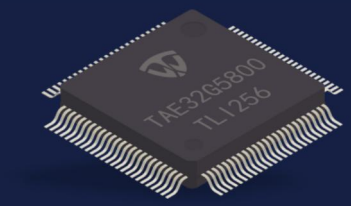
珠海泰为电子有限公司
Zhuhai Tai-Action Electronics CO., LTD.

TAE32G5800 系列



芯片手册

Reference manual

版本号：Rev 0.5

修订日期：2023 年 11 月 28 日

目录

1 文档说明

1.1 缩写词

缩写词	说明
读写（R/W）	软件可以读写该位
只读（R）	软件只能读取该位
只写（W）	软件只能写入该位，读取该位时将返回复位值
读清零（RC/W）	软件可以读写该位，读取该位时，将自动清零
写清零（R/WC）	软件可以读写该位，写入该位时，将自动清零
写 1 清零（R/W1C）	软件可以读写该位，写入 1 时，将自动清零
自动清零（R/WAC）	软件可以读写该位，动作完成，将自动清零

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：

- 在本文档中，将 Cortex™-M4 内核称为 M4
- CPU 内核集成了 SWD 调试端口：
  - SWD 调试端口（SWD-DP）提供一个基于串行线调试（SWD）协议的两引脚（时钟和数据）接口。
  - 有关 SWD 协议的信息，请参见《Cortex™-M4 技术参考手册》。
- FLASH：Embedded FLASH 的简称。
- 选项字节：存储于 FLASH 中的产品配置位。
- 字节：8 位数据。
- 半字：16 位数据。
- 字：32 位数据。
- 双字：64 位数据。
- IAP（In-Application Programming）：IAP 指在用户程序运行期间对芯片内的 FLASH 进行重复编程。
- ICP（In-Circuit Programming）：ICP 指芯片安装在用户应用电路板上时使用 SWD 协议或自举程序对芯片内的 FLASH 进行编程。
- I-Code：此总线用于将 CPU 的 I-BUS 连接到 FLASH 指令接口，通过此总线可执行预取操作。
- D-Code：此总线用于将 CPU 的 D-BUS（数据加载和调试访问）连接到 FLASH 数据接口。
- AHB：高级高性能总线

1.3 相关文档

- Cortex™-M4 技术参考手册，可按下述链接下载：
  http://infocenter.arm.com/help/topic/com.arm.doc.100166_0001_00_en/arm_cortexm4_processor_trm_100165_0201_00_en.pdf
- Cortex™-M4 Device 通用用户指南（含架构、指令集、核内外设等），可按下述链接下载：
  http://infocenter.arm.com/help/topic/com.arm.doc.dui0553b/DUI0553B_cortex_m4_dgug.pdf

2 特性列表

2.1 特性描述

- 采用 ARM Cortex-M4 系列内核
  - 系统工作频率 180MHz
  - 内核内置 32 位硬件乘/除法
- 硬件加速器(ERPU)
  - 内置 6 套 IIR 单元,支持 1/2/3/4 阶灵活可配
  - 内置 CORDIC 硬件加速算法，包括正余弦/反正切/求模值(求开方)等
- 存储资源
  - 内置 256KB 容量 FLASH 程序存储器
  - 支持读/写/擦保护及零延迟执行
  - 支持单/双 Bank 操作
  - 内置 128KB 系统 SRAM
- 时钟、复位和电源管理
  - 支持单电源输入，输入范围为 3.0V-3.6V
  - 支持上下电复位及欠压复位
  - 内置独立看门狗(IWDG)和窗口看门狗(WWDG)
  - 内置高频 RC8M 振荡器及低频 RC32K 振荡器
  - 支持带晶振和无晶振方案,支持 6M-26MHz 晶体振荡器
  - 内置时钟安全系统，晶振异常检测电路并自动切换
  - 内置温度传感器
- DMA 控制器
  - 内置 6 个独立的 DMA 通道
  - 支持内存与外设之间任意组合传输
- 通用 I/O(GPIO)
  - LQFP100PIN 封装有 85 个 I/O,LQFP80PIN 封装有 65 个 I/O，LQFP64PIN 封装有 51 个 I/O，LQFP48PIN 封装有 37 个 I/O，QFN48PIN 封装有 41 个 I/O
  - 支持 I/O 功能复用映射，支持位操作
  - 所有 I/O 内建上拉/下拉电阻
  - 支持外部中断输入，支持上下边沿触发，可用于产生中断、事件唤醒
  - 内置 2 档驱动可调，最大 24mA 驱动能力
- 通用定时器(TIMER)
  - 内置 9 套 16 位/32 位定时器
  - 支持定时、PWM 输出及 Capture 捕获功能
  - 支持自动重载功能
- 高精度脉宽调制器(HRPWM)
  - 支持 8 对 PWM(16 路)输出，每路 PWM 输出最小分辨率为 174ps
  - 支持互补/独立输出模式，支持对称/不对称波形输出
  - 内置死区及斩波插入机制机制，支持 10 路故障和事件输入，内置故障及异常保护功能
  - 支持多路 HRPWM 间的同步机制
- 模数转换器(ADC)
  - 内置 4 个完全独立高速模数转换器(ADC)
  - 支持 5M SPS 采样速率，ADC 分辨率达到 13 位，有效位数 11 位
  - 支持 42 路模拟采样通道(每个 ADC 20 路，其中采样引脚 42 路)
  - 支持规则序列转换和注入序列转换，支持增益补偿与偏置补偿机制
  - 支持单次/不连续/连续采样模式，支持事件触发采样，支持 HRPWM 同步触发采样
  - 内置 ADC 基准源，支持内部基准电压
- 数模转换器(DAC)及比较器(CMP)
  - 内置 9 套 DAC 和 9 套 CMP
  - DAC 分辨率为 12 位，INL/DNL 小于 5 个 LSB
  - 支持锯齿波、三角波输出，支持方向与幅度可编程
  - CMP 转换延迟的最小值/典型值/最大值为 15ns/18ns/21ns
- 通讯接口外设
  - 内置 5 套 UART 接口，支持 RS485 通信
  - 内置 2 套 SPI 接口
  - 内置 3 套 I2C 接口，支持 SMBus
  - 内置 2 套 CAN 控制器，支持 CAN2.0B FD 协议
  - 内置 1 套 USB 接口，支持 USB2.0 协议，

支持全速模式
- 内置 4 套 PDM 接口
- 内置 1 套 XIF 并口

- LVD 欠压监测功能(4 档可调)
- 128 位芯片唯一标识码
- 两种低功耗模式: 睡眠模式和停机模式
- 支持 2 线 SWD 调试接口
- 工业级结温范围：-40℃ - 150℃，ESD: ≥±2KV
- 封装：LQFP100(14*14),LQFP80(12*12), LQFP64(10*10),LQFP48(7*7),QFN48(7*7)

2.2 资源汇总

TAE32G5800 系列提供 48PIN、64PIN、80PIN、100PIN 封装。

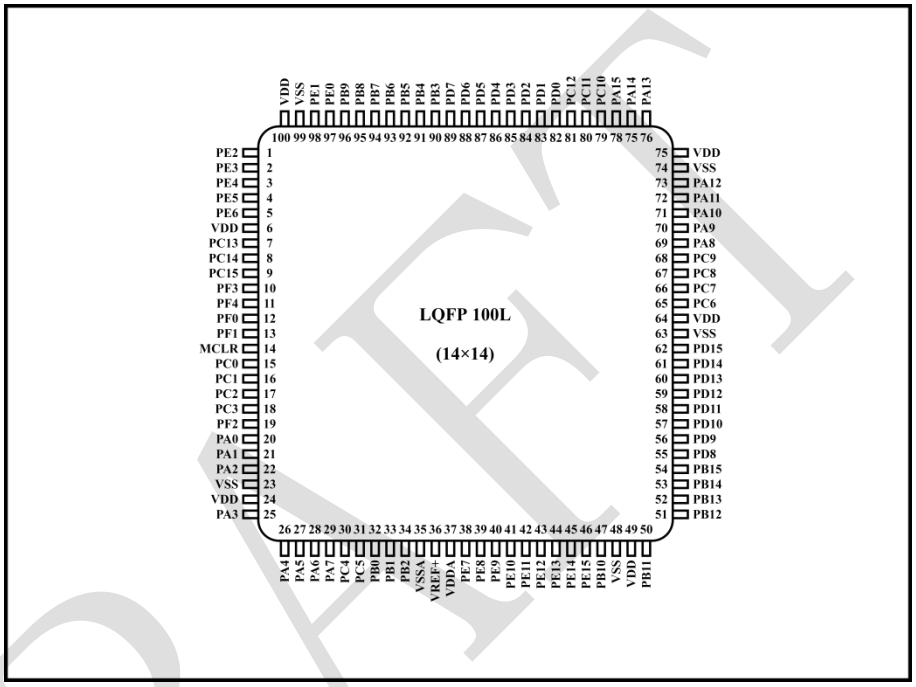
表 2-1 TAE32G5800 资源汇总

Peripheral		TAE32G5800 TQD256	TAE32G5800 TLD256	TAE32G5800 TLF256	TAE32G5800 TLH256	TAE32G5800 TLI256
Package		QFN 48L	LQFP 48L	LQFP 64L	LQFP 80L	LQFP 100L
Flash memory (Kbytes)		256	256	256	256	256
System SRAM (Kbytes)		128	128	128	128	128
Timer	Timer	9	9	9	9	9
	SysTick timer	1	1	1	1	1
	Watchdog(independent, window)	2	2	2	2	2
Comm. interfaces	HRPWM	5	5	7	8	8
	PDM	4	3	4	4	4
	UART	4	3	5	5	5
	I2C	3	3	3	3	3
	CAN	2	2	2	2	2
	USB	1	1	1	1	1
	QEI	3	3	3	3	3
	XIF	1	1	1	1	1
	SPI	2	2	2	2	2
GPIO	Normal I/Os	41	37	51	65	85
DMA channels		6	6	6	6	6
ADC channels		19	18	24	36	42
DAC channels		9	9	9	9	9
Ultra-fast analog comparator		9	9	9	9	9
CPU frequency		180Mhz				
Operating voltage		3.0 to 3.6 V				
Operating temperature		Ambient operating temperature: - 40 to 125 °C Junction temperature: -40℃ - 150℃				

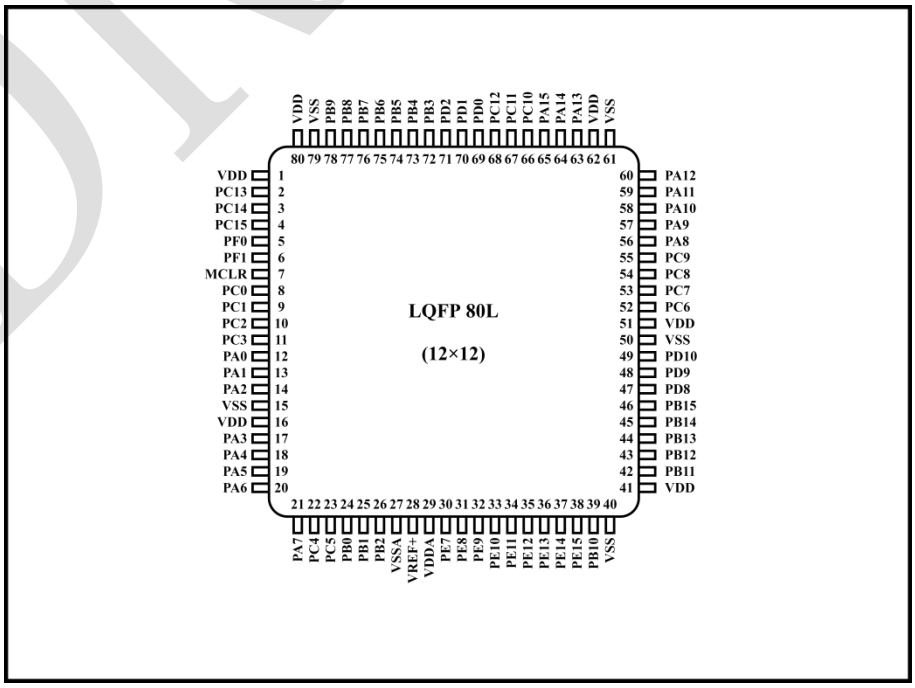
3 引脚定义

3.1 引脚框图

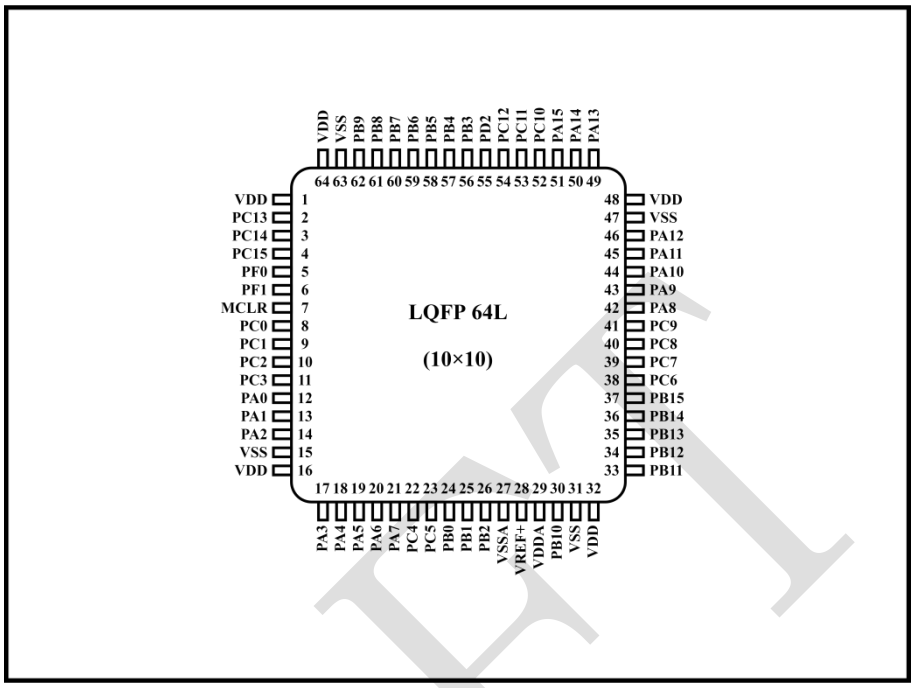
LQFP 100L:



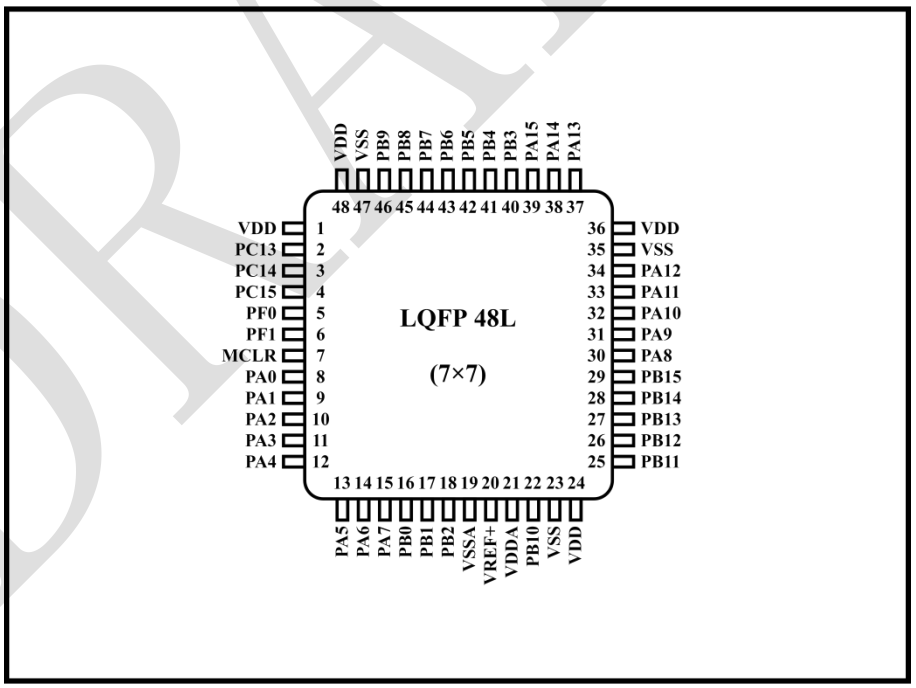
LQFP 80L:



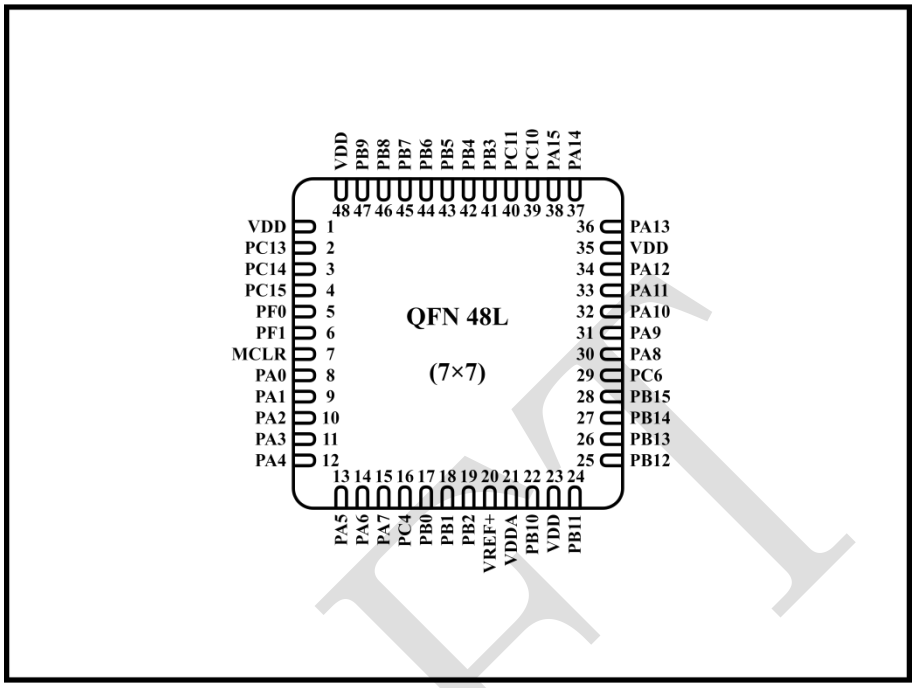
LQFP 64L:



LQFP 48L:



QFN 48L:



3.2 引脚描述

表 3-1 引脚定义缩写说明

名称	缩写	定义
引脚名称	除非在引脚名称下使用括号特别说明，否则在复位期间和复位之后的引脚功能均与实际引脚名称相同。	
引脚类型	S	电源引脚
	SO	电源输出引脚
	I	仅输入引脚
	I/O	输入/输出引脚
引脚架构	FT	5V I/O
	TT	3.3V I/O
	B	专用 BOOT 引脚
	MCLR	嵌入了弱上拉电阻的复位引脚
	TT or FT I/Os	
	_a	I/O，具有模拟复用功能的引脚
	_f	I/O，FM+引脚
	_fa	I/O，FM+引脚、具有模拟复用功能的引脚
	_u	I/O，USB 引脚
注释	除非通过注释特别说明，否则所有 I/O 在复位期间和复位之后均设置为悬空状态	
复用功能	通过 GPIOx_MUX 寄存器选择的功能	

表 3-2 引脚定义

Package					Pin name	Pin type	I/O structure	Digital Functions	Analog Functions
QFN 48	LQFP 48	LQFP 64	LQFP 80	LQFP 100					
-	-	-	-	1	PE2	I/O	FT	TRACECK，TMR4_CH0/ETR，QEI0_A，SPI0_CK，TMR10_CH0，UART0_RX，SPI1_CS，PDM0_CLK	-
-	-	-	-	2	PE3	I/O	FT	TRACED0，TMR4_CH1，QEI0_B，SPI0_CS，TMR10_CH1，UART0_TX，SPI1_CK，PDM0_DAT	-
-	-	-	-	3	PE4	I/O	FT	TRACED1，TMR4_CH2，QEI0_Z，SPI0_CS，TMR10_CH0N，UART0_DE，SPI1_MISO，PDM1_CLK	-
-	-	-	-	4	PE5	I/O	FT	TRACED2，TMR4_CH3，SPI0_MISO，TMR10_CH1N，SPI1_MOSI，PDM1_DAT	-
-	-	-	-	5	PE6	I/O	FT	TRACED3，SPI0_MOSI，TMR10_CH2N	-
1	1	1	1	6	VDD	S	-	-	-
2	2	2	2	7	PC13	I/O	FT	SWO，TMR0_CH0，TMR2_CH1，TMR10_CH3N，TMR9_CH0N，QEI1_A，TMR9_BK0，PDM2_CLK	-
3	3	3	3	8	PC14	I/O	FT	TMR0_CH1，TMR0_CH0N，TMR2_CH1N，TMR10_CH2，QEI1_B，PDM2_DAT	-
4	4	4	4	9	PC15	I/O	FT	TMR1_CH0，TMR0_CH1N，TMR10_CH3，QEI1_Z，PDM3_CLK	-
-	-	-	-	10	PF3	I/O	FT	TMR1_CH1，TMR1_CH0N，TMR3_CH3，SPI1_CK，TMR0_CH0，TMR10_BK0，PDM3_DAT	-
-	-	-	-	11	PF4	I/O	FT	TMR4_CH0/ETR，TMR1_CH1N，SPI1_CK，TMR0_CH1，TMR10_BK1	-
5	5	5	5	12	PF0	I/O	FT_fa	TMR4_CH1，I2C1_SDA，SPI1_CS，TMR9_CH2N	ADC0_IN10，CMP7_INM0，XOSCI
6	6	6	6	13	PF1	I/O	FT_a	I2C1_SCL，SPI1_CK	ADC1_IN10，CMP2_INM0，XOSCO
7	7	7	7	14	MCLR	I/O	TT	-	
-	-	8	8	15	PC0	I/O	FT_a	TMR3_CH0/ETR，QEI2_A，TMR9_CH0，UART0_RX，PDM3_DAT，PDM0_CLK	ADC0_IN6，ADC1_IN6，CMP2_INM1
-	-	9	9	16	PC1	I/O	TT_a	TMR3_CH1，QEI2_B，TMR9_CH1，UART0_TX，TMR10_CH1N，PDM3_CLK，PDM0_DAT	ADC0_IN7，ADC1_IN7，CMP2_INP1
-	-	10	10	17	PC2	I/O	FT_a	TMR3_CH2，QEI2_Z，TMR9_CH2，UART0_DE，TMR10_CH1，PDM1_CLK，CMP2_OUT	ADC0_IN8，ADC1_IN8

-	-	11	11	18	PC3	I/O	TT_a	TMR3_CH3，TMR2_CH0，TMR9_CH3，UART0_RE，TMR9_BK1，PDM1_DAT	ADC0_IN9，ADC1_IN9
-	-	-	-	19	PF2	I/O	FT	TMR2_CH0N，I2C1_SBA，TMR10_CH2，UART1_RE	-
8	8	12	12	20	PA0	I/O	TT_a	TMR3_CH0/ETR，TMR4_CH0/ETR，TMR10_BK0，TMR10_ETR，QEI0_A，UART1_DE，I2C0_SCL，PDM0_CLK，CMP0_OUT	ADC0_IN1，ADC1_IN1，CMP0_INM1，CMP2_INP0
9	9	13	13	21	PA1	I/O	TT_a	TMR3_CH1，TMR4_CH1，I2C1_SDA，UART1_DE，QEI0_B，UART2_TX，TMR0_CH0N，I2C0_SDA，PDM0_DAT，CMP8_OUT	ADC0_IN2，ADC1_IN2，CMP0_INP0
10	10	14	14	22	PA2	I/O	FT_a	TMR3_CH2，TMR4_CH2，I2C1_SCL，UART0_TX，SPI0_CS，UART1_TX，QEI0_Z，UART2_RX，TMR0_CH0，I2C0_SBA，CMP1_OUT	ADC0_IN3，CMP1_INM1
-	-	15	15	23	VSS	S	-	-	-
-	-	16	16	24	VDD	S	-	-	-
11	11	17	17	25	PA3	I/O	TT_a	TMR3_CH3，TMR4_CH3，TMR1_CH1，UART0_RX，SPI0_CK，UART1_RX，TMR0_CH1，PDM2_CLK	ADC0_IN4，CMP1_INP1
12	12	18	18	26	PA4	I/O	TT_a	TMR4_CH1，TMR1_CH1N，SPI0_CS，SPI0_MISO，SPI1_CS，TMR0_CH1N，PDM2_DAT	ADC1_IN16，DAC0_OUT，CMP0_INM0
13	13	19	19	27	PA5	I/O	TT_a	TMR3_CH0/ETR，TMR4_CH2，QEI2_A，SPI0_CK，SPI0_MOSI，PDM3_CLK，CMP3_OUT	ADC1_IN13，DAC1_OUT，CMP1_INM0
14	14	20	20	28	PA6	I/O	TT_a	TMR1_CH0，TMR4_CH0/ETR，QEI2_B，SPI0_MISO，TMR9_BK0，UART0_DE，TMR10_BK0，PDM3_DAT，CMP0_OUT	ADC1_IN3，DAC2_OUT，CMP7_INM1
15	15	21	21	29	PA7	I/O	TT_a	TMR2_CH0，TMR4_CH1，QEI2_Z，SPI0_MOSI，TMR9_CH0N，UART0_RE，TMR10_CH0N，PDM1_CLK，CMP1_OUT	ADC1_IN4，CMP1_INP0
16	-	22	22	30	PC4	I/O	FT_fa	TMR0_CH1，I2C1_SCL，UART0_TX，TMR9_ETR，PDM1_DAT，HRPWM_EVT1	ADC1_IN5，CMP7_INP0
-	-	23	23	31	PC5	I/O	TT_a	TMR0_CH0，，I2C1_SDA，TMR9_CH3N，UART0_RX，TMR0_BK0，PDM0_CLK，HRPWM_EVT9	ADC1_IN11，CMP8_INP0
17	16	24	24	32	PB0	I/O	TT_a	TMR4_CH2，TMR9_CH1N，TMR10_CH1N，PDM0_DAT，HRPWM_FLT6	ADC0_IN15，ADC2_IN12，CMP3_INP0
18	17	25	25	33	PB1	I/O	TT_a	TMR4_CH3，TMR9_CH2N，CMP3_OUT，TMR10_CH2N，UART0_DE，HRPWM_SCO	ADC0_IN12，ADC2_IN1，CMP0_INP1

19	18	26	26	34	PB2	I/O	TT_a	TMR4_CH0/ETR，I2C2_SBA，TMR10_CH0，HRPWM_SCI	ADC1_IN12，CMP3_INM1
-	19	27	27	35	VSSA	S	-	-	-
20	20	28	28	36	VREF+	S	-	-	-
21	21	29	29	37	VDDA	S	-	-	-
-	-	-	30	38	PE7	I/O	TT_a	TMR9_ETR，UART1_TX，XIF_D4，HRPWM_EVT2	ADC2_IN4，CMP3_INP1
-	-	-	31	39	PE8	I/O	FT_a	TMR3_CH0/ETR，TMR4_CH2，QEI0_A，TMR9_CH0N，UART1_RX，XIF_D5，HRPWM_FLT2	ADC2_IN6，ADC3_IN6，CMP3_INM0
-	-	-	32	40	PE9	I/O	FT_a	TMR3_CH1，TMR4_CH3，QEI0_B，TMR9_CH0，UART2_TX，XIF_D6	ADC2_IN2，CMP8_INM0
-	-	-	33	41	PE10	I/O	FT_a	TMR3_CH2，QEI0_Z，TMR9_CH1N，UART3_RX，XIF_D7	ADC2_IN14，ADC3_IN14
-	-	-	34	42	PE11	I/O	FT_a	TMR3_CH3，I2C1_SCL，SPI0_CS，TMR9_CH1，XIF_D8	ADC2_IN15，ADC3_IN15
-	-	-	35	43	PE12	I/O	FT_a	TMR4_CH0/ETR，I2C1_SDA，SPI0_CK，TMR9_CH2N，QEI1_A，XIF_D9	ADC2_IN16，ADC3_IN16
-	-	-	36	44	PE13	I/O	FT_a	TMR4_CH1，I2C1_SBA，SPI0_MISO，TMR9_CH2，QEI1_B，XIF_D10	ADC2_IN3
-	-	-	37	45	PE14	I/O	FT_a	TMR4_CH2，SPI0_MOSI，TMR9_CH3，QEI1_Z，TMR9_BK1，XIF_D11	ADC3_IN1
-	-	-	38	46	PE15	I/O	FT_a	TMR4_CH3，TMR9_CH3N，UART2_RX，TMR9_BK0，XIF_D12	ADC3_IN2
22	22	30	39	47	PB10	I/O	TT_a	TMR3_CH2，UART2_TX，UART0_RX，TMR9_BK0，HRPWM_FLT2，HRPWM_EVT0	CMP4_INM0，CMP8_INM1
-	23	31	40	48	VSS	S	-	-	-
23	24	32	41	49	VDD	S	-	-	-
24	25	33	42	50	PB11	I/O	TT_a	TMR3_CH3，UART2_RX，UART0_TX，HRPWM_FLT3，HRPWM_OUT7A	ADC0_IN14，ADC1_IN14，CMP5_INP0
25	26	34	43	51	PB12	I/O	TT_a	SYSDBG0，TMR4_CH0/ETR，I2C1_SBA，SPI1_CS，UART2_TX，UART0_DE，CAN1_RX，TMR9_BK0，HRPWM_OUT2A	ADC0_IN11，ADC3_IN3，CMP6_INM1
26	27	35	44	52	PB13	I/O	TT_a	SYSDBG1，SPI1_CK，TMR9_CH0N，UART2_RE，CAN1_TX，CMP2_OUT，HRPWM_OUT2B	ADC2_IN5，CMP4_INP0
27	28	36	45	53	PB14	I/O	TT_a	SYSDBG2，TMR0_CH0，TMR0_CH1N，SPI1_MISO，TMR9_CH1N，UART2_DE，CMP7_OUT，HRPWM_OUT3A	ADC0_IN5，ADC3_IN4，CMP6_INP0
28	29	37	46	54	PB15	I/O	TT_a	SYSDBG3，TMR0_CH1，TMR0_CH0N，SPI1_MOSI，TMR9_CH2N，CMP3_OUT，HRPWM_OUT3B	ADC1_IN15，ADC3_IN5，CMP5_INM1
-	-	-	47	55	PD8	I/O	TT_a	SYSDBG4，TMR9_CH3N，UART2_TX，	ADC0_IN13，

								XIF_D13，HRPWM_OUT6A	ADC3_IN12
-	-	-	48	56	PD9	I/O	TT_a	SYSDBG5，UART2_RX，XIF_D14，HRPWM_OUT6B	ADC3_IN13
-	-	-	49	57	PD10	I/O	FT_a	SYSDBG6，UART2_RE，XIF_D15，HRPWM_OUT7A	ADC2_IN7，ADC3_IN7，CMP5_INM0
-	-	-	-	58	PD11	I/O	TT_a	SYSDBG7，TMR3_CH0/ETR，I2C1_SBA，HRPWM_OUT7B	ADC2_IN8，ADC3_IN8，CMP5_INP1
-	-	-	-	59	PD12	I/O	TT_a	SYSDBG8，TMR3_CH0/ETR，QEI2_A，SPI0_CK，UART2_DE	ADC2_IN9，ADC3_IN9，CMP4_INP1
-	-	-	-	60	PD13	I/O	FT_a	SYSDBG9，TMR3_CH1，QEI2_B，SPI0_MISO，I2C1_SCL	ADC2_IN10，ADC3_IN10，CMP4_INM1
-	-	-	-	61	PD14	I/O	TT_a	SYSDBG10，TMR3_CH2，QEI2_Z，SPI0_MOSI，I2C1_SDA，XIF_D0	ADC2_IN11，ADC3_IN11，CMP6_INP1
-	-	-	-	62	PD15	I/O	FT_a	SYSDBG11，TMR3_CH3，SPI1_CS，SPI0_CS，I2C1_SBA，XIF_D1	CMP6_INM0
-	-	-	50	63	VSS	S	-	-	-
-	-	-	51	64	VDD	S	-	-	-
29	-	38	52	65	PC6	I/O	FT_f	SYSDBG12，TMR4_CH0/ETR，QEI0_A，TMR10_CH0，I2C0_SCL，I2C1_SCL，CMP5_OUT，HRPWM_EVT9，HRPWM_OUT5A	-
-	-	39	53	66	PC7	I/O	FT_f	SYSDBG13，TMR4_CH1，QEI0_B，TMR10_CH1，I2C0_SDA，I2C1_SDA，CMP4_OUT，HRPWM_FLT4，HRPWM_OUT5B	-
-	-	40	54	67	PC8	I/O	FT_f	SYSDBG14，TMR4_CH2，QEI0_Z，TMR10_CH2，I2C2_SCL，CMP6_OUT，HRPWM_OUT4A	-
-	-	41	55	68	PC9	I/O	FT_f	SYSDBG15，TMR4_CH3，TMR10_CH3，I2C2_SDA，TMR10_BK1，CMP7_OUT，HRPWM_OUT4B	-
30	30	42	56	69	PA8	I/O	FT_a	MCO，TMR3_CH0/ETR，I2C1_SDA，SPI0_CS，TMR9_CH0，I2C2_SCL，CAN1_RX，CMP6_OUT，HRPWM_OUT0A	ADC0_IN16，CMP7_INP1
31	31	43	57	70	PA9	I/O	FT_fa	TMR3_CH2，TMR0_BK0，I2C1_SCL，SPI0_CK，TMR9_CH1，UART0_TX，I2C2_SBA，CAN1_TX，CMP4_OUT，HRPWM_OUT0B	ADC2_IN13，CMP8_INP1
32	32	44	58	71	PA10	I/O	FT_f	TMR3_CH3，TMR2_BK0，I2C1_SBA，SPI0_MISO，TMR9_CH2，UART0_RX，TMR10_BK0，CMP5_OUT，HRPWM_OUT1A	-
33	33	45	59	72	PA11	I/O	FT_u	TMR3_CH0/ETR，TMR9_CH3，SPI0_MOSI，TMR9_CH0N，UART0_RE，CAN0_RX，	USB_DM

								TMR9_BK1，CMP0_OUT，HRPWM_OUT1B	
34	34	46	60	73	PA12	I/O	FT_u	TMR3_CH1，TMR1_CH0，TMR9_CH1N，UART0_DE，CAN0_TX，TMR9_ETR，HRPWM_FLT0，CMP1_OUT，HRPWM_OUT7B	USB_DP
-	35	47	61	74	VSS	S	-	-	-
35	36	48	62	75	VDD	S	-	-	-
36	37	49	63	76	PA13	I/O	FT_f	SWDAT，TMR1_CH0N，TMR4_CH2，I2C0_SCL，UART1_DE，I2C1_SCL，CMP8_OUT	-
37	38	50	64	77	PA14	I/O	FT_f	SWCLK，TMR1_CH1N，I2C0_SDA，TMR10_CH1，UART1_TX，I2C1_SBA，TMR9_BK0，CAN0_RX，HRPWM_FLT0	-
38	39	51	65	78	PA15	I/O	FT_f	-TMR3_CH0/ETR，TMR4_CH0/ETR，I2C0_SCL，SPI1_CS，TMR10_CH0，UART1_RX，I2C1_SDA，TMR9_BK0，SPI0_CS，CAN0_TX，UART3_DE，HRPWM_FLT1	-
39	-	52	66	79	PC10	I/O	FT	QEI1_A，TMR10_CH0N，UART2_TX，SPI0_CK，UART3_TX，CMP7_OUT，HRPWM_FLT5	-
40	-	53	67	80	PC11	I/O	FT_f	TMR3_CH0/ETR，QEI1_B，TMR10_CH1N，UART2_RX，SPI0_MISO，UART3_RX，I2C2_SDA，HRPWM_EVT1	-
-	-	54	68	81	PC12	I/O	FT	TMR3_CH1，QEI1_Z，TMR10_CH2N，UART4_TX，SPI0_MOSI，UART2_TX，HRPWM_EVT0	-
-	-	-	69	82	PD0	I/O	FT	TMR3_CH2，TMR10_CH3N，UART2_RX，CAN0_RX，XIF_D2，HRPWM_FLT7	-
-	-	-	70	83	PD1	I/O	FT	TMR3_CH3，TMR10_CH3，UART2_TX，CAN0_TX，TMR10_BK1，XIF_D3	-
-	-	55	71	84	PD2	I/O	FT	TMR4_CH0/ETR，UART4_RX，TMR10_CH2，UART2_RX，TMR10_BK0，XIF_RST	-
-	-	-	-	85	PD3	I/O	FT	TMR4_CH0/ETR，QEI2_A，UART4_TX，TMR10_CH1，UART1_RE，XIF_CS	-
-	-	-	-	86	PD4	I/O	FT	TMR4_CH1，TMR0_CH0，QEI2_B，TMR10_CH0，UART1_DE，XIF_CVT	-
-	-	-	-	87	PD5	I/O	FT	TMR4_CH2，TMR0_CH0N，QEI2_Z，UART1_TX，XIF_RD	-
-	-	-	-	88	PD6	I/O	FT	TMR4_CH3，TMR0_CH1，UART1_RX，XIF_BSY	-
-	-	-	-	89	PD7	I/O	FT	TMR4_CH2，TMR0_CH1N，SPI1_CS，SPI0_CS，，CMP8_OUT，XIF_CS	-
41	40	56	72	90	PB3	I/O	FT	SWO，TMR3_CH0/ETR，TMR4_CH0/ETR，TMR9_CH0N，SPI1_CK，SPI0_CK，UART1_TX，HRPWM_SCO，CAN0_RX，HRPWM_EVT3，HRPWM_EVT8	-
42	41	57	73	91	PB4	I/O	FT	-TMR1_CH0，TMR3_CH0/ETR，TMR9_CH1N，SPI1_MISO，SPI0_MISO，UART1_RX，	-

								UART4_DE，UART2_DE，TMR2_BK0，CAN0_TX，HRPWM_EVT6	
43	42	58	74	92	PB5	I/O	FT_f	-TMR2_CH0，TMR3_CH1，I2C0_SBA，SPI1_MOSI，SPI0_MOSI，UART1_DE，I2C2_SDA，CAN1_RX，TMR1_BK0，UART4_RE，TMR9_CH2N，HRPWM_EVT5	-
44	43	59	75	93	PB6	I/O	FT	TMR1_CH0N，TMR4_CH0/ETR，TMR10_CH0，TMR10_ETR，UART0_TX，CMP3_OUT，CAN1_TX，TMR10_BK1，HRPWM_SCI，HRPWM_EVT3	-
45	44	60	76	94	PB7	I/O	FT_f	TMR2_CH0N，TMR4_CH1，I2C0_SDA，TMR10_BK0，I2C1_SDA，UART0_RX，CMP2_OUT，UART3_RE，TMR4_CH3，XIF_CVT，HRPWM_EVT2	-
46	45	61	77	95	PB8	I/O	FT_f	BOOT，TMR1_CH0，TMR4_CH2，I2C0_SCL，TMR10_CH1，UART2_RX，CMP0_OUT，CAN1_RX，TMR9_BK0，CAN0_RX，XIF_RST，HRPWM_EVT7	-
47	46	62	78	96	PB9	I/O	FT_f	TMR2_CH0，TMR4_CH3，I2C0_SDA，TMR10_CH2，UART2_TX，CMP1_OUT，CAN1_TX，TMR9_CH2N，CAN0_TX，XIF_RST，HRPWM_EVT4	-
-	-	-	-	97	PE0	I/O	FT	TMR2_CH1，TMR10_CH3N，TMR1_CH0，TMR3_CH0/ETR，TMR10_CH2N，UART0_TX，CMP8_OUT，CAN1_RX，TMR9_CH2，TMR10_ETR，XIF_BSY	-
-	-	-	-	98	PE1	I/O	FT	TMR2_CH1N，TMR10_CH3，TMR2_CH0，TMR9_CH0，UART0_RX，CAN1_TX，XIF_BSY	-
-	47	63	79	99	VSS	S	-	-	-
48	48	64	80	100	VDD	S	-	-	-

**注意：（1）PA13/PA14 为调试引脚，PA13 数据接口内置上拉，PA14 时钟接口内置下拉。*

表 3-3 引脚功能复用表

Pin name	Functions															
	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
	IN	OUT	SYS	TMRx	TMRx	I2C/QEI	SPI/UART	TMRx/SPI	TMRx/UART	I2C/UART	CAN/UART	TMRx/SPI	I2C/CAN	XIF/PDM	HRPWM	Analog function
PA0	-	-	-	TMR3_CH0/ETR	TMR4_CH0/ETR	-	-	TMR10_BK0	TMR10_ETR	QEI0_A	UART1_DE	-	I2C0_SCL	PDM0_CLK	CMP0_OUT	ADC0_IN1,ADC1_IN1,CMP0_INM1,CMP2_INP0
PA1	-	-	-	TMR3_CH1	TMR4_CH1	I2C1_SDA	-	-	UART1_DE	QEI0_B	UART2_TX	TMR0_CH0N	I2C0_SDA	PDM0_DAT	CMP8_OUT	ADC0_IN2,ADC1_IN2,CMP0_INP0
PA2	-	-	-	TMR3_CH2	TMR4_CH2	I2C1_SCL	UART0_TX	SPI0_CS	UART1_TX	QEI0_Z	UART2_RX	TMR0_CH0	I2C0_SBA	-	CMP1_OUT	ADC0_IN3,CMP1_INM1
PA3	-	-	-	TMR3_CH3	TMR4_CH3	TMR1_CH1	UART0_RX	SPI0_CK	UART1_RX	-	-	TMR0_CH1	-	PDM2_CLK	-	ADC0_IN4,CMP1_INP1
PA4	-	-	-	-	TMR4_CH1	TMR1_CH1N	SPI0_CS	SPI0_MISO	SPI1_CS	-	-	TMR0_CH1N	-	PDM2_DAT	-	ADC1_IN16,DAC0_OUT,CMP0_INM0
PA5	-	-	-	TMR3_CH0/ETR	TMR4_CH2	QEI2_A	SPI0_CK	SPI0_MOSI	-	-	-	-	-	PDM3_CLK	CMP3_OUT	ADC1_IN13,DAC1_OUT,CMP1_INM0
PA6	-	-	-	TMR1_CH0	TMR4_CH0/ETR	QEI2_B	SPI0_MISO	TMR9_BK0	UART0_DE	-	-	TMR10_BK0	-	PDM3_DAT	CMP0_OUT	ADC1_IN3,DAC2_OUT,CMP7_INM1
PA7	-	-	-	TMR2_CH0	TMR4_CH1	QEI2_Z	SPI0_MOSI	TMR9_CH0N	UART0_RE	-	-	TMR10_CH0N	-	PDM1_CLK	CMP1_OUT	ADC1_IN4,CMP1_INP0
PA8	-	-	MCO	-	TMR3_CH0/ETR	I2C1_SDA	SPI0_CS	TMR9_CH0	-	I2C2_SCL	CAN1_RX	-	-	CMP6_OUT	HRPWM_OUT0A	ADC0_IN16,CMP7_INP1
PA9	-	-	-	TMR3_CH2	TMR0_BK0	I2C1_SCL	SPI0_CK	TMR9_CH1	UART0_TX	I2C2_SBA	CAN1_TX	-	-	CMP4_OUT	HRPWM_OUT0B	ADC2_IN13,CMP8_INP1
PA10	-	-	-	TMR3_CH3	TMR2_BK0	I2C1_SBA	SPI0_MISO	TMR9_CH2	UART0_RX	-	-	TMR10_BK0	-	CMP5_OUT	HRPWM_OUT1A	-
PA11	-	-	-	TMR3_CH0/ETR	TMR9_CH3	-	SPI0_MOSI	TMR9_CH0N	UART0_RE	-	CAN0_RX	TMR9_BK1	-	CMP0_OUT	HRPWM_OUT1B	USB_DM
PA12	-	-	-	TMR3_CH1	TMR1_CH0	-	-	TMR9_CH1N	UART0_DE	-	CAN0_TX	TMR9_ETR	HRPWM_FLT0	CMP1_OUT	HRPWM_OUT7B	USB_DP
PA13[2]	-	-	SWDAT	TMR1_CH0N	TMR4_CH2	I2C0_SCL	-	-	UART1_DE	I2C1_SCL	-	-	-	CMP8_OUT	-	-
PA14[2]	-	-	SWCLK	TMR1_CH1N	-	I2C0_SDA	-	TMR10_CH1	UART1_TX	I2C1_SBA	-	TMR9_BK0	CAN0_RX	-	HRPWM_FLT0	-
PA15	-	-		TMR3_CH0/ETR	TMR4_CH0/ETR	I2C0_SCL	SPI1_CS	TMR10_CH0	UART1_RX	I2C1_SDA	TMR9_BK0	SPI0_CS	CAN0_TX	UART3_DE	HRPWM_FLT1	-
PB0	-	-	-	TMR4_CH2	-	-	-	TMR9_CH1N	-	-	-	TMR10_CH1N	-	PDM0_DAT	HRPWM_FLT6	ADC0_IN15,ADC2_IN12,CMP3_INP0
PB1	-	-	-	TMR4_CH3	-	-	-	TMR9_CH2N	-	CMP3_OUT	-	TMR10_CH2N	-	UART0_DE	HRPWM_SCO	ADC0_IN12,ADC2_IN1,CMP0_INP1
PB2	-	-	-	TMR4_CH0/ETR	-	I2C2_SBA	-	TMR10_CH0	-	-	-	-	-	-	HRPWM_SCI	ADC1_IN12,CMP3_INM1
PB3	-	-	SWO	TMR3_CH0/ETR	TMR4_CH0/ETR	TMR9_CH0N	SPI1_CK	SPI0_CK	UART1_TX	-	-	HRPWM_SCO	CAN0_RX	HRPWM_EVT3	HRPWM_EVT8	-
PB4	-	-		TMR1_CH0	TMR3_CH0/ETR	TMR9_CH1N	SPI1_MISO	SPI0_MISO	UART1_RX	UART4_DE	UART2_DE	TMR2_BK0	CAN0_TX	-	HRPWM_EVT6	-
PB5	-	-		TMR2_CH0	TMR3_CH1	I2C0_SBA	SPI1_MOSI	SPI0_MOSI	UART1_DE	I2C2_SDA	CAN1_RX	TMR1_BK0	UART4_RE	TMR9_CH2N	HRPWM_EVT5	-
PB6	-	-	-	TMR1_CH0N	TMR4_CH0/ETR	-	TMR10_CH0	TMR10_ETR	UART0_TX	CMP3_OUT	CAN1_TX	TMR10_BK1	-	HRPWM_SCI	HRPWM_EVT3	-
PB7	-	-	-	TMR2_CH0N	TMR4_CH1	I2C0_SDA	TMR10_BK0	I2C1_SDA	UART0_RX	CMP2_OUT	UART3_RE	TMR4_CH3	-	XIF_CVT	HRPWM_EVT2	-
PB8	-	-	BOOT	TMR1_CH0	TMR4_CH2	I2C0_SCL	-	TMR10_CH1	UART2_RX	CMP0_OUT	CAN1_RX	TMR9_BK0	CAN0_RX	XIF_RST	HRPWM_EVT7	-
PB9	-	-	-	TMR2_CH0	TMR4_CH3	I2C0_SDA	-	TMR10_CH2	UART2_TX	CMP1_OUT	CAN1_TX	TMR9_CH2N	CAN0_TX	XIF_RST	HRPWM_EVT4	-
PB10	-	-	-	TMR3_CH2	-	-	-	-	UART2_TX	UART0_RX	-	TMR9_BK0	-	HRPWM_FLT2	HRPWM_EVT0	CMP4_INM0,CMP8_INM1
PB11	-	-	-	TMR3_CH3	-	-	-	-	UART2_RX	UART0_TX	-	-	-	HRPWM_FLT3	HRPWM_OUT7A	ADC0_IN14,ADC1_IN14,CMP5_INP0
PB12	-	-	SYSDBG0	-	TMR4_CH0/ETR	I2C1_SBA	SPI1_CS	-	UART2_TX	UART0_DE	CAN1_RX	TMR9_BK0	-	-	HRPWM_OUT2A	ADC0_IN11,ADC3_IN3,CMP6_INM1
PB13	-	-	SYSDBG1	-	-	-	SPI1_CK	TMR9_CH0N	UART2_RE	-	CAN1_TX	-	-	CMP2_OUT	HRPWM_OUT2B	ADC2_IN5,CMP4_INP0
PB14	-	-	SYSDBG2	TMR0_CH0	TMR0_CH1N	-	SPI1_MISO	TMR9_CH1N	UART2_DE	-	-	-	-	CMP7_OUT	HRPWM_OUT3A	ADC0_IN5,ADC3_IN4,CMP6_INP0
PB15	-	-	SYSDBG3	TMR0_CH1	TMR0_CH0N	-	SPI1_MOSI	TMR9_CH2N	-	-	-	-	-	CMP3_OUT	HRPWM_OUT3B	ADC1_IN15,ADC3_IN5,CMP5_INM1
PC0	-	-	-	TMR3_CH0/ETR	-	QEI2_A	-	TMR9_CH0	-	UART0_RX	-	-	PDM3_DAT	PDM0_CLK	-	ADC0_IN6,ADC1_IN6,CMP2_INM1
PC1	-	-	-	TMR3_CH1	-	QEI2_B	-	TMR9_CH1	-	UART0_TX	-	TMR10_CH1N	PDM3_CLK	PDM0_DAT	-	ADC0_IN7,ADC1_IN7,CMP2_INP1
PC2	-	-	-	TMR3_CH2	-	QEI2_Z	-	TMR9_CH2	-	UART0_DE	-	TMR10_CH1	-	PDM1_CLK	CMP2_OUT	ADC0_IN8,ADC1_IN8
PC3	-	-	-	TMR3_CH3	TMR2_CH0	-	-	TMR9_CH3	-	UART0_RE	-	TMR9_BK1	-	PDM1_DAT	-	ADC0_IN9,ADC1_IN9
PC4	-	-	-	TMR0_CH1	-	I2C1_SCL	-	-	UART0_TX	-	-	TMR9_ETR	-	PDM1_DAT	HRPWM_EVT1	ADC1_IN5,CMP7_INP0
PC5	-	-	-	TMR0_CH0	-	I2C1_SDA	-	TMR9_CH3N	UART0_RX	-	-	TMR0_BK0	-	PDM0_CLK	HRPWM_EVT9	ADC1_IN11,CMP8_INP0
PC6	-	-	SYSDBG12	TMR4_CH0/ETR	-	QEI0_A	-	TMR10_CH0	I2C0_SCL	I2C1_SCL	-	-	CMP5_OUT	HRPWM_EVT9	HRPWM_OUT5A	-
PC7	-	-	SYSDBG13	TMR4_CH1	-	QEI0_B	-	TMR10_CH1	I2C0_SDA	I2C1_SDA	-	-	CMP4_OUT	HRPWM_FLT4	HRPWM_OUT5B	-
PC8	-	-	SYSDBG14	TMR4_CH2	-	QEI0_Z	-	TMR10_CH2	-	I2C2_SCL	-	-	CMP6_OUT	-	HRPWM_OUT4A	-

PC9	-	-	SYSDBG15	TMR4_CH3	-	-	-	TMR10_CH3	-	I2C2_SDA	-	TMR10_BK1	CMP7_OUT	-	HRPWM_OUT4B	-
PC10	-	-	-	-	QEI1_A	TMR10_CH0N	UART2_TX	SPI0_CK	UART3_TX	-	-	-	-	CMP7_OUT	HRPWM_FLT5	-
PC11	-	-	-	TMR3_CH0/ETR	QEI1_B	TMR10_CH1N	UART2_RX	SPI0_MISO	UART3_RX	I2C2_SDA	-	-	-	-	HRPWM_EVT1	-
PC12	-	-	-	TMR3_CH1	QEI1_Z	TMR10_CH2N	UART4_TX	SPI0_MOSI	UART2_TX	-	-	-	-	-	HRPWM_EVT0	-
PC13	-	-	SWO	TMR0_CH0	-	TMR2_CH1	-	TMR10_CH3N	TMR9_CH0N	-	QEI1_A	TMR9_BK0	-	PDM2_CLK	-	-
PC14	-	-	-	TMR0_CH1	TMR0_CH0N	TMR2_CH1N	-	TMR10_CH2	-	-	QEI1_B	-	-	PDM2_DAT	-	-
PC15	-	-	-	TMR1_CH0	TMR0_CH1N	-	-	TMR10_CH3	-	-	QEI1_Z	-	-	PDM3_CLK	-	-
PD0	-	-	-	TMR3_CH2	-	TMR10_CH3N	-	-	UART2_RX	-	CAN0_RX	-	-	XIF_D2	HRPWM_FLT7	-
PD1	-	-	-	TMR3_CH3	-	-	-	TMR10_CH3	UART2_TX	-	CAN0_TX	TMR10_BK1	-	XIF_D3	-	-
PD2	-	-	-	-	TMR4_CH0/ETR	-	UART4_RX	TMR10_CH2	UART2_RX	-	-	TMR10_BK0	-	XIF_RST	-	-
PD3	-	-	-	TMR4_CH0/ETR	-	QEI2_A	UART4_TX	TMR10_CH1	UART1_RE	-	-	-	-	XIF_CS	-	-
PD4	-	-	-	TMR4_CH1	TMR0_CH0	QEI2_B	-	TMR10_CH0	UART1_DE	-	-	-	-	XIF_CVT	-	-
PD5	-	-	-	TMR4_CH2	TMR0_CH0N	QEI2_Z	-	-	UART1_TX	-	-	-	-	XIF_RD	-	-
PD6	-	-	-	TMR4_CH3	TMR0_CH1	-	-	-	UART1_RX	-	-	-	-	XIF_BSY	-	-
PD7	-	-	-	TMR4_CH2	TMR0_CH1N	-	SPI1_CS	SPI0_CS		-	-	-	CMP8_OUT	XIF_CS	-	-
PD8	-	-	SYSDBG4	-	-	-	-	TMR9_CH3N	UART2_TX	-	-	-	-	XIF_D13	HRPWM_OUT6A	ADC0_IN13,ADC3_IN12
PD9	-	-	SYSDBG5	-	-	-	-	-	UART2_RX	-	-	-	-	XIF_D14	HRPWM_OUT6B	ADC3_IN13
PD10	-	-	SYSDBG6	-	-	-	-	-	UART2_RE	-	-	-	-	XIF_D15	HRPWM_OUT7A	ADC2_IN7,ADC3_IN7,CMP5_INM0
PD11	-	-	SYSDBG7	-	TMR3_CH0/ETR	-	-	-	-	I2C1_SBA	-	-	-	-	HRPWM_OUT7B	ADC2_IN8,ADC3_IN8,CMP5_INP1
PD12	-	-	SYSDBG8	TMR3_CH0/ETR	-	QEI2_A	-	SPI0_CK	UART2_DE	-	-	-	-	-	-	ADC2_IN9,ADC3_IN9,CMP4_INP1
PD13	-	-	SYSDBG9	TMR3_CH1	-	QEI2_B	-	SPI0_MISO	-	I2C1_SCL	-	-	-	-	-	ADC2_IN10,ADC3_IN10,CMP4_INM1
PD14	-	-	SYSDBG10	TMR3_CH2	-	QEI2_Z	-	SPI0_MOSI	-	I2C1_SDA	-	-	-	XIF_D0	-	ADC2_IN11,ADC3_IN11,CMP6_INP1
PD15	-	-	SYSDBG11	TMR3_CH3	-	-	SPI1_CS	SPI0_CS	-	I2C1_SBA	-	-	-	XIF_D1	-	CMP6_INM0
PE0	-	-	-	TMR2_CH1	TMR10_CH3N	TMR1_CH0	TMR3_CH0/ETR	TMR10_CH2N	UART0_TX	CMP8_OUT	CAN1_RX	TMR9_CH2	TMR10_ETR	XIF_BSY	-	-
PE1	-	-	-	TMR2_CH1N	TMR10_CH3	TMR2_CH0	-	TMR9_CH0	UART0_RX	-	CAN1_TX	-	-	XIF_BSY	-	-
PE2	-	-	TRACECK	TMR4_CH0/ETR	-	QEI0_A	SPI0_CK	TMR10_CH0	-	UART0_RX	-	SPI1_CS	-	PDM0_CLK	-	-
PE3	-	-	TRACED0	TMR4_CH1	-	QEI0_B	SPI0_CS	TMR10_CH1	-	UART0_TX	-	SPI1_CK	-	PDM0_DAT	-	-
PE4	-	-	TRACED1	TMR4_CH2	-	QEI0_Z	SPI0_CS	TMR10_CH0N	-	UART0_DE	-	SPI1_MISO	-	PDM1_CLK	-	-
PE5	-	-	TRACED2	TMR4_CH3	-	-	SPI0_MISO	TMR10_CH1N	-	-	-	SPI1_MOSI	-	PDM1_DAT	-	-
PE6	-	-	TRACED3	-	-	-	SPI0_MOSI	TMR10_CH2N	-	-	-	-	-	-	-	-
PE7	-	-	-	-	-	-	-	TMR9_ETR	UART1_TX	-	-	-	-	XIF_D4	HRPWM_EVT2	ADC2_IN4,CMP3_INP1
PE8	-	-	-	TMR3_CH0/ETR	TMR4_CH2	QEI0_A	-	TMR9_CH0N	UART1_RX	-	-	-	-	XIF_D5	HRPWM_FLT2	ADC2_IN6,ADC3_IN6,CMP3_INM0
PE9	-	-	-	TMR3_CH1	TMR4_CH3	QEI0_B	-	TMR9_CH0	UART2_TX	-	-	-	-	XIF_D6	-	ADC2_IN2,CMP8_INM0
PE10	-	-	-	TMR3_CH2	-	QEI0_Z	-	TMR9_CH1N	UART3_RX	-	-	-	-	XIF_D7	-	ADC2_IN14,ADC3_IN14
PE11	-	-	-	TMR3_CH3	-	I2C1_SCL	SPI0_CS	TMR9_CH1	-	-	-	-	-	XIF_D8	-	ADC2_IN15,ADC3_IN15
PE12	-	-	-	TMR4_CH0/ETR	-	I2C1_SDA	SPI0_CK	TMR9_CH2N	-	QEI1_A	-	-	-	XIF_D9	-	ADC2_IN16,ADC3_IN16
PE13	-	-	-	TMR4_CH1	-	I2C1_SBA	SPI0_MISO	TMR9_CH2	-	QEI1_B	-	-	-	XIF_D10	-	ADC2_IN3
PE14	-	-	-	TMR4_CH2	-	-	SPI0_MOSI	TMR9_CH3	-	QEI1_Z	-	TMR9_BK1	-	XIF_D11	-	ADC3_IN1
PE15	-	-	-	TMR4_CH3	-	-	-	TMR9_CH3N	UART2_RX	-	-	TMR9_BK0	-	XIF_D12	-	ADC3_IN2
PF0	-	-	-	TMR4_CH1	-	I2C1_SDA	SPI1_CS	TMR9_CH2N	-	-	-	-	-	-	-	ADC0_IN10,CMP7_INM0,XOSCI
PF1	-	-	-	-	-	I2C1_SCL	SPI1_CK	-	-	-	-	-	-	-	-	ADC1_IN10,CMP2_INM0,XOSCO
PF2	-	-	-	-	TMR2_CH0N	I2C1_SBA	-	TMR10_CH2	-	-	UART1_RE	-	-	-	-	-
PF3	-	-	-	TMR1_CH1	TMR1_CH0N	TMR3_CH3	SPI1_CK	-	TMR0_CH0	-	-	TMR10_BK0	-	PDM3_DAT	-	-
PF4	-	-	-	TMR4_CH0/ETR	TMR1_CH1N	-	SPI1_CK	-	TMR0_CH1	-	-	TMR10_BK1	-	-	-	-

*注意：（1）PA13/PA14 为调试引脚，PA13 数据接口内置上拉，PA14 时钟接口内置下拉。

4 存储器和总线架构

4.1 系统架构

芯片主系统由多层 AHB 总线矩阵互联，如下图所示：

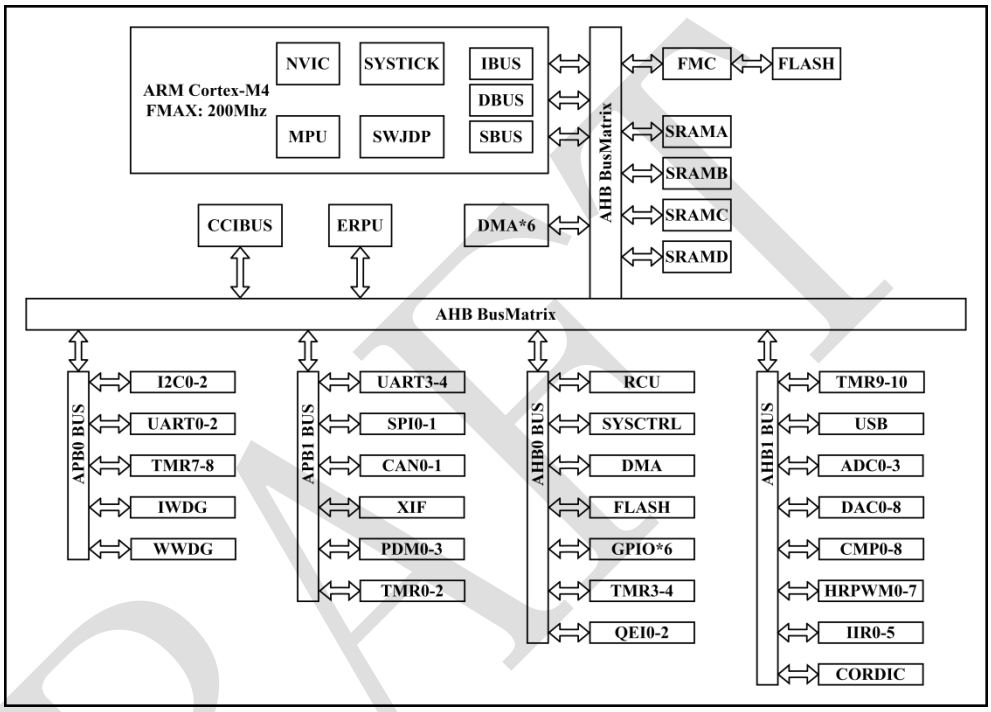


图 4-1 总线架构

4.2 总线矩阵

主系统由 32 位多层 AHB 总线矩阵构成，可实现以下部分的互连：

- 五条主控总线：
  - Cortex™-M4 内核 I 总线、D 总线和 S 总线
  - DMA 存储器总线 0
  - DMA 存储器总线 1
- 八条被控总线：
  - FLASH ICode 总线
  - FLASH DCode 总线
  - SRAM（128KB）
  - AHB0 外设
  - AHB1 外设
  - APB0 外设
  - APB1 外设

通过总线矩阵可以实现主控总线到被控总线的访问，这样即使在多个高速外设同时运行期间，

系统也可以实现并发访问和高效运行，此架构如图 4-2 所示。

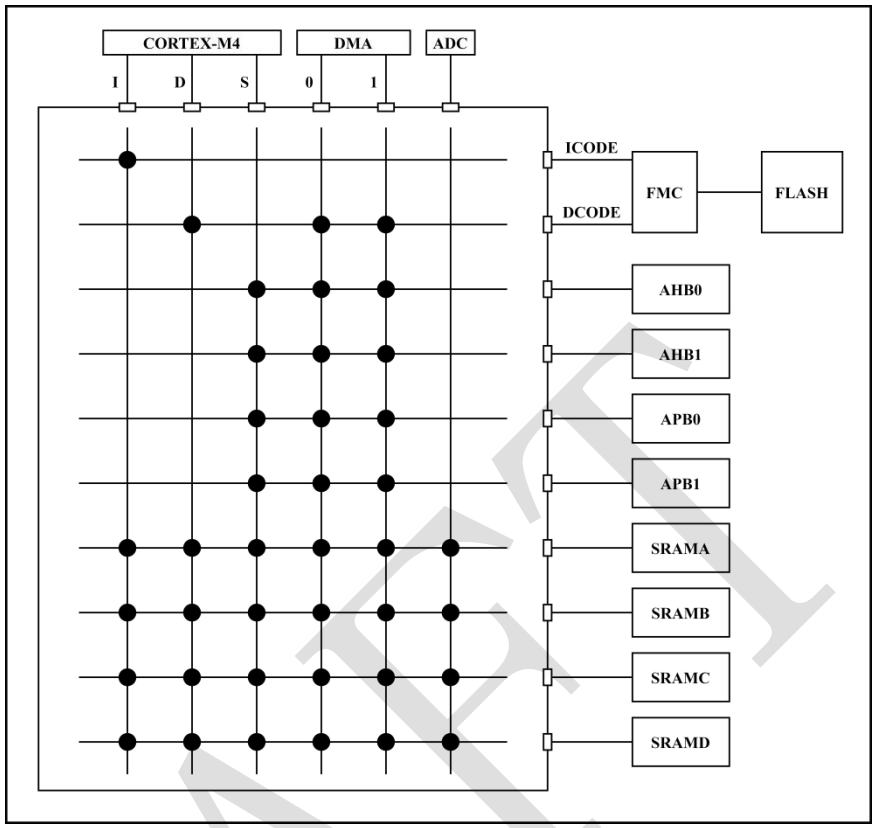


图 4-2　系统总线互联矩阵

I 总线：此总线用于将 Cortex™-M4 内核的指令总线连接到总线矩阵，内核通过此总线获取指令。此总线访问的对象是包含代码的存储器（内部 FLASH/SRAM）。

D 总线：此总线用于将 Cortex™-M4 数据总线连接到总线矩阵，内核通过此总线进行立即数加载和调试访问。此总线访问的对象是包含代码或数据的存储器（内部 FLASH/SRAM）。

S 总线：此总线用于 Cortex™-M4 内核的系统总线连接到总线矩阵。此总线用于访问位于外设或 SRAM 中的数据。也可通过此总线获取指令（效率低于 ICode）。此总线访问的对象是 128KB 的内部 SRAM 及 AHB0 外设、AHB1 外设、APB0 外设、APB1 外设。

DMA 存储器总线：此总线用于将 DMA 存储器总线主接口连接到总线矩阵。DMA 通过此总线来执行存储器数据的传入和传出。此总线访问的对象是数据存储器：内部 FLASH/SRAM。

总线矩阵：总线矩阵用于主控总线之间的访问仲裁管理。

AHB/APB 总线桥：通过一个 AHB 和两个 APB 总线桥（APB0/APB1），可在 AHB 总线与两个 APB 总线之间实现完全同步的连接，从而灵活选择外设频率。

4.3　存储器地址映射

程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个顺序的 4GB 地址空间内。

各字节按小端格式在存储器中编码，字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

未分配给片上存储器和外设的所有存储区域均视为“保留区”，有关外设寄存器映射的详细信息，请参见“4.4 外设寄存器映射”章节。

遵循 ARM®Cortex™-M4 对存储器的规定，存储器基本组织架构如下：

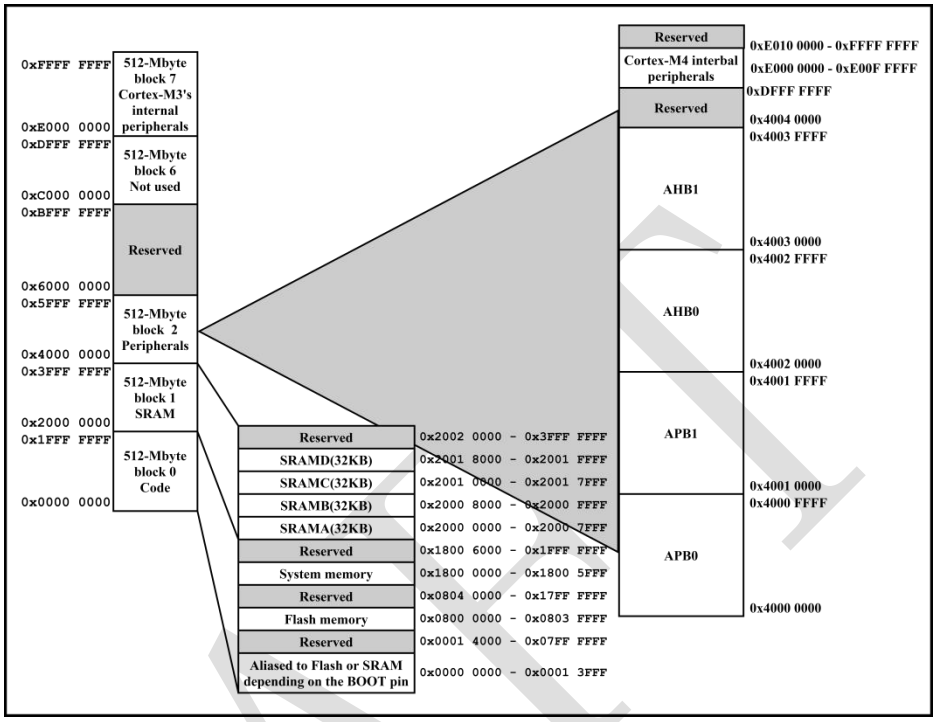


图 4-3 存储映射图

4.4 外设寄存器映射

下表提供了芯片外设地址边界划分，详细请参考表 4-1：

表 4-1 芯片外设地址映射表

总线	边界地址	外设
Reserved	0xE010 0000 - 0xFFFF FFFF	Reserved
Cortex™-M4	0xE000 0000 - 0xE00F FFFF	Cortex™-M4 internal peripherals
Reserved	0x4004 0000 - 0xDFFF FFFF	Reserved
AHB1	0x4003 F000 - 0x4003 FFFF	CORDIC
	0x4003 E500 - 0x4003 E5FF	IIR5
	0x4003 E400 - 0x4003 E4FF	IIR4
	0x4003 E300 - 0x4003 E3FF	IIR3
	0x4003 E200 - 0x4003 E2FF	IIR2
	0x4003 E100 - 0x4003 E1FF	IIR1
	0x4003 E000 - 0x4003 E0FF	IIR0
	0x4003 BF00 - 0x4003 BFFF	HRPWM-COMMON
	0x4003 B800 - 0x4003 B8FF	HRPWM7
	0x4003 B700 - 0x4003 B7FF	HRPWM6
	0x4003 B600 - 0x4003 B6FF	HRPWM5
	0x4003 B500 - 0x4003 B5FF	HRPWM4

	0x4003 B400 - 0x4003 B4FF	HRPWM3
	0x4003 B300 - 0x4003 B3FF	HRPWM2
	0x4003 B200 - 0x4003 B2FF	HRPWM1
	0x4003 B100 - 0x4003 B1FF	HRPWM0
	0x4003 B000 - 0x4003 B0FF	HRPWM-MASTER
	0x4003 A800 - 0x4003 A8FF	CMP8
	0x4003 A700 - 0x4003 A7FF	CMP7
	0x4003 A600 - 0x4003 A6FF	CMP6
	0x4003 A500 - 0x4003 A5FF	CMP5
	0x4003 A400 - 0x4003 A4FF	CMP4
	0x4003 A300 - 0x4003 A3FF	CMP3
	0x4003 A200 - 0x4003 A2FF	CMP2
	0x4003 A100 - 0x4003 A1FF	CMP1
	0x4003 A000 - 0x4003 A0FF	CMP0
	0x4003 9800 - 0x4003 98FF	DAC8
	0x4003 9700 - 0x4003 97FF	DAC7
	0x4003 9600 - 0x4003 96FF	DAC6
	0x4003 9500 - 0x4003 95FF	DAC5
	0x4003 9400 - 0x4003 94FF	DAC4
	0x4003 9300 - 0x4003 93FF	DAC3
	0x4003 9200 - 0x4003 92FF	DAC2
	0x4003 9100 - 0x4003 91FF	DAC1
	0x4003 9000 - 0x4003 90FF	DAC0
	0x4003 8C00 - 0x4003 8FFF	ADC3
	0x4003 8800 - 0x4003 8BFF	ADC2
	0x4003 8400 - 0x4003 87FF	ADC1
	0x4003 8000 - 0x4003 83FF	ADC0
	0x4003 5000 - 0x4003 5FFF	USB
	0x4003 1000 - 0x4003 1FFF	TMR10
	0x4003 0000 - 0x4003 0FFF	TMR9
AHB0	0x4002 F000 - 0x4002 FFFF	QEI2
	0x4002 E000 - 0x4002 EFFF	QEI1
	0x4002 D000 - 0x4002 DFFF	QEI0
	0x4002 B000 - 0x4002 BFFF	TMR4
	0x4002 A000 - 0x4002 AFFF	TMR3
	0x4002 9000 - 0x4002 9FFF	GPIOF
	0x4002 8000 - 0x4002 8FFF	GPIOE
	0x4002 7000 - 0x4002 7FFF	GPIOD
	0x4002 6000 - 0x4002 6FFF	GPIOC
	0x4002 5000 - 0x4002 5FFF	GPIOB
	0x4002 4000 - 0x4002 4FFF	GPIOA
	0x4002 3000 - 0x4002 3FFF	FLASH
	0x4002 2000 - 0x4002 2FFF	DMA

	0x4002 1000 - 0x4002 1FFF	SYSCTRL
	0x4002 0000 - 0x4002 0FFF	RCU
APB1	0x4001 A000 - 0x4001 AFFF	TMR2
	0x4001 9000 - 0x4001 9FFF	TMR1
	0x4001 8000 - 0x4001 8FFF	TMR0
	0x4001 7300 - 0x4001 73FF	PDM3
	0x4001 7200 - 0x4001 72FF	PDM2
	0x4001 7100 - 0x4001 71FF	PDM1
	0x4001 7000 - 0x4001 70FF	PDM0
	0x4001 6000 - 0x4001 6FFF	XIF
	0x4001 5000 - 0x4001 5FFF	CAN1
	0x4001 4000 - 0x4001 4FFF	CAN0
	0x4001 3000 - 0x4001 3FFF	SPI1
	0x4001 2000 - 0x4001 2FFF	SPI0
	0x4001 1000 - 0x4001 1FFF	UART4
	0x4001 0000 - 0x4001 0FFF	UART3
APB0	0x4000 D000 - 0x4000 DFFF	WWDG
	0x4000 C000 - 0x4000 CFFF	IWDG
	0x4000 9000 - 0x4000 9FFF	TMR8
	0x4000 8000 - 0x4000 8FFF	TMR7
	0x4000 5000 - 0x4000 5FFF	UART2
	0x4000 4000 - 0x4000 4FFF	UART1
	0x4000 3000 - 0x4000 3FFF	UART0
	0x4000 2000 - 0x4000 2FFF	I2C2
	0x4000 1000 - 0x4000 1FFF	I2C1
	0x4000 0000 - 0x4000 0FFF	I2C0

4.4.1 SRAM

系统 SRAM 可按字节、半字（16 位）或全字（32 位）进行访问，其读写操作均以 CPU 速度执行，且等待周期为 0，系统 SRAM 分为四个块：

- 映射在地址 0x2000 0000 的 SRAMA（32KB）块，可供所有 AHB 主控总线，DMA 主控总线及 ADC、HRPWM 访问。
- 映射在地址 0x2000 8000 的 SRAMB（32KB）块，可供所有 AHB 主控总线，DMA 主控总线及 ADC、HRPWM 访问。
- 映射在地址 0x2001 0000 的 SRAMC（32KB）块，可供所有 AHB 主控总线，DMA 主控总线及 ADC、HRPWM 访问。
- 映射在地址 0x2001 8000 的 SRAMD（32KB）块，可供所有 AHB 主控总线，DMA 主控总线及 ADC、HRPWM 访问。
- 当 BOOT=SRAM 时，系统 SRAM 将被映射到 0x0 地址以及 0x1000 0000 地址上。
- 当 BOOT=FLASH 时，系统 SRAM 块被映射到 0x1000 0000 地址上。

如果选择从 SRAM 自举，则 CPU 可通过系统总线或 I-Code/D-Code 总线访问系统 SRAM。要在 SRAM 执行期间获得最佳的性能，应选择物理重映射（通过自举管脚）。

4.4.2 FLASH

FLASH 控制器用于管理 CPU 通过 I 总线（I-Code）及 D 总线（D-Code）对 FLASH 的访问。通过 FLASH 控制器，可以对 FLASH 存储体执行擦除/编程、读/写保护等功能的操作。

FLASH 存储体结构如下：

- 主存储器块分为多个扇区。
- 选项字节，用于配置读保护、写保护等功能。

4.5 自举配置

存储器地址采用绝对地址映射，代码区域的起始地址为 0x0000 0000（可通过 I-Code/D-Code 总线访问），而数据区域的起始地址为 0x2000 0000（可通过系统总线访问）。芯片上电启动过程中，CPU 始终通过 I-Code 总线获取复位向量，这意味着只有代码区域（通常为 FLASH）可以提供自举空间。芯片结合调试工具也可以从其它存储器（如内部 SRAM）进行自举。在芯片上电启动中，可通过 BOOT 引脚选择两种不同的自举模式，如表 4-2 所示。

表 4-2 自举模式（默认从 BROM 自举）

BOOT LOCK	Hard/Soft(Sel2) 0: Soft 1: Hard - Pin	Pin(PB8) Default: PullUp	OPT2(Sel1)	OPT1(Sel0)	BOOT Type
1	x	x	x	x	FLASH BOOT
0	1	0	x	x	FLASH BOOT
0	1	1	x	0	SRAM BOOT
0	1	1	x	1	BROM BOOT
0	0	x	1	x	FLASH BOOT
0	0	x	0	0	SRAM BOOT
0	0	x	0	1	BROM BOOT

在芯片上电复位后，在 SYSCLK 的第四个上升沿将锁存 BOOT 引脚的值，用户可以通过设置 BOOT 引脚的电平来选择需要的自举模式。

BOOT 引脚只有在芯片重新上电时，才会对 BOOT 引脚重新采样。因此，芯片上电复位结束后，CPU 将从 0x0000 0000 地址获取栈顶值，然后从 0x0000 0004 地址获取程序入口地址并开始执行程序代码。

5 嵌入式 FLASH 接口（FLASH）

5.1 简介

FLASH Controller 接口用于管理 CPU 通过 I 总线（I-Code Bus）及 D 总线（D-Code Bus）对 FLASH 的访问。

通过 FLASH Controller 可以对 FLASH 存储体执行擦除与编程操作、实现读/写保护机制。

5.2 主要特性

FLASH 主要具有以下特性：

- 存储器高达 256KB（128KB x2）
- 支持按整片/分块/扇区擦除（Chip/Bank/Sector Erase）
- FLASH 作为启动代码存储空间(默认的自举存储)和执行用户代码
- FLASH 支持 256Kbyte 模式(Double Bank、Single Bank 模式)
  - FLASH 256Kbyte 双 Bank 模式下，两块可以相互独立运行，支持在线升级方案
  - FLASH 256Kbyte 单 Bank 模式下，两块并行处理，支持 128 位预取
  - FLASH 256Kbyte 单 Bank 模式下，共 32 个 Sector 扇区，每个 Sector 扇区大小为 8KB，256Kbyte 双 Bank 模式下，共 64 个 Sector 扇区，每个 Sector 扇区大小为 4KB
  - FLASH 选项字节，用于配置读写保护、看门狗默认使能、FLASH Bank 模式及 Boot 方式
- FLASH 支持 128Kbyte 模式(Single Bank 模式)
  - FLASH 128Kbyte 单 Bank 模式下，两块并行处理，支持 128 位预取
  - FLASH 128Kbyte 单 Bank 模式下，共 16 个 Sector 扇区，Sector 扇区大小为 8KB
  - FLASH 选项字节，用于配置读写保护、看门狗默认使能、FLASH Bank 模式及 Boot 方式
- FLASH 支持 64Bit 位宽编程功能，支持按字节/半字/字读取 FLASH
- 增强安全功能
  - FLASH 读保护功能（RDP）
  - FLASH 写保护功能（WRP）
- FLASH I/D 总线支持预取、缓存功能
  - FLASH I-Code Bus 上 1024Byte 缓存
  - FLASH D-Code Bus 上 256Byte 缓存
- 误码校验（ECC），支持纠一检二
- 支持低功耗模式

5.3 结构框图

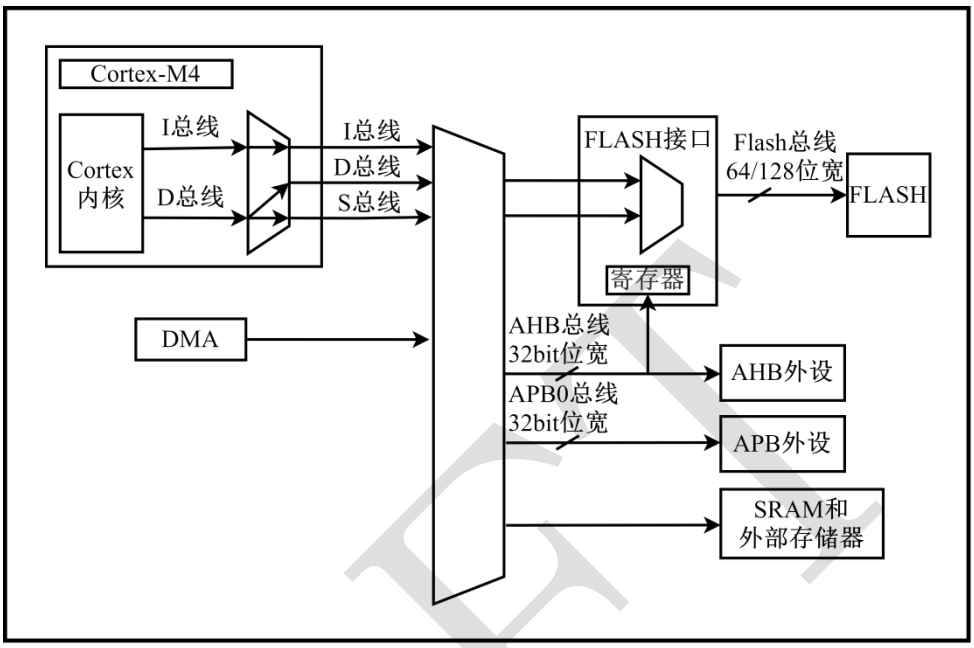


图 5-1 FLASH 顶层结构框图

5.4 功能描述

5.4.1 实时加速器

为了高效地发挥处理器的性能，通过加速器实现指令预取队列及分支缓存机制，减少 FLASH 读取指令时等待的情况，从而提高了 64/128 位 FLASH 程序的执行速度。

指令预取

FLASH 每次读操作可读取 64/128 位数据，即可以是 2/4 条 32 位指令，也可以是 4/8 条 16 位指令，具体取决于烧写在 FLASH 中的程序。因此对于顺序执行的代码，至少需要 RC+2 个 CPU 周期来执行前一次读取的 64/128 位指令行。

在 CPU 读取当前指令行时，可使用 I-Code 总线的预取操作读取 FLASH 中的下一个连续存放的 64/128 位指令行。可通过 FLASH_CR 寄存器中的 DBPE/ IBPE 位置 1 来使能预取功能。

图 5-2、图 5-3 为需要 3 个等待周期访问 FLASH 时连续 32 位指令的执行过程，分别介绍了不使用和使用预取操作两种情况（以 3 个等待周期为示例，实际需要 RC+2 个等待周期）。

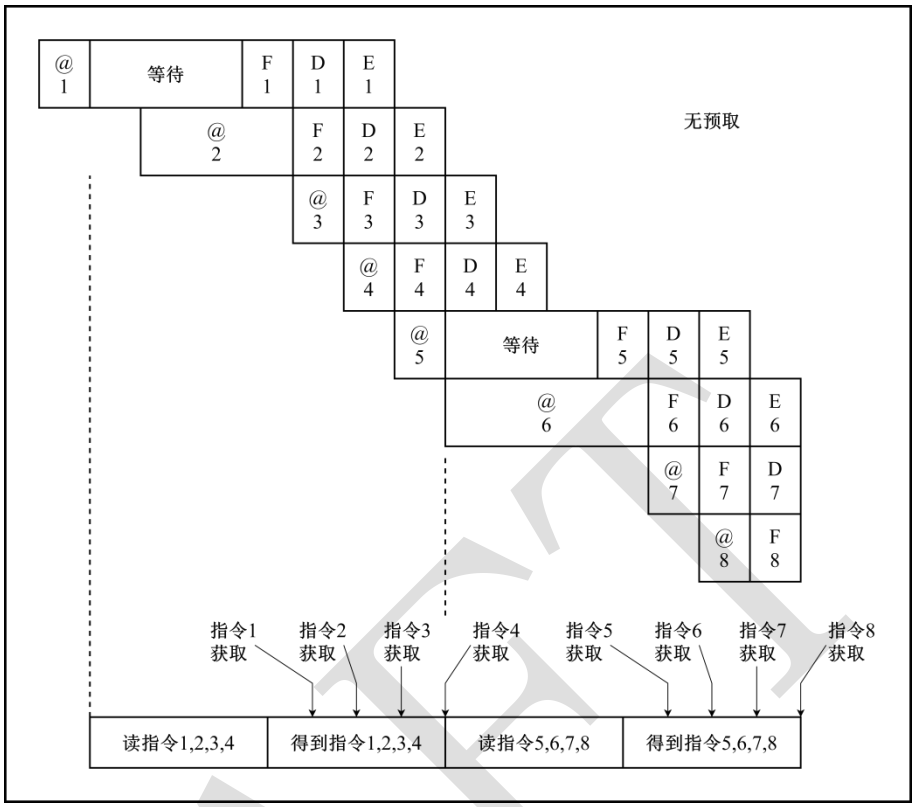


图 5-2　连续 32 位指令的执行过程（不使用预取操作）

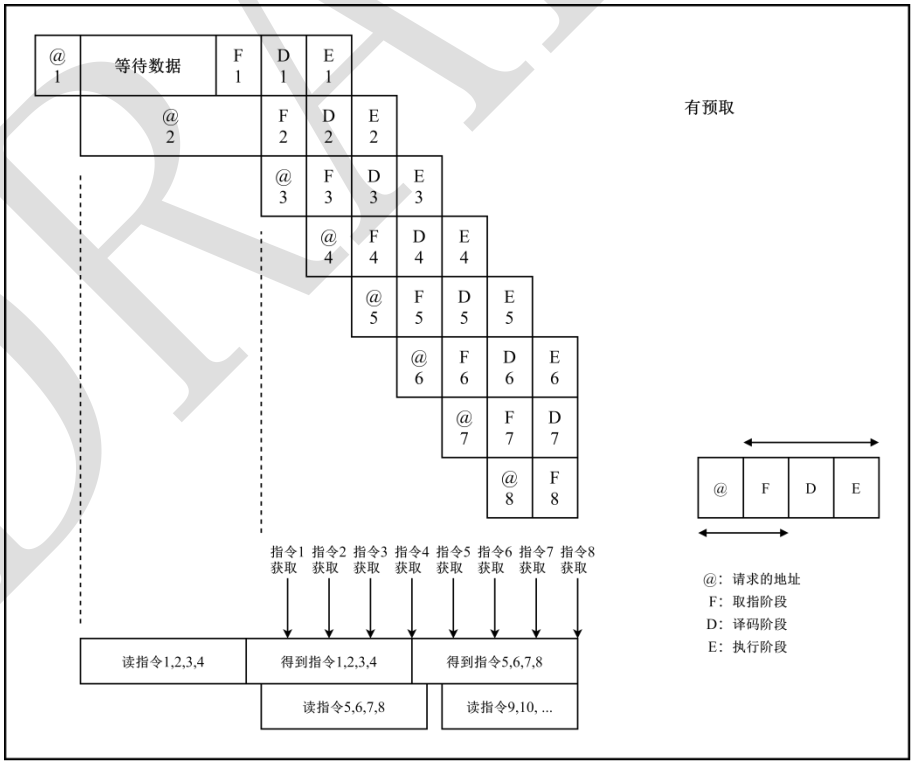


图 5-3　连续 32 位指令的执行过程（使用预取操作）

处理非顺序执行的代码（有分支）时，指令可能并不存在于当前使用的或预取的指令行中。这种情况下，CPU 等待时间至少等于 RC+2 个等待周期数。

指令缓存存储器

为了减少因指令跳转而产生的时间消耗，可将 256 条 32 位（1024 Byte）指令保存到指令缓存（I-Cache）存储器中。每当出现指令缺失（即请求的指令未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存区中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用最近最少使用策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

数据管理

在 CPU 流水线执行阶段，将通过 D-Code 总线访问 FLASH 中的数据缓冲池。因此，直到提供了请求的数据后，CPU 流水线才会继续执行。为了减少因此而产生的时间损耗，通过 AHB 数据总线 D-Code 进行的访问优先于通过 AHB 指令总线 I-Code 进行的访问。

为了应对频繁使用某些数据的情况，同样添加了数据缓存（D-Cache）功能，此特性的工作原理与指令缓存存储器类似，但保留的数据大小限制在 64 条 32 位（256 Byte）以内。

5.4.2 擦除和编程操作

执行任何 FLASH 编程/擦除操作时，需要先配置好 FLASH 的时钟，如果在 FLASH 操作期间发生器件复位，将无法保证 FLASH 中的内容。

在对 FLASH 执行写入或擦除操作期间，任何读取 FLASH 的尝试都会导致总线阻塞。只有在完成编程操作后，才能正确处理读操作。这意味着，写入/擦除操作进行期间不能从 FLASH 中执行代码或数据获取操作。

注意：FLASH 所有操作均为串行执行，确保上一笔操作完成后再发起下一笔操作。

5.4.2.1 FLASH 控制寄存器解锁

FLASH 中用于解锁控制寄存器的 Key 值有以下几种：

- KEY1 = 0x3FAC
- KEY2 = 0x87E4
- KEY3 = 0x124A
- KEY4 = 0xBC7F

FLASH 的写/擦除控制解锁

复位后，FLASH 控制寄存器（FLASH_CR）不允许执行写/擦操作（读不需要解锁），以防因电气干扰等原因出现对 FLASH 的意外操作。当需要执行 FLASH 的写/擦操作时，必须先执行解锁，然后发起相应的操作。

- 解锁方式
  向 FLASH 密钥寄存器（FLASH_KR）中依次写入 KEY1 和 KEY2，成功解锁后，FLASH_CR 寄存器中的 LCK 位将会自动置零，这时即可对 FLASH 进行写/擦除操作。

- 加锁方式
  完成所需的写/擦除操作后，通过软件将 FLASH_CR 寄存器中的 LCK 位置 1，将重新锁定 FLASH，实现加锁功能，下次再进行写/擦除操作前，需要重新进行解锁操作。

注意：

- *进行写/擦除操作前，必须对 FLASH 进行解锁，否则将导致写/擦除操作无效；*
- *解锁过程必须严格按照解锁顺序，如果顺序错误，则会解锁失败，FLASH_CR 寄存器中的 LCK 标志位将不会清除，解锁无效；*
- *建议在完成目标操作后，对 FLASH 进行加锁，避免意外操作。*

FLASH 的读/写保护操作解锁

在对 FLASH 进行读/写保护修改前，需要写入特定的密钥进行操作解锁，避免意外的读/写保护产生。

- 解锁方式
  向 FLASH 密钥寄存器（FLASH_KR）中依次写入 KEY1 和 KEY2，成功解锁后，FLASH_KR 寄存器中的 PLK 位将会自动置 1，这时即可对 FLASH 进行读/写保护操作。
- 加锁方式
  向 FLASH 密钥寄存器（FLASH_KR 寄存器）的 PLK 位写 1，即可完成读/写保护操作的加锁。

注意：读/写保护解锁后即可对读/写保护进行修改，直至对 FLASH_KR 寄存器的 PLK 位完成重新加锁。因此建议在完成读/写保护的修改后，对 FLASH_KR 寄存器的 PLK 位写 1 进行加锁，避免意外操作。

FLASH 低功耗寄存器（FLASH_LPR）解锁

FLASH 支持低功耗模式，可以按需进入/退出低功耗模式，降低系统功耗。操作前需要对低功耗操作进行解锁，避免意外的操作。

- 解锁方式
  向 FLASH 低功耗寄存器的解锁标志位（FLASH_LPR.LCK）写 1 即可解锁操作。
- 加锁方式
  向 FLASH 低功耗寄存器的低功耗控制位（FLASH_LPR.LW/LS）写 1 发起进入/退出低功耗操作，操作完成建议对 FLASH_LPR.LCK 位写“1”执行加锁。

5.4.2.2 FLASH 擦除

FLASH 擦除操作可针对扇区擦除（Sector Erase）或单个 Bank 擦除（或称 Bank Erase）或整个闪存擦除（或称整片擦除，Chip Erase）执行。

扇区擦除（Sector Erase）

扇区擦除的具体步骤如下：

1. 将 FLASH_PAR 寄存器中的 EM 位置 0x0；

2. 将目标扇区号写入 FLASH_PAR 寄存器的 PA 位中，详细参考 FLASH_PAR.PA 位的描述；
3. 对 FLASH_KR 寄存器依次写入 KEY1/KEY2，完成操作解锁；
4. 将 FLASH_CR 寄存器中的 ES 位置 1，启动擦擦操作；
5. 待 FLASH_CR 寄存器中的 ES 位自动清零，即表示擦除操作完成。

注意：发起扇区擦除操作时，需填写正确的扇区号，不得超过 FLASH 的扇区范围，否则状态寄存器（FLASH_SR）中的 OES 位将置 1（如果使能中断，将导致触发错误中断）。

Bank 擦除（Bank Erase）

Bank 擦除的具体步骤如下：

1. 配置 FLASH_PAR 寄存器中的 EM 位，来选择正确的 Bank，详细参考 FLASH_PAR.EM 位的描述；
2. 对 FLASH_KR 寄存器依次写入 KEY1/KEY2，完成操作解锁；
3. 将 FLASH_CR 寄存器中的 ES 位置 1，启动擦擦操作；
4. 待 FLASH_CR 寄存器中的 ES 位自动清零，即表示擦除操作完成。

注意：发起 Bank 擦除操作时，需要 FLASH_OPDR 寄存器中的 BMD 位来确定 FLASH 的 Bank 模式，仅在双 Bank 模式下支持 Bank Erase 操作，其他模式下无效。

闪存擦除（Chip Erase）

要执行批量擦除，建议采用以下步骤：

1. 在 FLASH_PAR 寄存器中 EM 位置 0x3；
2. 对 FLASH_KR 寄存器依次写入 KEY1/KEY2，完成操作解锁；
3. 将 FLASH_CR 寄存器中的 ES 位置 1，启动擦擦操作；
4. 待 FLASH_CR 寄存器中的 ES 位自动清零，即表示擦除操作完成。

5.4.2.3 FLASH 编程

FLASH 的编程顺序如下：

1. 向 FLASH_PDRx（x = 0 …1）中写入需要编程的 64bit 数据；
2. 向 FLASH_PAR 寄存器中 PA 位写入需编程的起始地址；
3. 对 FLASH_KR 寄存器依次写入 KEY1/KEY2，完成操作解锁；
4. 将 FLASH_CR 寄存器中的 PS 位置 1，启动编程操作；
5. 待 FLASH_CR 寄存器中的 PS 位自动清零，即表示擦除操作完成。

提示：将 FLASH 单元从逻辑“1”变为逻辑“0”时，无需擦除操作，只需执行写操作。但把 FLASH 单元从逻辑“0”变为逻辑“1”时，则必须执行 FLASH 擦除操作。

注意：

- *如果同时发起擦除和编程操作请求时（即 FLASH_CR 寄存器中的 ES 和 PS 同时置位），该操作将被认为无效操作，不会执行任何操作，需重新发起对应操作；*
- *当发起擦除和编程操作请求后（即 FLASH_CR 寄存器中的 ES 或 PS 置位），必须等待*

FLASH_CR 寄存器中的 ES 或 PS 位自动清零后，才能对 FLASH 再次发起其他操作，否则会导致 FLASH 异常，状态寄存器（FLASH_SR）中的 OES 位将置 1（如果使能中断，将导致触发错误中断）；

- *FLASH 写操作位宽必须按照 64 位对齐，若发起操作的地址非对齐，该操作将被认为无效操作，不会执行任何操作，并且 FLASH_SR 中的 OES 位将置 1（如果使能中断，将导致触发错误中断）；*
- *FLASH 写操作必须处于有效的地址范围内，否则该操作将被认为无效操作，不会执行任何操作，并且 FLASH_SR 中的 OES 位将置 1（如果使能中断，将导致触发错误中断）。*

编程与缓存（Cache）

如果 FLASH 编程操作地址正好命中缓存 Cache 中的数据，FLASH 的写操作会同时修改 FLASH Cashe 存储中的数据，同时刷新缓存。

如果 FLASH 中的擦除操作也涉及数据或指令缓存（D-Cache/I-Cache）中的数据，则也会同样实现缓存更新操作，将缓存进行初始化，保证与 FLASH 存储数据一致。

5.4.3 FLASH 读保护（RDP）

对 FLASH 中的用户代码区域实施读保护机制，防止用户代码泄露，读保护分三个级别。

级别 0（RDP Level 0）：无读保护

将 0xAA 写入 FLASH_RPR 寄存器 RPLV 位（读保护寄存器）时，读保护级别即设为 0。此时，在所有自举配置（FLASH 用户自举、调试或从 RAM 自举）中，均可执行对 FLASH 的读取/编程操作（如果未设置其他保护）。

级别 1（RDP Level 1）：存储器读保护（调试功能受限）

将任意值（分别用于设置级别 0 和级别 2 的 0xAA 和 0xCC 除外）写入 FLASH_RPR 寄存器 RPLV 位（读保护寄存器）时，即激活读保护级别 1。设置读保护级别 1 后：

- 在连接调试功能或从 RAM 自举时，则不允许 FLASH 进行访问（读取、擦除、编程）。如果发起读操作请求，将导致总线错误（BusFault 异常）。
- 从 FLASH 自举时（未连接调试器），允许通过用户代码对 FLASH 进行访问（读取、擦除、编程）。
- 激活级别 1 后，如果修改读保护寄存器回退到级别 0（RDP 降级），则将对 FLASH 执行闪存擦除（在读保护降级为级别 0 前，进行擦除动作）。

注意：

- *如果通过代码执行闪存擦除（Chip Erase）操作，将仅擦除闪存区域，读/写保护配置将不受影响，闪存擦除动作之后，依然保持原来的设置。*

级别 2（RDP Level 2）：禁止调试/芯片读保护

将 0xCC 写入 FLASH_RPR 寄存器 RPLV 位（读保护寄存器）时，可激活读保护级别 2。设置

读保护级别 2 后：

- 级别 1 提供的所有保护均有效
- 调试口（SW）处于禁止状态，RAM 自举也将无法通过调试口载入代码
- 从 FLASH 自举时，允许用户代码对 FLASH 进行读取、擦除、编程操作
- 读保护级别 2 的设置不可逆，无法再降级回到级别 0 或级别 1

不同读保护级别下的访问限制

表 5-2　不同读保护级别下的访问限制

存储区	保护级别	从 FLASH 自举			从 SRAM 自举或连接调试器		
		读	写	擦除	读	写	擦除
用户 FLASH	级别 1	Y			N		
	级别 2	Y			N(1)		

(1) 级别 2 的读保护权限下，调试口将被禁止。

不同保护级别之间的转换方案

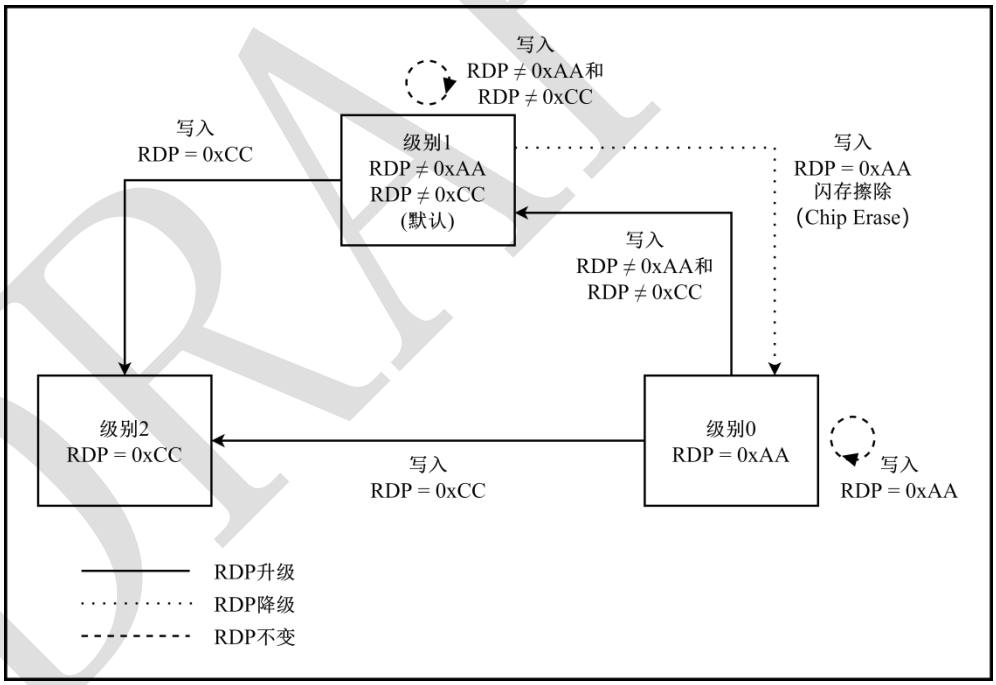


图 5-4　RDP 级别切换

图 5-4 为 FLASH 读保护级别切换的过程。

注意：

- *FLASH 读保护只能从级别 1 回退到级别 0，并导致闪存擦除（Chip Erase）；*
- *FLASH 读保护不能从级别 2 回退到级别 1 或级别 0；*
- *芯片出厂默认读保护为 Level0（无保护）状态，即 RDP=0xAA。*

5.4.4 FLASH 写保护（WRP）

FLASH 主区域 64 个用户扇区都具备写保护功能，用于避免 FLASH 的非易失性代码、数据被意外修改。写保护寄存器（FLASH_WPR）中每个 bit 分别对应 FLASH 2 个扇区的写保护功能，通过对该 FLASH_WPR 寄存器的 WRPC 写保护位的置 1/清 0，可以为相应的扇区开启/关闭写保护功能。

受到写保护的扇区，既不能对保护扇区进行扇区擦除、也不能编程。如果对 FLASH 中处于写保护状态的区域执行扇区擦除/编程操作，则 FLASH_SR 寄存器中的写保护错误标志位（FLASH_SR.WPS）将置 1。

注意：

- *如果执行闪存擦除（Chip Erase），一旦闪存中存在写保护（即 WPR 非全 1），擦除操作无效并 WPS 位置 1。*
- *如果执行闪存擦除（Chip Erase），将对整个闪存执行擦除，同时也会对写保护控制一并擦除。*
- *如果设置了读保护级别 1，并连接了调试器进行调试，即使指定扇区未设置写保护，也无法对相应的区域进行编程/擦除（包括扇区擦除和闪存擦除）操作。*
- *如果当前处于读保护级别 1，并执行读保护级别 1 退回到级别 0，即使设置了写保护，也同样发起闪存擦除操作，将整片 FLASH 擦除。*

写保护错误标志

如果对 FLASH 的写保护区域执行擦除/编程操作，则 FLASH_SR 寄存器中 WPS 位将置 1。

如果请求执行擦除操作，则以下情况下 WPS 位将置 1：

- 对写保护扇区执行扇区擦除
- 对在写保护状态下执行 Chip 擦除
- 对写保护扇区执行编程操作

5.4.4.1 FLASH 读/写保护配置

FLASH 的配置顺序如下：

1. 对 FLASH_KR 寄存器依次写入 KEY3/KEY4，完成操作解锁；
2. 将期望的读/写保护等级写入 FLASH_RPR/WPR 内，等待操作完成；

5.4.5 FLASH ECC 校验

如果 FLASH 存储数据受电子干扰或其他因素导致数据错误，会导致系统工作异常。为了提高系统稳定性及芯片可靠性，对 FLASH 加入了 ECC 校验功能，采用 8bit 纠 64bit 的 ECC 策略，可以实现 64bit 数据纠 1bit 检 2bit。

在对 FLASH 发起 Program 写操作时，ECC 功能模块会自动对 64bit 写数据进行 ECC 计算，生

成 ECC 校验码，将 Program 写数据与 ECC 校验码同时写入 FLASH 中；当进行读操作时，将数据和 ECC 校验码同时读回，进行 ECC 校验，如果存在 1bit 错误会自动将该 bit 纠正，同时产生 1bit ECC 错误（FLASH_SR 寄存器中的 SBC 位），并触发中断（如果 FLASH_CR 寄存器中 EIE 位使能，中断使能）；如果超过 1bit 错误会产生 2bit ECC 错误（FLASH_SR 寄存器中的 DBC 位），并触发 NMI 中断。

表 5-1　引发 NMI 中断的情况

中断事件	中断使能	中断标志	备注
多 ECC 错误	/	DBC	ECC 校验到超过 1Bit 错误

注意：内部 ECC 功能只能纠正 1bit 错误，超过 1bit 错误无法纠回。

5.4.6 FLASH 中断

FLASH 有下述几种情况将会引发中断，FLASH_SR 寄存器中相应的中断标志位将会置位，通过对相应的位写 1，将清除相应的中断标志位。

表 5-2　引发 FLASH 中断的情况

中断事件	中断使能	中断标志	备注
操作错误	EIE	OES	编程地址不对齐、编程/擦除地址超过范围、同时发起编程和擦除操作、在 FLASH Busy 状态下发起编程或擦除等操作、未解锁情况下发起编程/擦除操作
写保护错误	EIE	WPS	对 FLASH 存在写保护的区域执行擦除/编程操作或在写保护状态下执行 Chip 擦除
读保护错误	EIE	IOS	在读保护状态下，对 FLASH 进行擦除/编程操作；并在调试模式下，读操作还将产生 BusFault 异常中断；非法修改读保护等级(从读保护 Level1 降至 Level0)
单 ECC 错误	EIE	SBC	ECC 校验到只有 1Bit 错误

5.5 寄存器定义

5.5.1 寄存器列表

FLASH 基地址：

FLASH(0x40023000)

偏移	实例地址	名称	默认值	描述
0x00	FLASH 基地址+0x00	FLASH_CR	0x80000030	Flash 控制寄存器
0x04	FLASH 基地址+0x04	FLASH_LPR	0x00000000	Flash 低功耗寄存器
0x08	FLASH 基地址+0x08	FLASH_SR	0x00000000	Flash 状态寄存器
0x10	FLASH 基地址+0x10	FLASH_PDR0	0x00000000	Flash 数据寄存器 0
0x14	FLASH 基地址+0x14	FLASH_PDR1	0x00000000	Flash 数据寄存器 1
0x20	FLASH 基地址+0x20	FLASH_PAR	0x00000000	Flash 编程地址寄存器
0x24	FLASH 基地址+0x24	FLASH_KR	0x00000000	Flash 操作键寄存器
0x28	FLASH 基地址+0x28	FLASH_RPR	0x000000FF	Flash 读保护寄存器
0x2C	FLASH 基地址+0x2C	FLASH_WPR	0xFFFFFFFF	Flash 写保护寄存器
0x30	FLASH 基地址+0x30	FLASH_TR	0x0020CD16	Flash 时序寄存器
0x40	FLASH 基地址+0x40	FLASH_OPDR	0xFFFFF37F	Flash 数据寄存器

5.5.2 寄存器描述

5.5.2.1 Flash 控制寄存器（FLASH_CR）

- 名称：**Flash Control Register**
- 偏移地址：**0x00**
- 默认值：**0x80000030**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCK															
R/W															
0x1															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								EIE	FLE	DPE	IPE			ES	PS
								R/W	W	R/W	R/W			R/WAC	R/WAC
								0x0	0x0	0x1	0x1			0x0	0x0

字段	说明
[31] LCK	**Flash 解锁标志位(Flash Lock Flag)** 1: 写 1 加锁 0: 写 0 无效 解锁流程：参考 Key 寄存器描述； Note： 1) 对 Flash 完成解锁后，Flash 解锁标志 LCK 位将硬件置 0，表示解锁成功； 2) 当完成 Flash 的编程/擦除操作后，建议对 Flash 执行加锁操作，为了避免对 Flash 的误操作(对 LCK 位写 1 加锁，写 0 无效)
[7] EIE	**Flash 操作错误中断使能(Flash Operation Error Interrupt Enable)** 1: 使能 0: 关闭
[6] FLE	**Flash 保护更新使能(Flash Launch Enable)** 1: 使能 0: 关闭 Note: 按照 LCK 位的描述，先完成 Flash 的解锁，然后通过对 FLE 位写 1 来更新读保护级别。
[5] DPE	**Flash D 总线预取使能(Flash Dbus Prefetch Enable)** 1: 使能 0: 关闭
[4] IPE	**Flash I 总线预取使能(Flash Ibus Prefetch Enable)** 1: 使能 0: 关闭
[1] ES	**Flash 擦除启动位(Flash Erase Start)** 1: 擦除开始 0: 擦除完成 Note： 1) 写 1 开始擦除操作，擦除操作完成后硬件会自动清 0，写 0 无效； 2) 当发起擦除操作后，必须等待该 bit 位置 0 方可进行下一笔操作；
[0] PS	**Flash 编程启动位(Flash Program Start)** 1: 编程开始 0: 编程完成 Note： 1) 写 1 开始编程操作，编程操作完成后硬件会自动清 0，写 0 无效； 2) 当发起编程操作后，必须等待该 bit 位置 0 方可进行下一笔操作；

5.5.2.2 Flash 低功耗寄存器（FLASH_LPR）

- 名称：**Flash Lowpower Register**
- 偏移地址：**0x04**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCK															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SEL		LW	LS
												R/W		R/WAC	R/WAC
												0x0		0x0	0x0

字段	说明
[31] LCK	**Flash 低功耗模式解锁位(Flash Wakeup/Standby Unlock Bit)** 操作方式: 1.先对 LCK 位写 0x1 即可解锁; 2.然后发起 Wakeup/Standby 低功耗操作； Note：必须按照上述操作流程，否则操作无效，完成操作后，建议对 LCK 位写 0 完成加锁；
[3:2] SEL	**Flash 低功耗 BANK 选择(Flash Lowpower Bank Selection)** 00：Unselect 01：Bank0 10：Bank1 11：Bank0 和 Bank1 Note： Bit2 对应 Bank0，Bit3 对应 Bank1，可以选择单个 Bank 发起 Lowpower，或者两个 Bank 同时发起 Lowpower；
[1] LW	**Flash 低功耗唤醒使能(Flash Lowpower Wakeup)** 1: 使能 0: 关闭 Note： 1) 发起操作之前，需对 LCK 写 1 完成解锁，然后对 LW 位写 1 使能，写 0 无效； 2) 发起操作之后，必须等待 LS 位置 0 方可进行下一笔操作；
[0] LS	**Flash 低功耗进入使能(Flash Lowpower Start)** 1: 使能 0: 关闭 Note： 1) 发起操作之前，需对 LCK 写 1 完成解锁，然后对 LS 位写 1 使能，写 0 无效； 2) 发起操作之后，必须等待 LS 位置 0 方可进行下一笔操作； 3) 当 Flash 进入 Lowpower 模式后，Flash 不可以读写，无法执行取指操作；

5.5.2.3 Flash 状态寄存器（FLASH_SR）

- 名称：**Flash Status Register**
- 偏移地址：**0x08**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								BM	WM	DBC	SBC	BSY	IOS	WPS	OES
								R	R	R/W1C	R/W1C	R	R/W1C	R/W1C	R/W1C
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] BM	**Flash 工作 Bank 标志位(Flash BankMap Status)** 1: Bank1 0: Bank0 Note: 仅在双 Bank 下有效；
[6] WM	**Flash 工作模式标志位(Flash WorkMode Status)** 1: Normal Work Status，表示自检中各种校准参数正常； 0: Empty Status，表示自检中各种校准参数为空；
[5] DBC	**Flash 多个 Bit 错误标志位(Flash Double Bit Error Pending)** 1: Pending 0: No Pending Note:软件写 1 开清 0，写 0 无效；
[4] SBC	**Flash 单个 Bit 错误标志位(Flash Single Bit Error Pending)** 1: Pending 0: No Pending Note:软件写 1 开清 0，写 0 无效；
[3] BSY	**Flash 工作忙状态标志位(Flash In Busy Status)** 1: 处于 Busy 状态 0: 处于 Idle 状态 Note： 1. 该 Bit 位为只读； 2. 该 Bit 指示修改读/写保护及 Option Data 操作的状态；
[2] IOS	**Flash 非法操作错误状态(Flash Illegal Operation Error Status)** 1: 非法操作错误置起 0: 未发生非法操作错误 Note： 1. 该 Bit 位为写 1 清 0，写 0 无效； 2. 当 Flash 处于读保护状态下，对 Flash 执行擦/写操作，将产生非法操作错误； 3. 当非法修改读保护等级(从读保护 Level1 降至 Level0)
[1] WPS	**Flash 写保护错误状态(Flash Write Protect Error Status)** 1: 写保护操作错误置起 0: 未发生写保护操作错误 Note： 1. 该 Bit 位为写 1 清 0，写 0 无效； 2. 对 Flash 存在写保护的区域执行 Sector 擦/写操作或在写保护状态下执行 Chip 擦除，将产生非法操作错误；
[0] OES	**Flash 操作错误状态(Flash Operation Error Status)** 1: 操作错误置起 0: 未发生操作错误 Note： 1. 该 Bit 位为写 1 清 0，写 0 无效； 2. Flash 编程/擦除操作超出地址范围； 3. Flash 编程操作地址未按 64Bit 对齐；

4. Flash 同时发起擦除和编程操作；
5. Flash 处于 Busy 状态下发起其他操作；
6. Flash 在未解锁情况下，发起擦除和编程操作；

5.5.2.4 Flash 数据寄存器 0（FLASH_PDR0）

- 名称：**Flash Data Register 0**
- 偏移地址：**0x10**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD0															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD0															
R/W															
0x0															

字段	说明
[31:0] PD0	**Flash 编程数据(Flash Program Data)** Note：Flash 编程数据位宽为 64bit，必须按 64Bit 对齐进行 Program

5.5.2.5 Flash 数据寄存器 1（FLASH_PDR1）

- 名称：**Flash Data Register 1**
- 偏移地址：**0x14**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD1															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD1															
R/W															
0x0															

字段	说明
[31:0] PD1	**Flash 编程数据(Flash Program Data)** Note：Flash 编程数据位宽为 64bit，必须按 64Bit 对齐进行 Program

5.5.2.6 Flash 编程地址寄存器（FLASH_PAR）

- 名称：**Flash Address Register**
- 偏移地址：**0x20**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										EM			PA		
										R/W			R/W		
										0x0			0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA															
R/W															
0x0															

字段	说明
[21:20] EM	**Flash 擦写模式(Flash Erase Mode)** 11: Chip Erase 10: Bank Erase(Bank1) 01: Bank Erase(Bank0) 00: Sector Erase Note： 1）128KByte 模式下，允许 Sector 及 Chip 擦除，不支持 Bank 擦除； 2）256KByte+Single 模式下，允许 Sector 及 Chip 擦除，不支持 Bank 擦除； 3）256KByte+Double 模式下，允许 Sector 及 Bank0/1，Chip 擦除；
[18:0] PA	**Flash 编程/擦除地址(Flash Program/Erase Address)** 当发起 Program 编程操作： Main 区域地址:256K(0x0-0x3FFFF) Note: 1) Flash 编程地址，地址必须 64bit 对齐，否则报错退出； 2) Flash 一个 Sector 的大小为 4KByte，地址按照 0x1000 对齐； 当发起 Erase 擦除操作： 1）Sector 擦除：0x0-0x3F(地址为 Sector 的索引范围) 2）Chip 擦除：0x0

5.5.2.7 Flash 操作键寄存器（FLASH_KR）

- 名称：**Flash Key Register**
- 偏移地址：**0x24**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															PLK
															R/W
															0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY															
R/W															
0x0															

字段	说明
[16] PLK	**Flash 读写保护锁(Flash Read/Write Protect Lock)** 1: 写 1 加锁 0: 写 0 无效 解锁流程：参考 Key 寄存器描述；
[15:0] KEY	**Flash 秘钥寄存器(Flash Key Reigster)** Flash 包括以下几种解锁方式： 1.Flash 编程及擦除操作的解锁如下，当解锁完成后，CR 寄存器的 LOCK 位将置 1，可以通过对该位写 1 进行加锁： 1) 先向 KR 寄存器写 0x3fac; 2) 再向 KR 寄存器写 0x87e4； 2、Flash 读/写保护及 Option Data 操作解锁，当解锁完成后，PLK 位将置 1，可以通过对该位写 1 进行加锁： 1) 先向 KR 寄存器写 0x124A; 2) 再向 KR 寄存器写 0xbc7f;

5.5.2.8 Flash 读保护寄存器（FLASH_RPR）

- 名称：**Flash Read Protect Register**
- 偏移地址：**0x28**
- 默认值：**0x000000FF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RPLV							
								R/W							
								0xFF							

字段	说明
[7:0] RPLV	**Flash 读保护等级设置(Flash Read Protect Level)** Level0: 0xAA，此 Level 下无读保护功能，允许调试接口对 Flash 的读写及擦除操作； Level1: 其他值，此 Level 下有读保护功能，禁止调试接口对 Flash 的读写及擦除操作，保护级别可以回退； Level2: 0xCC，此 Level 下有读保护功能，禁用调试接口，保护级别不能回退； Note: 1.当 Flash 的保护等级由 Level1 回退到 Level0 时，硬件会自动擦除 Flash 中所有数据，用于保护用户代

码；
2.当 Flash 的保护等级设置为 level2 后，将无法回退，无法调试；

5.5.2.9 Flash 写保护寄存器（FLASH_WPR）

- **名称：Flash Write Protect Register**
- **偏移地址：0x14**
- **默认值：0xFFFFFFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPC															
R/W															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPC															
R/W															
0xFFFFFFFF															

字段	说明
[31:0] WRPC	**Flash 写保护设置(Flash Write Protect Level)** 1:存在写权限 0:关闭写权限 Note：Flash 写权限控制，每个 bit 对应 Flash 的 2 个 Sector 区域；

5.5.2.10 Flash 时序寄存器（FLASH_TR）

- **名称：Flash Timing Register**
- **偏移地址：0x30**
- **默认值：0x0020CD16**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										APG					
										R/W					
										0x20C					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APG				UNIT								RC			
R/W				R/W								R/W			
0x20C				0xD1								0x6			

字段	说明
[21:12] APG	**FLASH 编程 Timimg 设置(Flash Program Cycle Timing)** Note: 2.5us~3.5us(默认支持 175M~210M)
[11:4] UNIT	**FLASH Us 单位(Flash Us Unit Timing)** Note: 1us 的基准(默认支持 175M~210M)
[3:0] RC	**FLASH 读操作 Timing 设置(Flash Read Cycle Timing)** Note: Min-40ns

5.5.2.11 Flash 数据寄存器（FLASH_OPDR）

- 名称：**Flash Option Data Register**
- 偏移地址：**0x40**
- 默认值：**0xFFFFF37F**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAP				BMD				WWEN				IWEN			
R/W				R/W				R/W				R/W			
0xF				0xF				0xF				0xF			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBP						BORLV			BSEL			BLK			
R/W						R/W			R/W			R/W			
0xF						0x3			0x7			0xF			

字段	说明
[31:28] MAP	**FLASH 双 Bank Map 设置(FLASH BANK Mapping Config)** Note: 1) 0xA 表示 Bank1 Mapping，其他值为 Bank0 Mapping; 2) 当执行 Bank Mapping 操作之前，需将 SYSCTRL->SYSDCR.SEN 置 1，然后发起 Bank Map 操作，最后通过发起复位完成 Bank Map 并重新启动执行。 3)发起写操作后，需等待 SR->BSY 状态位清 0；
[27:24] BMD	**FLASH 双 Bank 模式配置(FLASH BANK Mode Config)** Note: 1) 0xA 表示双 Bank 模式，其他值表示单 Bank 模式; 2)发起写操作后，需等待 SR->BSY 状态位清 0；
[23:20] WWEN	**WWDG 默认使能位(WWDG Enable)** Note: 1) 0xA 表示开启，其他值表示关闭; 2)发起写操作后，需等待 SR->BSY 状态位清 0；
[19:16] IWEN	**IWDG 默认使能位(IWDG Enable)** Note: 1) 0xA 表示开启，其他值表示关闭; 2)发起写操作后，需等待 SR->BSY 状态位清 0；
[15:12] EBP	**FLASH 上电 ECC 错误配置(FLASH Power On ECC Bypass)** Note: 1) 0xA 表示非 Bypass，其他值表示 Bypass; 2) 上电启动时，当出现 1bit ECC 错误时，系统是否 Bypass 该 1bit 错误，当选择 Bypass 后芯片正常启动，否则将处于复位状态，默认为 Bypass 状态； 3)发起写操作后，需等待 SR->BSY 状态位清 0；
[9:8] BORLV	**BOR 电压阈值设置(BOR Voltage Limit Config)** 0: 2.4V 1: 2.55V 2: 2.7V 3: 2.85V Note: 1)发起写操作后，需等待 SR->BSY 状态位清 0；
[6:4] BSEL	**BOOT 选择配置位(Boot Selection)** Note: 1) Bit4-6 分别对应“4.5 自举配置”中自举模式表的 Sel0-2, 默认为硬件引脚方式(FLASH BOOT)，详细参考“自举模式”表的描述。 2)发起写操作后，需等待 SR->BSY 状态位清 0；
[3:0] BLK	**BOOT 锁定配置位(Boot Lock Config)** Note: 1) 0xA 为表示 Lock，其他值表示未 Lock; 2)发起写操作后，需等待 SR->BSY 状态位清 0；

6 电源控制器（PWR）

6.1 电源

芯片的工作电压（AVCC/VCC）要求介于 3.0V 到 3.6V 之间（当 AVCC 3.0V 供电时，会影响 ADC 的性能），通过内部 LDO（线性调压器）提供 1.2V 电压给数字逻辑电压域供电。

6.1.1 LDO（线性调压器）

嵌入式线性调压器为所有数字电路供电，调压器输出电压约为 1.2V。

此调压器需要将两个外部电容连接到专用引脚 VDD 和 VSS 之间，所有封装都配有 VDD 引脚，具体引脚与封装有关。

调压器在复位后始终处于使能状态，根据应用模式的不同，可选择以下两种不同的工作模式。

- 运行模式，调压器为 1.2 V 域（内核、存储器和数字外设）提供全功率，在此模式下，可通过软件配置 SYSCTRL_PWRCR.VDDSET 位将调压器的输出电压调整为 1.2V。
- 待机模式，调压器为 1.2 V 域（内核、存储器和数字外设）提供低功率，可以保留寄存器和内部 SRAM 中的内容。在此模式下，通过软件配置 SYSCTRL_PWRCR.VDDSET 位将调压器的输出电压配置调整为 0.9 V。

注意：通过软件配置 SYSCTRL_PWRCR.VDDSET 位时，注意 VDD 低电压检测（LVD）的相关配置，避免由于启动了 VDD 的低电压检测导致芯片误入复位状态。

6.2 电源监控器

6.2.1 上电复位（POR）/掉电复位（PDR）

本芯片内部集成有 POR/PDR 电路，可以从 2.8V 开始正常工作。

当 VCC/AVCC 低于指定阈值 $V_{POR/PDR}$ 时，器件无需外部复位电路便会保持复位状态。

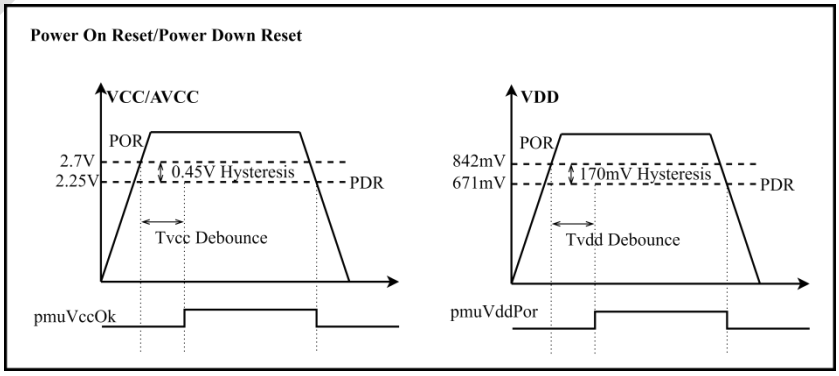


图 6-1 上电复位/掉电复位波形

6.2.2 可编程电压检测器（LVD）

可以使用 LVD 单元监视 VCC/VDD 的电源电压，对 VCC/VDD 的电压和过流进行实时监测，并将实时监测值与低电压检测模块中设置的阈值（SYSCTRL_PLCR.xxLVS）进行比较。

芯片内 VCC/VDD 的电压及过流监测功能可通过软件配置 LVD 模块中的模拟控制寄存器（SYSCTRL_PLCR.xxLVE）来使能对应的功能。

在 LVD 模块中的状态控制寄存器（SYSCTRL_PSR）提供了 VCC/VDD 欠压及过流的标志，用于指示电源电压是否小于用户设置的电压阈值或者电源电流是否大于用户设置的电流阈值。该事件标志会生成中断，并内部直接连接到 NMI 中断，如果中断使能，则可以产生不可屏蔽中断；该事件标志内部也会连接到系统复位，如果复位使能，则可以产生系统复位；该事件标志内部也会连接到事件系统，该功能的用处是可以在异常情况下执行紧急关闭的任务。

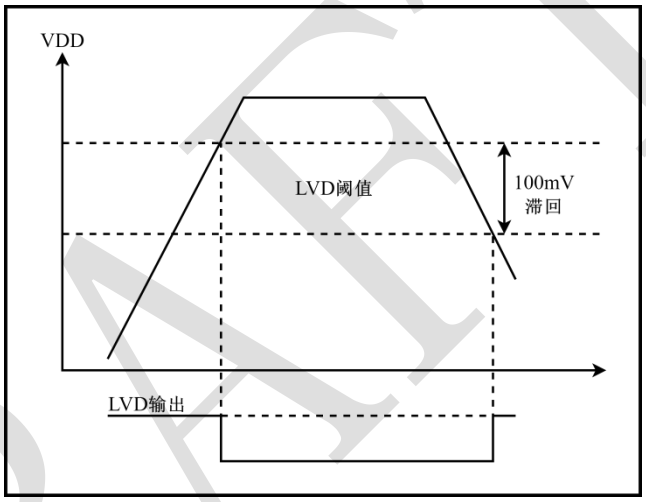


图 6-2 LVD 阈值

6.3 低功耗模式

默认情况下，系统复位或上电复位后，微控制器进入运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。系统提供了多个低功耗模式，可在 CPU 不需要运行时（例如等待外部事件时）节省功耗。由用户根据应用选择具体的低功耗模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

芯片有两个低功耗模式：

- 睡眠模式（Cortex™-M4 内核停止，外设保持运行）
- 停止模式（除 LSI 时钟外，所有时钟都停止）

此外，可通过下列方法之一降低运行模式的功耗：

- 降低系统时钟速度
- 不使用 APB 和 AHB 外设时，将对应的外设时钟关闭

表 6-1 低功耗模式汇总

模式名称	进入	唤醒	对 VDD 域时钟的影响	调压器
睡眠（立即休眠或退出时休眠）	WFI	任意中断	CPU CLK 关闭对其它时钟或模拟时钟源无影响	开启
	WFE	唤醒事件		
停止	SLEEPDEEP 位+WFI 或 WFE	任意中断	除 LSI 时钟外，所有时钟都停止	开启

6.3.1 降低系统时钟速度

在运行模式下，可通过切换系统时钟源或者对系统时钟的预分频寄存器进行编程来降低系统时钟（SYSCLK、HCLK0、HCLK1、PCLK0 和 PCLK1）速度。

在进入睡眠模式之前，也可以使用这些预分频器来降低外设运行速度。

有关详细信息，请参见 RCU 总线时钟控制寄存器（RCU_CCR）。

6.3.2 外设时钟门控

在运行模式下，可随时停止各外设和存储器的 HCLK 和 PCLK 以降低功耗。要进一步降低睡眠模式的功耗，可在执行 WFI 或 WFE 指令之前禁止外设时钟。

AHB0 外设时钟门控由 AHB0 时钟控制寄存器（RCU_AHB0ENR）。参见第 7.3.2.16 节：RCU AHB0 时钟控制寄存器（RCU_AHB0ENR）。

AHB1 外设时钟门控由 AHB1 时钟控制寄存器（RCU_AHB1ENR）。参见第 7.3.2.17 节：RCU AHB1 时钟控制寄存器（RCU_AHB1ENR）。

APB0 外设时钟门控由 APB0 时钟控制寄存器（RCU_APB0ENR）。参见第 7.3.2.14 节：RCU APB0 时钟控制寄存器（RCU_APB0ENR）。

APB1 外设时钟门控由 APB1 时钟控制寄存器（RCU_AHB1ENR）。参见第 7.3.2.15 节：RCU APB1 时钟控制寄存器（RCU_APB1ENR）。

6.3.3 睡眠模式

进入睡眠模式

执行 WFI（等待中断）或 WFE（等待事件）指令即可进入睡眠模式。根据 Cortex™-M4 系统控制寄存器中 SLEEPONEXIT 位的设置，可以通过两种方案选择睡眠模式进入机制：

- 立即休眠：如果 SLEEPONEXIT 位清零，MCU 将在执行 WFI 或 WFE 指令时立即进入睡眠模式。
- 退出时休眠：如果 SLEEPONEXIT 位置 1，MCU 将在退出优先级最低的 ISR 时立即进入睡眠模式。

有关如何进入睡眠模式的详细信息，请参见表 6-2 和表 6-3。

退出睡眠模式

如果使用 WFI 指令进入睡眠模式，则嵌套向量中断控制器（NVIC）确认的任意外设中断都会将器件从睡眠模式唤醒。

如果使用 WFE 指令进入睡眠模式，CPU 将在有事件发生时立即退出睡眠模式。唤醒事件可通过以下方式产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时使能 Cortex™-M4 系统控制寄存器中的 SEVONPEND 位。当 CPU 从 WFE 恢复时，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部中断为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

由于没有在进入/退出中断时浪费时间，此模式下的唤醒时间最短。

有关如何退出睡眠模式的详细信息，请参见表 6-2。

表 6-2 进入和退出休眠模式

休眠模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： ➢ SLEEPDEEP = 0 及 ➢ SLEEPONEXIT = 0 请参见 Cortex™-M4 系统控制寄存器。
退出模式	如果使用 WFI 进入： 中断：请参见表 11-1（芯片的中断向量表） 如果使用 WFE 进入： 唤醒事件：请参见退出睡眠模式中的唤醒事件产生的描述。

6.3.4 停止模式

停止模式基于 Cortex™-M4 深度睡眠模式与外设时钟门控。在停止模式下，数字电压 1.2 V 域中除了 LSI（32K）时钟外的所有时钟都会停止，PLL、HSI 和 HSE RC 振荡器也被关闭，内部 SRAM 和寄存器内容将保留。

进入停止模式

有关如何进入停止模式的详细信息，请参见表 6-3。

要进一步降低停止模式的功耗，可将内部调压器设置为低功耗模式。通过软件配置 SYSCTRL_PWRCR.VDDSET 位来将 VDD 的电压设置为 0.9 V

在停止模式下，ADC 或 DAC 也会产生功耗，在进入停止模式前需将其关闭。

退出停止模式

有关如何退出停止模式的详细信息，请参见表 6-3。

通过发出中断或唤醒事件退出停止模式时，将选择 LSI RC 振荡器作为系统时钟。

在停止模式下，调压器处于低功耗模式下工作，当从停止模式唤醒，需要将调压器设置为全功率的运行模式。

表6-3 进入和退出停止模式

停止模式	说明
进入模式	WFI（等待中断）或WFE（等待事件），且： ➢ 将Cortex™-M4系统控制寄存器中的SLEEPDEEP位置1 ➢ 将电源控制寄存器SYSCTRL_PWRCR中的VDD_SET位清零 注意：要进入停止模式之前，所有中断需处于挂起状态，否则将忽略进入停止模式这一过程，继续执行程序。
退出模式	如果使用WFI进入： 所有配置为中断模式的中断，必须在NVIC中使能对应的中断向量，请参见表11-1：芯片的中断向量表。 如果使用WFE进入： 唤醒事件：请参见退出睡眠模式中的唤醒事件产生的描述。

调试模式

默认情况下，如果使用调试功能时应用程序将 CPU 置于停止模式，调试连接将中断。这是因为 Cortex™-M4 内核时钟停止工作。

7 复位和时钟控制（RCU）

7.1 复位

共有二种类型的复位，分别为系统复位、电源复位。

7.1.1 系统复位

除了 SYSCTRL 模块中复位标志位、FLASH 自检参数外，系统复位将复位所有寄存器至复位状态，当发生以下任一事件时，将产生一个系统复位：

- NRST 引脚上的低电平（外部复位）
- 独立看门狗计数完成（IWDG 复位）
- 窗口看门狗计数完成（WWDG 复位）
- 软件复位

可通过查看 RCU 模块中的复位状态寄存器（RCU_SRSTSR）来识别复位事件的来源。若要对芯片进行软件复位，必须将 Cortex™-M4 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1。有关详细信息，请参见 Cortex™-M4 技术参考手册。

7.1.1.1 引脚复位

芯片具有一个独立的复位引脚 NRST，默认上拉，拉低该引脚将引起系统复位。

7.1.1.2 独立看门狗复位

详见独立看门狗模块介绍。

7.1.1.3 窗口看门狗复位

详见窗口看门狗模块介绍。

7.1.1.4 软件复位

通过将 Cortex™-M4 内核的“应用中断与复位控制寄存器”中的 SYSRESETREQ 位置 1，可实现软件复位（内核和外设）；该寄存器中的另一个控制位 VECTRESET 则只复位 Cortex™-M4 内核，不复位外设。

7.1.2 电源复位

以下事件之一发生时，将产生电源复位：

- 上电复位（POR 复位）
- 低电压复位（LVD 复位）

电源复位将复位所有寄存器。

7.1.2.1 上电复位

芯片内部有一个完整的上电复位（POR）电路，当供电电压达到 V_{POR} 时系统即能正常工作。当 VDD 低于指定的限位电压 V_{POR} 时，系统保持为复位状态。

7.1.2.2 低电压复位

低电压复位是一种强制性保护复位，用户可以通过配置相关的低电压监测使能位开启，当供电电压低于检测阈值时，将产生复位。

7.2 时钟

可以使用四种不同的时钟源来驱动系统时钟（SYSCLK）：

- LSI 振荡器时钟
- HSE 振荡器时钟
- PLL 时钟
- HSI 振荡器时钟

器件具有以下时钟源：

- 32KHz 低速内部 RC 振荡器（LSI），该 RC 振荡器用于驱动独立看门狗

对于每个时钟源来说，在未使用时都可单独打开或者关闭（LSI 除外），以降低功耗。

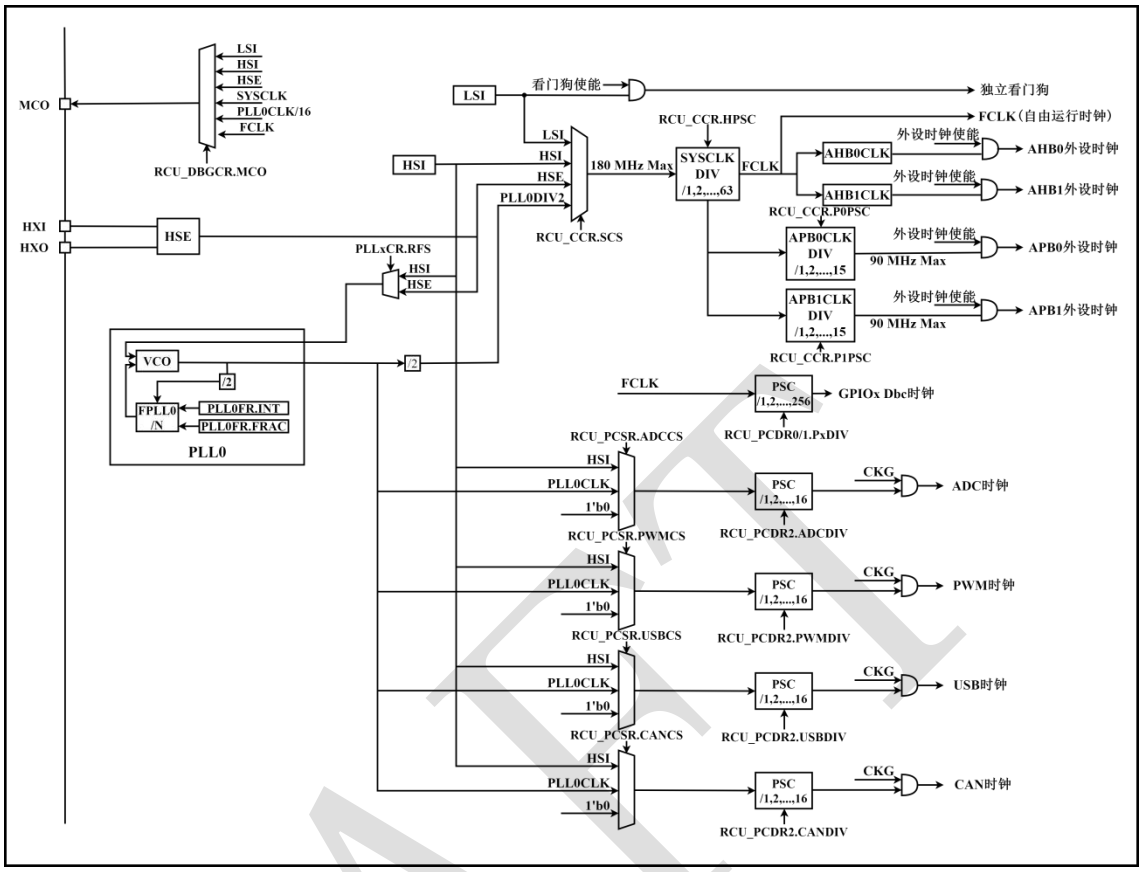


图 7-1　时钟电路简图

为了高度的灵活性，用户可选择使用外部晶振方案或者使用内部 RC 振荡器的无晶振方案，通过 PLL 的合理配置，既可满足系统的最高时钟频率，也可为 USB 以及 HRPWM、ADC 等需要特定时钟的外设保证合适的频率。

可通过多个预分频器配置 AHB0/1 频率、APB0/1 频率，AHB 域最高时钟频率为 180MHz，APB 域的最高时钟频率为 90MHz。

除以下时钟外，所有外设时钟均由系统时钟（SYSCLK）提供：

- 来自于特定 PLL0 输出的 USB 时钟（60MHz）
- 来自于特定 PLL0 输出的 CAN 时钟（120MHz）
- 来自于特定 PLL0 输出的 ADC 时钟（60MHz）
- 来自于特定 PLL0 输出的 HRPWM 时钟（180MHz）

RCU 送出一个 8 分频的 AHB 时钟（HCLK）到 Cortex 系统定时器（SysTick），SysTick 可使用此时钟作为时钟源，也可使用 HCLK 作为时钟源，具体可在 SysTick 控制和状态寄存器中配置。

FCLK 充当 Cortex™-M4 的自由运行时钟。有关详细信息，请参见 Cortex™-M4 技术参考手册。

7.2.1　HSE 晶振

外部晶振谐振器和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选振荡器的不同做适当调整。

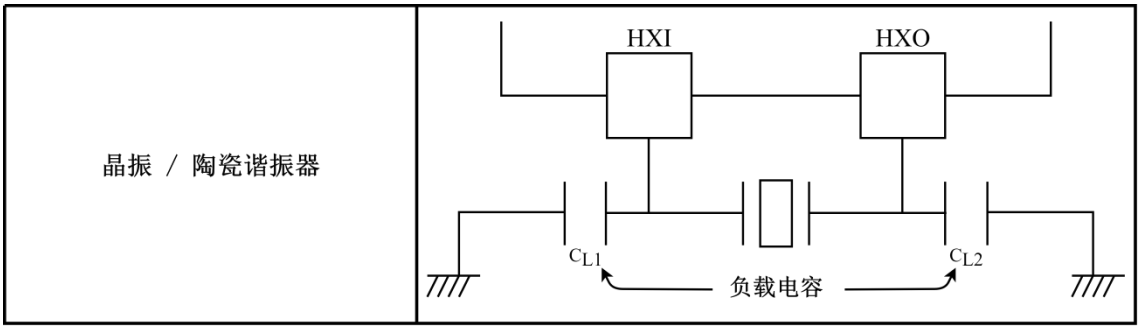


图 7-2 HSE 时钟源

外部晶振/陶瓷谐振器

HSE 的特点是精度非常高。

相关的硬件配置如上图所示。

在芯片上电完成后，晶振起振电路使能默认为 1，起振电路可以直接使用。

晶振起振电路可通过晶振控制寄存器 RCU_ XOSCCR 中的 XEN 位打开或关闭。

7.2.2 HSI 振荡器

HSI 时钟信号由内部 8 MHz RC 振荡器生成，可直接用作系统时钟，或者用作 PLL 输入。

HSI RC 振荡器的优点是成本较低（无需使用外部组件）。此外，其启动速度也要比 HSE 晶振快，但即使校准后，其精度也不及外部晶振或陶瓷谐振器。

当 HSI 作为 PLL 参考时钟源时，可以通过从 Flash 中读取 HSI 实际频率来配置分频系数，以得到准确的 PLL 输出时钟频率。

HSI RC 振荡器可通过晶振控制寄存器 RCU_ XOSCCR 中的 HEN 位打开或关闭。

HSI 信号还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见第 7.2.6 节：时钟安全系统（CSS）。

7.2.3 PLL 设置

- 由 HSE 或 HSI 振荡器提供参考时钟，并具有以下输出时钟：
  - 第一个输出用于生成高速系统时钟（最高达 180MHz）
  - 第二个输出用于生成 FLASH 工作时钟
- 用于生成 180M 时钟用于 HRPWM 模块，60M 时钟用于 USB 及 ADC 模块

由于在 PLL 使能后，PLL 配置参数不建议动态修改，所以要求先完成 PLL 参数的配置，然后再使能；如需要在 PLL 使能后调整其配置参数，需要先将作用的时钟源从 PLL 切换到一个稳定的时钟源上（例如 HSI），然后关闭 PLL，完成 PLL 参数的修改后，再使能 PLL。

PLL 的配置介绍：

- PLL 的参考时钟源
  可通过软件配置 RCU 模块中的 PLLx 控制寄存器（RCU_PLLxCR）中的 RCS 位来选择合适的 PLL 参考时钟。
  - 00：HSE
  - 01：HSI

- PLL 的分频模块：包含小数和整数部分
  1. PLL 内部 VCO 输出时钟经过硬件 2 分频后，送到 FPLLx 反馈分频模块，则输入到反馈分频模块 FPLLx 的频率为 PLLxCLK 输出频率除以 2。
  2. 输入时钟经过反馈分频模块（FPLLx）后，输出时钟应该等于 PLL 的参考输入时钟的频率，即反馈分频模块（FPLLx）的输入时钟频率/输出时钟频率（等于 PLL 的输入参考频率），按照上述公式可得到 PLL 的分频值。根据计算出来的分频值，将整数部分存至 RCU_PLLxFR.INT 内，小数部分乘以 2^{16} 后再取整存入 RCU_PLLxFR.FRAC 内。

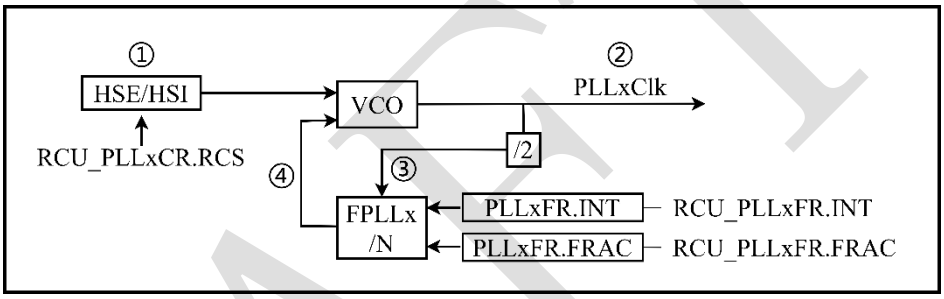


图 7-3　PLL 的配置流程

- HSE 晶振频率范围支持 6~26MHz，推荐晶振频率为 8MHz；
- 图中 PLLxClk 时钟频率 = 晶振频率*N*2；

如将 HSE 或 PLL（由 HSE 提供时钟信号）用作系统时钟，则在 HSE 发生故障时，PLL 也将由硬件禁止，可自由配置选择切换方案：第一种可以直接切换系统时钟至内部 HSI；第二种可以切换 PLL 的参考时钟至 HSI。

7.2.4　LSI 时钟

LSI RC 振荡器可作为低功耗时钟源在停机和待机模式下保持运行，供独立看门狗（IWDG）和自动唤醒单元使用，时钟频率在 32 kHz 左右，LSI RC 低频振荡器不可关闭。

7.2.5　系统时钟（SYSCLK）选择

在系统复位后，默认系统时钟为 HSI，若需要切换时钟，需要在目标时钟源已准备就绪（例如 PLL 时钟已锁定），才可从一个时钟源切换到另一个时钟源。

7.2.6　时钟安全系统（CSS）

为了增强系统的可靠性，防止因时钟失效造成系统出错甚至死机的严重后果，芯片内部增加了一个时钟安全监控模块（CSS）。CSS 监控模块用于监测 HSE 的振荡情况，包含晶振、陶

振、或外部时钟输入的情况，一旦发生停振或振荡异常（频率低于正常值或高于正常值），则发送异常标志，该异常标志可作为中断输入或 PWM 异常事件信号输入。

用户可通过配置 RCU_CSSCR 时钟安全控制寄存器中的 SWE 位来使能 CSS 模块，当发生异常时，用户可以选择系统时钟或 PLL 参考时钟是否自动切换到内部 HSI。

注意：一旦 CSS 中断产生，引起 NMI 被不断执行，直到 CSS 中断挂起位被清除。如果直接或间接使用 HSE 振荡器作为系统时钟（间接是指该振荡器用作 PLL 的参考时钟，并且该 PLL 时钟为系统时钟）并且检出故障，用户可选择将系统时钟或 PLL 参考时钟切换到 HSI 振荡器。

如果 HSE 振荡器时钟是充当系统时钟的 PLL 的参考时钟，则在发生故障时，用户可选择将 PLL 的参考时钟切换到 HSI 振荡器，由于内部 HSI 时钟偏差会导致 PLL 时钟输出频率出现稍微偏差；用户可以启动 PLL 的自动加载功能，通过使能 RCU_PLLxCR 寄存器中的 ARE 位来启动 PLL 的自动加载功能，则在发生故障时，PLL 的参考时钟切换到 HSI 振荡器的同时，也会自动加载 HSI TRIM 后的分频系数。

7.2.7 看门狗时钟

如果独立看门狗（IWDG）通过软件设置的方式启动，则 LSI 振荡器将为独立看门狗（IWDG）提供运行计数时钟，并且 LSI 振荡器不可关闭。

7.2.8 时钟输出功能

芯片时钟输出（MCO）引脚：
用户可通过可配置的预分配器（从 1 到 5）向 MCO 引脚（PA8）输出四个不同的时钟源：

- LSI
- HSI
- HSE
- SYSCLK
- PLL0/16
- FCLK
- ATECLK

所需的时钟源可通过 RCU_DBGCR 寄存器中 MCO 位来选择，通过置位 RCU_DBGCR 寄存器中 MCOEN 位来使能时钟 FanOut 的功能。

对于 MCO 引脚，必须将相应的 GPIO 端口（PA8）设置为 AF2 复用功能模式（MCO），才能使用 MCO 时钟输出。

MCO 输出时钟建议不超过 50 MHz（最大 I/O 速度），具体速率与 I/O 负载电容、I/O 驱动能力以及 Slew Rate 配置相关。

7.3 寄存器定义

7.3.1 寄存器列表

RCU 基地址：

RCU(0x40020000)

偏移	实例地址	名称	默认值	描述
0x00	RCU 基地址+0x00	RCU_PLL0CR	0x00000300	RCU PLL0 控制寄存器
0x04	RCU 基地址+0x04	RCU_PLL0FR	0x00190000	RCU PLL0 分频寄存器
0x30	RCU 基地址+0x30	RCU_CCR	0x00000100	RCU 总线时钟控制寄存器
0x38	RCU 基地址+0x38	RCU_PCSR	0x00000000	RCU 功能时钟源寄存器
0x40	RCU 基地址+0x40	RCU_PCDR0	0x00000000	RCU 功能时钟分频寄存器 0
0x44	RCU 基地址+0x44	RCU_PCDR1	0x00000000	RCU 功能时钟分频寄存器 1
0x48	RCU 基地址+0x48	RCU_PCDR2	0x00002551	RCU 功能时钟分频寄存器 2
0x4C	RCU 基地址+0x4C	RCU_PCENR	0x00000000	RCU 功能时钟控制寄存器
0x50	RCU 基地址+0x50	RCU_APB0ENR	0x00000000	RCU APB0 时钟控制寄存器
0x54	RCU 基地址+0x54	RCU_APB1ENR	0x00000000	RCU APB1 时钟控制寄存器
0x58	RCU 基地址+0x58	RCU_AHB0ENR	0x000000FE	RCU AHB0 时钟控制寄存器
0x5C	RCU 基地址+0x5C	RCU_AHB1ENR	0x00000000	RCU AHB1 时钟控制寄存器
0x60	RCU 基地址+0x60	RCU_APB0RSTR	0x000000FF	RCU APB0 复位控制寄存器
0x64	RCU 基地址+0x64	RCU_APB1RSTR	0x00003FFF	RCU APB1 复位控制寄存器
0x68	RCU 基地址+0x68	RCU_AHB0RSTR	0x00001FFF	RCU AHB0 复位控制寄存器
0x6C	RCU 基地址+0x6C	RCU_AHB1RSTR	0x00003FFF	RCU AHB1 复位控制寄存器
0x70	RCU 基地址+0x70	RCU_XOSCCR	0x00000016	RCU 晶振控制寄存器
0x74	RCU 基地址+0x74	RCU_CSSCR	0x00000000	RCU 时钟安全控制寄存器
0x78	RCU 基地址+0x78	RCU_DBGCR	0x00000000	RCU 内部时钟 FanOut
0x7C	RCU 基地址+0x7C	RCU_SRSTSR	0x00000100	RCU 系统复位状态寄存器
0x80	RCU 基地址+0x80	RCU_KEYR	0x00000000	RCU 键寄存器

7.3.2 寄存器描述

7.3.2.1 RCU PLL0 控制寄存器（RCU_PLL0CR）

- 名称：**PLL0 Control Register**
- 偏移地址：**0x00**
- 默认值：**0x00000300**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	LKF														
R/W	R														
0x0	0x0														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RFD	LPF	BDS						UG	ARE	RCS	
				R/W	R/W	R/W						W	R/W	R/W	
				0x0	0x0	0x3						0x0	0x0	0x0	

字段	说明
[31] EN	**PLL0 Enable 使能控制(PLL0 Enable)** 0:关闭 1:启动
[30] LKF	**PLL0 锁定标志(PLL0 Lock Flag)** 0:未锁定 1:已锁定
[11] RFD	**PLL0 参考前除频(PLL0 Refer Clock Division)** 0:不分频 1: 2 分频
[10] LPF	**PLL0 LPF 配置** 1: 参考时钟高于 12M 时配 1 0: 参考时钟低于 12M 时配 0
[9:8] BDS	**PLL0 VCO 振荡频率范围控制(PLL0 Band Selection)** 00: 最小频率 11: 最大频率 Note：具体 Band 值参考出厂校准配置；
[3] UG	**PLL0 倍频系数更新事件(PLL0 Update Generate)** 1: 生成更新事件 0: 无影响
[2] ARE	**PLL0 自动加载使能(PLL0 AutoReload Enable)** 1: 使能 0: 关闭
[1:0] RCS	**PLL0 参考时钟源(PLL0 Refer Clock Selection)** 00: HSE 01: HSI 10: 关闭 11: 外部时钟

7.3.2.2 RCU PLL0 分频寄存器（RCU_PLL0FR）

- 名称：**PLL0 Fractional Register**
- 偏移地址：**0x04**
- 默认值：**0x00190000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		INT													
		R/W													
		0x19													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAC															
R/W															
0x0															

字段	说明
[29:16] INT	**PLL0 倍频系数整数部分(PLL0 Integer part of multiplication factor)**
[15:0] FRAC	**PLL0 倍频系数小数部分(PLL0 Fractional part of multiplication factor)** Note: PLLx 输出频率经过硬件二分频得到 PLLxDiv2 时钟， PLLxDiv2 时钟经过(Int+Frac)分频后得到 PLLx 的 FeedBack 时钟，配置需确保计算出来的 FeedBack 时钟与参考时钟相等； 例如: 如果 PLLx 的参考时钟为 8M，需要 PLLx 的输出频率为 160M，按照上述流程，PLLxDiv2 时钟频率为 80M，然后用 80M 除以(Int+Frac)需要等于参考时钟 8M，即可得到除频系数为 Int=10，Frac=0；

7.3.2.3 RCU 总线时钟控制寄存器（RCU_CCR）

- 名称：**System Clock Config Register**
- 偏移地址：**0x30**
- 默认值：**0x00000100**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P1PSC				P0PSC				HPSC						SCS	
R/W				R/W				R/W						R/W	
0x0				0x1				0x0						0x0	

字段	说明
[15:12] P1PSC	**APB1 时钟分频配置(APB1 Clock Prescale)** 0: 1 分频(不分频) 1: 2 分频 … N: N+1 分频
[11:8] P0PSC	**APB0 时钟分频配置(APB0 Clock Prescale)** 1: 2 分频 … N: N+1 分频 Note: 最低 2 分频;
[7:2] HPSC	**AHB 时钟分频配置(AHB Clock Prescale)** 0: 1 分频(不分频) 1: 2 分频

	… N: N+1 分频
[1:0] SCS	**系统时钟源选择(System Clock Selection)** 00: RC8M 01: RC32K 10: PLL0/N 11: XOSC

7.3.2.4 RCU 功能时钟源寄存器（RCU_PCSR）

- **名称：Peripheral Function Clock Source Register**
- **偏移地址：0x38**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CANCS		USBCS		ADCCS		PWMCS	
								R/W		R/W		R/W		R/W	
								0x0		0x0		0x0		0x0	

字段	说明
[7:6] CANCS	**CAN 时钟源选择(CAN Clock Source Selection)** 00: RC8M 01: PLL0 1x: Reserved
[5:4] USBCS	**USB 时钟源选择(USB Clock Source Selection)** 00: RC8M 01: PLL0 1x: Reserved
[3:2] ADCCS	**ADC 时钟源选择(ADC Clock Source Selection)** 00: RC8M 01: PLL0 1x: Reserved
[1:0] PWMCS	**PWM 时钟源选择(PWM Clock Source Selection)** 00: RC8M 01: PLL0 1x: Reserved

7.3.2.5 RCU 功能时钟分频寄存器 0（RCU_PCDR0）

- 名称：**Peripheral Function Clock Division Register0**
- 偏移地址：**0x40**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDDIV								PCDIV							
R/W								R/W							
0x0								0x0							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBDIV								PADIV							
R/W								R/W							
0x0								0x0							

字段	说明
[31:24] PDDIV	**GPIOD 消抖时钟分频(GPIOD Debounce Clock Division)** 0: 1 分频(不分频) 1: 2 分频 … N: N+1 分频
[23:16] PCDIV	**GPIOC 消抖时钟分频(GPIOC Debounce Clock Division)** 0: 1 分频(不分频) 1: 2 分频 … N: N+1 分频
[15:8] PBDIV	**GPIOB 消抖时钟分频(GPIOB Debounce Clock Division)** 0: 1 分频(不分频) 1: 2 分频 … N: N+1 分频
[7:0] PADIV	**GPIOA 消抖时钟分频(GPIOA Debounce Clock Division)** 0: 1 分频(不分频) 1: 2 分频 … N: N+1 分频

7.3.2.6 RCU 功能时钟分频寄存器 1（RCU_PCDR1）

- 名称：**Peripheral Function Clock Division Register1**
- 偏移地址：**0x44**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFDIV								PEDIV							
R/W								R/W							
0x0								0x0							

字段	说明
[15:8] PFDIV	**GPIOF 消抖时钟分频(GPIOF Debounce Clock Division)** 0: 1 分频(不分频)

	1: 2 分频 … N: N+1 分频
[7:0] PEDIV	**GPIOE 消抖时钟分频(GPIOE Debounce Clock Division)** 0: 1 分频(不分频) 1: 2 分频 … N: N+1 分频

7.3.2.7 RCU 功能时钟分频寄存器 2（RCU_PCDR2）

- **名称：Peripheral Function Clock Division Register2**
- **偏移地址：0x48**
- **默认值：0x00002551**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CANDIV				USBDIV				ADCDIV				PWMDIV			
R/W				R/W				R/W				R/W			
0x2				0x5				0x5				0x1			

字段	说明
[15:12] CANDIV	**CAN 时钟分频选择(CAN Clock Division)** 0: 不分频 1: 2 分频 … N: N+1 分频
[11:8] USBDIV	**USB 时钟分频选择(USB Clock Division)** 0: 不分频 1: 2 分频 … N: N+1 分频
[7:4] ADCDIV	**ADC 时钟分频选择(ADC Clock Division)** 0: 不分频 1: 2 分频 … N: N+1 分频
[3:0] PWMDIV	**PWM 时钟分频选择(PWM Clock Division)** 0: 不分频 1: 2 分频 … N: N+1 分频

7.3.2.8 RCU 功能时钟控制寄存器（RCU_PCENR）

- 名称：Peripheral Function Clock Enable Register
- 偏移地址：0x4C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAN1 FEN	CAN0 FEN	USBFEN	PWM7 FEN	PWM6 FEN	PWM5 FEN	PWM4 FEN	PWM3 FEN	PWM2 FEN	PWM1 FEN	PWM0 FEN	ADC3 FEN	ADC2 FEN	ADC1 FEN	ADC0 FEN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[14] CAN1FEN	**CAN1 功能时钟使能(CAN1 Function Clock Enable)** 0：关闭 1：使能
[13] CAN0FEN	**CAN0 功能时钟使能(CAN0 Function Clock Enable)** 0：关闭 1：使能
[12] USBFEN	**USB 功能时钟使能(USB Function Clock Enable)** 0：关闭 1：使能
[11] PWM7FEN	**PWM7 功能时钟使能(PWM7 Function Clock Enable)** 0：关闭 1：使能
[10] PWM6FEN	**PWM6 功能时钟使能(PWM6 Function Clock Enable)** 0：关闭 1：使能
[9] PWM5FEN	**PWM5 功能时钟使能(PWM5 Function Clock Enable)** 0：关闭 1：使能
[8] PWM4FEN	**PWM4 功能时钟使能(PWM4 Function Clock Enable)** 0：关闭 1：使能
[7] PWM3FEN	**PWM3 功能时钟使能(PWM3 Function Clock Enable)** 0：关闭 1：使能
[6] PWM2FEN	**PWM2 功能时钟使能(PWM2 Function Clock Enable)** 0：关闭 1：使能
[5] PWM1FEN	**PWM1 功能时钟使能(PWM1 Function Clock Enable)** 0：关闭 1：使能
[4] PWM0FEN	**PWM0 功能时钟使能(PWM0 Function Clock Enable)** 0：关闭 1：使能
[3] ADC3FEN	**ADC3 功能时钟使能(ADC3 Function Clock Enable)** 0：关闭 1：使能
[2] ADC2FEN	**ADC2 功能时钟使能(ADC2 Function Clock Enable)** 0：关闭 1：使能
[1] ADC1FEN	**ADC1 功能时钟使能(ADC1 Function Clock Enable)**

	0：关闭 1：使能
[0] ADC0FEN	**ADC0 功能时钟使能(ADC0 Function Clock Enable)** 0：关闭 1：使能

7.3.2.9 RCU APB0 时钟控制寄存器（RCU_APB0ENR）

- **名称：APB0 Peripheral Clock Enable Register**
- **偏移地址：0x50**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TMR8EN	TMR7EN	UART2EN	UART1EN	UART0EN	I2C2EN	I2C1EN	I2C0EN
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] TMR8EN	**TMR8 时钟使能(TMR8 Clock Enable)** 0：关闭 1：使能
[6] TMR7EN	**TMR7 时钟使能(TMR7 Clock Enable)** 0：关闭 1：使能
[5] UART2EN	**UART2 时钟使能(UART2 Clock Enable)** 0：关闭 1：使能
[4] UART1EN	**UART1 时钟使能(UART1 Clock Enable)** 0：关闭 1：使能
[3] UART0EN	**UART0 时钟使能(UART0 Clock Enable)** 0：关闭 1：使能
[2] I2C2EN	**I2C2 时钟使能(I2C2 Clock Enable)** 0：关闭 1：使能
[1] I2C1EN	**I2C1 时钟使能(I2C1 Clock Enable)** 0：关闭 1：使能
[0] I2C0EN	**I2C0 时钟使能(I2C0 Clock Enable)** 0：关闭 1：使能

7.3.2.10 RCU APB1 时钟控制寄存器（RCU_APB1ENR）

- 名称：APB1 Peripheral Clock Enable Register
- 偏移地址：0x54
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		TMR2EN	TMR1EN	TMR0EN	PDM3EN	PDM2EN	PDM1EN	PDM0EN	XIFEN	CAN1EN	CAN0EN	SPI1EN	SPI0EN	UART4EN	UART3EN
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[13] TMR2EN	**TMR2 时钟使能(TMR2 Clock Enable)** 0: 关闭 1: 使能
[12] TMR1EN	**TMR1 时钟使能(TMR1 Clock Enable)** 0: 关闭 1: 使能
[11] TMR0EN	**TMR0 时钟使能(TMR0 Clock Enable)** 0: 关闭 1: 使能
[10] PDM3EN	**PDM3 时钟使能(PDM3 Clock Enable)** 0: 关闭 1: 使能
[9] PDM2EN	**PDM2 时钟使能(PDM2 Clock Enable)** 0: 关闭 1: 使能
[8] PDM1EN	**PDM1 时钟使能(PDM1 Clock Enable)** 0: 关闭 1: 使能
[7] PDM0EN	**PDM0 时钟使能(PDM0 Clock Enable)** 0: 关闭 1: 使能
[6] XIFEN	**XIF 时钟使能(XIF Clock Enable)** 0: 关闭 1: 使能
[5] CAN1EN	**CAN1 时钟使能(CAN1 Clock Enable)** 0: 关闭 1: 使能
[4] CAN0EN	**CAN0 时钟使能(CAN0 Clock Enable)** 0: 关闭 1: 使能
[3] SPI1EN	**SPI1 时钟使能(SPI1 Clock Enable)** 0: 关闭 1: 使能
[2] SPI0EN	**SPI0 时钟使能(SPI0 Clock Enable)** 0: 关闭 1: 使能
[1] UART4EN	**UART4 时钟使能(UART4 Clock Enable)** 0: 关闭 1: 使能
[0] UART3EN	**UART3 时钟使能(UART3 Clock Enable)**

0：关闭
1：使能

7.3.2.11 RCU AHB0 时钟控制寄存器（RCU_AHB0ENR）

- 名称：**AHB0 Peripheral Clock Enable Register**
- 偏移地址：**0x58**
- 默认值：**0x000000FE**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			QEI2EN	QEI1EN	QEI0EN	TMR4EN	TMR3EN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN	FLSEN	DMAEN
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0x0	0x0	0x0	0x0	0x0	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x0

字段	说明
[12] QEI2EN	**QEI2 时钟使能(QEI2 Clock Enable)** 0：关闭 1：使能
[11] QEI1EN	**QEI1 时钟使能(QEI1 Clock Enable)** 0：关闭 1：使能
[10] QEI0EN	**QEI0 时钟使能(QEI0 Clock Enable)** 0：关闭 1：使能
[9] TMR4EN	**TMR4 时钟使能(TMR4 Clock Enable)** 0：关闭 1：使能
[8] TMR3EN	**TMR3 时钟使能(TMR3 Clock Enable)** 0：关闭 1：使能
[7] PFEN	**GPIOF 时钟使能(GPIOF Clock Enable)** 0：关闭 1：使能
[6] PEEN	**GPIOE 时钟使能(GPIOE Clock Enable)** 0：关闭 1：使能
[5] PDEN	**GPIOD 时钟使能(GPIOD Clock Enable)** 0：关闭 1：使能
[4] PCEN	**GPIOC 时钟使能(GPIOC Clock Enable)** 0：关闭 1：使能
[3] PBEN	**GPIOB 时钟使能(GPIOB Clock Enable)** 0：关闭 1：使能
[2] PAEN	**GPIOA 时钟使能(GPIOA Clock Enable)** 0：关闭 1：使能
[1] FLSEN	**FLASH 时钟使能(FLASH Clock Enable)** 0：关闭 1：使能

