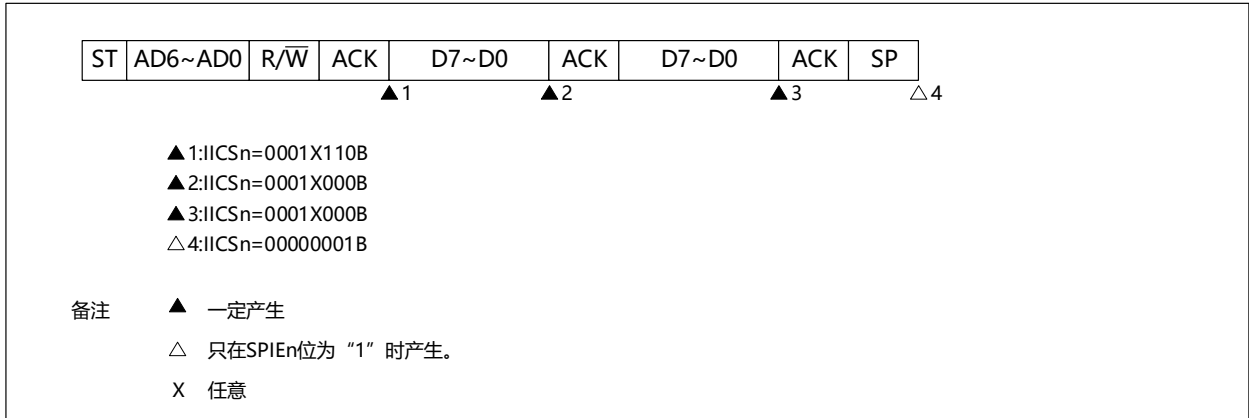


注: n=0。

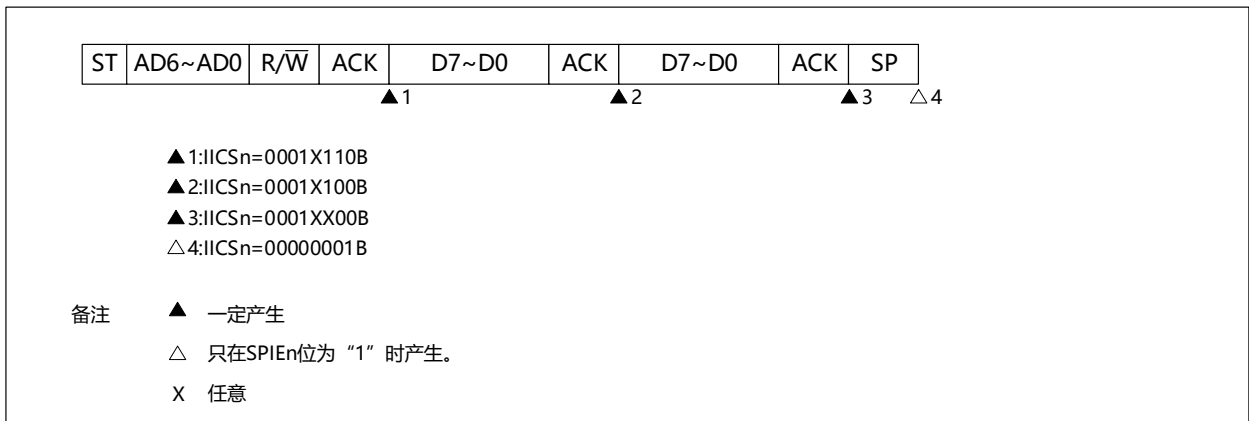
## (2) 从属运行 (接收从属地址的情况)

## ① Start~Address~Data~Data~Stop

## a) WTIMn=0 的情况



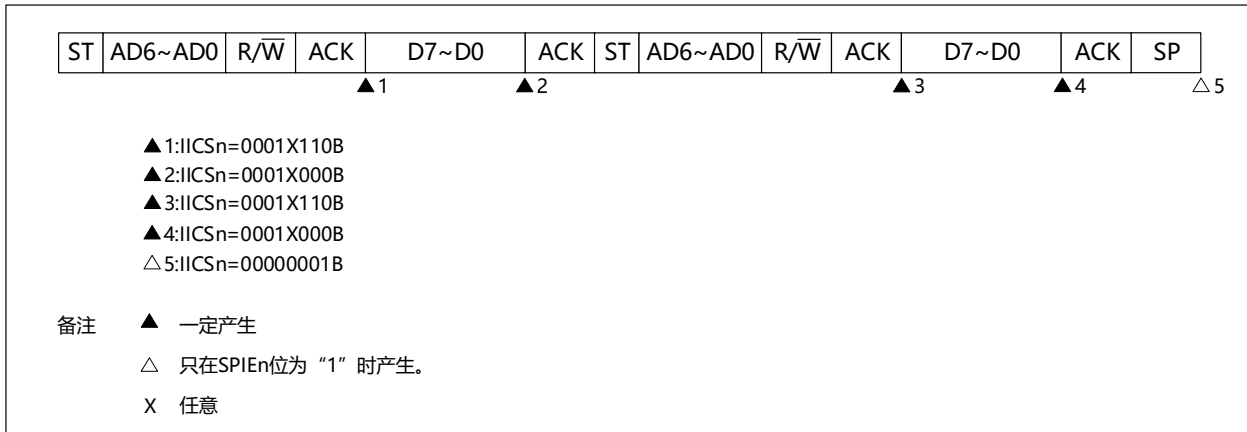
## b) WTIMn=1 的情况



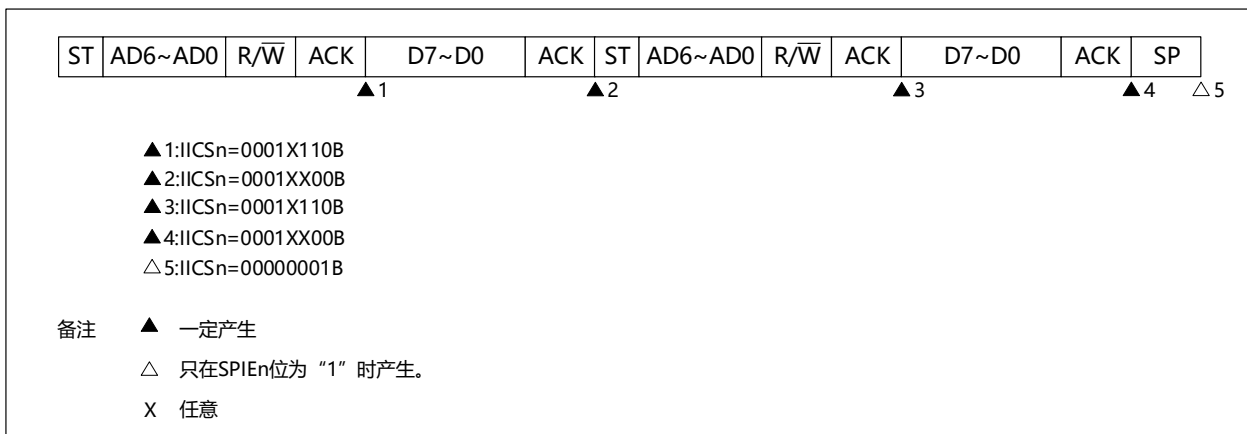
注: n=0。

## ② Start~Address~Data~Start~Address~Data~Stop

## a) WTIMn=0 的情况 (在重新开始后 SVAn 相同)



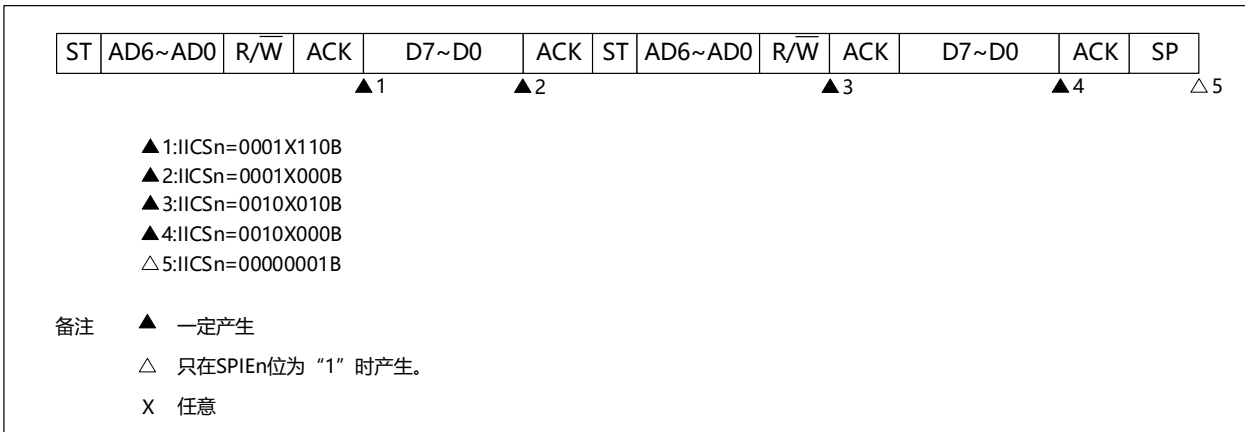
## b) WTIMn=1 的情况 (在重新开始后 SVAn 相同)



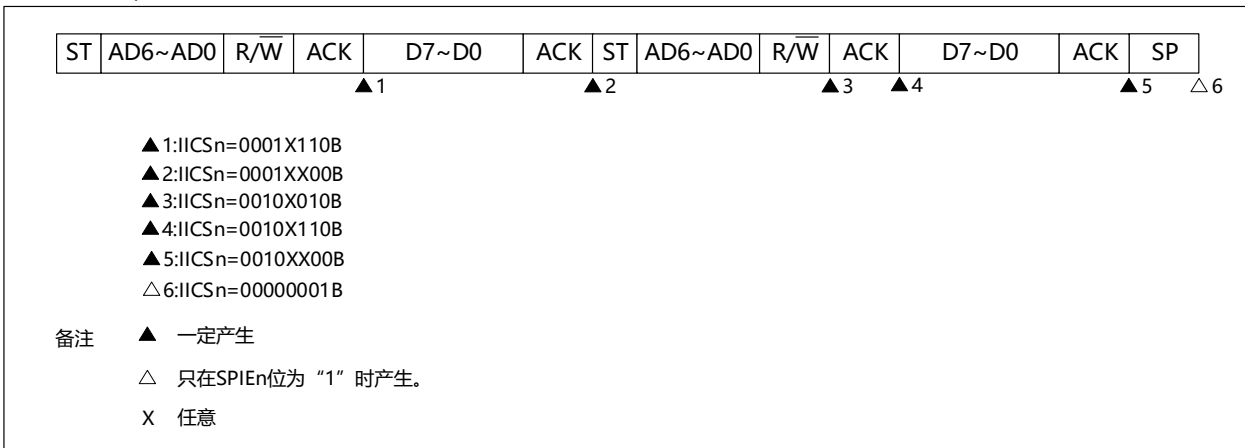
注: n=0。

## ③ Start~Address~Data~Start~Code~Data~Stop

## a) WTIMn=0 的情况 (在重新开始后地址不同 (扩展码))



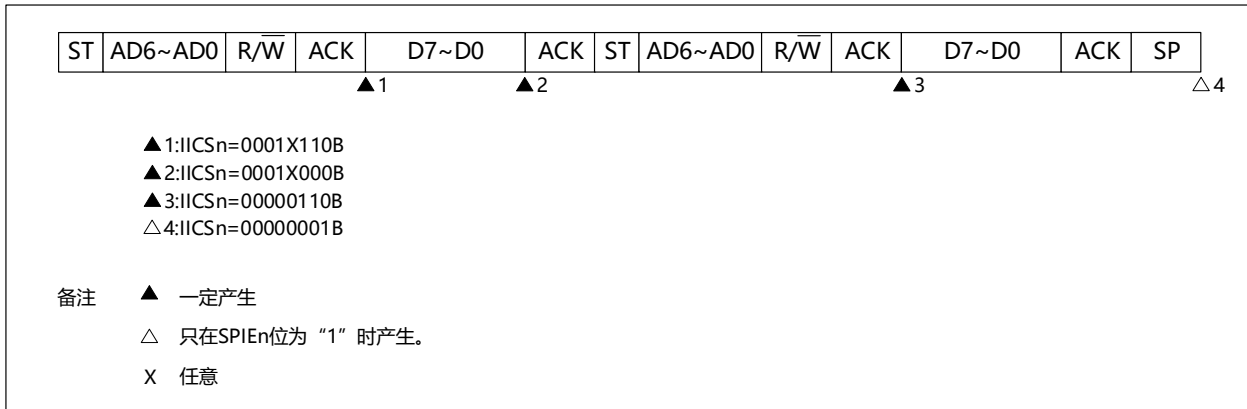
## b) WTIMn=1 的情况 (在重新开始后地址不同 (扩展码))



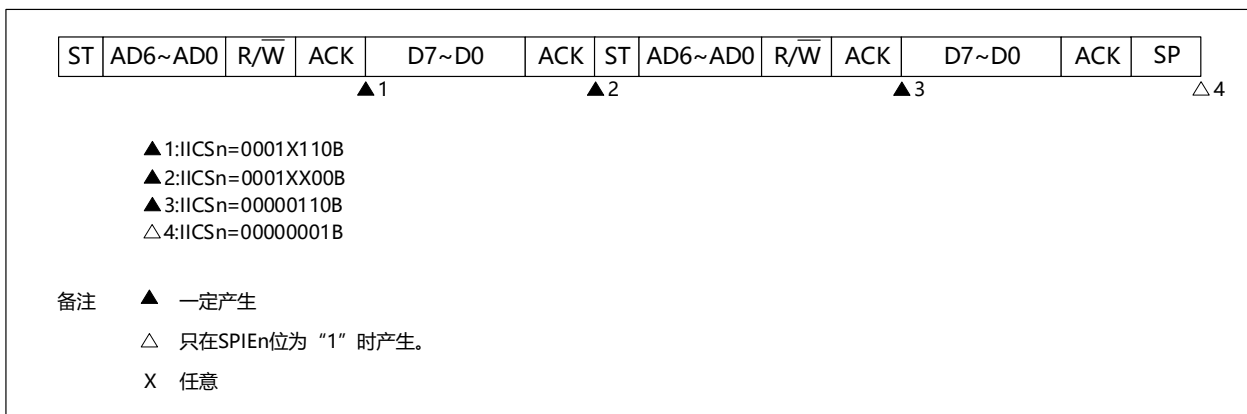
注: n=0。

## ④ Start~Address~Data~Start~Address~Data~Stop

## a) WTIMn=0 的情况 (在重新开始后地址不同 (非扩展码))



## b) WTIMn=1 的情况 (在重新开始后地址不同 (非扩展码))



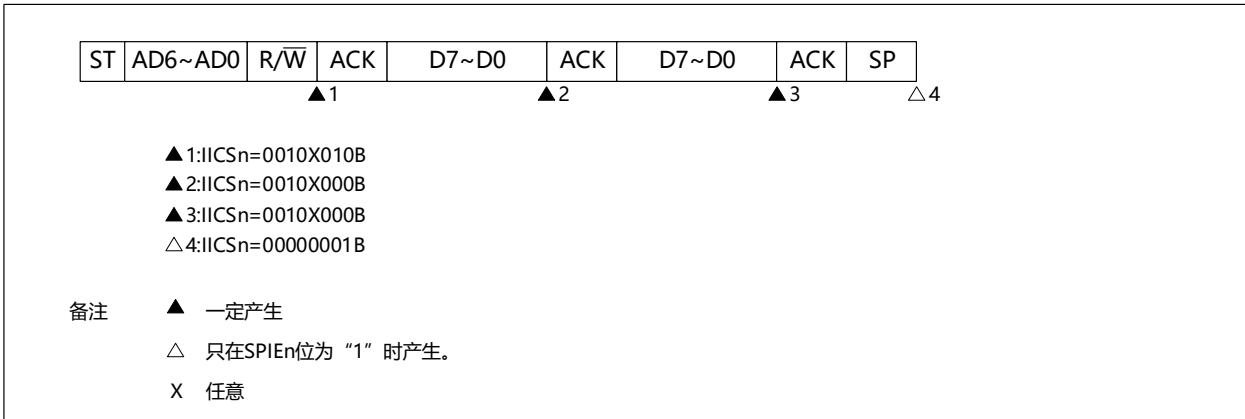
注: n=0。

## (3) 从属运行 (接收扩展码的情况)

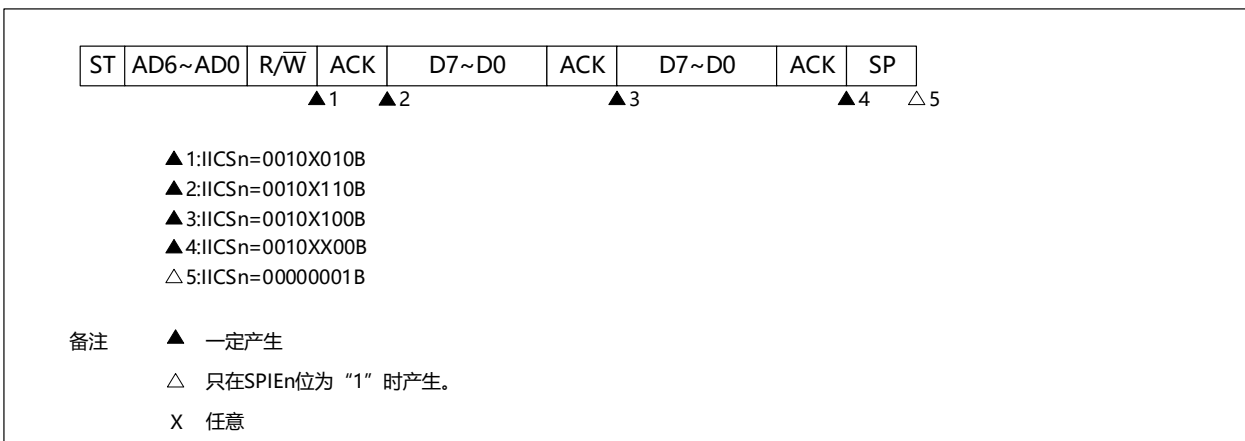
在接收扩展码时, 始终参加通信。

## ① Start~Code~Data~Data~Stop

## a) WTIMn=250 的情况



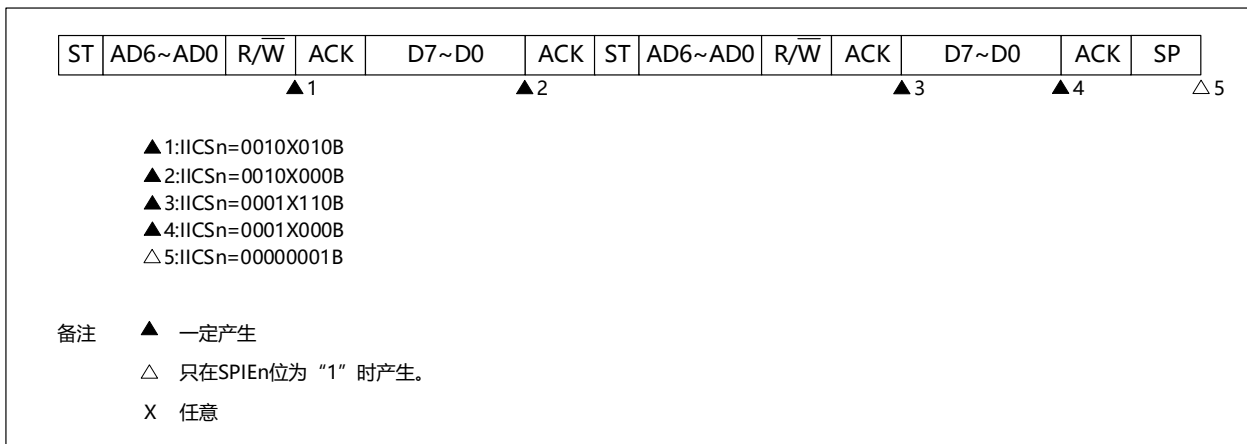
## b) WTIMn=1 的情况



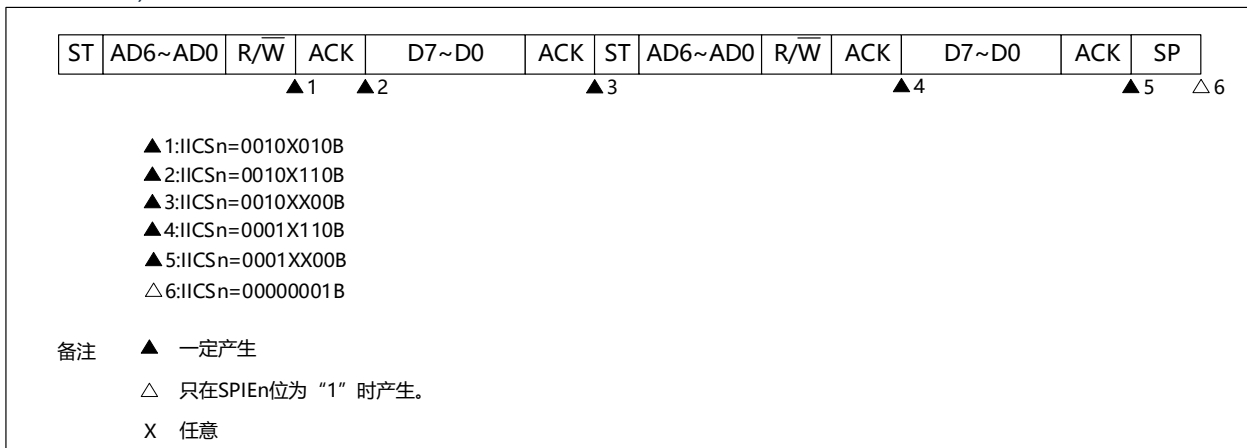
注: n=0。

## ② Start~Code~Data~Start~Address~Data~Stop

## a) WTIMn=0 的情况 (在重新开始后 SVAn 相同)



## b) WTIMn=1 的情况 (在重新开始后 SVAn 相同)

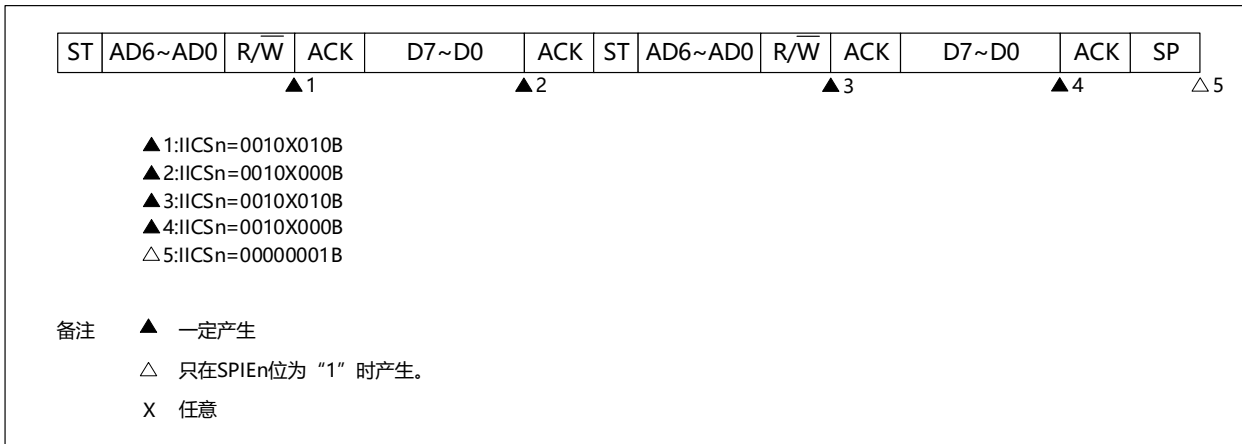


注: n=0。

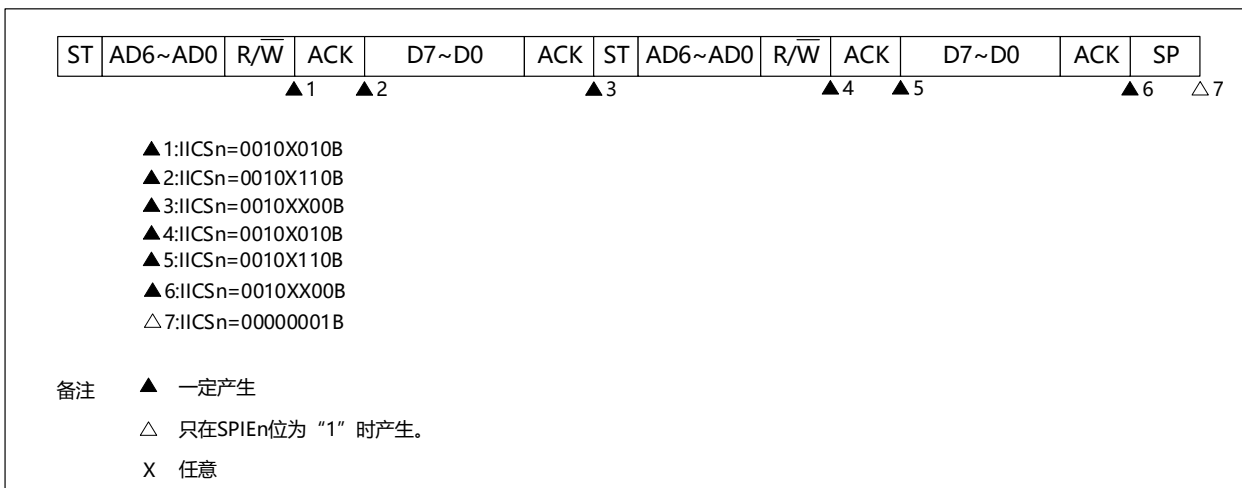


③ Start~Code~Data~Start~Code~Data~Stop

a) WTIMn=0 的情况 (在重新开始后接收扩展码)



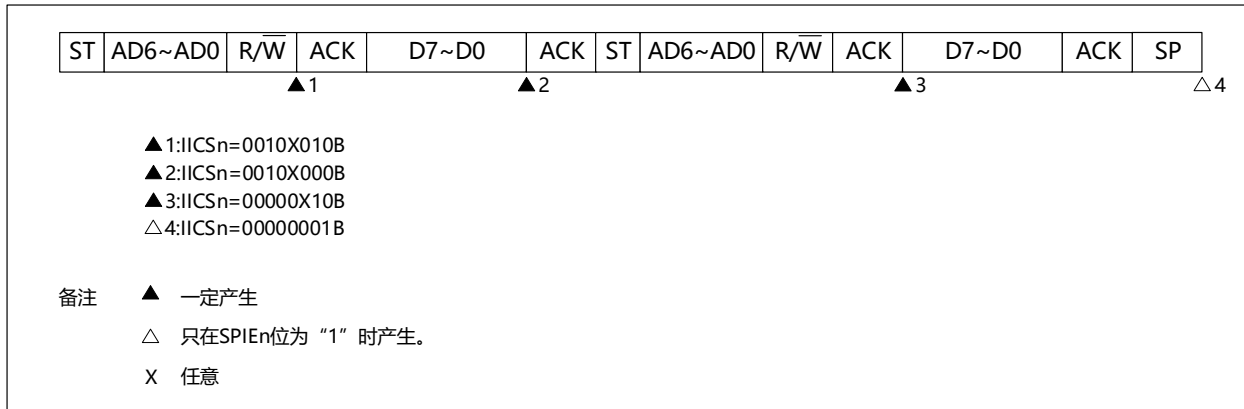
b) WTIMn=1 的情况 (在重新开始后接收扩展码)



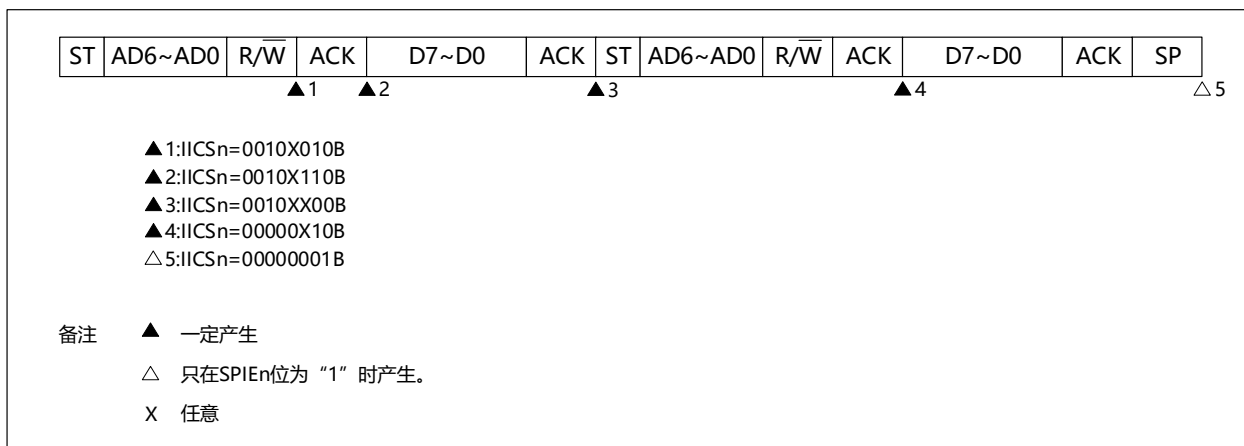
注: n=0。

## ④ Start~Code~Data~Start~Address~Data~Stop

a) WTIMn=0 的情况 (在重新开始后地址不同 (非扩展码))



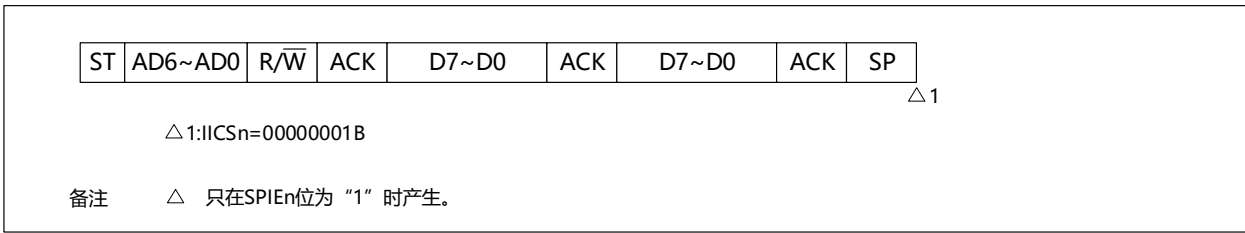
b) WTIMn=1 的情况 (在重新开始后地址不同 (非扩展码))



注: n=0。

(4) 不参加通信的运行

Start~Code~Data~Data~Stop

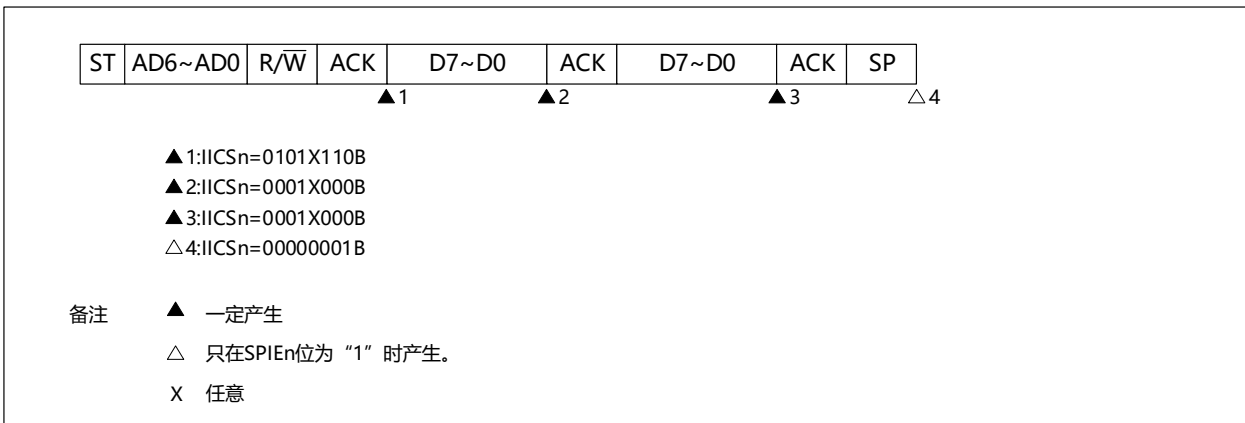


(5) 仲裁失败的运行 (在仲裁失败后作为从属设备运行)

在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

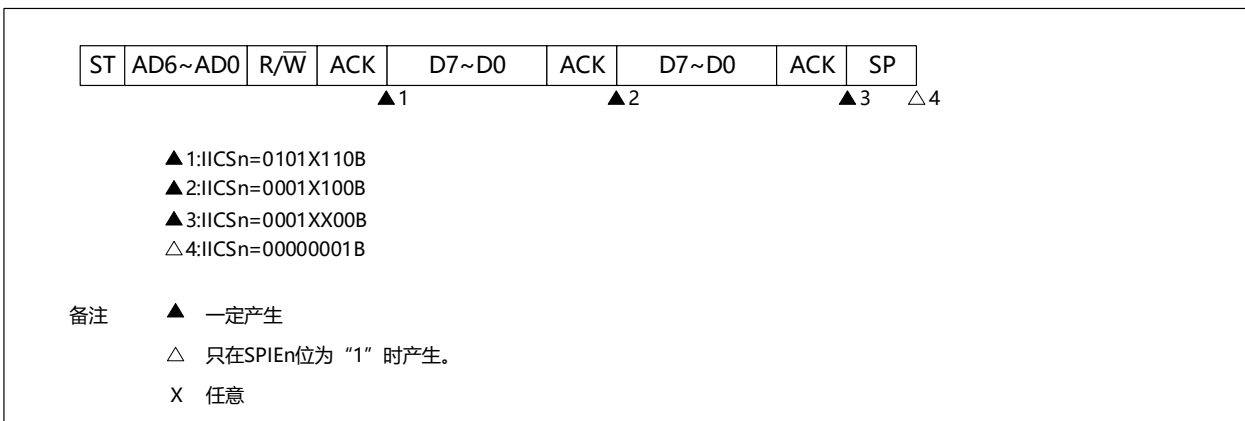
① 在发送从属地址数据的过程中仲裁失败的情况

a) WTIMn=0 的情况



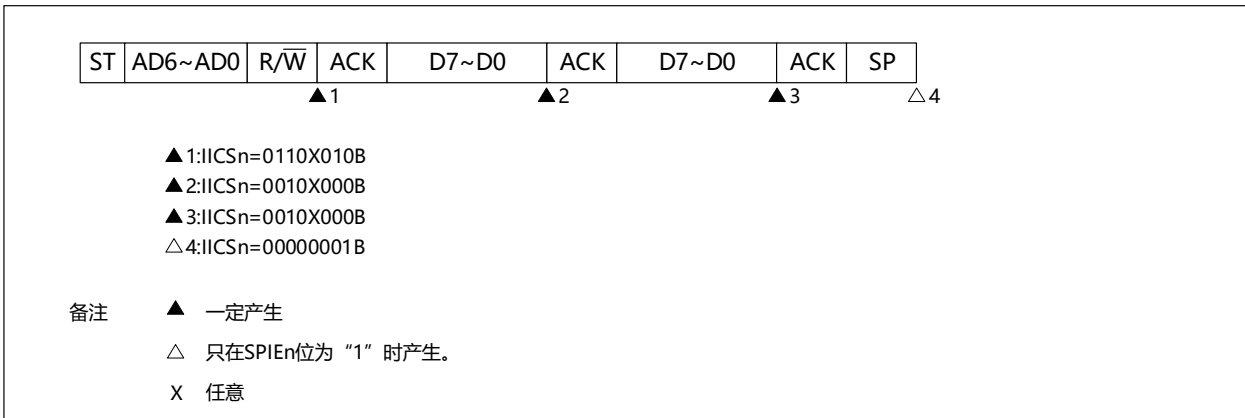
注：n=0。

b) WTIMn=1 的情况



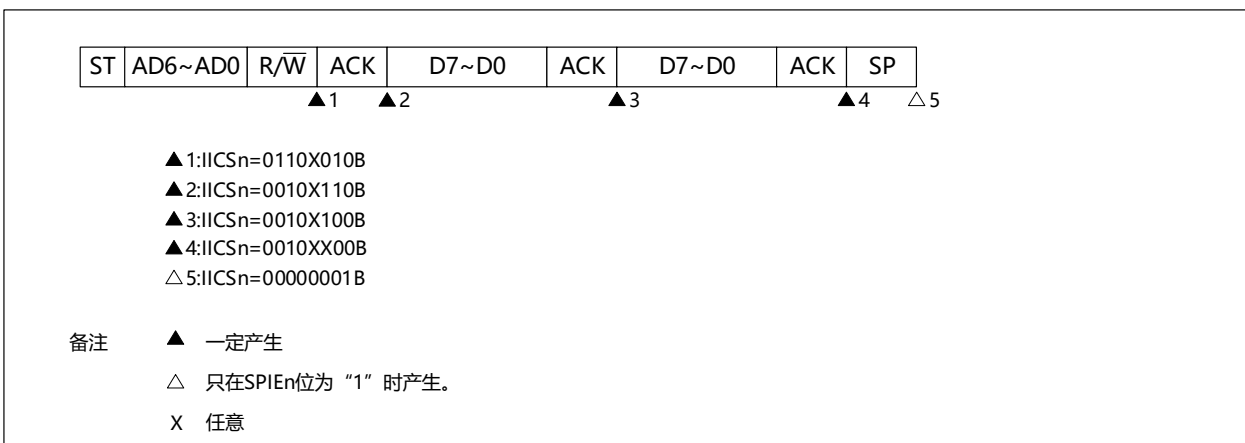
② 在发送扩展码的过程中仲裁失败的情况

a) WTIMn=0 的情况



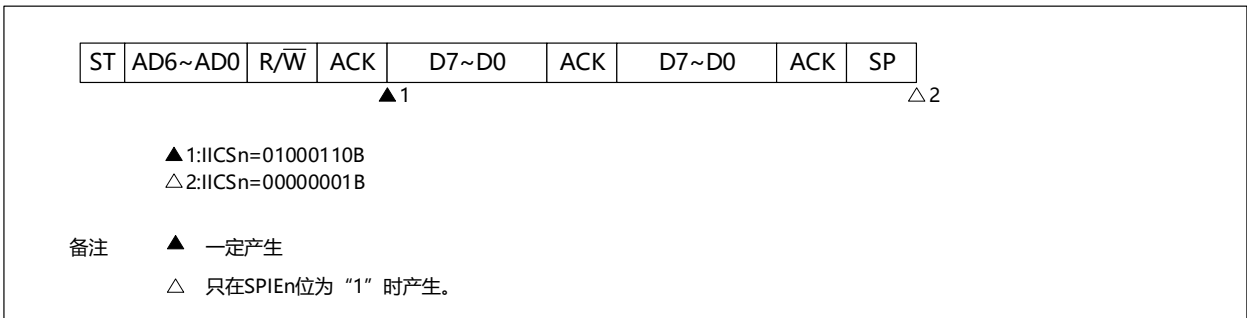
注: n=0。

b) WTIMn=1 的情况

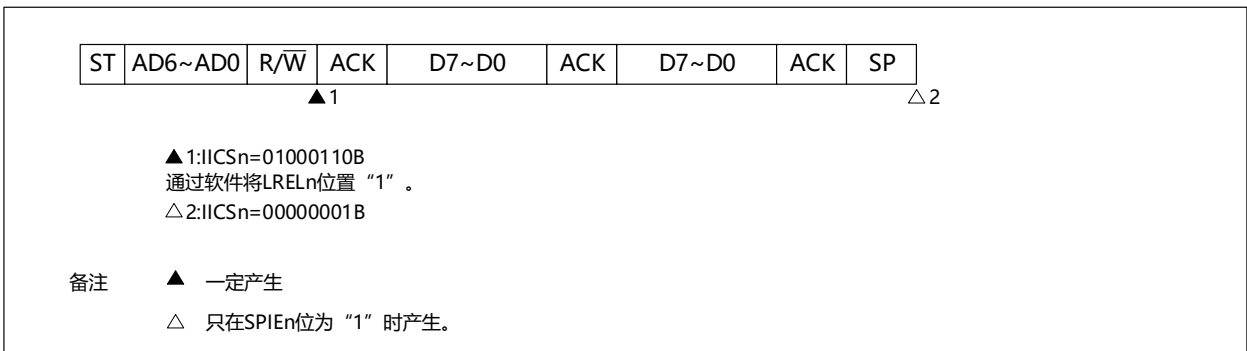
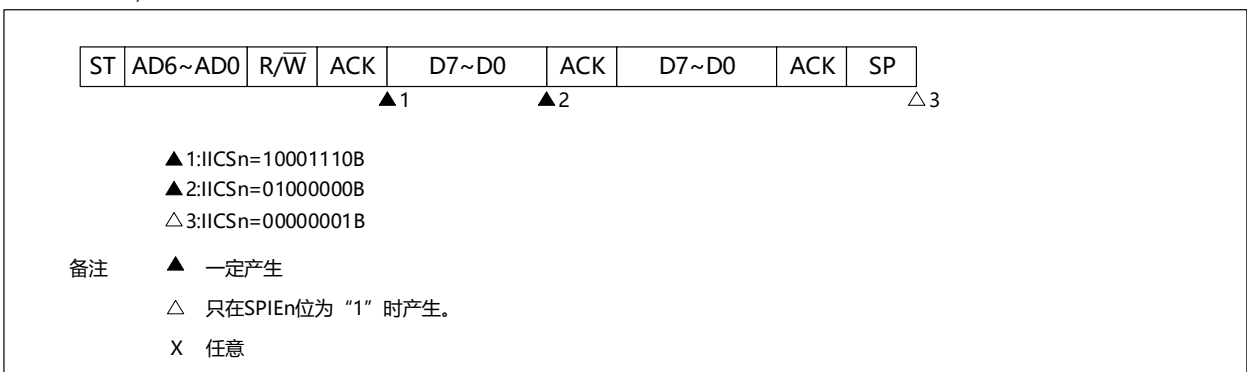


