

# UM32Mx56 用户手册

版本：V1.0



UNICMICRO  
广芯微电子

广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

## 条款协议

本文档的所有部分，其著作权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

## 目录

1	文档约定 .....	2
1.1	寄存器相关缩写词列表 .....	2
1.2	词汇表 .....	2
2	产品简介 .....	3
2.1	系统概述 .....	3
2.2	主要特性 .....	4
3	存储器和总线架构 .....	7
3.1	系统架构 .....	7
3.2	总线架构图 .....	8
3.3	存储器映射 (Memory Mapping) .....	9
4	处理器 .....	11
4.1	概述 .....	11
4.2	主要特性 .....	11
4.3	功能框图 .....	12
4.4	内核寄存器组 .....	12
5	系统配置 (SCU) .....	13
5.1	时钟框图 .....	13
5.2	时钟选择 .....	13
5.3	复位源 .....	14
5.3.1	内部 POR 上电复位 .....	14
5.3.2	LVR 复位 .....	14
5.3.3	RESETEN 复位 .....	14
5.3.4	LOCKUP 复位 .....	15
5.3.5	LVD 复位 .....	15
5.3.6	WDT 复位 .....	15
5.3.7	WWDT 复位 .....	15
5.3.8	SOFT_RESETN 复位 .....	15
5.3.9	模块复位 .....	15
5.4	低功耗模式 .....	15
5.4.1	Sleep 模式 .....	17
5.4.2	DeepSleep 模式 .....	17
5.4.3	Stop 模式 .....	18
5.5	系统寄存器 .....	19
5.5.1	系统控制寄存器 0 SCU_CTRL0 (偏移: 000h) .....	20

5.5.2	系统控制保护寄存器 SCU_CTRL_PROT (偏移: 008h)	21
5.5.3	时钟控制寄存器 SCU_OSC_CTRL (偏移: 0x00Ch)	22
5.5.4	外围模块时钟寄存器 SCU_PERI_CLKEN (偏移: 010h)	22
5.5.5	复位标识寄存器 SCU_RESET_FLAG (偏移: 020h)	23
5.5.6	外围模块复位控制寄存器 SCU_PERI_RESET (偏移: 024h)	25
5.5.7	外部复位滤波控制寄存器 SCU_EXTRST_CTRL (偏移: 028h)	26
5.5.8	端口 PA 功能配置寄存器 SCU_PA_SEL (偏移: 030h)	26
5.5.9	端口 PB 功能配置寄存器 SCU_PB_SEL (偏移: 034h)	29
5.5.10	端口 PC 功能配置寄存器 SCU_PC_SEL (偏移: 038h)	32
5.5.11	端口 PD 功能配置寄存器 SCU_PD_SEL (偏移: 03Ch)	35
5.5.12	端口数模配置寄存器 SCU_PAD_ADS (偏移: 054h)	37
5.5.13	端口驱动能力配置寄存器 SCU_PAD_DR (偏移: 060h)	38
5.5.14	端口上拉配置寄存器 SCU_PAD_PU (偏移: 06Ch)	40
5.5.15	端口下拉配置寄存器 SCU_PAD_PD (偏移: 078h)	41
5.5.16	端口开漏输出配置寄存器 SCU_PAD_OD (偏移: 084h)	43
5.5.17	端口输入类型配置寄存器 SCU_PAD_CS (偏移: 090h)	45
5.5.18	端口输入配置寄存器 SCU_PAD_IE (偏移: 09Ch)	46
5.5.19	端口输入电平寄存器 SCU_PAD_STATUS (偏移: 0A4h)	48
5.5.20	端口速度配置寄存器 SCU_PAD_SR (偏移: 0A8h)	50
5.5.21	IO 控制保护寄存器 SCU_IOCTRL_PROT (偏移: 0B4h)	51
5.5.22	LVD 配置寄存器 SCU_LVD_CFG (偏移: 0B8h)	52
5.5.23	PGA 控制寄存器 SCU_PGA_CFG (偏移: 0BCh)	53
5.5.24	外部复位端口选择寄存器 SCU_EXTRST_SEL (偏移: 0D0h)	55
5.5.25	停止模式选择寄存器 SCU_STOP_SEL (偏移: 0D4h)	55
5.5.26	软件复位寄存器 SCU_SOFT_RSTN (偏移: 0D8h)	56
5.5.27	中断向量地址重映射寄存器 SCU_VET_OFFSET (偏移: 0DCh)	56
5.5.28	LVD 中断寄存器 SCU_LVD_INTR (偏移: FCh)	56
5.5.29	VREF 控制寄存器 SCU_VREF (偏移: 188h)	56
6	EFC	57
6.1	概述	57
6.2	主要特性	57
6.3	eFlash 数据防护机制	57
6.3.1	eFlash 存储区域	58
6.3.2	EEPROM 配置字	59
6.3.3	SWD 硬件禁止	59
6.4	寄存器描述	60
6.4.1	EFC 控制寄存器 (EFC_CTRL) (偏移: 00h)	60
6.4.2	EFC OTP 状态寄存器 (EFC_OTPSTATUS) (偏移: 04h)	61

6.4.3	EFC KEY 输入寄存器 (EFC_KEY) (偏移: 08h)	62
6.4.4	EFC 中断状态寄存器 (EFC_INTSTATUS) (偏移: 0Ch)	62
6.4.5	EFC 中断使能寄存器 (EFC_INTEN) (偏移: 10h)	64
6.4.6	EFC 擦写控制寄存器 (EFC_EP_CTRL) (偏移: 14h)	65
6.4.7	EFC LDO TRIM 寄存器 (EFC_LDOTRIM) (偏移: 1Ch)	65
6.4.8	EFC RCH TRIM 寄存器 (EFC_RCHTRIM) (偏移: 20h)	66
6.4.9	EFC RCL TRIM 寄存器 (EFC_RCLTRIM) (偏移: 24h)	66
6.4.10	EFC PGA0 TRIM 寄存器 (EFC_PGA0TRIM) (偏移: 28h)	66
6.4.11	EFC PGA1 TRIM 寄存器 (EFC_PGA1TRIM) (偏移: 2Ch)	66
6.4.12	EFC ADC TRIM 寄存器 (EFC_ADCTRIM) (偏移: 30h)	66
6.5	软件流程	67
6.5.1	Write 操作	67
6.5.2	Erase 操作	68
7	NVIC	69
7.1	概述	69
7.2	主要特性	69
7.3	中断源	69
8	GPIO	71
8.1	概述	71
8.2	主要特性	71
8.3	寄存器描述	71
8.3.1	GPIO 数据方向寄存器 (GPIO_DIR) (偏移: 00h)	72
8.3.2	GPIO 输出置位寄存器 (GPIO_SET) (偏移: 08h)	72
8.3.3	GPIO 输出清零寄存器 (GPIO_CLR) (偏移: 0Ch)	72
8.3.4	GPIO 输出引脚映射寄存器 (GPIO_ODATA) (偏移: 10h)	73
8.3.5	GPIO 输入引脚映射寄存器 (GPIO_IDATA) (偏移: 14h)	73
8.3.6	GPIO 中断使能寄存器 (GPIO_IEN) (偏移: 18h)	73
8.3.7	GPIO 中断触发模式寄存器 (GPIO_IS) (偏移: 1Ch)	74
8.3.8	GPIO 中断边沿触发设置寄存器 (GPIO_IBE) (偏移: 20h)	74
8.3.9	GPIO 中断高低电平触发设置寄存器 (GPIO_IEV) (偏移: 24h)	74
8.3.10	GPIO 中断状态清除寄存器 (GPIO_IC) (偏移: 28h)	74
8.3.11	GPIO 原始中断状态寄存器 (GPIO_RIS) (偏移: 2Ch)	75
8.3.12	GPIO 屏蔽后中断状态寄存器 (GPIO_MIS) (偏移: 30h)	75
8.3.13	GPIO 输入滤波选择寄存器 (GPIO_FLS) (偏移: 34h)	75
8.4	使用流程	76
8.4.1	输入输出 IO	76
8.4.2	中断触发模式	76

8.4.3	清除中断 .....	76
8.4.4	输入滤波 .....	77
9	UART0/1 .....	78
9.1	概述 .....	78
9.2	主要特性 .....	78
9.3	寄存器描述 .....	78
9.3.1	中断状态寄存器 UART_ISR (偏移: 00h) .....	79
9.3.2	中断使能寄存器 UART_IER (偏移: 04h) .....	79
9.3.3	控制寄存器 UART_CR (偏移: 08h) .....	80
9.3.4	发送数据寄存器 UART_TDR (偏移: 0Ch) .....	81
9.3.5	接收数据寄存器 UART_RDR (偏移: 10h) .....	81
9.3.6	波特率参数低位寄存器 UART_BRR (偏移: 14h) .....	81
9.3.7	延迟超时寄存器 UART_DLTOR (偏移: 18h) .....	81
9.4	使用流程 .....	82
9.4.1	串口的发送和接收 .....	82
9.4.2	串口初始化 .....	82
9.4.3	串口发送字节 .....	82
9.4.4	串口接收字节 .....	83
10	I2C .....	84
10.1	概述 .....	84
10.2	主要特征 .....	84
10.3	寄存器描述 .....	84
10.3.1	I2C 主机配置寄存器 I2C_MCFG (偏移: 00h) .....	85
10.3.2	I2C 主机控制寄存器 I2C_MCON (偏移: 04h) .....	85
10.3.3	I2C 主机中断使能寄存器 I2C_MIE (偏移: 08h) .....	86
10.3.4	I2C 主机中断标志寄存器 I2C_MIF (偏移: 0Ch) .....	87
10.3.5	I2C 主机状态寄存器 I2C_MSTA (偏移: 10h) .....	87
10.3.6	I2C 主机波特率寄存器 I2C_MBRG (偏移: 14h) .....	88
10.3.7	I2C 主机收发缓存寄存器 I2C_MBUF (偏移: 18h) .....	88
10.3.8	I2C 主机时序控制寄存器 I2C_MTIM (偏移: 1Ch) .....	88
10.3.9	I2C 主机超时寄存器 I2C_MTO (偏移: 20h) .....	89
10.3.10	I2C 从机控制寄存器 I2C_SCON (偏移: 24h) .....	89
10.3.11	I2C 从机中断使能寄存器 I2C_SIE (偏移: 28h) .....	90
10.3.12	I2C 从机中断状态寄存器 I2C_SIF (偏移: 2Ch) .....	91
10.3.13	I2C 从机状态寄存器 I2C_SSTA (偏移: 30h) .....	92
10.3.14	I2C 从机收发缓冲寄存器 I2C_SBUF (偏移: 34h) .....	93
10.3.15	I2C 从机地址寄存器 0 I2C_SADR0 (偏移: 38h) .....	93

10.3.16	I2C 从机地址寄存器 1 I2C_SADR1 (偏移: 3Ch)	93
10.3.17	I2C 从机地址寄存器 2 I2C_SADR2 (偏移: 40h)	93
10.3.18	I2C 从机地址寄存器 3 I2C_SADR3 (偏移: 44h)	94
10.4	使用流程	95
10.4.1	主机发送	95
10.4.2	主机接收	96
10.4.3	从机发送	97
10.4.4	从机接收	98
11	SPI	99
11.1	概述	99
11.2	主要特性	99
11.3	寄存器描述	100
11.3.1	SPI 配置寄存器 (SPI_CR) (偏移: 00h)	100
11.3.2	SPI 主模式控制寄存器 0 (SPI_CS0) (偏移: 04h)	102
11.3.3	SPI 主模式控制寄存器 1 (SPI_CS1) (偏移: 08h)	103
11.3.4	SPI 过程控制寄存器 (SPI_OPCR) (偏移: 14h)	104
11.3.5	SPI 中断控制寄存器 (SPI_IE) (偏移: 18h)	105
11.3.6	SPI 中断标志寄存器 (SPI_IF) (偏移: 1Ch)	106
11.3.7	SPI 发送缓存寄存器 (SPI_TXBUF) (偏移: 20h)	107
11.3.8	SPI 接收缓存寄存器 (SPI_RXBUF) (偏移: 24h)	107
11.3.9	SPI DMA 接收设置寄存器 (SPI_DMARXLEV) (偏移: 28h)	107
11.3.10	SPI DMA 发送设置寄存器 (SPI_DMATXLEV) (偏移: 2Ch)	107
11.4	使用流程	108
11.4.1	初始化程序	108
11.4.2	发送流程	109
11.4.3	接收流程	109
11.4.4	SPI DMA 发送流程	109
11.4.5	SPI DMA 接收流程	110
12	ATIMER	111
12.1	概述	111
12.2	主要特性	111
12.3	ATIMER 结构框图	112
12.4	功能描述	112
12.4.1	定时单元	112
12.4.2	定时器工作模式	114
12.4.3	向上计数	114
12.4.4	向下计数	117

12.4.5	中心对齐计数	119
12.4.6	重复计数器	121
12.4.7	Preload 寄存器	122
12.4.8	计数器工作时钟	123
12.4.9	内部时钟模式	123
12.4.10	外部时钟模式 1	123
12.4.11	外部时钟模式 2	125
12.4.12	内部触发信号 (ITRx)	127
12.4.13	捕捉/比较通道	127
12.4.14	输入捕捉模式	129
12.4.15	软件 Force 输出	131
12.4.16	输出比较模式	131
12.4.17	PWM 输出	132
12.4.18	非对称 PWM 移相控制	133
12.4.19	互补输出和死区插入	134
12.4.20	刹车功能	135
12.4.21	6-step PWM 输出	136
12.4.22	单脉冲输出	137
12.4.23	外部事件清除 OCxREF	139
12.4.24	编码器接口模式	139
12.4.25	ATIMER 从机模式	140
12.4.26	复位模式	141
12.4.27	门控模式	141
12.4.28	触发模式	142
12.4.29	外部事件触发的外部时钟计数模式	143
12.4.30	DMA 访问	144
12.4.31	输入异或功能	145
12.4.32	Debug 模式	145
12.4.33	DMA Burst	145
12.5	寄存器描述	146
12.5.1	ATIMER 控制寄存器 1 ATIMER_CR1 (偏移: 00h)	147
12.5.2	ATIMER 控制寄存器 2 ATIMER_CR2 (偏移: 04h)	149
12.5.3	ATIMER 从机模式控制寄存器 ATIMER_SMCR (偏移: 08h)	151
12.5.4	ATIMER DMA 和中断使能寄存器 ATIMER_DIER (偏移: 0Ch)	153
12.5.5	ATIMER 状态寄存器 ATIMER_SR (偏移: 10h)	154
12.5.6	ATIMER 事件产生寄存器 ATIMER_EGR (偏移: 14h)	155
12.5.7	ATIMER 捕捉/比较模式寄存器 1 ATIMER_CCMR1 (偏移: 18h)	156
12.5.8	ATIMER 捕捉/比较模式寄存器 2 ATIMER_CCMR2 (偏移: 1Ch)	159

12.5.9	ATIMER 捕捉/比较使能寄存器 ATIMER_CCER (偏移: 20h)	161
12.5.10	ATIMER 计数器寄存器 ATIMER_CNT (偏移: 24h)	164
12.5.11	ATIMER 预分频寄存器 ATIMER_PSC (偏移: 28h)	164
12.5.12	ATIMER 自动重载寄存器 ATIMER_ARR (偏移: 2Ch)	164
12.5.13	ATIMER 重复计数寄存器 ATIMER_RCR (偏移: 30h)	164
12.5.14	ATIMER 捕捉/比较寄存器 1 ATIMER_CCR1 (偏移: 34h)	165
12.5.15	ATIMER 捕捉/比较寄存器 2 ATIMER_CCR2 (偏移: 38h)	165
12.5.16	ATIMER 捕捉/比较寄存器 3 ATIMER_CCR3 (偏移: 3Ch)	165
12.5.17	ATIMER 捕捉/比较寄存器 4 ATIMER_CCR4 (偏移: 40h)	166
12.5.18	ATIMER 刹车和死区控制寄存器 ATIMER_BDTR (偏移: 44h)	166
12.5.19	ATIMER DMA 控制寄存器 ATIMER_DCR (偏移: 48h)	167
12.5.20	ATIMER DMA 访问寄存器 ATIMER_DMAR (偏移: 4Ch)	168
12.5.21	ATIMER 捕捉/比较模式寄存器 3 ATIMER_CCMR3 (偏移: 50h)	168
12.5.22	ATIMER 捕捉/比较寄存器 5 ATIMER_CCR5 (偏移: 54h)	169
12.5.23	ATIMER PWM 移相使能寄存器 ATIMER_PMEN (偏移: 58h)	169
12.5.24	ATIMER 移相偏移寄存器 1 ATIMER_PMC1 (偏移: 5Ch)	170
12.5.25	ATIMER 移相偏移寄存器 2 ATIMER_PMC2 (偏移: 60h)	170
12.5.26	ATIMER 移相偏移寄存器 3 ATIMER_PMC3 (偏移: 64h)	170
12.5.27	ATIMER 刹车输入控制寄存器 ATIMER_BKCTL (偏移: 68h)	170
12.6	使用流程	172
12.6.1	定时计数模式	172
12.6.2	PWM 模式	172
12.6.3	输入捕捉模式	173
12.6.4	互补输出和死区插入	174
12.6.5	刹车功能	174
12.6.6	编码器接口模式	175
12.6.7	DMA 模式	175
13	通用定时器 GTIMER0	177
13.1	概述	177
13.2	主要特性	177
13.3	通用定时器 GTIMER0 结构框图	178
13.4	功能描述	178
13.4.1	定时单元	178
13.4.2	定时器工作模式	179
13.4.2.1	向上计数	179
13.4.2.2	向下计数	180
13.4.2.3	中心对齐计数	181
13.4.3	计数器工作时钟	182
13.4.3.1	APBCLK	182

13.4.3.2	GTIMER_CHx (x=1,2) 引脚输入——外部时钟模式 1	182
13.4.3.3	GTIMER_ETR 引脚输入——外部时钟模式 2	183
13.4.4	内部触发信号 (ITRx)	183
13.4.5	捕捉/比较通道	184
13.4.5.1	捕获通道	184
13.4.5.2	比较通道	184
13.4.6	输入捕捉模式	185
13.4.7	PWM 输入模式	186
13.4.8	软件强制输出	186
13.4.9	输出比较模式	187
13.4.10	PWM 输出	188
13.4.10.1	PWM 边缘对齐模式	188
13.4.10.2	PWM 中央对齐模式	189
13.4.11	输出比较快速使能模式	189
13.4.12	单脉冲输出	190
13.4.13	外部事件清除 OCxREF	191
13.4.14	编码器接口模式 (Encoder Interface)	192
13.4.15	GTIMER 从机模式	193
13.4.15.1	复位模式	193
13.4.15.2	门控模式	194
13.4.15.3	触发模式	195
13.4.16	定时器同步	196
13.4.17	DMA Burst	196
13.4.18	输入异或功能	197
13.4.19	Debug 模式	197
13.5	寄存器描述	197
13.5.1	GTIMER 寄存器分布	197
13.5.2	控制寄存器 1 GTIMER_CR1 (偏移: 00h)	198
13.5.3	控制寄存器 2 GTIMER_CR2 (偏移: 04h)	199
13.5.4	从机模式控制寄存器 GTIMER_SMCR (偏移: 08h)	200
13.5.5	DMA 和中断使能寄存器 GTIMER_DIER (偏移: 0Ch)	202
13.5.6	状态寄存器 GTIMER_SR (偏移: 10h)	204
13.5.7	事件产生寄存器 GTIMER_EGR (偏移: 14h)	205
13.5.8	捕捉/比较模式寄存器 1 GTIMER_CCMR1 (偏移: 18h)	205
13.5.9	捕捉/比较模式寄存器 2 GTIMER_CCMR2 (偏移: 1Ch)	208
13.5.10	捕捉/比较使能寄存器 GTIMER_CCER (偏移: 20h)	211
13.5.11	计数器寄存器 GTIMER_CNT (偏移: 24h)	212
13.5.12	预分频寄存器 GTIMER_PSC (偏移: 28h)	212
13.5.13	自动重载寄存器 GTIMER_ARR (偏移: 2Ch)	212
13.5.14	捕捉/比较寄存器 1 GTIMER_CCR1 (偏移: 34h)	212

13.5.15	捕捉/比较寄存器 2 GTIMER_CCR2 (偏移: 38h)	213
13.5.16	捕捉/比较寄存器 3 GTIMER_CCR3 (偏移: 3Ch)	213
13.5.17	捕捉/比较寄存器 4 GTIMER_CCR4 (偏移: 40h)	213
13.5.18	DMA 控制寄存器 GTIMER_DCR (偏移: 48h)	214
13.5.19	DMA 访问寄存器 GTIMER_DMAR (偏移: 4Ch)	214
14	通用定时器 GTIMER1/2	215
14.1	概述	215
14.2	主要特性	215
14.3	通用定时器 GTIMER1/2 结构框图	216
14.4	功能描述	216
14.4.1	定时单元	216
14.4.2	定时器工作模式	217
14.4.2.1	向上计数	217
14.4.2.2	向下计数	218
14.4.2.3	中心对齐计数	219
14.4.3	计数器工作时钟	220
14.4.3.1	APBCLK	220
14.4.3.2	GTIMER_CHx (x=1,2) 引脚输入——外部时钟模式 1	220
14.4.3.3	GTIMER_ETR 引脚输入——外部时钟模式 2	221
14.4.4	内部触发信号 (ITRx)	221
14.4.5	捕捉/比较通道	222
14.4.5.1	捕获通道	222
14.4.5.2	比较通道	222
14.4.6	输入捕捉模式	223
14.4.7	PWM 输入模式	224
14.4.8	软件强制输出	224
14.4.9	输出比较模式	225
14.4.10	PWM 输出	226
14.4.10.1	PWM 边缘对齐模式	226
14.4.10.2	PWM 中央对齐模式	227
14.4.11	输出比较快速使能模式	227
14.4.12	单脉冲输出	228
14.4.13	外部事件清除 OCxREF	230
14.4.14	编码器接口模式 (Encoder Interface)	230
14.4.15	GTIMER 从机模式	232
14.4.15.1	复位模式	232
14.4.15.2	门控模式	233
14.4.15.3	触发模式	233
14.4.16	定时器同步	234
14.4.17	DMA Burst	234
14.4.18	输入异或功能	235

14.4.19	Debug 模式 .....	235
14.5	寄存器描述 .....	235
14.5.1	GTIMER 寄存器分布 .....	235
14.5.2	控制寄存器 1 GTIMER_CR1 (偏移: 00h) .....	237
14.5.3	控制寄存器 2 GTIMER_CR2 (偏移: 04h) .....	238
14.5.4	从机模式控制寄存器 GTIMER_SMCR (偏移: 08h) .....	239
14.5.5	DMA 和中断使能寄存器 GTIMER_DIER (偏移: 0Ch) .....	240
14.5.6	状态寄存器 GTIMER_SR (偏移: 10h) .....	242
14.5.7	事件产生寄存器 GTIMER_EGR (偏移: 14h) .....	243
14.5.8	捕捉/比较模式寄存器 1 GTIMER_CCMR1 (偏移: 18h) .....	244
14.5.9	捕捉/比较模式寄存器 2 GTIMER_CCMR2 (偏移: 1Ch) .....	246
14.5.10	捕捉/比较使能寄存器 GTIMER_CCER (偏移: 20h) .....	249
14.5.11	计数器寄存器 GTIMER_CNT (偏移: 24h) .....	250
14.5.12	预分频寄存器 GTIMER_PSC (偏移: 28h) .....	250
14.5.13	自动重载寄存器 GTIMER_ARR (偏移: 2Ch) .....	250
14.5.14	捕捉/比较寄存器 1 GTIMER_CCR1 (偏移: 34h) .....	251
14.5.15	捕捉/比较寄存器 2 GTIMER_CCR2 (偏移: 38h) .....	251
14.5.16	捕捉/比较寄存器 3 GTIMER_CCR3 (偏移: 3Ch) .....	251
14.5.17	捕捉/比较寄存器 4 GTIMER_CCR4 (偏移: 40h) .....	252
14.5.18	DMA 控制寄存器 GTIMER_DCR (偏移: 48h) .....	252
14.5.19	DMA 访问寄存器 GTIMER_DMAR (偏移: 4Ch) .....	253
15	LPTIMER .....	254
15.1	概述 .....	254
15.2	主要特性 .....	254
15.3	结构框图 .....	255
15.4	定时器功能 .....	255
15.4.1	单次计数和连续计数 .....	255
15.4.2	ETR 触发计数 .....	255
15.4.3	ETR 脉冲计数 .....	256
15.4.4	Timeout 模式 .....	256
15.5	捕捉比较功能 .....	256
15.5.1	PWM 输出 .....	256
15.5.2	输入捕捉 .....	257
15.5.3	Debug 模式 .....	257
15.6	寄存器描述 .....	257
15.6.1	LPTIMER 控制寄存器 1 LPTIMER_CR1 (偏移: 00h) .....	257
15.6.2	LPTIMER 控制寄存器 2 LPTIMER_CR2 (偏移: 04h) .....	258
15.6.3	LPTIMER 中断使能寄存器 LPTIMER_IER (偏移: 08h) .....	259

15.6.4	LPTIMER 中断标志寄存器 LPTIMER_SR (偏移: 0Ch)	260
15.6.5	LPTIMER 计数值寄存器 LPTIMER_CNT (偏移: 10h)	260
15.6.6	LPTIMER 捕捉比较配置寄存器 1 LPTIMER_CCMCFG1 (偏移: 14h)	260
15.6.7	LPTIMER 捕捉比较配置寄存器 2 LPTIMER_CCMCFG2 (偏移: 18h)	261
15.6.8	LPTIMER 自动重载寄存器 LPTIMER_ARR (偏移: 1Ch)	262
15.6.9	LPTIMER 捕捉比较寄存器 1 LPTIMER_CCR1 (偏移: 20h)	262
15.6.10	LPTIMER 捕捉比较寄存器 2 LPTIMER_CCR2 (偏移: 24h)	262
15.6.11	LPTIMER 计数值 load 寄存器 LPTIMER_LOAD (偏移: 28h)	262
15.6.12	LPTIMER 计数缓存寄存器 LPTIMER_BUFFER (偏移: 2Ch)	262
15.7	使用流程	263
15.7.1	普通定时器	263
15.7.2	结合 DMA 输入捕获功能	263
15.7.3	PWM 输出	264
15.7.4	ETR 脉冲触发计数模式	264
15.7.5	ETR 脉冲计数模式	265
15.7.6	Timeout 模式	265
16	DMA	267
16.1	概述	267
16.2	主要特性	267
16.3	寄存器描述	267
16.3.1	DMA 通道源传送地址寄存器 (DMA_SRCADDR_Cx) (偏移: 20*x+00h) (x=0,1)	268
16.3.2	DMA 通道目的传送地址寄存器 (DMA_DSTADDR_Cx) (偏移: 20*x+04h) (x=0,1)	268
16.3.3	DMA 通道控制信息寄存器 (DMA_CHCTRL_Cx) (偏移: 20*x+08h) (x=0,1)	268
16.3.4	DMA 通道传送状态寄存器 (DMA_CHSTS_Cx) (偏移: 20*x +0Ch) (x=0,1)	270
16.3.5	DMA 通道源外设选择寄存器 (DMA_CHSPER_Cx) (偏移: 20*x +10h) (x=0,1)	270
16.3.6	DMA 通道目标外设选择寄存器 (DMA_CHDPER_Cx) (偏移: 20*x +14h) (x=0,1)	271
16.3.7	DMA 控制器使能寄存器 (DMA_EN) (偏移: 40h)	273
16.3.8	DMA 软复位寄存器 (DMA_SOFT_RESET) (偏移: 44h)	273
16.3.9	DMA 中断指示寄存器 (DMA_INT_STATUS) (偏移: 48h)	273
16.3.10	DMA 中断屏蔽寄存器 (DMA_INT_MASK) (偏移: 4Ch)	273
16.3.11	DMA 外设请求状态寄存器 (DMA_PER_REQ) (偏移: 54h)	274
16.4	使用流程	275
17	CRC	276

17.1	概述.....	276
17.2	寄存器描述.....	276
17.2.1	数据寄存器 CRC_DATA (偏移: 00H) .....	276
17.2.2	初始值寄存器 CRC_INIT (偏移: 04H) .....	277
17.2.3	控制寄存器 CRC_CTRL (偏移: 08H) .....	277
17.3	使用流程 .....	277
18	RNG.....	278
18.1	概述.....	278
18.2	主要特性 .....	278
18.3	寄存器描述.....	278
18.3.1	随机数控制寄存器 RNG_CR (偏移: 0E0h) .....	278
18.3.2	随机数种子寄存器 RNG_SEED (偏移: 0E4h) .....	278
18.3.3	随机数数据寄存器 RNG_DATA (偏移: 0E8h) .....	279
18.4	使用流程 .....	279
19	WDT .....	280
19.1	概述.....	280
19.2	主要特性 .....	280
19.3	寄存器描述.....	280
19.3.1	装载寄存器 WDT_LOAD (偏移: 00h) .....	281
19.3.2	计数寄存器 WDT_CNT (偏移: 04h) .....	281
19.3.3	控制寄存器 WDT_CTRL (偏移: 08h) .....	281
19.3.4	清除寄存器 WDT_CLR (偏移: 0ch) .....	281
19.3.5	RAW 中断状态寄存器 WDT_INTRAW (偏移: 10h) .....	282
19.3.6	MASK 中断状态寄存器 WDT_MINTS (偏移: 14h) .....	282
19.3.7	STALL 控制寄存器 WDT_STALL (偏移: 18h) .....	282
19.3.8	LOCK 寄存器 WDT_LOCK (偏移: 1ch) .....	282
19.4	使用流程 .....	283
19.4.1	WDT 定时器配置.....	283
19.4.2	WDT 喂狗流程配置.....	283
20	WWDT.....	284
20.1	概述.....	284
20.2	主要特性 .....	284
20.3	寄存器描述.....	284
20.3.1	控制寄存器 WWDT_CON (偏移: 00h) .....	285
20.3.2	配置寄存器 WWDT_CFG (偏移: 04h) .....	285
20.3.3	计数寄存器 WWDT_CNT (偏移: 08h) .....	285
20.3.4	中断使能寄存器 WWDT_IE (偏移: 0ch) .....	286

20.3.5	中断标志寄存器 WWDT_IF (偏移: 10h)	286
20.4	使用流程	286
21	ADC	287
21.1	概述	287
21.2	主要特性	287
21.3	ADC 管脚分布	288
21.4	寄存器描述	288
21.4.1	ADC 中断和状态寄存器 (ADC_ISR) (偏移: 000h)	289
21.4.2	ADC 中断使能寄存器 (ADC_IER) (偏移: 004h)	291
21.4.3	ADC 控制寄存器 (ADC_CR) (偏移: 008h)	292
21.4.4	ADC 配置寄存器 (ADC_CFGR) (偏移: 00Ch)	292
21.4.5	ADC 采样时间控制寄存器 (ADC_SMTR) (偏移: 010h)	294
21.4.6	ADC 通道控制寄存器 (ADC_CHER) (偏移: 014h)	295
21.4.7	ADC 通道选择寄存器 (ADC_SEQ) (偏移: 018h)	296
21.4.8	ADC 模拟看门狗阈值寄存器 (ADC_HLTR) (偏移: 020h)	296
21.4.9	ADC 注入通道控制寄存器 (ADC_IJCHER) (偏移: 024h)	296
21.4.10	ADC 注入通道选择寄存器 (ADC_IJSEQ) (偏移: 028h)	297
21.4.11	ADC 数据寄存器 0 (ADC_DR0) (偏移: 030h)	297
21.4.12	ADC 数据寄存器 1 (ADC_DR1) (偏移: 034h)	298
21.4.13	ADC 数据寄存器 2 (ADC_DR2) (偏移: 038h)	298
21.4.14	ADC 数据寄存器 3 (ADC_DR3) (偏移: 03Ch)	298
21.4.15	ADC 数据寄存器 4 (ADC_DR4) (偏移: 040h)	298
21.4.16	ADC 数据寄存器 5 (ADC_DR5) (偏移: 044h)	298
21.4.17	ADC 数据寄存器 6 (ADC_DR6) (偏移: 048h)	298
21.4.18	ADC 数据寄存器 7 (ADC_DR7) (偏移: 04Ch)	299
21.4.19	ADC 注入通道数据寄存器 0 (ADC_IJDR0) (偏移: 050h)	299
21.4.20	ADC 注入通道数据寄存器 1 (ADC_IJDR1) (偏移: 054h)	299
21.4.21	ADC 注入通道数据寄存器 2 (ADC_IJDR2) (偏移: 058h)	299
21.4.22	ADC 注入通道数据寄存器 3 (ADC_IJDR3) (偏移: 05Ch)	299
21.4.23	ADC 过采样平均结果寄存器 (ADC_DR_OVS) (偏移: 060h)	299
21.4.24	ADC 硬件触发使能配置寄存器 (ADC_HDT) (偏移: 0A8h)	300
21.4.25	ADC 注入通道硬件触发使能配置寄存器 (ADC_IJHDT) (偏移: 0ACh)	302
21.4.26	ADC 通道采样时间配置寄存器 0 (ADC_SMTR0) (偏移: 0B0h)	304
21.4.27	ADC 通道采样时间配置寄存器 1 (ADC_SMTR1) (偏移: 0B4h)	305
21.4.28	ADC 注入通道采样时间配置寄存器 (ADC_IJSMTR) (偏移: 0B8h)	306
21.5	ADC 使用流程	308
21.5.1	单次扫描模式单通道 A/D 转换	308
21.5.2	单次扫描模式多通道 A/D 转换	308

21.5.3	连续扫描模式单通道 A/D 转换 .....	309
21.5.4	连续扫描模式多通道 A/D 转换 .....	310
21.5.5	硬件触发事件 A/D 转换 .....	311
21.5.6	注入通道普通注入模式 A/D 转换 .....	312
21.5.7	注入通道自动注入模式 A/D 转换 .....	313
21.5.8	注意事项 .....	314
21.6	ADC 经 PGA 缓冲采样使用流程 .....	315
21.6.1	ADC 经 PGA 缓冲采样图 .....	315
21.6.2	ADC 经 PGA 缓冲采样流程图 .....	315
21.6.3	ADC 经 PGA 缓冲后采样使用流程 .....	315
22	COMP .....	316
22.1	概述 .....	316
22.2	主要特性 .....	316
22.3	COMP 功能框图 .....	317
22.4	COMP 滤波模式 .....	318
22.5	Window 功能 .....	319
22.6	寄存器描述 .....	319
22.6.1	COMP0 控制寄存器 COMP0_CFG (偏移: 118h) .....	319
22.6.2	COMP1 控制寄存器 COMP1_CFG (偏移: 11Ch) .....	321
22.6.3	COMP0 轮询寄存器 COMP0_POLL (偏移: 128h) .....	322
22.6.4	COMP1 轮询寄存器 COMP1_POLL (偏移: 12Ch) .....	323
22.6.5	COMP0 轮询 MASK 寄存器 COMP0_POLLMASK (偏移: 138h) .....	324
22.6.6	COMP1 轮询 MASK 寄存器 COMP1_POLLMASK (偏移: 13Ch) .....	324
22.6.7	COMP0 滤波设置寄存器 COMP0_LVSET (偏移: 148h) .....	324
22.6.8	COMP1 滤波设置寄存器 COMP1_LVSET (偏移: 14Ch) .....	325
22.6.9	COMP0 WINDOW 设置寄存器 COMP0_WINCFG (偏移: 158h) .....	325
22.6.10	COMP1 WINDOW 设置寄存器 COMP1_WINCFG (偏移: 15Ch) .....	327
22.6.11	COMP 状态寄存器 COMP0_INTSTAT (偏移: 168h) .....	330
22.6.12	COMP 状态寄存器 COMP1_INTSTAT (偏移: 16Ch) .....	330
22.6.13	COMP0 中断使能寄存器 COMP0_INTEN (偏移: 178h) .....	331
22.6.14	COMP1 中断使能寄存器 COMP1_INTEN (偏移: 17Ch) .....	332
22.7	使用流程 .....	332
23	DIV .....	333
23.1	概述 .....	333
23.2	主要特性 .....	333
23.3	寄存器描述 .....	333
23.3.1	被除数寄存器 DIV_DIVIDEND (偏移: 00h) .....	333

23.3.2	除数寄存器 DIV_DIVISOR (偏移: 04h)	334
23.3.3	余数寄存器 DIV_REMAIN (偏移: 08h)	334
23.3.4	商寄存器 DIV_QUOTIENT (偏移: 0Ch)	334
23.3.5	状态寄存器 DIV_STATUS (偏移: 10h)	334
23.4	使用流程	334
24	SQRT	335
24.1	概述	335
24.2	主要特性	335
24.3	寄存器描述	335
24.3.1	SQRT 被开方数据寄存器 (SQRT_VALUE) (偏移: 00h)	335
24.3.2	SQRT 开方运算结果寄存器 (SQRT_RESULT) (偏移: 04h)	335
24.3.3	SQRT 状态寄存器 (SQRT_SR) (偏移: 08h)	336
24.4	使用流程	336
25	SysTick	337
25.1	概述	337
25.2	寄存器描述	337
25.2.1	控制和状态寄存器 SysTick_CTRL (偏移: 00h)	338
25.2.2	重载值寄存器 SysTick_LOAD (偏移: 04h)	338
25.2.3	当前值寄存器 SysTick_VAL (偏移: 08h)	338
25.3	使用流程	339
26	调试支持 (DBG)	340
27	版本维护	341

## 图目录

图 3-1: 总线架构图.....	8
图 3-2: 存储器地址映射图.....	9
图 4-1: Cortex-M0+处理器功能框图 .....	12
图 4-2: Cortex-M0+的寄存器组 .....	12
图 5-1: 时钟模块框图.....	13
图 6-1: eFlash 存储区域.....	58
图 6-2: EEPROM8 关键字配置图 .....	59
图 6-3: 写操作流程.....	67
图 6-4: Sector 擦除操作流程 .....	68
图 6-5: Chip 擦除操作流程 .....	68
图 10-1: 主机发送流程图 .....	95
图 10-2: 主机接收流程图 .....	96
图 10-3: 从机发送流程图 .....	97
图 10-4: 从机接收流程图 .....	98
图 12-1: ATIMER 结构框图 .....	112
图 12-2: 预分频从 1 变为 2 的波形 .....	113
图 12-3: 预分频从 1 变为 4 的波形 .....	114
图 12-4: 向上计数波形, 内部时钟不分频.....	115
图 12-5: 向上计数波形, 内部时钟 2 分频 .....	115
图 12-6: ARPE=0 (ATIMER_ARR 没有预装载) 时的更新事件 .....	116
图 12-7: ARPE=1 (ATIMER_ARR 预装载) 时的更新事件.....	116
图 12-8: 向下计数, 内部时钟不分频.....	117
图 12-9: 向下计数, 内部时钟 2 分频.....	118
图 12-10: ARPE=0 (ATIMER_ARR 没有预装载) 时的更新事件 .....	118
图 12-11: ARPE=1 (ATIMER_ARR 预装载) 时的更新事件.....	119
图 12-12: 中心对齐计数器时序图, ATIMER_PSC=0, ATIMER_ARR=0x6 .....	120
图 12-13: 计数器时序图, ARPE=1 时的更新事件 (计数器下溢) .....	120
图 12-14: 计数器时序图, ARPE=1 时的更新事件 (计数器溢出) .....	121
图 12-15: 不同模式下更新速率的例子, 及 ATIMER_RCR 的寄存器设置 .....	122
图 12-16: 内部时钟源模式, 时钟分频因子为 1 .....	123
图 12-17: 外部时钟连接例子.....	124
图 12-18: 外部时钟模式 1 下的时序 .....	124
图 12-19: 外部时钟模式 1 下的时序 .....	125
图 12-20: 外部触发输入框图.....	125
图 12-21: 外部时钟模式 2 下的时序 1 .....	126
图 12-22: 外部时钟模式 2 下的时序 .....	127

图 12-23: 捕获/比较通道 (通道 1 输入部分) .....	128
图 12-24: 捕获/比较通道 1 的主电路 .....	128
图 12-25: 捕获/比较通道的输出部分 (通道 1 至 3) .....	128
图 12-26: 捕获/比较通道的输出部分 (通道 4) .....	129
图 12-27: PWM 输入捕获模式时序 .....	130
图 12-28: 输出比较模式, 翻转 OC1 .....	131
图 12-29: 边沿对齐的 PWM 波形 (ARR=7) .....	132
图 12-30: 中央对齐的 PWM 波形 (ARR=7) .....	133
图 12-31: 移相模式的 PWM 波形 (ARR=7) .....	134
图 12-32: 带死区插入的互补输出 .....	134
图 12-33: 死区波形延迟大于负脉冲 .....	135
图 12-34: 死区波形延迟大于正脉冲 .....	135
图 12-35: 响应刹车的输出 .....	136
图 12-36: 产生六步 PWM, 使用 COM 的例子 (OSSR=1) .....	137
图 12-37: 单脉冲模式的例子 .....	138
图 12-38: ETR 信号清除 ATIMER 的 OCxREF .....	139
图 12-39: 编码器模式下的计数器操作实例 .....	140
图 12-40: 复位模式下的时序 .....	141
图 12-41: 门控模式下的时序 .....	142
图 12-42: 触发器模式下的时序 .....	143
图 12-43: 外部时钟模式 2 和触发模式下的时序 .....	144
图 13-1: GTIMER 结构框图 .....	178
图 13-2: GTIMER 预分频从 1 变为 2 的波形图 .....	179
图 13-3: 向上计数波形图 .....	180
图 13-4: 向下计数图 .....	181
图 13-5: 中心对齐计数器时序图 .....	181
图 13-6: APBCLK 时钟源模式, 时钟分频因子为 1 .....	182
图 13-7: 捕获/比较通道 (通道 1 输入部分) .....	184
图 13-8: 捕获/比较通道 1 的主电路 .....	184
图 13-9: 捕获/比较通道的输出部分 .....	185
图 13-10: PWM 输入捕获 .....	186
图 13-11: 输出比较模式, 翻转 OC1 .....	187
图 13-12: 边沿对齐的 PWM 波形 (ARR=7) .....	188
图 13-13: 中央对齐的 PWM 波形 (APR=7) .....	189
图 13-14: 单脉冲模式的例子 .....	190
图 13-15: GTIMER_ETR 信号清除 GTIMER 的 OCxREF .....	192
图 13-16: 编码器模式下的计数器操作实例 .....	193
图 13-17: 复位模式下的时序 .....	194

图 13-18: 门控模式下的时序图 .....	195
图 13-19: 触发器模式下的时序 .....	196
图 14-1: GTIMER 结构框图 .....	216
图 14-2: GTIMER 预分频从 1 变为 2 的波形图 .....	217
图 14-3: 向上计数波形图 .....	218
图 14-4: 向下计数图 .....	219
图 14-5: 中心对齐计数器时序图 .....	219
图 14-6: APBCLK 时钟源模式, 时钟分频因子为 1 .....	220
图 14-7: 捕获/比较通道 (通道 1 输入部分) .....	222
图 14-8: 捕获/比较通道 1 的主电路 .....	222
图 14-9: 捕获/比较通道的输出部分 .....	223
图 14-10: PWM 输入捕获 .....	224
图 14-11: 输出比较模式, 翻转 OC1 .....	225
图 14-12: 边沿对齐的 PWM 波形 (ARR=7) .....	226
图 14-13: 中央对齐的 PWM 波形 (APR=7) .....	227
图 14-14: 单脉冲模式的例子 .....	229
图 14-15: GTIMER_ETR 信号清除 GTIMER 的 OCxREF .....	230
图 14-16: 编码器模式下的计数器操作实例 .....	231
图 14-17: 复位模式下的时序 .....	232
图 14-18: 门控模式下的时序图 .....	233
图 14-19: 触发器模式下的时序 .....	234
图 15-1: LPTIMER 结构框图 .....	255
图 21-1: ADC 经 PGA 缓冲采样图 .....	315
图 21-2: ADC 经 PGA 缓冲采样流程图 .....	315
图 22-1: COMP 模拟功能框图 .....	317
图 22-2: COMP 整体功能框图 .....	318

## 表目录

表 3-1: 存储器和模块地址分配 .....	10
表 5-1: 系统时钟选择 .....	13
表 5-2: 系统复位源 .....	14
表 5-3: 低功耗模式 .....	16
表 5-4: 系统寄存器列表 .....	19
表 6-1: EFC 寄存器列表 .....	60
表 7-1: 中断源 .....	69
表 8-1: GPIO 寄存器列表 .....	71
表 9-1: UART0/1 寄存器列表 .....	78
表 10-1: I2C 寄存器列表 .....	84
表 11-1: SPI 寄存器列表 .....	100
表 11-2: SPI 两种引脚搭配方式表 .....	108
表 12-1: 输入信号源 .....	127
表 12-2: Encoder Interface 计数方式 .....	139
表 12-3: DMA 访问计数方式 .....	144
表 12-4: ATIMER 寄存器列表 .....	146
表 13-1: GTIMER0/1/2 内部信号扩展表 .....	183
表 13-2: 编码计数方式与输入信号 .....	192
表 13-3: GTIMER 寄存器列表 .....	197
表 14-1: GTIMER0/1/2 内部信号扩展表 .....	221
表 14-2: 编码计数方式与输入信号 .....	231
表 14-3: GTIMER 寄存器列表 .....	235
表 15-1: LPTIMER 寄存器列表 .....	257
表 17-1: CRC 寄存器列表 .....	276
表 18-1: RNG 寄存器列表 .....	278
表 19-1: WDT 寄存器列表 .....	280
表 20-1: WWDT 寄存器列表 .....	284
表 21-1: ADC 管脚分布 .....	288
表 21-2: ADC 寄存器列表 .....	288
表 22-1: COMP 寄存器列表 .....	319
表 23-1: DIV 寄存器列表 .....	333
表 24-1: SQRT 寄存器列表 .....	335
表 25-1: SysTick 寄存器列表 .....	337

# 1 文档约定

## 1.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

R/W	可读可写，软件可以读写该位
R	只读，软件只能读取该位
W	只写，软件只能写入该位，软件读返回复位值
RC	读清 0
R/W1C	可读，写 1 清零软件可以读取该位。写 1 清除该位，写 0 无效
R/W0C	可读，写 0 清除，软件可以读取该位。写 0 清除该位，写 1 无效
RSV	保留位，必须保持复位值

## 1.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- CPU 内核集成了 SWD 调试端口 (SWD-DP)：提供基于串行线调试 (SWD) 协议的 2 引脚（时钟和数据）接口。
- 字：32 位数据/指令。
- 半字：16 位数据/指令。
- 字节：8 位数据。
- 双字：64 位数据。
- IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
- ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。
- 选项字节：存储于 Flash 中的产品配置位。
- OBL：选项字节加载器。
- AHB：高级高性能总线。
- APB：低速外设总线

## 2 产品简介

### 2.1 系统概述

UM32Mx56 系列芯片是广芯微电子（广州）股份有限公司研制的基于 ARM Cortex-M0+内核的 Low Pin Count、宽电压工作范围的 32 位处理器 SoC 芯片系列，重点面向汽车电动化、智能家居、电机控制及工业控制等应用场景。依据行业应用场景的具体应用需求，芯片内部集成了 12 位 SAR ADC、UART、SPI、I2C 等通用外围通讯接口，ADC、PGA、COMP 等传感获取接口，LPTIMER、WDT 等超低功耗模块接口，以及 SQRT、DIV 除法器硬件算法模块。具有高整合度、高抗干扰、高可靠性和超低功耗等技术特点。内置 RC 高频和低频振荡器，支持免晶振应用。支持 Keil MDK 集成开发环境，支持 C 语言和汇编语言进行软件开发。

UM32Mx56 系列芯片增加了芯片内数据防盗取等安全特性，增加了 SWD 读保护、SWD 接口物理禁止等功能。芯片内部 eFlash 存储器增加 ECC 纠错码和数据加密功能，ECC 纠错可以有效防止 eFlash 数据由于环境因素丢失或偶发读错误的现象。此外，芯片内部增加对部分 EEPROM 区和 32KB eFlash 主存储区的防擦写机制，可以有效防止客户应用数据、程序被改写。

#### 典型应用场景：

- 电机控制
- 工业物联网应用
- 智能交通，智慧城市，智能家居
- 智能门锁，资产追踪、无线监控等智能传感器终端应用

## 2.2 主要特性

- **超低功耗电源管理系统**
  - 低功耗模块 LPTIMER、WDT
  - 内置 ROSC/LDO/POR, 可免晶振/LDO/复位电路
- **处理器**
  - 32 位 ARM Cortex-M0+, 系统最高主频 64MHz
  - 单周期硬件乘法器
- **存储器**
  - 32KB FLASH
  - 4KB EEPROM
  - 6KB SRAM
  - Sector 大小: 512B, 擦写次数: 100,000 次
  - 数据保存时间: 10 年@常温
- **时钟**
  - 内部高速时钟 64MHz
  - 内部低速时钟 32kHz
- **模拟外设**
  - 1 个 12 位 1Msps ADC 转换器, 多达 16 通道
  - 3 个运算放大器
  - 2 个电压比较器
- **通信接口**
  - 最大 2 路 UART 串口
  - 1 路通用的 SPI 接口
  - 1 路 I2C 接口

- **定时器**

- 1 个 16 位高级定时器 ATIMER 支持输入捕获、死区互补 PWM 输出
- 1 个 32 位通用定时器、2 个 16 位通用定时器 GTIMER 支持输入捕获、PWM 输出
- 1 个 32 位低功耗定时器 LPTIMER 支持 2 路 PWM 输出
- 1 个 32 位低功耗看门狗 WDT，可复位/中断
- 1 个 18 位窗口看门狗 WWDI，可复位/中断
- 最大共支持 21 路 PWM 输出

- **GPIO 通用输入/输出端口**

- 最大 26 个通用输入/输出管脚
- 支持边沿/电平触发中断
- 16/8mA 两档驱动能力可配置

- **安全功能**

- 低电压检测 LVD
- 掉电复位 LVR，防死机设计
- CRC16-CCITT 数据校验算法硬件加速
- RNG 硬件随机数发生器
- 16 字节全球唯一芯片序列号 ID

- **硬件加速引擎**

- 除法器 DIV，8 个周期出结果
- 开方运算，4 个周期出结果

- **支持 SIP 预驱**

- UM32xxx
  - ✓ 内置 6 路 NMOS Driver
  - ✓ 栅极驱动电压从 8V 到 20V
  - ✓ 输出级拉电流/灌电流能力 1.5A/1.8A

- UM32xxx
  - ✓ 内置 6 路 PMOS+NMOS Driver
  - ✓ 内置 5V/40mA LDO
  - ✓ 栅极驱动电压从 5V 到 30V
  - ✓ 输出级拉电流/灌电流能力 50mA/300mA
- **主要电气参数**
  - 工作电压: 2.3 ~ 5.5V
  - 工作温度: -40 ~ +105°C
  - ESD 防护: ±8kV (HBM)
- **开发支持**
  - IAP 应用程序更新
  - JTAG->SWD 模式在线调试/下载功能
  - 完整 SDK 开发包、EVB 硬件开发套件
  - 离线烧录器

## 3 存储器和总线架构

### 3.1 系统架构

主系统由以下部分构成:

- 2个AHB总线Masters:
  - Cortex-M0+
  - DMA控制器
- 5个AHB总线Slaves:
  - FLASH存储器
  - SRAM存储器
  - AHB, AHB to APB Bridge, 包含所有APB接口外设和所有AHB接口外设
  - DIV除法器
  - SQRT开方运算器

### 3.2 总线架构图

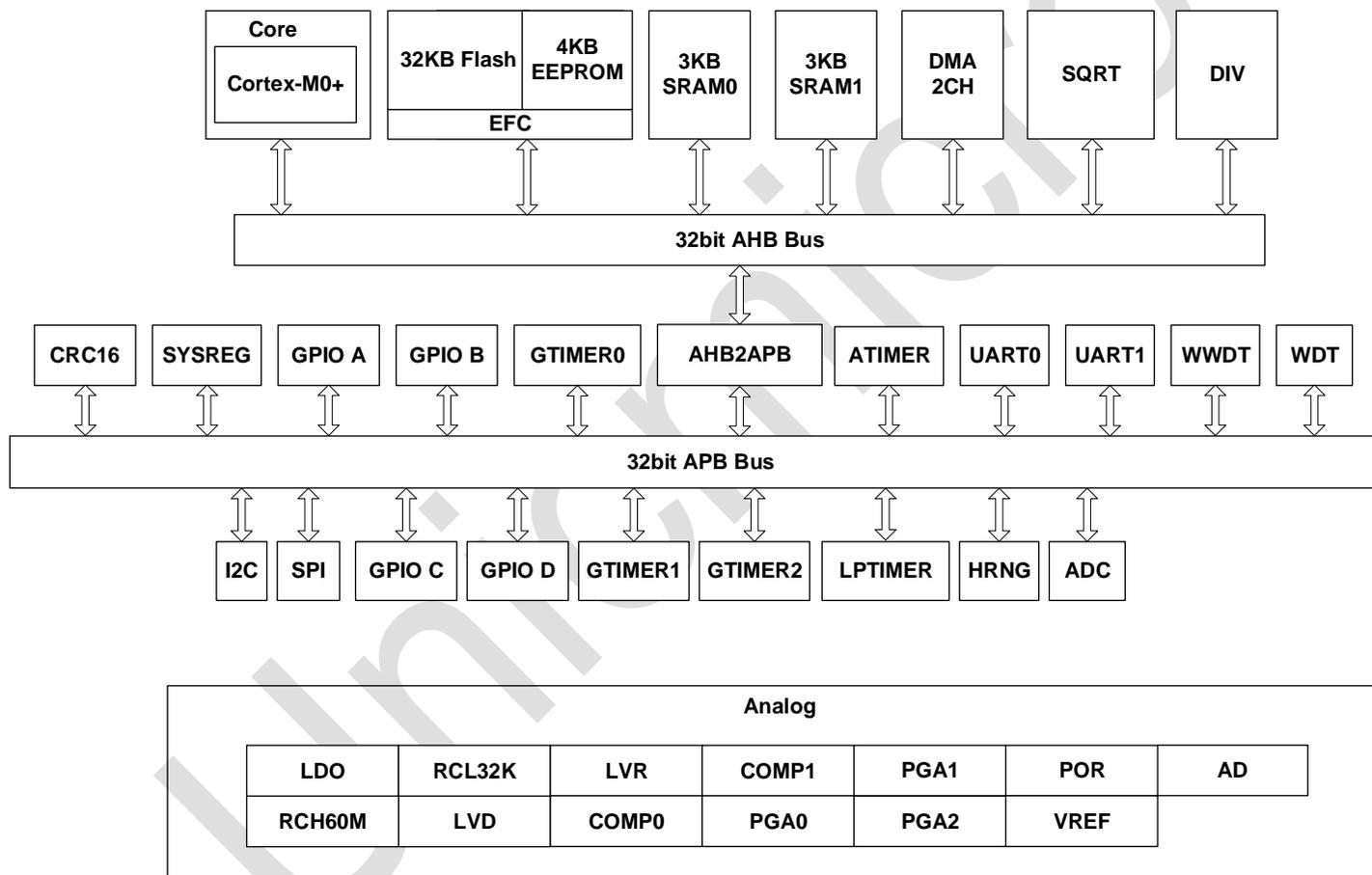


图 3-1: 总线架构图

### 3.3 存储器映射 (Memory Mapping)

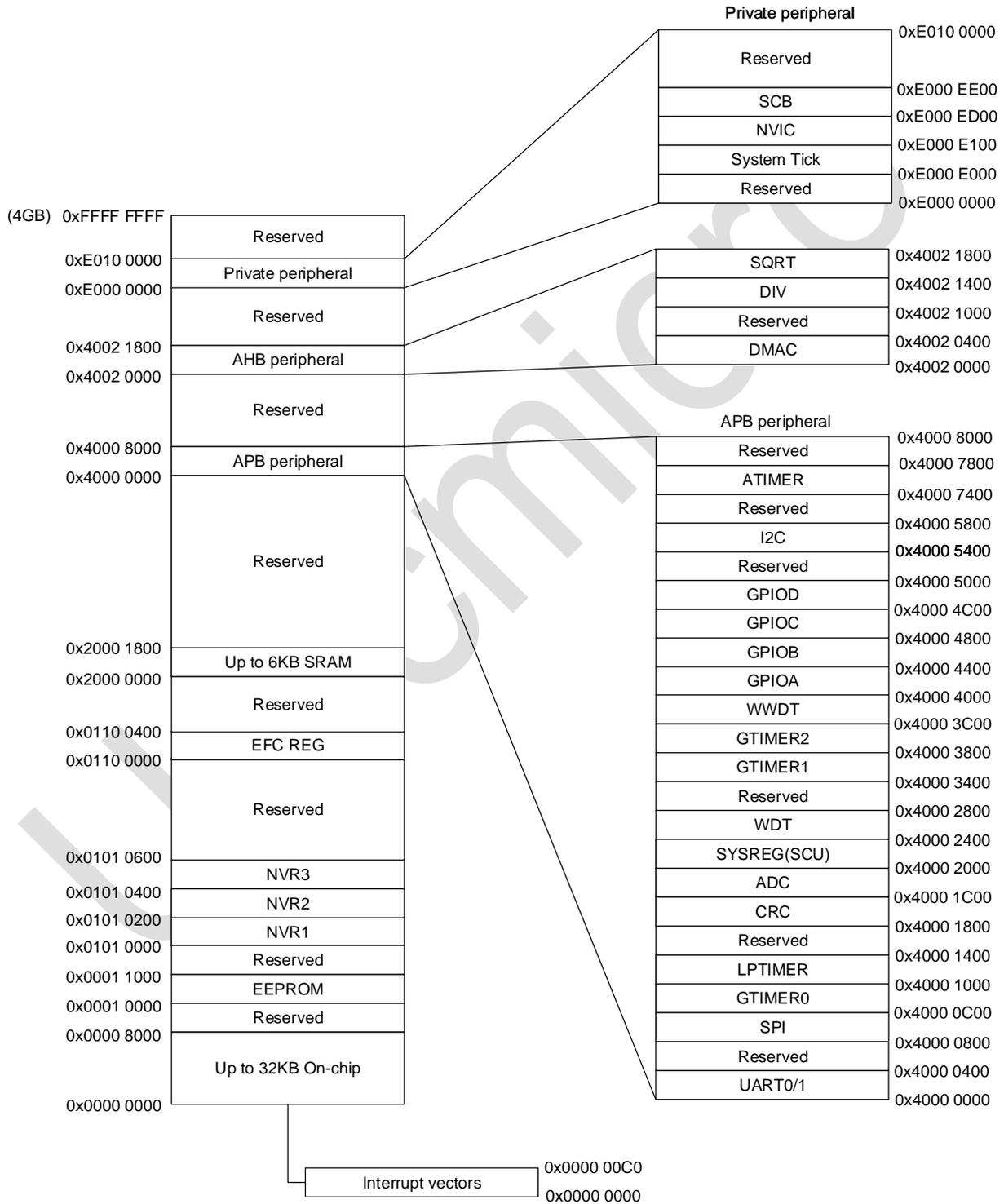


图 3-2: 存储器地址映射图

表 3-1: 存储器和模块地址分配

模块名	地址分配	大小
EFLASH	0x0000_0000—0x0000_8000	32kBytes
EEPROM	0x0001_0000—0x0001_1000	4KBytes
NVR1	0x0101_0000—0x0101_0200	512Bytes
NVR2	0x0101_0200—0x0101_0400	512Bytes
NVR3	0x0101_0400—0x0101_0600	512Bytes
EFC Register	0x0110_0000—0x0110_006C	108Bytes
SRAM1	0x2000_0000—0x2000_0C00	3kBytes
SRAM2	0x2000_0C00—0x2000_1800	3kBytes
UART0/1	0x4000_0000—0x4000_0400	1kBytes
SPI	0x4000_0800—0x4000_0C00	1kBytes
GTIMER0	0x4000_0C00—0x4000_1000	1kBytes
LPTIMER	0x4000_1000—0x4000_1400	1kBytes
CRC	0x4000_1800—0x4000_1C00	1kBytes
ADC	0x4000_1C00—0x4000_2000	1kBytes
SYSREG(SCU)	0x4000_2000—0x4000_2400	1kBytes
WDT	0x4000_2400—0x4000_2800	1kBytes
GTIMER1	0x4000_3400—0x4000_3800	1kBytes
GTIMER2	0x4000_3800—0x4000_3C00	1kBytes
WWDT	0x4000_3C00—0x4000_4000	1kBytes
GPIOA	0x4000_4000—0x4000_4400	1kBytes
GPIOB	0x4000_4400—0x4000_4800	1kBytes
GPIOC	0x4000_4800—0x4000_4C00	1kBytes
GPIOD	0x4000_4C00—0x4000_5000	1kBytes
I2C	0x4000_5400—0x4000_5800	1kBytes
ATIMER	0x4000_7400—0x4000_7800	1Kbytes
DMAC	0x4002_0000—0x4002_0C00	3kBytes
DIV	0x4002_1000—0x4002_1400	1kBytes
SQRT	0x4002_1400—0x4002_1800	1kBytes

## 4 处理器

### 4.1 概述

Cortex™ M0+处理器是 32 位的两级流水线 RISC 处理器，内嵌 AMBA-Lite 接口和嵌套向量中断控制器 (NVIC)。具有硬件调试功能，可以执行 Thumb 指令，并与其它 Cortex-M 系列兼容。同时加入多项全新设计，改进调试和追踪能力、减少每条指令循环 (IPC) 数量和改进 Flash 访问的两级流水线等，更纳入了节能降耗技术。Cortex M0+处理器全面支持已整合 Keil & IAR 调试器。

### 4.2 主要特性

- ARMv6 M Thumb
- Thumb/Thumb 2 技术
- ARMv6 M 兼容 24 位 SysTick 定时器
- 32 位硬件乘法器
- 系统接口支持小端 (little-endian) 数据访问
- 准确而及时的中断处理能力
- 加载、存储多个数据和多周期乘法指令可被终止然后重新开始从而实现快速中断处理
- C 应用程序二进制接口的异常兼容模式 (C-ABI); ARMv6 M 的模式允许用户使用纯 C 函数实现中断处理
- 使用中断唤醒 (WFI) 与事件唤醒 (WFE) 指令进入低功耗的休眠模式，或从中断退出休眠模式

### 4.3 功能框图

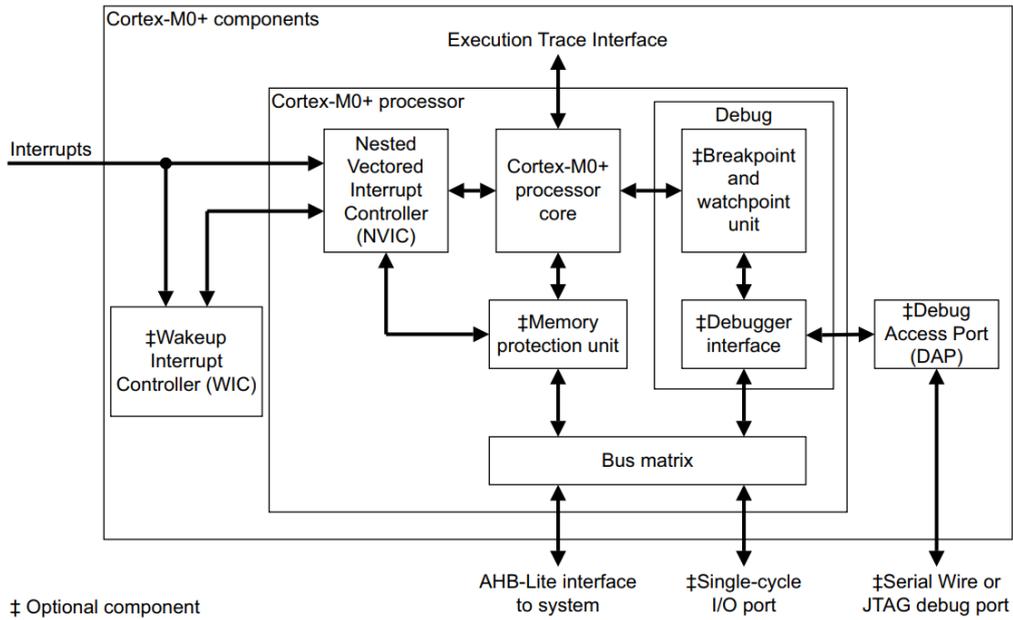


图 4-1: Cortex-M0+处理器功能框图

### 4.4 内核寄存器组

Cortex-M0+处理器寄存器组如下图:

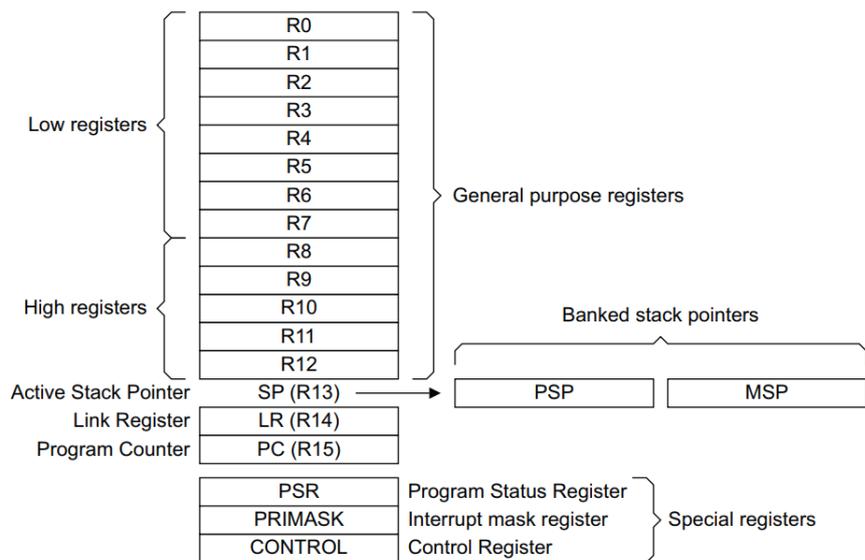


图 4-2: Cortex-M0+的寄存器组

## 5 系统配置 (SCU)

### 5.1 时钟框图

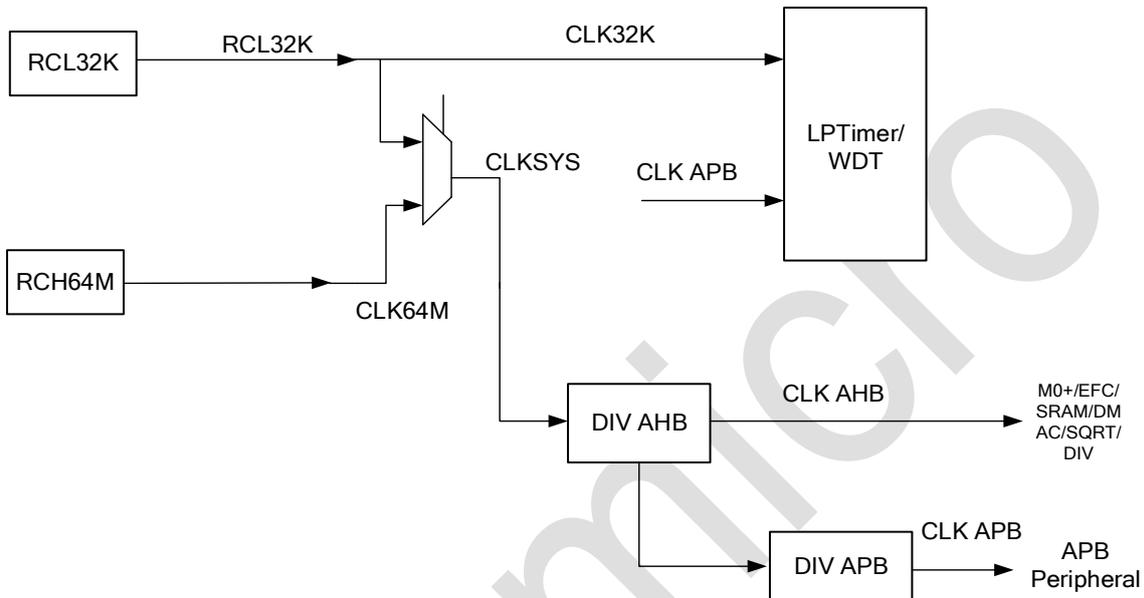


图 5-1: 时钟模块框图

### 5.2 时钟选择

系统存在 2 个时钟源:

- 64MHz 高精度内部时钟 RCH，作为系统时钟源。
- 32kHz 的内部时钟 RCL，作为低功耗时钟，可作为系统时钟源

根据工作模式不同，采用不同时钟方案，通过配置系统控制寄存器 0 (SYSCTRL0) [14]位 CLK\_SEL 来选择系统时钟的来源。关系如下表所示:

表 5-1: 系统时钟选择

CLK_SEL	系统时钟来源
0	RCH
CLK_SEL	系统时钟来源
1	RCL

## 5.3 复位源

芯片有多个复位源, 包括 POR 复位, RESETEEN 复位, WDT 复位, WWDT 复位, SOFT\_RESETEEN 复位, 模块软件复位, LVD 复位, LOCKUP 复位。具体复位源如下表:

表 5-2: 系统复位源

复位源	描述
内部模拟 POR 上电复位	复位所有
LVR 复位	
RESETEEN 复位	复位除 CPU DEBUG 逻辑外的所有
LOCKUP 复位	复位除EFC以外的其它逻辑
LVD 复位	
WDT	
WWDT	
SOFT_RESETEEN	
各模块复位	复位对应 IP 模块

### 5.3.1 内部 POR 上电复位

内部上电复位 POR: 无条件复位整个芯片。

### 5.3.2 LVR 复位

LVR 复位: 无条件复位整个芯片。

### 5.3.3 RESETEEN 复位

外部复位 RESETEEN 复位除 CPU DEBUG 逻辑外的整个芯片。在 RESETEEN 复位状态时, 芯片可以连接 SWD 接口。RESETEEN 管脚在上电完成后默认作为外部复位, 可以通过软件关闭外部复位功能。

### 5.3.4 LOCKUP 复位

当系统连续发生两次 HardFault 时, CPU 会进入 LOCKUP 状态, 系统会产生 LOCKUP 复位。  
LOCKUP 复位除 EFC 以外的其它逻辑。

### 5.3.5 LVD 复位

LVD 复位除 EFC 以外的其它逻辑。

### 5.3.6 WDT 复位

看门狗定时器复位除 EFC 外的整个用户电路, 缺省为释放状态。

当软件未能有效阻止超时事件时, 看门狗定时器复位。该复位仅发生在软件无法正常执行, 可能破坏数据时。在 CPU 处于 HALT 状态时, WDT 停止计数, 不会产生复位信号。

### 5.3.7 WWDT 复位

复位除 EFC 以外的其它逻辑。

### 5.3.8 SOFT\_RESETN 复位

此复位由系统产生。系统可以通过写软复位重启, 但不复位 EFC 控制器和 IO 相关设置。

### 5.3.9 模块复位

模块复位, 通过软件复位各数字模块。

## 5.4 低功耗模式

芯片除正常工作模式外, 为了降低芯片的电流消耗, 提供三种低功耗模式: 休眠 (Sleep) 模式、

深度休眠 (Deepsleep) 模式、停止 (Stop) 模式。

在休眠模式下, CM0+停止工作, 保留中断处理功能。其它外设等模块时钟和复位可由软件设置。

休眠模式由 M0+特定指令 WFI / WFE 进入, 唤醒由中断触发。

深度休眠模式是休眠模式的升级, 在此模式下, CM0+停止运行, 高速时钟停止运行, 低功耗功能模块 (LPTIMER、WDT) 可以运行。深度休眠模式要先设置 CM0+内部的 DEEPSLEEP 寄存器, 然后由 M0+特定指令 WFI / WFE 进入, 唤醒由中断触发。

停止模式下, 高速时钟和低速时钟均停止运行, 系统无任何运行的时钟, 一切外围模块均停止运行。上电复位信号有效, IO 状态保持, IO 中断有效, 所有寄存器, RAM 和 CPU 数据保存状态时的功耗; 停止模式要先设置系统寄存器中 STOPMODE\_SEL 寄存器和 CM0+内部的 DEEPSLEEP 寄存器, 然后由 M0+特定指令 WFI / WFE 进入, 唤醒可以通过 GPIO 边沿/电平唤醒或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒。

详细的描述如下表:

表 5-3: 低功耗模式

模式	模式描述	进入条件	退出条件
Sleep	LDO Active 供电, CPU 大部分休眠 (包括 NVIC), WIC 不休眠; 软件可关闭各模块时钟。	<ol style="list-style-type: none"> <li>1. 根据需要, 关闭各外设模块时钟, 仅留下需要监测中断事件的模块。</li> <li>2. 执行 WFI/WFE 指令。</li> </ol>	<ol style="list-style-type: none"> <li>1. CM0+检测到中断或事件发生。</li> <li>2. 进入中断服务程序清中断并返回。</li> <li>3. 继续执行后续指令。</li> </ol>
DeepSleep	LDO Standby 供电, CPU 大部分休眠 (包括 NVIC), WIC 不休眠; 高速时钟源关闭, RCL 低速时钟源运行。	<ol style="list-style-type: none"> <li>1. 根据需要, 关闭各外设模块时钟, 仅留下需要监测中断事件的模块。</li> <li>2. 设置 CM0+内部的 DeepSleep 寄存器。</li> <li>3. 执行 WFI/WFE 指令。</li> </ol>	<ol style="list-style-type: none"> <li>1. CM0+检测到中断或事件发生。</li> <li>2. 进入中断服务程序清中断并返回。</li> <li>3. 继续执行后续指令。</li> </ol>
Stop	LDO Standby 供电, 关闭系统所有时钟。	<ol style="list-style-type: none"> <li>1. 根据需要, 设置 IO 唤醒的条件。</li> <li>2. 设置 CM0+内部的 DeepSleep 寄存器。</li> <li>3. 设置系统寄存器中的 STOPMODE_SEL 寄存器。</li> <li>4. 执行 WFI/WFE 指令。</li> </ol>	<ol style="list-style-type: none"> <li>1. 外部 IO 唤醒事件到来。</li> <li>2. CM0+检测到 IO 唤醒事件中断发生。</li> <li>3. 进入中断服务程序清中断并返回。</li> <li>4. 继续执行后续指令。</li> </ol>

低功耗模式的进入和唤醒条件阐述如下:

- Sleep, DeepSleep, Stop 模式进入条件都需要调用 WFI/WFE; 三者模式的唤醒本质是都需要产生中断或者事件发生。
- Sleep 模式下, 高速时钟 RCH 以及内部低速时钟 RCL 维持进入低功耗模式前的设置, 只要系统产生中断就可以唤醒退出。
- DeepSleep 模式下, RCH 时钟关闭了, RCL 时钟根据设置在工作, 所以只有工作在低频时钟源的模块如 WDT、LPTIMER 可以产生中断唤醒退出, 以及 GPIO 边沿/电平模式在无时钟情况下产生中断唤醒退出。
- Stop 模式下, 所有时钟源关闭, 可以通过 GPIO 边沿/电平模式在无时钟情况下产生中断唤醒退出, 或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒退出。

### 5.4.1 Sleep 模式

进入 Sleep 模式设置:

- SCB->SCR 的 bit2 配置为 0
- 调用 WFI/WFE 进入 sleep 模式

唤醒条件: 中断唤醒。

### 5.4.2 DeepSleep 模式

DeepSleep 唤醒源包括: WDT、LPTIMER 中断唤醒, GPIO 边沿/电平中断唤醒。

下面以 PD3 管脚下降沿唤醒为例阐述配置流程:

1. 配置外围模块时钟控制寄存器 PERI\_CLKEN[19], 使能 GPIOD 时钟。
2. 配置外围模块复位控制寄存器 PERI\_RESET[19], GPIOD 设置正常工作。
3. 配置端口 PD 功能寄存器 PD\_SEL[15:12], 配置 PD3 为 GPIO 功能。
4. 配置 GPIOD 数据方向寄存器 GPIO\_DIR[3], PD3 为输入。
5. 配置 GPIOD 中断触发模式寄存器 GPIO\_IS[3], PD3 边沿触发中断。

6. 配置GPIO中断边沿触发设置寄存器GPIO\_IBE[3], PD3单边触发。
7. 配置GPIO中断高低电平触发设置寄存器GPIO\_IEV, 下降沿触发。
8. 配置GPIO中断使能寄存器GPIO\_IEN[3], 使能PD3中断。
9. 配置端口输入配置寄存器PAD\_IE[27], PD3输入使能。
10. 配置端口上拉配置寄存器PAD\_PU[27], PD3上拉使能。
11. SCB -> SCR[2]配置为 1。
12. 调用 WFI 进入 Deepsleep 模式。
13. 中断产生后, 能将系统从深度休眠模式下唤醒。

### 5.4.3 Stop 模式

Stop 模式唤醒源包括: GPIO 边沿/电平中断唤醒, 或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒。

下面以 PD3 管脚下降沿唤醒为例阐述配置流程:

1. 配置外围模块时钟控制寄存器PERI\_CLKEN[19], 使能GPIO时钟。
2. 配置外围模块复位控制寄存器PERI\_RESET[19], GPIO设置正常工作。
3. 配置端口PD功能寄存器PD\_SEL[15:12], 配置PD3为GPIO功能。
4. 配置GPIO数据方向寄存器GPIO\_DIR[3], PD3为输入。
5. 配置GPIO中断触发模式寄存器GPIO\_IS[3], PD3边沿触发中断。
6. 配置GPIO中断边沿触发设置寄存器GPIO\_IBE[3], PD3单边触发。
7. 配置GPIO中断高低电平触发设置寄存器GPIO\_IEV, 下降沿触发。
8. 配置GPIO中断使能寄存器GPIO\_IEN[3], 使能PD3中断。
9. 配置端口输入配置寄存器PAD\_IE[27], PD3输入使能。
10. 配置端口上拉配置寄存器PAD\_PU[27], PD3上拉使能。
11. 配置STOPMODE\_SEL = 0xA5A50001; //STOP模式有效。
12. SCB -> SCR[2]配置为 1。

13. 调用\_WFI(); //进入STOP模式。
14. 中断产生后, 能将系统从Stop模式下唤醒。

## 5.5 系统寄存器

SYSREG (SCU) 寄存器基地址: 0x4000\_2000

表 5-4: 系统寄存器列表

偏置	名称	描述
0x000	SCU_CTRL0	系统控制寄存器 0
0x008	SCU_CTRL_PROT	系统控制保护寄存器
0x00C	SCU_OSC_CTRL	时钟控制寄存器
0x010	SCU_PERI_CLKEN	外设时钟门控寄存器
0x020	SCU_RESET_FLAG	复位标识寄存器
0x024	SCU_PERI_RESET	外围模块复位控制寄存器
0x028	SCU_EXTRST_CTRL	外部复位滤波控制寄存器
0x030	SCU_PA_SEL	端口 PA 功能配置寄存器, 只能外部复位和 POR 复位此寄存器
0x034	SCU_PB_SEL	端口 PB 功能配置寄存器, 只能外部复位和 POR 复位此寄存器
0x038	SCU_PC_SEL	端口 PC 功能配置寄存器, 只能外部复位和 POR 复位此寄存器
0x03C	SCU_PD_SEL	端口 PD 功能配置寄存器, 只能外部复位和 POR 复位此寄存器
0x054	SCU_PAD_ADS	端口数模配置寄存器, 只能外部复位和 POR 复位此寄存器
0x060	SCU_PAD_DR	端口驱动能力配置寄存器, 只能外部复位和 POR 复位此寄存器
0x06C	SCU_PAD_PU	端口上拉配置寄存器, 只能外部复位和 POR 复位此寄存器
0x078	SCU_PAD_PD	端口下拉配置寄存器, 只能外部复位和 POR 复位此寄存器
0x084	SCU_PAD_OD	端口开漏输出配置寄存器, 只能外部复位和 POR 复位此寄存器
0x090	SCU_PAD_CS	端口输入类型配置寄存器, 只能外部复位和 POR 复位此寄存器

偏置	名称	描述
0x09C	SCU_PAD_IE	端口输入配置寄存器，只能外部复位和POR 复位此寄存器
0x0A4	SCU_PAD_STATUS	端口输入电平寄存器
0x0A8	SCU_PAD_SR	端口速率配置寄存器，只能外部复位和POR 复位此寄存器
0x0B4	SCU_IOCTL_PROT	IO 控制保护寄存器
0x0B8	SCU_LVD_CFG	LVD 控制寄存器
0x0BC	SCU_PGA_CFG	PGA 控制寄存器
0x0D0	SCU_EXTRST_SEL	外部复位端口选择寄存器
0x0D4	SCU_STOP_SEL	停止模式选择寄存器
0x0D8	SCU_SOFT_RSTN	软件复位寄存器
0x0DC	SCU_VET_OFFSET	中断向量地址重映射寄存器
0x0FC	SCU_LVD_INTR	LVD 中断寄存器
0x188	SCU_VREF	VREF 控制寄存器

### 5.5.1 系统控制寄存器 0 SCU\_CTRL0 (偏移: 000h)

比特	名称	属性	复位值	描述
31:29	RSV	-	-	保留
28:27	CLKOUT_SEL	RW	0x0	IO 作为时钟输出时，输出时钟选择： 00: HCLK_OUT (系统时钟) 01: RCL 时钟 10/11: PCLK_OUT
26:17	RSV	-	-	保留
16	SWD_WACK_EN	RW	0x1	连接 ULINK 或者 JLINK 后，在 DEEPSLEEP 或者 STOP 模式下，唤醒系统的使能位： 1: 在上述条件下，使用 NMI 中断唤醒系统 0: 在上述条件下，不唤醒系统
15	RSV	-	-	保留
14	CLK_SEL	RW	0x0	系统时钟来源选择： 0: 高速时钟 CLK_SEL_HF 1: 低速时钟 CLK_SEL_LF
13:11	RSV	-	-	保留

比特	名称	属性	复位值	描述
10:9	PCLK_DIV	RW	0x0	PCLK 分频选择: 00: HCLK 01: HCLK/2 10: HCLK/4 11: HCLK/8
8:6	HCLK_DIV	RW	0x3	HCLK 分频选择: 000: SystemClk 001: SystemClk/2 010: SystemClk/4 011: SystemClk/8 100: SystemClk/16 101: SystemClk/32 110: SystemClk/64 111: SystemClk/128
5:3	RSV	-	-	保留
2	RCL_EN	RW	0x1	内部低速时钟 RCL 使能控制: 0: 关闭 1: 使能
1	RSV	-	-	保留
0	RCH_EN	RW	0x1	内部高速时钟 RCH 使能信号: 0: 关闭 1: 使能 注: 当系统进入 DeepSleep, 此高速时钟会自动关闭。

### 5.5.2 系统控制保护寄存器 SCU\_CTRL\_PROT (偏移: 008h)

比特	名称	属性	复位值	描述
31:0	SYSCTRL_PROTECT	RW	0x0	寄存器 SCU_CTRL0 写保护的控制寄存器。给此寄存器写 0xA5A5_5A5A, 启动寄存器 SCU_CTRL0 的写使能。给此寄存器写其他值, 关闭它们的写使能。SCU_CTRL0 寄存器配置完后, 它们的写使能会自动关闭。 读此寄存器返回 SCU_CTRL0 寄存器的写使能状态。 0: 写未使能 1: 写已经使能

### 5.5.3 时钟控制寄存器 SCU\_OSC\_CTRL (偏移: 0x00Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	WAKP_delay	RW	0x25A	系统从 DEEPSLEEP/STOP 模式唤醒, CLK 给出的时间延时。此寄存器的时间为系统时钟计数的时钟周期。
15:0	RSV	-	-	保留

### 5.5.4 外围模块时钟寄存器 SCU\_PERI\_CLKEN (偏移: 010h)

比特	名称	属性	复位值	描述
31	DIV_CLKEN	RW	0x0	DIV 模块时钟使能: 1: 使能 0: 关闭
30	UART1_CLKEN	RW	0x0	UART1 模块时钟使能: 1: 使能 0: 关闭
29:28	RSV	-	-	保留
27	ATIMER_CLKEN	RW	0x0	ATIMER 模块时钟使能: 1: 使能 0: 关闭
26	SQRT_CLKEN	RW	0x0	SQRT 模块时钟使能: 1: 使能 0: 关闭
25	WWDT_CLKEN	RW	0x0	WWDT 模块时钟使能: 1: 使能 0: 关闭
24:23	RSV	-	-	保留
22	DMA_CLKEN	RW	0x0	DMA 模块时钟使能: 1: 使能 0: 关闭
21:20	RSV	-	-	保留
19	GPIOD_CLKEN	RW	0x0	GPIOD 模块时钟使能: 1: 使能 0: 关闭
18	GPIOC_CLKEN	RW	0x0	GPIOC 模块时钟使能: 1: 使能 0: 关闭
17	GPIOB_CLKEN	RW	0x0	GPIOB 模块时钟使能: 1: 使能 0: 关闭
16	GPIOA_CLKEN	RW	0x0	GPIOA 模块时钟使能: 1: 使能 0: 关闭
15	I2C_CLKEN	RW	0x0	I2C 模块时钟使能: 1: 使能 0: 关闭

比特	名称	属性	复位值	描述
14	ADC_CLKEN	RW	0x0	ADC 控制器模块时钟使能: 1: 使能 0: 关闭
13	RSV	-	-	保留
12	WDT_CLKEN	RW	0x0	WDT 模块时钟使能: 1: 使能 0: 关闭
11	CRC_CLKEN	RW	0x0	CRC16 模块时钟使能: 1: 使能 0: 关闭
10	RSV	-	-	保留
9	GTIMER0_CLKEN	RW	0x0	GTIMER0 模块时钟使能: 1: 使能 0: 关闭
8	LPTIMER_CLKEN	RW	0x0	LPTIMER 模块时钟使能: 1: 使能 0: 关闭
7:5	RSV	-	-	保留
4	SPI_CLKEN	RW	0x0	SPI 模块时钟使能: 1: 使能 0: 关闭
3	GTIMER2_CLKEN	RW	0x0	GTIMER 2 模块时钟使能: 1: 使能 0: 关闭
2	GTIMER1_CLKEN	RW	0x0	GTIMER 1 模块时钟使能: 1: 使能 0: 关闭
1	RSV	-	-	保留
0	UART0_CLKEN	RW	0x0	UART0 模块时钟使能: 1: 使能 0: 关闭