[0] DMAEN	**DMA 时钟使能(DMA Clock Enable)** 0: 关闭 1: 使能

7.3.2.12 RCU AHB1 时钟控制寄存器（RCU_AHB1ENR）

- **名称：AHB1 Peripheral Clock Enable Register**
- **偏移地址：0x5C**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CORDICEN	IIR5EN	IIR4EN	IIR3EN	IIR2EN	IIR1EN	IIR0EN	PWM7EN
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM6EN	PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	CMPxEN	DACxEN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	USBEN	TMR10EN	TMR9EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[23] CORDICEN	**CORDIC 时钟使能(CORDIC Clock Enable)** 0: 关闭 1: 使能
[22] IIR5EN	**IIR5 时钟使能(IIR5 Clock Enable)** 0: 关闭 1: 使能
[21] IIR4EN	**IIR4 时钟使能(IIR4 Clock Enable)** 0: 关闭 1: 使能
[20] IIR3EN	**IIR3 时钟使能(IIR3 Clock Enable)** 0: 关闭 1: 使能
[19] IIR2EN	**IIR2 时钟使能(IIR2 Clock Enable)** 0: 关闭 1: 使能
[18] IIR1EN	**IIR1 时钟使能(IIR1 Clock Enable)** 0: 关闭 1: 使能
[17] IIR0EN	**IIR0 时钟使能(IIR0 Clock Enable)** 0: 关闭 1: 使能
[16] PWM7EN	**PWM7 时钟使能(PWM7 Clock Enable)** 0: 关闭 1: 使能
[15] PWM6EN	**PWM6 时钟使能(PWM6 Clock Enable)** 0: 关闭 1: 使能
[14] PWM5EN	**PWM5 时钟使能(PWM5 Clock Enable)** 0: 关闭 1: 使能
[13] PWM4EN	**PWM4 时钟使能(PWM4 Clock Enable)** 0: 关闭 1: 使能
[12] PWM3EN	**PWM3 时钟使能(PWM3 Clock Enable)**

	0: 关闭 1: 使能
[11] PWM2EN	**PWM2 时钟使能(PWM2 Clock Enable)** 0: 关闭 1: 使能
[10] PWM1EN	**PWM1 时钟使能(PWM1 Clock Enable)** 0: 关闭 1: 使能
[9] PWM0EN	**PWM0 时钟使能(PWM0 Clock Enable)** 0: 关闭 1: 使能
[8] CMPxEN	**CMPx 时钟使能(CMPx Clock Enable)** 0: 关闭 1: 使能
[7] DACxEN	**DACx 时钟使能(DACx Clock Enable)** 0: 关闭 1: 使能
[6] ADC3EN	**ADC3 时钟使能(ADC3 Clock Enable)** 0: 关闭 1: 使能
[5] ADC2EN	**ADC2 时钟使能(ADC2 Clock Enable)** 0: 关闭 1: 使能
[4] ADC1EN	**ADC1 时钟使能(ADC1 Clock Enable)** 0: 关闭 1: 使能
[3] ADC0EN	**ADC0 时钟使能(ADC0 Clock Enable)** 0: 关闭 1: 使能
[2] USBEN	**USB 时钟使能(USB Clock Enable)** 0: 关闭 1: 使能
[1] TMR10EN	**TMR10 时钟使能(TMR10 Clock Enable)** 0: 关闭 1: 使能
[0] TMR9EN	**TMR9 时钟使能(TMR9 Clock Enable)** 0: 关闭 1: 使能

7.3.2.13 RCU APB0 复位控制寄存器（RCU_APB0RSTR）

- 名称：APB0 Peripheral Reset Register
- 偏移地址：0x60
- 默认值：0x000000FF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TMR8 RST	TMR7 RST	UART 2RST	UART 1RST	UART 0RST	I2C2R ST	I2C1R ST	I2C0R ST
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1

字段	说明
[7] TMR8RST	**TMR8 复位控制(TMR8 Reset)** 0: 复位 1: 释放
[6] TMR7RST	**TMR7 复位控制(TMR7 Reset)** 0: 复位 1: 释放
[5] UART2RST	**UART2 复位控制(UART2 Reset)** 0: 复位 1: 释放
[4] UART1RST	**UART1 复位控制(UART1 Reset)** 0: 复位 1: 释放
[3] UART0RST	**UART0 复位控制(UART0 Reset)** 0: 复位 1: 释放
[2] I2C2RST	**I2C2 复位控制(I2C2 Reset)** 0: 复位 1: 释放
[1] I2C1RST	**I2C1 复位控制(I2C1 Reset)** 0: 复位 1: 释放
[0] I2C0RST	**I2C0 复位控制(I2C0 Reset)** 0: 复位 1: 释放

7.3.2.14 RCU APB1 复位控制寄存器（RCU_APB1RSTR）

- 名称：APB1 Peripheral Reset Register
- 偏移地址：0x64
- 默认值：0x00003FFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		TMR2RST	TMR1RST	TMR0RST	PDM3RST	PDM2RST	PDM1RST	PDM0RST	XIFRST	CAN1RST	CAN0RST	SPI1RST	SPI0RST	UART4RST	UART3RST
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1

字段	说明
[13] TMR2RST	**TMR2 复位控制(TMR1 Reset)** 0: 复位 1: 释放
[12] TMR1RST	**TMR1 复位控制(TMR1 Reset)** 0: 复位 1: 释放
[11] TMR0RST	**TMR0 复位控制(TMR0 Reset)** 0: 复位 1: 释放
[10] PDM3RST	**PDM3 复位控制(PDM3 Reset)** 0: 关闭 1: 使能
[9] PDM2RST	**PDM2 复位控制(PDM2 Reset)** 0: 关闭 1: 使能
[8] PDM1RST	**PDM1 复位控制(PDM1 Reset)** 0: 关闭 1: 使能
[7] PDM0RST	**PDM0 复位控制(PDM0 Reset)** 0: 关闭 1: 使能
[6] XIFRST	**XIF 复位控制(XIF Reset)** 0: 复位 1: 释放
[5] CAN1RST	**CAN1 复位控制(CAN1 Reset)** 0: 复位 1: 释放
[4] CAN0RST	**CAN0 复位控制(CAN0 Reset)** 0: 复位 1: 释放
[3] SPI1RST	**SPI1 复位控制(SPI1 Reset)** 0: 复位 1: 释放
[2] SPI0RST	**SPI0 复位控制(SPI0 Reset)** 0: 复位 1: 释放
[1] UART4RST	**UART4 复位控制(UART4 Reset)** 0: 复位 1: 释放
[0] UART3RST	**UART3 复位控制(UART3 Reset)**

0：复位
1：释放

7.3.2.15 RCU AHB0 复位控制寄存器（RCU_AHB0RSTR）

- 名称：**AHB0 Peripheral Reset Register**
- 偏移地址：**0x68**
- 默认值：**0x00001FFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			QEI2RST	QEI1RST	QEI0RST	TMR4RST	TMR3RST	PFRST	PERST	PDRST	PCRST	PBRST	PARST	FLSRST	DMARST
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1

字段	说明
[12] QEI2RST	**QEI2 复位控制(QEI2 Reset)** 0：复位 1：释放
[11] QEI1RST	**QEI1 复位控制(QEI1 Reset)** 0：复位 1：释放
[10] QEI0RST	**QEI0 复位控制(QEI0 Reset)** 0：复位 1：释放
[9] TMR4RST	**TMR4 复位控制(TMR4 Reset)** 0：复位 1：释放
[8] TMR3RST	**TMR3 复位控制(TMR3 Reset)** 0：复位 1：释放
[7] PFRST	**GPIOF 复位控制(GPIOF Reset)** 0：复位 1：释放
[6] PERST	**GPIOE 复位控制(GPIOE Reset)** 0：复位 1：释放
[5] PDRST	**GPIOD 复位控制(GPIOD Reset)** 0：复位 1：释放
[4] PCRST	**GPIOC 复位控制(GPIOC Reset)** 0：复位 1：释放
[3] PBRST	**GPIOB 复位控制(GPIOB Reset)** 0：复位 1：释放
[2] PARST	**GPIOA 复位控制(GPIOA Reset)** 0：复位 1：释放
[1] FLSRST	**FLASH 复位控制(FLASH Reset)** 0：复位 1：释放

[0] DMARST	**DMA 复位控制(DMA Reset)** 0: 复位 1: 释放

7.3.2.16 RCU AHB1 复位控制寄存器（RCU_AHB1RSTR）

- **名称：AHB1 Peripheral Reset Register**
- **偏移地址：0x6C**
- **默认值：0x00003FFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		CORDICRST	IIR5RST	IIR4RST	IIR3RST	IIR2RST	IIR1RST	IIR0RST	PWMRST	CMPRST	DACRST	ADCRST	USBRST	TMR10RST	TMR9RST
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1

字段	说明
[13] CORDICRST	**CORDIC 复位控制(CORDIC Reset)** 0: 关闭 1: 使能
[12] IIR5RST	**IIR5 复位控制(IIR5 Reset)** 0: 关闭 1: 使能
[11] IIR4RST	**IIR4 复位控制(IIR4 Reset)** 0: 关闭 1: 使能
[10] IIR3RST	**IIR3 复位控制(IIR3 Reset)** 0: 关闭 1: 使能
[9] IIR2RST	**IIR2 复位控制(IIR2 Reset)** 0: 关闭 1: 使能
[8] IIR1RST	**IIR1 复位控制(IIR1 Reset)** 0: 关闭 1: 使能
[7] IIR0RST	**IIR0 复位控制(IIR0 Reset)** 0: 关闭 1: 使能
[6] PWMRST	**PWM 复位控制(PWM Reset)** 0: 关闭 1: 使能
[5] CMPRST	**CMP 复位控制(CMP Reset)** 0: 关闭 1: 使能
[4] DACRST	**DAC 复位控制(DAC Reset)** 0: 关闭 1: 使能
[3] ADCRST	**ADC 复位控制(ADC Reset)** 0: 关闭 1: 使能
[2] USBRST	**USB 复位控制(USB Reset)** 0: 关闭

	1: 使能
[1] TMR10RST	**TMR10 复位控制(TMR10 Reset)** 0: 关闭 1: 使能
[0] TMR9RST	**TMR9 复位控制(TMR9 Reset)** 0: 关闭 1: 使能

7.3.2.17 RCU 晶振控制寄存器（RCU_XOSCCR）

- **名称：XOSC Control Register**
- **偏移地址：0x70**
- **默认值：0x00000016**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											HEN	XDR			XEN
											R/W	R/W			R/W
											0x1	0x3			0x0

字段	说明
[4] HEN	**HSI 使能(HSI Enable)** 0: 关闭 1: 启动
[3:1] XDR	**HSE 启动电流配置(HSE Charge Pump Selection)** 000: 0.25mA 001: 0.375mA 010: 0.5mA 011: 0.625mA 100: 0.75mA 101: 0.875mA 110: 1.0mA 111: 1.125mA
[0] XEN	**HSE 使能(HSE Enable)** 0: 关闭 1: 启动

7.3.2.18 RCU 时钟安全控制寄存器（RCU_CSSCR）

- 名称：Clock Secure Control Register
- 偏移地址：0x74
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								LMT		WIN		LPD	LPE	SSE	SWE
								R/W		R/W		R	R/W	R/W	R/W
								0x0		0x0		0x0	0x0	0x0	0x0

字段	说明
[7:6] LMT	**XOSC 比较阈值宽度(XOSC Limit Window Width)** 0:±2Cycle 1:±4Cycle 2:±6Cycle 3:±8Cycle
[5:4] WIN	**XOSC 计数窗口宽度调整(XOSC AutoSwitch Window Width)** 0:3us 1:6us …
[3] LPD	**外部晶振异常 NMI 中断状态(XOSC Loss Pending)** 0：未出现异常 1：出现异常 Note：晶振异常时产生 Pending 状态标志位，该位只读。
[2] LPE	**外部晶振异常 NMI 中断使能(XOSC Loss NMI Enable)** 0:关闭 1:启动
[1] SSE	**系统时钟切换使能(SYSCLK Switch Enable)** 0: SYSCLK 源选择的是 HSE，当发生异常时，SYSCLK 源会直接切换到 HSI； 1: SYSCLK 源选择的是 HSE 或 PLL，当发生异常时，SYSCLK 源会直接切换到 HSI；
[0] SWE	**外部晶振安全监测使能(XOSC Secure Switch Enable)** 0:关闭 1:启动

7.3.2.19 RCU 内部时钟 FanOut（RCU_DBGCR）

- 名称：**Internal Clock Fanout Register**
- 偏移地址：**0x78**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										ECIE	MCOEN		MCO		
										R/W	R/W		R/W		
										0x0	0x0		0x0		

字段	说明
[5] ECIE	**芯片外部时钟输入使能(External Clock Input Enable)** 0: 关闭 1: 使能
[4] MCOEN	**芯片内部时钟输出使能(Internal Clock Fanout Enable)** 0: 关闭 1: 使能
[2:0] MCO	**芯片内部时钟输出源选择(Internal Clock Fanout Source)** 000: LSI 001: HSI 010: HSE 011: SYSCLK 100: PLL0/16 101: Reserved 110: FCLK 111: ATECLK

7.3.2.20 RCU 系统复位状态寄存器（RCU_SRSTSR）

- 名称：**System RstStatus Register**
- 偏移地址：**0x7C**
- 默认值：**0x00000100**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IWRSTE	WWRSTE	LKRSTE	SQRSTE			WWRST	IWRST	LKRST	SQRST	LPRST	MCRST
				R/W	R/W	R/W	R/W			R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
				0x0	0x0	0x0	0x1			0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[11] IWRSTE	**IWDG RESET Enable** 0: 关闭 1: 使能
[10] WWRSTE	**WWDG RESET Enable** 0: 关闭 1: 使能

[9] LKRSTE	**LKUP RESET Enable** 0: 关闭 1: 使能
[8] SQRSTE	**SystemREQ RESET Enable** 0: 关闭 1: 使能
[5] WWRST	**WWDG RESET Status** 1: WWDG 产生 Reset 0:无 Reset
[4] IWRST	**IWDG RESET Status** 1: IWDG 产生 Reset 0:无 Reset
[3] LKRST	**LKUP RESET Status** 1: LKUP 产生 Reset 0:无 Reset
[2] SQRST	**SystemREQ RESET Status** 1: SystemREQ 产生 Reset 0:无 Reset
[1] LPRST	**LowPower RESET Status** 1: LowPower 产生 Reset 0:无 Reset
[0] MCRST	**MCLR RESET Status** 1: MCLR 产生 Reset 0:无 Reset

7.3.2.21 RCU 键寄存器（RCU_KEYR）

- **名称：LockKey Register**
- **偏移地址：0x80**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															KEY
															R/W
															0x0

字段	说明
[0] KEY	**寄存器写保护(Register LockKey)** 解锁： 向 KEYR 寄存器写入 Key 值(0x3fac87e4)即可完成解锁，KEY 位将由硬件置 1，然后进行相关的操作； 加锁： 向 KEY 寄存器写 0x1 即完成加锁，KEY 位将清 0；

8 通用 I/O（GPIO）

8.1 简介

GPIO 是一个可编程通用 I/O 外设（programmable General Purpose I/O），共包含 6 组 GPIO（GPIOA~F），GPIOA~E 每组均有 16 个 I/O，GPIOF 共有 5 个 I/O。

8.2 主要特性

GPIO 主要具有以下特性：

- 受控 I/O 高达 85 个
- 输出状态：推挽或开漏 + 上拉/下拉
- 输入状态：浮空、上拉/下拉、模拟
- 输出数据寄存器或外设（复用功能输出）输出数据
- 数据输入到输入数据寄存器或外设（复用功能输入）
- 模拟功能
- 置位和复位寄存器，对输出数据寄存器具有按位写权限
- 可选的同步功能和消抖功能
- 可配的电流驱动能力
- 每个 I/O 均可独立配置输出压摆率（Slew Rate）和输入迟滞（Hysteresis）
- 复用功能输入/输出选择寄存器
- 复用功能高灵活性，允许将 I/O 引脚配置为 GPIO 或各种外设功能
- 所有 I/O 均可独立配置外部输入触发中断使能，上升沿/下降沿/双边沿触发，独立的中断挂起标志

8.3 结构框图

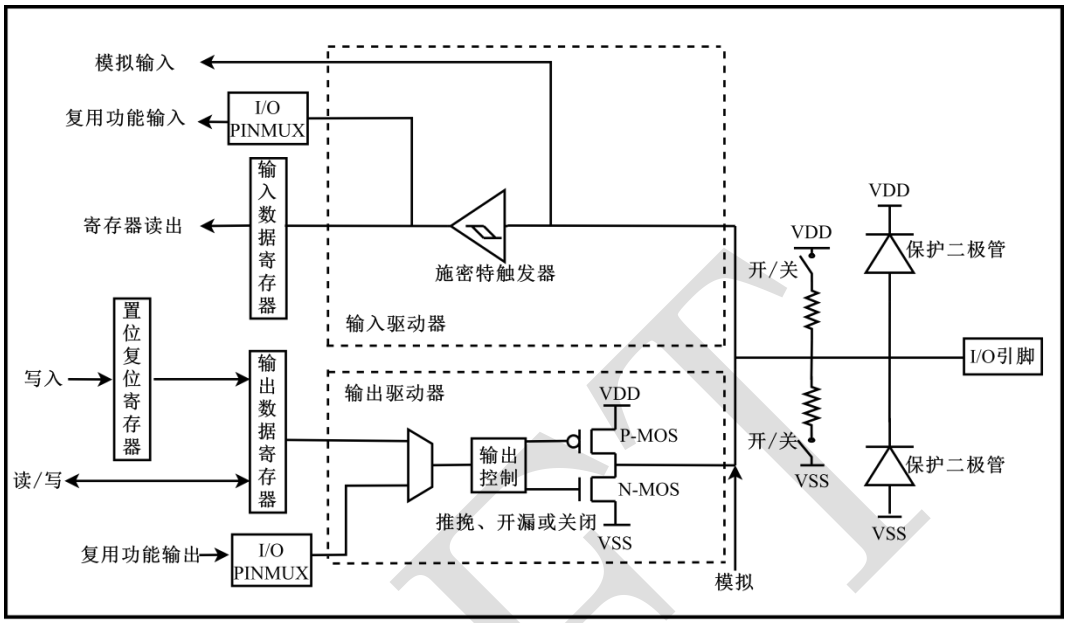


图 8-1 GPIO 结构框图

8.4 功能描述

通过软件可将 GPIO 各个端口分别配置为以下多种模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟功能
- 具有上拉或下拉的开漏输出
- 具有上拉或下拉的推挽输出
- 具有上拉或下拉的复用功能推挽
- 具有上拉或下拉的复用功能开漏

每个 I/O 端口的模式均可自由配置，模块内部寄存器必须按字（Word）进行操作。

为了实现一组 I/O 的原子置位/清零操作，通过端口置位/复位寄存器（GPIOx_BSR）的写“1”操作来完成对 I/O 输出数据寄存器（GPIOx_DOR）进行原子置位/清零操作。

8.4.1 I/O 复位状态

芯片复位期间：

- PB8 作为芯片 BOOT 引脚，默认为上拉状态，详细可参考“4.5 自举配置”
- 其他端口为浮空高阻模式

复位完成后：

- PA13/PA14 将默认复用为调试功能（AF2）

- PA13：SWDAT 处于上拉状态
- PA14：SWCLK 处于下拉状态

- 其他端口为浮空高阻模式

8.4.2 通用 I/O

当引脚配置为输出后，写入到输出数据寄存器（GPIOx_DOR）的值将在 I/O 引脚上输出。

当引脚配置为输入、输出及功能复用后，输入数据寄存器（GPIOx_DIR）每隔 1 个 CPU 时钟周期（F_{CLK}）捕获一次 I/O 引脚的数据，即配置为当引脚配置输出及功能复用后，作为 I/O 引脚的读回功能。

所有 GPIO 引脚都具有内部弱上/下拉电阻，可根据 GPIOx_UDR 寄存器中的值来选择打开/关闭。

8.4.3 引脚复用

I/O 引脚通过一个复用逻辑单元连接到不同外设模块，该复用逻辑单元一次仅允许一个外设复用到 I/O 引脚上，这样可以确保共用一个 I/O 引脚的外设之间不会发生冲突。

芯片内部每个 I/O 引脚都有一个复用逻辑单元，其支持 16 路复用功能输入（AF0 至 AF15），可通过 GPIOx_MR0（针对 I/O 引脚 0~7）和 GPIOx_MR1（针对 I/O 引脚 8~15）寄存器对复用功能进行配置：

- AF0 为 I/O 输入模式，AF1 为 I/O 输出模式
- 外设的复用功能映射到 AF2 至 AF14
- AF15 为模拟功能（Analog function）
- 芯片复位完成后，除 PA13/PA14 外（详见 8.4.1 I/O 复位状态），其他 I/O 都会映射到模拟功能复用（AF15）

I/O 复用架构除了具有灵活性，各外设还可以将复用功能映射到不同 I/O 引脚上，这可以优化在小型封装中可使用外设的数量。

复用功能配置说明

系统及调试功能：

- SWD：复位后，会将 PA13/14 引脚指定为 SWD 功能，供片上调试使用。
- MCO：复位后，默认为 AF15，若要使用 MCO 复用功能须配置为复用功能模式。

GPIO：

- 在 GPIOx_MR0/GPIOx_MR1 寄存器中将所需 I/O 配置为输出或输入功能。

外设复用功能：

- 对于 ADC/DAC/CMP 模拟功能以及 USB 模块，在 GPIOx_MR0/GPIOx_MR1 寄存器中将所需 I/O 配置为模拟功能复用（AF15）。
- 其他外设复用功能，在 GPIOx_MR0/GPIOx_MR1 寄存器中将所需 I/O 配置为对应的复用

功能(AF2~AF14)，引脚的特定复用功能分配请详见“3 引脚定义”。

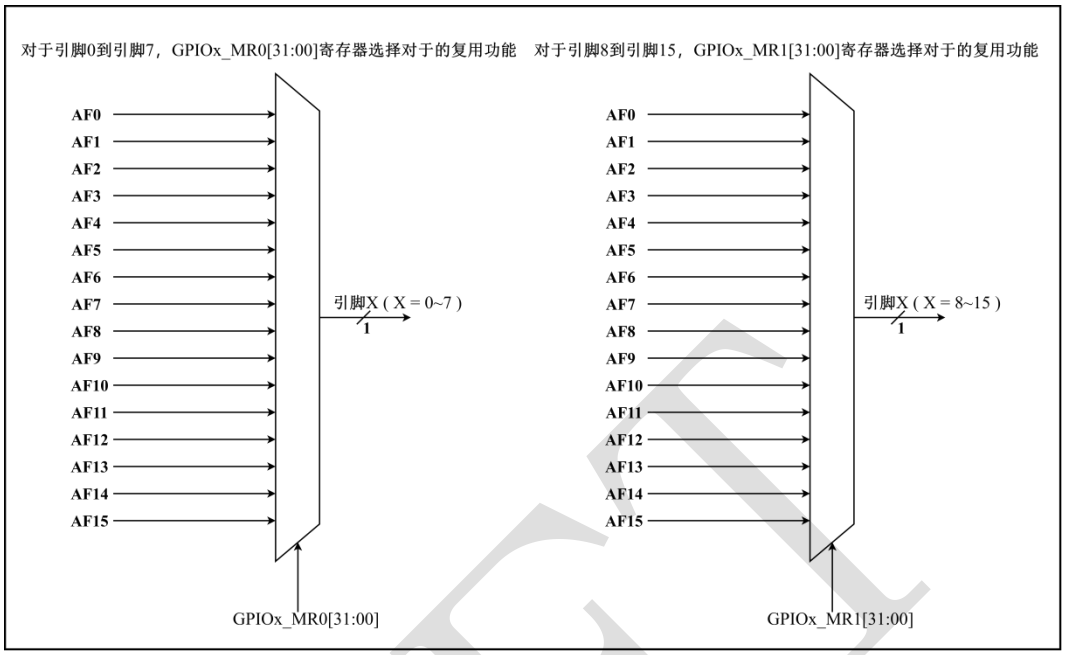


图 8-2 GPIO 选择复用功能示意图

8.4.4 I/O 控制寄存器

每个 GPIO 端口有 9 个控制寄存器（GPIOx_MR0、GPIOx_MR1、GPIOx_DIR、GPIOx_DOR、GPIOx_UDR、GPIOx_ODR、GPIOx_DHR、GPIOx_IHR、GPIOx_OSR），可配置多达 16 个 I/O。

其中 GPIOx_MR0、GPIOx_MR1 寄存器用于选择 I/O 复用功能（输入、输出、外设引脚复用、模拟功能）；GPIOx_ODR、GPIOx_UDR 寄存器分别用于选择 I/O 输出类型（推挽或开漏）和上下拉功能；GPIOx_OSR、GPIOx_IHR、GPIOx_DHR 寄存器用于配置 I/O 输出摆率、输出驱动能力及输入迟滞。

注意：I/O 上下拉功能不区分输入和输出方向，都可以配置，有关寄存器说明的详细信息，请参见下面 8.5 寄存器描述。

8.4.5 I/O 数据寄存器

每个GPIO都具有1个16位数据寄存器，其中输入和输出数据寄存器（GPIOx_DIR/GPIOx_DOR）既可用于存储输出数据，也可以用于存储输入数据。当用于输出时，GPIOx_DOR 用于存储待输出数据，可对其进行读/写访问；当用于输入时，通过 I/O 输入的数据存储到输入数据寄存器（GPIOx_DIR）中。

注意：有关寄存器说明的详细信息，请参见下面 8.5 寄存器描述。

8.4.6 I/O 数据位操作

每个 GPIO 都具有 1 个 32 位置位复位寄存器（GPIOx_BSR），其允许应用程序对输出数据中的各个单独的数据位执行置位和复位操作。置位复位寄存器是 32 位寄存器，其中低 16 位用于 Bit Set 功能，而高 16 位用于 Bit Reset 功能。当向 GPIOx_BSR 寄存器的低 16 位中某一位写 1 时，其对应位 I/O 将置位；当向 GPIOx_BSR 寄存器的高 16 位中某一位写 1 时，其对应位 I/O 将清零。

注意：

- *当向 GPIOx_BSR 寄存器写入 0 对 Bit Set 和 Reset 都不会产生任何影响。*
- *当向 GPIOx_BSR 寄存器中 Bit Set 和 Reset 同时写入 1 时也不会产生任何影响，不允许同时置位操作。*
- *通过 GPIOx_BSR 寄存器可以实现 AHB 总线原子写操作，实现寄存器单 Bit 或多 Bit 修改操作。*

8.4.7 I/O 复用功能

每个 GPIO 都具有 2 个 32 位的 I/O 功能复用寄存器，共 16 个 I/O，每个 I/O 对应 4Bit，用来选择 16 个复用功能。通过 GPIOx_MRn（n = 0 ~ 1）寄存器根据应用程序的要求将某一种复用功能映射到对应引脚上。

使用 GPIOx_MRn 复用功能寄存器，可以在任意一个 I/O 上灵活选择一个功能复用。具体每个 GPIO 引脚上能够复用哪些功能，请参见“3 引脚定义”。

8.4.8 外部中断功能

所有 GPIO 端口都具有外部中断功能，要使用外部中断功能，必须将端口配置为输入模式，同时配置 GPIOx_IMR 寄存器选择需要触发的边缘（上升沿/下降沿/双边沿），配置 GPIOx_IER 寄存器开启所需 GPIO 端口的中断使能。可通过 GPIOx_ISR 寄存器的标志位来检查对应 GPIO 端口是否触发了中断。

8.4.9 输入配置

当 I/O 端口配置为输入时：

- 输出寄存器被禁止
- 输入施密特触发可被激活（输入迟滞）
- 根据配置 GPIOx_UDR 寄存器来决定 I/O 是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 CPU 时钟周期（F_{CLK}）对 I/O 引脚上的数据进行一次采样
- 读取输入数据寄存器可获得对应 I/O 的状态

输入模式

注意：所有 GPIO 引脚都具有一个上拉和下拉电阻，通过对应寄存器配置可以打开或关闭。

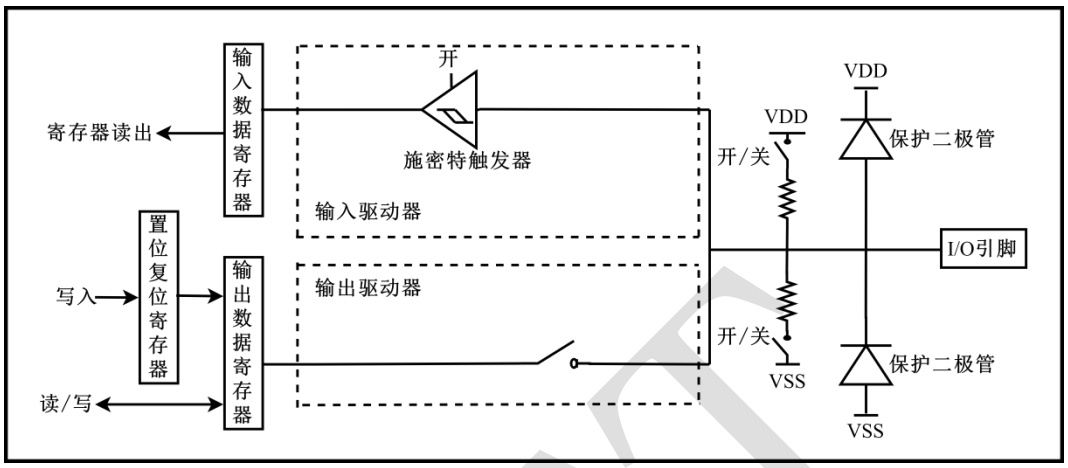


图 8-3　GPIO 输入上拉/下拉配置

浮空状态

当 I/O 引脚的上拉和下拉电阻都处于断开状态，此时 I/O 引脚处于悬空状态，容易收到外部干扰，输入数据寄存器中的数据并不准确。

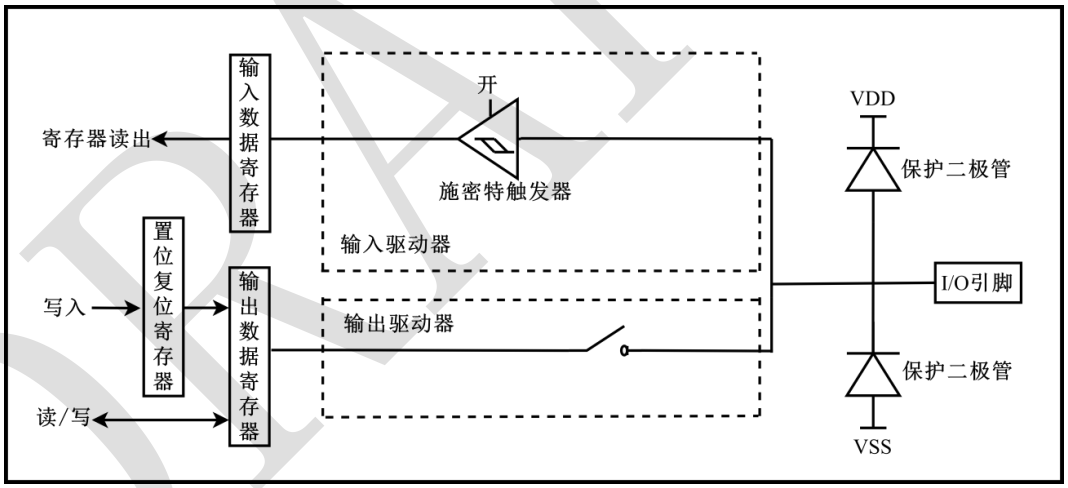


图 8-4　GPIO 输入浮空配置

输入上拉

打开 I/O 引脚的上拉电阻，将 I/O 引脚的不确定信号通过一个电阻嵌到高电平，电阻起到限流作用。

通常可以给 I/O 引脚外接一个开关，开关一端接 I/O 引脚，一端接地。当按下按钮时，读出低电平；松开按钮时，读出高电平。也可外接一个强上拉电阻，一直读取到高电平。

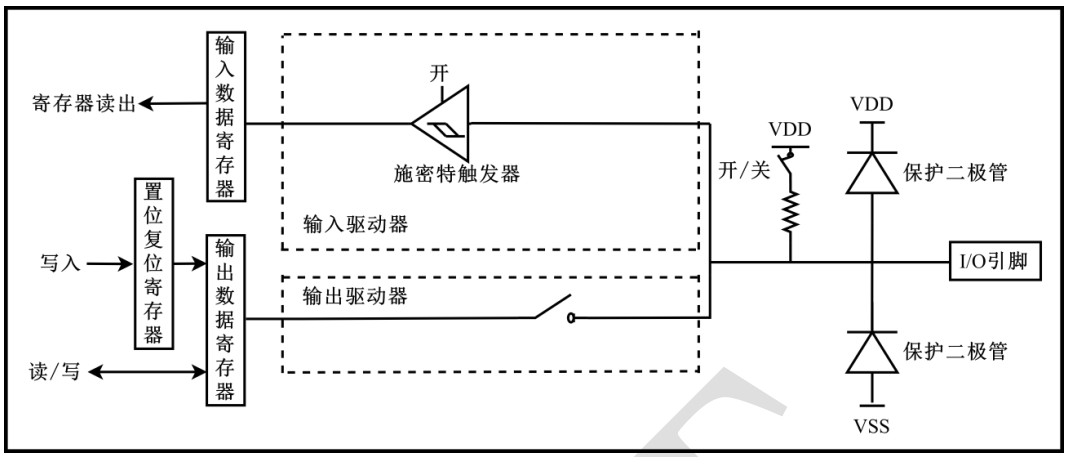


图 8-5　GPIO 输入上拉配置

输入下拉

与上拉输入原理一样，将电位拉到 GND，读到低电平。

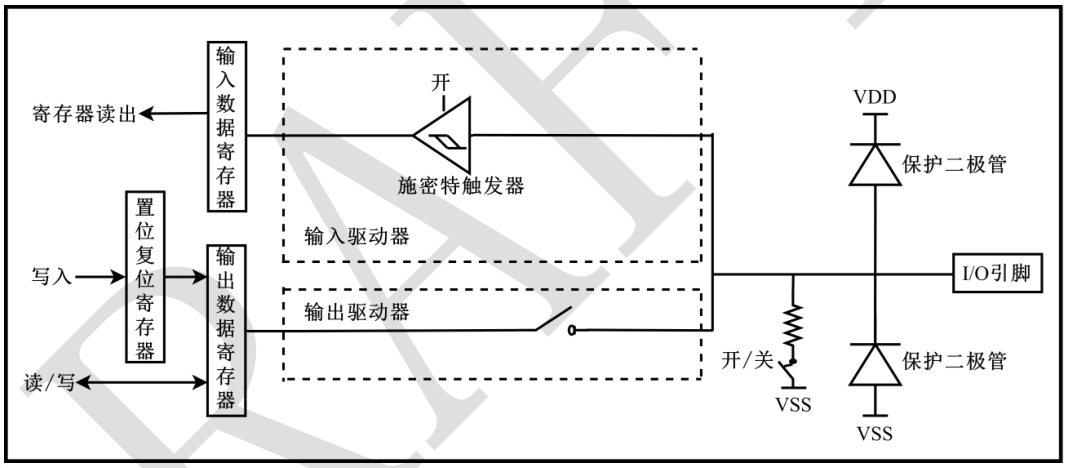


图 8-6　GPIO 输入下拉配置

8.4.10　输出配置

当 I/O 端口被配置为输出时：

- 输出数据寄存器被激活，输入数据寄存器被禁用
  - 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高阻状态（P-MOS 始终不激活）。
  - 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS。
- 根据配置 GPIOx_UDR 寄存器来决定 I/O 是否打开上拉和下拉电阻
- 对输出数据寄存器的读访问得到最后一次写的值

输出模式

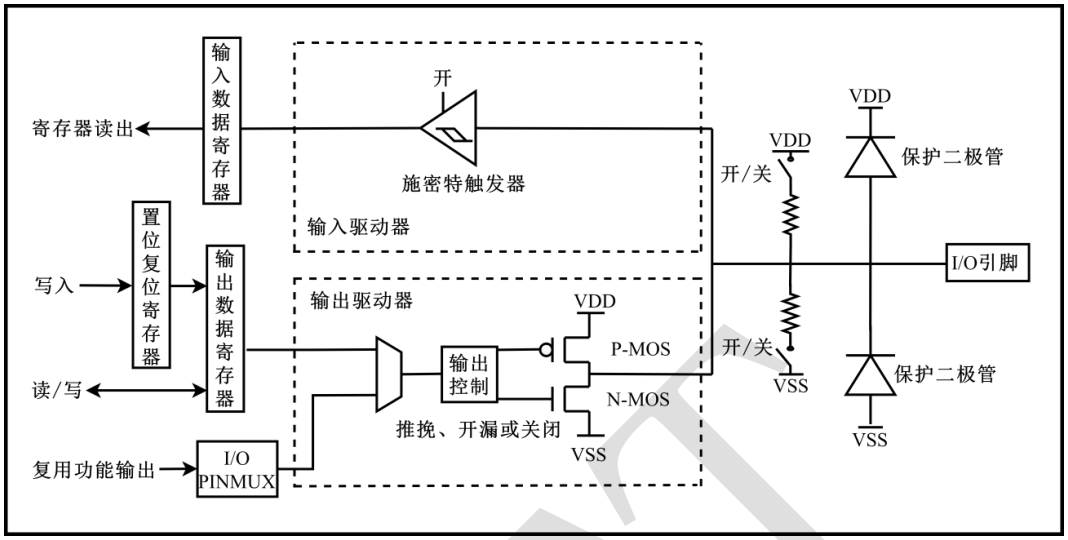


图 8-7　GPIO 输出推挽/开漏或关闭配置

推挽输出

可以输出高、低电平，连接数字器件；推挽结构一般是指两个三极管分别受两互补信号的控制，总是在一个三极管导通的时候另一个截止。

推挽输出，写 1

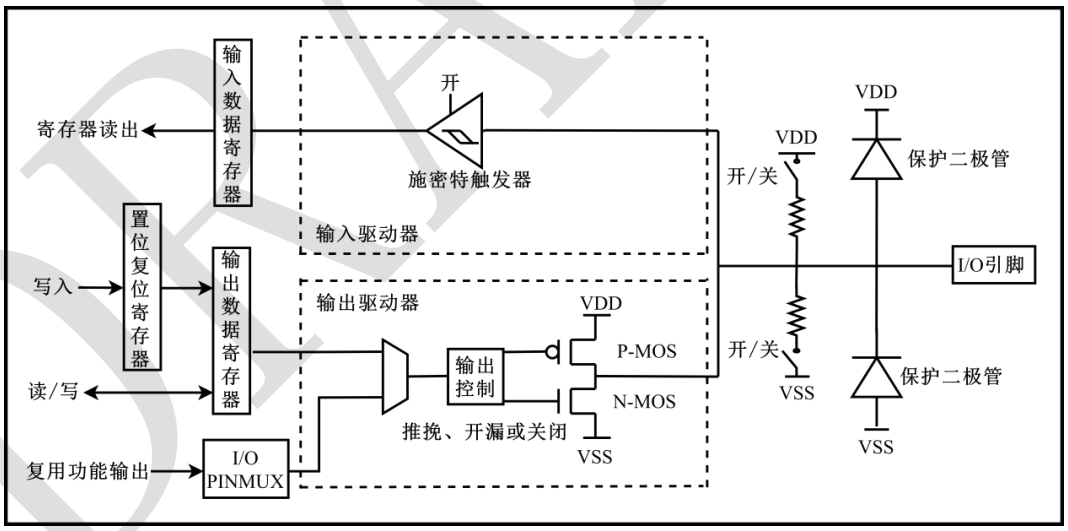


图 8-8　GPIO 推挽输出配置（写 1）

P-MOS 管激活，连接 VDD，I/O 引脚输出高电平，输入数据寄存器读到高电平。

推挽输出，写 0

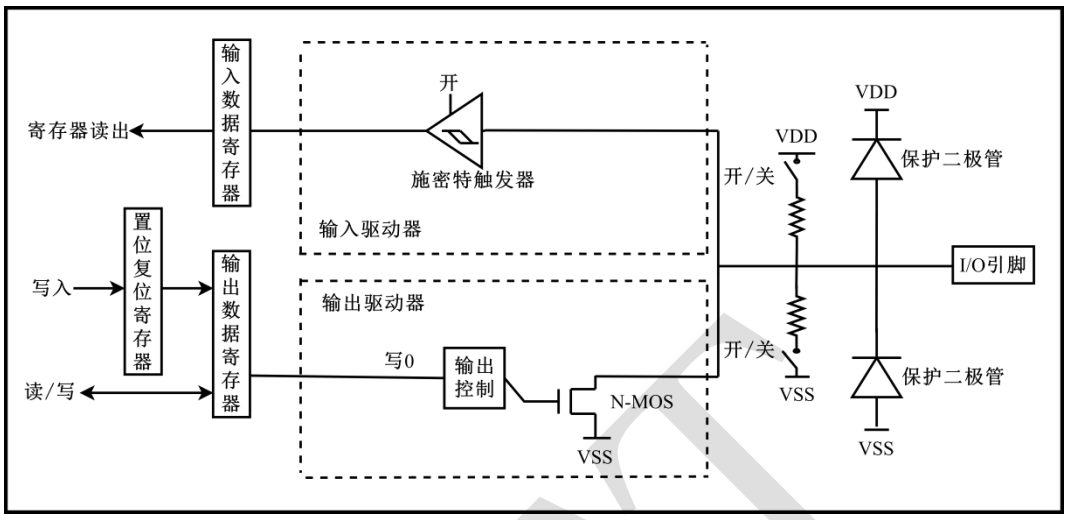


图 8-9　GPIO 推挽输出配置（写 0）

N-MOS 管被激活，连接到 VSS 接地，I/O 引脚输出低电平，输入数据寄存器读到低电平。

开漏输出

输出端相当于三极管的集电极，要得到高电平状态需要上拉电阻才行，适合于做电流型的驱动，其吸收电流的能力相对强。

开漏输出，写 1

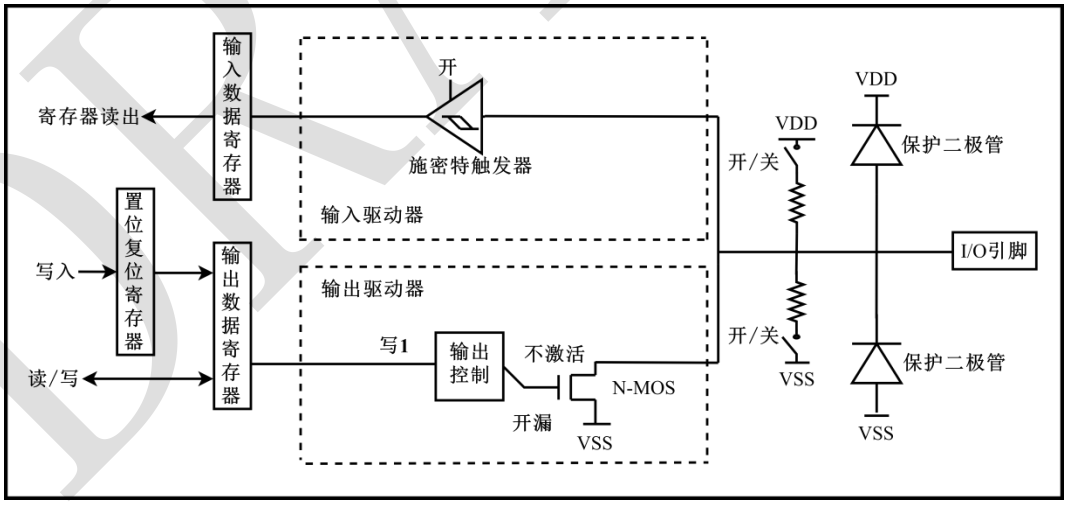


图 8-10　GPIO 开漏输出配置（写 1）

输出数据寄存器写入“1”，N-MOS 管不激活，引脚相当于浮空状态，引脚电平由外接电阻决定，接上拉电阻则为高电平，接下拉电阻相当于低电平。输出数据寄存器的写“1”不会使得引脚输出高电平，写“0”会使得引脚输出低电平。

开漏输出，写 0

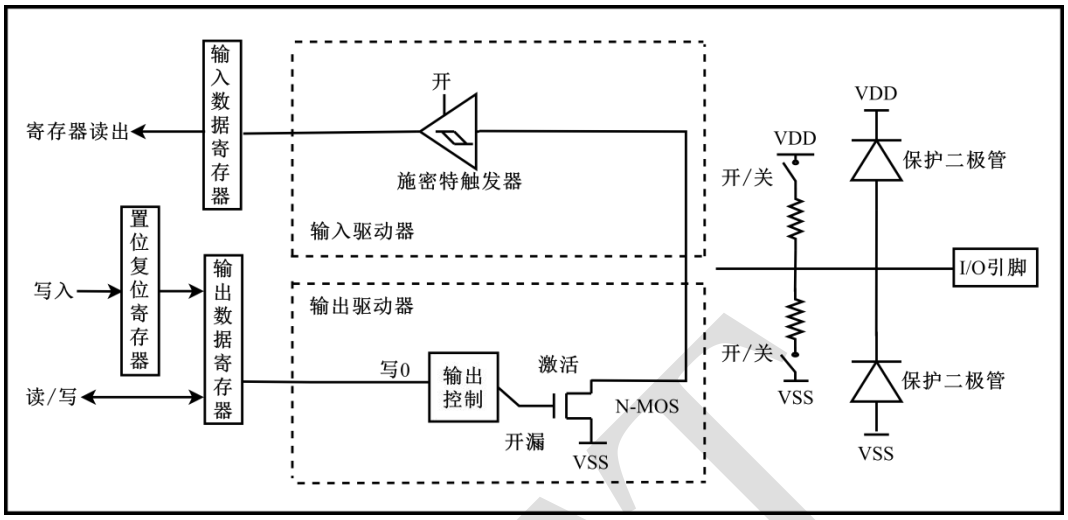


图 8-11　GPIO 开漏输出配置（写 0）

N-MOS 管激活，连接 VSS 接地，I/O 引脚输出低电平，输入数据寄存器也可以读到低电平。

8.4.11　复用功能配置

当 I/O 端口被配置为复用功能时：

- 可将输出配置为开漏或推挽式模式
- 输出缓冲器可由内置外设的信号驱动（复用功能输出）
- 根据配置 GPIOx_UDR 寄存器来决定 I/O 是否打开上拉和下拉电阻

复用推挽/开漏可以理解为 GPIO 口被用作复用功能时的配置情况，即非通用 I/O 口，GPIOx_MR0/ GPIOx_MR1 寄存器允许用户把一些复用功能重新映射到不同的引脚。

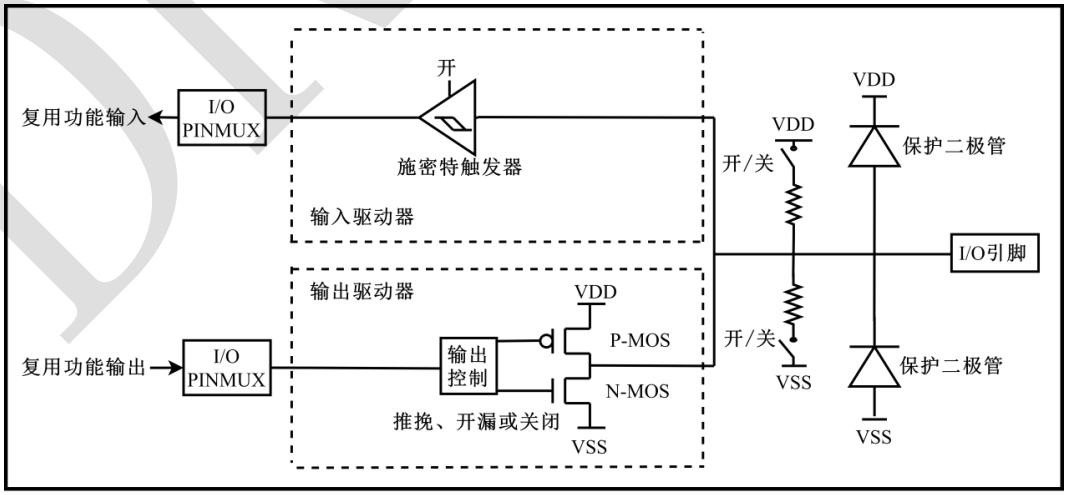


图 8-12　GPIO 复用功能配置

模拟输入配置

- 输出缓冲器被禁止
- 禁止施密特触发输入，实现了每个模拟 I/O 引脚上的零消耗，施密特触发输出值被强置为“0”

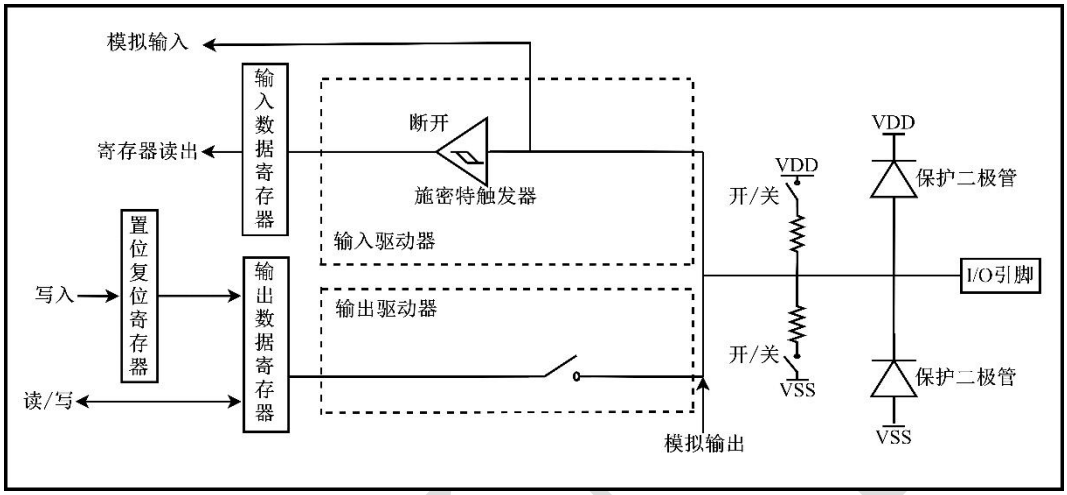


图 8-13　GPIO 模拟输入配置

注意：模拟输入，读到的不是高低电平，而是准确的电压值。

8.5 寄存器定义

8.5.1 寄存器列表

GPIO 基地址：
GPIOA(0x40024000) GPIOB(0x40025000) GPIOC(0x40026000)
GPIOD(0x40027000) GPIOE(0x40028000) GPIOF(0x40029000)

偏移	实例地址	名称	默认值	描述
0x00	GPIO 基地址+0x00	GPIO_BSR	0x00000000	GPIOx 端口置位/复位寄存器
0x04	GPIO 基地址+0x04	GPIO_DIR	0x00000000	GPIOx 输入数据寄存器
0x08	GPIO 基地址+0x08	GPIO_DOR	0x00000000	GPIOx 输出数据寄存器
0x10	GPIO 基地址+0x10	GPIO_IER	0x00000000	GPIOx 中断使能寄存器
0x14	GPIO 基地址+0x14	GPIO_IMR	0x00000000	GPIOx 中断模式寄存器
0x18	GPIO 基地址+0x18	GPIO_ISR	0x00000000	GPIOx 中断标志寄存器
0x20	GPIO 基地址+0x20	GPIO_MR0	0xFFFFFFFF	GPIOx 功能复用寄存器 0
0x24	GPIO 基地址+0x24	GPIO_MR1	0xFFFFFFFF	GPIOx 功能复用寄存器 1
0x28	GPIO 基地址+0x28	GPIO_SER	0x00000000	GPIOx 同步使能寄存器
0x2C	GPIO 基地址+0x2C	GPIO_DER	0x00000000	GPIOx 消抖使能寄存器
0x30	GPIO 基地址+0x30	GPIO_UDR	0x00000000	GPIOx 上拉/下拉使能寄存器
0x38	GPIO 基地址+0x38	GPIO_ODR	0x00000000	GPIOx 开漏使能寄存器
0x40	GPIO 基地址+0x40	GPIO_DHR	0x00000000	GPIOx 驱动强度寄存器
0x44	GPIO 基地址+0x44	GPIO_IHR	0x0000FFFF	GPIOx 输入迟滞寄存器
0x48	GPIO 基地址+0x48	GPIO_OSR	0x00000000	GPIOx 压摆控制寄存器

8.5.2 寄存器描述

8.5.2.1 GPIOx 端口置位/复位寄存器（GPIO_BSR）

- 名称：GPIOx Bit Set/Reset Register
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRn															
W1C															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSn															
W1C															
0x0															

字段	说明
[31:16] BRn	**GPIO 端口 x 引脚 n 输出复位(GPIO Bit Reset Contorl)** Note: 1. Bit Set/Reset 同时有效时，Set 优先级高； 2. n=0~15
[15:0] BSn	**GPIO 端口 x 引脚 n 输出置位(GPIO Bit Set Contorl)** Note: 1. Bit Set/Reset 同时有效时，Set 优先级高； 2. n=0~15

8.5.2.2 GPIOx 输入数据寄存器（GPIO_DIR）

- 名称：GPIOx Data Input Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIn															
R															
0x0															

字段	说明
[15:0] DIn	**GPIO 端口 x 引脚 n 输入数据寄存器(GPIO Data Input)** Note: 1. 当 I/O 配置为输入模式时，将端口数据采集到寄存器内，通过该寄存器来读取 I/O 上的数据输入； 2. n=0~15

8.5.2.3 GPIOx 输出数据寄存器（GPIO_DOR）

- 名称：GPIOx Data Output Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOn															
R/W															
0x0															

字段	说明
[15:0] DOn	**GPIO 端口 x 引脚 n 输出数据寄存器(GPIO Data Output)** Note： 1. 当 I/O 配置为输出模式时，将数据寄存器内数据映射到 I/O 端口上输出； 2. n=0~15

8.5.2.4 GPIOx 中断使能寄存器（GPIO_IER）

- 名称：GPIOx Interrupt Enable Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEn															
R/W															
0x0															

字段	说明
[15:0] IEn	**GPIO 端口 x 引脚 n 中断使能位(GPIO Interrupt Enable)** 0:关闭 1:启动 Note：n=0~15

8.5.2.5 GPIOx 中断模式寄存器（GPIO_IMR）

- 名称：GPIOx Interrupt Mode Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMn															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMn															
R/W															
0x0															

字段	说明
[31:0] IMn	**GPIO 端口 x 引脚 n 中断模式选择位(GPIO Interrupt Mode)** 00: 关闭 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

8.5.2.6 GPIOx 中断标志寄存器（GPIO_ISR）

- 名称：GPIOx Pending Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISn															
R/W1C															
0x0															

字段	说明
[15:0] ISn	**GPIO 端口 x 引脚 n 中断标志位(GPIO Interrupt Flag)** 0: No IRQ pending 1: Clear IRQ pending Note: 1. 该位写清 0，写 0 无效 2. n=0~15

8.5.2.7 GPIOx 功能复用寄存器 0（GPIO_MR0）

- 名称：GPIOx Mux Register0
- 偏移地址：0x20
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMn															
R/W															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMn															
R/W															
0xFFFFFFFF															

字段	说明
[31:0] PMn	**GPIO 端口 x 引脚 n 功能复用配置寄存器(GPIO MUX Selection)** 0000:AF0 …… 1111:AF15 Note：n=0~7

8.5.2.8 GPIOx 功能复用寄存器 1（GPIO_MR1）

- 名称：GPIOx Mux Register1
- 偏移地址：0x24
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMn															
R/W															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMn															
R/W															
0xFFFFFFFF															

字段	说明
[31:0] PMn	**GPIO 端口 x 引脚 n 功能复用配置寄存器(GPIO MUX Selection)** 0000:AF0 …… 1111:AF15 Note：n=8~15

8.5.2.9 GPIOx 同步使能寄存器（GPIO_SER）

- 名称：GPIOx Sync Register
- 偏移地址：0x28
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEn															
R/W															
0x0															

字段	说明
[15:0] SEn	**GPIO 端口 x 引脚 n 同步使能控制位(GPIO Sync Enable)** 0: 关闭 1: 使能 Note：n=0~15

8.5.2.10 GPIOx 消抖使能寄存器（GPIO_DER）

- 名称：GPIOx Debounce Register
- 偏移地址：0x2C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEn															
R/W															
0x0															

字段	说明
[15:0] DEn	**GPIO 端口 x 引脚 n 消抖使能控制位(GPIO Debounce Enable)** 0: 关闭 1: 使能 Note：n=0~15

8.5.2.11 GPIOx 上拉/下拉使能寄存器（GPIO_UDR）

- 名称：GPIOx Pull Up/Down Register
- 偏移地址：0x30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDn															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDn															
R/W															
0x0															

字段	说明
[31:0] UDn	**GPIO 端口 x 引脚 n 上下拉使能控制位(GPIO Pull Up/Down Enable)** 00: None 01: Pull Down 10: Pull Up 11: Pull Down/Up Note：n=0~15

8.5.2.12 GPIOx 开漏使能寄存器（GPIO_ODR）

- 名称：GPIOx Open Drain Register
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODn															
R/W															
0x0															

字段	说明
[15:0] ODn	**GPIO 端口 x 引脚 n 开漏使能控制位(GPIO Open Drain Enable)** 0: 推挽输出 1: 开漏输出 Note：n=0~15

8.5.2.13 GPIOx 驱动强度寄存器（GPIO_DHR）

- 名称：GPIOx Driver Register
- 偏移地址：0x40
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRn															
R/W															
0x0															

字段	说明
[15:0] DRn	**GPIO 端口 x 引脚 n 驱动能力选择(GPIO Driver Selection)** 0: 8mA 1: 24mA Note：n=0~15

8.5.2.14 GPIOx 输入迟滞寄存器（GPIO_IHR）

- 名称：GPIOx Hysteresis Register
- 偏移地址：0x44
- 默认值：0x0000FFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDn															
R/W															
0xFFFF															

字段	说明
[15:0] IDn	**GPIO 端口 x 引脚 n 输入迟滞使能控制位(GPIO Input Hysteresis Enable)** 0: 关闭 1: 使能 Note：n=0~15

8.5.2.15 GPIOx 压摆控制寄存器（GPIO_OSR）

- 名称：GPIOx Slew Register
- 偏移地址：0x48
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSn															
R/W															
0x0															

字段	说明
[15:0] OSn	**GPIO 端口 x 引脚 n 输出速率选择(GPIO Slew Selection)** 0: 正常压摆率 1: 增强压摆率 Note：n=0~15

9 系统配置控制器（SYSCTRL）

9.1 参考电压缓冲器（VREFBUF）

9.1.1 简介

芯片内置一个参考电压缓冲器，用于驱动 ADC、DAC 参考电压。基本功能包含：

- 两个参考电压挡位可配置：2.5V 和 2.9V。
- 可以关闭内部参考电压缓冲器，由片外参考源通过 VREF+引脚直接提供参考电压。

9.1.2 主要特性

- 参考电压可配置
  - 片内 2.5V
  - 片内 2.9V
  - 片外 2V~VDDA
- 内部参考电压温度系数<60ppm/℃
- 参考电压精度<4‰
- 最大负载电流能力 10mA

9.1.3 结构框图

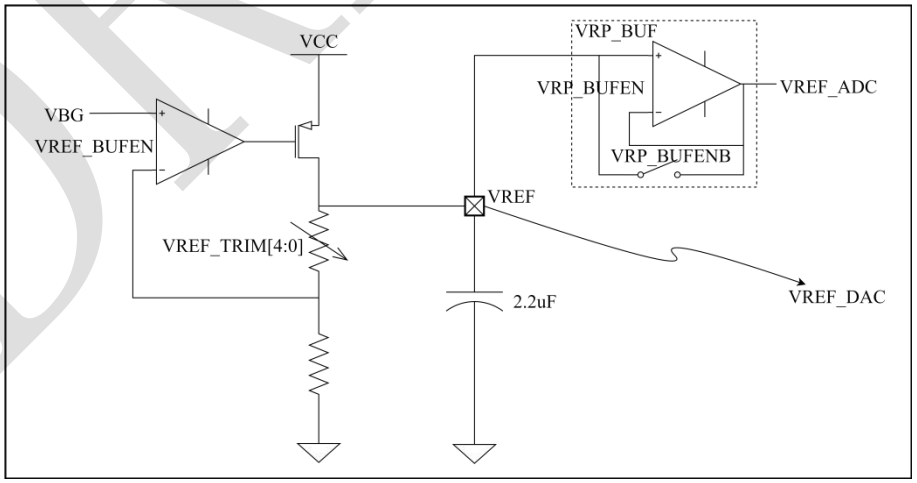


图 9-1 VREFBUF 结构框图

9.1.4 功能描述

VREFBUF 作为内部参考电压产生电路，负责给 ADC 和 DAC 提供稳定的低噪声参考电压源，并且具有一定的负载电流能力，总负载电流能力最大为 10mA，如果需要给外部其他模块提供参考电压，需要考虑内部负载电流情况，1 路 DAC 耗电最大 0.5mA 左右，使用时需要根据实际情况计算，保证参考能够工作在正常性能下。

VREFBUF 具有 5bit 内部电压校准寄存器，出厂会进行校准，将输出电压校准到 2.9V 和 2.5V。默认情况下 ATCR.VBFSEL=0，VREF+电压为 2.9V，会自动加载对应校准值。

实际参考电压和内部基准电压会经过芯片内部 Buffer 连接到 ADC 输入端，可以根据 ADC 实际转换值来推算 VREF 电压。内部基准电压值会在出厂测试的时候保存到 Flash 中，使用时通过 ADC 转换内部基准电压和外部参考即可计算出 VREF 电压。

假设基准电压为 VREFINT，ADC 参考为 VREF，ADC 对 VREFINT 转换后输出数据为 DINT，对 VREF 转换输出数据为 DEXT，可以计算得到 VREF 电压：

VREF=VREFINT*DEXT/DINT

9.2 寄存器定义

9.2.1 寄存器列表

SYSCTRL 基地址：

SYSCTRL(0x40021000)

偏移	实例地址	名称	默认值	描述
0x00	SYSCTRL 基地址+0x00	SYSCTRL_SYSDCR	0x00000003	SYSCTRL 系统调试配置寄存器
0x04	SYSCTRL 基地址+0x04	SYSCTRL_PHCR	0x00000000	SYSCTRL 外设配置寄存器
0x20	SYSCTRL 基地址+0x20	SYSCTRL_SYSATR	0x00000000	SYSCTRL 系统模拟配置寄存器
0x24	SYSCTRL 基地址+0x24	SYSCTRL_PWRCR	0x0000018A	SYSCTRL PMU 电源配置寄存器
0x30	SYSCTRL 基地址+0x30	SYSCTRL_PLCR	0x00000302	SYSCTRL 功率限制配置寄存器
0x34	SYSCTRL 基地址+0x34	SYSCTRL_PECR	0x00000020	SYSCTRL 事件控制寄存器
0x38	SYSCTRL 基地址+0x38	SYSCTRL_PSR	0x00000000	SYSCTRL 电源状态寄存器
0x3C	SYSCTRL 基地址+0x3C	SYSCTRL_PWRDR	0x00007717	SYSCTRL 模拟微调配置寄存器
0x40	SYSCTRL 基地址+0x40	SYSCTRL_CIDR	0x00025008	SYSCTRL ChipId 配置寄存器
0x44	SYSCTRL 基地址+0x44	SYSCTRL_KEYR	0x00000000	SYSCTRL 键寄存器
0x50	SYSCTRL 基地址+0x50	SYSCTRL_UID0	0xFFFFFFFF	SYSCTRL 系统用户 ID0 寄存器
0x54	SYSCTRL 基地址+0x54	SYSCTRL_UID1	0xFFFFFFFF	SYSCTRL 系统用户 ID1 寄存器
0x58	SYSCTRL 基地址+0x58	SYSCTRL_UID2	0xFFFFFFFF	SYSCTRL 系统用户 ID2 寄存器
0x5C	SYSCTRL 基地址+0x5C	SYSCTRL_UID3	0xFFFFFFFF	SYSCTRL 系统用户 ID3 寄存器
0x60	SYSCTRL 基地址+0x60	SYSCTRL_ATCR	0xFFF40820	SYSCTRL 模拟微调配置寄存器

9.2.2 寄存器描述

9.2.2.1 SYSCTRL 系统调试配置寄存器（SYSCTRL_SYSDCR）

- 名称：System Debug Config Register
- 偏移地址：0x00
- 默认值：0x00000003
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SEN	TMR10DEN	TMR9 DEN	TMR4 DEN	TMR3 DEN	TMR2 DEN	TMR1 DEN	TMR0 DEN	TMR8 DEN	TMR7 DEN	PWMD EN	WWD GDEN	IWDG DEN
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x1

字段	说明
[12] SEN	**Flash 双 Bank 映射切换使能(Flash Double Bank Switch Enable)** 1：使能 0：关闭
[11] TMR10DEN	**TMR10 调试使能(TMR10 Debug Enable)** 0：关闭 1：使能
[10] TMR9DEN	**TMR9 调试使能(TMR9 Debug Enable)** 0：关闭 1：使能
[9] TMR4DEN	**TMR4 调试使能(TMR4 Debug Enable)** 0：关闭 1：使能
[8] TMR3DEN	**TMR3 调试使能(TMR3 Debug Enable)** 0：关闭 1：使能
[7] TMR2DEN	**TMR2 调试使能(TMR2 Debug Enable)** 0：关闭 1：使能
[6] TMR1DEN	**TMR1 调试使能(TMR1 Debug Enable)** 0：关闭 1：使能
[5] TMR0DEN	**TMR0 调试使能(TMR0 Debug Enable)** 0：关闭 1：使能
[4] TMR8DEN	**TMR8 调试使能(TMR8 Debug Enable)** 0：关闭 1：使能
[3] TMR7DEN	**TMR7 调试使能(TMR7 Debug Enable)** 0：关闭 1：使能
[2] PWMDEN	**HRPWM 调试使能(HRPWM Debug Enable)** 0：关闭 1：使能
[1] WWDGDEN	**WWDG 调试使能(WWDG Debug Enable)** 0：关闭

	1：使能
[0] IWDGDEN	**IWDG 调试使能(IWDG Debug Enable)** 0：关闭 1：使能

9.2.2.2 SYSCTRL 外设配置寄存器（SYSCTRL_PHCR）

- **名称：Peripheral Config Register**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PF0FMEN	PC11FMEN	PC9FMEN	PC8FMEN	PC7FMEN	PC6FMEN	PC4FMEN	PB9FMEN	PB8FMEN	PB7FMEN	PB5FMEN	PA15FMEN	PA14FMEN	PA13FMEN	PA10FMEN	PA9FMEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[15] PF0FMEN	**PF0 FM+使能(PF0 FM+ Enable)** 0：关闭 1：使能
[14] PC11FMEN	**PC11 FM+使能(PC11 FM+ Enable)** 0：关闭 1：使能
[13] PC9FMEN	**PC9 FM+使能(PC9 FM+ Enable)** 0：关闭 1：使能
[12] PC8FMEN	**PC8 FM+使能(PC8 FM+ Enable)** 0：关闭 1：使能
[11] PC7FMEN	**PC7 FM+使能(PC7 FM+ Enable)** 0：关闭 1：使能
[10] PC6FMEN	**PC6 FM+使能(PC6 FM+ Enable)** 0：关闭 1：使能
[9] PC4FMEN	**PC4 FM+使能(PC4 FM+ Enable)** 0：关闭 1：使能
[8] PB9FMEN	**PB9 FM+使能(PB9 FM+ Enable)** 0：关闭 1：使能
[7] PB8FMEN	**PB8 FM+使能(PB8 FM+ Enable)** 0：关闭 1：使能
[6] PB7FMEN	**PB7 FM+使能(PB7 FM+ Enable)** 0：关闭 1：使能
[5] PB5FMEN	**PB5 FM+使能(PB5 FM+ Enable)** 0：关闭 1：使能
[4]	**PA15 FM+使能(PA15 FM+ Enable)**

PA15FMEN	0：关闭 1：使能
[3] PA14FMEN	**PA14 FM+使能(PA14 FM+ Enable)** 0：关闭 1：使能
[2] PA13FMEN	**PA13 FM+使能(PA13 FM+ Enable)** 0：关闭 1：使能
[1] PA10FMEN	**PA10 FM+使能(PA10 FM+ Enable)** 0：关闭 1：使能
[0] PA9FMEN	**PA9 FM+使能(PA9 FM+ Enable)** 0：关闭 1：使能

9.2.2.3 SYSCTRL 系统模拟配置寄存器（SYSCTRL_SYSATR）

- **名称：System Analog Config Register**
- **偏移地址：0x20**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABFSRC					ABFBYP	ABFEN								
	R/W					R/W	R/W								
	0x0					0x0	0x0								

字段	说明
[14:10] ABFSRC	**ADC Buffer 源选择(ADC Buffer Source Selection)** 0：AVSS，1：Temp SENSOR，2：VBGR，3：VR1P2， 4：VREFN，5：VREF，6：AVCC，7：AVDD 8~16：DAC0_OUT~DAC8_OUT，17：TOUT，18：PLL0_TOUT，19：Reserved 20~23：ADC0_IN[16]~ADC3_IN[16] 24：ADC0_VIN 25：ADC0_VIP 26：ADC1_VIN 27：ADC1_VIP 28：ADC2_VIN 29：ADC2_VIP 30：ADC3_VIN 31：ADC3_VIP
[9] ABFBYP	**ADC Buffer 旁路使能(ADC Buffer Bypass Enable)** 0：关闭 1：使能
[8] ABFEN	**ADC Buffer 使能(ADC Buffer Enable)** 0：关闭 1：使能

9.2.2.4 SYSCTRL PMU 电源配置寄存器（SYSCTRL_PWRCR）

- 名称：PMU Power Config Register
- 偏移地址：0x24
- 默认值：0x0000018A
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							AVDD DRD	VDDSET				AVDDSET		AVDD EN	TSE
							R/W	R/W				R/W		R/W	R/W
							0x1	0x8				0x2		0x1	0x0

字段	说明
[8] AVDDDRD	**AVDD 的无电容 LDO 下拉配置选择（AVDD Drop Down）** 0: 无下拉 1: 带有 0.5mA 下拉
[7:4] VDDSET	**VDD 电压控制(VDD Voltage Config)** Note: 默认为 1.2V，Step 为 40mv，调节范围为 0.88~1.48v
[3:2] AVDDSET	**AVDD 电压控制(AVDD Voltage Config)** 00: 1.0V 01: 1.1V 10: 1.2V 11: 1.3V Note: 默认为 1.2V
[1] AVDDEN	**AVDD 使能控制（AVDD Enable）** 0: 关闭 1: 使能
[0] TSE	**温度传感器使能控制（Temperature Sensor Enable）** 0: 关闭 1: 使能

9.2.2.5 SYSCTRL 功率限制配置寄存器（SYSCTRL_PLCR）

- 名称：Power Limit Config Register
- 偏移地址：0x30
- 默认值：0x00000302
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		VDDOCL		AVCCLVL		CPUL KEN	FLSEC CEN		VDDLVS			VDDO CE	AVCC LVE	VCCL VE	VDDL VE
		R/W		R/W		R/W	R/W		R/W			R/W	R/W	R/W	R/W
		0x0		0x0		0x1	0x1		0x0			0x0	0x0	0x1	0x0

字段	说明
[13:12] VDDOCL	**VDD 过流阈值设置(VDD OverCurrent Limit Config)** 0: 150mA 1: 175mA 2: 200mA 3: 225mA
[11:10] AVCCLVL	**AVCC 低电压阈值设置(AVCC LowVoltage Limit Config)** 0: 2.55V 1: 2.7V 2: 2.85V 3: 3V
[9]	**CPU Lockup 系统 Falut 使能(FLASH Lockup Falut Enable)**

CPULKEN	0: 关闭 1: 使能
[8] FLSECCEN	**FLASH 多 Bit ECC 错误系统 Falut 使能(FLASH Multi-Bit Error Falut Enable)** 0: 关闭 1: 使能
[6:4] VDDLVS	**VDD 低电压阈值设置(VDD LowVoltage Limit Config)** 0: 0.75V 1: 0.8V 2: 0.85V 3: 0.9V 4: 0.95V 5: 1.0V 6: 1.05V 7: 1.1V
[3] VDDOCE	**VDD 过流监测使能(VDD OverCurrent Enable)** 0: 关闭 1: 使能
[2] AVCCLVE	**AVCC 低电压监测使能(AVCC LowVoltage Enable)** 0: 关闭 1: 使能
[1] VCCLVE	**VCC 低电压监测使能(VCC LowVoltage Enable)** 0: 关闭 1: 使能
[0] VDDLVE	**VDD 低电压监测使能(VDD LowVoltage Enable)** 0: 关闭 1: 使能

9.2.2.6 SYSCTRL 事件控制寄存器（SYSCTRL_PECR）

- **名称：Power Event Control Register**
- **偏移地址：0x34**
- **默认值：0x00000020**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				VDDO CBE	AVCC LVBE	VCCL VBE	VDDL VBE	VDDO CRE	AVCC LVRE	VCCL VRE	VDDL VRE	VDDO CIE	AVCC LVIE	VCCL VIE	VDDL VIE
				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x0	0x0

字段	说明
[11] VDDOCBE	**VDD 过流刹车使能(VDD OverCurrent Brake Enable)** 0:正常 1:低电压
[10] AVCCLVBE	**AVCC 低电压刹车使能(AVCC LowVoltage Brake Enable)** 0:关闭 1:使能
[9] VCCLVBE	**VCC 低电压刹车使能(VCC LowVoltage Brake Enable)** 0:关闭 1:使能
[8] VDDLVBE	**VDD 低电压刹车使能(VDD LowVoltage Brake Enable)** 0:关闭 1:使能
[7] VDDOCRE	**VDD 过流复位使能(VDD OverCurrent Reset Enable)** 0:正常 1:低电压
[6] AVCCLVRE	**AVCC 低电压复位使能(AVCC LowVoltage Reset Enable)** 0:关闭 1:使能

[5] VCCLVRE	**VCC 低电压复位使能(VCC LowVoltage Reset Enable)** 0:关闭 1:使能
[4] VDDLVRE	**VDD 低电压复位使能(VDD LowVoltage Reset Enable)** 0:关闭 1:使能
[3] VDDOCIE	**VDD 过流中断使能(VDD OverCurrent Interrupt Enable)** 0:正常 1:低电压
[2] AVCCLVIE	**AVCC 低电压中断使能(AVCC LowVoltage Interrupt Enable)** 0:关闭 1:使能
[1] VCCLVIE	**VCC 低电压中断使能(VCC LowVoltage Interrupt Enable)** 0:关闭 1:使能
[0] VDDLVIE	**VDD 低电压中断使能(VDD LowVoltage Interrupt Enable)** 0:关闭 1:使能

9.2.2.7 SYSCTRL 电源状态寄存器（SYSCTRL_PSR）

- **名称：Power Status Register**
- **偏移地址：0x38**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												VDDOCS	AVCCLVS	VCCLVS	VDDLVS
												R	R	R	R
												0x0	0x0	0x0	0x0

字段	说明
[3] VDDOCS	**VDD 过流状态位(VDD OverCurrent Status)** 0:正常 1:低电压
[2] AVCCLVS	**AVCC 低电压状态位(AVCC LowVoltage Status)** 0:正常 1:低电压
[1] VCCLVS	**VCC 低电压状态位(VCC LowVoltage Status)** 0:正常 1:低电压
[0] VDDLVS	**VDD 低电压状态位(VDD LowVoltage Status)** 0:正常 1:低电压

9.2.2.8 SYSCTRL 模拟微调配置寄存器（SYSCTRL_PWRDR）

- 名称：**Power Debounce Register**
- 偏移地址：**0x3C**
- 默认值：**0x00007717**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDOCF				AVCCLVF				VCCLVF				VDDLVF			
R/W				R/W				R/W				R/W			
0x7				0x7				0x1				0x7			

字段	说明
[15:12] VDDOCF	**VDD 过流消抖配置(VDD OverCurrent Config)** 0:disable 1:1x0.5us … N:Nx0.5us
[11:8] AVCCLVF	**AVCC 欠压消抖配置(AVCC LowVoltage Config)** 0:disable 1:1x0.5us … N:Nx0.5us
[7:4] VCCLVF	**VCC 欠压消抖配置(VCC LowVoltage Config)** 0:disable 1:1x0.5us … N:Nx0.5us
[3:0] VDDLVF	**VDD 欠压消抖配置(VDD LowVoltage Config)** 0:disable 1:1x0.5us … N:Nx0.5us

9.2.2.9 SYSCTRL ChipId 配置寄存器（SYSCTRL_CIDR）

- 名称：**System ChipId Config Register**
- 偏移地址：**0x40**
- 默认值：**0x00025008**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								DCN							
								R							
								0x2							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CID															
R															
0x5008															

字段	说明
[23:16] DCN	**CHIP DCN**
[15:0] CID	**CHIP CID**

9.2.2.10 SYSCTRL 键寄存器（SYSCTRL_KEYR）

- 名称：LockKey Register
- 偏移地址：0x44
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														KST1	KST0
														R/W1C	R/W1C
														0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY															
R/W															
0x0															

字段	说明
[17] KST1	**校准寄存器保护状态位** 当对校准寄存器解锁完成后，KEYST1 标志位会置起，对该位写 1 操作进行加锁；
[16] KST0	**系统寄存器保护状态位** 当对系统寄存器解锁完成后，KEYST0 标志位会置起，对该位写 1 操作进行加锁；
[15:0] KEY	**寄存器写保护(Register LockKey)** 解锁： 1、系统寄存器解锁，向 KEY 寄存器写入 0x87e4 即可完成解锁，状态位 KST0 将由硬件置 1，然后即可进行相关的操作； 2、特殊寄存器解锁，向 KEY 寄存器写入 0x8a3d，然后再写入 0x19ec 进行解锁，状态位 KST1 将由硬件置 1，然后即可进行相关的操作； 加锁： 向状态位 KST0/1 寄存器写 1 即完成加锁，其状态位将清 0；

9.2.2.11 SYSCTRL 系统用户 ID0 寄存器（SYSCTRL_UID0）

- 名称：System UserID0 Register
- 偏移地址：0x50
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID0															
R															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID0															
R															
0xFFFFFFFF															

字段	说明
[31:0] UID0	**System UserID0 Register**

9.2.2.12 SYSCTRL 系统用户 ID1 寄存器（SYSCTRL_UID1）

- 名称：System UserID1 Register
- 偏移地址：0x54
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID1															
R															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID1															
R															
0xFFFFFFFF															

字段	说明
[31:0] UID1	**System UserID1 Register**

9.2.2.13 SYSCTRL 系统用户 ID2 寄存器（SYSCTRL_UID2）

- 名称：System UserID2 Register
- 偏移地址：0x58
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID2															
R															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID2															
R															
0xFFFFFFFF															

字段	说明
[31:0] UID2	**System UserID2 Register**

9.2.2.14 SYSCTRL 系统用户 ID3 寄存器（SYSCTRL_UID3）

- 名称：System UserID3 Register
- 偏移地址：0x5C
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID3															
R															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID3															
R															
0xFFFFFFFF															

字段	说明
[31:0] UID3	**System UserID3 Register**

9.2.2.15 SYSCTRL 模拟微调配置寄存器（SYSCTRL_ATCR）

- **名称：Analog Trim Config Register**
- **偏移地址：0x60**
- **默认值：0xFFF40820**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BGV													VBFTRIM		
R/W													R		
0xFFF													0x10		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBFTRIM		VBFSEL	VBFEN	REFITRIM						REFVTRIM					
R		R/W	R/W	R						R					
0x10		0x0	0x0	0x20						0x10					

字段	说明
[31:20] BGV	**内部 BGR 电压** 以 2.9V 为参考电压，将 BGR 输出电压转换为 12bit 数据存储，例如 BGV 为 0x46A，则得到实际 BGR 电压 VBGR 满足：VBGR=1130/212*2.9=0.8V
[18:14] VBFTRIM	**VREFBUF 输出电压校准** 00000：最大 11111：最小 Note：出厂校准参数；
[13] VBFSEL	**VREFBUF 输出电压选择** 0：2.9v 1：2.5v
[12] VBFEN	**VREFBUF 使能** 0: 关闭 1: 使能
[11:6] REFITRIM	**REF 参考电流校准** 000000：最小 111111：最大 Note：出厂校准参数；
[5:1] REFVTRIM	**REF 参考电压校准** 00000：最小 11111：最大 Note：出厂校准参数；

10 中断和事件

10.1 嵌套向量中断控制器（NVIC）

10.1.1 NVIC 特性

嵌套向量中断控制器 NVIC 包含以下特性：

- 芯片具有 85 个可屏蔽中断通道（不包括 Cortex™-M4 的 16 根中断线）
- 16 个可编程优先级（使用了 4 位中断优先级）
- 低延迟异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

嵌套向量中断控制器（NVIC）和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。

包括内核异常在内的所有中断均通过 NVIC 进行管理。更多关于异常和 NVIC 编程的说明，请参考《ARM Cortex™-M4 技术参考手册》中的第 10.1.3 节：中断和异常向量和第 10.1 节：嵌套向量中断控制器。

10.1.2 SysTick 校准值寄存器

SysTick 内部默认时钟源的频率为 HCLK，SysTick 外部时钟源的频率为 HCLK/8，当 SysTick 时钟源配置为外部时钟源 HCLK/8 时，为了产生 1 ms 时间基准，需要用户对校准值写入 20000。

10.1.3 中断和异常向量

请参见表 10-1，了解芯片的中断向量表。

表 10-1 芯片的中断向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000 0000
	-3	固定	Reset	复位	0x0000 0004
	-2	固定	NMI	不可屏蔽中断，RCU 时钟安全系统(CSS)和 FLASH 多 Bit ECC 错误连接到 NMI 向量。	0x0000 0008
	-1	固定	HardFault	所有类型的错误	0x0000 000C
	0	可设置	MemManage	存储器管理	0x0000 0010
	1	可设置	BusFault	预取指失败，存储器访问失败	0x0000 0014
	2	可设置	UsageFault	未定义的指令或非法状态	0x0000 0018

	-	-	-	保留	0x0000 001C - 0x0000 002B
	3	可设置	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
	4	可设置	Debug Monitor	调试监控器	0x0000 0030
	-	-	-	保留	0x0000 0034
	5	可设置	PendSV	可挂起的系统服务	0x0000 0038
	6	可设置	SysTick	SysTick 定时器	0x0000 003C
0	7	可设置	I2C0	I2C0 全局中断	0x0000 0040
1	8	可设置	I2C1	I2C1 全局中断	0x0000 0044
2	9	可设置	I2C2	I2C2 全局中断	0x0000 0048
3	10	可设置	UART0	UART0 全局中断	0x0000 004C
4	11	可设置	UART1	UART1 全局中断	0x0000 0050
5	12	可设置	UART2	UART2 全局中断	0x0000 0054
6	13	可设置	UART3	UART3 全局中断	0x0000 0058
7	14	可设置	UART4	UART4 全局中断	0x0000 005C
8	15	可设置	SPI0	SPI0 中断	0x0000 0060
9	16	可设置	SPI1	SPI1 中断	0x0000 0064
10	17	可设置	CAN0	CAN0 中断	0x0000 0068
11	18	可设置	CAN1	CAN1 中断	0x0000 006C
12	19	可设置	PDM0	PDM0 中断	0x0000 0070
13	20	可设置	PDM1	PDM1 中断	0x0000 0074
14	21	可设置	PDM2	PDM2 中断	0x0000 0078
15	22	可设置	PDM3	PDM3 中断	0x0000 007C
16	23	可设置	QEI0	QEI0 中断	0x0000 0080
17	24	可设置	QEI1	QEI1 中断	0x0000 0084
18	25	可设置	QEI2	QEI2 中断	0x0000 0088
19	26	可设置	DMA_CH0	DMA0 中断	0x0000 008C
20	27	可设置	DMA_CH1	DMA1 中断	0x0000 0090
21	28	可设置	DMA_CH2	DMA2 中断	0x0000 0094
22	29	可设置	DMA_CH3	DMA3 中断	0x0000 0098
23	30	可设置	DMA_CH4	DMA4 中断	0x0000 009C
24	31	可设置	DMA_CH5	DMA5 中断	0x0000 00A0
25	32	可设置	TMR7	TMR7 全局中断	0x0000 00A4
26	33	可设置	TMR8	TMR8 全局中断	0x0000 00A8
27	34	可设置	TMR0	TMR0 全局中断	0x0000 00AC
28	35	可设置	TMR1	TMR1 全局中断	0x0000 00B0
29	36	可设置	TMR2	TMR2 全局中断	0x0000 00B4
30	37	可设置	TMR3	TMR3 全局中断	0x0000 00B8
31	38	可设置	TMR4	TMR4 全局中断	0x0000 00BC
32	39	可设置	TMR9_BRK	TMR9 Break 中断	0x0000 00C0
33	40	可设置	TMR9_UPD	TMR9 Counter Update 中断	0x0000 00C4
34	41	可设置	TMR9_TRG	TMR9 Trigger 中断	0x0000 00C8
35	42	可设置	TMR9_CC	TMR9 Capture 及 Compare 中断	0x0000 00CC

36	43	可设置	TMR10_BRK	TMR10 Break 中断	0x0000 00D0
37	44	可设置	TMR10_UPD	TMR10 Counter Update 中断	0x0000 00D4
38	45	可设置	TMR10_TRG	TMR10 Trigger 中断	0x0000 00D8
39	46	可设置	TMR10_CC	TMR10 Capture 及 Compare 中断	0x0000 00DC
40	47	可设置	IWDG	独立看门狗中断	0x0000 00E0
41	48	可设置	WWDG	窗口看门狗中断	0x0000 00E4
42	49	可设置	GPIOA	GPIOA 中断	0x0000 00E8
43	50	可设置	GPIOB	GPIOB 中断	0x0000 00EC
44	51	可设置	GPIOC	GPIOC 中断	0x0000 00F0
45	52	可设置	GPIOD	GPIOD 中断	0x0000 00F4
46	53	可设置	GPIOE	GPIOE 中断	0x0000 00F8
47	54	可设置	GPIOF	GPIOF 中断	0x0000 00FC
48	55	可设置	FLASH	FLASH 全局中断	0x0000 0100
49	56	可设置	IIR	IIR 全局中断	0x0000 0104
50	57	可设置	CORDIC	CORDIC 中断	0x0000 0108
51	58	可设置	CMPG0	CMP0 中断	0x0000 010C
52	59	可设置	CMPG1	CMP1 中断	0x0000 0110
53	60	可设置	CMPG2	CMP2 中断	0x0000 0114
54	61	可设置	HRPWM_MST	HRPWM 主定时器中断	0x0000 0118
55	62	可设置	HRPWM_SLV0	HRPWM0 从定时器中断	0x0000 011C
56	63	可设置	HRPWM_SLV1	HRPWM1 从定时器中断	0x0000 0120
57	64	可设置	HRPWM_SLV2	HRPWM2 从定时器中断	0x0000 0124
58	65	可设置	HRPWM_SLV3	HRPWM3 从定时器中断	0x0000 0128
59	66	可设置	HRPWM_SLV4	HRPWM4 从定时器中断	0x0000 012C
60	67	可设置	HRPWM_SLV5	HRPWM5 从定时器中断	0x0000 0130
61	68	可设置	HRPWM_SLV6	HRPWM6 从定时器中断	0x0000 0134
62	69	可设置	HRPWM_SLV7	HRPWM7 从定时器中断	0x0000 0138
63	70	可设置	HRPWM_COM	HRPWM 中断	0x0000 013C
64	71	可设置	ADC0_NORM	ADC0 常规中断	0x0000 0140
65	72	可设置	ADC0_SAMP	ADC0 采样中断	0x0000 0144
66	73	可设置	ADC0_HALF	ADC0 DMA 半满中断	0x0000 0148
67	74	可设置	ADC0_FULL	ADC0 DMA 全满中断	0x0000 014C
68	75	可设置	ADC1_NORM	ADC1 常规中断	0x0000 0150
69	76	可设置	ADC1_SAMP	ADC1 采样中断	0x0000 0154
70	77	可设置	ADC1_HALF	ADC1 DMA 半满中断	0x0000 0158
71	78	可设置	ADC1_FULL	ADC1 DMA 全满中断	0x0000 015C
72	79	可设置	ADC2_NORM	ADC2 常规中断	0x0000 0160
73	80	可设置	ADC2_SAMP	ADC2 采样中断	0x0000 0164
74	81	可设置	ADC2_HALF	ADC2 DMA 半满中断	0x0000 0168
75	82	可设置	ADC2_FULL	ADC2 DMA 全满中断	0x0000 016C
76	83	可设置	ADC3_NORM	ADC3 常规中断	0x0000 0170
77	84	可设置	ADC3_SAMP	ADC3 采样中断	0x0000 0174
78	85	可设置	ADC3_HALF	ADC3 DMA 半满中断	0x0000 0178

79	86	可设置	ADC3_FULL	ADC3 DMA 全满中断	0x0000 017C
80	87	可设置	PMU	PMU 全局中断	0x0000 0180
81	88	可设置	USB_PWR	USB Power 中断	0x0000 0184
82	89	可设置	USB_DET	USB 插入拔出中断	0x0000 0188
83	90	可设置	USB_EP	USB 端点中断	0x0000 018C
84	91	可设置	XIF	XIF 中断	0x0000 0190

10.1.4 唤醒事件

芯片能够处理外部或内部事件来唤醒内核（WFE）。唤醒事件可通过以下方式产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时使能 Cortex™-M4 系统控制寄存器中的 SEVONPEND 位。当 CPU 从 WFE 恢复时，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部中断为事件模式，当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

11 DMA 控制器（DMA）

11.1 简介

直接存储器访问（DMA）用于在源端（外设/存储器）与目的端（外设/存储器）之间提供高速数据传输，可以在无需任何 CPU 操作的情况下通过 DMA 实现数据快速搬运，这样可以很大程度地节省 CPU 资源，减少中断进入次数，最终提高整体系统的性能。

DMA 控制器基于复杂的总线矩阵架构，DMA 控制器包含独立的 FIFO 和双 AHB 主总线架构，优化了系统带宽，有效地实现数据传输。

DMA 控制器有 6 个通道，每个通道总共可以有多达 32 个请求，6 个通道可以分配给一个或多个特定的外围设备进行数据传输，每个通道都有一个仲裁器，用于处理 DMA 请求间的优先级。

11.2 主要特性

DMA 主要具有以下特性：

- 包含 6 个通道，每个通道连接多个特定的外设请求
- 存储器和外围设备支持单一传输，4 拍、8 拍和 16 拍增量突发传输
- 支持外设 DMA：5 个 UART，3 个 I2C，2 个 SPI，4 个 PDM，1 个 XIF
- 当外围设备向存储器发送数据时支持改变存储器
- 支持对所有内部存储的 DMA 访问
- 支持软件优先级（低/中/高/超高）和硬件优先级（通道数越低，优先级越高）
- 存储器和外设的数据传输宽度可配置为：字节/半字/字
- 存储器和外设的数据传输支持固定和增量寻址
- 支持循环传输模式
- 支持单数据传输和多种数据传输方式
  - 多种数据传输方式：当存储器数据和外围数据的宽度不同时，自动打包/解包数据
  - 单数据传输模式：当且仅当 FIFO 为空时，数据从源地址读取，存储在 FIFO 中，FIFO 的数据写入目标地址。
- 每个通道具有独立的事件标志和中断

11.3 结构框图

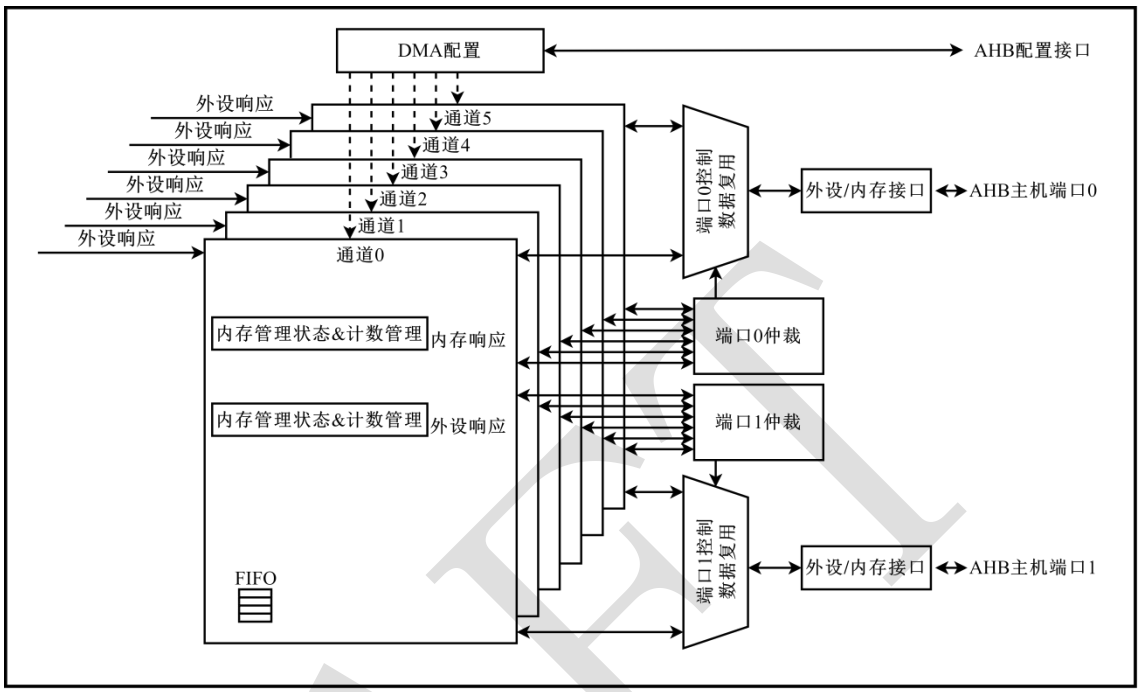


图 11-1　DMA 结构框图

11.4 功能描述

11.4.1 DMA 传输

DMA 控制器执行直接存储传输：DMA 传输接口采用 AHB 总线，可以控制 AHB 总线矩阵来启动 DMA 传输事务。

DMA 传输事务主要包括以下几项：

- 外设到存储器的传输
- 存储器到外设的传输
- 存储器到存储器的传输
- 外设到外设的传输

DMA 控制器提供两个 AHB 端口：可以访问存储器 SRAMA/B/C/D 和 AHB0/1 及 APB0/1 外设端口（连接外设及 SRAM）。

11.4.2 DMA 事务

一个 DMA 事务（Burst）由一串数据传输序列组成，需传输数据项的数目及宽度（8 位、16 位或 32 位）可由软件配置。

每个 DMA 传输都包含以下操作：

- 通过 SAR 寄存器寻址，从外设数据寄存器或存储器单元中加载数据；
- 通过 DAR 寄存器寻址，将加载的数据存储到外设数据存储器或存储器单元。

11.4.3 DMA 通道

DMA 模块共包含 6 个通道，软件可以自由选择通道作为 DMA 传输通道。

DMA 每个通道可软件自由配置源端/目标端的外设握手接口。通过配置相应通道的 DMA_CTRx 寄存器中的 SHF/DHF 位，选择源外设/目标外设的外设握手接口源。

表 11-1 给出了 DMA 握手接口映射。

表 11-1　DMA 握手接口映射表

DMA 握手接口 SHF/DHF 位	连接外设请求信号	说明
00000	I2C0_TX_REQ	-
00001	I2C0_RX_REQ	-
00010	I2C1_TX_REQ	-
00011	I2C1_RX_REQ	-
00100	I2C2_TX_REQ	-
00101	I2C2_RX_REQ	-
00110	UART0_TX_REQ	-
00111	UART0_RX_REQ	-
01000	UART1_TX_REQ	-
01001	UART1_RX_REQ	-
01010	UART2_TX_REQ	-
01011	UART2_RX_REQ	-
01100	UART3_TX_REQ	-
01101	UART3_RX_REQ	-
01110	UART4_TX_REQ	-
01111	UART4_RX_REQ	-
10000	SPI0_TX_REQ	-
10001	SPI0_RX_REQ	-
10010	SPI1_TX_REQ	-
10011	SPI1_RX_REQ	-
10101	XIF_RX_REQ	-
10111	PDM0_RX_REQ	-
11001	PDM1_RX_REQ	-
11011	PDM2_RX_REQ	-
11101	PDM3_RX_REQ	-

说明：如果源端/目标端配置的是 Memory，则该配置无效。

11.4.4 DMA 仲裁

仲裁器为两个 AHB 端口提供了基于优先级的 6 个通道请求的仲裁管理，并启动对外设/存储器的访问序列。

优先级管理分为两种：

- 软件：每个数据流优先级都可以在相应通道的 DMA_CTRx 寄存器中 PRI 位进行配置，分为 8 个级别：
  - 高优先级（0x7）
  - ……
  - 低优先级（0x0）
- 硬件：如果两个通道请求的 PRI 优先级位具有相同的优先级，则编号低的通道优先于编号高的通道，例如，通道 0 的优先级高于通道 1。

11.4.5 DMA 源、目标和传输模式

源传输和目标传输在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。

传输类型使用相应通道的 DMA_CTRx 寄存器中的 TC 位进行配置，有四种传输类型可以选择：存储器到外设、外设到存储器、存储器到存储器、外设到外设，下表介绍了四种传输类型。

表 11-2　传输类型

TC 位	类型
00	存储器到存储器（M2M）
01	存储器到外设（M2P）
10	外设到存储器（P2M）
11	外设到外设（P2P）

通过相应通道的 DMA_CTRx 寄存器中的 DDS/SDS 位进行配置）分别是半字或字时，写入该 DMA 通道的 DMA_SARx 或 DMA_DARx 寄存器的外设或存储器地址必须分别按字或半字地址的边界对齐。

存储器到存储器模式

当使能这种模式时，DMA 会根据内部 FIFO 的空间状态启动从 SRAM 到 FIFO 的数据传输。当达到 FIFO 的阈值级别时，会将 FIFO 的内容移出并通过总线写入到目标 SRAM 存储中。如果 DMA 使能位由软件清零或传输完成即会停止。

只有赢得了仲裁的通道才有权访问 AHB 的源或目标端口，用户可以通过配置相应通道的 DMA_CTRx 寄存器中 PRI 位来设置相应通道的优先级。

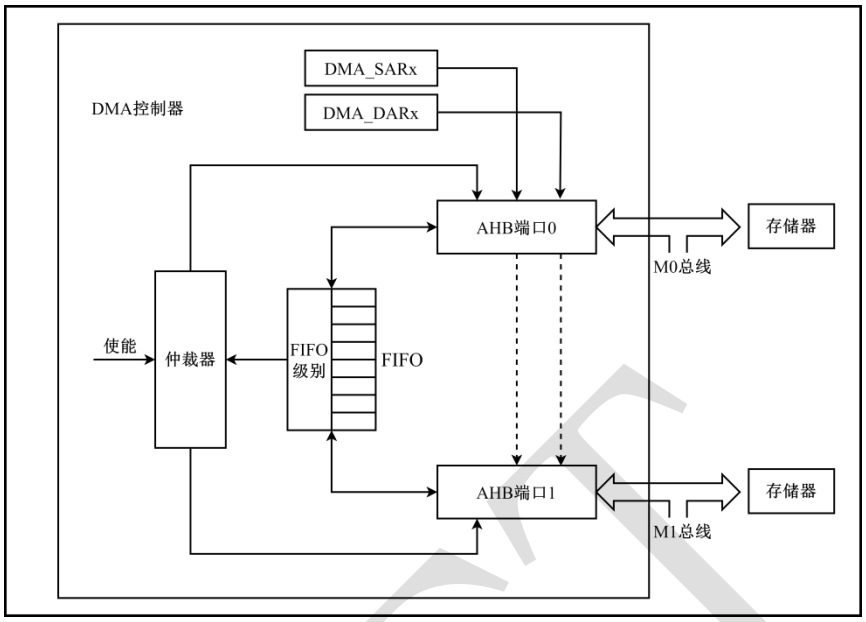


图 11-2　存储器到存储器模式示意图

外设到存储器模式

当使能这种模式时，每次产生外设请求，DMA 都会启动从外设到 DMA 内部 FIFO 的数据传输。当达到DMA 内部FIFO 的阈值级别时，会将FIFO 的内容移出并通过总线写入到目标SRAM 存储中。如果 DMA 使能位由软件清零或传输完成即会停止。

只有赢得了仲裁的通道才有权访问 AHB 的源或目标端口，用户可以通过配置相应通道的 DMA_CTRx 寄存器中 PRI 位来设置相应通道的优先级。

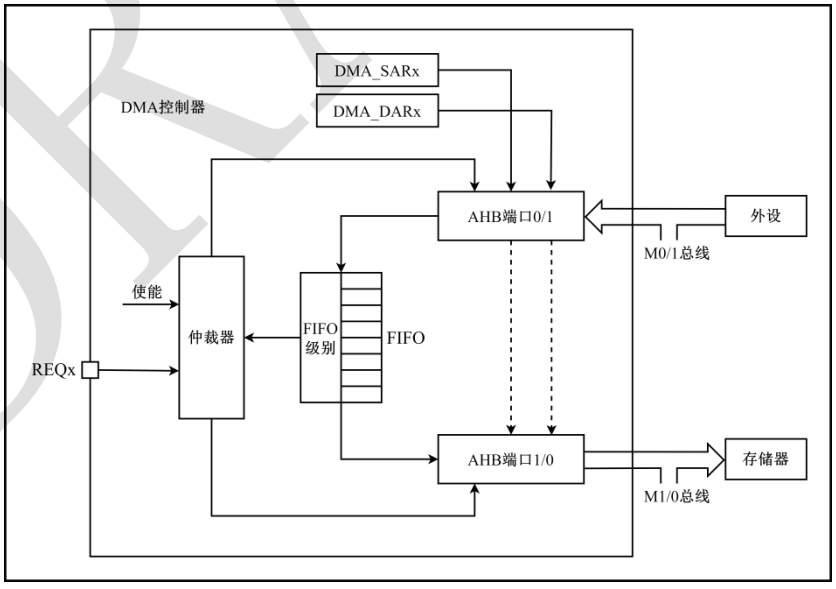


图 11-3　外设到存储器模式示意图

存储器到外设模式

当使能这种模式时，DMA 会根据内部 FIFO 的空间状态启动从 SRAM 到 FIFO 的数据传输。当达到 FIFO 的阈值级别时，会将 FIFO 的内容移出并通过总线写入到目标外设中。如果 DMA 使能位由软件清零或传输完成即会停止。

只有赢得了仲裁的通道才有权访问 AHB 的源或目标端口，用户可以通过配置相应通道的 DMA_CTRx 寄存器中 PRI 位来设置相应通道的优先级。

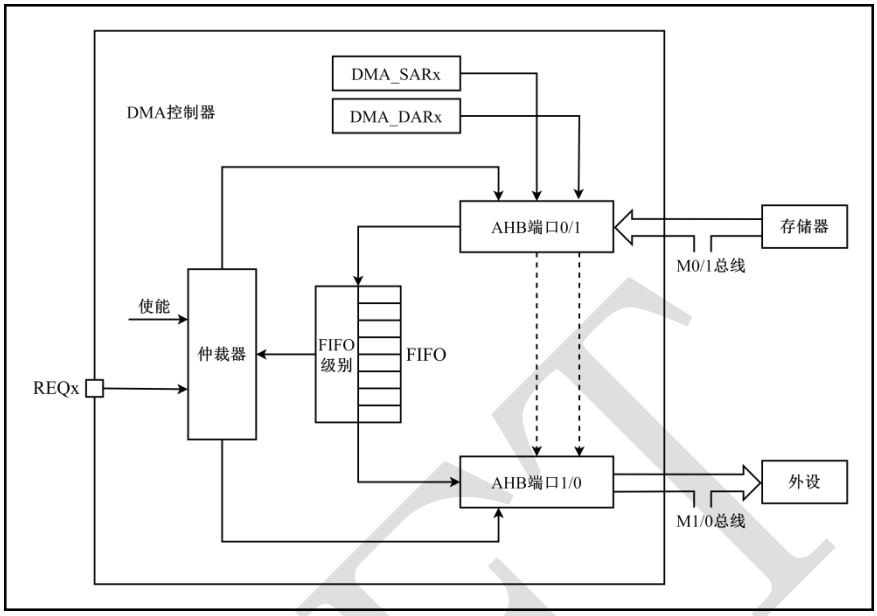


图 11-4　存储器到外设模式示意图

外设到外设模式

当使能这种模式时，每次产生外设请求，DMA 都会启动从外设到 DMA 内部 FIFO 的数据传输。当达到 DMA 内部 FIFO 的阈值级别时，会将 FIFO 的内容移出并通过总线写入到目标外设中。如果 DMA 使能位由软件清零或传输完成即会停止。

只有赢得了仲裁的通道才有权访问 AHB 的源或目标端口，用户可以通过配置相应通道的 DMA_CTRx 寄存器中 PRI 位来设置相应通道的优先级。

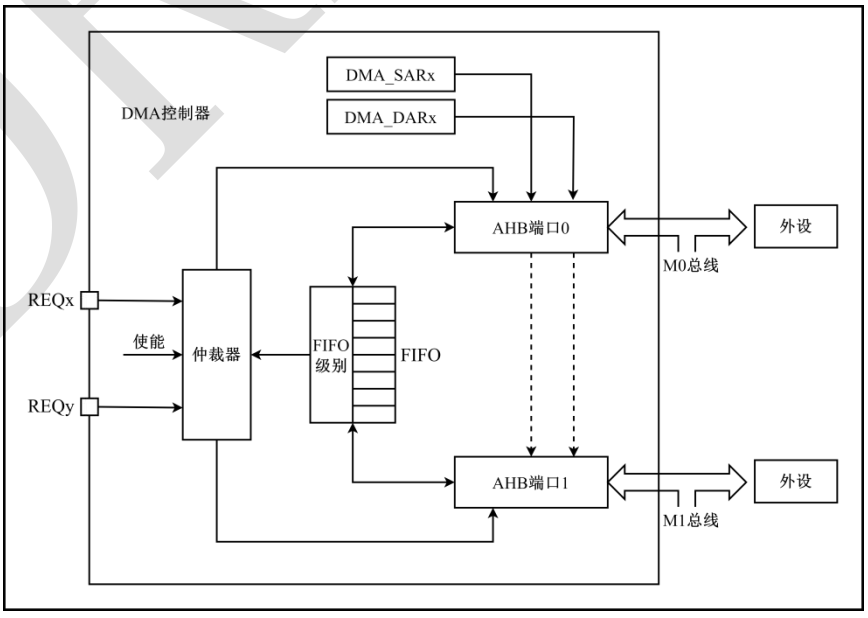


图 11-5　外设到外设模式示意图

11.4.6 DMA 地址控制

通过配置 DMA 通道的 DMA_CTRx 寄存器 SAI 和 DAI 位，来控制 DMA 的源/目的地址在每次传输后自动递增或保持不变，如果是通过单个寄存器访问外设源/目的数据时，可以关闭递增模式。

如果使能了递增模式，则根据 SDS/DDS 位配置的数据宽度，下一次传输的地址将在前一次传输的地址基础上递增 1（对于字节）、2（对于半字）或 4（对于字）。

为了优化封装操作，可以不管 AHB 外设端口数据位宽，将外设地址的增量偏移大小固定下来。例如通过配置 DMA_CTRx 寄存器中的 SDS/DDS 位，将地址的增量偏移大小与外设 AHB 端口固定为 32 位地址，此时地址递增 4，需要确保地址按 32 位数据大小对齐。

DMA 内部的 FIFO 用于存储在源端传输到目的端之前的临时数据，每个通道都有一个独立的 FIFO，FIFO 的阈值级别由 DMA_CTRx 寄存器中的 SDS/DDS 位来控制，当 FIFO 中数据达到一笔传输数量时，即可启动传输。

11.4.7 DMA 中断

DMA 中每个通道都有 3 种类型的状态/中断：

- Transfer Completed：DMA 完成所有数据传输后，产生传输完成标志，挂起相应的中断请求标志，并关闭 DMA 的通道使能。
- Transfer Half Completed：DMA 完成一半数据传输，产生传输一半完成标志，并挂起相应的中断请求标志。
- Transfer Error：在传输过程中由于传输出现异常（例如外设总线错误）或地址不对齐，产生传输错误标志，挂起相应的请求标志并且取消 DMA 传输、关闭通道使能。

11.5 寄存器定义

11.5.1 寄存器列表

DMA 基地址：
DMA0(0x40022000) DMA1(0x40022020) DMA2(0x40022040)
DMA3(0x40022060) DMA4(0x40022080) DMA5(0x400220A0)

偏移	实例地址	名称	默认值	描述
0x00	DMA 基地址+0x00+N*0x20[N=0-5]	DMA_SARx	0x00000000	DMA 源地址寄存器
0x04	DMA 基地址+0x04+N*0x20[N=0-5]	DMA_DARx	0x00000000	DMA 目标地址寄存器
0x08	DMA 基地址+0x08+N*0x20[N=0-5]	DMA_DBLx	0x00000000	DMA 传输数量长度寄存器
0x0C	DMA 基地址+0x0C+N*0x20[N=0-5]	DMA_CTRx	0x00000000	DMA 控制寄存器
0x10	DMA 基地址+0x10+N*0x20[N=0-5]	DMA_CERx	0x00000000	DMA 传输使能寄存器
0x14	DMA 基地址+0x14+N*0x20[N=0-5]	DMA_STRx	0x00000000	DMA 状态寄存器

11.5.2 寄存器描述

11.5.2.1 DMA 源地址寄存器（DMA_SARx）

- 名称：DMA Source Address Register
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAR															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR															
R/W															
0x0															

字段	说明
[31:0] SAR	**DMA 传输的源地址** Note: 1、DMA 传输地址是否进行递增、或保持不变由寄存器 CTR0 寄存器中 SAI 位决定； 2、SAR 地址必须与 CTR0 寄存器中 SDS 的位宽保持对齐； 3、DMA 通道使能由软件配置源地址。

11.5.2.2 DMA 目标地址寄存器（DMA_DARx）

- 名称：DMA Destination Address Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAR															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR															
R/W															
0x0															

字段	说明
[31:0] DAR	**DMA 传输的目的地址** Note: 1、DMA 传输地址是否进行递增、递减或者保持不变由 CTR0 寄存器中 DAI 位位决定； 2、DAR 地址必须与 CTR0 寄存器中 DDS 的位宽保持对齐； 3、DMA 通道使能由软件配置源地址。

11.5.2.3 DMA 传输数量长度寄存器（DMA_DBLx）

- 名称：DMA Block Length Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				BL											
				R/W											
				0x0											

字段	说明
[11:0] BL	**DMA 传输数量长度** Note：总数据量=BL*DataSize；

11.5.2.4 DMA 控制寄存器（DMA_CTRx）

- 名称：DMA Control Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMS	SMS	DBL		SBL		DHF					SHF				
R/W	R/W	R/W		R/W		R/W					R/W				
0x0	0x0	0x0		0x0		0x0					0x0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDS		SDS			CIE	HCIE	EIE	DAI	SAI	TC		TM	PRI		
R/W		R/W			R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W		
0x0		0x0			0x0	0x0	0x0	0x0	0x0	0x0		0x0	0x0		

字段	说明
[31] DMS	**DMA 目的端设备接口控制位（DMA Destination Master Select）** 参考目标设备所在的总线； 0: AHB Master1 1: AHB Master2
[30] SMS	**DMA 源端设备接口控制位（DMA Source Master Select）** 参考目标设备所在的总线； 0: AHB Master1 1: AHB Master2
[29:28] DBL	**DMA 目的端 Burst 长度设置（DMA Destination Burst Length）** 0: 1 1: 4 2: 8 3: 16
[27:26] SBL	**DMA 源端 Burst 长度设置（DMA Source Burst Length）** 0: 1 1: 4 2: 8 3: 16
[25:21] DHF	**DMA 目的端硬件握手接口索引（DMA Destination Handshake Index）** 参考 SHF 源端索引描述
[20:16]	**DMA 源端硬件握手接口索引（DMA Source Handshake Index）**

SHF	0:i2c0Tx 1:i2c0Rx 2:i2c1Tx 3:i2c1Rx 4:i2c2Tx 5:i2c2Rx 6:uart0Tx 7:uart0Rx 8:uart1Tx 9:uart1Rx 10:uart2Tx 11:uart2Rx 12:uart3Tx 13:uart3Rx 14:uart4Tx 15:uart4Rx 16:spi0Tx 17:spi0Rx 18:spi1Tx 19:spi1Rx 21:xifRx 23:pdm0Rx 25:pdm1Rx 27:pdm2Rx 29:pdm3Rx
[15:14] DDS	**DMA 目的端传输位宽选择（DMA Destination Data Width Select）** 0: Byte 1: HalfWord 2: Word x: Reserved
[13:12] SDS	**DMA 源端传输位宽选择（DMA Source Data Width Select）** 0: Byte 1: HalfWord 2: Word 3: Reserved Note:单次传输数据长度为 BL*SDS；
[10] CIE	**DMA 传输完成中断使能位（DMA Transfer Complete Interrupt Enable）** 0:不使能 1:使能
[9] HCIE	**DMA 传输一半完成中断使能位（DMA Transfer half Completed Interrupt Enable）** 0:不使能 1:使能
[8] EIE	**DMA 错误中断使能位（DMA Error Interrupt Enable）** 0:不使能 1:使能
[7] DAI	**DMA 目的端地址控制位（DMA Destination Address Control）** 0:递增 1:不变 Note: 该位指示每次传输完成时目的地址是否递增或者保持不变；如果设备正在向固定地址或者外设 FIFO 写数据，此字段设置为“不变”。
[6] SAI	**DMA 源端地址控制位（DMA Source Address Control）** 0:递增 1:不变 Note: 该位指示每次传输完成时源地址是否递增或者保持不变；如果设备正在向固定地址或者外设 FIFO 取数据，此字段设置为“不变”。
[5:4] TC	**DMA 传输类型控制（DMA Transfer Type Control）** 0:Memory 到 Memory 1:Memory 到外设 2:外设到 Memory 3:外设到外设
[3] TM	**DMA 传输模式控制（DMA Transfer Mode Control）** 0:单次传输模式 1:连续传输模式
[2:0] PRI	**DMA 通道优先级设置（DMA Channel Priority Control）(根据通道数目)** 0: 低优先级 7: 高优先级 Note: 1) 7 是最高优先级，0 是最低优先级。 2) 复位值是通道值；例如，通道 0 的情况下复位值为 0；通道 1 的情况下复位值为 1。

11.5.2.5 DMA 传输使能寄存器（DMA_CERx）

- 名称：DMA Channel Enable Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															CEN
															R/W
															0x0

字段	说明
[0] CEN	**DMA 传输使能（DMA Enable）** 0:关闭 1:使能

11.5.2.6 DMA 状态寄存器（DMA_STRx）

- 名称：DMA Status Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CS	HS	ES
													R/W1C	R/W1C	R/W1C
													0x0	0x0	0x0

字段	说明
[2] CS	**DMA 传输完成中断状态位（DMA Transfer Completed Interrupt Status）** 0: DMA 传输未完成 1: DMA 传输完成
[1] HS	**DMA 传输一半完成中断状态位（DMA Transfer Half Completed Interrupt Status）** 0: DMA 传输一半未完成 1: DMA 传输一半完成
[0] ES	**DMA 错误中断状态位（DMA Error Interrupt Status）** 0: DMA 未出现错误 1: DMA 出现错误

12 CORDIC 运算单元（CORDIC）

12.1 简介

CORDIC 是一种有效率的逼近算法，用于计算三角函数和双曲函数。它可分解为一系列加减和移位操作，故非常适合硬件实现。与软件实现相比，它加快了这些函数的计算速度，释放处理器以执行其他任务。

12.2 主要特性

CORDIC 主要有以下特性：

- 24 位 CORDIC 计算引擎
- 支持圆周系统和双曲系统
- 支持旋转模式和向量模式
- 支持函数：sine、cosine、sinh、cosh、atan、atan2、atanh、modulus（模数）、square root（平方根）、natural logarithm（自然数对数）
- 支持零等待模式（结果无需轮询或中断就可读取）
- 支持两套独立的 CORDIC 控制单元
- 支持 AHB 从接口

12.3 结构框图

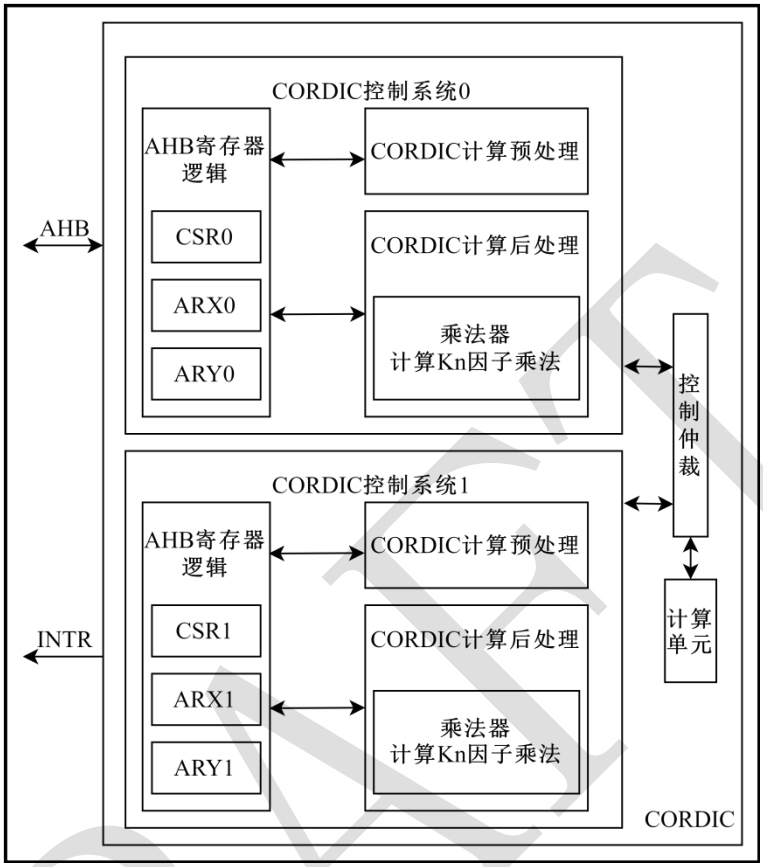


图 12-1 CORDIC 结构框图

12.4 功能描述

12.4.1 CORDIC 描述

CORDIC 是一种经济高效的逐次逼近算法，圆周系统可以被用来计算三角函数，而双曲系统可以被用来计算双曲函数。

在圆周系统下，角度θ的正弦和余弦是通过旋转单位向量逼近角度，直到旋转角度的累积和等于输入角度θ。旋转矢量的 x 和 y 则分别对应于θ的余弦和正弦。而对于反正切(y/x)，是通过旋转[x,y]依次递减角度，得到单位向量[1,0]。旋转角度的累积和就是原始矢量的角度。

CORDIC 算法也可用于计算双曲线函数（sinh、cosh、atanh），方法是通过沿双曲线逐步替换连续的圆周旋转。
其他的函数可以通过上述基本函数派生。

12.4.2 CORDIC 函数

可以通过配置 CORDIC_CSRx.FUNC 来确定处理器运算的函数，具体如表 12-1 所示：

表 12-1 CORDIC 函数

函数	第一个输入 ARG1	第二个输入 ARG2	第一个输出 RES1	第二个输出 RES1
余弦函数（Cosine）	角度θ	-	$\cos\theta$	$\sin\theta$
正弦函数（Sine）	角度θ	-	$\sin\theta$	$\cos\theta$
相角函数（Phase）	x	y	$\tan^{-1}\frac{y}{x}$	$\sqrt{x^2+y^2}$
取模函数（Modulus）	x	y	$\sqrt{x^2+y^2}$	$\tan^{-1}\frac{y}{x}$
反正切函数（Arctangent）	x	-	$\tan^{-1}x$	-
双曲余弦函数（Cosh）	x	-	$\cosh x$	$\sinh x$
双曲正弦函数（Sinh）	x	-	$\sinh x$	$\cosh x$
双曲反正切函数（Atanh）	x	-	$\tanh^{-1}x$	-
自然数对数函数（Ln）	x	-	$\ln x$	-
开平方根函数（Sqrt）	x	-	$\sqrt{x}$	-

1) 有些函数需要两个输入，ARG1 代表第一个输入，ARG2 代表第二个输入；
2) 有些函数产生两个输出，RES1 代表第一个输出，RES2 代表第二个输出。这意味着，有时候只需要一个操作就可以获得两个值。比如算 sin 的时候，也会产生 cos 的值，双曲函数也是如此

12.4.2.1 余弦函数（Cosine）

表 12-2 余弦函数（Cosine）

参数	描述	取值范围
ARG1	以弧度表示的角度θ，除以π	[-1,1)
ARG2	-	-
RES1	$\cos\theta$	[-1,1)
RES2	$\sin\theta$	[-1,1)
SCALE	不适用	0

1) 对于 ARG1，需要输入以弧度为单位的θ，同时还需要除以π，还要满足 1.31 格式输入
2) RES1 是角度的余弦，RES2 是角度的正弦
3) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-19}以内；如果是 q1.15 输出，精度在2^{-15}以内

12.4.2.2 正弦函数（Sine）

表 12-3 正弦函数（Sine）

参数	描述	取值范围
ARG1	以弧度表示的角度θ，除以π	[-1,1)
ARG2	-	-
RES1	$\sin\theta$	[-1,1)
RES2	$\cos\theta$	[-1,1)
SCALE	不适用	0

1) 对于 ARG1，需要输入以弧度为单位的θ，同时还需要除以π，还要满足 1.31 格式输入
2) RES1 是角度的正弦，RES2 是角度的余弦
3) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-19}以内；如果是 q1.15 输出，精度在2^{-15}以内

12.4.2.3 相角函数（Phase）

表 12-4 相角函数（Phase）

参数	描述	取值范围
ARG1	x 坐标系值	[-1,1)
ARG2	y 坐标系值	[-1,1)
RES1	以弧度表示的角度θ，除以π	[-1,1)
RES2	取模值	[0,1]
SCALE	不适用	0

1) 该函数计算的是$\tan^{-1}\frac{y}{x}$
2) 对于 ARG1，需要输入 x 的坐标，也就是向量在 x 轴方向的大小，如果 x 的值大于等于 1 或者小于-1，则用户需要做相应的缩放，使其范围在在[-1,1)，此外还要满足 1.31 格式输入
3) 对于 ARG2，需要输入 y 的坐标，也就是向量在 y 轴方向的大小，如果 y 的值大于等于 1 或者小于-1，则用户需要做相应的缩放，使其范围在[-1,1)，此外还要满足 1.31 格式输入
4) RES1 是相角θ/π，如果需要弧度值，用户需要将 RES1 的值乘以π；RES2 是$\sqrt{x^2+y^2}$
5) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-19}以内；如果是 q1.15 输出，精度在2^{-15}以内

12.4.2.4 取模函数（Modulus）

表 12-5 取模函数（Modulus）

参数	描述	取值范围
ARG1	x 坐标系值	[-1,1)
ARG2	y 坐标系值	[-1,1)
RES1	取模值	[0,1]
RES2	以弧度表示的角度θ，除以π	[-1,1)
SCALE	不适用	0

1) 该函数计算的是$\sqrt{x^2+y^2}$
2) 对于 ARG1，需要输入 x 的坐标，也就是向量在 x 轴方向的大小，如果 x 的值大于等于 1 或者小于-1，则用户需要做相应的缩放，使其范围在[-1,1)，此外还要满足 1.31 格式输入
3) 对于 ARG2，需要输入 y 的坐标，也就是向量在 y 轴方向的大小，如果 y 的值大于等于 1 或者小于-1，则用户需要做相应的缩放，使其范围在[-1,1)，此外还要满足 1.31 格式输入
4) RES1 是$\tan^{-1}\frac{y}{x}$；RES1 是相角θ/π，如果需要弧度值，用户需要将 RES1 的值乘以π
5) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-19}以内；如果是 q1.15 输出，精度在2^{-15}以内

12.4.2.5 反正切函数（Arctangent）

表 12-6 反正切函数（Arctangent）

参数	描述	取值范围
ARG1	$x\cdot 2^{-n}$	[-1,1)
ARG2	-	-
RES1	$2^{-n}\cdot\tan^{-1}x$的弧度值，除以π	[-1,1)
RES2	-	-
SCALE	n	[0,7]

1) 该函数计算的是$\tan^{-1}x$
2) 对于 ARG1，需要输入 x 的坐标，也就是向量在 x 轴方向的大小；对于 ARG2，该函数不需要第二个输入，因此不需要配置。
3) 如果 x 的值大于 1，则必须配置 CORDIC_CSRx.SCALE 将$x\cdot 2^{-n}$的值控制在[-1,1)范围内，缩放后软件还需要对 RES1 进行处理才能得到真实的结果，具体如下表所示：

表 12-7 反正切函数 SCALE 取值表

x 的取值范围	SCALE 取值	$\tan^{-1}x$的值	q1.31 精度	q1.15 精度
[-1,1)	0	$RES1\cdot\pi\cdot 2^{0}$	2^{-19}	2^{-15}
[-2,2)	1	$RES1\cdot\pi\cdot 2^{1}$	2^{-19}	2^{-14}
[-4,4)	2	$RES1\cdot\pi\cdot 2^{2}$	2^{-19}	2^{-13}
[-8,8)	3	$RES1\cdot\pi\cdot 2^{3}$	2^{-19}	2^{-12}
[-16,16)	4	$RES1\cdot\pi\cdot 2^{4}$	2^{-19}	2^{-11}
[-32,32)	5	$RES1\cdot\pi\cdot 2^{5}$	2^{-19}	2^{-10}
[-64,64)	6	$RES1\cdot\pi\cdot 2^{6}$	2^{-19}	2^{-9}
[-128,128)	7	$RES1\cdot\pi\cdot 2^{7}$	2^{-19}	2^{-8}

12.4.2.6 双曲余弦函数（Cosh）

表 12-8 双曲余弦函数（Cosh）

参数	描述	取值范围
ARG1	$x\cdot 2^{-n}$	[-0.559,0.559]
ARG2	-	-
RES1	$2^{-n}\cdot\cosh x$	[0.5,0.846]
RES2	$2^{-n}\cdot\sinh x$	[-0.683,0.683]
SCALE	n	1

1) 该函数计算的是$\cosh x$
2) 对于 ARG1，需要输入双曲角度 x，x 的范围是[-1.118,1.118]，超出了 1.31 能表示的范围，因此在运行双曲余弦函数时，必须将 CORDIC_CSRx.SCALE 配置为 1；对于 ARG2，该函数不需要第二个输入，因此不需要配置。
3) RES1 的结果是$2^{-n}\cdot\cosh x$，其中 n 固定为 1，因此软件如果要获取$\cosh x$的真实值，只需要将获取的 RES1 值乘以 2 即可。
4) RES2 的结果是$2^{-n}\cdot\sinh x$，其中 n 固定为 1，因此软件如果要获取$\sinh x$的真实值，只需要将获取的 RES2 值乘以 2 即可。
5) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-18}以内；如果是 q1.15 输出，精度在2^{-14}以内

12.4.2.7 双曲正弦函数（Sinh）

表 12-9 双曲正弦函数（Sinh）

参数	描述	取值范围
ARG1	$x\cdot 2^{-n}$	[-0.559,0.559]
ARG2	-	-
RES1	$2^{-n}\cdot\sinh x$	[-0.683,0.683]
RES2	$2^{-n}\cdot\cosh x$	[0.5,0.846]
SCALE	n	1

1) 该函数计算的是$\sinh x$
2) 对于 ARG1，需要输入双曲角度 x，x 的范围是[-1.118,1.118]，超出了 1.31 能表示的范围，因此在运行双曲正弦函数时，必须将 CORDIC_CSRx.SCALE 配置为 1；对于 ARG2，该函数不需要第二个输入，因此不需要配置。
3) RES1 的结果是$2^{-n}\cdot\sinh x$，其中 n 固定为 1，因此软件如果要获取$\sinh x$的真实值，只需要将获取的 RES1 值乘以 2 即可。
4) RES2 的结果是$2^{-n}\cdot\cosh x$，其中 n 固定为 1，因此软件如果要获取$\cosh x$的真实值，只需要将获取的 RES2 值乘以 2 即可。
5) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-18}以内；如果是 q1.15 输出，精度在2^{-14}以内

12.4.2.8 双曲反正切函数（Atanh）

表 12-10 双曲反正切函数（Atanh）

参数	描述	取值范围
ARG1	$x\cdot 2^{-n}$	[-0.403,0.403]
ARG2	-	-
RES1	$2^{-n}\cdot\tanh^{-1}x$	[-0.559,0.559]
RES2	-	-
SCALE	n	1

1) 该函数计算的是$\tanh^{-1}x$
2) 对于 ARG1，需要输入双曲角度 x，原本 x 的范围是[-0.806,0.806]，但是该范围计算出来的结果范围在[-1.118,1.118]，超出了 1.31 能表示的范围，因此在运行双曲反正切函数时，必须将 CORDIC_CSRx.SCALE 配置为 1；对于 ARG2，该函数不需要第二个输入，因此不需要配置。

3) cordic 计算过程迭代 24 次，如果是 q1.31 输出，精度在2^{-18}以内；如果是 q1.15 输出，精度在2^{-14}以内

12.4.2.9 自然数对数函数（Ln）

表 12-11 自然数对数函数（Ln）

参数	描述	取值范围
ARG1	$x \cdot 2^{-n}$	[0.054,0.875]
ARG2	-	-
RES1	$2^{-(n+1)} \cdot \ln x$	[-0.279,0.137]
RES2	-	-
SCALE	n	[1,4]

1) 该函数计算的是$\ln x$
2) 输入的 ARG1 的范围必须在[0.054,0.875]，如果超出了这个范围，则必须配置 CORDIC_CSRx.SCALE，缩放后软件还需要对 RES1 进行处理才能得到真实的结果，具体如下表所示：

表 12-12 自然数对数函数 SCALE 取值表

x 的取值范围	SCALE 取值	ARG1 的值	q1.31 精度	q1.15 精度
[0.107,1)	1	[0.0535,0.5)	2^{-18}	2^{-14}
[1,3)	2	[0.25,0.75)	2^{-18}	2^{-13}
[3,7)	3	[0.375,0.875)	2^{-18}	2^{-12}
[7,9.35]	4	[0.4375,0.584)	2^{-18}	2^{-11}

12.4.2.10 开平方根函数（Sqrt）

表 12-13 开方根函数（Sqrt）

参数	描述	取值范围
ARG1	$x \cdot 2^{-n}$	[0.027,0.875]
ARG2	-	-
RES1	$2^{-n} \cdot \sqrt{x}$	[0.04,1]
RES2	-	-
SCALE	n	[0,2]

1) 该函数计算的是$\ln x$
2) 输入的 ARG1 的范围必须在[0.027,0.875]，如果超出了这个范围，则必须配置 CORDIC_CSRx.SCALE，缩放后软件还需要对 RES1 进行处理才能得到真实的结果，具体如下表所示：

表 12-14 开平方根函数 SCALE 取值表

x 的取值范围	SCALE 取值	ARG1 的值	q1.31 精度	q1.15 精度
[0.027,0.75)	0	[0.027,0.75)	2^{-18}	2^{-15}
[0.75,1.75)	1	[0.375,0.875)	2^{-18}	2^{-14}
[1.75,2.341]	2	[0.4375,0.585)	2^{-18}	2^{-13}

12.4.3 CORDIC 定点

CORDIC 是以定点有整数格式运行，输入和输出值可以是 q1.31 或者 q1.15，不同的输入对应不同的精度，具体每个函数的精度可以查看 CORDIC 函数章节。

在 q1.31 格式中，数字由 1 个符号位和 31 个小数位组成，数值范围为-1(0x80000000)到 $1-2^{-31}$(0x7FFFFFFF)。

在 q1.15 格式中，数字由 1 个符号位和 15 个小数位组成，数值范围为-1(0x8000)到 $1-2^{-15}$(0x7FFF)。

此外还有比例因子(CORDIC_CSRx.SCALE)会影响输入输出的数值范围，CORDIC 函数章节列举了各个函数是否支持比例因子 SCALE，配置了比例因子可以扩展函数输入范围，以覆盖 CORDIC 所需的取值范围，且不会使得计算单元饱和。

CORDIC 定义了输入输出数据位宽可以为 16 位也可以为 32 位，根据不同的函数也定义了输入输出数据的个数，具体的配置如下表所示：

表 12-15 CORDIC 数据配置表

ARGSIZE=0		ARGSIZE=1
NARGS=0	NARGS=1	NARGS=0
ARX 写一个数启动计算	ARX 先写第一个数 ARY 写第二个数启动计算	ARX 写一个数启动计算
RESSIZE=0		RESSIZE=1
NRES=0	NRES=1	NRES=0
ARX 读一个数结束计算	ARX 先读第一个数 ARY 读第二个数结束计算	ARX 读一个数结束计算

注：

1. 在 ARGSIZE=1 时，NARGS 不可配置为 1。若只有一个输入数据，那么将其放置在 ARX 中的低 16 位；如果有两个数据，那么第一个数据放置在 ARX 中的低 16 位，第二个数据放置在 ARX 中的高 16 位。
2. 在 RESSIZE=1 时，NRES 不可配置为 1。若只有一个输出结果，该数据在 ARX 中的低 16 位；如果有两个数据，第一个数据在 ARX 中的低 16 位，第二个数据在 ARX 中的高 16 位。

12.4.4 CORDIC 操作

12.4.4.1 操作模式选择

零等待模式

使用 CORDIC 计算，最快的方式就是零等待模式。具体步骤如下：

1. 配置 CORDIC_CSRx 寄存器
   a) 配置 CORDIC_CSRx.FUNC 寄存器位确定接下来需要计算的函数。
   b) 配置 CORDIC_CSRx.SCALE 寄存器位确定比例因子
   c) 配置 CORDIC_CSRx.NRES 寄存器位确定输出数据量
   d) 配置 CORDIC_CSRx.NARGS 寄存器位确定输入数据量
   e) 配置 CORDIC_CSRx.RESSIZE 寄存器位确定输出的位宽
   f) 配置 CORDIC_CSRx.ARGSIZE 寄存器位确定输入的位宽
2. 根据 CORDIC_CSRx.NARGS 的值，对 ARXx 和 ARYx 寄存器写入需要计算的数据
3. 根据 CORDIC_CSRx.NRES 的值，读取 ARXx 和 ARYx 寄存器获取计算结果
4. 如果还需要继续计算，跳到 2；如果需要更改计算的函数等配置，跳到 1。

中断和轮询模式

此外，如果用户不想使用零等待模式，CORDIC 还提供了中断和轮询模式，具体步骤如下：

1. 配置 CORDIC_CSRx 寄存器
   a) 配置 CORDIC_CSRx.FUNC 寄存器位确定接下来需要计算的函数。
   b) 配置 CORDIC_CSRx.SCALE 寄存器位确定比例因子
   c) 配置 CORDIC_CSRx.NRES 寄存器位确定输出数据量
   d) 配置 CORDIC_CSRx.NARGS 寄存器位确定输入数据量
   e) 配置 CORDIC_CSRx.RESSIZE 寄存器位确定输出的位宽
   f) 配置 CORDIC_CSRx.ARGSIZE 寄存器位确定输入的位宽
   g) 如果使用中断模式，则还需要配置 CORDIC_CSRx.RRDYIEN 等于 1，如果是轮询则不用。
2. 根据 CORDIC_CSRx.NARGS 的值，对 ARXx 和 ARYx 寄存器写入需要计算的数据
3. 如下：
   a) 轮询模式：可以设置程序轮询 CORDIC_CSRx.RRDY 状态位等于 1，然后对 CORDIC_CSRx.RRDYCLR 写 1 清除状态位
   b) 中断模式：等待中断进入到中断处理函数，查询 CORDIC_CSRx.RRDY 状态位等于 1，然后对 CORDIC_CSRx.RRDYCLR 写 1 清除状态位
4. 根据 CORDIC_CSRx.NRES 的值，读取 ARXx 和 ARYx 寄存器获取计算结果
5. 如果还需要继续计算，跳到 2；如果需要更改计算的函数等配置，跳到 1。

12.4.4.2 双控制单元

CORDIC 中有两套独立的控制单元，每一套都能够实现零等待模式、中断模式和轮询模式，且都能独立控制 CORDIC 计算单元，但是 CORDIC 只有一套计算单元，这就意味着，如果两套控制单元同时运行时，需要通过仲裁机制来确定那一套获得控制权，具体如下图所示：

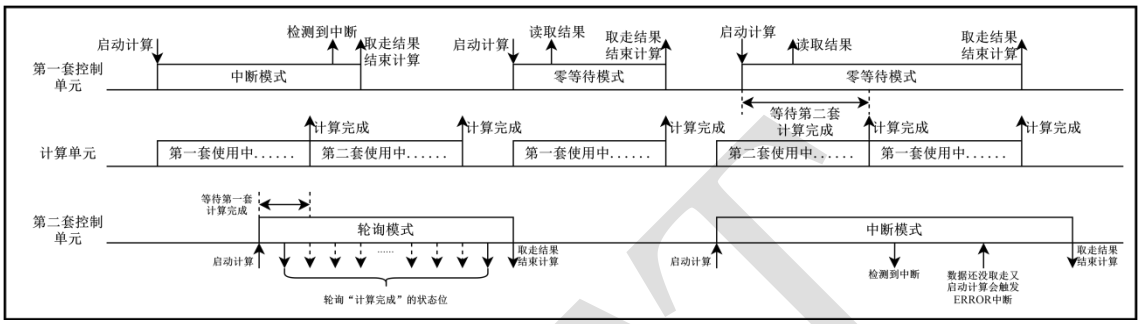


图 12-2 双控制单元仲裁图

上图中，首先是第一套控制单元启动了计算，此时计算单元处于空闲状态，因此第一套获得了仲裁使用了计算单元；在计算过程中，第二套启动了计算，此时计算单元处于繁忙的状态，因此第二套需要等待第一套计算完成之后才能获得计算单元的仲裁。

此外，上图在第二套第二次中断模式过程中，如果数据还没被取走就重新启动计算，会触发 ERR 中断（CORDIC_CSRx.ERRIEN），如果需要清除该中断，需要对 CORDIC_CSRx.ERRCLR 写 1 才能清除，计算过程中写入的数据都会被丢弃。

12.5 寄存器定义

12.5.1 寄存器列表

CORDIC 基地址：

CORDIC(0x4003F000)

偏移	实例地址	名称	默认值	描述
0x00	CORDIC 基地址+0x00	CORDIC_CSR0	0x00000000	CORDIC 控制/状态寄存器 0
0x04	CORDIC 基地址+0x04	CORDIC_ARX0	0x00000000	CORDIC 参数/结果 X 寄存器 0
0x08	CORDIC 基地址+0x08	CORDIC_ARY0	0x00000000	CORDIC 参数/结果 Y 寄存器 0
0x20	CORDIC 基地址+0x20	CORDIC_CSR1	0x00000000	CORDIC 控制/状态寄存器 1
0x24	CORDIC 基地址+0x24	CORDIC_ARX1	0x00000000	CORDIC 参数/结果 X 寄存器 1
0x28	CORDIC 基地址+0x28	CORDIC_ARY1	0x00000000	CORDIC 参数/结果 Y 寄存器 1

12.5.2 寄存器描述

12.5.2.1 CORDIC 控制/状态寄存器（CORDIC_CSR0）

- 名称：CORDIC Control / Status register 0
- 偏移地址：0x00+N*0x20[N=0-1]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RRDY	ERR	RRDY CLR	ERRC LR						ARGSI ZE	RESSI ZE	NARG S	NRES		ERRIE N	RRDYI EN
RAC	R	W	W						R/W	R/W	R/W	R/W		R/W	R/W
0x0	0x0	0x0	0x0						0x0	0x0	0x0	0x0		0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SCALE							FUNC			
					R/W							R/W			
					0x0							0x0			

字段	说明
[31] RRDY	**计算结果完成中断状态（Result Ready Interrupt Status)** 该位置 1 表示 CORDIC 已经完成了计算，此时用户可以读取数据寄存器取数据。 0: CORDIC 无计算或 CORDIC 还未计算完成 1: CORDIC 计算完成 Note： 1. 软件可以对 RRDYCLR 写 1 清除 RRDY 中断状态； 2. 当软件启动下一次计算时，该位会被硬件自动清 0，如果软件未完成对数据寄存器（CORDIC 结果）进行读取，那么硬件会丢弃上一次的计算结果。
[30] ERR	**错误中断状态(Error Interrupt Status)** 当 CORDIC 在执行运算过程中，如果软件对数据寄存器写入数据，或者对 CSR0 寄存器的配置进行了修改，硬件会报错，但是新写入的数据和配置均不会被锁存，CORDIC 仍会进行计算。 0: 无错误 1: 检测到错误 Note：软件可以对 ERRCLR 写 1 清除 ERR 中断状态。
[29] RRDYCLR	**计算结果完成中断清除(Result Ready Interrupt Clear)** 0: 无 1: 清除 RRDY 中断状态
[28] ERRCLR	**错误中断清除(Error Interrupt Clear)** 0: 无 1: 清除 ERR 中断状态
[22] ARGSIZE	**输入数据的宽度(Input Argument Size)** 0：输入数据的位宽是 32 位，数据寄存器的写入格式为 q1.31 的格式 1：输入数据的位宽是 16 位，数据寄存器的写入格式为 q1.15 的格式
[21] RESSIZE	**输出数据的宽度(Output Result Size)** 0：输出结果的位宽是 32 位，数据寄存器的读取格式为 q1.31 的格式 1：输出结果的位宽是 16 位，数据寄存器的读取格式为 q1.15 的格式
[20] NARGS	**输入数据的个数(Input Argument Number)** 0：输入数据有一个，写入 ARX0 数据寄存器后会启动传输，写入 ARY0 数据寄存器的值会被硬件忽略 1：输入数据有两个，需要先写入 ARX0 数据寄存器，再写入 ARY0 数据寄存器才会启动传输 Note： 1、该位只有在 ARGSIZE=0 时才有效，如果在 ARGSIZE=1 时该位只能配置成 0，如果配置成 1 数据结果错误 2、该位需要严格按照 CORDIC 各个函数的要求配置，如果配置错误会导致 CORDIC 无法启动传输，比如取模函数需要 2 个 32 位输入，该位必须配置成 1。
[19]	**输出数据的个数(Output Result Number)**

NRES	0：输出结果有一个，读取 ARX0 数据寄存器获得输出结果 1：输出结果有两个，读取 ARX0 数据寄存器获得第一个输出结果，读取 ARY0 数据寄存器获得第二个输出结果 Note： 1、该位只有在 RESSIZE=0 时才有效，如果在 RESSIZE=1 时该位只能配置成 0，如果配置成 1 数据结果错误 2、该位需要严格按照 CORDIC 各个函数的要求配置，如果函数只有一个输出数据的话，那么该位只能配置为 0，比如反正切函数；如果函数有两个输出数据的话，可以配一个结果也可以配两个结果。
[17] ERRIEN	**错误中断使能(Error Interrupt Enable)** 0: 不使能错误中断 1: 使能错误中断
[16] RRDYIEN	**计算结果完成中断使能（Result Ready Interrupt Enable)** 0: 不使能计算结果完成中断 1: 使能计算结果完成中断
[10:8] SCALE	**CORDIC 比例因子设置(CORDIC Scale Setting)** CORDIC 能计算的范围有限，为了拓宽计算的范围，可以配置 SCALE 来对输入数据进行缩放，缩放后相应的计算结果也需要进行扩大，具体参考各个函数。 0：不缩放 1：缩放 2^1 倍 2：缩放 2^2 倍 ..... 7：缩放 2^7 倍 Note：不是所有的函数都可以适配所有的比例因子，不同的函数对 SCALE 的限制是不同的，硬件不会阻止软件去配置错误的 SCALE，只是不能保证结果的正确性，SCALE 的限制具体如下所示： 1. SCALE 取值范围是 0 的函数：cos/sin/phase/modules 2. SCALE 取值范围是 1 的函数：cosh/sinh/artanh 3. SCALE 取值范围是[0,7]的函数：arctan 4. SCALE 取值范围是[1,4]的函数：ln 5. SCALE 取值范围是[0,2]的函数：sqrt
[3:0] FUNC	**cordic 计算的函数(Function)** 0: 余弦函数(Cosine) 1: 正弦函数(Sine) 2: 相角函数(Phase) 3: 取模函数(Modulus) 4: 反正切函数(Arctangent) 5: 双曲余弦函数(Hyperbolic cosine) 6: 双曲正弦函数(Hyperbolic sine) 7: 双曲反正切函数(Arctanh) 8: 自然对数函数(Natural logarithm) 9: 开平方根函数(Square Root) 10 to 15: reserved

12.5.2.2 CORDIC 参数/结果 X 寄存器（CORDIC_ARX0）

- **名称：CORDIC Argument / Result register 0**
- **偏移地址：0x04+N*0x20[N=0-1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARGRESX															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGRESX															
R/W															
0x0															

字段	说明
[31:0] ARGRESX	**数据寄存器 X（Argument/Result Data Register X）** 该寄存器是使用 CORDIC 写入输入数据和获取计算结果的主要寄存器，如果是写入该寄存器，则代表写入 CORDIC 将要计算的输入数据；如果是读取该寄存器，则代表读取 CORDIC 已经计算完成的数据。 关于写入和读取有以下注意项： 写入数据应注意： 1. 如果输入数据是 32 位（ARGSIZE=0），且需要一个输入数据（NARGS=0），对该位写入一个数据会触发 CORDIC 的计算。 2. 如果输入数据是 32 位（ARGSIZE=0），且需要两个输入数据（NARGS=1），对该位写入一个数据不会触发 CORDIC 的计算，还需要对 ARGRESY 写入一个输入数据才会触发 CORDIC 的计算。 3. 如果输入数据是 16 位（ARGSIZE=1），那么要求 NARGS 必须为 0，此时对该位写入一个数据会触发 CORDIC 的计算； 4. 如果输入数据是 16 位（ARGSIZE=1），如果输入数据有一个，那么只有 ARGRESX[15:0]有效；如果输入数据有两个，那么 ARGRESX[15:0]代表的是第一个输入数据，ARGRESX[31:16]代表的是第二个输入数据。 读取结果应注意： 1. 如果输出数据是 32 位（RESSIZE=0），读取该寄存器会得到函数的主要结果 2. 如是输出数据是 16 位（RESSIZE=1），读取该寄存器会得到函数的所有结果，如果函数只有一个结果，那么只有 ARGRESX[15:0]有效；如果是两个结果，那么 ARGRESX[15:0]代表的是函数的主要结果，ARGRESX[31:16]代表的是函数的次要结果。

12.5.2.3 CORDIC 参数/结果 Y 寄存器（CORDIC_ARY0）

- **名称：CORDIC Argument / Result register 0**
- **偏移地址：0x08+N*0x20[N=0-1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARGRESY															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGRESY															
R/W															
0x0															

字段	说明
[31:0] ARGRESY	**数据寄存器 Y（Argument/Result Data Register Y）** 该寄存器是使用 CORDIC 写入输入数据和获取计算结果的次要寄存器，如果是写入该寄存器，则代表写入 CORDIC 将要计算的输入数据；如果是读取该寄存器，则代表读取 CORDIC 已经计算完成的数据。 Note： 1. 该位只有在输入数据是 32 位（ARGSIZE=0），且需要两个输入数据（NARGS=1）时，写入该寄存器才有效，这种情况下需要先对 ARGRESX 写第一个输入数据，再对该寄存器写第二个输入数据才能触发 CORDIC 计算。 2. 该位只有在输出数据是 32 位（RESSIZE=0），且需要两个输出结果（NRES=1）时，读取该寄存器才有效，读取该寄存器将会获得函数的次要结果。

12.5.2.4 CORDIC 控制/状态寄存器（CORDIC_CSR1）

- **名称：CORDIC Control / Status register 1**
- **偏移地址：0x00+N*0x20[N=0-1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RRDY	ERR	RRDY CLR	ERRC LR						ARGSI ZE	RESSI ZE	NARG S	NRES		ERRIE N	RRDYI EN
RAC	R	W	W						R/W	R/W	R/W	R/W		R/W	R/W
0x0	0x0	0x0	0x0						0x0	0x0	0x0	0x0		0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SCALE							FUNC			
					R/W							R/W			
					0x0							0x0			

字段	说明
[31] RRDY	**计算结果完成中断状态（Result Ready Interrupt Status)** 该位置 1 表示 CORDIC 已经完成了计算，此时用户可以读取数据寄存器取数据。 0: CORDIC 无计算或 CORDIC 还未计算完成 1: CORDIC 计算完成 Note： 1. 软件可以对 RRDYCLR 写 1 清除 RRDY 中断状态； 2. 当软件启动下一次计算时，该位会被硬件自动清 0，如果软件未完成对数据寄存器（CORDIC 结果）进行读取，那么硬件会丢弃上一次的计算结果。
[30] ERR	**错误中断状态(Error Interrupt Status)** 当 CORDIC 在执行运算过程中，如果软件对数据寄存器写入数据，或者对 CSR1 寄存器的配置进行了修改，硬件会报错，但是新写入的数据和配置均不会被锁存，CORDIC 仍会进行计算。 0: 无错误

	1: 检测到错误 Note：软件可以对 ERRCLR 写 1 清除 ERR 中断状态。
[29] RRDYCLR	**计算结果完成中断清除(Result Ready Interrupt Clear)** 0: 无 1: 清除 RRDY 中断状态
[28] ERRCLR	**错误中断清除(Error Interrupt Clear)** 0: 无 1: 清除 ERR 中断状态
[22] ARGSIZE	**输入数据的宽度(Input Argument Size)** 0：输入数据的位宽是 32 位，数据寄存器的写入格式为 q1.31 的格式 1：输入数据的位宽是 16 位，数据寄存器的写入格式为 q1.15 的格式
[21] RESSIZE	**输出数据的宽度(Output Result Size)** 0：输出结果的位宽是 32 位，数据寄存器的读取格式为 q1.31 的格式 1：输出结果的位宽是 16 位，数据寄存器的读取格式为 q1.15 的格式
[20] NARGS	**输入数据的个数(Input Argument Number)** 0：输入数据有一个，写入 ARX1 数据寄存器后会启动传输，写入 ARY1 数据寄存器的值会被硬件忽略 1：输入数据有两个，需要先写入 ARX1 数据寄存器，再写入 ARY1 数据寄存器才会启动传输 Note： 1、该位只有在 ARGSIZE=0 时才有效，如果在 ARGSIZE=1 时该位只能配置成 0，如果配置成 1 数据结果错误 2、该位需要严格按照 CORDIC 各个函数的要求配置，如果配置错误会导致 CORDIC 无法启动传输，比如取模函数需要 2 个 32 位输入，该位必须配置成 1。
[19] NRES	**输出数据的个数(Output Result Number)** 0：输出结果有一个，读取 ARX1 数据寄存器获得输出结果 1：输出结果有两个，读取 ARX1 数据寄存器获得第一个输出结果，读取 ARY1 数据寄存器获得第二个输出结果 Note： 1、该位只有在 RESSIZE=0 时才有效，如果在 RESSIZE=1 时该位只能配置成 0，如果配置成 1 数据结果错误 2、该位需要严格按照 CORDIC 各个函数的要求配置，如果函数只有一个输出数据的话，那么该位只能配置为 0，比如反正切函数；如果函数有两个输出数据的话，可以配一个结果也可以配两个结果。
[17] ERRIEN	**错误中断使能(Error Interrupt Enable)** 0: 不使能错误中断 1: 使能错误中断
[16] RRDYIEN	**计算结果完成中断使能（Result Ready Interrupt Enable)** 0: 不使能计算结果完成中断 1: 使能计算结果完成中断
[10:8] SCALE	**CORDIC 比例因子设置(CORDIC Scale Setting)** CORDIC 能计算的范围有限，为了拓宽计算的范围，可以配置 SCALE 来对输入数据进行缩放，缩放后相应的计算结果也需要进行扩大，具体参考各个函数。 0：不缩放 1：缩放 2^1 倍 2：缩放 2^2 倍 ..... 7：缩放 2^7 倍 Note：不是所有的函数都可以适配所有的比例因子，不同的函数对 SCALE 的限制是不同的，硬件不会阻止软件去配置错误的 SCALE，只是不能保证结果的正确性，SCALE 的限制具体如下所示： 1. SCALE 取值范围是 0 的函数：cos/sin/phase/modules 2. SCALE 取值范围是 1 的函数：cosh/sinh/artanh 3. SCALE 取值范围是[0,7]的函数：arctan 4. SCALE 取值范围是[1,4]的函数：ln 5. SCALE 取值范围是[0,2]的函数：sqrt
[3:0] FUNC	**cordic 计算的函数(Function)** 0: 余弦函数(Cosine) 1: 正弦函数(Sine) 2: 相角函数(Phase) 3: 取模函数(Modulus) 4: 反正切函数(Arctangent) 5: 双曲余弦函数(Hyperbolic cosine)

6: 双曲正弦函数(Hyperbolic sine)
7: 双曲反正切函数(Arctanh)
8: 自然对数函数(Natural logarithm)
9: 开平方根函数(Square Root)
10 to 15: reserved

12.5.2.5 CORDIC 参数/结果 X 寄存器（CORDIC_ARX1）

- **名称：CORDIC Argument / Result register 1**
- **偏移地址：0x04+N*0x20[N=0-1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARGRESX															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGRESX															
R/W															
0x0															

字段	说明
[31:0] ARGRESX	**数据寄存器 X（Argument/Result Data Register X）** 该寄存器是使用 CORDIC 写入输入数据和获取计算结果的主要寄存器，如果是写入该寄存器，则代表写入 CORDIC 将要计算的输入数据；如果是读取该寄存器，则代表读取 CORDIC 已经计算完成的数据。 关于写入和读取有以下注意项： 写入数据应注意： 1. 如果输入数据是 32 位（ARGSIZE=0），且需要一个输入数据（NARGS=0），对该位写入一个数据会触发 CORDIC 的计算。 2. 如果输入数据是 32 位（ARGSIZE=0），且需要两个输入数据（NARGS=1），对该位写入一个数据不会触发 CORDIC 的计算，还需要对 ARGRESY 写入一个输入数据才会触发 CORDIC 的计算。 3. 如果输入数据是 16 位（ARGSIZE=1），那么要求 NARGS 必须为 0，此时对该位写入一个数据会触发 CORDIC 的计算； 4. 如果输入数据是 16 位（ARGSIZE=1），如果输入数据有一个，那么只有 ARGRESX[15:0]有效；如果输入数据有两个，那么 ARGRESX[15:0]代表的是第一个输入数据，ARGRESX[31:16]代表的是第二个输入数据。 读取结果应注意： 1. 如果输出数据是 32 位（RESSIZE=0），读取该寄存器会得到函数的主要结果 2. 如是输出数据是 16 位（RESSIZE=1），读取该寄存器会得到函数的所有结果，如果函数只有一个结果，那么只有 ARGRESX[15:0]有效；如果是两个结果，那么 ARGRESX[15:0]代表的是函数的主要结果，ARGRESX[31:16]代表的是函数的次要结果。

12.5.2.6 CORDIC 参数/结果 Y 寄存器（CORDIC_ARY1）

- 名称：CORDIC Argument / Result register 1
- 偏移地址：0x08+N*0x20[N=0-1]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARGRESY															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGRESY															
R/W															
0x0															

字段	说明
[31:0] ARGRESY	**数据寄存器 Y（Argument/Result Data Register Y）** 该寄存器是使用 CORDIC 写入输入数据和获取计算结果的次要寄存器，如果是写入该寄存器，则代表写入 CORDIC 将要计算的输入数据；如果是读取该寄存器，则代表读取 CORDIC 已经计算完成的数据。 Note： 1. 该位只有在输入数据是 32 位（ARGSIZE=0），且需要两个输入数据（NARGS=1）时，写入该寄存器才有效，这种情况下需要先对 ARGRESX 写第一个输入数据，再对该寄存器写第二个输入数据才能触发 CORDIC 计算。 2. 该位只有在输出数据是 32 位（RESSIZE=0），且需要两个输出结果（NRES=1）时，读取该寄存器才有效，读取该寄存器将会获得函数的次要结果。

13 数字滤波单元（IIR）

13.1 简介

IIR 加速单元对向量执行算术运算，它包含了一个乘法器/累加器单元。一共有 6 个独立的 IIR 加速单元，其中 3 套最高支持 2 阶(IIR3、IIR4、IIR5)，3 套最高支持 4 阶(IIR0、IIR1、IIR2)。该单元可以节省 CPU 一些频繁和冗长的乘累加滤波操作，从而减小 CPU 开销。

13.2 主要特性

IIR 主要具有以下特性：

- 48×32 位乘法器
- 64 位加法器
- 16 位输入和输出数据
- 支持 6 套 IIR（3 套最高 2 阶，3 套最高 4 阶）
- 支持阶数灵活可配（1-2 阶/1-4 阶）
- 支持 48bit 高精度乘累加
- 支持 48bit 反馈限幅
- 支持系数自动和手动选择

13.3 结构框图

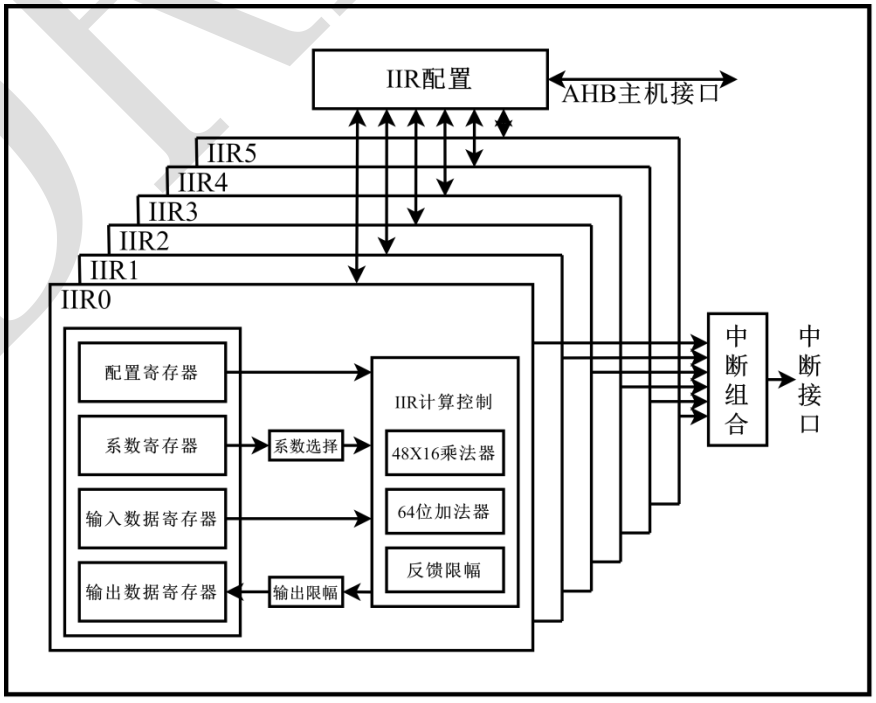


图 13-1 IIR 结构框图

13.4 功能描述

13.4.1 基本说明

数字滤波单元一共有 6 个 IIR，IIR0，IIR1，IIR2 这三个滤波器最大支持 4 阶；IIR3，IIR4，IIR5 这三个滤波器最大支持 2 阶。其他的都是一样的，因此后续主要对 4 阶的 IIR 展开说明。

IIR 最高支持 4 阶，因此最少需要 9 个计算系数，系数可以通过 IIR_GNB0R、IIR_GNB1R、IIR_GNB2R、IIR_GNB3R、IIR_GNB4R、IIR_GNA1R、IIR_GNA2R、IIR_GNA3R 和 IIR_GNA4R 寄存器配置，此外，IIR 最高支持 4 组可配置计算系数，用户可以根据不同的应用场景手动切换需要的系数组，也可以配置不同的边界值让硬件自动的进行切换系数组，关于自动系数切换的功能具体如下所示：

13.4.1.1 自动系数切换

可以配置 IIR_CR.ICS 寄存器来确定使用系数的组数，如果将 IIR_CR.ICS 配置成 3，那么自动系数切换一共有 4 个边界寄存器需要配置，边界为 L0、L1、L2 和 L3，分别是 IIR_IL0R.BL0R、IIR_IL0R.BL1R、IIR_IL1R.BL2R 和 IIR_IL1R.BL3R 寄存器来配置，其中 L0＜L1＜L2＜L3，如果边界大小不是按这个顺序来的话，会影响自动系数切换的准确性。

配置的 L0、L1、L2 和 L3 都是无符号的 15 位正整数，配置后硬件会生成对应的负数 LN0、LN1、LN2 和 LN3，此外生成 L1 和 L2 的平均值 L12（$L12 = \frac{L1+L2}{2}$），还有 L2 和 L3 的平均值 L23（$L23 = \frac{L2+L3}{2}$），L12 和 L23 也有对应的负数 LN12 和 LN23。硬件会根据输入数据的值和上述生成的正负值来划分区域（Zone 区域）

1. Z3 区域：输入数据的值大于 L3 的区域
2. Z2 区域：输入数据的值在 L2 和 L23 之间的区域
3. Z1 区域：输入数据的值在 L1 和 L12 之间的区域
4. Z0 区域：输入数据的值在 LN0 和 LN1 之间的区域
5. ZN1 区域：输入数据的值在 LN12 和 L1 之间的区域
6. ZN2 区域：输入数据的值在 LN23 和 L23 之间的区域
7. ZN3 区域：输入数据的值小于 LN3 的区域

此外还有一些不在 Zone 区域内的额外区域（Block 区域），它们的定义如下所示：

1. B0：输入数据的值在 LN3 和 LN23 之间的区域
2. B1：输入数据的值在 LN2 和 LN12 之间的区域
3. B2：输入数据的值在 LN1 和 LN0 之间的区域
4. B3：输入数据的值在 L0 和 L1 之间的区域
5. B4：输入数据的值在 L12 和 L2 之间的区域
6. B5：输入数据的值在 L23 和 L3 之间的区域

具体的区域划分如图 13-2 所示：

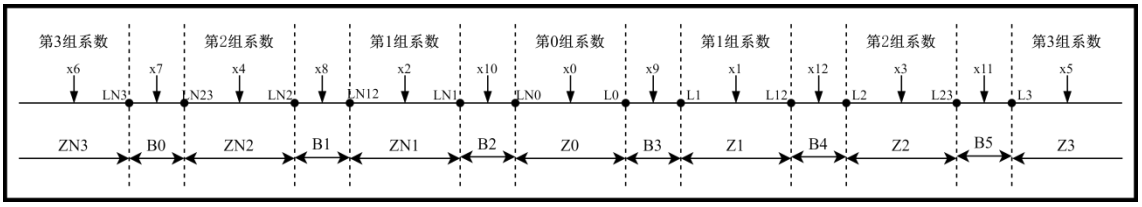


图 13-2　自动系数切换区域划分图

上图显示，不同的区域使用的系数组是不一样的，x0~x12 代表用户先后往 IIR 中配置的输入数据，不同的输入数据使用的系数组是不一样的，具体使用的系数组如下所示：

1. 如果输入数据落在 Z0 区域，则使用的是第 0 组系数。例如 x0 计算时采用的就是第 0 组系数。
2. 如果输入数据落在 Z1 或者 ZN1 区域，则使用的是第 1 组系数。例如 x1 和 x2 计算时采用的就是第 1 组系数。
3. 如果输入数据落在 Z2 或者 ZN2 区域，则使用的是第 2 组系数。例如 x3 和 x4 计算时采用的就是第 2 组系数。
4. 如果输入数据落在 Z3 或者 ZN3 区域，则使用的是第 3 组系数。例如 x5 和 x6 计算时采用的就是第 3 组系数。

若输入数据落在 Block 区域（B0~B5），那么需要根据上一个 Zone 区域的组系数来决定接下来使用的系数组是什么，关于 Block 区域的系数组选择有以下两个规则：

1. 规则 1：若输入数据上一次落在 Zone 区域，这次的输入数据落在了该 Zone 区域相邻的 Block 区域时，那么该输入数据使用的系数组和该 Zone 区域的系数组一致。
2. 规则 2：若输入数据上一次落在 Zone 区域，这次的输入数据没有落在该 Zone 区域相邻的 Block 区域时，此时的系统组采用的该 Block 区域和该 Zone 区域之间靠近 Block 区域的 Zone 区域的系数组。

对于图 17-2 中有上述规则的体现，具体如下所示：

1. x7 落在 B0 区域，x6 在 ZN3 区域使用的是第 3 组系数，根据规则 1， x7 采用的是第 3 组系数；
2. x8 在区域 B1 中，x7 使用的是第 3 组系数，也就是 ZN3 区域，B1 和 ZN3 区域不是相邻的，根据规则 2，此时 x8 使用的是 B1 和 ZN3 之间最靠近 B1 的 ZN2 区域系数组，也就是第 2 组系数。
3. x9 在区域 B3 中，x8 使用的是第 2 组系数，也就是 ZN2 区域，根据规则 2，x9 使用的是 B3 和 ZN2 之间最靠近 B3 的 Z0 区域系数组，也就是第 0 组系数。
4. x10 在 B2 区域中，x9 使用的是第 0 组系数，也就是 Z0 区域，根据规则 1，x10 使用的是第 0 组系数。
5. x11 使用规则 2 采用的是 Z2 区域的系数组，也就是第 2 组系数。
6. x12 使用规则 1 采用的是 Z2 区域的系数组，也就是第 2 组系数。

13.4.2　IIR 滤波器

无限脉冲响应（IIR）滤波器的公式如下：

$$y_n = \sum_{k=0}^{N} b_k x_{n-k} + \sum_{k=1}^{N} a_k y_{n-k}$$

$$Y = B * X + A * Y'$$

该公式指的是输出向量 Y 是长度为 N+1 的第一系数 B 和长度为 N+1 的向量 X 的卷积，此外还要加上延迟输出向量Y'与长度为 N 的第二系数 A 的卷积。这里的 N 也指的是滤波器的阶数，目前的 IIR 的 N 最大值为 4，因此第一系数一共有 5 个系数（分别为 b0,b1,b2,b3,b4），第二系数一共有 4 个系数（分别为 a1,a2,a3,a4），过滤器的结构如下所示：