**(6) 仲裁失败的运行 (在仲裁失败后不参加通信)**

在多主控系统中用作主控设备时, 必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位, 确认仲裁结果。

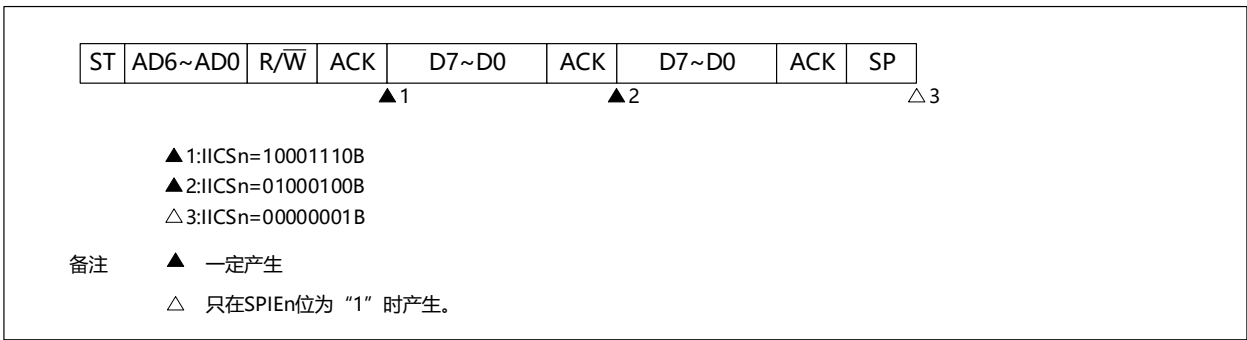
**① 在发送从属地址数据的过程中仲裁失败的情况 (WTIMn=1)**


注: n=0。

**② 在发送扩展码的过程中仲裁失败的情况**

**③ 在传送数据时仲裁失败的情况**
**a) WTIMn=0 的情况**


注: n=0。

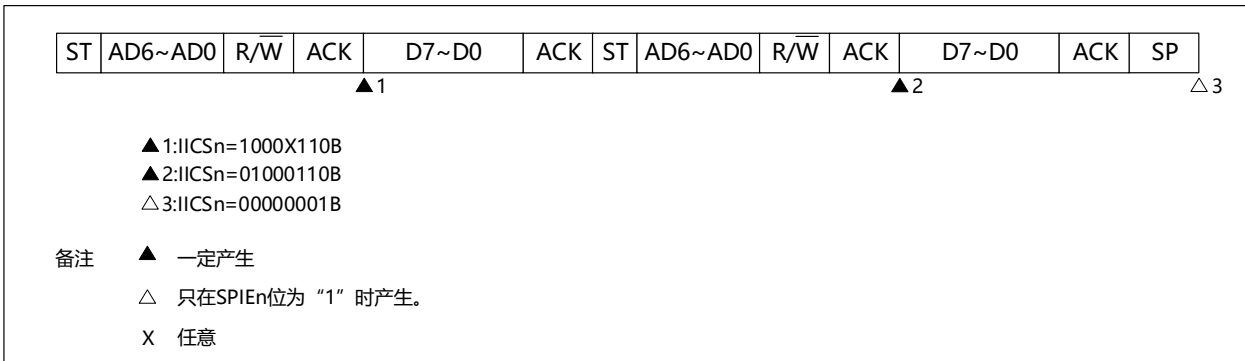
## b) WTIMn=1 的情况



注：n=0。

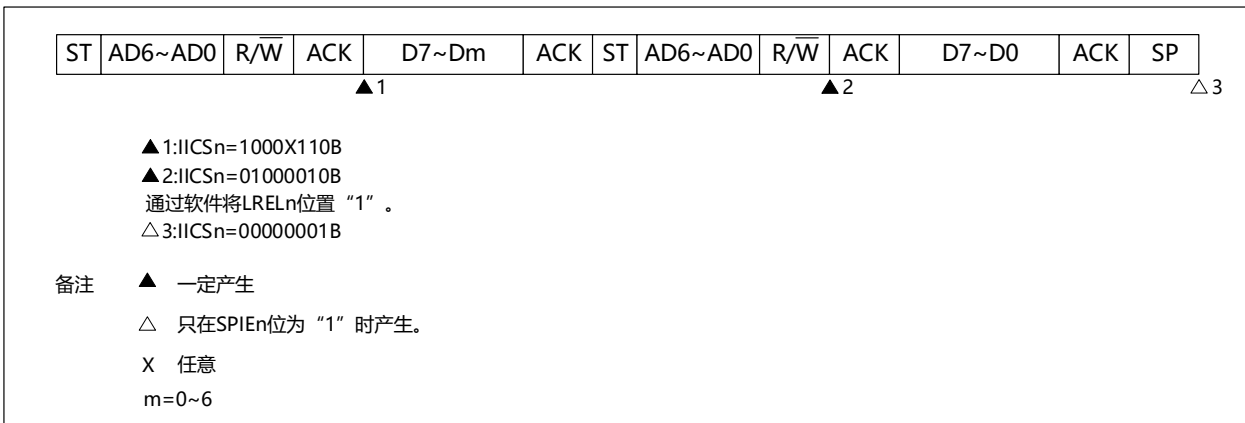
## ④ 在传送数据时因重新开始条件而仲裁失败的情况

## a) 非扩展码 (例如, SVAn 不同)

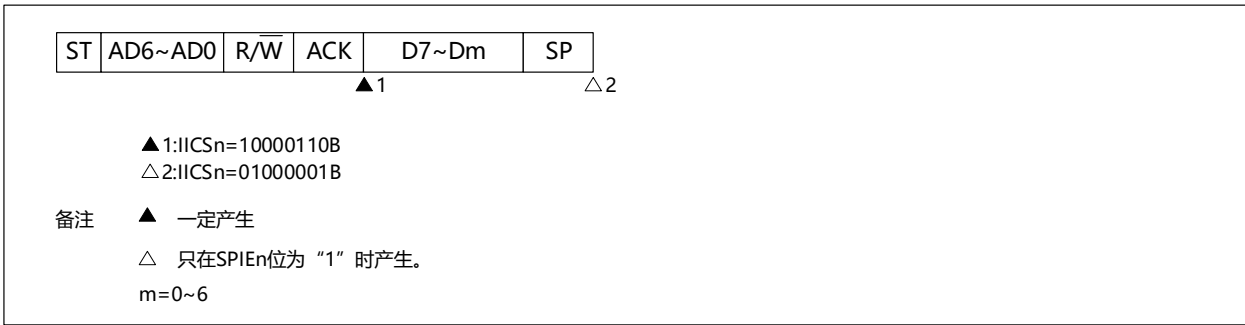


注: n=0。

## b) 扩展码



⑤ 在传送数据时因停止条件而仲裁失败的情况

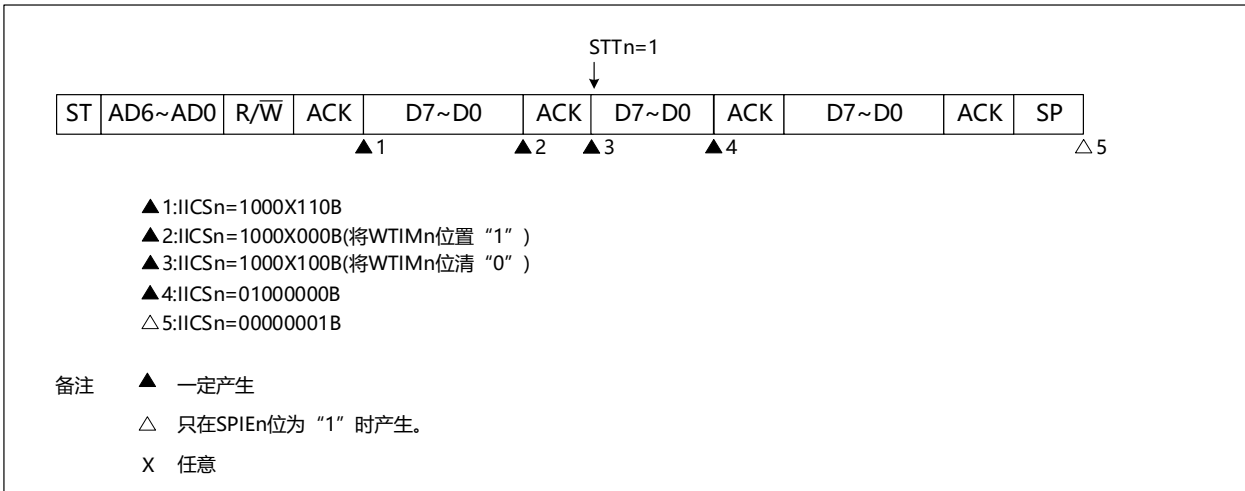


注：n=0。

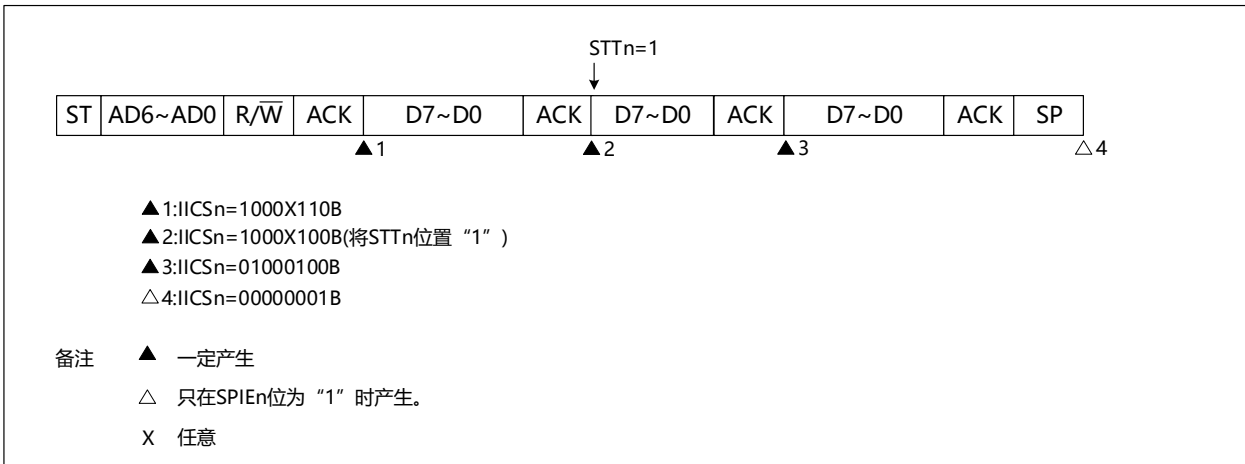


⑥ 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

a) WTIMn=0 的情况



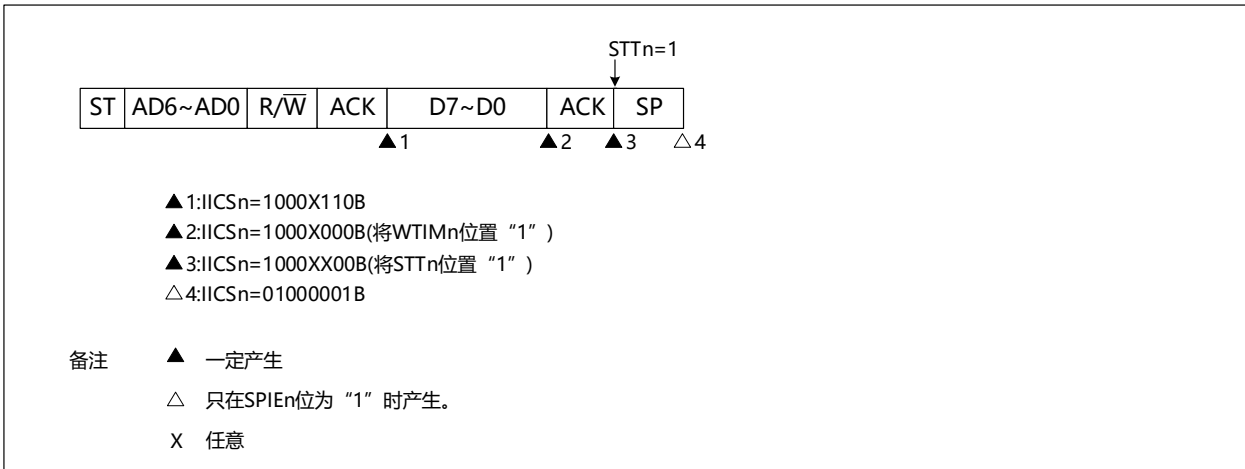
b) WTIMn=1 的情况



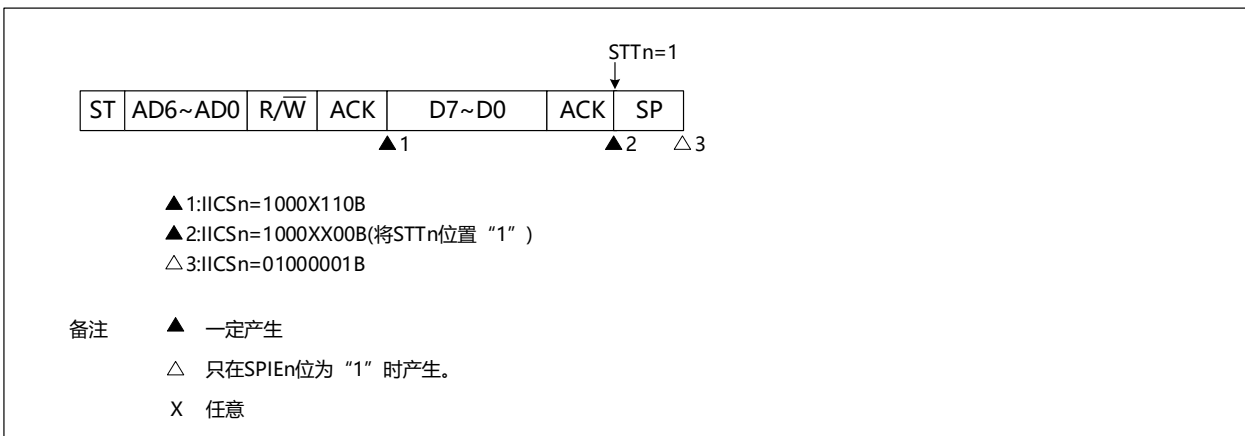
注: n=0。

⑦ 在想要生成重新开始条件时因停止条件而仲裁失败的情况

a) WTIMn=0 的情况



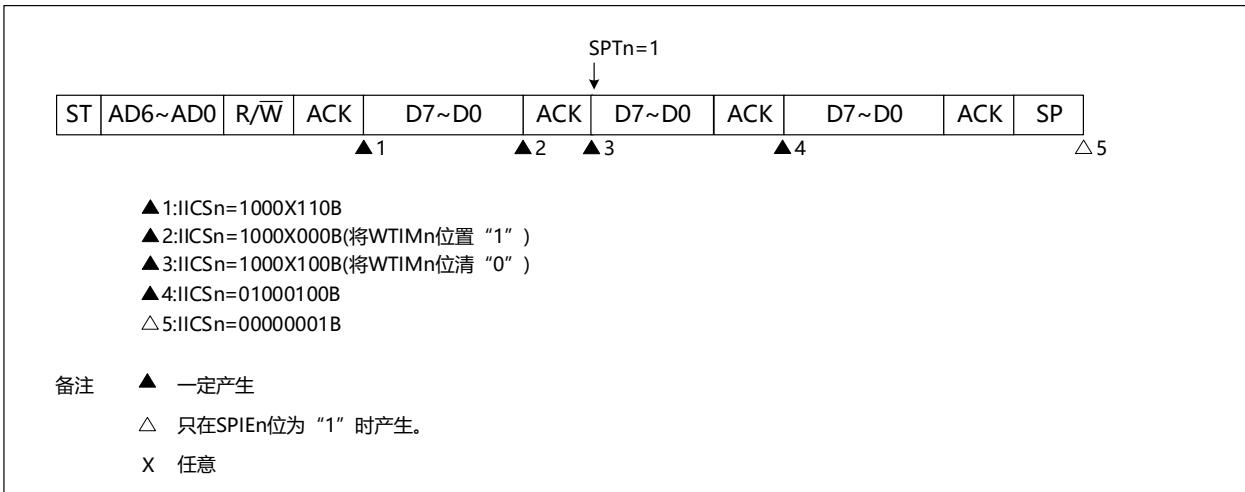
b) WTIMn=1 的情况



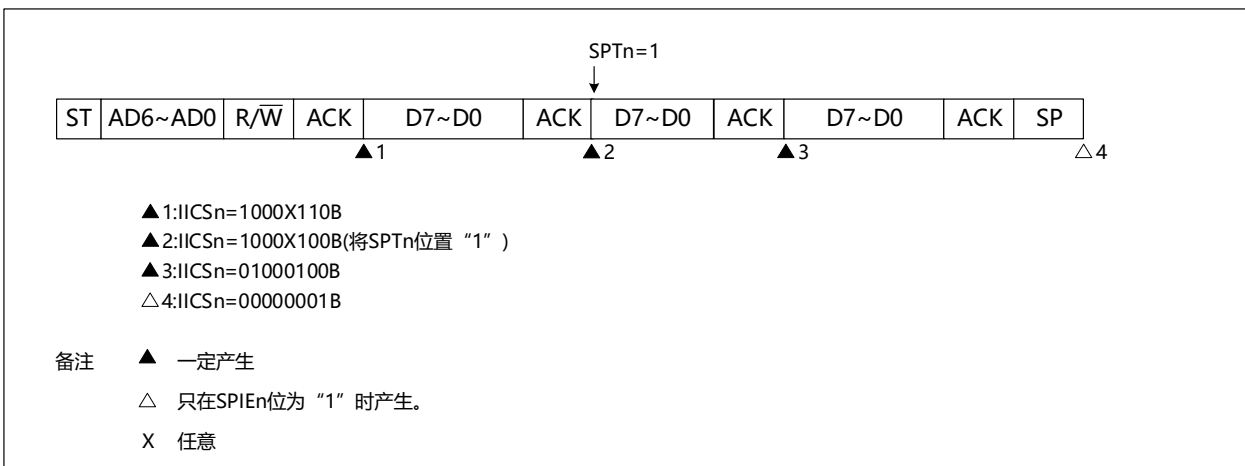
注: n=0。

⑧ 在想要生成停止条件时因数据为低电平而仲裁失败的情况

a) WTIMn=0 的情况



b) WTIMn=1 的情况



注: n=0。

## 16.8 时序图

在I<sup>2</sup>C总线模式中，主控设备通过给串行总线输出地址，从多个从属设备中选择一个通信对象的从属设备。主控设备在从属设备地址之后发送表示数据传送方向的TRCn位（IICA状态寄存器n（IICSn）的bit3），开始与从属设备进行串行通信。数据通信的时序图如图16-22和图16-23所示。

与串行时钟（SCLAn）的下降沿同步进行IICA移位寄存器n（IICAn）的移位，并且将发送数据传送到SO锁存器，以MSB优先从SDAAn引脚输出数据。

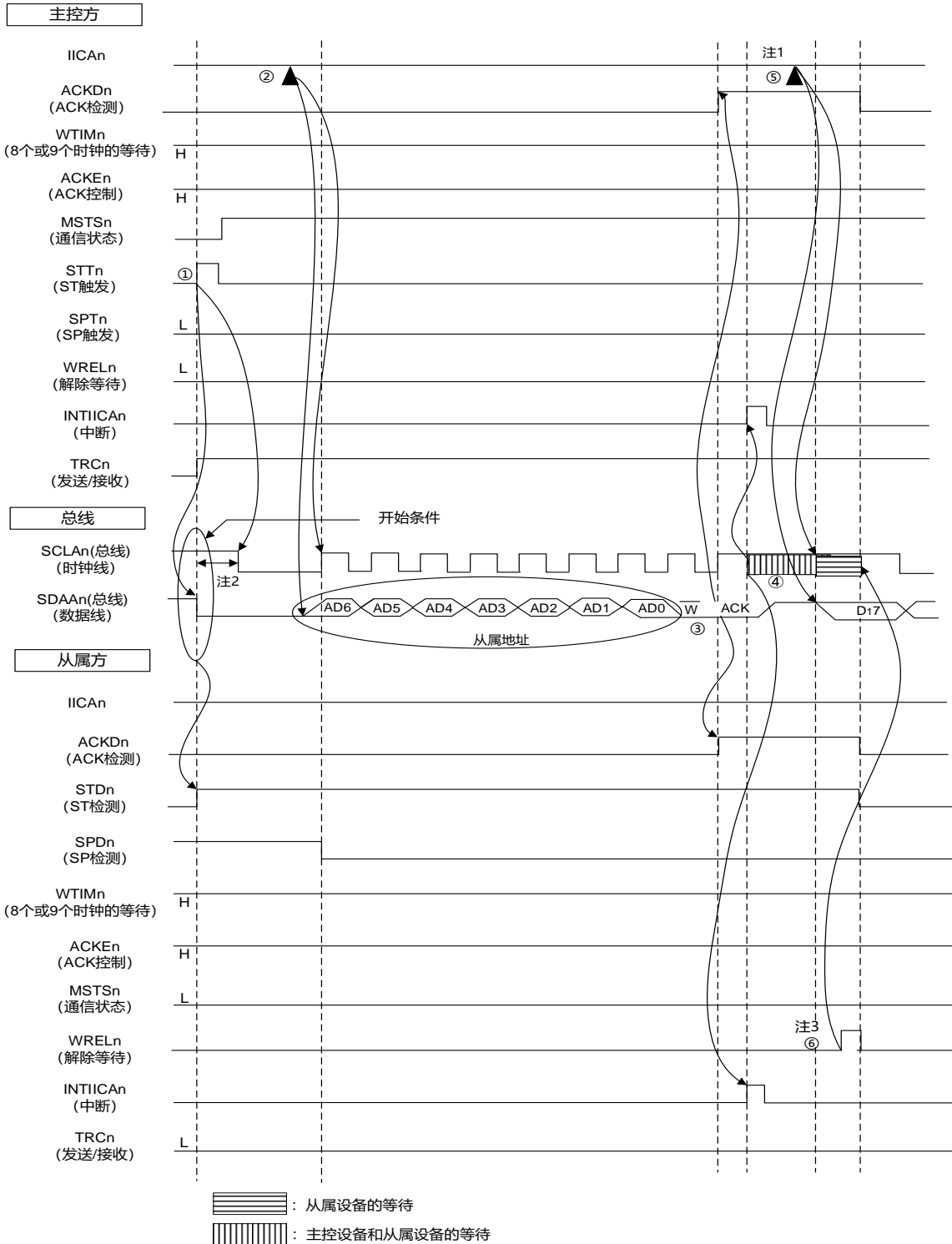
在SCLAn的上升沿将SDAAn引脚输入的数据取到IICAn。

注：n=0。

图16-22 主控设备&从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待) (1/4)

(1) 开始条件~地址~数据



注1：要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

注2：SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设定为标准模式时至少为4.0us，在设定为快速模式时至少为0.6us。

注3：要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图 16-22 的“(1)开始条件~地址~数据”的①~⑥的说明如下:

- ① 如果在主控方将开始条件触发置位 (STTn=1), 总线数据线 (SDAAn) 就下降, 生成开始条件 (通过 SCLAn=1 使 SDAAn 从“1”变为“0”)。此后, 如果检测到开始条件, 主控方就进入主控通信状态 (MSTSn=1), 在经过保持时间后总线时钟线下降 (SCLAn=0), 结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 n (IICAn) 写地址+W (发送), 就发送从属地址。
- ③ 在从属方, 如果接收地址和本地站地址 (SVAn 的值) 相同<sup>注</sup>, 就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ④ 主控方在第 9 个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 地址匹配中断)<sup>注</sup>。
- ⑤ 主控方给 IICAn 寄存器写发送数据, 解除主控方的等待。
- ⑥ 如果从属方解除等待 (WRELn=1), 主控方就开始给从属方传送数据。

注1: 如果发送的地址和从属地址不同, 从属方就不给主控方返回ACK (NACK: SDAAn=1), 并且不产生INTIICAn中断 (地址匹配中断), 也不进入等待状态。但是, 主控方对于ACK和NACK都产生INTIICAn中断 (地址发送结束中断)。

注2: 图16-22的①~⑮ 是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

注3: 图16-22的“(1)地址~数据~数据”说明步骤①~⑥。

注4: 图16-22的“(2)地址~数据~数据”说明步骤③~⑩。

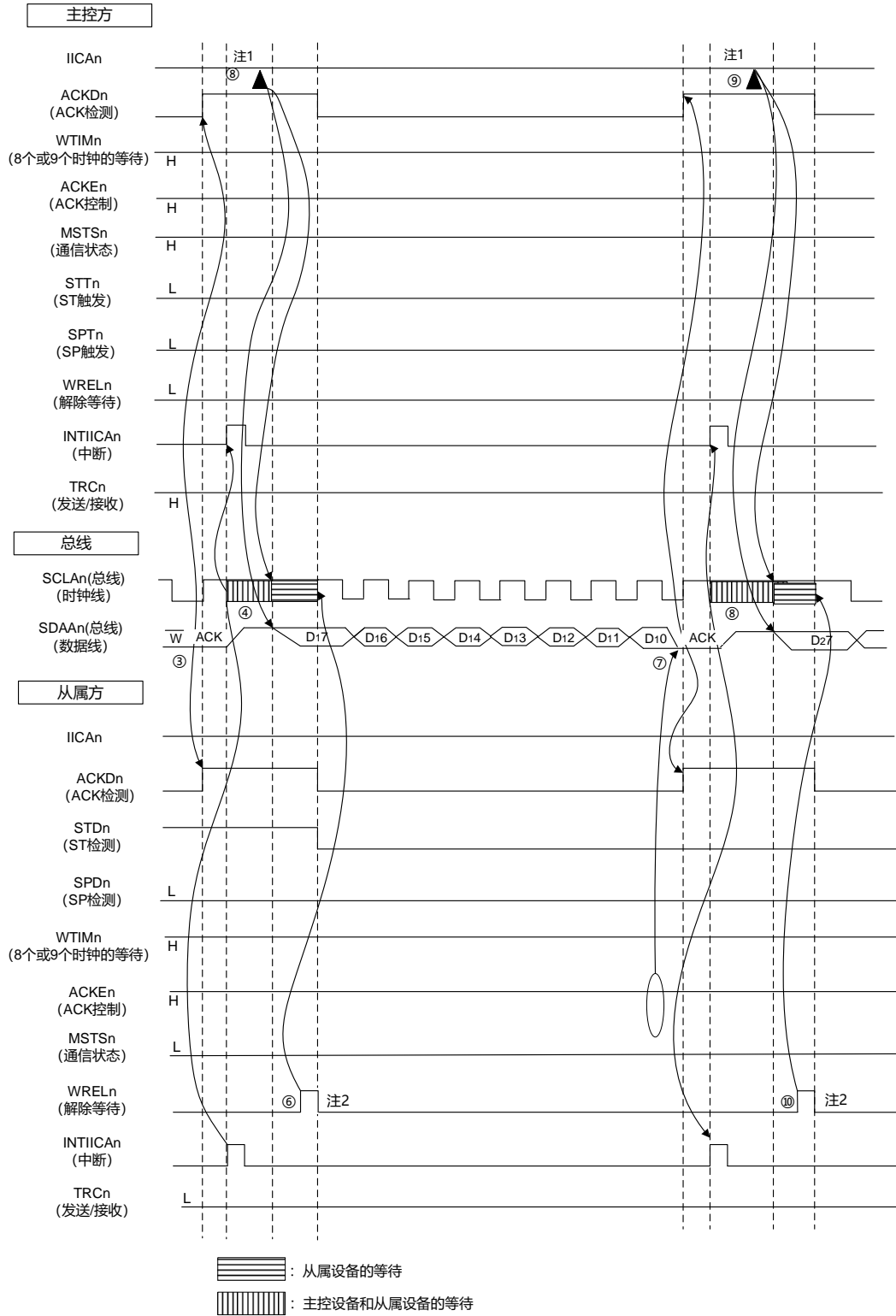
注5: 图16-22的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

注6: n=0。

图16-22 主控设备&从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待) (2/4)

(2) 地址~数据~数据



注1：要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

注2：要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

注3：n=0

图16-22的“(2)地址~数据~数据”的③~⑩的说明如下：

③在从属方，如果接收地址和本地站地址（SVAn的值）相同<sup>注</sup>，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

④主控方在第9个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断<sup>注</sup>）。

⑤主控方给IICA移位寄存器n（IICAn）写发送数据，解除主控方的等待。

⑥如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。

⑦在数据传送结束后，因为从属方的ACKEn位为“1”，所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

⑧主控方和从属方在第9个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。

⑨主控方给IICAn寄存器写发送数据，解除主控方的等待。

⑩如果从属方读接收数据并且解除等待（WRELn=1），主控方就开始给从属方传送数据。

注1：如果发送的地址和从属地址不同，从属方就不给主控方返回ACK（NACK：SDAAn=1），并且不产生INTIICAn中断（地址匹配中断），也不进入等待状态。但是，主控方对于ACK和NACK都产生INTIICAn中断（地址发送结束中断）。

注2：图16-22的①~⑮是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

注3：图16-22的“(1)地址~数据~数据”说明步骤①~⑥。

注4：图16-22的“(2)地址~数据~数据”说明步骤③~⑩。

注5：图16-22的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

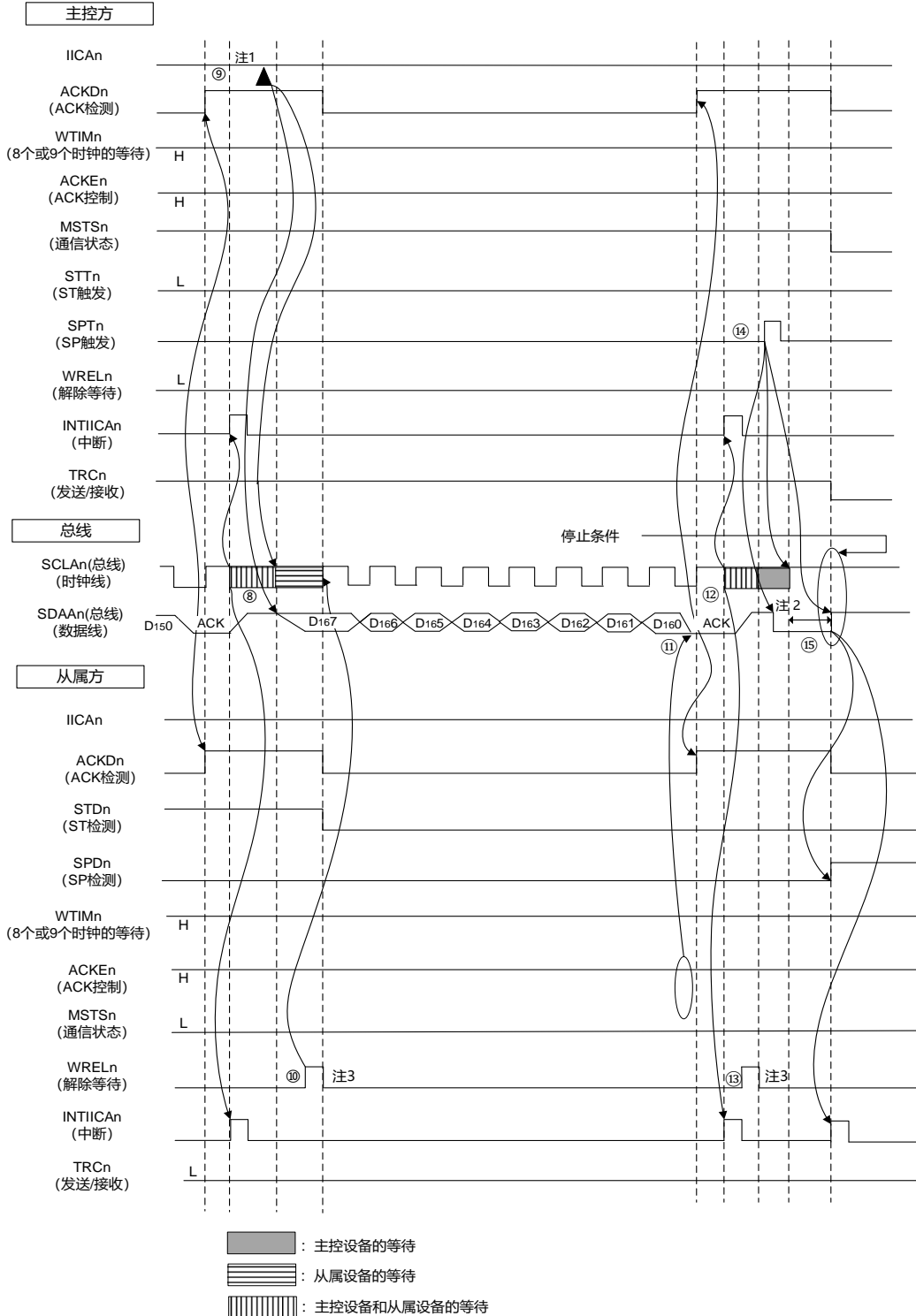
注6：n=0。



图16-22 主控设备&从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待) (3/4)

(3) 数据~数据~停止条件



注1: 要解除主控方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

注2: 在发行停止条件后, 从SCLAn引脚信号上升到生成停止条件的的时间, 在设定为标准模式时至少为4.0us, 在设定为快速模式时至少为0.6us。

注3: 要解除从属方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

图16-22的“(3)数据~数据~停止条件”的⑦~⑮的说明如下

- ⑦ 在数据传送结束后，因为从属方的ACKEn位为“1”，所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。
- ⑧ 主控方和从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0)，并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑨ 主控方给IICA移位寄存器n (IICAn) 写发送数据，解除主控方的等待。
- ⑩ 如果从属方读接收数据并且解除等待 (WRELn=1)，主控方就开始给从属方传送数据。
- ⑪ 在数据传送结束后，从属方 (ACKEn=1) 通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。
- ⑫ 主控方和从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0)，并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑬ 从属方读接收数据，解除等待 (WRELn=1)。
- ⑭ 如果在主控方将停止条件触发置位 (SPTn=1)，就清除总线数据线 (SDAAn=0) 并且将总线时钟线置位 (SCLAn=1)，在经过停止条件的准备时间后将总线数据线置位 (SDAAn=1)，生成停止条件 (通过SCLAn=1使SDAAn从“0”变为“1”)。
- ⑮ 如果生成停止条件，从属方就检测到停止条件并且产生中断 (INTIICAn: 停止条件中断)。

注1: 图16-22的①~⑮ 是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

注2: 图16-22的“(1)地址~数据~数据”说明步骤①~⑥。

注3: 图16-22的“(2)地址~数据~数据”说明步骤③~⑩。

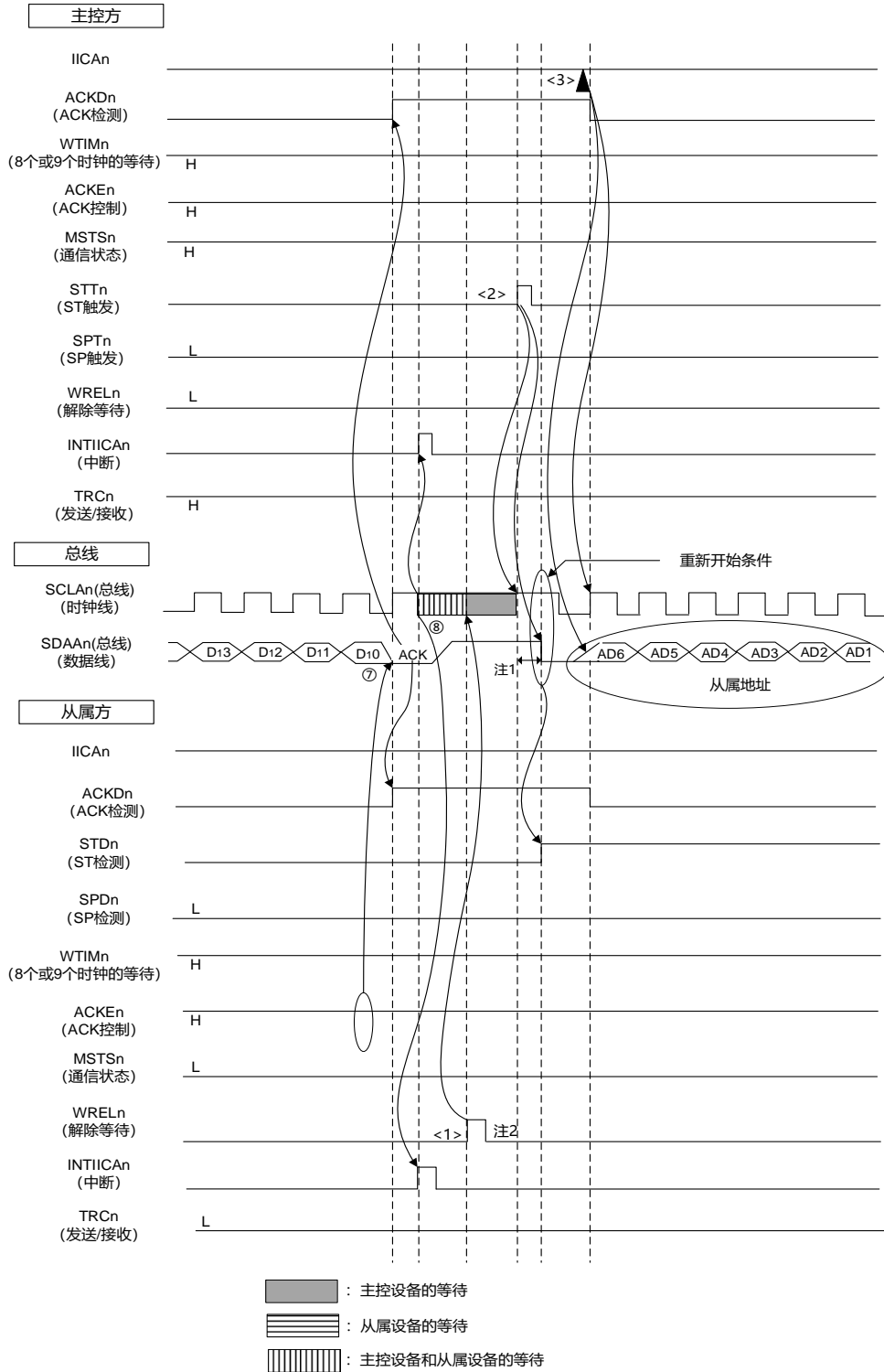
注4: 图16-22的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

注5: n=0。

图16-22 主控设备&从属设备的通信例子

(主控设备: 选择9个时钟的等待, 从属设备: 选择9个时钟的等待) (4/4)

(4) 数据~重新开始条件~地址



注1: 在发行重新开始条件后, 从SCLAn引脚信号上升到生成开始条件的的时间, 在设定为标准模式时至少为4.7us, 在设定为快速模式时至少为0.6us。

注2: 要解除从属方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注3: n=0

图16-22的“(4)数据~重新开始条件~地址”的运行说明如下。在执行步骤⑦和⑧后执行<1>~<3>,从而返回到步骤③的数据发送步骤。

⑦在数据传送结束后,因为从属方的ACKEn位为“1”,所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。

⑧主控方和从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0),并且都产生中断 (INTIICAn: 传送结束中断)。

<1>从属方读接收数据,解除等待 (WRELn=1)。

<2>如果在主控方再次将开始条件触发置位 (STTn=1),总线时钟线就上升 (SCLAn=1),并且在经过重新开始条件的准备时间后总线数据线下下降 (SDAAn=0),生成开始条件 (通过SCLAn=1使SDAAn从“1”变为“0”)。然后,如果检测到开始条件,就在经过保持时间后总线时钟线下降 (SCLAn=0),结束通信准备。

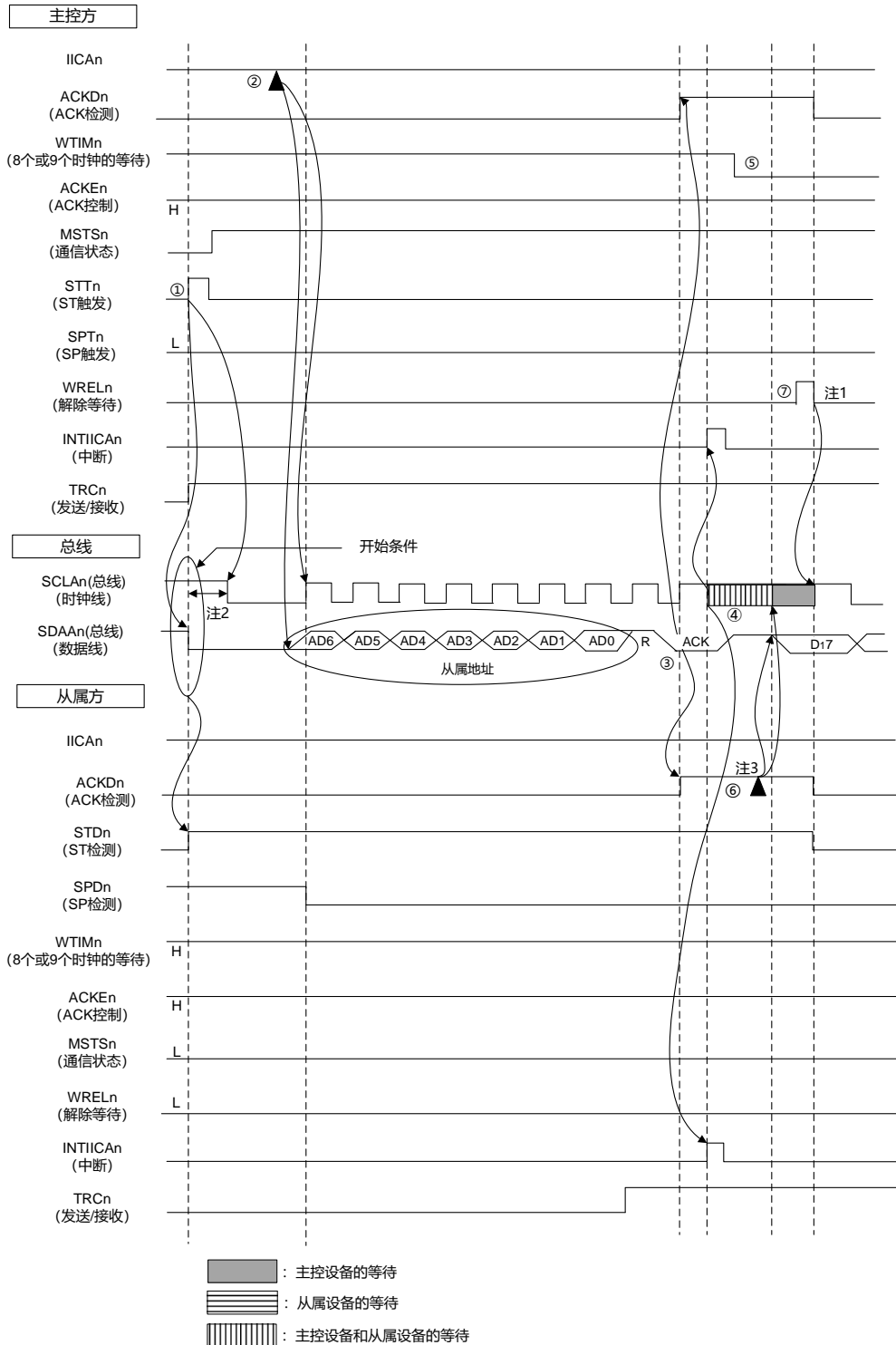
<3>如果主控方给IICA移位寄存器n (IICAn)写地址+R/W (发送),就发送从属地址。

注: n=0

图16-23 从属设备&主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待) (1/3)

(1) 开始条件~地址~数据



注1：要解除主控方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

注2：从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设定为标准模式时至少为4.0us，在设定为快速模式时至少为0.6us。

注3：要解除从属方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

图16-23的“(1)开始条件~地址~数据”的①~⑦的说明如下：

①如果在主控方将开始条件触发置位 (STTn=1)，总线数据线 (SDAAn) 就下降，生成开始条件 (通过 SCLAn=1使SDAAn从“1”变为“0”)。此后，如果检测到开始条件，主控方就进入主控通信状态 (MSTSn=1)，在经过保持时间后总线时钟线下降 (SCLAn=0)，结束通信准备。

②如果主控方给IICA移位寄存器n (IICAn) 写地址+R (接收)，就发送从属地址。

③在从属方，如果接收地址和本地站地址 (SVAn的值) 相同注，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。

④主控方在第9个时钟的下降沿产生中断 (INTIICAn：地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0)，并且产生中断 (INTIICAn：地址匹配中断) 注。

⑤主控方将等待时序改为第8个时钟 (WTIMn=0)。

⑥从属方给IICAn寄存器写发送数据，解除从属方的等待。

⑦主控方解除等待 (WRELn=1)，开始来自从属设备的数据传送。

注1：如果发送的地址和从属地址不同，从属方就不给主控方返回ACK (NACK：SDAAn=1)，并且不产生INTIICAn中断 (地址匹配中断)，也不进入等待状态。但是，主控方对于ACK和NACK都产生INTIICAn中断 (地址发送结束中断)。

注2：图16-23的①~⑱是通过I2C总线进行数据通信的一系列运行步骤。

注3：图16-23的“(1)开始条件~地址~数据”说明步骤①~⑦。

注4：图16-23的“(2)地址~数据~数据”说明步骤③~⑫。

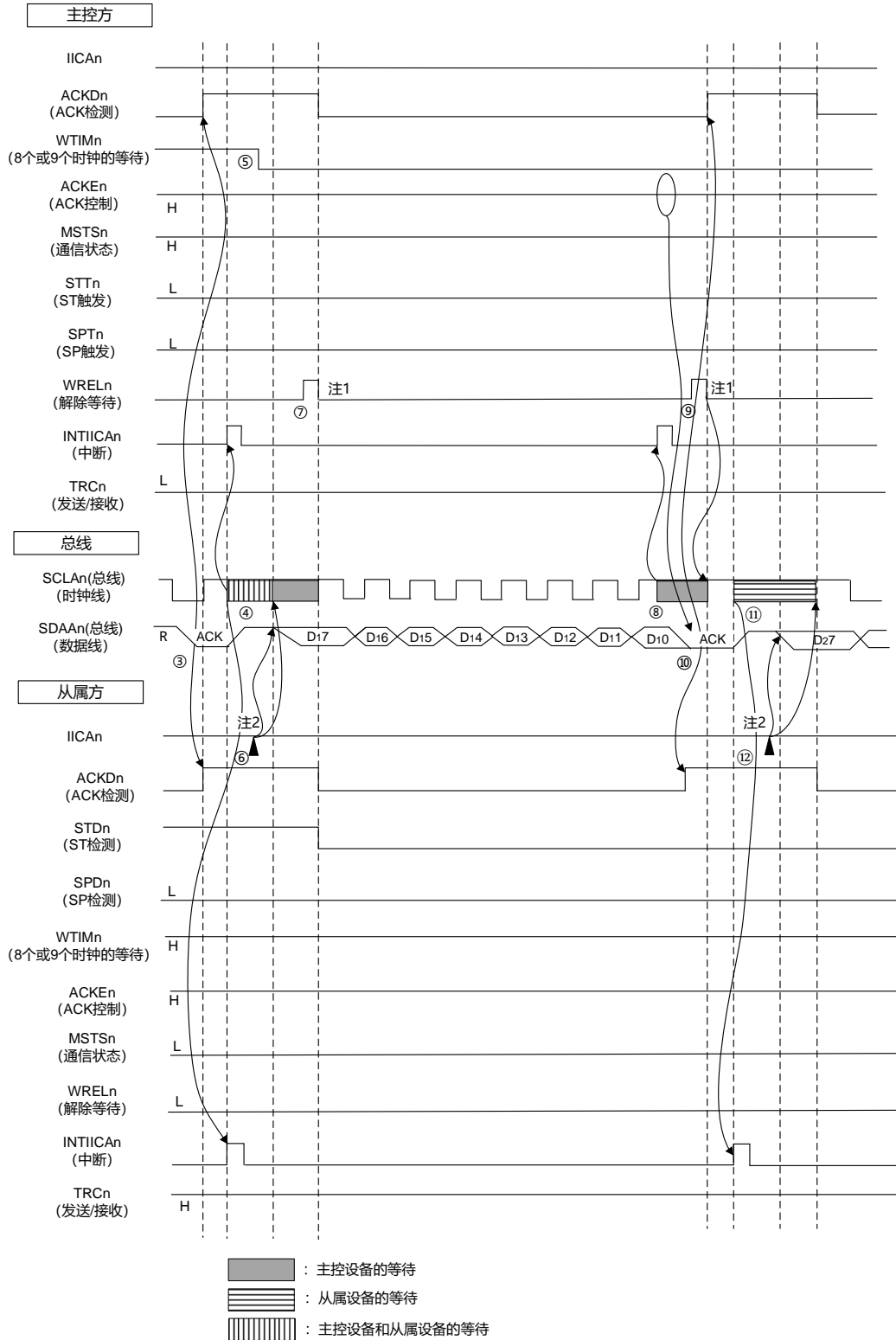
注5：图16-23的“(3)数据~数据~停止条件”说明步骤⑧~⑱。

注6：n=0。

图16-23 从属设备&主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待) (2/3)

(2) 地址~数据~数据



注1: 要解除主控方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

注2: 要解除从属方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

图16-23的“(2)地址~数据~数据”的③~⑫的说明如下:

③在从属方, 如果接收地址和本地站地址 (SVAn的值) 相同注, 就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。

④主控方在第9个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 地址匹配中断) 注。

⑤主控方将等待时序改为第8个时钟 (WTIMn=0)。

⑥从属方给IICA移位寄存器n (IICAn) 写发送数据, 解除从属方的等待。

⑦主控方解除等待 (WRELn=1), 开始来自从属设备的数据传送。

⑧主控方在第8个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。因为主控方的ACKEn位为“1”, 所以通过硬件给从属方发送ACK。

⑨主控方读接收数据, 解除等待 (WRELn=1)。

⑩从属方在第9个时钟的上升沿检测到ACK (ACKDn=1)。

⑪从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。

⑫如果从属方给IICAn寄存器写发送数据, 就解除从属方的等待, 开始从属方到主控方的数据传送。

注1: 如果发送的地址和从属地址不同, 从属方就不给主控方返回ACK (NACK: SDAAn=1), 并且不产生INTIICAn中断 (地址匹配中断), 也不进入等待状态。但是, 主控方对于ACK和NACK都产生INTIICAn中断 (地址发送结束中断)。

注2: 图16-23的①~⑱是通过I2C总线进行数据通信的一系列运行步骤。

注3: 图16-23的“(1)开始条件~地址~数据”说明步骤①~⑦。

注4: 图16-23的“(2)地址~数据~数据”说明步骤③~⑫。

注5: 图16-23的“(3)数据~数据~停止条件”说明步骤⑧~⑱。

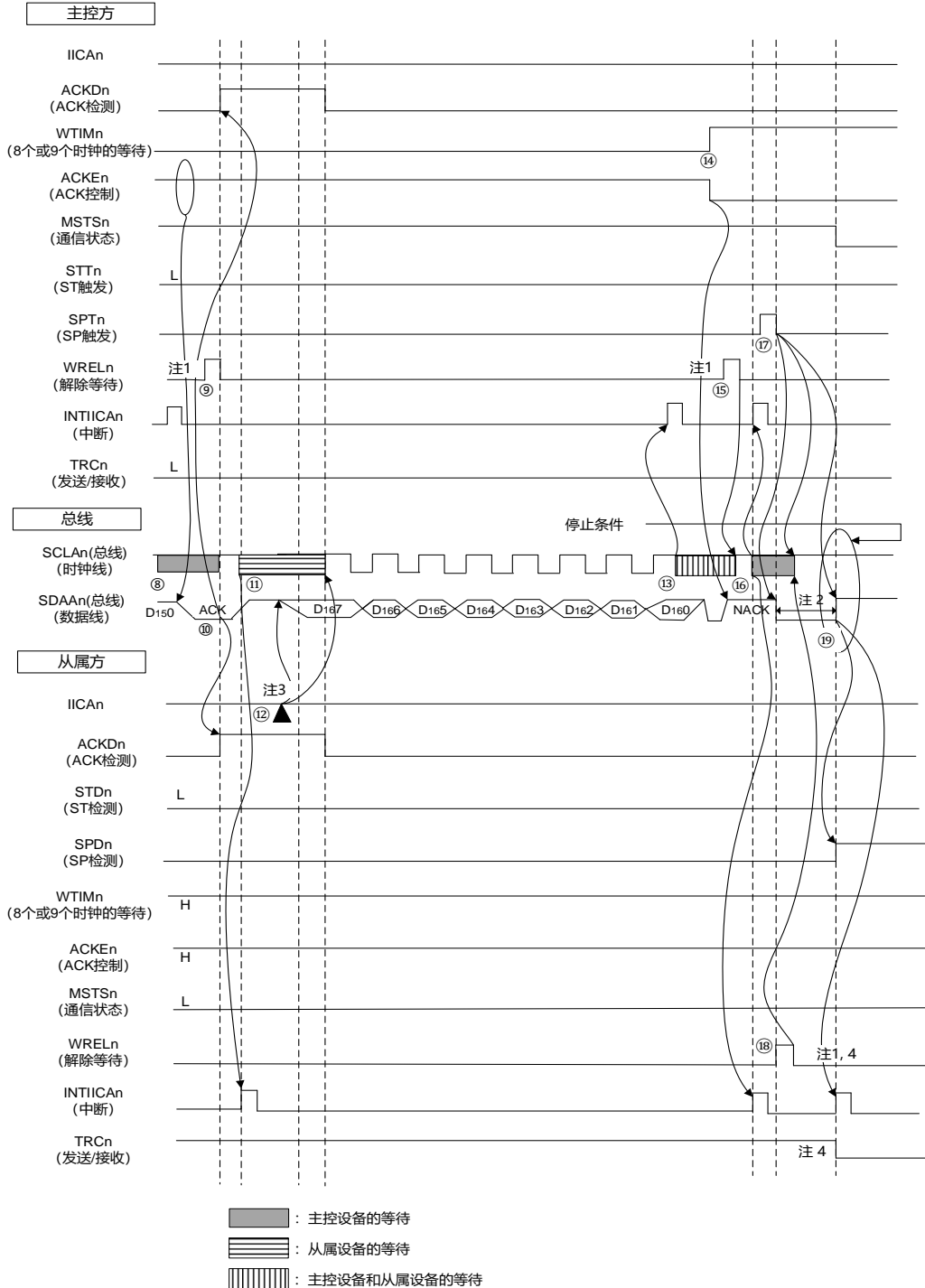
注6: n=0。



图16-23 从属设备&主控设备的通信例子

(主控设备：选择8个/9个时钟的等待，从属设备：选择9个时钟的等待) (3/3)

(3) 数据~数据~停止条件



注1: 要解除等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注2: 在发行停止条件后, 从SCLAn引脚信号上升到生成停止条件的的时间, 在设定为标准模式时至少为4.0us, 在设定为快速模式时至少为0.6us。

注3: 要解除从属方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

注4: 在从属方的发送期间, 如果通过WRELn位的置位来解除等待, 就清除TRCn位。

图16-23的“(3)数据~数据~停止条件”的⑧~⑲的说明如下：

- ⑧. 主控方在第8个时钟的下降沿进入等待状态 (SCLAn=0)，并且产生中断 (INTIICAn：传送结束中断)。因为主控方的ACKEn位为“0”，所以通过硬件给从属方发送ACK。
- ⑨. 主控方读接收数据，解除等待 (WRELn=1)。
- ⑩. 从属方在第9个时钟的上升沿检测到ACK (ACKDn=1)。
- ⑪. 从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0)，并且产生中断 (INTIICAn：传送结束中断)。
- ⑫. 如果从属方给IICA移位寄存器n (IICAn) 写发送数据，就解除从属方的等待，开始从属方到主控方的数据传送。
- ⑬. 主控方在第8个时钟的下降沿产生中断 (INTIICAn：传送结束中断)，并且进入等待状态 (SCLAn=0)。因为进行ACK控制 (ACKEn=1)，所以此阶段的总线数据线变为低电平 (SDAAn=0)。
- ⑭. 主控方设定为NACK应答 (ACKEn=0)，并且将等待时序改为第9个时钟 (WTIMn=1)。如果主控方解除等待 (WRELn=1)，从属方就在第9个时钟的上升沿检测到NACK (ACKDn=0)。
- ⑮. 主控方和从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0)，并且都产生中断 (INTIICAn：传送结束中断)。
- ⑯. 如果主控方发行停止条件 (SPTn=1)，就清除总线数据线 (SDAAn=0) 并且解除主控方的等待。此后，主控方处于待机状态，直到将总线时钟线置位 (SCLAn=1) 为止。
- ⑰. 从属方在确认NACK后停止发送，为了结束通信，解除等待 (WRELn=1)。如果解除从属方的等待，就将总线时钟线置位 (SCLAn=1)。
- ⑱. 如果主控方确认到总线时钟线被置位 (SCLAn=1)，就在经过停止条件准备时间后将总线数据线置位 (SDAAn=1) 然后发行停止条件 (通过SCLAn=1使SDAAn从“0”变为“1”)。如果生成停止条件，从属方就检测到停止条件，并且产生中断 (INTIICAn：停止条件中断)。

# 第17章 串行外围接口控制器(SSP/SPI)

## 17.1 概述

串行外围设备接口（SPI）是一个工作于全双工模式的同步串行数据通讯协议。设备可工作在主/从模式，利用4线双向接口相互通讯。当从一个外围设备接收数据时，SPI执行串--并的转换，而在数据向外围设备发送时执行并-串的转变。该SPI控制器可以配置为主设备或从设备。

## 17.2 特性

- ◆ 支持主机或从机模式。
- ◆ 全双工。
- ◆ 可配置发送的位长度（4bit-16bit）。
- ◆ MSB 优先发送/接收。
- ◆ 内置一个接收 Buffer 和一个发送 Buffer

## 17.3 寄存器映射

（SSP0基地址= 0x4006\_3000）

RO：只读；WO：只写；R/W：读写。

寄存器	偏移量	读/写	描述	复位值
CON	0x000	R/W	SSP控制寄存器	0x0
STAT	0x004	RO	SSP 状态寄存器	0x3
DAT	0x008	R/W	SSP 数据寄存器	0x0
CLK	0x00C	R/W	SSP 时钟控制寄存器	0x0
IMSC	0x010	R/W	SSP 中断使能寄存器	0x0
RIS	0x014	RO	SSP 中断源状态寄存器	0x8
MIS	0x018	RO	SSP 已使能中断状态寄存器	0x0
ICLR	0x01C	WO	SSP 中断清零寄存器	0x0
CSCR	0x028	R/W	SSP软件片选信号寄存器	0x0

