

8 位 MCU
ES7P003

数 据 手 册

- 产品简介
- 数据手册
- 产品规格

上海东软载波微电子有限公司

2018 年 5 月 9 日

东软载波 MCU 芯片使用注意事项

关于芯片的上/下电

东软载波 MCU 芯片具有独立电源管脚。当 MCU 芯片应用在多电源供电系统时，应先对 MCU 芯片上电，再对系统其他部件上电；反之，下电时，先对系统其他部件下电，再对 MCU 芯片下电。若操作顺序相反则可能导致芯片内部元件过压或过流，从而导致芯片故障或元件退化。具体可参照芯片的数据手册说明。

关于芯片的复位

东软载波 MCU 芯片具有内部上电复位。对于不同的快速上/下电或慢速上/下电系统，内部上电复位电路可能失效，建议用户使用外部复位、下电复位、看门狗复位等，确保复位电路正常工作。在系统设计时，若使用外部复位电路，建议采用三极管复位电路、RC 复位电路。若不使用外部复位电路，建议采用复位管脚接电阻到电源，或采取必要的电源抖动处理电路或其他保护电路。具体可参照芯片的数据手册说明。

关于芯片的时钟

东软载波 MCU 芯片具有内部和外部时钟源。内部时钟源会随着温度、电压变化而偏移，可能会影响时钟源精度；外部时钟源采用陶瓷、晶体振荡器电路时，建议使能起振延时；使用 RC 振荡电路时，需考虑电容、电阻匹配；采用外部有源晶振或时钟输入时，需考虑输入高/低电平电压。具体可参照芯片的数据手册说明。

关于芯片的初始化

东软载波 MCU 芯片具有各种内部和外部复位。对于不同的应用系统，有必要对芯片寄存器、内存、功能模块等进行初始化，尤其是 I/O 管脚复用功能进行初始化，避免由于芯片上电以后，I/O 管脚状态的不确定情况发生。

关于芯片的管脚

东软载波 MCU 芯片具有宽范围的输入管脚电平，建议用户输入高电平应在 V_{IHMIN} 之上，低电平应在 V_{ILMAX} 之下。避免输入电压介于 V_{IHMIN} 和 V_{ILMAX} 之间，以免波动噪声进入芯片。对于未使用的输入/输出管脚，建议用户设为输入状态，并通过电阻上拉至电源或下拉至地，或设置为输出管脚，输出固定电平并浮空。对未使用的管脚处理因应用系统而异，具体遵循应用系统的相关规定和说明。

关于芯片的 ESD 防护措施

东软载波 MCU 芯片具有满足工业级 ESD 标准保护电路。建议用户根据芯片存储/应用的环境采取适当静电防护措施。应注意应用环境的湿度；建议避免使用容易产生静电的绝缘体；存放和运输应在抗静电容器、抗静电屏蔽袋或导电材料容器中；包括工作台在内的所有测试和测量工具必须保证接地；操作者应该佩戴静电消除手腕环手套，不能用手直接接触芯片等。

关于芯片的 EFT 防护措施

东软载波 MCU 芯片具有满足工业级 EFT 标准的保护电路。当 MCU 芯片应用在 PCB 系统时，需要遵守 PCB 相关设计要求，包括电源、地走线（包括数字/模拟电源分离，单/多点接地等）、复位管脚保护电路、电源和地之间的去耦电容、高低频电路单独分别处理以及单/多层板选择等。

关于芯片的开发环境

东软载波 MCU 芯片具有完整的软/硬件开发环境，并受知识产权保护。选择上海东软载波微电子有限公司或其指定的第三方公司的汇编器、编译器、编程器、硬件仿真器开发环境，必须遵循与芯片相关的规定和说明。

注：在产品开发时，如遇到不清楚的地方，请通过销售或其它方式与上海东软载波微电子有限公司联系。

产品订购信息

Part NO.	工作电压	FLASH	SRAM	Data FLASH	I/O	ADC	UART	I2CS	SPI	Timer	封装类型
ES7P003FGTF	2.3V~5.5V	8K Word	1K Byte	256 Word	18	12-bit×10Ch	1	1	1	8-bit×1 16-bit×2	TSSOP20

注：8K Word FLASH 程序存储器中最后 256 Word 为固定的 Data Flash 存储空间。

地 址：中国上海市龙漕路 299 号天华信息科技园 2A 楼 5 层

邮 编：200235

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：<http://www.essemi.com>

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

修订历史

版本	修改日期	更改概要
Preliminary	2018-5-9	初版，预发布版本

CONFIDENTIAL

目 录

内容目录

第 1 章	芯片简介	17
1. 1	概述	17
1. 2	应用领域	20
1. 3	结构框图	20
1. 4	管脚分配图	21
1. 4. 1	20-pin	21
1. 5	管脚说明	21
第 2 章	内核特性	24
2. 1	CPU 内核概述	24
2. 2	硬件乘法器	24
2. 2. 1	概述	24
2. 2. 2	硬件乘法器操作	24
2. 3	硬件除法器	25
2. 3. 1	概述	25
2. 3. 2	硬件除法器操作	25
2. 4	特殊功能寄存器	25
2. 4. 1	程序状态字寄存器 (PSW)	26
2. 4. 2	程序计数器寄存器低 8 位 (PCRL)	26
2. 4. 3	程序计数器寄存器高 5 位 (PCRH)	27
2. 4. 4	累加器 A 寄存器 (AREG)	27
2. 4. 5	乘数 A 寄存器 (MULA)	27
2. 4. 6	乘数 B 寄存器 (MULB)	27
2. 4. 7	乘积低 8 位寄存器 (MULL)	27
2. 4. 8	乘积高 8 位寄存器 (MULH)	28
2. 4. 9	被除数低 8 位寄存器 (DIVEL)	28
2. 4. 10	被除数高 8 位寄存器 (DIVEH)	28
2. 4. 11	除数寄存器 (DIVS)	28
2. 4. 12	商低 8 位寄存器 (DIVQL)	28
2. 4. 13	商高 8 位寄存器 (DIVQH)	29
2. 4. 14	余数寄存器 (DIVR)	29
第 3 章	存储资源	30
3. 1	概述	30
3. 2	程序/数据寻址空间映射	30
3. 3	程序存储器	32
3. 3. 1	概述	32
3. 3. 2	程序计数器 (PC)	32
3. 3. 3	硬件堆栈	33
3. 3. 4	存储器 IAP 操作	33
3. 3. 4. 1	概述	33
3. 3. 4. 2	查表读操作	33
3. 3. 4. 3	存储器的 IAP 擦写	34

3. 3. 4. 4	存储器页更新流程.....	35
3. 3. 4. 5	操作参考例程.....	35
3. 3. 5	特殊功能寄存器.....	37
3. 3. 5. 1	查表地址寄存器低 8 位 (FRAL)	37
3. 3. 5. 2	查表地址寄存器高 8 位 (FRAH)	37
3. 3. 5. 3	查表地址取反寄存器低 8 位 (FRALN)	38
3. 3. 5. 4	查表地址取反寄存器高 8 位 (FRAHN)	38
3. 3. 5. 5	查表数据寄存器低 8 位 (ROMDL)	38
3. 3. 5. 6	查表数据寄存器高 8 位 (ROMDH)	38
3. 3. 5. 7	存储器控制寄存器 (ROMCL)	38
3. 3. 6	存储器控制寄存器 (ROMCH)	39
3. 4	在线编程 ISP 和在线调试 ICD.....	39
3. 5	数据存储器	40
3. 5. 1	概述.....	40
3. 5. 2	通用数据存储器.....	40
3. 5. 3	特殊功能寄存器.....	40
3. 5. 4	寻址方式	41
3. 5. 4. 1	直接寻址	41
3. 5. 4. 2	GPR 特殊寻址	42
3. 5. 4. 3	间接寻址	42
3. 5. 5	特殊功能寄存器.....	43
3. 5. 5. 1	间接寻址数据寄存器 (IAD)	43
3. 5. 5. 2	间接寻址索引寄存器低 8 位 (IAAL)	43
3. 5. 5. 3	间接寻址索引寄存器高 8 位 (IAAH)	44
3. 5. 5. 4	存储体选择寄存器 (BKSР)	44
第 4 章	输入/输出端口.....	45
4. 1	概述	45
4. 2	结构框图	46
4. 3	I/O 端口功能设置	46
4. 3. 1	I/O 端口输入/输出控制	46
4. 3. 2	I/O 端口弱上/下拉功能	46
4. 3. 3	I/O 端口模拟/数字类型选择功能.....	47
4. 3. 4	I/O 端口开漏输出	47
4. 3. 5	I/O 端口复用功能	47
4. 4	端口中断	47
4. 4. 1	外部端口中断 (PINT)	47
4. 5	I/O 端口操作注意事项.....	48
4. 6	特殊功能寄存器	48
4. 6. 1	PA 端口电平状态寄存器 (PA)	48
4. 6. 2	PA 端口输入输出控制寄存器 (PAT)	48
4. 6. 3	PA 端口弱上拉控制寄存器 (PAPU)	48
4. 6. 4	PA 端口弱下拉控制寄存器 (PAPD)	49
4. 6. 5	PB 端口电平状态寄存器 (PB)	49
4. 6. 6	PB 端口输入输出控制寄存器 (PBT)	49

4. 6. 7	PB 端口弱上拉控制寄存器 (PBPU)	49
4. 6. 8	PB 端口弱上拉控制寄存器 (PBPD)	50
4. 6. 9	PC 端口电平状态寄存器 (PC)	50
4. 6. 10	PC 端口输入输出控制寄存器 (PCT)	50
4. 6. 11	PC 端口弱上拉控制寄存器 (PCPU)	50
4. 6. 12	端口特殊功能控制寄存器 (PORTCTR)	51
第 5 章	特殊功能及操作特性.....	52
5. 1	系统时钟与振荡器	52
5. 1. 1	概述	52
5. 1. 2	内部结构图	52
5. 1. 3	时钟源	53
5. 1. 3. 1	内部高速 16MHz RC 振荡器模式 (INTHRC)	53
5. 1. 3. 2	内部低速 32kHz RC 振荡器模式 (INTLRC)	53
5. 2	看门狗定时器 (WDT)	54
5. 2. 1	概述	54
5. 2. 2	内部结构图	54
5. 2. 3	WDT 定时器	54
5. 2. 4	特殊功能寄存器	55
5. 2. 4. 1	WDT 控制寄存器 (WDTC)	55
5. 3	复位模块	56
5. 3. 1	概述	56
5. 3. 2	上电复位	56
5. 3. 3	掉电复位	56
5. 3. 4	外部 MRSTN 管脚复位	57
5. 3. 5	看门狗定时器溢出复位	58
5. 3. 6	RST 指令软件复位	59
5. 3. 7	特殊功能寄存器	59
5. 3. 7. 1	电源控制寄存器 (PWRC)	59
5. 3. 7. 2	功耗控制寄存器 (PWEN)	60
5. 4	低功耗操作	61
5. 4. 1	MCU 低功耗模式	61
5. 4. 2	低功耗模式配置	61
5. 4. 3	IDLE 唤醒方式配置	62
5. 4. 4	唤醒时序图	62
5. 4. 5	特殊功能寄存器	64
5. 4. 5. 1	唤醒延时控制寄存器 (WKDC)	64
第 6 章	外设	65
6. 1	8 位定时/计数器 (T8N)	65
6. 1. 1	概述	65
6. 1. 2	内部结构图	65
6. 1. 3	预分频器	65
6. 1. 4	工作模式	66
6. 1. 5	定时器模式	66
6. 1. 6	同步计数器模式	67

6.1.7	特殊功能寄存器	67
6.1.7.1	T8N 计数器寄存器 (T8N)	68
6.1.7.2	T8N 控制寄存器 (T8NC)	68
6.2	16 位高级定时/计数器 (ADTIM1)	69
6.2.1	概述	69
6.2.2	定时器时基单元	70
6.2.2.1	ADTIM1 寄存器的写操作	71
6.2.2.2	预分频器	71
6.2.2.3	递增计数模式	72
6.2.2.4	递减计数模式	76
6.2.2.5	中心对齐模式	79
6.2.2.6	重复计数器	83
6.2.3	时钟源	84
6.2.3.1	内部时钟源(CK_INT)	84
6.2.3.2	外部时钟源模式 1	85
6.2.3.3	外部时钟源模式 2	86
6.2.4	捕获/比较通道	87
6.2.4.1	输入捕获模式	88
6.2.4.2	PWM 输入模式	89
6.2.4.3	强制输出模式	90
6.2.4.4	输出比较模式	90
6.2.4.5	PWM 模式	91
6.2.4.6	单脉冲模式	94
6.2.4.7	带死区互补输出	96
6.2.4.8	使用刹车功能	97
6.2.4.9	外部事件清除 OCnREF 信号	99
6.2.4.10	产生六步 PWM 输出	100
6.2.4.11	编码器接口模式	101
6.2.5	定时器和外部触发的同步	103
6.2.5.1	从模式：复位模式	103
6.2.5.2	从模式：门控模式	104
6.2.5.3	从模式：触发模式	105
6.2.5.4	从模式：外部时钟模式 2+触发模式	106
6.2.6	特殊功能寄存器	106
6.2.6.1	控制寄存器 1 低 8 位 (TIM1_CR1L)	106
6.2.6.2	控制寄存器 1 高 8 位 (TIM1_CR1H)	108
6.2.6.3	控制寄存器 2 低 8 位 (TIM1_CR2L)	108
6.2.6.4	控制寄存器 2 高 8 位 (TIM1_CR2H)	109
6.2.6.5	从模式控制寄存器低 8 位 (TIM1_SMCR_L)	110
6.2.6.6	从模式控制寄存器高 8 位 (TIM1_SMCR_H)	111
6.2.6.7	中断使能寄存器低 8 位 (TIM1_IERL)	112
6.2.6.8	中断使能寄存器高 8 位 (TIM1_IERH)	113
6.2.6.9	中断禁止寄存器低 8 位 (TIM1_IDRL)	113
6.2.6.10	中断禁止寄存器高 8 位 (TIM1_IDRH)	114

6.2.6.11	中断有效状态寄存器低 8 位 (TIM1_IVSL)	114
6.2.6.12	中断有效状态寄存器高 8 位 (TIM1_IVSH)	115
6.2.6.13	中断标志寄存器低 8 位 (TIM1_RIFL)	116
6.2.6.14	中断标志寄存器高 8 位 (TIM1_RIFH)	118
6.2.6.15	中断标志屏蔽寄存器低 8 位 (TIM1_IFML)	119
6.2.6.16	中断标志屏蔽寄存器高 8 位 (TIM1_IFMH)	121
6.2.6.17	中断清零寄存器低 8 位 (TIM1_ICRL)	121
6.2.6.18	中断清零寄存器高 8 位 (TIM1_ICRH)	122
6.2.6.19	事件产生寄存器 (TIM1_EGRL)	123
6.2.6.20	捕捉/比较模式寄存器 1 低 8 位 (TIM1_CCMR1L)	124
6.2.6.21	捕捉/比较模式寄存器 1 高 8 位 (TIM1_CCMR1H)	126
6.2.6.22	捕捉/比较模式寄存器 2 低 8 位 (TIM1_CCMR2L)	129
6.2.6.23	捕捉/比较模式寄存器 2 高 8 位 (TIM1_CCMR2H)	131
6.2.6.24	捕捉/比较使能寄存器低 8 位 (TIM1_CCERL)	133
6.2.6.25	捕捉/比较使能寄存器高 8 位 (TIM1_CCERH)	136
6.2.6.26	计数器寄存器低 8 位 (TIM1_CNTL)	139
6.2.6.27	计数器寄存器高 8 位 (TIM1_CNTH)	139
6.2.6.28	预分频寄存器低 8 位 (TIM1_PSCL)	140
6.2.6.29	预分频寄存器高 8 位 (TIM1_PSCH)	140
6.2.6.30	自动重载寄存器低 8 位 (TIM1_ARRL)	140
6.2.6.31	自动重载寄存器高 8 位 (TIM1_ARRH)	140
6.2.6.32	重复计数寄存器 (TIM1_RCR)	141
6.2.6.33	捕捉/比较寄存器 1 低 8 位 (TIM1_CCR1L)	141
6.2.6.34	捕捉/比较寄存器 1 高 8 位 (TIM1_CCR1H)	141
6.2.6.35	捕捉/比较寄存器 2 低 8 位 (TIM1_CCR2L)	142
6.2.6.36	捕捉/比较寄存器 2 高 8 位 (TIM1_CCR2H)	142
6.2.6.37	捕捉/比较寄存器 3 低 8 位 (TIM1_CCR3L)	143
6.2.6.38	捕捉/比较寄存器 3 高 8 位 (TIM1_CCR3H)	143
6.2.6.39	捕捉/比较寄存器 4 低 8 位 (TIM1_CCR4L)	143
6.2.6.40	捕捉/比较寄存器 4 高 8 位 (TIM1_CCR4H)	144
6.2.6.41	刹车和死区时间寄存器低 8 位 (TIM1_BDTRL)	144
6.2.6.42	刹车和死区时间寄存器高 8 位 (TIM1_BDTRH)	145
6.3	16 位通用定时/计数器 (T21)	147
6.3.1	概述	147
6.3.2	内部结构图	148
6.3.3	预分频器和后分频器	148
6.3.4	工作模式	148
6.3.5	定时器模式	148
6.3.6	多精度 PWM 模式	149
6.3.7	捕捉器模式	150
6.3.8	比较器模式	151
6.3.9	特殊功能寄存器	152
6.3.9.1	计数器寄存器低 8 位 (T21L)	152
6.3.9.2	计数器寄存器高 8 位 (T21H)	153

6. 3. 9. 3	周期寄存器低 8 位 (T21PL)	153
6. 3. 9. 4	周期寄存器高 8 位 (T21PH)	153
6. 3. 9. 5	多功能寄存器 0 低 8 位 (T21R0L)	153
6. 3. 9. 6	多功能寄存器 0 高 8 位 (T21R0H)	154
6. 3. 9. 7	多功能寄存器 1 低 8 位 (T21R1L)	154
6. 3. 9. 8	多功能寄存器 1 高 8 位 (T21R1H)	154
6. 3. 9. 9	多功能寄存器 2 低 8 位 (T21R2L)	154
6. 3. 9. 10	多功能寄存器 2 高 8 位 (T21R2H)	155
6. 3. 9. 11	控制寄存器低 8 位 (T21CL)	155
6. 3. 9. 12	控制寄存器次高 8 位 (T21CM)	156
6. 3. 9. 13	控制寄存器高 8 位 (T21CH)	156
6. 3. 9. 14	输入输出控制寄存器 (T21OC)	157
6. 4	通用异步接收/发送器 (UART)	158
6. 4. 1	概述.....	158
6. 4. 2	内部结构图.....	158
6. 4. 3	波特率配置.....	159
6. 4. 4	传输数据格式.....	159
6. 4. 5	异步发送器.....	159
6. 4. 6	异步接收器.....	160
6. 4. 7	UART 使用注意事项	161
6. 4. 8	特殊功能寄存器.....	162
6. 4. 8. 1	UART 接收数据寄存器 (RXB)	162
6. 4. 8. 2	UART 接收控制/状态寄存器 (RXC)	162
6. 4. 8. 3	UART 发送数据寄存器 (TXB)	163
6. 4. 8. 4	UART 发送控制/状态寄存器 (TXC)	163
6. 4. 8. 5	UART 波特率寄存器 (BRR)	163
6. 5	I2C 总线从动器 (I2CS)	164
6. 5. 1	概述.....	164
6. 5. 2	内部结构	165
6. 5. 3	I2CS 端口配置	165
6. 5. 4	I2C 通讯协议	166
6. 5. 5	I2C 操作	167
6. 5. 6	起始位 START 和停止位 STOP	167
6. 5. 7	数据传输和应答.....	168
6. 5. 8	数据传输格式参考	169
6. 5. 9	I2C 通讯发送器	169
6. 5. 10	I2C 通讯接收器	170
6. 5. 11	I2C 时钟线自动下拉等待请求功能	170
6. 5. 12	特殊功能寄存器.....	171
6. 5. 12. 1	I2C 采样滤波寄存器 (I2CX16)	171
6. 5. 12. 2	I2C 控制寄存器 (I2CC)	171
6. 5. 12. 3	I2C 从机地址寄存器 (I2CSA)	172
6. 5. 12. 4	I2C 发送数据缓冲器 (I2CTB)	172
6. 5. 12. 5	I2C 接收数据缓冲器 (I2CRB)	173

6. 5. 12. 6	I2C 中断使能寄存器 (I2CIEC)	173
6. 5. 12. 7	I2C 中断标志寄存器 (I2CIFC)	174
6. 6	SPI 同步串口通讯模块 (SPI)	175
6. 6. 1	概述.....	175
6. 6. 2	结构框图	175
6. 6. 3	SPI 通讯模式.....	175
6. 6. 4	SPI 数据格式.....	175
6. 6. 5	SPI 同步发送器	177
6. 6. 6	SPI 同步接收器	177
6. 6. 7	SPI 通讯控制.....	178
6. 6. 8	SPI 延迟接收功能	179
6. 6. 9	SPI 应用说明.....	179
6. 6. 10	特殊功能寄存器.....	179
6. 6. 10. 1	SPI 控制寄存器 0 (SPICON0)	179
6. 6. 10. 2	SPI 控制寄存器 1 (SPICON1)	180
6. 6. 10. 3	SPI 发送数据写入寄存器 (SPITBW)	181
6. 6. 10. 4	SPI 接收数据读取寄存器 (SPIRBR)	181
6. 6. 10. 5	SPI 中断使能寄存器 (SPIIE)	181
6. 6. 10. 6	SPI 中断标志寄存器 (SPIIF)	182
6. 7	模/数转换器 (ADC)	184
6. 7. 1	概述.....	184
6. 7. 2	内部结构图.....	184
6. 7. 3	ADC 配置	185
6. 7. 4	ADC 转换步骤.....	185
6. 7. 5	AD 时序特征示意图.....	186
6. 7. 6	参考例程	187
6. 7. 7	特殊功能寄存器.....	187
6. 7. 7. 1	ADC 转换值寄存器 (ADCR)	188
6. 7. 7. 2	ADC 控制寄存器 (ADCCL)	188
6. 7. 7. 3	ADC 控制寄存器 (ADCCH)	189
6. 7. 7. 4	端口数模控制寄存器 (ANSL)	190
第 7 章	中断处理.....	191
7. 1	概述	191
7. 2	中断控制结构框图	191
7. 3	中断模式选择.....	192
7. 3. 1	默认中断模式	192
7. 3. 2	向量中断模式	193
7. 3. 2. 1	向量表配置.....	193
7. 3. 2. 2	中断分组配置	194
7. 3. 2. 3	中断使能配置	194
7. 4	中断现场保护.....	196
7. 5	中断操作	196
7. 5. 1	外部中断	196
7. 5. 2	ADC 中断	197

7.5.3	T8N 溢出中断.....	197
7.5.4	T2x(T21)匹配中断.....	197
7.5.5	T2x(T21)周期中断.....	197
7.5.6	UART 中断	198
7.5.7	I2CS 中断.....	198
7.5.8	SPI 中断	198
7.5.9	ADTIM1 中断	198
7.5.10	中断操作注意事项.....	198
7.6	特殊功能寄存器	199
7.6.1	中断全局寄存器 (INTG)	199
7.6.2	中断优先级寄存器 (INTP)	199
7.6.3	中断控制寄存器 0 (INTC0)	200
7.6.4	中断标志寄存器 0 (INTF0)	200
7.6.5	中断使能寄存器 0 (INTE0)	201
7.6.6	中断标志寄存器 1 (INTF1)	201
7.6.7	中断使能寄存器 1 (INTE1)	202
7.6.8	中断标志寄存器 2 (INTF2)	203
7.6.9	中断使能寄存器 2 (INTE2)	203
第 8 章	芯片配置字	205
第 9 章	芯片封装图	207
9.1	20-pin 封装图.....	207
附录 1	指令集.....	208
附录 1.1	概述	208
附录 1.2	寄存器操作指令	208
附录 1.3	程序控制指令	208
附录 1.4	算术/逻辑运算指令	209
附录 2	特殊功能寄存器总表.....	212
附录 3	电气特性	218
附录 3.1	参数特性表	218
附录 3.2	参数特性图	222

图目录

图 1-1 ES7P003 结构框图	20
图 1-2 ES7P003FGTF 顶视图	21
图 3-1 FLASH 程序存储区寻址空间映射图.....	31
图 3-2 堆栈示意图.....	33
图 3-3 页更新参考流程图.....	35
图 3-4 GPR 地址映射示意图.....	40
图 3-5 特殊功能寄存器空间.....	41
图 3-6 直接寻址示意图.....	42
图 3-7 GPR 特殊寻址示意图.....	42
图 3-8 间接寻址示意图.....	43
图 4-1 PA/PB/PC 端口结构图	46
图 5-1 系统时钟结构图.....	52
图 5-2 看门狗定时器内部结构图	54
图 5-3 芯片复位原理图.....	56
图 5-4 上电复位时序示意图	56
图 5-5 低电压复位时序示意图	57
图 5-6 外部 MRSTN 管脚复位	57
图 5-7 MRSTN 复位参考电路图 1	58
图 5-8 MRSTN 复位参考电路图 2	58
图 5-9 看门狗溢出复位.....	58
图 5-10 RST 指令软件复位	59
图 5-11 系统时钟为 INTHRC 时，系统唤醒 IDLE0 的时序图	63
图 5-12 系统时钟为 INTHRC 时，系统唤醒 IDLE1 的时序图	63
图 6-1 T8N 内部结构图	65
图 6-2 定时器模式时序图	67
图 6-3 计数器模式时序图 (T8NEG=0, T8NCKI 上升沿计数)	67
图 6-4 高级定时器内部结构图	70
图 6-5 预分频器分频比由 1 变更为 2 的计数器时序图	72
图 6-6 预分频器分频比由 1 变更为 4 的计数器时序图	72
图 6-7 计数器时序图 (内部时钟除以 1)	73
图 6-8 计数器时序图 (内部时钟除以 2)	74
图 6-9 计数器时序图 (内部时钟除以 4)	74
图 6-10 计数器时序图 (内部时钟除以 N)	75
图 6-11 计数器时序图 (当 ARPE=0 更新事件(TIMx_ARR 无预载))	75
图 6-12 计数器时序图 (当 ARPE=1 更新事件(TIMx_ARR 预载))	76
图 6-13 计数器时序图 (内部时钟分频因子为 1)	77
图 6-14 计数器时序图 (内部时钟分频因子为 2)	77
图 6-15 计数器时序图 (内部时钟分频因子为 4)	78
图 6-16 计数器时序图 (内部时钟分频因子 N)	78
图 6-17 计数器时序图 (重复计数器未使用时的更新事件)	79
图 6-18 计数器时序图, 内部时钟分频因子为 1,TIMx_ARR=0x05	80
图 6-19 计数器时序图, 内部时钟分频因子为 2	80
图 6-20 计数器时序图, 内部时钟分频因子为 4,TIMx_ARR=0x36	81

图 6-21 计数器时序图, 内部时钟分频因子为 N	81
图 6-22 计数器时序图, ARPE=1 时的更新事件(计数器下溢)	82
图 6-23 计数器时序图, ARPE=1 时的更新事件(计数器上溢)	82
图 6-24 以模式与 TIMx_RCR 寄存器设置为依据的更新比率范例	84
图 6-25 一般模式下的控制电路, 内部时钟分频因子为 1	85
图 6-26 TI1 外部时钟连接范例	85
图 6-27 外部时钟模式 1 的控制电路	86
图 6-28 ETR 给予一外部触发输入区块概述	86
图 6-29 外部时钟模式 2 的控制电路	87
图 6-30 捕获/比较通道(例如, 通道 1 输入阶段)	87
图 6-31 捕获/比较通道的输出阶段(通道 1 至 3)	88
图 6-32 捕捉/比较通道的输出阶段(通道 4)	88
图 6-33 PWM 输入模式时序	90
图 6-34 输出比较模式, 触发 OC1	91
图 6-35 边沿对齐 PWM 波形, 向上计数(ARR=8)	92
图 6-36 边沿对齐 PWM 波形, 向下计数(ARR=8)	93
图 6-37 中央对齐 PWM 波形(ARR=8)	94
图 6-38 单脉冲模式范例	95
图 6-39 互补输出含死区插入	96
图 6-40 死区波形含大于负脉冲的延迟	97
图 6-41 死区波形大于正脉冲的延迟	97
图 6-42 响应刹车的输出行为	99
图 6-43 清除 TIMx 的 OCnREF	100
图 6-44 六步 PWM 输出产生, 使用 COM 范例 (OSSR=1)	101
图 6-45 编码器模式下的计数器操作实例	103
图 6-46 IC1FP1 反相的编码器接口模式实例(CC1P='1', 其他配置与上例相同)	103
图 6-47 复位模式下的控制电路	104
图 6-48 门控模式下的控制电路	105
图 6-49 触发模式下的控制电路	105
图 6-50 外部时钟模式 2+触发模式下的控制电路	106
图 6-51 T2n 内部结构图	148
图 6-52 T2n 定时器模式时序图	149
图 6-53 T2n 多精度 PWM 模式示意图	150
图 6-54 T2n 捕捉器模式时序图 (每个脉冲上升沿捕捉信号)	151
图 6-55 T2n 比较器模式时序图	152
图 6-56 UART 发送端原理图	158
图 6-57 UART 接收端原理图	159
图 6-58 UART 数据格式示意图	159
图 6-59 UART 异步发送器操作流程图	160
图 6-60 UART 发送器发送数据时序图 (9 位数据格式, 第 9 位数据为“0”)	160
图 6-61 UART 异步接收器操作流程图	161
图 6-62 UART 接收器接收数据时序图 (9 位数据格式)	161
图 6-63 I2C 内部结构	165
图 6-64 I2C 总线通讯协议示意图	167

图 6-65 I2C 从动波形图	167
图 6-66 I2C 起始位和停止位	168
图 6-67 数据传输和应答	168
图 6-68 主控器写入从动器数据示意图	169
图 6-69 主控器读取从动器数据示意图	169
图 6-70 I2C 时钟线下拉等待波形示意图	170
图 6-71 SPI 电路结构框图	175
图 6-72 SPI 时钟上升沿发送，下降沿接收波形示意图	176
图 6-73 SPI 时钟下降沿发送，上升沿接收波形示意图	176
图 6-74 SPI 时钟上升沿接收，下降沿发送波形示意图	176
图 6-75 SPI 时钟下降沿接收，上升沿发送波形示意图	177
图 6-76 SPI 发送数据流示意图	177
图 6-77 SPI 接收数据流示意图	178
图 6-78 SPI 延迟接收功能波形示意图	179
图 6-79 ADC 内部结构图	184
图 6-80 ADC 时序特征示意图 (SMPS=0)	186
图 6-81 ADC 时序特征示意图 (SMPS=1)	187
图 7-1 默认中断模式中断控制逻辑	191
图 7-2 向量中断模式中断控制逻辑	192

表目录

表 1-1 管脚说明.....	23
表 3-1 在线编程/调试管脚说明.....	39
表 4-1 I/O 端口弱上拉	46
表 4-2 I/O 端口弱下拉	47
表 4-3 I/O 端口开漏输出	47
表 4-4 外部端口中断	47
表 5-1 低功耗模式配置表.....	61
表 5-2 唤醒方式配置表.....	62
表 5-3 唤醒时间计算表.....	62
表 6-1 T8N 预分频器配置表.....	66
表 6-2 T8N 工作模式配置表.....	66
表 6-3 计数方向与编码器信号的关系.....	102
表 6-4 带有刹车功能的互补 OC _n 和 OC _{nN} 的输出控制位.....	139
表 6-5 UART 波特率配置表	159
表 6-6 UART 数据格式配置表	159
表 6-7 I2CS 端口配置表	165
表 7-1 中断模式选择表.....	192
表 7-2 默认中断模式使能配置表	193
表 7-3 向量表配置表	194
表 7-4 向量中断模式中断分组配置表	194
表 7-5 向量中断模式使能配置表	195

第1章 芯片简介

1.1 概述

- ◆ 工作条件
 - ◇ 工作电压范围: 2.3V ~ 5.5V
 - ◇ 工作温度范围: -40 ~ 85°C
- ◆ 设计工艺及封装
 - ◇ 低功耗、高速 FLASH CMOS 工艺
 - ◇ 20 个管脚, 采用 TSSOP 封装
- ◆ 内核
 - ◇ ES7P RISC CPU 内核
 - ◇ 79 条精简指令
 - ◇ 系统时钟工作频率最高为 16MHz
 - ◇ 指令周期为 2 个系统时钟周期
 - ◇ 复位向量位于 0000_H, 默认中断向量位于 0004_H
 - ◇ 支持中断处理, 支持中断优先级和中断向量表
 - ◇ 支持硬件乘法/除法器
- ◆ 存储资源
 - ◇ 8K Word FLASH 程序存储器 (其中最后 1 页固定为 Data FLASH 存储器)
 - 共 32 页, 每页 256 Word
 - 支持 IAP 操作, 查表读, 页擦除和单地址编程
 - 擦写时, 支持定时器模块正常工作
 - 擦写时, 不支持中断处理
 - ◇ 256 Word Data FLASH 存储器
 - 共一页
 - 支持 IAP 操作, 查表读, 页擦除和单地址编程
 - 擦写时, 支持定时器模块正常工作
 - 擦写时, 不支持中断处理
 - ◇ 8 级程序堆栈
 - ◇ 1K Byte SRAM 数据存储器
 - ◇ 程序存储器支持直接寻址、相对寻址和查表读操作
 - ◇ 数据存储器支持直接寻址、GPR 特殊寻址和间接寻址
- ◆ 编程及调试接口
 - ◇ 支持在线编程 (ISP) 接口
 - ◇ 支持在线调试 (ICD) 功能
 - ◇ 支持编程代码加密保护
- ◆ I/O 端口

- ◆ 最多支持 18 个 I/O
 - PA 端口 (PA0~PA7)
 - PB 端口 (PB0~PB7)
 - PC 端口 (PC0~PC1)
- ◆ 支持 9 个外部端口中断 PINT (PINT0~PINT8 为输入端)
- ◆ 支持独立的可配置内部弱上/下拉输入端口
 - 输入端口上/下拉电阻的匹配精度为±3%以内 (常温 25°C, VDD=5V)
 - 支持 18 个独立可配置弱上拉输入端口
 - 支持 16 个独立可配置弱下拉输入端口
- ◆ 支持 2 个独立的可配置开漏输出端口
- ◆ 复位及时钟
 - ◇ 内嵌上电复位电路 POR
 - ◇ 内嵌掉电复位电路 BOR
 - BOR 复位电压档位: 2.1V, 2.5V, 3.1V
 - ◇ 支持外部复位 MRSTN
 - ◇ 支持独立硬件看门狗定时器
 - ◇ 支持指令 RST 复位
 - ◇ 支持内部高频 16MHz RC 振荡时钟源
 - 出厂前校准精度为±1% (常温 25°C)
- ◆ 功耗特性
 - ◇ IDLE0 电流
 - 5uA@5.0V, 25°C, LDO 禁止, 系统时钟源停振, 典型值
 - ◇ IDLE1 电流
 - 40uA@5.0V, 25°C, LDO 使能, 系统时钟源停振, 典型值
 - ◇ IDLE2 电流
 - 150uA@5.0V, 25°C, LDO 使能, 系统时钟源不停振, 典型值
 - ◇ 动态电流
 - 2mA@内部 16MHz, 5.0V, 25°C, 典型值
- ◆ 外设
 - ◇ 1 路 8 位定时器 T8N
 - 定时器模式 (计数时钟为系统时钟 2 分频)
 - 计数器模式 (外灌时钟或者内部低频 WDT_RC 时钟)
 - 支持可配置预分频器
 - 支持中断产生
 - ◇ 1 路 16 位高级定时/计数器 ADTIM1
 - 16 位递增、递减、递增/递减自动重载计数器
 - 16 位可编程预分频器, 用于对计数器时钟频率进行分频, 分频系数介于 1~65536 之间任意数值
 - 多达 4 个独立通道, 可用于:

- ✓ 输入捕获
 - ✓ 输出比较
 - ✓ PWM 生成（边沿和中心对齐模式）
 - ✓ 六步 PWM 输出
 - ✓ 单脉冲模式输出
 - ✓ 三组带死区互补 PWM 输出，死区时间可编程设置
 - 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器
 - 刹车功能，可将定时器的输出信号置于复位状态或已知状态
 - 以下事件将生成中断请求：
 - ✓ 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
 - ✓ 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
 - ✓ 输入捕获
 - ✓ 输出比较
 - ✓ 刹车输入
 - 支持多功能中断
- ◇ 1 路 16 位通用定时/计数器 T21
- 支持同步定时器模式（采用系统时钟分频作为时钟源）
 - 内置 4 位预分频器和 7 位的后分频器
 - 多达 3 个独立通道，可用于：
 - ✓ 输入捕获
 - ✓ 输出比较
 - ✓ PWM 生成（边沿模式）
 - 3 路 PWM 输出占空比可独立设置
 - 3 路 PWM 输出极性可独立设置
 - 支持计数器溢出中断、PWM 周期匹配中断、捕捉中断、比较中断
- ◇ 一路高速异步收发器 UART
- 支持异步全双工收发
 - 支持 8 位/9 位数据格式
 - 约定数据从最低位开始接收/发送
 - 支持中断产生
- ◇ 一路 I2C 总线
- 只支持从动模式
 - 支持标准 I2C 总线协议，最高传输速率 400Kbit/s
 - 支持 7 位寻址方式
 - 约定数据从最高位开始接收/发送
 - 支持中断产生
- ◇ 一路同步串口通讯模块 SPI
- 支持主控模式、从动模式
 - 支持 4 种数据传输格式
 - 支持主控模式通讯时钟速率可配置
 - 支持 4 级发送缓冲器和 4 级接收缓冲器
 - 支持发送和接收缓冲器空/满中断
 - 支持接收数据溢出中断、发送数据写错误中断、从动模式的发送数据错误中断

- 支持从动模式的片选变化中断、主控模式的空闲状态中断
- 支持主控模式延迟接收
- ◇ 模拟数字转换器 ADC
 - 支持 12 位数字转换精度
 - 支持 8+2 通道模拟输入端
 - 支持可选择参考源
 - 支持中断产生

1.2 应用领域

本芯片可用于门禁系统/警报器、温度传感设备、蓝牙音箱、电动车表头、数字电压表头、气体检测器、采集器、充电器、美容仪器、小家电等领域。

1.3 结构框图

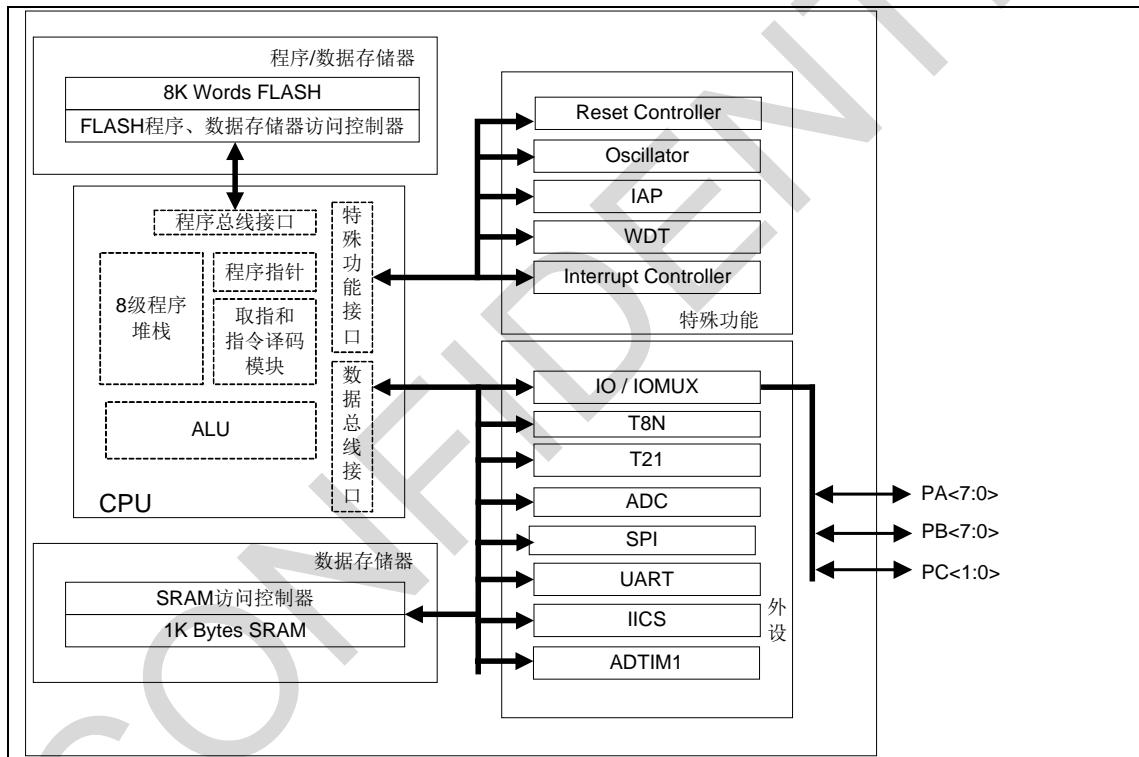


图 1-1 ES7P003 结构框图

1.4 管脚分配图

1.4.1 20-pin

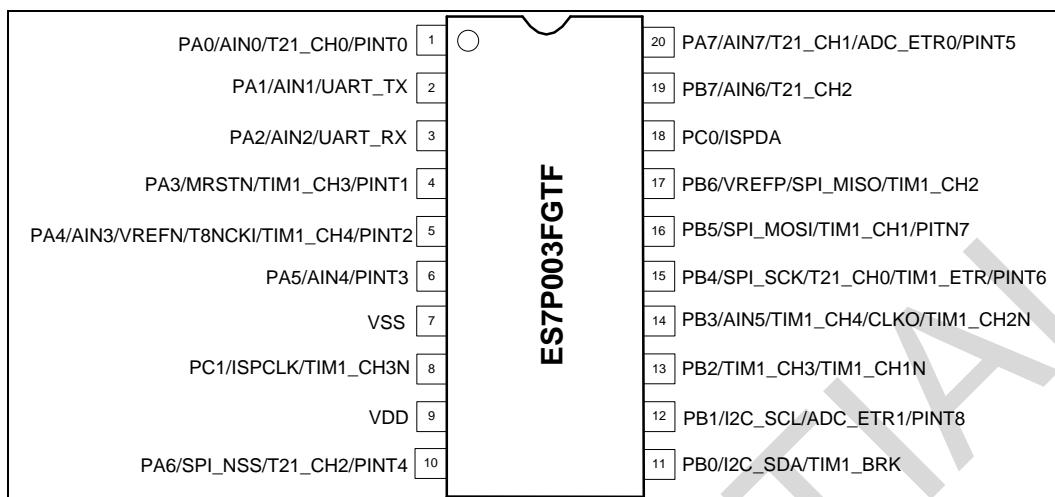


图 1-2 ES7P003FGTF 顶视图

- 注 1: MRSTN 表示低电平有效;
- 注 2: 外部复位管脚可以复用 PA3 为数字输入/输出功能;
- 注 3: PC0 和 PC1 作为一组在线编程/调试接口;
- 注 4: 芯片 I/O 端口输入电平不能高于芯片 VDD+0.3V 且不能低于 VSS-0.3V, 否则可能会影响芯片正常工作。

1.5 管脚说明

管脚名	管脚复用	输入类型	输出类型	A/D	端口说明	备注
PA0/AIN0/T21_CH0/ PINT0	PA0	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	AIN0	—	—	A	ADC 模拟通道 0 输入	
	T21_CH0	TTL	CMOS	D	T21 捕捉输入通道 0 PWM 输出通道 0 比较输出通道 0	
	PINT0	TTL	—	D	外部中断输入 0	
PA1/AIN1/UART_TX	PA1	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	AIN1	—	—	A	ADC 模拟通道 1 输入	
	UART_TX	—	CMOS	D	UART 发送输出	
PA2/AIN2/UART_RX	PA2	TTL	—	D	通用 I	支持弱上/下拉
	AIN2	—	—	A	ADC 模拟通道 2 输入	
	UART_RX	TTL	—	D	UART 接收输入	
PA3/MRSTN/ TIM1_CH3/PINT1	PA3	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	MRSTN	TTL	—	D	外部复位输入	
	TIM1_CH3	TTL	—	D	TIM1 模块通道 3 输入输出	
	PINT1	TTL	—	D	外部中断输入 1	
PA4/AIN3/VREFN/	PA4	TTL	CMOS	D	通用 I/O	支持弱上/下

管脚名	管脚复用	输入类型	输出类型	A/D	端口说明	备注
T8NCKI/ TIM4_CH4/ PINT2	AIN3	—	—	A	ADC 模拟通道 3 输入	拉
	VREFN	—	—	A	ADC 外部参考负输入	
	T8NCKI	TTL	—	A	T8N 外部时钟输入	
	TIM1_CH4	TTL	—	D	TIM1 模块通道 4 输入输出	
	PINT2	TTL	—	D	外部中断输入 2	
PA5/AIN4/PINT3	PA5	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	AIN4	—	—	A	ADC 模拟通道 4 输入	
	PINT3	TTL	—	D	外部中断输入 3	
PA6/SPI_NSS/ T21_CH2/PINT4	PA6	TTL	CMOS	D	通用 I/O	支持弱上、下拉
	SPI_NSS	TTL	—	D	SPI 从动模式片选输入	
	T21_CH2	TTL	CMOS	D	T21 捕捉输入通道 2 PWM 输出通道 2 比较输出通道 2	
	PINT4	TTL	—	D	外部中断输入 4	
PA7/AIN7/T21_CH1/ ADC_ETR0/PINT5	PB7	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	AIN7	—	—	A	ADC 模拟通道 7 输入	
	T21_CH1	TTL	CMOS	D	T21 捕捉输入通道 1 PWM 输出通道 1 比较输出通道 1	
	ADC_ETR0	TTL	—	D	外部中断 5 触发 AD 转换信号	
	PINT5	TTL	—	D	外部中断输入 5	
PB0/I2C_SDA/ TIM1_BRK	PB0	TTL	CMOS	D	通用 I/O	支持弱上/下拉/开漏输出
	I2C_SDA	TTL	CMOS	D	I2C 数据输入/输出	
	TIM1_BRK	TTL	—	D	TIM1 模块 PWM 关断事件输入，刹车输入	
PB1/I2C_SCL/ ADC_ETR1/PINT8	PB1	TTL	CMOS	D	通用 I/O	支持弱上/下拉/开漏输出
	I2C_SCL	TTL	—	D	I2C 时钟输入	
	ADC_ETR1	TTL	—	D	外部中断 8 触发 AD 转换信号	
	PINT8	TTL	—	D	外部中断输入 8	
PB2/TIM1_CH3/ TIM1_CH1N	PB2	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	TIM1_CH3	TTL	CMOS	D	TIM1 输入输出通道 3	
	TIM1_CH1N	TTL	CMOS	D	TIM1 输入，互补输出通道 1	
PB3/AIN5/ TIM1_CH4/ TIM1_CH2N/ CLKO	PB3	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	AIN5	—	—	A	ADC 模拟通道 5 输入	
	TIM1_CH4	TTL	CMOS	D	TIM1 输入输出通道 4	
	TIM1_CH2N	TTL	CMOS	D	TIM1 输入，互补输出通道 2	
	CLKO	—	CMOS	D	系统时钟 16 分频输出	

管脚名	管脚复用	输入类型	输出类型	A/D	端口说明	备注
PB4/SPI_SCK/ T21_CH0/ TIM1_ETR/ PINT6	PB4	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	SPI_SCK	TTL	—	D	SPI 从动模式时钟输入	
	T21_CH0	TTL	CMOS	D	T21 捕捉输入通道 0 PWM 输出通道 0 比较输出通道 0	
	TIM1_ETR	TTL	CMOS	D	TIM1 外部触发输入信号	
	PINT6	TTL	—	D	外部中断输入 6	
PB5/SPI_MOSI/ TIM1_CH1/PINT7	PB5	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	SPI_MOSI	TTL	CMOS	D	SPI 主控输出/从机输入端口	
	TIM1_CH1	TTL	CMOS	D	TIM1 输入输出通道 1	
	PINT7	TTL	—	D	外部中断输入 7	
PB6/VREFP/ SPI_MISO/ TIM1_CH2	PB6	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	VREFP	—	—	A	ADC 外部参考电压正输入	
	SPI_MISO	TTL	CMOS	D	SPI 主控输入/从机输出端口	
	TIM1_CH2	TTL	CMOS	D	TIM1 输入输出通道 2	
PB7/AIN6/T21_CH2	PB7	TTL	CMOS	D	通用 I/O	支持弱上/下拉
	AIN6	—	—	A	ADC 模拟通道 6 输入	
	T21_CH2	TTL	CMOS	D	T21 捕捉输入通道 2 PWM 输出通道 2 比较输出通道 2	
PC0/ISPDAT	PC0	TTL	CMOS	D	通用 I/O	支持弱上拉
	ISPDAT	TTL	CMOS	D	ISP 串行编程/调试数据输入输出	
PC1/ISPCLK	PC1	TTL	CMOS	D	通用 I/O	支持弱上拉
	ISPCK	TTL	—	D	ISP 串行编程/调试时钟输入	
VDD	VDD	—	—	P	电源	—
VSS	VSS	—	—	P	地, 0V 参考点	—

表 1-1 管脚说明

注 1: A = 模拟端口, D = 数字端口, P = 电源/地。

注 2: MRSTN 表示低电平复位有效。

注 3: 所有通用 I/O 端口均为 TTL 施密特输入和 CMOS 输出驱动。

第2章 内核特性

2.1 CPU内核概述

- ◇ 内核特性
- 采用高性能 ES7P RISC CPU 内核，79 条精简指令集
 - 采用 2T 架构，每个机器周期包括两个系统时钟周期
 - 系统时钟最高支持 20MHz，最小指令周期 100ns
 - 支持中断优先级和中断向量表
 - 支持硬件乘法器和除法器

2.2 硬件乘法器

2.2.1 概述

芯片指令集不包含乘法指令，内部集成独立的硬件乘法器，通过读写相应寄存器进行操作。

- ◇ 主要功能组件
- 8 位乘数 A 寄存器（MULA，只可写）
 - 8 位乘数 B 寄存器（MULB，只可写）
 - 16 位乘积寄存器（MULL/MULH，只可读）

2.2.2 硬件乘法器操作

硬件乘法器完成 8 位乘数 MULA 与 8 位乘数 MULB 的相乘操作： $<8 \text{ 位乘数 } A> \times <8 \text{ 位乘数 } B> = 16 \text{ 位乘积}$ 。

在 MULA 和 MULB 写入完成后的 1 个机器周期内，将 16 位结果高、低 8 位分别存储于 2 个寄存器 MULH 和 MULL。

MULA 和 MULL 共用一个寄存器地址，MULB 和 MULH 共用一个寄存器地址。乘数 A/B 设置完成后，下一条指令即可读取乘积结果。

应用实例：硬件乘法器操作应用程序

```
....  
MOVI    mul_operand_a  
MOVA    MULA          ; 写乘数 A  
MOVI    mul_operand_b  
MOVA    MULB          ; 写乘数 B  
MOV     MULL,0         ; 读乘积低 8 位  
....  
MOV     MULH,0         ; 读乘积高 8 位  
....
```

2.3 硬件除法器

2.3.1 概述

芯片指令集不包含除法指令，内部集成独立的硬件除法器，通过读写相应寄存器进行操作。

◇ 主要功能组件

- 16 位被除数寄存器 (DIVEL/DIVEH, 只可写)
- 8 位除数寄存器 (DIVS, 只可写)
- 16 位商寄存器 (DIVQL/DIVQH, 只可读)
- 8 位余数寄存器 (DIVR, 只可读)

2.3.2 硬件除法器操作

硬件除法器完成 16 位被除数 DIVEL、DIVEH 与 8 位除数 DIVS 的除法操作： $<16 \text{ 位被除数} > \div <8 \text{ 位除数} > = 16 \text{ 位商} \dots\dots 8 \text{ 位余数}$ 。

DIVEL 和 DIVQL 共用一个寄存器地址，DIVEH 和 DIVQH 共用一个寄存器地址，DIVS 和 DIVR 共用一个寄存器地址。被除数和除数设置完成后，需要插入 2 条 NOP 指令，才能读取商和余数。若除数为“0”，则商为 0xFFFF，余数为 0xFF，表示溢出。

应用实例：硬件除法器操作应用程序

```
....  
MOVI    div_operand_divel  
MOVA    DIVEL           ; 写被除数低 8 位  
MOVI    div_operand_diveh  
MOVA    DIVEH           ; 写被除数高 8 位  
MOVI    mul_operand_divs  
MOVA    DIVS            ; 写除数  
NOP  
NOP  
MOV    DIVQL,0          ; 读商低 8 位  
....  
MOV    DIVQH,0          ; 读商高 8 位  
....  
MOV    DIVR,0           ; 读余数  
....
```

2.4 特殊功能寄存器

CPU 相关寄存器包括 13-bit 程序计数器寄存器 PCRL/PCRH，程序状态字寄存器 PSW，累加器 A 寄存器 AREG，乘数寄存器 MULA、MULB 和乘积寄存器 MULL、MULH，被除数寄存器 DIVEL/DIVEH、除数寄存器 DIVS、商寄存器 DIVQL/DIVQH 和余数寄存器 DIVR。其中程序状态寄存器 PSW 用于存放各个状态标志位，包括程序出栈溢出、负数标志位、溢出标志位、零标志位、半进位/半借位标志位，以及全进位或全借位标志位等。

2.4.1 程序状态字寄存器 (PSW)

PSW: 程序状态字寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	UF	OF	N	OV	Z	DC	C
R/W	—	R	R	R/W	R/W	R/W	R/W	R/W
RESET	x	0	0	x	x	x	x	x

- Bit 7 未使用
- Bit 6 UF: 程序出栈溢出标志位
0: 程序出栈未溢出
1: 程序出栈溢出
- Bit 5 OF: 程序压栈溢出标志位
0: 程序压栈未溢出
1: 程序压栈溢出
- Bit 4 N: 负数标志位
0: 有符号算术或逻辑运算结果为正数
1: 结果为负数
- Bit 3 OV: 溢出标志位
0: 有符号算术运算未发生溢出
1: 发生溢出
- Bit 2 Z: 零标志位
0: 算术或逻辑运算的结果不为零
1: 算术或逻辑运算的结果为零
- Bit 1 DC: 半进位或半借位标志位
0: 低四位无进位或低四位有借位
1: 低四位有进位或低四位无借位
- Bit 0 C: 全进位或全借位标志位
0: 无进位或有借位
1: 有进位或无借位

注 1: 仅部分指令可对 PSW 寄存器进行写操作, 包括 JDEC、JINC、SWAP、BCC、BSS、BTT、MOVA 和 SETR。
其它指令对 PSW 寄存器的写操作, 只根据指令的运行结果影响相应状态标志位。

注 2: OF 和 UF 位为只读标志位, 仅上电复位、复位指令和 MRSTN 复位会将其清零, 其它复位不影响这两个标志位。

2.4.2 程序计数器寄存器低 8 位 (PCRL)

PCRL: 程序计数器寄存器低 8 位								
Bit	7	6	5	4	3	2	1	0
Name	PCRL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 PCRL<7:0>: 程序计数器寄存器低 8 位

2.4.3 程序计数器寄存器高 5 位 (PCRH)

PCRH: 程序计数器寄存器高 5 位								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	PCRH<4:0>				
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5 未使用

Bit 4~0 PCRH<4:0>: 程序计数器寄存器高 5 位

2.4.4 累加器A寄存器 (AREG)

AREG: 累加器 A 寄存器								
Bit	7	6	5	4	3	2	1	0
Name	AREG<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

Bit 7~0 AREG<7:0>: 累加器的值

2.4.5 乘数A寄存器 (MULA)

MULA: 乘数 A 寄存器								
Bit	7	6	5	4	3	2	1	0
Name	MULA<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	x	x	x	x	x	x	x	x

Bit 7~0 MULA<7:0>: 乘数 A

2.4.6 乘数B寄存器 (MULB)

MULB: 乘数 B 寄存器								
Bit	7	6	5	4	3	2	1	0
Name	MULB<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	x	x	x	x	x	x	x	x

Bit 7~0 MULB<7:0>: 乘数 B

2.4.7 乘积低 8 位寄存器 (MULL)

MULL: 乘积低 8 位寄存器								
Bit	7	6	5	4	3	2	1	0
Name	MULL<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	x	x	x	x	x	x	x	x

Bit 7~0 MUL<7:0>: 乘积低 8 位

2.4.8 乘积高 8 位寄存器 (MULH)

MULH: 乘积高 8 位寄存器								
Bit	7	6	5	4	3	2	1	0
Name	MULH<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0 MUL<15:8>: 乘积高 8 位

2.4.9 被除数低 8 位寄存器 (DIVEL)

DIVEL: 被除数低 8 位寄存器								
Bit	7	6	5	4	3	2	1	0
Name	DIVEL<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVEL<7:0>: 被除数低 8 位

2.4.10 被除数高 8 位寄存器 (DIVEH)

DIVEH: 被除数高 8 位寄存器								
Bit	7	6	5	4	3	2	1	0
Name	DIVEH<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVEH<7:0>: 被除数高 8 位

2.4.11 除数寄存器 (DIVS)

DIVS: 除数寄存器								
Bit	7	6	5	4	3	2	1	0
Name	DIVS<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVS<7:0>: 除数

2.4.12 商低 8 位寄存器 (DIVQL)

DIVQL: 商低 8 位寄存器								
Bit	7	6	5	4	3	2	1	0
Name	DIVQL<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVQL<7:0>: 商低 8 位

2.4.13 商高 8 位寄存器 (DIVQH)

DIVQH: 商高 8 位寄存器								
Bit	7	6	5	4	3	2	1	0
Name	DIVQH<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVQH<7:0>: 商高 8 位

2.4.14 余数寄存器 (DIVR)

DIVR: 余数寄存器								
Bit	7	6	5	4	3	2	1	0
Name	DIVR<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVR<7:0>: 余数

第3章 存储资源

3.1 概述

本芯片采用哈佛总线架构，程序寻址空间和数据寻址空间相互独立。

片内存储器资源包括：

- ◇ 约 8K word FLASH 程序存储器
 - 共分为 31 页，每页 256 word
 - 存储空间 $0000_H \sim 1EFF_H$
 - 支持应用中自编程操作 IAP (In-Application Programming)，包括编程和页擦
 - 支持查表指令读取数据
 - 支持页擦除，擦除时间至少为 2ms
 - 支持单地址编程，编程时间至少为 20us
 - 支持至少 10 万次擦写次数，10 年以上的数据保持时间
 - 擦写时，支持定时器模块正常工作，但不支持中断处理
- ◇ 256 word FLASH 数据存储区
 - 仅一页，为 256 word
 - 存储空间 $1F00_H \sim 1FFF_H$
 - 支持应用中自编程操作 IAP (In-Application Programming)，包括编程和页擦
 - 支持查表指令读取数据
 - 支持页擦除，擦除时间至少为 2ms
 - 支持单地址编程，编程时间至少为 20us
 - 支持至少 10 万次擦写次数，10 年以上的数据保持时间
 - 擦写时，支持定时器模块正常工作，但不支持中断处理
- ◇ 1K byte SRAM 数据存储器

其中 FLASH 程序存储器和 FLASH 数据存储器被映射到程序寻址空间，SRAM 数据存储器被映射到数据寻址空间。

3.2 程序/数据寻址空间映射

程序/数据寻址空间映射如下图：

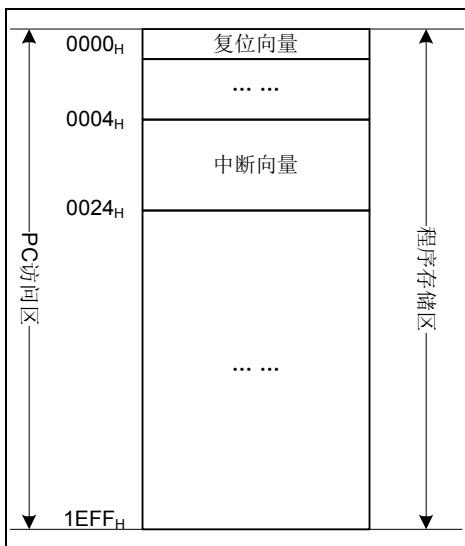


图 3-1 FLASH 程序存储区寻址空间映射图

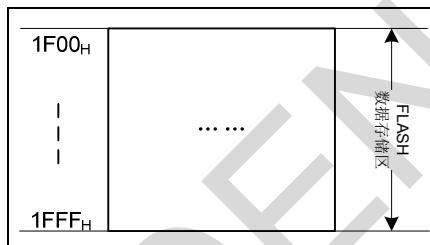


图 3-2 FLASH 数据存储区寻址空间映射图

3.3 程序存储器

3.3.1 概述

FLASH 程序存储器用于存储用户程序。由于芯片指令位宽为 16 位（2 个字节），因此约 8K Word FLASH 程序存储器被映射到程序寻址空间的地址空间 $0000_{\text{H}} \sim 1EFF_{\text{H}}$ ，每个访问地址对应于一个 16 位宽（2 个字节）的存储单元。通过 13 位程序计数器 PC 进行程序寻址访问。

3.3.2 程序计数器 (PC)

程序计数器中存放的是要执行的下一条指令的地址。CPU 运行时，PC 在每个指令周期后都会自动加 1，除非 PC 的值被指令或中断异常改写。13 位程序计数器 PC<12:0>，可寻址 8K 程序存储空间 $0000_{\text{H}} \sim 1EFF_{\text{H}}$ ，超出地址范围会导致 PC 循环（又从 0000_{H} 开始访问）。程序计数器 PC 的低 8 位 PC<7:0>可通过 PCRL 直接读写，而 PC 高 5 位不能直接读写，只能通过 PCRH 寄存器来间接赋值（在执行 RCALL、CALL、GOTO 等指令前，需先对 PCRH 寄存器赋值）。复位时，PCRL、PCRH 和 PC 都会被清零。PC 硬件堆栈操作不会影响 PCRH 的值。

当进行 FLASH 数据存储器进行擦写时，程序计数器 PC 暂停更新。

各种指令对 PC 的影响：

1. 通过指令直接修改 PC 值时，对 PCRL 为目标寄存器的操作可直接修改 PC<7:0>，即 $\text{PC}<7:0>=\text{PCRL}<7:0>$ ；而操作 PC<7:0>的同时也会执行 $\text{PC}<12:8>=\text{PCRH}<4:0>$ ，因此，修改 PC 时，应先修改 PCRH<4:0>，再修改 PCRL<7:0>。
2. 执行 RCALL 指令时，PC<7:0>为寄存器 R 中的值；而 PC<12:8> =PCRH<4:0>。
3. 执行 CALL, GOTO 指令时，PC<12:0>低 11 位为指令中 11 位立即数，而 PC<12> =PCRH<4>。
4. 执行 LCALL 指令时，该指令为双字指令共有 16 位立即数（操作数）。PC<12:0>被修改为该 16 位立即数的值的低 13 位；同时 PCRH<4:0>被修改为 I<12:8>的值。
5. 执行 AJMP 指令时，该指令为双字指令共有 16 位立即数（操作数）。PC<12:0> 被修改为该 16 位立即数的值低 13 位，同时 PCRH<4:0>修改为 I<12:8>的值。
6. 执行 PAGE 指令时，PCRH<4:3>的值将被该指令的立即数 I<1:0>替换。
7. 执行其他指令时，PC 值自动加 1。

应用实例：以 PCRL 为目标寄存器的指令应用程序

```

.....
MOVI    pageaddr
MOVA    PCRH      ; 设置表格页面地址
MOVI    tableaddr ; 设置偏移量给 A 寄存器
CALL    TABLE     ; 调用子程序方式查表
.....
TABLE:
ADD     PCRL, F   ; PC 加上偏移量，指向访问的地址
RETIA   0x01
RETIA   0x02
RETIA   0x03

```

3.3.3 硬件堆栈

芯片内有 8 级硬件堆栈，堆栈位宽与 PC 位宽相等，用于 PC 的压栈和出栈。执行 CALL、LCALL 和 RCALL 指令或中断被响应后，PC 自动压栈保护；当执行 RET、RETIA 或 RETIE 指令时，堆栈会将最近一次压栈的值恢复至 PC。

8 级硬件堆栈只支持 8 级缓冲操作，即硬件堆栈只保存最近的 8 次压栈值，对于连续超过 8 次的压栈操作，第 9 次的压栈数据会覆盖第 1 次压栈的数据，使得第 1 次的压栈数据丢失。同样，超过 8 次的连续出栈，第 9 次出栈操作，可能使得程序流程不可控。芯片复位后，堆栈指针将重新指向堆栈顶部。

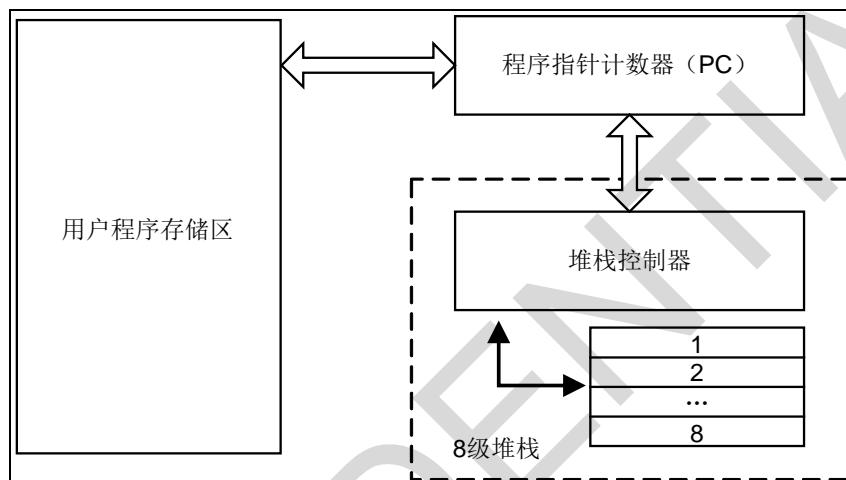


图 3-2 堆栈示意图

3.3.4 存储器IAP操作

3.3.4.1 概述

FLASH 存储器的 IAP 擦除操作是以页（Page）为单位，程序区和数据区每页为 256 个地址单元，页地址对应到 FRAH, FRAL。FLASH 存储器的读出和 IAP 编程写入操作以 1 个地址单元为单位，通过 FRA (FRAH, FRAL) 寻址。

当 FLASH 存储器进行 IAP 擦除或 IAP 写入操作时 CPU 内核暂停执行，外设可按预设状态继续运行，外设的中断请求将置位相应的中断标志。当 IAP 擦除或 IAP 写入操作完成时，CPU 内核恢复执行。

数据存储区和程序存储区都支持查表读和 IAP 擦写操作。

3.3.4.2 查表读操作

本芯片的程序区和数据区都支持查表读操作。通过查表读指令将 FRA (FRAH, FRAL) 所指向的存储器地址中的字 (Word) 读入 ROMD (ROMDH, ROMDL) 中。

本芯片查表写指令保留未用（执行时仅影响 FRA 寄存器）。

在进行查表读操作时，需要设置 IAPSEL<2:0> (ROMCL<6:4>) 选择操作区域，程序区查表读还需使能配置字 FREN (CFG_WD0<13>)，如果 FREN 为 0，禁止查表读，程序区读取的数据为全零；数据区查表读和配置字 FREN 是否使能无关。

应用例程 1：程序存储器查表读

```
MOVI    0x05      ; 读取程序存储器 0105H
MOVA    FRAL
MOVI    0x01
MOVA    FRAH
BSS     ROMCL,6   ; 选择操作区域为程序区
BCC     ROMCL,5   ; 选择操作区域为程序区
BSS     ROMCL,4   ; 选择操作区域为程序区
TBR
MOV     ROMDH, 0
.....
MOV     ROMDL, 0
.....
BCC     ROMCL,6   ; 退出 IAP 操作
BCC     ROMCL,5   ; 退出 IAP 操作
BCC     ROMCL,4   ; 退出 IAP 操作
```

应用例程 2：数据存储器查表读。

```
MOVI    0x05      ; 读取数据存储器 1F05H 单元
MOVA    FRAL
MOVI    0X1F
MOVA    FRAH
BSS     ROMCL,6   ; 选择操作区域为数据区
BCC     ROMCL,5   ; 选择操作区域为数据区
BCC     ROMCL,4   ; 选择操作区域为数据区
TBR
MOV     ROMDH, 0
.....
MOV     ROMDL, 0
.....
BCC     ROMCL,6   ; 退出 IAP 操作
BCC     ROMCL,5   ; 退出 IAP 操作
BCC     ROMCL,4   ; 退出 IAP 操作
```

3.3.4.3 存储器的IAP擦写

程序区和数据区都支持 IAP 擦写操作。

擦写操作需通过存储器控制寄存器（ROMCH, ROMCL）共同完成，同时地址寄存器 FRA 和 FRAN 的值需满足逻辑取反，如果 FRA 和 FRAN 不满足逻辑取反时，硬件自动清零擦写使能位 WREN。包括三个基本操作：数据备份，页擦除，自编程。

数据 FLASH 存储器的擦除以页为单位，通过存储器控制寄存器（ROMCH, ROMCL）可将 FRA 所指向的页擦除。每页擦除时间至少为 2ms。

自编程将 ROMD (ROMDH, ROMDL) 寄存器中的 16 位值写入 FRA 所指向的地址单

元。单个地址编程时间至少为 20us。

3.3.4.4 存储器页更新流程

◆ 编程模式

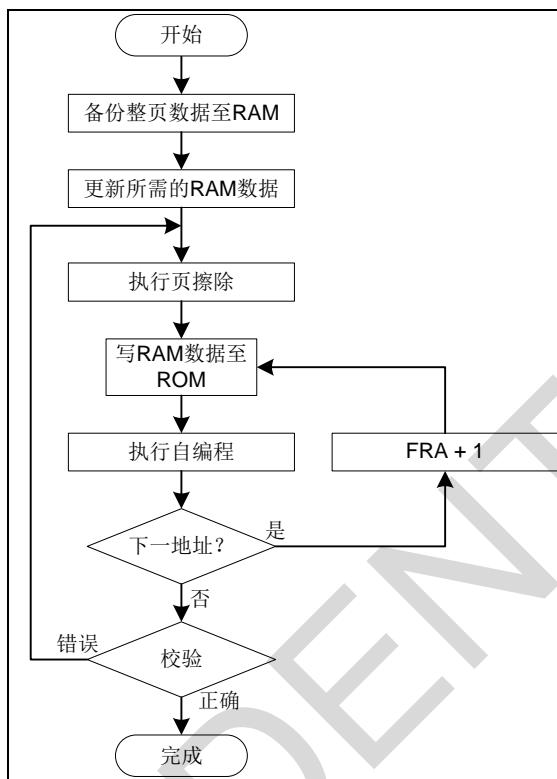


图 3-3 页更新参考流程图

更新一页程序存储器的步骤:

1. 用查表读指令将一页内容备份至数据存储空间 (需 512x2x8 位存储空间, 用于存放一页的数据量);
2. 修改备份数据存储空间要更新的值;
3. 通过设置寄存器 ROMCL 和 ROMCH 进行页擦除 (必须依照固定程序流程进行);
4. 通过寄存器 FRAL 和 FRAH 选择需要更新的地址, 以及设置寄存器 ROMDL 和 ROMDH 需要更新的数据;
5. 通过寄存器 ROMCL 和 ROMCH 将寄存器 ROMDL 和 ROMDH 中的内容写入 FRA 所指向的页中的地址(必须依照固定程序流程进行);
6. 重复 4、5 步骤直至完成整页编程;
7. 用查表读指令进行写入区的校验。

3.3.4.5 操作参考例程

以下例程以数据存储区的 IAP 页擦和编程为例, 同样, 当 IAP 操作选择程序存储区, 也支持 IAP 页擦和编程操作。

应用例程 3: 数据存储器页擦除。

除定时器/计数器可保持运行, 程序停止运行, 直至擦除操作完成自动恢复运行。

```
MOVI    0X00      ;设置擦除页面的地址
MOVA    FRAL
MOVI    0X1F
```

MOVA	FRAH	
MOVI	0xFF	; 设置页面的地址反码
MOVA	FRALN	
MOVI	0xE0	
MOVA	FRAHN	
BSS	ROMCL,6	; 选择操作区域为数据区
BCC	ROMCL,5	; 选择操作区域为数据区
BCC	ROMCL,4	; 选择操作区域为数据区
BSS	ROMCL, FPEE	; 选择擦除操作
BSS	ROMCL, WREN	; 打开 FLASH 擦除/编程使能
BCC	INTG, GIE	; 关闭全局中断（避免中断影响后续固定程序流程）
MOVI	0x55	
MOVA	ROMCH	
....		; 8 个 NOP 指令，或等待 8 个指令周期
MOVI	0xAA	
MOVA	ROMCH	
....		; 8 个 NOP 指令，或等待 8 个指令周期
BSS	ROMCL, WR	
.....		
BCC	ROMCL,6	; 退出 IAP 操作
BCC	ROMCL,5	; 退出 IAP 操作
BCC	ROMCL,4	; 退出 IAP 操作
CLR	FRALN	; 退出 IAP 操作
CLR	FRAHN	; 退出 IAP 操作

应用例程 4：将数据缓冲器写入数据存储器。

除定时器/计数器可保持运行，程序停止运行，直至擦除操作完成自动恢复运行。

MOVI	0x00	; 写入数据 FLASH 存储器的第 1 页第 1 个地址
MOVA	FRAL	
MOVI	0x1F	
MOVA	FRAH	
MOVI	0xFF	; 设置页面的地址反码
MOVA	FRALN	
MOVI	0xE0	
MOVA	FRAHN	
MOVI	0x12	
MOVA	ROMDH	
MOVI	0x34	
MOVA	ROMDL	; 写入数据 1234 _H
BSS	ROMCL,6	; 选择操作区域为数据区
BCC	ROMCL,5	; 选择操作区域为数据区
BCC	ROMCL,4	; 选择操作区域为数据区

BCC	ROMCL, FPEE	; 选择编程操作																														
BSS	ROMCL, WREN	; 打开 FLASH 擦除/编程使能																														
BCC	INTG, GIE	; 关闭全局中断（避免中断影响后续固定程序流程）																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>MOVI</td><td>0x55</td></tr> <tr><td>MOVA</td><td>ROMCH</td></tr> <tr><td>....</td><td></td></tr> <tr><td>MOVI</td><td>0xAA</td></tr> <tr><td>MOVA</td><td>ROMCH</td></tr> <tr><td colspan="2">BSS ROMCL, WR</td></tr> <tr><td colspan="3">.....</td></tr> <tr><td>BCC</td><td>ROMCL,6</td><td>; 退出 IAP 操作</td></tr> <tr><td>BCC</td><td>ROMCL,5</td><td>; 退出 IAP 操作</td></tr> <tr><td>BCC</td><td>ROMCL,4</td><td>; 退出 IAP 操作</td></tr> <tr><td>CLR</td><td>FRALN</td><td>; 退出 IAP 操作</td></tr> <tr><td>CLR</td><td>FRAHN</td><td>; 退出 IAP 操作</td></tr> </table>			MOVI	0x55	MOVA	ROMCH		MOVI	0xAA	MOVA	ROMCH	BSS ROMCL, WR				BCC	ROMCL,6	; 退出 IAP 操作	BCC	ROMCL,5	; 退出 IAP 操作	BCC	ROMCL,4	; 退出 IAP 操作	CLR	FRALN	; 退出 IAP 操作	CLR	FRAHN	; 退出 IAP 操作
MOVI	0x55																															
MOVA	ROMCH																															
....																																
MOVI	0xAA																															
MOVA	ROMCH																															
BSS ROMCL, WR																																
.....																																
BCC	ROMCL,6	; 退出 IAP 操作																														
BCC	ROMCL,5	; 退出 IAP 操作																														
BCC	ROMCL,4	; 退出 IAP 操作																														
CLR	FRALN	; 退出 IAP 操作																														
CLR	FRAHN	; 退出 IAP 操作																														