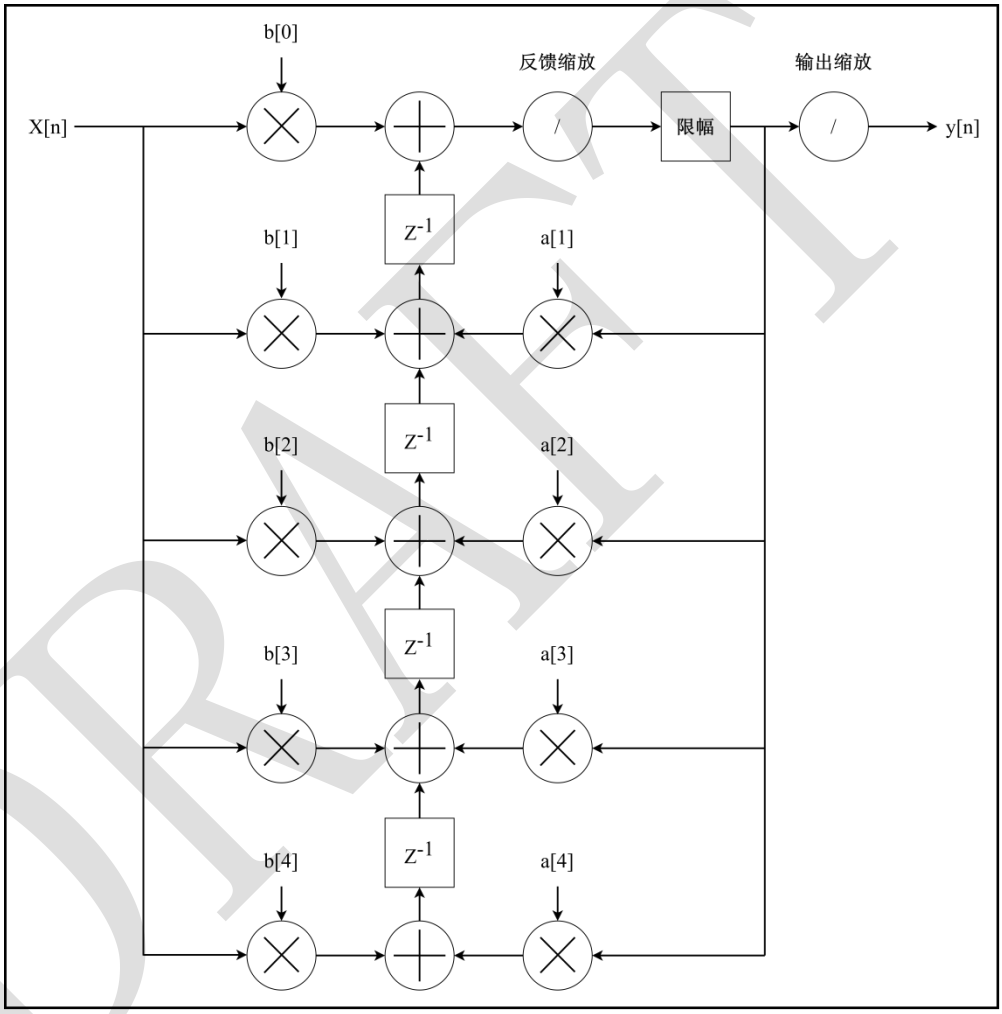


图 13-3　IIR 4 阶滤波器结构图

13.4.3　定点描述

IIR 以定点有符号数格式操作，输入和输出均为 q1.15。

在 q1.15 格式中，数字由一个有符号位和 15 个小数位（二进制小数点位数）表示。因此，数值范围是-1(0x8000)到 $1-2^{-15}$(0x7FFF)。46bit 累加器，其中 45 位为小数，1 位有符号位。它允许的数据范围是-1(0x200000000000)到 $1-2^{-45}$(0x1FFFFFFFFFFF)。

13.4.3.1 输出处理

对于图 17-3 中，输出之前一共经过 3 个后处理，分别是反馈缩放、输出限幅和输出缩放，反馈缩放是对乘累加的结果进行缩放，输出限幅是限制反馈缩放后的数据范围，输出缩放是在软件取数据之前对限幅后的结果进行缩放。

1. 反馈缩放：用户可以配置 IIR_SCL.FSCL 寄存器控制缩放的比例，最大支持2^{31}缩放
2. 输出限幅：用户可以配置 IIR_IMHR.MAXH 和 IIR_IMLR.MAXL 寄存器控制输出限幅的上限，如果输出超过了上限，则等于寄存器配置的上限值；另外可以配置 IIR_INHR.MINH 和 IIR_INLR.MINL 寄存器控制输出限幅的下限，如果输出低于下限，则等于寄存器配置的下限值。
3. 输出缩放：用户可以配置 IIR_SCL.OSCL 寄存器控制缩放的比例，最大支持2^{31}缩放。

13.4.4 操作流程

IIR 提供了两种操作模式，分别为轮询模式和输出零等待模式，首先 IIR 需要先进行初始化配置，初始化配置的流程如下所示：

1. IIR_CR.ORD 确定 IIR 使用的阶数，配置 IIR 需要的 Ax 和 Bx 系数寄存器，根据 IIR_CR.ICS 的值确定配置的组数。
2. 自动系数切换功能：IIR_CR.CMS 确定 IIR 是否需要自动系数切换功能，IIR_CR.ICS 确定 IIR 使用的系数组，IIR_IL0R 和 IIR_IL1R 确定自动系数切换的边界，IIR_GNB*R 和 IIR_GNA*R 配置 IIR 的系数
3. 输出限幅缩放功能：IIR_CR.ILE 确定 IIR 是否需要限幅功能，IIR_IMHR、IIR_IMLR、IIR_INHR 和 IIR_INLR 确定限幅范围，IIR_SCL 确定反馈缩放和输出缩放的裕度。
4. 对 IIR_CR.IEN 写 1 启动 IIR 计算单元

对于轮询模式和输出零等待模式，如图 13-4 所示：

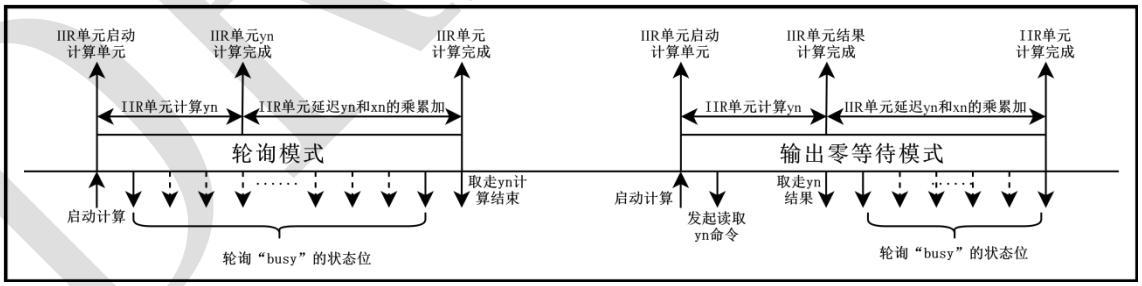


图 13-4 IIR 操作模式

对于 IIR 计算单元，在用户对 IIR_IDR 寄存器写入数据时就启动了计算，在 3 个系统时钟周期左右能计算出 yn 计算结果，此时已经可以读取 IIR_ODR 得到计算结果，此后 IIR 计算单元继续计算旧数据的 yn 和 xn 的乘累加，直至计算完成，在计算延迟 yn 和 xn 的乘累加期间，用户不允许启动计算，如果执意启动计算，那么会触发 IIR 错误中断（IIR_ISR.ERR 会被置 1）。关于轮询模式和输出零等待模式的操作流程如下所示：

轮询模式

1. 对 IIR_IDR 写入计算数据启动 IIR 单元进行计算，此时 IIR_ISR.BSY 状态会被置 1。
2. 用户需要定时轮询 IIR_ISR.BSY 状态位，直到 IIR_ISR.BSY=0 为止。

3. 轮询结束后可以读取 IIR_ODR 取走计算结果，如果需要再计算下一个结果可以跳回步骤 1。

输出零等待模式

1. 对 IIR_IDR 写入计算数据启动 IIR 单元进行计算，此时 IIR_ISR.BSY 状态会被置 1。
2. 发起读取 IIR_ODR 的命令，计算单元会拉住系统总线直至 IIR 计算完输出结果（3 个系统时钟周期），计算完之后会将结果直接赋予 IIR_ODR 寄存器，读取的结果就是计算结果。
3. 用户需要定时轮询 IIR_ISR.BSY 状态位，直到 IIR_ISR.BSY=0 为止。
4. 轮询结束后可以读取 IIR_ODR 取走计算结果，如果需要再计算下一个结果可以跳回步骤 1。

13.5 寄存器定义

13.5.1 寄存器列表

IIR 基地址：

IIR0(0x4003E000) IIR1(0x4003E100) IIR2(0x4003E200)

IIR3(0x4003E300) IIR4(0x4003E400) IIR5(0x4003E500)

偏移	实例地址	名称	默认值	描述
0x00	IIR 寄存器+0x00	IIR_CR	0x00000004	IIR 控制寄存器
0x04	IIR 寄存器+0x04	IIR_ISR	0x00000000	IIR 状态寄存器
0x08	IIR 寄存器+0x08	IIR_IDR	0x00000000	IIR 数据输入寄存器
0x0C	IIR 寄存器+0x0C	IIR_ODR	0x00000000	IIR 数据输出寄存器
0x20	IIR 寄存器+0x20	IIR_SCL	0x00000000	IIR 缩放寄存器
0x30	IIR 寄存器+0x30	IIR_G0B0R	0x00000000	IIR 第 0 组 B0 系数设置寄存器
0x34	IIR 寄存器+0x34	IIR_G0B1R	0x00000000	IIR 第 0 组 B1 系数设置寄存器
0x38	IIR 寄存器+0x38	IIR_G0B2R	0x00000000	IIR 第 0 组 B2 系数设置寄存器
0x3C	IIR 寄存器+0x3C	IIR_G0B3R	0x00000000	IIR 第 0 组 B3 系数设置寄存器
0x40	IIR 寄存器+0x40	IIR_G0B4R	0x00000000	IIR 第 0 组 B4 系数设置寄存器
0x50	IIR 寄存器+0x50	IIR_G0A1R	0x00000000	IIR 第 0 组 A1 系数设置寄存器
0x54	IIR 寄存器+0x54	IIR_G0A2R	0x00000000	IIR 第 0 组 A2 系数设置寄存器
0x58	IIR 寄存器+0x58	IIR_G0A3R	0x00000000	IIR 第 0 组 A3 系数设置寄存器
0x5C	IIR 寄存器+0x5C	IIR_G0A4R	0x00000000	IIR 第 0 组 A4 系数设置寄存器

13.5.2 寄存器描述

13.5.2.1 IIR 控制寄存器（IIR_CR）

- 名称：**IIR Control Register**
- 偏移地址：**0x00**
- 默认值：**0x00000004**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							RST	IEE				ORD		ILE	IEN
							W1C	R/W				R/W		R/W	R/W
							0x0	0x0				0x1		0x0	0x0

字段	说明
[8] RST	**IIR 软复位(IIR Soft Reset)** 0：不复位 1：复位 Note:禁止在模块运算过程中复位或者关使能 以避免造成不可预期的后果
[7] IEE	**IIR 错误中断使能位(IIR Error Interrupt Enable)** 0：不使能 1：使能
[3:2] ORD	**IIR 阶数选择(IIR Order Selection)** 0：1 阶 1：2 阶 2：3 阶 3：4 阶 Note:如果有中途切换阶数的需求，需按最大阶数配置，通过调整系数来实现低阶计算，延迟按最大阶数计算。
[1] ILE	**IIR 限幅使能(IIR Limit Enable)** 0：不使能 1：使能
[0] IEN	**IIR 使能(IIR Enable)** 0：不使能 1：使能

13.5.2.2 IIR 状态寄存器（IIR_ISR）

- 名称：IIR Status Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														BSY	ERR
														R	R/W1C
														0x0	0x0

字段	说明
[1] BSY	**IIR 忙标志位(IIR Busy Status)** 0：IIR 处于空闲状态 1：IIR 处于忙碌状态
[0] ERR	**IIR 错误中断标志(IIR Error Pending)** 在 IIR 处于忙碌状态（BSY=1）下或者 IIR 不使能（IEN=0）时对 IDR 寄存器写入了输入数据会产生错误 0：未检测到错误 1：检测到错误 Note：对 ERR 写 1 该位会清 0

13.5.2.3 IIR 数据输入寄存器（IIR_IDR）

- 名称：IIR Data-In Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID															
R/W															
0x0															

字段	说明
[15:0] ID	**IIR 数据输入寄存器(IIR Data-In Register)** 写入该寄存器表示 IIR 即将计算的输入数据 Note： 1、输入数据为 16bit； 2、当写入数据后将硬件自动启动计数；

13.5.2.4 IIR 数据输出寄存器（IIR_ODR）

- 名称：IIR Data-Out Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD															
R															
0x0															

字段	说明
[15:0] OD	**IIR 数据输出寄存器(IIR Data-Out Register)** 读取该寄存器获取 IIR 的计算结果

13.5.2.5 IIR 缩放寄存器（IIR_SCL）

- 名称：IIR Scale Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			OSCL								FSCL				
			R/W								R/W				
			0x0								0x0				

字段	说明
[12:8] OSCL	**IIR 输出缩放控制位(IIR Output Scale)** IIR 输出数据缩放寄存器，最大支持 2^31 缩放，缩放倍数如下: 0：不缩放 1：2^1 缩放 2：2^2 缩放 …… 31：2^31 缩放
[4:0] FSCL	**IIR 反馈缩放控制位(IIR Feedback Scale)** IIR 反馈数据缩放寄存器，最大支持 2^31 缩放，缩放倍数如下: 0：不缩放 1：2^1 缩放 2：2^2 缩放 …… 31：2^31 缩放

13.5.2.6 IIR 第 0 组 B0 系数设置寄存器（IIR_G0B0R）

- 名称：IIR Group 0 B0COEF RegNster
- 偏移地址：0x30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0B0COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0B0COEF															
R/W															
0x0															

字段	说明
[26:0] G0B0COEF	**IIR 第 0 组 B0 系数设置(IIR Group 0 B0COEF RegNster)** IIR 第 0 组 B0 系数设置（最多有 4 组）

13.5.2.7 IIR 第 0 组 B1 系数设置寄存器（IIR_G0B1R）

- 名称：IIR Group 0 B1COEF RegNster
- 偏移地址：0x34
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0B1COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0B1COEF															
R/W															
0x0															

字段	说明
[26:0] G0B1COEF	**IIR 第 0 组 B1 系数设置(IIR Group 0 B1COEF RegNster)** IIR 第 0 组 B1 系数设置（最多有 4 组）

13.5.2.8 IIR 第 0 组 B2 系数设置寄存器（IIR_G0B2R）

- 名称：IIR Group 0 B2COEF RegNster
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0B2COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0B2COEF															
R/W															
0x0															

字段	说明
[26:0] G0B2COEF	**IIR 第 0 组 B2 系数设置(IIR Group 0 B2COEF RegNster)** IIR 第 0 组 B2 系数设置（最多有 4 组）

13.5.2.9 IIR 第 0 组 B3 系数设置寄存器（IIR_G0B3R）

- 名称：IIR Group 0 B3COEF RegNster
- 偏移地址：0x3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0B3COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0B3COEF															
R/W															
0x0															

字段	说明
[26:0] G0B3COEF	**IIR 第 0 组 B3 系数设置(IIR Group 0 B3COEF RegNster)** IIR 第 0 组 B3 系数设置（最多有 4 组）

13.5.2.10 IIR 第 0 组 B4 系数设置寄存器（IIR_G0B4R）

- 名称：IIR Group 0 B4COEF RegNster
- 偏移地址：0x40
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0B4COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0B4COEF															
R/W															
0x0															

字段	说明
[26:0] G0B4COEF	**IIR 第 0 组 B4 系数设置(IIR Group 0 B4COEF RegNster)** IIR 第 0 组 B4 系数设置（最多有 4 组）

13.5.2.11 IIR 第 0 组 A1 系数设置寄存器（IIR_G0A1R）

- 名称：IIR Group 0 A1COEF RegNster
- 偏移地址：0x50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0A1COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0A1COEF															
R/W															
0x0															

字段	说明
[26:0] G0A1COEF	**IIR 第 0 组 A1 系数设置(IIR Group 0 A1COEF RegNster)** IIR 第 0 组 A1 系数设置（最多有 4 组）

13.5.2.12 IIR 第 0 组 A2 系数设置寄存器（IIR_G0A2R）

- 名称：IIR Group 0 A2COEF RegNster
- 偏移地址：0x54
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0A2COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0A2COEF															
R/W															
0x0															

字段	说明
[26:0] G0A2COEF	**IIR 第 0 组 A2 系数设置(IIR Group 0 A2COEF RegNster)** IIR 第 0 组 A2 系数设置（最多有 4 组）

13.5.2.13 IIR 第 0 组 A3 系数设置寄存器（IIR_G0A3R）

- 名称：IIR Group 0 A3COEF RegNster
- 偏移地址：0x58
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0A3COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0A3COEF															
R/W															
0x0															

字段	说明
[26:0] G0A3COEF	**IIR 第 0 组 A3 系数设置(IIR Group 0 A3COEF RegNster)** IIR 第 0 组 A3 系数设置（最多有 4 组）

13.5.2.14 IIR 第 0 组 A4 系数设置寄存器（IIR_G0A4R）

- 名称：**IIR Group 0 A4COEF RegNster**
- 偏移地址：**0x5C**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					G0A4COEF										
					R/W										
					0x0										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G0A4COEF															
R/W															
0x0															

字段	说明
[26:0] G0A4COEF	**IIR 第 0 组 A4 系数设置(IIR Group 0 A4COEF RegNster)** IIR 第 0 组 A4 系数设置（最多有 4 组）

14 模数转换器（ADC）

14.1 简介

芯片包含 4 个 ADC 内核，每路 ADC 由一个 13 位循环模数转换器组成，每个 ADC 具有各自的控制电路，可以在同步模式下运行，也可以在非同步模式下运行。

每个 ADC 具有多达 20 个复用通道，包括 16 个外部通道，还有 4 个内部通道，内部通道与 16 个外部通道共享。各种不同通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个或多个 16 位数据寄存器中。

每路 ADC 具有模拟看门狗功能，允许应用检测输入电压是否超过了用户定义的阈值上限或下限。每路 ADC 内置硬件过采样器，可以提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

14.2 主要特性

ADC 主要具有以下特性：

- 高性能特性
  - 支持 4 路 ADC，可以在同步模式下运行
  - 每路 ADC 连接到 16 个外部通道（5 个快速通道+11 个慢速通道）+ 4 个内部通道
  - 每路 ADC 支持 6 个快速通道(全为外部通道)+14 个慢速通道(外部通道+内部通道)
  - 每路 ADC 支持独立设置各通道的单端/差分输入
  - 每路 ADC 支持独立设置各通道的采样时间
  - 每路 ADC 支持多达 16 个常规转换
  - 每路 ADC 支持多达 4 个注入转换（模拟输入分配为常规转换或注入转换完全可配置）
  - 每路 ADC 支持 20 路 DMA 搬运数据，供常规转换使用
  - 每路 ADC 支持 4 个专用数据寄存器，供注入转换使用
- 过采样器
  - 每路 ADC 支持 16 位数据寄存器，数据位数最多为 16 位
  - 每路 ADC 支持 2~256 倍过采样
  - 每路 ADC 数据右移可编程
- 数据预处理
  - 每路 ADC 支持增益补偿，最多支持 4 组补偿系数
  - 每路 ADC 支持偏置补偿，最多支持 4 组补偿系数
- 转换启动
  - 每路 ADC 支持通过软件启动常规转换和注入转换
  - 每路 ADC 支持通过具有可配置极性的硬件触发事件（内部定时器事件或 GPIO 输入事件）启动常规转换和注入转换
- 转换模式
  - 每路 ADC 均可转换单个通道，也可扫描一个通道序列
  - 每路 ADC 在单次模式下在每次触发时单次地转换选定的输入

- 每路 ADC 在连续模式下在每次触发时连续地转换选定的输入
- 每路 ADC 在不连续模式在每次触发时分组地转换选定的输入
- 每路 ADC 支持在准备就绪、常规/注入转换结束、常规/注入序列转换结束，模拟看门狗 0/1/2 或数据溢出事件时生成中断

- 每个 ADC 支持 3 个模拟看门狗，模拟看门狗可以过滤并忽略超出范围的数据
- ADC 输入范围：GND≤V_{IN}≤3V

14.3 结构框图

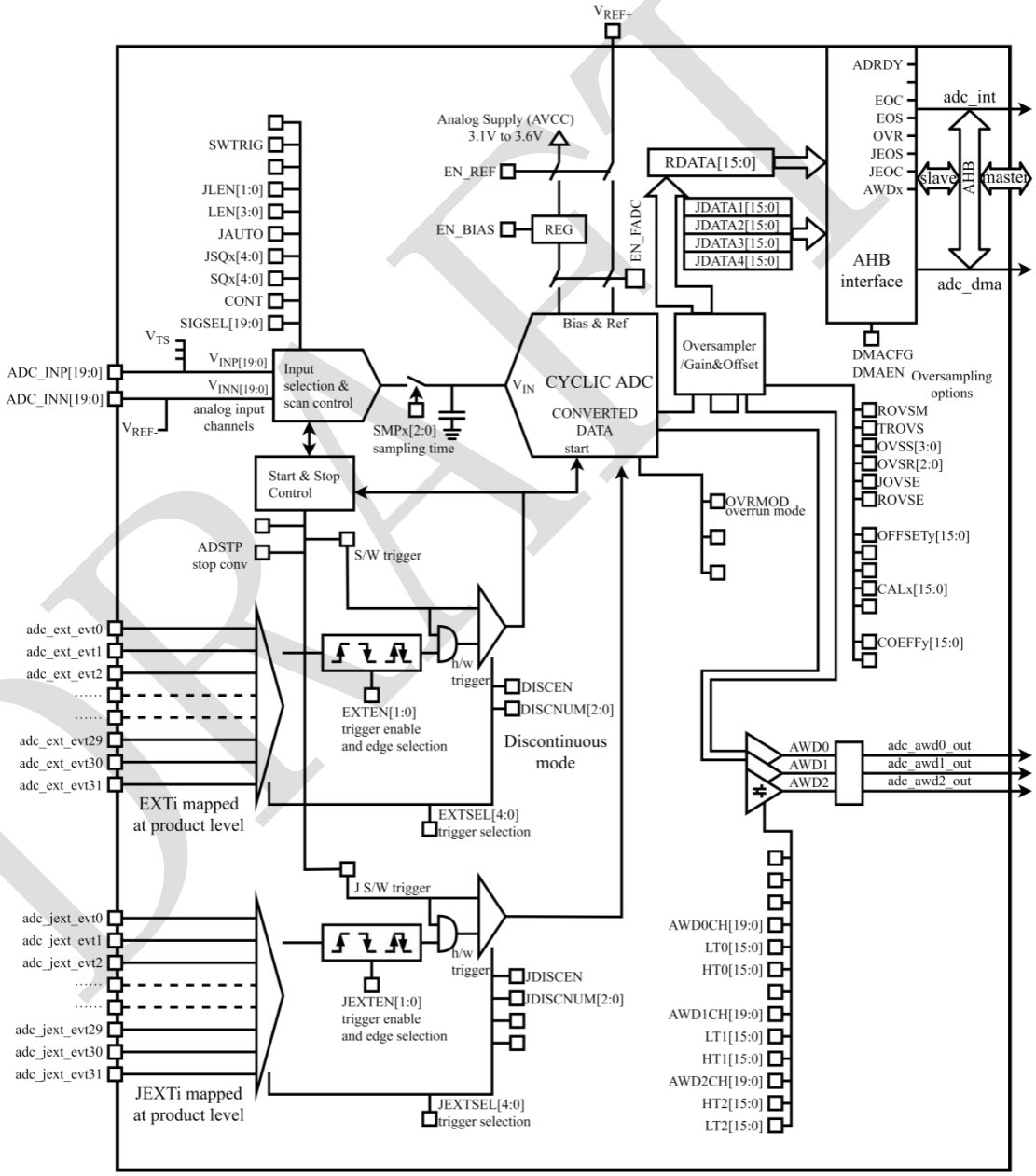


图 14-1　ADC 结构框图

14.4 功能描述

14.4.1 ADC 引脚和内部信号

表 14-1 ADC 内部输入/输出信号

内部信号名称	信号类型	说明
adcx_ext_evt[31:0]	输入	共有多达 32 个外部触发事件用于常规转换，这些事件输入对 ADC0~ADC3 一致。
adcx_jext_evt[31:0]	输入	共有多达 32 个外部触发事件用于注入转换，这些事件输入对 ADC0~ADC3 一致。
adcx_awd0_out adcx_awd1_out adcx_awd2_out	输出	ADCx 内部模拟看门狗输出信号，连接到片上定时器
adcx_norm_int	输出	ADCx 常规中断
adcx_samp_int	输出	ADCx 采样中断
adcx_half_int	输出	ADCx DMA 半完成中断
adcx_full_int	输出	ADC xDMA 完成中断
adc_hclk	输入	ADCx 总线时钟
adc_clk	输入	ADCx 功能时钟（固定为 60M）

表 14-2 ADC 输入/输出引脚

引脚名称	信号类型	说明
V_{REF+}	参考基准	ADC 的基准电压，典型值为 2.9V，由片内参考电压加 Buffer 产生
V_{DDA}	模拟电源	模拟电源，典型值为 3.3 V
V_{SSA}	模拟地	模拟地，典型值为 0 V
V_{INP}[19:0]	每路 ADC 正输入	连接到外部通道 ADCx_IN[i]或内部通道
V_{INN}[19:0]	每路 ADC 负输入	连接到外部通道 ADCx_IN[i+1]或内部通道
ADCx_IN[16:1]	外部输入信号	每路 ADC 支持 16 个外部输入通道： 16 个外部通道（ADCx_IN[1:16]，x=0~3）其中包括 ➢ 5 个快速通道（ADCx_IN[1:5]） ➢ 11 个慢速通道（ADCx_IN[6:16]） 输入信号范围为（0，3）V

14.4.2 ADC 连接关系

ADCx（x=0~3）紧密耦合并共用一些外部通道，如图 14-2 所示。

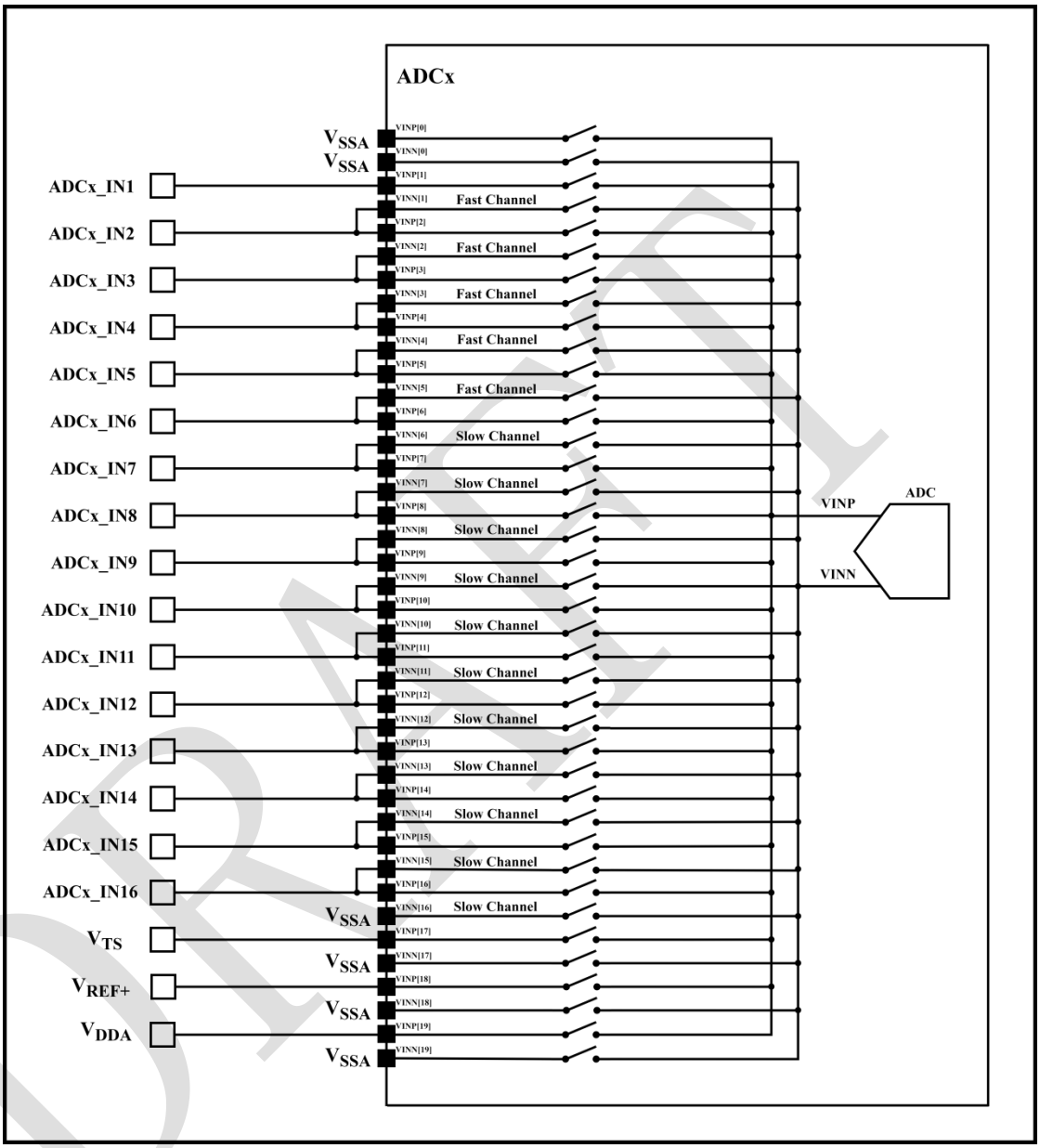


图 14-2 ADCx（x=0~3）连接关系图

14.4.3 单端和差分输入通道

可通过写入 ADCx_SIGSEL.SIGSEL 将通道配置为单端输入或差分输入。必须在禁用 ADC（ADCx_CR.ADEN = 0）时写入此配置。

在单端输入模式下，通道“i”转换的模拟电压是外部电压 ADCx_IN[i]（连接到 VINP[i]，ADC 正输入）和 VSSA（连接到 VREF-，ADC 负输入）之差。

在差分输入模式下，通道“i”转换的模拟电压是外部电压 ADCx_IN[i]（连接到 VINP[i]，ADC

正输入）和 ADCx_IN[i+1]（连接到 VINN[i]，ADC 负输入）之差。

差分模式的输出数据是无符号数据，当 VINP[i]为 VREF-、当 VINN[i]为 VREF+时，输出数据为 0x0000（16 位分辨率模式）；当 VINP[i]为 VREF+、当 VINN[i]为 VREF-时，输出数据为 0x1FFF。其中，VREF-为 0V，VREF+为 3V。

$$转换后的值 = \frac{ADC\_Full\_Scale}{2} \times \left[1 + \frac{V_{INP} - V_{INN}}{V_{REF+}}\right]$$

当 ADC 配置为差分模式时，两路输入的偏置电压均应为 $V_{REF+}/2$，输入信号应为差分信号（共模电压应固定）。

内部通道用于读取芯片内置温度传感器输出电压，仅在单端模式下使用。

注：在差分输入模式下配置通道“i”时，其负输入电压 $V_{INN[i]}$ 连接到 $V_{INP[i+1]}$，因此，通道“i+1”在单端模式或差分模式下不再可用，不应再次进行转换。

14.4.4 ADC 开关控制

可以使用 ADEN 控制位使能或禁止 ADC：

- ADCx_CR.ADEN = 1 使能 ADC。ADC 准备就绪后，将置位标志 ADCx_ISR.ADRDY。
- ADCx_CR.ADEN = 0 禁用 ADC。

随后可通过将 ADCx_CR.ADSTART 置 1 开始进行常规转换，如果常规触发事件已使能，也可在外部常规触发事件时开始进行常规转换。

可通过将 ADCx_CR.JADSTART 置 1 开始进行注入转换，如果注入触发事件已使能，也可在外部注入触发事件时开始进行注入转换。

通过软件使能 ADC 的流程

1. 向 ADCx_ISR.ADRDY 写入“1”，将其清零；
2. 将 ADCx_CR.ADEN 置 1；
3. 等待直至 ADCx_ISR.ADRDY = 1（ADCx_ISR.ADRDY 会在 ADC 启动时间后置 1）。这可以使用相关的中断来实现（将 ADCx_IER.ADRDYIE 置 1）；
4. 向 ADCx_ISR.ADRDY 写入“1”，将其清除（可选）。

通过软件禁止 ADC 的流程

1. 检查 ADCx_CR.ADSTART 和 ADCx_CR.JADSTART 是否均为 0，以确保当前未执行任何转换。如果需要，可将 ADCx_CR.ADSTP 置 1，并将 ADCx_CR.JADSTP 置 1，随后等待至 ADCx_CR.ADSTP = 0 且 ADCx_CR.JADSTP = 0，以停止任何正在进行的常规转换和注入转换；
2. 将 ADCx_CR.ADEN 置 0。

注：使能 ADCx_CR.ADEN 控制位必须要在使能 ADCx_CFGR1.CHEN、ADCx_CFGR1.REFEN 以及 ADCx_CFGR1.BIASEN 之后，否则 ADC 转换数据可能出错。

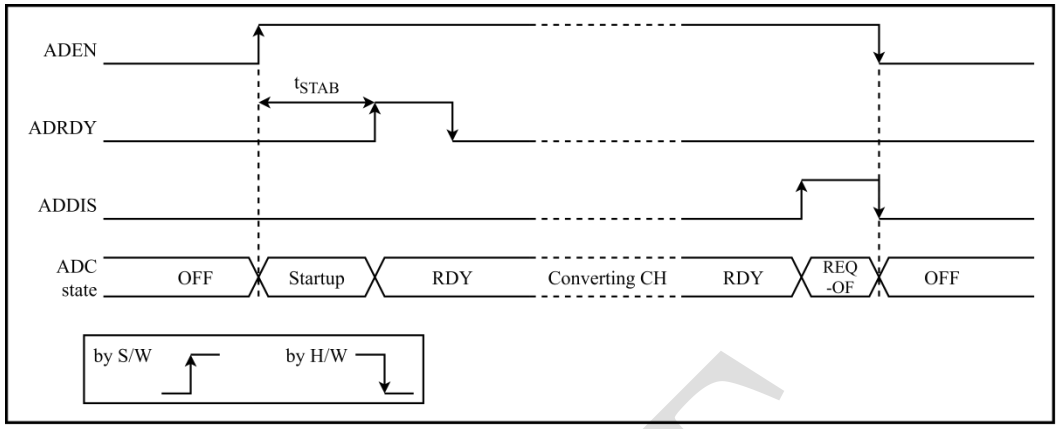


图 14-3　使能/禁止 ADC

14.4.5　写入 ADC 控制位时的限制

仅当 ADC 已禁止时（ADCx_CR.ADEN 必须等于 0），软件才可以写入 RCU 控制位以配置和启用 ADC 时钟（请参见 RCU 部分），写入 ADCx_SIGSEL.SIGSEL 和 ADCx_CR.ADEN。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADEN 必须等于 1）时，才允许软件向 ADCx_CR.ADSTART、ADCx_CR.JADSTART 执行写操作。

对于 ADCx_CFGR0，ADCx_SMPRx，ADCx_TRy，ADCx_SQRy，ADCx_JSQR，ADCx_OFRy，ADCx_GCRy 和 ADCx_IER 寄存器的所有其他控制位：

- 对于与常规转换配置相关的控制位，仅当 ADC 已使能（ADCx_CR.ADEN = 1）且未进行常规转换（ADCx_CR.ADSTART 必须等于 0）时，才允许软件对这些位执行写操作。
- 对于与注入转换配置相关的控制位，仅当 ADC 已使能（ADCx_CR.ADEN = 1）且未进行注入转换（ADCx_CR.JADSTART 必须等于 0）时，才允许软件对这些位执行写操作。

仅当 ADC 已使能且没有待处理的禁止 ADC 的请求（ADCx_CR.ADSTART 或 ADCx_CR.JADSTART 必须等于 1)时，软件才可以对 ADCx_CR.ADSTP 或 ADCx_CR.JADSTP 控制位执行写操作。

如果 ADC 已使能（ADCx_CR.ADEN = 1），软件可随时对 ADCx_JSQR 寄存器执行写操作。

注：没有硬件保护可以防止执行此类禁止的写操作，ADC 行为可能进入未知状态。要从此状态恢复，必须禁止 ADC（将 ADCx_CR.ADEN 清 0，并且将 ADCx_CFGR0 寄存器的所有位都清零）。

14.4.6　通道选择

每个 ADC 最多有 20 个多路复用通道：

- 来自 GPIO PAD 的 5 个快速通道外部输入（ADCx_IN[1:5]）
- 来自 GPIO PAD 的 11 个慢速通道外部输入（ADCx_IN[6:16]）
- ADC 连接到 4 路内部模拟输入
  - 内部模拟地（V_{SSA}）连接到 V_{INP}[0]

- 内部温度传感器（V_{TS}）连接到 V_{INP}[17]
- 内部参考电压（V_{REF+}）连接到 V_{INP}[18]
- 内部模拟电源（V_{DDA}）连接到 V_{INP}[19]

可以将转换分为两组：常规转换和注入转换。每个组包含一个转换序列，该序列可按任何顺序在任何通道上完成。例如，可以按以下顺序实现转换序列：ADCx_IN[3]，ADCx_IN[8]，ADCx_IN[2]，ADCx_IN[2]，ADCx_IN[1]，ADCx_IN[2]，ADCx_IN[2]，ADCx_IN[11]。

- 一个常规转换组最多由 16 个转换构成。必须在 ADCx_SQRy 寄存器中选择转换序列的常规转换及其顺序。常规转换组中的转换总数必须写入 ADCx_LR.LEN。
- 一个注入转换组最多由 4 个转换构成。必须在 ADCx_JSQR 寄存器中选择转换序列的注入转换及其顺序。注入转换组中的转换总数必须写入 ADCx_JLR.JLEN。

不可以在常规转换进行中对 ADCx_SQRy 寄存器进行修改。如果要修改 ADCx_SQRy 寄存器，必须先写入 ADCx_CR.ADSTP = 1 停止 ADC 常规转换。

进行注入转换时，可实时修改 ADCx_JSQR 寄存器。

温度传感器输出电压 V_{TS} 连接到通道 V_{INP}[17]，在进行温度测量时，一般建议开启 ADCBUF，否则需要增大 ADC 采样时间来保证 ADC 有足够的建立时间来获取 V_{TS} 电压。

要对其中一条内部模拟通道进行转换，必须先对 RCU 中的寄存器进行配置，以使能相应的模拟源。

14.4.7 可独立设置各通道采样时间

开始转换之前，ADC 必须在待测电压源与 ADC 内置采样电容之间建立直接连接。该采样时间必须足以使输入电压源为采样电容充电至输入电压水平。

对各通道进行采样时可以使用不同的采样时间，采样时间可通过 ADCx_SMPRx.SMP 进行编程。可选采样时间值如下：

- SMP = 000：2 个 ADC 时钟周期
- SMP = 001：6 个 ADC 时钟周期
- SMP = 010：14 个 ADC 时钟周期
- SMP = 011：30 个 ADC 时钟周期
- SMP = 100：62 个 ADC 时钟周期
- SMP = 101：126 个 ADC 时钟周期
- SMP = 110：254 个 ADC 时钟周期
- SMP = 111：510 个 ADC 时钟周期

总转换时间的计算如下：
TCONV =采样时间+ 10 个 ADC 时钟周期

例：adc_clk = 60 MHz，采样时间为 2 个 ADC 时钟周期：
TCONV =（2 + 10）ADC 时钟周期= 12 个 ADC 时钟周期= 200 ns

ADC 通过将状态位 ADCx_ISR.EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。

快速通道和慢速通道的采样时间限制：
对于每条通道必须对 ADCx_SMPRx.SMP 位进行编程，以符合数据手册 ADC 特性部分规定的

最短采样时间要求。

14.4.8 单次转换模式

单次转换模式下，ADC 会将通道的所有转换执行一次。ADCx_CFGR0.CONT 为 0 时，可通过以下方式启动此模式：

- 将 ADCx_CR.ADSTART 置 1（用于常规转换，需要选择软件触发）
- 将 ADCx_CR.JADSTART 置 1（用于注入转换，需要选择软件触发）
- 外部硬件触发事件（用于常规转换或注入转换）

触发外部事件之前，ADCx_CR.ADSTART 位或 ADCx_CR.JADSTART 位必须置 1。

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADCx_DR 寄存器中
- ADCx_ISR.EOC（常规转换结束）标志位置 1
- ADCx_IER.EOCIE 使能位置 1 时将产生中断

在注入序列中，每次转换完成后：

- 转换数据存储在四个 16 位 ADCx_JDRy 寄存器的其中之一中
- ADCx_ISR.JEOC（注入转换结束）标志位置 1
- ADCx_IER.JEOCIE 使能位置 1 时将产生中断

常规序列完成后：

- ADCx_ISR.EOS（常规序列结束）标志位置 1
- ADCx_IER.EOSIE 使能位置 1 时将产生中断

注入序列完成后：

- ADCx_ISR.JEOS（注入序列结束）标志位置 1
- ADCx_IER.JEOSIE 使能位置 1 时将产生中断

随后，ADC 会停止工作，直到发生新的外部常规或注入触发事件，或者 ADCx_CR.ADSTART 或 ADCx_CR.JADSTART 位再次置 1，ADC 才会再次工作。

注：如果要转换单个通道，可将序列长度编程为 1。

14.4.9 连续转换模式

连续转换模式仅适用于常规转换。

在连续转换模式下，如果发生软件或硬件的常规触发事件，ADC 将对通道的所有常规转换执行一次，随后会自动重启并持续执行序列的每个转换。ADCx_CFGR0.CONT 为 1 时，可通过外部触发事件或将 ADCx_CR.ADSTART 位置 1 来启动此模式。

在常规转换序列中，每次转换完成后：

- 转换数据存储在 16 位 ADCx_DR 寄存器中
- ADCx_ISR.EOC（转换结束）标志位置 1
- ADCx_IER.EOCIE 使能位置 1 时将产生中断

转换序列完成后：

- ADCx_ISR.EOS（序列结束）标志位置 1
- ADCx_IER.EOSIE 使能位置 1 时将产生中断

随后，会立即开启新转换序列，ADC 会继续重复执行转换序列。

注：如果要转换单个通道，可将序列长度编程为 1。

不可以同时使能不连续模式和连续模式：禁止同时将 ADCx_CFGR0.DISCEN 和 ADCx_CFGR0.CONT 位置 1。

注入转换不能连续转换。唯一例外的是，在连续转换模式下（ADCx_CFGR0.JAUTO）注入转换配置为在常规转换后的自动转换。

14.4.10 开始转换

软件通过将 ADCx_CR.ADSTART 置 1 的方式开始进行 ADC 常规转换。

ADCx_CR.ADSTART 置 1 后，会开始进行常规转换：

- ADCx_LR.EXTEN = 0x0（软件触发事件），常规转换立即开始
- ADCx_LR.EXTEN != 0x0（硬件触发事件），常规转换在所选常规触发事件的下一个有效沿后开始

软件通过将 ADCx_CR.JADSTART 置 1 的方式开始进行 ADC 注入转换。

ADCx_CR.JADSTART 置 1 后，会开始进行注入转换：

- ADCx_JLR.JEXTEN = 0x0（软件触发事件），注入转换立即开始
- ADCx_JLR.JEXTEN != 0x0（硬件触发事件），注入转换在所选注入触发事件的下一个有效沿后开始

注：在自动注入模式（ADCx_CFGR0.JAUTO = 1）下，使用 ADCx_CR.ADSTART 位开始常规转换，然后再进行自动注入转换（ADCx_CR.JADSTART 必须保持为 0）。

ADCx_CR.ADSTART 和 ADCx_CR.JADSTART 指示 ADC 当前是否处在工作状态。可以在 ADCx_CR.ADSTART = 0 且 ADCx_CR.JADSTART = 0（指示 ADC 处于空闲状态）时重新配置 ADC。

ADCx_CR.ADSTART 通过硬件清除：

- 在软件触发的单次模式下（ADCx_CFGR0.CONT = 0，ADCx_LR.EXTEN = 0x0，ADCx_CFGR0.DISCEN = 0）
  - 在转换序列结束后清零（ADCx_ISR.EOS 标志位置 1）
- 在软件触发的不连续转换模式下（ADCx_CFGR0.CONT = 0，ADCx_LR.EXTEN = 0x0，ADCx_CFGR0.DISCEN = 1）
  - 在转换子组结束后清零（ADCx_ISR.EOS 标志位置 1）
- 在所有其他情况下（ADCx_CFGR0.CONT = x，ADCx_LR.EXTEN = x）
  - 软件配置 ADCx_CR.ADSTP = 1 并执行之后清零（ADCx_CR.ADSTP 位清 0）

注：在连续模式下（ADCx_CFGR0.CONT = 1），由于序列会自动重新启动，因此当 ADCx_ISR.EOS 标志位置 1 时，ADCx_CR.ADSTART 位不会通过硬件清除。

如果在单次模式下选择了硬件事件触发（ADCx_CFGR0.CONT = 0 且 ADCx_LR.EXTSEL != 0x00），当 ADCx_ISR.EOS 标志位置 1 时，ADCx_CR.ADSTART 位不会通过硬件清除，这样软件无需为下一个硬件触发事件再次将 ADCx_CR.ADSTART 置 1。这样可以确保不会错过任何后续的硬件触发事件。

ADCx_CR.JADSTART 由硬件清除：

- 在软件触发的单次模式下（ADCx_JLR.JEXTEN = 0x0，ADCx_CFGR0.JDISCEN = 0）
  - 在注入转换序列结束后清零（ADCx_ISR.JEOS 标志位置 1）
- 在软件触发的不连续模式下（ADCx_JLR.JEXTEN = 0x0，ADCx_CFGR0.JDISCEN = 1）
  - 在注入转换序列结束后清零（ADCx_ISR.JEOS 标志位置 1）
- 在所有其他情况下（ADCx_JLR.JEXTEN = x）
  - 软件配置 ADCx_CR.JADSTP = 1 并执行之后清零（ADCx_CR.JADSTP 清 0）

注：选择软件触发事件时，如果 ADCx_ISR.EOC 标志位仍为高，则不应将 ADCx_CR.ADSTART 位置 1。

14.4.11 停止正在进行的转换

软件决定是否停止转换，要停止正在进行的常规转换，应将 ADCx_CR.ADSTP 置 1，要停止正在进行的注入转换，应将 ADCx_CR.JADSTP 置 1。

停止转换将复位正在进行的 ADC 操作，随后可以重新配置 ADC（例如：更改通道选择或触发事件选择），为新操作做好准备。

注：可以在常规转换仍在进行时停止注入转换，反之亦然。这样可以在常规转换仍在进行时重新配置注入转换序列及触发事件，反之亦然。

如果 ADCx_CR.ADSTP 位由软件置 1，则会中止任何正在进行的常规转换，并会丢弃部分转换结果（ADCx_DR 寄存器不会更新为当前转换结果）。

如果 ADCx_CR.JADSTP 位由软件置 1，则会中止任何正在进行的注入转换，并会丢弃部分转换结果（ADCx_JDRy 寄存器不会更新为当前转换结果）。扫描序列也会中止并会复位（这意味着重启 ADC 将重新开始新的序列）。

以上执行完毕后，ADCx_CR.ADSTP / ADCx_CR.ADSTART 位（常规转换）或 ADCx_CR.JADSTP / ADCx_CR.JADSTART 位（注入转换）会由硬件清除，软件必须查询 ADCx_CR.ADSTART（或 ADCx_CR.JADSTART）直到该位为 0，然后才能判定 ADC 已完全停止运行。

注：在自动注入模式下（ADCx_CFGR0.JAUTO = 1），将 ADSTP 位置 1 会中止常规转换和注入转换（不可以使用 ADCx_CR.JADSTP 位）。

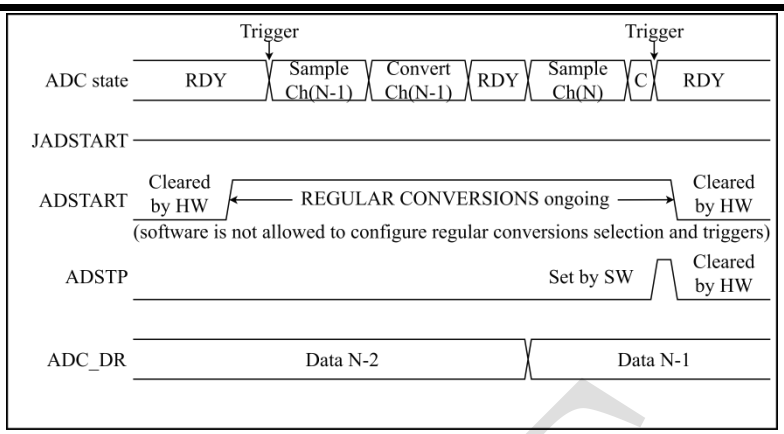


图 14-4　停止正在进行的常规转换

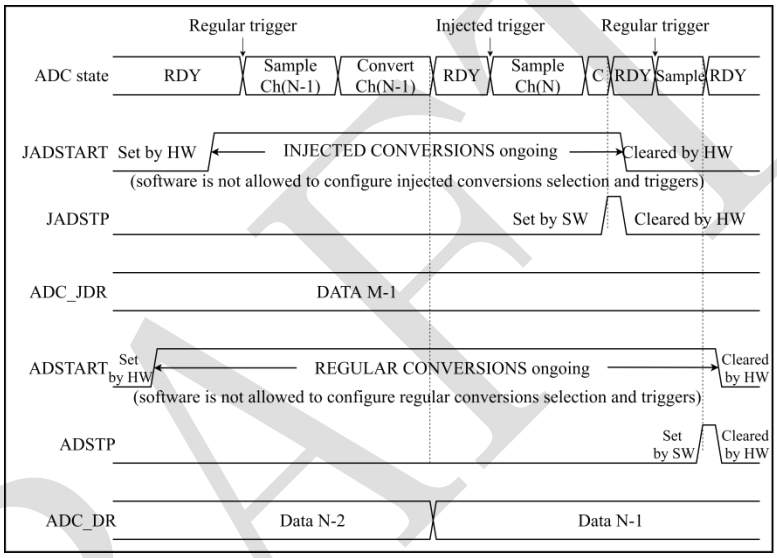


图 14-5　停止正在进行的常规转换和注入转换

14.4.12　外部触发转换和触发极性

可通过软件或外部事件（例如定时器捕获，输入引脚）触发单次转换或转换序列。如果 ADCx_LR.EXTEN 控制位（用于常规转换）或 ADCx_JLR.JEXTEN 位（用于注入转换）不配置为 0x0，则外部事件能够以所选极性触发转换。

软件将 ADCx_CR.ADSTART 置 1 后，常规触发事件选择生效；软件将 ADCx_CR.JADSTART 置 1 之后，注入触发事件选择生效。

在转换进行时发生的任何硬件触发事件会被忽略。

- 如果 ADCx_CR.ADSTART = 0，会忽略任何发生的常规触发事件。
- 如果 ADCx_CR.JADSTART = 0，会忽略任何发生的注入触发事件。

表 14-3 提供了 ADCx_LR.EXTEN 和 ADCx_JLR.JEXTEN 的值与触发极性之间的对应关系。

表 14-3　配置常规序列外部触发事件的极性

ADCx_LR.EXTEN	来源
00	禁用硬件触发事件，启用软件触发事件
01	硬件触发事件，上升沿有效
10	硬件触发事件，下降沿有效
11	硬件触发事件，上升沿和下降沿均有效

注：不能实时更改常规触发事件的极性。

表 14-4　配置注入序列的外部触发事件的极性

ADCx_JLR.JEXTEN	来源
00	禁用硬件触发事件，启用软件触发事件
01	硬件触发事件，上升沿有效
10	硬件触发事件，下降沿有效
11	硬件触发事件，上升沿和下降沿均有效

ADCx_LR.EXTSEL 和 ADCx_JLR.JEXTSEL 控制位用于选择常规序列与注入序列的外部触发事件，总共有 32 个事件来源。

常规序列转换可以被注入触发事件中断。

所有 ADC（主 ADC 与从 ADC）使用相同的外部触发事件，如图 14-6 所示。

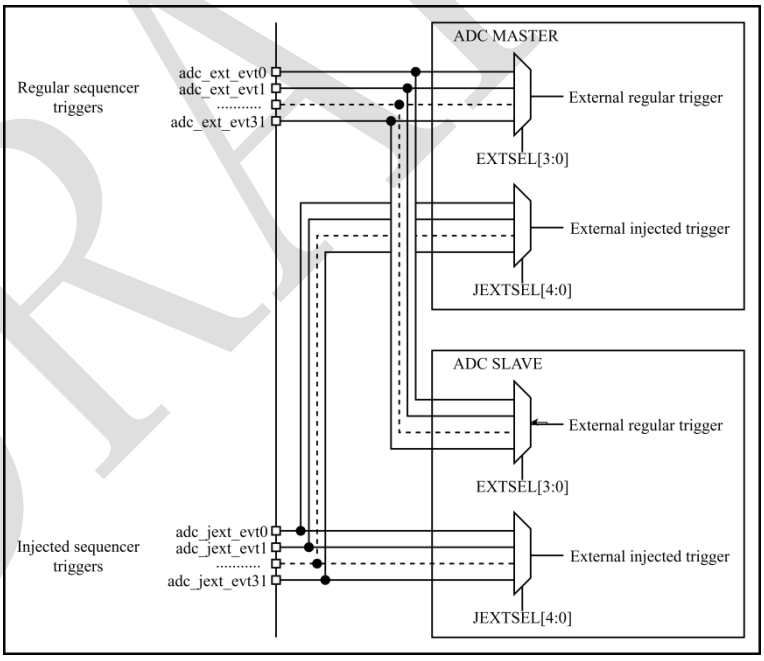


图 14-6　ADC0 与 ADC1 共用触发事件

表 14-5 至表 14-6 列出了 ADC0~ADC3 的所有外部触发事件，包括常规触发事件和注入触发事件。

表 14-5　ADC0~ADC3 常规触发事件

Name	Source	Type	EXTSEL[4:0]
adc_ext_evt0	TMR7_TRGO	来自片上定时器的内部信号	00000
adc_ext_evt1	TMR8_TRGO	来自片上定时器的内部信号	00001

adc_ext_evt2	TMR0_CC0	来自片上定时器的内部信号	00010
adc_ext_evt3	TMR1_CC0	来自片上定时器的内部信号	00011
adc_ext_evt4	TMR2_CC0	来自片上定时器的内部信号	00100
adc_ext_evt5	TMR0_TRGO	来自片上定时器的内部信号	00101
adc_ext_evt6	TMR1_TRGO	来自片上定时器的内部信号	00110
adc_ext_evt7	TMR2_TRGO	来自片上定时器的内部信号	00111
adc_ext_evt8	TMR3_CC0	来自片上定时器的内部信号	01000
adc_ext_evt9	TMR3_CC1	来自片上定时器的内部信号	01001
adc_ext_evt10	TMR3_TRGO	来自片上定时器的内部信号	01010
adc_ext_evt11	TMR4_CC0	来自片上定时器的内部信号	01011
adc_ext_evt12	TMR4_CC1	来自片上定时器的内部信号	01100
adc_ext_evt13	TMR4_TRGO	来自片上定时器的内部信号	01101
adc_ext_evt14	TMR9_CC0	来自片上定时器的内部信号	01110
adc_ext_evt15	TMR9_CC1	来自片上定时器的内部信号	01111
adc_ext_evt16	TMR9_TRGO	来自片上定时器的内部信号	10000
adc_ext_evt17	TMR10_CC0	来自片上定时器的内部信号	10001
adc_ext_evt18	TMR10_CC1	来自片上定时器的内部信号	10010
adc_ext_evt19	TMR10_CC2	来自片上定时器的内部信号	10011
adc_ext_evt20	TMR10_TRGO	来自片上定时器的内部信号	10100
adc_ext_evt21	HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	10101
adc_ext_evt22	HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	10110
adc_ext_evt23	HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	10111
adc_ext_evt24	HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	11000
adc_ext_evt25	HRPWM_ADC_TRG4	来自片上 PWM 的内部信号	11001
adc_ext_evt26	HRPWM_ADC_TRG5	来自片上 PWM 的内部信号	11010
adc_ext_evt27	HRPWM_ADC_TRG6	来自片上 PWM 的内部信号	11011
adc_ext_evt28	HRPWM_ADC_TRG7	来自片上 PWM 的内部信号	11100
adc_ext_evt29	HRPWM_ADC_TRG8	来自片上 PWM 的内部信号	11101
adc_ext_evt30	HRPWM_ADC_TRG9	来自片上 PWM 的内部信号	11110
adc_ext_evt31	PA8	外部引脚	11111

表 14-6　ADC0~ADC3 注入触发事件

Name	Source	Type	JEXTSEL[4:0]
adc_jext_evt0	TMR7_TRGO	来自片上定时器的内部信号	00000
adc_jext_evt1	TMR8_TRGO	来自片上定时器的内部信号	00001
adc_jext_evt2	TMR0_CC0	来自片上定时器的内部信号	00010
adc_jext_evt3	TMR1_CC0	来自片上定时器的内部信号	00011
adc_jext_evt4	TMR2_CC0	来自片上定时器的内部信号	00100
adc_jext_evt5	TMR0_TRGO	来自片上定时器的内部信号	00101
adc_jext_evt6	TMR1_TRGO	来自片上定时器的内部信号	00110
adc_jext_evt7	TMR2_TRGO	来自片上定时器的内部信号	00111
adc_jext_evt8	TMR3_CC0	来自片上定时器的内部信号	01000
adc_jext_evt9	TMR3_CC1	来自片上定时器的内部信号	01001

adc_jext_evt10	TMR3_TRGO	来自片上定时器的内部信号	01010
adc_jext_evt11	TMR4_CC0	来自片上定时器的内部信号	01011
adc_jext_evt12	TMR4_CC1	来自片上定时器的内部信号	01100
adc_jext_evt13	TMR4_TRGO	来自片上定时器的内部信号	01101
adc_jext_evt14	TMR9_CC0	来自片上定时器的内部信号	01110
adc_jext_evt15	TMR9_CC1	来自片上定时器的内部信号	01111
adc_jext_evt16	TMR9_TRGO	来自片上定时器的内部信号	10000
adc_jext_evt17	TMR10_CC0	来自片上定时器的内部信号	10001
adc_jext_evt18	TMR10_CC1	来自片上定时器的内部信号	10010
adc_jext_evt19	TMR10_CC2	来自片上定时器的内部信号	10011
adc_jext_evt20	TMR10_TRGO	来自片上定时器的内部信号	10100
adc_jext_evt21	HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	10101
adc_jext_evt22	HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	10110
adc_jext_evt23	HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	10111
adc_jext_evt24	HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	11000
adc_jext_evt25	HRPWM_ADC_TRG4	来自片上 PWM 的内部信号	11001
adc_jext_evt26	HRPWM_ADC_TRG5	来自片上 PWM 的内部信号	11010
adc_jext_evt27	HRPWM_ADC_TRG6	来自片上 PWM 的内部信号	11011
adc_jext_evt28	HRPWM_ADC_TRG7	来自片上 PWM 的内部信号	11100
adc_jext_evt29	HRPWM_ADC_TRG8	来自片上 PWM 的内部信号	11101
adc_jext_evt30	HRPWM_ADC_TRG9	来自片上 PWM 的内部信号	11110
adc_jext_evt31	PA9	外部引脚	11111

14.4.13 注入序列管理

触发注入模式

要使用触发注入，必须将 ADCx_CFGR0.JAUTO 清零。

1. 通过外部触发或将 ADCx_CR.ADSTART 置 1 来启动常规转换组转换。
2. 如果在常规转换组转换期间出现外部注入触发事件，或者 ADCx_CR.JADSTART 置 1，则当前的常规转换会复位，并会启动注入序列转换（所有注入转换都会转换一次）。
3. 然后，常规转换组的常规转换会从上次被中断的常规转换处恢复。
4. 如果在注入转换期间出现常规触发事件，注入转换不会中断，但在注入序列结束时会执行常规序列转换。注入序列相应的时序图如图 14-7 所示。

注：使用触发注入时，必须确保触发事件之间的间隔长于注入序列长度。例如，如果注入序列长度为 12 个 ADC 时钟周期（即，采样时间为 2 个 ADC 时钟周期的一次转换），则触发事件的最小间隔不能小于 13 个 ADC 时钟周期。

自动注入模式

如果将 ADCx_CFGR0.JAUTO 置 1，则注入组中的通道会在常规组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADCx_SQRy 和 ADCx_JSQR 寄存器中编程。

在该模式下，必须将 ADCx_CR.ADSTART 置 1 以开始常规转换，然后再进行注入转换（ADCx_CR.JADSTART 必须保持清零）。将 ADCx_CR.ADSTP 位置 1 会中止常规转换和注入转换（不得使用 ADCx_CR.JADSTP 位）。

在此模式下，必须禁止注入转换上的外部事件触发。

如果 ADCx_CFGR0.CONT 位和 ADCx_CFGR0.JAUTO 位均已置 1，则在转换注入转换之后会继续转换常规转换。

注：不能同时使用自动注入和不连续采样模式。

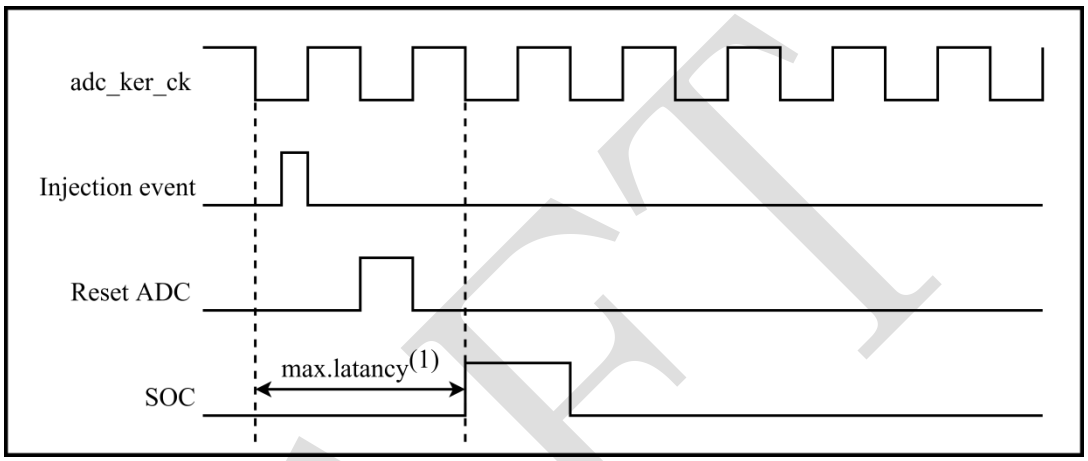


图 14-7　注入转换延迟

14.4.14　不连续模式

常规组模式

可将 ADCx_CFGR0.DISCEN 置 1 来使能此模式。

该模式用于转换含有 n（n≤8）个转换的短序列（子组），该短序列是在 ADCx_SQRy 寄存器中选择的转换序列的一部分。可通过写入 ADCx_CFGR0.DISCNUM 来指定 n 的值。

出现外部触发时，将启动在 ADCx_SQRy 寄存器中选择的接下来 n 个转换，直到序列中的所有转换均完成为止。通过 ADCx_LR.LEN 位定义总序列长度。

示例：

- ADCx_CFGR0.DISCEN = 1，n = 3，待转换的通道= 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换的通道为 1、2、3，每次转换都会生成 EOC 事件。
  - 第二次触发：转换的通道为 6、7、8，每次转换都会生成 EOC 事件。
  - 第三次触发：转换的通道为 9、10、11，每次转换都会生成 EOC 事件，并会在通道 11 转换完成后生成 EOS 事件。
  - 第四次触发：转换的通道为 1、2、3，每次转换都会生成 EOC 事件。
- ADCx_CFGR0.DISCEN = 0，待转换的通道= 1，2，3，6，7，8，9，10，11
  - 第一次触发：转换整个序列：通道 1、2、3、6、7、8、9、10 和 11。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。
  - 所有后续触发事件都将重启整个序列。

注：在不连续模式下转换常规组时，不会出现翻转（序列的最后一个子组的转换次数少于 n

次）。

转换完所有子组后，下一个触发事件将启动第一个子组的转换。在上述示例中，第四次重新转换了第一个子组中的通道 1、2 和 3。

不可以同时使能不连续模式和连续模式。如果同时使能两种模式（即 ADCx_CFGR0.DISCEN = 1、ADCx_CFGR0.CONT = 1），ADC 会认定连续模式已禁止并继续执行相关操作。

注入组模式

可将 ADCx_CFGR0.JDISCEN 置 1 来使能此模式。在出现外部注入触发事件之后，该模式下会逐通道转换在 ADCx_JSQR 寄存器中选择的序列，相当于不连续模式下常规转换“n”固定为 1 的情况。

出现外部触发事件时，将启动在 ADCx_JSQR 寄存器中选择的下一个转换，直到序列中的所有转换均完成为止。通过 ADCx_LR.JLEN 定义总序列长度。

示例：

- ADCx_CFGR0.JDISCEN = 1，待转换的通道 = 1、2、3
  - 第一次触发：转换通道 1，生成 JEOC 事件
  - 第二次触发：转换通道 2，生成 JEOC 事件
  - 第三次触发：转换通道 3，生成 JEOC 事件和 JEOS 事件

注：转换完所有注入转换后，下一个触发事件将启动第一个注入转换的转换。在上述示例中，第四次触发重新转换了第一个注入转换。

不可以同时使用自动注入模式和不连续模式：当 ADCx_CFGR0.JAUTO 置 1 时，ADCx_CFGR0.DISCEN 和 ADCx_CFGR0.JDISCEN 位必须通过软件保持清零状态。

14.4.15 转换结束

每次出现常规转换结束（EOC）事件和注入转换结束（JEOC）事件时，ADC 都会通知应用程序。

新的常规转换数据出现在 ADCx_DR 寄存器中后，ADC 会立即将 ADCx_ISR.EOC 标志位置 1。如果 ADCx_IER.EOCIE 使能位置 1，可产生中断。ADCx_ISR.EOC 标志位通过由软件向其写入 1 或读取 ADCx_DR 寄存器的方式来清零。

新的注入转换数据出现在相应的 ADCx_JDRy 寄存器中后，ADC 会立即将 ADCx_ISR.JEOC 标志位置 1。如果 ADCx_IER.JEOCIE 使能位置 1，可产生中断。ADCx_ISR.JEOC 标志位可通过由软件向其写入 1 或读取相应 ADCx_JDRy 寄存器的方式来清零。

14.4.16 转换序列结束

每次出现常规序列结束（EOS）事件和注入序列结束（JEOS）事件时，ADC 都会通知应用程序。

常规转换序列的最后一个数据出现在 ADCx_DR 寄存器中时，ADC 会立即将 EOS 标志位置 1。如果 ADCx_IER.EOSIE 使能位置 1，可产生中断。ADCx_ISR.EOS 标志位可通过由软件向其写入 1 的方式来清零。

注入转换序列的最后一个数据出现在 ADCx_JDRy 寄存器中时，ADC 会立即将 ADCx_ISR.JEOS 标志位置 1。如果 ADCx_IER.JEOSIE 使能位置 1，可产生中断。ADCx_ISR.JEOS 标志位可通过由软件向其写入 1 的方式来清零。

14.4.17 时序图示例

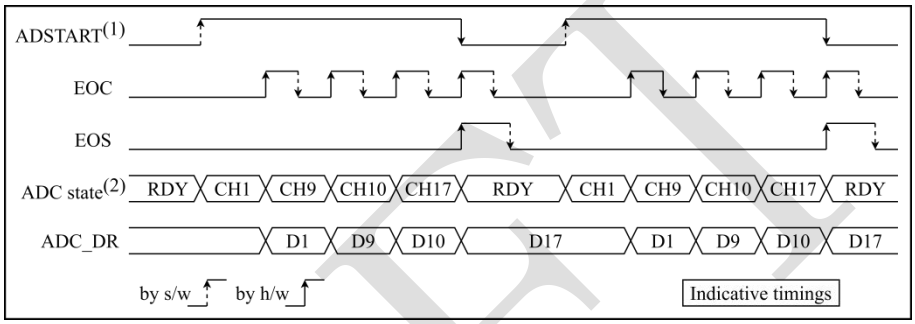


图 14-8　单次序列转换，软件触发

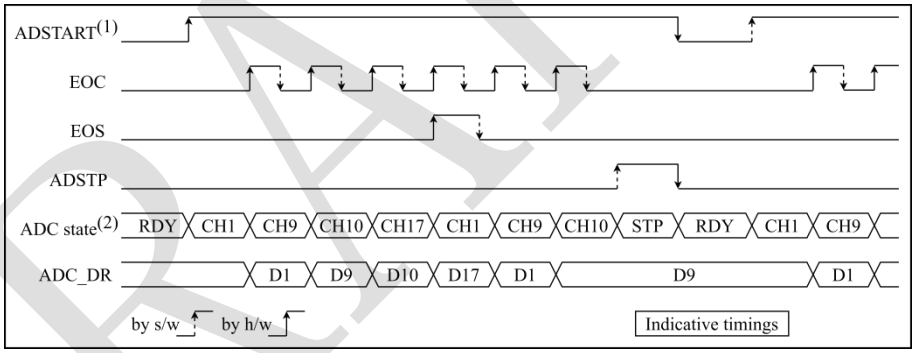


图 14-9　连续序列转换，软件触发

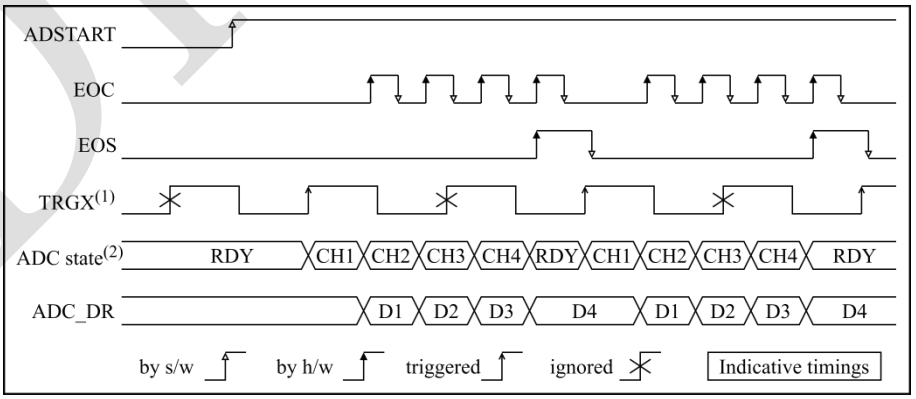


图 14-10　单次序列转换，硬件触发

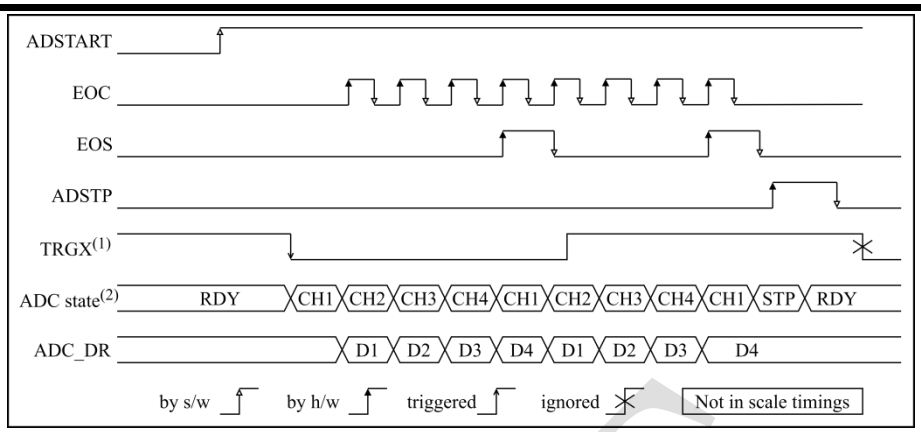


图 14-11　连续序列转换，硬件触发

14.4.18　数据管理

数据与对齐

每次常规转换通道结束时（发生 EOC 事件时），转换后数据的结果都会存储在宽度为 16 位的 ADCx_DR 数据寄存器中。

每次注入转换通道结束时（发生 JEOC 事件时），转换后数据的结果都会存储在宽度为 16 位的相应 ADCx_JDRy 数据寄存器中。

正常模式下和过采样模式下，ADC 数据均为右对齐。

偏置补偿

每次转换结束时，转换后的数据都会经过偏置和增益补偿后，存储在相应的数据寄存器。每次转换后，将使用以下公式计算数据：

校准数据 = （转换数据×增益补偿系数/8192）+ 偏置补偿系数

对各通道进行采样时可以使用不同的偏置补偿，偏置补偿系数总共四组，每组包含一个通道偏置，可通过 ADCx_OFRy 寄存器进行配置。

各通道的补偿系数可通过 ADCx_CALRx.CALx 来选择，可选的补偿系数为 ADCx_OFRy（y=0..3）的其中一组。

这种情况下，转换后的数据会减去写入到 ADCx_OFRy 中的用户自定义偏移。结果可能是负值，因此读取的数据为有符号数，最高位代表符号位。

注：过采样模式下也支持偏置补偿。对于过采样模式，会在应用 ADCx_CFGR0.OVSS 右移之后减去偏置。

模拟看门狗模式支持偏置补偿，模拟看门狗会对偏置补偿之后的数据进行比较。

增益补偿

对各通道进行采样时可以使用不同的增益补偿，增益补偿系数总共四组，每组包含一个通道增益，可通过 ADCx_GCRy 寄存器进行配置。

各通道的补偿系数可通过 ADCx_CALRx.CALx 来选择，可选的补偿系数为 ADCx_GCRy（y=0..3）的其中一组。

由于 GAIN 可以编程为 0 至 65535，因此实际增益补偿因子的范围可以为 0 至 7.999878。

在将结果数据存储到 ADCx_DR 或 ADCx_JDRy 寄存器之前，补偿单元会对 LSB-1 值进行估算，对数据进行四舍五入以最大程度地减少误差。

增益补偿对于过采样也有效。当增益补偿用于过采样模式时，增益计算会在累加和右移操作之后执行以最大程度地减少功耗（增益计算仅执行一次，而不是在每次转换时执行）。

出厂校准

芯片在出厂测试后具有自带的增益/偏置补偿系数，自带的增益/偏置补偿系数在每次上电时自动加载到相应位置并自动生效、无须用户干预。

用户可以通过修改增益/偏置补偿寄存器的系数，对 ADC 增益/偏置进行二次校准，二次校准在已校准数据上执行、不影响出厂校准结果。

如前所述、用户修改的增益补偿寄存器为 ADCx_GCRy（y=0~3），偏置补偿寄存器为 ADCx_OFRy（y=0~3），实现应用层面消除板级误差的目的。

温度传感器

芯片在出厂测试后具有自带的温度传感器的增益/偏置补偿系数，这组补偿系数在每次上电时自动加载到 ADCx_GGCTR/ADCx_GOFTR 寄存器、不过不会自动生效。

用户使用温度传感器时，需要读取 ADCx_GGCTR/ADCx_GOFTR 寄存器中的补偿系数、填入与温度传感器通道相对应的 ADCx_GCRy/ADCx_OFRy 寄存器，转换后可以读出温度传感器的数值。

ADC 溢出

如果常规转换后的数据未在新转换数据可用之前（由 CPU 或 DMA）读取，会由溢出标志位（ADCx_ISR.OVR）指示缓冲区溢出事件。

如果新转换完成时 ADCx_ISR.EOC 标志位仍为 1，则 ADCx_ISR.OVR 标志位会置 1。如果 ADCx_ISR.OVRIE 使能位置 1，可产生中断。

如果发生溢出情况，ADC 仍会保持工作状态并可继续进行转换，除非通过软件将 ADCx_CR.ADSTP 位置 1，从而停止并复位序列。

ADCx_ISR.OVR 标志位可通过软件向其写入 1 的方式来清零。

可对控制位 ADCx_CFGR0.OVRMOD 进行编程，从而配置发生溢出事件时是保留数据还是覆盖数据：

- ADCx_CFGR0.OVRMOD = 0：溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留原数据，并会丢弃新的转换结果。如果 ADCx_ISR.OVR 保持为 1，可继续正常进行转换，但还是会丢弃结果数据。
- ADCx_CFGR0.OVRMOD = 1：数据寄存器会由上一次转换结果覆盖，之前未读取的数据会丢失。如果 ADCx_ISR.OVR 保持为 1，可继续正常进行转换，并且 ADCx_DR 寄存器将始终包含最新的转换数据。

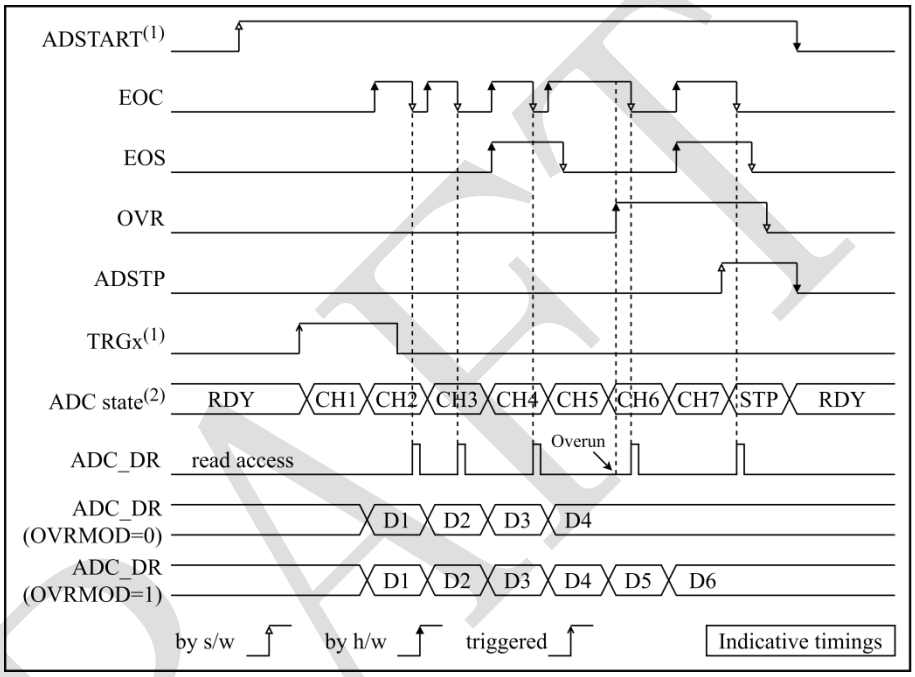


图 14-12　溢出示例

注：由于四条注入转换均有专用的数据寄存器，因此不会对注入转换进行溢出检测。

在不使用 DMA 的情况下管理转换序列

如果转换过程足够慢，则可使用软件处理转换序列。在这种情况下，软件必须使用 ADCx_ISR.EOC 标志位及其相关中断来处理各个数据。每次转换完成时，ADCx_ISR.EOC 都会置 1，并且可以读取 ADCx_DR 寄存器。ADCx_CFGR0.OVRMOD 应配置为 0，以便将溢出事件作为错误进行管理。

在不使用 DMA 且不溢出的情况下管理转换序列

ADC 在转换一个或多个通道时不是每次都读取数据的情况下，这可能会很有用（例如，存在模拟看门狗时）。在这种情况下，ADCx_CFGR0.OVRMOD 位必须配置为 1，ADCx_ISR.OVR 标志位应被软件忽略。溢出事件不会阻止 ADC 继续进行转换，ADCx_DR 寄存器始终包含最新的转换结果。

使用 DMA 管理转换

由于转换得出的数据会存储在唯一的数据寄存器中，因此，对于多个通道的转换，使用 DMA 非常有帮助。这样可以避免丢失在下一次写入之前还未被读出的 ADCx_DR 寄存器中的数据。

DMA 模式使能时，会在每次通道转换后发起 DMA 传输。DMA 传输分 20 路通道进行，每路通道可以独立配置传输地址、传输长度、传输模式，20 路通道分时进行传输。这样可以将转换后的数据从 ADCx_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果因 DMA 无法及时处理 DMA 传输请求而发生数据溢出（ADCx_ISR.OVR = 1），ADC 会停止发起 DMA 传输，新转换对应的数据不会通过 DMA 进行传输。这意味着可将传输到 RAM 的所有数据都视为有效数据。

根据 ADCx_CFGR0.OVRMOD 位的配置，可保留或覆盖数据。

DMA 传输请求会停止，直至软件将 ADCx_ISR.OVR 位清零。

根据应用用途的不同，推荐使用两种不同的 DMA 模式，使用 ADCx_TCRx.CIRC 位配置：

- DMA 单次模式（ADCx_TCRx.CIRC = 0）
  如果将 DMA 设置为传输固定数目的数据，应选择此模式。
- DMA 循环模式（ADCx_TCRx.CIRC = 1）
  如果将 DMA 设置为传输连续数目的数据，应选择此模式。

DMA 单次模式（ADCx_TCRx.CIRC = 0）

在该模式下，每次出现新的转换数据时，ADC 都会进行 DMA 传输，DMA 到达最后一个 DMA 传输操作时，即使转换已再次开始，ADC 也会停止 DMA 传输。

DMA 循环模式（ADCx_TCRx.CIRC = 1）

在该模式下，每次数据寄存器中出现新的转换数据时，ADC 都会进行 DMA 传输，即使 DMA 已到达最后一次 DMA 传输操作也不例外。这样可以处理连续的模拟输入数据流。

14.4.19 模拟看门狗

三个 AWD 模拟看门狗会监测一些通道是否保持在配置的电压范围（窗口）内。

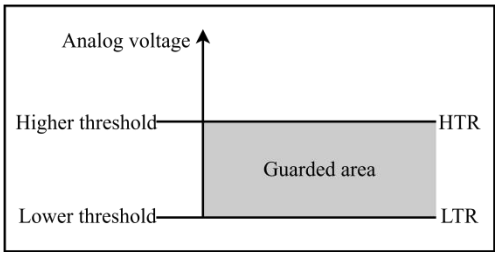


图 14-13　模拟看门狗的保护区域

AWDy 标志和中断

可通过将 ADCx_IER.AWDyIE 置 1 的方式分别为 3 个模拟看门狗使能中断。

AWDy（y = 0,1,2）标志位可通过软件向其写入 1 的方式来清零。

在增益补偿和偏移补偿之后，会将 ADC 转换结果与阈值上限和下限进行比较。

模拟看门狗的说明

模拟看门狗 0~2 非常灵活，可通过编程 ADCx_AWDyCR.AWyCH（y = 0~2）中的相应位来监测多条已选通道。

ADCx_AWDyCR.AWyCH（y = 0~2）的任意位置 1 时，会使能相应的看门狗。

阈值最高可达到 16 位（13 位分辨率，过采样，OSR=256），通过 ADCx_TR0、ADCx_TR1 和 ADCx_TR2 寄存器进行编程。

ADCx_AWDy_OUT 信号输出生成

每个模拟看门狗都关联到一个硬件触发事件 ADCx_AWDy_OUT，该信号直接连接到 HRPWM 以及 TIMER 作为事件输入。

当关联的模拟看门狗使能时，ADCx_AWDy_OUT 会激活：

- 当受监测的转换超出编程阈值时，ADCx_AWDy_OUT 会置 1。
- 在编程阈值范围内的下一受监测转换结束后，ADCx_AWDy_OUT 会复位（如果下一受监测转换仍超出编程阈值范围，该位仍保持置 1）。
- 关闭 ADC 时（将 ADCx_CR.ADEN 置 0 时）ADCx_AWDy_OUT 也保持复位状态。注意停止常规转换或注入转换（将 ADCx_CR.ADSTP 置 1 或 ADCx_CR.JADSTP 置 1）对 ADCx_AWDy_OUT 的生成没有任何影响。

注：AWDx 标志位由硬件置 1，并由软件复位；AWDx 标志位对 ADCx_AWDy_OUT 的生成没有影响（例如：如果软件未将 AWDx 标志清除，即 AWDx 标志位会保持为 1，此时 ADCx_AWDy_OUT 仍可翻转）。

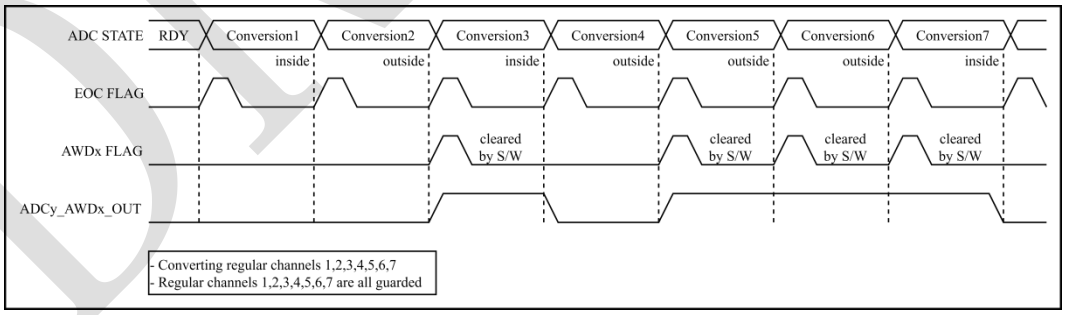


图 14-14　ADCx_AWDy_OUT 事件生成（多个常规转换上）

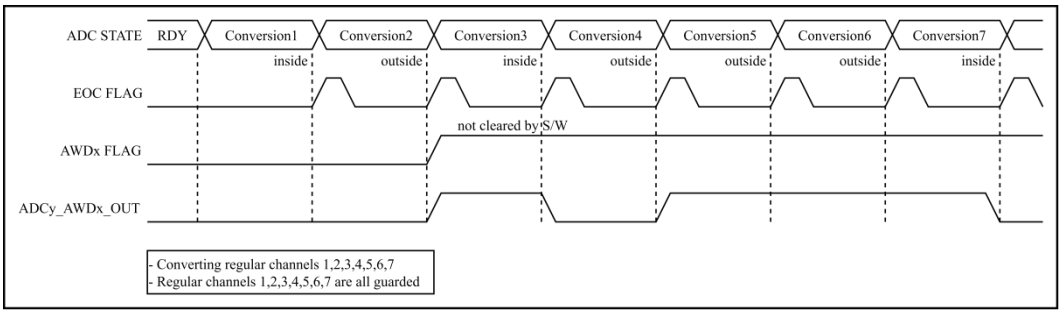


图 14-15　ADCx_AWDy_OUT 事件生成（AWDx 标志位未通过软件清零）

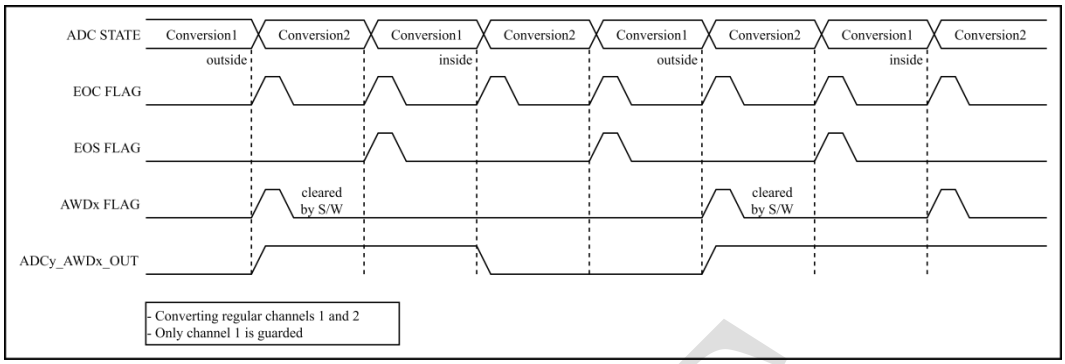


图 14-16　ADCx_AWDy_OUT 事件生成（单个常规转换上）

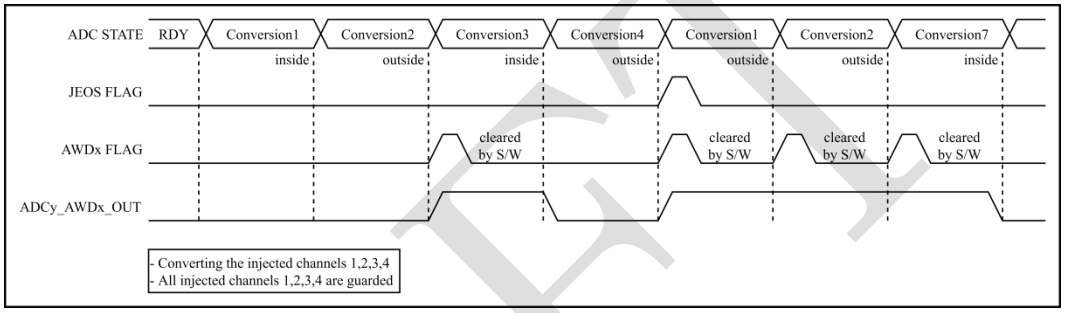


图 14-17　ADCx_AWDy_OUT 事件生成（所有注入转换上）

模拟看门狗阈值控制

ADC 正在进行转换时，同样可以更改 ADCx_TRx.LTx 和 ADCx_TRx.HTx 的配置。

加入增益和偏置补偿的模拟看门狗

加入增益和失调补偿后，模拟看门狗会在数据补偿之后再与阈值进行比较。

14.4.20　过采样器

过采样单元会进行数据预处理，以减轻 CPU 的负担。过采样单元还能处理多个转换，并计算多个转换结果的平均值，得到数据宽度增大（多达 16 位）的单个数据。

提供的结果采用以下形式，其中的 N 和 M 可以进行调整：

$$\text{结果} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{转换}(t_n)$$

允许通过硬件执行以下功能：计算平均值，降低数据速率，改善 SNR 以及基本滤波。

过采样率 N 通过 ADCx_CFGR0.OVSR 定义，范围为 2x 到 256x。分频系数 M 通过向右移位来实现（最多可移 8 位），并且通过 ADCx_CFGR0.OVSS 定义。

求和单元可得到最高 20 位（256×12 位结果）的结果，结果进行右移，会使用移位后剩下的最

低有效位四舍五入为最接近的数值，然后将得到的结果传输到 ADCx_DR 数据寄存器中。

注：如果移位后的中间结果超过 16 位，则结果将被截断而不进行饱和。

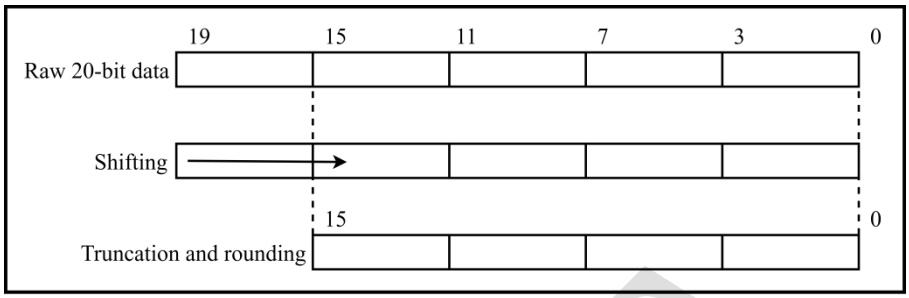


图 14-18　20-bit 到 16-bit 结果饱和处理

原始 20 位累加数据到最终 16 位计算结果的处理示例如图 14-19 所示。

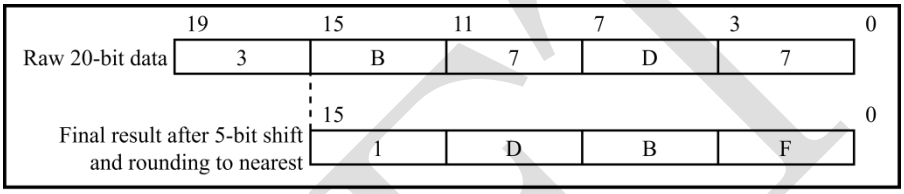


图 14-19　5-bit 移位及取整处理示例

原始转换数据等于 0x1FFF 时，各种 N 和 M 组合的数据格式在表 14-7 中所示。

表 14-7 最大输出结果与 N 和 M 的关系（灰色单元格表示截断）

Over sampling ratio	Max Raw data	1-bit shift OVSS = 0001	2-bit shift OVSS = 0010	3-bit shift OVSS = 0011	4-bit shift OVSS = 0100	5-bit shift OVSS = 0101	6-bit shift OVSS = 0110	7-bit shift OVSS = 0111	8-bit shift OVSS = 1000
2x	0x3FFE	0x1FFF	0x1000	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
4x	0x7FFC	0x3FFE	0x1FFF	0x1000	0x0800	0x0400	0x0200	0x0100	0x0080
8x	0xFFF8	0x7FFC	0x3FFE	0x1FFF	0x1000	0x0800	0x0400	0x0200	0x0100
16x	0x1FFF0	0xFFF8	0x7FFC	0x3FFE	0x1FFF	0x1000	0x0800	0x0400	0x0200
32x	0x3FFE0	0xFFF0	0xFFF8	0x7FFC	0x3FFE	0x1FFF	0x1000	0x0800	0x0400
64x	0x7FFC0	0xFFE0	0xFFF0	0xFFF8	0x7FFC	0x3FFE	0x1FFF	0x1000	0x0800
128x	0xFFF80	0xFFC0	0xFFE0	0xFFF0	0xFFF8	0x7FFC	0x3FFE	0x1FFF	0x1000
256x	0x1FFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0xFFF8	0x7FFC	0x3FFE	0x1FFF

过采样模式下的转换时序不会发生变化：在整个过采样序列中，采样时间保持不变。每完成 N 次转换都会提供一个新数据，等效延迟等于 $N \times T_{CONV} = N \times (t_{SMPL} + t_{SAR})$。各标志位的置位情况如下：

- 如果过采样结果可用，每 N 次转换后都会发生转换结束事件（EOC）
- 过采样数据序列完成后（即 Nx 序列长度次转换之后），会发生序列结束事件（EOS）

过采样模式支持的 ADC 工作模式

在过采样模式下，大部分 ADC 工作模式都会保留：

- 单次转换或连续模式转换
- 可由软件或外部事件启动 ADC 转换

- ADC 在转换过程中停止（中止）
- 通过 CPU 或 DMA 在支持溢出检测的情况下读取数据

触发式过采样

均值计算单元还可用于基本滤波，虽然它不是非常强大的滤波器（滚降缓慢、阻带衰减有限），但它可用作陷波滤波器，用于抑制恒定的寄生频率（通常来自电源或开关模式电源）。为此，可以使用 ADCx_CFGR0.TROVS 使能特定的不连续模式，以获得由用户定义，且与转换时间本身无关的过采样频率。

图 14-20 显示了在不连续模式下如何响应触发从而开始转换。

如果 ADCx_CFGR0.TROVS 置 1，则会忽略 ADCx_CFGR0.DISCEN 的内容，并将该位视为 1。

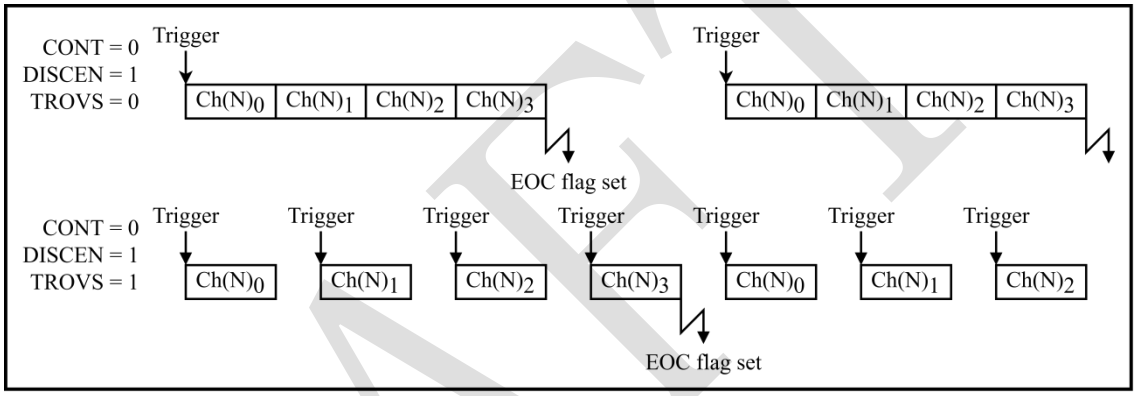


图 14-20 常规序列过采样（事件触发模式，TROVS=1）

过采样时的注入序列和常规序列

在过采样模式下，注入序列和常规序列可以执行不同操作。如果两个序列必须同时使用，则可为它们使能过采样并设定一些限制条件（与唯一的累加单元相关）。

常规序列过采样

常规过采样模式位 ADCx_CFGR0.ROVSM 定义了常规过采样序列在被注入转换中断的情况下如何恢复：

- 在连续模式下，会从上一有效数据开始重新累加（在由于注入事件而中止常规转换之前）。这样可确保在任何注入频率下均可以完成过采样（假设触发事件之间至少可完成一次常规转换）；
- 在恢复模式下，会从 0 开始重新累加（会忽略之前的转换结果）。该模式可确保所有用于过采样的数据在单个时隙内进行连续地转换。需要注意的是，注入触发事件周期必须超过过采样周期。如果该条件未得到满足，将无法完成过采样，常规序列将被停止。

图 14-21 以 4x 过采样率为例进行了说明。

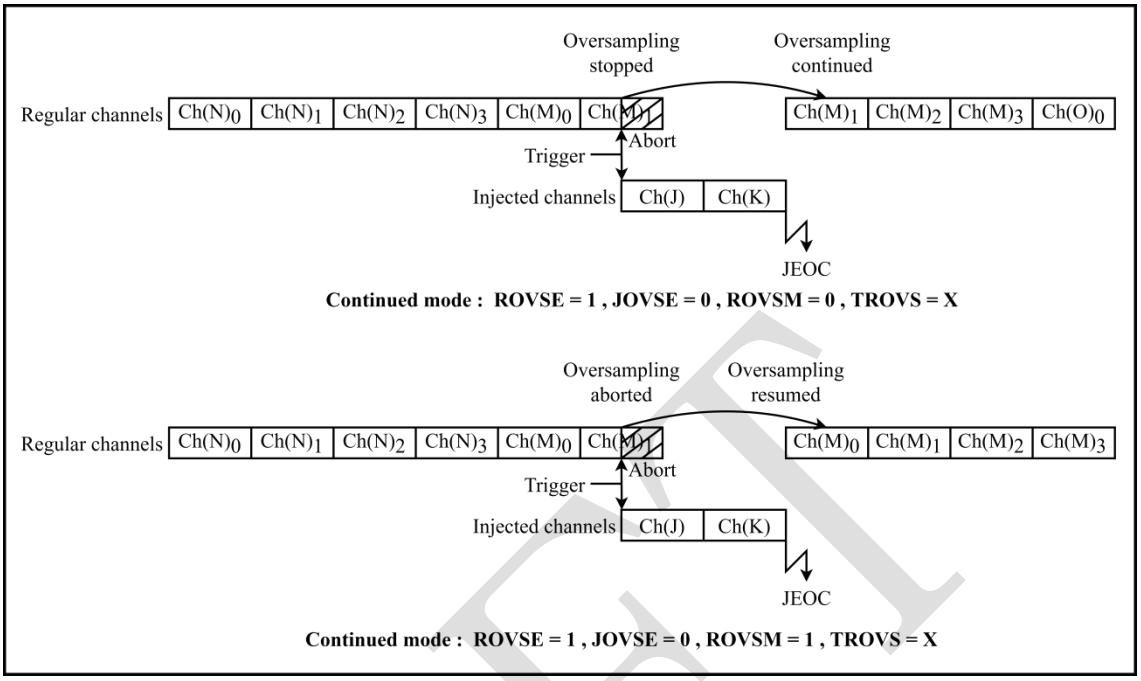


图 14-21　常规过采样模式（4x 过采样率）

仅对注入序列进行过采样

注入序列可以单独启动过采样，通过设置注入过采样位 ADCx_CFGR0.JOVSE 启动注入序列过采样。

对常规序列和注入序列进行过采样

可以将 ADCx_CFGR0.ROVSE 和 ADCx_CFGR0.JOVSE 都置 1。在这种情况下，常规过采样模式会强制进入恢复模式（忽略 ADCx_CFGR0.ROVSM），如图 14-22 所示。

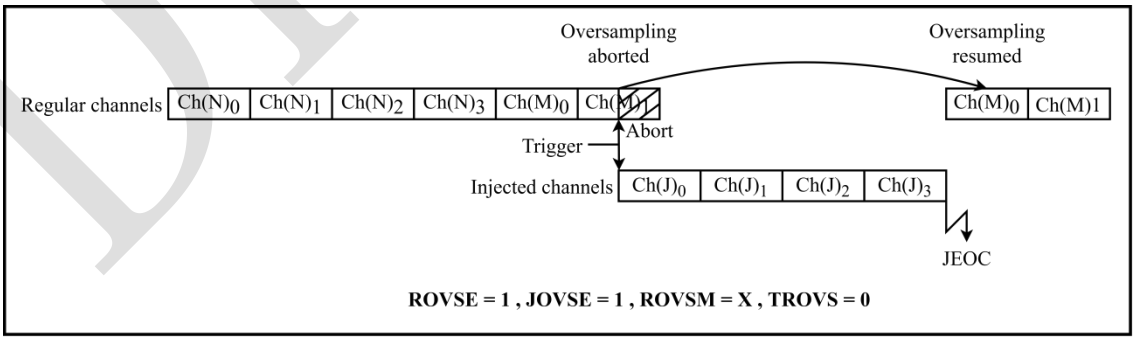


图 14-22　同时使用常规和注入过采样模式

触发式常规过采样支持注入转换

触发式常规过采样支持注入转换。在这种情况下，必须禁止注入过采样模式，并且会忽略 ADCx_CFGR0.ROVSM（强制进入恢复模式），ADCx_CFGR0.JOVSE 位必须复位。具体时序

如图 14-23 所示。

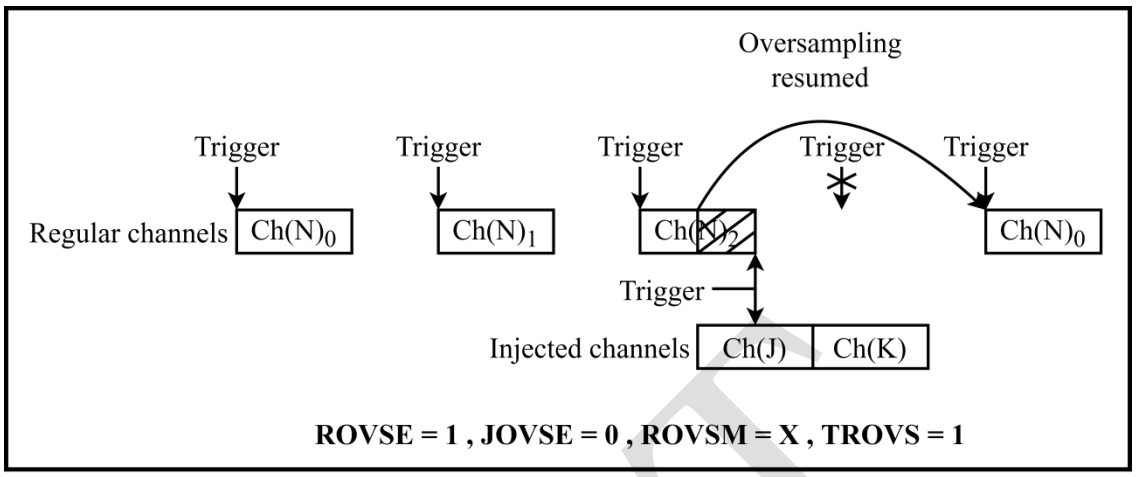


图 14-23 触发式常规过采样支持注入转换

自动注入模式

可以对自动注入序列进行过采样，并将所有转换结果存储在寄存器中，以节省 DMA 资源。使用该模式的前提条件是常规和注入过采样均激活：ADCx_CFGR0.JAUTO = 1，ADCx_CFGR0.ROVSE = 1 且 ADCx_CFGR0.JOVSE = 1，不支持其他组合。在自动注入模式，会忽略 ADCx_CFGR0.ROVSM 位。

图 14-24 显示了自动注入模式下过采样的转换顺序。

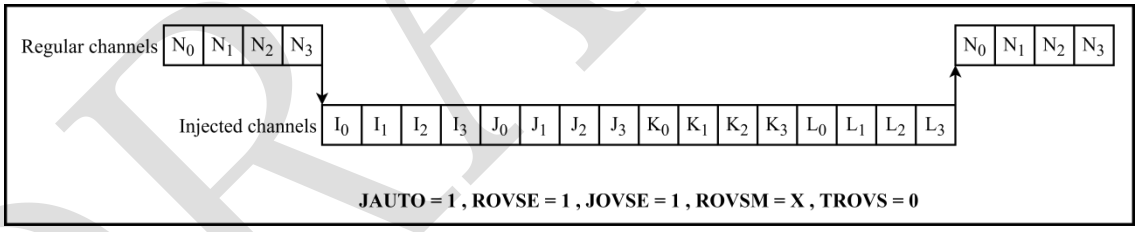


图 14-24 在自动注入模式下进行过采样

此外，还可以使用 ADCx_CFGR0.TROVS 位使能触发式过采样。在这种情况下，ADC 必须进行如下配置：ADCx_CFGR0.JAUTO = 1，ADCx_CFGR0.DISCEN = 0，ADCx_CFGR0.JDISCEN = 0，ADCx_CFGR0.ROVSE = 1，ADCx_CFGR0.JOVSE = 1 且 ADCx_CFGR0.TROVS = 1。

组合模式汇总

表 14-8 汇总了所有组合，包括不支持的模式。

表 14-8　ADC 采样模式汇总

常规过采样 ROVSE	注入过采样 JOVSE	过采样模式 ROVSM	触发式过采样 TROVS	注释
1	0	0	0	常规序列连续模式
1	0	0	1	不支持
1	0	1	0	常规序列恢复模式
1	0	1	1	触发式常规序列恢复模式
1	1	0	X	不支持
1	1	1	0	注入序列和常规序列恢复模式
1	1	1	1	不支持
0	1	X	X	注入序列过采样

14.4.21　Dual ADC 模式

芯片支持两个或多个 ADC 使用 Dual 模式：

- ADC0/ ADC1 可以在 Dual 模式下共同使用（ADC0 为主 ADC）
- ADC2/ ADC3 可以在 Dual 模式下共同使用（ADC2 为主 ADC）

在 Dual ADC 模式下、通过主 ADC 到从 ADC 的交替触发或同时触发来启动转换，具体取决于 ADCx_CCR.DUAL 所选的模式。

可以实现以下四种模式：

- 注入同步模式
- 常规同步模式
- 常规交错模式
- 交替触发模式

也可以按以下方式组合使用这些模式：

- 注入同步模式+常规同步模式
- 常规同步模式+交替触发模式

要在 Dual 模式下开始转换，用户需要对主 ADC 的 ADCx_LR.EXTEN/ ADCx_LR.EXTSEL / ADCx_JLR.JEXTEN/ADCx_JLR.JEXTSEL 位进行编程，以配置软件/硬件触发以及常规/注入触发，从 ADC 的相应域配置无效果。

在 Dual 模式下、可通过读取 ADC 通用数据寄存器（ADCx_CDR）的方式同时读取主 ADC 和从 ADC 的已转换数据。此外，还可以通过读取 Dual 模式状态寄存器（ADCx_CSR）的方式同时读取主 ADC 和从 ADC 的状态位。

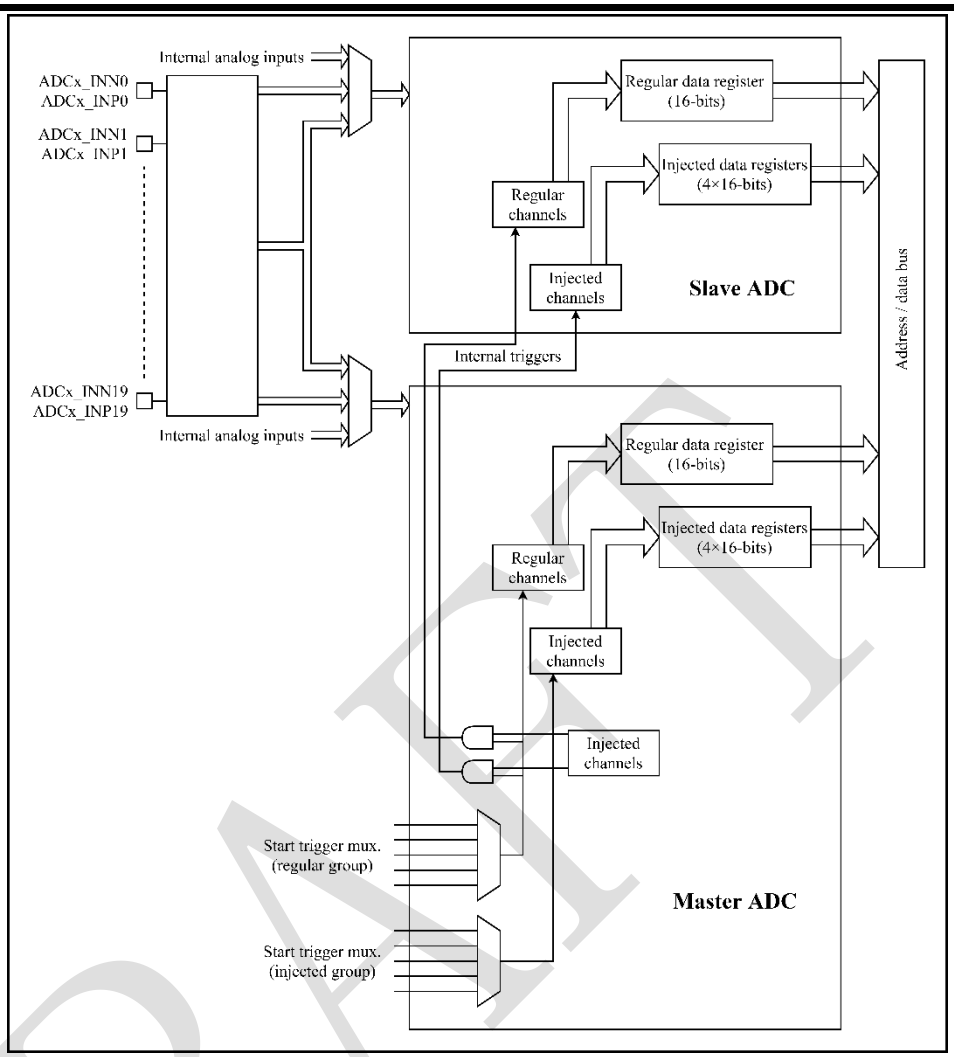


图 14-25Dual ADC 总体框图

注入同步模式

通过将 ADCx_CCR.DUAL 配置为 0x4 来选择注入同步模式。

注入同步模式可同步转换注入序列，外部触发事件来自主 ADC 的注入事件，通过 ADCx_JLR.JEXTSEL 进行选择，触发事件同步用于从 ADC。

注：不要在两个 ADC 上转换同一通道（转换同一通道时，不允许两个 ADC 采样时间重叠）。

注：在注入同步模式下，转换的序列长度必须相同，并且在序列中主 ADC 和从 ADC 的第 N 次转换必须配置为采用相同的采样时间。

注入同步模式下，常规序列可在一个或所有 ADC 上执行，它们彼此之间都是独立的，而且会在出现注入事件时中断，常规序列会在注入序列结束时恢复转换。

- 当主 ADC 上出现注入转换序列结束 JEOS 事件时，已转换数据会存储到主 ADCx_JDRy 寄存器中，并会产生 JEOS 中断标志。
- 当从 ADC 上出现注入转换序列结束 JEOS 事件时，已转换数据会存储到从 ADCx_JDRy 寄存器中，并会产生 JEOS 中断标志。
- 如果主注入序列的持续时间与从注入序列的持续时间相等，软件可以只关注两个 JEOS 中断标志中的一个（例如主 JEOS），并从主 ADCx_JDRy 和从 ADCx_JDRy 寄存器中读取

两者的已转换数据。

如果 ADCx_CFGR0.JDISCEN=1，注入序列的每个同步转换都需要在出现一个注入触发事件后才能进行。

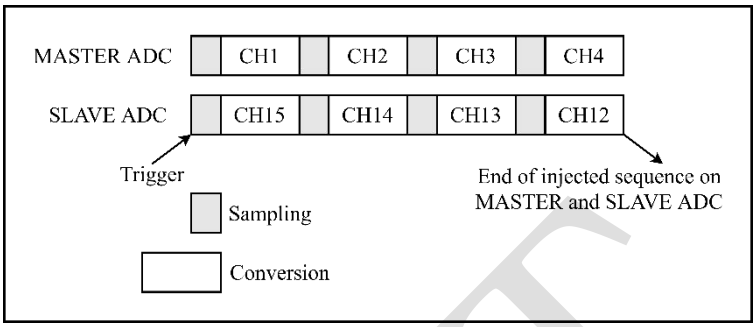


图 14-26 注入同步模式-Dual ADC 模式

支持独立注入的常规同步模式

通过将 ADCx_CCR.DUAL 配置为 0x5 来选择此模式。

此模式可用于同步转换常规序列，外部触发事件来自主 ADC 的常规事件，通过 ADCx_LR.EXTSEL 进行选择，触发事件同步用于从 ADC。

常规同步模式下支持独立注入转换，主 ADC 或从 ADC 上的注入请求将中止当前的常规转换，并在注入转换结束后重新开始常规转换。

注：不要在两个 ADC 上转换同一通道（转换同一通道时，不允许两个 ADC 采样时间重叠）。

注：在常规同步模式下，转换的序列长度必须相同，并且在序列中主 ADC 和从 ADC 的第 N 次转换必须配置为采用相同的采样时间。

软件能够读取数据时会通过中断的方式获得通知：

- 每次主 ADC 上出现转换结束 EOC 事件时，会产生主 EOC 中断标志，软件可读取主 ADC 的 ADCx_DR。
- 每次从 ADC 上出现转换结束 EOC 事件时，会产生从 EOC 中断标志，软件可读取从 ADC 的 ADCx_DR。
- 如果主常规序列的持续时间与从常规序列的持续时间相等，软件可以只关注两个 EOC 中断标志中的一个（例如主 EOC），并从通用数据寄存器 ADCx_CDR 中读取两者的已转换数据，如下图所示。

如果 ADCx_CFGR0.DISCEN=1，常规序列的每“n”个同步转换都需要在出现一次常规触发事件后才能进行（“n”由 ADCx_CFGR0.DISCNUM 定义）。

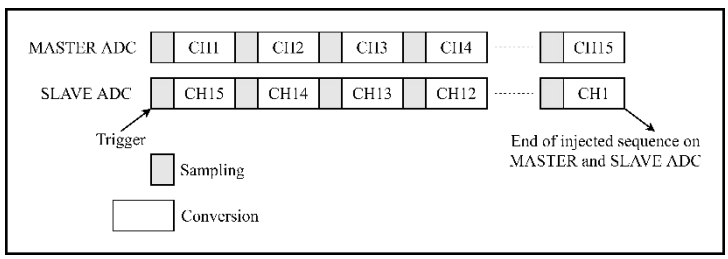


图 14-27 常规同步模式-Dual ADC 模式

支持独立注入的常规交错模式

通过将 ADCx_CCR.DUAL 配置为 0x6 来选择常规交错模式。

常规交错模式只能用于常规序列（通常为一个通道），外部触发事件来自主 ADC 的常规事件，通过 ADCx_LR.EXTSEL 进行选择，触发事件同步用于从 ADC。

出现外部触发事件之后：

- 主 ADC 立即启动。
- 从 ADC 在主 ADC 采样阶段开始后的多个 ADC 时钟周期延时后启动。

常规交错模式下 2 个转换之间的最小延迟通过 ADCx_CCR.DELAY 进行配置。该延时会在主转换的采样阶段开始后开始计时。这样一来需要保证延迟时间大于主 ADC 采样时间，如果某个 ADC 的互补 ADC 仍在对其输入进行采样，则该 ADC 无法启动转换（在给定时间内，只有一个 ADC 能够对输入信号采样）。

出现外部触发之后：

- 最小延时为采样时间+1 个时钟周期，用以确保主 ADC 采样阶段模拟开关断开与从 ADC 采样阶段模拟开关闭合之间至少有一个周期时间。
- 最大延时为 1024 个时钟周期，用户必须正确计算该延时，以确保在某个 ADC 仍在对其输入进行采样时不会有其他 ADC 开始转换。

每次从 ADC 上出现转换结束 EOC 事件时，都会在软件能够读取数据时以中断方式通知软件，会产生从 EOC 中断标志和主 EOC 中断标志，软件可读取从/主 ADC 的 ADCx_DR。

注：可以仅使能从 ADC 的 EOC 中断并读取通用数据寄存器 ADCx_CDR。

如果 ADCx_CFGR0.DISCEN=1，常规序列的每“n”（“n”由 ADCx_CFGR0.DISCNUM 定义）个同步转换需要在出现一个常规触发事件后才能进行。

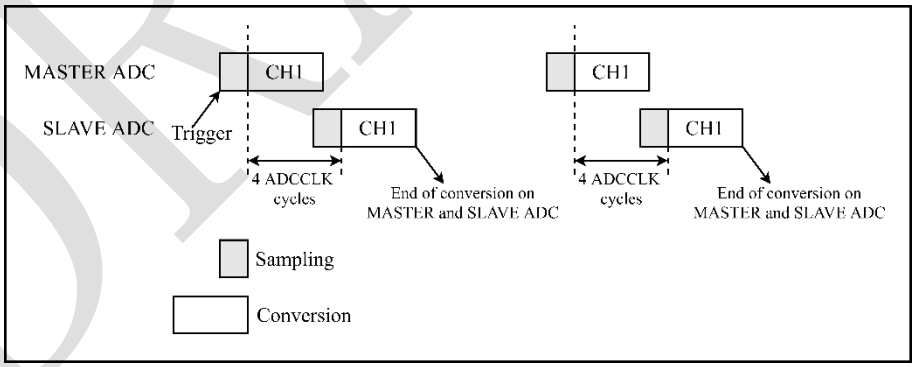


图 14-28 常规交错模式-Dual ADC 模式

交替触发模式

通过将 ADCx_CCR.DUAL 配置为 0x7 来选择交替触发模式。

交替触发模式只能用于注入序列，外部触发事件来自主 ADC 的注入事件。

不使能注入不连续模式（两个 ADC 的 ADCx_CFGR0.JDISCEN 均为 0）情况下：

- 发生第一次触发时，将转换组中主 ADC 的所有注入转换。
- 发生第二次触发时，将转换组中从 ADC 的所有注入转换。
- 以此类推。

软件能够读取数据时会通过中断的方式获得通知：

- 当注入序列中主 ADC 的所有通道都转换完成后，会生成一个 JEOS 中断标志。
- 当注入序列中从 ADC 的所有通道都转换完成后，会生成一个 JEOS 中断标志。
- 主 ADC 和从 ADC 每次注入转换后也会生成 JEOC 中断标志。

如果所有注入序列都完成转换后出现另一个外部触发事件，则可通过转换组中主 ADC 的注入转换来重新启动交替触发过程。

注入同步模式下，常规序列可在一个或所有 ADC 上执行，它们彼此之间都是独立的，而且会在出现注入事件时中断，常规序列会在注入序列结束时恢复转换。

两次触发事件之间的时间间隔必须大于或等于 1 个 ADC 时钟周期。同一个 ADC 上可启动转换的两个触发事件之间的最小时间间隔与单个 ADC 模式下相同。

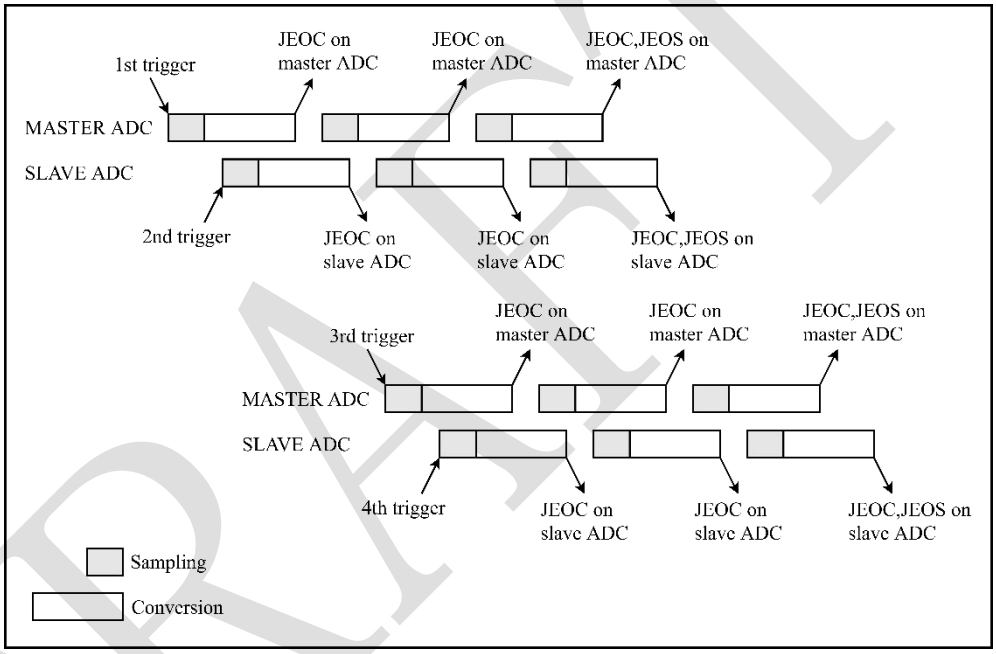


图 14-29 交替触发模式（触发注入）-Dual ADC 模式

使能注入不连续模式（两个 ADC 的 ADCx_CFGR0.JDISCEN 均为 1）情况下：

如果使能主 ADC 和从 ADC 的注入不连续模式：

- 发生第一次触发时，将转换主 ADC 的第一个注入转换。
- 发生第二次触发时，将转换从 ADC 的第一个注入转换。
- 以此类推。

软件能够读取数据时会通过中断的方式获得通知：

- 当组中主 ADC 的所有通道都转换完成后，会生成一个 JEOS 中断标志。
- 当组中从 ADC 的所有通道都转换完成后，会生成一个 JEOS 中断标志。
- 主 ADC 和从 ADC 每次注入转换后也会生成 JEOC 中断标志。

如果注入序列完成转换后出现另一个外部触发事件，则会重新启动交替触发过程。

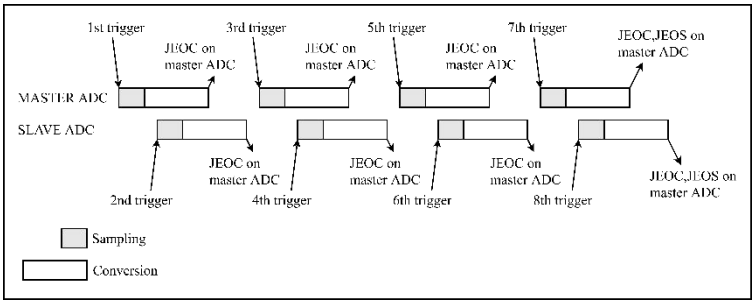


图 14-30 交替触发模式（不连续注入）-Dual ADC 模式

混合型常规/注入同步模式

通过将 ADCx_CCR.DUAL 编程为 0x1 来选择常规同步模式+注入同步模式。

可以中断常规序列的同步转换，然后开始注入序列的同步转换。

注：转换的序列长度必须相同，并且在给定序列中，主 ADC 和从 ADC 的第 N 次转换必须配置为采用相同的采样时间，或确保触发事件的间隔长于 2 个序列中较长的转换时间。如果不遵循上述条件，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。

常规同步+交替触发组合模式

通过将 ADCx_CCR.DUAL 编程为 0x2 来选择常规同步模式+交替触发模式。

可以中断常规序列的同步转换，然后开始注入序列的交替触发转换，交替触发模式中断同步常规转换的行为如图 142 所示。

注入事件后立即开始注入交替转换，当常规转换处于运行状态时，为确保在注入转换后实现同步，所有的（主/从）ADC 常规转换均将停止，并会在注入转换结束时得以恢复运行。

转换的序列长度必须相同，并且在给定序列中，主模式和从模式的第 N 次转换必须配置为采用相同的采样时间，或确保触发事件之间的间隔长于 2 个序列中较长的转换时间。如果不遵循上述条件，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。

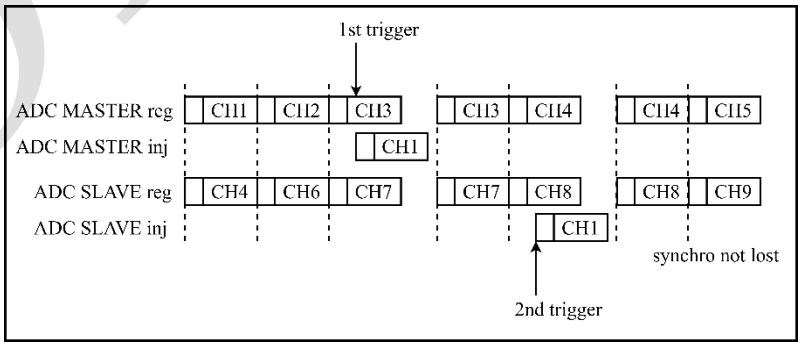


图 14-31 常规同步+交替触发模式-Dual ADC 模式

14.4.22 中断

- 每个 ADC 可以产生 10 个常规中断：
  - 常规组的任何采样结束时（ADCx_ISR.EOSMP 标志）
  - ADC 使能后准备就绪时（ADCx_ISR.ADRDY 标志）
  - 常规组的任何转换结束时（ADCx_ISR.EOC 标志）
  - 常规组的转换序列结束时（ADCx_ISR.EOS 标志）
  - 注入组的任何转换结束时（ADCx_ISR.JEOC 标志）
  - 注入组的转换序列结束时（ADCx_ISR.JEOS 标志）
  - 发生数据溢出时（ADCx_ISR.OVR 标志）
  - 发生模拟看门狗检测时（ADCx_ISR.AWD0，ADCx_ISR.AWD1 和 ADCx_ISR.AWD2 标志）
- 每个 ADC 可以产生 20 个采样中断：
  - 模拟通道 x 的采样转换完成时（ADCx_DISR.DONx 标志）
- 每个 ADC 可以产生 20 个半完成中断：
  - 模拟通道 x 的 DMA 半传输完成时（ADCx_HISR.HLFx 标志）
- 每个 ADC 可以产生 20 个完成中断：
  - 模拟通道 x 的 DMA 传输完成时（ADCx_FISR.FULx 标志）
- 每个中断都设置有单独的中断使能位，以实现最大的灵活性。

表 14-9 每个 ADC 的中断事件总结

中断向量	中断事件	中断标志	使能控制位	标志清除位
adcx_norm_int	常规转换结束事件	EOC	EOCIE	EOC
	常规序列结束事件	EOS	EOSIE	EOS
	注入转换结束事件	JEOC	JEOCIE	JEOC
	注入序列结束事件	JEOS	JEOSIE	JEOS
	数据溢出事件	OVR	OVRIE	OVR
	模拟看门狗 0 检测事件	AWD0	AWD0IE	AWD0
	模拟看门狗 1 检测事件	AWD1	AWD1IE	AWD1
	模拟看门狗 2 检测事件	AWD2	AWD2IE	AWD2
	ADC 准备就绪事件	ADRDY	ADRDYIE	ADRDY
	常规采样结束事件	EOSMP	EOSMPIE	EOSMP
adcx_samp_int	通道 0~19 采样完成事件	DONx	DONIEx	DONx
adcx_half_int	通道 0~19 DMA 半完成事件	HLFx	HLFIEx	HLFx
adcx_full_int	通道 0~19 DMA 完成事件	FULx	FULIEx	FULx

14.5 寄存器定义

14.5.1 寄存器列表

ADC 基地址：

ADC0(0x40038000) ADC1(0x40038400) ADC2(0x40038800) ADC3(0x40038C00)

偏移	实例地址	名称	默认值	描述
0x00	ADC 基地址+0x00	ADCx_CR	0x00000000	ADC 控制寄存器
0x04	ADC 基地址+0x04	ADCx_CFGR0	0x00000000	ADC 配置寄存器 0
0x08	ADC 基地址+0x08	ADCx_CFGR1	0x00000000	ADC 配置寄存器 1
0x10	ADC 基地址+0x10	ADCx_ISR	0x00000000	ADC 中断状态寄存器
0x14	ADC 基地址+0x14	ADCx_IER	0x00000000	ADC 中断使能寄存器
0x18	ADC 基地址+0x18	ADCx_SIGSEL	0x000FFFFF	ADC 转换模式选择寄存器
0x20	ADC 基地址+0x20	ADCx_SMPR0	0x00000000	ADC 采样时间寄存器 0
0x24	ADC 基地址+0x24	ADCx_SMPR1	0x00000000	ADC 采样时间寄存器 1
0x28	ADC 基地址 0x28	ADCx_SMPR2	0x00000000	ADC 采样时间寄存器 2
0x30	ADC 基地址+0x30	ADCx_CALR0	0x00000000	ADC 校准数据寄存器 0
0x34	ADC 基地址+0x34	ADCx_CALR1	0x00000000	ADC 校准数据寄存器 1
0x38	ADC 基地址+0x38	ADCx_CALR2	0x00000000	ADC 校准数据寄存器 2
0x40	ADC 基地址+0x40	ADCx_SQR0	0x00000000	ADC 常规序列寄存器 0
0x44	ADC 基地址+0x44	ADCx_SQR1	0x00000000	ADC 常规序列寄存器 1
0x48	ADC 基地址+0x48	ADCx_SQR2	0x00000000	ADC 常规序列寄存器 2
0x4C	ADC 基地址+0x4C	ADCx_SQR3	0x00000000	ADC 常规序列寄存器 3
0x50	ADC 基地址+0x50	ADCx_LR	0x00000000	ADC 常规序列长度寄存器
0x54	ADC 基地址+0x54	ADCx_DR	0x00000000	ADC 常规序列数据寄存器
0x60	ADC 基地址+0x60	ADCx_JSQR	0x00000000	ADC 注入序列寄存器
0x64	ADC 基地址+0x64	ADCx_JLR	0x00000000	ADC 注入序列长度寄存器
0x70	ADC 基地址+0x70	ADCx_JDR0	0x00000000	ADC 注入序列数据寄存器 0
0x74	ADC 基地址+0x74	ADCx_JDR1	0x00000000	ADC 注入序列数据寄存器 1
0x78	ADC 基地址+0x78	ADCx_JDR2	0x00000000	ADC 注入序列数据寄存器 2
0x7C	ADC 基地址+0x7C	ADCx_JDR3	0x00000000	ADC 注入序列数据寄存器 3
0x80	ADC 基地址+0x80	ADCx_TR0	0x00000000	ADC 看门狗 0 阈值寄存器
0x84	ADC 基地址+0x84	ADCx_TR1	0x00000000	ADC 看门狗 1 阈值寄存器
0x88	ADC 基地址 0x88	ADCx_TR2	0x00000000	ADC 看门狗 2 阈值寄存器
0x90	ADC 基地址+0x90	ADCx_AWD0CR	0x00000000	ADC 看门狗 0 控制寄存器
0x94	ADC 基地址+0x94	ADCx_AWD1CR	0x00000000	ADC 看门狗 1 控制寄存器
0x98	ADC 基地址+0x98	ADCx_AWD2CR	0x00000000	ADC 看门狗 2 控制寄存器
0xA0	ADC 基地址+0xA0	ADCx_OFR0	0x00000000	ADC 偏置补偿寄存器 0
0xA4	ADC 基地址+0xA4	ADCx_OFR1	0x00000000	ADC 偏置补偿寄存器 1
0xA8	ADC 基地址+0xA8	ADCx_OFR2	0x00000000	ADC 偏置补偿寄存器 2

0xAC	ADC 基地址+0xAC	ADCx_OFR3	0x00000000	ADC 偏置补偿寄存器 3
0xC0	ADC 基地址+0xC0	ADCx_GCR0	0x00002000	ADC 增益系数寄存器 0
0xC4	ADC 基地址+0xC4	ADCx_GCR1	0x00002000	ADC 增益系数寄存器 1
0xC8	ADC 基地址+0xC8	ADCx_GCR2	0x00002000	ADC 增益系数寄存器 2
0xCC	ADC 基地址+0xCC	ADCx_GCR3	0x00002000	ADC 增益系数寄存器 3
0xE0	ADC 基地址+0xE0	ADCx_GOFTR	0x00000000	ADC 偏置补偿系数寄存器
0xF0	ADC 基地址+0xF0	ADCx_GGCTR	0x00002000	ADC 增益补偿系数寄存器
0x100	ADC 基地址+0x100	ADCx_DISR	0x00000000	ADC 数据中断状态寄存器
0x104	ADC 基地址+0x104	ADCx_DIER	0x00000000	ADC 数据中断使能寄存器
0x110	ADC 基地址+0x110	ADCx_CDR0	0x00000000	ADC 通道 0 数据寄存器
0x114	ADC 基地址+0x114	ADCx_CDR1	0x00000000	ADC 通道 1 数据寄存器
0x118	ADC 基地址+0x118	ADCx_CDR2	0x00000000	ADC 通道 2 数据寄存器
0x11C	ADC 基地址+0x11C	ADCx_CDR3	0x00000000	ADC 通道 3 数据寄存器
0x120	ADC 基地址+0x120	ADCx_CDR4	0x00000000	ADC 通道 4 数据寄存器
0x124	ADC 基地址+0x124	ADCx_CDR5	0x00000000	ADC 通道 5 数据寄存器
0x128	ADC 基地址+0x128	ADCx_CDR6	0x00000000	ADC 通道 6 数据寄存器
0x12C	ADC 基地址+0x12C	ADCx_CDR7	0x00000000	ADC 通道 7 数据寄存器
0x130	ADC 基地址+0x130	ADCx_CDR8	0x00000000	ADC 通道 8 数据寄存器
0x134	ADC 基地址 0x134	ADCx_CDR9	0x00000000	ADC 通道 9 数据寄存器
0x138	ADC 基地址+0x138	ADCx_CDR10	0x00000000	ADC 通道 10 数据寄存器
0x13C	ADC 基地址+0x13C	ADCx_CDR11	0x00000000	ADC 通道 11 数据寄存器
0x140	ADC 基地址+0x140	ADCx_CDR12	0x00000000	ADC 通道 12 数据寄存器
0x144	ADC 基地址+0x144	ADCx_CDR13	0x00000000	ADC 通道 13 数据寄存器
0x148	ADC 基地址+0x148	ADCx_CDR14	0x00000000	ADC 通道 14 数据寄存器
0x14C	ADC 基地址+0x14C	ADCx_CDR15	0x00000000	ADC 通道 15 数据寄存器
0x150	ADC 基地址+0x150	ADCx_CDR16	0x00000000	ADC 通道 16 数据寄存器
0x154	ADC 基地址 0x154	ADCx_CDR17	0x00000000	ADC 通道 17 数据寄存器
0x158	ADC 基地址+0x158	ADCx_CDR18	0x00000000	ADC 通道 18 数据寄存器
0x15C	ADC 基地址+0x15C	ADCx_CDR19	0x00000000	ADC 通道 19 数据寄存器
0x180	ADC 基地址+0x180	ADCx_HISR	0x00000000	ADC 传输半完成中断状态寄存器
0x184	ADC 基地址+0x184	ADCx_HIER	0x00000000	ADC 传输半完成中断使能寄存器
0x188	ADC 基地址+0x188	ADCx_FISR	0x00000000	ADC 传输完成中断状态寄存器
0x18C	ADC 基地址+0x18C	ADCx_FIER	0x00000000	ADC 传输完成中断使能寄存器
0x190	ADC 基地址+0x190	ADCx_TCR0	0x00000000	ADC 传输控制寄存器 0
0x194	ADC 基地址+0x194	ADCx_TAR0	0x20010000	ADC 传输地址寄存器 0
0x198	ADC 基地址+0x198	ADCx_TLR0	0x00000000	ADC 传输长度寄存器 0
0x1A0	ADC 基地址+0x1A0	ADCx_TCR1	0x00000000	ADC 传输控制寄存器 1
0x1A4	ADC 基地址+0x1A4	ADCx_TAR1	0x00000000	ADC 传输控制寄存器 1
0x1A8	ADC 基地址+0x1A8	ADCx_TLR1	0x00000000	ADC 传输长度寄存器 1
0x1B0	ADC 基地址+0x1B0	ADCx_TCR2	0x00000000	ADC 传输控制寄存器 2
0x1B4	ADC 基地址+0x1B4	ADCx_TAR2	0x20010000	ADC 传输地址寄存器 2
0x1B8	ADC 基地址+0x1B8	ADCx_TLR2	0x00000000	ADC 传输长度寄存器 2
0x1C0	ADC 基地址+0x1C0	ADCx_TCR3	0x00000000	ADC 传输控制寄存器 3

0x1C4	ADC 基地址+0x1C4	ADCx_TAR3	0x20010000	ADC 传输地址寄存器 3
0x1C8	ADC 基地址+0x1C8	ADCx_TLR3	0x00000000	ADC 传输长度寄存器 3
0x1D0	ADC 基地址+0x1D0	ADCx_TCR4	0x00000000	ADC 传输控制寄存器 4
0x1D4	ADC 基地址+0x1D4	ADCx_TAR4	0x20010000	ADC 传输地址寄存器 4
0x1D8	ADC 基地址+0x1D8	ADCx_TLR4	0x00000000	ADC 传输长度寄存器 4
0x1E0	ADC 基地址+0x1E0	ADCx_TCR5	0x00000000	ADC 传输控制寄存器 5
0x1E4	ADC 基地址+0x1E4	ADCx_TAR5	0x20010000	ADC 传输地址寄存器 5
0x1E8	ADC 基地址+0x1E8	ADCx_TLR5	0x00000000	ADC 传输长度寄存器 5
0x1F0	ADC 基地址+0x1F0	ADCx_TCR6	0x00000000	ADC 传输控制寄存器 6
0x1F4	ADC 基地址+0x1F4	ADCx_TAR6	0x20010000	ADC 传输地址寄存器 6
0x1F8	ADC 基地址+0x1F8	ADCx_TLR6	0x00000000	ADC 传输长度寄存器 6
0x200	ADC 基地址+0x200	ADCx_TCR7	0x00000000	ADC 传输控制寄存器 7
0x204	ADC 基地址+0x204	ADCx_TAR7	0x20010000	ADC 传输地址寄存器 7
0x208	ADC 基地址+0x208	ADCx_TLR7	0x00000000	ADC 传输长度寄存器 7
0x210	ADC 基地址+0x210	ADCx_TCR8	0x00000000	ADC 传输控制寄存器 8
0x214	ADC 基地址+0x214	ADCx_TAR8	0x20010000	ADC 传输地址寄存器 8
0x218	ADC 基地址+0x218	ADCx_TLR8	0x00000000	ADC 传输长度寄存器 8
0x220	ADC 基地址+0x220	ADCx_TCR9	0x00000000	ADC 传输控制寄存器 9
0x224	ADC 基地址+0x224	ADCx_TAR9	0x20010000	ADC 传输地址寄存器 9
0x228	ADC 基地址+0x228	ADCx_TLR9	0x00000000	ADC 传输长度寄存器 9
0x230	ADC 基地址+0x230	ADCx_TCR10	0x00000000	ADC 传输控制寄存器 10
0x234	ADC 基地址+0x234	ADCx_TAR10	0x20010000	ADC 传输地址寄存器 10
0x238	ADC 基地址+0x238	ADCx_TLR10	0x00000000	ADC 传输长度寄存器 10
0x240	ADC 基地址+0x240	ADCx_TCR11	0x00000000	ADC 传输控制寄存器 11
0x244	ADC 基地址+0x244	ADCx_TAR11	0x20010000	ADC 传输地址寄存器 11
0x248	ADC 基地址+0x248	ADCx_TLR11	0x00000000	ADC 传输长度寄存器 11
0x250	ADC 基地址+0x250	ADCx_TCR12	0x00000000	ADC 传输控制寄存器 12
0x254	ADC 基地址+0x254	ADCx_TAR12	0x20010000	ADC 传输地址寄存器 12
0x258	ADC 基地址+0x258	ADCx_TLR12	0x00000000	ADC 传输长度寄存器 12
0x260	ADC 基地址+0x260	ADCx_TCR13	0x00000000	ADC 传输控制寄存器 13
0x264	ADC 基地址+0x264	ADCx_TAR13	0x20010000	ADC 传输地址寄存器 13
0x268	ADC 基地址+0x268	ADCx_TLR13	0x00000000	ADC 传输长度寄存器 13
0x270	ADC 基地址+0x270	ADCx_TCR14	0x00000000	ADC 传输控制寄存器 14
0x274	ADC 基地址+0x274	ADCx_TAR14	0x20010000	ADC 传输地址寄存器 14
0x278	ADC 基地址+0x278	ADCx_TLR14	0x00000000	ADC 传输长度寄存器 14
0x280	ADC 基地址+0x280	ADCx_TCR15	0x00000000	ADC 传输控制寄存器 15
0x284	ADC 基地址+0x284	ADCx_TAR15	0x20010000	ADC 传输地址寄存器 15
0x288	ADC 基地址+0x288	ADCx_TLR15	0x00000000	ADC 传输长度寄存器 15
0x290	ADC 基地址+0x290	ADCx_TCR16	0x00000000	ADC 传输控制寄存器 16
0x294	ADC 基地址+0x294	ADCx_TAR16	0x20010000	ADC 传输地址寄存器 16
0x298	ADC 基地址+0x298	ADCx_TLR16	0x00000000	ADC 传输长度寄存器 16
0x2A0	ADC 基地址+0x2A0	ADCx_TCR17	0x00000000	ADC 传输控制寄存器 17
0x2A4	ADC 基地址+0x2A4	ADCx_TAR17	0x20010000	ADC 传输地址寄存器 17

0x2A8	ADC 基地址+0x2A8	ADCx_TLR17	0x00000000	ADC 传输长度寄存器 17
0x2B0	ADC 基地址+0x2B0	ADCx_TCR18	0x00000000	ADC 传输控制寄存器 18
0x2B4	ADC 基地址+0x2B4	ADCx_TAR18	0x20010000	ADC 传输地址寄存器 18
0x2B8	ADC 基地址+0x2B8	ADCx_TLR18	0x00000000	ADC 传输长度寄存器 18
0x2C0	ADC 基地址+0x2C0	ADCx_TCR19	0x00000000	ADC 传输控制寄存器 19
0x2C4	ADC 基地址+0x2C4	ADCx_TAR19	0x20010000	ADC 传输地址寄存器 19
0x2C8	ADC 基地址+0x2C8	ADCx_TLR19	0x00000000	ADC 传输长度寄存器 19
0x300	ADC 基地址+0x300	ADCx_CCR	0x00000000	ADC 通用控制寄存器
0x304	ADC 基地址+0x304	ADCx_CSR	0x00000000	ADC 通用状态寄存器
0x308	ADC 基地址+0x308	ADCx_CDR	0x00000000	ADC 共用常规序列数据寄存器

14.5.2 寄存器描述

14.5.2.1 ADC 控制寄存器（ADCx_CR）

- 名称：ADC Control Register
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
										R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] JADSTP	**ADC 停止注入转换命令** 该位通过软件置位，通过硬件清零 1：ADC 注入模式停止中 0：ADC 注入模式已停止
[4] ADSTP	**ADC 停止常规转换命令** 该位通过软件置位，该位通过硬件清零 1：ADC 常规模式停止中 0：ADC 常规模式已停止
[3] JADSTART	**ADC 开始注入转换命令** 该位通过软件置位，该位通过硬件清零 1：ADC 注入模式运行中 0：ADC 注入模式不运行
[2] ADSTART	**ADC 开始常规转换命令** 该位通过软件置位，该位通过硬件清零 1：ADC 常规模式运行中 0：ADC 常规模式不运行
[1] ADDIS	**ADC 停止模块使能命令** 该位通过软件置位，该位通过硬件清零 1：ADC 模块使能停止中 0：ADC 模块使能已停止
[0] ADEN	**ADC 开始模块使能命令** 该位通过软件置位，该位通过硬件清零 1：ADC 模块使能运行中 0：ADC 模块使能不运行

14.5.2.2 ADC 配置寄存器 0（ADCx_CFGR0）

- 名称：ADC Configuration Register0
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CONT	JAUTO		JDISCEN	DISCNUM			DISCEN
								R/W	R/W		R/W	R/W			R/W
								0x0	0x0		0x0	0x0			0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVRMOD			TROVS	OVSS				ROVSM	OVSR					JOVSE	ROVSE
R/W			R/W	R/W				R/W	R/W					R/W	R/W
0x0			0x0	0x0				0x0	0x0					0x0	0x0

字段	说明
[23] CONT	**常规序列的单次 / 连续转换模式** 1：连续转换模式 0：单次转换模式
[22] JAUTO	**注入序列自动转换** 1：自动注入使能 0：自动注入不使能
[20] JDISCEN	**注入序列的不连续采样模式** 1：非连续转换模式 0：单次转换模式
[19:17] DISCNUM	**不连续采样模式通道计数** 7：8 次转换 6：7 次转换 5：6 次转换 4：5 次转换 3：4 次转换 2：3 次转换 1：2 次转换 0：1 次转换
[16] DISCEN	**常规序列的单次 / 不连续转换模式** 1：不连续转换模式 0：单次转换模式
[15] OVRMOD	**数据溢出模式** 1：数据覆盖模式 0：数据保留模式
[12] TROVS	**触发式规则序列过采样使能** 1：使能 0：不使能
[11:8] OVSS	**过采样数据移位位数** 9-15：Reserved 8：右移 8 位 7：右移 7 位 6：右移 6 位 5：右移 5 位 4：右移 4 位 3：右移 3 位 2：右移 2 位 1：右移 1 位 0：不右移
[7] ROVSM	**常规通道的过采样模式** 1：采样恢复模式

	0：采样继续模式
[6:4] OVSR	**过采样数据基础比率** 7：256 倍 6：128 倍 5：64 倍 4：32 倍 3：16 倍 2：8 倍 1：4 倍 0：2 倍
[1] JOVSE	**注入序列过采样使能** 1：使能 0：不使能
[0] ROVSE	**规则序列过采样使能** 1：使能 0：不使能

14.5.2.3 ADC 配置寄存器 1（ADCx_CFGR1）

- **名称：ADC Configuration Register1**
- **偏移地址：0x08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	JAWD 2EN	JAWD 1EN	JAWD 0EN		AWD2 EN	AWD1 EN	AWD0 EN			ISEL		CHEN		REFE N	BIASE N
	R/W	R/W	R/W		R/W	R/W	R/W			R/W		R/W		R/W	R/W
	0x0	0x0	0x0		0x0	0x0	0x0			0x0		0x0		0x0	0x0

字段	说明
[14] JAWD2EN	**模拟看门狗 2 监测注入序列** 1：使能 0：不使能
[13] JAWD1EN	**模拟看门狗 1 监测注入序列** 1：使能 0：不使能
[12] JAWD0EN	**模拟看门狗 0 监测注入序列** 1：使能 0：不使能
[10] AWD2EN	**模拟看门狗 2 监测规则序列** 1：使能 0：不使能
[9] AWD1EN	**模拟看门狗 1 监测规则序列** 1：使能 0：不使能
[8] AWD0EN	**模拟看门狗 0 监测规则序列** 1：使能 0：不使能
[5:4] ISEL	**偏置电流档位** 3：8 uA 2：14 uA 1：12 uA

	0：10 uA
[3] CHEN	**通道使能** 1：使能 0：不使能 Note：使能 CHEN 之后才可以使能 ADEN，否则转换数据可能出错
[1] REFEN	**Reference 模块使能** 1：使能 0：不使能 Note：使能 REFEN 之后才可以使能 ADEN，否则转换数据可能出错
[0] BIASEN	**内部偏置模块使能** 1：使能 0：不使能 Note：使能 BIASEN 之后才可以使能 ADEN，否则转换数据可能出错

14.5.2.4 ADC 中断状态寄存器（ADCx_ISR）

- **名称：ADC Interrupt Status Register**
- **偏移地址：0x10**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						EOSMP	ADRDY	AWD2	AWD1	AWD0	OVR	JEOS	JEOC	EOS	EOC
						R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
						0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[9] EOSMP	**采样阶段结束中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[8] ADRDY	**ADC 就绪中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[7] AWD2	**模拟看门狗 2 中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[6] AWD1	**模拟看门狗 1 中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[5] AWD0	**模拟看门狗 0 中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[4] OVR	**ADC 数据溢出中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[3]	**注入序列结束中断标志**

JEOS	1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[2] JEOC	**注入转换结束中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[1] EOS	**常规序列结束中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[0] EOC	**常规转换结束中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0

14.5.2.5 ADC 中断使能寄存器（ADCx_IER）

- **名称：ADC Interrupt Enable Register**
- **偏移地址：0x14**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						EOSMPIE	ADRDYIE	AWD2IE	AWD1IE	AWD0IE	OVRIE	JEOSIE	JEOCIE	EOSIE	EOCIE
						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
						0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[9] EOSMPIE	**采样阶段结束中断使能** 1：中断使能 0：中断不使能
[8] ADRDYIE	**ADC 就绪中断使能** 1：中断使能 0：中断不使能
[7] AWD2IE	**模拟看门狗 2 中断使能** 1：中断使能 0：中断不使能
[6] AWD1IE	**模拟看门狗 1 中断使能** 1：中断使能 0：中断不使能
[5] AWD0IE	**模拟看门狗 0 中断使能** 1：中断使能 0：中断不使能
[4] OVRIE	**ADC 数据溢出中断使能** 1：中断使能 0：中断不使能
[3] JEOSIE	**注入序列结束中断使能** 1：中断使能 0：中断不使能
[2] JEOCIE	**注入转换结束中断使能** 1：中断使能

	0：中断不使能
[1] EOSIE	**常规序列结束中断使能** 1：中断使能 0：中断不使能
[0] EOCIE	**常规转换结束中断使能** 1：中断使能 0：中断不使能

14.5.2.6 ADC 转换模式选择寄存器（ADCx_SIGSEL）

- **名称：ADC Single-End Select Register**
- **偏移地址：0x18**
- **默认值：0x000FFFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												SIGSEL			
												R/W			
												0xfffff			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGSEL															
R/W															
0xfffff															

字段	说明
[19:0] SIGSEL	**ADC 通道转换模式选择** SIGSEL[i] = 0x1：第 i 路通道为差分模式 SIGSEL[i] = 0x0：第 i 路通道为单端模式

14.5.2.7 ADC 采样时间寄存器 0（ADCx_SMPR0）

- **名称：ADC Sample Time Register 0**
- **偏移地址：0x20**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SMP7				SMP6				SMP5				SMP4		
	R/W				R/W				R/W				R/W		
	0x0				0x0				0x0				0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SMP3				SMP2				SMP1				SMP0		
	R/W				R/W				R/W				R/W		
	0x0				0x0				0x0				0x0		

字段	说明
[30:28] SMP7	**通道 7 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期

[26:24] SMP6	**通道 6 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[22:20] SMP5	**通道 5 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[18:16] SMP4	**通道 4 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[14:12] SMP3	**通道 3 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[10:8] SMP2	**通道 2 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[6:4] SMP1	**通道 1 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[2:0] SMP0	**通道 0 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期

4：62 个 ADC 时钟周期
5：126 个 ADC 时钟周期
6：254 个 ADC 时钟周期
7：510 个 ADC 时钟周期

14.5.2.8 ADC 采样时间寄存器 1（ADCx_SMPR1）

- 名称：**ADC Sample Time Register 1**
- 偏移地址：**0x24**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SMP15				SMP14				SMP13				SMP12		
	R/W				R/W				R/W				R/W		
	0x0				0x0				0x0				0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SMP11				SMP10				SMP9				SMP8		
	R/W				R/W				R/W				R/W		
	0x0				0x0				0x0				0x0		

字段	说明
[30:28] SMP15	**通道 15 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[26:24] SMP14	**通道 14 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[22:20] SMP13	**通道 13 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[18:16] SMP12	**通道 12 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期

	7：510 个 ADC 时钟周期
[14:12] SMP11	**通道 11 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[10:8] SMP10	**通道 10 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[6:4] SMP9	**通道 9 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[2:0] SMP8	**通道 8 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期

14.5.2.9 ADC 采样时间寄存器 2（ADCx_SMPR2）

- **名称：ADC Sample Time Register 2**
- **偏移地址：0x28**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SMP19				SMP18				SMP17				SMP16		
	R/W				R/W				R/W				R/W		
	0x0				0x0				0x0				0x0		

字段	说明
[14:12] SMP19	**通道 19 采样时间选择位** 0：2 个 ADC 时钟周期

	1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[10:8] SMP18	**通道 18 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[6:4] SMP17	**通道 17 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期
[2:0] SMP16	**通道 16 采样时间选择位** 0：2 个 ADC 时钟周期 1：6 个 ADC 时钟周期 2：14 个 ADC 时钟周期 3：30 个 ADC 时钟周期 4：62 个 ADC 时钟周期 5：126 个 ADC 时钟周期 6：254 个 ADC 时钟周期 7：510 个 ADC 时钟周期

14.5.2.10 ADC 校准数据寄存器 0（ADCx_CALR0）

- **名称：ADC Calibration Data Register 0**
- **偏移地址：0x30**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAT7		CAL7		SAT6		CAL6		SAT5		CAL5		SAT4		CAL4	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAT3		CAL3		SAT2		CAL2		SAT1		CAL1		SAT0		CAL0	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	

字段	说明
[31] SAT7	**通道 7 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[29:28] CAL7	**通道 7 校准系数选择位** 0：OFFSET0 / GAIN COEFF0

	1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[27] SAT6	**通道 6 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[25:24] CAL6	**通道 6 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[23] SAT5	**通道 5 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[21:20] CAL5	**通道 5 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[19] SAT4	**通道 4 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[17:16] CAL4	**通道 4 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[15] SAT3	**通道 3 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[13:12] CAL3	**通道 3 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[11] SAT2	**通道 2 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[9:8] CAL2	**通道 2 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[7] SAT1	**通道 1 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[5:4] CAL1	**通道 1 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[3] SAT0	**通道 0 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[1:0] CAL0	**通道 0 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3

14.5.2.11 ADC 校准数据寄存器 1（ADCx_CALR1）

- 名称：ADC Calibration Data Register 1
- 偏移地址：0x34
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAT15		CAL15		SAT14		CAL14		SAT13		CAL13		SAT12		CAL12	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAT11		CAL11		SAT10		CAL10		SAT9		CAL9		SAT8		CAL8	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	

字段	说明
[31] SAT15	**通道 15 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[29:28] CAL15	**通道 15 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[27] SAT14	**通道 14 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[25:24] CAL14	**通道 14 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[23] SAT13	**通道 13 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[21:20] CAL13	**通道 13 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[19] SAT12	**通道 12 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[17:16] CAL12	**通道 12 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[15] SAT11	**通道 11 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[13:12] CAL11	**通道 11 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2

	3：OFFSET3 / GAIN COEFF3
[11] SAT10	**通道 10 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[9:8] CAL10	**通道 10 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[7] SAT9	**通道 9 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[5:4] CAL9	**通道 9 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[3] SAT8	**通道 8 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[1:0] CAL8	**通道 8 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3

14.5.2.12 ADC 校准数据寄存器 2（ADCx_CALR2）

- **名称：ADC Calibration Data Register 2**
- **偏移地址：0x38**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAT19		CAL19		SAT18		CAL18		SAT17		CAL17		SAT16		CAL16	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	

字段	说明
[15] SAT19	**通道 19 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[13:12] CAL19	**通道 19 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[11] SAT18	**通道 18 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[9:8] CAL18	**通道 18 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1

	2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[7] SAT17	**通道 17 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[5:4] CAL17	**通道 17 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3
[3] SAT16	**通道 16 饱和运算选择位** 1：不支持饱和运算 0：支持饱和运算
[1:0] CAL16	**通道 16 校准系数选择位** 0：OFFSET0 / GAIN COEFF0 1：OFFSET1 / GAIN COEFF1 2：OFFSET2 / GAIN COEFF2 3：OFFSET3 / GAIN COEFF3

14.5.2.13 ADC 常规序列寄存器 0（ADCx_SQR0）

- **名称：ADC Regular Sequence Register0**
- **偏移地址：0x40**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SQ5						SQ4						SQ3
			R/W						R/W						R/W
			0x0						0x0						0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ3					SQ2						SQ1				
R/W					R/W						R/W				
0x0					0x0						0x0				

字段	说明
[28:24] SQ5	**常规序列中的第 5 次转换**
[22:18] SQ4	**常规序列中的第 4 次转换**
[16:12] SQ3	**常规序列中的第 3 次转换**
[10:6] SQ2	**常规序列中的第 2 次转换**
[4:0] SQ1	**常规序列中的第 1 次转换**

14.5.2.14 ADC 常规序列寄存器 1（ADCx_SQR1）

- 名称：ADC Regular Sequence Register1
- 偏移地址：0x44
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SQ10						SQ9						SQ8
			R/W						R/W						R/W
			0x0						0x0						0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ8					SQ7						SQ6				
R/W					R/W						R/W				
0x0					0x0						0x0				

字段	说明
[28:24] SQ10	常规序列中的第 10 次转换
[22:18] SQ9	常规序列中的第 9 次转换
[16:12] SQ8	常规序列中的第 8 次转换
[10:6] SQ7	常规序列中的第 7 次转换
[4:0] SQ6	常规序列中的第 6 次转换

14.5.2.15 ADC 常规序列寄存器 2（ADCx_SQR2）

- 名称：ADC Regular Sequence Register2
- 偏移地址：0x48
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SQ15						SQ14						SQ13
			R/W						R/W						R/W
			0x0						0x0						0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ13					SQ12						SQ11				
R/W					R/W						R/W				
0x0					0x0						0x0				

字段	说明
[28:24] SQ15	常规序列中的第 15 次转换
[22:18] SQ14	常规序列中的第 14 次转换
[16:12] SQ13	常规序列中的第 13 次转换
[10:6] SQ12	常规序列中的第 12 次转换
[4:0] SQ11	常规序列中的第 11 次转换

14.5.2.16 ADC 常规序列寄存器 3（ADCx_SQR3）

- 名称：ADC Regular Sequence Register3
- 偏移地址：0x4C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											SQ16				
											R/W				
											0x0				

字段	说明
[4:0] SQ16	常规序列中的第 16 次转换

14.5.2.17 ADC 常规序列长度寄存器（ADCx_LR）

- 名称：ADC Regular Length Register
- 偏移地址：0x50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				LEN				EXTEN			EXTSEL				
				R/W				R/W			R/W				
				0x0				0x0			0x0				

字段	说明
[11:8] LEN	常规序列的长度 0xf：16 次转换 0xe：15 次转换 0xd：14 次转换 0xc：13 次转换 0xb：12 次转换 0xa：11 次转换 0x9：10 次转换 0x8：9 次转换 0x7：8 次转换 0x6：7 次转换 0x5：6 次转换 0x4：5 次转换 0x3：4 次转换 0x2：3 次转换 0x1：2 次转换 0x0：1 次转换
[7:6] EXTEN	常规序列的硬件触发使能和极性选择 0x3：硬件触发上升 / 下降沿有效 0x2：硬件触发下降沿有效

	0x1：硬件触发上升沿有效 0x0：硬件触发不使能，软件触发使能
[4:0] EXTSEL	**常规序列的硬件触发选择** 0x1f：PA8 0x1e：HRPWM_ADCTRG9 0x1d：HRPWM_ADCTRG8 0x1c：HRPWM_ADCTRG7 0x1b：HRPWM_ADCTRG6 0x1a：HRPWM_ADCTRG5 0x19：HRPWM_ADCTRG4 0x18：HRPWM_ADCTRG3 0x17：HRPWM_ADCTRG2 0x16：HRPWM_ADCTRG1 0x15：HRPWM_ADCTRG0 0x14：TMR10_TRGO 0x13：TMR10_CC2 0x12：TMR10_CC1 0x11：TMR10_CC0 0x10：TMR9_TRGO 0xf：TMR9_CC1 0xe：TMR9_CC0 0xd：TMR4_TRGO 0xc：TMR4_CC1 0xb：TMR4_CC0 0xa：TMR3_TRGO 0x9：TMR3_CC1 0x8：TMR3_CC0 0x7：TMR2_TRGO 0x6：TMR1_TRGO 0x5：TMR0_TRGO 0x4：TMR2_CC0 0x3：TMR1_CC0 0x2：TMR0_CC0 0x1：TMR8_TRGO 0x0：TMR7_TRGO

14.5.2.18 ADC 常规序列数据寄存器（ADCx_DR）

- **名称：ADC Regular Data Register**
- **偏移地址：0x54**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA															
R/W															
0x0															

字段	说明
[15:0] RDATA	**常规序列已转换的数据**

14.5.2.19 ADC 注入序列寄存器（ADCx_JSQR）

- 名称：ADC Injected Sequence Register
- 偏移地址：0x60
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									JSQ4						JSQ3
									R/W						R/W
									0x0						0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ3					JSQ2						JSQ1				
R/W					R/W						R/W				
0x0					0x0						0x0				

字段	说明
[22:18] JSQ4	**注入序列中的第 4 次转换**
[16:12] JSQ3	**注入序列中的第 3 次转换**
[10:6] JSQ2	**注入序列中的第 2 次转换**
[4:0] JSQ1	**注入序列中的第 1 次转换**

14.5.2.20 ADC 注入序列长度寄存器（ADCx_JLR）

- 名称：ADC Injected Length Register
- 偏移地址：0x64
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						JLEN		JEXTEN			JEXTSEL				
						R/W		R/W			R/W				
						0x0		0x0			0x0				

字段	说明
[9:8] JLEN	**注入序列的长度** 0x3：4 次转换 0x2：3 次转换 0x1：2 次转换 0x0：1 次转换
[7:6] JEXTEN	**注入序列的硬件触发使能和极性选择** 0x3：硬件触发上升 / 下降沿有效 0x2：硬件触发下降沿有效 0x1：硬件触发上升沿有效 0x0：硬件触发不使能，软件触发使能
[4:0] JEXTSEL	**注入序列的硬件触发选择** 0x1f：PA9 0x1e：HRPWM_ADCTRG9 0x1d：HRPWM_ADCTRG8

0x1c：HRPWM_ADCTRG7
0x1b：HRPWM_ADCTRG6
0x1a：HRPWM_ADCTRG5
0x19：HRPWM_ADCTRG4
0x18：HRPWM_ADCTRG3
0x17：HRPWM_ADCTRG2
0x16：HRPWM_ADCTRG1
0x15：HRPWM_ADCTRG0
0x14：TMR10_TRGO
0x13：TMR10_CC2
0x12：TMR10_CC1
0x11：TMR10_CC0
0x10：TMR9_TRGO
0xf：TMR9_CC1
0xe：TMR9_CC0
0xd：TMR4_TRGO
0xc：TMR4_CC1
0xb：TMR4_CC0
0xa：TMR3_TRGO
0x9：TMR3_CC1
0x8：TMR3_CC0
0x7：TMR2_TRGO
0x6：TMR1_TRGO
0x5：TMR0_TRGO
0x4：TMR2_CC0
0x3：TMR1_CC0
0x2：TMR0_CC0
0x1：TMR8_TRGO
0x0：TMR7_TRGO

14.5.2.21 ADC 注入序列数据寄存器 0（ADCx_JDR0）

- **名称：ADC Injected Data Register0**
- **偏移地址：0x70**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															
R/W															
0x0															

字段	说明
[15:0] JDATA	**注入序列的第 0 个已转换数据**

14.5.2.22 ADC 注入序列数据寄存器 1（ADCx_JDR1）

- 名称：ADC Injected Data Register1
- 偏移地址：0x74
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															
R/W															
0x0															

字段	说明
[15:0] JDATA	注入序列的第 1 个已转换数据

14.5.2.23 ADC 注入序列数据寄存器 2（ADCx_JDR2）

- 名称：ADC Injected Data Register2
- 偏移地址：0x78
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															
R/W															
0x0															

字段	说明
[15:0] JDATA	注入序列的第 2 个已转换数据

14.5.2.24 ADC 注入序列数据寄存器 3（ADCx_JDR3）

- 名称：ADC Injected Data Register3
- 偏移地址：0x7C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															
R/W															
0x0															

字段	说明
[15:0] JDATA	**注入序列的第 3 个已转换数据**

14.5.2.25 ADC 看门狗 0 阈值寄存器（ADCx_TR0）

- 名称：ADC Watchdog0 Threshold Register
- 偏移地址：0x80
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT0															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT0															
R/W															
0x0															

字段	说明
[31:16] HT0	**模拟看门狗 0 阈值上限** 模拟看门狗 0 监测的通道转换值高于 HT0 时产生事件 / 中断 格式为 16 位有符号数，范围[-32768, 32767]
[15:0] LT0	**模拟看门狗 0 阈值下限** 模拟看门狗 0 监测的通道转换值低于 LT0 时产生事件 / 中断 格式为 16 位有符号数，范围[-32768, 32767]

14.5.2.26 ADC 看门狗 1 阈值寄存器（ADCx_TR1）

- 名称：ADC Watchdog1 Threshold Register
- 偏移地址：0x84
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT1															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT1															
R/W															
0x0															

字段	说明
[31:16] HT1	**模拟看门狗 1 阈值上限** 模拟看门狗 1 监测的通道转换值高于 HT1 时产生事件 / 中断 格式为 16 位有符号数，范围[-32768, 32767]
[15:0] LT1	**模拟看门狗 1 阈值下限** 模拟看门狗 1 监测的通道转换值低于 LT1 时产生事件 / 中断 格式为 16 位有符号数，范围[-32768, 32767]