## 17.4 寄存器说明

### 17.4.1 SSP控制寄存器 (CON)

位	符号	描述	复位值
31:12	-	保留	-
11	LBM	回环模式使能位 0: 正常工作模式 1: 回环模式, 串行输入接到串行输出	0
10	SSPEN	SSP使能位 0: 禁止 1: 使能	0
9	MS	主机/从机模式选择位 0: 主机模式 1: 从机模式	0
8	-	保留	0
7	CPH	时钟相位控制位 0: SSP在第一个时钟边沿采样数据 1: SSP在第二个时钟边沿采样数据	0
6	CPO	时钟输出极性选择位 0: SPI_CLK在空闲时为低电平 1: SPI_CLK在空闲时为高电平	0
5:4	FRF	帧格式 0x0: SPI-兼容帧格式 0x1: TISS-兼容帧格式 0x2: Microwire-兼容帧格式 0x3: 保留	0x0
3:0	DSS	数据传输长度选择位 0x0: 保留 0x1: 保留 0x2: 保留 0x3: 4位长度 0x4: 5位长度 0x5: 6位长度 0x6: 7位长度 0x7: 8位长度 0x8: 9位长度 0x9: 10位长度 0xA: 11位长度 0xB: 12位长度 0xC: 13位长度 0xD: 14位长度 0xE: 15位长度 0xF: 16位长度	0x0

## 17.4.2 SSP状态寄存器 (STAT)

位	符号	描述	复位值
31:5	-	保留	-
4	BSY	忙标志位, 只读 0: SSP空闲 1: SSP正在发送/接收数据或发送Buffer已写入数据	0
3:0	-	保留	0x3

## 17.4.3 SSP数据寄存器 (DAT)

位	符号	描述	复位值
31:16	-	保留	-
15:0	DATA	写数据到该寄存器, 当总线上没有数据在发送时, 该数据会写入到发送寄存器被发送出去; 当总线上有数据在发送时, 该数据会存入Buffer并在上一次数据传输完成后发送。发送时间的间隔最短为3个SSPCLK时钟。 当数据长度小于16位时, 需右对齐。 读该寄存器, 读到的是最近接收到的数据, 当数据长度小于16位时, 需右对齐。	0x0

## 17.4.4 SSP时钟控制器 (CLK)

位	符号	描述	复位值
31:16	-	保留	-
15:8	M	SSPCLK = PCLK / ((M+1) × N) N为2-254的偶数	0x0
7:0	N		0x0

## 17.4.5 SSP中断使能寄存器 (IMSC)

位	符号	描述	复位值
31:4	-	保留	-
3	TXIM	发送Buffer中断使能位 0: 禁止发送Buffer中无数据中断 1: 使能发送Buffer中无数据中断	0
2	RXIM	接收Buffer中断使能位 0: 禁止接收Buffer收到数据中断 1: 使能接收Buffer收到数据中断	0
1	RTIM	接收Buffer定时器溢出中断使能位 0: 禁止接收Buffer定时器溢出中断 1: 使能接收Buffer定时器溢出中断 (溢出时间为: 32×SSPCLK)	0
0	RORIM	接收Buffer溢出中断使能位 0: 禁止接收Buffer溢出中断 1: 使能接收Buffer溢出中断	0

### 17.4.6 SSP中断源状态寄存器 (RIS)

位	符号	描述	复位值
31:4	-	保留	-
3	TXRIS	当发送 Buffer 无数据时或发送 Buffer 中数据被发送后该位置位 (当发送 Buffer 中有数据时, 该位自动清零)	1
2	RXRIS	当接收 Buffer 收到数据时该位置位 (当接收 Buffer 没有数据时或接收 Buffer 中数据被读取后, 该位自动清零)	0
1	RTRIS	当接收 Buffer 收到数据, 且超时未被读取时该位置位 (读取数据寄存器或写 ICLR 寄存器清零)	0
0	RORRIS	当接收 Buffer 收到数据且未被读取期间, 又接收到一帧数据时该位置位, 新数据将会丢失 (写 ICLR 寄存器清零)	0

### 17.4.7 SSP已使能中断状态寄存器 (MIS)

位	符号	描述	复位值
31:4	-	保留	-
3	TXMIS	= TXIM & TXRIS	0
2	RXMIS	= RXIM & RXRIS	0
1	RTMIS	= RTIM & RTRIS	0
0	RORMIS	= RORIM & RORRIS	0

### 17.4.8 SSP中断清零寄存器 (ICLR)

位	符号	描述	复位值
31:2	-	保留	-
1	RTIC	写1清零RTRIS标志位	0
0	RORIC	写1清零RORRIS标志位	0

### 17.4.9 SSP软件片选信号寄存器 (CSCR)

位	符号	描述	复位值
31:5	-	保留	-
4	SPH	从机片选信号 (SPI作从机时有效) 0: 每帧数据传输完成后片选信号拉高 1: 最后一帧数据传输完成后片选信号拉高	0
3	SWCS	主机模式下软件片选信号控制位 0: 输出低电平 1: 输出高电平	0
2	SWSEL	主机模式下片选信号选择 0: 片选信号由SPI模块自动控制 1: 片选信号由SWCS位控制	0
1:0	-	保留	-

# 第18章 增强型DMA

## 18.1 DMA的功能

DMA是不使用CPU而在存储器之间进行数据传送的功能。通过外围功能中断启动DMA进行数据传送。当DMA和CPU同时访问FLASH, SRAM0, SRAM1或外围模块中的同一单元时, 其总线使用权高于CPU。当DMA和CPU分别访问FLASH, SRAM0, SRAM1或外围模块中的不同单元时, 两者互不干扰, 可以并行执行。

DMA的规格如表18-1所示。

表18-1 DMA的规格(1/2)

项目		规格
启动源		最大8个启动源
可分配的控制数据		8组
可传送的地址空间	地址空间	全地址范围空间
	源	全地址范围空间可选
	目标	全地址范围空间可选
最大传送次数	正常模式	65535次
	重复模式	65535次
最大传送块大小	正常模式 (8位传送)	65535字节
	正常模式 (16位传送)	131070字节
	正常模式 (32位传送)	262140字节
	重复模式	65535字节
传送单位		8位/16位/32位
传送模式	正常模式	在进行DMACTj寄存器从“1”变为“0”的传送后结束。
	重复模式	在DMACTj寄存器从“1”变为“0”的传送结束后, 对重复区的地址进行初始化, 在将DMRLDj寄存器的值重新加载到DMACTj寄存器后继续传送。
地址控制	正常模式	固定或者递增
	重复模式	固定或者递增非重复区的地址。
启动源的优先级		参照“表18-4 DMA启动源和向量地址

表18-1 DMA的规格(2/2)

项目		规格
中断请求	正常模式	在进行DMACTj寄存器从“1”变为“0”的数据传送时，向CPU请求启动源的中断，并进行中断处理。
	重复模式	在DMACRj寄存器的RPTINT位为“1”允许产生中断，并且进行DMACTj寄存器从“1”变为“0”的数据传送时，向CPU请求启动源的中断，并进行中断处理。
传送开始		如果将DMAENi寄存器的DMAENi0~DMAENi7位置“1”（允许启动），就在每次发生DMA启动源时开始传送数据。
传送停止	正常模式	<ul style="list-style-type: none"> <li>•将DMAENi0~DMAENi7位置“0”（禁止启动）。</li> <li>•当DMACTj寄存器从“1”变为“0”的数据传送结束时</li> </ul>
	重复模式	<ul style="list-style-type: none"> <li>•将DMAENi0~DMAENi7位置“0”（禁止启动）。</li> <li>•当RPTINT位为“1”（允许产生中断）并且DMACTj寄存器从“1”变为“0”的数据传送结束时</li> </ul>

注1：在深度睡眠模式，因为闪存停止运行，所以不能作为DMA传送源。

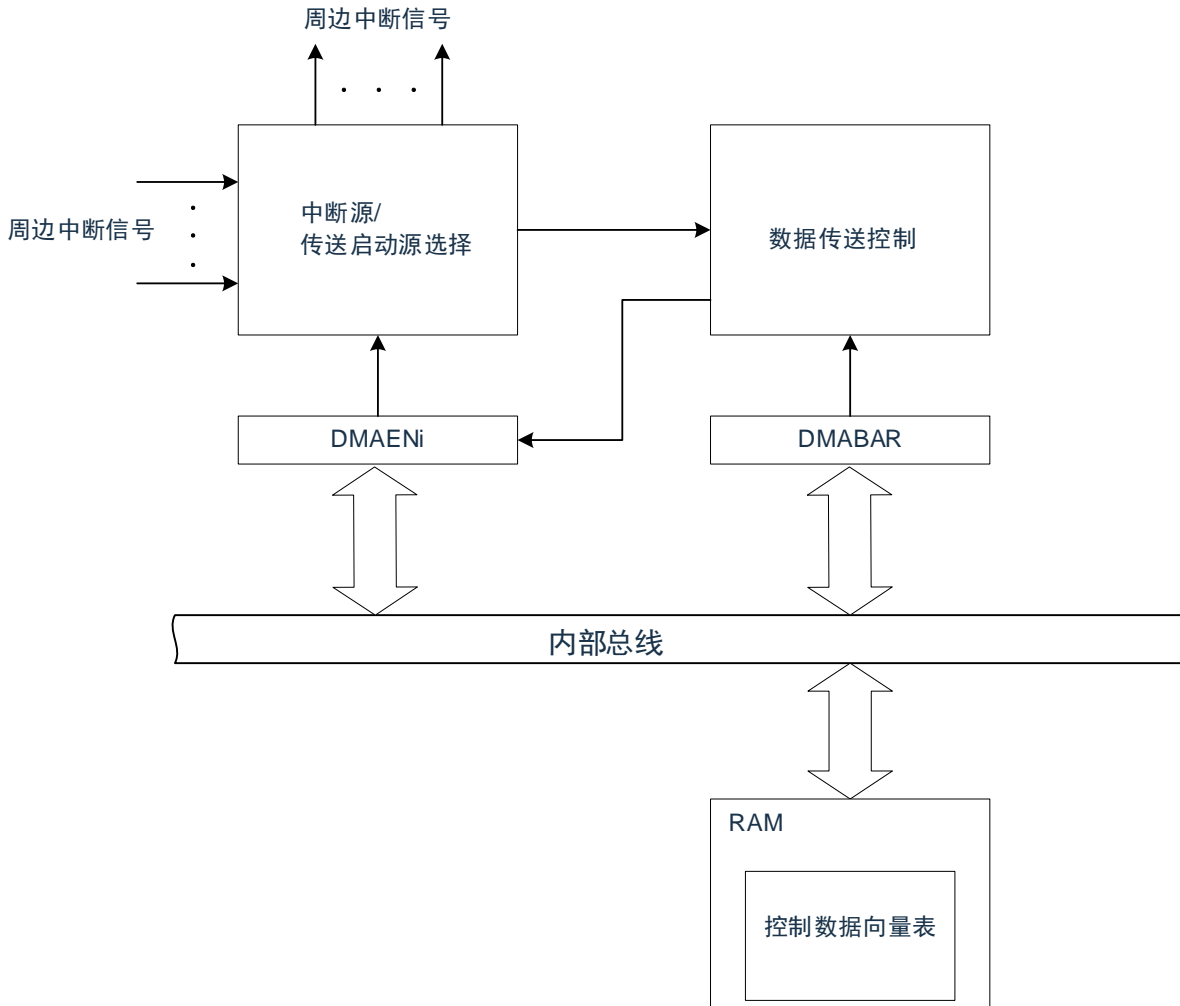
注2：i=0, j=0~7。



## 18.2 DMA的结构

DMA的框图如图18-1所示

图18-1 DMA的框图



## 18.3 寄存器映射

(以下寄存器基地址 = 0x40020810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER10	0x000	R/W	外围允许寄存器10	0x00

(以下寄存器基地址 = 0x40005000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
DMAEN0	0x000	R/W	DMA启动允许寄存器0	0x00
DMABAR	0x008	R/W	DMA基址寄存器	0x00

(以下寄存器基地址 = 0x40020D00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
DMATGS	0x000	R/W	DMA触发源选择寄存器	0x00
TGSEN0	0x004	R/W	触发源信号触发DMA使能寄存器0	0x00
TGSEN1	0x008	R/W	触发源信号触发DMA使能寄存器1	0x00
TGSEN2	0x00C	R/W	触发源信号触发DMA使能寄存器2	0x00

DMA的控制数据如表18-2所示。

DMA的控制数据分配在RAM的DMA控制数据区。通过DMABAR寄存器设置DMA控制数据区和包含DMA向量表区（保存控制数据的起始地址）的144字节区域。

表18-2 DMA的控制数据

寄存器名	符号
DMA控制寄存器j	DMACRj
DMA块大小寄存器j	DMBLSj
DMA传送次数寄存器j	DMACTj
DMA传送次数重加载寄存器j	DMRLDj
DMA源地址寄存器j	DMSARj
DMA目标地址寄存器j	DMDARj

注: j=0~7。

## 18.4 控制DMA的寄存器

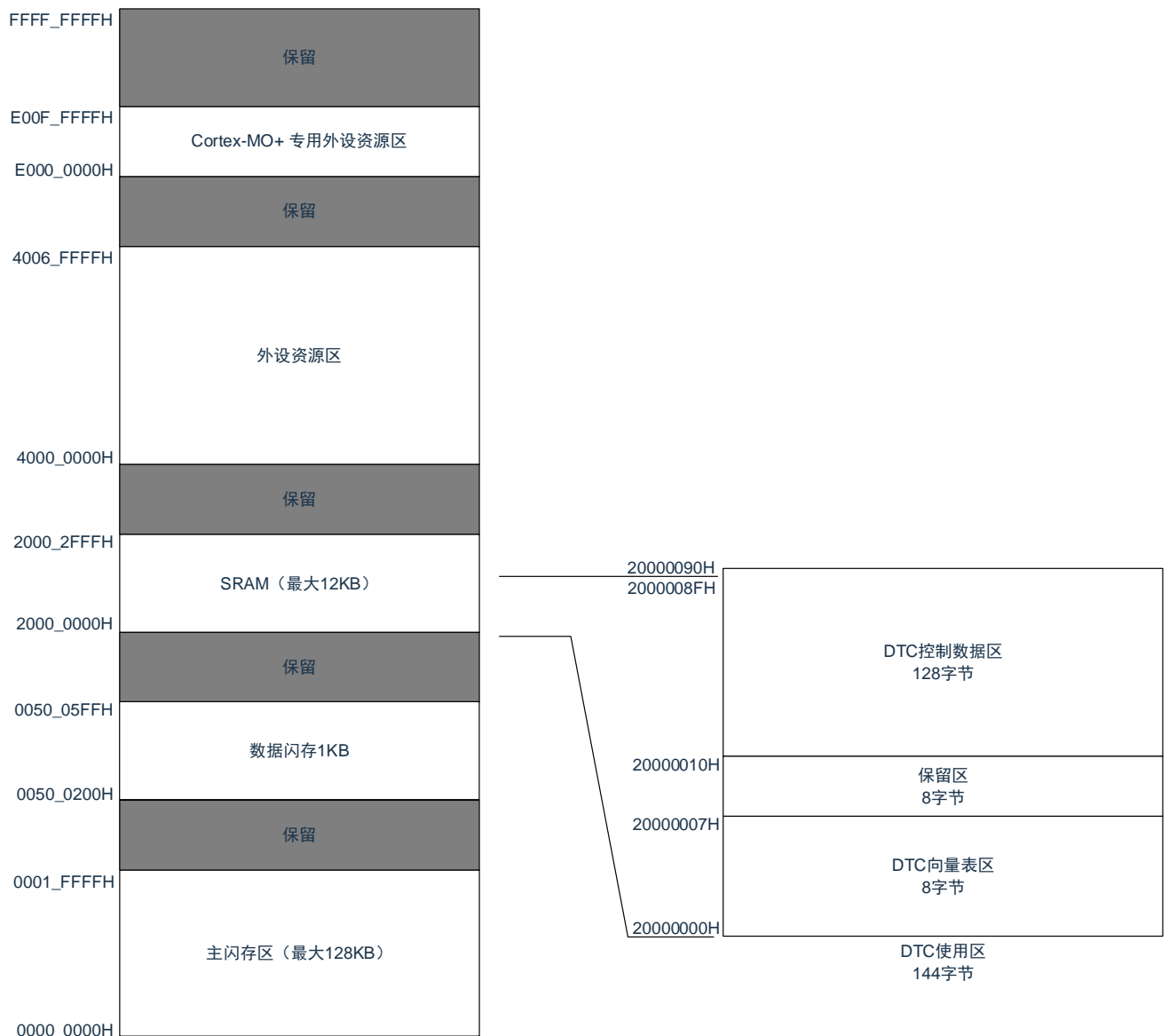
### 18.4.1 DMA控制数据区和DMA向量表区的分配

通过DMABAR寄存器将分配DMA的控制数据和向量表的144字节区域设置到RAM区。

DMABAR寄存器的设置值为“20000000H”时的存储器映像例子如图18-2所示。

DMA控制数据区的144字节中DMA不使用的空间能用作RAM。

图18-2 DMABAR寄存器的设置值为“20000000H”时的存储器映像例子



### 18.4.2 控制数据的分配

从起始地址开始，按照DMACR<sub>j</sub>、DMBLS<sub>j</sub>、DMACT<sub>j</sub>、DMRLD<sub>j</sub>、DMSAR<sub>j</sub>、DMDAR<sub>j</sub> (j=0~7) 寄存器的顺序分配控制数据。

起始地址由DMABAR寄存器设置，低8位由各启动源分配的向量表分别设置。

控制数据的分配如图18-3所示。

注1：必须在对应的DMAEN<sub>i</sub> (i=0) 的DMAEN<sub>i0</sub>~DMAEN<sub>i7</sub>位为“0”（禁止启动）时更改DMACR<sub>j</sub>、DMBLS<sub>j</sub>、DMACT<sub>j</sub>、DMRLD<sub>j</sub>、DMSAR<sub>j</sub>、DMDAR<sub>j</sub>寄存器的数据。

注2：不能通过DMA传送进行DMACR<sub>j</sub>、DMBLS<sub>j</sub>、DMACT<sub>j</sub>、DMRLD<sub>j</sub>、DMSAR<sub>j</sub>和DMDAR<sub>j</sub>的存取。

图18-3 控制数据的分配(DMABAR设置为20000000H)

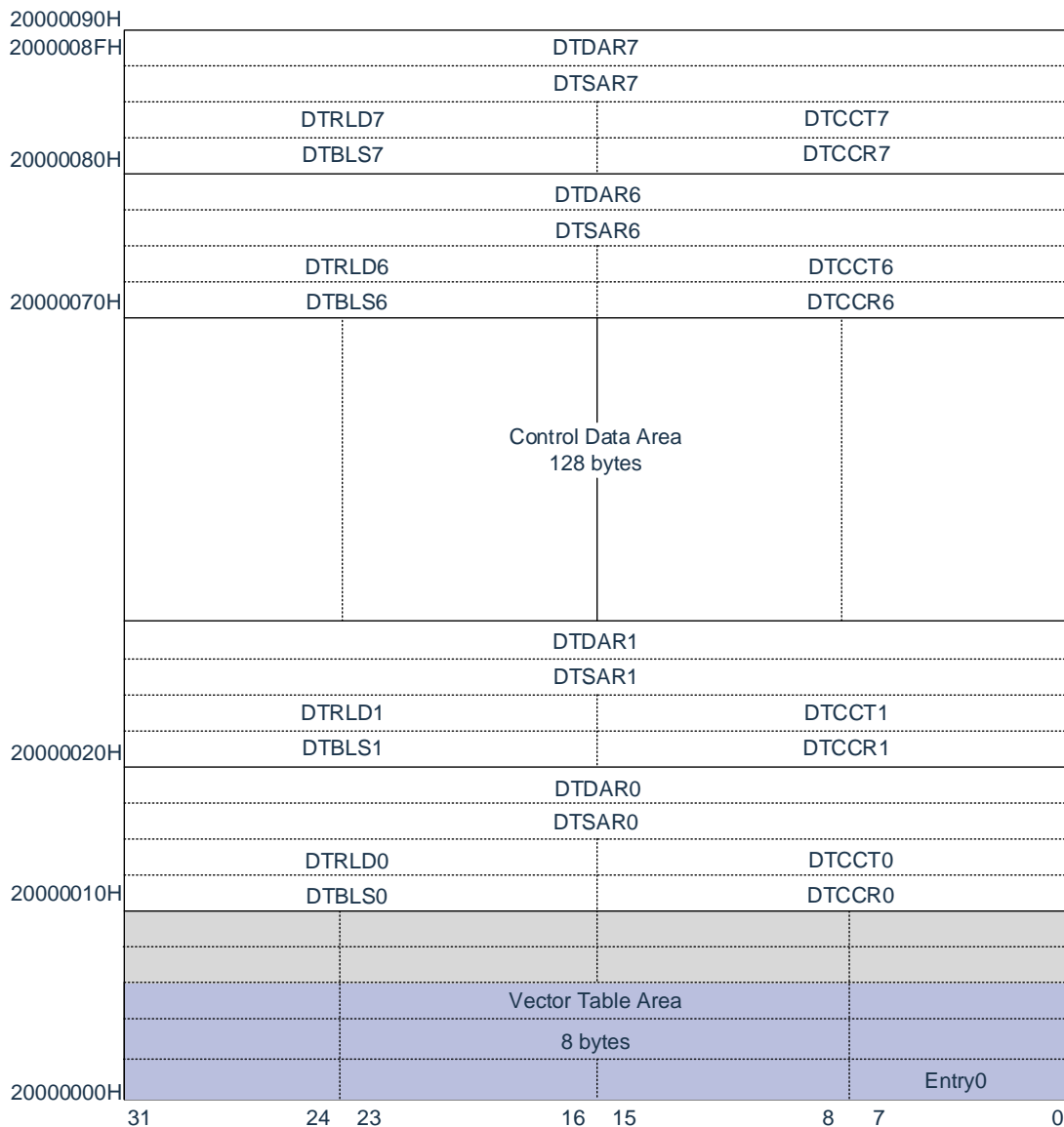


表18-3 控制数据的起始地址

j	地址
7	baseaddr+80H
6	baseaddr+70H
5	baseaddr+60H
4	baseaddr+50H
3	baseaddr+40H
2	baseaddr+30H
1	baseaddr+20H
0	baseaddr+10H

注：baseaddr：DMABAR寄存器的设置值。

### 18.4.3 向量表

DMA一旦启动，就通过从各启动源分配的向量表读取的数据来决定控制数据，读被分配在DMA控制数据区的控制数据。

DMA启动源和向量地址如表18-4所示。各启动源的向量表有1字节，保存“00H”~“07H”的数据，从8组的控制数据中选择1组数据。向量地址的高24位由DMABAR寄存器设置，低8位被分配了对应启动源的“00H”~“17H”。

注：必须在对应的DMAENi (i=0) 寄存器的DMAENi0位为“0”（禁止启动）时更改设置在向量表中的DMA控制数据区的起始地址。

图18-4 控制数据的起始地址和向量表

DMABAR寄存器的设置值为“20000000H”的情况（例）

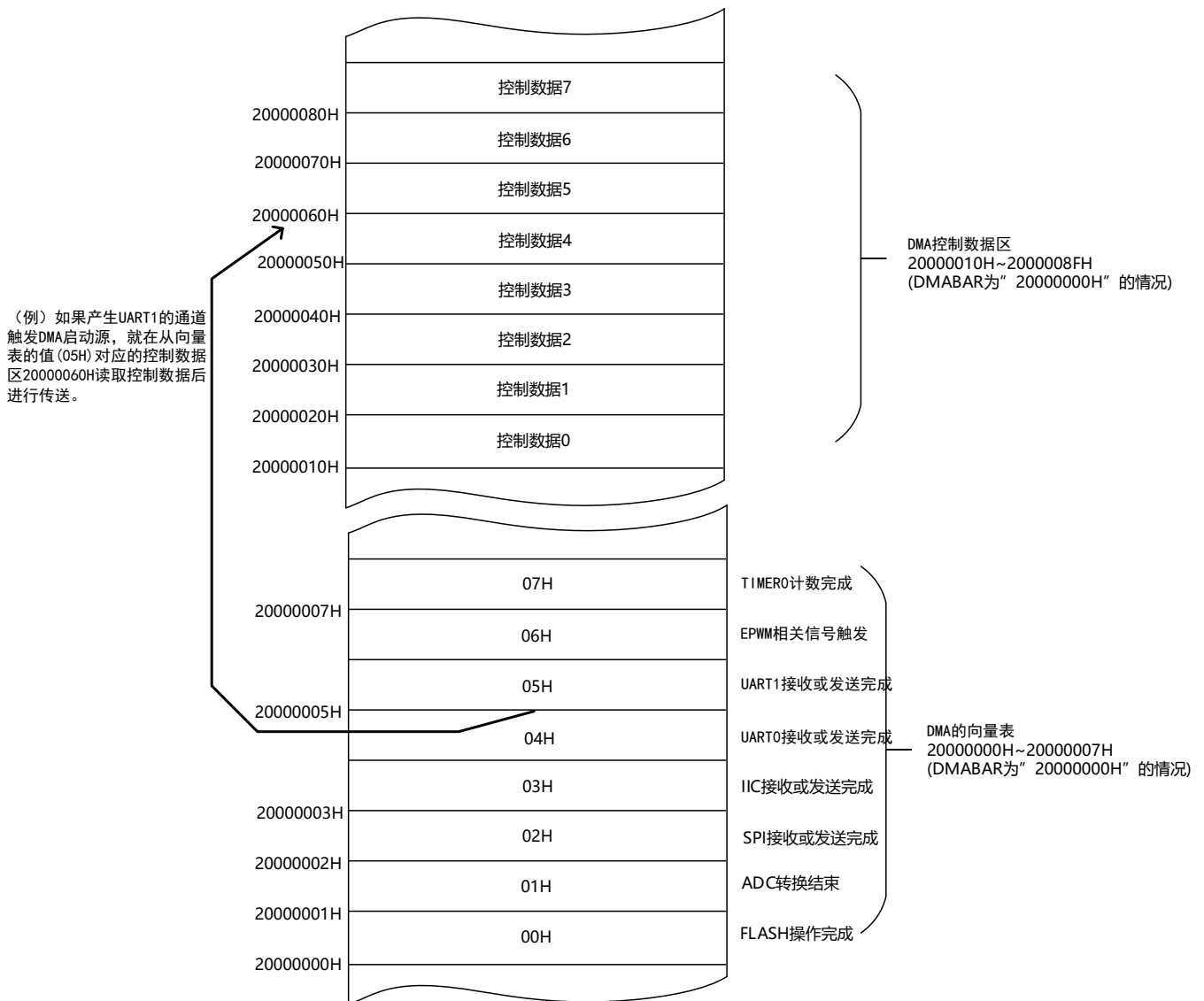



表18-4 DMA启动源和向量地址

DMA启动源（中断请求发生源）	源号	向量地址	优先级
Flash读写擦除结束	0	DMABAR寄存器的设置地址+00H	高  低
ADC转换结束	1	DMABAR寄存器的设置地址+01H	
SPI	2	DMABAR寄存器的设置地址+02H	
IIC	3	DMABAR寄存器的设置地址+03H	
UART0	4	DMABAR寄存器的设置地址+04H	
UART1	5	DMABAR寄存器的设置地址+05H	
EPWM	6	DMABAR寄存器的设置地址+06H	
TIMER0	7	DMABAR寄存器的设置地址+07H	

## 18.4.4 外围允许寄存器10 (PER10)

PER10寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。要使用DMA时，必须将bit3 (DMAEN) 置“1”。

通过8位存储器操作指令设置PER10寄存器。在产生复位信号后，此寄存器的值变为“00H”。

外围允许寄存器10 (PER10)

位	符号	描述	复位值
7:4	--	保留	0x0
3	DMAEN	提供DMA模块输入时钟的控制 0: 停止提供输入时钟，DMA不能运行 1: 提供输入时钟，DMA能运行	0
2:0	--	保留	0x0

## 18.4.5 DMA触发源选择寄存器 (DMATGS)

位	符号	描述	复位值
31:8	-	保留	-
7	TIMER0_SEL	TIMER0请求DMA信号源选择 0: TIMER0中断信号 1: TIMER0触发信号	0
6	EPWM_SEL	EPWM请求DMA信号源选择 0: EPWM中断信号 1: EPWM触发信号	0
5	UART1_SEL	UART1请求DMA信号源选择 0: UART1中断信号 1: UART1触发信号	0
4	UART0_SEL	UART0请求DMA信号源选择 0: UART0中断信号 1: UART0触发信号	0
3	-	保留	-
2	SPI_SEL	SPI请求DMA信号源选择 0: SPI中断信号 1: SPI触发信号	0
1	ADC_SEL	ADC请求DMA信号源选择 0: ADC中断信号 1: ADC触发信号	0
0	-	保留	-



### 18.4.6 触发源信号触发DMA使能寄存器0 (TGSEN0)

位	符号	描述	复位值
31:28	-	保留	-
27	ADCOMPEN	ADC数字比较完成请求DMA使能位 0: 请求DMA禁止 1: 请求DMA使能	0
26:0	ADCHEFn (n=0~26)	ADC通道转换完成请求DMA使能位 0: 请求DMA禁止 1: 请求DMA使能	0x0

### 18.4.7 触发源信号触发DMA使能寄存器1 (TGSEN1)

位	符号	描述	复位值
31:18	-	保留	-
17	PWMPTG1EN	PWM比较点1请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
16	PWMPTG0EN	PWM比较点0请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
15:8	PWMZIFENn (n=0~7)	PWM零点请求DMA使能位 0: 请求DMA禁止 1: 请求DMA使能	0x0
7:0	PWMPIFENn (n=0~7)	PWM周期点请求DMA使能位 0: 请求DMA禁止 1: 请求DMA使能	0x0

### 18.4.8 触发源信号触发DMA使能寄存器2 (TGSEN2)

位	符号	描述	复位值
31:18	-	保留	-
15	UART1BUFEN	UART1发送数据缓存空请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
14	UART1TXOEN	UART0发送完成请求DMA使能 (此时数据已经发送完成) 0: 请求DMA禁止 1: 请求DMA使能	0
13	UART1TXDEN	UART0发送完成请求DMA使能 (最后一位数据从缓存发出) 0: 请求DMA禁止 1: 请求DMA使能	0
12	UART1RXEN	UART1接收完成请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
11	UART0BUFEN	UART0发送数据缓存空请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
10	UART0TXOEN	UART0发送完成请求DMA使能 (此时数据已经发送完成) 0: 请求DMA禁止 1: 请求DMA使能	0
9	UART0TXDEN	UART0发送完成请求DMA使能 (最后一位数据从缓存发出) 0: 请求DMA禁止 1: 请求DMA使能	0
8	UART0RXEN	UART0接收完成请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
7:4	-	保留	-
3	SPITXBUSY	SPI总线空闲请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
2	SPITXBUFEN	SPI数据缓存空请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
1	SPITXEN	SPI发送完成请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0
0	SPIRXEN	SPI接收完成请求DMA使能 0: 请求DMA禁止 1: 请求DMA使能	0

## 18.4.9 DMA控制寄存器j (DMACRj) (j=0~7)

DMACRj寄存器控制DMA的运行模式。

地址：参照“18.4.2 控制数据的分配”。复位后：不定值 R/W

位	符号	描述	复位值
15:8	-	保留	不定值
7: 6	SZ	传送数据长度的选择 00: 8位 01: 16位 10: 32位 11: 禁止设置	
5	RPTINT	重复模式中中断的允许/禁止 <sup>注1</sup> 0: 禁止发生中断。 1: 允许发生中断。	
4	CHNE	链传送的允许/禁止 <sup>注2</sup> 0: 禁止链传送。 1: 允许链传送。	
3	DAMOD	传送目标地址的控制 <sup>注3</sup> 0: 固定 1: 递增	
2	SAMOD	传送源地址的控制 <sup>注4</sup> 0: 固定 1: 递增	
1	RPTSEL	重复区的选择 <sup>注5</sup> 0: 传送目标为重复区。 1: 传送源为重复区。	
0	MODE	传送模式的选择 0: 正常模式 1: 重复模式	

注1：在MODE位为“0”（正常模式）时，RPTINT位的设置无效。

注2：必须将DMACR7寄存器的CHNE位置“0”（禁止链传送）。

注3：在MODE位为“1”（重复模式）并且RPTSEL位为“0”（传送目标为重复区）时，DAMOD位的设置无效。

注4：在MODE位为“1”（重复模式）并且RPTSEL位为“1”（传送源为重复区）时，SAMOD位的设置无效。

注5：在MODE位为“0”（正常模式）时，RPTSEL位的设置无效。

注6：不能通过DMA传送进行DMACRj寄存器的存取。

## 18.4.10 DMA块大小寄存器j (DMBLSj) (j=0~7)

此寄存器设置1次启动传送数据的块大小。

地址：参照“18.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMBLSj	设置1次启动传送数据的块大小	不定值

DMBLSj	传送块大小		
	8位传送	16位传送	32位传送
00H	禁止设置	禁止设置	禁止设置
01H	1字节	2字节	4字节
02H	2字节	4字节	8字节
03H	3字节	6字节	12字节
⋮	⋮	⋮	⋮
FDH	253字节	506字节	1012字节
FEH	254字节	508字节	1016字节
FFH	255字节	510字节	1020字节
⋮	⋮	⋮	⋮
FFFFH	65535字节	131070字节	262140字节

注1：不能通过DMA传送进行DMBLSj寄存器的存取。

## 18.4.11 DMA传送次数寄存器j (DMACTj) (j=0~7)

此寄存器设置DMA的数据传送次数。每当启动1次DMA传送就减1。

地址：参照“18.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMACTj	设置DMA的数据传送次数，每当启动1次DMA传动就减1	不定值

DMACTj	传送次数
00H	禁止设置
01H	1次
02H	2次
03H	3次
⋮	⋮
⋮	⋮
⋮	⋮
FDH	253次
FEH	254次
FFH	255次
⋮	⋮
⋮	⋮
⋮	⋮
FFFFH	65535次

注1：不能通过DMA传送进行DMACTj寄存器的存取。

## 18.4.12 DMA传送次数重加载寄存器j (DMRLDj) (j=0~7)

此寄存器设置重复模式中的传送次数寄存器的初始值。在重复模式中，因为将此寄存器的值重新加载到DMACT寄存器，所以设置值必须和DMACT寄存器的初始值相同。

地址：参照“18.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMRLDj	设置重复模式中的传送次数寄存器的初始值	不定值

注：不能通过DMA传送进行DMRLDj寄存器的存取。

### 18.4.13 DMA源地址寄存器j (DMSARj) (j=0~7)

此寄存器指定数据传送时的传送源地址。

当DMACRj寄存器的SZ位为“01”（16位传送）时，忽视最低位而作为偶地址进行处理。

当DMACRj寄存器的SZ位为“10”（32位传送）时，忽视低2位而作为word地址进行处理。

地址：参照“18.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
31:0	DMSARj	设置数据传送时的传送源地址	不定值

注：不能通过DMA传送进行DMSARj寄存器的存取

### 18.4.14 DMA目标地址寄存器j (DMDARj) (j=0~7)

此寄存器指定数据传送时的传送目标地址。

当DMACRj寄存器的SZ位为“01”（16位传送）时，忽视最低位而作为偶地址进行处理。

当DMACRj寄存器的SZ位为“10”（32位传送）时，忽视低2位而作为word地址进行处理。

地址：参照“18.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
31:0	DMDARj	设置数据传送时的目标地址	不定值

注：不能通过DMA传送进行DMDARj寄存器的存取。

## 18.4.15 DMA启动允许寄存器i (DMAEN0)

这是控制允许或者禁止通过各中断源启动DMA的8位寄存器。中断源和DMAEN0位的对应如表18-5所示。能通过8位存储器操作指令设置DMAEN寄存器。

注1：必须在不产生对应该位的启动源的位置更改DMAENi0~DMAENi7位。

注2：不能通过DMA传送进行DMAENi寄存器的存取。

注3：分配的功能因产品而不同，必须将没有分配功能的位置“0”。

位	符号	描述	复位值
7:0	DMAEN0n (n=0~7)	DMA启动的允许in <sup>注1</sup> (n=0~7) 0: 禁止启动。 1: 允许启动。	0x0

注1：根据传送结束中断的发生条件，DMAENin位变为“0”（禁止启动）。n=0,1,2,3,4,5,6,7。

表18-5 中断源和DMAEN00~DMAEN07位的对应

寄存器	DMAENi7位	DMAENi6位	DMAENi5位	DMAENi4位	DMAENi3位	DMAENi2位	DMAENi1位	DMAENi0位
DMAEN0	TIMER0溢出	EPWM转换完成	UART1发送或接收完成	UART0发送或接收完成	IIC发送或接收完成	SPI发送或接收完成	ADC转换结束位	Flash擦除/写入结束

注1：必须将没有分配功能的位置“0”。

注2：i=0



## 18.4.16 DMA基址寄存器 (DMABAR)

这是32位寄存器，设置保存DMA控制数据区起始地址的向量地址以及DMA控制数据区的地址。

注1：必须在将全部的DMA启动源设置为禁止启动的状态下更改DMABAR寄存器。

注2：只能改写1次DMABAR寄存器。

注3：不能通过DMA传送进行DMABAR寄存器的存取。

注4：有关DMA控制数据区和DMA向量表区的分配，请参照“18.4.1 DMA控制数据区和DMA向量表区的分配”的注意。

注5：设置该寄存器请保持256Byte对齐，也就是低8位设置为零。DMA硬件忽略低8位。

注6：该寄存器只能WORD访问，BYTE和HALFWORD访问忽略。

位	符号	描述	复位值
31:8	DMABARj	设置数据传送时的目标地址高24位	0x0
7:0	DMABARj	设置数据传送时的目标地址低8位，因256字节对齐，故设置为0	0x0

注：j=0

## 18.5 DMA的运行

DMA一旦启动，就从DMA控制数据区读控制数据，根据此控制数据进行数据传送，并且将数据传送后的控制数据回写到DMA控制数据区。能将8组控制数据保存到DMA控制数据区，并且进行8组数据的传送。传送模式有正常模式和重复模式，传送大小有8位传送，16位传送和32位传送。在DMACRj (j=0~7) 寄存器的CHNE位为“1”（允许链传送）时，通过1个启动源读多个控制数据进行连续的数据传送（链传送）。

通过32位DMSARj寄存器和32位DMDARj寄存器分别指定传送源地址和传送目标地址。在数据传送后，根据控制数据递增或者固定DMSARj寄存器和DMDARj寄存器的值。

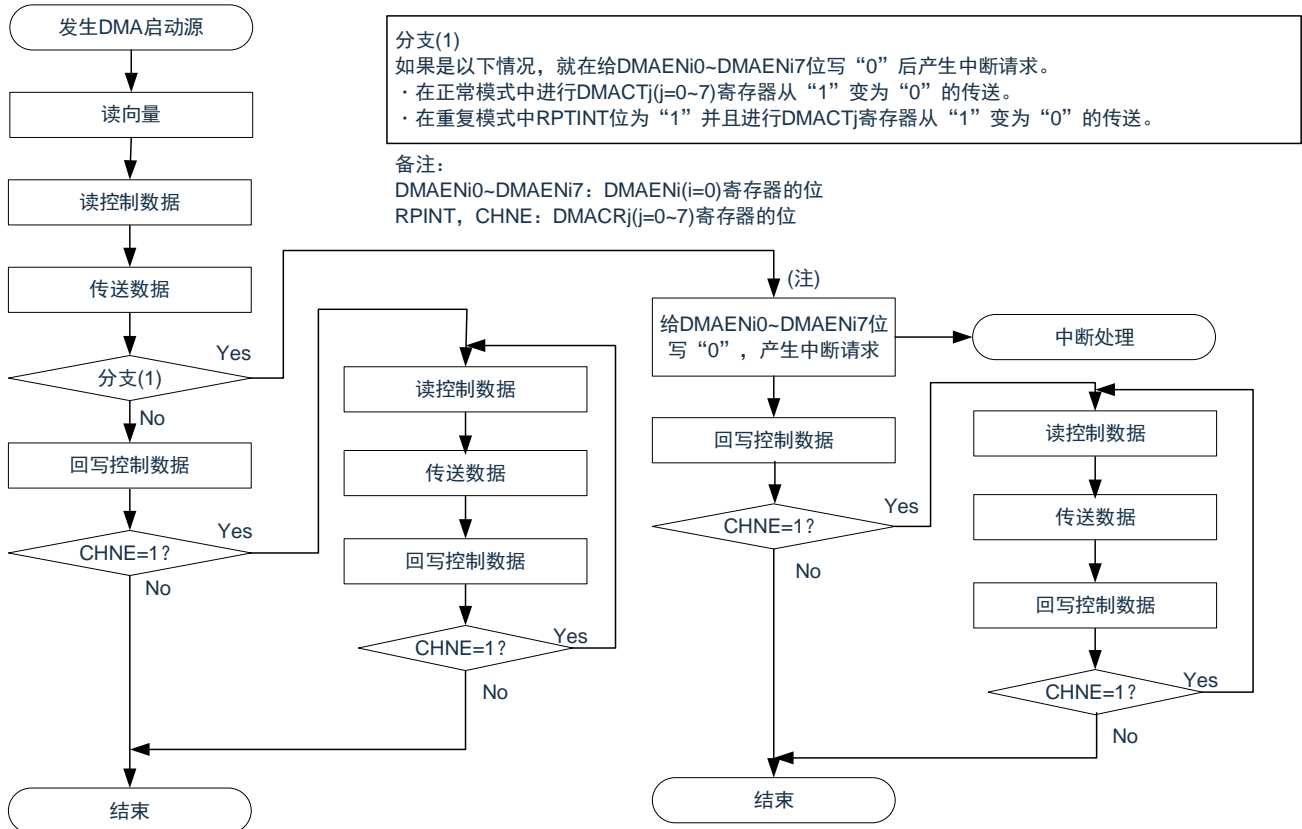
### 18.5.1 启动源

通过外围功能的中断信号启动DMA，并且通过DMAENi (i=0) 寄存器选择启动DMA的中断信号。当数据传送（在进行链传送时，连续进行最初的传送）的设置为下述两种情况时，就在DMA运行中将对应的DMAENi寄存器的DMAENi0~DMAENi7位置“0”（禁止启动）。

- 在正常模式中，进行DMACTj (j=0~7) 寄存器变为“0”的传送。
- 在重复模式中，DMACRj寄存器的RPTINT位为“1”（允许发生中断）并且进行DMACTj寄存器变为“0”的传送。

DMA的内部运行流程图如图18-5所示。

图18-5 DMA的内部运行流程图



注：在通过允许链传送(CHNE=1)的设置启动的数据传送中，不给DMAENi0~DMAENi7位写“0”且不产生中断请求。

### 18.5.2 正常模式

在8位传送时，1次启动的传送数据为1~65535字节；在16位传送时，1次启动的传送数据为2~131070字节；在32位传送时，1次启动的传送数据为4~262140字节。传送次数为1~65535次。如果进行DMACTj (j=0~7) 寄存器变为“0”的数据传送，就在DMA运行中向中断控制器产生对应启动源的中断请求，并且将对应的DMAENi (i=0) 寄存器的DMAENi0~DMAENi7位置“0”（禁止启动）。

正常模式的寄存器功能和数据传送分别如表18-6和图18-6所示

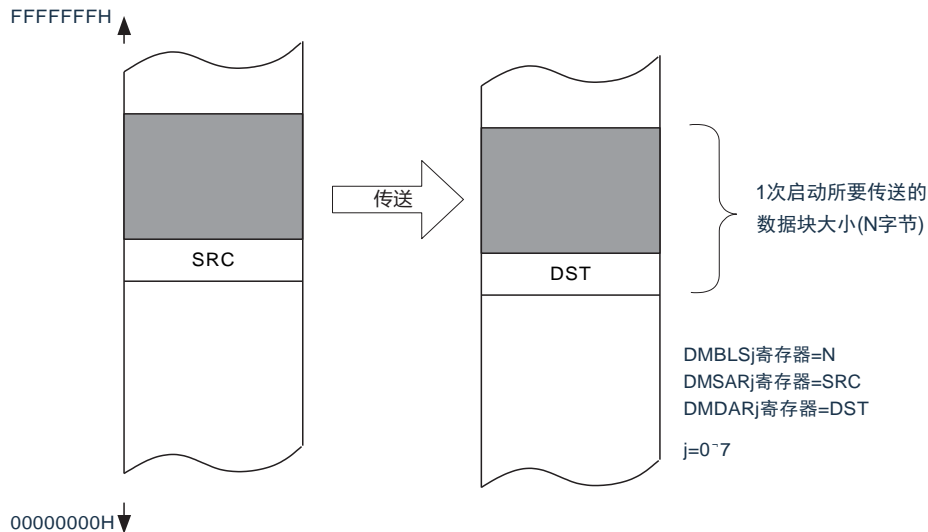
注：j=0

表18-6 正常模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重载寄存器j	DMRLDj	不使用注。
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

注：j=0

图18-6 正常模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	固定	固定	SRC	DST
0	1	X	0	递增	固定	SRC+N	DST
1	0	X	0	固定	递增	SRC	DST+N
1	1	X	0	递增	递增	SRC+N	DST+N

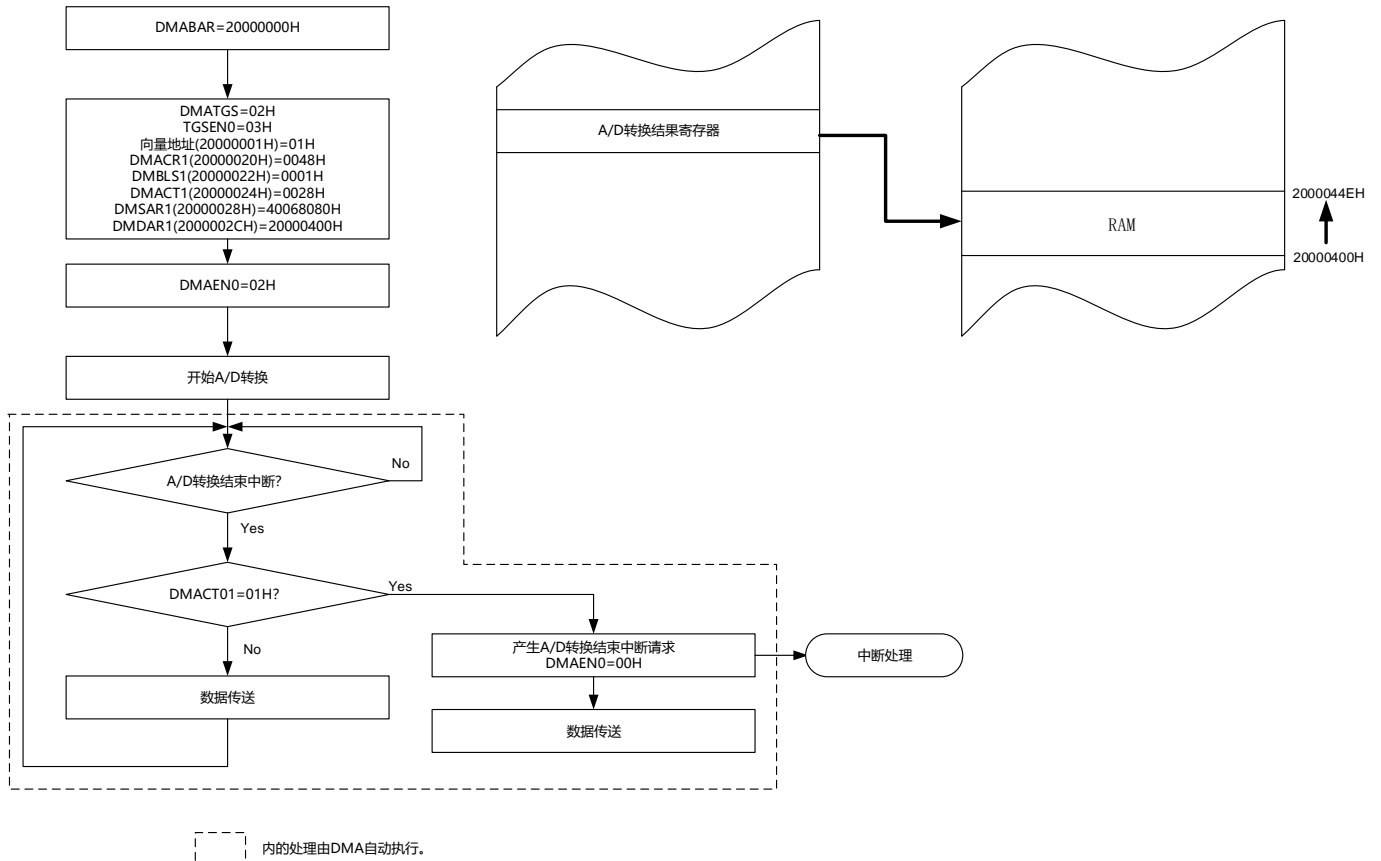
X: “0”或者“1”

(1) 正常模式的使用例子 1: 连续取 A/D 转换结果

通过A/D转换结束中断启动DMA，并且将A/D转换结果寄存器的值传送到RAM。

- 向量地址分配在20000001H，控制数据分配在20000020H~2000002FH。
- 将A/D转换结果寄存器（40068080H，40068081H）的2字节数据传送40次到RAM的20000400H~2000044FH的80字节。