### 5.5.5 复位标识寄存器 SCU\_RESET\_FLAG (偏移: 020h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	SYS_RSTREQ_FLAG	W1C	0x0	CPU 复位状态, 需要软件初始化和清除。 1: Cortex M0+系统复位发生 0: 无复位发生 此位只能被 PORN 和外部 RESETN 复位, 写 1 清 0。

比特	名称	属性	复位值	描述
6	LOCKUP_RSTN_FLAG	W1C	0x0	CPU 死锁复位状态，需要软件初始化和清除。 1: Lockup 复位发生 0: 无复位发生 此位只能被 PORN 和外部 RESETN 复位，写 1 清 0。
5	LVD_RSTN_FLAG	W1C	0x0	低电压复位状态，需要软件初始化和清除。 1: LVD 复位发生 0: 无复位发生 此位只能被 PORN 和外部 RESETN 复位，写 1 清 0。
4	WWDT_FLAG	W1C	0x0	窗口看门狗复位状态，需要软件初始化和清除。 1: WWDT 复位发生 0: 无复位发生 此位只能被 PORN 和外部 RESETN 复位，写 1 清 0。
3	WDT_FLAG	W1C	0x0	看门狗复位状态，需要软件初始化和清除。 1: WDT 复位发生 0: 无复位发生 此位只能被 PORN 和外部 RESETN 复位，写 1 清 0。
2	RESETN_FLAG	W1C	0x0	外部复位状态，需要软件初始化和清除。 1: 外部复位发生 0: 无复位发生 此位只能被 PORN 复位，写 1 清 0。
1	SOFTR_FLAG	W1C	0x0	Soft 复位状态，需要软件初始化和清除。 1: Soft 复位发生 0: 无复位发生 此位只能被 PORN 和外部 RESETN 复位，写 1 清 0。
0	RSV	-	-	保留

注：复位标识寄存器只能被 PORN 复位。

### 5.5.6 外围模块复位控制寄存器 SCU\_PERI\_RESET (偏移: 024h)

比特	名称	属性	复位值	描述
31	DIV_RESET	RW	0x0	DIV 模块复位使能: 1: 正常工作 0: 模块处于复位状态
30	UART1_RESET	RW	0x0	UART1 模块复位使能: 1: 正常工作 0: 模块处于复位状态
29:28	RSV	-	-	保留
27	ATIMER_RESET	RW	0x0	ATIMER 模块复位使能: 1: 正常工作 0: 模块处于复位状态
26	SQRT_RESET	RW	0x0	SQRT 模块复位使能: 1: 正常工作 0: 模块处于复位状态
25	WWDT_RESET	RW	0x0	WWDT 模块复位使能: 1: 正常工作 0: 模块处于复位状态
24:23	RSV	-	-	保留
22	DMA_RESET	RW	0x0	DMA 控制器模块复位使能: 1: 正常工作 0: 模块处于复位状态
21:20	RSV	-	-	保留
19	GPIOD_RESET	RW	0x0	GPIOD 模块复位使能: 1: 正常工作 0: 模块处于复位状态
18	GPIOC_RESET	RW	0x0	GPIOC 模块复位使能: 1: 正常工作 0: 模块处于复位状态
17	GPIOB_RESET	RW	0x0	GPIOB 模块复位使能: 1: 正常工作 0: 模块处于复位状态
16	GPIOA_RESET	RW	0x0	GPIOA 模块复位使能: 1: 正常工作 0: 模块处于复位状态
15	I2C_RESET	RW	0x0	I2C 模块复位使能: 1: 正常工作 0: 模块处于复位状态
14	ADC_RESET	RW	0x0	ADC 控制器模块复位使能: 1: 正常工作 0: 模块处于复位状态
13	RSV	-	-	保留
12	WDT_RESET	RW	0x0	WDT 模块复位使能: 1: 正常工作 0: 模块处于复位状态
11	CRC_RESET	RW	0x0	CRC16 模块复位使能: 1: 正常工作 0: 模块处于复位状态
10	RSV	-	-	保留

比特	名称	属性	复位值	描述
9	GTIMER0_RESET	RW	0x0	GTIMER0 模块复位使能: 1: 正常工作 0: 模块处于复位状态
8	LPTIMER_RESET	RW	0x0	LPTIMER 模块复位使能: 1: 正常工作 0: 模块处于复位状态
7:5	RSV	-	-	保留
4	SPI_RESET	RW	0x0	SPI 控制器模块复位使能: 1: 正常工作 0: 模块处于复位状态
3	GTIMER2_RESET	RW	0x0	GTIMER2 模块复位使能: 1: 正常工作 0: 模块处于复位状态
2	GTIMER1_RESET	RW	0x0	GTIMER1 模块复位使能: 1: 正常工作 0: 模块处于复位状态
1	RSV	-	-	保留
0	UART0_RESET	RW	0x0	UART0 模块复位使能: 1: 正常工作 0: 模块处于复位状态

### 5.5.7 外部复位滤波控制寄存器 SCU\_EXTRST\_CTRL (偏移: 028h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	EXT_FILTER_EN	RW	0x0	外部复位滤波使能位: 1: 外部复位滤波使能 0: 外部复位滤波禁止

### 5.5.8 端口 PA 功能配置寄存器 SCU\_PA\_SEL (偏移: 030h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:24	PA6_SEL	RW	0x0	端口 PA6 功能选择: 4'b0000: GPIO PA6 4'b0001: GTIM1_CH1 4'b0010: GTIM1_ETR 4'b0011: GTIM2_CH1 4'b0100: SPI_SCK 4'b0101: ATIM_CH1 4'b0110: ATIM_CH1N 4'b0111: ATIM_CH2

比特	名称	属性	复位值	描述
				4'b1000: ATIM_CH2N 4'b1001: ATIM_CH3 4'b1010: ATIM_CH3N 4'b1011: ATIM_CH4
23:20	PA5_SEL	RW	0x0	端口 PA5 功能选择: 4'b0000: GPIO PA5 4'b0001: GTIM1_CH2 4'b0010: GTIM2_CH2 4'b0011: SPI_CSN0 4'b0100: ATIM_CH1 4'b0101: ATIM_CH1N 4'b0110: ATIM_CH2 4'b0111: ATIM_CH2N 4'b1000: ATIM_CH3 4'b1001: ATIM_CH3N 4'b1010: ATIM_CH4 4'b1011: ATIM_BK0
19:16	PA4_SEL	RW	0x0	端口 PA4 功能选择: 4'b0000: GPIO PA4 4'b0001: UART1_RX 4'b0010: GTIM1_CH2 4'b0011: SPI_MISO 4'b0100: ATIM_CH1 4'b0101: ATIM_CH1N 4'b0110: ATIM_CH2 4'b0111: ATIM_CH2N 4'b1000: ATIM_CH3 4'b1001: ATIM_CH3N 4'b1010: ATIM_CH4 4'b1011: LPTIM_CAP2
15:12	PA3_SEL	RW	0x0	端口 PA3 功能选择: 4'b0000: GPIO PA3 4'b0001: UART1_TX 4'b0010: GTIM0_CH3 4'b0011: GTIM1_CH4 4'b0100: ATIM_CH1 4'b0101: ATIM_CH1N

比特	名称	属性	复位值	描述
				4'b0110: ATIM_CH2 4'b0111: ATIM_CH2N 4'b1000: ATIM_CH3 4'b1001: ATIM_CH3N 4'b1010: ATIM_CH4 4'b1011: LVD_OUT
11:8	PA2_SEL	RW	0x0	端口 PA2 功能选择: 4'b0000: GPIO PA2 4'b0001: NC 4'b0010: GTIM0_CH1 4'b0011: GTIM1_CH2 4'b0100: GTIM2_CH1 4'b0101: GTIM2_CH3 4'b0110: SPI_MOSI 4'b0111: ATIM_ETR 4'b1000: LPTIM_ETR 4'b1001: LPTIM_CAP1 4'b1010: I2C_SCL 4'b1011: I2C_SDA
7:4	PA1_SEL	RW	0x0	端口 PA1 功能选择: 4'b0000: GPIO PA1 4'b0001: UART0_RX 4'b0010: GTIM0_CH2 4'b0011: GTIM1_CH3 4'b0100: GTIM2_ETR 4'b0101: ATIM_CH1 4'b0110: ATIM_CH1N 4'b0111: ATIM_CH2 4'b1000: ATIM_CH2N 4'b1001: ATIM_CH3 4'b1010: ATIM_CH3N 4'b1011: ATIM_CH4
3:0	PA0_SEL	RW	0x0	端口 PA0 功能选择: 4'b0000: GPIO PA0 4'b0001: UART0_TX 4'b0010: GTIM0_CH1 4'b0011: GTIM1_CH2

比特	名称	属性	复位值	描述
				4'b0100: GTIM1_ETR 4'b0101: ATIM_CH1 4'b0110: ATIM_CH1N 4'b0111: ATIM_CH2 4'b1000: ATIM_CH2N 4'b1001: ATIM_CH3 4'b1010: ATIM_CH3N 4'b1011: ATIM_CH4

### 5.5.9 端口 PB 功能配置寄存器 SCU\_PB\_SEL (偏移: 034h)

比特	名称	属性	复位值	描述
31:28	PB7_SEL	RW	0x0	端口 PB7 功能选择: 4'b0000: GPIO PB7 4'b0001: UART1_RX 4'b0010: GTIM0_CH2 4'b0011: GTIM0_CH3 4'b0100: GTIM0_ETR 4'b0101: GTIM1_CH2 4'b0110: GTIM1_CH3 4'b0111: GTIM1_ETR 4'b1000: GTIM2_CH3 4'b1001: GTIM2_ETR 4'b1010: SPI_CSN0 4'b1011: SPI_MOSI 4'b1100: ATIM_ETR 4'b1101: ATIM_BK4 4'b1110: LPTIM_ETR 4'b1111: I2C_SCL
27:24	PB6_SEL	RW	0x0	端口 PB6 功能选择: 4'b0000: GPIO PB6 4'b0001: GTIM0_CH1 4'b0010: GTIM0_CH4 4'b0011: GTIM1_CH3 4'b0100: GTIM2_CH1 4'b0101: GTIM2_CH2 4'b0110: SPI_SCK

比特	名称	属性	复位值	描述
				4'b0111: ATIM_ETR 4'b1000: LPTIM_OUT1 4'b1001: LPTIM_IN 4'b1010: LPTIM_CAP2 4'b1011: COMP0_OUT
23:20	PB5_SEL	RW	0x0	端口 PB5 功能选择: 4'b0000: GPIO PB5 4'b0001: UART1_RX 4'b0010: GTIM0_CH2 4'b0011: GTIM1_CH1 4'b0100: GTIM1_CH4 4'b0101: GTIM2_CH2 4'b0110: GTIM2_CH3 4'b0111: SPI_CSN1 4'b1000: SPI_MISO 4'b1001: SPI_MOSI 4'b1010: CLK_OUT 4'b1011: LPTIM_OUT2 4'b1100: COMP1_OUT
19:16	PB4_SEL	RW	0x0	端口 PB4 功能选择: 4'b0000: GPIO PB4 4'b0001: UART1_TX 4'b0010: GTIM0_CH3 4'b0011: GTIM0_ETR 4'b0100: GTIM1_CH2 4'b0101: GTIM1_ETR 4'b0110: GTIM2_CH3 4'b0111: GTIM2_ETR 4'b1000: SPI_CSN0 4'b1001: SPI_MISO 4'b1010: SPI_MOSI 4'b1011: ATIM_CH4 4'b1100: I2C_SDA
15:12	PB3_SEL	RW	0x0	端口 PB3 功能选择: 4'b0000: GPIO PB3 4'b0001: UART0_TX 4'b0010: UART0_RX

比特	名称	属性	复位值	描述
				4'b0011: GTIM0_CH4 4'b0100: GTIM1_CH1 4'b0101: GTIM1_CH3 4'b0110: GTIM2_CH1 4'b0111: SPI_CSN1 4'b1000: SPI_SCK 4'b1001: SPI_MOSI 4'b1010: ATIM_CH1 4'b1011: ATIM_CH1N
11:8	PB2_SEL	RW	0x0	端口 PB2 功能选择: 4'b0000: GPIO PB2 4'b0001: GTIM0_CH1 4'b0010: GTIM0_CH3 4'b0011: GTIM0_ETR 4'b0100: GTIM2_CH1 4'b0101: GTIM2_CH4 4'b0110: SPI_MISO 4'b0111: SPI_MI1 4'b1000: ATIM_CH1 4'b1001: ATIM_CH1N 4'b1010: ATIM_CH2 4'b1011: ATIM_CH2N 4'b1100: ATIM_BK1 4'b1101: I2C_SCL
7:4	PB1_SEL	RW	0x0	端口 PB1 功能选择: 4'b0000: GPIO PB1 4'b0001: UART0_TX 4'b0010: GTIM0_CH2 4'b0011: GTIM0_ETR 4'b0100: GTIM1_CH3 4'b0101: GTIM2_CH2 4'b0110: SPI_CSN1 4'b0111: SPI_SCK 4'b1000: SPI_MOSI 4'b1001: ATIM_CH2N 4'b1010: ATIM_CH3 4'b1011: ATIM_CH3N

比特	名称	属性	复位值	描述
				4'b1100: CLK_OUT 4'b1101: I2C_SDA
3:0	PB0_SEL	RW	0x0	端口 PB0 功能选择: 4'b0000: GPIO PB0 4'b0001: UART0_RX 4'b0010: GTIM0_CH1 4'b0011: GTIM0_ETR 4'b0100: GTIM1_CH2 4'b0101: GTIM1_ETR 4'b0110: GTIM2_CH3 4'b0111: SPI_CSN0 4'b1000: SPI_MOSI 4'b1001: ATIM_CH1 4'b1010: ATIM_CH1N 4'b1011: ATIM_CH2 4'b1100: ATIM_CH2N 4'b1101: ATIM_CH3N 4'b1110: I2C_SCL

### 5.5.10 端口 PC 功能配置寄存器 SCU\_PC\_SEL (偏移: 038h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:24	PC6_SEL	RW	0x1	端口 PC6 功能选择: 4'b0000: GPIO PC6 4'b0001: SWCLK 4'b0010: UART0_TX 4'b0011: GTIM1_CH3 4'b0100: GTIM2_CH1 4'b0101: GTIM2_CH4 4'b0110: SPI_CSN0 4'b0111: SPI_MISO 4'b1000: SPI_MOSI 4'b1001: ATIM_ETR 4'b1010: I2C_SDA 4'b1011: COMP1_OUT
23:20	PC5_SEL	RW	0x1	端口 PC5 功能选择:

比特	名称	属性	复位值	描述
				4'b0000: GPIO PC5 4'b0001: SWIO 4'b0010: UART0_RX 4'b0011: UART1_TX 4'b0100: GTIM1_CH1 4'b0101: GTIM2_CH2 4'b0110: GTIM2_CH3 4'b0111: SPI_SCK 4'b1000: SPI_MI1 4'b1001: CLK_OUT 4'b1010: I2C_SCL 4'b1011: COMP0_OUT
19:16	PC4_SEL	RW	0x0	端口 PC4 功能选择: 4'b0000: GPIO PC4 4'b0001: UART0_RX 4'b0010: GTIM0_CH1 4'b0011: GTIM1_CH2 4'b0100: GTIM1_CH3 4'b0101: GTIM2_CH3 4'b0110: GTIM2_ETR 4'b0111: SPI_MOSI 4'b1000: ATIM_CH2N 4'b1001: ATIM_CH3 4'b1010: ATIM_ETR 4'b1011: ATIM_BK5 4'b1100: ADC_TRO 4'b1101: I2C_SDA 4'b1110: COMP1_OUT
15:12	PC3_SEL	RW	0x0	端口 PC3 功能选择: 4'b0000: GPIO PC3 4'b0001: UART0_TX 4'b0010: GTIM0_CH2 4'b0011: GTIM0_ETR 4'b0100: GTIM1_CH1 4'b0101: GTIM1_CH3 4'b0110: GTIM1_CH4 4'b0111: GTIM2_CH1

比特	名称	属性	复位值	描述
				4'b1000: GTIM2_CH3 4'b1001: ATIM_CH1N 4'b1010: ATIM_ETR 4'b1011: ADC_TRO 4'b1100: LPTIM_IN 4'b1101: LPTIM_CAP2 4'b1110: CAMPO_OUT
11:8	PC2_SEL	RW	0x0	端口 PC2 功能选择: 4'b0000: GPIO PC2 4'b0001: UART0_RX 4'b0010: GTIM0_CH1 4'b0011: GTIM0_CH3 4'b0100: GTIM0_ETR 4'b0101: GTIM1_CH2 4'b0110: GTIM1_CH3 4'b0111: GTIM1_ETR 4'b1000: GTIM2_CH4 4'b1001: ATIM_ETR 4'b1010: ATIM_BK6 4'b1011: ADC_TRO 4'b1100: COMP1_OUT
7:4	PC1_SEL	RW	0x0	端口 PC1 功能选择: 4'b0000: GPIO PC1 4'b0001: UART1_TX 4'b0010: GTIM0_CH1 4'b0011: GTIM0_CH4 4'b0100: GTIM1_CH4 4'b0101: GTIM2_CH2 4'b0110: GTIM2_ETR 4'b0111: SPI_SCK 4'b1000: LVD_OUT 4'b1001: LPTIM_OUT2 4'b1010: LPTIM_IN 4'b1011: LPTIM_CAP2
3:0	PC0_SEL	RW	0x0	端口 PC0 功能选择: 4'b0000: GPIO PC0 4'b0001: UART1_RX

比特	名称	属性	复位值	描述
				4'b0010: GTIM0_CH2 4'b0011: GTIM1_CH1 4'b0100: GTIM2_CH2 4'b0101: GTIM2_CH3 4'b0110: SPI_CSN1 4'b0111: SPI_MISO 4'b1000: SPI_MI1 4'b1001: LPTIM_OUT1 4'b1010: LPTIM_CAP1 4'b1011: I2C_SDA

### 5.5.11 端口 PD 功能配置寄存器 SCU\_PD\_SEL (偏移: 03Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	PD3_SEL	RW	0x0	端口 PD3 功能选择 4'b0000: GPIO PD3 4'b0001: UART1_RX 4'b0010: GTIM0_CH2 4'b0011: GTIM2_CH1 4'b0100: SPI_MOSI 4'b0101: ATIM_CH1 4'b0110; ATIM_CH1N 4'b0111: ATIM_CH2 4'b1000: ATIM_CH2N 4'b1001: ATIM_CH3 4'b1010: ATIM_CH3N 4'b1011: ATIM_CH4 4'b1100: ATIM_ETR 4'b1101: ATIM_BK2 4'b1110: ADC_TRO 4'b1111: I2C_SDA
11:8	PD2_SEL	RW	0x0	端口 PD2 功能选择 4'b0000: GPIO PD2 4'b0001: UART1_TX 4'b0010: GTIM0_CH3 4'b0011: GTIM0_ETR

比特	名称	属性	复位值	描述
				4'b0100: GTIM2_CH1 4'b0101: GTIM2_CH2 4'b0110: GTIM2_CH3 4'b0111: SPI_SCK 4'b1000: SPI_MOSI 4'b1001: ATIM_CH1 4'b1010: ATIM_CH1N 4'b1011: ATIM_CH2 4'b1100: ATIM_CH2N 4'b1101: ATIM_CH3 4'b1110: ATIM_CH3N 4'b1111: I2C_SCL
7:4	PD1_SEL	RW	0x0	端口 PD1 功能选择 4'b0000: GPIO PD1 4'b0001: UART0_TX 4'b0010: UART1_RX 4'b0011: GTIM0_CH1 4'b0100: GTIM0_ETR 4'b0101: GTIM1_ETR 4'b0110: GTIM2_CH2 4'b0111: SPI_MISO 4'b1000: SPI_MI1 4'b1001: ATIM_CH3 4'b1010: ATIM_CH3N 4'b1011: ATIM_ETR 4'b1100: ADC_TRO 4'b1101: I2C_SDA 4'b1110: COMP0_OUT
3:0	PD0_SEL	RW	0x0	端口 PD0 功能选择 4'b0000: GPIO PD0 4'b0001: UART0_TX 4'b0010: GTIM0_CH2 4'b0011: GTIM0_CH3 4'b0100: GTIM0_ETR 4'b0101: GTIM1_CH1 4'b0110: SPI_CSN0 4'b0111: SPI_CSN1

比特	名称	属性	复位值	描述
				4'b1000: SPI_SCK 4'b1001: ATIM_CH2N 4'b1010: ATIM_CH3 4'b1011: ATIM_ETR 4'b1100: ATIM_BK3 4'b1101: ADC_TRO 4'b1110: I2C_SCL 4'b1111: COMP1_OUT

### 5.5.12 端口数模配置寄存器 SCU\_PAD\_ADS (偏移: 054h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_ADS	RW	0x0	端口 PD3 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
26	PD2_ADS	RW	0x0	端口 PD2 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
25	PD1_ADS	RW	0x0	端口 PD1 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
24	PD0_ADS	RW	0x0	端口 PD0 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
23:21	RSV	-	-	保留
20	PC4_ADS	RW	0x0	端口 PC4 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
19	PC3_ADS	RW	0x0	端口 PC3 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
18	PC2_ADS	RW	0x0	端口 PC2 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
17	PC1_ADS	RW	0x0	端口 PC1 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
16	PC0_ADS	RW	0x0	端口 PC0 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
15	PB7_ADS	RW	0x0	端口 PB7 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
14	PB6_ADS	RW	0x0	端口 PB6 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口

比特	名称	属性	复位值	描述
13	PB5_ADS	RW	0x0	端口 PB5 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
12	PB4_ADS	RW	0x0	端口 PB4 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
11	PB3_ADS	RW	0x0	端口 PB3 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
10	PB2_ADS	RW	0x0	端口 PB2 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
9	PB1_ADS	RW	0x0	端口 PB1 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
8	PB0_ADS	RW	0x0	端口 PB0 数模配置寄存器: 0: 配置为数字接口 1: 配置为模拟接口
7:0	RSV	-	-	保留

### 5.5.13 端口驱动能力配置寄存器 SCU\_PAD\_DR (偏移: 060h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_DR	RW	0x0	端口 PD3 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
26	PD2_DR	RW	0x0	端口 PD2 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
25	PD1_DR	RW	0x0	端口 PD1 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
24	PD0_DR	RW	0x0	端口 PD0 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
23	RSV	-	-	保留
22	PC6_DR	RW	0x0	端口 PC6 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
21	PC5_DR	RW	0x0	端口 PC5 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
20	PC4_DR	RW	0x0	端口 PC4 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
19	PC3_DR	RW	0x0	端口 PC3 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力

比特	名称	属性	复位值	描述
18	PC2_DR	RW	0x0	端口 PC2 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
17	PC1_DR	RW	0x0	端口 PC1 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
16	PC0_DR	RW	0x0	端口 PC0 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
15	PB7_DR	RW	0x0	端口 PB7 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
14	PB6_DR	RW	0x0	端口 PB6 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
13	PB5_DR	RW	0x0	端口 PB5 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
12	PB4_DR	RW	0x0	端口 PB4 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
11	PB3_DR	RW	0x0	端口 PB3 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
10	PB2_DR	RW	0x0	端口 PB2 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
9	PB1_DR	RW	0x0	端口 PB1 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
8	PB0_DR	RW	0x0	端口 PB0 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
7	RSV	-	-	保留
6	PA6_DR	RW	0x0	端口 PA6 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
5	PA5_DR	RW	0x0	端口 PA5 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
4	PA4_DR	RW	0x0	端口 PA4 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
3	PA3_DR	RW	0x0	端口 PA3 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
2	PA2_DR	RW	0x0	端口 PA2 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
1	PA1_DR	RW	0x0	端口 PA1 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力
0	PA0_DR	RW	0x0	端口 PA0 驱动能力配置寄存器: 0: 高驱动能力 1: 低驱动能力

## 5.5.14 端口上拉配置寄存器 SCU\_PAD\_PU (偏移: 06Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_PU	RW	0x0	端口 PD3 上拉配置寄存器: 0: 禁止 1: 使能
26	PD2_PU	RW	0x0	端口 PD2 上拉配置寄存器: 0: 禁止 1: 使能
25	PD1_PU	RW	0x0	端口 PD1 上拉配置寄存器: 0: 禁止 1: 使能
24	PD0_PU	RW	0x0	端口 PD0 上拉配置寄存器: 0: 禁止 1: 使能
23	RSV	-	-	保留
22	PC6_PU	RW	0x0	端口 PC6 上拉配置寄存器: 0: 禁止 1: 使能
21	PC5_PU	RW	0x1	端口 PC5 上拉配置寄存器: 0: 禁止 1: 使能
20	PC4_PU	RW	0x0	端口 PC4 上拉配置寄存器: 0: 禁止 1: 使能
19	PC3_PU	RW	0x0	端口 PC3 上拉配置寄存器: 0: 禁止 1: 使能
18	PC2_PU	RW	0x0	端口 PC2 上拉配置寄存器: 0: 禁止 1: 使能
17	PC1_PU	RW	0x0	端口 PC1 上拉配置寄存器: 0: 禁止 1: 使能
16	PC0_PU	RW	0x0	端口 PC0 上拉配置寄存器: 0: 禁止 1: 使能
15	PB7_PU	RW	0x0	端口 PB7 上拉配置寄存器: 0: 禁止 1: 使能
14	PB6_PU	RW	0x0	端口 PB6 上拉配置寄存器: 0: 禁止 1: 使能
13	PB5_PU	RW	0x0	端口 PB5 上拉配置寄存器: 0: 禁止 1: 使能
12	PB4_PU	RW	0x0	端口 PB4 上拉配置寄存器: 0: 禁止 1: 使能
11	PB3_PU	RW	0x0	端口 PB3 上拉配置寄存器: 0: 禁止 1: 使能

比特	名称	属性	复位值	描述
10	PB2_PU	RW	0x0	端口 PB2 上拉配置寄存器: 0: 禁止 1: 使能
9	PB1_PU	RW	0x0	端口 PB1 上拉配置寄存器: 0: 禁止 1: 使能
8	PB0_PU	RW	0x0	端口 PB0 上拉配置寄存器: 0: 禁止 1: 使能
7	RSV	-	-	保留
6	PA6_PU	RW	0x0	端口 PA6 上拉配置寄存器: 0: 禁止 1: 使能
5	PA5_PU	RW	0x0	端口 PA5 上拉配置寄存器: 0: 禁止 1: 使能
4	PA4_PU	RW	0x0	端口 PA4 上拉配置寄存器: 0: 禁止 1: 使能
3	PA3_PU	RW	0x0	端口 PA3 上拉配置寄存器: 0: 禁止 1: 使能
2	PA2_PU	RW	0x1	端口 PA2 上拉配置寄存器: 0: 禁止 1: 使能
1	PA1_PU	RW	0x0	端口 PA1 上拉配置寄存器: 0: 禁止 1: 使能
0	PA0_PU	RW	0x0	端口 PA0 上拉配置寄存器: 0: 禁止 1: 使能

### 5.5.15 端口下拉配置寄存器 SCU\_PAD\_PD (偏移: 078h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_PD	RW	0x0	端口 PD3 下拉配置寄存器: 0: 禁止 1: 使能
26	PD2_PD	RW	0x0	端口 PD2 下拉配置寄存器: 0: 禁止 1: 使能
25	PD1_PD	RW	0x0	端口 PD1 下拉配置寄存器: 0: 禁止 1: 使能
24	PD0_PD	RW	0x0	端口 PD0 下拉配置寄存器: 0: 禁止 1: 使能
23	RSV	-	-	保留

比特	名称	属性	复位值	描述
22	PC6_PD	RW	0x1	端口 PC6 下拉配置寄存器: 0: 禁止 1: 使能
21	PC5_PD	RW	0x0	端口 PC5 下拉配置寄存器: 0: 禁止 1: 使能
20	PC4_PD	RW	0x0	端口 PC4 下拉配置寄存器: 0: 禁止 1: 使能
19	PC3_PD	RW	0x0	端口 PC3 下拉配置寄存器: 0: 禁止 1: 使能
18	PC2_PD	RW	0x0	端口 PC2 下拉配置寄存器: 0: 禁止 1: 使能
17	PC1_PD	RW	0x0	端口 PC1 下拉配置寄存器: 0: 禁止 1: 使能
16	PC0_PD	RW	0x0	端口 PC0 下拉配置寄存器: 0: 禁止 1: 使能
15	PB7_PD	RW	0x0	端口 PB7 下拉配置寄存器: 0: 禁止 1: 使能
14	PB6_PD	RW	0x0	端口 PB6 下拉配置寄存器: 0: 禁止 1: 使能
13	PB5_PD	RW	0x0	端口 PB5 下拉配置寄存器: 0: 禁止 1: 使能
12	PB4_PD	RW	0x0	端口 PB4 下拉配置寄存器: 0: 禁止 1: 使能
11	PB3_PD	RW	0x0	端口 PB3 下拉配置寄存器: 0: 禁止 1: 使能
10	PB2_PD	RW	0x0	端口 PB2 下拉配置寄存器: 0: 禁止 1: 使能
9	PB1_PD	RW	0x0	端口 PB1 下拉配置寄存器: 0: 禁止 1: 使能
8	PB0_PD	RW	0x0	端口 PB0 下拉配置寄存器: 0: 禁止 1: 使能
7	RSV	-	-	保留
6	PA6_PD	RW	0x0	端口 PA6 下拉配置寄存器: 0: 禁止 1: 使能
5	PA5_PD	RW	0x0	端口 PA5 下拉配置寄存器: 0: 禁止 1: 使能
4	PA4_PD	RW	0x0	端口 PA4 下拉配置寄存器: 0: 禁止 1: 使能

比特	名称	属性	复位值	描述
3	PA3_PD	RW	0x0	端口 PA3 下拉配置寄存器: 0: 禁止 1: 使能
2	PA2_PD	RW	0x0	端口 PA2 下拉配置寄存器: 0: 禁止 1: 使能
1	PA1_PD	RW	0x0	端口 PA1 下拉配置寄存器: 0: 禁止 1: 使能
0	PA0_PD	RW	0x0	端口 PA0 下拉配置寄存器: 0: 禁止 1: 使能

### 5.5.16 端口开漏输出配置寄存器 SCU\_PAD\_OD (偏移: 084h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_OD	RW	0x0	端口 PD3 开漏输出配置寄存器: 0: 禁止 1: 使能
26	PD2_OD	RW	0x0	端口 PD2 开漏输出配置寄存器: 0: 禁止 1: 使能
25	PD1_OD	RW	0x0	端口 PD1 开漏输出配置寄存器: 0: 禁止 1: 使能
24	PD0_OD	RW	0x0	端口 PD0 开漏输出配置寄存器: 0: 禁止 1: 使能
23	RSV	-	-	保留
22	PC6_OD	RW	0x0	端口 PC6 开漏输出配置寄存器: 0: 禁止 1: 使能
21	PC5_OD	RW	0x0	端口 PC5 开漏输出配置寄存器: 0: 禁止 1: 使能
20	PC4_OD	RW	0x0	端口 PC4 开漏输出配置寄存器: 0: 禁止 1: 使能
19	PC3_OD	RW	0x0	端口 PC3 开漏输出配置寄存器: 0: 禁止 1: 使能
18	PC2_OD	RW	0x0	端口 PC2 开漏输出配置寄存器: 0: 禁止 1: 使能
17	PC1_OD	RW	0x0	端口 PC1 开漏输出配置寄存器: 0: 禁止 1: 使能
16	PC0_OD	RW	0x0	端口 PC0 开漏输出配置寄存器: 0: 禁止 1: 使能

比特	名称	属性	复位值	描述
15	PB7_OD	RW	0x0	端口 PB7 开漏输出配置寄存器: 0: 禁止 1: 使能
14	PB6_OD	RW	0x0	端口 PB6 开漏输出配置寄存器: 0: 禁止 1: 使能
13	PB5_OD	RW	0x0	端口 PB5 开漏输出配置寄存器: 0: 禁止 1: 使能
12	PB4_OD	RW	0x0	端口 PB4 开漏输出配置寄存器: 0: 禁止 1: 使能
11	PB3_OD	RW	0x0	端口 PB3 开漏输出配置寄存器: 0: 禁止 1: 使能
10	PB2_OD	RW	0x0	端口 PB2 开漏输出配置寄存器: 0: 禁止 1: 使能
9	PB1_OD	RW	0x0	端口 PB1 开漏输出配置寄存器: 0: 禁止 1: 使能
8	PB0_OD	RW	0x0	端口 PB0 开漏输出配置寄存器: 0: 禁止 1: 使能
7	RSV	-	-	保留
6	PA6_OD	RW	0x0	端口 PA6 开漏输出配置寄存器: 0: 禁止 1: 使能
5	PA5_OD	RW	0x0	端口 PA5 开漏输出配置寄存器: 0: 禁止 1: 使能
4	PA4_OD	RW	0x0	端口 PA4 开漏输出配置寄存器: 0: 禁止 1: 使能
3	PA3_OD	RW	0x0	端口 PA3 开漏输出配置寄存器: 0: 禁止 1: 使能
2	PA2_OD	RW	0x0	端口 PA2 开漏输出配置寄存器: 0: 禁止 1: 使能
1	PA1_OD	RW	0x0	端口 PA1 开漏输出配置寄存器: 0: 禁止 1: 使能
0	PA0_OD	RW	0x0	端口 PA0 开漏输出配置寄存器: 0: 禁止 1: 使能

### 5.5.17 端口输入类型配置寄存器 SCU\_PAD\_CS (偏移: 090h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_CS	RW	0x0	端口 PD3 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
26	PD2_CS	RW	0x0	端口 PD2 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
25	PD1_CS	RW	0x0	端口 PD1 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
24	PD0_CS	RW	0x0	端口 PD0 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
23	RSV	-	-	保留
22	PC6_CS	RW	0x0	端口 PC6 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
21	PC5_CS	RW	0x0	端口 PC5 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
20	PC4_CS	RW	0x0	端口 PC4 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
19	PC3_CS	RW	0x0	端口 PC3 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
18	PC2_CS	RW	0x0	端口 PC2 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
17	PC1_CS	RW	0x0	端口 PC1 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
16	PC0_CS	RW	0x0	端口 PC0 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
15	PB7_CS	RW	0x0	端口 PB7 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
14	PB6_CS	RW	0x0	端口 PB6 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
13	PB5_CS	RW	0x0	端口 PB5 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
12	PB4_CS	RW	0x0	端口 PB4 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
11	PB3_CS	RW	0x0	端口 PB3 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer

比特	名称	属性	复位值	描述
10	PB2_CS	RW	0x0	端口 PB2 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
9	PB1_CS	RW	0x0	端口 PB1 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
8	PB0_CS	RW	0x0	端口 PB0 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
7	RSV	-	-	保留
6	PA6_CS	RW	0x0	端口 PA6 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
5	PA5_CS	RW	0x0	端口 PA5 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
4	PA4_CS	RW	0x0	端口 PA4 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
3	PA3_CS	RW	0x0	端口 PA3 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
2	PA2_CS	RW	0x0	端口 PA2 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
1	PA1_CS	RW	0x0	端口 PA1 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer
0	PA0_CS	RW	0x0	端口 PA0 输入类型配置寄存器: 0: Schmitt input buffer 1: CMOS input buffer

### 5.5.18 端口输入配置寄存器 SCU\_PAD\_IE (偏移: 09Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_IE	RW	0x0	端口 PD3 输入配置寄存器: 0: 输入禁止 1: 输入使能
26	PD2_IE	RW	0x0	端口 PD2 输入配置寄存器: 0: 输入禁止 1: 输入使能
25	PD1_IE	RW	0x0	端口 PD1 输入配置寄存器: 0: 输入禁止 1: 输入使能
24	PD0_IE	RW	0x0	端口 PD0 输入配置寄存器: 0: 输入禁止 1: 输入使能
23	RSV	-	-	保留

比特	名称	属性	复位值	描述
22	PC6_IE	RW	0x1	端口 PC6 输入配置寄存器: 0: 输入禁止 1: 输入使能
21	PC5_IE	RW	0x1	端口 PC5 输入配置寄存器: 0: 输入禁止 1: 输入使能
20	PC4_IE	RW	0x0	端口 PC4 输入配置寄存器: 0: 输入禁止 1: 输入使能
19	PC3_IE	RW	0x0	端口 PC3 输入配置寄存器: 0: 输入禁止 1: 输入使能
18	PC2_IE	RW	0x0	端口 PC2 输入配置寄存器: 0: 输入禁止 1: 输入使能
17	PC1_IE	RW	0x0	端口 PC1 输入配置寄存器: 0: 输入禁止 1: 输入使能
16	PC0_IE	RW	0x0	端口 PC0 输入配置寄存器: 0: 输入禁止 1: 输入使能
15	PB7_IE	RW	0x0	端口 PB7 输入配置寄存器: 0: 输入禁止 1: 输入使能
14	PB6_IE	RW	0x0	端口 PB6 输入配置寄存器: 0: 输入禁止 1: 输入使能
13	PB5_IE	RW	0x0	端口 PB5 输入配置寄存器: 0: 输入禁止 1: 输入使能
12	PB4_IE	RW	0x0	端口 PB4 输入配置寄存器: 0: 输入禁止 1: 输入使能
11	PB3_IE	RW	0x0	端口 PB3 输入配置寄存器: 0: 输入禁止 1: 输入使能
10	PB2_IE	RW	0x0	端口 PB2 输入配置寄存器: 0: 输入禁止 1: 输入使能
9	PB1_IE	RW	0x0	端口 PB1 输入配置寄存器: 0: 输入禁止 1: 输入使能
8	PB0_IE	RW	0x0	端口 PB0 输入配置寄存器: 0: 输入禁止 1: 输入使能
7	RSV	-	-	保留
6	PA6_IE	RW	0x0	端口 PA6 输入配置寄存器: 0: 输入禁止 1: 输入使能
5	PA5_IE	RW	0x0	端口 PA5 输入配置寄存器: 0: 输入禁止 1: 输入使能
4	PA4_IE	RW	0x0	端口 PA4 输入配置寄存器: 0: 输入禁止 1: 输入使能

比特	名称	属性	复位值	描述
3	PA3_IE	RW	0x0	端口 PA3 输入配置寄存器: 0: 输入禁止 1: 输入使能
2	PA2_IE	RW	0x1	端口 PA2 输入配置寄存器: 0: 输入禁止 1: 输入使能
1	PA1_IE	RW	0x0	端口 PA1 输入配置寄存器: 0: 输入禁止 1: 输入使能
0	PA0_IE	RW	0x0	端口 PA0 输入配置寄存器: 0: 输入禁止 1: 输入使能

### 5.5.19 端口输入电平寄存器 SCU\_PAD\_STATUS (偏移: 0A4h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_I	R	0x0	端口 PD3 输入电平寄存器: 0: 输入低电平 1: 输入高电平
26	PD2_I	R	0x0	端口 PD2 输入电平寄存器: 0: 输入低电平 1: 输入高电平
25	PD1_I	R	0x0	端口 PD1 输入电平寄存器: 0: 输入低电平 1: 输入高电平
24	PD0_I	R	0x0	端口 PD0 输入电平寄存器: 0: 输入低电平 1: 输入高电平
23	RSV	-	-	保留
22	PC6_I	R	0x1	端口 PC6 输入电平寄存器: 0: 输入低电平 1: 输入高电平
21	PC5_I	R	0x1	端口 PC5 输入电平寄存器: 0: 输入低电平 1: 输入高电平
20	PC4_I	R	0x0	端口 PC4 输入电平寄存器: 0: 输入低电平 1: 输入高电平
19	PC3_I	R	0x0	端口 PC3 输入电平寄存器: 0: 输入低电平 1: 输入高电平
18	PC2_I	R	0x0	端口 PC2 输入电平寄存器: 0: 输入低电平 1: 输入高电平
17	PC1_I	R	0x0	端口 PC1 输入电平寄存器: 0: 输入低电平 1: 输入高电平
16	PC0_I	R	0x0	端口 PC0 输入电平寄存器: 0: 输入低电平 1: 输入高电平

比特	名称	属性	复位值	描述
15	PB7_I	R	0x0	端口 PB7 输入电平寄存器: 0: 输入低电平 1: 输入高电平
14	PB6_I	R	0x0	端口 PB6 输入电平寄存器: 0: 输入低电平 1: 输入高电平
13	PB5_I	R	0x0	端口 PB5 输入电平寄存器: 0: 输入低电平 1: 输入高电平
12	PB4_I	R	0x0	端口 PB4 输入电平寄存器: 0: 输入低电平 1: 输入高电平
11	PB3_I	R	0x0	端口 PB3 输入电平寄存器: 0: 输入低电平 1: 输入高电平
10	PB2_I	R	0x0	端口 PB2 输入电平寄存器: 0: 输入低电平 1: 输入高电平
9	PB1_I	R	0x0	端口 PB1 输入电平寄存器: 0: 输入低电平 1: 输入高电平
8	PB0_I	R	0x0	端口 PB0 输入电平寄存器: 0: 输入低电平 1: 输入高电平
7	RSV	-	-	保留
6	PA6_I	R	0x0	端口 PA6 输入电平寄存器: 0: 输入低电平 1: 输入高电平
5	PA5_I	R	0x0	端口 PA5 输入电平寄存器: 0: 输入低电平 1: 输入高电平
4	PA4_I	R	0x0	端口 PA4 输入电平寄存器: 0: 输入低电平 1: 输入高电平
3	PA3_I	R	0x0	端口 PA3 输入电平寄存器: 0: 输入低电平 1: 输入高电平
2	PA2_I	R	0x1	端口 PA2 输入电平寄存器: 0: 输入低电平 1: 输入高电平
1	PA1_I	R	0x0	端口 PA1 输入电平寄存器: 0: 输入低电平 1: 输入高电平
0	PA0_I	R	0x0	端口 PA0 输入电平寄存器: 0: 输入低电平 1: 输入高电平

### 5.5.20 端口速度配置寄存器 SCU\_PAD\_SR (偏移: 0A8h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	PD3_SR	RW	0x1	端口 PD3 速度配置寄存器: 0: 高速 1: 低速
26	PD2_SR	RW	0x1	端口 PD2 速度配置寄存器: 0: 高速 1: 低速
25	PD1_SR	RW	0x1	端口 PD1 速度配置寄存器: 0: 高速 1: 低速
24	PD0_SR	RW	0x1	端口 PD0 速度配置寄存器: 0: 高速 1: 低速
23	RSV	-	-	保留
22	PC6_SR	RW	0x1	端口 PC6 速度配置寄存器: 0: 高速 1: 低速
21	PC5_SR	RW	0x1	端口 PC5 速度配置寄存器: 0: 高速 1: 低速
20	PC4_SR	RW	0x1	端口 PC4 速度配置寄存器: 0: 高速 1: 低速
19	PC3_SR	RW	0x1	端口 PC3 速度配置寄存器: 0: 高速 1: 低速
18	PC2_SR	RW	0x1	端口 PC2 速度配置寄存器: 0: 高速 1: 低速
17	PC1_SR	RW	0x1	端口 PC1 速度配置寄存器: 0: 高速 1: 低速
16	PC0_SR	RW	0x1	端口 PC0 速度配置寄存器: 0: 高速 1: 低速
15	PB7_SR	RW	0x1	端口 PB7 速度配置寄存器: 0: 高速 1: 低速
14	PB6_SR	RW	0x1	端口 PB6 速度配置寄存器: 0: 高速 1: 低速
13	PB5_SR	RW	0x1	端口 PB5 速度配置寄存器: 0: 高速 1: 低速
12	PB4_SR	RW	0x1	端口 PB4 速度配置寄存器: 0: 高速 1: 低速
11	PB3_SR	RW	0x1	端口 PB3 速度配置寄存器: 0: 高速 1: 低速

比特	名称	属性	复位值	描述
10	PB2_SR	RW	0x1	端口 PB2 速度配置寄存器: 0: 高速 1: 低速
9	PB1_SR	RW	0x1	端口 PB1 速度配置寄存器: 0: 高速 1: 低速
8	PB0_SR	RW	0x1	端口 PB0 速度配置寄存器: 0: 高速 1: 低速
7	RSV	-	-	保留
6	PA6_SR	RW	0x1	端口 PA6 速度配置寄存器: 0: 高速 1: 低速
5	PA5_SR	RW	0x1	端口 PA5 速度配置寄存器: 0: 高速 1: 低速
4	PA4_SR	RW	0x1	端口 PA4 速度配置寄存器: 0: 高速 1: 低速
3	PA3_SR	RW	0x1	端口 PA3 速度配置寄存器: 0: 高速 1: 低速
2	PA2_SR	RW	0x1	端口 PA2 速度配置寄存器: 0: 高速 1: 低速
1	PA1_SR	RW	0x1	端口 PA1 速度配置寄存器: 0: 高速 1: 低速
0	PA0_SR	RW	0x1	端口 PA0 速度配置寄存器: 0: 高速 1: 低速

### 5.5.21 IO 控制保护寄存器 SCU\_IOCTLL\_PROT (偏移: 0B4h)

比特	名称	属性	复位值	描述
31:0	IOCTRL_PROTECT	RW	0x0	IO 寄存器 PA_SEL/PB_SEL/PC_SEL/PD_SEL/ PAD_ADS /PAD_DR /PAD_PU /PAD_PD /PAD_OD /PAD_CS /PAD_IE /PAD_SR 保护的寄存器。 给此寄存器写 0xA5A5_5A5A, 启动这些 IO 寄存 器的写使能。配置完 IO 寄存器后, 它们的写使能 不会自动关闭。可以给此寄存器写其它值, 来关 闭 IO 寄存器的写使能。 读取此寄存器返回 IO 寄存器的写使能状态: 0: 写未使能 1: 写已经使能

## 5.5.22 LVD 配置寄存器 SCU\_LVD\_CFG (偏移: 0B8h)

比特	名称	属性	复位值	描述																																				
31	BUF_VADJ	RW	0x0	BUF 输出电压设置: 1: BUF 输出 1.5V 0: BUF 输出 0.8V																																				
30	BUF_O2P	RW	0x0	BUF 输出到 PD2 输出使能: 1: BUF 输出到 PD2 0: BUF 不输出到 PD2																																				
29	BUF_EN	RW	0x0	BUF 使能: 1: BUF 打开 0: BUF 关闭																																				
28:25	RSV	-	-	保留																																				
24	LVD_LVEN	RW	0x1	LVD 滤波使能位																																				
23:16	LVD_FILTER	RW	0x20	LVD 滤波配置位: 0: 对 LVD 滤除 1 个 32K 系统低速时钟的毛刺 1: 对 LVD 滤除 2 个 32K 系统低速时钟的毛刺 ..... 255: 对 LVD 滤除 256 个 32K 系统低速时钟毛刺																																				
15:14	RSV	-	-	保留																																				
13:10	LVRS	RW	0x0	LVR 检测点电压设置: <table border="1"> <thead> <tr> <th>LVRS</th> <th>LVR POINT</th> <th>LVRS</th> <th>LVR POINT</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>1.89</td> <td>1000</td> <td>4.2</td> </tr> <tr> <td>0001</td> <td>2.1</td> <td>1001</td> <td>4.48</td> </tr> <tr> <td>0010</td> <td>2.39</td> <td>1010</td> <td>4.81</td> </tr> <tr> <td>0011</td> <td>2.7</td> <td>1011</td> <td>保留</td> </tr> <tr> <td>0100</td> <td>3</td> <td>1100</td> <td>保留</td> </tr> <tr> <td>0101</td> <td>3.3</td> <td>1101</td> <td>保留</td> </tr> <tr> <td>0110</td> <td>3.6</td> <td>1110</td> <td>保留</td> </tr> <tr> <td>0111</td> <td>3.87</td> <td>1111</td> <td>保留</td> </tr> </tbody> </table>	LVRS	LVR POINT	LVRS	LVR POINT	0000	1.89	1000	4.2	0001	2.1	1001	4.48	0010	2.39	1010	4.81	0011	2.7	1011	保留	0100	3	1100	保留	0101	3.3	1101	保留	0110	3.6	1110	保留	0111	3.87	1111	保留
LVRS	LVR POINT	LVRS	LVR POINT																																					
0000	1.89	1000	4.2																																					
0001	2.1	1001	4.48																																					
0010	2.39	1010	4.81																																					
0011	2.7	1011	保留																																					
0100	3	1100	保留																																					
0101	3.3	1101	保留																																					
0110	3.6	1110	保留																																					
0111	3.87	1111	保留																																					
9	LVD_INTR_EN	RW	0x0	LVD 中断使能控制位: 0: 不使能 LVD 中断 1: 使能 LVD 中断																																				
8	LVD_RESET_EN	RW	0x0	LVD 复位使能控制位: 0: 不使能 LVD 复位 1: 使能 LVD 复位																																				

比特	名称	属性	复位值	描述																																				
7:4	LVDS	RW	0x0	LVD 检测点电压设置: <table border="1"> <thead> <tr> <th>LVDS</th> <th>LVD POINT</th> <th>LVDS</th> <th>LVD POINT</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>1.89</td> <td>1000</td> <td>4.2</td> </tr> <tr> <td>0001</td> <td>2.1</td> <td>1001</td> <td>4.48</td> </tr> <tr> <td>0010</td> <td>2.39</td> <td>1010</td> <td>4.81</td> </tr> <tr> <td>0011</td> <td>2.7</td> <td>1011</td> <td>保留</td> </tr> <tr> <td>0100</td> <td>3</td> <td>1100</td> <td>保留</td> </tr> <tr> <td>0101</td> <td>3.3</td> <td>1101</td> <td>保留</td> </tr> <tr> <td>0110</td> <td>3.6</td> <td>1110</td> <td>保留</td> </tr> <tr> <td>0111</td> <td>3.87</td> <td>1111</td> <td>保留</td> </tr> </tbody> </table>	LVDS	LVD POINT	LVDS	LVD POINT	0000	1.89	1000	4.2	0001	2.1	1001	4.48	0010	2.39	1010	4.81	0011	2.7	1011	保留	0100	3	1100	保留	0101	3.3	1101	保留	0110	3.6	1110	保留	0111	3.87	1111	保留
LVDS	LVD POINT	LVDS	LVD POINT																																					
0000	1.89	1000	4.2																																					
0001	2.1	1001	4.48																																					
0010	2.39	1010	4.81																																					
0011	2.7	1011	保留																																					
0100	3	1100	保留																																					
0101	3.3	1101	保留																																					
0110	3.6	1110	保留																																					
0111	3.87	1111	保留																																					
3:1	RSV	-	-	保留																																				
0	LVD_EN	RW	0x0	LVD 模块使能寄存器: 0: 禁止 1: 使能																																				

### 5.5.23 PGA 控制寄存器 SCU\_PGA\_CFG (偏移: 0BCh)

比特	名称	属性	复位值	描述
31:26	RSV	-	-	保留
25	PGA_VREFSEL	RW	0x0	VREF 电压的选择信号: 0: VREFLDO 电压, 1.2V/3.6/4.2V/AVDD 1: PD3 输入电压
24	PGA1_VSEL	RW	0x0	PGA1 和 PGA2 的输出共模电压选择信号: 0: VREF (经 PGA_VREFSEL 选择输出的电压) 1: VBG (BUF LDO)
23	PGA0_VSEL	RW	0x0	PGA0 的输出共模电压选择信号: 0: VREF (经 PGA_VREFSEL 选择输出的电压) 1: VBG (BUF LDO)
22:21	PGA0_RSEL	RW	0x0	VOUT0 输出开关导通阻值选择信号: 00: 200ohm 01: 1kohm 10: 10kohm 11: HiZ
20	PGA2_POSEL	RW	0x0	PGA2 的正输入端电压选择信号: 0: 选择外部输入, PC4 输入

比特	名称	属性	复位值	描述
				1: 内部接到 AVSS
19	PGA1_POSSEL	RW	0x0	PGA1 的正输入端电压选择信号: 0: 选择外部输入, PC1 输入 1: 内部接到 AVSS
18	PGA0_POSSEL	RW	0x0	PGA0 的正输入端电压选择信号: 0: 选择外部输入, PB6 输入 1: 内部接到 AVSS
17	PGA2_NEGSEL	RW	0x0	PGA2 的负输入端电压选择信号: 0: 选择外部输入, PC3 输入 1: 内部接到 AVSS
16	PGA1_NEGSEL	RW	0x0	PGA1 的负输入端电压选择信号: 0: 选择外部输入, PC0 输入 1: 内部接到 AVSS
15	PGA0_NEGSEL	RW	0x0	PGA0 的负输入端电压选择信号: 0: 选择外部输入, PB5 输入 1: 内部接到 AVSS
14:12	PGA2_SELGAIN	RW	0x0	PGA2 增益控制信号: PGA2_SELGAIN =000, 1 倍放大 PGA2_SELGAIN =001, 2 倍放大 PGA2_SELGAIN =010, 4 倍放大 PGA2_SELGAIN =011, 8 倍放大 PGA2_SELGAIN =100, 16 倍放大 PGA2_SELGAIN =101, 32 倍放大 PGA2_SELGAIN =110, 64 倍放大
11:9	PGA1_SELGAIN	RW	0x0	PGA1 增益控制信号: PGA1_SELGAIN =000, 1 倍放大 PGA1_SELGAIN =001, 2 倍放大 PGA1_SELGAIN =010, 4 倍放大 PGA1_SELGAIN =011, 8 倍放大 PGA1_SELGAIN =100, 16 倍放大 PGA1_SELGAIN =101, 32 倍放大 PGA1_SELGAIN =110, 64 倍放大
8:6	PGA0_SELGAIN	RW	0x0	PGA0 增益控制信号: PGA0_SELGAIN =000, 1 倍放大 PGA0_SELGAIN =001, 2 倍放大 PGA0_SELGAIN =010, 4 倍放大 PGA0_SELGAIN =011, 8 倍放大

比特	名称	属性	复位值	描述
				PGA0_SELGAIN =100, 16 倍放大 PGA0_SELGAIN =101, 32 倍放大 PGA0_SELGAIN =110, 64 倍放大
5	PGA2_EN	RW	0x0	使能信号, EN_PGA2=1 时 PGA2 工作
4	PGA1_EN	RW	0x0	使能信号, EN_PGA1=1 时 PGA1 工作
3	PGA0_EN	RW	0x0	使能信号, EN_PGA0=1 时 PGA0 工作
2	PGA2_EN2P	RW	0x0	VOUT2 输出开关使能信号: 0: VOUT2PAD=HiZ 1: VOUT2PAD=VOUT2
1	PGA1_EN2P	RW	0x0	VOUT1 输出开关使能信号: 0: VOUT1PAD=HiZ 1: VOUT1PAD=VOUT1
0	PGA0_EN2P	RW	0x0	VOUT0 输出开关使能信号: 0: VOUT0PAD=HiZ 1: VOUT0PAD=VOUT0

#### 5.5.24 外部复位端口选择寄存器 SCU\_EXTRST\_SEL (偏移: 0D0h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	RESETN_SEL	RW	0x0	外部复位端口选择寄存器。只有该寄存器的【31:16】的高 16 位写 0xA5A5 时才能写这个 bit。 1: 外部复位信号无效。该管脚可以作为 PA2 其他功能使用。 0: 外部复位信号有效。

#### 5.5.25 停止模式选择寄存器 SCU\_STOP\_SEL (偏移: 0D4h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	STOPMODE_SEL	RW	0x0	停止模式选择寄存器。只有该寄存器的【31:16】的高 16 位写 0xA5A5 时才能写这个 bit。 1: STOP mode 停止模式有效 0: STOP mode 停止模式无效

### 5.5.26 软件复位寄存器 SCU\_SOFT\_RSTN (偏移: 0D8h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	GRSTN	W	0x0	向此寄存器写 0, 芯片会产生一个复位。

### 5.5.27 中断向量地址重映射寄存器 SCU\_VET\_OFFSET (偏移: 0DCh)

比特	名称	属性	复位值	描述
31:10	VECTOROFFSET	RW	0x0	中断向量重映射功能使能后, 中断向量的基地址是为{VECTOROFFSET,10{1'b0}}
9:1	RSV	-	-	保留
0	VECTOROFFSET_EN	RW	0x0	中断向量重映射功能使能: 0: 不使能中断向量重映射功能 1: 使能中断向量重映射功能

### 5.5.28 LVD 中断寄存器 SCU\_LVD\_INTR (偏移: FCh)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	LVD_INTR	W1C	0x0	LVD 中断寄存器: 1: LVD 产生中断 0: LVD 未产生中断 软件写 1 清 0。

### 5.5.29 VREF 控制寄存器 SCU\_VREF (偏移: 188h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:1	VREF_SEL	RW	0x0	VREF 选择: 00: 1.2V 01: 3.6V 10: 4.2V 11: 模拟电源电压
0	VREF_EN	RW	0x0	VREF 使能

## 6 EFC

### 6.1 概述

芯片上集成了32KB的eFlash存储器，用于保存芯片所有的关键脱机信息和数据。EFC为eFlash控制器，在CPU的配合下，完成Flash读、写、擦除等操作。

### 6.2 主要特性

- 支持 eFlash 的读写（8/16/32bits）、sector 擦除和 chip 擦除等操作流程
- 读等待时间可以配置
- 主区有 64 个 sector，每个 512 字节
- 支持擦写保护功能
- 支持自动锁总线功能
- 支持多层次防护机制确保 eFlash 数据安全性、完整性与可靠性

### 6.3 eFlash 数据防护机制

- SWD 硬件保护

通过配置 EEPROM 中的“SWD\_DSIABLE\_PIN”关键字硬件禁止 SWD 接口，禁止 SWD 接口通信，从而禁止通过 SWD 接口的物理访问或非法调试，防止恶意代码注入或固件篡改。

- SWD 读保护

通过配置 EEPROM 中的“SWD\_READ\_DISABLE”关键字禁止通过 SWD 接口对 eFlash 存储器的读操作，保护固件代码及敏感数据免遭提取。客户只有在本次上电过程中，对 eFlash 的 Main 区进行擦除后，才能在本次上电过程中解锁 eFlash 的读操作。

- eFlash 分区域写保护

对 eFlash 32KB Main 区存储空间进行分区管理,通过配置 EEPROM 中的关键字“OTP\_MAIN\_EN”对不同区域的数据进行写保护。

- ECC 功能

通过纠错码技术检测并纠正 eFlash 存储单元的读取数据错误,确保数据完整性,防止因硬件老化或环境干扰导致的读取数据损坏。

- EEPROM 分区域写保护

对 EEPROM 存储空间进行分区管理,通过配置 OTP 中的关键字“OTP\_EEP0\_EN”、“OTP\_EEP1\_EN”、“OTP\_EEP2\_EN”对不同区域的数据进行擦写保护。OTP 关键字使能后,EEPROM 对应区域的值将无法被改写,其中包括 SWD\_READ\_DISABLE、SWD\_DSIABLE\_PIN、OTP\_MAIN\_EN 等关键字。

### 6.3.1 eFlash 存储区域

eFlash存储空间主要分为Main区,EEPROM区以及NVR区。Main区为32KB大小,主要用于应用程序和应用数据存储。EEPROM区用于一些应用数据存储,对Main区的擦除操作,不会改变EEPROM区内存储的数据。NVR区为芯片出厂后的一些配置信息存储区域,芯片出厂后内部存储值无法被擦写更改。

0x0101 0600	NVR3
0x0101 0400	NVR2
0x0101 0200	NVR1
0x0101 0000	Reserved
0x0001 1000	EEPROM8
0x0001 0E00	EEPROM7
0x0001 0C00	EEPROM6
0x0001 0A00	EEPROM5
0x0001 0800	EEPROM4
0x0001 0600	EEPROM3
0x0001 0400	EEPROM2
0x0001 0200	EEPROM1
0x0001 0000	Reserved
0x0000 8000	Up to 32KB On-chip
0x0000 0000	

图 6-1: eFlash 存储区域

## 6.3.2 EEPROM 配置字

下图为EEPROM8内的关键字配置图。

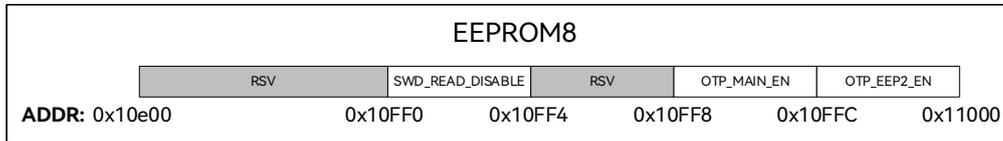


图 6-2: EEPROM8 关键字配置图

当SWD\_READ\_DISABLE值为0x55aaaa55时，芯片在经过外部管脚复位或者重新上电后，SWD读保护功能生效。此功能生效后，通过SWD读取eFlash中的数据，无论eFlash中数据为何值，读取出来的值都为0xFFFFFFFF。只有在当前外部复位释放和上电期内，对eFlash Main区发起ChipErase操作后，才能关闭此功能。在SWD\_READ\_DISABLE值保持0x55aaaa55不变时，经过外部复位或者重新上下电后，SWD读保护功能再次生效。

OTP\_MAIN\_EN为eFlash分区域写保护使能位。OTP\_MAIN\_EN[15:0]每个Bit代表2KB的Main区域是否需要写保护。32KB的Main区被划分为16个区域，每个区域由OTP\_MAIN\_EN[15:0]的一个Bit进行设定是否允许软件擦写，对应位为1，代表对应区域擦写禁止，0为允许擦写。OTP\_MAIN\_EN[0]控制地址0x0 ~ 0x800的区域，OTP\_MAIN\_EN[1]控制地址0x800 ~ 0x1000的区域，依次类推，OTP\_MAIN\_EN[15]控制地址0xF800 ~ 0x10000区域。OTP\_MAIN\_EN[15:0]内存储的关键字为与OTP\_MAIN\_EN[31:16]互为取反关系时，OTP\_MAIN\_EN[15:0]关键字才生效，否则OTP\_MAIN\_EN[15:0]关键字内不管为0还是为1，均不影响eFlash Main区的擦写。

OTP\_EEP2\_EN是EEPROM8擦写保护位。当其值为0x55aaaa55时，芯片在经过外部复位或上下电后，即开启EEPROM8擦写保护，包括SWD\_READ\_DISABLE、OTP\_MAIN\_EN、OTP\_EEP2\_EN关键字在内的EEPROM8区域，将无法被擦写。

## 6.3.3 SWD 硬件禁止

SWD硬件禁止关键字SWD\_DISABLE\_PIN位于NVR1内，需要调用芯片SDK中已有的库函数使能此功能。使能此功能后，芯片SWD相关的IO与CPU接口SWD信号之间将不再联通，硬件上禁止SWD接口。

## 6.4 寄存器描述

寄存器基地址：0x0110\_0000

表 6-1: EFC 寄存器列表

偏置	名称	描述
0x00	EFC_CTRL	EFC控制寄存器
0x04	EFC_OTPSTATUS	EFC OTP状态寄存器
0x08	EFC_KEY	EFC KEY输入寄存器
0x0c	EFC_INTSTATUS	EFC中断状态寄存器
0x10	EFC_INTEN	EFC中断使能寄存器
0x14	EFC_EP_CTRL	EFC擦写控制寄存器
0x1C	EFC_LDOTRIM	EFC LDO TRIM寄存器
0x20	EFC_RCHTRIM	EFC RCH TRIM寄存器
0x24	EFC_RCLTRIM	EFC RCL TRIM寄存器
0x28	EFC_PGA0TRIM	EFC PGA0 TRIM寄存器
0x2C	EFC_PGA1TRIM	EFC PGA1 TRIM寄存器
0x30	EFC_ADCTRIM	EFC ADC TRIM寄存器

### 6.4.1 EFC 控制寄存器 (EFC\_CTRL) (偏移: 00h)

比特	名称	属性	默认值	功能描述
31:21	RSV	-	-	保留
20	VMON3	RW	0x0	VMON3模式设置位: 1: VMON3模式使能 0: VMON3模式禁止 只用于测试, 不对客户开放
19	VMON2	RW	0x0	VMON2模式设置位: 1: VMON2模式使能 0: VMON2模式禁止 只用于测试, 不对客户开放
18	VMON1	RW	0x0	VMON1模式设置位: 1: VMON1模式使能 0: VMON1模式禁止 只用于测试, 不对客户开放

比特	名称	属性	默认值	功能描述
17	READM1	RW	0x0	READM1模式设置位： 1: READM1模式使能 0: READM1模式禁止 只用于测试，不对客户开放
16	READM0	RW	0x0	READM0模式设置位： 1: READM0模式使能 0: READM0模式禁止 只用于测试，不对客户开放
15:12	RSV	-	-	保留
11	PRFM_EN	RW	0x0	Prefetch模式带多个（4个）buf缓存： 1: Prefetch buf mul Mode模式使能 0: Prefetch buf mul Mode模式禁止
10	PRFS_EN	RW	0x0	Prefetch模式带单个buf缓存： 1: Prefetch buf Mode模式使能 0: Prefetch buf Mode模式禁止
9	KEYERR_CLR	RW	0x0	擦写Flash时，输入KEY不匹配， ERROR状态后解除控制寄存器： 1: 解除ERROR锁定状态 0: 维持当前状态不变
8	PRF_EN	RW	0x0	Prefetch模式设置位： 1: Prefetch Mode模式使能 0: Prefetch Mode模式禁止
7:1	RSV	-	-	保留
0	RD_WAIT	RW	0x1	Flash 读等待周期配置： 0: 等待 0 个系统时钟周期 1: 等待 1 个系统时钟周期

#### 6.4.2 EFC OTP 状态寄存器 (EFC\_OTPSTATUS) (偏移: 04h)

比特	名称	属性	默认值	功能描述
31:16	OTP_MAIN	R	0x0	MAIN区OTP擦写锁定状态位。每位代表Flash Main区可擦写标志，控制2K Bytes区域： OTP_MAIN[0]: 此比特为1，0~2KBytes空间不允许擦写 ... OTP_MAIN[15]: 此比特为1，30~32KBytes空间不允许擦写
15:9	RSV	-	-	保留

比特	名称	属性	默认值	功能描述
8	OTP_NVR2	R	0x0	NVR2区OTP擦写锁定状态位: 1: NVR2区不可擦写 0: NVR2区可以擦写
7	OTP_NVR1	R	0x0	NVR1区OTP擦写锁定状态位: 1: NVR1区不可擦写 0: NVR1区可以擦写
6	OTP_NVR0	R	0x0	NVR0区OTP擦写锁定状态位: 1: NVR0区不可擦写 0: NVR0区可以擦写
5	OTP_EEP2	R	0x0	EEPROM区第8个sector OTP擦写锁定状态位: 1: EEPROM第8个sector不可擦写 0: EEPROM第8个sector可以擦写
4	OTP_EEP1	R	0x0	EEPROM区第7个sector OTP擦写锁定状态位: 1: EEPROM第7个sector不可擦写 0: EEPROM第7个sector可以擦写
3	OTP_EEP0	R	0x0	EEPROM区第6个sector OTP擦写锁定状态位: 1: EEPROM第6个sector不可擦写 0: EEPROM第6个sector可以擦写
2:0	RSV	-	-	保留

### 6.4.3 EFC KEY 输入寄存器 (EFC\_KEY) (偏移: 08h)

比特	名称	属性	默认值	功能描述
31:0	KEY	W	0x0	Flash擦写Key输入寄存器，软件或者SWD在启动擦写前必须正确地向此地址写入合法KEY序列。空地址，物理上无寄存器实现。

### 6.4.4 EFC 中断状态寄存器 (EFC\_INTSTATUS) (偏移: 0Ch)

比特	名称	属性	默认值	功能描述
31:16	MAIN_ERR	W1C	0x0	MAIN区发生操作错误。当对应的MAIN区Lock住时，对此MAIN区域进行写擦操作或者Chip Erase操作时，此位均会置1。 MAIN_ERR[0]: 此比特为1, 0~2KBytes空间发生操作错误 ... MAIN_ERR[15]: 此比特为1, 30~32KBytes空间发生操

比特	名称	属性	默认值	功能描述
				作错误 对应bit位软件写1清0。
15:12	RSV	-	-	保留
11:9	KEY_STATE	R	0x0	3'b000: 默认状态 3'b001: chip_erase操作KEY正确 3'b010: sector_erase操作KEY正确 3'b011: normal_prog操作KEY正确 3'b100: KEY有误
8	NVR2_ERR	W1C	0x0	1: NVR区发生操作错误。当对应的NVR2区Lock住时, 对此NVR2区域进行写擦操作或者Chip Erase操作时, 此位均会置1 0: 正常状态 软件写1清0。
7	NVR1_ERR	W1C	0x0	1: NVR区发生操作错误。当对应的NVR1区Lock住时, 对此NVR1区域进行写擦操作或者Chip Erase操作时, 此位均会置1 0: 正常状态 软件写1清0。
6	NVR0_ERR	W1C	0x0	1: NVR区发生操作错误。当对应的NVR0区Lock住时, 对此NVR0区域进行写擦操作或者Chip Erase操作时, 此位均会置1 0: 正常状态 软件写1清0。
5	EEP2_ERR	W1C	0x0	1: EEPROM区发生操作错误。当对应的EEPROM2区Lock住时, 对此EEPROM2区域进行写擦操作或者Chip Erase操作时, 此位均会置1 0: 正常状态 软件写1清0。
4	EEP1_ERR	W1C	0x0	1: EEPROM区发生操作错误。当对应的EEPROM1区Lock住时, 对此EEPROM1区域进行写擦操作或者Chip Erase操作时, 此位均会置1 0: 正常状态 软件写1清0。
3	EEP0_ERR	W1C	0x0	1: EEPROM区发生操作错误。当对应的EEPROM0区Lock住时, 对此EEPROM0区域进行写擦操作或者Chip Erase操作时, 此位均会置1 0: 正常状态

比特	名称	属性	默认值	功能描述
				软件写1清0。
2	KEY_ERR	W1C	0x0	FLASH擦写过程中输入的KEY有误状态位： 1: 输入的KEY有误 0: 输入的KEY正确 软件写1清0。
1	PR_DONE	W1C	0x0	写完成中断状态位： 1: 写完成，写1清除该位。如果中断允许，则产生中断。 0: 写未完成。
0	ER_DONE	W1C	0x0	擦除完成中断状态位： 1: 擦除完成，写1清除该位。如果中断允许，则产生中断。 0: 擦除未完成。

#### 6.4.5 EFC 中断使能寄存器 (EFC\_INTEN) (偏移: 10h)

比特	名称	属性	默认值	功能描述
31:16	MAIN_ERR_IEN	RW	0x0	MAIN ERROR中断使能位： 1: 使能中断 0: 禁止中断 MAIN_ERR_IEN[0]: 此比特为1, 0~2KBytes空间使能中断 ... MAIN_ERR_IEN[15]: 此比特为1, 30~32KBytes空间使能中断
15:9	RSV	-	-	保留
8	NVR2_ERR_IEN	RW	0x0	NVR2 ERROR中断使能位： 1: 使能中断 0: 禁止中断
7	NVR1_ERR_IEN	RW	0x0	NVR1 ERROR中断使能位： 1: 使能中断 0: 禁止中断
6	NVR0_ERR_IEN	RW	0x0	NVR0 ERROR中断使能位： 1: 使能中断 0: 禁止中断
5	EEP2_ERR_IEN	RW	0x0	EEP2 ERROR中断使能位： 1: 使能中断 0: 禁止中断

比特	名称	属性	默认值	功能描述
4	EEP1_ERR_IEN	RW	0x0	EEP1 ERROR中断使能位: 1: 使能中断 0: 禁止中断
3	EEP0_ERR_IEN	RW	0x0	EEP0 ERROR中断使能位: 1: 使能中断 0: 禁止中断
2	KEY_ERR_IEN	RW	0x0	KEY ERROR中断使能位: 1: 使能中断 0: 禁止中断
1	PR_DONE_IEN	RW	0x0	写完成中断使能位: 1: 使能中断 0: 禁止中断
0	ER_DONE_IEN	RW	0x0	擦除完成中断使能位: 1: 使能中断 0: 禁止中断

#### 6.4.6 EFC 擦写控制寄存器 (EFC\_EP\_CTRL) (偏移: 14h)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9	ER_T	RW	0x0	Flash 擦除类型配置: 0: Sector Erase 1: Chip Erase
8:2	RSV	-	-	保留
1	PR_B	RW	0x0	Program Request: 软件置位, 硬件完成编程后自动清零
0	ER_B	RW	0x0	Erase Request: 软件置位, 硬件完成擦除后自动清零

#### 6.4.7 EFC LDO TRIM 寄存器 (EFC\_LDOTRIM) (偏移: 1Ch)

比特	名称	属性	默认值	功能描述
31:5	RSV	-	-	保留
4:0	LDO_TRIM	RW	0x10	LDO TRIM值

### 6.4.8 EFC RCH TRIM 寄存器 (EFC\_RCHTRIM) (偏移: 20h)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9:0	RCH_TRIM	RW	0x200	64M RCH TRIM 值

### 6.4.9 EFC RCL TRIM 寄存器 (EFC\_RCLTRIM) (偏移: 24h)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9:0	RCL_TRIM	RW	0x200	32k RCL TRIM 值

### 6.4.10 EFC PGA0 TRIM 寄存器 (EFC\_PGA0TRIM) (偏移: 28h)

比特	名称	属性	默认值	功能描述
31:13	RSV	-	-	保留
12:8	PGA1_TRIM	RW	0x0	-
7:5	-	-	-	保留
4:0	PGA0_TRIM	RW	0x0	-

### 6.4.11 EFC PGA1 TRIM 寄存器 (EFC\_PGA1TRIM) (偏移: 2Ch)

比特	名称	属性	默认值	功能描述
31:5	RSV	-	-	保留
4:0	PGA2_TRIM	RW	0x0	-

### 6.4.12 EFC ADC TRIM 寄存器 (EFC\_ADCTRIM) (偏移: 30h)

比特	名称	属性	默认值	功能描述
31:13	RSV	-	-	保留
12:0	ADC_TRIM	RW	0x0	ADC TRIM 值: [12]: 补偿方向 0: 正向补偿 1: 反向补偿 [11:0]: ADC 输入补偿

## 6.5 软件流程

### 6.5.1 Write 操作

eFlash上电稳定后可以执行写操作。写操作之前需要向EFC\_KEY寄存器内写对应密钥, 否则EFC忽略此次写操作。

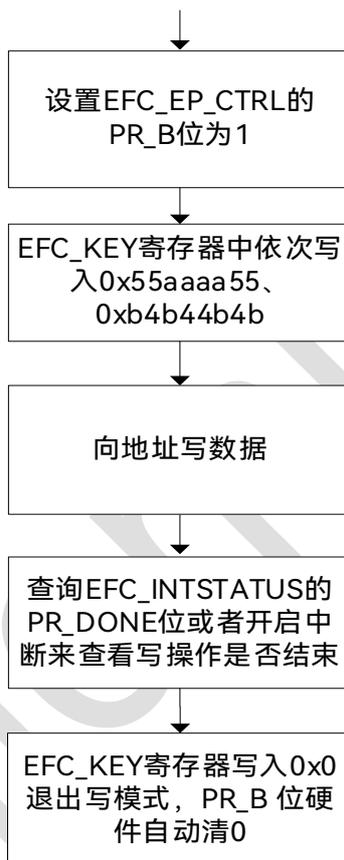


图 6-3: 写操作流程

## 6.5.2 Erase 操作



图 6-4: Sector 擦除操作流程

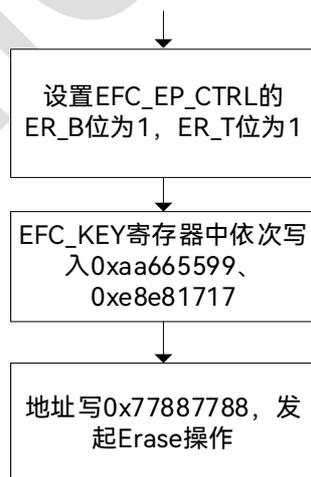


图 6-5: Chip 擦除操作流程

# 7 NVIC

## 7.1 概述

内嵌套向量中断控制器 (NVIC) 是 Cortex-M0+ 的一个重要组成部分。它与 CPU 处理器内核紧密耦合，实现低中断延迟以及对新到中断的有效处理，外部中断信号连接到 NVIC，NVIC 将对这些中断进行优先级排序。

Cortex-M0+ 处理器内置了嵌套向量中断控制器 (NVIC)，可支持最多 21 个中断请求 (IRQ) 输入：有 4 个中断优先级，可处理复杂逻辑，能进行实时控制和中断处理。

所有的 NVIC 寄存器只能采用字传输。任何试图读/写半字或字节的结果都是不可预知的。

该模块以最小的中断延迟提供灵活的中断管理功能。

(关于 NVIC 更详细的内容可查看 Cortex-M0+ 系列内核的相关官方文档)

## 7.2 主要特性

- 32 个外部中断，每个中断具有 4 级优先级
- 中断唤醒控制器，支持极低功耗休眠模式

## 7.3 中断源

表 7-1: 中断源

中断号	中断源	备注
[0]	GPIO_PA	-
[1]	GPIO_PB	-
[2]	GPIO_PC	-
[3]	GPIO_PD	-
[4]	DMA	-
[5]	UART0	-
[6]	COMP0	-

中断号	中断源	备注
[7]	COMP1	-
[8]	I2C	-
[9]	SPI	-
[10]	GTIMER0	-
[11]	GTIMER1	-
[12]	GTIMER2	-
[13]	LPTIMER	-
[14]	ANALOG	-
[15]	ATIMER	-
[16]	WDT	-
[17]	ADC	-
[18]	WWDT	-
[19]	UART1	-
[20]	EFC	-

## 8 GPIO

### 8.1 概述

GPIO 包含通用数据输入输出接口，这些管脚可以与其他功能管脚共享，这取决于芯片的配置。通过这些数据接口，可以配置任意数目的管脚作为中断信号。芯片有 4 组 GPIO，分别是 GPIOA、GPIOB、GPIOC、GPIOD 分别简称为 PA、PB、PC、PD。GPIO 的相关寄存器的功能需要设置对应的比特位，例如设置 PA1 方向为输出，GPIO\_DIR 的 bit[1]控制位需要设置为 1，其他位的设置遵循此原则，也即是 PAx 对应寄存器 GPIO\_DIR 的 bit[x]控制位。

### 8.2 主要特性

- 所有输入/输出引脚方向都可以通过软件进行配置
- 每个 GPIO\_IN 引脚可配置成边沿或电平方式触发中断
- 每个 GPIO\_IN 引脚可以选择是否电平滤波

### 8.3 寄存器描述

GPIOA 寄存器基地址: 0x4000\_4000

GPIOB 寄存器基地址: 0x4000\_4400

GPIOC 寄存器基地址: 0x4000\_4800

GPIOD 寄存器基地址: 0x4000\_4C00

表 8-1: GPIO 寄存器列表

偏置	名称	描述
0x00	GPIO_DIR	GPIO 数据方向寄存器
0x08	GPIO_SET	GPIO 输出置位寄存器
0x0C	GPIO_CLR	GPIO 输出清零寄存器
0x10	GPIO_ODATA	GPIO 输出引脚映射寄存器

偏置	名称	描述
0x14	GPIO_IDATA	GPIO 输入引脚映射寄存器
0x18	GPIO_IEN	GPIO 中断使能寄存器
0x1C	GPIO_IS	GPIO 中断触发模式寄存器
0x20	GPIO_IBE	GPIO 中断边沿触发设置寄存器
0x24	GPIO_IEV	GPIO 中断高低电平触发设置寄存器
0x28	GPIO_IC	GPIO 中断状态清除寄存器
0x2C	GPIO_RIS	GPIO 原始中断状态寄存器
0x30	GPIO_MIS	GPIO 屏蔽后中断状态寄存器
0x34	GPIO_FLS	GPIO 输入滤波选择寄存器

### 8.3.1 GPIO 数据方向寄存器 (GPIO\_DIR) (偏移: 00h)

比特	名称	属性	复位值	描述
7:0	DIR	RW	0x0	8 位寄存器，GPIO 输入输出控制寄存器： 0: 输入 1: 输出

注：GPIOx\_DIR[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_DIR[1]与 PA1 对应。

### 8.3.2 GPIO 输出置位寄存器 (GPIO\_SET) (偏移: 08h)

比特	名称	属性	复位值	描述
7:0	SET	W	0x0	8 位寄存器，GPIO 输出置位寄存器： 0: 无效操作 1: 当 IO 为输出时，IO 置位

注：GPIOx\_SET[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_SET [1]与 PA1 对应。

### 8.3.3 GPIO 输出清零寄存器 (GPIO\_CLR) (偏移: 0Ch)

比特	名称	属性	复位值	描述
7:0	CLR	W	0x0	8 位寄存器，GPIO 输出清零寄存器： 0: 无效操作 1: 当 IO 为输出时，IO 清零

注：GPIOx\_CLR [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_CLR [1]与 PA1 对应。

### 8.3.4 GPIO 输出引脚映射寄存器 (GPIO\_ODATA) (偏移: 10h)

比特	名称	属性	复位值	描述
7:0	ODATA	RW	0x0	8 位寄存器，GPIO 输出引脚映射寄存器： 当 GPIO 方向为输出有效，写操作直接写至外部引脚，读获得外部引脚值。

注：GPIOx\_ODATA [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_ODATA [1]与 PA1 对应。

### 8.3.5 GPIO 输入引脚映射寄存器 (GPIO\_IDATA) (偏移: 14h)

比特	名称	属性	复位值	描述
7:0	IDATA	R	0x0	8 位寄存器，GPIO 输入引脚映射寄存器： 当 GPIO 方向为输入有效，读获得外部引脚值；此寄存器为只读寄存器。

注：GPIOx\_IDATA [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IDATA [1]与 PA1 对应。

### 8.3.6 GPIO 中断使能寄存器 (GPIO\_IEN) (偏移: 18h)

比特	名称	属性	复位值	描述
7:0	IEN	RW	0x0	8 位寄存器，GPIO 中断使能寄存器： 0：禁止相应引脚中断 1：使能相应引脚中断

注：GPIOx\_IEN [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IEN [1]与 PA1 对应。

### 8.3.7 GPIO 中断触发模式寄存器 (GPIO\_IS) (偏移: 1Ch)

比特	名称	属性	复位值	描述
7:0	IS	RW	0x0	7 位寄存器, GPIO 中断触发模式: 0: 边沿检测 1: 电平检测

注: GPIOx\_IS[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_IS[1]与 PA1 对应。

### 8.3.8 GPIO 中断边沿触发设置寄存器 (GPIO\_IBE) (偏移: 20h)

比特	名称	属性	复位值	描述
7:0	IBE	RW	0x0	8 位寄存器, GPIO 中断边沿触发设置: 0: 单边沿触发 1: 双边沿触发

注: GPIOx\_SET[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_SET[1]与 PA1 对应。

### 8.3.9 GPIO 中断高低电平触发设置寄存器 (GPIO\_IEV) (偏移: 24h)

比特	名称	属性	复位值	描述
7:0	IEV	RW	0x0	8 位寄存器, GPIO 中断高低电平触发设置: 0: 下降沿/低电平触发 1: 上升沿/高电平触发

注: GPIOx\_IEV[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_IEV[1]与 PA1 对应。

### 8.3.10 GPIO 中断状态清除寄存器 (GPIO\_IC) (偏移: 28h)

比特	名称	属性	复位值	描述
7:0	IC	W	0x0	8 位寄存器, GPIO 中断清除寄存器: 0: 无效操作 1: 清除对应引脚中断

注：GPIOx\_IC [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IC [1] 与 PA1 对应。

### 8.3.11 GPIO 原始中断状态寄存器 (GPIO\_RIS) (偏移: 2Ch)

比特	名称	属性	复位值	描述
7:0	RIS	R	0x0	8 位寄存器，GPIO 原始中断寄存器： 0: 对应引脚无中断挂起 1: 对应引脚有中断挂起

注：GPIOx\_RIS [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_RIS [1] 与 PA1 对应。

### 8.3.12 GPIO 屏蔽后中断状态寄存器 (GPIO\_MIS) (偏移: 30h)

比特	名称	属性	复位值	描述
7:0	MIS	R	0x0	32 位寄存器，GPIO 屏蔽后中断状态寄存器：反映对应引脚屏蔽后的中断状态。

注：GPIOx\_MIS [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_MIS [1] 与 PA1 对应。

### 8.3.13 GPIO 输入滤波选择寄存器 (GPIO\_FLS) (偏移: 34h)

比特	名称	属性	复位值	描述
7:0	FLS	RW	0x0	8 位寄存器，GPIO 输入滤波选择寄存器： 反映对应引脚输入是否需要滤波。 1: 使能滤波 0: 不使能滤波

注：GPIOx\_FLS [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_FLS [1] 与 PA1 对应。

## 8.4 使用流程

### 8.4.1 输入输出 IO

1. 配置 GPIO\_DIR 寄存器，选择 GPIO 方向。
2. 可使用 GPIO\_SET/GPIO\_CLR 或 GPIO\_ODATA 来设置输出电平。
3. 配置 GPIO\_FLS 选择输入的电平是否需要滤波。
4. 使用 GPIO\_IDATA 来获取输入引脚电平。

### 8.4.2 中断触发模式

中断初始化过程:

1. 设置 GPIO\_DIR 为输入。
2. 清除 GPIO\_IEN 以避免异常。
3. 配置寄存器 GPIO\_IS，选择是边沿/电平触发类型。
4. 在单边沿触发方式下，配置寄存器 GPIO\_IBE，确定是单边触发还是双边触发。
5. 在单边沿触发方式下，配置寄存器 GPIO\_IEV，确定是哪种边沿触发类型。
6. 在电平触发方式下，配置寄存器 GPIO\_IEV，确定是哪种电平触发类型。
7. 配置寄存器 GPIO\_IC 来清除中断。
8. 配置寄存器 GPIO\_IEN 使能相应位中断。

### 8.4.3 清除中断

ISR 写 GPIO\_IC 来清除中断状态。如果在清除寄存器的同时有新的边沿触发中断产生，这个新的中断将会保持有效直到下一次清除。读取中断状态操作应该在清 GPIO\_IEN 之前进行，清 GPIO\_IEN 操作将清除相应中断状态。

#### 8.4.4 输入滤波

在非低功耗模式下，可以通过配置 GPIO\_FLS 寄存器来选择对应的引脚输入电平是否需要数字滤波功能，若关闭则电平信号直接输出，若开启则需要改输入电平维持至少 3 个系统时钟的稳定电平才会进入 GPIO 内被检测。

## 9 UART0/1

### 9.1 概述

UART0和UART1串口模块，支持全双工数据交换，支持与外部接口设备的串行通信。

### 9.2 主要特性

- 提供标准的异步通讯位（起始位、奇偶位和停止位）
  - 生成 1 位起始位
  - 支持 8bit 的数据位宽
  - 生成 1 位校验位(可设置奇校验或偶校验)，或无校验位
  - 生成 1/2 位停止位
  - 字节从低位到高位依次传输
- 可编程波特率
- 可支持 9600bps、19200bps、115200bps 等常见波特率的传输
- UART0 支持 TX 端口单线传输模式

### 9.3 寄存器描述

UART0 寄存器基地址：0x4000\_0000

UART1 寄存器基地址：0x4000\_0040

表 9-1: UART0/1 寄存器列表

偏置	名称	描述
0x00	UART_ISR	中断状态寄存器
0x04	UART_IER	中断使能寄存器
0x08	UART_CR	控制寄存器
0x0C	UART_TDR	发送数据寄存器

偏置	名称	描述
0x10	UART_RDR	接收数据寄存器
0x14	UART_BRR	波特率参数低位寄存器
0x18	UART_DLTOR	延迟超时寄存器

### 9.3.1 中断状态寄存器 UART\_ISR (偏移: 00h)

比特	名称	属性	复位值	描述
31:19	RSV	-	-	保留
18	PE	W1C	0x0	奇偶校验错误标志, 软件写 1 清零。
17	FE	W1C	0x0	帧格式错误标志, 软件写 1 清零
16	OVERE	W1C	0x0	接收缓存器溢出错误标志, 软件写 1 清零或读取 RDR 时清零
15:12	RSV	-	-	保留
11	RXTO	W1C	0x0	接收超时标志, 软件写 1 清零
10:9	RSV	-	-	保留
8	RXF	W1C	0x0	接收缓存器满标志, 软件写 1 清零或读取 RDR 时清零
7	NEGWKF	W1C	0x0	下降沿唤醒标志, 软件写 1 清零
6:2	RSV	-	-	保留
1	TXE	R	0x1	发送缓存空标志, 软件写入 TDR 时清零
0	TC	W1C	0x0	发送完成标志, 软件写 1 清零或者软件写入 TDR 时清零

### 9.3.2 中断使能寄存器 UART\_IER (偏移: 04h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11	RXTOIE	RW	0x0	接收超时中断使能, 1 有效
10	RXERRIE	RW	0x0	接收错误中断使能, 1 有效
9	RSV	-	-	保留
8	RXFIE	RW	0x0	接收缓存满中断使能, 1 有效
7	NEGWKIE	RW	0x0	下降沿唤醒使能, 1 有效
6:2	RSV	-	-	保留
1	TXIE	RW	0x0	发送缓存空中断使能, 1 有效
0	TCIE	RW	0x0	发送完成标志中断使能, 1 有效

### 9.3.3 控制寄存器 UART\_CR (偏移: 08h)

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留
13	BUSY	R	0x0	UART 数据传输繁忙标志: 0: 空闲状态 1: UART 正在传输数据
12	RXTOEN	RW	0x0	接收数据超时使能: 0: 不使能 1: 使能
11	NEGEWKE	RW	0x0	下降沿唤醒使能: 0: 不使能 1: 使能
10	TX_EN	RW	0x0	UART 单线模式使能控制: 0: 不使能 1: 使能
9	TX_OEN	RW	0x0	UART 单线模式 TX 管脚数据传输方向控制: 0: TX 管脚作为数据输出脚 1: TX 管脚作为数据输入脚
8	STOPLEN	RW	0x0	停止位宽度配置: 0: 1 位停止位 1: 2 位停止位
7: 6	DATALEN	RW	0x1	每帧的数据长度选择: 00: 7 位数据 01: 8 位数据 10: 9 位数据 11: 预留
5: 4	PARITYSL	RW	0x0	校验位配置: 00: 无校验位 01: 偶校验 10: 奇校验 11: 预留
3	RXPOL	RW	0x0	接收数据极性配置: 0: 正向 1: 取反

比特	名称	属性	复位值	描述
2	TXPOL	RW	0x0	发送数据极性配置: 0: 正向 1: 取反
1	RXEN	RW	0x0	接收使能, 1 有效
0	TXEN	RW	0x0	发送使能, 1 有效

### 9.3.4 发送数据寄存器 UART\_TDR (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:0	TDR	W	0x0	存放待发送的数据

### 9.3.5 接收数据寄存器 UART\_RDR (偏移: 10h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:0	RDR	R	0x0	存放待接收到的数据

### 9.3.6 波特率参数低位寄存器 UART\_BRR (偏移: 14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	BRR	RW	0x340	波特率产生器寄存器值 波特率计算公式: 当 BRR<16 时, $F=UART\_CLK/16$ 当 BRR≥16 时, $F=UART\_CLK/(BRR+1)$

### 9.3.7 延迟超时寄存器 UART\_DLTOR (偏移: 18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	TXDLY	RW	0x0	发送数据延迟时间, 单位为 1 时钟周期
7:0	RXDLY	RW	0xFF	接收数据超时时间, 单位为 1 时钟周期

## 9.4 使用流程

### 9.4.1 串口的发送和接收

1. 配置系统配置寄存器的串口模块时钟。
2. 配置系统配置寄存器的串口引脚复用功能。
3. 配置串口中断。
4. 配置串口中断使能寄存器（是否使用中断）。
5. 配置串口控制寄存器（奇偶校验位等）。
6. 配置波特率。
7. 使能串口。

### 9.4.2 串口初始化

1. 清除 UART\_ISR 寄存器，写 0 清除。
2. 配置 UART\_IER，中断使能寄存器，是否产生相应的中断脉冲。
3. 清除 UART\_CR 寄存器，写 0 清除。
4. 配置 UART\_BRR。

### 9.4.3 串口发送字节

1. 发送、接收数据前软件可以配置波特率参数，奇偶校验类型、中断使能。
2. 设置 UART\_CR.TXEN=1。
3. 写入第一个字节数据到 UART\_TDR。
4. 查询发送完成标志 UART\_ISR.TC，如果 TC=1 表示当前数据发送完成；软件清除此位（写 0 清除）。
5. 可以继续写入下一个字节到 UART\_TDR。

#### 9.4.4 串口接收字节

1. 发送、接收数据前软件可以配置波特率参数、奇偶校验类型、中断使能。
2. 接收数据，查询 UART\_ISR 标志位或者等待中断。
3. 读取 UART\_RDR。
4. 继续接收数据。

# 10 I2C

## 10.1 概述

I2C 总线接口连接微控制器和串行 I2C 总线。I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。I2C 模块通过数据引脚 SDA 和时钟引脚 SCL 连接到 I2C 总线，控制所有 I2C 总线规定的时序。本模块支持主模式和从模式。

## 10.2 主要特征

- 支持标准（100Kbps）/快速（400Kbps）/高速（1Mbps）三种工作速率
- 支持 7 位寻址功能和 10 位寻址功能
- 支持中断查询功能
- 低功耗从机设计，可以在没有系统时钟的情况下收发数据
- 支持异步从机地址匹配唤醒、数据帧接收完成唤醒或 START 检测唤醒

## 10.3 寄存器描述

I2C 寄存器基地址：0x4000\_5400

表 10-1: I2C 寄存器列表

偏置	名称	描述
0x00	I2C_MCFG	I2C 主机配置寄存器
0x04	I2C_MCON	I2C 主机控制寄存器
0x08	I2C_MIE	I2C 主机中断使能寄存器
0x0C	I2C_MIF	I2C 主机中断标志寄存器
0x10	I2C_MSTA	I2C 主机状态寄存器
0x14	I2C_MBRG	I2C 主机波特率寄存器
0x18	I2C_MBUF	I2C 主机收发缓存寄存器
0x1C	I2C_MTIM	I2C 主机时序控制寄存器

偏置	名称	描述
0x20	I2C_MTO	I2C 主机超时寄存器
0x24	I2C_SCON	I2C 从机控制寄存器
0x28	I2C_SIE	I2C 从机中断使能寄存器
0x2C	I2C_SIF	I2C 从机中断标志寄存器
0x30	I2C_SSTA	I2C 从机状态寄存器
0x34	I2C_SBUF	I2C 从机收发缓存寄存器
0x38	I2C_SADR0	I2C 从机地址寄存器 0
0x3C	I2C_SADR1	I2C 从机地址寄存器 1
0x40	I2C_SADR2	I2C 从机地址寄存器 2
0x44	I2C_SADR3	I2C 从机地址寄存器 3

### 10.3.1 I2C 主机配置寄存器 I2C\_MCFG (偏移: 00h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	MAUST	RW	0x0	DMA 传输自动回复 STOP 配置位: 1: DMA 指定长度传输完成后, 自动发送 STOP 时序 0: DMA 指定长度传输完成后, 等待软件接管
2	MDMAEN	RW	0x0	主机 DMA 使能: 0: 关闭 DMA 功能 1: 使能 DMA 功能
1	MTOEN	RW	0x0	SCL 拉低超时使能: 1: 使能超时功能 0: 关闭超时功能
0	MSPEN	RW	0x0	I2C 主机模块使能控制位: 1: 主机使能 0: 主机禁止

### 10.3.2 I2C 主机控制寄存器 I2C\_MCON (偏移: 04h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	MREN	RW	0x0	主机接收模式下, 接收使能位: 1: 主机接收使能 0: 接收禁止 主机通信中, 软件在发送完地址字节后, 通过置位

比特	名称	属性	复位值	描述
				MREN 将传输方向切换为主机接收，然后可以接收来自从机的数据。
2	MSTS	RW	0x0	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零
1	MRSTRTS	RW	0x0	Repeat START 时序产生使能控制位，软件写 1 发送 Repeat START 时序，发送完成后硬件自动清零
0	MSTRTS	RW	0x0	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零

### 10.3.3 I2C 主机中断使能寄存器 I2C\_MIE (偏移: 08h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	MWCIE	RW	0x0	MWCIF 中断使能寄存器: 1: 允许写冲突中断 0: 禁止写冲突中断
5	MTOIE	RW	0x0	SCL 超时中断使能寄存器: 1: 允许超时中断 0: 禁止超时中断
4	MSIE	RW	0x0	START 时序中断使能寄存器: 1: 允许 START 时序中断 0: 禁止 START 时序中断
3	MSPIE	RW	0x0	STOP 时序中断使能寄存器: 1: 允许 STOP 时序中断 0: 禁止 STOP 时序中断
2	MNAIE	RW	0x0	主机发送模式下 NACK 中断使能寄存器: 1: 允许收到 NACK 产生中断 0: 禁止产生 NACK 中断
1	MTXIE	RW	0x0	I2C 主机发送完成中断使能: 1: 允许发送完成中断 0: 禁止发送完成中断
0	MRXIE	RW	0x0	I2C 主机接收完成中断使能: 1: 允许接收完成中断 0: 禁止接收完成中断

### 10.3.4 I2C 主机中断标志寄存器 I2C\_MIF (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	MWCIF	W1C	0x0	写冲突检测位, MCU 只能在完成 START 时序或发送完成一帧读写之后才能写 MBUF, 否则发生写冲突; 硬件置位, 软件写 1 清零 1: 发送写冲突 0: 未发生冲突
5	MTOIF	W1C	0x0	SCL 超时中断标志, 仅在 TOEN 为 1 时工作, 软件写 1 清零 1: 发生 SCL 超时 0: 没有发生 SCL 超时
4	MSIF	W1C	0x0	START 时序发送完成中断标志, 硬件置位, 软件写 1 清零
3	MSPIF	W1C	0x0	STOP 时序发送完成中断标志, 硬件置位, 软件写 1 清零
2	MNAIF	W1C	0x0	主控发送模式下, 来自从机的回应信号; 当主机发送后收到 NACK, 此标志可以产生中断; 硬件置位, 软件写 1 清零。 1: 从机回应 NACK 0: 从机回应 ACK
1	MTXIF	W1C	0x0	I2C 主机发送完成中断标志, 硬件置位, 软件写 1 清零。 此标志寄存器在主机接收完从机回发的 ACK 或 NACK 后置位。
0	MRXIF	W1C	0x0	I2C 主机接收完成中断标志, 硬件置位, 软件写 1 清零。 此标志寄存器在主机回发完 ACK 或 NACK 后置位。

### 10.3.5 I2C 主机状态寄存器 I2C\_MSTA (偏移: 10h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5	MBUSY	R	0x0	I2C 通信状态位: 1: 接口处于读写状态, 正在进行数据传输 0: 已完成数据传输
4	MRW	R	0x0	I2C 传输方向状态位: 1: 主机从从机读取数据 0: 主机向从机写入数据
3	RSV	-	-	保留
2	MBUFF	R	0x0	缓冲器满状态位: 接收: 1: 接收完成, MBUF 满

比特	名称	属性	复位值	描述
				0: 接收未完成, MBUF 空 发送: 1: 正在发送, MBUF 满 0: 发送完成, MBUF 空
1	RSV	-	-	保留
0	MACKST	RW	0x0	主控接收模式下, 主机回应信号的状态: 1: 主机回发 NACK 0: 主机回发 ACK

### 10.3.6 I2C 主机波特率寄存器 I2C\_MBRG (偏移: 14h)

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
24:16	MBRGH	RW	0x13	主机发送的 SCL 时钟高电平宽度, 以 PCLK 计数
15:9	RSV	-	-	保留
8:0	MBRGL	RW	0x13	主机发送的低电平宽度, 以 PCLK 计数

### 10.3.7 I2C 主机收发缓存寄存器 I2C\_MBUF (偏移: 18h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	MBUF	RW	0x0	I2C 主机数据传输寄存器

### 10.3.8 I2C 主机时序控制寄存器 I2C\_MTIM (偏移: 1Ch)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
8:0	MDY	RC	0x4	定义 SDA 相对于 SCL 下降沿的保持时间参数, 以 I2C 工作时钟计数 注: 最小有效值为 1, 最大有效值为 MBRGL

### 10.3.9 I2C 主机超时寄存器 I2C\_MTO (偏移: 20h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	MTO	RW	0xFFF	定义从机 SCL 低电平延展超时周期, 软件可以在 MSPEN 为 0 的情况下改写

### 10.3.10 I2C 从机控制寄存器 I2C\_SCON (偏移: 24h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	SSCLLEN	RW	0x1	I2C 从机时钟延展使能: 0: 禁止 slave clock stretching 1: 使能 slave clock stretching
5	SDMAEN	RW	0x0	I2C 从机 DMA 使能: 1: 使能 DMA 功能 0: 关闭 DMA 功能
4	SACKEN	RW	0x1	ACK 使能位: 1: Slave 接收完成后将回发 ACK 0: Slave 不回发 ACK
3	SSDADYEN	RW	0x0	SDA 从机输出延迟使能: 1: 使能从机 SDA 输出延迟 0: 从机 SDA 输出延迟
2	SSCLLVEN	RW	0x0	SCL 从机输入模拟滤波使能: 1: 使能模拟滤波 0: bypass 模拟滤波
1	AD10EN	RW	0x0	10 位地址使能: 1: Slave 使用 10bit 地址 0: Slave 使用 7bit 地址
0	SSPEN	RW	0x0	I2C 从机使能位: 1: 使能 I2C 从机 0: 关闭 I2C 从机

### 10.3.11 I2C 从机中断使能寄存器 I2C\_SIE (偏移: 28h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	AD3IE	RW	0x0	从机地址 3 匹配中断使能: 1: 从地址 3 匹配中断使能 0: 从地址 3 匹配中断禁止
11	AD2IE	RW	0x0	从机地址 2 匹配中断使能: 1: 从地址 2 匹配中断使能 0: 从地址 2 匹配中断禁止
10	AD1IE	RW	0x0	从机地址 1 匹配中断使能: 1: 从地址 1 匹配中断使能 0: 从地址 1 匹配中断禁止
9	AD0IE	RW	0x0	从机地址 0 匹配中断使能: 1: 从地址 0 匹配中断使能 0: 从地址 0 匹配中断禁止
8	RSV	-	-	保留
7	ADER	RW	0x0	从机地址错误中断使能: 1: 地址错误中断使能 0: 地址错误中断禁止
6	SSIE	RW	0x0	从机 Start 中断使能: 1: Start 中断使能 0: Start 中断禁止
5	SSPIE	RW	0x0	从机 Stop 中断使能: 1: Stop 中断使能 0: Stop 中断禁止
4	SWCIE	RW	0x0	从机 SWCIF 中断使能: 1: 写冲突中断使能 0: 写冲突中断禁止
3	SSOVIE	RW	0x0	从机接收溢出中断使能: 1: 接收溢出中断使能 0: 接收溢出中断禁止
2	SADIE	RW	0x0	I2C 从机地址匹配中断使能: 1: 地址匹配中断使能 0: 地址匹配中断禁止
1	STXIE	RW	0x0	I2C 从机发送完成中断使能: 1: 发送中断使能

比特	名称	属性	复位值	描述
				0: 发送中断禁止
0	SRXIE	RW	0x0	I2C 从机接收完成中断使能: 1: 接收中断使能 0: 接收中断禁止

### 10.3.12 I2C 从机中断状态寄存器 I2C\_SIF (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	AD3IF	W1C	0x0	从机地址 3 匹配标志: 1: 接收到的 7bit I2C_SADR3 寄存器内容一致 0: 接收到的地址与 I2C_SADR3 不一致 硬件置位, 软件写 1 清零。
11	AD2IF	W1C	0x0	从机地址 2 匹配标志: 1: 接收到的 7bit I2C_SADR2 寄存器内容一致 0: 接收到的地址与 I2C_SADR2 不一致 硬件置位, 软件写 1 清零。
10	AD1IF	W1C	0x0	从机地址 1 匹配标志: 1: 接收到的 7bit I2C_SADR1 寄存器内容一致 0: 接收到的地址与 I2C_SADR1 不一致 硬件置位, 软件写 1 清零。
9	AD0IF	W1C	0x0	从机地址 0 匹配标志: 1: 接收到的 7bit 或 10bit 地址与 I2C_SADR0 寄存器内容一致 0: 接收到的地址与 I2C_SADR0 不一致 硬件置位, 软件写 1 清零。
8	RSV	-	-	保留
7	ADER	W1C	0x0	从机地址格式错误: 1: 在 7 位地址情况下收到 11110 开头的地址字节, 或者在 10 位地址情况下第一个字节不以 11110 开头时, 触发 ADER 0: 未检测到地址格式错误 硬件置位, 软件写 1 清零。
6	SSIF	RC	0x0	START 中断位: 1: 从机检测到 START 0: 从机未检测到 START 硬件置位, 软件读取清零。

比特	名称	属性	复位值	描述
5	SSPIF	RC	0x0	STOP 中断位： 1: 从机检测到 STOP 0: 从机未检测到 STOP 硬件置位，软件读取清零。
4	SWCIF	W1C	0x0	写冲突标志： 1: 在 SBUFF=1 的情况下，软件向 SSPBUF 写入新的数据 0: 无写入冲突 当溢出发生时，新的数据将被丢弃。 硬件置位，软件写 1 清零。
3	SSOVIF	W1C	0x0	SBUF 溢出标志： 1: 在 SBUFF 为 1 的情况下，从机又接收到新的数据 0: 没有接收溢出 如果从机使能 SCL 延展，不会出现接收数据溢出的情况。 硬件置位，软件写 1 清零。
2	SADIF	W1C	0x0	从机地址匹配标志： 1: 接收到的 7bit 或 10bit 地址与 I2C_SADRx 寄存器内容一致 0: 接收到的地址与 I2C_SADRx 不一致 硬件置位，软件写 1 清零。
1	STXIF	W1C	0x0	从机发送完成标志： 1: 发送完成 0: 发送未完成 硬件置位，软件写 1 清零。
0	SRXIF	W1C	0x0	从机接收完成标志： 1: 接收完成 0: 接收未完成 硬件置位，软件写 1 清零。

### 10.3.13 I2C 从机状态寄存器 I2C\_SSTA (偏移: 30h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	SBUSY	R	0x0	从机通信标志： 1: 从机数据收发中 0: 从机空闲
2	SRW	R	0x0	读写方向状态寄存器： 1: 从机收到 RW 为 1，从机需要发送数据给主机

比特	名称	属性	复位值	描述
				0: 从机处于接收数据状态
1	DA	R	0x0	帧指示: 1: 上一字节收到的是数据 0: 上一字节收到的是地址
0	SBUFF	R	0x0	从机数据缓冲区满标志: 1: SBUFF 满 0: SBUFF 空

### 10.3.14 I2C 从机收发缓冲寄存器 I2C\_SBUF (偏移: 34h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	SBUF	RW	0x0	I2C 从机数据传输寄存器

### 10.3.15 I2C 从机地址寄存器 0 I2C\_SADR0 (偏移: 38h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9:0	SADR0	RW	0x0	I2C 从地址寄存器 0

### 10.3.16 I2C 从机地址寄存器 1 I2C\_SADR1 (偏移: 3Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	SADR1	RW	0x0	I2C 从地址寄存器 1
0	SADR1EN	RW	0x0	1: SADR1 使能 0: SADR1 未使能

### 10.3.17 I2C 从机地址寄存器 2 I2C\_SADR2 (偏移: 40h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	SADR2	RW	0x0	I2C 从地址寄存器 2
0	SADR2EN	RW	0x0	1: SADR2 使能 0: SADR2 未使能

**10.3.18 I2C 从机地址寄存器 3 I2C\_SADR3 (偏移: 44h)**

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	SADR3	RW	0x0	I2C 从地址寄存器 3
0	SADR3EN	RW	0x0	1: SADR3 使能 0: SADR3 未使能

## 10.4 使用流程

### 10.4.1 主机发送

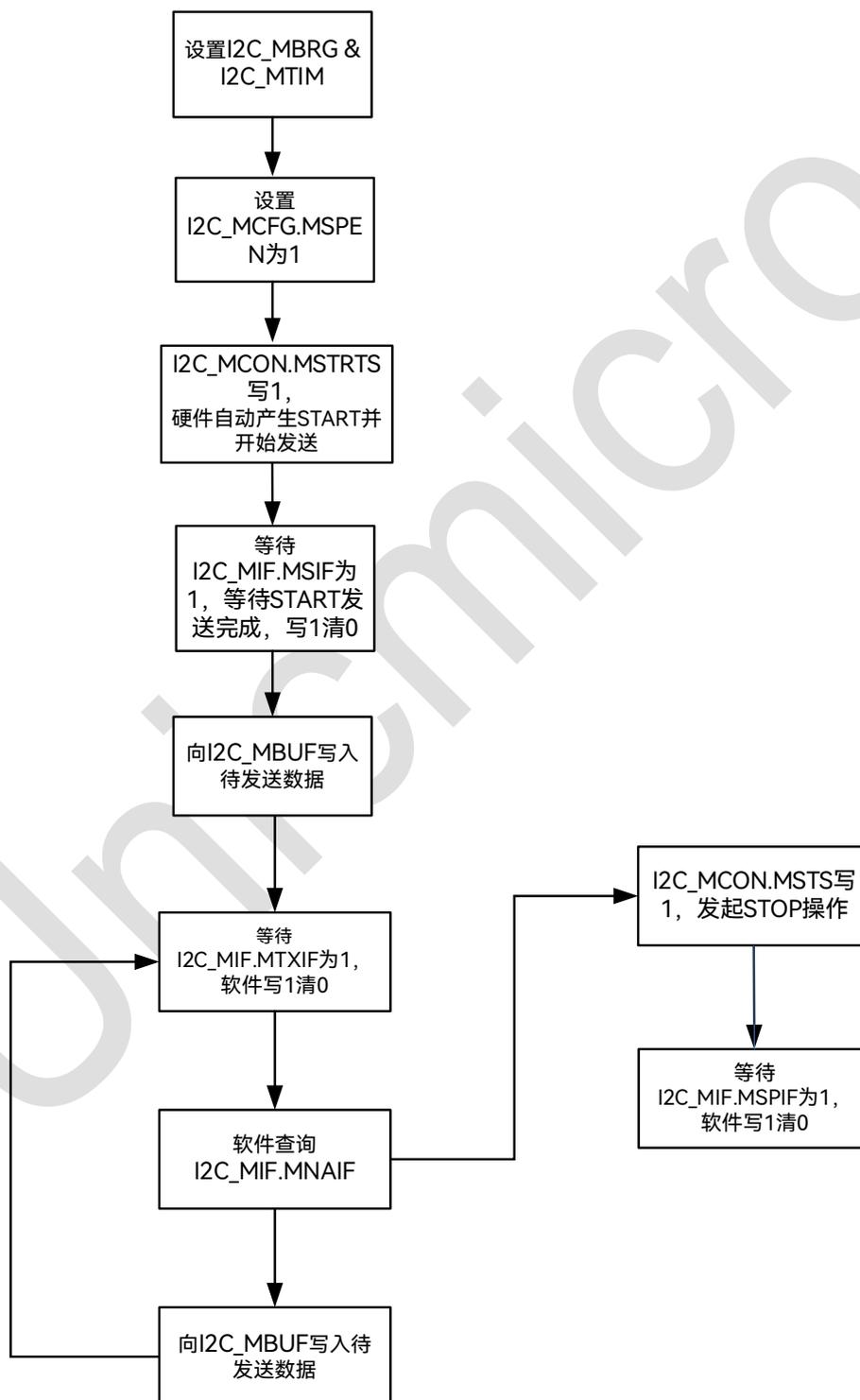


图 10-1: 主机发送流程图

## 10.4.2 主机接收

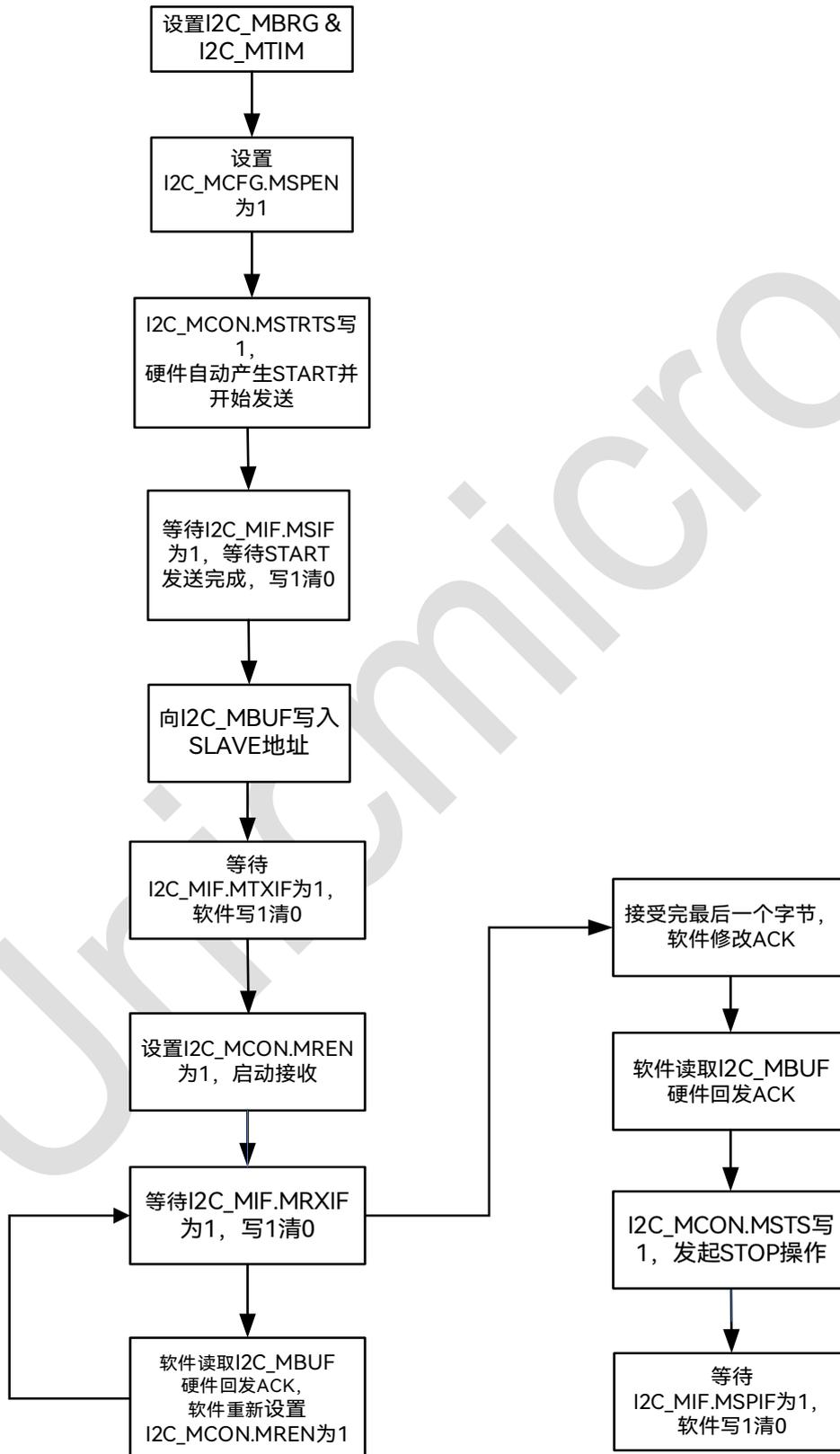


图 10-2: 主机接收流程图

### 10.4.3 从机发送

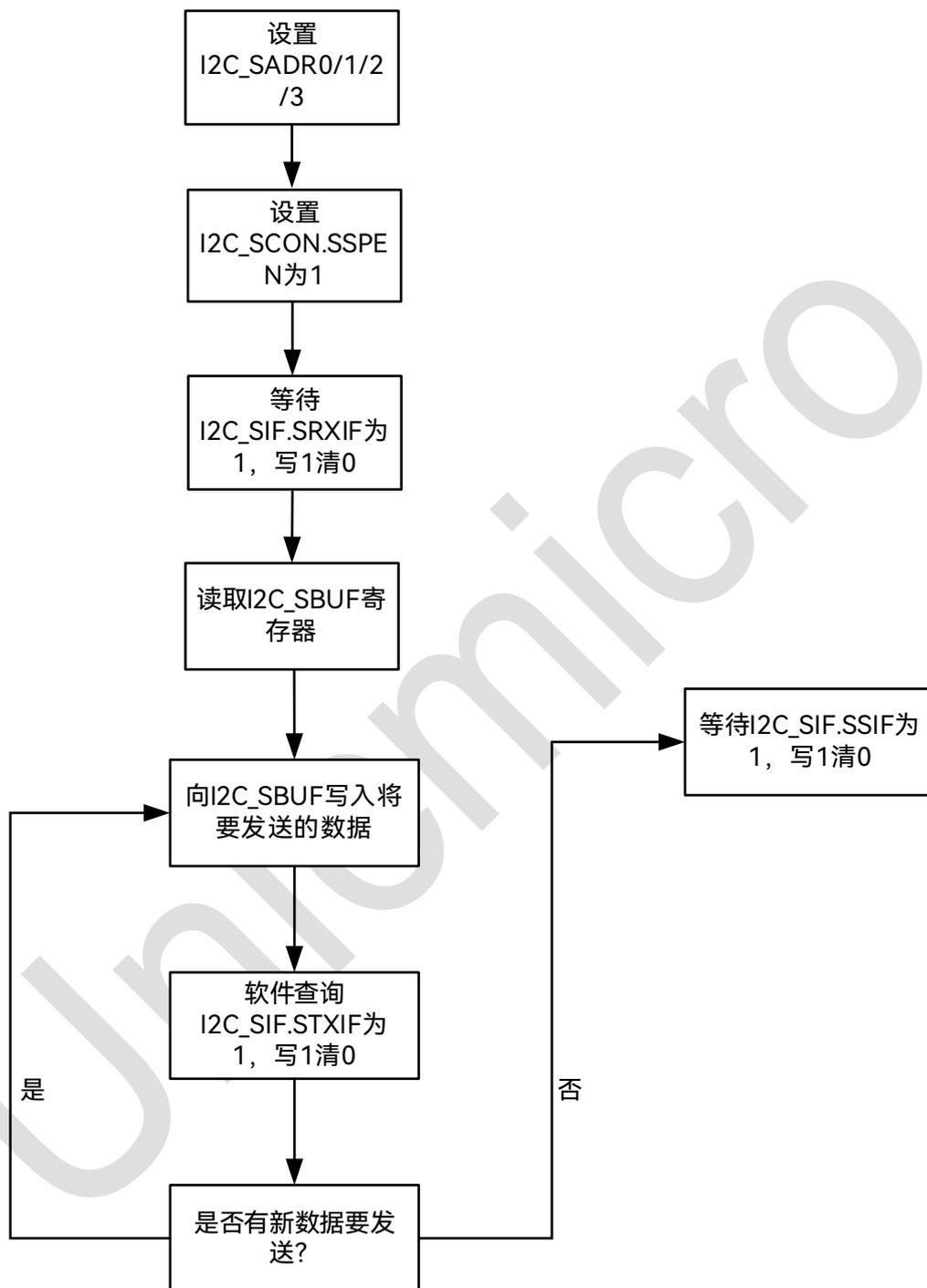


图 10-3: 从机发送流程图

### 10.4.4 从机接收

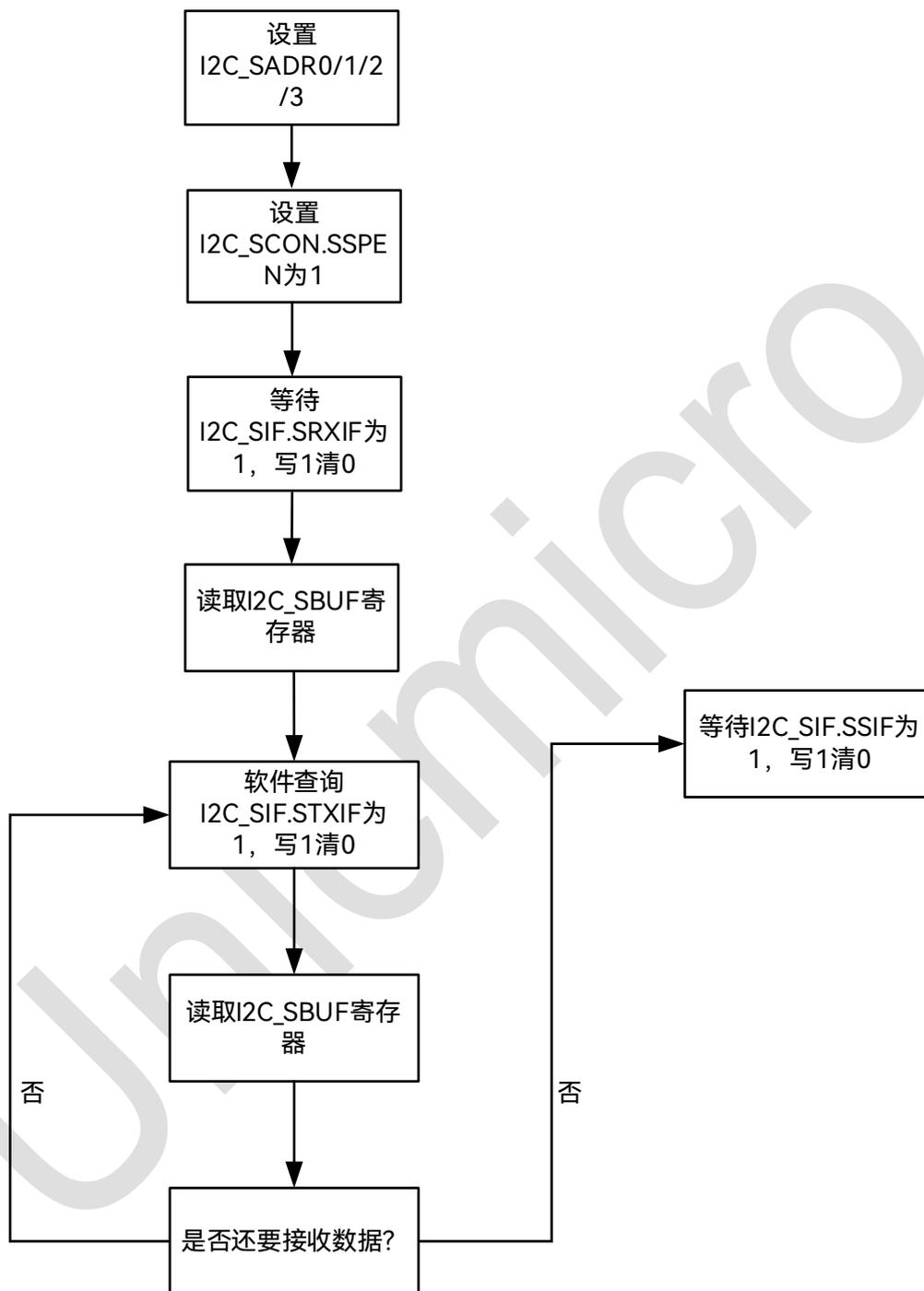


图 10-4: 从机接收流程图

# 11 SPI

## 11.1 概述

串行外设接口（Serial Peripheral Interface, SPI）是外部设备通过单线交换数据的串行同步通讯手段。芯片提供了 1 个 SPI 接口模块，可配置为主设备或从设备，实现与外部的 SPI 通信。

## 11.2 主要特性

- 全双工或半双工 3/4 线串行同步收发
- 主从模式
- 可编程时钟极性和相位（支持模式 0、1、2、3）
- 可编程比特速率
- 从模式最大频率为  $f_{sys}/2$
- 支持 6/7/8/9bits 数据位传输
- 支持 DC 控制
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持 DMA
- 深度为 8 的 9-bit 发送和接收 FIFO

## 11.3 寄存器描述

SPI 寄存器基地址: 0x4000\_0800

表 11-1: SPI 寄存器列表

偏置	名称	描述
0x00	SPI_CR	SPI 配置寄存器
0x04	SPI_CS0	SPI 主模式控制寄存器 0
0x08	SPI_CS1	SPI 主模式控制寄存器 1
0x14	SPI_OPCR	SPI 过程控制寄存器
0x18	SPI_IE	SPI 中断控制寄存器
0x1C	SPI_IF	SPI 中断标志寄存器
0x20	SPI_TXBUF	SPI 发送缓存寄存器
0x24	SPI_RXBUF	SPI 接收缓存寄存器
0x28	SPI_DMARXLEV	SPI DMA 接收设置寄存器
0x2c	SPI_DMATXLEV	SPI DMA 发送设置寄存器

### 11.3.1 SPI 配置寄存器 (SPI\_CR) (偏移: 00h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19	MOSI_POL	RW	0x0	SPI 主模式下, MOSI 对应默认电平选择: 1: MOSI 空闲时为高电平 0: MOSI 空闲时为低电平 注 1: 当通信在进行时不能改变该位的值 注 2: 当 SSN 为低时不能改变该位的值
18	DC_REG	RW	0x0	DC 输出寄存器控制位: 1: DC 输出为 1 0: DC 输出取决于 FIFO 中的第 8bit
17	DC_DIR	RW	0x0	DC 方向控制使能: 1: DC 为输出模式 0: DC 为输入模式
16	DC_EN	RW	0x0	DC 控制使能: 1: 使能 DC 模式, DC 与 MOSI 复用 0: 关闭 DC 模式

比特	名称	属性	复位值	描述
15:14	SPI_FRS	RW	0x2	SPI 帧格式设定: 0: 6-bit 格式 1: 7-bit 格式 2: 8-bit 格式 3: 9-bit 格式
13	DMA_TX_EN	RW	0x0	DMA TX 使能: 1: 使能 DMA TX 请求 0: 关闭 DMA TX 请求
12	DMA_RX_EN	RW	0x0	DMA RX 使能: 1: 使能 DMA RX 请求 0: 关闭 DMA RX 请求
11	FLTEN	RW	0x1	Slave 输入管脚滤波使能 (SSN/SCK/MOSI): 1: 使能 4ns 滤波 0: 不滤波
10	SSNM	RW	0x0	Master 模式下 SSN 控制模式选择: 1: 每发送完 8bits 后 Master 拉高 SSN, 维持高电平时间由 WAIT 寄存器控制 0: 每发送完 8bits 后 Master 保持 SSN 为低, 维持低电平时间由 WAIT 寄存器控制
9	TXO_AC	RW	0x1	TXONLY 硬件自动清零的使能: 1: TXONLY 硬件自动清零有效, 软件使能 TXO 后, 等待发送完毕后, 硬件清零 0: 关闭 TXONLY 硬件自动清零
8	TXO	RW	0x0	TXONLY 控制位: 1: 启动 Master 的单发送模式 0: 关闭单发送模式
7	MSPA	RW	0x0	Master Sampling Position Adjustment, Master 对 MISO 信号的采样位置调整, 用于高速通信时补偿 PCB 走线延迟: 1: 采样点延迟半个 SCK 周期 0: 不调整
6	SSPA	RW	0x0	Slave Sending Position Adjustment, Slave MISO 发送位置调整: 1: 提前半个 SCK 周期发送 0: 不调整

比特	名称	属性	复位值	描述
5	MM	RW	0x0	Master/Slave 模式选择: 1: Master 模式 0: Slave 模式
4:3	WAIT	RW	0x0	Master 模式下, 每发完 8bits 后加入至少 (1+WAIT) 个 SCK cycle 等待时间再传输下一个 8bits 的数据
2	RSV	-	-	RFU: 未实现, 读为 0
1	SSNSEN	RW	0x0	Master 模式下, 软件控制 SSN 使能: 1: Master 模式下 SSN 输出由软件控制 0: Master 模式下 SSN 输出由硬件自动控制, SSN0/ SSN1 此时需要置 1
0	SPIEN	RW	0x0	SPI 使能, 采用关闭时钟的方式来关闭使能: 1: 使能 SPI0 0: 关闭 SPI0, 清空发送接收缓存

### 11.3.2 SPI 主模式控制寄存器 0 (SPI\_CS0) (偏移: 04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	TRIO_EN	RW	0x0	SPI 三线模式使能: 1: 使能三线模式 0: 禁止三线模式
6	SSN0	RW	0x0	SPI 主模式下, CS0 对应 Master 模式下, 如果 SSNSEN 为 1, 软件可以通过此位控制 SSN 输出电平: 1: SSN 输出低电平 0: SSN 输出高电平 注: SSNSEN 为 0 (硬件自动控制时) 时, 必须置 1。
5:3	BAUD0	RW	0x1	SPI 主模式下, CS0 对应 Master 模式波特率配置位 (通信速率最高设置为 30M) : 0: $f_{PCLK}/2$ 1: $f_{PCLK}/4$ 2: $f_{PCLK}/8$ 3: $f_{PCLK}/16$ 4: $f_{PCLK}/32$ 5: $f_{PCLK}/64$ 6: $f_{PCLK}/128$

比特	名称	属性	复位值	描述
				7: $f_{PCLK}/256$ 当通信正在进行的时候, 不能修改这些位。
2	LSBF0	RW	0x0	SPI 主模式下, CS0 对应帧格式 (Frame format) : 0: 先发送 MSB 1: 先发送 LSB 注: 当通信在进行时不能改变该位的值。
1	CPOLO	RW	0x0	SPI 主模式下, CS0 对应时钟极性选择: 1: 串行时钟停止在高电平 0: 串行时钟停止在低电平 注 1: 当通信在进行时不能改变该位的值。 注 2: 当 SSN 为低时不能改变该位的值
0	CPHA0	RW	0x0	SPI 主模式下, CS0 对应时钟相位选择: 1: 第二个时钟边沿是第一个捕捉边沿 0: 第一个时钟边沿是第一个捕捉边沿 注: 当通信在进行时不能改变该位的值。

### 11.3.3 SPI 主模式控制寄存器 1 (SPI\_CS1) (偏移: 08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	TRI1_EN	RW	0x0	SPI 三线模式使能: 1: 使能三线模式 0: 禁止三线模式
6	SSN1	RW	0x0	SPI 主模式下, CS1 对应 Master 模式下, 如果 SSNSEN 为 1, 软件可以通过此位控制 SSN 输出电平: 1: SSN 输出低电平 0: SSN 输出高电平 注: SSNSEN 为 0 (硬件自动控制时) 时, 必须置 1。
5:3	BAUD1	RW	0x1	SPI 主模式下, CS1 对应 Master 模式波特率配置位 (通信速率最高设置为 30M) : 0: $f_{PCLK}/2$ 1: $f_{PCLK}/4$ 2: $f_{PCLK}/8$ 3: $f_{PCLK}/16$ 4: $f_{PCLK}/32$ 5: $f_{PCLK}/64$

比特	名称	属性	复位值	描述
				6: $f_{PCLK}/128$ 7: $f_{PCLK}/256$ 当通信正在进行的时候, 不能修改这些位。
2	LSBF1	RW	0x0	SPI 主模式下, CS1 对应帧格式 (Frame format) : 0: 先发送 MSB 1: 先发送 LSB 注: 当通信在进行时不能改变该位的值。
1	CPOL1	RW	0x0	SPI 主模式下, CS1 对应时钟极性选择: 1: 串行时钟停止在高电平 0: 串行时钟停止在低电平 注 1: 当通信在进行时不能改变该位的值。 注 2: 当 SSN 为低时不能改变该位的值
0	CPHA1	RW	0x0	SPI 主模式下, CS1 对应时钟相位选择: 1: 第二个时钟边沿是第一个捕捉边沿 0: 第一个时钟边沿是第一个捕捉边沿 注: 当通信在进行时不能改变该位的值。

### 11.3.4 SPI 过程控制寄存器 (SPI\_OPCR) (偏移: 14h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留 读为 0
5	SSNNEGC	W1C	0x0	SSN Negedge Clear, 软件写 1 清除 SSN 下降沿中断, 写 0 无效
4	SSNPOSC	W1C	0x0	SSN Posedge Clear, 软件写 1 清除 SSN 上升沿中断, 写 0 无效
3	TXBFC	W1C	0x0	Transmit Buffer Clear, 软件写 1 清除发送缓存, 写 0 无效
2	RXBFC	W1C	0x0	Receive Buffer Clear, 软件写 1 清除接收缓存, 写 0 无效
1	MERRC	W1C	0x0	Master Error Clear, 软件写 1 清除 SPI_IF.MERR 位
0	SERRC	W1C	0x0	Slave Error Clear, 软件写 1 清除 SPI_IF.SERR 位

### 11.3.5 SPI 中断控制寄存器 (SPI\_IE) (偏移: 18h)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留 读为 0
10	SSNNEGIE	RW	0x0	SSN Negedge 中断使能: 1: 使能 0: 不使能
9	SSNPOSIE	RW	0x0	SSN Posedge 中断使能: 1: 使能 0: 不使能
8	RNFIE	RW	0x0	RX FIFO Full 中断使能: 1: 使能 0: 不使能
7	TNFIE	RW	0x0	TX FIFO Not Full 中断使能: 1: 使能 0: 不使能
6	MERRIE	RW	0x0	Master Error 中断使能: 1: 使能 0: 不使能
5	SERRIE	RW	0x0	Slave Error 中断使能: 1: 使能 0: 不使能
4	RXCOLIE	RW	0x0	接收缓存溢出中断使能: 1: 使能 0: 不使能
3	TXCOLIE	RW	0x0	发送缓存溢出中断使能: 1: 使能 0: 不使能
2	IDLEIE	RW	0x0	SPI 空闲标志中断使能: 1: 使能 0: 不使能
1	TXBEIE	RW	0x0	TX Buffer Empty 中断使能: 1: 使能 0: 不使能
0	RXBFIE	RW	0x0	RX Buffer 非空中断使能: 1: 使能 0: 不使能

### 11.3.6 SPI 中断标志寄存器 (SPI\_IF) (偏移: 1Ch)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:20	RXFIFO_LEVEL	R	0x0	RX FIFO 当前所存数据个数
19:16	TXFIFO_LEVEL	R	0x0	TX FIFO 当前所存数据个数
15:11	RSV	-	-	保留, 读为 0
10	SSNNEG	R	0x0	SSN Negedge 标志: 检测到 SSN 下降沿, SSNNEG 置位。
9	SSNPOS	R	0x0	SSN Posedge 标志: 检测到 SSN 上升沿, SSNPOS 置位。
8	RNF	R	0x0	SPI RX FIFO Full: 1: SPI0 RX FIFO 满 0: SPI0 RX FIFO 未滿
7	TNF	R	0x1	SPI TX FIFO Not Full: 1: SPI0 TX FIFO 未滿 0: SPI0 TX FIFO 满
6	MERR	R	0x0	Master Error 标志: 当 Master 下传输未滿 8 位 SSN 就被拉高时, MERR 置位。
5	SERR	R	0x0	Slave Error 标志: 当 Slave 下传输未滿 8 位 SSN 就被拉高时, SERR 置位。
4	RXCOL	W1C	0x0	接收缓存溢出: 1: 接收缓存溢出 0: 接收缓存未溢出 注: 软件写 1 清零。
3	TXCOL	W1C	0x0	发送缓存溢出: 1: 发送缓存溢出 0: 发送缓存未溢出 注: 软件写 1 清零。
2	IDLE	R	0x1	SPI0 空闲标志, 只读: 1: SPI0 传输空闲 0: SPI0 传输进行中
1	TXBE	R	0x1	TX Buffer Empty 标志位: 1: 发送缓存空, 软件写 TXBUF 清零 0: 发送缓存非空

比特	名称	属性	复位值	描述
0	RXBF	R	0x0	RX Buffer 非空标志位: 1: 接收缓存非空 0: 接收缓存空

### 11.3.7 SPI 发送缓存寄存器 (SPI\_TXBUF) (偏移: 20h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留 读为 0
8:0	TXBUF	W	0x0	SPI 发送缓存, 发送 FIFO 入口地址。此 IP 一共含有 8 个 9-bit 数据的发送 FIFO, 写此地址, 将要发送的数据写入 FIFO 中。

### 11.3.8 SPI 接收缓存寄存器 (SPI\_RXBUF) (偏移: 24h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留 读为 0
8:0	RXBUF	R	0x0	SPI 接收缓存, 接收 FIFO 入口地址。此 IP 一共含有 8 个 9-bit 数据的接收 FIFO, 读此地址, 将接收的数据从 FIFO 中读取出来。

### 11.3.9 SPI DMA 接收设置寄存器 (SPI\_DMARXLEV) (偏移: 28h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留 读为 0
2:0	DMA_RX_LEV	RW	0x0	SPI0 接收 FIFO DMA 请求设置。 当 RX FIFO 中的数据个数大于此寄存器设置值时, 产生 DMA RX 请求。

### 11.3.10 SPI DMA 发送设置寄存器 (SPI\_DMATXLEV) (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留 读为 0
3:0	DMA_TX_LEV	RW	0x0	SPI0 发送 FIFO DMA 请求设置。 当 TX FIFO 中的数据个数小于此寄存器设置值时, 产生 DMA TX 请求。

## 11.4 使用流程

SPI 两种引脚搭配方式如下表所示：

表 11-2: SPI 两种引脚搭配方式表

信号描述功能位	SPI 配置 1	SPI 配置 2
CS	SPI_CS0	SPI_CS1
MISO	SPI_MISO	SPI_MI1
MOSI	SPI_MOSI	SPI_MOSI
CLK	SPI_SCK	SPI_SCK

注:引脚 SPI\_CS1 和 SPI\_MI1 搭配使用, SPI\_CS0 与 SPI\_MISO 搭配使用。

### 11.4.1 初始化程序

1. 配置开启 SPI 模块时钟与复位 PERI\_CLKEN /PERI\_RESET。
2. 配置相应的 GPIO 引脚复用为 SPI\_SCK、SPI\_MISO、SPI\_MOSI、SPI\_CLK, 用作输入功能的引脚需配置 PADIE<sub>x</sub> 寄存器使能引脚输入。
3. 配置 SPI\_CR[5], 设置主从模式。
4. 配置 SPI\_CR[10], 设置 SSN 控制模式。
5. 配置 SPI\_CR[1], 设置 SSN 输出由软件还是硬件控制。
6. 配置 SPI\_CS<sub>x</sub>[6], 设置 SSN 输出高电平还是低电平。
7. 配置 SPI\_CS<sub>x</sub>[2], 设置先发送的是 MSB 还是 LSB。
8. 配置 SPI\_CS<sub>x</sub>[0], 设置第一个时钟边沿采样还是第二个时钟边沿采样。
9. 配置 SPI\_CS<sub>x</sub>[1], 设置串行时钟停止在高电平还是低电平。
10. 配置 SPI\_CS<sub>x</sub>.BAUD<sub>x</sub>[2:0], 以设置串行时钟波特率 (若为从器件模式则不用设置, 串行时钟速率由主器件决定)。需要中断时, 配置 SPI\_IE 使能相应的中断。
11. 配置 SPI\_CR[0], 使能 SPI。

## 11.4.2 发送流程

➤ 主器件发送流程:

配置 SPI\_CSx[6]拉低 SSN 引脚启动传输, 配置 SPI\_CR[8]位为高, 将数据写入 SPI\_TXBUF 寄存器, 等待 SPI\_IF[2]置位发送完成, 配置 SPI\_CR[8]位为低, 传输完成后将 SSN 拉高。

➤ 从器件发送流程:

配置 SPI\_CR[8]为高, 将数据写入 SPI\_TXBUF 寄存器, 等待 SPI\_IF[2]置位发送完成, 配置 SPI\_CR[8]为低。

## 11.4.3 接收流程

➤ 主器件接收流程:

配置 SPI\_CSx[6]拉低 SSN 引脚启动传输, 将数据写入 SPI\_TXBUF 寄存器, 等待 SPI\_IF[0]置位, 读取 SPI\_RXBUF 寄存器数据完成数据接收, 传输完成后将 SSN 拉高。

➤ 从器件接收流程:

等待 SPI\_IF[0]置位, 读取 SPI\_RXBUF 寄存器数据完成数据接收。

## 11.4.4 SPI DMA 发送流程

1. 配置 SPI\_DMATXLEV[2:0], 设置产生 DMA TX 请求的 FIFO 数据个数。
2. 使能 SPI\_CR[13], 使能 DMA TX 请求。
3. 配置开启 DMA 模块时钟与复位 PERI\_CLKEN / PERI\_RESET。
4. 配置通道控制信息寄存器 DMA\_CHCTRLCx, 根据实际应用配置数据位宽, 传输模式 (8 位位宽、内存到外设模式)。
5. 配置 DMA\_CHCTRLCx[4:3]和 DMA\_CHCTRLCx[2:1], 选择【目的外设】和【源外设】(目的外设为 SPI 发送, 源外设为 MEM), 此位在用到外设的模式下起效。
6. 配置 DMA\_CHCTRLCx[2:1]和 DMA\_CHCTRLCx[4:3], 选择【目的地址】和【源地址】是否随数据传输递增 (源地址递增、目的地址不变)。

7. 如需使用中断，则配置 DMA 中断屏蔽寄存器 DMA\_INTMASK，使能对应的通道中断。
8. 配置 DMA\_SRCADDRx，配置通道源地址。
9. 配置 DMA\_DSTADDRx，配置通道目的地址。
10. 配置 DMA\_CHCTRLCx[29:15]，配置传输块数量。
11. 等待上述配置、以及相应的源地址和目的地址准备就绪，使能 DMA (DMAC\_EN)。
12. 配置 DMA\_CHCTRLCx[0]，使能 DMA 通道传输。
13. 等待 DMA\_CHCTRLCx[0]为 0，传输完成。若使能了传输结束中断，则等待传输结束中断后再处理。

### 11.4.5 SPI DMA 接收流程

1. 配置 SPI\_DMARXLEV[2:0]，设置产生 DMA RX 请求的 FIFO 数据个数。
2. 使能 SPI\_CR[12]，使能 DMA RX 请求。
3. 配置开启 DMA 模块时钟与复位 PERI\_CLKEN / PERI\_RESET。
4. 配置通道控制信息寄存器 DMA\_CHCTRLCx，根据实际应用配置数据位宽、传输模式(8 位位宽、外设到内存模式)。
5. 配置 DMA\_CHCTRLCx[4:3]和 DMA\_CHCTRLCx[2:1]，选择【目的外设】和【源外设】(目的外设为 MEM，源外设为 SPI 接收)，此位在用到外设的模式下起效。
6. 配置 DMA\_CHCTRLCx[2:1]和 DMA\_CHCTRLCx[4:3]，选择【目的地址】和【源地址】是否随数据传输递增(源地址不变、目的地址递增)。
7. 如需使用中断，则配置 DMA 中断屏蔽寄存器 DMA\_INTMASK，使能对应的通道中断。
8. 配置 DMA\_SRCADDRx，配置通道源地址。
9. 配置 DMA\_DSTADDRx，配置通道目的地址。
10. 配置 DMA\_CHCTRLCx[29:15]，配置传输块数量。
11. 等待上述配置、以及相应的源地址和目的地址准备就绪，使能 DMA (DMAC\_EN)。
12. 配置 DMA\_CHCTRLCx[0]，使能 DMA 通道传输。
13. 等待 DMA\_CHCTRLCx[0]为 0，传输完成。若使能了传输结束中断，则等待传输结束中断后再处理。

# 12 ATIMER

## 12.1 概述

包含一个高级定时器。高级定时器包含一个 16 位自动重载计数器及一个可编程预分频器。高级定时器可以支持多种应用，如捕捉、输出比较、PWM、带死区插入的互补 PWM。

## 12.2 主要特性

- 16 位向上、向下、向上/下计数自动重载计数器
- 16 位可编程预分频器，支持实时调整计数时钟分频
- 4 个独立通道可用于输入捕捉、输出比较、PWM、单脉冲输出
- 可编程死区插入的互补输出
- 支持定时器间的级联
- 重复计数器，支持定时器多个循环后更新状态
- 7 路刹车引脚输入、比较器刹车、SVD 刹车，刹车信号滤波和极性选择，刹车信号组合配置
- 支持在以下事件发生时产生中断或 DMA 事件：
  - 计数器上/下溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕获
  - 输出比较
  - 刹车信号输入
- 支持增量正交编码器和霍尔传感器

## 12.3 ATIMER 结构框图

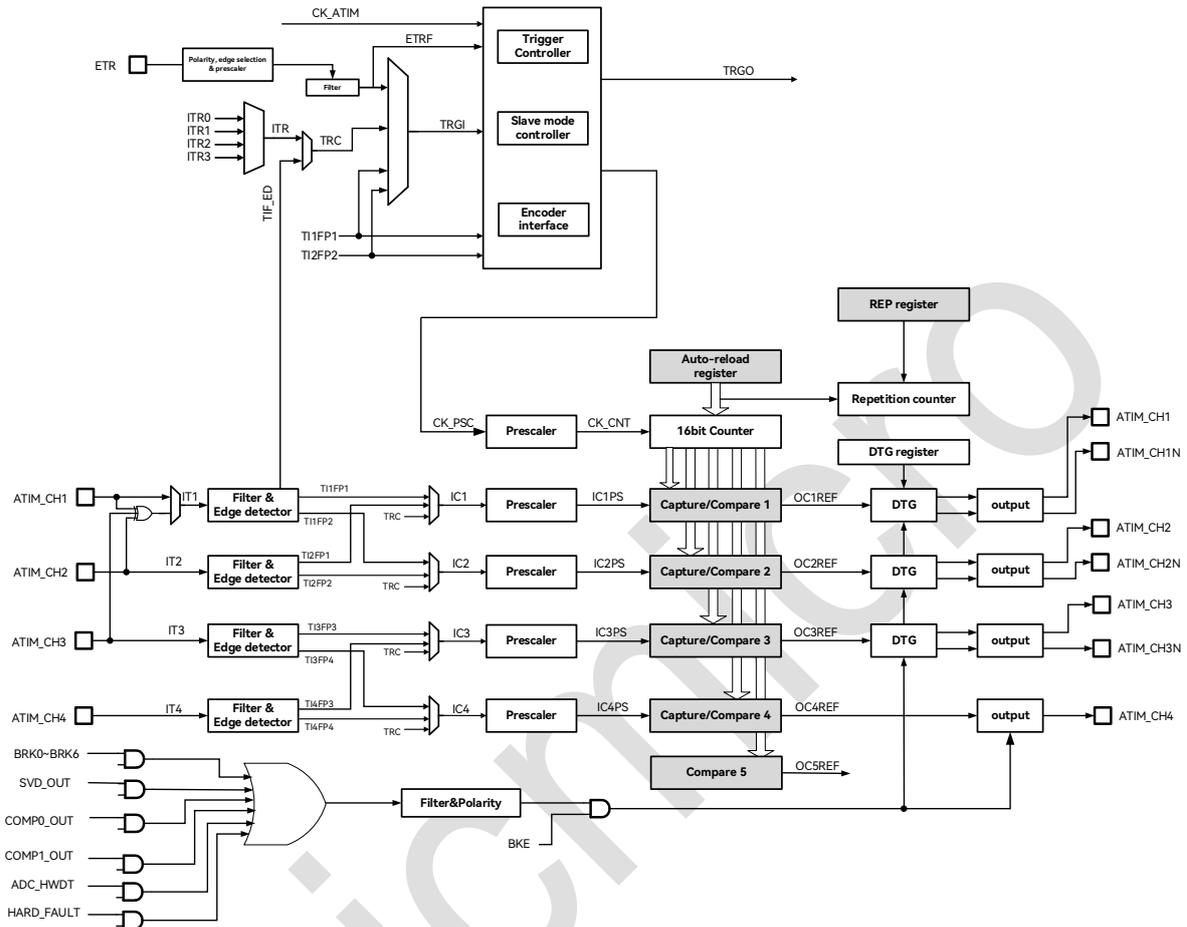


图 12-1: ATIMER 结构框图

## 12.4 功能描述

### 12.4.1 定时单元

定时单元由一个 16 位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过 16 位预分频器对时钟进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器 (ATIMER\_CNT)
- 预分频寄存器 (ATIMER\_PSC)
- 自动重载寄存器 (ATIMER\_ARR)
- 重复计数寄存器 (ATIMER\_RCR)

ATIMER\_ARR 包含预装载功能, 该功能通过 ARPE (Auto Reload Preload Enable) 寄存器控制。当 ARPE=0 时, 对 ATIMER\_ARR 寄存器执行写入, 写入数据将直接传入到影子寄存器; 当 ARPE=1 时, 对 ATIMER\_ARR 寄存器执行写入的数据在 update event (ATIMER\_CNT 上溢出或者下溢出) 发生时, 传送到影子寄存器。软件也可以通过寄存器操作主动触发 ATIMER\_ARR 更新 (UEV)。

ATIMER\_CNT 工作时钟由 ATIMER\_PSC 产生的分频时钟驱动, 只有在计数器使能寄存器 (CEN) 置位时, ATIMER\_CNT 才开始计数。当 ATIMER\_CNT=ATIMER\_ARR 时, 本轮计数结束, 发送 update event。

ATIMER\_PSC 是一个同步预分频器, 能够对时钟进行 1~65536 分频。ATIMER\_PSC 寄存器同样被缓存, 改写 ATIMER\_PSC 实际不改写影子寄存器, 只有当新的 update event 到来时, 才会从 ATIMER\_PSC 更新至影子寄存器。因此在 ATIMER\_CNT 计数过程中, 软件可以实时改写 ATIMER\_PSC, 而新的预分频比将在下一更新事件发生时被采用。

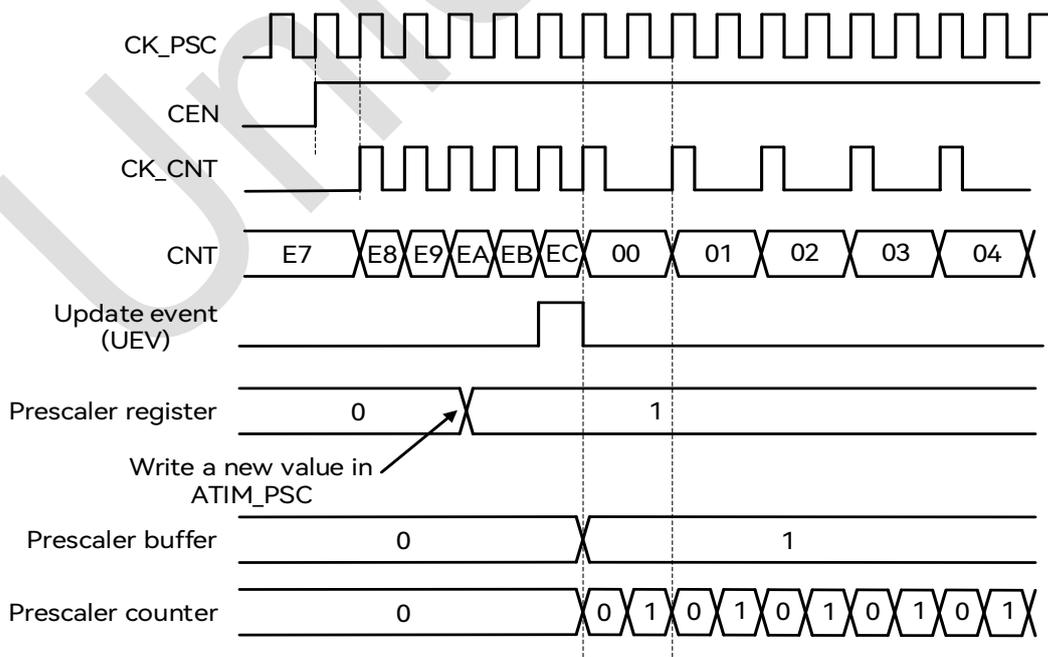


图 12-2: 预分频从 1 变为 2 的波形

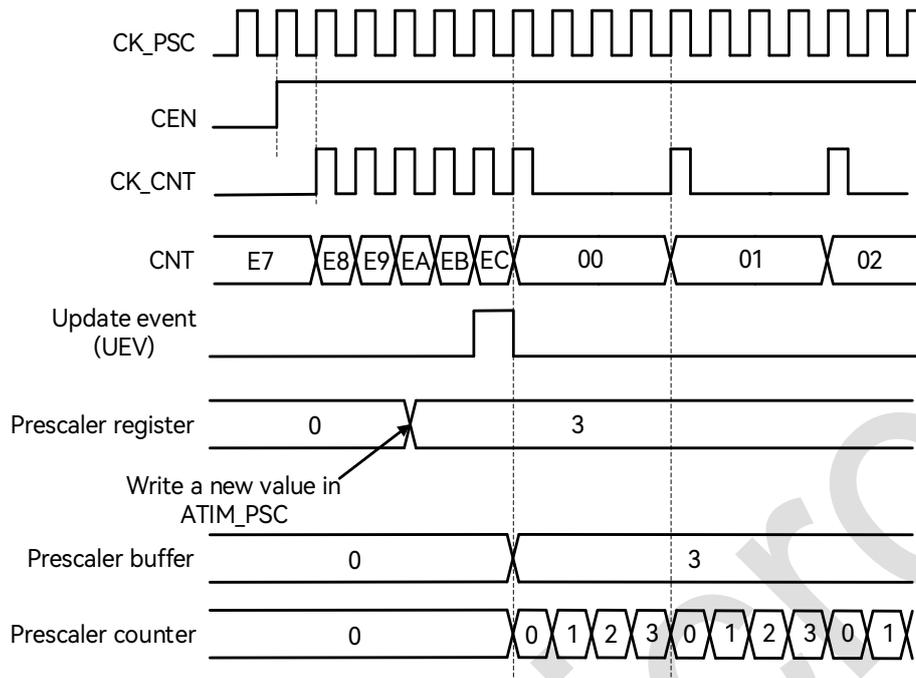


图 12-3: 预分频从 1 变为 4 的波形

## 12.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

## 12.4.3 向上计数

此模式中，计数器使能后从 0 开始计数，直到  $ATIMER\_CNT=ATIMER\_ARR$ ，产生溢出事件，然后重新从 0 开始计数。

如果使能了重复计数功能，则计数器按照  $ATIMER\_RCR$  的定义重复上述过程若干次 ( $ATIMER\_RCR+1$ )，才会产生溢出事件。

软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器自动清零。设置 UG 寄存器是否触发 UIF (Update Interrupt Flag) 中断标志置位由 URS 寄存器的设置决定。

通过设置 UDIS 寄存器可以禁止 update event，这样可以避免将 preload 寄存器中的值更新到工作寄存器中。

当 update event 发生时，以下寄存器被更新，并且 UIF 置位：

- ATIMER\_RCR 影子寄存器被更新为 ATIMER\_RCR 内容
- ATIMER\_ARR 影子寄存器被更新为 ATIMER\_ARR 内容
- ATIMER\_PSC 影子寄存器被更新为 ATIMER\_PSC 内容

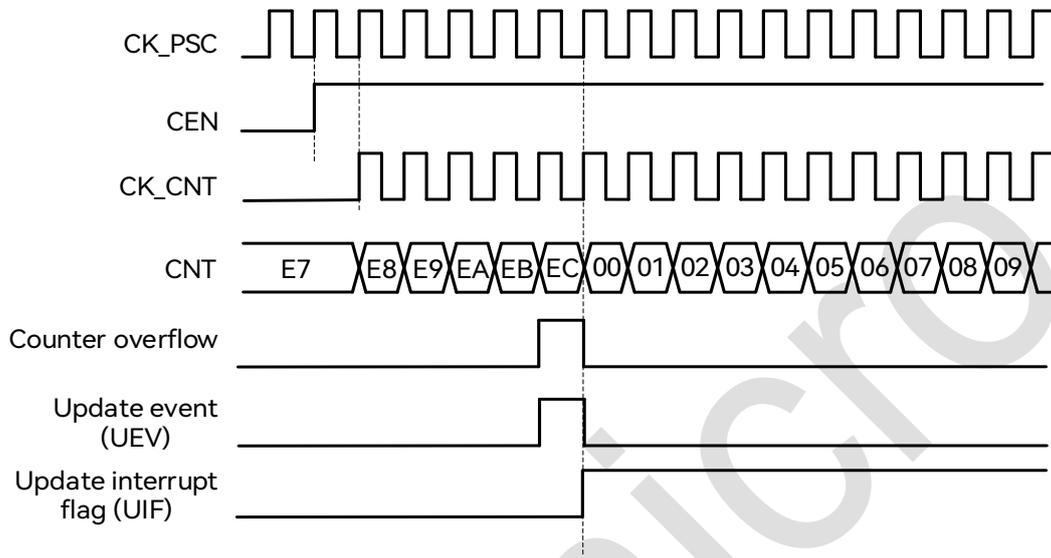


图 12-4: 向上计数波形, 内部时钟不分频

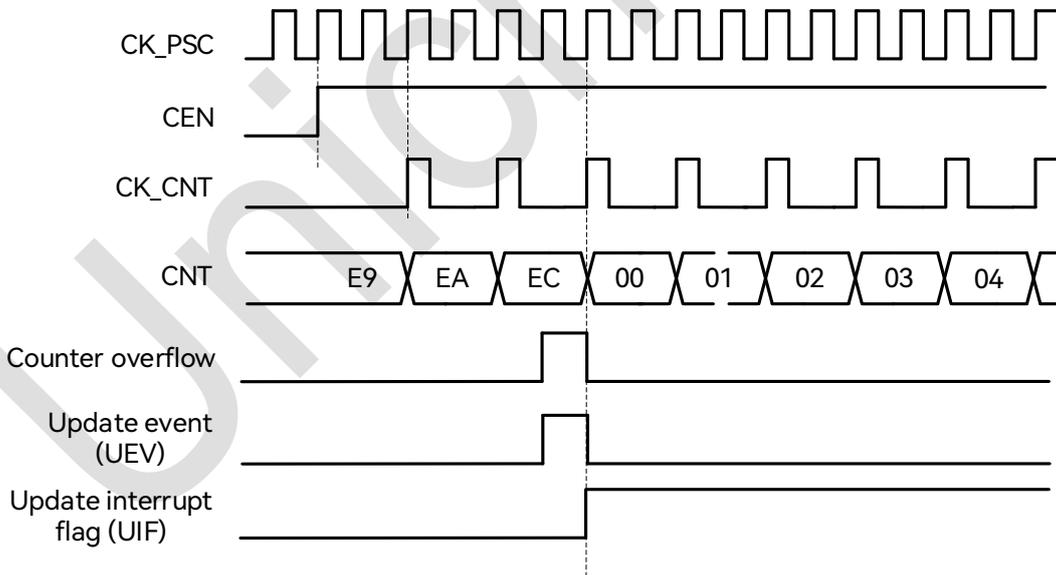


图 12-5: 向上计数波形, 内部时钟 2 分频

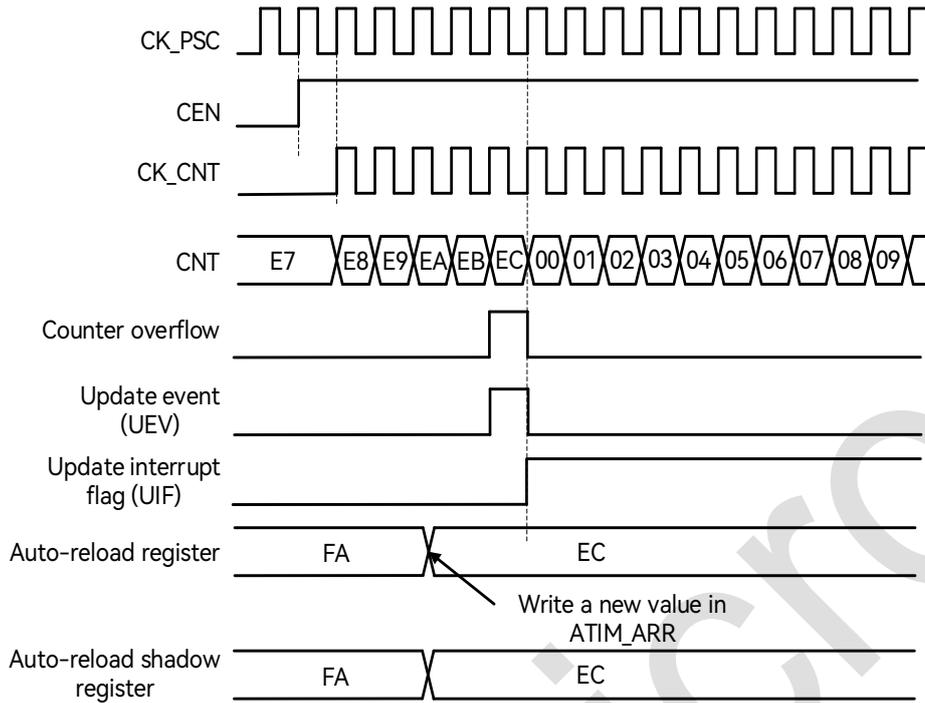


图 12-6: ARPE=0 (ATIMER\_ARR 没有预装载) 时的更新事件

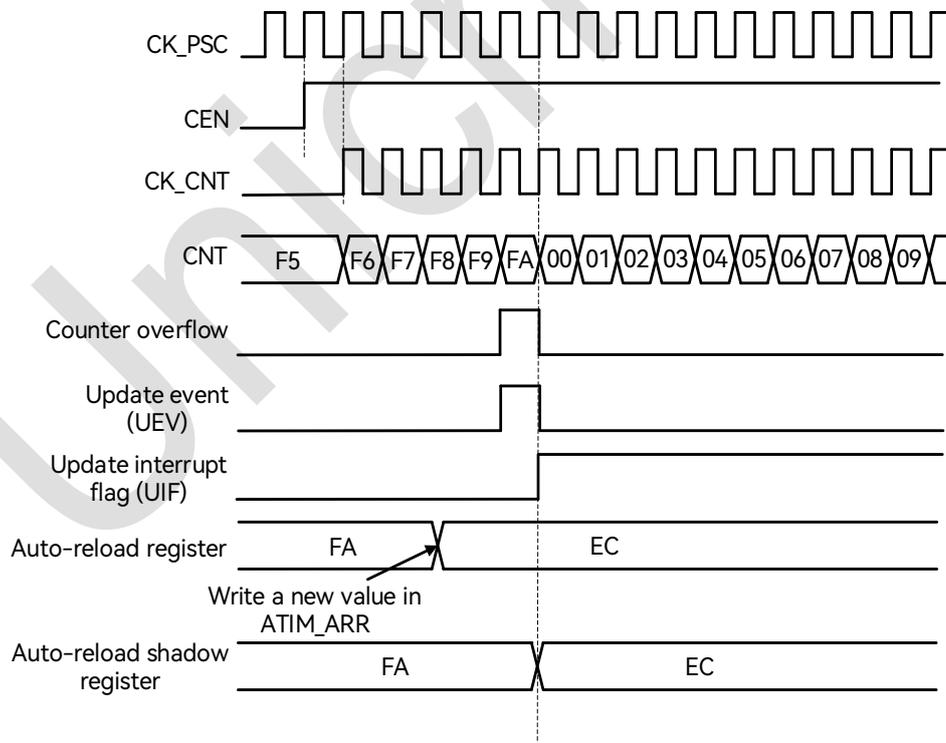


图 12-7: ARPE=1 (ATIMER\_ARR 预装载) 时的更新事件

## 12.4.4 向下计数

向下计数模式中，计数器从 ATIMER\_ARR 值开始递减，到 0 后产生下溢出事件，并且重新从 ATIMER\_ARR 开始计数。

如果使能了重复计数功能，则计数器按照 ATIMER\_RCR 的定义重复上述过程若干次 (ATIMER\_RCR+1)，才会产生溢出事件。

软件可以通过设置 UG 寄存器直接触发 update event，此时 ATIMER\_CNT 和预分频计数器自动清零。设置 UG 寄存器是否触发 UIF (Update Interrupt Flag) 中断标志置位由 URS 寄存器的设置决定。

通过设置 UDIS 寄存器可以禁止 update event，这样可以避免将 preload 寄存器中的值更新到工作寄存器中。

当 update event 发生时，以下寄存器被更新，并且 UIF 置位：

- ATIMER\_RCR 影子寄存器被更新为 ATIMER\_RCR 内容。
- ATIMER\_ARR 影子寄存器被更新为 ATIMER\_ARR 内容。
- ATIMER\_PSC 影子寄存器被更新为 ATIMER\_PSC 内容。

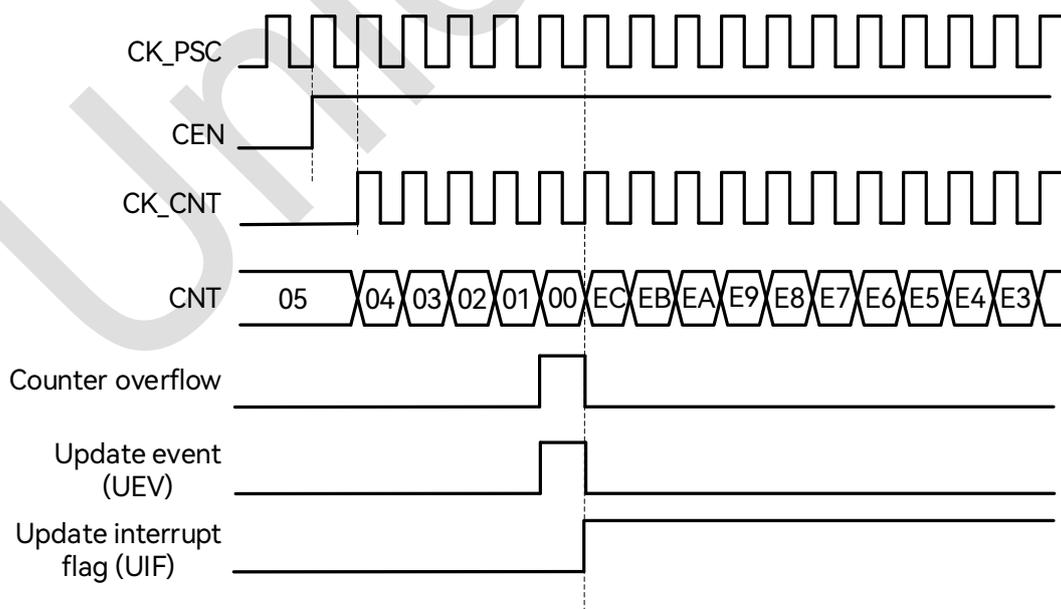


图 12-8: 向下计数，内部时钟不分频

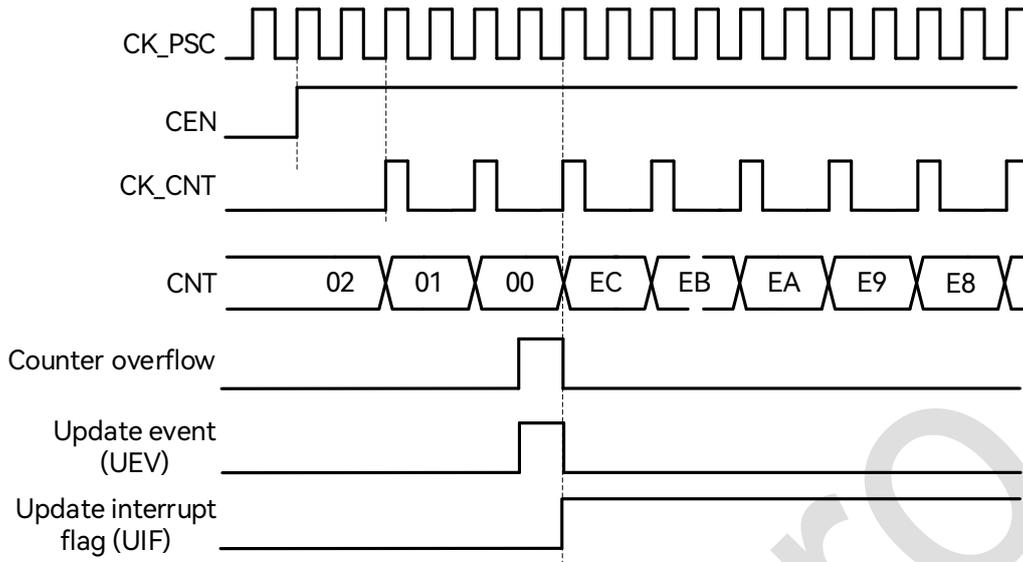


图 12-9: 向下计数, 内部时钟 2 分频

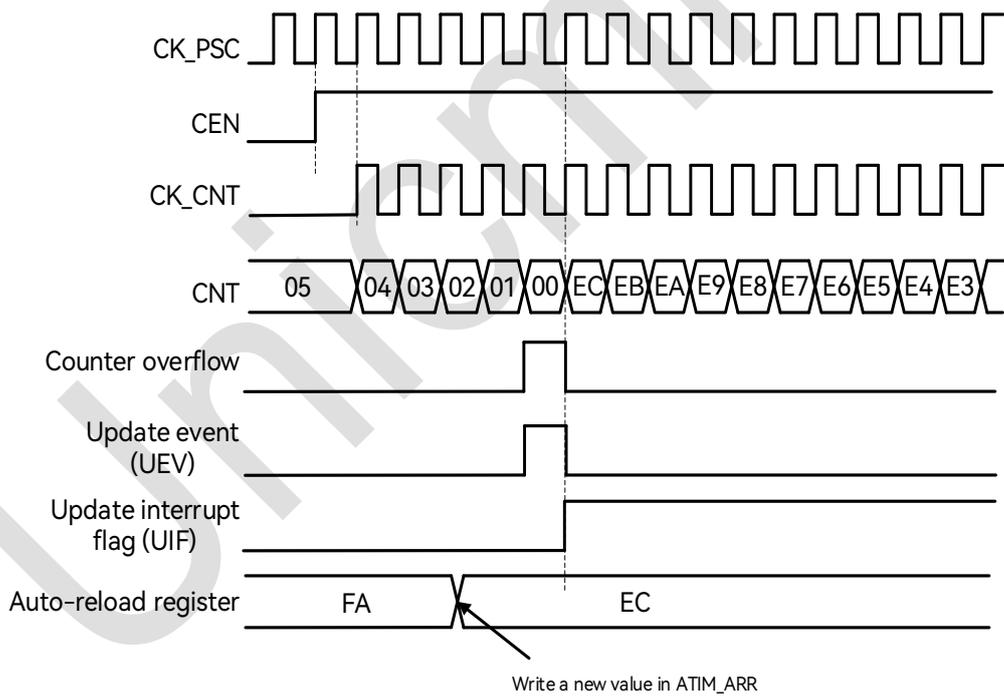


图 12-10: ARPE=0 (ATIMER\_ARR 没有预装载) 时的更新事件

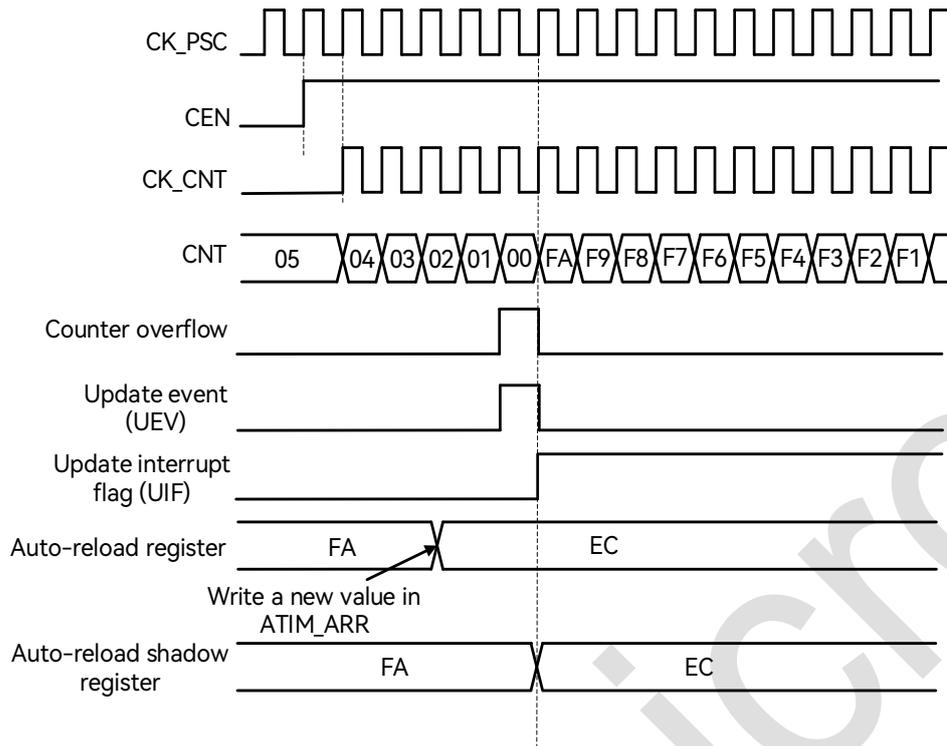


图 12-11: ARPE=1 (ATIMER\_ARR 预装载) 时的更新事件

### 12.4.5 中心对齐计数

在中心对齐模式下，计数器从 0 开始向上计数，到  $ATIMER\_ARR-1$  产生上溢出事件，然后从  $ATIMER\_ARR$  开始向下计数到 1，产生下溢出事件，再从 0 重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当 CMS!=00 时为中心对齐计数，当 CMS=01 时，输出比较功能仅在向下计数时有效，当 CMS=10 时，输出比较功能仅在向上计数时有效，当 CMS=11 时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR 寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在 overflow 和 underflow 的事件上都会更新 ATIMER\_ARR、ATIMER\_PSC 和 ATIMER\_RCR 的影子寄存器。