14.5.2.27 ADC 看门狗 2 阈值寄存器（ADCx_TR2）

- 名称：ADC Watchdog2 Threshold Register
- 偏移地址：0x88
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT2															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT2															
R/W															
0x0															

字段	说明
[31:16] HT2	**模拟看门狗 2 阈值上限** 模拟看门狗 2 监测的通道转换值高于 HT2 时产生事件 / 中断 格式为 16 位有符号数，范围[-32768, 32767]
[15:0] LT2	**模拟看门狗 2 阈值下限** 模拟看门狗 2 监测 Channel 转换值低于 LT2 时产生事件 / 中断 格式为 16 位有符号数，范围[-32768, 32767]

14.5.2.28 ADC 看门狗 0 控制寄存器（ADCx_AWD0CR）

- 名称：ADC Watchdog0 Control Register
- 偏移地址：0x90
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				AWDFILT								AW0CH			
				R/W								R/W			
				0x0								0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AW0CH															
R/W															
0x0															

字段	说明
[27:24] AWDFILT	**模拟看门狗 0 滤波点数** 0xf：模拟看门狗 0 监测 16 次超过阈值的转换产生事件 / 中断 …… 0x1：模拟看门狗 0 监测 2 次超过阈值的转换产生事件 / 中断 0x0：模拟看门狗 0 不做滤波
[19:0] AW0CH	**模拟看门狗 0 通道选择** AW0CH[x] =1：ADC 通道 x 受模拟看门狗 0 监测 AW0CH[x] =0：ADC 通道 x 不受模拟看门狗 0 监测 AW0CH=0x0，模拟看门狗 0 等同于被关闭

14.5.2.29 ADC 看门狗 1 控制寄存器（ADCx_AWD1CR）

- 名称：ADC Watchdog1 Control Register
- 偏移地址：0x94
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				AWDFILT								AW1CH			
				R/W								R/W			
				0x0								0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AW1CH															
R/W															
0x0															

字段	说明
[27:24] AWDFILT	**模拟看门狗 1 滤波点数** 0xf：模拟看门狗 1 监测 16 次超过阈值的转换产生事件 / 中断 …… 0x1：模拟看门狗 1 监测 2 次超过阈值的转换产生事件 / 中断 0x0：模拟看门狗 1 不做滤波
[19:0] AW1CH	**模拟看门狗 1 通道选择** AW1CH[x] =1：ADC 通道 x 受模拟看门狗 1 监测 AW1CH[x] =0：ADC 通道 x 不受模拟看门狗 1 监测 AW1CH=0x0，模拟看门狗 1 等同于被关闭

14.5.2.30 ADC 看门狗 2 控制寄存器（ADCx_AWD2CR）

- 名称：ADC Watchdog 2 Control Register
- 偏移地址：0x98
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				AWDFILT								AW2CH			
				R/W								R/W			
				0x0								0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AW2CH															
R/W															
0x0															

字段	说明
[27:24] AWDFILT	**模拟看门狗 2 滤波点数** 0xf：模拟看门狗 2 监测 16 次超过阈值的转换产生事件 / 中断 …… 0x1：模拟看门狗 2 监测 2 次超过阈值的转换产生事件 / 中断 0x0：模拟看门狗 2 不做滤波
[19:0] AW2CH	**模拟看门狗 2 通道选择** AW2CH[x] =1：ADC 通道 x 受模拟看门狗 2 保护 AW2CH[x] =0：ADC 通道 x 不受模拟看门狗 2 保护 AW2CH=0x0，模拟看门狗 2 等同于被关闭

14.5.2.31 ADC 偏置补偿寄存器 0（ADCx_OFR0）

- 名称：ADC Offset Register0
- 偏移地址：0xA0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R/W															
0x0															

字段	说明
[15:0] OFFSET	**第 0 组 ADC 偏置补偿系数** ADC 增益后的数据将会加上该系数，完成去偏置操作 本组系数为用户可选的第 0 组系数，输入通道对应的 CALx = 0，则选用本组系数

14.5.2.32 ADC 偏置补偿寄存器 1（ADCx_OFR1）

- 名称：ADC Offset Register1
- 偏移地址：0xA4
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R/W															
0x0															

字段	说明
[15:0] OFFSET	**第 1 组 ADC 偏置补偿系数** ADC 增益后的数据将会加上该系数，完成去偏置操作 本组系数为用户可选的第 1 组系数，输入通道对应的 CALx = 1，则选用本组系数

14.5.2.33 ADC 偏置补偿寄存器 2（ADCx_OFR2）

- 名称：ADC Offset Register2
- 偏移地址：0xA8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R/W															
0x0															

字段	说明
[15:0] OFFSET	**第 2 组 ADC 偏置补偿系数** ADC 增益后的数据将会加上该系数，完成去偏置操作 本组系数为用户可选的第 2 组系数，输入通道对应的 CALx = 2，则选用本组系数

14.5.2.34 ADC 偏置补偿寄存器 3（ADCx_OFR3）

- 名称：ADC Offset Register3
- 偏移地址：0xAC
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R/W															
0x0															

字段	说明
[15:0] OFFSET	**第 3 组 ADC 偏置补偿系数** ADC 增益后的数据将会加上该系数，完成去偏置操作 本组系数为用户可选的第 3 组系数，输入通道对应的 CALx = 3，则选用本组系数 本组系数建议用作温度传感器补偿系数，如果不使用温度传感器，也可以用于其他场景

14.5.2.35 ADC 增益系数寄存器 0（ADCx_GCR0）

- 名称：ADC Gain Coeff Register0
- 偏移地址：0xC0
- 默认值：0x00002000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		COEFF													
		R/W													
		0x2000													

字段	说明
[13:0] COEFF	**第 0 组 ADC 增益补偿系数** ADC 转换后的数据乘上该系数，结果右移 12 位 本组系数为用户可选的第 0 组系数，输入通道对应的 CALx = 0，则选用本组系数

14.5.2.36 ADC 增益系数寄存器 1（ADCx_GCR1）

- 名称：ADC Gain Coeff Register1
- 偏移地址：0xC4
- 默认值：0x00002000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		COEFF													
		R/W													
		0x2000													

字段	说明
[13:0] COEFF	**第 1 组 ADC 增益补偿系数** ADC 转换后的数据乘上该系数，结果右移 12 位 本组系数为用户可选的第 1 组系数，输入通道对应的 CALx = 1，则选用本组系数

14.5.2.37 ADC 增益系数寄存器 2（ADCx_GCR2）

- 名称：ADC Gain Coeff Register2
- 偏移地址：0xC8
- 默认值：0x00002000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		COEFF													
		R/W													
		0x2000													

字段	说明
[13:0] COEFF	**第 2 组 ADC 增益补偿系数** ADC 转换后的数据乘上该系数，结果右移 12 位 本组系数为用户可选的第 2 组系数，输入通道对应的 CALx = 2，则选用本组系数

14.5.2.38 ADC 增益系数寄存器 3（ADCx_GCR3）

- 名称：ADC Gain Coeff Register3
- 偏移地址：0xCC
- 默认值：0x00002000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		COEFF													
		R/W													
		0x2000													

字段	说明
[13:0] COEFF	**第 3 组 ADC 增益补偿系数** ADC 转换后的数据乘上该系数，结果右移 12 位 本组系数为用户可选的第 3 组系数，输入通道对应的 CALx = 3，则选用本组系数 本组系数建议用作温度传感器补偿系数，如果不使用温度传感器，也可以用于其他场景

14.5.2.39 ADC 偏置补偿系数寄存器（ADCx_GOFTR）

- 名称：ADC Single-End Ground Offset Trim Register
- 偏移地址：0xE0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R/W															
0x0															

字段	说明
[15:0] OFFSET	**温度传感器内置 ADC 偏置补偿系数** 本组系数在出厂时校准完毕，使用温度传感器时需读取此偏置补偿系数 软件读取偏置补偿系数后，将补偿系数填入与温度传感器通道对应的 OFRy 寄存器

14.5.2.40 ADC 增益补偿系数寄存器（ADCx_GGCTR）

- 名称：ADC Single-End Ground Gain Coeff Trim Register
- 偏移地址：0xF0
- 默认值：0x00002000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COEFF														
	R/W														
	0x2000														

字段	说明
[14:0] COEFF	**温度传感器内置 ADC 增益补偿系数** 本组系数在出厂时校准完毕，使用温度传感器时需读取此偏置补偿系数 软件读取偏置补偿系数后，将补偿系数填入与温度传感器通道对应的 GCRy 寄存器

14.5.2.41 ADC 数据中断状态寄存器（ADCx_DISR）

- 名称：ADC Data Interrupt Status Register
- 偏移地址：0x100
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												DON19	DON18	DON17	DON16
												R/W1C	R/W1C	R/W1C	R/W1C
												0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DON15	DON14	DON13	DON12	DON11	DON10	DON9	DON8	DON7	DON6	DON5	DON4	DON3	DON2	DON1	DON0
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[19] DON19	**通道 19 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[18] DON18	**通道 18 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[17] DON17	**通道 17 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[16] DON16	**通道 16 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[15]	**通道 15 转换完成中断标志**

DON15	1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[14] DON14	**通道 14 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[13] DON13	**通道 13 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[12] DON12	**通道 12 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[11] DON11	**通道 11 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[10] DON10	**通道 10 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[9] DON9	**通道 9 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[8] DON8	**通道 8 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[7] DON7	**通道 7 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[6] DON6	**通道 6 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[5] DON5	**通道 5 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[4] DON4	**通道 4 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[3] DON3	**通道 3 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[2] DON2	**通道 2 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[1] DON1	**通道 1 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0

[0] DON0	**通道 0 转换完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0

14.5.2.42 ADC 数据中断使能寄存器（ADCx_DIER）

- **名称：ADC Data Interrupt Enable Register**
- **偏移地址：0x104**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												DONIE 19	DONIE 18	DONIE 17	DONIE 16
												R/W	R/W	R/W	R/W
												0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DONIE 15	DONIE 14	DONIE 13	DONIE 12	DONIE 11	DONIE 10	DONIE 9	DONIE 8	DONIE 7	DONIE 6	DONIE 5	DONIE 4	DONIE 3	DONIE 2	DONIE 1	DONIE 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[19] DONIE19	**通道 19 转换完成中断使能** 1：中断使能 0：中断不使能
[18] DONIE18	**通道 18 转换完成中断使能** 1：中断使能 0：中断不使能
[17] DONIE17	**通道 17 转换完成中断使能** 1：中断使能 0：中断不使能
[16] DONIE16	**通道 16 转换完成中断使能** 1：中断使能 0：中断不使能
[15] DONIE15	**通道 15 转换完成中断使能** 1：中断使能 0：中断不使能
[14] DONIE14	**通道 14 转换完成中断使能** 1：中断使能 0：中断不使能
[13] DONIE13	**通道 13 转换完成中断使能** 1：中断使能 0：中断不使能
[12] DONIE12	**通道 12 转换完成中断使能** 1：中断使能 0：中断不使能
[11] DONIE11	**通道 11 转换完成中断使能** 1：中断使能 0：中断不使能
[10] DONIE10	**通道 10 转换完成中断使能** 1：中断使能 0：中断不使能
[9] DONIE9	**通道 9 转换完成中断使能** 1：中断使能 0：中断不使能

[8] DONIE8	**通道 8 转换完成中断使能** 1：中断使能 0：中断不使能
[7] DONIE7	**通道 7 转换完成中断使能** 1：中断使能 0：中断不使能
[6] DONIE6	**通道 6 转换完成中断使能** 1：中断使能 0：中断不使能
[5] DONIE5	**通道 5 转换完成中断使能** 1：中断使能 0：中断不使能
[4] DONIE4	**通道 4 转换完成中断使能** 1：中断使能 0：中断不使能
[3] DONIE3	**通道 3 转换完成中断使能** 1：中断使能 0：中断不使能
[2] DONIE2	**通道 2 转换完成中断使能** 1：中断使能 0：中断不使能
[1] DONIE1	**通道 1 转换完成中断使能** 1：中断使能 0：中断不使能
[0] DONIE0	**通道 0 转换完成中断使能** 1：中断使能 0：中断不使能

14.5.2.43 ADC 通道 0 数据寄存器（ADCx_CDR0）

- **名称：ADC Channel Data Register0**
- **偏移地址：0x110**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	**通道 0 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）**

14.5.2.44 ADC 通道 1 数据寄存器（ADCx_CDR1）

- 名称：ADC Channel Data Register1
- 偏移地址：0x114
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 1 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.45 ADC 通道 2 数据寄存器（ADCx_CDR2）

- 名称：ADC Channel Data Register2
- 偏移地址：0x118
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 2 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.46 ADC 通道 3 数据寄存器（ADCx_CDR3）

- 名称：ADC Channel Data Register3
- 偏移地址：0x11C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 3 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.47 ADC 通道 4 数据寄存器（ADCx_CDR4）

- 名称：ADC Channel Data Register4
- 偏移地址：0x120
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 4 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.48 ADC 通道 5 数据寄存器（ADCx_CDR5）

- 名称：ADC Channel Data Register5
- 偏移地址：0x124
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 5 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.49 ADC 通道 6 数据寄存器（ADCx_CDR6）

- 名称：ADC Channel Data Register6
- 偏移地址：0x128
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 6 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.50 ADC 通道 7 数据寄存器（ADCx_CDR7）

- 名称：ADC Channel Data Register7
- 偏移地址：0x12C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 7 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.51 ADC 通道 8 数据寄存器（ADCx_CDR8）

- 名称：ADC Channel Data Register8
- 偏移地址：0x130
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 8 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.52 ADC 通道 9 数据寄存器（ADCx_CDR9）

- 名称：ADC Channel Data Register9
- 偏移地址：0x134
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 9 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.53 ADC 通道 10 数据寄存器（ADCx_CDR10）

- 名称：ADC Channel Data Register10
- 偏移地址：0x138
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 10 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.54 ADC 通道 11 数据寄存器（ADCx_CDR11）

- 名称：ADC Channel Data Register11
- 偏移地址：0x13C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 11 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.55 ADC 通道 12 数据寄存器（ADCx_CDR12）

- 名称：ADC Channel Data Register12
- 偏移地址：0x140
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 12 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.56 ADC 通道 13 数据寄存器（ADCx_CDR13）

- 名称：ADC Channel Data Register13
- 偏移地址：0x144
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 13 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.57 ADC 通道 14 数据寄存器（ADCx_CDR14）

- 名称：ADC Channel Data Register14
- 偏移地址：0x148
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 14 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.58 ADC 通道 15 数据寄存器（ADCx_CDR15）

- 名称：ADC Channel Data Register15
- 偏移地址：0x14C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 15 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.59 ADC 通道 16 数据寄存器（ADCx_CDR16）

- 名称：ADC Channel Data Register16
- 偏移地址：0x150
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 16 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.60 ADC 通道 17 数据寄存器（ADCx_CDR17）

- 名称：ADC Channel Data Register17
- 偏移地址：0x154
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 17 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.61 ADC 通道 18 数据寄存器（ADCx_CDR18）

- 名称：ADC Channel Data Register18
- 偏移地址：0x158
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 18 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.62 ADC 通道 19 数据寄存器（ADCx_CDR19）

- 名称：ADC Channel Data Register19
- 偏移地址：0x15C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDATA															
R															
0x0															

字段	说明
[15:0] CDATA	通道 19 ADC 转换数据结果（若 OVR 位置 1 则该结果不可用）

14.5.2.63 ADC 传输半完成中断状态寄存器（ADCx_HISR）

- 名称：ADC Half Interrupt Status Register
- 偏移地址：0x180
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												HLF19	HLF18	HLF17	HLF16
												R/W1C	R/W1C	R/W1C	R/W1C
												0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLF15	HLF14	HLF13	HLF12	HLF11	HLF10	HLF9	HLF8	HLF7	HLF6	HLF5	HLF4	HLF3	HLF2	HLF1	HLF0
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[19] HLF19	**通道 19 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[18] HLF18	**通道 18 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[17] HLF17	**通道 17 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[16] HLF16	**通道 16 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[15] HLF15	**通道 15 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0

[14] HLF14	**通道 14 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[13] HLF13	**通道 13 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[12] HLF12	**通道 12 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[11] HLF11	**通道 11 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[10] HLF10	**通道 10 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[9] HLF9	**通道 9 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[8] HLF8	**通道 8 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[7] HLF7	**通道 7 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[6] HLF6	**通道 6 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[5] HLF5	**通道 5 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[4] HLF4	**通道 4 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[3] HLF3	**通道 3 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[2] HLF2	**通道 2 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[1] HLF1	**通道 1 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[0] HLF0	**通道 0 DMA 传输半完成中断标志** 1：中断标志产生 0：中断标志未产生

Note：中断标志写 1 清 0

14.5.2.64 ADC 传输半完成中断使能寄存器（ADCx_HIER）

- 名称：**ADC Half Interrupt Enable Register**
- 偏移地址：**0x184**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												HLFIE 19	HLFIE 18	HLFIE 17	HLFIE 16
												R/W	R/W	R/W	R/W
												0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLFIE 15	HLFIE 14	HLFIE 13	HLFIE 12	HLFIE 11	HLFIE 10	HLFIE 9	HLFIE 8	HLFIE 7	HLFIE 6	HLFIE 5	HLFIE 4	HLFIE 3	HLFIE 2	HLFIE 1	HLFIE 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[19] HLFIE19	**通道 19 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[18] HLFIE18	**通道 18 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[17] HLFIE17	**通道 17 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[16] HLFIE16	**通道 16 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[15] HLFIE15	**通道 15 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[14] HLFIE14	**通道 14 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[13] HLFIE13	**通道 13 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[12] HLFIE12	**通道 12 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[11] HLFIE11	**通道 11 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[10] HLFIE10	**通道 10 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[9] HLFIE9	**通道 9 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[8] HLFIE8	**通道 8 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能

[7] HLFIE7	**通道 7 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[6] HLFIE6	**通道 6 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[5] HLFIE5	**通道 5 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[4] HLFIE4	**通道 4 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[3] HLFIE3	**通道 3 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[2] HLFIE2	**通道 2 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[1] HLFIE1	**通道 1 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能
[0] HLFIE0	**通道 0 DMA 传输半完成中断使能** 1：中断使能 0：中断不使能

14.5.2.65 ADC 传输完成中断状态寄存器（ADCx_FISR）

- **名称：ADC Full Interrupt Status Register**
- **偏移地址：0x188**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												FUL19	FUL18	FUL17	FUL16
												R/W1C	R/W1C	R/W1C	R/W1C
												0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUL15	FUL14	FUL13	FUL12	FUL11	FUL10	FUL9	FUL8	FUL7	FUL6	FUL5	FUL4	FUL3	FUL2	FUL1	FUL0
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[19] FUL19	**通道 19 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[18] FUL18	**通道 18 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[17] FUL17	**通道 17 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[16] FUL16	**通道 16 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生

	Note：中断标志写 1 清 0
[15] FUL15	**通道 15 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[14] FUL14	**通道 14 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[13] FUL13	**通道 13 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[12] FUL12	**通道 12 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[11] FUL11	**通道 11 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[10] FUL10	**通道 10 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[9] FUL9	**通道 9 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[8] FUL8	**通道 8 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[7] FUL7	**通道 7 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[6] FUL6	**通道 6 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[5] FUL5	**通道 5 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[4] FUL4	**通道 4 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[3] FUL3	**通道 3 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[2] FUL2	**通道 2 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0
[1] FUL1	**通道 1 DMA 传输完成中断标志** 1：中断标志产生

	0：中断标志未产生 Note：中断标志写 1 清 0
[0] FUL0	**通道 0 DMA 传输完成中断标志** 1：中断标志产生 0：中断标志未产生 Note：中断标志写 1 清 0

14.5.2.66 ADC 传输完成中断使能寄存器（ADCx_FIER）

- **名称：ADC Full Interrupt Enable Register**
- **偏移地址：0x18C**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												FULIE 19	FULIE 18	FULIE 17	FULIE 16
												R/W	R/W	R/W	R/W
												0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FULIE 15	FULIE 14	FULIE 13	FULIE 12	FULIE 11	FULIE 10	FULIE 9	FULIE 8	FULIE 7	FULIE 6	FULIE 5	FULIE 4	FULIE 3	FULIE 2	FULIE 1	FULIE 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[19] FULIE19	**通道 19 传输完成中断使能** 1：中断使能 0：中断不使能
[18] FULIE18	**通道 18 传输完成中断使能** 1：中断使能 0：中断不使能
[17] FULIE17	**通道 17 传输完成中断使能** 1：中断使能 0：中断不使能
[16] FULIE16	**通道 16 传输完成中断使能** 1：中断使能 0：中断不使能
[15] FULIE15	**通道 15 传输完成中断使能** 1：中断使能 0：中断不使能
[14] FULIE14	**通道 14 传输完成中断使能** 1：中断使能 0：中断不使能
[13] FULIE13	**通道 13 传输完成中断使能** 1：中断使能 0：中断不使能
[12] FULIE12	**通道 12 传输完成中断使能** 1：中断使能 0：中断不使能
[11] FULIE11	**通道 11 传输完成中断使能** 1：中断使能 0：中断不使能
[10] FULIE10	**通道 10 传输完成中断使能** 1：中断使能 0：中断不使能
[9] FULIE9	**通道 9 传输完成中断使能**

	1：中断使能 0：中断不使能
[8] FULIE8	**通道 8 传输完成中断使能** 1：中断使能 0：中断不使能
[7] FULIE7	**通道 7 传输完成中断使能** 1：中断使能 0：中断不使能
[6] FULIE6	**通道 6 传输完成中断使能** 1：中断使能 0：中断不使能
[5] FULIE5	**通道 5 传输完成中断使能** 1：中断使能 0：中断不使能
[4] FULIE4	**通道 4 传输完成中断使能** 1：中断使能 0：中断不使能
[3] FULIE3	**通道 3 传输完成中断使能** 1：中断使能 0：中断不使能
[2] FULIE2	**通道 2 传输完成中断使能** 1：中断使能 0：中断不使能
[1] FULIE1	**通道 1 传输完成中断使能** 1：中断使能 0：中断不使能
[0] FULIE0	**通道 0 传输完成中断使能** 1：中断使能 0：中断不使能

14.5.2.67 ADC 传输控制寄存器 0（ADCx_TCR0）

- **名称：ADC Transfer Control Register0**
- **偏移地址：0x190**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零

[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.68 ADC 传输地址寄存器 0（ADCx_TAR0）

- **名称：ADC Transfer Address Register0**
- **偏移地址：0x194**
- **默认值：0x20010000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 0 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.69 ADC 传输长度寄存器 0（ADCx_TLR0）

- **名称：ADC Transfer Length Register0**
- **偏移地址：0x198**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 0 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.70 ADC 传输控制寄存器 1（ADCx_TCR1）

- 名称：ADC Transfer Control Register1
- 偏移地址：0x1A0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.71 ADC 传输地址寄存器 1（ADCx_TAR1）

- 名称：ADC Transfer Address Register0
- 偏移地址：0x194
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 1 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.72 ADC 传输长度寄存器 1（ADCx_TLR1）

- 名称：ADC Transfer Length Register1
- 偏移地址：0x1A8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 1 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.73 ADC 传输控制寄存器 2（ADCx_TCR2）

- 名称：ADC Transfer Control Register2
- 偏移地址：0x1B0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.74 ADC 传输地址寄存器 2（ADCx_TAR2）

- 名称：ADC Transfer Address Register2
- 偏移地址：0x1B4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 2 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.75 ADC 传输长度寄存器 2（ADCx_TLR2）

- 名称：ADC Transfer Length Register2
- 偏移地址：0x1B8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 2 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.76 ADC 传输控制寄存器 3（ADCx_TCR3）

- 名称：ADC Transfer Control Register3
- 偏移地址：0x1C0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.77 ADC 传输地址寄存器 3（ADCx_TAR3）

- 名称：ADC Transfer Address Register3
- 偏移地址：0x1C4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 3 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.78 ADC 传输长度寄存器 3（ADCx_TLR3）

- 名称：ADC Transfer Length Register3
- 偏移地址：0x1C8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 3 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.79 ADC 传输控制寄存器 4（ADCx_TCR4）

- 名称：ADC Transfer Control Register4
- 偏移地址：0x1D0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.80 ADC 传输地址寄存器 4（ADCx_TAR4）

- 名称：ADC Transfer Address Register4
- 偏移地址：0x1D4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 4 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.81 ADC 传输长度寄存器 4（ADCx_TLR4）

- 名称：ADC Transfer Length Register4
- 偏移地址：0x1D8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 4 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.82 ADC 传输控制寄存器 5（ADCx_TCR5）

- 名称：ADC Transfer Control Register5
- 偏移地址：0x1E0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.83 ADC 传输地址寄存器 5（ADCx_TAR5）

- 名称：ADC Transfer Address Register5
- 偏移地址：0x1E4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 5 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.84 ADC 传输长度寄存器 5（ADCx_TLR5）

- 名称：ADC Transfer Length Register5
- 偏移地址：0x1E8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 5 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.85 ADC 传输控制寄存器 6（ADCx_TCR6）

- 名称：ADC Transfer Control Register6
- 偏移地址：0x1F0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.86 ADC 传输地址寄存器 6（ADCx_TAR6）

- 名称：ADC Transfer Address Register6
- 偏移地址：0x1F4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 6 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.87 ADC 传输长度寄存器 6（ADCx_TLR6）

- 名称：ADC Transfer Length Register6
- 偏移地址：0x1F8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 6 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.88 ADC 传输控制寄存器 7（ADCx_TCR7）

- 名称：ADC Transfer Control Register7
- 偏移地址：0x200
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.89 ADC 传输地址寄存器 7（ADCx_TAR7）

- 名称：ADC Transfer Address Register7
- 偏移地址：0x204
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 7 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.90 ADC 传输长度寄存器 7（ADCx_TLR7）

- 名称：ADC Transfer Length Register7
- 偏移地址：0x208
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 7 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.91 ADC 传输控制寄存器 8（ADCx_TCR8）

- 名称：ADC Transfer Control Register8
- 偏移地址：0x210
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.92 ADC 传输地址寄存器 8（ADCx_TAR8）

- 名称：ADC Transfer Address Register8
- 偏移地址：0x214
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 8 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.93 ADC 传输长度寄存器 8（ADCx_TLR8）

- 名称：ADC Transfer Length Register8
- 偏移地址：0x218
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 8 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.94 ADC 传输控制寄存器 9（ADCx_TCR9）

- 名称：ADC Transfer Control Register9
- 偏移地址：0x220
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.95 ADC 传输地址寄存器 9（ADCx_TAR9）

- 名称：ADC Transfer Address Register9
- 偏移地址：0x224
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 9 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.96 ADC 传输长度寄存器 9（ADCx_TLR9）

- 名称：ADC Transfer Length Register9
- 偏移地址：0x228
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 9 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.97 ADC 传输控制寄存器 10（ADCx_TCR10）

- 名称：ADC Transfer Control Register10
- 偏移地址：0x230
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.98 ADC 传输地址寄存器 10（ADCx_TAR10）

- 名称：ADC Transfer Address Register10
- 偏移地址：0x234
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 10 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.99 ADC 传输长度寄存器 10（ADCx_TLR10）

- 名称：ADC Transfer Length Register10
- 偏移地址：0x238
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 10 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.100 ADC 传输控制寄存器 11（ADCx_TCR11）

- 名称：ADC Transfer Control Register11
- 偏移地址：0x240
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.101 ADC 传输地址寄存器 11（ADCx_TAR11）

- 名称：ADC Transfer Address Register11
- 偏移地址：0x244
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 11 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.102 ADC 传输长度寄存器 11（ADCx_TLR11）

- 名称：ADC Transfer Length Register11
- 偏移地址：0x248
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 11 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.103 ADC 传输控制寄存器 12（ADCx_TCR12）

- 名称：ADC Transfer Control Register12
- 偏移地址：0x250
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.104 ADC 传输地址寄存器 12（ADCx_TAR12）

- 名称：ADC Transfer Address Register12
- 偏移地址：0x254
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 12 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.105 ADC 传输长度寄存器 12（ADCx_TLR12）

- 名称：ADC Transfer Length Register12
- 偏移地址：0x258
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 12 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.106 ADC 传输控制寄存器 13（ADCx_TCR13）

- 名称：ADC Transfer Control Register13
- 偏移地址：0x260
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.107 ADC 传输地址寄存器 13（ADCx_TAR13）

- 名称：ADC Transfer Address Register13
- 偏移地址：0x264
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 13 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.108 ADC 传输长度寄存器 13（ADCx_TLR13）

- 名称：ADC Transfer Length Register13
- 偏移地址：0x268
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 13 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.109 ADC 传输控制寄存器 14（ADCx_TCR14）

- 名称：ADC Transfer Control Register14
- 偏移地址：0x270
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.110 ADC 传输地址寄存器 14（ADCx_TAR14）

- 名称：ADC Transfer Address Register14
- 偏移地址：0x274
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 14 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.111 ADC 传输长度寄存器 14（ADCx_TLR14）

- 名称：ADC Transfer Length Register14
- 偏移地址：0x278
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 14 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.112 ADC 传输控制寄存器 15（ADCx_TCR15）

- 名称：ADC Transfer Control Register15
- 偏移地址：0x280
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.113 ADC 传输地址寄存器 15（ADCx_TAR15）

- 名称：ADC Transfer Address Register15
- 偏移地址：0x284
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 15 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.114 ADC 传输长度寄存器 15（ADCx_TLR15）

- 名称：ADC Transfer Length Register15
- 偏移地址：0x288
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 15 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.115 ADC 传输控制寄存器 16（ADCx_TCR16）

- 名称：ADC Transfer Control Register16
- 偏移地址：0x290
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.116 ADC 传输地址寄存器 16（ADCx_TAR16）

- 名称：ADC Transfer Address Register16
- 偏移地址：0x294
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 16 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.117 ADC 传输长度寄存器 16（ADCx_TLR16）

- 名称：ADC Transfer Length Register16
- 偏移地址：0x298
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 16 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.118 ADC 传输控制寄存器 17（ADCx_TCR17）

- 名称：ADC Transfer Control Register17
- 偏移地址：0x2A0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.119 ADC 传输地址寄存器 17（ADCx_TAR17）

- 名称：ADC Transfer Address Register17
- 偏移地址：0x2A4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 17 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.120 ADC 传输长度寄存器 17（ADCx_TLR17）

- 名称：ADC Transfer Length Register17
- 偏移地址：0x2A8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 17 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.121 ADC 传输控制寄存器 18（ADCx_TCR18）

- 名称：ADC Transfer Control Register18
- 偏移地址：0x2B0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.122 ADC 传输地址寄存器 18（ADCx_TAR18）

- 名称：ADC Transfer Address Register18
- 偏移地址：0x2B4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 18 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.123 ADC 传输长度寄存器 18（ADCx_TLR18）

- 名称：ADC Transfer Length Register18
- 偏移地址：0x2B8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 18 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.124 ADC 传输控制寄存器 19（ADCx_TCR19）

- 名称：ADC Transfer Control Register19
- 偏移地址：0x2C0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIX	CIRC	STP	START
												R/W	R/W	R/WAC	R/WAC
												0x0	0x0	0x0	0x0

字段	说明
[3] FIX	**DMA 固定地址选择位** 0：DMA 固定地址不开启 1：DMA 固定地址开启
[2] CIRC	**DMA 循环模式选择位** 0：DMA 循环模式不开启 1：DMA 循环模式开启
[1] STP	**DMA 模式停止位** 0：DMA 模式不停止 1：DMA 模式写 1 停止，传输结束后自动清零
[0] START	**DMA 模式启动位** 0：DMA 模式不启动 1：DMA 模式写 1 启动，传输结束后自动清零

14.5.2.125 ADC 传输地址寄存器 19（ADCx_TAR19）

- 名称：ADC Transfer Address Register19
- 偏移地址：0x2C4
- 默认值：0x20010000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
R/W															
0x20010000															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R/W															
0x20010000															

字段	说明
[31:0] ADDR	**通道 19 DMA 传输的首地址** DMA 固定地址开启则 DMA 传输固定于首地址

14.5.2.126 ADC 传输长度寄存器 19（ADCx_TLR19）

- 名称：ADC Transfer Length Register19
- 偏移地址：0x2C8
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LENG												
			R/W												
			0x0												

字段	说明
[12:0] LENG	**通道 19 DMA 传输的长度（单位为 byte）** 最小长度为 2、最大长度为 8190，不支持长度配置为 0

14.5.2.127 ADC 通用控制寄存器（ADCx_CCR）

- 名称：ADC Common Control Registe
- 偏移地址：0x300
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														DELAY	
														R/W	
														0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY												DUAL			
R/W												R/W			
0x0												0x0			

字段	说明
[17:8] DELAY	**ADC DUAL 采样相位延迟（仅 Master ADC 配置有效）** 此位仅在双 ADC 常规交错触发模式下有效
[3:0] DUAL	**ADC DUAL 模式选择（仅 Master ADC 配置有效）** Others：Reserved 7：交替触发模式 6：常规交错模式 5：常规同步模式 4：注入同步模式 2：常规同步模式+交替触发模式 1：常规同步模式+注入同步模式 0：独立模式 ADC QUAD 模式选择（仅 ADC0/ADC2 同时配置有效） Others：Reserved 13：常规同步模式 12：注入同步模式

14.5.2.128 ADC 通用状态寄存器（ADCx_CSR）

- 名称：ADC Common Status Register
- 偏移地址：0x304
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						EOSMP_SLV	ADRDY_SLV	AWD2_SLV	AWD1_SLV	AWD0_SLV	OVR_SLV	JEOS_SLV	JEOC_SLV	EOS_SLV	EOC_SLV
						R	R	R	R	R	R	R	R	R	R
						0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						EOSMP_MST	ADRDY_MST	AWD2_MST	AWD1_MST	AWD0_MST	OVR_MST	JEOS_MST	JEOC_MST	EOS_MST	EOC_MST
						R	R	R	R	R	R	R	R	R	R
						0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[25] EOSMP_SLV	**采样阶段结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[24] ADRDY_SLV	**ADC 就绪中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[23] AWD2_SLV	**模拟看门狗 2 中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[22] AWD1_SLV	**模拟看门狗 1 中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[21] AWD0_SLV	**模拟看门狗 0 中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[20] OVR_SLV	**ADC 数据溢出中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[19] JEOS_SLV	**注入序列结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[18] JEOC_SLV	**注入转换结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[17] EOS_SLV	**常规序列结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[16] EOC_SLV	**常规转换结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生

	0：中断标志未产生
[9] EOSMP_MST	**采样阶段结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[8] ADRDY_MST	**ADC 就绪中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[7] AWD2_MST	**模拟看门狗 2 中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[6] AWD1_MST	**模拟看门狗 1 中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[5] AWD0_MST	**模拟看门狗 0 中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[4] OVR_MST	**ADC 数据溢出中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[3] JEOS_MST	**注入序列结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[2] JEOC_MST	**注入转换结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[1] EOS_MST	**常规序列结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生
[0] EOC_MST	**常规转换结束中断标志（仅 Master ADC 读取有效）** 此位为 ADCx_ISR 寄存器中相应标志的副本，不可写 1 清 0 1：中断标志产生 0：中断标志未产生

14.5.2.129 ADC 共用常规序列数据寄存器（ADCx_CDR）

- 名称：ADC Common Regular Data Register
- 偏移地址：0x308
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA_SLV															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA_MST															
R/W															
0x0															

字段	说明
[31:16] RDATA_SLV	Slave ADC 常规序列已转换的数据（仅 Master ADC 读取有效）
[15:0] RDATA_MST	Master ADC 常规序列已转换的数据（仅 Master ADC 读取有效）

15 数模转换器（DAC）

15.1 简介

DAC 模块是 12 位电压输出数模转换器。DAC 数据为 12 位模式，在 12 位模式下，数据可以左对齐或右对齐。DAC 支持最多 3 个输出通道，DAC0~DAC2 支持通道输出（运放/不加运放）。DAC 每个转换器的转换独立进行，支持独立的三角波与锯齿波发生器。

15.2 主要特性

DAC 的主要功能如下：

- 9 个 DAC 接口，DAC0~DAC2 支持通道输出（运放/不加运放）
- 支持三角波生成模式
- 支持锯齿波生成模式
- 支持软件触发转换
- 支持外部事件触发转换
- 支持连接到片上外设（比较器）

15.3 结构框图

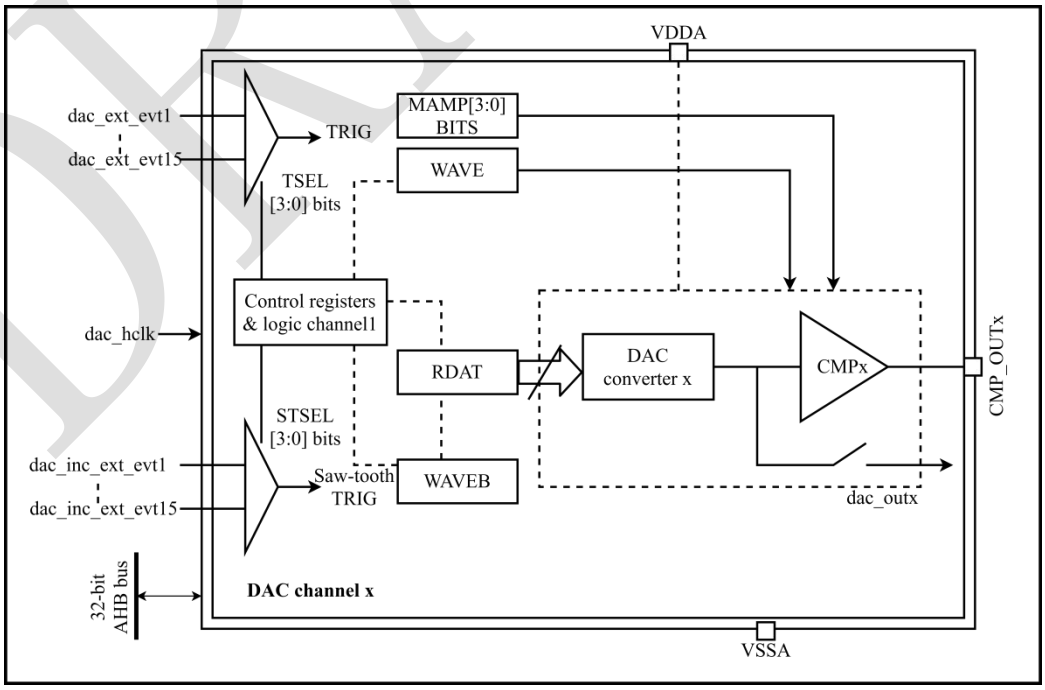


图 15-1 DAC 结构框图

15.4 功能描述

15.4.1 DAC 引脚和内部信号

表 15-1 DAC 输入/输出引脚

Pin name	Signal type	Remarks
AVCC	模拟电源输入	模拟电源
AVSS	模拟电源地输入	模拟地
VCC	模拟电源输入	模拟电源
VDD	数字电源输入	数字电源
VSS	数字地输入	数字地

表 15-2 DAC 输入/输出信号

Internal signal name	Signal type	Description
dac0_ext_evt[15:0]	输入	DAC 通道 0 外部事件输入
dac1_ext_evt[15:0]	输入	DAC 通道 1 外部事件输入
dac2_ext_evt[15:0]	输入	DAC 通道 2 外部事件输入
dac3_ext_evt[15:0]	输入	DAC 通道 3 外部事件输入
dac4_ext_evt[15:0]	输入	DAC 通道 4 外部事件输入
dac5_ext_evt[15:0]	输入	DAC 通道 5 外部事件输入
dac6_ext_evt[15:0]	输入	DAC 通道 6 外部事件输入
dac7_ext_evt[15:0]	输入	DAC 通道 7 外部事件输入
dac8_ext_evt[15:0]	输入	DAC 通道 8 外部事件输入
dac0_inc_ext_evt[15:0]	输入	DAC 通道 0 增量外部事件输入
dac1_inc_ext_evt[15:0]	输入	DAC 通道 1 增量外部事件输入
dac2_inc_ext_evt[15:0]	输入	DAC 通道 2 增量外部事件输入
dac3_inc_ext_evt[15:0]	输入	DAC 通道 3 增量外部事件输入
dac4_inc_ext_evt[15:0]	输入	DAC 通道 4 增量外部事件输入
dac5_inc_ext_evt[15:0]	输入	DAC 通道 5 增量外部事件输入
dac6_inc_ext_evt[15:0]	输入	DAC 通道 6 增量外部事件输入
dac7_inc_ext_evt[15:0]	输入	DAC 通道 7 增量外部事件输入
dac8_inc_ext_evt[15:0]	输入	DAC 通道 8 增量外部事件输入
dac_hclk	输入	DAC 外设时钟
DAC0_OUT	模拟输出	DAC 通道 0 输出
DAC1_OUT	模拟输出	DAC 通道 1 输出
DAC2_OUT	模拟输出	DAC 通道 2 输出
DAC3_OUT	模拟输出	DAC 通道 3 输出
DAC4_OUT	模拟输出	DAC 通道 4 输出
DAC5_OUT	模拟输出	DAC 通道 5 输出
DAC6_OUT	模拟输出	DAC 通道 6 输出

DAC7_OUT	模拟输出	DAC 通道 7 输出
DAC8_OUT	模拟输出	DAC 通道 8 输出

15.4.2 DAC 通道使能

将 DACx_CR.PEN 置 1，即可使能对应 DAC 通道。通道使能后需经过一段时间，完成 DAC 的转换。如果将 DACx_CR.OEN 置 1，则 DAC 转换后将输出到相应的 IO 上。

- PEN 位只会使能模拟 DAC 通道。即使将 DACx_CR.PEN 位复位，DAC 通道 x 数字接口仍处于使能状态。
- 即便不使能 DAC 输出功能（DACx_CR.OEN=0），DAC 内部模拟电路同样会根据配置进行转换，并可以通过内部信号传递到相关的外设（例如 CMP 模块）。
- PEN 位需要在 DAC 参考使能 100us 后使能，否则容易造成 DAC 参考建立不完全，此时 DAC 输出值不符合预期（DAC 参考见 RCU 部分）

15.4.3 DAC 转换输出模式

DAC 模块支持两种输出模式：

- 直接输出模式
- 波形输出模式

15.4.3.1 DAC 直接输出模式

DAC 配置使能（DACx_CR.PEN=1）后，假如触发不使能（DACx_CR.TEN=0），转换数据写入到 DACx_WDR 寄存器后，在 1 个 HCLK 时钟周期后，写入的数据自动转移到 DACx_RDR 寄存器，同时 DAC 模块内部立即开始进行数模转换。经过数模转换时间 $t_{SETTING}$ 后，如果使能 DAC 输出功能（DACx_CR.OEN=1）并配置了相应的引脚复用，则 DAC 模块将转换后的电平输出到相应的复用引脚上。

DAC 配置使能（DACx_CR.PEN=1）后，假如触发使能（DACx_CR.TEN=1），转换数据写入到 DACx_WDR 寄存器后，触发事件到来 1 个 HCLK 时钟周期后，写入的数据转移到 DACx_RDR 寄存器，同时 DAC 模块内部开始进行数模转换。经过数模转换时间 $t_{SETTING}$ 后，如果使能 DAC 输出功能（DACx_CR.OEN=1）并配置了相应的引脚复用，则 DAC 模块将转换后的电平输出到相应的复用引脚上。

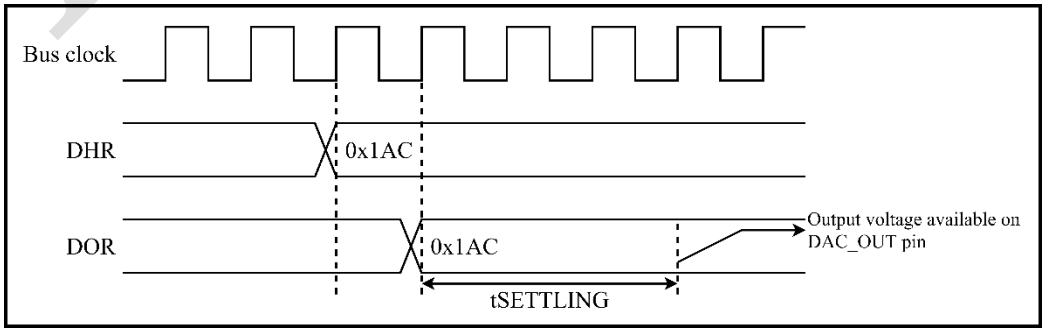


图 15-2　DAC 转换时序图（PEN=1，TEN=0）

注意：

- *直接输出模式 DACx_CR.PEN=0，数据写入 DACx_WDR 寄存器后，DAC 内部将立即开始数模转换*
- *DAC 要经过数模转换时间 $t_{SETTING}$ 才完成数模转换，具体时间还取决于电源电压和模拟输出负载*

15.4.3.2 DAC 波形输出模式

DAC 的波形输出模式，目前支持指定两种波形输出：

- 三角波输出（Triangle Wave，简称 TG）
- 锯齿波输出（Sawtooth Wave，简称 ST）

通过配置 DACx_CR 寄存器中的 TGE 或 STE 置 1，使能三角波或锯齿波输出模式。

DAC 的波形输出模式下，波形的输出支持软件（SWT）或外部事件（EXT）两种触发方式，通过配置 DAC 控制寄存器 DACx_CR.TGTRIG（三角波触发源选择）位域，或 DACx_CR.STINCTRIG/ DACx_CR.STRSTTRIG（锯齿波步进/复位触发源选择）位域进行触发事件的选择。每次软件或外部事件的触发事件都将引发一次 DAC 触发事件。

在 DAC 波形输出模式下，每次触发事件（软件或外部事件触发）发生后，经过 3 个 HCLK 后将新的数据自动转移到 DACx_RDR 寄存器，同时 DAC 模块内部根据波形输出的配置，立即对波形补偿后新数据进行数模转换，经过数模转换时间 $t_{SETTING}$（详细见数据手册）完成转换后，如果使能了 DAC 输出功能（DACx_CR.OEN=1）并配置了相应的引脚复用，则 DAC 模块将转换后的电平输出到相应的复用引脚上。

注意：

- *同一个 DAC 模块内，仅能配置三角波或者锯齿波输出模式，不可同时配置，如果同时配置的 TGE=1 以及 STE=1，则三角波相关的配置无效。*
- *波形输出模式通过触发事件来触发下一次 DAC 转换，更多触发事件描述请参看 15.4.5 DAC 触发事件选择。*
- *DAC 要经过数模转换时间 $t_{SETTING}$ 才完成数模转换，具体时间还取决于电源电压和模拟输出负载。*

15.4.4 DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 V_{REF} 之间的输出电压。

各个 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$DACoutput = V_{REF} \times \frac{RDARx[11:0]}{4096}$$

15.4.5 DAC 触发事件选择

DAC 的波形输出模式下，触发事件源支持以下两种触发事件：

- SWT 软件触发（Software Trigger）
  通过配置 DAC 软件触发寄存器（DAC_SWTR）中相应位为 1 触发相应的触发事件。
- EXT 外部事件（External event Trigger）
  每个触发源都支持 15 种外部事件选择，以外部事件的上升沿进行触发新的触发事件。

三角波输出模式时（DACx_CR.TGE=1& DACx_CR.STE=0），通过配置 DACx_CR.TGTRIG 选择三角波的触发事件，每次触发事件都将三角波步进输出。触发事件源选择见“表 15-3/5/7/9. DACx 三角波触发事件&锯齿波复位触发事件选择”。

锯齿波输出模式时（DACx_CR.STE=1），通过配置 DACx_CR.STRSTTRIG 选择锯齿波的复位触发事件，配置 DACx_CR.STINCTRIG 选择锯齿波的步进触发事件，每次触发事件都将触发锯齿波步进或复位输出。

注意：触发源配置必须在 DAC 使能前（DACx_CR.PEN=1），使能后不可再改变。

表 15-3　DAC0 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG0[3:0]/STRSTTRIG0[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG0	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-4　DAC0 锯齿波步进触发事件选择

Source	Type	STINCTRIG0[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111

HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-5　DAC1 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG1[3:0]/STRSTTRIG1[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG1	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-6　DAC1 锯齿波步进触发事件选择

Source	Type	STINCTRIG1[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010

HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-7　DAC2 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG2[3:0]/STRSTTRIG2[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG2	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-8　DAC2 锯齿步进触发事件选择

Source	Type	STINCTRIG2[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-9　DAC3 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG3[3:0]/STRSTTRIG3[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG0	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-10　DAC3 锯齿波步进触发事件选择

Source	Type	STINCTRIG3[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-11　DAC4 三角波触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG0[3:0]/STRSTTRIG0[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG1	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-12　DAC4 锯齿波步进触发事件选择

Source	Type	STINCTRIG0[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-13　DAC5 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG1[3:0]/STRSTTRIG1[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG2	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-14　DAC5 锯齿波步进触发事件选择

Source	Type	STINCTRIG1[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-15　DAC6 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG2[3:0]/STRSTTRIG2[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG0	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-16　DAC6 锯齿步进触发事件选择

Source	Type	STINCTRIG2[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-17　DAC7 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG3[3:0]/STRSTTRIG3[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG1	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-18　DAC7 锯齿波步进触发事件选择

Source	Type	STINCTRIG3[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

表 15-19　DAC8 通道触发事件&锯齿波复位触发事件选择

Source	Type	TGTRIG3[3:0]/STRSTTRIG3[3:0]
SWT	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR9_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG0	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG2	来自片上 PWM 的内部信号	0101
HRPWM_DAC_RESET_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_RESET_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_RESET_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_RESET_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_RESET_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_RESET_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_RESET_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_RESET_TRG7	来自片上 PWM 的内部信号	1101
HRPWM_DAC_TRG2	来自片上 PWM 的内部信号	1110
PA10	外部引脚	1111

表 15-20　DAC8 锯齿波步进触发事件选择

Source	Type	STINCTRIG3[3:0]
SWTB	软件控制位	0000
TMR7_TRGO	来自片上定时器的内部信号	0001
TMR8_TRGO	来自片上定时器的内部信号	0010
TMR10_TRGO	来自片上定时器的内部信号	0011
HRPWM_ADC_TRG1	来自片上 PWM 的内部信号	0100
HRPWM_ADC_TRG3	来自片上 PWM 的内部信号	0101
HRPWM_DAC_STEP_TRG0	来自片上 PWM 的内部信号	0110
HRPWM_DAC_STEP_TRG1	来自片上 PWM 的内部信号	0111
HRPWM_DAC_STEP_TRG2	来自片上 PWM 的内部信号	1000
HRPWM_DAC_STEP_TRG3	来自片上 PWM 的内部信号	1001
HRPWM_DAC_STEP_TRG4	来自片上 PWM 的内部信号	1010
HRPWM_DAC_STEP_TRG5	来自片上 PWM 的内部信号	1011
HRPWM_DAC_STEP_TRG6	来自片上 PWM 的内部信号	1100
HRPWM_DAC_STEP_TRG7	来自片上 PWM 的内部信号	1101
TMR2_TRGO	来自片上定时器的内部信号	1110
PA11	外部引脚	1111

15.4.6　DAC 三角波生成

可以用于在直流信号或慢变信号上叠加一个小幅三角波。配置三角波输出可按以下步骤：

1. 配置 DACx_CR.TGDIR，选择三角波的起始步进方向；
2. 配置 DACx_CR.TGAMP，选择三角波的最大幅度；
3. 配置 DACx_CR.TGTRIG，选择三角波步进的触发源；
4. 配置 DACx_CR.TGE 为 1，使能三角波输出功能；
5. 向 DACx_WDR 寄存器写入数据，作为三角波输出的基准电压；
6. 配置 DACx_CR.OEN 为 1（如果需要输出到 IO）；
7. 最后配置 DACx_CR.PEN 为 1，使能 DAC 模块。

配置完成后，DAC 模块每次发生触发事件后，经过三个 HCLK 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DACx_WDR 寄存器内容相加，所得总和将传输到 DACx_RDR 寄存器中。只要小于 DACx_CR.TGAMP 定义的最大幅度，三角波计数器就会一直递增。一旦达到配置的幅度后，计数开始递减至 0，然后再次递增，以此类推。

注意：

- *要使能三角波输出，必须保证锯齿波输出使能位置 0（DACx_CR.STE=0），否则将三角波的配置无效；*
- *应当在配置完三角波相关的设置后，最后再使能 DAC 输出功能（DACx_CR.OEN=1）和 DAC 模块（DACx_CR. PEN=1）。*

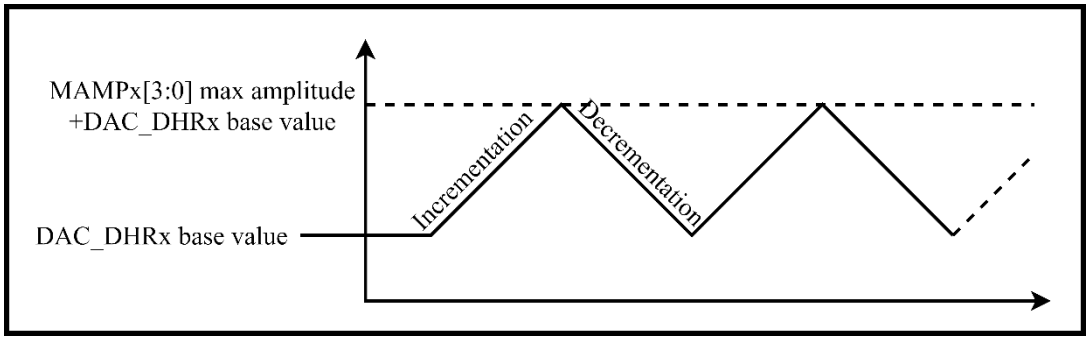


图 15-3　DAC 三角波生成

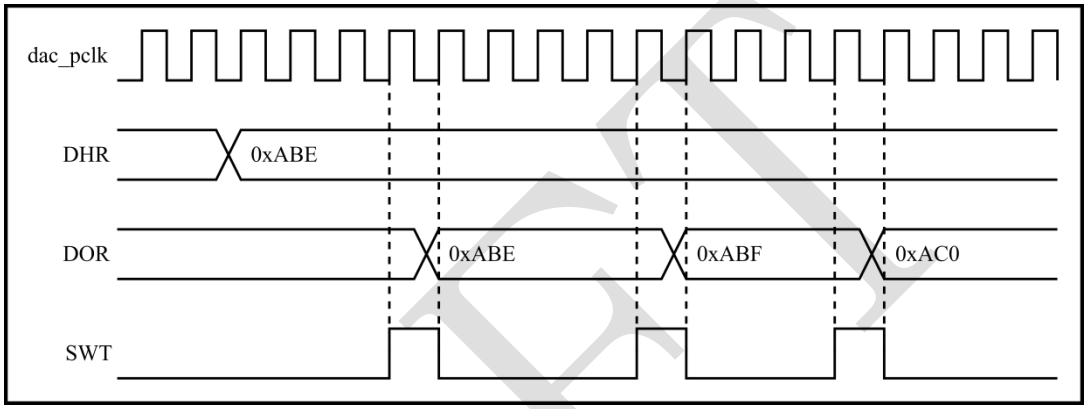


图 15-4　DAC 三角波生成的转换（软件触发模式为例）

15.4.7　DAC 锯齿波生成

DAC 可以生成锯齿波，软件可按以下步骤对初始值、步进值和方向进行特定的寄存器配置：

1. 配置 DACx_CR.STDIR，选择锯齿波的步进方向；
2. 配置 DACx_CR.STINCTRIG，选择锯齿波的步进触发源；
3. 配置 DACx_CR.STRSTTRIG，选择锯齿波的复位触发源；
4. 配置 DACx_SIDR.SID，设置锯齿波的步进步长（12.4 位格式）；
5. 配置 DACx_SRDR.SRD，设置锯齿波的复位值；
6. 配置 DACx_CR.OEN 为 1（如果需要输出到 IO）；
7. 最后配置 DACx_CR.PEN 为 1，使能 DAC 模块。

配置完成后，DAC 模块发生复位触发事件后锯齿波计数从 DACx_SRDR.SRD 开始，后续每个步进触发事件之后向制定的方向步进 DACx_SIDR.SID 值（12.4 位格式，见后续的格式描述）。

DAC 使用锯齿波计数值的高 12 位（位[15:4]）输出。当计数达到 0x0000 或 0xFFFF 时，计数发生饱和。锯齿波复位触发事件将计数值初始化为 DACx_SRDR.SRD 的值。

注意：

- *应当在配置完锯齿波相关的设置后，最后再使能 DAC 输出功能（DACx_CR.OEN =1）和 DAC 模块（DACx_CR. PEN =1）；*
- *锯齿波的复位触发优先级高于步进优先级，如果复位与步进同时触发，则只处理复位事件，步进触发无效；*
- *如果复位触发和步进触发非同时触发，则 DAC 模块按触发信号到达的顺序进行处理。*

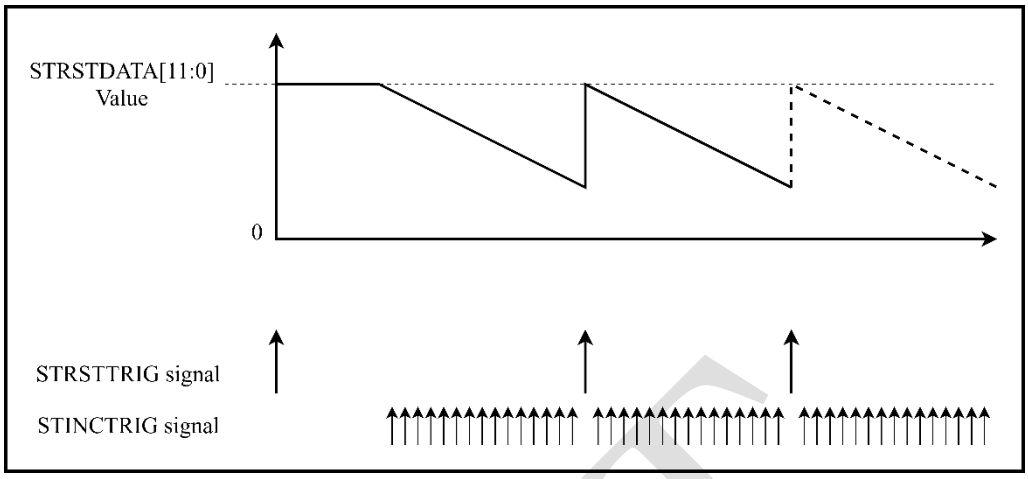


图 15-5　DAC 锯齿波生成示例（STDIR=0）

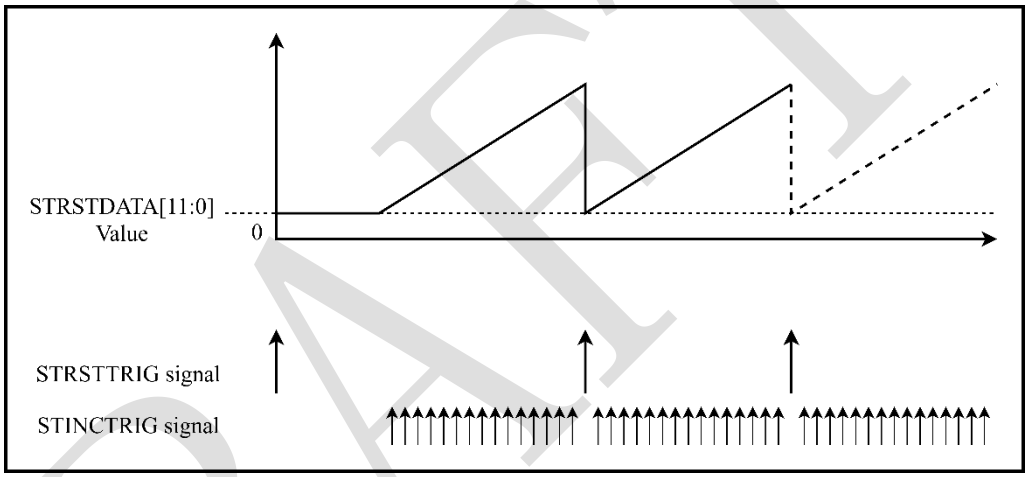


图 15-6　DAC 锯齿波生成示例（STDIR=1）

15.5 寄存器定义

15.5.1 寄存器列表

DAC 基地址：
DAC0(0x40039000) DAC1(0x40039100) DAC2(0x40039200)
DAC3(0x40039300) DAC4(0x40039400) DAC5(0x40039500)
DAC6(0x40039600) DAC7(0x40039700) DAC8(0x40039800)

偏移	实例地址	名称	默认值	描述
0x00	DAC 基地址+0x00	DACx_CR	0x00000000	DACx 配置寄存器
0x04	DAC 基地址+0x04	DACx_WDR	0x00000000	DACx 写入数据寄存器
0x08	DAC 基地址+0x08	DACx_RDR	0x00000000	DACx 读取数据寄存器
0x0C	DAC 基地址+0x0C	DACx_SIDR	0x00000000	DACx 锯齿波形步进数据寄存器
0x10	DAC 基地址+0x10	DACx_SRDR	0x00000000	DACx 锯齿波形复位数据寄存器
0x14	DAC 基地址+0x14	DACx_SWTR	0x00000000	DACx 软件触发寄存器
0x18	DAC 基地址+0x18	DACx_SR	0x00000000	DACx 中断状态寄存器

15.5.2 寄存器描述

15.5.2.1 DACx 配置寄存器（DACx_CR）

- 名称：**DACx Config Register**
- 偏移地址：**0x00**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						STE	STDIR	STINCTRIG				STRSTTRIG			
						R/W	R/W	R/W				R/W			
						0x0	0x0	0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		TGE	TGDIR	TGAMP				TGTRIG				TEN	OEN	BEN	PEN
		R/W	R/W	R/W				R/W				R/W	R/W	R/W	R/W
		0x0	0x0	0x0				0x0				0x0	0x0	0x0	0x0

字段	说明
[25] STE	**锯齿波产生使能位** 0：不使能 1：使能 Note：如果三角波产生同时被使能（TGE=1），则生成锯齿波，三角波配置无效
[24] STDIR	**锯齿波步进方向选择位** 0：向下（幅度减小） 1：向上（幅度增大）
[23:20] STINCTRIG	**锯齿波步进触发源选择位** 0x0：Soft Trigger B 0x1：TMR7_TRGO 0x2：TMR8_TRGO 0x3：TMR10_TRGO 0x4：HRPWM_ADCTRG1 0x5：HRPWM_ADCTRG3 0x6：HRPWM_DACINC0 0x7：HRPWM_DACINC1 0x8：HRPWM_DACINC2 0x9：HRPWM_DACINC3 0xa：HRPWM_DACINC4 0xb：HRPWM_DACINC5 0xc：HRPWM_DACINC6 0xd：HRPWM_DACINC7 0xe：TMR2_TRGO 0xf：PA11
[19:16] STRSTTRIG	**锯齿波复位触发源选择位** 0x0：Soft Trigger 0x1：TMR7_TRGO 0x2：TMR8_TRGO 0x3：TMR9_TRGO 0x4：HRPWM_ADCTRG0 0x5：HRPWM_ADCTRG2 0x6：HRPWM_DACRST0 0x7：HRPWM_DACRST1 0x8：HRPWM_DACRST2 0x9：HRPWM_DACRST3 0xa：HRPWM_DACRST4

	0xb：HRPWM_DACRST5 0xc：HRPWM_DACRST6 0xd：HRPWM_DACRST7 0xe：HRPWM_DACTRGx 0xf：PA10
[13] TGE	**三角波产生使能位** 0：不使能 1：使能
[12] TGDIR	**三角波起始方向选择** 0：幅度变小 1：幅度变大
[11:8] TGAMP	**三角波增长最大幅度选择位** 0x0：幅度为 1 0x1：幅度为 3 0x2：幅度为 7 0x3：幅度为 15 0x4：幅度为 31 0x5：幅度为 63 0x6：幅度为 127 0x7：幅度为 255 0x8：幅度为 511 0x9：幅度为 1023 0xa：幅度为 2047 0xb：幅度为 4095 Others：Reserved
[7:4] TGTRIG	**三角波触发源选择位** 0x0：Soft Trigger 0x1：TMR7_TRGO 0x2：TMR8_TRGO 0x3：TMR9_TRGO 0x4：HRPWM_ADCTRG0 0x5：HRPWM_ADCTRG2 0x6：HRPWM_DACRST0 0x7：HRPWM_DACRST1 0x8：HRPWM_DACRST2 0x9：HRPWM_DACRST3 0xa：HRPWM_DACRST4 0xb：HRPWM_DACRST5 0xc：HRPWM_DACRST6 0xd：HRPWM_DACRST7 0xe：HRPWM_DACTRGx 0xf：PA10
[3] TEN	**DAC 触发转换使能位** 0：不使能 1：使能
[2] OEN	**DAC Buffer 输出使能位** 此位仅 DAC0/1/2 配置有效 0：不使能 1：使能
[1] BEN	**DAC Bypass Buffer 输出使能位** 此位仅 DAC0/1/2 配置有效 0：不使能 1：使能
[0] PEN	**DAC 使能位** 0：不使能 1：使能 Note：DAC 参考使能 100 us 后才可以使能 DAC 外设，DAC 参考使能见 RCU 部分

15.5.2.2 DACx 写入数据寄存器（DACx_WDR）

- 名称：DACx Write Data Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				WDAT											
				R/W											
				0x0											

字段	说明
[11:0] WDAT	**DAC 写入输出数据** 输出锯齿波（STE=1）波形时，写入该寄存器无效； 输出三角波（TGE=1）波形时，写入该寄存器将作为三角波的起始数据，三角波输出以此数据对应电平作为基准，触发进行幅值增长或减少； 非波形输出且触发使能（TEN=1）时，写入该寄存器将等待触发进行 DAC 转换； 非波形输出且触发不使能（TEN=0）时，写入该寄存器将立即进行 DAC 转换

15.5.2.3 DACx 读取数据寄存器（DACx_RDR）

- 名称：DACx Read Data Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RDAT											
				R											
				0x0											

字段	说明
[11:0] RDAT	**DAC 读取输出数据（斜坡补偿后）** 输出波形模式下（三角波/锯齿波），读取该寄存器的值为经过波形补偿后的值

15.5.2.4 DACx 锯齿波形步进数据寄存器（DACx_SIDR）

- 名称：DACx Sawtooth Increment Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID															
R/W															
0x0															

字段	说明
[15:0] SID	**DAC 锯齿波形增量数据** 12.4 位格式（12 位整数部分，4 位小数部分）

15.5.2.5 DACx 锯齿波形复位数据寄存器（DACx_SRDR）

- 名称：DACx Sawtooth Reset Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRD															
R/W															
0x0															

字段	说明
[15:4] SRD	**DAC 锯齿波形复位数据（初始数据）** 12.4 位格式（12 位整数部分，4 位小数部分）

15.5.2.6 DACx 软件触发寄存器（DACx_SWTR）

- **名称：DACx Software Trigger Register**
- **偏移地址：0x14**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														SWTB	SWT
														R/WAC	R/WAC
														0x0	0x0

字段	说明
[1] SWTB	**DAC 软件触发 SWTB 事件** 0：不触发 1：触发 输出锯齿波（STE=1），并且锯齿波复位选择 SWTB 作为触发事件（STRSTTRIG[3:0] 选择 SWTB），将会触发锯齿波步进
[0] SWT	**DAC 软件触发 SWT 事件** 0：不触发 1：触发 输出锯齿波（STE=1），并且锯齿波步进选择 SWT 作为触发事件（STINCTRIG[3:0]选择 SWT），将会触发锯齿波步进 输出三角波（TGE=1），并且三角波步进选择 SWT 作为触发事件（TGTRIG[3:0]选择 SWT），将会触发三角波步进

15.5.2.7 DACx 中断状态寄存器（DACx_SR）

- 名称：DACx Status Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														DONB	DON
														R/W1C	R/W1C
														0x0	0x0

字段	说明
[1] DONB	**DAC DONB 标志位** 0：未完成 1：完成 输出锯齿波（STE=1），触发锯齿波步进（软件或事件触发），DAC 输出完成后，标志位置位。 标志位由软件向该位写 1 清 0。
[0] DON	**DAC DON 中断标志位** 0：未完成 1：完成 输出锯齿波（STE=1），触发锯齿波复位（软件或事件触发），DAC 输出完成后，标志位置位； 输出三角波（TGE=1），触发三角波步进（软件或事件触发），DAC 输出完成后，标志位置位。 标志位由软件向该位写 1 清 0。

16 比较器（CMP）

16.1 简介

芯片内置 9 路模拟比较器通道，可用于各种功能，包括：

- 模拟信号调理
- 与定时器的 OC 输出结合使用时，构成逐周期电流控制环路

16.2 主要特性

- 可选的负端模拟输入
  - I/O 引脚输入（每个比较通道均有两个引脚可选）
  - DAC 通道输出（内部信号）
  - 内部接至 GND
- 可选的正端模拟输入
  - I/O 引脚输入（每个比较通道均有两个引脚可选）
- 可编程迟滞
- 将输出映射到 I/O，并可配消抖后输出
- 将输出重定向到用于触发以下事件的定时器输入
  - 捕获事件
  - 断路事件（用于快速 PWM 关断）
- 消隐比较器输出
- 上升沿和下降沿中断

16.3 结构框图

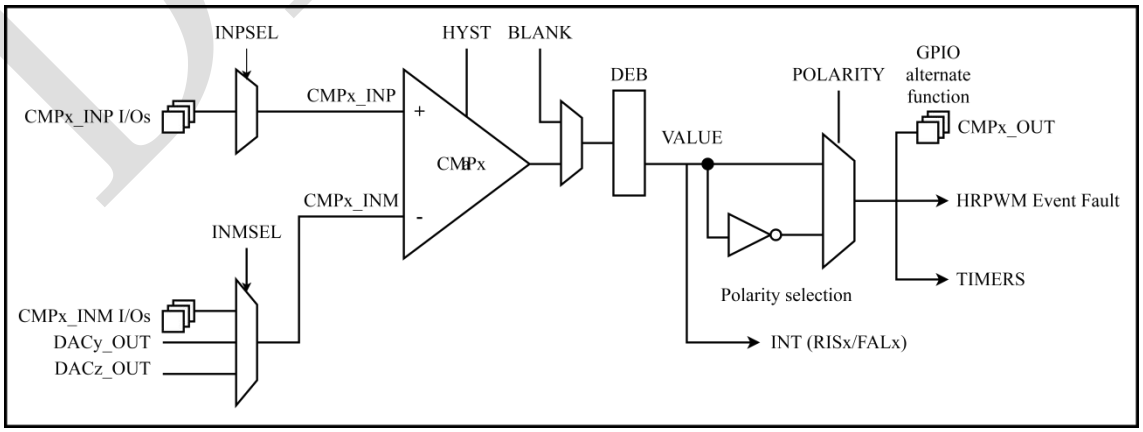


图 16-1　CMP 结构框图

16.4 功能描述

16.4.1 引脚和内部信号

表 16-1 CMPx 正端输入引脚

INPSEL	CMP0_INP	CMP1_INP	CMP2_INP	CMP3_INP	CMP4_INP
0	PA1	PA7	PA0	PB0	PB13
1	PB1	PA3	PC1	PE7	PD12
INPSEL	**CMP5_INP**	**CMP6_INP**	**CMP7_INP**	**CMP8_INP**	
0	PB11	PB14	PC4	PC5	
1	PD11	PD14	PA8	PA9	

表 16-2 CMPx 负端输入引脚

INMSEL[1:0]	CMP0_INM	CMP1_INM	CMP2_INM	CMP3_INM	CMP4_INM
00	PA4	PA5	PF1	PE8	PB10
01	PA0	PA2	PC0	PB2	PD13
10	DAC3_OUT	DAC4_OUT	DAC3_OUT	DAC4_OUT	DAC5_OUT
11	DAC0_OUT	DAC1_OUT	DAC0_OUT	DAC0_OUT	DAC1_OUT
INMSEL[1:0]	**CMP5_INM**	**CMP6_INM**	**CMP7_INM**	**CMP8_INM**	
00	PD10	PD15	PF0	PE9	
01	PB15	PB12	PA6	PB10	
10	DAC6_OUT	DAC5_OUT	DAC8_OUT	DAC8_OUT	
11	DAC2_OUT	DAC2_OUT	DAC7_OUT	DAC7_OUT	

表 16-3 CMPx 数字输出引脚

-	CMP0_OUT	CMP1_OUT	CMP2_OUT	CMP3_OUT	CMP4_OUT
-	PA0、PA6、PA11、PB8	PA2、PA7、PA12、PB9	PC2、PB13、PB7	PA5、PB1、PB15、PB6	PC7、PA9
	CMP5_OUT	**CMP6_OUT**	**CMP7_OUT**	**CMP8_OUT**	
	PC6、PA10	PC8、PA8	PB14、PC9、PC10	PA1、PA13、PD7、PE0	

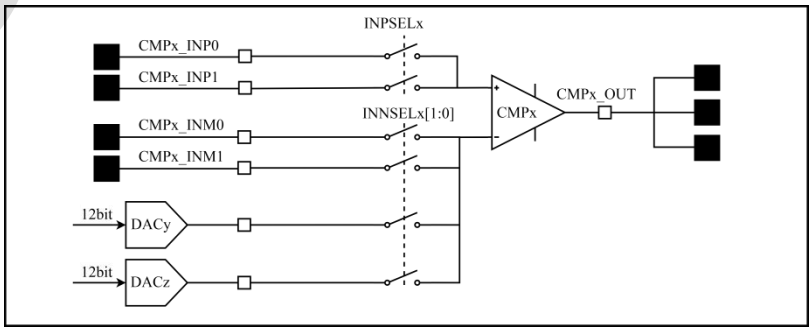


图 16-2 CMP 关系示意图（x=0~8）

16.4.2 迟滞

比较器具有可编程迟滞，可在有信号噪声时避免发生意外输出转换。比较器迟滞是非对称的，仅作用于比较器输出的下降沿。迟滞可在不需要时禁止，以便使用外部组件强制迟滞功能。

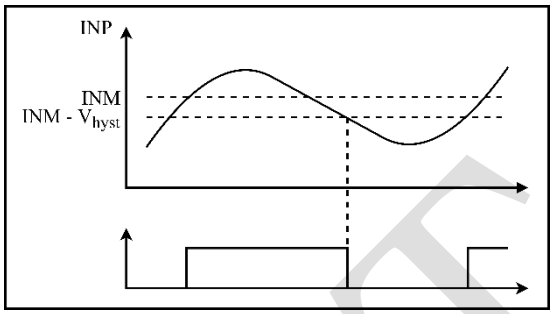


图 16-3　比较器迟滞

16.4.3 输出消隐

消隐功能的目的是防止电流调节在 PWM 周期开始处出现短暂电流尖峰（通常是功率开关反向并联二极管中的恢复电流）时发生跳闸。该功能使用通过定时器输出比较信号定义的消隐窗口。比较器消隐源通过软件配置相应的 CMPx_CR.BLANKING 来选择，如表 16-4 所示。消隐信号对内部比较器输出进行门控，以便使 cmp_out 免受因电流尖峰而导致的寄生脉冲的干扰，如图 16-4 所示。

表 16-4　CMP 输入消隐源

BLANK [2:0]	CMP0	CMP1	CMP2	CMP3	CMP4
000	无消隐	无消隐	无消隐	无消隐	无消隐
001	TMR0_OC0	TMR0_OC0	TMR0_OC0	TMR0_OC0	TMR0_OC0
010	TMR1_OC0	TMR1_OC0	TMR1_OC0	TMR1_OC0	TMR1_OC0
011	TMR2_OC0	TMR2_OC0	TMR2_OC0	TMR2_OC0	TMR2_OC0
100	TMR3_OC0	TMR3_OC0	TMR3_OC0	TMR3_OC0	TMR3_OC0
101	TMR4_OC0	TMR4_OC0	TMR4_OC0	TMR4_OC0	TMR4_OC0
110	TMR9_OC0	TMR9_OC0	TMR9_OC0	TMR9_OC0	TMR9_OC0
111	TMR10_OC0	TMR10_OC0	TMR10_OC0	TMR10_OC0	TMR10_OC0
BLANK [2:0]	**CMP5**	**CMP6**	**CMP7**	**CMP8**	
000	无消隐	无消隐	无消隐	无消隐	
001	TMR0_OC0	TMR0_OC0	TMR0_OC0	TMR0_OC0	
010	TMR1_OC0	TMR1_OC0	TMR1_OC0	TMR1_OC0	
011	TMR2_OC0	TMR2_OC0	TMR2_OC0	TMR2_OC0	
100	TMR3_OC0	TMR3_OC0	TMR3_OC0	TMR3_OC0	
101	TMR4_OC0	TMR4_OC0	TMR4_OC0	TMR4_OC0	
110	TMR9_OC0	TMR9_OC0	TMR9_OC0	TMR9_OC0	
111	TMR10_OC0	TMR10_OC0	TMR10_OC0	TMR10_OC0	

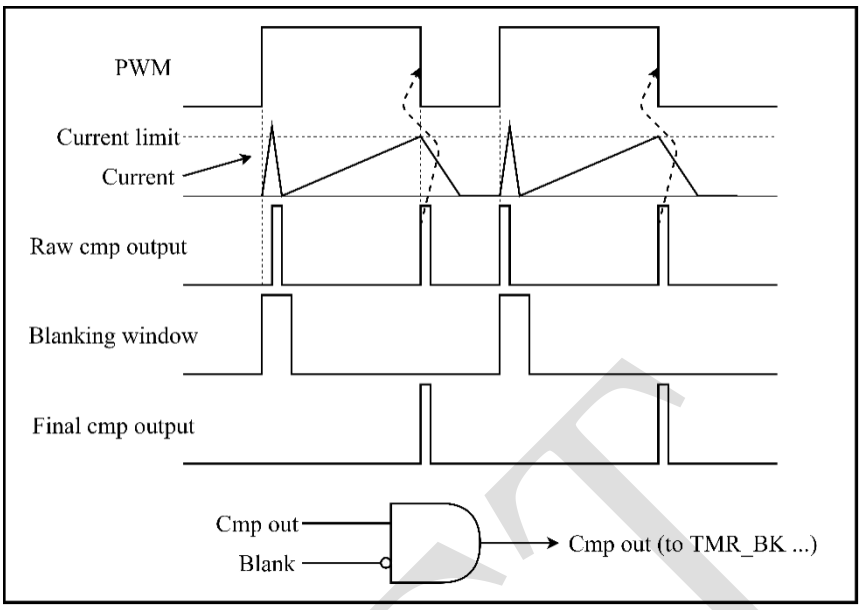


图 16-4　比较器输出消隐

注意：TMRx_OCy 指的是 TIMERx 模块输出 OCy 内部信号，需在相应的 TIMERx 模块内进行相应的配置。

16.4.4　输出重定向

任一 CMP 通道的输出均可重定向到 HRPWM 的事件输入（HRPWM_EVT、HRPWM_FLT），也可重定向作为通用定时器（TIMER）的输入捕获源。

16.4.5　中断

中断向量	中断事件	中断标志	使能控制位	标志清除位
cmp_g0_int	CMPx 上升沿检测事件（x=0/1/2）	RISx	RISIEx	RISx
	CMPx 下降沿检测事件（x=0/1/2）	FALx	FALIEx	FALx
cmp_g1_int	CMPx 上升沿检测事件（x=3/4/5）	RISx	RISIEx	RISx
	CMPx 下降沿检测事件（x=3/4/5）	FALx	FALIEx	FALx
cmp_g2_int	CMPx 上升沿检测事件（x=6/7/8）	RISx	RISIEx	RISx
	CMPx 下降沿检测事件（x=6/7/8）	FALx	FALIEx	FALx

16.5 寄存器定义

16.5.1 寄存器列表

CMP 基地址：
CMP0(0x4003A000) CMP1(0x4003A100) CMP2(0x4003A200)
CMP3(0x4003A300) CMP4(0x4003A400) CMP5(0x4003A500)
CMP6(0x4003A600) CMP7(0x4003A700) CMP8(0x4003A800)

偏移	实例地址	名称	默认值	描述
0x00	CMP 基地址+0x00	CMPx_CR	0x000000F0	CMPx 配置寄存器
0x04	CMP 基地址+0x04	CMPx_DEBR	0x00000000	CMPx 消抖寄存器
0x08	CMP 基地址+0x08	CMPx_IER	0x00000000	CMPx 中断使能寄存器
0x0C	CMP 基地址+0x0C	CMPx_ISR	0x00000000	CMPx 中断状态寄存器

16.5.2 寄存器描述

16.5.2.1 CMPx 配置寄存器（CMPx_CR）

- 名称：**CMPx Config Register**
- 偏移地址：**0x00**
- 默认值：**0x000000F0**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			OPOL	ODEB	BLANKING			INM		HYST		INP			CEN
			R/W	R/W	R/W			R/W		R/W		R/W			R/W
			0x0	0x0	0x0			0x3		0x3		0x0			0x0

字段	说明
[12] OPOL	**比较输出极性选择** 0：输出不翻转 1：输出翻转
[11] ODEB	**比较输出源模式选择** 0：禁止输出消抖 1：使能输出消抖 Note：当配置为消抖后输出，还需要配置 CMPx_DEB 寄存器，设定消抖宽度
[10:8] BLANKING	**比较消隐事件选择** 0：No Blanking 1：TMR0_OC0 2：TMR1_OC0 3：TMR2_OC0 4：TMR3_OC0 5：TMR4_OC0 6：TMR9_OC0 7：TMR10_OC0
[7:6] INM	**比较负端源选择** 0：CMPx_INM0 1：CMPx_INM1 2：DACy_OUT 3：DACz_OUT
[5:4] HYST	**比较迟滞** 0：无迟滞（0 mV） 1：低迟滞（10 mV） 2：中迟滞（20 mV） 3：高迟滞（30 mV）
[3] INP	**比较正端源选择** 0：CMPx_INP0 1：CMPx_INP1
[0] CEN	**比较使能** 0：不使能 1：使能

16.5.2.2 CMPx 消抖寄存器（CMPx_DEBR）

- 名称：CMPx Debounce Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DEB							
								R/W							
								0x0							

字段	说明
[7:0] DEB	**比较输出消抖值** 定义比较输出消抖宽度，比较输入变化的持续时间需要大于 N+1 个系统时钟，才会影响到输出 开启比较输出消抖会引入额外的输出延迟，引入的延迟为 N+3 个系统时钟

16.5.2.3 CMPx 中断使能寄存器（CMPx_IER）

- 名称：CMPx Interrupt Enable Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														FALIE	RISIE
														R/W	R/W
														0x0	0x0

字段	说明
[1] FALIE	**比较下降沿中断使能** 0：不使能 1：使能
[0] RISIE	**比较上升沿中断使能** 0：不使能 1：使能

16.5.2.4 CMPx 中断状态寄存器（CMPx_ISR）

- 名称：CMPx Interrupt Status Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													OVAL	FAL	RIS
													R	R/W1C	R/W1C
													0x0	0x0	0x0

字段	说明
[2] OVAL	**比较输出状态** 0：输出为低（输出极性变化之前，消隐处理之后） 1：输出为高（输出极性变化之前，消隐处理之后）
[1] FAL	**比较下降沿中断标志位** 0：无下降沿（输出极性变化之前，消隐处理之后） 1：有下降沿（输出极性变化之前，消隐处理之后） Note：中断标志位由软件向该位写 1 清 0
[0] RIS	**比较上升沿中断标志位** 0：无上升沿（输出极性变化之前，消隐处理之后） 1：有上升沿（输出极性变化之前，消隐处理之后） Note：中断标志位由软件向该位写 1 清 0

17 高精度脉宽调制器（HRPWM）

17.1 简介

高分辨率脉宽调制器（HRPWM）可生成多达 16 路高精度定时的数字信号，主要用于驱动开关模式电源或照明系统等电源转换系统，但也可用于，一般对时间分辨率有极高要求的应用。

该调制器采用模块化架构，可生成独立波形或耦合波形。波形由独立式定时信号（使用计数器和比较单元）以及多种外部事件（如模拟或数字反馈以及同步信号）确定，因此可生成大量不同的控制信号（PWM、相移、恒定 Ton…），从而满足大部分转换拓扑的需求。

为实现控制和监测用途，该调制器还具有事件触发功能，并连接到内置的 ADC 和 DAC 转换器。此外，该调制器能够处理各种故障机制，从而实现安全关断。

17.2 主要特性

- 多个定时单元
  - 174 ps 分辨率，已针对电压和温度变化进行补偿
  - 所有输出均支持高分辨率，可在各种模式下调整占空比，频率和脉宽
  - 8 个 16 位定时单元（每个定时单元包含一个独立计数器、4 个比较单元和 4 个捕获单元）
  - 16 路输出可通过定时单元控制，每条通道多达 22 个置位/复位源
  - 模块化结构可满足多种配有 1 或 2 个开关的独立转换器的需求，也可满足少数大型多开关拓扑的需求
- 多达 10 个外部事件，可用于任何定时单元
  - 可编程极性和有效边沿
  - 快速异步模式
  - 可编程数字滤波器
  - 使用消隐和加窗模式实现伪事件过滤
- 多条通道可连接到内置模拟外设
  - 10 个连接到 ADC 转换器的触发事件
  - 3 个连接到 DAC 转换器（三角波补偿）的触发事件
  - 8 个连接到 DAC 转换器（锯齿波补偿）的复位事件
  - 8 个连接到 DAC 转换器（锯齿波补偿）的步进事件
- 全面的保护机制
  - 8 个故障输入可组合使用并关联到任何定时单元
  - 可编程极性和有效边沿
  - 可编程数字滤波器
- 多个 HRPWM 单元可与外部同步输入/输出同步
- 多功能输出级
  - 高分辨率的死区插入

- 可编程输出极性
- 斩波模式

- 突发模式控制器，可同时处理多路转换器上的轻载操作
- 10 个中断向量，每个向量最多具有 15 个源
- 9 个 DMA 请求，每个向量最多具有 15 个源，可通过突发 DMA 实现多寄存器更新

17.3 结构框图

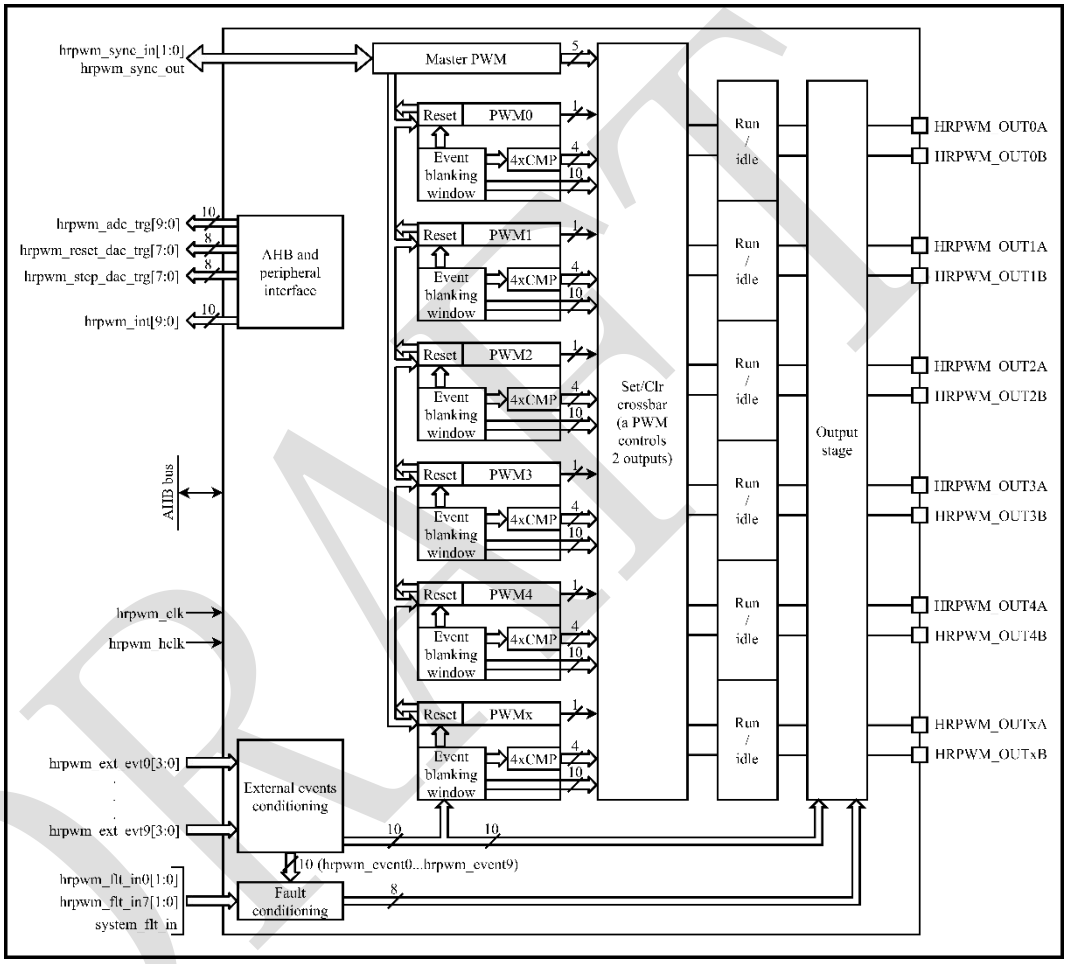


图 17-1　HRPWM 结构框图

17.4 功能描述

17.4.1 常规功能说明

HRPWM 可分为以下几个子实体：

- 主定时器（Master PWM）
- 定时单元（PWM0 至 PWM7）
- 输出级
- 突发模式控制器

- 外部事件和故障信号调节逻辑，由所有定时器共用
- 系统接口

主定时器基于一个 16 位计数器。它可以通过 4 个比较值来置位/复位 16 路输出中的任何一个，并向 8 个定时单元提供同步信号。其主要用途是使各个定时器单元受唯一的时钟源控制。交错降压转换器是一个典型的应用示例，主定时器在其中管理着多个定时单元的相移。

定时器单元既可以独立工作，也可以与其他定时器（包括主定时器）配合工作。每个定时器都可控制两路输出。输出置位/复位事件可以由定时单元的比较寄存器触发，或者由来自主定时器的事件或外部事件触发。

输出级有多种用途：

- 在互补输出模式下为两路输出添加死区
- 将载波频率添加在调制信号上
- 通过将输出异步置为预定义的安全电平来管理故障事件

外部事件和外部故障调节逻辑包括：

- 输入选择 MUX（例如，为给定外部事件通道选择数字输入或片上时钟源）
- 极性和有效边沿编程
- 数字滤波和边沿计数

系统接口允许 HRPWM 与 MCU 的其余部分进行交互：

- 向 CPU 发出中断请求
- 触发 ADC 和 DAC 转换器
- 触发 DAC 转换器进行斜坡补偿

HRPWM 寄存器分为 10 组：

- 主定时器寄存器
- 定时器 0 至定时器 7 寄存器
- 通用寄存器，用于所有定时器单元共用的功能

注意：根据文档编写约定，在文本和寄存器中对 8 个定时单元的引用统一用“x”字母表示（x 可以是 0 到 7 的任意值）。

17.4.2 HRPWM 引脚和内部信号

片上和片外的 HRPWM 输入和输出总结在本节的表 17-1 中。

表 17-1 HRPWM 输入/输出总结

信号名称	信号类型	信号描述
HRPWM_OUT0A HRPWM_OUT0B HRPWM_OUT1A HRPWM_OUT1B HRPWM_OUT2A HRPWM_OUT2B HRPWM_OUT3A	Outputs	外部输出，总共支持 8 对外部输出，每个输出可以独立工作，也可以通过插入死区（HRPWM_OUTxA 和 HRPWM_OUTxB）来成对工作，这些输出都会连接到片外。

HRPWM_OUT3B HRPWM_OUT4A HRPWM_OUT4B HRPWM_OUT5A HRPWM_OUT5B HRPWM_OUT6A HRPWM_OUT6B HRPWM_OUT7A HRPWM_OUT7B		
hrpwm_flt_in0[1:0] hrpwm_flt_in1[1:0] hrpwm_flt_in2[1:0] hrpwm_flt_in3[1:0] hrpwm_flt_in4[1:0] hrpwm_flt_in5[1:0] hrpwm_flt_in6[1:0] hrpwm_flt_in7[1:0]	Digital input	故障事件，总共支持 8 组故障事件，每组故障事件可以从 4 个来源选择，这些来源可以是外部事件，可以是片内比较器输出、也可以是 HRPWM_FLTx 输入引脚的输入，故障事件置位后立即停止 HRPWM 输出并切换为安全电平
system_flt_in	Digital input	系统故障主要指 MCU 内部故障事件，包括 XOSC 时钟丢失，CPU 锁定，LVD 输出等事件。
hrpwm_sync_in[1:0]	Digital Input	同步输入，用于将整个 HRPWM 与其他内部或外部计时器资源同步： hrpwm_sync_in[0]：来源是 TIMER0_TRGO 输出 hrpwm_sync_in[1]：来源是 HRPWM_SCIN 输入引脚
hrpwm_sync_out	Digital output	同步输出，用于级联或同步片上或片外的多个 HRPWM 实例： hrpwm_sync_out：可以通过 HRPWM_SCOUT 输出引脚连接到片外 HRPWM 或其他外设
hrpwm_upd_in[2:0]	Digital input	更新请求输入（片上互连信号），边沿可以触发从预加载寄存器到有效寄存器的更新
hrpwm_bst_in[2:0]	Digital input	突发模式时钟（片上互连信号）
hrpwm_ext_evt0[3:0] hrpwm_ext_evt1[3:0] hrpwm_ext_evt2[3:0] hrpwm_ext_evt3[3:0] hrpwm_ext_evt4[3:0] hrpwm_ext_evt5[3:0] hrpwm_ext_evt6[3:0] hrpwm_ext_evt7[3:0] hrpwm_ext_evt8[3:0] hrpwm_ext_evt9[3:0]	Digital input	外部事件，总共支持 10 组外部事件，每组外部事件可以从 4 个来源中选择 1 个，来源可以是片内其他外设事件，如比较器输出、ADC 模拟看门狗事件、TIMx 定时器触发事件，也可以是片外事件，如 HRPWM_EVTx 引脚的输入
hrpwm_adc_trg0 hrpwm_adc_trg1 hrpwm_adc_trg2 hrpwm_adc_trg3 hrpwm_adc_trg4	Digital output	ADC 触发事件，主要用于 ADC 的转换开启以及 DAC 的三角波生成。后面 hrpwm_adc_trg0 至 hrpwm_adc_trg9 简称为 ADC 事件 0 至 ADC 事件 9

hrpwm_adc_trg5 hrpwm_adc_trg6 hrpwm_adc_trg7 hrpwm_adc_trg8 hrpwm_adc_trg9		
hrpwm_reset_dac_trg0 hrpwm_reset_dac_trg1 hrpwm_reset_dac_trg2 hrpwm_reset_dac_trg3 hrpwm_reset_dac_trg4 hrpwm_reset_dac_trg5 hrpwm_reset_dac_trg6 hrpwm_reset_dac_trg7	Digital output	双通道 DAC 触发事件，主要用于 DAC 锯齿波的复位
hrpwm_step_dac_trg0 hrpwm_step_dac_trg1 hrpwm_step_dac_trg2 hrpwm_step_dac_trg3 hrpwm_step_dac_trg4 hrpwm_step_dac_trg5 hrpwm_step_dac_trg6 hrpwm_step_dac_trg7	Digital output	双通道 DAC 触发事件，主要用于 DAC 锯齿波的步进
hrpwm_mst_int hrpwm_slv0_int hrpwm_slv1_int hrpwm_slv2_int hrpwm_slv3_int hrpwm_slv4_int hrpwm_slv5_int hrpwm_slv6_int hrpwm_slv7_int hrpwm_flt_int	Digital output	中断请求，每个中断请求对应一个中断入口地址
hrpwm_hclk	-	总线时钟（来自 AHB 时钟）
hrpwm_clk	-	功能时钟（来自 RC8M/PLL0 时钟）

表 17-2　HRPWM 与 ADC 事件连接

ADC 触发事件	ADC0	ADC1	ADC2	ADC3
hrpwm_adc_trg0	Yes	Yes	Yes	Yes
hrpwm_adc_trg1	Yes	Yes	Yes	Yes
hrpwm_adc_trg2	Yes	Yes	Yes	Yes
hrpwm_adc_trg3	Yes	Yes	Yes	Yes
hrpwm_adc_trg4	Yes	Yes	Yes	Yes
hrpwm_adc_trg5	Yes	Yes	Yes	Yes
hrpwm_adc_trg6	Yes	Yes	Yes	Yes
hrpwm_adc_trg7	Yes	Yes	Yes	Yes

hrpwm_adc_trg8	Yes	Yes	Yes	Yes
hrpwm_adc_trg9	Yes	Yes	Yes	Yes

表 17-3　HRPWM 与 DAC 事件连接

DAC 触发事件	DAC0	DAC1	DAC2	DAC3	DAC4
hrpwm_adc_trg0	No	No	No	No	No
hrpwm_adc_trg1	No	No	No	No	No
hrpwm_adc_trg2	Yes	Yes	Yes	Yes	Yes
hrpwm_adc_trg3	Yes	Yes	Yes	Yes	Yes
hrpwm_adc_trg4	No	No	No	No	No
hrpwm_adc_trg5	No	No	No	No	No
hrpwm_adc_trg6	No	No	No	No	No
hrpwm_adc_trg7	No	No	No	No	No
hrpwm_adc_trg8	No	No	No	No	No
hrpwm_adc_trg9	No	No	No	No	No
hrpwm_dac_trg0	Yes	No	No	Yes	No
hrpwm_dac_trg1	No	Yes	No	No	Yes
hrpwm_dac_trg2	No	No	Yes	No	No
hrpwm_reset_dac_trg0 hrpwm_step_dac_trg0	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg1 hrpwm_step_dac_trg1	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg2 hrpwm_step_dac_trg2	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg3 hrpwm_step_dac_trg3	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg4 hrpwm_step_dac_trg4	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg5 hrpwm_step_dac_trg5	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg6 hrpwm_step_dac_trg6	Yes	Yes	Yes	Yes	Yes
hrpwm_reset_dac_trg7 hrpwm_step_dac_trg7	Yes	Yes	Yes	Yes	Yes

表 17-4　HRPWM 与 DAC 事件连接

DAC 触发事件	DAC5	DAC6	DAC7	DAC8	
hrpwm_adc_trg0	No	No	No	No	
hrpwm_adc_trg1	No	No	No	No	
hrpwm_adc_trg2	Yes	Yes	Yes	Yes	
hrpwm_adc_trg3	Yes	Yes	Yes	Yes	
hrpwm_adc_trg4	No	No	No	No	
hrpwm_adc_trg5	No	No	No	No	
hrpwm_adc_trg6	No	No	No	No	

hrpwm_adc_trg7	No	No	No	No	
hrpwm_adc_trg8	No	No	No	No	
hrpwm_adc_trg9	No	No	No	No	
hrpwm_dac_trg0	No	Yes	No	No	
hrpwm_dac_trg1	No	No	Yes	No	
hrpwm_dac_trg2	Yes	No	No	Yes	
hrpwm_reset_dac_trg0 hrpwm_step_dac_trg0	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg1 hrpwm_step_dac_trg1	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg2 hrpwm_step_dac_trg2	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg3 hrpwm_step_dac_trg3	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg4 hrpwm_step_dac_trg4	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg5 hrpwm_step_dac_trg5	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg6 hrpwm_step_dac_trg6	Yes	Yes	Yes	Yes	
hrpwm_reset_dac_trg7 hrpwm_step_dac_trg7	Yes	Yes	Yes	Yes	

17.4.3 时钟

HRPWM 必须由 PLL0 提供时钟才能实现高分辨率。FHRPWM 时钟周期被均匀分为多达 32 个中间步长，通过边沿定位逻辑实现。HRPWM 中的所有时钟均由该参考时钟生成。

术语定义（Definition of terms）

FHRPWM ：主 HRPWM 时钟（hrpwm_clk）。所有后续时钟均由该时钟源生成，并与该时钟源同步。

f_{HRCK} ：高分辨率等效时钟。
考虑 f_{HRCK} 周期为 FHRPWM 时钟周期除以 32，f_{HRCK} 等效频率为 180 × 32 = 5.76 GHz。

f_{DTG} ：死区发生器时钟。

f_{CHPFRQ} ：斩波级时钟源。

f_{1STPW} ：定义斩波模式下初始脉冲长度的时钟源。

$f_{SAMPLING}$ ：采样故障输入或外部事件输入所需的时钟。

f_{FLTS} ：派生自 FHRPWM 的时钟，用于对故障事件进行过滤。

f_{EEVS} ：派生自 FHRPWM 的时钟，用于对外部事件进行过滤。

定时器时钟和预分频（Timer clock and prescaler）

HRPWM 中的每个定时器都有独立的时钟预分频，以供用户调整定时器分辨率，如表 17-5 所示。

表 17-5　定时器分辨率和最低 PWM 频率（FHRPWM=180MHz）

CKPSC[2:0]	预分频比例	f_{HRCK} 等效频率	分辨率	最低 PWM 频率
000	1	180 x 32 MHz = 5.76 GHz	173 ps	87.9 kHz
001	2	180 x 16 MHz = 2.88 GHz	347 ps	44.0 kHz
010	4	180 x 8 MHz = 1.44 GHz	694 ps	22.0 kHz
011	8	180 x 4 MHz = 720 MHz	1.39 ns	11.0 kHz
100	16	180 x 2 MHz = 360 MHz	2.77 ns	5.50 kHz
101	32	180 MHz	5.56 ns	2.75 kHz
110	64	180/2 MHz = 90 MHz	11.1 ns	1.37 kHz
111	128	180/4 MHz = 45 MHz	22.2 ns	0.69 kHz

高分辨率可用于边沿定位，PWM 周期调整以及外部触发的脉冲持续时间。

高分辨率不适用于以下功能：

- 定时器计数值的读写访问

对于时钟预分频比低于 32（HRPWM_MCR0.CKPSC 或 HRPWM_PWMxCR0.CKPSC<5）的情形，计数器的最低有效位无意义。最低有效位无法被写入，读取时返回 0。

例如，如果 HRPWM_MCR0.CKPSC 或 HRPWM_PWMxCR0.CKPSC = 2（预分频比例为 4），写入 0xFFFF 到计数值寄存器产生的有效值为 0xFFF8。相反地，介于 0xFFFF 和 0xFFF8 之间的任何计数值都读取为 0xFFF8。

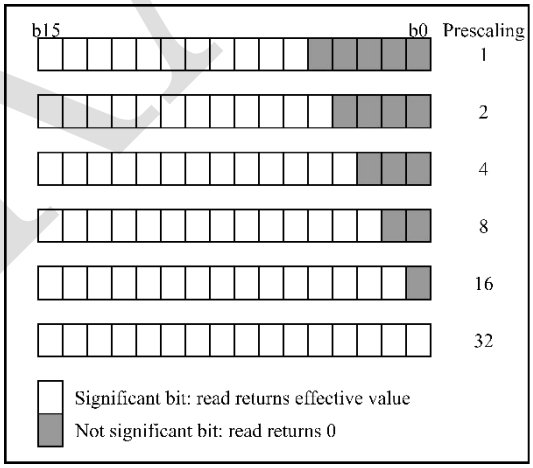


图 17-2　计数寄存器格式与时钟预分频系数

初始化（Initialization）

启动时，必须先初始化预分频位，然后再写入比较值和周期值。定时器使能后（HRPWM_MCR1.MCEN 或 HRPWM_MCR1.CENx 置 1），不能修改预分频。

警告：仅当计数器和输出行为与其他定时器的信息和信号无关时，主定时器和定时器 0~7 中

才可以配置不同的预分频比。如果以下某个事件从一个定时单元（或主定时器）传播到另一个定时单元，必须在这些定时单元中配置相同的预分频比：输出置位/复位事件，计数器复位事件，更新事件。预分频系数不相等会导致结果无法预测。

死区发生器时钟（Deadtime generator clock）

死区时间预分频器由（FHRPWM）/ $2^{(CKPSC[2:0])}$ 提供，通过 HRPWM_PWMxCR0.CKPSC 位进行编程。

对于 FHRPWM = 180 MHz，t_{DTG} 的范围为 173 ps 至 22.2 ns。

斩波级时钟（Chopper stage clock）

斩波级时钟源 f_{CHPFRQ} 由 FHRPWM 生成，使用 16 到 256 的分频系数，对于 FHRPWM = 180 MHz，f_{CHPFRQ} 的范围为 703.125 kHz≤f_{CHPFRQ}≤11.125 MHz。

t_{1STPW} 是斩波模式下的初始脉冲长度，通过 HRPWM_CHPxR.STRPW 进行配置，具体计算公式如下：

$$t_{1STPW} =（STRPW[3:0] +1）\times 16 \times THRPWM。$$

计算时使用 FHRPWM / 16 作为时钟源（对于 FHRPWM = 180 MHz 为 11.125 MHz）。

故障输入采样时钟（Fault input sampling clock）

故障输入噪声抑制滤波器的时间常数由 $f_{SAMPLING}$ 定义，可以为 FHRPWM 或 f_{FLTS}。

f_{FLTS} 生成自 FHRPWM，对于 FHRPWM = 180 MHz，f_{FLTS} 的范围为 180 MHz 至 22.5 MHz。

外部事件输入采样时钟（External event input sampling clock）

故障输入噪声抑制滤波器的时间常数由 $f_{SAMPLING}$ 定义，可以为 FHRPWM 或 f_{EEVS}。

f_{EEVS} 生成自 FHRPWM，对于 FHRPWM = 180 MHz，f_{EEVS} 的范围为 180 MHz 至 22.5 MHz。

17.4.4 定时器 0~7 定时单元

HRPWM 嵌入了 8 路完全相同的定时单元，这些定时单元由采用自动重载机制的 16 位递增计数器组成，用于定义计数周期和 4 个比较单元，如图 17-3 所示。每路单元都包含对 2 路输出的所有控制功能，因此可作为独立定时器工作。

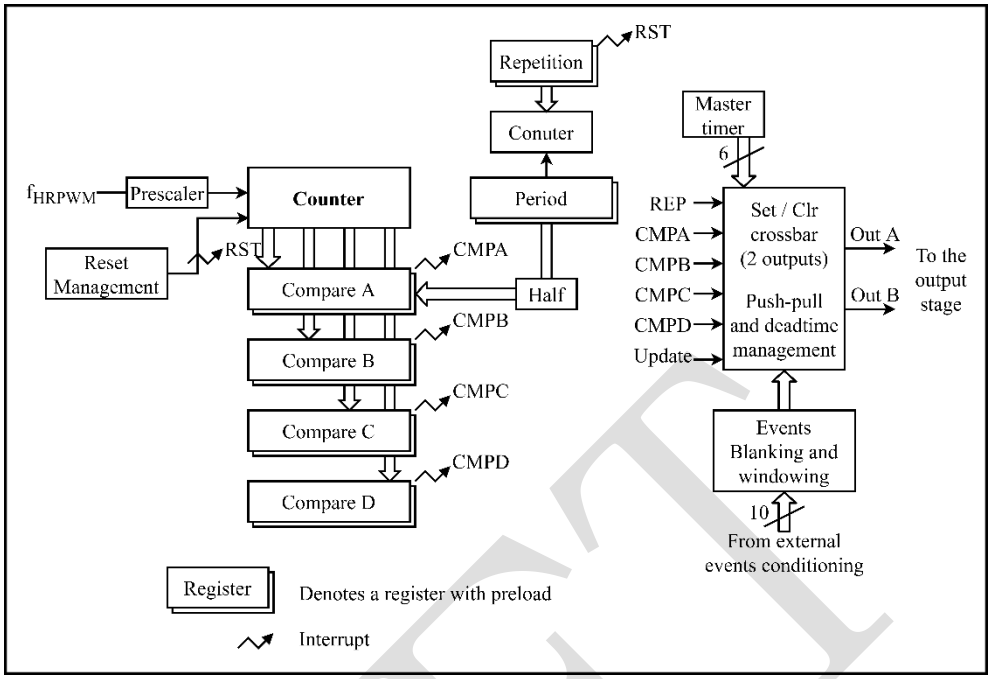


图 17-3　定时器 0~5 概览

周期值和比较值必须在指定的上下限范围内，上下限范围与高分辨率实现相关，具体数值如表 17-6 所示：

- 最小值必须大于或等于 3 个 FHRPWM 时钟周期。
- 最大值必须小于或等于 0xFFFD - 1 个 FHRPWM 时钟周期。

表 17-6 周期值和比较值寄存器的最小值和最大值

CKPSC[2:0]	PERxR Min	PERxR Max	CMPxR Min	CMPxR Max
0	0x0060	0xFFDF	0x0060	PERxR – 0x0040
1	0x0030	0xFFEF	0x0030	PERxR – 0x0020
2	0x0018	0xFFF7	0x0018	PERxR – 0x0010
3	0x000C	0xFFFB	0x000C	PERxR – 0x0008
4	0x0006	0xFFFD	0x0006	PERxR – 0x0004
≥ 5	0x0003	0xFFFD	0x0003	PERxR – 0x0002

注：如果比较值大于周期值，则不会生成比较匹配事件。

计数器工作模式（Counter operating mode）

定时器 0~7 可在连续模式（自由运行）下工作，也可以单次模式工作，此时会由复位事件触发开始计数，工作模式通过 HRPWM_PWMxCR0.CONT 设置。另外一个 HRPWM_PWMxCR0.RETRIG 位可用于选择单次模式时可重触发的还是不可重触发的。表 17-6 和图 17-4 以及图 17-5 总结了工作模式的详细信息。

表 17-7 定时器工作模式

CONT	RETRG	工作模式	启动/停止条件&计时/事件生成
0	0	单次不可重触发	将 CENx 位置 1 会使能定时器，但不会启动计数。第一个复位事件启动计数，随后直到计数达到 PER 值为止的任何复位事件都将被忽

			略。然后计数停止并生成 PER 事件。复位事件重新从 0x0000 开始启动计数。
0	1	单次可重触发	将 CENx 位置 1 会使能定时器，但不会启动计数。如果计数没有启动，则复位事件将启动计数，否则复位事件将计数清零。当计数达到 PER 值时计数停止并生成 PER 事件。复位事件重新从 0x0000 开始启动计数。
1	X	连续模式	将 CENx 位置 1 会使能定时器，并同时启动计数。当计数达到 PER 值时，它将翻转到 0x0000 并恢复计数。任何时刻都可以复位计数。

可以随时清零 CENx 位，以禁止定时器并停止计数。

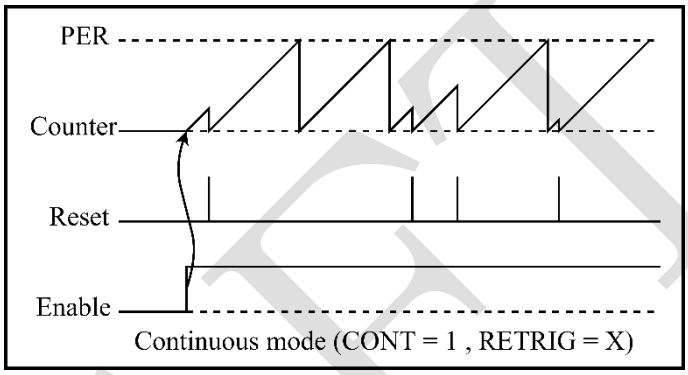


图 17-4　定时器连续工作模式

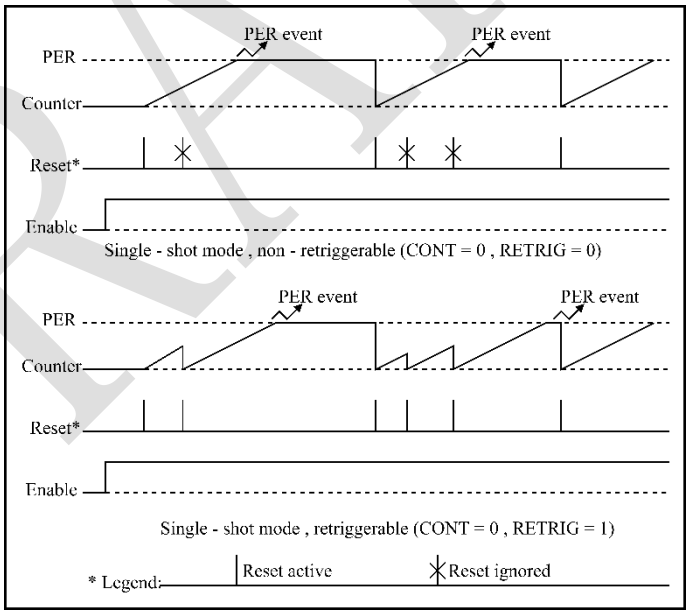


图 17-5　PWM 单次模式

翻转事件（Roll-over event）

在连续模式下，如果计数器在达到 HRPWM_PERxR 寄存器中设置的周期值后恢复为 0，则会生成计数器 roll-over 翻转事件。

该事件在 HRPWM 中有多种用途：

- 置位/复位输出

- 触发寄存器内容更新（从预加载寄存器加载到有效寄存器）
- 产生中断请求
- 产生 ADC 触发事件
- 使重复计数器递减

如果初始计数器值大于定时器启动时的周期值，或者在计数器已超过该值时设置了新周期值，计数器在连续模式下会复位、并且重新开始计数，在单次模式下会继续增加，将在达到最大周期值时溢出。

定时器复位（Timer reset）

定时单元计数器的复位可通过多达 38 个事件触发，这些事件可同时在 HRPWM_RSTxR 寄存器中选择，具体包括以下复位源：

- 定时单元：CMPB，CMPD 和寄存器更新（3 个事件）
- 主定时器：计数器周期事件，CMPA~CMPD（5 个事件）
- 外部事件：EXTEVNT0~9（10 个事件）
- 所有其他定时单元：CMPA、CMPB 和 CMPD（20 个事件）

可同时选择多个事件作为复位源。在这种情况下，会对多个复位请求进行或运算。如果在同一 FHRPWM 时钟周期内生成 2 个计数器复位事件，则会考虑后一个定时器复位事件。

此外，还可以使用 HRPWM_CR2.RSTx 对计数器执行软件复位。这些控制位分组到一个寄存器中，从而可同时复位多个计数器。

仅当相关计数器已使能（HRPWM_MCR1.CENx 置 1）后，才会考虑复位请求。

如果 FHRPWM 时钟预分频比大于 32 时，计数器复位事件会延迟到预分频时钟的下一有效边沿，这样可确保在输出跳变同步到复位事件（通常是恒定 Ton 时间转换器）时生成的波形无抖动。

图 17-6 显示了时钟预分频比为 128（FHRPWM 除以 4）时的复位处理方式。

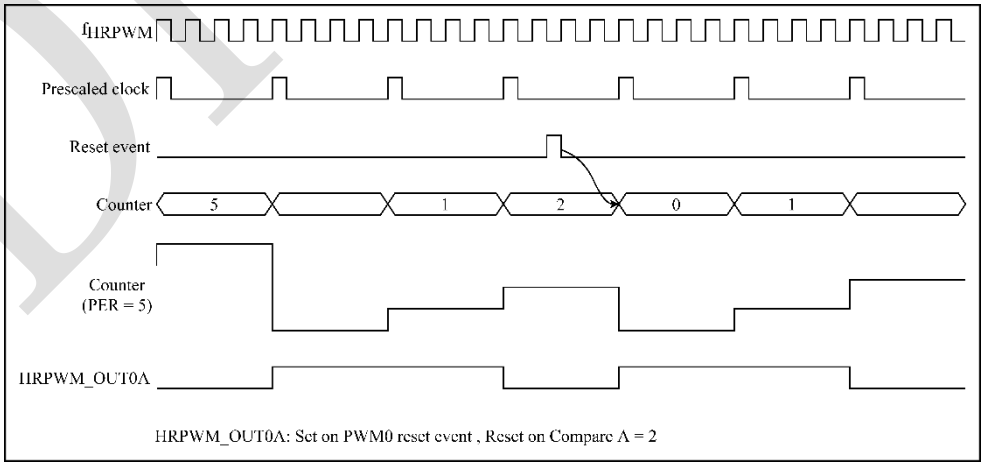


图 17-6　定时器复位重新同步（预分频比大于 32）

重复计数器（Repetition counter）

重复计数器的主要用途是通过分离开关频率和中断频率，来调整周期中断频率并减轻 CPU 的

负荷。

定时单元包含重复计数器。该计数器无法读取，只能使用 HRPWM_CNTxR 寄存器进行编程，计数器可以自动重载。

定时器使能后（HRPWM_MCR1.CENx 置 1），重复计数器会初始化为 HRPWM_CNTxR 寄存器的内容。定时器使能后，每次计数器由于复位事件或计数器翻转而清零时，重复计数器都会减 1。当重复计数达到 0 后，会发出 REP 中断请求（若使能 HRPWM_PWMxDIER.REPIE）。

如果 HRPWM_CNTxR 寄存器设为 0，会在每个周期都产生中断。如果值大于 0，则会在（HRPWM_CNTxR+ 1）个周期后产生 REP 中断。图 17-7 显示了连续模式下重复计数器取不同值时的操作。

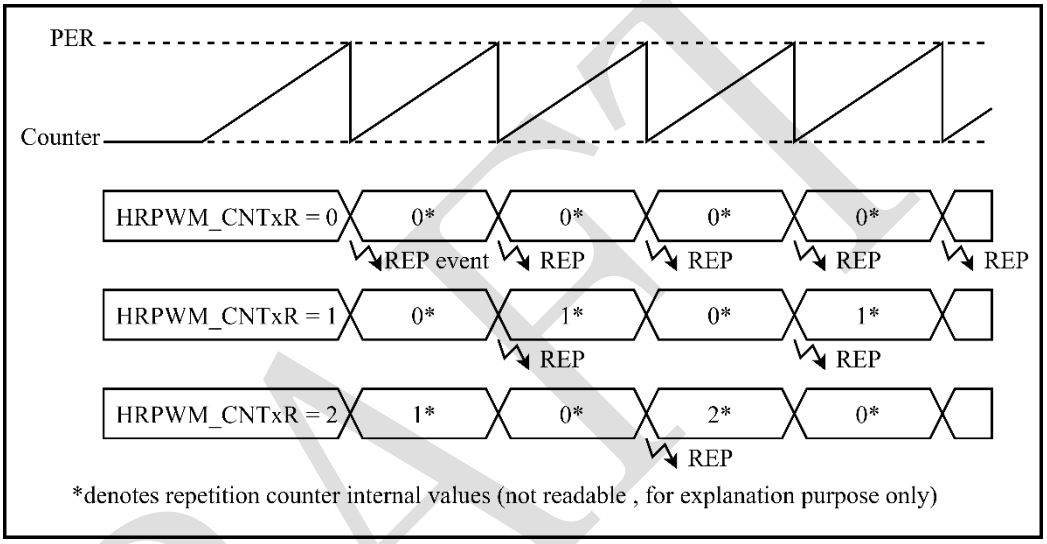


图 17-7　连续模式下的重复事件

无论在连续模式还是单次模式下，如果计数器在达到周期值（可变频率操作）之前复位，则也可使用重复计数器（如图 17-8 所示）。复位会使重复计数器在计数器使能后（HRPWM_MCR1.CENx 置 1）执行第一次启动时递减。

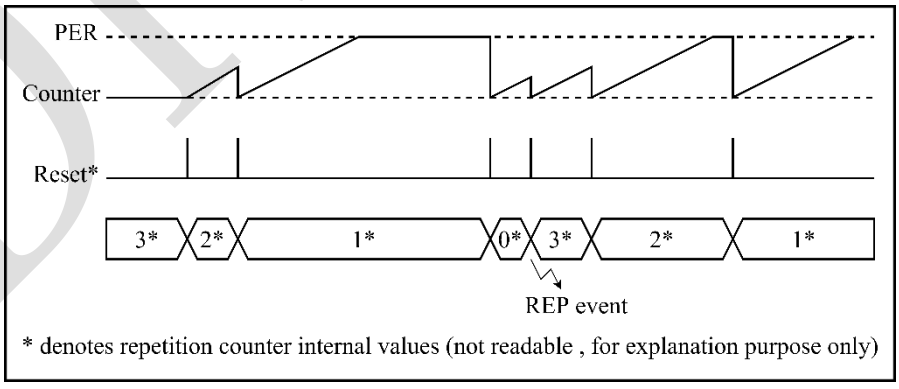


图 17-8　单次模式下的周期重复事件

来自 hrpwm_sync_in 源的复位或启动事件会像其他任何复位事件一样使重复计数器递减。但在通过 SYNCIN 启动的单次模式下（HRPWM_PWMxCR0.SYNCSTRT 置 1），重复计数器仅会在周期后出现第一个复位事件时递减。任何后续的复位事件都不会更改重复计数器的值，直至计数器通过新的 hrpwm_sync_in 输入请求重启。

置位/复位交错配置矩阵（Set / Clear crossbar）

置位事件相当于输出从无效状态跳变为有效状态，而复位事件相当于输出从有效状态转换到无效状态。

波形的极性在输出级中定义，以适应正逻辑或负逻辑外部组件：对于正极性（HRPWM_OUTxR.POLx = 0），有效电平对应于逻辑电平 1，而对于负极性（HRPWM_OUTxR.POLx = 1），有效电平对应于逻辑电平 0。

每个定时单元都会控制两路输出的置位/复位交错配置矩阵，这两路输出可通过多达 19 种事件置位、复位或翻转，这些事件可从以下源中选择：

- 定时单元：周期，CMPA~CMPD，寄存器更新（6 个事件）
- 主定时器：周期，CMPA~CMPD，HRPWM 同步（6 个事件）
- 外部事件：HRPWM_Event0~9（10 个事件）
- 软件强制（1 个事件）

注意：在 Updown 模式（UDM 位设置为 1）中，计数周期事件是根据 HRPWM_PWMxCR1.OUTROM 的设置生成的。

事件源会进行或运算，可同时选择多个事件。

每路输出均由两个 40 位寄存器控制，一个寄存器用于输出置位（HRPWM_SETxyR），一个寄存器用于输出复位（HRPWM_CLRxyR），其中 x 表示定时单元 0~7，y 表示输出 A 或 B（例如 HRPWM_SET1AR，HRPWM_CLR1AR ...）。

如果为置位和复位选择了相同事件，则会翻转输出状态。每个 THRPWM 周期输出状态的翻转次数不能超过 1 次：如果同一周期中有两个连续的翻转事件，则仅会考虑第一个翻转事件。

仅当计数器使能后（HRPWM_MCR1.CENx 置 1），才会考虑置位和复位请求，但软件在定时器启动时强制请求允许预置输出的情况除外。

使用两个比较事件生成 PWM 波形的示例如图 17-9 所示。

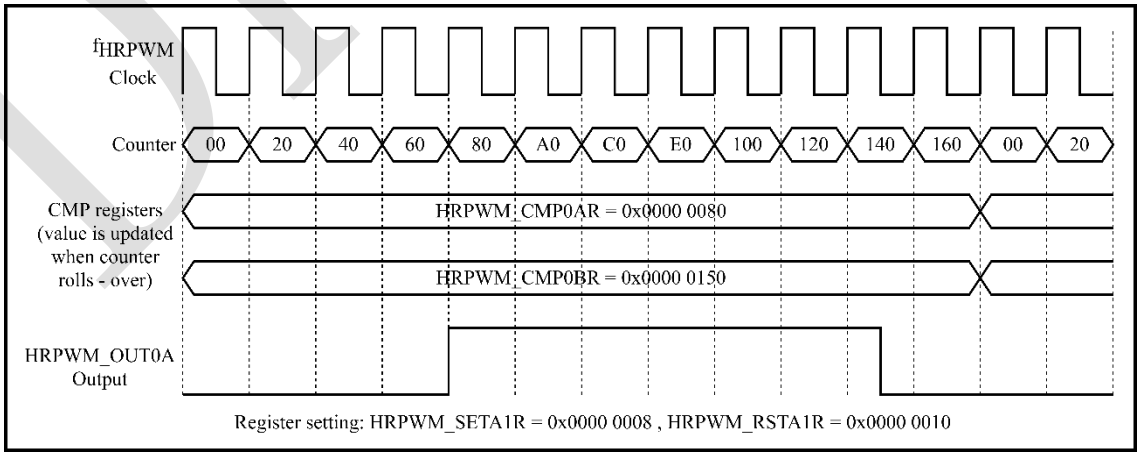


图 17-9　输出因比较事件而动作：CMPA 置位，CMPB 复位

更新事件置位/复位输出（Set / reset on update events）

更新事件可以将输出置位/复位，不过置位/复位事件是在低分辨率下完成的。当HRPWM_PWMxCR0.CKPSC<5 时，高分辨率延迟被设置为其最大值，因此与其他比较置位/复位事件相比，更新时的置位/复位事件总是滞后，抖动在 0 到 31/32 个 FHRPWM 时钟周期的时间之间变化。

半模式（Half mode）

此模式用于生成占空比固定 50%、频率可变的方波信号（通常用于使用谐振拓扑的转换器）。允许在设定新周期值时自动将占空比强制设为周期值的一半。

要使能此模式，应向 HRPWM_PWMxCR0.HALF 写入 1。HRPWM_PERxR 寄存器写入数值后，会自动将 CMPA 值更新为 HRPWM_PERxR/2 值。

生成方波的输出必须编程为在发生比较 A 事件时进行一次跳变，在发生周期事件进行一次跳变，具体如下：

- HRPWM_SETxyR = 0x0000 0008，HRPWM_CLRxyR = 0x0000 0004
- HRPWM_SETxyR = 0x0000 0004，HRPWM_CLRxyR = 0x0000 0008

半模式会覆盖 HRPWM_CMPAxR 寄存器的内容。访问 HRPWM_PERxR 寄存器不仅会更新比较 A 内部寄存器。用户可访问的 HRPWM_CMPAxR 寄存器也会更新为 HRPWM_PERxR/2 值。

当使能预加载功能（HRPWM_PWMxCR0.PREEN = 1）时，则会在发生更新事件时刷新 HRPWM_CMPA 有效寄存器。如果禁止预加载功能（HRPWM_PWMxCR0.PREEN = 0），则在 HRPWM_PERxR 写入数值后，HRPWM_CMPA 有效寄存器会立即更新。

如果使能了半模式，周期必须大于或等于 6 个 FHRPWM 时钟周期（如果 HRPWM_PWMxCR0.CKPSC = 0，则为 0xC0；如果 HRPWM_PWMxCR0.CKPSC = 1，则为 0x60；如果 HRPWM_PWMxCR0.CKPSC = 2，则为 0x30...）。

交错模式（Interleaved mode）

交错模式补充完善了半模式，可以帮助实现一些交错的拓扑。

当 HRPWM_PERxR 寄存器更新时，会自动重新计算各个比较寄存器的内容。

使用 HRPWM_MCR0.HALF 和 HRPWM_PWMxCR0.HALF 和 HRPWM_MCR0. INTLVD 和 HRPWM_PWMxCR0.INTLVD 使能交错模式，如下表 17-8 所示。

表 17-8　交错模式选择

HALF 位	INTLVD [1:0] 位	交错模式
0	00	无影响
1	00	双相交错 180°
0	01	三相交错 120°
0	10	四相交错 90°

表 17-9 给出了三种交错模式下的比较值。交错模式下相应比较寄存器的内容会被覆盖。相应的比较事件可用于触发输出置位/复位或复位定时器。

表 17-9　交错模式下的 HRPWM_CMPxAR~HRPWM_CMPxCR 值

模式	双相交错 180°	三相交错 120°	四相交错 90°
HRPWM_CMPxAR	HRPWM_PERxR/2	HRPWM_PERxR/3	HRPWM_PERxR/4
HRPWM_CMPxBR	无影响	2x（HRPWM_PERxR/3）	HRPWM_PERxR/2
HRPWM_CMPxCR	无影响	无影响	3x（HRPWM_PERxR/4）

注：在半模式和交错模式下，比较值寄存器由硬件控制，寄存器写入无效。不过写入寄存器的值存储在预加载寄存器中，在退出这些模式后下个更新事件后生效。

交换模式（Swap mode）

交换模式允许配置一个寄存器位交换两个输出：输出 A 信号连接到输出 B 引脚，输出 B 信号连接到输出 A 引脚。输出交换由 HRPWM_CR1.SWPx 配置，并在下一个更新事件处生效。

输出在进入置位/复位交错配置矩阵之前已经发生交换，具体如下：

- 如果 SWPx = 0，HRPWM_SETxAR 和 HRPWM_CLRxAR 编码控制输出 A，HRPWM_SETxBR 和 HRPWM_CLRxBR 编码控制输出 B
- 如果 SWPx = 1，HRPWM_SETxAR 和 HRPWM_CLRxAR 编码控制输出 B，HRPWM_SETxBR 和 HRPWM_CLRxBR 编码控制输出 A

交换模式只影响预加载寄存器，而不影响有效寄存器。

注：使用交换模式时，必须使能预加载模式。

交换模式不会修改 HRPWM_SETxy 和 HRPWM_RSTxy 状态标志，以及相应的中断请求。例如，当 SWPx = 0 时，HRPWM_SETxA 标志与输出 A 相关；当 SWPx = 1 时，HRPWM_SETxA 标志与输出 B 相关。

类似地，交换模式不会更改 HRPWM_OUTxR 寄存器中的控制位（HRPWM_OUTxR.CHPx，HRPWM_OUTxR.FAULTx，HRPWM_OUTxR.IDLESx，HRPWM_OUTxR.POLx）的属性。例如，无论 SWP 位的值如何，POLA 位都控制输出 A 的极性。

注：在推挽模式下（HRPWM_PWMxCR0.PSHPLL = 1），SWPx 位将被忽略。

捕获（Capture）

定时单元能够在内部和外部事件的触发下捕获计数值。捕获的目的是：

- 测量事件到达时间或发生间隔。
- 自动延迟模式下更新比较 B 和比较 D 值。

捕获以 FHRPWM 分辨率执行。

定时单元包含 2 个捕获寄存器： HRPWM_CAPAxR 和 HRPWM_CAPBxR。捕获触发事件在 HRPWM_CAPAxCR/HRPWM_CAPAxCER 和 HRPWM_CAPBxCR/HRPWM_CAPBxCER 寄存器中编程。

定时单元计数器的捕获可通过多达 40 种事件触发，这些事件可同时在 HRPWM_CAPAxCR/HRPWM_CAPAxCER 和 HRPWM_CAPBxCR/HRPWM_CAPBxCER 寄存

器中选择，具体包括以下触发源：

- 外部事件，HRPWM_Event0~9（10 个事件）
- 其他定时单元：比较 A、B 和输出 A 置位/复位事件（28 个事件）
- 自身定时单元：更新（1 个事件）
- 软件捕获（1 个事件）

可以同时选择多个事件处理多个捕获触发信号。在这种情况下，会对多个并发触发请求进行或运算。如果 HRPWM_PWMxDIER.CAPxIE 和 HRPWM_PWMxDIER.CAPxDE 已置 1，捕获可生成中断或 DMA 请求。

电路未采用避免重复捕获的机制：即使前一个捕获值未读取、或者捕获标志未清零，也会触发新捕获。

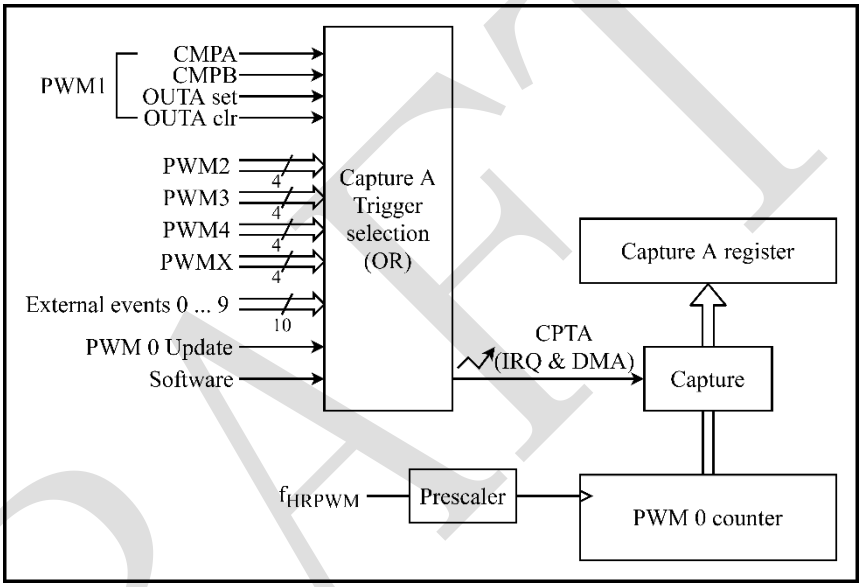


图 17-10　PWM0 定时单元捕获电路

自动延迟模式（Auto-delayed mode）

自动延迟模式可以相对于捕获事件生成比较事件，这样便可在捕获后的设定时间进行输出更改等操作。在这种情况下，比较匹配独立于定时器计数值进行。这样可生成时序与外部事件同步的波形，而无需进行软件计算和中断维护。

只要没有捕获事件被触发，便会忽略 HRPWM_CMPxR 寄存器的内容（计数值与比较值相匹配的情况下，不会生成比较事件）。一旦触发了捕获事件，则会对在 HRPWM_CMPxR 中编程的比较值与 HRPWM_CAPxyR 中捕获的计数值求和，并会用得出的结果更新内部自动延迟比较计数器。自动延迟比较寄存器属于定时单元内部的寄存器，不能读取。

HRPWM_CMPxR 预加载寄存器的内容在计算后不会修改。

自动延迟模式仅适用于比较 B 和比较 D 寄存器。比较 B 寄存器与捕获 A 相关联，而比较 D 寄存器与捕获 B 相关联。与常规模式一样， HRPWM_CMPBxR 和 HRPWM_CMPDxR 比较寄存器不能编程为小于 3 个 FHRPWM 时钟周期的值。

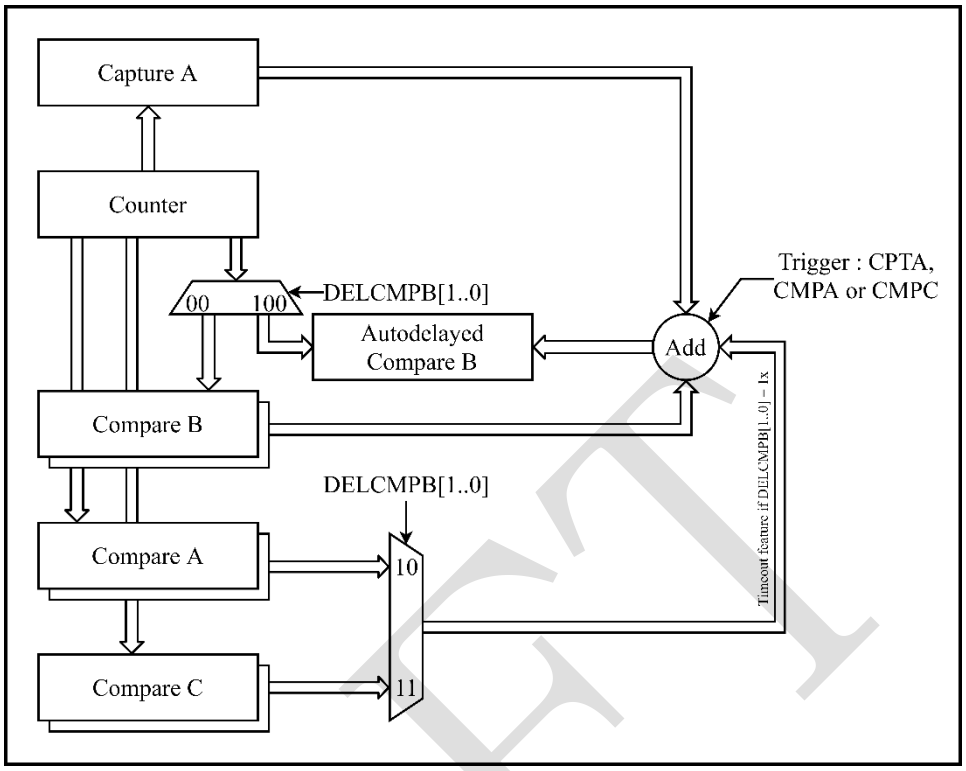


图 17-11 自动延迟模式原理图（仅显示 CMPB）

自动延迟比较寄存器的有效期从捕获开始，到周期事件为止：计数器达到周期值后，系统会重新设置，比较寄存器在发生捕获之前会处于禁用状态。

HRPWM_PWMxCR0.DELCMPB 和 HRPWM_PWMxCR0.DELCMPD 可配置自动延迟模式，具体如下：

- 00 常规比较模式：会直接将 HRPWM_CMPBxR 和 HRPWM_CMPDxR 寄存器的内容与计数值进行比较。
- 01 自动延迟模式：会重新计算比较 B 和比较 D 寄存器值，并在捕获 A/B 事件后用计算值与计数值进行比较。
- 1X 具有超时的自动延迟模式：会重新计算比较 B 和比较 D 寄存器值，并在捕获 A/B 事件后用计算值与计数器值进行比较；如果缺少捕获 A/B 事件，则在比较 A 匹配（HRPWM_PWMxCR0.DELCMPx = 10）或比较 C 匹配（HRPWM_PWMxCR0.DELCMPx = 11）后用计算值与计数值进行比较，以便实现超时功能。

进行捕获时，会与 (HRPWM_CMPB/DxR + HRPWM_CAPA/BxR)值进行比较。如果周期内未触发捕获，行为将取决于 HRPWM_PWMxCR0.DELCMPx 值：

- HRPWM_PWMxCR0.DELCMPx = 01：未生成比较事件
- HRPWM_PWMxCR0.DELCMPx = 10 或 11：
  与 2 个比较寄存器值之和进行比较（例如 HRPWM_CMPBxR + HRPWM_CMPAxR）。如果捕获是在 HRPWM_CMPx+HRPWM_CMPA（或 HRPWM_CMPx+HRPWM_CMPC）后触发的，则不会考虑捕获。

下一 PWM 周期开始时会再次使能捕获。

如果自动延迟求和的结果大于 0xFFFF（溢出），则会忽略该值，并且新周期开始之前不会生成比较事件。

注：如果从一个值重新编程为另一个值，为了正确地重新初始化自动延迟机制，必须复位 HRPWM_PWMxCR0.DELCMPx 位域，例如：

- HRPWM_PWMxCR0.DELCMPx = 10
- HRPWM_PWMxCR0.DELCMPx = 00
- HRPWM_PWMxCR0.DELCMPx = 11

下图举例说明了以下信号是怎样生成的：

- 计数值等于比较 A 值时，输出置位
- 给定外部事件下降沿 4 个周期后，输出复位

为了简化示意图，图中所示的外部事件信号不含任何重新同步延迟：实际情况下，由于需要通过内部重新同步阶段来处理外部输入信号，因此下降沿与捕获事件之间会有 2 到 3 个 FHRPWM 时钟周期的延迟。

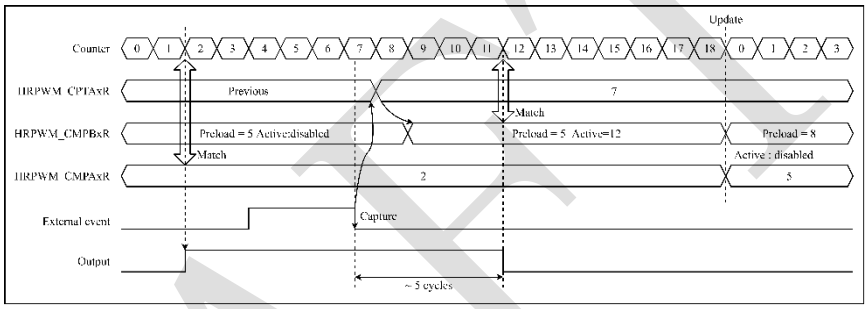


图 17-12 自动延迟模式比较

使用常规比较通道（例如比较 A）进行输出置位：计数器与比较寄存器的内容匹配后，输出会立即变为有效状态。

使用延迟比较进行输出复位：仅当发生了捕获事件时，才会生成比较事件。如果计数器与延迟比较值（计数值 =4）匹配，则不会生成事件。捕获事件由外部事件触发后，捕获寄存器的内容会立即与延迟比较值相加，得出新的比较值。示例中，自动延迟值 4 与捕获值 7 相加，得出自动延迟比较寄存器中的值 11。从此时起，可生成比较事件，并会在计数器等于 11 时生成，比较事件会使输出复位。

自动延迟模式下的重复捕获管理

使能自动延迟模式时（HRPWM_PWMxCR0.DELCMPx = 01、10、11）将阻止重复捕获。

如果同一计数周期内出现多个捕获请求，只会考虑使用第一个捕获请求来计算自动延迟比较值。仅可在以下条件下进行新捕获：

- 自动延迟比较具有匹配的计数器值（比较事件）
- 计数器已翻转（周期）
- 定时器已复位

更改自动延迟比较值

如果已预加载自动延迟比较值（HRPWM_PWMxCR0.PREEN 置 1），则无论比较寄存器何时写入数值、是否发生了捕获事件，都会在发生下一更新事件（例如周期事件）时更新新的比较值。如图 190 所示，其中延迟是在计数器翻转时更改的。

触发半模式（Triggered-half mode）

触发半模式的目的是使得两个支持不同频率操作且需要 180°相移的交错转换器同步，通过主机-从机方式实现。从机的同步是根据主机上一个开关周期来连续进行调节。

触发半模式是使用捕获单元完成的，主机的开关周期被捕获除以 2，然后由硬件存储在比较 B 寄存器中。比较 B 寄存器包含一个等于捕获周期的一半的值，也就是主机开关周期的一半。比较 B 事件可以用来触发下一个控制从机的定时单元。

触发半模式通过设置 HRPWM_PWMxCR0.TRGHLF 来使能，该位在 PWM 运行（HRPWM_MCR1.CENx 置 1）之后不能更改。

触发半触发模式不能与使用 CMPB 的其他模式同时使用，例如双通道 DAC 触发/交错模式/均衡空闲模式。

CMPB 初始值可以由用户写入，但在第一次捕获之后会被忽略。当 TRGHLF 位置 0 时，CMPB 的预加载机制不生效。

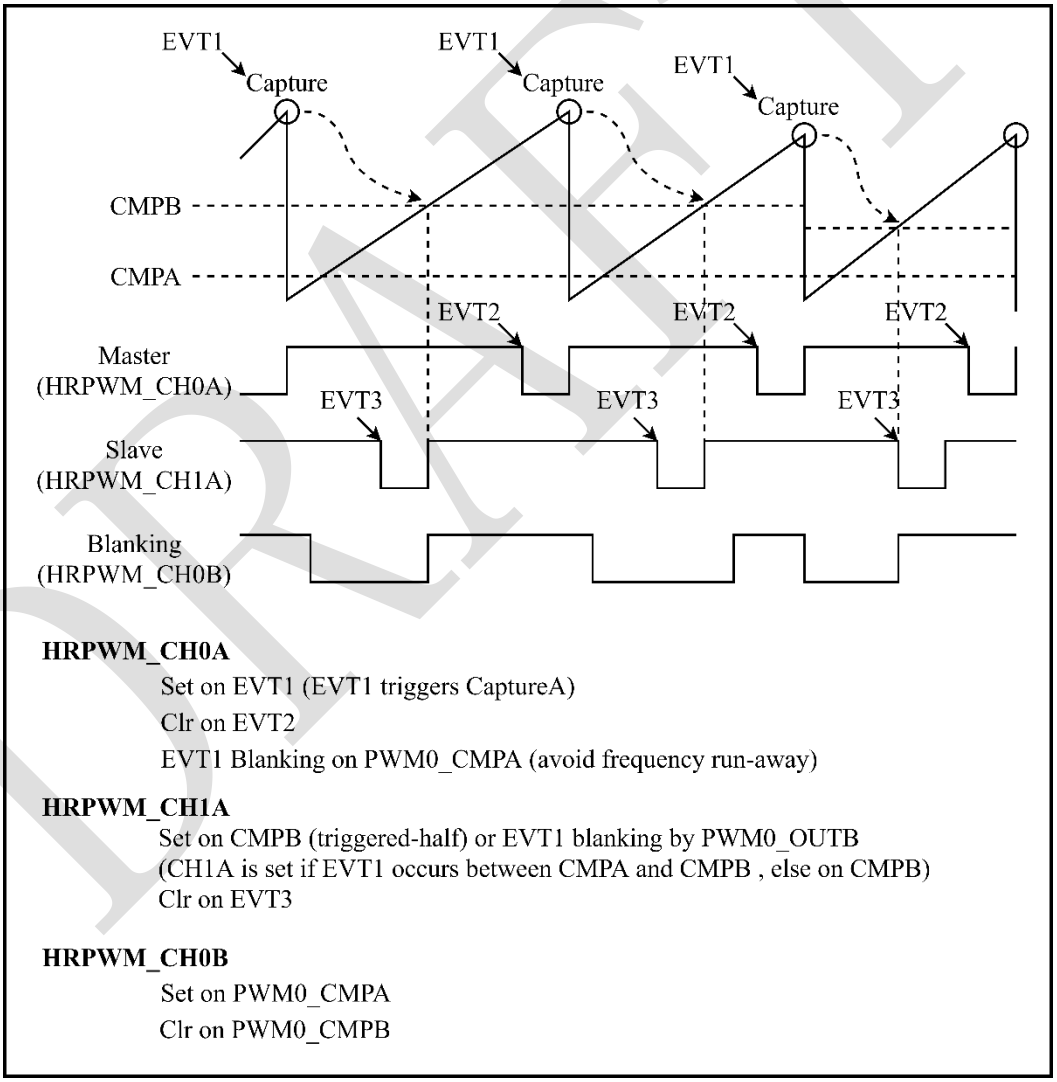


图 17-13 触发半模式示例

注:在触发半模式下，比较 B 寄存器由硬件控制，写入寄存器无效果。写入的寄存器值存储在预加载寄存器中，并在退出模式后下一个更新事件处生效。

推挽模式（Push-pull mode）

该模式的主要目的是使用推挽拓扑驱动转换器。

通过将 HRPWM_PWMxCR0.PSHPLL 置 1，即可使能推挽模式。

在该模式下，会按照周期将交错配置矩阵生成的信号交替地施加到输出 A 和输出 B 上（如果信号施加到输出 A 上，则输出 B 保持在无效状态，反之亦然）。重定向速率（推挽频率）由定时器的周期事件定义，如图 17-14 所示。推挽周期是定时器计数周期的两倍。

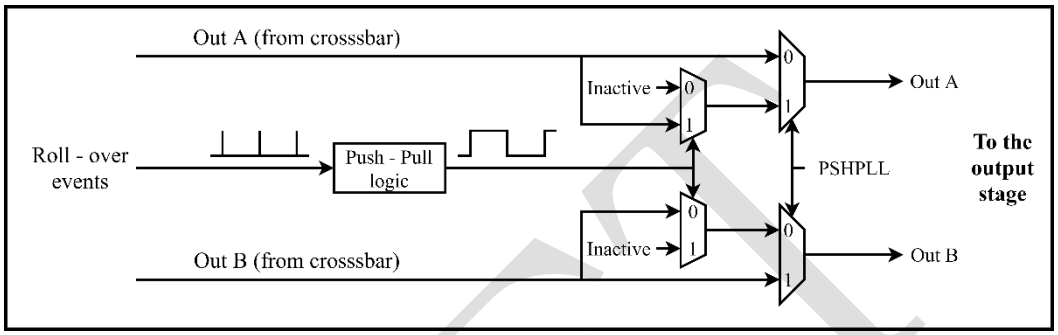


图 17-14　推挽模式框图

定时器在连续模式和单次模式（可重触发/不可重触发）下工作时，可以使用推挽模式；需要禁止定时器以停止推挽操作，并且重新使能推挽操作之前需要将计数器复位。

两路输出的信号波形由 HRPWM_SETxyR 和 HRPWM_CLRxyR 决定。如果希望两路输出的波形完全相同，并实现平衡的操作，需要配置 HRPWM_SETxAR = HRPWM_SETxBR 和 HRPWM_CLRxAR = HRPWM_CLRxBR。不过，仍可对两路输出进行不同的编程，以实现其他用途。

在图 17-15 中提供的示例中，定时器内部波形的定义如下：

- 发生周期事件时，输出置位
- 发生 CMPA 匹配事件时，输出复位

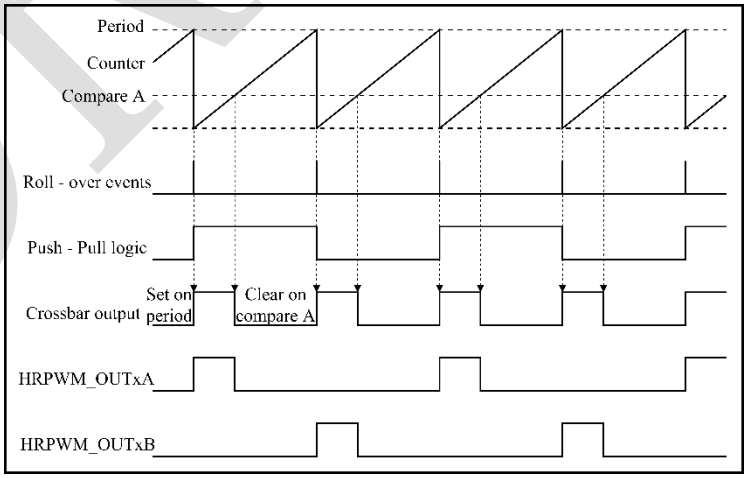


图 17-15　推挽模式示例

推挽模式下的死区插入机制如图 17-16 所示，这里包含了正向死区和负向死区。在这种情况下，输出不再是互补的，两路输出会独立地插入死区时间（交错配置矩阵的 OUTA 和 OUTB 都有效）。

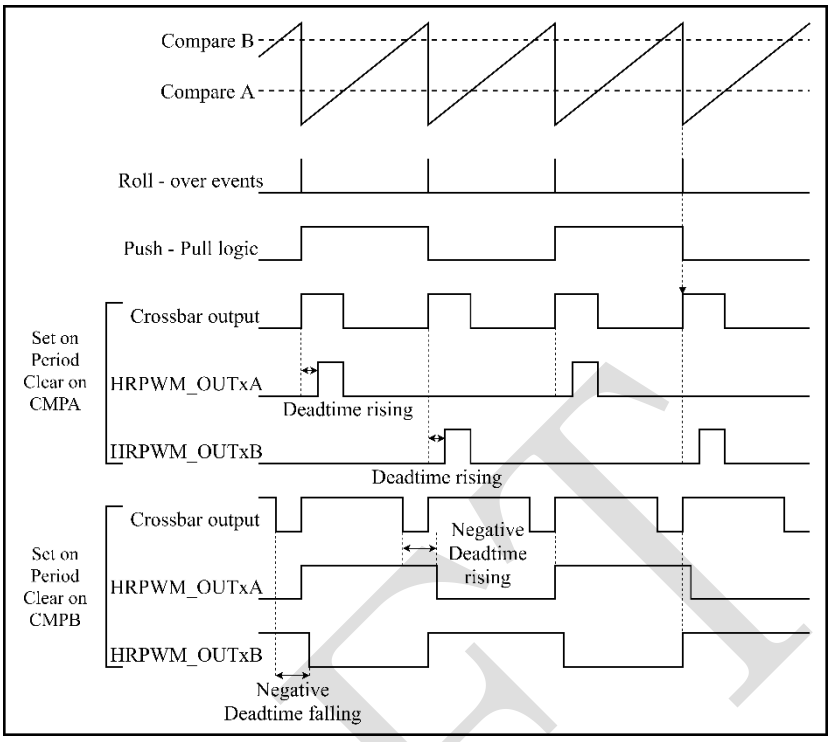


图 17-16　带有死区时间的 PWM 推挽模式

死区（Deadtime）

死区插入单元可通过单个参考波形生成一对互补信号，并且有效状态跳变之间的延迟可编程。这通常用于使用半桥或全桥的拓扑，可简化软件操作流程：只需要对一个波形进行编程和控制即可驱动两路输出。

死区插入通过将 HRPWM_OUTxR.DTEN 置 1 来使能。互补信号基于输出 A 定义的参考波形构建而成，使用 HRPWM_SETxAR 和 HRPWM_CLRxAR 寄存器；如果 HRPWM_OUTxR.DTEN 置 1，则 HRPWM_SETxBR 和 HRPWM_CLRxBR 寄存器无意义。

可按照与参考波形上升沿和下降沿的关系定义两个死区，如图 17-17 所示。

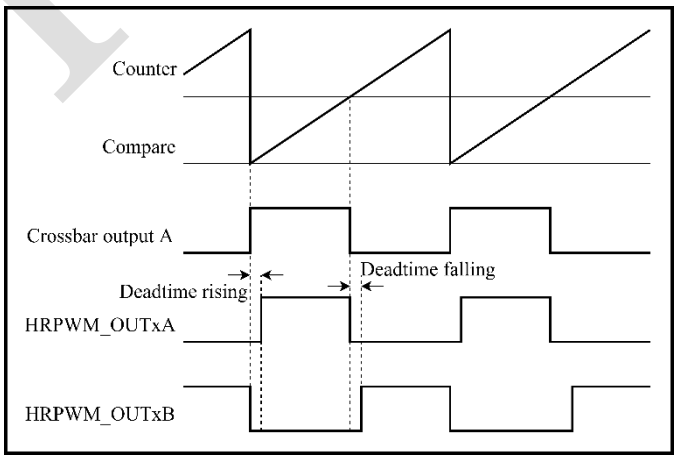


图 17-17　已插入死区的互补输出

如果需要使用一些控制重叠，可定义负死区，此时要使用死区符号位（HRPWM_DTxR.SDTF 和 HRPWM_DTxR.SDTR）。

图 17-18 显示了各符号位对应的互补信号波形。

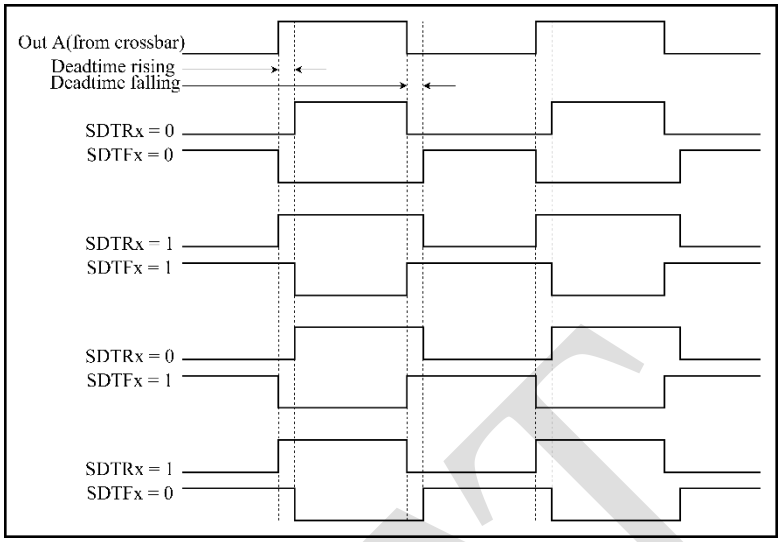


图 17-18　死区插入与死区符号位（符号位为 1 表示负死区）

死区值使用 HRPWM_DTxR.DTF 和 HRPWM_DTxR.DTR 定义的，基于根据 HRPWM_PWMxCR0.CKPSC 预分频的特定时钟，具体如下：

$$t_{DTx} = +/- DTx[11:0] \times t_{DTG}$$

其中 x 为 R 或 F，t_{DTG} =（$2^{(CKPSC[2:0])}$）x（THRPWM）。

表 17-10 给出了根据预分频值得出的分辨率和最大绝对值。

表 17-10　死区分辨率和最大死区时间

CKPSC[2:0]	t_{DTG}	T_{DTx} max	FHRPWM = 180 MHz	
			t_{DTG}	$\|t_{DTx}\|$ max
000	**THRPWM / 32**	**4095 * t_{DTG}**	173 ps	0.71 µs
001	**THRPWM / 16**		347 ps	1.42 µs
010	**THRPWM / 8**		694 ps	2.84 µs
011	**THRPWM / 4**		1.39 ns	5.69 µs
100	**THRPWM / 2**		2.77 ns	11.38 µs
101	**THRPWM**		5.56 ns	22.75 µs
110	**2 * THRPWM**		11.1 ns	45.50 µs
111	**4 *THRPWM**		22.2 ns	91.0 µs

图 17-19 到图 17-22 显示了在各个死区配置下对脉冲宽度小于死区值的参考波形的处理方式。

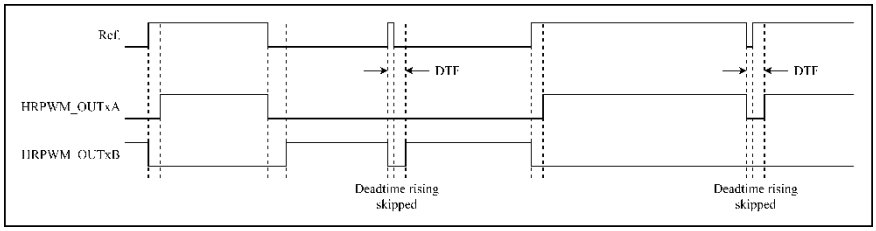


图 17-19 窄脉冲宽度下的互补输出（SDTRx = SDTFx = 0）

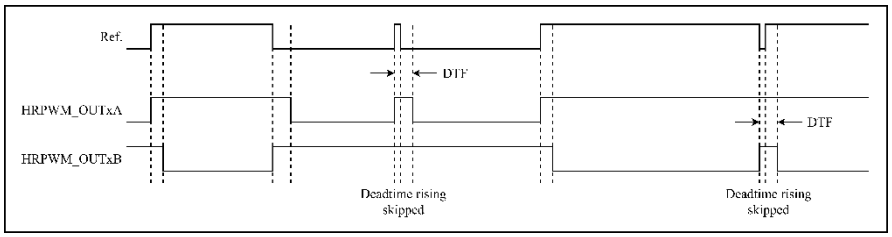


图 17-20　窄脉冲宽度下的互补输出（SDTRx = SDTFx = 1）

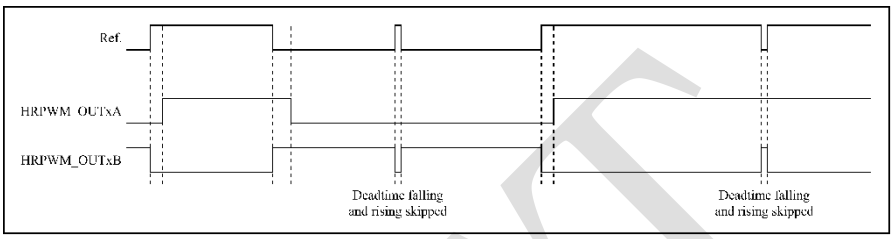


图 17-21　窄脉冲宽度下的互补输出（SDTRx = 0, SDTFx = 1）

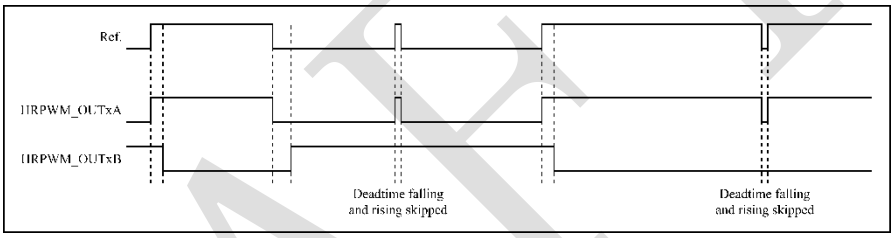


图 17-22　窄脉冲宽度下的互补输出（SDTRx = 1, SDTFx = 0）

注：以下情况下，不得更改 HRPWM_OUTxR.DTEN 位：

- *定时器使能时（HRPWM_MCR1.CENx 置 1）*
- *定时器输出由另一定时器置位/复位时（HRPWM_MCR1.CENx 复位时）*

否则会引发不可预测的行为。

因此，需要禁止定时器（HRPWM_MCR1.CENx 复位）并禁止相应的输出。

17.4.5　主定时器

主定时器的主要用途是向 8 个定时单元提供公共信号，以便进行同步或置位/复位输出。主定时器不会直接控制任何输出，但可以间接地控制输出，由置位/复位交错配置矩阵实现。

图 17-23 给出了主定时器的概览。

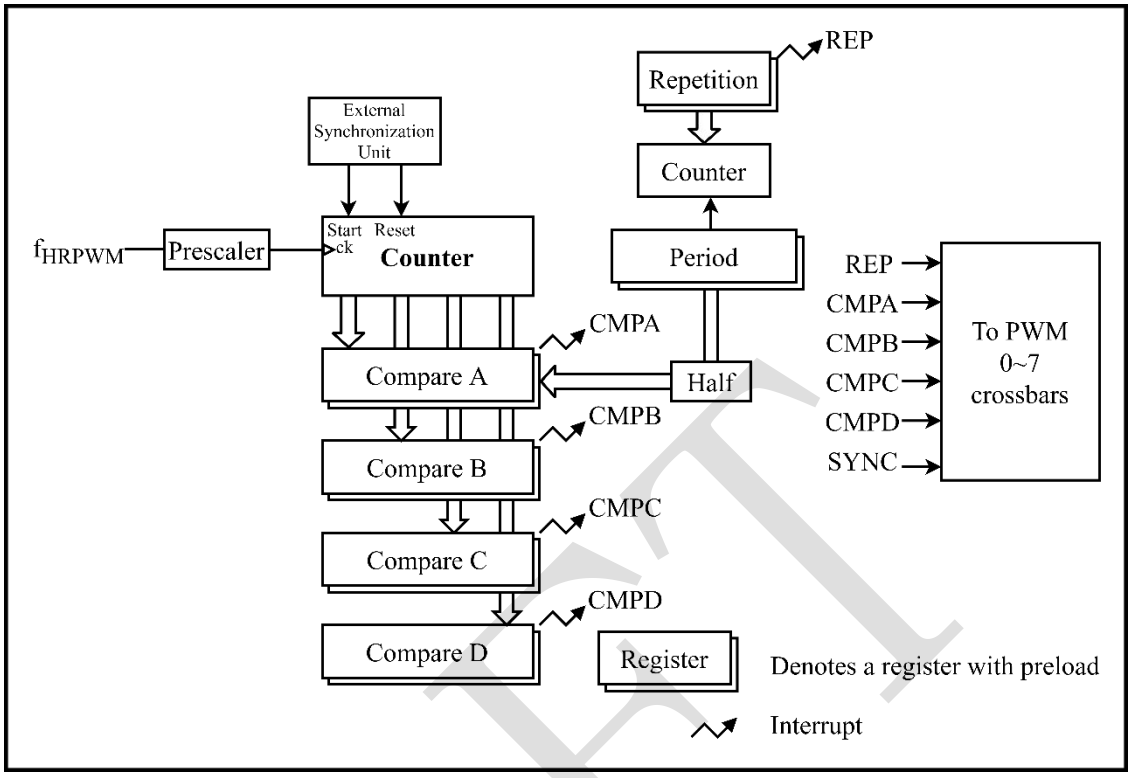


图 17-23　主定时器概览

主定时器采用的架构与定时单元非常相似，二者的不同之处如下：

- 主定时器未关联输出，也没有输出相关的控制
- 主定时器没有自己的交错配置矩阵，也没有推挽或死区模式
- 主定时器只能通过软件或者外部同步电路复位
- 主定时器不包含外部事件消隐和开窗电路
- 主定时器中断请求数量有限：比较 A~比较 D、周期事件、重复事件、寄存器更新事件和同步事件。

主定时器控制寄存器包含主定时器和定时单元 0~7 的所有定时器使能位。这样可通过单次写访问同时启动所有定时器。

主定时器还会利用 MCU 内部和外部（输入/输出）资源，处理整个 HRPWM 定时器的外部同步。

主定时器控制寄存器的映射偏移与定时单元寄存器的偏移相同。

17.4.6　上-下计数模式

HRPWM 默认的计数模式为上计数模式（递增），然后 HRPWM 同样支持上-下计数模式（递增递减），也称为中心对齐模式。

通过 HRPWM_PWMxCR1.UDM 位使能上-下计数模式。一旦定时器开始工作（HRPWM_MCR1.CENx 置 1），此位不可以被更改。上-下计数模式只适用于定时器 x，主定时器仅在上计数模式下工作。

本节详细介绍了上-下计数模式与上计数模式的功能差异。

在上-下计数模式下，HRPWM_PERxR 中的周期值必须开启预加载功能（或保持为静态值）。周期值只能在周期事件或者计数复位的情况下更新：

置位/复位交错配置矩阵的不同之处如下：

来自定时单元的事件对输出置位/复位的效果与计数的上-下方向有关：

- 如果事件在 HRPWM_SETxyR 寄存器中使能，上计数期间的事件将输出置位，下计数期间的事件将输出复位。
- 如果事件在 HRPWM_CLRxyR 寄存器中使能，上计数期间的事件将输出复位，下计数期间的事件将输出置位。
- 如果事件在 HRPWM_SETxyR 和 HRPWM_CLRxyR 寄存器中同时使能，事件将输出翻转。