图18-7 正常模式的使用例子 1: 连续取A/D转换结果



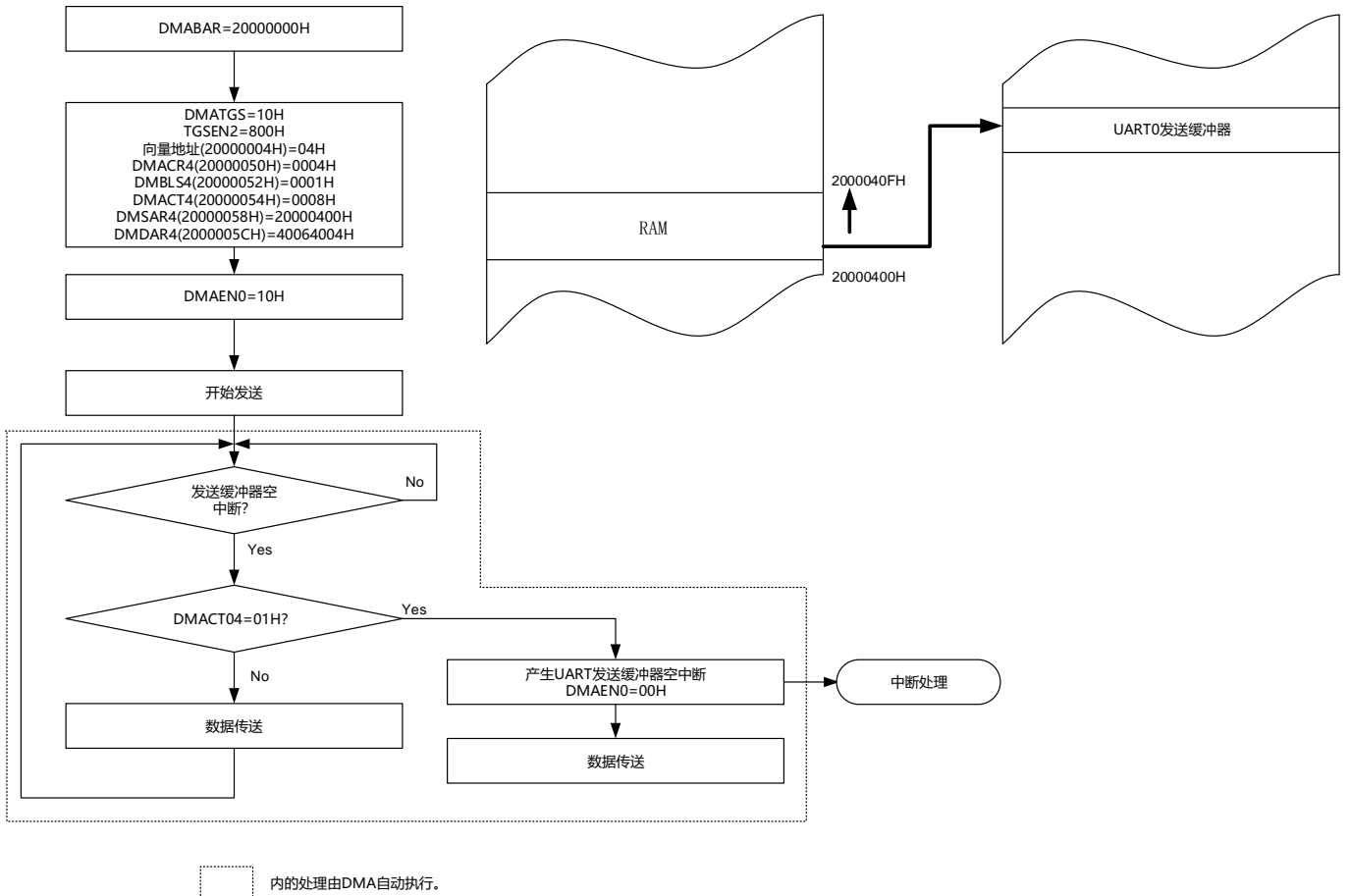
因为是正常模式，所以不使用DMRLD01寄存器的值。

(2) 正常模式的使用例子2: UART0连续发送

通过UART0的发送缓冲器空中断启动DMA, 并且将RAM的值传送到UART0的发送缓冲器。

- 向量地址分配在20000004H, 控制数据分配在20000050H~2000005FH。
- 将RAM的20000400H~20000407H的8字节传送到UART0的发送缓冲器(40064004H)。

图18-8 正常模式的使用例子2: UART0连续发送



因为是正常模式, 所以不使用DMRLD12寄存器的值。必须通过软件开始第1次的UART0发送。通过发送缓冲器空中断启动DMA, 然后自动进行第2次以后的发送。

### 18.5.3 重复模式

1次启动的传送数据为1~65535字节。将传送源或者传送目标指定为重复区，传送次数为1~65535次。一旦指定次数的传送结束，就对DMACTj (j=0~7) 寄存器以及指定为重复区的地址进行初始化，然后重复进行传送。当DMACRj寄存器的RPTINT位为“1”（允许发生中断）并且进行DMACTj寄存器变为“0”的数据传送时，就在DMA运行中向中断控制器产生对应启动源的中断请求，并且将对应的DMAENi (i=0) 寄存器的DMAENi0~DMAENi7位置“0”（禁止启动）。当DMACRj寄存器的RPTINT位为“0”（禁止发生中断）时，即使进行DMACTj寄存器变为“0”的数据传送，也不产生中断请求，而且DMAENi0~DMAENi7位不变为“0”。

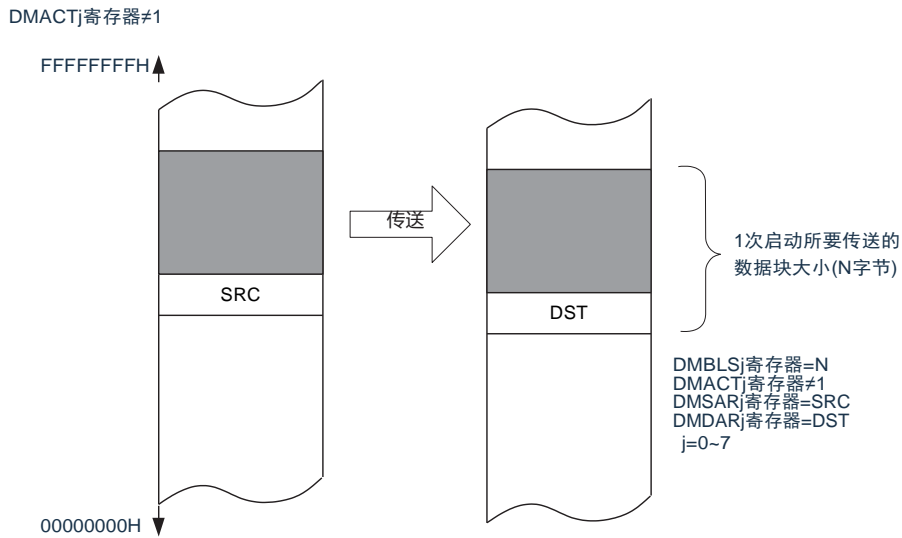
重复模式的寄存器功能和数据传送分别如表18-7和图18-9所示。

表18-7 重复模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	将此寄存器的值重新加载到DMACT寄存器。 (对数据的传送次数进行初始化)
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

注：j=0~7

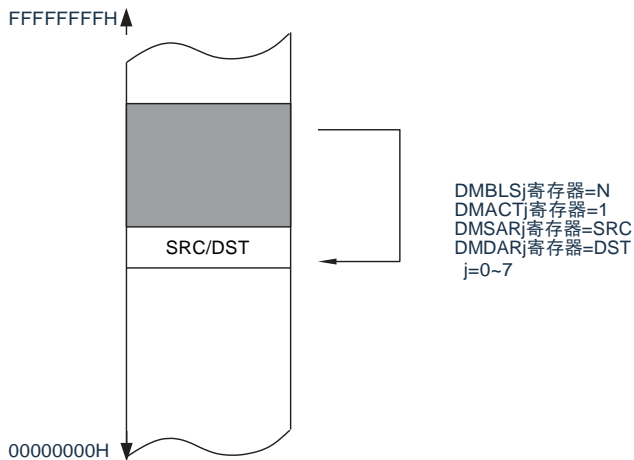
图18-9 重复模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC+N	DST
1	X	1	1	重复区	递增	SRC+N	DST+N
X	0	0	1	固定	重复区	SRC	DST+N
X	1	0	1	递增	重复区	SRC+N	DST+N

备注：X：“0”或者“1”

DMACTj寄存器=1



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC	DST
1	X	1	1	重复区	递增	SRC	DST+N
X	0	0	1	固定	重复区	SRC	DST
X	1	0	1	递增	重复区	SRC+N	DST

备注：X：“0”或者“1”

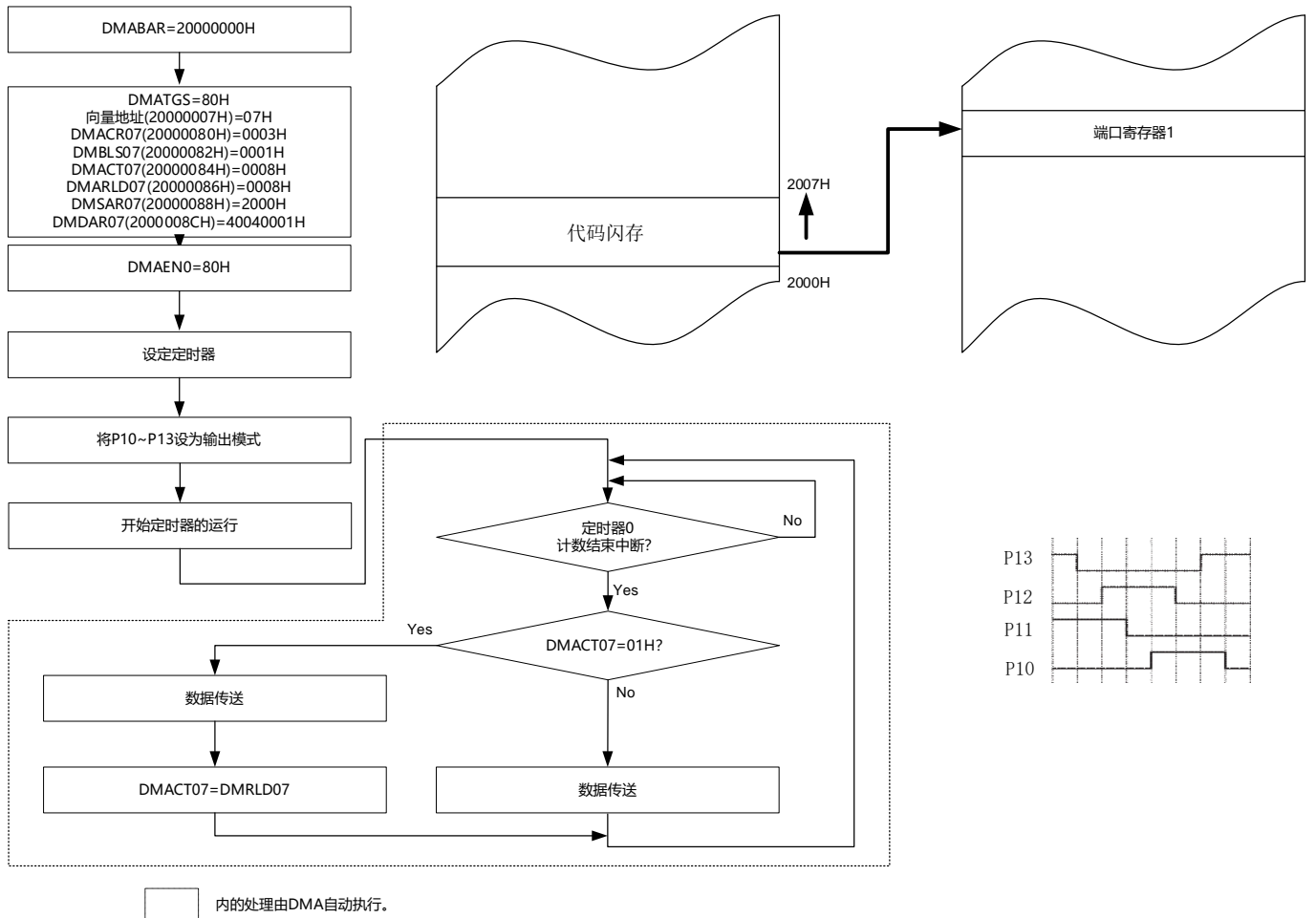
注：在使用重复模式时，必须将重复区的数据长度设置在65535字节以内。

(1) 重复模式的使用例子：使用端口的步进马达控制脉冲输出

使用TIMER0定时器功能启动DMA，并且将保存在代码闪存的马达控制脉冲的模式传送到通用端口。

- 向量地址分配在20000007H，控制数据分配在20000080H~2000008FH。
- 将代码闪存的02000H~02007H的8字节传送到端口寄存器1（40040001H）。
- 禁止重复模式中断。

图18-10重复模式的使用例子1：使用端口的步进马达控制脉冲输出



要停止输出时，必须在停止定时器的运行后清除DMAEN0的bit7。



### 18.5.4 链传送

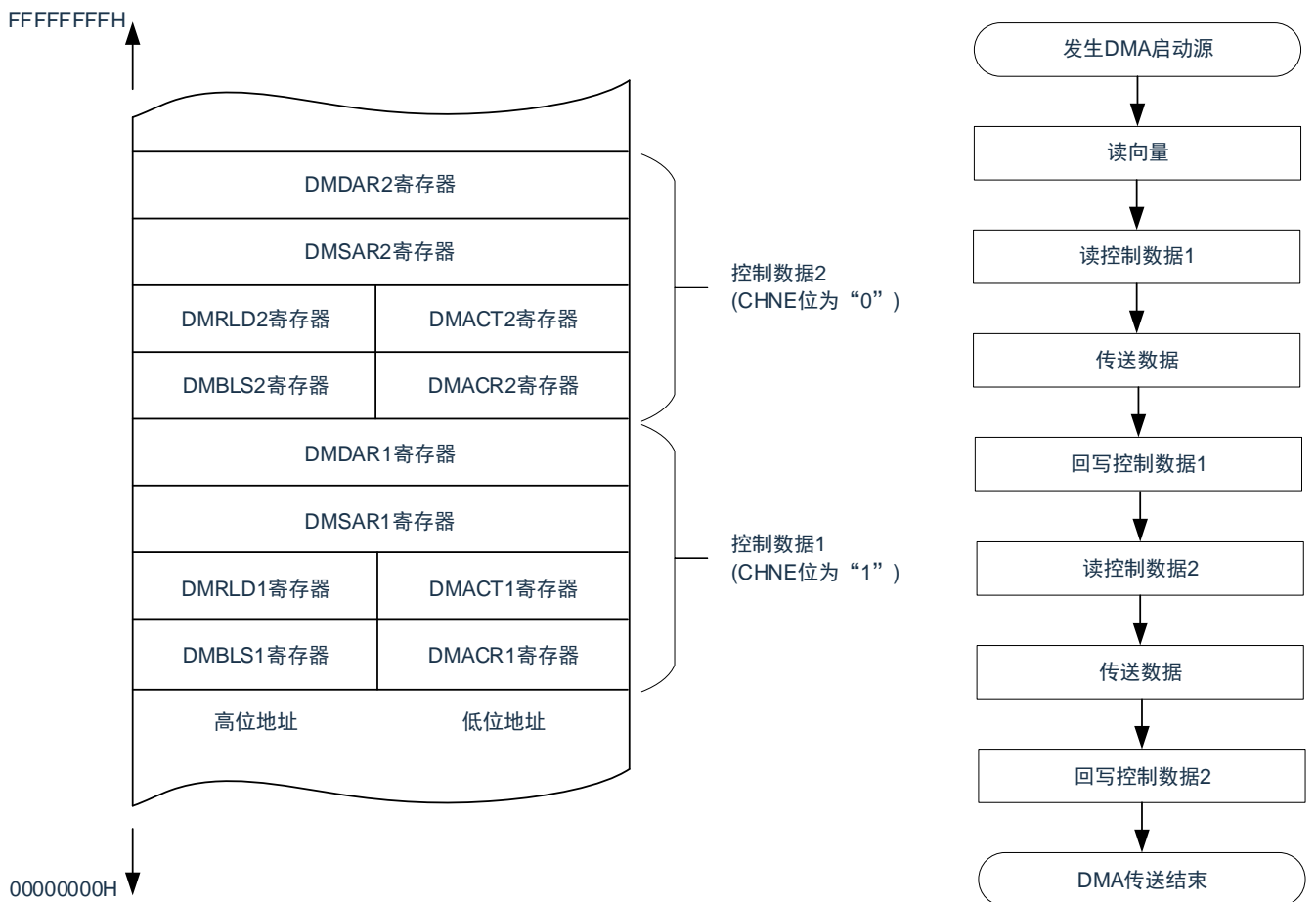
当DMACRj (j=0~7) 寄存器的CHNE位为“1 (允许链传送)”时，能通过1个启动源连续进行多个数据的传送。

DMA一旦启动，就通过从启动源对应的向量地址读取的数据来选择控制数据，读被分配在DMA控制数据区的控制数据。如果读到的控制数据的CHNE位为“1” (允许链传送)，就在传送结束后读下一个被分配的控制数据，继续进行传送。重复此操作，直到CHNE位为“0” (禁止链传送)的控制数据传送结束为止。

在使用多个控制数据进行链传送时，第一个控制数据设置的传送次数有效，而第2个以后处理的控制数据的传送次数无效。

链传送的流程图如图18-11所示。

图18-11 链传送的流程图



注1：必须将DMACR07寄存器的CHNE位置“0” (禁止链传送)。

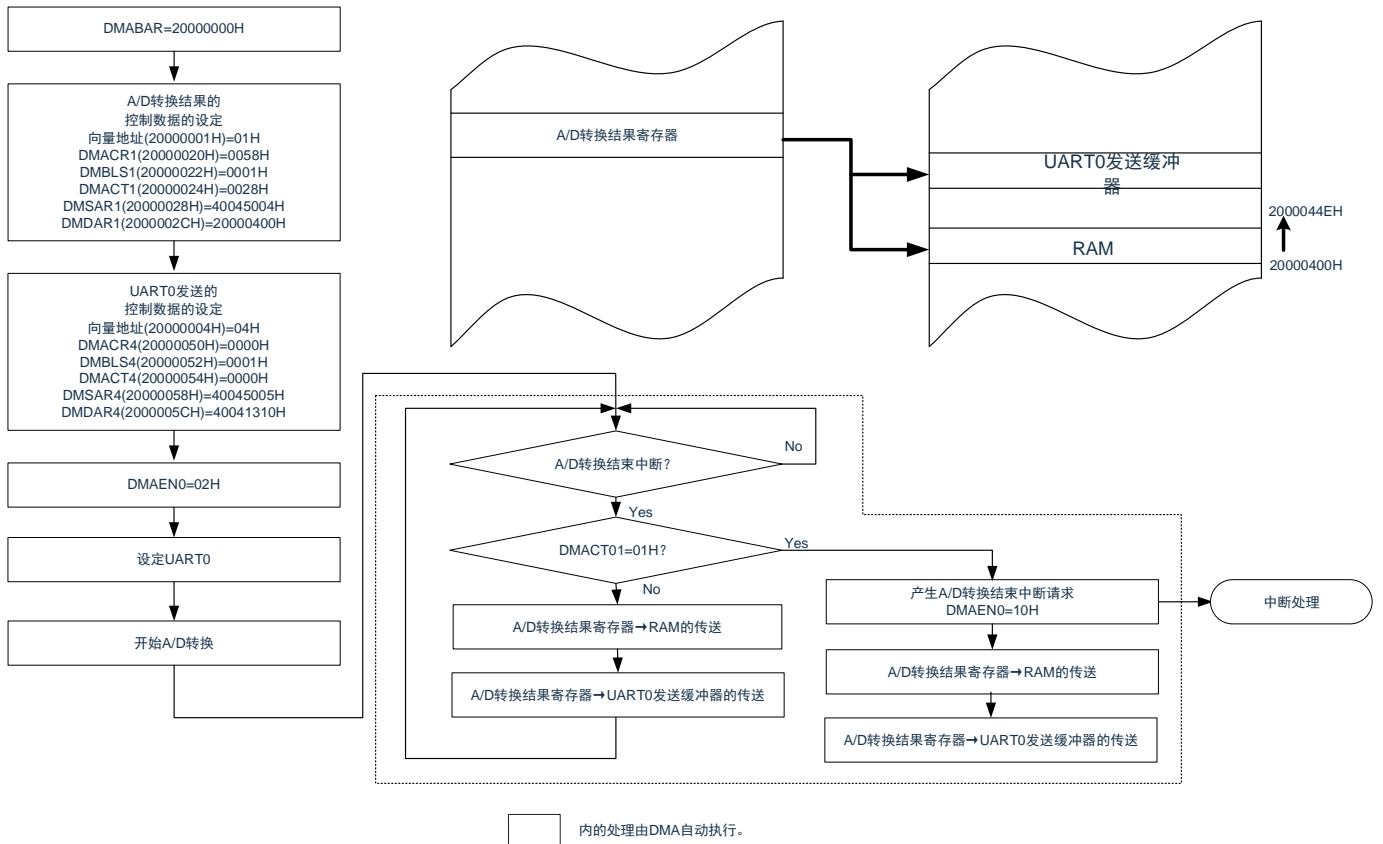
注2：在链传送的第2次以后的数据传送时，DMAENi (i=0) 寄存器的DMAENi0~DMAENi7位不变为“0” (禁止DMA启动)，并且不产生中断请求。

(1) 链传送的使用例子：连续取A/D转换结果进行UART0发送

通过A/D转换结束中断启动DMA，并且将A/D转换结果传送到RAM进行UART0发送。

- 向量地址分别为20000001H和20000004H。
- A/D转换结果的控制数据分配在20000020H~2000002FH。
- UART0发送的控制数据分配在20000050H~2000005FH。
- 将A/D转换结果寄存器（40068080H，40068081H）的2字节数据传送到RAM的20000400H~2000044FH，并且将A/D转换结果寄存器的高位1字节（40068081H）传送到UART0的发送缓冲器（40064004H）。

图18-12 链传送的使用例子：连续取A/D转换结果进行UART0发送



## 18.6 使用DMA时的注意事项

### 18.6.1 DMA控制数据和向量表的设置

- 必须在将全部的DMA启动源设置为禁止启动的状态下更改DMA基址寄存器（DMABAR）。
- 只能改写1次DMA基址寄存器（DMABAR）。
- 必须在对应的DMAENi（i=0）寄存器的DMAENi0~DMAENi7位为“0”（禁止DMA启动）时更改DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj寄存器的数据。
- 必须在对应的DMAENi（i=0）寄存器的DMAENi0~DMAENi7位为“0”（禁止DMA启动）时更改设置在向量表中的DMA控制数据区的起始地址。

### 18.6.2 DMA控制数据区和DMA向量表区的分配

能分配DMA控制数据和向量表的区域因产品和使用条件而不同。

- 堆栈区、DMA控制数据区和DMA向量表区不能重叠。

### 18.6.3 DMA的执行时钟数

DMA启动时的执行情况和所需的时钟数如表18-8所示。

表18-8 DMA启动时的执行情况和所需的时钟数

读向量	控制数据		读数据	写数据
	读	回写		
1	4	注1	注2	注2

注1：有关回写控制数据所需的时钟数，请参照“表18-9回写控制数据所需的时钟数”

注2：有关读写数据所需的时钟数，请参照“表18-10读写数据所需的时钟数”。

表18-9 回写控制数据所需的时钟数

DMACR寄存器的设置				地址设置		控制寄存器的回写				时钟数
DAMOD	SAMOD	RPTSEL	MODE	源	目标	DMACTj 寄存器	DMRLDj 寄存器	DMSARj 寄存器	DMDARj 寄存器	
0	0	X	0	固定	固定	回写	回写	不回写	不回写	1
0	1	X	0	递增	固定	回写	回写	回写	不回写	2
1	0	X	0	固定	递增	回写	回写	不回写	回写	2
1	1	X	0	递增	递增	回写	回写	回写	回写	3
0	X	1	1	重复区	固定	回写	回写	回写	不回写	2
1	X	1	1		递增	回写	回写	回写	回写	3
X	0	0	1	重复区	固定	回写	回写	不回写	回写	2
X	1	0	1		递增	回写	回写	回写	回写	3

注：j=0~7，X：“0”或者“1”

表18-10 读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	特殊功能寄存器 (SFR)	扩展特殊功能寄存器 (2ndSFR)	
					无等待	等待
读数据	1	2	4	1	1	1+等待数
写数据	1	—	—	1	1	1+等待数

## 18.6.4 DMA的响应时间

DMA响应时间如表18-11所示。DMA响应时间是指从检测到DMA启动源到开始DMA传送的时间，不包括DMA的执行时钟数。

表18-11 DMA的响应时间

	最短时间	最长时间
响应时间	3个时钟	23个时钟

但是，在以下情况下DMA的响应可能还会延迟。延迟的时钟数因条件而不同。

- 从内部RAM执行指令的情况最长响应时间：20个时钟

注：1个时钟：1/f<sub>CLK</sub>（f<sub>CLK</sub>：CPU/外围硬件时钟）

## 18.6.5 DMA的启动源

- 不能在从输入DMA启动源到结束DMA传送的期间输入相同的启动源。
- 在产生DMA启动源的位置，不能操作该启动源对应的DMA启动允许位。
- 如果DMA启动源发送竞争，就在CPU接受DMA传送时判断优先级，决定启动启动源。有关启动源的优先级，请参照“18.4.3 向量表”。

## 18.6.6 待机模式中的运行

状态	DMA运行
睡眠模式	能运行（禁止在低功耗RTC模式中运行）。
深度睡眠模式	能接受DMA启动源，并进行DMA传送 <sup>注1</sup>

注1：在深度睡眠模式中，能在检测到DMA启动源后进行DMA传送，并且在传送结束后返回到深度睡眠模式。但是，因为在深度睡眠模式中代码闪存和数据闪存停止运行，所以不能将闪存设置为传送源。

# 第19章 快速模数转换 (ADC)

## 19.1 概述

芯片包含一个12位27通道快速逐次逼近型模数转换器(ADC)。

## 19.2 特性

- ◆ 模拟输入电压范围：VSS ~ AVDD。
- ◆ 最大采样速率：1.2Msps。
- ◆ 多达27路单端模拟输入通道。
- ◆ 支持两种功耗模式：高速模式与低电流模式。
- ◆ 高速模式下完成一次转换时间为： $52 \cdot T_{ADCK}$ (采样时间设置为 $13.5 \cdot T_{ADCK}$ )。
- ◆ 单次模式：对指定通道执行一次A/D转换。
- ◆ 连续模式：对所有选定的通道都执行A/D转换。
- ◆ 支持外部输入信号触发ADC转换。
- ◆ 支持转换完毕产生中断。
- ◆ 内置AD转换结果比较器。
- ◆ 每个通道的转换结果都存储在对应的数据寄存器中。

## 19.3 功能描述

### 19.3.1 ADC的通道

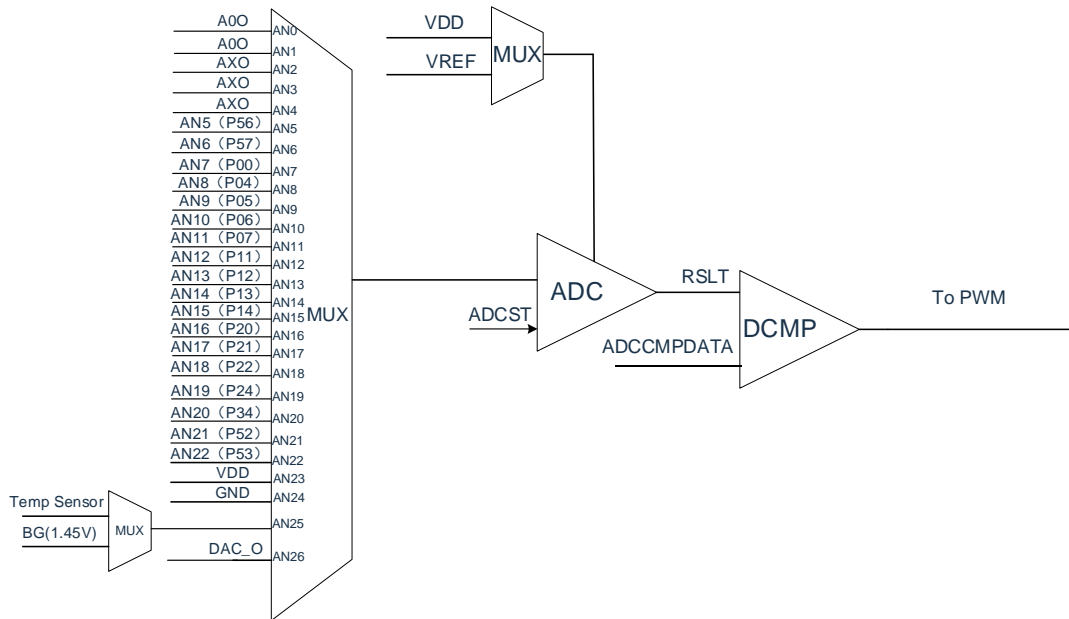
ADC通道号 (支持硬件触发)	ADC通道	ADC通道优先级	说明
0	AN0 (A0O)	最高	PGA0通道(详见第20章)
1	AN1 (A0O)		PGA0通道(详见第20章)
2	AN2 (AXO)		PGA1通道(详见第20章)
3	AN3 (AXO)		PGA2通道(详见第20章)
4	AN4 (AXO)		PGA3通道(详见第20章)
5	AN5 (P56)		外部通道5
6	AN6 (P57)		外部通道6
7	AN7 (P00)		外部通道7
8	AN8 (P04)		外部通道8
9	AN9 (P05)		外部通道9
10	AN10 (P06)		外部通道10
11	AN11 (P07)		外部通道11
12	AN12 (P11)		外部通道12
13	AN13 (P12)		外部通道13
14	AN14 (P13)		外部通道14
15	AN15 (P14)		外部通道15
16	AN16 (P20)		外部通道16
17	AN17 (P21)		外部通道17
18	AN18 (P22)		外部通道18
19	AN19 (P24)		外部通道19
20	AN20 (P34)		外部通道20
21	AN21 (P52)		外部通道21
22	AN22 (P53)		外部通道22
23	AN23 (VDD)		
24	AN24 (GND)		
25	AN25 (BG2AD)	专用通道	BG1.45V/温度传感器通道
26	AN26	专用通道	

注：AN0-AN26 任意组合通道支持连续模式转换。

#### ADC 的内部通道

ADC内部通道号	ADC内部通道	说明
0~3	-	禁止选择
4	IAN_4 (DAC_O)	DAC输出通道(详见第22章)

### 19.3.2 ADC结构框图



### 19.3.3 ADC的功耗模式

ADC工作模式分为两种：高速模式与低电流模式。

高速模式：该模式下转换速度较快。

低电流模式：该模式转换速度稍慢，ADC的工作电流显著下降。对转换速率要求不高的应用来说，可采用此模式来降低ADC的功耗。此模式的逐次比较时间比高速模式多10个 $T_{ADCK}$ 的时间。

### 19.3.4 ADC的转换模式

ADC 转换模式分为两种：单次转换模式与连续转换模式。

单次转换模式：

对已使能优先级最高的通道进行一次转换之后结束操作并产生中断标志位。

连续转换模式：

对所有已使能的通道进行转换之后结束操作并产生中断标志位，未使能的通道忽略跳过。

ADCSWCHE=0 时，软件通道开启禁止，ADC 通道的选择与开启由硬件自动控制。

ADCSWCHE=1 时，软件通道开启使能，ADC 通道的选择与开启由 ADCSWCHS 来控制，ADCSWCHS 选择某一通道后，该通道自动开启（ADCEN 必须为 1）。单次模式与连续模式在此条件下实际是对 ADCSWCHS 选择的通道进行转换。



### 19.3.5 ADC的时钟

ADC 的时钟来源于 APB 时钟，可选择 8 种分频 1/2/4/8/16/32/64/128，通过 ADCCON.ADCDIV 来配置。

高速模式下单次转换模式 AD 转换的时间 ( $T_{ADCK}$  为 ADC 不分频情况下的时钟)：

$$1 * T_{ADCK} + 2 * T_{ADCK} \text{ (默认开关稳定时间)} + 13.5 * T_{ADCK} \text{ (默认采样时间)} + 31.5 * T_{ADCK} \text{ (逐次比较时间)} + 4 * T_{ADCK}$$

高速模式模式下连续转换模式完成一次 AD 转换的时间 ( $T_{ADC}$  为 ADC 不分频情况下的时钟)：

$$2 * T_{ADCK} \text{ (默认开关稳定时间)} + 13.5 * T_{ADCK} \text{ (默认采样时间)} + 31.5 * T_{ADCK} \text{ (逐次比较时间)} + 3 * T_{ADCK}$$

ADCSWCHE=1 时，实际的开关稳定时间为选择到某一通道后到开始转换的时间。

### 19.3.6 ADC的通道选择及中断产生

ADCSWCHE	ADCMS	通道说明	结果存放	中断产生
0	0	开启硬件已使能的优先级最高的通道	将结果存放在硬件已使能优先级最高的结果寄存器中	中断产生在硬件已使能优先级最高的通道中
0	1	依次开启硬件已使能的通道，直至所有已使能通道都完成转换	转换完成一个通道，将结果存放在转换通道对应的结果寄存器中	转换完成一个通道将产生对应通道的中断标志位
1	0	开启 ADCSWCHS 中设置的通道	将结果存放在硬件已使能的优先级最高的通道对应的结果寄存器中	产生硬件已使能的优先级最高的通道的中断标志位
1	1	开启 ADCSWCHS 中设置的通道。硬件已使能通道有多少个，就连续转换多少次（开启的通道始终为 ADCSWCHE 所设置的）	结果依次存放在硬件已使能的通道对应的结果寄存器中	中断依次产生在硬件已使能的通道中

注1：硬件已使能的通道是指：当由EPWM零点触发，对应CHZIFTG 寄存器已使能的通道；当由EPWM周期点触发，则对应CHPIFTG 寄存器已使能的通道；当由EPWM比较点0/1触发，则对应CHPTG0/1 寄存器已使能的通道；当EPWM输出触发时，则对应CHEPWM寄存器已使能的通道；其余软件触发，内部触发和外部触发均为SCAN寄存器已使能的通道。

注2：硬件自动选择通道模式下，当两个或多个触发信号同时到来时，硬件开启通道选择的寄存器优先级如下：

$$CHZIFTG > CHPIFTG > CHPTG1 > CHPTG0 > CHEPWM > SCAN。$$

注3：当使用软件选择ADC转换通道时，不要在ADCCON.ADCST为1的时候通过软件切换通道。

### 19.3.7 ADC软件启动

在寄存器ADCCON2.ADCST位中写入1，将启动ADC转换。转换完毕后，该位硬件自动清零。在ADC转换期间，任何软件和硬件触发启动信号将被忽略。

### 19.3.8 ADC硬件触发启动

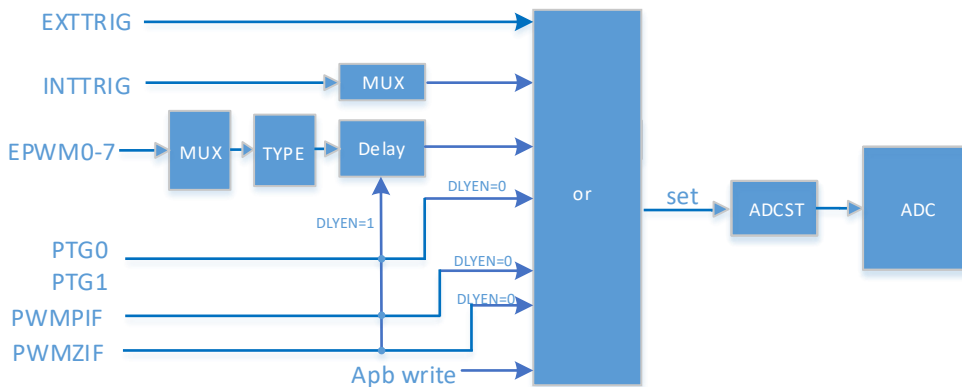
**触发源:**

ADC除了软件启动转换之外，还可通过硬件触发ADC转换。硬件触发源的种类有：

- 1) 外部触发
- 2) 内部触发
- 3) EPWM 输出通道触发
- 4) EPWM 周期点触发
- 5) EPWM 零点触发
- 6) EPWM 计数比较器 0 触发
- 7) EPWM 计数比较器 1 触发

不同种类的触发源可同时有效，同一种类的触发源可能含有不同触发信号，如EPWM输出通道触发，可选择EPWM0-EPWM7其中之一的触发信号。

图19-1：ADC硬件触发启动



**内部触发:**

内部触发包括：ADC，ACMP0，ACMP1，TIMER0/1触发。

ADC：ADC的转换结束

ACMP0：ACMP0的事件输出

ACMP1：ACMP1的事件输出

Timer0：Timer0的已使能中断(TMR0MIS)

Timer1：Timer1的已使能中断(TMR1MIS)

**EPWM输出通道触发:**

EPWM 输出通道触发可选择上升沿，下降沿，零点，周期点启动 ADC，如检测到 EPWM 触发信号，可选择经过一定延时之后启动 ADC 转换（无单独延时触发使能位，若延时数据不为 0，则自动经过延时）。若 EPWM 的输出通道经过重映射，则 EPWM 触发信号为重映射前的信号，IPGn 信号。

EPWM 输出通道触发仅支持硬件选择通道，可单独设置 ADC 的转换通道。即 EPWM 输出通道触发信号产生后，将按照独自的设置通道转换。EPWM 输出通道触发 ADC 的转换通道在 ADCCHEPWM 寄存器中设置。转换完毕后将恢复为 ADCSCAN 寄存器中的通道设置。

### EPWM零点和周期点触发:

EPWM 零点和周期点触发源可选择 EPWM 任意通道的零点和周期点，方式同 EPWM 通道触发相同，也可选择经过一定延时之后启动 ADC 转换，并且可通过 ADCHWTG. ADCEPWMZDLYEN 和 ADCHWTG. ADCEPWMPDLYEN 单独设置零点、周期点触发 ADC 是否经过延时。

EPWM 零点、周期点触发仅支持硬件选择通道，可单独设置 ADC 的转换通道。即零点或周期点触发信号产生后，将按照独自的设置通道转换。EPWM 零点触发 ADC 的转换通道在 ADCCHZIFTG 寄存器中设置。EPWM 周期点触发 ADC 的转换通道在 ADCCHPIFTG 寄存器中设置。转换完毕后将恢复为 ADCSCAN 寄存器中的通道设置。

### EPWM计数器比较器触发:

EPWM 计数比较器 0/1 触发可设置在 EPWMn 周期内任意时刻触发启动 ADC，方式同 EPWM 通道触发相同，也可选择经过一定延时之后启动 ADC 转换，并且可通过 ADCHWTG. ADCPTG0DLYEN 和 ADCHWTG. ADCPTG1DLYEN 单独设置比较器 0/1 触发 ADC 是否经过延时。

EPWM 计数比较器 0/1 触发仅支持硬件选择通道，可单独设置 ADC 的转换通道。即触发信号产生后，将按照独自的设置通道转换。EPWM 计数比较器 0 触发 ADC 的转换通道在 ADCCHPTG0 寄存器中设置。EPWM 计数比较器 1 触发 ADC 的转换通道在 ADCCHPTG1 寄存器中设置。转换完毕后将恢复为 ADCSCAN 寄存器中的通道设置。

**注：**在延时触发模式下，当一个触发信号到来后，在延时的过程中，有已使能优先级更高的触发信号到来时，延时结束后，将按照高优先级触发信号设置的通道进行转换（优先级见 19.3.6 章节内容）。例如，当 EPWM 周期点和 EPWM 零点延时触发皆使能时，EPWM 周期点触发信号到来后，内部延时计数器开始计数，期间若 EPWM 零点触发信号到来，延时计数器值等于设置的延时数据后，系统将自动开启 ADCCHZIFTG 寄存器使能的通道。

### EPWM触发延时:

ADCEPWMTGDLY 寄存器决定 EPWM 触发启动 ADC 延时的时间（该延时时间为有效触发信号的延变化到 ADCST 信号上升沿的时间）：

$$(ADCEPWMTGDLY[9:0]+3)*T_{PCLK}$$

EPWM触发延时的范围如下:

	pclk 64MHz (延时范围)	pclk 72MHz (延时范围)
延时使能且延时数据不为 0	0.062us~16.03us	0.056us~14.25us
延时禁止或延时数据为 0	0.047us	0.042us

若 ADCEPWMTGDLY=0 或对应的延时触发禁止时,EPWM 比较器 0/EPWM 比较器 1/EPWM 输出/周期点/零点延时 3 个  $T_{PCLK}$  时钟启动 ADC 转换。

### EPWM 触发启动 ADC 设置

EPWM触发ADC转换在一些应用中有特殊的时间要求。针对该需求，ADC内部支持不同的EPWM触发条件可设置独立的转换通道。例如：

EPWM输出通道触发可选择AN0、AN1、AN2通道转换。

EPWM比较器0触发可选择AN18通道转换。

EPWM比较器1触发可选择AN19通道转换。

EPWM周期点触发可选择AN20通道转换

EPWM零点触发可选择AN21通道转换

软件启动或其他触发启动选择的通道为AN5、AN6、AN7、AN8。

没有EPWM触发条件时，默认的转换通道为AN5-AN8。

若EPWM的输出通道触发后将仅选择AN0-AN2 3个通道进行AD转换，转换完毕后自动切换为AN5-AN8通道使能。

若EPWM的比较0触发后，仅选择AN18通道进行AD转换，转换完毕后自动切换为AN5-AN8通道使能。

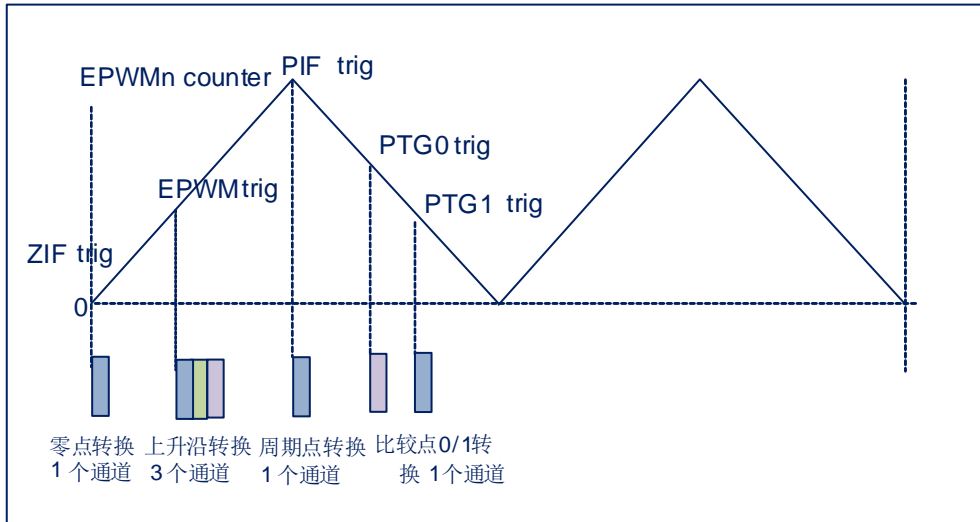
若EPWM的比较1触发后，仅选择AN19通道进行AD转换，转换完毕后自动切换为AN5-AN8通道使能。

若EPWM的周期点触发后，仅选择AN20通道进行AD转换，转换完毕后自动切换为AN5-AN8通道使能。

若EPWM的零点触发后，仅选择AN21通道进行AD转换，转换完毕后自动切换为AN5-AN8通道使能。

需要注意的是，在AD转换未结束期间，其他的任何触发信号将被忽略。

图19-2: EPWM触发启动ADC设置



注 1：零点，周期，EPWM 输出上升沿，EPWM 输出下降沿触发 ADC 使能转换的通道由 ADCCHPEM 决定。

注 2：比较点 0 触发 ADC 使能的转换通道由 ADCCHPTG0 决定。

注 3：比较点 1 触发 ADC 使能的转换通道由 ADCCHPTG1 决定。

注 4：周期点触发 ADC 使能的转换通道由 ADCCHPIFTG 决定。

注 5：零点触发 ADC 使能的转换通道由 ADCCHZIFTG 决定。

注 6：其他方式启动 ADC 使能的转换通道由 ADCSCAN 或 ADCSWCHS 决定。

## 19.4 寄存器映射

(ADC基地址 = 0x4006\_8000)      RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CON <sub>(P1B)</sub>	0x000	R/W	ADC控制寄存器	0x000D0000
CON2 <sub>(P1B)</sub>	0x004	R/W	ADC控制寄存器2	0x003F0000
HWTG <sub>(P1B)</sub>	0x008	R/W	ADC 硬件触发控制寄存器	0x00000000
PWMTGDLY <sub>(P1B)</sub>	0x00C	R/W	ADC EPWM触发延时数据寄存器	0x00000000
SCAN <sub>(P1B)</sub>	0x010	R/W	ADC 扫描寄存器	0x00000000
CMP0 <sub>(P1B)</sub>	0x014	R/W	ADC 比较器0控制寄存器	0x00000000
CMP1 <sub>(P1B)</sub>	0x018	R/W	ADC 比较器1控制寄存器	0x00000000
IMSC <sub>(P1B)</sub>	0x01C	R/W	ADC 中断使能寄存器	0x00000000
RIS	0x020	RO	ADC 中断源状态寄存器	0x00000000
MIS	0x024	RO	ADC 已使能中断状态寄存器	0x00000000
ICLR	0x028	WO	ADC 中断清零寄存器	0x00000000
LOCK	0x02C	R/W	ADC 写使能控制寄存器	0x00000000
CHEPWM <sub>(P1B)</sub>	0x030	R/W	ADC EPWM输出触发转换通道寄存器	0x00000000
CHPTG0 <sub>(P1B)</sub>	0x034	R/W	ADC EPWM 比较器0触发转换通道寄存器	0x00000000
CHPTG1 <sub>(P1B)</sub>	0x038	R/W	ADC EPWM比较器1触发转换通道寄存器	0x00000000
CHPFTG <sub>(P1B)</sub>	0x03C	R/W	ADC EPWM 周期点触发转换通道寄存器	0x00000000
CHZIFTG <sub>(P1B)</sub>	0x040	R/W	ADC EPWM 零点触发转换通道寄存器	0x00000000
TEST <sub>(P1B)</sub>	0x048	R/W	ADC测试寄存器	0x00000000
RESULT	0x050	RO	ADC转换结果寄存器, ADC转换完成刷新	0x00000000
DATA0	0x080	RO	ADC 通道0转换结果寄存器	0x00000000
DATA1	0x084	RO	ADC 通道1转换结果寄存器	0x00000000
DATA2	0x088	RO	ADC 通道2转换结果寄存器	0x00000000
DATA3	0x08C	RO	ADC 通道3转换结果寄存器	0x00000000
DATA4	0x090	RO	ADC 通道4转换结果寄存器	0x00000000
DATA5	0x094	RO	ADC 通道5转换结果寄存器	0x00000000
DATA6	0x098	RO	ADC 通道6转换结果寄存器	0x00000000
DATA7	0x09C	RO	ADC 通道7转换结果寄存器	0x00000000
DATA8	0x0A0	RO	ADC 通道8转换结果寄存器	0x00000000
DATA9	0x0A4	RO	ADC 通道9转换结果寄存器	0x00000000
DATA10	0x0A8	RO	ADC 通道10转换结果寄存器	0x00000000
DATA11	0x0AC	RO	ADC 通道11转换结果寄存器	0x00000000
DATA12	0x0B0	RO	ADC 通道12转换结果寄存器	0x00000000
DATA13	0x0B4	RO	ADC 通道13转换结果寄存器	0x00000000
DATA14	0x0B8	RO	ADC 通道14转换结果寄存器	0x00000000
DATA15	0x0BC	RO	ADC 通道15转换结果寄存器	0x00000000
DATA16	0x0C0	RO	ADC 通道16转换结果寄存器	0x00000000
DATA17	0x0C4	RO	ADC 通道17转换结果寄存器	0x00000000
DATA18	0x0C8	RO	ADC 通道18转换结果寄存器	0x00000000
DATA19	0x0CC	RO	ADC 通道19转换结果寄存器	0x00000000
DATA20	0x0D0	RO	ADC 外部通道20转换结果寄存器	0x00000000
DATA21	0x0D4	RO	ADC 外部通道21转换结果寄存器	0x00000000
DATA22	0x0D8	RO	ADC 外部通道22转换结果寄存器	0x00000000
DATA23	0x0DC	RO	ADC 外部通道23转换结果寄存器	0x00000000
DATA24	0x0E0	RO	ADC 外部通道24转换结果寄存器	0x00000000

DATA25	0x0E4	RO	ADC 外部通道25转换结果寄存器	0x00000000
DATA26	0x0E8	RO	ADC 外部通道26转换结果寄存器	0x00000000

注1: (P1B)标注的寄存器为被保护的寄存器。

注2: (P1B): LOCK==55H时, 标注的寄存器允许写入; LOCK==其他值, 禁止写入。

## 19.5 寄存器说明

### 19.5.1 ADC控制寄存器(CON)

位	符号	描述	复位值
31	ADCRST	ADC模块复位控制位 0: --- 1: ADC模块复位	0
30:26	-	保留, 必须为0	0x0
25:24	ADMODE10	ADC功耗模式选择位 00: 高速模式 01: 保留, 禁止选择 10: 保留, 禁止选择 11: 低电流模式	0x0
23:16	ADCNSMP	ADC内部采样时间选择位 0000000: 至 禁止选择 00000100: 5.5 ADC clockcycles 00000110: 6.5 ADC clockcycles 00000111: 7.5 ADC clockcycles 00001000: 8.5 ADC clockcycles 00001001: 9.5 ADC clockcycles 00001010: 10.5 ADC clockcycles 00001011: 11.5 ADC clockcycles 00001100: 12.5 ADC clockcycles 00001101: 13.5 ADC clockcycles 00001000: ... 11111110: 254.5 ADC clockcycles 11111111: 255.5 ADC clockcycles	0x0D
15:14	-	保留, 必须为0	0x0
13	ADCSWCHE	ADC通道软件开启使能位 0: 由硬件自动开启 1: 通道开启由ADCSWCHS决定	0
12	ADCNDISEN	ADC充放电功能选择位 0: 放电 1: 充电	0
11:8	ADCNDISTS	ADC充放电时间选择位 0000: 不进行充电或放电 0001: 禁止选择 0010: 2 ADC clockcycles 0011: 3 ADC clockcycles ... 1111: 15 ADC clockcycles	0x0
7:6	ADCVS	ADC正端参考选择位 00: 选择VDD 01: 选择VREF 10: 保留 11: 禁止选择	0x0



5	-	保留, 必须为0	0
4	ADCEN	ADC使能控制位 0: 禁止 1: 使能	0
3	ADCMS	ADC转换模式选择位 0: 单次转换 1: 连续转换 (一次转换完所有使能的ADC通道, 顺序为通道0至通道26, 没有使能的通道硬件自动忽略, 不会产生转换操作)	0
2:0	ADCDIV	ADC时钟预分频选择位 $F_{ADC} = PCLK/2^{ADCDIV}$	0x0



## 19.5.2 ADC控制寄存器2(CON2)

位	符号	描述	复位值
31:16	-	保留	-
15:13	ADCICHES	ADC内部通道(AN26)选择位 100: 选择内部通道4 其他: 禁止选择	0x0
12	ADCSF4	ADC转换状态标志位4 (仅读) 0: - 1: 单次转换完成	0
11	ADCSF3	ADC转换状态标志位3 (仅读) 0: - 1: 转换完成前一个ADC clock cycles	0
10	ADCSF2	ADC转换状态标志位2 (仅读) 0: - 1: 转换完成前两个ADC clock cycles	0
9	ADCSF1	ADC转换状态标志位1 (仅读) 0: - 1: 转换过程中	0
8	ADCSF0	ADC转换状态标志位0 (仅读) 0: - 1: 采样过程中	0
7	ADCST	ADC转换开始(转换结束后硬件自动清零) 0: 转换结束或ADC处于空闲模式 (写0无效) 1: 开始转换(ADCEN必须为1)	0
6	ADCSMPWAIT	ADC采样时间延长控制位 0: - 1: 采样过程中强制保持采样状态	0
5	BG2ADSEL	TS通道(AN25)选择位 0: TS温度传感器 1: BG基准电压1.45V	0
4:0	ADCSWCHS	ADC通道软件选择选择位 (需要ADCSWCHE=1才能生效) 00000: 选择通道0 00001: 选择通道1 ... 11001: 选择通道25 11010: 选择通道26 其他: 保留	0x0