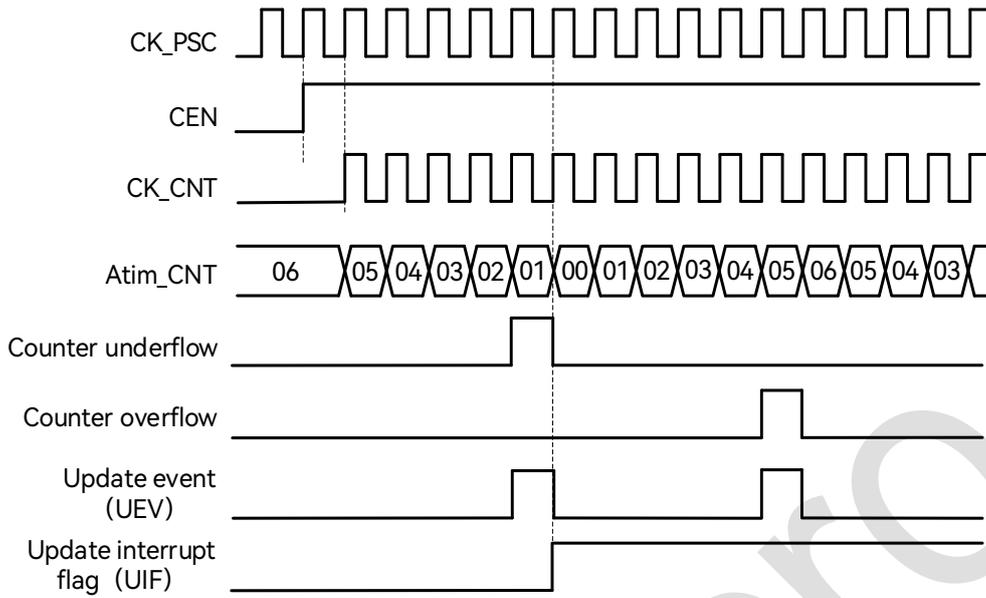


图 12-12: 中心对齐计数器时序图, ATIMER\_PSC=0, ATIMER\_ARR=0x6

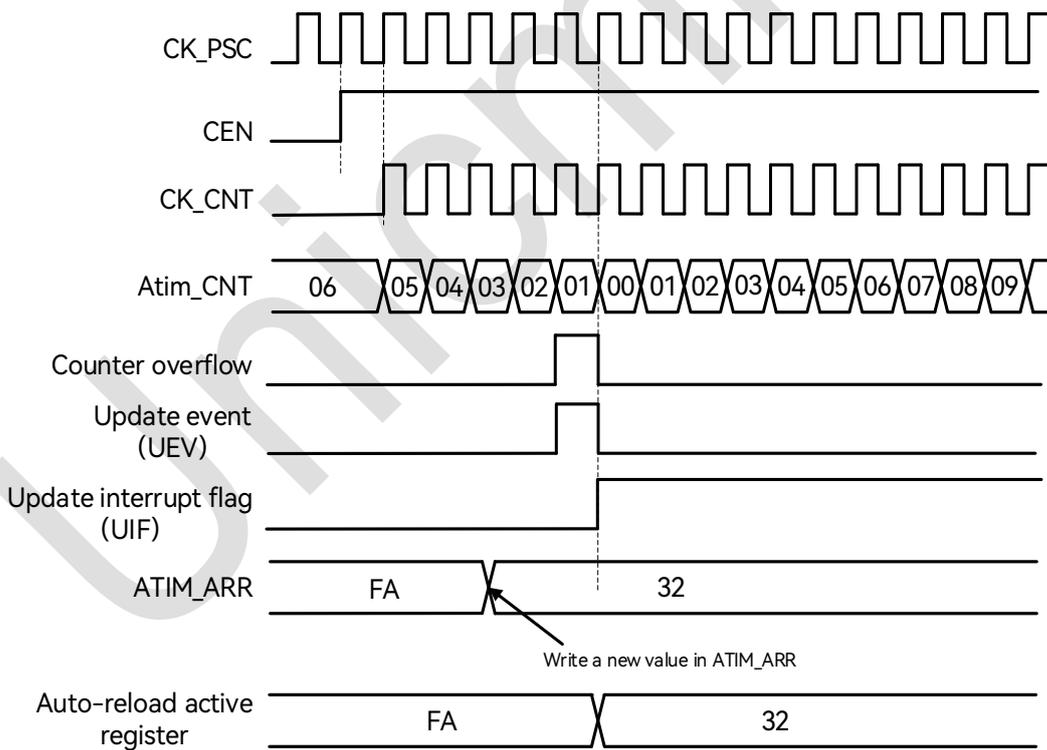


图 12-13: 计数器时序图, ARPE=1 时的更新事件 (计数器下溢)

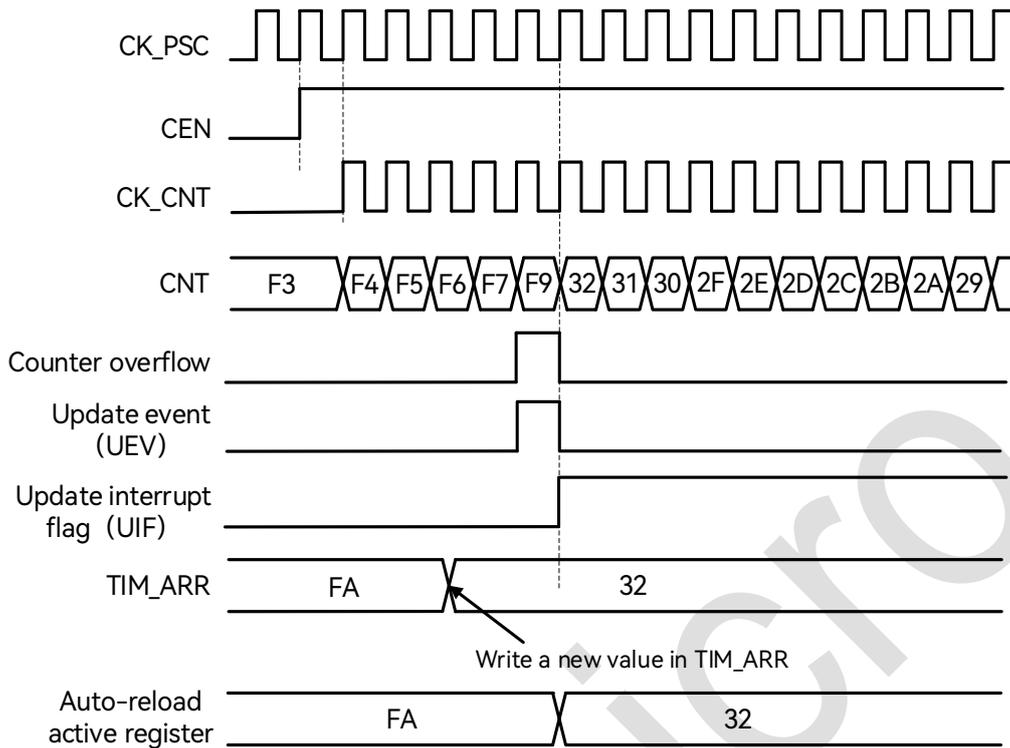


图 12-14: 计数器时序图, ARPE=1 时的更新事件 (计数器溢出)

## 12.4.6 重复计数器

Update event 在计数器 overflow 或 underflow, 并且重复计数器为 0 的情况下产生。这意味着 ATIMER\_ARR、ATIMER\_PSC、ATIMER\_CCR (比较/捕捉寄存器, 输出比较模式下) 的 preload 寄存器会在 N+1 次 overflow 或 underflow 之后, 才将数据传输给影子寄存器, 其中 N 是 ATIMER\_RCR 寄存器值。

重复计数器在以下情况下递减:

- 向上计数模式下发生上溢出
- 向下计数模式下发生下溢出
- 中心计数模式下每次上溢出或者下溢出

注: 当 update event 由软件或 slave mode controller 触发时, 更新事件会立即发生, 而不管当前 ATIMER\_RCR 是什么值, 同时重复计数器也会被立即更新为 ATIMER\_RCR 的值。

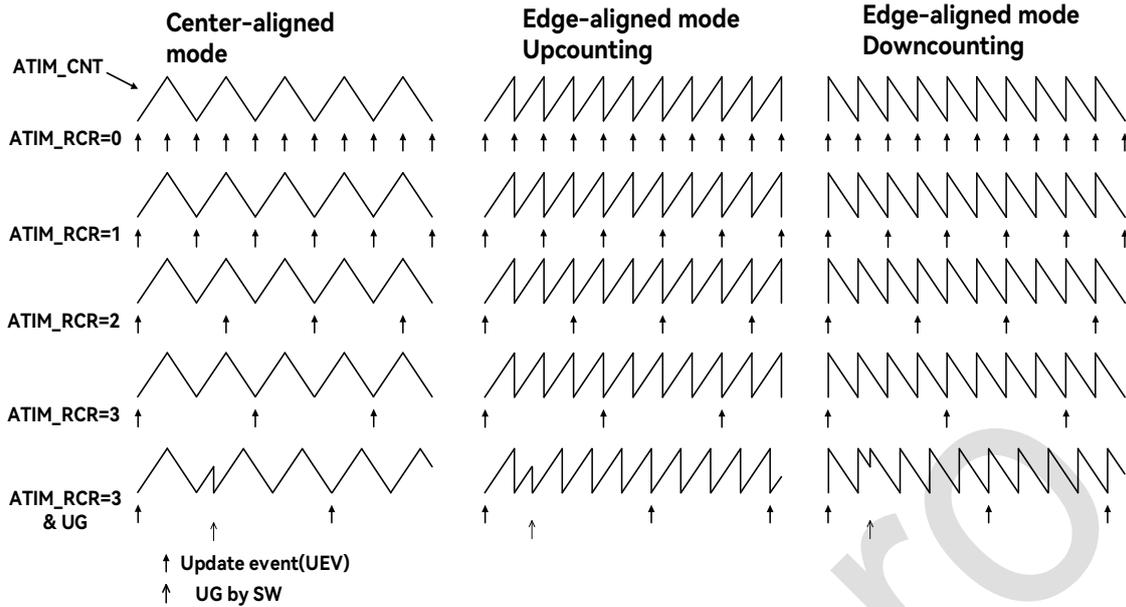


图 12-15: 不同模式下更新速率的例子, 及 ATIMER\_RCR 的寄存器设置

## 12.4.7 Preload 寄存器

以下功能寄存器支持 Preload 功能:

- 自动重载寄存器 ATIMER\_ARR
- 预分频寄存器 ATIMER\_PSC (不可关闭 preload 功能)
- 通道控制寄存器 ATIMER\_CCR
- CCxE 和 CCxNE 控制寄存器
- OCxM 控制寄存器

以上寄存器, 除了 ATIMER\_PSC 之外, 都可以由软件选择使能或者禁止 preload 功能。

具备 Preload 功能的寄存器, 包含两组物理实体:

- Shadow register (影子寄存器): 实际定时器正在使用的寄存器
- Preload register (预装载寄存器): 软件可以访问的寄存器

当禁止 Preload 时, 具备 Preload 功能的寄存器特性如下:

- Preload 寄存器可以实时由软件访问、改写
- Shadow 寄存器与 Preload 寄存器同步更新

如果使能了 Preload，则：

- 所有软件操作访问的是 preload 寄存器
- 当 update event 发生时，所有 Preload 寄存器内容将同步被转移到对应的 shadow 寄存器

## 12.4.8 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK (CK\_INT) ——内部时钟模式
- 外部引脚输入时钟 (Tix) ——外部时钟模式 1
- 外部引脚触发输入 (ETR) ——外部时钟模式 2
- 内部触发 (ITRx) ——使用一个 timer 的触发输出 (TRGO) 作为计数时钟

## 12.4.9 内部时钟模式

内部时钟模式下，禁止从机模式 (SMS=000)，CEN、DIR、UG 等寄存器位都是软件控制。软件操作 UG 寄存器后，update 信号经过 CLK\_PSC 同步后，计数器值将被重新初始化。

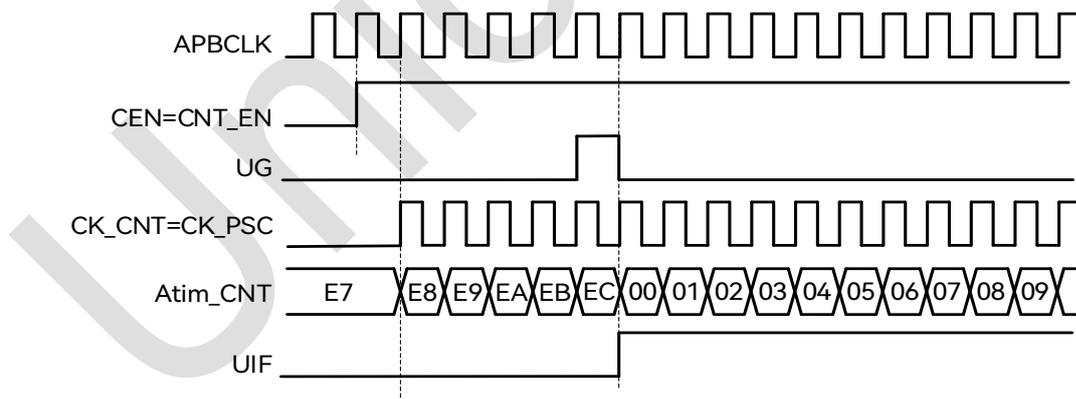


图 12-16: 内部时钟源模式，时钟分频因子为 1

## 12.4.10 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置 SMS=111，计数边沿可以配置为上升或下降沿。

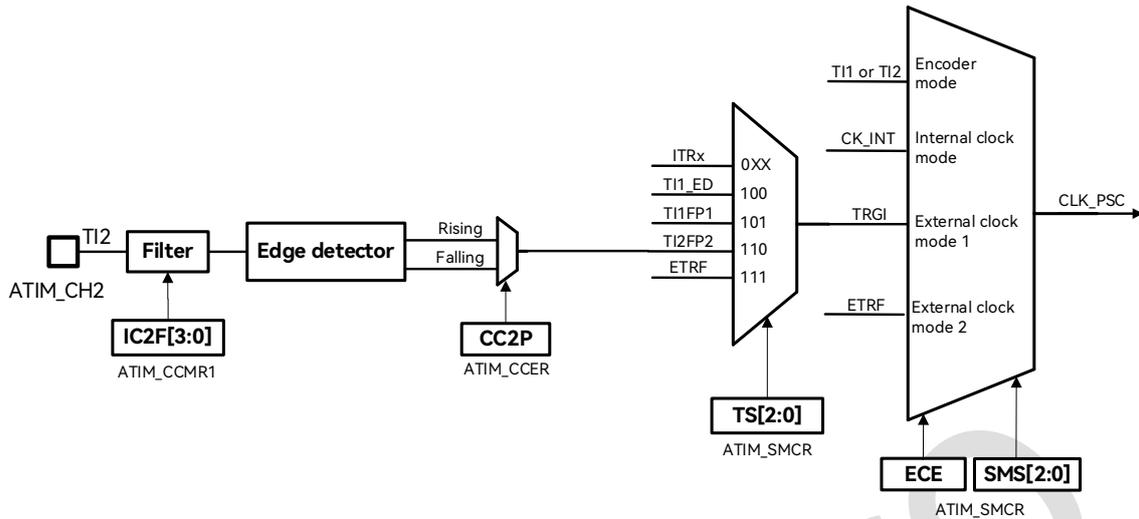


图 12-17: 外部时钟连接例子

外部输入信号在触发计数器计数前，会先经过内部时钟的同步过程，同时输入信号的有效沿会触发 TIF 标志。

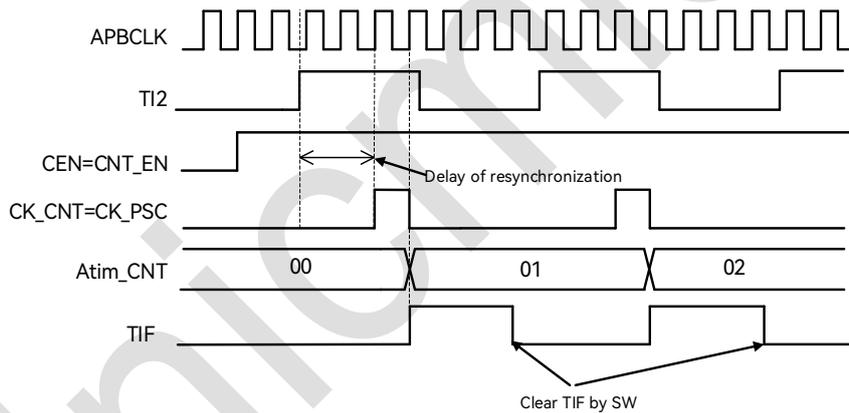


图 12-18: 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能 ATIMER 的内部时钟 (ATIMERx\_clk)，因为 ATIMER 要使用 ATIMERx\_clk 来对外部输入时钟进行同步和滤波。在外部时钟模式 1 下，外部输入时钟首先经过滤波和边沿选择，得到有效的计数沿，作为有效工作时钟 (CLK\_PSC) 输入给预分频模块。

外部时钟同步采用简单的 2 级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于 2 个 APBCLK 周期。

此模式下只有通道 1 和 2 的输入可以用做时钟输入，所需配置如下：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH2 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC2E=0，确保之后通道配置成功。

3. 选择输入通道，配置 `ATIMER_CCMR1.CC2S=01`，IC2 映射到 TI2。
4. 选择计数有效沿，配置 `ATIMER_CCER.CC2P=0`，选择上沿或者下沿。
5. 配置输入滤波时间，配置 `ATIMER_CCMR1.IC2F[3:0]` (`IC2F=0000`，不进行输入滤波)。
6. 使能外部时钟模式 1，配置 `ATIMER_SMCR.SMCR=111`。
7. 选择触发输入源，配置 `ATIMER_SMCR.TS=110`，选定 TI2 作为触发输入源。
8. 打开通道使能，配置 `ATIMER_CCER.CC2E=1`。
9. 使能计数器，配置 `ATIMER_CR1.CEN=1`。

下图是一个典型的外部时钟计数模式 1 的示例：

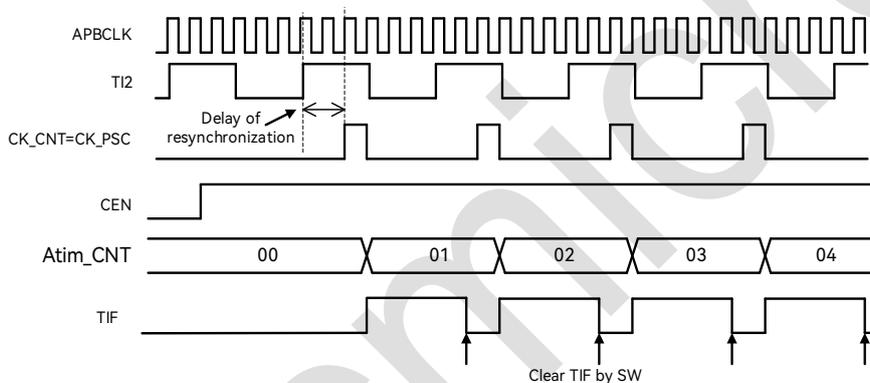


图 12-19: 外部时钟模式 1 下的时序

## 12.4.11 外部时钟模式 2

此模式下使用 `ATIMER_ETR` 管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

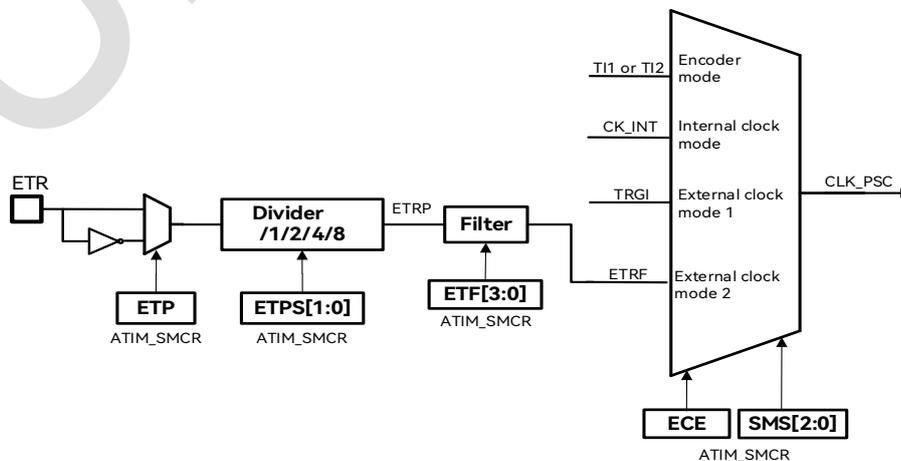


图 12-20: 外部触发输入框图

下图是使用 ETR 二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于 ETR 输入上升沿。

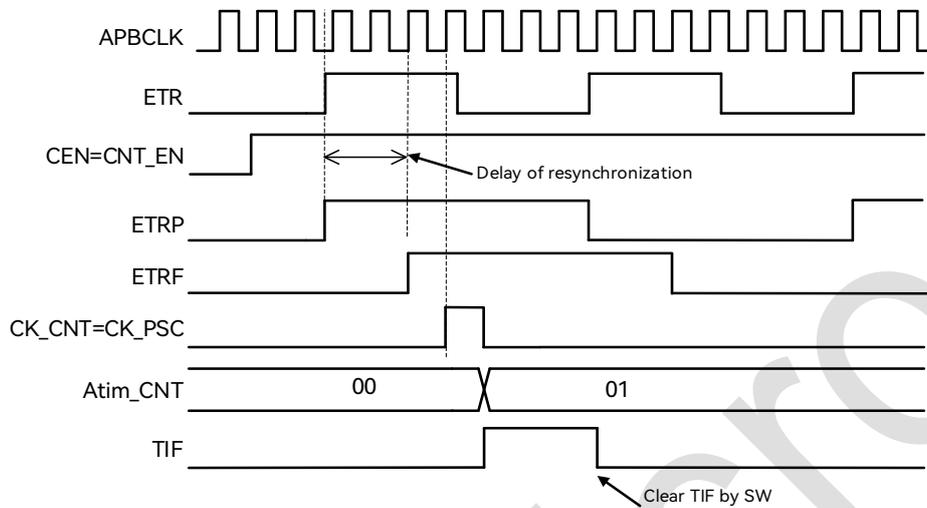


图 12-21: 外部时钟模式 2 下的时序 1

与外部时钟模式 1 的主要差别是，ETR 输入直接被分频后再进行滤波，产生 CK\_PSC 时钟，这意味着可以支持 ETR 输入频率高于 APBCLK 的应用场景，这种情况下，需要首先对 ETR 输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_ETR 功能。
2. 设置 ETP 进行沿选择，`ATIMER_SMCR.ETP=0`。
3. 设置 ETR 分频比，配置 `ATIMER_SMCR.ETPS[1:0]=01`。
4. 配置输入滤波时间，`ATIMER_SMCR.ETF[3:0]=0000`。
5. 置位 ECE 寄存器，使能外部时钟模式 2，`ATIMER_SMCR.ECE=1`，`ATIMER_SMCR.SMS=000`。
6. 使能计数器，配置 `ATIMER_CR1.CEN=1`。

下图是一个典型的外部时钟模式 2 的示例：

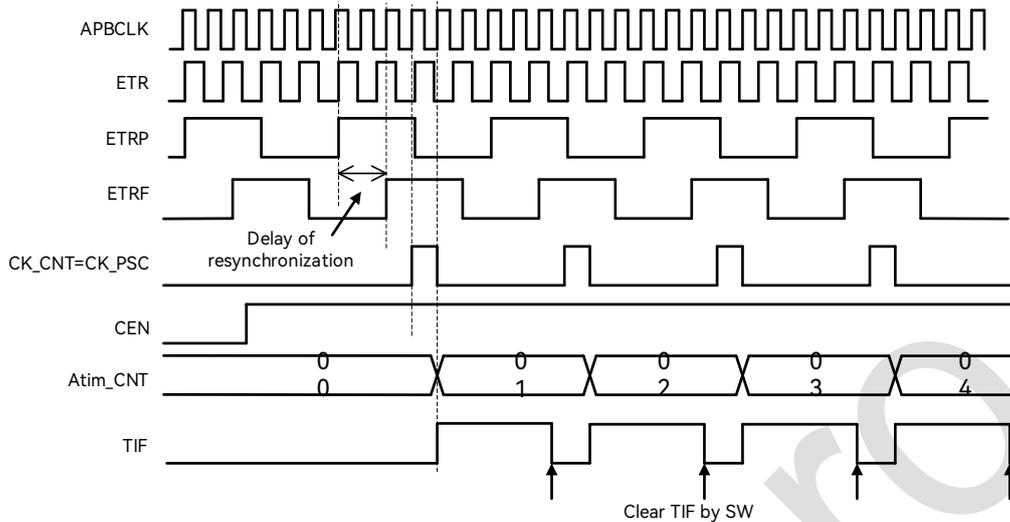


图 12-22：外部时钟模式 2 下的时序

在使用外部时钟模式 2 时，仍可以将 ATIMER 配置为 slave 模式：比如使用 ETR 输入计数，同时使用另一个 Timer 的 TRGO 作为触发信号，当触发事件到来时，复位计数器重新开始计数。

### 12.4.12 内部触发信号 (ITRx)

ATIMER 支持 4 个 ITR 输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需将 TS 配置为 000~011 用于选择 ITR0~ITR3，并将 CCxS 配置为 11，即将 TRC 选为捕捉信号。

ITR 输入支持 4 个内部信号扩展，由 TS 寄存器配置。输入信号源参考下表：

表 12-1：输入信号源

Slave	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
ATIMER	GTIMER1_TRGO	GTIMER2_TRGO	COMP0_IN	COMP1_IN

### 12.4.13 捕捉/比较通道

ATIMER 包含 4 个捕捉通道，5 个比较通道，每个通道由一个捕捉比较寄存器 (CCR) (包含影子寄存器)、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样 Tlx 输入并产生滤波后的信号 TlxF，然后边沿检测和极性选择产生对应的 TlxFPx 信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

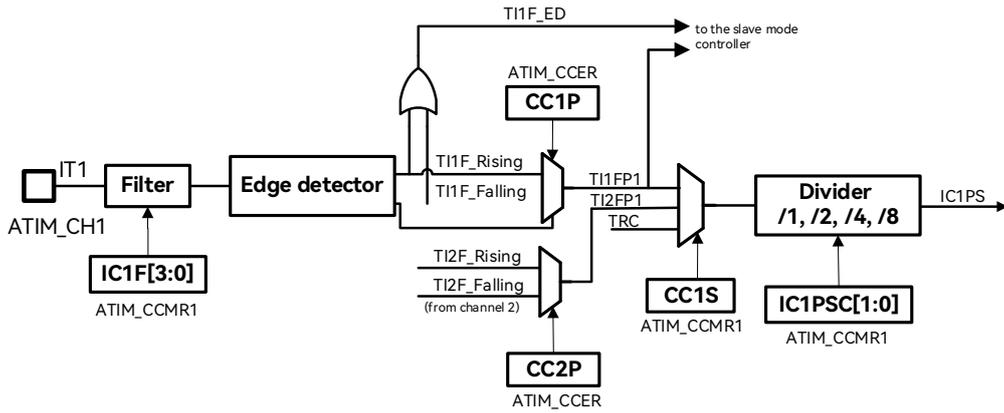


图 12-23: 捕获/比较通道 (通道 1 输入部分)

输出级电路会产生一个输出基准信号 OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。其中通道 1~3 支持互补输出和死区插入，通道 4 则比较简单，不支持互补输出。

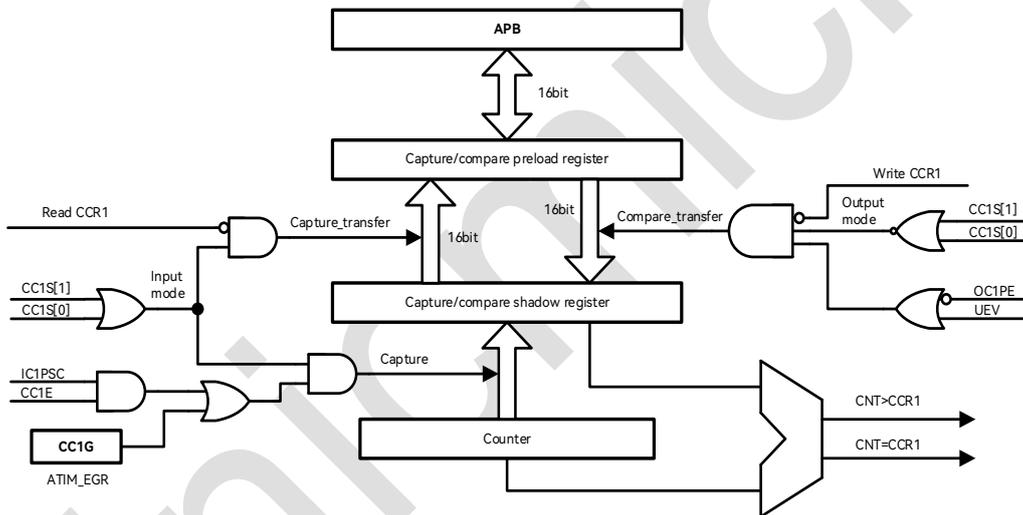


图 12-24: 捕获/比较通道 1 的主电路

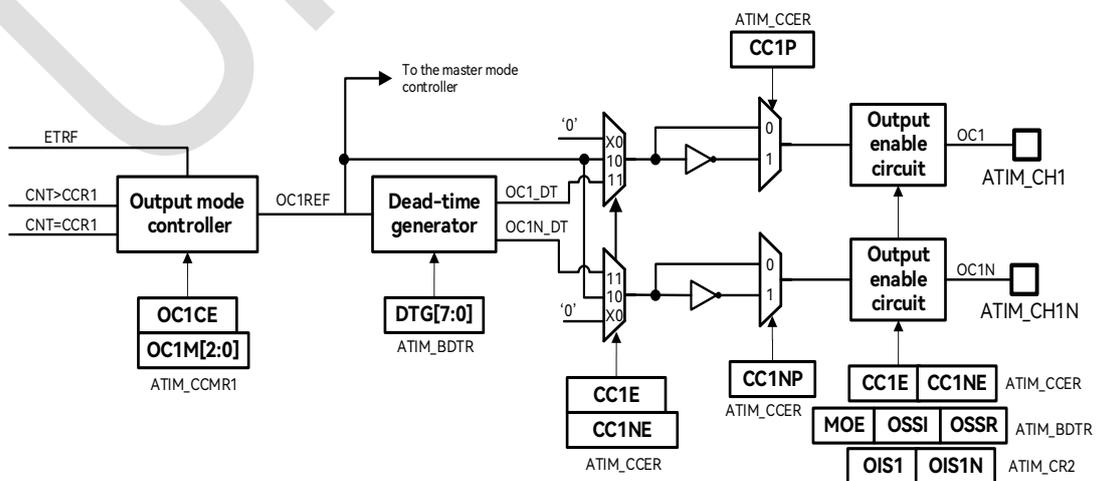


图 12-25: 捕获/比较通道的输出部分 (通道 1 至 3)

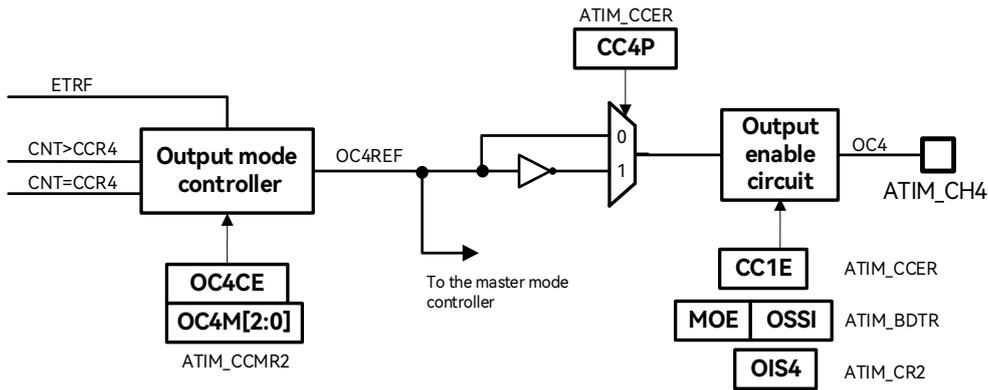


图 12-26: 捕获/比较通道的输出部分 (通道 4)

捕获/比较寄存器 (CCR) 包含 Preload 寄存器和 shadow 寄存器, 软件读写总是访问 Preload 寄存器。在捕获模式下, 捕获值保存在 shadow 寄存器中并复制到 Preload 寄存器。在比较模式下, Preload 寄存器的值被拷贝到 shadow 寄存器用来与计数器比较。

#### 12.4.14 输入捕捉模式

当 ICx 信号上出现预期的电平变换, 将触发一次 capture, 当前计数器值被锁存进 ATIMER\_CCR, 与此同时, CCxIF 中断标志置位, 并且可以触发对应的中断或者 DMA 请求。如果一个捕捉事件在 CCxIF 为高的情况下出现, 则捕捉数据冲突标志 (CCxOF, Over-Capture) 置位 (CCR 中上次捕捉值被覆盖)。CCxIF 可以由软件清零, 或者通过读取 ATIMER\_CCR 寄存器自动清零。CCxOF 标志通过软件写 1 清零。

通过两个或更多通道配合, 可以实现 PWM 信号的输入捕捉。比如要计算一个输入信号的周期和占空比, 可以将此信号从 TI1 引脚输入, 芯片内部将滤波后的信号取上升沿得到 TI1FP1, 将滤波后的信号取下降沿得到 TI1FP2, 将 TI1FP1 输入给捕捉通道 1, 将 TI1FP2 输入给捕捉通道 2, 即可实现通道 1 对输入信号上升沿捕捉, 同时通道 2 对输入信号下降沿捕捉; 捕捉中断定期发生后, 软件通过 ATIMER\_CCR1 和 ATIMER\_CCR2 寄存器的值, 即可计算输入信号的周期和占空比。

实现在 TI1 输入的上升沿捕获计数器的值到 ATIMER\_CCR1 寄存器, 配置步骤如下:

1. 在 GPIO 模块中, 配置相应管脚为 ATIMER\_CH1 功能。
2. 关闭通道使能, 配置 ATIMER\_CCER.CC1E=0, 确保之后通道配置成功。
3. 选择输入通道, 配置 ATIMER\_CCMR1.CC1S=01, IC1 映射到 TI1。

4. 选择计数有效沿，配置 ATIMER\_CCER.CC1P，选择上沿或者下沿。
5. 配置输入滤波时间，配置 ATIMER\_CCMR1.IC1F[3:0]。
6. 配置输入预分频器，配置 ATIMER\_CCMR1.IC1PS[1:0]。
7. 打开通道使能，配置 ATIMER\_CCER.CC1E=1。

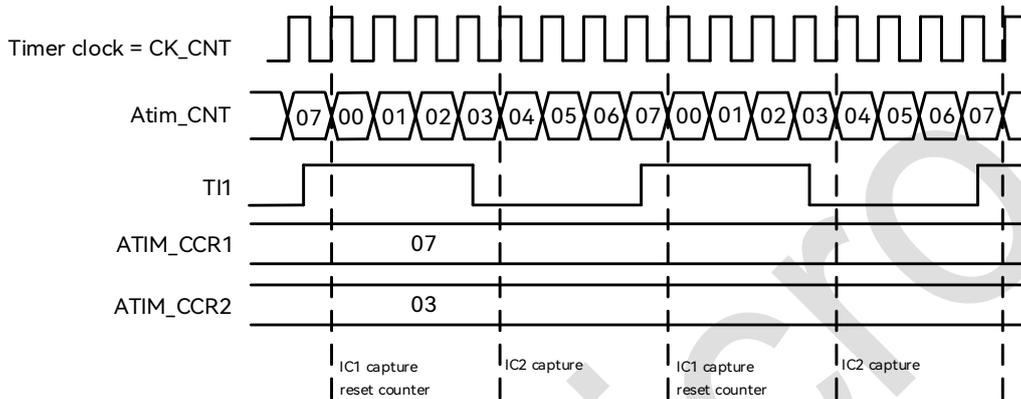


图 12-27: PWM 输入捕获模式时序

若想实现 PWM 输入捕获功能，需进行如下设置：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0, ATIMER\_CCER.CC2E=0 确保之后通道配置成功。
3. 选择输入通道，两个通道 IC1、IC2 被映射到同一个 TI1 输入口，配置 ATIMER\_CCMR1.CC1S=01, ATIMER\_CCMR1.CC2S=10。
4. 选择计数有效沿，两个通道 IC1、IC2 有效沿极性相反，配置 ATIMER\_CCER.CC1P=0, ATIMER\_CCER.CC2P=1。
5. 配置输入滤波时间，配置 ATIMER\_CCMR1.IC1F[3:0], ATIMER\_CCMR1.IC2F[3:0]。
6. 配置输入预分频器，配置 ATIMER\_CCMR1.IC1PS[1:0] , ATIMER\_CCMR1.IC2PS[1:0]。
7. 选择触发输入信号，配置 ATIMER\_SMCR.TS[2:0]=101。
8. 设定从模式控制器为复位模式，配置 ATIMER\_SMCR.SMS[2:0]=100。
9. 打开通道使能，配置 ATIMER\_CCER.CC1E=1, ATIMER\_CCER.CC2E=1。

## 12.4.15 软件 Force 输出

在比较输出模式下，软件可以直接将 OCxREF force 成特定电平，而独立于 ATIMER\_CCR 和 ATIMER\_CNT 的比较结果。

软件通过写 OCxM=101 寄存器，可以直接将 OCxREF 强制为有效（OCxREF 固定为高有效），通过写 OCxM=100 可以直接将 OCxREF 强制为无效（低电平）。但是软件 force 操作不会取消比较过程，ATIMER\_CCR 和 ATIMER\_CNT 的比较还会一直进行。

## 12.4.16 输出比较模式

输出比较模式下，当 ATIMER\_CCR 与 ATIMER\_CNT 相等，OCxREF 可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA 请求可以发送（改写配置寄存器）。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。使用步骤如下：

1. 选择计数时钟（内部、外部、预分频等）。
2. 向 ATIMER\_ARR 和 ATIMER\_CCR 寄存器写入期望数据。
3. 根据需要设置中断使能和 DMA 使能。
4. 选择输出模式。
5. 使能计数器。

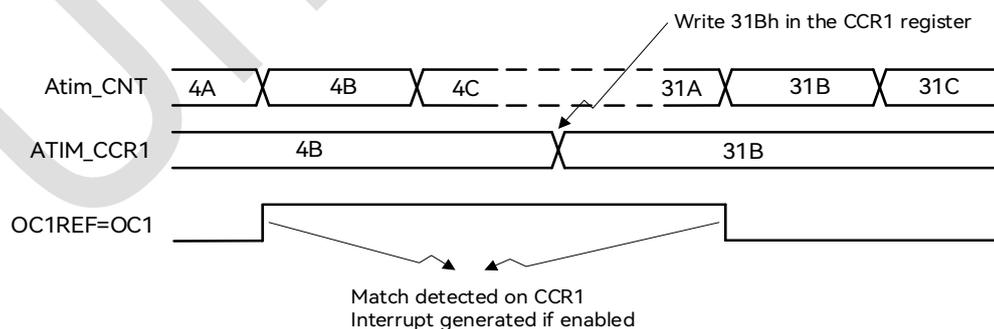


图 12-28: 输出比较模式，翻转 OC1

在不使能 Preload 的情况下，软件可以随时改写 ATIMER\_CCR 寄存器实现对输出波形的实时控制。如果使能了 Preload，则 ATIMER\_CCR 的 shadow 寄存器仅在下一次 update event 发生时更新为 Preload 寄存器的内容。

## 12.4.17 PWM 输出

PWM 模式可以输出脉宽调制信号,其周期由 ATIMER\_ARR 寄存器决定, 占空比由 ATIMER\_CCR 寄存器决定。

输出信号的极性可以由 CCxP 寄存器配置。PWM 模式工作中, ATIMER\_CNT 和 ATIMER\_CCR 实时比较。由于计数器支持边缘对齐和中央对齐计数模式, PWM 输出也支持边缘对齐和中央对齐模式。

- PWM 边缘对齐模式

在向上计数的情况下, 配置为 PWM 模式 1 时, OCxREF 信号在  $ATIMER\_CNT < ATIMER\_CCR$  时为高电平, 否则为低电平。如果 ATIMER\_CCR 值大于 ATIMER\_ARR 值, 则 OCxREF 被固定为 1; 如果 CCR 为 0, 则 OCxREF 被固定为 0。

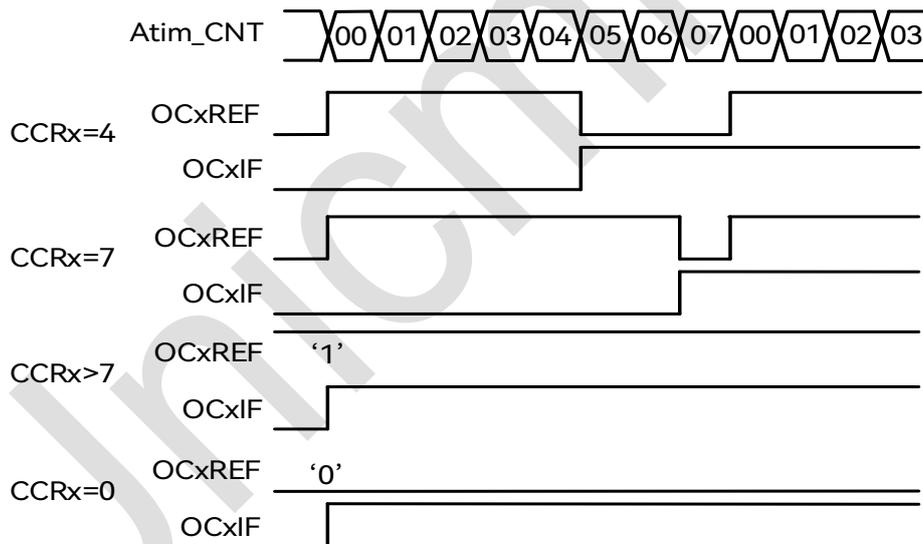


图 12-29: 边沿对齐的 PWM 波形 (ARR=7)

在向下计数时, OCxREF 电平高低定义与向上计数时相同。

- PWM 中央对齐模式

OCxREF 电平定义与边缘对齐模式相同。下图是一个示例：

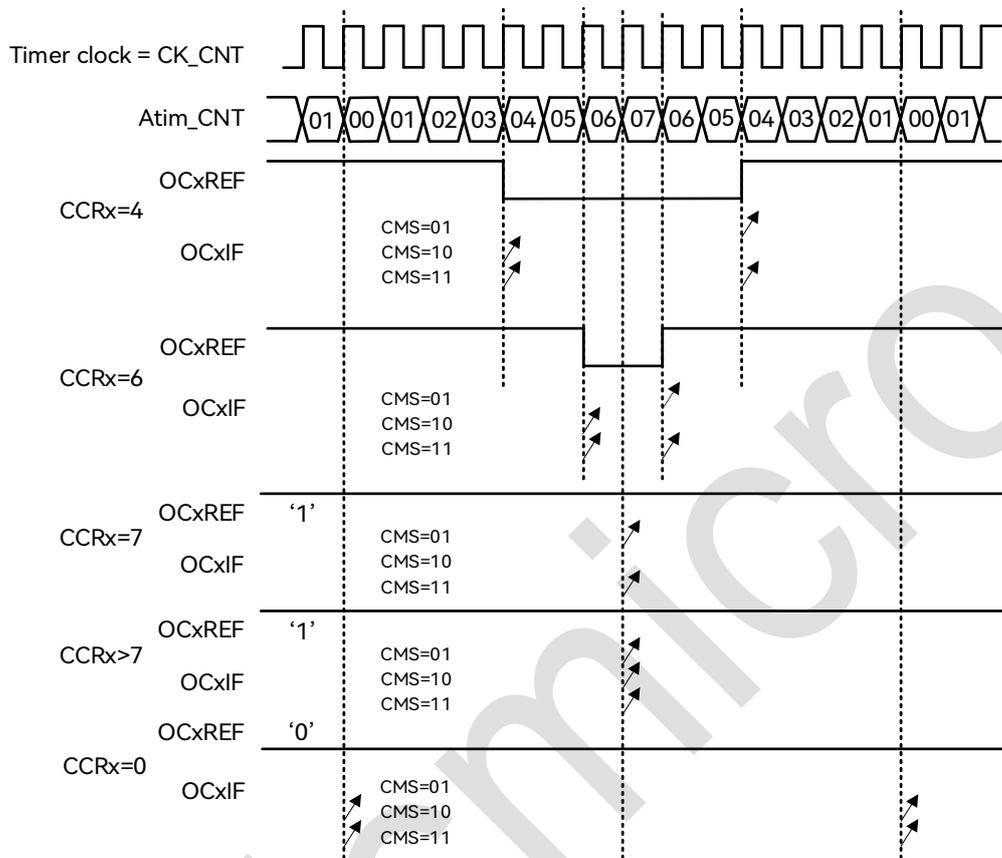


图 12-30: 中央对齐的 PWM 波形 (ARR=7)

当启动中央对齐计数时，一开始的计数方向是由 DIR 寄存器决定的；随后在计数过程中，DIR 寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过 UG 寄存器做一次 update，并且在计数过程中不要改写计数器。

### 12.4.18 非对称 PWM 移相控制

控制 PWM 信号的方法主要包括调整周期、占空比、相位等。而调整策略则需根据具体的电路和应用场景来定制。例如，在电源转换应用中，一般通过调整占空比来控制输出电压或电流。在电机控制中，除了占空比外，还可以通过控制 PWM 的移相来实现输出多种波形，从而优化电机性能。

计数器在中央对齐模式下，Atimer 可生成相移可编程的 PWM 信号。通道 1~3 支持 PWM 的移相功能。ATIMER\_PMEN[2:0] 寄存器用于使能不同通道的 PWM 移相输出，PWM 的周期由

ATIMER\_ARR 寄存器决定，占空比由 ATIMER\_CCR 和 ATIMER\_PMC 决定（ATIMER\_CNT 向上计数时和 ATIMER\_CCR 比较，向下计数时和 ATIMER\_PMC 比较），由此来实现对 PWM 输出的实时控制。

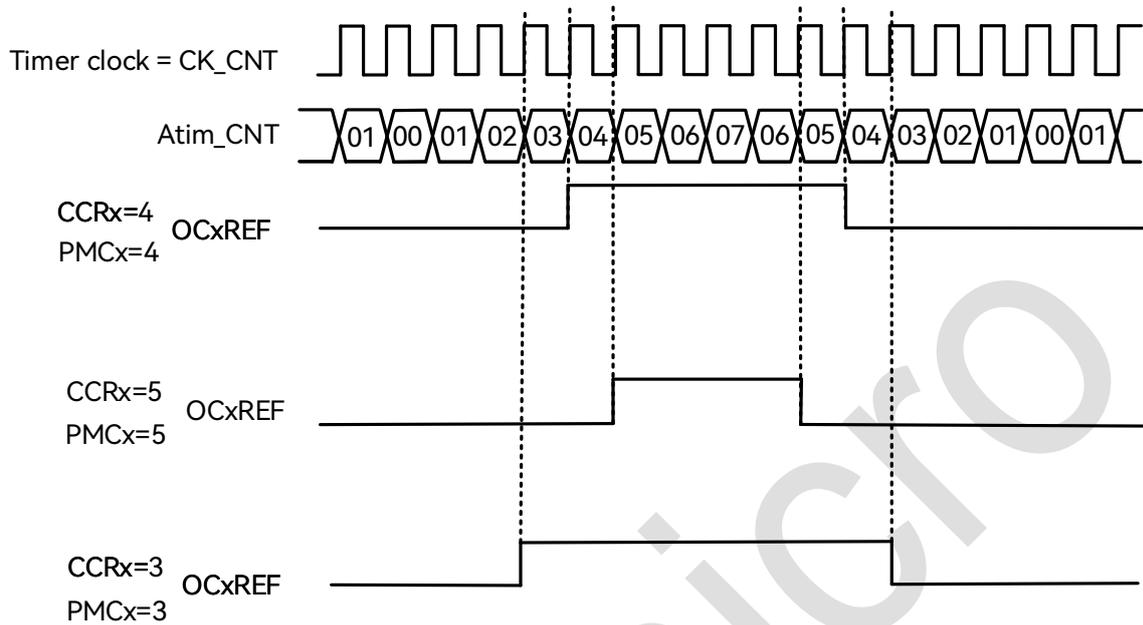


图 12-31: 移相模式的 PWM 波形 (ARR=7)

### 12.4.19 互补输出和死区插入

ATIMER 的通道 1~3 支持互补输出和死区插入。DTG[7:0]寄存器用于设置死区时间（对所有通道同时有效）。输出信号 OCx 与参考信号 OCxREF 同相，OCxN 与参考信号反相；OCx 的上升沿是 OCxREF 上升沿的 delay，OCxN 的上升沿是 OCxREF 下降沿的 delay。

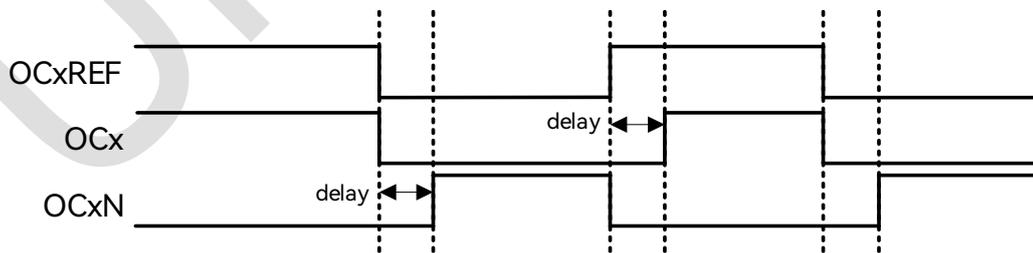


图 12-32: 带死区插入的互补输出

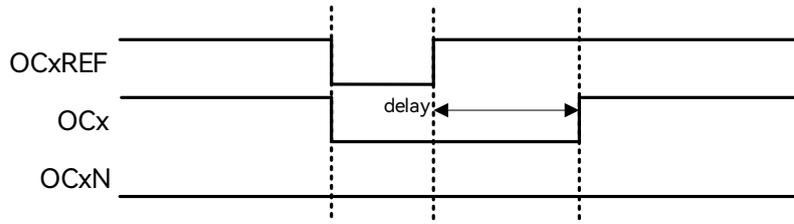


图 12-33: 死区波形延迟大于负脉冲

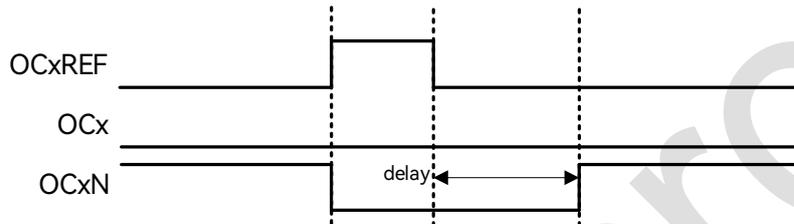


图 12-34: 死区波形延迟大于正脉冲

## 12.4.20 刹车功能

刹车功能可以使用外部 BRK 引脚输入的 7 路刹车信号，或者比较器、SVD、XTHF 停振检测产生的有效输出；上电复位后刹车电路被禁止，用户通过置位 BKE 寄存器使能刹车功能；7 路刹车输入可以配置为相与或者相或操作。组合后的刹车信号可以配置有效极性以及数字滤波。

ATIMER\_BRKx 复用 GPIO 功能，当 GPIO 设置为数字外设功能时，其输入信号直接连接到 ATIMER 的刹车输入上；当 GPIO 设置为其他功能时，ATIMER 的刹车输入端口被固定成 1。通过 BRKxGATE 寄存器，可以控制门控后的 BRKx 信号的实际电平，软件能够灵活的将不使用的 BRKx 设置为 0 或者 1 电平，以适应后级逻辑电路的需要。

当一个刹车事件发生时：

- 输出使能寄存器被异步清零，可通过 OSSI 寄存器选择输出被强制为 inactive/idle/reset 状态。
- 每个输出通道被驱动为 OISx 寄存器定义的电平。
- 当互补输出使能时，输出被异步置位成 inactive 和 reset 状态，死区插入电路开始工作，在死区时间后驱动输出为 OISx 和 OISxN 定义的电平。
- 刹车标志寄存器置位，根据配置可以触发中断。

- 如果使能了自动输出 (AOE=1)，输出使能位 (MOE) 将在下一个 update event 发生时被自动置位；否则 MOE 将保持为 0 直到被软件重新置位。

注：BRK 信号是电平有效的，因此在 BRK 保持有效的情况下，无法使能 MOE，同时刹车标志 BIF 也无法清除。

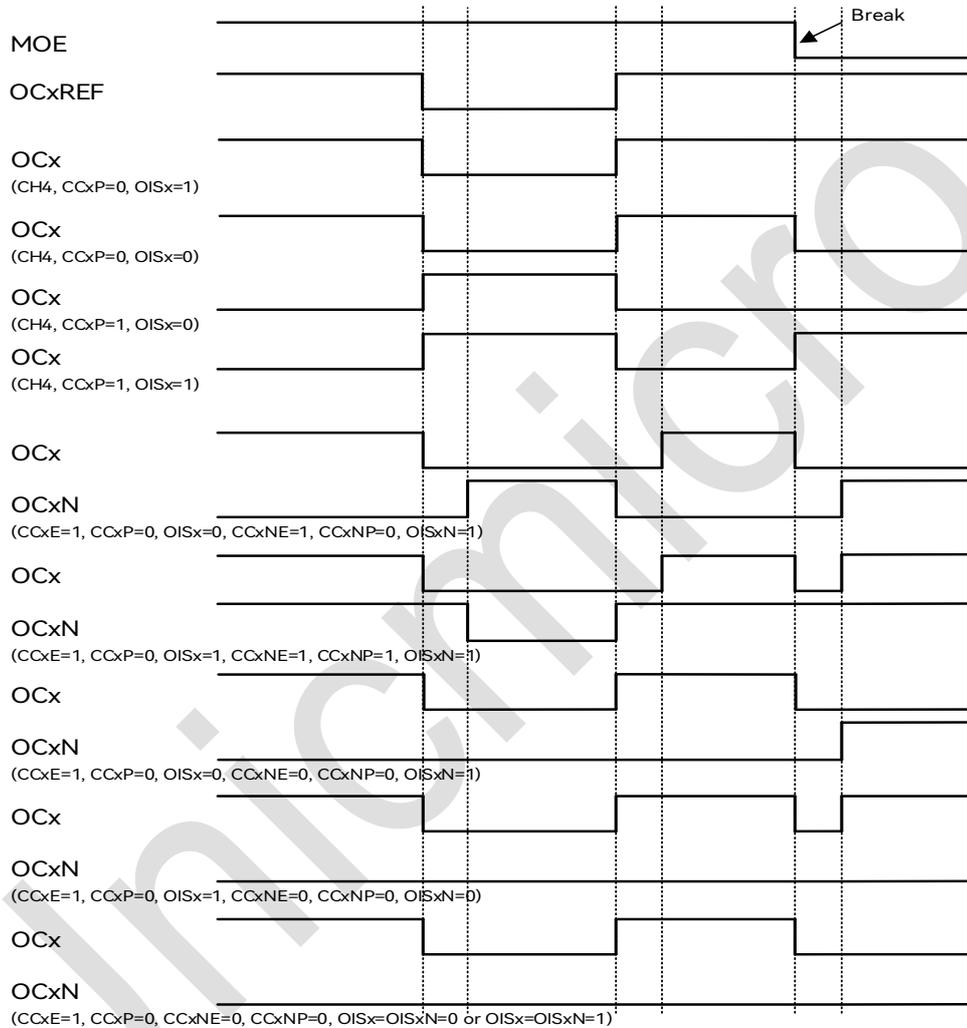


图 12-35: 响应刹车的输出

### 12.4.21 6-step PWM 输出

当某个通道使用互补输出时，OCxM, CCxE, CCxNE 寄存器支持 Preload 功能，Preload 寄存器的值在换相 (COM) 事件发生时被装载到 shadow 寄存器中。用户因此可以预先设置下一步配置，并在 COM 事件发生时同步更新所有通道。COM 事件可以由软件写 ATIMER\_EGR 中的 COM 位触发，或者由 TRGI 上升沿硬件触发。

当 COM 事件发生时，换相标志寄存器置位，并且可以产生中断或 DMA 请求。

下图是一个 6 步换相控制的例子，当 COM 事件发生时，三个例子显示不同配置下的输出变化。

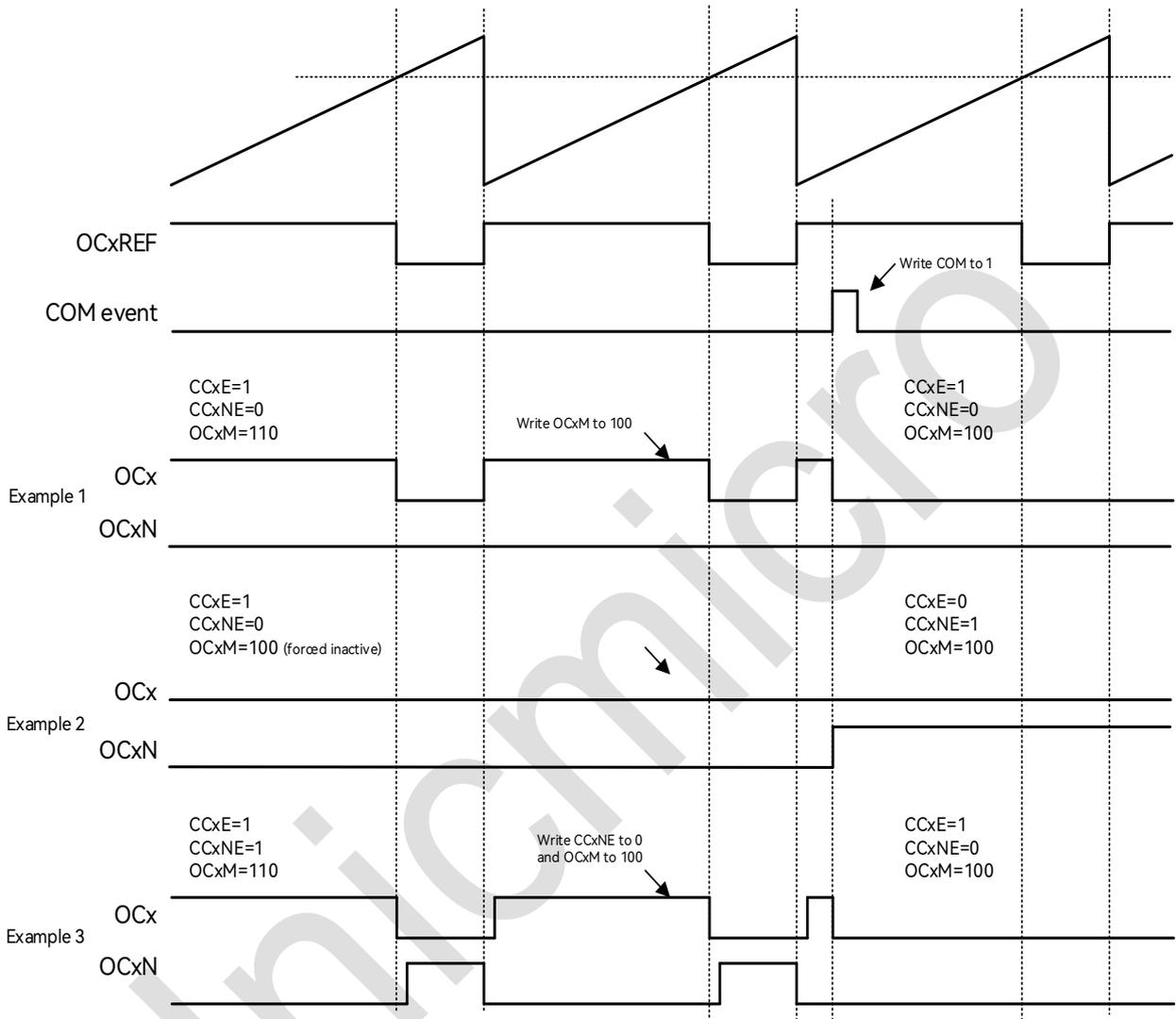


图 12-36: 产生六步 PWM，使用 COM 的例子 (OSSR=1)

### 12.4.22 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下次 update event 到来时，计数器会自动停止。只有当 ATIMER\_CCR 和 ATIMER\_CNT 初值不同时，脉冲才有可能正确输出。在向上计数时，要求  $ATIMER\_CNT < ATIMER\_CCR \leq ATIMER\_ARR$ ；在向下计数时，要求  $ATIMER\_CNT > ATIMER\_CCR$ 。

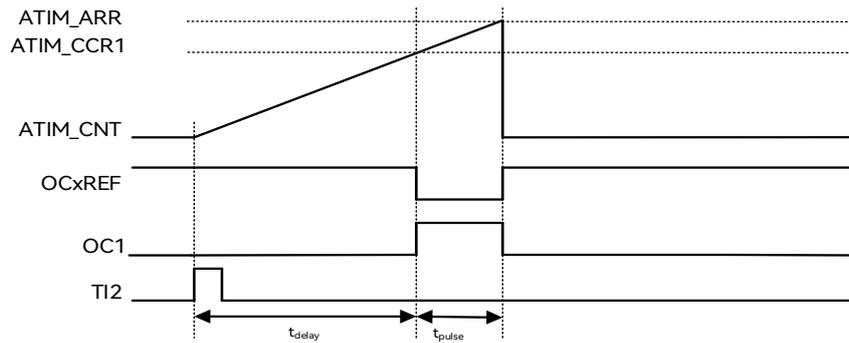


图 12-37: 单脉冲模式的例子

上图是以 TI2 输入为计数器触发信号，计数值等于 ATIMER\_CCR 后 OCxREF 输出低电平，计数到 ATIMER\_ARR 后 OCxREF 回到高电平，并且计数器回滚到 0，停止计数。

#### 实现上述功能 TI2 作为输入触发的配置如下:

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH2 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC2E=0，确保之后通道配置成功。
3. 选择输入通道，配置 ATIMER\_CCMR1.CC2S=01。
4. 选择计数有效沿，配置 ATIMER\_CCER.CC2P=0。
5. 选择触发输入信号，配置 ATIMER\_SMCR.TS[2:0]=110，TI2FP2 作为 TRGI。
6. 设定从模式控制器为触发模式，配置 ATIMER\_SMCR.SMS[2:0]=110，I2FP2 用来启动计数器。
7. 打开通道使能，配置 ATIMER\_CCER.CC2E=1。

#### 实现上述功能 OC1 作为输出的配置如下:

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0，确保之后通道配置成功。
3. 输出通道，配置 ATIMER\_CCMR1.CC1S=00。
4. 选择计数有效沿，配置 ATIMER\_CCMR1.OC1M=111，PWM 模式 2。
5. 打开通道使能，配置 ATIMER\_CCER.CC1E=1。

#### OPM 波形产生时基的特殊设置:

1. ATIMER\_CCR1 的值决定了  $t_{\text{delay}}$ 。
2. ATIMER\_ARR 和 ATIMER\_CCR1 的差值决定了  $t_{\text{pulse}}$  (ATIMER\_ARR-ATIMER\_CCR1)。
3. 设置为单脉冲模式，配置 ATIMER\_CR1.OPM=1。

### 12.4.23 外部事件清除 OCxREF

OCxREF 的有效状态为高电平，通过对外部 ETR 引脚施加高电平，可以直接拉低 OCxREF，直到下一次 update event。此功能仅在输出比较和 PWM 模式下有效，无法在软件 force 模式下起作用。使能此功能需要将 OCxCE 置 1。

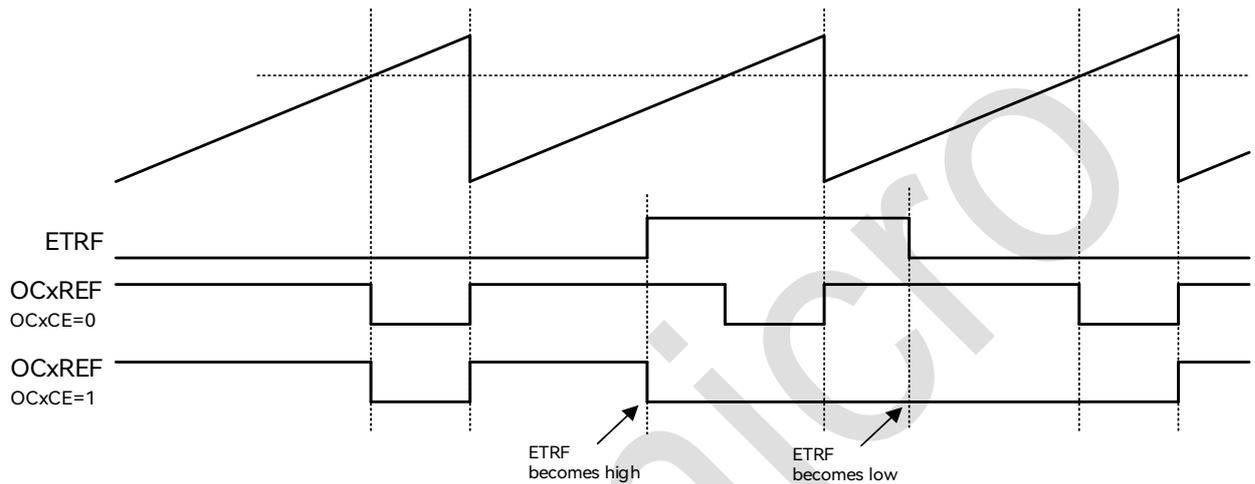


图 12-38: ETR 信号清除 ATIMER 的 OCxREF

### 12.4.24 编码器接口模式

编码器接口模式涉及到两个外部输入信号，ATIMER 根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

表 12-2: Encoder Interface 计数方式

有效沿	对应信号的电平 (T11 对应 TI2, TI2 对应 TI1)	TI1 信号		TI2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

比如在计数器以 TI1 信号为时钟计数时，如果 TI1 上升沿采样到 TI2 为高电平，则计数器递减；如果 TI1 下降沿采样到 TI2 为高电平，则计数器递增。

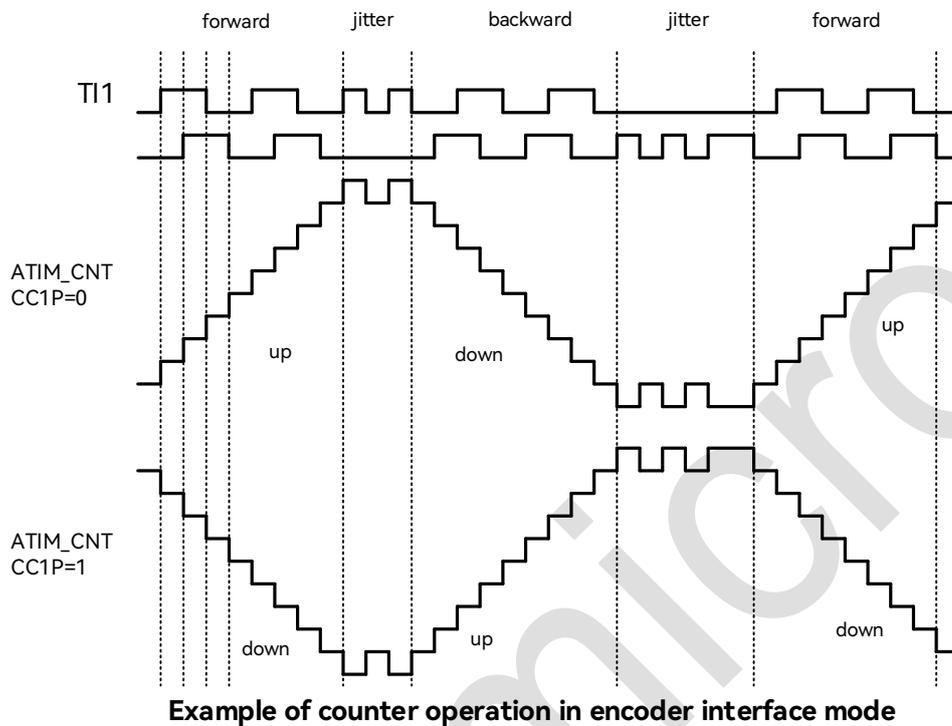


图 12-39: 编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1，ATIMER\_CH2 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0，ATIMER\_CCER.CC2E=0，确保之后通道配置成功。
3. 选择输入通道，配置 ATIMER\_CCMR1.CC1S=01，ATIMER\_CCMR1.CC2S=01。
4. 选择计数有效沿，配置 ATIMER\_CCER.CC1P=0，ATIMER\_CCER.CC2P=0。
5. 设定从模式控制器为编码模式 3，配置 ATIMER\_SMCR.SMS[2:0]=011。
6. 打开通道使能，配置 ATIMER\_CCER.CC1E=1，ATIMER\_CCER.CC2E=1。

### 12.4.25 ATIMER 从机模式

ATIMER 作为 slave 时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

## 12.4.26 复位模式

此模式下，外部输入的事件将导致 ATIMER 内部所有 Preload 寄存器重新初始化，CNT 回到 0 开始计数。以下图为例，计数器正常计数，外部 T11 输入上升沿时，触发计数器清零，重新开始计数。配置步骤如下：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 ATIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 ATIMER\_CCER.CC1P=0。
5. 选择触发输入信号，配置 ATIMER\_SMCR.TS[2:0]=101，T11FP1 作为 TRGI。
6. 设定从模式控制器为复位模式，配置 ATIMER\_SMCR.SMS[2:0]=100。
7. 打开通道使能，配置 ATIMER\_CCER.CC1E=1。
8. 使能计数器，配置 ATIMER\_CR1.CEN=1。

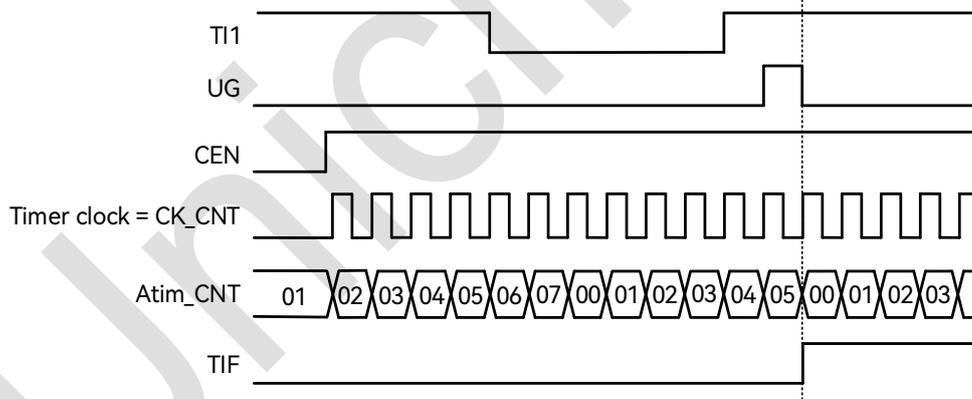


图 12-40: 复位模式下的时序

## 12.4.27 门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。图例中的配置步骤如下：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1 功能。

2. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 ATIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 ATIMER\_CCER.CC1P=1。
5. 选择触发输入信号，配置 ATIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
6. 设定从模式控制器为门控模式，配置 ATIMER\_SMCR.SMS[2:0]=101。
7. 打开通道使能，配置 ATIMER\_CCER.CC1E=1。
8. 使能计数器，配置 ATIMER\_CR1.CEN=1。

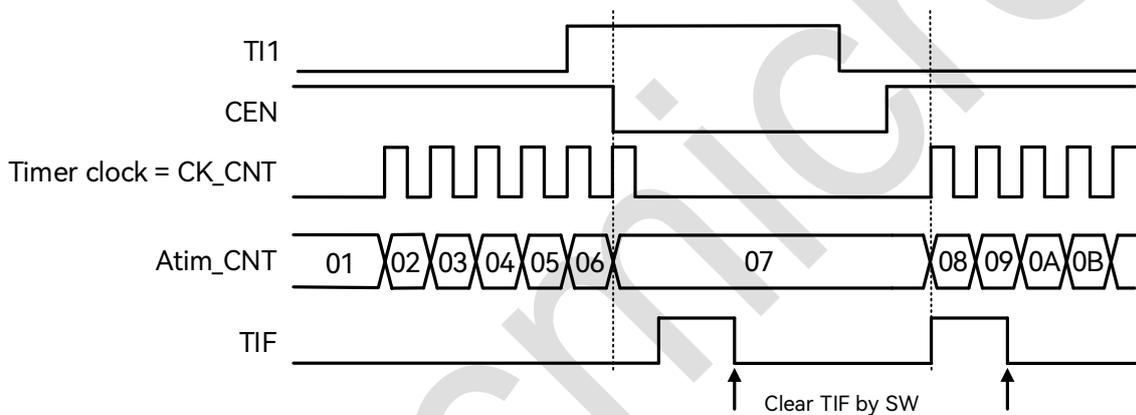


图 12-41: 门控模式下的时序

### 12.4.28 触发模式

计数器在外部输入的某个事件到来后才开始计数。**错误!未找到引用源。**例中的配置步骤如下:

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1 功能。
2. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 ATIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 ATIMER\_CCER.CC1P=0。
5. 选择触发输入信号，配置 ATIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
6. 设定从模式控制器为触发模式，配置 ATIMER\_SMCR.SMS[2:0]=110。
7. 打开通道使能，配置 ATIMER\_CCER.CC1E=1。

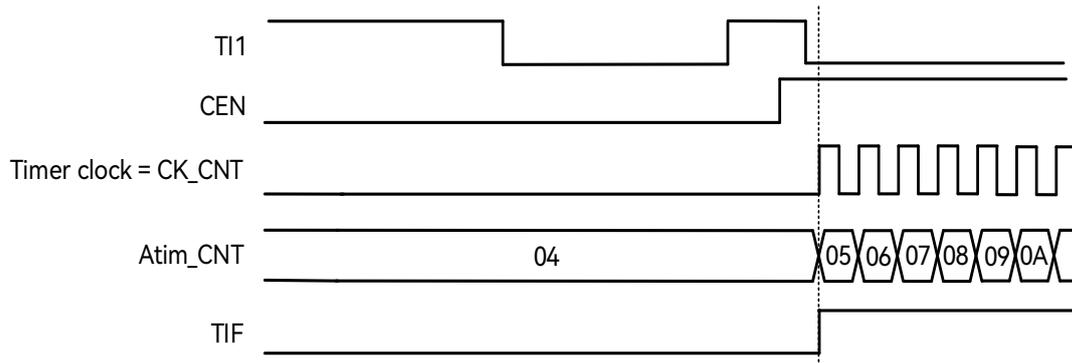


图 12-42: 触发器模式下的时序

### 12.4.29 外部事件触发的外部时钟计数模式

可以将 ETR 设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到 TI1 的上升沿之后，计数器开始以 ETR 输入的上升沿计数。配置步骤如下：

1. 在 GPIO 模块中，配置相应管脚为 ATIMER\_CH1，ATIMER\_ETR 功能。
2. 设置 ETP 进行沿选择，ATIMER\_SMCR.ETP=0。
3. 设置 ETR 分频比，配置 ATIMER\_SMCR.ETPS[1:0]=01。
4. 配置输入滤波时间，ATIMER\_SMCR.ETF[3:0]=0000。
5. 置位 ECE 寄存器，使能外部时钟模式 2，ATIMER\_SMCR.ECE=1。
6. 关闭通道使能，配置 ATIMER\_CCER.CC1E=0 确保之后通道配置成功。
7. 选择输入通道，配置 ATIMER\_CCMR1.CC1S=01。
8. 选择计数有效沿，配置 ATIMER\_CCER.CC1P=0。
9. 选择触发输入信号，配置 ATIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
10. 设定从模式控制器为触发模式，配置 ATIMER\_SMCR.SMS[2:0]=110。
11. 打开通道使能，配置 ATIMER\_CCER.CC1E=1。

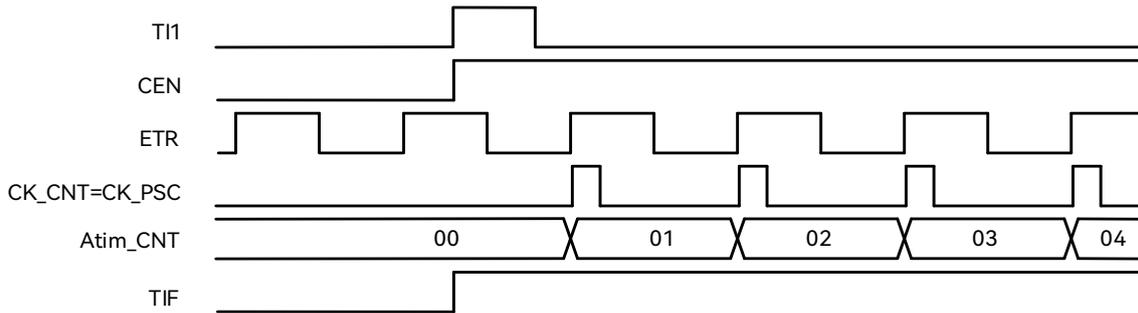


图 12-43: 外部时钟模式 2 和触发模式下的时序

### 12.4.30 DMA 访问

ATIMER 支持 7 种 DMA 请求，分别为 4 个 CC 通道请求、外部触发请求、更新事件触发请求和 COM 触发请求。

其中每个 CC 通道各自产生一个 DMA 请求，在捕捉模式下用于将 ATIMER\_CCRx 中的内容传输给 RAM，在比较模式下则用于将 RAM 中的数据写入 CCRx；CC 通道的 DMA 请求可以配置为单次传输或 Burst 传输 (CCxBURSTEN)，单次传输仅访问 ATIMER\_CCRx 寄存器，Burst 传输则根据 ATIMER\_DCR 寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件、更新事件和 COM 事件也可以产生 DMA 请求，当这些请求发生时，会启动 DMA Burst 传输，向 ATIMER 内部 1 个或多个寄存器写入数据，或者从 ATIMER 读取 1 个或多个寄存器值。

表 12-3: DMA 访问计数方式

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIMER_CH1	0	0	Read CCR1	1
		1	Write CCR1	
ATIMER_CH1	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_CH2	0	0	Read CCR2	1
		1	Write CCR2	
ATIMER_CH2	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_CH3	0	0	Read CCR3	1
		1	Write CCR3	

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIMER_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

此外，外部触发事件、更新事件和 COM 事件也可以产生 DMA 请求，当这些请求发生时，会启动 DMA Burst 传输，向 ATIMER 内部 1 个或多个寄存器写入数据，或者从 ATIMER 读取 1 个或多个寄存器值。

### 12.4.31 输入异或功能

通道 1~3 的输入信号可以被异或起来之后，接入到通道 1 的滤波和边沿电路输入，用于通道 1 的输入捕捉或者触发。

ATIMER\_CR2 寄存器的 TI1S 位用于选择通道 1 的输入是否来自于三个通道输入的异或。

### 12.4.32 Debug 模式

当进入 debug 模式后，定时器可以停止或继续工作，其行为由 STOP\_EN 位定义。

Debug 时当定时器被停止后，其输出会被禁止（MOE 清零），根据寄存器配置，此时的输出信号可以被 force 成 inactive 或由 GPIO 模块控制。

### 12.4.33 DMA Burst

ATIMER 支持 DMA 和 DMA-Burst 访问，可以配置 ATIMER 在特定事件发生时触发 DMA 请求，

可以将 CCR 中的捕捉结果写入 RAM，或者从 RAM 中将一个或多个寄存器内容写入 ATIMER 的 Preload 寄存器中。

DMA-Burst 支持一个事件触发连续多次 DMA 请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA 控制器需将外设目标地址指向一个虚拟寄存器 ATIMER\_DMAR。在特定的定时器事件发生时，ATIMER 会连续发射多个 DMA 请求。每个 DMA 对 ATIMER\_DMAR 的写操作都会被 ATIMER 重新定向到实际的功能寄存器上。

DBL 寄存器用于设置 DMA burst 长度，DBA 寄存器用于设置 DMA 访问 ATIMER 内部的基地址（相对于 ATIMER\_CR 的 offset）。

DMA-Burst 模式下，DMA 所有访问都要指向 DMAR 虚拟寄存器，由 ATIMER 自动根据访问来累加内部 offset 地址。DBA 寄存器用于指定 ATIMER 内部首次 DMA 传输的目标地址，而 DBL 用于指定 Burst 长度。

## 12.5 寄存器描述

ATIMER 寄存器基地址：0x40007400

表 12-4: ATIMER 寄存器列表

偏置	名称	描述
0x00	ATIMER_CR1	ATIMER 控制寄存器 1
0x04	ATIMER_CR2	ATIMER 控制寄存器 2
0x08	ATIMER_SMCR	ATIMER 从机模式控制寄存器
0x0C	ATIMER_DIER	ATIMER DMA 和中断使能寄存器
0x10	ATIMER_SR	ATIMER 状态寄存器
0x14	ATIMER_EGR	ATIMER 事件产生寄存器
0x18	ATIMER_CCMR1	ATIMER 捕捉/比较模式寄存器 1
0x1C	ATIMER_CCMR2	ATIMER 捕捉/比较模式寄存器 2
0x20	ATIMER_CCER	ATIMER 捕捉/比较使能寄存器
0x24	ATIMER_CNT	ATIMER 计数器寄存器
0x28	ATIMER_PSC	ATIMER 预分频寄存器
0x2C	ATIMER_ARR	ATIMER 自动重载寄存器
0x30	ATIMER_RCR	ATIMER 重复计数寄存器

偏置	名称	描述
0x34	ATIMER_CCR1	ATIMER 捕捉/比较寄存器 1
0x38	ATIMER_CCR2	ATIMER 捕捉/比较寄存器 2
0x3C	ATIMER_CCR3	ATIMER 捕捉/比较寄存器 3
0x40	ATIMER_CCR4	ATIMER 捕捉/比较寄存器 4
0x44	ATIMER_BDTR	ATIMER 刹车和死区控制寄存器
0x48	ATIMER_DCR	ATIMER DMA 控制寄存器
0x4C	ATIMER_DMAR	ATIMER DMA 访问寄存器
0x50	ATIMER_CCMR3	ATIMER 比较模式寄存器 3
0x54	ATIMER_CCR5	ATIMER 比较寄存器 5
0x58	ATIMER_PMEN	ATIMER PWM 移相使能寄存器
0x5c	ATIMER_PMC1	ATIMER 移相偏移寄存器 1
0x60	ATIMER_PMC2	ATIMER 移相偏移寄存器 2
0x64	ATIMER_PMC3	ATIMER 移相偏移寄存器 3
0x68	ATIMER_BKCTL	ATIMER 刹车输入控制寄存器

### 12.5.1 ATIMER 控制寄存器 1 ATIMER\_CR1 (偏移: 00h)

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	STOP_EN	RW	0x1	ATIM 断点使能位, 在软件设置断点后, 禁止计数器计数: 0: 计数器正常工作 1: 计数器停止计数
16	DBOEN	RW	0x0	Debug 模式下三路互补输出使能控制寄存器: 0: Debug 模式下三路互补输出高阻态 1: Debug 模式下三路互补输出电平由 DBCC*寄存器决定
15	DBCC3N	RW	0x0	Debug 模式下 OC3N 输出寄存器: 0: Debug 模式下输出 0 1: Debug 模式下输出 1
14	DBCC3	RW	0x0	Debug 模式下 OC3 输出寄存器: 0: Debug 模式下输出 0 1: Debug 模式下输出 1
13	DBCC2N	RW	0x0	Debug 模式下 OC2N 输出寄存器: 0: Debug 模式下输出 0 1: Debug 模式下输出 1
12	DBCC2	RW	0x0	Debug 模式下 OC2 输出寄存器: 0: Debug 模式下输出 0