注：上述应用例程方框中的程序为固定操作格式，使用时不可改变。

3.3.5 特殊功能寄存器

3.3.5.1 查表地址寄存器低 8 位 (FRAL)

FRAL: 查表地址寄存器低 8 位								
Bit	7	6	5	4	3	2	1	0
Name	FRAL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~0 FRAL<7:0>: 查表地址低 8 位

3.3.5.2 查表地址寄存器高 8 位 (FRAH)

FRAH: 查表地址寄存器高 8 位								
Bit	7	6	5	4	3	2	1	0
Name	FRAH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~0 FRAH<7:0>: 查表地址高 8 位

3.3.5.3 查表地址取反寄存器低 8 位 (FRALN)

FRALN: 查表地址取反寄存器低 8 位								
Bit	7	6	5	4	3	2	1	0
Name	FRALN<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 FRALN<7:0>: 查表地址取反低 8 位

3.3.5.4 查表地址取反寄存器高 8 位 (FRAHN)

FRAHN: 查表地址取反寄存器高 8 位								
Bit	7	6	5	4	3	2	1	0
Name	FRAHN<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 FRAHN<7:0>: 查表地址取反高 8 位

3.3.5.5 查表数据寄存器低 8 位 (ROMDL)

ROMDL: 查表数据寄存器低 8 位								
Bit	7	6	5	4	3	2	1	0
Name	ROMDL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~0 ROMDL<7:0>: 查表数据低 8 位

3.3.5.6 查表数据寄存器高 8 位 (ROMDH)

ROMDH: 查表数据寄存器高 8 位								
Bit	7	6	5	4	3	2	1	0
Name	ROMDH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~0 ROMDH<7:0>: 查表数据高 8 位

3.3.5.7 存储器控制寄存器 (ROMCL)

ROMCL: 存储器控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	IAPSEL<2:0>			FPEE	WREN	WR	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit7 未使用

Bit 6~4 IAPSEL<2:0>: IAP 操作区间选择

110: 保留

	101: 程序区
	100: 数据区
	其它: 禁止 IAP 操作
Bit 3	FPEE: 存储器页擦除/编程选择位 0: 编程 1: 擦除
Bit 2	WREN: 存储器页擦除/编程使能位 0: 禁止, 如果 FRA 和 FRAN 不满足逻辑取反时, 硬件自动清零 WREN 1: 使能
Bit 1	WR: 存储器擦除/编程触发位 0: 未启动擦除/编程操作, 或操作已完成 1: 擦除/编程正在进行操作 (硬件自动清零)
Bit 0	未使用, 必须保持复位值 0

3.3.6 存储器控制寄存器 (ROMCH)

ROMCH: 存储器控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	ROMCH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 ROMCH<7:0>: 存储器擦除/编程控制字

注: ROMCH 寄存器为虚拟寄存器, 对该寄存器读出始终为全 0。

3.4 在线编程ISP和在线调试ICD

FLASH 存储器和 FLASH 数据存储器具有可重复烧写的功能, 便于客户代码和数据的更新升级。为了实现开发人员在开发过程中能够更轻松地进行代码的调试、更新、升级, 本芯片还支持在线编程 ISP 和在线调试 ICD, 用户只需在电路系统板上引出五根编程和调试接口线, 即可实现程序的重新烧录和调试, 更加方便高效。

芯片管脚	管脚说明
ISPCK ^{注1}	编程/调试串行时钟端口
ISPDA ^{注1}	编程/调试串行数据端口
VDD	电源
VSS	地

表 3-1 在线编程/调试管脚说明

注: 对编程/调试接口中的 ISPCK 和 ISPDA 管脚, 芯片支持 PC0 和 PC1 作为一组编程/调试接口。

3.5 数据存储器

3.5.1 概述

- ◇ 数据存储器由 2 部分组成
 - 通用数据存储器 GPR
 - 特殊功能寄存器 SFR
- ◇ 物理存储包括
 - 1K Byte 数据存储器
 - 128 个特殊寄存器
- ◇ 支持 3 种寻址方式
 - 直接寻址
 - GPR 特殊寻址
 - 间接寻址

3.5.2 通用数据存储器

通用数据存储器被用于临时存放数据和控制信息，可以在程序控制下进行读写操作。本芯片通用数据存储器空间为 1K Byte，支持 8 个存储体组，地址范围为 0000_H ~ $03FF_H$ 。程序控制过程中，对这些存储体访问时，需通过寄存器 BKSR 选择存储体，实现在不同存储体间的跳转。通用数据存储器的内容在上电复位后是不确定的，未掉电的其它复位后，将保存复位前的内容。

地址映射如下：

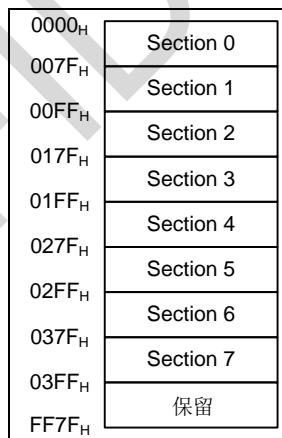


图 3-4 GPR 地址映射示意图

3.5.3 特殊功能寄存器

特殊功能寄存器用于芯片对外设操作的控制设定。本芯片支持 128 个特殊寄存器，地址范围 $FF80_H$ ~ $FFFF_H$ 。大多数寄存器都是可以读写的，仅有少数寄存器作为保留使用，用户程序不能进行读写。相关功能所使用的寄存器将分别在各个章节中描述。

FF80 _H	IAD	FFA0 _H	SPICON0	FFC0 _H	TIM1_CR2H	FFE0 _H	RXB
FF81 _H	IAAL	FFA1 _H	SPICON1	FFC1 _H	TIM1_SMCR1	FFE1 _H	RXC
FF82 _H	IAAH	FFA2 _H	SPIIE	FFC2 _H	TIM1_SMCRH	FFE2 _H	TXB
FF83 _H	BKSR	FFA3 _H	SPIIF	FFC3 _H	TIM1_IERL	FFE3 _H	TXC
FF84 _H	PSW	FFA4 _H	SPIRBR	FFC4 _H	TIM1_IERH	FFE4 _H	BRR
FF85 _H	AREG	FFA5 _H	SPITBW	FFC5 _H	TIM1_IDRL	FFE5 _H	T21R1L
FF86 _H	PCRL	FFA6 _H	PWRC	FFC6 _H	TIM1_IDRH	FFE6 _H	T21R1H
FF87 _H	PCRH	FFA7 _H	WDTC	FFC7 _H	TIM1_IVSL	FFE7 _H	T21R2L
FF88 _H	MULA/MULL	FFA8 _H	WKDC	FFC8 _H	TIM1_IVSH	FFE8 _H	T21R2H
FF89 _H	MULB/MULH	FFA9 _H	PWEN	FFC9 _H	TIM1_RIFL	FFE9 _H	T21OC
FF8A _H	DIVEL/DIVQL	FFAA _H	PA	FFCA _H	TIM1_RIFH	FFEA _H	TIM1_CCMR1H
FF8B _H	DIVEH/DIVQH	FFAB _H	PAT	FFCB _H	TIM1_IFML	FFEB _H	TIM1_CCMR2L
FF8C _H	DIVS/DIVR	FFAC _H	PB	FFCC _H	TIM1_IFMH	FFEC _H	TIM1_CCMR2H
FF8D _H	TIM1_CCR2H	FFAD _H	PBT	FFCD _H	TIM1_ICRL	FFED _H	TIM1_CCERL
FF8E _H	TIM1_CCR3L	FFAE _H	PC	FFCE _H	TIM1_ICRH	FFEE _H	TIM1_CCERH
FF8F _H	TIM1_CCR3H	FFAF _H	PCT	FFCF _H	TIM1_EGRL	FFEF _H	I2CX16
FF90 _H	FRAL	FFB0 _H	PAPU	FFD0 _H	TIM1_CCMR1L	FFF0 _H	I2CC
FF91 _H	FRAH	FFB1 _H	PBPU	FFD1 _H	T21L	FFF1 _H	I2CSA
FF92 _H	ROMDL	FFB2 _H	PCPU	FFD2 _H	T21H	FFF2 _H	I2CTB
FF93 _H	ROMDH	FFB3 _H	TIM1_BDTRH	FFD3 _H	T21PL	FFF3 _H	I2CRB
FF94 _H	ROMCL	FFB4 _H	TIM1_CCR4H	FFD4 _H	T21PH	FFF4 _H	I2CIEC
FF95 _H	ROMCH	FFB5 _H	PORTCTR	FFD5 _H	T21R0L	FFF5 _H	I2CIFC
FF96 _H	INTG	FFB6 _H	TIM1_BDTRL	FFD6 _H	T21R0H	FFF6 _H	TIM1_CNTL
FF97 _H	INTP	FFB7 _H	PAPD	FFD7 _H	T21CL	FFF7 _H	TIM1_CNTH
FF98 _H	INTCO	FFB8 _H	PBDP	FFD8 _H	T21CM	FFF8 _H	TIM1_PSCL
FF99 _H	TIM1_CCR4L	FFB9 _H	FRALN	FFD9 _H	T21CH	FFF9 _H	TIM1_PSCH
FF9A _H	INTE0	FFBA _H	FRAHN	FFDA _H	ADCRL	FFFA _H	TIM1_ARRL
FF9B _H	INTFO	FFBB _H	T8N	FFDB _H	ADCRH	FFFB _H	TIM1_ARRH
FF9C _H	INTE1	FFBC _H	T8NC	FFDC _H	ADCCL	FFFC _H	TIM1_RCR
FF9D _H	INTF1	FFBD _H	TIM1_CR1L	FFDD _H	ADCCH	FFFD _H	TIM1_CCR1L
FF9E _H	INTE2	FFBE _H	TIM1_CR1H	FFDE _H	ANSL	FFFE _H	TIM1_CCR1H
FF9F _H	INTF2	FFBF _H	TIM1_CR2L	FFDF _H	—	FFFF _H	TIM1_CCR2L

图 3-5 特殊功能寄存器空间

3.5.4 寻址方式

SRAM 数据存储器的寻址方式支持直接寻址、GPR 特殊寻址和间接寻址。

3.5.4.1 直接寻址

直接寻址的地址信息由两部分组成，BKSR 和指令中的 8 位地址信息。BKSR 用于选择存储体组，指令中的 8 位地址信息用于在 BKSR 所选的存储体组中寻址。

在直接寻址时，当指令中的 8 位地址信息大于或等于 80_H 时，将忽略 BKSR 而直接寻址 SFR 映射区。当指令中的 8 位地址信息小于 80_H 时，访问 GPR 地址映射区。

示意图如下：

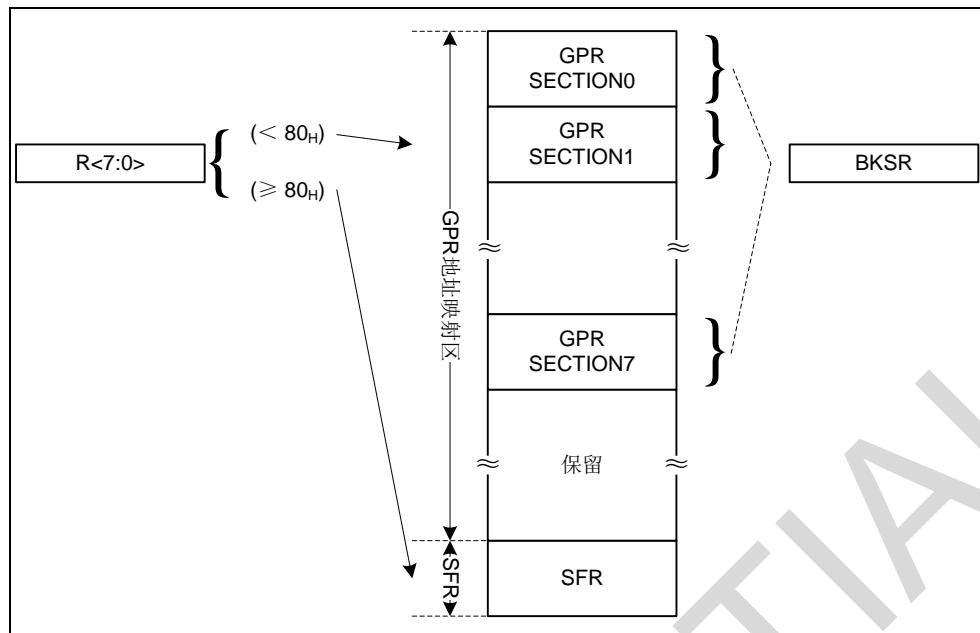


图 3-6 直接寻址示意图

3.5.4.2 GPR 特殊寻址

为方便较大的数据段（例如数组）在 GPR 中的移动，指令 MOVAR 和 MOVRA 用于对 GPR 进行特殊寻址操作，本芯片 MOVAR 和 MOVRA 指令最大支持 10 位地址信息 ($R<9:0>$)，可直接寻址 1K 字节地址空间。无需进行 SECTION 间切换。

MOVAR 和 MOVRA 指令无法访问 SFR。

示意图如下：

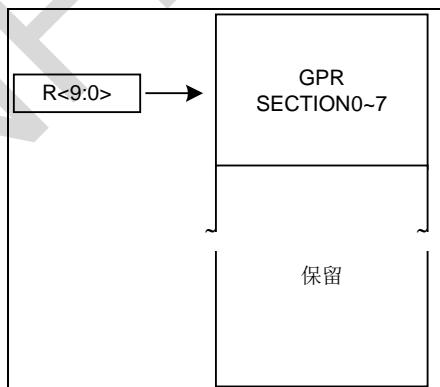


图 3-7 GPR 特殊寻址示意图

3.5.4.3 间接寻址

间接寻址是通过 16 位间接地址寄存器 IAA（由 2 个 8 位寄存器 IAAL 和 IAAB 组成）和 8 位虚拟数据寄存器 IAD，间接访问数据寻址空间中的存储单元。先将访问目的地址存放于间接地址寄存器 IAA，再通过指令对 IAD 进行读/写操作，实际的读/写操作对象则是 IAA 指向的数据寻址空间中的目的地址单元。

IAD 寄存器本身也映射到数据寻址空间的 $FF80H$ 地址，因此当 IAA 存放的地址值为 $FF80H$

时, 读/写 IAD 相当于用间接寻址方式访问虚拟寄存器 IAD 本身, 此时读操作将始终读出为 $00H$, 写操作则是一个空操作 (可能影响状态位)。

ISTEP 指令, 用来对 16 位间接地址寄存器 IAA 进行偏移操作。执行该指令时, 先将指令字中的 8 位有符号立即数进行符号位扩展为 16 位数, 再将 IAA 的值加上这个数的结果存回 IAA 寄存器。ISTEP 可实现的偏移范围为 -128~127。

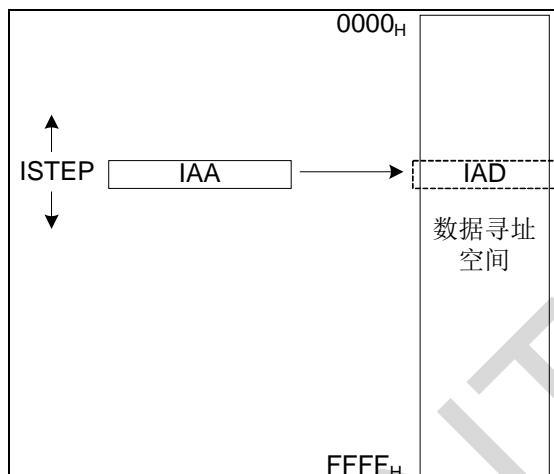


图 3-8 间接寻址示意图

3.5.5 特殊功能寄存器

3.5.5.1 间接寻址数据寄存器 (IAD)

IAD: 间接寻址数据寄存器								
Bit	7	6	5	4	3	2	1	0
Name	IAD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 IAD<7:0>: 间接寻址数据

3.5.5.2 间接寻址索引寄存器低 8 位 (IAAL)

IAAL: 间接寻址索引寄存器低 8 位								
Bit	7	6	5	4	3	2	1	0
Name	IAAL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 IAAL<7:0>: 间接寻址索引低 8 位

3.5.5.3 间接寻址索引寄存器高 8 位 (IAAH)

IAAH: 间接寻址索引寄存器高 8 位								
Bit	7	6	5	4	3	2	1	0
Name	IAAH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 IAAH<7:0>: 间接寻址索引高 8 位

3.5.5.4 存储体选择寄存器 (BKSR)

BKSR: 存储体选择寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	DBKSR<2:0>		
R/W	—	—	—	R/W	—	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~5 未使用

Bit 4 未使用, 必须保持复位值, 或者软件设置为 0

Bit 3 未使用

Bit 2~0 DBKSR<2:0>: 数据存储体选择位

000: 选择存储体 0

001: 选择存储体 1

010: 选择存储体 2

011: 选择存储体 3

100: 选择存储体 4

101: 选择存储体 5

110: 选择存储体 6

111: 选择存储体 7

注: BKSR<4>位必须保持复位值 0, 或者软件设置为 0。

第4章 输入/输出端口

4.1 概述

输入/输出端口是芯片的最基本组成部分，本芯片最多支持 18 个输入端口。所有 I/O 端口都是 TTL/SMT 输入和 CMOS 输出驱动。

- ◇ PA 输入/输出端口功能组件
 - 8 位双向输入/输出端口
 - TTL/SMT 输入和 CMOS 输出驱动
 - 端口输入输出控制寄存器（PAT）
 - 端口弱上拉控制寄存器（PAPU）
 - 端口弱下拉控制寄存器（PAPD）
 - 数/模端口控制寄存器（ANSL）
 - PA0, PA3~7 支持外部端口中断功能
- ◇ PB 输入/输出端口功能组件
 - 8 位双向输入/输出端口
 - TTL/SMT 输入和 CMOS 输出驱动
 - 端口输入输出控制寄存器（PBT）
 - 端口弱上拉控制寄存器（PBPU）
 - 端口弱下拉控制寄存器（PBPD）
 - 端口 PB0~1 有开漏输出功能
 - PB4~5, PB1 支持外部端口中断功能
- ◇ PC 输入/输出端口功能组件
 - 2 位双向输入/输出端口
 - TTL/SMT 输入和 CMOS 输出驱动
 - 端口输入输出控制寄存器（PCT）
 - 端口弱上拉控制寄存器（PCPU）

注：当端口设置为输出或者模拟输入端口时，内部弱上/下拉自动禁止。

4.2 结构框图

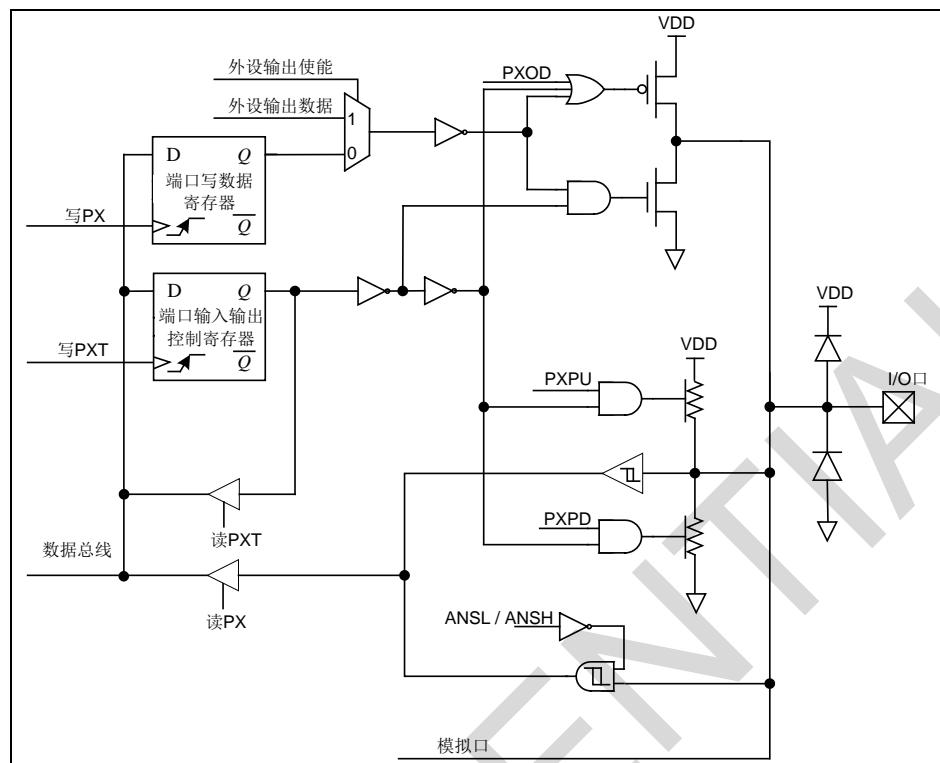


图 4-1 PA/PB/PC 端口结构图

注 1: PC 端口无弱下拉功能。

注 2: 除 PB0/PB1 外, 其它端口无开漏输出功能。

4.3 I/O 端口功能设置

4.3.1 I/O 端口输入/输出控制

芯片中的所有 I/O 端口都具有输入/输出的能力, 端口控制寄存器 PAT/PBT/PCT 用于相应端口的输入或输出功能选择。当 I/O 端口设置为数字输出状态时, I/O 端口输出 PA/PB/PC 寄存器内容, 即相应 I/O 端口电平状态, 读取 PA/PB/PC 寄存器的操作实际为读取相应 I/O 端口电平状态。当 I/O 端口设置为数字输入状态时, 读取 PA/PB/PC 寄存器的操作实际为读取相应 I/O 端口电平状态。

4.3.2 I/O 端口弱上/下拉功能

很多产品的应用中需要端口连接上拉或下拉电阻, 使端口固定在一个稳定的电平状态, 防止外界干扰以及其它影响。(外部复位端口内部弱上拉默认使能)

管脚	0	1	2	3	4	5	6	7
PA	支持							
PB	支持							
PC	支持	支持	—	—	—	—	—	—

表 4-1 I/O 端口弱上拉

管脚	0	1	2	3	4	5	6	7
PA	支持							
PB	支持							
PC	—	—	—	—	—	—	—	—

表 4-2 I/O 端口弱下拉

4.3.3 I/O 端口模拟/数字类型选择功能

当数字信号和模拟信号共用管脚时，在使用对应端口的数字信号或模拟信号功能前，须正确设置端口的类型，否则可能不会达到预期的结果。本芯片中部分 I/O 均具有独立的模拟/数字信号选择功能，由 ANSL 寄存器控制选择。当端口被配置为模拟端口时，读相应的 PA/PB 寄存器时，始终读到“0”。

4.3.4 I/O 端口开漏输出

端口 PB0~1 支持开漏输出功能，可独立设置为开漏输出。

管脚	0	1	2	3	4	5	6	7
PA	—	—	—	—	—	—	—	—
PB	支持	支持	—	—	—	—	—	—
PC	—	—	—	—	—	—	—	—

表 4-3 I/O 端口开漏输出

4.3.5 I/O 端口复用功能

为了使资源合理利用最优化，本芯片的所有 I/O 端口都具有复用功能。当端口用于复用功能时，管脚电平由复用功能决定。

4.4 端口中断

4.4.1 外部端口中断 (PINT)

本芯片支持 9 个外部端口中断。当 PINT0~PINT8 复用端口被配置为数字输入端口，且输入信号变化满足触发条件时，将产生 PINT0~PINT8 外部端口中断。可配置为上升沿触发、下降沿触发或双沿触发。外部端口中断可由 PIE0~PIE8 使能。中断产生将影响相应的中断标志 PIF0~PIF8。在 IDLE 模式下，此中断能唤醒 CPU。

管脚名	端口输入	边沿选择	中断名	中断使能	中断标志
PA0, PA3~PA7, PB4~PB5, PB1	PINT0~8	PEGx	PINT0~8	PIE0~8	PIF0~8

表 4-4 外部端口中断

4.5 I/O端口操作注意事项

当执行以端口寄存器为目标的算术或逻辑运算指令（除位操作指令）时，芯片实际执行读-修改-写过程，即先读取该组全部I/O端口的电平，修改后再写回端口寄存器。位操作指令对I/O的修改操作只影响选定的位，对同组其它I/O不影响。因此建议用户对单个I/O的修改采用位操作指令。此外在I/O复用功能使能和关闭时，应充分考虑当前I/O端口的输出寄存器值，并判断是否需要重新对这些I/O端口进行初始化赋值。

4.6 特殊功能寄存器

4.6.1 PA端口电平状态寄存器（PA）

PA: PA 端口电平状态寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PA<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~0 PA<7:0>: PA 端口电平状态

- 0: 低电平
- 1: 高电平

4.6.2 PA端口输入输出控制寄存器（PAT）

PAT: PA 端口输入输出控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PAT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1

Bit 7~0 PAT<7:0>: PA 端口输入输出状态控制位

- 0: 输出状态
- 1: 输入状态

注：PA3 只有在配置为IO端口时，输入/输出功能才都有效，用作外部复位管脚时，固定为输入。

4.6.3 PA端口弱上拉控制寄存器（PAPU）

PAPU: PA 端口弱上拉控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PAPU<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	0	0	0

Bit 7~0 PAPU<7:0>: PA 端口内部弱上拉控制位

- 0: 禁止
- 1: 使能

4.6.4 PA端口弱下拉控制寄存器 (PAPD)

PAPD: PA 端口弱下拉控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PAPD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 **PAPD<7:0>: PA 端口内部弱下拉控制位**

0: 禁止

1: 使能

注 1: PA3 只有在配置为 IO 端口时, 弱下拉功能、输入/输出功能才有效;

注 2: PA3 配置为外部复位管脚时, 固定为输入;

注 3: PA3 默认弱上拉使能。

4.6.5 PB端口电平状态寄存器 (PB)

PB: PB 端口电平状态寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PB<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~0 **PB<7:0>: PB 端口电平状态**

0: 低电平

1: 高电平

4.6.6 PB端口输入输出控制寄存器 (PBT)

PBT: PB 端口输入输出控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PBT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1

Bit 7~0 **PBT<7:0>: PB 端口输入输出状态控制位**

0: 输出状态

1: 输入状态

4.6.7 PB端口弱上拉控制寄存器 (PBPU)

PBPU: PB 端口弱上拉控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PBPU<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 **PBPU<7:0>: PB 端口内部弱上拉控制位**

0: 禁止
1: 使能

4.6.8 PB端口弱上拉控制寄存器 (PBPD)

PBPD: PB 端口弱下拉控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	PBPD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 PBPD<7:0>: PB 端口内部弱下拉控制位

0: 禁止
1: 使能

4.6.9 PC端口电平状态寄存器 (PC)

PC: PC 端口电平状态寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PC<1:0>	
R/W	—	—	—	—	—	—	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~2 未使用

Bit 1~0 PC<1:0>: PC1~PC0 端口电平状态

0: 低电平
1: 高电平

4.6.10 PC端口输入输出控制寄存器 (PCT)

PCT: PC 端口输入输出控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PCT<1:0>	
R/W	—	—	—	—	—	—	R/W	R/W
RESET	0	0	0	0	0	0	1	1

Bit 7~2 未使用

Bit 1~0 PCT<1:0>: PC1~PC0 端口输入输出状态控制位

0: 输出状态
1: 输入状态

4.6.11 PC端口弱上拉控制寄存器 (PCPU)

PCPU: PC 端口弱上拉控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PCCPU<1:0>	
R/W	—	—	—	—	—	—	R/W	R/W
RESET	0	0	0	0	0	0	0	0

- Bit 7~2 未使用
- Bit 1~0 PCPU<1:0>: PC1~PC0 端口内部弱上拉控制位
0: 禁止
1: 使能

4.6.12 端口特殊功能控制寄存器 (PORTCTR)

PORTCTR: 端口特殊功能控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	TIM1_CH4EN	TIM1_CH3EN	AD_ETR1EN	ADC_ETR0EN	PBOD<1:0>	
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

- Bit 7~6 保留未用
- bit5 高级 TIM1 通道 4 输入/输出选择位
0: PB3 输入/输出使能
1: PA4 输入/输出使能
- bit4 高级 TIM1 通道 3 输入/输出选择位
0: PB2 输入/输出使能
1: PA3 输入/输出使能
- bit3 ADC_ETR1: 外部中断 8 触发 ADC 转换使能位
0: 禁止
1: 使能
- bit2 ADC_ETR0: 外部中断 5 触发 ADC 转换使能位
0: 禁止
1: 使能
- Bit 1~0 PBOD<1:0>: PB1~PB0 端口开漏输出控制位
0: 禁止
1: 使能

注: 具体 IO 端口的驱动能力, 可参考附录 3《电气特性》。

第5章 特殊功能及操作特性

5.1 系统时钟与振荡器

5.1.1 概述

芯片运行所需要的时钟源由振荡器提供，不同的振荡器选择可以让使用者在不同的应用需求中实现更大范围的功能。本款芯片所提供的振荡器有2种：内部高速RC振荡器（16MHz）和内部低速RC振荡器（32KHz）。灵活选择振荡器，使得产品在速度和功耗方面可以达到最优化。振荡器除了作为系统时钟源外，还可以为看门狗定时器、ADC电路、Timer等提供所需要的时钟源。

◇ INTHRC

- 内部 16MHz RC 振荡器
- 出厂前，内部 RC 振荡器频率已经在常温下校准，校准精度在±1%以内
- 支持多种分频时钟，可通过配置字选择

◇ INTLRC

- 内部 32KHz RC 振荡器
- WDT 计数时钟
- T8N 计数时钟
- AD 转换时钟

◇ 振荡和暂停

- 在 IDLE0 模式下，主系统时钟振荡器暂停振荡，LDO 暂停
- 在 IDLE1 模式下，主系统时钟 RC 振荡器暂停振荡，LDO 工作
- 在 IDLE2 模式下，主系统时钟 RC 振荡器不停振，LDO 工作

5.1.2 内部结构图

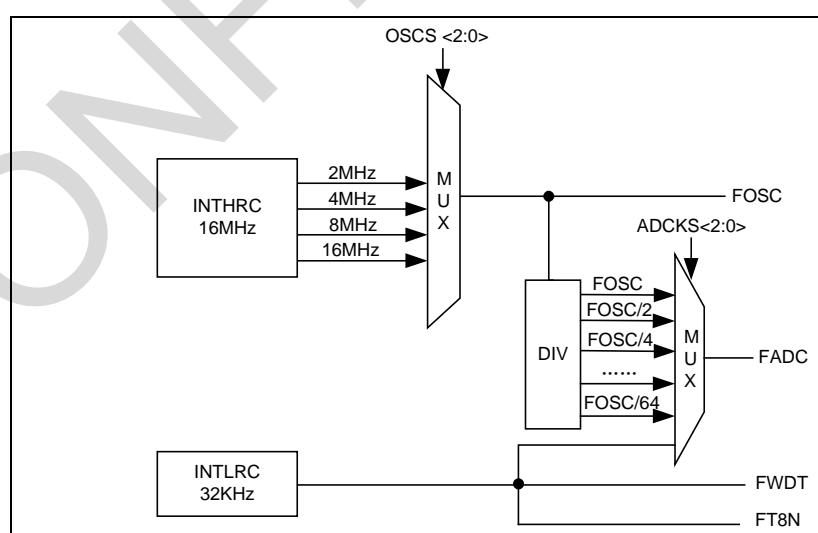


图 5-1 系统时钟结构图

5.1.3 时钟源

5.1.3.1 内部高速 16MHz RC 振荡器模式 (INTHRC)

芯片内置 16MHz RC 时钟振荡器，不需要外接其它外部器件。

当芯片配置字 OSCS<2:0> = 111、101、100 或 011 时，配置为 INTOSCIO 模式，此时 PB3 管脚复用为通用 I/O 端口。

当芯片配置字 OSCS<2:0> = 110 时，配置为 INTOSC 模式，此时 PB3 管脚复用输出 CLKO，CLKO 输出系统时钟的 16 分频时钟 (Fosc/16)。客户通过编程界面选择。

在出厂前，芯片已经在常温下校准，在工作电压范围内，INTHRC 时钟频率校准精度在 ±1% 以内。

5.1.3.2 内部低速 32kHz RC 振荡器模式 (INTLRC)

芯片内置 32KHz RC 时钟振荡器，不需要外接其它外部器件，可用作 WDT、ADC 模块、T8N 模块的时钟源。

5.2 看门狗定时器 (WDT)

5.2.1 概述

看门狗定时器是芯片的一个组成部分，它可以在发生软件故障时，将芯片复位。若系统进入了错误的工作状态，看门狗可以在合理的时间范围内使芯片复位。使能看门狗时，若用户程序清除看门狗定时器失败，则在预定的时间范围内，看门狗会使系统复位。

◇ WDT 定时器

- 8 位 WDT 定时计数器（无实际物理地址，不可读写）
- 定时器时钟源为内部 32KHz RC 时钟
- 8 位预分频器（无实际物理地址，不可读写）
- WDT 控制寄存器 (WDTC)
- 唤醒功能
- 复位功能

5.2.2 内部结构图

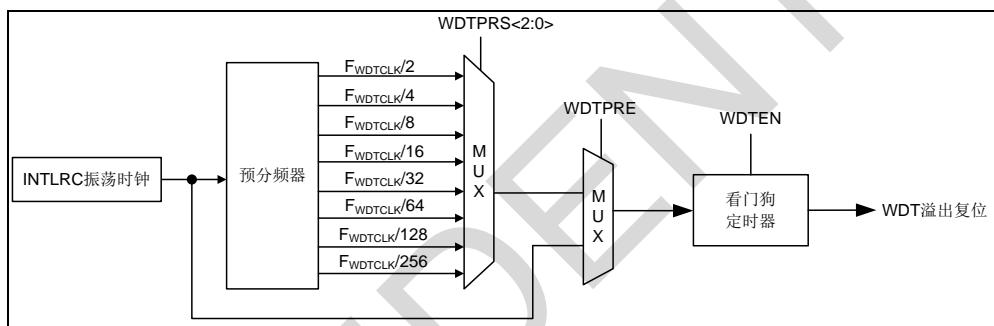


图 5-2 看门狗定时器内部结构图

5.2.3 WDT 定时器

芯片提供 8 位 WDT 定时计数器，通过芯片配置字 WDTEN 可使能硬件看门狗 WDT。当芯片配置字 WDTEN 使能时，WDT 定时器计数使能；当 WDTEN 关闭时，WDT 定时器计数禁止。客户可通过编程器界面选择。

在 IDLE 模式下，WDT 计数溢出会唤醒 CPU，不复位芯片；在正常运行模式下，WDT 计数溢出会复位芯片。为了避免不必要的复位，需使用 CWDT 指令适时清零 WDT 计数器。

WDT 支持一个预分频器，由 WDTC 寄存器中的 WDTPRE 位控制。当 WDTPRE 位清零，禁止预分频器时，常温下 WDT 的计数溢出时间约为 8ms。

当 WDTPRE 位置 1，使能预分频器时，可通过 WDTC 寄存器中的 WDTPRS<2:0>位设置 WDT 时钟源的预分频比，再将分频后的时钟信号作为 WDT 定时器的计数时钟。

5.2.4 特殊功能寄存器

5.2.4.1 WDT控制寄存器 (WDTC)

WDTC: WDT 控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WDTPRE	WDTPRS<2:0>		
R/W	—	—	—	—	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	1	1	1

Bit 7~4 未使用

Bit 3 WDTPRE: WDT 预分频器使能位

0: 禁止

1: 使能

Bit 2~0 WDTPRS <2:0>: WDT 预分频器分频比选择位

000: 1:2

001: 1:4

010: 1:8

011: 1:16

100: 1:32

101: 1:64

110: 1:128

111: 1:256

5.3 复位模块

5.3.1 概述

- ◇ 上电复位 POR
- ◇ 掉电复位 BOR，复位电压点可配置
- ◇ 外部端口 MRSTN 复位，低电平复位有效
- ◇ 看门狗定时器 WDT 溢出复位
- ◇ 软件执行指令 RST 复位

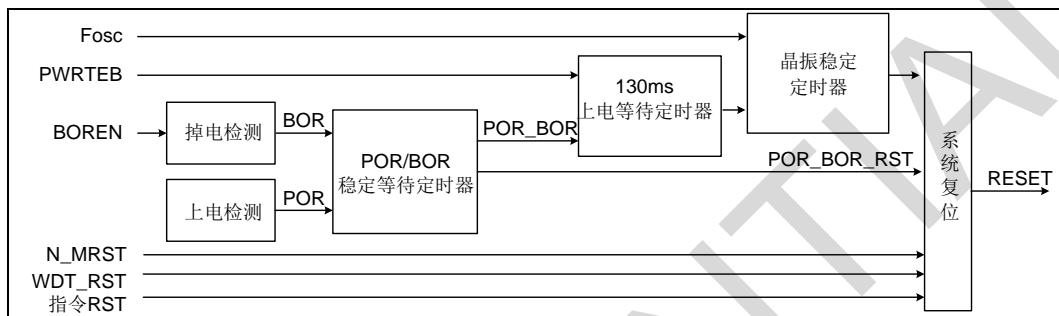


图 5-3 芯片复位原理图

5.3.2 上电复位

芯片上电过程中会产生 POR 复位，并且该复位信号将会一直保持到电源电压升高到芯片能够正常工作的电压为止。系统上电过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。上电复位的时序如下：

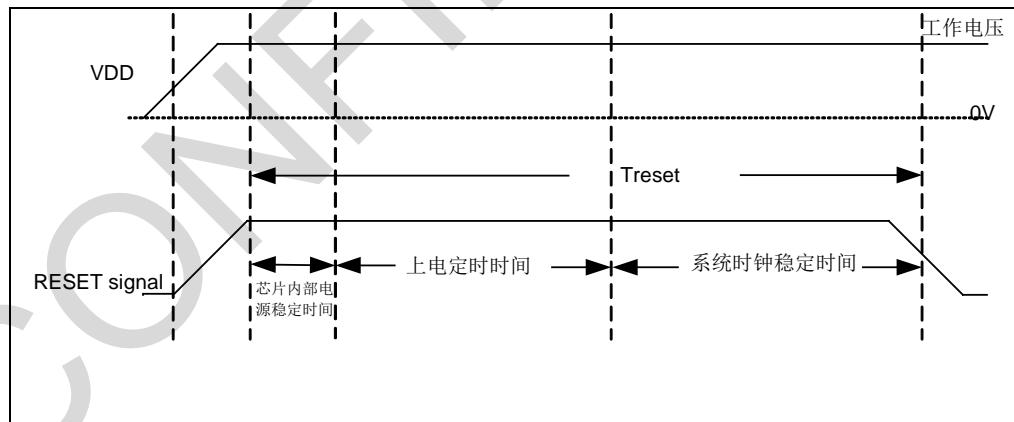


图 5-4 上电复位时序示意图

5.3.3 掉电复位

掉电复针对外部因素引起的系统电压跌落情形（例如：更换电池），掉电时可能会引起系统工作状态不正常或程序执行错误，掉电复位电路可保障芯片在异常掉电过程中处于复位状态，避免出现误操作。对电压跌落的滤波时间 Tfilter，可通过寄存器 PWEN<3:2>进行设置，根据所配置的 BOR 低电压档位和应用系统的供电情况，选择合适的滤波时间，通常保持为默认值。

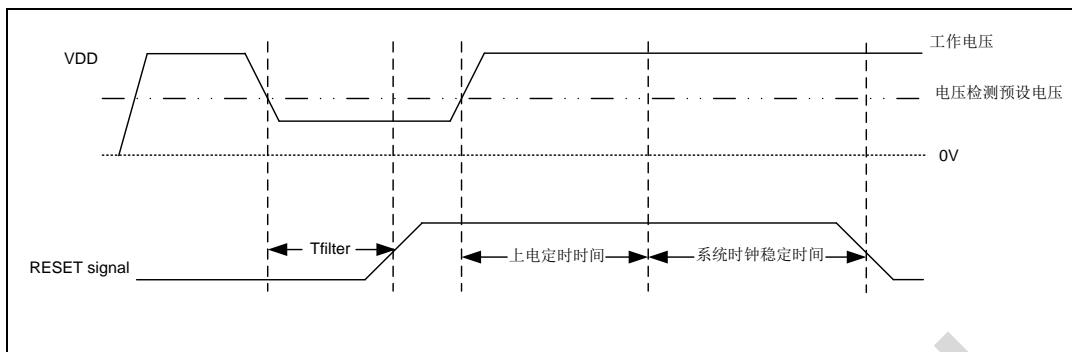


图 5-5 低电压复位时序示意图

5.3.4 外部MRSTN管脚复位

芯片提供外部 MRSTN 管脚，用于系统复位。当复位管脚输入低电平信号时，系统复位。当复位管脚处于高电平时，系统正常运行。需要注意的是，芯片配置为外部复位功能时，在系统上电完成后，外部复位管脚必须输入高电平，否则系统将一直保持在复位状态。另外，需要特别注意的是，禁止将 MRSTN 管脚直接连接到 VDD 上。外部复位滤波时间 Tfilter 为 200us 左右，可滤除外部复位管脚上脉宽小于 200us 的干扰脉冲信号。

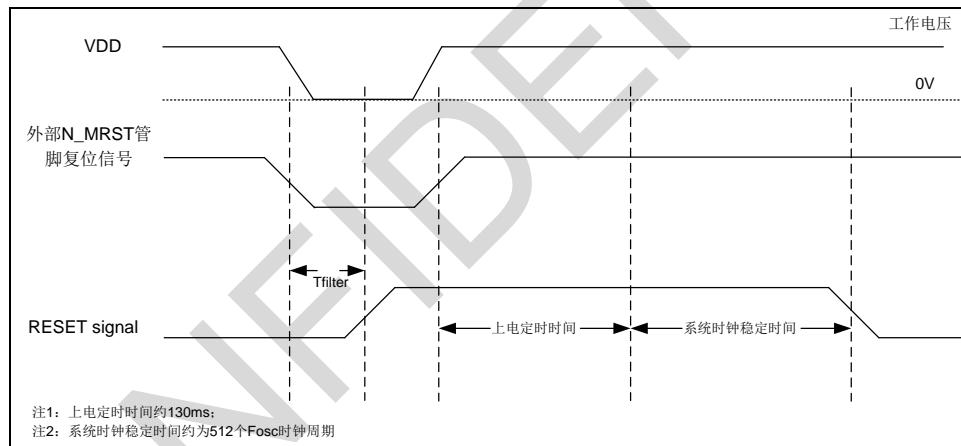


图 5-6 外部 MRSTN 管脚复位

- 注 1：当芯片配置字 MRSTEN 配置 1 为外部复位时，上电定时器，可以通过 PWRTEB 屏蔽。而当 MRSTEN 配置为 0 为数字输入输出端口时，上电定时器固定使能，但上电定时时间可通过配置字配置选择，最大为 130ms。
注 2：芯片内部 LDO 电源稳定时间约为 10ms；
注 3：上电时，系统时钟稳定时间约为 512 个 Fosc 时钟周期。

外部 MRSTN 管脚复位电路有多种，以下介绍两种比较典型的连接电路。

1. RC 复位

RC 复位电路是外部 MRSTN 管脚复位电路最简单的一种，对外界环境条件要求不高的情况下，可以采用此种连接方式。

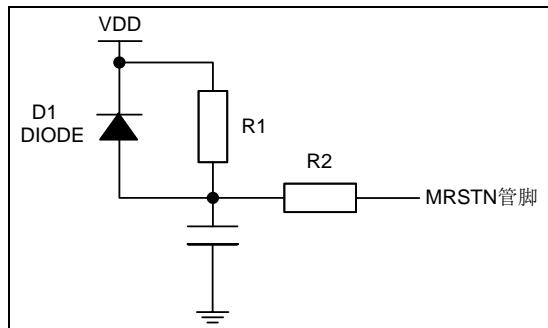


图 5-7 MRSTN 复位参考电路图 1

注：采样 RC 复位，其中 $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$ ，电容 C1 ($0.1\mu\text{F}$)，R2 为限流电阻， $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ 。

2. PNP 三极管复位

PNP 三极管复位电路适用于对电源干扰较强的的场合。

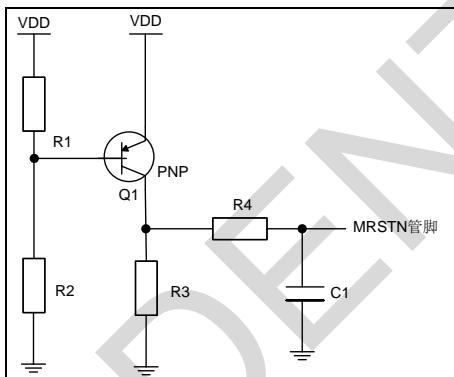


图 5-8 MRSTN 复位参考电路图 2

注：采用 PNP 三极管复位，通过 R1 ($2\text{K}\Omega$) 和 R2 ($10\text{K}\Omega$) 分压作为基极输入，发射极接 VDD，集电极一路通过 R3 ($20\text{K}\Omega$) 接地，另一路通过 R4 ($1\text{K}\Omega$) 和 C1 ($0.1\mu\text{F}$) 接地，C1 另一端作为 MRSTN 输入。

5.3.5 看门狗定时器溢出复位

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，程序无法清除看门狗，导致看门狗定时器计数溢出，产生系统复位。看门狗溢出复位后，系统重启进入正常状态。

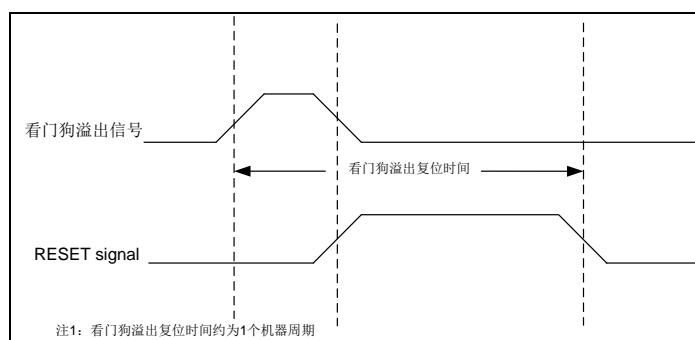


图 5-9 看门狗溢出复位

5.3.6 RST指令软件复位

整个芯片可通过执行 RST 指令进行复位，复位后，全部寄存器状态位都将被影响。

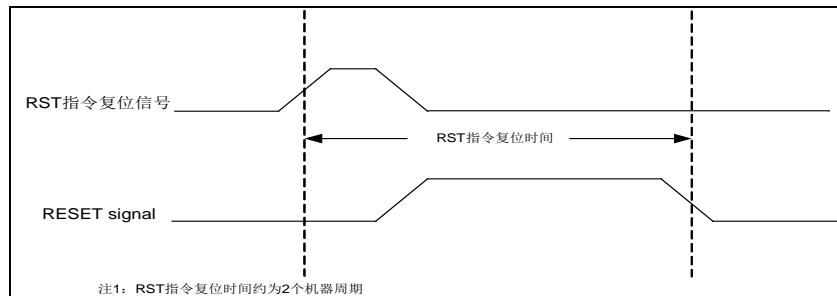


图 5-10 RST 指令软件复位

5.3.7 特殊功能寄存器

5.3.7.1 电源控制寄存器 (PWRC)

PWRC: 电源控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	LPM<1:0>		—	N_RSTI	N_TO	N_PD	N_POR	N_BOR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	1	1	1	0	0

Bit 7-6 LPM<1:0>: 休眠模式选择位
00: IDLE0 模式, 深睡眠模式
01: IDLE1 模式, 浅睡眠模式
10/11: IDLE2 模式, 快速唤醒睡眠模式

Bit 5 保留未用

Bit 4 N_RSTI: 复位指令标志位
0: 执行复位指令 (清零后必须用软件置位)
1: 未执行复位指令

Bit 3 N_TO: WDT 溢出标志位
0: WDT 计数溢出时被清零
1: 上电复位或执行 CWDT、IDLE 指令后被置 1

Bit 2 N_PD: 低功耗标志位
0: 执行 IDLE 指令后清零
1: 上电复位或执行 CWDT 指令后置 1

Bit 1 N_POR: 上电复位状态位
0: 上电复位发生 (上电复位后, 必须软件置位)
1: 无上电复位发生

Bit 0 N_BOR: 低电压复位状态位
0: 低电压复位发生 (低电压复位后, 必须软件置位)
1: 无低电压复位发生

注: LDO 为芯片内置供电模块, 给芯片内部电路模块供电, 建议客户配置为默认值。

5.3.7.2 功耗控制寄存器 (PWEN)

PWEN: 功耗控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	MRSTF	PORLOST	BORFLT<1:0>		RCEN	—
R/W	—	R/W	R/W	R	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	0	1	1

Bit 7~6 未使用

Bit 5 MRSTF: 外部复位状态位

0: 已发生外部复位 (外部复位后, 必须软件置 1)

1: 未发生外部复位

Bit 4 PORLOST: 上电复位失效状态位

0: 已发生上电复位

1: 未发生上电复位

Bit 3~2 BORFLT<1:0>: BOR 滤波时间选择位

00: 约 3 个 WDT_RC 时钟周期

01: 约 5 个 WDT_RC 时钟周期

10: 约 7 个 WDT_RC 时钟周期

11: 约 9 个 WDT_RC 时钟周期

Bit 1 RCEN: WDT 计数使能位 (建议设置为 1)

0: 关闭 WDT

1: 使能 WDT

Bit 0 保留, 需保持复位值

注: RCEN 的设置, 如果进行数据 FLASH 擦写操作时, 必须设置 RCEN=0, 关闭 WDT 功能模块, 其它情况下, 禁止关闭。

5.4 低功耗操作

5.4.1 MCU低功耗模式

芯片支持三种休眠模式：IDLE0 模式、IDLE1 和 IDLE2 模式。

◇ IDLE0 深睡眠模式

- 时钟源停振，主系统时钟暂停
- 程序暂停、同步模块暂停、异步模块运行，器件功耗降低
- 支持低功耗唤醒，唤醒时间可配，LDO 停止工作，需要稳定时间
- 所有 I/O 端口将保持进入 IDLE0 模式前的状态
- 若使能 WDT，则 WDT 将被清零并保持运行
- N_PD 位被清零，N_TO 位被置 1

◇ IDLE1 浅睡眠模式

- 时钟源停振，主系统时钟暂停
- 程序暂停、同步模块暂停、异步模块运行，器件功耗降低
- 支持低功耗唤醒，唤醒时间可配，LDO 保持工作
- 所有 I/O 端口将保持进入 IDLE1 前的状态
- 若使能 WDT，则 WDT 将被清零并保持运行
- N_PD 位被清零，N_TO 位被置 1

◇ IDLE2 快速唤醒睡眠模式

- 时钟源工作，主系统时钟暂停
- 程序暂停、同步模块暂停、异步模块运行，器件功耗降低
- 支持快速唤醒，LDO 保持工作
- 所有 I/O 端口将保持进入 IDLE2 前的状态
- 若使能 WDT，则 WDT 将被清零并保持运行
- N_PD 位被清零，N_TO 位被置 1

5.4.2 低功耗模式配置

三种低功耗模式 IDLE0、IDLE1 和 IDLE2 模式的选择，由 PWRC 寄存器中的 LPM<1:0>位控制。当 LPM = 00 时，执行 IDLE 指令，芯片进入 IDLE0 模式；当 LPM = 01 时，执行 IDLE 指令，芯片进入 IDLE1 模式；当 LPM = 10 或者 11 时，执行 IDLE 指令，芯片进入 IDLE2 模式。

LPM (PWRC<7:6>)	低功耗模式
00	IDLE0 模式
01	IDLE1 模式
10/11	IDLE2 模式

表 5-1 低功耗模式配置表

为了降低功耗，所有 I/O 管脚都应保持为 VDD 或 VSS。为了避免输入管脚悬空而引入开关电流，应在外部将高阻输入的 I/O 管脚拉为高电平或低电平，MRSTN 管脚必须处于逻辑高电平。

5.4.3 IDLE唤醒方式配置

当系统进入低功耗模式后，程序处于暂停状态，以下几种方式可将系统唤醒。

序号	唤醒方式	中断屏蔽	中断使能	中断模式	备注
1	MRSTN	—	—	—	—
2	WDT	—	—	—	WDT 溢出
3	PINTx	—	PIEx	默认/向量	—
5	I2CINT	—	I2CIE	默认/向量	

表 5-2 唤醒方式配置表

注 1：低功耗唤醒与全局中断使能无关。在低功耗模式时，若外设产生中断信号，即使默认中断模式下，全局中断使能 GIE 为 0，或向量中断模式下，高优先级中断使能 GIE 和低优先级中断使能 GIEL 均为 0，低功耗模式依然会被唤醒，只是唤醒后不会执行中断程序。

5.4.4 唤醒时序图

当唤醒事件发生后，芯片根据配置字 OSCTS<2:0>的配置执行下述操作：

当 OSCTS<2:0>配置为 INTOSCO/INTOSC 模式时：

- ◇ 在 IDLE0 模式 (LPM=00) 下，芯片需要先等待 VRwkdy 时间（由 WKDC<7: 0> 设定），此时间称为 LDO 稳定时间，最短为 64 个 Thrc16m。之后芯片主时钟运行一段 Twkdly 时间后才执行 IDLE 下一条指令，Twkdly 称为唤醒延时，唤醒延时为 4 个 Tosc 系统时钟周期；
- ◇ 在 IDLE1 模式 (LPM=01) 下，无 LDO 稳定时间，只有唤醒延时，唤醒延时为 16 个 Tosc 系统时钟周期。
- ◇ 在 IDLE2 模式 (LPM=10 或者 11) 下，无 LDO 稳定时间，无唤醒延时，可快速唤醒。

低功耗模式	计算公式	
IDLE0 模式	VRwkdy	(WKDC<7:0>) × 4 × Thrc16m
	Twkdly	4Tosc
IDLE1 模式	VRwkdy	0
	Twkdly	16Tosc
IDLE2 模式	VRwkdy	0
	Twkdly	0

表 5-3 唤醒时间计算表

注：以上唤醒时间为设计值，内部 16M 晶振未稳定时时钟频率可能会低于 16MHz，以实际芯片为准。

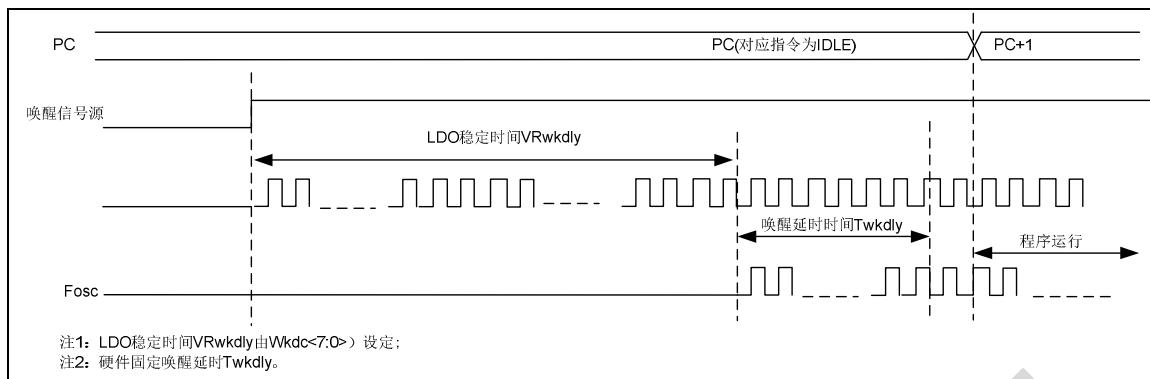


图 5-11 系统时钟为 INTHRC 时，系统唤醒 IDLE0 的时序图

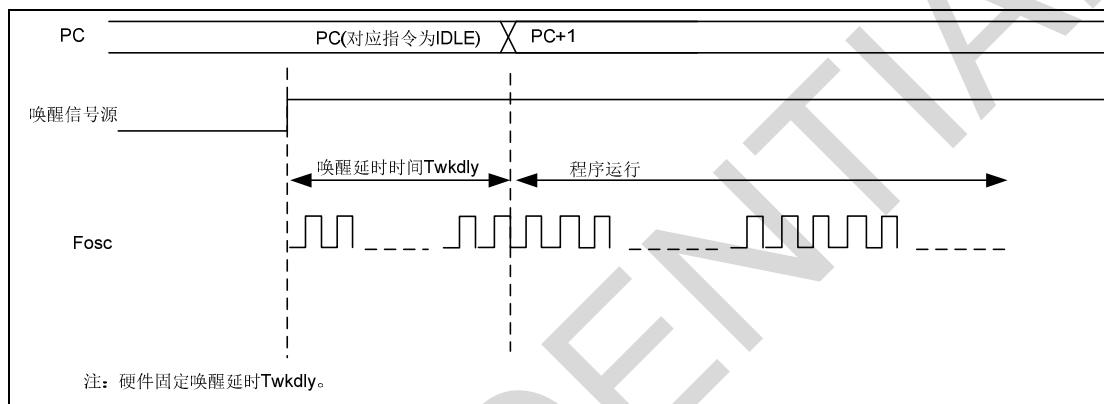


图 5-12 系统时钟为 INTHRC 时，系统唤醒 IDLE1 的时序图

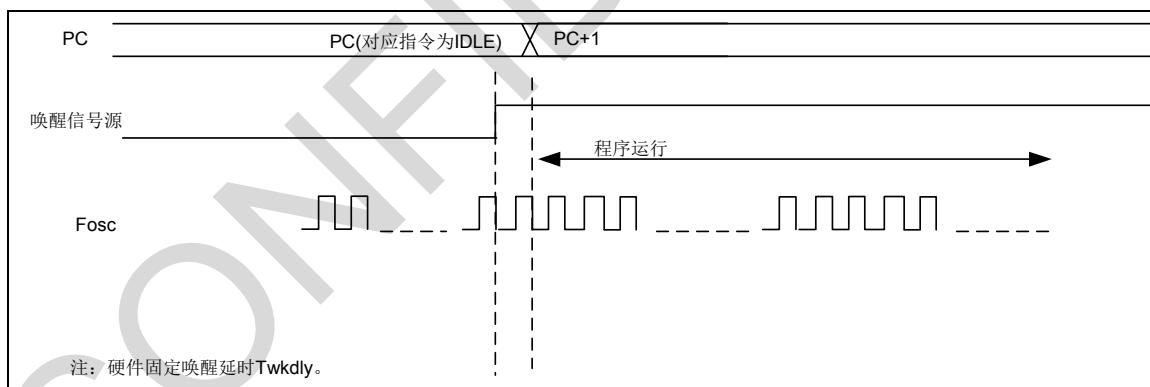


图 5-15 系统时钟为 INTHRC 时，系统唤醒 IDLE2 的时序图

5.4.5 特殊功能寄存器

5.4.5.1 唤醒延时控制寄存器（WKDC）

WKDC: 唤醒延时控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	WKDC<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1

Bit 7~0 WKDC<7:0>: 唤醒延时时间设置位

0F_H: 延时最短

.....

FF_H: 延时最长

第6章 外设

6.1 8位定时/计数器 (T8N)

6.1.1 概述

8位定时器/计数器包括定时器和计数器两种工作模式。定时器模式根据寄存器制定的定时时间进行定时，可以使定时器有选择地产生中断请求或完成其它操作。计数器模式用于对外部时钟信号 (T8NCKI) 进行计数。

◇ T8N 支持两种工作模式

- 定时器模式（时钟源为系统时钟二分频 (Fosc/2) 或者 WDT_RC (INTLRC) 时钟）
- 同步计数器模式（时钟源为外部输入时钟 T8NCKI）

◇ T8N 支持以下功能组件

- 8位预分频器（无实际物理地址，不可读写）
- 8位计数器寄存器 (T8N)
- 8位控制寄存器 (T8NC)

◇ 中断和暂停

- 支持溢出中断标志 (T8NIF)
- 支持中断处理
- 在 IDLE 模式下，T8N 暂停

6.1.2 内部结构图

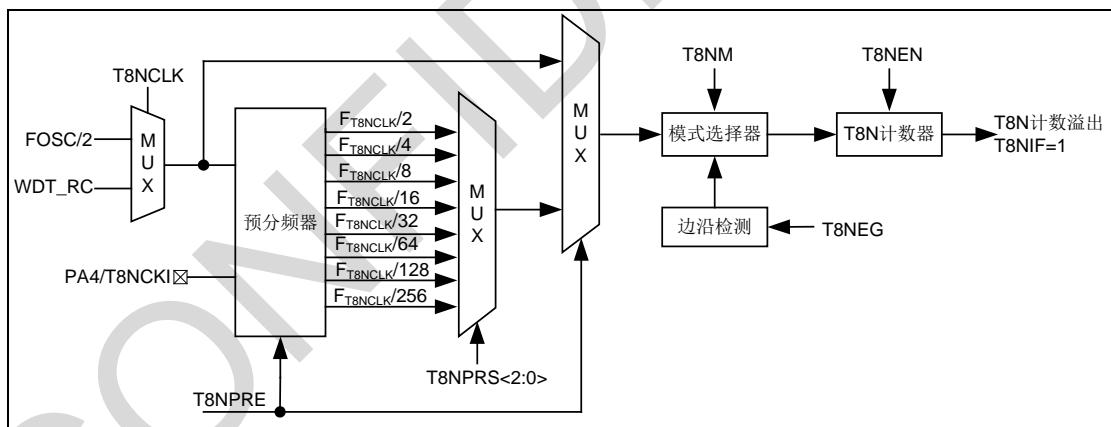


图 6-1 T8N 内部结构图

6.1.3 预分频器

预分频器可以提供定时器/计数器一个更长的溢出周期。当 T8NC 寄存器中的 T8NPRE 为“1”时，使能 T8N 预分频器。任何对 T8N 计数器的写操作都会清零预分频器，改写后立即更新，但不影响预分频器的分频比设置，预分频器的计数值无法读写。预分频器的分频比可通过 T8NC 寄存器中的 T8NPRES <2:0> 位进行设置，预分频比范围为 1:2~1:256。

注 1：当使用 WDT_RC 为计数时钟时，必须使能预分频控制位。

注 2：建议不要在 T8N 计数时，改写 T8N 计数器和预分频值，会影响改写后的第一次计数的时间。

工作模式	T8NPRE	T8NPRS<2:0>	T8N 计数时钟	
			T8NCLK=0	T8NCLK=1
定时器模式	0	—	Fosc/2	—
	1	000	(Fosc/2) /2	Fwdt_rc /2
	1	001	(Fosc/2) /4	Fwdt_rc /4
	1	010	(Fosc/2) /8	Fwdt_rc /8
	1	011	(Fosc/2) /16	Fwdt_rc /16
	1	100	(Fosc/2) /32	Fwdt_rc /32
	1	101	(Fosc/2) /64	Fwdt_rc /64
	1	110	(Fosc/2) /128	Fwdt_rc /128
	1	111	(Fosc/2) /256	Fwdt_rc /256
工作模式	T8NPRE	T8NPRS<2:0>	T8N 计数时钟	
计数器模式	0	—	T8NCKI	
	1	000	T8NCKI /2	
	1	001	T8NCKI /4	
	1	010	T8NCKI /8	
	1	011	T8NCKI /16	
	1	100	T8NCKI /32	
	1	101	T8NCKI /64	
	1	110	T8NCKI /128	
	1	111	T8NCKI /256	

表 6-1 T8N 预分频器配置表

6.1.4 工作模式

T8N 有两种工作模式，定时器模式和计数器模式，通过 T8NM 进行选择。定时器和计数器计数模式均支持预分频器。配置为定时器模式时，T8N 计数器的时钟源可通过 T8NC 寄存器中的 T8NCLK 位选择为系统时钟 2 分频 (Fosc/2) 或 WDT_RC；配置为计数器模式时，T8N 计数器的时钟源为经二分频后的系统时钟 Fosc/2 同步的外部输入时钟 T8NCKI，因此 T8NCKI 输入时钟信号的高电平和低电平时间都至少为一个机器周期。通过 T8NC 寄存器中的 T8NEG 位选择外部时钟的计数边沿为上升沿或下降沿。T8NCKI 所在 IO 端口必须配置为数字输入状态。

T8NM	T8NCLK	工作模式	时钟源
0	0	定时器模式	Fosc/2
0	1	定时器模式	WDT_RC
1	0	同步计数器模式	T8NCKI

表 6-2 T8N 工作模式配置表

6.1.5 定时器模式

T8N 计数器为递增计数，计数值由 FF_H 变为 00_H 时，T8N 计数器发生溢出并重新开始计数。T8N 计数器发生溢出时，中断标志 T8NIF 位被置“1”，产生 T8N 溢出中断。在 CPU

进入休眠模式后，T8N 模块不工作，因此不产生中断。

当 T8N 配置为定时器模式时，若禁止预分频器，则 T8N 计数器的时钟只能选择为系统时钟二分频 ($F_{osc}/2$)，不能选择为 WDT_RC；若使能预分频器，分频器对 $F_{osc}/2$ 或 WDT_RC 进行分频，此时，T8N 计数器的计数时钟为分频后的时钟。

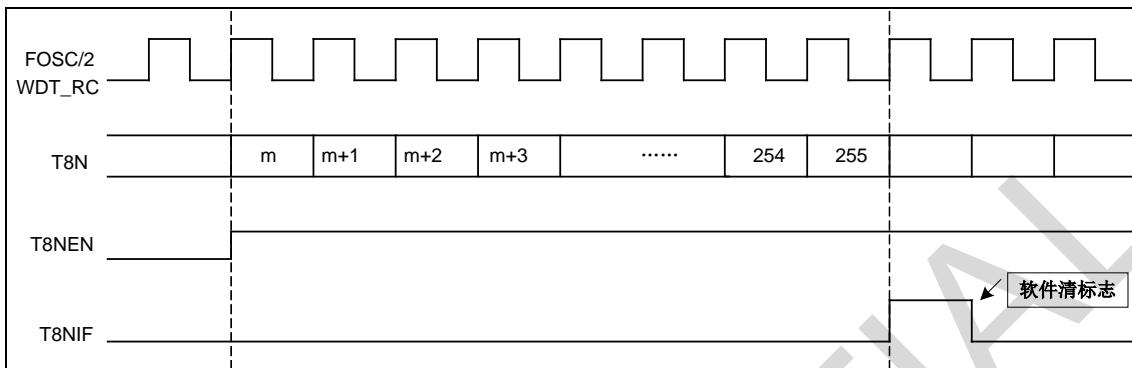


图 6-2 定时器模式时序图

6.1.6 同步计数器模式

当 T8N 配置为同步计数器模式时，若禁止预分频器，T8N 计数器的时钟为外部输入时钟 T8NCKI，内部相位时钟 p2 将对时钟 T8NCKI 进行同步。所以 T8NCKI 保持高电平或者低电平的时间至少为一个机器周期。通过设置 T8NEG (T8NC<4>) 选择外部时钟的计数边沿为上升沿或下降沿。

同样，同步计数器模式也支持预分频器对外部时钟 T8NCKI 进行分频。并且，T8NCKI 复用的 IO 端口必须配置为数字输入状态。

T8N 计数器为递增计数，计数值由 FFH 变为 00H 时，T8N 计数器发生溢出并重新开始计数。T8N 计数器发生溢出时，中断标志 T8NIF 位被置“1”，产生 T8N 溢出中断。在 CPU 进入休眠模式后，T8N 模块不工作，因此不产生中断。

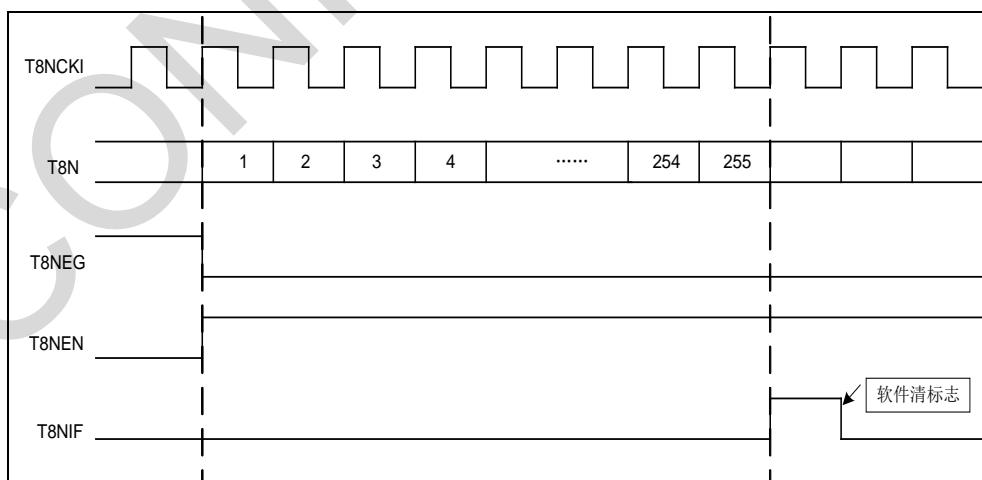


图 6-3 计数器模式时序图 (T8NEG=0, T8NCKI 上升沿计数)

6.1.7 特殊功能寄存器

8 位定时器/计数器 T8N 由两个寄存器控制，一个 8 位计数器寄存器 T8N 和一个控制寄

存器 T8NC。T8N 寄存器用于存放计数值，T8NC 控制寄存器用于控制 T8N 的使能、T8N 的模式选择、T8NCKI 计数边沿选择、预分频器使能位以及预分频器分频比选择。

6.1.7.1 T8N计数器寄存器 (T8N)

T8N: T8N 计数器寄存器								
Bit	7	6	5	4	3	2	1	0
Name	T8N <7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 T8N <7:0>: 8 位 T8N 计数值

6.1.7.2 T8N控制寄存器 (T8NC)

T8NC: T8N 控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	T8NEN	T8NCLK	T8NM	T8NEG	T8NPRE	T8NPRS<2:0>		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7 T8NEN: T8N 模块使能位

- 0: 关闭
- 1: 使能

Bit 6 T8NCLK: T8N 定时时钟源选择位

- 0: 系统时钟二分频 Fosc/2
- 1: WDT_RC 时钟（必须同时使能 T8N 的预分频控制位，即 T8NPRE=1）

Bit 5 T8NM: T8N 模式选择位

- 0: 定时器模式
- 1: 同步计数器模式

Bit 4 T8NEG: T8NCKI 同步计数边沿选择位

- 0: T8NCKI 上升沿计数
- 1: T8NCKI 下降沿计数

Bit 3 T8NPRE: 预分频器使能位

- 0: 禁止
- 1: 使能

Bit 2~0 T8NPRS <2:0>: 预分频器分频比选择位

- 000: 1:2
- 001: 1:4
- 010: 1:8
- 011: 1:16
- 100: 1:32
- 101: 1:64
- 110: 1:128
- 111: 1:256

6.2 16 位高级定时/计数器 (ADTIM1)

6.2.1 概述

高级控制定时器 (ADTIM1) 由一个可编程预分频器驱动的 16 位自动重载计数器构成，可用于多种用途，包含测量输入信号脉冲长度 (输入捕获) 或产生输出波形 (输出比较/PWM/带死区的互补 PWM/单脉冲模式)，可使用定时预分频器与时钟控制预分频器来调校脉冲长度与波形周期，由数微秒至数毫秒。

高级控制(ADTIM1)定时器的时钟源可以选择内部时钟或者外部时钟。

高级控制定时器特征包含：

- ◇ 16 位向上，向下，向上/下计数自动加载计数器
- ◇ 16 位可编程预分频器，其计数器时钟频率分频比允许为 1 到 65536 间的任何值 (在运行中)
- ◇ 高达四个独立通道
 - 输入捕获
 - 输出比较
 - PWM 产生 (边沿与中央对齐模式)
 - 六步 PWM 输出
 - 单脉冲模式输出
 - 支持三组带互补 PWM 输出，并且死区时间可编程设置
- ◇ 互补输出支持可编程死区 (**dead-time**) 寄存器
- ◇ 在指定的计数器周期之后，可更新定时寄存器的重复计数器，
- ◇ 暂停输入以将定时器输出信号置于重置状态或是已知状态
- ◇ 中断产生以下事件：
 - 更新：计数器上溢/下溢，计数器初始化 (通过软件或内部/外部触发)
 - 触发事件 (计数器起始，停止，初始化或由内部/外部触发计数)
 - 输入捕获 (捕获寄存器)
 - 输出比较 (计数寄存器配对比较寄存器)
 - 刹车信号输入
- ◇ 支持编码器接口模式
- ◇ 外部时钟触发输入

注：本节中 TIMx 代表 TIM1；文中的 n 代表 1、2、3、4。

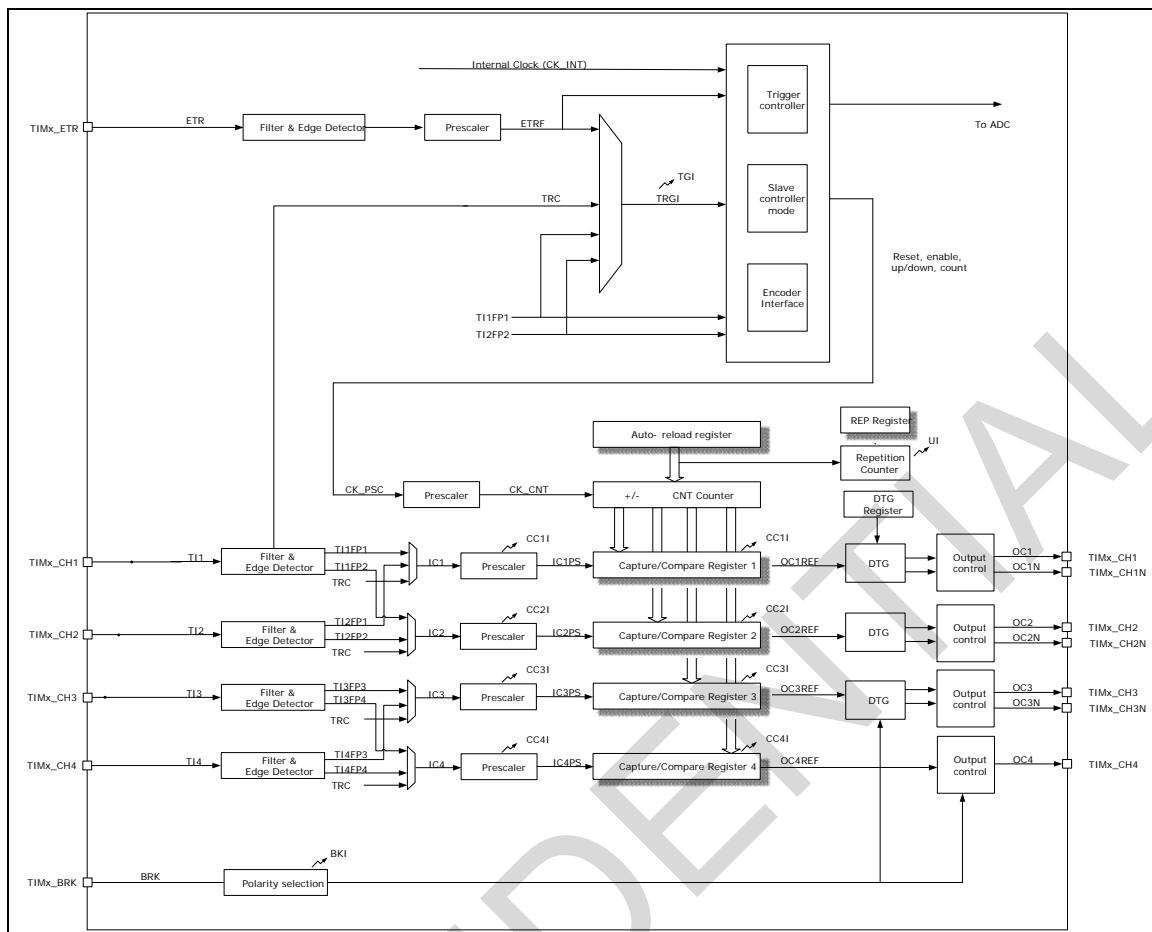


图 6-4 高级定时器内部结构图

6.2.2 定时器时基单元

编程高级控制定时器主区块为一组自动加载寄存器的 16 位计数器。计数器可递增、递减或同时增减计算。计数时钟可被预分频器分频。

计数器、预分频器、自动重载寄存器和重复计数寄存器都可由软件读写。

时基寄存器包含：

- ◇ 16 位的计数寄存器(TIMx_CNT)
- ◇ 16 位的预分频寄存器(TIMx_PSC)
- ◇ 16 位的自动重载寄存器(TIMx_ARR)
- ◇ 8 位的重复计数寄存器(TIMx_RCR)