### 19.5.3 ADC硬件触发控制寄存器(HWTG)

位	符号	描述	复位值
31	ADCEPWMZEN	ADC EPWM零点输出触发使能位 0: 禁止 1: 使能	0
30:28	ADCEPWMZSS	ADC EPWM零点输出触发源通道选择位 000: 触发源为EPWM0 001: 触发源为EPWM1 010: 触发源为EPWM2 011: 触发源为EPWM3 100: 触发源为EPWM4 101: 触发源为EPWM5 110: 触发源为EPWM6 111: 触发源为EPWM7	0x0
27	ADCEPWMPEN	ADC EPWM周期点输出触发使能位 0: 禁止 1: 使能	0
26:24	ADCEPWMPSS	ADC EPWM周期点输出触发源通道选择位 000: 触发源为EPWM0 001: 触发源为EPWM1 010: 触发源为EPWM2 011: 触发源为EPWM3 100: 触发源为EPWM4 101: 触发源为EPWM5 110: 触发源为EPWM6 111: 触发源为EPWM7	0x0
23:22	-	保留	-
21	ADCEPWMZDLYEN	ADC EPWM零点延时触发使能位 0: 使能 1: 禁止(不经过延时)	0
20	ADCEPWMPDLYEN	ADC EPWM周期点延时触发使能位 0: 使能 1: 禁止(不经过延时)	0
19:18	-	保留	-
17	ADCEXTEN	ADC 外部触发使能位 0: 禁止 1: 使能	0
16	ADCEXTES	ADC 外部触发边沿选择位 0: 下降沿 1: 上升沿	0
15	ADCINTTGEN	ADC 内部功能触发使能位 0: 禁止 1: 使能	0
14:12	ADCINTTGSS	ADC 内部功能触发源通道选择位 000: 保留 001: ADC转换结束信号	0x0

		010: ACMP0事件 011: ACMP1事件 100: Timer0中断信号 101: Timer1中断信号	
11	ADCPTG1DLYEN	ADC EPWM计数比较器1延时触发使能位 0: 使能 1: 禁止 (不经过延时)	0
10	ADCPTG0DLYEN	ADC EPWM计数比较器0延时触发使能位 0: 使能 1: 禁止 (不经过延时)	0
9	ADCPTG1EN	ADC EPWM计数比较器1触发使能位 0: 禁止 1: 使能	0
8	ADCPTG0EN	ADC EPWM计数比较器0触发使能位 0: 禁止 1: 使能	0
7	ADCEPWMTEN	ADC EPWM输出触发使能位 0: 禁止 1: 使能	0
6:4	ADCEPWTSS	ADC EPWM输出触发源通道选择位 000: 触发源为EPWM0 001: 触发源为EPWM1 010: 触发源为EPWM2 011: 触发源为EPWM3 100: 触发源为EPWM4 101: 触发源为EPWM5 110: 触发源为EPWM6 111: 触发源为EPWM7	0x0
3:2	-	保留	-
1:0	ADCPEWMTSP	ADC EPWMn触发方式选择位(n=0-7) 00: EPWMn波形的上升沿 01: EPWMn周期点 (IPGn) 10: EPWMn波形的下降沿 11: EPWMn的零点 (IPGn)	0x0

### 19.5.4 ADC EPWM触发延时寄存器(EPWMTGDLY)

位	符号	描述	复位值
31:10	-	保留	-
9:0	ADCEPWMTGDLY	ADC EPWM触发延时数据 EPWM (包括输出通道触发、EPWM零点触发、EPWM周期点触发与EPWM比较器0/1的触发) 延时触发ADC延时数据 (详见19.3.8ADC硬件触发启动章节EPWM触发延时部分)。	0x0

### 19.5.5 ADC扫描寄存器(SCAN)

位	符号	描述	复位值
31:27	-	保留	-
26:0	ADCEn	ADC通道n使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.6 ADC EPWM输出触发转换通道使能寄存器(CHEPWM)

位	符号	描述	复位值
31:27	-	保留	-
26:0	ADCCHPWMn	ADC EPWM输出触发转换通道使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.7 ADC EPWM周期点触发转换通道使能寄存器(CHPIFTG)

位	符号	描述	复位值
31:27	-	保留	-
26:0	ADCCHPIFTGn	ADC EPWM周期点触发转换通道使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.8 ADC EPWM零点触发转换通道使能寄存器(CHZIFTG)

位	符号	描述	复位值
31:27	-	保留	-
26:0	ADCCHZIFTGn	ADC EPWM零点触发转换通道使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.9 ADC EPWM比较器0触发转换通道使能寄存器(CHPTG0)

位	符号	描述	复位值
31:27	-	保留	-
26:0	ADCCHPTG0n	ADC EPWM比较器0触发转换通道使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.10 ADC EPWM比较器1触发转换通道使能寄存器(CHPTG1)

位	符号	描述	复位值
31:27	-	保留	-
26:0	ADCCHPTG1n	ADC EPWM比较器1触发转换通道使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.11 ADC测试寄存器(TEST)

位	符号	描述	复位值
31:24	ADCSWT	ADC模拟开关切换延时时间： (即从模拟开关打开到开始采样时间) (ADCSWT+2) ADC clock cycles 注：建议对弱信号进行ADC转换时延长该时间	0x0
23:0	-	保留，必须为0	0x0

### 19.5.12 ADC转换结果寄存器(RESULT)

位	符号	描述	复位值
31:12	-	保留	-
11:0	RESULT	ADC 转换结果，每次转换完成刷新该值	0x0

### 19.5.13 ADC通道转换结果寄存器(DATAx) x=0~26

位	符号	描述	复位值
31:12	-	保留	-
11:0	RESULT	ADC 转换结果	0x0

### 19.5.14 ADC比较控制寄存器0(CMPx) x=0~1

位	符号	描述	复位值
31	ADCCMPxEN	ADC比较器x使能位 0: 禁止 1: 使能	0
30	ADCCMPxO	ADC比较器x结果位(只读) (选择的通道转换完毕后自动更新该位) 0: 不满足比较的条件 1: 满足比较条件	0
29	-	保留	-
28	ADCCMPxCOND	ADC比较器x比较条件选择位 0: ADC结果<预设值 1: ADC结果>=预设值	0
27:24	ADCCMPxMCNT	ADC比较器x匹配次数预设值 指定的通道的模数转换结果和比较条件匹配时,内部计数器加1,当内部计数器等于ADCCMPxMCNT+1的值时,之后内部计数器值自动清零。若累加过程中不满足匹配条件,内部计数器值也将自动清零,即该功能具有滤波功能。 匹配的同时产生ADC比较事件,该事件可以作为触发EPWM的刹车操作的信号。 注意: ADC比较器0比较事件将置中断标志ADCCMP0IF为1;	0x0
23:21	-	保留	-
20:16	ADCCMPxCHS	ADC比较器x比较通道选择位 00000: 通道0 ..... 11001: 通道25 11010: 通道26 其他: 保留	0x0
15:12	-	保留	-
11:0	ADCCMPxDATA	ADC比较器x数据预设值(12位)	0x0

### 19.5.15 ADC中断使能寄存器(IMSC)

位	符号	描述	复位值
31	ADCIMSC31	ADC比较器0中断使能位 0: 禁止 1: 使能	0
30:27	-	保留	-
26:0	ADCIMSCn	ADC通道n中断使能位(n=26-0) 0: 禁止 1: 使能	0x0

### 19.5.16 ADC中断源状态寄存器(RIS)

位	符号	描述	复位值
31	ADC RIS31	ADC比较器0中断源状态 0: 中断源未产生中断 1: 中断源产生中断	0
30:27	-	保留	-
26:0	ADC RISn	ADC通道n中断源状态(n=26-0) 0: 中断源未产生中断 1: 中断源产生中断	0x0

### 19.5.17 ADC已使能中断状态寄存器(MIS)

位	符号	描述	复位值
31	ADC MIS31	ADC比较器0中断状态 0: 未产生中断 1: 使能且产生中断	0
30:27	-	保留	-
26:0	ADC MISn	ADC通道n中断状态(n=26-0) 0: 未产生中断 1: 使能且产生中断	0x0

### 19.5.18 ADC中断清零寄存器(ICLR)

位	符号	描述	复位值
31	ADC ICLR31	写1清零ADC比较器0中断状态 写0不影响	0
30:27	-	保留	-
26:0	ADC ICLRn	写1清零ADC通道n中断状态(n=26-0) 写0不影响	0x0

### 19.5.19 ADC写使能控制寄存器(LOCK)

位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0x55时, 使能操作ADC相关寄存器 (详见寄ADC寄存器映射说明) LOCK=其他值时, 禁止操作ADC相关寄存器	0x0

## 第20章 可编程增益放大器 (PGA0/1/2/3)

### 20.1 概述

芯片包含四路可编程增益放大器。利用少量外围元器件可实现基本的信号放大与信号运算功能。

### 20.2 特性

#### PGA0 (可编程增益放大器0)

- ◆ 增益可调节：1X/2X/2.5X/5X/7.5X/10X/15X。
- ◆ 正端输入：PGA0P；负端输入：PGA0N
- ◆ 参考电压可选：BGR (0.8V) 或 VREF/2
- ◆ PGA0 可选择多种输出方式：
  - (1) 可直接输出到 ADC 第 0、1 通道
  - (2) 可直接输出到比较器
  - (3) 可直接输出到 PAD (A00)
  - (4) 可通过 10K 电阻后输出到 PAD (A00)

#### PGA1 (可编程增益放大器1)

- ◆ 增益可调节：1X/2X/2.5X/5X/7.5X/10X/15X。
- ◆ 正端输入：PGA1P；负端输入：PGA1N
- ◆ 参考电压：VREF/2
- ◆ PGA1 输出方式可选：
  - (1) 可直接输出到 ADC 第 2 通道
  - (2) 可直接输出到比较器

#### PGA2 (可编程增益放大器2)

- ◆ 增益可调节：1X/2X/2.5X/5X/7.5X/10X/15X。
- ◆ 正端输入：PGA2P；负端输入：PGA2N
- ◆ 参考电压：VREF/2
- ◆ PGA2 输出方式可选：
  - (1) 可直接输出到 ADC 第 3 通道
  - (2) 可直接输出到比较器

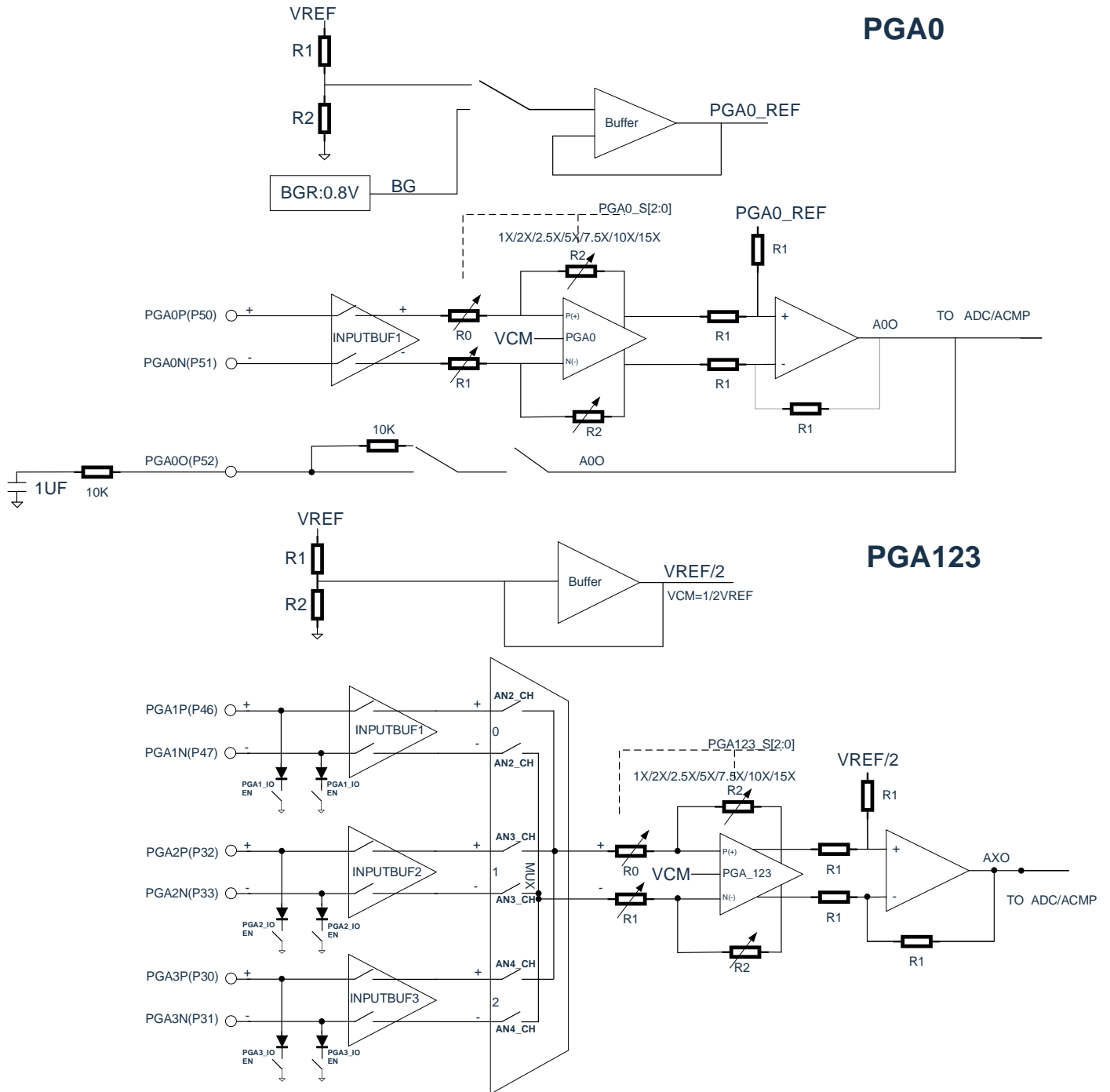
#### PGA3 (可编程增益放大器3)

- ◆ 增益可调节：1X/2X/2.5X/5X/7.5X/10X/15X。
- ◆ 正端输入：PGA3P；负端输入：PGA3N
- ◆ 参考电压：VREF/2
- ◆ PGA2 输出方式可选：
  - (1) 可直接输出到 ADC 第 4 通道
  - (2) 可直接输出到比较器



## 20.3 结构框图

图20-1 PGA结构图



## 20.4 寄存器映射

(PGA基地址 = 0x4006\_8300) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PGA0CON0	0x000	R/W	PGA0控制寄存器0	0x00000000
PGANCON0	0x004	R/W	PGA1/2/3控制寄存器0	0x00000000
PGANCON1	0x008	R/W	PGA1/2/3控制寄存器1	0x00000000
PGANSW	0x00C	R/W	PGA1/2/3输入通道选择寄存器	0x00000000
PGALOCK	0x010	R/W	PGA访问寄存器使能	0x00000000

## 20.5 寄存器说明

### 20.5.1 PGA0控制寄存器0 PGA0CON0

位	符号	描述	复位值
31:19	-	保留	-
18	PGA0_S_DRV	PGA0 INPUTBUF驱动电流选择信号 0: 20uA电流, 外挂1K/20K电阻时使用 1: 40uA电流, 外挂1K电阻时选择此档提升驱动能力	0
17	PGA0_OTEN	PGA0输出到PAD通道使能 0: 禁止 1: 使能	0
16	PGA0_OT_SELR	PGA0输出到PAD串联电阻选择 0: 内部不串电阻 1: 内部串10K电阻	0
15:13	-	保留	-
12	PGA0_EN	PGA0使能位 0: 禁止 1: 使能	0
11:9	-	保留, 必须为0	0
8	PGA0S_VREF	PGA0参考电压选择位 0: VREF/2 1: BG (0.8V)	0
7	-	保留	-
6:4	PGA0_BW	PGA0前端滤波带宽选择 000: 290ns 111: 30ns Step Size: 37ns	0x0
3	-	保留	-
2:0	PGA0_S	PGA0增益选择 000: 1X 001: 2X 010: 2.5X 011: 5X 100: 7.5X 101: 10X 110: 15X 111: 15X	0x0

## 20.5.2 PGA1/2/3控制寄存器0 PGANCON0

位	符号	描述	复位值
31:16	-	保留	-
15	PGA2_EN	PGA2使能位 0: 禁止 1: 使能	0
14:11	-	保留	-
10:8	PGA2_BW	PGA2前端带宽滤波选择 000: 290ns 111: 30ns Step Size :37ns	0x0
7	PGA1_EN	PGA1使能位 0: 禁止 1: 使能	0
6:3	-	保留	-
2:0	PGA1_BW	PGA1前端带宽滤波选择 000: 290ns 111: 30ns Step Size :37ns	0x0

## 20.5.3 PGA1/2/3控制寄存器1 PGANCON1

位	符号	描述	复位值
31:14	-	保留	-
13	PGA123_S_DRV	PGA123 INPUTBUF驱动电流选择信号 0: 20uA电流, 外挂1K/20K电阻时使用 1: 40uA电流, 外挂1K电阻时选择此档提升驱动能力	0
12	-	必须为0	0
11	-	保留	-
10:8	PGA123_S	PGA123 增益选择 000: 1X 001: 2X 010: 2.5X 011: 5X 100: 7.5X 101: 10X 110: 15X 111: 15X	0x0
7	PGA3_EN	PGA3使能位 0: 禁止 1: 使能	0
6:3	-	保留	-
2:0	PGA3_BW	PGA3前端带宽滤波选择 000: 290ns 111: 30ns Step Size :37ns	0x0

## 20.5.4 PGA1/2/3输入通道开关寄存器1 PGANSW

位	符号	描述	复位值
31:16	PGA123SW_SEL	PGA1/2/3输入通道开关选择位 0x55aa: 由PGA123_SW控制 其他: 由ADC通道开关控制, 和PGA123_SW状态无关。 PGA1_SW对应ADC通道2, PGA2_SW对应ADC通道3, PGA3_SW对应ADC通道4	0x0
15:11	-	保留	-
10	PGA3_IOEN	PGA3的IO接入钳位二极管使能 0: 禁止 1: 使能	0
9	PGA2_IOEN	PGA2的IO接入钳位二极管使能 0: 禁止 1: 使能	0
8	PGA1_IOEN	PGA1的IO接入钳位二极管使能 0: 禁止 1: 使能	0
7:2	-	保留	-
1:0	PGA123_SW	PGA1/2/3通道开关选择 00: PGA1/2/3通道禁止 01: PGA1通道开关使能, PGA2/3通道禁止 10: PGA2通道开关使能, PGA1/3通道禁止 11: PGA3通道开关使能, PGA1/2通道禁止	0x0

## 20.5.5 PGA访问寄存器使能 PGALOCK

位	符号	描述	复位值
31:8	-	保留	-
7:0	PGA_LOCK	PGA0/1/2/3寄存器访问使能位 0x55: 运行访问PGA0/1/2/3相关寄存器 其他: 禁止访问	0x0

# 第21章 模拟比较器 (ACMP0/1)

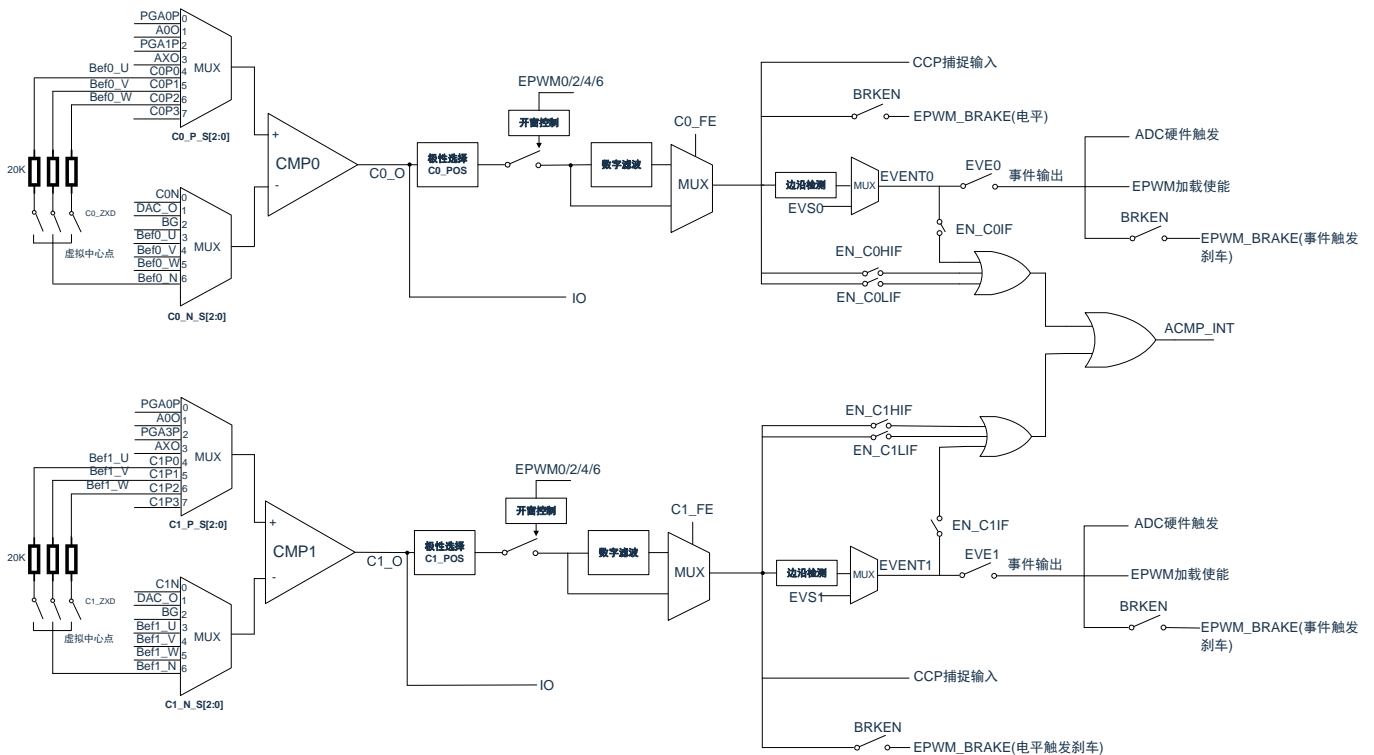
## 21.1 概述

芯片内部包含两个模拟比较器。可按照比较器的配置适用于不同的应用场合。当正端电压大于负端电压时，比较器输出逻辑1，反之输出0。该比较器支持以下功能：

- (1) 可以通过输出极性选择位进行改变。
- (2) 具有开窗功能，开窗极性可设置。
- (3) 输出可滤波，滤波时间可选择。
- (4) 比较器的输出可以作为CCP1的CAP3的捕捉输入。
- (5) 比较器产生事件输出可作为ADC的硬件触发。
- (6) 比较器输出和产生事件输出也均可作为增强型PWM的刹车触发信号。
- (7) 当比较器输出值发生改变时，每路比较器都可通过配置产生中断。

## 21.2 结构框图

图 21-1：比较器结构框图



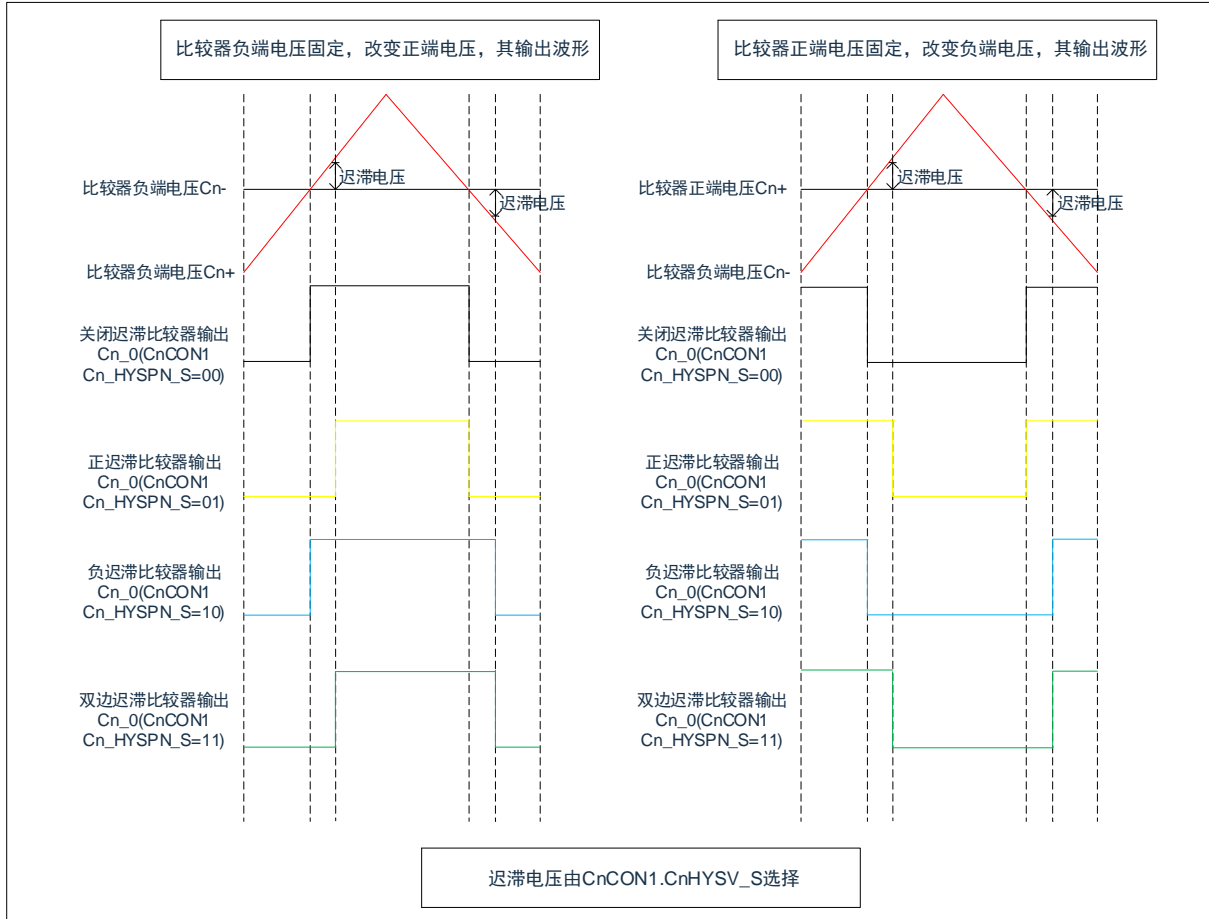
## 21.3 特性

- ◆ 模拟输入电压范围：(0~VDD)V。
- ◆ 支持单边/双边迟滞功能。
- ◆ 支持迟滞电压选择(10mV/20mV/60mV-典型值)。
- ◆ 每个比较器正端可多路选择
- ◆ 每个比较器负端可选择端口输入与内部参考电压。
- ◆ 输出可滤波时间可选择：0~512\* $T_{sys}$ 。
- ◆ 比较器输出极性可选。
- ◆ 比较器输出和产生事件输出均可作为增强型 PWM 的刹车触发信号。
- ◆ 事件产生以及比较器的输出电平均可产生中断。
- ◆ 支持开窗功能，开窗极性可设置。
- ◆ ACMP0/1 均有四路 EPWM 开窗通道可选。

## 21.4 功能说明

### 21.4.1 迟滞功能

图 21-2: 比较器迟滞功能结构框图

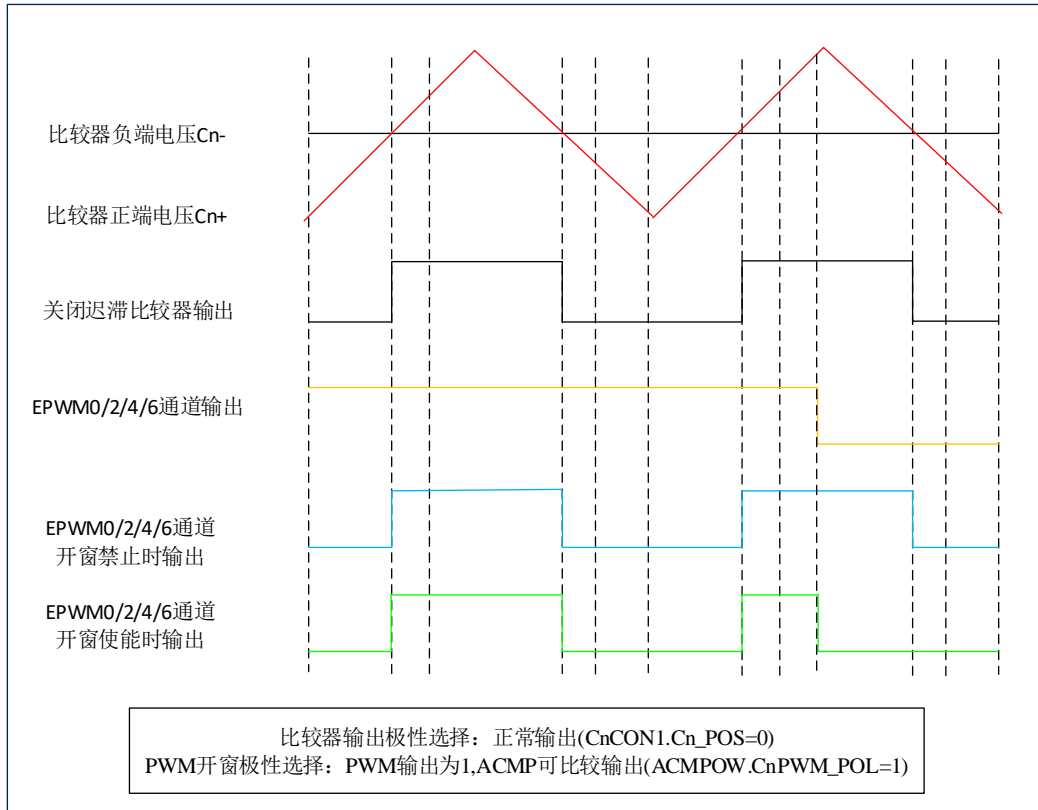


注：n=0, 1。

## 21.4.2 开窗控制

开窗控制功能结构框图如下图所示。

图 21-3: 开窗控制功能结构框图



注: n=0, 1。

当模拟比较器 0/1 的 EPWM0/2/4/6 通道的开窗禁止时, 开窗控制的输出为比较器极性选择后的输出。

当 PWM 开窗极性选择位(ACMPOW.CnPWM\_POL)为 1 时, EPWM0/2/4/6 通道的开窗使能, 开窗控制的输出在 EPWM0/2/4/6 通道输出为 1 时, 为比较器极性选择后的输出, 在 EPWM0/2/4/6 通道输出为 0 时, 开窗控制输出为 0。

当 PWM 开窗极性选择位(ACMPOW.CnPWM\_POL)为 0 时, EPWM0/2/4/6 通道的开窗使能, 开窗控制的输出在 EPWM0/2/4/6 通道输出为 0 时, 为比较器极性选择后的输出, 在 EPWM0/2/4/6 通道输出为 1 时, 开窗控制输出为 0。



### 21.4.3 滤波功能

比较器模块的主时钟为 PCLK，数字滤波也以此时钟为滤波时钟，设置滤波使能后会对开窗控制后的输出信号进行滤波，即只有信号稳定时间超过设定的滤波设置时间才能通过滤波器，滤波后的信号才会发生变化，如果输入信号稳定时间不足滤波设置时间就发生变化，则滤波后的信号维持原值不变。

设置滤波使能后会发生以下几种情况：

- (1) 滤波输入信号宽度 < 滤波设置时间：无信号输出。
- (2) 滤波输入信号宽度 = 滤波设置时间：输出一个时钟周期的脉冲信号（无法使 EPWM 刹车）。
- (3) 滤波输入信号宽度 > 滤波设置时间：正常滤波输出。

注：滤波时间由 CnCON1.Cn\_FS[3:0] 设置，n=0, 1。

## 21.5 寄存器映射

(ACMP基地址 = 0x4006\_8200) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
C0CON0 <sub>(P1B)</sub>	0x00	R/W	模拟比较器0控制寄存器0	0x00
C0CON1 <sub>(P1B)</sub>	0x04	R/W	模拟比较器0控制寄存器1	0x00
C1CON0 <sub>(P1B)</sub>	0x08	R/W	模拟比较器1控制寄存器0	0x00
C1CON1 <sub>(P1B)</sub>	0x0C	R/W	模拟比较器1控制寄存器1	0x00
CVECON <sub>(P1B)</sub>	0x10	R/W	模拟比较器事件控制寄存器	0x00
IMSC <sub>(P1B)</sub>	0x14	R/W	模拟比较器中断使能寄存器	0x00
RIS	0x18	RO	模拟比较器中断源状态寄存器	0x00
MIS	0x1C	RO	模拟比较器已使能中断状态寄存器	0x00
ICLR	0x20	WO	模拟比较器中断清零寄存器	0x00
ACMPOW <sub>(P1B)</sub>	0x24	R/W	模拟比较器开窗寄存器	0x00
LOCK	0x28	R/W	模拟比较器写使能寄存器	0x00

注1: (P1B)标注的寄存器为被保护的寄存器。

注2: (P1B): LOCK=55H时, 标注的寄存器允许写入; =其他值, 禁止写入。

## 21.6 寄存器说明

### 21.6.1 模拟比较器0控制寄存器0(C0CON0)

位	符号	描述	复位值
31:17	-	保留	-
16	C0_ZXD	模拟比较器0中心点选择使能 0: 禁止 1: 使能	0
15	C0_EN	模拟比较器0使能位 0: 禁止 1: 使能	0
14	C0_OEN	模拟比较器0输出使能位 0: 禁止 1: 使能	0
13:9	-	保留	-
8	C0_OUT	模拟比较器0结果位(只读)	0
7	-	保留	-
6:4	C0_P_S	模拟比较器0正端通道选择 000: PGA0P 001: A0O 010: PGA1P 011: AXO 100: C0P0 101: C0P1 110: C0P2 111: C0P3	0x0
3	-	保留	-
2:0	C0_N_S	模拟比较器0负端通道选择 000: C0N 001: DAC_O 010: BG (0.8V) 011: BEF0_U (C0P0) 100: BEF0_V (C0P1) 101: BEF0_W (C0P2) 110: BEF0_N 111: 禁止选择	0x0

## 21.6.2 模拟比较器0控制寄存器1(C0CON1)

位	符号	描述	复位值
31:14	-	保留	-
13:12	C0_HYSPN_S	模拟比较器0迟滞方式选择 00: 无迟滞 01: 正迟滞 10: 负迟滞 11: 正负迟滞	0x0
11:10	C0_HYSV_S	模拟比较器0迟滞电压选择 00: 无迟滞 01: 10mV 10: 20mV 11: 60mV	0x0
9	C0_POS	模拟比较器0输出极性选择位 0: 正常输出 1: 反相输出	0
8	C0_FE	模拟比较器0输出滤波使能位 0: 禁止 1: 使能	0
7:4	-	保留	-
3:0	C0_FS	模拟比较器0输出滤波时间选择位 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk 其他: (0~1)*Tpclk	0x0

### 21.6.3 模拟比较器1控制寄存器0(C1CON0)

位	符号	描述	复位值
31:17	-	保留	-
16	C1_ZXD	模拟比较器1中心点选择使能 0: 禁止 1: 使能	0
15	C1_EN	模拟比较器1使能位 0: 禁止 1: 使能	0
14	C1_OEN	模拟比较器1输出使能位 0: 禁止 1: 使能	0
13:9	-	保留	-
8	C1_OUT	模拟比较器1结果位(只读)	
7	-	保留	-
6:4	C1_P_S	模拟比较器1正端通道选择 000: PGA0P 001: A0O 010: PGA3P 011: AXO 100: C1P0 101: C1P1 110: C1P2 111: C1P3	0x0
3	-	保留	-
2:0	C1_N_S	模拟比较器1负端通道选择 000: C1N 001: DAC_O 010: BG(0.8V) 011: BEF1_U (C1P0) 100: BEF1_V (C1P1) 101: BEF1_W (C1P2) 110: BEF1_N 其他: 禁止	0x0

## 21.6.4 模拟比较器1控制寄存器1(C1CON1)

位	符号	描述	复位值
31:14	-	保留	-
13:12	C1HYSPN_S	模拟比较器1迟滞方式选择 00: 无迟滞 01: 正迟滞 10: 负迟滞 11: 正负迟滞	0x0
11:10	C1_HYSV_S	模拟比较器1迟滞电压选择 00: 无迟滞 01: 10mV 10: 20mV 11: 60mV	0x0
9	C1_POS	模拟比较器1输出极性选择位 0: 正常输出 1: 反相输出	0
8	C1_FE	模拟比较器1输出滤波使能位 0: 禁止 1: 使能	0
7:4	-	保留	-
3:0	C1_FS	模拟比较器1输出滤波时间选择位 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk 其他: (0~1)*Tpclk	0x0

### 21.6.5 模拟比较器事件控制寄存器(CEVCON)

位	符号	描述	复位值
31:6	-	保留	-
5	EVE1	模拟比较器1事件输出使能位(不影响中断产生) 0: 禁止 1: 使能	0
4	EVE0	模拟比较器0事件输出使能位(不影响中断产生) 0: 禁止 1: 使能	0
3:2	EVS1	模拟比较器1事件产生条件选择位 00: 比较器1输出从0->1的跳变 01: 比较器1输出从1->0的跳变 10: 比较器1输出从0->1的跳变或从1->0的跳变 11: 保留	0x0
1:0	EVS0	模拟比较器0事件产生条件选择位 00: 比较器0输出从0->1的跳变 01: 比较器0输出从1->0的跳变 10: 比较器0输出从0->1的跳变或从1->0的跳变 11: 保留	0x0

### 21.6.6 模拟比较器中断使能寄存器(IMSC)

位	符号	描述	复位值
31:7	-	保留	-
6	EN_C1LIF	模拟比较器1输出低电平中断使能位 0: 禁止 1: 允许	0
5	EN_C1HIF	模拟比较器1输出高电平中断使能位 0: 禁止 1: 允许	0
4	EN_C1IF	模拟比较器1事件中断使能位 0: 禁止 1: 允许	0
3	-	保留	-
2	EN_C0LIF	模拟比较器0输出低电平中断使能位 0: 禁止 1: 允许	0
1	EN_C0HIF	模拟比较器0输出高电平中断使能位 0: 禁止 1: 允许	0
0	EN_C0IF	模拟比较器0事件中断使能位 0: 禁止 1: 允许	0

## 21.6.7 模拟比较器中断源状态寄存器(RIS)

位	符号	描述	复位值
31:7	-	保留	-
6	RIS_C1LIF	模拟比较器1低电平中断状态位 0: 未产生中断 1: 已产生中断(低电平产生)	0
5	RIS_C1HIF	模拟比较器1高电平中断状态位 0: 未产生中断 1: 已产生中断(高电平产生)	0
4	RIS_C1IF	模拟比较器1事件中断状态位 0: 未产生中断 1: 已产生中断(事件产生)	0
3	-	保留	-
2	RIS_C0LIF	模拟比较器0低电平中断状态位 0: 未产生中断 1: 已产生中断(低电平产生)	0
1	RIS_C0HIF	模拟比较器0高电平中断状态位 0: 未产生中断 1: 已产生中断(高电平产生)	0
0	RIS_C0IF	模拟比较器0事件中断状态位 0: 未产生中断 1: 已产生中断(事件产生)	0



### 21.6.8 模拟比较器已使能中断源状态寄存器(MIS)

位	符号	描述	复位值
31:7	-	保留	-
6	MIS_C1LIF	模拟比较器1低电平输出中断标志位, 已使能中断状态位 0: 未产生中断 1: 已产生中断	0
5	MIS_C1HIF	模拟比较器1高电平输出中断标志位, 已使能中断状态位 0: 未产生中断 1: 已产生中断	0
4	MIS_C1IF	模拟比较器1事件中断标志位, 已使能中断状态位 0: 未产生中断 1: 已产生中断	0
3	-	保留	-
2	MIS_C0LIF	模拟比较器0低电平输出中断标志位, 已使能中断状态位 0: 未产生中断 1: 已产生中断	0
1	MIS_C0HIF	模拟比较器0高电平输出中断标志位, 已使能中断状态位 0: 未产生中断 1: 已产生中断	0
0	MIS_C0IF	模拟比较器0事件中断标志位, 已使能中断状态位 0: 未产生中断 1: 已产生中断	0

### 21.6.9 模拟比较器中断清零控制寄存器(ICLR)

位	符号	描述	复位值
31:7	-	保留	-
6	ICLR_C1LIF	模拟比较器1低电平中断清零控制位 写0: 不影响 写1: 清零RIS_C1LIF标志位	0
5	ICLR_C1HIF	模拟比较器1高电平中断清零控制位 写0: 不影响 写1: 清零RIS_C1HIF标志位	0
4	ICLR_C1IF	模拟比较器1事件中断清零控制位 写0: 不影响 写1: 清零RIS_C1IF标志位	0
3	-	保留	-
2	ICLR_C0LIF	模拟比较器0低电平中断清零控制位 写0: 不影响 写1: 清零RIS_C0LIF标志位	0
1	ICLR_C0HIF	模拟比较器0高电平中断清零控制位 写0: 不影响 写1: 清零RIS_C0HIF标志位	0
0	ICLR_C0IF	模拟比较器0事件中断清零控制位 写0: 不影响 写1: 清零RIS_C0IF标志位	0

### 21.6.10 模拟比较器开窗控制寄存器 (ACMPOW)

位	符号	描述	复位值
31:16	-	保留	0
15	C1PWM_POL	模拟比较器1 PWM开窗极性选择 0: PWM输出0时, ACMP1可比较输出 1: PWM输出1时, ACMP1可比较输出	0
14:12	-	保留	-
11	C1PWM6_WEN	模拟比较器1的EPWM6通道开窗使能 0: 开窗禁止 1: 开窗使能	0
10	C1PWM4_WEN	模拟比较器1的EPWM4通道开窗使能 0: 开窗禁止 1: 开窗使能	0
9	C1PWM2_WEN	模拟比较器1的EPWM2通道开窗使能 0: 开窗禁止 1: 开窗使能	0
8	C1PWM0_WEN	模拟比较器1的EPWM0通道开窗使能 0: 开窗禁止 1: 开窗使能	0
7	C0PWM_POL	模拟比较器0 PWM开窗极性选择 0: PWM输出0时, ACMP0可比较输出 1: PWM输出1时, ACMP0可比较输出	0
6:4	-	保留	-
3	C0PWM6_WEN	模拟比较器0的EPWM6通道开窗使能 0: 开窗禁止 1: 开窗使能	0
2	C0PWM4_WEN	模拟比较器0的EPWM4通道开窗使能 0: 开窗禁止 1: 开窗使能	0
1	C0PWM2_WEN	模拟比较器0的EPWM2通道开窗使能 0: 开窗禁止 1: 开窗使能	0
0	C0PWM0_WEN	模拟比较器0的EPWM0通道开窗使能 0: 开窗禁止 1: 开窗使能	0

### 21.6.11 模拟比较器写使能控制寄存器(LOCK)

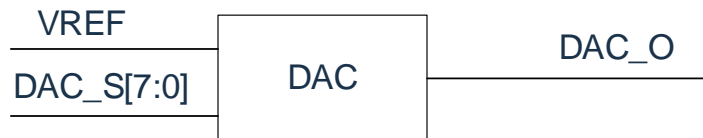
位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0x55时, 使能操作ACMP相关寄存器 (详见寄ACMP寄存器映射说明) LOCK=其他值时, 禁止操作ACMP相关寄存器	0x0

## 第22章 DAC

### 22.1 概述

芯片内部包含一个数模转换器

### 22.2 结构框图



### 22.3 特性

- ◆ 模拟参考电压输入可选 ADCLDO 的输出。
- ◆ 输出电压多级可选。

### 22.4 寄存器映射

(DAC基地址 = 0x4006\_8360)      RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CON0 <sub>(P1B)</sub>	0x000	R/W	DAC控制寄存器0	0x0
LOCK	0x004	R/W	DAC寄存器使能控制位	0x0

注1: (P1B)标注的寄存器为被保护的寄存器。

注2: (P1B): LOCK==55H时, 标注的寄存器允许写入; =其他值, 禁止写入。

## 22.5 寄存器说明

### 22.5.1 DAC控制寄存器0(CON0)

位	符号	描述	复位值
31:17	-	保留	-
17	DAC_OEN2	DAC输出到端口使能 1: 使能输出到端口 P53 0: 禁止输出到端口	0
16	DAC_OEN1	DAC输出到端口使能 1: 使能输出到端口 P12 0: 禁止输出到端口	0
15:9	-	保留	-
8	DAC_EN	DAC模块使能 1: 使能 0: 禁止	0
7:0	DAC_S	DAC数字信号输入	0x0

### 22.5.2 DAC写使能控制寄存器(LOCK)

位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0x55时, 使能操作DAC相关寄存器 (详见寄DAC存器映射说明) LOCK=其他值时, 禁止操作DAC相关寄存器	0x0

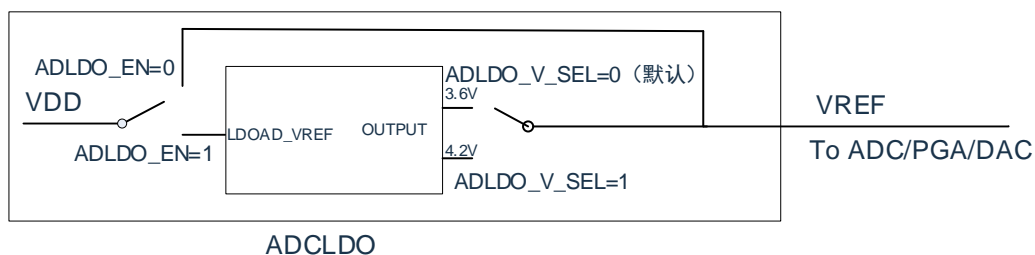
## 第23章 ADCLDO概述

芯片内部包含一个LDO，可向部分模块提供参考电压

### 23.1 特性

- ◆ 模拟输入电压范围：VDD。
- ◆ 输出电压：VDD，4.2V，3.6V 可选。

### 23.2 结构框图



### 23.3 寄存器映射

(ADCLDO基地址 = 0x4006\_8340) RO: 只读, WO: 只写, RW: 读写

寄存器	偏移量	读/写	描述	复位值
CON0(P1B)	0x000	R/W	ADCLDO控制寄存器0	0x100000
LOCK	0x00C	R/W	ADCLDO寄存器使能控制位	0x0

注1: (P1B)标注的寄存器为被保护的寄存器。

注2: (P1B): LOCK==55H时, 标注的寄存器允许写入; =其他值, 禁止写入。

## 23.4 寄存器说明

### 23.4.1 ADCLDO控制寄存器0(CON0)

位	符号	描述	复位值
31:9	-	保留	-
8	ADCLDO_EN	ADCLDO模块使能 0: 禁止, ADLDO输出VDD 1: 使能, ADLDO输出LDO电压	0
7:0	ADCLDO_V_SEL	ADCLDO输出电压选择 0x55: LDO电压输出4.2V 其他: LDO电压输出3.6V	0x0

### 23.4.2 ADCLDO写使能控制寄存器(LOCK)

位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0x55时, 使能操作ADCLDO相关寄存器 (详见寄ADCLDO存器映射说明) LOCK=其他值时, 禁止操作ADCLDO相关寄存器	0x0

## 第24章 嵌套向量中断控制器(NVIC)

Cortex®-M0+ CPU提供一个用于中断处理的嵌套向量中断控制器（NVIC）。

### 24.1 特性

- ◆ 支持嵌套向量中断。
- ◆ 自动保存和恢复处理器状态。
- ◆ 动态改变优先级。
- ◆ 简化和确定的中断时间。

NVIC按照优先级处理所有支持的异常。所有的异常在“Handler模式”处理。NVIC支持23个(IRQ[31:0])离散中断，每个中断支持4级中断优先级。所有的中断和大部分系统异常可以配置成不同的优先级。当一个中断发生时，NVIC将比较新中断与当前中断的优先级，如果新中断优先级高，则立即处理新中断。

当接受一个中断后，中断服务程序（ISR）的开始地址可从内存中的向量表取得。软件不需要决定哪个中断被响应，也不用分配相关ISR的开始地址。当取得开始地址后，NVIC将自动保存处理器状态寄存器（PC、PSR、LR、R0~R3、R12）的值到栈中。在ISR结束后，NVIC将从栈中恢复相关寄存器的值，并运行在正常状态。因此花费少量且确定的时间处理中断请求。

NVIC支持“末尾连锁”，可以有效的处理背对背中断，即无需保存和恢复当前状态，从而减少结束当前ISR切换到挂起ISR的延迟时间。NVIC还支持“Late Arrival”，因此可以提高并发中断的效率。当较高优先级中断请求发生在当前ISR开始执行之前（保存处理器状态和获取起始地址阶段），NVIC将立即处理更高优先级的中断，从而提高实时性。

更多详细信息，请参考“ARM®Cortex®-M0+ 技术参考手册”和“ARM®v6-M 架构参考手册”。

### 24.2 异常模式和系统中断映射

下表列出了该系列产品支持的异常模式。与所有中断一样，软件可以为其中一些异常设置4级优先级。用户可配置的最高优先级为0，最低优先级为3。所有用户可配置中断的默认优先级为0。

异常名称	异常编号	优先级
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4~10	保留
SVCall	11	可配置
Reserved	12~13	保留
PendSV	14	可配置
SysTick	15	可配置
Interrupt (IRQ0~IRQ31)	16~47	可配置