比特	名称	属性	复位值	描述
				1: Debug 模式下输出 1
11	DBCC1N	RW	0x0	Debug 模式下 OC1N 输出寄存器: 0: Debug 模式下输出 0 1: Debug 模式下输出 1
10	DBCC1	RW	0x0	Debug 模式下 OC1 输出寄存器: 0: Debug 模式下输出 0 1: Debug 模式下输出 1
9:8	CKD	RW	0x0	Dead time 和数字滤波时钟频率分频寄存器 (相对 CK_INT 的分频比): 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2*t_{CK\_INT}$ 10: $t_{DTS}=4*t_{CK\_INT}$ 11: RFU, 禁止使用
7	ARPE	RW	0x0	Auto-reload 预装载使能: 0: ARR 寄存器不使能 preload 1: ARR 寄存器使能 preload
6:5	CMS	RW	0x0	计数器对齐模式选择: 00: 边沿对齐模式 01: 中央对齐模式 1, 输出比较中断标志仅在计数器向下计数的过程中置位 10: 中央对齐模式 2, 输出比较中断标志仅在计数器向上计数的过程中置位 11: 中央对齐模式 3, 输出比较中断标志在计数器向上向下计数的过程中都会置位
4	DIR	RW	0x0	计数方向寄存器: 0: 向上计数 1: 向下计数 注: 当定时器配置为中央计数模式或编码器模式时, 此寄存器只读。
3	OPM	RW	0x0	单脉冲输出模式: 0: Update Event 发生时计数器不停止 1: Update Event 发生时计数器停止 (自动清零 CEN)
2	URS	RW	0x0	更新请求选择: 0: 以下事件能够产生 update 中断或 DMA 请求 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update

比特	名称	属性	复位值	描述
				1: 仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	UDIS	RW	0x0	禁止 update: 0: 使能 update 事件; 以下事件发生时产生 update 事件 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1: 禁止 update 事件, 不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	RW	0x0	计数器使能: 0: 计数器关闭 1: 计数器使能 注: 外部触发模式可以自动置位 CEN。

### 12.5.2 ATIMER 控制寄存器 2 ATIMER\_CR2 (偏移: 04h)

比特	名称	属性	复位值	描述
31:19	RSV	-	-	保留
18	CC4BUEN	RW	0x0	捕捉比较通道 4 的 DMA 模式配置: 0: single 传输, DMA 读取 CCR4 寄存器的值传输到 RAM 1: RAM->ATIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
17	CC3BUEN	RW	0x0	捕捉比较通道 3 的 DMA 模式配置: 0: CCR3->RAM, DMA 读取 CCR3 寄存器的值传输到 RAM 1: RAM->ATIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
16	CC2BUEN	RW	0x0	捕捉比较通道 2 的 DMA 模式配置: 0: CCR2->RAM, DMA 读取 CCR2 寄存器的值传输到 RAM 1: RAM->ATIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
15	CC1BUEN	RW	0x0	捕捉比较通道 1 的 DMA 模式配置: 0: CCR1->RAM, DMA 读取 CCR1 寄存器的值传输到 RAM 1: RAM->ATIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
14	OIS4	RW	0x0	参考 OIS1
13	OIS3N	RW	0x0	参考 OIS1N
12	OIS3	RW	0x0	参考 OIS1

比特	名称	属性	复位值	描述
11	OIS2N	RW	0x0	参考 OIS1N
10	OIS2	RW	0x0	参考 OIS1
9	OIS1N	RW	0x0	定义 OC1N 的输出 IDLE 状态: 0: 当 MOE=0 时, 经过 dead time 后, OC1N=0 1: 当 MOE=0 时, 经过 dead time 后, OC1N=1
8	OIS1	RW	0x0	定义 OC1 的输出 IDLE 状态: 0: 当 MOE=0 时 (如果使能了互补输出, 需经过 dead time 后), OC1=0 1: 当 MOE=0 时 (如果使能了互补输出, 需经过 dead time 后), OC1=1
7	TI1S	RW	0x0	ATIMER 输入 TI1 选择: 0: ATIMER_CH1 引脚连接到 TI1 输入 1: ATIMER_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	MMS	RW	0x0	主机模式选择, 用于配置主机模式下向从机发送的同步触发信号 (TRGO) 源: 000: ATIMER_EGR 的 UG 寄存器被用作 TRGO 001: 计数器使能信号 CNT_EN 被用作 TRGO, 可用于同时启动多个定时器 010: UE (update event) 信号被用作 TRGO 011: 比较脉冲, 如果 CC1IF 标志将要置位, TRGO 输出一个正脉冲 100: OC1REF 用作 TRGO 101: OC2REF 用作 TRGO 110: OC3REF 用作 TRGO 111: OC4REF 用作 TRGO 注: 从机定时器或 ADC 必须事先使能工作时钟, 才能接收主机定时器发送的 TRGO。
3	CCDS	RW	0x0	捕捉/比较 DMA 选择: 0: 捕捉/比较事件发生时发送 DMA 请求 1: Update Event 发生时发送 DMA 请求
2	CCUS	RW	0x0	捕捉/比较控制寄存器更新选择: 0: 当捕捉/比较控制寄存器使能了 preload (CCPC=1), 他们仅在置位 COMG 寄存器时更新。 1: 当捕捉/比较控制寄存器使能了 preload (CCPC=1), 他们在置位 COMG 寄存器或者 TRGI 上升沿时更新。
1	RSV	-	-	保留

比特	名称	属性	复位值	描述
0	CCPC	RW	0x0	捕捉/比较预装载控制： 0: CCxE, CCxNE, OCxM 寄存器不使能 preload 1: CCxE, CCxNE, OCxM 寄存器使能 preload 注：此寄存器仅在拥有互补输出功能的通道上有效。

### 12.5.3 ATIMER 从机模式控制寄存器 ATIMER\_SMCR (偏移: 08h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	ETP	RW	0x0	外部触发信号极性配置： 0: 高电平或上升沿有效 1: 低电平或下降沿有效
14	ECE	RW	0x0	外部时钟使能： 0: 关闭外部时钟模式 2 1: 使能外部时钟模式 2，计数器时钟为 ETRF 有效沿
13:12	ETPS	RW	0x0	外部触发信号预分频寄存器： 外部触发信号 ETRP 的频率最多只能是 ATIMER 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频
11:8	ETF	RW	0x0	外部触发信号滤波时钟和长度选择： 0000: 无滤波 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: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16, N=8$

比特	名称	属性	复位值	描述
				1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=5$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=6$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=8$
7	MSM	RW	0x0	主/从模式选择: 0: 无动作, 以便于通过 TRGO 将主机定时器和从机定时器完美同步起来 (完美同步时主机寄存器 CR2.MMS 需配为 010) 1: 延迟主机 TRGI 触发信号 注: 仅用于从机模式的触发模式 (SMS=110), 且由从机模式配置。
6:4	TS	RW	0x0	触发选择, 用于选择同步计数器的触发源: 000: 内部触发信号 (ITR0) 001: 内部触发信号 (ITR1) 010: 内部触发信号 (ITR2) 011: 内部触发信号 (ITR3) 100: TI1 边沿检测 (TI1F_ED) 101: 滤波后 TI1 (TI1FP1) 110: 滤波后 TI2 (TI2FP2) 111: 外部触发输入 (ETRF) 注: 仅当 SMS=000 即禁止从机模式的情况下, 可以改写 TS 寄存器。
3	RSV	-	-	保留
2:0	SMS	RW	0x0	从机模式选择: 000: 从机模式禁止; CEN 使能后预分频电路时钟源来自内部时钟 001: Encoder 模式 1; 计数器使用 TI2FP2 边沿, 根据 TI1FP1 电平高低来计数 010: Encoder 模式 2; 计数器使用 TI1FP1 边沿, 根据 TI2FP2 电平高低来计数 011: Encoder 模式 3; 计数器同时使用 TI1FP1 和 TI2FP2 边沿, 根据其他输入信号电平来计数 100: 复位模式; TRGI 上升沿初始化计数器, 并触发寄存器 update 101: 闸门模式; TRGI 为高电平时, 计数时钟使能, TRGI 为低电平时, 计数时钟停止 110: 触发模式; TRGI 上升沿触发计数器开始计数 (不会复位计数器) 111: 外部时钟模式 1; TRGI 上升沿直接驱动计数器

## 12.5.4 ATIMER DMA 和中断使能寄存器 ATIMER\_DIER (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	CC5IE	RW	0x0	捕捉/比较通道 5 中断使能: 0: 禁止捕捉/比较 5 中断 1: 允许捕捉/比较 5 中断
15	RSV	-	-	保留
14	TDE	RW	0x0	外部触发 DMA 请求使能: 0: 从机模式下, 禁止外部触发事件产生 DMA 请求 1: 从机模式下, 允许外部触发事件产生 DMA 请求 (可用于自动更新 preload 寄存器)
13	COMDE	RW	0x0	COM 事件 DMA 请求使能: 0: COM 事件发生时, 禁止产生 DMA 请求 1: COM 事件发生时, 允许产生 DMA 请求
12	CC4DE	RW	0x0	捕捉比较通道 4 的 DMA 请求使能: 0: 禁止 CC4 DMA 请求 1: 允许 CC4 DMA 请求
11	CC3DE	RW	0x0	捕捉比较通道 3 的 DMA 请求使能: 0: 禁止 CC3 DMA 请求 1: 允许 CC3 DMA 请求
10	CC2DE	RW	0x0	捕捉比较通道 2 的 DMA 请求使能: 0: 禁止 CC2 DMA 请求 1: 允许 CC2 DMA 请求
9	CC1DE	RW	0x0	捕捉比较通道 1 的 DMA 请求使能: 0: 禁止 CC1 DMA 请求 1: 允许 CC1 DMA 请求
8	UDE	RW	0x0	Update Event DMA 请求使能: 0: Update Event 发生时, 禁止产生 DMA 请求 1: Update Event 发生时, 允许产生 DMA 请求
7	BIE	RW	0x0	刹车事件中断使能: 0: 禁止刹车事件中断 1: 允许刹车事件中断
6	TIE	RW	0x0	触发事件中断使能: 0: 禁止触发事件中断 1: 允许触发事件中断

比特	名称	属性	复位值	描述
5	COMIE	RW	0x0	COM 事件中断使能: 0: 禁止 COM 事件中断 1: 允许 COM 事件中断
4	CC4IE	RW	0x0	捕捉/比较通道 4 中断使能: 0: 禁止捕捉/比较 4 中断 1: 允许捕捉/比较 4 中断
3	CC3IE	RW	0x0	捕捉/比较通道 3 中断使能: 0: 禁止捕捉/比较 3 中断 1: 允许捕捉/比较 3 中断
2	CC2IE	RW	0x0	捕捉/比较通道 2 中断使能: 0: 禁止捕捉/比较 2 中断 1: 允许捕捉/比较 2 中断
1	CC1IE	RW	0x0	捕捉/比较通道 1 中断使能: 0: 禁止捕捉/比较 1 中断 1: 允许捕捉/比较 1 中断
0	UIE	RW	0x0	Update 事件中断使能: 0: 禁止 Update 事件中断 1: 允许 Update 事件中断

### 12.5.5 ATIMER 状态寄存器 ATIMER\_SR (偏移: 10h)

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	HARDFULT	W1C	0x0	Hardfault 状态位, 硬件置位, 软件写 1 清零
16	CC5IF	W1C	0x0	捕捉/比较通道 5 的中断: 参考CC1IF
15:13	RSV	-	-	保留
12	CC4OF	W1C	0x0	捕捉/比较通道 4 的 Overcapture 状态 参考CC1OF
11	CC3OF	W1C	0x0	捕捉/比较通道 3 的 Overcapture 状态 参考CC1OF
10	CC2OF	W1C	0x0	捕捉/比较通道 2 的 Overcapture 状态 参考CC1OF
9	CC1OF	W1C	0x0	捕捉/比较通道 1 的 Overcapture 状态: 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位, 软件写 1 清零。

比特	名称	属性	复位值	描述
				0: 无 overcapture 事件 1: 在CC1IF标志为1的情况下发生新的捕捉
8	RSV	-	-	保留
7	BIF	W1C	0x0	刹车事件中断标志, 硬件置位, 软件写 1 清零
6	TIF	W1C	0x0	触发事件中断标志, 硬件置位, 软件写 1 清零
5	COMIF	W1C	0x0	COM 事件中断标志, 硬件置位, 软件写 1 清零
4	CC4IF	W1C	0x0	捕捉/比较通道 4 中断标志 参考CC1IF
3	CC3IF	W1C	0x0	捕捉/比较通道 3 中断标志 参考CC3IF
2	CC2IF	W1C	0x0	捕捉/比较通道 2 中断标志 参考CC2IF
1	CC1IF	W1C	0x0	捕捉/比较通道 1 中断标志: 如果 CC1 通道配置为输出: CC1IF 在计数值等于比较值时置位, 软件写 1 清零。 如果CC1通道配置为输入: 发生捕捉事件时置位, 软件写1清零, 或者软件读ATIMER_CCR1自动清零。
0	UIF	W1C	0x0	Update 事件中断标志, 硬件置位, 软件写 1 清零。 当以下事件发生时, UIF 置位, 并更新 shadow 寄存器 -重复计数器=0, 并且 UDIS=0 的情况下, 计数器发生溢出 -URS=0 且 UDIS=0 的情况下, 软件置位 UG 寄存器初始化计数器 -URS=0且UDIS=0的情况下, 触发事件初始化计数器

### 12.5.6 ATIMER 事件产生寄存器 ATIMER\_EGR (偏移: 14h)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	CC5G	W	0x0	捕捉/比较通道 5 软件触发, 参考 CC1G
15:8	RSV	-	-	保留
7	BG	W	0x0	软件刹车事件, 软件置位此寄存器产生刹车事件, 硬件自动清零
6	TG	W	0x0	软件触发事件, 软件置位此寄存器产生触发事件, 硬件自动清零
5	COMG	W	0x0	软件 COM 事件, 软件写 1 置位, 硬件自动清零
4	CC4G	W	0x0	捕捉/比较通道 4 软件触发, 参考 CC1G
3	CC3G	W	0x0	捕捉/比较通道 3 软件触发, 参考 CC1G

比特	名称	属性	复位值	描述
2	CC2G	W	0x0	捕捉/比较通道 2 软件触发, 参考 CC1G
1	CC1G	W	0x0	捕捉/比较通道 1 软件触发, 硬件自动清零。 如果 CC1 通道配置为输出: CC1IF 置位, 在使能的情况下可以产生相应的中断和 DMA 请求。 如果 CC1 通道配置为输入: 当前计数值被捕捉到 ATIMER_CCR1 寄存器, CC1IF 置位, 在使能的情况下可以产生相应的中断和 DMA 请求。
0	UG	W	0x0	软件 Update 事件, 软件置位此寄存器产生 Update 事件, 硬件自动清零。 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器, 预分频计数器被清零。

### 12.5.7 ATIMER 捕捉/比较模式寄存器 1 ATIMER\_CCMR1 (偏移: 18h)

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能:

#### 1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC2CE	RW	0x0	输出比较 2 清零使能, 参考 OC1CE
14:12	OC2M	RW	0x0	输出比较 2 模式配置, 参考 OC1M
11	OC2PE	RW	0x0	输出比较 2 预装载使能, 参考 OC1PE
10	OC2FE	RW	0x0	输出比较 2 快速使能, 参考 OC1FE
9:8	CC2S	RW	0x0	捕捉/比较 2 通道选择: 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: CC2S 仅在通道关闭时 (CC2E=0) 可以写
7	OC1CE	RW	0x0	输出比较 1 清零使能: 0: OC1REF 不受 ETRF 影响 1: 检测到 ETRF 高电平时, 自动清零 OC1REF
6:4	OC1M	RW	0x0	输出比较 1 模式配置, 此寄存器定义 OC1REF 信号的行为: 000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出

比特	名称	属性	复位值	描述
				001: CCR1=CNT 时, 将 OC1REF 置高 010: CCR1=CNT 时, 将 OC1REF 置低 011: CCR1=CNT 时, 翻转 OC1REF 100: OC1REF 固定为低 (inactive) 101: OC1REF 固定为高 (active) 110: PWM 模式 1 - 在向上计数时, OC1REF 在 CNT<CCR1 时置高, 否则置低; 在向下计数时, OC1REF 在 CNT>CCR1 时置低, 否则置高。 111: PWM 模式 2 - 在向上计数时, OC1REF 在 CNT<CCR1 时置低, 否则置高; 在向下计数时, OC1REF 在 CNT>CCR1 时置高, 否则置低。
3	OC1PE	RW	0x0	输出比较 1 预装载使能 0: CCR1 preload 寄存器无效, CCR1 可以直接写入 1: CCR1 preload 寄存器有效, 针对 CCR1 的读写操作都是访问 preload 寄存器, 当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	OC1FE	RW	0x0	输出比较 1 快速使能 0: 关闭快速使能, trigger 输入不会影响比较输出 1: 打开快速使能, trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出, 而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	CC1S	RW	0x0	捕捉/比较 1 通道选择 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: CC1S 仅在通道关闭时 (CC1E=0) 可以写

## 2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC2F	RW	0x0	输入捕捉 2 滤波
11:10	IC2PSC	RW	0x0	输入捕捉 2 预分频
9:8	CC2S	RW	0x0	捕捉/比较 2 通道选择: 00: CC2 通道配置为输出

比特	名称	属性	复位值	描述
				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: CC2S仅在通道关闭时 (CC2E=0) 可以写
7:4	IC1F	RW	0x0	输入捕捉 1 滤波: 此寄存器定义 TI1 的采样频率和滤波长度 0000: 无滤波, 使用 $f_{DTS}$ 采样 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
3:2	IC1PSC	RW	0x0	输入捕捉 1 预分频: 00: 无分频 01: 每 2 个事件输入产生一次捕捉 10: 每 4 个事件输入产生一次捕捉 11: 每 8 个事件输入产生一次捕捉 IC1PSC寄存器在CC1E=0时复位
1:0	CC1S	RW	0x0	捕捉/比较 1 通道选择: 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: CC1S仅在通道关闭时 (CC1E=0) 可以写

## 12.5.8 ATIMER 捕捉/比较模式寄存器 2 ATIMER\_CCMR2 (偏移: 1Ch)

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能:

### 1. 输出比较模式

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17:16	OC4SEL	RW	0x0	OC4REF 上升沿/下降沿脉冲触发选择: 00: 无脉冲触发 01: OC4REF 上升沿触发脉冲 10: OC4REF 下降沿触发脉冲 11: OC4REF上升沿/下降沿触发脉冲
15	OC4CE	RW	0x0	输出比较 4 清零使能, 参考 OC1CE
14:12	OC4M	RW	0x0	输出比较 4 模式配置, 参考 OC1M
11	OC4PE	RW	0x0	输出比较 4 预装载使能, 参考 OC1PE
10	OC4FE	RW	0x0	输出比较 4 快速使能, 参考 OC1FE
9:8	CC4S	RW	0x0	捕捉/比较 4 通道选择: 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: CC4S仅在通道关闭时 (CC4E=0) 可以写
7	OC3CE	RW	0x0	输出比较 4 清零使能: 0: OC4REF 不受 ETRF 影响 1: 检测到ETRF高电平时, 自动清零OC4REF
6:4	OC3M	RW	0x0	输出比较 3 模式配置, 此寄存器定义 OC3REF 信号的行为: 000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001: CCR3=CNT 时, 将 OC3REF 置高 010: CCR3=CNT 时, 将 OC3REF 置低 011: CCR3=CNT 时, 翻转 OC3REF 100: OC3REF 固定为低 (inactive) 101: OC3REF 固定为高 (active) 110: PWM 模式 1——在向上计数时, OC3REF 在 CNT<CCR3 时置高, 否则置低; 在向下计数时, OC3REF 在 CNT>CCR3 时置低, 否则置高。 111: PWM模式2——在向上计数时, OC3REF在CNT<CCR3

比特	名称	属性	复位值	描述
				时置低，否则置高；在向下计数时，OC3REF在CNT>CCR3时置高，否则置低。
3	OC3PE	RW	0x0	输出比较 3 预装载使能： 0: CCR3 preload 寄存器无效，CCR4 可以直接写入 1: CCR3 preload寄存器有效，针对CCR4的读写操作都是访问preload寄存器，当update event发生时才将preload寄存器的内容转移到shadow寄存器中。
2	OC3FE	RW	0x0	输出比较 3 快速使能： 0: 关闭快速使能，trigger 输入不会影响比较输出 1: 打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况。 此功能仅在当前通道配置为PWM1或PWM2模式时有效。
1:0	CC3S	RW	0x0	捕捉/比较 3 通道选择： 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：CC3S仅在通道关闭时（CC3E=0）可以写

## 2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC4F	RW	0x0	输入捕捉 4 滤波
11:10	IC4PSC	RW	0x0	输入捕捉 4 预分频
9:8	CC4S	RW	0x0	捕捉/比较 4 通道选择： 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：CC4S仅在通道关闭时（CC4E=0）可以写
7:4	IC3F	RW	0x0	输入捕捉 3 滤波： 此寄存器定义 TI3 的采样频率和滤波长度 0000: 无滤波，使用 $f_{DTS}$ 采样 0001: $f_{SAMPLING}=f_{CK\_INT}$ , N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$ , N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$ , N=8

比特	名称	属性	复位值	描述
				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: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=8$
3:2	IC3PSC	RW	0x0	输入捕捉 3 预分频: 00: 无分频 01: 每 2 个事件输入产生一次捕捉 10: 每 4 个事件输入产生一次捕捉 11: 每 8 个事件输入产生一次捕捉 注: IC3PSC寄存器在CC3E=0时复位
1:0	CC3S	RW	0x0	捕捉/比较 3 通道选择: 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: CC3S仅在通道关闭时 (CC1E=0) 可以写

### 12.5.9 ATIMER 捕捉/比较使能寄存器 ATIMER\_CCER (偏移: 20h)

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	CC5P	RW	0x0	捕捉/比较 5 输出极性, 参考 CC1P
16	CC5E	RW	0x0	捕捉/比较 5 输出使能, 参考 CC1E
15:14	RSV	-	-	保留
13	CC4P	RW	0x0	捕捉/比较 4 输出极性, 参考 CC1P
12	CC4E	RW	0x0	捕捉/比较 4 输出使能, 参考 CC1E
11	CC3NP	RW	0x0	捕捉/比较 3 互补输出极性, 参考 CC1NP
10	CC3NE	RW	0x0	捕捉/比较 3 互补输出使能, 参考 CC1NE

比特	名称	属性	复位值	描述
9	CC3P	RW	0x0	捕捉/比较 3 输出极性, 参考 CC1P
8	CC3E	RW	0x0	捕捉/比较 3 输出使能, 参考 CC1E
7	CC2NP	RW	0x0	捕捉/比较 2 互补输出极性, 参考 CC1NP
6	CC2NE	RW	0x0	捕捉/比较 2 互补输出使能, 参考 CC1NE
5	CC2P	RW	0x0	捕捉/比较 2 输出极性, 参考 CC1P
4	CC2E	RW	0x0	捕捉/比较 2 输出使能, 参考 CC1E
3	CC1NP	RW	0x0	捕捉/比较 1 互补输出极性: 0: OC1N 高电平为 active 1: OC1N 低电平为 active
2	CC1NE	RW	0x0	捕捉/比较 1 互补输出使能: 0: OC1N 无效, OC1N 电平由 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 寄存器决定
1	CC1P	RW	0x0	捕捉/比较 1 输出极性: CC1 通道配置为输出时: 0: OC1 高电平 active 1: OC1 低电平 active CC1 通道配置为输入时: 0: 非取反模式-捕捉在 IC1 的上升沿进行 1: 取反模式-捕捉在 IC1 的下降沿进行
0	CC1E	RW	0x0	捕捉/比较 1 输出使能: CC1 通道配置为输出时: 0: OC1 不 active 1: OC1 active CC1 通道配置为输入时: 0: 关闭捕捉功能 1: 使能捕捉功能

以下控制寄存器和互补输出通道的状态对应表, 其中 MOE 为定时器总输出使能位, OSSI 为 IDLE 状态 (MOE=0) 下的 off\_state 选择位, OSSR 为 RUN 状态 (MOE=1) 下的 off\_state 选择位。

Off-state:

控制寄存器					输出状态	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止 (不由 ATIMER 驱动), OCx=0, OCx_EN=0	输出禁止 (不由 ATIMER 驱动), OCxN=0, OCxN_EN=0

控制寄存器					输出状态	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态
		0	0	1	输出禁止 (不由ATIMER驱动), OCx=0, OCx_EN=0	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	输出禁止 (不由 ATIMER 驱动), OCx=0, OCx_EN=0
		0	1	1	OCREF + Polarity + dead-time, OCx_EN=1	OCREF反相 + Polarity + dead-time, OCxN_EN=1
		1	0	0	输出禁止 (不由 ATIMER 驱动), OCx=CCxP, OCx_EN=0	输出禁止 (不由 ATIMER 驱动), OCxN=CCxNP, OCxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP, OCx_EN=1	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + Polarity + dead-time, OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time, OCxN_EN=1
0	0	X	0	0	输出关闭 (不由ATIMER驱动), OCx=CCxP, OCx_EN=0	输出关闭 (不由ATIMER驱动), OCxN=CCxNP, OCxN_EN=0
	0		0	1	输出关闭 (不由ATIMER驱动)	
	0		1	0	如果无时钟: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0	
	0		1	1	如果有时钟: 经过死区时间后OCx=OISx, OCxN=OISxN	
	1		0	0	输出关闭 (不由ATIMER驱动), OCx=CCxP, OCx_EN=0	输出关闭 (不由ATIMER驱动), OCxN=CCxNP, OCxN_EN=0
	1		0	1	Off-state (输出使能, inactive输出)	
	1		1	0	如果无时钟: OCx=CCxP, OCx_EN=1, OCxN=CCxNP	

控制寄存器					输出状态	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态
	1		1	1	OCxN_EN=1 如果有时钟：经过死区时间后OCx=OISx, OCxN=OISxN	

### 12.5.10 ATIMER 计数器寄存器 ATIMER\_CNT (偏移: 24h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT	RW	0x0	计数值

### 12.5.11 ATIMER 预分频寄存器 ATIMER\_PSC (偏移: 28h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC	RW	0x0	计数器时钟 (CK_CNT) 预分频值: $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ 这是一个preload寄存器, 在update事件发生时其内容被载入shadow寄存器。 注: 支持的最高的PWM输出为30MHz, 配置的PSC和ARR需注意该条件。

### 12.5.12 ATIMER 自动重载寄存器 ATIMER\_ARR (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR	RW	0x0	计数溢出时的自动重载值: 这是一个preload寄存器, 在update事件发生时其内容被载入shadow寄存器。

### 12.5.13 ATIMER 重复计数寄存器 ATIMER\_RCR (偏移: 30h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	REP	RW	0x0	重复计数值。REP不为0时, 每次update条件发生时REP递减, 当REP=0时触发update事件。

### 12.5.14 ATIMER 捕捉/比较寄存器 1 ATIMER\_CCR1 (偏移: 34h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR1	RW	0x0	捕捉/比较通道1寄存器 <b>如果通道1配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC1输出。 <b>如果通道1配置为输入:</b> CCR1保存最近一次输入捕捉事件发生时的计数器值, 此时CCR1为只读。

### 12.5.15 ATIMER 捕捉/比较寄存器 2 ATIMER\_CCR2 (偏移: 38h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR2	RW	0x0	捕捉/比较通道2寄存器 <b>如果通道2配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC2输出。 <b>如果通道2配置为输入:</b> CCR2保存最近一次输入捕捉事件发生时的计数器值, 此时CCR2为只读。

### 12.5.16 ATIMER 捕捉/比较寄存器 3 ATIMER\_CCR3 (偏移: 3Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR3	RW	0x0	捕捉/比较通道3寄存器 <b>如果通道3配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC3输出。 <b>如果通道3配置为输入:</b> CCR3保存最近一次输入捕捉事件发生时的计数器值, 此时CCR3为只读。

### 12.5.17 ATIMER 捕捉/比较寄存器 4 ATIMER\_CCR4 (偏移: 40h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR4	RW	0x0	<p>捕捉/比较通道4寄存器</p> <p><b>如果通道4配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC4输出。</p> <p><b>如果通道4配置为输入:</b> CCR4保存最近一次输入捕捉事件发生时的计数器值, 此时CCR4为只读。</p>

### 12.5.18 ATIMER 刹车和死区控制寄存器 ATIMER\_BDTR (偏移: 44h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	MOE	RW	0x0	<p>输出使能主控:</p> <p>此寄存器控制所有通道的输出使能, 每个通道独立的输出使能还需要CCxE和CCxNE来控制。MOE由软件置位, 或者在AOE=1的情况下硬件触发自动置位。当软件进入DBG时会MOE一直置0, 需要再次硬件复位才会退出; 当刹车输入有效时, MOE被硬件异步清零。</p> <p>0: 关闭OC和OCN输出, 具体IO输出状态由OSS1决定 1: 使能OC和OCN输出 (仍需各个通道的CCxE和CCxNE状态来决定是否输出)</p>
14	AOE	RW	0x0	<p>自动输出使能:</p> <p>0: MOE仅能由软件置位 1: MOE可以软件置位, 或者由update事件自动置位</p>
13	BKP	RW	0x0	<p>刹车极性:</p> <p>0: 刹车输入为低电平有效 1: 刹车输入为高电平有效</p>
12	BKE	RW	0x0	<p>刹车使能:</p> <p>0: 禁止刹车输入 1: 允许刹车输入</p>
11	OSSR	RW	0x0	<p>运行状态下的输出关闭状态选择:</p> <p>仅在MOE=1的情况下, 针对使能了互补输出的通道有效。 0: 输出通道不使能时, OC和OCN不驱动GPIO</p>

比特	名称	属性	复位值	描述
				1: 输出通道不使能时, OC和OCN驱动GPIO为无效状态
10	OSSI	RW	0x0	IDLE状态下的输出关闭状态选择: 仅在MOE=0的情况下, 针对输出通道有效。 0: 输出通道不使能时, OC和OCN不驱动GPIO 1: 输出通道不使能时, OC和OCN先驱动空闲状态, 待死区时间结束后, 启动无效状态
9:8	LOCK	RW	0x0	寄存器写保护配置: 00: 无写保护 01: 保护等级1 – DTG, OISx, OISxN, BKE, BKP, AOE不能改写 10: 保护等级2 –在等级1基础上, CCxP, CCxNP, OSSR, OSSI不能改写 11: 保护等级3 –在等级2基础上, OCxM, OcxPE在相应通道配置为输出时不能改写 注: LOCK寄存器在被写入非00值之后无法再改写, 写保护后的寄存器只有在ATIMER模块被复位后才能重新写入。
7:0	DTG	RW	0x0	死区时间插入, 用于配置互补输出插入的死区时间长度: DTG[7:5]=0xx: $DT = DTG[7:0] * t_{DTS}$ DTG[7:5]=10x: $DT = (64+DTG[5:0]) * 2 * t_{DTS}$ DTG[7:5]=110: $DT = (32+DTG[4:0]) * 8 * t_{DTS}$ DTG[7:5]=111: $DT = (32+DTG[4:0]) * 16 * t_{DTS}$

### 12.5.19 ATIMER DMA 控制寄存器 ATIMER\_DCR (偏移: 48h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12:8	DBL	RW	0x0	DMA Burst长度: 对 ATIMER_DMAR 寄存器的读写将触发 burst DMA操作, burst长度为1~18 00000: 长度=1 00001: 长度=2 ..... 10001: 长度=18 其他: 无效值, 禁止写入
7:5	RSV	-	-	保留
4:0	DBA	RW	0x0	DMA基地址, 定义指向寄存器的偏移地址: 00000: ATIMER_CR1

比特	名称	属性	复位值	描述
				00001: ATIMER_CR2 00010: ATIMER_SMCR ..... 注: 当DBA+DBL超出了ATIMER寄存器地址范围, 则实际burst传输到ATIMER最高寄存器地址后自动停止, 即burst长度会缩短。

### 12.5.20 ATIMER DMA 访问寄存器 ATIMER\_DMAR (偏移: 4Ch)

比特	名称	属性	复位值	描述
31:0	DMAR	RW	0x20000	DMA burst访问寄存器: 在使用DMA burst传输时, 将DMA通道外设地址设置为ATIMER_DMAR, ATIMER会根据DBL的值产生多次DMA请求

### 12.5.21 ATIMER 捕捉/比较模式寄存器 3 ATIMER\_CCMR3 (偏移: 50h)

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17:16	OC5SEL	RW	0x0	OC5REF 上升沿/下降沿脉冲触发选择: 00: 无脉冲触发 01: OC5REF 上升沿触发脉冲 10: OC5REF 下降沿触发脉冲 11: OC5REF上升沿/下降沿触发脉冲
15:8	RSV	-	-	保留
7	OC5CE	RW	0x0	输出比较 5 清零使能: 0: OC5REF 不受 ETRF 影响 1: 检测到ETRF高电平时, 自动清零OC5REF
6:4	OC5M	RW	0x0	输出比较 5 模式配置, 此寄存器定义 OC5REF 信号的行为: 000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001: CCR5=Cnt 时, 将 OC5REF 置高 010: CCR5=Cnt 时, 将 OC5REF 置低 011: CCR5=Cnt 时, 翻转 OC5REF 100: OC5REF 固定为低 (inactive) 101: OC5REF 固定为高 (active)

比特	名称	属性	复位值	描述
				110: PWM 模式 1 -在向上计数时, OC5REF 在 CNT<CCR5 时置高, 否则置低; 在向下计数时, OC5REF 在 CNT>CCR5 时置低, 否则置高。 111: PWM模式2 -在向上计数时, OC5REF在CNT<CCR5时置低, 否则置高; 在向下计数时, OC5REF在CNT>CCR5 时置高, 否则置低。
3	OC5PE	RW	0x0	输出比较 5 预装载使能: 0: CCR5 preload 寄存器无效, CCR5 可以直接写入。 1: CCR5 preload寄存器有效, 针对CCR5的读写操作都是访问preload寄存器, 当update event发生时才将preload寄存器的内容转移到shadow寄存器中。
2	OC5FE	RW	0x0	输出比较 5 快速使能: 0: 关闭快速使能, trigger 输入不会影响比较输出。 1: 打开快速使能, trigger 输入会立即将 OC5REF 改变为比较值匹配时的输出, 而不管当前实际比较情况。 此功能仅在当前通道配置为PWM1或PWM2模式时有效。
1:0	RSV	-	-	保留

### 12.5.22 ATIMER 捕捉/比较寄存器 5 ATIMER\_CCR5 (偏移: 54h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR5	RW	0x0	捕捉/比较通道5寄存器: 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC5输出。

### 12.5.23 ATIMER PWM 移相使能寄存器 ATIMER\_PMEN (偏移: 58h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	PM3EN	RW	0x0	通道3移相使能位: 0: 禁止移相功能; 1: 允许移相功能
1	PM2EN	RW	0x0	通道2移相使能位: 0: 禁止移相功能; 1: 允许移相功能
0	PM1EN	RW	0x0	通道1移相使能位: 0: 禁止移相功能; 1: 允许移相功能

**12.5.24 ATIMER 移相偏移寄存器 1 ATIMER\_PMC1 (偏移: 5Ch)**

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PMC1	RW	0x0	比较通道1偏移寄存器

**12.5.25 ATIMER 移相偏移寄存器 2 ATIMER\_PMC2 (偏移: 60h)**

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PMC2	RW	0x0	比较通道2偏移寄存器

**12.5.26 ATIMER 移相偏移寄存器 3 ATIMER\_PMC3 (偏移: 64h)**

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PMC3	RW	0x0	比较通道3偏移寄存器

**12.5.27 ATIMER 刹车输入控制寄存器 ATIMER\_BKCTL (偏移: 68h)**

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
14:12	BKINSEL	RW	0x0	管脚刹车源选择信号: 000: BRK0 001: BRK1 010: BRK2 011: BRK3 100: BRK4 101: BRK5 110: BRK6 111: 未定义
11	RSV	-	-	保留
10	CMP0_BRKEN	RW	0x0	比较器0输出刹车信号使能: 0: 禁止CMP0刹车信号 1: 使能CMP0刹车信号

比特	名称	属性	复位值	描述
9	ADC_BRKEN	RW	0x0	ADC模拟看门狗刹车信号使能: 0: 禁止ADC模拟看门狗刹车信号 1: 使能ADC模拟看门狗刹车信号
8	BRKGATE	RW	0x1	ATIM_BRK引脚输入门控信号: 0: 禁止ATIM_BKR引脚输入门控信号 1: 使能ATIM_BKR引脚输入门控信号
7:4	BRKF	RW	0x0	刹车信号的滤波时钟和长度选择: 0000: 无滤波 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
3	RSV	-	-	保留
2	CMP1_BRKEN	RW	0x0	比较器1输出刹车信号使能: 0: 禁止CMP1刹车信号 1: 使能CMP1刹车信号
1	SVD_BRKEN	RW	0x0	LVD刹车信号使能: 0: 禁止LVD刹车信号 1: 使能LVD刹车信号
0	CODE_BRKEN	RW	0x0	程序出错 (Hardfault) 刹车信号使能: 0: 禁止程序出错刹车信号 1: 使能程序出错刹车信号

## 12.6 使用流程

### 12.6.1 定时计数模式

1. 配置 ATIMER\_CR1 的 DIR，设置计数方向。
2. 配置 ATIMER\_CR1 的 APRE 为 1，使能 Auto-reload 预装载。
3. 配置 ATIMER\_PSC，设置预分频值。
4. 配置 ATIMER\_ARR，设置自动重载值。
5. 配置 ATIMER\_RCR 为 0，不进行重复计数。
6. 配置 ATIMER\_CR1 的 URS 为 1，设置仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求。
7. 配置 ATIMER\_CR1 的 UDIS 为 0，使能 update 事件。
8. 配置 ATIMER\_EGR 的 UG 为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
9. 配置 ATIMER\_CR1 的 CEN 为 1，使能计数器。
10. 配置 ATIMER\_DIER 的 UIE 为 1，允许 Update 事件中断。

### 12.6.2 PWM 模式

1. 配置 ATIMER\_CR1 的 DIR，设置计数方向。
2. 配置 ATIMER\_CR1 的 APRE 为 1，使能 Auto-reload 预装载。
3. 配置 ATIMER\_PSC，设置预分频值。
4. 配置 ATIMER\_ARR，设置自动重载值。
5. 配置 ATIMER\_RCR 为 0，不进行重复计数。
6. 根据输出通道配置 ATIMER\_CCMRx 的 CCxS 为 0，设置通道 x 为输出。
7. 配置 ATIMER\_CCMRx 的 OCxM，设置为 PWM 模式 1/2。
8. 配置 ATIMER\_CCER 的 CCxP，设置输出极性。
9. 配置 ATIMER\_CCER 的 CCxE 为 1，通道 x 输出使能。

10. 配置 ATIMER\_BDTR 的 MOE 为 1，该位为输出使能主控，使能 OC 和 OCN 输出。
11. 配置 ATIMER\_CR1 的 URS 为 1，设置仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求。
12. 配置 ATIMER\_CR1 的 UDIS 为 0，使能 update 事件。
13. 配置 ATIMER\_EGR 的 UG 为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
14. 配置 ATIMER\_CR1 的 CEN 为 1，使能计数器。
15. 配置 ATIMER\_DIER 的 UIE 为 1，允许 Update 事件中断。
16. 配置 ATIMER\_CCRx，设置通道 x 的比较值。

### 12.6.3 输入捕捉模式

1. 配置 ATIMER\_CR1 的 DIR，设置计数方向。
2. 配置 ATIMER\_CR1 的 APRE 为 1，使能 Auto-reload 预装载。
3. 配置 ATIMER\_PSC，设置预分频值。
4. 配置 ATIMER\_ARR，设置自动重载值。
5. 配置 ATIMER\_RCR 为 0，不进行重复计数。
6. 配置 ATIMER\_CCMRx 的 CCxS，设置 CCx 通道为输入，并更根据需求映射。
7. 配置 ATIMER\_CCER 的 CCxP 和 ATIMER\_CCER 的 CCxNP，设置捕捉极性。
8. 配置 ATIMER\_CCMRx 的 ICxF，设置采样频率和滤波长度，一般设置为 0 即可。
9. 配置 ATIMER\_CCMRx 的 ICxPSC，设置输入捕捉预分频。
10. 配置 ATIMER\_CCER 的 CCxE 为 1，使能捕捉功能。
11. 配置 ATIMER\_EGR 的 UG 为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
12. 配置 ATIMER\_CR1 的 CEN 为 1，使能计数器。
13. 配置 ATIMER\_DIER 的 CCxIE 为 1，允许通道 x 捕捉中断。

## 12.6.4 互补输出和死区插入

64MHz 的周期时间  $t_{DTS} = 15.63 \text{ ns}$

1.  $DT = (0 \sim 127) * 15.63 = 0 \sim 1985.01 \text{ ns}$ ,  $DTG[7:5]=0xx$ :  $DT=DTG[7:0] * t_{DTS}$
2.  $DT = (64 + (0 \sim 63)) * 2 * 15.63 = 2000.64 \sim 3970.02 \text{ ns}$ ,  $DTG[7:5]=10x$ :  $DT=(64+DTG[5:0]) * 2 * t_{DTS}$
3.  $DT = (32 + (0 \sim 31)) * 8 * 15.63 = 4001.28 \sim 7877.52 \text{ ns}$ ,  $DTG[7:5]=110$ :  $DT=(32+DTG[4:0]) * 8 * t_{DTS}$
4.  $DT = (32 + (0 \sim 31)) * 16 * 15.63 = 8002.56 \sim 15755.04 \text{ ns}$ ,  
 $DTG[7:5]=111$ :  $DT=(32+DTG[4:0]) * 16 * t_{DTS}$

在初始化 PWM 模式前，补充以下配置：

1. 配置 ATIMER\_BDTR 的 DTG，设置互补输出的死区时间长度。
2. 配置 ATIMER\_CCER 的 CCxNE 为 1，设置预分频值。

## 12.6.5 刹车功能

在初始化 PWM 模式前，补充以下配置：

1. 配置 ATIMER\_BDTR 的 OSSR，设置运行状态下的输出关闭状态。
2. 配置 ATIMER\_BDTR 的 OSSI，设置空闲状态下的输出关闭状态。
3. 配置 ATIMER\_BDTR 的 BKP，设置刹车极性。
4. 配置 ATIMER\_CCER 的 CCxP，设置 OCx 的输出极性。
5. 配置 ATIMER\_CCER 的 CCxNP，设置 OCxN 的输出极性。
6. 配置 ATIMER\_CR2 的 OISx，设置 OCx 的空闲输出状态。
7. 配置 ATIMER\_CR2 的 OISxN，设置 OCxN 的空闲输出状态。
8. 配置 ATIMER\_BDTR 的 AOE，设置 ATIMER\_BDTR 的 MOE 置位方式。
9. 配置 ATIMER\_BDTR 的 BKE 为 1，允许刹车输入。

## 12.6.6 编码器接口模式

1. 配置 ATIMER\_CR1 的 DIR，设置计数方向。
2. 配置 ATIMER\_CR1 的 APRE 为 1，使能 Auto-reload 预装载。
3. 配置 ATIMER\_PSC，设置预分频值。
4. 配置 ATIMER\_ARR，设置自动重载值。
5. 配置 ATIMER\_RCR 为 0，不进行重复计数。
6. 配置 ATIMER\_CCMR1 的 CC1S 为 1，设置 CC1 通道为输入，IC1 映射到 TI1。
7. 配置 ATIMER\_CCMR1 的 CC2S 为 1，设置 CC2 通道为输入，IC2 映射到 TI2。
8. 配置 ATIMER\_CCER 的 CC1P 和 ATIMER\_CCER 的 CC1NP，设置捕捉极性。
9. 配置 ATIMER\_CCER 的 CC2P 和 ATIMER\_CCER 的 CC2NP，设置捕捉极性。
10. 配置 ATIMER\_CCMR1 的 IC1F，设置采样频率和滤波长度，一般设置为 0 即可。
11. 配置 ATIMER\_CCMR1 的 IC2F，设置采样频率和滤波长度，一般设置为 0 即可。
12. 配置 ATIMER\_SMCR 的 SMS，设置 Encoder 模式 1/2/3。
13. 配置 ATIMER\_CCER 的 CC1E 为 1，使能通道 1 捕捉功能。
14. 配置 ATIMER\_CCER 的 CC2E 为 1，使能通道 2 捕捉功能。
15. 配置 ATIMER\_EGR 的 UG 为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
16. 配置 ATIMER\_CR1 的 CEN 为 1，使能计数器。
17. 配置 ATIMER\_DIER 的 CC1IE 为 1，允许通道 1 捕捉中断。

## 12.6.7 DMA 模式

输出比较模式下，SRAM 中的值通过 DMA 传输到 ATIMER 的比较寄存器：

1. 在 PWM 模式中软件置位 UG 和使能计数器前，补充以下配置。
2. 配置 ATIMER\_DCR 的 DBL，设置 DMA Burst 长度。

3. 配置 ATIMER\_DCR 的 DBA，设置 DMA 基地址，一般该处的基地址选择相应比较通道对应的捕捉/比较寄存器。
4. 配置 ATIMER\_DIER 的 CCxDE 为 1，允许 CCx DMA 请求。
5. 配置 ATIMER\_CR2 的 CCDS 为 0，发生 CCx 事件时产生 CCxDMA 请求。
6. DMA 控制器配置详细请看 [DMA 章节](#)。
7. 开启 DMA 传输后，当计数器计数值等于比较值时，DMA 将 SRAM 中的值传到基地址。

# 13 通用定时器 GTIMER0

## 13.1 概述

通用定时/计数器 (GTIMER0) 包含一个 32 位自动重载计数器及一个可编程预分频器，有自己独立的中断，可以有多种用途，包括测量输入信号的脉冲宽度 (输入捕获)，产生输出波形。计数器可以向上，向下，向上/下三种计数方向，且计数值可以随时由软件读取。

## 13.2 主要特性

- 32 位向上、向下、向上/下计数自动重载计数器
- 16 位可编程预分频器，支持实时调整计数时钟分频
- 灵活的计数时钟源选择
- 通道可用于输入捕获、输出比较、PWM (边沿或中央对齐模式)、单脉冲输出
- 支持定时器间的级联
- 中断在以下几种情况产生：
  - Update 中断：计数器向上/向下溢出
  - 输入捕获
  - 输出比较
- 支持增量正交编码器和霍尔传感器

## 13.3 通用定时器 GTIMER0 结构框图

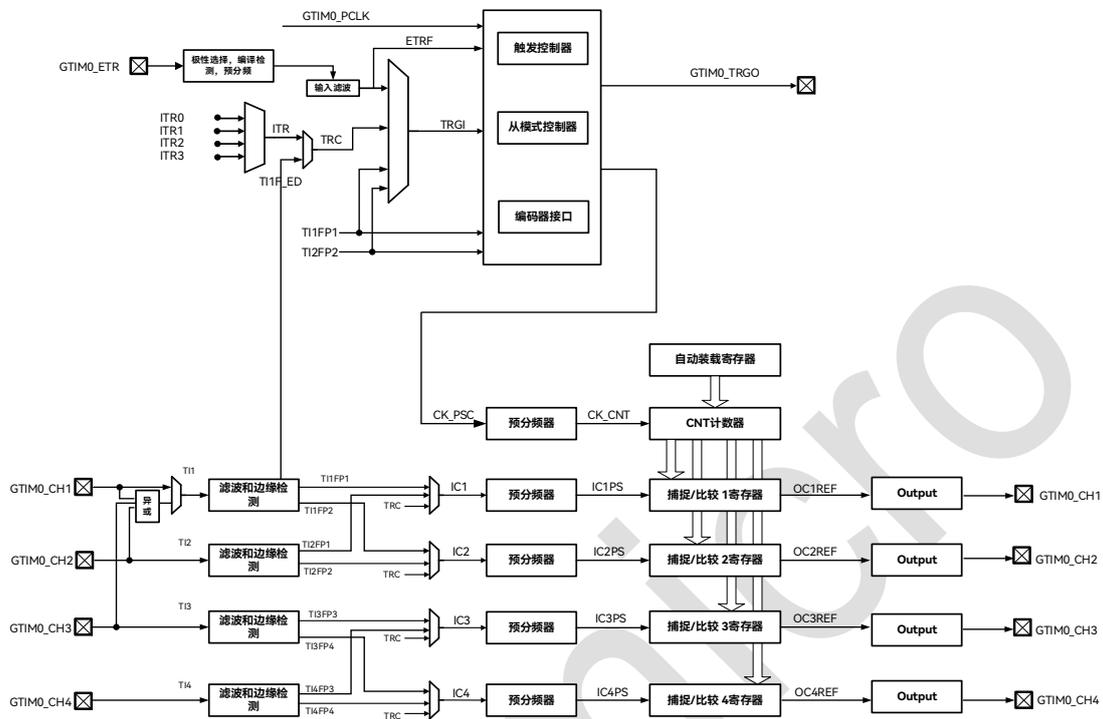


图 13-1: GTIMER 结构框图

## 13.4 功能描述

### 13.4.1 定时单元

GTIMER0 主要部分是一个 32 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

定时单元包含如下寄存器：

- 计数器寄存器 (GTIMER\_CNT)
- 预分频寄存器 (GTIMER\_PSC)
- 自动装载寄存器 (GTIMER\_ARR)

GTIMER\_ARR 包含预装载功能, 根据在 GTIMER\_CR1 寄存器中的自动装载预装载使能位 (ARPE)

的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件（向下计数时的下溢条件）并当 GTIMER\_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。软件也可以通过寄存器（GTIMER\_EGR[0]写 1）操作主动触发 GTIMER\_ARR 更新。

计数器工作时钟由 GTIMER\_PSC 产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，GTIMER\_CNT 才开始计数。

GTIMER\_PSC 是一个同步预分频器，预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是一个 16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下次更新事件到来时被采用。

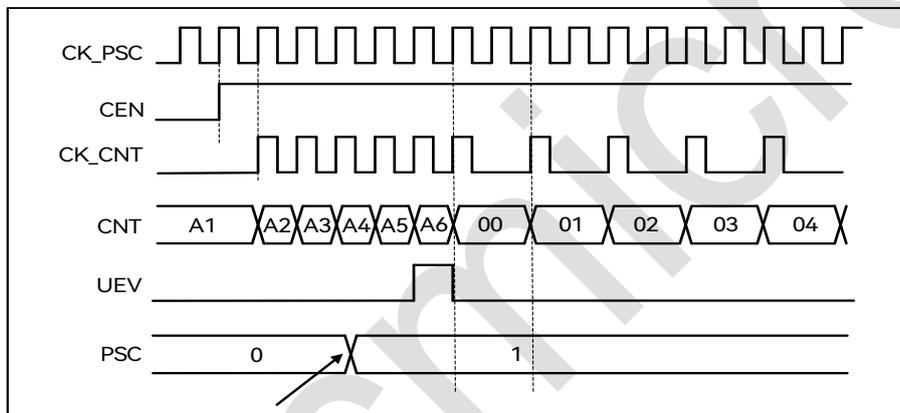


图 13-2: GTIMER 预分频从 1 变为 2 的波形图

## 13.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

### 13.4.2.1 向上计数

在向上计数模式中，计数器从 0 计数到自动加载值（GTIMER\_ARR 计数器的值），然后重新从 0 开始计数并且产生一个计数器溢出事件。

在 GTIMER\_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。设置 GTIMER\_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清 0 之前，将不产生更新事件。但是在

应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。

如果设置了 GTIMER\_CR1 寄存器中的 URS 位，设置 UG 位将产生一个更新事件 UEV，但硬件不产生 UIF 标志。

当发生一个更新事件时，以下寄存器都被更新，硬件同时置位 UIF 标志位：

- GTIMER\_ARR 影子寄存器被更新为 GTIMER\_ARR 内容。
- GTIMER\_PSC 影子寄存器被更新为 GTIMER\_PSC 内容。

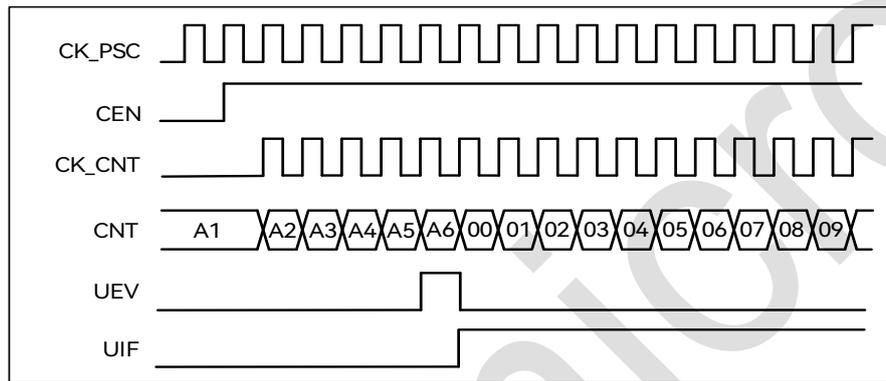


图 13-3: 向上计数波形图

### 13.4.2.2 向下计数

在向下计数模式中，计数器从 GTIMER\_ARR 开始倒数到 0，然后重新从 GTIMER\_ARR 开始计数并且产生一个计数器溢出事件。

在 GTIMER\_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。设置 GTIMER\_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清 0 之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。

如果设置了 GTIMER\_CR1 寄存器中的 URS 位，设置 UG 位将产生一个更新事件 UEV，但硬件不产生 UIF 标志。

当发生一个更新事件时，以下寄存器都被更新，硬件同时置位 UIF 标志位：

- GTIMER\_ARR 影子寄存器被更新为 GTIMER\_ARR 内容。
- GTIMER\_PSC 影子寄存器被更新为 GTIMER\_PSC 内容。

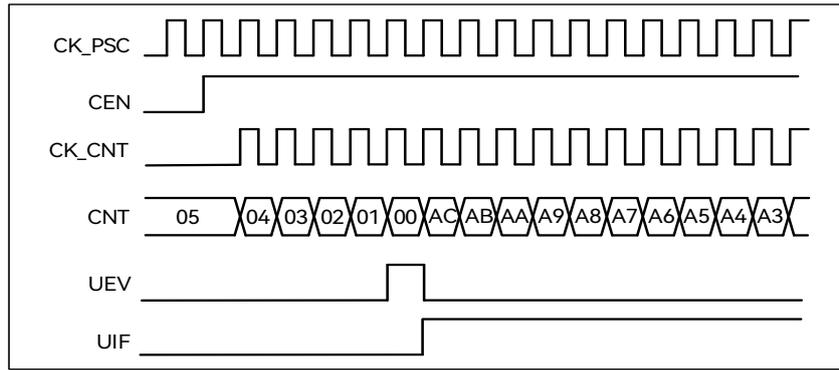


图 13-4: 向下计数图

### 13.4.2.3 中心对齐计数

在中心对齐模式下,计数器从 0 开始向上计数,到 ARR-1 产生上溢出事件,然后从 GTIMER\_ARR 开始向下计数到 1,产生下溢出事件,再从 0 重新开始向上计数。

GTIMER\_CR1[6:5]寄存器用于使能中心对齐模式,并选择中心对齐模式下的输出比较工作方式:

- CMS = 00: 边沿对齐模式
- 01: 中央对齐模式 1, 输出比较中断标志仅在计数器向下计数的过程中置位。
- 10: 中央对齐模式 2, 输出比较中断标志仅在计数器向上计数的过程中置位。
- 11: 中央对齐模式 3, 输出比较中断标志在计数器向上向下计数的过程中都会置位。
- 中心对齐模式下, DIR 寄存器无法由软件改写,而是随着计数方向变化硬件自动更新,表示当前计数方向。计数器在上溢和下溢的事件上都会更新如下寄存器:
  - GTIMER\_ARR 影子寄存器被更新为 GTIMER\_ARR 内容。
  - GTIMER\_PSC 影子寄存器被更新为 GTIMER\_PSC 内容。

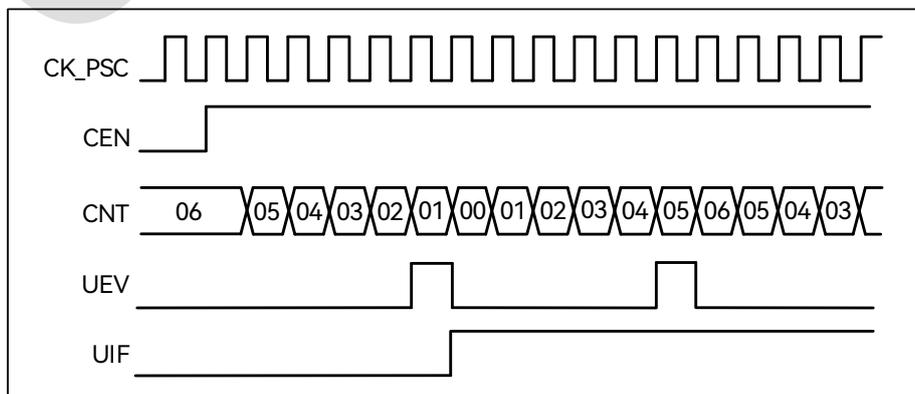


图 13-5: 中心对齐计数器时序图

### 13.4.3 计数器工作时钟

计数器可以使用如下时钟工作：

- 内部时钟 APBCLK (CK\_INT)
- 外部引脚输入时钟 GTIMER\_CHx (x=1,2)
- 外部引脚触发输入 (GTIMER\_ETR)
- 内部触发 (ITRx)

#### 13.4.3.1 APBCLK

当选择 APBCLK 作为时钟源时，禁止 GTIMER 设置为从机模式 (SMS=000)，CEN、DIR、UG 等寄存器位都是软件控制。软件操作 UG 寄存器后，分频器，计数器值将被重新初始化。

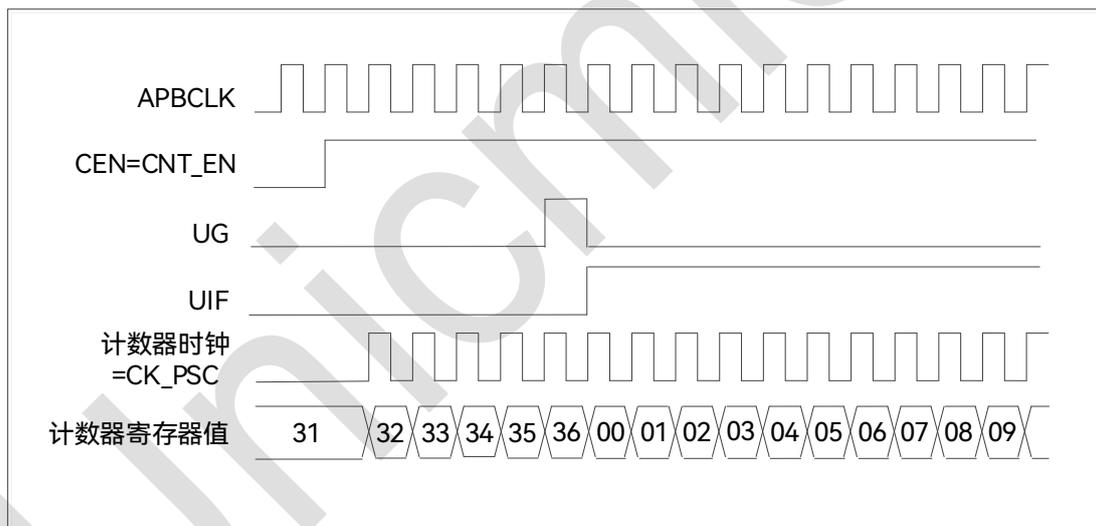


图 13-6: APBCLK 时钟源模式，时钟分频因子为 1

#### 13.4.3.2 GTIMER\_CHx (x=1,2) 引脚输入——外部时钟模式 1

配置 SMS=111，此模式下直接使用外部引脚 GTIMER\_CHx 输入信号作为计数时钟，计数边沿可以选择为上升或下降沿。

此模式下只有通道 1 和 2 的输入可以用做时钟输入，以 GTIMER\_CH1 为例，配置流程如下：

1. 在 IO 模块种配置相应管脚为 GTIMER\_CH2 功能。

2. 配置 GTIMER->CCMR1[9:8]=01，配置通道 2 检测 TI2 输入的上升沿。
3. 配置 GTIMER->CCMR1[15:12]为需要滤波的值，选择输入滤波器带宽（如果不需要滤波器，保持 CCMR1[15:12]=0000）。
4. 配置 GTIMER->CCER[5]=0，选定 TI2 上升沿极性。
5. 配置 GTIMER->SMCR[2:0]=111，选择定时器外部时钟模式 1。
6. 配置 GTIMER->SMCR[6:4]=110，选定 TI2 作为触发输入源。
7. 打开通道使能，配置 GTIMER->CCER[0]=1。
8. 使能计数器，配置 GTIMER->CR1[0]=1。

### 13.4.3.3 GTIMER\_ETR 引脚输入——外部时钟模式 2

此模式下使用 GTIMER\_ETR 管脚输入信号的上升沿或下降沿来计数。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的同步电路。

在使用外部时钟模式 2 时，仍可以将 GTIMER 配置为 slave 模式：比如使用 GTIMER\_ETR 输入计数，同时使用另一个 GTIMER 的 TRGO 作为触发信号，当触发事件到来时，复位计数器重新开始计数。

### 13.4.4 内部触发信号 (ITRx)

GTIMER 支持 4 个 ITR 输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需将 TS 配置为 000~011 用于选择 ITR0~ITR3，并将 CCxS 配置为 11，即将 TRC 选为捕捉信号。

每个 ITR 输入支持 4 个内部信号扩展，由 TS 寄存器配置。输入信号源参考下表：

表 13-1: GTIMER0/1/2 内部信号扩展表

Slave	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
GTIMER0	RCL32k	ATIMER_TRGO	COMP0_IN	COMP1_IN
GTIMER1	RCL32k	GTIMER0_TRGO	COMP0_IN	COMP1_IN
GTIMER2	RCL32k	GTIMER1_TRGO	COMP0_IN	COMP1_IN

### 13.4.5 捕捉/比较通道

通用定时器 1 包含 4 个捕捉/比较通道，每个通道由一个捕捉/比较寄存器 GTIMER\_CCR（包含影子寄存器）、输入电路（数字滤波、多路复用和预分频器）和输出电路（比较器和输出控制）组成。

#### 13.4.5.1 捕捉通道

输入电路对相应的  $TIx$  ( $x=1\dots4$ ) 进行采样并产生滤波后的信号  $TIxF$  ( $x=1\dots4$ )，然后经过边沿检测和极性选择产生  $TIxFPx$  ( $x=1\dots4$ ) 信号，此信号可作为计数触发源或者捕捉源，并通过预分频器后进入捕捉通道。

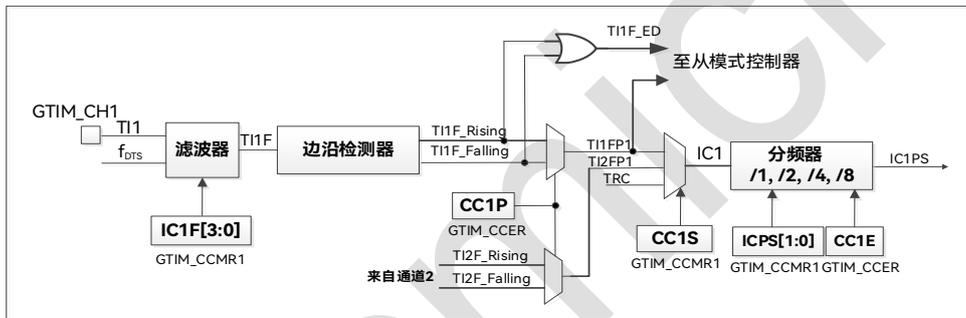


图 13-7: 捕捉/比较通道 (通道 1 输入部分)

#### 13.4.5.2 比较通道

输出级电路产生一个中间信号  $OCxREF$  (高有效) 作为基准，给最终输出电路作为参考输入。

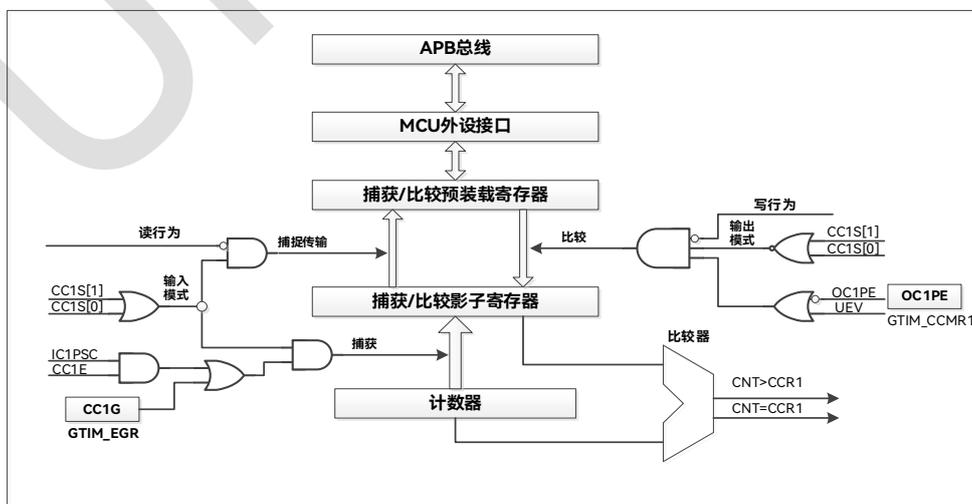


图 13-8: 捕捉/比较通道 1 的主电路

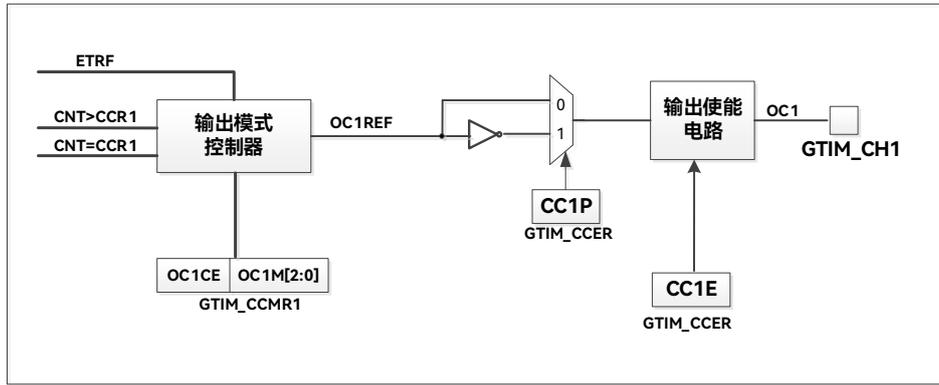


图 13-9: 捕获/比较通道的输出部分

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

### 13.4.6 输入捕捉模式

在输入捕获模式下，当检测到 ICx 信号上出现相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（GTIMER\_CCRx）中。当发生捕获事件时，相应的 CCxIF 标志（GTIMER\_SR 寄存器）被置 1，如果使能了中断或者 DMA，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（GTIMER\_SR 寄存器）被置 1。写 CCxIF 或者读取存储在 GTIMER\_CCRx 寄存器中的捕获数据都可清除 CCxIF。写 CCxOF=1 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值，步骤如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 选择输入通道，配置 GTIMER->CCMR1.CC1S=01, IC1 映射到 TI1。
3. 选择 TI1 通道的计数有效沿，配置 GTIMER\_CCER.CC1P=0，选择上升沿计数。
4. 配置输入滤波时间，配置 GTIMER\_CCMR1.IC1F。
5. 配置输入预分频器，配置 GTIMER\_CCMR1.IC1PS。
6. 设置 GTIMER\_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
7. 根据需要打开相应中断使能或 DMA 请求使能。
8. 打开通道使能，配置 GTIMER\_CCER[0]=1。

### 13.4.7 PWM 输入模式

该模式是输入捕获模式的一个特例，通过两个通道配合可以实现 PWM 信号脉宽的输入捕捉。

实现 PWM 输入捕获功能，配置流程如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 选择 GTIMER\_CCR1 的有效输入：置 GTIMER\_CCMR1 寄存器的 CC1S=01（选中 TI1）。
3. 选择 TI1FP1 的有效极性：置 CC1P=0（上升沿有效）。
4. 选择 GTIMER\_CCR2 的有效输入：置 GTIMER\_CCMR1 寄存器的 CC2S=10（选中 TI1）。
5. 选择 TI1FP2 的有效极性：置 CC2P=1（下降沿有效）。
6. 选择有效的触发输入信号：置 GTIMER\_SMCR 寄存器中的 TS=101。
7. 配置从模式控制器为复位模式：置 GTIMER\_SMCR 中的 SMS=100。
8. 使能捕获：置 GTIMER\_CCER 寄存器中 CC1E=1 且 CC2E=1。

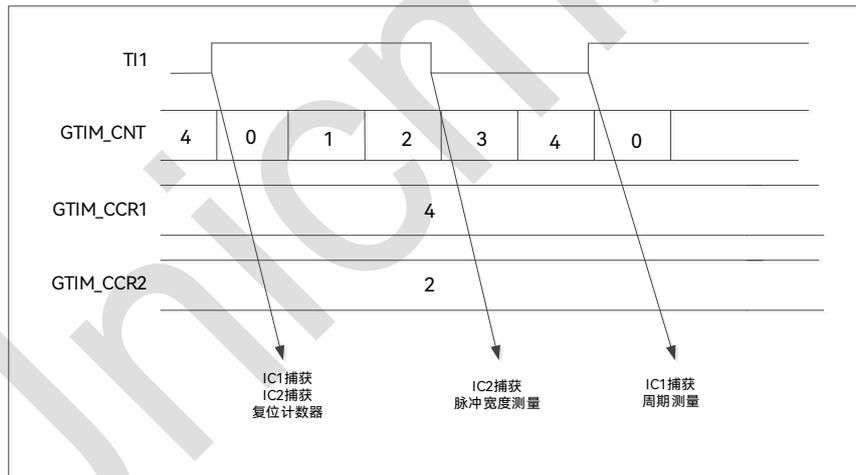


图 13-10: PWM 输入捕获

### 13.4.8 软件强制输出

在强制输出模式下，输出比较信号（OCxREF 和相应的 OCx/OCxN）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 GTIMER\_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号为有效状态。这样 OCxREF 被强置为高电平，同时 OCx 得到 CCxP 极性相反的信号。例如：CCxP=0（OCx 高电平有

效), 则 OCx 被强置为高电平。置 GTIMER\_CCMRx 寄存器中的 OCxM=100, 可强置 OCxREF 信号为低。

### 13.4.9 输出比较模式

输出比较模式下, 当 GTIMER\_CCR 与 GTIMER\_CNT 相等, OCxREF 可以被置位成有效、无效、或电平翻转。同时, 中断标志也会置位, DMA 请求可以发送。输出比较也可以被用于输出一个特定宽度的脉冲信号 (单次输出)。

使用步骤:

1. 选择计数时钟 (内部、外部、预分频等)。
2. 设置 GTIMER\_ARR 和 GTIMER\_CCR 寄存器的值。
3. 根据需要相应设置中断使能和 DMA 使能。
4. 选择输出模式, 例如:
  - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚, 设置 OCxM=011。
  - 置 OCxPE = 0 禁用预装载寄存器。
  - 置 CCxP = 0 选择极性为高电平有效。
  - 置 CCxE = 1 使能输出。

设置计数器使能 CEN=1。

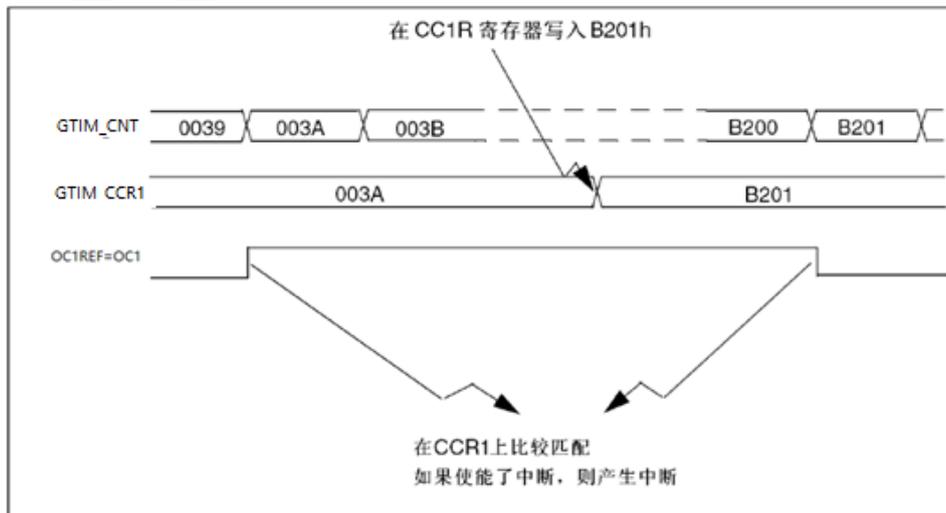


图 13-11: 输出比较模式, 翻转 OC1

在不使能 preload 的情况下, 软件可以随时改写 GTIMER\_CCR 寄存器实现对输出波形的实时控制。如果使能了 preload, 则 GTIMER\_CCR 的 shadow 寄存器仅在下一次 update event 发生时更新为 preload 寄存器的内容。

### 13.4.10 PWM 输出

PWM 模式可以输出脉宽调制信号, 其周期由 GTIMER\_ARR 寄存器决定, 占空比由 GTIMER\_CCR 寄存器决定。输出信号的极性可以由 CCxP 寄存器配置。PWM 模式工作中, GTIMER\_CNT 和 GTIMER\_CCR 实时比较。由于计数器支持边缘对齐和中央对齐计数模式, PWM 输出也支持边缘对齐和中央对齐模式。

#### 13.4.10.1 PWM 边缘对齐模式

在向上计数的情况下, 配置为 PWM 模式 1 时, OCxREF 信号在  $GTIMER\_CNT < GTIMER\_CCR$  时为高电平, 否则为低电平。如果 GTIMER\_CCR 值大于 GTIMER\_ARR 值, 则 OCxREF 被固定为 1; 如果 CCR 为 0 则 OCxREF 被固定为 0。

在向下计数时, OCxREF 电平高低定义与向上计数时相同。

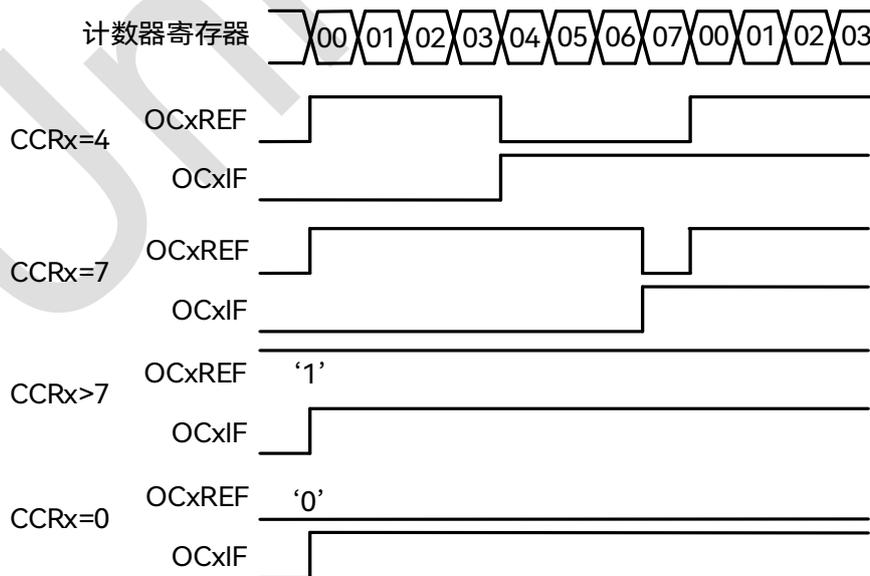


图 13-12: 边沿对齐的 PWM 波形 (ARR=7)

### 13.4.10.2 PWM 中央对齐模式

OCxREF 电平定义与边缘对齐模式相同。

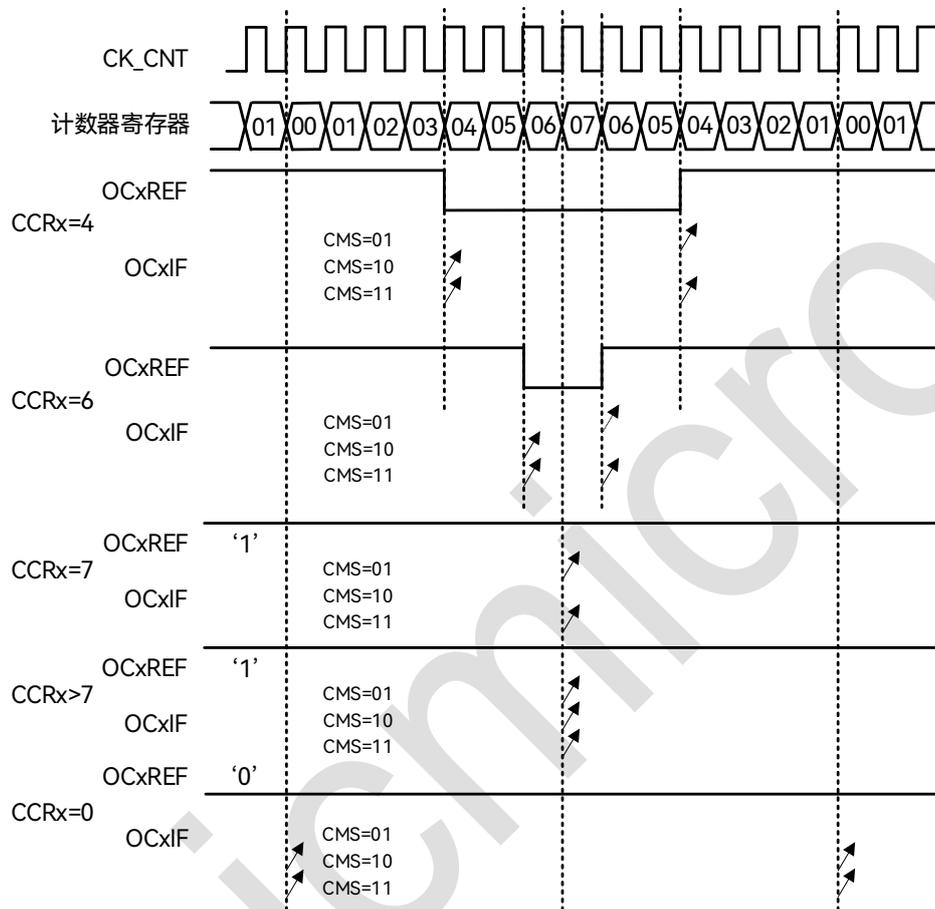


图 13-13: 中央对齐的 PWM 波形 (APR=7)

当启动中央对齐计数时,一开始的计数方向是由 DIR 寄存器决定的;随后在计数过程中,DIR 寄存器的状态由硬件直接控制。安全起见,建议用户程序在启动计数器之前,通过 UG 寄存器做一次 update,并且在计数过程中不要改写计数器。

### 13.4.11 输出比较快速使能模式

打开快速使能模式 (CCMR1[2, 10]、CCMR2[2, 10]) 后 GTIMER 的比较输出结果受 TRGI 信号影响。TRGI 信号选择是通过寄存器 SMCR[4:6]来设置的。以通道 1 为例:

**假设 CC1P 是 0 的情况:**

- 当 TRGI 信号到来后,如果采用 PWM 模式 1,不管 PWM 本来是什么电平,GTIMER\_CH1 都被

强制变为 0 电平，等到计数到达 GTIMER\_ARR 后恢复为正常。

- 当 TRGI 信号到来后，如果采用 PWM 模式 2，不管 PWM 本来是什么电平，GTIMER\_CH1 都被强制变为 1 电平，等到计数到达 GTIMER\_ARR 后恢复为正常。

**假设 CC1P 是 1 的情况：**

- 当 TRGI 信号到来后，如果采用 PWM 模式 1，不管 PWM 本来是什么电平，GTIMER\_CH1 都被强制变为 1 电平，等到计数到达 GTIMER\_ARR 后恢复为正常。
- 当 TRGI 信号到来后，如果采用 PWM 模式 2，不管 PWM 本来是什么电平，GTIMER\_CH1 都被强制变为 0 电平，等到计数到达 GTIMER\_ARR 后恢复为正常。

### 13.4.12 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下次 update event 到来时，计数器会自动停止。只有当 GTIMER\_CCR 和 GTIMER\_CNT 初值不同时，脉冲才有可能正确输出。在向上计数时，要求  $GTIMER\_CNT < GTIMER\_CCR \leq GTIMER\_ARR$ ，在向下计数时，要求  $GTIMER\_CNT > GTIMER\_CCR$ 。

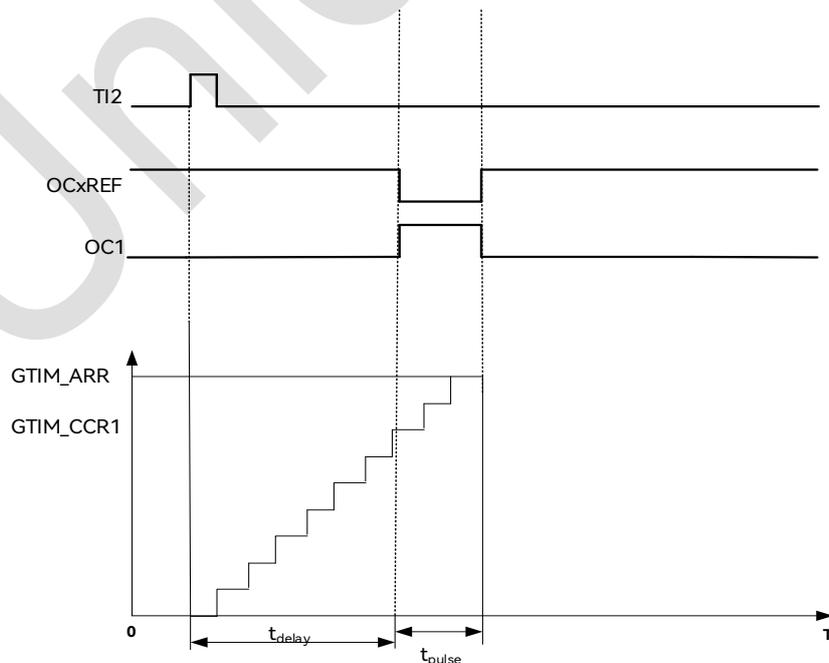


图 13-14: 单脉冲模式的例子

所示是以 TI2 输入为计数器触发信号，计数值等 GTIMER\_CCR 后 OCxREF 输出低电平，计数到 GTIMER\_ARR 后 OCxREF 回到高电平，并且计数器回 0，停止计数。

#### 实现上述功能 TI2 作为输入触发的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH2 功能。
- 关闭通道使能，配置 GTIMER->CCER.CC2E=0，确保之后的配置成功。
- 选择输入通道，配置 GTIMER->CCMR1.CC2S=01。
- 选择计数有效沿，配置 GTIMER->CCER.CC2P=0。
- 选择触发输入信号，配置 GTIMER->SMCR.TS[2:0]=110，TI2FP2 作为 TRGI。
- 设定从模式控制器为触发模式，配置 GTIMER->SMCR.SMS[2:0]=110，TI2FP2 用来启动计数器。
- 打开通道使能，配置 GTIMER->CCER.CC2E=1。

#### 实现上述功能 OC1 作为输出的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
- 关闭通道使能，配置 GTIMER->CCER.CC1E=0，确保之后的配置成功。
- 输出通道，配置 GTIMER->CCMR1.CC1S=00。
- 选择计数有效沿，配置 GTIMER->CCMR1.OC1M=111，PWM 模式 2。
- 打开通道使能，配置 GTIMER->CCER.CC1E=1。

#### 单脉冲波形产生时基的特殊设置：

- GTIMER->CCR1 的值决定了  $t_{\text{delay}}$ 。
- GTIMER->ARR 和 GTIMER->CCR1 的差值决定了  $t_{\text{pulse}}$  (GTIMER->ARR - GTIMER->CCR1)。
- 设置为单脉冲模式，配置 GTIMER\_CR1.OPM=1。

### 13.4.13 外部事件清除 OCxREF

OCxREF 的有效状态为高电平，通过对外部 GTIMER\_ETR 引脚施加高电平，可以直接拉低 OCxREF，直到下一次 update event。此功能仅在输出比较和 PWM 模式下有效，无法在软件 force 模式下起作用。使能此功能需要将 OCxCE 置 1。