当 TIMx_CR1 寄存器的自动预载使能位(ARPE)为 1 时, 写入自动重载寄存器(TIMx_ARR)的值存入预载寄存器, 当下次发生更新事件 (UEV) 时, 预载寄存器的内容被传输至影子寄存器。

当 TIMx_CR1 寄存器的自动预载使能位(ARPE)为 0 时, 写入自动重载寄存器(TIMx_ARR)的值立即传输至影子寄存器。

预载寄存器和影子寄存器为芯片内部缓冲寄存器，无实际物理地址。

当 **TIMx_CR1** 寄存器的更新事件使能位 **UDIS** 为 0 时，使能更新事件，则以下任一事件会产生一个更新事件：

- 计数器到达上溢或向下计数时的下溢
- 软件设置 **UG** 位，由软件产生
- 从模式控制器产生的更新

当 **TIMx_CR1** 寄存器的更新事件使能位 **UDIS** 为 1 时，禁止更新事件产生。

计数器是由预分频器输出的计数时钟 **CK_CNT** 计时，仅当 **TIMx_CR1** 寄存器的计数器使能位(**CEN**)被设置 1 时，计数器计数被使能(参照从模式控制器描述以取得更多计数器使能信息)。

6.2.2.1 ADTIM1 寄存器的写操作

16 位的计数寄存器(**TIMx_CNT**)的写操作没有缓冲，并且可以在任何时候写 **TIMx_CNTL** 和 **TIMx_CNTH**。建议不要在计数器正常工作时改写计数寄存器的值，以免造成工作异常。

16 位的自动重载寄存器(**TIMx_ARR**)的写操作，要求先写高 8 位寄存器 **TIMx_ARRH**，再写低 8 位寄存器 **TIMx_ARRL**。高 8 位寄存器的值会在写低 8 位寄存器值时同时更新写入 **TIMx_ARR** 寄存器。

16 位的捕捉/比较寄存器(**TIMx_CCRx**)的写操作，要求先写高 8 位寄存器 **TIMx_CCRxH**，再写低 8 位寄存器 **TIMx_CCRxL**。高 8 位寄存器的值会在写低 8 位寄存器值时同时更新写入 **TIMx_CCRx** 寄存器。

6.2.2.2 预分频器

预分频器可将计数器时钟频率依照 1 至 65536 间的任意因子分频。基于一 16 位寄存器控制的 16 位计数器(**TIMx_PSC** 寄存器内)。因其具有缓冲功能，可于运行时被变更。新的预分频器参数于下一次更新事件到达前被采用。预分频器被设定后，一旦预分频计数器到达 0，计数器则会向上或向下计数，此外计数器会随着时钟一同更新。图 6-5 与图 6-6 标示 **TIMx_ARR=0xFC**，当预分频器于运作时变更计数器预分频器的行为范例。

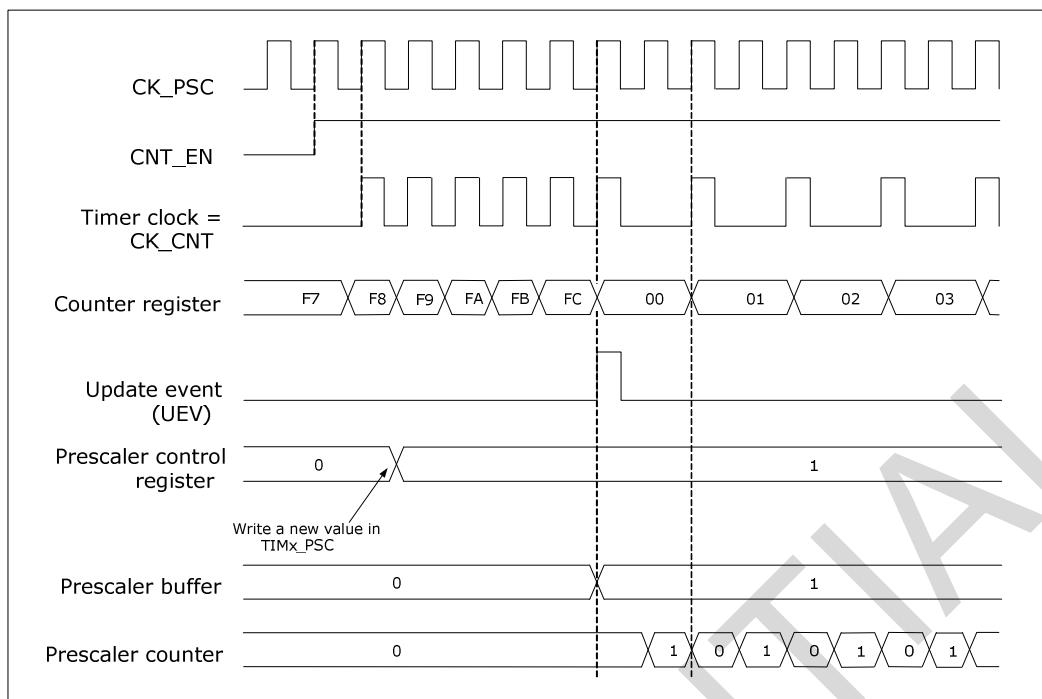


图 6-5 预分频器分频比由 1 变更为 2 的计数器时序图

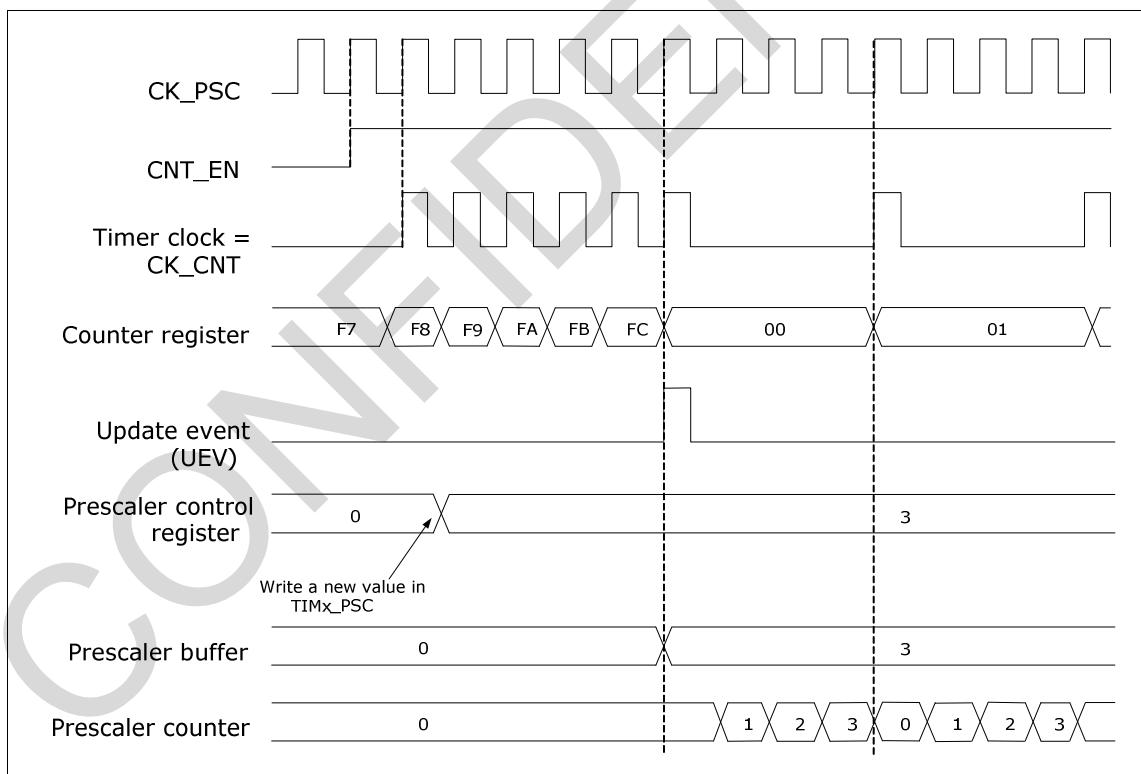


图 6-6 预分频器分频比由 1 变更为 4 的计数器时序图

6.2.2.3 16 位计数器模式

6.2.2.4 递增计数模式

在递增计数模式，计数器由 0 计算至自动加载值(TIMx_ARR 寄存器内容)，再由 0 重新开始计数并产生一个计数器上溢事件。

重复计数后产生。此外更新事件会于每次计数器上溢时产生。

更新事件可于每次计数器上溢或设置 **TIMx_EGR** 寄存器 **UG** 位产生(通过软件或从模式控制器)。

UEV 事件可由软件通过设置 **TIMx_CR1** 寄存器的 **UDIS** 位以关闭。这是为了避免更新写入新值至预载寄存器时，影子寄存器也一并更新。**UDIS** 位被写为 1 时，都不会有更新事件产生。但在本应该产生更新事件的时刻，计数器会被清零重新计数，预分频器的计数也会被清零并重新开始计数(但预分频比不会变更)。

当更新事件发生时，所有寄存器皆被更新且更新标志(**TIMx_RIF** 寄存器 **UIF** 位)皆被设定：

- ◇ 重复计数器随 **TIMx_RCR** 寄存器内容重新加载
- ◇ 自动重载影子寄存器被更新为预载值(**TIMx_ARR**)
- ◇ 预分频器的缓冲器随着预载值重新加载(**TIMx_PSC** 寄存器的内容)

下列图表为计数器在 **TIMx_ARR=0x36** 时对应不同时钟频点的行为范例：

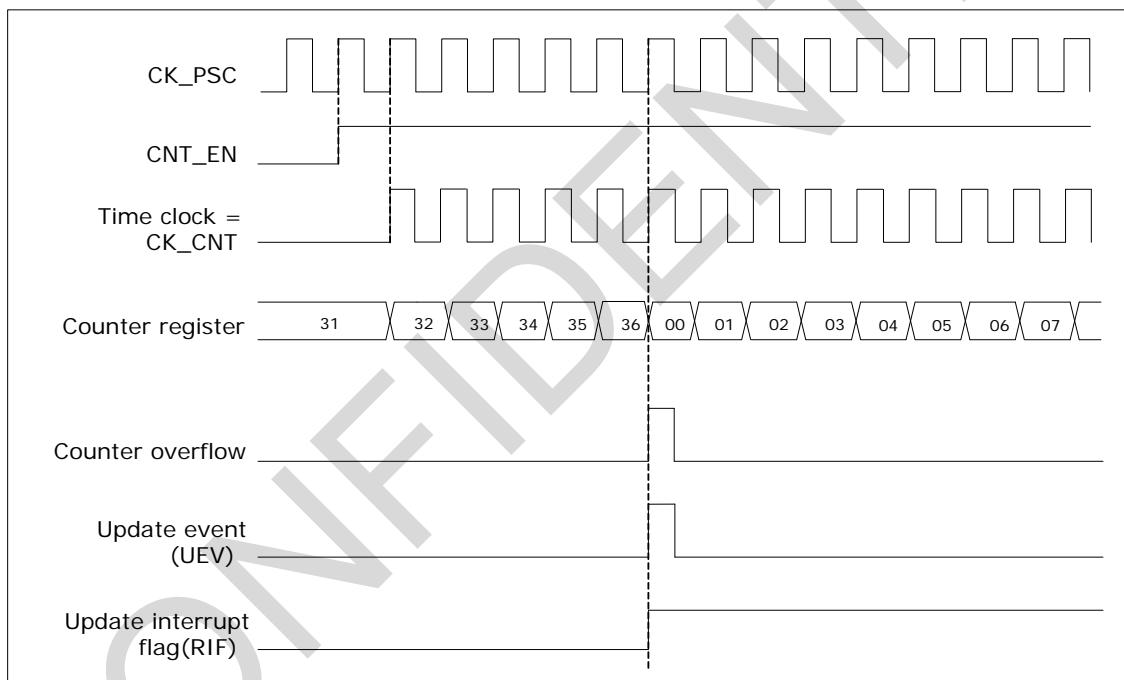


图 6-7 计数器时序图（内部时钟除以 1）

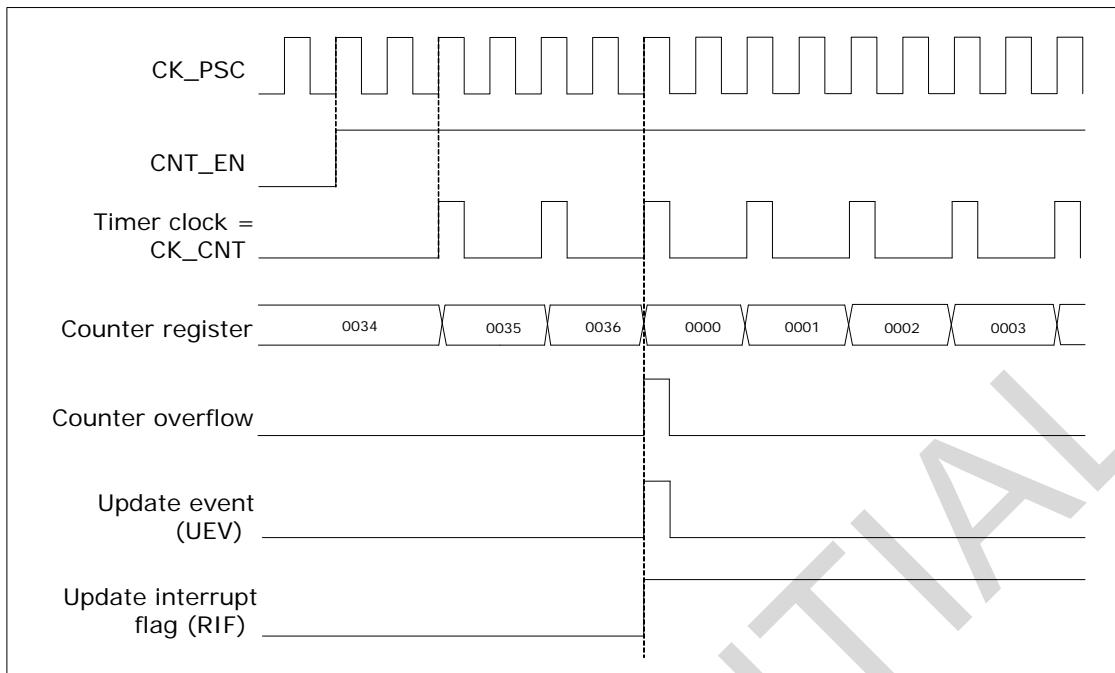


图 6-8 计数器时序图（内部时钟除以 2）

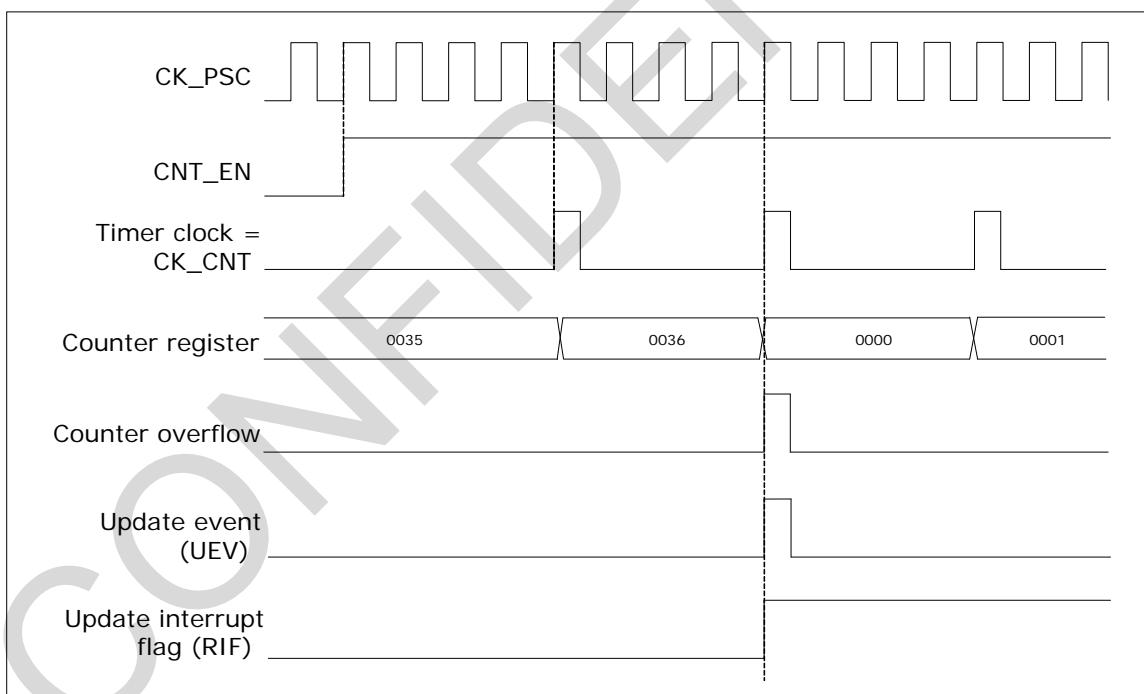


图 6-9 计数器时序图（内部时钟除以 4）

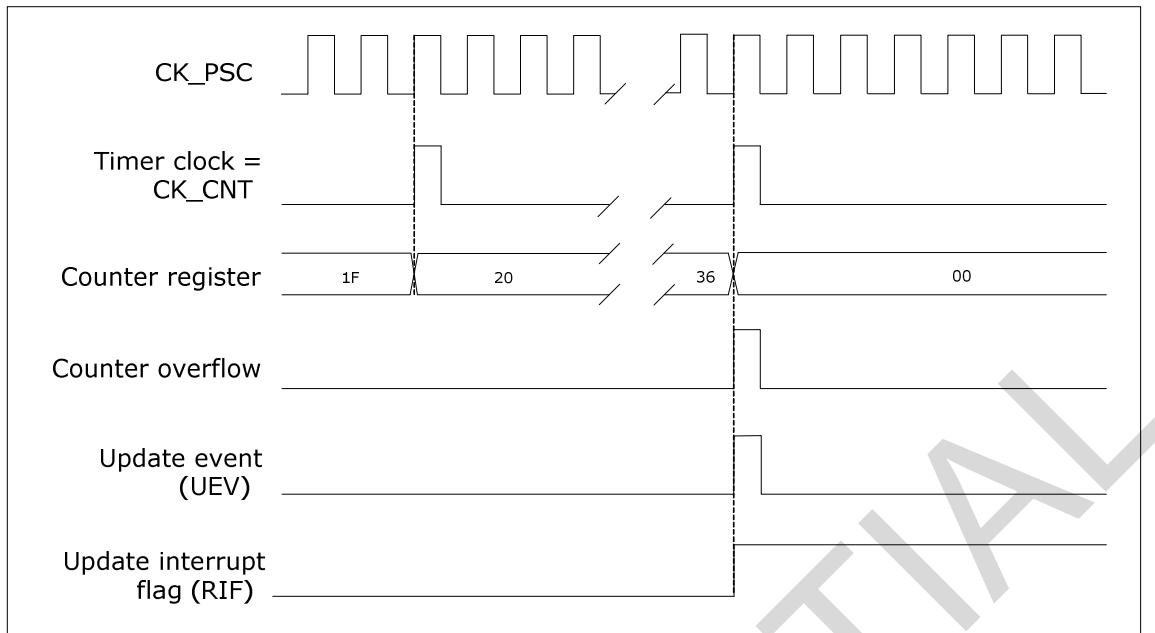


图 6-10 计数器时序图（内部时钟除以 N）

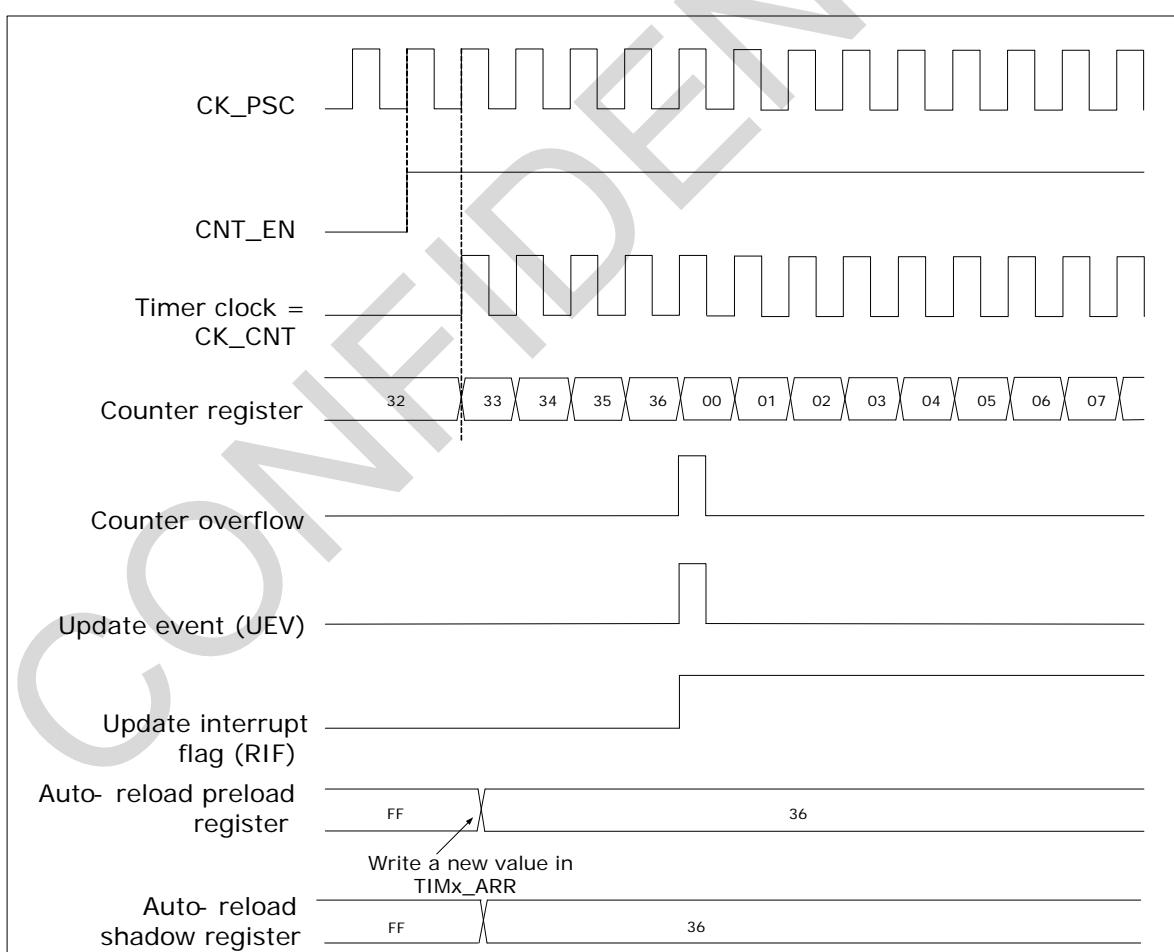


图 6-11 计数器时序图（当 ARPE=0 更新事件(TIMx_ARR 无预载)）

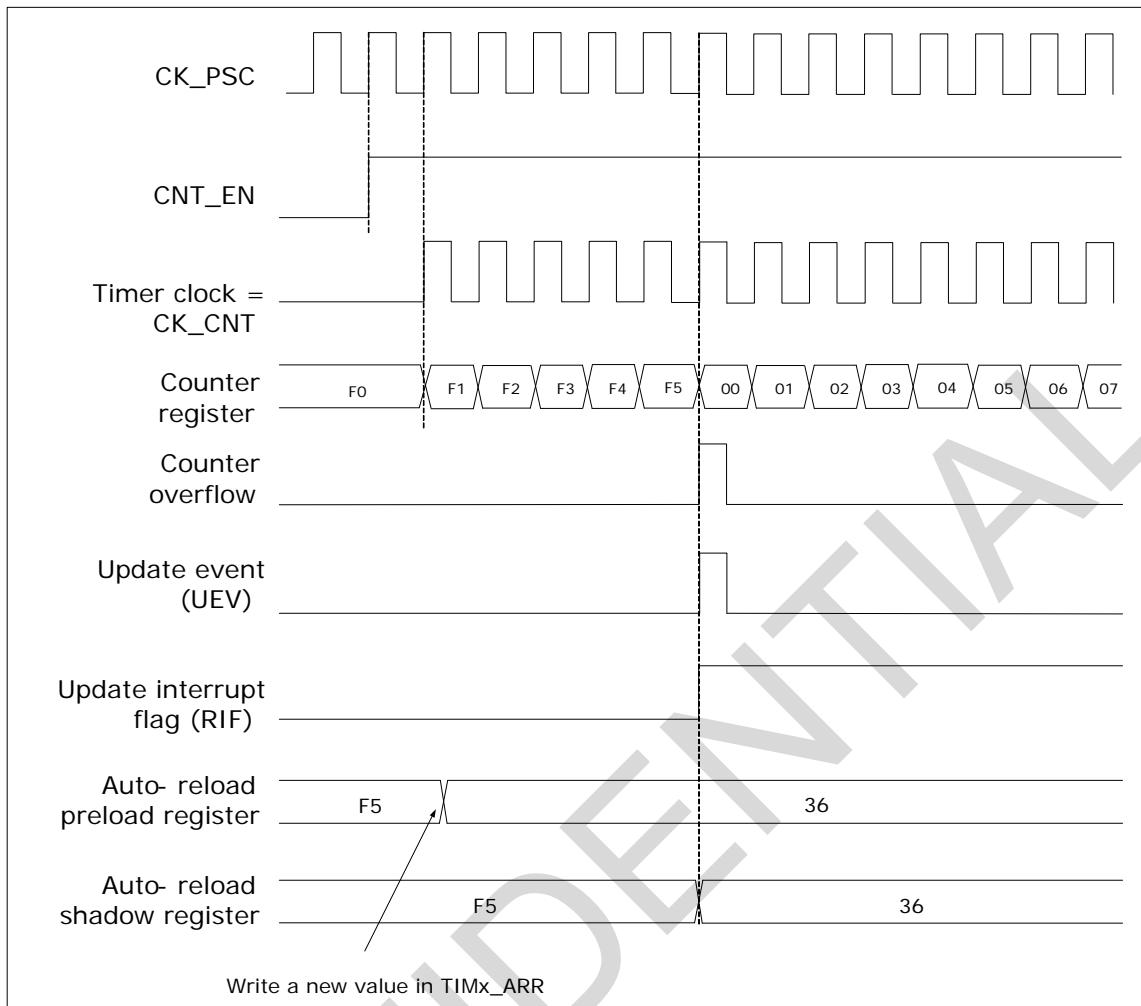


图 6-12 计数器时序图 (当 ARPE=1 更新事件(TIMx_ARR 预载))

6.2.2.5 递减计数模式

在递减计数模式，计数器由自动加载值(TIMx_ARR 寄存器内容)开始递减计算至 0，再由自动加载值重新开始计数，并产生计数下溢事件。

若重复计数器被使用，更新事件(UEV)会在向下计数依重复计数器(TIMx_RCR)编程次数重复计数后产生。此外更新事件会于每次计数器下溢产生。

设定 TIMx_EGR 寄存器的 UG 位(通过软件或从模式控制器)亦可产生更新事件。

UEV 事件可由软件通过设置 TIMx_CR1 寄存器的 UDIS 位以关闭。这是为了避免更新写入新值至预载寄存器时，影子寄存器也一并更新。UDIS 位被写为 1，都不会有更新事件产生。但在本应该产生更新事件的时刻，计数器会清零重新开始计数，预分频器的计数也会被清零并重新开始计数(但预分频比不会变更)。

当更新事件发生时，所有寄存器被更新且更新状态(TIMx_RIF 寄存器的 UIF 位)也被设置：

- ◇ 寄存器重复计数器重载 TIMx_RCR 寄存器内容
- ◇ 预分频器的缓冲重载预分频值(TIMx_PSC 寄存器内容)
- ◇ 自动重载行为寄存器依预载值更新(TIMx_ARR 寄存器内容)

注：自动重载会在计数器重载前更新，因此下一周期将会是预期值。

下列图表为计数器在 $\text{TIMx_ARR}=0x36$ 时对应不同时钟频点的行为范例。

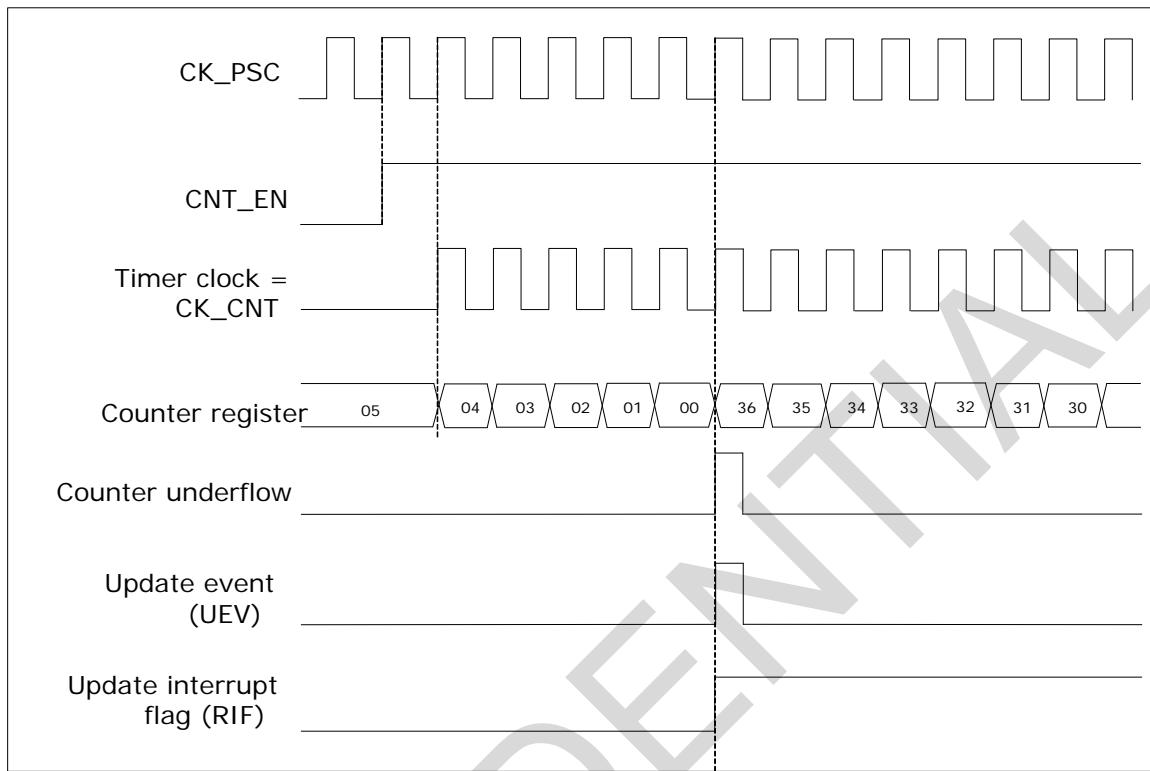


图 6-13 计数器时序图（内部时钟分频因子为 1）

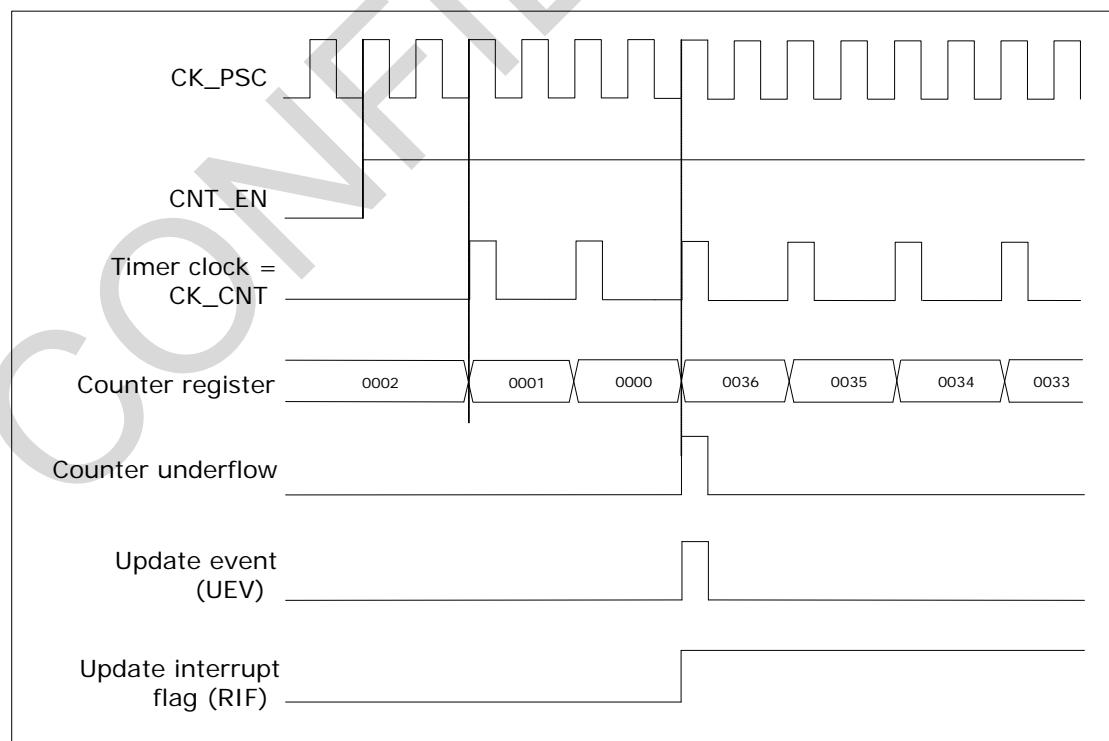


图 6-14 计数器时序图（内部时钟分频因子为 2）

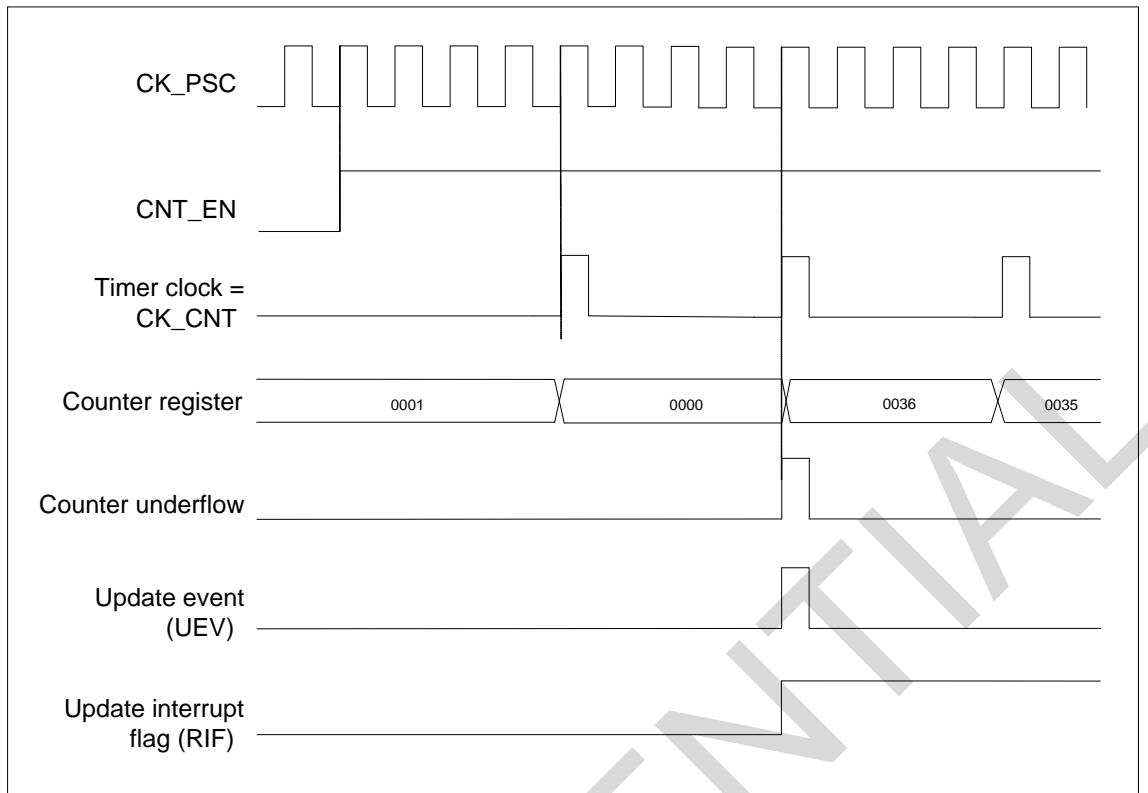


图 6-15 计数器时序图（内部时钟分频因子为 4）

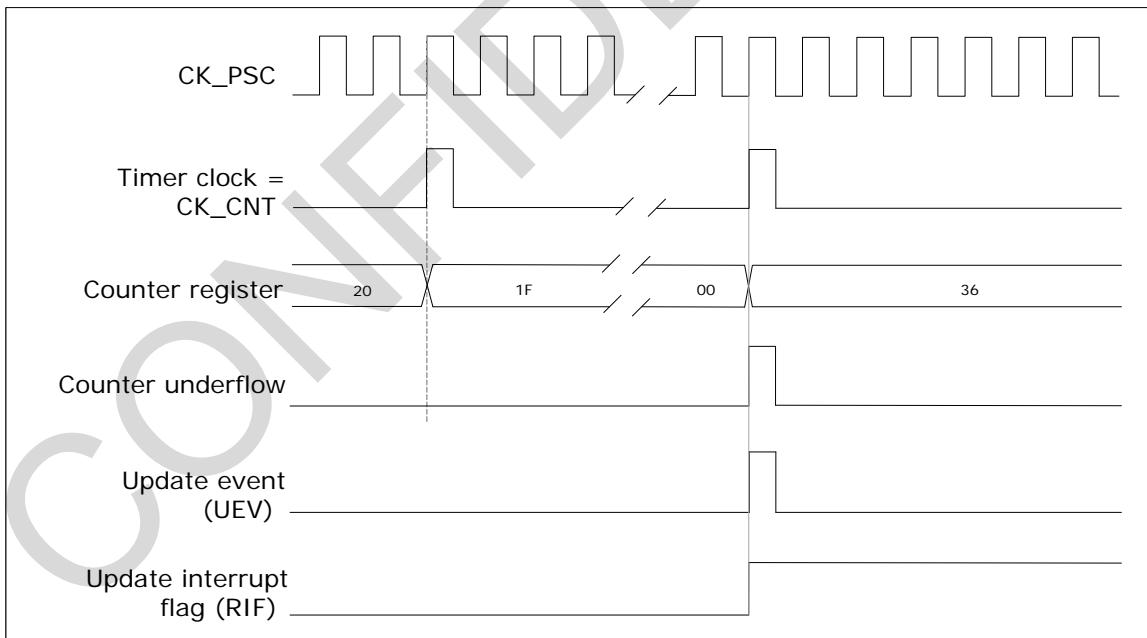


图 6-16 计数器时序图（内部时钟分频因子 N ）

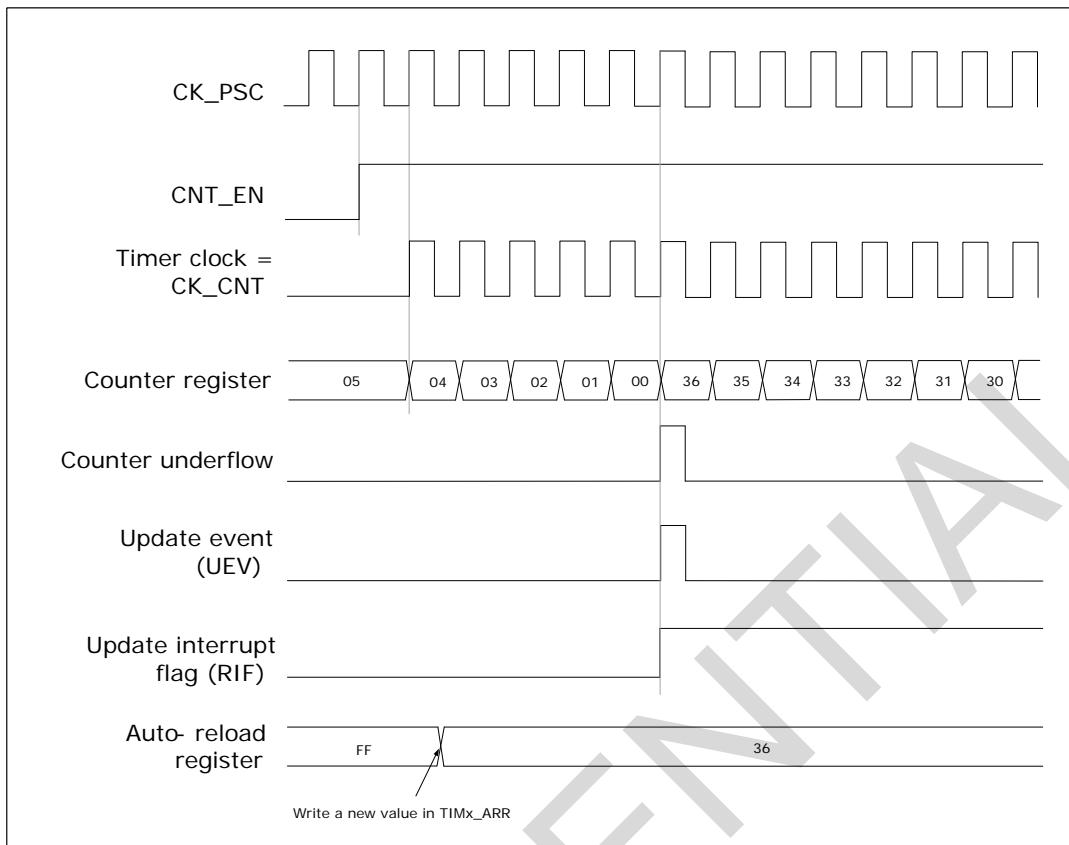


图 6-17 计数器时序图（重复计数器未使用时的更新事件）

6.2.2.5 中心对齐模式

在中心对齐模式时，计数器由 0 开始递增计算至自动重载值(TIMx_ARR 寄存器的内容)减 1，产生计数器上溢事件，接着由自动重载值递减计数至 1，并产生计数器下溢事件。然后计数器重新由 0 开始计算。

TIMx_CR1 寄存器的 CMS 位不等于'00'时，中心对齐模式方可作用。各通道的输出比较中断标志输出，当计数器递减计算(中心对齐模式 1,CMS="01")，计数器递增计算(中心对齐模式 2,CMS="10")，计数器递增/递减计算(中心对齐模式 3,CMS="11")。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位将无法被写。将由硬件对其进行更新并给予计数器的计数方向。

更新事件可在每次计数器的上溢/下溢产生、也可通过设置 TIMx_EGR 寄存器的 UG 位(通过软件置 1 或使用从模式控制器的复位模式)产生一更新事件。此时计数器将重新由 0 开始计数，预分频器的计数也会被清零并重新开始计数。

UEV 更新事件可通过设置软件或 TIMx_CR1 寄存器的 UDIS 位关闭。这是为了避免写入新值至预载寄存器时影子寄存器也一并更新。UDIS 位被写入 1 时，都不会有更新事件发生。但计数器仍由现有自动加载值向上或向下计数。这是为了避免在发生捕获事件并清除计数器时，同时产生更新与捕获中断。

当更新事件发生时，所有寄存器皆更新且状态(TIMx_RIF 寄存器的 UIF 位)更新为：

- ◇ 重复计数器重载 TIMx_RCR 寄存器内容

- ◇ 预分频缓冲器重载预载值(TIMx_PSC 寄存器内容)
- ◇ 自动重载作用寄存器藉由预载值更新(TIMx_ARR 寄存器内容)

注：若更新源为计数器上溢，自动重载会在计数器重载前更新，因此下一周期为预期值(计数器载入新值)。

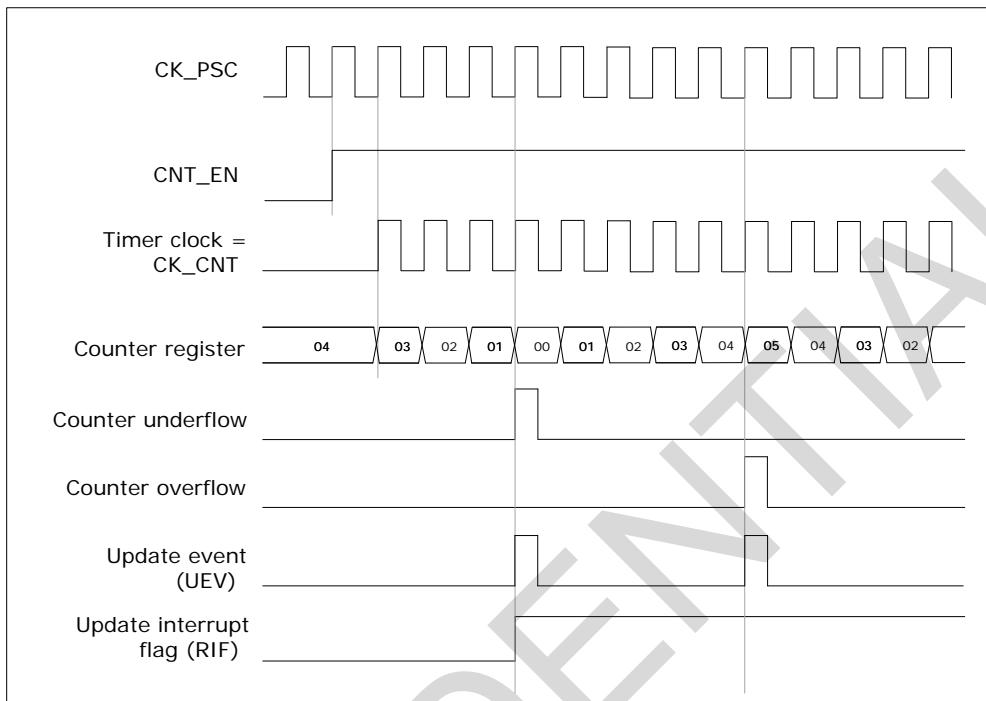


图 6-18 计数器时序图，内部时钟分频因子为 1,TIMx_ARR=0x05

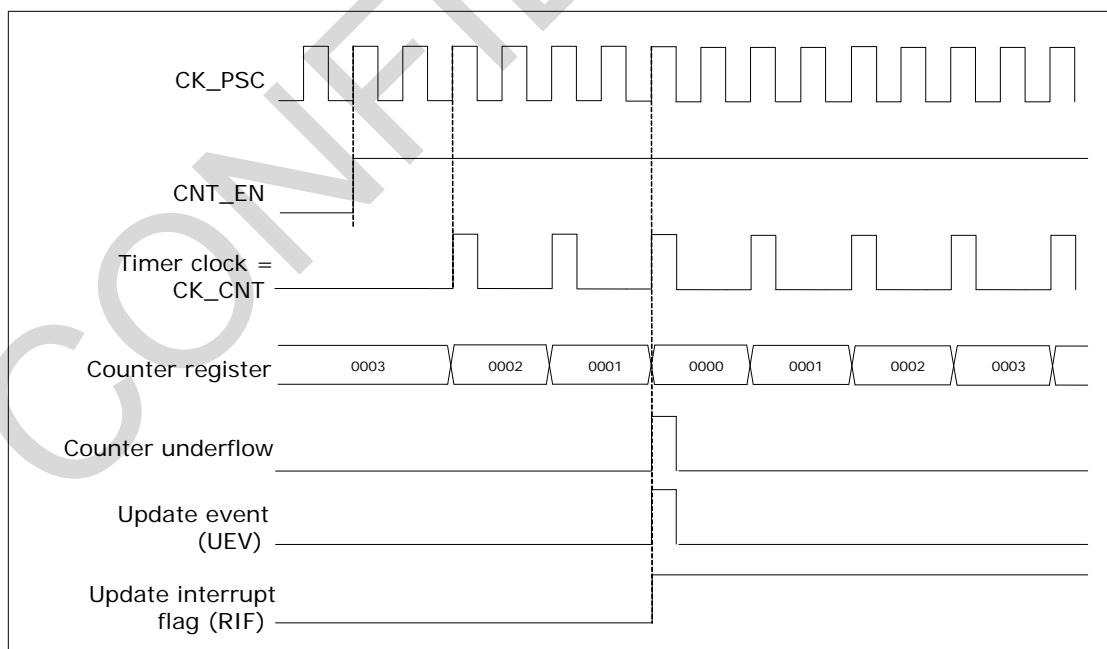


图 6-19 计数器时序图，内部时钟分频因子为 2

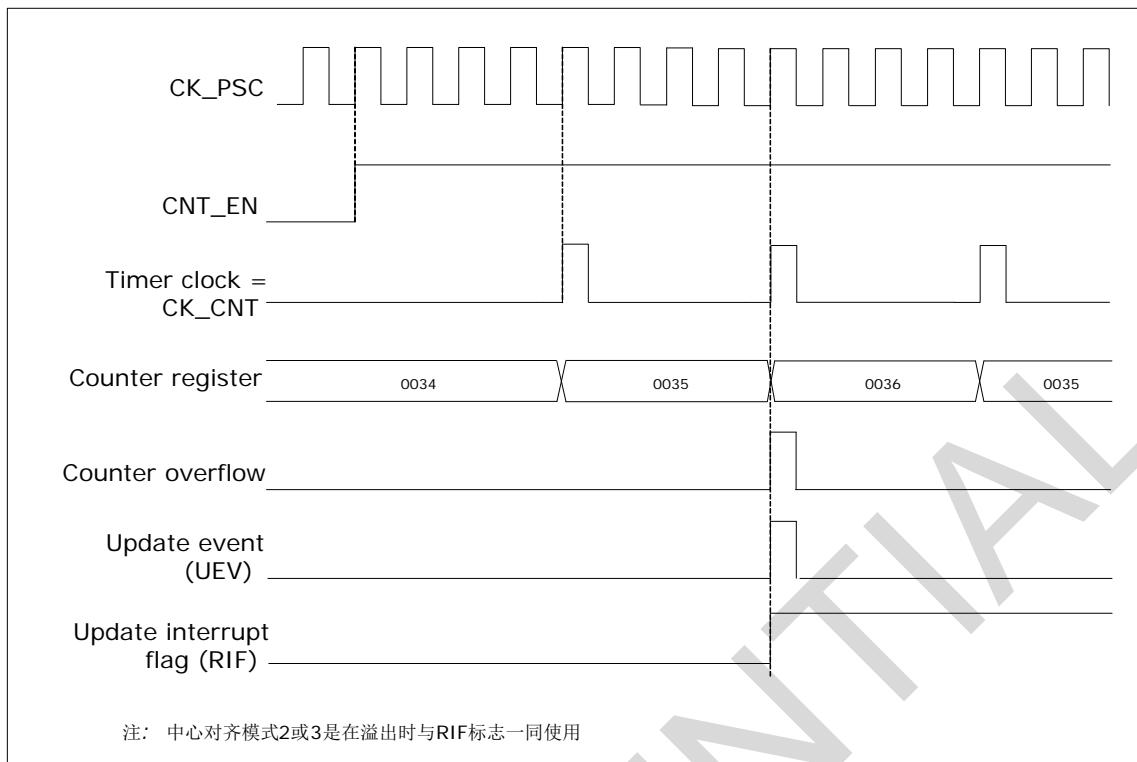


图 6-20 计数器时序图，内部时钟分频因子为 4,TIMx_ARR=0x36

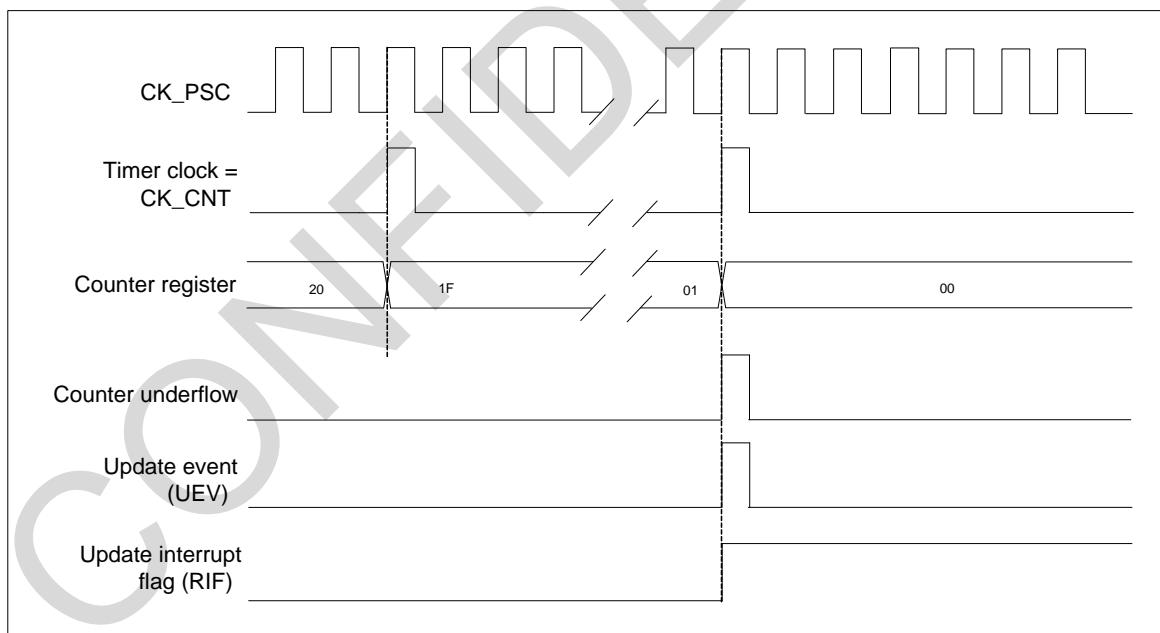


图 6-21 计数器时序图，内部时钟分频因子为 N

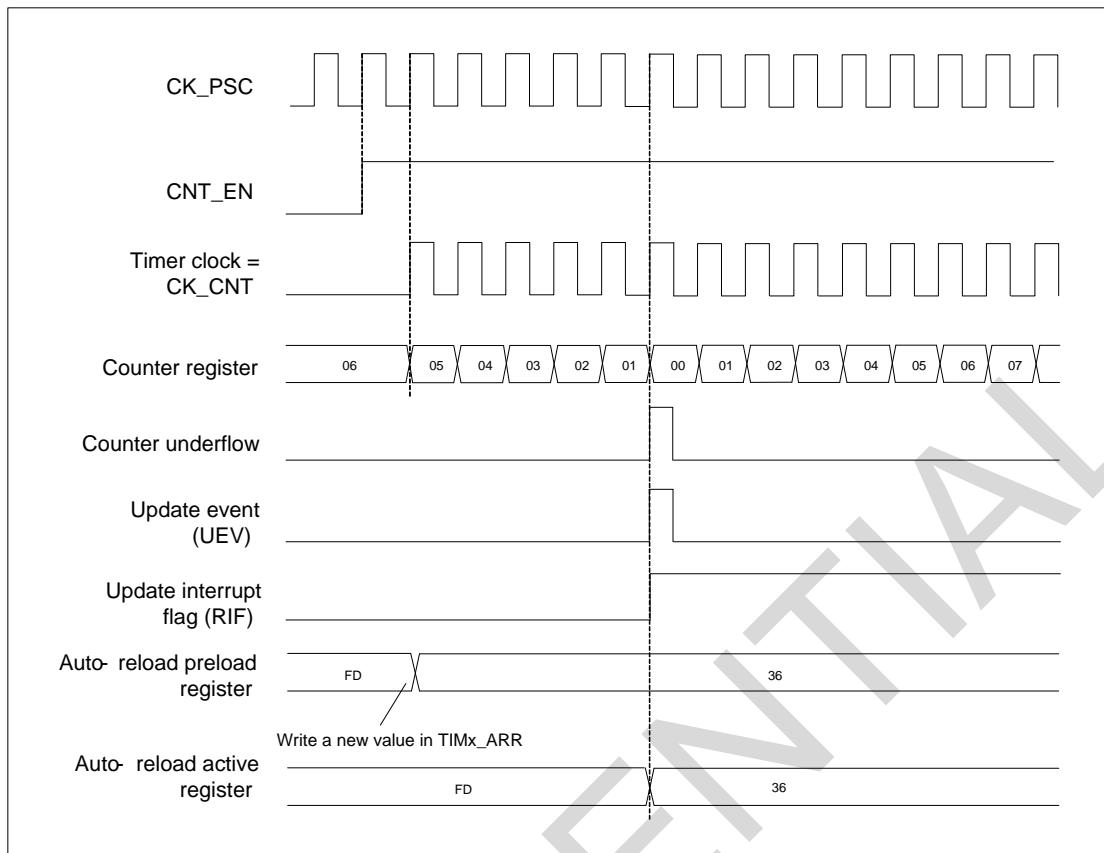


图 6-22 计数器时序图, ARPE=1 时的更新事件(计数器下溢)

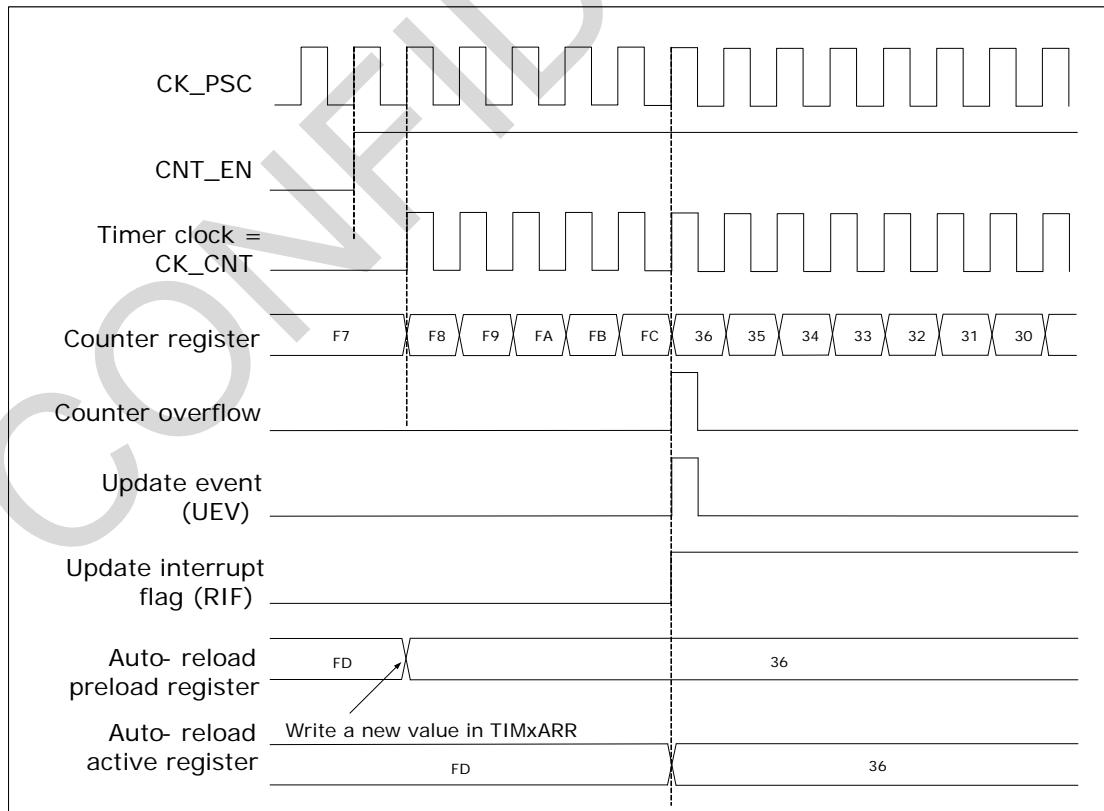


图 6-23 计数器时序图, ARPE=1 时的更新事件(计数器上溢)

6.2.2.6 重复计数器

更新事件只在重复计数器到达零时产生。这一特性可被用于产生 PWM 信号。定时器时基单元描述了如何通过计数器上溢/下溢产生更新事件 (UEV)。

每 N 次计数器上溢或下溢时，数据皆由预载寄存器传输至影子寄存器(TIMx_ARR 预载寄存器, TIMx_PSC 预分频寄存器, 比较模式下的 TIMx_CCRn 捕获/比较寄存器)。N 表示 TIMx_RCR 重复计数寄存器的值。

重复计数器会因下列任一原因递减：

- ◇ 在每次计数器向上计数上溢时
- ◇ 在每次计数器向下计数下溢时
- ◇ 在每次中央对齐模式时计数器上溢与下溢时。虽然限制了 PWM 的最大循环周期为 128，仍造成每次 PWM 周期更新两次占空比的可能。在每次 PWM 周期中央对齐模式仅刷新一次比较寄存器时，根据波形的对称性，其最大分辨率为 $2 \times T_{ck}$

重复计数器是自动加载的，重复速率为 TIMx_RCR 寄存器的值(参考图 6-24)。当更新事件由软件(设定 TIMx_EGR 寄存器 UG 位)或是由硬件通过从模式控制器产生时，无论重复计数器值以及重新加载寄存器的值为多少，都会立即发生更新事件，并且 TIMx_RCR 寄存器的值被重新载入重复计数器。

在中央对齐模式下，以 TIMx_RCR 奇数值为例，更新事件会在 TIMx_RCR 寄存器的上溢或下溢时发生，以及这取决于何时写入 TIMx_RCR 寄存器以及何时启动计数器。若 TIMx_RCR 在启动计数器前被写，UEV 发生于下溢事件。若 TIMx_RCR 在启动计数器后被写，则 UEV 发生于上溢。举例说明当 RCR=3，根据 TIMx_RCR 被写入时刻，UEV 在每四个上溢或下溢事件产生。

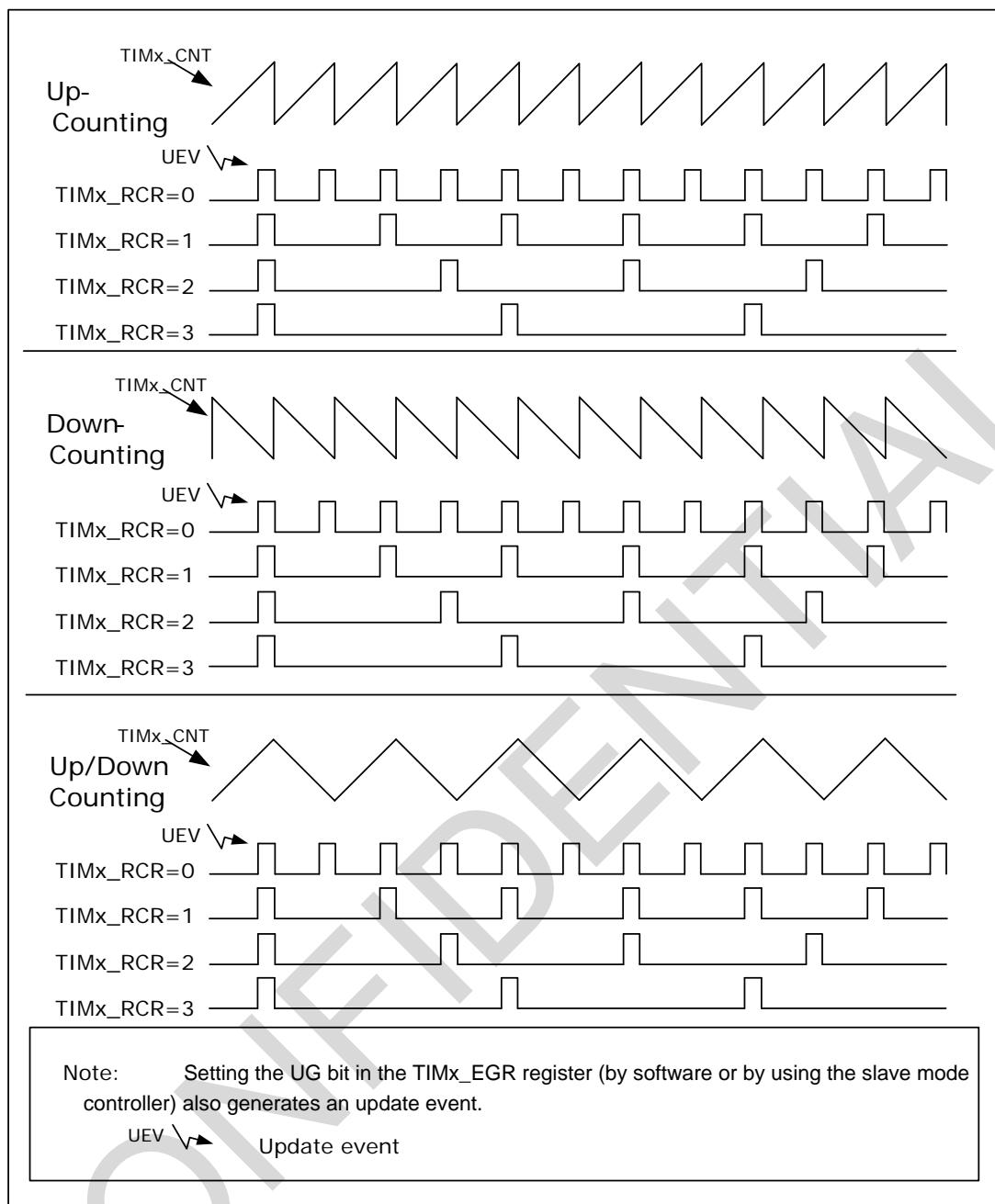


图 6-24 以模式与 TIMx_RCR 寄存器设置为依据的更新比率范例

6.2.3 时钟源

计数器时钟可由下列时钟源提供:

- ◊ 内部时钟(CK_INT)
- ◊ 外部时钟模式 1: 外部输入引脚
- ◊ 外部时钟模式 2: 外部触发输入 ETR

6.2.3.1 内部时钟源(CK_INT)

若从模式控制器被关闭(TIMx_SMCR 寄存器内, SMS=000), CEN、DIR(TIMx_CR1 寄存器)与 UG 位(TIMx_EGR 寄存器)皆为实际控制位, 且只可被软件修改(UG 位仍被自动

清除), 只要 CEN 位被写为'1', 预分频器的时钟即由内部时钟 CK_INT 提供。

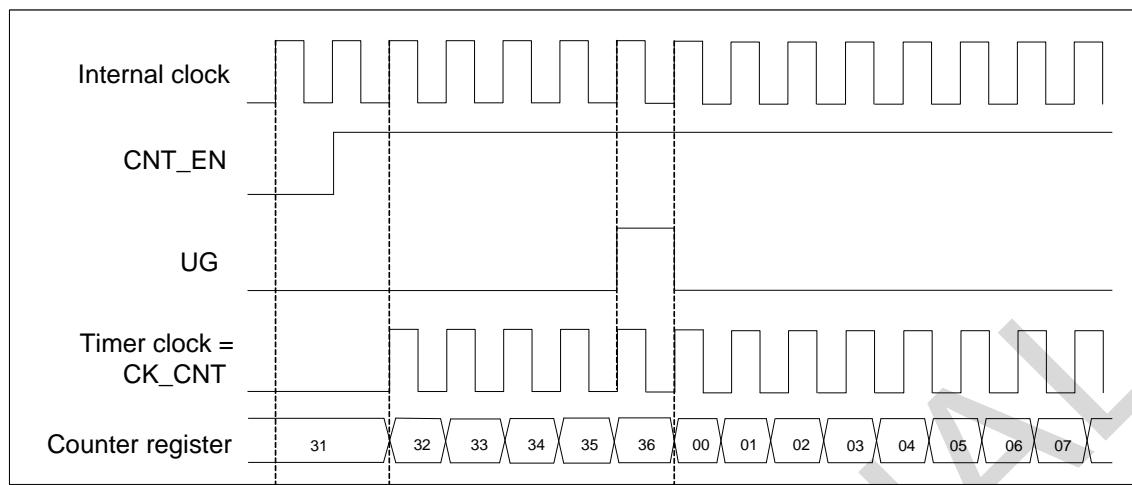


图 6-25 一般模式下的控制电路, 内部时钟分频因子为 1

6.2.3.2 外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时, 此模式会被选定。计数器在被选定的输入端的每个上升或下降沿计数。

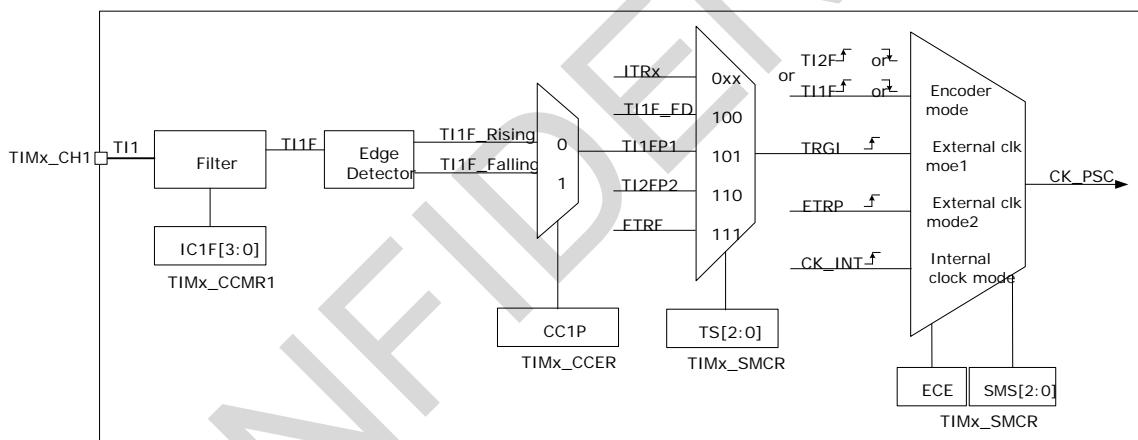


图 6-26 TI1 外部时钟连接范例

例：要配置向上计数器在 TI1 输入端的上升沿计数，如以下步骤：

- ◇ 配置 TIMx_CCMR 寄存器 CC1S=01，配置通道 1 以检测 TI1 输入的上升沿。
- ◇ 写 TIMx_CCMR 寄存器的 IC1F<3:0>位以配置输入滤波器期间(若没有滤波器需求，维持 IC1F=0000)
- ◇ 写 TIMx_CCER 寄存器的 CC1P=0，以选择上升沿极性
- ◇ 写 TIMx_SMCR 寄存器的 SMS=111，以配置定时器外部时钟模式 1
- ◇ 写 TIMx_SMCR 寄存器中的 TS=101，以选定 TI1 作为触发输入源
- ◇ 写 TIMx_CR1 寄存器的 CEN=1 以便能计数器

当上升沿发生于 TI1，计数器计数一次且 TIF 标志被设置。TI1 上升沿与实际时钟间的延时，取决于 TI1 输入的再同步电路。

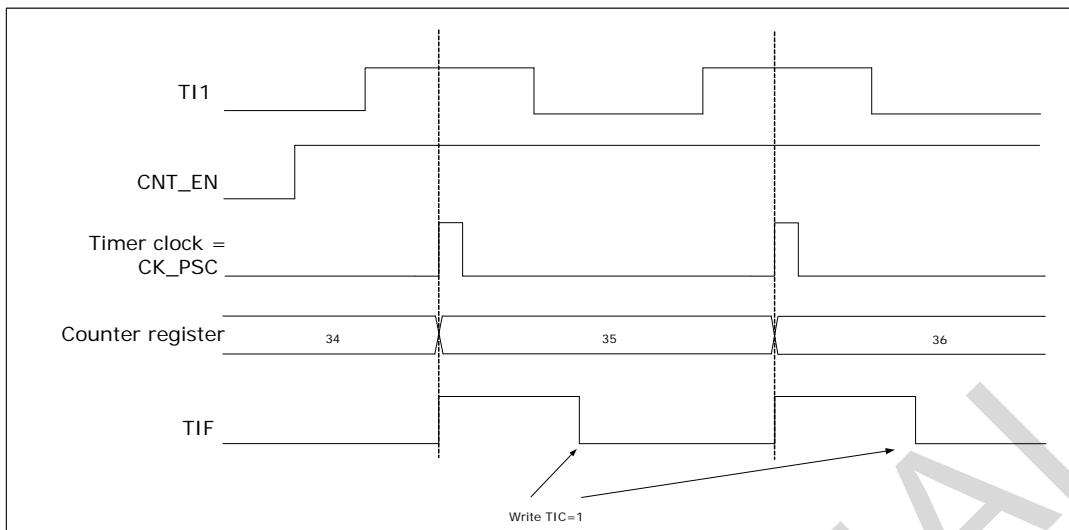


图 6-27 外部时钟模式 1 的控制电路

6.2.3.3 外部时钟源模式 2

此模式通过写 TIMx_SMCR 寄存器的 $\text{ECE}=1$ 被选定，计数器可于每个外部触发输入 ETR 上升或下降沿计数，并可搭配预分频器。

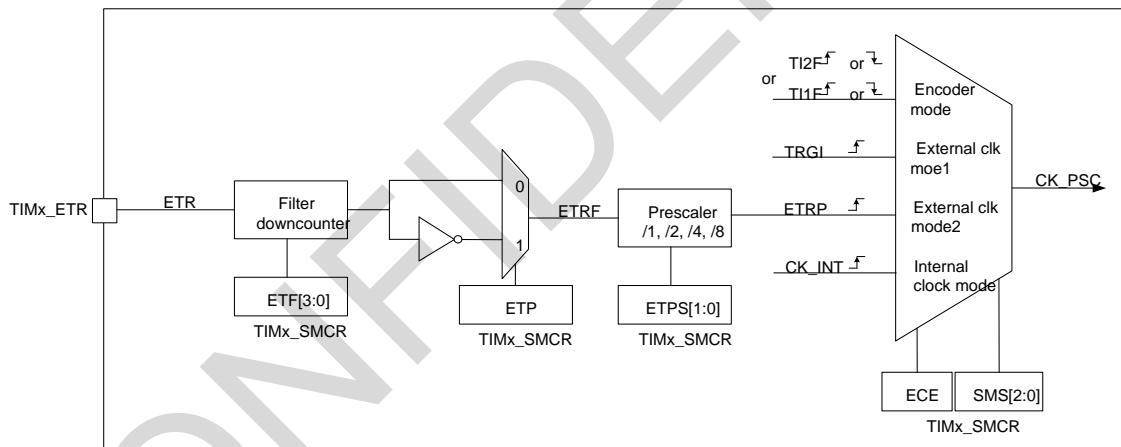


图 6-28 ETR 给予一外部触发输入区块概述

例：欲设置上述计数器于每两个 ETR 上升沿计数，则使用下列方式：

- ◇ 此例中无须使用滤波器，写 TIMx_SMCR 寄存器的 $\text{ETF}<3:0>=0000$
- ◇ 写 TIMx_SMCR 寄存器的 $\text{ETPS}<1:0>=01$ 以设置预分频器
- ◇ 写 TIMx_SMCR 寄存器的 $\text{ETP}=0$ 以选择 ETR 引脚的上升沿侦测
- ◇ 写 TIMx_SMCR 寄存器的 $\text{ECE}=1$ 以使能外部时钟模式 2
- ◇ 写 TIMx_CR1 寄存器的 $\text{CEN}=1$ 以使能计数器