注：优先级0在系统为第4优先级，排在“Reset”、“NMI”和“Hard Fault”三个系统异常之后。

## 24.3 向量表

异常编号	中断号	向量地址	异常类型	DMA	描述
1-15	-	0x00-0x3c	系统异常		
16	0	0x40	INTLVI		电压检测
17	1	0x44	INTP0		引脚输入的边沿检测
18	2	0x48	INTP1		引脚输入的边沿检测
19	3	0x4c	INTP2		引脚输入的边沿检测
20	4	0x50	INTP3		引脚输入的边沿检测
21	5	0x54	INTTM01H		定时器通道01计数或捕捉结束（高8位定时器工作）
22	6	0x58	INTCCP		CCP中断
23	7	0x5c	INTACMP		比较器完成中断
24	8	0x60	INTADC	DTC1	ADC中断
25	9	0x64	INTSPI	DTC2	SPI中断
26	10	0x68	INTIICA0	DTC3	IIC中断
27	11	0x6c	INTUART0	DTC4	UART0中断
28	12	0x70	INTUART1	DTC5	UART1中断
29	13	0x74	INTEPWM	DTC6	EPWM中断
30	14	0x78	INTTIMER0	DTC7	TIMER0中断
31	15	0x7c	INTTIMER1		TIMER1中断
32	16	0x80	INTHALL		HALL中断
33	17	0x84	INTLSITIMER		LSI定时器中断
34	18	0x88	保留		
35	19	0x8c	INTTM00		定时器通道00计数结束或捕捉结束
36	20	0x90	INTTM01		定时器通道01计数结束或捕捉结束
37	21	0x94	INTTM02		定时器通道02计数结束或捕捉结束
38	22	0x98	INTTM03		定时器通道03计数结束或捕捉结束
39	23	0x9c	保留		
40	24	0xa0	保留		
41	25	0xa4	保留		
42	26	0xa8	保留		
43	27	0xac	保留		
44	28	0xb0	保留		
45	29	0xb4	保留		
46	30	0xb8	保留		
47	31	0xbc	INTFL	DTC0	FLASH编程完成



## 24.4 寄存器映射

(NVIC基地址 = 0xE000\_E000) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
ISER	0x100	R/W	中断设置使能控制寄存器	0x0
ICER	0x180	R/W	中断清除使能控制寄存器	0x0
ISPR	0x200	R/W	中断设置挂起控制寄存器	0x0
ICPR	0x280	R/W	中断清除挂起控制寄存器	0x0
IPR0	0x400	R/W	IRQ0~IRQ3中断优先级寄存器	0x0
IPR1	0x404	R/W	IRQ4~IRQ7中断优先级寄存器	0x0
IPR2	0x408	R/W	IRQ8~IRQ11中断优先级寄存器	0x0
IPR3	0x40C	R/W	IRQ12~IRQ15中断优先级寄存器	0x0
IPR4	0x410	R/W	IRQ16~IRQ19中断优先级寄存器	0x0
IPR5	0x414	R/W	IRQ20~IRQ23中断优先级寄存器	0x0
IPR6	0x418	R/W	IRQ24~IRQ27中断优先级寄存器	0x0
IPR7	0x41C	R/W	IRQ28~IRQ31中断优先级寄存器	0x0

(INTM基地址 = 0x4004\_5B38) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
EGP0	0x000	R/W	外部中断上升沿允许寄存器	0x0
EGN0	0x001	R/W	外部中断下降沿允许寄存器	0x0

## 24.5 寄存器说明

### 24.5.1 中断设置使能控制寄存器 (ISER)

位	符号	描述	复位值
31:0	SETENA	中断使能位 使能一个或多个中断。每一位代表一个从IRQ0~IRQ31的中断（向量号从16~47）。 写操作： 0: 无效 1: 写1使能相关中断 读操作： 0: 相关中断状态禁止 1: 相关中断状态使能 注：读该寄存器值表明当前使能状态。	0x0

### 24.5.2 中断清使能控制寄存器 (ICER)

位	符号	描述	复位值
31:0	CLRENA	中断禁用位 禁用一个或多个中断。每一位表示一个从IRQ0~IRQ31的中断（向量号从16~47）。 写操作： 0: 无效 1: 写1禁止相关中断 读操作： 0: 相关中断状态禁止 1: 相关中断状态使能 注：读该寄存器值表明当前使能状态。	0x0

### 24.5.3 中断设置挂起控制寄存器 (ISPR)

位	符号	描述	复位值
31:0	SETPEND	设置中断挂起位 写操作： 0: 无效 1: 写1设置挂起状态。每一位表示一个从IRQ0~IRQ31的中断（向量号从16~47） 读操作： 0: 相关中断不在挂起状态 1: 相关中断在挂起状态 注：读该寄存器值表明当前挂起状态。	0x0

### 24.5.4 中断清挂起控制寄存器 (ICPR)

位	符号	描述	复位值
31:0	CLRPEND	清中断挂起位 写操作： 0: 无效 1: 写1清除挂起状态。每一位表示一个从IRQ0~IRQ31的中断（向量号从16~47） 读操作： 0: 相关中断不在挂起状态 1: 相关中断在挂起状态 注：读该寄存器值表明当前挂起状态。	0x0

### 24.5.5 IRQ0~IRQ3中断优先级寄存器 (IPR0)

位	符号	描述	复位值
31:30	PRI_3	IRQ3 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_2	IRQ2 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_1	IRQ1 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_0	IRQ0 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.6 IRQ4~IRQ7中断优先级寄存器 (IPR1)

位	符号	描述	复位值
31:30	PRI_7	IRQ7 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_6	IRQ6 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_5	IRQ5 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_4	IRQ4 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.7 IRQ8~IRQ11中断优先级寄存器 (IPR2)

位	符号	描述	复位值
31:30	PRI_11	IRQ11 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_10	IRQ10 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_9	IRQ9 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_8	IRQ8 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.8 IRQ12~IRQ15中断优先级寄存器 (IPR3)

位	符号	描述	复位值
31:30	PRI_15	IRQ15 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_14	IRQ14 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_13	IRQ13 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_12	IRQ12 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.9 IRQ16~IRQ19中断优先级寄存器 (IPR4)

位	符号	描述	复位值
31:30	PRI_19	IRQ19 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_18	IRQ18 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_17	IRQ17 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_16	IRQ16 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.10 IRQ20~IRQ23中断优先级寄存器 (IPR5)

位	符号	描述	复位值
31:30	PRI_23	IRQ23 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_22	IRQ22 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_21	IRQ21 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_20	IRQ20 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.11 IRQ24~IRQ27中断优先级寄存器 (IPR6)

位	符号	描述	复位值
31:30	PRI_27	IRQ27 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_26	IRQ26 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_25	IRQ25 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_24	IRQ24 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.12 IRQ28~IRQ31中断优先级寄存器 (IPR7)

位	符号	描述	复位值
31:30	PRI_31	IRQ31 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
29:24	-	保留	-
23:22	PRI_30	IRQ30 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
21:16	-	保留	-
15:14	PRI_29	IRQ29 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
13:8	-	保留	-
7:6	PRI_28	IRQ28 优先级 0 表示最高优先级, 3 表示最低优先级	0x0
5:0	-	保留	-

### 24.5.13 外部中断上升沿允许寄存器 (EGP0)

EGP0和EGN0寄存器用于设定INTP0~INTP3的有效边沿。通过8位存储器操作指令设定EGP0、EGN0寄存器。在产生复位信号后，这些寄存器的值变为“00H”。

位	符号	描述	复位值
7:4	-	保留	0x0
3	EGP3	INTP3 中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
2	EGP2	INTP2 部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
1	EGP1	INTP1 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
0	EGP0	INTP0 外部中断上升沿允许寄存 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0

### 24.5.14 外部中断下降沿允许寄存器 (EGN0)

位	符号	描述	复位值
7:4	-	保留	0x0
3	EGN3	INTP3 外部中断下降沿允许寄存器： 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
2	EGN2	INTP2 外部中断下降沿允许寄存器： 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
1	EGN1	INTP1 外部中断下降沿允许寄存器： 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
0	EGN0	INTP0 外部中断下降沿允许寄存器： 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0

EGPn	EGNn	INTPn引脚的有效边沿选择 (n=0~3)
0	0	禁止检测边沿。
0	1	下降沿
1	0	上升沿
1	1	上升和下降的双边沿

对应EGPn位和EGNn位的端口如表24-1所示。

表24-1 对应EGPn位和EGNn位的中断请求信号

检测允许位		中断请求信号
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3

注1：如果将外部中断功能使用的输入端口切换到输出模式，就可能检测到有效边沿而产生INTPn中断。当切换到输出模式时，必须在禁止检测边沿后（EGPn、EGNn=0、0）将端口模式寄存器（PMxx）置“0”。

注2：有关边沿检测的端口，请参照“2.1端口功能”。



## 第25章 待机功能

### 25.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下2种模式。

#### (1)睡眠模式

睡眠模式是停止 CPU 运行时钟的模式。在设定睡眠模式前，如果高速内部振荡器或者低速内部振荡器正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到深度睡眠模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

#### (2)深度睡眠模式

深度睡眠模式是停止高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小 CPU 的工作电流。

因为深度睡眠模式能通过中断请求来解除，所以也能进行间歇运行。但是，因为在解除深度睡眠模式时需要确保振荡稳定的等待时间，所以如果需要通过中断请求立即开始处理，就必须选择睡眠模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设定为待机模式前的内容，并且还保持输入/输出端口的输出锁存器和输出缓冲器的状态。

注1：在转移到深度睡眠模式时，必须在停止以系统时钟运行的外围硬件后执行WFI指令。

注2：为了减小 A/D 转换器的工作电流，必须将 A/D 转换器控制寄存器（CON）bit4（ADCEN）和控制寄存器（CON2）bit7（ADCST 转换完成硬件清 0）清“0”后，再停止 A/D 转换运行后执行 WFI 指令。

注3：能通过选项字节选择在睡眠模式或者深度睡眠模式中是继续还是停止低速内部振荡器的振荡。详细内容请参照“第31章 选项字节”。

## 25.2 睡眠模式

### 25.2.1 睡眠模式的设定

在 SCR 寄存器的 SLEEPDEEP 位为 0 时，执行 WFI 指令，就进入了睡眠模式。在睡眠模式，CPU 停止动作，但是内部寄存器的值仍被保持，周边模块也保持进入睡眠模式之前的状态。周边模块，振荡器等在睡眠模式下的状态见表 25-1。

无论设定前的 CPU 时钟是高速内部振荡器时钟或者低速内部振荡器时钟，都能设定睡眠模式。

注意：中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除睡眠模式。因此，即使在此情况下执行 WFI 指令，也不转移到睡眠模式。

表25-1 睡眠模式中的运行状态

项目		睡眠模式
系统时钟	$f_{IH}$	继续运行
	$f_{IL}$	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态, 保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM0		停止运行
SRAM1		停止运行
端口 (锁存器)		保持休眠之前的设定
通用定时单元		能运行
LSI_定时器		能运行
时钟输出		能运行
看门狗定时器		休眠前设定了继续计数则休眠后可以继续计数
除法与开方运算单元		停止运算
除法运算单元		停止运算
定时器 TIMER0/1		能运行
捕捉/比较/脉宽调制模块(CCP0/1)		能运行
HALL 信号处理		能运行
增强型 PWM(EPWM)		能运行
通用异步收发器(UART)		能运行
串行接口(IICA)		能运行
串行外围接口控制器(SSP/SPI)		能运行
数据传送控制器(DMA)		能运行
快速模数转换(ADC)		能运行
可编程增益放大器(PGA0/1/2)		能运行
模拟比较器(ACMP0/1)		能运行
DAC		能运行
ADCLDO		能运行
上电复位功能		能运行
电压检测功能		能运行
外部中断		能运行
CRC 运算	高速 CRC	能运行
	通用 CRC	停止运行
SFR 保护功能		停止运行

注1: 停止运行: 在转移到睡眠模式时停止运行。

注2:  $f_{IH}$ : 高速内部振荡器时钟。

注3:  $f_{IL}$ : 低速内部振荡器时钟。

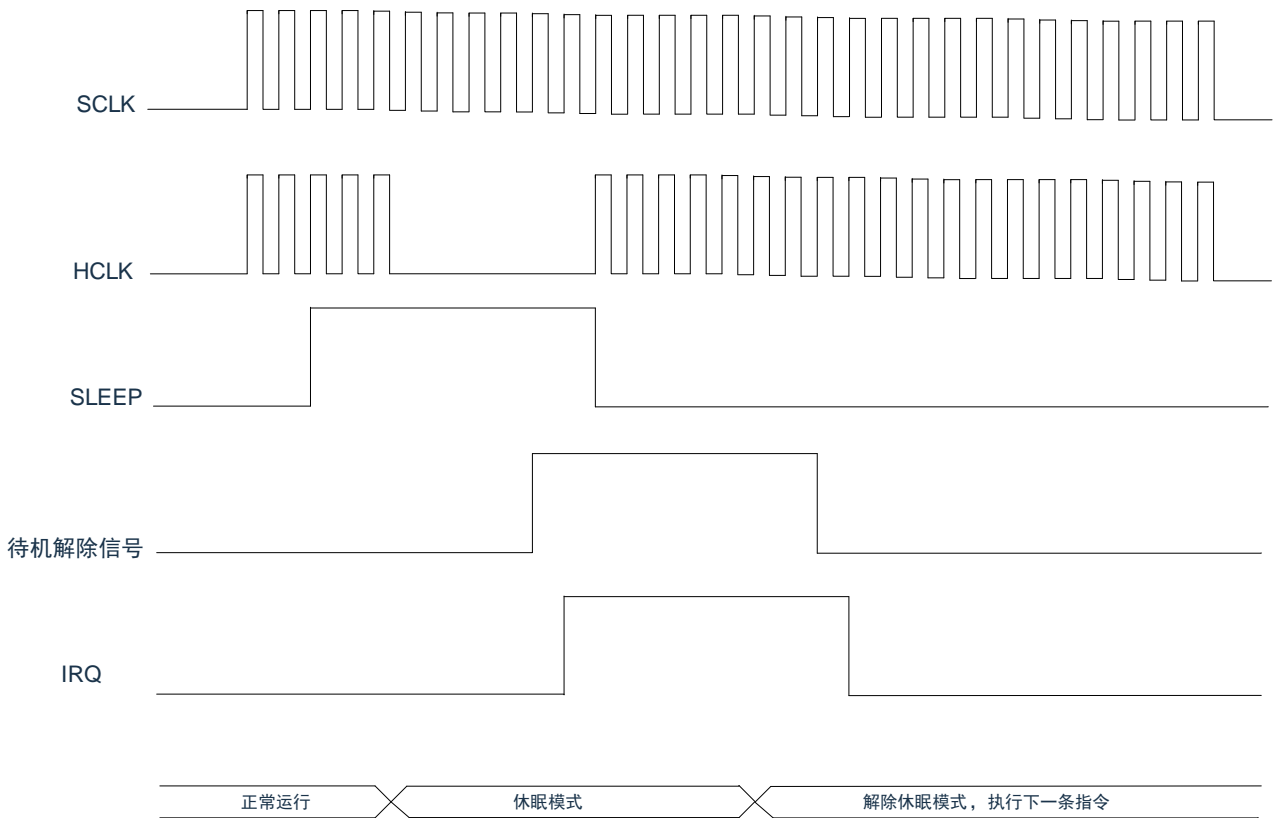
## 25.2.2 睡眠模式的解除

睡眠模式可以被任意中断以及外部复位，POR 复位，低电压检测复位以及 WDT 复位解除。

### (1) 通过中断解除

当产生一个中断请求，且处于允许接受中断的状态时，睡眠模式就被解除，CPU 开始处理中断服务程序。

图25-1 通过中断请求解除睡眠模式



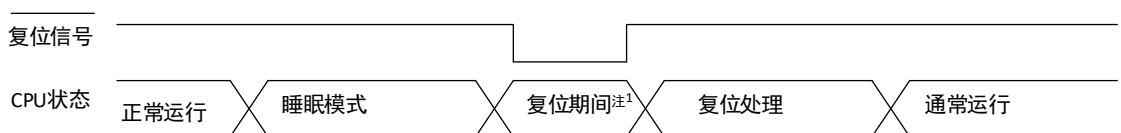
注1：从待机解除信号产生到睡眠模式解除，开始执行中断服务程序，需要16个时钟。

注2：部分待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

### (2) 通过复位解除

当有复位信号产生时，CPU 处于复位状态，睡眠模式被解除。和通常的复位相同，在转移到复位向量地址后执行程序。

图25-2 通过复位解除睡眠模式



注1：有关复位处理，请参照“第26章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理，请参照“第27章 上电复位电路”。

## 25.3 深度睡眠模式

### 25.3.1 深度睡眠模式的设定

在 SCR 寄存器的 SLEEPDEEP 位为 1 时，执行 WFI 指令，就进入了深度睡眠模式。在这个模式，CPU，大多数的周边模块，以及振荡器都停止运行。但是，CPU 内部寄存器的值，RAM 数据，周边模块，I/O 的状态被保持。周边模块，振荡器在深度睡眠模式的运行状态见表 25-2。

注意当中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除深度睡眠模式。因此，如果在此情况下执行WFI指令，就在一旦进入深度睡眠模式后立即被解除。在执行WFI指令并且经过深度睡眠模式解除时间后返回到运行模式。

表25-2 深度睡眠模式中的运行状态

项目		深度睡眠模式
系统时钟	$f_{IH}$	停止运行
	$f_{IL}$	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态，保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM0		停止运行
SRAM1		停止运行
端口（锁存器）		保持休眠之前的设定
通用定时单元 TIMER4		禁止运行
LSI_定时器		能运行
时钟输出		禁止运行
看门狗定时器		能运行
除法与开方运算单元		停止运算
除法运算单元		停止运算
定时器 TIMER0/1		禁止运行
捕捉/比较/脉宽调制模块(CCP0/1)		禁止运行
HALL 信号处理		禁止运行
增强型 PWM(EPWM)		禁止运行
通用异步收发器(UART)		禁止运行
串行接口(IICA)		能通过地址匹配进行唤醒
串行外围接口控制器(SSP/SPI)		禁止运行
数据传送控制器(DMA)		能接受 DMA 启动源
快速模数转换(ADC)		禁止运行
可编程增益放大器(PGA0/1/2)		禁止运行
模拟比较器(ACMP0/1)		禁止运行
DAC		禁止运行
ADCLDO		禁止运行
上电复位功能		能运行
电压检测功能		能运行

外部中断		能运行
CRC 运算	高速 CRC	停止运行
	通用 CRC	停止运行
SFR 保护功能		停止运行

注1：停止运行：在转移到深度睡眠模式时停止运行。

禁止运行：在转移到深度睡眠模式前停止运行。

注2： $f_{IH}$ ：高速内部振荡器时钟。

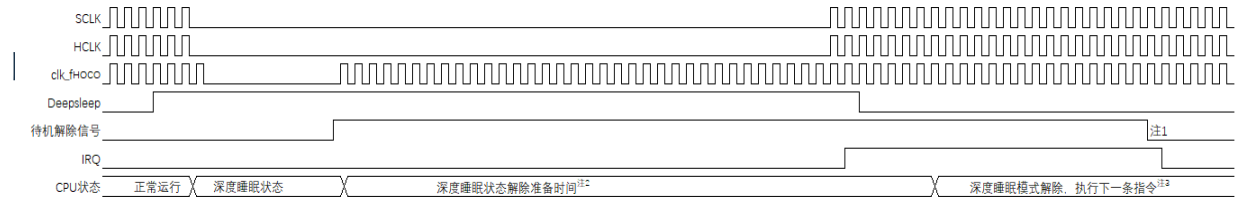
注3： $f_{IL}$ ：低速内部振荡器时钟。

### 25.3.2 深度睡眠模式的解除

能通过以上2种方法解除深度睡眠模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生 LVD 检测、INTP0-3、LSI 定时器或者 WDT 中断请求，就解除深度睡眠模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。



注1：待机解除信号：有关待机解除信号的详细内容，请参照中断相关内容。

注2：深度睡眠状态解除准备时间：进入深度睡眠模式前CPU时钟为高速内部振荡时时：至少20us。

注3：等待：从CPU.IRQ有效到开始执行中断服务程序，需要14个时钟。

备注：高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

(b) 通过产生复位信号进行的解除

通过产生复位信号来解除深度睡眠模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图25-3 通过复位解除深度睡眠模式



注1：有关复位处理，请参照“第26章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理，请参照“第27章 上电复位电路”。

## 25.4 部分掉电的深度睡眠模式

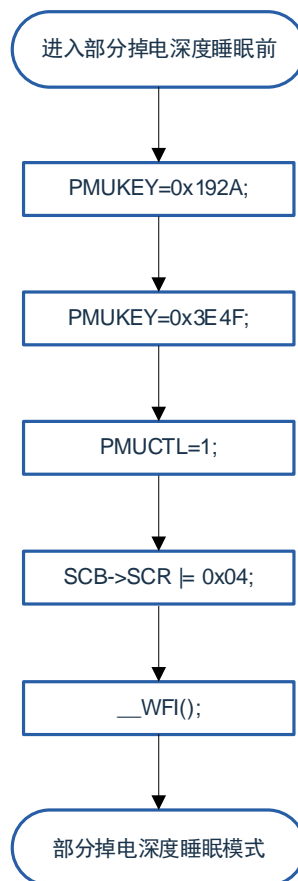
### 25.4.1 部分掉电的深度睡眠模式的设定

部分掉电的深度睡眠模式是在深度睡眠模式的基础上关闭部分周边的电源供电而进一步节省电源消耗的深度睡眠模式。进入部分掉电的深度睡眠模式需要配置 PMUCTL 寄存器的 PWDNEN 位，该控制位的写入受供电模式控制保护寄存器（PMUKEY）的保护，解除部分掉电的深度睡眠模式时要求重新初始化掉电的周边后才能重新正常运行，需要重新初始化的周边模块详情请参考表 25-3 部分掉电的深度睡眠模式中的运行状态。

在 SCR 寄存器的 SLEEPDEEP 位为 1 并且 PMUCTL 寄存器的 PWDNEN 位也为 1 时，执行 WFI 指令就可进入部分掉电的深度睡眠模式。在这个模式，CPU，以及振荡器都停止运行，大多数的周边模块会被关闭电源供给。但是，CPU 内部寄存器的值，RAM0 数据，I/O 的状态会被保持。周边模块，振荡器在部分掉电的深度睡眠模式的运行状态见表 25-3。

PMUCTL 寄存器的 PWDNEN 位的控制参考 4.4.8 供电模式控制保护寄存器（PMUKEY）和 4.4.9 供电模式控制寄存器（PMUCTL）部分。

图 25-4 进入部分掉电的深度睡眠模式流程图



注意：当中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除深度睡眠模式。因此，如果在此情况下执行 WFI 指令，就在一旦进入深度睡眠模式后立即被解除，此种情况下不会部分掉电模式。在执行 WFI 指令并且经过深度睡眠模式解除时间后返回到运行模式。



表25-3 部分掉电的深度睡眠模式中的运行状态

项目		部分掉电深度睡眠模式
系统时钟	$f_{IH}$	停止运行
	$f_{IL}$	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态, 保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM0		停止运行
SRAM1		停止运行
端口 (锁存器)		保持休眠之前的设定
通用定时单元 TIMER4		禁止运行
LSI_定时器		能运行
时钟输出		停止运行
看门狗定时器		能运行
除法与开方运算单元		停止运算
除法运算单元		停止运算
定时器 TIMER0/1		禁止运行
捕捉/比较/脉宽调制模块(CCP0/1)		禁止运行
HALL 信号处理		禁止运行
增强型 PWM(EPWM)		禁止运行
通用异步收发器(UART)		禁止运行
串行接口(IICA)		禁止运行
串行外围接口控制器(SSP/SPI)		禁止运行
数据传送控制器(DMA)		禁止运行
快速模数转换(ADC)		停止运行
可编程增益放大器(PGA0/1/2)		禁止运行
模拟比较器(ACMP0/1)		禁止运行
DAC		禁止运行
ADCLDO		禁止运行
上电复位功能		能运行
电压检测功能		能运行
外部中断		能运行
CRC 运算	高速 CRC	停止运行
	通用 CRC	停止运行
SFR 保护功能		停止运行

注1: 停止运行: 在转移到部分掉电的深度睡眠模式时自动停止运行。

禁止运行: 在转移到部分掉电的深度睡眠模式前停止运行。

转移到部分掉电的深度睡眠模式后停止模块的电源供应, 解除模式后需要重新初始化模块。

注2:  $f_{IH}$ : 高速内部振荡器时钟。

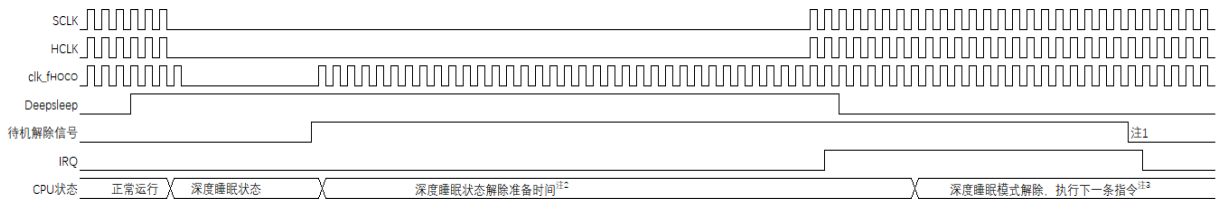
注3:  $f_{IL}$ : 低速内部振荡器时钟。

## 25.4.2 部分掉电的深度睡眠模式的解除

能通过以上 2 种方法解除部分掉电的深度睡眠模式。

### (a) 通过中断请求进行部分掉电的深度睡眠的解除

如果 INTP0-3、LSITIMER 定时器中断、LVI 中断及 WDT 中断请求，可解除部分掉电的深度睡眠模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。



注 1：待机解除信号：INTP0-3、LSITIMER 定时器中断、LVI 中断及 WDT 中断请求信号。

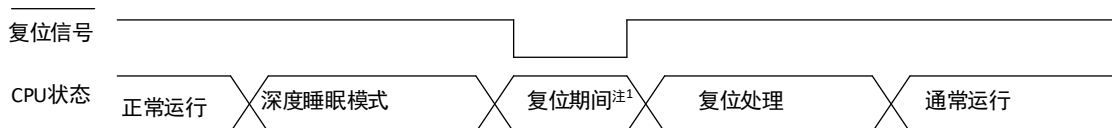
注 2：部分掉电的深度睡眠状态解除准备：

需要重新初始化周边功能及 RAM1 等功能，才能保证程序继续正常运行。

### (b) 通过产生复位信号进行的解除

通过产生复位信号来解除部分掉电的深度睡眠模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图25-5 通过复位解除部分掉电的深度睡眠模式



注1：有关复位处理，请参照“第26章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理，请参照“第27章 上电复位电路”。

## 第26章 复位功能

以下6种方法产生复位信号。

- (1)通过 RESETB 引脚输入外部复位。
- (2)通过看门狗定时器的程序失控检测产生内部复位。
- (3)通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- (4)通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- (5)因系统复位请求寄存器位(AIRC.R.SYSRESETREQ)被置为 1 而产生内部复位。
- (6)因存取非法存储器而产生内部复位。

内部复位和外部复位相同，在产生复位信号后，从用户自定义的程序起始地址开始执行程序。

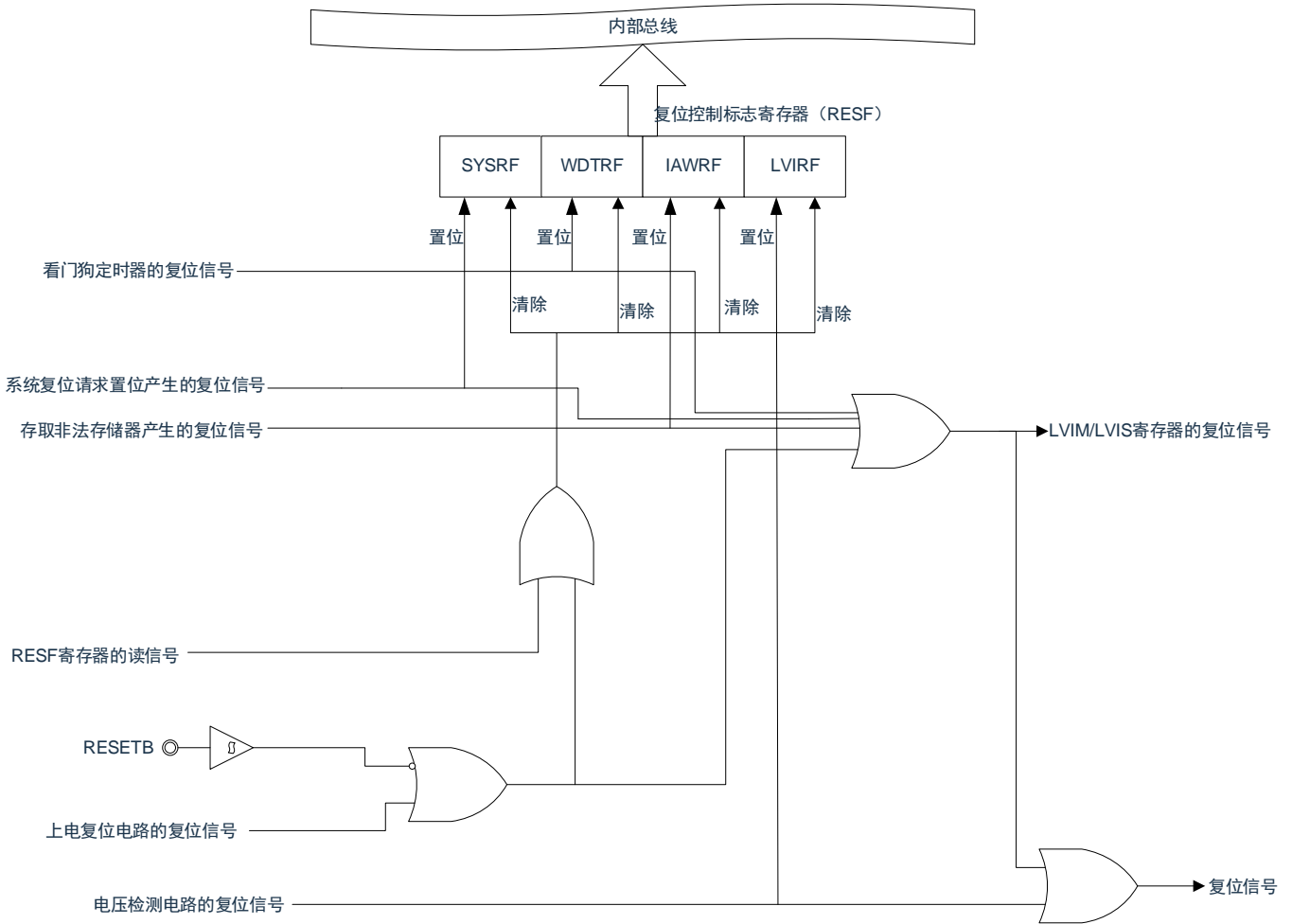
当给 RESETB 引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者系统复位请求位被置位，或者存取非法存储器时，产生复位并且各硬件变为如表 26-1 所示的状态。

注 1：在进行外部复位时，必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位，就必须在给 RESETB 引脚输入低电平后接通电源，而且在用户手册的 AC 特性所示的工作电压范围内至少保持 10us 的低电平，然后输入高电平。

注 2：如果发生复位，就对各 SFR 进行初始化，因此端口引脚变为以下状态：

- 在外部复位或者 POR 复位的期间除 P01, P02, P03 接内部上拉，其他 IO 均处于高阻态。

图26-1 复位功能的框图



注1: LVD电路的内部复位不会对LVD电路进行复位。

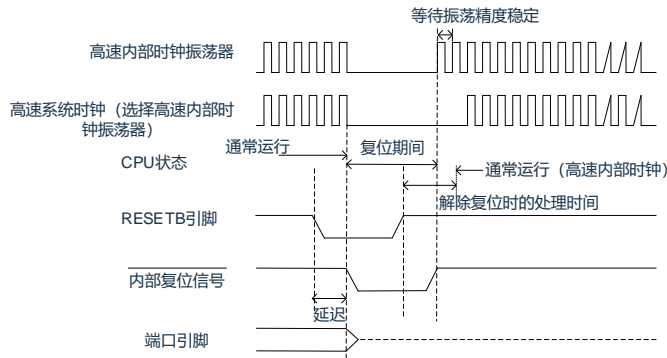
注2: LVIM: 电压检测寄存器。

注3: LVIS: 电压检测电平寄存器。

## 26.1 复位时序

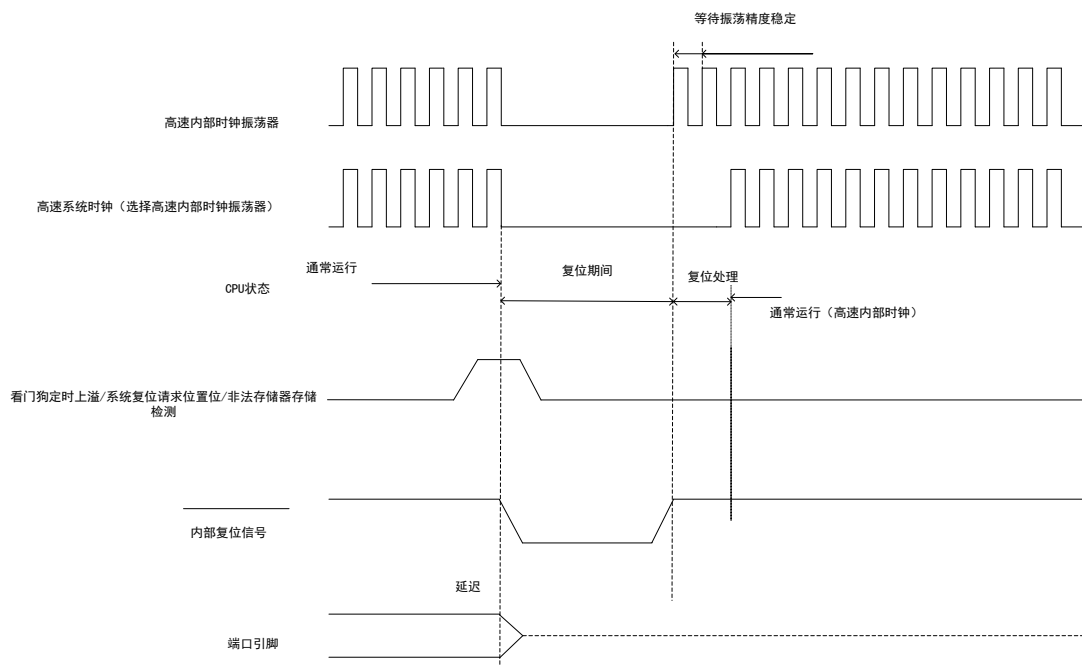
当给 RESETB 引脚输入低电平时，产生复位。然后，如果给 RESETB 引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 26-2 RESETB 输入的复位时序



对于因看门狗定时器的上溢、系统复位请求位的置位或者非法存储器存取的检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 26-3 因看门狗定时器的上溢、系统复位请求位的置位或者非法存储器存取的检测而产生的复位时序。



注1：端口引脚P01，P02，P03变为以下状态：

- 在外部复位期间P01为外部输入信号确定，P02，P03为高电平。
- POR复位的期间为P01，P02，P03为高电平。

注2：看门狗定时器也不例外，在发生内部复位时进行复位。

对于由 POR 电路和 LVD 电路的电压检测产生的复位，如果在复位后满足  $V_{DD} \geq V_{POR}$  或者  $V_{DD} \geq V_{LVD}$ ，就解除复位状态，并且在复位处理后以高速内部振荡器时钟开始执行程序。详细内容请参照“第 27 章 上电复位电路”和“第 28 章 电压检测电路”。

备注： $V_{POR}$ ：POR 电源电压上升检测电压•在外部复位期间 P01 为外部输入信号确定，P02，P03 为高电平。

$V_{LVD}$ ：LVD 检测电压。

表 26-1 复位期间的运行状态

项目		复位期间					
		外部复位 RESINB	看门狗复位	上电复位	LVD 电压检测复位	写复位寄存器复位	存取非法存储器复位
系统时钟	$f_{IH}$	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
	$f_{IL}$	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
CPU		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
代码闪存		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SRAM0		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SRAM1		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
端口（锁存器）		P01,P02,P03 上电后为高，其他脚复位前中后均为高阻态。	P01,P02,P03 上电后为高，其他脚复位前中后均为高阻态。	P01,P02,P03 上电后为高，其他脚复位前中后均为高阻态。	P01,P02,P03 上电后为高，其他脚复位前中后均为高阻态。	P01,P02,P03 上电后为高，其他脚复位前中后均为高阻态。	P01,P02,P03 上电后为高，其他脚复位前中后均为高阻态。
通用定时单元 TIMER4		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
LSI_定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
时钟输出		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
看门狗定时器		计数停止	停止运行	停止运行	停止运行	停止运行	停止运行
除法与开方运算单元		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
除法运算单元		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
定时器 TIMER0/1		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
捕捉/比较/脉宽调制模块 (CCP0/1)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
HALL 信号处理		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
增强型 PWM(EPWM)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
通用异步收发器(UART)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
I2C 串行接口控制器(IICA)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
串行外围接口控制器 (SSP/SPI)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
快速模数转换(ADC)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
可编程增益放大器 (PGA0/1/2/3)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
模拟比较器(ACMP0/1)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
DAC		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
ADCLDO		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
上电复位功能		能进行检测运行	能进行检测运行	能进行检测运行	能进行检测运行	能进行检测运行	能进行检测运行
电压检测功能		不能运行			能运行	不能运行	不能运行
外部中断		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
数据传送控制器(DMA)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
CRC 运算功能	高速 CRC	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
	通用 CRC	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SFR 保护功能		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
复位完成后上电状态(选项字节配置)		重新配置	重新配置	重新配置	重新配置	重新配置	重新配置

注1: P02上电后为高（外部复位时为外部输入的状态），P02,P03为高，其他脚复位前中后均为高阻态。

 注2:  $f_{IH}$ : 高速内部振荡器时钟;  $f_{IL}$ : 低速内部振荡器时钟。

## 26.2 确认复位源的寄存器

### 26.2.1 寄存器映射

(复位控制基地址 = 0x4002\_0440) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
RESF	0x000	RO	复位控制标志寄存器	-

### 26.2.2 复位控制标志寄存器 (RESF)

该微控制器存在多种内部复位发生源。复位控制标志寄存器 (RESF) 保存发生复位请求的复位源。能通过 8 位存储器操作指令读 RESF 寄存器。

通过 RESETB 的输入、上电复位 (POR) 电路的复位和 RESF 寄存器的读取, 清除 SYSRF、WDTRF、IAWRF、LVIRF 标志。要判断复位源时, 必须将 RESF 寄存器的值保存到任意的 RAM, 然后通过其 RAM 值进行判断。

位	符号	描述	复位值
7	SYSRF	系统复位请求位被置位而产生的内部复位请求 0: 没有产生内部复位请求或清除了 RESF 寄存器。 1: 产生内部复位请求。	-
6:5	-	保留	-
4	WDTRF	看门狗定时器 (WDT) 产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-
3:2	-	保留	-
1	IAWRF	存取非法存储器产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-
0	LVIRF	电压检测电路 (LVD) 产生的内部复位请求 0: 没有产生内部复位请求或者清除了 RESF 寄存器 1: 产生内部复位请求。	-

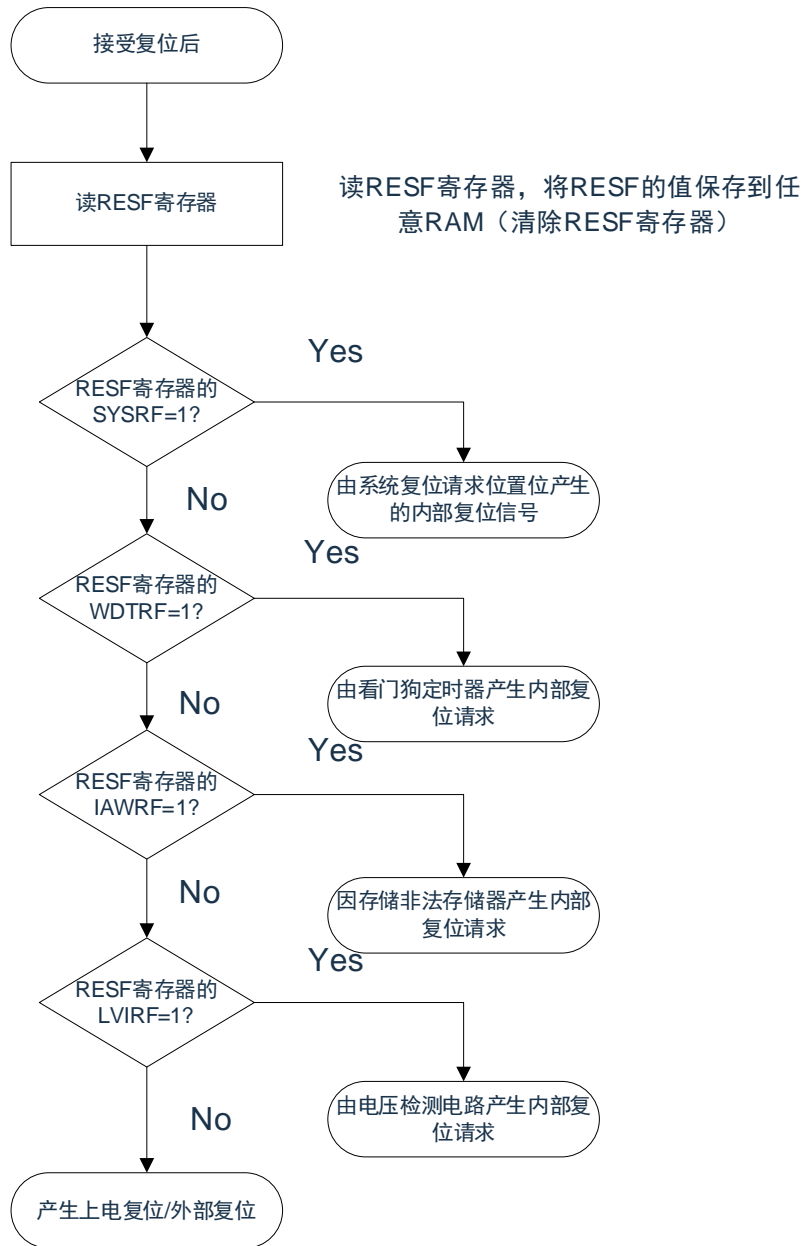
发生复位请求时的RESF寄存器状态如下表所示。

标志 \ 复位源	RESETB 输入	POR产生的复位	系统复位请求位 置位产生的复位	WDT产生的复位	存取非法存储器 产生的复位	LVD产生的复位
SYSRF	清“0”	清“0”	置“1”	保持	保持	保持
WDTRF			保持	置“1”		
IAWRF				保持	置“1”	
LVIRF			保持	置“1”		



复位源的确认步骤如图26-4所示。

图26-4 复位源的确认步骤



注意：上述流程是确认步骤的例子。

## 第27章 上电复位电路

### 27.1 上电复位电路的功能

上电复位电路（POR）有以下功能。

- 在接通电源时产生内部复位信号。

如果电源电压（ $V_{DD}$ ）超过检测电压（ $V_{POR}$ ），就解除复位。但是，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态。

- 将电源电压（ $V_{DD}$ ）和检测电压（ $V_{PDR}$ ）进行比较。当  $V_{DD} < V_{PDR}$  时，产生内部复位信号。但是，当电源电压下降时，必须在电源电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注 1：当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清“00H”。

注 2：芯片内置多个产生内部复位信号的硬件。当由看门狗定时器（WDT）、电压检测（LVD）电路、系统复位请求位置位或者非法存储器的存取而产生内部复位信号时，用于表示复位源的标志分配在 RESF 寄存器；当由 WDT、LVD、系统复位请求位的置位或者非法存储器的存取而产生内部复位信号时，不将 RESF 寄存器清“00H”，而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 26 章 复位功能”。

注 3： $V_{POR}$ ：POR 电源电压上升检测电压

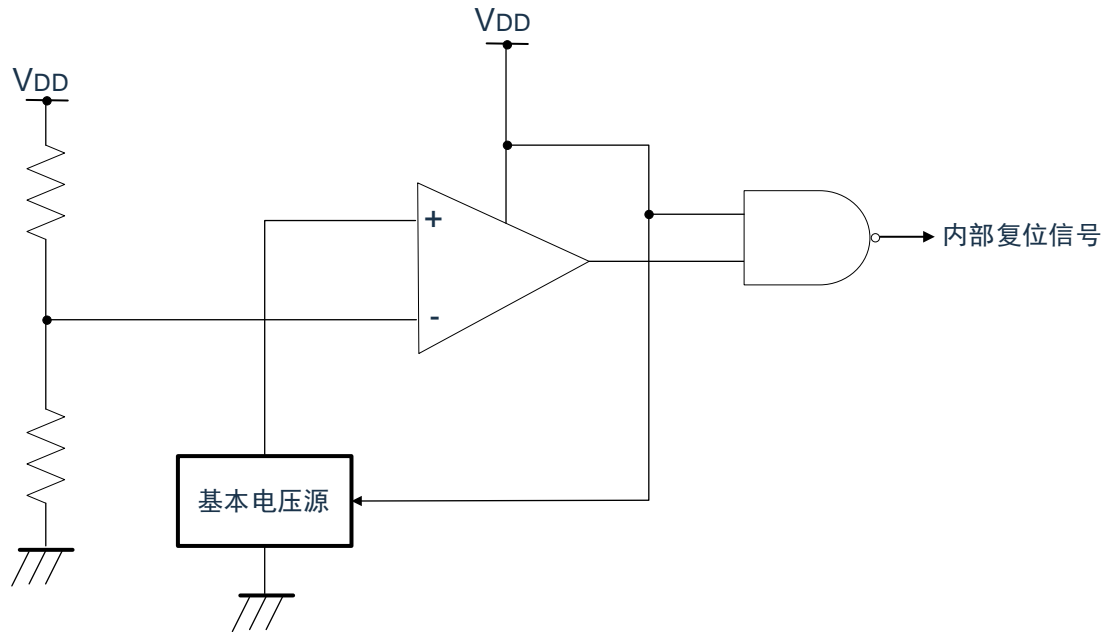
注 4： $V_{PDR}$ ：POR 电源电压下降检测电压

注 5：有关  $V_{POR}$ 、 $V_{PDR}$  详细内容请参照数据手册的 POR 电路特性。

## 27.2 上电复位电路的结构

上电复位电路的框图如图27-1所示。

图27-1 上电复位电路的框图

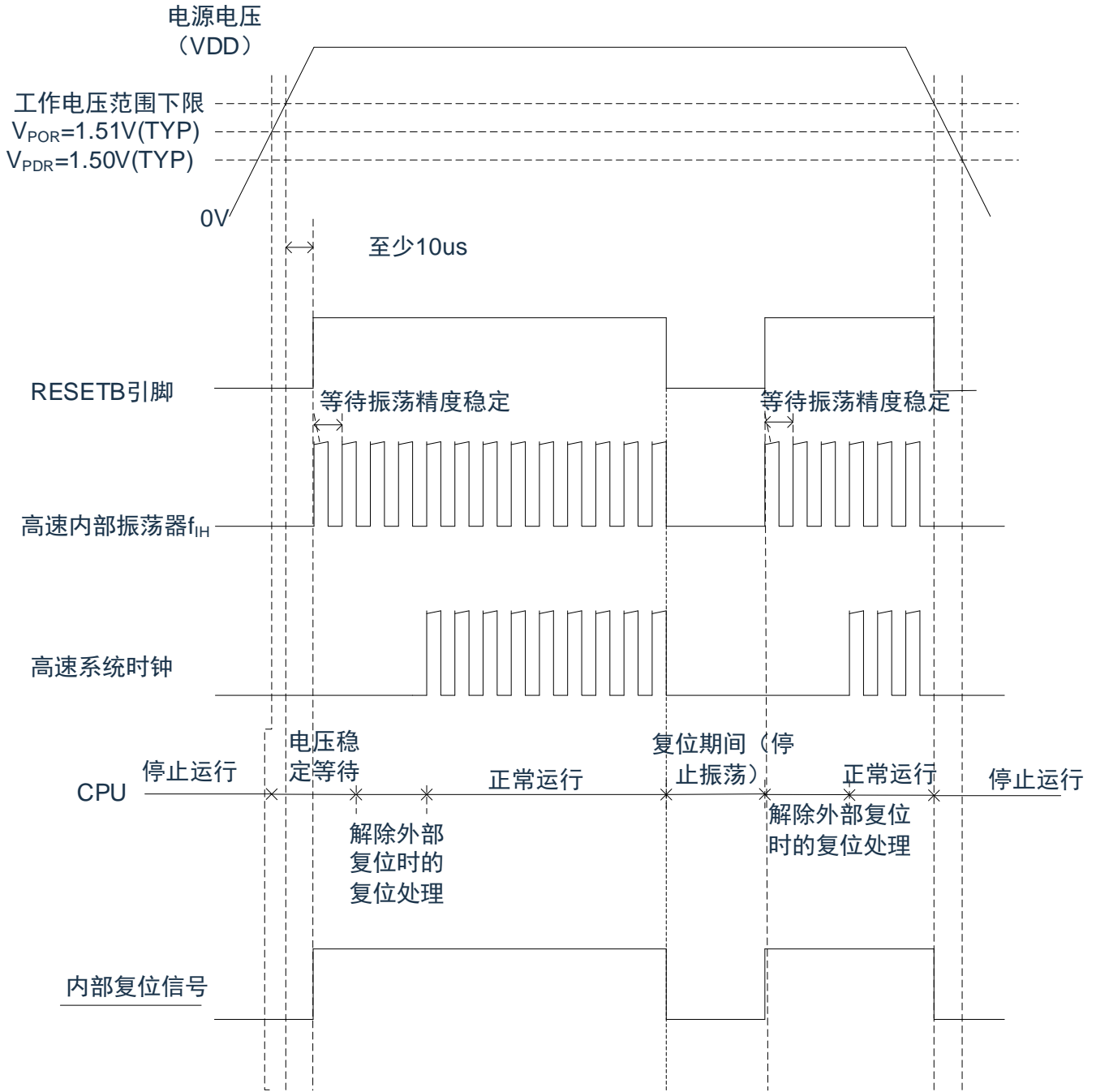


## 27.3 上电复位电路的运行

上电复位电路和电压检测电路的内部复位信号的产生时序如下所示。

图 27-2 上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

(1) 使用 RESETB 引脚的外部复位输入的情况



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

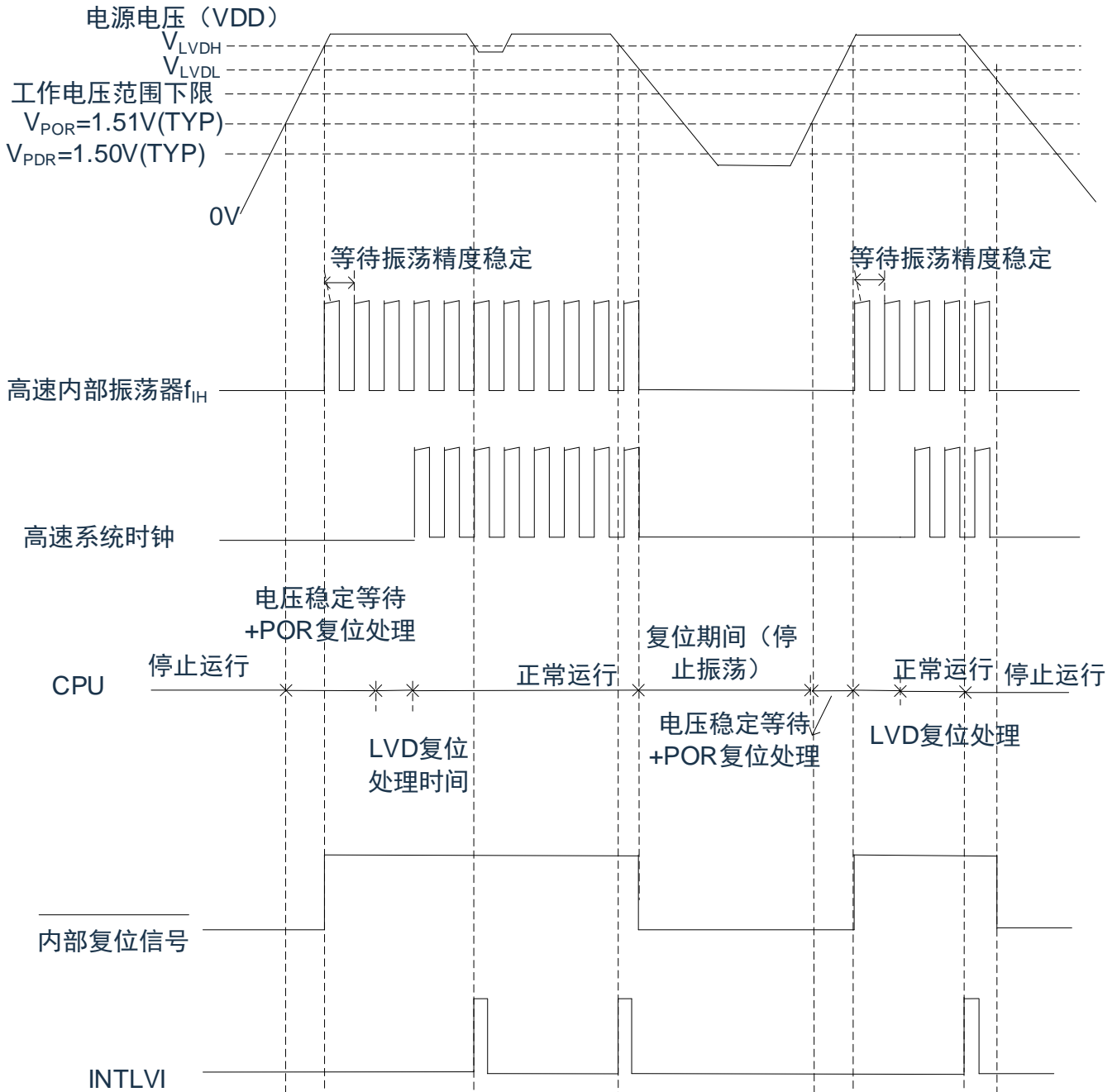
注3： $V_{POR}$ ：POR电源电压上升检测电压。

注4:  $V_{PDR}$ : POR电源电压下降检测电压。

注5: 在LVD为OFF时, 必须使用RESETB引脚的外部复位。详细内容请参照“第28章 电压检测电路”。

图27-2 上电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

(2) LVD为中断&复位模式的情况 (选项字节000C1H的LVIMDS1、LVIMDS0=1、0)