以上机制适用于：

- 定时单元：周期，比较 A~比较 D、寄存器更新（6 个事件）
- 主定时器：周期，比较 A~比较 D、HRPWM 同步（6 个事件）

图 17-24 显示了如何生成基本的波形。

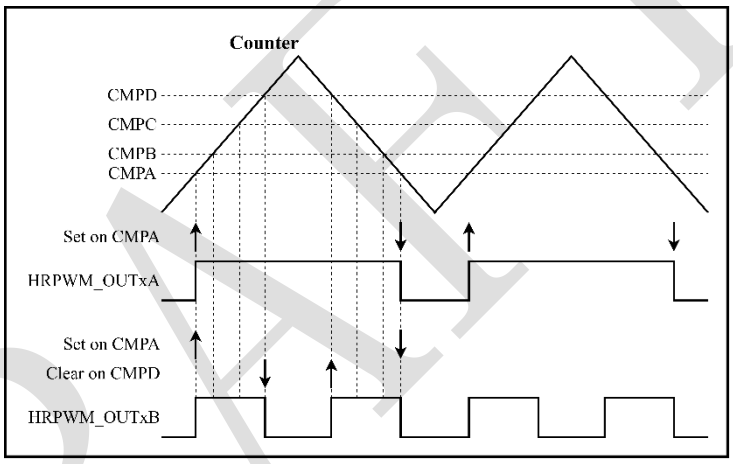


图 17-24　上-下模式下的简单对称波形

图 17-25 显示了如何生成一些更复杂的波形，使用了 4 个比较单元以及输出翻转模式。

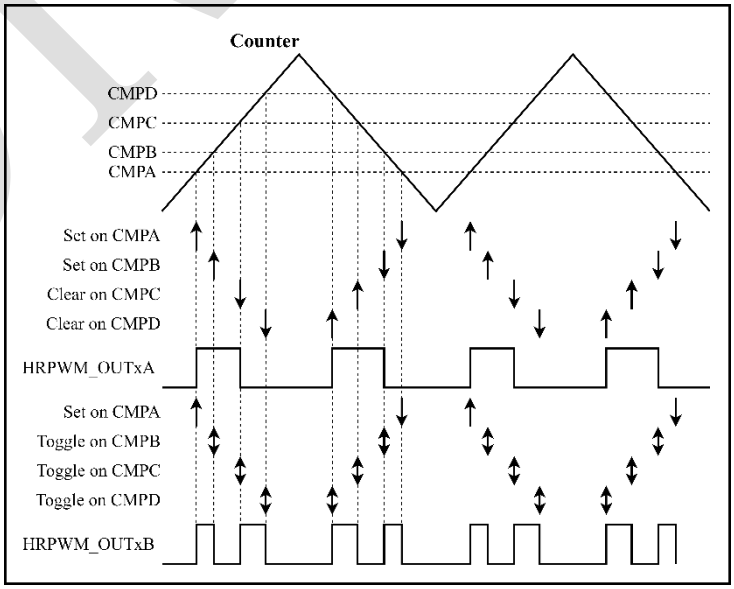


图 17-25　上-下计数模式下的复杂对称波形

图 17-26 显示了如何生成一个非对称的波形。在这种情形下，需要注意 CMPB 值必须大于 CMPA 值才能确保波形不对称。

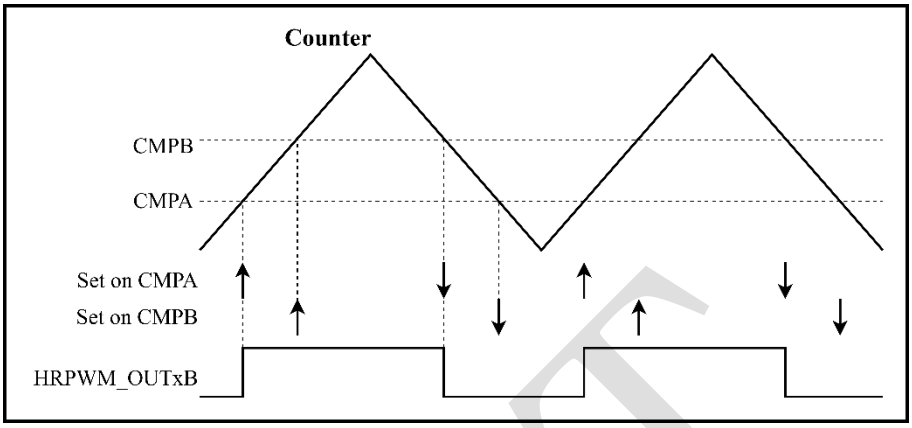


图 17-26　上-下计数模式下的非对称波形

注：对于非对称的情形，需要满足 CMPB> CMPA。

软件强制位和外部事件 Event 0~9 的行为在上计数和上-下计数模式下是相同的。图 17-27 显示了脉冲宽度可以通过外部事件进行缩短。

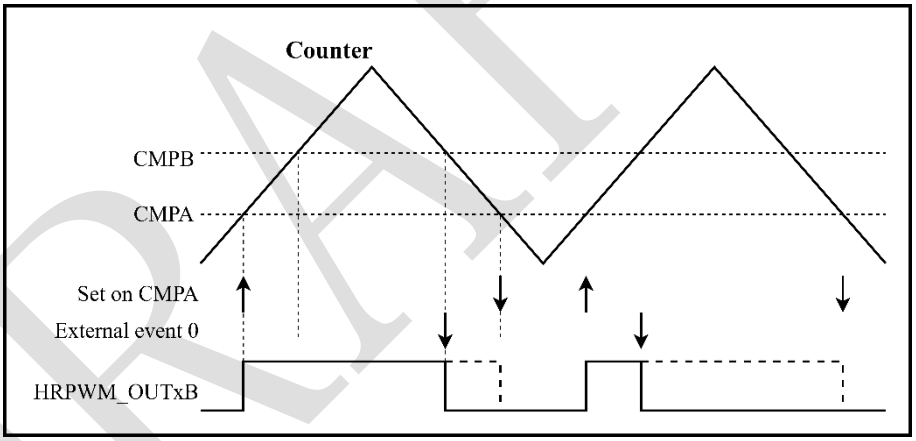


图 17-27　上-下计数模式下的外部事件处理

上-下计数模式适用于连续模式和单次（可重触发与不可重触发）工作模式。复位请求会导致计数器从 0 重新启动。图 17-28、图 17-29 显示了定时器 1 在单次可重触发模式下的计数器行为。

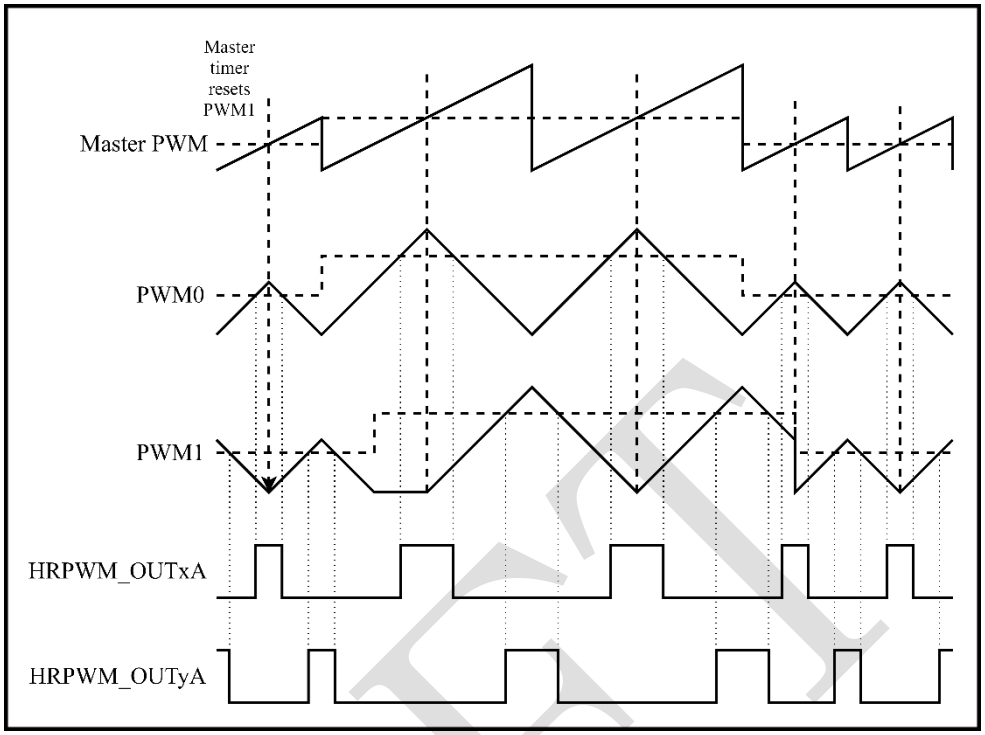


图 17-28　交错计数在上-下模式下的示例

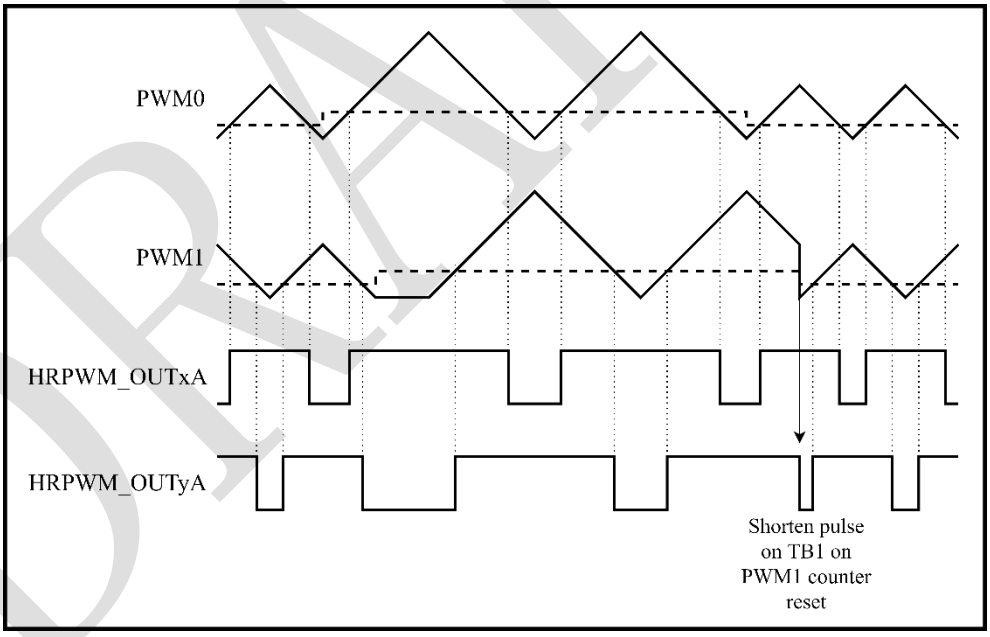


图 17-29　交错计数在上-下模式下的示例

注：在上-下计数模式下，比较值必须小于周期值减去 3×FHRPWM 时钟周期。
若 HRPWM_PWMxCR0.CKPSC = 0，则比较值上限为（HRPWM_PERxR - 0xC0）
若 HRPWM_PWMxCR0.CKPSC = 1，则比较值上限为（HRPWM_PERxR - 0x60）
若 HRPWM_PWMxCR0.CKPSC = 2，则比较值上限为（HRPWM_PERxR - 0x30）

以上适用于在定时单元内部产生的比较事件。

上-下计数模式支持以下功能：

- 交错模式
- 死区插入
- 推挽模式，计数器= 0 时完成推挽交替（如图 17-22 所示）
- 延迟空闲模式
- 突发模式
- 带有大于比较的 PWM 模式

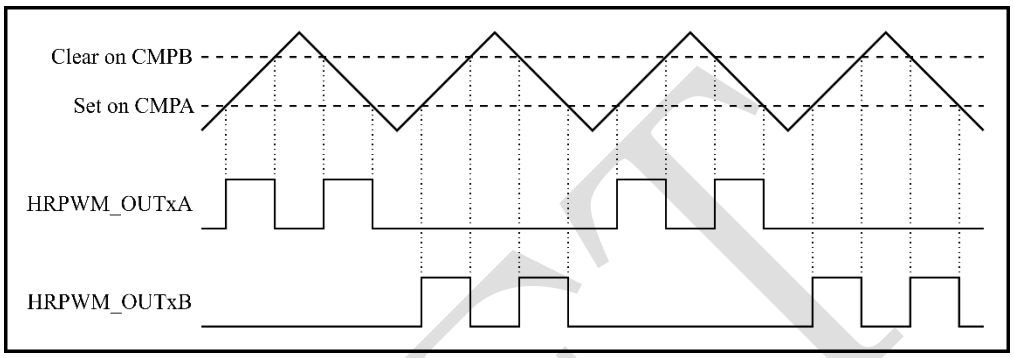


图 17-30　推挽模式在上-下计数模式的示例

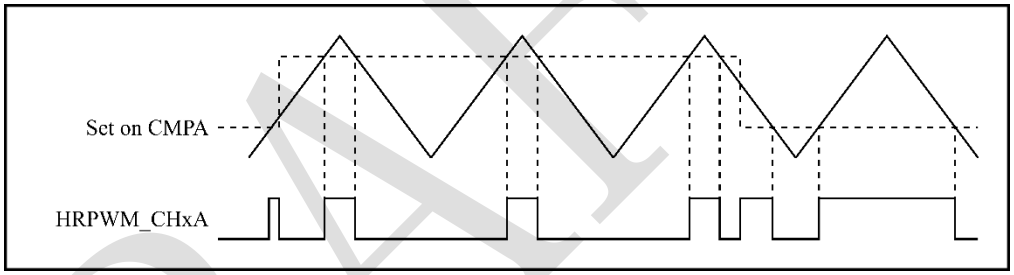


图 17-31 大于比较模式在上-下计数模式的示例

注意：上下计数模式不支持以下特性：

- 自动延迟模式
- 均衡空闲模式
- 触发半模式

上下计数模式支持的捕获功能区别如下：

- 当上计数时，捕获值以计数开始作为起始点
- 当下计数时，捕获值以 PER 事件作为起始点
- 捕获寄存器的第 16 位保存计数方向状态

计数翻转事件在上-下计数模式下的定义与上计数模式下有所不同，以支持不同的工作条件。计数翻转事件可以由以下方式生成：

- 在计数值等于 0 时生成（波谷模式）
- 在计数值等于 HRPWM_PERxR 时生成（波峰模式）
- 在计数值等于 0 或 HRPWM_PERxR 时生成（波峰-波谷模式）

计数翻转事件在 HRPWM 中会用在很多地方，根据不同用途可以配置不同的生成方式（波谷、波峰或波峰-波谷）。表 17-11 总结了不同用途下相应的计数翻转模式配置位（HRPWM_PWMxCR1.xxROM）。

表 17-11　翻转事件用途和生成方式编程

翻转事件用途	编程控制位
输出置位/复位	OUTROM[1:0]
中断请求	ROM[1:0]
ADC 触发事件	ADROM[1:0]
外部事件滤波	EEVROM[1:0]
故障/事件计数器	FLTROM[1:0]

注：对于同时考虑复位以及翻转的事件（IRQ 和 RSTU），HRPWM_PWMxCR1.ROM 只影响翻转事件产生。无论 HRPWM_PWMxCR1.ROM 值是多少，复位事件总会被考虑。

计数器翻转事件生成方式由 HRPWM_PWMxCR1.xxROM 的配置定义：

- HRPWM_PWMxCR1.xxROM = 00：在计数值等于 0 或者 HRPWM_PERxR 时生成（波峰-波谷模式）
- HRPWM_PWMxCR1.xxROM = 01：在计数值等于 0 时生成（波谷模式）
- HRPWM_PWMxCR1.xxROM = 10：在计数值等于 HRPWM_PERxR 时生成（波峰模式）

图 17-32 显示了推挽输出在上-下计数模式下的行为，输出在周期（计数器翻转）事件置位，（HRPWM_PWMxCR1.OUTROM =10）。

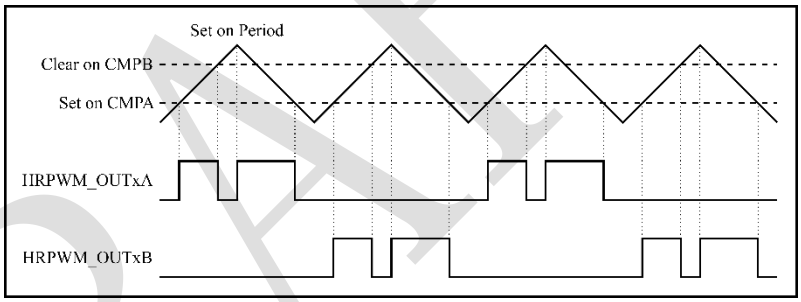


图 17-32　上-下计数模式下输出在周期事件置位（OUTROM[1:0]=10）

图 17-33 显示了在上-下计数模式下重复计数器是如何递减的。

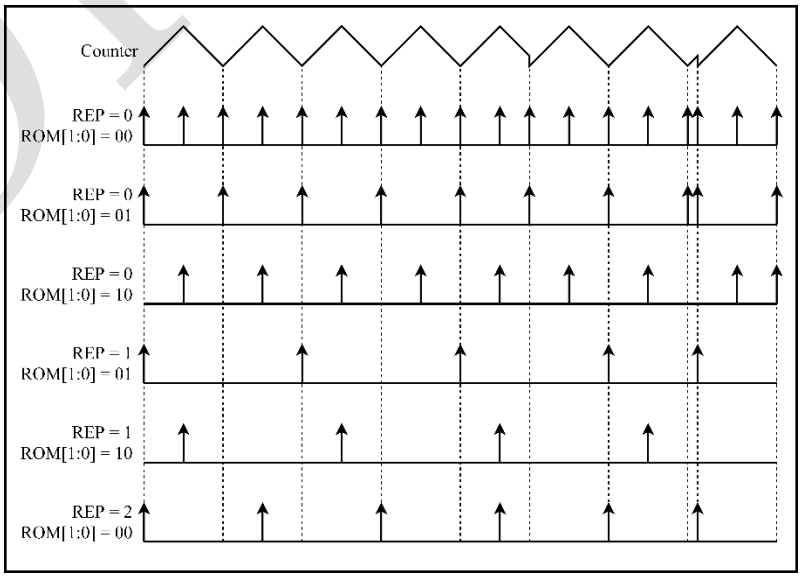


图 17-33　上-下计数模式下的周期重复事件

DAC 触发事件在上-下计数模式下与在上计数模式下一致。

事件消隐和加窗的功能在上-下计数模式下与在上计数模式下不同。
HRPWM_EEFxRx.EExFLTR 的功能依赖于 HRPWM_PWMxCR1.UDM 的配置，如表 17-12 所示，可以看出消隐/开窗区间根据 HRPWM_PWMxCR1.UDM 的不同有所不同。无论何时将翻转事件用于事件消隐或开窗时，翻转事件的生成受 HRPWM_PWMxCR1.EEVROM 控制。

表 17-12　EExFLTR[3:0]的功能与 UDM 位设置

EExFLTR[3:0]	上计数模式 （UDM = 0）	上-下计数模式 （UDM = 1）
1111	无消隐与加窗滤波	从上阶段 CMPB 到下阶段 CMPC 加窗
1110	无消隐与加窗滤波	从下阶段 CMPB 到下阶段 CMPC 加窗
1101	无消隐与加窗滤波	从上阶段 CMPB 到上阶段 CMPC 加窗
1100	从复位 / 翻转到 CMPD 加窗	从复位 / 翻转到 CMPD 加窗
1011	从复位 / 翻转到 CMPC 加窗	从复位 / 翻转到 CMPC 加窗
1010	从复位 / 翻转到 CMPB 加窗	从复位 / 翻转到 CMPB 加窗
1001	从复位 / 翻转到 CMPA 加窗	从复位 / 翻转到 CMPA 加窗
1000	无消隐与加窗滤波	从下阶段 CMPC 到下阶段 CMPD 消隐
0111	无消隐与加窗滤波	从下阶段 CMPA 到下阶段 CMPB 消隐
0110	无消隐与加窗滤波	从上阶段 CMPC 到上阶段 CMPD 消隐
0101	无消隐与加窗滤波	从上阶段 CMPA 到上阶段 CMPB 消隐
0100	从复位 / 翻转到 CMPD 消隐	从复位 / 翻转到 CMPD 消隐
0011	从复位 / 翻转到 CMPC 消隐	从复位 / 翻转到 CMPC 消隐
0010	从复位 / 翻转到 CMPB 消隐	从复位 / 翻转到 CMPB 消隐
0001	从复位 / 翻转到 CMPA 消隐	从复位 / 翻转到 CMPA 消隐
0000	无消隐与加窗滤波	无消隐与加窗滤波

17.4.7　置位/复位优先级和窄脉冲管理

本节介绍了在 3 个连续的 FHRPWM 时钟周期内出现多个置位和/或复位请求时，输出波形是如何产生的。

情形 1：时钟预分频 HRPWM_PWMxCR0.CKPSC < 5

每个 FHRPWM 周期内都会执行仲裁，具体分为三步：

1. 对于每个有效事件，会确定所需输出跳变（置位、复位或翻转）。
2. 在有效事件之间执行预定义的仲裁（从最高到最低优先级）：
   PER → CMPD → CMPC → CMPB → CMPA。
3. 在预定义高分辨率延迟仲裁中，在低分辨率事件和已经获得预定于仲裁的事件中，CLEAR 事件具有最高优先级。

如果 SET 和 CLEAR 请求之间的间隔小于 2×FHRPWM 周期，则行为取决于时间间隔和与 FHRPWM 时钟的对齐方式，如图 17-34 所示。

注：如果置位和复位请求是同时发生的，并且来自同一个定时单元，则 CMPx 的优先级生效，

如上面的步骤 2 所示。

例如，假设 CMPB = CMPD：

- 如果 CMPB 置位，CMPD 复位，输出会复位。
- 如果 CMPB 复位，CMPD 置位，输出会置位。

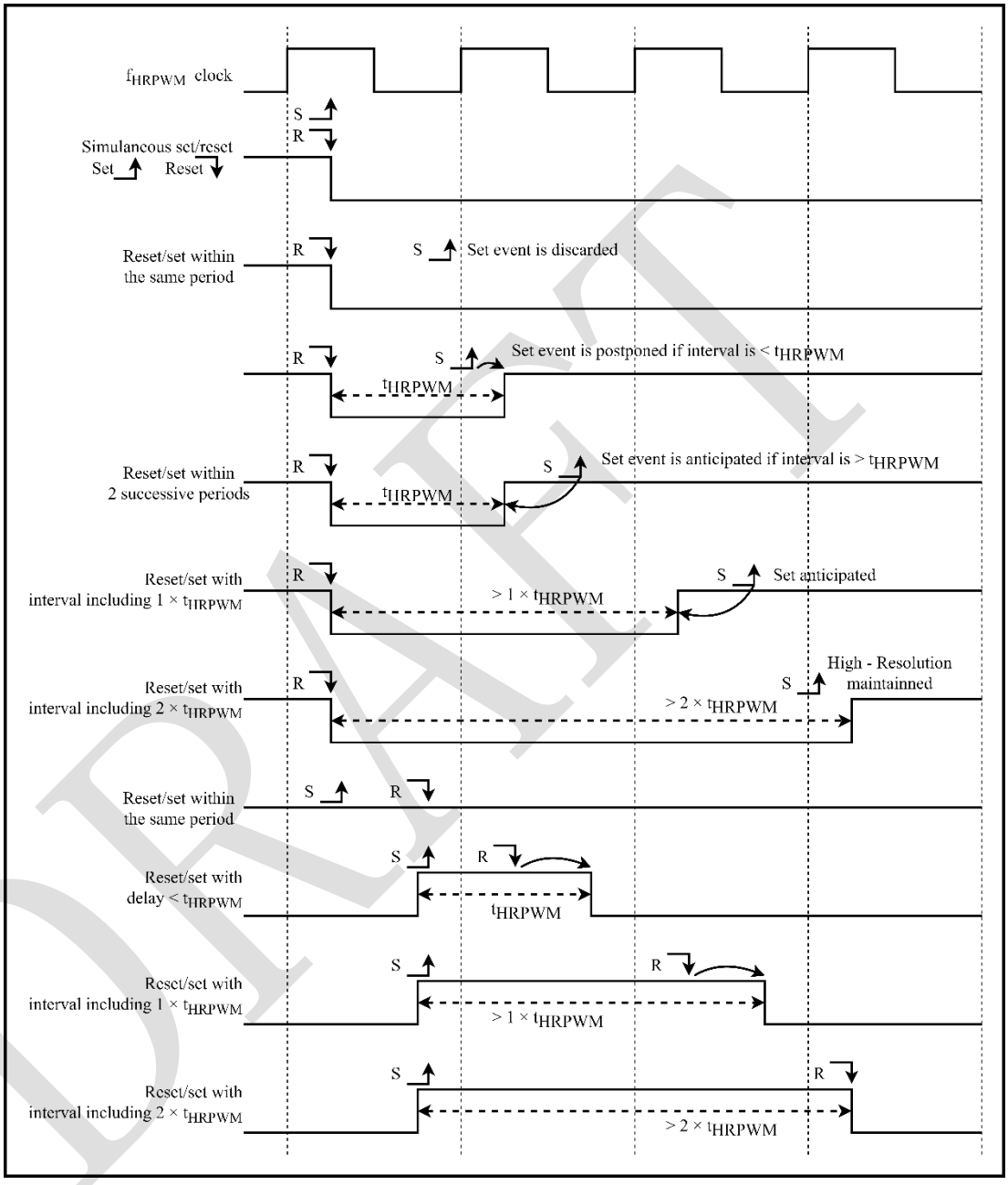


图 17-34　窄脉冲管理：近距离复位/置位事件的管理

如果置位和复位事件在相同的 FHRPWM 周期内产生，则复位事件具有最高优先级，置位事件会被忽略。

如果置位和复位事件生成的间隔小于 1 个 FHRPWM 时钟周期且跨越 2 个周期，会生成宽度为 1 个 FHRPWM 周期的脉冲。

如果置位和复位事件生成的间隔小于 2 个 FHRPWM 时钟周期，会生成宽度为 2 个 FHRPWM 周期的脉冲。

如果置位和复位事件生成的间隔为 2~3 个 FHRPWM 时钟周期，高分辨率在间隔超过 2 个完整 FHRPWM 时钟周期时有效。

如果置位和复位事件生成的间隔大于 3 个 FHRPWM 时钟周期，高分辨率一直有效。

并发置位请求/并发复位请求

如果为置位事件选择了多个源，则在同一 FHRPWM 时钟周期内出现多个置位请求时，会执行仲裁。

如果主定时器在同一 FHRPWM 时钟周期内发出多个请求，则会应用预先定义的仲裁，并会考虑单个有效的请求（优先级从高到低），无论有效的高分辨率设置如何：

MSTCMPER → MSTCMPD → MSTCMPC → MSTCMPB → MSTCMPA

注：建议避免从主定时器产生多组时间间隔低于 3×FHRPWM（CLEAR）请求到给定的定时单元，以保持高分辨率。

如果在同一 FHRPWM 时钟周期内出现多个定时器内部请求，则会应用预先定义的仲裁，并会按照以下优先级顺序考虑请求（从最高到最低），而不考虑有效定时：

PER → CMPD → CMPC → CMPB → CMPA

最后，最高优先级会分配给低分辨率事件：EXTEVNT0~9，RESYNC（来自 SYNC 事件（HRPWM_PWMxCR0.SYNCRST 或 HRPWM_PWMxCR0.SYNCSTRT 置 1 时）或软件复位），更新和软件置位（HRPWM_SETxAR.SST 或 HRPWM_SETxBR.SST）。

总之，对于并发事件（在相同的 FHRPWM 时钟周期内发生的事件），有效置位/复位事件将在以下事件之间进行仲裁：

- 来自主定时器的单个源（按照上文所述的固定仲裁）
- 来自定时单元的单个源
- 低分辨率事件

仲裁原则同样适用于并发复位请求。在这种情况下，复位请求具有最高优先级。

情形 2：时钟预分频 HRPWM_PWMxCR0.CKPSC ≥ 5

窄脉冲管理在高分辨率无效时可以简化。

在预分频时钟周期内发生的置位或者复位事件会延迟到预分频时钟的下一个有效边沿（比如计数器复位时间），即使仲裁仍在每个 FHRPWM 周期内进行。

17.4.8 外部事件全局调节

HRPWM 定时器可处理并非定时器中生成的事件，此类事件被称为外部事件，外部事件来自多个片上或者片外来源：

- 内置比较器
- 数字输入引脚（通常连接到片外比较器和过零检测器）
- 其他外设的片上事件（ADC 的模拟看门狗和通用定时器触发输出）

表 17-13 总结了可用于 10 个外部事件的源。

表 17-13 外部事件输入来源和相关功能

外部事件	外部事件来源（EExSRC [1:0]）				不同封装下的比较器和输入源	
	Src0（00）	Src1（01）	Src2（10）	Src3（11）	48-pin	64-pin and 100-pin
hrpwm_ext_evt0	HRPWM_EVT0	CMP1_OUT	TMR9_TRGO	ADC0_AWD0	Comp	Comp & input
hrpwm_ext_evt1	HRPWM_EVT1	CMP3_OUT	TMR3_TRGO	ADC0_AWD1	Comp	Comp & input
hrpwm_ext_evt2	HRPWM_EVT2	CMP5_OUT	TMR4_TRGO	CMP7_OUT	Comp & input	Comp & input
hrpwm_ext_evt3	HRPWM_EVT3	CMP0_OUT	CMP4_OUT	ADC1_AWD0	Comp & input	Comp & input
hrpwm_ext_evt4	HRPWM_EVT4	CMP2_OUT	CMP6_OUT	ADC1_AWD1	Comp & input	Comp & input
hrpwm_ext_evt5	HRPWM_EVT5	CMP1_OUT	CMP0_OUT	CMP8_OUT	Comp & input	Comp & input
hrpwm_ext_evt6	HRPWM_EVT6	CMP3_OUT	TMR8_TRGO	ADC2_AWD0	Comp & input	Comp & input
hrpwm_ext_evt7	HRPWM_EVT7	CMP5_OUT	CMP2_OUT	ADC3_AWD0	Comp & input	Comp & input
hrpwm_ext_evt8	HRPWM_EVT8	CMP4_OUT	TMR0_TRGO	CMP3_OUT	Comp & input	Comp & input
hrpwm_ext_evt9	HRPWM_EVT9	CMP6_OUT	TMR7_TRGO	PDM0_CMPH	Comp & input	Comp & input

外部事件调节电路可为给定通道选择事件源（使用 4:1 多路复用器），并可将信号源转换为可由交错配置矩阵处理的形式（例如，通过在外部事件通道上检测到的下降沿来触发输出复位）。

最多可调节 10 个外部事件通道，这些外部事件通道可同时用于 10 个定时器中的任何一个。由于这种调节通常取决于外部组件（比如过零检测器）和环境条件（滤波器设置通常与应用噪声级和信号有关），因此对于所有定时器是通用的。图 17-35 显示了单条事件通道的调节逻辑概览。

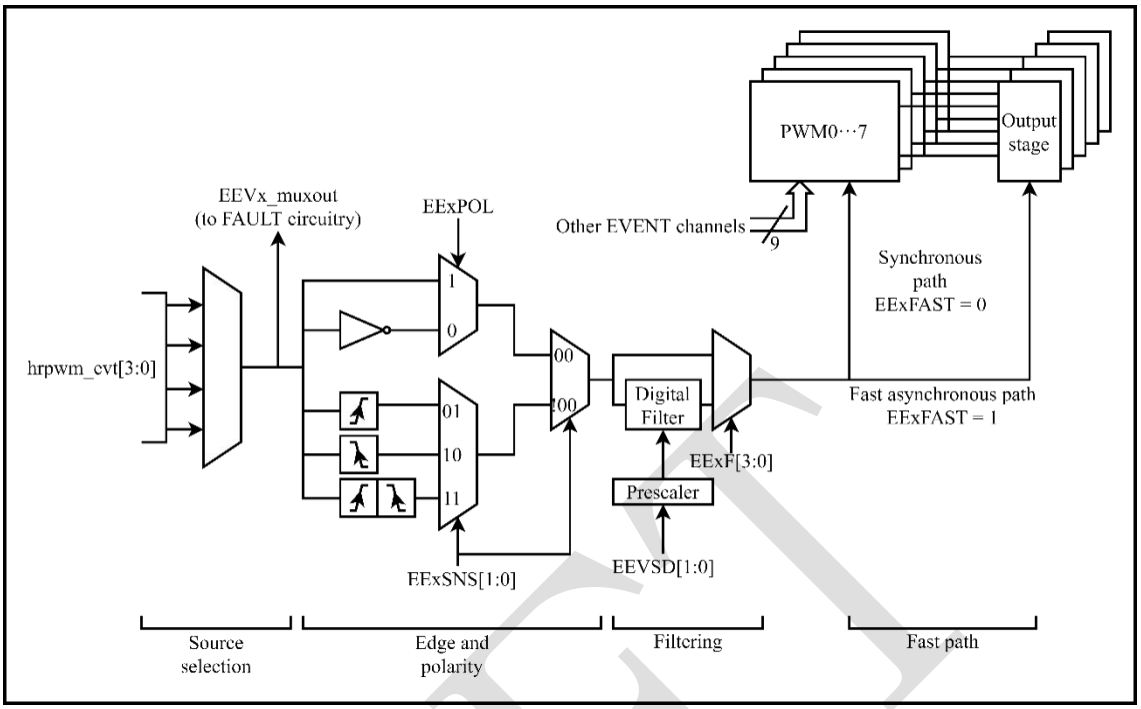


图 17-35　外部事件调节概览（显示了 1 条通道）

10 个外部事件通过 HRPWM_EECR0、HRPWM_EECR1 和 HRPWM_EECR2 寄存器初始化：

- 使用 HRPWM_EECRx.EExSRC 选择事件源，每个外部事件有 4 个源
- 使用 HRPWM_EECRx.EExSNS 选择有效沿，电平有效或者边沿有效（上升沿、下降沿或上升/下降沿），
- 使用 HRPWM_EECRx.EExPOL 选择触发电平的极性，极性在电平有效与边沿有效下均生效，
- 使用 HRPWM_EECRx.EExFAST 配置外部事件 0~9 的低延迟模式

注：即使 HRPWM_EECRx.EExSNS=0（选择电平有效），用作复位、ADC 触发事件的外部事件也是边沿有效：如果 HRPWM_EECRx.EExPOL＝0，触发事件在外部事件上升沿有效，如果 HRPWM_EECRx.EExPOL＝1，触发事件在外部事件下降沿有效。

计数器禁止后（HRPWM_MCR1.CENx=0），会立即丢弃外部事件，以防止输出状态更改和计数器复位，外部事件用于 ADC 触发事件时不受影响。

此外，还可以使用 HRPWM_EECRx.EExF 位为外部事件使能数字噪声滤波器。

数字滤波器由计数器组成，需要使用 N 个有效采样来验证输出跳变。如果输入值在计数器达到 N 之前发生变化，计数器会复位，跳变会被丢弃（视为伪事件）。如果计数器达到 N，跳变被视为有效，并会作为正确的外部事件进行传输。因此，数字滤波器会向正在进行滤波的外部事件添加延迟，延迟时长视采样时钟和滤波器长度（预期的有效采样数）而定。

采样时钟为 FHRPWM 时钟或由 FHRPWM 预分频而生成的特定时钟 f_{EEVS}，预分频通过 HRPWM_EECR2.EEVSD 定义。

外部事件延迟

外部事件调节能够根据性能预期调整外部事件的处理时间（以及相关延迟）：

- 常规工作模式：在该模式下，会在对交错配置矩阵进行操作之前，使用时钟对外部事件进行重新采样。此过程会增加一些延迟，但可以使用所有交错配置矩阵功能，从而生成外部事件触发的高分辨率脉冲。
- 快速工作模式：在该模式下，外部事件与输出动作之间的延迟得到最大限度地缩短，该模式便于实现超快速的过流保护等功能。

使用 HRPWM_EECRx.EExFAST 位来定义外部事件 0~9 的工作模式，这会影响输出脉冲上存在的延时和抖动，如下表 17-14 所示。

表 17-14　输出置位/复位的延迟/抖动与外部事件工作模式

EExFAST	反应时间延迟	响应时间抖动	输出脉冲抖动 （由 Ext 事件计数器复位）
0	5~6 个 FHRPWM 时钟周期	1 个 FHRPWM 时钟周期	无抖动，脉冲宽度保持高分辨率
1	最小延迟（取决于使用比较器还是数字输入）	最小抖动	1 个 FHRPWM 时钟周期的抖动 脉宽分辨率下降到 THRPWM

HRPWM_EECRx.EExFAST 模式仅适用于电平有效模式（HRPWM_EECRx.EExSNS＝00），不适用于边沿有效模式。

HRPWM_EECRx.EExFAST 模式可以对外部事件应用事件过滤（HRPWM_EEFxRx.EExFLTR!= 0x0，消隐和加窗），在这种情况下 HRPWM_EEFxRx.EExLTCH 必须为 0，不支持延迟模式。

注：相关 EExFAST 位置 1 后，不得修改外部事件配置（来源和极性）。

快速外部事件不能用于翻转输出，必须在 HRPWM_SETxyR 或 HRPWM_CLRxyR 寄存器中使能，而不能在两个寄存器中同时使能。

如果置位事件和复位事件（来自 2 个独立的快速外部事件）同时发生，则复位事件在交错配置矩阵中的优先级最高，输出变为无效状态。

如果 HRPWM_EECRx.EExFAST 置 1，则在外部事件发生后的 11 个 FHRPWM 时钟周期内，输出都不能更改。

图 17-36 和图 17-37 给出了进行输出置位/复位和计数器复位时对外部事件的实际响应时间示例。

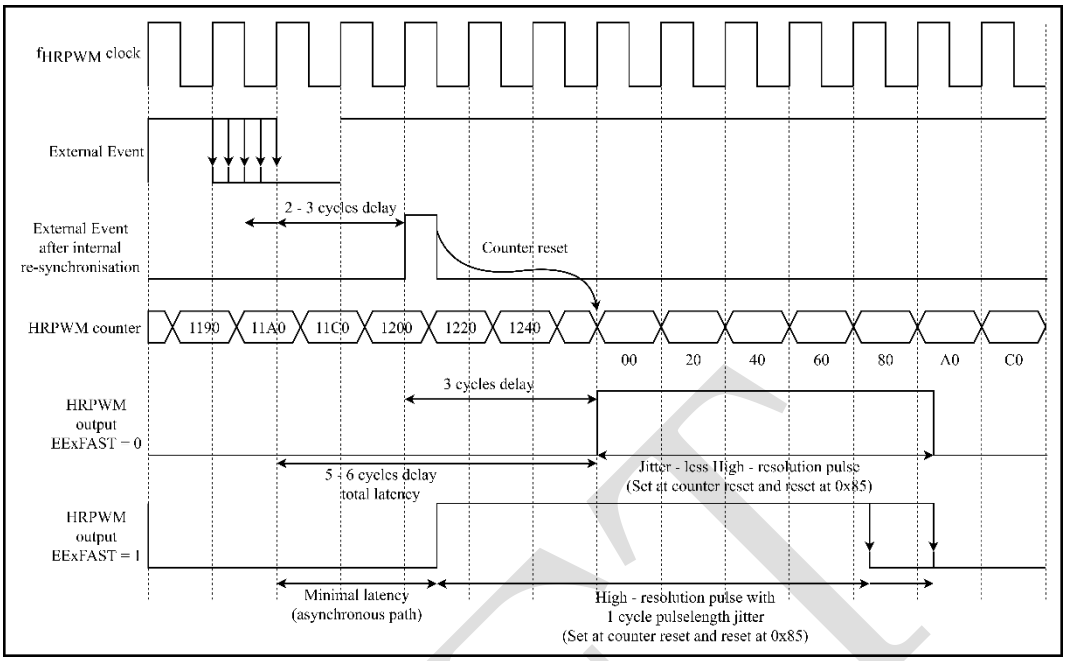


图 17-36　外部事件的延迟（外部事件使计数器复位和输出置位）

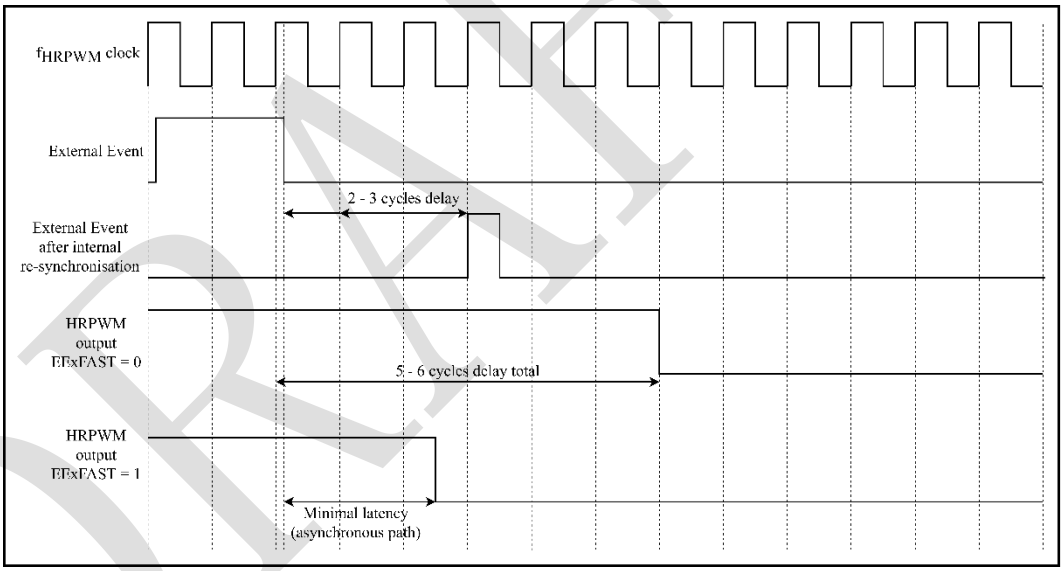


图 17-37　外部事件的延迟（外部事件使输出复位）

17.4.9 定时单元中的外部事件过滤

经过调节后，10 个外部事件可供所有定时单元使用。

这些事件可直接使用，且在定时单元计数器使能后（HRPWM_MCR1.CENx 置 1）立即生效。

此外，还可对这些事件进行过滤，以便在限制时间段内执行操作，该时间段通常与计数周期有关。可执行两种操作：

- 消隐模式，在定义的时间段内屏蔽外部事件，
- 加窗模式，仅在定义的时间段内使能外部事件。

这些模式通过 HRPWM_EEFxRx.EExFLTR 来使能，8 个定时单元都具有针对这 10 个外部事件的可编程过滤器设置。

消隐模式

在事件消隐模式下，如果外部事件发生在给定的消隐周期内，则会被忽略，如图 17-38 所示。举例来说，为了避免 PWM 周期开始时电流限制因开关噪声而失效，可使用此模式。当 HRPWM_EEFxRx.EExFLTR 的值在 0001 变化到 1000 范围内时，该模式有效。

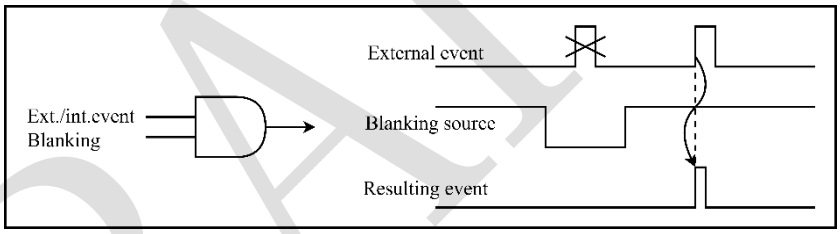


图 17-38　事件消隐模式

在事件延迟模式下，不会立即考虑外部事件，而是会将其保存（锁存）下来，并在消隐周期结束后立即生成外部事件，如图 17-39 所示。通过将 HRPWM_EEFxRx.EExLTCH 置 1 来使能此模式。

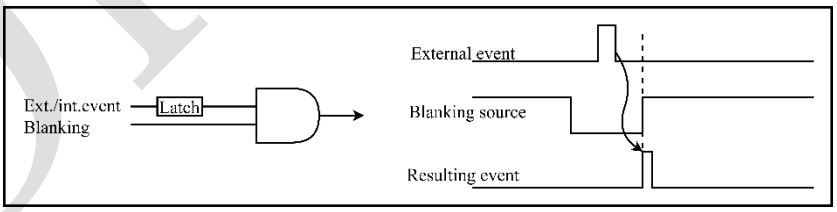


图 17-39　事件延迟模式

消隐信号来自多个来源，编码如下：

- 定时器本身：消隐持续时间从计数器复位开始，到比较值匹配为止（HRPWM_EEFxRx.EExFLTR＝0001 到 0100），也可以从一个比较值匹配开始，到另一个比较值匹配为止（HRPWM_EEFxRx.EExFLTR=0101 到 1000）。在 Updown 模式（HRPWM_PWMxCR1.UDM 设置为 1）下，计数复位事件与 HRPWM_PWMxCR1.ROM 设置有关。

图 17-40、图 17-41 举例说明了常规模式和延迟模式下，各种边沿有效和电平有效的外部事件消隐。

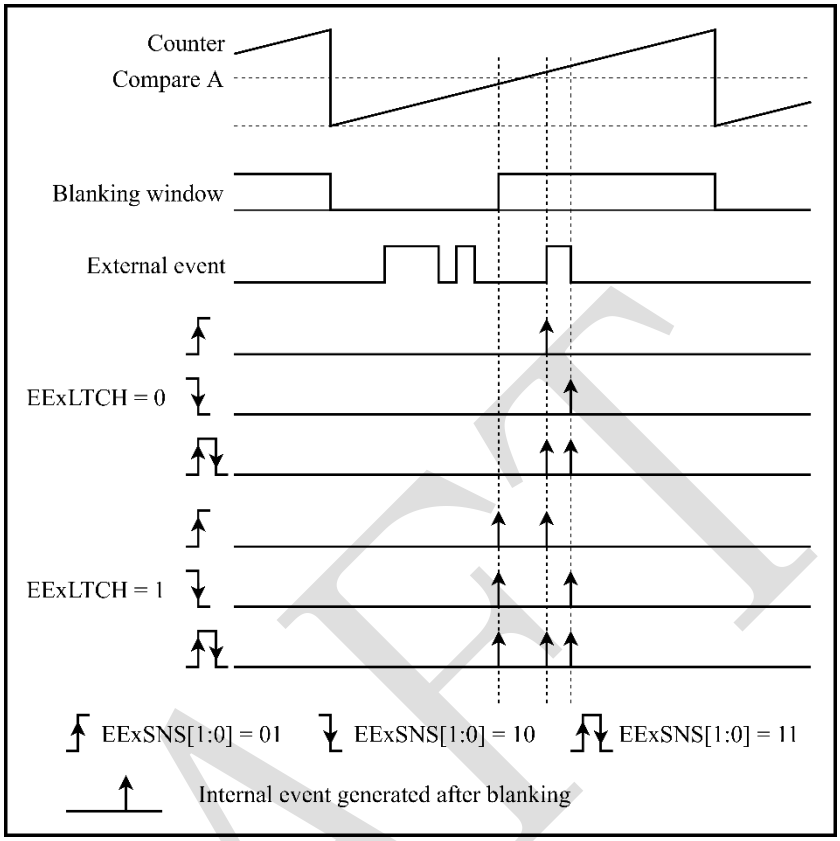


图 17-40　采用边沿有效的外部事件消隐

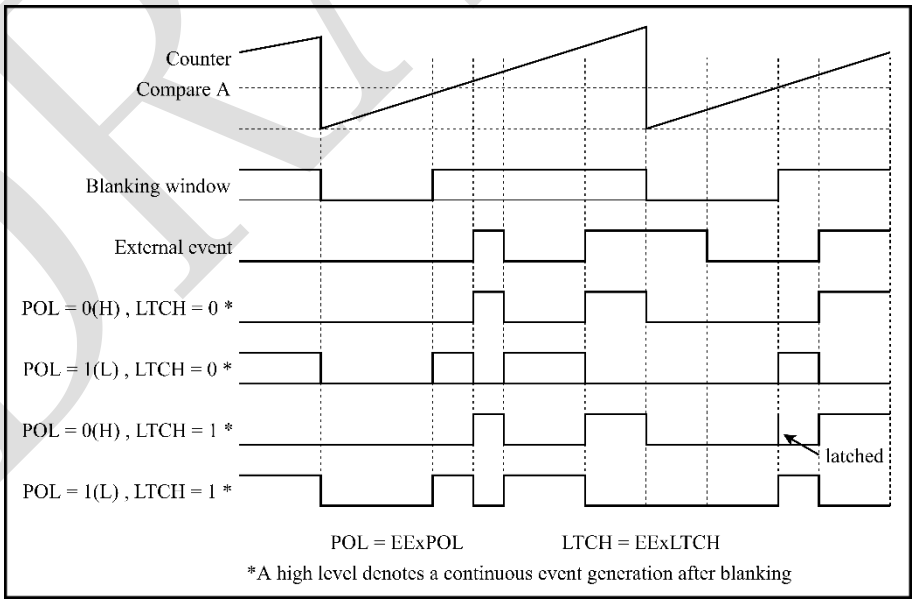


图 17-41　采用电平有效的外部事件消隐

加窗模式

在事件加窗模式中，在非延迟模式下，仅当事件在给定事件窗口内发生时才会被考虑，否则会被忽略；在延迟模式下，事件在给定事件窗口外发生时也会被考虑。当 HRPWM_EEFxRx.EExFLTR 的值在 1001 到 1111 范围内时，该模式有效。

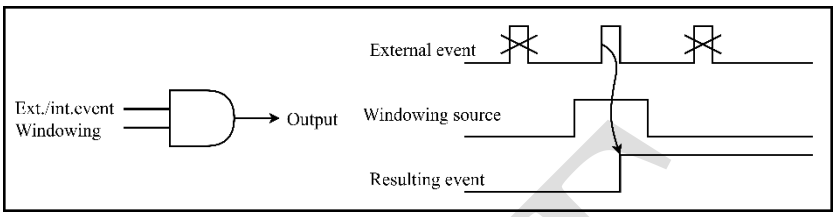


图 17-42　事件加窗模式

使用 HRPWM_EEFxRx.EExLTCH 开启事件延迟模式

- 当事件延迟模式有效（HRPWM_EEFxRx.EExLTCH=1）
  - 如果事件在窗口期间发生，则事件直接生效
  - 如果事件未在窗口期间发生，则事件不会生效（不产生超时事件）
- 当事件延迟模式无效（HRPWM_EEFxRx.EExLTCH=0）
  - 如果事件在窗口期间发生，则事件直接生效
  - 如果事件未在窗口期间发生，则事件延迟到窗口末尾生效（产生超时事件）

加窗模式可用于过滤同步事件。超时事件可用于在预期发生的同步事件未发生时，产生一个默认的同步事件，比如用于转换器启动阶段。

加窗信号来自多个来源，编码如下：

- 定时器本身：加窗持续时间从计数器复位开始，到比较值匹配为止（HRPWM_EEFxRx.EExFLTR＝1001 到 1100），也可以从一个比较值匹配开始，到另一个比较值匹配为止（HRPWM_EEFxRx.EExFLTR =1101 到 1111）。在 Updown 模式（HRPWM_PWMxCR1.UDM 设置为 1）下，计数复位事件与 HRPWM_PWMxCR1.ROM 设置有关。

图 17-43 和图 17-44 举例说明了在常规模式和事件延迟模式下，各种边沿有效和电平有效的外部事件加窗。

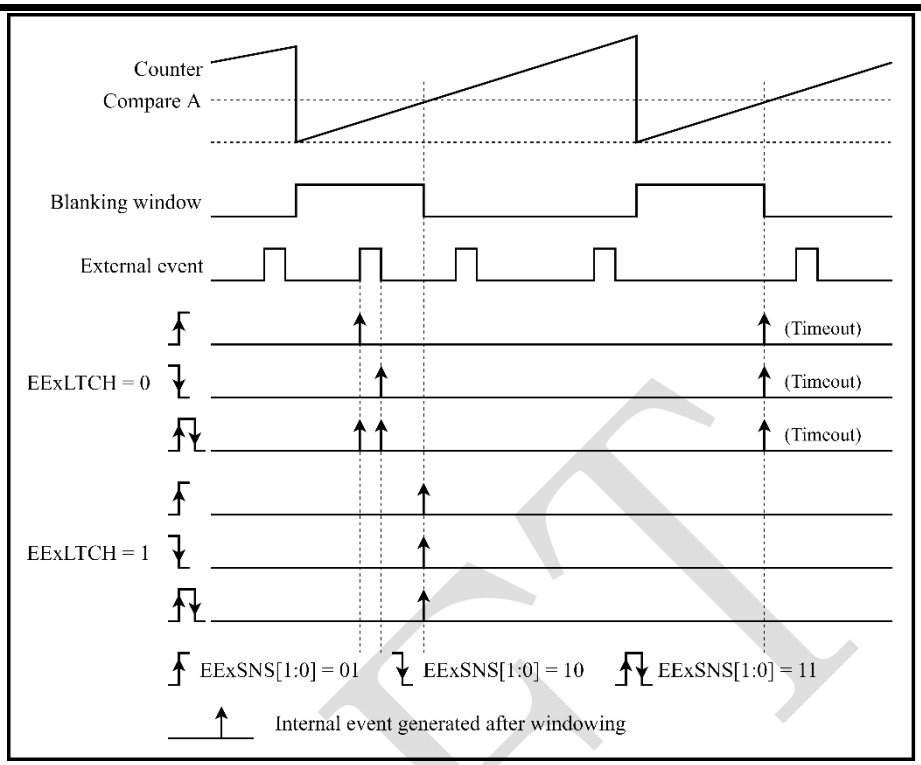


图 17-43　采用边沿有效的外部事件加窗

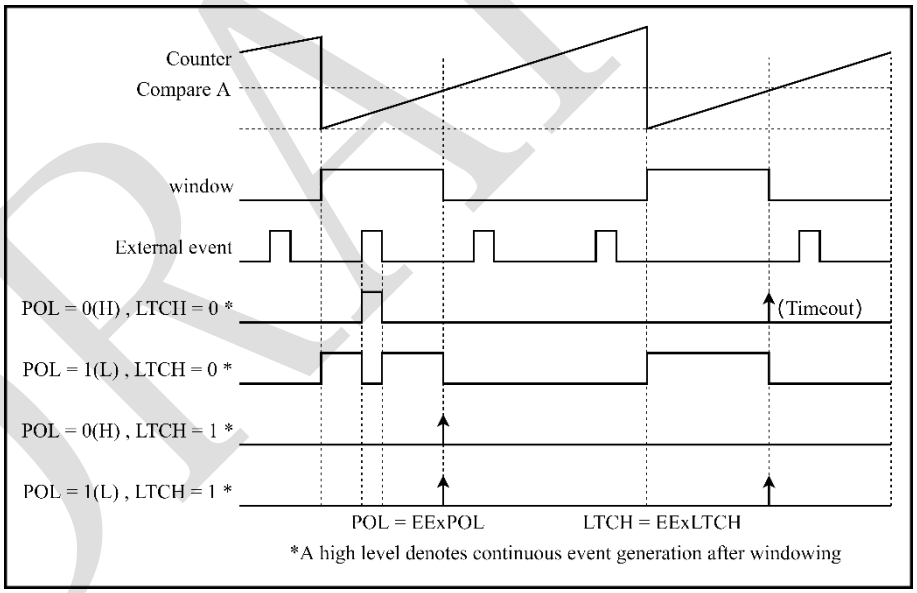


图 17-44　采用电平有效的外部事件加窗

外部事件计数器

每个定时单元在事件过滤之后，有一个外部事件计数器，通常实现波谷信号过滤。计数器可以对经过事件过滤的 10 个外部事件进行滤波，如图 17-45 所示。

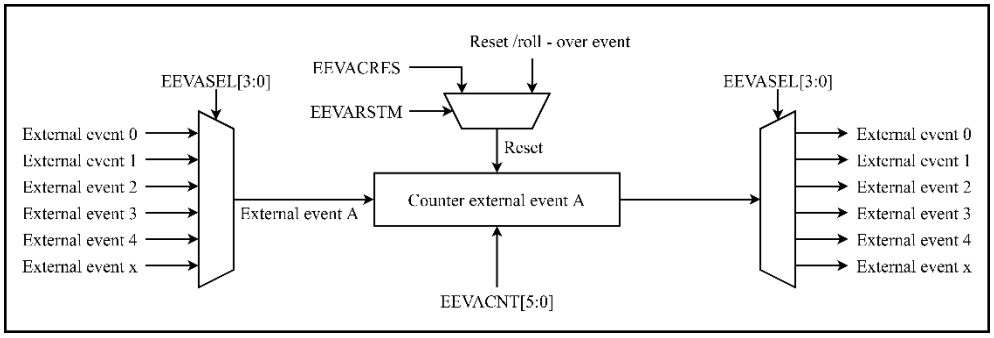


图 17-45　外部事件计数器-通道 A

外部事件计数器通过配置 HRPWM_EEFxR2.EEVACE 使能。此模式仅适用于边沿有效的外部事件（HRPWM_EECRx.EExSNS＝01,10 或 11）。

只有当有效边沿的数目大于或等于（HRPWM_EEFxR2.EEVACNT + 1）编程设定的值时，外部事件才会传输到定时器。

计数器支持两种工作模式：

- 无累积模式：当 HRPWM_EEFxR2.EEVARSTM=0 时，外部事件计数器在每一个复位/翻转事件处清零：外部事件只有在给定的 PWM 周期内出现多次时才是有效的。
- 累积模式：当 HRPWM_EEFxR2.EEVARSTM=1 时，只有在上一 PWM 周期内没有出现外部事件时，外部事件计数器才会清零。这是一种累积模式，外部事件必须在多个 PWM 周期内都至少出现一次，如下图 17-46 所示。

必须先设定计数器值之后再使能外部事件计数器（设置 EEVACNT[5:0]位之后设置 EVACE 位）。

计数器使能之后，HRPWM_EEFxR2.EEVACNT 可在任何时间即时改变。
HRPWM_EEFxR2.EEVACNT 的值在下一复位/翻转事件（根据 HRPWM_EEFxR2.EEVARSTM 的配置生成）处生效，或者在软件复位（HRPWM_EEFxR2.EEVACRES 置 1）之后生效。

HRPWM_EEFxR2.EEVACE 置 1 之后，不可以修改 HRPWM_EEFxR2.EEVASEL。

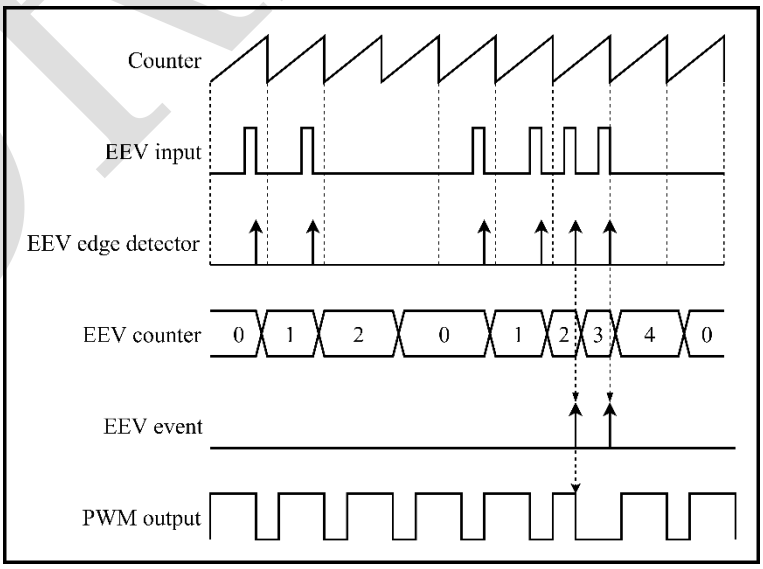


图 17-46　外部事件计数器累积模式（EEVXRSTM＝1，EEVXCNT＝2）

17.4.10 延迟保护

HRPWM 提供特定的保护机制，适用于需要在有效脉冲结束后或推挽周期结束后以延迟的方式关断 PWM 输出的情形。

这些功能通过 HRPWM_OUTxR.DLYPRTEN 使能，并将使用特定的外部事件。

延迟空闲模式

在该模式下，有效电平会在保护有效之前结束。所选外部事件会使输出在有效脉冲结束时进入空闲模式，有效脉冲由 HRPWM_CLRxAR 或 HRPWM_CLRxBR 中的输出复位事件定义。

一旦保护被触发，会永久保持空闲模式，但计数器会继续工作，直至输出重新使能。HRPWM_OENR.OENxA 和 HRPWM_OENR.OENxB 不受延迟空闲模式进入的影响。要退出延迟空闲模式并恢复操作，需要将 HRPWM_OENR.OENxA 和 HRPWM_OENR.OENxB 重写为 1。输出状态将在发出输出使能命令后第一次跳变为有效状态时更改。

注：在有效脉冲结束之前，进入了延迟空闲模式便不能立即退出：务必先确保输出处于空闲状态，然后再重新开始运行模式。具体做法可以是等到下一周期，或者轮询 HRPWM_PWMxISR.OUTA 和/或 HRPWM_PWMxISR.OUTB 状态位。

延迟空闲模式可应用于单路输出（HRPWM_PWMxISR.DLYPRT = x00 或 x01），也可应用于两路输出（HRPWM_PWMxISR.DLYPRT = x10）。

延迟空闲模式进入后，可以生成中断或 DMA 请求。外部事件到达后，会立即将 HRPWM_PWMxISR.DLYPRT 标志置 1，和输出上的有效脉冲是否结束无关。

延迟空闲模式触发后，通过 HRPWM_PWMxISR.OUTASTA 和 HRPWM_PWMxISR.OUTBSTA 确定输出状态。即使延迟空闲模式应用于单路输出，也会更新这两个状态位。如果使能了推挽模式，HRPWM_PWMxISR.IPPSTA 标志会指示延迟保护请求在哪一周期发生。

无论定时器采用何种工作模式（常规、推挽、死区），均可使用此模式。此模式仅支持 2 个外部事件

- HRPWM_Event5 和 HRPWM_Event6（对于 PWM0~PWM3）
- HRPWM_Event7 和 HRPWM_Event8（对于 PWM4~PWM7）

仅当计数器使能后（HRPWM_MCR1.CENx 置 1），才能触发延迟保护模式。即使 HRPWM_MCR1.CENx 位已复位，在 HRPWM_OENR.OENxy 置 1 之前，延迟保护模式仍保持有效状态。

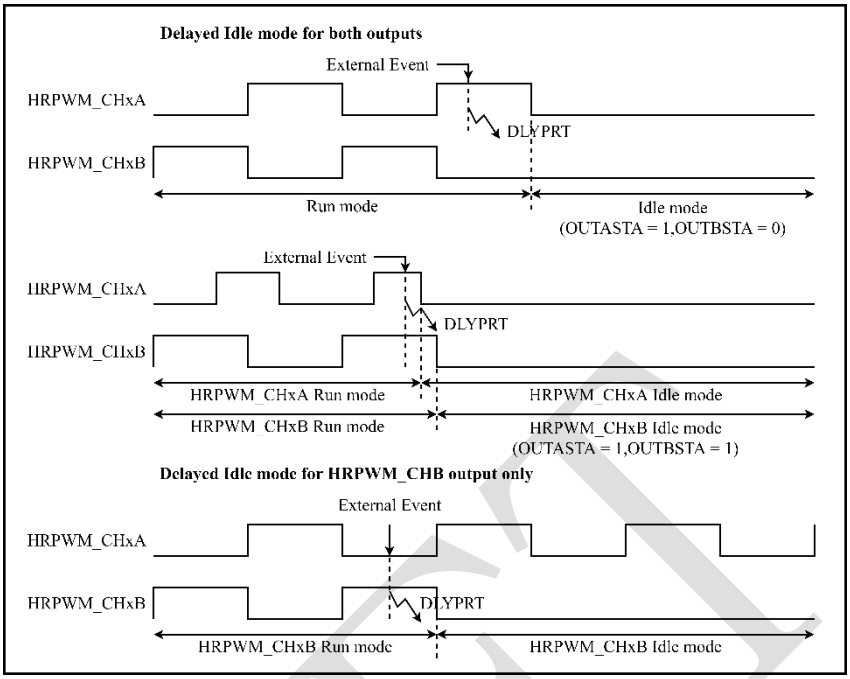


图 17-47　延时空闲模式进入

延迟空闲模式的优先级要高于突发模式：一旦触发了延迟空闲保护，就会丢弃任何突发模式退出请求。相反，如果在突发模式有效时退出延迟保护，突发模式将正常恢复，输出将保持为空闲状态，直至退出突发模式。这些不同情况的概述如图 17-48 所示。

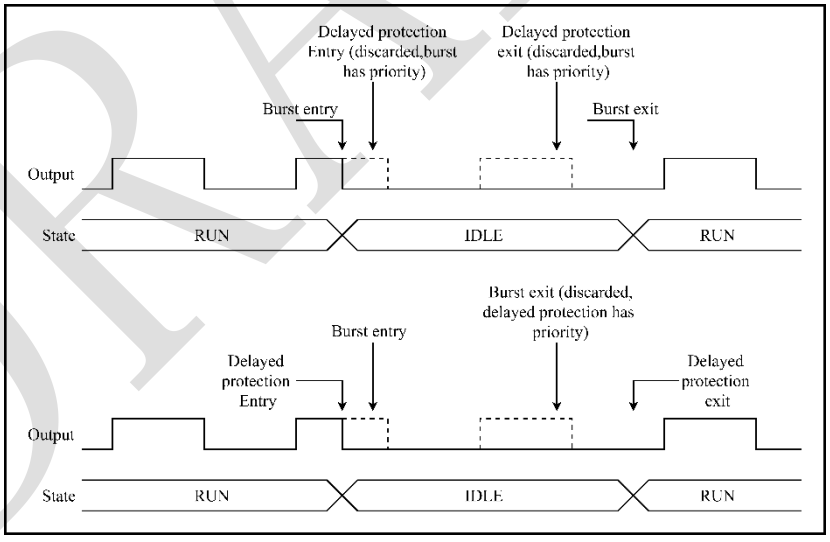


图 17-48 突发模式和延迟保护的优先级（DIDL = 0）

如果使能突发模式延迟进入（HRPWM_OUTxR.DIDLx 置 1）则会使用相同的优先级，如图 17-49 所示。

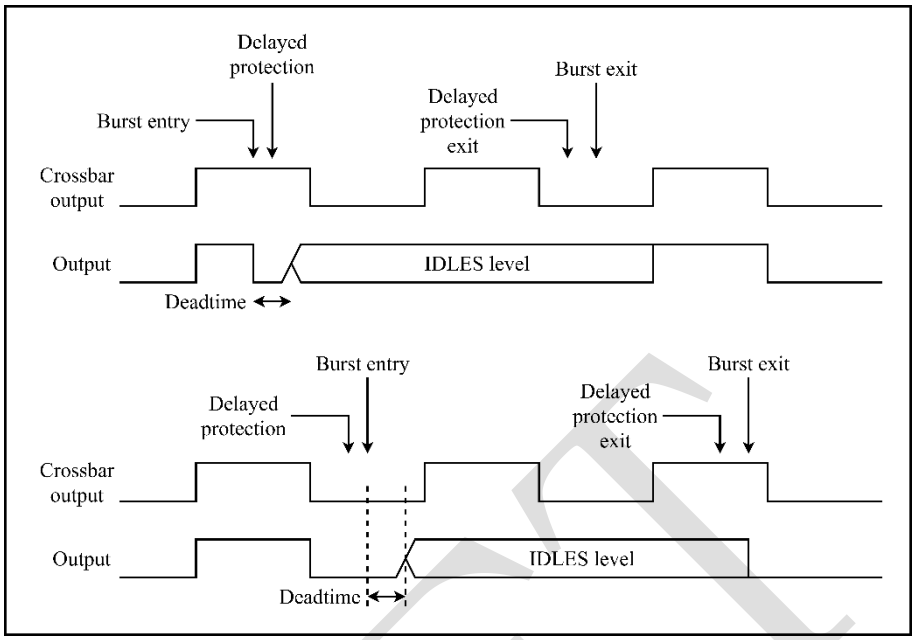


图 17-49　突发模式和延迟保护的优先级（DIDL = 1）

均衡空闲

均衡空闲仅在推挽模式下可用，在有效脉冲因保护的原因而缩短的情况下，可以在两路输出上实现脉冲宽度均衡。早于设定时间提前终止的脉冲宽度会复制到另一路输出上，两路输出随后会进入空闲状态，直至通过软件恢复正常工作。此模式通过向 HRPWM_OUTxR.DLYPRT 写入 x11 来使能。

均衡空闲模式仅支持 2 个外部事件：

- HRPWM_Event5 和 HRPWM_Event6（对于 PWM0~PWM3）
- HRPWM_Event7 和 HRPWM_Event8（对于 PWM4~PWM7）

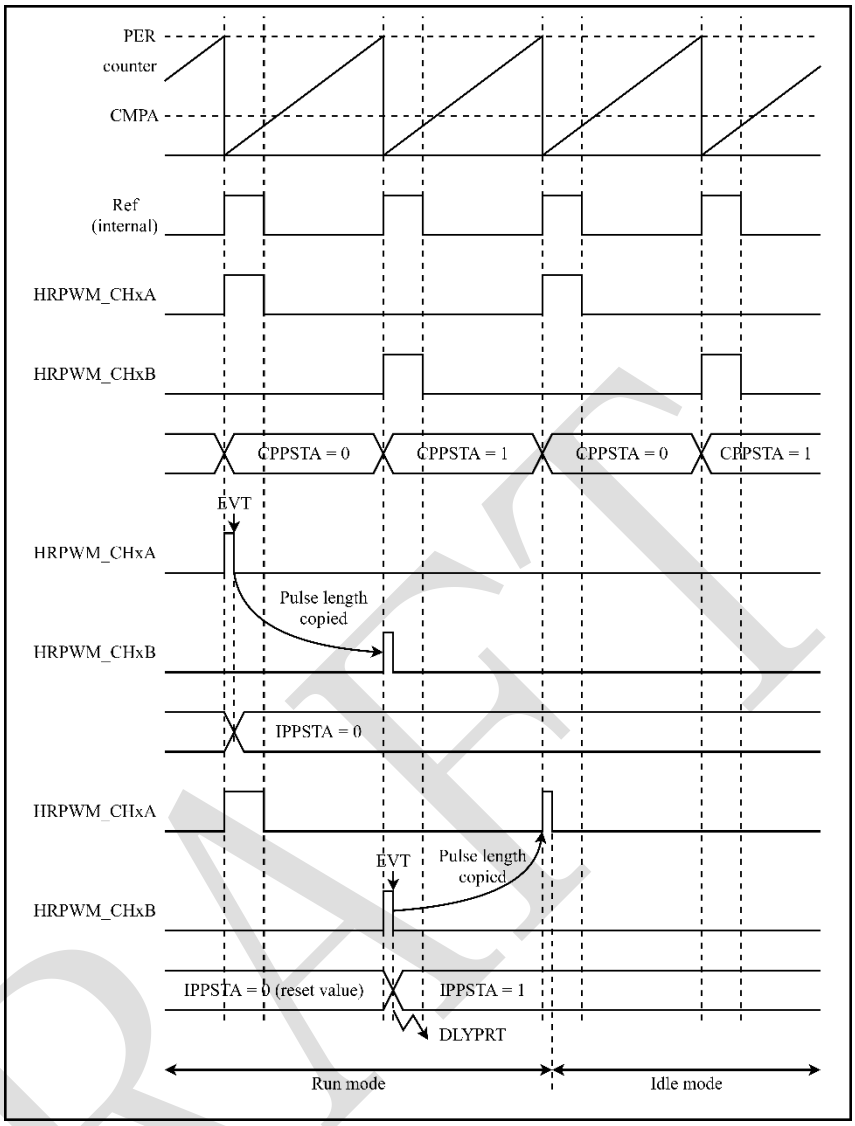


图 17-50 均衡空闲保护示例

如果均衡空闲模式使能，所选外部事件会触发将计数值捕获到比较 D 有效寄存器（该捕获值用户不可访问）。推挽模式会额外保持一个周期，以重复缩短的脉冲：常规输出置位事件保持的同时，会生成新的输出复位事件。

随后会进入空闲模式，输出会采用由 HRPWM_OUTxR.IDLESx 定义的电平。均衡空闲模式进入由 HRPWM_OUTxR.DLYPRT 标志指示，HRPWM_PWMxISR.IPPSTA 标志则指示外部事件在哪一周期发生，可用于确定缩短脉冲的顺序（先 OUTA 后 OUTB，或者相反）。

定时单元工作不会中断，计数器继续运行。

使能均衡空闲模式，需要执行下列初始化操作：

- 定时器在连续模式下工作（HRPWM_PWMxCR0.CONT = 1）
- 使能推挽模式（HRPWM_PWMxCR0.PSHPLL = 1）
- HRPWM_CMPDxR 设为 0 已更新到有效寄存器（使用软件更新）
- HRPWM_PWMxCR0.DELCMPD 设为 00（自动延迟模式禁止）
- HRPWM_PWMxISR.DLYPRT 设为 x11（延迟保护使能）

注：均衡空闲工作期间，不得对 HRPWM_CMPDxR 寄存器执行写操作。保留 CMPD 事件，不得用于其他用途。

注：在均衡空闲模式下，建议避免使用多个外部事件或基于软件的复位事件，后者会导致输出复位。如果同一周期内此类事件先于均衡空闲请求到达，则会导致输出脉冲不均衡（第 1 个脉冲长度由外部事件或软件复位决定，而第 2 个脉冲长度则由均衡空闲模式进入决定）。

均衡空闲模式下可用的最小脉宽为 4 个 FHRPWM 时钟周期。

如果在计数器达到其最小值之前进行捕获，会先将当前脉冲最多延长为 4 个 FHRPWM 时钟周期，然后再将其复制到次级输出。在任何情况下，脉宽始终都是均衡的。

HRPWM_OENR.OENxA 和 HRPWM_OENR.OENxB 不受均衡空闲模式进入的影响。要退出均衡空闲模式并恢复运行，需要同时将 HRPWM_OENR.OENxA 和 HRPWM_OENR.OENxB 重写为 1。输出状态将在输出使能后的第一次有效跳变时更改。

可采用与延迟空闲模式进入相似的方法恢复运行。例如，如果外部事件到达时输出 A 有效（延迟空闲模式在输出 B 脉冲后生效），可先为输出 A 启动重启序列。为此，需要轮询 HRPWM_PWMxISR.CPPSTA。假如 HRPWM_PWMxISR.IPPSTA 标志等于 0，运行将在 HRPWM_PWMxISR.CPPSTA 为 0 时恢复。

为了获得特定的重启序列，可轮询 CPPSTA 了解哪一输出将先有效。这样一来，便可使用与空闲模式进入序列相同的序列进行重启：如果外部事件在输出 A 有效期间到达，则将在输出 A 有效时(HRPWM_PWMxISR.CPPSTA = 0)发起重启序列。

注：正在执行脉冲均衡序列时，不得禁止均衡空闲模式。需要等到 CMPD 标志置 1（指示序列已结束）才能复位 HRPWM_OUTxR.DLYPRTEN。

仅当计数器使能后（HRPWM_MCR1.CENx 置 1），才能触发均衡空闲保护模式。即使 HRPWM_MCR1.CENx 已复位，在 HRPWM_OENR.OENxy 置 1 之前，均衡空闲保护模式仍保持有效状态。

下列情况下，均衡空闲模式可与突发模式一起使用：

- HRPWM_BMCR.BMDISx 必须为 0。
- 输出处于突发空闲状态时，不得触发均衡空闲保护。

均衡空闲模式的优先级要高于突发模式：一旦触发了均衡空闲保护，就会丢弃任何突发模式退出请求。相反，如果在突发模式有效时退出延迟保护，突发模式将正常恢复。

均衡空闲自动恢复

均衡空闲模式可以配置为在一个触发事件结束后自动恢复运行模式。

一旦被缩短的脉冲宽度被复制到另外一个输出之后，脉冲宽度恢复为原始值，定时单元恢复运行:两个输出保持在 RUN 状态模式。

均衡空闲自动恢复通过配置 HRPWM_OUTxR.BIAR 来使能。

均衡空闲自动恢复只能在 HRPWM_PERxR 中周期大于 6 个 FHRPWM 时钟周期时才可以使用。

注：只有当 HRPWM_OUTxR.DLYPRT = 011 或 111 时，此位才有意义，否则将被忽略。

注：在均衡空闲自动恢复模式下，必须将 HRPWM_OUTxR.IDLESx 状态设置为无效电平。

17.4.11 寄存器预加载和更新管理

大部分 HRPWM 寄存器均已缓存，可根据需要进行预加载。这样做通常可避免波形被与有效事件（置位/复位）不同步的寄存器更新更改。

使能预加载模式后，系统访问到的寄存器变为影子寄存器。收到软件发出的更新请求或与事件同步的更新请求之后，影子寄存器的内容会传输到有效寄存器。

默认情况下，HRPWM_MCR0.PREEN 和 HRPWM_PWMxCR0.PREEN 会复位，寄存器不会预加载：任何写操作均会直接更新有效寄存器。如果 HRPWM_MCR0.PREEN 或 HRPWM_PWMxCR0.PREEN 在定时器运行且预加载已使能时复位，预加载寄存器的内容会直接传输到有效寄存器中。

每个定时单元和主定时器都有自己的 HRPWM_MCR0.PREEN 或 HRPWM_PWMxCR0.PREEN。如果 HRPWM_MCR0.PREEN 或 HRPWM_PWMxCR0.PREEN 位置 1，仅当发生更新事件时，预加载寄存器才会使能，其内容才会传输到有效寄存器。

如果需要使用预加载功能，可选择两种方式初始化定时器：

- 刚好在定时器初始化结束时使能 HRPWM_MCR0.PREEN 或 HRPWM_PWMxCR0.PREEN，可以在定时器使能之前将预加载寄存器内容传输到有效寄存器（通过将 HRPWM_MCR1.MCEN 和 HRPWM_MCR1.CENx 置 1）。
- 在初始化过程中的任何时间使能 HRPWM_MCR0.PREEN 或 HRPWM_PWMxCR0.PREEN，并恰好在启动之前强制进行软件更新。

表 17-15 列出了可预加载的寄存器，并总结了可用更新事件。

表 17-15 HRPWM 可预加载的寄存器和相关的更新源

定时器	可预加载寄存器	预加载使能	更新源
主定时器	HRPWM_MPER.MPER HRPWM_MREP.MREP HRPWM_MCMPAR.MCMPA HRPWM_MCMPBR.MCMPB HRPWM_MCMPCR.MCMPC HRPWM_MCMPDR.MCMPD	HRPWM_MCR0.PREEN	软件更新 周期重复事件 复位/翻转事件
定时器x（x=0~7）	HRPWM_PERxR.PER HRPWM_REPxR.REP HRPWM_CMPxAR.CMPA HRPWM_CMPxBR.CMPB HRPWM_CMPxCR.CMPC HRPWM_CMPxDR.CMPD HRPWM_DTxR.ALL HRPWM_DTxR.ALL HRPWM_SETxAR.ALL HRPWM_CLRxAR.ALL	HRPWM_PWMxCR0.PREEN	软件更新 周期重复事件 复位翻转事件 硬件更新（来自Master PWM或PWMy）

	HRPWM_SETxBR.ALL HRPWM_CLRxBR.ALL HRPWM_RSTxR.ALL HRPWM_RSTxER.ALL HRPWM_RSTxR.ALL HRPWM_CAPAxCR.ALL HRPWM_CAPAxCER.ALL HRPWM_CAPBxCR.ALL HRPWM_CAPBxCER.ALL		
HRPWM 突发模式寄存器	HRPWM_BMPER.BMPER HRPWM_BMCMPR.BMCMP	HRPWM_BMCR.BMPREN	硬件更新（来自突发模式控制器）
HRPWM 通用寄存器	HRPWM_ADC0R.ALL HRPWM_ADC1R.ALL HRPWM_ADC2R.ALL HRPWM_ADC3R.ALL HRPWM_ADC4R.ALL HRPWM_ADC5R.ALL HRPWM_ADC0ER.ALL HRPWM_ADC1ER.ALL HRPWM_ADC2ER.ALL HRPWM_ADC3ER.ALL	硬件更新（来自主定时器或定时器x） 多个更新源可选，在HRPWM_CR0.USRCx中配置 （所选定时器x的PREEN = 1）	

主定时器有 5 个更新选项：

- 软件更新：将 1 写入 HRPWM_CR2.MSWU 会立即强制更新寄存器。在这种情况下，会取消任何待定的硬件更新请求。
- 复位事件更新：当主计数器复位或者主计数器翻转时，完成寄存器更新。主定时器默认使能此更新。
- 重复事件更新：当主计数器复位或者主计数器翻转，且主重复计数器等于 0 时，完成寄存器更新。当 HRPWM_MCR0.MREPU 置 1 时，会使能此更新。启用重复事件更新后复位事件更新失效。
- 突发 DMA 结束后进行更新：当 HRPWM_MCR0.BRSTDMA = 10 时使能此更新，可以同时设置 HRPWM_MCR0.MREPU = 1 和 HRPWM_MCR0.BRSTDMA = 10。
- 突发 DMA 结束后主计数器翻转时进行更新：当 HRPWM_MCR0.BRSTDMA = 11 时使能此更新。

主定时器更新事件可以生成中断请求或 DMA 请求。

各个定时器具有 8 个更新源：

- 软件更新：将 1 写入 HRPWM_CR2.SWUx 会立即强制更新寄存器。在这种情况下，会取消任何待定的硬件更新请求。
- 复位事件更新：当计数器复位或计数器翻转时，完成寄存器更新。当 HRPWM_PWMxCR0.UPDRST 置 1 时，会使能此更新。
- 重复事件更新：当计数器复位或计数器翻转，且重复计数器等于 0 时，完成寄存器更新。当 HRPWM_PWMxCR0.UPDREP 置 1 时，会使能此更新。启用重复事件更新后复位事件更新失效。
- 突发 DMA 结束后立即进行更新：当 HRPWM_PWMxCR0.UPDGAT = 1000 时使能此更新。

- 突发 DMA 结束后发生更新事件时进行更新
  （更新事件通过 HRPWM_PWMxCR0.UPDRST、HRPWM_PWMxCR0.UPDREP、HRPWM_PWMxCR1.MUPD 或 HRPWM_PWMxCR1.UPDx 使能）：当 HRPWM_PWMxCR0.UPDGAT = 1100 时使能此更新。
- 更新请求输入 0~2 上接收到事件时进行更新：当 HRPWM_PWMxCR0.UPDGAT = 0010、0011、0100 时使能此更新。
- 更新请求输入 0~2 上接收到事件后发生更新事件时进行更新
  （更新事件可通过 HRPWM_PWMxCR0.UPDRST、HRPWM_PWMxCR0.UPDREP、HRPWM_PWMxCR1.MUPD 或 HRPWM_PWMxCR1.UPDx 使能）：当 HRPWM_PWMxCR0.UPDGAT = 0101、0110、0111 时使能此更新。
- 硬件更新：其他定时器或主定时器更新时，完成寄存器更新（例如定时器 0 可与定时器 1 同步更新）。当 HRPWM_PWMxCR0.MUPD 或 HRPWM_PWMxCR1.UPDx 置 1 将使能此更新，适用于需要使用多个定时器的转换器。

更新请求输入 0~2 可使更新事件与来自通用定时器的事件同步，这些输入为上升沿有效。

表 17-16 列出了更新请求输入与片上来源的连接关系。

表 17-16　更新请求输入以及来源

hrpwm_upd_in	更新事件来源
hrpwm_upd_in0	TMR1_OC0
hrpwm_upd_in1	TMR2_OC0
hrpwm_upd_in2	TMR7_TRGO

这样可将低频更新请求与高频信号同步，例如可以使用一个慢速事件更新一个快速定时器。

注：HRPWM_MCR0.CKPSC 或 HRPWM_PWMxCR0.CKPSC > 5 时，更新事件会同步到预分频器时钟

来自邻近定时器的更新（当 HRPWM_PWMxCR1.MUPD，HRPWM_PWMxCR1.UPDx 置 1 时）或来自软件更新的更新（HRPWM_CR2.SWUx）都可以立即发生，或者与定时器的复位/翻转事件同步发生。这通过 HRPWM_PWMxCR0.RSYNCU 实现，如下图 17-51 所示：

- HRPWM_PWMxCR0.RSYNCU = 0：来自邻近定时器的更新立即发生
- HRPWM_PWMxCR0.RSYNCU = 1：来自邻近定时器的更新在下一复位/翻转事件处发生。

定时器更新事件可以产生中断请求或 DMA 请求。

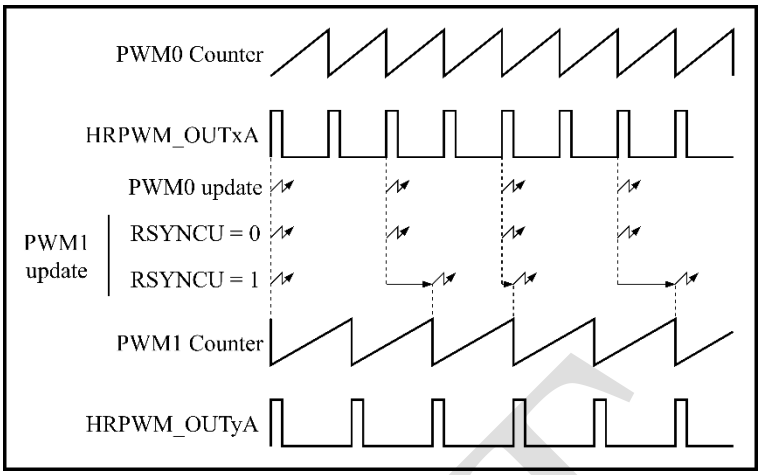


图 17-51　定时器的同步更新（PWM1CR0 的 UPD0=1）

无论选择哪一种更新事件，HRPWM_CR0.MUDIS 和 HRPWM_CR0.UDISx 都可以暂时禁止将预加载寄存器的内容传输至有效寄存器。这样便可以修改并联的定时器中的多个寄存器。这些位复位后，会立即产生常规更新事件。

HRPWM_CR0.MUDIS 和 HRPWM_CR0.UDISx 位全部分组到同一寄存器中。这样可以同时禁止和恢复更新并联工作的定时器（不需要同步）。

下例为实际用例。第一个电源转换器通过主定时器，定时器 1 和定时器 2 控制。定时器 1 和定时器 2 必须通过主定时器重复事件更新。第二个转换器需要与定时器 0，定时器 3 和定时器 4 同时工作，定时器 3，定时器 4 必须通过定时器 0 重复事件更新。

第一个转换器

若 HRPWM_MCR0.MREPU 置 1，将在主定时器计数器重复周期结束时进行更新。PWM1CR1.MUPD 和 PWM2CR0.MUPD 必须置 1，才能通过主定时器同时更新定时器 1 和定时器 2。

如果必须通过软件调整电源转换器的设定值，则必须在对寄存器执行写访问更新数值之前，将 HRPWM_CR0.MUDIS，HRPWM_CR0.UDIS1 和 HRPWM_CR0.UDIS2 置 1。从此刻起，会忽略任何硬件更新请求，并可无风险地访问预加载寄存器，不会将其中的内容传输到有效寄存器中。软件处理结束后，必须将 HRPWM_CR0.MUDIS，HRPWM_CR0.UDIS1 和 HRPWM_CR0.UDIS2 置 0：发生主定时器重复事件后，会立即将预加载寄存器的内容传输到有效寄存器。

第二个功率转换器

若 PWM0CR0.UPDREP 置 1,将在定时器 0 计数器重复周期结束时进行更新。PWM3CR1.UPD0 和 PWM4CR1.UPD0 必须置 1，才能通过定时器 0 同时更新定时器 3 和定时器 4。

如果必须通过软件调整电源转换器的设定值，则必须在对寄存器执行写访问更新数值之前，将 HRPWM_CR0.UDIS0，HRPWM_CR0.UDIS3 和 HRPWM_CR0.UDIS4 置 1。从此刻起，会忽略任何硬件更新请求，并可无风险地访问预加载寄存器，不会将其中的内容传输到有效寄

存器中。软件处理结束后，必须将 HRPWM_CR0.UDIS0，HRPWM_CR0.UDIS3 和 HRPWM_CR0.UDIS4 置 0：发生定时器 0 重复事件后，会立即将预加载寄存器的内容传输到有效寄存器。

17.4.12 带有大于比较的 PWM 模式

HRPWM 支持一种无延迟更新模式、可用于 CMPA 和 CMPC 寄存器以产生 PWM 输出。这种模式不必等待当前 PWM 周期的完成，可以在当前 PWM 周期内应用新的占空比值，从而减少软件控制循环的整体延迟时间，如下图所示：

- 如果新的比较值在当前计数值之下，同时旧的比较值在当前计数值之上，此时写入新的比较值的时候，输出信号提前关断。
- 如果新的比较值在当前计数值之上，同时旧的比较值在当前计数值之下，此时写入新的比较值的时候，输出信号提前开通。
- 如果新的比较值与旧的比较值都在当前计数值之下，输出信号保持不变。

该模式仅对 CMPA 或 CMPC 的 CLR 事件有效，并通过 HRPWM_PWMxCR0.GTCMPA 和 HRPWM_PWMxCR0.GTCMPC 使能。

比较寄存器对应的 HRPWM_PWMxCR0.GTCMPx 置 1 时，无论 HRPWM_PWMxCR0.PREEN 值是多少，预加载机制都无效。该模式希望新的比较值在写入后尽快生效，不用等待预加载寄存器更新。

比较 A 和比较 C 操作模式如下所示：

- HRPWM_PWMxCR0.GTCMPx = 0:当计数值等于比较值时，产生比较事件(等于比较模式)。如果比较值动态更改，则可能不会生成比较事件。
- HRPWM_PWMxCR0.GTCMPx = 1:当计数器大于比较值时，产生比较事件(大于比较模式)。如果比较值动态更改，则新的比较值与当前计数值和进行比较，可以生成输出 SET 或 CLR 事件。

大于比较模式使得 PWM 输出根据比较结果的不同而不同。假设 CMPA 事件配置为输出 CLR 动作，当新的比较值写入时，有以下两种情况：

- 如果新的比较值在当前计数值之下，则产生 CLR 事件并最终使得输出提前关断
- 如果新的比较值在当前计数值之上，则产生 SET 事件并最终使得输出提前开通
- 大于比较模式支持 SET 和 CLR 两种动作

大于比较模式主要用于以下情况：

1. 对于固定频率配置，周期事件触发输出置位，比较事件触发输出复位；反之亦可，周期事件使得输出复位，比较使得输出置位。

2. 对于可变频率配置，用来复位计数器的事件也必须用来置位或复位定时单元输出，并且与比较事件置位或复位的方向相反。

注：大于比较模式使用时，不能使用由硬件控制的 CMPA 和/或 CMPC 半模式/交错模式。

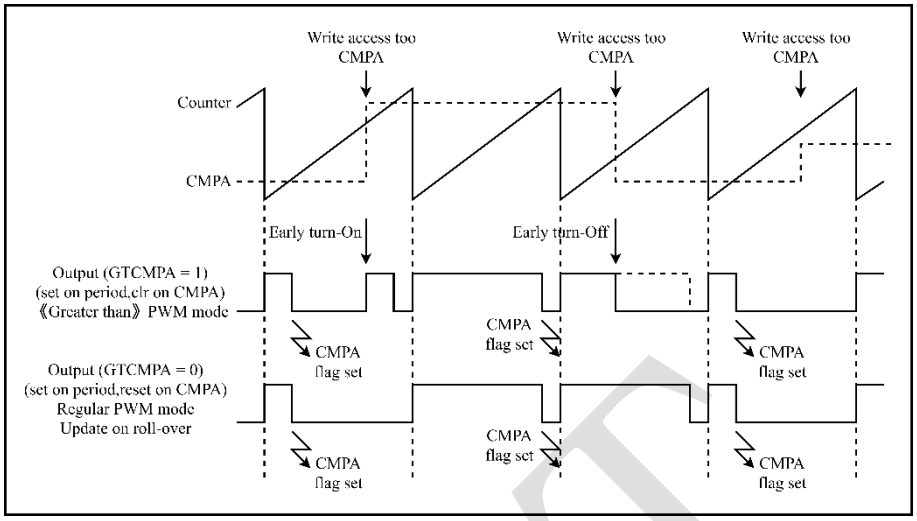


图 17-52　大于比较模式的提前开通和提前关断动作

无论 HRPWM_PWMxCR0.PREEN 值是多少，当 HRPWM_PWMxCR0.GTCMPA 和/或 HRPWM_PWMxCR0.GTCMPC 置 1 时，它们各自的预加载机制被禁用（对应于 HRPWM_CMPAxR 和/或 HRPWM_CMPCxR 寄存器）。

注：比较中断标志（HRPWM_PWMxISR.CMPA 和 HRPWM_PWMxISR.CMPC）在提前开启/提前关断的情况下不会产生，如图 17-52 所示。

注：不能在单个输出上同时应用 CMPA 和 CMPC 的大于比较模式（HRPWM_PWMxCR0.GTCMPA 和 HRPWM_PWMxCR0.GTCMPC 不能同时置 1）。

17.4.13　输出管理

每个定时单元都控制着一对输出，输出有三种工作状态：

- RUN：主要工作状态，在软件启动输出后进入 RUN 状态，在 RUN 模式下，输出根据 CrossBar 的设置变为有效电平或无效电平。
- IDLE：初始工作状态，在复位释放或软件停止输出后进入 IDLE 状态，在 IDLE 模式下，输出维持有效电平或者无效电平。
- FAULT：安全工作状态，在 HRPWM_OUTxR.FAULTx 故障信号产生后进入 FAULT 状态，在 FAULT 模式下，输出维持有效电平、无效电平或高阻态。

输出状态由 HRPWM_OENR.OENxy 和 HRPWM_ODISR.ODISxy 指示，如表 17-17 所示。

表 17-17　输出状态编程 x = 0~7 , y= A~B

OENxy（控制位/状态位） （由软件设置，由硬件清除）	TxyODIS（控制位/状态位） （由软件设置，由硬件清除）	输出工作状态
1	X	运行
0	0	空闲
0	1	故障

HRPWM_OENR.OENxy 和 HRPWM_ODISR.ODISxy 既是控制位又是状态位：

HRPWM_OENR.OENxy 必须通过软件置 1，使输出进入 RUN 模式。输出恢复到 IDLE 或 FAULT 模式时，HRPWM_OENR.OENxy 会通过硬件清 0。HRPWM_ODISR.ODISxy 必须通过软件置 1，使输出返回到 IDLE 模式。HRPWM_OENR.OENxy 为 0 时，HRPWM_ODISR.ODISxy 会指示输出是处于 IDLE 还是 FAULT 状态。

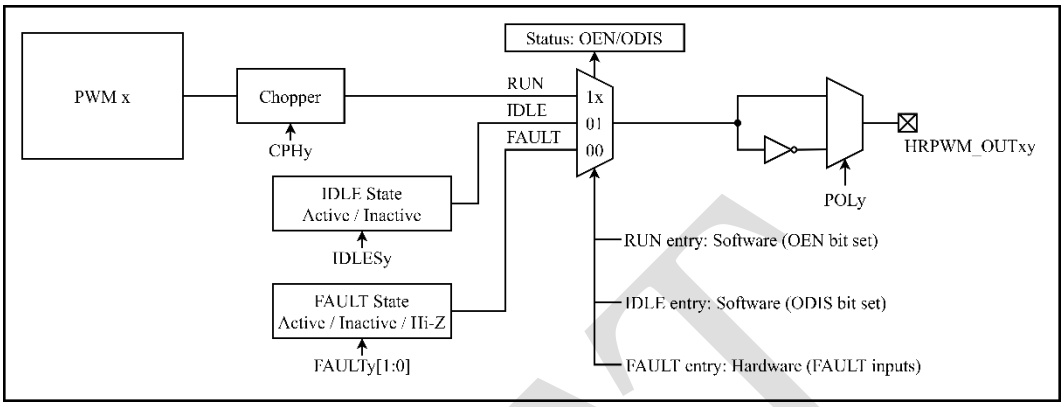


图 17-53　输出管理概览

图 17-54 总结了三种状态对应的位值以及转换的触发方式。任何外部或内部故障源均可触发故障。

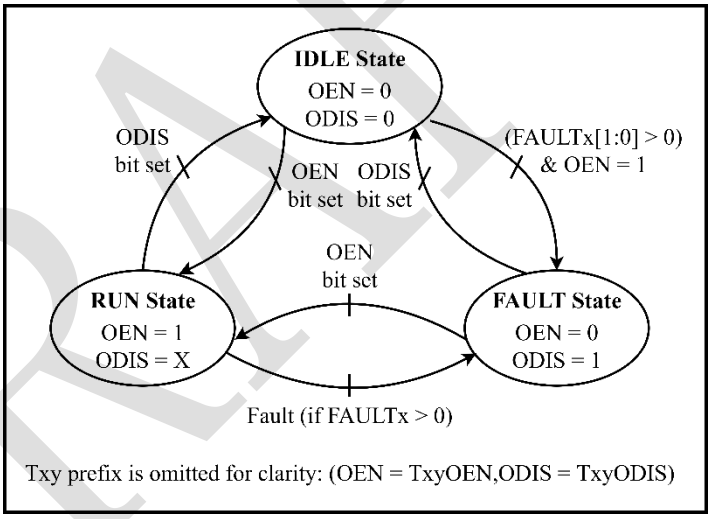


图 17-54　HRPWM 输出状态与转换图

FAULT 和 IDLE 电平定义为有效电平或无效电平。有效电平是指 PWM 输出中引发电源开关闭合的电平，无效电平是指 PWM 输出中引发电源开关关断的电平。

IDLE 状态具有最高优先级：即使 FAULT 条件仍然有效，依然可以实现将 HRPWM_ODISR.ODISxy 位置 1 触发的 FAULT→IDLE 转换。

FAULT 状态的优先级高于 RUN 状态：如果将 HRPWM_OENR.OENxy 位置 1 的同时发生了故障事件，则将进入 FAULT 状态。IDLE→FAULT 转换中给出了相应的条件：需要使能故障保护（HRPWM_OUTxR.FAULTx [1:0]位= 01、10、11），且 HRPWM_OENR.OENxy 在故障有效时置 1，此时由 IDLE 状态进入 FAULT 状态，如图 44 所示。

输出极性可以使用 HRPWM_OUTxR.POLx 设置。如果 HRPWM_OUTxR.POLx = 0，极性为正（输出高电平有效），如果 HRPWM_OUTxR.POLx = 1，极性为负（输出低电平有效）。实际上，极性的定义取决于要驱动的电源开关（PMOS 与 NMOS）或栅级驱动器的极性。

每路输出 FAULT 状态下的输出电平通过 HRPWM_OUTxR.FAULTx 配置，具体如下：

- 00：输出不会进入 FAULT 状态（保持在 RUN 或 IDLE 状态）
- 01：输出在 FAULT 状态下呈现为有效电平
- 10：输出在 FAULT 状态下呈现为无效电平
- 11：输出在 FAULT 状态下呈现为高阻态（浮空状态）。必须通过上拉或下拉电阻在外部强制进入安全状态。

注：一旦输出处于 FAULT 状态，不得更改 HRPWM_OUTxR.FAULTx 位。

每路输出 IDLE 状态下的输出电平通过 HRPWM_OUTxR.IDLESx 配置，具体如下：

- 0：输出在 IDLE 状态下呈现为无效电平
- 1：输出在 IDLE 状态下呈现为有效电平

当 HRPWM_OENR.OENxy 置 1 进入 RUN 状态时，输出会立即连接到交错配置矩阵输出。如果定时器时钟停止，电平将为无效电平（HRPWM 复位后）或保持之前电平不变（定时器停止、且输出禁止时）。

在 HRPWM 初始化过程中，在使输出进入 RUN 模式之前，可在 HRPWM_SETxyR 和 HRPWM_RSTxyR 寄存器中使用软件强制输出置位和复位预先设定输出电平。

17.4.14 突发模式控制器

突发模式控制器使输出交替地进入 IDLE 状态和 RUN 状态，从而跳过一些开关周期，其中周期和占空比可编程。

突发模式运行常用于轻载运行的电源转换器中，突发模式减少了输出变化次数以及相关的开关损耗，从而显著提高转换器效率。

突发模式下运行时，会在空闲周期（长度等于几个计数周期）后输出一个或几个脉冲，空闲周期内不会生成任何输出脉冲，如图 17-55 所示。

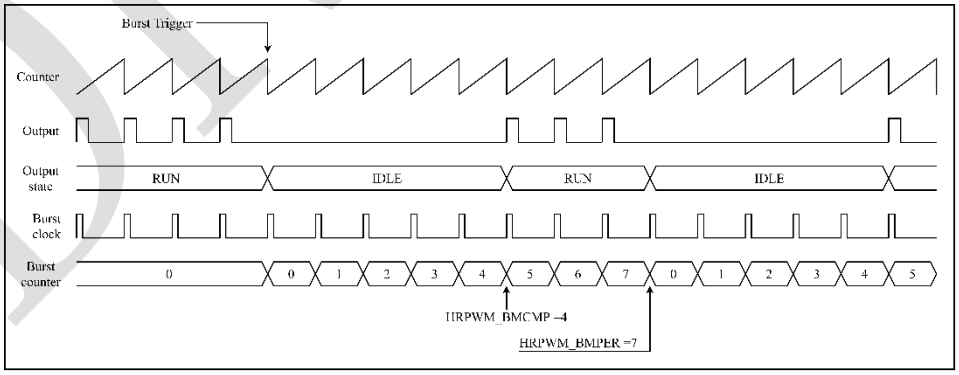


图 17-55 突发模式运行示例

突发模式控制器包括：

- 计数器：支持各种来源作为时钟，来源于 HRPWM 之内或之外，通常是 PWM 周期事件。
- 比较寄存器（HRPWM_BMCMP）：定义空闲周期。
- 周期寄存器（HRPWM_BMPER）：定义总突发模式周期，也即空闲周期与运行周期之和。

突发模式控制器能够接管控制 16 路 PWM 输出中的任何一路。突发模式工作期间，每路输出

的状态通过 HRPWM_OUTxR.IDLESx 和 HRPWM_OUTxR.IDLEMx 编程，如表 17-18 所述。

表 17-18　突发模式下的输出状态

IDLEMx	IDLESx	突发模式下输出状态
0	X	无动作：输出不受突发模式动作影响
1	0	突发模式下输出无效电平
1	1	突发模式下输出有效电平

注：突发模式有效时，不得更改 HRPWM_OUTxR.IDLEMx。

突发模式控制器仅作用于输出级。空闲周期内仍会生成一些事件：

- 输出置位/复位中断以及 DMA 请求
- 由输出置位/复位触发的捕获事件

突发模式期间，即使 BME 位已置 1，SCOUT 输出上也不会生成启动事件或复位事件。

工作模式

在使用突发模式之前，必须使能定时单元计数器（HRPWM_MCR1.CENx 置 1）。突发模式通过 HRPWM_BMCR.BME 使能。

通过配置 HRPWM_BMCR.BMOM，突发模式可以选择在连续模式或单发模式下运行。

配置 HRPWM_BMCR.BMOM = 1 使能连续模式，在对 HRPWM_BMCR.BMEXIT 写 1 终止突发模式运行之前，都会保持突发模式运行状态。

配置HRPWM_BMCR.BMOM = 0 使能单次模式，在突发模式触发事件之后执行一次空闲序列，随后会立即恢复正常定时单元操作。

空闲周期和运行周期的持续时间由突发模式计数器和 2 个寄存器来定义。 HRPWM_BMCMP 寄存器定义了定时单元处于空闲状态的持续时间，即空闲周期。HRPWM_BMPER 寄存器定义总突发模式周期，即空闲周期与运行周期之和。初始突发模式触发事件之后，空闲周期长度为 BMCMPR+1，总突发周期为 BMPER+1。

注：突发模式周期不得小于或等于通过 HRPWM_DTxR.DTR 和 HRPWM_DTxR.DTF 位域定义的死区持续时间。

突发模式工作期间，定时单元和主定时器的计数器支持暂停和恢复。HRPWM_BMCR 使用了 9 个控制位来实现此目的：MBMDIS（主定时器）和 BMDIS0~BMDIS7（用于定时器 x）。

如果 HRPWM_BMCR.MBMDIS 或 HRPWM_BMCR.BMDISx 置 0，会保留计数器时钟。

如果 HRPWM_BMCR.MBMDIS 或 HRPWM_BMCR.BMDISx 置 1，则在突发空闲周期期间，相应的计数器会暂停，并会保持在当前状态下。这样便可使定时器在退出空闲状态时重新恢复一个完整的周期。

注：当均衡空闲模式使能时(HRPWM_PWMxISR.DLYPRT=0x11)，不得将 HRPWM_BMCR.BMDISx 置 1。

突发模式时钟

突发模式计数器可由多个时钟源提供时钟，通过 HRPWM_BMCR.BMCLK 来选择：

- HRPWM_BMCR.BMCLK = 0000 到 1000
  主定时器和定时器 x 复位/翻转事件。这样可使突发模式空闲周期和运行周期与定时单元计数周期对齐，无论处于自由运行和计数器复位模式。
- HRPWM_BMCR.BMCLK = 1100 到 1110
  时钟由通用定时器通过 hrpwm_bst_in 提供，如下表所示。在这种情况下，突发模式空闲周期和运行周期不必与定时单元计数周期对齐，输出脉冲可能中断，从而导致波形的占空比被修改。
- HRPWM_BMCR.BMCLK=1111
  使用预分频后 FHRPWM 时钟，通过 HRPWM_BMCR.BMPRSC 定义预分频系数。在这种情况下，突发模式空闲周期和运行周期不必与定时单元计数周期对齐，输出脉冲可能中断，从而导致波形的占空比被修改。

表 17-19　来自通用定时器的突发模式时钟来源

hrpwm_bst_in	BMCLK[3:0]	突发模式时钟
hrpwm_bst_in0	1100	TMR1_OC0
hrpwm_bst_in1	1101	TMR2_OC0
hrpwm_bst_in2	1110	TMR8_TRGO

TMRx_OC 上的脉冲宽度必须至少为 N 个 FHRPWM 时钟周期才能被 HRPWM 突发模式检测到。

突发模式触发事件

可以使用 40 个触发事件启动突发模式工作，事件通过 HRPWM_BMTRGRx 寄存器选择：

- 软件触发事件
- 主定时器事件：重复、复位/翻转、比较 A 到 D
- 定时器 x 事件：重复、复位/翻转、比较 A 到 B
- HRPWM_Event7（包含 PWM0 事件过滤）和 HRPWM_Event8（包含 PWM4 事件过滤）
- HRPWM_Event7 之后的定时器 0 周期（包含 PWM0 事件过滤）
- HRPWM_Event8 之后的定时器 4 周期（包含 PWM4 事件过滤）
- 来自其他通用定时器的片上事件（hrpwm_bst_in 输入：TIMx_TRGO 输出）

这些触发事件可结合起来实现多事件并行触发。

突发模式不可再次被触发：在连续模式下，在突发模式终止之前会忽略新的触发事件；在单次模式下，在包含当前运行周期的突发结束之前（BMPER+1 个周期），会忽略触发事件。

图 17-56 显示了突发模式是如何响应外部事件启动的，可以立即启动也可以在外部事件后的定时器周期启动。

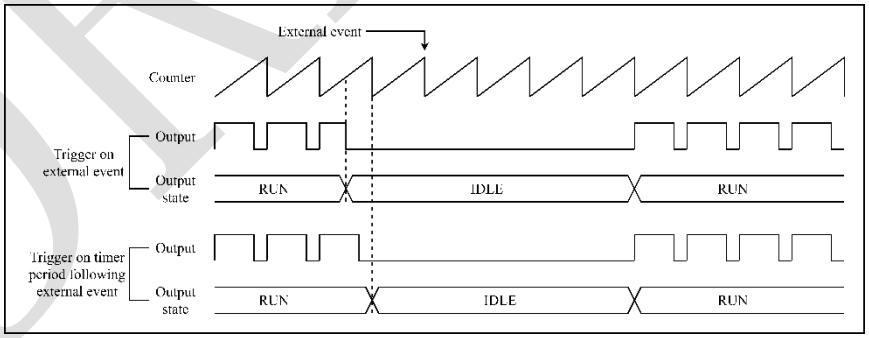


图 17-56　外部事件触发突发模式

对于 Event7 和 Event8 组合触发（在外部事件后的定时器周期触发），无论采用何种突发模式编程以及正在进行何种突发操作，外部事件检测始终有效：

- 如果在外部事件与定时器周期事件之间突发模式已使能（BME=1）或触发已使能（HRPWM_BMTRGR0.PER0EEV7 或 HRPWM_BMTRGR0.PER4EEV8 置 1），会触发突发模式。
- 即使外部事件在突发结束之前发生，也会再次触发单次突发模式，只要突发之后发生相应的周期事件。

注：HRPWM_BMTRGR0.PER0EEV7 和 HRPWM_BMTRGR0.PER4EEV8 触发仅在周期事件后

有效。如果计数器在周期事件之前复位，则会丢弃挂起的 HRPWM_Event7/8 事件。

突发模式延迟进入

默认情况下输出将在突发模式触发事件后立即进入空闲电平，空闲电平取决于 HRPWM_OUTxR.IDLESx 设置。

输出也可以延迟进入突发模式，并在输出进入空闲状态之前的可配置周期内将输出强制为无效状态。驱动两路互补输出，一路输出空闲状态为有效电平时，可利用此特性避免死区违反，如图 17-57 所示。

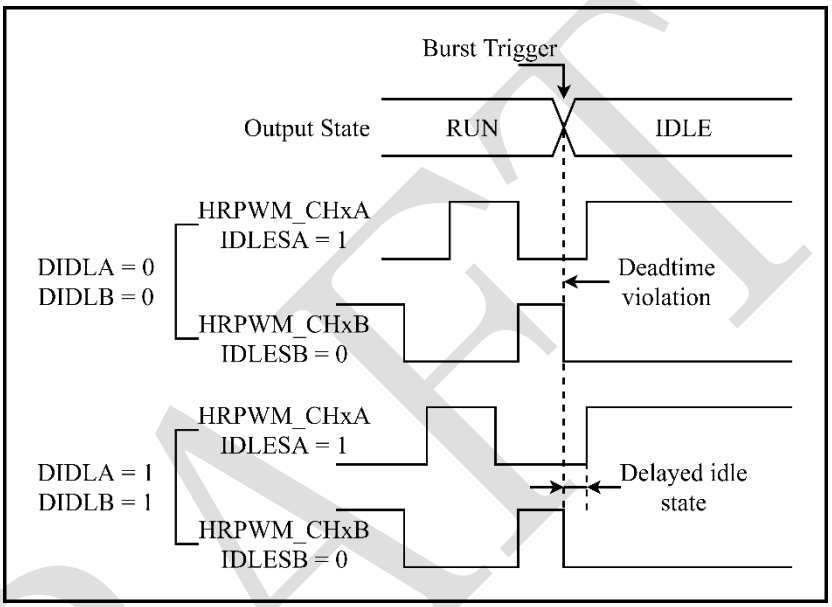


图 17-57　死区使能且 IDLESx = 1 时的突发模式延迟进入

突发模式延迟进入通过 HRPWM_OUTxR.DIDLx 使能，每路输出有单独的使能位。

该模式会在输出获取其空闲状态之前强制插入死区。每路 PWMx 输出都有自己的死区值：

- HRPWM_OUTxR.DIDLA = 1 时，输出 A 对应 HRPWM_DTxR.DTR
- HRPWM_OUTxR.DIDLB = 1 时，输出 B 对应 HRPWM_DTxR.DTF

仅当其中一路输出在突发模式期间具有有效空闲电平（HRPWM_OUTxR.IDLESx =1）、且使用死区为正死区（HRPWM_DTxR.SDTR/ HRPWM_DTxR.SDTF 置 0）时，才可将 HRPWM_OUTxR.DIDLx 置 1。

如果由 HRPWM_DTxR.DTR 和 HRPWM_DTxR.DTF 定义的死区持续时间小于 3 个 FHRPWM 时钟周期，则必须注意与窄脉冲相关的限制。

突发模式延迟进入使用死区功能实现。如果突发模式进入在常规死区内出现，死区会中止并会重新开始新的死区，对应于无效周期，如图 17-58 所示。

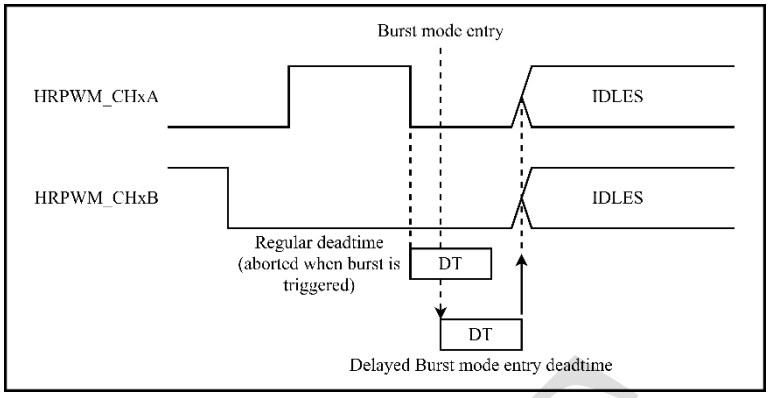


图 17-58　死区期间的突发模式延迟进入

突发模式退出

突发模式在连续模式下需要软件强制退出，在单发模式下经过空闲周期到达后退出。两种情况下，计数器都会立即恢复（如果计数器通过 HRPWM_BMCR.MBMDIS 或 HRPWM_BMCR.BMDISx 置 1 保持在暂停状态），但输出状态仅会在置位/复位事件后才会从空闲模式有效跳变为工作模式。

如果 HRPWM_IER.BMPERIE 置 1，则会在单次模式和连续模式下生成突发周期中断。该中断可用于在连续模式下将突发模式退出与突发周期进行同步。

图 17-59 显示了死区使能后是如何恢复正常工作的。突发模式退出是立即进行的，但仅对任何互补输出上的第一个置位事件有效。

图中显示了两种不同情况：

1.输出波形上的信号无效时，突发模式结束：OutA 上发生置位事件时会恢复有效状态，OutB 输出不会在退出突发模式时获取互补电平。

2.输出波形上的信号有效时，突发模式结束：OutB 上发生置位事件时会恢复有效状态，OutA 不会在退出突发模式时立即获取有效电平。

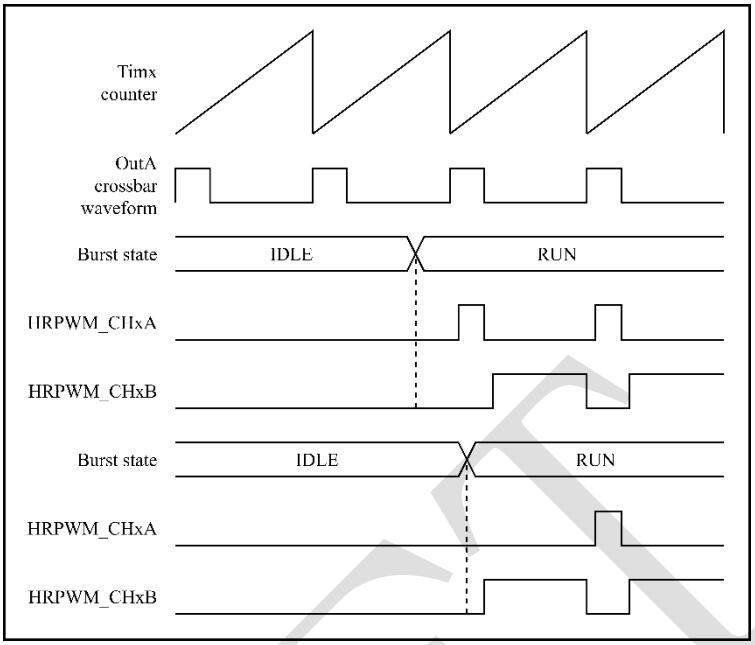


图 17-59 死区使能时的突发模式退出

如果使能了推挽模式，上述行为会略有不同。如果输出无效，推挽模式会在周期开始时强制进行输出复位，如果前一周期内输出为高电平，则会对称地将输出强制设置为有效电平。

突发模式寄存器预加载和更新

HRPWM_BMCR.BMPREN 为突发模式预加载使能，设置突发模式比较寄存器和周期寄存器（HRPWM_BMCMP 和 HRPWM_BMPER）是否支持预加载。

当 HRPWM_BMCR.BMPREN 置 1 时，预加载寄存器在以下事件处更新至有效寄存器：

- 突发模式使能时 (HRPWM_BMCR.BME= 1)
- 突发模式周期结束时

如果将 HRPWM_CR0.BMUDIS 置 1，则会暂时禁止 HRPWM_BMPER 和 HRPWM_BMCMP 的更新，这样可以确保两个寄存器在进行修改时同步生效。

17.4.15 斩波器

可以在定时单元输出信号上添加高频载波（斩波），以驱动隔离变压器。斩波是在插入极性之前对输出级执行的，如下图所示，使用 HRPWM_OUTxR.CHPA 和 HRPWM_OUTxR.CHPB 分别在 OUTA 和 OUTB 上使能斩波。

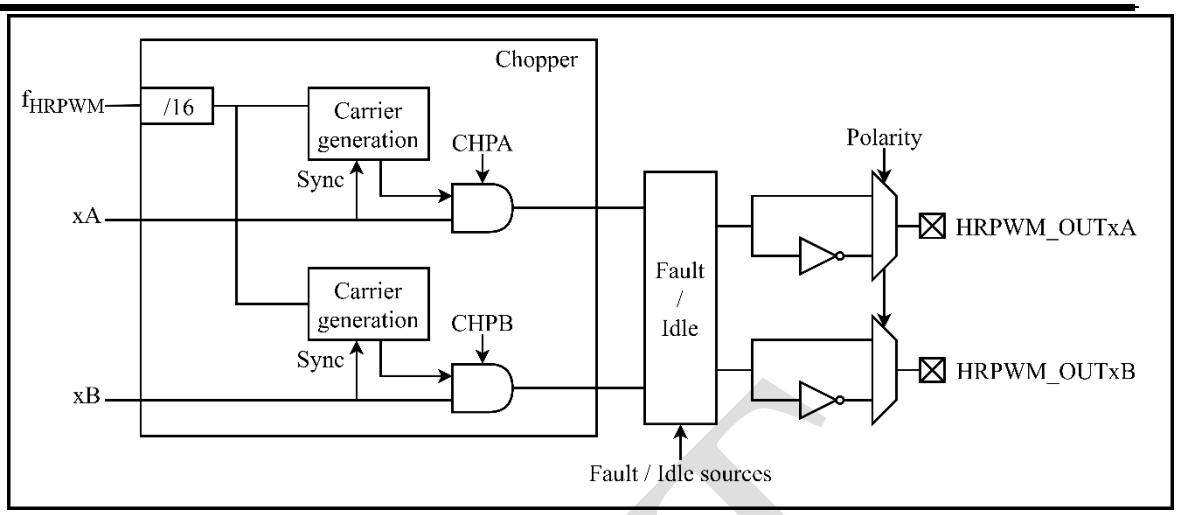


图 17-60　载波频率信号插入

使用 HRPWM_CHPxR 寄存器调整斩波参数，以在斩波开始处定义一个特定脉宽，后跟频率和占空比可编程的载频信号，如图 17-61 所示。

载波频率通过 HRPWM_CHPxR.CARFRQ 定义，公式 F_{CHPFRQ} = FHRPWM /（16×（CARFRQ [3:0] +1））， 范围从 703.1kHz 到 11.25MHz（对于 FHRPWM = 180 MHz）。

载波占空比可以通过 HRPWM_CHPxR.CARDTY 进行调节（步长为 1/8），占空比范围为 0/8 到 7/8。HRPWM_CHPxR.CARDTY = 000（占空比= 0/8）时，输出波形仅包含参考波形上升沿之后的启动脉冲，不会添加任何载波。

初始脉冲的脉宽通过 HRPWM_CHPxR.STRPW 定义，具体如下：t_{1STPW} =（STRPW [3:0] +1）×16×THRPWM，范围从 88.9 ns 到 1.42 µs（对于 FHRPWM = 180 MHz）。

载波频率参数是根据 FHRPWM 频率定义的，与 HRPWM_MCR0.CKPSC 和 HRPWM_PWMxCR0.CKPSC 设置无关。

在斩波模式下，载波频率和初始脉宽会通过逻辑与函数与参考波形相结合。初始脉冲结束时会执行同步，以获得重复的信号波形。

斩波信号在输出波形无效状态结束时停止，不会等到当前载波周期结束。因此，斩波信号包含的脉冲可能比编程的脉冲短。

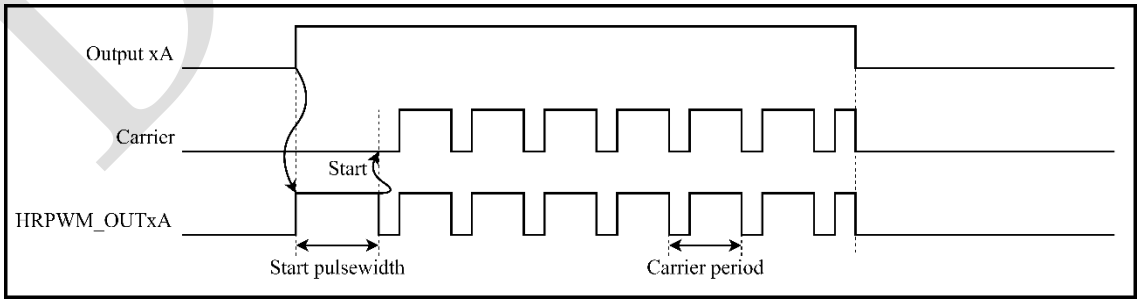


图 17-61　使能斩波模式的 HRPWM 输出

注：必须先将 HRPWM_OUTxR.CHPx 置 1，再使能 HRPWM_OENR.OENxy 使能输出。

斩波模式有效时（两个 HRPWM_OUTxR.CHPx 中至少一个置 1），不能修改 HRPWM_CHPxR.CARFRQ，HRPWM_CHPxR.CARDTY 和 HRPWM_CHPxR.STRPW。

17.4.16 故障保护

HRPWM 具有通用的故障保护电路，可在发生异常操作时禁止输出。一旦触发故障，输出便会进入预定义的安全状态。输出通过软件重新使能之前，都会保持该状态。如果是永久故障请求，即使软件尝试重新使能输出，输出也将保持其故障状态，直至故障源消失。

HRPWM 有 8 个故障输入通道，所有通道均可用，并可结合起来用于 8 个定时单元中的每一个，如图 17-62 所示。

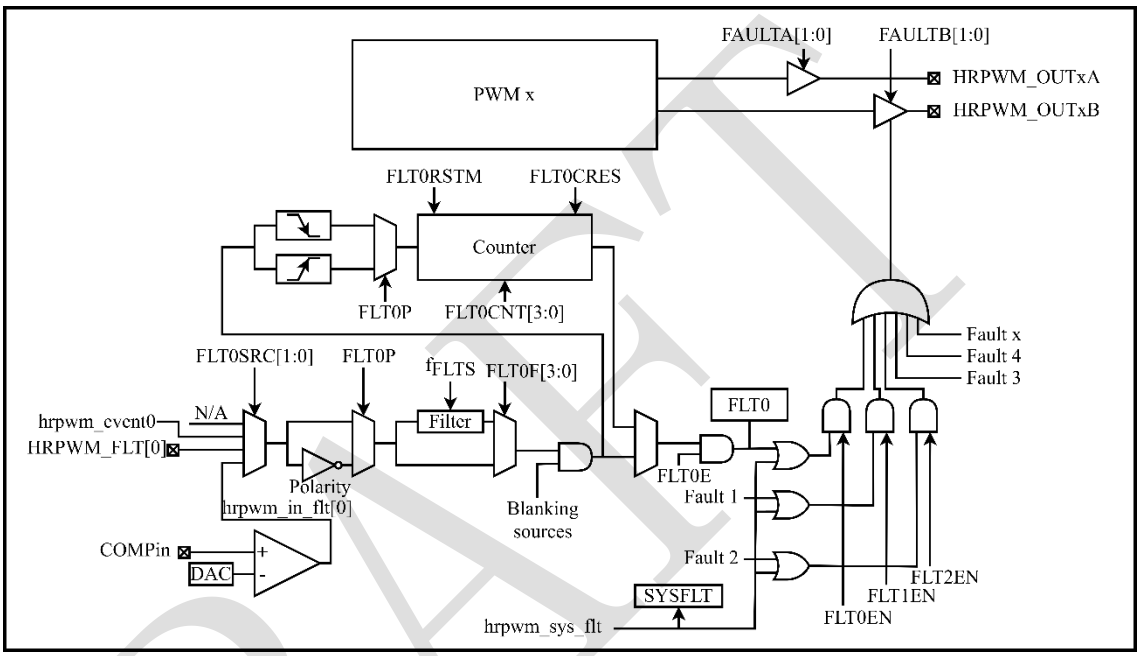


图 17-62　故障保护电路概览（显示了 1 条通道）

在连接到定时单元之前，每条故障通道均可完全通过 HRPWM_FLTINR0、HRPWM_FLTINR1、HRPWM_FLTINR2 和 HRPWM_FLTINR3 寄存器进行配置。HRPWM_FLTINR0.FLTxSRC 用于选择故障信号源，可以是数字输入或内部事件（内置比较器输出）。

表 17-20 总结了可用于 8 个故障通道的源。

表 17-20　故障事件输入来源

故障事件	FLTxSRC[1:0] = 00	FLTxSRC[1:0] = 01	FLTxSRC[1:0] = 10	FLTxSRC[1:0] = 11
hrpwm_flt_in0	HRPWM_FLT0	CMP1_OUT	HRPWM_Event0	N/A
hrpwm_flt_in1	HRPWM_FLT1	CMP3_OUT	HRPWM_Event1	N/A
hrpwm_flt_in2	HRPWM_FLT2	CMP5_OUT	HRPWM_Event2	N/A
hrpwm_flt_in3	HRPWM_FLT3	CMP0_OUT	HRPWM_Event3	N/A
hrpwm_flt_in4	HRPWM_FLT4	CMP2_OUT	HRPWM_Event4	N/A
hrpwm_flt_in5	HRPWM_FLT5	CMP4_OUT	HRPWM_Event5	N/A
hrpwm_flt_in6	HRPWM_FLT6	CMP7_OUT	HRPWM_Event6	N/A
hrpwm_flt_in7	HRPWM_FLT7	CMP8_OUT	HRPWM_Event7	N/A

上表 17-16 中提到的 HRPWM_Eventx 事件，是由 hrpwm_ext_evt[3:0]输入多路复用器决定（受 HRPWM_FLTINR0.FLTxSRC 控制），有关详细信息见图 17-26。

可通过 HRPWM_FLTINR0.FLTxP 极性位选择信号的极性，以定义有效电平。

- 如果 HRPWM_FLTINR0.FLTxP = 0，信号高电平有效；
- 如果 HRPWM_FLTINR0.FLTxP = 1，信号低电平有效。

可在极性设置后过滤故障信息。如果 HRPWM_FLTINR1.FLTxF 设为 0000，则不会对信号进行滤波，信号将独立于 FHRPWM 时钟异步操作。对于所有其他当 HRPWM_FLTINR1.FLTxF 值，会对信号进行数字滤波。数字滤波器由计数器组成，需要使用 N 个有效采样来验证输出跳变。如果输入值在计数器达到 N 之前发生变化，计数器会复位，跳变会被丢弃（视为伪事件）。如果计数器达到 N，跳变被视为有效，并会作为正确的外部事件进行传输。因此，数字滤波会向正在进行滤波的故障信号添加延迟，延迟时长视采样时钟和滤波器长度（预期的有效采样数）而定。图 17-63 显示了伪故障信号的滤波方式。

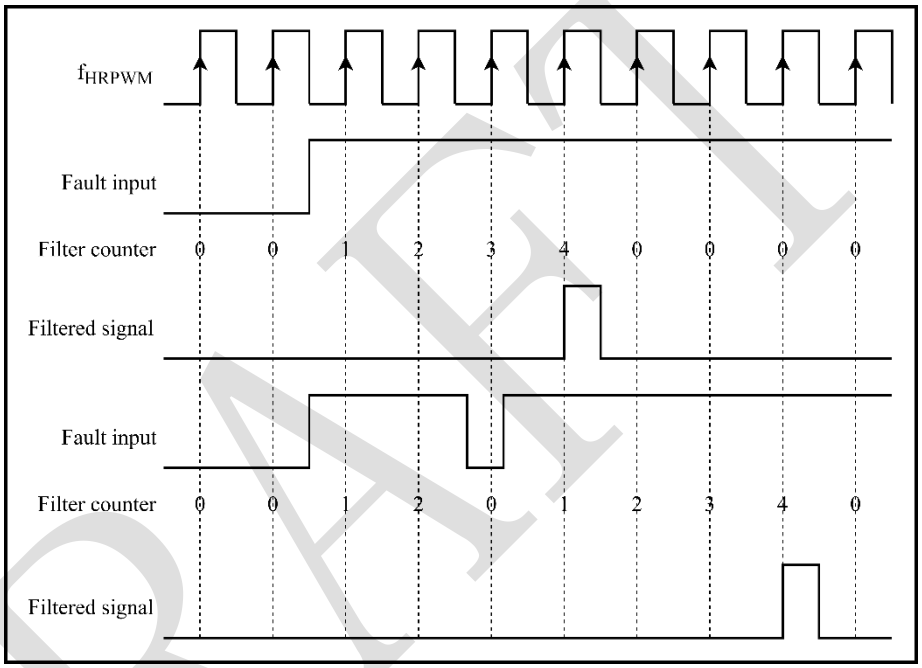


图 17-63　故障信号滤波（FLTxF[3:0] = 0010：fSAMPLING = FHRPWM，N = 4）

滤波周期从 2 个 FHRPWM 时钟周期到 8 个 32 分频的 f_{FLTS} 时钟周期。f_{FLTS} 通过 HRPWM_FLTINR1.FLTSD 定义。

表 17-21 总结了采样速率和滤波长度。考虑到采样引起的不确定性，必须将滤波器长度减去 1 个采样时钟周期的抖动，从而实现有效滤波。

表 17-21　采样速率和滤波长度与 FLTFxF [3:0] 和时钟设置

-	f_{FLTS} vs FLTSD[1:0]				滤波长度 FHRPWM = 160MHz	
FLTxF[3:0]	00	01	10	11	Min	Max
0001	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=6.25ns	N=50ns
0010	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=12.5ns	N=100ns
0011	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=18.75ns	N=150ns
0100	FHRPWM	FHRPWM/	FHRPWM/	FHRPWM/	N=25ns	N=200ns

		2	4	8		
0101	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=31.25ns	N=250ns
0110	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=37.5ns	N=300ns
0111	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=43.75ns	N=350ns
1000	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=50ns	N=400ns
1001	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=56.25ns	N=450ns
1010	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=62.5ns	N=500ns
1011	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=68.75ns	N=550ns
1100	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=75ns	N=600ns
1101	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=81.25ns	N=650ns
1110	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=87.5ns	N=700ns
1111	FHRPWM	FHRPWM/2	FHRPWM/4	FHRPWM/8	N=93.75ns	N=750ns

故障消隐与事件计数

故障输入可以被暂时禁止以消隐伪故障事件，消隐源列在下表 17-22 中。

表 17-22　故障输入消隐事件

-	FLTxBLKS = 0，固定窗口		FLTxBLKS =1，移动窗口	
故障事件	消隐窗口开始	消隐窗口停止	消隐窗口开始	消隐窗口停止
hrpwm_flt_in0[1:0]	PWM0 复位/翻转事件	PWM0 CMPC 事件	PWM0 CMPD 事件	PWM0 CMPC 事件
hrpwm_flt_in1[1:0]	PWM1 复位/翻转事件	PWM1 CMPC 事件	PWM1 CMPD 事件	PWM1 CMPC 事件
hrpwm_flt_in2[1:0]	PWM2 复位/翻转事件	PWM2 CMPC 事件	PWM2 CMPD 事件	PWM2 CMPC 事件
hrpwm_flt_in3[1:0]	PWM3 复位/翻转事件	PWM3 CMPC 事件	PWM3 CMPD 事件	PWM3 CMPC 事件
hrpwm_flt_in4[1:0]	PWM4 复位/翻转事件	PWM4 CMPC 事件	PWM4 CMPD 事件	PWM4 CMPC 事件
hrpwm_flt_in5[1:0]	PWM5 复位/翻转事件	PWM5 CMPC 事件	PWM5 CMPD 事件	PWM5 CMPC 事件
hrpwm_flt_in6[1:0]	PWM6 复位/翻转事件	PWM6 CMPC 事件	PWM6 CMPD 事件	PWM6 CMPC 事件

hrpwm_flt_in7[1:0]	PWM7 复位/翻转事件	PWM7 CMPC 事件	PWM7 CMPD 事件	PWM7 CMPC 事件

HRPWM_FLTINR3.FLTxCNT 设置了故障 x 计数器阈值。一个故障仅当事件计数达到 HRPWM_FLTINR3.FLTxCNT 时才会认为有效。

HRPWM_FLTINR2.FLTxRSTM 设置了故障 x 计数器复位模式：

- 0：故障计数器在复位/翻转事件时硬件复位，如下表所示。
- 1：仅当上一计数周期内未发生任何故障事件时，故障计数器在复位/翻转事件时复位，如下图 17-64 所示。

故障计数器可通过软件操作 HRPWM_FLTINR2.FLTxCRES 在任意时刻进行复位。

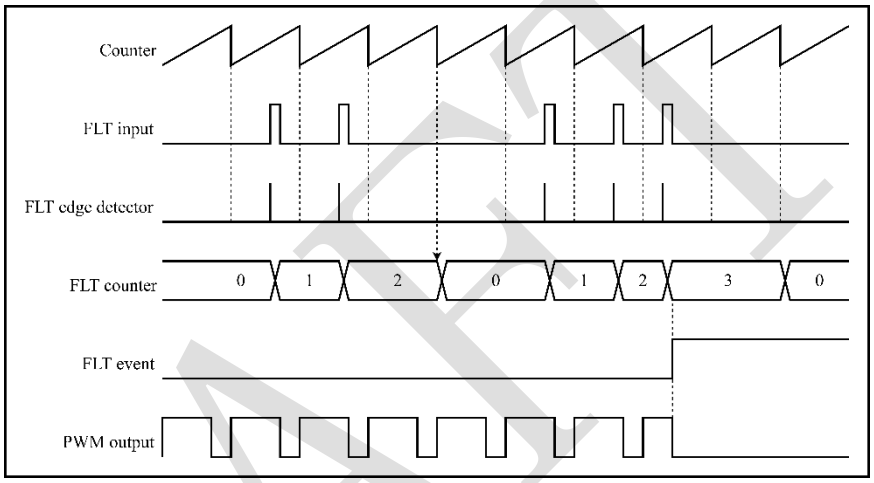


图 17-64　故障计数器累积模式（FLTxRSTM = 1，FLTxCNT [3:0] = 2）

一个给定的 FLTx 故障计数器只能被单一来源复位。表 17-23 描述了给定故障与哪个定时单元相关。这并不影响一个故障事件可以被多个定时器共用（例如，故障计数器使能的 FLT0 可以同时作用于定时器 0、定时器 1 和定时器 2）。

表 17-23　故障事件 0~7 计数复位来源

故障事件输入	故障计数复位来源
hrpwm_flt_in0[1:0]	定时器 0 复位/翻转事件
hrpwm_flt_in1[1:0]	定时器 1 复位/翻转事件
hrpwm_flt_in2[1:0]	定时器 2 复位/翻转事件
hrpwm_flt_in3[1:0]	定时器 3 复位/翻转事件
hrpwm_flt_in4[1:0]	定时器 4 复位/翻转事件
hrpwm_flt_in5[1:0]	定时器 5 复位/翻转事件
hrpwm_flt_in6[1:0]	定时器 6 复位/翻转事件
hrpwm_flt_in7[1:0]	定时器 7 复位/翻转事件

系统故障输入（hrpwm_sys_flt）

系统故障由 MCU 提供，对应于来自以下源的系统故障：

- 时钟安全系统
- SRAM 奇偶校验器

- Cortex™-M4-lockup 信号
- LVD 检测器
- FLASH ECC 双重错误检测

此输入会覆盖 FAULT 输入，并禁止所有 HRPWM_OUTxR.FAULTx = 01、10、11 的输出。

对于任一定时单元，都会使用 HRPWM_FLTxR.FLTxEN 使能 8 条故障通道，并且可同时选择 8 条通道（只要输出受故障机制保护，便会自动使能 sysfault）。这样便可实现：

- 一条故障通道同时禁止多个定时单元
- 多条故障通道进行或运算来禁止单个定时单元

对于每个定时器，故障期间的输出状态是通过 HRPWM_OUTxR. FAULTx 定义。

17.4.17 将 HRPWM 与其他定时器或 HRPWM 示例同步

HRPWM 可作为主单元（生成同步信号）或从单元（等待触发被同步）来同步多个 HRPWM 实例。此功能也可用于 HRPWM 与其他片外或片上定时器进行同步。同步电路在主定时器内进行控制。

同步输出

本节介绍了如何配置 HRPWM 才能同步外部资源并充当主定时器单元。

可选择四种事件作为要发送到同步输出的源。通过 HRPWM_MCR0.SYNCOUTSRC 进行配置，具体如下：

- 00：主定时器启动
  当 HRPWM_MCR1.MCEN 置 1、或者定时器在单次模式下达到周期值后重新启动时，会生成该事件。如果计数期间发生复位（HRPWM_MCR0.CONT 或 HRPWM_MCR0.RETRIG 置 1）时，也会生成该事件。
- 01：主定时器 CMPA 事件
- 10：定时器 0 启动
  当 HRPWM_MCR1.CEN0 置 1 时、或者计数器复位并重新开始计数时，会生成该事件。以下计数器复位事件不会传播到同步输出：连续模式下的计数器翻转、单次不可重触发模式下被丢弃的复位请求。仅当计数期间发生复位时（HRPWM_MCR0.CONT 或 HRPWM_MCR0.RETRIG 置 1），才会考虑复位。
- 11：定时器 0 CMPA 事件

HRPWM_MCR0.SYNCOUTEN、HRPWM_MCR0.SYNCOUTPOL 指定了同步事件的生成方式。

如果 HRPWM_MCR0.SYNCOUTEN = 1，则会在 HRPWM_SCOUT 输出引脚上生成同步脉冲。SYNCOUTPOL 位指定了同步信号的极性。如果 HRPWM_MCR0.SYNCOUTPOL = 0，HRPWM_SCOUT 引脚具有低空闲电平，并会发出长度为 16 个 FHRPWM 时钟周期的正脉冲进行同步。如果 HRPWM_MCR0.SYNCOUTPOL = 1，空闲电平为高电平，并会生成负脉冲。

注：同步脉冲后会有 16 个 FHRPWM 时钟周期的空闲电平，在此期间，会丢弃任何新的同步请求。因此，最大同步频率为 FHRPWM/32。

必须在配置 MCU 输出和计数器使能之前执行同步输出初始化步骤，具体执行步骤如下：

1. 配置 HRPWM_MCR0.SYNCOUTEN、HRPWM_MCR0.SYNCOUTPOL 和 HRPWM_MCR0.SYNCOUTSRC
2. 配置 HRPWM_SCOUT 引脚（参见 GPIO/Pinmux 部分）
3. 使能主定时器或定时器 0 计数器（MCEN 或 CENx 位置 1）

使能同步输入模式并同时启动计数器（使用 HRPWM_MCR0.SYNCSTRTM / HRPWM_PWMxCR0.SYNCSTRT）和同步输出模式（HRPWM_MCR0.SYNCOUTSRC = 00 或 10）后，仅当计数器将在运行时启动或复位时，才会生成输出脉冲。如果复位请求将计数器清零而不启动计数器，则不会影响同步输出。

同步输入

HRPWM 可通过外部源进行同步，具体通过对 HRPWM_MCR0.SYNCINEN、HRPWM_MCR0.SYNCINSRC 进行编程来实现。

如果 HRPWM_MCR0.SYNCINEN = 0，同步输入功能被禁止。HRPWM_MCR0.SYNCINSRC 指定同步输入的来源。如果 HRPWM_MCR0.SYNCINEN = 0，同步输入来源为片上通用定时器（TMR9_TRGO）。如果 HRPWM_MCR0.SYNCINEN = 1，同步输入来源为 HRPWM_SCIN 输入引脚上的正脉冲。

主定时器或定时单元（HRPWM_MCR1.MCEN 和/或 HRPWM_MCR1.CENx 置 1）使能后，不能更改此位。

HRPWM_SCIN 输入为上升沿有效。定时器行为通过 HRPWM_MCR0 和 HRPWM_PWMxCR0 寄存器中的以下位定义（有关详细信息，请参见表 17-20）：

- 同步启动：输入信号会启动定时器的计数器（SYNCSTRTM 和/或 SYNCSTRTx 位置 1）。CENx（MCEN）位必须置 1 才能使能定时器并使计数器准备号启动。在连续模式下，计数器在收到同步信号之前不会启动。
- 同步复位：输入信号会复位计数器（HRPWM_MCR0.SYNCRSTM 和/或 HRPWM_PWMxCR0.SYNCRST 置 1）。该事件会像其他复位事件一样使重复计数器递减。

仅当相关计数器使能后（HRPWM_MCR1.MCEN 或 HRPWM_MCR1.CENx 置 1），才会考虑同步事件。同步事件会触发 SYNCIN 中断。

注：如果当前计数器值大于有效周期值，同步启动事件会复位计数器。

同步事件的作用取决于定时器工作模式，参见表 17-24 中总结的内容。

表 17-24 同步事件的作用与定时器工作模式

PWM 工作模式	SYNCRSTx	SYNCSTRTx	同步复位或同步启动的行为
单次模式（不可再触发）	0	1	当计数器停止时启动事件在下列情况下有效: ● MCEN 或者 CENx 位置 1 ● 计数到达周期值 当计数器在周期值停止时发生启动事件，启动事件会复位计数器。复位请求会清除计数器但不会启动计数器（计数器只能够通过同步事件重新启动）。在计数期间任何复位事件都会被忽

			略（正如在通常的不可重复触发模式下）
	1	x	同步复位事件可以启动计时器计数。当计数器停止时，复位事件在下列情况下有效： ● MCEN 或者 CENx 位置 1 ● 计数到达周期值 当有多个复位请求（包括同步复位以及内部复位）时，只考虑第一个复位请求。
单次模式（可再触发）	0	1	仅在计数器没有启动或者到达周期值之后，计数器的同步启动才会生效。在计数器启动之后任何同步事件都不会生效。 当计数器在周期值停止时发生启动事件，启动事件会复位计数器。复位请求会清除计数器但不会启动计数器（计数器只能够通过同步事件重新启动）。在计数期间任何复位事件不会被忽略（正如在通常的可重复触发模式下）
	1	x	来自 HRPWM_SCIN 的复位请求有效，正如 HRPWM 任意一个从内部事件中产生的复位请求一样，这些复位请求可以启动或重新启动计时器计数。 当有多个复位请求时，只考虑第一个复位请求。
连续模式	0	1	计数器使能（MCEN 或者 CENx 置 1）之后会等待同步事件，当同步事件到来后计时器启动计数。计数器启动之后的同步事件都是无效的（计数器只能够被同步事件启动）。复位请求会清除计数器，但不会启动计数器。
	1	x	来自 HRPWM_SCIN 的复位请求有效，正如 HRPWM 任意一个从内部事件中产生的复位请求一样，这些复位请求可以启动或重新启动计时器计数。 当有多个复位请求时，只考虑第一个复位请求。

图 17-65 显示了单次模式下如何进行同步启动。

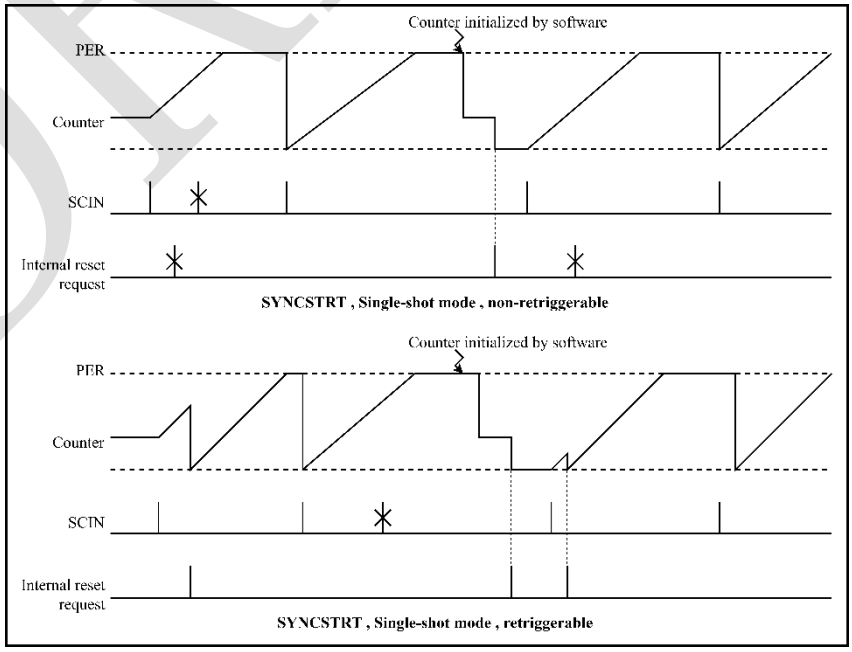


图 17-65　同步启动模式下计数器的行为

17.4.18 ADC/DAC 触发事件

ADC/DAC 转换器可由主定时器和 8 个定时单元触发。

HRPWM 总共支持 10 个 ADC 触发事件，可通过 10 个独立的触发事件同时启动各路 ADC 的常规转换序列和注入转换序列，或者启动 DAC 三角波生成和锯齿波生成。10 个触发事件对应在 ADC0R~ADC5R 及 ADC0ER~ADC3ER 寄存器中，最多可为每路触发输出合并 40 个事件（逻辑或运算），如下图所示。ADC 触发事件 0/2/4/6/8 和 1/3/5/7/9 将使用一组相同的源。

外部事件可用作触发信号，这类触发信号会在 HRPWM_EECRx 寄存器中定义的调节结束后立即获取，并且与 HRPWM_EEFxR1 和 HRPWM_EEFxR2 寄存器的配置无关。

可通过同时选择多个源的方式在单个开关周期内进行多次触发。典型案例是非重叠多相转换器，可通过单个 ADC 触发事件对所有相位进行连续采样。

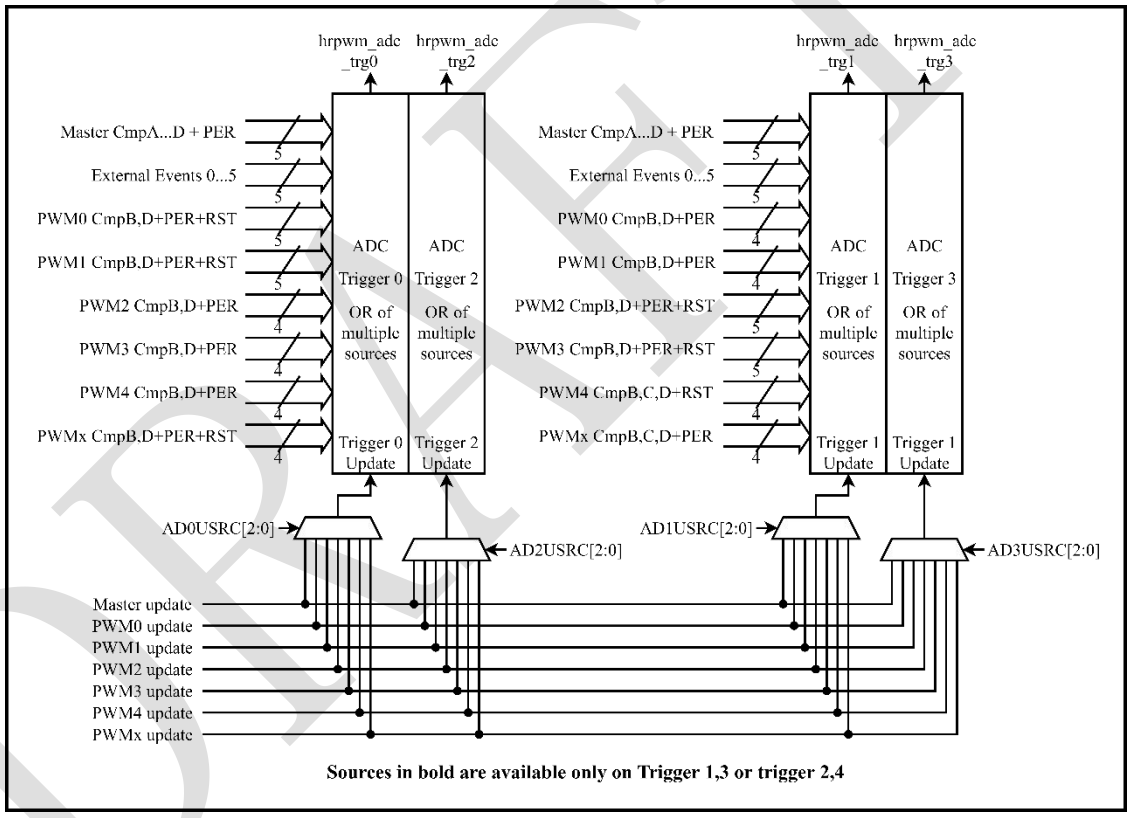


图 17-66　ADC 触发事件概览

ADC0R~ADC5R 及 ADC0ER~ADC5ER 寄存器支持预加载，并可与相关定时器同步更新。更新源是通过 HRPWM_CR0.USRCx 定义的。

举例来说，如果 ADC 触发 1 输出定时器 0 CMPB 事件（HRPWM_ADC0R = 0x00000400），HRPWM_ADC0R 通常将与定时器 0 同步更新（USRC0[2:0] = 001）。

如果源定时器中禁止预加载（HRPWM_MCR0.PREEN 或 HRPWM_PWMxCR0.PREEN 置 0），ADCxR 与 ADCxER 寄存器也不会预加载，写访问会立即更新触发源。

ADC 降采样

降采样单元可以降低 ADC 触发事件的频率，如下图 17-67 所示。

每个 ADC 触发事件的频率可以使用 HRPWM_ADPSRx.PSCx 分别进行调整。

在中心对齐模式下，ADC 触发事件频率还取决于各个源定时器中控制的 HRPWM_PWMxCR1.ADROM，如图 17-68 所示。HRPWM_PWMxCR1.ADROM 可用于任何可以触发 ADC 的事件，包括复位事件，翻转（周期）事件和比较事件：

- HRPWM_PWMxCR1.ADROM＝00：上/下计数阶段都会产生事件
- HRPWM_PWMxCR1.ADROM＝01：下计数阶段会产生事件
- HRPWM_PWMxCR1.ADROM＝10：上计数阶段会产生事件

ADC 降采样寄存器是可预加载寄存器，可以在不停止定时器工作的情况下即时更新。

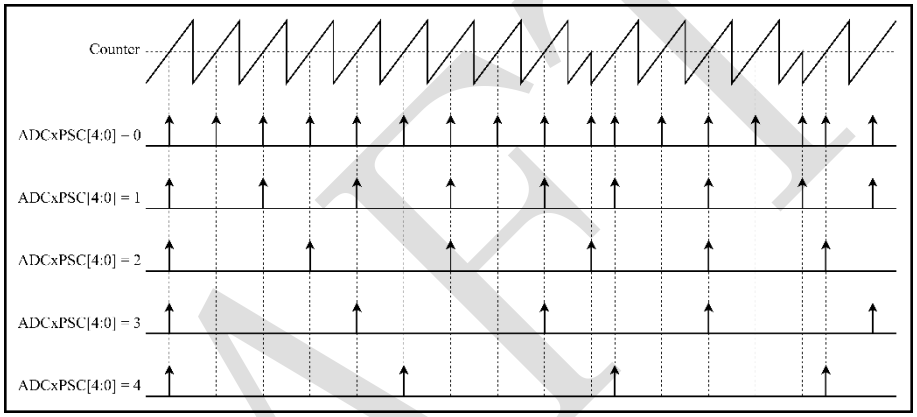


图 17-67　上计数模式下的 ADC 触发事件减采样

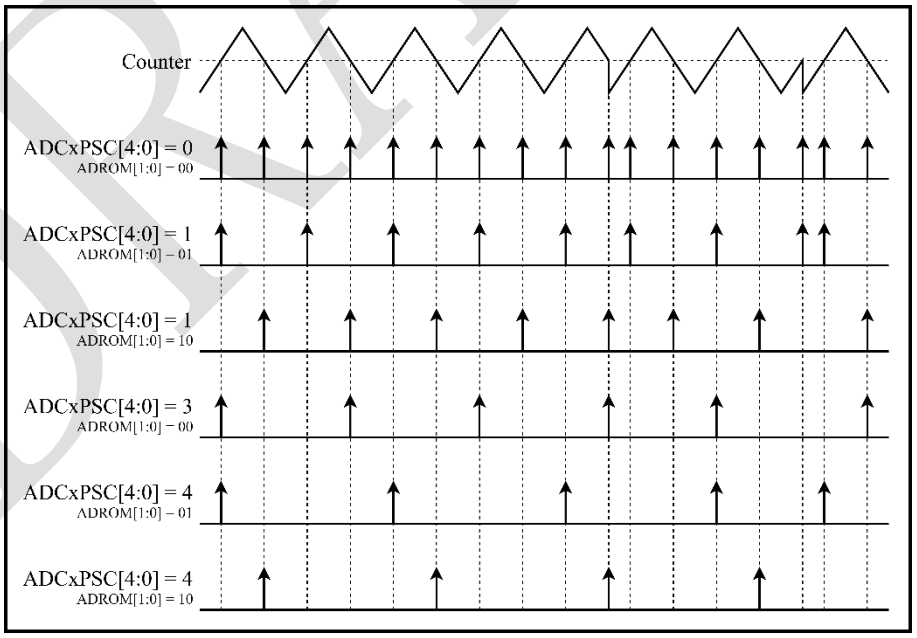


图 17-68　上/下计数模式下的 ADC 触发事件减采样

DAC 更新事件

HRPWM 支持 DAC 更新事件、使得 DAC 可以与定时单元同步更新。

主定时器和定时单元的更新事件可以在 hrpwm_dac_trg0~hrpwm_dac_trg2 中任意一路上生成 DAC 更新事件。

注：每个定时单元都有自己的 DAC 相关控制寄存器。

HRPWM_MCR0.DACSYNC 和 HRPWM_PWMxCR0.DACSYNC 的编程如下：

- 00：未生成更新事件
- 01：在 hrpwm_dac_trg0 上生成更新事件
- 10：在 hrpwm_dac_trg1 上生成更新事件
- 11：在 hrpwm_dac_trg2 上生成更新事件

更新事件会在 hrpwm_dac_trgx 上生成一个长度为 16 个 FHRPWM 时钟周期的输出脉冲。

如果多个定时单元中使能了 HRPWM_MCR0.DACSYNC 和 HRPWM_PWMxCR0.DACSYNC，hrpwm_dac_trgx 输出将由所有定时器更新事件的或运算结果决定。例如，如果定时器 0 和定时器 1 中 HRPWM_PWMxCR0.DACSYNC = 1，那么定时器 0 中的更新事件将与定时器 1 中的更新事件进行或运算，在相应的 hrpwm_dac_trg0 输出上生成 DAC 更新事件，如图 17-69 所示。

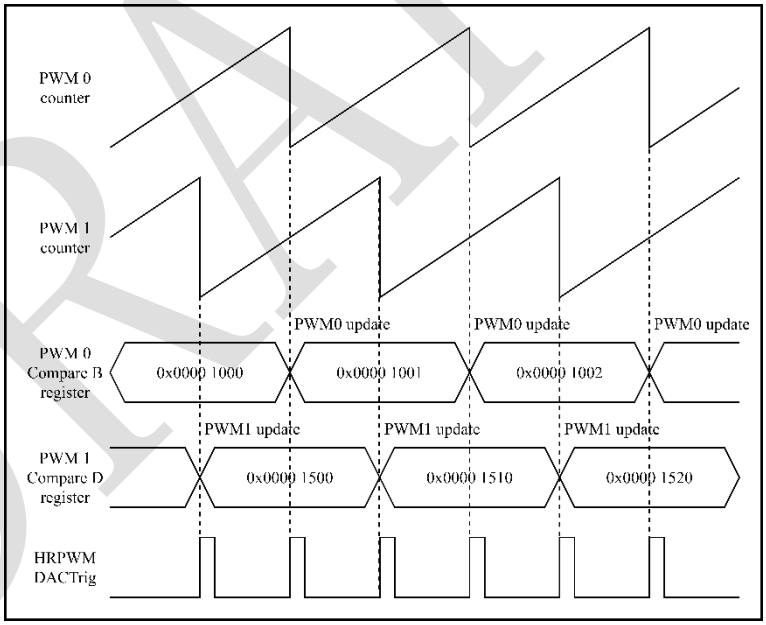


图 17-69 多个更新事件组合为 hrpwm_dac_trgx 输出

DAC 锯齿波触发事件

斜坡补偿技术以及迟滞控制可以使用 HRPWM 和 DAC 锯齿波发生器实现。基本原理是通过 DAC 产生一个递减的与 PWM 周期同步的锯齿波，或者一个与 PWM 信号同步的方波。

此模式通过 HRPWM_PWMxCR1.DCDE 使能。定时器开始工作后（HRPWM_MCR1.CENx 置 1），这一位不可以更改。

此模式使用了两个触发事件，如下图 17-70 所示：

- hrpwm_reset_dac_trg 产生 DAC 复位/更新事件
- hrpwm_step_dac_trg 产生 DAC 步进事件

HRPWM_PWMxCR1.DCDR 决定 hrpwm_reset_dac_trg 触发事件何时生成：

- HRPWM_PWMxCR1.DCDR = 0：触发事件在计数器复位或翻转事件处产生
- HRPWM_PWMxCR1.DCDR = 1：触发事件在输出 A 置位事件处产生

注：当 HRPWM_PWMxCR1.DCDE 复位时，DCDR 位无意义（DAC 锯齿波触发被禁止）

HRPWM_PWMxCR1.DCDS 决定 hrpwm_step_dac_trg 触发事件何时生成：

- HRPWM_PWMxCR1.DCDS = 0：CMPD 事件产生触发事件
- HRPWM_PWMxCR1.DCDS = 1：OUTA 复位事件产生触发事件

HRPWM_PWMxCR1.DCDR 和 HRPWM_PWMxCR1.DCDS 可以覆盖以下的应用场景：

- 边沿对齐的斜坡补偿（HRPWM_PWMxCR1.DCDR=HRPWM_PWMxCR1.DCDS = 0）：DAC 的锯齿波在 PWM 周期开始处启动，且在 PWM 周期内会产生多个触发事件
- 中心对齐的斜坡补偿（HRPWM_PWMxCR1.DCDR = 1，HRPWM_PWMxCR1.DCDS = 0）：DAC 的锯齿波在输出置位事件处启动，且在 PWM 周期内会产生多个触发事件
- 迟滞控制器：当输出状态更改时，每个周期必须更改 DAC 值两次。每个 PWM 周期产生 2 个触发事件。在边沿对齐模式（HRPWM_PWMxCR1.DCDR = 0，HRPWM_PWMxCR1.DCDS = 1）时，触发事件将在计数器复位或翻转时产生。在居中对齐模式下（HRPWM_PWMxCR1.DCDR = 1，HRPWM_PWMxCR1.DCDS = 1），触发事件将在输出置位时产生。

当 HRPWM_PWMxCR1.DCDE 被置位且 HRPWM_PWMxCR1.DCDS 被复位时，CMPD 会进入一种特定的工作模式。每次生成比较值匹配事件后，有效的比较值就会自动更新，这样触发事件可以周期性地重复产生，其中周期为 CMPD 值。

注：CMPD 值可以即时更改，新的值会在下一个比较值匹配事件处更新。

表 17-25 给出了一个例子，在一个 PWM 周期内生成 6 个触发事件。表格说明了有必要对除法结果四舍五入取上舍入值。

考虑一个计数器周期 PWMxPER =8192。将 8192 除以 6 得到 1365.33。

- 下舍入值：1365：会生成 7 个触发事件，第 6 个触发事件和第 7 个触发事件非常接近（分别对应计数值= 8190 和 8192）
- 上舍入值：1366：会生成 6 个触发事件，hrpwm_step_dac_trg 的第 6 个触发事件（对应计数值= 8192）被舍弃，因为计数器从 8192 到 0 发生翻转。

表 17-25 DAC 锯齿波触发事件示例

-	CMPD = 1365	reset_dac_trg	step_dac_trg	CMPD = 1366	reset_dac_trg	step_dac_trg
Counter value	1365	-	1	1366	-	1
	2730	-	2	2732	-	2
	4095	-	3	4098	-	3
	5460	-	4	5464	-	4
	6825	-	5	6830	-	5

	8190	-	6	8192	6	-
	8192	7	-	1366	-	1
	1365	-	1	2732	-	2
	...	-	-	...	-	-

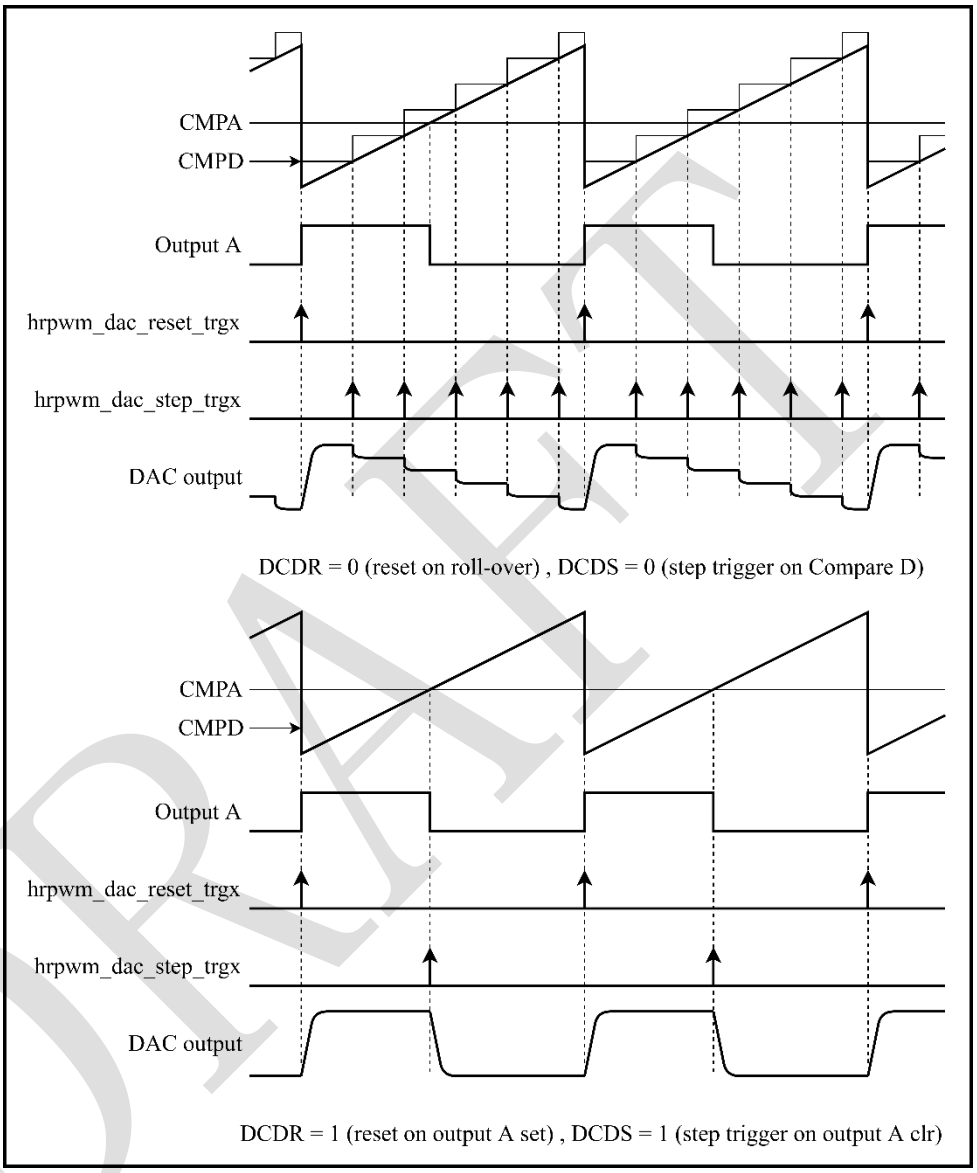


图 17-70　用于斜坡补偿的 DAC 锯齿波触发事件

图 17-71 显示了所有可用的 DAC 触发事件的一个综述。

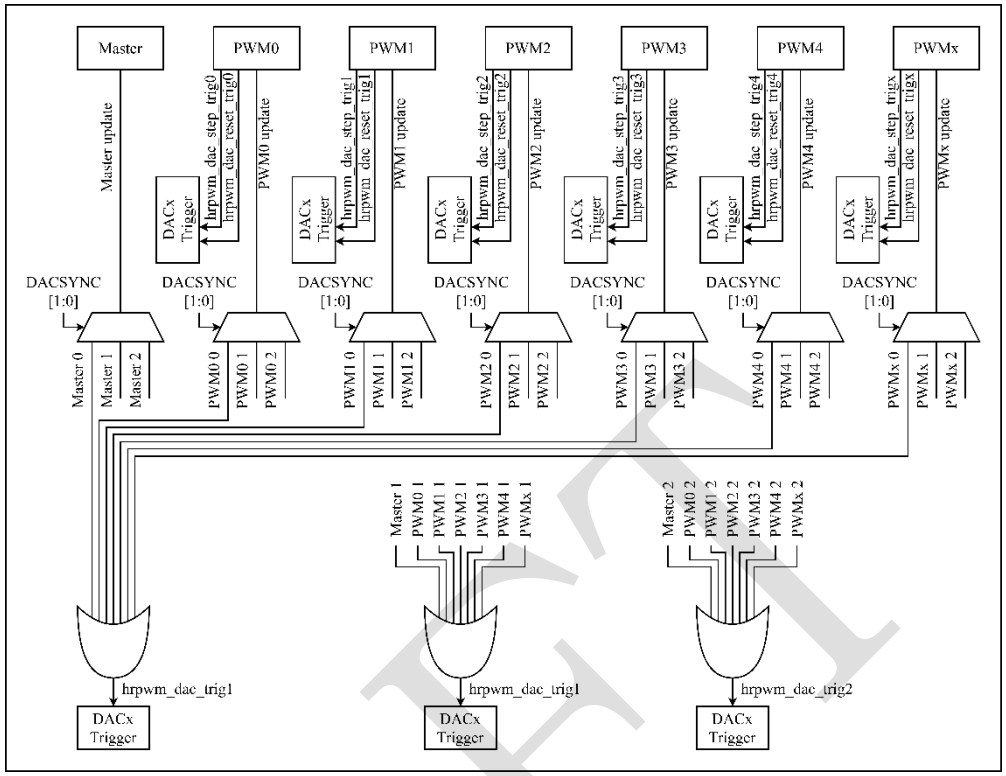


图 17-71　DAC 触发事件综述

17.4.19　HRPWM 中断

主定时器可生成 9 个中断：

- 接收同步事件
- 主定时器周期事件
- 主定时器重复事件
- 主定时器复位事件
- 主定时器寄存器更新
- 主定时器比较 A~比较 D 事件

每个定时单元可生成 15 个中断：

- 延迟保护触发事件
- 捕获 A 和捕获 B 事件
- 计数器周期（翻转）事件
- 计数器复位事件
- 输出 A 和输出 B 复位（从有效电平跳变为无效电平）
- 输出 A 和输出 B 置位（从无效电平跳变为有效电平）
- 定时单元寄存器更新
- 定时单元重复事件
- 定时单元比较 A~比较 D 事件