计数器在每两个上升沿计数。

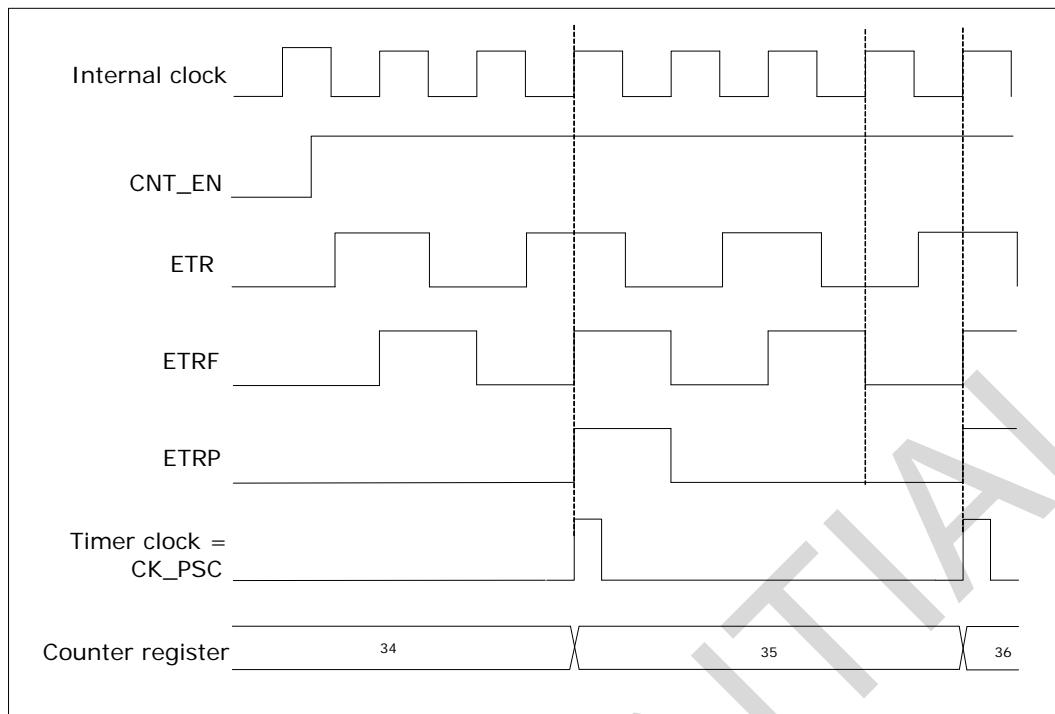


图 6-29 外部时钟模式 2 的控制电路

6.2.4 捕获/比较通道

每个捕获/比较通道，都是围绕着捕获/比较寄存器(包含影子寄存器)，包含捕获输入部分(含数字滤波器，多路复用与预分频器)以及输出阶段(含比较器与输出控制)。

下图 6-30 为单个捕获/比较通道的概述。

输入阶段范例，对应的 TIx 输入信号经过滤波器后产生一滤波后的 TIx_F 信号，接着通过一个带极性边沿侦测器产生 $TIxFPn$ 信号， $TIxFPn$ 信号可作为从模式控制器的触发输入或是作为捕获控制。该信号经过预分频后再进入捕获寄存器。

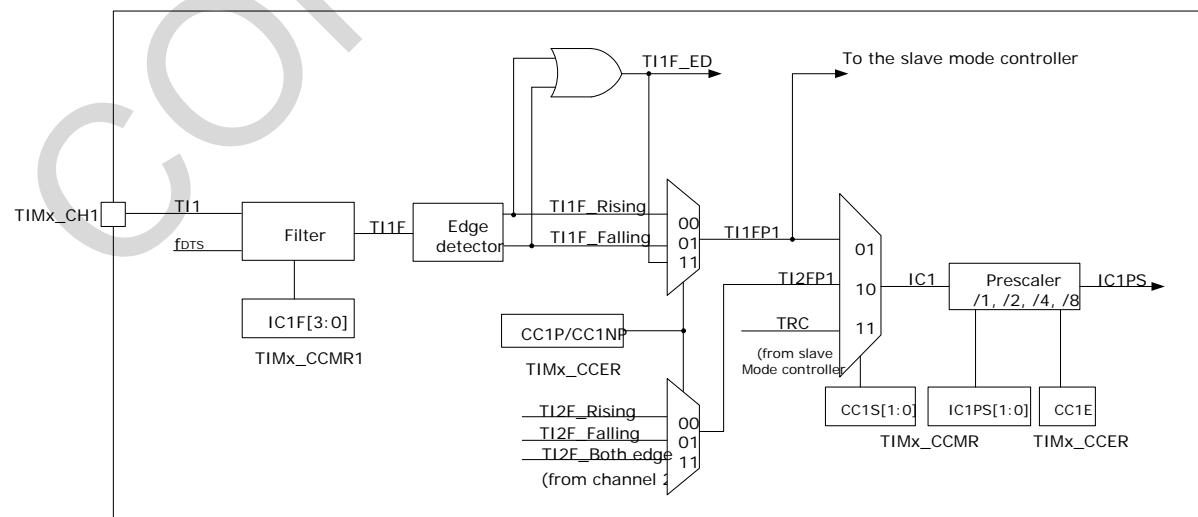


图 6-30 捕获/比较通道(例如，通道 1 输入阶段)

输出部分产生一个中间波形 OCnRef(高有效)作为基准，链结的末端决定最终输出信号的极性。

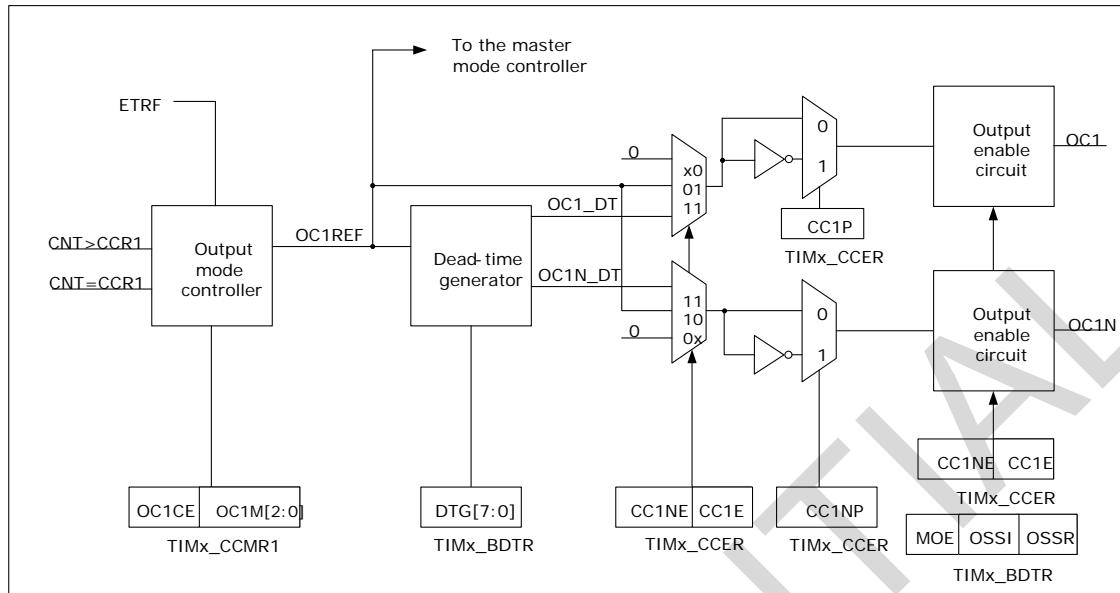


图 6-31 捕获/比较通道的输出阶段(通道 1 至 3)

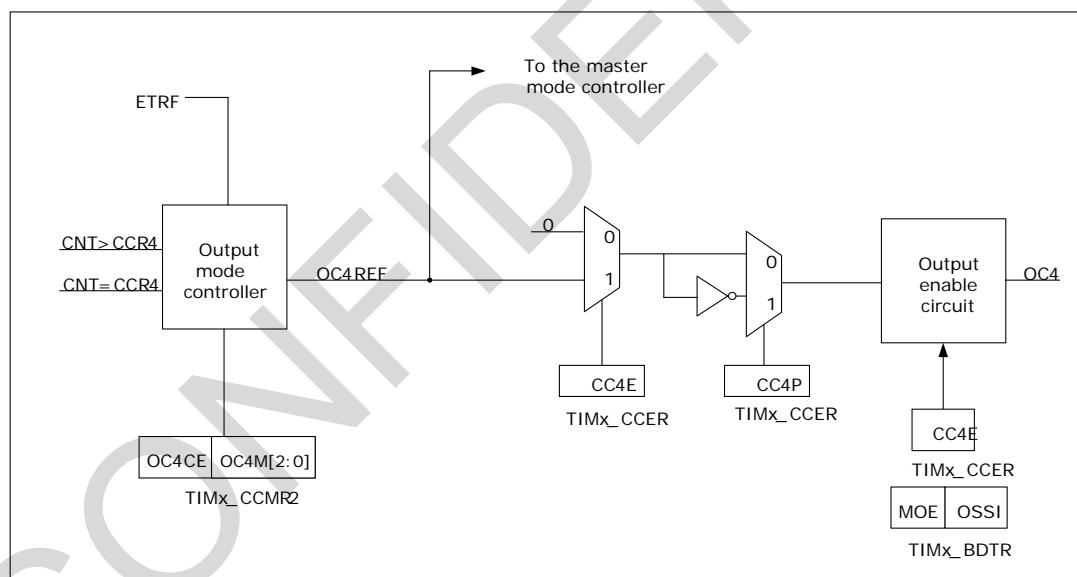


图 6-32 捕获/比较通道的输出阶段(通道 4)

捕获/比较区块是由一个预载寄存器与影子寄存器组成。读写皆可存取预载计存器：

- ◇ 在捕获模式下，捕获实际上皆于影子寄存器完成并复制到预载寄存器。
- ◇ 在比较模式下，预载寄存器内容被复制到影子寄存器与计数器做比较。

6.2.4.1 输入捕获模式

在输入捕获模式下，当检测到 ICn 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器(TIMx_CCRn)中。当发生捕获时，相应的中断标志位 CCxIF 被置一（对 TIMx_ICR 寄存器写 1 可以清除 TIMx_RIF 中断标志位），ADTIM1 总中断标志位 TIM1IF 也被置一。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

- ◇ 选择有效输入端：如果 TIMx_CCR1 必须连接到 TI1 输入，需写入 TIMx_CCMR1 寄存器中的 CC1S 为 01，此通道被配置为输入，并且 TIMx_CCR1 寄存器硬件自动变为只读。
- ◇ 参照连结至定时器的信号，编程你所需要的输入滤波器期间(当输入为 T1x 输入，则对 TIMx_CCMR1 寄存器进行编程设置。当触发时输入信号在五个内部时钟周期内并不稳定，用户需编程比五个时钟周期还长的滤波器期间。可在 8 个新的电平连续采样被侦测到(于 f_{DTS} 频率采样)时开始 TI1 传输。接着写 TIMx_CCMR1 寄存器的 IC1F 位为 0011。)
- ◇ 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P 与 CC1NP 为 0(上升沿)
- ◇ 设置输入预分频器。如果希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止使用(写 TIMx_CCMR1 寄存器的 IC1PS 位为‘00’)。
- ◇ 设定 TIMx_CCER 寄存器 CC1E 位为 1，以使能捕捉计数器至捕捉寄存器
- ◇ 可通过设定 TIMx_IER 寄存器的 CC1IE 位和 ADTIM1 模块总中断使能 TIM1IE 位为 1，以开启相关中断需求

当输入捕获中断发生时：

- ◇ 捕捉计数器的值传送到 TIMx_CCR1 寄存器
- ◇ CC1IF、TIM1IF 标志位被置一(中断状态)。当至少有 2 个连续捕获发生，而标志位 CC1IF 未被清除，捕获溢出标志位 CC1OF 也会被置 1。
- ◇ 如设置 TIMx_IER 寄存器的 CC1IE 位为 1，以及 TIM1IE 位为 1，则会产生一个中断。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：ICn 中断需求可由软件设置 TIMx_EGR 寄存器内对应 CCnG 位产生。

6.2.4.2 PWM输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- ◇ 两个 ICn 信号被映射至同一个 Tlx 输入
- ◇ 这 2 个 ICn 信号为边沿有效，但是极性相反
- ◇ 其中一个 TlxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式

例：使用以下方法可测量输入到 TI1 上的 PWM 信号的周期(TIMx_CCR1 寄存器)和占空比(TIMx_CCR2 寄存器)，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)：

- ◇ 选择 TIMx_CCR1 的有效输入：配置 TIMx_CCMR1 寄存器的 CC1S=01(选中 TI1)
- ◇ 选择 TI1FP1 的有效极性(用来捕获数据到 TIMx_CCR1 中和清除计数器)：配置 CC1P 为 0(上升沿有效)

- ◇ 选择 **TIMx_CCR2** 有效输入：配置 **TIMx_CCMR1** 寄存器的 **CC2S=10**(选中 **TI1**)
- ◇ 选择 **TI1FP2** 有效极性(用于 **TIMx_CCR2** 捕捉)：配置 **CC2P** 位为‘1’(下降沿有效)
- ◇ 选择有效的触发输入信号：配置 **TIMx_SMCR** 寄存器的 **TS** 位为 **101**(选择 **TI1FP1**)
- ◇ 设置从模式控制器于复位触发模式：配置 **TIMx_SMCR** 寄存器的 **SMS** 位为 **100**
- ◇ 使能捕获：配置 **TIMx_CCER** 寄存器中 **CC1E=1, CC2E=1**。

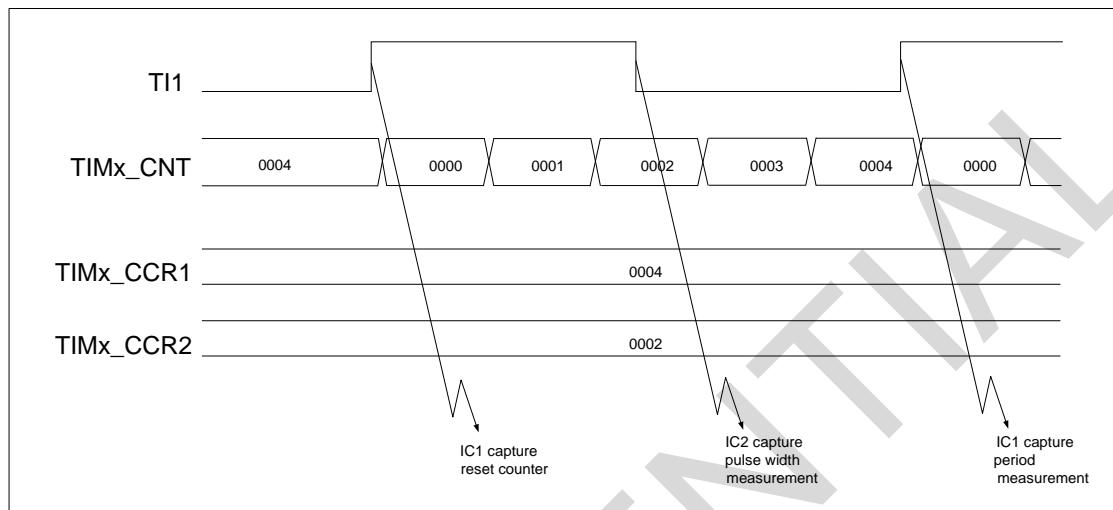


图 6-33 PWM 输入模式时序

6.2.4.3 强制输出模式

在输出模式(**TIMx_CCMRn** 寄存器中 **CCnS=00**)下，输出比较信号(**OCnREF** 和相应的 **OCn/OCnN**)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 **TIMx_CCMRn** 寄存器中相应的 **OCnM=101**，即可强置输出比较信号(**OCnREF/OCn**)为有效状态。这样 **OCnREF** 被强置为高电平(**OCnREF** 始终为高电平有效)，同时 **OCn** 得到 **CCnP** 极性相反的信号。

例如：**CCnP=0**(**OCn** 高电平有效)，则 **OCx** 被强置为高电平。

置 **TIMx_CCMRn** 寄存器中的 **OCnM=100**，可强置 **OCnREF** 信号为低。

该模式下，在 **TIMx_CCRn** 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断。这将会在下面的输出比较模式一节中介绍。

6.2.4.4 输出比较模式

此功能用于控制输出波形或是指示一段给定的时间已经到时。

当于捕捉/比较寄存器与计数器发现匹配时，输出比较功能：

- ◇ 将输出比较模式(**TIMx_CCMRn** 寄存器中的 **OCnM** 位)和输出极性(**TIMx_CCER** 寄存器中的 **CCnP** 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(**OCnM=000**)、被设置成有效电平(**OCnM=001**)、被设置成无效电平(**OCnM=010**)或进行翻转(**OCnM=011**)。

- ◇ 设置中断状态寄存器中的标志位(TIMx_RIF 寄存器中的 CCnIF 位)。
- ◇ 若设置了相应的中断使能位(TIMx_IER 寄存器中的 CCnIE 位和 ADTIM1 总中断使能位 TIM1IE)，则产生一个中断。

TIMx_CCMRn 寄存器的 OCnPE 位用于选择 TIMx_CCRn 寄存器是否需要使用预装载寄存器在输出比较模式下，更新事件 UEV 对于 OCnREF 与 OCn 输出没有效果。输出比较模式也可被用来输出单一脉冲(单脉冲模式)。

输出比较模式的配置过程：

- ◇ 选定计数器时钟(内部,外部,预分频)。
- ◇ 写入需要的数据至 TIMx_ARR 与 TIMx_CCRn 寄存器。
- ◇ 若有中断需求，则设置 TIMx_IER 的 CCnIE 位和 TIM1IE 位为 1
- ◇ 选择输出模式，例：
 - 设置 OCnM=011，当计数器 CNT 与 CCRn 匹配时触发 OCn 输出信号翻转
 - 写 OCnPE=0 禁用预载寄存器
 - 写 CCnP=0 选择高电平作为有效电平
 - 写 CCnE=1 使能输出
- ◇ 设置 TIMx_CR1 寄存器的 CEN 位以使能计数器

TIMx_CCRn 寄存器可于任何时间被软件更新以控制输出波形，条件是预载寄存器并未使能(OCxPE='0'，此外 TIMx_CCRn 影子寄存器只有在下一个更新事件 UEV 被更新)。下图提供一范例：

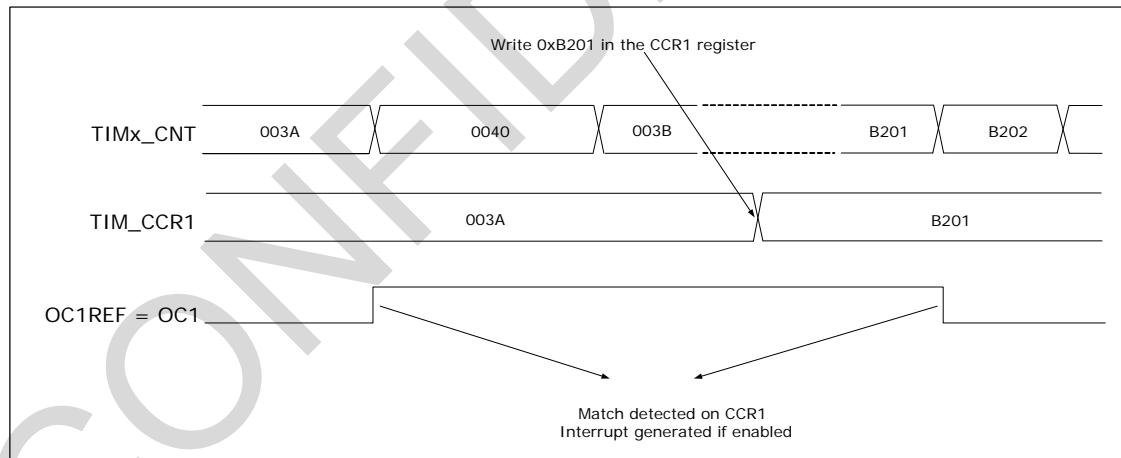


图 6-34 输出比较模式，触发 OC1

6.2.4.5 PWM模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRn 寄存器确定占空比的信号。

在 TIMx_CCMRn 寄存器中的 OCnM 位写入‘110’(PWM 模式 1)或‘111’(PWM 模式 2)，能够独立地设置每个 OCn 输出通道产生一路 PWM。必须设置 TIMx_CCMRn 寄存器 OCnPE 位以使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 **TIMx_EGR** 寄存器中的 **UG** 位以初始化所有寄存器。

OCn 的极性可以通过软件在 **TIMx_CCER** 寄存器中的 **CCnP** 位设置，它可以设置为高电平有效或低电平有效。**TIMx_CCER** 寄存器中的 **CCnE,CCnNE,MOE,OSSI** 和 **OSSR** 位联合控制 **OCn** 输出使能(**TIMx_CCER** 与 **TIMx_BDTR** 寄存器)。详见 **TIMx_CCER** 寄存器的描述。

在 **PWM** 模式(模式 1 或模式 2)下，**TIMx_CNT** 和 **TIMx_CCRn** 始终在进行比较，(依据计数器的计数方向)以确定是否符合 **TIMx_CCRn≤TIMx_CNT** 或者 **TIMx_CNT≤TIMx_CCRn**。

根据 **TIMx_CR1** 寄存器中 **CMS** 位的状态，定时器能够产生边沿对齐的 **PWM** 信号或中央对齐的 **PWM** 信号。

注：当 **TIM1_CR1H** 寄存器中的 **HTOEON** 为 1 时，在 ICD 调试模式下 **HALT** 暂停时关断 **PWM** 输出；**HTOEON** 为 0 时，在 ICD **HALT** 模式下使能 **PWM** 输出。

PWM 边沿对齐模式

◇ 向上计数配置

当 **TIMx_CR1** 寄存器中的 **DIR** 位为低的时候执行向上计数。

参考以下为一 **PWM** 模式 1 的例子。当 **TIMx_CNT<TIMx_CCRn** 时，**PWM** 参考信号 **OCnREF** 为高，否则为低；当 **TIMx_CCRn** 中的比较值大于自动重装载值(**TIMx_ARR**)，则 **OCnREF** 一直保持为‘1’；当 **TIMx_CCRn** 中的比较值为 0，则 **OCnREF** 一直保持为‘0’。下图为 **TIMx_ARR=8** 时边沿对齐的 **PWM** 波形实例。

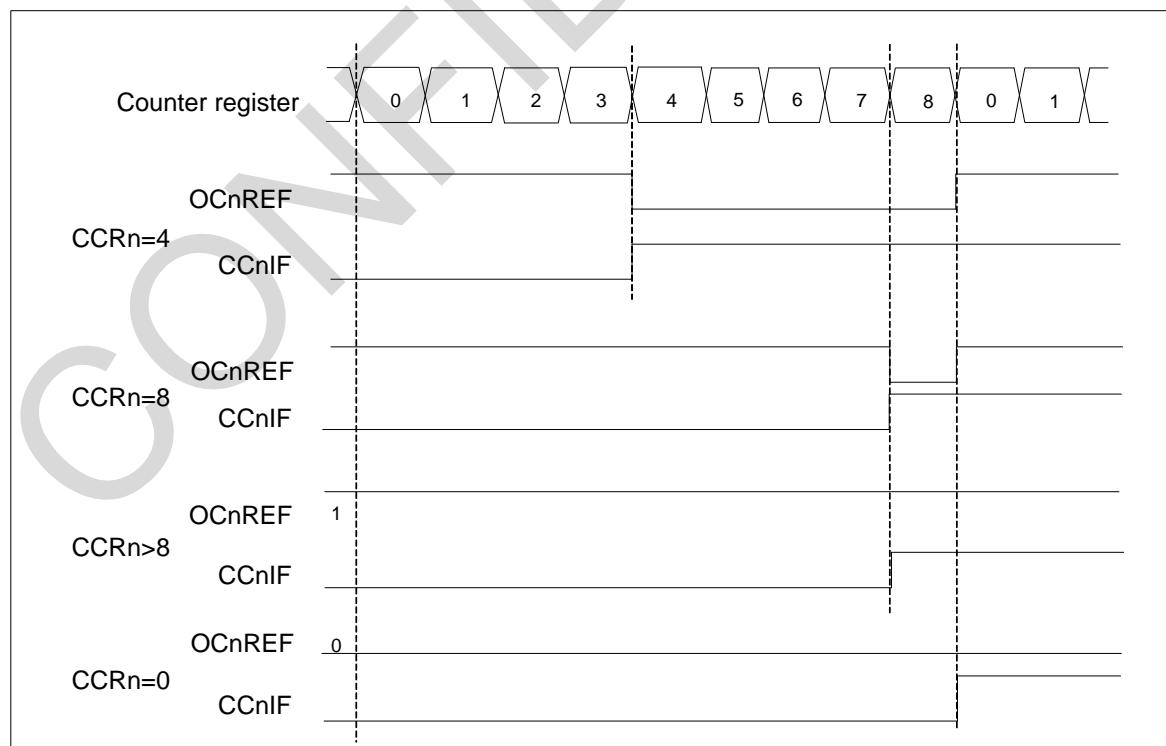


图 6-35 边沿对齐 **PWM** 波形，向上计数(**ARR=8**)

◇ 向下计数配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1，当 $\text{TIMx_CNT} > \text{TIMx_CCRn}$ 时参考信号 OCnREF 为低，否则为高。当 TIMx_CCRn 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCnREF 一直保持为'1'。该模式下不能产生 0% 的 PWM 波形。

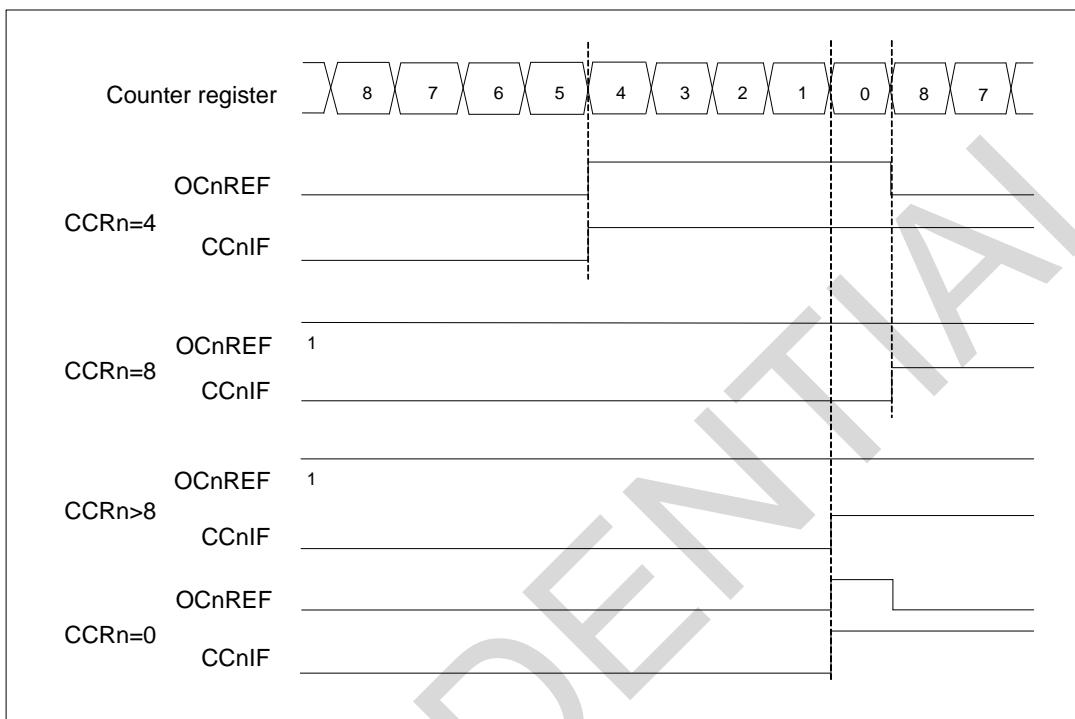


图 6-36 边沿对齐 PWM 波形，向下计数($\text{ARR}=8$)

PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为'00'时，为中央对齐模式(所有其他的配置对 OCnREF/OCn 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。 TIMx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不需要用软件修改。

下图给出了一些中央对齐的 PWM 波形的例子：

◇ $\text{TIMx_ARR}=8$

◇ PWM 模式 1

TIMx_CR1 寄存器的 CMS=01/10/11，在中央对齐模式 1 下，标志位产生的位置，下图中已经用箭头标出。

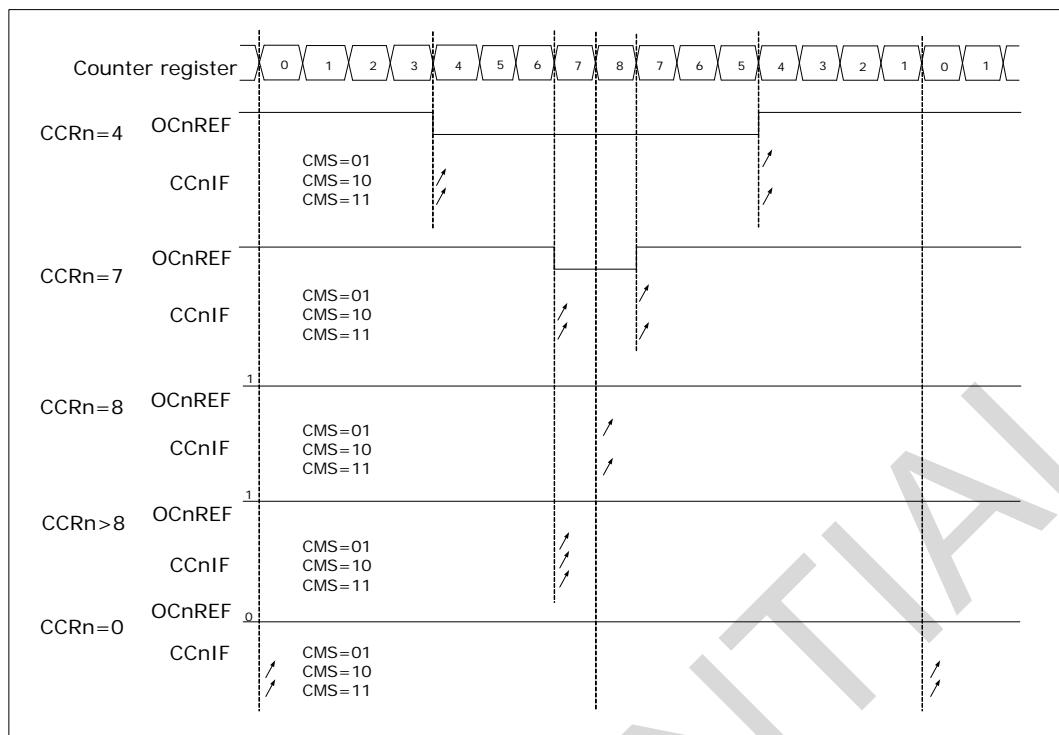


图 6-37 中央对齐 PWM 波形(ARR=8)

使用中央对齐模式的提示:

- ◇ 进入中央对齐模式时, 使用当前的向上/向下计数配置; 这就意味着计数器向上还是向下计数取决于 **TIMx_CR1** 寄存器中 **DIR** 位的当前值。此外, 软件不能同时修改 **DIR** 和 **CMS** 位。
- ◇ 不推荐当运行在中央对齐模式时改写计数器, 因为这会产生不可预知的结果。特别地:
 - 如果写入计数器的值大于自动重加载的值(**TIMx_CNT**>**TIMx_ARR**), 则方向不会被更新。例如, 如果计数器正在向上计数, 它就会继续向上计数。
 - 如果将 0 或者 **TIMx_ARR** 的值写入计数器, 方向被更新, 但不产生更新事件 **UEV**。
- ◇ 使用中央对齐模式最保险的方法, 就是在启动计数器之前产生一个软件更新(设置 **TIMx_EGR** 位中的 **UG** 位), 并且不要在计数进行过程中修改计数器的值。

6.2.4.6 单脉冲模式

单脉冲模式为前一模式的特例。其允许计数器响应一个激励, 并在可编程延迟后产生一个可编程长度的脉冲。

计数器可由从模式控制器控制启动。由输出比较模式或 **PWM** 模式产生波形。可通过设置 **TIMx_CR1** 寄存器的 **OPM** 位以选择单脉冲模式。如此计数器会在下一个更新事件 **UEV** 自动停止。

仅当比较值不同于计数器的初始值时, 单脉冲才被正确产生。在开始之前(当定时器正在等待触发器时), 必须依照下列方式配置:

- ◇ 向上计数: **CNT**<**CCRn** \leq **ARR**(特例, **0**<**CCRn**)
- ◇ 向下计数: **CNT**>**CCRn**

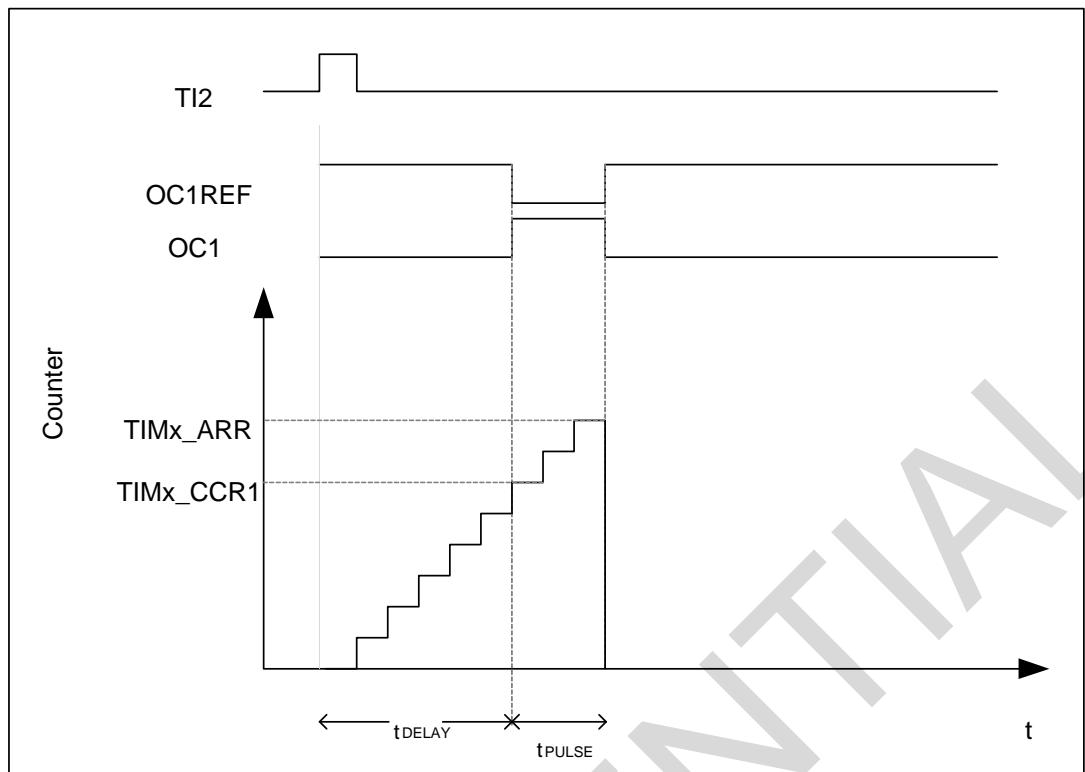


图 6-38 单脉冲模式范例

例如，你需要在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲，从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后。

假定 TI2FP2 作为触发 1：

- ◇ 置 TIMx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- ◇ 置 TIMx_CCER 寄存器中的 CC2P=0 和 CC2NP=0，使 TI2FP2 能够检测上升沿。
- ◇ 置 TIMx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发(TRGI)。
- ◇ 置 TIMx_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。

单脉冲模式的波形由写入比较寄存器决定(要考虑时钟频率和计数器预分频器)

- ◇ t_{DELAY} 由 TIMx_CCR1 寄存器中的值定义。
- ◇ t_{PULSE} 由自动装载值和比较值之间的差值定义($TIMx_ARR-TIMx_CCR1$)。
- ◇ 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的变换；首先要置 TIMx_CCMR1 寄存器的 OC1M=111，选择 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE=1；此时必须向 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置 0。

因为只需要一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=1，在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。当 TIMx_CR1 寄存器中的 OPM=0 时，进入重复模式。

特殊情况：OCx 快速使能

在单脉冲模式下，在 Tlx 输入脚的边沿检测逻辑会设置 CEN 位为 1 来启动计数器。然后计数器和比较值间的比较结果产生了单脉冲的输出。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。

如果需要以最小延时输出波形，可以设置 $TIMx_CCMRn$ 寄存器中的 OCnFE 位；此时 OCnREF(和 OCn)直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCnFE 只在通道配置为 PWM 模式 1 和 PWM 模式 2 时起作用。

6.2.4.7 带死区互补输出

高级控制定时器可输出两个互补信号并管理开关切换

该时间通常被称作死区，用户可根据连接的装置输出与其特性来调整死区时间。

通过写 $TIMx_CCER$ 寄存器的 CCnP 与 CCnNP 位，可单独选择输出极性(主要输出 OCn 或互补性 OCnN)。

互补性信号 OCn 与 OCnN 可由以下几个控制位的结合进行控制： $TIMx_CCER$ 寄存器的 CCnE 与 CCnNE 位， $TIMx_BDTR$ 与 $TIMx_CR2$ 寄存器的 MOE, OISn, OISnN, OSSI 与 OSSR 位。特别是，当切换至 IDLE(MOE 下降至 0)状态时，死区控制会被激活。

死区插入可同时设定 CCnE 与 CCxNE 位使能，如果有刹车电路，则还需设置 MOE 位。每个通道都有一个 8 位死区发生器，参考波形 OCnREF，产生 2 个输出 OCn 及 OCnN，若 OCn 与 OCnN 高有效：

◇ OCn 输出信号与参考信号相同，只是 OCn 输出的上升沿和参考信号的上升沿有一个延时。

OCnN 输出信号与参考信号相反，只是 OCnN 输出的上升沿和参考信号的下降沿有一个延时

若延迟大于有效输出带宽(OCn 或 OCnN)，那么对应脉冲不会产生。

下图显示死区发生器输出信号与参考信号 OCnREF 间的关系(假设 CCnP=1,CCnNP=0,MOE=1,CCnE=1 且于这些范例中 CCnNE=1)

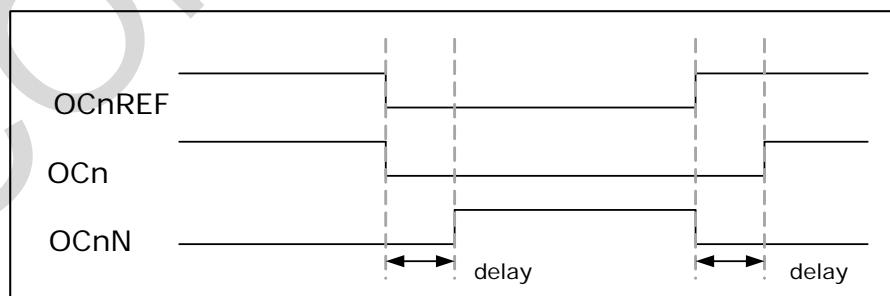


图 6-39 互补输出含死区插入

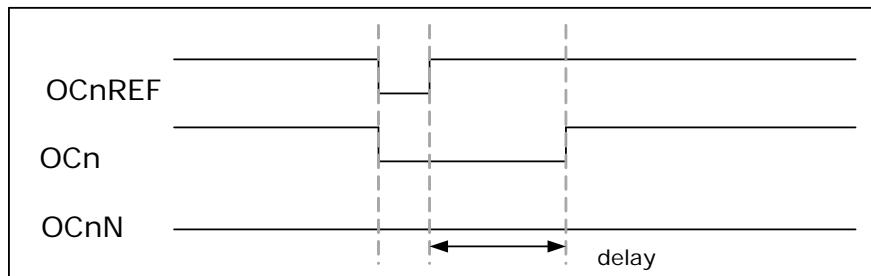


图 6-40 死区波形含大于负脉冲的延迟

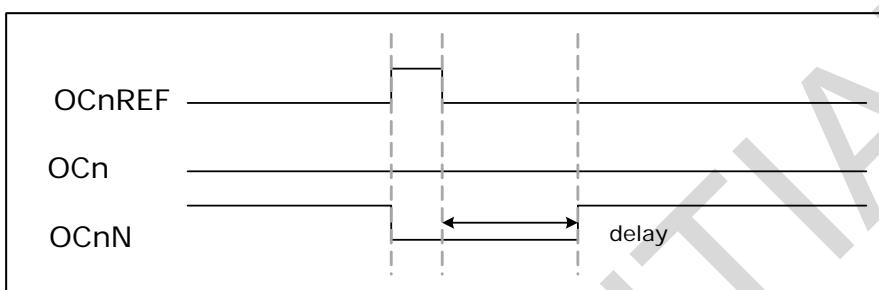


图 6-41 死区波形大于正脉冲的延迟

重新导向 OCnREF 至 OCn 或 OCnN

在输出模式下(强制, 输出比较或 PWM), OCnRF 可通过设置 TIMx_CCER 寄存器的 CCnE 与 CCnNE 位, OCnREF 可被重新导向至 OCn 输出或至 OCnN 输出。

这个功能可以在互补输出处于无效电平时, 在某个输出上可发送特定波形(如 PWM 或静止作用电平)。此外, 可让两个输出同时处于无效电平, 或者处于有效电平。

注: 当只有 OCnN 使能时(CCnE=0,CCnNE=1), 其输出不互补, 只有当 OCnREF 拉高时才会效。举例说明, 若 CCnNP=1 则 OCnN=OCnRef。另一方面, 当 OCn 与 OCnN 都被使能(CCnE=CCnNE=1), 当 OCnREF 为高时 OCn 开始有效; 然而 OCnN 相反, 当 OCnREF 为低时 OCnN 开始有效。

6.2.4.8 使用刹车功能

当使用刹车功能时, 依据相应的控制位(TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位, TIMx_CR2 寄存器中的 OISn 和 OISnN 位), 输出使能信号和无效电平都会被修改。但无论何时, OCn 和 OCnN 输出不能在同一时间同时处于有效电平上。详见带刹车功能的互补输出通道 OCn 和 OCnN 的控制位。

系统复位后, 刹车电路被禁止, MOE 位为低。设置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车功能, 刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时, 在真正写入之前会有 1 个时钟周期的延迟, 因此需要等待一个时钟周期之后, 才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的, 在实际信号(作用在输出端)和同步控制位(在 TIMx_BDTR 寄存器中)之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的, 如果当它为低时写 MOE=1, 则读出它之前必须先插入一个延时(空指令)才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时(在刹车输入端出现选定的电平), 有下述动作:

- ◇ MOE 位被异步地清除, 将输出置于无效状态、空闲状态或者复位状态(由 OSS1 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- ◇ 一旦 MOE=0, 每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSS1=0, 则定时器释放使能输出, 否则使能输出始终为高。
- ◇ 当使用互补输出时:
 - 输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作, 即使定时器没有时钟时, 此功能也有效。
 - 如果定时器的时钟依然存在, 死区生成器将会重新生效, 在死区之后根据 OISn 和 OISnN 位指示的电平驱动输出端口。即使在这种情况下, OCn 和 OCnN 也不能被同时驱动到有效的电平。注, 因为重新同步 MOE, 死区时间比通常情况下长一些(大约 2 个时钟周期)。
 - 如果 OSS1=0, 定时器释放使能输出, 否则保持使能输出; 或一旦 CCnE 与 CCnNE 的一变高时, 使能输出变为高。
- ◇ 如果设置了 TIMx_IER 寄存器中的 BIE 位, 当刹车状态标志(TIMx_RIF 寄存器中的 BIF 位)为'1'时, 则产生一个中断。
- ◇ 如果设置了 TIMx_BDTR 寄存器中的 AOE 位, 在下一个更新事件 UEV 时 MOE 位被自动置位; 例如, 这可以用来进行整形。否则, MOE 始终保持低直到被再次置'1'; 此时, 这个特性可以被用在安全方面, 你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

注: 刹车输入为电平有效。所以, 当刹车输入有效时, 不能同时(自动地或者通过软件)设置 MOE。同时, 状态标志 BIF 不能被清除。

刹车由 BRK 输入产生, 它的有效极性是可编程的, 且由 TIMx_BDTR 寄存器中的 BKE 位开启。

除了刹车输入和输出管理, 刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度, OCn/OCnN 极性和被禁止的状态, OCnM 配置, 刹车使能和极性)。用户可以通过 TIMx_BDTR 寄存器中的 LOCK 位, 从三级保护中选择一种, 参照刹车和死区寄存器(TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

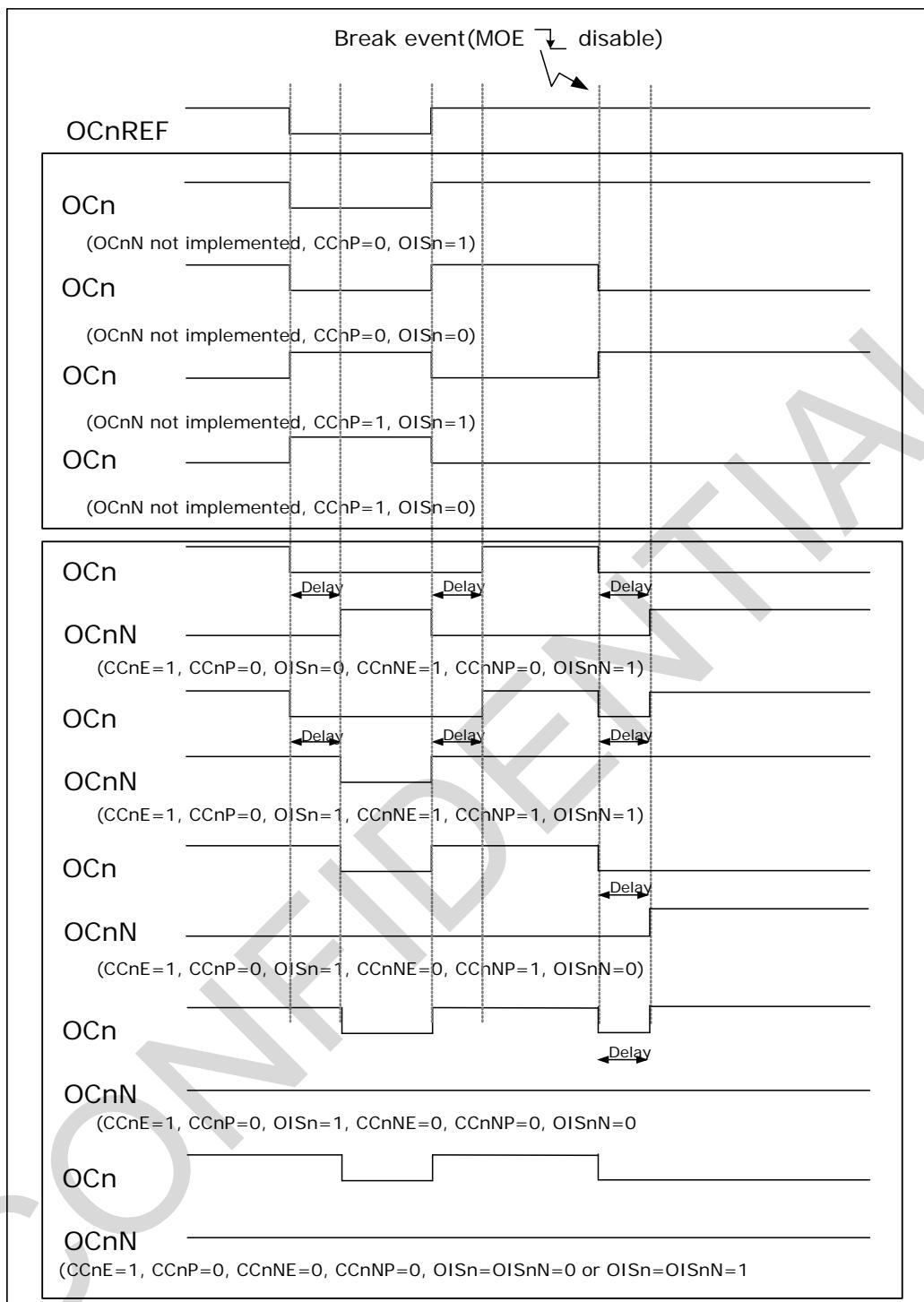


图 6-42 响应刹车的输出行为

6.2.4.9 外部事件清除OCnREF信号

当 TIMx_CCMR 寄存器的 OCnCE 使能位设置为 1 和 TIMx_SMCR 寄存器的 OCCS 位设置为 1 时，外部 ETRF 输入端的高电平能够拉低 OCnREF 信号。OCnREF 保持为低，直到下一更新事件(UEV)发生。此功能仅可被用于 PWM 模式。不能用于强制模式下。

举例说明，OCnREF 信号可被连接到比较器输出，用于电流控制。ETR 必须设置如下：

- ◇ 必须禁用外部触发预分频器：TIMx_SMC 寄存器的 ETPS<1:0>位设置为‘00’。
- ◇ 必须禁用外部时钟模式 2：TIMx_SMCR 寄存器的 ECE 位位设置为‘0’。
- ◇ 外部触发极性(ETP)与外部触发滤波器(ETF)可根据用户需求设置。

下图显示，当 ETRF 输入变成高电平时，对应不同的 OCnCE 使能位的值，OCnREF 位的信号变化。此例中定时器 TIMx 被设置为 PWM 模式。

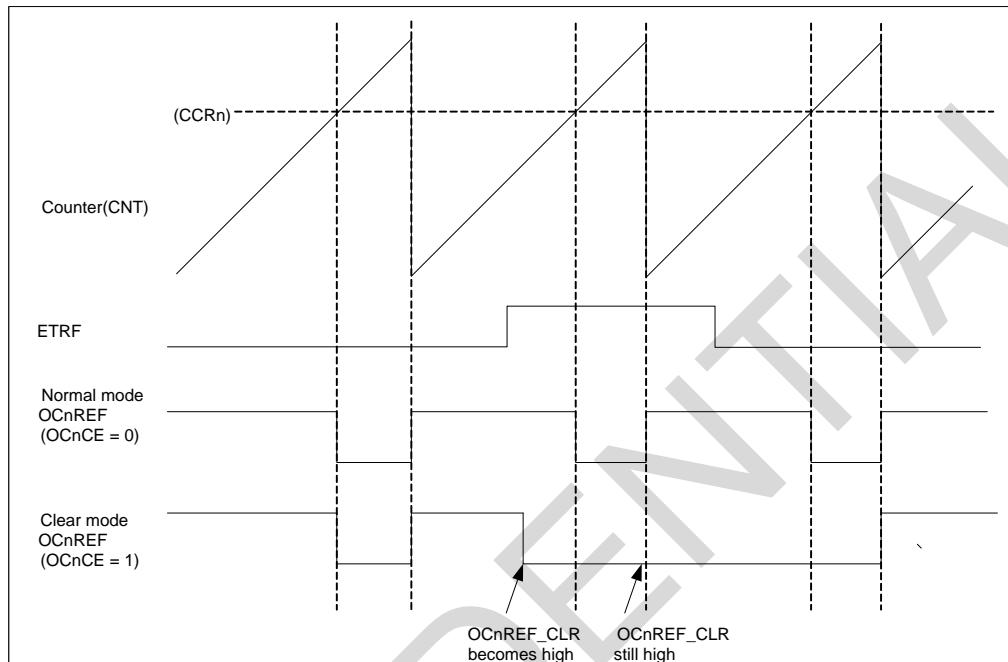


图 6-43 清除 TIMx 的 OCnREF

注：若为占空比 100% 的 PWM(若 CCRn>ARR),OCnREF 会于下一个计数器上溢再度使能。

6.2.4.10 产生六步PWM输出

当在一个通道上需要互补输出时，预装载位有 OCnM、CCnE 和 CCnNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样你就可以预先设置好下一步骤配置，并在同一个时刻同时修改所有通道的配置。COM 可以通过设置 TIMx_EGR 寄存器的 COMG 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位(TIMx_RIF 寄存器中的 COMIF 位)，这时如果已设置了 TIMx_IER 寄存器的 COMIE 位和 ADTIM1 总中断使能位 TIM1IE，则产生一个中断。

下图显示当发生 COM 事件时，三种不同配置下 OCn 和 OCnN 输出。

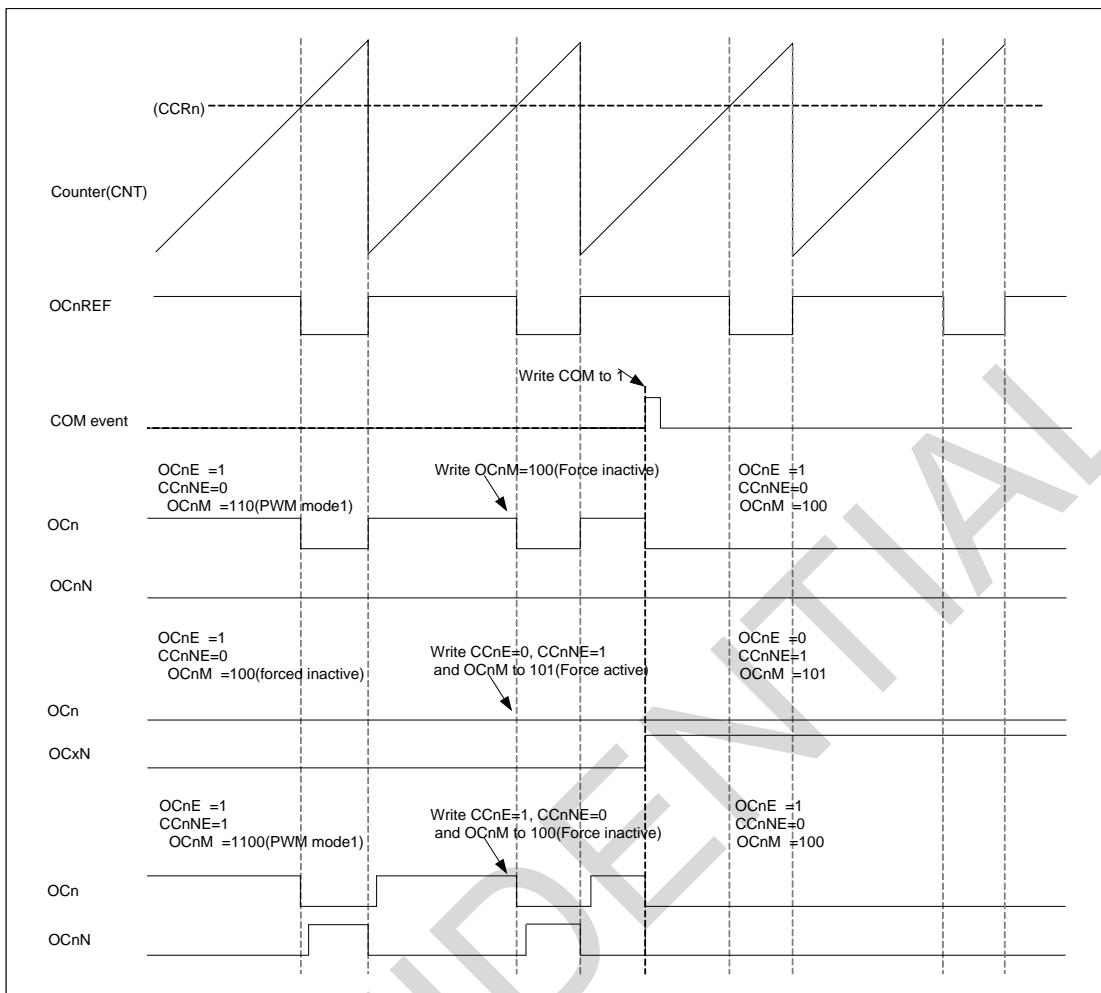


图 6-44 六步 PWM 输出产生，使用 COM 范例 (OSSR=1)

6.2.4.11 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则设置 TIMx_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则设置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则设置 SMS=011。

通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。假定计数器已经启动(TIMx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上产生的有效跳变时计数。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和极性变化，则 TI1FP1=TI1, TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行了相应的设置。不管计数器是依靠 TI1 还是 TI2 或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数(根据方向，或是 0 到 ARR 计数，

或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIMx_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

有效边沿	有效边沿相对 信号的电平 (TI1FP1 对应 TI2, TI2FP2 对 应 TI1)	TI1FP1 信号		TI2FP2 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 TI1 计数	高	向下	向上	不计数	不计数
	低	向上	向下	不计数	不计数
仅在 TI2 计数	高	不计数	不计数	向上	向下
	低	不计数	不计数	向下	向上
在 TI1 和 TI2 上计数	高	向下	向上	向上	向下
	低	向上	向下	向下	向上

表 6-3 计数方向与编码器信号的关系

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图提供一计数器运作范例，显示计数信号产生与方向控制。亦可显示如何补偿输入抖动当上下沿皆被选定时。该状况可能发生于若传感器被设置于接近其中一切换点。我们假设此例设置如下：

- ◆ CC1S=01 (TIMx_CCMR1 寄存器, IC1FP1 映射到 TI1)
- ◆ CC2S=01 (TIMx_CCMR2 寄存器, IC2FP2 映射到 TI2)
- ◆ CC1P=0 (TIMx_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- ◆ CC2P=0 (TIMx_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)
- ◆ SMS=011 (TIMx_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效)。
- ◆ CEN=1 (TIMx_CR1 寄存器, 计数器使能)

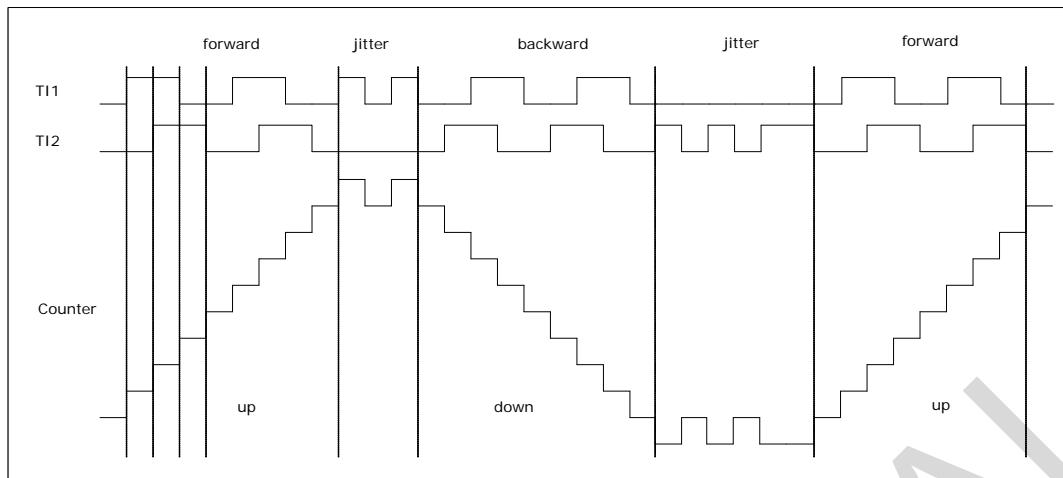


图 6-45 编码器模式下的计数器操作实例

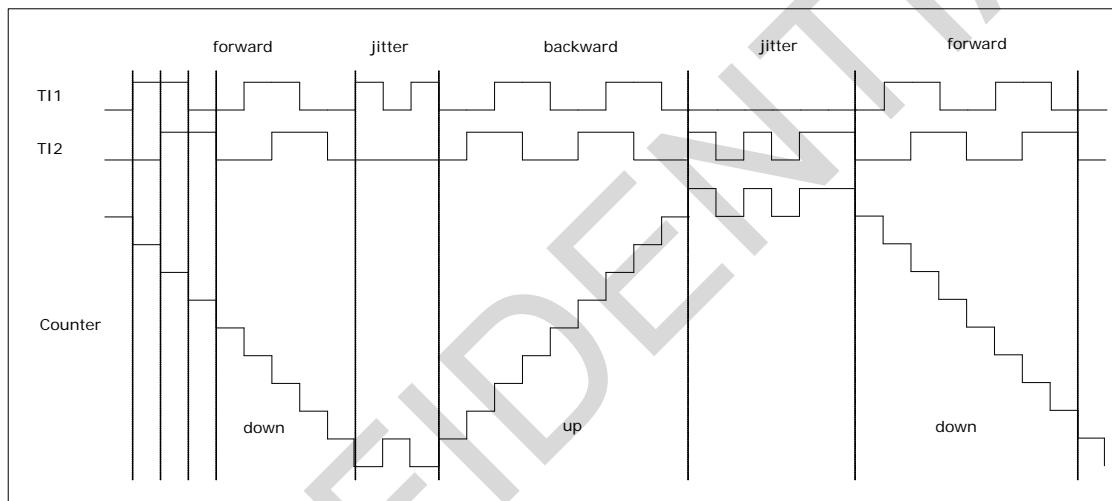


图 6-46 IC1FP1 反相的编码器接口模式实例(CC1P='1'，其他配置与上例相同)

当设置编码器接口模式时，定时器提供传感器当前位置信息。可通过量测两个第二定时器设定为捕捉模式的编码器事件周期得到动态信息(速度，加速度与减速度)。编码器输出标示机械零位可做为此功能使用。根据两个事件间的时间可知，计数器亦能于通常时间被读取。若可行亦可通过锁存计数器值转换为第三输入捕捉寄存器(捕捉信号必须具周期性且可被另一定时器产生)。

6.2.5 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部触发同步：复位模式、门控模式和触发模式。

6.2.5.1 从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIMx_ARR, TIMx_CCRn)都被更新。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

◇ 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤

波器，因此保持 $IC1F=0000$)。触发操作中不使用捕获预分频器，所以不需要配置。 $CC1S$ 位只用于选择输入捕获源，即 $TIMx_CCMR1$ 寄存器中 $CC1S=01$ 。配置 $TIMx_CCER$ 寄存器中 $CC1P=0$ 和 $CC1NP=0$ 以确定极性(只检测上升沿)。

- ◇ 配置 $TIMx_SMCR$ 寄存器中 $SMS=100$ ，配置定时器为复位模式；配置 $TIMx_SMCR$ 寄存器中 $TS=101$ ，选择 $TI1$ 作为输入源。
- ◇ 配置 $TIMx_CR1$ 寄存器中 $CEN=1$ ，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 $TI1$ 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志($TIMx_RIF$ 寄存器中的 TIF 位)被设置，并根据 $TIMx_IER$ 寄存器中 TIE (中断使能)位的设置，产生一个中断请求。

下图显示当自动重装载寄存器 $TIMx_ARR=0x36$ 时的动作。在 $TI1$ 上升沿和计数器的实际复位之间的延时取决于 $TI1$ 输入端的重同步电路。

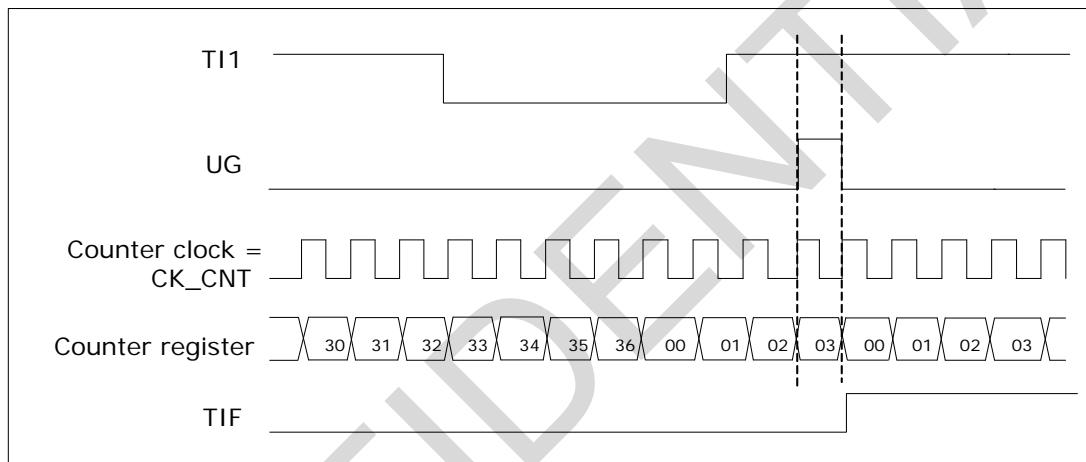


图 6-47 复位模式下的控制电路

6.2.5.2 从模式：门控模式

计数器可由选中的输入信号的电平来使能。

在如下的例子中，计数器只在 $TI1$ 为低时向上计数：

- ◇ 配置通道 1 用于检测 $TI1$ 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 $IC1F=0000$)。触发操作中不使用捕获预分频器，所以不需要配置。 $CC1S$ 位用于选择输入捕获源，置 $TIMx_CCMR1$ 寄存器中 $CC1S=01$ 。配置 $TIMx_CCER$ 寄存器中 $CC1P=1$ 和 $CC1NP=0$ 以确定极性(只检测低电平)。
- ◇ 配置 $TIMx_SMCR$ 寄存器中 $SMS=101$ ，配置定时器为门控模式；配置 $TIMx_SMCR$ 寄存器中 $TS=101$ ，选择 $TI1$ 作为输入源。
- ◇ 配置 $TIMx_CR1$ 寄存器中 $CEN=1$ ，启动计数器。(在门控模式下，如果 $CEN=0$ ，则计数器不能启动，不论触发输入电平如何)

只要 $TI1$ 为低，计数器开始依据内部时钟计数，一旦 $TI1$ 变高则停止计数。当计数器开始或停止时， $TIMx_RIF$ 中的 TIF 标志位都会被置一。

$TI1$ 上升沿和计数器实际停止之间的延时取决于 $TI1$ 输入端的重同步电路。

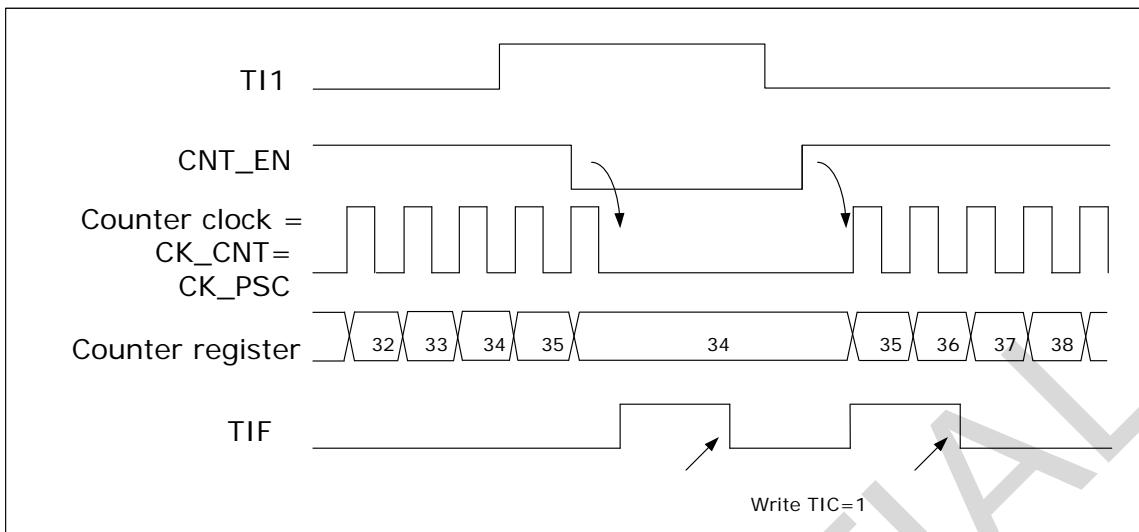


图 6-48 门控模式下的控制电路

6.2.5.3 从模式：触发模式

计数器的使能依赖于选中的输入端的事件。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- ◇ 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。配置 TIMx_CCER 寄存器中 CC2P=1 和 CC1NP=0 以确定极性(只检测低电平)。
- ◇ 配置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；配置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下开始计数，同时设置 TIF 标志为 1。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

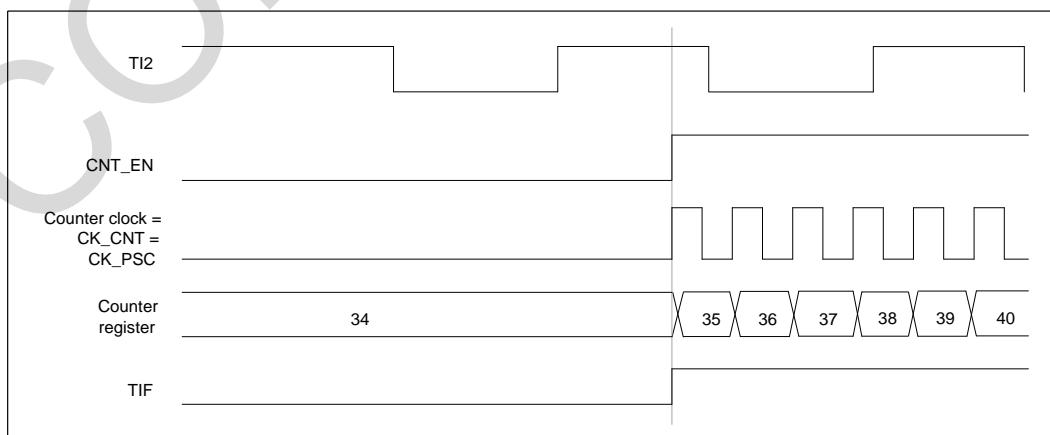


图 6-49 触发模式下的控制电路

6.2.5.4 从模式：外部时钟模式 2+触发模式

外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时, ETR 信号被用作外部时钟的输入, 可以选择另一个输入作为触发输入(在复位模式、门控模式或触发模式)。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中, 在 TI1 上出现一个上升沿后, 向上计数器在 ETR 的每一个上升沿加一:

- ◇ 通过 TIMx_SMCR 寄存器配置外部触发输入电路: ETF=0000: 没有滤波
 - ETPS=00: 不用预分频器
 - ETP=0: 检测 ETR 的上升沿, 置 ECE=1 使能外部时钟模式 2。
- ◇ 按如下配置通道 1, 检测 TI 的上升沿:
 - IC1F=0000: 没有滤波
 - 触发操作中不使用捕获预分频器, 不需要配置
 - 配置 TIMx_CCMR1 寄存器中 CC1S=01, 选择输入捕获源
 - 配置 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0 以确定极性(只检测上升沿)
- ◇ 配置 TIMx_SMCR 寄存器中 SMS=110, 配置定时器为触发模式。配置 TIMx_SMCR 寄存器中 TS=101, 选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时, TIF 标志位被置一, 计数器开始在 ETR 的上升沿计数。

ETR 信号的上升沿和计数器实际复位间的延时, 取决于 ETRP 输入端的重同步电路。

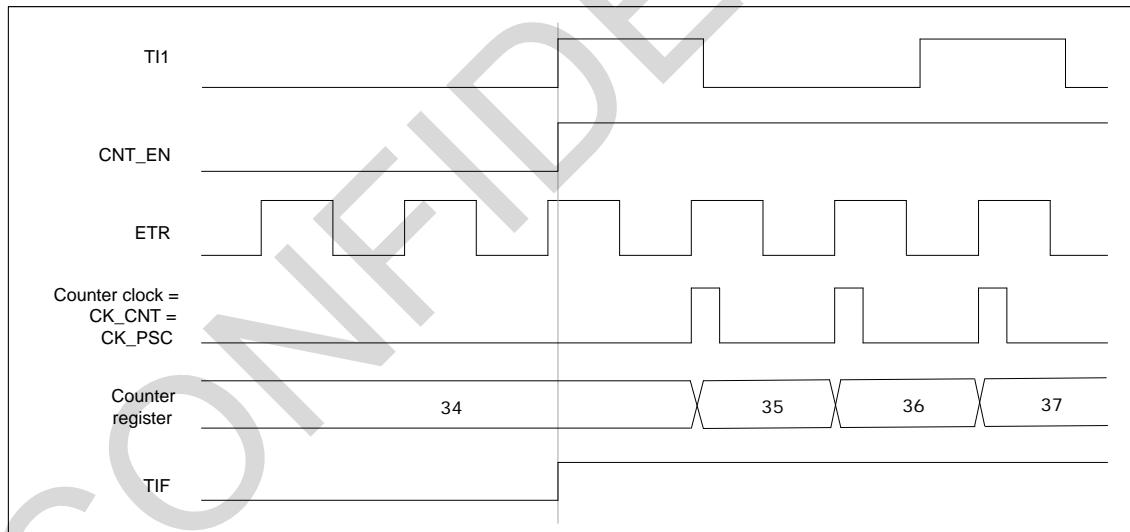


图 6-50 外部时钟模式 2+触发模式下的控制电路

6.2.6 特殊功能寄存器

6.2.6.1 控制寄存器 1 低 8 位 (TIM1_CR1L)

TIMx_CR1L: 控制寄存器 1 (TIM1_CR1L)								
Bit	7	6	5	4	3	2	1	0
Name	ARPE	CMS<1:0>		DIR	OPM	URS	UDIS	CEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	ARPE: 自动重载预载使能位 0: TIMx_ARR 寄存器没有缓冲，可被直接写入 1: TIMx_ARR 寄存器由预载缓冲器缓冲
Bit 6~5	CMS<1:0>: 中央对齐模式选择位 00: 边沿对齐模式，计数器根据方向位（DIR）向上或向下计数。 01: 中央对齐模式 1，计数器以交替方式向上或向下计数。仅当计数器向下计数时，配置为输出的通道（TIMx_CCMRn 寄存器中 CCnS=00）的输出比较中断标志位才会被置 1。 10: 中央对齐模式 2，计数器以交替方式向上或向下计数。仅当计数器向上计数时，配置为输出的通道（TIMx_CCMRn 寄存器中 CCnS=00）的输出比较中断标志位才会被置 1。 11: 中央对齐模式 3，计数器以交替方式向上或向下计数。当计数器向上或向下计数时，配置为输出的通道（TIMx_CCMRn 寄存器中 CCnS=00）的输出比较中断标志位均会被置 1。 注意：当计数器使能时（CEN=1），不允许从边沿对齐模式转换到中央对齐模式。
Bit 4	DIR: 计数器计数方向选择位 0: 计数器向上计数 1: 计数器向下计数 注意：当计数器配置为中央对齐模式或者编码器模式时，该位只可读。
Bit 3	OPM: 单脉冲模式 0: 当发生更新事件时，计数器不停止。 1: 当发生下一次更新事件（CEN 位清零）时，计数器停止。
Bit 2	URS: 更新请求源 该位由软件置 1 或清零，来选择 UEV 事件源。 0: 如果更新中断使能，则下述任一事件都可产生更新中断： <ul style="list-style-type: none">- 计数器上溢/下溢- 软件设置 UG 位为 1- 从模式控制器产生的更新 1: 如果更新中断使能，仅计数器上溢/下溢才能产生更新中断
Bit 1	UDIS: 更新禁止 该位由软件置 1 或清零来使能/禁止 UEV 事件的产生。 0: UEV 使能，更新事件(UEV)由下列任一事件产生： <ul style="list-style-type: none">- 计数器上溢/下溢- 软件设置 UG 位为 1- 从模式控制器产生的更新 缓冲寄存器载入它们的预载值。 1: UEV 禁止，不产生更新事件，影子寄存器保持它们的值(ARR, PSC, CCRn)。如果从模式控制器接收到硬件复位，计数器和预分频器将被重新初始化。
Bit 0	CEN: 计数器使能位 0: 计数器禁止 1: 计数器使能 注意：如果软件设置了 CEN 位，外部时钟，门控模式和编码器模式才能工作。触发模式可由硬件自动设置 CEN 位。