注1: 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

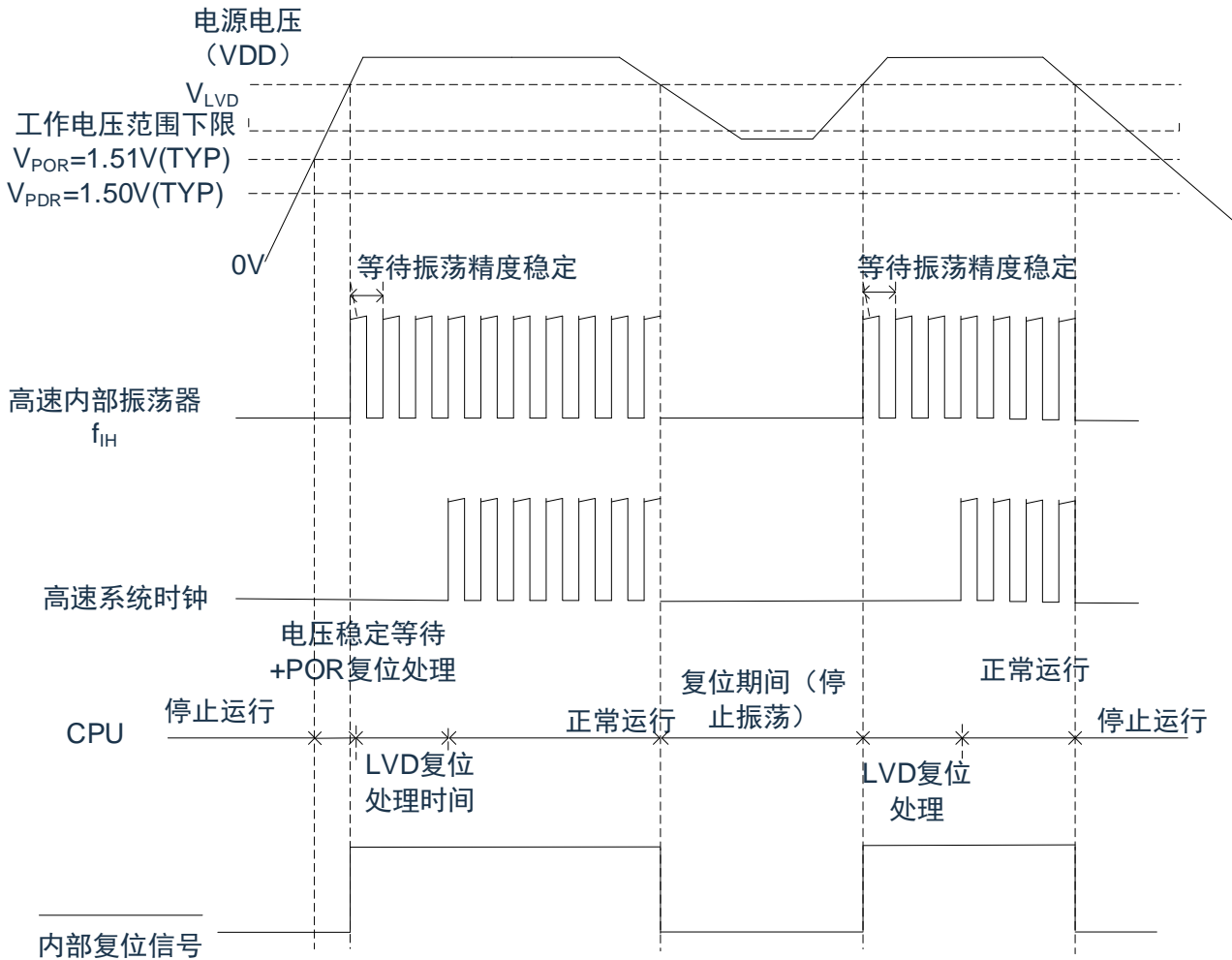
注2: 在产生中断请求信号 (INTLVI) 后, 自动将电压检测电平寄存器 (LVIS) 的LVILV位和LVIMD位置“1”。因此, 必须考虑可能出现电源电压在不低于低电压检测电压 ( $V_{LVDL}$ ) 的状态下恢复到高电压检测电压 ( $V_{LVDH}$ ) 或者更高的情况, 在产生INTLVI后按照“图28-6 工作电压的确认/复位的设定步骤”和“图28-7中断&复位模式的初始设定步骤”进行设定。

注3: 到开始通常运行为止的时间除了达到 $V_{POR}$  (1.51V(TYP.)) 后的“电压稳定等待+POR复位处理”以外, 在达到LVD检测电平 ( $V_{LVDH}$ ) 后还需要以下的“LVD复位处理”。

- 注4:  $V_{LVDH}$ 、 $V_{LVDL}$  : LVD检测电压。
- 注5:  $V_{POR}$  : POR电源电压上升检测电压。
- 注6:  $V_{PDR}$  : POR电源电压下降检测电压。

图27-2 上电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

(3)LVD复位模式的情况（选项字节000C1H的LVIMDS1、LVIMDS0=1、1）



注1: 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2: 到开始通常运行为止的时间除了达到 $V_{POR}$  (1.51V(TYP.))后的“电压稳定等待+POR复位处理”以外, 在达到LVD检测电平 ( $V_{LVD}$ ) 后还需要“LVD复位处理”。

注3: 在电源电压下降时, 如果只在发生电压检测电路 (LVD) 的内部复位后恢复电源电压, 就在达到LVD检测电平 ( $V_{LVD}$ ) 后需要“LVD复位处理”。

注4:  $V_{LVDH}$ 、 $V_{LVDL}$ :LVD检测电压。

注5:  $V_{POR}$ :POR电源上升检测电压。

注6:  $V_{PDR}$ :POR电源下降检测电压。

注7: 当选择LVD中断模式（选项字节000C1H的LVIMD1、LVIMD0=0、1）时, 从接通电源到开始正常运行的时间和“图27-2(3/3)LVD复位模式的情况”的LVD复位处理的时间相同。

# 第28章 电压检测电路

## 28.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设定运行模式和检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ ）。电压检测（LVD）电路有以下功能。

- 将电源电压（ $V_{DD}$ ）和检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ ）进行比较，产生内部复位或者内部中断信号。
- 电源电压的检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ ）能通过选项字节选择 12 种检测电平（参照“第 31 章 选项字节”）。
- 也能在深度睡眠模式中运行
- 当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节

(a) 中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）

通过选项字节 000C1H 选择 2 个检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ ），高电压检测电平（ $V_{LVDH}$ ）用于解除复位或者产生中断，低电压检测电平（ $V_{LVDL}$ ）用于产生复位。

(b) 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）

将选项字节 000C1H 选择的 1 个检测电压（ $V_{LVD}$ ）用于产生或者解除复位。

(c) 中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）

将选项字节 000C1H 选择的 1 个检测电压（ $V_{LVD}$ ）用于产生中断或者解除复位。在各模式中，产生以下的中断信号和内部复位信号。

模式	中断&复位模式	复位模式	中断模式
设置	(LVIMDS1、LVIMDS0=1、0)	(LVIMDS1、LVIMDS0=1、1)	(LVIMDS1、LVIMDS0=0、1)
运行过程	在工作电压下降时，当检测到 $V_{DD} < V_{LVDH}$ 时，产生中断请求信号；当检测到 $V_{DD} < V_{LVDL}$ 时，产生内部复位；当检测到 $V_{DD} \geq V_{LVDH}$ 时，解除内部复位。	当检测到 $V_{DD} \geq V_{LVD}$ 时，解除内部复位；当检测到 $V_{DD} < V_{LVD}$ 时，产生内部复位。	在发生复位后，LVD的内部复位状态继续保持到 $V_{DD} \geq V_{LVD}$ 为止。当检测到 $V_{DD} \geq V_{LVD}$ 时，解除LVD的内部复位。在解除LVD的内部复位后，如果检测到 $V_{DD} < V_{LVD}$ 或者 $V_{DD} \geq V_{LVD}$ 时，就产生中断请求信号（INTLVI）。

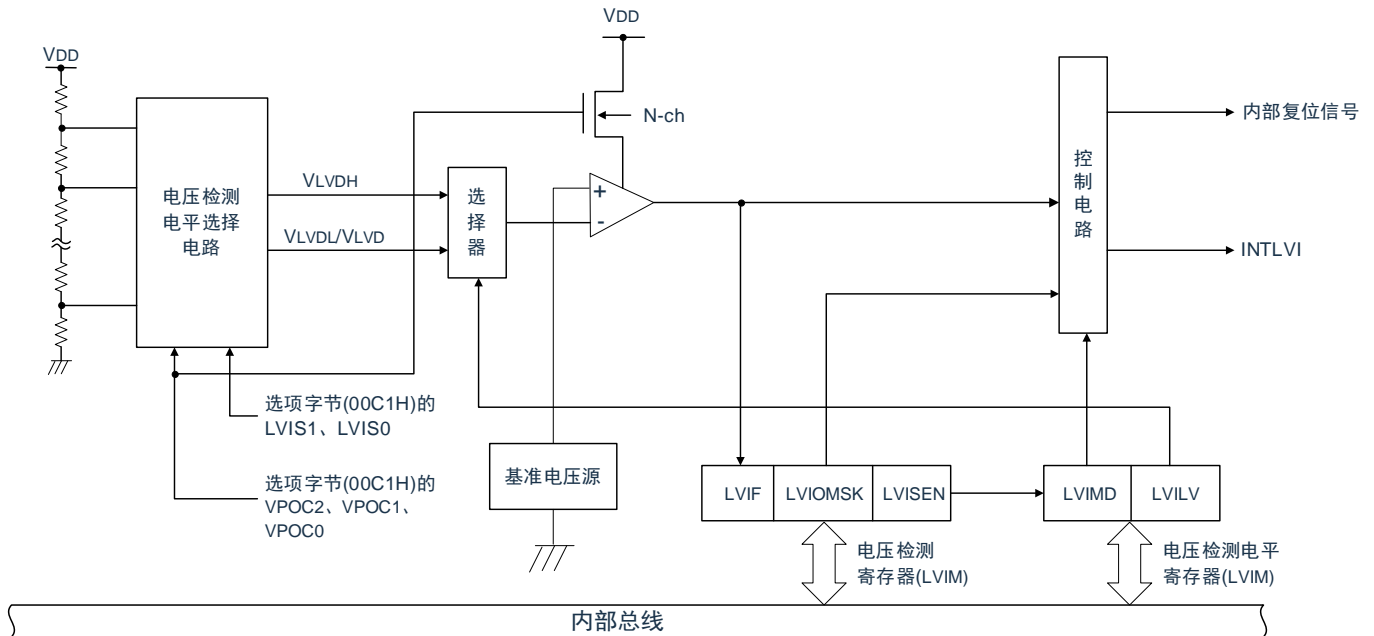
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

如果发生复位，就将复位控制标志寄存器（RESF）的 bit0（LVIRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 26 章 复位功能”。

## 28.2 电压检测电路的结构

电压检测电路的框图如图28-1所示。

图28-1 电压检测电路的框图



## 28.3 寄存器映射

(电压检测基地址 = 0x4002\_0441) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LVIM	0x000	R/W	电压检测寄存器	0x0
LVIS	0x001	R/W	电压检测电平寄存器	0x0

## 28.4 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器 (LVIM)
- 电压检测电平寄存器 (LVIS)



## 28.4.1 电压检测寄存器 (LVIM)

此寄存器设定允许或者禁止改写电压检测电平寄存器 (LVIS)，并且确认LVD输出的屏蔽状态。通过8位存储器操作指令设定LVIM寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	LVISEN <sup>注1</sup>	电压检测电平寄存器 (LVIS) 的允许/禁止改写的设定 0: 禁止 1: 允许	-
6:2	-	保留	-
1	LVIOMSK	LVD 输出的屏蔽状态标志 0: 屏蔽无效 1: 屏蔽有效 <sup>注2</sup>	-
0	LVIF	电压检测标志 0: 电源电压 (V <sub>DD</sub> ) ≥ 检测电压 (V <sub>LVD</sub> ) 或者 LVD 为 OFF。 1: 电源电压 (V <sub>DD</sub> ) < 检测电压 (V <sub>LVD</sub> )	-

注1: 只有在选择中断&复位模式 (选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”) 时才能设定, 在其他模式中不能更改初始值。

注2: 只有在选择中断&复位模式 (选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”) 时, LVIOMSK位才在以下期间自动变为“1”, 屏蔽LVD产生的复位或者中断。

- LVISEN=1的期间
- 从发生LVD中断开始到LVD检测电压稳定为止的等待时间
- 从更改LVILV位的值到LVD检测电压稳定为止的等待时间

## 28.4.2 电压检测电平寄存器 (LVIS)

这是设定电压检测电平的寄存器。

通过 8 位存储器操作指令设定 LVIS 寄存器。在产生复位信号后，此寄存器的值变为“00H/01H/81H”注 1。

位	符号	描述	复位值
7	LVIMD注1	电压检测的运行模式 0: 中断模式 1: 复位模式	0
6:1	-	保留	-
0	LVILV注2	LVD 检测电平 0: 高电压检测电平 ( $V_{LVDH}$ ) 1: 低电压检测电平 ( $V_{LVDL}$ 或者 $V_{LVD}$ )	0

注1: 复位值因复位源和选项字节的设定而变。在发生LVD复位时，不将此寄存器清“00H”。

在发生LVD以外的复位时，此寄存器的值如下：

- 选项字节的LVIMDS1、LVIMDS0=1、0时：00H
- 选项字节的LVIMDS1、LVIMDS0=1、1时：81H
- 选项字节的LVIMDS1、LVIMDS0=0、1时：01H

注2: 只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时才能写“0”。在其他情况下不能设定。在中断&复位模式中，通过产生复位或者中断自动进行值的替换。

注3: 要改写LVIS寄存器时，必须按照图28-和图28-的步骤进行。

注4: 通过选项字节000C1H选择LVD的运行模式和各模式的检测电压 ( $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ )。用户选项字节 (000C1H/010C1H) 的详细内容，请参照“第31章 选项字节”。

## 28.5 电压检测电路的运行

### 28.5.1 用作复位模式时的设定

通过选项字节 000C1H 设定运行模式（复位模式（LVIMDS1、LVIMDS0=1、1）和检测电压（ $V_{LVD}$ ）。如果设定复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“81H”。bit7（LVIMD）为“1”（复位模式）。bit0（LVILV）为“1”（电压检测电平： $V_{LVD}$ ）。

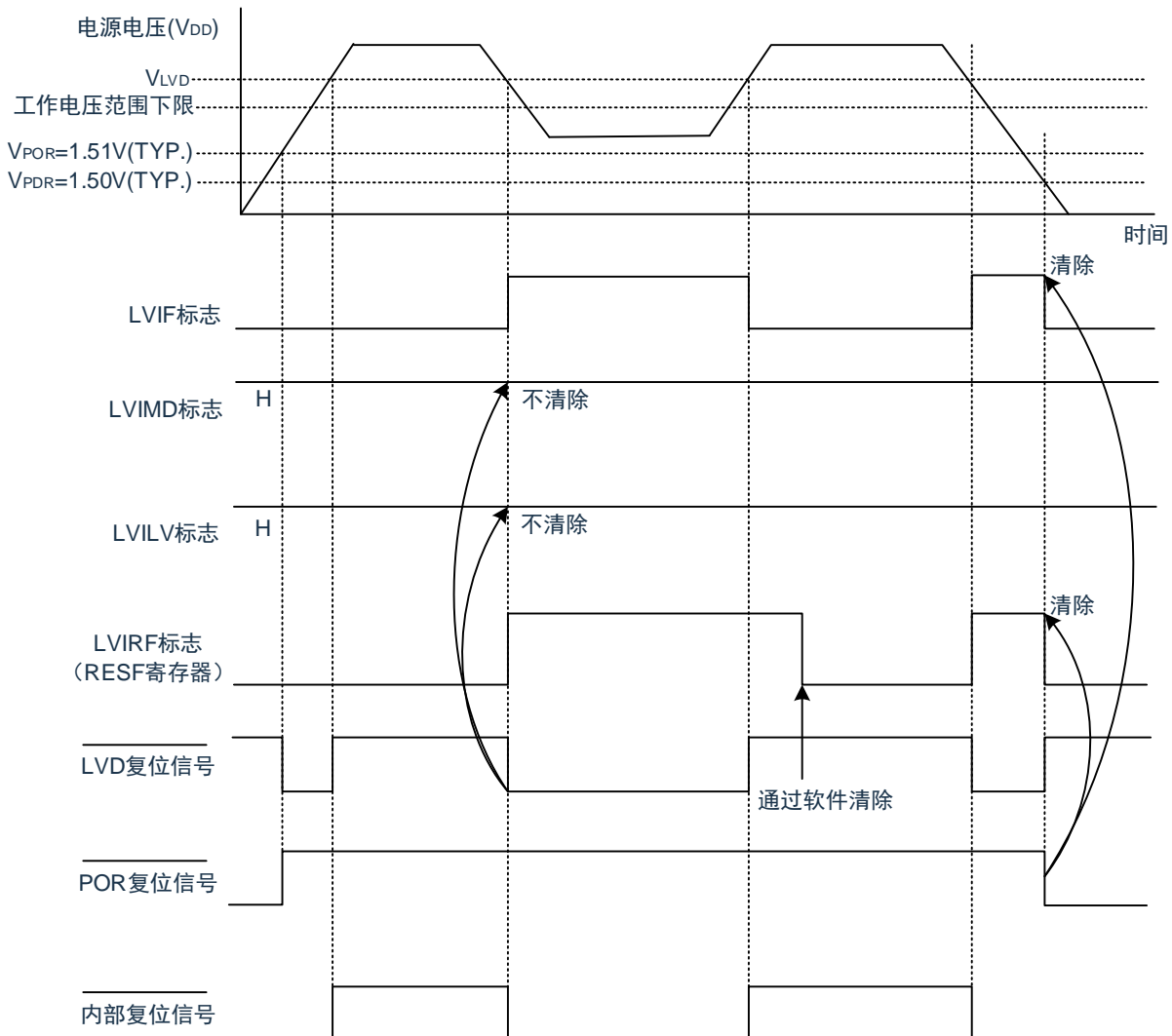
#### •LVD 复位模式的运行

当接通电源时，复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）在电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ）前保持 LVD 的内部复位状态。如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就解除内部复位。

当工作电压下降时，如果电源电压（ $V_{DD}$ ）低于电压检测电平（ $V_{LVD}$ ），就产生 LVD 的内部复位。

LVD 复位模式的内部复位信号的产生时序如图 28-2 所示。

图28-2 内部复位信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、1）



注1:  $V_{POR}$ : POR电源电压上升检测电压

注2:  $V_{PDR}$ : POR电源电压下降检测电压

## 28.6 用作中断模式时的设定

通过选项字节 000C1H 设定运行模式（中断模式（LVIMDS1、LVIMDS0=0、1）和检测电压（ $V_{LVD}$ ）。如果设定中断模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“01H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“1”（电压检测电平： $V_{LVD}$ ）。

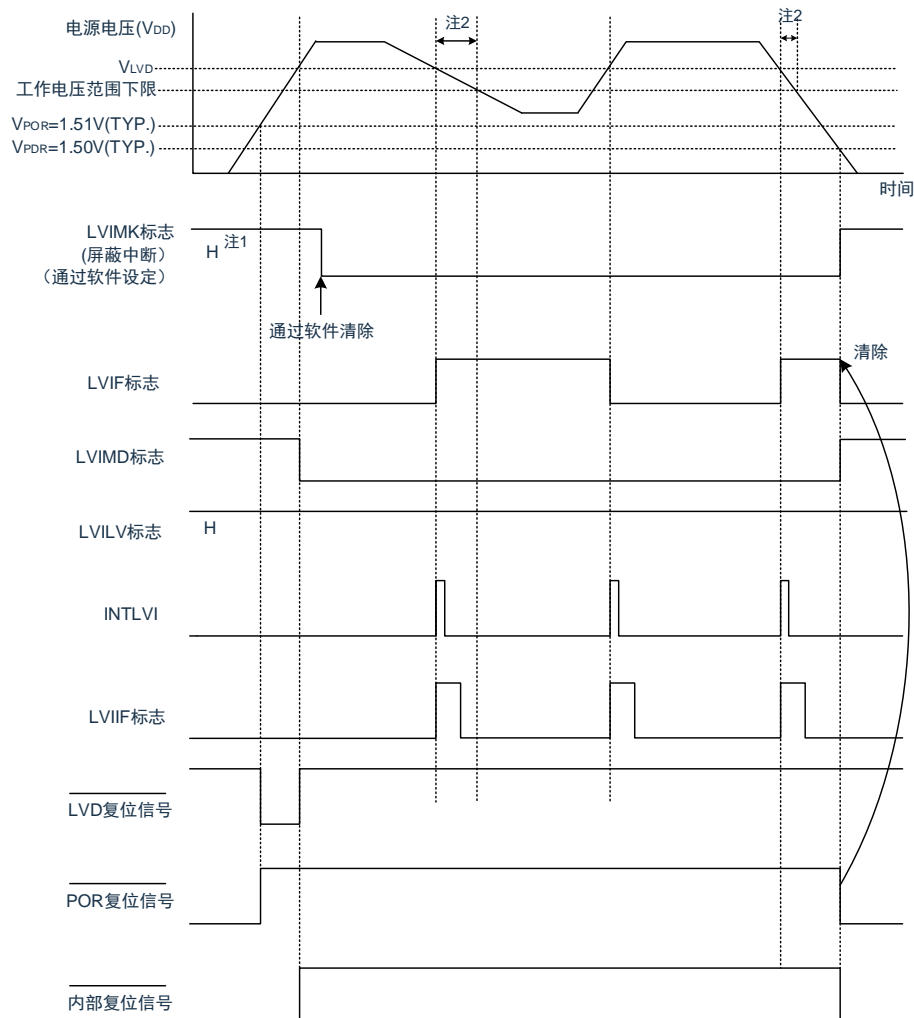
### ●LVD 中断模式的运行

在产生复位后，中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）在电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ）前保持 LVD 的内部复位状态。如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就解除 LVD 的内部复位。

在解除 LVD 的内部复位后，如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就产生 LVD 的中断请求信号（INTLVI）。当工作电压下降时，必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD 中断模式的中断请求信号的产生时序如图 28-3 所示。

图28-3 中断信号的产生时序（选项字节的LVIMDS1、LVIMDS0=0、1）



注1: 在产生复位信号后, LVIMK标志变为“1”。

注2: 当工作电压下降时, 必须在工作电压低于数据手册的AC特性所示的工作电压范围前, 通过深度睡眠模式的转移或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

备注:  $V_{POR}$ : POR电源电压上升检测电压。

$V_{PDR}$ : POR电源电压下降检测电压。

## 28.7 用作中断&复位模式时的设定

通过选项字节 000C1H 设定运行模式（中断&复位模式（LVIMDS1、LVIMDS0=1、0）和检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ ）。

如果设定中断&复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“00H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“0”（高电压检测电平： $V_{LVDH}$ ）。

### ●LVD 中断&复位模式的运行

当接通电源时，中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（ $V_{DD}$ ）超过高电压检测电平（ $V_{LVDH}$ ）前保持 LVD 的内部复位状态。如果电源电压（ $V_{DD}$ ）超过高电压检测电平（ $V_{LVDH}$ ），就解除内部复位。

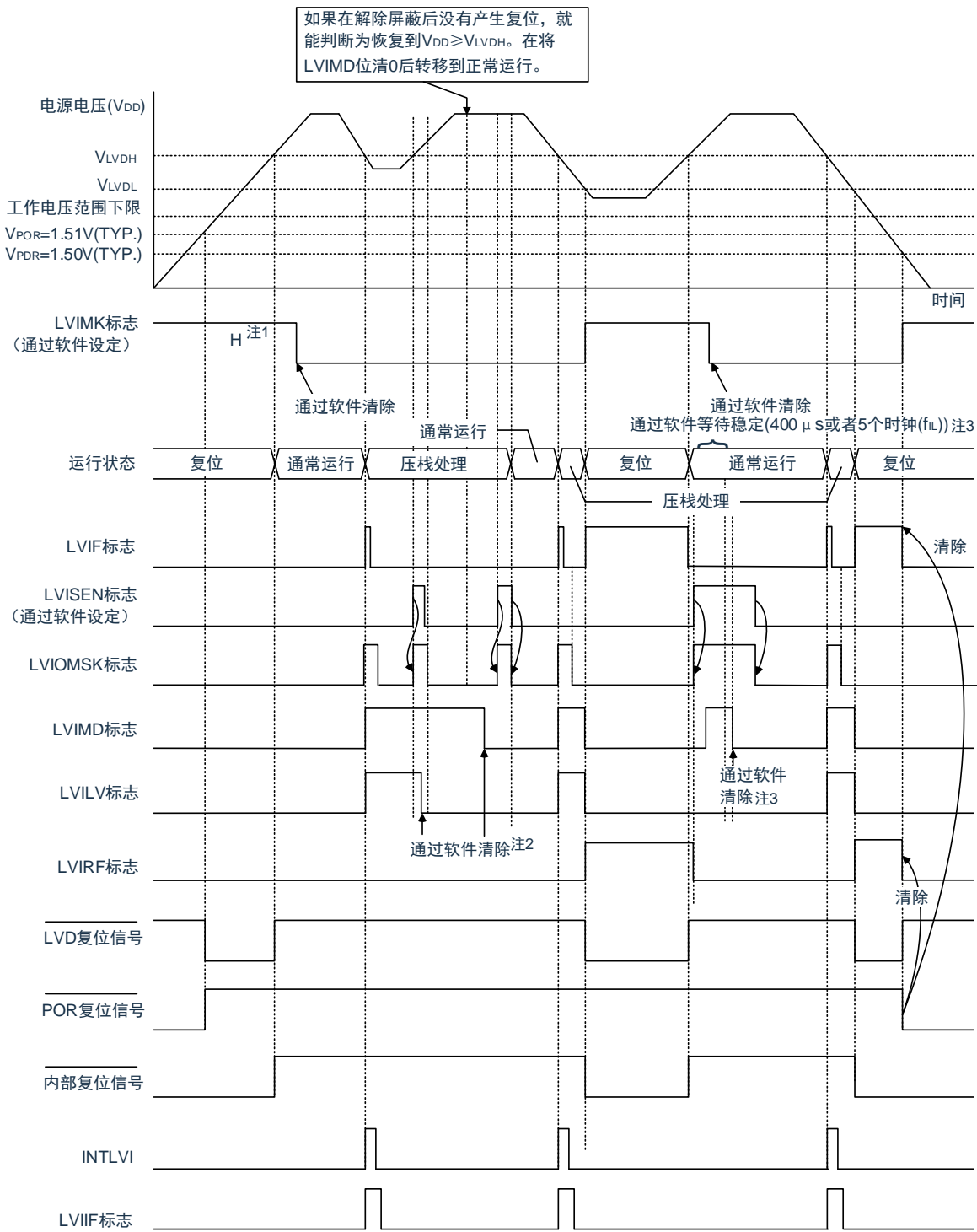
当工作电压下降时，如果电源电压（ $V_{DD}$ ）低于高电压检测电平（ $V_{LVDH}$ ），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（ $V_{DD}$ ）低于低电压检测电平

（ $V_{LVDL}$ ），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压（ $V_{DD}$ ）在不低于低电压检测电压（ $V_{LVDL}$ ）的状态下恢复到高电压检测电压（ $V_{LVDH}$ ）或者更高，也不产生中断请求信号。

当使用 LVD 中断&复位模式时，必须按照“图 28-6 工作电压的确认/复位的设定步骤”和“图 28-7 中断&复位模式的初始设定步骤”所示的流程图的步骤进行设定。

LVD 中断&复位模式的内部复位信号和中断信号的产生时序如图 28-4 示。

图 28-4 复位&中断信号的产生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (1/2)



注1: 在产生复位信号后, LVIMK标志变为“1”。

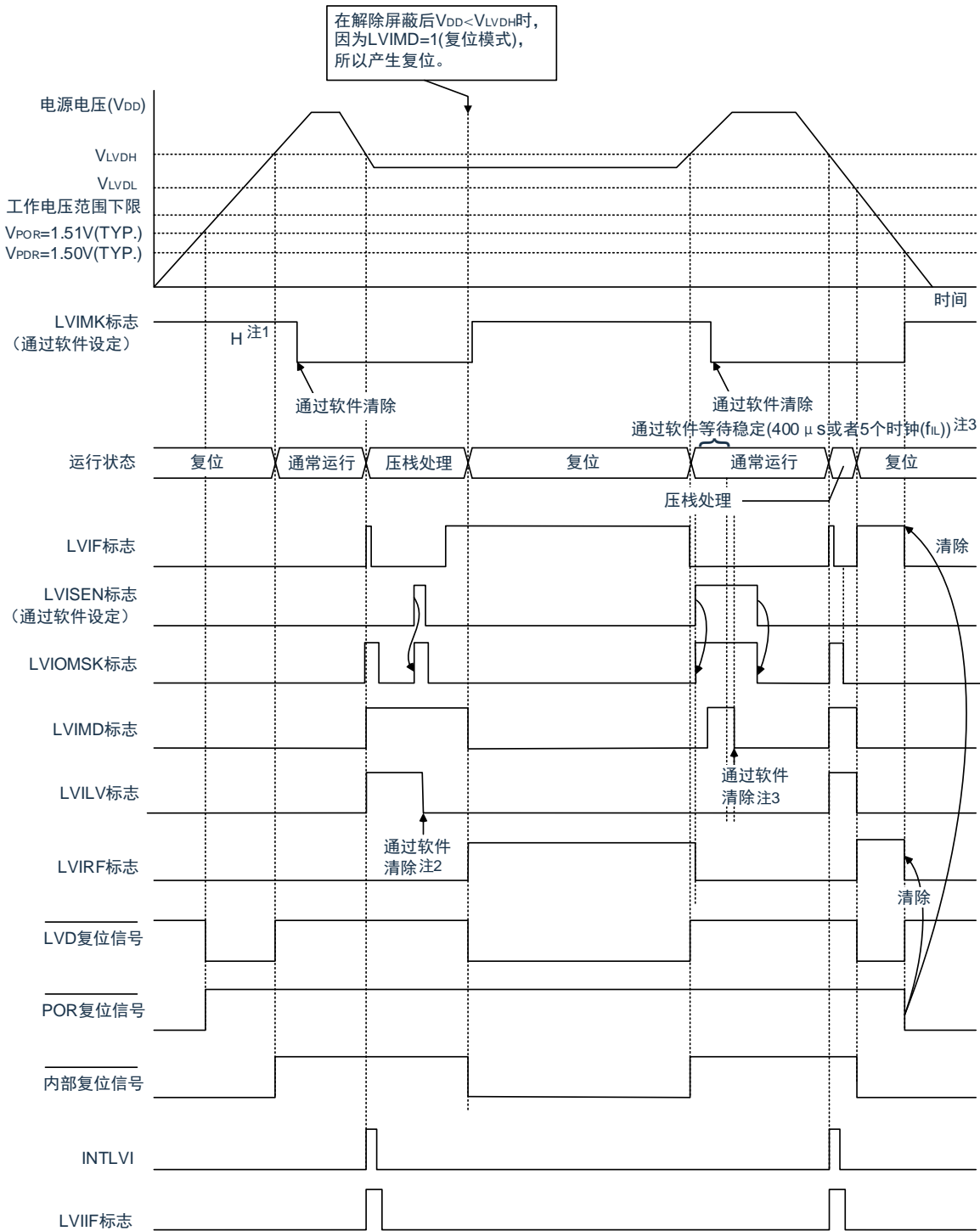
注2: 当使用中断&复位模式时, 必须在发生中断后按照“图28-6 工作电压的确认/复位的设定步骤”进行设定。

注3: 当使用中断&复位模式时, 必须在解除复位后按照“图28-7 中断&复位模式的初始设定步骤”进行设定。

注4:  $V_{POR}$ : POR电源电压上升检测电压。

$V_{PDR}$ : POR电源电压下降检测电压。

图 28-5 中断&复位信号的产生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (2/2)



注1: 在产生复位信号后, LVIMK标志变为“1”。

注2: 当使用中断&复位模式时, 必须在发生中断后按照“图28-6 工作电压的确认/复位的设定步骤”进行设定。

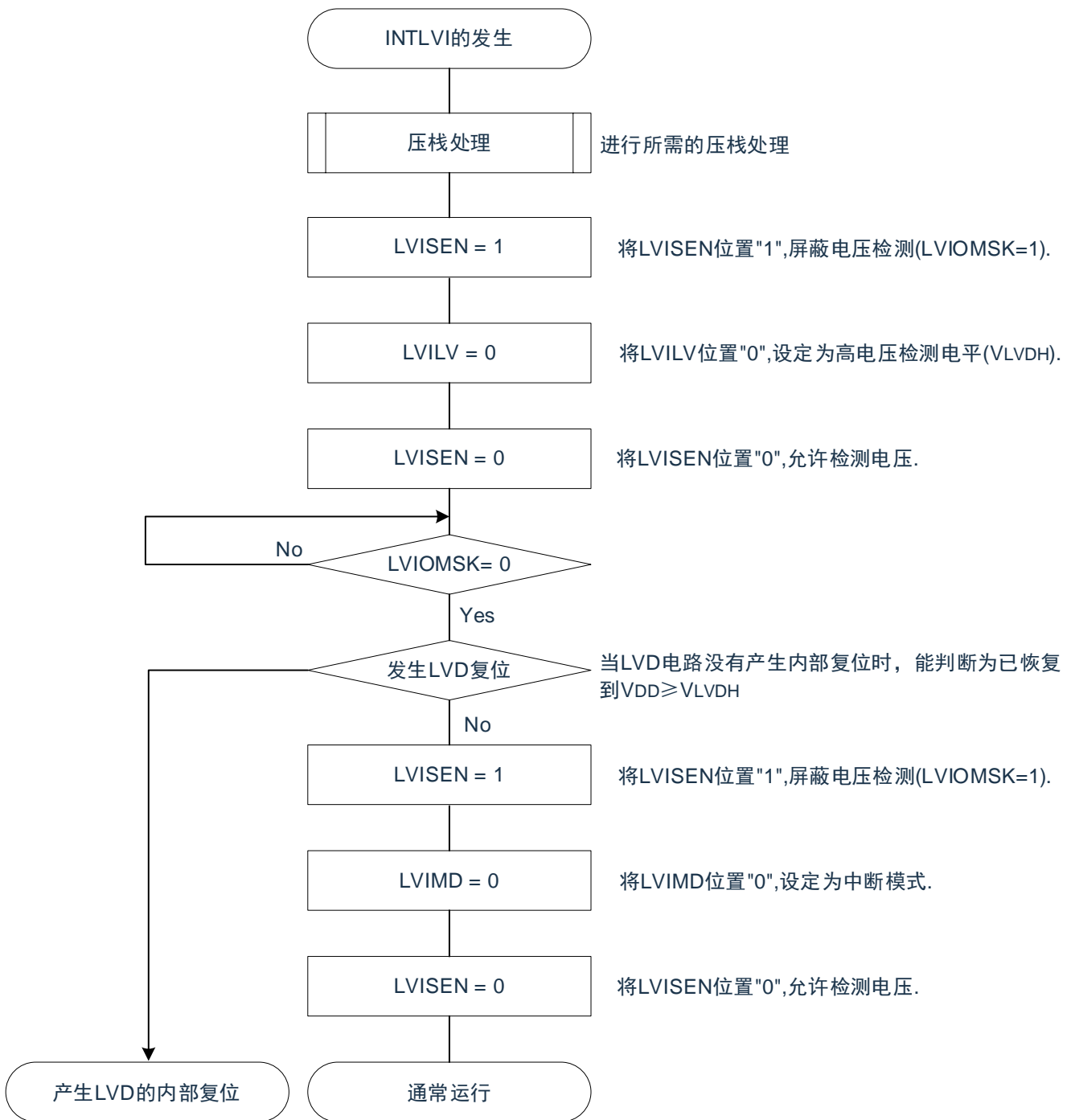
注3: 当使用中断&复位模式时, 必须在解除复位后按照“图28-7 中断&复位模式的初始设定步骤”进行设定。

备注:  $V_{POR}$ : POR电源电压上升检测电压。

$V_{PDR}$ : POR电源电压下降检测电压。



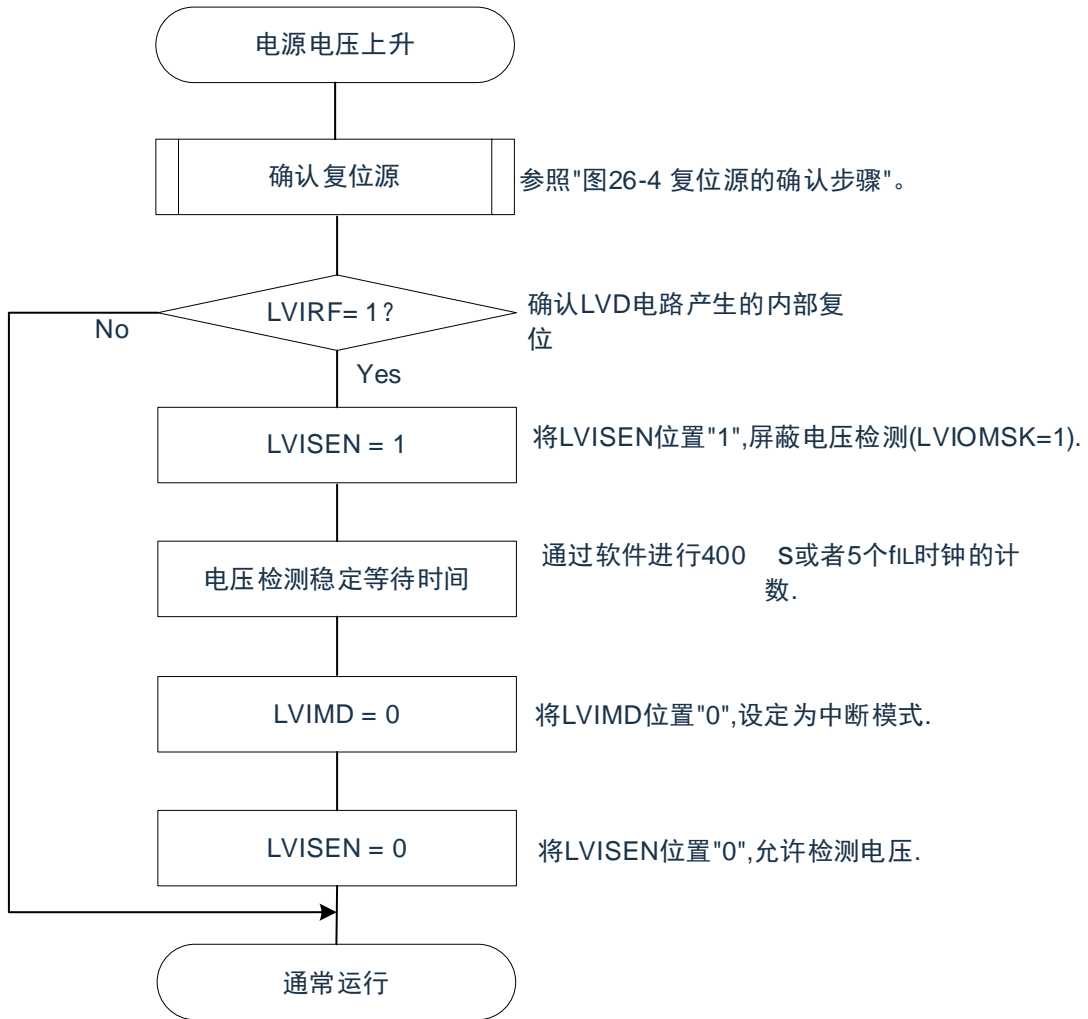
图28-6 工作电压的确认/复位的设定步骤



如果设定中断&复位模式 (LVIMDS1、LVIMDS0=1、0)，就在解除 LVD 复位 (LVIRF=1) 后需要 400us 或者 5 个  $f_{IL}$  时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将 LVIMD 位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写 LVIMD 位时，必须将 LVISEN 位置“1”，屏蔽 LVD 产生的复位或者中断的产生。

中断&复位模式的初始设定步骤如图 28-7 所示。

图28-7 中断&复位模式的初始设定步骤



备注:  $f_{IL}$ : 低速内部振荡器时钟频率

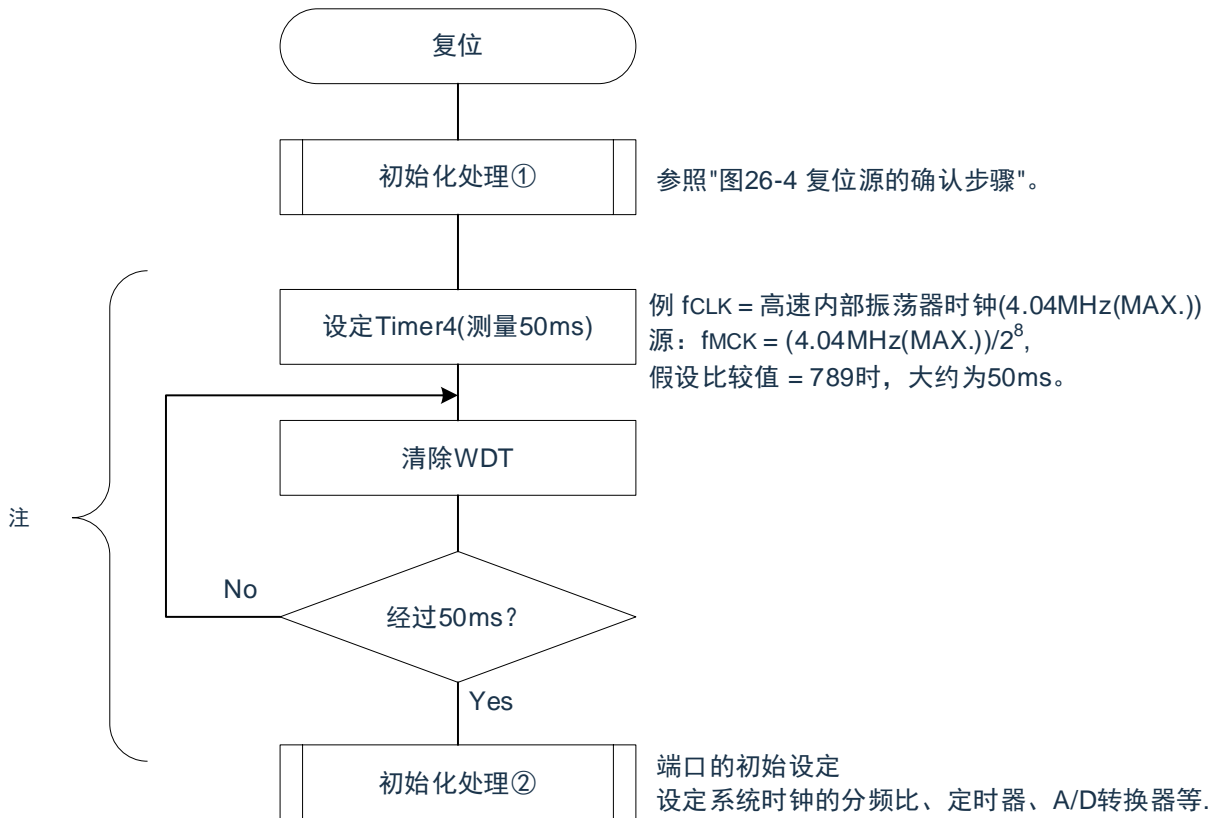
## 28.8 电压检测电路的注意事项

### (1) 有关接通电源时的电压波动

对于电源电压 ( $V_{DD}$ ) 在 LVD 检测电压附近发生一定时间波动的系统, 有可能重复进入复位状态和复位解除状态。能通过以下的处理, 任意设定解除复位到单片机开始运行的时间。

在解除复位后, 必须通过使用定时器的软件计数器, 在等待各系统不同的电源电压波动时间后进行端口等的初始设定。

图 28-8 LVD 检测电压附近的电源电压波动不超过 50ms 时的软件处理例子

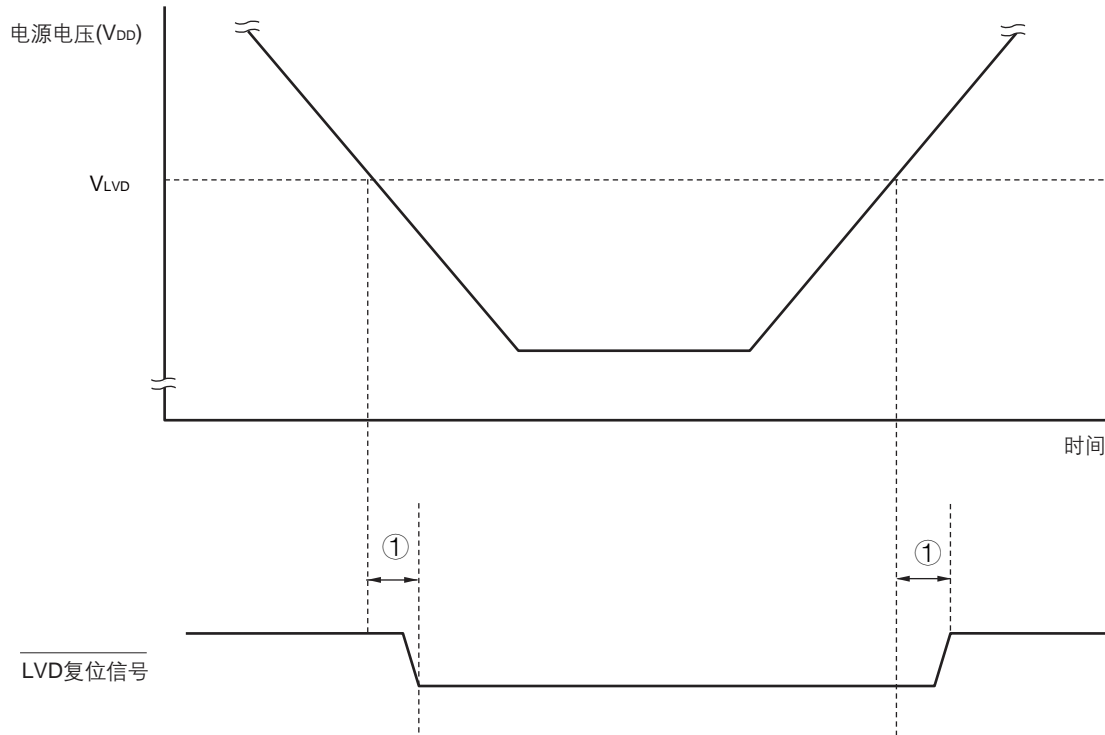


注1: 如果在此期间再次发生复位, 就不转移到初始化处理②。

(2)从产生LVD复位源到产生或者解除LVD复位的延迟

从满足电源电压 ( $V_{DD}$ ) < LVD 检测电压 ( $V_{LVD}$ ) 到产生 LVD 复位为止会发生延迟。同样，从 LVD 检测电压 ( $V_{LVD}$ ) ≤ 电源电压 ( $V_{DD}$ ) 到解除 LVD 复位为止也会发生延迟 (参照图 28-9)。

图28-9 从产生LVD复位源到产生或者解除LVD复位的延迟



① : 检测延迟 (300us(MAX.))