整个 HRPWM 生成 10 个故障中断：

- 突发模式周期事件
- 系统故障事件

- 故障 0~故障 7 事件（不考虑定时单元的因素）

中断请求会分到 10 个向量组中，具体如下：

- hrpwm_mst_int：主定时器中断
- hrpwm_slv0_int：定时器 0 中断
- hrpwm_slv1_int：定时器 1 中断
- hrpwm_slv2_int：定时器 2 中断
- hrpwm_slv3_int：定时器 3 中断
- hrpwm_slv4_int：定时器 4 中断
- hrpwm_slv5_int：定时器 5 中断
- hrpwm_slv6_int：定时器 6 中断
- hrpwm_slv7_int：定时器 7 中断
- hrpwm_flt_int：HRPWM 故障中断

表 17-26 总结了中断请求及其映射以及相关控制和状态位。

表 17-26 HRPWM 中断汇总

中断向量	HRPWM 事件	中断标志	使能控制位	标志清除位
hrpwm_mst_int	同步事件接收事件	SYNC	SYNCIE	SYNC
	主寄存器更新事件	MUPD	MUPDIE	MUPD
	主周期重复事件	MREP	MREPIE	MREP
	主复位事件	MRST	MRSTIE	MRST
	主周期事件	MPER	MPERIE	MPER
	主比较 A～D 事件	MCMPA	MCMPAIE	MCMPA
		MCMPB	MCMPBIE	MCMPB
		MCMPC	MCMPCIE	MCMPC
		MCMPD	MCMPDIE	MCMPD
hrpwm_slv0_int hrpwm_slv1_int hrpwm_slv2_int hrpwm_slv3_int hrpwm_slv4_int hrpwm_slv5_int hrpwm_slv6_int hrpwm_slv7_int	延迟保护触发事件	DLYPRT	DLYPRTDE	DLYPRT
	捕获 A 和捕获 B 事件	CAPA	CAPADE	CAPA
		CAPB	CAPBDE	CAPB
	周期重复事件	REP	REPIE	REP
	计数复位事件	RST	RSTIE	RST
	输出 A 和输出 B 复位事件（有效电平到无效电平）	CLRA	CLRAIE	CLRA
		CLRB	CLRBIE	CLRB
	输出 A 和输出 B 置位事件（无效电平到有效电平）	SETA	SETAIE	SETA
		SETB	SETBIE	SETB
	寄存器更新事件	UPD	UPDIE	UPD
	周期翻转事件	PER	PERIE	PER
	比较 A～D 事件	CMPA	CMPAIE	CMPA
		CMPB	CMPBIE	CMPB
		CMPC	CMPCIE	CMPC
		CMPD	CMPDIE	CMPD
hrpwm_flt_int	突发模式周期事件	BMPER	BMPERIE	BMPER
	系统故障事件	SYSFLT	SYSFLTIE	SYSFLT
	故障 0～7 事件	FLT0	FLT0IE	FLT0

		FLT1	FLT1IE	FLT1
		FLT2	FLT2IE	FLT2
		FLT3	FLT3IE	FLT3
		FLT4	FLT4IE	FLT4
		FLT5	FLT5IE	FLT5
		FLT6	FLT6IE	FLT6
		FLT7	FLT7IE	FLT7

17.4.20 HRPWM DMA

HRPWM 主定时器和定时单元的事件除了生成中断之外，也可以生成 DMA 请求，给到内部的突发 DMA 处理。每个定时器（主定时器、定时器 x）都有自己的 DMA 使能寄存器。

各个定时单元都会生成各自的 DMA 请求，DMA 请求在进行或运算后给到突发 DMA：

- 主定时器 1 个请求：hrpwm_mst_req
- 每个定时单元 1 个请求：hrpwm_slvx_req

表 17-27 总结了事件及其关联的 DMA 使能位。

表 17-27 HRPWM DMA 请求汇总

DMA 请求	HRPWM 事件	DMA 支持	使能控制位
hrpwm_mst_req	同步事件接收事件	支持	SYNCDE
	主寄存器更新事件	支持	MUPDDE
	主周期重复事件	支持	MREPDE
	主复位事件	支持	MRSTDE
	主周期事件	支持	MPERDE
	主比较 A～D 事件	支持	MCMPADE
		支持	MCMPBDE
		支持	MCMPCDE
		支持	MCMPDDE
hrpwm_slv0_req hrpwm_slv1_req hrpwm_slv2_req hrpwm_slv3_req hrpwm_slv4_req hrpwm_slv5_req hrpwm_slv6_req hrpwm_slv7_req	延迟保护触发事件	支持	DLYPRTDE
	捕获 A 和捕获 B 事件	支持	CAPADE
		支持	CAPBDE
	周期重复事件	支持	REPDE
	计数复位事件	支持	RSTDE
	输出 A 和输出 B 复位事件（有效电平到无效电平）	支持	CLRADE
		支持	CLRBDE
	输出 A 和输出 B 置位事件（无效电平到有效电平）	支持	SETADE
		支持	SETBDE
	寄存器更新事件	支持	UPDDE
	周期翻转事件	支持	PERDE
	比较 A～D 事件	支持	CMPADE
		支持	CMPBDE
		支持	CMPCDE

		支持	CMPDDE
-	突发模式周期事件	不支持	-
	系统故障事件	不支持	-
	故障 0 ~ 7 事件	不支持	-
		不支持	-
		不支持	-
		不支持	-
		不支持	-
		不支持	-
		不支持	-
		不支持	-

突发 DMA 传输

为了响应各个定时单元的 DMA 请求，HRPWM 支持突发 DMA 控制器，可通过单个 DMA 请求更新多个寄存器。具体包括：

- 通过一个 DMA 请求更新多个寄存器
- 动态对一个或多个定时单元进行重新配置，适用于使用多路定时器输出的转换器。

突发 DMA 功能可以选择 9 个 DMA 请求中的任意一个进行突发 DMA 传输。

突发 DMA 可以控制写入哪些寄存器，主定时器和定时器 0~7 包含突发 DMA 更新寄存器，各个寄存器都关联到选择位：HRPWM_BDMUPR、HRPWM_BDUPR0~HRPWM_BDUPR7（DMA 仅支持寄存器写访问）。突发 DMA 重定向机制允许自动将 DMA 写访问转发到相关的寄存器，如图 17-72 所示。

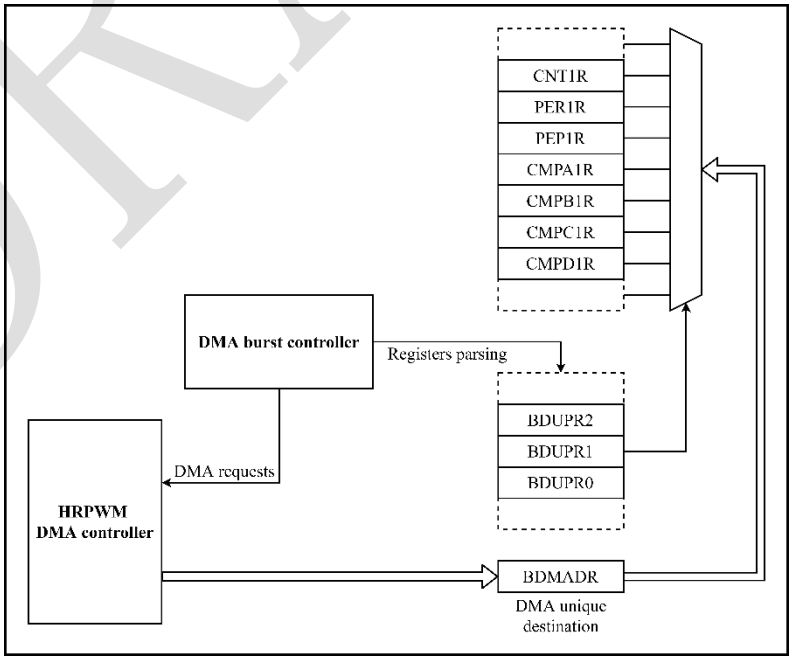


图 17-72 突发 DMA 综述

接收 DMA 请求时，HRPWM 会解析更新寄存器，如果控制位置 1，DMA 写访问会重定向到

相关的寄存器。如果控制位置 0，则会跳过寄存器更新并解析下一个控制位，判断下一个控制位是否置 1。HRPWM 完成 9 个更新寄存器（HRPWM_BDMUPR、HRPWM_BDUPR0~7）解析后，突发 DMA 传输结束，HRPWM 准备好处理下一个 DMA 请求，如图 17-73 所示。

任何在突发正在进行时发生的请求都会被丢弃，但最后一次数据传输期间发生的请求除外。

突发 DMA 模式永久使能（没有使能位）。突发 DMA 动作是通过第一次对 HRPWM_BDMADR 寄存器的写访问启动的。

使用突发 DMA 功能需要在 HRPWM_BDMADR 寄存器指向的地址存放更新数据，HRPWM 会根据更新寄存器的配置从 HRPWM_BDMADR 寄存器指向的地址中取出对应个数的数据。

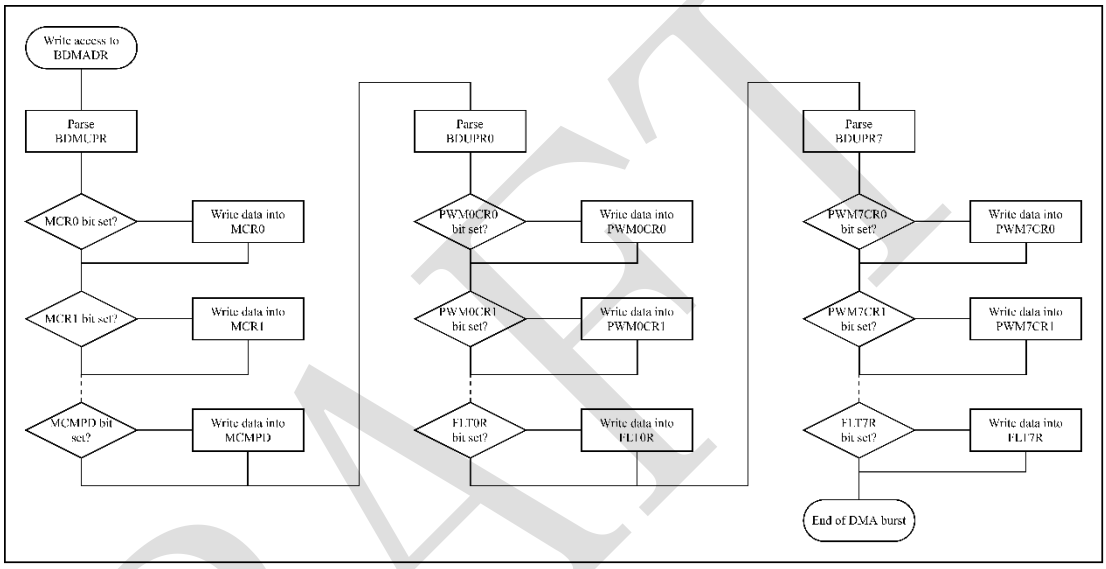


图 17-73　突发 DMA 运行流程图

突发 DMA 结束后，寄存器更新有多种选项可供使用，具体视更新策略而定。

如果 PREEN 位置 0（预加载禁止），通过突发 DMA 写入的值会立即传输到有效寄存器中，按照 DMA 写入的速度连续更新寄存器。

如果 PREEN 位置 1（预加载使能），总共有 3 个选项：

1. 更新独立于突发 DMA 传输进行（HRPWM_PWMxCR0.UPDGAT=0000，HRPWM_MCR0.BRSTDMA = 00）。在这种情况下，如果希望所有传输的数据同时更新，用户必须检查突发 DMA 传输是否在发生更新事件之前结束。相反，如果更新事件在 DMA 传输进行时发生，则仅会装载部分寄存器，完整的寄存器更新将需要 2 个连续的更新事件。

2. 更新在突发 DMA 传输完成后进行（HRPWM_PWMxCR0.UPDGAT = 1000，HRPWM_MCR0.BRSTDMA = 10）。这种模式可确保所有传输新寄存器值同时更新，更新操作独立于计数值进行，可与常规更新事件相结合（例如 HRPWM_PWMxCR0.UPDRST 置 1 的情况下，在发生计数器复位事件时进行更新）。

3. 更新在突发 DMA 传输完成后发生更新事件时进行（HRPWM_PWMxCR0.UPDGAT = 1100，HRPWM_MCR0.BRSTDMA = 11）。这种模式可确保对所有已传输数据的更新与常规更新事件、与定时单元计数器的更新同步。在这种情况下，如果正在进行传输时发生常规更新事件，该事件会被丢弃，并会在下一次发出更新事件时进行更新。

图 17-74 时序图显示了 3 种情况下的有效寄存器：

1. HRPWM_PWMxCR0.PREEN=0
2. HRPWM_PWMxCR0.PREEN = 1 以及 HRPWM_PWMxCR0.UPDGAT[3:0] = 1000
3. HRPWM_PWMxCR0.PREEN = 1 以及 HRPWM_PWMxCR0.UPDGAT[3:0] = 1100。

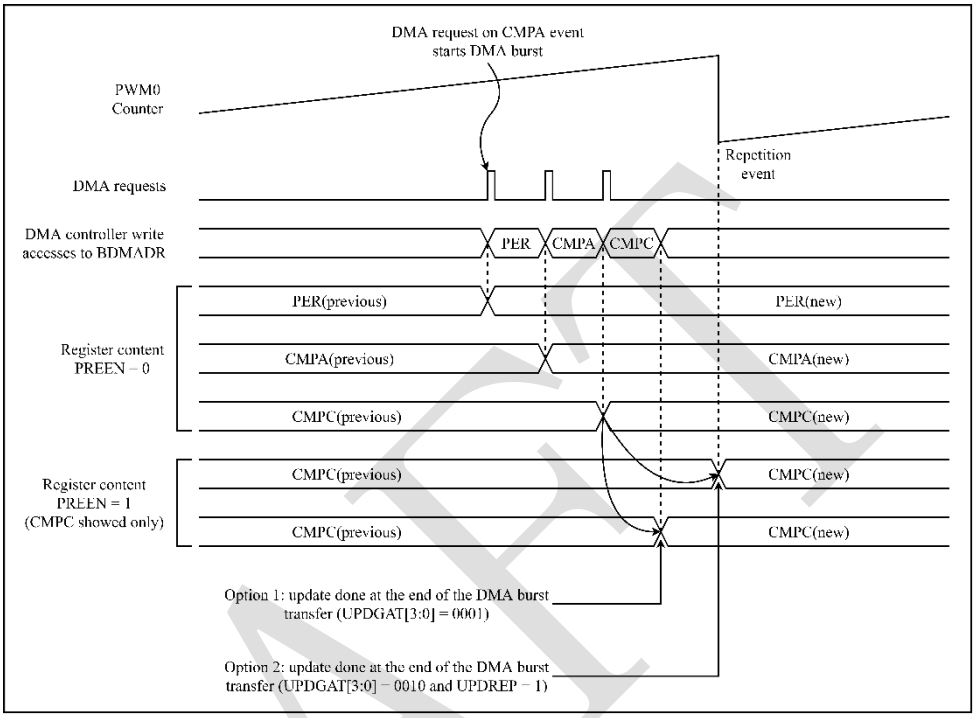


图 17-74　DMA 突发传输后的寄存器更新

17.5 寄存器定义

17.5.1 寄存器列表

HRPWM 基地址：
HRPWM0(0x4003B100) HRPWM1(0x4003B200) HRPWM2(0x4003B300) HRPWM3(0x4003B400)
HRPWM4(0x4003B500) HRPWM5(0x4003B600) HRPWM6(0x4003B700) HRPWM7(0x4003B800)

偏移	实例地址	名称	默认值	描述
0x00	HRPWM 基地址+0x00	HRPWM_MCR0	0x00000000	HRPWM 主定时器控制寄存器 0
0x04	HRPWM 基地址+0x04	HRPWM_MCR1	0x00000000	HRPWM 主定时器控制寄存器 1
0x08	HRPWM 基地址+0x08	HRPWM_MISR	0x00000000	HRPWM 主定时器中断状态寄存器
0x0C	HRPWM 基地址+0x0C	HRPWM_MDIER	0x00000000	HRPWM 主定时器中断使能寄存器
0x10	HRPWM 基地址+0x10	HRPWM_MCNTR	0x00000000	HRPWM 主定时器计数值寄存器
0x14	HRPWM 基地址+0x14	HRPWM_MPER	0x0000FFDF	HRPWM 主定时器周期值寄存器
0x18	HRPWM 基地址+0x18	HRPWM_MREP	0x00000000	HRPWM 主定时器重复寄存器
0x1C	HRPWM 基地址+0x1C	HRPWM_MCMPAR	0x00000000	HRPWM 主定时器比较值 A 寄存器
0x20	HRPWM 基地址+0x20	HRPWM_MCMPBR	0x00000000	HRPWM 主定时器比较值 B 寄存器
0x24	HRPWM 基地址+0x24	HRPWM_MCMPCR	0x00000000	HRPWM 主定时器比较值 C 寄存器
0x28	HRPWM 基地址+0x28	HRPWM_MCMPDR	0x00000000	HRPWM 主定时器比较值 D 寄存器
0x100+N*0x100 [N=0-7]	HRPWM 基地址+0x100+N*0x100[N=0-7]	HRPWM_PWMxCR0	0x00000000	HRPWM 定时器 x 控制寄存器 0
0x104+N*0x100 [N=0-7]	HRPWM 基地址+0x104+N*0x100[N=0-7]	HRPWM_PWMxCR1	0x00000000	HRPWM 定时器 x 控制寄存器 1
0x108+N*0x100 [N=0-7]	HRPWM 基地址+0x108+N*0x100[N=0-7]	HRPWM_PWMxISR	0x00000000	HRPWM 定时器 x 中断状态寄存器
0x10C+N*0x100 [N=0-7]	HRPWM 基地址+0x10C+N*0x100[N=0-7]	HRPWM_PWMxDIER	0x00000000	HRPWM 定时器 x 中断使能寄存器
0x110+N*0x100 [N=0-7]	HRPWM 基地址+0x110+N*0x100[N=0-7]	HRPWM_CNTxR	0x00000000	HRPWM 定时器 x 计数值寄存器
0x114+N*0x100 [N=0-7]	HRPWM 基地址+0x114+N*0x100[N=0-7]	HRPWM_PERxR	0x0000FFDF	HRPWM 定时器 x 周期值寄存器
0x118+N*0x100 [N=0-7]	HRPWM 基地址+0x118+N*0x100[N=0-7]	HRPWM_REPxR	0x00000000	HRPWM 定时器 x 重复寄存器
0x11C+N*0x100 [N=0-7]	HRPWM 基地址+0x11C+N*0x100[N=0-7]	HRPWM_CMPAxR	0x00000000	HRPWM 定时器 x 比较值 A 寄存器
0x120+N*0x100 [N=0-7]	HRPWM 基地址+0x120+N*0x100[N=0-7]	HRPWM_CMPBxR	0x00000000	HRPWM 定时器 x 比较值 B 寄存器
0x124+N*0x100 [N=0-7]	HRPWM 基地址+0x124+N*0x100[N=0-7]	HRPWM_CMPCxR	0x00000000	HRPWM 定时器 x 比较值 C 寄存器
0x128+N*0x100	HRPWM 基地址+0x128+N*0x100[N=0-7]	HRPWM_CMPDxR	0x00000000	HRPWM 定时器 x 比较值 D 寄存器

[N=0-7]				
0x12C+N*0x100 [N=0-7]	HRPWM 基地址+0x12C+N*0x100[N=0-7]	HRPWM_CAPAxR	0x00000000	HRPWM 定时器 x 捕获 A 寄存器
0x130+N*0x100 [N=0-7]	HRPWM 基地址+0x130+N*0x100[N=0-7]	HRPWM_CAPBxR	0x00000000	HRPWM 定时器 x 捕获 B 寄存器
0x134+N*0x100 [N=0-7]	HRPWM 基地址+0x134+N*0x100[N=0-7]	HRPWM_DTxR	0x00000000	HRPWM 定时器 x 死区时间寄存器
0x138+N*0x100 [N=0-7]	HRPWM 基地址+0x138+N*0x100[N=0-7]	HRPWM_SETxAR	0x00000000	HRPWM 定时器 x 输出 A 置位寄存器
0x13C+N*0x100 [N=0-7]	HRPWM 基地址+0x13C+N*0x100[N=0-7]	HRPWM_CLRxAR	0x00000000	HRPWM 定时器 x 输出 A 复位寄存器
0x140+N*0x100 [N=0-7]	HRPWM 基地址+0x140+N*0x100[N=0-7]	HRPWM_SETxBR	0x00000000	HRPWM 定时器 x 输出 B 置位寄存器
0x144+N*0x100 [N=0-7]	HRPWM 基地址+0x144+N*0x100[N=0-7]	HRPWM_CLRxBR	0x00000000	HRPWM 定时器 x 输出 B 复位寄存器
0x148+N*0x100 [N=0-7]	HRPWM 基地址+0x148+N*0x100[N=0-7]	HRPWM_EEFxR0	0x00000000	HRPWM 定时器 x 外部事件寄存器 0
0x14C+N*0x100 [N=0-7]	HRPWM 基地址+0x14C+N*0x100[N=0-7]	HRPWM_EEFxR1	0x00000000	HRPWM 定时器 x 外部事件寄存器 1
0x150+N*0x100 [N=0-7]	HRPWM 基地址+0x150+N*0x100[N=0-7]	HRPWM_EEFxR2	0x00000000	HRPWM 定时器 x 外部事件寄存器 2
0x154+N*0x100 [N=0-7]	HRPWM 基地址+0x154+N*0x100[N=0-7]	HRPWM_RST0R HRPWM_RST1R HRPWM_RST2R HRPWM_RST3R HRPWM_RST4R HRPWM_RST5R HRPWM_RST6R HRPWM_RST7R	0x00000000	HRPWM 定时器 x 复位寄存器
0x158+N*0x100 [N=0-7]	HRPWM 基地址+0x158+N*0x100[N=0-7]	HRPWM_RST0ER HRPWM_RST1ER HRPWM_RST2ER HRPWM_RST3ER HRPWM_RST4ER HRPWM_RST5ER HRPWM_RST6ER HRPWM_RST7ER	0x00000000	HRPWM 计数器复位寄存器
0x15C+N*0x100 [N=0-7]	HRPWM 基地址+0x15C+N*0x100[N=0-7]	HRPWM_CHPxR	0x00000000	HRPWM 定时器 x 斩波寄存器
0x160+N*0x100 [N=0-7]	HRPWM 基地址+0x160+N*0x100[N=0-7]	HRPWM_CAPA0CR HRPWM_CAPA1CR HRPWM_CAPA2CR HRPWM_CAPA3CR HRPWM_CAPA4CR	0x00000000	HRPWM 定时器 x 捕获 A 控制寄存器

		HRPWM_CAPA5CR HRPWM_CAPA6CR HRPWM_CAPA7CR		
0x164+N*0x100 [N=0-7]	HRPWM 基地址+0x164+N*0x100[N=0-7]	HRPWM_CAPA0CER HRPWM_CAPA1CER HRPWM_CAPA2CER HRPWM_CAPA3CER HRPWM_CAPA4CER HRPWM_CAPA5CER HRPWM_CAPA6CER HRPWM_CAPA7CER	0x00000000	HRPWM 定时器 x 捕获 A 控制扩展寄存器
0x168+N*0x100 [N=0-7]	HRPWM 基地址+0x168+N*0x100[N=0-7]	HRPWM_CAPB0CR HRPWM_CAPB1CR HRPWM_CAPB2CR HRPWM_CAPB3CR HRPWM_CAPB4CR HRPWM_CAPB5CR HRPWM_CAPB6CR HRPWM_CAPB7CR	0x00000000	HRPWM 定时器 x 捕获 B 控制寄存器
0x16C+N*0x100 [N=0-7]	HRPWM 基地址+0x16C+N*0x100[N=0-7]	HRPWM_CAPB0CER HRPWM_CAPB1CER HRPWM_CAPB2CER HRPWM_CAPB3CER HRPWM_CAPB4CER HRPWM_CAPB5CER HRPWM_CAPB6CER HRPWM_CAPB7CER	0x00000000	HRPWM 定时器 x 捕获 B 控制扩展寄存器
0x170+N*0x100 [N=0-7]	HRPWM 基地址+0x170+N*0x100[N=0-7]	HRPWM_OUTxR	0x00000000	HRPWM 定时器 x 输出控制寄存器
0x174+N*0x100 [N=0-7]	HRPWM 基地址+0x174+N*0x100[N=0-7]	HRPWM_FLTxR	0x00000000	HRPWM 定时器 x 故障使能寄存器
0xF00	HRPWM 基地址+0xF00	HRPWM_CR0	0x00000000	HRPWM 控制寄存器 0
0xF04	HRPWM 基地址+0xF04	HRPWM_CR1	0x00000000	HRPWM 控制寄存器 1
0xF08	HRPWM 基地址+0xF08	HRPWM_CR2	0x00000000	HRPWM 控制寄存器 2
0xF10	HRPWM 基地址+0xF10	HRPWM_ISR	0x00000000	HRPWM 中断状态寄存器
0xF14	HRPWM 基地址+0xF14	HRPWM_IER	0x00000000	HRPWM 中断使能寄存器
0xF18	HRPWM 基地址+0xF18	HRPWM_OENR	0x00000000	HRPWM 输出使能寄存器
0xF1C	HRPWM 基地址+0xF1C	HRPWM_ODISR	0x00000000	HRPWM 输出关闭寄存器
0xF20	HRPWM 基地址+0xF20	HRPWM_EECR0	0x00000000	HRPWM 外部事件寄存器 0
0xF24	HRPWM 基地址+0xF24	HRPWM_EECR1	0x00000000	HRPWM 外部事件寄存器 1
0xF28	HRPWM 基地址+0xF28	HRPWM_EECR2	0x00000000	HRPWM 外部事件寄存器 2
0xF2C	HRPWM 基地址+0xF2C	HRPWM_EECR3	0x00000000	HRPWM 外部事件寄存器 3
0xF30	HRPWM 基地址+0xF30	HRPWM_ADC0R	0x00000000	HRPWM ADC 触发事件 0 寄存器
0xF34	HRPWM 基地址+0xF34	HRPWM_ADC0ER	0x00000000	HRPWM ADC 触发事件 0 扩展寄存器

0xF38	HRPWM 基地址+0xF38	HRPWM_ADC1R	0x00000000	HRPWM ADC 触发事件 1 寄存器
0xF3C	HRPWM 基地址+0xF3C	HRPWM_ADC1ER	0x00000000	HRPWM ADC 触发事件 1 扩展寄存器
0xF40	HRPWM 基地址+0xF40	HRPWM_ADC2R	0x00000000	HRPWM ADC 触发事件 2 寄存器
0xF44	HRPWM 基地址+0xF44	HRPWM_ADC2ER	0x00000000	HRPWM ADC 触发事件 2 扩展寄存器
0xF48	HRPWM 基地址+0xF48	HRPWM_ADC3R	0x00000000	HRPWM ADC 触发事件 3 寄存器
0xF4C	HRPWM 基地址+0xF4C	HRPWM_ADC3ER	0x00000000	HRPWM ADC 触发事件 3 扩展寄存器
0xF50	HRPWM 基地址+0xF50	HRPWM_ADC4R	0x00000000	HRPWM ADC 触发事件 4 寄存器
0xF54	HRPWM 基地址+0xF54	HRPWM_ADC5R	0x00000000	HRPWM ADC 触发事件 5 寄存器
0xF58	HRPWM 基地址+0xF58	HRPWM_ADCUR	0x00000000	HRPWM ADC 更新寄存器
0xF5C	HRPWM 基地址+0xF5C	HRPWM_ADCLR	0x00000000	HRPWM ADC 长度寄存器
0xF60	HRPWM 基地址+0xF60	HRPWM_ADPSR0	0x00000000	HRPWM ADC 触发事件后分频寄存器 0
0xF64	HRPWM 基地址+0xF64	HRPWM_ADPSR1	0x00000000	HRPWM ADC 触发事件后分频寄存器 1
0xF68	HRPWM 基地址+0xF68	HRPWM_DLLCR	0x00000006	HRPWM DLL 控制寄存器
0xF70	HRPWM 基地址+0xF70	HRPWM_FLTINR0	0x00000000	HRPWM 故障输入寄存器 0
0xF78	HRPWM 基地址+0xF78	HRPWM_FLTINR1	0x00000000	HRPWM 故障输入寄存器 1
0xF7C	HRPWM 基地址+0xF7C	HRPWM_FLTINER	0x00000000	HRPWM 故障输入扩展寄存器
0xF80	HRPWM 基地址+0xF80	HRPWM_FLTINR2	0x00000000	HRPWM 故障输入寄存器 2
0xF88	HRPWM 基地址+0xF88	HRPWM_FLTINR3	0x00000000	HRPWM 故障输入寄存器 3
0xF90	HRPWM 基地址+0xF90	HRPWM_BMCR	0x00000000	HRPWM Burst Mode 控制寄存器
0xF94	HRPWM 基地址+0xF94	HRPWM_BMTRGR0	0x00000000	HRPWM Burst Mode 触发寄存器 0
0xF98	HRPWM 基地址+0xF98	HRPWM_BMTRGR1	0x00000000	HRPWM Burst Mode 触发寄存器 1
0xFA0	HRPWM 基地址+0xFA0	HRPWM_BMPER	0x00000000	HRPWM Burst Mode 周期寄存器
0xFA4	HRPWM 基地址+0xFA4	HRPWM_BMCMPR	0x00000000	HRPWM Burst Mode 比较寄存器
0xFB0	HRPWM 基地址+0xFB0	HRPWM_BDMUPR	0x00000000	HRPWM Burst DMA 主机更新寄存器
0xFB4	HRPWM 基地址+0xFB4	HRPWM_BDUPR0	0x00000000	HRPWM Burst DMA 更新寄存器 0
0xFB8	HRPWM 基地址+0xFB8	HRPWM_BDUPR1	0x00000000	HRPWM Burst DMA 更新寄存器 1
0xFBC	HRPWM 基地址+0xFBC	HRPWM_BDUPR2	0x00000000	HRPWM Burst DMA 更新寄存器 2
0xFC0	HRPWM 基地址+0xFC0	HRPWM_BDUPR3	0x00000000	HRPWM Burst DMA 更新寄存器 3
0xFC4	HRPWM 基地址+0xFC4	HRPWM_BDUPR4	0x00000000	HRPWM Burst DMA 更新寄存器 4
0xFC8	HRPWM 基地址+0xFC8	HRPWM_BDUPR5	0x00000000	HRPWM Burst DMA 更新寄存器 5
0xFCC	HRPWM 基地址+0xFCC	HRPWM_BDUPR6	0x00000000	HRPWM Burst DMA 更新寄存器 6
0xFD0	HRPWM 基地址+0xFD0	HRPWM_BDUPR7	0x00000000	HRPWM Burst DMA 更新寄存器 7
0xFF0	HRPWM 基地址+0xFF0	HRPWM_BDMADR	0x00000000	HRPWM Burst DMA 数据起始地址寄存器

17.5.2 寄存器描述

17.5.2.1 HRPWM 主定时器控制寄存器 0（HRPWM_MCR0）

- 名称：HRPWM Master PWM Control Register0
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRSTDMA				MREPU	MRSTU	PREEN	DACSYNC								
R/W				R/W	R/W	R/W	R/W								
0x0				0x0	0x0	0x0	0x0								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNCOUTSRC		SYNCOUTEN	SYNCOUTPOL	SYNCSTRTM	SYNCRSTM	SYNCINEN	SYNCINSRC	INTLVD		HALF	RETRIG	CONT	CKPSC		
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W		
0x0		0x0	0x0	0x0	0x0	0x0	0x0	0x0		0x0	0x0	0x0	0x0		

字段	说明
[31:30] BRSTDMA	**Burst DMA 更新** 此位定义了 Burst DMA 动作产生 Master PWM 更新事件的时间点 Others：Reserved 3：更新事件产生于 Burst DMA 动作完成后的下一个更新事件 2：更新事件产生于 Burst DMA 动作完成 0：更新事件独立于 Burst DMA 动作完成
[27] MREPU	**Master PWM Repetition 更新** 此位定义了 Master PWM 更新是否由 Master PWM Repetition 事件触发 1：Master PWM Repetition 事件触发 Master PWM 更新 0：Master PWM Repetition 事件不触发 Master PWM 更新
[26] MRSTU	**Master PWM Reset/RollOver 更新** 此位定义了 Master PWM 更新是否由 Master PWM Reset/RollOver 事件触发 1：Master PWM Reset/RollOver 事件触发 Master PWM 更新 0：Master PWM Reset/RollOver 事件不触发 Master PWM 更新
[25] PREEN	**Master PWM 预加载使能** 此位定义了是否开启预加载功能，定义了寄存器写访问直接作用于工作寄存器还是作用于影子寄存器 1：预加载开启，写访问作用于影子寄存器 0：预加载关闭，写访问直接作用于工作寄存器
[24:23] DACSYNC	**DAC Trigger 同步** DAC Trigger 仅在更新事件处产生，此位定义了 Master PWM 更新时是否产生 DAC Trigger 3：在 hrpwm_dac_trg2 上产生 DAC Trigger 2：在 hrpwm_dac_trg1 上产生 DAC Trigger 1：在 hrpwm_dac_trg0 上产生 DAC Trigger 0：不产生 DAC Trigger
[15:14] SYNCOUTSRC	**同步事件输出来源** 此位定义了同步事件输出的来源 11：PWM0 Compare A 10：PWM0 Start/Reset 01： Master PWM Compare A 00： Master PWM Start
[13] SYNCOUTEN	**同步事件输出使能** 此位定义了是否使能同步事件输出 1：同步事件输出使能 0：同步事件输出不使能

[12] SYNCOUTPOL	**同步事件输出极性** 此位定义了同步事件输出的极性 1：同步事件输出低有效（输出一个低电平脉冲，16 个 fHRPWM 时钟周期） 0：同步事件输出高有效（输出一个高电平脉冲，16 个 fHRPWM 时钟周期）
[11] SYNCSTRTM	**同步事件启动 Master PWM** 此位定义了同步事件后的 Master PWM 行为 1：同步事件启动 Master PWM 0：同步事件不启动 Master PWM
[10] SYNCRSTM	**同步事件复位 Master PWM** 此位定义了同步事件后的 Master PWM 行为 1：同步事件复位 Master PWM 0：同步事件不复位 Master PWM
[9] SYNCINEN	**同步事件输入使能** 此位定义了是否使能同步事件输入用以触发 HRPWM 1：同步事件输入使能，上升沿有效 0：同步事件输入不使能
[8] SYNCINSRC	**同步事件输入来源** 此位定义了同步事件输入的来源 1：HRPWM_SCIN 0：TIM9_TRGO
[7:6] INTLVD	**Master PWM Interleaved 模式** 此位配置 Master PWM Interleaved 模式，1/3 Interleaved 模式下 Compare A 数值自动被设为 Period 数值的 1/3，Compare B 数值自动被设为 Period 数值的 2/3，1/4 Interleaved 模式下 Compare A 数值自动被设为 Period 数值的 1/4，Compare B 数值自动被设为 Period 数值的 2/4，Compare C 数值自动被设为 Period 数值的 3/4 3：Reserved 2：1/4 Interleaved 模式开启 1：1/3 Interleaved 模式开启 0：Interleaved 模式关闭
[5] HALF	**Master PWM Half 模式** 此位使能 Master PWM Half 模式，Half 模式下 Compare A 数值自动被设为 Period 数值的 1/2 1：Half 模式开启 0：Half 模式关闭
[4] RETRIG	**Master PWM 可重触发模式** 此位定义了 PWMx 单次模式下的特性 1：Master PWM 可重触发，无论计数状态如何都可以进行计数复位 0：Master PWM 不可重触发，在计数停止后才可以进行计数复位
[3] CONT	**Master PWM 连续模式** 此位定义了 Master PWM 工作模式 1：Master PWM 工作在连续模式，计数到达 Period 值后翻转到 0 0：Master PWM 工作在单次模式，计数到达 Period 值后停止
[2:0] CKPSC	**Master PWM 时钟预分频比** 此位定义了 Master PWM 高分辨率时钟预分频比

17.5.2.2 HRPWM 主定时器控制寄存器 1（HRPWM_MCR1）

- 名称：HRPWM Master PWM Control Register1
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							CEN7	CEN6	CEN5	CEN4	CEN3	CEN2	CEN1	CEN0	MCEN
							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字段	说明
[24] CEN7	**PWM7 使能** 此位定义是否使能 PWM7 1：PWM7 使能 0：PWM7 不使能
[23] CEN6	**PWM6 使能** 此位定义是否使能 PWM6 1：PWM6 使能 0：PWM6 不使能
[22] CEN5	**PWM5 使能** 此位定义是否使能 PWM5 1：PWM5 使能 0：PWM5 不使能
[21] CEN4	**PWM4 使能** 此位定义是否使能 PWM4 1：PWM4 使能 0：PWM4 不使能
[20] CEN3	**PWM3 使能** 此位定义是否使能 PWM3 1：PWM3 使能 0：PWM3 不使能
[19] CEN2	**PWM2 使能** 此位定义是否使能 PWM2 1：PWM2 使能 0：PWM2 不使能
[18] CEN1	**PWM1 使能** 此位定义是否使能 PWM1 1：PWM1 使能 0：PWM1 不使能
[17] CEN0	**PWM0 使能** 此位定义是否使能 PWM0 1：PWM0 使能 0：PWM0 不使能
[16] MCEN	**Master PWM 使能** 此位定义是否使能 Master PWM 1：Master PWM 使能 0：Master PWM 不使能

17.5.2.3 HRPWM 主定时器中断状态寄存器（HRPWM_MISR）

- 名称：HRPWM Master PWM Interrupt Status Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							MREP	MRST	MUPD	SYNC	MPER	MCMPD	MCMPC	MCMPB	MCMPA
							R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[8] MREP	**Master Repetition 中断标志** 1：Repetition 中断标志产生 0：Repetition 中断标志未产生 Note：写 1 清除中断标志
[7] MRST	**Master Reset 中断标志** 1：Reset 中断标志产生 0：Reset 中断标志未产生 Note：写 1 清除中断标志
[6] MUPD	**Master Update 中断标志** 1：Update 中断标志产生 0：Update 中断标志未产生 Note：写 1 清除中断标志
[5] SYNC	**Sync Input 中断标志** 1：Sync Input 中断标志产生 0：Sync Input 中断标志未产生 Note：写 1 清除中断标志
[4] MPER	**Master Period 中断标志** 1：Period 中断标志产生 0：Period 中断标志未产生 Note：写 1 清除中断标志
[3] MCMPD	**Master Compare D 中断标志** 1：Compare D 中断标志产生 0：Compare D 中断标志未产生 Note：写 1 清除中断标志
[2] MCMPC	**Master Compare C 中断标志** 1：Compare C 中断标志产生 0：Compare C 中断标志未产生 Note：写 1 清除中断标志
[1] MCMPB	**Master Compare B 中断标志** 1：Compare B 中断标志产生 0：Compare B 中断标志未产生 Note：写 1 清除中断标志
[0] MCMPA	**Master Compare A 中断标志** 1：Compare A 中断标志产生 0：Compare A 中断标志未产生 Note：写 1 清除中断标志

17.5.2.4 HRPWM 主定时器中断使能寄存器（HRPWM_MDIER）

- 名称：HRPWM Master PWM DMA Interrupt Enable Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							MREPDE	MRSTDE	MUPDDE	SYNCDE	MPERDE	MCMPDDE	MCMPCDE	MCMPBDE	MCMPADE
							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							MREPIE	MRSTIE	MUPDIE	SYNCIE	MPERIE	MCMPDIE	MCMPCIE	MCMPBIE	MCMPAIE
							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[24] MREPDE	**Master Repetition DMA 使能** 1：Repetition DMA 使能 0：Repetition DMA 不使能
[23] MRSTDE	**Master Reset DMA 使能** 1：Reset DMA 使能 0：Reset DMA 不使能
[22] MUPDDE	**Master Update DMA 使能** 1：Update DMA 使能 0：Update DMA 不使能
[21] SYNCDE	**Sync Input DMA 使能** 1：Sync Input DMA 使能 0：Sync Input DMA 不使能
[20] MPERDE	**Master Period DMA 使能** 1：Period DMA 使能 0：Period DMA 不使能
[19] MCMPDDE	**Master Compare D DMA 使能** 1：Compare D DMA 使能 0：Compare D DMA 不使能
[18] MCMPCDE	**Master Compare C DMA 使能** 1：Compare C DMA 使能 0：Compare C DMA 不使能
[17] MCMPBDE	**Master Compare B DMA 使能** 1：Compare B DMA 使能 0：Compare B 中断不使能
[16] MCMPADE	**Master Compare A DMA 使能** 1：Compare A DMA 使能 0：Compare A DMA 不使能
[8] MREPIE	**Master Repetition 中断使能** 1：Repetition 中断使能 0：Repetition 中断不使能
[7] MRSTIE	**Master Reset 中断使能** 1：Reset 中断使能 0：Reset 中断不使能
[6] MUPDIE	**Master Update 中断使能** 1：Update 中断使能 0：Update 中断不使能
[5] SYNCIE	**Sync Input 中断使能** 1：Sync Input 中断使能 0：Sync Input 中断不使能
[4]	**Master Period 中断使能**

MPERIE	1：Period 中断使能 0：Period 中断不使能
[3] MCMPDIE	**Master Compare D 中断使能** 1：Compare D 中断使能 0：Compare D 中断不使能
[2] MCMPCIE	**Master Compare C 中断使能** 1：Compare C 中断使能 0：Compare C 中断不使能
[1] MCMPBIE	**Master Compare B 中断使能** 1：Compare B 中断使能 0：Compare B 中断不使能
[0] MCMPAIE	**Master Compare A 中断使能** 1：Compare A 中断使能 0：Compare A 中断不使能

17.5.2.5 HRPWM 主定时器计数值寄存器（HRPWM_MCNTR）

- **名称：HRPWM Master PWM Counter Register**
- **偏移地址：0x10**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												CNTWR	CNTRD		
												R/WAC	R/WAC		
												0x0	0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCNT															
R/W															
0x0															

字段	说明
[19] CNTWR	**Master PWM Counter 写入** 对此位写 1 同时写 MCNT，可以将 MCNT 写入到 Master PWM Counter 写 1 等待此位自动清零、此时 MCNT 已写入到 Master PWM Counter
[18] CNTRD	**Master PWM Counter 读取** 对此位写 1，可以将 Master PWM Counter 当前值读取到 MCNT 写 1 等待此位自动清零、此时 Master PWM Counter 已读取到 MCNT
[15:0] MCNT	**Master PWM Counter 数值** 此位保持 Master PWM Counter 数值，当高精度模式（CKPSC＜5）开启时，此位的低有效位无意义，这些位无法被写入，读取时返回为 0

17.5.2.6 HRPWM 主定时器周期值寄存器（HRPWM_MPER）

- 名称：HRPWM Master PWM Period Register
- 偏移地址：0x14
- 默认值：0x0000FFDF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPER															
R/W															
0xffdf															

字段	说明
[15:0] MPER	**Master PWM Period 数值** 此位保持 Master PWM Period 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Period 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Period 最大值为 0xFFDF

17.5.2.7 HRPWM 主定时器重复寄存器（HRPWM_MREP）

- 名称：HRPWM Master PWM Repetition Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MREP							
								R/W							
								0x0							

字段	说明
[7:0] MREP	**Master PWM Repetition Period 数值** 此位保持 Master PWM Repetition Period 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值

17.5.2.8 HRPWM 主定时器比较值 A 寄存器（HRPWM_MCMPAR）

- 名称：HRPWM Master PWM Compare A Register
- 偏移地址：0x1C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMPA															
R/W															
0x0															

字段	说明
[15:0] MCMPA	**Master PWM Compare A 数值** 此位保持 Master PWM Compare A 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare A 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare A 数值必须小于或等于 MPER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.9 HRPWM 主定时器比较值 B 寄存器（HRPWM_MCMPBR）

- 名称：HRPWM Master PWM Compare B Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMPB															
R/W															
0x0															

字段	说明
[15:0] MCMPB	**Master PWM Compare B 数值** 此位保持 Master PWM Compare B 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare B 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare B 数值必须小于或等于 MPER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.10 HRPWM 主定时器比较值 C 寄存器（HRPWM_MCMPCR）

- 名称：HRPWM Master PWM Compare C Register
- 偏移地址：0x24
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMPC															
R/W															
0x0															

字段	说明
[15:0] MCMPC	**Master PWM Compare C 数值** 此位保持 Master PWM Compare C 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare C 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare C 数值必须小于或等于 MPER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.11 HRPWM 主定时器比较值 D 寄存器（HRPWM_MCMPDR）

- 名称：HRPWM Master PWM Compare D Register
- 偏移地址：0x28
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMPD															
R/W															
0x0															

字段	说明
[15:0] MCMPD	**Master PWM Compare D 数值** 此位保持 Master PWM Compare D 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare D 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare D 数值必须小于或等于 MPER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.12 HRPWM 定时器 x 控制寄存器 0（HRPWM_PWMxCR0）

- 名称：HRPWM PWMx Control Register0
- 偏移地址：0x100+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDGAT				UPDREP	UPDRST	PREEN	DACSYNC		GTCMPC	GTCMPA	TRGHLF	DELCMPD		DELCMPB	
R/W				R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W		R/W	
0x0				0x0	0x0	0x0	0x0		0x0	0x0	0x0	0x0		0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				SYNCSTRT	SYNCRST	RSYNCU	PSHPLL	INTLVD		HALF	RETRIG	CONT	CKPSC		
				R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W		
				0x0	0x0	0x0	0x0	0x0		0x0	0x0	0x0	0x0		

字段	说明
[31:28] UPDGAT	**更新门控** 此位定义了 hrpwm_upd_in 到来之后产生 PWMx 更新事件的时间点 Others：Reserved 12：更新事件产生于 Burst DMA 动作完成后的下一个更新事件 8：更新事件产生于 Burst DMA 动作完成 7：更新事件产生于 hrpwm_upd_in[2] 的上升沿后的下一个更新事件 6：更新事件产生于 hrpwm_upd_in[1] 的上升沿后的下一个更新事件 5：更新事件产生于 hrpwm_upd_in[0] 的上升沿后的下一个更新事件 4：更新事件产生于 hrpwm_upd_in[2] 的上升沿 3：更新事件产生于 hrpwm_upd_in[1] 的上升沿 2：更新事件产生于 hrpwm_upd_in[0] 的上升沿 0：更新事件独立于 Burst DMA 动作完成
[27] UPDREP	**PWMx Repetition 更新** 此位定义了 PWMx 更新是否由 PWMx Repetition 事件触发 1：PWMx Repetition 事件触发 PWMx 更新 0：PWMx Repetition 事件不触发 PWMx 更新
[26] UPDRST	**PWMx Reset 更新** 此位定义了 PWMx 更新是否由 PWMx Reset / Roll-Over 事件触发 1：PWMx Reset / Roll-Over 事件触发 PWMx 更新 0：PWMx Reset / Roll-Over 事件不触发 PWMx 更新
[25] PREEN	**PWMx 预加载使能** 此位定义了是否开启预加载功能，定义了寄存器写访问直接作用于工作寄存器还是作用于影子寄存器 1：预加载开启，写访问作用于影子寄存器 0：预加载关闭，写访问直接作用于工作寄存器
[24:23] DACSYNC	**DAC Trigger 同步** DAC Trigger 仅在更新事件处产生，此位定义了 PWMx 更新时是否产生 DAC Trigger 3：在 hrpwm_dac_trg2 上产生 DAC Trigger 2：在 hrpwm_dac_trg1 上产生 DAC Trigger 1：在 hrpwm_dac_trg0 上产生 DAC Trigger 0：不产生 DAC Trigger
[22] GTCMPC	**PWMx CMPC Greater Than 模式** 此位开启 CMPC Greater Than 模式 1：Greater Than 模式开启 0：Greater Than 模式关闭
[21] GTCMPA	**PWMx CMPA Greater Than 模式** 此位开启 CMPA Greater Than 模式 1：Greater Than 模式开启 0：Greater Than 模式关闭
[20] TRGHLF	**PWMx Triggered-half 模式** 此位开启 Triggered-half 模式

	1：Triggered-half 模式开启 0：Triggered-half 模式关闭
[19:18] DELCMPD	**PWMx CMPD Auto-Delayed 模式** 此位配置 PWMx CMPD Auto-Delayed 模式 3：CMPD 在 CAPB 事件后或者在 CMPC 事件后有效，如果没有 CAPB 事件则会超时 2：CMPD 在 CAPB 事件后或者在 CMPA 事件后有效，如果没有 CAPB 事件则会超时 1：CMPD 在 CAPB 事件后有效 0：CMPD 一直有效
[17:16] DELCMPB	**PWMx CMPB Auto-Delayed 模式** 此位配置 PWMx CMPB Auto-Delayed 模式 3：CMPB 在 CAPA 事件后或者在 CMPC 事件后有效，如果没有 CAPA 事件则会超时 2：CMPB 在 CAPA 事件后或者在 CMPA 事件后有效，如果没有 CAPA 事件则会超时 1：CMPB 在 CAPA 事件后有效 0：CMPB 一直有效
[11] SYNCSTRT	**同步事件启动 PWMx** 此位定义了同步事件后的 PWMx 行为 1：同步事件启动 PWMx 0：同步事件不启动 PWMx
[10] SYNCRST	**同步事件复位 PWMx** 此位定义了同步事件后的 PWMx 行为 1：同步事件复位 PWMx 0：同步事件不复位 PWMx
[9] RSYNCU	**PWMx 重同步更新** 此位定义了来自 PWMx 外部的 Update 事件是否需要同步 1：来自 PWMx 外部的 Update 事件立即生效 0：来自 PWMx 外部的 Update 事件等待下个 Reset / Roll-Over 事件后生效
[8] PSHPLL	**PWMx Push-pull 模式** 此位开启 Push-pull 模式 1：Push-pull 模式开启 0：Push-pull 模式关闭
[7:6] INTLVD	**PWMx Interleaved 模式** 此位配置 PWMx Interleaved 模式，1/3 Interleaved 模式下 Compare A 数值自动被设为 Period 数值的 1/3，Compare B 数值自动被设为 Period 数值的 2/3，1/4 Interleaved 模式下 Compare A 数值自动被设为 Period 数值的 1/4，Compare B 数值自动被设为 Period 数值的 2/4，Compare C 数值自动被设为 Period 数值的 3/4 3：Reserved 2：1/4 Interleaved 模式开启 1：1/3 Interleaved 模式开启 0：Interleaved 模式关闭
[5] HALF	**PWMx Half 模式** 此位使能 PWMx Half 模式，Half 模式下 Compare A 数值自动被设为 Period 数值的 1/2 1：Half 模式开启 0：Half 模式关闭
[4] RETRIG	**PWMx 可重触发模式** 此位定义了 PWMx 单次模式下的特性 1：PWMx 可重触发，无论计数状态如何都可以进行计数复位 0：PWMx 不可重触发，在计数停止后才可以进行计数复位
[3] CONT	**PWMx 连续模式** 此位定义了 PWMx 工作模式 1：PWMx 工作在连续模式，计数到达 Period 值后翻转到 0 0：PWMx 工作在单次模式，计数到达 Period 值后停止
[2:0] CKPSC	**PWMx 时钟预分频比** 此位定义了 PWMx 高分辨率时钟预分频比

17.5.2.13 HRPWM 定时器 x 控制寄存器 1（HRPWM_PWMxCR1）

- 名称：HRPWM PWMx Control Register1
- 偏移地址：0x104+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							UPD7	UPD6	UPD5	UPD4	UPD3	UPD2	UPD1	UPD0	MUPD
							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLTROM		EEVROM		ADROM		OUTROM		ROM			UDM		DCDR	DCDS	DCDE
R/W		R/W		R/W		R/W		R/W			R/W		R/W	R/W	R/W
0x0		0x0		0x0		0x0		0x0			0x0		0x0	0x0	0x0

字段	说明
[24] UPD7	**PWM7 更新** 此位定义了 PWMx 更新是否由 PWM7 更新事件触发 1：PWM7 更新事件触发 PWMx 更新 0：PWM7 更新事件不触发 PWMx 更新
[23] UPD6	**PWM6 更新** 此位定义了 PWMx 更新是否由 PWM6 更新事件触发 1：PWM6 更新事件触发 PWMx 更新 0：PWM6 更新事件不触发 PWMx 更新
[22] UPD5	**PWM5 更新** 此位定义了 PWMx 更新是否由 PWM5 更新事件触发 1：PWM5 更新事件触发 PWMx 更新 0：PWM5 更新事件不触发 PWMx 更新
[21] UPD4	**PWM4 更新** 此位定义了 PWMx 更新是否由 PWM4 更新事件触发 1：PWM4 更新事件触发 PWMx 更新 0：PWM4 更新事件不触发 PWMx 更新
[20] UPD3	**PWM3 更新** 此位定义了 PWMx 更新是否由 PWM3 更新事件触发 1：PWM3 更新事件触发 PWMx 更新 0：PWM3 更新事件不触发 PWMx 更新
[19] UPD2	**PWM2 更新** 此位定义了 PWMx 更新是否由 PWM2 更新事件触发 1：PWM2 更新事件触发 PWMx 更新 0：PWM2 更新事件不触发 PWMx 更新
[18] UPD1	**PWM1 更新** 此位定义了 PWMx 更新是否由 PWM1 更新事件触发 1：PWM1 更新事件触发 PWMx 更新 0：PWM1 更新事件不触发 PWMx 更新
[17] UPD0	**PWM0 更新** 此位定义了 PWMx 更新是否由 PWM0 更新事件触发 1：PWM0 更新事件触发 PWMx 更新 0：PWM0 更新事件不触发 PWMx 更新
[16] MUPD	**Master PWM 更新** 此位定义了 PWMx 更新是否由 Master PWM 更新事件触发 1：Master PWM 更新事件触发 PWMx 更新 0：Master PWM 更新事件不触发 PWMx 更新
[15:14] FLTROM	**故障 Roll-Over 模式选择** 此位定义了 Up-Down 模式下何时生成 Roll-Over 事件，这里 Roll-Over 事件对应于 FLTINR2，用于产生 Fault 计数的复位信号 3：Reserved 2：Roll-Over 事件在计数为 Period 时产生

	1：Roll-Over 事件在计数为 0 时产生 0：Roll-Over 事件在计数为 Period / 0 时产生
[13:12] EEVROM	**事件 Roll-Over 模式选择** 此位定义了 Up-Down 模式下何时生成 Roll-Over 事件，这里 Roll-Over 事件对应于 EEFxR1，用于产生 Event A 计数的复位信号 3：Reserved 2：Roll-Over 事件在计数为 Period 时产生 1：Roll-Over 事件在计数为 0 时产生 0：Roll-Over 事件在计数为 Period / 0 时产生
[11:10] ADROM	**ADC Roll-Over 模式选择** 此位定义了 Up-Down 模式下何时生成 Roll-Over 事件，这里 Roll-Over 事件对应于 ADC0R-ADC7R，用于产生 Adc Trigger 输出 3：Reserved 2：Roll-Over 事件在计数为 Period 时产生 1：Roll-Over 事件在计数为 0 时产生 0：Roll-Over 事件在计数为 Period / 0 时产生
[9:8] OUTROM	**输出 Roll-Over 模式选择** 此位定义了 Up-Down 模式下何时生成 Roll-Over 事件，这里 Roll-Over 事件对应于 SETxyR、CLRxyR，用于产生 PWM 输出 3：Reserved 2：Roll-Over 事件在计数为 Period 时产生 1：Roll-Over 事件在计数为 0 时产生 0：Roll-Over 事件在计数为 Period / 0 时产生
[7:6] ROM	**Roll-Over 模式选择** 此位定义了 Up-Down 模式下何时生成 Roll-Over 事件，这里 Roll-Over 事件用于更新影子寄存器，产生中断标志位，减少重复计数值以及进行外部事件滤波 3：Reserved 2：Roll-Over 事件在计数为 Period 时产生 1：Roll-Over 事件在计数为 0 时产生 0：Roll-Over 事件在计数为 Period / 0 时产生
[4] UDM	**Up-Down 模式选择** 此位定义了 PWM 在何种模式下工作 1：PWM 在 Up-Down 模式下工作 0：PWM 在 Up 模式下工作
[2] DCDR	**DAC Reset Trigger 来源** 此位定义了何时生成 DAC Reset Trigger 事件 1：DAC Reset Trigger 产生于 SETA 事件 0：DAC Reset Trigger 产生于 RST 事件
[1] DCDS	**DAC Step Trigger 来源** 此位定义了何时生成 DAC Step Trigger 事件 1：DAC Step Trigger 产生于 CLRA 事件 0：DAC Step Trigger 产生于 CMPD 事件
[0] DCDE	**DAC Reset / Step Trigger 使能** 此位定义了是否使能 DAC Reset / Step Trigger 事件 1：DAC Reset / Step Trigger 使能 0：DAC Reset / Step Trigger 不使能

17.5.2.14 HRPWM 定时器 x 中断状态寄存器（HRPWM_PWMxISR）

- 名称：HRPWM PWMx Interrupt Status Register
- 偏移地址：0x108+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										OUTB	OUTA	OUTB STA	OUTA STA	IPPST A	CPPST A
										R	R	R	R	R	R
										0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DLYPR T	CAPB	CAPA	REP	RST	CLRB	SETB	CLRA	SETA	UPD	PER	CMPD	CMPC	CMPB	CMPA
	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[21] OUTB	**OUTB 当前输出（输出级之前）** 1：输出有效 0：输出无效
[20] OUTA	**OUTA 当前输出（输出级之前）** 1：输出有效 0：输出无效
[19] OUTBSTA	**OUTB 输出状态（延迟空闲保护发生时）** 1：输出有效 0：输出无效
[18] OUTASTA	**OUTA 输出状态（延迟空闲保护发生时）** 1：输出有效 0：输出无效
[17] IPPSTA	**PushPull 输出状态（延迟空闲/均衡空闲保护发生时）** 1：OUTB 推挽输出、OUTA 强制无效 0：OUTA 推挽输出、OUTB 强制无效
[16] CPPSTA	**PushPull 当前状态** 1：OUTB 推挽输出、OUTA 强制无效 0：OUTA 推挽输出、OUTB 强制无效
[14] DLYPRT	**Delayed Protection 中断标志** 1：Delayed Protection 中断标志产生 0：Delayed Protection 中断标志未产生 Note：写 1 清除中断标志
[13] CAPB	**Capture B 中断标志** 1：Capture B 中断标志产生 0：Capture B 中断标志未产生 Note：写 1 清除中断标志
[12] CAPA	**Capture A 中断标志** 1：Capture A 中断标志产生 0：Capture A 中断标志未产生 Note：写 1 清除中断标志
[11] REP	**Repetition 中断标志** 1：Repetition 中断标志产生 0：Repetition 中断标志未产生 Note：写 1 清除中断标志
[10] RST	**Reset 中断标志** 1：Reset 中断标志产生 0：Reset 中断标志未产生 Note：写 1 清除中断标志
[9] CLRB	**Out B Clear 中断标志** 1：Out B Clear 中断标志产生

	0：Out B Clear 中断标志未产生 Note：写 1 清除中断标志
[8] SETB	**Out B Set 中断标志** 1：Out B Set 中断标志产生 0：Out B Set 中断标志未产生 Note：写 1 清除中断标志
[7] CLRA	**Out A Clear 中断标志** 1：Out A Clear 中断标志产生 0：Out A Clear 中断标志未产生 Note：写 1 清除中断标志
[6] SETA	**Out A Set 中断标志** 1：Out A Set 中断标志产生 0：Out A Set 中断标志未产生 Note：写 1 清除中断标志
[5] UPD	**Update 中断标志** 1：Update 中断标志产生 0：Update 中断标志未产生 Note：写 1 清除中断标志
[4] PER	**Roll-Over 中断标志** 1：Roll-Over 中断标志产生 0：Roll-Over 中断标志未产生 Note：写 1 清除中断标志
[3] CMPD	**Compare D 中断标志** 1：Compare D 中断标志产生 0：Compare D 中断标志未产生 Note：写 1 清除中断标志
[2] CMPC	**Compare C 中断标志** 1：Compare C 中断标志产生 0：Compare C 中断标志未产生 Note：写 1 清除中断标志
[1] CMPB	**Compare B 中断标志** 1：Compare B 中断标志产生 0：Compare B 中断标志未产生 Note：写 1 清除中断标志
[0] CMPA	**Compare A 中断标志** 1：Compare A 中断标志产生 0：Compare A 中断标志未产生 Note：写 1 清除中断标志

17.5.2.15 HRPWM 定时器 x 中断使能寄存器（HRPWM_PWMxDIER）

- 名称：HRPWM PWMx DMA Interrupt Enable Register
- 偏移地址：0x10C+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DLYPRTDE	CAPBDE	CAPADE	REPDE	RSTDE	CLRBDE	SETBDE	CLRADE	SETADE	UPDDE	PERDE	CMPDDE	CMPCDE	CMPBDE	CMPADE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DLYPRTIE	CAPBIE	CAPAIE	REPIE	RSTIE	CLRBIE	SETBIE	CLRAIE	SETAIE	UPDIE	PERIE	CMPDIE	CMPCIE	CMPBIE	CMPAIE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[30] DLYPRTDE	**Delayed Protection DMA 使能** 1：Delayed Protection DMA 使能 0：Delayed Protection DMA 不使能
[29] CAPBDE	**Capture B DMA 使能** 1：Capture B DMA 使能 0：Capture B DMA 不使能
[28] CAPADE	**Capture A DMA 使能** 1：Capture A DMA 使能 0：Capture A DMA 不使能
[27] REPDE	**Repetition DMA 使能** 1：Repetition DMA 使能 0：Repetition DMA 不使能
[26] RSTDE	**Reset DMA 使能** 1：Reset DMA 使能 0：Reset DMA 不使能
[25] CLRBDE	**Out B Clear DMA 使能** 1：Out B Clear DMA 使能 0：Out B Clear DMA 不使能
[24] SETBDE	**Out B Set DMA 使能** 1：Out B Set DMA 使能 0：Out B Set DMA 不使能
[23] CLRADE	**Out A Clear DMA 使能** 1：Out A Clear DMA 使能 0：Out A Clear DMA 不使能
[22] SETADE	**Out A Set DMA 使能** 1：Out A Set DMA 使能 0：Out A Set DMA 不使能
[21] UPDDE	**Update DMA 使能** 1：Update DMA 使能 0：Update DMA 不使能
[20] PERDE	**Roll-Over DMA 使能** 1：Roll-Over DMA 使能 0：Roll-Over DMA 不使能
[19] CMPDDE	**Compare D DMA 使能** 1：Compare D DMA 使能 0：Compare D DMA 不使能
[18] CMPCDE	**Compare C DMA 使能** 1：Compare C DMA 使能 0：Compare C DMA 不使能
[17]	**Compare B DMA 使能**

CMPBDE	1：Compare B DMA 使能 0：Compare B DMA 不使能
[16] CMPADE	**Compare A DMA 使能** 1：Compare A DMA 使能 0：Compare A DMA 不使能
[14] DLYPRTIE	**Delayed Protection 中断使能** 1：Delayed Protection 中断使能 0：Delayed Protection 中断不使能
[13] CAPBIE	**Capture B 中断使能** 1：Capture B 中断使能 0：Capture B 中断不使能
[12] CAPAIE	**Capture A 中断使能** 1：Capture A 中断使能 0：Capture A 中断不使能
[11] REPIE	**Repetition 中断使能** 1：Repetition 中断使能 0：Repetition 中断不使能
[10] RSTIE	**Reset 中断使能** 1：Reset 中断使能 0：Reset 中断不使能
[9] CLRBIE	**Out B Clear 中断使能** 1：Out B Clear 中断使能 0：Out B Clear 中断不使能
[8] SETBIE	**Out B Set 中断使能** 1：Out B Set 中断使能 0：Out B Set 中断不使能
[7] CLRAIE	**Out A Clear 中断使能** 1：Out A Clear 中断使能 0：Out A Clear 中断不使能
[6] SETAIE	**Out A Set 中断使能** 1：Out A Set 中断使能 0：Out A Set 中断不使能
[5] UPDIE	**Update 中断使能** 1：Update 中断使能 0：Update 中断不使能
[4] PERIE	**Roll-Over 中断使能** 1：Roll-Over 中断使能 0：Roll-Over 中断不使能
[3] CMPDIE	**Compare D 中断使能** 1：Compare D 中断使能 0：Compare D 中断不使能
[2] CMPCIE	**Compare C 中断使能** 1：Compare C 中断使能 0：Compare C 中断不使能
[1] CMPBIE	**Compare B 中断使能** 1：Compare B 中断使能 0：Compare B 中断不使能
[0] CMPAIE	**Compare A 中断使能** 1：Compare A 中断使能 0：Compare A 中断不使能

17.5.2.16 HRPWM 定时器 x 计数值寄存器（HRPWM_CNTxR）

- 名称：HRPWM PWMx Counter Register
- 偏移地址：0x110+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												CNTWR	CNTRD		
												R/WAC	R/WAC		
												0x0	0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															
0x0															

字段	说明
[19] CNTWR	**PWMx Counter 写入** 对此位写 1 同时写 CNT，可以将 CNT 写入到 PWMx Counter 写 1 等待此位自动清零、此时 CNT 已写入到 PWMx Counter
[18] CNTRD	**PWMx Counter 读取** 对此位写 1，可以将 PWMx Counter 当前值读取到 CNT 写 1 等待此位自动清零、此时 PWMx Counter 已读取到 CNT
[15:0] CNT	**PWMx Counter 数值** 此位保持 PWMx Counter 数值，当高精度模式（CKPSC＜5）开启时，此位的低有效位无意义，这些位无法被写入，读取时返回为 0

17.5.2.17 HRPWM 定时器 x 周期值寄存器（HRPWM_PERxR）

- 名称：HRPWM PWMx Period Register
- 偏移地址：0x114+N*0x100[N=0-7]
- 默认值：0x0000FFDF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PER															
R/W															
0xffdf															

字段	说明
[15:0] PER	**PWMx Period 数值** 此位保持 PWMx Period 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Period 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Period 最大值为 0xFFDF，在 Auto-Delayed 模式下最大值为 0xFFBF

17.5.2.18 HRPWM 定时器 x 重复寄存器（HRPWM_REPxR）

- 名称：HRPWM PWMx Repetition Register
- 偏移地址：0x118+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								REP							
								R/W							
								0x0							

字段	说明
[7:0] REP	**PWMx Repetition Period 数值** 此位保持 PWMx Repetition Period 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值

17.5.2.19 HRPWM 定时器 x 比较值 A 寄存器（HRPWM_CMPAxR）

- 名称：HRPWM PWMx Compare A Register
- 偏移地址：0x11C+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA															
R/W															
0x0															

字段	说明
[15:0] CMPA	**PWMx Compare A 数值** 此位保持 PWMx Compare A 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare A 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare A 数值必须小于或等于 PER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.20 HRPWM 定时器 x 比较值 B 寄存器（HRPWM_CMPBxR）

- 名称：HRPWM PWMx Compare B Register
- 偏移地址：0x120+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB															
R/W															
0x0															

字段	说明
[15:0] CMPB	**PWMx Compare B 数值** 此位保持 PWMx Compare B 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare B 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare B 数值必须小于或等于 PER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.21 HRPWM 定时器 x 比较值 C 寄存器（HRPWM_CMPCxR）

- 名称：HRPWM PWMx Compare C Register
- 偏移地址：0x124+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPC															
R/W															
0x0															

字段	说明
[15:0] CMPC	**PWMx Compare C 数值** 此位保持 PWMx Compare C 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare C 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare C 数值必须小于或等于 PER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.22 HRPWM 定时器 x 比较值 D 寄存器（HRPWM_CMPDxR）

- 名称：**HRPWM PWMx Compare D Register**
- 偏移地址：**0x128+N*0x100[N=0-7]**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPD															
R/W															
0x0															

字段	说明
[15:0] CMPD	**PWMx Compare D 数值** 此位保持 PWMx Compare D 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值 Compare D 数值必须大于或等于 3 个 fHRPWM 周期，若 CKPSC＝0 为 0x60，若 CKPSC＝1 为 0x30，若 CKPSC＝2 为 0x18 Compare D 数值必须小于或等于 PER 减去 2 个 fHRPWM 周期，若 CKPSC＝0 为 0x40，若 CKPSC＝1 为 0x20，若 CKPSC＝2 为 0x10

17.5.2.23 HRPWM 定时器 x 捕获 A 寄存器（HRPWM_CAPAxR）

- 名称：**HRPWM PWMx Capture A Register**
- 偏移地址：**0x12C+N*0x100[N=0-7]**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															DIR
															R
															0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPA															
R															
0x0															

字段	说明
[16] DIR	**捕获 A 方向** 1：向下计数 0：向上计数 该位指示 CAPA 事件发生时的计数方向 在向上计数模式下（UDM=0）此位一直为 0
[15:0] CAPA	**捕获 A 数值** 此位指示 CAPA 事件发生时的计数值

17.5.2.24 HRPWM 定时器 x 捕获 B 寄存器（HRPWM_CAPBxR）

- 名称：HRPWM PWMx Capture B Register
- 偏移地址：0x130+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															DIR
															R
															0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPB															
R															
0x0															

字段	说明
[16] DIR	**捕获 B 方向** 1：向下计数 0：向上计数 该位指示 CAPB 事件发生时的计数方向 在向上计数模式下（UDM=0）此位一直为 0
[15:0] CAPB	**捕获 B 数值** 此位指示 CAPB 事件发生时的计数值

17.5.2.25 HRPWM 定时器 x 死区时间寄存器（HRPWM_DTxR）

- 名称：HRPWM PWMx DeadTime Register
- 偏移地址：0x134+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SDTF	DTF											
			R/W	R/W											
			0x0	0x0											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SDTR	DTR											
			R/W	R/W											
			0x0	0x0											

字段	说明
[28] SDTF	**下降沿死区符号** 1：下降沿死区为负向（信号重叠） 0：下降沿死区为正向（信号无重叠） 该位确定死区时间是正向还是负向
[27:16] DTF	**下降沿死区时间** 此位定义了在 PWM 参考信号下降沿之后的死区时间 tDTR ＝ DTR＊tDTG
[12] SDTR	**上升沿死区符号** 1：上升沿死区为负向（信号重叠） 0：上升沿死区为正向（信号无重叠） 该位确定死区时间是正向还是负向
[11:0] DTR	**上升沿死区时间** 此位定义了在 PWM 参考信号上升沿之后的死区时间 tDTR ＝ DTR＊tDTG

17.5.2.26 HRPWM 定时器 x 输出 A 置位寄存器（HRPWM_SETxAR）

- 名称：HRPWM PWMx Output A Set Register
- 偏移地址：0x138+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									UPD	RESYNC	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5
									R/W	R/W	R/W	R/W	R/W	R/W	R/W
									0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	EXTEVNT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	PER	CMPD	CMPC	CMPB	CMPA	SST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[22] UPD	**PWMx Update 事件使 PWMx Out A 进入有效状态**
[21] RESYNC	**Sync Input Event 事件使 PWMx Out A 进入有效状态**
[20] EXTEVNT9	**PWMx Event 9 事件使 PWMx Out A 进入有效状态**
[19] EXTEVNT8	**PWMx Event 8 事件使 PWMx Out A 进入有效状态**
[18] EXTEVNT7	**PWMx Event 7 事件使 PWMx Out A 进入有效状态**
[17] EXTEVNT6	**PWMx Event 6 事件使 PWMx Out A 进入有效状态**
[16] EXTEVNT5	**PWMx Event 5 事件使 PWMx Out A 进入有效状态**
[15] EXTEVNT4	**PWMx Event 4 事件使 PWMx Out A 进入有效状态**
[14] EXTEVNT3	**PWMx Event 3 事件使 PWMx Out A 进入有效状态**
[13] EXTEVNT2	**PWMx Event 2 事件使 PWMx Out A 进入有效状态**
[12] EXTEVNT1	**PWMx Event 1 事件使 PWMx Out A 进入有效状态**
[11] EXTEVNT0	**PWMx Event 0 事件使 PWMx Out A 进入有效状态**
[10] MSTPER	**Master PWM Period 事件使 PWMx Out A 进入有效状态**
[9] MSTCMPD	**Master PWM Compare D 事件使 PWMx Out A 进入有效状态**
[8] MSTCMPC	**Master PWM Compare C 事件使 PWMx Out A 进入有效状态**
[7] MSTCMPB	**Master PWM Compare B 事件使 PWMx Out A 进入有效状态**
[6] MSTCMPA	**Master PWM Compare A 事件使 PWMx Out A 进入有效状态**
[5] PER	**PWMx Roll-Over 事件使 PWMx Out A 进入有效状态**
[4] CMPD	**PWMx Compare D 事件使 PWMx Out A 进入有效状态**
[3] CMPC	**PWMx Compare C 事件使 PWMx Out A 进入有效状态**
[2] CMPB	**PWMx Compare B 事件使 PWMx Out A 进入有效状态**
[1] CMPA	**PWMx Compare A 事件使 PWMx Out A 进入有效状态**

[0] SST	Software Set Trigger 事件使 PWMx Out A 进入有效状态

17.5.2.27 HRPWM 定时器 x 输出 A 复位寄存器（HRPWM_CLRxAR）

- 名称：HRPWM PWMx Output A Clear Register
- 偏移地址：0x13C+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									UPD	RESYNC	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5
									R/W	R/W	R/W	R/W	R/W	R/W	R/W
									0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	EXTEVNT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	PER	CMPD	CMPC	CMPB	CMPA	SST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[22] UPD	PWMx Update 事件使 PWMx Out A 进入无效状态
[21] RESYNC	Sync Input Event 事件使 PWMx Out A 进入无效状态
[20] EXTEVNT9	PWMx Event 9 事件使 PWMx Out A 进入无效状态
[19] EXTEVNT8	PWMx Event 8 事件使 PWMx Out A 进入无效状态
[18] EXTEVNT7	PWMx Event 7 事件使 PWMx Out A 进入无效状态
[17] EXTEVNT6	PWMx Event 6 事件使 PWMx Out A 进入无效状态
[16] EXTEVNT5	PWMx Event 5 事件使 PWMx Out A 进入无效状态
[15] EXTEVNT4	PWMx Event 4 事件使 PWMx Out A 进入无效状态
[14] EXTEVNT3	PWMx Event 3 事件使 PWMx Out A 进入无效状态
[13] EXTEVNT2	PWMx Event 2 事件使 PWMx Out A 进入无效状态
[12] EXTEVNT1	PWMx Event 1 事件使 PWMx Out A 进入无效状态
[11] EXTEVNT0	PWMx Event 0 事件使 PWMx Out A 进入无效状态
[10] MSTPER	Master PWM Period 事件使 PWMx Out A 进入无效状态
[9] MSTCMPD	Master PWM Compare D 事件使 PWMx Out A 进入无效状态
[8] MSTCMPC	Master PWM Compare C 事件使 PWMx Out A 进入无效状态
[7] MSTCMPB	Master PWM Compare B 事件使 PWMx Out A 进入无效状态
[6] MSTCMPA	Master PWM Compare A 事件使 PWMx Out A 进入无效状态
[5] PER	PWMx Roll-Over 事件使 PWMx Out A 进入无效状态
[4] CMPD	PWMx Compare D 事件使 PWMx Out A 进入无效状态
[3] CMPC	PWMx Compare C 事件使 PWMx Out A 进入无效状态

[2] CMPB	**PWMx Compare B 事件使 PWMx Out A 进入无效状态**
[1] CMPA	**PWMx Compare A 事件使 PWMx Out A 进入无效状态**
[0] SST	**Software Set Trigger 事件使 PWMx Out A 进入无效状态**

17.5.2.28 HRPWM 定时器 x 输出 B 置位寄存器（HRPWM_SETxBR）

- **名称：HRPWM PWMx Output B Set Register**
- **偏移地址：0x140+N*0x100[N=0-7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									UPD	RESYNC	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5
									R/W	R/W	R/W	R/W	R/W	R/W	R/W
									0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	EXTEVNT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	PER	CMPD	CMPC	CMPB	CMPA	SST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[22] UPD	**PWMx Update 事件使 PWMx Out B 进入有效状态**
[21] RESYNC	**Sync Input Event 事件使 PWMx Out B 进入有效状态**
[20] EXTEVNT9	**PWMx Event 9 事件使 PWMx Out B 进入有效状态**
[19] EXTEVNT8	**PWMx Event 8 事件使 PWMx Out B 进入有效状态**
[18] EXTEVNT7	**PWMx Event 7 事件使 PWMx Out B 进入有效状态**
[17] EXTEVNT6	**PWMx Event 6 事件使 PWMx Out B 进入有效状态**
[16] EXTEVNT5	**PWMx Event 5 事件使 PWMx Out B 进入有效状态**
[15] EXTEVNT4	**PWMx Event 4 事件使 PWMx Out B 进入有效状态**
[14] EXTEVNT3	**PWMx Event 3 事件使 PWMx Out B 进入有效状态**
[13] EXTEVNT2	**PWMx Event 2 事件使 PWMx Out B 进入有效状态**
[12] EXTEVNT1	**PWMx Event 1 事件使 PWMx Out B 进入有效状态**
[11] EXTEVNT0	**PWMx Event 0 事件使 PWMx Out B 进入有效状态**
[10] MSTPER	**Master PWM Period 事件使 PWMx Out B 进入有效状态**
[9] MSTCMPD	**Master PWM Compare D 事件使 PWMx Out B 进入有效状态**
[8] MSTCMPC	**Master PWM Compare C 事件使 PWMx Out B 进入有效状态**
[7] MSTCMPB	**Master PWM Compare B 事件使 PWMx Out B 进入有效状态**
[6] MSTCMPA	**Master PWM Compare A 事件使 PWMx Out B 进入有效状态**
[5] PER	**PWMx Roll-Over 事件使 PWMx Out B 进入有效状态**

[4] CMPD	**PWMx Compare D 事件使 PWMx Out B 进入有效状态**
[3] CMPC	**PWMx Compare C 事件使 PWMx Out B 进入有效状态**
[2] CMPB	**PWMx Compare B 事件使 PWMx Out B 进入有效状态**
[1] CMPA	**PWMx Compare A 事件使 PWMx Out B 进入有效状态**
[0] SST	**Software Set Trigger 事件使 PWMx Out B 进入有效状态**

17.5.2.29 HRPWM 定时器 x 输出 B 复位寄存器（HRPWM_CLRxBR）

- **名称：HRPWM PWMx Output B Clear Register**
- **偏移地址：0x144+N*0x100[N=0-7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									UPD	RESYNC	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5
									R/W	R/W	R/W	R/W	R/W	R/W	R/W
									0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	EXTEVNT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	PER	CMPD	CMPC	CMPB	CMPA	SST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[22] UPD	**PWMx Update 事件使 PWMx Out B 进入无效状态**
[21] RESYNC	**Sync Input Event 事件使 PWMx Out B 进入无效状态**
[20] EXTEVNT9	**PWMx Event 9 事件使 PWMx Out B 进入无效状态**
[19] EXTEVNT8	**PWMx Event 8 事件使 PWMx Out B 进入无效状态**
[18] EXTEVNT7	**PWMx Event 7 事件使 PWMx Out B 进入无效状态**
[17] EXTEVNT6	**PWMx Event 6 事件使 PWMx Out B 进入无效状态**
[16] EXTEVNT5	**PWMx Event 5 事件使 PWMx Out B 进入无效状态**
[15] EXTEVNT4	**PWMx Event 4 事件使 PWMx Out B 进入无效状态**
[14] EXTEVNT3	**PWMx Event 3 事件使 PWMx Out B 进入无效状态**
[13] EXTEVNT2	**PWMx Event 2 事件使 PWMx Out B 进入无效状态**
[12] EXTEVNT1	**PWMx Event 1 事件使 PWMx Out B 进入无效状态**
[11] EXTEVNT0	**PWMx Event 0 事件使 PWMx Out B 进入无效状态**
[10] MSTPER	**Master PWM Period 事件使 PWMx Out B 进入无效状态**
[9] MSTCMPD	**Master PWM Compare D 事件使 PWMx Out B 进入无效状态**
[8] MSTCMPC	**Master PWM Compare C 事件使 PWMx Out B 进入无效状态**
[7] MSTCMPB	**Master PWM Compare B 事件使 PWMx Out B 进入无效状态**

[6] MSTCMPA	**Master PWM Compare A 事件使 PWMx Out B 进入无效状态**
[5] PER	**PWMx Roll-Over 事件使 PWMx Out B 进入无效状态**
[4] CMPD	**PWMx Compare D 事件使 PWMx Out B 进入无效状态**
[3] CMPC	**PWMx Compare C 事件使 PWMx Out B 进入无效状态**
[2] CMPB	**PWMx Compare B 事件使 PWMx Out B 进入无效状态**
[1] CMPA	**PWMx Compare A 事件使 PWMx Out B 进入无效状态**
[0] SST	**Software Set Trigger 事件使 PWMx Out B 进入无效状态**

17.5.2.30 HRPWM 定时器 x 外部事件寄存器 0（HRPWM_EEFxR0）

- **名称：HRPWM PWMx External Event Register0**
- **偏移地址：0x148+N*0x100[N=0-7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							EE4FLTR				EE4LTCH	EE3FLTR			
							R/W				R/W	R/W			
							0x0				0x0	0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE3LTCH	EE2FLTR				EE2LTCH	EE1FLTR				EE1LTCH	EE0FLTR				EE0LTCH
R/W	R/W				R/W	R/W				R/W	R/W				R/W
0x0	0x0				0x0	0x0				0x0	0x0				0x0

字段	说明
[24:21] EE4FLTR	**Event 4 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[20] EE4LTCH	**Event 4 锁存位** 1：Event 4 出现在消隐期间，则事件被锁存 0：Event 4 出现在消隐期间，则事件不被锁存
[19:16] EE3FLTR	**Event 3 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过

	1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[15] EE3LTCH	**Event 3 锁存位** 1：Event 3 出现在消隐期间，则事件被锁存 0：Event 3 出现在消隐期间，则事件不被锁存
[14:11] EE2FLTR	**Event 2 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[10] EE2LTCH	**Event 2 锁存位** 1：Event 2 出现在消隐期间，则事件被锁存 0：Event 2 出现在消隐期间，则事件不被锁存
[9:6] EE1FLTR	**Event 1 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[5] EE1LTCH	**Event 1 锁存位** 1：Event 1 出现在消隐期间，则事件被锁存 0：Event 1 出现在消隐期间，则事件不被锁存

[4:1] EE0FLTR	**Event 0 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[0] EE0LTCH	**Event 0 锁存位** 1：Event 0 出现在消隐期间，则事件被锁存 0：Event 0 出现在消隐期间，则事件不被锁存

17.5.2.31 HRPWM 定时器 x 外部事件寄存器 1（HRPWM_EEFxR1）

- **名称：HRPWM PWMx External Event Register1**
- **偏移地址：0x14C+N*0x100[N=0-7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							EE9FLTR				EE9LTCH	EE8FLTR			
							R/W				R/W	R/W			
							0x0				0x0	0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE8LTCH	EE7FLTR				EE7LTCH	EE6FLTR				EE6LTCH	EE5FLTR				EE5LTCH
R/W	R/W				R/W	R/W				R/W	R/W				R/W
0x0	0x0				0x0	0x0				0x0	0x0				0x0

字段	说明
[24:21] EE9FLTR	**Event 9 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过

	0000：无消隐 / 加窗滤波功能
[20] EE9LTCH	**Event 9 锁存位** 1：Event 9 出现在消隐期间，则事件被锁存 0：Event 9 出现在消隐期间，则事件不被锁存
[19:16] EE8FLTR	**Event 8 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[15] EE8LTCH	**Event 8 锁存位** 1：Event 8 出现在消隐期间，则事件被锁存 0：Event 8 出现在消隐期间，则事件不被锁存
[14:11] EE7FLTR	**Event 7 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[10] EE7LTCH	**Event 7 锁存位** 1：Event 7 出现在消隐期间，则事件被锁存 0：Event 7 出现在消隐期间，则事件不被锁存
[9:6] EE6FLTR	**Event 6 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过

	0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[5] EE6LTCH	**Event 6 锁存位** 1：Event 6 出现在消隐期间，则事件被锁存 0：Event 6 出现在消隐期间，则事件不被锁存
[4:1] EE5FLTR	**Event 5 滤波位** 1111：从 Up 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1110：从 Down 阶段 Compare B 事件到 Down 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1101：从 Up 阶段 Compare B 事件到 Up 阶段 Compare C 事件期间加窗，加窗期间以外事件无法通过 1100：从 Reset / Roll-Over 事件到 Compare D 事件期间加窗，加窗期间以外事件无法通过 1011：从 Reset / Roll-Over 事件到 Compare C 事件期间加窗，加窗期间以外事件无法通过 1010：从 Reset / Roll-Over 事件到 Compare B 事件期间加窗，加窗期间以外事件无法通过 1001：从 Reset / Roll-Over 事件到 Compare A 事件期间加窗，加窗期间以外事件无法通过 1000：从 Down 阶段 Compare C 事件到 Down 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0111：从 Down 阶段 Compare A 事件到 Down 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0110：从 Up 阶段 Compare C 事件到 Up 阶段 Compare D 事件期间消隐，消隐期间事件无法通过 0101：从 Up 阶段 Compare A 事件到 Up 阶段 Compare B 事件期间消隐，消隐期间事件无法通过 0100：从 Reset / Roll-Over 事件到 Compare D 事件期间消隐，消隐期间事件无法通过 0011：从 Reset / Roll-Over 事件到 Compare C 事件期间消隐，消隐期间事件无法通过 0010：从 Reset / Roll-Over 事件到 Compare B 事件期间消隐，消隐期间事件无法通过 0001：从 Reset / Roll-Over 事件到 Compare A 事件期间消隐，消隐期间事件无法通过 0000：无消隐 / 加窗滤波功能
[0] EE5LTCH	**Event 5 锁存位** 1：Event 5 出现在消隐期间，则事件被锁存 0：Event 5 出现在消隐期间，则事件不被锁存

17.5.2.32 HRPWM 定时器 x 外部事件寄存器 2（HRPWM_EEFxR2）

- **名称：HRPWM PWMx External Event Register2**
- **偏移地址：0x150+N*0x100[N=0-7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		EEVACNT						EEVASEL					EEVARSTM	EEVACRES	EEVACE
		R/W						R/W					R/W	R/WAC	R/W
		0x0						0x0					0x0	0x0	0x0

字段	说明
[13:8] EEVACNT	**Event A 计数阈值** 当事件总数达到（EEVACNT＋1）时认为事件有效
[7:4] EEVASEL	**Event A 来源选择** Others：Reserved 9：Event 9 为 Event A 8：Event 8 为 Event A 7：Event 7 为 Event A 6：Event 6 为 Event A 5：Event 5 为 Event A

	4：Event 4 为 Event A 3：Event 3 为 Event A 2：Event 2 为 Event A 1：Event 1 为 Event A 0：Event 0 为 Event A
[2] EEVARSTM	**Event A 计数复位模式** 1：Event A 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Event A 计数在 Reset / Roll-Over 事件处复位
[1] EEVACRES	**Event A 计数复位** 1：Event A 计数复位 0：Event A 计数不复位[SEP]此位将 Event A 计数复位，软件写 1 后等复位完成后硬件自动清 0
[0] EEVACE	**Event A 计数使能** 1：Event A 计数使能 0：Event A 计数不使能

17.5.2.33 HRPWM 定时器 0 复位寄存器（HRPWM_RST0R）

- **名称：HRPWM PWM0 Reset Register**
- **偏移地址：0x154+N*0x100[N=0]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV4CMPD	SLV4CMPB	SLV4CMPA	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV5CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[30] SLV4CMPD	**PWM4 Compare D 事件使 PWMx 计数器复位**
[29] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[28] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**
[27] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[26] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[25] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[24] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[23] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[22] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[21] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[20] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[19] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**

[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.34 HRPWM 定时器 1 复位寄存器（HRPWM_RST1R）

- **名称：HRPWM PWM1 Reset Register**
- **偏移地址：0x154+N*0x100[N=1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV4CMPD	SLV4CMPB	SLV4CMPA	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV5CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[30] SLV4CMPD	**PWM4 Compare D 事件使 PWMx 计数器复位**

[29] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[28] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**
[27] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[26] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[25] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[24] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[23] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[22] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**
[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.35 HRPWM 定时器 2 复位寄存器（HRPWM_RST2R）

- 名称：HRPWM PWM2 Reset Register
- 偏移地址：0x154+N*0x100[N=2]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV4CMPD	SLV4CMPB	SLV4CMPA	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV5CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[30] SLV4CMPD	**PWM4 Compare D 事件使 PWMx 计数器复位**
[29] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[28] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**
[27] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[26] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[25] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[24] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[23] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[22] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**
[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**

字段	说明
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.36 HRPWM 定时器 3 复位寄存器（HRPWM_RST3R）

- **名称：HRPWM PWM3 Reset Register**
- **偏移地址：0x154+N*0x100[N=3]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV4CMPD	SLV4CMPB	SLV4CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV5CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[30] SLV4CMPD	**PWM4 Compare D 事件使 PWMx 计数器复位**
[29] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[28] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**
[27] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[26] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[25] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[24] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[23] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[22] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**

[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.37 HRPWM 定时器 4 复位寄存器（HRPWM_RST4R）

- 名称：HRPWM PWM4 Reset Register
- 偏移地址：0x154+N*0x100[N=4]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV5CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[30] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[29] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[28] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[27] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[26] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[25] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[24] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[23] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[22] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**
[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**

字段	说明
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.38 HRPWM 定时器 5 复位寄存器（HRPWM_RST5R）

- **名称：HRPWM PWM5 Reset Register**
- **偏移地址：0x154+N*0x100[N=5]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV4CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[30] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[29] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[28] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[27] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[26] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[25] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[24] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[23] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[22] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**

[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**

17.5.2.39 HRPWM 定时器 6 复位寄存器（HRPWM_RST6R）

- 名称：HRPWM PWM6 Reset Register
- 偏移地址：0x154+N*0x100[N=6]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV4CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[30] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[29] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[28] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[27] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[26] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[25] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[24] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[23] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[22] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**
[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**

字段	说明
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**

17.5.2.40 HRPWM 定时器 7 复位寄存器（HRPWM_RST7R）

- **名称：HRPWM PWM7 Reset Register**
- **偏移地址：0x154+N*0x100[N=7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV3CMPD	SLV3CMPB	SLV3CMPA	SLV2CMPD	SLV2CMPB	SLV2CMPA	SLV1CMPD	SLV1CMPB	SLV1CMPA	SLV0CMPD	SLV0CMPB	SLV0CMPA	SLVCMPD	SLVCMPB	SLVUPD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	MSTPER	MSTCMPD	MSTCMPC	MSTCMPB	MSTCMPA	SLV4CMPA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件使 PWMx 计数器复位**
[30] SLV3CMPD	**PWM3 Compare D 事件使 PWMx 计数器复位**
[29] SLV3CMPB	**PWM3 Compare B 事件使 PWMx 计数器复位**
[28] SLV3CMPA	**PWM3 Compare A 事件使 PWMx 计数器复位**
[27] SLV2CMPD	**PWM2 Compare D 事件使 PWMx 计数器复位**
[26] SLV2CMPB	**PWM2 Compare B 事件使 PWMx 计数器复位**
[25] SLV2CMPA	**PWM2 Compare A 事件使 PWMx 计数器复位**
[24] SLV1CMPD	**PWM1 Compare D 事件使 PWMx 计数器复位**
[23] SLV1CMPB	**PWM1 Compare B 事件使 PWMx 计数器复位**
[22] SLV1CMPA	**PWM1 Compare A 事件使 PWMx 计数器复位**
[21] SLV0CMPD	**PWM0 Compare D 事件使 PWMx 计数器复位**

[20] SLV0CMPB	**PWM0 Compare B 事件使 PWMx 计数器复位**
[19] SLV0CMPA	**PWM0 Compare A 事件使 PWMx 计数器复位**
[18] SLVCMPD	**PWMx Compare D 事件使 PWMx 计数器复位**
[17] SLVCMPB	**PWMx Compare B 事件使 PWMx 计数器复位**
[16] SLVUPD	**PWMx Update 事件使 PWMx 计数器复位**
[15] EXTEVT9	**PWMx Event 9 事件使 PWMx 计数器复位**
[14] EXTEVT8	**PWMx Event 8 事件使 PWMx 计数器复位**
[13] EXTEVT7	**PWMx Event 7 事件使 PWMx 计数器复位**
[12] EXTEVT6	**PWMx Event 6 事件使 PWMx 计数器复位**
[11] EXTEVT5	**PWMx Event 5 事件使 PWMx 计数器复位**
[10] EXTEVT4	**PWMx Event 4 事件使 PWMx 计数器复位**
[9] EXTEVT3	**PWMx Event 3 事件使 PWMx 计数器复位**
[8] EXTEVT2	**PWMx Event 2 事件使 PWMx 计数器复位**
[7] EXTEVT1	**PWMx Event 1 事件使 PWMx 计数器复位**
[6] EXTEVT0	**PWMx Event 0 事件使 PWMx 计数器复位**
[5] MSTPER	**Master PWM Period 事件使 PWMx 计数器复位**
[4] MSTCMPD	**Master PWM Compare D 事件使 PWMx 计数器复位**
[3] MSTCMPC	**Master PWM Compare C 事件使 PWMx 计数器复位**
[2] MSTCMPB	**Master PWM Compare B 事件使 PWMx 计数器复位**
[1] MSTCMPA	**Master PWM Compare A 事件使 PWMx 计数器复位**
[0] SLV4CMPA	**PWM4 Compare A 事件使 PWMx 计数器复位**

17.5.2.41 HRPWM 定时器 0 复位扩展寄存器（HRPWM_RST0ER）

- 名称：HRPWM PWM0 Reset Extended Register
- 偏移地址：0x158+N*0x100[N=0]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7C MPD	SLV7C MPB	SLV7C MPA	SLV6C MPD	SLV6C MPB	SLV6C MPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	PWM7 Compare D 事件使 PWMx 计数器复位
[4] SLV7CMPB	PWM7 Compare B 事件使 PWMx 计数器复位
[3] SLV7CMPA	PWM7 Compare A 事件使 PWMx 计数器复位
[2] SLV6CMPD	PWM6 Compare D 事件使 PWMx 计数器复位
[1] SLV6CMPB	PWM6 Compare B 事件使 PWMx 计数器复位
[0] SLV6CMPA	PWM6 Compare A 事件使 PWMx 计数器复位

17.5.2.42 HRPWM 定时器 1 复位扩展寄存器（HRPWM_RST1ER）

- 名称：HRPWM PWM1 Reset Extended Register
- 偏移地址：0x158+N*0x100[N=1]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7C MPD	SLV7C MPB	SLV7C MPA	SLV6C MPD	SLV6C MPB	SLV6C MPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	PWM7 Compare D 事件使 PWMx 计数器复位
[4] SLV7CMPB	PWM7 Compare B 事件使 PWMx 计数器复位
[3] SLV7CMPA	PWM7 Compare A 事件使 PWMx 计数器复位
[2] SLV6CMPD	PWM6 Compare D 事件使 PWMx 计数器复位
[1] SLV6CMPB	PWM6 Compare B 事件使 PWMx 计数器复位

[0] SLV6CMPA	**PWM6 Compare A 事件使 PWMx 计数器复位**

17.5.2.43 HRPWM 定时器 2 复位扩展寄存器（HRPWM_RST2ER）

- **名称：HRPWM PWM2 Reset Extended Register**
- **偏移地址：0x158+N*0x100[N=2]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7C MPD	SLV7C MPB	SLV7C MPA	SLV6C MPD	SLV6C MPB	SLV6C MPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	**PWM7 Compare D 事件使 PWMx 计数器复位**
[4] SLV7CMPB	**PWM7 Compare B 事件使 PWMx 计数器复位**
[3] SLV7CMPA	**PWM7 Compare A 事件使 PWMx 计数器复位**
[2] SLV6CMPD	**PWM6 Compare D 事件使 PWMx 计数器复位**
[1] SLV6CMPB	**PWM6 Compare B 事件使 PWMx 计数器复位**
[0] SLV6CMPA	**PWM6 Compare A 事件使 PWMx 计数器复位**

17.5.2.44 HRPWM 定时器 3 复位扩展寄存器（HRPWM_RST3ER）

- **名称：HRPWM PWM3 Reset Extended Register**
- **偏移地址：0x158+N*0x100[N=3]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7C MPD	SLV7C MPB	SLV7C MPA	SLV6C MPD	SLV6C MPB	SLV6C MPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	**PWM7 Compare D 事件使 PWMx 计数器复位**
[4] SLV7CMPB	**PWM7 Compare B 事件使 PWMx 计数器复位**
[3] SLV7CMPA	**PWM7 Compare A 事件使 PWMx 计数器复位**

[2] SLV6CMPD	**PWM6 Compare D 事件使 PWMx 计数器复位**
[1] SLV6CMPB	**PWM6 Compare B 事件使 PWMx 计数器复位**
[0] SLV6CMPA	**PWM6 Compare A 事件使 PWMx 计数器复位**

17.5.2.45 HRPWM 定时器 4 复位扩展寄存器（HRPWM_RST4ER）

- **名称：HRPWM PWM4 Reset Extended Register**
- **偏移地址：0x158+N*0x100[N=4]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7CMPD	SLV7CMPB	SLV7CMPA	SLV6CMPD	SLV6CMPB	SLV6CMPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	**PWM7 Compare D 事件使 PWMx 计数器复位**
[4] SLV7CMPB	**PWM7 Compare B 事件使 PWMx 计数器复位**
[3] SLV7CMPA	**PWM7 Compare A 事件使 PWMx 计数器复位**
[2] SLV6CMPD	**PWM6 Compare D 事件使 PWMx 计数器复位**
[1] SLV6CMPB	**PWM6 Compare B 事件使 PWMx 计数器复位**
[0] SLV6CMPA	**PWM6 Compare A 事件使 PWMx 计数器复位**

17.5.2.46 HRPWM 定时器 5 复位扩展寄存器（HRPWM_RST5ER）

- **名称：HRPWM PWM5 Reset Extended Register**
- **偏移地址：0x158+N*0x100[N=5]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7CMPD	SLV7CMPB	SLV7CMPA	SLV6CMPD	SLV6CMPB	SLV6CMPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	**PWM7 Compare D 事件使 PWMx 计数器复位**

[4] SLV7CMPB	**PWM7 Compare B 事件使 PWMx 计数器复位**
[3] SLV7CMPA	**PWM7 Compare A 事件使 PWMx 计数器复位**
[2] SLV6CMPD	**PWM6 Compare D 事件使 PWMx 计数器复位**
[1] SLV6CMPB	**PWM6 Compare B 事件使 PWMx 计数器复位**
[0] SLV6CMPA	**PWM6 Compare A 事件使 PWMx 计数器复位**

17.5.2.47 HRPWM 定时器 6 复位扩展寄存器（HRPWM_RST6ER）

- **名称：HRPWM PWM6 Reset Extended Register**
- **偏移地址：0x158+N*0x100[N=6]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV7C MPD	SLV7C MPB	SLV7C MPA	SLV5C MPD	SLV5C MPB	SLV5C MPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV7CMPD	**PWM7 Compare D 事件使 PWMx 计数器复位**
[4] SLV7CMPB	**PWM7 Compare B 事件使 PWMx 计数器复位**
[3] SLV7CMPA	**PWM7 Compare A 事件使 PWMx 计数器复位**
[2] SLV5CMPD	**PWM5 Compare D 事件使 PWMx 计数器复位**
[1] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.48 HRPWM 定时器 7 复位扩展寄存器（HRPWM_RST7ER）

- 名称：HRPWM PWM7 Reset Extended Register
- 偏移地址：0x158+N*0x100[N=7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SLV6C MPD	SLV6C MPB	SLV6C MPA	SLV5C MPD	SLV5C MPB	SLV5C MPA
										R/W	R/W	R/W	R/W	R/W	R/W
										0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[5] SLV6CMPD	**PWM6 Compare D 事件使 PWMx 计数器复位**
[4] SLV6CMPB	**PWM6 Compare B 事件使 PWMx 计数器复位**
[3] SLV6CMPA	**PWM6 Compare A 事件使 PWMx 计数器复位**
[2] SLV5CMPD	**PWM5 Compare D 事件使 PWMx 计数器复位**
[1] SLV5CMPB	**PWM5 Compare B 事件使 PWMx 计数器复位**
[0] SLV5CMPA	**PWM5 Compare A 事件使 PWMx 计数器复位**

17.5.2.49 HRPWM 定时器 x 斩波寄存器（HRPWM_CHPxR）

- 名称：HRPWM PWMx Chopper Register
- 偏移地址：0x15C+N*0x100[N=0-7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRPW							CARDTY					CARFRQ			
R/W							R/W					R/W			
0x0							0x0					0x0			

字段	说明
[15:12] STRPW	**PWMx 启动脉冲宽度** 此位定义在输出信号的上升沿之后的初始脉冲宽度 t1STPW ＝ （STRPW＋1）＊16＊THRPWM 1111：1.6 us（16/10MHz） …… 0000：100 ns（1/10MHz）
[8:6] CARDTY	**PWMx 斩波占空比值** 此位定义斩波信号的占空比 111：7/8 ……

	000：0/8（仅存在第一个脉冲）
[3:0] CARFRQ	**PWMx 载波频率值** 此位定义载波频率为 FCHPFRQ＝FHRPWM / （16*（CARFRQ＋1）） 1111：625 KHz（FHRPWM / 256） …… 0000：10 MHz（FHRPWM / 16）

17.5.2.50 HRPWM 定时器 0 捕获 A 控制寄存器（HRPWM_CAPA0CR）

- **名称：HRPWM PWM0 Capture A Control Register**
- **偏移地址：0x160+N*0x100[N=0]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA	SLV4C MPB	SLV4C MPA	SLV4C LRA	SLV4S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV2C MPB	SLV2C MPA	SLV2C LRA	SLV2S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture A 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[23] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[22] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[21] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[20] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[19] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[18] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[17] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**
[16] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[15] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[14]	**PWM1 Compare A 事件触发 Capture A 事件**

SLV1CMPA	
[13] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[12] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.51 HRPWM 定时器 1 捕获 A 控制寄存器（HRPWM_CAPA1CR）

- **名称：HRPWM PWM1 Capture A Control Register**
- **偏移地址：0x160+N*0x100[N=1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA	SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[25]	**PWM4 OutA Clr 事件触发 Capture A 事件**

SLV4CLRA	
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[23] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[22] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[21] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[20] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[19] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[18] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[17] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**
[16] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.52 HRPWM 定时器 2 捕获 A 控制寄存器（HRPWM_CAPA2CR）

- 名称：HRPWM PWM2 Capture A Control Register
- 偏移地址：0x160+N*0x100[N=2]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA	SLV4C MPB	SLV4C MPA	SLV4C LRA	SLV4S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV1C MPB	SLV1C MPA	SLV1C LRA	SLV1S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture A 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[23] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[22] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[21] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[20] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture A 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**

[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.53 HRPWM 定时器 3 捕获 A 控制寄存器（HRPWM_CAPA3CR）

- **名称：HRPWM PWM3 Capture A Control Register**
- **偏移地址：0x160+N*0x100[N=3]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA	SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA	SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture A 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**

[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture A 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.54 HRPWM 定时器 4 捕获 A 控制寄存器（HRPWM_CAPA4CR）

- 名称：HRPWM PWM4 Capture A Control Register
- 偏移地址：0x160+N*0x100[N=4]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV2C MPB	SLV2C MPA	SLV2C LRA	SLV2S ETA	SLV1C MPB	SLV1C MPA	SLV1C LRA	SLV1S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**
[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture A 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**

[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.55 HRPWM 定时器 5 捕获 A 控制寄存器（HRPWM_CAPA5CR）

- **名称：HRPWM PWM5 Capture A Control Register**
- **偏移地址：0x160+N*0x100[N=5]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA	SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[30] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[29] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture A 事件**
[28] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**

[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture A 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.56 HRPWM 定时器 6 捕获 A 控制寄存器（HRPWM_CAPA6CR）

- 名称：HRPWM PWM6 Capture A Control Register
- 偏移地址：0x160+N*0x100[N=6]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4C MPB	SLV4C MPA	SLV4C LRA	SLV4S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV2C MPB	SLV2C MPA	SLV2C LRA	SLV2S ETA	SLV1C MPB	SLV1C MPA	SLV1C LRA	SLV1S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[30] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[29] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture A 事件**
[28] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**
[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture A 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**

[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.57 HRPWM 定时器 7 捕获 A 控制寄存器（HRPWM_CAPA7CR）

- **名称：HRPWM PWM7 Capture A Control Register**
- **偏移地址：0x160+N*0x100[N=7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA	SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件触发 Capture A 事件**
[30] SLV4CMPA	**PWM4 Compare A 事件触发 Capture A 事件**
[29] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture A 事件**
[28] SLV4SETA	**PWM4 OutA Set 事件触发 Capture A 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture A 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture A 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture A 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture A 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture A 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture A 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture A 事件**

[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture A 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture A 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture A 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture A 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture A 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture A 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture A 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture A 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture A 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture A 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture A 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture A 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture A 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture A 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture A 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture A 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture A 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture A 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture A 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture A 事件**
[0] SWCPT	**Software Capture 事件触发 Capture A 事件**

17.5.2.58 HRPWM 定时器 0 捕获 A 扩展寄存器（HRPWM_CAPA0CER）

- 名称：HRPWM PWM0 Capture A Control Extended Register
- 偏移地址：0x164+N*0x100[N=0]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture A 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture A 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture A 事件
[4] SLV7SETA	PWM7 OutA Set 事件触发 Capture A 事件
[3] SLV6CMPB	PWM6 Compare B 事件触发 Capture A 事件
[2] SLV6CMPA	PWM6 Compare A 事件触发 Capture A 事件
[1] SLV6CLRA	PWM6 OutA Clr 事件触发 Capture A 事件
[0] SLV6SETA	PWM6 OutA Set 事件触发 Capture A 事件

17.5.2.59 HRPWM 定时器 1 捕获 A 扩展寄存器（HRPWM_CAPA1CER）

- 名称：HRPWM PWM1 Capture A Control Extended Register
- 偏移地址：0x164+N*0x100[N=1]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture A 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture A 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture A 事件

[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture A 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture A 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture A 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture A 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture A 事件**

17.5.2.60 HRPWM 定时器 2 捕获 A 扩展寄存器（HRPWM_CAPA2CER）

- **名称：HRPWM PWM2 Capture A Control Extended Register**
- **偏移地址：0x164+N*0x100[N=2]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	**PWM7 Compare B 事件触发 Capture A 事件**
[6] SLV7CMPA	**PWM7 Compare A 事件触发 Capture A 事件**
[5] SLV7CLRA	**PWM7 OutA Clr 事件触发 Capture A 事件**
[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture A 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture A 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture A 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture A 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture A 事件**

17.5.2.61 HRPWM 定时器 3 捕获 A 扩展寄存器（HRPWM_CAPA3CER）

- 名称：HRPWM PWM3 Capture A Control Extended Register
- 偏移地址：0x164+N*0x100[N=3]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture A 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture A 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture A 事件
[4] SLV7SETA	PWM7 OutA Set 事件触发 Capture A 事件
[3] SLV6CMPB	PWM6 Compare B 事件触发 Capture A 事件
[2] SLV6CMPA	PWM6 Compare A 事件触发 Capture A 事件
[1] SLV6CLRA	PWM6 OutA Clr 事件触发 Capture A 事件
[0] SLV6SETA	PWM6 OutA Set 事件触发 Capture A 事件

17.5.2.62 HRPWM 定时器 4 捕获 A 扩展寄存器（HRPWM_CAPA4CER）

- 名称：HRPWM PWM4 Capture A Control Extended Register
- 偏移地址：0x164+N*0x100[N=4]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture A 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture A 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture A 事件

[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture A 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture A 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture A 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture A 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture A 事件**

17.5.2.63 HRPWM 定时器 5 捕获 A 扩展寄存器（HRPWM_CAPA5CER）

- **名称：HRPWM PWM5 Capture A Control Extended Register**
- **偏移地址：0x164+N*0x100[N=5]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	**PWM7 Compare B 事件触发 Capture A 事件**
[6] SLV7CMPA	**PWM7 Compare A 事件触发 Capture A 事件**
[5] SLV7CLRA	**PWM7 OutA Clr 事件触发 Capture A 事件**
[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture A 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture A 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture A 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture A 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture A 事件**

17.5.2.64 HRPWM 定时器 6 捕获 A 扩展寄存器（HRPWM_CAPA6CER）

- 名称：HRPWM PWM6 Capture A Control Extended Register
- 偏移地址：0x164+N*0x100[N=6]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	**PWM7 Compare B 事件触发 Capture A 事件**
[6] SLV7CMPA	**PWM7 Compare A 事件触发 Capture A 事件**
[5] SLV7CLRA	**PWM7 OutA Clr 事件触发 Capture A 事件**
[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture A 事件**
[3] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[2] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[1] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[0] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**

17.5.2.65 HRPWM 定时器 7 捕获 A 扩展寄存器（HRPWM_CAPA7CER）

- 名称：HRPWM PWM7 Capture A Control Extended Register
- 偏移地址：0x164+N*0x100[N=7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA	SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV6CMPB	**PWM6 Compare B 事件触发 Capture A 事件**
[6] SLV6CMPA	**PWM6 Compare A 事件触发 Capture A 事件**
[5] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture A 事件**

[4] SLV6SETA	**PWM6 OutA Set 事件触发 Capture A 事件**
[3] SLV5CMPB	**PWM5 Compare B 事件触发 Capture A 事件**
[2] SLV5CMPA	**PWM5 Compare A 事件触发 Capture A 事件**
[1] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture A 事件**
[0] SLV5SETA	**PWM5 OutA Set 事件触发 Capture A 事件**

17.5.2.66 HRPWM 定时器 0 捕获 B 控制寄存器（HRPWM_CAPB0CR）

- **名称：HRPWM PWM0 Capture B Control Register**
- **偏移地址：0x168+N*0x100[N=0]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA	SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture B 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture B 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture B 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture B 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[23] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[22] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[21] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[20] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[19] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[18] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[17] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**
[16] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**

[15] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[14] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[13] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[12] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.67 HRPWM 定时器 1 捕获 B 控制寄存器（HRPWM_CAPB1CR）

- **名称：HRPWM PWM1 Capture B Control Register**
- **偏移地址：0x168+N*0x100[N=1]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA	SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture B 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture B 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture B 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture B 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**

[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[23] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[22] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[21] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[20] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[19] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[18] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[17] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**
[16] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.68 HRPWM 定时器 2 捕获 B 控制寄存器（HRPWM_CAPB2CR）

- 名称：HRPWM PWM2 Capture B Control Register
- 偏移地址：0x168+N*0x100[N=2]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA	SLV4C MPB	SLV4C MPA	SLV4C LRA	SLV4S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV1C MPB	SLV1C MPA	SLV1C LRA	SLV1S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture B 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture B 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture B 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture B 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[23] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[22] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[21] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[20] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**

[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.69 HRPWM 定时器 3 捕获 B 控制寄存器（HRPWM_CAPB3CR）

- **名称：HRPWM PWM3 Capture B Control Register**
- **偏移地址：0x168+N*0x100[N=3]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA	SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA	SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture B 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture B 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture B 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture B 事件**
[27] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**
[26] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[25] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[24] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**

[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.70 HRPWM 定时器 4 捕获 B 控制寄存器（HRPWM_CAPB4CR）

- 名称：HRPWM PWM4 Capture B Control Register
- 偏移地址：0x168+N*0x100[N=4]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV5C MPB	SLV5C MPA	SLV5C LRA	SLV5S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV2C MPB	SLV2C MPA	SLV2C LRA	SLV2S ETA	SLV1C MPB	SLV1C MPA	SLV1C LRA	SLV1S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV5CMPB	**PWM5 Compare B 事件触发 Capture B 事件**
[30] SLV5CMPA	**PWM5 Compare A 事件触发 Capture B 事件**
[29] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture B 事件**
[28] SLV5SETA	**PWM5 OutA Set 事件触发 Capture B 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**
[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**

[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.71 HRPWM 定时器 5 捕获 B 控制寄存器（HRPWM_CAPB5CR）

- **名称：HRPWM PWM5 Capture B Control Register**
- **偏移地址：0x168+N*0x100[N=5]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA	SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**
[30] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[29] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[28] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**

[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.72 HRPWM 定时器 6 捕获 B 控制寄存器（HRPWM_CAPB6CR）

- 名称：HRPWM PWM6 Capture B Control Register
- 偏移地址：0x168+N*0x100[N=6]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4C MPB	SLV4C MPA	SLV4C LRA	SLV4S ETA	SLV3C MPB	SLV3C MPA	SLV3C LRA	SLV3S ETA	SLV2C MPB	SLV2C MPA	SLV2C LRA	SLV2S ETA	SLV1C MPB	SLV1C MPA	SLV1C LRA	SLV1S ETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0C MPB	SLV0C MPA	SLV0C LRA	SLV0S ETA	EXTE VT9	EXTE VT8	EXTE VT7	EXTE VT6	EXTE VT5	EXTE VT4	EXTE VT3	EXTE VT2	EXTE VT1	EXTE VT0	UPDC PT	SWCP T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**
[30] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[29] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[28] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**
[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**

[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.73 HRPWM 定时器 7 捕获 B 控制寄存器（HRPWM_CAPB7CR）

- **名称：HRPWM PWM7 Capture B Control Register**
- **偏移地址：0x168+N*0x100[N=7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLV4CMPB	SLV4CMPA	SLV4CLRA	SLV4SETA	SLV3CMPB	SLV3CMPA	SLV3CLRA	SLV3SETA	SLV2CMPB	SLV2CMPA	SLV2CLRA	SLV2SETA	SLV1CMPB	SLV1CMPA	SLV1CLRA	SLV1SETA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV0CMPB	SLV0CMPA	SLV0CLRA	SLV0SETA	EXTEVT9	EXTEVT8	EXTEVT7	EXTEVT6	EXTEVT5	EXTEVT4	EXTEVT3	EXTEVT2	EXTEVT1	EXTEVT0	UPDCPT	SWCPT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] SLV4CMPB	**PWM4 Compare B 事件触发 Capture B 事件**
[30] SLV4CMPA	**PWM4 Compare A 事件触发 Capture B 事件**
[29] SLV4CLRA	**PWM4 OutA Clr 事件触发 Capture B 事件**
[28] SLV4SETA	**PWM4 OutA Set 事件触发 Capture B 事件**
[27] SLV3CMPB	**PWM3 Compare B 事件触发 Capture B 事件**
[26] SLV3CMPA	**PWM3 Compare A 事件触发 Capture B 事件**
[25] SLV3CLRA	**PWM3 OutA Clr 事件触发 Capture B 事件**
[24] SLV3SETA	**PWM3 OutA Set 事件触发 Capture B 事件**
[23] SLV2CMPB	**PWM2 Compare B 事件触发 Capture B 事件**
[22] SLV2CMPA	**PWM2 Compare A 事件触发 Capture B 事件**
[21] SLV2CLRA	**PWM2 OutA Clr 事件触发 Capture B 事件**

[20] SLV2SETA	**PWM2 OutA Set 事件触发 Capture B 事件**
[19] SLV1CMPB	**PWM1 Compare B 事件触发 Capture B 事件**
[18] SLV1CMPA	**PWM1 Compare A 事件触发 Capture B 事件**
[17] SLV1CLRA	**PWM1 OutA Clr 事件触发 Capture B 事件**
[16] SLV1SETA	**PWM1 OutA Set 事件触发 Capture B 事件**
[15] SLV0CMPB	**PWM0 Compare B 事件触发 Capture B 事件**
[14] SLV0CMPA	**PWM0 Compare A 事件触发 Capture B 事件**
[13] SLV0CLRA	**PWM0 OutA Clr 事件触发 Capture B 事件**
[12] SLV0SETA	**PWM0 OutA Set 事件触发 Capture B 事件**
[11] EXTEVT9	**PWMx Event 9 事件触发 Capture B 事件**
[10] EXTEVT8	**PWMx Event 8 事件触发 Capture B 事件**
[9] EXTEVT7	**PWMx Event 7 事件触发 Capture B 事件**
[8] EXTEVT6	**PWMx Event 6 事件触发 Capture B 事件**
[7] EXTEVT5	**PWMx Event 5 事件触发 Capture B 事件**
[6] EXTEVT4	**PWMx Event 4 事件触发 Capture B 事件**
[5] EXTEVT3	**PWMx Event 3 事件触发 Capture B 事件**
[4] EXTEVT2	**PWMx Event 2 事件触发 Capture B 事件**
[3] EXTEVT1	**PWMx Event 1 事件触发 Capture B 事件**
[2] EXTEVT0	**PWMx Event 0 事件触发 Capture B 事件**
[1] UPDCPT	**PWMx Update 事件触发 Capture B 事件**
[0] SWCPT	**Software Capture 事件触发 Capture B 事件**

17.5.2.74 HRPWM 定时器 0 捕获 B 扩展寄存器（HRPWM_CAPB0CER）

- 名称：HRPWM PWM0 Capture B Control Extended Register
- 偏移地址：0x16C+N*0x100[N=0]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture B 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture B 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture B 事件
[4] SLV7SETA	PWM7 OutA Set 事件触发 Capture B 事件
[3] SLV6CMPB	PWM6 Compare B 事件触发 Capture B 事件
[2] SLV6CMPA	PWM6 Compare A 事件触发 Capture B 事件
[1] SLV6CLRA	PWM6 OutA Clr 事件触发 Capture B 事件
[0] SLV6SETA	PWM6 OutA Set 事件触发 Capture B 事件

17.5.2.75 HRPWM 定时器 1 捕获 B 扩展寄存器（HRPWM_CAPB1CER）

- 名称：HRPWM PWM1 Capture B Control Extended Register
- 偏移地址：0x16C+N*0x100[N=1]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture B 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture B 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture B 事件

[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture B 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture B 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture B 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture B 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture B 事件**

17.5.2.76 HRPWM 定时器 2 捕获 B 扩展寄存器（HRPWM_CAPB2CER）

- **名称：HRPWM PWM2 Capture B Control Extended Register**
- **偏移地址：0x16C+N*0x100[N=2]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	**PWM7 Compare B 事件触发 Capture B 事件**
[6] SLV7CMPA	**PWM7 Compare A 事件触发 Capture B 事件**
[5] SLV7CLRA	**PWM7 OutA Clr 事件触发 Capture B 事件**
[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture B 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture B 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture B 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture B 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture B 事件**

17.5.2.77 HRPWM 定时器 3 捕获 B 扩展寄存器（HRPWM_CAPB3CER）

- 名称：HRPWM PWM3 Capture B Control Extended Register
- 偏移地址：0x16C+N*0x100[N=3]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture B 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture B 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture B 事件
[4] SLV7SETA	PWM7 OutA Set 事件触发 Capture B 事件
[3] SLV6CMPB	PWM6 Compare B 事件触发 Capture B 事件
[2] SLV6CMPA	PWM6 Compare A 事件触发 Capture B 事件
[1] SLV6CLRA	PWM6 OutA Clr 事件触发 Capture B 事件
[0] SLV6SETA	PWM6 OutA Set 事件触发 Capture B 事件

17.5.2.78 HRPWM 定时器 4 捕获 B 扩展寄存器（HRPWM_CAPB4CER）

- 名称：HRPWM PWM4 Capture B Control Extended Register
- 偏移地址：0x16C+N*0x100[N=4]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7C MPB	SLV7C MPA	SLV7C LRA	SLV7S ETA	SLV6C MPB	SLV6C MPA	SLV6C LRA	SLV6S ETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture B 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture B 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture B 事件

[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture B 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture B 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture B 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture B 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture B 事件**

17.5.2.79 HRPWM 定时器 5 捕获 B 扩展寄存器（HRPWM_CAPB5CER）

- **名称：HRPWM PWM5 Capture B Control Extended Register**
- **偏移地址：0x16C+N*0x100[N=5]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	**PWM7 Compare B 事件触发 Capture B 事件**
[6] SLV7CMPA	**PWM7 Compare A 事件触发 Capture B 事件**
[5] SLV7CLRA	**PWM7 OutA Clr 事件触发 Capture B 事件**
[4] SLV7SETA	**PWM7 OutA Set 事件触发 Capture B 事件**
[3] SLV6CMPB	**PWM6 Compare B 事件触发 Capture B 事件**
[2] SLV6CMPA	**PWM6 Compare A 事件触发 Capture B 事件**
[1] SLV6CLRA	**PWM6 OutA Clr 事件触发 Capture B 事件**
[0] SLV6SETA	**PWM6 OutA Set 事件触发 Capture B 事件**

17.5.2.80 HRPWM 定时器 6 捕获 B 扩展寄存器（HRPWM_CAPB6CER）

- 名称：HRPWM PWM6 Capture B Control Extended Register
- 偏移地址：0x16C+N*0x100[N=6]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV7CMPB	SLV7CMPA	SLV7CLRA	SLV7SETA	SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV7CMPB	PWM7 Compare B 事件触发 Capture B 事件
[6] SLV7CMPA	PWM7 Compare A 事件触发 Capture B 事件
[5] SLV7CLRA	PWM7 OutA Clr 事件触发 Capture B 事件
[4] SLV7SETA	PWM7 OutA Set 事件触发 Capture B 事件
[3] SLV5CMPB	PWM5 Compare B 事件触发 Capture B 事件
[2] SLV5CMPA	PWM5 Compare A 事件触发 Capture B 事件
[1] SLV5CLRA	PWM5 OutA Clr 事件触发 Capture B 事件
[0] SLV5SETA	PWM5 OutA Set 事件触发 Capture B 事件

17.5.2.81 HRPWM 定时器 7 捕获 B 扩展寄存器（HRPWM_CAPB7CER）

- 名称：HRPWM PWM7 Capture B Control Extended Register
- 偏移地址：0x16C+N*0x100[N=7]
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SLV6CMPB	SLV6CMPA	SLV6CLRA	SLV6SETA	SLV5CMPB	SLV5CMPA	SLV5CLRA	SLV5SETA
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] SLV6CMPB	PWM6 Compare B 事件触发 Capture B 事件
[6] SLV6CMPA	PWM6 Compare A 事件触发 Capture B 事件
[5] SLV6CLRA	PWM6 OutA Clr 事件触发 Capture B 事件

[4] SLV6SETA	**PWM6 OutA Set 事件触发 Capture B 事件**
[3] SLV5CMPB	**PWM5 Compare B 事件触发 Capture B 事件**
[2] SLV5CMPA	**PWM5 Compare A 事件触发 Capture B 事件**
[1] SLV5CLRA	**PWM5 OutA Clr 事件触发 Capture B 事件**
[0] SLV5SETA	**PWM5 OutA Set 事件触发 Capture B 事件**

17.5.2.82 HRPWM 定时器 x 输出控制寄存器（HRPWM_OUTxR）

- **名称：HRPWM PWMx Output Register**
- **偏移地址：0x170+N*0x100[N=0-7]**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTEN	DLYPRTEN	DLYPRT				BIAR			DIDLB	IDLEMB	CHPB	IDLESB	FAULTB		POLB
R/W	R/W	R/W				R/W			R/W	R/W	R/W	R/W	R/W		R/W
0x0	0x0	0x0				0x0			0x0	0x0	0x0	0x0	0x0		0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									DIDLA	IDLEMA	CHPA	IDLESA	FAULTA		POLA
									R/W	R/W	R/W	R/W	R/W		R/W
									0x0	0x0	0x0	0x0	0x0		0x0

字段	说明
[31] DTEN	**死区使能** 此位使得输出 A 和输出 B 之间插入死区 1：死区开启（根据输出 A 产生死区输出） 0：死区关闭
[30] DLYPRTEN	**延迟保护使能** 此位使得输出 A 和输出 B 执行延迟保护 1：延迟保护开启 0：延迟保护关闭
[29:27] DLYPRT	**延迟保护机制** 此位选择输出 A 和输出 B 延迟保护机制 111：输出 A 和输出 B 在外部事件 x 上均衡空闲 110：输出 A 和输出 B 在外部事件 x 上延迟空闲 101：输出 B 在外部事件 x 上延迟空闲 100：输出 A 在外部事件 x 上延迟空闲 011：输出 A 和输出 B 在外部事件 y 上均衡空闲 010：输出 A 和输出 B 在外部事件 y 上延迟空闲 001：输出 B 在外部事件 y 上延迟空闲 000：输出 A 在外部事件 y 上延迟空闲 Note：对于 PWM0 - PWM3，x = 6，y = 5 对于 PWM4 - PWM7，x = 8，y = 7
[25] BIAR	**均衡空闲自动恢复** 1：均衡空闲自动恢复开启 0：均衡空闲自动恢复关闭 Note：此位仅在 DLYPRT = 011 或 111 时有效
[22] DIDLB	**输出 B 空闲模式死区使能** 此位使得输出 B 在 Burst Mode 下进入空闲状态时插入死区 1：空闲模式死区开启 0：空闲模式死区关闭
[21]	**输出 B 空闲模式**

IDLEMB	此位控制输出 B 空闲模式 1：Burst Mode 开启 0：Burst Mode 关闭
[20] CHPB	**输出 B 斩波使能** 此位使得输出 B 开启斩波 1：输出斩波开启 0：输出斩波关闭
[19] IDLESB	**输出 B 空闲状态** 此位控制输出 B 在空闲状态的输出电平 1：有效电平 0：无效电平
[18:17] FAULTB	**输出 B 故障状态** 此位控制输出 B 在故障状态的输出电平 11：高阻态 10：无效电平 01：有效电平 00：无动作
[16] POLB	**输出 B 极性** 此位选择输出 B 的极性 1：负极性（输出低有效） 0：正极性（输出高有效）
[6] DIDLA	**输出 A 空闲模式死区使能** 此位使得输出 A 在 Burst Mode 下进入空闲状态时插入死区 1：空闲模式死区开启 0：空闲模式死区关闭
[5] IDLEMA	**输出 A 空闲模式** 此位控制输出 A 空闲模式 1：Burst Mode 开启 0：Burst Mode 关闭
[4] CHPA	**输出 A 斩波使能** 此位使得输出 A 开启斩波 1：输出斩波开启 0：输出斩波关闭
[3] IDLESA	**输出 A 空闲状态** 此位控制输出 A 在空闲状态的输出电平 1：有效电平 0：无效电平
[2:1] FAULTA	**输出 A 故障电平** 此位控制输出 A 在故障状态的输出电平 11：高阻态 10：无效电平 01：有效电平 00：无动作
[0] POLA	**输出 A 极性** 此位选择输出 A 的极性 1：负极性（输出低有效） 0：正极性（输出高有效）

17.5.2.83 HRPWM 定时器 x 故障使能寄存器（HRPWM_FLTxR）

- 名称：**HRPWM PWMx Fault Register**
- 偏移地址：**0x174+N*0x100[N=0-7]**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FLT7EN	FLT6EN	FLT5EN	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLT0EN
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] FLT7EN	**Fault 7 使能** 1：Fault 7 开启，可以停止 PWM 输出 0：Fault 7 关闭
[6] FLT6EN	**Fault 6 使能** 1：Fault 6 开启，可以停止 PWM 输出 0：Fault 6 关闭
[5] FLT5EN	**Fault 5 使能** 1：Fault 5 开启，可以停止 PWM 输出 0：Fault 5 关闭
[4] FLT4EN	**Fault 4 使能** 1：Fault 4 开启，可以停止 PWM 输出 0：Fault 4 关闭
[3] FLT3EN	**Fault 3 使能** 1：Fault 3 开启，可以停止 PWM 输出 0：Fault 3 关闭
[2] FLT2EN	**Fault 2 使能** 1：Fault 2 开启，可以停止 PWM 输出 0：Fault 2 关闭
[1] FLT1EN	**Fault 1 使能** 1：Fault 1 开启，可以停止 PWM 输出 0：Fault 1 关闭
[0] FLT0EN	**Fault 0 使能** 1：Fault 0 开启，可以停止 PWM 输出 0：Fault 0 关闭

17.5.2.84 HRPWM 控制寄存器 0（HRPWM_CR0）

- 名称：HRPWM Control Register0
- 偏移地址：0xF00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
USRC3				USRC2				USRC1				USRC0			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMUDIS							UDIS7	UDIS6	UDIS5	UDIS4	UDIS3	UDIS2	UDIS1	UDIS0	MUDIS
R/W							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31:28] USRC3	**Adc Trigger 3 更新来源** 此位定义了触发 ADC3R 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[27:24] USRC2	**Adc Trigger 2 更新来源** 此位定义了触发 ADC2R 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[23:20] USRC1	**Adc Trigger 1 更新来源** 此位定义了触发 ADC1R 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[19:16] USRC0	**Adc Trigger 0 更新来源** 此位定义了触发 ADC0R 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6

	0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[15] BMUDIS	**Burst Mode 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[8] UDIS7	**PWM7 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[7] UDIS6	**PWM6 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[6] UDIS5	**PWM5 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[5] UDIS4	**PWM4 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[4] UDIS3	**PWM3 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[3] UDIS2	**PWM2 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[2] UDIS1	**PWM1 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[1] UDIS0	**PWM0 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新
[0] MUDIS	**Master PWM 更新禁止** 此位用于暂时禁止更新，软件需要写入并同时加载多个寄存器时需要暂时禁止 1：禁止更新 0：允许更新

17.5.2.85 HRPWM 控制寄存器 1（HRPWM_CR1）

- 名称：HRPWM Control Register1
- 偏移地址：0xF04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TLEN3				TLEN2				TLEN1				TLEN0			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SWP7	SWP6	SWP5	SWP4	SWP3	SWP2	SWP1	SWP0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31:28] TLEN3	**Adc Trigger 3 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[27:24] TLEN2	**Adc Trigger 2 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[23:20] TLEN1	**Adc Trigger 1 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[19:16] TLEN0	**Adc Trigger 0 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[7] SWP7	**PWM7 输出交换** 此位用于交换 PWM7 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET7A->SET7B，SET7B->SET7A，CLR7A->CLR7B，CLR7B->CLR7A 0：SET7A->SET7A，SET7B->SET7B，CLR7A->CLR7A，CLR7B->CLR7B
[6] SWP6	**PWM6 输出交换** 此位用于交换 PWM6 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义

	1：SET6A->SET6B，SET6B->SET6A，CLR6A->CLR6B，CLR6B->CLR6A 0：SET6A->SET6A，SET6B->SET6B，CLR6A->CLR6A，CLR6B->CLR6B
[5] SWP5	**PWM5 输出交换** 此位用于交换 PWM5 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET5A->SET5B，SET5B->SET5A，CLR5A->CLR5B，CLR5B->CLR5A 0：SET5A->SET5A，SET5B->SET5B，CLR5A->CLR5A，CLR5B->CLR5B
[4] SWP4	**PWM4 输出交换** 此位用于交换 PWM4 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET4A->SET4B，SET4B->SET4A，CLR4A->CLR4B，CLR4B->CLR4A 0：SET4A->SET4A，SET4B->SET4B，CLR4A->CLR4A，CLR4B->CLR4B
[3] SWP3	**PWM3 输出交换** 此位用于交换 PWM3 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET3A->SET3B，SET3B->SET3A，CLR3A->CLR3B，CLR3B->CLR3A 0：SET3A->SET3A，SET3B->SET3B，CLR3A->CLR3A，CLR3B->CLR3B
[2] SWP2	**PWM2 输出交换** 此位用于交换 PWM2 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET2A->SET2B，SET2B->SET2A，CLR2A->CLR2B，CLR2B->CLR2A 0：SET2A->SET2A，SET2B->SET2B，CLR2A->CLR2A，CLR2B->CLR2B
[1] SWP1	**PWM1 输出交换** 此位用于交换 PWM1 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET1A->SET1B，SET1B->SET1A，CLR1A->CLR1B，CLR1B->CLR1A 0：SET1A->SET1A，SET1B->SET1B，CLR1A->CLR1A，CLR1B->CLR1B
[0] SWP0	**PWM0 输出交换** 此位用于交换 PWM0 输出，注意当推挽模式开启时（PSHPLL＝1），此位无意义 1：SET0A->SET0B，SET0B->SET0A，CLR0A->CLR0B，CLR0B->CLR0A 0：SET0A->SET0A，SET0B->SET0B，CLR0A->CLR0A，CLR0B->CLR0B

17.5.2.86 HRPWM 控制寄存器 2（HRPWM_CR2）

- **名称：HRPWM Control Register2**
- **偏移地址：0xF08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							RST7	RST6	RST5	RST4	RST3	RST2	RST1	RST0	MRST
							R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SWU7	SWU6	SWU5	SWU4	SWU3	SWU2	SWU1	SWU0	MSWU
							R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC	R/WAC
							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[24] RST7	**PWM7 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[23] RST6	**PWM6 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[22] RST5	**PWM5 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器

[21] RST4	**PWM4 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[20] RST3	**PWM3 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[19] RST2	**PWM2 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[18] RST1	**PWM1 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[17] RST0	**PWM0 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[16] MRST	**Master PWM 软件复位** 此位写 1 强制将计数器复位，写 1 后等待硬件自动清零，此时计数器复位完成 1：软件复位计数器 0：软件不复位计数器
[8] SWU7	**PWM7 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[7] SWU6	**PWM6 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[6] SWU5	**PWM5 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[5] SWU4	**PWM4 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[4] SWU3	**PWM3 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[3] SWU2	**PWM2 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[2] SWU1	**PWM1 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[1] SWU0	**PWM0 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器 0：软件不更新工作寄存器
[0] MSWU	**Master PWM 软件更新** 此位写 1 强制将影子寄存器更新到工作寄存器，写 1 后等待硬件自动清零，此时寄存器更新完成 1：软件更新工作寄存器

0：软件不更新工作寄存器

17.5.2.87 HRPWM 中断状态寄存器（HRPWM_ISR）

- 名称：**HRPWM Interrupt Status Register**
- 偏移地址：**0xF10**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMPER							FLT7	FLT6	FLT5	FLT4	FLT3	FLT2	FLT1	FLT0	SYSFLT
R/W1C							R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
0x0							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[15] BMPER	**Burst Mode Period 中断标志** 1：Burst Mode Period 中断标志产生 0：Burst Mode Period 中断标志未产生 Note：写 1 清除中断标志
[8] FLT7	**Fault 7 中断标志** 1：Fault 7 中断标志产生 0：Fault 7 中断标志未产生 Note：写 1 清除中断标志
[7] FLT6	**Fault 6 中断标志** 1：Fault 6 中断标志产生 0：Fault 6 中断标志未产生 Note：写 1 清除中断标志
[6] FLT5	**Fault 5 中断标志** 1：Fault 5 中断标志产生 0：Fault 5 中断标志未产生 Note：写 1 清除中断标志
[5] FLT4	**Fault 4 中断标志** 1：Fault 4 中断标志产生 0：Fault 4 中断标志未产生 Note：写 1 清除中断标志
[4] FLT3	**Fault 3 中断标志** 1：Fault 3 中断标志产生 0：Fault 3 中断标志未产生 Note：写 1 清除中断标志
[3] FLT2	**Fault 2 中断标志** 1：Fault 2 中断标志产生 0：Fault 2 中断标志未产生 Note：写 1 清除中断标志
[2] FLT1	**Fault 1 中断标志** 1：Fault 1 中断标志产生 0：Fault 1 中断标志未产生 Note：写 1 清除中断标志
[1] FLT0	**Fault 0 中断标志** 1：Fault 0 中断标志产生 0：Fault 0 中断标志未产生 Note：写 1 清除中断标志
[0]	**System Fault 中断标志**

SYSFLT	1：System Fault 中断标志产生 0：System Fault 中断标志未产生 Note：写 1 清除中断标志

17.5.2.88 HRPWM 中断使能寄存器（HRPWM_IER）

- **名称：HRPWM Interrupt Enable Register**
- **偏移地址：0xF14**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMPERIE							FLT7IE	FLT6IE	FLT5IE	FLT4IE	FLT3IE	FLT2IE	FLT1IE	FLT0IE	SYSFLTIE
R/W							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[15] BMPERIE	**Burst Mode Period 中断使能** 1：Burst Mode Period 中断使能 0：Burst Mode Period 中断不使能
[8] FLT7IE	**Fault 7 中断使能** 1：Fault 7 中断使能 0：Fault 7 中断不使能
[7] FLT6IE	**Fault 6 中断使能** 1：Fault 6 中断使能 0：Fault 6 中断不使能
[6] FLT5IE	**Fault 5 中断使能** 1：Fault 5 中断使能 0：Fault 5 中断不使能
[5] FLT4IE	**Fault 4 中断使能** 1：Fault 4 中断使能 0：Fault 4 中断不使能
[4] FLT3IE	**Fault 3 中断使能** 1：Fault 3 中断使能 0：Fault 3 中断不使能
[3] FLT2IE	**Fault 2 中断使能** 1：Fault 2 中断使能 0：Fault 2 中断不使能
[2] FLT1IE	**Fault 1 中断使能** 1：Fault 1 中断使能 0：Fault 1 中断不使能
[1] FLT0IE	**Fault 0 中断使能** 1：Fault 0 中断使能 0：Fault 0 中断不使能
[0] SYSFLTIE	**System Fault 中断使能** 1：System Fault 中断使能 0：System Fault 中断不使能

17.5.2.89 HRPWM 输出使能寄存器（HRPWM_OENR）

- 名称：HRPWM Output Enable Register
- 偏移地址：0xF18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OEN7B	OEN7A	OEN6B	OEN6A	OEN5B	OEN5A	OEN4B	OEN4A	OEN3B	OEN3A	OEN2B	OEN2A	OEN1B	OEN1A	OEN0B	OEN0A
R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[15] OEN7B	**PWM7 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS7B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[14] OEN7A	**PWM7 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS7A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[13] OEN6B	**PWM6 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS6B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[12] OEN6A	**PWM6 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS6A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[11] OEN5B	**PWM5 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS5B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[10] OEN5A	**PWM5 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS5A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[9] OEN4B	**PWM4 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS4B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[8] OEN4A	**PWM4 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS4A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[7]	**PWM3 Out B 启动**

OEN3B	此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS3B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[6] OEN3A	**PWM3 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS3A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[5] OEN2B	**PWM2 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS2B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[4] OEN2A	**PWM2 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS2A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[3] OEN1B	**PWM1 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS1B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[2] OEN1A	**PWM1 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS1A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态
[1] OEN0B	**PWM0 Out B 启动** 此位写 1 启动输出 B，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS0B 读出 1：Out B 启动 0：Out B 停止，可能处在故障状态或空闲状态
[0] OEN0A	**PWM0 Out A 启动** 此位写 1 启动输出 A，输出从空闲状态或故障状态进入启动状态，此位写 0 没有效果读此位返回输出启动状态，输出停止状态可能是故障或空闲状态，由 ODIS0A 读出 1：Out A 启动 0：Out A 停止，可能处在故障状态或空闲状态

17.5.2.90 HRPWM 输出关闭寄存器（HRPWM_ODISR）

- 名称：HRPWM Output Disable Register
- 偏移地址：0xF1C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODIS7B	ODIS7A	ODIS6B	ODIS6A	ODIS5B	ODIS5A	ODIS4B	ODIS4A	ODIS3B	ODIS3A	ODIS2B	ODIS2A	ODIS1B	ODIS1A	ODIS0B	ODIS0A
R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S	R/W1S
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[15] ODIS7B	**PWM7 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[14] ODIS7A	**PWM7 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[13] ODIS6B	**PWM6 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[12] ODIS6A	**PWM6 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[11] ODIS5B	**PWM5 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[10] ODIS5A	**PWM5 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[9] ODIS4B	**PWM4 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[8] ODIS4A	**PWM4 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果；读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[7]	**PWM3 Out B 停止**

ODIS3B	此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[6] ODIS3A	**PWM3 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[5] ODIS2B	**PWM2 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[4] ODIS2A	**PWM2 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[3] ODIS1B	**PWM1 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[2] ODIS1A	**PWM1 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态
[1] ODIS0B	**PWM0 Out B 停止** 此位写 1 停止输出 B，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out B 停止，在故障状态 0：Out B 停止，在空闲状态
[0] ODIS0A	**PWM0 Out A 停止** 此位写 1 停止输出 A，输出从运行状态或故障状态进入空闲状态，此位写 0 没有效果读此位返回输出停止状态，PWM 输出有效时此位无意义 1：Out A 停止，在故障状态 0：Out A 停止，在空闲状态

17.5.2.91 HRPWM 外部事件寄存器 0（HRPWM_EECR0）

- 名称：HRPWM External Event Register0
- 偏移地址：0xF20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		EE4FAST	EE4SNS		EE4POL	EE4SRC		EE3FAST	EE3SNS		EE3POL	EE3SRC		EE2FAST	EE2SNS
		R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W
		0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE2SNS	EE2POL	EE2SRC		EE1FAST	EE1SNS		EE1POL	EE1SRC		EE0FAST	EE0SNS		EE0POL	EE0SRC	
R/W	R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W	
0x0	0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0	

字段	说明
[29] EE4FAST	**Event 4 快速模式** 1：Event 4 异步作用于输出 0：Event 4 同步之后作用于输出
[28:27] EE4SNS	**Event 4 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[26] EE4POL	**Event 4 输入极性** 1： Event 4 输入低有效 0： Event 4 输入高有效
[25:24] EE4SRC	**Event 4 输入来源** 11：ADC1_AWD1 10：CMP6_OUT 01：CMP2_OUT 00：HRPWM_EVT4
[23] EE3FAST	**Event 3 快速模式** 1：Event 3 异步作用于输出 0：Event 3 同步之后作用于输出
[22:21] EE3SNS	**Event 3 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[20] EE3POL	**Event 3 输入极性** 1： Event 3 输入低有效 0： Event 3 输入高有效
[19:18] EE3SRC	**Event 3 输入来源** 11：ADC1_AWD0 10：CMP4_OUT 01：CMP0_OUT 00：HRPWM_EVT3
[17] EE2FAST	**Event 2 快速模式** 1：Event 2 异步作用于输出 0：Event 2 同步之后作用于输出
[16:15] EE2SNS	**Event 2 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效

[14] EE2POL	**Event 2 输入极性** 1： Event 2 输入低有效 0： Event 2 输入高有效
[13:12] EE2SRC	**Event 2 输入来源** 11：CMP7_OUT 10：TMR4_TRGO 01：CMP5_OUT 00：HRPWM_EVT2
[11] EE1FAST	**Event 1 快速模式** 1：Event 1 异步作用于输出 0：Event 1 同步之后作用于输出
[10:9] EE1SNS	**Event 1 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[8] EE1POL	**Event 1 输入极性** 1：Event 1 输入低有效 0：Event 1 输入高有效
[7:6] EE1SRC	**Event 1 输入来源** 11：ADC0_AWD1 10：TMR3_TRGO 01：CMP3_OUT 00：HRPWM_EVT1
[5] EE0FAST	**Event 0 快速模式** 1：Event 0 异步作用于输出 0：Event 0 同步之后作用于输出
[4:3] EE0SNS	**Event 0 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[2] EE0POL	**Event 0 输入极性** 1：Event 0 输入低有效 0：Event 0 输入高有效
[1:0] EE0SRC	**Event 0 输入来源** 11：ADC0_AWD0 10：TMR9_TRGO 01：CMP1_OUT 00：HRPWM_EVT0

17.5.2.92 HRPWM 外部事件寄存器 1（HRPWM_EECR1）

- 名称：HRPWM External Event Register1
- 偏移地址：0xF24
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		EE9FAST	EE9SNS		EE9POL	EE9SRC		EE8FAST	EE8SNS		EE8POL	EE8SRC		EE7FAST	EE7SNS
		R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W
		0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE7SNS	EE7POL	EE7SRC		EE6FAST	EE6SNS		EE6POL	EE6SRC		EE5FAST	EE5SNS		EE5POL	EE5SRC	
R/W	R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W		R/W	R/W	
0x0	0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0		0x0	0x0	

字段	说明
[29] EE9FAST	**Event 9 快速模式** 1：Event 9 异步作用于输出 0：Event 9 同步之后作用于输出
[28:27] EE9SNS	**Event 9 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[26] EE9POL	**Event 9 输入极性** 1：Event 9 输入低有效 0：Event 9 输入高有效
[25:24] EE9SRC	**Event 9 输入来源** 11：PDM0_CMPH 10：TMR7_TRGO 01：CMP6_OUT 00：HRPWM_EVT9
[23] EE8FAST	**Event 8 快速模式** 1：Event 8 异步作用于输出 0：Event 8 同步之后作用于输出
[22:21] EE8SNS	**Event 8 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[20] EE8POL	**Event 8 输入极性** 1：Event 8 输入低有效 0：Event 8 输入高有效
[19:18] EE8SRC	**Event 8 输入来源** 11：CMP3_OUT 10：TMR0_TRGO 01：CMP4_OUT 00：HRPWM_EVT8
[17] EE7FAST	**Event 7 快速模式** 1：Event 7 异步作用于输出 0：Event 7 同步之后作用于输出
[16:15] EE7SNS	**Event 7 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效

[14] EE7POL	**Event 7 输入极性** 1：Event 7 输入低有效 0：Event 7 输入高有效
[13:12] EE7SRC	**Event 7 输入来源** 11：ADC3_AWD0 10：CMP2_OUT 01：CMP5_OUT 00：HRPWM_EVT7
[11] EE6FAST	**Event 6 快速模式** 1：Event 6 异步作用于输出 0：Event 6 同步之后作用于输出
[10:9] EE6SNS	**Event 6 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[8] EE6POL	**Event 6 输入极性** 1：Event 6 输入低有效 0：Event 6 输入高有效
[7:6] EE6SRC	**Event 6 输入来源** 11：ADC2_AWD0 10：TMR8_TRGO 01：CMP3_OUT 00：HRPWM_EVT6
[5] EE5FAST	**Event 5 快速模式** 1：Event 5 异步作用于输出 0：Event 5 同步之后作用于输出
[4:3] EE5SNS	**Event 5 输入有效沿** 11：上升 / 下降沿有效 10：下降沿有效 01：上升沿有效 00：电平有效
[2] EE5POL	**Event 5 输入极性** 1：Event 5 输入低有效 0：Event 5 输入高有效
[1:0] EE5SRC	**Event 5 输入来源** 11：CMP8_OUT 10：CMP0_OUT 01：CMP1_OUT 00：HRPWM_EVT5

17.5.2.93 HRPWM 外部事件寄存器 2（HRPWM_EECR2）

- 名称：HRPWM External Event Register2
- 偏移地址：0xF28
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EEVSD												EE4F			
R/W												R/W			
0x0												0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE3F				EE2F				EE1F				EE0F			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

字段	说明
[31:30] EEVSD	**Event 采样时钟分频比例** 此位决定 Event 采样时钟与 HRPWM 模块时钟的分频比例 11：8 分频（f_sample = f_hrpwm/8） 10：4 分频（f_sample = f_hrpwm/4） 01：2 分频（f_sample = f_hrpwm/2） 00：1 分频（f_sample = f_hrpwm）
[19:16] EE4F	**Event 4 滤波长度** 此位定义用于 Event 4 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[15:12] EE3F	**Event 3 滤波长度** 此位定义用于 Event 3 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[11:8] EE2F	**Event 2 滤波长度** 此位定义用于 Event 2 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[7:4] EE1F	**Event 1 滤波长度** 此位定义用于 Event 1 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[3:0] EE0F	**Event 0 滤波长度** 此位定义用于 Event 0 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出

17.5.2.94 HRPWM 外部事件寄存器 3（HRPWM_EECR3）

- 名称：HRPWM External Event Register3
- 偏移地址：0xF2C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												EE9F			
												R/W			
												0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE8F				EE7F				EE6F				EE5F			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

字段	说明
[19:16] EE9F	**Event 9 滤波长度** 此位定义用于 Event 9 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[15:12] EE8F	**Event 8 滤波长度** 此位定义用于 Event 8 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[11:8] EE7F	**Event 7 滤波长度** 此位定义用于 Event 7 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[7:4] EE6F	**Event 6 滤波长度** 此位定义用于 Event 6 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出
[3:0] EE5F	**Event 5 滤波长度** 此位定义用于 Event 5 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效假设滤波长度为 N，滤波总时长为 N×1/f_sample，滤波长度为 0 时不滤波，Event 信号抓一拍即输出

17.5.2.95 HRPWM ADC 触发事件 0 寄存器（HRPWM_ADC0R）

- 名称：HRPWM ADC Trigger Register 0
- 偏移地址：0xF30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC0 RST5	ADC0 PER5	ADC0 CMPD 5	ADC0 CMPC 5	ADC0 CMPB 5	ADC0 PER4	ADC0 CMPD 4	ADC0 CMPC 4	ADC0 PER3	ADC0 CMPD 3	ADC0 CMPC 3	ADC0 PER2	ADC0 CMPD 2	ADC0 CMPC 2	ADC0 RST1	ADC0 PER1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC0 CMPD 1	ADC0 CMPC 1	ADC0 RST0	ADC0 PER0	ADC0 CMPD 0	ADC0 CMPC 0	ADC0 EEV4	ADC0 EEV3	ADC0 EEV2	ADC0 EEV1	ADC0 EEV0	ADC0 MPER	ADC0 MCMP D	ADC0 MCMP C	ADC0 MCMP B	ADC0 MCMP A
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] ADC0RST5	**PWM5 Reset/RollOver 事件会产生 Adc Trigger 0**
[30] ADC0PER5	**PWM5 Period 事件会产生 Adc Trigger 0**

[29] ADC0CMPD5	**PWM5 Compare D 事件会产生 Adc Trigger 0**
[28] ADC0CMPC5	**PWM5 Compare C 事件会产生 Adc Trigger 0**
[27] ADC0CMPB5	**PWM5 Compare B 事件会产生 Adc Trigger 0**
[26] ADC0PER4	**PWM4 Period 事件会产生 Adc Trigger 0**
[25] ADC0CMPD4	**PWM4 Compare D 事件会产生 Adc Trigger 0**
[24] ADC0CMPC4	**PWM4 Compare C 事件会产生 Adc Trigger 0**
[23] ADC0PER3	**PWM3 Period 事件会产生 Adc Trigger 0**
[22] ADC0CMPD3	**PWM3 Compare D 事件会产生 Adc Trigger 0**
[21] ADC0CMPC3	**PWM3 Compare C 事件会产生 Adc Trigger 0**
[20] ADC0PER2	**PWM2 Period 事件会产生 Adc Trigger 0**
[19] ADC0CMPD2	**PWM2 Compare D 事件会产生 Adc Trigger 0**
[18] ADC0CMPC2	**PWM2 Compare C 事件会产生 Adc Trigger 0**
[17] ADC0RST1	**PWM1 Reset/RollOver 事件会产生 Adc Trigger 0**
[16] ADC0PER1	**PWM1 Period 事件会产生 Adc Trigger 0**
[15] ADC0CMPD1	**PWM1 Compare D 事件会产生 Adc Trigger 0**
[14] ADC0CMPC1	**PWM1 Compare C 事件会产生 Adc Trigger 0**
[13] ADC0RST0	**PWM0 Reset/RollOver 事件会产生 Adc Trigger 0**
[12] ADC0PER0	**PWM0 Period 事件会产生 Adc Trigger 0**
[11] ADC0CMPD0	**PWM0 Compare D 事件会产生 Adc Trigger 0**
[10] ADC0CMPC0	**PWM0 Compare C 事件会产生 Adc Trigger 0**
[9] ADC0EEV4	**External Event 4 事件会产生 Adc Trigger 0**
[8] ADC0EEV3	**External Event 3 事件会产生 Adc Trigger 0**
[7] ADC0EEV2	**External Event 2 事件会产生 Adc Trigger 0**
[6] ADC0EEV1	**External Event 1 事件会产生 Adc Trigger 0**
[5] ADC0EEV0	**External Event 0 事件会产生 Adc Trigger 0**
[4] ADC0MPER	**Master PWM Period 事件会产生 Adc Trigger 0**
[3] ADC0MCMPD	**Master PWM Compare D 事件会产生 Adc Trigger 0**
[2] ADC0MCMPC	**Master PWM Compare C 事件会产生 Adc Trigger 0**
[1] ADC0MCMPB	**Master PWM Compare B 事件会产生 Adc Trigger 0**
[0] ADC0MCMPA	**Master PWM Compare A 事件会产生 Adc Trigger 0**

17.5.2.96 HRPWM ADC 触发事件 0 扩展寄存器（HRPWM_ADC0ER）

- 名称：HRPWM ADC Trigger Extended Register 0
- 偏移地址：0xF34
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ADC0 RST7	ADC0 PER7	ADC0 CMPD 7	ADC0 CMPC 7	ADC0 RST6	ADC0 PER6	ADC0 CMPD 6	ADC0 CMPC 6
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] ADC0RST7	**PWM7 Reset/RollOver 事件会产生 Adc Trigger 0**
[6] ADC0PER7	**PWM7 Period 事件会产生 Adc Trigger 0**
[5] ADC0CMPD7	**PWM7 Compare D 事件会产生 Adc Trigger 0**
[4] ADC0CMPC7	**PWM7 Compare C 事件会产生 Adc Trigger 0**
[3] ADC0RST6	**PWM6 Reset/RollOver 事件会产生 Adc Trigger 0**
[2] ADC0PER6	**PWM6 Period 事件会产生 Adc Trigger 0**
[1] ADC0CMPD6	**PWM6 Compare D 事件会产生 Adc Trigger 0**
[0] ADC0CMPC6	**PWM6 Compare C 事件会产生 Adc Trigger 0**

17.5.2.97 HRPWM ADC 触发事件 1 寄存器（HRPWM_ADC1R）

- 名称：HRPWM ADC Trigger Register 1
- 偏移地址：0xF38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC1 PER5	ADC1 CMPD 5	ADC1 CMPC 5	ADC1 CMPB 5	ADC1 RST4	ADC1 CMPD 4	ADC1 CMPC 4	ADC1 CMPB 4	ADC1 RST3	ADC1 PER3	ADC1 CMPD 3	ADC1 CMPB 3	ADC1 RST2	ADC1 PER2	ADC1 CMPD 2	ADC1 CMPB 2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC1 PER1	ADC1 CMPD 1	ADC1 CMPB 1	ADC1 PER0	ADC1 CMPD 0	ADC1 CMPB 0	ADC1 EEV9	ADC1 EEV8	ADC1 EEV7	ADC1 EEV6	ADC1 EEV5	ADC1 MPER	ADC1 MCMP D	ADC1 MCMP C	ADC1 MCMP B	ADC1 MCMP A
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] ADC1PER5	**PWM5 Period 事件会产生 Adc Trigger 1**
[30] ADC1CMPD5	**PWM5 Compare D 事件会产生 Adc Trigger 1**

[29] ADC1CMPC5	**PWM5 Compare C 事件会产生 Adc Trigger 1**
[28] ADC1CMPB5	**PWM5 Compare B 事件会产生 Adc Trigger 1**
[27] ADC1RST4	**PWM4 Reset/RollOver 事件会产生 Adc Trigger 1**
[26] ADC1CMPD4	**PWM4 Compare D 事件会产生 Adc Trigger 1**
[25] ADC1CMPC4	**PWM4 Compare C 事件会产生 Adc Trigger 1**
[24] ADC1CMPB4	**PWM4 Compare B 事件会产生 Adc Trigger 1**
[23] ADC1RST3	**PWM3 Reset/RollOver 事件会产生 Adc Trigger 1**
[22] ADC1PER3	**PWM3 Period 事件会产生 Adc Trigger 1**
[21] ADC1CMPD3	**PWM3 Compare D 事件会产生 Adc Trigger 1**
[20] ADC1CMPB3	**PWM3 Compare B 事件会产生 Adc Trigger 1**
[19] ADC1RST2	**PWM2 Reset/RollOver 事件会产生 Adc Trigger 1**
[18] ADC1PER2	**PWM2 Period 事件会产生 Adc Trigger 1**
[17] ADC1CMPD2	**PWM2 Compare D 事件会产生 Adc Trigger 1**
[16] ADC1CMPB2	**PWM2 Compare B 事件会产生 Adc Trigger 1**
[15] ADC1PER1	**PWM1 Period 事件会产生 Adc Trigger 1**
[14] ADC1CMPD1	**PWM1 Compare D 事件会产生 Adc Trigger 1**
[13] ADC1CMPB1	**PWM1 Compare B 事件会产生 Adc Trigger 1**
[12] ADC1PER0	**PWM0 Period 事件会产生 Adc Trigger 1**
[11] ADC1CMPD0	**PWM0 Compare D 事件会产生 Adc Trigger 1**
[10] ADC1CMPB0	**PWM0 Compare B 事件会产生 Adc Trigger 1**
[9] ADC1EEV9	**External Event 9 事件会产生 Adc Trigger 1**
[8] ADC1EEV8	**External Event 8 事件会产生 Adc Trigger 1**
[7] ADC1EEV7	**External Event 7 事件会产生 Adc Trigger 1**
[6] ADC1EEV6	**External Event 6 事件会产生 Adc Trigger 1**
[5] ADC1EEV5	**External Event 5 事件会产生 Adc Trigger 1**
[4] ADC1MPER	**Master PWM Period 事件会产生 Adc Trigger 1**
[3] ADC1MCMPD	**Master PWM Compare D 事件会产生 Adc Trigger 1**
[2] ADC1MCMPC	**Master PWM Compare C 事件会产生 Adc Trigger 1**
[1] ADC1MCMPB	**Master PWM Compare B 事件会产生 Adc Trigger 1**
[0] ADC1MCMPA	**Master PWM Compare A 事件会产生 Adc Trigger 1**

17.5.2.98 HRPWM ADC 触发事件 1 扩展寄存器（HRPWM_ADC1ER）

- 名称：HRPWM ADC Trigger Extended Register 1
- 偏移地址：0xF3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ADC1 RST7	ADC1 PER7	ADC1 CMPD 7	ADC1 CMPB 7	ADC1 RST6	ADC1 PER6	ADC1 CMPD 6	ADC1 CMPB 6
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] ADC1RST7	**PWM7 Reset/RollOver 事件会产生 Adc Trigger 1**
[6] ADC1PER7	**PWM7 Period 事件会产生 Adc Trigger 1**
[5] ADC1CMPD7	**PWM7 Compare D 事件会产生 Adc Trigger 1**
[4] ADC1CMPB7	**PWM7 Compare B 事件会产生 Adc Trigger 1**
[3] ADC1RST6	**PWM6 Reset/RollOver 事件会产生 Adc Trigger 1**
[2] ADC1PER6	**PWM6 Period 事件会产生 Adc Trigger 1**
[1] ADC1CMPD6	**PWM6 Compare D 事件会产生 Adc Trigger 1**
[0] ADC1CMPB6	**PWM6 Compare B 事件会产生 Adc Trigger 1**

17.5.2.99 HRPWM ADC 触发事件 2 寄存器（HRPWM_ADC2R）

- 名称：HRPWM ADC Trigger Register 2
- 偏移地址：0xF40
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC2 RST5	ADC2 PER5	ADC2 CMPD 5	ADC2 CMPC 5	ADC2 CMPB 5	ADC2 PER4	ADC2 CMPD 4	ADC2 CMPC 4	ADC2 PER3	ADC2 CMPD 3	ADC2 CMPC 3	ADC2 PER2	ADC2 CMPD 2	ADC2 CMPC 2	ADC2 RST1	ADC2 PER1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2 CMPD 1	ADC2 CMPC 1	ADC2 RST0	ADC2 PER0	ADC2 CMPD 0	ADC2 CMPC 0	ADC2 EEV4	ADC2 EEV3	ADC2 EEV2	ADC2 EEV1	ADC2 EEV0	ADC2 MPER	ADC2 MCMP D	ADC2 MCMP C	ADC2 MCMP B	ADC2 MCMP A
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] ADC2RST5	**PWM5 Reset/RollOver 事件会产生 Adc Trigger 2**
[30] ADC2PER5	**PWM5 Period 事件会产生 Adc Trigger 2**

[29] ADC2CMPD5	**PWM5 Compare D 事件会产生 Adc Trigger 2**
[28] ADC2CMPC5	**PWM5 Compare C 事件会产生 Adc Trigger 2**
[27] ADC2CMPB5	**PWM5 Compare B 事件会产生 Adc Trigger 2**
[26] ADC2PER4	**PWM4 Period 事件会产生 Adc Trigger 2**
[25] ADC2CMPD4	**PWM4 Compare D 事件会产生 Adc Trigger 2**
[24] ADC2CMPC4	**PWM4 Compare C 事件会产生 Adc Trigger 2**
[23] ADC2PER3	**PWM3 Period 事件会产生 Adc Trigger 2**
[22] ADC2CMPD3	**PWM3 Compare D 事件会产生 Adc Trigger 2**
[21] ADC2CMPC3	**PWM3 Compare C 事件会产生 Adc Trigger 2**
[20] ADC2PER2	**PWM2 Period 事件会产生 Adc Trigger 2**
[19] ADC2CMPD2	**PWM2 Compare D 事件会产生 Adc Trigger 2**
[18] ADC2CMPC2	**PWM2 Compare C 事件会产生 Adc Trigger 2**
[17] ADC2RST1	**PWM1 Reset/RollOver 事件会产生 Adc Trigger 2**
[16] ADC2PER1	**PWM1 Period 事件会产生 Adc Trigger 2**
[15] ADC2CMPD1	**PWM1 Compare D 事件会产生 Adc Trigger 2**
[14] ADC2CMPC1	**PWM1 Compare C 事件会产生 Adc Trigger 2**
[13] ADC2RST0	**PWM0 Reset/RollOver 事件会产生 Adc Trigger 2**
[12] ADC2PER0	**PWM0 Period 事件会产生 Adc Trigger 2**
[11] ADC2CMPD0	**PWM0 Compare D 事件会产生 Adc Trigger 2**
[10] ADC2CMPC0	**PWM0 Compare C 事件会产生 Adc Trigger 2**
[9] ADC2EEV4	**External Event 4 事件会产生 Adc Trigger 2**
[8] ADC2EEV3	**External Event 3 事件会产生 Adc Trigger 2**
[7] ADC2EEV2	**External Event 2 事件会产生 Adc Trigger 2**
[6] ADC2EEV1	**External Event 1 事件会产生 Adc Trigger 2**
[5] ADC2EEV0	**External Event 0 事件会产生 Adc Trigger 2**
[4] ADC2MPER	**Master PWM Period 事件会产生 Adc Trigger 2**
[3] ADC2MCMPD	**Master PWM Compare D 事件会产生 Adc Trigger 2**
[2] ADC2MCMPC	**Master PWM Compare C 事件会产生 Adc Trigger 2**
[1] ADC2MCMPB	**Master PWM Compare B 事件会产生 Adc Trigger 2**
[0] ADC2MCMPA	**Master PWM Compare A 事件会产生 Adc Trigger 2**

17.5.2.100 HRPWM ADC 触发事件 2 扩展寄存器（HRPWM_ADC2ER）

- 名称：HRPWM ADC Trigger Extended Register 2
- 偏移地址：0xF44
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ADC2 RST7	ADC2 PER7	ADC2 CMPD 7	ADC2 CMPC 7	ADC2 RST6	ADC2 PER6	ADC2 CMPD 6	ADC2 CMPC 6
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] ADC2RST7	**PWM7 Reset/RollOver 事件会产生 Adc Trigger 2**
[6] ADC2PER7	**PWM7 Period 事件会产生 Adc Trigger 2**
[5] ADC2CMPD7	**PWM7 Compare D 事件会产生 Adc Trigger 2**
[4] ADC2CMPC7	**PWM7 Compare C 事件会产生 Adc Trigger 2**
[3] ADC2RST6	**PWM6 Reset/RollOver 事件会产生 Adc Trigger 2**
[2] ADC2PER6	**PWM6 Period 事件会产生 Adc Trigger 2**
[1] ADC2CMPD6	**PWM6 Compare D 事件会产生 Adc Trigger 2**
[0] ADC2CMPC6	**PWM6 Compare C 事件会产生 Adc Trigger 2**