6.2.6.2 控制寄存器 1 高 8 位 (TIM1_CR1H)

TIMx_CR1H: 控制寄存器 1 (TIM1_CR1H)								
Bit	7	6	5	4	3	2	1	0
Name	—	HTOE OFF	—	—	—	CMPSELP	CKD<1:0>	
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 未使用，必须保持复位值

Bit 6 HTOE_{OFF}: 在 ICD HALT 模式下关断 PWM 输出

0: 关断无效

1: 关断 PWM

Bit 5~3 未使用，必须保持复位值

Bit 2 CMPSELP: 比较输入极性选择位

0: 比较输入为低有效

1: 比较输入为高有效

Bit 1~0 CKD<1:0>: 时钟分频

该位域为计数器时钟(CK_INT)频率，死区时间和由死区发生器与数字滤波器(ETR, TI_x)所用的采样时钟(t_{DTS})之间的分频关系。

00: $t_{DTS}=t_{CK_INT}$

01: $t_{DTS}=2*t_{CK_INT}$

10: $t_{DTS}=4*t_{CK_INT}$

11: 预留，禁止使用

6.2.6.3 控制寄存器 2 低 8 位 (TIM1_CR2L)

TIMx_CR2L: 控制寄存器 2 (TIM1_CR2L)								
Bit	7	6	5	4	3	2	1	0
Name	—	MMS<2:0>			—	CCUS	—	CCPC
R/W	R/W	R/W	R/W	R/W	—	R/W	—	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 保留未用,必须保持复位值'0'

Bit 6~4 MMS<2:0>: 主模式选择

该位域可选择在主模式下发送到从计数器的同步信息 (TRGO)。

000: 复位 – TIMx_EGRL 寄存器中的 UG 位被采用为触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置复位模式)，则相较与实际复位，TRGO 上的信号将会延迟。

001: 使能 – 计数器使能信号 CNT_EN 被用作触发输出(TRGO)。在从计数器使能的情况下，该设置用于在同一时间启动数次或者用来控制窗口。计数器使能信号是由 CEN 控制位与配置为门控模式的触发输入进行逻辑或操作产生的。当计数器使能信号由触发输入控制，TRGO 上会产生延迟，除非被选为主/从模式 (参考 TIMx_SMCR 寄存器中的 MSM 位的描述)。

010: 更新 – 更新事件被选为触发输出(TRGO)。举例，主计数器可被用作从计数器的预分频器。

	011: 比较脉冲 – 一旦捕捉或者比较匹配发生, 当 CC1IF 标志位被置起 (即便已为高电平), 触发输出会发送一个正脉冲(TRGO)。
	100: 比较 - OC1REF 信号用作触发输出(TRGO)
	101: 比较- OC2REF 信号用作触发输出 (TRGO)
	110: 比较- OC3REF 信号用作触发输出 (TRGO)
	111: 比较- OC4REF 信号用作触发输出 (TRGO)
Bit 3	预留, 必须保持复位值。
Bit 2	CCUS: 捕捉/比较控制更新选择位 0: 当捕捉/比较控制位为预载值(CCPC=1), 则当设置 COMG 位为 1 时这些控制位才会被更新。 1: 当捕捉/比较控制位为预载值(CCPC=1), 则当设置 COMG 位为 1 或者当 TRGI 发生上升沿时这些控制位才会被更新。 注意: 该位只有用作于互补输出的通道时才有效。
Bit 1	预留, 必须保持复位值。
Bit 0	CCPC: 捕捉/比较预载控制位 0: CCnE, CCnNE 和 OCnM 不为预载值。 1: CCnE, CCnNE 和 OCnM 为预载值。当写入预载值后, 仅当发生换相事件(COM)时 (COMG 位设置为 1 或者 TRGI 上检测到上升沿由 CCUS 位决定), 才会被更新。 注意: 该位只有用作于互补输出的通道时才有效。

6.2.6.4 控制寄存器 2 高 8 位 (TIM1_CR2H)

TIMx_CR2H: 控制寄存器 2 (TIM1_CR2H)								
Bit	7	6	5	4	3	2	1	0
Name	—	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
R/W	—	R/W						
POR	0	0	0	0	0	0	0	0
Bit 7	预留, 必须保持复位值。							
Bit 6	OIS4: 输出空闲状态 4 (OC4 输出) 0: 当 MOE=0, 如果 OC4N 有效, 在一段死区时间后, OC4=0 1: 当 MOE=0, 如果 OC4N 有效, 在一段死区时间后, OC4=1							
Bit 5	OIS3N: 输出空闲状态 3 (OC3N 输出) 0: 当 MOE=0, 在一段死区时间后, OC3N=0 1: 当 MOE=0, 在一段死区时间后, OC3N=1 注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 1,2,或 3 后, OIS3N 不可更改。							
Bit 4	OIS3: 输出空闲状态 3 (OC3 输出) 0: 当 MOE=0, 如果 OC3N 有效, 在一段死区时间后, OC3=0 1: 当 MOE=0, 如果 OC3N 有效, 在一段死区时间后, OC3=1							
Bit 3	OIS2N: 输出空闲状态 2 (OC2N 输出) 0: 当 MOE=0, 在一段死区时间后, OC2N=0 1: 当 MOE=0, 在一段死区时间后, OC2N=1 注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 1,2,或 3 后, OIS2N 不可更改。							

Bit 7	预留, 必须保持复位值。
Bit 6	OIS4: 输出空闲状态 4 (OC4 输出) 0: 当 MOE=0, 如果 OC4N 有效, 在一段死区时间后, OC4=0 1: 当 MOE=0, 如果 OC4N 有效, 在一段死区时间后, OC4=1
Bit 5	OIS3N: 输出空闲状态 3 (OC3N 输出) 0: 当 MOE=0, 在一段死区时间后, OC3N=0 1: 当 MOE=0, 在一段死区时间后, OC3N=1 注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 1,2,或 3 后, OIS3N 不可更改。
Bit 4	OIS3: 输出空闲状态 3 (OC3 输出) 0: 当 MOE=0, 如果 OC3N 有效, 在一段死区时间后, OC3=0 1: 当 MOE=0, 如果 OC3N 有效, 在一段死区时间后, OC3=1
Bit 3	OIS2N: 输出空闲状态 2 (OC2N 输出) 0: 当 MOE=0, 在一段死区时间后, OC2N=0 1: 当 MOE=0, 在一段死区时间后, OC2N=1 注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 1,2,或 3 后, OIS2N 不可更改。

Bit 2	OIS2: 输出空闲状态 2 (OC2 输出)
	0: 当 MOE=0, 如果 OC2N 有效, 在一段死区时间后, OC2=0
	1: 当 MOE=0, 如果 OC2N 有效, 在一段死区时间后, OC2=1
Bit 1	OIS1N: 输出空闲状态 1 (OC1N 输出)
	0: 当 MOE=0, 在一段死区时间后, OC1N=0
	1: 当 MOE=0, 在一段死区时间后, OC1N=1
	注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 1,2, 或 3 后, OIS1N 不可更改。
Bit 0	OIS1: 输出空闲状态 1 (OC1 输出)
	0: 当 MOE=0, 如果 OC1N 有效, 在一段死区时间后, OC1=0
	1: 当 MOE=0, 如果 OC1N 有效, 在一段死区时间后, OC1=1

6.2.6.5 从模式控制寄存器低 8 位 (TIM1_SMCRL)

TIMx_SMCRL: 从模式控制寄存器 (TIM1_SMCRL)								
Bit	7	6	5	4	3	2	1	0
Name	MSM	TS<2:0>			OCCS	SMS<2:0>		
R/W	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	MSM: 主/从模式
	0: 无动作
	1: 延迟触发输入 (TRGI) 上的事件来允许当前计时器和其从器件 (通过 TRGO) 之间的完美同步。该设置有效用于使用单个外部事件来同步多个计时器。
Bit 6~4	TS<2:0>: 触发选择
	该位域用来选择不同的触发输入来同步计数器。
	000~011: 保留
	100: TI1 边沿检测器(TI1F_ED)
	101: 滤波计时器输入 1 (TI1FP1)
	110: 滤波计时器输入 2 (TI2FP2)
	111: 外部触发输入(ETRF)
	注意: 为了避免错误边沿检测, 该位域在不使用时 (SMS=000) 才能改变。
Bit 3	OCCS: 外部事件清除 OCnREF 信号使能位
	0: 禁止
	1: 使能
Bit 2~0	SMS<2:0>: 从模式选择
	当选择外部信号, 触发信号 TRGI 的有效边沿与外部输入的极性有关系 (详见输入控制寄存器和控制寄存器描述)
	000: 禁止从模式 - 如果 CEN = '1', 则预分频器直接由内部时钟计数。
	001: 编码器模式 1 - 计数器向上/向下计数 TI2FP1 边沿, 取决于 TI1FP2 电平
	010: 编码器模式 2 - 计数器向上/向下计数 TI1FP2 边沿, 取决于 TI2FP1 电平
	011: 编码器模式 3 - 计数器向上/向下计数 TI1FP1 和 TI2FP2 边沿, 取决于另一个输入的电平
	100: 复位模式 - 选中的触发输入的上升沿重新初始化计数器, 并生成寄存器的更新信号

101: 门控模式 – 当触发输入 TRGI 为高电平, 计数器时钟使能。一旦触发变为低电平, 计数器停止计数 (并非复位)。计数器的启动和停止均受控制。

110: 触发模式 – 计数器在触发信号 TRGI 的上升沿处启动 (不复位)。仅计数器的启动受控制。

111: 外部时钟模式 1 – 计数器在触发信号 TRGI 的上升沿计数

注意: 如果 TI1F_ED 被选为触发输入 (TS='100'), 不能使用门控模式。TI1F 每一次转换, TI1F_ED 就会输出 1 个脉冲, 而门控模式则是检查触发信号的电平。

注意: 在发生来自主计时器的接收事件之前, 从计时器的时钟必须先使能, 且在接收来自主计时器的触发过程中, 从计数器时钟不能即时更改。

6.2.6.6 从模式控制寄存器高 8 位 (TIM1_SMCRH)

TIMx_SMCRH: 从模式控制寄存器 (TIM1_SMCRH)								
Bit	7	6	5	4	3	2	1	0
Name	ETP	ECE	ETPS<1:0>			ETF<3:0>		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	ETP: 外部触发极性
	0: 正向 ETR, 即高电平有效或上升沿有效
	1: 反向 ETR, 即低电平有效或下降沿有效
Bit 6	ECE: 外部时钟使能
	该位使能外部时钟模式 2
	0: 禁止外部时钟模式 2
	1: 使能外部时钟模式 2, 计数器由 ETRF 信号上的有效边沿计数。
	注意:
	1: 设置 ECE 位为 1 的效果与选择外部时钟模式 1 且 TRGI 与 ETRF 相连接 (SMS=111 和 TS=111) 相同。
	2: 可同时使用外部时钟模式 2 与下列从模式: 复位模式, 门控模式和触发模式。在这种情况下, TRGI 不能与 ETRF 相连接 (TS 不能设置为 111)。
	3: 如果外部时钟模式 1 和外部时钟模式 2 同时使能, 外部时钟输入为 ETRF。
Bit 5~4	ETPS<1:0>: 外部触发预分频器
	外部触发信号 ETRP 频率最多为 TIMxCLK 频率的 1/4。可使能预分频器来降低 ETRP 的频率。该位域有效用于输入高速外部时钟的情况。
	00: 预分频器关闭
	01: ETRP 频率 2 分频
	10: ETRP 频率 4 分频
	11: ETRP 频率 8 分频
Bit 3~0	ETF<3:0>: 外部触发滤波器
	该位域定义了 ETRP 信号的采样频率和数字滤波器的滤波长度。
	数字滤波器由一个事件计数器组成, 每 N 个连续事件才视为一个有效边沿
	0000: 无滤波器, 采样频率为 f _{DTS}
	0001: f _{SAMPLING} = f _{CK_INT} , N = 2
	0010: f _{SAMPLING} = f _{CK_INT} , N = 4
	0011: f _{SAMPLING} = f _{CK_INT} , N = 8

- 0100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2, N = 6$
 0101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2, N = 8$
 0110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4, N = 6$
 0111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4, N = 8$
 1000: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8, N = 6$
 1001: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8, N = 8$
 1010: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 5$
 1011: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 6$
 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 8$
 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 5$
 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 6$
 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 8$

注意：当 $\text{ETF} < 3:0 > = 1, 2 \text{ or } 3$ 时，公式中的 f_{DTS} 由 CK_INT 取代。

6.2.6.7 中断使能寄存器低 8 位 (TIM1_IERL)

TIMx_IERL: 中断使能寄存器 (TIM1_IERL)								
Bit	7	6	5	4	3	2	1	0
Name	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0
Bit 7	BIE: 刹车中断使能位							
	0: 禁止							
	1: 使能刹车中断							
Bit 6	TIE: 触发中断使能位							
	0: 禁止							
	1: 使能触发中断							
Bit 5	COMIE: COM 中断使能位							
	0: 禁止							
	1: 使能 COM 中断							
Bit 4	CC4IE: 捕捉/比较 4 中断使能位							
	0: 禁止							
	1: 使能 CC4 中断							
Bit 3	CC3IE: 捕捉/比较 3 中断使能位							
	0: 禁止							
	1: 使能 CC3 中断							
Bit 2	CC2IE: 捕捉/比较 2 中断使能位							
	0: 禁止							
	1: 使能 CC2 中断							
Bit 1	CC1IE: 捕捉/比较 1 中断使能位							
	0: 禁止							
	1: 使能 CC1 中断							
Bit 0	UIE: 更新中断使能位							

- 0: 禁止
1: 使能更新中断

6.2.6.8 中断使能寄存器高 8 位 (TIM1_IERH)

TIMx_IERH: 中断使能寄存器 (TIM1_IERH)								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CC4OIE	CC3OIE	CC2OIE	CC1OIE	—
R/W	—	—	—	W	W	W	W	—
POR	0	0	0	0	0	0	0	0

- Bit 7~5 预留, 必须保持复位值
Bit 4 CC4OIE: 捕捉/比较 4 捕获溢出中断使能位
0: 禁止
1: 使能 CC4 捕获溢出中断
Bit 3 CC3OIE: 捕捉/比较 3 捕获溢出中断使能位
0: 禁止
1: 使能 CC3 捕获溢出中断
Bit 2 CC2OIE: 捕捉/比较 2 捕获溢出中断使能位
0: 禁止
1: 使能 CC2 捕获溢出中断
Bit 1 CC1OIE: 捕捉/比较 1 捕获溢出中断使能位
0: 禁止
1: 使能 CC1 捕获溢出中断
Bit 0 预留, 必须保持复位值

6.2.6.9 中断禁止寄存器低 8 位 (TIM1_IDRL)

TIMx_IDRL: 中断禁止寄存器 (TIM1_IDRL)								
Bit	7	6	5	4	3	2	1	0
Name	BID	TID	COMID	CC4ID	CC3ID	CC2ID	CC1ID	UID
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

- Bit 7 BID: 刹车中断禁止
0: 无效
1: 禁止刹车中断
Bit 6 TID: 触发中断禁止
0: 无效
1: 禁止触发中断
Bit 5 COMID: COM 中断禁止
0: 无效
1: 禁止 COM 中断
Bit 4 CC4ID: 捕捉/比较 4 中断禁止
0: 无效
1: 禁止 CC4 中断

Bit 3	CC3ID: 捕捉/比较 3 中断禁止 0: 无效 1: 禁止 CC3 中断
Bit 2	CC2ID: 捕捉/比较 2 中断禁止 0: 无效 1: 禁止 CC2 中断
Bit 1	CC1ID: 捕捉/比较 1 中断禁止 0: 无效 1: 禁止 CC1 中断
Bit 0	UID: 更新中断禁止 0: 无效 1: 禁止更新中断

6.2.6.10 中断禁止寄存器高 8 位 (TIM1_IDRH)

TIMx_IDRH: 中断禁止寄存器 (TIM1_IDRH)								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CC4OID	CC3OID	CC2OID	CC1OID	—
R/W	—	—	—	W	W	W	W	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	预留, 必须保持复位值
Bit 4	CC4OID: 捕捉/比较 4 捕获溢出中断禁止 0: 无效 1: 禁止 CC4 捕获溢出中断
Bit 3	CC3OID: 捕捉/比较 3 捕获溢出中断禁止 0: 无效 1: 禁止 CC3 捕获溢出中断
Bit 2	CC2OID: 捕捉/比较 2 捕获溢出中断禁止 0: 无效 1: 禁止 CC2 捕获溢出中断
Bit 1	CC1OID: 捕捉/比较 1 捕获溢出中断禁止 0: 无效 1: 禁止 CC1 捕获溢出中断
Bit 0	预留, 必须保持复位值

6.2.6.11 中断有效状态寄存器低 8 位 (TIM1_IVSL)

TIMx_IVSL: 中断有效状态寄存器 (TIM1_IVSL)								
Bit	7	6	5	4	3	2	1	0
Name	BIS	TIS	COMIS	CC4IS	CC3IS	CC2IS	CC1IS	UIS
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7 BIS: 刹车中断有效状态

	0: 刹车中断禁止 1: 刹车中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 6	TIS: 触发中断有效状态 0: 触发中断禁止 1: 触发中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 5	COMIS: COM 中断有效状态 0: COM 中断禁止 1: COM 中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 4	CC4IS: 捕捉/比较 4 中断有效状态 0: CC4 中断禁止 1: CC4 中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 3	CC3IS: 捕捉/比较 3 中断有效状态 0: CC3 中断禁止 1: CC3 中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 2	CC2IS: 捕捉/比较 2 中断有效状态 0: CC2 中断禁止 1: CC2 中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 1	CC1IS: 捕捉/比较 1 中断有效状态 0: CC1 中断禁止 1: CC1 中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 0	UIS: 更新中断有效状态 0: 更新中断禁止 1: 更新中断使能 对 IER/IDR 写 1 来使能或禁止该位

6.2.6.12 中断有效状态寄存器高 8 位 (TIM1_IVSH)

TIMx_IVSH: 中断有效状态寄存器 (TIM1_IVSH)								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CC4OIS	CC3OIS	CC2OIS	CC1OIS	—
R/W	—	—	—	R	R	R	R	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	预留, 必须保持复位值
Bit 4	CC4OIS: 捕捉/比较 4 捕获溢出中断有效状态 0: CC4 捕获溢出中断禁止 1: CC4 捕获溢出中断使能 对 IER/IDR 写 1 来使能或禁止该位

Bit 3	CC3OIS: 捕捉/比较 3 捕获溢出中断有效状态 0: CC3 捕获溢出中断禁止 1: CC3 捕获溢出中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 2	CC2OIS: 捕捉/比较 2 捕获溢出中断有效状态 0: CC2 捕获溢出中断禁止 1: CC2 捕获溢出中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 1	CC1OIS: 捕捉/比较 1 捕获溢出中断有效状态 0: CC1 捕获溢出中断禁止 1: CC1 捕获溢出中断使能 对 IER/IDR 写 1 来使能或禁止该位
Bit 0	预留, 必须保持复位值

6.2.6.13 中断标志寄存器低 8 位 (TIM1_RIFL)

TIMx_RIFL: 中断标志寄存器 (TIM1_RIFL)								
Bit	7	6	5	4	3	2	1	0
Name	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7	BIF: 刹车中断标志 如果刹车中断使能, 一旦刹车输入有效, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。 0: 未发生刹车事件 1: 在刹车输入上检测到有效电平
Bit 6	TIF: 触发中断标志 如果触发中断使能, 当从模式控制器在门控模式以外的所有模式下使能, 发生触发事件时 (TRGI 上检测到有效边沿), 该标志位被硬件置 1。对 TIMx_ICR 写 1 来清 0。 0: 未发生触发事件 1: 触发中断等待响应
Bit 5	COMIF: COM 中断标志 如果 COM 中断使能, 当发生 COM 事件(当捕捉/比较控制位 CCnE, CCnNE, OCnM 更新后), 该标志位被硬件置 1。对 TIMx_ICR 写 1 来清 0。 0: 未发生 COM 事件 1: COM 中断等待响应
Bit 4	CC4IF: 捕捉/比较 4 中断标志 如果 CC4 通道配置为输出: 如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。 0: 不匹配 1: TIMx_CNT 计数值与 TIMx_CCR4 值匹配。当 TIMx_CCR4 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模

式), CC4IF 为被置 1。

如果 CC4 通道配置为输入:

发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR4 寄存器来清零。

0: 未发生输入捕捉

1: 计数值捕捉至 TIMx_CCR4 寄存器 (IC4 上检测到与选中极性匹配的边沿)

Bit 3

CC3IF: 捕捉/比较 3 中断标志

如果 CC3 通道配置为输出:

如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。

0: 不匹配

1: TIMx_CNT 计数值与 TIMx_CCR3 值匹配。当 TIMx_CCR3 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC3IF 为被置 1。

如果 CC3 通道配置为输入:

发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR3 寄存器来清零。

0: 未发生输入捕捉

1: 计数值捕捉至 TIMx_CCR3 寄存器 (IC3 上检测到与选中极性匹配的边沿)

Bit 2

CC2IF: 捕捉/比较 2 中断标志

如果 CC2 通道配置为输出:

如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。

0: 不匹配

1: TIMx_CNT 计数值与 TIMx_CCR2 值匹配。当 TIMx_CCR2 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC2IF 为被置 1。

如果 CC2 通道配置为输入:

发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR2 寄存器来清零。

0: 未发生输入捕捉

1: 计数值捕捉至 TIMx_CCR2 寄存器 (IC2 上检测到与选中极性匹配的边沿)

Bit 1

CC1IF: 捕捉/比较 1 中断标志

如果 CC1 通道配置为输出:

如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。

0: 不匹配

1: TIMx_CNT 计数值与 TIMx_CCR1 值匹配。当 TIMx_CCR1 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC1IF 为被置 1。

如果 CC1 通道配置为输入:

发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR1 寄存器来清零。

Bit 0	0: 未发生输入捕捉
	1: 计数值捕捉至 TIMx_CCR1 寄存器 (IC1 上检测到与选中极性匹配的边沿)
UIF:	更新中断标志 如果更新中断使能, 当发生更新事件, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0
0:	未发生更新
1:	更新中断被挂起. 当寄存器更新时, 该位被硬件置 1: -当重复计数器值发生上溢或者下溢 (若重复计数器=0, 则更新) 和当 TIMx_CR1 寄存器中 UDIS=0 -当使用 TIMx_EGR 寄存器中的 UG 位来由软件重新初始化 CNT 时, 如果 TIMx_CR1 寄存中的 URS=0 和 UDIS=0 -当 CNT 由触发事件来重新初始化, 如果 TIMx_CR1 寄存中的 URS=0 和 UDIS=0

6.2.6.14 中断标志寄存器高 8 位 (**TIM1_RIFH**)

TIMx_RIFH: 中断标志寄存器 (TIM1_RIFH)								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CC4OIF	CC3OIF	CC2OIF	CC1OIF	—
R/W	—	—	—	R	R	R	R	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	预留, 必须保持复位值
Bit 4	CC4OIF: 捕捉/比较 4 捕获溢出中断标志 仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件设置。对 TIMx_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CC4IF 标志位置 1 时, 捕捉计数器值至 TIMx_CCR4 寄存器
Bit 3	CC3OIF: 捕捉/比较 3 捕获溢出中断标志 仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件设置。对 TIMx_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CC3IF 标志位置 1 时, 捕捉计数器值至 TIMx_CCR3 寄存器
Bit 2	CC2OIF: 捕捉/比较 2 捕获溢出中断标志 仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件设置。对 TIMx_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CC2IF 标志位置 1 时, 捕捉计数器值至 TIMx_CCR2 寄存器
Bit 1	CC1OIF: 捕捉/比较 1 捕获溢出中断标志 仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件设置。对 TIMx_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CC1IF 标志位置 1 时, 捕捉计数器值至 TIMx_CCR1 寄存器
Bit 0	预留, 必须保持复位值

6.2.6.15 中断标志屏蔽寄存器低 8 位 (TIM1_IFML)

TIMx_IFML: 中断标志屏蔽 (TIM1_IFML)								
Bit	7	6	5	4	3	2	1	0
Name	BIM	TIM	COMIM	CC4IM	CC3IM	CC2IM	CC1IM	UIM
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0
Bit 7	BIM: 刹车中断标志屏蔽							
	如果刹车中断使能, 一旦刹车输入有效, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。							
	0: 未发生刹车事件 1: 在刹车输入上检测到有效电平							
Bit 6	TIM: 触发中断标志屏蔽							
	如果触发中断使能, 当从模式控制器在门控模式以外的所有模式下使能, 发生触发事件时 (TRGI 上检测到有效边沿), 该标志位被硬件置 1。对 TIMx_ICR 写 1 来清 0。							
	0: 未发生触发事件 1: 触发中断被挂起							
Bit 5	COMIM: COM 中断标志屏蔽							
	如果 COM 中断使能, 当发生 COM 事件(当捕捉/比较控制位 CCnE, CCnNE, OCnM 更新后), 该标志位被硬件置 1。对 TIMx_ICR 写 1 来清 0。							
	0: 未发生 COM 事件 1: COM 中断被挂起							
Bit 4	CC4IM: 捕捉/比较 4 中断标志屏蔽							
	如果 CC4 通道配置为输出: 如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。							
	0: 不匹配 1: TIMx_CNT 计数值与 TIMx_CCR4 值匹配。当 TIMx_CCR4 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC4IF 为被置 1。 如果 CC4 通道配置为输入: 发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR4 寄存器来清零。							
	0: 未发生输入捕捉 1: 计数值捕捉至 TIMx_CCR4 寄存器 (IC4 上检测到与选中极性匹配的边沿)							
Bit 3	CC3IM: 捕捉/比较 3 中断标志屏蔽							
	如果 CC3 通道配置为输出: 如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。							
	0: 不匹配 1: TIMx_CNT 计数值与 TIMx_CCR3 值匹配。当 TIMx_CCR3 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC3IF 为被置 1。							

	<p>如果 CC3 通道配置为输入:</p> <p>发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR3 寄存器来清零。</p> <p>0: 未发生输入捕捉 1: 计数值捕捉至 TIMx_CCR3 寄存器 (IC3 上检测到与选中极性匹配的边沿)</p>
Bit 2	<p>CC2IM: 捕捉/比较 2 中断标志屏蔽</p> <p>如果 CC2 通道配置为输出:</p> <p>如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。</p> <p>0: 不匹配 1: TIMx_CNT 计数值与 TIMx_CCR2 值匹配。当 TIMx_CCR2 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC2IF 为被置 1。</p> <p>如果 CC2 通道配置为输入:</p> <p>发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR2 寄存器来清零。</p> <p>0: 未发生输入捕捉 1: 计数值捕捉至 TIMx_CCR2 寄存器 (IC2 上检测到与选中极性匹配的边沿)</p>
Bit 1	<p>CC1IM: 捕捉/比较 1 中断标志屏蔽</p> <p>如果 CC1 通道配置为输出:</p> <p>如果中断使能, 除去中央对齐模式的情况(参考 TIMx_CR1 寄存器中 CMS 的描述), 当计数值与比较值匹配, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。</p> <p>0: 不匹配 1: TIMx_CNT 计数值与 TIMx_CCR1 值匹配。当 TIMx_CCR1 寄存器值大于 TIMx_ARR 值, 发生计数器上溢时 (递增模式和递增/递减模式) 或下溢时 (递减模式), CC1IF 为被置 1。</p> <p>如果 CC1 通道配置为输入:</p> <p>发生捕捉时, 该位由硬件置 1。该位可通过软件或者读取 TIMx_CCR1 寄存器来清零。</p> <p>0: 未发生输入捕捉 1: 计数值捕捉至 TIMx_CCR1 寄存器 (IC1 上检测到与选中极性匹配的边沿)</p>
Bit 0	<p>UIM: 更新中断标志屏蔽</p> <p>如果更新中断使能, 当发生更新事件, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清除原始中断。</p> <p>0: 未发生更新 1: 更新中断被挂起。当寄存器更新时, 该位被硬件置 1: -当重复计数器值发生上溢或者下溢 (若重复计数器=0, 则更新) 和当 TIMx_CR1 寄存器中 UDIS=0 -当使用 TIMx_EGR 寄存器中的 UG 位来由软件重新初始化 CNT 时, 如果 TIMx_CR1 寄存中的 URS=0 和 UDIS=0 -当 CNT 由触发事件来重新初始化, 如果 TIMx_CR1 寄存中的 URS=0 和 UDIS=0</p>

6.2.6.16 中断标志屏蔽寄存器高 8 位 (TIM1_IFMH)

TIMx_IFMH: 中断标志屏蔽 (TIM1_IFMH)								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CC4OIM	CC3OIM	CC2OIM	CC1OIM	—
R/W	—	—	—	R	R	R	R	—
POR	0	0	0	0	0	0	0	0

- Bit 7~5 预留, 必须保持复位值
- Bit 4 CC4OIM: 捕捉/比较 4 捕获溢出中断标志屏蔽
仅当相应的通道配置为捕捉输入状态时, 该标志位由硬件置 1。对 TIMx_ICR 写 1 来清 0。
0: 未检测到捕获溢出
1: 当 CC4IF 标志为置 1 时, 捕捉计数器值至 TIMx_CCR1 寄存器
- Bit 3 CC3OIM: 捕捉/比较 3 捕获溢出中断标志屏蔽
仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件置 1。对 TIMx_ICR 写 1 来清 0。
0: 未检测到捕获溢出
1: 当 CC3IF 标志为置 1 时, 捕捉计数器值至 TIMx_CCR1 寄存器
- Bit 2 CC2OIM: 捕捉/比较 2 捕获溢出中断标志屏蔽
仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件置 1。对 TIMx_ICR 写 1 来清 0。
0: 未检测到捕获溢出
1: 当 CC2IF 标志为置 1 时, 捕捉计数器值至 TIMx_CCR1 寄存器
- Bit 1 CC1OIM: 捕捉/比较 1 捕获溢出中断标志屏蔽
仅当相应的通道配置为捕捉输入状态时, 该标志位才由硬件置 1。对 TIMx_ICR 写 1 来清 0。
0: 未检测到捕获溢出
1: 当 CC1IF 标志为置 1 时, 捕捉计数器值至 TIMx_CCR1 寄存器
- Bit 0 预留, 必须保持复位值

6.2.6.17 中断清零寄存器低 8 位 (TIM1_ICRL)

TIMx_ICRL: 中断清零寄存器 (TIM1_ICRL)								
Bit	7	6	5	4	3	2	1	0
Name	BIC	TIC	COMIC	CC4IC	CC3IC	CC2IC	CC1IC	UIC
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

- Bit 7 BIC: 刹车中断清零
0: 无效
1: 刹车中断清零 (TIMx_RIF), 写 1 清零
- Bit 6 TIC: 触发中断清零
0: 无效
1: 触发中断清零 (TIMx_RIF) , 写 1 清零
- Bit 5 COMIC: COM 中断清零

	0: 无效 1: COM 中断清零 (TIMx_RIF) , 写 1 清零
Bit 4	CC4IC: 捕捉/比较 4 中断清零 0: 无效 1: 捕捉/比较中断清零 (TIMx_RIF) , 写 1 清零
Bit 3	CC3IC: 捕捉/比较 3 中断清零 0: 无效 1: 捕捉/比较中断清零 (TIMx_RIF) , 写 1 清零
Bit 2	CC2IC: 捕捉/比较 2 中断清零 0: 无效 1: 捕捉/比较中断清零 (TIMx_RIF) , 写 1 清零
Bit 1	CC1IC: 捕捉/比较 1 中断清零 0: 无效 1: 捕捉/比较中断清零 (TIMx_RIF) , 写 1 清零
Bit 0	UIC: 更新中断清零 0: 无效 1: 更新中断清零 (TIMx_RIF) , 写 1 清零

6.2.6.18 中断清零寄存器高 8 位 (TIM1_ICRH)

TIMx_ICRH: 中断清零寄存器 (TIM1_ICRH)								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CC4OIC	CC3OIC	CC2OC	CC1OC	—
R/W	—	—	—	W	W	W	W	—
POR	0	0	0	0	0	0	0	0
Bit 7~5	预留, 必须保持复位值							
Bit 4	CC4OIC: 捕捉/比较 4 捕获溢出中断清零 0: 无效 1: CC4OI 中断清零 (TIMx_RIF) , 写 1 清零							
Bit 3	CC3OIC: 捕捉/比较 3 捕获溢出中断清零 0: 无效 1: CC3OI 中断清零 (TIMx_RIF) , 写 1 清零							
Bit 2	CC2OIC: 捕捉/比较 2 捕获溢出中断清零 0: 无效 1: CC2OI 中断清零 (TIMx_RIF) , 写 1 清零							
Bit 1	CC1OIC: 捕捉/比较 1 捕获溢出中断清零 0: 无效 1: CC1OI 中断清零 (TIMx_RIF) , 写 1 清零							
Bit 0	预留, 必须保持复位值							

Bit 7~5	预留, 必须保持复位值
Bit 4	CC4OIC: 捕捉/比较 4 捕获溢出中断清零 0: 无效 1: CC4OI 中断清零 (TIMx_RIF) , 写 1 清零
Bit 3	CC3OIC: 捕捉/比较 3 捕获溢出中断清零 0: 无效 1: CC3OI 中断清零 (TIMx_RIF) , 写 1 清零
Bit 2	CC2OIC: 捕捉/比较 2 捕获溢出中断清零 0: 无效 1: CC2OI 中断清零 (TIMx_RIF) , 写 1 清零
Bit 1	CC1OIC: 捕捉/比较 1 捕获溢出中断清零 0: 无效 1: CC1OI 中断清零 (TIMx_RIF) , 写 1 清零
Bit 0	预留, 必须保持复位值

6.2.6.19 事件产生寄存器 (TIM1_EGRL)

TIMx_EGRL: 事件产生寄存器 (TIM1_EGRL)								
Bit	7	6	5	4	3	2	1	0
Name	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0
Bit 7	BG: 刹车生成 该位由软件设置来生成刹车事件，可由硬件自动清零。 0: 无动作 1: 产生刹车事件。MOE 清零，BIF 标志位置 1，产生相关中断							
Bit 6	TG: 触发生成 该位由软件设置来生成触发事件，可由硬件自动清零。 0: 无动作 1: TIMx_RIF 寄存器中的 TIF 被置 1，产生相关中断							
Bit 5	COMG: 捕捉/比较控制更新生成 该位由软件设置，由硬件自动清零。 0: 无动作 1: 当 CCPC 被置 1，则可更新 CCxE, CCxNE 和 OCxM 注意：该位只有用作于通道时才有互补输出。							
Bit 4	CC4G: 捕捉/比较 4 生成 该位由软件设置来生成事件，可由硬件自动清零。 0: 无动作 1: 通道 4 上产生捕捉/比较事件： 如果通道 4 配置为输出： CC4IF 标志位被置起，产生相应中断请求发送。 如果通道 4 配置为输入： 当前计数值捕捉至 TIMx_CCR1 寄存器。CC4IF 标志位被置 1，产生相应中断请求发送。如果 CC4IF 标志位为高电平，CC4OIF 标志位置 1。							
Bit 3	CC3G: 捕捉/比较 3 生成 该位由软件设置来生成事件，可由硬件自动清零。 0: 无动作 1: 通道 3 上产生捕捉/比较事件： 如果通道 3 配置为输出： CC3IF 标志位被置 1，产生相应中断请求发送。 如果通道 3 配置为输入： 当前计数值捕捉至 TIMx_CCR1 寄存器。CC3IF 标志位被置 1，产生相应中断请求发送。如果 CC3IF 标志位为高电平，CC3OIF 标志位置 1。							
Bit 2	CC2G: 捕捉/比较 2 生成 该位由软件设置来生成事件，可由硬件自动清零。 0: 无动作 1: 通道 2 上产生捕捉/比较事件： 如果通道 2 配置为输出： CC2IF 标志位被置起，产生相应中断请求发送。							

如果通道 2 配置为输入：

当前计数值捕捉至 TIMx_CCR1 寄存器。CC2IF 标志位被置 1，产生相应中断请求发送。如果 CC2IF 标志位为高电平，CC2OIF 标志位置 1。

Bit 1 CC1G：捕捉/比较 1 生成

该位由软件设置来生成事件，可由硬件自动清零。

0: 无动作

1: 通道 1 上产生捕捉/比较事件：

如果通道 1 配置为输出：

CC1IF 标志位被置起，产生相应中断请求发送。

如果通道 1 配置为输入：

当前计数值捕捉至 TIMx_CCR1 寄存器。CC1IF 标志位被置 1，产生相应中断请求发送。如果 CC1IF 标志位为高电平，CC1OIF 标志位置 1。

Bit 0 UG：更新生成

该位由软件设置，可由硬件自动清零。

0: 无动作

1: 重新初始化计数器，更新寄存器。注意，预分频器也会被清零（但预分频比不会受到影响）。如果使用中央对齐模式或者 DIR=0 (递增)，则计数器将清零；否则如果 DIR=1 (递减)，则将使用自动重载入值。

6.2.6.20 捕捉/比较模式寄存器 1 低 8 位 (TIM1_CCMR1L)

TIMx_CCMR1L: 捕捉/比较模式寄存器 1 (TIM1_CCMR1L)								
Bit	7	6	5	4	3	2	1	0
Name (Compare)	OC1CE	OC1M<2:0>			OC1PE	OC1FE	CC1S<1:0>	
Name (Capture)	IC1F<3:0>				IC1PSC<1:0>		CC1S<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

输出比较模式

Bit 7 OC1CE：输出比较 1 清零使能

0: OC1REF 不会受到 ETRF 输入影响

1: 当 ETRF 输入上检测到高电平时，OC1REF 将被清零

Bit 6~4 OC1M<2:0>：输出比较 1 模式

该位域定义了输出参考信号 OC1REF 的行为。OC1REF 为高有效，OC1 和 OC1N 的有效电平由 CC1P 和 CC1NP 位决定。

000: 冻结 – 输出比较寄存器 TIMx_CCR1 寄存器和 TIMx_CNT 计数器之间的比较对输出无效（该模式用来生成时基）。

001: 发生匹配时设置通道 1 为有效电平 – 当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR1 的值发生匹配后，OC1REF 信号强制为高电平。

010: 发生匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR1 的值发生匹配后，OC1REF 信号强制为低电平。

011: 翻转 - 当 TIMx_CNT=TIMx_CCR1，OC1REF 发生翻转。

100: 强制为无效电平 - OC1REF 强制为低电平。

101: 强制为有效电平 - OC1REF 强制为高电平。

110: PWM 模式 1 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR1, 通道 1 为有效电平, 否则, 通道 1 为无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR1, 通道 1 为无效电平 (OC1REF='0'), 否则通道 1 为有效电平 (OC1REF='1')。

111: PWM 模式 2 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR1, 通道 1 为无效电平, 否则, 通道 1 有无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR1, 通道 1 为有效电平, 否则通道 1 为无效电平。

注意:

1: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 3, 且 CC1S=00 (通道为输出模式), 该位域将不能更改。

2: 在 PWM 模式 1 和 2 中, 仅当比较结果更改或当输出比较模式从冻结模式转换成 PWM 模式, OCREF 电平才会更改。

3: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, OC1M 有效位才会设置为预载值中新的值。

Bit 3

OC1PE: 输出比较 1 预载使能

0: 禁止 TIMx_CCR1 寄存器的预载功能。TIMx_CCR1 在任何时候都可写, 新写入的值将立刻生效。

1: 使能 TIMx_CCR1 寄存器的预载功能。读/写操作可访问预载寄存器。每当发生一次更新事件, TIMx_CCR1 预载入值将会被填入当前寄存器。

注意:

1: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 3, 且 CC1S=00 (通道为输出模式), 该位域将不能更改。

2: 仅在单脉冲模式下 (TIMx_CR1 寄存器中的 OPM 设置为 1), PWM 模式可在不经过验证预载寄存器的情况下使用。其他情况下的行为不做保证。

Bit 2

OC1FE: 输出比较 1 高速使能

该位用来加速在 CC 输出上的输入触发事件的效应。

0: 当触发开启, CC1 运作正常取决于计数器和 CCR1 的值。当触发输入上发现边沿时, 至少需要 5 个时钟周期来激活 CC1 输出。

1: 触发输入上的有效沿类似于 CC1 输出上的比较匹配。设置 OC 为 1 用来比较电平, 采样触发输入和激活 CC1 输出的延时将会减少至 3 个时钟周期。只有当通道配置为 PWM1 或 PWM2 模式, OCFE 才会起作用。

Bit 1~0

CC1S<1:0>: 捕捉/比较 1 选择

该位域定义了通道和使用的输入的方向。

00: CC1 通道配置为输出

01: CC1 通道配置为输入, IC1 映射到 TI1

10: CC1 通道配置为输入, IC1 映射到 TI2

11: CC1 通道配置为输入, IC1 映射到 TRC

注意: 当通道关闭(TIMx_CCER 寄存器中的 CC1E = '0'), CC1S 为只写。

输入捕捉模式

Bit 7~4

IC1F<3:0>: 输入捕捉 1 滤波器

该位域定义了 TI1 输入的采样频率和数字滤波器的长度。

数字滤波器由一个事件计数器组成, 每 N 个连续事件才视为一个有效边沿:

0000: 无滤波器, 采样频率为 f_{DTS}
 0001: $f_{SAMPLING} = f_{CK_INT}$, $N = 2$
 0010: $f_{SAMPLING} = f_{CK_INT}$, $N = 4$
 0011: $f_{SAMPLING} = f_{CK_INT}$, $N = 8$
 0100: $f_{SAMPLING} = f_{DTS} / 2$, $N = 6$
 0101: $f_{SAMPLING} = f_{DTS} / 2$, $N = 8$
 0110: $f_{SAMPLING} = f_{DTS} / 4$, $N = 6$
 0111: $f_{SAMPLING} = f_{DTS} / 4$, $N = 8$
 1000: $f_{SAMPLING} = f_{DTS} / 8$, $N = 6$
 1001: $f_{SAMPLING} = f_{DTS} / 8$, $N = 8$
 1010: $f_{SAMPLING} = f_{DTS} / 16$, $N = 5$
 1011: $f_{SAMPLING} = f_{DTS} / 16$, $N = 6$
 1100: $f_{SAMPLING} = f_{DTS} / 16$, $N = 8$
 1101: $f_{SAMPLING} = f_{DTS} / 32$, $N = 5$
 1110: $f_{SAMPLING} = f_{DTS} / 32$, $N = 6$
 1111: $f_{SAMPLING} = f_{DTS} / 32$, $N = 8$

注意: 当 $ICxF<3:0> = 1, 2 \text{ or } 3$ 时, 公式中的 f_{DTS} 由 CK_INT 取代

Bit 3~2	IC1PSC<1:0>: 输入捕捉 1 预分频器	该位域定义了作用在 CC1 输入 (IC1) 上的预分频比。当 $CC1E='0'$ (TIMx_CCER 寄存器), 预分频器将复位。
	00: 无预分频器。每当捕捉输入上检测到边沿时, 发生捕捉动作。 01: 每发生 2 次事件, 执行一次捕捉 10: 每发生 4 次事件, 执行一次捕捉 11: 每发生 8 次事件, 执行一次捕捉	
Bit 1~0	CC1S<1:0>: 输入捕捉 1 选择	该位域定义了通道和使用的输入的方向。
	00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 10: CC1 通道配置为输入, IC1 映射到 TI2 11: CC1 通道配置为输入, IC1 映射到 TRC	

注意: 当通道关闭(TIMx_CCER 寄存器中的 $CC1E = '0'$), CC1S 为只写。

6.2.6.21 捕捉/比较模式寄存器 1 高 8 位 (TIM1_CCMR1H)

TIMx_CCMR1H: 捕捉/比较模式寄存器 1 (TIM1_CCMR1H)								
Bit	7	6	5	4	3	2	1	0
Name (Compare)	OC2CE	OC2M<2:0>			OC2PE	OC2FE	CC2S<1:0>	
Name (Capture)	IC2F<3:0>			IC2PSC<1:0>		CC2S<1:0>		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

输出比较模式

Bit 7 OC2CE: 输出比较 2 清零使能

Bit 6~4

0: OC2REF 不会受到 ETRF 输入影响

1: 当 ETRF 输入上检测到高电平时, OC2REF 将被清零

OC2M<2:0>: 输出比较 2 模式

该位域定义了输出参考信号 OC2REF 的行为。OC2REF 为高有效, OC2 和 OC2N 的有效电平由 CC2P 和 CC2NP 位决定。

000: 冻结 – 输出比较寄存器 TIMx_CCR2 寄存器和 TIMx_CNT 计数器之间的比较对输出无效 (该模式用来生成时基)。

001: 发生匹配时设置通道 2 为有效电平 – 当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR2 发生匹配后, OC2REF 信号强制为高电平。

010: 发生匹配时设置通道 2 为无效电平。当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR2 发生匹配后, OC2REF 信号强制为低电平。

011: 翻转 - 当 TIMx_CNT=TIMx_CCR2, OC2REF 发生翻转

100: 强制为无效电平 – OC2REF 强制为低电平

101: 强制为有效电平 – OC2REF 强制为高电平

110: PWM 模式 1 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR2, 通道 2 为有效电平, 否则, 通道 2 为无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR2, 通道 2 为无效电平 (OC2REF='0'), 否则通道 2 为有效电平 (OC2REF='1')。

111: PWM 模式 2 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR2, 通道 2 为无效电平, 否则, 通道 2 有无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR2, 通道 2 为有效电平, 否则通道 2 为无效电平。

注意:

1: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 3, 且 CC2S=00 (通道为输出模式), 该位域将不能更改。

2: 在 PWM 模式 1 和 2 中, 仅当比较结果更改或当输出比较模式从冻结模式转换成 PWM 模式, OCREF 电平才会更改。

3: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, OC2M 有效位才会设置为预载值中新的值。

Bit 3

OC2PE: 输出比较 2 预载使能

0: TIMx_CCR2 的预载寄存器禁止。TIMx_CCR2 在任何时候都可写, 新写入的值将立刻生效。

1: TIMx_CCR2 的预载寄存器禁止。读/写操作可访问预载寄存器。每当发生一次更新事件, TIMx_CCR2 预载入值将会被填入有效寄存器。

注意:

1: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 3, 且 CC2S=00 (通道为输出模式), 该位域将不能更改。

2: 仅在单脉冲模式下 (TIMx_CR1 寄存器中的 OPM 设置为 1), PWM 模式可在不经过验证预载寄存器的情况下使用。其他情况下的行为不做保证。

Bit 2

OC2FE: 输出比较 2 高速使能

该位用来加速在 CC 输出上的输入触发事件的效应。

0: 当触发开启, CC2 运作正常取决于计数器和 CCR2 的值。当触发输入上发现边沿时, 至少需要 5 个时钟周期来激活 CC2 输出。

1: 触发输入上的有效沿类似于 CC2 输出上的比较匹配。设置 OC 为 1 用来比较电

平，采样触发输入和激活 CC2 输出的延时将会减少至 3 个时钟周期。只有当通道配置为 PWM1 或 PWM2 模式，OCFE 才会起作用。

Bit 1~0 CC2S<1:0>: 输出比较 2 选择

该位域定义了通道以及使用的输入的方向（输入/输出）

00: CC2 通道配置为输出

01: CC2 通道配置为输入, IC2 映射到 TI2

10: CC2 通道配置为输入, IC2 映射到 TI1

11: CC2 通道配置为输入, IC2 映射到 TRC

注意：当通道为关闭状态时(TIMx_CCER 中 CC2E = '0')，CC2S 为只写。

输入捕捉模式

Bit 7~4 IC2F<3:0>: 输入捕捉 2 滤波器

该位域定义了 TI1 输入的采样频率和数字滤波器的长度。

数字滤波器由一个事件计数器组成，每 N 个连续事件才视为一个有效边沿：

0000: 无滤波器，采样频率为 f_{DTS}

0001: $f_{SAMPLING} = f_{CK_INT}$, $N = 2$

0010: $f_{SAMPLING} = f_{CK_INT}$, $N = 4$

0011: $f_{SAMPLING} = f_{CK_INT}$, $N = 8$

0100: $f_{SAMPLING} = f_{DTS} / 2$, $N = 6$

0101: $f_{SAMPLING} = f_{DTS} / 2$, $N = 8$

0110: $f_{SAMPLING} = f_{DTS} / 4$, $N = 6$

0111: $f_{SAMPLING} = f_{DTS} / 4$, $N = 8$

1000: $f_{SAMPLING} = f_{DTS} / 8$, $N = 6$

1001: $f_{SAMPLING} = f_{DTS} / 8$, $N = 8$

1010: $f_{SAMPLING} = f_{DTS} / 16$, $N = 5$

1011: $f_{SAMPLING} = f_{DTS} / 16$, $N = 6$

1100: $f_{SAMPLING} = f_{DTS} / 16$, $N = 8$

1101: $f_{SAMPLING} = f_{DTS} / 32$, $N = 5$

1110: $f_{SAMPLING} = f_{DTS} / 32$, $N = 6$

1111: $f_{SAMPLING} = f_{DTS} / 32$, $N = 8$

注意：当 ICxF <3:0> = 1, 2 or 3 时，公式中的 f_{DTS} 由 CK_INT 取代

Bit 3~2 IC2PSC<1:0>: 输入捕捉 2 预分频器

该位域定义了作用在 CC2 输入 (IC2) 上的预分频比。当 CC2E='0' (TIMx_CCER 寄存器)，预分频器将复位。

00: 无预分频器。每当捕捉输入上检测到边沿时，发生捕捉动作。

01: 每发生 2 次事件，执行一次捕捉

10: 每发生 4 次事件，执行一次捕捉

11: 每发生 8 次事件，执行一次捕捉

Bit 1~0 CC2S<1:0>: 输入捕捉 2 选择

该位域定义了通道和使用的输入的方向。

00: CC2 通道配置为输出

01: CC2 通道配置为输入, IC2 映射到 TI2

10: CC2 通道配置为输入, IC2 映射到 TI1

11: CC2 通道配置为输入, IC2 映射到 TRC

注意：当通道关闭(TIMx_CCER 寄存器中的 CC2E = '0')，CC2S 为只写。

6.2.6.22 捕捉/比较模式寄存器 2 低 8 位 (TIM1_CCMR2L)

TIMx_CCMR2L: 捕捉/比较模式寄存器 2 (TIM1_CCMR2L)								
Bit	7	6	5	4	3	2	1	0
Name (Compare)	OC3CE	OC3M<2:0>			OC3PE	OC3FE	CC3S<1:0>	
Name (Capture)	IC3F<3:0>			IC3PSC<1:0>		CC3S<1:0>		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

输出比较模式

- Bit 7 OC3CE: 输出比较 3 清零使能
0: OC3REF 不会受到 ETRF 输入影响
1: 当 ETRF 输入上检测到高电平时, OC3REF 将被清零
- Bit 6~4 OC3M<2:0>: 输出比较 3 模式
该位域定义了输出参考信号 OC3REF 的行为。OC3REF 为高有效, OC3 和 OC3N 的有效电平由 CC3P 和 CC3NP 位决定。
000: 冻结 – 输出比较寄存器 TIMx_CCR3 寄存器和 TIMx_CNT 计数器之间的比较对输出无效 (该模式用来生成时基)。
001: 发生匹配时设置通道 3 为有效电平 – 当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR3 发生匹配后, OC3REF 信号强制为高电平。
010: 发生匹配时设置通道 3 为无效电平。当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR3 发生匹配后, OC3REF 信号强制为低电平。
011: 翻转 - 当 TIMx_CNT=TIMx_CCR3, OC3REF 发生翻转。
100: 强制为无效电平 – OC3REF 强制为低电平。
101: 强制为有效电平 – OC3REF 强制为高电平。
110: PWM 模式 1 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR3, 通道 3 为有效电平, 否则, 通道 3 为无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR3, 通道 3 为无效电平 (OC3REF='0'), 否则通道 3 为有效电平 (OC3REF='1')。
111: PWM 模式 2 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR3, 通道 3 为无效电平, 否则, 通道 3 有无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR3, 通道 3 为有效电平, 否则通道 3 为无效电平。
- 注意:
- 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 3, 且 CC3S=00 (通道为输出模式), 该位域将不能更改。
 - 在 PWM 模式 1 和 2 中, 仅当比较结果更改或当输出比较模式从冻结模式转换成 PWM 模式, OCREF 电平才会更改。
 - 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, OC3M 有效位才会设置为预载值中新的值。
- Bit 3 OC3PE: 输出比较 3 预载使能
0: TIMx_CCR3 的预载寄存器禁止。TIMx_CCR3 在任何时候都可写, 新写入的值将立刻生效。
1: TIMx_CCR3 的预载寄存器禁止。读/写操作可访问预载寄存器。每当发生一次

更新事件, TIMx_CCR3 预载入值将会被填入有效寄存器。

注意:

1: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 3, 且 CC3S=00 (通道为输出模式), 该位域将不能更改。

2: 仅在单脉冲模式下 (TIMx_CR1 寄存器中的 OPM 设置为 1), PWM 模式可在不经过验证预载寄存器的情况下使用。其他情况下的行为不做保证。

Bit 2

OC3FE: 输出比较 3 高速使能

该位用来加速在 CC 输出上的输入触发事件的效应。

0: 当触发开启, CC3 运作正常取决于计数器和 CCR3 的值。当触发输入上发现边沿时, 至少需要 5 个时钟周期来激活 CC3 输出。

1: 触发输入上的有效沿类似于 CC3 输出上的比较匹配。设置 OC 为 1 用来比较电平, 采样触发输入和激活 CC3 输出的延时将会减少至 3 个时钟周期。只有当通道配置为 PWM1 或 PWM2 模式, OCFE 才会起作用。

Bit 1~0

CC3S<1:0>: 捕捉/比较 3 选择

该位域定义了通道和使用的输入的方向。

00: CC3 通道配置为输出

01: CC3 通道配置为输入, IC3 映射到 TI3

10: CC3 通道配置为输入, IC3 映射到 TI4

11: CC3 通道配置为输入, IC3 映射到 TRC

注意: 当通道关闭(TIMx_CCER 寄存器中的 CC3E = '0'), CC3S 为只写。

输入捕捉模式

Bit 7~4

IC3F<3:0>: 输入捕捉 3 滤波器

该位域定义了 TI1 输入的采样频率和数字滤波器的长度。

数字滤波器由一个事件计数器组成, 每 N 个连续事件才视为一个有效边沿:

0000: 无滤波器, 采样频率为 f_{DTS}

0001: $f_{SAMPLING} = f_{CK_INT}$, $N = 2$

0010: $f_{SAMPLING} = f_{CK_INT}$, $N = 4$

0011: $f_{SAMPLING} = f_{CK_INT}$, $N = 8$

0100: $f_{SAMPLING} = f_{DTS} / 2$, $N = 6$

0101: $f_{SAMPLING} = f_{DTS} / 2$, $N = 8$

0110: $f_{SAMPLING} = f_{DTS} / 4$, $N = 6$

0111: $f_{SAMPLING} = f_{DTS} / 4$, $N = 8$

1000: $f_{SAMPLING} = f_{DTS} / 8$, $N = 6$

1001: $f_{SAMPLING} = f_{DTS} / 8$, $N = 8$

1010: $f_{SAMPLING} = f_{DTS} / 16$, $N = 5$

1011: $f_{SAMPLING} = f_{DTS} / 16$, $N = 6$

1100: $f_{SAMPLING} = f_{DTS} / 16$, $N = 8$

1101: $f_{SAMPLING} = f_{DTS} / 32$, $N = 5$

1110: $f_{SAMPLING} = f_{DTS} / 32$, $N = 6$

1111: $f_{SAMPLING} = f_{DTS} / 32$, $N = 8$

注意: 当 ICxF <3:0> = 1, 2 or 3 时, 公式中的 f_{DTS} 由 CK_INT 取代

Bit 3~2

IC3PSC<1:0>: 输入捕捉 3 预分频器

该位域定义了作用在 CC3 输入 (IC3) 上的预分频比。当 CC3E='0' (TIMx_CCER 寄存器), 预分频器将复位。

00: 无预分频器。每当捕捉输入上检测到边沿时, 发生捕捉动作。

01: 每发生 2 次事件, 执行一次捕捉

10: 每发生 4 次事件, 执行一次捕捉

11: 每发生 8 次事件, 执行一次捕捉

Bit 1~0 CC3S<1:0>: 输入捕捉 3 选择

该位域定义了通道和使用的输入的方向。

00: CC3 通道配置为输出

01: CC3 通道配置为输入, IC3 映射到 TI3

10: CC3 通道配置为输入, IC3 映射到 TI4

11: CC3 通道配置为输入, IC3 映射到 TRC

注意: 当通道关闭(TIMx_CCER 寄存器中的 CC3E = '0'), CC3S 为只写。

6.2.6.23 捕捉/比较模式寄存器 2 高 8 位 (TIM1_CCMR2H)

TIMx_CCMR2H: 捕捉/比较模式寄存器 2 (TIM1_CCMR2H)								
Bit	7	6	5	4	3	2	1	0
Name (Compare)	OC4CE	OC4M<2:0>			OC4PE	OC4FE	CC4S<1:0>	
Name (Capture)	IC4F<3:0>				IC4PSC<1:0>		CC4S<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

输出比较模式

Bit 7 OC4CE: 输出比较 4 清零使能

0: OC4REF 不会受到 ETRF 输入影响

1: 当 ETRF 输入上检测到高电平时, OC4REF 将被清零

Bit 6~4 OC4M<2:0>: 输出比较 4 模式

该位域定义了输出参考信号 OC4REF 的行为。OC4REF 为高有效, OC4 和 OC4N 的有效电平由 CC4P 和 CC4NP 位决定。

000: 冻结 - 输出比较寄存器 TIMx_CCR4 寄存器和 TIMx_CNT 计数器之间的比较对输出无效 (该模式用来生成时基)。

001: 发生匹配时设置通道 4 为有效电平 - 当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR4 发生匹配后, OC4REF 信号强制为高电平。

010: 发生匹配时设置通道 4 为无效电平。当计数器 TIMx_CNT 与捕捉/比较寄存器 1 TIMx_CCR4 发生匹配后, OC4REF 信号强制为低电平。

011: 翻转 - 当 TIMx_CNT=TIMx_CCR4, OC4REF 发生翻转。

100: 强制为无效电平 - OC4REF 强制为低电平。

101: 强制为有效电平 - OC4REF 强制为高电平。

110: PWM 模式 1 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR4, 通道 4 为有效电平, 否则, 通道 4 为无效电平。在递减模式下, 当 TIMx_CNT>TIMx_CCR4, 通道 2 为无效电平 (OC4REF='0'), 否则通道 4 为有效电平 (OC4REF='1')。

111: PWM 模式 2 - 在递增模式下, 当 TIMx_CNT<TIMx_CCR4, 通道 4 为无效电

平，否则，通道 4 为有无效电平。在递减模式下，当 $\text{TIMx_CNT} > \text{TIMx_CCR4}$ ，通道 4 为有效电平，否则通道 4 为无效电平。

注意：

- 1: 当 TIMx_BDTR 寄存器中的 **LOCK** 位被设置为锁定级别 3，且 $\text{CC4S}=00$ （通道为输出模式），该位域将不能更改。
- 2: 在 PWM 模式 1 和 2 中，仅当比较结果更改或当输出比较模式从冻结模式转换成 PWM 模式，**OCREF** 电平才会更改。
- 3: 对于有互补输出的通道，该位域设置为预载值。如果 TIMx_CR2 寄存器中的 **CCPC** 位设置为 1，则只有当 **COM** 事件发生时，**OC4M** 有效位才会设置为预载值中新的值。

Bit 3

OC4PE: 输出比较 4 预载使能

- 0: TIMx_CCR4 的预载寄存器禁止。 TIMx_CCR4 在任何时候都可写，新写入的值将立刻生效。
1: TIMx_CCR4 的预载寄存器禁止。读/写操作可访问预载寄存器。每当发生一次更新事件， TIMx_CCR4 预载入值将会被填入有效寄存器。

注意：

- 1: 当 TIMx_BDTR 寄存器中的 **LOCK** 位被设置为锁定级别 3，且 $\text{CC4S}=00$ （通道为输出模式），该位域将不能更改。
- 2: 仅在单脉冲模式下 (TIMx_CR1 寄存器中的 **OPM** 设置为 1)，PWM 模式可在不经过验证预载寄存器的情况下使用。其他情况下的行为不做保证。

Bit 2

OC4FE: 输出比较 4 高速使能

该位用来加速在 CC 输出上的输入触发事件的效应。

- 0: 当触发开启，**CC4** 运作正常取决于计数器和 **CCR4** 的值。当触发输入上发现边沿时，至少需要 5 个时钟周期来激活 **CC4** 输出。
1: 触发输入上的有效沿类似于 **CC4** 输出上的比较匹配。设置 **OC** 为 1 用来比较电平，采样触发输入和激活 **CC4** 输出的延时将会减少至 3 个时钟周期。只有当通道配置为 PWM1 或 PWM2 模式，**OCFE** 才会起作用。

Bit 1~0

CC4S<1:0>: 输出比较 4 选择

该位域定义了通道以及使用的输入的方向（输入/输出）。

- 00: **CC4** 通道配置为输出
01: **CC4** 通道配置为输入，**IC4** 映射到 **TI4**
10: **CC4** 通道配置为输入，**IC4** 映射到 **TI3**
11: **CC4** 通道配置为输入，**IC4** 映射到 **TRC**

注意：当通道为关闭状态时(TIMx_CCER 中 **CC4E** = '0')，**CC4S** 为只写。

输入捕捉模式

Bit 7~4

IC4F<3:0>: 输入捕捉 4 滤波器

该位域定义了 **TI1** 输入的采样频率和数字滤波器的长度。

数字滤波器由一个事件计数器组成，每 N 个连续事件才视为一个有效边沿：

- 0000: 无滤波器，采样频率为 f_{DTS}
0001: $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, $N = 2$
0010: $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, $N = 4$
0011: $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, $N = 8$
0100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2$, $N = 6$
0101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2$, $N = 8$

- 0110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4, N = 6$
 0111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4, N = 8$
 1000: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8, N = 6$
 1001: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8, N = 8$
 1010: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 5$
 1011: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 6$
 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 8$
 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 5$
 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 6$
 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 8$

注意: 当 ICxF <3:0> = 1, 2 or 3 时, 公式中的 f_{DTS} 由 CK_INT 取代

Bit 3~2 IC4PSC<1:0>: 输入捕捉 4 预分频器

该位域定义了作用在 CC4 输入 (IC4) 上的预分频比。当 CC4E='0' (TIMx_CCER 寄存器), 预分频器将复位。

00: 无预分频器。每当捕捉输入上检测到边沿时, 发生捕捉动作。

01: 每发生 2 次事件, 执行一次捕捉

10: 每发生 4 次事件, 执行一次捕捉

11: 每发生 8 次事件, 执行一次捕捉

Bit 1~0 CC4S<1:0>: 输入捕捉 4 选择

该位域定义了通道和使用的输入的方向。

00: CC4 通道配置为输出

01: CC4 通道配置为输入, IC4 映射到 TI4

10: CC4 通道配置为输入, IC4 映射到 TI3

11: CC4 通道配置为输入, IC4 映射到 TRC

注意: 当通道关闭(TIMx_CCER 寄存器中的 CC4E = '0'), CC4S 为只写。

6.2.6.24 捕捉/比较使能寄存器低 8 位 (TIM1_CCERL)

TIMx_CCERL: 捕捉/比较使能寄存器 (TIM1_CCERL)								
Bit	7	6	5	4	3	2	1	0
Name	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 CC2NP: 捕捉/比较 2 互补输出极性

CC2 通道配置为输出:

0: OC2N 高有效

1: OC2N 低有效

CC2 通道配置为输入:

该位需和 CC1P 一起使用来定义 TI1FP1 和 TI2FP1 的极性。

注意: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC2NP 有效位才会设置为预载值中新的值。

注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 2 或 3, 且 CC2S=00 (通道为输出模式), 该位将不可写。

Bit 6	CC2NE: 捕捉/比较 2 互补输出使能 0: 关闭 – OC2N 无效。OC2N 电平取决于 MOE, OSS1, OSSR, OIS2, OIS2N 和 CC2E 的功能 1: 开启 – OC2N 为对应输出引脚上的输出信号, 由 MOE, OSS1, OSSR, OIS2, OIS2N 和 CC2E 决定。 注意: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC2NE 有效位才会设置为预载值中新的值。
Bit 5	CC2P: 捕捉/比较 2 输出极性 CC2 通道配置为输出: 0: OC2 高有效 1: OC2 低有效 CC2 通道配置为输入: CC2NP/CC2P 为触发和捕捉操作选择 TI1FP1 和 TI2FP1 的有效极性。 00: 正向/上升沿 电路对 TIxFP1 上升沿敏感 (在复位, 外部时钟或触发模式下, 进行捕捉或触发), TIxFP1 不反向 (门控模式或编码器模式下, 进行触发)。 01: 反向/下降沿 电路对 TIxFP1 下降沿敏感 (在复位, 外部时钟或触发模式下, 进行捕捉或触发), TIxFP1 反向 (门控模式或编码器模式下, 进行触发)。 10: 预留, 不要使用该配置 11: 正向/上升沿和下降沿 电路对 TIxFP1 上升沿和下降沿均敏感 (在复位, 外部时钟或触发模式下, 进行捕捉或触发), TIxFP1 不反向 (门控模式或编码器模式下, 进行触发) 注意: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC2P 有效位才会设置为预载值中新的值。 注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 2 或 3, 且 CC2S=00 (通道为输出模式), 该位将不可写
Bit 4	CC2E: 捕捉/比较 2 输出使能 CC2 通道配置为输出: 0: 关闭 – OC2 无效。OC2N 电平取决于 MOE, OSS1, OSSR, OIS2, OIS2N 和 CC2E 的功能 1: 开启 – OC2 为对应输出引脚上的输出信号, 由 MOE, OSS1, OSSR, OIS2, OIS2N 和 CC2E 决定 CC2 通道配置为输入: 该位决定了计数值是否能捕捉到输入捕捉/比较寄存器 2(TIMx_CCR2)。 0: 禁止捕捉 1: 使能捕捉 注意: 对于有互补输出的通道, 该位设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC2E 有效位才会设置为预载值中新的值。
Bit 3	CC1NP: 捕捉/比较 1 互补输出极性

CC1 通道配置为输出:

0: OC1N 高有效

1: OC1N 低有效

CC1 通道配置为输入:

该位需和 CC1P 一起使用来定义 TI1FP1 和 TI2FP1 的极性。参考 CC1P 描述。

注意: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC1NP 有效位才会设置为预载值中新的值。

注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 2 或 3, 且 CC1S=00 (通道为输出模式), 该位将不可写。

Bit 2

CC1NE: 捕捉/比较 1 互补输出使能

0: 关闭 - OC1N 无效。OC1N 电平取决于 MOE, OSS1, OSSR, OIS1, OIS1N 和 CC1E 的功能

1: 开启 - OC1N 为对应输出引脚上的输出信号, 由 MOE, OSS1, OSSR, OIS1, OIS1N 和 CC1E 决定。

注意: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC1NE 有效位才会设置为预载值中新的值。

Bit 1

CC1P: 捕捉/比较 1 输出极性

CC1 通道配置为输出:

0: OC1 高有效

1: OC1 低有效

CC1 通道配置为输入:

CC1NP/CC1P 为触发和捕捉操作选择 TI1FP1 和 TI2FP1 的有效极性。

00: 正向/上升沿

电路对 TIxFP1 上升沿敏感 (在复位, 外部时钟或触发模式下, 进行捕捉或触发), TIxFP1 不反向 (门控模式或编码器模式下, 进行触发)。

01: 反向/下降沿

电路对 TIxFP1 下降沿敏感 (在复位, 外部时钟或触发模式下, 进行捕捉或触发), TIxFP1 反向 (门控模式或编码器模式下, 进行触发)。

10: 预留, 不要使用该配置。

11: 正向/上升沿和下降沿

电路对 TIxFP1 上升沿和下降沿均敏感 (在复位, 外部时钟或触发模式下, 进行捕捉或触发), TIxFP1 不反向 (门控模式或编码器模式下, 进行触发)

注意: 对于有互补输出的通道, 该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1, 则只有当 COM 事件发生时, CC1P 有效位才会设置为预载值中新的值。

注意: 当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 2 或 3, 且 CC1S=00 (通道为输出模式), 该位将不可写

Bit 0

CC1E: 捕捉/比较 1 输出使能

CC1 通道配置为输出:

0: 关闭 - OC1 无效。OC1N 电平取决于 MOE, OSS1, OSSR, OIS1, OIS1N 和 CC1E 的功能

1: 开启 - OC1 为对应输出引脚上的输出信号, 由 MOE, OSS1, OSSR, OIS1, OIS1N

和 CC1E 决定

CC1 通道配置为输入：

该位决定了计数值是否能捕捉到输入捕捉/比较寄存器 1(TIMx_CCR1)。

0：禁止捕捉

1：使能捕捉

注意：对于有互补输出的通道，该位设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1，则只有当 COM 事件发生时，CC1E 有效位才会设置为预载值中新的值。

6.2.6.25 捕捉/比较使能寄存器高 8 位 (TIM1_CCERH)

TIMx_CCERH：捕捉/比较使能寄存器 (TIM1_CCERH)								
Bit	7	6	5	4	3	2	1	0
Name	CC4NP	—	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 CC4NP：捕捉/比较 4 互补输出极性

CC4 通道配置为输出：

0：OC4N 高有效

1：OC4N 低有效

CC4 通道配置为输入：

该位需和 CC4P 一起使用来定义 TI1FP1 和 TI2FP1 的极性。

注意：对于有互补输出的通道，该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1，则只有当 COM 事件发生时，CC4NP 有效位才会设置为预载值中新的值。

注意：当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 2 或 3，且 CC4S=00（通道为输出模式），该位将不可写。

Bit 6 预留，必须保持为复位值

Bit 5 CC4P：捕捉/比较 4 输出极性

CC4 通道配置为输出：

0：OC4 高有效

1：OC4 低有效

CC4 通道配置为输入：

CC4NP/CC4P 为触发和捕捉操作选择 TI1FP1 和 TI2FP1 的有效极性。

00：正向/上升沿

电路对 TIxFP1 上升沿敏感（在复位，外部时钟或触发模式下，进行捕捉或触发），TIxFP1 不反向（门控模式或编码器模式下，进行触发）。

01：反向/下降沿

电路对 TIxFP1 下降沿敏感（在复位，外部时钟或触发模式下，进行捕捉或触发），TIxFP1 反向（门控模式或编码器模式下，进行触发）。

10：预留，不要使用该配置。.