(3) 有关将 LVD 置为 OFF 时接通电源的情况

当将 LVD 置为 OFF 时，必须使用 RESETB 引脚的外部复位。

在进行外部复位时，必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位，就必须在给 RESETB 引脚输入低电平后接通电源，而且在数据手册的 AC 特性所示的工作电压范围内至少保持 10us 的低电平，然后输入高电平。

(4) 有关将 LVD 置为 OFF 并且设定为 LVD 中断模式时工作电压下降的情况

在将 LVD 置为 OFF 并且设定为 LVD 中断模式的情况下，如果工作电压下降，就必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

## 第29章 安全功能

### 29.1 安全功能的概要

为了对应 IEC60730 和 IEC61508 安全标准，CMS32M67xx 内置以下安全功能。

此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

#### (1) 闪存 CRC 运算功能（高速 CRC、通用 CRC）

通过 CRC 运算检测闪存的数据错误。能根据不同的用途和使用条件，分别使用以下 2 个 CRC。

- “高速 CRC”... 在初始化程序中，能停止 CPU 的运行并且高速检查整个代码闪存区。
- “通用 CRC”... 在 CPU 运行中，不限于代码闪存区而能用于多用途的检查。

#### (2) SFR 保护功能

防止因 CPU 失控而改写 SFR。

#### (3) 频率检测功能

能使用通用定时器单元进行 CPU/外围硬件时钟频率的自检。

#### (4) A/D 测试功能

能通过 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (ANI)、温度传感器输出和内部基准电压输出的 A/D 转换进行 A/D 转换器的自检。

#### (5) 输入/输出端口的数字输出信号电平检测功能

在输入/输出端口为输出模式时，能读引脚的输出电平。

## 29.2 寄存器映射

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
•闪存CRC控制寄存器 (CRC0CTL) •闪存CRC运算结果寄存器 (PGCRCL)	闪存CRC运算功能 (高速CRC)
•CRC输入寄存器 (CRCIN) •CRC数据寄存器 (CRCD)	CRC运算功能 (通用CRC)
•特殊SFR保护控制寄存器 (SFRGD)	SFR保护功能
•定时器输入选择寄存器0 (TIOS0)	频率检测功能
•A/D测试寄存器 (CON2)	A/D测试功能
•端口模式选择寄存器 (PMS)	输入/输出引脚的数字输出信号电平检测功能

(闪存CRC基地址 = 0x4002\_1810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CRC0CTL	0x000	R/W	闪存CRC控制寄存器	0x0
PGCRCL	0x002	R/W	闪存CRC运算结果寄存器	0x0

(通用CRC基地址 = 0x4004\_32FA) RO: 只读, WO: 只写, R/W: 读写

寄存器	物理地址	读/写	描述	复位值
CRCIN	0x4004_33AC	R/W	闪存CRC控制寄存器	0x0
CRCD	0x4004_32FA	R/W	闪存CRC运算结果寄存器	0x0

(SFR基地址 = 0x4004\_0478) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
SFRGD	0x000	R/W	SFR保护控制寄存器	0x0

(端口控制基地址 = 0x4004\_087B) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PMS	0x000	R/W	端口模式选择寄存器	0x0

(UID基地址 = 0x0050\_0894) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
UID0	0x000	RO	产品唯一的身份标识的第[31:0]位	-
UID1	0X004	RO	产品唯一的身份标识的第[63:32]位	-
UID2	0X008	RO	产品唯一的身份标识的第[95:64]位	-
UID3	0X00C	RO	产品唯一的身份标识的第[127:96]位	-

## 29.3 安全功能的运行

### 29.3.1 闪存CRC运算功能（高速CRC）

IEC60730 标准要求确认闪存中的数据，并且建议 CRC 为确认手段。此高速 CRC 能在初始设定（初始化）程序中检查整个代码闪存区。

高速 CRC 停止 CPU 的运行并且通过 1 个时钟从闪存读 32 位数据进行运算。因此，其特点是完成检查的时间较短（例如，128KB 闪存：1820.44us@72MHz<sup>注1</sup>）。

CRC 生成多项式对应 CRC-16-CCITT 的“X16+X12+X5+1”。

以 bit31--bit0 的 MSB 优先进行运算。

注1：此为系统时钟，系统时钟与闪存CRC运算的时钟的关系为4:1，系统时钟为72MHZ，则闪存CRC的运算时钟为18MHZ。

注2：因为通用CRC为LSB优先，所以运算结果不同。

闪存 CRC 控制寄存器（CRC0CTL）

这是设定高速 CRC 运算器的运行控制和运算范围的寄存器。通过 8 位存储器操作指令设定 CRC0CTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	CRC0EN	高速CRC运算器的运行控制 0: 停止运行 1: 通过执行 WFE 指令开始运算	0
6	CRCCHK124	124K演算范围选择 0: 演算范围由[5:0]位控制 1: 00000H~1EFFFH(124K-4byte)	0
5:0	FEA	高速CRC的演算范围 0000: 00000H~1FFBH(8K-4byte) 0001: 00000H~3FFBH(16K-4byte) 0010: 00000H~5FFBH(24K-4byte) 0011: 00000H~7FFBH(32K-4byte) 0100: 00000H~9FFBH(40K-4byte) 0101: 00000H~BFFBH(48K-4byte) 0110: 00000H~DFFBH(56K-4byte) 0111: 00000H~FFFH(64K-4byte) 1000: 00000H~11FFBH(72K-4byte) 1001: 00000H~13FFBH(80K-4byte) 1010: 00000H~15FFBH(88K-4byte) 1011: 00000H~17FFBH(96K-4byte) 1100: 00000H~19FFBH(104K-4byte) 1101: 00000H~1BFFBH(112K-4byte) 1110: 00000H~1DFFBH(120K-4byte) 1111: 00000H~1FFFH(128K-4byte)	0x0

注1：bit4~5必须设置为0。

注2：必须事先将用于比较的CRC运算结果期待值存入闪存的最后4字节，因此运算范围为减去4字节的范围。

### 29.3.1.1 闪存CRC运算结果寄存器L (PGCRCL)

这是保存高速 CRC 运算结果的寄存器。

通过 16 位存储器操作指令设定 PGCRCL 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15:0	PGCRCL	保存高速 CRC 运算结果 0000H ~FFFFH	0x0

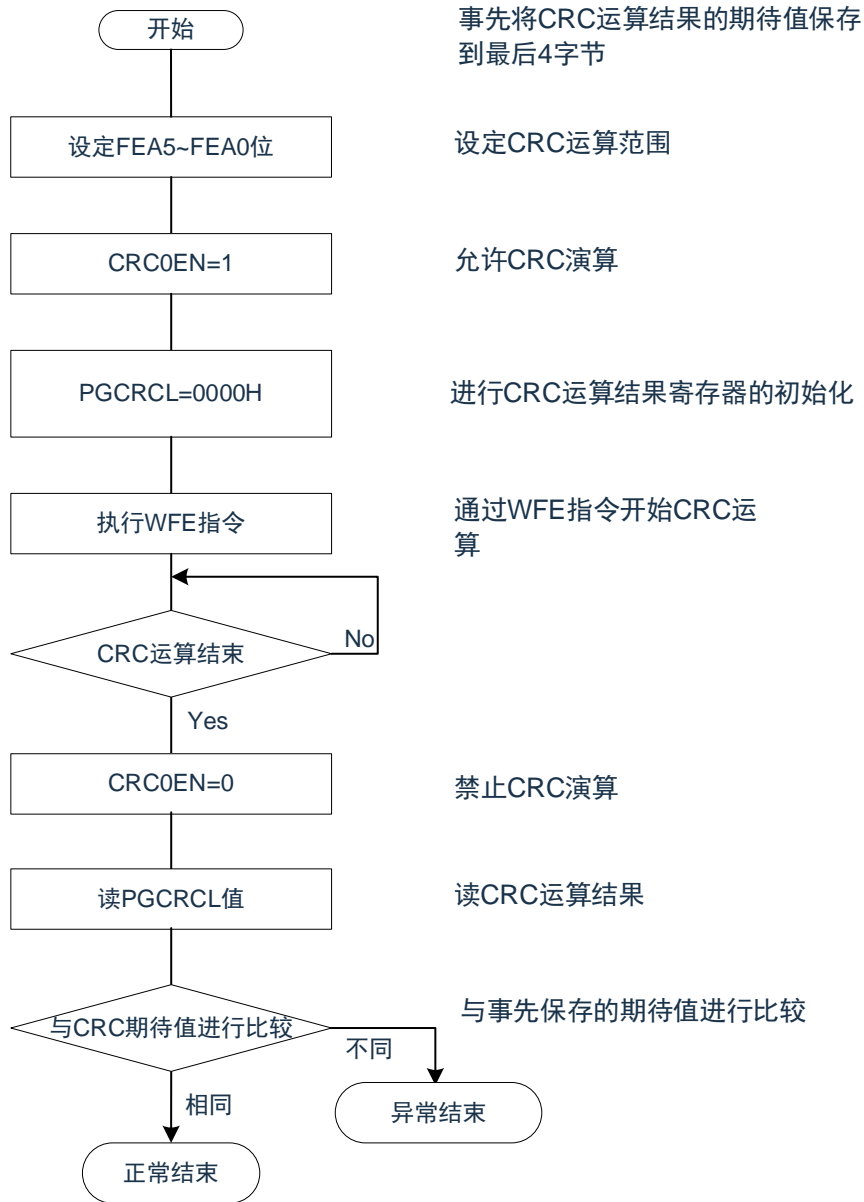
注：只有在CRC0EN（CRC0CTL寄存器的bit7）位为“1”时才能写PGCRCL寄存器。



闪存CRC运算功能（高速CRC）的流程图如图29-1所示。

<操作流程>

图29-1：闪存CRC运算功能（高速CRC）的流程图



注1：只以代码闪存为CRC运算的对象。

注2：必须将CRC运算的期待值保存在代码闪存中的运算范围后的区域。

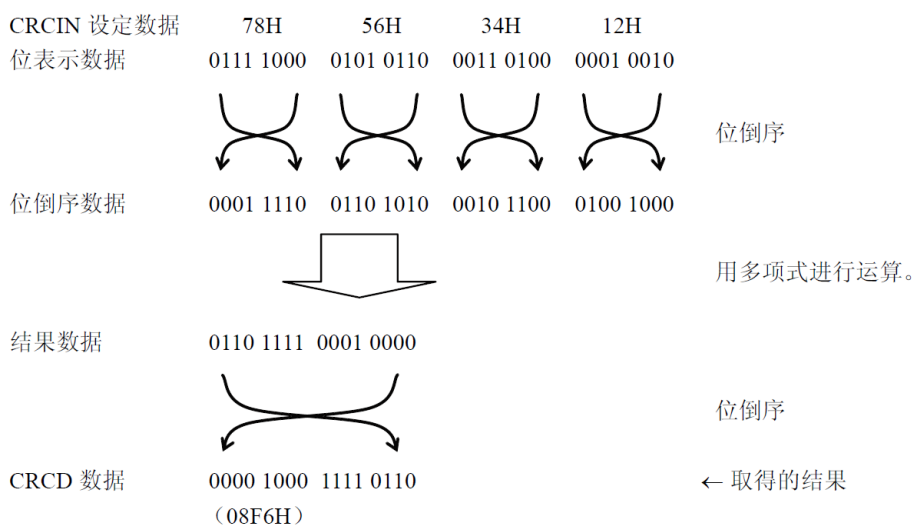
### 29.3.1.2 CRC运算功能（通用CRC）

为了必须保证运行过程中的安全，IEC61508 标准要求即使在 CPU 运行中也需要确认数据。

此通用 CRC 能在 CPU 运行中作为外围功能进行 CRC 运算。通用 CRC 不限于代码闪存区而能用于多用途的检查。通过软件（用户程序）指定要确认的数据。

在主系统时钟运行模式运行模式中，都能使用 CRC 运算功能。

CRC 生成多项式使用 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。因为考虑到是以 LSB 优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从 LSB 发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给 CRCIN 寄存器写值，从 CRCD 寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行 CRC 运算的结果。



注：在执行程序的过程中，因为调式程序将软件断点的设定行改写为断点指令，所以如果在 CRC 运算的对象区设定软件断点，CRC 的运算结果就不同。

### 29.3.1.3 CRC输入寄存器（CRCIN）

这是设定通用CRC的CRC计算数据的8位寄存器。能设定的范围为“00H~FFH”。

通过8位存储器操作指令设定CRCIN寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:0	CRCIN	通用 CRC 数据输入 00H ~FFH	0x0

### 29.3.1.4 CRC数据寄存器 (CRCD)

这是保存通用 CRC 运算结果的寄存器。能设定的范围为“0000H~FFFFH”。

在写 CRCIN 寄存器后经过 1 个 CPU/外围硬件时钟 ( $f_{CLK}$ )，将 CRC 运算结果保存到 CRCD 寄存器。通过 16 位存储器操作指令设定 CRCD 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

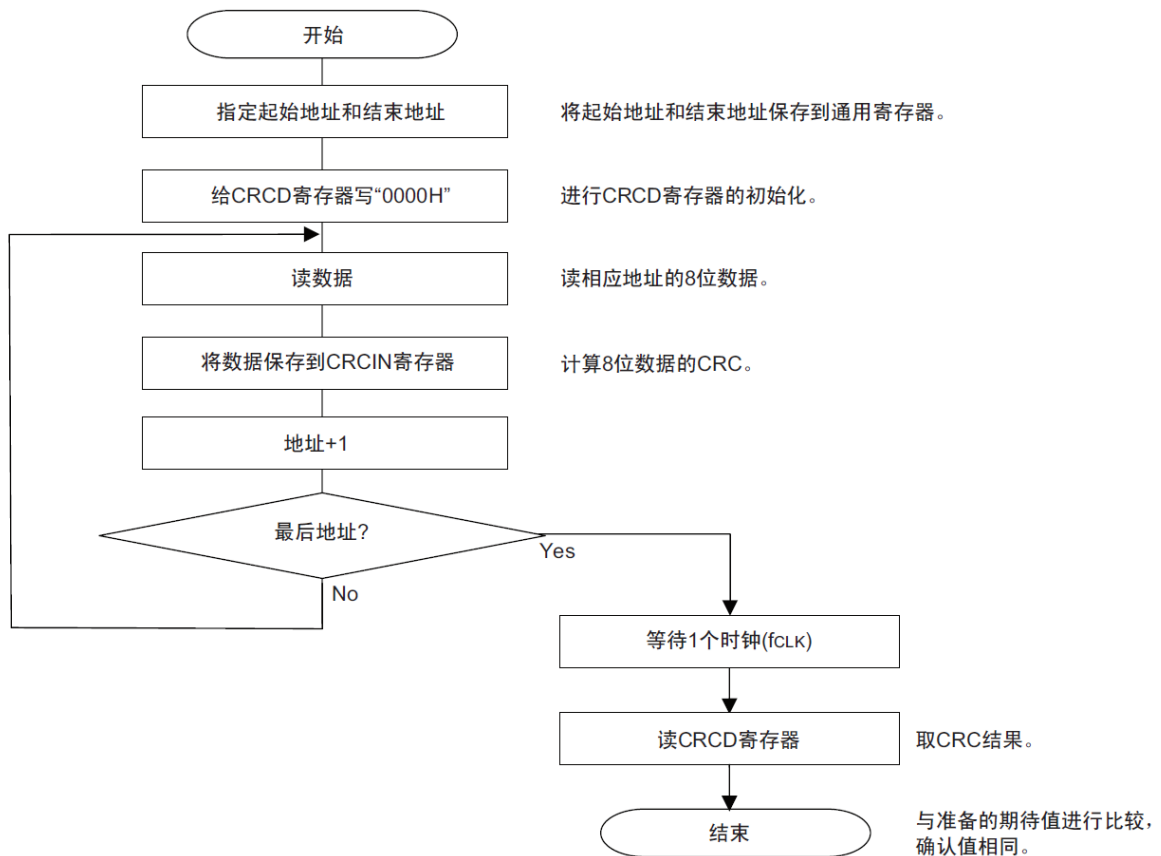
位	符号	描述	复位值
15:0	CRCD	保存通用 CRC 运算结果(0000H ~FFFFH)	0x0

注1: 要读CRCD寄存器的写入值时，必须在写CRCIN寄存器前读CRCD寄存器。

注2: 如果CRCD寄存器的写操作与运算结果的保存发生竞争，就忽视写操作。

#### <操作流程>

图29-2 CRC运算功能（通用CRC）的流程图



## 29.3.2 SFR保护功能

为了必须保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护重要的 SFR，使其免遭改写 SFR 保护功能用于保护端口功能、时钟控制功能和电压检测电路的控制寄存器的数据。

如果设定为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

### 29.3.2.1 SFR保护控制寄存器（SFRGD）

此寄存器控制 SFR 保护功能是否有效。

SFR 保护功能使用 GPORT 位和 GCSC 位。

通过 8 位存储器操作指令设定 SFRGD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:3	保留	-	-
2	GPORT	端口功能的控制寄存器的保护 0: 无效，能读写端口功能的控制寄存器。 1: 有效，端口功能的控制寄存器的写操作无效，能读。 [被保护的 SFR]PMxx、PUxx、PDxx、POMxx、 PMCxx、PxxCFG <sup>注</sup> 。	0
1	-	保留	-
0	GCSC	时钟控制功能、电压检测电路的控制寄存器的保护 0: 无效，能读写时钟控制功能和电压检测电路的控制寄存器。 1: 有效。时钟控制功能和电压检测电路的控制寄存器的写操作无效，能读。 [被保护的 SFR]CSC、OSTS、CKC、PERx、 OSMC、LVIM、LVIS。	0

注：不保护Pxx（端口寄存器）。

### 29.3.3 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

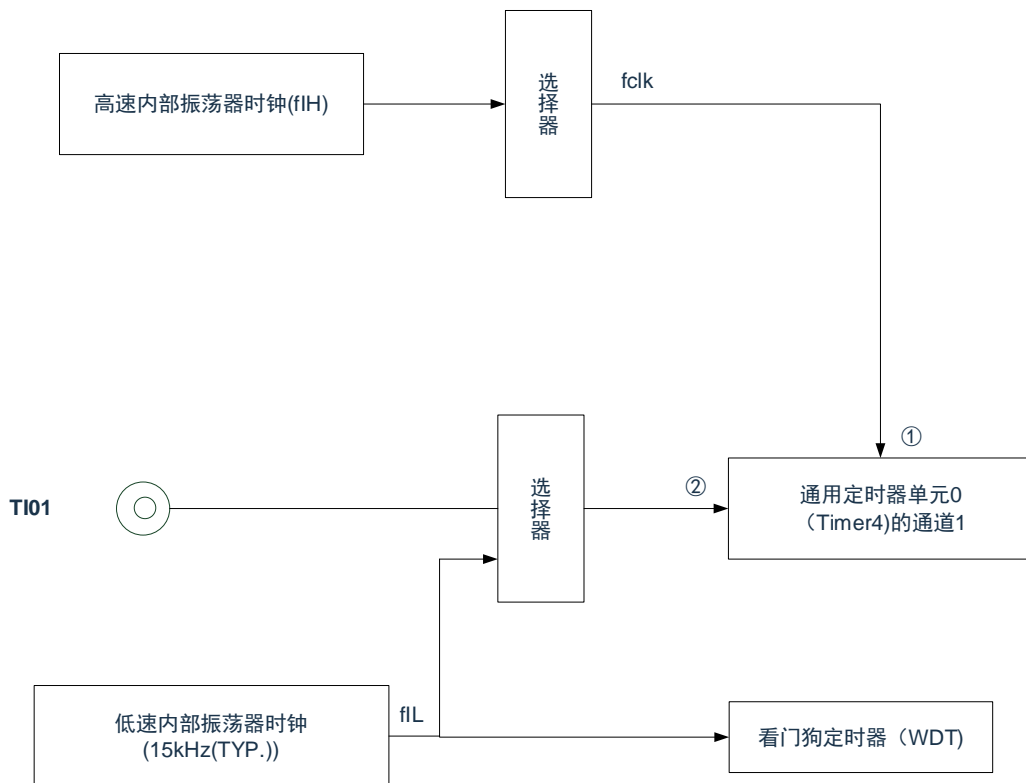
频率检测功能可使用 CPU/外围硬件的时钟频率 ( $f_{CLK}$ )，并且能通过测量 Timer4 道 1 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果某 1 个时钟或者 2 个时钟停止振荡，就不能判断 2 个时钟的比率关系。

<要比较的时钟>

- ①CPU/外围硬件的时钟频率 ( $f_{CLK}$ ):
- 高速内部振荡器时钟 ( $f_{IH}$ )
- ②Timer4 通道 1 输入:
- 通道 1 的定时器输入 (TI01)
- 低速内部振荡器时钟 ( $f_{IL}$ : 15kHz(TYP.))

图29-3 频率检测功能的结构



当输入脉冲间隔的测量结果为异常值时，能判断为“时钟频率异常”。有关输入脉冲间隔的测量方法，请参照“5.7.4 作为输入脉冲间隔测量的运行”。

注：只有内置副系统时钟的产品才能选择。

#### 29.3.3.1 定时器输入输出选择寄存器0 (TIOS0)

寄存器说明请参考5.2.11节。

## 29.3.4 A/D测试功能

IEC60730标准要求对A/D转换器进行测试。此A/D测试功能通过对A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(AN)、温度传感器的输出电压和内部基准电压进行A/D转换,确认A/D转换器是否正常运行。

能通过以下步骤确认模拟多路转换器:

- 1) 通过CON2寄存器选择ANx引脚作为A/D转换对象(ADCSWCHS = 00100)。
- 2) 对ANx引脚进行A/D转换(转换结果1-1)。
- 3) 通过CON2寄存器选择A/D转换器的负(-)基准电压作为A/D转换对象(ADCSWCHS = 11000)。
- 4) 对A/D转换器的负(-)基准电压进行A/D转换(转换结果2-1)。
- 5) 通过CON2寄存器选择ANx引脚作为A/D转换对象(ADCSWCHS = 00100)。
- 6) 对ANx引脚进行A/D转换(转换结果1-2)。
- 7) 通过CON2寄存器选择A/D转换器的正(+)基准电压作为A/D转换对象(ADCSWCHS = 10111)。
- 8) 对A/D转换器的正(+)基准电压进行A/D转换(转换结果2-2)。
- 9) 通过CON2寄存器选择ANx引脚作为A/D转换对象(ADCSWCHS = 00100)。
- 10) 对ANx引脚进行A/D转换(转换结果1-3)。
- 11) 确认“转换结果1-1”、“转换结果1-2”和“转换结果1-3”相同。
- 12) 确认“转换结果2-1”的A/D转换结果全部为“0”并且“转换结果2-2”的A/D转换结果全部为“1”。通过以上步骤,能选择模拟多路转换器以及确认布线没有断线。

注1: 在1)~10)的转换过程中,如果模拟输入电压可变,就必须采用其他方法来确认模拟多路转换器。

注2: 转换结果含有误差,因此必须在比较转换结果时要适当考虑误差。

### 29.3.4.1 A/D测试寄存器(CON2)

此寄存器选择A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(ANxx)、温度传感器的输出电压和内部基准电压(1.45V)作为A/D转换对象。

当用作A/D测试功能时,进行以下的设定:

- 在测量零刻度时,选择负(-)基准电压作为A/D转换对象。
- 在测量满刻度时,选择正(+)基准电压作为A/D转换对象。

A/D寄存器说明请参考19.5.2。

## 29.3.5 输入/输出引脚的数字输出信号电平检测功能

IEC60730 标准要求确认 I/O 功能是否正常。

输入/输出引脚的数字输出信号电平检测功能在引脚为输出模式时，能读引脚的数字输出电平。

### 29.3.5.1 端口模式选择寄存器（PMS）

此寄存器选择在引脚为输出模式（端口模式寄存器（PMm）的 PMmn 位为“0”）时是读端口的输出锁存器的值还是读引脚的输出电平。

通过 8 位存储器操作指令设定 PMS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	-	保留	-
0	PMS0	在引脚为输出模式时读数据的选择 0: 读 Pmn 寄存器的值。 1: 读引脚的数字输出电平。	0

注：m=0~5，n=0~7。

## 29.3.6 产品唯一身份标识寄存器

产品唯一的身份标识非常适合：

用来作为序列号(例如 USB 字符序列号或者其他的终端应用)

用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器内的安全性。

用来激活带安全机制的自举过程

128 位的产品唯一身份标识所提供的参考号码对任意一个微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

产品唯一身份标识寄存器 0 (UID0)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[31:0]位，其值在出厂时编写	-

产品唯一身份标识寄存器 1 (UID1)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[63:32]位，其值在出厂时编写	-

产品唯一身份标识寄存器 2 (UID2)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[95:64]位，其值在出厂时编写	-

产品唯一身份标识寄存器 3 (UID3)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[127:96]位，其值在出厂时编写	-

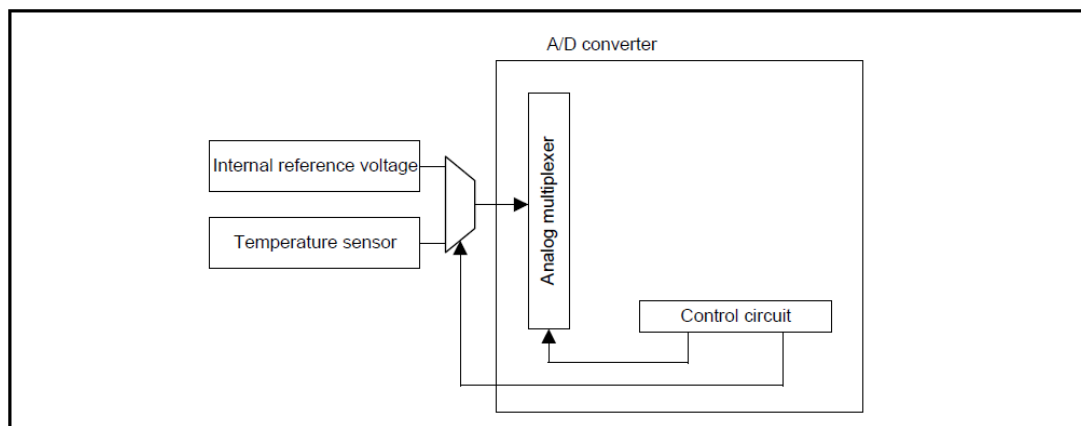


## 第30章 温度传感器

### 30.1 温度传感器的功能

片上的温度传感器可以对产品的核心温度进行测量和监控，从而保证产品的可靠运行。温度传感器输出的电压与核心温度成正比，并且电压和温度之间是线性关系。其输出电压提供给 ADC 进行转换。图 30-1 显示了温度传感器框图。

图 30-1温度传感器框图



### 30.2 寄存器映射

RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
TSN25	0x0050066C	RO	温度传感器校准数据寄存器	-

### 30.3 温度传感器的寄存器

#### 30.3.1 温度传感器校准数据寄存器 TSN25

位	符号	描述	复位值
15:12	-	保留	-
11:0	TSN25	校准数据，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。	-

## 30.4 温度传感器的使用说明

温度 (T) 与传感器电压输出 (Vs) 成正比, 因此温度的计算公式如下:

$$T = (V_s - V_1) / \text{slope} + 25^\circ\text{C}$$

T: 测量的温度 (°C)

Vs: 温度传感器在温度测量时的输出电压 (V)

V1: 温度传感器测量25°C时的电压输出 (V)

Slope: 温度传感器的温度斜率(V/°C), slope = -3.5 mV/°C

备注: 温度传感器的精度较低, 不建议使用在精度要求高的场合。

## 第31章 选项字节

### 31.1 选项字节的功能

芯片的闪存 000C0H~000C3H, 500004H 为选项字节区。

选项字节由用户选项字节 (000C0H~000C2H) 和闪存数据保护选项字节 (000C3H, 500004H) 构成。在接通电源或者复位启动时, 自动参照选项字节进行指定功能的设定。在使用本产品时, 必须通过选项字节进行以下功能的设定。对于没有配置功能的位, 不能更改初始值。

注意: 是否使用各功能无关, 必须设定选项字节。

#### 31.1.1 用户选项字节 (000C0H~000C2H)

##### (1) 000C0H

- 看门狗定时器的运行
  - 允许或者禁止计数器的运行。
  - 在睡眠/深度睡眠模式中允许或者停止计数器的运行。
- 看门狗定时器的上溢时间的设定
  - 看门狗定时器的窗口打开期间的设定
- 看门狗定时器的间隔中断的设定
  - 使用或者不使用间隔中断。

##### (2) 000C1H

- LVD运行模式的设定
  - 中断&复位模式
  - 复位模式
  - 中断模式
  - LVD为OFF (使用RESETB引脚的外部复位输入)。
- LVD检测电平 ( $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ ) 的设定

注1: 当电源电压上升时, 必须在电源电压达到数据手册的AC特性所示的工作电压范围前, 通过电压检测电路或者外部复位保持复位状态; 当电源电压下降时, 必须在电源电压低于工作电压范围前, 通过深度睡眠模式的转移、电压检测电路或者外部复位, 置为复位状态。

注2: 工作电压范围取决于用户选项字节 (000C2H) 的设定。

##### (3) 000C2H

- 高速内部振荡器的频率设定
  - 从 2MHz~36MHz、64MHz、72MHz 中选择。

## 31.2 闪存数据保护选项字节（000C3H，500004H）

- 片上调试时闪存数据保护的

Level0：允许通过 debugger 对闪存数据进行读出/写入/擦除操作

Level1：允许通过 debugger 对闪存数据进行 chip 全擦除操作，不允许进行读写操作。

Level2：不允许通过 debugger 对闪存数据进行操作。

## 31.3 寄存器映射

寄存器	地址	读/写	描述	复位值
选项字节0	0x00C0H	R/W	看门狗定时器状态控制寄存器	0xFF
选项字节1	0x00C1H	R/W	LVD状态控制寄存器	0xFF
选项字节2	0x00C2H	R/W	高速内部振荡器频率控制寄存器	0xEC
选项字节3	0x00C3H	R/W	片上调试闪存数据保护控制寄存器1	0xFF
选项字节4	0x500004H	R/W	片上调试闪存数据保护控制寄存器2	0xFF

## 31.4 用户选项字节

### 31.4.1 用户选项字节（000C0H）

位	符号	描述	复位值
7	WDTINT	看门狗定时器的间隔中断的使用/不使用 0: 不使用间隔中断 1: 当达到上溢时间的75%+1/2f <sub>IL</sub> 时, 产生间隔中断	1
6:5	WINDOW[1:0]	看门狗定时器的窗口打开期间 0X: 禁止设定。 10: 75% 11: 100%	0x3
4	WDTON	看门狗定时器的计数器运行控制 0: 禁止计数器的运行 (解除复位后停止计数) 1: 允许计数器的运行 (解除复位后开始计数)	1
3:1	WDTCS[2:0]	看门狗定时器的上溢时间 (f <sub>IL</sub> =15kHz) 000: 2 <sup>6</sup> /f <sub>IL</sub> (4.3ms) 001: 2 <sup>7</sup> /f <sub>IL</sub> (8.5ms) 010: 2 <sup>8</sup> /f <sub>IL</sub> (17.0ms) 011: 2 <sup>9</sup> /f <sub>IL</sub> (34.0ms) 100: 2 <sup>11</sup> /f <sub>IL</sub> (135.9ms) 101: 2 <sup>13</sup> /f <sub>IL</sub> (543.5ms) 110: 2 <sup>14</sup> /f <sub>IL</sub> (1086.9ms) 111: 2 <sup>16</sup> /f <sub>IL</sub> (4347.8ms)	0x7
0	WDSTBYON	看门狗定时器的计数器运行控制 (睡眠模式) 0: 在睡眠模式中, 停止计数器的运行 <sup>注1</sup> 1: 在睡眠模式中, 允许计数器的运行	1

注1: 当WDSTBYON位为“0”时, 与WINDOW1位和WINDOW0位的值无关, 窗口打开期间为100%。

注2: f<sub>IL</sub>: 低速内部振荡器的时钟频率。

### 31.4.2 用户选项字节（000C1H）

位	符号	描述	复位值
7:5	VPOC[2:0]	检测电压设定	0x7
4	-	保留（须为1）	1
3:2	LVIS[1:0]	检测电压设定	0x2
1:0	LVIMDS[1:0]	模式选择 10: 中断&复位模式 11: 复位模式 01: 中断模式	0x3

• LVD检测电压设定（中断&复位模式）

检测电压			选项字节的设定值						
V <sub>LVDH</sub>		V <sub>LVDL</sub>	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降	下降						LVIMDS1	LVIMDS0
1.77V	1.73V	1.63V	0	0	0	1	0	1	0
1.88V	1.84V					0	1		
2.92V	2.86V					0	0		
1.98V	1.94V	1.84V		0	1	1	0		
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V		1	0	1	0		
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V		1	1	1	0		
3.02V	2.96V					0	1		
4.06V	3.98V					0	0		
—			禁止设定上述以外的值。						

注1：必须给bit4写“1”。

注2：有关LVD电路的详细内容，请参照“第28章 电压检测电路”。

注3：检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

## •LVD检测电压设定（复位模式）

检测电压		选项字节的设定值								
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
上升	下降						LVIMDS1	LVIMDS0		
1.67V	1.63V	0	0	0	1	1	1	1		
1.77V	1.73V		0	0	1	0				
1.88V	1.84V		0	1	1	1				
1.98V	1.94V		0	1	1	0				
2.09V	2.04V		0	1	0	1				
2.50V	2.45V		1	0	1	1				
2.61V	2.55V		1	0	1	0				
2.71V	2.65V		1	0	0	1				
2.81V	2.75V		1	1	1	1				
2.92V	2.86V		1	1	1	0				
3.02V	2.96V		1	1	0	1				
3.13V	3.06V		0	1	0	0				
3.75V	3.67V		1	0	0	0				
4.06V	3.98V		1	1	0	0				
—			禁止设定上述以外的值。							

注1：必须给bit4写“1”。

注2：有关LVD电路的详细内容，请参照“第28章 电压检测电路”。

注3：检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

## • LVD检测电压设定（中断模式）

检测电压		选项字节的设定值								
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
上升	下降						LVIMDS1	LVIMDS0		
1.67V	1.63V	0	0	0	1	1	0	1		
1.77V	1.73V		0	0	1	0				
1.88V	1.84V		0	1	1	1				
1.98V	1.94V		0	1	1	0				
2.09V	2.04V		0	1	0	1				
2.50V	2.45V		1	0	1	1				
2.61V	2.55V		1	0	1	0				
2.71V	2.65V		1	0	0	1				
2.81V	2.75V		1	1	1	1				
2.92V	2.86V		1	1	1	0				
3.02V	2.96V		1	1	0	1				
3.13V	3.06V		0	1	0	0				
3.75V	3.67V		1	0	0	0				
4.06V	3.98V		1	1	0	0				
—			禁止设定上述以外的值。							

注1：必须给bit4写“1”。

注2：有关LVD电路的详细内容，请参照“第28章 电压检测电路”。

注3：检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。



## • LVD为OFF时的设定（使用RESETB引脚的外部复位输入）

检测电压		选项字节的设定值						
$V_{LVDH}$		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值。						

注意1：必须给bit4写“1”。

注意2：当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H）的设定。

注1：×：忽略。

注2：有关LVD电路的详细内容，请参照“第28章 电压检测电路”。

注3：检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

### 31.4.3 用户选项字节（000C2H）

位	符号	描述	复位值
7:5	-	保留（须为1）	0x7
4:0	FRQSE[4:0]	高速内部振荡器时钟频率选择	0x0C

FRQSE4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率	
					f <sub>HOCO</sub>	f <sub>IH</sub>
1	x	0	0	0	72MHz	72MHz
1	x	0	0	1	72MHz	36MHz
1	x	0	1	0	72MHz	18MHz
1	x	0	1	1	72MHz	9MHz
1	x	1	0	0	72MHz	4.5MHz
0	x	0	0	0	64MHz	64MHz
0	x	0	0	1	64MHz	32MHz
0	x	0	1	0	64MHz	16MHz
0	x	0	1	1	64MHz	8MHz
0	x	1	0	0	64MHz	4MHz
0	x	1	0	1	64MHz	2MHz
上述以外					禁止设定。	

注1：必须给bit7~5写“1”。

注2：工作频率范围和工作电压范围因闪存各运行模式而不同。详细内容请参照数据手册的AC特性。

### 31.4.4 闪存数据保护选项字节（000C3H）

位	符号	描述	复位值
7:0	OCDEN[7:0]	闪存数据保护的位	0xFF

### 31.4.5 闪存数据保护选项字节（500004H）

位	符号	描述	复位值
7:0	OCDM[7:0]	闪存数据保护的位	0xFF

OCDM	OCDEN	闪存数据保护的位
3C	C3	不允许通过debugger对闪存数据进行操作。
3C以外的值	C3	允许通过debugger对闪存数据进行chip全擦除操作，不允许进行读写操作。
上記以外		允许通过debugger对闪存数据进行读出/写入/擦除操作

注：50\_0004H地址属于数据闪存区，如果使用该地址做数据存储用，需先确定数值不会引起保护选项的误设。

## 第32章 FLASH控制

### 32.1 FLASH控制功能描述

本制品包含一颗 128KByte 容量的 FLASH 存储器，共划分为 256 个 Sector，每个 Sector 容量为 512Byte。可做为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。

### 32.2 FLASH存储器结构



## 32.3 寄存器映射

(FLASH控制基地址 = 0x4002\_0000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
FLSTS	0x000	R/W	FLASH状态寄存器	0x0
FLOPMD1	0x004	R/W	FLASH操作控制寄存器1	0x0
FLOPMD2	0x008	R/W	FLASH操作控制寄存器2	0x0
FLERMD	0x00C	R/W	FLASH擦除控制寄存器	0x0
FLCERCNT	0x010	R/W	FLASH全片擦除时间控制寄存器	0xCE
FLSERCNT	0x014	R/W	FLASH页擦除时间控制寄存器	0x149
FLPROCNT	0x01C	R/W	FLASH写入时间控制寄存器	0XB0007E
FLPROT	0x020	R/W	FLASH写保护寄存器	0x0

## 32.4 寄存器说明

### 32.4.1 Flash写保护寄存器(FLPROT)

Flash保护寄存器用于对FLASH操作控制寄存器进行保护。

位	符号	描述	复位值
31:8	-	保留	-
7:1	PRKEY	WRP写保护 78h: 允许改写WRP 其他: 不允许改写WRP	0x0
0	WRP	操作寄存器 (FLOPMD1/FLOPMD2) 写保护 1: 允许改FLOPMD1/FLOPMD2 0: 不允许改写FLOPMD1/ FLOPMD2	0

### 32.4.2 FLASH操作控制寄存器(FLOPMD1)

Flash操作控制寄存器, 用于设定FLASH的擦除和写入操作。

位	符号	描述	复位值
31:8	-	保留	-
7:0	FLOPMD1	FLASH操作选择位: 0x55: 当FLOPMD2=0xAA时: 擦除操作 0xAA: 当FLOPMD2=0x55时: 写操作 0x00: 当FLOPMD2=0x00时: 读取操作 上記以外: 禁止设定	0x0

### 32.4.3 FLASH操作控制寄存器(FLOPMD2)

Flash操作控制寄存器，用于设定FLASH的擦除和写入操作。

位	符号	描述	复位值
31:8	-	保留	-
7:0	FLOPMD2	FLASH操作选择位： 0xAA: 当FLOPMD1=0x55时：擦除操作 0x55: 当FLOPMD1=0xAA时：写操作 0x00: 当FLOPMD1=0x00时：读取操作 上記以外：禁止设定	0x0

### 32.4.4 Flash擦除控制寄存器(FLERMD)

Flash擦除控制寄存器，用于设定FLASH擦除操作的类型。

位	符号	描述	复位值
7:5	-	保留	-
4:3	ERMD	擦除操作控制位： 0: sector擦除，擦除后不进行硬件校验 1: chip擦除（注） 2: sector擦除，擦除后进行硬件校验 3: 禁止设定	0x0
2:0	-	保留	-

注：chip擦除只擦除代码闪存区域，不擦除数据闪存区域。且chip擦除不支持硬件校验。

### 32.4.5 Flash状态寄存器(FLSTS)

通过状态寄存器可以查询FLASH控制器的状态。

位	符号	描述	复位值
7:3	-	保留	-
2	EVF <sup>(注2)</sup>	FLASH擦除硬件校验错误标志位： 0: 擦除后，硬件校验没有发生错误 1: 硬件校验发生错误	0
1	-	保留	-
0	OVF <sup>(注1)</sup>	FLASH擦写操作完成标志位： 0: 擦写操作未完成 1: 擦完操作完成擦除	0

注1：OVF需要软件写“1”进行清除。若不清除，不能进行下一次擦写操作。

注2：EVF需要软件写“1”进行清除。

### 32.4.6 Flash全片擦除时间控制寄存器(FLCERCNT)

通过FLCERCNT寄存器可以设置FLASH全片擦除的时间。

位	符号	描述	复位值
31	Load	全片擦除时间设定选择（注： 0: 使用硬件设定的擦除时间 1: 使用软件设定的擦除时间（FLCERCNT[10:0]）	0
30:11	-	保留	-
10:0	FLCERCNT	软件擦除时间设定： Chip擦除时间=（CERCNT*2048*Tfclk），需满足>30ms的硬件要求	0xCE

注：在主时钟为内部高速OCO时钟 $\leq 20\text{M}$ 时，可以使用硬件设定时间，不设定FLCERCNT。

### 32.4.7 Flash页擦除时间控制寄存器（FLSERCNT）

通过FLSERCNT寄存器可以设置FLASH全片擦除的时间。

位	符号	描述	复位值
31	Load	页擦除时间设定选择（注： 0: 使用硬件设定的擦除时间 1: 使用软件设定的擦除时间（FLSERCNT[10:0]）	0
30:11	-	保留	-
10:0	FLSERCNT	软件擦除时间设定： sector擦除时间=（SERCNT*256*Tfclk），需满足>2ms的硬件要求	0x149

注1：在主时钟为内部高速OCO时钟 $\leq 20\text{M}$ 时，可以使用硬件设定时间，不设定FLSERCNT。

### 32.4.8 Flash 写入时间控制寄存器（FLPROCNT）

通过FLPROCNT寄存器可以设置FLASH WORD写入的时间。

位	符号	描述	复位值
31	Load1	写入动作建立时间（Tpgs）设定选择 <sup>注1</sup> ： 0: 使用硬件设定的时间 1: 使用软件设定的时间 FLPGSCNT[12:0]	0
30:29	-	保留	-
28:16	FLPGSCNT	软件写入动作建立时间设定： 写入动作建立时间=（PGSCNT*Tfclk），需满足>70us的硬件要求	0xB0
15	Load0	写入时间（Tprog）设定选择 <sup>注2</sup> ： 0: 使用硬件设定的时间 1: 使用软件设定的时间 FLPROCNT[8:0]	0
14:10	-	保留	-
9:0	FLPROCNT	软件写入时间设定： 写入时间=（PROCNT*Tfclk），需满足>7.5us的硬件要求	0x7E

注1：在主时钟为内部高速OCO时钟 $\leq 20\text{M}$ 时，可以使用硬件设定时间，不设定FLPGSCNT。

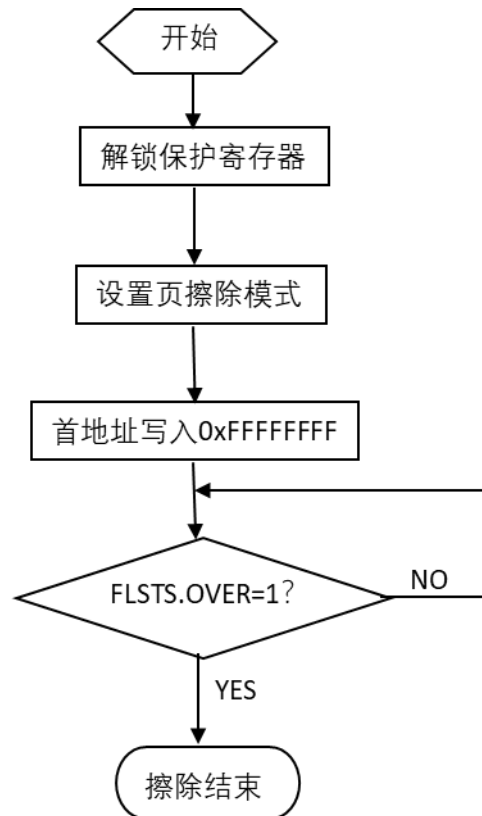
注2：在主时钟为内部高速OCO时钟 $\leq 20\text{M}$ 时，可以使用硬件设定时间，不设定FLPROCNT。

## 32.5 FLASH操作方法

### 32.5.1 页擦除 (sector erase)

sector 擦除，擦除时间由硬件实现，也可以通过 FLSERCNT 配置。操作流程如下：

- (1) 设置 FLERMD.ERMD0 为 1'b0，选择 sector 擦除模式,根据是否需要硬件校验选择设置 ERMD1 的值；
- (2) 设置 FLPROT 为 0xF1，解除 FLOPMD 的保护。然后将 FLOPMD1 设置为 0x55，FLOPMD2 设置为 0xAA，
- (3)向擦除目标 sector 的首地址写入任意数据。例：`*((unsigned long *) 0x00000200) = 0xffffffff。`
- (4)软件查询状态寄存器 FLSTS.OVF,OVF=1 时，表示擦除操作完成。
- (5)如果设置了擦除后进行硬件校验（ERMD1=1），可以通过软件判断 FLSTS.EVF，查询是否校验正确。
- (6)进行下次操作前，软件置"1"来清除 FLSTS。





## 32.5.2 全片擦除 (chip erase)

chip擦除，擦除时间由硬件实现，也可以通过FLCERCNT配置。操作流程如下：

- 1)设置 FLERMD.ERMD0 为 1'b1，选择 chip 擦除模式；
- 2)设置 FLPROT 为 0xF1，解除 FLOPMD 的保护。然后将 FLOPMD1 设置为 0x55，FLOPMD2 设置为 0xAA，
- 3)向代码闪存区域的任意地址写入任意数据。
- 4)软件查询状态寄存器 FLSTS.OVF,OVF=1 时，表示擦除操作完成。
- 5)进行下次操作前，软件置"1"来清除 FLSTS。

## 32.5.3 编程 (word program)

编程，写入时间由硬件实现，也可以通过PROCNT配置。操作流程如下：

- 1)设置 FLPROT 为 0xF1，解除 FLOPMD 的保护。然后将 FLOPMD1 设置为 0xAA，FLOPMD2 设置为 0x55，
- 2)向目标地址写入相应的数据。
- 3)软件查询状态寄存器 FLSTS.OVF,OVF=1 时，表示写入操作完成。
- 4)进行下次操作前，软件置"1"来清除 FLSTS。

## 32.6 FLASH操作的注意事项

1) FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。擦写参数的设置可以由硬件实现，也可通过修改参数寄存器进行软件修改；在使用内部高速 OCO，MAINOSC/外部输入时钟=20M 时，推荐使用硬件设置的擦写参数，无需设置参数寄存器。

2) 如果擦写操作从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待操作完成后，继续下一条指令。如果该操作从 RAM 里执行，CPU 不会停止取指，当前可以继续下一条指令。

3) 在 FLASH 处于编程操作中时，如果 CPU 执行进入深睡眠的指令，系统将等待编程动作结束后才会进入深睡眠。

## 第33章 测试相关

### 33.1 寄存器映射

(TEST基地址 = 0x4006\_9000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
TEST1	0x000	R/W	测试寄存器1	0x0
LOCK	0x00C	R/W	测试寄存器访问使能控制寄存器	0x0

### 33.2 寄存器说明

#### 33.2.1 测试寄存器1(TEST1)

位	符号	描述	复位值
31:3	-	保留	-
2	PGA123_OEN	PGA123输出到PAD 0: 禁止 1: 使能(输出到P53)	0
1:0	-	必须为0	0x0

#### 33.2.2 测试寄存器访问使能控制寄存器

位	符号	描述	复位值
31:8	-	保留	-
7:0	LOCK	当LOCK=0x55时, 使能操作测试寄存器 当LOCK=其他值时, 禁止操作测试寄存器	0x0

## 第34章 附录 修订记录

版本	日期	修订内容
V0.0.1	2023年7月	初始版本
V0.0.2	2023年9月	1) 更正20.5.3章节内容
V0.0.3		2) 修改格式 3) 更正5.2.6 章节定时器模式寄存器中通道运行模式的设定 4) 修改14.3.7 章节连续模式说明 5) 更正14.3.14 内容 6) 删除29.3.2.1 章节的PIORx
V0.1.0	2023年10月	1) 更正5.2.11 章节内容 2) 更正29.3 章节内容 3) 更正30.2 章节内容 4) 更正表5 9 定时器时钟选择寄存器m (TPSm) 的表格 5) 更正5.8.2 作为 PWM 功能的运行内容 6) 更正6.4 12位间隔定时器的控制寄存器的注2 7) 更正7.3.2时钟输出选择寄存器的CKS1的注解 8) 修改18.4.2 控制数据的分配和18.4.3向量表章节的内容 9) 删除18.4.5 DMA触发源选择寄存器的BACK_INTEN 10) 更正18.4.7 触发源信号触发DMA使能寄存器1 11) 更正20.2 特性章节 12) 增加2.4.3章节引脚上拉电阻说明 13) 增加模拟比较器开窗控制和滤波功能描述 14) 修改23.2章节结构框图
V0.1.1	2024年3月	1) 修改19.3.1ADC通道内容 2) 修改19.3.3ADC结构图 3) 修改19.5.2ADC控制寄存器2的AN25通道选择内容 4) 更正图18.2 5) 增加8.3.4看门狗配置寄存器章节 6) 增加PGA123_ O端口说明和PA1/2/3通道选择位 7) 修改22.5.1寄存器说明 8) 更正12.5寄存器说明
V0.5.0	2024年7月	1) 修改第7章, 删除有关PCUBZ1描述 2) 修改20.5.1 PGA0CON0寄存器说明 3) 修改19.5.2 ADC CON2寄存器说明
V0.5.1	2024年11月	删除1.5章节有误内容