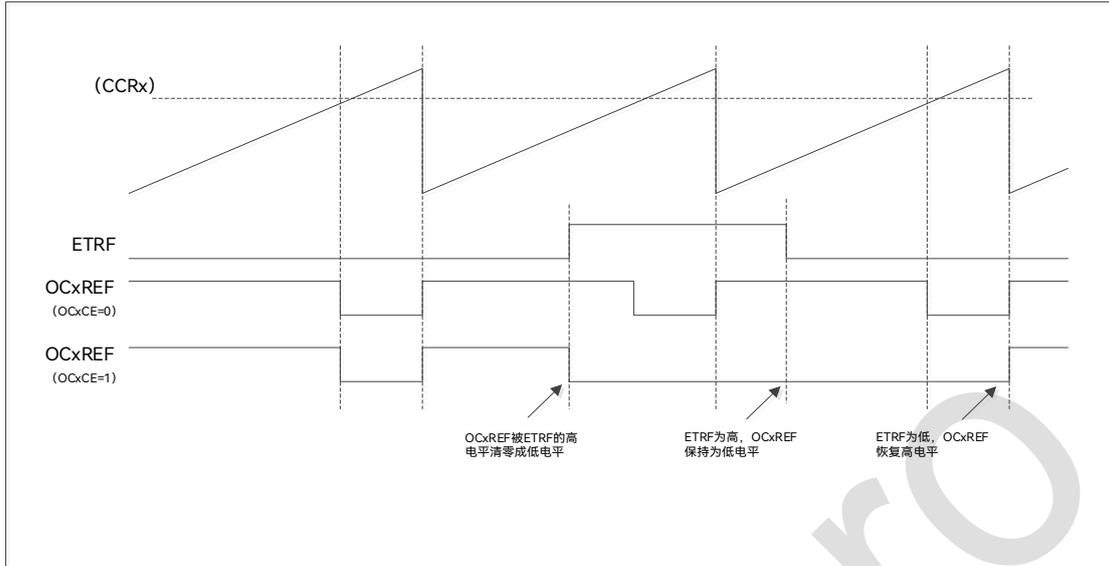


图 13-15: GTIMER\_ETRF 信号清除 GTIMER 的 OCxREF

### 13.4.14 编码器接口模式 (Encoder Interface)

编码器接口模式涉及到两个外部输入信号，GTIMER 根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

表 13-2: 编码计数方式与输入信号

计数信号	T11 计数时 T12 电平/ T12 计数时 T11 电平	T11 信号		T12 信号	
		上升沿	下降沿	上升沿	下降沿
T11 处计数	H	递减计数	递增计数	不计数	不计数
	L	递增计数	递减计数	不计数	不计数
T12 处计数	H	不计数	不计数	递增计数	递减计数
	L	不计数	不计数	递减计数	递增计数
T11 和 T12 处计数	H	递减计数	递增计数	递增计数	递减计数
	L	递增计数	递减计数	递减计数	递增计数

比如在计数器以 TI1 信号为时钟计数时, 如果 TI1 上升沿采样到 TI2 为高电平, 则计数器递减; 如果 TI1 下降沿采样到 TI2 为高电平, 则计数器递增。

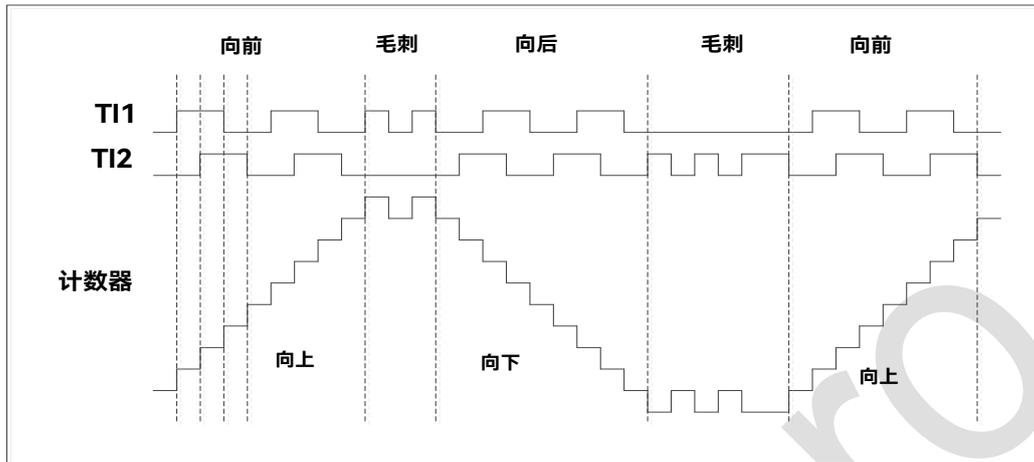


图 13-16: 编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置:

- 在 GPIO 模块中, 配置相应管脚为 GTIMER\_CH1, GTIMER\_CH2 功能。
- 关闭通道使能, 配置  $GTIMER \rightarrow CCER.CC1E=0$ ,  $GTIMER \rightarrow CCER.CC2E=0$ , 确保之后通道配置成功。
- 选择输入通道, 配置  $GTIMER \rightarrow CCMR1.CC1S=01$ ,  $GTIMER \rightarrow CCMR1.CC2S=01$ 。
- 选择计数有效沿, 配置  $GTIMER \rightarrow CCER.CC1P=0$ ,  $GTIMER \rightarrow CCER.CC2P=0$ 。
- 设定从模式控制器为编码模式 3, 配置  $GTIMER \rightarrow SMCR.SMS[2:0]=011$ 。

### 13.4.15 GTIMER 从机模式

GTIMER 作为 slave 时 (外部事件触发), 可配置为三种工作模式: 复位模式、门控模式、触发模式。

#### 13.4.15.1 复位模式

此模式下, 外部输入的事件将导致 GTIMER 内部所有 preload 寄存器重新初始化, CNT 回到 0 开始计数。计数器正常计数, 外部 TI1 输入上升沿时, 触发计数器清零, 重新开始计数。其配置流程如下:

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 关闭通道使能，配置 GTIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 GTIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 GTIMER\_CCER.CC1P=0。
5. 选择触发输入信号，配置 GTIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
6. 设定从模式控制器为复位模式，配置 GTIMER\_SMCR.SMS[2:0]=100。
7. 打开通道使能，配置 GTIMER\_CCER.CC1E=1。
8. 使能计数器，配置 GTIMER\_CR1.CEN=1。

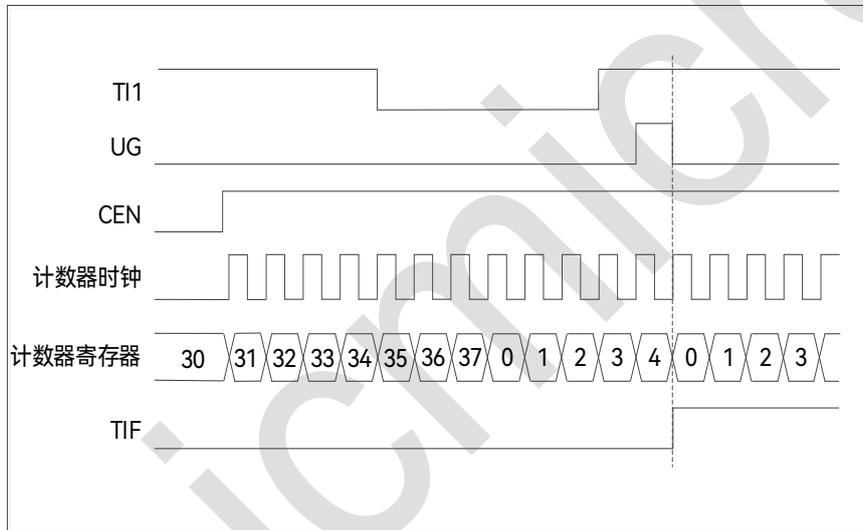


图 13-17: 复位模式下的时序

### 13.4.15.2 门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。时序配置如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 关闭通道使能，配置 GTIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 GTIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 GTIMER\_CCER.CC1P=1。
5. 选择触发输入信号，配置 GTIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。

6. 设定从模式控制器为门控模式，配置  $GTIMER\_SMCR.SMS[2:0]=101$ 。
7. 打开通道使能，配置  $GTIMER\_CCER.CC1E=1$ 。
8. 使能计数器，配置  $GTIMER\_CR1.CEN=1$ 。

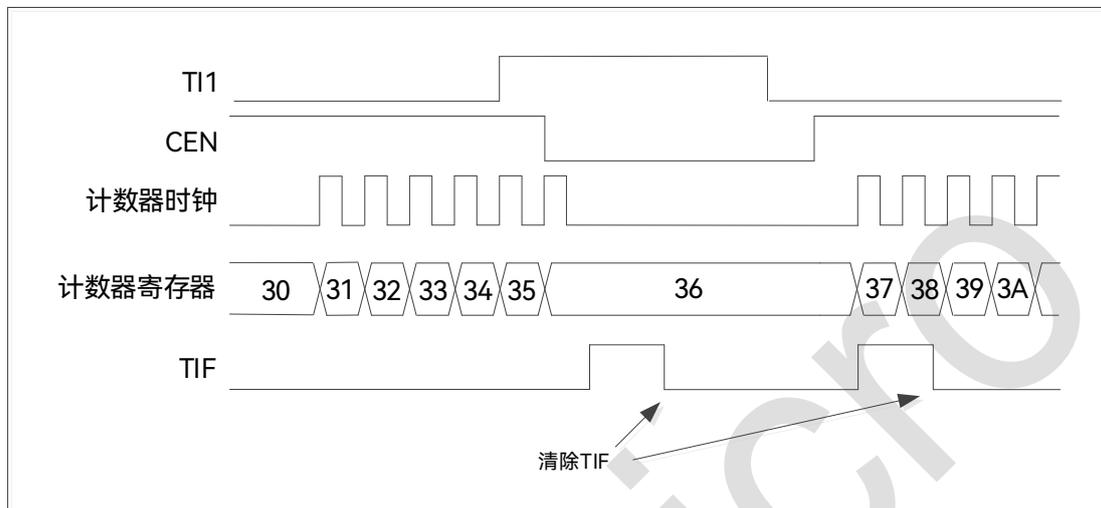


图 13-18: 门控模式下的时序图

### 13.4.15.3 触发模式

计数器在外部输入的某个事件到来后才开始计数。所示时序的流程配置如下：

1. 在 GPIO 模块中，配置相应管脚为  $GTIMER\_CH1$  功能。
2. 关闭通道使能，配置  $GTIMER \rightarrow CCER.CC1E=0$  确保之后通道配置成功。
3. 选择输入通道，配置  $GTIMER \rightarrow CCMR1.CC1S=01$ 。
4. 选择计数有效沿，配置  $GTIMER \rightarrow CCER.CC1P=0$ 。
5. 选择触发输入信号，配置  $GTIMER \rightarrow SMCR.TS[2:0]=101$ ， $TI1FP1$  作为 TRGI。
6. 设定从模式控制器为触发模式，配置  $GTIMER \rightarrow SMCR.SMS[2:0]=110$ 。
7. 打开通道使能，配置  $GTIMER \rightarrow CCER.CC1E=1$ 。

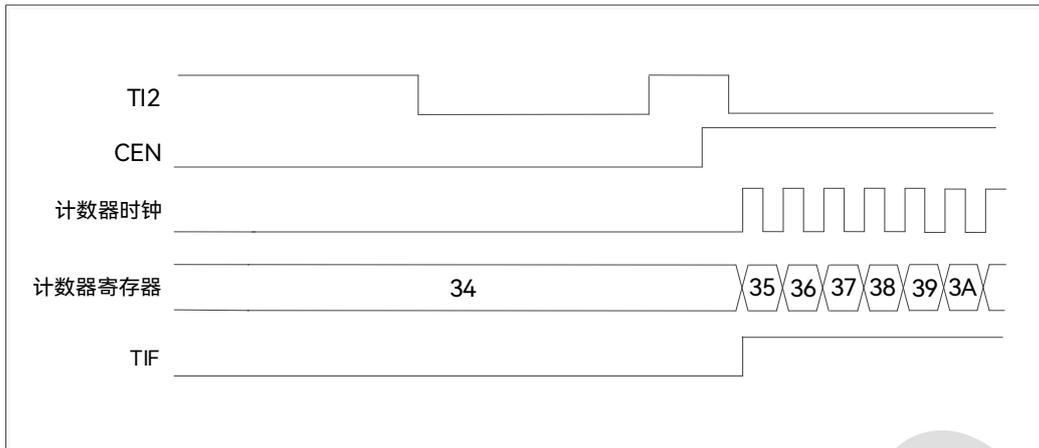


图 13-19: 触发器模式下的时序

### 13.4.16 定时器同步

定时器之间可以通过触发事件级联起来，实现定时器同步或级联。通用定时器可以通过另外 2 个通用定时器或高级定时器的输出来触发计数。而通用定时器的触发输出则可以接到其他定时器的内部触发输入上。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

### 13.4.17 DMA Burst

GTIMER 支持 DMA 和 DMA-Burst 访问, 可以配置 GTIMER 在特定事件发生时触发 DMA 请求, 可以将 CCR 中的捕捉结果写入 RAM, 或者从 RAM 中将一个或多个寄存器内容写入 GTIMER 的 preload 寄存器中。

DMA-Burst 支持一个事件触发连续多次 DMA 请求, 主要作用是在事件发生后连续更新多个寄存器的内容, 因此可以实现动态实时调整输出波形等功能。

DMA 控制器需将外设目标地址指向一个虚拟寄存器 GTIMER\_DMAR。在特定的定时器事件发生时, GTIMER 会连续发射多个 DMA 请求。每个 DMA 对 GTIMER\_DMAR 的写操作都会被 GTIMER 重新定向到实际的功能寄存器上。

DBL 寄存器用于设置 DMA burst 长度, DBA 寄存器用于设置 DMA 访问 GTIMER 内部的基地址 (相对于 GTIMER\_CR 的 offset)。

### 13.4.18 输入异或功能

通道 1~3 的输入信号可以被异或起来之后，接入到通道 1 的滤波和边沿电路输入，用于通道 1 的输入捕捉或者触发。GTIMER\_CR2 寄存器的 TI1S 位用于选择通道 1 的输入是否来自于三个通道输入的异或。

### 13.4.19 Debug 模式

当 Cortex-M0 进入 debug 模式后，定时器可以停止或继续工作，其行为由 STOP\_EN 寄存器定义。

## 13.5 寄存器描述

### 13.5.1 GTIMER 寄存器分布

GTIMER0 寄存器基地址：0x4000C000

表 13-3: GTIMER 寄存器列表

偏移地址	寄存器	描述
0x00	GTIMER_CR1	GTIM 控制寄存器 1
0x04	GTIMER_CR2	GTIM 控制寄存器 2
0x08	GTIMER_SMCR	GTIM 从机模式控制寄存器
0x0C	GTIMER_DIER	GTIM 中断和 DMA 使能寄存器
0x10	GTIMER_SR	GTIM 状态寄存器
0x14	GTIMER_EGR	GTIM 事件产生寄存器
0x18	GTIMER_CCMR1	GTIM 捕捉/比较模式寄存器 1
0x1C	GTIMER_CCMR2	GTIM 捕捉/比较模式寄存器 2
0x20	GTIMER_CCER	GTIM 捕捉/比较使能寄存器
0x24	GTIMER_CNT	GTIM 计数寄存器
0x28	GTIMER_PSC	GTIM 预分频寄存器
0x2C	GTIMER_ARR	GTIM 自动重载寄存器
0x34	GTIMER_CCR1	GTIM 捕捉/比较寄存器 1
0x38	GTIMER_CCR2	GTIM 捕捉/比较寄存器 2
0x3C	GTIMER_CCR3	GTIM 捕捉/比较寄存器 3
0x40	GTIMER_CCR4	GTIM 捕捉/比较寄存器 4

偏移地址	寄存器	描述
0x48	GTIMER_DCR	GTIM DMA 控制寄存器
0x4C	GTIMER_DMAR	GTIM DMA 访问寄存器

### 13.5.2 控制寄存器 1 GTIMER\_CR1 (偏移: 00h)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留
10	STOP_EN	RW	0x1	断点使能位, 在软件设置断点后, 禁止计数器计数: 0: 计数器正常工作 1: 计数器停止计数
9:8	CKD	RW	0x0	数字滤波时钟频率分频寄存器 (相对 CK_INT 的分频比): 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2*t_{CK\_INT}$ 10: $t_{DTS}=4*t_{CK\_INT}$ 11: RFU, 禁止使用
7	ARPE	RW	0x0	Auto-reload 预装载使能: 0: ARR 寄存器不使能 preload 1: ARR 寄存器使能 preload
6:5	CMS	RW	0x0	计数器对齐模式选择: 00: 边沿对齐模式 01: 中央对齐模式 1, 输出比较中断标志仅在计数器向下计数的过程中置位 10: 中央对齐模式 2, 输出比较中断标志仅在计数器向上计数的过程中置位 11: 中央对齐模式 3, 输出比较中断标志在计数器向上向下计数的过程中都会置位
4	DIR	RW	0x0	计数方向寄存器: 0: 向上计数 1: 向下计数 注: 当定时器配置为中央计数模式或编码器模式时, 此寄存器只读。
3	OPM	RW	0x0	单脉冲输出模式: 0: Update Event 发生时计数器不停止 1: Update Event 发生时计数器停止 (自动清零 CEN)

比特	名称	属性	复位值	描述
2	URS	RW	0x0	更新请求选择: 0: 以下事件能够产生 update 中断或 DMA 请求 <ul style="list-style-type: none"> <li>● 计数器上溢出或下溢出</li> <li>● 软件置位 UG 寄存器</li> <li>● 从机控制器产生 update</li> </ul> 1: 仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	UDIS	RW	0x0	禁止 update: 0: 使能 update 事件; 以下事件发生时产生 update 事件 <ul style="list-style-type: none"> <li>● 计数器上溢出或下溢出</li> <li>● 软件置位 UG 寄存器</li> <li>● 从机控制器产生 update</li> </ul> 1: 禁止 update 事件, 不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	RW	0x0	计数器使能: 0: 计数器关闭 1: 计数器使能 注: 外部触发模式可以自动置位 CEN。

### 13.5.3 控制寄存器 2 GTIMER\_CR2 (偏移: 04h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11	BPS4	RW	0x0	TI4边沿检测通道选择: 0: TRC通道 1: TRC映射到TI4边沿检测 (TI4F_ED)
10	BPS3	RW	0x0	TI3边沿检测通道选择: 0: TRC通道 1: TRC映射到TI3边沿检测 (TI3F_ED)
9	BPS2	RW	0x0	TI2边沿检测通道选择: 0: TRC通道 1: TRC映射到TI2边沿检测 (TI2F_ED)
8	BPS1	RW	0x0	TI1边沿检测通道选择: 0: TRC通道 1: TRC映射到TI1边沿检测 (TI1F_ED)

比特	名称	属性	复位值	描述
7	TI1S	RW	0x0	GTIMER 输入 TI1 选择: 0: GTIMER_CH1 引脚连接到 TI1 输入 1: GTIMER_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	MMS	RW	0x0	主机模式选择, 用于配置主机模式下向从机发送的同步触发信号 (TRGO) 源: 000: GTIMER_EGR 的 UG 寄存器被用作 TRGO 001: 计数器使能信号 CNT_EN 被用作 TRGO, 可用于同时启动多个定时器 010: UE (update event) 信号被用作 TRGO 011: 比较脉冲, 如果 CC1IF 标志将要置位, TRGO 输出一个正脉冲 100: OC1REF 用作 TRGO 101: OC2REF 用作 TRGO 110: OC3REF 用作 TRGO 111: OC4REF 用作 TRGO 注: 从机定时器或 ADC 必须事先使能工作时钟, 才能接收主机定时器发送的 TRGO。
3	CCDS	RW	0x0	捕捉/比较 DMA 选择: 0: 捕捉/比较事件发生时发送 DMA 请求 1: Update Event 发生时发送 DMA 请求
2:0	RSV	-	-	保留

### 13.5.4 从机模式控制寄存器 GTIMER\_SMCR (偏移: 08h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	ETP	RW	0x0	外部触发信号极性配置: 0: 高电平或上升沿有效 1: 低电平或下降沿有效
14	ECE	RW	0x0	外部时钟使能: 0: 关闭外部时钟模式 2 1: 使能外部时钟模式 2, 计数器时钟为 ETRF 有效沿
13:12	ETPS	RW	0x0	外部触发信号预分频寄存器: 外部触发信号 ETRP 的频率最多只能是 GTIMER 工作时钟的 1/4, 当输入信号频率较高时, 可以使用预分频。 00: 不分频

比特	名称	属性	复位值	描述
				01: 2分频 10: 4分频 11: 8分频
11:8	ETF	RW	0x0	外部触发信号滤波时钟和长度选择: 0000: 无滤波 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 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$ , N=6 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$ , N=8
7	MSM	RW	0x0	主/从模式选择: 0: 无动作, 以便于通过 TRGO 将主机定时器和从机定时器完美同步起来 (完美同步时主机寄存器 CR2.MMM 需配为 010) 1: 延迟主机 TRGI 触发信号 注: 仅用于从机模式的触发模式 (SMS=110), 且由从机模式配置。
6:4	TS	RW	0x0	触发选择, 用于选择同步计数器的触发源: 000: 内部触发信号 (ITR0) 001: 内部触发信号 (ITR1) 010: 内部触发信号 (ITR2) 011: 内部触发信号 (ITR3) 100: TI1 边沿检测 (TI1F_ED) 101: 滤波后 TI1 (TI1FP1) 110: 滤波后 TI2 (TI2FP2) 111: 外部触发输入 (ETRF) 注: 仅当 SMS=000 即禁止从机模式的情况下, 可以改写 TS 寄存器。
3	RSV	-	-	保留

比特	名称	属性	复位值	描述
2:0	SMS	RW	0x0	<p>从机模式选择:</p> <p>000: 从机模式禁止; CEN 使能后预分频电路时钟源来自内部时钟</p> <p>001: Encoder 模式 1; 计数器使用 TI2FP2 边沿, 根据 TI1FP1 电平高低来计数</p> <p>010: Encoder 模式 2; 计数器使用 TI1FP1 边沿, 根据 TI2FP2 电平高低来计数</p> <p>011: Encoder 模式 3; 计数器同时使用 TI1FP1 和 TI2FP2 边沿, 根据其他输入信号电平来计数</p> <p>100: 复位模式; TRGI 上升沿初始化计数器, 并触发寄存器 update</p> <p>101: 闸门模式; TRGI 为高电平时, 计数时钟使能, TRGI 为低电平时, 计数时钟停止</p> <p>110: 触发模式; TRGI 上升沿触发计数器开始计数 (不会复位计数器)</p> <p>111: 外部时钟模式 1; TRGI 上升沿直接驱动计数器</p>

### 13.5.5 DMA 和中断使能寄存器 GTIMER\_DIER (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
19	CC4BUEN	RW	0x1	<p>捕捉比较通道 4 的 DMA 模式配置:</p> <p>0: Single 传输, DMA 读取 CCR4 寄存器的值传输到 RAM</p> <p>1: RAM-&gt;GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度</p>
18	CC3BUEN	RW	0x1	<p>捕捉比较通道 3 的 DMA 模式配置:</p> <p>0: CCR3-&gt;RAM, DMA 读取 CCR3 寄存器的值传输到 RAM</p> <p>1: RAM-&gt;GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度</p>
17	CC2BUEN	RW	0x1	<p>捕捉比较通道 2 的 DMA 模式配置:</p> <p>0: CCR2-&gt;RAM, DMA 读取 CCR2 寄存器的值传输到 RAM</p> <p>1: RAM-&gt;GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度</p>
16	CC1BUEN	RW	0x1	<p>捕捉比较通道 1 的 DMA 模式配置:</p> <p>0: CCR1-&gt;RAM, DMA 读取 CCR1 寄存器的值传输到 RAM</p> <p>1: RAM-&gt;GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度</p>
15	RSV	-	-	保留

比特	名称	属性	复位值	描述
14	TDE	RW	0x0	外部触发 DMA 请求使能: 0: 从机模式下, 禁止外部触发事件产生 DMA 请求 1: 从机模式下, 允许外部触发事件产生 DMA 请求 (可用于自动更新 preload 寄存器)
13	RSV	-	-	保留
12	CC4DE	RW	0x0	捕捉比较通道 4 的 DMA 请求使能: 0: 禁止 CC4 DMA 请求 1: 允许 CC4 DMA 请求
11	CC3DE	RW	0x0	捕捉比较通道 3 的 DMA 请求使能: 0: 禁止 CC3 DMA 请求 1: 允许 CC3 DMA 请求
10	CC2DE	RW	0x0	捕捉比较通道 2 的 DMA 请求使能: 0: 禁止 CC2 DMA 请求 1: 允许 CC2 DMA 请求
9	CC1DE	RW	0x0	捕捉比较通道 1 的 DMA 请求使能: 0: 禁止 CC1 DMA 请求 1: 允许 CC1 DMA 请求
8	UDE	RW	0x0	Update Event DMA 请求使能: 0: Update Event 发生时, 禁止产生 DMA 请求 1: Update Event 发生时, 允许产生 DMA 请求
7	RSV	-	-	保留
6	TIE	RW	0x0	触发事件中断使能: 0: 禁止触发事件中断 1: 允许触发事件中断
5	RSV	-	-	保留
4	CC4IE	RW	0x0	捕捉/比较通道 4 中断使能: 0: 禁止捕捉/比较 4 中断 1: 允许捕捉/比较 4 中断
3	CC3IE	RW	0x0	捕捉/比较通道 3 中断使能: 0: 禁止捕捉/比较 3 中断 1: 允许捕捉/比较 3 中断
2	CC2IE	RW	0x0	捕捉/比较通道 2 中断使能: 0: 禁止捕捉/比较 2 中断 1: 允许捕捉/比较 2 中断

比特	名称	属性	复位值	描述
1	CC1IE	RW	0x0	捕捉/比较通道 1 中断使能: 0: 禁止捕捉/比较 1 中断 1: 允许捕捉/比较 1 中断
0	UIE	RW	0x0	Update 事件中断使能: 0: 禁止 Update 事件中断 1: 允许 Update 事件中断

### 13.5.6 状态寄存器 GTIMER\_SR (偏移: 10h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	CC4OF	W1C	0x0	捕捉/比较通道 4 的 Overcapture 状态 参考CC1OF
11	CC3OF	W1C	0x0	捕捉/比较通道 3 的 Overcapture 状态 参考CC1OF
10	CC2OF	W1C	0x0	捕捉/比较通道 2 的 Overcapture 状态 参考CC1OF
9	CC1OF	W1C	0x0	捕捉/比较通道 1 的 Overcapture 状态 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0: 无 overcapture 事件 1: 在CC1IF标志为1的情况下发生新的捕捉
8:7	RSV	-	-	保留
6	TIF	W1C	0x0	触发事件中断标志，硬件置位，软件写 1 清零
5	RSV	-	-	保留
4	CC4IF	W1C	0x0	捕捉/比较通道 4 中断标志 参考CC1IF
3	CC3IF	W1C	0x0	捕捉/比较通道 3 中断标志 参考CC3IF
2	CC2IF	W1C	0x0	捕捉/比较通道 2 中断标志 参考CC2IF
1	CC1IF	W1C	0x0	捕捉/比较通道 1 中断标志 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果CC1通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读CCR1自动清零。

比特	名称	属性	复位值	描述
0	UIF	W1C	0x0	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器： -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0且UDIS=0的情况下，触发事件初始化计数器

### 13.5.7 事件产生寄存器 GTIMER\_EGR (偏移: 14h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	TG	W	0x0	软件触发，软件置位此寄存器产生触发事件，硬件自动清零
5	RSV	-	-	保留
4	CC4G	W	0x0	捕捉/比较通道 4 软件触发，参考 CC1G
3	CC3G	W	0x0	捕捉/比较通道 3 软件触发，参考 CC1G
2	CC2G	W	0x0	捕捉/比较通道 2 软件触发，参考 CC1G
1	CC1G	W	0x0	捕捉/比较通道 1 软件触发，硬件自动清零： 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求。 如果CC1通道配置为输入：当前计数值被捕捉到CCR1寄存器，CC1IF置位，在使能的情况下可以产生相应的中断和DMA请求。
0	UG	W	0x0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零。 软件置位UG时会重新初始化计数器并更新shadow寄存器，预分频计数器被清零。

### 13.5.8 捕捉/比较模式寄存器 1 GTIMER\_CCMR1 (偏移: 18h)

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能：

#### 1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC2CE	RW	0x0	输出比较 2 清零使能，参考 OC1CE
14:12	OC2M	RW	0x0	输出比较 2 模式配置，参考 OC1M

比特	名称	属性	复位值	描述
11	OC2PE	RW	0x0	输出比较 2 预装载使能, 参考 OC1PE
10	OC2FE	RW	0x0	输出比较 2 快速使能, 参考 OC1FE
9:8	CC2S	RW	0x0	捕捉/比较 2 通道选择: 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 (如果 BPS2 置 1, 则映射到 TI2 边沿检测) 注: CC2S仅在通道关闭时 (CC2E=0) 可以写。
7	OC1CE	RW	0x0	输出比较 1 清零使能: 0: OC1REF 不受 ETRF 影响 1: 检测到 ETRF 高电平时, 自动清零 OC1REF。
6:4	OC1M	RW	0x0	输出比较 1 模式配置, 此寄存器定义 OC1REF 信号的行为: 000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001: CCR1=CNT 时, 将 OC1REF 置高 010: CCR1=CNT 时, 将 OC1REF 置低 011: CCR1=CNT 时, 翻转 OC1REF 100: OC1REF 固定为低 (inactive) 101: OC1REF 固定为高 (active) 110: PWM 模式 1——在向上计数时, OC1REF 在 CNT<CCR1 时置高, 否则置低; 在向下计数时, OC1REF 在 CNT>CCR1 时置低, 否则置高。 111: PWM 模式 2——在向上计数时, OC1REF 在 CNT<CCR1 时置低, 否则置高; 在向下计数时, OC1REF 在 CNT>CCR1 时置高, 否则置低。
3	OC1PE	RW	0x0	输出比较 1 预装载使能: 0: CCR1 preload 寄存器无效, CCR1 可以直接写入 1: CCR1 preload 寄存器有效, 针对 CCR1 的读写操作都是访问 preload 寄存器, 当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中。
2	OC1FE	RW	0x0	输出比较 1 快速使能: 0: 关闭快速使能, trigger 输入不会影响比较输出 1: 打开快速使能, trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出, 而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效。

比特	名称	属性	复位值	描述
1:0	CC1S	RW	0x0	捕捉/比较 1 通道选择: 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 (如果 BPS1 置 1, 则映射到 TI1 边沿检测) 注: CC1S仅在通道关闭时 (CC1E=0) 可以写。

## 2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC2F	RW	0x0	输入捕捉 2 滤波
11:10	IC2PSC	RW	0x0	输入捕捉 2 预分频
9:8	CC2S	RW	0x0	捕捉/比较 2 通道选择: 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 (如果 BPS2 置 1, 则映射到 TI2 边沿检测) 注: CC2S仅在通道关闭时 (CC2E=0) 可以写。
7:4	IC1F	RW	0x0	输入捕捉 1 滤波: 此寄存器定义 TI1 的采样频率和滤波长度: 0000: 无滤波, 使用 $f_{DTS}$ 采样 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: $f_{SAMPLING}=f_{DTS}/32$ , $N=5$

比特	名称	属性	复位值	描述
				1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$ , $N=6$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$ , $N=8$
3:2	IC1PSC	RW	0x0	输入捕捉 1 预分频: 00: 无分频 01: 每 2 个事件输入产生一次捕捉 10: 每 4 个事件输入产生一次捕捉 11: 每 8 个事件输入产生一次捕捉 注: IC1PSC寄存器在CC1E=0时复位。
1:0	CC1S	RW	0x0	捕捉/比较 1 通道选择: 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 (如果 BPS1 置 1, 则映射到 TI1 边沿检测) 注: CC1S仅在通道关闭时 (CC1E=0) 可以写。

### 13.5.9 捕捉/比较模式寄存器 2 GTIMER\_CCMR2 (偏移: 1Ch)

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能:

#### 1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC4CE	RW	0x0	输出比较 4 清零使能, 参考 OC1CE
14:12	OC4M	RW	0x0	输出比较 4 模式配置, 参考 OC1M
11	OC4PE	RW	0x0	输出比较 4 预装载使能, 参考 OC1PE
10	OC4FE	RW	0x0	输出比较 4 快速使能, 参考 OC1FE
9:8	CC4S	RW	0x0	捕捉/比较 4 通道选择: 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 (如果 BPS4 置 1, 则映射到 TI4 边沿检测) 注: CC4S仅在通道关闭时 (CC4E=0) 可以写。
7	OC3CE	RW	0x0	输出比较 4 清零使能: 0: OC4REF 不受 ETRF 影响

比特	名称	属性	复位值	描述
				1: 检测到ETRF高电平时, 自动清零OC4REF
6:4	OC3M	RW	0x0	<p>输出比较 3 模式配置, 此寄存器定义 OC3REF 信号的行为:</p> <p>000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出</p> <p>001: CCR3=CNT 时, 将 OC3REF 置高</p> <p>010: CCR3=CNT 时, 将 OC3REF 置低</p> <p>011: CCR3=CNT 时, 翻转 OC3REF</p> <p>100: OC3REF 固定为低 (inactive)</p> <p>101: OC3REF 固定为高 (active)</p> <p>110: PWM 模式 1 -在向上计数时, OC3REF 在 CNT&lt;CCR3 时置高, 否则置低; 在向下计数时, OC3REF 在 CNT&gt;CCR3 时置低, 否则置高。</p> <p>111: PWM模式2 -在向上计数时, OC3REF在CNT&lt;CCR3时置低, 否则置高; 在向下计数时, OC3REF在CNT&gt;CCR3 时置高, 否则置低。</p>
3	OC3PE	RW	0x0	<p>输出比较 3 预装载使能:</p> <p>0: CCR3 preload 寄存器无效, CCR4 可以直接写入</p> <p>1: CCR3 preload寄存器有效, 针对CCR4的读写操作都是访问preload寄存器, 当update event发生时才将preload寄存器的内容转移到shadow寄存器中。</p>
2	OC3FE	RW	0x0	<p>输出比较 3 快速使能:</p> <p>0: 关闭快速使能, trigger 输入不会影响比较输出</p> <p>1: 打开快速使能, trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出, 而不管当前实际比较情况</p> <p>注: 此功能仅在当前通道配置为PWM1或PWM2模式时有效。</p>
1:0	CC3S	RW	0x0	<p>捕捉/比较 3 通道选择:</p> <p>00: CC3 通道配置为输出</p> <p>01: CC3 通道配置为输入, IC3 映射到 TI3</p> <p>10: CC3 通道配置为输入, IC3 映射到 TI4</p> <p>11: CC3 通道配置为输入, IC3 映射到 TRC (如果 BPS3 置 1, 则映射到 TI3 边沿检测)</p> <p>注: CC3S仅在通道关闭时 (CC3E=0) 可以写。</p>

## 2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC4F	RW	0x0	输入捕捉 4 滤波
11:10	IC4PSC	RW	0x0	输入捕捉 4 预分频
9:8	CC4S	RW	0x0	捕捉/比较 4 通道选择: 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 (如果 BPS4 置 1, 则映射到 TI4 边沿检测) 注: CC4S 仅在通道关闭时 (CC4E=0) 可以写
7:4	IC3F	RW	0x0	输入捕捉 3 滤波: 此寄存器定义 TI3 的采样频率和滤波长度: 0000: 无滤波, 使用 $f_{DTS}$ 采样 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: $f_{SAMPLING}=f_{DTS}/32$ , $N=8$
3:2	IC3PSC	RW	0x0	输入捕捉 3 预分频: 00: 无分频 01: 每 2 个事件输入产生一次捕捉 10: 每 4 个事件输入产生一次捕捉 11: 每 8 个事件输入产生一次捕捉 注: IC3PSC 寄存器在 CC3E=0 时复位。

比特	名称	属性	复位值	描述
1:0	CC3S	RW	0x0	捕捉/比较 3 通道选择: 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 (如果 BPS3 置 1, 则映射到 TI3 边沿检测) 注: CC3S 仅在通道关闭时 (CC1E=0) 可以写。

### 13.5.10 捕捉/比较使能寄存器 GTIMER\_CCER (偏移: 20h)

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留
13	CC4P	RW	0x0	捕捉/比较 4 输出极性, 参考 CC1P
12	CC4E	RW	0x0	捕捉/比较 4 输出使能, 参考 CC1E
11:10	RSV	-	-	保留
9	CC3P	RW	0x0	捕捉/比较 3 输出极性, 参考 CC1P
8	CC3E	RW	0x0	捕捉/比较 3 输出使能, 参考 CC1E
7:6	RSV	-	-	保留
5	CC2P	RW	0x0	捕捉/比较 2 输出极性, 参考 CC1P
4	CC2E	RW	0x0	捕捉/比较 2 输出使能, 参考 CC1E
3:2	RSV	-	-	保留
1	CC1P	RW	0x0	捕捉/比较 1 输出极性: CC1 通道配置为输出时: 0: OC1 高电平 active 1: OC1 低电平 active CC1 通道配置为输入时: 0: 非取反模式-捕捉在 IC1 的上升沿进行 1: 取反模式-捕捉在 IC1 的下降沿进行
0	CC1E	RW	0x0	捕捉/比较 1 输出使能: CC1 通道配置为输出时: 0: OC1 不 active 1: OC1 active CC1 通道配置为输入时: 0: 关闭捕捉功能 1: 使能捕捉功能

### 13.5.11 计数器寄存器 GTIMER\_CNT (偏移: 24h)

比特	名称	属性	复位值	描述
31:0	CNT	RW	0x0	计数值

### 13.5.12 预分频寄存器 GTIMER\_PSC (偏移: 28h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC	RW	0x0	<p>计数器时钟 (CK_CNT) 预分频值:  <math>f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)</math></p> <p>这是一个preload寄存器, 在update事件发生时其内容被载入shadow寄存器。</p> <p>注: 支持的最高PWM输出为30MHz, 配置的PSC和ARR需注意该条件。</p>

### 13.5.13 自动重载寄存器 GTIMER\_ARR (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:0	ARR	RW	0x0	<p>计数溢出时的自动重载值;</p> <p>这是一个preload寄存器, 在update事件发生时其内容被载入shadow寄存器。</p>

### 13.5.14 捕捉/比较寄存器 1 GTIMER\_CCR1 (偏移: 34h)

比特	名称	属性	复位值	描述
31:0	CCR1	RW	0x0	<p>捕捉/比较通道1寄存器</p> <p><b>如果通道1配置为输出:</b></p> <p>这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC1输出。</p> <p><b>如果通道1配置为输入:</b></p> <p>CCR1保存最近一次输入捕捉事件发生时的计数器值, 此时CCR1为只读。</p>

**13.5.15 捕捉/比较寄存器 2 GTIMER\_CCR2 (偏移: 38h)**

比特	名称	属性	复位值	描述
31:0	CCR2	RW	0x0	捕捉/比较通道2寄存器 <b>如果通道2配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC2输出。 <b>如果通道2配置为输入:</b> CCR2保存最近一次输入捕捉事件发生时的计数器值, 此时CCR2为只读。

**13.5.16 捕捉/比较寄存器 3 GTIMER\_CCR3 (偏移: 3Ch)**

比特	名称	属性	复位值	描述
31:0	CCR3	RW	0x0	捕捉/比较通道3寄存器 <b>如果通道3配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC3输出。 <b>如果通道3配置为输入:</b> CCR3保存最近一次输入捕捉事件发生时的计数器值, 此时CCR3为只读。

**13.5.17 捕捉/比较寄存器 4 GTIMER\_CCR4 (偏移: 40h)**

比特	名称	属性	复位值	描述
31:0	CCR4	RW	0x0	捕捉/比较通道4寄存器 <b>如果通道4配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC4输出。 <b>如果通道4配置为输入:</b> CCR4保存最近一次输入捕捉事件发生时的计数器值, 此时CCR4为只读。

### 13.5.18 DMA 控制寄存器 GTIMER\_DCR (偏移: 48h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12:8	DBL	RW	0x0	DMA Burst长度: 对DMAR寄存器的读写将触发Burst DMA操作, Burst长度为1~18。 00000: 长度=1 00001: 长度=2 ..... 10001: 长度=18 其它: 无效值, 禁止写入
7:5	RSV	-	-	保留
4:0	DBA	RW	0x0	DMA基地址, 定义指向寄存器的偏移地址: 00000: CR1 00001: CR2 00010: SMCR ..... 01011: ARR 01100: RSV 01101: CCR1 ..... 10000: CCR4 10001: RSV 11000: DCR 注: 当DBA+DBL超出了GTIMER寄存器地址范围, 则实际Burst传输到GTIMER最高寄存器地址后自动停止, 即Burst长度会缩短。

### 13.5.19 DMA 访问寄存器 GTIMER\_DMAR (偏移: 4Ch)

比特	名称	属性	复位值	描述
31:0	DMAR	RW	0x400	DMA burst访问寄存器 在使用DMA burst传输时, 将DMA通道外设地址设置为DMAR, GTIMER会根据DBL的值产生多次DMA请求。

# 14 通用定时器 GTIMER1/2

## 14.1 概述

通用定时/计数器 (GTIMER1/2) 包含一个 16 位自动重载计数器及一个可编程预分频器, 有自己独立的中断, 可以有多种用途, 包括测量输入信号的脉冲宽度 (输入捕获) 和产生输出波形。计数器可以向上, 向下, 向上/下三种计数方向, 且计数值可以随时由软件读取。

## 14.2 主要特性

- 16 位向上、向下、向上/下计数自动重载计数器
- 16 位可编程预分频器, 支持实时调整计数时钟分频
- 灵活的计数时钟源选择
- 通道可用于输入捕获、输出比较、PWM (边沿或中央对齐模式)、单脉冲输出
- 支持定时器间的级联
- 中断在以下几种情况产生:
  - Update 中断: 计数器向上/向下溢出
  - 输入捕获
  - 输出比较
- 支持增量正交编码器和霍尔传感器

## 14.3 通用定时器 GTIMER1/2 结构框图

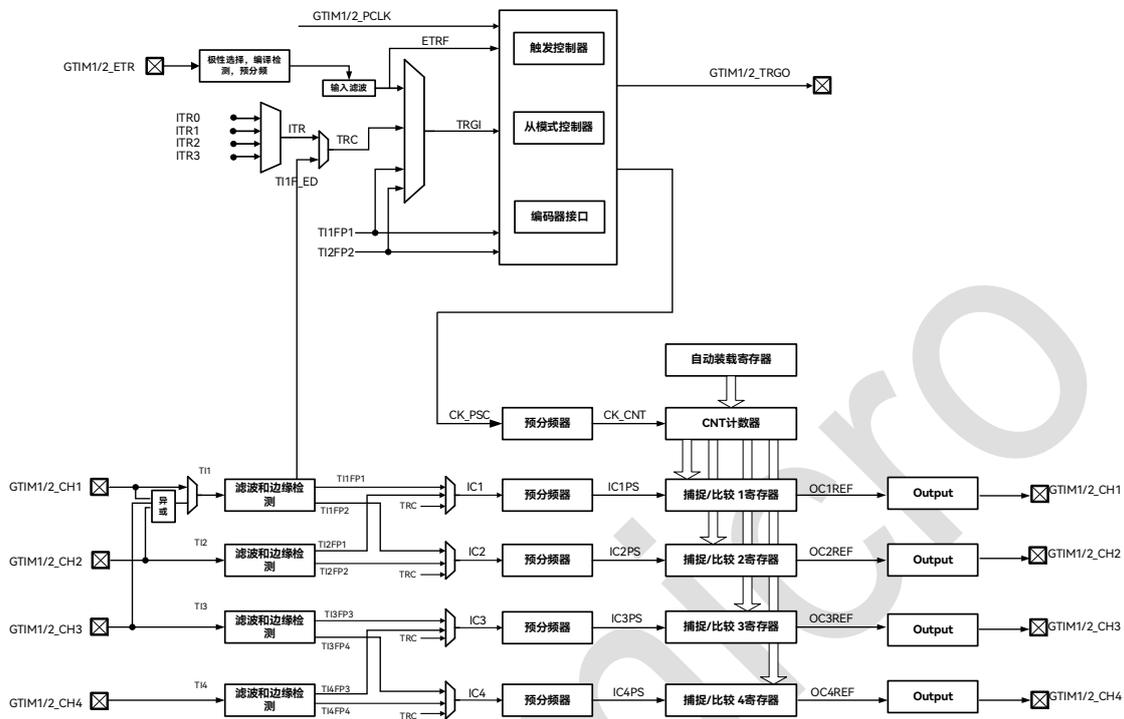


图 14-1: GTIMER 结构框图

## 14.4 功能描述

### 14.4.1 定时单元

GTIMER1/2 主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

定时单元包含如下寄存器：

- 计数器寄存器 (GTIMER\_CNT)
- 预分频寄存器 (GTIMER\_PSC)
- 自动装载寄存器 (GTIMER\_ARR)

GTIMER\_ARR 包含预装载功能, 根据在 GTIMER\_CR1 寄存器中的自动装载预装载使能位 (ARPE)

的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件（向下计数时的下溢条件）并当 GTIMER\_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。软件也可以通过寄存器（GTIMER\_EGR[0]写 1）操作主动触发 ARR 更新。

计数器工作时钟由 GTIMER\_PSC 产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，GTIMER\_CNT 才开始计数。

GTIMER\_PSC 是一个同步预分频器，预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是一个 16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

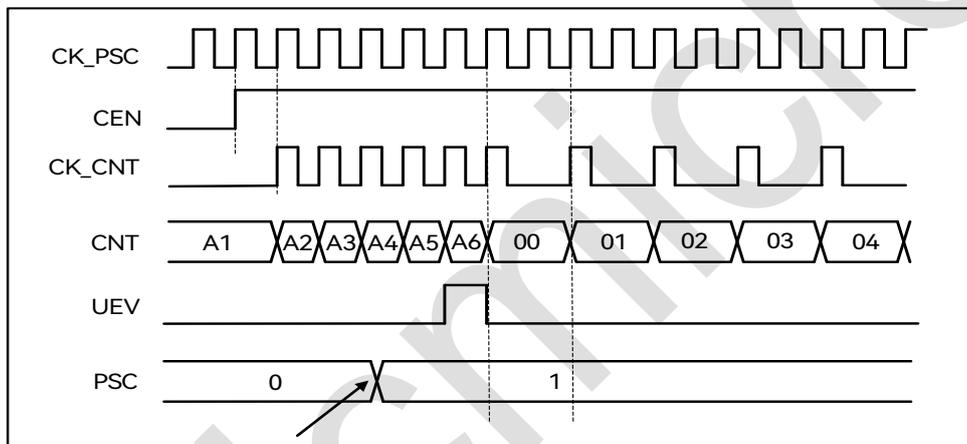


图 14-2: GTIMER 预分频从 1 变为 2 的波形图

## 14.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

### 14.4.2.1 向上计数

在向上计数模式中，计数器从 0 计数到自动加载值（ARR 计数器的值），然后重新从 0 开始计数并且产生一个计数器溢出事件。

在 GTIMER\_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。设置 GTIMER\_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清 0 之前，将不产生更新事件。但是在应

该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。

如果设置了 GTIMER\_CR1 寄存器中的 URS 位，设置 UG 位将产生一个更新事件 UEV，但硬件不产生 UIF 标志。

当发生一个更新事件时，以下寄存器都被更新，硬件同时置位 UIF 标志位：

- GTIMER\_ARR 影子寄存器被更新为 GTIMER\_ARR 内容。
- GTIMER\_PSC 影子寄存器被更新为 GTIMER\_PSC 内容。

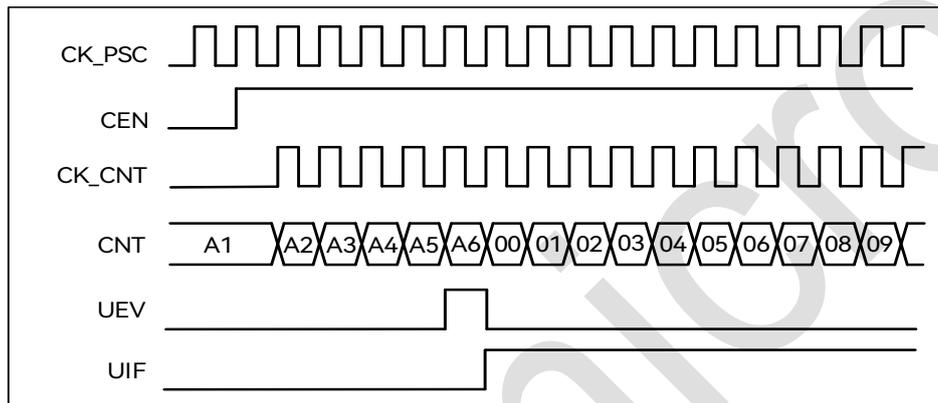


图 14-3: 向上计数波形图

#### 14.4.2.2 向下计数

在向下计数模式中，计数器从 GTIMER\_ARR 开始倒数计数到 0，然后重新从 GTIMER\_ARR 开始计数并且产生一个计数器溢出事件。

在 GTIMER\_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。设置 GTIMER\_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清 0 之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。

如果设置了 GTIMER\_CR1 寄存器中的 URS 位，设置 UG 位将产生一个更新事件 UEV，但硬件不产生 UIF 标志。

当发生一个更新事件时，以下寄存器都被更新，硬件同时置位 UIF 标志位：

- GTIMER\_ARR 影子寄存器被更新为 GTIMER\_ARR 内容。
- GTIMER\_PSC 影子寄存器被更新为 GTIMER\_PSC 内容。

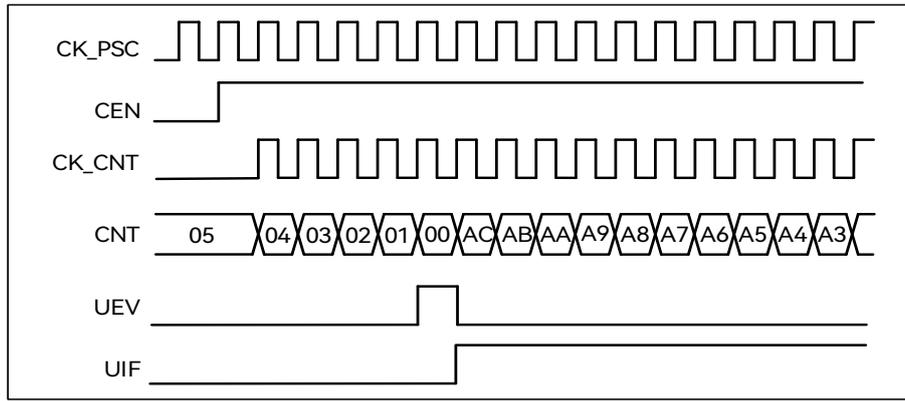


图 14-4: 向下计数图

### 14.4.2.3 中心对齐计数

在中心对齐模式下，计数器从 0 开始向上计数，到 GTIMER\_ARR-1 产生上溢出事件，然后从 GTIMER\_ARR 开始向下计数到 1，产生下溢出事件，再从 0 重新开始向上计数。

CR1[6:5]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式：

- CMS = 00: 边沿对齐模式
- 01: 中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位。
- 10: 中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位。
- 11: 中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位。

中心对齐模式下，DIR 寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。计数器在上溢和下溢的事件上都会更新如下寄存器：

- GTIMER\_ARR 影子寄存器被更新为 GTIMER\_ARR 内容。
- GTIMER\_PSC 影子寄存器被更新为 GTIMER\_PSC 内容。

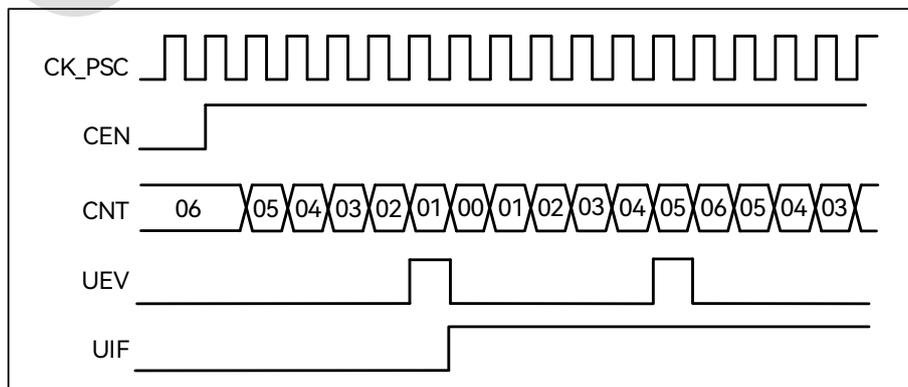


图 14-5: 中心对齐计数器时序图

### 14.4.3 计数器工作时钟

计数器可以使用如下时钟工作：

- 内部时钟 APBCLK (CK\_INT)
- 外部引脚输入时钟 GTIMER\_CHx (x=1,2)
- 外部引脚触发输入 (GTIMER\_ETR)
- 内部触发 (ITRx)

#### 14.4.3.1 APBCLK

当选择 APBCLK 作为时钟源时，禁止 GTIMER 设置为从机模式 (SMS=000)，CEN、DIR、UG 等寄存器位都是软件控制。软件操作 UG 寄存器后，分频器，计数器值将被重新初始化。

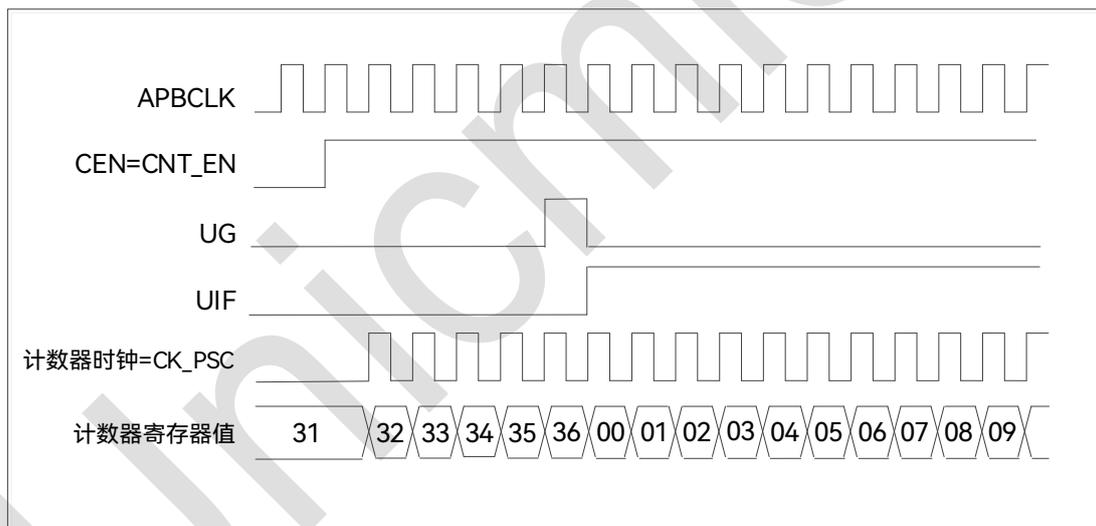


图 14-6: APBCLK 时钟源模式，时钟分频因子为 1

#### 14.4.3.2 GTIMER\_CHx (x=1,2) 引脚输入——外部时钟模式 1

配置 SMS=111，此模式下直接使用外部引脚 GTIMER\_CHx 输入信号作为计数时钟，计数边沿可以选择为上升或下降沿。

此模式下只有通道 1 和 2 的输入可以用做时钟输入，以 GTIMER\_CH1 为例，配置流程如下：

1. 在 IO 模块种配置相应管脚为 GTIMER\_CH2 功能。

2. 配置 GTIMER->CCMR1[9:8]=01，配置通道 2 检测 TI2 输入的上升沿。
3. 配置 GTIMER->CCMR1[15:12]为需要滤波的值，选择输入滤波器带宽（如果不需要滤波器，保持 CCMR1[15:12]=0000）。
4. 配置 GTIMER->CCER[5]=0，选定 TI2 上升沿极性。
5. 配置 GTIMER->SMCR[2:0]=111，选择定时器外部时钟模式 1。
6. 配置 GTIMER->SMCR[6:4]=110，选定 TI2 作为触发输入源。
7. 打开通道使能，配置 GTIMER->CCER[0]=1。
8. 使能计数器，配置 GTIMER->CR1[0]=1。

#### 14.4.3.3 GTIMER\_ETR 引脚输入——外部时钟模式 2

此模式下使用 GTIMER\_ETR 管脚输入信号的上升沿或下降沿来计数。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的同步电路。

在使用外部时钟模式 2 时，仍可以将 GTIMER 配置为 slave 模式：比如使用 GTIMER\_ETR 输入计数，同时用另一个 Timer 的 TRGO 作为触发信号，当触发事件到来时，复位计数器重新开始计数。

#### 14.4.4 内部触发信号 (ITRx)

GTIMER 支持 4 个 ITR 输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将 TS 配置为 000~011 用于选择 ITR0~ITR3，并将 CCxS 配置为 11，即将 TRC 选为捕捉信号。

每个 ITR 输入支持 4 个内部信号扩展，由 TS 寄存器配置。输入信号源参考下表：

表 14-1: GTIMER0/1/2 内部信号扩展表

Slave	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
GTIMER0	RCL32k	ATIMER_TRGO	COMP0_IN	COMP1_IN
GTIMER1	RCL32k	GTIMER0_TRGO	COMP0_IN	COMP1_IN
GTIMER2	RCL32k	GTIMER1_TRGO	COMP0_IN	COMP1_IN

### 14.4.5 捕捉/比较通道

通用定时器 1 包含 4 个捕捉/比较通道，每个通道由一个捕捉/比较寄存器 CCR（包含影子寄存器），输入电路（数字滤波、多路复用和预分频器），和输出电路（比较器和输出控制）组成。

#### 14.4.5.1 捕捉通道

输入电路对相应的  $Tl_x$  ( $x=1...4$ ) 进行采样并产生滤波后的信号  $Tl_xF$  ( $x=1...4$ )，然后经过边沿检测和极性选择产生  $Tl_xFP_x$  ( $x=1...4$ ) 信号，此信号可作为计数触发源或者捕捉源，并通过预分频器后进入捕捉通道。

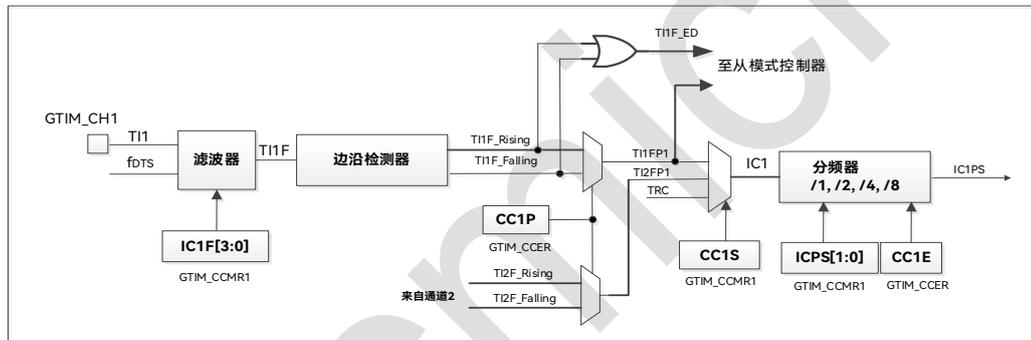


图 14-7: 捕捉/比较通道 (通道 1 输入部分)

#### 14.4.5.2 比较通道

输出级电路产生一个中间信号  $OC_xREF$  (高有效) 作为基准，给最终输出电路作为参考输入。

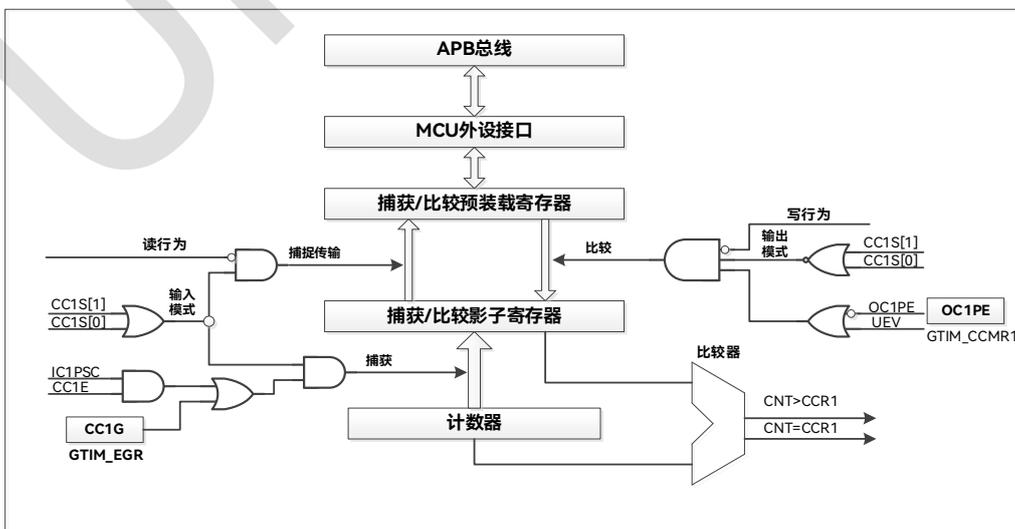


图 14-8: 捕捉/比较通道 1 的主电路

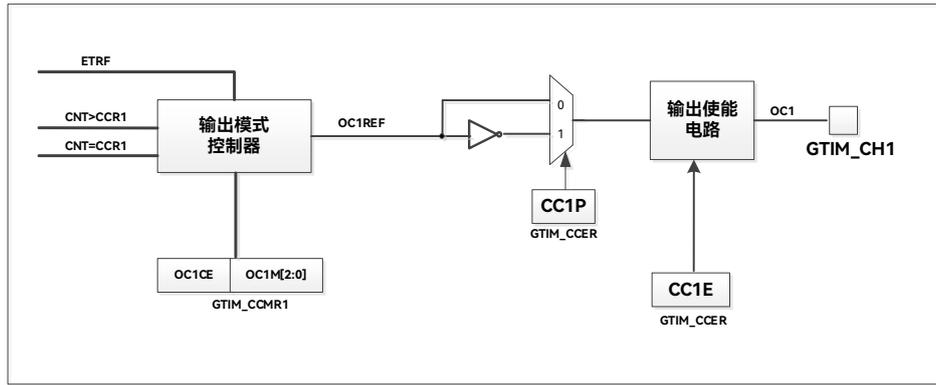


图 14-9: 捕获/比较通道的输出部分

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

#### 14.4.6 输入捕捉模式

在输入捕获模式下，当检测到 ICx 信号上出现相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（GTIMER\_CCRx）中。当发生捕获事件时，相应的 CCxIF 标志（GTIMER\_SR 寄存器）被置 1，如果使能了中断或者 DMA，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（GTIMER\_SR 寄存器）被置 1。写 CCxIF 或者读取存储在 GTIMER\_CCRx 寄存器中的捕获数据都可清除 CCxIF。写 CCxOF=1 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值，步骤如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 选择输入通道，配置 GTIMER->CCMR1.CC1S=01，IC1 映射到 TI1。
3. 选择 TI1 通道的计数有效沿，配置 GTIMER\_CCER.CC1P=0，选择上升沿计数。
4. 配置输入滤波时间，配置 GTIMER\_CCMR1.IC1F。
5. 配置输入预分频器，配置 GTIMER\_CCMR1.IC1PS。
6. 设置 GTIMER\_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
7. 根据需要打开相应中断使能或 DMA 请求使能。
8. 打开通道使能，配置 GTIMER\_CCER[0]=1。

### 14.4.7 PWM 输入模式

该模式是输入捕获模式的一个特例，通过两个通道配合，可以实现 PWM 信号脉宽的输入捕捉。

实现 PWM 输入捕获功能，配置流程如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 选择 GTIMER\_CCR1 的有效输入：置 GTIMER\_CCMR1 寄存器的 CC1S=01（选中 TI1）。
3. 选择 TI1FP1 的有效极性：置 CC1P=0（上升沿有效）。
4. 选择 GTIMER\_CCR2 的有效输入：置 GTIMER\_CCMR1 寄存器的 CC2S=10（选中 TI1）。
5. 选择 TI1FP2 的有效极性：置 CC2P=1（下降沿有效）。
6. 选择有效的触发输入信号：置 GTIMER\_SMCR 寄存器中的 TS=101。
7. 配置从模式控制器为复位模式：置 GTIMER\_SMCR 中的 SMS=100。
8. 使能捕获：置 GTIMER\_CCER 寄存器中 CC1E=1 且 CC2E=1。

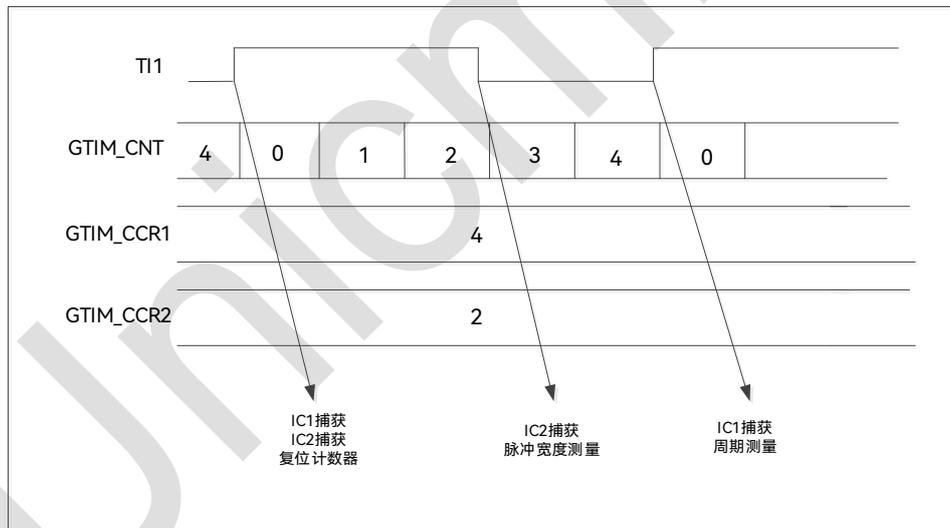


图 14-10: PWM 输入捕获

### 14.4.8 软件强制输出

在强制输出模式下，输出比较信号（OCxREF 和相应的 OCx/OCxN）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 GTIMER\_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号为有效状态。这样

OCxREF 被强置为高电平，同时 OCx 得到 CCxP 极性相反的信号。例如：CCxP=0 (OCx 高电平有效)，则 OCx 被强置为高电平。置 GTIMER\_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

### 14.4.9 输出比较模式

输出比较模式下，当 CCR 与计数器值相等，OCxREF 可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA 请求可以发送。输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

1. 选择计数时钟（内部、外部、预分频等）。
2. 设置 GTIMER\_ARR 和 GTIMER\_CCR 寄存器的值。
3. 根据需要相应设置中断使能和 DMA 使能。
4. 选择输出模式，例如：
  - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011。
  - 置 OCxPE = 0 禁用预装载寄存器。
  - 置 CCxP = 0 选择极性为高电平有效。
  - 置 CCxE = 1 使能输出。

设置计数器使能 CEN=1。

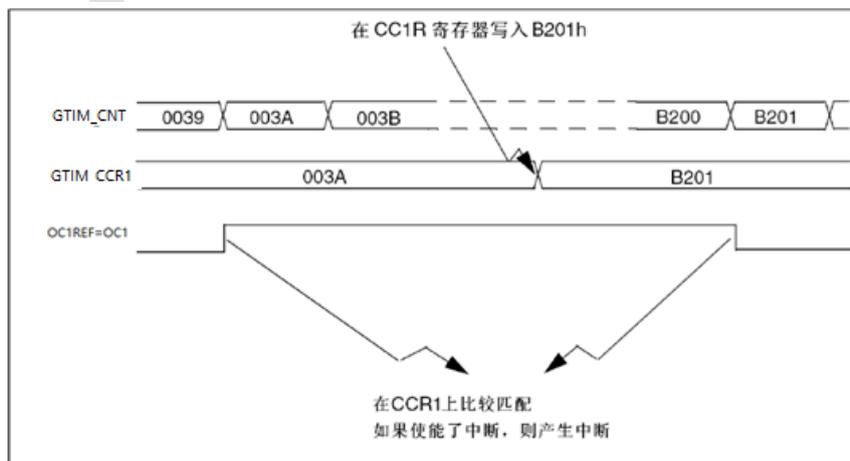


图 14-11: 输出比较模式，翻转 OC1

在不使能 preload 的情况下，软件可以随时改写 CCR 寄存器实现对输出波形的实时控制。如果使能了 preload，则 CCR shadow 寄存器仅在下一次 update event 发生时更新为 preload 寄存器的内容。

## 14.4.10 PWM 输出

PWM 模式可以输出脉宽调制信号，其周期由 ARR 寄存器决定，占空比由 CCR 寄存器决定。输出信号的极性可以由 CCxP 寄存器配置。PWM 模式工作中，CNT 和 CCR 实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM 输出也支持边缘对齐和中央对齐模式。

### 14.4.10.1 PWM 边缘对齐模式

在向上计数的情况下，配置为 PWM 模式 1 时，OCxREF 信号在  $CNT < CCR$  时为高电平，否则为低电平。如果 CCR 值大于 ARR 值，则 OCxREF 被固定为 1；如果 CCR 为 0，则 OCxREF 被固定为 0。

在向下计数时，OCxREF 电平高低定义与向上计数时相同。

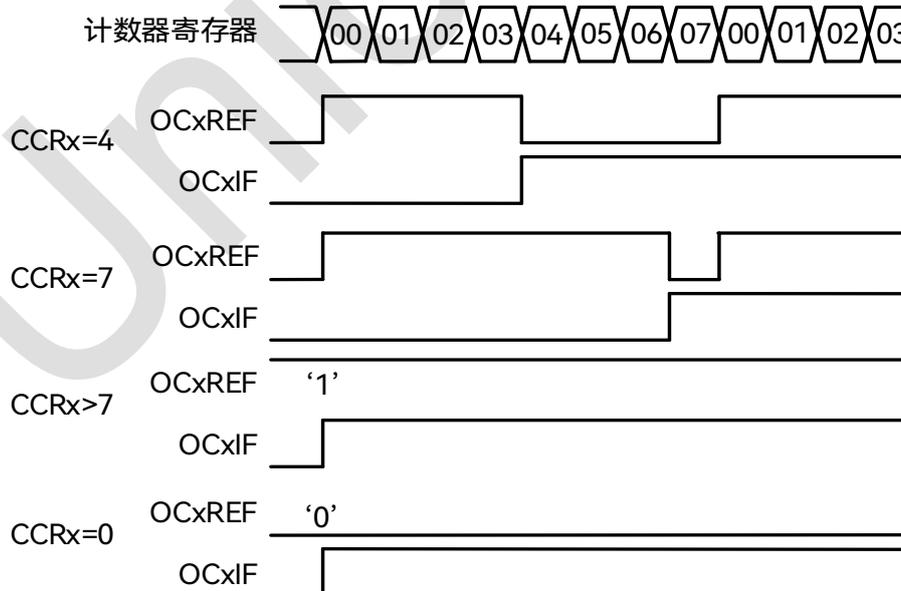


图 14-12: 边沿对齐的 PWM 波形 (ARR=7)

### 14.4.10.2 PWM 中央对齐模式

OCxREF 电平定义与边缘对齐模式相同。

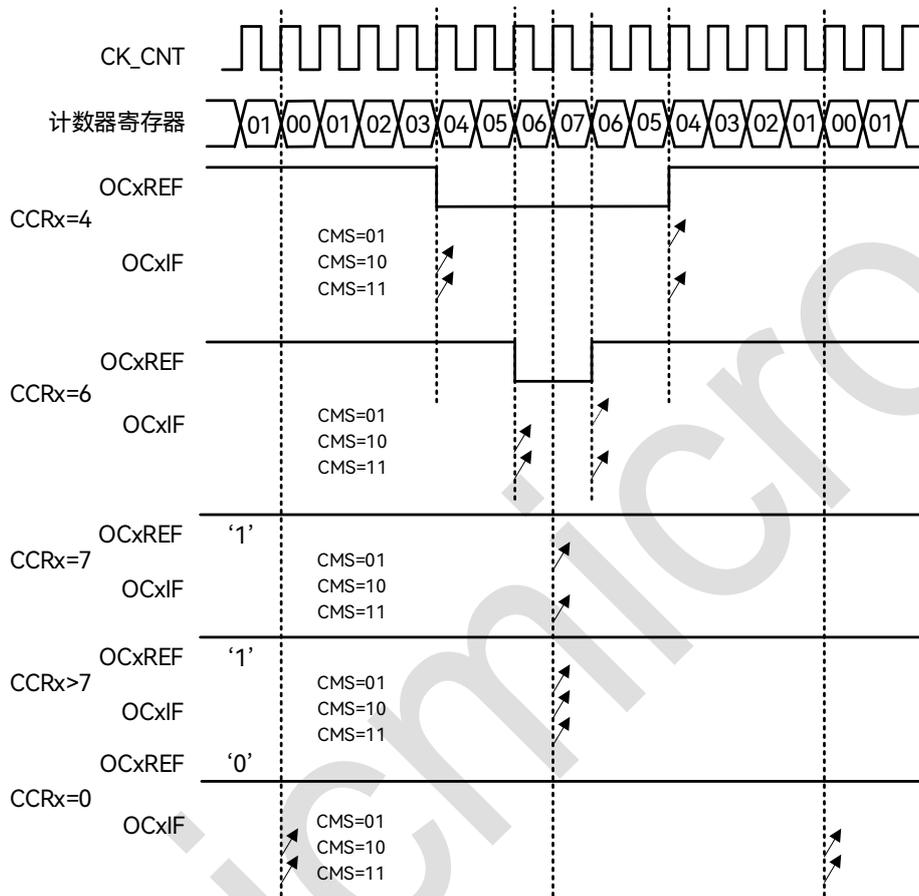


图 14-13: 中央对齐的 PWM 波形 (APR=7)

当启动中央对齐计数时,一开始的计数方向是由 DIR 寄存器决定的;随后在计数过程中,DIR 寄存器的状态由硬件直接控制。安全起见,建议用户程序在启动计数器之前,通过 UG 寄存器做一次 update,并且在计数过程中不要改写计数器。

### 14.4.11 输出比较快速使能模式

打开快速使能模式 (CCMR1[2, 10]、CCMR2[2, 10]) 后 GTIMER 的比较输出结果受 TRGI 信号影响。TRGI 信号选择是通过寄存器 SMCR[4:6]来设置的。以通道 1 为例:

**假设 CC1P 是 0 的情况:**

- 当 TRGI 信号到来后,如果采用 PWM 模式 1,不管 PWM 本来是什么电平,GTIMER\_CH1 都被

强制变为 0 电平，等到计数到达 ARR 后恢复为正常。

- 当 TRGI 信号到来后，如果采用 PWM 模式 2，不管 PWM 本来是什么电平，GTIMER\_CH1 都被强制变为 1 电平，等到计数到达 ARR 后恢复为正常。

**假设 CC1P 是 1 的情况：**

- 当 TRGI 信号到来后，如果采用 PWM 模式 1，不管 PWM 本来是什么电平，GTIMER\_CH1 都被强制变为 1 电平，等到计数到达 ARR 后恢复为正常。
- 当 TRGI 信号到来后，如果采用 PWM 模式 2，不管 PWM 本来是什么电平，GTIMER\_CH1 都被强制变为 0 电平，等到计数到达 ARR 后恢复为正常。

#### 14.4.12 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下次 update event 到来时，计数器会自动停止。只有当 CCR 和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求  $CNT < CCR \leq ARR$ ；在向下计数时，要求  $CNT > CCR$ 。

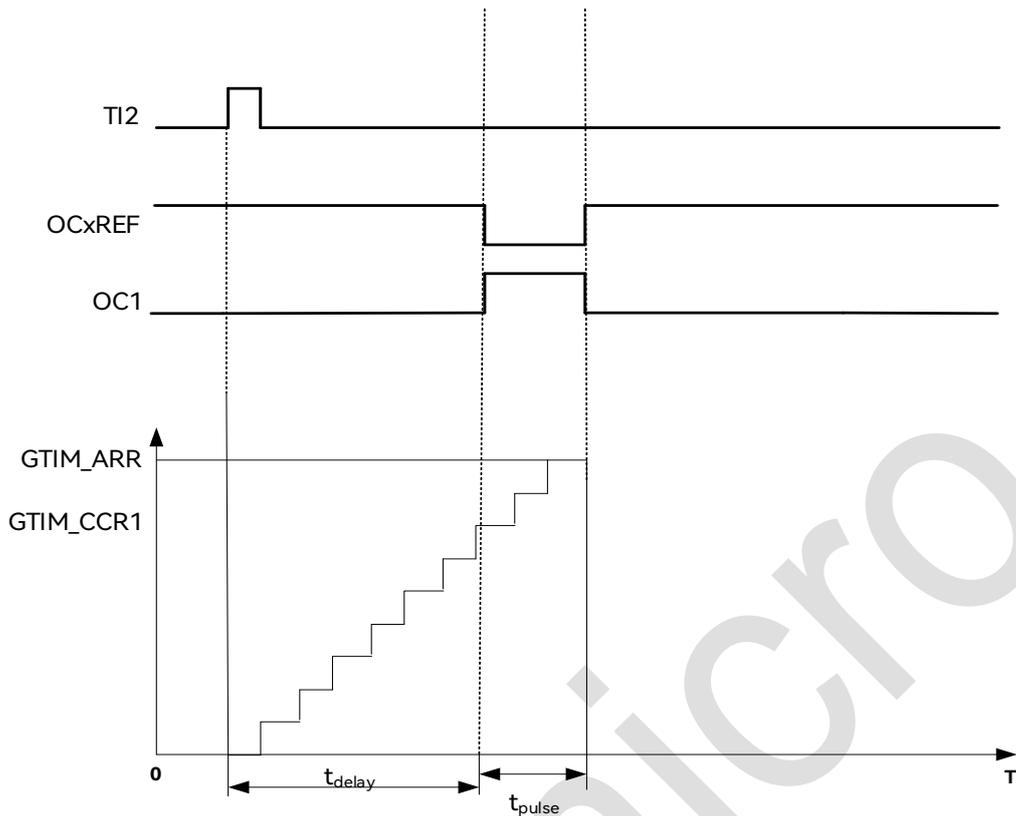


图 14-14: 单脉冲模式的例子

图 14-14 所示是以 TI2 输入为计数器触发信号，计数值等于 CCR 后 OCxREF 输出低电平，计数到 ARR 后 OCxREF 回到高电平，并且计数器回 0，停止计数。

**实现上述功能 TI2 作为输入触发的配置如下：**

- 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH2 功能。
- 关闭通道使能，配置 `GTIMER->CCER.CC2E=0`，确保之后的配置成功。
- 选择输入通道，配置 `GTIMER->CCMR1.CC2S=01`。
- 选择计数有效沿，配置 `GTIMER->CCER.CC2P=0`。
- 选择触发输入信号，配置 `GTIMER->SMCR.TS[2:0]=110`，TI2FP2 作为 TRGI。
- 设定从模式控制器为触发模式，配置 `GTIMER->SMCR.SMS[2:0]=110`，TI2FP2 用来启动计数器。
- 打开通道使能，配置 `GTIMER->CCER.CC2E=1`。

**实现上述功能 OC1 作为输出的配置如下：**

- 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。

- 关闭通道使能，配置 GTIMER->CCER.CC1E=0，确保之后的配置成功。
- 输出通道，配置 GTIMER->CCMR1.CC1S=00。
- 选择计数有效沿，配置 GTIMER->CCMR1.OC1M=111，PWM 模式 2。
- 打开通道使能，配置 GTIMER->CCER.CC1E=1。

#### 单脉冲波形产生时基的特殊设置：

- GTIMER->CCR1 的值决定了  $t_{\text{delay}}$ 。
- GTIMER->ARR 和 GTIMER->CCR1 的差值决定了  $t_{\text{pulse}}$  (GTIMER->ARR - GTIMER->CCR1)。
- 设置为单脉冲模式，配置 GTIMER\_CR1.OPM=1。

### 14.4.13 外部事件清除 OCxREF

OCxREF 的有效状态为高电平，通过对外部 GTIMER\_ETR 引脚施加高电平，可以直接拉低 OCxREF，直到下一次 update event。此功能仅在输出比较和 PWM 模式下有效，无法在软件 force 模式下起作用。使能此功能需要将 OCxCE 置 1。

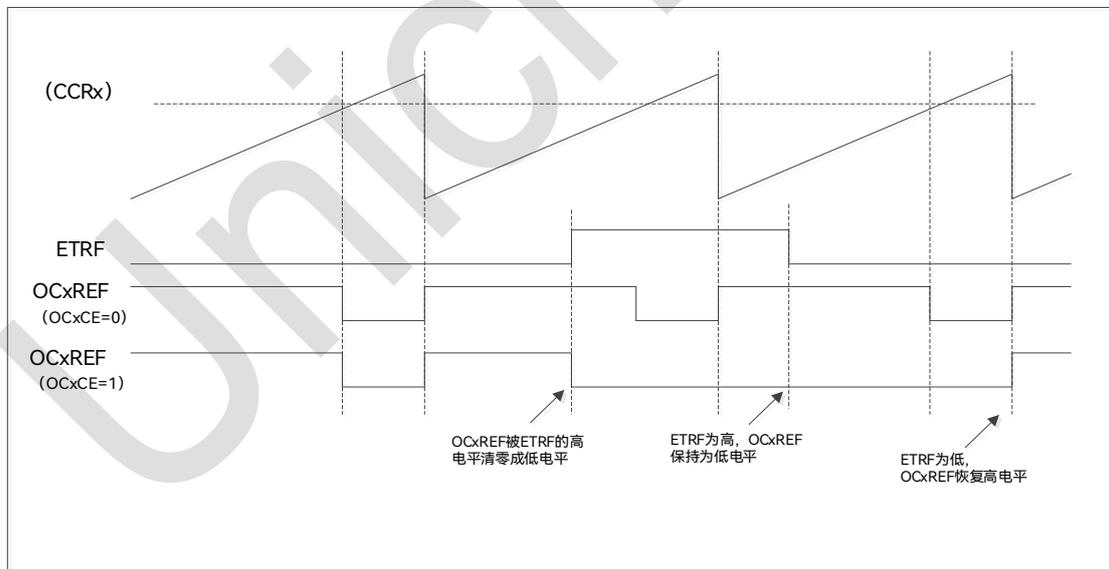


图 14-15: GTIMER\_ETR 信号清除 GTIMER 的 OCxREF

### 14.4.14 编码器接口模式 (Encoder Interface)

编码器接口模式涉及到两个外部输入信号，GTIMER 根据其中一个信号的边沿相对于另一个信

号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

表 14-2: 编码计数方式与输入信号

计数信号	TI1 计数时 TI2 电平/ TI2 计数时 TI1 电平	TI1 信号		TI2 信号	
		上升沿	下降沿	上升沿	下降沿
TI1 处计数	H	递减计数	递增计数	不计数	不计数
	L	递增计数	递减计数	不计数	不计数
TI2 处计数	H	不计数	不计数	递增计数	递减计数
	L	不计数	不计数	递减计数	递增计数
TI1 和 TI2 处计数	H	递减计数	递增计数	递增计数	递减计数
	L	递增计数	递减计数	递减计数	递增计数

比如在计数器以 TI1 信号为时钟计数时，如果 TI1 上升沿采样到 TI2 为高电平，则计数器递减；如果 TI1 下降沿采样到 TI2 为高电平，则计数器递增。

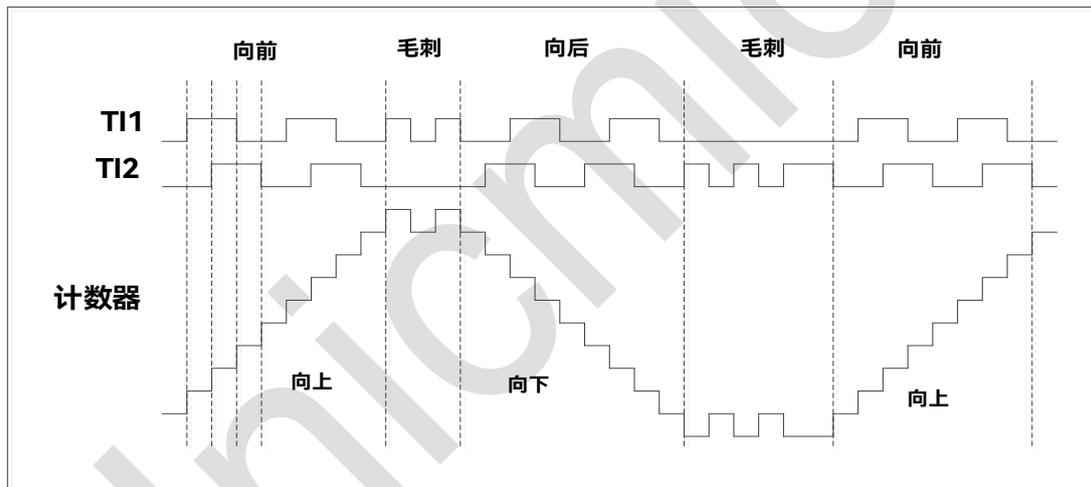


图 14-16: 编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1，GTIMER\_CH2 功能。
- 关闭通道使能，配置 GTIMER->CCER.CC1E=0，GTIMER->CCER.CC2E=0，确保之后通道配置成功。
- 选择输入通道，配置 GTIMER->CCMR1.CC1S=01，GTIMER->CCMR1.CC2S=01。
- 选择计数有效沿，配置 GTIMER->CCER.CC1P=0，GTIMER->CCER.CC2P=0。
- 设定从模式控制器为编码模式 3，配置 GTIMER->SMCR.SMS[2:0]=011。

## 14.4.15 GTIMER 从机模式

GTIMER 作为 slave 时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

### 14.4.15.1 复位模式

此模式下，外部输入的事件将导致 GTIMER 内部所有 preload 寄存器重新初始化，CNT 回到 0 开始计数。计数器正常计数，外部 TI1 输入上升沿时，触发计数器清零，重新开始计数。其配置流程如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 关闭通道使能，配置 GTIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 GTIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 GTIMER\_CCER.CC1P=0。
5. 选择触发输入信号，配置 GTIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
6. 设定从模式控制器为复位模式，配置 GTIMER\_SMCR.SMS[2:0]=100。
7. 打开通道使能，配置 GTIMER\_CCER.CC1E=1。
8. 使能计数器，配置 GTIMER\_CR1.CEN=1。