11：正向/上升沿和下降沿

电路对 TIxFP1 上升沿和下降沿均敏感（在复位，外部时钟或触发模式下，进行捕捉或触发），TIxFP1 不反向（门控模式或编码器模式下，进行触发）

注意：对于有互补输出的通道，该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1，则只有当 COM 事件发生时，CC4P 有效位才会设置为预载值中新的值。

注意：当 TIMx_BDTR 寄存器中的 LOCK 位被设置为锁定级别 2 或 3，且 CC4S=00（通道为输出模式），该位将不可写

Bit 4 CC4E：捕捉/比较 4 输出使能

CC4 通道配置为输出：

0: 关闭 – OC4 无效。OC2N 电平取决于 MOE, OSS1, OSSR, OIS4, OIS4N 和 CC4E 的功能

1: 开启 – OC4 为对应输出引脚上的输出信号，由 MOE, OSS1, OSSR, OIS4, OIS4N 和 CC4E 决定

CC4 通道配置为输入：

该位决定了计数值是否能捕捉到输入捕捉/比较寄存器 4(TIMx_CCR4)。

0: 禁止捕捉

1: 使能捕捉

注意：对于有互补输出的通道，该位设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1，则只有当 COM 事件发生时，CC4E 有效位才会设置为预载值中新的值。

Bit 3 CC3NP：捕捉/比较 3 互补输出极性

CC3 通道配置为输出：

0: OC3N 高有效

1: OC3N 低有效

CC3 通道配置为输入：

该位需和 CC3P 一起使用来定义 TI1FP1 和 TI2FP1 的极性。

注意：对于有互补输出的通道，该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1，则只有当 COM 事件发生时，CC3NP 有效位才会设置为预载值中新的值。

Bit 2 CC3NE：捕捉/比较 3 互补输出使能

0: 关闭 – OC3N 无效。OC3N 电平取决于 MOE, OSS1, OSSR, OIS3, OIS3N 和 CC3E 的功能

1: 开启 – OC3N 为对应输出引脚上的输出信号，由 MOE, OSS1, OSSR, OIS3, OIS3N 和 CC3E 决定。

注意：对于有互补输出的通道，该位域设置为预载值。如果 TIMx_CR2 寄存器中的 CCPC 位设置为 1，则只有当 COM 事件发生时，CC3NE 有效位才会设置为预载值中新的值。

Bit 1 CC3P：捕捉/比较 3 输出极性

CC3 通道配置为输出：

0: OC3 高有效

1: OC3 低有效

CC3 通道配置为输入：

CC3NP/CC3P 为触发和捕捉操作选择 TI1FP1 和 TI2FP1 的有效极性。

00: 正向/上升沿

电路对 TIxFP1 上升沿敏感（在复位，外部时钟或触发模式下，进行捕捉或触发），

TIxFP1 不反向（门控模式或编码器模式下，进行触发）。

01: 反向/下降沿

电路对 **TIxFP1** 下降沿敏感（在复位，外部时钟或触发模式下，进行捕捉或触发），
TIxFP1 反向（门控模式或编码器模式下，进行触发）。

10: 预留，不要使用该配置。

11: 正向/上升沿和下降沿

电路对 **TIxFP1** 上升沿和下降沿均敏感（在复位，外部时钟或触发模式下，进行捕捉或触发），
TIxFP1 不反向（门控模式或编码器模式下，进行触发）

注意：对于有互补输出的通道，该位域设置为预载值。如果 **TIMx_CR2** 寄存器中的
CCPC 位设置为 1，则只有当 COM 事件发生时，CC3P 有效位才会设置为预载值
中新的值。

注意：当 **TIMx_BDTR** 寄存器中的 LOCK 位被设置为锁定级别 2 或 3，且 CC3S=00
（通道为输出模式），该位将不可写

Bit 0

CC3E: 捕捉/比较 3 输出使能

CC3 通道配置为输出：

0: 关闭 – OC3 无效。 OC3N 电平取决于 MOE, OSSI, OSSR, OIS3, OIS3N 和
CC3E 的功能

1: 开启 – OC3 为对应输出引脚上的输出信号，由 MOE, OSSI, OSSR, OIS3,
OIS3N 和 CC3E 决定

CC3 通道配置为输入：

该位决定了计数值是否能捕捉到输入捕捉/比较寄存器 3(**TIMx_CCR3**)。

0: 禁止捕捉

1: 使能捕捉

注意：对于有互补输出的通道，该位设置为预载值。如果 **TIMx_CR2** 寄存器中的
CCPC 位设置为 1，则只有当 COM 事件发生时，CC3E 有效位才会设置为预载值
中新的值。

MOE bit	OSSI bit	OSSR bit	CCnE bit	CCnNE bit	OCn 输出状态	OCnN 输出状态
1	X	0	0	0	输出禁止（非计时器驱动） OCn=0, OCn_EN=1	输出禁止（非计时器驱动） OCnN=0, OCnN_EN=1
		0	0	1	输出禁止（非计时器驱动） OCn=0, OCn_EN=1	OCnREF + 极性 OCnN=OCnREF xor CCnP, OCn_EN=0
		0	1	0	OCnREF + 极性 OCn=OCnREF xor CCnP, OCn_EN=0	输出禁止（非计时器驱动） OCnN=0, OCnN_EN=1
		0	1	1	OCREF + 极性+ 死区时间 OCn_EN=0	互补 OCREF (非 OCREF) + 极性+ 死区时间 OCnN_EN=0
		1	0	0	输出禁止（非计时器驱动） OCn=CCnP, OCn_EN=1	输出禁止（非计时器驱动） OCnN=CCnP, OCnN_EN=1
		1	0	1	关闭状态 (带无效状态的输出 使能) OCn=CCnP, OCn_EN=0	OCnREF + 极性 OCnN=OCnREF xor CCnP, OCn_N_EN=0
		1	1	0	OCnREF + 极性	关闭状态 (带无效状态的输出

MOE bit	OSSI bit	OSSR bit	CCnE bit	CCnNE bit	OCn 输出状态	OCnN 输出状态
					OCn=OCnREF xor CCnP, OCn_EN=0	使能) OCnN=CCnP, OCnN_EN=0
0	0	X	1	1	OCREF + 极性 + 死区时间 OCn_EN=0	互补 OCREF (非 OCREF) + 极性+ 死区时间 OCnN_EN=0
	0		0	0	输出禁止 (非计时器驱动) OCn=CCnP, OCn_EN=1	输出禁止 (非计时器驱动) OCnN=CCnP, OCnN_EN=1
	0		0	1	输出禁止 (非计时器驱动)	
	0		1	0	异步: 如果存在时钟: OCn=CCnP, OCn_EN=1, OCnN=CCnP, OCnN_EN=1, 在死区时间后 OCn=OISn, OCnN=OISnN, 则假定 OISn 和 OISnN 在有效状态下与 OCN 和 OCnN 不对应。	
	0		1	1	输出禁止 (非计时器驱动) OCn=CCnP, OCn_EN=1	
	1		0	0	输出禁止 (非计时器驱动) OCn=CCnP, OCn_EN=1	输出禁止 (非计时器驱动) OCnN=CCnP, OCnN_EN=1
	1		0	1	关闭状态 (带无效状态的输出使能)	
	1		1	0	异步: 如果存在时钟: OCn=CCnP, OCn_EN=1, OCnN=CCnP, OCnN_EN=1, 在死区时间后 OCn=OISn, OCnN=OISnN, 则假定 OISn 和 OISnN 在有效状态下与 OCN 和 OCnN 不对应。	
	1		1	1	输出禁止 (非计时器驱动) OCn=CCnP, OCn_EN=1	

表 6-4 带有刹车功能的互补 OCn 和 OCnN 的输出控制位

6.2.6.26 计数器寄存器低 8 位 (TIM1_CNTL)

TIMx_CNTL: 计数器 (TIM1_CNTL)								
Bit	7	6	5	4	3	2	1	0
Name	CNT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CNT<7:0>: 计数值

6.2.6.27 计数器寄存器高 8 位 (TIM1_CNTH)

TIMx_CNTH: 计数器 (TIM1_CNTH)								
Bit	7	6	5	4	3	2	1	0
Name	CNT<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CNT<15:8>: 计数值

6.2.6.28 预分频寄存器低 8 位 (TIM1_PSCL)

TIMx_PSCL: 预分频器 (TIM1_PSCL)								
Bit	7	6	5	4	3	2	1	0
Name	PSC<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 PSC<7:0>: 预分频器值

计数器时钟频率 (CK_{CNT}) = $f_{CK_PSC} / (PSC<15:0> + 1)$ 。

每发生一次更新事件(包括当计数器由 TIMx_ERG 寄存器中的 UG 位清零或当配置为复位模式时, 通过触发控制器清零), PSC 包含的值需填入到有效的预分频寄存器内。

6.2.6.29 预分频寄存器高 8 位 (TIM1_PSCH)

TIMx_PSCH: 预分频器 (TIM1_PSCH)								
Bit	7	6	5	4	3	2	1	0
Name	PSC<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 PSC<15:8>: 预分频器值

计数器时钟频率 (CK_{CNT}) = $f_{CK_PSC} / (PSC<15:0> + 1)$ 。

每发生一次更新事件(包括当计数器由 TIMx_ERG 寄存器中的 UG 位清零或当配置为复位模式时, 通过触发控制器清零), PSC 包含的值需填入到有效的预分频寄存器内。

6.2.6.30 自动重载寄存器低 8 位 (TIM1_ARRL)

TIMx_ARRL: 自动重载寄存器 (TIM1_ARRL)								
Bit	7	6	5	4	3	2	1	0
Name	ARR<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 ARR<7:0>: 自动重载值

ARR 中的值将被载入实际的自动重载寄存器中。

当自动重载值为空, 计数器被屏蔽。

6.2.6.31 自动重载寄存器高 8 位 (TIM1_ARRH)

TIMx_ARRH: 自动重载寄存器 (TIM1_ARRH)								
Bit	7	6	5	4	3	2	1	0
Name	ARR<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 ARR<15:8>: 自动重载值

ARR 中的值将被载入实际的自动重载寄存器中。
当自动重载值为空，计数器被屏蔽。

6.2.6.32 重复计数寄存器 (TIM1_RCR)

TIMx_RCR: 重复计数寄存器 (TIM1_RCR)								
Bit	7	6	5	4	3	2	1	0
Name	REP<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 REP<7:0>: 重复计数值

当预载寄存器使能，该位域允许用户设置比较寄存器的更新率（例如：预载到有效寄存器的周期性传输），同样也可以设置更新中断生成率。每次当 REP_CNT 的相关递减计数器递减至 0，会产生更新事件，会从 REP 值重新计数。因为只有当发生重复更新事件 U_RC 时，REP_CNT 才会重新载入 REP 值，所以只有在发生下一次重复更新事件时，写入 TIMx_RCR 寄存器的值才会生效。

即，在 PWM 模式下，(REP+1) 相当于：

- 在边沿对齐模式下，(REP+1) 对应的是 PWM 的周期数
- 在中央对齐模式下，(REP+1) 对应的是 1/2 PWM 的周期数

6.2.6.33 捕捉/比较寄存器 1 低 8 位 (TIM1_CCR1L)

TIMx_CCR1L: 捕捉/比较寄存器 1 (TIM1_CCR1L)								
Bit	7	6	5	4	3	2	1	0
Name	CCR1<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR1<7:0>: 捕捉/比较值 1

如果通道 CCn 配置为输出：

CCRn 中的值将被载入实际的捕捉/比较寄存器中（预载值）。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中，CCRn 中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到当前捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较，并在 OCn 上输出。

如果通道 CCn 配置为输入：

CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.34 捕捉/比较寄存器 1 高 8 位 (TIM1_CCR1H)

TIMx_CCR1H: 捕捉/比较寄存器 1 (TIM1_CCR1H)								
Bit	7	6	5	4	3	2	1	0
Name	CCR1<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR1<15:8>: 捕捉/比较值 1

如果通道 CCn 配置为输出：

CCRn 中的值将被载入实际的捕捉/比较寄存器中（预载值）。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中，CCRn 中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较，并在 OCn 上输出。

如果通道 CCn 配置为输入：

CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.35 捕捉/比较寄存器 2 低 8 位 (TIM1_CCR2L)

TIMx_CCR2L: 捕捉/比较寄存器 2 (TIM1_CCR2L)								
Bit	7	6	5	4	3	2	1	0
Name	CCR2<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR2<7:0>: 捕捉/比较值 2

如果通道 CCn 配置为输出：

CCRn 中的值将被载入实际的捕捉/比较寄存器中（预载值）。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中，CCRn 中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较，并在 OCn 上输出。

如果通道 CCn 配置为输入：

CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.36 捕捉/比较寄存器 2 高 8 位 (TIM1_CCR2H)

TIMx_CCR2H: 捕捉/比较寄存器 2 (TIM1_CCR2H)								
Bit	7	6	5	4	3	2	1	0
Name	CCR2<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR2<15:8>: 捕捉/比较值 2

如果通道 CCn 配置为输出：

CCRn 中的值将被载入实际的捕捉/比较寄存器中（预载值）。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中，CCRn 中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较，并在 OCn 上输出。

如果通道 CCn 配置为输入：

CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.37 捕捉/比较寄存器 3 低 8 位 (TIM1_CCR3L)

TIMx_CCR3L: 捕捉/比较寄存器 3 (TIM1_CCR3L)								
Bit	7	6	5	4	3	2	1	0
Name	CCR3<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR3<7:0>: 捕捉/比较值 3

如果通道 CCn 配置为输出:

CCRn 中的值将被载入实际的捕捉/比较寄存器中 (预载值)。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中, CCRn 中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较, 并在 OCn 上输出。

如果通道 CCn 配置为输入:

CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.38 捕捉/比较寄存器 3 高 8 位 (TIM1_CCR3H)

TIMx_CCR3H: 捕捉/比较寄存器 3 (TIM1_CCR3H)								
Bit	7	6	5	4	3	2	1	0
Name	CCR3<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR3<15:8>: 捕捉/比较值 3

如果通道 CCn 配置为输出:

CCRn 中的值将被载入实际的捕捉/比较寄存器中 (预载值)。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中, CCRn 中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较, 并在 OCn 上输出。

如果通道 CCn 配置为输入:

CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.39 捕捉/比较寄存器 4 低 8 位 (TIM1_CCR4L)

TIMx_CCR4L: 捕捉/比较寄存器 4 (TIM1_CCR4L)								
Bit	7	6	5	4	3	2	1	0
Name	CCR4<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR4<7:0>: 捕捉/比较值 4

如果通道 CCn 配置为输出:

CCRn 中的值将被载入实际的捕捉/比较寄存器中 (预载值)。

如果在 TIMx_CCMRn 寄存器中的预载功能没有选中, CCRn 中的值将被永久载入; 否则, 每当发生更新事件, 预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较, 并在 OCn 上输出。

如果通道 CCn 配置为输入：
CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.40 捕捉/比较寄存器 4 高 8 位 (TIM1_CCR4H)

TIMx_CCR4H: 捕捉/比较寄存器 4 (TIM1_CCR4H)								
Bit	7	6	5	4	3	2	1	0
Name	CCR4<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 CCR4<15:8>: 捕捉/比较值 4

如果通道 CCn 配置为输出：
CCRn 中的值将被载入实际的捕捉/比较寄存器中（预载值）。
如果在 TIMx_CCMRn 寄存器中的预载功能没有选中，CCRn 中的值将被永久载入；
否则，每当发生更新事件，预载值将会复制到有效的捕捉/比较寄存器中。有效捕捉/比较寄存器中包含的值将会与 TIMx_CNT 中的值进行比较，并在 OCn 上输出。
如果通道 CCn 配置为输入：
CCRn 为由上一个输入捕捉事件(ICn)传输的计数值。

6.2.6.41 刹车和死区时间寄存器低 8 位 (TIM1_BDTRL)

TIMx_BDTRL: 刹车和死区时间寄存器 (TIM1_BDTRL)								
Bit	7	6	5	4	3	2	1	0
Name	DTG<7:0>							
R/W	R/W							
POR	0	0	0	0	0	0	0	0

Bit 7~0 DTG<7:0>: 死区时间生成设置

该位域定义了互补输出之间插入的死区时间。DT 对应的就是该时间段。
当 DTG<7:5>=0xx 时, DT=DTG<7:0> x tdtg , 其中 tdtg=t_{DTS}
当 DTG<7:5>=10x 时, DT=(64+DTG<5:0>) x tdtg, 其中 Tdtg=2xt_{DTS}
当 DTG<7:5>=110 时, DT=(32+DTG<4:0>) x tdtg , 其中 Tdtg=8xt_{DTS}
当 DTG<7:5>=111 时, DT=(32+DTG<4:0>) x tdtg , 其中 Tdtg=16xt_{DTS}
举例: 如果 TDTS=125 ns (8 MHz), 可能的死区时间为:
当 DTG<7:0>=00H ~ 7FH 时, 死区时间为 0 到 15875 ns (步长为 125ns),
当 DTG<7:0>=80H ~ BFH 时, 死区时间为 16 us 到 31750 ns (步长为 250ns),
当 DTG<7:0>=C0H ~ DFH 时, 死区时间为 32 us 到 63 us (步长为 1us),
当 DTG<7:0>=E0H ~ FFH 时, 死区时间为 64 us 到 126 us (步长为 2us)

注意: 当 TIMx_BDTR 寄存器中的 LOCK 位已被设置为锁定级别 1, 2 或 3, 则该位不可更改。

6.2.6.42 刹车和死区时间寄存器高 8 位 (TIM1_BDTRH)

TIMx_BDTRH: 刹车和死区时间寄存器 (TIM1_BDTRH)								
Bit	7	6	5	4	3	2	1	0
Name	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0
Bit 7	MOE: 主输出使能							
	0: OC 和 OCN 输出禁止或强制为空闲状态。							
	1: 如果 OC 和 OCN 各自的使能位都置 1(TIMx_CCER 寄存器中的 CCnE, CCnNE), 则 OC 和 OCN 输出使能。							
	一旦刹车输入有效, 该位会由硬件异步清零。该位可由软件置 1 或自动置 1, 取决于 AOE 位。该位仅作用于配置为输出的通道。							
Bit 6	AOE: 自动输出使能							
	0: MOE 仅可由软件置位							
	1: 在下一个更新事件发生时 (如果刹车输入无效), MOE 可由软件或自动置位。							
	注意: 当 TIMx_BDTR 寄存器中的 LOCK 位已被设置为锁定级别 1, 则该位不可更改。							
Bit 5	BKP: 刹车极性							
	0: 刹车输入 BRK 为低有效							
	1: 刹车输入 BRK 为高有效							
	注意: 当 TIMx_BDTR 寄存器中的 LOCK 位已被设置为锁定级别 1, 则该位不可更改; 任何对该位的写操作都要延时 1 个时钟周期后才变为有效。							
Bit 4	BKE: 刹车使能							
	0: 刹车输入 (BRK 和 CCS 时钟失效事件) 禁止							
	1: 刹车输入 (BRK 和 CCS 时钟失效事件) 使能							
	注意: 当 TIMx_BDTR 寄存器中的 LOCK 位已被设置为锁定级别 1, 则该位不可更改; 任何对该位的写操作都要延时 1 个时钟周期后才变为有效。							
Bit 3	OSSR: 运行模式下的关闭状态选择							
	该位使用于, 当 MOE=1 时, 被配置为输出并使用互补输出的通道。如果计时器中没有使用互补输出, 则 OSSR 不使用。							
	0: 无效状态时, OC/OCN 输出禁止 (OC/OCN 使能输出信号=0)							
	1: 无效状态时, 当 CCnE=1 或 CCnNE=1 时, 便使能 OC/OCN 输出并将其设为无效电平。(OC/OCN 使能输出信号=1)							
	注意: 当 TIMx_BDTR 寄存器中的 LOCK 位已被设置为锁定级别 2, 则该位不可更改。							
Bit 2	OSSI: 空闲模式下的关闭状态选择							
	该位使用于, 当 MOE=0 时, 被配置为输出通道							
	0: 无效状态时, OC/OCN 输出禁止 (OC/OCN 使能输出信号=0)							
	1: 无效状态时, 当 CCnE=1 或 CCnNE=1, 便将 OC/OCN 输出首先强制为其空闲电平。(OC/OCN 使能输出信号=1)							
	注意: 当 TIMx_BDTR 寄存器中的 LOCK 位已被设置为锁定级别 2, 则该位不可更改。							
Bit 1~0	LOCK<1:0>: 锁定配置							

针对软件错误，该位域提供写保护。

00: 锁定关闭 – 不提供写保护

01: 锁定级别 1 = TIMx_BDTR 寄存器中的 DTG, TIMx_CR2 寄存器中的 OISn 和 OISnN, 和 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 不再可写

10: 锁定级别 2 = 锁定级别 1 + CC 极性位 (TIMx_CCER 寄存器中的 CCnP/CCnNP, 只要相关通道由 CCnS 配置为输出) 以及 OSSR 和 OSSI 都不再可写

11: 锁定级别 3 = 锁定级别 2 + CC 控制位 (TIMx_CCMRx 寄存器中的 OCnM 和 OCnPE, 只要相关通道由 CCnS 配置为输出) 都不再可写

注意：锁定配置为仅在复位后可写。一旦 TIMx_BDTR 已写，其设置内容在下一个复位前都处于冻结状态。

CONFIDENTIAL

6.3 16 位通用定时/计数器 (T21)

6.3.1 概述

16 位多功能定时器共有 4 种工作模式，定时器模式、多精度 PWM 模式、捕捉器模式、比较器模式。

◇ T21 支持 4 种工作模式

- 定时器模式（时钟源为 Fosc）
- 多精度 PWM 模式，支持 3 路独立的 PWM 输出，输出占空比、极性都可独立设置
- 捕捉器模式，支持 3 路独立的捕捉输入，每路输入捕捉沿可分别设置
- 比较器模式，支持 3 路独立的比较输出

◇ T2n 支持以下功能组件

- 4 位预分频器和 7 位的后分频器（无实际物理地址，不可读写）
- 16 位计数器 T2n（计数器初始值可写）
- 16 位捕捉/比较寄存器 T2nR0、T2nR1 和 T2nR2

◇ 中断和暂停

- 支持溢出中断 T2nVIF、周期匹配中断 T2nPIF 和多功能中断 T2nMIF
- 在 IDLE 模式下，停止工作

注 1：本节中 T2n 代表 T21；

注 2：精度寄存器和周期寄存器的更新，在写低 8 位寄存器时，同时加载高位寄存器的值，因此寄存器更新时，需要先写高 8 位寄存器，再写低 8 位寄存器；

注 3：管脚图中的 T2n_CH0/T2n_CH1/T2n_CH2 分别代表的是本节中多精度 PWM 模式的输出通道 PWM2n0/PWM2n1/PWM2n2、捕捉输入模式的通道 T2nCI0/T2nCI1/T2nCI2、比较输出模式的输出通道 T2nCO0/T2nCO1/T2nCO2。

6.3.2 内部结构图

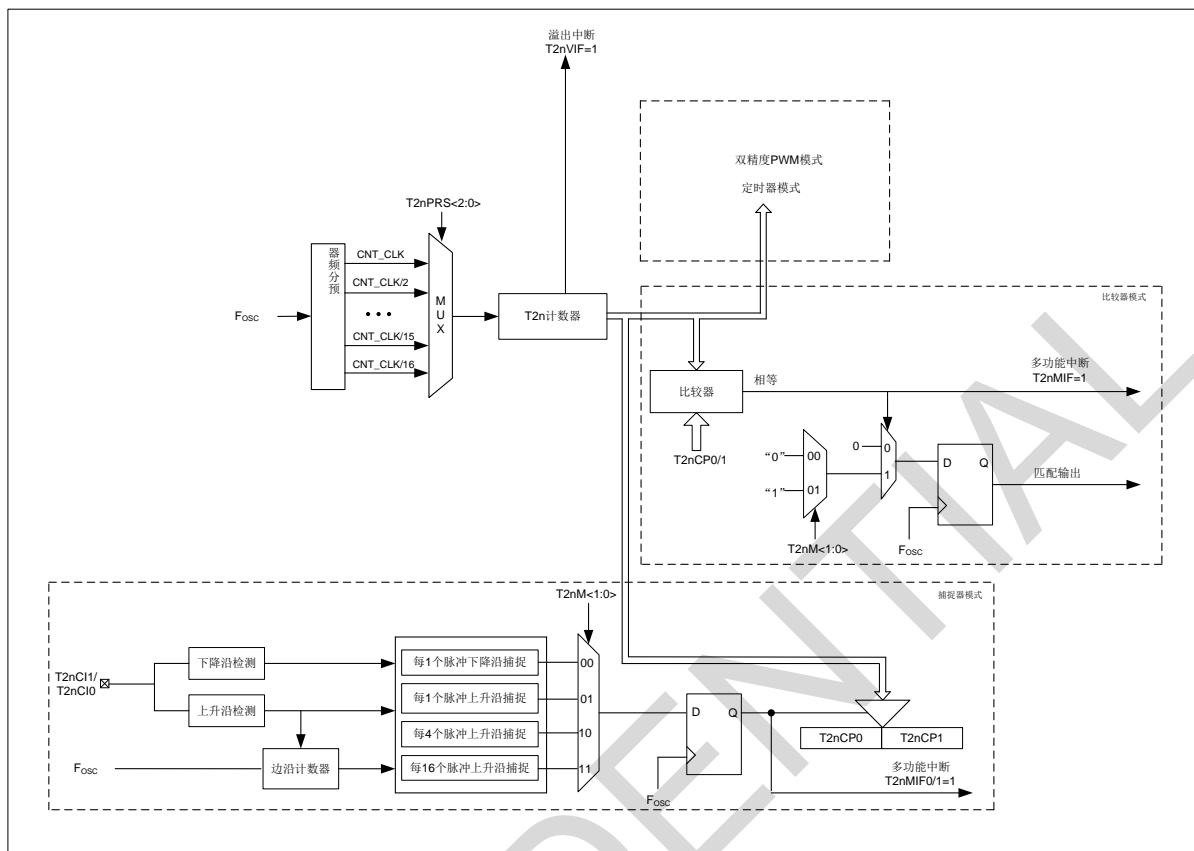


图 6-51 T2n 内部结构图

6.3.3 预分频器和后分频器

预分频器可以提供一个更长的溢出周期。T2n 支持可配置的预分频器。通过 T2nCM 寄存器中的 T2nPRS 位配置预分频器的分频比，预分频比的范围为 1:1 ~ 1:16。任何对计数器或控制寄存器的写操作都会使预分频器和后分频器清零，但不改变配置的分频比。预分频器和后分频器的计数值不可读写。

后分频器的分频比可通过 T2nCH 寄存器中的 T2nPOS<6:0>位进行设置，后分频比范围为 1:1~1:128，通过计数器与周期寄存器值的匹配，来进行后分频。

6.3.4 工作模式

T2n 有 4 种工作模式，定时器模式、多精度 PWM 模式、捕捉器模式、比较器模式，通过 T2nM<3:0>进行模式选择。

T2nEN 置 1 使能之前，需先由 T2nM 设定工作模式，选择计数时钟，并配置好预分频器、后分频器、周期等参数，且需保证计数时钟在 T2nEN 使能时已稳定。

6.3.5 定时器模式

当 T2nM 为 0000 时，Tn 工作在定时器模式。

T2n 计数器的时钟源为系统时钟 Fosc，并支持预分频器和后分频器。

T2n 计数器为可读写寄存器，支持计数初值的设置和计数过程中的计数值改变。

T2nEN 使能时，16 位定时器 T2n 对计数时钟进行递增计数，当 T2n 的计数值与周期寄存器 T2nP 相等时，后分频计数器加 1，同时 T2n 计数器被自动清零并重新开始计数。

当后分频器的计数值与后分频器分频比相同时，复位后分频器，并将定时溢出中断标志 T2nVIF 置“1”，该中断标志需要软件清零。

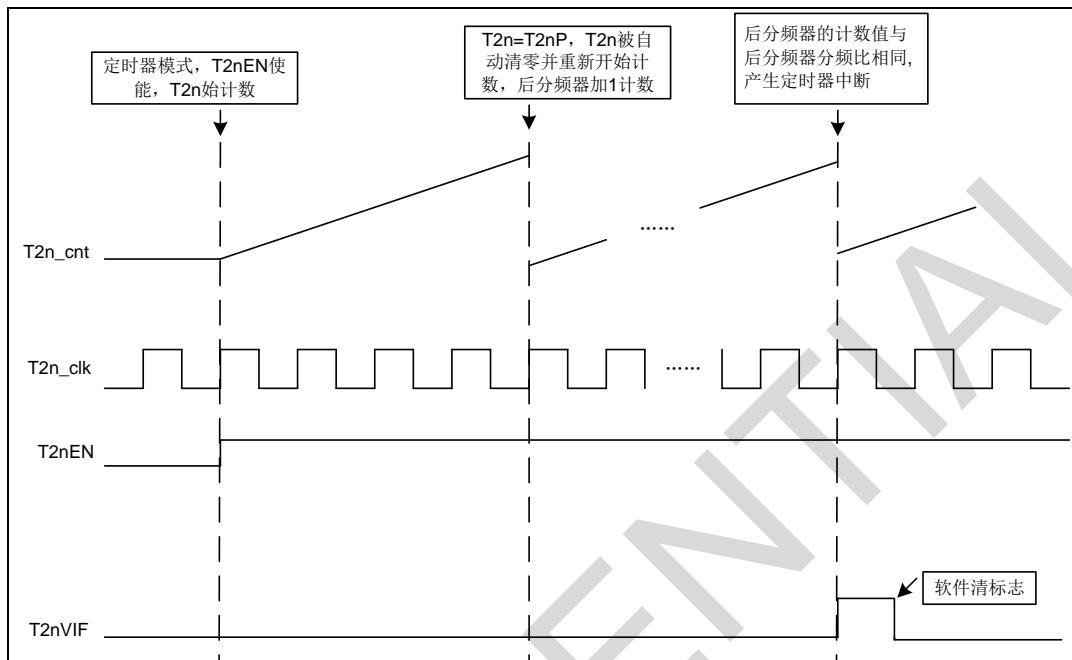


图 6-52 T2n 定时器模式时序图

6.3.6 多精度 PWM 模式

T2nM 为“1100”时，T2n 工作在多精度 PWM 模式。每路支持 3 个 PWM 输出端口 PWM2n0、PWM2n1 和 PWM2n2，可分别对应于 T2nR0、T2nR1 和 T2nR2 独立设置 PWM 占空比，并可独立设置输出极性。

多精度 PWM 模式计数时钟源为系统时钟 Fosc，并支持预分频器和后分频器。此模式下，后分频比不影响 PWM 周期，只影响计数溢出中断标志 T2nVIF。

如下图示，当 T2nEN 使能，T2nTR 为 0 时，PWM 输出关闭，并保持 PWM2n0/1 输出为 0；设置 T2nTR 为 1 时，PWM 输出波形启动，PWM2n0/ PWM2n1/ PWM2n2 输出起始为 1，同时分别将 16 位周期寄存器 T2nP 和 16 位精度寄存器 T2nR0/T2nR1/T2nR2 寄存器的内容，更新至 16 位 PWM 周期缓冲器 PRDBUF 和 16 位精度缓冲器 RESBUF0/RESBUF1/RESBUF2（该缓冲器软件不可读写），随后 16 位计数器 T2n 从零开始递增计数，当 T2n 与 RESBUF0/RESBUF1/RESBUF2 的值相等时，PWM0/ PWM1/ PWM 2 输出改变为 0，并继续递增计数。当 T2n 的计数值与 PRDBUF 相等时，后分频计数器加 1，PWM0/PWM1/PWM2 输出恢复为 1，同时 PRDBUF 和 RESBUF0/RESBUF1/RESBUF2 再次分别载入 T2nP 和 T2nR0/T2nR1/T2nR2 寄存器的值，并产生周期中断标志 T2nPIF，该中断标志需要软件清零。至此一个完整的 PWM 周期完成，随后计数器 T2n 从零开始递增计数，继续循环产生新的 PWM 周期。当后分频器的计数值与后分频器分频比相同时，复位后分频器，并将计数溢出中断标志 T2nVIF 置“1”，该中断标志需要软件清零。

特别的，若 RESBUF 的值不小于 PRDBUF，则当前 PWM 周期内 PWM 输出始终为 1。

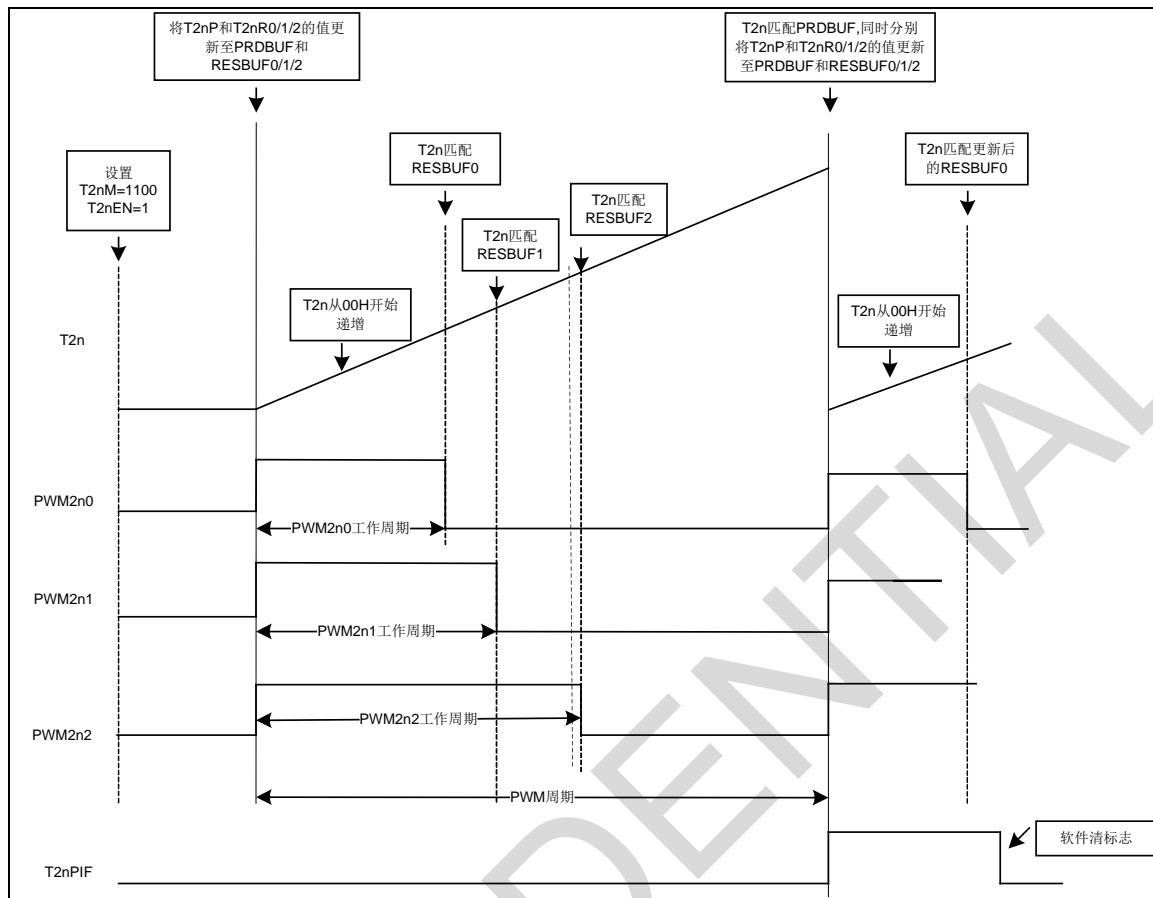


图 6-53 T2n 多精度 PWM 模式示意图

PWM 计算公式如下：

$$\text{PWM 周期} = (T2nP + 1) \times Tosc \times \text{<预分频器分频比>}$$

$$\text{PWM 频率} = 1 / \text{<PWM 周期>}$$

$$\text{PWM 脉宽} = (T2nR0/1 + 1) \times Tosc \times \text{<预分频器分频比>}$$

$$\text{PWM 占空比} = \text{<PWM 脉宽>} / \text{<PWM 周期>}$$

给定 PWM 频率，PWM 的最大分辨率可计算为：

$$\text{分辨率} = \frac{\log(\frac{Fosc}{F_{pwm} * F_{ckps}})}{\log 2}$$

F_{ckps} 是计数器的预分频器的分频比。

注：当 $T2nR0/1/2=0$ 时， $\text{PWM 脉宽} = Tosc \times \text{<预分频器分频比>}.$

6.3.7 捕捉器模式

T2nM 设置为“01xx”时，T2n 工作在捕捉器模式，T2n 支持 3 个相互独立的捕捉输入端口 T2nCI0、T2nCI1 和 T2nCI2。该模式下支持 3 种捕捉条件选择。

T2n 配置为捕捉模式时，时钟源为系统时钟 (Fosc)。在此模式下，16 位计数器 T2n 进行递增计数，当 T2nCI0/T2nCI1/T2nCI2 输入信号的变化状态满足捕捉条件时，计数器 T2n 的值将被载入到相应的 16 位捕捉寄存器 T2nR0/T2nR1/T2nR2 中，并产生多功能中

断 T2nMIF0/T2nMIF1/T2nMIF2，该中断必须由软件清零。计数器继续递增计数。若下一次捕捉事件发生时，捕捉寄存器 T2nR0/T2nR1/T2nR2 中的值未被及时读取，将被新捕捉的值覆盖。当计数值溢出时（即从 $FFFFH$ 变为 $0000H$ ），产生溢出中断 T2nVIF，该中断必须软件清零。

T2n 支持 1 个用于捕捉条件判断的边沿计数器。此边沿计数器仅在捕捉模式有效。当 T2n 关闭或切换为其它模式时，该边沿计数器被清零。但在 T2n 的 4 种捕捉模式相互切换时，该边沿计数器不会被清零。因此，当切换捕捉模式后，首次捕捉可能存在误差，同时也可能导致错误的中断产生。为了避免产生错误中断，用户在改变模式时应该禁止 T2n 相应中断使能位，并且清除中断标志。

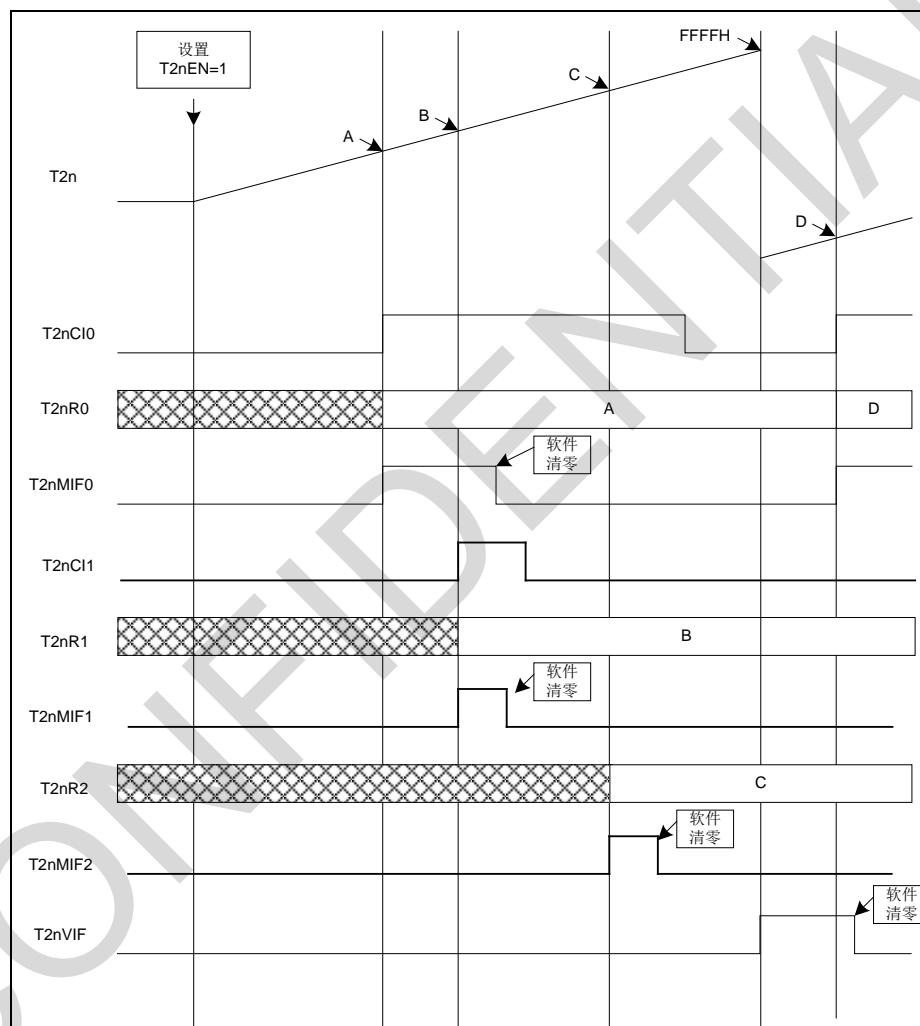


图 6-54 T2n 捕捉器模式时序图 (每个脉冲上升沿捕捉信号)

6.3.8 比较器模式

T2nM 设置为 “10xx” 时，T2n 工作在比较器模式。

比较器模式时钟源为系统时钟 (F_{osc})。在此模式下，16 位计数器 T2n 进行递增计数。当计数器 T2n 的计数值与比较寄存器 T2nR0/T2nR1/T2nR2 中的比较值相等时，执行相应的比较匹配事件，并产生多功能中断 T2nMIF0/T2nMIF1/T2nMIF2，该中断必须软件清零。当计数值溢出时（即从 $FFFFH$ 变为 $0000H$ ），产生溢出中断 T2nVIF，该中断必须

软件清零；计数溢出后，继续从 0 开始递增计数。

比较器模式配置位 T2nM 配置为 1000 或 1001 时，当 T2n 计数器匹配 T2nR0L/T2nR0H 时，PWM2n0 端口输出 0 或 1 并保持；当 T2n 计数器匹配 T2nR1L/T2nR1H 时，PWM2n1 端口输出 0 或 1 并保持；当 T2n 计数器匹配 T2nR2L/T2nR2H 时，PWM2n2 端口输出 0 或 1 并保持。

T2nCL 寄存器的 T2nM 设置为 1011 对应的比较器模式，匹配 T2nR0L/T2nR0H、T2nR1L/T2nR1H 或 T2nR2L/T2nR2H 时，T2n 被清零，并可触发 ADC 转换。ADC 必须先使能，且设置为硬件采样，即 ADCCL 寄存器的 ADEN 和 SMPS 控制位都需设置为 1。特别注意：因匹配时 T2n 被清零，因此只有 T2nR0L/T2nR0H、T2nR1L/T2nR1H 或 T2nR2L/T2nR2H 中的较小值在该模式下有效。

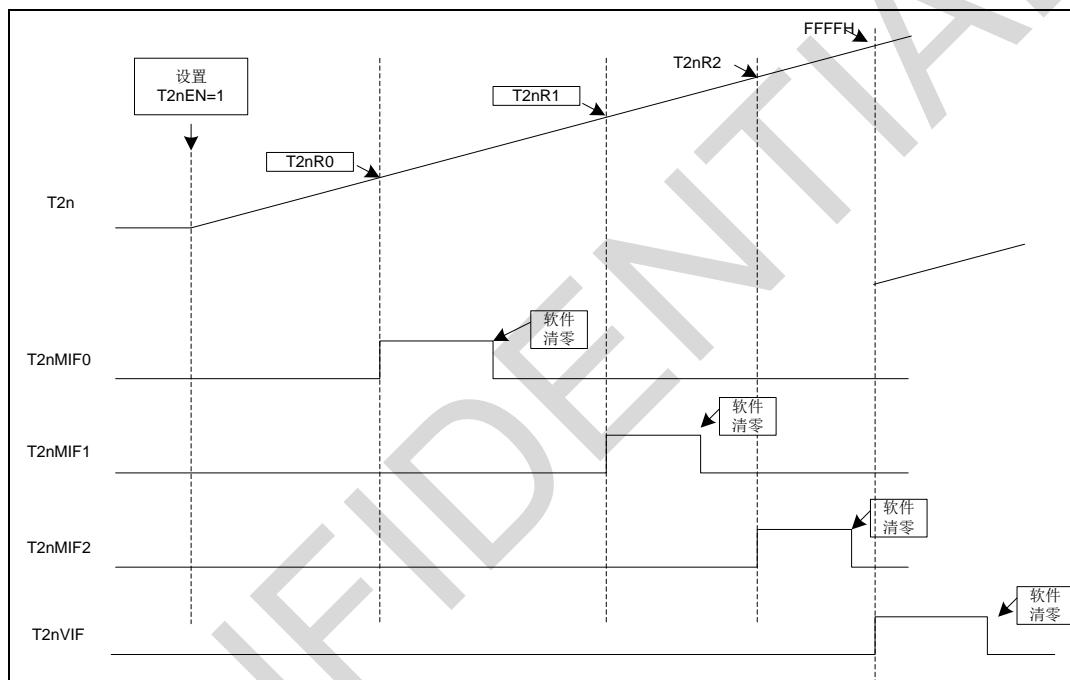


图 6-55 T2n 比较器模式时序图

6.3.9 特殊功能寄存器

6.3.9.1 计数器寄存器低 8 位 (T21L)

T2nL: T2n 计数器低 8 位 (T21L)								
Bit	7	6	5	4	3	2	1	0
Name	T2n<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2n<7:0>: T2n 计数器低 8 位

6.3.9.2 计数器寄存器高 8 位 (T21H)

T2nH: T2n 计数器高 8 位 (T21H)								
Bit	7	6	5	4	3	2	1	0
Name	T2n<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2n<15:8>: T2n 计数器高 8 位

6.3.9.3 周期寄存器低 8 位 (T21PL)

T2nPL: T2n 周期寄存器低 8 位 (T21PL)								
Bit	7	6	5	4	3	2	1	0
Name	T2nP<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 T2nP<7:0>:

多精度 PWM 模式: PWM 周期值低 8 位

6.3.9.4 周期寄存器高 8 位 (T21PH)

T2nPH: T2n 周期寄存器高 8 位 (T21PH)								
Bit	7	6	5	4	3	2	1	0
Name	T2nP<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 T2nP<15:8>:

多精度 PWM 模式: PWM 周期值高 8 位

6.3.9.5 多功能寄存器 0 低 8 位 (T21R0L)

T2nR0L: T2n 精度寄存器 0 低 8 位 (T21R0L)								
Bit	7	6	5	4	3	2	1	0
Name	T2nR0<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2nR0<7:0>:

多精度 PWM 模式: PWM2n0 精度值低 8 位

捕捉模式: T2nCl0 捕捉值低 8 位

比较模式: T2nCo0 比较值低 8 位

6.3.9.6 多功能寄存器 0 高 8 位 (T21R0H)

T2nR0H: T2n 精度寄存器 0 高 8 位 (T21R0H)								
Bit	7	6	5	4	3	2	1	0
Name	T2nR0<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2nR0<15:8>:

- 多精度 PWM 模式: PWM2n0 精度值高 8 位
捕捉模式: T2nCl0 捕捉值高 8 位
比较模式: T2nCo0 比较值高 8 位

6.3.9.7 多功能寄存器 1 低 8 位 (T21R1L)

T2nR1L: T2n 精度寄存器 1 低 8 位 (T21R1L)								
Bit	7	6	5	4	3	2	1	0
Name	T2nR1<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2nR1<7:0>:

- 多精度 PWM 模式: PWM2n1 精度值低 8 位
捕捉模式: T2nCl1 捕捉值低 8 位
比较模式: T2nCo1 比较值低 8 位

6.3.9.8 多功能寄存器 1 高 8 位 (T21R1H)

T2nR1H: T2n 精度寄存器 1 高 8 位 (T21R1H)								
Bit	7	6	5	4	3	2	1	0
Name	T2nR1<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2nR1<15:8>:

- 多精度 PWM 模式: PWM2n1 精度值高 8 位
捕捉模式: T2nCl1 捕捉值高 8 位
比较模式: T2nCo1 比较值高 8 位

6.3.9.9 多功能寄存器 2 低 8 位 (T21R2L)

T2nR2L: T2n 精度寄存器 2 低 8 位 (T21R2L)								
Bit	7	6	5	4	3	2	1	0
Name	T2nR2<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2nR2<7:0>:

- 多精度 PWM 模式: PWM2n2 精度值低 8 位

捕捉模式: T2nCl2 捕捉值低 8 位
比较模式: T2nCo2 比较值低 8 位

6.3.9.10 多功能寄存器 2 高 8 位 (T21R2H)

T2nR2H: T2n 精度寄存器 2 高 8 位 (T21R2H)								
Bit	7	6	5	4	3	2	1	0
Name	T2nR2<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T2nR2<15:8>:

双精度 PWM 模式: PWM2n2 精度值高 8 位

捕捉模式: T2nCl2 捕捉值高 8 位

比较模式: T2nCo2 比较值高 8 位

6.3.9.11 控制寄存器低 8 位 (T21CL)

T2nCL: T2n 控制寄存器低 8 位 (T21CL)								
Bit	7	6	5	4	3	2	1	0
Name	T2nM<3:0>				CAP1S<1:0>		CAP0S<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~4 T2nM <3:0>: T2n 工作模式选择位

0000: 定时器模式 (计数器时钟源为 Fosc)

0000~0011: 保留未用

01xx: 捕捉模式

1000: 比较器模式, 匹配时输出 1

1001: 比较器模式, 匹配时输出 0

1010: 比较器模式, 匹配时不改变输出, 由 IO 输出状况决定

1011: 比较器模式, 匹配时复位 T2n, 并触发 ADC 转换, 且不改变输出, 由 IO 输出状况决定

1100: 多精度 PWM 模式

其它: 保留

Bit 3~2 CAP1S<1:0>: CAPTURE1 捕捉输入沿选择位

00: 禁止

01: 捕捉每 1 个脉冲下降沿

10: 捕捉每 1 个脉冲上升沿

11: 捕捉每 1 个脉冲双沿

Bit 1~0 CAP0S<1:0>: CAPTURE0 捕捉输入沿选择位

00: 禁止

01: 捕捉每 1 个脉冲下降沿

10: 捕捉每 1 个脉冲上升沿

11: 捕捉每 1 个脉冲双沿

6.3.9.12 控制寄存器次高 8 位 (T21CM)

T2nCM: T2n 控制寄存器中 8 位 (T21CM)								
Bit	7	6	5	4	3	2	1	0
Name	CAP2S<1:0>		T2nOM20	T2nOM21	T2nPRS<3:0>			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 CAP2S<1:0>: CAPTURE2 捕捉输入沿选择位

- 00: 禁止
- 01: 捕捉每 1 个脉冲下降沿
- 10: 捕捉每 1 个脉冲上升沿
- 11: 捕捉每 1 个脉冲双沿

Bit 5 T2nOM20: PWM2n0 输出极性选择位

- 0: PWM2n0 高有效
- 1: PWM2n0 低有效

Bit 4 T2nOM21: PWM2n1 输出极性选择位

- 0: PWM2n1 高有效
- 1: PWM2n1 低有效

Bit 3~0 T2nPRS<3:0>: T2n 预分频器分频比选择位

- 0000: 分频比为 1:1
- 0001: 分频比为 1:2
- 0010: 分频比为 1:3
- 0011: 分频比为 1:4
- 0100: 分频比为 1:5
- 0101: 分频比为 1:6
- 0110: 分频比为 1:7
- 0111: 分频比为 1:8
- 1000: 分频比为 1:9
- 1001: 分频比为 1:10
- 1010: 分频比为 1:11
- 1011: 分频比为 1:12
- 1100: 分频比为 1:13
- 1101: 分频比为 1:14
- 1110: 分频比为 1:15
- 1111: 分频比为 1:16

6.3.9.13 控制寄存器高 8 位 (T21CH)

T2nCH: T2n 控制寄存器高 8 位 (T21CH)								
Bit	7	6	5	4	3	2	1	0
Name	T2nEN	T2nPOS<6:0>						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 T2nEN : T2n 使能位

0: 关闭
1: 使能
Bit 6~0 T2nPOS<6:0>: T2n 后分频器分频值
后分频次数 = T2nPOS<6:0> + 1

6.3.9.14 输入输出控制寄存器 (T21OC)

T2nOC: T2n 输入输出控制寄存器 (T21OC)								
Bit	7	6	5	4	3	2	1	0
Name	T2nTR	—	PT2EN<1:0>		T2nOM2n2	PT1EN	PT0EN<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 T2nTR: PWM 输出触发位
0: PWM 输出为 0
1: 启动 PWM 输出
Bit 6 保留未用
Bit 5~4 PT2EN<1:0>: CAP2/PWM2/COMP2 端口输入/出使能位
00: 关闭
01: PA6 输入/输出使能
10: PB7 输入/输出使能
11: 保留未用
Bit 3 T2nOM2n2: PWM2n2 输出极性选择位
0: PWM2n2 高有效
1: PWM2n2 低有效
Bit 2 PT1EN: CAP1/PWM1/COMP1 端口输入/出使能位
0: 关闭
1: PA7 输入/输出使能
Bit 1~0 PT0EN<1:0>: CAP0/PWM0/COMP0 端口输入/出使能位
00: 关闭
01: PA0 输入/输出使能
10: PB4 输入/输出使能
11: 保留未用

6.4 通用异步接收/发送器 (UART)

6.4.1 概述

本芯片支持一组全双工的通用异步接收发送器 UART，它采用串行收发方式与外部设备进行数据传输，可以和其它具有异步接收发送器的外部设备进行通讯。

- ◇ 支持两种工作模式
 - 异步接收器
 - 异步发送器
- ◇ 传输波特率配置
 - 高速模式
 - 低速模式
- 支持 8/9 位传输数据格式，约定数据从最低位开始接收/发送
- ◇ 支持全双工模式
- ◇ UART 支持以下功能组件
 - 接收数据寄存器 (RXB)
 - 接收控制寄存器 (RXC)
 - 发送数据寄存器 (TXB)
 - 发送控制寄存器 (TXC)
 - 发送移位寄存器 (TXR) (无实际物理地址，不可读写)
 - 波特率寄存器 (BRR)
- ◇ 中断和暂停
 - 支持接收中断标志 (RXIF, 只可读)
 - 支持发送中断标志 (TXIF, 只可读)
 - 支持中断处理
 - 在 IDLE 模式下，暂停接收和发送
- ◇ 兼容 RS-232/RS-442/RS-485 的通讯接口

注：本节中 TX/RX 端口代表的是芯片管脚图中的 UART_TX/UART_RX 端口。

6.4.2 内部结构图

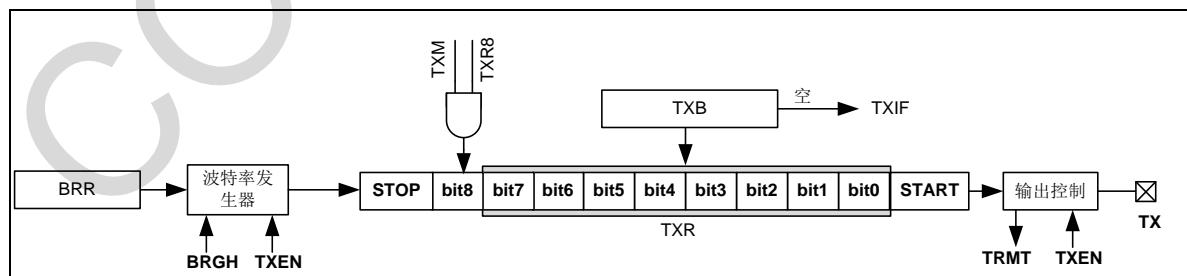


图 6-56 UART 发送端原理图

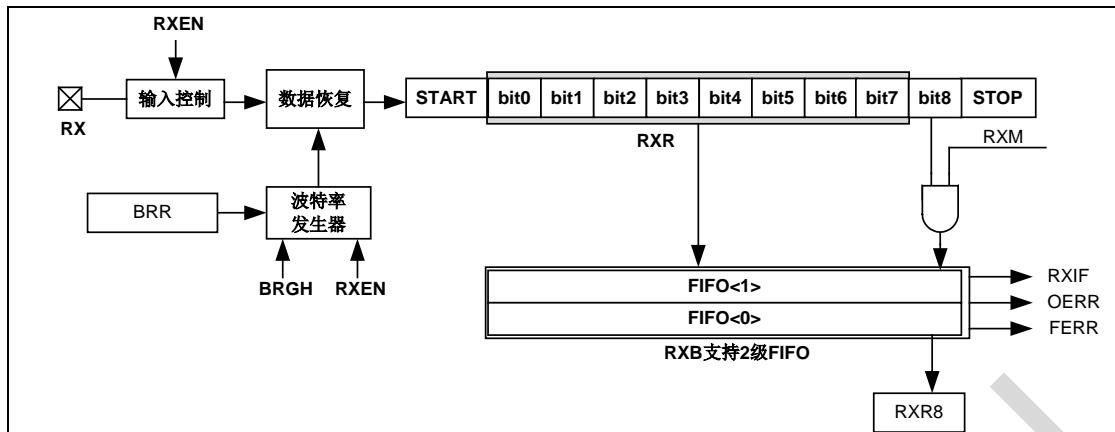


图 6-57 UART 接收端原理图

6.4.3 波特率配置

UART 自身具有一个波特率发生器，通过它可以设定数据传输速率。波特率是由一个独立的内部 8 位计数器产生，它由 BRR 寄存器和 TXC 寄存器的 BRGH 来控制。BRGH 是决定波特率发生器处于高速模式还是低速模式，从而决定计算公式的选用。

BRGH	波特率	计算公式
0	低速模式	$Fosc/(64x(BRR<7:0>+1))$
1	高速模式	$Fosc/(16x(BRR<7:0>+1))$

表 6-5 UART 波特率配置表

6.4.4 传输数据格式

UART 的传输数据格式有两种选择，8 位或 9 位。接收 9 位数据时，读取 RXC 寄存器中的 RXR8 位可以得到接收的第 9 位数据。发送 9 位数据时，可通过 TXC 寄存器中的 TXR8 位设置将要发送的第 9 位数据。

RXM	TXM	传输数据格式
0	0	8 位
1	1	9 位

表 6-6 UART 数据格式配置表

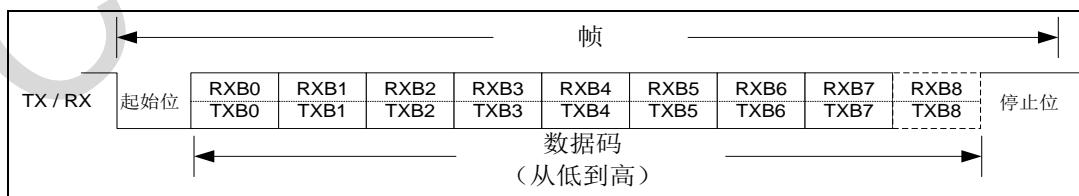


图 6-58 UART 数据格式示意图

6.4.5 异步发送器

异步发送器发送数据时，起始位（START）和结束位（STOP）由芯片内部产生，用户只需要使能异步发送器，并将所要发送的数据写入 TXB 和 TXR8 内，就能实现异步发送，

异步发送器还可以实现数据连续发送。注意，在发送数据时，用户使能异步发送器后，需先查询标志位 TRMT，检查发送移位寄存器（TXR）是否为空，只有在发送移位寄存器为空时，才能发送数据。由于 UART 发送器发送端口 TX 与 I/O 端口复用，在使用 UART 发送端口前，需首先设置所复用的 I/O 端口处于输出状态。

操作流程图如下：

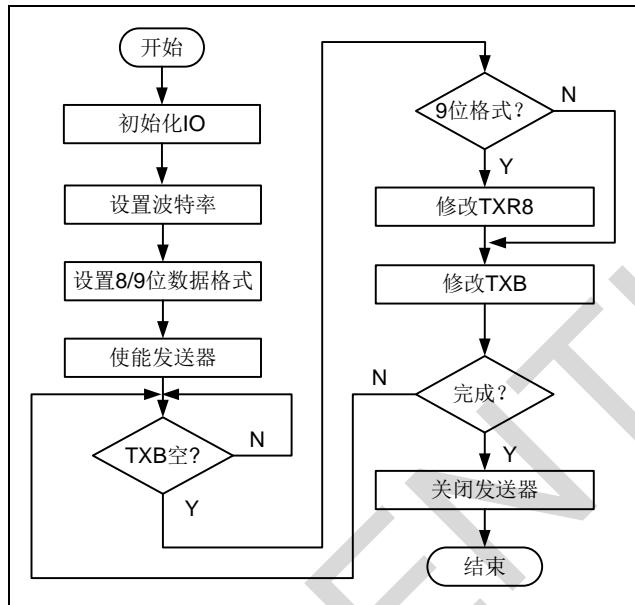


图 6-59 UART 异步发送器操作流程图

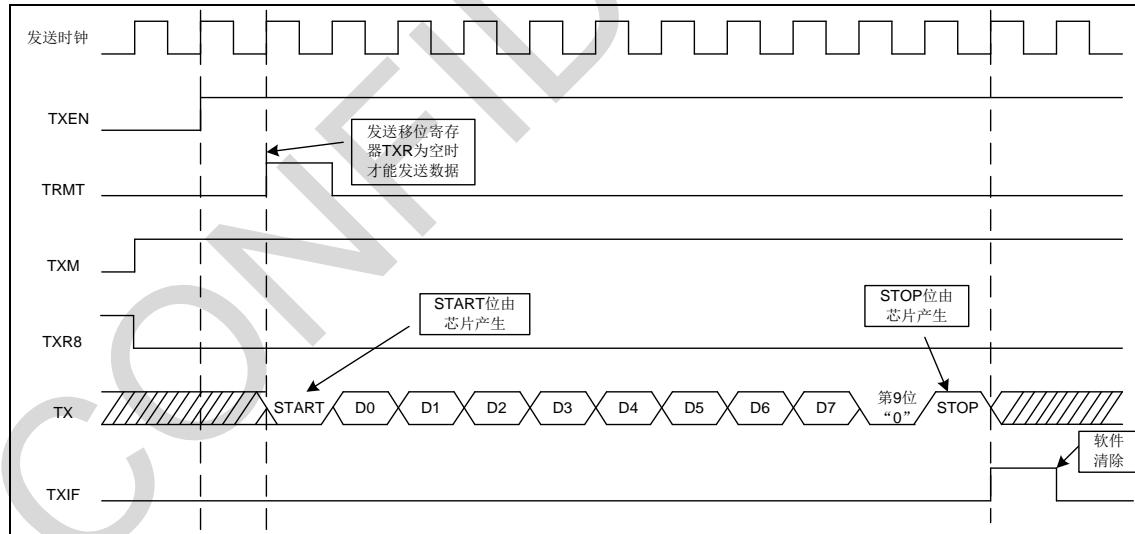


图 6-60 UART 发送器发送数据时序图 (9 位数据格式, 第 9 位数据为“0”)

6.4.6 异步接收器

异步接收器接收数据时，用户可以查询 RXIF 中断标志位，来判断是否收到完整的一帧数据，并通过读取 RXB 和 RXR8 获得数据。芯片内部提供 2 级 9 位 FIFO 作为接收数据缓冲区，若用户在第 3 个数据接收完毕前，未读取 RXB，则溢出标志位 OERR 将置 1。FERR 在用户未接收到结束位 STOP 时置 1。注意，由于 UART 接收器接收端口 RX 与 I/O 端口复用，在使用 UART 接收端口前，需首先设置所复用的 I/O 端口处于输入状态。

操作流程图如下：

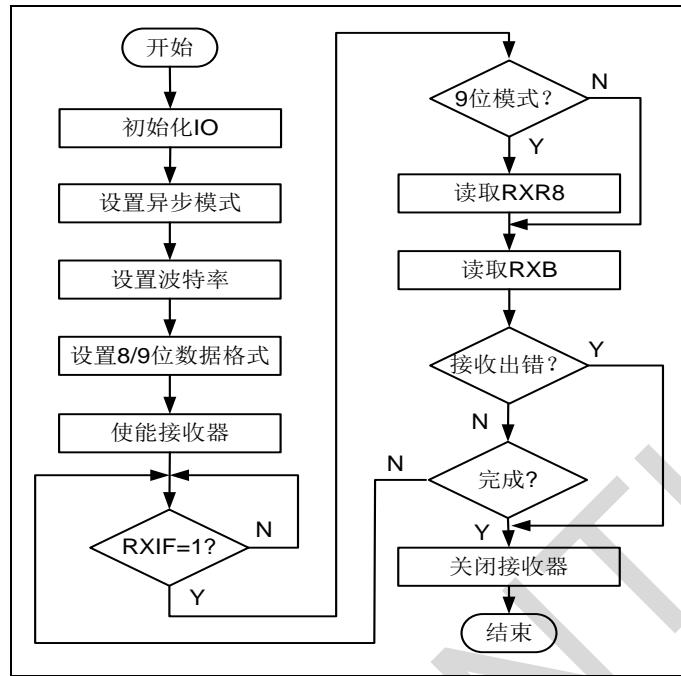


图 6-61 UART 异步接收器操作流程图

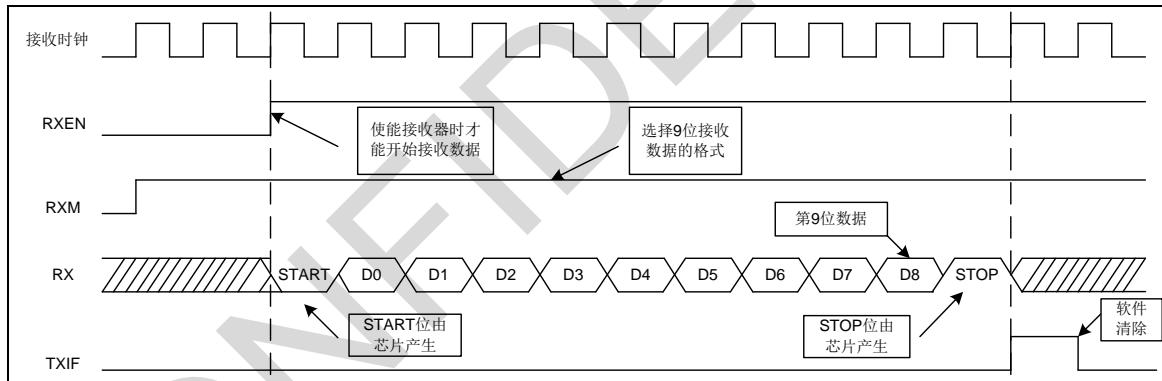


图 6-62 UART 接收器接收数据时序图 (9 位数据格式)

6.4.7 UART 使用注意事项

在 UART 模块使能前，首先要设置复用的 I/O 端口为数字端口，并将 TX 管脚设置为输出口，RX 管脚设置为输入口，才能保证在 UART 模块使能后，成功进行数据的发送或接收。另外，程序中不建议用户频繁切换 TX 和 RX 管脚的输入/输出类型。

当配置字 CFGUART (CFG_WD<8>) 为 1 时，PC1 为 TX 发送端口，PC0 为 RX 接收端口；当配置字 CFGUART (CFG_WD<8>) 为 0 时，PC1 为 TX/RX 分时发送/接收端口，此时，需要用户软件进行 RXEN 和 TXEN 的设置，以及相应的输入/输出控制寄存器的设置。

6.4.8 特殊功能寄存器

UART 模块相关的有 5 个寄存器，包括一个接收数据寄存器 RXB、一个接收控制/状态寄存器 RXC、一个发送数据寄存器 TXB、一个发送控制/状态寄存器 TXC，以及一个波特率寄存器 BRR。RXB 和 TXB 寄存器分别用于存放接收到的数据和待发送的数据，RXC 寄存器用于接收器的使能控制、接收数据格式的选择、以及接收溢出标志、帧格式错标志和存放第 9 位接收数据等。TXC 寄存器用于发送器的使能控制、发送数据格式的选择、波特率模式的选择、存放第 9 位发送数据以及发送移位寄存器（TXR）空标志的显示等。BRR 寄存器用于 UART 的波特率设置，设置范围为 $00_{\text{H}} \sim FF_{\text{H}}$ 。

6.4.8.1 UART 接收数据寄存器 (RXB)

RXB: UART 接收数据寄存器								
Bit	7	6	5	4	3	2	1	0
Name	RXB<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 RXB<7:0>: UART 接收到的数据

6.4.8.2 UART 接收控制/状态寄存器 (RXC)

RXC: UART 接收控制/状态寄存器								
Bit	7	6	5	4	3	2	1	0
Name	RXEN	RXM	—	—	—	OERR	FERR	RXR8
R/W	R/W	R/W	—	—	—	R	R	R
POR	0	0	0	0	0	0	0	x

Bit 7 RXEN: 接收器使能位

- 0: 禁止
- 1: 使能

Bit 6 RXM: 接收器数据格式选择位

- 0: 8 位数据接收格式
- 1: 9 位数据接收格式

Bit 5~3 未使用

Bit 2 OERR: 接收溢出标志位

- 0: 无溢出错误
- 1: 有溢出错误 (清 RXEN 位可将此位清除)

Bit 1 FERR: 帧格式错标志位

- 0: 无帧格式错误
- 1: 帧格式错 (读 RXB, 该位被刷新)

Bit 0 RXR8: 第 9 位接收数据位

- 0: 第 9 位数据为 0
- 1: 第 9 位数据为 1

6.4.8.3 UART发送数据寄存器 (TXB)

TXB: UART 发送数据寄存器								
Bit	7	6	5	4	3	2	1	0
Name	TXB<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 TXB<7:0>: UART 发送的数据

6.4.8.4 UART发送控制/状态寄存器 (TXC)

TXC: UART 发送控制/状态寄存器								
Bit	7	6	5	4	3	2	1	0
Name	TXEN	TXM	BRGH	—	—	—	TRMT	TXR8
R/W	R/W	R/W	R/W	—	—	—	R	R/W
POR	0	0	0	0	0	0	1	0

Bit 7 TXEN: 发送器使能位

0: 禁止

1: 使能

Bit 6 TXM: 发送器数据格式选择位

0: 8 位数据发送格式

1: 9 位数据发送格式

Bit 5 BRGH: 波特率模式选择位

0: 低速模式

1: 高速模式

Bit 4~2 未使用

Bit 1 TRMT: 发送移位寄存器 (TXR) 空标志位

0: TXR 不空

1: TXR 空

Bit 0 TXR8: 第 9 位发送数据设置位

0: 第 9 位数据为 0

1: 第 9 位数据为 1

6.4.8.5 UART波特率寄存器 (BRR)

BRR: UART 波特率寄存器								
Bit	7	6	5	4	3	2	1	0
Name	BRR<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 BRR<7:0>: UART 波特率设置, 00H~FFH

6. 5 I2C总线从动器 (I2CS)

6. 5. 1 概述

本系列芯片支持一路 I2C 从动器，从动模块支持 7 位从机地址匹配，由 I2C 主机控制发送或接收数据。

- 只支持从动模式
- 支持 7 位从机地址
- 支持标准 I2C 总线协议，最大传输速率 400Kbit/s
- 支持 I2CS 端口 SCL/SDA 开漏或者推挽输出
- 支持 2 级发送/接收缓冲
- 支持自动时钟下拉等待功能
- 支持自动发送“未应答”功能
- 约定数据从最高位开始接收/发送
- ◇ I2CS 支持以下功能组件
 - 5 位 I2C 采样滤波寄存器 (I2CX16)
 - I2C 控制寄存器 (I2CC)
 - 从机地址寄存器 (I2CSA)
 - 发送数据缓冲器 (I2CTB)
 - 接收数据缓冲器 (I2CRB)
 - 中断使能寄存器 (I2CIEC)
 - 中断标志寄存器 (I2CIFC)
- ◇ 中断和暂停
 - 支持接收“起始位+从机地址匹配+发送应答位”中断标志 (I2CSRIF)
 - 支持接收结束位中断标志 (I2CSPIF)
 - 支持发送空中断标志 (I2CTBIF, 只可读)
 - 支持接收满中断标志 (I2CRBIF, 只可读)
 - 支持发送错误标志 (I2CTEIF)
 - 支持接收溢出中断标志 (I2CROIF)
 - 支持接收未应答标志 (I2CNAIF)
 - 在 IDLE 模式下，如果禁止采样滤波，那么“起始位+从机地址匹配+发送应答位”可唤醒 CPU

注：本节中 SDA/SCL 端口代表的是芯片管脚图中的 I2C_SDA/I2C_SCL 端口。

6.5.2 内部结构

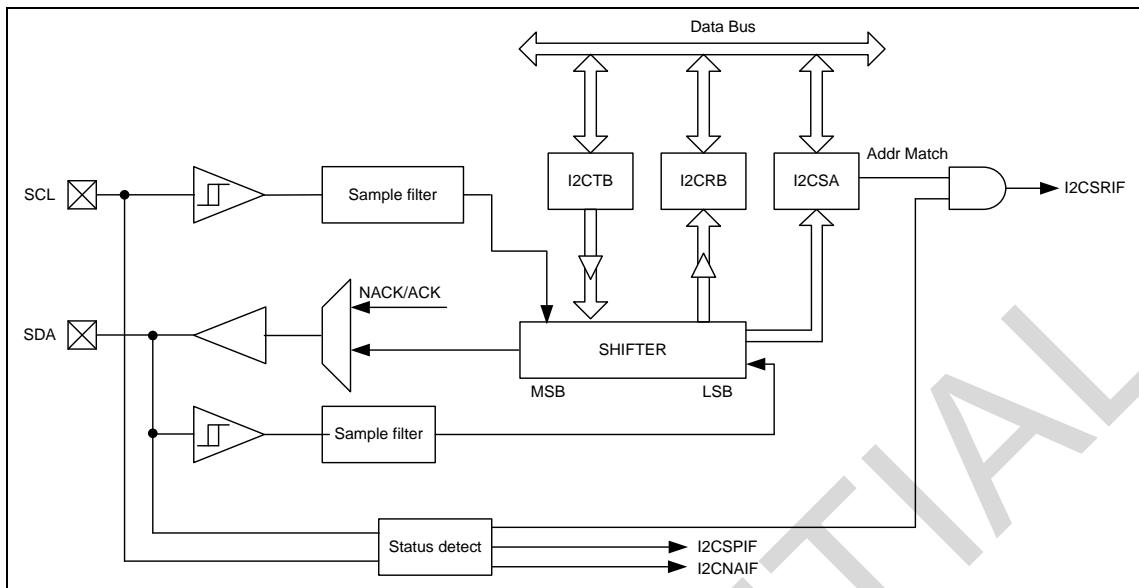


图 6-63 I2C 内部结构

6.5.3 I2CS 端口配置

I2C 总线从动器包括一条串行数据线 SDA 和一条串行时钟线 SCL。I2C 端口与普通 I/O 端口复用，可通过 I2CC 寄存器中的 I2CTE 位选择复用端口的功能。当 I2CTE=0 时，复用端口作为普通 I/O；当 I2CTE=1 时，复用端口作为 I2C 通信端口使用。

I2CTE	I2CS 时钟端口配置	I2CS 数据端口配置
1	SCL	SDA
0	PB1	PB0

表 6-7 I2CS 端口配置表

I2C 通讯端口 SCL 和 SDA，均支持推挽输出和开漏输出两种模式，配置 I2CC 寄存器的 I2COD，可分别进行选择。

推挽输出是 I/O 端口的标准输出，输出数据 0 和 1 时，I/O 端口电平也分别为 0 和 1。

对推挽输出模式，存在端口电平冲突的风险。例如：当主控器输出 0，而从动器输出 1 时，会发生端口信号电平冲突，导致端口状态不确定。

开漏输出是 I2C 总线协议中的标准模式，可以避免端口电平冲突问题。开漏输出端口的示意图如下所示：

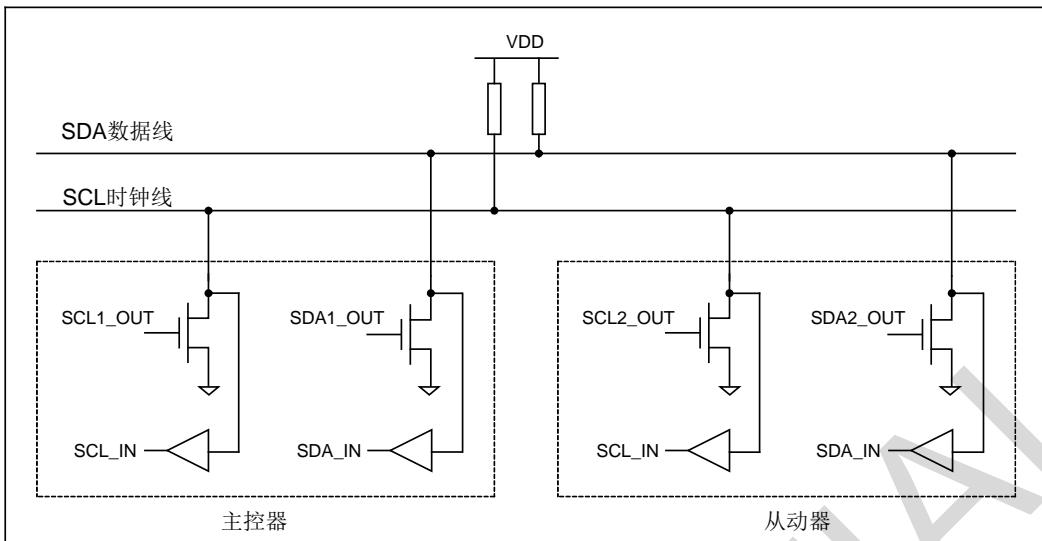


图 6-20 开漏输出端口示意图

开漏输出端口的高电平由 I2C 总线的上拉电阻提供，而低电平由主控器与从动器共同决定。任何一方都可以将总线电平下拉到 0，但只有当双方都释放总线后，总线电平才能被上拉到 1。

6.5.4 I2C 通讯协议

I2C 通讯中，必须遵循以下协议：

- ◇ 通讯由主控器发起，发送启动信号 S（开始）控制总线，发送停止信号 P（结束）释放总线；
- ◇ 总线上可以同时有多个主控器（前提是每个主控器都支持多主机仲裁机制），但至少需要一个从动器，且每一个从动器都必须有一个独立且唯一的寻呼地址；
- ◇ 主控器在发送启动信号后，紧接着寻呼从动器地址以及发送读写方式位；
- ◇ 读写控制位 R/#W（称为方向位）用于通知从动器数据传送的方向，“0”表示这次通讯是由主控器向从动器“写”数据，“1”表示这次通讯是由主控器向从动器“读”数据；
- ◇ I2C 通讯协议支持应答机制，即发送方每传送一个字节的数据（包括寻呼地址），接收方必须回答一个应答信号(ACK 或者 NACK)，发送方再根据应答信号进行下一步的操作；
- ◇ 如果主控器和从动器的时钟线（SCL）都使用开漏设计，且主控器支持时钟线等待请求操作，那么从动器可以在时钟线为低电平时下拉时钟线，使主控器等待从动器，直到从动器释放时钟线；
- ◇ 每个数据字节在传送时都是高位在前。