17.5.2.101 HRPWM ADC 触发事件 3 寄存器（HRPWM_ADC3R）

- 名称：HRPWM ADC Trigger Register 3
- 偏移地址：0xF48
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC3 PER5	ADC3 CMPD 5	ADC3 CMPC 5	ADC3 CMPB 5	ADC3 RST4	ADC3 CMPD 4	ADC3 CMPC 4	ADC3 CMPB 4	ADC3 RST3	ADC3 PER3	ADC3 CMPD 3	ADC3 CMPB 3	ADC3 RST2	ADC3 PER2	ADC3 CMPD 2	ADC3 CMPB 2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 PER1	ADC3 CMPD 1	ADC3 CMPB 1	ADC3 PER0	ADC3 CMPD 0	ADC3 CMPB 0	ADC3 EEV9	ADC3 EEV8	ADC3 EEV7	ADC3 EEV6	ADC3 EEV5	ADC3 MPER	ADC3 MCMP D	ADC3 MCMP C	ADC3 MCMP B	ADC3 MCMP A
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] ADC3PER5	**PWM5 Period 事件会产生 Adc Trigger 3**
[30] ADC3CMPD5	**PWM5 Compare D 事件会产生 Adc Trigger 3**

[29] ADC3CMPC5	**PWM5 Compare C 事件会产生 Adc Trigger 3**
[28] ADC3CMPB5	**PWM5 Compare B 事件会产生 Adc Trigger 3**
[27] ADC3RST4	**PWM4 Reset/RollOver 事件会产生 Adc Trigger 3**
[26] ADC3CMPD4	**PWM4 Compare D 事件会产生 Adc Trigger 3**
[25] ADC3CMPC4	**PWM4 Compare C 事件会产生 Adc Trigger 3**
[24] ADC3CMPB4	**PWM4 Compare B 事件会产生 Adc Trigger 3**
[23] ADC3RST3	**PWM3 Reset/RollOver 事件会产生 Adc Trigger 3**
[22] ADC3PER3	**PWM3 Period 事件会产生 Adc Trigger 3**
[21] ADC3CMPD3	**PWM3 Compare D 事件会产生 Adc Trigger 3**
[20] ADC3CMPB3	**PWM3 Compare B 事件会产生 Adc Trigger 3**
[19] ADC3RST2	**PWM2 Reset/RollOver 事件会产生 Adc Trigger 3**
[18] ADC3PER2	**PWM2 Period 事件会产生 Adc Trigger 3**
[17] ADC3CMPD2	**PWM2 Compare D 事件会产生 Adc Trigger 3**
[16] ADC3CMPB2	**PWM2 Compare B 事件会产生 Adc Trigger 3**
[15] ADC3PER1	**PWM1 Period 事件会产生 Adc Trigger 3**
[14] ADC3CMPD1	**PWM1 Compare D 事件会产生 Adc Trigger 3**
[13] ADC3CMPB1	**PWM1 Compare B 事件会产生 Adc Trigger 3**
[12] ADC3PER0	**PWM0 Period 事件会产生 Adc Trigger 3**
[11] ADC3CMPD0	**PWM0 Compare D 事件会产生 Adc Trigger 3**
[10] ADC3CMPB0	**PWM0 Compare B 事件会产生 Adc Trigger 3**
[9] ADC3EEV9	**External Event 9 事件会产生 Adc Trigger 3**
[8] ADC3EEV8	**External Event 8 事件会产生 Adc Trigger 3**
[7] ADC3EEV7	**External Event 7 事件会产生 Adc Trigger 3**
[6] ADC3EEV6	**External Event 6 事件会产生 Adc Trigger 3**
[5] ADC3EEV5	**External Event 5 事件会产生 Adc Trigger 3**
[4] ADC3MPER	**Master PWM Period 事件会产生 Adc Trigger 3**
[3] ADC3MCMPD	**Master PWM Compare D 事件会产生 Adc Trigger 3**
[2] ADC3MCMPC	**Master PWM Compare C 事件会产生 Adc Trigger 3**
[1] ADC3MCMPB	**Master PWM Compare B 事件会产生 Adc Trigger 3**
[0] ADC3MCMPA	**Master PWM Compare A 事件会产生 Adc Trigger 3**

17.5.2.102 HRPWM ADC 触发事件 3 扩展寄存器（HRPWM_ADC3ER）

- 名称：HRPWM ADC Trigger Extended Register 3
- 偏移地址：0xF4C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ADC3 RST7	ADC3 PER7	ADC3 CMPD 7	ADC3 CMPB 7	ADC3 RST6	ADC3 PER6	ADC3 CMPD 6	ADC3 CMPB 6
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] ADC3RST7	**PWM7 Reset/RollOver 事件会产生 Adc Trigger 3**
[6] ADC3PER7	**PWM7 Period 事件会产生 Adc Trigger 3**
[5] ADC3CMPD7	**PWM7 Compare D 事件会产生 Adc Trigger 3**
[4] ADC3CMPB7	**PWM7 Compare B 事件会产生 Adc Trigger 3**
[3] ADC3RST6	**PWM6 Reset/RollOver 事件会产生 Adc Trigger 3**
[2] ADC3PER6	**PWM6 Period 事件会产生 Adc Trigger 3**
[1] ADC3CMPD6	**PWM6 Compare D 事件会产生 Adc Trigger 3**
[0] ADC3CMPB6	**PWM6 Compare B 事件会产生 Adc Trigger 3**

17.5.2.103 HRPWM ADC 触发事件 4 寄存器（HRPWM_ADC4R）

- 名称：HRPWM ADC Trigger Register 4
- 偏移地址：0xF50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										ADC6TRG					
										R/W					
										0x0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		ADC5TRG								ADC4TRG					
		R/W								R/W					
		0x0								0x0					

字段	说明
[21:16] ADC6TRG	**Adc Trigger 6 来源** Others：Reserved 39：PWM7 Reset/RollOver 事件 38：PWM7 Period 事件 37：PWM7 Compare D 事件 36：PWM7 Compare C 事件

	35：PWM6 Reset/RollOver 事件 34：PWM6 Period 事件 33：PWM6 Compare D 事件 32：PWM6 Compare C 事件 31：PWM5 Reset/RollOver 事件 30：PWM5 Period 事件 29：PWM5 Compare D 事件 28：PWM5 Compare C 事件 27：PWM5 Compare B 事件 26：PWM4 Period 事件 25：PWM4 Compare D 事件 24：PWM4 Compare C 事件 23：PWM3 Period 事件 22：PWM3 Compare D 事件 21：PWM3 Compare C 事件 20：PWM2 Period 事件 19：PWM2 Compare D 事件 18：PWM2 Compare C 事件 17：PWM1 Reset/RollOver 事件 16：PWM1 Period 事件 15：PWM1 Compare D 事件 14：PWM1 Compare C 事件 13：PWM0 Reset/RollOver 事件 12：PWM0 Period 事件 11：PWM0 Compare D 事件 10：PWM0 Compare C 事件 9：External Event 4 事件 8：External Event 3 事件 7：External Event 2 事件 6：External Event 1 事件 5：External Event 0 事件 4：Master PWM Period 事件 3：Master PWM Compare D 事件 2：Master PWM Compare C 事件 1：Master PWM Compare B 事件 0：Master PWM Compare A 事件
[13:8] ADC5TRG	**Adc Trigger 5 来源** Others：Reserved 39：PWM7 Reset/RollOver 事件 38：PWM7 Period 事件 37：PWM7 Compare D 事件 36：PWM7 Compare B 事件 35：PWM6 Reset/RollOver 事件 34：PWM6 Period 事件 33：PWM6 Compare D 事件 32：PWM6 Compare B 事件 31：PWM5 Period 事件 30：PWM5 Compare D 事件 29：PWM5 Compare C 事件 28：PWM5 Compare B 事件 27：PWM4 Reset/RollOver 事件 26：PWM4 Compare D 事件 25：PWM4 Compare C 事件 24：PWM4 Compare B 事件 23：PWM3 Reset/RollOver 事件 22：PWM3 Period 事件 21：PWM3 Compare D 事件 20：PWM3 Compare B 事件 19：PWM2 Reset/RollOver 事件

	18：PWM2 Period 事件 17：PWM2 Compare D 事件 16：PWM2 Compare B 事件 15：PWM1 Period 事件 14：PWM1 Compare D 事件 13：PWM1 Compare B 事件 12：PWM0 Period 事件 11：PWM0 Compare D 事件 10：PWM0 Compare B 事件 9：External Event 9 事件 8：External Event 8 事件 7：External Event 7 事件 6：External Event 6 事件 5：External Event 5 事件 4：Master PWM Period 事件 3：Master PWM Compare D 事件 2：Master PWM Compare C 事件 1：Master PWM Compare B 事件 0：Master PWM Compare A 事件
[5:0] ADC4TRG	**Adc Trigger 4 来源** Others：Reserved 39：PWM7 Reset/RollOver 事件 38：PWM7 Period 事件 37：PWM7 Compare D 事件 36：PWM7 Compare C 事件 35：PWM6 Reset/RollOver 事件 34：PWM6 Period 事件 33：PWM6 Compare D 事件 32：PWM6 Compare C 事件 31：PWM5 Reset/RollOver 事件 30：PWM5 Period 事件 29：PWM5 Compare D 事件 28：PWM5 Compare C 事件 27：PWM5 Compare B 事件 26：PWM4 Period 事件 25：PWM4 Compare D 事件 24：PWM4 Compare C 事件 23：PWM3 Period 事件 22：PWM3 Compare D 事件 21：PWM3 Compare C 事件 20：PWM2 Period 事件 19：PWM2 Compare D 事件 18：PWM2 Compare C 事件 17：PWM1 Reset/RollOver 事件 16：PWM1 Period 事件 15：PWM1 Compare D 事件 14：PWM1 Compare C 事件 13：PWM0 Reset/RollOver 事件 12：PWM0 Period 事件 11：PWM0 Compare D 事件 10：PWM0 Compare C 事件 9：External Event 4 事件 8：External Event 3 事件 7：External Event 2 事件 6：External Event 1 事件 5：External Event 0 事件 4：Master PWM Period 事件 3：Master PWM Compare D 事件 2：Master PWM Compare C 事件

1：Master PWM Compare B 事件
0：Master PWM Compare A 事件

17.5.2.104 HRPWM ADC 触发事件 5 寄存器（HRPWM_ADC5R）

- **名称：HRPWM ADC Trigger Register 5**
- **偏移地址：0xF54**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										ADC9TRG					
										R/W					
										0x0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		ADC8TRG								ADC7TRG					
		R/W								R/W					
		0x0								0x0					

字段	说明
[21:16] ADC9TRG	**Adc Trigger 9 来源** Others：Reserved 39：PWM7 Reset/RollOver 事件 38：PWM7 Period 事件 37：PWM7 Compare D 事件 36：PWM7 Compare B 事件 35：PWM6 Reset/RollOver 事件 34：PWM6 Period 事件 33：PWM6 Compare D 事件 32：PWM6 Compare B 事件 31：PWM5 Period 事件 30：PWM5 Compare D 事件 29：PWM5 Compare C 事件 28：PWM5 Compare B 事件 27：PWM4 Reset/RollOver 事件 26：PWM4 Compare D 事件 25：PWM4 Compare C 事件 24：PWM4 Compare B 事件 23：PWM3 Reset/RollOver 事件 22：PWM3 Period 事件 21：PWM3 Compare D 事件 20：PWM3 Compare B 事件 19：PWM2 Reset/RollOver 事件 18：PWM2 Period 事件 17：PWM2 Compare D 事件 16：PWM2 Compare B 事件 15：PWM1 Period 事件 14：PWM1 Compare D 事件 13：PWM1 Compare B 事件 12：PWM0 Period 事件 11：PWM0 Compare D 事件 10：PWM0 Compare B 事件 9：External Event 9 事件 8：External Event 8 事件 7：External Event 7 事件 6：External Event 6 事件 5：External Event 5 事件

	4：Master PWM Period 事件 3：Master PWM Compare D 事件 2：Master PWM Compare C 事件 1：Master PWM Compare B 事件 0：Master PWM Compare A 事件
[13:8] ADC8TRG	**Adc Trigger 8 来源** Others：Reserved 39：PWM7 Reset/RollOver 事件 38：PWM7 Period 事件 37：PWM7 Compare D 事件 36：PWM7 Compare C 事件 35：PWM6 Reset/RollOver 事件 34：PWM6 Period 事件 33：PWM6 Compare D 事件 32：PWM6 Compare C 事件 31：PWM5 Reset/RollOver 事件 30：PWM5 Period 事件 29：PWM5 Compare D 事件 28：PWM5 Compare C 事件 27：PWM5 Compare B 事件 26：PWM4 Period 事件 25：PWM4 Compare D 事件 24：PWM4 Compare C 事件 23：PWM3 Period 事件 22：PWM3 Compare D 事件 21：PWM3 Compare C 事件 20：PWM2 Period 事件 19：PWM2 Compare D 事件 18：PWM2 Compare C 事件 17：PWM1 Reset/RollOver 事件 16：PWM1 Period 事件 15：PWM1 Compare D 事件 14：PWM1 Compare C 事件 13：PWM0 Reset/RollOver 事件 12：PWM0 Period 事件 11：PWM0 Compare D 事件 10：PWM0 Compare C 事件 9：External Event 4 事件 8：External Event 3 事件 7：External Event 2 事件 6：External Event 1 事件 5：External Event 0 事件 4：Master PWM Period 事件 3：Master PWM Compare D 事件 2：Master PWM Compare C 事件 1：Master PWM Compare B 事件 0：Master PWM Compare A 事件
[5:0] ADC7TRG	**Adc Trigger 7 来源** Others：Reserved 39：PWM7 Reset/RollOver 事件 38：PWM7 Period 事件 37：PWM7 Compare D 事件 36：PWM7 Compare B 事件 35：PWM6 Reset/RollOver 事件 34：PWM6 Period 事件 33：PWM6 Compare D 事件 32：PWM6 Compare B 事件 31：PWM5 Period 事件 30：PWM5 Compare D 事件

29：PWM5 Compare C 事件
28：PWM5 Compare B 事件
27：PWM4 Reset/RollOver 事件
26：PWM4 Compare D 事件
25：PWM4 Compare C 事件
24：PWM4 Compare B 事件
23：PWM3 Reset/RollOver 事件
22：PWM3 Period 事件
21：PWM3 Compare D 事件
20：PWM3 Compare B 事件
19：PWM2 Reset/RollOver 事件
18：PWM2 Period 事件
17：PWM2 Compare D 事件
16：PWM2 Compare B 事件
15：PWM1 Period 事件
14：PWM1 Compare D 事件
13：PWM1 Compare B 事件
12：PWM0 Period 事件
11：PWM0 Compare D 事件
10：PWM0 Compare B 事件
9：External Event 9 事件
8：External Event 8 事件
7：External Event 7 事件
6：External Event 6 事件
5：External Event 5 事件
4：Master PWM Period 事件
3：Master PWM Compare D 事件
2：Master PWM Compare C 事件
1：Master PWM Compare B 事件
0：Master PWM Compare A 事件

17.5.2.105 HRPWM ADC 更新寄存器（HRPWM_ADCUR）

- **名称：HRPWM ADC Update Register**
- **偏移地址：0xF58**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								USRC9				USRC8			
								R/W				R/W			
								0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USRC7				USRC6				USRC5				USRC4			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

字段	说明
[23:20] USRC9	**Adc Trigger 9 更新来源** 此位定义了触发 ADC9TRG 更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2

	0010：PWM1 0001：PWM0 0000：Master PWM
[19:16] USRC8	**Adc Trigger 8 更新来源** 此位定义了触发 ADC8TRG 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[15:12] USRC7	**Adc Trigger 7 更新来源** 此位定义了触发 ADC7TRG 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[11:8] USRC6	**Adc Trigger 6 更新来源** 此位定义了触发 ADC6TRG 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[7:4] USRC5	**Adc Trigger 5 更新来源** 此位定义了触发 ADC5TRG 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3 0011：PWM2 0010：PWM1 0001：PWM0 0000：Master PWM
[3:0] USRC4	**Adc Trigger 4 更新来源** 此位定义了触发 ADC4TRG 寄存器更新的来源，定义了来源未定义更具体的触发条件 Others：Reserved 1000：PWM7 0111：PWM6 0110：PWM5 0101：PWM4 0100：PWM3

0011：PWM2
0010：PWM1
0001：PWM0
0000：Master PWM

17.5.2.106 HRPWM ADC 长度寄存器（HRPWM_ADCLR）

- **名称：HRPWM ADC Length Register**
- **偏移地址：0xF5C**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								TLEN9				TLEN8			
								R/W				R/W			
								0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TLEN7				TLEN6				TLEN5				TLEN4			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

字段	说明
[23:20] TLEN9	**Adc Trigger 9 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[19:16] TLEN8	**Adc Trigger 8 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[15:12] TLEN7	**Adc Trigger 7 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[11:8] TLEN6	**Adc Trigger 6 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期

[7:4] TLEN5	**Adc Trigger 5 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期
[3:0] TLEN4	**Adc Trigger 4 事件长度** 15：Adc Trigger 为 1 个 hrpwm_clk 周期 14：Adc Trigger 为 2 个 hrpwm_clk 周期 …… 2：Adc Trigger 为 14 个 hrpwm_clk 周期 1：Adc Trigger 为 15 个 hrpwm_clk 周期 0：Adc Trigger 为 16 个 hrpwm_clk 周期 Note：Adc_Trigger 长度不小于 2 个 adc_clk 周期，小于 2 个 adc_clk 周期无法正常触发 Adc 采样 假设 adc_clk 为 60M，hrpwm_clk 为 180M，则 Adc_Trigger 长度不小于 6 个 hrpwm_clk 周期

17.5.2.107 HRPWM ADC 触发事件后分频寄存器 0（HRPWM_ADPSR0）

- **名称：HRPWM ADC Trigger Post Scaler Register0**
- **偏移地址：0xF60**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			PSC4						PSC3						PSC2
			R/W						R/W						R/W
			0x0						0x0						0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC2					PSC1						PSC0				
R/W					R/W						R/W				
0x0					0x0						0x0				

字段	说明
[28:24] PSC4	**Adc Trigger 4 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 4 产生 1 个 Adc Trigger 4
[22:18] PSC3	**Adc Trigger 3 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 3 产生 1 个 Adc Trigger 3
[16:12] PSC2	**Adc Trigger 2 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 2 产生 1 个 Adc Trigger 2
[10:6] PSC1	**Adc Trigger 1 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 1 产生 1 个 Adc Trigger 1
[4:0] PSC0	**Adc Trigger 0 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 0 产生 1 个 Adc Trigger 0

17.5.2.108 HRPWM ADC 触发事件后分频寄存器 1（HRPWM_ADPSR1）

- 名称：HRPWM ADC Trigger Post Scaler Register1
- 偏移地址：0xF64
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			PSC9						PSC8						PSC7
			R/W						R/W						R/W
			0x0						0x0						0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC7					PSC6						PSC5				
R/W					R/W						R/W				
0x0					0x0						0x0				

字段	说明
[28:24] PSC9	**Adc Trigger 9 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 9 产生 1 个 Adc Trigger 9
[22:18] PSC8	**Adc Trigger 8 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 8 产生 1 个 Adc Trigger 8
[16:12] PSC7	**Adc Trigger 7 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 7 产生 1 个 Adc Trigger 7
[10:6] PSC6	**Adc Trigger 6 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 6 产生 1 个 Adc Trigger 6
[4:0] PSC5	**Adc Trigger 5 降采样比例** 此位配置 Adc Trigger 降采样功能，(PSC+1)个 Adc Trigger 5 产生 1 个 Adc Trigger 5

17.5.2.109 HRPWM DLL 控制寄存器（HRPWM_DLLCR）

- 名称：HRPWM DLL Control Register
- 偏移地址：0xF68
- 默认值：0x00000006
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												DLLSTART	DLLGCP		DLLEN
												R/W	R/W		R/W
												0x0	0x3		0x0

字段	说明
[3] DLLSTART	**DLL 启动位** 1：DLL 启动 0：DLL 停止
[2:1] DLLGCP	**DLL 电流选择位** 11：8uA 10：6uA 01：6uA 00：4uA
[0] DLLEN	**DLL 使能位** 1：DLL 开启 0：DLL 关闭

17.5.2.110 HRPWM 故障输入寄存器 0（HRPWM_FLTINR0）

- 名称：HRPWM Fault Input Register0
- 偏移地址：0xF70
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT7SRC		FLT7P	FLT7E	FLT6SRC		FLT6P	FLT6E	FLT5SRC		FLT5P	FLT5E	FLT4SRC		FLT4P	FLT4E
R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W
0x0		0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT3SRC		FLT3P	FLT3E	FLT2SRC		FLT2P	FLT2E	FLT1SRC		FLT1P	FLT1E	FLT0SRC		FLT0P	FLT0E
R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W
0x0		0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0

字段	说明
[31:30] FLT7SRC	**Fault 7 输入来源** 3：Reserved 2：HRPWM Event 7 1：CMP8_OUT 0：HRPWM_FLT7
[29] FLT7P	**Fault 7 输入极性** 1：Fault 7 输入低有效 0：Fault 7 输入高有效
[28] FLT7E	**Fault 7 输入使能** 1：Fault 7 开启 0：Fault 7 关闭
[27:26] FLT6SRC	**Fault 6 输入来源** 3：Reserved 2：HRPWM Event 6 1：CMP7_OUT 0：HRPWM_FLT6
[25] FLT6P	**Fault 6 输入极性** 1：Fault 6 输入低有效 0：Fault 6 输入高有效
[24] FLT6E	**Fault 6 输入使能** 1：Fault 6 开启 0：Fault 6 关闭
[23:22] FLT5SRC	**Fault 5 输入来源** 3：Reserved 2：HRPWM Event 5 1：CMP4_OUT 0：HRPWM_FLT5
[21] FLT5P	**Fault 5 输入极性** 1：Fault 5 输入低有效 0：Fault 5 输入高有效
[20] FLT5E	**Fault 5 输入使能** 1：Fault 5 开启 0：Fault 5 关闭
[19:18] FLT4SRC	**Fault 4 输入来源** 3：Reserved 2：HRPWM Event 4 1：CMP2_OUT 0：HRPWM_FLT4
[17] FLT4P	**Fault 4 输入极性** 1：Fault 4 输入低有效 0：Fault 4 输入高有效

[16] FLT4E	**Fault 4 输入使能** 1：Fault 4 开启 0：Fault 4 关闭
[15:14] FLT3SRC	**Fault 3 输入来源** 3：Reserved 2：HRPWM Event 3 1：CMP0_OUT 0：HRPWM_FLT3
[13] FLT3P	**Fault 3 输入极性** 1：Fault 3 输入低有效 0：Fault 3 输入高有效
[12] FLT3E	**Fault 3 输入使能** 1：Fault 3 开启 0：Fault 3 关闭
[11:10] FLT2SRC	**Fault 2 输入来源** 3：Reserved 2：HRPWM Event 2 1：CMP5_OUT 0：HRPWM_FLT2
[9] FLT2P	**Fault 2 输入极性** 1：Fault 2 输入低有效 0：Fault 2 输入高有效
[8] FLT2E	**Fault 2 输入使能** 1：Fault 2 开启 0：Fault 2 关闭
[7:6] FLT1SRC	**Fault 1 输入来源** 3：Reserved 2：HRPWM Event 1 1：CMP3_OUT 0：HRPWM_FLT1
[5] FLT1P	**Fault 1 输入极性** 1：Fault 1 输入低有效 0：Fault 1 输入高有效
[4] FLT1E	**Fault 1 输入使能** 1：Fault 1 开启 0：Fault 1 关闭
[3:2] FLT0SRC	**Fault 0 输入来源** 3：Reserved 2：HRPWM Event 0 1：CMP1_OUT 0：HRPWM_FLT0
[1] FLT0P	**Fault 0 输入极性** 1：Fault 0 输入低有效 0：Fault 0 输入高有效
[0] FLT0E	**Fault 0 输入使能** 1：Fault 0 开启 0：Fault 0 关闭

17.5.2.111 HRPWM 故障输入寄存器 1（HRPWM_FLTINR1）

- 名称：HRPWM Fault Input Register1
- 偏移地址：0xF78
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTSD								FLT5F				FLT4F			
R/W								R/W				R/W			
0x0								0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT3F				FLT2F				FLT1F				FLT0F			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

字段	说明
[31:30] FLTSD	**Fault 采样时钟分频比例** 此位决定 Fault 采样时钟与 HRPWM 模块时钟的分频比例 11：8 分频（f_sample = f_hrpwm/8） 10：4 分频（f_sample = f_hrpwm/4） 01：2 分频（f_sample = f_hrpwm/2） 00：1 分频（f_sample = f_hrpwm）
[23:20] FLT5F	**Fault 5 滤波长度** 此位定义用于 Fault 5 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出
[19:16] FLT4F	**Fault 4 滤波长度** 此位定义用于 Fault 4 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出
[15:12] FLT3F	**Fault 3 滤波长度** 此位定义用于 Fault 3 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出
[11:8] FLT2F	**Fault 2 滤波长度** 此位定义用于 Fault 2 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出
[7:4] FLT1F	**Fault 1 滤波长度** 此位定义用于 Fault 1 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出
[3:0] FLT0F	**Fault 0 滤波长度** 此位定义用于 Fault 0 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出

17.5.2.112 HRPWM 故障输入扩展寄存器（HRPWM_FLTINER）

- 名称：HRPWM Fault Input Extend Register
- 偏移地址：0xF7C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FLT7F				FLT6F			
								R/W				R/W			
								0x0				0x0			

字段	说明
[7:4] FLT7F	**Fault 7 滤波长度** 此位定义用于 Fault 7 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出
[3:0] FLT6F	**Fault 6 滤波长度** 此位定义用于 Fault 6 的数字滤波长度，数字滤波由事件计数实现，连续 N 个事件有效则输出有效 假设滤波长度为 N，滤波总时长为 N＊1/f_sample，滤波长度为 0 时不滤波，Fault 信号抓一拍即输出

17.5.2.113 HRPWM 故障输入寄存器 2（HRPWM_FLTINR2）

- 名称：HRPWM Fault Input Register2
- 偏移地址：0xF80
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT7RSTM	FLT7CRES	FLT7BLKS	FLT7BLKE	FLT6RSTM	FLT6CRES	FLT6BLKS	FLT6BLKE	FLT5RSTM	FLT5CRES	FLT5BLKS	FLT5BLKE	FLT4RSTM	FLT4CRES	FLT4BLKS	FLT4BLKE
R/W	R/WAC	R/W	R/W	R/W	R/WAC	R/W	R/W	R/W	R/WAC	R/W	R/W	R/W	R/WAC	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT3RSTM	FLT3CRES	FLT3BLKS	FLT3BLKE	FLT2RSTM	FLT2CRES	FLT2BLKS	FLT2BLKE	FLT1RSTM	FLT1CRES	FLT1BLKS	FLT1BLKE	FLT0RSTM	FLT0CRES	FLT0BLKS	FLT0BLKE
R/W	R/WAC	R/W	R/W	R/W	R/WAC	R/W	R/W	R/W	R/WAC	R/W	R/W	R/W	R/WAC	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] FLT7RSTM	**Fault 7 复位模式** 1：Fault 7 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 7 计数在 Reset / Roll-Over 事件处复位
[30] FLT7CRES	**Fault 7 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 7 计数复位 0：Fault 7 计数不复位
[29] FLT7BLKS	**Fault 7 消隐源（窗口大小由 PWM7 决定）** 1：Fault 7 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 7 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[28] FLT7BLKE	**Fault 7 消隐使能** 1：Fault 7 消隐开启 0：Fault 7 消隐关闭

[27] FLT6RSTM	**Fault 6 复位模式** 1：Fault 6 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 6 计数在 Reset / Roll-Over 事件处复位
[26] FLT6CRES	**Fault 6 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 6 计数复位 0：Fault 6 计数不复位
[25] FLT6BLKS	**Fault 6 消隐源（窗口大小由 PWM6 决定）** 1：Fault 6 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 6 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[24] FLT6BLKE	**Fault 6 消隐使能** 1：Fault 6 消隐开启 0：Fault 6 消隐关闭
[23] FLT5RSTM	**Fault 5 复位模式** 1：Fault 5 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 5 计数在 Reset / Roll-Over 事件处复位
[22] FLT5CRES	**Fault 5 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 5 计数复位 0：Fault 5 计数不复位
[21] FLT5BLKS	**Fault 5 消隐源（窗口大小由 PWM5 决定）** 1：Fault 5 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 5 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[20] FLT5BLKE	**Fault 5 消隐使能** 1：Fault 5 消隐开启 0：Fault 5 消隐关闭
[19] FLT4RSTM	**Fault 4 复位模式** 1：Fault 4 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 4 计数在 Reset / Roll-Over 事件处复位
[18] FLT4CRES	**Fault 4 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 4 计数复位 0：Fault 4 计数不复位
[17] FLT4BLKS	**Fault 4 消隐源（窗口大小由 PWM4 决定）** 1：Fault 4 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 4 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[16] FLT4BLKE	**Fault 4 消隐使能** 1：Fault 4 消隐开启 0：Fault 4 消隐关闭
[15] FLT3RSTM	**Fault 3 复位模式** 1：Fault 3 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 3 计数在 Reset / Roll-Over 事件处复位
[14] FLT3CRES	**Fault 3 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 3 计数复位 0：Fault 3 计数不复位
[13] FLT3BLKS	**Fault 3 消隐源（窗口大小由 PWM3 决定）** 1：Fault 3 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 3 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[12] FLT3BLKE	**Fault 3 消隐使能** 1：Fault 3 消隐开启 0：Fault 3 消隐关闭
[11] FLT2RSTM	**Fault 2 复位模式** 1：Fault 2 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 2 计数在 Reset / Roll-Over 事件处复位
[10] FLT2CRES	**Fault 2 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 2 计数复位

	0：Fault 2 计数不复位
[9] FLT2BLKS	**Fault 2 消隐源（窗口大小由 PWM2 决定）** 1：Fault 2 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 2 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[8] FLT2BLKE	**Fault 2 消隐使能** 1：Fault 2 消隐开启 0：Fault 2 消隐关闭
[7] FLT1RSTM	**Fault 1 复位模式** 1：Fault 1 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 1 计数在 Reset / Roll-Over 事件处复位
[6] FLT1CRES	**Fault 1 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 1 计数复位 0：Fault 1 计数不复位
[5] FLT1BLKS	**Fault 1 消隐源（窗口大小由 PWM1 决定）** 1：Fault 1 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 1 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[4] FLT1BLKE	**Fault 1 消隐使能** 1：Fault 1 消隐开启 0：Fault 1 消隐关闭
[3] FLT0RSTM	**Fault 0 复位模式** 1：Fault 0 计数在 Reset / Roll-Over 事件处复位（仅当上个周期无事件发生时） 0：Fault 0 计数在 Reset / Roll-Over 事件处复位
[2] FLT0CRES	**Fault 0 计数复位** 此位将故障计数复位，软件写 1 后等复位完成后硬件自动清 0 1：Fault 0 计数复位 0：Fault 0 计数不复位
[1] FLT0BLKS	**Fault 0 消隐源（窗口大小由 PWM0 决定）** 1：Fault 0 移动窗口，窗口开始于 CompareD 事件，结束于 CompareC 事件 0：Fault 0 固定窗口，窗口开始于 Reset/Roll-Over 事件，结束于 CompareC 事件
[0] FLT0BLKE	**Fault 0 消隐使能** 1：Fault 0 消隐开启 0：Fault 0 消隐关闭

17.5.2.114 HRPWM 故障输入寄存器 3（HRPWM_FLTINR3）

- **名称：HRPWM Fault Input Register3**
- **偏移地址：0xF88**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT7CNT				FLT6CNT				FLT5CNT				FLT4CNT			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT3CNT				FLT2CNT				FLT1CNT				FLT0CNT			
R/W				R/W				R/W				R/W			
0x0				0x0				0x0				0x0			

字段	说明
[31:28] FLT7CNT	**Fault 7 计数阈值** FLT7CNT 为 0：故障计数不使能 FLT7CNT 不为 0：当故障计数总数等于 FLT7CNT+1 时认为故障有效
[27:24] FLT6CNT	**Fault 6 计数阈值** FLT6CNT 为 0：故障计数不使能 FLT6CNT 不为 0：当故障计数总数等于 FLT6CNT+1 时认为故障有效

[23:20] FLT5CNT	**Fault 5 计数阈值** FLT5CNT 为 0：故障计数不使能 FLT5CNT 不为 0：当故障计数总数等于 FLT5CNT+1 时认为故障有效
[19:16] FLT4CNT	**Fault 4 计数阈值** FLT4CNT 为 0：故障计数不使能 FLT4CNT 不为 0：当故障计数总数等于 FLT4CNT+1 时认为故障有效
[15:12] FLT3CNT	**Fault 3 计数阈值** FLT3CNT 为 0：故障计数不使能 FLT3CNT 不为 0：当故障计数总数等于 FLT3CNT+1 时认为故障有效
[11:8] FLT2CNT	**Fault 2 计数阈值** FLT2CNT 为 0：故障计数不使能 FLT2CNT 不为 0：当故障计数总数等于 FLT2CNT+1 时认为故障有效
[7:4] FLT1CNT	**Fault 1 计数阈值** FLT1CNT 为 0：故障计数不使能 FLT1CNT 不为 0：当故障计数总数等于 FLT1CNT+1 时认为故障有效
[3:0] FLT0CNT	**Fault 0 计数阈值** FLT0CNT 为 0：故障计数不使能 FLT0CNT 不为 0：当故障计数总数等于 FLT0CNT+1 时认为故障有效

17.5.2.115 HRPWM Burst Mode 控制寄存器（HRPWM_BMCR）

- **名称：HRPWM Burst Mode Control Register**
- **偏移地址：0xF90**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMSTAT	BMEXIT						BMDIS7	BMDIS6	BMDIS5	BMDIS4	BMDIS3	BMDIS2	BMDIS1	BMDIS0	MBMDIS
R	R/WAC						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0						0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					BMPREN	BMPRSC				BMCLK				BMOM	BME
					R/W	R/W				R/W				R/W	R/W
					0x0	0x0				0x0				0x0	0x0

字段	说明
[31] BMSTAT	**Burst Mode 状态** 此位给出了当前 Burst Mode 状态 1：Burst Mode 进行中 0：Burst Mode 无效
[30] BMEXIT	**Burst Mode 强制退出** 此位写 1 强制从 Burst Mode 退出，退出完成后硬件自动清零，写 0 没有效果 1：强制退出 Burst Mode 0：不强制退出 Burst Mode
[24] BMDIS7	**Burst Mode 下 PWM7 计数停止** 此位定义了 PWM7 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[23] BMDIS6	**Burst Mode 下 PWM6 计数停止** 此位定义了 PWM6 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[22] BMDIS5	**Burst Mode 下 PWM5 计数停止** 此位定义了 PWM5 在 Burst Mode 下计数是否停止 1：计数停止

	0：计数不停止
[21] BMDIS4	**Burst Mode 下 PWM4 计数停止** 此位定义了 PWM4 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[20] BMDIS3	**Burst Mode 下 PWM3 计数停止** 此位定义了 PWM3 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[19] BMDIS2	**Burst Mode 下 PWM2 计数停止** 此位定义了 PWM2 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[18] BMDIS1	**Burst Mode 下 PWM1 计数停止** 此位定义了 PWM1 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[17] BMDIS0	**Burst Mode 下 PWM0 计数停止** 此位定义了 PWM1 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[16] MBMDIS	**Burst Mode 下 Master PWM 计数停止** 此位定义了 Master PWM 在 Burst Mode 下计数是否停止 1：计数停止 0：计数不停止
[10] BMPREN	**Burst Mode 预加载使能** 此位定义了是否开启预加载功能，定义了寄存器写访问直接作用于工作寄存器还是作用于影子寄存器 1：预加载开启，写访问作用于影子寄存器 0：预加载关闭，写访问直接作用于工作寄存器
[9:6] BMPRSC	**Burst Mode 预分频** 此位定义了 Burst Mode 使用的时钟预分频 15：32768 倍预分频 14：16384 倍预分频 13：8192 倍预分频 12：4096 倍预分频 11：2048 倍预分频 10：1024 倍预分频 9：512 倍预分频 8：256 倍预分频 7：128 倍预分频 6：64 倍预分频 5：32 倍预分频 4：16 倍预分频 3：8 倍预分频 2：4 倍预分频 1：2 倍预分频 0：无预分频
[5:2] BMCLK	**Burst Mode 时钟来源** 此位定义了 Burst Mode 使用的时钟来源 Others：Reserved 15：根据 BMPRSC 预分频的 fHRPWM 时钟 14：hrpwm_bm_in2 事件 13：hrpwm_bm_in1 事件 12：hrpwm_bm_in0 事件 8：PWM7 Reset/RollOver 事件 7：PWM6 Reset/RollOver 事件 6：PWM5 Reset/RollOver 事件 5：PWM4 Reset/RollOver 事件 4：PWM3 Reset/RollOver 事件

	3：PWM2 Reset/RollOver 事件 2：PWM1 Reset/RollOver 事件 1：PWM0 Reset/RollOver 事件 0：Master PWM Reset/RollOver 事件
[1] BMOM	**Burst Mode 连续模式** 此位定义了 Burst Mode 工作模式 1：Burst Mode 工作在连续模式，计数到达 Period 值后翻转到 0 0：Burst Mode 工作在单次模式，计数到达 Period 值后停止
[0] BME	**Burst Mode 使能** 此位定义是否使能 Burst Mode 1：Burst Mode 使能 0：Burst Mode 不使能

17.5.2.116 HRPWM Burst Mode 触发寄存器 0（HRPWM_BMTRGR0）

- **名称：HRPWM Burst Mode Trigger Register 0**
- **偏移地址：0xF94**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCHPEV	EEV8	EEV7	PER4EEV8	PER0EEV7	CMPA5	REP5	RST5	CMPB4	CMPA4	REP4	CMPB3	REP3	RST3	CMPA2	REP2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST2	CMPB1	CMPA1	REP1	RST1	CMPB0	CMPA0	REP0	RST0	MCMPD	MCMPC	MCMPB	MCMPA	MREP	MRST	SW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[31] OCHPEV	**hrpwm_bm_in2 事件会启动 Burst Mode**
[30] EEV8	**PWM4 Event 8 事件会启动 Burst Mode**
[29] EEV7	**PWM0 Event 7 事件会启动 Burst Mode**
[28] PER4EEV8	**PWM4 Event 8 事件后的 Period 事件会启动 Burst Mode**
[27] PER0EEV7	**PWM0 Event 7 事件后的 Period 事件会启动 Burst Mode**
[26] CMPA5	**PWM5 Compare A 事件会启动 Burst Mode**
[25] REP5	**PWM5 Repet 事件会启动 Burst Mode**
[24] RST5	**PWM5 Reset/RollOver 事件会启动 Burst Mode**
[23] CMPB4	**PWM4 Compare B 事件会启动 Burst Mode**
[22] CMPA4	**PWM4 Compare A 事件会启动 Burst Mode**
[21] REP4	**PWM4 Repet 事件会启动 Burst Mode**
[20] CMPB3	**PWM3 Compare A 事件会启动 Burst Mode**
[19] REP3	**PWM3 Repet 事件会启动 Burst Mode**
[18] RST3	**PWM3 Reset/RollOver 事件会启动 Burst Mode**

[17] CMPA2	**PWM2 Compare A 事件会启动 Burst Mode**
[16] REP2	**PWM2 Repet 事件会启动 Burst Mode**
[15] RST2	**PWM2 Reset/RollOver 事件会启动 Burst Mode**
[14] CMPB1	**PWM1 Compare B 事件会启动 Burst Mode**
[13] CMPA1	**PWM1 Compare A 事件会启动 Burst Mode**
[12] REP1	**PWM1 Repet 事件会启动 Burst Mode**
[11] RST1	**PWM1 Reset/RollOver 事件会启动 Burst Mode**
[10] CMPB0	**PWM0 Compare B 事件会启动 Burst Mode**
[9] CMPA0	**PWM0 Compare A 事件会启动 Burst Mode**
[8] REP0	**PWM0 Repet 事件会启动 Burst Mode**
[7] RST0	**PWM0 Reset/RollOver 事件会启动 Burst Mode**
[6] MCMPD	**Master PWM Compare D 事件会启动 Burst Mode**
[5] MCMPC	**Master PWM Compare C 事件会启动 Burst Mode**
[4] MCMPB	**Master PWM Compare B 事件会启动 Burst Mode**
[3] MCMPA	**Master PWM Compare A 事件会启动 Burst Mode**
[2] MREP	**Master PWM Repet 事件会启动 Burst Mode**
[1] MRST	**Master PWM Reset/Rollover 事件会启动 Burst Mode**
[0] SW	**Software 事件启动 Burst Mode**

17.5.2.117 HRPWM Burst Mode 触发寄存器 1（HRPWM_BMTRGR1）

- **名称：HRPWM Burst Mode Trigger Register 1**
- **偏移地址：0xF98**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMPB7	CMPA7	REP7	RST7	CMPB6	CMPA6	REP6	RST6
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] CMPB7	**PWM7 Compare B 事件会启动 Burst Mode**
[6] CMPA7	**PWM7 Compare A 事件会启动 Burst Mode**
[5] REP7	**PWM7 Repet 事件会启动 Burst Mode**

[4] RST7	**PWM7 Reset/RollOver 事件会启动 Burst Mode**
[3] CMPB6	**PWM6 Compare B 事件会启动 Burst Mode**
[2] CMPA6	**PWM6 Compare A 事件会启动 Burst Mode**
[1] REP6	**PWM6 Repet 事件会启动 Burst Mode**
[0] RST6	**PWM6 Reset/RollOver 事件会启动 Burst Mode**

17.5.2.118 HRPWM Burst Mode 周期寄存器（HRPWM_BMPER）

- **名称：HRPWM Burst Mode Period Register**
- **偏移地址：0xFA0**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMPER															
R/W															
0x0															

字段	说明
[15:0] BMPER	**Burst Mode Period 数值** 此位保持 Burst Mode Period 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值

17.5.2.119 HRPWM Burst Mode 比较寄存器（HRPWM_BMCMPR）

- **名称：HRPWM Burst Mode Compare Register**
- **偏移地址：0xFA4**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMCMP															
R/W															
0x0															

字段	说明
[15:0] BMCMP	**Burst Mode Compare 数值** 此位保持 Burst Mode Compare 数值，此位为影子寄存器（预加载）数值，如果预加载功能被禁用，则此位同时也是工作寄存器数值

17.5.2.120 HRPWM Burst DMA 主机更新寄存器（HRPWM_BDMUPR）

- 名称：HRPWM Burst DMA Master Update Register
- 偏移地址：0xFB0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MCMPDR	MCMPCR	MCMPBR	MCMPAR	MREP	MPER	MCNTR	MDIER	MISR	MCR1	MCR0
					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[10] MCMPDR	**MCMPDR 寄存器会被 Burst DMA 动作更新**
[9] MCMPCR	**MCMPCR 寄存器会被 Burst DMA 动作更新**
[8] MCMPBR	**MCMPBR 寄存器会被 Burst DMA 动作更新**
[7] MCMPAR	**MCMPAR 寄存器会被 Burst DMA 动作更新**
[6] MREP	**MREP 寄存器会被 Burst DMA 动作更新**
[5] MPER	**MPER 寄存器会被 Burst DMA 动作更新**
[4] MCNTR	**MCNTR 寄存器会被 Burst DMA 动作更新**
[3] MDIER	**MDIER 寄存器会被 Burst DMA 动作更新**
[2] MISR	**MISR 寄存器会被 Burst DMA 动作更新**
[1] MCR1	**MCR1 寄存器会被 Burst DMA 动作更新**
[0] MCR0	**MCR0 寄存器会被 Burst DMA 动作更新**

17.5.2.121 HRPWM Burst DMA 更新寄存器 0（HRPWM_BDUPR0）

- 名称：HRPWM Burst DMA Update Register 0
- 偏移地址：0xFB4
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT0R	OUT0R	CAPB0CER	CAPB0CR	CAPA0CER	CAPA0CR	CHP0R	RST0ER	RST0R	EEF0R2	EEF0R1	EEF0R0	CLR0BR	SET0BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR0AR	SET0AR	DT0R	CAPB0R	CAPA0R	CMPD0R	CMPC0R	CMPB0R	CMPA0R	REP0R	PER0R	CNT0R	PWM0DIER	PWM0ISR	PWM0CR1	PWM0CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT0R	**FLT0R 寄存器会被 Burst DMA 动作更新**
[28] OUT0R	**OUT0R 寄存器会被 Burst DMA 动作更新**
[27] CAPB0CER	**CAPB0CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB0CR	**CAPB0CR 寄存器会被 Burst DMA 动作更新**
[25] CAPA0CER	**CAPA0CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA0CR	**CAPA0CR 寄存器会被 Burst DMA 动作更新**
[23] CHP0R	**CHP0R 寄存器会被 Burst DMA 动作更新**
[22] RST0ER	**RST0ER 寄存器会被 Burst DMA 动作更新**
[21] RST0R	**RST0R 寄存器会被 Burst DMA 动作更新**
[20] EEF0R2	**EEF0R2 寄存器会被 Burst DMA 动作更新**
[19] EEF0R1	**EEF0R1 寄存器会被 Burst DMA 动作更新**
[18] EEF0R0	**EEF0R0 寄存器会被 Burst DMA 动作更新**
[17] CLR0BR	**CLR0BR 寄存器会被 Burst DMA 动作更新**
[16] SET0BR	**SET0BR 寄存器会被 Burst DMA 动作更新**
[15] CLR0AR	**CLR0AR 寄存器会被 Burst DMA 动作更新**
[14] SET0AR	**SET0AR 寄存器会被 Burst DMA 动作更新**
[13] DT0R	**DT0R 寄存器会被 Burst DMA 动作更新**
[12] CAPB0R	**CAPB0R 寄存器会被 Burst DMA 动作更新**
[11] CAPA0R	**CAPA0R 寄存器会被 Burst DMA 动作更新**
[10] CMPD0R	**CMPD0R 寄存器会被 Burst DMA 动作更新**
[9] CMPC0R	**CMPC0R 寄存器会被 Burst DMA 动作更新**
[8] CMPB0R	**CMPB0R 寄存器会被 Burst DMA 动作更新**

[7] CMPA0R	**CMPA0R 寄存器会被 Burst DMA 动作更新**
[6] REP0R	**REP0R 寄存器会被 Burst DMA 动作更新**
[5] PER0R	**PER0R 寄存器会被 Burst DMA 动作更新**
[4] CNT0R	**CNT0R 寄存器会被 Burst DMA 动作更新**
[3] PWM0DIER	**PWM0DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM0ISR	**PWM0ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM0CR1	**PWM0CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM0CR0	**PWM0CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.122 HRPWM Burst DMA 更新寄存器 1（HRPWM_BDUPR1）

- **名称：HRPWM Burst DMA Update Register 1**
- **偏移地址：0xFB8**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT1R	OUT1R	CAPB1CER	CAPB1CR	CAPA1CER	CAPA1CR	CHP1R	RST1ER	RST1R	EEF1R2	EEF1R1	EEF1R0	CLR1BR	SET1BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR1AR	SET1AR	DT1R	CAPB1R	CAPA1R	CMPD1R	CMPC1R	CMPB1R	CMPA1R	REP1R	PER1R	CNT1R	PWM1DIER	PWM1ISR	PWM1CR1	PWM1CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT1R	**FLT1R 寄存器会被 Burst DMA 动作更新**
[28] OUT1R	**OUT1R 寄存器会被 Burst DMA 动作更新**
[27] CAPB1CER	**CAPB1CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB1CR	**CAPB1CR 寄存器会被 Burst DMA 动作更新**
[25] CAPA1CER	**CAPA1CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA1CR	**CAPA1CR 寄存器会被 Burst DMA 动作更新**
[23] CHP1R	**CHP1R 寄存器会被 Burst DMA 动作更新**
[22] RST1ER	**RST1ER 寄存器会被 Burst DMA 动作更新**
[21] RST1R	**RST1R 寄存器会被 Burst DMA 动作更新**
[20] EEF1R2	**EEF1R2 寄存器会被 Burst DMA 动作更新**
[19] EEF1R1	**EEF1R1 寄存器会被 Burst DMA 动作更新**
[18] EEF1R0	**EEF1R0 寄存器会被 Burst DMA 动作更新**
[17] CLR1BR	**CLR1BR 寄存器会被 Burst DMA 动作更新**

[16] SET1BR	**SET1BR 寄存器会被 Burst DMA 动作更新**
[15] CLR1AR	**CLR1AR 寄存器会被 Burst DMA 动作更新**
[14] SET1AR	**SET1AR 寄存器会被 Burst DMA 动作更新**
[13] DT1R	**DT1R 寄存器会被 Burst DMA 动作更新**
[12] CAPB1R	**CAPB1R 寄存器会被 Burst DMA 动作更新**
[11] CAPA1R	**CAPA1R 寄存器会被 Burst DMA 动作更新**
[10] CMPD1R	**CMPD1R 寄存器会被 Burst DMA 动作更新**
[9] CMPC1R	**CMPC1R 寄存器会被 Burst DMA 动作更新**
[8] CMPB1R	**CMPB1R 寄存器会被 Burst DMA 动作更新**
[7] CMPA1R	**CMPA1R 寄存器会被 Burst DMA 动作更新**
[6] REP1R	**REP1R 寄存器会被 Burst DMA 动作更新**
[5] PER1R	**PER1R 寄存器会被 Burst DMA 动作更新**
[4] CNT1R	**CNT1R 寄存器会被 Burst DMA 动作更新**
[3] PWM1DIER	**PWM1DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM1ISR	**PWM1ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM1CR1	**PWM1CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM1CR0	**PWM1CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.123 HRPWM Burst DMA 更新寄存器 2（HRPWM_BDUPR2）

- **名称：HRPWM Burst DMA Update Register 2**
- **偏移地址：0xFBC**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT2R	OUT2R	CAPB2CER	CAPB2CR	CAPA2CER	CAPA2CR	CHP2R	RST2ER	RST2R	EEF2R2	EEF2R1	EEF2R0	CLR2BR	SET2BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR2AR	SET2AR	DT2R	CAPB2R	CAPA2R	CMPD2R	CMPC2R	CMPB2R	CMPA2R	REP2R	PER2R	CNT2R	PWM2DIER	PWM2ISR	PWM2CR1	PWM2CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT2R	**FLT2R 寄存器会被 Burst DMA 动作更新**
[28] OUT2R	**OUT2R 寄存器会被 Burst DMA 动作更新**
[27] CAPB2CER	**CAPB2CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB2CR	**CAPB2CR 寄存器会被 Burst DMA 动作更新**

[25] CAPA2CER	**CAPA2CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA2CR	**CAPA2CR 寄存器会被 Burst DMA 动作更新**
[23] CHP2R	**CHP2R 寄存器会被 Burst DMA 动作更新**
[22] RST2ER	**RST2ER 寄存器会被 Burst DMA 动作更新**
[21] RST2R	**RST2R 寄存器会被 Burst DMA 动作更新**
[20] EEF2R2	**EEF2R2 寄存器会被 Burst DMA 动作更新**
[19] EEF2R1	**EEF2R1 寄存器会被 Burst DMA 动作更新**
[18] EEF2R0	**EEF2R0 寄存器会被 Burst DMA 动作更新**
[17] CLR2BR	**CLR2BR 寄存器会被 Burst DMA 动作更新**
[16] SET2BR	**SET2BR 寄存器会被 Burst DMA 动作更新**
[15] CLR2AR	**CLR2AR 寄存器会被 Burst DMA 动作更新**
[14] SET2AR	**SET2AR 寄存器会被 Burst DMA 动作更新**
[13] DT2R	**DT2R 寄存器会被 Burst DMA 动作更新**
[12] CAPB2R	**CAPB2R 寄存器会被 Burst DMA 动作更新**
[11] CAPA2R	**CAPA2R 寄存器会被 Burst DMA 动作更新**
[10] CMPD2R	**CMPD2R 寄存器会被 Burst DMA 动作更新**
[9] CMPC2R	**CMPC2R 寄存器会被 Burst DMA 动作更新**
[8] CMPB2R	**CMPB2R 寄存器会被 Burst DMA 动作更新**
[7] CMPA2R	**CMPA2R 寄存器会被 Burst DMA 动作更新**
[6] REP2R	**REP2R 寄存器会被 Burst DMA 动作更新**
[5] PER2R	**PER2R 寄存器会被 Burst DMA 动作更新**
[4] CNT2R	**CNT2R 寄存器会被 Burst DMA 动作更新**
[3] PWM2DIER	**PWM2DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM2ISR	**PWM2ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM2CR1	**PWM2CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM2CR0	**PWM2CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.124 HRPWM Burst DMA 更新寄存器 3（HRPWM_BDUPR3）

- 名称：HRPWM Burst DMA Update Register 3
- 偏移地址：0xFC0
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT3R	OUT3R	CAPB3CER	CAPB3CR	CAPA3CER	CAPA3CR	CHP3R	RST3ER	RST3R	EEF3R2	EEF3R1	EEF3R0	CLR3BR	SET3BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR3AR	SET3AR	DT3R	CAPB3R	CAPA3R	CMPD3R	CMPC3R	CMPB3R	CMPA3R	REP3R	PER3R	CNT3R	PWM3DIER	PWM3ISR	PWM3CR1	PWM3CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT3R	**FLT3R 寄存器会被 Burst DMA 动作更新**
[28] OUT3R	**OUT3R 寄存器会被 Burst DMA 动作更新**
[27] CAPB3CER	**CAPB3CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB3CR	**CAPB3CR 寄存器会被 Burst DMA 动作更新**
[25] CAPA3CER	**CAPA3CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA3CR	**CAPA3CR 寄存器会被 Burst DMA 动作更新**
[23] CHP3R	**CHP3R 寄存器会被 Burst DMA 动作更新**
[22] RST3ER	**RST3ER 寄存器会被 Burst DMA 动作更新**
[21] RST3R	**RST3R 寄存器会被 Burst DMA 动作更新**
[20] EEF3R2	**EEF3R2 寄存器会被 Burst DMA 动作更新**
[19] EEF3R1	**EEF3R1 寄存器会被 Burst DMA 动作更新**
[18] EEF3R0	**EEF3R0 寄存器会被 Burst DMA 动作更新**
[17] CLR3BR	**CLR3BR 寄存器会被 Burst DMA 动作更新**
[16] SET3BR	**SET3BR 寄存器会被 Burst DMA 动作更新**
[15] CLR3AR	**CLR3AR 寄存器会被 Burst DMA 动作更新**
[14] SET3AR	**SET3AR 寄存器会被 Burst DMA 动作更新**
[13] DT3R	**DT3R 寄存器会被 Burst DMA 动作更新**
[12] CAPB3R	**CAPB3R 寄存器会被 Burst DMA 动作更新**
[11] CAPA3R	**CAPA3R 寄存器会被 Burst DMA 动作更新**
[10] CMPD3R	**CMPD3R 寄存器会被 Burst DMA 动作更新**
[9] CMPC3R	**CMPC3R 寄存器会被 Burst DMA 动作更新**
[8] CMPB3R	**CMPB3R 寄存器会被 Burst DMA 动作更新**

[7] CMPA3R	**CMPA3R 寄存器会被 Burst DMA 动作更新**
[6] REP3R	**REP3R 寄存器会被 Burst DMA 动作更新**
[5] PER3R	**PER3R 寄存器会被 Burst DMA 动作更新**
[4] CNT3R	**CNT3R 寄存器会被 Burst DMA 动作更新**
[3] PWM3DIER	**PWM3DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM3ISR	**PWM3ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM3CR1	**PWM3CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM3CR0	**PWM3CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.125 HRPWM Burst DMA 更新寄存器 4（HRPWM_BDUPR4）

- **名称：HRPWM Burst DMA Update Register 4**
- **偏移地址：0xFC4**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT4R	OUT4R	CAPB4CER	CAPB4CR	CAPA4CER	CAPA4CR	CHP4R	RST4ER	RST4R	EEF4R2	EEF4R1	EEF4R0	CLR4BR	SET4BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR4AR	SET4AR	DT4R	CAPB4R	CAPA4R	CMPD4R	CMPC4R	CMPB4R	CMPA4R	REP4R	PER4R	CNT4R	PWM4DIER	PWM4ISR	PWM4CR1	PWM4CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT4R	**FLT4R 寄存器会被 Burst DMA 动作更新**
[28] OUT4R	**OUT4R 寄存器会被 Burst DMA 动作更新**
[27] CAPB4CER	**CAPB4CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB4CR	**CAPB4CR 寄存器会被 Burst DMA 动作更新**
[25] CAPA4CER	**CAPA4CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA4CR	**CAPA4CR 寄存器会被 Burst DMA 动作更新**
[23] CHP4R	**CHP4R 寄存器会被 Burst DMA 动作更新**
[22] RST4ER	**RST4ER 寄存器会被 Burst DMA 动作更新**
[21] RST4R	**RST4R 寄存器会被 Burst DMA 动作更新**
[20] EEF4R2	**EEF4R2 寄存器会被 Burst DMA 动作更新**
[19] EEF4R1	**EEF4R1 寄存器会被 Burst DMA 动作更新**
[18] EEF4R0	**EEF4R0 寄存器会被 Burst DMA 动作更新**
[17] CLR4BR	**CLR4BR 寄存器会被 Burst DMA 动作更新**

[16] SET4BR	**SET4BR 寄存器会被 Burst DMA 动作更新**
[15] CLR4AR	**CLR4AR 寄存器会被 Burst DMA 动作更新**
[14] SET4AR	**SET4AR 寄存器会被 Burst DMA 动作更新**
[13] DT4R	**DT4R 寄存器会被 Burst DMA 动作更新**
[12] CAPB4R	**CAPB4R 寄存器会被 Burst DMA 动作更新**
[11] CAPA4R	**CAPA4R 寄存器会被 Burst DMA 动作更新**
[10] CMPD4R	**CMPD4R 寄存器会被 Burst DMA 动作更新**
[9] CMPC4R	**CMPC4R 寄存器会被 Burst DMA 动作更新**
[8] CMPB4R	**CMPB4R 寄存器会被 Burst DMA 动作更新**
[7] CMPA4R	**CMPA4R 寄存器会被 Burst DMA 动作更新**
[6] REP4R	**REP4R 寄存器会被 Burst DMA 动作更新**
[5] PER4R	**PER4R 寄存器会被 Burst DMA 动作更新**
[4] CNT4R	**CNT4R 寄存器会被 Burst DMA 动作更新**
[3] PWM4DIER	**PWM4DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM4ISR	**PWM4ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM4CR1	**PWM4CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM4CR0	**PWM4CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.126 HRPWM Burst DMA 更新寄存器 5（HRPWM_BDUPR5）

- **名称：HRPWM Burst DMA Update Register 5**
- **偏移地址：0xFC8**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT5R	OUT5R	CAPB5CER	CAPB5CR	CAPA5CER	CAPA5CR	CHP5R	RST5ER	RST5R	EEF5R2	EEF5R1	EEF5R0	CLR5BR	SET5BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR5AR	SET5AR	DT5R	CAPB5R	CAPA5R	CMPD5R	CMPC5R	CMPB5R	CMPA5R	REP5R	PER5R	CNT5R	PWM5DIER	PWM5ISR	PWM5CR1	PWM5CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT5R	**FLT5R 寄存器会被 Burst DMA 动作更新**
[28] OUT5R	**OUT5R 寄存器会被 Burst DMA 动作更新**
[27] CAPB5CER	**CAPB5CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB5CR	**CAPB5CR 寄存器会被 Burst DMA 动作更新**

[25] CAPA5CER	**CAPA5CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA5CR	**CAPA5CR 寄存器会被 Burst DMA 动作更新**
[23] CHP5R	**CHP5R 寄存器会被 Burst DMA 动作更新**
[22] RST5ER	**RST5ER 寄存器会被 Burst DMA 动作更新**
[21] RST5R	**RST5R 寄存器会被 Burst DMA 动作更新**
[20] EEF5R2	**EEF5R2 寄存器会被 Burst DMA 动作更新**
[19] EEF5R1	**EEF5R1 寄存器会被 Burst DMA 动作更新**
[18] EEF5R0	**EEF5R0 寄存器会被 Burst DMA 动作更新**
[17] CLR5BR	**CLR5BR 寄存器会被 Burst DMA 动作更新**
[16] SET5BR	**SET5BR 寄存器会被 Burst DMA 动作更新**
[15] CLR5AR	**CLR5AR 寄存器会被 Burst DMA 动作更新**
[14] SET5AR	**SET5AR 寄存器会被 Burst DMA 动作更新**
[13] DT5R	**DT5R 寄存器会被 Burst DMA 动作更新**
[12] CAPB5R	**CAPB5R 寄存器会被 Burst DMA 动作更新**
[11] CAPA5R	**CAPA5R 寄存器会被 Burst DMA 动作更新**
[10] CMPD5R	**CMPD5R 寄存器会被 Burst DMA 动作更新**
[9] CMPC5R	**CMPC5R 寄存器会被 Burst DMA 动作更新**
[8] CMPB5R	**CMPB5R 寄存器会被 Burst DMA 动作更新**
[7] CMPA5R	**CMPA5R 寄存器会被 Burst DMA 动作更新**
[6] REP5R	**REP5R 寄存器会被 Burst DMA 动作更新**
[5] PER5R	**PER5R 寄存器会被 Burst DMA 动作更新**
[4] CNT5R	**CNT5R 寄存器会被 Burst DMA 动作更新**
[3] PWM5DIER	**PWM5DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM5ISR	**PWM5ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM5CR1	**PWM5CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM5CR0	**PWM5CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.127 HRPWM Burst DMA 更新寄存器 6（HRPWM_BDUPR6）

- 名称：HRPWM Burst DMA Update Register 6
- 偏移地址：0xFCC
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT6R	OUT6R	CAPB6CER	CAPB6CR	CAPA6CER	CAPA6CR	CHP6R	RST6ER	RST6R	EEF6R2	EEF6R1	EEF6R0	CLR6BR	SET6BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR6AR	SET6AR	DT6R	CAPB6R	CAPA6R	CMPD6R	CMPC6R	CMPB6R	CMPA6R	REP6R	PER6R	CNT6R	PWM6DIER	PWM6ISR	PWM6CR1	PWM6CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT6R	**FLT6R 寄存器会被 Burst DMA 动作更新**
[28] OUT6R	**OUT6R 寄存器会被 Burst DMA 动作更新**
[27] CAPB6CER	**CAPB6CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB6CR	**CAPB6CR 寄存器会被 Burst DMA 动作更新**
[25] CAPA6CER	**CAPA6CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA6CR	**CAPA6CR 寄存器会被 Burst DMA 动作更新**
[23] CHP6R	**CHP6R 寄存器会被 Burst DMA 动作更新**
[22] RST6ER	**RST6ER 寄存器会被 Burst DMA 动作更新**
[21] RST6R	**RST6R 寄存器会被 Burst DMA 动作更新**
[20] EEF6R2	**EEF6R2 寄存器会被 Burst DMA 动作更新**
[19] EEF6R1	**EEF6R1 寄存器会被 Burst DMA 动作更新**
[18] EEF6R0	**EEF6R0 寄存器会被 Burst DMA 动作更新**
[17] CLR6BR	**CLR6BR 寄存器会被 Burst DMA 动作更新**
[16] SET6BR	**SET6BR 寄存器会被 Burst DMA 动作更新**
[15] CLR6AR	**CLR6AR 寄存器会被 Burst DMA 动作更新**
[14] SET6AR	**SET6AR 寄存器会被 Burst DMA 动作更新**
[13] DT6R	**DT6R 寄存器会被 Burst DMA 动作更新**
[12] CAPB6R	**CAPB6R 寄存器会被 Burst DMA 动作更新**
[11] CAPA6R	**CAPA6R 寄存器会被 Burst DMA 动作更新**
[10] CMPD6R	**CMPD6R 寄存器会被 Burst DMA 动作更新**
[9] CMPC6R	**CMPC6R 寄存器会被 Burst DMA 动作更新**
[8] CMPB6R	**CMPB6R 寄存器会被 Burst DMA 动作更新**

[7] CMPA6R	**CMPA6R 寄存器会被 Burst DMA 动作更新**
[6] REP6R	**REP6R 寄存器会被 Burst DMA 动作更新**
[5] PER6R	**PER6R 寄存器会被 Burst DMA 动作更新**
[4] CNT6R	**CNT6R 寄存器会被 Burst DMA 动作更新**
[3] PWM6DIER	**PWM6DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM6ISR	**PWM6ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM6CR1	**PWM6CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM6CR0	**PWM6CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.128 HRPWM Burst DMA 更新寄存器 7（HRPWM_BDUPR7）

- **名称：HRPWM Burst DMA Update Register 7**
- **偏移地址：0xFD0**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		FLT7R	OUT7R	CAPB7CER	CAPB7CR	CAPA7CER	CAPA7CR	CHP7R	RST7ER	RST7R	EEF7R2	EEF7R1	EEF7R0	CLR7BR	SET7BR
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR7AR	SET7AR	DT7R	CAPB7R	CAPA7R	CMPD7R	CMPC7R	CMPB7R	CMPA7R	REP7R	PER7R	CNT7R	PWM7DIER	PWM7ISR	PWM7CR1	PWM7CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[29] FLT7R	**FLT7R 寄存器会被 Burst DMA 动作更新**
[28] OUT7R	**OUT7R 寄存器会被 Burst DMA 动作更新**
[27] CAPB7CER	**CAPB7CER 寄存器会被 Burst DMA 动作更新**
[26] CAPB7CR	**CAPB7CR 寄存器会被 Burst DMA 动作更新**
[25] CAPA7CER	**CAPA7CER 寄存器会被 Burst DMA 动作更新**
[24] CAPA7CR	**CAPA7CR 寄存器会被 Burst DMA 动作更新**
[23] CHP7R	**CHP7R 寄存器会被 Burst DMA 动作更新**
[22] RST7ER	**RST7ER 寄存器会被 Burst DMA 动作更新**
[21] RST7R	**RST7R 寄存器会被 Burst DMA 动作更新**
[20] EEF7R2	**EEF7R2 寄存器会被 Burst DMA 动作更新**
[19] EEF7R1	**EEF7R1 寄存器会被 Burst DMA 动作更新**
[18] EEF7R0	**EEF7R0 寄存器会被 Burst DMA 动作更新**
[17] CLR7BR	**CLR7BR 寄存器会被 Burst DMA 动作更新**

[16] SET7BR	**SET7BR 寄存器会被 Burst DMA 动作更新**
[15] CLR7AR	**CLR7AR 寄存器会被 Burst DMA 动作更新**
[14] SET7AR	**SET7AR 寄存器会被 Burst DMA 动作更新**
[13] DT7R	**DT7R 寄存器会被 Burst DMA 动作更新**
[12] CAPB7R	**CAPB7R 寄存器会被 Burst DMA 动作更新**
[11] CAPA7R	**CAPA7R 寄存器会被 Burst DMA 动作更新**
[10] CMPD7R	**CMPD7R 寄存器会被 Burst DMA 动作更新**
[9] CMPC7R	**CMPC7R 寄存器会被 Burst DMA 动作更新**
[8] CMPB7R	**CMPB7R 寄存器会被 Burst DMA 动作更新**
[7] CMPA7R	**CMPA7R 寄存器会被 Burst DMA 动作更新**
[6] REP7R	**REP7R 寄存器会被 Burst DMA 动作更新**
[5] PER7R	**PER7R 寄存器会被 Burst DMA 动作更新**
[4] CNT7R	**CNT7R 寄存器会被 Burst DMA 动作更新**
[3] PWM7DIER	**PWM7DIER 寄存器会被 Burst DMA 动作更新**
[2] PWM7ISR	**PWM7ISR 寄存器会被 Burst DMA 动作更新**
[1] PWM7CR1	**PWM7CR1 寄存器会被 Burst DMA 动作更新**
[0] PWM7CR0	**PWM7CR0 寄存器会被 Burst DMA 动作更新**

17.5.2.129 HRPWM Burst DMA 数据起始地址寄存器（HRPWM_BDMADR）

- **名称：HRPWM Burst DMA Address Register**
- **偏移地址：0xFF0**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BDMADR															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDMADR															
R/W															
0x0															

字段	说明
[31:0] BDMADR	**Burst DMA 数据起始地址** 此位存放 Burst DMA 数据起始地址，Burst DMA 从起始地址开始顺次取出数据并更新到寄存器

18 内部集成接口（I2C）

18.1 简介

I2C 总线是一个两线串行接口，其中两线位串行数据（SDA）和串行时钟（SCL）线在总线器件间传递信息。每个器件都有一个唯一的地址识别，而且都可以作为一个发送或接收器。除了发送器和接收器外，器件在执行数据传输时也可以被看做是主机或者从机。主机是初始化总线的数据传输并产生允许传输的时钟信号的器件。I2C 支持标准模式（Sm）、快速模式（Fm）和快速模式增强（Fm+）。此外它还与 SMBUS（系统管理总线）。

18.2 主要特性

- I2C 特性
  - 支持主模式和从模式
  - 支持标准模式（高达 100 kHz）
  - 支持快速模式（高达 400 kHz）
  - 支持快速+模式（高达 1 MHz）
  - 支持 7 位和 10 位寻址模式
  - 支持 DMA 操作
  - 支持可编程建立时间和保持时间
  - 支持产生 START、STOP、ACK 信号的检测
  - 支持错误指示
  - 支持深度 16 的 RXFIFO 和 TXFIFO
  - 支持中断和轮询操作
- SMBUS 特性
  - 支持硬件 PEC（数据包错误检查）的生成和验证
  - 支持地址解析协议（ARP）支持
  - 支持主模式和从模式
  - 支持 SMBUS 警报的产生和检测
  - 支持 SMBUS 超时和空闲状态检测

18.3 结构框图

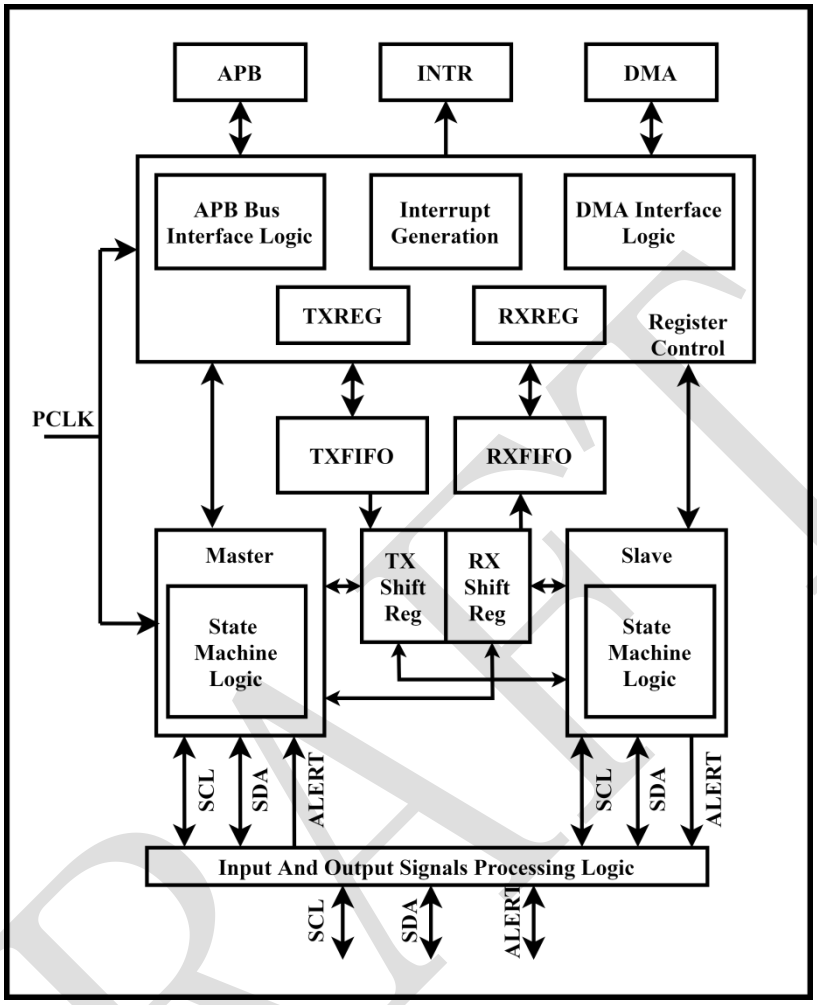


图 18-1 I2C 顶层结构框图

18.4 功能描述

18.4.1 I2C 协议

18.4.1.1 I2C 电气结构

SDA 和 SCL 都是双向线路，两者均连接了上拉电阻。（见图 18-2）当 I2C 总线处于空闲状态时，SCL 和 SDA 两条线均为高电平。连接到总线的设备的输出极必须处于漏极开路或集电极开路才能执行线与功能。

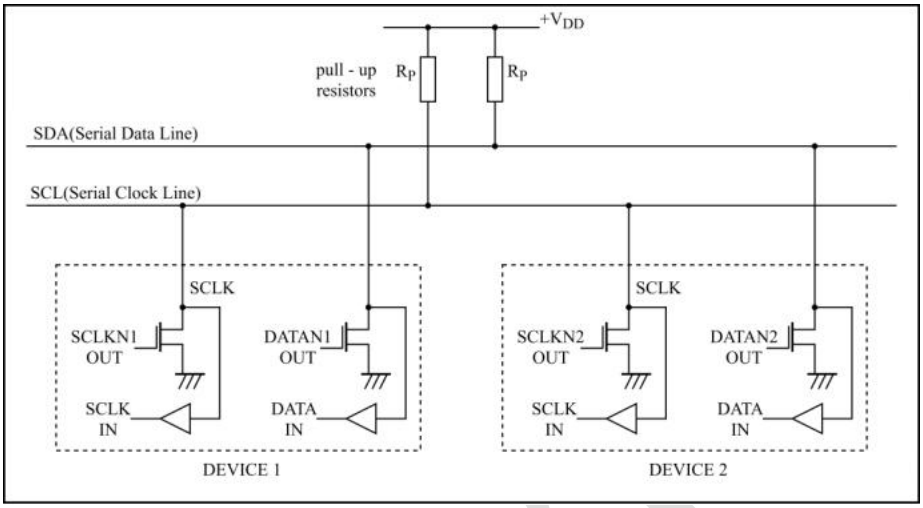


图 18-2　I2C 总线电气结构图

18.4.1.2　I2C 基本结构

I2C 基本结构由下：

- Transmitter：将数据发送到总线的设备。Transmitter 可以是发送数据到总线的主机设备（Master-Transmitter），也可以是响应主机的要求将数据发送到总线的从机设备（Slave-Transmitter）。
- Receiver：从总线接收数据的设备。Receiver 可以是根据自己的请求接收数据的设备（Master-Receiver），也可以是响应于来自主设备的请求的设备（Slave-Receiver）。
- Master：初始化传输（START 命令），生成时钟（SCL）信号并终止传输（STOP 命令）的组件。主机可以是 Transmitter 或 Receiver。
- Slave：主机寻址的设备。从机可以是 Transmitter 或 Receiver。
- SDA：数据信号线（串行数据）。
- SCL：时钟信号线（串行时钟）。
- START（RESTART）：数据传输以 START 或 RESTART 开始。SDA 数据线的电平从高电平变为低电平，而 SCL 时钟线保持高电平。当发生这种情况时，总线进入忙碌状态。
- STOP：数据传输因 STOP 而终止。当 SDA 数据线上的电平从低电平变为高电平，而 SCL 时钟线保持高电平时，就会发生这种情况。数据传输终止后，总线将再次空闲。

上述结构关系如图 18-3 所示。

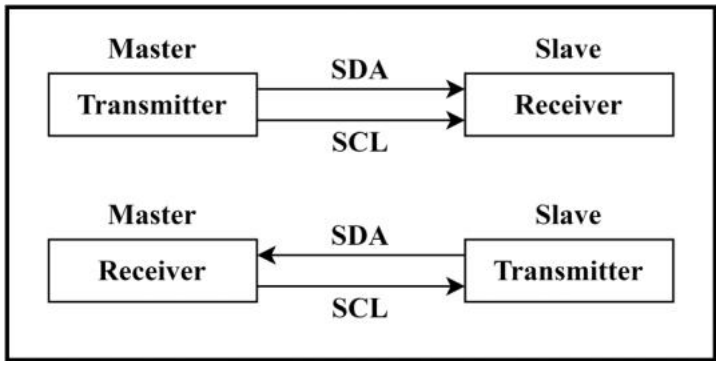


图 18-3　主/从和发送器/接收器的关系

Master 负责产生时钟并控制数据传输。Slave 负责向 Master 发送数据或从 Master 接收数据。数据的 ACK/NACK 由接收数据的设备发送，该设备可以是 Master，也可以是 Slave。

每个 Slave 都有一个由系统设计人员确定的唯一地址。当 Master 要与 Slave 通信时，Master 会先发送一个 START / RESTART 命令，接着发送 Slave 的地址和一个控制位（R/W）。Slave 在接收到地址之后如果匹配会发送一个 ACK 脉冲。

如果 Master（Transmitter）发送一个主机写的命令，此时 Receiver 应准备接收数据，此传输在检测到 Master 发送了 STOP 时终止。如果 Master 发送一个主机读的命令，此时 Receiver 应准备发送数据，接收过程中 Master 将回复 ACK，Master 在接收到最后一个字节后发送 NACK，Slave 收到 NACK 后放开总线，此时 Master 发出 STOP 或者发出 RESTART 进行下一次传输。上述行为如图 18-4 所示。

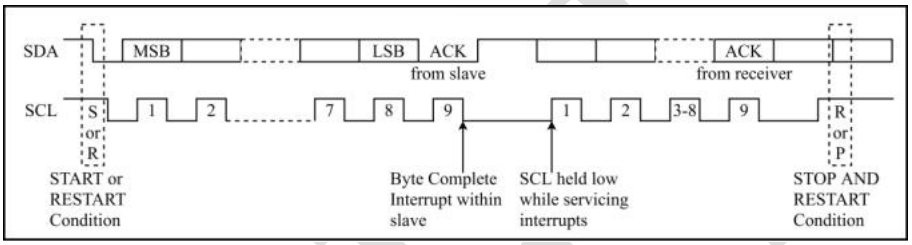


图 18-4　I2C 总线的数据传输

18.4.1.3　STOP 和 START

总线空闲时，SCL 和 SDA 信号均通过总线上的外部上拉电阻上拉。当 Master 需要在总线上开始传输时，Master 发出 START（SCL 为高时 SDA 信号由高到低的跳变）；当 Master 需要终止传输时，主机发出 STOP（SCL 为高时 SDA 线从低到高的跳变）。当 SCL 为高时，SDA 线必须稳定。图 18-5 显示了 START 和 STOP 的时序。

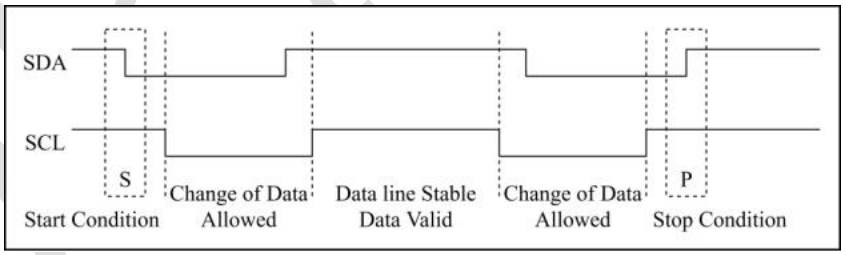


图 18-5　START 和 STOP 的时序

18.4.1.4　寻址从协议

地址格式有两种：7 位地址格式和 10 位地址格式。

7 位地址格式

在 7 位地址格式中，第一个字节的前七个位（位 7:1）设置从机地址，而 LSB 位（位 0）为 R/W 位，如图 18-6 所示。当位 0（R/W）设置为 0 时，Master 写入 Slave。当位 0（R/W）设置为 1 时，Master 将从 Slave 读取数据。

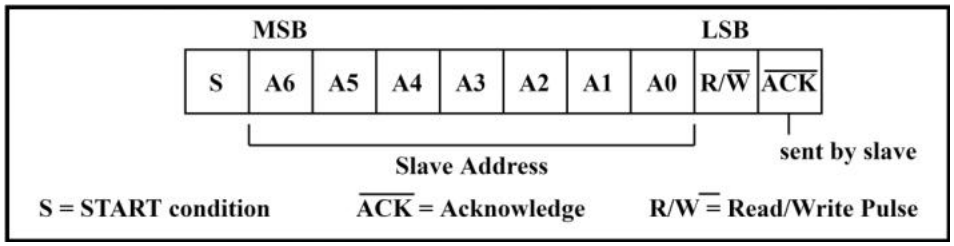


图 18-6　7 位地址格式

10 位地址格式

在 10 位寻址期间，将传输两个字节以设置 10 位地址。第一个字节的传输包含以下位定义。前五个位（位 7:3）通知 Slave 这是 10 位传输，其后是后两个位（位 2:1），它们是 Slave 地址位 9:8，LSB 位是 R / W 位。传输的第二个字节设置 Slave 地址的位 7:0。10 位地址格式如图 22-7 所示。

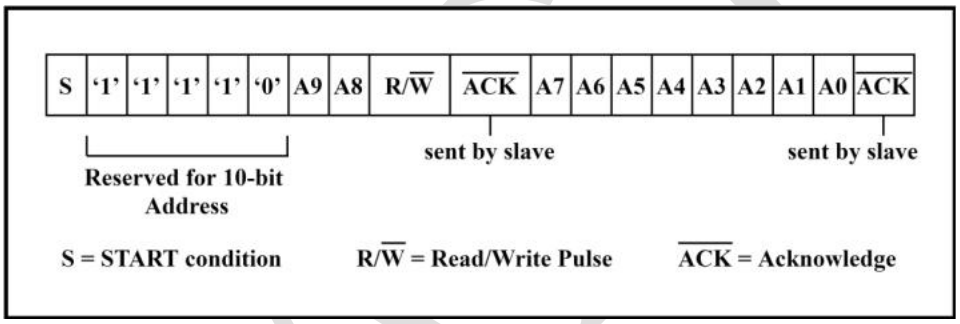


图 18-7　10 位地址格式

表 18-1 定义了特殊用途和保留的第一个字节地址。

表 18-1　I2C / SMBus 第一个字节中的位定义

Slave Address	R/W Bit	Description
0000 000	0	通用呼叫地址。i2c 将数据放置在 RXFIFO 中，并发出常规调用中断。
0000 000	1	START 字节。
0000 001	X	CBUS 地址。i2c 忽略这些访问。
0000 010	X	Reserved.
0000 011	X	Reserved
0000 1XX	X	高速主模式
1111 1XX	X	Reserved
1111 0XX	X	10 位从机寻址。
0001 000	X	SMBus Host
0001 100	X	SMBus 警报响应地址
1100 001	X	SMBus 设备默认地址

I2C 不会限制使用这些保留的地址。但是，如果使用这些保留的地址，则可能会与其他 I2C 组件不兼容。

地址组合模式

在有 10 位寻址的传输中可能会有不同的读/写格式组合，可能的数据传输格式有：

1. 主机发送器将 10 位从机地址发送到从机接收器

如图 18-8 所示，传输的方向不会改变，当起始条件后有 10 位地址时，每个从机将从机地址第一个字节的头 7 位(11110XX)与自己的地址比较并测试看第 8 位(R/W)方向位是否为 0。此时，从机若发现第一个地址相同，则回复一个 ACK。主机开始发送第二个字节的 8 位地址(XXXXXXXX)，从机进行比较，若匹配则回复 ACK。匹配的从机将保持被主机寻址，直到接收到 STOP 或者从机地址不同的 RESTART。

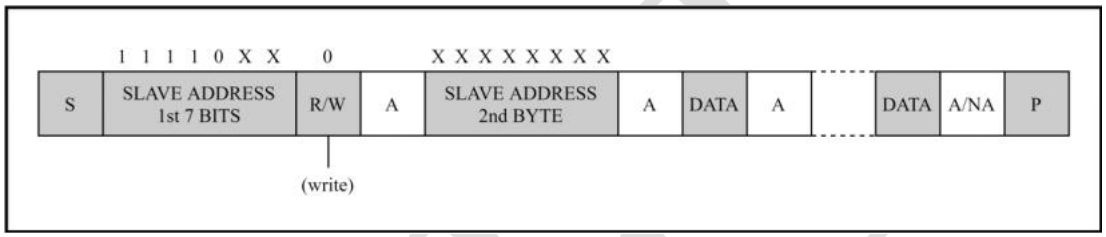


图 18-8　主机发送器寻址从机接收器

2. 主机接收器将 10 位从机地址发送到从机发送器

如图 18-9 所示，传输方向在第 2 个 R/W 位改变。在发送 RESTART（Sr）之前都和主机发送器将 10 位从机地址发送到从机接收器的寻址方式相同，在发送 RESTART（Sr）之后，匹配的从机地址记得它之前被寻址过，所以此时只需要检查第一个字节的头 7 位是否和 START（S）之后的相同，并检查第 8 位 R/W 是否为 1，如果第一个字节也相同并且 R/W 也为 1，从机会认为它作为发送器被寻址，然后回复 ACK。若多从机的情况下，RESTART 后的这个字节也会被发送到其他的从机，但是因为其他的从机没有事先被寻址，所以不能匹配。

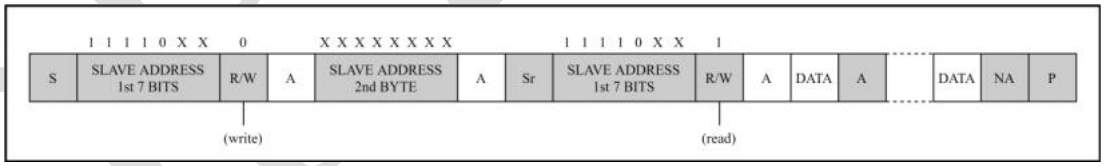


图 18-9　主机接收器寻址从机发送器

3. 组合格式 1。

主机发送数据到从机然后从相同的主机读数据，相同的主机始终占用着总线，但是传输方向在第二个 R/W 位发生改变，具体如图 18-10 所示。

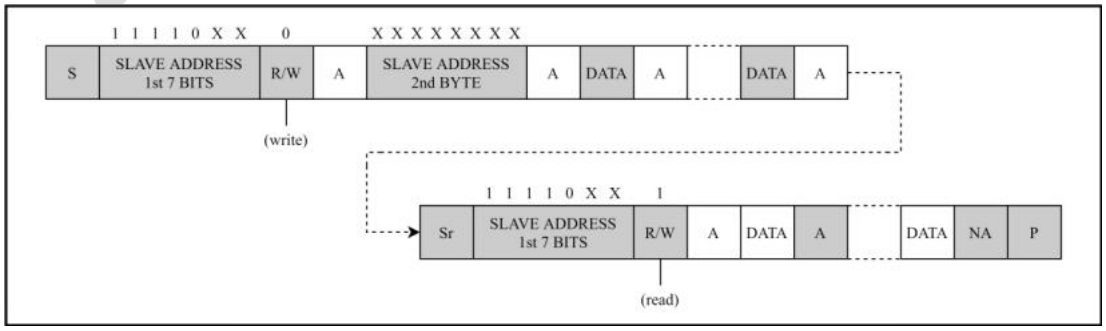


图 18-10　主机组合寻址从机（先主机写后主机读）

4. 组合格式 2。

主机发送数据到一个从机，然后发送 RESTART 再发送一次数据，具体如图 18-11 所示。

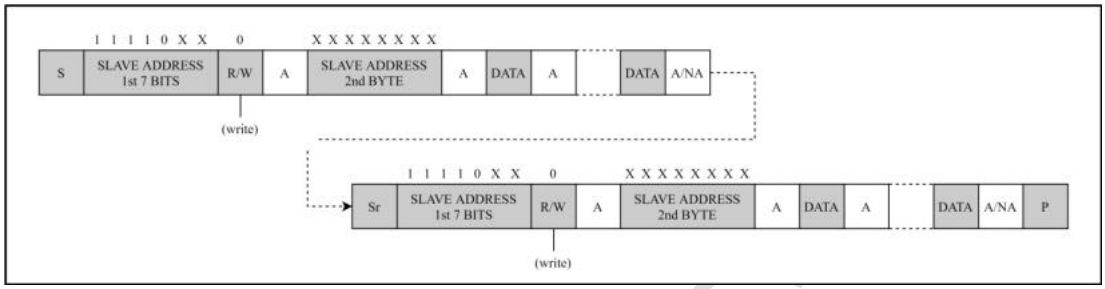


图 18-11　主机组合寻址从机（先主机写后主机写）

18.4.1.5　收发协议

Master 可以作为 Master-Transmitter 或 Master-Receiver。Slave 响应来自 Master 的请求，以分别向总线发送数据或从总线接收数据，分别充当 Slave-Transmitter 或 Slave-Receiver。

主发送器和从接收器

所有数据均以字节格式传输，每次数据传输的字节数没有限制。在 Master 发送地址和 R/W 位或 Master 向 Slave 发送数据字节之后，Slave-Receiver 必须回复 ACK。当 Slave-Receiver 回复 NACK 时，Master 通过发出 STOP 终止传输。如图 18-12 所示，Slave 必须将放开 SDA 线，以便 Master 可以终止传输。如果 Master-Transmitter 正在发送数据， Slave-Receiver 在接收到每个字节的数据后用 ACK 响应 Master-Transmitter。

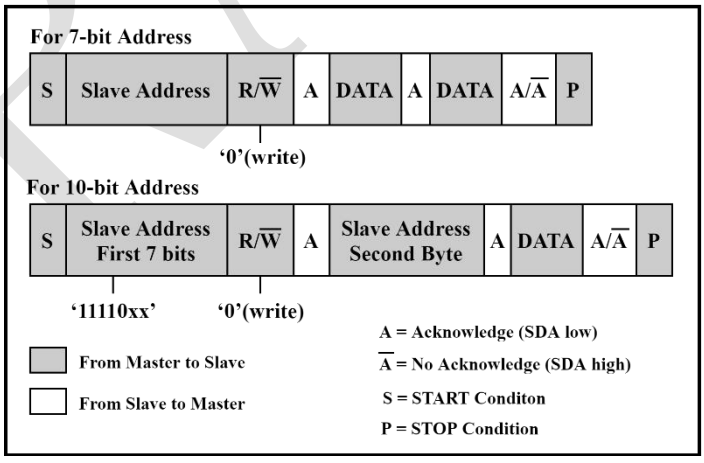


图 18-12　主发送器协议

主接收器和从发送器

如图 18-13 所示，如果 Master 正在接收数据，则在接收到一个字节的数据（最后一个字节除外）之后，Master 将回复 ACK；如果 Master 回复的是 NACK，那么 Slave-Transmitter 应该认为这是最后一个字节，Slave-Transmitter 检测到无应答（NACK）后放开 SDA 线，Master 可以发出 STOP。

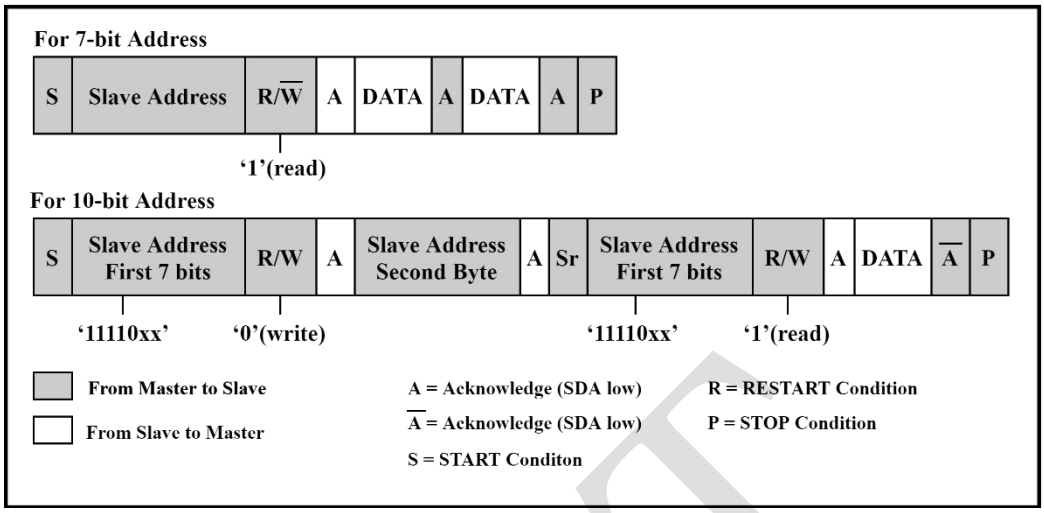


图 18-13　主接收器协议

当 Master 不希望以 STOP 放弃总线时，Master 可以发出 RESTART。RESTART 波形与 START 相同。

18.4.2　I2C 从机模式

I2C 在同一时间里只能设置为 I2C 主机模式或者 I2C 从机模式，不能同时进行。通过配置 I2C_ENABLE.MSTEN 位来配置 I2C 作为 Master 还是 Slave，MSTEN 为 1 时是主机，MSTEN 为 0 时是从机。参考的从机配置流程图如图 18-14 所示。

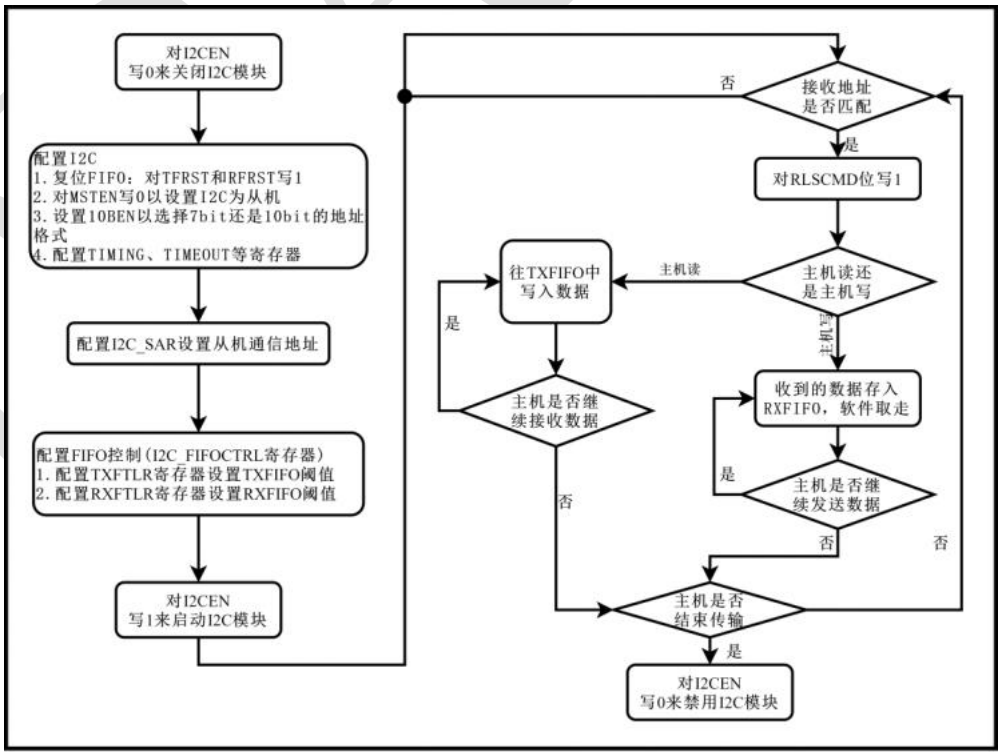


图 18-14　从机配置流程

18.4.2.1 初始化配置

要将 I2C 用作从机时，需要在启动 I2C 前执行以下步骤：

1. 对 I2C_ENABLE.I2CEN 位写入 0 来关闭 I2C
2. 写入 I2C_SAR 寄存器来设置从机通讯地址。
3. 写入 I2C_ENABLE.A10BEN 位来确定用哪种寻址类型（A10BEN 为 1 是 10 位寻址，为 0 是 7 位寻址）
4. I2C_ENABLE.MSTEN 位写入 0 来跑 I2C 从机模式
5. 中断、时序和超时选配（I2C_INTREN 寄存器、I2C_TIMING 寄存器和 I2C_TIMEOUT 寄存器）
6. 对 I2C_ENABLE.I2CEN 位写入 1 来启动 I2C

注意：步骤 2,3,4,5 可以不需要严格按照顺序配置。

18.4.2.2 从机接收器

当总线上的一个主设备寻址 I2C 并发送数据时，此时 I2C 充当从机接收器，具体如下：

1. 外部主设备启动 I2C 传输，首先会发送 START 信号。I2C 检测到 START 信号会触发 STDETI 中断（I2C_INTR.STDETI 位）
2. 发送完 START 信号之后外部主设备会进行寻址。若地址和 I2C_SAR 不匹配，I2C 会回复 NACK，并触发 NACKI 中断（I2C_INTR. NACKI 位），I2C_STATUS 寄存器中相应的 S7BNMTC、S10B1NMTC 和 S10B2NMTC 状态位会更新；若地址和 I2C_SAR 寄存器匹配，I2C 会回复 ACK，同时触发 RXADI 中断（I2C_INTR.RXADI 位），代表寻址完成。
3. I2C 检测到 RXADI 中断(I2C_INTR.RXADI 位)代表寻址完成，对 I2C_ENABLE.RLSCMD 位写入 1 以清除计数
4. 外部设备收到寻址的 ACK 之后会发送数据，I2C 收到数据会先放入接收缓冲区（RXFIFO），I2C 可以根据 RXFIFO 的状态和中断来取数据（读取 I2C_RXDATA 寄存器）。

注意：主设备可以通过拉住 SCL 为低来保持住总线，也可以通过发送 STOP 信号来释放总线

18.4.2.3 从机发送器

当总线上的一个主设备寻址 I2C 并请求数据时，此时 I2C 充当从机发送器，具体如下：

1. 外部主设备启动 I2C 传输，首先会发送 START 信号。I2C 检测到 START 信号会触发 STDETI 中断（I2C_INTR.STDETI 位）
2. 发送完 START 信号之后外部主设备会进行寻址。若地址和 I2C_SAR 寄存器不匹配，I2C 会回复 NACK，并触发 NACKI 中断（I2C_INTR. NACKI 位），I2C_STATUS 寄存器中相应的 S7BNMTC、S10B1NMTC 和 S10B2NMTC 状态位会更新；若地址和 I2C_SAR 寄存器匹配，I2C 会回复 ACK，同时触发 TXADI 中断（I2C_INTR.TXADI 位），代表寻址完成。
3. I2C 检测到 RXADI 中断(I2C_INTR.RXADI 位)代表寻址完成，对 I2C_ENABLE.RLSCMD 位写入 1 以清除计数
4. I2C 在收到 TXADI 中断(I2C_INTR.TXADI 位)后，软件需要第一时间将数据填到 TXFIFO

中，此时 I2C 会拉住 SCL 直到 TXFIFO 中有数为止，该状态为从机等待 TXFIFO 填数的状态，每次进入该状态，相应的 SWTXI 中断（I2C_INTR.SWTXI 位）将会被置起。

5. I2C 根据 TXFIFO 的状态和中断来填数，直到主机回复 NACK 或者发起 STOP 为止。

18.4.3 I2C 主机模式

I2C 在同一时间里只能设置为 I2C 主机模式或者 I2C 从机模式，不能同时进行。通过配置 I2C_ENABLE.MSTEN 位来配置 I2C 作为 Master 还是 Slave，MSTEN 为 1 时是主机，MSTEN 为 0 时是从机。参考的主机配置流程图如图 18-15 所示。

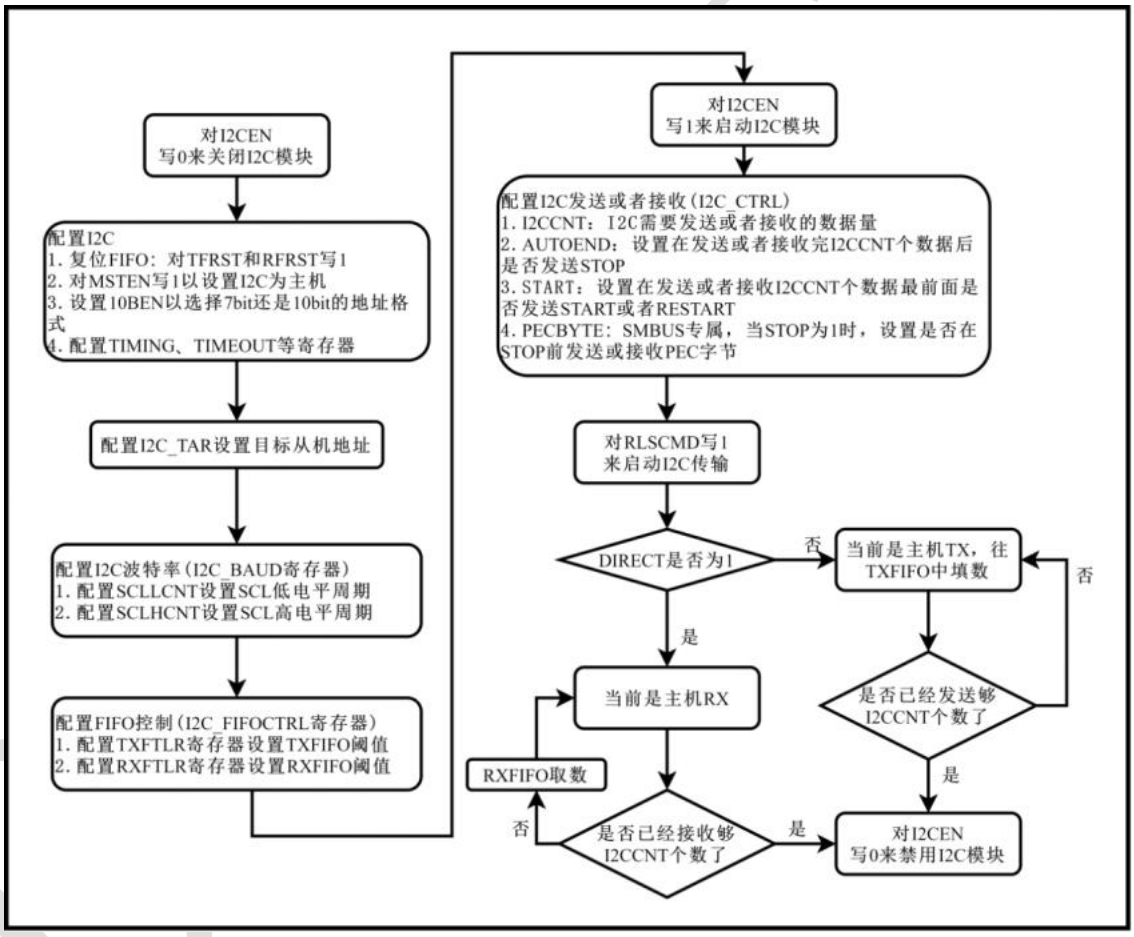


图 18-15　主机参考配置流程

18.4.3.1 初始化配置

要将 I2C 用作主机时，需要在启动 I2C 前执行以下步骤：

1. 对 I2C_ENABLE.I2CEN 位写入 0 来关闭 I2C
2. 写入 I2C_TAR 寄存器来设置主机要通讯的从机地址。
3. 写入 I2C_ENABLE.A10BEN 位来确定用哪种寻址类型（A10BEN 为 1 是 10 位寻址，为 0 是 7 位寻址）
4. I2C_ENABLE.MSTEN 位写入 1 来跑 I2C 主机模式
5. 写入 I2C_BAUD.SCLHCNT 位和 I2C_BAUD.SCLLCNT 来确定 I2C 总线上的速度

6. 中断、时序和超时选配（I2C_INTREN 寄存器、I2C_TIMING 寄存器和 I2C_TIMEOUT 寄存器）
7. 对 I2C_ENABLE.I2CEN 位写入 1 来启动 I2C

注意：步骤 2,3,4,5,6 可以不需要严格按照顺序配置。

18.4.3.2 主机发送器

当 I2C 作为主设备寻址外面的一个从设备并发送数据时，此时 I2C 充当主机发送器，具体如下：

1. 初始化配置之后，对 I2C_CTRL 寄存器写入指令，包括发送的数据量（I2CCNT），传输方向设置为 0（DIRECT），是否在发完 I2CCNT 个数据后发出 STOP（AUTOEND），是否在发送的第一个数前发送 START 或者 RESTART（START 和是否在 STOP 之前发送 PEC 数据（PECBYTE），上述配置都一起写入 I2C_CTRL 寄存器中
2. 对 I2C_ENABLE.RLSCMD 位写入 1，如果此时 START 位为 1，I2C 会发送 START 和地址来寻址，如果地址接收到 NACK，会触发 NACKI 中断，相应的 M7BNACK 或者 M10B1NACK 或者 M10B2NACK 状态位会起来；如果地址接收到 ACK，那么寻址成功。
3. 寻址成功后，I2C 会查看 TXFIFO 情况，如果 FIFO 是空的，那么 I2C 会拉住 SCL 为低，等待软件给 TXFIFO 填数；若 TXFIFO 不为空，那么 I2C 会将 TXFIFO 中的数据发送出去。
4. 发送完 I2CCNT 个数据之后，会根据 STOP 的配置决定是否需要发送 STOP，若 STOP 为 1，则发送 STOP；若 STOP 为 0，则 I2C 拉住 SCL 为低。此外，发送完 I2CCNT 个数据之后会触发 MDEI 中断（I2C_INTR.MDEI 位）。
5. 如果需要继续其他指令，则跳回到 1，如果不需要则对 I2CEN 写 0 关闭 I2C 传输。

18.4.3.3 主机接收器

当 I2C 作为主设备寻址外面的一个从设备并发送数据时，此时 I2C 充当主机发送器，具体如下：

1. 初始化配置之后，对 I2C_CTRL 整个寄存器写入指令，包括发送的数据量（I2CCNT），传输方向设置为 1（DIRECT），是否在发完 I2CCNT 个数据后发出 STOP（AUTOEND），是否在发送的第一个数前发送 START 或者 RESTART（START 和是否在 STOP 之前发送 PEC 数据（PECBYTE），上述配置都一起写入 I2C_CTRL 寄存器中
2. 对 I2C_ENABLE.RLSCMD 位写入 1，如果此时 START 位为 1，I2C 会发送 START 和地址来寻址，如果地址接收到 NACK，会触发 NACKI 中断，相应的 M7BNACK 或者 M10B1NACK 或者 M10B2NACK 状态位会起来；如果地址接收到 ACK，那么寻址成功。
3. 寻址成功后，I2C 会检查从机是否拉住总线，因为从机收到主机读指令之后要快速准备好数据可供发送，如果 I2C 检查到从机拉住了总线，则会停止运行，等待从机放开总线。从机放开总线之后，主机会发送 SCL 时钟进行接收数据。
4. 接收完 I2CCNT 个数据之后，会根据 STOP 的配置决定是否需要发送 STOP，若 STOP 为 1，则发送 STOP；若 STOP 为 0，则 I2C 拉住 SCL 为低。此外，发送完 I2CCNT 个数据之后会触发 MDEI 中断。
5. 如果需要继续其他指令，则跳回到 1，如果不需要则对 I2CEN 写 0 关闭 I2C 传输。

18.4.4 I2C 时序配置

根据 I2C 标准协议文档（www.i2c.org）和 SMBUS 标准协议文档（http://smbus.org）的时序，整理出如表 18-2 所示的时序整合

表 18-2 I2C 和 SMBUS 标准时序表

参数	描述	标准模式		快速模式		快速+模式		单位
		Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL 时钟频率	-(I2C) 10(SMBUS)	100	-(I2C) 10(SMBUS)	400	-(I2C) 10(SMBUS)	1000	kHz
$t_{HD:STA}$	START 的保持时间。	4.0	-	0.6	-	0.26	-	us
$t_{SU:STA}$	START 的建立时间	4.7	-	0.6	-	0.26	-	us
$t_{SU:STO}$	STOP 的建立时间	4.0	-	0.6	-	0.26	-	us
$t_{HD:DAT}$ ①	数据保持时间	0	-	0	-	0	-	us
$t_{SU:DAT}$ ②	数据建立时间	250	-	100	-	50	-	ns
t_{BUF}	STOP 和 START 之间的总线空闲时间	4.7	-	1.3	-	0.5	-	us
t_{LOW} ③	SCL 时钟的低电平时间	4.7	-	1.3	-	0.5	-	us
t_{HIGH} ③	SCL 时钟的高电平时间	4.0	-	0.6	-	0.26	-	us
t_r	SDA 和 SCL 的上升时间	-	1000	-	300	-	120	ns
t_f	SDA 和 SCL 的下降时间	-	300	-	300	-	120	ns
t_{SP} ④	输入滤波器必须抑制的毛刺脉宽	-	-	0	50	0	50	ns
$t_{LOW:SEXT}$ ⑤	从设备检测时钟低电平延长时间(SMBUS)	-	25	-	25	-	25	ms
$t_{LOW:MEXT}$ ⑤	主设备检测时钟低电平延长时间(SMBUS)	-	10	-	10	-	10	ms

根据上表所示，有一些注意点如下所示：

1. $t_{HD:DAT}$ 是数据保持时间，从 SCL 的下降沿开始计算，适用于传输中的数据和 ACK，器件必须提供足够的保持时间，以弥补 SCL 下降沿的未定义区域。在 I2C 中，I2C_TIMING 寄存器中的 HDDAT 寄存器就是配置 $t_{HD:DAT}$ 的。
2. $t_{SU:DAT}$ 是数据建立时间，主要是产生一个采样点在 SCL 上升沿。在 I2C 中，I2C_TIMING.SUDAT 位就是配置 $t_{SU:DAT}$ 的
3. t_{LOW} 是 SCL 时钟的低电平时间，t_{HIGH} 是 SCL 时钟的高电平时间，这两个时钟主要是限制 SCL 的频率。在 I2C 中，I2C_BAUD.SCLLCNT 位控制 t_{LOW}，而 SCLHCNT 位控制 t_{HIGH}。
4. t_{SP} 表示了在快速模式下，协议规定了输入滤波器需要抑制小于 50ns 的毛刺。在 I2C 中，I2C_TIMING.SPKLEN 位就是来配置 t_{SP} 的。
5. $t_{LOW:SEXT}$ 是 SMBUS 规定的从机允许在一个信息中延长时钟周期的累计时间，从 START 到 STOP 的时间。$t_{LOW:MEXT}$ 是 SMBUS 规定的主机允许信息中的每个字节内延长时钟周期的累计时间，定义了从 START 到 ACK，ACK 到 ACK 或者 ACK 到 STOP 的时长。在 I2C 中，I2C_TIMEOUT.SEXTTO 位来配置 $t_{LOW:SEXT}$，MEXTTO 位来配置 $t_{LOW:MEXT}$。
6. 其他的时序都是 I2C 自动生成的，是不可配置的，具体如图 18-16 所示

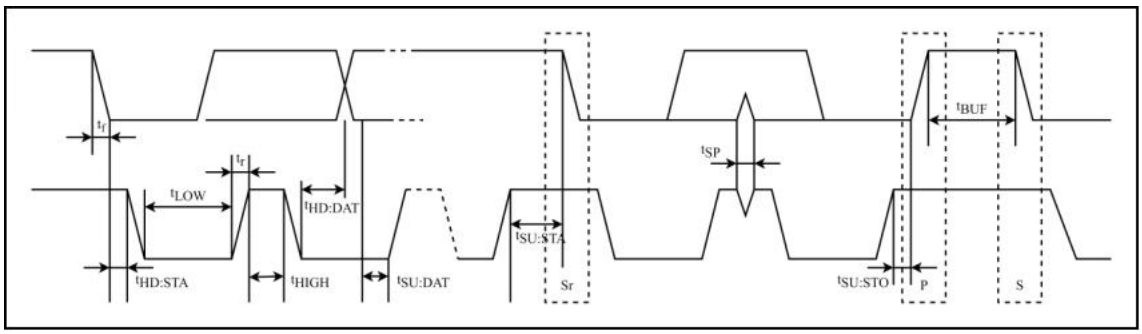


图 18-16　I2C 标准时序图

下面提供了如何合理配置 I2C 时序才能符合 I2C 规范的案例

表 18-3　50MHz 的配置示例表

参数	标准模式（SM）	快速模式（FM）	快速+模式（FM+）
SCL 频率	100kHz	400kHz	1000kHz
SCLLCNT	0x107	0x3C	0x14
t_{LOW}	(263+5+2) x 20ns = 5.4us	(60+5+2) x 20ns = 1.34us	(20+5+2) x 20ns = 540ns
SCLHCNT	0xDF	0x33	0x10
t_{HIGH}	(223+5+2) x 20ns = 4.6us	(51+5+2) x 20ns = 1.16us	(16+5+2) x 20ns = 460ns
HDDAT	0xF	0xF	0xF
$t_{HD:DAT}$	15 x 20ns = 300ns	15 x 20ns = 300ns	15 x 20ns = 300ns
SUDAT	0xD	0xD	0xD
$t_{SU:DAT}$	13 x 20ns = 260ns	13 x 20ns = 260ns	13 x 20ns = 260ns
SPIKEN	0x2	0x2	0x2
t_{SP}	2 x 20ns = 50ns	2 x 20ns = 50ns	2 x 20ns = 50ns

18.4.5　SMBUS 协议

18.4.5.1　SMBUS 介绍

系统管理总线（SMBUS）是一个双线接口，通过 SMBUS，各个设备之间可以相互通信，它是基于 I2C 的工作原理，芯片与 SMBUS 规范兼容（http://smbus.org）。SMBUS 既可以做主机也可以做从机，主设备是发出命令、生成时钟和中止传输的设备，从设备是接收或响应命令的设备。

对于任何给定的设备，有十一种可能的命令协议。一个设备可以使用十一个协议中的任何一个或者全部进行通信。协议包括快速命令、发送字节、写入字节、写入字、读取字节、读取字、进程调用、块读取、块写入和块写入块读取进程调用。这些协议由用户软件实现，有关这些协议的更多详细信息，请参阅 SMBUS 规范。

18.4.5.2　地址解析协议（ARP）

SMBUS 从机地址冲突可以通过为每个从机设备动态分配新的唯一地址来解决。为了提供一种

机制来隔离每个设备以进行地址分配，每个设备必须实现一个唯一的设备标识符（UDID），该标识符包含 128 位由软件来定义。

对于地址解析协议（ARP），可以通过配置 I2C_ENABLE.SMARPEN 位为 1 来启用 SMBUS 设备默认地址（0b1100 001）。ARP 命令应由用户软件执行。

此外，可以通过配置 I2C_ENABLE.SMHEN 位为 1 来支持主机通知协议，在这种情况下，SMBUS 默认的主机地址为 0b0001 000。

有关 SMBUS 地址解析协议的更多详细信息，请参阅 SMBUS 规范（http://smbus.org）。

18.4.5.3 SMBUS 警报

芯片支持 SMBUS 警报可选信号。只有从设备可以通过 SMBALERT 引脚向主机发送“想要通讯”的信号，引脚默认为上拉 1 的状态。当有设备拉低 SMBALERT 引脚时，主机收到信号后会立即处理并通过警报相应地址（0b0001 100）同时访问所有从设备，只有将 SMBALERT 拉低的设备才会响应警报地址。

当 SMBUS 作为从设备时，通过配置 I2C_ENABLE.SMALEN 位为 1，SMBALERT 引脚会被拉低，与此同时 SMBUS 此时只能响应警报地址。

当 SMBUS 作为主设备时，需要将 I2C_ENABLE.SMALEN 位配置成 1 启动警报功能，在收到 SMBALERT 为低的 ALDETI 中断，此时软件需要做出处理才能使从设备放开 SMBALERT 引脚。

18.4.5.4 数据包错误检测（PEC）

SMBUS 规范中引入了数据包错误检查机制，以提高可靠性和通信健壮性。数据包错误检查是通过在每次消息传输结束时附加数据包错误代码（PEC）来实现的，PEC 通过在所有消息字节上使用 $C(x)=x^8+x^2+x+1$ 的 CRC-8 多项式计算（包括地址和 R/W 位）。

SMBUS 内嵌硬件 PEC 计算器，当接收到的字节与硬件计算的 PEC 不匹配时，允许自动发送 NACK。

18.4.6 错误状态指示

18.4.6.1 上溢和下溢错误

在接收时，RXFIFO 最多能存储 16 个数据，如果接收的字节一直未被读走，在存满之后，再存在一个新数据时，新接收的字节会丢失，此时会触发 RXOFI 中断。若读取 RXFIFO 的数据量大于了 RXFIFO 存储的数据量，也就是在 RXFIFO 为空的时候，软件还读取了 RXFIFO，那么会触发 RXUFI 中断。

在发送时，TXFIFO 最多能存储 16 个数据，如果软件在 TXFIFO 为满的情况下还往 TXFIFO

写入一个数据，那么会触发 TXOFI 中断，写入的那个数会丢失。

18.4.6.2 I2C 中断

根据寻址协议的不同，可以分为 7bit 地址寻址和 10bit 地址寻址，图 18-17 展示的是在 7bit 地址下主发从收、主收从发和寻址失败的时序图，图 18-18 展示的是在 10bit 地址下主发从收、主收从发和寻址失败的时序图。

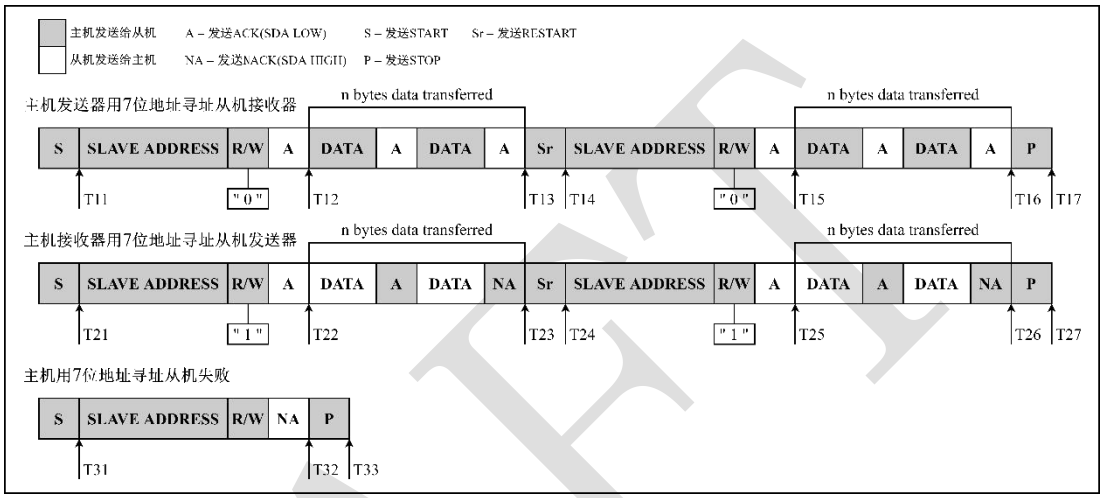


图 18-17　7bit 地址模式下发送接收

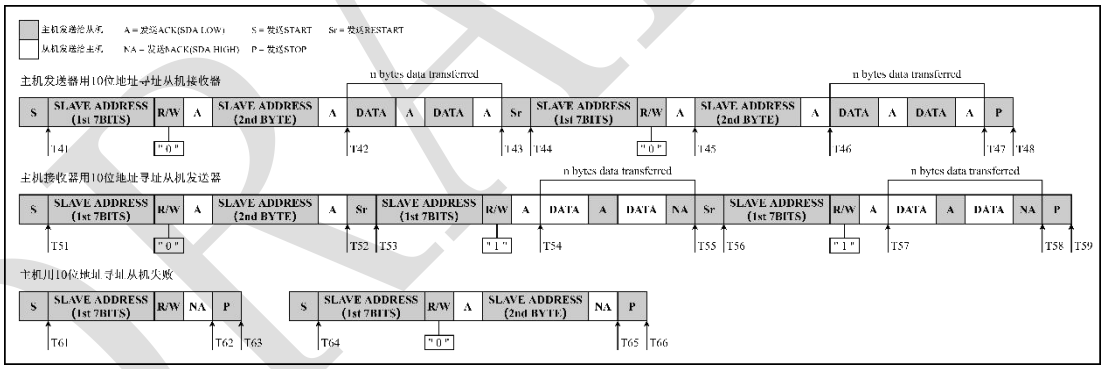


图 18-18　10bit 地址模式下发送接收

下表中列出了 I2C 的中断位含义以及它们对应上面两图的置位时间点。

表 18-4 中断位的描述

中断位	描述	置位时间点
MTXAI	作为主机完成了发送地址（无论收到 ACK 还是 NACK 都会起来）	T12，T15，T22，T25，T32，T42，T46，T52，T57
RXGCI	作为从机收到了 General Call 地址	-
PECRXI	收到 PEC 数据	-
SEXTOI	作为从机，START->STOP 的累计时间 Tlow:sext 超时了	-
MEXTOI	作为主机，START->ACK、ACK->ACK 和 ACK->STOP 的单次时间 Tlow:sext 超时	-
ALEDETI	作为主机检测是否收到了从机发出的 ALERT 信号	-
NACKI	作为主机检测收到了 NACK	T32，T62，T65

	作为从机检测收到了 NACK	T23，T26，T55，T58
MOHI	I2C 作为主机且当前没有数据可以发送和接收，也没有 STOP 位的产生也就是 SCL 为低，主机处于挂起状态	-
SPDETI	I2C 检测到了 STOP 位	T17，T27，T33，T48，T59，T63，T66
STDETI	I2C 检测到了 START 和 RESTART 位	T11，T14，T21，T24，T31，T41，T44，T51，T53，T56，T61，T64
RXADI	I2C 作为从机时接收到了完整的 10bit 地址或者 7bit 地址并且此时的命令是从机接收	T22，T25，T54，T57
TXADI	I2C 作为从机时接收到了完整的 10bit 地址或者 7bit 地址并且此时的命令是从机发送	T12，T15，T42，T45
MDEI	主机发送或接受完 I2C_CNT 个数据时起中断（发送会完全发完才起中断）	T13，T16，T23，T26，T43，T47，T55，T58
RXOFI	当 RXFIFO 已满并且新数据被接收写入 FIFO 中时，会起溢出错误中断。	-
TXEI	当 TXFIFO 中的数据量小于或者等于预设值（TXFTLR）时置位。	-
RXFI	当 RXFIFO 中的数据量大于或等于预设值（RXFTLR）时置位。	-

18.5 寄存器定义

18.5.1 寄存器列表

I2C 基地址：

I2C0(0x40000000) I2C1(0x40001000) I2C2(0x40002000)

偏移	实例地址	名称	默认值	描述
0x00	I2C 基地址+0x00	I2C_ENABLE	0x00000000	I2C 使能寄存器
0x04	I2C 基地址+0x04	I2C_CTRL	0x00000000	I2C 控制寄存器
0x08	I2C 基地址+0x08	I2C_BAUD	0x00000000	I2C 计数寄存器
0x10	I2C 基地址+0x10	I2C_FIFOCTRL	0x00000000	I2C FIFO 控制寄存器
0x14	I2C 基地址+0x14	I2C_TAR	0x00000000	I2C 主机目标地址寄存器
0x18	I2C 基地址+0x18	I2C_SAR	0x00000000	I2C 从机接收地址寄存器
0x20	I2C 基地址+0x20	I2C_PEC	0x00000000	I2C RX PEC 寄存器
0x24	I2C 基地址+0x24	I2C_TIMING	0x00180002	I2C 时间配置寄存器
0x28	I2C 基地址+0x28	I2C_TIMEOUT	0x07A2004F	I2C 时序配置寄存器
0x30	I2C 基地址+0x30	I2C_INTREN	0x00000000	I2C 中断使能寄存器
0x34	I2C 基地址+0x34	I2C_INTR	0x00000002	I2C 中断寄存器
0x38	I2C 基地址+0x38	I2C_STATUS	0x20200000	I2C 状态寄存器
0x40	I2C 基地址+0x40	I2C_TXDATA	0x00000000	I2C 数据写寄存器
0x44	I2C 基地址+0x44	I2C_RXDATA	0x00000000	I2C 数据读寄存器
0x48	I2C 基地址+0x48	I2C_RXADDR	0x00000000	I2C Debug 寄存器

18.5.2 寄存器描述

18.5.2.1 I2C 使能寄存器（I2C_ENABLE）

- **名称：I2C Enable Register**
- **偏移地址：0x00**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMTOEN	SMHEN	SMARPEN	SMALEN		GCEN	A10BEN	MSTEN		DMATXEN	DMARXEN	RLSCMD	TFRST	RFRST		I2CEN
R/W	R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	W	W	W		R/W
0x0	0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0	0x0	0x0		0x0

字段	说明
[15] SMTOEN	**SMBUS 超时计数使能位（SMBUS Timeout Counter Enable）** 0：不使能 1：使能
[14] SMHEN	**SMBUS Host 使能位（SMBUS Host Enable）** 该位启用/禁用从机对 Host 地址的应答 1：HOST 使能 0：HOST 不使能
[13] SMARPEN	**SMBUS ARP 使能位（SMBUS APR Enable）** 该位启用/禁用从机对 ARP 地址的应答 0：ARP 不使能 1：ARP 使能
[12] SMALEN	**SMBUS Alert 使能位（SMBUS ALERT ENABLE）** 该位启用/禁用主机对 Alert 信号的检测/从机拉低 Alert 引脚的动作 0：Alert 不使能 1：Alert 使能
[10] GCEN	**广播使能位（General Call Enable）** 该位启用/禁用从机对 General Call 地址的应答 0：General Call 地址不使能 1：General Call 地址使能
[9] A10BEN	**I2C 10bit 地址使能位（I2C 10BIT Address Enable）** 0：7bit 模式 1：10bit 模式
[8] MSTEN	**I2C 主机使能位（I2C Master Enable）** 0：I2C 作为从机 1：I2C 作为主机
[6] DMATXEN	**I2C 发送 DMA 使能位（TX DMA Enable）** 该位启用/禁用发送 FIFO DMA 通道 0：禁用发送 DMA 通道 1：使能发送 DMA 通道
[5] DMARXEN	**I2C 接收 DMA 使能位（RX DMA Enable）** 该位启用/禁用接收 FIFO DMA 通道 0：禁用接收 DMA 通道 1：使能接收 DMA 通道
[4] RLSCMD	**I2C 命令释放（Release Command）** 写入 CTRL 寄存器之后需将该位写 1 才能启动 I2C 传输，该寄存器写 0 无效

	0：无动作 1：释放命令
[3] TFRST	**TXFIFO 复位（TXFIFO RESET）** TXFIFO 复位信号，自清 0 0：TXFIFO 正常 1：TXFIFO 复位
[2] RFRST	**RXFIFO 复位（RXFIFO RESET）** RXFIFO 复位信号，自清 0 0：RXFIFO 正常 1：RXFIFO 复位
[0] I2CEN	**I2C 使能位（I2C ENABLE）** 使能时，将不能对控制寄存器进行配置。 0：I2C 不使能 1：I2C 使能

18.5.2.2 I2C 控制寄存器（I2C_CTRL）

- **名称：I2C Control Register**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AUTO END		PECB YTE	NACK	START	STOP	DIREC T	I2CCNT							
	R/W		R/W	R/WA C	R/WA C	R/WA C	R/W	R/W							
	0x0		0x0	0x0	0x0	0x0	0x0	0x0							

字段	说明
[14] AUTOEND	**I2C 自动结束模式（I2C Automatic End Mode）** 此位由软件置 1 和清 0，在从模式下该位不起作用 0：软件结束模式：当 I2CCNT 数据传输完成时，MDEI 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束 1：自动结束模式：当 I2CCNT 数据传输完成时，将自动发送停止位。
[12] PECBYTE	**SMBUS PEC 模式（SMBUS PECBYTE Mode）** 主机：发送 PEC 字节，PECBYTE 发送后自动发送停止位 0：STOP 之前不发送 PEC 字节 1：STOP 之前发送 PEC 字节 Note：主机发送 PEC 该位置 1，此时的 I2CCNT 只计算 DATA 的数量；主机如果要接收 PEC 该位置 0，此时的 I2CCNT 计算 PEC 和 DATA 的数据量总和，也就是 DATA 的数据量+1 从机：在 I2CCNT 个数据之后进入到 PEC 模式，发送或者接收 PEC 0：从机不发送或者不接收 PEC 1：从机发送或者接收 PEC
[11] NACK	**I2C NACK 生成模式（I2C Nack Generation Mode）** 此位由软件置 1，并可在接收到停止位时、或者 I2CEN=0 时由硬件清零 在主模式下该位不起作用，向该位写入 0 不起作用 0：在当前接收的字节后发送 ACK 1：在当前接收的字节后发送 NACK
[10] START	**I2C START 生成（I2C START Configuration）** 此位由软件置 1、并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者 I2CEN=0 时由硬件清 0，此位置 1 时 Release Bus 可以启动带有起始位/重复起始位的序列 在从模式下该位不起作用，向该位写入 0 不起作用

	0：不生成起始位 1：生成重复起始/起始位。 -如果 I2C 已处于主模式下且 AUTOEND=0，则将该位置 1 会在 I2CCNT 传输结束后生成重复起始位 -否则，将该位置 1 会在总线释放后立即生成起始位
[9] STOP	**I2C STOP 生成（I2C STOP Configuration）** 此位由软件置 1，并可在检测到停止位时或 I2CEN=0 时由硬件清零 在从模式下该位不起作用，向该位写入 0 不起作用 0：不生成停止位。 1：在当前字节传输完成后生成停止位。
[8] DIRECT	**I2C 传输方向配置（I2C Direct Configuration）** 此位控制是执行读操作还是写操作 此位由软件置 1 和清 0，在从模式下该位不起作用 0：主机 TX 1：主机 RX
[7:0] I2CCNT	**I2C 发送/接收数据量（I2C TX/RX Data Count）** I2C 数据计数寄存器，写入当前要发送或者接收的数据量

18.5.2.3 I2C 计数寄存器（I2C_BAUD）

- **名称：I2C Baud Rate Registers**
- **偏移地址：0x08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						SCLHCNT									
						R/W									
						0x0									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						SCLLCNT									
						R/W									
						0x0									

字段	说明
[25:16] SCLHCNT	**SCL 高电平计数（SCL High Count）** SCL 高电平持续的时长（系统时钟周期为单位），实际值是配置值+1（原始配置）+SPKLEN+1（尖峰抑制）+3（时钟同步）
[9:0] SCLLCNT	**SCL 低电平计数（SCL Low Count）** SCL 低电平持续的时长（系统时钟周期为单位），实际值是配置值+1（原始配置）+SPKLEN+1（尖峰抑制）+3（时钟同步）

18.5.2.4 I2C FIFO 控制寄存器（I2C_FIFOCTRL）

- 名称：I2C FIFO Control Registers
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RXFTLR				TXFTLR			
								R/W				R/W			
								0x0				0x0			

字段	说明
[7:4] RXFTLR	**RXFIFO 满中断阈值（RXFIFO Full Threshold）** 0x0：FIFO 中有 1 个或 1 个以上数据就会触发 RXFIFO 满中断 0x1：FIFO 中有 2 个或 2 个以上数据就会触发 RXFIFO 满中断 ..... 0xe：FIFO 中有 15 个或 15 个以上数据就会触发 RXFIFO 满中断 0xf：FIFO 中有 16 个数据就会触发 RXFIFO 满中断
[3:0] TXFTLR	**TXFIFO 空中断阈值（TXFIFO Empty Threshold）** 0x0：FIFO 中没有数据就会触发 TXFIFO 空中断 0x1：FIFO 中有 1 个或 1 个以下数据就会触发 TXFIFO 空中断 ..... 0xe：FIFO 中有 14 个或 14 个以下数据就会触发 TXFIFO 空中断 0xf：FIFO 中有 15 个或 15 个以下数据就会触发 TXFIFO 空中断

18.5.2.5 I2C 主机目标地址寄存器（I2C_TAR）

- 名称：I2C Target Address Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TAR									
						R/W									
						0x0									

字段	说明
[9:0] TAR	**主机目标地址寄存器（Master Target Address Register）** 作为主机时要发送的地址

18.5.2.6 I2C 从机接收地址寄存器（I2C_SAR）

- 名称：I2C Receive Address Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						SAR									
						R/W									
						0x0									

字段	说明
[9:0] SAR	**从机接收地址寄存器（Slave Receive Address Register ）** 作为从机时和主机通信的设备地址

18.5.2.7 I2C RX PEC 寄存器（I2C_PEC）

- 名称：I2C RX PEC Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDPEC								RXPEC							
R								R							
0x0								0x0							

字段	说明
[15:8] IDPEC	**SMBUS 计算 PEC 数据寄存器（SMBUS ID PEC Data Register ）** 计算得到的 PEC 值存放在该寄存器，当 PECRXI 中断起来时 RXPEC 与 IDPEC 不一致 关闭 I2C 使能将会清除数据
[7:0] RXPEC	**SMBUS 接收 PEC 数据寄存器（SMBUS RX PEC Data Register ）** 接收得到的 PEC 值存放在该寄存器，当 PECRXI 中断起来时 RXPEC 与 IDPEC 不一致 关闭 I2C 使能将会清除数据

18.5.2.8 I2C 时间配置寄存器（I2C_TIMING）

- 名称：I2C Timing Register
- 偏移地址：0x24
- 默认值：0x00180002
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPKLEN								SLVSUDAT							
R/W								R/W							
0x0								0x18							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUDAT								HDDAT							
R/W								R/W							
0x0								0x2							

字段	说明
[31:24] SPKLEN	**尖峰抑制限制寄存器（Spike Suppression Limit Register）** 该寄存器用于配置尖峰抑制逻辑滤除的最长尖峰的持续时间，对应 I2C 的时序为 TSP，实际值是配置值
[23:16] SLVSUDAT	**从机数据建立时间配置寄存器（Slave Data Setup Time Configuration Register）** 从机作为 TX 时，数据建立时间配置，单位是系统时钟周期，用于生成 TSU:DAT 时序，实际值是配置值（原始配置）+SPKLEN（尖峰抑制）+3（时钟同步）
[15:8] SUDAT	**主从数据建立时间配置寄存器（Data Setup Time Configuration Register）** 数据建立时间配置，也就是主从作为 RX 时数据采样的延迟，单位是系统时钟周期，用于生成 TSU:DAT 时序，实际值是配置值（原始配置）+1+SPKLEN（尖峰抑制）+3（时钟同步）
[7:0] HDDAT	**主从数据保持时间配置寄存器（Data Hold Time Configuration Register）** 数据保持时间配置，也就是主从作为 TX 时数据准备的延迟，单位是系统时钟周期，用于生成 THD:DAT 时序，实际值是配置值（原始配置）+1+SPKLEN（尖峰抑制）+3（时钟同步）

18.5.2.9 I2C 时序配置寄存器（I2C_TIMEOUT）

- 名称：I2C Timeout Register
- 偏移地址：0x28
- 默认值：0x07A2004F
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEXTTO															
R/W															
0x7a2															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEXTTO															
R/W															
0x4f															

字段	说明
[31:16] SEXTTO	**从机 Tsext 时序配置寄存器（Slave Tsext Timing Configuration Register）** 作为从机时，用于生成 TLOW:SEXT 时序，实际值=（配置值+1）*256
[15:0] MEXTTO	**主机 Tmext 时序配置寄存器（Master Tmext Timing Configuration Register）** 作为主机时，用于生成 TLOW:MEXT 时序，实际值=（配置值+1）*256

18.5.2.10 I2C 中断使能寄存器（I2C_INTREN）

- 名称：I2C Interrupt Enable Registers
- 偏移地址：0x30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									MOHIE	SWTXIE	MTXAIE	RXGCIE	PECRXIE	SEXTOIE	MEXTOIE
									R/W	R/W	R/W	R/W	R/W	R/W	R/W
									0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALDETIE	NACKIE	RSDETIE	SPDETIE	STDETIE	RXADIE	TXADIE	MDEIE	BUSEIE	ARBFIE	TXUFIE	TXOFIE	RXUFIE	RXOFIE	TXEIE	RXFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[22] MOHIE	**MOHI 中断使能（Master On Hold Interrupt Enable）** 0：不使能 1：使能
[21] SWTXIE	**SWTXI 中断使能（Slave Wait TX Data Interrupt Enable）** 0：不使能 1：使能
[20] MTXAIE	**MTXAI 中断使能（Master TX Adderss Done Interrupt Enable）** 0：不使能 1：使能
[19] RXGCIE	**RXGCI 中断使能（Slave RX General Call Interrupt Enable）** 0：不使能 1：使能
[18] PECRXIE	**PECRXI 中断使能（Master or Slave RX PEC Error Interrupt Enable）** 0：不使能 1：使能
[17] SEXTOIE	**SEXTOI 中断使能（Slave Tsext Timeout Interrupt Enable）** 0：不使能 1：使能
[16] MEXTOIE	**MEXTOI 中断使能（Master Tmext Timeout Interrupt Enable）** 0：不使能 1：使能
[15] ALDETIE	**ALDETI 中断使能（Master Detect Alert Signal Interrupt Enable）** 0：不使能 1：使能
[14] NACKIE	**NACKI 中断使能（Master or Slave RX NACK Interrupt Enable）** 0：不使能 1：使能
[13] RSDETIE	**RSDETI 中断使能（Master or Slave Detect RESTART Interrupt Enable）** 0：不使能 1：使能
[12] SPDETIE	**SPDETI 中断使能（Master or Slave Detect STOP Interrupt Enable）** 0：不使能 1：使能
[11] STDETIE	**STDETI 中断使能（Master or Slave Detect START Interrupt Enable）** 0：不使能 1：使能
[10] RXADIE	**RXADI 中断使能（Slave RX Address And Command is SLAVE RX Interrupt Enable）** 0：不使能 1：使能
[9]	**TXADI 中断使能（Slave RX Address And Command is SLAVE TX Interrupt Enable）**

TXADIE	0：不使能 1：使能
[8] MDEIE	**MDEI 中断使能（Master TX/RX Done Interrupt Enable）** 0：不使能 1：使能
[7] BUSEIE	**BUSEI 中断使能（Bus Error Interrupt Enable）** 0：不使能 1：使能
[6] ARBFIE	**ARBFI 中断使能（Arbitration Fail Interrupt Enable）** 0：不使能 1：使能
[5] TXUFIE	**TXUFI 中断使能（TXFIFO Underflow Interrupt Enable）** 0：不使能 1：使能
[4] TXOFIE	**TXOFI 中断使能（TXFIFO Overflow Interrupt Enable）** 0：不使能 1：使能
[3] RXUFIE	**RXUFI 中断使能（RXFIFO Underflow Interrupt Enable）** 0：不使能 1：使能
[2] RXOFIE	**RXOFI 中断使能（RXFIFO Overflow Interrupt Enable）** 0：不使能 1：使能
[1] TXEIE	**TXEI 中断使能（TXFIFO Empty Interrupt Enable）** 0：不使能 1：使能
[0] RXFIE	**RXFI 中断使能（RXFIFO Full Interrupt Enable）** 0：不使能 1：使能

18.5.2.11 I2C 中断寄存器（I2C_INTR）

- **名称：I2C Interrupt Registers**
- **偏移地址：0x34**
- **默认值：0x00000002**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCNT									MOHI	SWTXI	MTXAI	RXGCI	PECRXI	SEXTOI	MEXTOI
R									R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
0x0									0x0	0x0	0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALDETI	NACKI	RSDETI	SPETI	STDETI	RXADI	TXADI	MDEI	BUSEI	ARBFI	TXUFI	TXOFI	RXUFI	RXOFI	TXEI	RXFI
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R	R
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0

字段	说明
[31:24] RCNT	**主机/从机当前传输剩余数据（Master/Slave Remain Counter）** 主机/从机当前会话中剩余的数据个数 Note：当总线上有数据传输时候，该值仅反映当前剩余数据
[22] MOHI	**主机挂起中断（Master On Hold Interrupt ）** I2C 作为主机且当前没有数据可以发送和接收，也没有 STOP 位的产生也就是 SCL 为低，主机处于挂起状态 0：主机处于其他状态 1：主机处于挂起状态 Note：EN 关闭、主机发送或者接收数据、检测到 STOP 位都会清除 MOHI 中断

[21] SWTXI	**从机等待 TX 数据中断（Slave Wait TX Data Interrupt）** 作为从机，在收到主机发送的主机读命令之后，如果 TXFIFO 中没有数据，会处于等待软件写 TX 数据的状态，此状态会拉住总线 0：无 1：从机处于等待 TX 数据状态 Note：对该位写 1 将清除 SWTXI 中断
[20] MTXAI	**主机发送地址完成中断（Master TX Adderss Done Interrupt ）** 作为主机完成了发送地址（无论收到 ACK 还是 NACK 都会起来） 0：无 1：发送地址完毕 Note：对该位写 1 将清除 MTXAI 中断
[19] RXGCI	**从机收到广播中断（Slave RX General Call Interrupt ）** 作为从机收到了 General Call 地址 0：无 1：收到了 General Call 地址 Note：对该位写 1 将清除 RXGCI 中断
[18] PECRXI	**主从收到 PEC 错误中断（Master or Slave RX PEC Error Interrupt ）** 收到 PEC 数据且 PEC 数据对比错误 0：未收到 PEC 数据或 PEC 数据正确 1：收到 PEC 数据且 PEC 数据错误 Note：对该位写 1 将清除 PECRXI 中断
[17] SEXTOI	**从机 Tsext 超时中断（Slave Tsext Timeout Interrupt ）** 作为从机，START->STOP 的单次时间 Tlow:sext 超时 0：未超时 1：超时 Note：对该位写 1 将清除 SEXTOI
[16] MEXTOI	**主机 Tmext 超时中断（Master Tmext Timeout Interrupt ）** 作为主机，START->ACK、ACK->ACK 和 ACK->STOP 的单次时间 Tlow:mext 超时 0：未超时 1：超时 Note：对该位写 1 将清除 MEXTOI 中断
[15] ALDETI	**主从检测 Alert 中断（Master or Slave Detect Alert Signal Interrupt ）** 作为主机检测是否收到了从机发出的 ALERT 信号 0：未收到 ALERT 信号 1：收到 ALERT 信号 Note：对该位写 1 将清除 ALDETI 中断
[14] NACKI	**主从收到 NACK 中断（Master or Slave RX NACK Interrupt ）** I2C 检测收到了 NACK 0：未收到 NACK 1：收到 NACK Note：对该位写 1 将清除 NACKI 中断
[13] RSDETI	**主从检测 RESTART 中断（Master or Slave Detect RESTART Interrupt ）** I2C 检测到了 RESTART 位 0：未检测到 RESTART 位 1：检测到了 RESTART 位 Note：对该位写 1 将清除 RSDETI 中断
[12] SPETI	**主从检测 STOP 中断（Master or Slave Detect STOP Interrupt ）** I2C 检测到了 STOP 位 0：未检测到 STOP 位 1：检测到了 STOP 位 Note：对该位写 1 将清除 SPETI 中断
[11] STDETI	**主从检测 START 中断（Master or Slave Detect START Interrupt ）** I2C 检测到了 START 位 0：未检测到 START 位 1：检测到了 START 位 Note：对该位写 1 将清除 STDETI 中断
[10] RXADI	**从机收到地址且命令是从机 RX（Slave RX Address And Command is SLAVE RX Interrupt ）** I2C 作为从机时接收到了完整的 10bit 地址或者 7bit 地址并且此时的命令是从机接收 0：地址未接收完毕且命令是从机接收

	1：地址接收完毕且命令是从机接收 Note：对该位写 1 将清除 RXADI 中断
[9] TXADI	**从机收到地址且命令是从机 TX（Slave RX Address And Command is SLAVE TX Interrupt）** I2C 作为从机时接收到了完整的 10bit 地址或者 7bit 地址并且此时的命令是从机发送 0：地址未接收完毕且命令是从机发送 1：地址接收完毕且命令是从机发送 Note：对该位写 1 将清除 TXADI 中断
[8] MDEI	**主机发送接收完成中断（Master TX/RX Done Interrupt）** 主机发送或接受完 CNT 个数据时起中断 0：未发送或者接收完毕 1：发送或者接收完毕 Note：对该位写 1 将清除 MDEI 中断
[7] BUSEI	**总线错误中断（Bus Error Interrupt）** 当检测到错误的起始位或停止位且外设在参与传输时，会起总线错误中断。从机的地址阶段不会起总线错误中断。作为主机看到总线错误中断后，需要将 I2CEN 置 0 再置 1 以重置 I2C。 0：未发生总线错误 1：发生总线错误 Note：对该位写 1 将清除 BUSEI 中断
[6] ARBFI	**仲裁失败中断（Arbitration Fail Interrupt）** 当主机或从机发生仲裁失败时，会起仲裁失败中断。作为主机看到仲裁失败中断后，需要将 I2CEN 置 0 再置 1 以重置 I2C。 0：未发生仲裁失败 1：发生仲裁失败 Note：对该位写 1 将清除 ARBFI 中断
[5] TXUFI	**TXFIFO 下溢错误位（TXFIFO Underflow Interrupt）** 当 TXFIFO 已空并且但是还是有从 FIFO 取数的操作，会起下溢错误中断。 0：无溢出错误 1：溢出错误 Note：对该位写 1 将清除 TXUFI 中断。
[4] TXOFI	**TXFIFO 溢出错误位（TXFIFO Overflow Interrupt）** 当 TXFIFO 已满并且新数据被写入 FIFO 中时，会起溢出错误中断。此时 FIFO 中的数据将保留，而写入的新数据将丢失。 0：无溢出错误 1：溢出错误 Note：对该位写 1 将清除 TXOFI 中断。
[3] RXUFI	**RXFIFO 下溢错误位（RXFIFO Underflow Interrupt）** 当 RXFIFO 已空并且但是还是有从 FIFO 取数的操作，会起下溢错误中断。 0：无溢出错误 1：溢出错误 Note：对该位写 1 将清除 RXUFI 中断。
[2] RXOFI	**RXFIFO 溢出中断（RXFIFO Overflow Interrupt）** RXFIFO 溢出错误位。当 RXFIFO 已满并且新数据被接收写入 FIFO 中时，会起溢出错误中断。此时 FIFO 中的数据将保留，而接收的新数据将丢失。 0：无溢出错误 1：溢出错误 Note：对该位写 1 将清除 RXOFI 中断。
[1] TXEI	**TXFIFO 空中断（TXFIFO Empty Interrupt ）** 与 TXFTLR 寄存器有关，当 TXFIFO 中的数据量小于或者等于预设值（TXFTLR）时置位。 0：TXFIFO 数据量大于预设值 1：TXFIFO 数据量小于等于预设值 Note：当 TXFIFO 数据量大于预设值时 TXEI 中断自动清 0
[0] RXFI	**RXFIFO 满中断（RXFIFO Full Interrupt ）** 与 RXFTLR 寄存器有关，当 RXFIFO 中的数据量大于预设值（RXFTLR）时置位。 0：RXFIFO 数据量小于等于预设值 1：RXFIFO 数据量大于预设值 Note：当 RXFIFO 数据量小于等于预设值时 RXFI 中断自动清 0

18.5.2.12 I2C 状态寄存器（I2C_STATUS）

- 名称：I2C Status Register
- 偏移地址：0x38
- 默认值：0x20200000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFF	RFE	RXFLR						TFF	TFE	TXFLR				
	R	R	R						R	R	R				
	0x0	0x1	0x0						0x0	0x1	0x0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														BUSBSY	FSMBSY
														R	R
														0x0	0x0

字段	说明
[30] RFF	**接收 FIFO 满标志（RXFIFO Full Flag）** 0：接收 FIFO 未满 1：接收 FIFO 已满 Note：当 RX FIFO 不再满时，该位被清除。
[29] RFE	**接收 FIFO 空标志（RXFIFO Empty Flag）** 0：接收 FIFO 不为空 1：接收 FIFO 为空 Note：当 RX FIFO 不为空时，该位被清除。
[28:24] RXFLR	**RXFIFO 实时剩余数据量（RXFIFO Level Register）**
[22] TFF	**发送 FIFO 满标志（TXFIFO Full Flag）** 0：发送 FIFO 未满 1：发送 FIFO 为满 Note：当 TX FIFO 不再满时，该位被清除。
[21] TFE	**发送 FIFO 空标志（TXFIFO Empty Flag）** 0：发送 FIFO 不为空 1：发送 FIFO 为空 Note：当 TX FIFO 不为空时，该位被清除。
[20:16] TXFLR	**TXFIFO 实时剩余数据量（TXFIFO Level Register）**
[1] BUSBSY	**总线 busy 状态（Bus Busy Status）** 0：无 1：总线 busy
[0] FSMBSY	**主从状态机 busy 状态（Master or Slave Busy Status）** 0：无 1：状态机 busy

18.5.2.13 I2C 数据写寄存器（I2C_TXDATA）

- 名称：I2C TXFIFO Data Registers
- 偏移地址：0x40
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TXDATA							
								W							
								0x0							

字段	说明
[7:0] TXDATA	**I2C 数据写寄存器（I2C Data Write Register）** 往 FIFO 写要发送的数据

18.5.2.14 I2C 数据读寄存器（I2C_RXDATA）

- 名称：I2C RXFIFO Data Registers
- 偏移地址：0x44
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RXDATA							
								R							
								0x0							

字段	说明
[7:0] RXDATA	**I2C 数据读寄存器（I2C Data Read Register）** 读取从外界收到的并已经存到 FIFO 中的数据

18.5.2.15 I2C Debug 寄存器（I2C_RXADDR）

- 名称：I2C RX ADDR Register
- 偏移地址：0x48
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ADDR							DIR
								R							R
								0x0							0x0

字段	说明
[7:1] ADDR	**从机地址匹配代码（Slave Address Match Code）** 从机在地址匹配时更新上一笔传输的匹配地址 Note：在 10bit 地址情况下、提供 10bit 地址头字节
[0] DIR	**从机传输方向（Slave Transfer Direction）** 从机在地址匹配时更新上一笔传输的传输方向 0：写传输 1：读传输

19 通用异步收发器（UART）

19.1 简介

通用异步收发传输器（Universal Asynchronous Receiver/Transmitter），通常称作异步收发传输器，支持全双工双线通信和半双工单线通信。

19.2 主要特性

UART 主要具有以下特性：

- 全双工异步通信
- 支持 8 倍和 16 倍过采样
- 支持 2 个用于收发数据的内部 FIFO，同时也支持非 FIFO 模式
- 支持串行数据字长度可编程（5 位~8 位）
- 支持可编程的奇偶检验位和 STOP 位
- 支持拓展位功能
- 支持 RS-485 收发器的硬件流控制（DE 和 RE）
- 支持通信控制/错误检测标志
- 支持单线模式
- 支持中断和轮询操作
- 支持 DMA 传输

19.3 结构框图

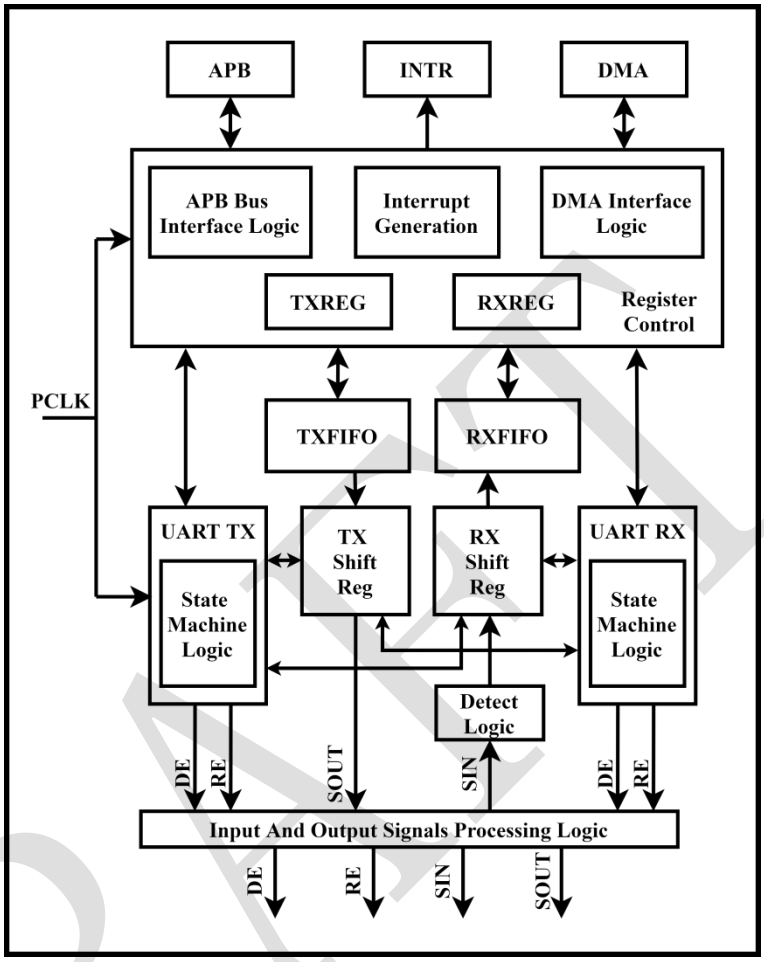


图 19-1 UART 结构框图

19.4 功能描述

19.4.1 UART 信号描述

UART 属于双向通信，同时支持 RS485，因此一共有 4 个与外界交互的引脚，具体如表 19-1 所示。

表 19-1 UART 信号列表

信号	名称	详细描述
TX	串行数据输出	TX 引脚极性默认为高电平无效，是串行数据输出的引脚。可以通过寄存器位 TXPO 配置默认无效电平；还可以通过寄存器位 TRSW 变换 TX 引脚的属性使其可以变为串行数据输入引脚。
RX	串行数据输入	RX 引脚极性默认为高电平无效，是串行数据输入的引脚。可以通过寄存器位 RXPO 配置默认无效电平；还可以通过寄存器位 TRSW 变换 RX 引脚的属性使其可以变为串行数据输出引脚。

DE	驱动器输出使能	当 DE 引脚为逻辑高时，UART 发送使能；当 DE 引脚为逻辑低时，UART 接收使能。
RE	驱动器输入使能	RE 是 DE 的差分信号

19.4.2 UART 协议分析

由于 UART 与设备间的通信是异步的，在串行数据中要加入两个数据位（start 和 stop）来标志通信的开始和结束。通过这两个数据位可以让两个设备间进行同步。

串行数据的格式如图 19-2 所示：

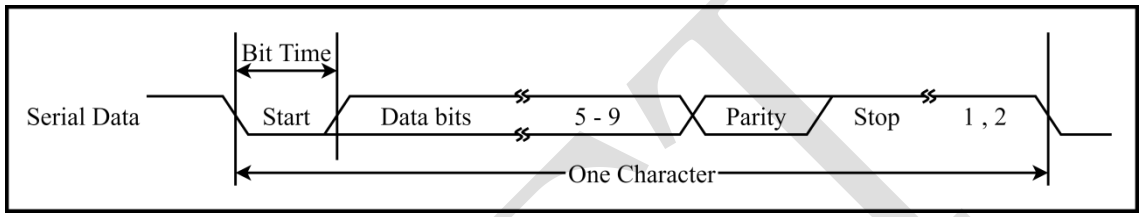


图 19-2 UART 串行数据格式

UART 传输的字符中可以加入校验位，校验位处于最后一位数据位之后，STOP 位之前。主要用来对 UART 接收的数据进行简单的错误检查。UART 可以通过配置寄存器来控制串行字符特征。START 位发出之后，发送可选的 5~9 位数据位，可以采用 LSB 传输也可以采用 MSB 传输，紧接着是可选的校验位和 STOP 位（可以为 1 位或者 2 位）。

在 UART 传输过程中，每一位的传输时间都是一样的，每一位的传输时间称为位周期，每个位周期都是通过波特率生成的。为了保证线上传输的稳定性，接收端检测到 START 位后，在位周期的中点处开始采集输入的数据。因为每个数据位传输时间的波特率是已知的，所以不难计算每一位数据采集的中点。此外需要对输入进行去抖动，这样短暂的不稳定信号可以通过去抖动滤掉，这种采样方式也可以避免检测到错误的 START 位。如图 19-3 是 UART 的线上采样图。

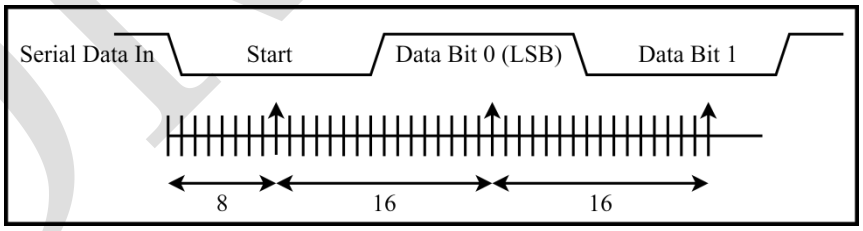


图 19-3 UART 接收数据采样

19.4.3 UART 字符说明

可通过配置 UART_CR1.LEN 位来确定 UART 帧字符长度

- 5 位字符长度：LEN[1:0]=00
- 6 位字符长度：LEN[1:0]=01
- 7 位字符长度：LEN[1:0]=10
- 8 位字符长度：LEN[1:0]=11

在默认情况下，TX 和 RX 信号在起始位期间处于低电平状态，在停止位期间处于高电平状态，通过极性配置（UART_CR0.TPOL 位和 UART_CR0.RPOL 位）控制，可以单独针对每个信号进行取反。

停止字符也可以通过配置 UART_CR1.STP 位来确定停止位的个数

- 一个停止字符长度：STP=0
- 两个停止字符长度：STP=1

奇偶检验位通过配置 UART_CR1.PEN、UART_CR1.PSEL 和 UART_CR1.SPE 来确定，具体如下

a) 无校验位：PEN 配置为 0，PSEL 配置为任意值，SPE 配置为任意值
b) 发送或者检测奇校验：PEN 配置为 1，PSEL 配置为 1，SPE 配置为 0
c) 发送或者检测偶校验：PEN 配置为 1，PSEL 配置为 0，SPE 配置为 0
d) 发送或者检测强制校验 0：PEN 配置为 1，PSEL 配置为 0，SPE 配置为 1
e) 发送或者检测强制校验 1：PEN 配置为 1，PSEL 配置为 1，SPE 配置为 1

19.4.4 UART FIFO 和阈值

UART 内部有一个发送 FIFO（TXFIFO）和一个接收 FIFO（RXFIFO）。通过配置 UART_CR0.NFE 位为 0，使得 UART 工作在 FIFO 模式下，否则 UART 将工作在非 FIFO 模式下。TXFIFO 中最大数据字长度为 9 位，RXFIFO 中最大数据长度为 11 位，因为除了接收数据的 9 位，此外还有一位指示奇偶检验位是否异常和一位指示停止位是否异常的状态位。

可以配置触发 TXFIFO 空中断和 RXFIFO 满中断的阈值，分别由 FIFOCTRL.TXFT 和 FIFOCTRL.RXFT 来控制，具体如下：

1. TXFIFO 中的数据数量小于等于 FIFOCTRL.TXFT 位的配置值，TXEI 中断会被置起。
2. TXFIFO 中的数据数量大于 FIFOCTRL.TXFT 位的配置值，TXEI 中断将会被清除。
3. RXFIFO 中的数据数量小于 FIFOCTRL.RXFT 位的配置值，RXFI 中断将会被清除
4. RXFIFO 中的数据数量大于等于 FIFOCTRL.RXFT 位的配置值，RXFI 中断将会被置起

注意：

1) TXFIFO 还有溢出错误的 TXOFI 中断，当 TXFIFO 已满且此时还有新的数据即将写入 FIFO，TXOFI 会被置起，此时 TXFIFO 中的数据将被保留，而即将写入的新数据会被丢弃。
2) RXFIFO 还有溢出错误的 RXOFI 中断，当 RXFIFO 已满且此时还有新的数据即将写入 FIFO，RXOFI 会被置起，此时 RXFIFO 中的数据将被保留，而即将写入的新数据会被丢弃。

19.4.5 UART 波特率

接收器和发送器的波特率均由 UART_BAUD 寄存器来控制，具体公式如下所示：

$$\text{TX/RX baud} = \frac{\text{PCLK}}{\text{BAUD}}$$

其中 PCLK 代表的是 UART 的系统时钟，BAUD 代表的是 UART_BAUD.BAUD 位配置的值。

此外 UART 还可以通过配置 UART_CR0.OVER8 位来决定 UART 的采样类型，OVER8=0 时是 16 倍过采样，此时 BAUD 的配置值必须大于等于 0x10（十进制的 16）；OVER8=1 时是 8 倍过采样，此时 BAUD 的配置值必须大于等于 0x8（十进制的 8）。关于过采样 8 倍和 16 倍的区别如下图所示：

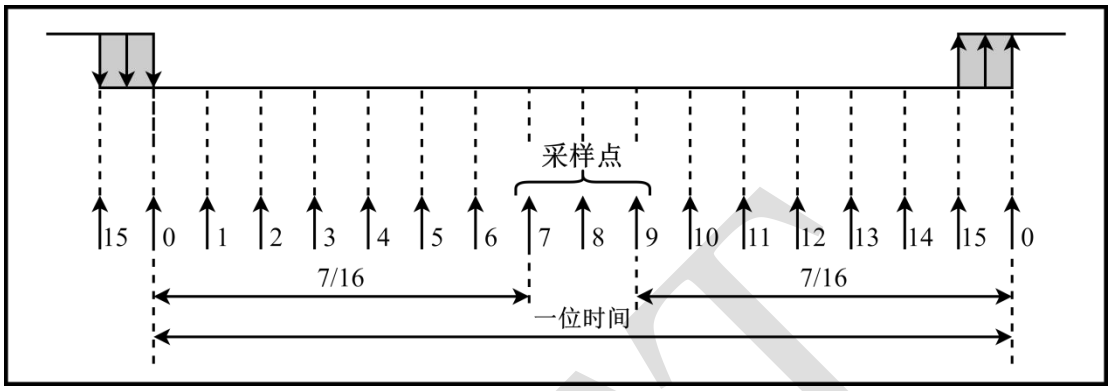


图 19-4　UART 过采样 16 倍采样图

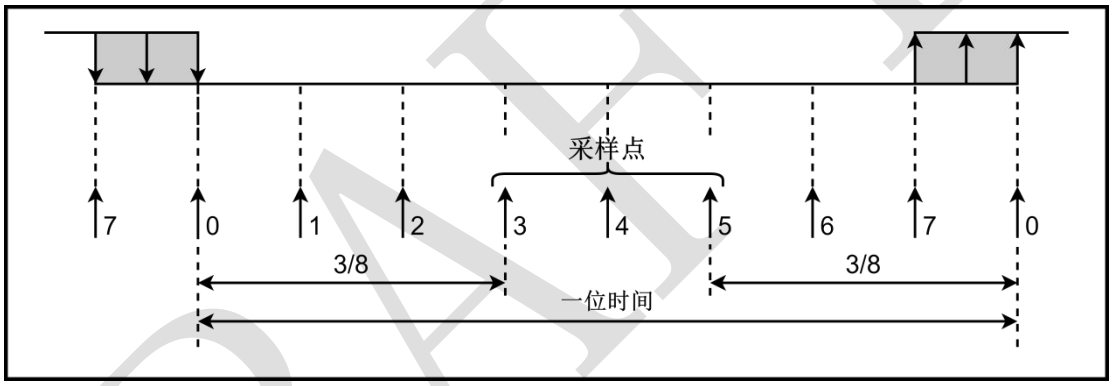


图 19-5　UART 过采样 8 倍采样图

19.4.6　UART 发送器

要使能发送器功能，必须将发送使能位（UART_CR0.TE）置 1。

要发送数据，需要遵循以下步骤：

1. 配置 UART_CR1.LEN 位以确定发送字符长度
2. 配置 UART_BAUD.BAUD 位以确定所需波特率
3. 配置 UART_CR1.STP 位以确定停止位数量
4. 配置 UART_CR1.PEN 位、UART_CR1.PSEL 位和 UART_CR1.SPE 位以确定奇偶检验位
5. 配置 UART_CR0.UE 位以使能 UART 模块
6. 配置 UART_CR0.TE 位以使能 UART 发送器
7. 在 UART_TDR 寄存器中写入要发送的数据
   a) 禁用 FIFO 模式（UART_CR0.NFE=0）时，往 UART_TDR 寄存器写 1 个数后 TXEI 会被清掉，知道写入的数据发送完成了才会重新置 1。
   b) 使能 FIFO 模式（UART_CR0.NFE=1）时，往 UART_TDR 寄存器写 1 个数 TXFIFO 中会多一个数据（UART_STATUS.TFL），在 UART_STATUS.TFF 位未被硬件置 1 前，软件都可以往 UART_TDR 寄存器写要发送的数据。

8. 将最后一个数据写入 UART_TDR 寄存器后，需要等待 UART_INT.TDIF 位被硬件 1。
   a) 禁用 FIFO 模式时，这表示最后一个帧数据发送已完成。
   b) 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

19.4.7 UART 接收器

要使能接收器功能，必须将发送使能位（UART_CR0.RE 位）置 1。

要发送数据，需要遵循以下步骤：

1. 配置 UART_CR1.LEN 位以确定发送字符长度
2. 配置 UART_BAUD.BAUD 位以确定所需波特率
3. 配置 UART_CR1.STP 位以确定停止位数量
4. 配置 UART_CR1.PEN 位、UART_CR1.PSEL 位和 UART_CR1.SPE 位以确定奇偶检验位
5. 配置 UART_CR0.UE 位以使能 UART 模块
6. 配置 UART_CR0.RE 位以使能 UART 接收器
7. 接收到字符后
   a) 禁用 FIFO 模式（UART_CR0.NFE=1）时，UART_INT.RXFI 位如果置 1，表明移位寄存器的内容已经被存入 UART_RDR 寄存器中，可以直接读取 UART_RDR 寄存器获得收到的数据。
   b) 使能 FIFO 模式（UART_CR0.NFE=0）时，UART_STATUS.RFE 位如果置 0，表示当前 RXFIFO 中至少有一个数据，读取 UART_RDR 寄存器会返回最先被存进 RXFIFO 的那个数据，同时相应的错误位也会随着读取 UART_RDR 寄存器被读回。
   c) 在接收期间如果检测到帧错误或者奇偶校验错误时，相应的错误中断状态被置 1