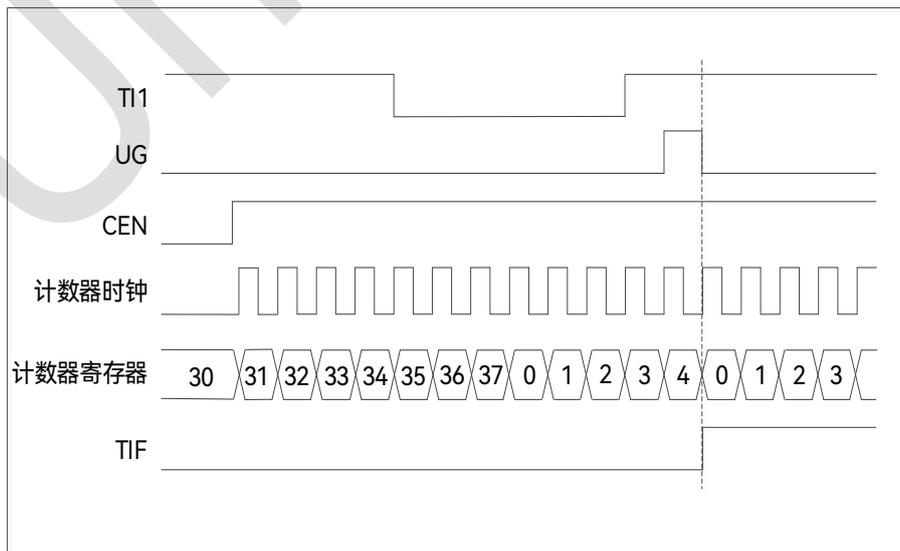


图 14-17: 复位模式下的时序

### 14.4.15.2 门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。时序配置如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 关闭通道使能，配置 GTIMER\_CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 GTIMER\_CCMR1.CC1S=01。
4. 选择计数有效沿，配置 GTIMER\_CCER.CC1P=1。
5. 选择触发输入信号，配置 GTIMER\_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
6. 设定从模式控制器为门控模式，配置 GTIMER\_SMCR.SMS[2:0]=101。
7. 打开通道使能，配置 GTIMER\_CCER.CC1E=1。
8. 使能计数器，配置 GTIMER\_CR1.CEN=1。

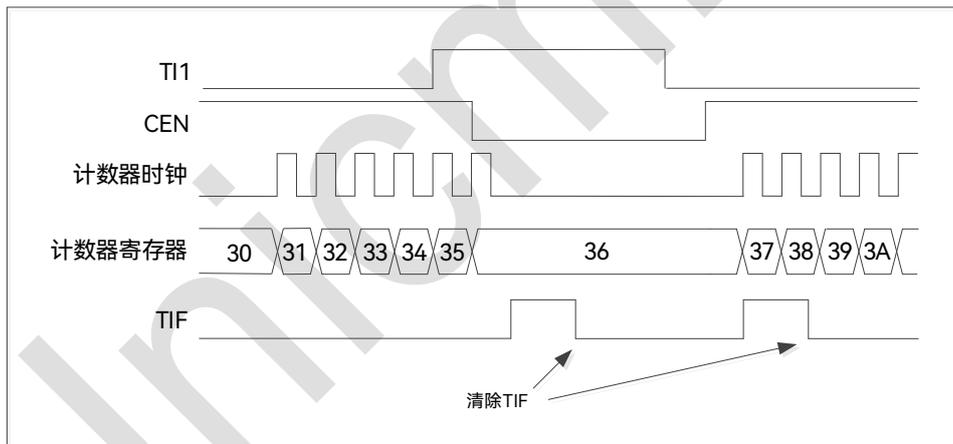


图 14-18: 门控模式下的时序图

### 14.4.15.3 触发模式

计数器在外部输入的某个事件到来后才开始计数。所示时序的流程配置如下：

1. 在 GPIO 模块中，配置相应管脚为 GTIMER\_CH1 功能。
2. 关闭通道使能，配置 GTIMER->CCER.CC1E=0 确保之后通道配置成功。
3. 选择输入通道，配置 GTIMER->CCMR1.CC1S=01。

4. 选择计数有效沿，配 GTIMER->CCER.CC1P=0。
5. 选择触发输入信号，配置 GTIMER->SMCR.TS[2:0]=101，TI1FP1 作为 TRGI。
6. 设定从模式控制器为触发模式，配置 GTIMER->SMCR.SMS[2:0]=110。
7. 打开通道使能，配置 GTIMER->CCER.CC1E=1。

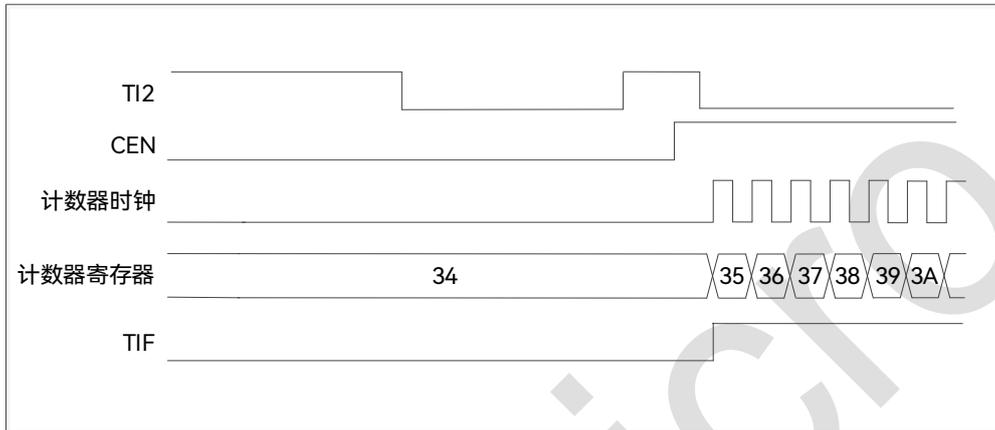


图 14-19: 触发器模式下的时序

#### 14.4.16 定时器同步

定时器之间可以通过触发事件级联起来，实现定时器同步或级联。通用定时器可以通过另外 2 个通用定时器或高级定时器的输出来触发计数，而通用定时器的触发输出则可以接到其他定时器的内部触发输入上。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

#### 14.4.17 DMA Burst

GTIMER 支持 DMA 和 DMA-Burst 访问，可以配置 GTIMER 在特定事件发生时触发 DMA 请求，可以将 CCR 中的捕捉结果写入 RAM，或者从 RAM 中将一个或多个寄存器内容写入 GTIMER 的 preload 寄存器中。

DMA-Burst 支持一个事件触发连续多次 DMA 请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA 控制器需将外设目标地址指向一个虚拟寄存器 GTIMER\_DMAR。在特定的定时器事件发

生时, GTIMER 会连续发射多个 DMA 请求。每个 DMA 对 GTIMER\_DMAR 的写操作都会被 GTIMER 重新定向到实际的功能寄存器上。

DBL 寄存器用于设置 DMA burst 长度, DBA 寄存器用于设置 DMA 访问 GTIMER 内部的地址 (相对于 GTIMER\_CR 的 offset)。

### 14.4.18 输入异或功能

通道 1~3 的输入信号可以被异或起来之后, 接入到通道 1 的滤波和边沿电路输入, 用于通道 1 的输入捕捉或者触发。GTIMER\_CR2 寄存器的 TI1S 位用于选择通道 1 的输入是否来自于三个通道输入的异或。

### 14.4.19 Debug 模式

当 Cortex-M0 进入 debug 模式后, 定时器可以停止或继续工作, 其行为由 STOP\_EN 寄存器定义。

## 14.5 寄存器描述

### 14.5.1 GTIMER 寄存器分布

GTIMER1 寄存器基地址: 0x40003400

GTIMER2 寄存器基地址: 0x40003800

表 14-3: GTIMER 寄存器列表

偏移地址	寄存器	描述
0x00	GTIMER_CR1	GTIM 控制寄存器 1
0x04	GTIMER_CR2	GTIM 控制寄存器 2
0x08	GTIMER_SMCR	GTIM 从机模式控制寄存器
0x0C	GTIMER_DIER	GTIM 中断和 DMA 使能寄存器
0x10	GTIMER_SR	GTIM 状态寄存器
0x14	GTIMER_EGR	GTIM 事件产生寄存器
0x18	GTIMER_CCMR1	GTIM 捕捉/比较模式寄存器 1
0x1C	GTIMER_CCMR2	GTIM 捕捉/比较模式寄存器 2

偏移地址	寄存器	描述
0x20	GTIMER_CCER	GTIM 捕捉/比较使能寄存器
0x24	GTIMER_CNT	GTIM 计数寄存器
0x28	GTIMER_PSC	GTIM 预分频寄存器
0x2C	GTIMER_ARR	GTIM 自动重载寄存器
0x34	GTIMER_CCR1	GTIM 捕捉/比较寄存器 1
0x38	GTIMER_CCR2	GTIM 捕捉/比较寄存器 2
0x3C	GTIMER_CCR3	GTIM 捕捉/比较寄存器 3
0x40	GTIMER_CCR4	GTIM 捕捉/比较寄存器 4
0x48	GTIMER_DCR	GTIM DMA 控制寄存器
0x4C	GTIMER_DMAR	GTIM DMA 访问寄存器

## 14.5.2 控制寄存器 1 GTIMER\_CR1 (偏移: 00h)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留
10	STOP_EN	RW	0x1	断点使能位, 在软件设置断点后, 禁止计数器计数: 0: 计数器正常工作 1: 计数器停止计数
9:8	CKD	RW	0x0	数字滤波时钟频率分频寄存器 (相对 CK_INT 的分频比): 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2*t_{CK\_INT}$ 10: $t_{DTS}=4*t_{CK\_INT}$ 11: RFU, 禁止使用
7	ARPE	RW	0x0	Auto-reload 预装载使能: 0: ARR 寄存器不使能 preload 1: ARR 寄存器使能 preload
6:5	CMS	RW	0x0	计数器对齐模式选择: 00: 边沿对齐模式 01: 中央对齐模式 1, 输出比较中断标志仅在计数器向下计数的过程中置位 10: 中央对齐模式 2, 输出比较中断标志仅在计数器向上计数的过程中置位 11: 中央对齐模式 3, 输出比较中断标志在计数器向上向下计数的过程中都会置位
4	DIR	RW	0x0	计数方向寄存器: 0: 向上计数 1: 向下计数 注: 当定时器配置为中央计数模式或编码器模式时, 此寄存器只读。
3	OPM	RW	0x0	单脉冲输出模式: 0: Update Event 发生时计数器不停止 1: Update Event 发生时计数器停止 (自动清零 CEN)
2	URS	RW	0x0	更新请求选择: 0: 以下事件能够产生 update 中断或 DMA 请求 <ul style="list-style-type: none"> <li>● 计数器上溢出或下溢出</li> <li>● 软件置位 UG 寄存器</li> <li>● 从机控制器产生 update</li> </ul> 1: 仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求

比特	名称	属性	复位值	描述
1	UDIS	RW	0x0	禁止 update: 0: 使能 update 事件; 以下事件发生时产生 update 事件 <ul style="list-style-type: none"> <li>● 计数器上溢出或下溢出</li> <li>● 软件置位 UG 寄存器</li> <li>● 从机控制器产生 update</li> </ul> 1: 禁止 update 事件, 不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	CEN	RW	0x0	计数器使能: 0: 计数器关闭 1: 计数器使能 注: 外部触发模式可以自动置位 CEN。

### 14.5.3 控制寄存器 2 GTIMER\_CR2 (偏移: 04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	TI1S	RW	0x0	GTIMER 输入 TI1 选择: 0: GTIMER_CH1 引脚连接到 TI1 输入 1: GTIMER_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	MMS	RW	0x0	主机模式选择, 用于配置主机模式下向从机发送的同步触发信号 (TRGO) 源: 000: GTIMER_EGR 的 UG 寄存器被用作 TRGO 001: 计数器使能信号 CNT_EN 被用作 TRGO, 可用于同时启动多个定时器 010: UE (update event) 信号被用作 TRGO 011: 比较脉冲, 如果 CC1IF 标志将要置位, TRGO 输出一个正脉冲 100: OC1REF 用作 TRGO 101: OC2REF 用作 TRGO 110: OC3REF 用作 TRGO 111: OC4REF 用作 TRGO 注: 从机定时器或 ADC 必须事先使能工作时钟, 才能接收主机定时器发送的 TRGO。
3	CCDS	RW	0x0	捕捉/比较 DMA 选择: 0: 捕捉/比较事件发生时发送 DMA 请求 1: Update Event 发生时发送 DMA 请求
2:0	RSV	-	-	保留

## 14.5.4 从机模式控制寄存器 GTIMER\_SMCR (偏移: 08h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	ETP	RW	0x0	外部触发信号极性配置: 0: 高电平或上升沿有效 1: 低电平或下降沿有效
14	ECE	RW	0x0	外部时钟使能: 0: 关闭外部时钟模式 2 1: 使能外部时钟模式2, 计数器时钟为ETRF有效沿
13:12	ETPS	RW	0x0	外部触发信号预分频寄存器: 外部触发信号 ETRP 的频率最多只能是 GTIMER 工作时钟的 1/4, 当输入信号频率较高时, 可以使用预分频。 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频
11:8	ETF	RW	0x0	外部触发信号滤波时钟和长度选择: 0000: 无滤波 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: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=5$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=6$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=8$
7	MSM	RW	0x0	主/从模式选择: 0: 无动作, 以便于通过 TRGO 将主机定时器和从机定时器完美同步起来 (完美同步时主机寄存器 CR2.MMS 需配为 010)

比特	名称	属性	复位值	描述
				1: 延迟主机 TRGI 触发信号 注: 仅用于从机模式的触发模式 (SMS=110), 且由从机模式配置。
6:4	TS	RW	0x0	触发选择, 用于选择同步计数器的触发源: 000: 内部触发信号 (ITR0) 001: 内部触发信号 (ITR1) 010: 内部触发信号 (ITR2) 011: 内部触发信号 (ITR3) 100: TI1 边沿检测 (TI1F_ED) 101: 滤波后 TI1 (TI1FP1) 110: 滤波后 TI2 (TI2FP2) 111: 外部触发输入 (ETRF) 注: 仅当 SMS=000 即禁止从机模式的情况下, 可以改写 TS 寄存器。
3	RSV	-	-	保留
2:0	SMS	RW	0x0	从机模式选择: 000: 从机模式禁止; CEN 使能后预分频电路时钟源来自内部时钟 001: Encoder 模式 1; 计数器使用 TI2FP2 边沿, 根据 TI1FP1 电平高低来计数 010: Encoder 模式 2; 计数器使用 TI1FP1 边沿, 根据 TI2FP2 电平高低来计数 011: Encoder 模式 3; 计数器同时使用 TI1FP1 和 TI2FP2 边沿, 根据其他输入信号电平来计数 100: 复位模式; TRGI 上升沿初始化计数器, 并触发寄存器 update 101: 闸门模式; TRGI 为高电平时, 计数时钟使能, TRGI 为低电平时, 计数时钟停止 110: 触发模式; TRGI 上升沿触发计数器开始计数 (不会复位计数器) 111: 外部时钟模式 1; TRGI 上升沿直接驱动计数器

### 14.5.5 DMA 和中断使能寄存器 GTIMER\_DIER (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
19	CC4BUEN	RW	0x1	捕捉比较通道 4 的 DMA 模式配置: 0: Single 传输, DMA 读取 CCR4 寄存器的值传输到 RAM

比特	名称	属性	复位值	描述
				1: RAM->GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
18	CC3BUEN	RW	0x1	捕捉比较通道 3 的 DMA 模式配置: 0: CCR3->RAM, DMA 读取 CCR3 寄存器的值传输到 RAM 1: RAM->GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
17	CC2BUEN	RW	0x1	捕捉比较通道 2 的 DMA 模式配置: 0: CCR2->RAM, DMA 读取 CCR2 寄存器的值传输到 RAM 1: RAM->GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
16	CC1BUEN	RW	0x1	捕捉比较通道 1 的 DMA 模式配置: 0: CCR1->RAM, DMA 读取 CCR1 寄存器的值传输到 RAM 1: RAM->GTIMER 的 Burst 模式, 通过 DCR 配置访问的地址和长度
15	RSV	-	-	保留
14	TDE	RW	0x0	外部触发 DMA 请求使能: 0: 从机模式下, 禁止外部触发事件产生 DMA 请求 1: 从机模式下, 允许外部触发事件产生 DMA 请求 (可用于自动更新 preload 寄存器)
13	RSV	-	-	保留
12	CC4DE	RW	0x0	捕捉比较通道 4 的 DMA 请求使能: 0: 禁止 CC4 DMA 请求 1: 允许 CC4 DMA 请求
11	CC3DE	RW	0x0	捕捉比较通道 3 的 DMA 请求使能: 0: 禁止 CC3 DMA 请求 1: 允许 CC3 DMA 请求
10	CC2DE	RW	0x0	捕捉比较通道 2 的 DMA 请求使能: 0: 禁止 CC2 DMA 请求 1: 允许 CC2 DMA 请求
9	CC1DE	RW	0x0	捕捉比较通道 1 的 DMA 请求使能: 0: 禁止 CC1 DMA 请求 1: 允许 CC1 DMA 请求
8	UDE	RW	0x0	Update Event DMA 请求使能: 0: Update Event 发生时, 禁止产生 DMA 请求 1: Update Event 发生时, 允许产生 DMA 请求
7	RSV	-	-	保留

比特	名称	属性	复位值	描述
6	TIE	RW	0x0	触发事件中断使能： 0: 禁止触发事件中断 1: 允许触发事件中断
5	RSV	-	-	保留
4	CC4IE	RW	0x0	捕捉/比较通道 4 中断使能： 0: 禁止捕捉/比较 4 中断 1: 允许捕捉/比较 4 中断
3	CC3IE	RW	0x0	捕捉/比较通道 3 中断使能： 0: 禁止捕捉/比较 3 中断 1: 允许捕捉/比较 3 中断
2	CC2IE	RW	0x0	捕捉/比较通道 2 中断使能： 0: 禁止捕捉/比较 2 中断 1: 允许捕捉/比较 2 中断
1	CC1IE	RW	0x0	捕捉/比较通道 1 中断使能： 0: 禁止捕捉/比较 1 中断 1: 允许捕捉/比较 1 中断
0	UIE	RW	0x0	Update 事件中断使能： 0: 禁止 Update 事件中断 1: 允许 Update 事件中断

### 14.5.6 状态寄存器 GTIMER\_SR (偏移: 10h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	CC4OF	W1C	0x0	捕捉/比较通道 4 的 Overcapture 状态 参考CC1OF
11	CC3OF	W1C	0x0	捕捉/比较通道 3 的 Overcapture 状态 参考CC1OF
10	CC2OF	W1C	0x0	捕捉/比较通道 2 的 Overcapture 状态 参考CC1OF
9	CC1OF	W1C	0x0	捕捉/比较通道 1 的 Overcapture 状态 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0: 无 overcapture 事件 1: 在CC1IF标志为1的情况下发生新的捕捉
8:7	RSV	-	-	保留

比特	名称	属性	复位值	描述
6	TIF	W1C	0x0	触发事件中断标志，硬件置位，软件写 1 清零
5	RSV	-	-	保留
4	CC4IF	W1C	0x0	捕捉/比较通道 4 中断标志 参考CC1IF
3	CC3IF	W1C	0x0	捕捉/比较通道 3 中断标志 参考CC3IF
2	CC2IF	W1C	0x0	捕捉/比较通道 2 中断标志 参考CC2IF
1	CC1IF	W1C	0x0	捕捉/比较通道 1 中断标志： 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果CC1通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读CCR1自动清零。
0	UIF	W1C	0x0	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0且UDIS=0的情况下，触发事件初始化计数器

### 14.5.7 事件产生寄存器 GTIMER\_EGR (偏移: 14h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	TG	W	0x0	软件触发，软件置位此寄存器产生触发事件，硬件自动清零
5	RSV	-	-	保留
4	CC4G	W	0x0	捕捉/比较通道 4 软件触发，参考 CC1G
3	CC3G	W	0x0	捕捉/比较通道 3 软件触发，参考 CC1G
2	CC2G	W	0x0	捕捉/比较通道 2 软件触发，参考 CC1G
1	CC1G	W	0x0	捕捉/比较通道 1 软件触发，硬件自动清零，硬件自动清零 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果CC1通道配置为输入：当前计数值被捕捉到CCR1寄存器，CC1IF置位，在使能的情况下可以产生相应的中断和DMA请求
0	UG	W	0x0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零

比特	名称	属性	复位值	描述
				软件置位UG时会重新初始化计数器并更新shadow寄存器, 预分频计数器被清零。

### 14.5.8 捕捉/比较模式寄存器 1 GTIMER\_CCMR1 (偏移: 18h)

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能:

#### 1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC2CE	RW	0x0	输出比较 2 清零使能, 参考 OC1CE
14:12	OC2M	RW	0x0	输出比较 2 模式配置, 参考 OC1M
11	OC2PE	RW	0x0	输出比较 2 预装载使能, 参考 OC1PE
10	OC2FE	RW	0x0	输出比较 2 快速使能, 参考 OC1FE
9:8	CC2S	RW	0x0	捕捉/比较 2 通道选择: 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 (如果 BPS2 置 1, 则映射到 TI2 边沿检测) 注意: CC2S仅在通道关闭时 (CC2E=0) 可以写
7	OC1CE	RW	0x0	输出比较 1 清零使能: 0: OC1REF 不受 ETRF 影响 1: 检测到ETRF高电平时, 自动清零OC1REF
6:4	OC1M	RW	0x0	输出比较 1 模式配置, 此寄存器定义 OC1REF 信号的行为: 000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001: CCR1=CNT 时, 将 OC1REF 置高 010: CCR1=CNT 时, 将 OC1REF 置低 011: CCR1=CNT 时, 翻转 OC1REF 100: OC1REF 固定为低 (inactive) 101: OC1REF 固定为高 (active) 110: PWM 模式 1——在向上计数时, OC1REF 在 CNT<CCR1 时置高, 否则置低; 在向下计数时, OC1REF 在 CNT>CCR1 时置低, 否则置高。 111: PWM模式2——在向上计数时, OC1REF在CNT<CCR1

比特	名称	属性	复位值	描述
				时置低，否则置高；在向下计数时，OC1REF在CNT>CCR1时置高，否则置低。
3	OC1PE	RW	0x0	输出比较 1 预装载使能： 0: CCR1 preload 寄存器无效，CCR1 可以直接写入 1: CCR1 preload寄存器有效，针对CCR1的读写操作都是访问preload寄存器，当update event发生时才将preload寄存器的内容转移到shadow寄存器中。
2	OC1FE	RW	0x0	输出比较 1 快速使能： 0: 关闭快速使能，trigger 输入不会影响比较输出 1: 打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况 注：此功能仅在当前通道配置为PWM1或PWM2模式时有效。
1:0	CC1S	RW	0x0	捕捉/比较 1 通道选择： 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 (如果 BPS1 置 1，则映射到 TI1 边沿检测) 注：CC1S仅在通道关闭时 (CC1E=0) 可以写。

## 2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC2F	RW	0x0	输入捕捉 2 滤波
11:10	IC2PSC	RW	0x0	输入捕捉 2 预分频
9:8	CC2S	RW	0x0	捕捉/比较 2 通道选择： 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 (如果 BPS2 置 1，则映射到 TI2 边沿检测) 注：CC2S仅在通道关闭时 (CC2E=0) 可以写。
7:4	IC1F	RW	0x0	输入捕捉 1 滤波： 此寄存器定义 TI1 的采样频率和滤波长度。 0000: 无滤波，使用 $f_{DTS}$ 采样 0001: $f_{SAMPLING}=f_{CK\_INT}$ , $N=2$

比特	名称	属性	复位值	描述
				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: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=8$
3:2	IC1PSC	RW	0x0	输入捕捉 1 预分频: 00: 无分频 01: 每 2 个事件输入产生一次捕捉 10: 每 4 个事件输入产生一次捕捉 11: 每 8 个事件输入产生一次捕捉 注: IC1PSC寄存器在CC1E=0时复位。
1:0	CC1S	RW	0x0	捕捉/比较 1 通道选择: 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 T11 10: CC1 通道配置为输入, IC1 映射到 T12 11: CC1 通道配置为输入, IC1 映射到 TRC (如果 BPS1 置 1, 则映射到 T11 边沿检测) 注: CC1S仅在通道关闭时 (CC1E=0) 可以写。

### 14.5.9 捕捉/比较模式寄存器 2 GTIMER\_CCMR2 (偏移: 1Ch)

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能。

#### 1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC4CE	RW	0x0	输出比较 4 清零使能, 参考 OC1CE

比特	名称	属性	复位值	描述
14:12	OC4M	RW	0x0	输出比较 4 模式配置, 参考 OC1M
11	OC4PE	RW	0x0	输出比较 4 预装载使能, 参考 OC1PE
10	OC4FE	RW	0x0	输出比较 4 快速使能, 参考 OC1FE
9:8	CC4S	RW	0x0	捕捉/比较 4 通道选择: 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 (如果 BPS4 置 1, 则映射到 TI4 边沿检测) 注: CC4S仅在通道关闭时 (CC4E=0) 可以写。
7	OC3CE	RW	0x0	输出比较 4 清零使能: 0: OC4REF 不受 ETRF 影响 1: 检测到ETRF高电平时, 自动清零OC4REF
6:4	OC3M	RW	0x0	输出比较 3 模式配置, 此寄存器定义 OC3REF 信号的行为: 000: 输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001: CCR3=CNT 时, 将 OC3REF 置高 010: CCR3=CNT 时, 将 OC3REF 置低 011: CCR3=CNT 时, 翻转 OC3REF 100: OC3REF 固定为低 (inactive) 101: OC3REF 固定为高 (active) 110: PWM 模式 1——在向上计数时, OC3REF 在 CNT<CCR3 时置高, 否则置低; 在向下计数时, OC3REF 在 CNT>CCR3 时置低, 否则置高。 111: PWM模式2——在向上计数时, OC3REF在CNT<CCR3 时置低, 否则置高; 在向下计数时, OC3REF在CNT>CCR3 时置高, 否则置低。
3	OC3PE	RW	0x0	输出比较 3 预装载使能: 0: CCR3 preload 寄存器无效, CCR4 可以直接写入 1: CCR3 preload寄存器有效, 针对CCR4的读写操作都是访问 preload寄存器, 当update event发生时才将preload寄存器的内容转移到shadow寄存器中。
2	OC3FE	RW	0x0	输出比较 3 快速使能: 0: 关闭快速使能, trigger 输入不会影响比较输出 1: 打开快速使能, trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出, 而不管当前实际比较情况。 注: 此功能仅在当前通道配置为PWM1或PWM2模式时有效。

比特	名称	属性	复位值	描述
1:0	CC3S	RW	0x0	捕捉/比较 3 通道选择: 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 (如果 BPS3 置 1, 则映射到 TI3 边沿检测) 注: CC3S仅在通道关闭时 (CC3E=0) 可以写。

## 2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC4F	RW	0x0	输入捕捉 4 滤波
11:10	IC4PSC	RW	0x0	输入捕捉 4 预分频
9:8	CC4S	RW	0x0	捕捉/比较 4 通道选择: 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 (如果 BPS4 置 1, 则映射到 TI4 边沿检测) 注: CC4S仅在通道关闭时 (CC4E=0) 可以写。
7:4	IC3F	RW	0x0	输入捕捉 3 滤波: 此寄存器定义 TI3 的采样频率和滤波长度。 0000: 无滤波, 使用 $f_{DTS}$ 采样 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: $f_{SAMPLING}=f_{DTS}/32$ , $N=5$

比特	名称	属性	复位值	描述
				1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=6$ 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32, N=8$
3:2	IC3PSC	RW	0x0	输入捕捉 3 预分频: 00: 无分频 01: 每 2 个事件输入产生一次捕捉 10: 每 4 个事件输入产生一次捕捉 11: 每 8 个事件输入产生一次捕捉 注: IC3PSC寄存器在CC3E=0时复位。
1:0	CC3S	RW	0x0	捕捉/比较 3 通道选择: 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 (如果 BPS3 置 1, 则映射到 TI3 边沿检测) 注: CC3S仅在通道关闭时 (CC1E=0) 可以写。

#### 14.5.10 捕捉/比较使能寄存器 GTIMER\_CCER (偏移: 20h)

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留
13	CC4P	RW	0x0	捕捉/比较 4 输出极性, 参考 CC1P
12	CC4E	RW	0x0	捕捉/比较 4 输出使能, 参考 CC1E
11:10	RSV	-	-	保留
9	CC3P	RW	0x0	捕捉/比较 3 输出极性, 参考 CC1P
8	CC3E	RW	0x0	捕捉/比较 3 输出使能, 参考 CC1E
7:6	RSV	-	-	保留
5	CC2P	RW	0x0	捕捉/比较 2 输出极性, 参考 CC1P
4	CC2E	RW	0x0	捕捉/比较 2 输出使能, 参考 CC1E
3:2	RSV	-	-	保留
1	CC1P	RW	0x0	捕捉/比较 1 输出极性: CC1 通道配置为输出时: 0: OC1 高电平 active 1: OC1 低电平 active CC1 通道配置为输入时: 0: 非取反模式-捕捉在 IC1 的上升沿进行 1: 取反模式-捕捉在 IC1 的下降沿进行

比特	名称	属性	复位值	描述
0	CC1E	RW	0x0	捕捉/比较 1 输出使能： CC1 通道配置为输出时： 0: OC1 不 active 1: OC1 active CC1 通道配置为输入时： 0: 关闭捕捉功能 1: 使能捕捉功能

#### 14.5.11 计数器寄存器 GTIMER\_CNT (偏移: 24h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT	RW	0x0	计数值

#### 14.5.12 预分频寄存器 GTIMER\_PSC (偏移: 28h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC	RW	0x0	计数器时钟 (CK_CNT) 预分频值 $f_{CK\_CNT} = f_{CK\_PSC} / (PSC[15:0] + 1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器。 注: 支持的最高的PWM输出为30MHz, 配置的PSC和ARR需注意该条件。

#### 14.5.13 自动重载寄存器 GTIMER\_ARR (偏移: 2Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR	RW	0x0	计数溢出时的自动重载值 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器。

**14.5.14 捕捉/比较寄存器 1 GTIMER\_CCR1 (偏移: 34h)**

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR1	RW	0x0	捕捉/比较通道1寄存器 <b>如果通道1配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC1输出。 <b>如果通道1配置为输入:</b> CCR1保存最近一次输入捕捉事件发生时的计数器值, 此时CCR1为只读。

**14.5.15 捕捉/比较寄存器 2 GTIMER\_CCR2 (偏移: 38h)**

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR2	RW	0x0	捕捉/比较通道2寄存器 <b>如果通道2配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC2输出。 <b>如果通道2配置为输入:</b> CCR2保存最近一次输入捕捉事件发生时的计数器值, 此时CCR2为只读。

**14.5.16 捕捉/比较寄存器 3 GTIMER\_CCR3 (偏移: 3Ch)**

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR3	RW	0x0	捕捉/比较通道3寄存器 <b>如果通道3配置为输出:</b> 这是一个preload寄存器, 其内容被载入shadow寄存器后用于与计数器比较产生OC3输出。 <b>如果通道3配置为输入:</b> CCR3保存最近一次输入捕捉事件发生时的计数器值, 此时CCR3为只读。

### 14.5.17 捕捉/比较寄存器 4 GTIMER\_CCR4 (偏移: 40h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR4	RW	0x0	捕捉/比较通道4寄存器 <b>如果通道4配置为输出:</b> 这是一个preload寄存器,其内容被载入shadow寄存器后用于与计数器比较产生OC4输出。 <b>如果通道4配置为输入:</b> CCR4保存最近一次输入捕捉事件发生时的计数器值,此时CCR4为只读。

### 14.5.18 DMA 控制寄存器 GTIMER\_DCR (偏移: 48h)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12:8	DBL	RW	0x0	DMA Burst长度: 对DMAR寄存器的读写将触发burst DMA操作, burst长度为1~18 00000: 长度=1 00001: 长度=2 ..... 10001: 长度=18 其他: 无效值, 禁止写入
7:5	RSV	-	-	保留
4:0	DBA	RW	0x0	DMA基地址, 定义指向寄存器的偏移地址: 00000: CR1 00001: CR2 00010: SMCR ..... 01011: ARR 01100: RSV 01101: CCR1 ..... 10000: CCR4 10001: RSV 11000: DCR

比特	名称	属性	复位值	描述
				注意：当DBA+DBL超出了GTIMER寄存器地址范围，则实际burst传输到GTIMER最高寄存器地址后自动停止，即burst长度会缩短。

#### 14.5.19 DMA 访问寄存器 GTIMER\_DMAR (偏移: 4Ch)

比特	名称	属性	复位值	描述
31:0	DMAR	RW	0x400	DMA burst访问寄存器： 在使用DMA burst传输时，将DMA通道外设地址设置为DMAR，GTIMER会根据DBL的值产生多次DMA请求。

# 15 LPTIMER

## 15.1 概述

LPTIMER 是一个 32 位的低功耗定时/计数器模块。由于其时钟源具有多样性，因此能够在所有电源模式下保持运行状态，并且只消耗很低的功耗。LPTIMER 可以在没有内部时钟的条件下工作，实现休眠模式下的外部脉冲计数功能，还可以与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。

## 15.2 主要特性

- 独立的 32 位向上计数器
- 3 位异步时钟预分频器，8 种分频系数 (1、2、4、8、16、32、64、128)
- 可选工作时钟：
  - 内部时钟源：LSCLK (CLK32K)、RCLP (LVD)、PCLK
  - 外部时钟源：LPTIM\_IN (带有模拟滤波)、COMP\_IN (COMP0~COMP1)
- 32 位比较/捕捉寄存器
- 32 位目标值寄存器
- 连续/单触发模式
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 支持 PWM 输出

## 15.3 结构框图

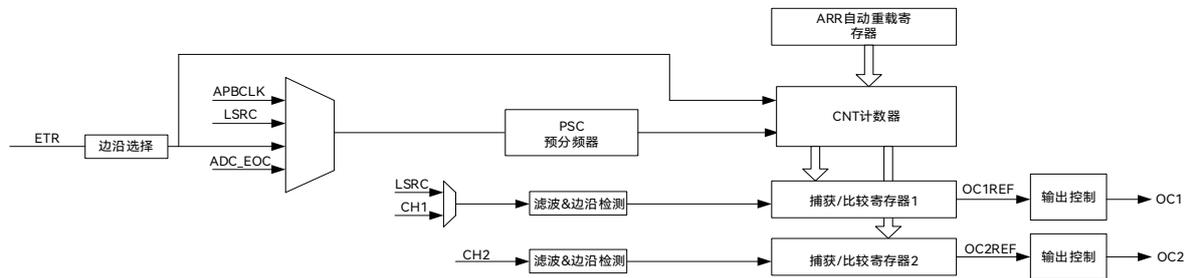


图 15-1: LPTIMER 结构框图

## 15.4 定时器功能

LPTIM 工作模式：计数功能、ETR 触发计数、ETR 脉冲计数、Timeout 模式。在 LPTIM 计数工作模式下软件流程如下：

- 选择工作时钟
- 设置 ARR 寄存器
- 配置 CR1 的 EN 使能位
- 等待计数器值等于 ARR，产生 UIF 中断，计数器值返回 0

### 15.4.1 单次计数和连续计数

LPTIM 有两种计数模式：单次计数和连续计数。

单次计数模式：计数器被触发后计数到 ARR 后回到 0，并自动停止，产生溢出中断，同时硬件自动清除 EN。

连续计数模式：计数器启动后保持运行，直到被关闭为止。计数器达到 ARR 后回到 0 重新开始计数，并产生溢出中断。

### 15.4.2 ETR 触发计数

外部 ETR 脉冲触发计数模式下，LPTIM 将 LPTIM\_ETR 引脚输入的信号作为触发信号使用。

LPTIM\_ETR 信号首先经过 LPTIM 工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发定时器递增。

### 15.4.3 ETR 脉冲计数

外部 ETR 脉冲计数模式下，LPTIM 将 ETR 引脚输入的信号直接作为计数时钟使用。软件可以选择计数沿和分频值。

注意：和 ETR 触发计数模式不同的是，当用 ETR 触发计数工作时，如果触发时钟非常快，经 LPTIM 工作时钟并没有采到，那 ETR 触发计数模式将不计数，而 ETR 脉冲模式不需要采样触发，作为工作时钟可以计数。

### 15.4.4 Timeout 模式

Timeout 模式下，LPTIM 将 ETR 引脚输入的信号作为触发信号使用，定时器使用内部时钟 CLK\_LPT 工作。Timeout 模式下定时器使能后，等待第一个 ETR 信号的有效沿到来，触发定时器开始计数，此后每个新的 ETR 有效沿都会清零计数器，并重新开始计数。如果定时器出现溢出，则定时器产生溢出中断，计数值回到 0，并自动清除 EN 结束计数过程。

## 15.5 捕捉比较功能

LPTIM 有两路捕捉和比较功能，当采用捕捉功能时，可以捕捉外部或内部信号的周期长度。当采用比较功能时，可以根据不同的比较值输出对应的 PWM 波形。

### 15.5.1 PWM 输出

LPTIM 的两个捕捉/比较通道都可以输出 32 位 PWM 波形。PWM 功能需要将捕捉/比较通道配置为比较输出。使能 PWM 功能后 LPTIM 从 0 开始计数，计数值等于 CCR 时输出置高，计数值等于 ARR 时输出变低；PWM 周期由 ARR 寄存器决定，占空比由 CCRx 寄存器决定。软件可以配置输出波形的极性。

## 15.5.2 输入捕捉

LPTIM 的输入捕捉可以配置为针对输入信号的上升沿、下降沿或上升下降沿进行捕捉，把捕捉到的 CNT 计数值存入 CCRx 寄存器并产生捕捉中断。

## 15.5.3 Debug 模式

Cortex-M0 进入 debug 模式后, 定时器可以停止或继续工作, 其行为由 STOP\_EN 寄存器定义。

## 15.6 寄存器描述

LPTIMER 寄存器基地址: 0x40001000

表 15-1: LPTIMER 寄存器列表

偏置	名称	描述
0x00	LPTIMER_CR1	LPTIMER 控制寄存器 1
0x04	LPTIMER_CR2	LPTIMER 控制寄存器 2
0x08	LPTIMER_IER	LPTIMER 中断使能寄存器
0x0C	LPTIMER_SR	LPTIMER 中断标志寄存器
0x10	LPTIMER_CNT	LPTIMER 计数值寄存器
0x14	LPTIMER_CCMCFG1	LPTIMER 捕捉比较配置寄存器 1
0x18	LPTIMER_CCMCFG2	LPTIMER 捕捉比较配置寄存器 2
0x1C	LPTIMER_ARR	LPTIMER 自动重装载寄存器 1
0x20	LPTIMER_CCR1	LPTIMER 捕捉比较寄存器 1
0x24	LPTIMER_CCR2	LPTIMER 捕捉比较寄存器 2
0x28	LPTIMER_LOAD	LPTIMER 计数值 load 寄存器 1
0x2C	LPTIMER_BUFFER	LPTIMER 计数缓存寄存器 1

### 15.6.1 LPTIMER 控制寄存器 1 LPTIMER\_CR1 (偏移: 00h)

比特	名称	属性	默认值	功能描述
31:5	RSV	-	-	保留
4	STOP_EN	RW	0x1	LPTIMER断点使能位, 在软件设置断点后, 禁止计数器计数: 0: 计数器正常工作 1: 计数器停止计数

比特	名称	属性	默认值	功能描述
3:1	RSV	-	-	保留
0	EN	RW	0x0	LPTIMER使能位： 1: 使能计数器计数 0: 禁止计数器计数

### 15.6.2 LPTIMER 控制寄存器 2 LPTIMER\_CR2 (偏移: 04h)

比特	名称	属性	默认值	功能描述
31:15	RSV	-	-	保留
14:12	CAP1SRSEL	RW	0x0	通道1捕捉信号源选择： 000: LPTIMER_CH1输入 001: RCLP 010: UART0_RX 011: UART1_RX 100: I2C_SCL 101: SPI_SCK 110: COMP0_IN1 111: COMP1_IN1
11:9	DIVCFG	RW	0x0	计数时钟分频选择： 000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频
8:6	CLKSEL	RW	0x0	时钟源选择： 000: LSCLK (由SYSREG->SYSCTRL0[12]选择的系统低速时钟CLK32K, 即RCL或XTL) 作为计数时钟 001: RCLP作为计数时钟 (LVD) 010: PCLK的门控时钟 (PCLK) 作为计数时钟 011: ETR输入作为计数时钟 100: COMP0_IN 101: COMP1_IN 其它: COMP0_IN

比特	名称	属性	默认值	功能描述
5	EDGESEL	RW	0x0	ETR计数时钟输入边沿选择: 0: ETR的上升沿计数 1: ETR的下降沿计数
4	SINGLE	RW	0x0	单次计数模式使能: 0: 连续计数模式: 计数器被触发后保持运行, 直到被关闭为止。计数器达到目标值后回到0重新开始计数, 并产生溢出中断。 1: 单次计数模式: 计数器被触发后计数到目标值后回到0, 并自动停止, 产生溢出中断。
3:2	TRIGEDGE	RW	0x0	ETR触发边沿选择: 00: ETR输入信号上升沿触发 01: ETR输入信号下降沿触发 10/11: ETR输入信号上升下降沿触发
1:0	LPMOD	RW	0x0	工作模式选择: 00: 计数模式 01: ETR脉冲触发计数模式 10: ETR脉冲计数模式 11: Timeout模式

### 15.6.3 LPTIMER 中断使能寄存器 LPTIMER\_IER (偏移: 08h)

比特	名称	属性	默认值	功能描述
31:4	RSV	-	-	保留
3	TIE	RW	0x0	LPTIMER触发事件中断使能: 0: 禁止触发事件中断 1: 使能触发事件中断
2	CC2IE	RW	0x0	LPTIMER捕捉/比较通道2中断使能: 0: 禁止捕捉/比较通道2中断 1: 使能捕捉/比较通道2中断
1	CC1IE	RW	0x0	LPTIMER捕捉/比较通道1中断使能: 0: 禁止捕捉/比较通道1中断 1: 使能捕捉/比较通道1中断
0	UIE	RW	0x0	LPTIMER更新中断使能: 0: 禁止更新事件中断 1: 使能更新事件中断

### 15.6.4 LPTIMER 中断标志寄存器 LPTIMER\_SR (偏移: 0Ch)

比特	名称	属性	默认值	功能描述
31:10	RSV	-	-	保留
9	LPTIM_CH2	R	0x0	LPTIMER捕捉通道2电平状态
8	LPTIM_CH1	R	0x0	LPTIMER捕捉通道1电平状态
7:4	RSV	-	-	保留
3	TIF	W1C	0x0	LPTIMER触发中断标志, 硬件置位, 软件写1清零 0: 无触发事件 1: 触发中断挂起
2	CC2IF	W1C	0x0	LPTIMER捕捉/比较通道2中断标志, 硬件置位, 软件写1清零 0: 无比较或捕捉中断产生 1: CNT值和CCR2相等, 或者发生输入捕捉事件
1	CC1IF	W1C	0x0	LPTIMER捕捉/比较通道1中断标志, 硬件置位, 软件写1清零 0: 无比较或捕捉中断产生 1: CNT值和CCR1相等, 或者发生输入捕捉事件
0	UIF	W1C	0x0	LPTIMER更新中断标志, 硬件置位, 软件写1清零 0: 无更新中断产生 1: CNT值等于ARR值产生中断

### 15.6.5 LPTIMER 计数值寄存器 LPTIMER\_CNT (偏移: 10h)

比特	名称	属性	默认值	功能描述
31:0	CNT	R	0x0	计数器计数值

### 15.6.6 LPTIMER 捕捉比较配置寄存器 1 LPTIMER\_CCMCFG1 (偏移: 14h)

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5:4	CAP1EDGE	RW	0x0	通道1捕捉边沿选择: 00: 上升沿捕捉 01: 下降沿捕捉

比特	名称	属性	默认值	功能描述
				10: 上升下降沿捕捉 11: 未定义
3	RSV	-	-	保留
2	CC1P	RW	0x0	通道1比较输出波形极性选择: 0: CNT≤CCR1时置低, CNT>CCR1时为高 1: CNT≤CCR1时置高, CNT>CCR1时为低
1	CC1S	RW	0x0	通道1捕捉/比较输出选择: 0: 通道1配置为输出 1: 通道1配置为输入
0	CC1E	RW	0x0	通道1捕捉使能: 0: 通道1捕获功能禁止 1: 通道1捕获功能使能

### 15.6.7 LPTIMER 捕捉比较配置寄存器 2 LPTIMER\_CCMCFG2 (偏移: 18h)

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5:4	CAP2EDGE	RW	0x0	通道2捕捉边沿选择: 00: 上升沿捕捉 01: 下降沿捕捉 10: 上升下降沿捕捉 11: 未定义
3	RSV	-	-	保留
2	CC2P	RW	0x0	通道2输出极性选择: 0: CNT≤CCR2时置低, CNT>CCR2时为高 1: CNT≤CCR2时置高, CNT>CCR2时为低
1	CC2S	RW	0x0	通道2捕捉/比较选择: 0: 通道2配置为输出 1: 通道2配置为输入
0	CC2E	RW	0x0	通道2捕捉使能: 0: 通道2捕获功能禁止 1: 通道2捕获功能使能

### 15.6.8 LPTIMER 自动重载置寄存器 LPTIMER\_ARR (偏移: 1Ch)

比特	名称	属性	默认值	功能描述
31:0	ARR	RW	0x0	自动重载目标寄存器: 当计数器计数值等于ARR时, 计数器回到0。 注: 支持的最高的PWM输出为12MHz, 配置的DIVCFG和ARR需注意该条件。

### 15.6.9 LPTIMER 捕捉比较寄存器 1 LPTIMER\_CCR1 (偏移: 20h)

比特	名称	属性	默认值	功能描述
31:0	CCR1	RW	0x0	捕捉/比较值寄存器1: 当ARR=CCR1时, 以CCR1为准。

### 15.6.10 LPTIMER 捕捉比较寄存器 2 LPTIMER\_CCR2 (偏移: 24h)

比特	名称	属性	默认值	功能描述
31:0	CCR2	RW	0x0	捕捉/比较值寄存器2: 当ARR=CCR2时, 以CCR2为准。

### 15.6.11 LPTIMER 计数值 load 寄存器 LPTIMER\_LOAD (偏移: 28h)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	LOAD	RW	0x0	1: 表示正在进行LOAD CNT的操作 0: 表示LOAD CNT操作结束

### 15.6.12 LPTIMER 计数缓存寄存器 LPTIMER\_BUFFER (偏移: 2Ch)

比特	名称	属性	默认值	功能描述
31:0	BUFFER	R	0x0	CNT缓存寄存器: 寄存器用于当软件发出LOAD指令后, 存储计数器CNT的当前值

## 15.7 使用流程

下文详细介绍软件使用流程。

### 15.7.1 普通定时器

1. 初始化 LPTIMER 时钟模块。
2. 配置 LPTIMER\_CR2.LPMOD，选择工作模式。
3. 配置 LPTIMER\_CR2.SINGLE，设置计数模式。
4. 配置 LPTIMER\_CR2.DIVCFG，设置分频值。
5. 配置 LPTIMER\_CR2.CLKSEL，设置时钟源。
6. 配置 LPTIMER\_ARR1 目标寄存器值。
7. 使能 LPTIMER\_IER 中断寄存器，选择溢出中断。
8. 使能 LPTIMER\_CR1.EN，启动计数器。

### 15.7.2 结合 DMA 输入捕获功能

1. 初始化 LPTIMER 时钟模块。
2. 配置 LPTIMER\_CR2.SINGLE，设置计数模式。
3. 配置 LPTIMER\_CR2.DIVCFG，设置分频值。
4. 配置 LPTIMER\_CR2.CLKSEL，设置时钟源。
5. 配置 LPTIMER\_CR2.TRIGEDGE，设置外部输入信号捕捉边沿。
6. DMA 模块时钟初始化。
7. 配置 DMA\_CHSPERC、DMA\_CHCTRLC、REG\_DMA\_CHDPERC，设置源传输和目的传输数据格式，选择传输通道。
8. 配置 DMA\_SRCADDRC、REG\_DMA\_CHCTRLC，设置源地址和目的地址，设置传输块大小。
9. 配置 DMA\_EN=1，使能 DMA 传输。

10. 使能 LPTIM\_CR1.EN 启动计数器。

### 15.7.3 PWM 输出

1. 初始化 LPTIMER 时钟模块。
2. 配置引脚为复用为 LPTIM\_OUT。
3. 配置 LPTIMER\_CR2.SINGLE，设置计数模式。
4. 配置 LPTIMER\_CR2.DIVCFG，设置分频值。
5. 配置 LPTIMER\_CR2.CLKSEL，设置时钟源。
6. 配置 LPTIMER\_CCR1 捕捉比较寄存器值。
7. 配置 LPTIMER\_ARR1 目标寄存器值。
8. 配置 LPTIMER\_CCMCFG1.CC1P，选择 PWM 输出波形极性。
9. 配置 LPTIMER\_CCMCFG1.CC1S，选择 PWM 输出模式。
10. 配置 LPTIMER\_CCMCFG1.CC1E，使能 LPTIM0 捕获比较功能。
11. 使能 LPTIMER\_IER 中断寄存器，打开中断。
12. 使能 LPTIMER\_CR1.EN，启动计数器。

### 15.7.4 ETR 脉冲触发计数模式

1. 初始化 LPTIMER 时钟模块。
2. 配置引脚为复用为 LPTIM\_ETR。
3. 配置 LPTIMER\_CR2.SINGLE，设置计数模式。
4. 配置 LPTIMER\_CR2.DIVCFG，设置分频值。
5. 配置 LPTIMER\_CR2.CLKSEL，设置时钟源用来采样 ETR 信号。
6. 配置 LPTIMER\_ARR1 目标寄存器值。
7. 配置 LPTIMER\_CR2.TRIGEDGE，设置 ETR 触发边沿。
8. 配置 LPTIMER\_CR2.LPMOD，选择 Trigger 脉冲触发计数模式。

9. 使能 LPTIMER\_IER.TIE 中断寄存器，打开外部触发中断。
10. 使能 LPTIMER\_CR1.EN，启动计数器。

### 15.7.5 ETR 脉冲计数模式

1. 初始化 LPTIMER 时钟模块。
2. 配置引脚为复用为 LPTIM\_IN。
3. 配置 LPTIMER\_CR2.SINGLE，设置计数模式。
4. 配置 LPTIMER\_CR2.DIVCFG，设置分频值。
5. 配置 LPTIMER\_CR2.CLKSEL，设置时钟源为 0x011，将 LPTIM\_IN 输入作为计数时钟。
6. 配置 LPTIMER\_ARR1 目标寄存器值。
7. 配置 LPTIMER\_CR2.EDGESEL，设置 ETR 时钟输入边沿。
8. 配置 LPTIMER\_CR2.LPMOD，选择 ETR 脉冲计数模式。
9. 使能 LPTIMER\_IER.TIE 中断寄存器，打开中断。
10. 使能 LPTIMER\_CR1.EN，启动计数器。

### 15.7.6 Timeout 模式

1. 初始化 LPTIMER 时钟模块。
2. 配置引脚为复用为 LPTIM\_EXT。
3. 配置 LPTIMER\_CR2.SINGLE，设置计数模式。
4. 配置 LPTIMER\_CR2.DIVCFG，设置分频值。
5. 配置 LPTIMER\_CR2.CLKSEL，设置时钟源。
6. 配置 LPTIMER\_ARR1 目标寄存器值。
7. 配置 LPTIMER\_CR2.TRIGEDGE，设置外部触发边沿。
8. 配置 LPTIMER\_CR2.LPMOD，选择 Timeout 模式。
9. 使能 LPTIMER\_IER 中断寄存器，打开溢出中断。

10. 使能 LPTIMER\_CR1.EN, 启动计数器。

注：计数器溢出前没有出现新的 trigger, 则产生溢出中断并停止计数, 并清除使能, 如果要重新使用, 需要再次使能该中断。

# 16 DMA

## 16.1 概述

直接存储器访问 (DMA)，支持 2 通道数据传输。

## 16.2 主要特性

- 单 MASTER 口
- 可以控制 FLASH、SRAM、SPI、I2C、ADC、LPTIMER、GTIMER、ATIMER 之间的数据传输，其中 FLASH 仅可以作为源地址
- 支持 Memory to Memory 模式、Memory to Peripheral 模式、Peripheral to Memory 模式、Peripheral to Peripheral 模式
- 内部含有 2 个 DMA 通道
- 数据传输的位宽可设、传输的 Block 长度可设
- 支持源地址不变传输、递增传输、递减传输，支持目的地址的不变传输、递增传输、递减传输，支持无限传输
- 支持 burst 功能，其中传输数目可配 2~16 可配置

## 16.3 寄存器描述

DMA 寄存器基地址：0x40020000

偏置	名称	描述
0x00	DMA_SRCADDR_C0	DMA 通道 0 源传送地址寄存器
0x04	DMA_DSTADDR_C0	DMA 通道 0 目的传送地址寄存器
0x08	DMA_CHCTRL_C0	DMA 通道 0 控制信息寄存器
0x0C	DMA_CHSTS_C0	DMA 通道 0 传送状态寄存器
0x10	DMA_CHSPER_C0	DMA 通道 0 源外设选择寄存器

偏置	名称	描述
0x14	DMA_CHDPER_C0	DMA 通道 0 目标外设选择寄存器
0x20	DMA_SRCADDR_C1	DMA 通道 1 源传送地址寄存器
0x24	DMA_DSTADDR_C1	DMA 通道 1 目的传送地址寄存器
0x28	DMA_CHCTRL_C1	DMA 通道 1 控制信息寄存器
0x2C	DMA_CHSTS_C1	DMA 通道 1 传送状态寄存器
0x30	DMA_CHSPER_C1	DMA 通道 1 源外设选择寄存器
0x34	DMA_CHDPER_C1	DMA 通道 1 目标外设选择寄存器
0x40	DMA_EN	DMA 控制器使能寄存器
0x44	DMA_SOFT_RESET	DMA 软复位寄存器
0x48	DMA_INT_STATUS	DMA 中断指示寄存器
0x4C	DMA_INT_MASK	DMA 中断屏蔽寄存器
0x54	DMA_PER_REQ	DMA 外设请求状态寄存器

### 16.3.1 DMA 通道源传送地址寄存器 (DMA\_SRCADDR\_Cx) (偏移: 20\*x+00h) (x=0,1)

比特	名称	属性	复位值	描述
31:0	SRC_ADDR	RW	0x0	源地址

### 16.3.2 DMA 通道目的传送地址寄存器 (DMA\_DSTADDR\_Cx) (偏移: 20\*x+04h) (x=0,1)

比特	名称	属性	复位值	描述
31:0	DST_ADDR	RW	0x0	目的地址

### 16.3.3 DMA 通道控制信息寄存器 (DMA\_CHCTRL\_Cx) (偏移: 20\*x+08h) (x=0,1)

比特	名称	属性	复位值	描述
31:30	WIDTH	RW	0x0	数据位宽: 0: 8 位数据位宽 1: 16 位数据位宽 2: 32 位数据位宽

比特	名称	属性	复位值	描述																				
				3: 非法, 但模块表现为 32 位的读写 源和目的数据位宽一样																				
29:15	XFER_SIZE	RW	0x0	传输块大小: 对 8 位数据位宽支持 32767 bytes 的块。 对 16 位数据位宽支持 32767 halfwords 的块。 对 32 位数据位宽支持 32767 words 的块。 注: 本次 block 传输完成后清零。																				
14:12	FLOW_CTRL	RW	0x0	流控模式: <table border="1"> <thead> <tr> <th>数据值</th> <th>源</th> <th>目的</th> <th>流控</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Memory</td> <td>Memory</td> <td>DMA</td> </tr> <tr> <td>1</td> <td>Memory</td> <td>Peripheral</td> <td>DMA</td> </tr> <tr> <td>2</td> <td>Peripheral</td> <td>Memory</td> <td>DMA</td> </tr> <tr> <td>3</td> <td>Peripheral</td> <td>Peripheral</td> <td>DMA</td> </tr> </tbody> </table>	数据值	源	目的	流控	0	Memory	Memory	DMA	1	Memory	Peripheral	DMA	2	Peripheral	Memory	DMA	3	Peripheral	Peripheral	DMA
数据值	源	目的	流控																					
0	Memory	Memory	DMA																					
1	Memory	Peripheral	DMA																					
2	Peripheral	Memory	DMA																					
3	Peripheral	Peripheral	DMA																					
11	TRANS_FREE	RW	0x0	无限传输使能位: 1: 无限传输开启 0: 无限传输结束 无限传输功能是传输数量不受 XFER_SIZE 寄存器控制, 可以一直传输下去。																				
10	WAP_SRC_EN	RW	0x0	Src Burst 功能使能: 1: 开启 Src Burst 功能 0: 禁止 Src Burst 功能 Src Burst 功能是开启 Burst 传输, 可以循环读取 2、3、4、5、6、7、8、9 个连续的地址空间的数据。读取方式可以递增、递减。																				
9:8	RSV	-	-	保留																				
7	WAP_DST_EN	RW	0x0	Dst Burst 功能使能: 1: 开启 Dst Burst 功能 0: 禁止 Dst Burst 功能 Dst Burst 功能是开启 Burst 传输, 可以循环读取 2、3、4、5、6、7、8、9 个连续的地址空间的数据。读取方式可以递增、递减。																				
6:5	RSV	-	-	保留																				
4:3	DST_INC	RW	0x0	目的地址递增指示位, 如果有效, 则目的地址将随读取的数据递增, 否则保持不变 01: 地址递增 10: 地址递减																				
2:1	SRC_INC	RW	0x0	源地址递增指示位, 如果有效, 则源地址将随读取的																				

比特	名称	属性	复位值	描述
				数据递增，否则保持不变 01: 地址递增 10: 地址递减
0	CH_EN	RW	0x0	通道使能标志，对于 DMA 流控时，块传送结束后自动清 0

### 16.3.4 DMA 通道传送状态寄存器 (DMA\_CHSTS\_Cx) (偏移: 20\*x + 0Ch) (x=0,1)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:1	LENGTH	R	0x0	在 DMA 传输时，表示此通道已经传输的数据长度
0	CH_BUSY	R	0x0	通道工作状态信息: 0: Idle 1: Busy

### 16.3.5 DMA 通道源外设选择寄存器 (DMA\_CHSPER\_Cx) (偏移: 20\*x + 10h) (x=0,1)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:5	WAP_SRC_NUM	RW	0x0	SRC burst 传输数目配置: 0: SRC burst 2 1: SRC burst 3 2: SRC burst 4 3: SRC burst 5 4: SRC burst 6 5: SRC burst 7 6: SRC burst 8 7: SRC burst 9
4:0	SPER	RW	0x0	源外设，主要用于源外设的请求选取 具体外设分配为: 1: SPI 接收 2: I2C 接收

比特	名称	属性	复位值	描述
				4: ADC 接收
				5: GTIMER0_UEV
				6: GTIMER0_TG
				7: GTIMER0_CC4
				8: GTIMER0_CC3
				9: GTIMER0_CC2
				10: GTIMER0_CC1
				11: GTIMER1_UEV
				12: GTIMER1_TG
				13: GTIMER1_CC4
				14: GTIMER1_CC3
				15: GTIMER1_CC2
				16: GTIMER1_CC1
				17: GTIMER2_UEV
				18: GTIMER2_TG
				19: GTIMER2_CC4
				20: GTIMER2_CC3
				21: GTIMER2_CC2
				22: GTIMER2_CC1
				23: LPTIMER_RX2
				24: LPTIMER_RX1
				25: ATIMER_CTU
				26: ATIMER_CC4
				27: ATIMER_CC3
				28: ATIMER_CC2
				29: ATIMER_CC1
				30: ATIMER_TG
				31: ATIMER_UEV

### 16.3.6 DMA 通道目标外设选择寄存器 (DMA\_CHDPER\_Cx) (偏移: 20\*x +14h) (x=0,1)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:5	WAP_DST_NUM	RW	0x0	DST burst 传输数目配置: 0: DST burst 2

比特	名称	属性	复位值	描述
				1: DST burst 3 2: DST burst 4 3: DST burst 5 4: DST burst 6 5: DST burst 7 6: DST burst 8 7: DST burst 9
4:0	DPER	RW	0x0	目的外设，主要用于目的外设的请求选取 具体外设分配为： 0: SPI 发送 3: I2C 发送 5: GTIMER0_UEV 6: GTIMER0_TG 7: GTIMER0_CC4 8: GTIMER0_CC3 9: GTIMER0_CC2 10: GTIMER0_CC1 11: GTIMER1_UEV 12: GTIMER1_TG 13: GTIMER1_CC4 14: GTIMER1_CC3 15: GTIMER1_CC2 16: GTIMER1_CC1 17: GTIMER2_UEV 18: GTIMER2_TG 19: GTIMER2_CC4 20: GTIMER2_CC3 21: GTIMER2_CC2 22: GTIMER2_CC1 25: ATIMER_CTU 26: ATIMER_CC4 27: ATIMER_CC3 28: ATIMER_CC2 29: ATIMER_CC1 30: ATIMER_TG 31: ATIMER_UEV

### 16.3.7 DMA 控制器使能寄存器 (DMA\_EN) (偏移: 40h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1	GRAND_SET	RW	0x0	1: 使能通道优先级的轮询功能, 下一次传输通道优先级将进行轮询 (初始通道 1 的优先级高于通道 0, 若当前传输下通道 1 优先级高于通道 0, 下次传输时通道 0 优先级高于通道 1) 0: 通道之间为固定优先级 (默认为通道 0 优先级高于通道 1)
0	DMA_EN	RW	0x0	1: 使能 DMA 控制器 0: 关闭 DMA 控制寄存器

### 16.3.8 DMA 软复位寄存器 (DMA\_SOFT\_RESET) (偏移: 44h)

比特	名称	属性	复位值	描述
31:0	DMA_SOFT_RESET	W	0x0	本寄存器为写操作虚拟寄存器, 当 DMA 模块采样到对此寄存器写任意值, DMA 将复位状态机以及需要复位的寄存器。

### 16.3.9 DMA 中断指示寄存器 (DMA\_INT\_STATUS) (偏移: 48h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	INT_TC_C1	W1C	0x0	通道 1 块传输结束中断指示, 写 1 清 0
2	INT_TC_C0	W1C	0x0	通道 0 块传输结束中断指示, 写 1 清 0
1	INT_ERR_C1	W1C	0x0	通道 1 总线出错中断指示, 写 1 清 0
0	INT_ERR_C01	W1C	0x0	通道 0 总线出错中断指示, 写 1 清 0

### 16.3.10 DMA 中断屏蔽寄存器 (DMA\_INT\_MASK) (偏移: 4Ch)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	MASK_TC_C1	RW	0x0	通道 1 块传输结束中断屏蔽, 如果为低, 将不输出 IntTc 中断, 即 IntTc=0
2	MASK_TC_C0	RW	0x0	通道 0 块传输结束中断屏蔽, 如果为低, 将不输

比特	名称	属性	复位值	描述
				出 IntTc 中断, 即 IntTc=0
1	MASK_ERR_C1	RW	0x0	通道 1 总线出错中断屏蔽; 如果为低, 将不输出 IntErr 中断, 即 IntErr=0
0	MASK_ERR_C0	RW	0x0	通道 0 总线出错中断屏蔽; 如果为低, 将不输出 IntErr 中断, 即 IntErr=0

### 16.3.11 DMA 外设请求状态寄存器 (DMA\_PER\_REQ) (偏移: 54h)

比特	名称	属性	复位值	描述
31	ATIMER_UEV_REQ	R	0x0	ATIMER update 事件请求
30	ATIMER_TG_REQ	R	0x0	ATIMER 外部触发请求
29	ATIMER_CC1_REQ	R	0x0	ATIMER 通道 1 请求
28	ATIMER_CC2_REQ	R	0x0	ATIMER 通道 2 请求
27	ATIMER_CC3_REQ	R	0x0	ATIMER 通道 3 请求
26	ATIMER_CC4_REQ	R	0x0	ATIMER 通道 4 请求
25	ATIMER_CTU_REQ	R	0x0	ATIMER COM 事件请求
24	LPTIMER_RX0_REQ	R	0x0	LPTIMER RX0 接收请求
23	LPTIMER_RX1_REQ	R	0x0	LPTIMER RX1 接收请求
22	GTIMER2_CC1_REQ	R	0x0	GTIMER2 通道 1 请求
21	GTIMER2_CC2_REQ	R	0x0	GTIMER2 通道 2 请求
20	GTIMER2_CC3_REQ	R	0x0	GTIMER2 通道 3 请求
19	GTIMER2_CC4_REQ	R	0x0	GTIMER2 通道 4 请求
18	GTIMER2_TG_REQ	R	0x0	GTIMER2 外部触发请求
17	GTIMER2_UEV_REQ	R	0x0	GTIMER2 update 事件请求
16	GTIMER1_CC1_REQ	R	0x0	GTIMER1 通道 1 请求
15	GTIMER1_CC2_REQ	R	0x0	GTIMER1 通道 2 请求
14	GTIMER1_CC3_REQ	R	0x0	GTIMER1 通道 3 请求
13	GTIMER1_CC4_REQ	R	0x0	GTIMER1 通道 4 请求
12	GTIMER1_TG_REQ	R	0x0	GTIMER1 外部触发请求
11	GTIMER1_UEV_REQ	R	0x0	GTIMER1 update 事件请求
10	GTIMER0_CC1_REQ	R	0x0	GTIMER0 通道 1 请求
9	GTIMER0_CC2_REQ	R	0x0	GTIMER0 通道 2 请求
8	GTIMER0_CC3_REQ	R	0x0	GTIMER0 通道 3 请求
7	GTIMER0_CC4_REQ	R	0x0	GTIMER0 通道 4 请求
6	GTIMER0_TG_REQ	R	0x0	GTIMER0 外部触发请求
5	GTIMER0_UEV_REQ	R	0x0	GTIMER0 update 事件请求

比特	名称	属性	复位值	描述
4	ADC_REQ	R	0x0	ADC 接收请求
3	I2C_TX_REQ	R	0x0	I2C 发送请求
2	I2C_RX_REQ	R	0x0	I2C 接收请求
1	SPI_RX_REQ	R	0x0	SPI 接收请求
0	SPI_TX_REQ	R	0x0	SPI 发送请求

## 16.4 使用流程

软件配置步骤:

1. 配置 DMA\_CHCTRL\_Cx[14:12], 选择 DMA 传输模式。
2. 配置 DMA\_CHCTRL\_Cx[4:0]和 DMA\_CHDPER\_Cx[4:0], 选择外设握手信号 (传输地址为外设时才需要设置)。
3. 配置 DMA\_CHCTRL\_Cx[4:3]和 DMA\_CHCTRL\_Cx [2:1], 选择源地址和目的地址是否递增或不变。
4. 配置 DMA\_CHCTRL\_Cx[31:30], 选择传输数据的位宽。
5. 配置 DMAC\_EN[0]为 1, 使能 DMA 控制器。
6. 配置 DMA\_SRCADDR\_Cx, 配置通道源地址。
7. 配置 DMA\_DSTADDR\_Cx, 配置通道目的地址。
8. 配置 DMA\_CHCTRL\_Cx[29:15], 配置传输块数量。
9. 配置 DMA\_CHCTRL\_Cx[0]为 1, 使能 DMA 通道传输。
10. 等待 DMA\_CHCTRL\_Cx[0]为 0, 传输完成。若使能了传输结束中断, 则等待传输结束中断后再处理。

# 17 CRC

## 17.1 概述

CRC 是一个以多项式  $G(x) = x^{16} + x^{12} + x^5 + 1$  为计算式的硬件 16 位 CRC 循环冗余校验计算电路。可以根据用户预设的 CRC 初值和通讯数据计算出合适的 CRC 结果，并且支持设置输入数据与结果的正反向。

## 17.2 寄存器描述

CRC 寄存器基地址：0x4000\_1800

表 17-1: CRC 寄存器列表

偏置	名称	描述
0x00	CRC_DATA	CRC 数据寄存器
0x04	CRC_INIT	CRC 初始值寄存器
0x08	CRC_CTRL	CRC 控制寄存器

### 17.2.1 数据寄存器 CRC\_DATA (偏移: 00H)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	RSLT2	R	0x0	读出 16 位 CRC 计算结果的高 8 位
7:0	RSLT1	RW	0x0	写：写入需要进行 CRC 校验计算的数据，如需要校验的数据不止 8 位需按顺序逐次写入 读：读出 16 位 CRC 计算结果的低 8 位

低 8 位对于需校验数据来说为只写，写入后无法再次读出。

读操作返回 16 位 CRC 计算结果，其中低 8 位为与数据寄存器复用。

读操作将会对 CRC 计算清零，即读操作后将会重新载入初始值，次输入数据时会进行与读之前结果无关的新一轮计算。

### 17.2.2 初始值寄存器 CRC\_INIT (偏移: 04H)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	INIT	RW	0x0	写入 16 位 CRC 初始值

### 17.2.3 控制寄存器 CRC\_CTRL (偏移: 08H)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	RSLT_REV	RW	0x0	CRC 计算结果是否进行高低位倒序: 1: 倒序 0: 不倒序
1	DATA_REV	RW	0x0	CRC 计算数据是否进行高低位倒序: 1: 倒序 0: 不倒序
0	INITIAL_REV	RW	0x0	CRC 初始值是否进行高低位倒序: 1: 倒序 0: 不倒序

## 17.3 使用流程

1. 设置 16 位初始值 CRC\_INIT[15:0]。
2. 设置 CRC\_CTRL[2:0]，选择数据是否倒序。
3. 向 CRC\_DATA 中写入 8 位 CRC 计算数据，如没有完成 CRC 数据输入顺次输入之后 8 位数据直至输入完成。
4. 读 CRC\_DATA，将一次返回 CRC 计算结果。

注意：读取结果后 CRC 计算模块将结束当前计算并重新载入初始值以备下次计算使用。

# 18 RNG

## 18.1 概述

RNG 是一款随机数生成器。可通过写入不同的随机数种子来生成不同的随机数序列。

## 18.2 主要特性

- 32 位随机数
- 可连续读取随机数序列

## 18.3 寄存器描述

寄存器基地址：0x4000\_2000

表 18-1: RNG 寄存器列表

偏置	名称	描述
0x0E0	RNG_CR	随机数控制寄存器
0x0E4	RNG_SEED	随机数种子寄存器
0x0E8	RNG_DATA	随机数数据寄存器

### 18.3.1 随机数控制寄存器 RNG\_CR (偏移: 0E0h)

比特	名称	属性	复位值	描述
31:1	RSV	R	0	保留
0	EN	RW	0x0	1: 随机数使能 0: 随机数禁止

### 18.3.2 随机数种子寄存器 RNG\_SEED (偏移: 0E4h)

比特	名称	属性	复位值	描述
31:0	SEED	RW	0x0	随机数种子寄存器

### 18.3.3 随机数数据寄存器 RNG\_DATA (偏移: 0E8h)

比特	名称	属性	复位值	描述
31:0	DATA	R	0xFF00FF	随机数寄存器。读取此寄存器，读出随机数值。

## 18.4 使用流程

1. 配置 RNG\_CR[0]为 1，使能随机数。
2. 向 RNG\_SEED[31:0]写入随机数种子值。
3. 读取 RNG\_DATA[31:0]，读出随机数值，可连续读取。

# 19 WDT

## 19.1 概述

看门狗定时器在到达超时的值的时候可以产生不可屏蔽中断或者是复位。当系统由于软件错误或是由于外部设备故障而无法按照预期的方式响应的时候,使用看门狗定时器可以重新获得控制权。

## 19.2 主要特性

- 32 位递减并且可编程装载的寄存器
- 独立的看门狗时钟使能
- 带中断屏蔽的中断生成逻辑
- 软件跑飞保护锁定寄存器
- 复位使能/禁止产生逻辑
- 调试期间,微处理器的 CPU 暂停时,用户可使能的停滞

## 19.3 寄存器描述

WDT 寄存器基地址: 0x40002400

表 19-1: WDT 寄存器列表

偏置	名称	描述
0x00	WDT_LOAD	装载寄存器
0x04	WDT_CNT	计数寄存器
0x08	WDT_CTRL	控制寄存器
0x0C	WDT_CLR	清除寄存器
0x10	WDT_INTRAW	RAW 中断状态寄存器
0x14	WDT_MINTS	MASK 中断状态寄存器
0x18	WDT_STALL	STALL 寄存器
0x1C	WDT_LOCK	LOCK 寄存器

### 19.3.1 装载寄存器 WDT\_LOAD (偏移: 00h)

比特	名称	属性	默认值	功能描述
31:0	LOAD	RW	0xFFFFFFFF	WDOG初始装载值

### 19.3.2 计数寄存器 WDT\_CNT (偏移: 04h)

比特	名称	属性	默认值	功能描述
31:0	CNT	R	0xFFFFFFFF	WDOG内部CNT计数值

### 19.3.3 控制寄存器 WDT\_CTRL (偏移: 08h)

比特	名称	属性	默认值	功能描述
31	WRC	R	0x1	WDT加载值设置或写WDT_CTRL寄存器生效。向WDT_LOAD或者WDT_CTRL寄存器进行写操作时，设置位生效会有一定时间的延时。 0: 设置为仍未生效 1: 设置位生效
30:2	RSV	-	-	保留
1	RSTEN	RW	0x0	WDT溢出复位使能: 0: 不使能溢出复位功能 1: 使能溢出复位功能
0	INTEN	RW	0x0	WDT中断使能: 0: 不使能中断 1: 使能中断

### 19.3.4 清除寄存器 WDT\_CLR (偏移: 0ch)

比特	名称	属性	默认值	功能描述
31:0	CLR_CARRY	W	0x0	向此寄存器写入任何值，将清除WDT溢出状态，从而清除掉中断和复位。

### 19.3.5 RAW 中断状态寄存器 WDT\_INTRAW (偏移: 10h)

比特	名称	属性	默认值	功能描述
31:0	INTRAW	R	0x0	原始中断寄存器，未经中断使能屏蔽： 0: WDT内部未发生溢出 1: WDT内部发生溢出

### 19.3.6 MASK 中断状态寄存器 WDT\_MINTS (偏移: 14h)

比特	名称	属性	默认值	功能描述
31:0	INTMS	R	0x0	0: WDT未产生中断 1: WDT产生中断

### 19.3.7 STALL 控制寄存器 WDT\_STALL (偏移: 18h)

比特	名称	属性	默认值	功能描述
31:9	RSV	-	-	保留
8	STALL	RW	0x0	WDT在芯片处于HALT状态时不计数功能的使能位： 0: 不使能HALT状态计数器停止工作功能 1: 使能HALT状态计数器停止工作功能
7:0	RSV	-	-	保留

### 19.3.8 LOCK 寄存器 WDT\_LOCK (偏移: 1ch)

比特	名称	属性	默认值	功能描述
31:0	LOCK	W	0x0	WDT LOCK功能使能，当使能LOCK功能时，除此寄存器外的所有WDT寄存器均不可写。向此寄存器写任意值，使能WDT LOCK功能，向此寄存器写0x1ACCE551清除LOCK功能。

## 19.4 使用流程

### 19.4.1 WDT 定时器配置

1. 向 WDT\_LOCK 寄存器写入 0x1ACCE551 解锁寄存器。
2. 在 WDT\_STALL 寄存器设置分频值。
3. 在 WDT\_CTRL 寄存器使能 INTEN 中断功能。
4. 等待 WDT\_CTRL 寄存器的 WRC 位被置位，即设置生效。
5. 在 WDT\_LOAD 寄存器里装载所需要的加载值。
6. 等待 WDT\_CTRL 寄存器的 WRC 位被置位，即设置生效。
7. 向 WDT\_LOCK 寄存器写入任意值锁定寄存器。
8. 系统根据装载值及计数时钟分频值定时进入中断。

### 19.4.2 WDT 喂狗流程配置

1. 向 WDT\_LOCK 寄存器写入 0x1ACCE551 解锁寄存器。
2. 在 WDT\_STALL 寄存器设置分频值。
3. 在 WDT\_CTRL 寄存器使能 RSTEN 复位功能，选择使能 INTEN 中断功能；等待 WDT\_CTRL 寄存器的 WRC 位被置位，即设置生效。
4. 在 WDT\_LOAD 寄存器里装载所需要的加载值。
5. 等待 WDT\_CTRL 寄存器的 WRC 位被置位，即设置生效。
6. 向 WDT\_LOCK 寄存器写入任意值锁定寄存器。
7. 系统可通过软件定时更新 WDT\_LOAD 或者通过清中断标志位定时喂狗。若未能在定时时间内喂狗，则系统会在 2 倍定时时间后复位。

## 20 WWDT

### 20.1 概述

窗口看门狗是一个与 CPU 同步运行的看门狗，目的是实时监控 CPU 运行状态，在 CPU 运行异常的情况下复位 CPU，避免不可预计的后果。

### 20.2 主要特性

- 18 位递增并且可编程装载的寄存器
- 系统内部的故障探测器
- 时钟与系统时钟相同
- 用于监视软件错误
- 窗口前喂狗或超时不喂狗都会触发复位（喂狗有效窗口为 50%-100%时间内）
- 计数器达到溢出时间的 75%时触发预警中断

### 20.3 寄存器描述

WWDT 寄存器基地址：0x40003C00

表 20-1: WWDT 寄存器列表

偏置	名称	描述
0x00	WWDT_CON	控制寄存器
0x04	WWDT_CFG	配置寄存器
0x08	WWDT_CNT	计数寄存器
0x0C	WWDT_IE	中断使能寄存器
0x10	WWDT_IF	中断标志寄存器

### 20.3.1 控制寄存器 WWDT\_CON (偏移: 00h)

比特	名称	属性	默认值	功能描述
31:8	RSV	-	-	保留
7:0	CON	W	0x0	当CPU向此地址写入0x5A时启动WWDT定时器 在启动WWDT后, 当CPU向此地址写入0xAC时清零计数器

### 20.3.2 配置寄存器 WWDT\_CFG (偏移: 04h)

比特	名称	属性	默认值	功能描述
31:5	RSV	-	-	保留
4	STOP_EN	RW	0x1	WWDT断点使能位, 在软件设置断点后, 禁止计数器计数: 0: 计数器正常工作 1: 计数器停止计数
3:0	CFG	RW	0x0	配置看门狗溢出时间: 0000: $t_{PCLK} * 4096 * 1$ 0001: $t_{PCLK} * 4096 * 4$ 0010: $t_{PCLK} * 4096 * 16$ 0011: $t_{PCLK} * 4096 * 64$ 0100: $t_{PCLK} * 4096 * 128$ 0101: $t_{PCLK} * 4096 * 256$ 0110: $t_{PCLK} * 4096 * 512$ 0111: $t_{PCLK} * 4096 * 1024$ 1000: $t_{PCLK} * 4096 * 2048$ 1001: $t_{PCLK} * 4096 * 4096$ 1010: $t_{PCLK} * 4096 * 8192$ 1011: $t_{PCLK} * 4096 * 16384$ 1100: $t_{PCLK} * 4096 * 32768$ 1101: $t_{PCLK} * 4096 * 65536$ 1110: $t_{PCLK} * 4096 * 131072$ 1111: $t_{PCLK} * 4096 * 262144$

### 20.3.3 计数寄存器 WWDT\_CNT (偏移: 08h)

比特	名称	属性	默认值	功能描述
31:18	RSV	-	-	保留

比特	名称	属性	默认值	功能描述
17:0	CNT	R	0x0	WWDT计数寄存器值，软件可通过查询此寄存器了解WWDT计时进度

### 20.3.4 中断使能寄存器 WWDT\_IE (偏移: 0ch)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	IE	RW	0x0	WWDT中断使能: 0: 中断使能禁止 1: 中断使能打开

### 20.3.5 中断标志寄存器 WWDT\_IF (偏移: 10h)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	IF	W1C	0x0	WWDT 75%计时中断标志，写1清零: 0: 无中断产生 1: 中断标志置位

## 20.4 使用流程

- WWDT 定时器配置:

1. 在 WWDT\_CFG 寄存器设置溢出时间长度。
2. 在 WWDT\_IE 寄存器里打开中断使能。
3. 在 WWDT\_CON 寄存器中写入 0x5A 启动 WWDT 定时器。
4. 等待发生中断 (计数到 75%时间产生中断)。
5. 等待发生复位 (溢出后产生复位)。

- WWDT 喂狗流程配置:

在计数时间 50%~100%之内，在 WWDT\_CON 寄存器中写入 0xAC 清零计数器。

# 21 ADC

## 21.1 概述

这是一个 12 位的 ADC 逐次接近型模数转换器。它具有多达 16 个输入通道，可测量来自 11 个外部源的信号、3 个内部 PGA 输出、1 个内部 LDO 输出和 1 个内部  $1/4 V_{DDH}$  输出。这些通道的 A/D 转换可在单次或连续扫描模式下进行。ADC 控制器实现 CPU 和 SAR ADC 之间的通信。ADC 转换的结果存储在数据寄存器的低 12 位。

## 21.2 主要特性

- 支持 DMA 传输模式
- 5 位的可编程分频器，用于产生 A/D 时钟
- 支持 12 位分辨率 A/D 输入数据，最大采样率为 1MSps，采样率可通过软件配置
- 支持 16 通道 ADC 输入：11 个引脚通道、3 个内部 PGA 输出、1 个内部 LDO 输出和 1 个内部  $1/4 V_{DDH}$  输出
- 支持关闭模拟 ADC
- 支持轮询 (poll) 和中断 (interrupt) 传输模式
- 支持单次扫描或连续扫描模式
- 中断源：单次转换结束中断、转换序列结束中断、注入通道单次转换结束中断、注入通道转换序列结束中断、模拟看门狗超出上限中断、模拟看门狗低于下限中断、数据冲突中断；通道数据有效标志 (8 个普通通道、4 个注入通道、过采样各有一个标志)
- 支持片内外设触发 ADC 转换
- ADC 电压输入范围：0~Vref
- ADC 参考电压源可选择：芯片供电电压 VDDH、IO 管脚外接电压 VREFIO、内置 VREF

## 21.3 ADC 管脚分布

表 21-1: ADC 管脚分布

通道	引脚名称
ADC_CH0	AIN[0], PC2
ADC_CH1	AIN[1], PD0
ADC_CH2	AIN[2], PD1
ADC_CH3	AIN[3], PD2
ADC_CH4	AIN[4], PD3
ADC_CH5	AIN[5], PB7
ADC_CH6	AIN[6], PB4
ADC_CH7	AIN[7], PB3
ADC_CH8	AIN[8], PB2
ADC_CH9	AIN[9], PB1
ADC_CH10	AIN[10], PB0
ADC_CH11	AIN[11], PGA0_VOUT
ADC_CH12	AIN[12], PGA1_VOUT
ADC_CH13	AIN[13], PGA2_VOUT
ADC_CH14	内部通道 VDDH 1/4
ADC_CH15	内部通道 VREF

## 21.4 寄存器描述

ADC 寄存器基地址: 0x4000\_1C00

表 21-2: ADC 寄存器列表

偏置	名称	描述
0x00	ADC_ISR	ADC 中断和状态寄存器
0x04	ADC_IER	ADC 中断使能寄存器
0x08	ADC_CR	ADC 控制寄存器
0x0C	ADC_CFGR	ADC 配置寄存器
0x10	ADC_SMTR	ADC 采样时间控制寄存器
0x14	ADC_CHER	ADC 通道控制寄存器
0x18	ADC_SEQ	ADC 通道选择寄存器
0x20	ADC_HLTR	ADC 模拟看门狗阈值寄存器
0x24	ADC_IJCHER	ADC 注入通道控制寄存器

偏置	名称	描述
0x28	ADC_IJSEQ	ADC 注入通道选择寄存器
0x30	ADC_DR0	ADC 数据寄存器 0
0x34	ADC_DR1	ADC 数据寄存器 1
0x38	ADC_DR2	ADC 数据寄存器 2
0x3C	ADC_DR3	ADC 数据寄存器 3
0x40	ADC_DR4	ADC 数据寄存器 4
0x44	ADC_DR5	ADC 数据寄存器 5
0x48	ADC_DR6	ADC 数据寄存器 6
0x4C	ADC_DR7	ADC 数据寄存器 7
0x50	ADC_IJDR0	ADC注入通道数据寄存器0
0x54	ADC_IJDR1	ADC 注入通道数据寄存器 1
0x58	ADC_IJDR2	ADC 注入通道数据寄存器 2
0x5C	ADC_IJDR3	ADC 注入通道数据寄存器 3
0x60	ADC_DR_OVS	ADC 过采样平均结果寄存器
0xA8	ADC_HDT	ADC 硬件触发使能配置寄存器
0xAC	ADC_IJHDT	ADC 注入通道硬件触发使能配置寄存器
0xB0	ADC_SMTR0	ADC 通道采样时间配置寄存器 0
0xB4	ADC_SMTR1	ADC 通道采样时间配置寄存器 1
0xB8	ADC_IJSMTR	ADC 注入通道采样时间配置寄存器

#### 21.4.1 ADC 中断和状态寄存器 (ADC\_ISR) (偏移: 000h)

比特	名称	属性	复位值	描述
31:29	RSV	-	-	保留
28	DR_OVS_V	W1C	0x0	ADC_DR_OVS 寄存器数据有效标志, 读 ADC_DR_OVS 寄存器或写 1 清零 0: ADC_DR_OVS 寄存器中无新数据 1: ADC_DR_OVS 寄存器中有新数据
27	IJDR3_V	W1C	0x0	ADC_IJDR3 寄存器数据有效标志, 读 ADC_IJDR3 寄存器或写 1 清零 0: ADC_IJDR3 寄存器中无新数据 1: ADC_IJDR3 寄存器中有新数据
26	IJDR2_V	W1C	0x0	ADC_IJDR2 寄存器数据有效标志, 读 ADC_IJDR2 寄存器或写 1 清零 0: ADC_IJDR2 寄存器中无新数据 1: ADC_IJDR2 寄存器中有新数据

比特	名称	属性	复位值	描述
25	IJDR1_V	W1C	0x0	ADC_IJDR1 寄存器数据有效标志, 读 ADC_IJDR1 寄存器或写 1 清零 0: ADC_IJDR1 寄存器中无新数据 1: ADC_IJDR1 寄存器中有新数据
24	IJDR0_V	W1C	0x0	ADC_IJDR0 寄存器数据有效标志, 读 ADC_IJDR0 寄存器或写 1 清零 0: ADC_IJDR0 寄存器中无新数据 1: ADC_IJDR0 寄存器中有新数据
23	DR7_V	W1C	0x0	ADC_DR7 寄存器数据有效标志, 读 ADC_DR7 寄存器或写 1 清零 0: ADC_DR7 寄存器中无新数据 1: ADC_DR7 寄存器中有新数据
22	DR6_V	W1C	0x0	ADC_DR6 寄存器数据有效标志, 读 ADC_DR6 寄存器或写 1 清零 0: ADC_DR6 寄存器中无新数据 1: ADC_DR6 寄存器中有新数据
21	DR5_V	W1C	0x0	ADC_DR5 寄存器数据有效标志, 读 ADC_DR5 寄存器或写 1 清零 0: ADC_DR5 寄存器中无新数据 1: ADC_DR5 寄存器中有新数据
20	DR4_V	W1C	0x0	ADC_DR4 寄存器数据有效标志, 读 ADC_DR4 寄存器或写 1 清零 0: ADC_DR4 寄存器中无新数据 1: ADC_DR4 寄存器中有新数据
19	DR3_V	W1C	0x0	ADC_DR3 寄存器数据有效标志, 读 ADC_DR3 寄存器或写 1 清零 0: ADC_DR3 寄存器中无新数据 1: ADC_DR3 寄存器中有新数据
18	DR2_V	W1C	0x0	ADC_DR2 寄存器数据有效标志, 读 ADC_DR2 寄存器或写 1 清零 0: ADC_DR2 寄存器中无新数据 1: ADC_DR2 寄存器中有新数据
17	DR1_V	W1C	0x0	ADC_DR1 寄存器数据有效标志, 读 ADC_DR1 寄存器或写 1 清零 0: ADC_DR1 寄存器中无新数据 1: ADC_DR1 寄存器中有新数据
16	DR0_V	W1C	0x0	ADC_DR0 寄存器数据有效标志, 读 ADC_DR0 寄存器或

比特	名称	属性	复位值	描述
				写 1 清零 0: ADC_DR0 寄存器中无新数据 1: ADC_DR0 寄存器中有新数据
15:9	RSV	-	-	保留
8	IJ_EOS	W1C	0x0	注入通道转换序列结束标志，所有使能通道都转换完成后，置位 IJ_EOS，软件写 1 清零。
7	IJ_EOC	W1C	0x0	注入通道单次转换结束标志，每个通道转换完成后，置位 IJ_EOC，软件写 1 清零或读 ADC_IJDR0~ADC_IJDR3 数据寄存器清零。
6	AWD_AH	W1C	0x0	模拟看门狗超出上限中断标志，当采样值高于 AWD_HT 时，硬件置位，软件写 1 清零。
5	AWD_UL	W1C	0x0	模拟看门狗低于下限中断标志，当采样值低于 AWD_LT 时，硬件置位，软件写 1 清零。
4:3	RSV	-	-	保留
2	OVR	W1C	0x0	数据冲突标志，硬件置位，软件写 1 清零。当 DR 寄存器中的上一次转换结果还未被读取，新的转换结果又到来时，硬件置位 OVR 标志。 0: 没有数据冲突 1: 出现数据冲突
1	EOS	W1C	0x0	转换序列结束 (End of Sequence)，所有使能通道都转换完成后，置位 EOS，软件写 1 清零。
0	EOC	W1C	0x0	单次转换结束 (End of Conversion)，每个通道转换完成后，置位 EOC，软件写 1 清零或读 ADC_DR0~ADC_DR7、ADC_DR_OVS 数据寄存器清零。

### 21.4.2 ADC 中断使能寄存器 (ADC\_IER) (偏移: 004h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8	IJ_EOSIE	RW	0x0	注入通道转换序列结束中断使能寄存器，1使能，0关闭。
7	IJ_EOCIE	RW	0x0	注入通道单次转换结束中断使能寄存器，1使能，0关闭。
6	AWD_AHIE	RW	0x0	模拟看门狗采样值高于上限中断使能，1使能，0关闭。
5	AWD_ULIE	RW	0x0	模拟看门狗采样值低于下限中断使能，1使能，0关闭。
4:3	RSV	-	-	保留
2	OVRIE	RW	0x0	数据冲突中断使能寄存器，1使能，0关闭。
1	EOSIE	RW	0x0	转换序列结束 (End of Sequence) 中断使能寄存器，1

比特	名称	属性	复位值	描述
				使能, 0关闭。
0	EOCIE	RW	0x0	单次转换结束 (End of Conversion) 中断使能寄存器, 1使能, 0关闭。

### 21.4.3 ADC 控制寄存器 (ADC\_CR) (偏移: 008h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	AWD_AH_EN	RW	0x0	模拟看门狗采样值高于上限触发ATimer刹车使能: 1: 使能 0: 关闭
5	AWD_UL_EN	RW	0x0	模拟看门狗采样值低于下限触发ATimer刹车使能: 1: 使能 0: 关闭
4	BUFEN	RW	0x0	使能模拟输入buffer: 0: 不使能buffer 1: 使能buffer
3	STOP	W	0x0	ADC停止转换寄存器, 软件写1停止连续转换模式。
2	IJSTART	RW	0x0	ADC注入通道启动转换寄存器, 软件写1启动, 硬件自动清零。
1	START	RW	0x0	ADC启动转换寄存器, 软件写1启动, 硬件自动清零。
0	ADEN	RW	0x0	ADC使能寄存器。在启动转换前先要置位ADEN, 等待至少5个MCLK后再启动转换。 0: 关闭ADC 1: 打开ADC

### 21.4.4 ADC 配置寄存器 (ADC\_CFGR) (偏移: 00Ch)

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:26	AWDCH	RW	0x0	模拟窗口看门狗监视通道选择, 仅在AWDSC=1时有效: 0000: AWD监视SEQ0 0001: AWD监视SEQ1 0010: AWD监视SEQ2 0011: AWD监视SEQ3 0100: AWD监视SEQ4 0101: AWD监视SEQ5

比特	名称	属性	复位值	描述
				0110: AWD监视SEQ6 0111: AWD监视SEQ7 1000: AWD监视ISEQ0 1001: AWD监视ISEQ1 1010: AWD监视ISEQ2 1011: AWD监视ISEQ3 Others: 保留
25	AWDSC	RW	0x0	模拟窗口看门狗序列选择: 0: AWD监视所有被使能的外部输入通道 1: AWD监视AWDCH指定的单个通道
24	AWDEN	RW	0x0	模拟窗口看门狗使能寄存器, 仅能在START=0的情况下使能AWD: 0: 关闭AWD 1: 使能AWD
23:20	OVSS	RW	0x0	过采样移位控制寄存器: 0000: 不移位 0001: 右移1 bit 0010: 右移2 bits 0011: 右移3 bits 0100: 右移4 bits 0101: 右移5 bits 0110: 右移6 bits 0111: 右移7 bits 1000: 右移8 bits Others: 预留
19:17	OVSR	RW	0x0	过采样率控制: 000: 采集2次 001: 采集4次 010: 采集8次 011: 采集16次 100: 采集32次 101: 采集64次 110: 采集128次 111: 采集256次
16	OVSEN	RW	0x0	过采样使能: 0: 禁止过采样 1: 使能过采样

比特	名称	属性	复位值	描述
15	RSV	-	-	保留
14	IOTRFEN	RW	0x0	引脚触发信号数字滤波使能： 0: 禁止数字滤波 1: 使能数字滤波
13:12	TRGCFG	RW	0x0	硬件触发信号使能和极性选择： 00: 禁止硬件触发 01: 上升沿触发 10: 下降沿触发 11: 上升、下降沿都触发
11	SEMI	RW	0x0	单次转换半自动模式，仅在单次转换 (CONT=0) 时有效 0: 自动模式 1: 半自动模式
10	WAIT	RW	0x0	等待模式控制： 0: 无等待，如果上次转换数据没有及时读取，则可能出现Overrun 1: 等待模式，在上次转换数据被读取前，不会立刻启动下一次转换
9	CONT	RW	0x0	连续转换模式使能： 0: 单次转换 1: 连续转换
8	OVRM	RW	0x0	Overrun模式控制： 0: 当overrun发生时，保持上次数据，丢弃本次转换值 1: 当overrun发生时，覆盖上次数据
7:1	RSV	-	-	保留
0	DMAEN	RW	0x0	DMA使能： 0: 禁止DMA 1: 使能DMA

#### 21.4.5 ADC 采样时间控制寄存器 (ADC\_SMTR) (偏移: 010h)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20:16	CKD	RW	0x0	ADCCLK分频寄存器： 0x00、0x01: 系统时钟频率2分频 0x02: 系统时钟频率3分频 .....