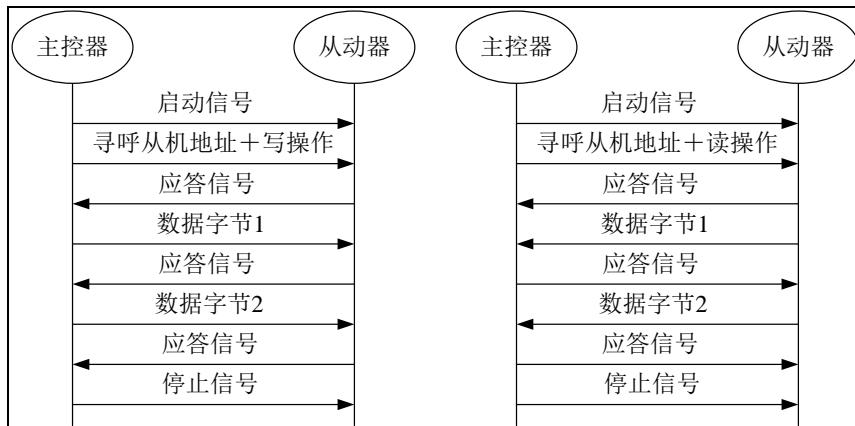


图 6-64 I2C 总线通讯协议示意图

6.5.5 I2C操作

I2C 总线上可以同时有多个主控器（前提是每个主控器都支持多主机仲裁机制），但至少需要一个从动器。总线上的每个设备都没有选择线，但分别与唯一的地址一一对应，用于 I2C 通信。

从动模块包括主机向从机发送数据以及主机读取从机数据两种操作方式。

当主机向从机发送数据时，从机通常判断接收满中断标志位 I2CRBIF，如果接收缓冲器不空，即接收到主机数据，则读接收缓冲器的数据。

当主机读取从机数据时，从机通常判断发送缓冲器未满中断标志位 I2CTBIF，如果发送缓冲器未满，则依次写入需要发送的数据。

为了避免误发数据，建议每次完整的通讯结束（例如收到 STOP 标志），就采用软件复位 I2C 模块位 I2CRST 置位复位一次 I2C 模块来清空接收和发送数据缓冲器，同时再重新初始化 I2CC 和 I2CIEC 寄存器，为下次 I2C 通讯做好准备。（I2C Master 给 I2C Slaver 发送读命令，发送从机地址后，需要等待大约 30us 左右再读取数据）

在 IDLE 模式下，如果禁止采样滤波，I2CWKUPEN 寄存器为 1，唤醒使能，则“起始位+从机地址匹配+发送应答位”可唤醒 CPU。

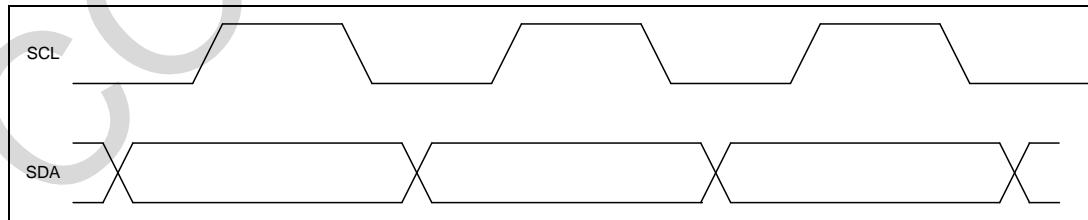


图 6-65 I2C 从动波形图

6.5.6 起始位START和停止位STOP

根据 I2C 协议：在数据传输期间，当 SCL 为高电平时，SDA 必须保持固定电平，波形如图所示；在没有数据传输期间，SCL 和 SDA 应该保持高电平。当 SCL 线是高电平时，SDA 线从高电平向低电平切换表示起始条件（S）。当 SCL 是高电平时，SDA 线由低电平向高电平切换表示停止条件（P）。起始和停止条件一般由主机产生，如图所示。

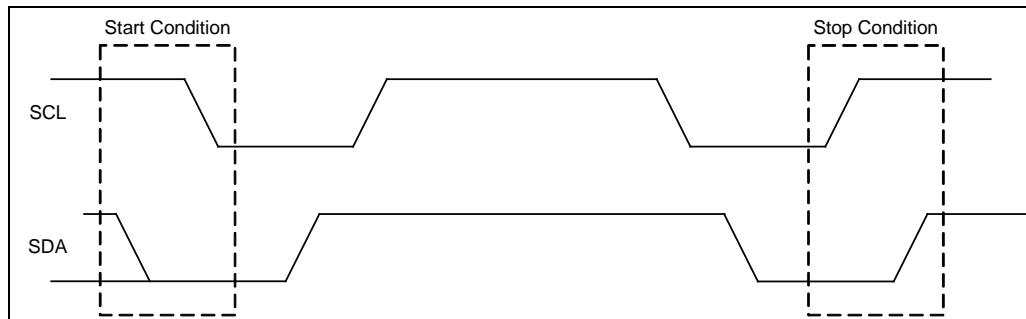


图 6-66 I2C 起始位和停止位

6.5.7 数据传输和应答

进入起始条件 (S) 之后，数据按照一个字节串行传输 (8 位)。接收方每次接收完毕 8 位数据之后，需要发送应答信号给发送方。当数据传输到第 8 个 SCL 下降沿时，接收方立即发送应答信号，此时发送方释放 SDA 控制权，而接收方将 SDA 变为低电平。当发送一个字节的数据紧邻于先前一个字节的接收（或者当接收方切换到发送方，并开始数据传输）时，接收方在第 9 个 SCL 下降沿释放 SDA 控制权。

当主机为发送方时，如果自动未应答 I2CNAE 位不使能，寻呼地址匹配后，从机会发送 ACK 应答信号，和 I2CTAS 选择位无关；如果自动未应答 I2CNAE 位使能，寻呼地址匹配后，从机发送 ACK 还是 NACK 信号，如果 2 级接收缓冲器未空，则发送 NACK 信号，如果未满则发送 ACK 信号，与 I2CTAS 选择位无关；从机接收到数据时，如果自动未应答 I2CNAE 位使能，时钟下拉等待 I2CCSE 位不使能，且 2 级接收缓冲器满，一个字节的数据接收完成后，会发送 NACK 信号；否则发送 ACK 或 NACK，受 I2CTAS 选择位控制。

当主机为接收方时，如果自动未应答 I2CNAE 位不使能，寻呼地址匹配后，从机会发送 ACK 应答信号，和 I2CTAS 选择位无关；如果自动未应答 I2CNAE 位使能，寻呼地址匹配后，从机发送 ACK 还是 NACK 信号，如果 2 级发送缓冲器未满，则发送 NACK 信号，如果满则发送 ACK 信号，与 I2CTAS 选择位无关。每接收一个字节的数据时，主机发出 ACK 应答信号，当接收到从机的最后一个字节后不产生应答信号，以告知发送设备数据传输完成。在第 9 个 SCL 下降沿，从机（发送方）继续释放 SDA 控制权以便主机可以产生停止条件 (P)。

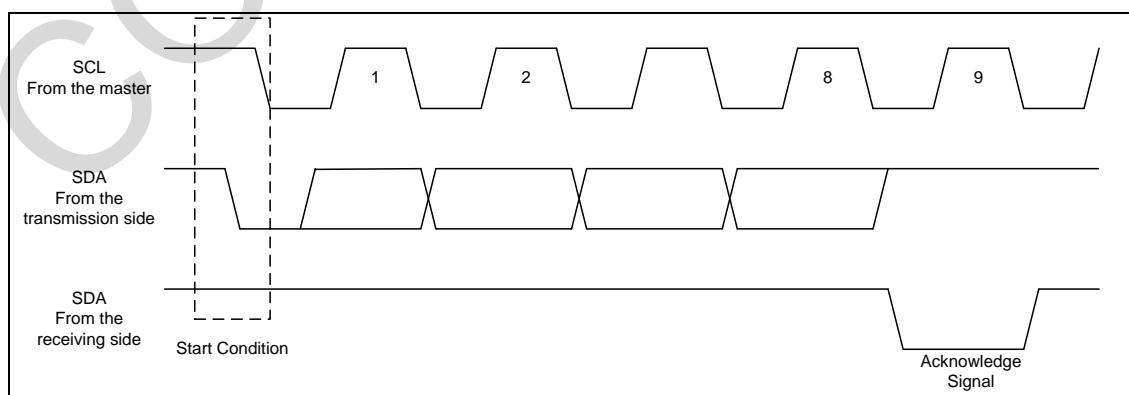


图 6-67 数据传输和应答

6.5.8 数据传输格式参考

I2C 存储器的数据传输参考格式如下：

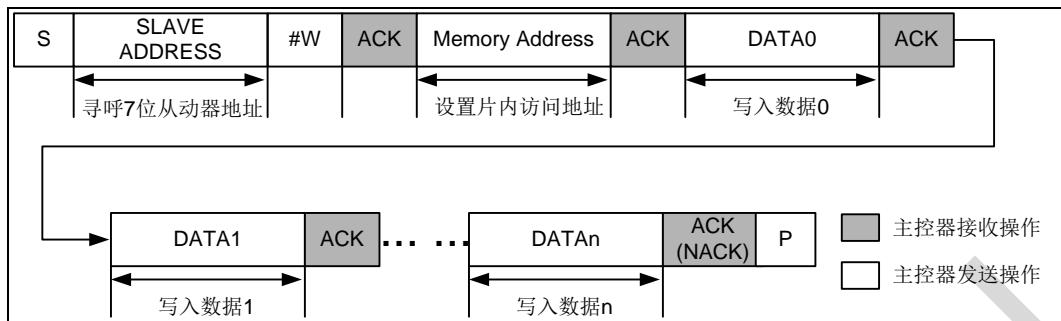


图 6-68 主控器写入从动器数据示意图

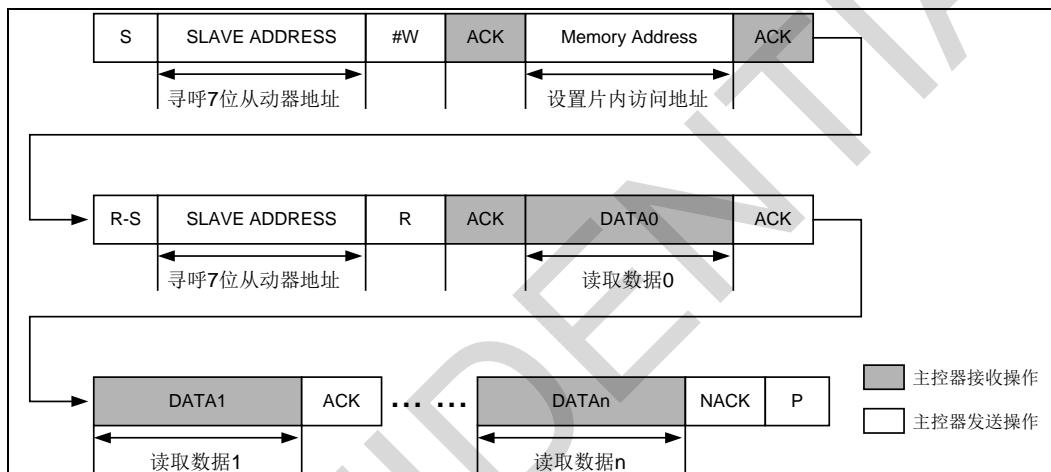


图 6-69 主控器读取从动器数据示意图

6.5.9 I2C 通讯发送器

支持 2 级发送缓冲器 TB0, TB1 和 1 级发送移位寄存器，可进行数据的连续发送，直到发送缓冲器和移位寄存器全空，最多可连续写入和发送 3 帧数据。发送缓冲器 TB0~TB1 为只读寄存器，只能通过发送数据寄存器 I2CTB 写入。

写 I2CTB 寄存器地址单元时，实际上是将发送数据写入到发送缓冲器 TB0~TB1 中，其中低字节存放在 TB1 中，再传输到发送移位寄存器，通过数据端口 SDA 进行数据发送。

发送数据从写入到发送到端口的数据流示意图如下所示：



图 6-26 I2C 发送数据流示意图

支持发送缓冲器空中断，发送缓冲器 TB0 和 TB1 均为空时，会置起 I2CIFC 寄存器的中断标志 I2CTBIF；

支持发送数据寄存器 I2CTB 误写中断，当对 I2CTB 的写入方式，与发送缓冲器 TB0~TB1 的空闲状态冲突时，或写入操作访问错误时，会置起 I2CIFC 寄存器误写中断标志 I2CTEIF。

6.5.10 I2C通讯接收器

支持 2 级接收缓冲器 RB0, RB1 和 1 级接收移位寄存器，可进行数据的连续接收，直到接收缓冲器和移位寄存器全满，最多可连续接收 3 帧数据，再执行数据读取操作。读取接收数据寄存器 I2CRB，可得到接收的数据。

读取 I2CRB 时，实际是同时读取接收缓冲器 RB0 和 RB1 的数据，其中 RB0 中的数据为低字节。

接收数据从数据端口到各级缓冲器的数据流示意图如下所示：



图 6-27 I2C 接收数据流示意图

当 2 级接收缓冲器和 1 级接收移位寄存器均满时，会立即置起接收数据溢出中断标志 I2CIFC 寄存器中 I2CROIF，同时不会接收新数据。

支持接收缓冲器满中断，接收缓冲器 RB0 和 RB1 均为满时，会置起 I2CIFC 寄存器的中断标 I2CRBIF。

6.5.11 I2C时钟线自动下拉等待请求功能

I2C 从动模式，支持时钟线自动下拉等待请求功能，配置 I2CC 寄存器的 I2CCSE，可使能该功能。

为实现 I2C 时钟线的下拉等待请求功能，还需配置 I2CC 寄存器的 I2COD，将通讯端口 SCL 选择为开漏输出模式，通过上拉电阻提供高电平，使从动器可对时钟线下拉控制，使主控器等待。

在通常情况下，从动器处于释放时钟线的状态，时钟线 SCL 完全由主控器控制。但当从动器出现异常情况，短时间内无法继续进行数据传输时，从动器可以在时钟线 SCL 为低电平时输出 0（不可以在高电平时输出 0，否则会破坏数据传输过程），强行使 SCL 保持低电平，使主控器进入通讯等待状态，直到从动器释放时钟线。时钟线下拉等待请求波形示意图如下：

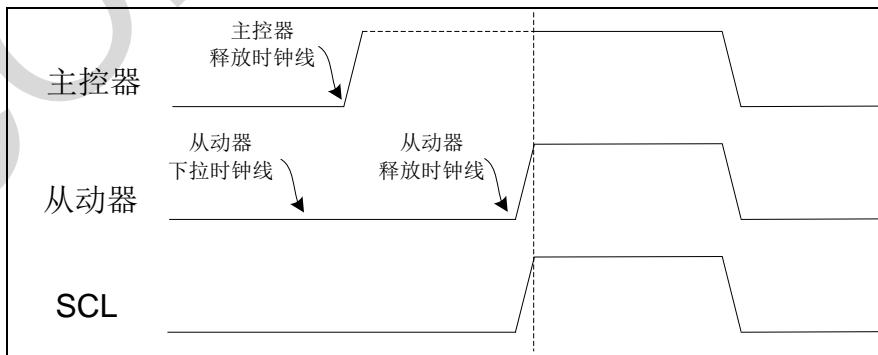


图 6-70 I2C 时钟线下拉等待波形示意图

I2C 从动模式下，时钟自动下拉等待请求功能使能后，当接收到本芯片寻呼地址、读操作位和应答信号时，如果 I2C 的发送缓冲器和发送移位寄存器全空，则会自动将时钟线下拉，等待发送缓冲器写入数据；当接收到本芯片寻呼地址、写操作位和应答信号时，

如果 I²C 的接收缓冲器和接收移位寄存器全满，则会自动将时钟线下拉，等待接收缓冲器被读取。

6.5.12 特殊功能寄存器

6.5.12.1 I²C采样滤波寄存器 (I²CX16)

I ² CX16: I ² C 采样滤波寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	—	—	I ² CX16<4:0>				
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5 未使用

Bit 4~0 I²CX16<4:0>：采样滤波控制位

00_H: 禁止采样滤波

01_H~1F_H: 通信时钟和数据采样滤波时间为 T_{osc} × (I²CX16+1) × 3

6.5.12.2 I²C控制寄存器 (I²CC)

I ² CC: I ² C 控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	I ² CTE	I ² CPU	I ² COD	I ² CTAS	I ² CANAE	I ² CCSE	I ² CRST	I ² CEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 I²CTE: I²C 通信端口使能位

0: 禁止

1: 使能

Bit 6 I²CPU: I²C 内部弱上拉使能

0: 禁止

1: 使能

当 I²CTE=1 时，I²CPU 控制 SCL/SDA 端口的弱上拉功能。

Bit 5 I²COD: I²C 开漏输出使能位

0: 推挽输出

1: 开漏输出

Bit 4 I²CTAS: I²C 发送应答设置位

0: 发送 ACK

1: 发送 NACK

Bit 3 I²CANAE: I²C 自动未应答使能位

0: 禁止

1: 使能

Bit 2 I²CCSE: I²C 时钟下拉等待使能位

0: 禁止

1: 使能

Bit 1 I²CRST: 软件复位 I²C 模块位

0: 禁止

Bit 0	1: 使能 (复位后, 硬件自动清零)
	I2CEN: I2C 模块使能位
	0: 禁止 1: 使能

注 1:

1) 使能时钟自动下拉功能后 (且使能发送 ACK 功能 (I2CTAS=0)), 当从机收到匹配的寻呼地址时:
 1.1) 如果为主机读操作, 当发送缓冲器和移位寄存器全空时, 先发送完 ACK, 再下拉时钟, 写完发送的数据后, 释放时钟线, 直接发送数据位;
 1.2) 如果为主机写操作, 当接收缓冲器和移位寄存器全满时, 先发送完 ACK, 再下拉时钟, 读取已接收的数据后, 释放时钟线, 接收新的数据位。

注 2: 当 I2C 自动未应答使能,
当片外主控器寻呼本机地址+R 时, 若 2 级发送数据缓冲器全空, 则本机地址后的应答位为“NACK”;
当片外主控器寻呼本机地址+#W 时, 若在数据接收之前, 2 级接收数据缓冲器全满时, 本机地址后的应答位为“NACK”; 若在接收数据后, 且 I2CCSE=0, 2 级接收数据缓冲器全满时, 则接收数据后的应答位为“NACK”。

注 3: 当 I2CTE=1 时, I2CPU 控制 SCL/SDA 端口的弱上拉功能; 否则, 由 PBPU<1:0>控制 PB 端口的弱上拉功能。

6.5.12.3 I2C从机地址寄存器 (I2CSA)

I2CSA: I2C 从机地址寄存器								
Bit	7	6	5	4	3	2	1	0
Name	I2CSADR<6:0>							I2CRW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
POR	0	0	0	0	0	0	0	0

Bit 7~1	I2CSADR<6:0>: 从机地址 接收到“启动/重启动”后用于匹配比较
Bit 0	I2CRW: 从机地址匹配后, 自动更新读/写位 0: 写 1: 读

6.5.12.4 I2C发送数据缓冲器 (I2CTB)

I2CTB: I2C 发送数据缓冲器								
Bit	7	6	5	4	3	2	1	0
Name	I2CTB<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 I2CTB<7:0>: 发送数据缓冲器

注: 第一个需要发送的数据, 在发送使能前写入发送数据缓冲器。

6.5.12.5 I2C接收数据缓冲器 (I2CRB)

I2CRB: I2C 接收数据缓冲器								
Bit	7	6	5	4	3	2	1	0
Name	I2CRB<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 I2CRB<7:0>: 接收数据缓冲器

6.5.12.6 I2C中断使能寄存器 (I2CIEC)

I2CIEC: I2C 中断使能寄存器								
Bit	7	6	5	4	3	2	1	0
Name	I2CWUPEN	I2CNAIE	I2CROIE	I2CTEIE	I2CRBIE	I2CTBIE	I2CSPIE	I2CSRUE
R/W	R/W							

Bit 7 I2CWUPEN: I2C 地址匹配唤醒使能位

- 0: 禁止
- 1: 使能

Bit 6 I2CNAIE: I2C 接收未应答中断使能位

- 0: 禁止
- 1: 使能

Bit 5 I2CROIE: I2C 接收溢出中断使能位

- 0: 禁止
- 1: 使能

Bit 4 I2CTEIE: I2C 发送错误中断使能位

- 0: 禁止
- 1: 使能

Bit 3 I2CRBIE: I2C 接收满中断使能位

- 0: 禁止
- 1: 使能

Bit 2 I2CTBIE: I2C 发送缓冲器未满中断使能位

- 0: 禁止
- 1: 使能

Bit 1 I2CSPIE: I2C 接收结束位中断使能位

- 0: 禁止
- 1: 使能

Bit 0 I2CSRUE: I2C 接收“起始位+从机地址匹配+发送应答位”中断使能位

- 0: 禁止
- 1: 使能

6.5.12.7 I²C中断标志寄存器 (I²CIFC)

I ² CIFC: I ² C 中断标志寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	I ² CNAIF	I ² CCROIF	I ² CTEIF	I ² CRBIF	I ² CTBIF	I ² CSPIF	I ² CSRIF
R/W	—	R/W	R/W	R/W	R	R	R/W	R/W
POR	1	0	0	0	0	1	0	0
Bit 7	未使用							
Bit 6	I ² CNAIF: I ² C 接收未应答中断标志位 0: 未接收或者未发送 NACK 1: 接收或发送 NACK, 产生中断标志 (软件清零)							
Bit 5	I ² CCROIF: I ² C 接收溢出中断标志位 0: 2 级接收数据缓冲器和 I ² C 移位寄存器未溢出 1: 2 级接收数据缓冲器和 I ² C 移位寄存器溢出, 产生中断标志 (软件清零)							
Bit 4	I ² CTEIF: I ² C 发送错误中断标志位 0: 未发生发送错误 1: 发生发送错误: 2 级发送数据缓冲器全空, 又收到主控提供的通讯时钟。产生中断标志 (软件清零)							
Bit 3	I ² CRBIF: I ² C 接收满中断标志位 0: 2 级接收数据缓冲器未满 1: 2 级接收数据缓冲器满时, 产生中断标志 (读 I ² CRB 寄存器可清除中断标志位)							
Bit 2	I ² CTBIF: I ² C 发送缓冲器未满中断标志位 0: 2 级发送数据缓冲器满 1: 2 级发送数据缓冲器未满时, 产生中断标志 (写 I ² CTB 寄存器可清除中断标志位)							
Bit 1	I ² CSPIF: I ² C 接收结束位中断标志位 0: 未接收到结束位 1: 接收到结束位, 产生中断标志 (软件清零)							
Bit 0	I ² CSRIF: I ² C 接收“起始位+从机地址匹配+发送应答位”中断标志位 0: 未接收到“起始位+地址位且地址匹配+发送应答位” 1: 接收到“起始位+地址位且地址匹配+发送应答位”后, 产生中断标志 (软件清零)							

注 1: 清总中断标志位 I²CIF 前, 先清除 I²CIFC 寄存器的相关中断标志位。

注 2: 连续接收数据超过 2 个时, 发生接收溢出, 并且第 3 个接收数据会丢失。

注 3: I²C 模块在每帧数据发送完成后, 接收到结束位时, 硬件自动清零发送缓冲寄存器。

6.6 SPI同步串口通讯模块（SPI）

6.6.1 概述

- ◇ 支持主控模式、从动模式
- ◇ 支持 4 种数据传输格式
- ◇ 支持主控模式通讯时钟速率可配置
- ◇ 支持 4 级发送缓冲器和 4 级接收缓冲器
- ◇ 支持发送和接收缓冲器空/满中断
- ◇ 支持接收数据溢出中断、发送数据写错误中断、从动模式的发送数据错误中断
- ◇ 支持从动模式的片选变化中断、主控模式的空闲状态中断
- ◇ 支持主控模式延迟接收

注：本节中 SCK/NSS/MISO/MOSI 端口代表的是芯片管脚图中的 SPI_SCK/SPI_NSS/SPI_MISO/SPI_MOSI 端口。

6.6.2 结构框图

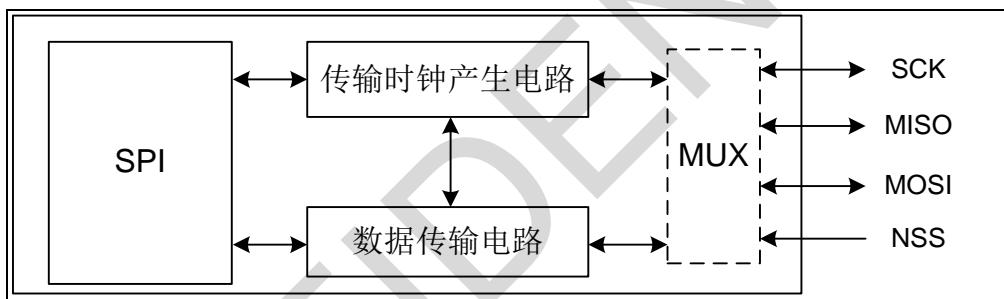


图 6-71 SPI 电路结构框图

6.6.3 SPI 通讯模式

SPI 支持主控和从动两种通讯模式，配置 SPICON1 寄存器的 MS 位，可选择通讯模式。

通讯时钟端口为 SCK，从动模式下的片选信号端口为 NSS，数据输出端口 MISO，数据输入端口 MOSI，主控模式下可使用普通 I/O 端口作为片外从设备的片选信号端口，数据输出端口 MOSI，数据输入端口 MISO。具体见下表所示：

SPI 通讯端口	SPI 主控模式	SPI 从动模式
SCK	支持	支持
MOSI	支持	支持
MISO	支持	支持
NSS	—	支持

6.6.4 SPI 数据格式

配置 SPICON1 寄存器的 DFS，可选择 SPI 通讯数据格式，数据发送和接收时，均是高位在前，低位在后。如果是发送数据在先，接收数据在后，输出端口 MOSI(或 MISO)会在第一个 SCK 时钟边沿，输出数据的 MSB 位；反之输出端口 MOSI (或 MISO) 在第一个

SCK 时钟边沿之前，输出数据的 MSB 位。

以下以 SPI 从动通讯模式为例，对数据通讯时序进行说明。

SPICON1 寄存器中 $\text{DFS}\langle 1:0 \rangle = 00$ ，上升沿发送（先），下降沿接收（后）：

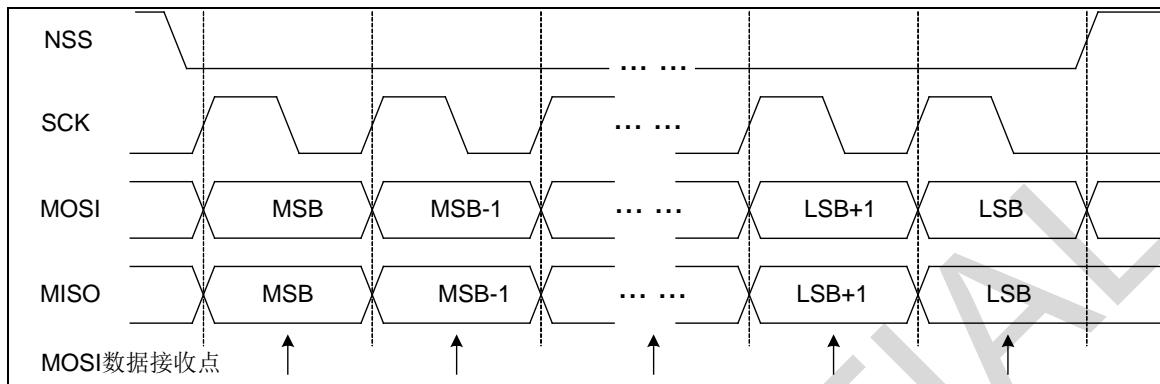


图 6-72 SPI 时钟上升沿发送, 下降沿接收波形示意图

$\text{DFS}\langle 1:0 \rangle = 01$ ，下降沿发送（先），上升沿接收（后）：

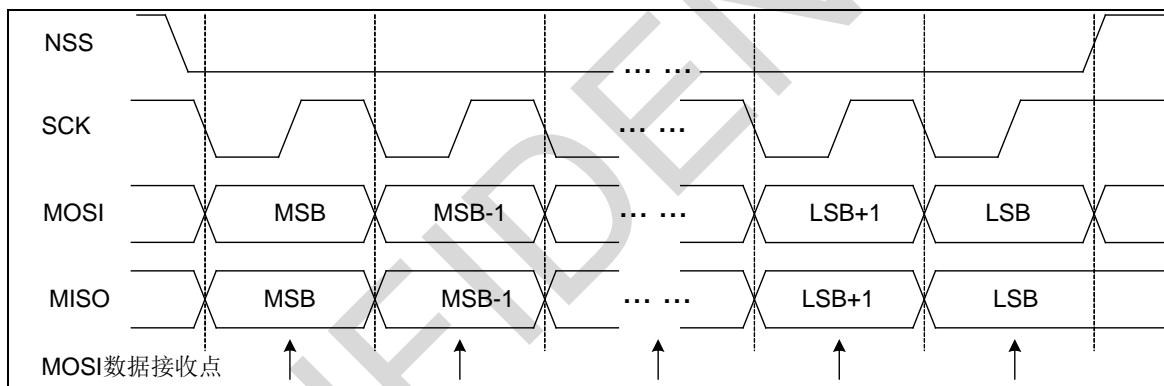


图 6-73 SPI 时钟下降沿发送, 上升沿接收波形示意图

$\text{DFS}\langle 1:0 \rangle = 10$ ，上升沿接收（先），下降沿发送（后）：

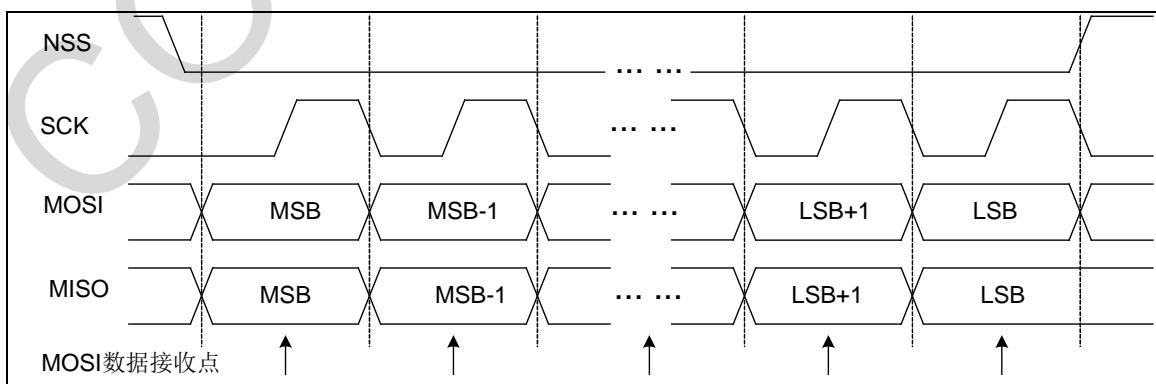


图 6-74 SPI 时钟上升沿接收, 下降沿发送波形示意图

DFS<1:0> = 11, 下降沿接收(先), 上升沿发送(后):

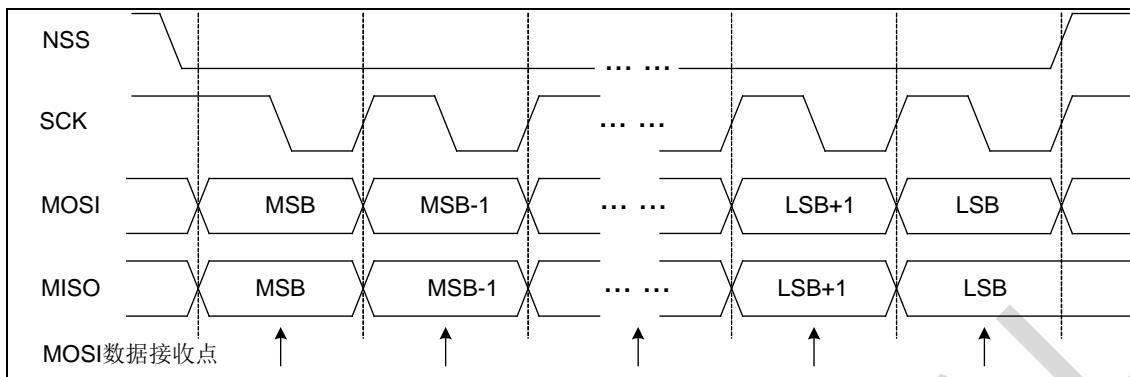


图 6-75 SPI 时钟下降沿接收, 上升沿发送波形示意图

6.6.5 SPI同步发送器

支持4级发送缓冲器TB0, TB1, TB2, TB3和1级发送移位寄存器, 可进行数据的连续发送, 直到发送缓冲器和移位寄存器全空, 最多可连续写入和发送5帧数据。发送缓冲器TB0~TB3, 发送移位寄存器为虚拟寄存器, 只能通过发送数据寄存器SPITBW写入。

发送数据寄存器SPITBW, 写该寄存器地址单元时, 实际上是将发送数据写入到发送缓冲器TB0中, 再逐级传输到发送移位寄存器, 通过发送数据端口MOSI(或MISO)进行数据发送。

发送数据从写入到发送到端口的数据流示意图如下所示(以主控模式为例):



图 6-76 SPI 发送数据流示意图

支持发送缓冲器空中断, 配置SPICON0寄存器的TBIM, 可选择中断模式。

TBIM<1:0>=00, 为字节空产生中断, 即发送缓冲器TB0为空时, SPIIF寄存器会置起中断标志TBIF;

TBIM<1:0>=01, 为半字空产生中断, 即发送缓冲器TB0和TB1均为空时, SPIIF寄存器会置起中断标志TBIF;

TBIM<1:0>=10, 为字空产生中断, 即发送缓冲器TB0~TB3均为空时, SPIIF寄存器会置起中断标志TBIF。

支持发送数据寄存器SPITBW误写中断, 当对SPITBW的写入方式, 与发送缓冲器TB0~TB3的空闲状态冲突时, 或写入操作访问错误时, SPIIF寄存器会置起误写中断标志TBWEIF。

6.6.6 SPI同步接收器

支持4级接收缓冲器RB0, RB1, RB2, RB3和1级接收移位寄存器, 可进行数据的连续接收, 直到接收缓冲器和移位寄存器全满, 最多可连续接收5帧数据, 再执行数据读取操作。RB0, RB1, RB2, RB3和1级接收移位寄存器为虚拟寄存器, 读取接收数据寄存器

SPIRBR，可得到接收的数据。

接收数据寄存器 SPIRBR，读该寄存器地址单元时，实际上是读取接收缓冲器 RB0 中的数据。

接收数据从接收端口到各级缓冲器的数据流示意图如下所示（以主控模式为例）：



图 6-77 SPI 接收数据流示意图

同步接收器的接收顺序如下：

当同步接收器全空时，接收移位寄存器的数据自动移入 RB0；

仅 RB1~ RB3 空时，接收移位寄存器的数据自动移入 RB1；

仅 RB2~ RB3 空时，接收移位寄存器的数据自动移入 RB2；

仅 RB3 空时，接收移位寄存器的数据自动移入 RB3。

当 4 级接收缓冲器和 1 级接收移位寄存器均满时，如果再次接收到数据位，SPIIF 寄存器中会置起接收数据溢出中断标志 ROIF，同时不会接收新数据，缓冲器数据仍保持。

支持接收缓冲器满中断，配置 SPICON0 寄存器的 RBIM，可选择中断模式。

RBIM<1:0>=00，为字节满产生中断，即接收缓冲器 RB0 为满时，SPIIF 寄存器中会置起中断标志 RBIF；

RBIM<1:0>=01，为半字满产生中断，即接收缓冲器 RB0 和 RB1 均为满时，SPIIF 寄存器中会置起中断标志 RBIF；

RBIM<1:0>=10，为字满产生中断，即接收缓冲器 RB0~RB3 均为满时，SPIIF 寄存器中会置起中断标志 RBIF。

6.6.7 SPI 通讯控制

配置 SPI 通讯模式，数据格式；对主控制模式，还需配置 SPICON0 的 CKS 寄存器，设定传输时钟速率，对从动模式，传输时钟由主机方提供；SPICON1 寄存器中配置 EN 和 REN，使能数据发送和接收；将要发送的数据写入发送数据寄存器 SPITBW，就可以开始数据的发送，读取接收数据寄存器 SPIRBR，可以获得接收到的数据。

SPI 主控模式下，发送缓冲器和发送移位寄存器中的数据发送完毕后，进入空闲状态，SPI_IF 寄存器中产生空闲中断标志 IDIF。

SPI 从动模式下，如果发送缓冲器 TB0~TB3 和发送移位寄存器均为空时，又收到主机提供的通讯时钟，则 SPIIF 寄存器中会置起发送错误中断标志 TEIF。

SPI 从动模式，支持片选信号变化中断，配置 SPIIE 寄存器的 NSSIE，可使能该中断。

配置 SPICON1 寄存器的 SPIRST 位，可将 SPI 通讯模块软件复位，复位后：禁止数据通讯 EN=0；SPIIE 寄存器中禁止相关中断 TBIE=0, TBWEIE=0, RBIE=0, TEIE=0, ROIE=0, IDIE=0, NSSIE=0；SPIIF 寄存器中复位相关中断标志为默认值 TBIF=1, TBWEIF=0, RBIF=0, TEIF=0, ROIF=0, IDIF=0, NSSIF=0。

6. 6. 8 SPI延迟接收功能

SPI 通讯时，是利用时钟的上升/下降沿分别对数据的发送和接收进行同步。正常通讯时，对主机接收数据来说，从机送出的数据应在半个时钟周期内，到达主机接收端口，否则会造成主机接收数据的丢失。

SPI 主控模式支持延迟接收功能，配置 SPICON1 寄存器的 DRE，可使能该功能，主机会再延迟半个时钟周期，在下一个发送时钟边沿处，进行数据的接收采集。所以延迟接收功能使能后，从机发送端口和主机接收端口之间的线路延时，最大可接近 1 个通讯时钟周期。

举例说明 SPI 延迟接收功能：SPICON1 寄存器中 DFS<1:0>=00，上升沿发送（先），下降沿接收（后）。

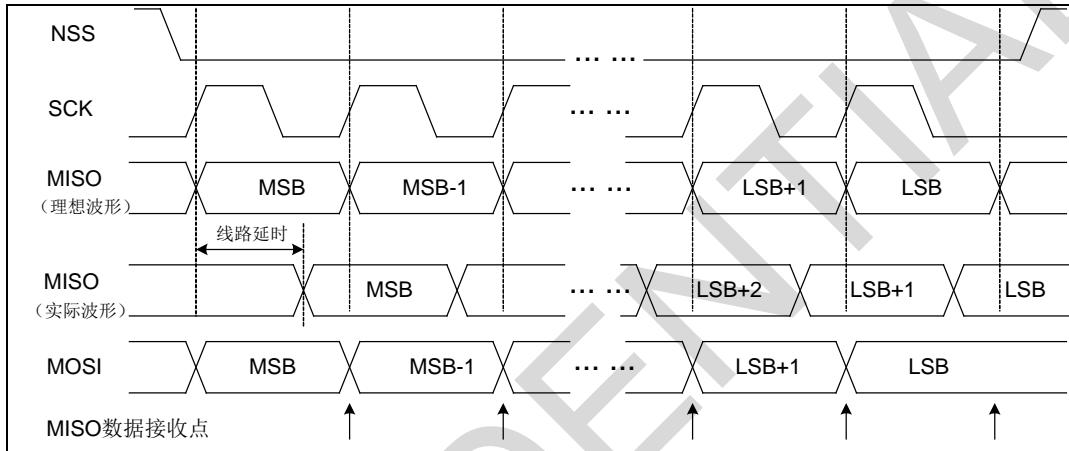


图 6-78 SPI 延迟接收功能波形示意图

6. 6. 9 SPI应用说明

为保证 SPI 的正常通信，SPI 的配置需遵循如下要求：

由于不同的通讯数据格式对端口的初始电平要求是不同的（参见上面各通讯波形示意图），因此若无法确定使能 SPI 之前的 SPI 端口的初始值，必须先配置通讯数据格式控制位，对 SPI 端口初始电平进行自动设置；然后再通过对 SPICON1 寄存器的 SPIEN 和 REN 置 1 来使能 SPI 发送和接收，即对 SPICON1 寄存器分两步写入。

6. 6. 10 特殊功能寄存器

6. 6. 10. 1 SPI控制寄存器 0 (SPICON0)

SPICON0: SPI 控制寄存器 0								
Bit	7	6	5	4	3	2	1	0
Name	RXCLR	TXCLR	CKS<1:0>		RBIM<1:0>		TBIM<1:0>	
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	1	0	0	0	0

Bit 7 RXCLR: SPI 接收缓冲器清空控制位

0: 无效

1: 清空接收缓冲器

Bit 6 TXCLR: SPI 发送缓冲器清空控制位

0: 无效

	1: 清空发送缓冲器
Bit 5~4	CKS<1:0>: SPI 通讯波特率选择位（仅主控模式支持）
	00: Fosc/2
	01: Fosc/4
	10: Fosc/8
	11: Fosc/16
Bit 3~2	RBIM<1:0>: SPI 接收缓冲器满中断模式选择位
	00: RB0 满产生中断
	01: RB0 与 RB1 满产生中断
	10: RB0~ RB3 全满产生中断
	11: 保留
Bit 1~0	TBIM<1:0>: SPI 发送缓冲器空中断模式选择位
	00: TB0 空产生中断
	01: TB0 与 TB1 空产生中断
	10: TB0~TB3 全空产生中断
	11: 保留

6. 6. 10. 2 SPI控制寄存器 1 (SPICON1)

SPICON1: SPI 控制寄存器 1								
Bit	7	6	5	4	3	2	1	0
Name	DFS<1:0>	DRE	—	REN	MS	SPIRST	SPIEN	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6	DFS<1:0>: SPI 通讯数据格式
	00: 上升沿发送（先），下降沿接收（后）
	01: 下降沿发送（先），上升沿接收（后）
	10: 上升沿接收（先），下降沿发送（后）
	11: 下降沿接收（先），上升沿发送（后）
Bit 5	DRE: SPI 延迟接收使能位（仅主控模式支持）
	0: 禁止
	1: 使能
Bit 4	保留未用
Bit 3	REN: SPI 接收使能位
	0: 禁止
	1: 使能（需 SPIEN 同时使能）
Bit 2	MS: SPI 通讯模式选择位
	0: 主控模式
	1: 从动模式
Bit 1	SPIRST: SPI 软件复位
	0: 读取时始终为 0
	1: 软件复位 SPI 模块，自动清零
Bit 0	SPIEN: SPI 通讯使能位
	0: 禁止

1: 使能 (SPI 通讯使能, 但仅使能数据发送)

注: 由于不同的通讯数据格式对端口的初始电平要求是不同的(参见上面各通讯波形示意图), 因此若无法确定使能 SPI 之前的 SPI 端口的初始值, 必须先配置通讯数据格式控制位, 对 SPI 端口初始电平进行自动设置; 然后再通过对 SPICON1 寄存器的 SPIEN 和 REN 置 1 来使能 SPI 发送和接收。即 SPICON1 寄存器需要分两次写入, 否则易产生通信错误。

6. 6. 10. 3 SPI发送数据写入寄存器 (SPITBW)

SPITBW: SPI 发送数据写入寄存器								
Bit	7	6	5	4	3	2	1	0
Name	TBW<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

Bit 7~0 TBW<7:0>: 写入的发送数据

字节写入时: 仅允许对 TBW<7:0>写入

6. 6. 10. 4 SPI接收数据读取寄存器 (SPIRBR)

SPIRBR: SPI 接收数据读取寄存器								
Bit	7	6	5	4	3	2	1	0
Name	RBR<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 RBR<7:0>: 读取的接收数据

字节读取时: 仅允许对 RBR<7:0>读取

6. 6. 10. 5 SPI中断使能寄存器 (SPIIE)

SPIIE: SPI 中断使能寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	TBWEIE	NSSIE	IDIE	ROIE	TEIE	RBIE	TBIE
R/W	—	R/W						
POR	0	0	0	0	0	0	0	0

Bit 7 未使用

Bit 6 TBWEIE: SPI 发送数据写错误中断使能位

0: 禁止

1: 使能

Bit 5 NSSIE: SPI 片选变化中断使能位 (仅从动模式支持)

0: 禁止

1: 使能

Bit 4 IDIE: SPI 空闲状态中断使能位 (仅主控模式支持)

0: 禁止

1: 使能

Bit 3	ROIE: SPI 接收数据溢出中断使能位 0: 禁止 1: 使能
Bit 2	TEIE: SPI 发送数据错误中断使能位 (仅从动模式支持) 0: 禁止 1: 使能
Bit 1	RBIKE: SPI 接收缓冲器满中断使能位 0: 禁止 1: 使能
Bit 0	TBIE: SPI 发送缓冲器空中断使能位 0: 禁止 1: 使能

6. 6. 10. 6 SPI中断标志寄存器 (SPIIF)

SPIIF: SPI 中断标志寄存器								
Bit	7	6	5	4	3	2	1	0
Name	—	TBWEIF	NSSIF	IDIF	ROIF	TEIF	RBIIF	TBIF
R/W	—	R/W	R/W	R/W	R/W	R/W	R	R
POR	0	0	0	0	0	0	0	1

Bit 7	未使用
Bit 6	TBWEIF: SPI 发送数据写错误中断标志位 0: 未发生写错误 1: 发生写错误, 可能会出现下列错误: 对 SPITBW 写入时, TB0~TB3 全满; 软件写 1 清除标志位, 写 0 无效
Bit 5	NSSIF: SPI 片选变化中断标志位 (仅从动模式支持) 0: 片选信号未发生变化 1: 片选信号发生变化 软件写 1 清除标志位, 写 0 无效
Bit 4	IDIF: SPI 空闲中断标志位 (仅主控模式支持) 0: 未进入空闲状态 1: 进入空闲状态 软件写 1 清除标志位, 写 0 无效; 或软件写寄存器 SPITBW 清除标志位
Bit 3	ROIF: SPI 接收数据溢出中断标志位 0: 未溢出 1: 溢出 软件写 1 清除标志位, 写 0 无效
Bit 2	TEIF: SPI 发送错误中断标志位 (仅从动模式支持) 0: 未发生发送错误 1: 发生发送错误: 发送缓冲器和发送移位寄存器全空时, 又收到主控方提供的通讯时钟 软件写 1 清除标志位, 写 0 无效
Bit 1	RBIIF: SPI 接收缓冲器满中断标志位 0: 接收缓冲器未满

	1: 接收缓冲器满 读 SPIRBR 寄存器可清除中断标志位
Bit 0	TBIF: SPI 发送缓冲器空中断标志位 0: 发送缓冲器未空 1: 发送缓冲器空 写 SPITBW 寄存器可清除中断标志位

注 1: SPI 中断禁止时, 如果满足条件仍会置起对应的中断标志位, 只是不会产生中断请求。

注 2: 对 SPIIF 寄存器中的各中断标志位, 写 0 无效, 写 1 才能清除标志位; 读操作时, 读取的值为 1 表示有中断发生。

CONFIDENTIAL

6.7 模/数转换器 (ADC)

6.7.1 概述

模拟数字转换器用于将模拟信号转化成一组二进制代码组成的数字信号。模拟信号经由多路复用输入脚输入，通过一个采样-保持电路连接至转换器的输入端。

芯片支持 12-bit 10 通道的 A/D 转换器，经过 A/D 转换器转换的 12-bit 二进制数据存入 ADC 数据寄存器 ADCRH、ADCRL 中。

◇ 模/数转换器特性

- 12 位 A/D 采样精度
- 8+2 个模拟输入通道可选
- 12 位转换结果，支持高位对齐放置或低位对齐放置
- 可配置 A/D 采样时间
- 多种转换时钟频率可选
- 可配置多种参考源，当使用外部参考电压时，参考电压不能低于 1.3V

◇ 主要功能组件

- ADC 转换值寄存器 (ADCRH, ADCRL)
- ADC 控制寄存器 (ADCCL, ADCCH, PWMxC)
- 数模端口控制寄存器 (ANSL)

◇ 中断和暂停

- 支持 AD 转换中断 (ADIE/ADIF)
- 在 IDLE 模式下，当使用 Fosc 时钟时，A/D 转换暂停；当使用 INTLRC 时钟时，A/D 继续转换

6.7.2 内部结构图

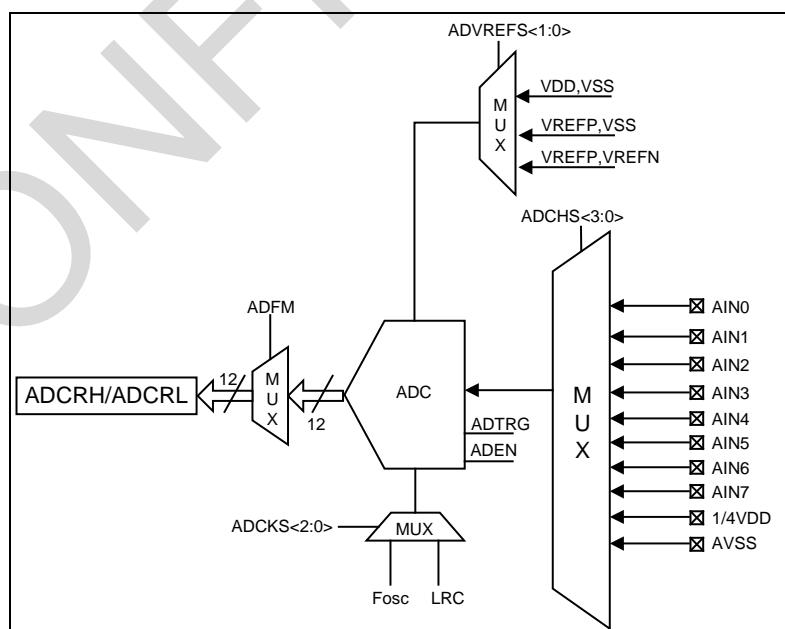


图 6-79 ADC 内部结构图

6.7.3 ADC配置

ADC 电路使用前，根据需要对以下几个方面进行正确的配置，才可得到正确转换结果。

时钟选择

ADC 电路的转换时钟有 8 组可选， $\text{Fosc}\sim\text{Fosc}/64$ 或 INTLRC，可通过 ADCCH 寄存器 ADCKS<2:0>位选择所需要的时钟。

参考电压选择

ADC 电路可选择采用外部参考电压输入，分别为参考电压正极性输入和参考电压负极性输入，对应外部参考电压输入脚分别为 VREFP 和 VREFN，可通过 ADCCH 寄存器中的 ADVREFS <1:0>位选择。

采样时间选择

ADC 电路的采样时间可通过 ADCCH 寄存器中的 ADST<1:0>位选择，采样时间有 2 个 Tadclk、4 个 Tadclk、8 个 Tadclk 以及 16 个 Tadclk 四种选项。如果转换信号跳变比较大，建议采样时间设置较长的档位，如 8 个 Tadclk 或者 16 个 Tadclk。

采样模式选择及控制

本芯片 ADC 可选择软件采样和硬件采样两种模式，通过 ADCCL 寄存器中的 SMPS 位选择。选用软件采样时，可通过 ADCCL 寄存器中的 SMPON 位控制采样的启动和停止。

复用端口类型选择

芯片中 ADC 电路的所有模拟输入通道 AINx、参考电压外部输入脚均和 PA/PB 端口复用，在使用 ADC 电路转换前，须先将所使用的管脚通过 ANSL/ANSH 寄存器设置为模拟类型。

模拟信号输入通道选择

ADC 电路使能前，需先选择 A/D 模拟通道。本芯片 ADC 电路支持 8 个外部通道 AIN0~AIN7，2 个内部通道 1/4VDD 和 AVSS。A/D 模拟通道可通过 ADCCL 寄存器中的 ADCHS <3:0>位选择。

对齐方式选择

本芯片 ADC 电路转换的结果支持两种对齐方式，低位对齐和高位对齐，可通过 ADCCH 寄存器中的 ADFM 位进行选择。

6.7.4 ADC转换步骤

下面概述实现 ADC 转换过程的各个步骤，实际应用中还需要考虑 ADC 自身工作的建立（参见寄存器 ADCCL 后面的备注描述内容）。

Step1: 选择 ADC 转换时钟，通过 ADCCH 寄存器中的 ADCKS<2:0>选择 ADC 转换时钟。当选择 VDD 或外部 VREFP 作为正端参考电压时，ADC 转换时钟频率可设置在 32KHz~8MHz 之间。

Step 2: 选择 ADC 参考电压源，通过 ADCCH 寄存器中的 ADVREFS <1:0>位进行选择。

Step 3: 选择 ADC 采样时间，通过 ADCCH 寄存器中的 A/D 采样时间选择位 ADST <1:0>设定。

Step 4: 选择 ADC 采样模式，通过 ADCCL 寄存器中的 A/D 采样模式选择位 SMPS 选择软件采样或硬件采样。

Step 5: 设置复用端口设为模拟类型，即选择哪些管脚作为 ADC 转换输入管脚，由端口数模控制寄存器 ANSL 控制选择。

Step 6: 选择模拟信号输入通道 AINx，通过 ADCCL 寄存器中的 ADCHS <3:0>选择 ADC 模拟通道。

Step7: 设置转换结果对齐方式，通过 ADCCH 寄存器中的 ADFM 位，选择高位对齐放置还是低位对齐放置。

Step 8: 如果要使用中断，则中断控制寄存器需要正确地设置，以确保 A/D 中断功能被正确激活。在默认中断模式下，需将全局中断使能位/高优先级中断使能位 GIE 置“1”，将 ADC 中断使能位置“1”；在向量中断模式下，需将全局中断使能位/高优先级中断使能位 GIE 置“1”，根据 A/D 中断所在组的优先级决定是否要开启低优先级中断使能位 GIEL，将 ADC 中断使能位置“1”。

Step 9: 使能 ADC 电路，将 ADCCL 寄存器中的 ADC 使能位 ADEN 设置为“1”。

Step 11: 当 ADCCL 寄存器中的 SMPS=0 时，选择软件采样模式，设置 ADCCL 寄存器中的 SMPON=1 启动采样，ADCCL 寄存器中的 ADTRG 位硬件自动置 1；当 SMPS=1 时，选择硬件采样模式，将 ADC 转换启动位 ADTRG 位设置为“1”，开始 ADC 转换。

Step 10: 轮询 ADCCL 寄存器中的转换状态位 ADTRG 位，确定此次 A/D 转换是否完成。

Step 11: 读取 ADCRH 和 ADCRL 寄存器中的转换结果。

6.7.5 AD时序特征示意图

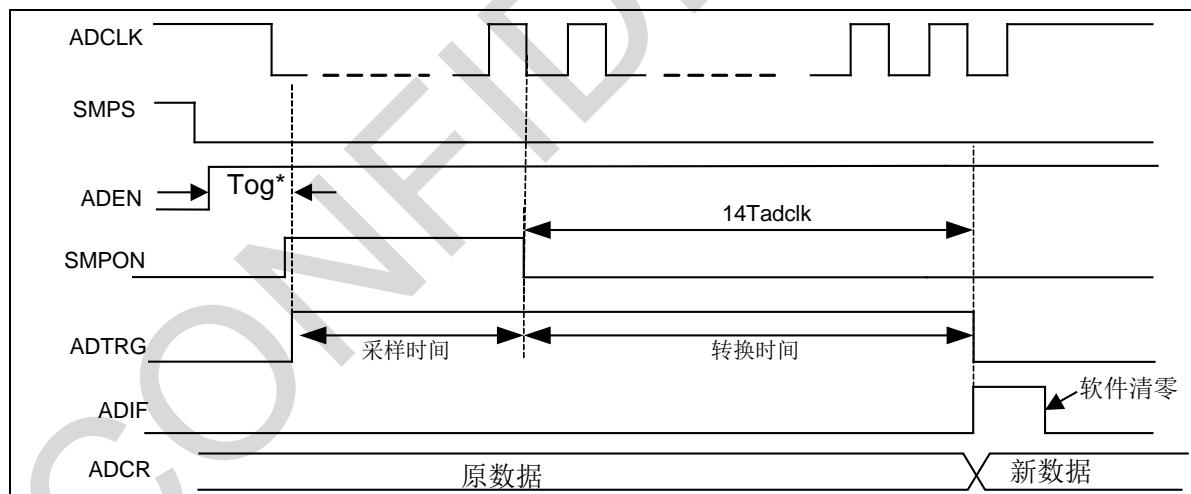


图 6-80 ADC 时序特征示意图 (SMPS=0)

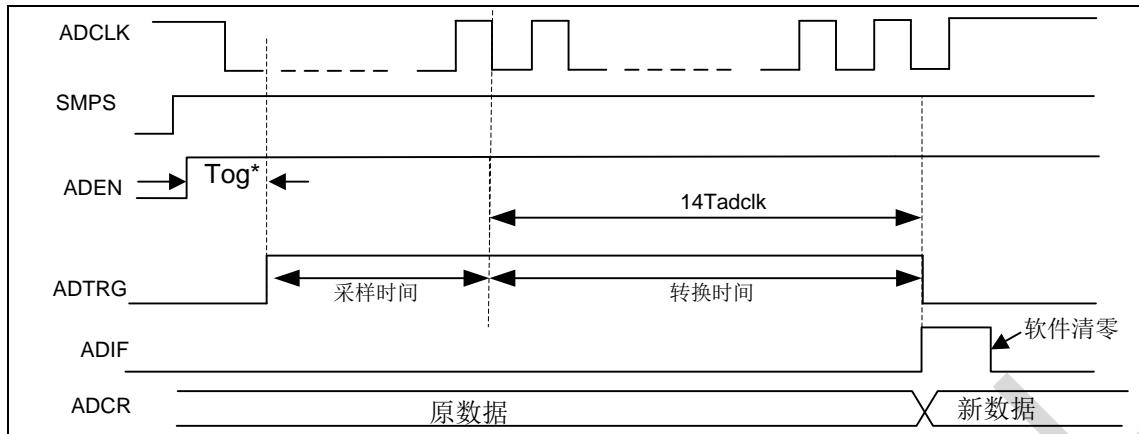


图 6-81 ADC 时序特征示意图 (SMPS=1)

注 1: $Tog > 0$;

注 2: AD 转换时钟周期 $Tadclk$, 可通过 $ADCCKS<2:0>$ 寄存器配置不同的频率。

6.7.6 参考例程

应用例程 1：对模拟输入通道 0 (AIN0) 进行模数转换

```

.....
BSS      ADCCL, ADEN          ;使能 ADC 模块
BCC      ADCCH, ADFM          ; 转换结果高位对齐放置
MOVI    0x05                  ; 硬件控制 ADC 采样模式
MOVA    ADCCL                ; 使能 ADC 转换器, 选中通道 0
BSS      ADCCL, ADTRG         ; 触发 ADC 转换
AD_WAIT:
JBC      ADCCL, ADTRG         ; 等待 ADC 转换完成
GOTO    AD_WAIT
MOV      ADCRH, 0              ; 读取高 8 位转换结果
.....
MOV      ADCRL, 0              ; 读取低 4 位转换结果
.....

```

注：当选择 INTLRC 时钟源时，ADC 中断可以唤醒 IDLE 模式。但在启动 ADC (ADTRG) 和执行 IDLE 指令之间必须保证 2 条指令执行的时间，可以使用 2 条 NOP 指令。

6.7.7 特殊功能寄存器

ADC 功能是由四个控制寄存器和两个数据寄存器控制实现的。其中 ADCRL 和 ADCRH 寄存器用于存储 ADC 转换的数据结果，结果对齐方式由 ADCCH 寄存器中的 ADFM 位控制选择；ADCCL 寄存器用于 ADC 模块的使能控制、ADC 采样模式选择、ADC 转换启动控制以及 ADC 模拟通道选择等；ADCCH 寄存器用于 ADC 采样时间选择、正负参考电压选择、ADC 时钟选择以及结果对齐方式选择等；ANSL 寄存器用于控制复用端口的数模类型。

6.7.7.1 ADC转换值寄存器 (ADCR)

ADFM	ADCRH: ADC 转换值寄存器高 8 位								ADCRL: ADC 转换值寄存器低 8 位							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	—	—	—	—	ADCR<11:8>				ADCR<7:0>							
0	ADCR<11:4>								ADCR<3:0>			—	—	—	—	—

ADCR<11:0>: A/D 转换结果

6.7.7.2 ADC控制寄存器 (ADCCL)

ADCCL: ADC 控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	ADCHS<3:0>				SMPON	SMPS	ADTRG	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	0	1	0	0

Bit 7~4 ADCHS<3:0>: A/D 模拟通道选择位

- 0000: 通道 0 (AIN0)
- 0001: 通道 1 (AIN1)
- 0010: 通道 2 (AIN2)
- 0011: 通道 3 (AIN3)
- 0100: 通道 4 (AIN4)
- 0101: 通道 5 (AIN5)
- 0110: 通道 6 (AIN6)
- 0111: 通道 7 (AIN7)
- 1000: 1/4 VDD
- 1001~1110: VSS
- 1111: 屏蔽通道选择

Bit 3 SMPON: A/D 采样软件控制位

- 0: 结束采样
- 1: 启动采样

Bit 2 SMPS: A/D 采样模式选择位

- 0: 使能软件采样, 硬件采样禁止
- 1: 禁止软件采样, 硬件采样使能

Bit 1 ADTRG: A/D 采样转换状态位

- 0: A/D 未进行转换, 或 A/D 采样转换已完成
- 1: A/D 采样转换正在进行

当 SMPS=1 时, 该位软件置 1 启动 A/D 采样转换; 当 SMPS=0 时, SMPON=1 启动采样, 该位硬件自动置 1。

Bit 0 ADEN: A/D 转换使能位

- 0: 关闭 A/D 转换器
- 1: 运行 A/D 转换器

注 1: 在 ADEN 使能后, ADC 需要先完成自身工作建立, 才能得到正确的转换结果。ADC 电路工作建立方式为, ADEN 使能后, 延时 80us 以上, 启动第一次 ADC 转换 (ADTRG=1), 转换结束后, 再延时 30us 以上, ADC 工作建立完成, 后续启动 ADC 转换, 即可得到正确的转换结果。对应用程序来说, 在 ADEN 使能后, 第一次

ADC 转换前和转换后，分别添加至少 80us 和 30us 延时，同时因 ADC 建立过程中得到的转换结果与理论值偏差极大且不可预知，所以在应用程序中需要丢弃 ADEN 使能后的第一次转换结果。

注 2：因每次 ADEN 重新使能后，均需要执行上述 ADC 工作建立过程，所以应用中，在芯片正常运行时不建议关闭 ADC，保持 ADEN=1，只在进入 IDLE0 休眠模式前，可以关闭 ADC。

6.7.7.3 ADC控制寄存器 (ADCCH)

ADCCH: ADC 控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	ADFM	ADCKS<2:0>			ADST<1:0>		ADVREFS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	0	1	0	0	0

Bit 7 ADFM: A/D 转换数据放置格式选择位

- 0: 高位对齐 (ADCRH<7:0>, ADCRL<7:4>)
- 1: 低位对齐 (ADCRH<3:0>, ADCRL<7:0>)

Bit 6~4 ADCKS<2:0>: A/D 转换时钟频率 (Tadclk) 选择位

- 000: Fosc
- 001: Fosc/2
- 010: Fosc/4
- 011: Fosc/8
- 100: Fosc/16
- 101: Fosc/32
- 110: Fosc/64
- 111: INTLRC (32KHz WDT RC 时钟)

Bit 3~2 ADST<1:0>: A/D 硬件采样时间选择位

- 00: 大约 2 个 Tadclk
- 01: 大约 4 个 Tadclk
- 10: 大约 8 个 Tadclk
- 11: 大约 16 个 Tadclk

Bit 1~0 ADVREFS<1:0>: 参考源选择位

- 00: 参考电压正端为 VDD，负端为 VSS
- 01: 参考电压正端为 VDD，负端为 VREFN
- 10: 参考电压正端为外部 VREFP，负端为 VSS
- 11: 参考电压正端为外部 VREFP，负端为外部 VREFN

注 1：如果在 A/D 转换过程中，进行转换时钟切换，切换后第一次 A/D 转换结果有可能存在误差；

注 2：A/D 转换时钟频率建议选择不要大于 8MHz；

注 3：当使用外部参考电压时，参考电压不能低于 1.3V，否则会导致 ADC 工作异常。

6.7.7.4 端口数模控制寄存器（ANSL）

ANSL: 端口数模控制寄存器								
Bit	7	6	5	4	3	2	1	0
Name	ANSL7	ANSL6	ANSL5	ANSL4	ANSL3	ANSL2	ANSL1	ANSL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 ANSL7: PA7/AIN7 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 6 ANSL6: PB7/AIN6 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 5 ANSL5: PB3/AIN5 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 4 ANSL4: PA5/AIN4 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 3 ANSL3: PA4/AIN3 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 2 ANSL2: PA2/AIN2 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 1 ANSL1: PA1/AIN1 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

Bit 0 ANSL0: PA0/AIN0 端口数模选择位

0: 模拟输入端口

1: 数字输入端口

第7章 中断处理

7.1 概述

中断是芯片的一个重要功能，它能将芯片从睡眠模式中唤醒，也可以使系统在正常运行过程中响应突发事件，中止并保存当前运行程序的信息，跳转到请求中断服务程序的入口地址，执行相对应的中断服务程序，处理突发事件。本芯片支持的中断模式有两种：默认中断模式和向量中断模式，最多可支持 22 个中断源：1 个软件中断和 21 个硬件中断。

7.2 中断控制结构框图

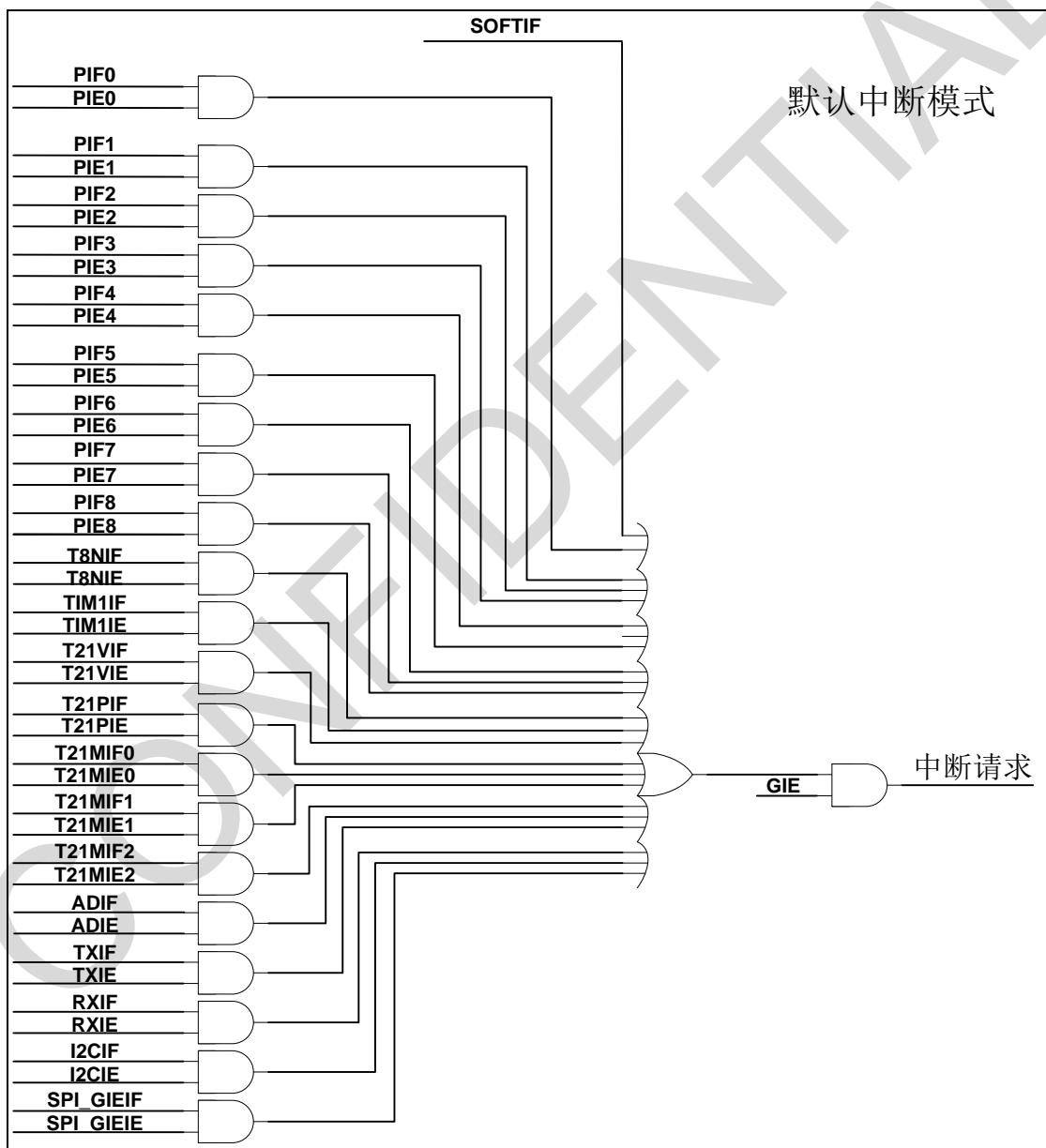


图 7-1 默认中断模式中断控制逻辑

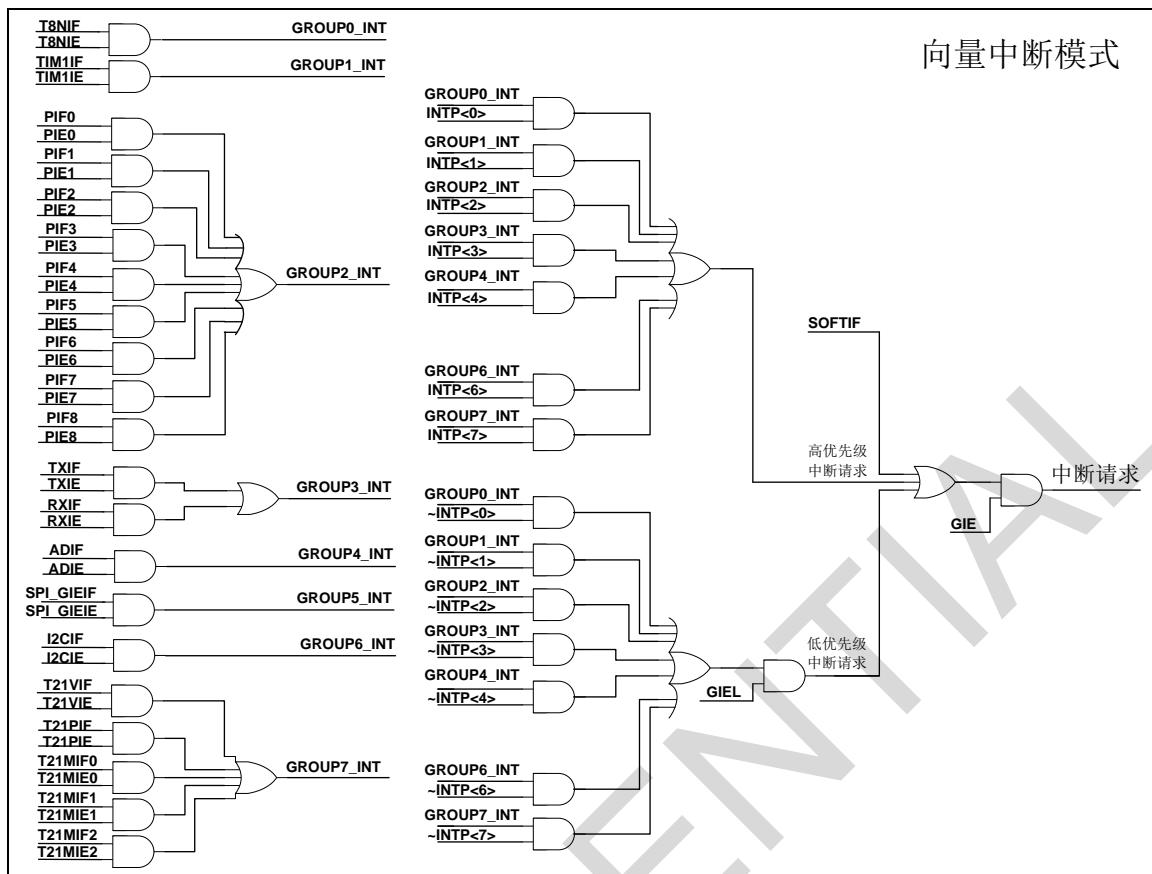


图 7-2 向量中断模式中断控制逻辑

7.3 中断模式选择

芯片支持两种中断模式，默认中断模式和向量中断模式，可通过 INTG 寄存器中的 INTVEN0 位和芯片配置字中的 INTVEN1 位选择。需要注意的是，INTVEN0 和 INTVEN1 只有同时置“1”时，向量中断模式才有效。

INTVEN0 (INTG<2>)	INTVEN1 (芯片配置字 bit 11)	中断模式
0	0	默认中断模式
0	1	
1	0	
1	1	向量中断模式

表 7-1 中断模式选择表

默认中断模式只支持 1 个中断入口，即 0004_H 入口地址，不支持中断优先级和中断嵌套。

向量中断模式支持多个中断入口，且支持中断优先级和中断嵌套。

7.3.1 默认中断模式

当配置为默认中断模式时，所有中断向量的入口地址均位于 0004_H 。用户需通过中断服务程序对各中断标志及中断使能位进行判断，确认引起中断操作的中断源，从而执行相应的中断服务子程序。该模式不支持中断优先级配置。

序号	中断源	中断名	中断标志	中断使能	全局使能	备注
1	软中断	软中断	SOFTIF	—	GIE	SOFTIF 软件置 1
2	外部中断	PINT0	PIF0	PIE0	GIE	—
3		PINT1	PIF1	PIE1	GIE	—
4		PINT2	PIF2	PIE2	GIE	—
5		PINT3	PIF3	PIE3	GIE	—
6		PINT4	PIF4	PIE4	GIE	—
7		PINT5	PIF5	PIE5	GIE	—
8		PINT6	PIF6	PIE6	GIE	—
9		PINT7	PIF7	PIE7	GIE	—
10		PINT8	PIF8	PIE8	GIE	—
11	T8N 定时器/计数器溢出中断	T8NINT	T8NIF	T8NIE	GIE	—
12	高级 TIM1 中断	TIM1INT	TIM1IF	TIM1IE	GIE	—
13	T21 定时器溢出中断	T21VINT	T21VIF	T21VIE	GIE	—
14	T21 周期匹配中断	T21PINT	T21PIF	T21PIE	GIE	—
15	T21 捕捉/比较中断 0	T21MINT0	T21MIF0	T21MIE0	GIE	—
16	T21 捕捉/比较中断 1	T21MINT1	T21MIF1	T21MIE1	GIE	—
17	T21 捕捉/比较中断 2	T21MINT2	T21MIF2	T21MIE2	GIE	—
18	ADC 中断	ADINT	ADIF	ADIE	GIE	—
19	UART TX 中断	TXINT	TXIF	TXIE	GIE	—
20	UART RX 中断	RXINT	RXIF	RXIE	GIE	—
21	I2CS 通讯总中断	I2CINT	I2CIF	I2CIE	GIE	—
22	SPI 通讯总中断	SPIINT	SPI_GIE IF	SPI_GIEI E	GIE	—