比特	名称	属性	复位值	描述
				0x1f: 系统时钟频率32分频
15:12	RSV	-	-	保留
11:8	CHCG	RW	0x8	ADC采样通道切换等待时间: 0000、0001、0010: $2 * t_{ADCCCLK}$ 0011: $3 * t_{ADCCCLK}$ 0100: $4 * t_{ADCCCLK}$ 0101: $5 * t_{ADCCCLK}$ 0110: $6 * t_{ADCCCLK}$ 0111: $7 * t_{ADCCCLK}$ 1000: $8 * t_{ADCCCLK}$ 1001: $9 * t_{ADCCCLK}$ 1010: $10 * t_{ADCCCLK}$ 其他: $11 * t_{ADCCCLK}$ 注: $t_{ADCCCLK}$ 为ADCCCLK时钟周期
7:0	RSV	-	-	保留

#### 21.4.6 ADC 通道控制寄存器 (ADC\_CHER) (偏移: 014h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21	CH_EN	R	0x1	模拟通道使能: 0: 使能 1: 关闭
20:17	CUR	RW	0x0	BUFF和CMP偏置电流配置
16	RDTEST	RW	0x1	写'0'进入RD测试模式
15:11	CAPTEST	RW	0x0	1.5V输入信号电容失配选择 (用于RD测试)
10:9	VCMSEL	RW	0x0	CMP VCM基准电压选择
8:7	AZDELY	RW	0x0	1.5V输入信号延时选择
6:5	CMPOPT	RW	0x0	CMP增益选择
4:3	VREFOPT	RW	0x0	ADC positive reference电压选择: 0: disable (floating) 1: VREFPAVDD 2: VREFPLDO 3: PD3
2:0	SEQNUM	RW	0x0	ADC通道序列的总长度设置位, 一次AD转换过程的模拟通道总数为SEQNUM +1

### 21.4.7 ADC 通道选择寄存器 (ADC\_SEQ) (偏移: 018h)

比特	名称	属性	复位值	描述
31:28	SEQ7	RW	0x0	模拟通道 7 选择寄存器组, 分别指定对应的模拟输入通道。
27:24	SEQ6	RW	0x0	模拟通道 6 选择寄存器组, 分别指定对应的模拟输入通道。
23:20	SEQ5	RW	0x0	模拟通道 5 选择寄存器组, 分别指定对应的模拟输入通道。
19:16	SEQ4	RW	0x0	模拟通道 4 选择寄存器组, 分别指定对应的模拟输入通道。
15:12	SEQ3	RW	0x0	模拟通道 3 选择寄存器组, 分别指定对应的模拟输入通道。
11:8	SEQ2	RW	0x0	模拟通道 2 选择寄存器组, 分别指定对应的模拟输入通道。
7:4	SEQ1	RW	0x0	模拟通道 1 选择寄存器组, 分别指定对应的模拟输入通道。
3:0	SEQ0	RW	0x0	模拟通道 0 选择寄存器组, 分别指定对应的模拟输入通道。

### 21.4.8 ADC 模拟看门狗阈值寄存器 (ADC\_HLTR) (偏移: 020h)

比特	名称	属性	复位值	描述
31:16	AWD_HT	RW	0x0	AWD 监视高阈值
15:0	AWD_LT	RW	0x0	AWD 监视低阈值

### 21.4.9 ADC 注入通道控制寄存器 (ADC\_IJCHER) (偏移: 024h)

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
14	IJIOTRFEN	RW	0x0	注入通道引脚触发信号数字滤波使能: 0: 禁止数字滤波 1: 使能数字滤波
13:12	IJTRGCFG	RW	0x0	注入通道触发信号使能和极性选择: 00: 禁止触发 01: 上升沿触发

比特	名称	属性	复位值	描述
				10: 下降沿触发 11: 上升、下降沿都触发
11:5	RSV	-	-	保留
4	IJ_SEMI	RW	0x0	ADC注入通道转换模式选择位: 0: 一次触发转换所有通道 1: 一次触发转换一个通道 注: 该功能仅在ADC_CFGR.CONT = 0且 ADC_CFGR.SEMI = 1时有效。
3	IJ_AUTO	RW	0x0	ADC注入通道自动模式选择位: 0: 禁止自动模式 1: 使能自动模式
2:1	IJ_SEQNUM	RW	0x0	ADC注入通道通道序列的总长度设置位, 一次 AD转换过程的模拟通道总数为IJ_SEQNUM +1
0	IJ_EN	RW	0x0	ADC注入通道使能位: 0: 禁止注入通道 1: 使能注入通道

#### 21.4.10 ADC 注入通道选择寄存器 (ADC\_IJSEQ) (偏移: 028h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IJ_SEQ3	RW	0x0	模拟通道3选择寄存器组, 分别指定对应的模 拟输入通道。
11:8	IJ_SEQ2	RW	0x0	模拟通道2选择寄存器组, 分别指定对应的模 拟输入通道。
7:4	IJ_SEQ1	RW	0x0	模拟通道1选择寄存器组, 分别指定对应的模 拟输入通道。
3:0	IJ_SEQ0	RW	0x0	模拟通道0选择寄存器组, 分别指定对应的模 拟输入通道。

#### 21.4.11 ADC 数据寄存器 0 (ADC\_DR0) (偏移: 030h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器0

**21.4.12 ADC 数据寄存器 1 (ADC\_DR1) (偏移: 034h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器1

**21.4.13 ADC 数据寄存器 2 (ADC\_DR2) (偏移: 038h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器2

**21.4.14 ADC 数据寄存器 3 (ADC\_DR3) (偏移: 03Ch)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器3

**21.4.15 ADC 数据寄存器 4 (ADC\_DR4) (偏移: 040h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器4

**21.4.16 ADC 数据寄存器 5 (ADC\_DR5) (偏移: 044h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器5

**21.4.17 ADC 数据寄存器 6 (ADC\_DR6) (偏移: 048h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器6

**21.4.18 ADC 数据寄存器 7 (ADC\_DR7) (偏移: 04Ch)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC结果寄存器7

**21.4.19 ADC 注入通道数据寄存器 0 (ADC\_IJDR0) (偏移: 050h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC注入通道结果寄存器0

**21.4.20 ADC 注入通道数据寄存器 1 (ADC\_IJDR1) (偏移: 054h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC注入通道结果寄存器1

**21.4.21 ADC 注入通道数据寄存器 2 (ADC\_IJDR2) (偏移: 058h)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC注入通道结果寄存器2

**21.4.22 ADC 注入通道数据寄存器 3 (ADC\_IJDR3) (偏移: 05Ch)**

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	DATA	R	0x0	ADC注入通道结果寄存器3

**21.4.23 ADC 过采样平均结果寄存器 (ADC\_DR\_OVS) (偏移: 060h)**

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19:0	DATA	R	0x0	ADC过采样平均结果

## 21.4.24 ADC 硬件触发使能配置寄存器 (ADC\_HDT) (偏移: 0A8h)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20	ATIMER_OC45TRGO_HT	RW	0x0	ATIMER OC4/OC5 TRGO 硬件触发使能: 1: ATIMER OC4/OC5 TRGO 硬件触发使能 0: ATIMER OC4/OC5 TRGO 硬件触发关闭
19	ATIMER_OC5TRGO_HT	RW	0x0	ATIMER OC5 TRGO 硬件触发使能: 1: ATIMER OC5 TRGO 硬件触发使能 0: ATIMER OC5 TRGO 硬件触发关闭
18	ATIMER_OC4TRGO_HT	RW	0x0	ATIMER OC4 TRGO 硬件触发使能: 1: ATIMER OC4 TRGO 硬件触发使能 0: ATIMER OC4 TRGO 硬件触发关闭
17	PD3_HT	RW	0x0	PD3 硬件触发使能: 1: PD3 硬件触发使能 0: PD3硬件触发关闭
16	PD1_HT	RW	0x0	PD1 硬件触发使能: 1: PD1 硬件触发使能 0: PD1硬件触发关闭
15	PD0_HT	RW	0x0	PD0 硬件触发使能: 1: PD0 硬件触发使能 0: PD0 硬件触发关闭
14	PC4_HT	RW	0x0	PC4 硬件触发使能: 1: PC4 硬件触发使能 0: PC4 硬件触发关闭
13	PC3_HT	RW	0x0	PC3 硬件触发使能: 1: PC3 硬件触发使能 0: PC3 硬件触发关闭
12	PC2_HT	RW	0x0	PC2 硬件触发使能: 1: PC2 硬件触发使能 0: PC2 硬件触发关闭
11	RSV	-	-	保留
10	LVD_HT	RW	0x0	LVD 硬件触发使能: 1: LVD 硬件触发使能 0: LVD 硬件触发关闭

比特	名称	属性	复位值	描述
9	LPTIMER_HT2	RW	0x0	LPTIMER OUT2 硬件触发使能: 1: LPTIMER OUT2 硬件触发使能 0: LPTIMER OUT2 硬件触发关闭
8	LPTIMER_HT1	RW	0x0	LPTIMER OUT1 硬件触发使能: 1: LPTIMER OUT1 硬件触发使能 0: LPTIMER OUT1 硬件触发关闭
7	GTIMER2_TRGO_HT	RW	0x0	GTIMER2 TRGO 硬件触发使能: 1: GTIMER2 TRGO 硬件触发使能 0: GTIMER2 TRGO 硬件触发关闭
6	GTIMER1_TRGO_HT	RW	0x0	GTIMER1 TRGO 硬件触发使能: 1: GTIMER1 TRGO 硬件触发使能 0: GTIMER1 TRGO 硬件触发关闭
5	GTIMER0_TRGO_HT	RW	0x0	GTIMER0 TRGO 硬件触发使能: 1: GTIMER0 TRGO 硬件触发使能 0: GTIMER0 TRGO 硬件触发关闭
4	CMP1_HT	RW	0x0	CMP1 硬件触发使能: 1: CMP1 硬件触发使能 0: CMP1 硬件触发关闭
3	CMP0_HT	RW	0x0	CMP0 硬件触发使能: 1: CMP0 硬件触发使能 0: CMP0 硬件触发关闭
2	ATIMER_OC5REF_HT	RW	0x0	ATIMER OC5 REF 硬件触发使能: 1: ATIMER OC5 REF 硬件触发使能 0: ATIMER OC5 REF 硬件触发关闭
1	ATIMER_OC4REF_HT	RW	0x0	ATIMER OC4 REF 硬件触发使能: 1: ATIMER OC4 REF 硬件触发使能 0: ATIMER OC4 REF 硬件触发关闭
0	ATIMER_TRGO_HT	RW	0x0	ATIMER TRGO 硬件触发使能: 1: ATIMER TRGO 硬件触发使能 0: ATIMER TRGO 硬件触发关闭

## 21.4.25 ADC 注入通道硬件触发使能配置寄存器 (ADC\_IJHDT) (偏移: 0ACh)

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20	ATIMER_OC45TRGO_IJHT	RW	0x0	ATIMER OC4/OC5 TRGO 硬件触发使能: 1: ATIMER OC4/OC5 TRGO 硬件触发使能 0: ATIMER OC4/OC5 TRGO 硬件触发关闭
19	ATIMER_OC5TRGO_IJHT	RW	0x0	ATIMER OC5 TRGO 硬件触发使能: 1: ATIMER OC5 TRGO 硬件触发使能 0: ATIMER OC5 TRGO 硬件触发关闭
18	ATIMER_OC4TRGO_IJHT	RW	0x0	ATIMER OC4 TRGO 硬件触发使能: 1: ATIMER OC4 TRGO 硬件触发使能 0: ATIMER OC4 TRGO 硬件触发关闭
17	PD3_IJHT	RW	0x0	PD3 硬件触发使能: 1: PD3 硬件触发使能 0: PD3硬件触发关闭
16	PD1_IJHT	RW	0x0	PD1 硬件触发使能: 1: PD1 硬件触发使能 0: PD1硬件触发关闭
15	PD0_IJHT	RW	0x0	PD0 硬件触发使能: 1: PD0 硬件触发使能 0: PD0 硬件触发关闭
14	PC4_IJHT	RW	0x0	PC4 硬件触发使能: 1: PC4 硬件触发使能 0: PC4 硬件触发关闭
13	PC3_IJHT	RW	0x0	PC3 硬件触发使能: 1: PC3 硬件触发使能 0: PC3 硬件触发关闭
12	PC2_IJHT	RW	0x0	PC2 硬件触发使能: 1: PC2 硬件触发使能 0: PC2 硬件触发关闭
11	RSV	-	-	保留
10	LVD_IJHT	RW	0x0	LVD 硬件触发使能: 1: LVD 硬件触发使能 0: LVD 硬件触发关闭

比特	名称	属性	复位值	描述
9	LPTIMER_IJHT2	RW	0x0	LPTIMER OUT2 硬件触发使能: 1: LPTIMER OUT2 硬件触发使能 0: LPTIMER OUT2 硬件触发关闭
8	LPTIMER_IJHT1	RW	0x0	LPTIMER OUT1 硬件触发使能: 1: LPTIMER OUT1 硬件触发使能 0: LPTIMER OUT1 硬件触发关闭
7	GTIMER2_TRGO_IJHT	RW	0x0	GTIMER2 TRGO 硬件触发使能: 1: GTIMER2 TRGO 硬件触发使能 0: GTIMER2 TRGO 硬件触发关闭
6	GTIMER1_TRGO_IJHT	RW	0x0	GTIMER1 TRGO 硬件触发使能: 1: GTIMER1 TRGO 硬件触发使能 0: GTIMER1 TRGO 硬件触发关闭
5	GTIMER0_TRGO_IJHT	RW	0x0	GTIMER0 TRGO 硬件触发使能: 1: GTIMER0 TRGO 硬件触发使能 0: GTIMER0 TRGO 硬件触发关闭
4	CMP1_IJHT	RW	0x0	CMP1 硬件触发使能: 1: CMP1 硬件触发使能 0: CMP1 硬件触发关闭
3	CMP0_IJHT	RW	0x0	CMP0 硬件触发使能: 1: CMP0 硬件触发使能 0: CMP0 硬件触发关闭
2	ATIMER_OC5REF_IJHT	RW	0x0	ATIMER OC5 REF 硬件触发使能: 1: ATIMER OC5 REF 硬件触发使能 0: ATIMER OC5 REF 硬件触发关闭
1	ATIMER_OC4REF_IJHT	RW	0x0	ATIMER OC4 REF 硬件触发使能: 1: ATIMER OC4 REF 硬件触发使能 0: ATIMER OC4 REF 硬件触发关闭
0	ATIMER_TRGO_IJHT	RW	0x0	ATIMER TRGO 硬件触发使能: 1: ATIMER TRGO 硬件触发使能 0: ATIMER TRGO 硬件触发关闭

## 21.4.26 ADC 通道采样时间配置寄存器 0 (ADC\_SMTR0) (偏移: 0B0h)

比特	名称	属性	复位值	描述
31:24	SMTR3	RW	0x0	<p>用于配置模拟通道3的采样时间 (ADC_SMTR3+2) * <math>t_{ADCCCLK}</math>, 即:</p> <p>8'd0: <math>2 * t_{ADCCCLK}</math>  8'd1: <math>3 * t_{ADCCCLK}</math>  8'd2: <math>4 * t_{ADCCCLK}</math>  ...  8'd253: <math>255 * t_{ADCCCLK}</math>  8'd254: <math>256 * t_{ADCCCLK}</math>  8'd255: <math>257 * t_{ADCCCLK}</math></p> <p>注: <math>t_{ADCCCLK}</math> 为 ADCCLK 时钟周期。</p>
23:16	SMTR2	RW	0x0	<p>用于配置模拟通道2的采样时间 (ADC_SMTR2+2) * <math>t_{ADCCCLK}</math>, 即:</p> <p>8'd0: <math>2 * t_{ADCCCLK}</math>  8'd1: <math>3 * t_{ADCCCLK}</math>  8'd2: <math>4 * t_{ADCCCLK}</math>  ...  8'd253: <math>255 * t_{ADCCCLK}</math>  8'd254: <math>256 * t_{ADCCCLK}</math>  8'd255: <math>257 * t_{ADCCCLK}</math></p> <p>注: <math>t_{ADCCCLK}</math> 为 ADCCLK 时钟周期。</p>
15:8	SMTR1	RW	0x0	<p>用于配置模拟通道1的采样时间 (ADC_SMTR1+2) * <math>t_{ADCCCLK}</math>, 即:</p> <p>8'd0: <math>2 * t_{ADCCCLK}</math>  8'd1: <math>3 * t_{ADCCCLK}</math>  8'd2: <math>4 * t_{ADCCCLK}</math>  ...  8'd253: <math>255 * t_{ADCCCLK}</math>  8'd254: <math>256 * t_{ADCCCLK}</math>  8'd255: <math>257 * t_{ADCCCLK}</math></p> <p>注: <math>t_{ADCCCLK}</math> 为 ADCCLK 时钟周期。</p>

比特	名称	属性	复位值	描述
7:0	SMTR0	RW	0x0	用于配置模拟通道0的采样时间 $(ADC\_SMTR0+2) * t_{ADCCLK}$ ， 即： 8'd0: $2 * t_{ADCCLK}$ 8'd1: $3 * t_{ADCCLK}$ 8'd2: $4 * t_{ADCCLK}$ ... 8'd253: $255 * t_{ADCCLK}$ 8'd254: $256 * t_{ADCCLK}$ 8'd255: $257 * t_{ADCCLK}$ 注: $t_{ADCCLK}$ 为 ADCCLK 时钟周期。

#### 21.4.27 ADC 通道采样时间配置寄存器 1 (ADC\_SMTR1) (偏移: 0B4h)

比特	名称	属性	复位值	描述
31:24	SMTR7	RW	0x0	用于配置模拟通道7的采样时间 $(ADC\_SMTR7+2) * t_{ADCCLK}$ ， 即： 8'd0: $2 * t_{ADCCLK}$ 8'd1: $3 * t_{ADCCLK}$ 8'd2: $4 * t_{ADCCLK}$ ... 8'd253: $255 * t_{ADCCLK}$ 8'd254: $256 * t_{ADCCLK}$ 8'd255: $257 * t_{ADCCLK}$ 注: $t_{ADCCLK}$ 为 ADCCLK 时钟周期。
23:16	SMTR6	RW	0x0	用于配置模拟通道6的采样时间 $(ADC\_SMTR6+2) * t_{ADCCLK}$ ， 即： 8'd0: $2 * t_{ADCCLK}$ 8'd1: $3 * t_{ADCCLK}$ 8'd2: $4 * t_{ADCCLK}$ ... 8'd253: $255 * t_{ADCCLK}$ 8'd254: $256 * t_{ADCCLK}$ 8'd255: $257 * t_{ADCCLK}$ 注: $t_{ADCCLK}$ 为 ADCCLK 时钟周期。

比特	名称	属性	复位值	描述
15:8	SMTR5	RW	0x0	用于配置模拟通道5的采样时间 $(ADC\_SMTR5+2) * t_{ADCCLK}$ ， 即： 8'd0: $2 * t_{ADCCLK}$ 8'd1: $3 * t_{ADCCLK}$ 8'd2: $4 * t_{ADCCLK}$ ... 8'd253: $255 * t_{ADCCLK}$ 8'd254: $256 * t_{ADCCLK}$ 8'd255: $257 * t_{ADCCLK}$ 注: $t_{ADCCLK}$ 为 ADCCLK 时钟周期。
7:0	SMTR4	RW	0x0	用于配置模拟通道4的采样时间 $(ADC\_SMTR4+2) * t_{ADCCLK}$ ， 即： 8'd0: $2 * t_{ADCCLK}$ 8'd1: $3 * t_{ADCCLK}$ 8'd2: $4 * t_{ADCCLK}$ ... 8'd253: $255 * t_{ADCCLK}$ 8'd254: $256 * t_{ADCCLK}$ 8'd255: $257 * t_{ADCCLK}$ 注: $t_{ADCCLK}$ 为 ADCCLK 时钟周期。

#### 21.4.28 ADC 注入通道采样时间配置寄存器(ADC\_IJSMTR) (偏移: 0B8h)

比特	名称	属性	复位值	描述
31:24	IJSMTR3	RW	0x0	用于配置注入通道3的采样时间 $(ADC\_IJSMTR3+2) * t_{ADCCLK}$ ， 即： 8'd0: $2 * t_{ADCCLK}$ 8'd1: $3 * t_{ADCCLK}$ 8'd2: $4 * t_{ADCCLK}$ ... 8'd253: $255 * t_{ADCCLK}$ 8'd254: $256 * t_{ADCCLK}$ 8'd255: $257 * t_{ADCCLK}$ 注: $t_{ADCCLK}$ 为 ADCCLK 时钟周期

比特	名称	属性	复位值	描述
23:16	IJSMTR2	RW	0x0	<p>用于配置注入通道2的采样时间 <math>(ADC\_IJSMTR2+2) * t_{ADCCLK}</math>，即：</p> <p>8'd0: <math>2 * t_{ADCCLK}</math>            8'd1: <math>3 * t_{ADCCLK}</math>            8'd2: <math>4 * t_{ADCCLK}</math>            ...            8'd253: <math>255 * t_{ADCCLK}</math>            8'd254: <math>256 * t_{ADCCLK}</math>            8'd255: <math>257 * t_{ADCCLK}</math></p> <p>注: <math>t_{ADCCLK}</math> 为 ADCCLK 时钟周期。</p>
15:8	IJSMTR1	RW	0x0	<p>用于配置注入通道1的采样时间 <math>(ADC\_IJSMTR1+2) * t_{ADCCLK}</math>，即：</p> <p>8'd0: <math>2 * t_{ADCCLK}</math>            8'd1: <math>3 * t_{ADCCLK}</math>            8'd2: <math>4 * t_{ADCCLK}</math>            ...            8'd253: <math>255 * t_{ADCCLK}</math>            8'd254: <math>256 * t_{ADCCLK}</math>            8'd255: <math>257 * t_{ADCCLK}</math></p> <p>注: <math>t_{ADCCLK}</math> 为 ADCCLK 时钟周期。</p>
7:0	IJSMTR0	RW	0x0	<p>用于配置注入通道0的采样时间 <math>(ADC\_IJSMTR0+2) * t_{ADCCLK}</math>，即：</p> <p>8'd0: <math>2 * t_{ADCCLK}</math>            8'd1: <math>3 * t_{ADCCLK}</math>            8'd2: <math>4 * t_{ADCCLK}</math>            ...            8'd253: <math>255 * t_{ADCCLK}</math>            8'd254: <math>256 * t_{ADCCLK}</math>            8'd255: <math>257 * t_{ADCCLK}</math></p> <p>注: <math>t_{ADCCLK}</math> 为 ADCCLK 时钟周期。</p>

## 21.5 ADC 使用流程

### 21.5.1 单次扫描模式单通道 A/D 转换

单次扫描模式下，ADC 启动转换后只执行一次转换。

1. 配置 ADC\_CR.ADEN 为 0，模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK，设置 ADC 时钟分频。
3. 配置 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1，设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT，设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 0，选择单次扫描模式。
6. 配置 ADC\_CFGR.SEMI，选择单次触发模式（半自动/全自动）。
7. 配置 ADC\_CHER.SEQNUM 为 0，设置扫描序列为单通道。
8. 配置 ADC\_SEQ.SEQ0，设置通道对应的模拟输入通道。
9. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口（PAD\_ADS）。
10. 配置 ADC\_CR.ADEN 为 1，启用 ADC 模块。
11. 配置 ADC\_CR.START 为 1，启动 ADC 转换。
12. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1，读取对应通道 ADC\_DR 中的数据。
13. 如需进行多次单次转换，则重复执行步骤 11 和 12。
14. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断，则等待中断触发后读取 ADC\_DR。

### 21.5.2 单次扫描模式多通道 A/D 转换

单次扫描模式下，同时启用了多个通道时，开启 ADC 转换后，会依次按通道 SEQ0~SEQ7 进行转换，其中通道 SEQ0~SEQ7 对应的输入通道可配，当通道 SEQ7 转换完成，就代表此次单次扫描转换完成。

1. 配置 ADC\_CR.ADEN 为 0，模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK，设置 ADC 时钟分频。

3. 配置 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1, 设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT, 设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 0, 选择单次扫描模式。
6. 配置 ADC\_CFGR.SEMI, 选择单次转换触发模式 (半自动/全自动)。
7. 配置 ADC\_CHER.SEQNUM 为 n, 设置扫描序列的通道个数。
8. 配置 ADC\_SEQ, 设置各个通道对应的模拟输入通道 (即设置通道优先级)。
9. 根据 ADC 输入通道对应的 GPIO 管脚, 将待转换通道配置为模拟接口 (PAD\_ADS)。
10. 配置 ADC\_CR.ADEN 为 1, 启用 ADC 模块。
11. 配置 ADC\_CR.START 为 1, 启动 ADC 转换。
12. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1, 读取对应通道 ADC\_DR 中的数据。
13. 重复执行上述操作 (如果配置单次转换半自动触发模式, 重复执行步骤 11 和 12, 全自动触发则只需重复执行步骤 12) 直到 ADC\_ISR.EOS 转换序列结束标志位为 1, 表示单次转换中所有通道转换完成。
14. 如需进行多次单次转换, 则重复执行步骤 12、13、14。
15. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断, 则等待中断触发后读取 ADC\_DR。

注:

- 在使用多通道转换时, 可以等待 ADC\_ISR.EOS 转换序列结束标志有效后, 再去读取每个通道的 ADC\_DR 中的数据。
- 使用多通道转换时, 若采样精度不足, 建议适当修改 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1 的值。

### 21.5.3 连续扫描模式单通道 A/D 转换

连续扫描模式下, 启动一次 ADC 转换后会对所选通道进行不断地连续转换, 将 ADC\_CR.STOP 写 1 可停止转换。

1. 配置 ADC\_CR.ADEN 为 0, 模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK, 设置 ADC 时钟分频。

3. 配置 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1, 设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT, 设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 1, 选择连续扫描模式。
6. 配置 ADC\_CHER.SEQNUM 为 0, 设置扫描序列为单通道。
7. 配置 ADC\_SEQ.SEQ0, 设置通道对应的模拟输入通道。
8. 根据 ADC 输入通道对应的 GPIO 管脚, 将待转换通道配置为模拟接口 (PAD\_ADS)。
9. 配置 ADC\_CR.ADEN 为 1, 启用 ADC 模块。
10. 配置 ADC\_CR.START 为 1, 启动 ADC 转换。
11. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1, 读取对应通道 ADC\_DR 中的数据。
12. 如需读取多个 ADC 数据, 则重复以上 1 个步骤。
13. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断, 则等待中断触发后读取 ADC\_DR。

#### 21.5.4 连续扫描模式多通道 A/D 转换

连续扫描模式下, 启动一次 ADC 转换后会对所选通道进行不断地连续转换, 将 ADC\_CR.STOP 写 1 可停止转换。当同时启用了多个通道时, 会依次按通道 SEQ0~SEQ7 进行转换, 其中通道 SEQ0~SEQ7 对应的输入通道可配, 当通道 SEQ7 转换完成, 就代表此次单次扫描转换完成。

1. 配置 ADC\_CR.ADEN 为 0, 模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK, 设置 ADC 时钟分频。
3. 配置 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1, 设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT, 设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 1, 选择连续扫描模式。
6. 配置 ADC\_CHER.SEQNUM 为 n, 设置扫描序列的通道个数。
7. 配置 ADC\_SEQ, 设置各个通道对应的模拟输入通道 (即设置通道优先级)。
8. 根据 ADC 输入通道对应的 GPIO 管脚, 将待转换通道配置为模拟接口 (PAD\_ADS)。
9. 配置 ADC\_CR.ADEN 为 1, 启用 ADC 模块。

10. 配置 ADC\_CR.START 为 1，启动 ADC 转换。
11. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1，读取对应通道 ADC\_DR 中的数据。
12. 如需读取多个 ADC 数据，则重复以上 1 个步骤。
13. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断，则等待中断触发后读取 ADC\_DR。

注:

- 在使用多通道转换时，可以等待 ADC\_ISR.EOS 转换序列结束标志有效后，再去读取每个通道的 ADC\_DR 中的数据。
- 使用多通道转换时，若采样精度不足，建议适当修改 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1 的值。

### 21.5.5 硬件触发事件 A/D 转换

硬件触发事件只能触发单次扫描模式，ADC 启动转换后只执行一次转换。

1. 配置 ADC\_CR.ADEN 为 0，模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK，设置 ADC 时钟分频。
3. 配置 ADC\_SMTR.CHCG 和 ADC\_SMTR0/1，设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT，设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 0，选择单次扫描模式。
6. 配置 ADC\_CFGR.SEMI，选择单次触发模式（半自动/全自动）。
7. 配置 ADC\_CHER.SEQNUM 为 0，设置扫描序列为单通道。
8. 配置 ADC\_SEQ.SEQ0，设置通道对应的模拟输入通道。
9. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口（PAD\_ADS）。
10. 配置 ADC\_CR.ADEN 为 1，启用 ADC 模块。
11. 对需要作为硬件触发 ADC 转换的硬件初始化。
12. 配置 ADC\_HDT 相应的硬件触发事件为 1，使能硬件触发。
13. 配置 ADC\_CFGR.IOTRFEN 引脚触发信号数字滤波。
14. 配置 ADC\_CFGR.TRGCFG，选择硬件触发事件的极性。

15. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1，读取对应通道 ADC\_DR 中的数据。
16. 如需进行多次单次转换，则硬件多次触发事件即可。
17. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断，则等待中断触发后读取 ADC\_DR。

### 21.5.6 注入通道普通注入模式 A/D 转换

ADC 最多可以注入四个转换通道，注入通道的转换和正常转换类似，转换结果存储在 ADC\_IJDR0~ADC\_IJDR3 中。

1. 配置 ADC\_CR.ADEN 为 0，模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK，设置 ADC 时钟分频。
3. 配置 ADC\_SMTR.CHCG 和 ADC\_IJSMTR，设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT，设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 0，选择单次扫描模式。
6. 配置 ADC\_CFGR.SEMI，选择单次转换触发模式（半自动/全自动）。
7. 配置 ADC\_CHER.SEQNUM 为 n，设置扫描序列的通道个数。
8. 配置 ADC\_SEQ，设置各个通道对应的模拟输入通道（即设置通道优先级）。
9. 配置 ADC\_IJCHER.IJ\_AUTO 为 0，设置注入通道为普通注入模式。
10. 配置 ADC\_IJCHER.IJ\_SEQNUM 为 n，设置注入通道扫描序列的通道个数。
11. 配置 ADC\_IJSEQ，设置注入通道各个通道对应的模拟输入通道（即设置通道优先级）。
12. 配置 ADC\_IJCHER.IJ\_EN 为 1，使能注入通道。
13. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口（PAD\_ADS）。
14. 配置 ADC\_CR.ADEN 为 1，启用 ADC 模块。
15. 配置 ADC\_CR.START 为 1，启动 ADC 转换。
16. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1，读取对应通道 ADC\_DR 中的数据。
17. 配置 ADC\_CR.IJSTART 为 1，启动 ADC 注入通道转换。
18. 等待 ADC\_ISR.IJ\_EOC 单次转换结束标志位为 1，读取对应通道 ADC\_IJDR 中的数据。

19. 重复执行上述操作直到 ADC\_ISR.EOS 和 ADC\_ISR.IJ\_EOS 转换序列结束标志位为 1，表示单次转换中所有通道和注入通道转换完成。
20. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断，则等待中断触发后读取 ADC\_DR。如使能了 ADC\_IER.IJ\_EOCIE 和 ADC\_IER.IJ\_EOSIE 中断，则等待中断触发后读取 ADC\_IJDR。

### 21.5.7 注入通道自动注入模式 A/D 转换

ADC 最多可以注入四个转换通道，注入通道的转换和正常转换类似，转换结果存储在 ADC\_IJDR0~ADC\_IJDR3 中。

1. 配置 ADC\_CR.ADEN 为 0，模拟 ADC 复位释放和上电。
2. 配置 ADC\_SMTR.CDK，设置 ADC 时钟分频。
3. 配置 ADC\_SMTR.CHCG 和 ADC\_IJSMTR，设置 ADC 转换速度。
4. 配置 ADC\_CHER.VREFOPT，设置 ADC 参考电压源。
5. 配置 ADC\_CFGR.CONT 为 0，选择单次扫描模式。
6. 配置 ADC\_CFGR.SEMI，选择单次转换触发模式（半自动/全自动）。
7. 配置 ADC\_CHER.SEQNUM 为 n，设置扫描序列的通道个数。
8. 配置 ADC\_SEQ，设置各个通道对应的模拟输入通道（即设置通道优先级）。
9. 配置 ADC\_IJCHER.IJ\_AUTO 为 1，设置注入通道为自动注入模式。
10. 配置 ADC\_IJCHER.IJ\_SEQNUM 为 n，设置注入通道扫描序列的通道个数。
11. 配置 ADC\_IJSEQ，设置注入通道各个通道对应的模拟输入通道（即设置通道优先级）。
12. 配置 ADC\_IJCHER.IJ\_EN 为 1，使能注入通道。
13. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口（PAD\_ADS）。
14. 配置 ADC\_CR.ADEN 为 1，启用 ADC 模块。
15. 配置 ADC\_CR.START 为 1，启动 ADC 转换。
16. 等待 ADC\_ISR.EOC 单次转换结束标志位为 1，读取对应通道 ADC\_DR 中的数据。
17. 配置 ADC\_CR.IJSTART 为 1，启动 ADC 注入通道转换。

18. 重复执行上述操作直到 ADC\_ISR.EOS 转换序列结束标志位为 1, 表示单次转换中所有通道转换完成。
19. 等待 ADC\_ISR.IJ\_EOC 单次转换结束标志位为 1, 读取对应通道 ADC\_IJDR 中的数据。
20. 重复执行上述操作直到 ADC\_ISR.IJ\_EOS 转换序列结束标志位为 1, 表示单次转换中所有注入通道转换完成。
21. 如使能了 ADC\_IER.EOCIE 和 ADC\_IER.EOSIE 中断, 则等待中断触发后读取 ADC\_DR。如使能了 ADC\_IER.IJ\_EOCIE 和 ADC\_IER.IJ\_EOSIE 中断, 则等待中断触发后读取 ADC\_IJDR。

## 21.5.8 注意事项

- 每个通道被采样后, 数据保存在 ADC\_DR0~ADC\_DR7 寄存器中, 软件要在下次转换前及时读走数据, 或者通过 DMA 进行数据搬运。如果不能及时取走数据, 将引起 Overrun, 置位 overrun 标志, 并可以发出中断。ADC 控制器支持自动等待, 如果 WAIT 寄存器被软件置位, 那么在 ADC\_DATA 寄存器被读取之前, ADC 控制器不会发起新的转换; 在等待状态中到来的硬件触发事件也会被忽略。WAIT 寄存器在 DMA 模式下也同样有效, 即 DMA 没有读取上一次转换结果的情况下, ADC 控制器不会启动新的转换。
- Overrun 模式和 WAIT 模式同样支持注入通道的数据 ADC\_IJDR0~ADC\_IJDR3。
- DMA 只支持常规通道 ADC\_DR0~ADC\_DR7 的数据搬运, 不支持注入通道的数据搬运。
- 每产生一次硬件触发事件, ADC 只转换一次, 即为单次采样模式, 使能硬件触发时建议将 ADC 模式配置为单次扫描模式。

## 21.6 ADC 经 PGA 缓冲采样使用流程

### 21.6.1 ADC 经 PGA 缓冲采样图

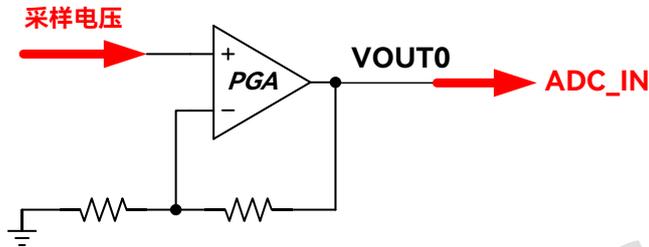


图 21-1: ADC 经 PGA 缓冲采样图

### 21.6.2 ADC 经 PGA 缓冲采样流程图

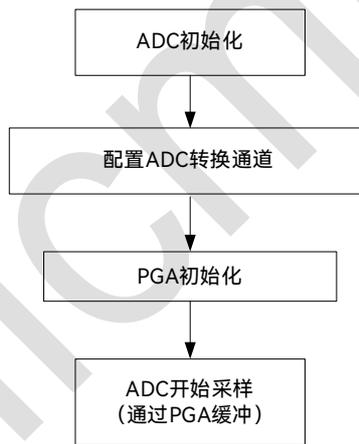


图 21-2: ADC 经 PGA 缓冲采样流程图

### 21.6.3 ADC 经 PGA 缓冲后采样使用流程

1. ADC 初始化。选择 ADC 参考电压源、ADCCLK 分频和采样模式。
2. 配置 ADC 转换通道。
3. PGA 初始化。配置成 PGA 正端选择采样电压（外部输入或 AVSS），PGA 负端通过内部连接输出，使能 ADC 输入通道经 PGA 缓冲。
4. ADC 开始进行采样。

## 22 COMP

### 22.1 概述

COMP 是一款具有轨到轨输入的迟滞比较器,输入端可根据需要配置。COMP 可用作电压比较,有两个输入端 IN+和 IN-,可选择其中一个输入端作为参考点来比较,当另一输入端电压小于参考电压时比较器输出低电平,反之输出高电平。

### 22.2 主要特性

- 2 个电压比较器
- 可产生比较中断
- 比较器精度可调
- 比较结果可 Pad 输出

## 22.3 COMP 功能框图

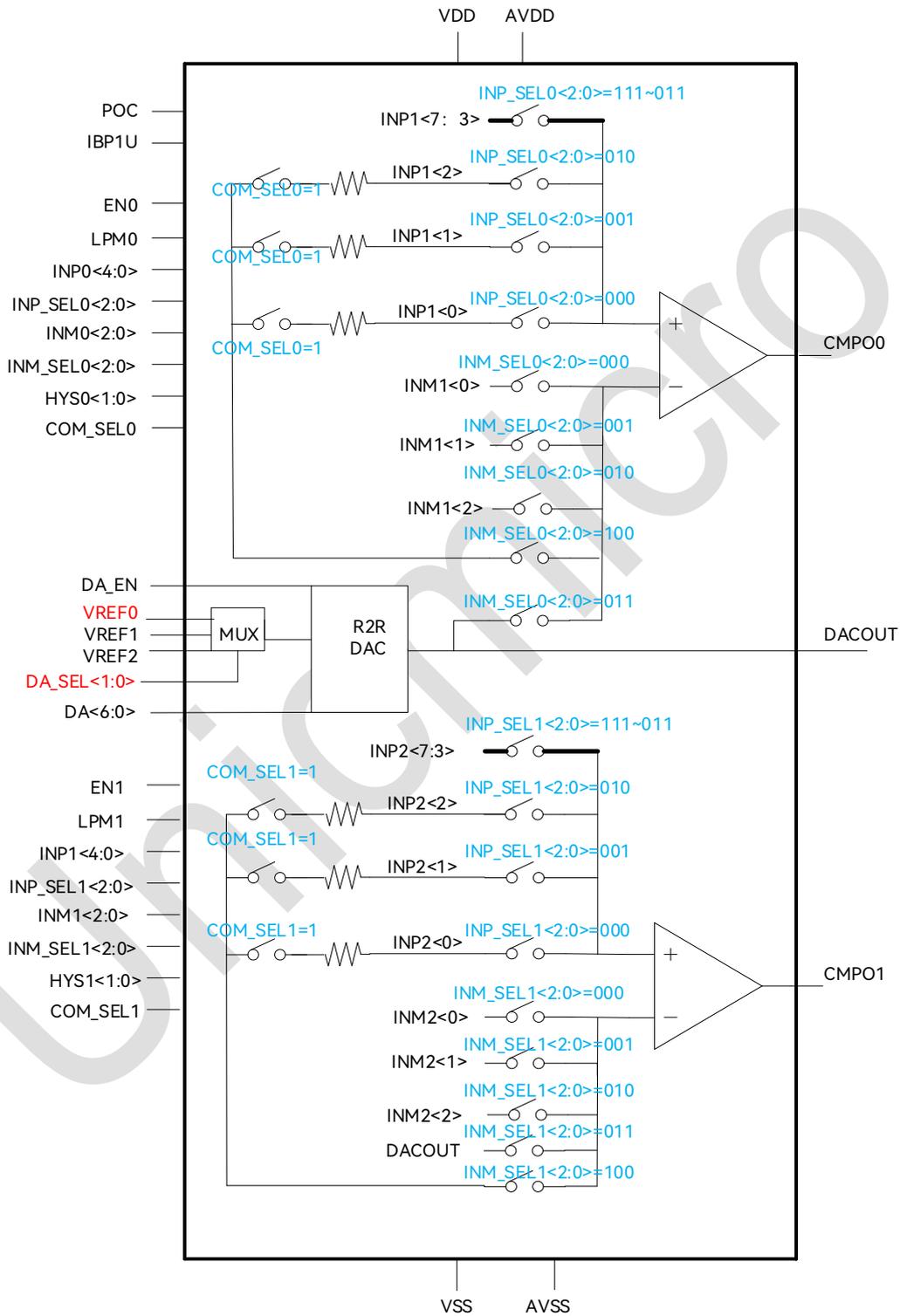


图 22-1: COMP 模拟功能框图

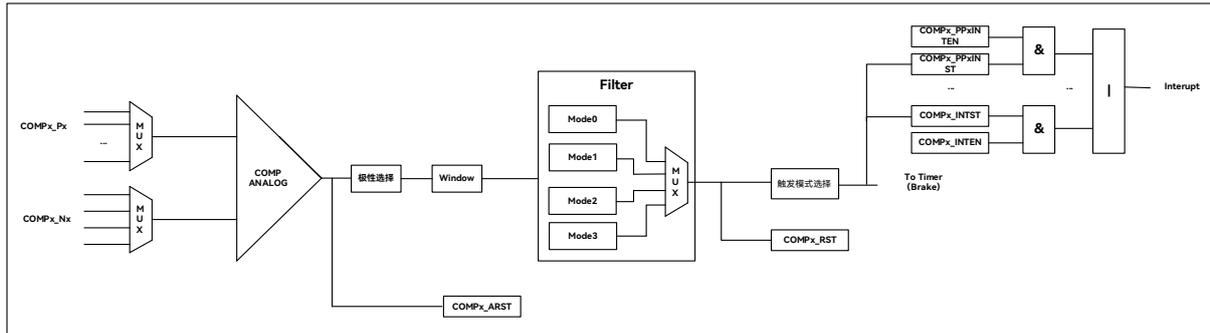


图 22-2: COMP 整体功能框图

## 22.4 COMP 滤波模式

在应用中，四种滤波模式在同一时刻仅可以选择一种滤波模式使用，不可并行使用两种以上的滤波模式。四种滤波模式的滤波时间均由 COMP0\_LVTSET.COMP0\_LVTSET 寄存器设置。**(软件加清 CNT 位)**

### 1. Mode 1

当模拟比较器的输出，经过极性选择设置之后，结果为 1 时，计数器开始计时。当计时器计数结果等于 COMP0\_LVTSET 寄存器设置值后，滤波结果为 1，即本次比较器比较结果有效。当模拟比较器的输出，经过极性选择设置之后，结果为 0 时，计数器的计数值清 0。

### 2. Mode 2

当模拟比较器的输出，经过极性选择设置之后，结果为 1 时，计数器开始计时。当计时器计数结果等于 COMP0\_LVTSET 寄存器设置值后，滤波结果为 1，即本次比较器比较结果有效。当模拟比较器的输出，经过极性选择设置之后，结果为 0 时，计数器的计数值保持不变。

### 3. Mode 3

当模拟比较器的输出，经过极性选择设置之后，结果为 1 时，计数器开始计时。当计时器计数结果等于 COMP0\_LVTSET 寄存器设置值后，滤波结果为 1，即本次比较器比较结果有效。当模拟比较器的输出，经过极性选择设置之后，结果为 0 时，计数器的计数值减 1，直到计时器值减为 0。

### 4. Mode 4

当模拟比较器的输出，经过极性选择设置之后，结果为 1 时，计数器开始计时。当计时器计数结果等于 COMP0\_LVTSET 寄存器设置值后，滤波结果为 1，即本次比较器比较结果有效。当模拟比

较器的输出，经过极性选择设置之后，结果为 0 时，计数器的计数值减 2，直到计时器值减为 0。

## 22.5 Window 功能

通过 WINCFG 寄存器设置选择 PWM 的边沿后一段时间，比较器的结果无效，不参与后面的滤波和结果的计算。

## 22.6 寄存器描述

寄存器基地址：0x40002000

表 22-1: COMP 寄存器列表

偏置	名称	描述
0x118	COMP0_CFG	COMP0 控制寄存器
0x11C	COMP1_CFG	COMP1 控制寄存器
0x128	COMP0_POLL	COMP0 轮询寄存器
0x12C	COMP1_POLL	COMP1 轮询寄存器
0x138	COMP0_POLLMASK	COMP0 轮询 MASK 寄存器
0x13C	COMP1_POLLMASK	COMP1 轮询 MASK 寄存器
0x148	COMP0_LVSET	COMP0 滤波设置寄存器
0x14C	COMP1_LVSET	COMP1 滤波设置寄存器
0x158	COMP0_WINCFG	COMP0 WINDOW 设置寄存器
0x15C	COMP1_WINCFG	COMP1 WINDOW 设置寄存器
0x168	COMP0_INTSTAT	COMP0 中断状态寄存器
0x16C	COMP1_INTSTAT	COMP1 中断状态寄存器
0x178	COMP0_INTEN	COMP0 中断使能寄存器
0x17C	COMP1_INTEN	COMP1 中断使能寄存器

### 22.6.1 COMP0 控制寄存器 COMP0\_CFG (偏移: 118h)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	LOCK	RW	0x0	COMP0 输出锁存位： 1: COPM0 保持当前输出值 0: COPM0 正常输出

比特	名称	属性	复位值	描述
26:20	DA_VSET	RW	0x0	DAC 输出电压设置: $VOUT = ((DA\_VSET+1) / 128) * VREF$
19:18	DA_VSSEL	RW	0x0	DAC 参考电压 (VREF) 选择: 00: VREF0 输入, VREFLDO 输入 01: VREF1 输入, PD3 输入 10: VREF2 输入, BUF 0.8V/1.5V 输入 11: 禁用
17	DA_EN	RW	0x0	DAC 使能位: 1: DAC 使能 0: DAC 禁止
16:15	EDSET	RW	0x0	COMP0 触发模式设置: 00: 电平模式 01: COMP0 检测结果上升沿锁存状态 10: COMP0 检测结果下降沿锁存状态 11: COMP0 检测结果上升沿或下降沿锁存状态
14:12	RSV	-	-	保留
11	COMNSEL	RW	0x0	COMP0 的 P 端 P0、P1、P2 共模电压设置位: 1: 开启共模电压功能 0: 禁止共模电压功能
10:9	HYS	RW	0x0	COMP0 迟滞电压设置位: 00: 0mV 01: 10mV 10: 20mV 11: 30mV
8	POL	RW	0x0	COMP0 极性选择: 1: COMP0 N 端电压高时输出 1 0: COMP0 P 端电压高时输出 1
7:5	INMSEL	RW	0x0	COMP0 负向输入端通道选择: 000: COMP0_N0 输入, PC0 输入 001: COMP0_N1 输入, PB5 输入 010: COMP0_N2 输入, PB1 输入 011: DAC 输入 100: 共模信号输入
4:2	INPSEL	RW	0x0	COMP0 正向输入端通道选择: 000: COMP0_P0 输入, PC1 输入 001: COMP0_P1 输入, PB7 输入 010: COMP0_P2 输入, PB6 输入

比特	名称	属性	复位值	描述
				011: COMP0_P3 输入, PB4 输入 100: COMP0_P4 输入, PC2 输入 101: COMP0_P5 输入, PGA0_VOUT 输入 110: COMP0_P6 输入, PGA1_VOUT 输入 111: COMP0_P7 输入, PGA2_VOUT 输入
1	LMP	RW	0x0	COMP0 低功耗模式控制信号: 1: 高速 0: 低速
0	EN	RW	0x0	COMP0 使能: 1: 使能 COMP0 0: 关闭 COMP0

### 22.6.2 COMP1 控制寄存器 COMP1\_CFG (偏移: 11Ch)

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27	LOCK	RW	0x0	COMP1 输出锁存位: 1: COMP1 保持当前输出值 0: COMP1 正常输出
26:17	RSV	-	-	保留
16:15	EDSET	RW	0x0	COMP1 触发模式设置: 00: 电平模式 01: COMP1 检测结果上升沿锁存状态 10: COMP1 检测结果下降沿锁存状态 11: COMP1 检测结果上升沿或下降沿锁存状态
14:12	RSV	-	-	保留
11	COMNSEL	RW	0x0	COMP1 的 P 端 P0、P1、P2 共模电压设置位: 1: 开启共模电压功能 0: 禁止共模电压功能
10:9	HYS	RW	0x0	COMP1 迟滞电压设置位: 00: 0mV 01: 10mV 10: 20mV 11: 30mV
8	POL	RW	0x0	COMP1 极性选择: 1: COMP1 N 端电压高时输出 1

比特	名称	属性	复位值	描述
				0: COMP1 P 端电压高时输出 1
7:5	INMSEL	RW	0x0	COMP1 负向输入端通道选择: 000: COMP1_N0 输入, PC0 输入 001: COMP1_N1 输入, PB5 输入 010: COMP1_N2 输入, PB3 输入 011: DAC 输入 100: 共模信号输入
4:2	INPSEL	RW	0x0	COMP1 正向输入端通道选择: 000: COMP1_P0 输入, PB2 输入 001: COMP1_P1 输入, PB1 输入 010: COMP1_P2 输入, PB0 输入 011: COMP1_P3 输入, PB4 输入 100: COMP1_P4 输入, PB7 输入 101: COMP1_P5 输入, PGA0_VOUT 输入 110: COMP1_P6 输入, PGA1_VOUT 输入 111: COMP1_P7 输入, PGA2_VOUT 输入
1	LMP	RW	0x0	COMP1 低功耗模式控制信号: 1: 高速 0: 低速
0	EN	RW	0x0	COMP1 使能: 1: 使能 COMP1 0: 关闭 COMP1

### 22.6.3 COMP0 轮询寄存器 COMP0\_POLL (偏移: 128h)

比特	名称	属性	复位值	描述
31:9	PERIOD	RW	0x0	轮询等待周期 (Polling Wait Cycle)。每个通道比较器时间为: $4 * t_{pclk} * COMP0\_PERIOD$
8:6	POUT	R	0x0	轮询通道输出, 反映轮询通道输出状态: BIT6 对应通道 COMP0_P0, BIT7 对应通道 COMP0_P1, BIT8 对应通道 COMP0_P2。 1: 比较器输出结果为 1 0: 比较器输出结果为 0
5:4	RSV	-	-	保留

比特	名称	属性	复位值	描述
3	POLMASKEN	RW	0x0	COMP0 轮询模式 MASK 功能使能: 1: 使能 COMP0 轮询模式 MASK 功能 0: 关闭 COMP0 轮询模式 MASK 功能
2	FIXN	RW	0x0	COMP0 轮询通道反相输入端固定设置寄存器: 1: 轮询通道反相输入固定 0: 轮询通道反相输入不固定
1	POLCH	RW	0x0	COMP0 轮询通道设置: 1: 轮询通道 COMP0_P0/ COMP0_P1/ COMP0_P2 0: 轮询通道 COMP0_P0/ COMP0_P1
0	POLEN	RW	0x0	COMP0 轮询模式使能: 1: 使能 COMP0 轮询模式 0: 关闭 COMP0 轮询模式

#### 22.6.4 COMP1 轮询寄存器 COMP1\_POLL (偏移: 12Ch)

比特	名称	属性	复位值	描述
31:9	PERIOD	RW	0x0	轮询等待周期 (Polling Wait Cycle)。每个通道比较器时间为: $4 * t_{pclk} * COMP1\_PERIOD$
8:6	POUT	R	0x0	轮询通道输出, 反映轮询通道输出状态: BIT6 对应通道 COMP1_P0, BIT7 对应通道 COMP1_P1, BIT8 对应通道 COMP1_P2。 1: 比较器输出结果为 1 0: 比较器输出结果为 0
5:4	RSV	-	-	保留
3	POLMASKEN	RW	0x0	COMP1 轮询模式 MASK 功能使能: 1: 使能 COMP1 轮询模式 MASK 功能 0: 关闭 COMP1 轮询模式 MASK 功能
2	FIXN	RW	0x0	COMP1 轮询通道反相输入端固定设置寄存器: 1: 轮询通道反相输入固定 0: 轮询通道反相输入不固定
1	POLCH	RW	0x0	COMP1 轮询通道设置: 1: 轮询通道 COMP1_P0/ COMP1_P1/ COMP1_P2 0: 轮询通道 COMP1_P0/ COMP1_P1

比特	名称	属性	复位值	描述
0	POLEN	RW	0x0	COMP1 轮询模式使能: 1: 使能 COMP1 轮询模式 0: 关闭 COMP1 轮询模式

### 22.6.5 COMP0 轮询 MASK 寄存器 COMP0\_POLLMASK (偏移: 138h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:0	POLLMASK	RW	0x0	COMP0 轮询模式通道切换 MASK 时间设置寄存器 轮询通道切换 MASK 时间为: $t_{pclk} * COMP0\_POLLMASK$

### 22.6.6 COMP1 轮询 MASK 寄存器 COMP1\_POLLMASK (偏移: 13Ch)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:0	POLLMASK	RW	0x0	COMP1 轮询模式通道切换 MASK 时间设置寄存器 轮询通道切换 MASK 时间为: $t_{pclk} * COMP1\_POLLMASK$

### 22.6.7 COMP0 滤波设置寄存器 COMP0\_LVSET (偏移: 148h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:4	LVTSET	RW	0x0	COMP0 滤波时间设置寄存器 COMP0 滤波时间为: $t_{pclk} * COMP0\_LVTSET$
3	LV_CNT_RST	W	0x0	COMP0 软件清除滤波时间计数器 写 1 清 0
2:1	LVMSEL	RW	0x0	COMP0 滤波模式选择: 00: 滤波模式 0 01: 滤波模式 1 10: 滤波模式 2 11: 滤波模式 3

比特	名称	属性	复位值	描述
0	LVEN	RW	0x0	COMP0 滤波模式使能: 1: COMP0 滤波功能使能 0: COMP0 滤波功能禁止

### 22.6.8 COMP1 滤波设置寄存器 COMP1\_LVSET (偏移: 14Ch)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:4	LVTSET	RW	0x0	COMP1 滤波时间设置寄存器 COMP1 滤波时间为: $t_{\text{pclk}} * \text{COMP1\_LVTSET}$
3	LV_CNT_RST	W	0x0	COMP1 软件清除滤波时间计数器 写 1 清 0
2:1	LVMSEL	RW	0x0	COMP1 滤波模式选择: 00: 滤波模式 0 01: 滤波模式 1 10: 滤波模式 2 11: 滤波模式 3
0	LVEN	RW	0x0	COMP1 滤波模式使能: 1: COMP1 滤波功能使能 0: COMP1 滤波功能禁止

### 22.6.9 COMP0 WINDOW 设置寄存器 COMP0\_WINCFG (偏移: 158h)

比特	名称	属性	复位值	描述
31:24	WINSET	RW	0x0	Window 时间设置位: Window 时间为: $2 * t_{\text{pclk}} * (\text{WINSET}+1)$
23:22	WINEDS	RW	0x0	Window PWM 边沿设置位: 00: PWM 不参与 Window 功能 01: PWM 下降沿参与 Window 功能 10: PWM 上升沿参与 Window 功能 11: PWM 双边沿参与 Window 功能
21	LPTC2S	RW	0x0	LPTIM_CH2 PWM Window 功能设置位: 1: LPTIMER_CH2 参与 Window 功能 0: LPTIMER_CH2 不参与 Window 功能

比特	名称	属性	复位值	描述
20	LPTC1S	RW	0x0	LPTIMER_CH1 PWM Window 功能设置位: 1: LPTIMER_CH1 参与 Window 功能 0: LPTIMER_CH1 不参与 Window 功能
19	ATOC4S	RW	0x0	ATIMER OC4REF PWM Window 功能设置位: 1: ATIMER OC4REF 参与 Window 功能 0: ATIMER OC4REF 不参与 Window 功能
18	ATOC3NS	RW	0x0	ATIMER_CH3N PWM Window 功能设置位: 1: ATIMER_CH3N 参与 Window 功能 0: ATIMER_CH3N 不参与 Window 功能
17	ATOC3S	RW	0x0	ATIMER OC3REF PWM Window 功能设置位: 1: ATIMER OC3REF 参与 Window 功能 0: ATIMER OC3REF 不参与 Window 功能
16	ATOC2NS	RW	0x0	ATIMER_CH2N PWM Window 功能设置位: 1: ATIMER_CH2N 参与 Window 功能 0: ATIMER_CH2N 不参与 Window 功能
15	ATOC2S	RW	0x0	ATIMER OC2REF PWM Window 功能设置位: 1: ATIMER OC2REF 参与 Window 功能 0: ATIMER OC2REF 不参与 Window 功能
14	ATOC1NS	RW	0x0	ATIMER_CH1N PWM Window 功能设置位: 1: ATIMER_CH1N 参与 Window 功能 0: ATIMER_CH1N 不参与 Window 功能
13	ATOC1S	RW	0x0	ATIMER OC1REF PWM Window 功能设置位: 1: ATIMER OC1REF 参与 Window 功能 0: ATIMER OC1REF 不参与 Window 功能
12	GT2OC4S	RW	0x0	GTIMER2 OC4REF PWM Window 功能设置位: 1: GTIMER2 OC4REF 参与 Window 功能 0: GTIMER2 OC4REF 不参与 Window 功能
11	GT2OC3S	RW	0x0	GTIMER2 OC3REF PWM Window 功能设置位: 1: GTIMER2 OC3REF 参与 Window 功能 0: GTIMER2 OC3REF 不参与 Window 功能
10	GT2OC2S	RW	0x0	GTIMER2 OC2REF PWM Window 功能设置位: 1: GTIMER2 OC2REF 参与 Window 功能 0: GTIMER2 OC2REF 不参与 Window 功能
9	GT2OC1S	RW	0x0	GTIMER2 OC1REF PWM Window 功能设置位: 1: GTIMER2 OC1REF 参与 Window 功能 0: GTIMER2 OC1REF 不参与 Window 功能

比特	名称	属性	复位值	描述
8	GT1OC4S	RW	0x0	GTIMER1 OC4REF Window 功能设置位: 1: GTIMER1 OC4REF 参与 Window 功能 0: GTIMER1 OC4REF 不参与 Window 功能
7	GT1OC3S	RW	0x0	GTIMER1 OC3REF Window 功能设置位: 1: GTIMER1 OC3REF 参与 Window 功能 0: GTIMER1 OC3REF 不参与 Window 功能
6	GT1OC2S	RW	0x0	GTIMER1 OC2REF Window 功能设置位: 1: GTIMER1 OC2REF 参与 Window 功能 0: GTIMER1 OC2REF 不参与 Window 功能
5	GT1OC1S	RW	0x0	GTIMER1 OC1REF Window 功能设置位: 1: GTIMER1 OC1REF 参与 Window 功能 0: GTIMER1 OC1REF 不参与 Window 功能
4	GT0OC4S	RW	0x0	GTIMER0 OC4REF Window 功能设置位: 1: GTIMER0 OC4REF 参与 Window 功能 0: GTIMER0 OC4REF 不参与 Window 功能
3	GT0OC3S	RW	0x0	GTIMER0 OC3REF Window 功能设置位: 1: GTIMER0 OC3REF 参与 Window 功能 0: GTIMER0 OC3REF 不参与 Window 功能
2	GT0OC2S	RW	0x0	GTIMER0 OC2REF Window 功能设置位: 1: GTIMER0 OC2REF 参与 Window 功能 0: GTIMER0 OC2REF 不参与 Window 功能
1	GT0OC1S	RW	0x0	GTIMER0 OC1REF Window 功能设置位: 1: GTIMER0 OC1REF 参与 Window 功能 0: GTIMER0 OC1REF 不参与 Window 功能
0	WINEN	RW	0x0	比较器 0 Window 功能使能位: 1: 使能 Window 功能 0: 禁止 Window 功能

## 22.6.10 COMP1 WINDOW 设置寄存器 COMP1\_WINCFG (偏移: 15Ch)

比特	名称	属性	复位值	描述
31:24	WINSET	RW	0x0	Window 时间设置位: Window 时间为: $2 * t_{\text{pclk}} * (\text{WINSET}+1)$

比特	名称	属性	复位值	描述
23:22	WINEDS	RW	0x0	Window PWM 边沿设置位: 00: PWM 不参与 Window 功能 01: PWM 下降沿参与 Window 功能 10: PWM 上升沿参与 Window 功能 11: PWM 双边沿参与 Window 功能
21	LPTC2S	RW	0x0	LPTIMER_CH2 PWM Window 功能设置位: 1: LPTIMER_CH2 参与 Window 功能 0: LPTIMER_CH2 不参与 Window 功能
20	LPTC1S	RW	0x0	LPTIMER_CH1 PWM Window 功能设置位: 1: LPTIMER_CH1 参与 Window 功能 0: LPTIMER_CH1 不参与 Window 功能
19	ATOC4S	RW	0x0	ATIMER OC4REF PWM Window 功能设置位: 1: ATIMER OC4REF 参与 Window 功能 0: ATIMER OC4REF 不参与 Window 功能
18	ATC3NS	RW	0x0	ATIMER_CH3N PWM Window 功能设置位: 1: ATIMER_CH3N 参与 Window 功能 0: ATIMER_CH3N 不参与 Window 功能
17	ATOC3S	RW	0x0	ATIMER OC3REF PWM Window 功能设置位: 1: ATIMER OC3REF 参与 Window 功能 0: ATIMER OC3REF 不参与 Window 功能
16	ATC2NS	RW	0x0	ATIMER_CH2N PWM Window 功能设置位: 1: ATIMER_CH2N 参与 Window 功能 0: ATIMER_CH2N 不参与 Window 功能
15	ATOC2S	RW	0x0	ATIMER OC2REF PWM Window 功能设置位: 1: ATIMER OC2REF 参与 Window 功能 0: ATIMER OC2REF 不参与 Window 功能
14	ATOC1NS	RW	0x0	ATIMER_CH1N PWM Window 功能设置位: 1: ATIMER_CH1N 参与 Window 功能 0: ATIMER_CH1N 不参与 Window 功能
13	ATOC1S	RW	0x0	ATIMER OC1REF PWM Window 功能设置位: 1: ATIMER OC1REF 参与 Window 功能 0: ATIMER OC1REF 不参与 Window 功能
12	GT2OC4S	RW	0x0	GTIMER2 OC4REF PWM Window 功能设置位: 1: GTIMER2 OC4REF 参与 Window 功能 0: GTIMER2 OC4REF 不参与 Window 功能

比特	名称	属性	复位值	描述
11	GT2OC3S	RW	0x0	GTIMER2 OC3REF PWM Window 功能设置位: 1: GTIMER2 OC3REF 参与 Window 功能 0: GTIMER2 OC3REF 不参与 Window 功能
10	GT2OC2S	RW	0x0	GTIMER2 OC2REF PWM Window 功能设置位: 1: GTIMER2 OC2REF 参与 Window 功能 0: GTIMER2 OC2REF 不参与 Window 功能
9	GT2OC1S	RW	0x0	GTIMER2 OC1REF PWM Window 功能设置位: 1: GTIMER2 OC1REF 参与 Window 功能 0: GTIMER2 OC1REF 不参与 Window 功能
8	GT1OC4S	RW	0x0	GTIMER1 OC4REF Window 功能设置位: 1: GTIMER1 OC4REF 参与 Window 功能 0: GTIMER1 OC4REF 不参与 Window 功能
7	GT1OC3S	RW	0x0	GTIMER1 OC3REF Window 功能设置位: 1: GTIMER1 OC3REF 参与 Window 功能 0: GTIMER1 OC3REF 不参与 Window 功能
6	GT1OC2S	RW	0x0	GTIMER1 OC2REF Window 功能设置位: 1: GTIMER1 OC2REF 参与 Window 功能 0: GTIMER1 OC2REF 不参与 Window 功能
5	GT1OC1S	RW	0x0	GTIMER1 OC1REF Window 功能设置位: 1: GTIMER1 OC1REF 参与 Window 功能 0: GTIMER1 OC1REF 不参与 Window 功能
4	GT0OC4S	RW	0x0	GTIMER0 OC4REF Window 功能设置位: 1: GTIMER0 OC4REF 参与 Window 功能 0: GTIMER0 OC4REF 不参与 Window 功能
3	GT0OC3S	RW	0x0	GTIMER0 OC3REF Window 功能设置位: 1: GTIMER0 OC3REF 参与 Window 功能 0: GTIMER0 OC3REF 不参与 Window 功能
2	GT0OC2S	RW	0x0	GTIMER0 OC2REF Window 功能设置位: 1: GTIMER0 OC2REF 参与 Window 功能 0: GTIMER0 OC2REF 不参与 Window 功能
1	GT0OC1S	RW	0x0	GTIMER0 OC1REF Window 功能设置位: 1: GTIMER0 OC1REF 参与 Window 功能 0: GTIMER0 OC1REF 不参与 Window 功能
0	WINEN	RW	0x0	比较器 0 Window 功能使能位: 1: 使能 Window 功能 0: 禁止 Window 功能

### 22.6.11 COMP 状态寄存器 COMP0\_INTSTAT (偏移: 168h)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	ARST	R	0x0	COMP0 模拟部分实时状态寄存器: 1: 当前 COMP0 模拟部分比较结果为 1 0: 当前 COMP0 模拟部分比较结果为 0
8	RST	R	0x0	COMP0 实时状态寄存器: 1: 当前 COMP0 比较结果为 1 0: 当前 COMP0 比较结果为 0
7	RSV	-	-	保留
6	PP2INST	W1C	0x0	轮询模式下, COMP0_P2 通道中断状态寄存器: 1: 产生比较结果为 1 0: 未产生比较结果为 1 的状态 写 1 清 0
5	PP1INST	W1C	0x0	轮询模式下, COMP0_P1 通道中断状态寄存器: 1: 产生比较结果为 1 0: 未产生比较结果为 1 的状态 写 1 清 0
4	PP0INST	W1C	0x0	轮询模式下, COMP0_P0 通道中断状态寄存器: 1: 产生比较结果为 1 0: 未产生比较结果为 1 的状态 写 1 清 0
3:1	RSV	-	-	保留
0	INTST	W1C	0x0	COMP0 中断状态寄存器: 1: COMP0 产生比较结果为 1 0: COMP0 未产生比较结果为 1 的状态 写 1 清 0

### 22.6.12 COMP 状态寄存器 COMP1\_INTSTAT (偏移: 16Ch)

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	ARST	R	0x0	COMP1 模拟部分实时状态寄存器: 1: 当前 COMP1 模拟部分比较结果为 1 0: 当前 COMP1 模拟部分比较结果为 0

比特	名称	属性	复位值	描述
8	RST	R	0x0	COMP1 实时状态寄存器: 1: 当前 COMP1 比较结果为 1 0: 当前 COMP1 比较结果为 0
7	RSV	-	-	保留
6	PP2INST	W1C	0x0	轮询模式下, COMP1_P2 通道中断状态寄存器: 1: 产生比较结果为 1 0: 未产生比较结果为 1 的状态 写 1 清 0
5	PP1INST	W1C	0x0	轮询模式下, COMP1_P1 通道中断状态寄存器: 1: 产生比较结果为 1 0: 未产生比较结果为 1 的状态 写 1 清 0
4	PP0INST	W1C	0x0	轮询模式下, COMP1_P0 通道中断状态寄存器: 1: 产生比较结果为 1 0: 未产生比较结果为 1 的状态 写 1 清 0
3:1	RSV	-	-	保留
0	INTST	W1C	0x0	COMP1 中断状态寄存器: 1: COMP1 产生比较结果为 1 0: COMP1 未产生比较结果为 1 的状态 写 1 清 0

### 22.6.13 COMP0 中断使能寄存器 COMP0\_INTEN (偏移: 178h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	PP2INTEN	RW	0x0	COMP0_PP2INST 中断使能寄存器: 1: COMP0_PP2INST 中断使能 0: COMP0_PP2INST 中断禁止
5	PP1INTEN	RW	0x0	COMP0_PP1INST 中断使能寄存器: 1: COMP0_PP1INST 中断使能 0: COMP0_PP1INST 中断禁止
4	PP0INTEN	RW	0x0	COMP0_PP0INST 中断使能寄存器: 1: COMP0_PP0INST 中断使能 0: COMP0_PP0INST 中断禁止
3:1	RSV	-	-	保留

比特	名称	属性	复位值	描述
0	INTEN	RW	0x0	COMP0_INTST 中断使能寄存器: 1: COMP0_INTST 中断使能 0: COMP0_INTST 中断禁止

### 22.6.14 COMP1 中断使能寄存器 COMP1\_INTEN (偏移: 17Ch)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	PP2INTEN	RW	0x0	COMP1_PP2INST 中断使能寄存器: 1: COMP1_PP2INST 中断使能 0: COMP1_PP2INST 中断禁止
5	PP1INTEN	RW	0x0	COMP1_PP1INST 中断使能寄存器: 1: COMP1_PP1INST 中断使能 0: COMP1_PP1INST 中断禁止
4	PP0INTEN	RW	0x0	COMP1_PP0INST 中断使能寄存器: 1: COMP1_PP0INST 中断使能 0: COMP1_PP0INST 中断禁止
3:1	RSV	-	-	保留
0	INTEN	RW	0x0	COMP1_INTST 中断使能寄存器: 1: COMP1_INTST 中断使能 0: COMP1_INTST 中断禁止

## 22.7 使用流程

COMP 比较器基本配置:

1. 在 COMPx\_CFG 寄存器选择正向负向输入端通道和触发模式。
2. 在 COMPx\_POLL 寄存器可选择是否打开轮询模式以及配置轮询等待周期。
3. 在 COMPx\_POLLMASK 寄存器设置轮询通道切换时间。
4. 在 COMPx\_LVSET 寄存器可选择滤波功能包括滤波模式和滤波时间等。
5. 在 COMPx\_WINCFG 寄存器选择窗口触发位、窗口时间以及窗口使能。
6. 在 COMPx\_INTEN 寄存器中打开中断使能。
7. 打开 COMPx\_EN 使能, 观察中断比较结果。

## 23 DIV

### 23.1 概述

DIV 的实现目标是能够支持不超过 32bits 的除法。除数不大于 32bits，被除数可以为任意 bit。主要应用于除数小于 32bits 的应用。

### 23.2 主要特性

- 有符号整数运算
- 支持不超过 32bits 的除法
- 除以 0 警告功能
- 8 个 AHB 时钟周期出运算结果

### 23.3 寄存器描述

DIV 模块基地址为: 0x4002\_1000

表 23-1: DIV 寄存器列表

偏置	名称	描述
0x00	DIV_DIVIDEND	被除数寄存器
0x04	DIV_DIVISOR	除数寄存器
0x08	DIV_REMAIN	余数寄存器
0x0C	DIV_QUOTIENT	商寄存器
0x10	DIV_STATUS	状态寄存器

#### 23.3.1 被除数寄存器 DIV\_DIVIDEND (偏移: 00h)

比特	名称	属性	复位值	描述
31:0	DIVIDEND	RW	0x0	被除数

### 23.3.2 除数寄存器 DIV\_DIVISOR (偏移: 04h)

比特	名称	属性	复位值	描述
31:0	DIVISOR	RW	0x1	除数

### 23.3.3 余数寄存器 DIV\_REMAIN (偏移: 08h)

比特	名称	属性	复位值	描述
31:0	REMAIN	R	0x0	余数

### 23.3.4 商寄存器 DIV\_QUOTIENT (偏移: 0Ch)

比特	名称	属性	复位值	描述
31:0	QUOTIENT	R	0x0	商

### 23.3.5 状态寄存器 DIV\_STATUS (偏移: 10h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留位
1	DB0	R	0x0	除数为 0 标志: 1: 除数为 0 0: 除数不为 0
0	DONE	R	0x1	运算完成标志位: 1: 运算完成 0: 运算未完成

## 23.4 使用流程

1. 往除数寄存器中写入除数。
2. 往被除数寄存器中写入被除数。
3. 隔 8 个算法时钟周期或者查询完成状态位从商寄存器中读出商，如果被除数还未输入完，则转至步骤 2，如果输入完则转至步骤 4。
4. 从余数寄存器中读出余数，运算结束。

## 24 SQRT

### 24.1 概述

SQRT模块为开方运算模块。

### 24.2 主要特性

- AHB 总线接口，4 个 AHB 总线时钟出结果

### 24.3 寄存器描述

SQRT 寄存器基地址：0x4002\_1400

表 24-1: SQRT 寄存器列表

偏置	名称	描述
0x00	SQRT_VALUE	SQRT 被开方数据寄存器
0x04	SQRT_RESULT	SQRT 开方运算结果寄存器
0x08	SQRT_SR	SQRT 状态寄存器

#### 24.3.1 SQRT 被开方数据寄存器 (SQRT\_VALUE) (偏移: 00h)

比特	名称	属性	复位值	描述
31:0	VALUE	RW	0x0	被开放数据寄存器： 写此寄存器，写入的数据即刻开始开方运算。4 个 AHB 时钟周期出结果。

#### 24.3.2 SQRT 开方运算结果寄存器 (SQRT\_RESULT) (偏移: 04h)

比特	名称	属性	复位值	描述
31:0	RESULT	R	0x0	此寄存器在开方运算完毕后，保存运算结果。

### 24.3.3 SQRT 状态寄存器 (SQRT\_SR) (偏移: 08h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	BUSY	R	0x0	开方运算状态标志: 1: 开方运算正在进行, 未完成 0: 开方运算完成, 硬件空闲

## 24.4 使用流程

1. 向 SQRT\_VALUE 写入被开方数值。
2. 软件 delay 或者查询 SQRT\_SR 寄存器, 等待值为 0。
3. 读取 SQRT\_RESULT, 得到开方结果。

## 25 SysTick

### 25.1 概述

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0+处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常(异常编号为 15)，该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询溢出标志。

### 25.2 寄存器描述

SysTick 寄存器基地址：0xE000\_E010

表 25-1: SysTick 寄存器列表

偏置	名称	描述
0x00	SysTick_CTRL	SysTick 控制和状态寄存器
0x04	SysTick_LOAD	SysTick 重载值寄存器
0x08	SysTick_VAL	SysTick 当前值寄存器

### 25.2.1 控制和状态寄存器 SysTick\_CTRL (偏移: 00h)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	COUNTFLAG	R	0x0	Systick定时器溢出标志: 1: Systick定时器发生下溢出 0: Systick定时器未发生溢出 读该寄存器, 可清除COUNTFLAG标志。
15:2	RSV	-	-	保留
1	TICKINT	RW	0x0	SysTick中断使能: 1: 使能中断 0: 禁止中断
0	ENABLE	RW	0x0	SysTick定时器使能: 1: 使能SysTick 0: 禁止SysTick

### 25.2.2 重载值寄存器 SysTick\_LOAD (偏移: 04h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:0	RELOAD	RW	0xFFFFFFFF	SysTick定时器重载值

### 25.2.3 当前值寄存器 SysTick\_VAL (偏移: 08h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:0	CURRENT	RW	0xFFFFFFFF	读该寄存器, 获取 SysTick 定时器的当前计数值; 写任意值到该寄存器, 清零该寄存器及 COUNTFLAG。

## 25.3 使用流程

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

1. 配置 SysTick\_CTRL[0]为 0，禁止 SysTick。
2. 配置 SysTick\_LOAD，选择 SysTick 的溢出周期。
3. 向 SysTick\_VAL 写入任意值，清零 SysTick\_VAL 及 SysTick\_CTRL。
4. COUNTFLAG。
5. 配置 SysTick\_CTRL[1]为 1，使能 SysTick 中断。
6. 配置 SysTick\_CTRL[0]为 1，使能 SysTick。
7. 查询等待定时器溢出标志到来之后关闭和清空计数器，或者在中断服务程序中读取 SysTick\_CTRL 以清除溢出标志。

## 26 调试支持 (DBG)

集成硬件调试模块的 Cortex®-M0 内核，支持指令断点（指令取值时停止）和数据断点（数据访问时停止）。当内核停止时，用户可以查看内核的内部状态和系统的外部状态。用户查询操作完成后，可以恢复内核和外设，继续执行相应的程序。芯片内核的硬件调试模块在连接到调试器时即可使用（在未被禁用的情况下）。

支持串行接口（两线 SWD）调试接口。

## 27 版本维护

日期	版本	描述
2024.11.18	V1.0	初始版