表 7-2 默认中断模式使能配置表

7.3.2 向量中断模式

7.3.2.1 向量表配置

当配置为向量中断模式时，各中断源按组划分，每组中断对应一个中断向量入口地址。软中断入口地址为 0004_H ，优先级最高；其它硬件中断分 8 组(IG0~IG7)，通过中断全局寄存器 INTG 中的 INTV<1:0>配置支持不同的向量表优先级排序，并对应 8 个中断入口地址。每组硬件中断可以分别设置高低优先级，响应中断嵌套。通过中断优先级寄存器 INTP 中的 IGPx 配置，将所有硬件中断源分为高低两个优先级仲裁区。根据 INTV<1:0>的设置，对处于该仲裁区内的硬件中断组，进行优先级排序，并响应优先级最高的。高低两个优先级仲裁区分别由高优先级中断使能位 GIE 和低优先级中断使能位 GIEL 来使能。在执行低优先级中断服务程序时，可嵌套响应高优先级中断组。

优先级		0 (高)	1	2	3	4	5	6	7	8 (低)
入口地址		0004_H	0008_H	$000C_H$	0010_H	0014_H	0018_H	$001C_H$	0020_H	0024_H
INTV	00	软中断	IG0	IG1	IG2	IG3	IG4	IG5	IG6	IG7
	01		IG0	IG1	IG6	IG7	IG4	IG5	IG2	IG3

优先级	0 (高)	1	2	3	4	5	6	7	8 (低)
入口地址	0004 _H	0008 _H	000C _H	0010 _H	0014 _H	0018 _H	001C _H	0020 _H	0024 _H
10		IG4	IG5	IG2	IG3	IG0	IG1	IG6	IG7
11		IG7	IG6	IG5	IG4	IG3	IG2	IG1	IG0

表 7-3 向量表配置表

7.3.2.2 中断分组配置

序号	中断组号	高低优先级选择	中断名	备注
1	IG0	IGP0	T8NINT	—
2	IG1	IGP1	TIM1INT	—
3	IG2	IGP2	PINT0	—
4			PINT1	—
5			PINT2	—
6			PINT3	—
7			PINT4	—
8			PINT5	—
9			PINT6	—
10			PINT7	—
11			PINT8	—
12	IG3	IGP3	TXINT	—
13			RXINT	—
14	IG4	IGP4	ADINT	—
15	IG5	IGP5	SPIINT	—
16	IG6	IGP6	I2CINT	—
17	IG7	IGP7	T21VINT	—
18			T21PINT	—
19			T21MINT0	—
20			T21MINT1	—
21			T21MINT2	—

表 7-4 向量中断模式中断分组配置表

7.3.2.3 中断使能配置

序号	中断源	中断名	中断标志	中断使能	IGPx	低优先级中断使能位	全局中断使能位	备注
1	软中断	软中断	SOFTIF	—	—	—	GIE	SOFTIF 软件置 1
2	外部中断	PINT0	PIF0	PIE0	0	GIEL	GIE	—
3					1	—	GIE	—
4		PINT1	PIF1	PIE1	0	GIEL	GIE	—
					1	—	GIE	—
		PINT2	PIF2	PIE2	0	GIEL	GIE	—

序号	中断源	中断名	中断标志	中断使能	IGPx	低优先级中断使能位	全局中断使能位	备注
					1	—	GIE	—
5		PINT3	PIF3	PIE3	0	GIEL	GIE	—
					1	—	GIE	—
6		PINT4	PIF4	PIE4	0	GIEL	GIE	—
					1	—	GIE	—
7		PINT5	PIF5	PIE5	0	GIEL	GIE	—
					1	—	GIE	—
8		PINT6	PIF6	PIE6	0	GIEL	GIE	—
					1	—	GIE	—
9		PINT7	PIF7	PIE7	0	GIEL	GIE	—
					1	—	GIE	—
10		PINT8	PIF8	PIE8	0	GIEL	GIE	—
					1	—	GIE	—
11	T8N 定时器/计数器溢出中断	T8NINT	T8NIF	T8NIE	0	GIEL	GIE	—
					1	—	GIE	—
12	TIM1 定时器溢出中断	TIM1OVIN T	TIM1OVI F	TIM1OVI E	0	GIEL	GIE	—
					1	—	GIE	—
13	T21 定时器溢出中断	T21VINT	T21VIF	T21VIE	0	GIEL	GIE	—
					1	—	GIE	—
14	T21 周期溢出中断	T21PINT	T21PIF	T21PIE	0	GIEL	GIE	—
					1	—	GIE	—
15	T21 捕捉/比较中断 0	T21MINT0	T21MIF0	T21MIE0	0	GIEL	GIE	—
					1	—	GIE	—
16	T21 捕捉/比较中断 1	T21MINT1	T21MIF1	T21MIE1	0	GIEL	GIE	—
					1	—	GIE	—
17	T21 捕捉/比较中断 2	T21MINT2	T21MIF2	T21MIE1	0	GIEL	GIE	—
					1	—	GIE	—
18	ADC 中断	ADINT	ADIF	ADIE	0	GIEL	GIE	—
					1	—	GIE	—
19	UART TX 中断	TXINT	TXIF	TXIE	0	GIEL	GIE	—
					1	—	GIE	—
20	UART RX 中断	RXINT	RXIF	RXIE	0	GIEL	GIE	—
					1	—	GIE	—
21	I2CS 通讯总中断	I2CINT	I2CIF	I2CIE	0	GIEL	GIE	—
					1	—	GIE	—
22	SPI 通讯总中断	SPIINT	SPI_GIEI F	SPI_GIEI E	0	GIEL	GIE	—
					1	—	GIE	—

表 7-5 向量中断模式使能配置表

7.4 中断现场保护

中断现场保护是中断服务程序中一个很重要的组成部分。

指令系统中有 PUSH（压栈）和 POP（出栈）指令，可以方便的实现当前工作状态的保存和恢复。A、PSW、PCRH 和 BKSR 寄存器，分别有各自的两级镜像寄存器 AS1、PSWS1、PCRHS1、BKSRS1 和 AS0、PSWS0、PCRHS0、BKSRS0，用于对相应寄存器的保存和恢复。镜像寄存器无物理地址，他们只能通过 PUSH 和 POP 指令自动完成相应的保存与恢复动作，两级镜像寄存器采用堆栈的操作方式。

7.5 中断操作

若中断事件条件产生，相关中断标志将被置为“1”。中断标志产生后程序要跳转至相应的服务程序地址执行，需满足以下条件：

- 1) 当对应中断使能位为“1”时，继续判断第二个条件是否满足；当对应中断使能位为“0”时，即使中断标志为“1”，中断也不会发生，程序也不会跳转至中断服务程序地址执行。
- 2) 在默认中断模式下，当全局中断使能位 GIE 为“0”时，将屏蔽所有中断请求。当全局中断使能位 GIE 为“1”时，程序将跳至中断服务程序地址执行。在向量中断模式下，当全局中断使能位 GIE 为“0”时，将屏蔽所有中断请求。当全局中断使能位 GIE 为“1”时，若对应中断所在组为高优先级，程序将跳至中断服务程序地址执行；若对应中断所在组为低优先级，当低优先级中断使能位 GI_EL 为“1”时，在无高优先级中断请求时，程序将跳至中断服务地址执行，当低优先级中断使能位 GI_EL 为“0”时，将屏蔽所有低优先级中断请求。

为确保对寄存器 GIE 和 GI_EL 的软件写操作成功，需按如下步骤进行：

- 1) 在默认中断模式或向量中断模式下，对 GIE 位的软件清 0 操作，需先关闭所有外设中断使能，再将 GIE 位清 0；或在 GIE 位清 0 操作后，查询 GIE 位是否为 0，不为 0 则继续执行清 0 操作，直到成功为止；
对 GIE 位的软件置 1 操作，无特殊要求，推荐先打开所需的外设中断使能，再将 GIE 位置 1。
- 2) 在向量中断模式下，对 GI_EL 位的软件清 0 操作，需先关闭所有外设中断使能，再将 GI_EL 位清 0；或在 GI_EL 位清 0 操作后，查询 GI_EL 位是否为 0，不为 0 则继续执行清 0 操作，直到成功为止；
对 GI_EL 位的软件置 1 操作，需与 GIE 位同时置 1，或先将 GI_EL 位置 1，再将 GIE 位置 1。

7.5.1 外部中断

当 PINTx 复用端口被配置为数字输入端口，且输入信号变化满足触发条件时，将产生 PINTx 外部端口中断，相应的中断标志 PI_Fx 被置“1”。当全局中断控制位 GIE 和外部端口中断控制位 PI_Ex 都被置为“1”时，则向 CPU 发出 PINTx 外部端口中断请求。当中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。

需要注意的是，相应中断标志位 PI_Fx 和中断使能位 PI_Ex 都需通过软件清除，INTC0 寄存器用于配置触发条件，可分别配置为上升沿触发、下降沿触发或双沿触发。

7.5.2 ADC中断

ADC 中断由 ADC 转换动作控制，当 ADC 转换完成时，将产生 ADC 中断，ADC 中断标志位 ADIF 被置“1”。当 ADC 中断控制位 ADIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 ADC 中断请求。CPU 根据中断的优先级响应当前中断的请求，当 ADC 中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，ADC 中断标志位 ADIF 和中断使能位 ADIE 都需通过软件清除。

7.5.3 T8N溢出中断

8 位定时器/计数器 T8N 处于定时器模式或计数器模式，当 T8N 计数器递增计数由 FF_H 变为 00_H 时，T8N 计数器发生溢出，将中断标志 T8NIF 位置“1”。当 T8N 溢出中断使能位 T8NIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 T8N 溢出中断请求。CPU 根据中断的优先级响应当前中断的请求，当 T8N 溢出中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，T8N 溢出中断标志位 T8NIF 和中断使能位 T8NIE 都需通过软件清除。

7.5.4 T2x(T21)匹配中断

16 位定时器 T2x 各种工作模式都可产生匹配中断：

定时器模式/PWM 模式/捕捉模式/比较模式

16 位定时器 T2x 处于定时器模式/PWM 模式/捕捉模式/比较模式。

定时匹配中断

对计数时钟进行递增计数，当 T2x 后分频器的计数值与后分频器分频比相同时，产生匹配中断。

捕捉模式

当输入信号的变化状态满足捕捉条件时，将 T2x 计数器的值捕捉到寄存器(T2xRH:T2xRL)中，并产生捕捉中断。

T2x 匹配中断产生时，将中断标志 T2xIF 位置“1”。当 T2x 匹配中断使能位 T2xIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 T2x 定时中断请求。CPU 根据中断的优先级响应当前中断的请求，当 T2x 匹配中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，T2x 匹配中断标志位 T2xIF 和中断使能位 T2xIE 都需通过软件清除。

7.5.5 T2x(T21)周期中断

16 位定时器 T2x 处于 PWM 模式，T2x 从零开始递增计数，当 T2x 与 T2xP 寄存器的值相等时，将产生 T2x 周期中断，中断标志 T2xPIF 被置“1”。如果中断使能位 T2xPIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 T2x 周期中断请求。CPU 根据中断的优先级响应当前中断的请求，当 T2x 周期中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，T2x 周期中断标志位 T2xPIF 和中断使能位 T2xPIE 都需通过软件清除。

7.5.6 UART中断

UART 中断包括两种：发送中断和接收中断。

当 UART 异步发送器的发送数据寄存器 TXB 为空，或异步接收器完成一个数据接收时，产生 UART 发送/接收中断，发送/接收中断标志位 RXIF/TXIF 被置为“1”。如果发送/接收中断使能位 RXIE/TXIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 UART 发送/接收中断请求。CPU 根据中断的优先级响应当前中断的请求，当 UART 发送/接收条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，接收/发送中断标志位 RXIF/TXIF 为只读，不可软件清零，读接收数据寄存器 RXB，可清除 RXIF，写发送数据寄存器 TXB，可清除 TXIF；接收/发送中断使能位 RXIE/TXIE 需通过软件清除。

7.5.7 I2CS中断

当 I2CSRIF、I2CSPIF、I2CTBIF、I2CRBIF、I2CTEIF、I2CROIF 和 I2CNAIF 中任何一个中断标志位置 1 时，I2C 总中断标志位 I2CIF 就会置 1。如果 I2C 中断使能位 I2CIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 I2C 中断请求。CPU 根据中断的优先级响应当前中断的请求，当 I2C 中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，需要软件清零 I2CIF，但在清零 I2CIF 总中断标志位之前，需先清零 I2CSRIF、I2CSPIF、I2CTEIF、I2CROIF 和 I2CNAIF 等相关中断标志位。

7.5.8 SPI中断

当 TBWEIF、NSSIF、IDIF、ROIF、TEIF、RBIF 和 TBIF 中任何一个中断标志位置 1 时，I2C 总中断标志位 SPI_GIEIF 就会置 1。如果 SPI 中断使能位 SPI_GIEIE 置为“1”，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 SPI 中断请求。CPU 根据中断的优先级响应当前中断的请求，当 SPI 中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，需要软件清零 SPI_GIEIF，但在清零 SPI_GIEIF 总中断标志位之前，需先清零 TBWEIF、NSSIF、IDIF、ROIF、TEIF、RBIF 和 TBIF 等相关中断标志位。

7.5.9 ADTIM1 中断

当 BIF、TIF、COMIF、CC4IF、CC3IF、CC2IF、CC1IF、UIF、CC4OIF、CC3OIF、CC2OIF、CC1OIF 中任何一个中断标志位置 1 时，ADTIM1 总中断标志位 TIM1IF 就会置 1。如果相应的中断使能位 BIE、TIE、COMIE、CC4IE、CC3IE、CC2IE、CC1IE、UIE、CC4OIE、CC3OIE、CC2OIE、CC1OIE 置 1，ADTIM1 总中断使能位置 1，且全局中断控制位 GIE 和低优先级中断使能位 GIEL 根据中断模式正确使能时，则向 CPU 发出 ADTIM1 中断请求。CPU 根据中断的优先级响应当前中断的请求，当 ADTIM1 中断条件允许时，系统将进入相应中断的服务程序入口地址，进行中断程序处理。需要注意的是，需要软件清零 TIM1IF，但在清零 TIM1IF 总中断标志位之前，需先清零 BIF、TIF、COMIF、CC4IF、CC3IF、CC2IF、CC1IF、UIF、CC4OIF、CC3OIF、CC2OIF、CC1OIF 等相关中断标志位。

7.5.10 中断操作注意事项

用户在使能中断前需先清除相应的中断标志，避免中断的误触发。

除只读的中断标志（由硬件清除）外，其余的中断标志必须通过软件清除。

为避免中断的发生与中断标志清除操作冲突时，清除中断标志不成功，建议用户在进行中断标志清除操作后，对中断标志清除成功与否进行软件判断。如果操作不成功则再次进行中断标志清除操作，直到中断标志清除成功为止。用户也可以连续执行两次中断标志清除操作达到相同目的。

7.6 特殊功能寄存器

中断功能由一系列的控制寄存器和芯片配置字共同控制。

7.6.1 中断全局寄存器 (INTG)

INTG: 中断全局寄存器								
Bit	7	6	5	4	3	2	1	0
Name	GIE	GIEL	—	—	SOFTIF	INTVEN	INTV<1:0>	
R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 GIE: 全局中断使能位，或高优先级中断使能位
0: 禁止所有的中断，或禁止高优先级中断
1: 使能所有未屏蔽的中断，或使能高优先级中断
- Bit 6 GIEL: 低优先级中断使能位（向量中断模式）
0: 禁止低优先级中断
1: 使能低优先级中断
- Bit 5~4 未使用
- Bit 3 SOFTIF: 软中断标志位
0: 无软中断
1: 有软中断
- Bit 2 INTVEN: 中断模式选择位
0: 默认中断模式
1: 向量中断模式（芯片配置字 INTVEN1 必须为 1）
- Bit 1~0 INTV<1:0>: 中断向量表选择位，参考向量表配置

7.6.2 中断优先级寄存器 (INTP)

INTP: 中断优先级寄存器								
Bit	7	6	5	4	3	2	1	0
Name	IGP<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~0 IGP<7:0>: IG7-IG0 中断优先级设置
0: 低优先级
1: 高优先级

7.6.3 中断控制寄存器 0 (INTC0)

INTC0: 中断控制寄存器 0								
Bit	7	6	5	4	3	2	1	0
Name	PEG3<1:0>		PEG2<1:0>		PEG1<1:0>		PEG0<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 PEG3<1:0>: PINT8~PINT6 触发边沿选择位

00: PINT8~PINT6 下降沿触发

01: PINT8~PINT6 上升沿触发

1x: PINT8~PINT6 双沿触发

Bit 5~4 PEG2<1:0>: PINT5~PINT4 触发边沿选择位

00: PINT5~PINT4 下降沿触发

01: PINT5~PINT4 上升沿触发

1x: PINT5~PINT4 双沿触发

Bit 3~2 PEG1<1:0>: PINT3~PINT2 触发边沿选择位

00: PINT3~PINT2 下降沿触发

01: PINT3~PINT2 上升沿触发

1x: PINT3~PINT2 双沿触发

Bit 1~0 PEG0<1:0>: PINT1~PINT0 触发边沿选择位

00: PINT1~PINT0 下降沿触发

01: PINT1~PINT0 上升沿触发

1x: PINT1~PINT0 双沿触发

7.6.4 中断标志寄存器 0 (INTF0)

INTF0: 中断标志寄存器 0								
Bit	7	6	5	4	3	2	1	0
Name	SPI_GIEIF	—	—	—	PIF8	TIM1IF	T8NIF	ADIF
R/W	R/W	—	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 SPI_GIEIF: SPI 模块总中断标志位

0: 未发生 SPI 中断

1: 发生 SPI 中断 (必须用软件清零)

Bit 6~4 保留未用

Bit 3 PIF8: 外部端口中断 8 标志位

0: 外部端口 PINT8 上无中断信号

1: 外部端口 PINT8 上有中断信号 (必须用软件清零)

Bit 2 TIM1IF: TTIM1 中断标志位

0: TIM1 未发生匹配

1: TIM1 发生中断 (必须软件清零)

Bit 1 T8NIF: T8N 溢出中断标志位

0: T8N 计数未溢出

1: T8N 计数溢出 (必须用软件清零)

- Bit 0 ADIF: ADC 中断标志位
0: 正在进行 AD 转换
1: AD 转换已经完成 (必须用软件清零)

7.6.5 中断使能寄存器 0 (INTE0)

INTE0: 中断使能寄存器 0								
Bit	7	6	5	4	3	2	1	0
Name	SPI_GIEIE	—	—	—	PIE8	TIM1IE	T8NIE	ADIE
R/W	R/W	—	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 SPI_GIEIE: SPI 模块总中断使能位

0: 禁止

1: 使能

Bit 6~4 保留未用

Bit 3 PIE8: 外部端口中断 8 使能位

0: 禁止

1: 使能

Bit 2 TIM1IF: TTIM1 中断使能位

0: 禁止

1: 使能

Bit 1 T8NIE: T8N 溢出中断使能位

0: 禁止

1: 使能

Bit 0 ADIE: ADC 中断使能位

0: 禁止

1: 使能

7.6.6 中断标志寄存器 1 (INTF1)

INTF1: 中断标志寄存器 1								
Bit	7	6	5	4	3	2	1	0
Name	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
R/W	R/W							
POR	0	0	0	0	0	0	0	0

Bit 7 PIF7: 外部端口中断 7 标志位

0: 外部端口 PINT7 上无中断信号

1: 外部端口 PINT7 上有中断信号 (必须用软件清零)

Bit 6 PIF6: 外部端口中断 6 标志位

0: 外部端口 PINT6 上无中断信号

1: 外部端口 PINT6 上有中断信号 (必须用软件清零)

Bit 5 PIF5: 外部端口中断 5 标志位

0: 外部端口 PINT5 上无中断信号

1: 外部端口 PINT5 上有中断信号 (必须用软件清零)

Bit 4	PIF4: 外部端口中断 4 标志位 0: 外部端口 PINT4 上无中断信号 1: 外部端口 PINT4 上有中断信号 (必须用软件清零)
Bit 3	PIF3: 外部端口中断 3 标志位 0: 外部端口 PINT3 上无中断信号 1: 外部端口 PINT3 上有中断信号 (必须用软件清零)
Bit 2	PIF2: 外部端口中断 2 标志位 0: 外部端口 PINT2 上无中断信号 1: 外部端口 PINT2 上有中断信号 (必须用软件清零)
Bit 1	PIF1: 外部端口中断 1 标志位 0: 外部端口 PINT1 上无中断信号 1: 外部端口 PINT1 上有中断信号 (必须用软件清零)
Bit 0	PIFO: 外部端口中断 0 标志位 0: 外部端口 PINT0 上无中断信号 1: 外部端口 PINT0 上有中断信号 (必须用软件清零)

7.6.7 中断使能寄存器 1 (INTE1)

INTE1: 中断使能寄存器 1								
Bit	7	6	5	4	3	2	1	0
Name	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0
Bit 7	PIE7: 外部端口中断 7 使能位 0: 禁止 1: 使能							
Bit 6	PIE6: 外部端口中断 6 使能位 0: 禁止 1: 使能							
Bit 5	PIE5: 外部端口中断 5 使能位 0: 禁止 1: 使能							
Bit 4	PIE4: 外部端口中断 4 使能位 0: 禁止 1: 使能							
Bit 3	PIE3: 外部端口中断 3 使能位 0: 禁止 1: 使能							
Bit 2	PIE2: 外部端口中断 2 使能位 0: 禁止 1: 使能							
Bit 1	PIE1: 外部端口中断 1 使能位 0: 禁止 1: 使能							

Bit 7	PIE7: 外部端口中断 7 使能位 0: 禁止 1: 使能
Bit 6	PIE6: 外部端口中断 6 使能位 0: 禁止 1: 使能
Bit 5	PIE5: 外部端口中断 5 使能位 0: 禁止 1: 使能
Bit 4	PIE4: 外部端口中断 4 使能位 0: 禁止 1: 使能
Bit 3	PIE3: 外部端口中断 3 使能位 0: 禁止 1: 使能
Bit 2	PIE2: 外部端口中断 2 使能位 0: 禁止 1: 使能
Bit 1	PIE1: 外部端口中断 1 使能位 0: 禁止 1: 使能

Bit 0 PIE0: 外部端口中断 0 使能位
0: 禁止
1: 使能

7.6.8 中断标志寄存器 2 (INTF2)

INTF2: 中断标志寄存器 2								
Bit	7	6	5	4	3	2	1	0
Name	T21MIF2	I2CIF	T21MIF1	T21MIF0	T21PIF	T21VIF	RXIF	TXIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
POR	0	0	0	0	0	0	0	0

Bit 7 T21MIF2: T21 捕捉/比较中断 2 标志位
0: 无中断请求
1: T21 多功能中断 2 请求

Bit 6 I2CIF: I2CS 通讯总中断标志位
0: 未发生通讯中断
1: 发生通讯中断

Bit 5 T21MIF1: T21 捕捉/比较中断 1 标志位
0: 无中断请求
1: T21 多功能中断 1 请求

Bit 4 T21MIF0: T21 捕捉/比较中断 0 标志位
0: 无中断请求
1: T21 多功能中断 0 请求

Bit 3 T21PIF: T21 周期中断标志位
0: 无中断请求
1: T21 周期中断请求

Bit 2 T21VIF: T21 溢出中断标志位
0: 无中断请求
1: T21 溢出中断请求

Bit 1 RXIF: UART 接收中断标志位
0: 接收缓冲区空 (接收未完成)
1: 接收缓冲区满 (接收完成), 读 RXB 清零

Bit 0 TXIF: UART 发送中断标志位
0: 发送缓冲区满 (发送未完成)
1: 发送缓冲区空 (发送完成), 写 TXB 清零

7.6.9 中断使能寄存器 2 (INTE2)

INTE2: 中断使能寄存器 2								
Bit	7	6	5	4	3	2	1	0
Name	T21MIE2	I2CIE	T21MIE1	T21MIE0	T21PIE	T21VIE	RXIE	TXIE
R/W	R/W							
POR	0	0	0	0	0	0	0	0

Bit 7 T21MIE2: T21 捕捉/比较中断 2 使能位

	0: 禁止 1: 使能
Bit 6	I2CIE: I2CS 通讯总中断使能位 0: 禁止 1: 使能
Bit 5	T21MIE1: T21 捕捉/比较中断 1 使能位 0: 禁止 1: 使能
Bit 4	T21MIE0: T21 捕捉/比较中断 0 使能位 0: 禁止 1: 使能
Bit 3	T21PIE: T21 周期中断使能位 0: 禁止 1: 使能
Bit 2	T21VIE: T21 溢出中断使能位 0: 禁止 1: 使能
Bit 1	RXIE: UART 接收中断使能位 0: 禁止 1: 使能
Bit 0	TXIE: UART 发送中断使能位 0: 禁止 1: 使能

第8章 芯片配置字

寄存器名称	芯片配置字 (CFG_WD0)	
地址	8001 _H	
OSCS <2:0>	bit2-0	<p>振荡器选择位</p> <p>000: 001: 保留未用 010: 保留未用 011: INTOSCIO 2MHz 模式, PB3 为 I/O 管脚 100: INTOSCIO 4MHz 模式, PB3 为 I/O 管脚 101: INTOSCIO 8MHz 模式, PB3 为 I/O 管脚 110: INTOSC 16MHz 模式, PB3 管脚功能为 CLKO, 主系统时钟为 INTHRC^{注2} 111: INTOSCIO 16MHz 模式, PB3 为 I/O 管脚, 主系统时钟为 INTHRC</p>
WDTEN	bit3	<p>硬件看门狗使能位</p> <p>0: 禁止 1: 使能</p>
PWRTEB	bit4	<p>上电/低电压定时器使能位</p> <p>当 PA3 管脚用于外部复位时</p> <p>0: 使能 1: 禁止</p> <p>当 PA3 管脚用于数字输入输出时, 固定为使能</p>
MRSTEN	bit5	<p>MRSTN 管脚功能选择位</p> <p>0: 管脚用于数字输入输出 1: 管脚用于外部复位</p>
BORVS	bit7-6	<p>低电压选择位</p> <p>00: 3.1V 01: 2.5V 10: 2.1V 11: 保留未用</p>
—	Bit9-8	固定为全 1
ICDEN	bit10	<p>ICD 调试模式使能位</p> <p>0: 使能 1: 禁止</p>
INTVEN1	bit11	<p>中断模式选择位</p> <p>0: 默认中断模式 1: 向量中断模式 (控制寄存器位 INTVEN0 也必须为 1)</p>
—	bit12	固定为 0
FREN	Bit13	<p>FLASH 程序存储区查表指令读使能位</p> <p>0: 禁止 1: 使能</p>
—	Bit15-14	固定为全 1

注 1：芯片配置字通过编程界面配置；

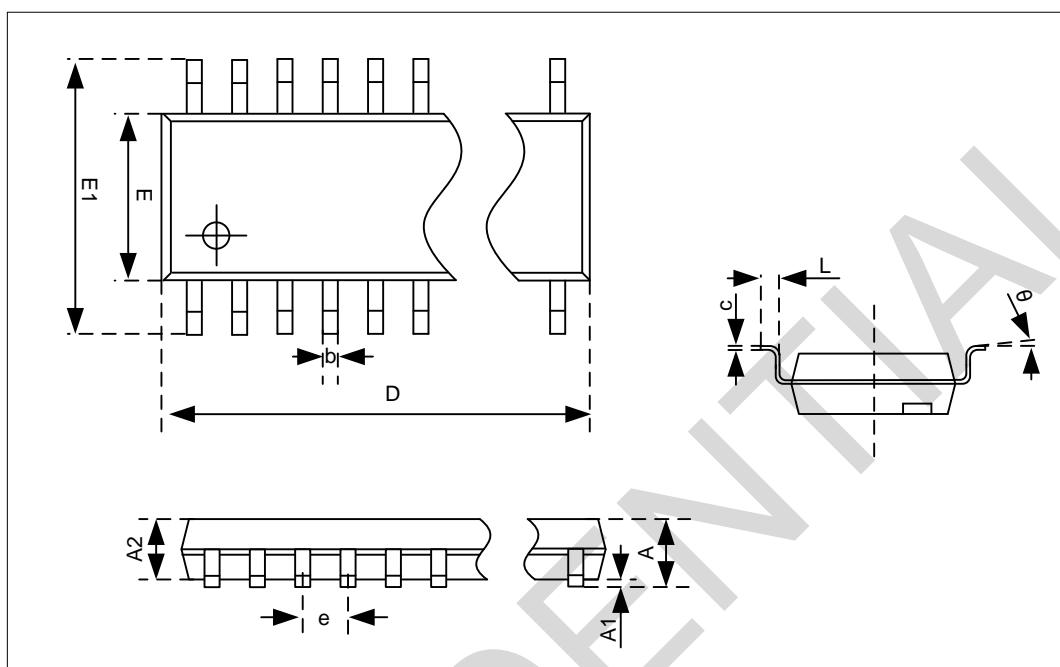
注 2：CLKO 为系统时钟 16 分频输出；

CONFIDENTIAL

第9章 芯片封装图

9.1 20-pin 封装图

TSSOP20



标号	公制 (mm)		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	—	1.00
b	0.19	—	0.30
c	0.09	—	0.20
D	6.40	—	6.60
E	4.30	—	4.50
E1	6.25	—	6.55
e	0.65BSC		
L	0.50	—	0.70
θ	1°	—	7°

附录1 指令集

附录1.1 概述

本芯片提供了 79 条精简指令。

汇编指令为了方便程序设计者使用，指令名称大多是由指令功能的英文缩写所组成的。这些指令所组成的程序经过编译器的编译与连接后，会被转换为相对应的指令码。转换后的指令码可以分为操作码（OP Code）与操作数（Operand）两个部分。操作码部分对应到指令本身。

芯片运行在 4MHz 振荡时钟时，一个机器周期的时间为 500ns。

按照指令执行的机器周期数可将指令分为双周期指令和单周期指令，其中 JUMP、AJMP、GOTO、CALL、LCALL、RCALL、RET、RETIA、RETIE、TBR、TBR#1、TBR_1、TBR1#、TBW、TBW#1、TBW_1、TBW1#为双周期指令；满足跳转条件时，JBC、JBS、JCAIE、JCAIG、JCAIL、JCRAE、JCRAAG、JCRAL、JCCRE、JCCRG、JCCRL、JDEC、JINC 指令为双周期指令，否则为单周期指令；其它指令为单周期指令。

附录1.2 寄存器操作指令

序号	指令		影响状态位	机器周期	操作
1	SECTION	I<7:0>	—	1	I<7:0>->BKSRI<7:0>
2	PAGE	I<8:0>	—	1	I<1:0>->PCRHI<4:3>
3	ISTEP	I<7:0>	—	1	IAA+i->IAA(-128≤i≤127)
4	MOVI	I<7:0>	—	1	I<7:0>->(A)
5	MOV	R<7:0>,F	Z,N	1	(R)->(目标)
6	MOVA	R<7:0>	—	1	(A)->(R)
7	MOVAR	R<10:0>	—	1	(A)->(R) (R 为 GPR)
8	MOVRA	R<10:0>	—	1	(R)->(A) (R 为 GPR)

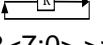
附录1.3 程序控制指令

序号	指令		影响状态位	机器周期	操作
9	JUMP	I<7:0>	—	2	PC+1+i<7:0>->PC (-128≤i≤127)
10	AJMP	I<19:0>	—	2	I<12:0>->PC<12:0> I<12:8>->PCRH<4:0>
11	GOTO	I<10:0>	—	2	I<10:0>->PC<10:0>, PCRH<4:3>->PC<12:11>
12	CALL	I<10:0>	—	2	PC+1->TOS,I<10:0>->PC<10:0>, PCRH<4:3>->PC<12:11>
13	LCALL	I<19:0>	—	2	PC+1->TOS,I<12:0>->PC<12:0>

序号	指令		影响状态位	机器周期	操作
					I<12:8>->PCRH<4:0>
14	RCALL	R<7:0>	—	2	PC+1→TOS, (R)→PC<7:0>, PCRH<4:0>->PC<12:8>,
15	JBC	R<7:0>,B<2:0>	—	2 或 1	当 R = 0 时跳过下一条指令
16	JBS	R<7:0>,B<2:0>	—	2 或 1	当 R = 1 时跳过下一条指令
17	JCAIE	I<7:0>	—	2 或 1	当(A) = I 时跳过下一条指令
18	JCAIG	I<7:0>	—	2 或 1	当(A) > I 时跳过下一条指令
19	JCAIL	I<7:0>	—	2 或 1	当(A) < I 时跳过下一条指令
20	JCRAE	R<7:0>	—	2 或 1	当(R) = (A) 时跳过下一条指令
21	JCRAG	R<7:0>	—	2 或 1	当(R) > (A) 时跳过下一条指令
22	JCRAL	R<7:0>	—	2 或 1	当(R) < (A) 时跳过下一条指令
23	JCCRE	R<7:0>,B<2:0>	—	2 或 1	当 C = R(B) 时跳过下一条指令
24	JCCRG	R<7:0>,B<2:0>	—	2 或 1	当 C > R(B) 时跳过下一条指令
25	JCCRL	R<7:0>,B<2:0>	—	2 或 1	当 C < R(B) 时跳过下一条指令
26	JDEC	R<7:0>,F	—	2 或 1	(R-1)->(目标寄存器), 当目标寄存器的值为 0 时则跳过下一条指令
27	JINC	R<7:0>,F	—	2 或 1	(R+1)->(目标寄存器), 当目标寄存器的值为 0 时则跳过下一条指令
28	NOP	—	—	1	空操作
29	POP	—	—	1	AS->A,PSWS->PSW, BKSR->BKSR,PCRHS->PCRH
30	PUSH	—	—	1	A->AS,PSW->PSWS, BKSR->BKSR,PCRH->PCRHS
31	RET	—	—	2	TOS->PC
32	RETIA	I<7:0>	—	2	I->(A),TOS->PC
33	RETIE	—	—	2	TOS->PC,1->GIE
34	RST	—	全部状态位均被影响	1	软件复位指令
35	CWDT	—	N_TO,N_PD	1	00 _H ->WDT,0->WDTPrescaler, 1-> N_TO, 1-> N_PD
36	IDLE	—	N_TO,N_PD	1	00 _H ->WDT,0->WDTPrescaler, 1-> N_TO, 0-> N_PD

附录1.4 算术/逻辑运算指令

序号	指令		影响状态位	机器周期	操作
37	ADD	R<7:0>,F	C,DC,Z,OV,N	1	(R)+(A)->(目标)
38	ADDC	R<7:0>,F	C,DC,Z,OV,N	1	(R)+(A)+C->(目标)
39	ADDCI	I<7:0>	C,DC,Z,OV,N	1	I+(A)+C->(A)
40	ADDI	I<7:0>	C,DC,Z,OV,N	1	I+(A)->(A)
41	AND	R<7:0>,F	Z,N	1	(A).AND.(R)->(目标)

序号	指令	影响状态位	机器周期	操作
42	ANDI	I<7:0>	Z,N	1 I.AND.(A)->(A)
43	BCC	R<7:0>,B<2:0>	—	1 0->R
44	BSS	R<7:0>,B<2:0>	—	1 1->R
45	BTT	R<7:0>,B<2:0>	—	1 (~R)->R
46	CLR	R<7:0>	Z	1 (R)=0
47	SETR	R<7:0>	—	1 FF _H ->(R)
48	NEG	R<7:0>	C,DC,Z,OV,N	1 ~(R)+1->(R)
49	COM	R<7:0>,F	Z,N	1 (~R)->(目标)
50	DAR	R<7:0>,F	C	1 对(R)十进制调整->(目标)
51	DAA	—	C	1 对(A)十进制调整->(A)
52	DEC	R<7:0>,F	C,DC,Z,OV,N	1 (R-1)->(目标)
53	INC	R<7:0>,F	C,DC,Z,OV,N	1 (R+1)->(目标)
54	IOR	R<7:0>,F	Z,N	1 (A).OR.(R)->(目标)
55	IORI	I<7:0>	Z,N	1 I.OR.(A)->(A)
56	RLB	R<7:0>,F,B<2:0>	C,Z,N	1  C<< R<7:0> (R 带 C 向左循环移位)
57	RLBNC	R<7:0>,F,B<2:0>	Z,N	1  R<7><< R<7:0> (R 不带 C 向左循环移位)
58	RRB	R<7:0>,F,B<2:0>	C,Z,N	1  C>> R<7:0> (R 带 C 向右循环移位)
59	RRBNC	R<7:0>,F,B<2:0>	Z,N	1  R<7:0>>> R<0> (R 不带 C 向右循环移位)
60	SUB	R<7:0>,F	C,DC,Z,OV,N	1 (R)-(A)->(目标)
61	SUBC	R<7:0>,F	C,DC,Z,OV,N	1 (R)-(A)- (~C)->(目标)
62	SUBCI	I<7:0>	C,DC,Z,OV,N	1 I-(A)- (~C)->(A)
63	SUBI	I<7:0>	C,DC,Z,OV,N	1 I-(A)->(A)
64	SSUB	R<7:0>,F	C,DC,Z,OV,N	1 (A)-(R)->(目标)
65	SSUBC	R<7:0>,F	C,DC,Z,OV,N	1 (A)-(R)- (~C)->(目标)
66	SSUBCI	I<7:0>	C,DC,Z,OV,N	1 (A)-I- (~C)->(A)
67	SSUBI	I<7:0>	C,DC,Z,OV,N	1 (A)-I->(A)
68	SWAP	R<7:0>,F	—	1 R<3:0>->(目标)<7:4>, R<7:4>->(目标)<3:0>
69	TBR	—	—	2 Pmem(FRA)->ROMD

序号	指令	影响状态位	机器周期	操作
70	TBR#1	—	—	2 Pmem(FRA)-> ROMD, FRA+1->FRA
71	TBR_1	—	—	2 Pmem(FRA)-> ROMD, FRA-1->FRA
72	TBR1#	—	—	2 FRA+1->FRA, Pmem(FRA)-> ROMD
73	TBW	—	—	2 ROMD->prog buffer
74	TBW#1	—	—	2 ROMD>prog buffer, FRA+1->FRA
75	TBW_1	—	—	2 ROMD->prog buffer, FRA-1->FRA
76	TBW1#	—	—	2 FRA+1->FRA, ROMD->prog buffer
77	XOR	R<7:0>, F	Z,N	1 (A).XOR.(R)->(目标)
78	XORI	I<7:0>	Z,N	1 I.XOR.(A)->(A)

注：指令集说明

- 1: i—立即数， F—标志位， A—寄存器 A， R—寄存器 R， B—寄存器 R 的第 B 位或移动 B 位。
- 2: C—进位/借位， DC—半进位/半借位， Z—零标志位， OV—溢出标志位， N—负标志位。
- 3: TOS—顶级堆栈。
- 4: 如果标志位 F = 0，则目标寄存器为寄存器 A；如果标志位 F = 1，则目标寄存器为寄存器 R。
- 5: 79 条指令中另有一条 NOP 指令未在上表中描述。
- 6: SECTION 指令中，N 的位数，视实际芯片而定。对本芯片，通用数据存储器 GPR 分为 8 个存储体组，所以 N 的位数是 3 位。
- 7: PAGE 指令中，N 的位数，视实际芯片而定。对本芯片，没有 PCRU 寄存器，N 的位数是 2 位。
- 8: PC 的位数以及 PCRU 寄存器，视实际芯片而定。对本芯片，PC 的位数是 13 位，没有 PCRU 寄存器。

附录2 特殊功能寄存器总表

地址	名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	上电复位值
FF80 _H	IAD	IAD<7:0>								
FF81 _H	IAAL	IAAL<7:0>								
FF82 _H	IAAH	IAAH<7:0>								
FF83 _H	BKSR	—	—	—	—	—	DBKSR<2:0>			0000 0000
FF84 _H	PSW	—	UF	OF	N	OV	Z	DC	C	x00 xxxx
FF85 _H	AREG	AREG<7:0>								
FF86 _H	PCRL	PCRL<7:0>								
FF87 _H	PCRH	—	—	—	PCRH<4:0>					
FF88 _H	MULA/MULL	MULA<7:0> / MULL<7:0>								
FF89 _H	MULB/MULH	MULB<7:0> / MULH<7:0>								
FF8A _H	DIVEL/DIVQL	DIVEL<7:0> / DIVQL<7:0>								
FF8B _H	DIVEH/DIVQH	DIVEH<7:0>/ DIVQH<7:0>								
FF8C _H	DIVS/DIVR	DIVS<7:0>/DIVR<7:0>								
FF8D _H	TIM1_CCR2H	CCR2<15:8>								
FF8E _H	TIM1_CCR3L	CCR3<7:0>								
FF8F _H	TIM1_CCR3H	CCR3<15:8>								
FF90 _H	FRAL	FRAL<7:0>								
FF91 _H	FRAH	FRAH<7:0>								
FF92 _H	ROMDL	ROMDL<7:0>								
FF93 _H	ROMDH	ROMDH<7:0>								
FF94 _H	ROMCL	—	IAPSEL<2:0>			FPEE	WREN	WR	—	0000 0000
FF95 _H	ROMCH	ROMCH<7:0>								
FF96 _H	INTG	GIE	GIEL	—	—	SOFTIF	INTVEN0	INTV<1:0>		0000 0000

地址	名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	上电复位值
FF97 _H	INTP	IGP<7:0>								0000 0000
FF98 _H	INTC0	PEG3<1:0>		PEG2<1:0>		PEG1<1:0>		PEG0<1:0>		0000 0000
FF99 _H	TIM1_CCR4L	CCR4<7:0>								
FF9A _H	INTE0	SPI_GIEIE	—	—	—	PIE8	TIM1IE	T8NIE	ADIE	0000 0000
FF9B _H	INTF0	SPI_GEIF	—	—	—	PIF8	TIM1IF	T8NIF	ADIF	0000 0000
FF9C _H	INTE1	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0	0000 0000
FF9D _H	INTF1	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0000 0000
FF9E _H	INTE2	T21MIE2	I2CIE	T21MIE1	T21MIE0	T21PIE	T21VIE	RXIE	TXIE	0000 0000
FF9F _H	INTF2	T21MIF2	I2CIF	T21MIF1	T21MIF0	T21PIF	T21VIF	RXIF	TXIF	0000 0000
FFA0 _H	SPICON0	RXCLR	TXCLR	CKS<1:0>		RBIM<1:0>		TBIM<1:0>		0011 0000
FFA1 _H	SPICON1	DFS<1:0>		DRE	—	REN	MS	SPIRST	SPIEN	0000 0000
FFA2 _H	SPIIE	—	TBWEIE	NSSIE	IDIE	ROIE	TEIE	RBIE	TBIE	0000 0000
FFA3 _H	SPIIF	—	TBWEIF	NSSIF	IDIF	ROIF	TEIF	RBIF	TBIF	0000 0001
FFA4 _H	SPIRBR	RBR<7:0>								
FFA5 _H	SPITBW	TBW<7:0>								
FFA6 _H	PWRC	LPM<1:0>		—	N_RSTI	N_TO	N_PD	N POR	N_BOR	0001 1100
FFA7 _H	WDTC	—	—	—	—	WDTPRE	WDTPRS<2:0>			0000 1111
FFA8 _H	WKDC	WKDC <7:0>								
FFA9 _H	PWEN	—	—	MRSTF	PORLOST	BORFLT<1:0>		RCEN	—	0000 1011
FFAA _H	PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
FFAB _H	PAT	PAT7	PAT6	PAT5	PAT4	PAT3	PAT2	PAT1	PAT0	1111 1111
FFAC _H	PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
FFAD _H	PBT	PBT7	PBT6	PBT5	PBT4	PBT3	PBT2	PBT1	PBT0	1111 1111
FFAE _H	PC	—	—	—	—	—	—	PC1	PC0	xxxx xxxx
FFAF _H	PCT	—	—	—	—	—	—	PCT1	PCT0	0000 0011

地址	名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	上电复位值
FFB0_H	PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0	0000 1000
FFB1_H	PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0	0000 0000
FFB2_H	PCPU	—	—	—	—	—	—	PCPU1	PCPU0	0000 0000
FFB3_H	TIM1_BDTRH	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK<1:0>		0000 0000
FFB4_H	TIM1_CCR4H					CCR4<15:8>				0000 0000
FFB5_H	PORTCTR	—	—	TIM1_CH4EN	TIM1_CH3EN	AD_ETR1EN	ADC_ETR0EN	PBOD<1:0>		0000 0000
FFB6_H	TIM1_BDTRL					DTG<7:0>				0000 0000
FFB7_H	PAPD	PAPD7	PAPD6	PAPD5	PAPD4	PAPD3	PAPD2	PAPD1	PAPD0	0000 0000
FFB8_H	PBPD	PBPD7	PBPD6	PBPD5	PBPD4	PBPD3	PBPD2	PBPD1	PBPD0	0000 0000
FFB9_H	FRALN					FRALN<7:0>				0000 0000
FFBA_H	FRAHN					FRAHN<7:0>				0000 0000
FFBB_H	T8N					T8N<7:0>				0000 0000
FFBC_H	T8NC	T8NEN	T8NCLK	T8NM	T8NEG	T8NPRE		T8NPRS<2:0>		0000 0000
FFBD_H	TIM1_CR1L	ARPE		CMS<1:0>	DIR	OPM	URS	UDIS	CEN	0000 0000
FFBE_H	TIM1_CR1H	—	HTOE OFF	—	—	—	CMPSELP	CKD<1:0>		0000 0000
FFBF_H	TIM1_CR2L	TI1S		MMS<2:0>	—	CCUS	—	CCPC		0000 0000
FFC0_H	TIM1_CR2H	—	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	0000 0000
FFC1_H	TIM1_SMCRL	MSM		TS<2:0>	OCCS			SMS<2:0>		0000 0000
FFC2_H	TIM1_SMCRH	ETP	ECE	ETPS<1:0>			ETF<3:0>			0000 0000
FFC3_H	TIM1_IERL	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE	0000 0000
FFC4_H	TIM1_IERH	—	—	—	CC4OIE	CC3OIE	CC2OIE	CC1OIE	—	0000 0000
FFC5_H	TIM1_IDRL	BID	TID	COMID	CC4ID	CC3ID	CC2ID	CC1ID	UID	0000 0000
FFC6_H	TIM1_IDRH	—	—	—	CC4OID	CC3OID	CC2OID	CC1OID	—	0000 0000
FFC7_H	TIM1_IVSL	BIS	TIS	COMIS	CC4IS	CC3IS	CC2IS	CC1IS	UIS	0000 0000

地址	名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	上电复位值							
FFC8 _H	TIM1_IVSH	—	—	—	CC4OIS	CC3OIS	CC2OIS	CC1OIS	—	0000 0000							
FFC9 _H	TIM1_RIFL	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF	0000 0000							
FFCA _H	TIM1_RIFH	—	—	—	CC4OIF	CC3OIF	CC2OIF	CC1OIF	—	0000 0000							
FFCB _H	TIM1_IFML	BIM	TIM	COMIM	CC4IM	CC3IM	CC2IM	CC1IM	UIM	0000 0000							
FFCC _H	TIM1_IFMH	—	—	—	CC4OIM	CC3OIM	CC2OIM	CC1OIM	—	0000 0000							
FFCD _H	TIM1_ICRL	BIC	TIC	COMIC	CC4IC	CC3IC	CC2IC	CC1IC	UIC	0000 0000							
FFCE _H	TIM1_ICRH	—	—	—	CC4OIC	CC3OIC	CC2OC	CC1OC	—	0000 0000							
FFCF _H	TIM1_EGRL	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG	0000 0000							
FFD0 _H	TIM1_CCMR1L	OC1CE	OC1M<2:0>			OC1PE	OC1FE	CC1S<1:0>		0000 0000							
FFD0 _H	TIM1_CCMR1L	IC1F<3:0>				IC1PSC<1:0>		CC1S<1:0>		0000 0000							
FFD1 _H	T21L	T21<7:0>								0000 0000							
FFD2 _H	T21H	T21<15:8>								0000 0000							
FFD3 _H	T21PL	T21P<7:0>								1111 1111							
FFD4 _H	T21PH	T21P<15:8>								1111 1111							
FFD5 _H	T21R0L	T21R0<7:0>								0000 0000							
FFD6 _H	T21R0H	T21R0<15:8>								0000 0000							
FFD7 _H	T21CL	T21M<3:0>			CAP1S<1:0>		CAP0S<1:0>		0000 0000								
FFD8 _H	T21CM	CAP2S<1:0>		T21OM20	T21OM21	T21PRS<3:0>											
FFD9 _H	T21CH	T21EN	T21POS<6:0>														
FFDA _H	ADCRL	ADCRL<7:0>								xxxx xxxx							
FFDB _H	ADCRH	ADCRH<7:0>								xxxx xxxx							
FFDC _H	ADCCL	ADCHS<3:0>				SMPON	SMPS	ADTRG	ADEN	1111 0100							
FFDD _H	ADCCH	ADFM	ADCKS<2:0>			ADST<1:0>		ADVREFS<1:0>		0100 1000							
FFDE _H	ANSL	—	ANSL6	ANSL5	ANSL4	ANSL3	ANSL2	ANSL1	ANSL0	0000 0000							
FFDF _H	—	—	—	—	—	—	—	—	—	0000 0000							

地址	名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	上电复位值
FFE0 _H	RXB									0000 0000
FFE1 _H	RXC	RXEN	RXM	—	—	—	OERR	FERR	RXR8	0000 000x
FFE2 _H	TXB									0000 0000
FFE3 _H	TXC	TXEN	TXM	BRGH	—	—	—	TRMT	TXR8	0000 0010
FFE4 _H	BRR									0000 0000
FFE5 _H	T21R1L									0000 0000
FFE6 _H	T21R1H									0000 0000
FFE7 _H	T21R2L									0000 0000
FFE8 _H	T21R2H									0000 0000
FFE9 _H	T21OC	T21TR	—	PT2EN<1:0>		T21OM2n2	PT1EN		PT0EN<1:0>	0000 0000
FFEA _H	TIM1_CCMR1H	OC2CE		OC2M<2:0>		OC2PE	OC2FE		CC2S<1:0>	0000 0000
FFEA _H	TIM1_CCMR1H			IC2F<3:0>		IC2PSC<1:0>			CC2S<1:0>	0000 0000
FFEB _H	TIM1_CCMR2L	OC3CE		OC3M<2:0>		OC3PE	OC3FE		CC3S<1:0>	0000 0000
FFEB _H	TIM1_CCMR2L			IC3F<3:0>		IC3PSC<1:0>			CC3S<1:0>	0000 0000
FFEC _H	TIM1_CCMR2H	OC4CE		OC4M<2:0>		OC4PE	OC4FE		CC4S<1:0>	0000 0000
FFEC _H	TIM1_CCMR2H			IC4F<3:0>		IC4PSC<1:0>			CC4S<1:0>	0000 0000
FFED _H	TIM1_CCERL	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E	0000 0000
FFEE _H	TIM1_CCERH	CC4NP	—	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	0000 0000
FFEF _H	I2CX16	—	—	—						0000 0000
FFF0 _H	I2CC	I2CTE	I2CPU	I2COD	I2CTAS	I2CANAE	I2CCSE	I2CRST	I2CEN	0000 0000
FFF1 _H	I2CSA								I2CRW	0000 0000
FFF2 _H	I2CTB									0000 0000
FFF3 _H	I2CRB									0000 0000
FFF4 _H	I2CIEC	I2CWUPEN	I2CNAIE	I2CROIE	I2CTEIE	I2CRBIE	I2CTBIE	I2CSPIE	I2CSRIE	0000 0000
FFF5 _H	I2CIFC	—	I2CNAIF	I2CROIF	I2CTEIF	I2CRBIF	I2CTBIF	I2CSPIF	I2CSRIF	1000 0100

地址	名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	上电 复位值
FFF6 _H	TIM1_CNTL				CNT<7:0>					0000 0000
FFF7 _H	TIM1_CNTH				CNT<15:8>					0000 0000
FFF8 _H	TIM1_PSCL				PSC<7:0>					0000 0000
FFF9 _H	TIM1_PSCH				PSC<15:8>					0000 0000
FFFA _H	TIM1_ARRL				ARR<7:0>					1111 1111
FFFB _H	TIM1_ARRH				ARR<15:8>					1111 1111
FFFC _H	TIM1_RCR				REP<7:0>					0000 0000
FFFD _H	TIM1_CCR1L				CCR1<7:0>					0000 0000
FFFE _H	TIM1_CCR1H				CCR1<15:8>					0000 0000
FFFF _H	TIM1_CCR2L				CCR2<7:0>					0000 0000

附录3 电气特性

附录3.1 参数特性表

◆ 最大标称值

参数	符号	条件	标称值	单位
电源电压	VDD	—	-0.3 ~ 7.5	V
输入电压	V _{IN}	—	-0.3 ~ VDD + 0.3	V
输出电压	V _{OUT}	—	-0.3 ~ VDD + 0.3	V
存储温度	T _{STG}	—	-55 ~ 125	°C
操作温度	T _{OPR}	VDD: 2.3 ~ 5.5V	-40 ~ 85	°C

◆ 芯片功耗特性参数表

参数	符号	最小值	典型值	最大值	单位	工作条件
芯片供电电压	VDD	2.3	—	5.5	V	-40°C ~ 85°C
芯片静态电流	I _{DD}	—	900	—	uA	25°C, VDD = 5V, BOR 不使能, 所有的 I/O 端口输入低电平, MRSTN = 0。
IDLE0 休眠模式下芯片电流	I _{PDO}	—	5	—	μA	25°C, VDD = 5V, BOR 使能, WDT 使能。LDO 禁止, 时钟源停振
IDLE1 休眠模式下芯片电流	I _{PD1}	—	40	—	μA	25°C, VDD = 5V, BOR 使能, WDT 使能, 内部 RC 振荡器。LDO 使能, 时钟源停振
IDLE2 休眠模式下芯片电流	I _{PD2}	—	150	—	μA	25°C, VDD = 5V, BOR 使能, WDT 使能, 内部 RC 振荡器。LDO 使能, 时钟源不停振
正常运行模式芯片电流	I _{OP}	—	2	—	mA	25°C, VDD = 5V, 正常运行模式, 内部 16MHz 时钟, I/O 端口输出固定电平, 无负载。
VDD 管脚的最大输入电流	I _{MAXVDD}	—	80	—	mA	25°C, VDD = 5V
VSS 管脚的最大输出电流	I _{MAXVSS}	—	200	—	mA	25°C, VDD = 5V
I/O 端口灌电流	I _{OL}	—	24	—	mA	25°C, VDD = 5V V _{OL} = 0.6V
I/O 端口拉电流	I _{OH}	—	12	—	mA	25°C, VDD = 5V V _{OH} = 4.4V

◆ 芯片输入端口特性表

芯片工作温度范围: -40°C ~ 85°C						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输入高电平 (有施密特输入特性)	V_{IH}	0.8VDD	—	VDD	V	2.3V ≤ VDD ≤ 5.5V
主复位信号 MRSTN 输入高电平 (有施密特输入特性)		0.8VDD	—	VDD	V	
I/O 端口输入低电平	V_{IL}	VSS	—	0.18VDD	V	2.3V ≤ VDD ≤ 5.5V
主复位信号 MRSTN 输入低电平		VSS	—	0.20VDD	V	
I/O 端口输入漏电流	I_{IL}	—	—	+1	μA	2.3V ≤ VDD ≤ 5.5V VSS ≤ Vpin ≤ VDD (端口处于高阻状态)
主复位端口漏电流		—	—	5	μA	
I/O 端口输入弱上拉电流	I_{WPU}	—	300	—	μA	25°C, VDD=5.0V Vpin = VSS
I/O 端口输入弱下拉电流	I_{WPD}	—	300	—	μA	25°C, VDD=5.0V Vpin = VDD
I/O 输入端口 VDD/2 输出	$V_{VDD/2}$	—	±3%	—	—	25°C, VDD=5V, 弱上拉和弱下拉同时使能

◆ 芯片输出端口特性表

芯片工作温度范围: -40°C ~ 85°C						
参数	符号	最小值	典型值	最大值	单位	测试条件
I/O 端口输出高电平	V_{OH}	VDD-0.7	—	—	V	2.3V ≤ VDD ≤ 5.5V $I_{OH} = 6.0 \text{ mA}$
I/O 端口输出低电平	V_{OL}	—	—	0.6	V	2.3V ≤ VDD ≤ 5.5V $I_{OL} = 12 \text{ mA}$

◆ 系统时钟要求表

参数	符号	最小值	典型值	最大值	单位	测试条件
系统时钟频率	F_{osc}	—	—	16M	Hz	2.3V ≤ VDD ≤ 5.5V
系统时钟周期	T_{osc}	62.5	—	—	ns	2.3V ≤ VDD ≤ 5.5V
机器周期	T_{inst}	125	—	—	ns	—
外部时钟高电平 和低电平时间	T_{OSL}, T_{OSH}	15	—	—	ns	—
外部时钟上升 和下降时间	T_{OSR}, T_{OSF}	—	—	15	ns	—
WDT 溢出时间 (不分频)	T_{WDT}	4.7 (54KHz)	8 (32KHz)	26.7 (9.6KHz)	ms	2.3V ≤ VDD ≤ 5.5V -40°C ~ 85°C

◆ 12 位 ADC 特性表

参数	符号	最小值	典型值	最大值	单位	测试条件
电源电压	VDD	2.3	—	5.5	V	—
分辨率	R _R	—	—	12	bit	—
差分线性度	DNL	—	—	±2	LSB	—
积分线性度	INL	—	—	±2	LSB	—
失调误差	E _{OFF}	—	±2	±3	LSB	参见表格下方的模拟小信号 ADC offset 特性表
参考电压	V _{REF}	1.3	—	VDD	V	—
模拟输入电压	V _{ADIN}	—	—	V _{REF}	V	—
输入电容	C _{ADIN}	—	—	40	Pf	—
输入电阻	R _{ADIN}	—	—	10	KΩ	—
转换时钟频率	F _{ADCLK}	32KHz	—	8	MHz	AD 转换选择 VDD 或外部 VREFP 作为正端参考电压;
转换时间 (不包括采样时间)	T _{ADC}	—	13	—	Tadclk	—
采样时间	T _{ADS}	250	—	—	ns	—

注 1: 建议 ADC 转换在高速模式下进行。

注 2: 建议 ADC 转换时钟频率设置在 512KHz~2MHz 之间, 频率设置过高或过低, 都可能会导致 ADC 转换精度降低。

◆ 模拟小信号 ADC offset 特性表

1) ADC 时钟频率为 1MHz、采样时间为 8Tad 时, 不同 ADC 参考电压, 不同 VDD, 对应于不同模拟输入 Vain 电压小信号的 ADC offset 典型值如下:

参数	最小值	典型值	最大值	测试条件			
				模拟输入电压 Vain	参考电压	ADC 时钟频率	电源电压 VDD
ADC offset	—	2LSB	—	0mV	外部 VREF1.8V	1MHz	3.0V
	—	-2LSB	—		外部 VREF1.8V		5.0V
	—	1LSB	—		VDD		3.0V
	—	-2LSB	—		VDD		5.0V
	—	2LSB	—	4mV	外部 VREF1.8V	1MHz	3.0V
	—	-2LSB	—		外部 VREF1.8V		5.0V
	—	1LSB	—		VDD		3.0V
	—	-2LSB	—		VDD		5.0V

◆ ADC 转换时间对照表, ADC 转换正端参考电压选择为 VDD 或外部 VREFP

A/D 时钟源选择	工作频率			
	16M	8M	4M	1M
Fosc	不推荐使用	$T_{ADCLK} = 0.125\mu s$	$T_{ADCLK} = 0.25\mu s$	$T_{ADCLK} = 1\mu s$
Fosc/2	$T_{ADCLK} = 0.125\mu s$	$T_{ADCLK} = 0.25\mu s$	$T_{ADCLK} = 0.5\mu s$	$T_{ADCLK} = 2\mu s$
Fosc/4	$T_{ADCLK} = 0.25\mu s$	$T_{ADCLK} = 0.5\mu s$	$T_{ADCLK} = 1\mu s$	$T_{ADCLK} = 4\mu s$
Fosc/8	$T_{ADCLK} = 0.5\mu s$	$T_{ADCLK} = 1\mu s$	$T_{ADCLK} = 2\mu s$	$T_{ADCLK} = 8\mu s$
Fosc/16	$T_{ADCLK} = 1\mu s$	$T_{ADCLK} = 2\mu s$	$T_{ADCLK} = 4\mu s$	$T_{ADCLK} = 16\mu s$
Fosc/32	$T_{ADCLK} = 2\mu s$	$T_{ADCLK} = 4\mu s$	$T_{ADCLK} = 8\mu s$	$T_{ADCLK} = 32\mu s$
Fosc/64	$T_{ADCLK} = 4\mu s$	$T_{ADCLK} = 8\mu s$	$T_{ADCLK} = 16\mu s$	不推荐使用

◆ 内部 16MHz 时钟校准性表

参数	符号	最小值	典型值	最大值	单位	测试条件
内部 16MHz 时钟频率	f_{intosc}	15.84	16	16.16	MHz	25°C, VDD = 5V
		15.68	16	16.32	MHz	-40°C ~ 85°C, VDD=2.3V~5.5V

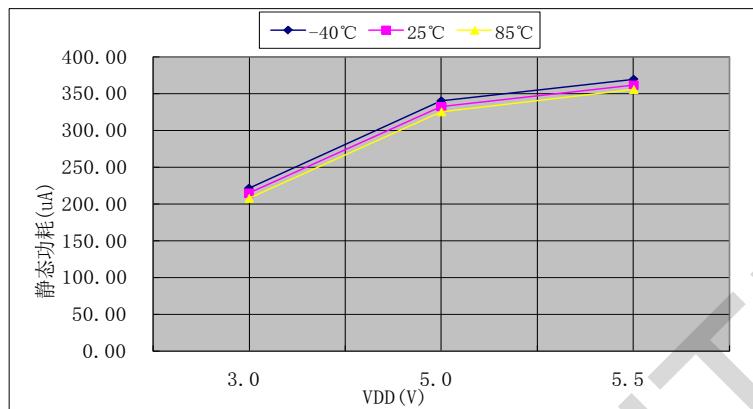
◆ 低电压复位 BOR 电压特性

参数	符号	最小值	典型值	最大值	单位	测试条件
BOR 低电压设定电压 1	V_{bor1}	2.9	3.1	3.3	V	25°C,
BOR 低电压设定电压 2	V_{bor2}	2.3	2.5	2.7	V	25°C,
BOR 低电压设定电压 3	V_{bor3}	1.9	2.1	2.3	V	25°C,
BOR 低电压复位脉宽	T_{bor}	-	220	-	us	设计理论值

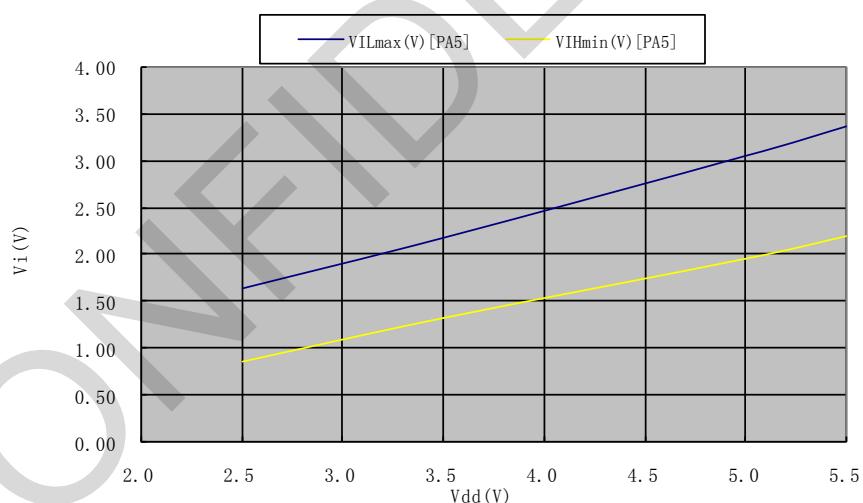
附录3. 2 参数特性图

本节中所列图示均为抽样测试，仅作为设计参考之用。其中部分图示中所列的数据已超出指定的操作范围，此类信息也仅供参考，芯片只保证在指定的范围内正常工作。

◆ 芯片静态电流随芯片电压变化特性图

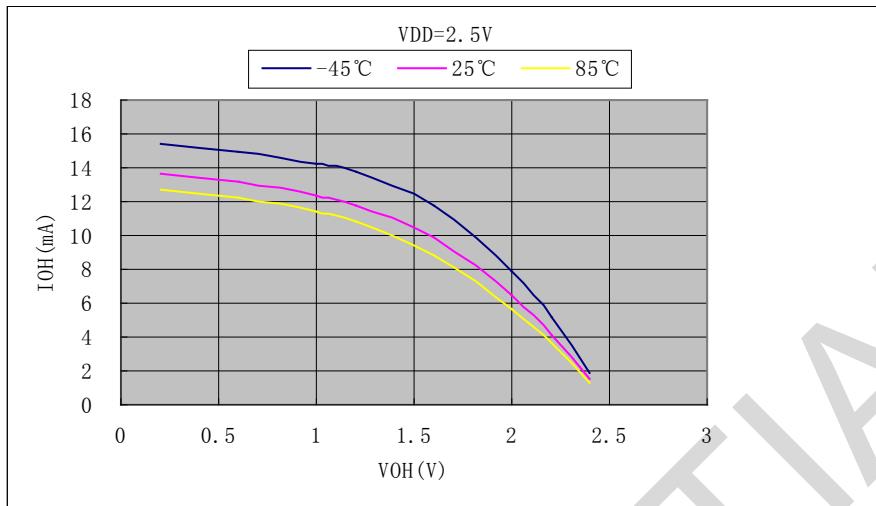


◆ I/O 端口信号输入特性图（室温 25°C）

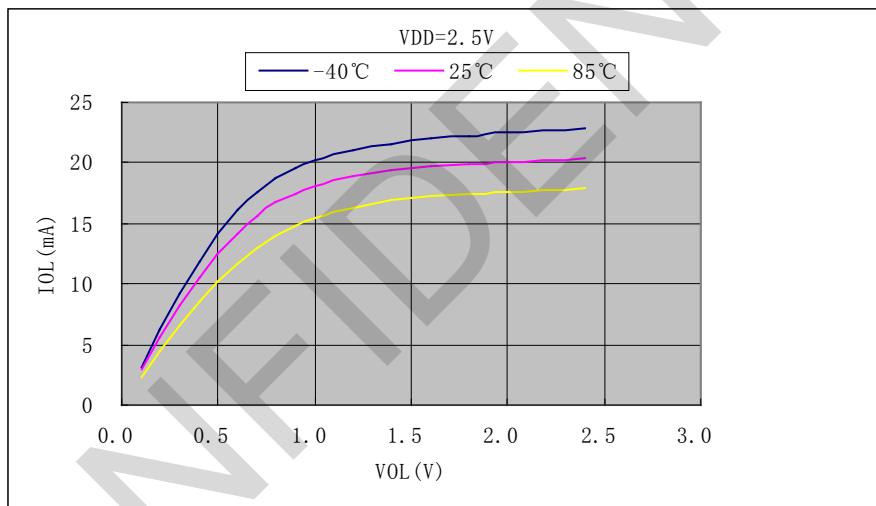


◆ I/O 端口信号输出特性图

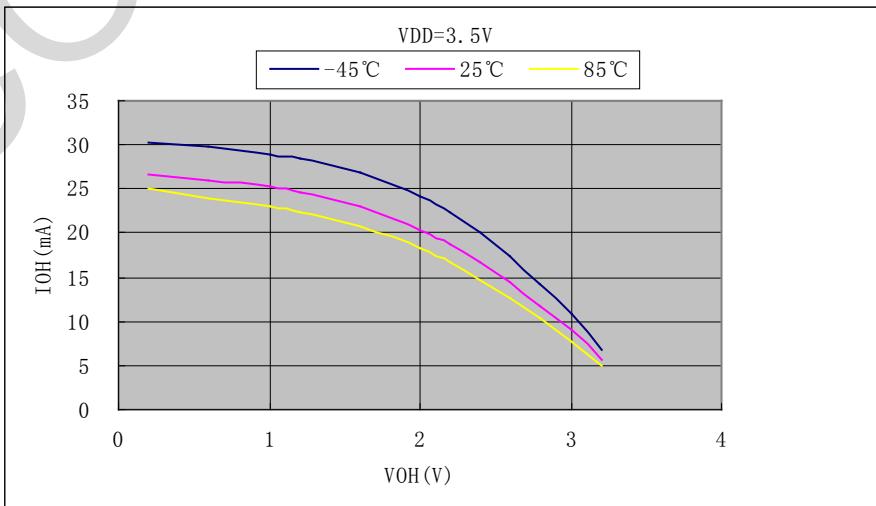
A: V_{OH} vs I_{OH} @ $VDD=2.5V$



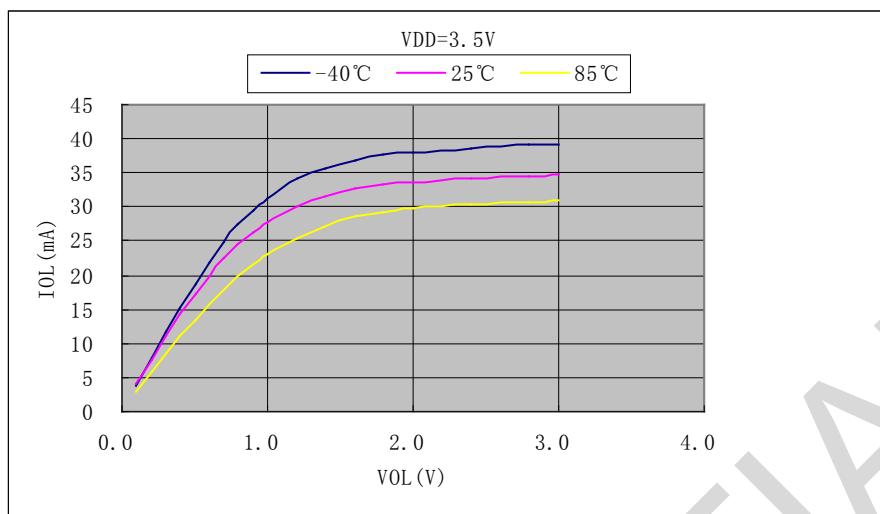
B: V_{OL} vs I_{OL} @ $VDD=2.5V$



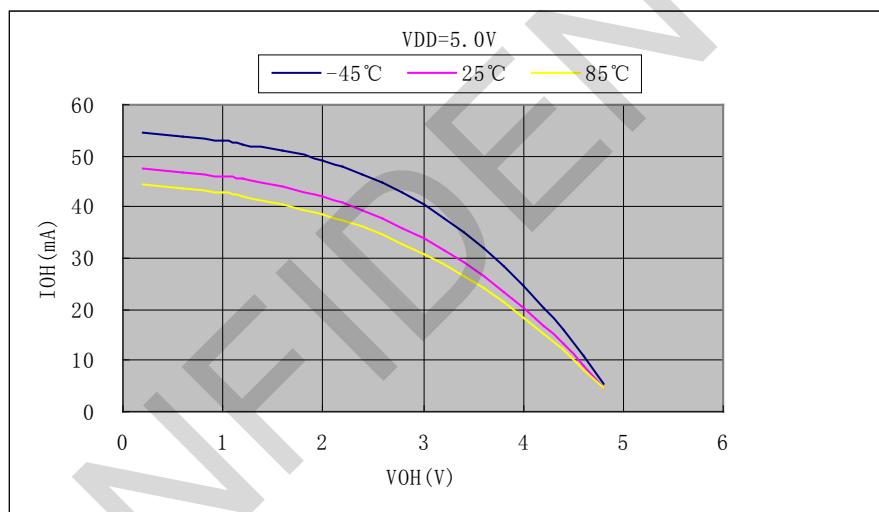
C: V_{OH} vs I_{OH} @ $VDD=3.5V$



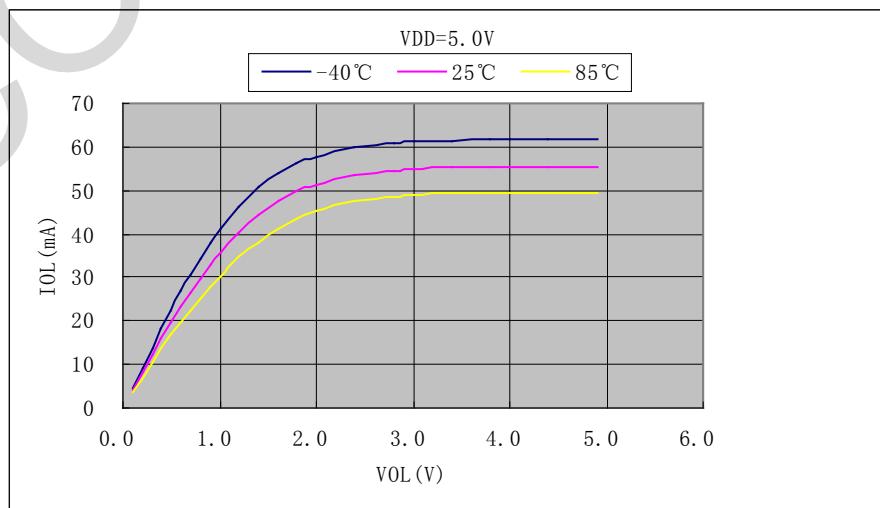
D: V_{OL} vs I_{OL} @VDD=3.5V

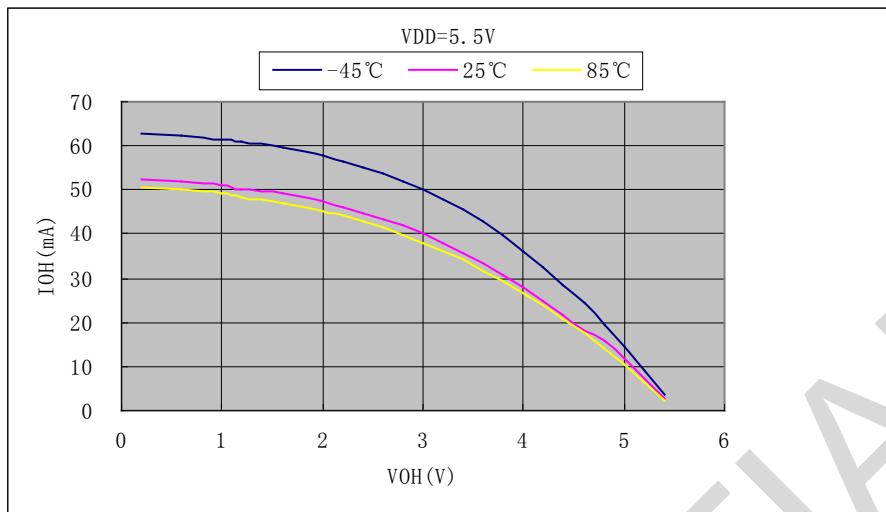


E: V_{OH} vs I_{OH} @VDD=5.0V



F: V_{OL} vs I_{OL} @VDD=5.0V



G: V_{OH} vs I_{OH} @ $VDD=5.5V$ H: V_{OL} vs I_{OL} @ $VDD=5.5V$ 