19.4.8 RS485 接口

RS485 和 RS232 一样，都是串行通信标准，都是 RS485 总线弥补了 RS232 通信距离短，速率低的缺点，RS485 理论上能达到高达 10Mbit/s，理论通讯距离可达 1200 米。RS485 和 RS232 的单端传输是不一样的，RS485 是差分传输，使用了一对双绞线进行传输。

UART 支持 RS485 接口支持，可以使用 RS485 接口传输串行数据。驱动器使能（DE）和接收器使能（RE）的产生是为了启用 RS485 接口支持。DE 和 RE 信号由硬件产生，这些信号的断言和解除断言的时间都是可编程的，有效电平也是可编程的。

配置 UART 用于 RS485 接口的步骤流程如下所示：

1. 配置 UART_CR1.RS485E 位来使能 RS485 模式
2. UART_CR1.REP 位和 UART_CR1.DEP 位分别控制 RE 和 DE 信号的极性。
3. UART_TIMING0 寄存器用于编程 DE 信号的驱动禁止时间和驱动使能时间。
4. UART_TIMING1 寄存器用于编程 DE 到 RE 以及 RE 到 DE 的切换时间。

注意：以上配置流程没有严格的顺序，在软件应用中可以采用任何顺序对 RS485 进行配置。

DE 信号的驱动禁止时间和驱动使能时间都是通过 UART_TIMING0 寄存器控制的。

- DE 的驱动使能时间（UART_TIMING0.DEAT 位）。驱动使能时间指的是 DE 信号的有效到 START 位开始之间的时间长度，所表示的值是以 UART 的系统时钟周期 pclk 为单位

的。

- DE 的驱动禁止时间（UART_TIMING0.DEDT 位）。驱动禁止时间指的是最后一个 STOP 位的结束到 DE 信号的无效的时间长度，所表示的值是以 UART 的系统时钟周期 pclk 为单位的。

DE 的驱动禁止时间和驱动使能时间具体如图 19-4 所示，其中 t1 代表 DE 驱动使能时间，t2 代表 DE 驱动禁止时间。

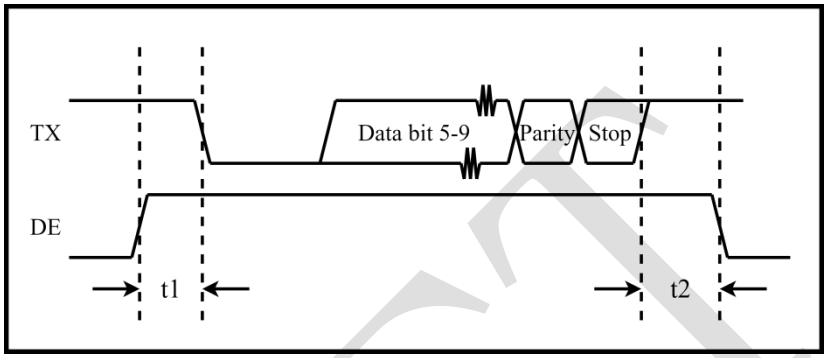


图 19-6　DE 线上时序图

注意：上图中只是说明了一个数据，然而如果 TXFIFO 中还有更多的数据，DE 就不会被解除信号，只有在所有的数据字符传输完毕后，DE 才会被解除。

此外硬件还需要确保在从 RE 切换到 DE 或者从 DE 切换到 RE 时保持一个适当的切换时间。可以通过配置 TIMING1.RDET 寄存器保证 RE 切换到 DE 的切换时间，配置 TIMING1.DRET 寄存器保证 DE 切换到 RE 的切换时间。具体切换时间时序如图 19-5 所示。

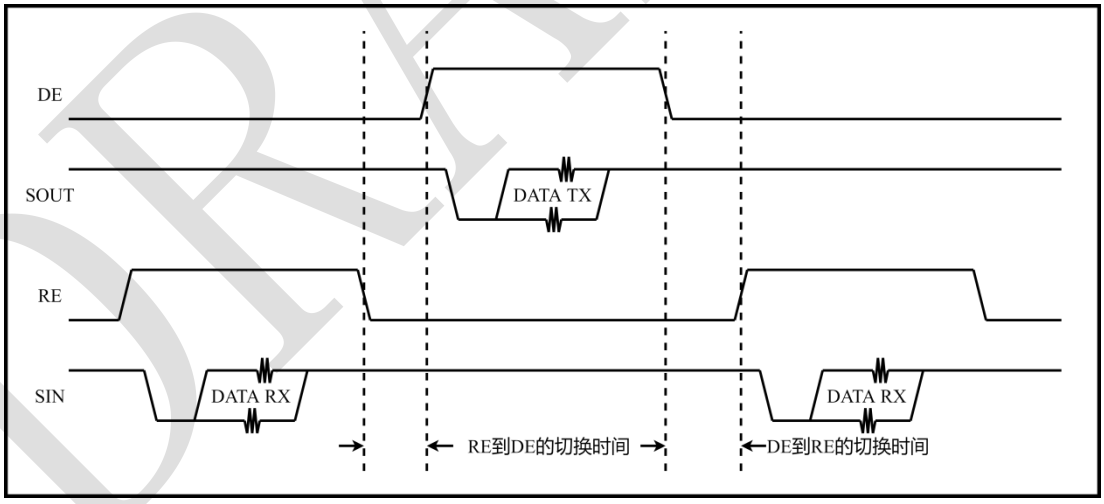


图 19-7　DE/RE 切换时间时序图

19.4.9　UART 拓展位

使用了拓展位 UART_CR1.EBE 位之后，实际传输的数据域宽度将在 UART_CR1.LEN 位的基础上增加一位。该位可以通过 UART_CR1.TEM 位和 UART_CR1.REM 位来确定要不要跑特殊模式。如果配置成 1，则是地址匹配模式，拓展位为 1 代表地址字节，为 0 代表数据字节，若配置了 UART_CR1.REM 位为 1，从机在收到地址字节时会和自己预设的地址进行比较，如果匹配，后续的数据才会被接收。图 19-6 显示了在拓展位配置成 1 且 UART_CR1.LEN 位配置

为 3 时，也就是 9BIT 串行传输地址字节和数据字节的拓展位时序图。

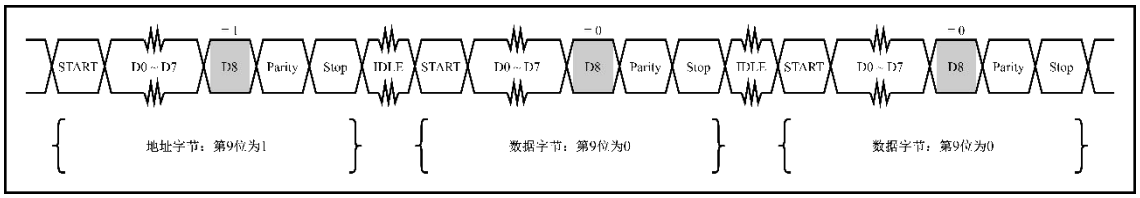


图 19-8　拓展位串行传输时序图

UART 实现拓展位传输主要配置流程如下所示：

1. 将 UART_CR1.EBE 位设置为 1 以启用 UART 的拓展位
2. 通过配置 UART_CR1.TEM 位来确定当前发送拓展模式
3. 通过配置 UART_TAR 寄存器来确定要发送的地址字节
4. 通过配置 UART_CR1.REM 位来确定当前接收拓展模式
5. 通过配置 UART_RAR 寄存器来确定需要进行匹配的地址字节

注意：以上配置不需要严格按照顺序配置。

19.4.9.1 普通模式

普通模式和 UART 5-8BIT 的发送接收一样，只是多了一位传输。

19.4.9.2 发送地址匹配模式

启用发送地址匹配模式需要将 UART_CR1.TEM 位配置为 1，同时要在 TAR 寄存器中写入需要发送的地址。UART 将发送 TAR 寄存器中预先设置的地址数据，拓展位自动置 1 表示为地址字节，此外为了防止 TXFIFO 数据与发送地址数据出现不可控的先后问题，该位的使用有以下的限制：

1. TX 使能（UART_CR0.TE 位)前，允许预先对 TXFIFO 填入数据，并在 UART_CR0.TE 位置 1 之前，配置 UART_CR1.TEM 位为 1。如此在 TX 使能后，UART 先发出地址数据，再发送 TXFIFO 内预置的数据
2. TX 使能（UART_CR0.TE 位)后，必须在 TXFIFO 为空后,即当前 TX 传输完成后，才允许设置 TEM=1 以启动下一次地址的发送。在 TXFIFO 非空期间，将 UART_CR1.TEM 位置 1 会发出全 0 地址。
3. 地址帧发送完成后，TEM 自动清 0

19.4.9.3 接收地址匹配模式

启用接收地址匹配模式需要将 UART_CR1.REM 位配置为 1，同时要在 UART_RAR 寄存器中写入需要进行匹配的地址。在 UART 接收过程中有以下情况：

1. 接收到地址字节，地址字节和 UART_RAR 的值匹配，后续的所有数据字节都会被接收，直到收到新的地址字节会重新进行判决。
2. 接收到地址字节，地址字节和 UART_RAR 的值不匹配，后续的所有数据字节都不会被接收，直到收到新的地址字节会重新进行判决。

3. 接收到数据字节，如果此前的地址字节和 UART_RAR 匹配，则该数据字节会被 UART 接收；如果此前的地址字节和 UART_RAR 不匹配或者此前还没有发过地址字节，则该数据字节不会被 UART 接收。
4. 在地址已经匹配的情况下，软件改变了 UART_RAR 寄存器的值，赋予了一个新的 UART_RAR 地址，那么在没有收到新的地址字节之前，所有的数据字节还是会被接收，收到新的地址字节之后，新的地址字节会和新的 UART_RAR 地址进行匹配。

19.5 寄存器定义

19.5.1 寄存器列表

UART 基地址：
UART0(0x40003000) UART1(0x40004000) UART2(0x40005000)
UART3(0x40010000) UART4(0x40011000)

偏移	实例地址	名称	默认值	描述
0x00	UART 基地址+0x00	UART_CR0	0x00000000	UART 使能寄存器 0
0x04	UART 基地址+0x04	UART_CR1	0x00000000	UART 控制寄存器 1
0x08	UART 基地址+0x08	UART_BAUD	0x00000000	UART 波特率寄存器
0x10	UART 基地址+0x10	UART_FIFOCTRL	0x00000000	UART FIFO 控制寄存器
0x14	UART 基地址+0x14	UART_TIMING0	0x00000000	UART DE 时序寄存器
0x18	UART 基地址+0x18	UART_TIMING1	0x00000000	UART 周转时序寄存器
0x20	UART 基地址+0x20	UART_TDR	0x00000000	UART TX 数据寄存器
0x24	UART 基地址+0x24	UART_RDR	0x00000000	UART RX 数据寄存器
0x28	UART 基地址+0x28	UART_TAR	0x00000000	UART 发送地址寄存器
0x2C	UART 基地址+0x2C	UART_RAR	0x00000000	UART 接收地址寄存器
0x30	UART 基地址+0x30	UART_RTO	0x00000000	UART 接收器超时时序寄存器
0x34	UART 基地址+0x34	UART_INTEN	0x00000000	UART 中断使能寄存器
0x38	UART 基地址+0x38	UART_INT	0x00000002	UART 中断寄存器
0x3C	UART 基地址+0x3C	UART_STATUS	0x00000202	UART 状态寄存器

19.5.2 寄存器描述

19.5.2.1 UART 使能寄存器 0（UART_CR0）

- 名称：UART Control Register0
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		NFE	OVER8	RTOE	TPOL	RPOL	SWAP	OWE	DTE	DRE	TFR	RFR	TE	RE	UE
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W	R/W
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[13] NFE	**UART 非 FIFO 模式使能（UART Non-FIFO Mode Enable）** 0：FIFO 模式 1：非 FIFO 模式 Note：仅在 UE=0 时才能修改；
[12] OVER8	**UART 过采样模式选择（UART Oversampling mode Select）** 0：16 倍过采样 1：8 倍过采样 Note：仅在 UE=0 时才能修改；
[11] RTOE	**UART RX 超时使能（UART RX Timeout Enable）** 0：关闭 1：使能
[10] TPOL	**UART TX 引脚极性选择（UART TX Polarity Select）** 0：不取反，默认无效电平为高电平 1：取反，默认无效电平为低电平
[9] RPOL	**UART RX 引脚极性选择（UART RX Polarity Select）** 0：不取反，默认无效电平为高电平 1：取反，默认无效电平为低电平
[8] SWAP	**UART TX 和 RX 引脚交换使能（UART TX/RX Swap Enable）** 0：关闭 1：使能
[7] OWE	**UART 单线模式使能（UART One-Wire Enable）** 0：关闭 1：使能 Note： 1）在 OneWire 模式下，当同时使能 TX/RX，硬件将自动切换 Uart TX/RX 传输模式，默认为 RX 模式，当 TX-FIFO 非空且 RX 总线 IDLE 则切换到 TX 模式，TX 传输完成自动切换到 RX； 2）在 OneWire 模式下，当仅使能 RX 或 TX 模式下，硬件将根据软件配置进行 TX 或 RX 传输； Note：仅在 UE=0 时才能修改；
[6] DTE	**UART TX-DMA 使能（UART TX DMA Enable）** 0：关闭 1：使能
[5] DRE	**UART RX-DMA 使能（UART RX DMA Enable）** 0：关闭 1：使能
[4]	**UART TX-FIFO 复位（UART TX-FIFO Reset）**

TFR	0：放开正常 1：发起复位 Note: 1）该 Bit 只写； 2）向该位写入“1”将清空整个 TX-FIFO，会将 TX-FIFO Empty 标志置位；
[3] RFR	**UART RX-FIFO 复位（UART RX-FIFO Reset）** 0：放开正常 1：发起复位 Note: 1）该 Bit 只写； 2）向该位写入“1”将清空整个 RX-FIFO，可以丢弃接受的数据而不需对其进行读操作，避免造成上溢情况；
[2] TE	**UART TX 使能（UART TX Enable）** 0：关闭 1：使能
[1] RE	**UART RX 使能（UART RX Enable）** 0：关闭 1：使能
[0] UE	**UART 使能（UART Enable）** 0：关闭 1：使能

19.5.2.2 UART 控制寄存器 1（UART_CR1）

- **名称：UART Control Register1**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											IDR	BKR	TEM	REM	EBE
											R/W	R/W	R/WAC	R/W	R/W
											0x0	0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					DEP	REP	RS485E	MSB		SPE	PSEL	PEN	STP	LEN	
					R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	
					0x0	0x0	0x0	0x0		0x0	0x0	0x0	0x0	0x0	

字段	说明
[20] IDR	**UART IDLE 帧请求位（UART IDLE Request）** 0：关闭 Idle 操作 1：发起 Idle 操作 Note： 1）向该位写入“1”,在完成当前帧传输后在 TX 线上发送 Idle，Idle 的长度由帧长度控制；向该位写“0”则立即停止 Idle 发送； 2）该位由硬件自动清 0，软件需等待该位清 0，表示 Idle 帧发送完成；
[19] BKR	**UART Break 帧请求位（UART Break Request）** 0：关闭 Break 操作 1：发起 Break 操作 Note： 1）向该位写入“1”,在完成当前帧传输后在 TX 线上发送 Break，BREAK 的长度由长度由帧长度控制；向该位写“0”则立即停止 Break 发送； 2）如果应用需要在所有数据（包括尚未发送的数据）后发送 Break 字符，软件应等到 TXEI 或 TDIF 标志后将 BKE 位置 1。 3）该位由硬件自动清 0，软件需等待该位清 0，表示 Break 帧发送完成；
[18]	**UART TX 拓展模式选择（UART TX Extend-Mode Select）**

TEM	0：普通模式 1：地址发送模式 Note： 1）普通模式下，UART 的拓展位将被当作普通数据位，与数据宽度设定的数据一起被正常写入 TXFIFO 发送至总线。 * 注意，普通模式下拓展位的值交由用户自行设置，用户软件上需要自行保证发送的数据中拓展位的值，需符合对方接收设备对拓展位的定义要求。 2）地址发送模式，设置 TEM=1，UART 将发送 TAR 寄存器中预先设置的地址数据，拓展位自动置 1 标识为地址帧。此外，为防止 TXFIFO 数据与发送地址数据出现不可控的先后问题，该位的使用有如下限制： * TX 使能前（TE=0 时），允许预先对 TXFIFO 填入数据，并在 TE=1 之前，设置 TEM=1。如此在 TX 使能（TE=1）后，UART 先发出地址数据，再发送 TXFIFO 内预置的数据。 * TX 使能后（TE=1 时），必须在 TXFIFO 为空后,即当前 TX 传输完成后，才允许设置 TEM=1 以启动下一次地址的发送。在 TXFIFO 非空期间，设置 TEM 将发出全 0 地址。 * 地址帧发送完成后，TEM 自动清 0。 3）TEM 位仅在启用拓展位功能（EBE=1）时有效，否则设置无效。
[17] REM	**UART RX 拓展模式选择（UART RX Extend-Mode Select）** 0：普通模式 1：地址匹配模式 Note： 1）在匹配模式下，UART 检测到拓展位为 1 时，将这一帧数据与 RAR 寄存器预设值的地址进行匹配， * 如果匹配一致，将数据存入 RXFIFO（包括拓展位），触发地址匹配中断(AMIF=1)，且后续数据将正常接收。直到下一次地址匹配（下一次拓展位=1）进行重新匹配。 * 如果匹配不一致，则地址数据被丢弃，并且后续数据不会被接收。直到下一次地址匹配（下一次拓展位=1）进行重新匹配。 2)普通模式下,UART 的拓展位将被当作普通数据位,与数据宽度设定的数据一起被正常接收至 RXFIFO 内。 3）REM 位仅在启用拓展位功能（EBE=1）时有效，否则设置无效。
[16] EBE	**UART 拓展位使能（UART Extend-Bit Enable）** 0：关闭 1：使能 Note： 1）拓展位定义：使能拓展位后，实际传输的数据域宽度将在配置 LEN 表示的宽度基础上+1 位。例如 5bit(LEN=0）时，总线数据如下： | START | 6(5+1 位拓展）bit 数据域 | PE | STOP | 注：数据域（包括拓展位）在总线传输时，都将受 MSB/LSB 控制； 2）拓展位的位置：定义于 TX 数据寄存器（TDR）的 LEN 表示的宽度+1 的位置上。例如 5bit（LEN=0）位宽时，TDR 的 bit[4:0]为数据位，bit[5]为拓展位；RDR 同理。 3）仅在 UE=0 时才能修改；
[10] DEP	**UART DE 极性（UART DE Polarity）** 0：DE 低电平有效 1：DE 高电平有效
[9] REP	**UART RE 极性（UART RE Polarity）** 0：RE 低电平有效 1：RE 高电平有效
[8] RS485E	**UART RS485 功能使能（UART RS485 Enable）** 0：关闭 1：使能 Note： 1）使能 RS485 功能后，DE/RE 信号将根据 TX/RX 的使能情况自动进行转换： * 仅开 TX 使能时（TE=1,RE=0），DE 信号在 TXFIFO 有数据需要传输后输出有效电平，数据发送完成后 DE 转为无效电平；RE 保持无效电平。 * 仅开 RX 使能时（TE=0,RE=1），DE 信号保持无效电平，RE 信号保持有效电平。 * TX-RX 都使能时（TE=1,RE=1），DE 信号在 TXFIFO 有数据需要传输后输出有效电平，RE 信号转为无效电平；数据发送完成后 DE 转为无效电平，RE 转为有效电平。 * 此外，DE/RE 的电平转换时序符合 TIMING0/TIMING1 寄存器的时序控制。 2）仅在 UE=0 时才能修改；
[7] MSB	**UART MSB 使能（UART MSB Enable）** 0：LSB

	1：MSB Note： 1）设置 UART 总线上数据域（如果使能拓展位，则包括拓展位）的数据排列方式，MSB 为高位在前，LSB 为低位在前 2）仅在 UE=0 时才能修改；
[5] SPE	**UART 强制校验使能位（UART Stick Parity Enable）** 0：关闭 1：使能 Note： 1）强制奇偶校验位用于在奇偶校验位使能（PEN 使能）的情况下； 2）根据 PSEL 位的配置，当 PSEL=1 时，强制发送或者检测逻辑 1；当 PSEL=0 时，强制发送或者检测逻辑 0；
[4] PSEL	**UART 奇偶检验选择（UART Parity Select）** 0：偶校验 1：奇校验
[3] PEN	**UART 奇偶检验位使能（UART Parity Enable）** 0：关闭 1：使能 Note: 仅在 UE=0 时才能修改；
[2] STP	**UART Stop 位选择（UART Stop Select）** 0：1 个 Stop 位 1：2 个 Stop 位 Note: 仅在 UE=0 时才能修改；
[1:0] LEN	**UART 数据长度选择（UART Length Select）** 00：5bit 01：6bit 10：7bit 11：8bit Note: 仅在 UE=0 时才能修改；

19.5.2.3 UART 波特率寄存器（UART_BAUD）

- **名称：UART Baud Rate Register**
- **偏移地址：0x08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAUD															
R/W															
0x0															

字段	说明
[15:0] BAUD	**UART 波特率配置（UART Baud Rate）** UART 波特率设置，公式为：波特率=PCLK 频率/BAUD Note: 仅在 UE=0 时才能修改；

19.5.2.4 UART FIFO 控制寄存器（UART_FIFOCTRL）

- 名称：UART FIFO Control Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TXFT				RXFT			
								R/W				R/W			
								0x0				0x0			

字段	说明
[7:4] TXFT	**UART TXFIFO 空中断阈值（UART TXFIFO Level-Empty Threshold）** 0x0：FIFO 中没有数据就会触发 TX-FIFO 空中断 0x1：FIFO 中有 1 个或 1 个以下就会触发 TXFIFO 空中断 …… 0xe：FIFO 中有 14 个或 14 个以下就会触发 TXFIFO 空中断 0xf：FIFO 中有 15 个或 15 个以下就会触发 TXFIFO 空中断
[3:0] RXFT	**UART RXFIFO 满中断阈值（UART RX-FIFO Level-Full Threshold）** 0x0：FIFO 内有 1 个或 1 个以上数据就会触发 RX-FIFO 满中断 0x1：FIFO 内有 2 个或 2 个以上数据就会触发 RX-FIFO 满中断 …… 0xe：FIFO 中有 15 个或 15 个以上数据就会触发 RX-FIFO 满中断 0xf：FIFO 中有 16 个数据就会触发 RX-FIFO 满中断

19.5.2.5 UART DE 时序寄存器（UART_TIMING0）

- 名称：UART Timing Register0
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEDT															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEAT															
R/W															
0x0															

字段	说明
[31:16] DEDT	**UART DE 驱动禁止时间（UART DE Deassertion Timing）** 从发送完最后一个 STOP 位到取消 DE 信号之间的时间。 Note： 1) 实际时间为(DEDT+1)*16 个时钟周期。 2) 仅在 UE=0 时才能修改；
[15:0] DEAT	**UART DE 驱动使能时间（UART DE Assertion Timing）** 从激活 DE 的有效沿到发送 START 位之间的时间。 Note： 1) 实际时间为(DEAT+1)*16 个时钟周期。 2) 仅在 UE=0 时才能修改；

19.5.2.6 UART 周转时序寄存器（UART_TIMING1）

- 名称：UART Timing Register1
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRET															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDET															
R/W															
0x0															

字段	说明
[31:16] DRET	**UART DE 到 RE 切换时间（UART DE2RE Turn-Around Timing）** DE 无效沿到 RE 有效沿的周转时间。 Note： 1) 实际时间为(DRET+1)*16 个时钟周期。 2) 仅在 UE=0 时才能修改；
[15:0] RDET	**UART RE 到 DE 切换时间（UART RE2DE Turn-Around Timing）** RE 无效沿到 DE 有效沿的周转时间。 Note： 1) 实际时间为(RDET+1)*16 个时钟周期。 2) 仅在 UE=0 时才能修改；

19.5.2.7 UART TX 数据寄存器（UART_TDR）

- 名称：UART Transmit Data Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TD								
							R/W								
							0x0								

字段	说明
[8:0] TD	**UART TX 数据寄存器（UART Transmit Data Register）** Uart TX 要发送的数据。 Note： 1）TDR 寄存器的 TD 位域的实际有效位宽，与 LEN 表示的位宽一致 2）如果启用拓展位功能（EBE=1），则实际有效位宽+1，拓展位定义在实际有效位宽的最高位。

19.5.2.8 UART RX 数据寄存器（UART_RDR）

- **名称：UART Receive Data Register**
- **偏移地址：0x24**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					FMST	PRST	RD								
					R	R	R								
					0x0	0x0	0x0								

字段	说明
[10] FMST	**UART RX 帧状态（UART RX Frame Status）** 0：Stop 位正常 1：Stop 位错误 Note：当前读回来的数据的 Stop 位是否正确；
[9] PRST	**UART RX 奇偶校验位状态（UART RX Parity Status）** 0：奇偶校验位正常 1：奇偶校验位错误 Note：当前读回来的数据的奇偶校验位是否正确；
[8:0] RD	**UART RX 数据寄存器（UART Receive Data Register）** Uart RX 接收的数据。 Note： 1）RDR 寄存器的 RD 位域的实际有效位宽，与 LEN 表示的位宽一致 2）如果启用拓展位功能（EBE=1），则实际有效位宽+1，拓展位定义在实际有效位宽的最高位。

19.5.2.9 UART 发送地址寄存器（UART_TAR）

- **名称：UART Transmit Address Register**
- **偏移地址：0x28**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TAR							
								R/W							
								0x0							

字段	说明
[7:0] TAR	**UART TX 发送地址寄存器（UART Transmit Address Register）** Note： 1）仅当使能拓展位，且 TEM 写 1 进行地址发送模式时有效 2）TAR 寄存器设置发送地址的有效位宽与 LEN 表示的位宽一致；

19.5.2.10 UART 接收地址寄存器（UART_RAR）

- 名称：UART Receive Address Register
- 偏移地址：0x2C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RAR							
								R/W							
								0x0							

字段	说明
[7:0] RAR	**UART RX 接收地址寄存器（UART Receive Address Register）** Note： 1）仅当使能拓展位，且 REM 配置为地址匹配模式后，该寄存器才有用； 2）RAR 寄存器设置接收地址的有效位宽与 LEN 表示的位宽一致；

19.5.2.11 UART 接收器超时时序寄存器（UART_RTO）

- 名称：UART Receiver Timeout Register
- 偏移地址：0x30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO															
R/W															
0x0															

字段	说明
[15:0] RTO	**UART RX 超时阈值（Receiver Timeout Timing）** 为 RTOI 中断提供超时时长，单位为 1 个位(波特率)持续时间，实际值等于配置值+1；

19.5.2.12 UART 中断使能寄存器（UART_INTEN）

- 名称：UART Interrupt Enable Register
- 偏移地址：0x34
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AMIE	IDLE	TBIE	TDIE	RTIE	BKIE	FEIE	PEIE				TOIE	RUIE	ROIE	TEIE	RFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0				0x0	0x0	0x0	0x0	0x0

字段	说明
[15] AMIE	**UART RX 地址匹配中断使能（UART RX Address Match Interrupt Enable）** 0：关闭 1：使能
[14] IDLE	**UART TX Idle 完成中断使能（UART TX Idle Done Interrupt Enable）** 0：关闭 1：使能
[13] TBIE	**UART TX Break 完成中断使能（UART TX Break Done Interrupt Enable）** 0：关闭 1：使能
[12] TDIE	**UART TX 传输完成中断使能（UART TX Done Interrupt Enable）** 0：关闭 1：使能
[11] RTIE	**UART RX 传输超时中断使能（UART RX Timeout Interrupt Enable）** 0：关闭 1：使能
[10] BKIE	**UART RX Break 中断使能（UART RX Break Interrupt Enable）** 0：关闭 1：使能
[9] FEIE	**UART RX 传输帧错误中断使能（UART Frame Error Interrupt Enable）** 0：关闭 1：使能
[8] PEIE	**UART RX 奇偶校验错误中断使能（UART Parity Error Interrupt Enable）** 0：关闭 1：使能
[4] TOIE	**UART TX-FIFO 上溢中断使能（UART TX-FIFO Overflow Interrupt Enable）** 0：关闭 1：使能
[3] RUIE	**UART RX-FIFO 下溢中断使能（UART RX-FIFO Underflow Interrupt Enable）** 0：关闭 1：使能
[2] ROIE	**UART RX-FIFO 上溢中断使能（UART RX-FIFO Overflow Interrupt Enable）** 0：关闭 1：使能
[1] TEIE	**UART TX-FIFO 空中断使能（UART TX-FIFO Empty Interrupt Enable）** 0：关闭 1：使能
[0] RFIE	**UART RX-FIFO 满中断使能（UART RX-FIFO Full Interrupt Enable）** 0：关闭 1：使能

19.5.2.13 UART 中断寄存器（UART_INT）

- 名称：**UART Interrupt Register**
- 偏移地址：**0x38**
- 默认值：**0x00000002**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AMIF	IDLF	TBIF	TDIF	RTOI	BKIF	FEIF	PEIF				TOIF	RUIF	ROIF	TXEI	RXFI
R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C				R/W1C	R/W1C	R/W1C	R	R
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0				0x0	0x0	0x0	0x1	0x0

字段	说明
[15] AMIF	**UART RX 地址匹配中断标志（UART RX Address Match Interrupt Flag）** 0：无匹配中断 1：匹配中断 Note: 1) 在 RX 地址匹配模式下，接收的地址会与 RAR 进行比较，当匹配时会置起该中断； 2) 该位写 1 清 0；
[14] IDLF	**UART TX Idel 完成中断标志（UART TX Idle Done Interrupt Flag）** 0：Idle 未完成 1：Idle 完成标志 Note： 1）在 TX 模式下，完成 Idle 请求发送标志； 2）该位写 1 清 0；
[13] TBIF	**UART TX Break 完成中断标志（UART TX Break Done Interrupt Flag）** 0：Break 未完成 1：Break 完成标志 Note： 1）在 TX 模式下，完成 Break 请求发送标志； 2）该位写 1 清 0；
[12] TDIF	**UART 发送完成中断标志（UART TX Done Interrupt Flag）** 0：未发送完成 1：发送完成 Note： 1）UART TX 发送完成且 TX-FIFO 为空将置起该中断； 2）该位写 1 清 0；
[11] RTOI	**UART RX 传输超时中断标志（UART RX Timeout Interrupt Flag）** 0：无超时 1：超时 Note： 1）该位写 1 清 0； 2）FIFO 模式 在 RX 模式下，从上一笔 STOP 开始，当总线处于空闲而且 FIFO 非空状态下，超过 RTO 个位长度时间将置起该中断； 非 FIFO 模式 在 RX 模式下，从上一笔 STOP 开始，当总线处于空闲状态下，超过 RTO 个位长度时间将置起该中断；
[10] BKIF	**UART RX BREAK 中断标志（UART Break Interrupt Flag）** 0：非 break 状态 1：break 状态 Note： 1）在 RX 模式下，用于指示 RX 线上检测到串行数据上的 break 序列； 2）在 RX 模式下，输入 RX 线上逻辑 0 状态的保持时间长于 START 位+LEN 个串行数据位+奇偶检验位+STOP 位的总和；

	3）该位写 1 清 0；
[9] FEIF	**UART RX 帧错误中断标志（UART Frame Error Interrupt Flag）** 0：没有帧错误 1：帧错误 Note: 1）在 RX 模式下，UART RX 线上检测到接收数据中没有有效的 Stop 位时就会发生帧错误中断； 2) 该位写 1 清 0；
[8] PEIF	**UART 奇偶校验错误中断标志（UART Parity Error Interrupt Flag）** 0：无奇偶校验位错误 1：奇偶校验位错误 Note： 1）在 RX 模式下，UART 检测到接收数据中的奇偶校验位错误时就会置位； 2) 该位写 1 清 0；
[4] TOIF	**UART TX-FIFO 上溢错误中断标志（UART TX-FIFO Overflow Interrupt Flag）** 0：无溢出错误 1：溢出错误 Note： 1）在 FIFO 模式下，在 TX 传输中，当 TXFIFO 已满并且新数据被写入 FIFO 中时，会起溢出错误中断。此时 FIFO 中的数据将保留，而接收的新数据将丢失。 2）在非 FIFO 模式下，在 TX 传输中，当 TDR 已经写入新数据，再次发起写 TDR 时就会产生溢出错误中断，此时数据会被覆盖。 3）该位写 1 清 0；
[3] RUIF	**UART RX-FIFO 下溢中断标志（UART RX-FIFO Underflow Interrupt Flag）** 0：无溢出错误 1：溢出错误 Note： 1）在 FIFO 模式下，在 RX 模式下，当 RX-FIFO 已空并且但是还是有从 FIFO 取数的操作，会起下溢错误中断 2）在非 FIFO 模式下，在 RX 模式下，当 RDR 已空并且还是有从 TDR 取数的操作，会起下溢错误中断 3）该位写 1 清 0；
[2] ROIF	**UART RX-FIFO 上溢中断标志（UART RX-FIFO Overflow Interrupt Flag）** 0：无溢出错误 1：溢出错误 Note： 1）在 FIFO 模式下，在 RX 模式下，当 RX-FIFO 已满并且但是还有写 FIFO 的操作，会起上溢错误中断 2）在非 FIFO 模式下，在 RX 模式下，当 RDR 已有数据并且还有写 TDR 的操作，会起上溢错误中断 3）该位写 1 清 0；
[1] TXEI	**UART TX-FIFO 阈值空中断标志（UART TX-FIFO Level-Empty Interrupt Flag）** FIFO 模式： 0：TXFIFO 数据量大于预设值 1：TXFIFO 数据量小于等于预设值 非 FIFO 模式： 0：TDR 内写入了新数据 1：TDR 内数据没有数据 Note： 1）当 TXFIFO 中的数据量小于或者等于预设值（TXFTLR）时置位； 2）当 TXFIFO 数据量大于预设值时 TXFIFOEI 中断自动清 0；
[0] RXFI	**UART RX-FIFO 阈值满中断标志（UART RX-FIFO Level-Full Interrupt Flag）** FIFO 模式： 0：RX-FIFO 数据量小于预设值 1：RX-FIFO 数据量大于等于预设值 非 FIFO 模式： 0：RDR 内数据没有数据 1：RDR 内写入了新数据 Note： 1）当 RXFIFO 中的数据量大于或等于预设值（RXFTLR）时置位； 2）当 RXFIFO 数据量小于预设值时 RXFIFOFI 中断自动清 0；

19.5.2.14 UART 状态寄存器（UART_STATUS）

- 名称：UART Status Register
- 偏移地址：0x3C
- 默认值：0x00000202
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFL					RFF	RFE	RBSY	TFL					TFF	TFE	TBSY
R					R	R	R	R					R	R	R
0x0					0x0	0x1	0x0	0x0					0x0	0x1	0x0

字段	说明
[15:11] RFL	**UART RX-FIFO 实时剩余数据量（UART RX-FIFO Level Register）**
[10] RFF	**UART RX 接收 FIFO 满标志（UART RX-FIFO Full Flag）** 0：接收 FIFO 未满 1：接收 FIFO 已满 Note：当 RX-FIFO 未满时，该位自动清 0。
[9] RFE	**UART RX 接收 FIFO 空标志（UART RX-FIFO Empty Flag）** 0：接收 FIFO 不为空 1：接收 FIFO 为空 Note：当 RX-FIFO 未空时，该位自动清 0。
[8] RBSY	**UART RX 总线忙碌标志（UART RX Busy Flag）** 0：RX 空闲 1：RX 忙碌
[7:3] TFL	**UART TX-FIFO 实时剩余数据量（UART TX-FIFO Level Register）**
[2] TFF	**UART TX 发送 FIFO 满标志（UART TX-FIFO Full Flag）** 0：发送 FIFO 未满 1：发送 FIFO 为满 Note：当 TX FIFO 未满时，该位自动清 0。
[1] TFE	**UART TX 发送 FIFO 空标志（UART TX-FIFO Empty Flag）** 0：发送 FIFO 不为空 1：发送 FIFO 为空 Note：当 TX-FIFO 未空时，该位自动清 0。
[0] TBSY	**UART TX 总线忙碌标志（UART TX Busy Flag）** 0：TX 空闲 1：TX 忙碌

20 控制器局域网络（CAN）

20.1 简介

CAN 协议中存在两种类型的帧：标准帧和扩展帧，两者的区别主要是体现在 ID 长度的不同，标准帧的 ID 是 11 位的，扩展帧是 29 位的，此外，CAN2.0 数据帧的最大数据长度最多为 8 个字节，CANFD 数据帧的最大数据长度最多为 64 个字节。

数据寻址是通过发送 ID 来完成的，在 CAN 网络中，每一个节点都会有自己的 ID 标识符，每次传输只有一个节点会发送带有特定 ID 的消息，此时所有节点都应接收该消息，并且每个节点通过将接收的 ID 标识符和自己特定的 ID 标识符对比来决定是否接收该消息。为了减少主机控制器的负载，CAN 使用了验收滤波器，这些滤波器将所有接收到的 ID 标识符与用户配置的 ID 进行比较。仅当消息通过验收滤波器时，CAN 才会将数据全部接收并回复 ACK。

CAN 2.0B 标准协议规定了最高 1Mbit/s 的数据比特率，而 CAN FD 协议并没有规定最高的数据比特率，在 CAN 中最高的 CANFD 数据比特率为 5Mbit/s。

20.2 主要特性

- 支持 CAN2.0 协议
- 支持 CAN FD 协议
- 支持 NON-ISO FD 和 ISO FD 协议
- CAN FD 最多支持 64 字节传输，CAN 2.0 最多支持 8 字节传输
- 支持可编程数据速率（CAN 2.0 最高 1Mbit/s 的数据速率，CAN FD 最高 5Mbit/s 的数据速率）
- 支持可编程波特率预分频器（1/2 至 1/256）
- 支持 1 帧数据的接收缓冲区（RB）大小
- 两个发送缓冲区（一个主发送缓冲器 PTB 和一个辅助发送缓冲器 STB）
- 支持 16 个独立的可编程内部 29 位验收滤波器
- 支持单发传输模式（用于 PTB 和 STB）
- 支持仅收听模式
- 支持回环模式（内部回环和外部回环）
- 支持错误状态的指示（捕获上次发生的错误、捕获仲裁丢失的位置和可编程的错误警告极限）
- 支持中断模式

20.3 结构框图

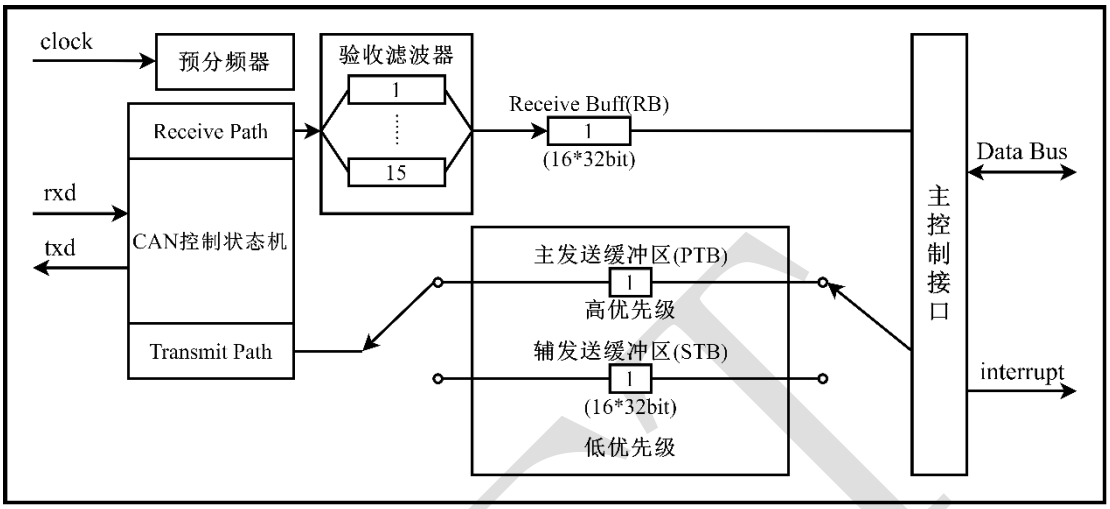


图 20-1 CAN 结构框图

20.4 功能描述

表 20-1 英文缩写列表

英文缩写	英文全称	中文描述
ACF	Acceptance Filter	验收滤波器
PTB	Primary Transmit Buffer (high priority)	主要发送缓冲区（高优先级）
RB	Receive Buffer	接收缓冲区
SP	Sample Point	采样点
STB	Secondary Transmit Buffer (low priority)	辅助发送缓冲区（低优先级）
TQ	Time Quanta (CAN specification)	CAN 时间量

20.4.1 时钟域交叉

CAN 中一共有两个时钟，一个是系统时钟用于主机接口读写寄存器，一个是功能时钟用于 CAN 收发。在启动 CAN 之前，可以使用最适合 CAN 总线数据速率的时钟来操作 CAN 进行收发，同时主机可以在不同（更高）的时钟速度下自由操作寄存器。

对于系统时钟域生成的寄存器信号需要同步到 CAN 功能时钟域才能被使用，同时 CAN 功能时钟域产生的状态寄存器也需要同步到系统时钟域才能被 CPU 读取，这就需要时钟域交叉。时钟域交叉（CDC）是通过双缓冲同步器和双向握手机制完成的。以下提供了有关 CDC 的详细信息：

1. 控制寄存器由系统时钟域移动到功能时钟域，状态寄存器由功能时钟域移动到系统时钟域。
2. 当 RESET 位变为无效（配置基本寄存器结束）时，验收滤波器在启动期间被同步到 CAN 功能时钟域。当 CAN 节点在总线空闲时间后参与通信时，验收滤波器复制完毕。
3. 接收到的 CAN 数据帧会先存储在一个临时接收缓冲区中，传输结束会将这些数据复制到

RB 中

4. 要传输的帧会从 TBUF 中取出，然后放入一个临时传输缓冲区。

使用时钟域交叉（CDC）时会导致一些延迟，握手机制需要每个时钟域中最多 3 个时钟。具体如图 20-2 所示：

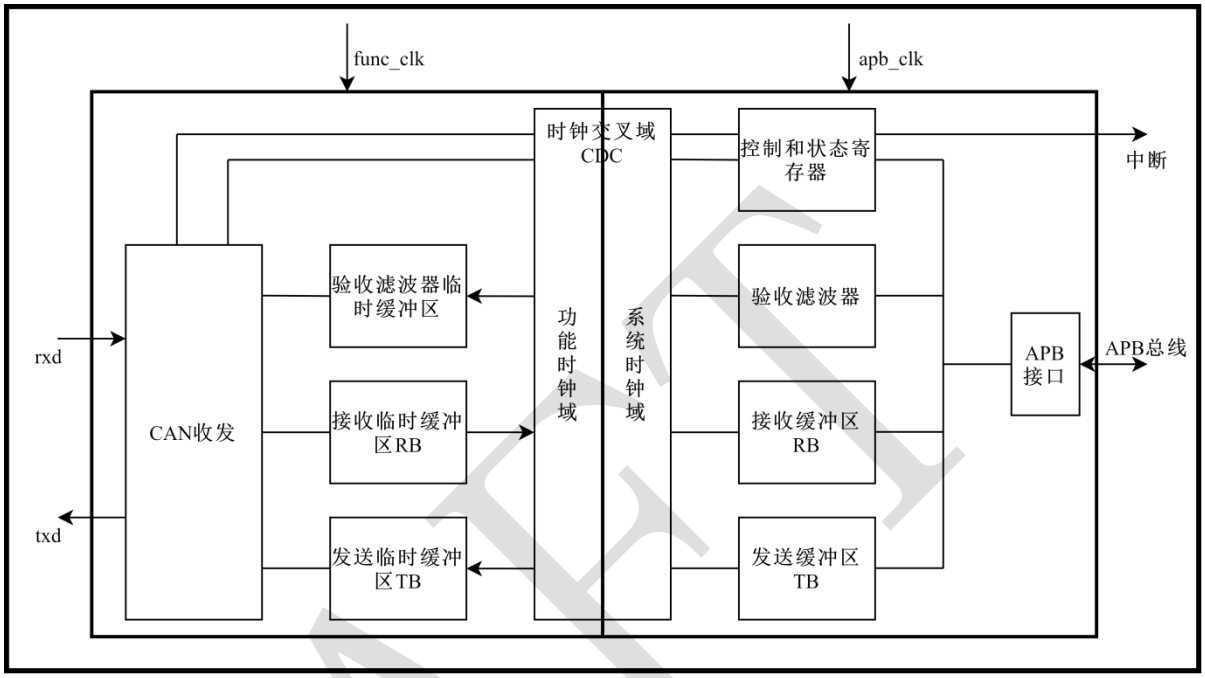


图 20-2 时钟域交叉示意图

20.4.2 CAN 位时间

20.4.2.1 标准 CAN 时序

图 20-3 为标准 CAN 的一个位时序

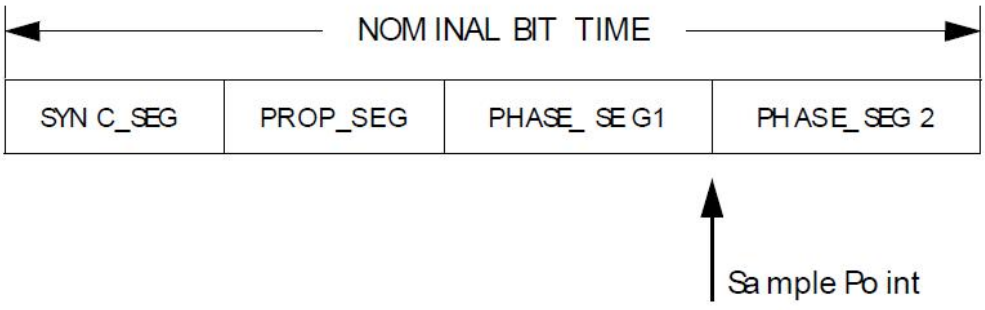


图 20-3 标准 CAN 的位时序组成

对于上图中的各个时间段，解释如表 20-2 所示

表 20-2 BIT 时序描述表

BIT 时序组成	描述	TQ 数
SYNC_SEG（同步段）	位时间的同步段用于同步总线上不同的节点。这一段内要有一个跳变沿	1
PROP_SEG（传播时间段）	传播段用于补偿网络内的物理延时时间。 它是总线上输入比较器延时和输出驱动器延时总和的两倍。	1~8
PHASE SEG1（相位缓冲段 1）	相位缓冲段用于补偿边沿阶段的错误。这两个段可以通过重新同步加长或缩短。	1~8
PHASE SEG2（相位缓冲段 2）		2~8
Sample Point（采样点）	采样点是读总线电平并解释各位的值的一个时间点。采集点位于相位缓冲段 1（PHASE_SEG1）之后。	-

20.4.2.2 CAN 时序配置

CAN 中的 BIT 时序参考的就是标准 BIT 时序，也就是如图 20-2 所示，但是 CAN 做了精简的配置，对于 SYNC_SEG、PROP_SEG 和 PHASE_SEG 这三个阶段都可以认为是采样前，而 PHASE_SEG2 属于采样后，于是在 IP 中 seg_1 的寄存器配置的就是 SYNC_SEG、PROP_SEG 和 PHASE_SEG1 三个阶段合起来的 TQ 数，seg_2 就是配置的 PHASE_SEG2 的 TQ 数。而 prescaler 寄存器配的就是 TQ 的周期长度，具体参考图 20-4 实例：

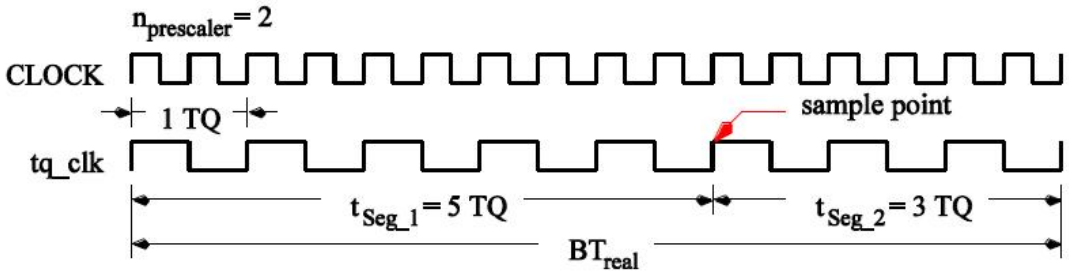


图 20-4 CAN 配置实例

上图是 CAN 配置的一个实例，对于 CAN 寄存器中 seg_1 和 seg_2 都有相应配置要求，具体如表 20-3 所示

表 20-3 CAN 时序设置

时序	协议要求	配置
t_{seg_1}	CAN 2.0 BIT：[3…17] TQ CAN FD 普通 BIT：[3…65] TQ CAN FD 数据 BIT：[2…17] TQ	t_{seg_1} = (S_SEG_1+2) TQ t_{seg_1} = (F_SEG_1+2) TQ
t_{seg_2}	CAN 2.0 BIT：[2…8] TQ （$t_{seg\_1} \geq t_{seg\_2}+2$） CAN FD 普通 BIT：[2…32] TQ（$t_{seg\_1} \geq t_{seg\_2}+2$） CAN FD 数据 BIT：[2…8] TQ（$t_{seg\_1} \geq t_{seg\_2}+1$）	tseg_2 = (S_SEG_2+1) TQ tseg_2 = (F_SEG_2+1) TQ
t_{SJW}	CAN 2.0 BIT：[1…4] TQ （$t_{seg\_2} \geq t_{seg\_2}$） CAN FD 普通 BIT：[1…16] TQ（$t_{seg\_2} \geq t_{SJW}$） CAN FD 数据 BIT：[1…4] TQ（$t_{seg\_2} \geq t_{SJW}$）	t_{SJW} = (S_SJW+1) TQ t_{SJW} = (F_SJW+1) TQ

TQ 的配置如下公式所示：

$$TQ = \frac{n_{prescaler}}{f_{clock}}$$

$$n_{prescaler} = S\_PRESC + 1$$

$$n_{prescaler} = F\_PRESC + 1$$

再看图 20-3 的实例配置，是一个 16MHz 的时钟，分出总线速率为 1MHz 的配置：

1. **第一步，确定 $n_{prescaler}$ 和 nTQ 数据。**
   16MHz 分频 1MHz 的总线速率，首先要考虑选择合适的分频值（S_SPRESC 和 F_PRESC）来确定比较理想的 Btreal（t_{seg_1}+t_{seg_2}），在这个实例中 $n_{prescaler}$=2，总的 TQ 数（nTQ）为 8 作为最理想的匹配，如果有 FD 的时序，则需要额外配置 F_开头的寄存器，否则只需要配置好 S_开头的寄存器。
2. **第二步，确定 t_{seg_1} 和 t_{seg_2} 的值。**
   由第一步得知，nTQ=8，根据上表的限制 tseg_1≥tseg_2+2，确定了 tseg_1=5，tseg_2=3
3. **第三步，配置寄存器 S_PRESC、S_SEG_1 和 S_SEG_2 或者 F_PRESC、F_SEG_1 和 F_SEG_2。**
   1） 当 $n_{prescaler}$=2，S_PRESC=1 或者 F_PRESC=1；
   2） 当 tseg_1=5，S_SEG_1=3 或者 F_SEG_1=3；
   3） 当 tseg_2=3，S_SEG_2=2 或者 F_SEG_2=2；
4. **第四步，配置寄存器 S_SJW 或者 F_SJW。**
   t_{sjw} 是同步跳转宽度，一般建议配置为 S_SEG_2 或者 F_SEG_2 的一半是最合适。所以 S_{SJW}=1 或者 F_{SJW}=1；

20.4.3 验收滤波器（ACF）

为了减轻主机控制器接收到的帧的负载，CAN 使用了验收滤波器。CAN 在接收总线数据时会检查 ID 号。因此，每个验收滤波器的长度为 29 位。一共有 16 个独立的验收滤波器，每个独立的验收滤波器都有各自独立的开关，其中第 0 个验收滤波器默认是开启的。

只要报文通过了其中一个过滤器，则它将被接收，且报文数据将被存储到 RB 中，如果 RIE 置 1，则会产生 RIF 中断。如果不接收该报文，则不会产生 RIF 中断，不接收的消息将被丢弃，并被下一条报文覆盖。任何未接收的报文都不会覆盖已存储的有效消息。

接收码位 ACODE 寄存器表示的是 CAN 即将接收的完整匹配值，只有接收的 ID 号与配置的 ACODE 完全一致时，报文才会被接收。而屏蔽码位 AMASK 寄存器可以配置是否进行对应位检查，若某一位配置为 0，则接收的 ID 号对应的码位需要和 ACODE 对应位完全匹配才可以被接收；若某一位配置为 1，则接收的 ID 号对应的码位无论为何值都会被接收。

比如 ACODE 配置为 0x7ff，AMASK 寄存器配置为 0x123，那么硬件在接收 ID 的时候只要 ID=$(11x\_11x1\_11xx)_b$ 的都可以接收，其中 x 可以是 0 也可以是 1。
ID 号将与相应的接收码位 ACODE 进行如下比较：

- 标准帧：ID（10：0）和 ACODE（10：0）
- 扩展帧：ID（28：0）和 ACODE（28：0）

20.4.4 初始化操作

软件初始化的流程步骤如下所示：

1. 复位 CAN

通过对 CAN_CTRL.RESET 位写 1 来复位 CAN

2. 配置 CAN 总线位时间

通过配置 CAN_PRESC 寄存器和 CAN_BITTIME 寄存器来配置总线位时间，具体参考 CAN 位时间章节

3. 配置 CAN 的验收滤波器

可以通过配置 CAN_ACFCTRL 和 CAN_ACF 来配置各个验收滤波器，具体如下所示：

1) 配置 CAN_ACFCTRL.ACFADR 为 0，代表当前配置的是第 0 个验收滤波器；
2) 配置 CAN_ACFCTRL.SELMASK 为 0，代表当前配置的是 CODE
3) 配置 CAN_ACF.ACF_X 配置验收滤波器 0 的 CODE，CAN 在收到报文时，需要与配置的 CODE 值的 ID 号完全匹配才能被接收。
4) 配置 CAN_ACFCTRL.SELMASK 为 1，代表当前配置的是 MASK
5) 配置 CAN_ACF.ACF_X 配置验收滤波器 0 的 MASK，MASK 的哪一位为 1，代表 CAN 不会匹配那一位值。

其余的每个验收滤波器也可以通过上述 5 个步骤来配置。

4. 启动传输

通过对 CAN_CTRL.RESET 位写 0 来启动 CAN 传输

20.4.5 接收报文

CAN 报文会经过验收滤波器进行比较，如果接收到的报文 ID 和其中任意一个验收滤波器的过滤配置相匹配，那么它将会被存储在接收缓冲区（RB）中，接收缓冲区具有类似 FIFO 的行为，若 RB 存满了，读取 RB 总是读取最旧的数据。它最大支持存储 1 帧 CAN 报文。

RB 缓冲区拥有丰富的中断和状态设置，可以帮助用户知道 RB 的状态，根据中断状态位可以及时读取 CAN 报文，具体如下表所示：

中断状态名	清除方式	描述
RIF	对该位写 1 清 0	如果该位置 1，则代表 CAN 接收到了一帧有效的报文并且已经存入 RB 中。此时 RB 中最少有一帧可读的 CAN 报文。
ROIF	对该位写 1 清 0	如果该位置 1，则代表 RB 在已满的情况下，CAN 接收到了一帧有效的报文。
RFIF	对该位写 1 清 0	如果该位置 1，则代表 RB 已经被填满
RAFIF	对该位写 1 清 0	如果该位置 1，则代表 RB 中的 CAN 报文数大于或者等于 AFWL 配置的值

对于 ROIF 中断，如果 RB 已满，总线上收到的新的 CAN 报文会被临时存储，如果在第 6 个 EOF 位起来之前 RB 中还没有额外空间，那么最早的报文会被最新的报文覆盖，且会产生 ROIF 中断。如果在 RB 已满且新的报文被接收前将 CAN_CTRL.RREL 置 1，则不会覆盖任何报文，也不会产生 ROIF 中断，但最早的报文将被丢弃。

RB 中包含了很多的空间，读取 RB 获得 CAN 报文的最佳步骤如下：

1. 根据 RB 的中断状态位来读取 RBUF 寄存器从 RB 中读取最早的报文
2. 若当前报文已经读取完毕想要释放并且读取下一条报文时，对 CAN_CTRL.RREL 写 1 更新 RB，同时也会更新 CAN_STATUS.RSTAT 状态，此时再读取 RBUF 将会读取新的报文。
3. 重复第一步和第二步操作，直到 CAN_STATUS.RSTAT 状态位指示 RB 为空为止。

20.4.6 发送报文

在开始任何传输之前，至少一个传输缓冲区（PTB 或 STB）必须装有一条完整报文。TPE 状态指示 PTB 是否锁定，TSSTAT 状态指示当前 STB 的填充状态。

TBUF 寄存器提供对 PTB 和 STB 的访问。以下是推荐的编程流程：

1. 将 TBSEL 配置为所需值以选择 PTB 或 STB。
2. 将帧写入 TBUF 寄存器。
3. 对于 STB，将 TSNEXT 置 1 完成 STB 当前 CAN 报文的加载。
4. 对于 PTB，将 TPE 置 1 完成 PTB 当前 CAN 报文的加载

CAN 2.0 协议定义的报文最大有效载荷长度为 8 字节，而 CANFD 协议定义的报文最大有效载荷长度为 64 字节。每条报文的单独长度由 DLC 定义。对于远程帧（RTR 位为 1），DLC 变得毫无意义，因为远程帧的数据长度始终为 0 字节。软件将 TSNEXT 配置为 1 以跳到下一个 STB 插槽去填写数据。所有 TBUF 字节均可按任何顺序写入。

如果 TBSEL=0 选择了 PTB，则将 TSNEXT 置 1 是没有意义的。在这种情况下，TSNEXT 会自动清除，不会造成错误。使用 PTB 时，应将 TPE 位置 1 以开始发送。要使用 STB，必须将 TSONE 置 1 指示 CAN 传输单个消息（最旧的消息），或者将 TSALL 置 1 指示 CAN 传输所有消息。

PTB 始终具有比 STB 高的优先级。如果两个发送缓冲区都有数据，则无论帧标识符如何，始终将先发送 PTB 报文。如果来自 STB 的传输已经处于发送状态，则 CAN 的下一个传输将发送来自 PTB 的报文。 PTB 传输完成或中止之后，CAN 才会返回处理来自 STB 的其他被挂起的报文。

传输完成后，有以下中断可能会置起：

- 对于 PTB，如果 TPIE 置 1，则 TPIF 中断会置 1。
- 对于使用 TSONE 的 STB，如果完成了一条消息与此同时 TSIE 为 1，则 TSIF 中断会置 1。
- 对于使用 TSALL 的 STB，如果所有消息均已完成与此同时 TSIE 为 1，则 TSIF 中断会置 1。换句话说：如果 STB 为空，则 TSIF 中断会置 1。因此，如果在开始 TSALL 传输之后，软件将附加报文写入 STB，则 TSIF 中断置 1 之前，还将发送新加载的附加消息。

STB 为空时将 TSONE 或 TSALL 置 1 是没有意义的。在这种情况下，TSONE 或 TSALL 将自动重置。不会起任何中断标志，也不会发送任何帧。

20.4.6.1 传输中止

如果出现这种情况，即由于其低优先级而无法发送 TX 缓冲区中的报文，这将长时间阻塞缓冲区。为了避免这种情况，如果尚未开始传输，则软件可以通过分别将 TPA 置 1 或者将 TSA 置

1 来撤回传输请求。

TPA 和 TSA 都提供一个中断标志：AIF。CAN 仅在不向 CAN 总线传输任何内容时才执行中止操作。因此，以下内容是有效的：

- 总线仲裁期间不中止。
  - 如果节点丢失仲裁，则中止将在之后执行。
  - 如果节点赢得仲裁，则将发送该帧。
- 传输帧时不中止。
  - 如果成功发送了帧，则将成功发送信号通知主机控制器。在这种情况下，不会发出中止的信号。这是通过相应的中断和状态位完成的。
  - 在 CAN 节点未收到确认的传输失败后，错误计数器将递增，并执行中止操作。
  - 如果主机已命令发送所有帧（TSALL = 1），则 STB 中至少剩余一帧，则已完成的帧和中止都将信号发送给主机。

由于上述两种情况，中止传输可能需要一些时间，具体取决于 CAN 通信速度和帧长度。如果执行中止，则将导致以下操作：

- TPA 释放 PTB，导致 TPE = 0。释放 PTB 之后，帧数据仍存储在 PTB 中。
- TSA 释放一个（最旧的）消息插槽或 STB FIFO 的所有消息插槽。这取决于是使用 TSONE 还是 TSALL 来开始传输。TSSTAT 将相应更新。

在 STB 中释放帧会导致丢弃帧数据，因为主机由于类似 FIFO 的行为而无法访问它。

不建议同时将 TPA 和 TSA 置 1。如果软件执意执行此操作，则 AIF 将会被置 1，并且 PTB 和 STB 的传输都将中止。为了避免上述欠款，用户可以配置以下中断组合：

- AIF（一次用于 PTB 和 STB 传输中止）
- TPIF + AIF
- TSIF + AIF
- TPIF + TSIF（很少，仅当主机不立即处理 TPIF 时才会发生）
- TPIF + TSIF + AIF（很少，只有在主机不立即处理 TPIF 和 TSIF 的情况下才会发生）

要清除整个 STB，需要同时将 TSALL 和 TSA 都置 1。为了检测是否由于丢失仲裁而无法长时间发送消息，主机可以使用 ALIF/ALIE。

20.4.6.2 STB 填满

首先将 TSNEXT 置 1，当报文写入 STB 并且填满一个缓冲区槽时，会自动将 BUFF 写指针定位到下一个空闲报文槽，当前缓冲区槽填满后，硬件会将 TSNEXT 自动清 0。但是如果当前报文填入的是 STB 的最后一个缓冲区槽时，也就是说该报文填入之后，STB 被填满了，此时 TSNEXT 将不会被清 0，直到有新的缓冲区槽可用才会被清 0，在 TSNEXT=1 期间，会禁止任何新的报文写入 TBUF。

20.4.7 扩展状态和错误报告

在 CAN 总线通信期间，可能会发生传输错误。以下功能支持对其进行检测和分析。

20.4.7.1 总线关闭状态

寄存器 STATUS 中的状态位 BUSOFF 指示总线关闭状态。如果 CAN 节点的传输错误计数器大于 255，则会自动进入总线关闭状态。然后，它将不再参与进一步的通信，直到它重新返回到主动错误状态。如果 EIE 置 1，BUSOFF 置 1 的同时也会产生 EIF 中断。如果 CAN 节点被软件复位或接收到 128 个 11 位隐性位序列（恢复序列），则它将返回到主动错误状态。

在总线关闭状态下，RECNT 用于计数恢复序列，而 TECNT 保持不变。如果节点从总线关闭状态恢复，则 RECNT 和 TECNT 将自动重置为 0。

20.4.7.2 可错误编程警告极限

RECNT 和 TECNT 对接收/发送期间的错误进行计数。主机控制器可使用可编程错误警告限制 EWL，对错误事件数量进行限制。可以从 8 到 128 的 8 个误差步长中选择极限值：

错误计数限制=（EWL + 1）* 8

如果在以下情况下如果 EIE 中断置 1，则中断 EIF 将置 1：

- RECNT 或 TECNT 已经计数超过了上述计算的“错误计数限制”
- BUSOFF 位已在任一方向上更改。

20.4.7.3 仲裁丢失捕获（ALC）

CAN 能够检测仲裁丢失的仲裁字段中的确切位位置。如果 ALIE 置 1，则可以通过 ALIF 中断来产生信号给用户。如果节点能够赢得仲裁，则 ALC 的值保持不变。然后，ALC 保留最后一次仲裁失败的旧值。

ALC 的值定义如下：帧从 SOF 位开始，然后发送 ID 的第一位。该第一 ID 位的 ALC 值为 0，第二 ID 位的 ALC 值为 1，依此类推。仅在仲裁字段中允许仲裁。因此，ALC 的最大值为 31，这是扩展帧中的 RTR 位。

注意：如果标准远程帧与扩展帧进行仲裁，则扩展帧将在 IDE 位丢失仲裁，ALC 将为 12。

20.4.7.4 错误种类（KOER）

CAN 识别总线上的错误，并将最后的错误事件存储在 KOER 位中。如果使能了 BEIF 中断，则可以指示 CAN 总线错误。每个新的错误事件都会覆盖 KOER 的先前存储的值。因此，主机控制器必须对错误事件做出快速反应。成功发送或接收帧后，将重置 KOER。

20.4.8 扩展功能

20.4.8.1 单发传输

默认情况下，CAN 在发送失败之后会进行自动重发，如果不需要自动重发，只想发送一次，那么可以通过将 PTB 的 TPSS 位和 STB 的 TSSS 分别配置为仅发送一次消息。在这种情况下，如果选定的传输处于活动状态，则在发生错误或仲裁丢失的情况下，将不会执行任何重新传输。

成功传输的情况下，与正常传输没有区别。但是，如果在传输过程中发生错误，则将 TPIF 置 1（如果 TPIE 置 1），则会清除相应的传输缓冲区，并更新 KOER 和错误计数器。因此，对于单发传输，仅 TPIF 不能指示帧是否已成功传输或发生了错误。

如果将单发传输与 TSALL 一起使用，并且 STB 中有一个以上的帧，则对于每一帧都将进行单发传输。无论是否成功传输了任何帧（例如由于 ACK 错误），CAN 都会切换到下一个帧，如果 STB 为空则停止。

因此，在这种情况下只有错误计数器指示发生了什么。这可能非常复杂，因为如果两个帧之一出现错误，主机将无法检测到哪一个是成功的。

20.4.8.2 仅收听模式（LOM）

LOM 提供了监视 CAN 总线的能力，而对总线没有任何影响。

另一个应用是自动比特率检测，其中主机控制器尝试不同的时序配置，直到没有错误发生为止。

在 LOM 中，CAN 无法将显性位写入总线（没有活动的错误标志或过载标志，没有 ACK），但是这些显性位在内部环回模式下在内部重新回环，以便内核接收自己的位。LOM 模式会监听总线上的错误，检测到的错误会反馈到 KOER 寄存器位。

对于 LOM 模式：

- 无论任何错误情况，错误计数器均保持不变。
- 如果一个节点发送帧，则当至少一个附加节点连接到不在 LOM 中的总线时，才会生成 ACK。这样就不会有错误，并且所有节点（包括 LOM 中的那些节点）都将接收该帧。

如果除发送节点之外的所有节点都在 LOM 中，则总线上将没有 ACK，并且发送器将生成 ACK 错误，并且可能会重新发送帧。因为仅当总线是隐性的，直到 ACK 之后帧比特的第 6 个末尾才接收到帧，因此错误帧将违反此规则，因此 LOM 中的节点不会接收到该帧。

CAN 处于正常传输模式时，不应该使能 LOM 模式。如果使能了 LOM 模式，没有来自 CAN 的附加保护，同时也无法开始传输。

20.4.8.3 总线连接测试

要检查节点是否连接到总线，必须执行以下步骤：

- 传送框架。如果节点连接到总线，则其 TX 位在其 RX 输入上可见。
- 如果还有其他节点连接到 CAN 总线，则期望传输成功（包括来自其他节点的 ACK），没有错误信号。
- 如果该节点是连接到 CAN 总线的唯一节点（但是总线，收发器和 CAN 之间的连接很好），则第一个常规错误情况发生在 ACK 插槽中，因为没有来自用户的确认其他节点。如果 BEIE 置 1 且 KOER =“100”（ACK 错误），则将产生 BEIF 错误中断。
- 如果与收发器或总线的连接断开，则在 SOF 位之后硬件会产生 BEIF 错误中断，并且 KOER =“001”（位错误）。

环回模式（LBMI 和 LBME）

CAN 支持两种环回模式：内部（LBMI）和外部（LBME）。两种模式都导致接收到自己发送的帧，这对于自检很有用。

在 LBMI 中，CAN 与和总线断开连接，并且 TXD 输出设置为隐性。输出数据流将反馈到输入。

在 LBME 中，CAN 保持与收发器的连接，并且已传输的帧在总线上可见。在这种模式下，CAN 接收自己的帧。是否存在来自另一个节点的 ACK 无关紧要。

在环回模式下，CAN 接收自己的消息，将其存储在 RBUF 中，并置位相应的接收中断标志。在回送模式传输期间，将禁用 ACK 错误生成。

LBMI 对于芯片内部和软件测试很有用，而 LBME 可以测试收发器及其连接。

传输处于活动状态时，软件不应同时激活两种环回模式。

如果该节点已连接到 CAN 总线，则不能仅通过将 LBMI 置 0 来完成从 LBMI 切换回正常传输模式，因为这可能会影响其他 CAN 节点的传输。在这种情况下，应通过将 RESET 位置 1 来切换回正常工作。RESET 位置 1 后将自动将 LBMI 清零。最后，可以将 RESET 置 0，CAN 将返回正常工作状态。LBME 则不受限制。

20.4.9 软件复位

如果 RESET 位置 1，则软件复位有效。 如果 RESET = 1，大部分寄存器将被强制进入复位状态，但是 RESET 不会复位所有寄存器。某些寄存器仅对可以被硬件复位复位。对于软件和硬件复位，所有位的复位值相同。

RESET 不会复位 BITTIME、PRESC、错误计数（RECNT 和 TECNT）、中断使能、ACF 相关的寄存器，RESET 后，所有的 RB 都会被清空，所有的 PTB 和 STB 都会被清空，TBUF 会指向 PTB，总线上若正处于接收状态，则会被立即取消，并且不会产生 ACK；若正处于发送状态，则直接终止，这将导致错误的帧，其他的节点能检测到。

20.4.10 CAN 中断

在 CAN 通信过程中，不同的事件会产生不同的中断，有关 CAN 中断请求的详细说明如表 20-4 所示

表 20-4 CAN 中断表

中断名称	中断使能	置起方式	清除方式
AIF	-	报文传输中止	对 AIF 位写 1 清 0
EIF	EIE	错误数量超过了 EWL 的值	对 EIF 位写 1 清 0
TSIF	TSIE	STB 已经传输完毕	对 TSIF 位写 1 清 0
TPIF	TPIE	PTB 已经传输完毕	对 TPIF 位写 1 清 0
RAFIF	RAFIE	STB 数据量大于等于 AFWL 的值	对 RAFIF 位写 1 清 0
RFIF	RFIE	RB 接收 buffer 已满	对 RFIF 位写 1 清 0
ROIF	ROIE	RB 至少有一个接收的报文被覆盖	对 ROIF 位写 1 清 0
RIF	RIE	接收到数据或者远程帧，当前 RB 有效可读取	对 RIF 位写 1 清 0
BEIF	BEIE	发生总线错误	对 BEIF 位写 1 清 0
ALIF	ALIE	发生冲裁丢失	对 ALIF 位写 1 清 0
EPIF	EPIE	发生被动错误	对 EPIF 位写 1 清 0

20.5 寄存器描述

20.5.1 寄存器列表

CAN 基地址：

CAN0(0x40014000) CAN1(0x40015000)

偏移	实例地址	名称	默认值	描述
0x00	CAN 基地址+0x00	CAN_CTRL	0x1B000010	CAN 控制寄存器
0x04	CAN 基地址+0x04	CAN_STATUS	0x00000000	CAN 状态寄存器
0x08	CAN 基地址+0x08	CAN_INTREN	0x00000000	CAN 中断使能寄存器
0x0C	CAN 基地址+0x0C	CAN_INTRST	0x00000000	CAN 中断状态寄存器
0x10	CAN 基地址+0x10	CAN_BITTIME	0x00324283	CAN 位定时段寄存器
0x14	CAN 基地址+0x14	CAN_PRESC	0x00000101	CAN 预分频寄存器
0x18	CAN 基地址+0x18	CAN_ERRST	0x00000000	CAN 错误状态寄存器
0x30	CAN 基地址+0x30	CAN_ACFEN	0x00000001	CAN ACF 使能寄存器
0x34	CAN 基地址+0x34	CAN_ACFCTRL	0x00000000	CAN ACF 控制寄存器
0x38	CAN 基地址+0x38	CAN_ACF	0x00000000	CAN ACF 数据寄存器
0x40	CAN 基地址+0x40	CAN_RBUFID	0x00000000	CAN ID 读取寄存器
0x44	CAN 基地址+0x44	CAN_RBUFCR	0x00000000	CAN 控制信息读取寄存器
0x48	CAN 基地址+0x48	CAN_RBUFDTx	0x00000000	CAN 数据读取寄存器
0x88	CAN 基地址+0x88	CAN_TBUFID	0x00000000	CAN ID 写入寄存器
0x8C	CAN 基地址+0x8C	CAN_TBUFCR	0x00000000	CAN 控制信息写入寄存器
0x90	CAN 基地址+0x90	CAN_TBUFDTx	0x00000000	CAN 数据写入寄存器

20.5.2 寄存器描述

20.5.2.1 CAN 控制寄存器（CAN_CTRL）

- 名称：CAN Control Register
- 偏移地址：0x00
- 默认值：0x1B000010
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFWL				EWL							RREL		FD_EN	FD_ISO	TSNEXT
R/W				R/W							R/W		R/W	R/W	R/W
0x1				0xB							0x00		0x00	0x00	0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MUXSEL	TBSEL	LOM		TPE	TPA	TSONE	TSALL	TSA	RESET	LBME	LBMI	TPSS	TSSS
		R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		0x00	0x00	0x0		0x0	0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x0

字段	说明
[31:28] AFWL	**接收缓冲区几乎满限制（RB Almost Full Warning Limit）** AFWL 是接收缓冲区几乎满（RAFIF 中断）限制，其中可用 RB 报文的最大数量为 1。 将 AFWL 与已填充的 RB 报文数进行比较，如果相等，则触发 RAFIF。 Note：AFWL > 1 是没有意义的，会自动视为 1。 Note：AFWL = 1 是有效值，但请注意，此时 RFIF 也会置起。
[27:24] EWL	**错误警告极限（Error Warning Limit）** 可编程错误警告极限=（EWL + 1）* 8。可能的极限值：8，16，…128。EWL 的值控制 EIF 中断产生。
[20] RREL	**接收缓冲区释放（Receive Buffer Release）** 软件清空当前的 RB 插槽。然后，CAN 指向下一个 RB 插槽。RSTAT 更新。 0：无 1：清空当前的 RB 插槽。
[18] FD_EN	**CAN FD 使能（CAN FD Enable）** 0：不使能 1：使能
[17] FD_ISO	**CAN FD ISO 使能（CAN FD ISO Enable）** 0：FD non-ISO 使能 1：FD ISO 使能
[16] TSNEXT	**发送缓冲区下一个使能（Transmit Buffer Secondary Next）** 0：无动作 1：STB 当前 FIFO 插槽已满，选择下一个 FIFO 插槽（一个插槽可以存一帧 CAN 数据帧）。 将所有帧字节写入 TBUF 寄存器后，软件必须将 TSNEXT 设置为 1，表明当前插槽已满，CAN 将 TBUF 寄存器跳转到下一个 FIFO 插槽指针。一旦将插槽标记为已填充，就可以使用 TSONE 或 TSALL 开始传输。可以在一次写访问中一起设置 TSNEXT 和 TSONE 或 TSALL。 Note：TSNEXT 必须由软件置 1，并在置 1 后，CAN 硬件成功跳转指针后会自动清 0。 Note：仅当 TBSEL=1 选择了 STB 时，设置 TSNEXT 才有意义。如果 TBSEL = 0，此时将 TSNEXT 置 1 会被忽略并自动清除。读写该位将不会产生任何影响。 Note：如果 STB 所有插槽均已填满，则 TSNEXT 保持等于 1 的状态，直到有一个插槽可用为止。
[13] MUXSEL	**CAN 输入信号通路选择（Receive Multiplexer Select）** 0：CAN0 接收 CAN0_RX 信号(仅 CAN0 配置有效)/CAN1 接收 CAN1_RX 信号（仅 CAN1 配置有效） 1：CAN0 接收 CAN1_RX 信号(仅 CAN0 配置有效)/CAN1 接收 CAN0_RX 信号（仅 CAN1 配置有效） Note：该位写 1 时，应启用仅收听模式 LOM，否则无法正常接收总线数据。
[12] TBSEL	**发送缓冲器选择（Transmit Buffer Select）** 选择要加载消息的发送缓冲区。使用 TBUF 寄存器进行访问。写 TBUF 寄存器以及设置 TSNEXT 时 TBSEL 软件必须保持稳定不能被更改，否则可能会导致填充数据失败。 0：PTB（高优先级缓冲区）

	1：STB（类似于 FIFO）
[11] LOM	**仅收听模式（Listen Only Mode）** 0：禁用 1：已启用 Note：传输处于活动状态时，不应启用 LOM。如果启用了 LOM，则无法开始传输。
[9] TPE	**PTB 传输启用（PTB Transceiver Enable）** 0：PTB 无传输 1：高优先级 PTB 传输使能 如果设置了 TPE，则来自 PTB 的消息将在下一个可能的发送位置发送。STB 开始的传输将会继续传输，但是待发送的新消息会延迟到 PTB 消息传输完成之后。 Note：TPE 会保持等于 1 的状态，直到成功发送了消息或使用 TPA 中止消息为止。 Note：如果 RESET = 1，BUSOFF = 1 或 LOM = 1，则该位将重置为 0。
[8] TPA	**PTB 传送中止（PTB Transmit Abort）** 0：不中止 1：中止 TPE = 1 请求并且停止来自 PTB 尚未开始的传输。（消息的数据字节保留在 PTB 中。） Note：该位只能写 1，且 CAN 硬件自清 0。任何时候都不建议 TPA 与 TPE 同时置 1。 Note：如果 RESET = 1 或 BUSOFF = 1，则该位将重置为 0。
[7] TSONE	**STB 传输下一帧（STB Transmit One Frame）** 0：STB 没有传输。 1：启用 STB 中最早的消息的传输。一旦总线空闲并且没有待处理的 PTB（位 TPE）请求，控制器就开始传输。 Note：该位只能写 1，一帧传输完成后且硬件自清 0。也可以通过对 TSA 写 1 来清除该位。 Note：如果 RESET = 1，BUSOFF = 1 或 LOM = 1，则该位将重置为 0。
[6] TSALL	**STB 传输所有帧（STB Transmit All Frames）** 0：STB 没有传输。 1：启用 STB 中所有消息的传输。一旦总线空闲并且没有待处理的 PTB（位 TPE）请求，控制器就开始传输。 Note：该位只能写 1，传输结束且 STB 为空时硬件自清 0，如果 STB 一直保持非空的状态，该位不会被清 0。另外也可以通过对 TSA 写 1 来清除该位。 Note：如果 RESET = 1 或 BUSOFF = 1 或 STBY = 1 或 LOM = 1，则该位将重置为 0。
[5] TSA	**STB 传输中止（Transmit STB Abort）** 0：不中止 1：中止来自 STB 的已请求但尚未开始的传输。对于 TSONE 传输，仅中止一帧，而对于 TSALL 传输，所有帧均被中止。STB 中的所有帧数据都会被清空， TSSTAT 状态位也会更新。 Note：该位只能写 1，写 1 时，会取消 TSONE 或 TSALL 传送。任何时候都不建议 TSA 和 TSONE 或 TSALL 同时置 1。 Note：如果 RESET = 1 或 BUSOFF = 1，则该位将重置为 0。
[4] RESET	**CAN 复位控制位（RESET request bit）** 0：不发生复位 1：复位 CAN Note：仅当 RESET = 1 时才能修改节点配置（BITTIME，PRESC，ACF_EN 和 ACF）。如果节点进入总线关闭状态，则 RESET 会被硬件置 1。
[3] LBME	**外部的环回模式（External Loop Back）** 0：不使能 1：使能 Note：传输处于活动状态时，不应启用 LBME。
[2] LBMI	**内部的环回模式（Internal Loop Back）** 0：不使能 1：使能 Note：传输处于活动状态时，不应启用 LBMI。
[1] TPSS	**PTB 的单发传输模式（Transmission Primary Single Shot mode for PTB）** 0：不使能 1：使能
[0] TSSS	**STB 的单发传输模式（Transmission Secondary Single Shot mode for STB）** 0：不使能 1：使能

20.5.2.2 CAN 状态寄存器（CAN_STATUS）

- **名称：CAN Status Register**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													ROV	RSTAT	
													R	R	
													0x00	0x00	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TSSTAT										RACTIVE	TACTIVE	BUSOFF
			R										R	R	R
			0x00										0x0	0x0	0x0

字段	说明
[18] ROV	**接收缓冲区溢出（Receive Buffer Overflow）** 0：无溢出。 1：溢出。至少丢失一条消息。 Note：通过设置 RREL = 1 清除 ROV。
[17:16] RSTAT	**接收缓冲区状态（Receive Buffer Status）** 00：空 01：有数但数量小于 AFWL 设置的值 10：数据量大于等于 AFWL 设置的值但是此时没有满 11：满
[12:8] TSSTAT	**第二传输 buffer 状态位（Transmission Secondary Status bits）** 第二传输 buffer（STB）的状态位，表示已填满消息的 buffer 的数量（0…16）
[2] RACTIVE	**接收状态位（Receive Status bit）** 0：不进行接收 1：控制器正在接收数据帧或远程帧
[1] TACTIVE	**发送状态位（Transmit Status bit）** 0：不进行发送 1：控制器正在发送数据帧或远程帧
[0] BUSOFF	**总线启动状态位（Bus Status bit）** 0：总线启动 1：总线关闭

20.5.2.3 CAN 中断使能寄存器（CAN_INTREN）

- 名称：CAN Interrupt Enable Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					EPIE	ALIE	BEIE	RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF
					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[10] EPIE	**被动错误中断使能位（Error Passive Interrupt Enable）** 0：不使能 1：使能
[9] ALIE	**仲裁丢失中断使能位（Arbitration Lost Interrupt Enable）** 0：不使能 1：使能
[8] BEIE	**总线错误中断使能位（Bus Error Interrupt Enable）** 0：不使能 1：使能.
[7] RIE	**中断接收使能位（Receive Interrupt Enable）** 0：不使能 1：使能
[6] ROIE	**接收 buffer 的溢出中断使能位（RB Overrun Interrupt Enable）** 0：不使能 1：使能
[5] RFIE	**接收 buffer 已满中断使能位（RB Full Interrupt Enable）** 0：不使能 1：使能
[4] RAFIE	**接收 buffer 将满中断使能位（RB Almost Full Interrupt Enable）** 0：不使能 1：使能
[3] TPIE	**PTB 传输中断使能位（PTB Transmission Interrupt Enable）** 0：不使能 1：使能
[2] TSIE	**STB 传输中断使能位（STB Transmission Interrupt Enable）** 0：不使能 1:：使能
[1] EIE	**出错中断使能位（Error Interrupt Enable）** 0：不使能 1：使能
[0] TSFF	**发送辅助缓冲区满标志（Transmit Secondary Buffer Full Flag）** 0：STB 未满 1：STB 满

20.5.2.4 CAN 中断状态寄存器（CAN_INTRST）

- 名称：CAN Interrupt Status Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														EWARN	EPASS
														R	R
														0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					EPIF	ALIF	BEIF	RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
					R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[17] EWARN	**达到错误警告限制（Error Warning Limit Reached）** 0：两个计数器中的值均小于 EWL 1：错误计数器 RECNT 或 TECNT 之一等于或大于 EWL
[16] EPASS	**错误被动模式有效（Error Passive Mode Active）** 0：不活动（节点错误活动） 1：主动（节点是错误被动）
[10] EPIF	**错误被动中断标志（Error Passive Interrupt Flag）** 如果错误状态从主动错误变为被动错误或者被动错误变为主动错误，该位会被置起。 Note：对该位写 1 可以清除该位。
[9] ALIF	**仲裁丢失中断标志（Arbitration Lost Interrupt Flag）** 0：仲裁没有丢失 1：仲裁丢失 Note：对该位写 1 可以清除该位。
[8] BEIF	**总线错误中断标志（Bus Error Interrupt Flag）** 0：没有检测到总线错误 1：检测到总线错误 Note：对该位写 1 可以清除该位。
[7] RIF	**接收中断标志（Receive Interrupt Flag）** 0：没有接收到帧。 1：数据或远程帧已被接收，并且在接收缓冲区中可用。 Note：对该位写 1 可以清除该位。
[6] ROIF	**RB 溢出中断标志（RB Overflow Interrupt Flag）** 0：没有 RB 被覆盖。 1：RB 中至少覆盖了一条接收到的消息。 Note：如果发生溢出，硬件会同时将 ROIF 和 RFIF 置 1。 Note：对该位写 1 可以清除该位。
[5] RFIF	**RB 满中断标志（RB Full Interrupt Flag）** 0：RB FIFO 未满。 1：所有 RB 已满。如果在接收到下一条有效消息之前没有释放任何 RB，则最早的消息将丢失。 Note：对该位写 1 可以清除该位。
[4] RAFIF	**RB 几乎满中断标志（RB Almost Full Interrupt Flag）** 0：已填充的 RB 插槽数小于 AFWL 配置的值 1：已填充的 RB 插槽数量大于等于 AFWL 配置的值 Note：对该位写 1 可以清除该位。
[3] TPIF	**PTB 传输中断标志（PTB Transmission Interrupt Flag）** 0：PTB 的传输尚未完成。 1：请求的 PTB 传输已成功完成。 Note：对该位写 1 可以清除该位。
[2] TSIF	**STB 传输中断标志（STB Transmission Interrupt Flag）** 0：STB 的传输尚未完成。

	1：请求的 STB 传输已成功完成。 Note：对该位写 1 可以清除该位。
[1] EIF	**错误中断标志（Error Interrupt Flag）** 0：不变。 1：错误警告限制的边界已沿任一方向越过，或者 BUSOFF 位已沿任一方向被改变。 Note：对该位写 1 可以清除该位。
[0] AIF	**中止中断标志（Abort Interrupt Flag）** 0：没有执行中止。 1：设置 TPA 或 TSA 后，相应的消息已中止。建议不要同时设置 TPA 和 TSA，因为这两个源都是 AIF。 Note：AIF 没有关联的使能寄存器。 Note：对该位写 1 可以清除该位。

20.5.2.5 CAN 位定时段寄存器（CAN_BITTIME）

- **名称：CAN Bittime Setting Register**
- **偏移地址：0x10**
- **默认值：0x00324283**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								F_SEG1				S_SJW			
								R/W				R/W			
								0x3				0x2			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_SEG2			S_SEG2					F_SJW		S_SEG1					
R/W			R/W					R/W		R/W					
0x2			0x2					0x2		0x3					

字段	说明
[23:20] F_SEG1	**快速模式下的位定时段 1（Bit Timing Segment 2 (fast speed)）** 启动采样前时段： tSeg_1=(Seg_1+2)*TQ Note：配置 Seg_1=0 无意义并且会被自动变为 2。
[19:16] S_SJW	**慢速模式下的同步跳转宽度（Bit Timing Segment 2 (slow speed)）** 同步跳转宽度，即 tSJW=(SJW+1)*TQ 是缩短或延长重新同步的位定时的最大时间，TQ 是时间量程。
[15:13] F_SEG2	**快速模式下的位定时段 2（Bit Timing Segment 2 (fast speed)）** 启动采样时段：Time tSeg_2=(Seg_2+1)*TQ Note：配置 Seg_2=0 无意义并且会被自动变为 1。
[12:8] S_SEG2	**慢速模式小的为定时段 2（Bit Timing Segment 2(slow speed)）** 启动采样时段：Time tSeg_2=(Seg_2+1)*TQ Note：Seg_2=0 无意义并且会被自动变为 1。
[7:6] F_SJW	**快速模式下的同步跳转宽度（Synchronization Jump Width (fast speed)）** 同步跳转宽度，即：tSJW = (SJW+1)*TQ 是缩短或延长重新同步位时间的最大时间，TQ 是时间量程。
[5:0] S_SEG1	**慢速模式下的位定时段 1（Bit Timing Segment 1 (slow speed)）** 启动采样前时段：tSeg_1=(Seg_1+2)*TQ Note：Seg_1=0 无意义并且会被自动变为 2。

20.5.2.6 CAN 预分频寄存器（CAN_PRESC）

- 名称：CAN Prescaler Registers
- 偏移地址：0x14
- 默认值：0x00000101
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								TDCEN			SSPOFF				
								R/W			R/W				
								0x0			0x0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_PRESC								S_PRESC							
R/W								R/W							
0x1								0x1							

字段	说明
[23] TDCEN	**发送延时补偿使能位（Transmitter Delay Compensation Enable）** 如果 TDCEN = 1 和 BRS 有效，则在 CAN FD 帧的数据周期间将激活发送延时补偿（TDC）。
[20:16] SSPOFF	**第二采样点偏移控制位（Secondary Sample Point OFFset）** 发送器延迟脉冲 SSPOFF 定义了 TDC 的第二采样点的时间。SSPOFF 表示 TQ 的数量。
[15:8] F_PRESC	**快速模式下的预分频（Prescaler (fast speed)）** 预分频器将系统时钟分频以获得子时钟 tq_clk。 PRESC=[0x01, 0xff]对应分频 2 到 256。 Note：不允许 PRESC=0，当 PRESC=0 时会自动将其视为 1。
[7:0] S_PRESC	**低速模式下的预分频（Prescaler (slow speed)）** 预分频器将系统时钟分频以获得子时钟 tq_clk。 PRESC=[0x01, 0xff]对应分频 2 到 256。 Note：不允许 PRESC=0，当 PRESC=0 时会自动将其视为 1。

20.5.2.7 CAN 错误状态寄存器（CAN_ERRST）

- 名称：CAN Error Status Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RECNT								TECNT							
R								R							
0x0								0x0							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								KOER			ALC				
								R			R				
								0x0			0x0				

字段	说明
[31:24] RECNT	**接收错误计数（number of errors during transmission）** 该位标志在接收数据期间发生的错误的个数。该位不会溢出，0xff＝255 时最大值。
[23:16] TECNT	**发送错误计数（number of errors during reception）** 该位表示发送数据期间产生的错误的个数。该位不会溢出，0xff＝255 时最大值。
[7:5] KOER	**错误类型标志位(Kind Of Error (Error code))** 000：没有发生错误 001：位错误 010：格式错误 011：位填充错误

	100：应答错误 101：CRC 错误 110：其他错误，自身错误标志后的显性位，接收到的有效错误标志太长，ACK 错误后的 Passive-Error-Flag 期间的显性位 111：不可用 Note：成功发送或接收 1 帧后该位清 0。
[4:0] ALC	**仲裁丢失捕获（Arbitration Lost Capture）** 仲裁丢失的帧的位位置。

20.5.2.8 CAN ACF 使能寄存器（CAN_ACFEN）

- **名称：CAN ACF Enable**
- **偏移地址：0x30**
- **默认值：0x00000001**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AE15	AE14	AE13	AE12	AE11	AE10	AE9	AE8	AE7	AE6	AE5	AE4	AE3	AE2	AE1	AE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x1

字段	说明
[15] AE15	**验收过滤器 15 启用（Acceptance filter 15 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[14] AE14	**验收过滤器 14 启用（Acceptance filter 14 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[13] AE13	**验收过滤器 13 启用（Acceptance filter 13 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[12] AE12	**验收过滤器 12 启用（Acceptance filter 12 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[11] AE11	**验收过滤器 11 启用（Acceptance filter 11 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[10] AE10	**验收过滤器 10 启用（Acceptance filter 10 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[9]	**验收过滤器 9 启用（Acceptance filter 9 Enable）**

AE9	1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[8] AE8	**验收过滤器 8 启用（Acceptance filter 8 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[7] AE7	**验收过滤器 7 启用（Acceptance filter 7 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[6] AE6	**验收过滤器 6 启用（Acceptance filter 6 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[5] AE5	**验收过滤器 5 启用（Acceptance filter 5 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[4] AE4	**验收过滤器 4 启用（Acceptance filter 4 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[3] AE3	**验收过滤器 3 启用（Acceptance filter 3 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[2] AE2	**验收过滤器 2 启用（Acceptance filter 2 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[1] AE1	**验收过滤器 1 启用（Acceptance filter 1 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。
[0] AE0	**验收过滤器 0 启用（Acceptance filter 0 Enable）** 1 - 启用接受过滤器 0 - 验收过滤器禁用 Note：每个验收过滤器（AMASK / ACODE）可以单独启用或禁用。 禁用的过滤器拒绝消息，只有启用的过滤器才能接受消息。

20.5.2.9 CAN ACF 控制寄存器（CAN_ACFCTRL）

- **名称：CAN ACF Control Register**
- **偏移地址：0x34**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SELMASK		ACFADR			
										R/W		R/W			
										0x0		0x0			

字段	说明
[5] SELMASK	**验收码和屏蔽码的选择位（Select acceptance MASK）** 0：ACF_x 寄存器指向验收 code 1：ACF_x 寄存器指向验收 mask Note：ACFADR 位选择一个具体的验收滤波器。
[3:0] ACFADR	**验收滤波器的地址（acceptance filter address）** 该位指向一个具体的验收滤波器。被选中的滤波器可以用 ACF_x 寄存器进行访问。SELMASK 在所选的验收滤波器的验收码或验收屏蔽之间选择。

20.5.2.10 CAN ACF 数据寄存器（CAN_ACF）

- **名称：CAN ACF Data Register**
- **偏移地址：0x38**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AIDEE	AIDE	ACF_X												
	R/W	R/W	R/W												
	0x0	0x0	0x0												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACF_X															
R/W															
0x0															

字段	说明
[30] AIDEE	**验收掩码 IDE 位的检查使能位（Acceptance mask IDE bit check enable）** 0：验收滤波器接收标准帧和扩展帧 1：验收滤波器接收由 AIDE 定义的格式，标准帧或扩展帧 Note：只有滤波器 0 受到上电复位的影响，其他的滤波器都不会被初始化。 Note：只有当 SELMASK=1 时才有效！
[29] AIDE	**验收掩码 IDE 位的控制位（Acceptance mask IDE bit value）** 当 AIDEE = 1 时： 0：验收滤波器只接收标准帧 1：验收滤波器只接收扩展帧 Note：只有滤波器 0 受到上电复位的影响，其他的滤波器都不会被初始化。 Note：只有当 SELMASK=1 时才有效！
[28-0] ACF_X	**当 SELMASK=0：** 验收滤波器 Code（Acceptance CODE）（ACF_X = ACODE_X） 0：ACC 位值与接收到的消息的 ID 位比较

1：ACC 位值与接收到的消息的 ID 位比较
ACODE_X(10:0)将用于扩展帧。
ACODE_X(28:0)将用于扩展帧。
只有滤波器 0 受上电复位影响。所有其他过滤器保持未初始化。
ACODE_X 受 ACFADR 影响，X=ACFADR

当 SELMASK=1：
验收滤波器屏蔽（Acceptance MASK）（ACF_X = AMASK_X）
0：接收标识符启用这些位的接受检查
1：禁用接收标识符的这些位的验收检查
AMASK_X(10:0)将用于扩展帧。
AMASK_X(28:0)将用于扩展帧。
禁用位导致接受消息。因此，过滤器 0 重置后的默认配置接受所有消息。
只有滤波器 0 受上电复位影响。所有其他过滤器保持未初始化。
AMASK_X 受 ACFADR 影响，X=ACFADR

20.5.2.11 CAN ID 读取寄存器（CAN_RBUFID）

- **名称：CAN RX Buffer Read Register(ID)**
- **偏移地址：0x40**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXD_ID															
R															
0x00															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXD_ID															
R															
0x00															

字段	说明
[31:0] RXD_ID	**CAN 接收的 ID 信息（CAN RX Identifier Data）** 31 ： ESI 30:29 ： Reserved 28:0 ： CAN ID

20.5.2.12 CAN 控制信息读取寄存器（CAN_RBUFCR）

- 名称：CAN RX Buffer Read Register(Control)
- 偏移地址：0x44
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RXD_CR				RXD_DLC			
								R				R			
								0x00				0x00			

字段	说明
[7:4] RXD_CR	**CAN 接收的控制信息（CAN RX Control Data）** 0000：CAN 标准数据帧 0100：CAN 标准远程帧 1000：CAN 扩展标准数据帧 1100：CAN 扩展标准远程帧 0010：CAN FD 数据帧 0011：CAN FD 加速数据帧 1010：CAN FD 扩展数据帧 1011：CAN FD 扩展加速数据帧 其他：Reserved
[3:0] RXD_DLC	**CAN 接收的数据量（CAN RX DATA Count）** CAN 协议中的 DLC 0000： 0 字节 0001： 1 字节 0010： 2 字节 0011： 3 字节 0100： 4 字节 0101： 5 字节 0110： 6 字节 0111： 7 字节 1000： 8 字节 1001： 12 字节 1010： 16 字节 1011： 20 字节 1100： 24 字节 1101： 32 字节 1110： 48 字节 1111： 64 字节

20.5.2.13 CAN 数据读取寄存器（CAN_RBUFDTx）

- 名称：CAN RX Buffer Read Register(Data)
- 偏移地址：0x48
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXD_BYTE3								RXD_BYTE2							
R								R							
0x00								0x00							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXD_BYTE1								RXD_BYTE0							
R								R							
0x00								0x00							

字段	说明
[31:24] RXD_BYTE3	**CAN 接收的数据字节 3（CAN RX Data Byte 3）** 读回来的第 4*N+3 个 data byte （N 的取值范围为 0~15)）
[23:16] RXD_BYTE2	**CAN 接收的数据字节 2（CAN RX Data Byte 2）** 读回来的第 4*N+2 个 data byte （N 的取值范围为 0~15)）
[15:8] RXD_BYTE1	**CAN 接收的数据字节 1（CAN RX Data Byte 1）** 读回来的第 4*N+1 个 data byte （N 的取值范围为 0~15)）
[7:0] RXD_BYTE0	**CAN 接收的数据字节 0（CAN RX Data Byte 0）** 读回来的第 4*N+0 个 data byte （N 的取值范围为 0~15)）

20.5.2.14 CAN ID 写入寄存器（CAN_TBUFID）

- 名称：CAN TX Buffer Write Register(ID)
- 偏移地址：0x88
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXD_ID															
W															
0x00															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXD_ID															
W															
0x00															

字段	说明
[31:0] TXD_ID	**CAN 发送的 ID 信息（CAN TX Identifier Data）** 31 ： ESI 30:29 ： Reserved 28:0 ： CAN ID

20.5.2.15 CAN 控制信息写入寄存器（CAN_TBUFCR）

- 名称：CAN TX Buffer Write Register(Control)
- 偏移地址：0x8C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TXD_CR				TXD_DLC			
								W				W			
								0x00				0x00			

字段	说明
[7:4] TXD_CR	**CAN 发送的控制信息（CAN TX Control Data）** 000x：CAN 标准数据帧 01xx：CAN 标准远程帧 100x：CAN 扩展标准数据帧 11xx：CAN 扩展标准远程帧 0010：CAN FD 数据帧 0011：CAN FD 加速数据帧 1010：CAN FD 扩展数据帧 1011：CAN FD 扩展加速数据帧 其他：Reserved
[3:0] TXD_DLC	**CAN 发送的数据量（CAN TX DATA Count）** CAN 协议中的 DLC 0000： 0 字节 0001： 1 字节 0010： 2 字节 0011： 3 字节 0100： 4 字节 0101： 5 字节 0110： 6 字节 0111： 7 字节 1000： 8 字节 1001： 12 字节（FD 帧才有效，如果是标准帧，为 8 字节） 1010： 16 字节（FD 帧才有效，如果是标准帧，为 8 字节） 1011： 20 字节（FD 帧才有效，如果是标准帧，为 8 字节） 1100： 24 字节（FD 帧才有效，如果是标准帧，为 8 字节） 1101： 32 字节（FD 帧才有效，如果是标准帧，为 8 字节） 1110： 48 字节（FD 帧才有效，如果是标准帧，为 8 字节） 1111： 64 字节（FD 帧才有效，如果是标准帧，为 8 字节）

20.5.2.16 CAN 数据写入寄存器（CAN_TBUFDTx）

- 名称：CAN TX Buffer Write Register(Data)
- 偏移地址：0x90
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXD_BYTE3								TXD_BYTE2							
W								W							
0x00								0x00							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXD_BYTE1								TXD_BYTE0							
W								W							
0x00								0x00							

字段	说明
[31:24] TXD_BYTE3	**CAN 发送的数据字节 3（CAN TX Data Byte 3）** 写入的第 4*N+3 个 data byte （N 的取值范围为 0~16)
[23:16] TXD_BYTE2	**CAN 发送的数据字节 2（CAN TX Data Byte 2）** 写入的第 4*N+2 个 data byte （N 的取值范围为 0~16)
[15:8] TXD_BYTE1	**CAN 发送的数据字节 1（CAN TX Data Byte 1）** 写入的第 4*N+1 个 data byte （N 的取值范围为 0~16)
[7:0] TXD_BYTE0	**CAN 发送的数据字节 0（CAN TX Data Byte 0）** 写入的第 4*N+0 个 data byte （N 的取值范围为 0~16)

21 通用串行总线（USB）

21.1 简介

USB 实现了全速 USB2.0 总线和 AHB 总线之间的接口。软件可以通过 AHB 总线对 USB 进行配置以及进行数据交互。

21.2 主要特性

- USB2.0 协议，支持全速模式（不支持高速和低速模式）
- 包括端点 0 一共有 3 个端点，都支持 RX 和 TX
- 256 bytes 的专用数据包缓冲存储器
- 循环冗余码校验（CRC）生成/校验，不归零反相（NRZI）编码/解码和位填充
- 支持控制和中断
- USB 插入/拔出检测
- USB 挂起/唤醒功能

21.3 结构框图

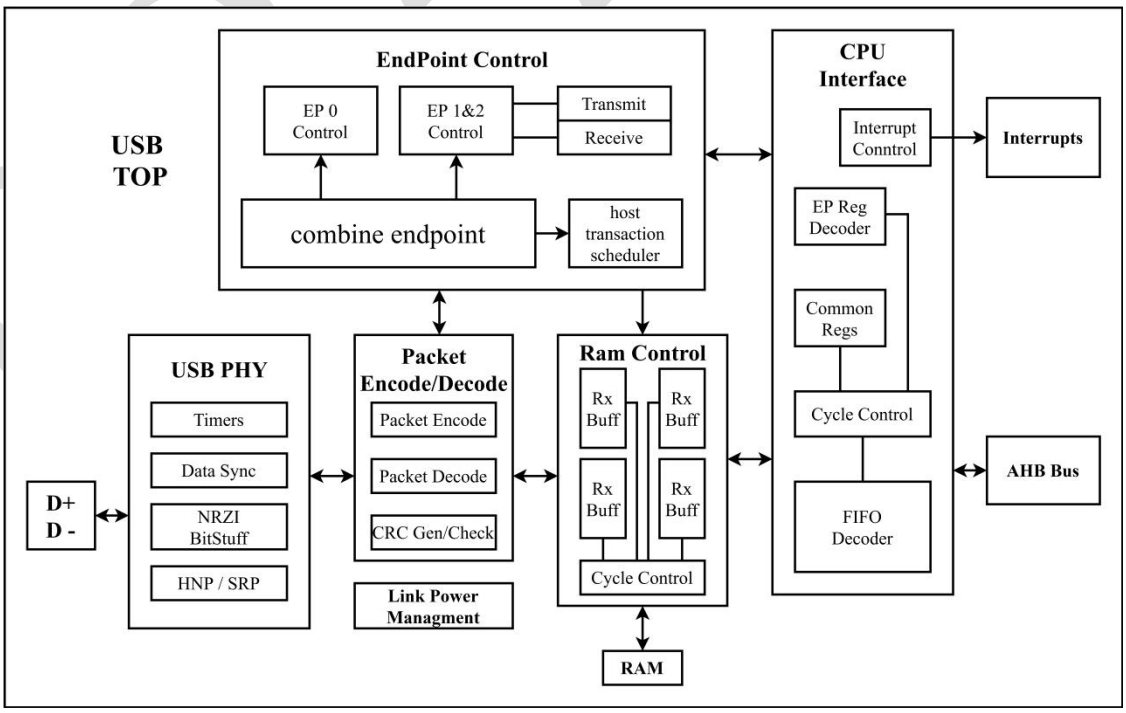


图 21-1 USB 连接框图

21.4 功能描述

21.4.1 插入拔出检测

USB2.0 协议中全速设备识别如图 21-2 所示：

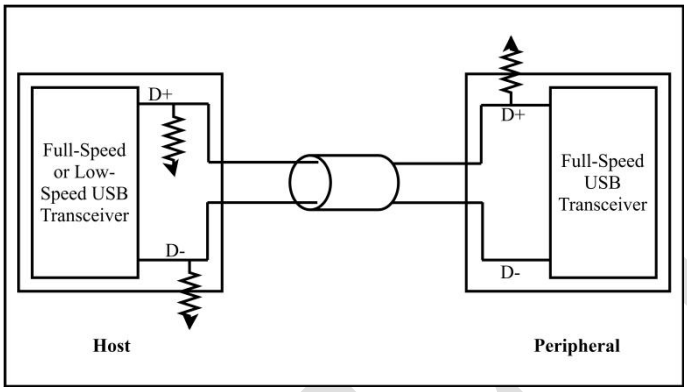


图 21-2　标准全速设备识别方式图

USB 作为主机时，会在 D+和 D-分别施加 15k 的下拉电阻；USB 作为从机时，会在 D+上施加一个 1.5k 的上拉电阻。当设备没有连入主机时，主机 D+和 D-都为 0，设备的 D+为 1，D-为高阻态；当设备连入主机时，主机 D+的 15k 下拉相对于设备 D+的 1.5K 上拉为弱下拉电阻，因此此时 D+为 1；设备的 D-悬空，因此 D-为 0。

如果采用图 21-2 所示的全速设备识别模式，那么作为设备来说在不插入时的 D-位不定态，无法确定 D-的值，而插入后 D-为 0，在没有 vbus 的前提下比较难去做一个设备检测是否插入主机，因此，在不影响主机的判断的情况下，在我们芯片对 D-做了一个弱上拉，芯片中的 USB 采用的是如图 21-3 所示的全速识别方式。

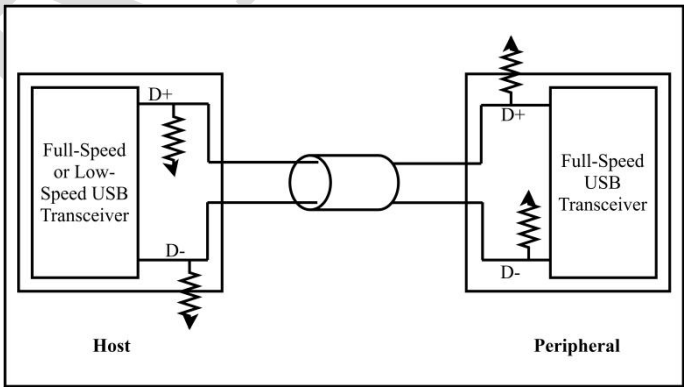


图 21-3　芯片全速识别方式图

在设备端加入 D-的 300k 上拉电阻，当设备没有连入主机时，主机 D+和 D-都为 0，设备的 D+为 1，D-不再为高阻态，而是 1；当设备连入主机时，主机 D+的 15k 下拉相对于设备 D+的 1.5k 上拉为弱下拉电阻，因此此时 D+为 1；主机 D-的 15k 下拉相对于设备 D-的 300k 上拉电阻为强下拉电阻，因此此时 D-为 0；

上述不会破坏主机的判断，同时从机也可以更好的做插入拔出检测，无插入时，设备的 D+和 D-都为 1；插入后，设备的 D+为 1 和 D-为 0；再拔出时，设备的 D+和 D-都为 1。插入检测

就是 D+和 D-全是 1 的状态转换到 D+和 D-不全是 1 的状态；拔出检测就是 D+和 D-不全是 1 的状态转换到 D+和 D-全是 1 的状态。

21.4.1.1 配置流程

1. 在检测 DP 和 DM 的沿变化时，需要对其进行去抖处理，需要配置 USB_IUINTREN 寄存器的 DETDB，滤除一些由于干扰而产生的 DP 和 DM 沿变化。
2. 开启插入检测中断使能（USB_IUINTREN 寄存器的 ISDETIEN）和拔出检测中断使能（USB_IUINTREN 寄存器的 UPDETIEN）
3. 此时会有两种情况
   - 检测到插入：此时 USB_IUINTR 的 ISDETI 会置起，要清除的话需要对 ISDETI 位写 1
   - 检测到拔出：此时 USB_IUINTR 的 UPDETI 会置起，要清除的话需要对 UPDETI 位写 1

21.4.2 USB 复位

当主机向 USB 发送 RESET 信号时，USB 会禁用所有的端点寄存器的通讯，仅仅会开放端点 0 寄存器的通讯；并且 USB_CTRL 寄存器的 FADDR 的设备地址会被清 0；此时会起一个 RESET 中断（USBINTR 寄存器）告知软件主机已经和 USB 设备连接成功了，可以着手准备枚举了。

21.4.3 IN 事务

当接收到 IN 令牌包时，如果接收到的地址与已配置的有效端点匹配，USB 将访问与寻址端点相关的缓冲区并获取数据。再次访问数据包存储器以读取要发送的第一个字节，并开始发送 DATA0 或 DATA1 PID。PID 完成后，将从缓冲存储器读取的第一个字节加载到输出移位寄存器中，以在 USB 总线上传输。发送完最后一个数据字节后，将发送计算出的 CRC。如果寻址的端点无效，则根据 USB_TXCTRL 寄存器中的 SENTSTA 位发送 NAK 或 STALL 握手数据包，而不是数据包。

21.4.4 OUT/SETUP 事务

在芯片设计中，OUT 和 SETUP 包以相同的方式处理；当接收到 OUT / SETUP PID 时，如果地址与有效端点匹配，USB 随后将接收到的数据字节打包为字节，然后传输到指定的数据包缓冲区，同时每收到一个字节 RXCOUNT 将会自增 1。当检测到 DATA 数据包的末尾时，将测试接收到的 CRC 的正确性，并且只有在接收过程中没有发生错误时，才会将 ACK 握手数据包发送回发送主机。

如果出现错误的 CRC 或其他类型的错误（位冲突，帧错误等），数据字节仍会复制到数据包存储缓冲区中，至少直到错误检测点为止，但不会发送 ACK 数据包。但是，在这种情况下，通常不需要采取任何软件操作：USB 外设从接收错误中恢复并为下一次握手做好准备。如果寻址的端点无效，则根据 USB_TXCTRL 寄存器中的 SENTSTA 位发送 NAK 或 STALL 握手数

据包，而不是 ACK，并且不将任何数据写入接收存储器缓冲区。

21.4.5 控制传输

端点 0 是核心的主要控制端点，需要软件来处理可能通过端点 0 发送或接收的所有标准设备请求。USB 外围设备收到的标准设备请求可以分为三类：零数据请求（命令中包括所有信息），写入请求（命令后跟其他数据）和读取请求（要求设备将数据发送回主机）。

21.4.5.1 中断生成

端点 0 中断生成：

- 接收到有效令牌并将数据写入 FIFO 后，内核将 RXPR 位（USB_TXCTRL.D16）置 1 时。
- 当 FIFO 中的数据包已成功发送到主机后，内核清除 TXPR 位（USB_TXCTRL.D17）。
- 由于协议冲突导致控制握手结束后，内核将 SENTSTA 位（USB_TXCTRL.D18）置 1 时。
- CPU 将 SETUPEND 位（USB_TXCTRL.D20）置 1 时，因为在设置 DATAEND（USB_TXCTRL.D10）之前控制传输已结束。

21.4.5.2 零数据请求

零数据请求的所有信息都包含在 8 字节命令中，不需要传输其他数据。“零数据”标准设备请求的示例包括：SET_FEATURE，CLEAR_FEATURE，SET_ADDRESS，SET_CONFIGURATION，SET_INTERFACE。

与所有请求一样，事件序列将在软件收到端点 0 中断时开始。RXPR 位（USB_TXCTRL.D16）也将被置位。然后应从端点 0 FIFO 中读取 8 字节命令，对其进行解码并采取适当的措施。例如，如果命令为 SET_ADDRESS，则命令中包含的 7 位地址值应写入 USB_CTRL 寄存器的 FADDR。然后，应写入 USB_TXCTRL 寄存器以设置 SRPCLR 位（USB_TXCTRL.D22）（指示已从 FIFO 读取命令）并设置 DATAEND（USB_TXCTRL.D19）（指示此请求不需要其他数据）。当主机移至请求的状态阶段时，将生成第二个 Endpoint 0 中断以指示请求已完成。该软件无需采取进一步措施：第二个中断只是对请求成功完成的确认。

如果该命令是无法识别的命令，或者由于某些其他原因而无法执行，则在对其进行解码后，应写入对 SRPCLR 位写 1（USB_TXCTRL.D22）和对 SENDSTA 位写 1（USB_TXCTRL.D21）。当主机移至请求的状态阶段时，USB 将发送一个 STALL 通知主机未执行请求。将产生第二个端点 0 中断，并将 SENTSTA 位（USB_TXCTRL.D18）置 1。如果在将 DATAEND（USB_TXCTRL.D19）位置 1 之后主机发送更多数据，则 USB 将发送一个 STALL。将产生一个端点 0 中断，并且 SENTSTA 位（USB_TXCTRL.D18）将被置 1。

21.4.5.3 写入请求

写请求涉及在 8 字节命令之后从主机发送的一个或多个附加数据包。“写入”标准设备请求的一个示例是：SET_DESCRIPTOR。

与所有请求一样，事件序列将在软件收到端点 0 中断时开始。RXPR 位（USB_TXCTRL.D16）也将被置位。然后应从端点 0 FIFO 中读取 8 字节命令并进行解码。与零数据请求一样，然后应写入 USB_TXCTRL 寄存器以将 SRPCLR 位（USB_TXCTRL.D22）置 1（指示已从 FIFO 中读取命令），但在这种情况下，不应设置 DATAEND（USB_TXCTRL.D19）（指示预计会有更多数据）。收到第二个端点 0 中断时，应读取 USB_TXCTRL 寄存器以检查端点状态。RXPR 位（USB_TXCTRL.D16）应设置为指示已接收到数据包。然后应读取 RXCOUNT 寄存器以确定此数据包的大小。然后可以从端点 0 FIFO 中读取数据包。

如果与请求关联的数据长度（由命令中的 wLength 字段指示）大于端点 0 的最大数据包大小，则将发送其他数据包。在这种情况下，应写入 USB_TXCTRL 以将 SRPCLR 位置 1，但不应将 DATAEND 位置 1。当接收到所有预期的数据包时，应写入 USB_TXCTRL 寄存器以将 SRPCLR 位置 1 并将 DATAEND 位置 1（指示不需要更多数据）。当主机移至请求的状态阶段时，将生成另一个端点 0 中断以指示请求已完成。该软件不需要采取进一步的措施，中断只是对请求成功完成的确认。

如果该命令是无法识别的命令，或者由于某些其他原因而无法执行，则在对其进行解码后，应写入 TXCSRN 寄存器以设置 SRPCLR 位（USB_TXCTRL.D22）和设置 SENDSTA 位（USB_TXCTRL.D21）。当主机发送更多数据时，USB 将发送一个 STALL 通知主机未执行该请求。将产生一个端点 0 中断，并且 SENTSTA 位（USB_TXCTRL.D18）将被置 1。如果主机在设置了 DATAEND（USB_TXCTRL.D19）之后发送更多数据，则 USB 将发送一个 STALL。将产生一个端点 0 中断，并且 SENTSTA 位（USB_TXCTRL.D18）将被置 1。

21.4.5.4 读取请求

读请求具有一个发送到主机的 8 字节数据包。“读取”标准设备请求的示例包括：GET_CONFIGURATION，GET_INTERFACE，GET_DESCRIPTOR，GET_STATUS，SYNCH_FRAME。

与所有请求一样，事件序列将在软件收到端点 0 中断时开始。RXPR 位（USB_TXCTRL.D16）也将被置位。然后应从端点 0 FIFO 中读取 8 字节命令并进行解码。然后应写入 USB_TXCTRL 寄存器以将 SRPCLR 位（USB_TXCTRL.D22）置 1（指示该命令已从 FIFO 中读取）。然后应将要发送到主机的数据写入端点 0 FIFO。如果要发送的数据大于端点 0 的最大数据包大小，则应仅将最大数据包大小写入 FIFO。然后，应写入 USB_TXCTRL 寄存器以设置 TXPR 位（USB_TXCTRL.D17）（指示 FIFO 中有要发送的数据包）。将数据包发送到主机后，将生成另一个端点 0 中断，并且可以将下一个数据包写入 FIFO。将最后一个数据包写入 FIFO 后，应写入 TXCSRN 寄存器以设置 TXPR 位和设置 DATAEND（USB_TXCTRL.D19）（指示此数据包之后没有更多数据）。当主机移至请求的状态阶段时，将生成另一个端点 0 中断以指示请求已完成。该软件不需要采取进一步的措施：中断只是对请求成功完成的确认。

如果该命令是无法识别的命令，或者由于某些其他原因而无法执行，则在对其进行解码后，应写入 TXCSRN 寄存器以将 SRPCLR 位（USB_TXCTRL.D22）和 SENTSTA 位（USB_TXCTRL.D18）（D5）置 1。当主机请求数据时，USB 将发送一个 STALL 来通知主机该请求未执行。将产生一个端点 0 中断，并且 SENTSTA 位（USB_TXCTRL.D18）将被置 1。如果在设置了 DATAEND（USB_TXCTRL.D19）之后主机请求更多数据，则 USB 将发送一个 STALL。将产生一个端点 0 中断，并且 SENTSTA 位（USB_TXCTRL.D18）将被置 1。

21.4.6 BULK 传输

21.4.6.1 IN 事务传输

批量输入事务用于将非周期性数据从功能控制器传输到主机。两个可选功能可用于批量 IN 事务的 Tx 端点一起使用：

- 双包缓冲
  当写入 TXMAXP 寄存器的值小于或等于分配给端点的 FIFO 大小的一半时，将自动启用双包缓冲。启用后，最多可在 FIFO 中存储两个包，等待传输到主机。
- AUTOSET
  启用 AUTOSET 后，当将一个 TXMAP 字节的数据包加载到 FIFO 中时，TXPR 位（USB_TXCTRL.D16）将被自动设置。

在为批量事务配置 Tx 端点时，必须使用端点的最大数据包大小（以字节为单位）写入 TXMAXP 寄存器。此值应与端点的标准端点描述符的 wMaxPacketSize 字段相同。此外，应将 USB_TXCTRL 寄存器的高字节按如表 21-1 所示设置：

表 21-1 TXCTRL 寄存器高字节功能描述表

D31	AUTOSET	0/1	如果需要 AutoSet 功能，则设置为 1。
D30	ISO	0	设置为 0 以启用批量传输。
D29	MODE	1	设置为 1 以确保启用 FIFO。
D27	FRCDATTOG	0	设置为 0 允许正常的数据切换操作。

首次配置端点时（遵循端点 0 上的 SET_CONFIGURATION 或 SET_INTERFACE 命令），应写入 USB_TXCTRL 的低字节以将 SRPCLR 位（D22）置 1。这将确保数据切换（由 USB 自动处理）以正确的状态启动。同样，如果 FIFO 中有任何数据包（通过设置 TXPR 位（USB_TXCTRL.D17）指示），则应通过将 DATAEND 位（USB_TXCTRL.D19）置位来清除它们。注意：如果启用了双重缓冲，则可能有必要连续两次将该位置位。

操作指南

当要通过 Bulk IN 管道传输数据时，需要将数据包加载到 FIFO 中，并写入 USB_TXCTRL 寄存器以设置 TXPR 位（D16）。发送数据包后，USB 将清除 TXPR 位，并产生一个中断，以便可以将下一个数据包加载到 FIFO 中。如果启用了双重数据包缓冲，则在第一个数据包已装入并且 TXPR 位置 1 后，USB 将立即清除 TXPR 位，并产生一个中断，以便将第二个数据包装入 FIFO。该软件应以相同的方式运行，在接收到中断时加载数据包，而不管是否启用了双重数据包缓冲。

通常情况下，数据包的大小不得超过 TXMAXP 寄存器的低 11 位所指定的大小。寄存器的这一部分定义了通过 USB 传输的有效负载（数据包大小），并且 USB 规范要求其为 8、16、32、64（全速或高速）或 512 字节（高速）。如果要传输的数据量超过此数量，则需要将其作为多个 USB 数据包发送，大小应均为 TXMAXP [D10：D0]，但最后一个保留残差的数据包除外。

主机可以通过知道期望的数据总量来确定用于传输的所有数据。可替代地，当它接收到小于

规定的数据包（TXMAXP [D10：D0]）时，它可以推断出所有数据已被发送。在后一种情况下，如果数据块的总大小是此有效负载的倍数，则该功能将有必要在所有数据都已发送之后发送空数据包。通过在接收到下一个中断时将 TXPR 设置为完成，而不将任何数据加载到 FIFO 中。

如果软件要关闭 Bulk IN 管道，则应将 SETUPEND 位（USB_TXCTRL.D20）置 1（注意：SETUPEND 在端点 0 代表的是 SETUP 结束标志，在端点 1 代表的是发送 STALL，具体查看寄存器描述）。当 USB 收到下一个 IN 令牌时，它将向主机发送一个 STALL，将 SENDSTA 位（USB_TXCTRL.D21）置 1 并产生一个中断。当软件收到已将 SENDSTA 位（USB_TXCTRL.D21）置 1 的中断时，应清除 SENDSTA 位。但是，它应该将 SETUPEND 位（USB_TXCTRL.D20）置 1，直到准备重新启用 Bulk IN 管道为止。注意：如果主机由于某种原因未能接收到 STALL 数据包，它将发送另一个 IN 令牌，因此建议将 SETUPEND 位置 1，直到软件准备好重新启用 Bulk IN 管道为止。重新启用管道后，应通过将 USB_TXCTRL 寄存器（D22）中的 SRPCLR 位置 1（注意：SRPCLR 在端点 0 代表的是 RXPR 自清功能，在端点 1 代表的是重置 DATA PID 为 DATA0，具体查看寄存器描述），重新启动数据切换序列。

21.4.6.2 OUT 事务传输

Bulk OUT 事务用于将非定期数据从主机传输到功能控制器。三个可选功能可用于大容量 OUT 事务的 Rx 端点一起使用：

- 双包缓冲
  当写入 RxMaxP 寄存器的值小于或等于分配给端点的 FIFO 大小的一半时，将自动启用双包缓冲。启用后，FIFO 中最多可以存储两个包。
- AUTOCLR
  启用 AUTOCLR 功能后，当已从 FIFO 中卸载 RXMAXP 字节的数据包时，RXPR 位（USB_RXCTRL.D16）将被自动清除（例外，请参见寄存器说明）。

在为批量输出事务配置 Rx 端点时，必须使用该端点的最大数据包大小（以字节为单位）写入 RXMAXP 寄存器。该值应与端点的标准端点描述符的 wMaxPacketSize 字段相同。此外，应将 USB_RXCTRL 寄存器的高字节应如表 21-2 所示：

表 21-2 RXCTRL 寄存器高字节功能描述表

D31	AUTOCLR	0/1	如果需要 AutoSet 功能，则设置为 1。
D30	ISO	0	设置为 0 以启用批量传输。

首次配置端点时（遵循端点 0 上的 SET_CONFIGURATION 或 SET_INTERFACE 命令），应写入 USB_RXCTRL 以将 CDATTOG 位（D23）置 1。这将确保数据切换（由 USB 自动处理）以正确的状态启动。同样，如果 FIFO 中有任何数据包（由 RXPR 位（USB_RXCTRL.D16）指示），则应通过将 FLFIFO 位（USB_RXCTRL.D20）置位来刷新它们。注意：如果启用了双重缓冲，则可能有必要连续两次将该位置位。

操作指南

当 Bulk Rx 端点接收到数据包时，RXPR 位（USB_RXCTRL.D16）被置位并产生中断。软件应读取端点的 RXCOUNT 寄存器，以确定数据包的大小。应从 FIFO 中读取数据包，然后应

清除 RXPR。如果在清除 RXPR 时将 FIFOFULL 位设置为 1，则 USB 将首先清除 FIFOFULL 位。然后它将再次设置 RXPR，以指示 FIFO 中有另一个要卸载的数据包。

收到的数据包不应超过 RXMAXP 寄存器中指定的大小（因为这应该是在发送给主机的端点描述符的 wMaxPacketSize 字段中设置的值）。当需要将大于 wMaxPacketSize 的数据块发送给函数时，它将作为多个数据包发送。除最后一个包含残差的数据包外，所有数据包的大小均为 wMaxPacketSize。软件可以使用特定于应用的方法来确定块的总大小，并因此确定何时已接收到最后一个数据包。可替代地，当其接收到大小小于 wMaxPacketSize 的分组时，可以推断出整个块已经被接收。（如果数据块的总大小是 wMaxPacketSize 的倍数，则将在数据之后发送一个空数据包以表示传输已完成。）

21.4.7 中断传输

中断 IN 事务使用与批量 IN 事务相同的协议，并且可以以相同的方式使用。同样，中断 OUT 事务使用与 Bulk OUT 事务几乎相同的协议，并且可以以相同的方式使用。

Tx 端点支持 Bulk IN 事务中不支持的 Interrupt IN 事务的一项功能——支持数据切换位的连续切换。通过将 USB_TXCTRL 寄存器中的 FRCDATTOG 位置 1，可以启用此功能。当该位设置为“1”时，无论是否从主机接收到 ACK，USB 都会将数据包视为已成功发送，并切换端点的数据位。

另一个区别是中断端点不支持 PING 流控制。这意味着 USB 绝不应该响应 NYET 握手，而只能响应 ACK / NAK / STALL。为确保这一点，应将 USB_RXCTRL 寄存器（D12）中的 DISNYET 位设置为 1，以禁止传输 NYET 握手。

21.4.8 ISO 传输

21.4.8.1 IN 事务传输

同步 IN 事务用于将周期性数据从功能控制器传输到主机。本节描述了 USB 全速同步 Tx 端点的使用。可将三种可用于同步 IN 事务的 Tx 端点一起使用：

- 双包缓冲
  当写入 TxMaxP 寄存器的值小于或等于分配给端点的 FIFO 大小的一半时，将自动启用双包缓冲。启用后，最多可在 FIFO 中存储两个包，等待传输到主机。注意：通常建议对同步事务使用双包缓冲
- AUTOSET
  当为低带宽同步端点启用自动设置功能时，当将 TXMAXP 字节的数据包加载到 FIFO 中时，TXPR 位将自动设置。但是，此功能对于同步端点不是特别有用，因为传输的数据包通常不是最大数据包大小，并且在每个数据包之后都需要访问 USB_TXCTRL 寄存器以检查是否运行错误。

在为同步 IN 事务配置 Tx 端点时，必须使用端点的最大数据包大小（以字节为单位）写入 TXMAXP 寄存器。此值应与端点的标准端点描述符的 wMaxPacketSize 字段相同。此外，应将 USB_TXCTRL 寄存器的高字节按如表 21-3 所示设置：

表 21-3 TXCTRL 寄存器高字节功能描述表

D31	AUTOSET	0/1	如果需要 AutoSet 功能，则设置为 1。
D30	ISO	1	设置为 1 以启用 ISO 传输。
D29	MODE	1	设置为 1 以确保启用 FIFO。
D27	FRCDATTOG	0	设置为 0 允许正常的数据切换操作。

操作指南

同步端点不支持数据重试，因此，必须在接收 IN 令牌之前将要发送到主机的数据加载到 FIFO 中。主机每帧将发送一个 IN 令牌，但是帧（或微帧）内的时序可能会有所不同。如果在一个帧的结尾附近接收到 IN 令牌，然后在下一帧的开始处接收到 IN 令牌，则将没有时间重新加载 FIFO。因此，通常需要对端点进行双重缓冲。

AUTOSET 功能可与低带宽同步 Tx 端点一起使用，其方式与批量 Tx 端点相同。但是，除非数据以绝对一致的速率从源到达，并且与主机的帧时钟同步，否则发送到主机的数据包的大小必须逐帧增加或减少（或从微帧到微帧）才能匹配源数据速率。这意味着实际的数据包大小将不总是 TXMAXP，从而使自动设置功能无效。

每当将数据包发送到主机时，都会产生一个中断，并且软件可以使用该中断将下一个数据包加载到 FIFO 中，并以与批量 Tx 端点相同的方式将 USB_TXCTRL 寄存器中的 TXPR 位置 1。由于中断可能几乎在一帧（/微帧）内的任何时间发生，具体取决于主机安排事务的时间，因此这可能会导致 FIFO 加载请求的时序不规则。如果端点的数据源来自某些外部硬件，则在加载 FIFO 之前等到每个帧（/微帧）结束可能会更方便，因为这将最大程度地减少对额外缓冲的需求。这可以通过使用 SOF 中断（IntrUSB.D3）触发下一个数据包的加载来完成。中断仍可用于设置 USB_TXCTRL 中的 TXPR 位并检查数据超限/不足运行（请参见下面的“错误处理”）。

启动双缓冲同步 IN 管道可能会引起问题。双缓冲要求数据包只有在加载后的帧（/微帧）之后才发送。如果函数在主机设置管道之前（因此开始发送 IN 令牌），至少将第一个数据包加载到一个帧（/微帧），则没有问题。但是，如果主机在第一个数据包加载时已经开始发送 IN 令牌，则取决于在 IN 之前还是之后加载，可以在加载该数据包的同一帧（/微帧）中传输该数据包。令牌已收到。可以通过将 USB_CTRL 寄存器中的 ISOUP 位置 1 来避免此潜在问题。当该位设置为 1 时，加载到同步 Tx 端点 FIFO 中的任何数据包都不会被发送，直到接收到下一个 SOF 包之后，从而确保该数据包不会过早发送。

错误处理

如果端点在接收到 IN 令牌后在其 FIFO 中没有数据，它将向主机发送一个空数据包，并将 USB_TXCTRL 寄存器中的 SENTSTA 位（对于端点 0 代表的是发送 STALL 使能，对于端点 1 代表的是未设置 TXPR 但是收到了 IN 令牌）置 1。这表明软件无法为主机提供足够快的数据。由应用程序确定如何处理此错误情况。

如果软件正在每帧(/微帧)加载一个数据包，并且发现它想加载下一个数据包时 USB_TXCTRL 寄存器中的 TXPR 位被置位，则表明尚未发送数据包（也许 因为来自主机的 IN 令牌已损坏）。取决于应用程序如何处理这种情况：它可以选择通过将 USB_TXCTRL 寄存器中的 DATAEND 位（对于端点 0 代表的是数据结束设置，对于端点 1 代表的是 FIFO 刷新指令）置 1 来刷新未

发送的数据包，或者可以选择跳过当前数据包。

21.4.8.2 OUT 事务传输

同步 OUT 事务用于将周期性数据从主机传输到功能控制器。本节描述了在外围模式下全速同步 Rx 端点的使用三种可选功能可用于同步 OUT 事务的 Rx 端点一起使用：

- 双包缓冲
  当写入 RXMAXP 寄存器的值小于或等于分配给端点的 FIFO 大小的一半时，将自动启用双包缓冲。启用后，最多可在 FIFO 中存储两个包，等待传输到主机。注意：通常建议对同步事务使用双包缓冲，以避免溢出错误。
- AUTOCLR
  启用自动清除功能后，当已从 FIFO 中卸载 RXMAXP 字节的数据包时，RXPR 位将被自动清除（例外，请参见寄存器说明）。但是，此功能在同步端点上不是特别有用，因为传输的数据包通常不是最大数据包大小，并且在每个数据包之后都需要访问 USB_RXCTRL 寄存器以检查溢出或 CRC 错误。

在为批量输出事务配置 Rx 端点时，必须使用该端点的最大数据包大小（以字节为单位）写入 RXMAXP 寄存器。该值应与端点的标准端点描述符的 wMaxPacketSize 字段相同。此外，应将 USB_RXCTRL 寄存器的高字节应如表 21-4 所示：

表 21-4 RXCTRL 寄存器高字节功能描述表

D31	AUTOCLR	0/1	如果需要 AutoSet 功能，则设置为 1。
D30	ISO	1	设置为 1 以启用 ISO 传输。

操作指南

同步端点不支持数据重试，因此，如果要避免数据溢出，则在接收数据包时，FIFO 中必须有空间来接受数据包。主机每帧（或高速模式下的微帧）将发送一个数据包，但是帧内的时间可能会有所不同。如果在一个帧（/微帧）的末尾附近接收到一个数据包，而另一个在下一帧的开始处到达，则将没有时间卸载 FIFO。因此，通常需要对端点进行双重缓冲。

AUTOCLR 功能可与同步 Rx 端点一起使用，方式与批量 Rx 端点相同。但是，除非数据接收器以绝对一致的速率接收数据并同步到主机的帧时钟，否则从主机发送的数据包的大小必须逐帧增加或减少（或从微帧到微帧）才能匹配所需的数据速率。这意味着实际的数据包大小并不总是 RXMAXP，从而使 AUTOCLR 功能失效。

每当从主机接收到数据包时，都会产生一个中断，并且软件可以使用此中断从 FIFO 卸载数据包并以与批量 Rx 端点相同的方式清除 USB_RXCTRL 寄存器中的 RXPR 位。由于中断几乎可以在一帧（/微帧）内的任何时间发生，具体取决于主机安排事务的时间，因此 FIFO 卸载请求的时间可能不规则。如果端点的数据接收器要连接到某些外部硬件，则最好等到每个帧（/微帧）结束后再卸载 FIFO，以最大程度地减少对额外缓冲的需求。这可以通过使用 SOF 中断（USBINTR.D3）。中断仍可用于清除 USB_RXCTRL 中的 RXPR 位并检查数据是否超限/不足运行（请参见下面的“错误处理”）。

错误处理

如果从主机接收到数据包时，如果 FIFO 中没有空间存储数据包，则 USB_RXCTRL 寄存器中的 OVERRUN 位将被置 1。这表明该软件无法为主机快速卸载数据。由应用程序确定如何处理此错误情况。

如果 USB 发现接收到的数据包存在 CRC 错误，它将仍然将数据包存储在 FIFO 中，并将 RXPR 位（USB_RXCTRL.D16）和 DATAERR 位（USB_RXCTRL.D18）置 1。由应用程序决定如何处理此错误情况。

21.5 寄存器定义

21.5.1 寄存器列表

USB 基地址：

USB(0x40035000)

偏移	实例地址	名称	默认值	描述
0x00	USB 基地址+0x00	USB_CTRL	0x00000000	USB 控制寄存器
0x0C	USB 基地址+0x0C	USB_INDEX	0x00000000	USB 索引寄存器
0x10	USB 基地址+0x10	USB_TX0CTRL	0x00000000	USB TX 端点 0 控制寄存器
0x10	USB 基地址+0x10	USB_TXnCTRL	0x00000000	USB TX 端点 n 控制寄存器
0x14	USB 基地址+0x14	USB_RXCTRL	0x00000000	USB RX 端点控制寄存器
0x18	USB 基地址+0x18	USB_RXCOUNT	0x00000000	USB RXFIFO 总字节寄存器
0x1C	USB 基地址+0x1C	USB_FIFOSIZE	0x00000000	USB FIFO 大小寄存器
0x60	USB 基地址+0x60	USB_FIFOSZ	0x00000000	USB 动态 FIFO 寄存器
0x64	USB 基地址+0x64	USB_FIFOAD	0x00000000	USB FIFO 起始地址寄存器
0x400	USB 基地址+0x400	USB_PINCTRL	0x55442204	USB DP/DM 控制寄存器
0x410	USB 基地址+0x410	USB_IUINTREN	0x00000400	USB 插拔检测中断使能寄存器
0x414	USB 基地址+0x414	USB_EPINTREN	0x00000000	USB Tx/Rx 端口中断使能寄存器
0x418	USB 基地址+0x418	USB_USBINTREN	0x00000000	USB Power 中断使能寄存器
0x420	USB 基地址+0x420	USB_IUINTR	0x00000000	USB 插拔检测中断寄存器
0x424	USB 基地址+0x424	USB_EPINTR	0x00000000	USB Tx/Rx 端口中断寄存器
0x428	USB 基地址+0x428	USB_USBINTR	0x00000000	USB Power 中断寄存器
0x440	USB 基地址+0x440	USB_FIFO0	0x00000000	USB 端点 0 FIFO 读写寄存器
0x444	USB 基地址+0x444	USB_FIFO1	0x00000000	USB 端点 1 FIFO 读写寄存器
0x448	USB 基地址+0x448	USB_FIFO2	0x00000000	USB 端点 2 FIFO 读写寄存器

21.5.2 寄存器描述

21.5.2.1 USB 控制寄存器（USB_CTRL）

- 名称：**USB Control Register Include Address And Power Control**
- 偏移地址：**0x00**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISOUP	SCONN			RESET	RESUME	SUSMOD		FADDR							
R/W	R/W			R	R/WAC	RC/W		R/W							
0x00	0x00			0x00	0x00	0x00		0x00							

字段	说明
[15] ISOUP	**ISO 更新使能（ISO Upgrade Enable）** 置 1 时，USB 将等待从设置 TxPktRdy 开始的 SOF 令牌，然后再发送数据包。如果在 SOF 令牌之前收到 IN 令牌，则将发送零长度的数据包。 Note：只会影响执行同步传输的端点
[14] SCONN	**软件连接使能（Software Connect Enable）** 1 ：将启用 USB D + / D-线 0 ：将其变为三态。
[11] RESET	**复位检测信号（Reset Detect Flag）** 当总线上存在复位信号时，该位置 1。
[10] RESUME	**唤醒使能（Resume Enable）** 当设备处于挂起模式时，由 CPU 置 1 以生成恢复信令。在外设模式下，CPU 应在 10 毫秒（最多 15 毫秒）后清除此位，以结束挂起恢复通信。
[9] SUSMOD	**挂起使能（Suspend Enable）** 在外设模式下，此位设置为进入挂起模式。当 CPU 读取中断寄存器或将上面的 Resume 位设置为 1 时，该位被清除。
[7:0] FADDR	**设备地址（address of the peripheral）** 设备地址设置

21.5.2.2 USB 索引寄存器（USB_INDEX）

- **名称：USB Index And Frame Number Register**
- **偏移地址：0x0C**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												INDEX			
												R/W			
												0x00			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FNUM															
R															
0x00															

字段	说明
[19:16] INDEX	**端点索引寄存器（Endpoint Index Register）** INDEX 是一个 4 位寄存器，它确定要访问的端点控制/状态寄存器。
[15:0] FNUM	**SOF 帧号（SOF Frame Number）** 保存最后收到的帧号

21.5.2.3 USB TX 端点 0 控制寄存器（USB_TX0CTRL）

- **名称：USB TX Endpoint 0 Control Register**
- **偏移地址：0x10**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							FLFIFO	SSECLR	SRPCLR	SENDSTA	SETUPEND	DATAEND	SENTSTA	TXPR	RXPR
							WAC	WAC	WAC	WAC	R/W	WAC	R/W	WAC	WAC
							0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMAXP															
R/W															
0x00															

字段	说明
[24] FLFIFO	**端点 0 FIFO 刷新使能（Flush FIFO Enable）** 对于端点 0（INDEX=0） CPU 向该位写入 1，以刷新要从端点 0 FIFO 读取/发送的下一个数据包。FIFO 指针被复位并且 TxPktRdy / RxPktRdy 位被清除。 Note：仅当设置了 TxPktRdy / RxPktRdy 时，才应使用 FlushFIFO。在其他时候，这可能会导致数据损坏。
[23] SSECLR	**端点 0 SETUPEND 位清除（SetupEnd Clear）** 对于端点 0（INDEX=0） CPU 将 1 写入该位以清除 SETUPEND 位。它会自动清除。
[22] SRPCLR	**端点 0 RXPR 清除（RxPktRdy Clear）** 对于端点 0（INDEX=0） CPU 将 1 写入该位以清除 RXPR 位。自动清除
[21] SENDSTA	**端点 0 发送 Stall 使能（Send Stall Enable）** 对于端点 0（INDEX=0） CPU 向该位写入 1 以终止当前事务。将发送 STALL 握手信号，然后自动清除该位。
[20] SETUPEND	**端点 0 Setup 结束标志（Setup Transfer End Flag）** 对于端点 0（INDEX=0）

	当控制事务在 DataEnd 位置 1 之前结束时，该位置 1。此时将产生一个中断并刷新 FIFO。CPU 通过将 1 写入 SSE 位来清除该位。
[19] DATAEND	**端点 0 数据结束标志（Last Data End Flag）** 对于端点 0（INDEX=0） CPU 置 1： 1.在为最后一个数据包设置 TXPR 时。 2.卸载最后一个数据包后清除 RXPR。 3.为零长度的数据包设置 TXPR 时。 Note：自动清除
[18] SENTSTA	**端点 0 发送完 Stall 标志（Sent Stall Flag）** 对于端点 0（INDEX=0） 发送停止握手时，该位置 1。CPU 应该清除该位。
[17] TXPR	**端点 0 TX 准备就绪（Endpoint 0 TX Ready）** 对于端点 0（INDEX=0） 将数据包加载到 FIFO 后，CPU 将该位置 1。传输数据包后，它将自动清除。此时也会生成一个中断
[16] RXPR	**端点 0 RX 准备就绪（Endpoint 0 RX Ready）** 对于端点 0（INDEX=0） 当接收到数据包时该位置位。该位置 1 时产生中断。CPU 通过将 SRPCLR 位置 1 清除该位。
[15:0] TXMAXP	**端点 TX 最大数据量（Endpoint TX Max Data Number）** 定义单个事务中传输的最大有效负载。该值最多可以为 256 字节，USB 只有一个 256 字节的缓存空间

21.5.2.4 USB TX 端点 n 控制寄存器（USB_TXnCTRL）

- **名称：USB TX Endpoint n Control Register**
- **偏移地址：0x10**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AUTOSET	ISO	MODE		FRCDATTOG					CLRDT	SENTSTA	SENDSTA	FLFIFO	UNRUN	FIFONE	TXPR
R/W	R/W	R/W		R/W					W	R/W	R/W	W	R/W	R	R/W
0x00	0x00	0x00		0x00					0x00	0x00	0x00	0x00	0x00	0x00	0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMAXP															
R/W															
0x00															

字段	说明
[31] AUTOSET	**端点 n 自动设置使能（Auto Set Enable）** 对于非端点 0（INDEX=1,2） 如果 CPU 将该位置 1，则将最大数据包大小（TxMaxP 中的值）的数据加载到 TX FIFO 中时，TXPR 将自动设置。如果加载的数据包小于最大数据包大小，则必须手动设置 TXPR。
[30] ISO	**端点 n ISO 传输使能（ISO Transfer Enable）** 对于非端点 0（INDEX=1,2） CPU 将该位置 1 以使 TX 端点启用同步传输，并清除该位以使 TX 端点启用批量或中断传输。
[29] MODE	**端点 n 方向模式选择（Endpoint Direction Select）** 对于非端点 0（INDEX=1,2） CPU 将该位置 1 以将端点方向启用为 TX，将其清除以将其启用为 Rx。
[27] FRCDATTOG	**端点 n 强制数据切换（Force Data Toggle）** 对于非端点 0（INDEX=1,2） CPU 将该位置 1，以强制端点数据切换，并强制从 FIFO 中清除数据包，而不管是否收到 ACK。可以用于中断 TX 端点，用于为同步端点传递速率反馈。
[22] CLRDT	**端点 n 清除数据切换（Clear Data Toggle）** 对于非端点 0（INDEX=1,2） CPU 在该位写入 1 以将端点数据切换重置为 0，接下来发送 Data0
[21]	**端点 n 发送完 STALL（Send STALL End）**

SENTSTA	对于非端点 0（INDEX=1,2） 发送停止握手时，此位置 1。刷新 FIFO 并清除 TXPR 位（见下文）。对该位写 0 清除该位。
[20] SENDSTA	**端点 n 发送 STALL（Send STALL）** 对于非端点 0（INDEX=1,2） CPU 对此位写入 1，收到 IN 令牌将会发出 STALL 握手信号。CPU 对该位写 0 清除该位。注意：在将端点用于同步传输的位置，此位无效。
[19] FLFIFO	**端点 n FIFO 刷新使能（Flush FIFO Enable）** 对于非端点 0（INDEX=1,2） CPU 对此位写入 1，以刷新来自端点 TX FIFO 的最新数据包。复位 FIFO 指针，清除 TxPktRdy 位（下），并产生中断。可以与 TxPktRdy 同时设置以中止当前正在加载到 FIFO 中的数据包。注意：仅当设置了 TxPktRdy 时，才应使用 FlushFIFO。在其他时候，这可能会导致数据损坏。另请注意，如果 FIFO 是双缓冲的，则可能需要将 FlushFIFO 设置两次以完全清除 FIFO。
[18] UNRUN	**端点 n 欠限（Underrun）** 对于非端点 0： 如果未设置 TXPR 时接收到 IN 令牌，则 USB 会将该位置 1。对该位写 1 清除该位。
[17] FIFONE	**端点 n FIFO 非空（FIFO Not Empty）** 对于非端点 0（INDEX=1,2） 当 TX FIFO 中至少有 1 个数据包时，USB 将该位置 1
[16] TXPR	**端点 n TX 准备就绪（Endpoint n TX Ready）** 对于非端点 0（INDEX=1,2） 将数据包加载到 FIFO 后，CPU 将该位置 1。传输数据包后，它将自动清除。此时也会产生一个中断。在将第二个数据包加载到双缓冲 FIFO 中之前，TXPR 也会自动清除。
[15:0] TXMAXP	**端点 TX 最大数据量（Endpoint TX Max Data Number）** 定义单个事务中传输的最大有效负载。该值最多可以为 256 字节，USB 只有一个 256 字节的缓存空间

21.5.2.5 USB RX 端点控制寄存器（USB_RXCTRL）

- **名称：USB RX Endpoint Control Register**
- **偏移地址：0x14**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AUTO CLR	ISO						INCRX	CDAT TOG	SENTS TA	SEND STA	FLFIF O	DATA ERR	OVER RUN	FIFOF ULL	RXPR
R/W	R/W						R	W	R/W	R/W	W	R	R/W	R	R/W
0x00	0x00						0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXMAXP															
R/W															
0x00															

字段	说明
[31] AUTOCLR	**自动清除使能（Auto Clear Enable）** 如果 CPU 将该位置 1，则当已从 Rx FIFO 卸载 RxMaxP 字节的数据包时，RxPktRdy 位将被自动清除。当小于最大数据包大小的数据包被卸载时，必须手动清除 RxPktRdy。使用 DMA 卸载 RxFIFO 时，无论 RxMaxP 是多少，都以 4 字节块的形式从 RxFIFO 读取数据。因此，RxPktRdy 位将被清除
[30] ISO	**ISO 传输使能（ISO Transfer Enable）** CPU 将该位置 1 以使 Rx 端点启用同步传输，并清除该位以使 Rx 端点启用批量/中断传输。
[24] INCRX	**接收不完整数据包指示（Incomplete Data Flag）** 如果由于未接收到部分数据而导致 Rx FIFO 中的数据包不完整，则在高带宽同步/中断传输中将该位置 1。清除 RxPktRdy 时清除。 Note：在同步传输以外的任何方式下，该位将始终返回 0。
[23] CDATTOG	**清除数据切换（Clear Data Toggle）** CPU 将 1 写入该位以将端点数据触发器重置为 0。
[22] SENTSTA	**发送完 Stall 标志（Sent Stall Flag）** 发送停止握手时，该位置 1。对该位写 0 清除该位。

[21] SENDSTA	**发送 Stall 使能（Send Stall Enable）** CPU 向该位写入 1 以发出 STALL 握手信号。CPU 清除该位以终止停止条件。 Note：当端点用于同步传输时，此位无效
[20] FLFIFO	**FIFO 刷新使能（Flush FIFO Enable）** CPU 向该位写入 1，以刷新要从端点 Rx FIFO 读取的下一个数据包。FIFO 指针被复位并且 RxPktRdy 位（如下）被清除。注意：仅当设置了 RxPktRdy 时，才应使用 FlushFIFO。在其他时候，这可能会导致数据损坏。另请注意，如果 FIFO 是双缓冲的，则可能需要将 FlushFIFO 设置两次以完全清除 FIFO。
[19] DATAERR	**数据包错误（Data Error Flag）** 如果数据包具有 CRC 或位填充错误，则在设置 RxPktRdy 时设置此位。清除 RXPR 时清除。 Note：仅当端点在 ISO 模式下运行时，此位才有效。在批量模式下，它始终返回零。
[18] OVERRUN	**RXFIFO 溢出（RXFIFO Overrun）** 如果无法将 OUT 数据包加载到 Rx FIFO 中，则该位置 1。对该位写 0 清除该位。 Note：仅当端点在 ISO 模式下运行时，此位才有效。在批量模式下，它始终返回零。
[17] FIFOFULL	**FIFO 满标志（FIFO Full Flag）** 当没有更多的数据包可以加载到 Rx FIFO 中时，该位置 1
[16] RXPR	**端点 RX 准备就绪（Endpoint RX Ready）** 当接收到数据包时该位置位。从 Rx FIFO 卸载数据包后，对该位写 0 清除该位。当该位置 1 时，产生一个中断。
[15:0] RXMAXP	**端点 RX 最大数据量（Endpoint RX Max Data Number）** 定义单个事务中传输的最大有效负载。该值最多可以为 256 字节，USB 只有一个 256 字节的缓存空间

21.5.2.6 USB RXFIFO 总字节寄存器（USB_RXCOUNT）

- **名称：USB Number of bytes to be read from RX endpoint FIFO**
- **偏移地址：0x18**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		RXCOUNT													
		R													
		0x00													

字段	说明
[13:0] RXCOUNT	**收到的数据字节数（RX Data Count）** RxCount 是一个 14 位只读寄存器，其中包含当前要从 Rx FIFO 中读取的行中数据包中的数据字节数。如果数据包是作为多个批量数据包发送的，则给出的数字将用于组合数据包。

21.5.2.7 USB FIFO 大小寄存器（USB_FIFOSIZE）

- 名称：**USB Returns the configured size of the selected RX FIFO and TX FIFOs**
- 偏移地址：**0x1C**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFIFOSIZE				TXFIFOSIZE											
R				R											
0x00				0x00											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字段	说明
[31:28] RXFIFOSIZE	**RXFIFO 大小（RXFIFO Size）** 返回与所选附加 TX / Rx 端点关联的 FIFO 的大小。下半字节对所选 TX 端点 FIFO 的大小进行编码。高半字节对所选 Rx 端点 FIFO 的大小进行编码。值 3 – 13 对应于 2n 字节（8 – 8192 字节）的 FIFO 大小。
[27:24] TXFIFOSIZE	**TXFIFO 大小（TXFIFO Size）** 返回与所选附加 TX / Rx 端点关联的 FIFO 的大小。下半字节对所选 TX 端点 FIFO 的大小进行编码。高半字节对所选 Rx 端点 FIFO 的大小进行编码。值 3 – 13 对应于 2n 字节（8 – 8192 字节）的 FIFO 大小。

21.5.2.8 USB 动态 FIFO 寄存器（USB_FIFOSZ）

- 名称：**The Dynamic FIFO registers**
- 偏移地址：**0x60**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			RXDPB	RXSZ							TXDPB	TXSZ			
			R/W	R/W							R/W	R/W			
			0x00	0x00							0x00	0x00			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字段	说明
[28] RXDPB	**RX 端点双数据包缓冲使能（RX Endpoint Double Packet Buffer Enable）** 定义是否支持双数据包缓冲。 1：支持双数据包缓冲。 0：仅支持单数据包缓冲。
[27:24] RXSZ	**RX 端点最大数据包大小（RX Endpoint Maximum Packet Size）** 0000：8byte 0001： 16byte 0010： 32byte 0011： 64byte 0100： 128byte 0101： 256byte Note：如果 RXDPB=0，FIFO 也将是此大小；如果 DPB=1，FIFO 将是此大小的 2 倍
[20]	**TX 端点双数据包缓冲使能（TX Endpoint Double Packet Buffer Enable）**

TXDPB	定义是否支持双数据包缓冲。 1：支持双数据包缓冲。 0：仅支持单数据包缓冲。
[19:16] TXSZ	**TX 端点最大数据包大小（TX Endpoint Maximum Packet Size）** 0000：8byte 0001： 16byte 0010： 32byte 0011： 64byte 0100： 128byte 0101： 256byte Note：如果 TXDPB=0，FIFO 也将是此大小；如果 DPB=1，FIFO 将是此大小的 2 倍

21.5.2.9 USB FIFO 起始地址寄存器（USB_FIFOAD）

- **名称：USB FIFO Start Address Register**
- **偏移地址：0x64**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			RXAD												
			R/W												
			0x00												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TXAD												
			R/W												
			0x00												

字段	说明
[28:16] RXAD	**RX 端点 FIFO 的起始地址（Start Address Of RX Endpoint FIFO）** 配置数以 8 字节为单位，公式为 RXAD*8 byte： 0x0： 0byte 0x1： 8byte 0x2： 16byte 0x3： 24byte … 0x20：256byte
[12:0] TXAD	**TX 端点 FIFO 的起始地址（Start Address Of TX Endpoint FIFO）** 配置数以 8 字节为单位，公式为 TXAD*8 byte： 0x0： 0byte 0x1： 8byte 0x2： 16byte 0x3： 24byte … 0x20：256byte

21.5.2.10 USB DP/DM 控制寄存器（USB_PINCTRL）

- 名称：USB DP/DM PIN Control Register
- 偏移地址：0x400
- 默认值：0x55442204
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMOE	DMOE_EN	DPOE	DPOE_EN	DMIE	DMIE_EN	DPIE	DPIE_EN	DMSR	DMTRIM			DPSR	DPTRIM		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W		
0x0	0x1	0x0	0x1	0x0	0x1	0x0	0x1	0x0	0x4			0x0	0x4		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMPD	DMPD_EN	DMPU	DMPU_EN	DPPD	DPPD_EN	DPPU	DPPU_EN		Test_En	VBUS_VALID	AVALID	VBUS_LO	CID	TM1	PHY_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0	0x0	0x1	0x0	0x0	0x0	0x1	0x0		0x0	0x0	0x0	0x0	0x1	0x0	0x0

字段	说明
[31] DMOE	**DM 输出使能（DM Output Enable）** 1：DM 输出使能 0：DM 输出不使能
[30] DMOE_EN	**DM 软件输出使能（DM Software Output Enable）** 1：使用 bit[31] 0：使用硬件生成的 DM_OE
[29] DPOE	**DP 输出使能（DP Output Enable）** 1：DP 输出使能 0：DP 输出不使能
[28] DPOE_EN	**DP 软件输出使能（DP Software Output Enable）** 1：使用 bit[29] 0：使用硬件生成的 DP_OE
[27] DMIE	**DM 输入使能（DM Input Enable）** 1：DM 输入使能 0：DM 输入不使能
[26] DMIE_EN	**DM 软件输入使能（DM Software Input Enable）** 1：使用 bit[27] 0：使用硬件生成的 DM_OE
[25] DPIE	**DP 输入使能（DP Input Enable）** 1：DP 输入使能 0：DP 输入不使能
[24] DPIE_EN	**DP 软件输入使能（DP Software Input Enable）** 1：使用 bit[25] 0：使用硬件生成的 DP_OE
[23] DMSR	**DM PAD 输出 slewrate 增强使能（DM Output Slewrate Enhancement Enable）** 1：使能 0：不使能
[22:20] DMTRIM	**DM 端电阻修调（DM Resistor Trimming）** 000：MIN …… 111：MAX
[19] DPSR	**DP PAD 输出 slewrate 增强使能（DP Output Slewrate Enhancement Enable）** 1：使能 0：不使能
[18:16] DPTRIM	**DP 端电阻修调（DP Resistor Trimming）** 000：MIN …… 111：MAX
[15] DMPD	**DM 下拉使能（DM Pull Down Enable）** 1：使能

	0：不使能
[14] DMPD_EN	**DM 下拉软件使能（DM Software Pull Down Enable）** 1：使用 bit[15] 0：使用硬件生成的 DMPD
[13] DMPU	**DM 上拉使能（DM Pull Up Enable）** 1：使能 0：不使能
[12] DMPU_EN	**DM 上拉软件使能（DM Software Pull Up Enable）** 1：使用 bit[13] 0：使用硬件生成的 DMPU
[11] DPPD	**DP 下拉使能（DP Pull Down Enable）** 1：使能 0：不使能
[10] DPPD_EN	**DP 下拉软件使能（DP Software Pull Down Enable）** 1：使用 bit[11] 0：使用硬件生成的 DPPD
[9] DPPU	**DP 上拉使能（DP Pull Up Enable）** 1：使能 0：不使能
[8] DPPU_EN	**DP 上拉软件使能（DP Software Pull Up Enable）** 1：使用 bit[9] 0：使用硬件生成的 DPPU
[6] Test_En	**设置为 1 时开启 USB 测试模式，DPDM 会持续输出 6MHz 的时钟**
[5] VBUSVALID	**Vbus 高于 VBusValid 阈值使能**
[4] AVALID	**Vbus 高于 Vbus A 设备会话阈值使能**
[3] VBUSLO	**Vbus 超过会话结束阈值使能**
[2] CID	**mini AB 连接器 ID 引脚使能**
[1] TM1	**测试模式使能**
[0] PHY_EN	**PHY 使能，USB 使用就开**

21.5.2.11 USB 插拔检测中断使能寄存器（USB_IUINTREN）

- **名称：USB Insert/Unplug Detect Interrupt Enable Register**
- **偏移地址：0x410**
- **默认值：0x00000400**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DETDB														UPDE TIEN	ISDET IEN
R/W														R/W	R/W
0x40														0x0	0x0

字段	说明
[15:4] DETDB	**去抖配置（Debounce Config)** 插入拔出检测去抖计数器配置
[1] UPDETIEN	**USB 拔出检测中断使能（Unplug Detect Interrupt Enable）** 0：不使能

	1：使能
[0] ISDETIEN	**USB 插入检测中断使能（Insert Detect Interrupt Enable）** 0：不使能 1：使能

21.5.2.12 USB Tx/Rx 端口中断使能寄存器（USB_EPINTREN）

- **名称：USB Endpoint Tx/Rx Interrupt Enable Register**
- **偏移地址：0x414**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													EP2RX INTEN	EP1RX INTEN	
													R/W	R/W	
													0x0	0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													EP2TX INTEN	EP1TX INTEN	EP0IN TEN
													R/W	R/W	R/W
													0x0	0x0	0x0

字段	说明
[18] EP2RXINTEN	**EP2RXINT 中断使能（Endpoint 2 RX Interrupt Enable）** 0：不使能 1：使能
[17] EP1RXINTEN	**EP1RXINT 中断使能（Endpoint 1 RX Interrupt Enable）** 0：不使能 1：使能
[2] EP2TXINTEN	**EP2TXINT 中断使能（Endpoint 2 TX Interrupt Enable）** 0：不使能 1：使能
[1] EP1TXINTEN	**EP1TXINT 中断使能（Endpoint 1 TX Interrupt Enable）** 0：不使能 1：使能
[0] EP0INTEN	**EP0INT 中断使能（Endpoint 0 Interrupt Enable）** 0：不使能 1：使能

21.5.2.13 USB Power 中断使能寄存器（USB_USBINTREN）

- 名称：USB Power Interrupt Enable Register
- 偏移地址：0x418
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DISCONINTEN		SOFINTEN	RESETINTEN	RESUMEINTEN	SUSPENDINTEN
										R/W		R/W	R/W	R/W	R/W
										0x0		0x0	0x0	0x0	0x0

字段	说明
[5] DISCONINTEN	**DISCON 中断使能（Peripheral Disconnect Interrupt Enable）** 0：不使能 1：使能
[3] SOFINTEN	**SOF 中断使能（RX First SOF Interrupt Enable）** 0：不使能 1：使能
[2] RESETINTEN	**RESET 中断使能（Detect Reset Interrupt Enable）** 0：不使能 1：使能
[1] RESUMEINTEN	**RESUME 中断使能（Resume Interrupt Enable）** 0：不使能 1：使能
[0] SUSPENDINTEN	**SUSPEND 中断使能（Suspend Interrupt Enable）** 0：不使能 1：使能

21.5.2.14 USB 插拔检测中断寄存器（USB_IUINTR）

- 名称：USB Insert/Unplug Detect Interrupt Register
- 偏移地址：0x420
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														UPDETI	ISDETI
														R/W1C	R/W1C
														0x0	0x0

字段	说明
[1] UPDETI	**USB 拔出检测中断（Unplug Detect Interrupt）** 0：USB 未检测到拔出 1：USB 检测到拔出 Note：对该位写 1 清 0

[0] ISDETI	**USB 插入检测中断（Insert Detect Interrupt）** 0：USB 未检测到插入 1：USB 检测到插入 Note：对该位写 1 清 0

21.5.2.15 USB Tx/Rx 端口中断寄存器（USB_EPINTR）

- **名称：USB Endpoint Tx/Rx Interrupt Register**
- **偏移地址：0x424**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													EP2RX INT	EP1RX INT	
													R	R	
													0x00	0x00	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													EP2TX INT	EP1TX INT	EP0IN T
													R	R	R
													0x00	0x00	0x00

字段	说明
[18] EP2RXINT	**端点 2 RX 中断（Endpoint 2 RX Interrupt）** 端点 2 收到了 OUT 包 Note：读取该位清 0
[17] EP1RXINT	**端点 1 RX 中断（Endpoint 1 RX Interrupt）** 端点 1 收到了 OUT 包 Note：读取该位清 0
[2] EP2TXINT	**端点 2 TX 中断（Endpoint 2 TX Interrupt）** 端点 2 收到了 IN 包 Note：读取该位清 0
[1] EP1TXINT	**端点 1 TX 中断（Endpoint 1 TX Interrupt）** 端点 1 收到了 IN 包 Note：读取该位清 0
[0] EP0INT	**端点 0 中断（Endpoint 0 Interrupt）** 端点 1 收到了 IN 包、OUT 包、SETUP 包 Note：读取该位清 0

21.5.2.16 USB Power 中断寄存器（USB_USBINTR）

- 名称：USB Power Interrupt Register
- 偏移地址：0x428
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DISCON		SOF	RESET	RESUME	SUSPEND
										R		R	R	R	R
										0x00		0x00	0x00	0x00	0x00

字段	说明
[5] DISCON	**外设失去连接中断（Peripheral Disconnect Interrupt）** 在会话结束后置 1 Note：读取该位清 0
[3] SOF	**收到第一个 SOF 包中断（RX First SOF Interrupt）** 当一个新的帧开始时置 1 Note：读取该位清 0
[2] RESET	**检测到复位信号中断（Detect Reset Interrupt）** 当在总线上检测到复位信号时置 1 Note：读取该位清 0
[1] RESUME	**唤醒中断（Resume Interrupt）** 当 USB 处于挂起模式时，在总线上检测到继续命令时置 1。 Note：读取该位清 0
[0] SUSPEND	**挂起中断（Suspend Interrupt）** 在总线上检测到挂起信号时置 1 Note：读取该位清 0

21.5.2.17 USB 端点 0 FIFO 读写寄存器（USB_FIFO0）

- 名称：FIFOs for Endpoints 0
- 偏移地址：0x440
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFO0															
R/W															
0x00															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO0															
R/W															
0x00															

字段	说明
[31:0] FIFO0	**端点 0 的 FIFO 读写（Endpoint 0 FIFO Write/Read）** R：读取端点 0 的 RXFIFO W：填充端点 0 的 TXFIFO Note：支持按 Word、Half Word 和 Byte 读写

21.5.2.18 USB 端点 1 FIFO 读写寄存器（USB_FIFO1）

- 名称：**FIFOs for Endpoints 1**
- 偏移地址：**0x444**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFO1															
R/W															
0x00															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO1															
R/W															
0x00															

字段	说明
[31:0] FIFO1	**端点 1 的 FIFO 读写（Endpoint 1 FIFO Write/Read）** R：读取端点 1 的 RXFIFO W：填充端点 1 的 TXFIFO Note：支持按 Word、Half Word 和 Byte 读写

21.5.2.19 USB 端点 2 FIFO 读写寄存器（USB_FIFO2）

- 名称：**FIFOs for Endpoints 2**
- 偏移地址：**0x448**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFO2															
R/W															
0x00															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO2															
R/W															
0x00															

字段	说明
[31:0] FIFO2	**端点 2 的 FIFO 读写（Endpoint 2 FIFO Write/Read）** R：读取端点 2 的 RXFIFO W：填充端点 2 的 TXFIFO Note：支持按 Word、Half Word 和 Byte 读写

22 串行外设接口（SPI）

22.1 简介

串行外围接口（SPI）协议支持与外部设备的半双工和全双工的同步串行通信。SPI 可以作为主机，在这种情况，SPI 向外部从设备提供通信时钟（SCLK）和片选信号（NSS）。

22.2 主要特性

SPI 的主要功能如下：

- 在 4 根线下支持全双工同步传输
- 在 3 根线下支持半双工同步传输
- 支持主机模式和从机模式
- 支持 2~16 位的数据位长度发送和接收
- 支持可配的时钟极性和相位
- 支持可配的 NSS 信号极性和时序
- 支持 MSB 和 LSB 传输
- 支持可配的传输数据量
- 主机支持最高系统时钟的 2 分频通信速度
- 从机支持最高系统时钟的 8 分频通信速度
- 支持中断和轮询方式
- 支持 DMA 模式
- 支持主机 RX 的 delay chain 配置

22.3 结构框图

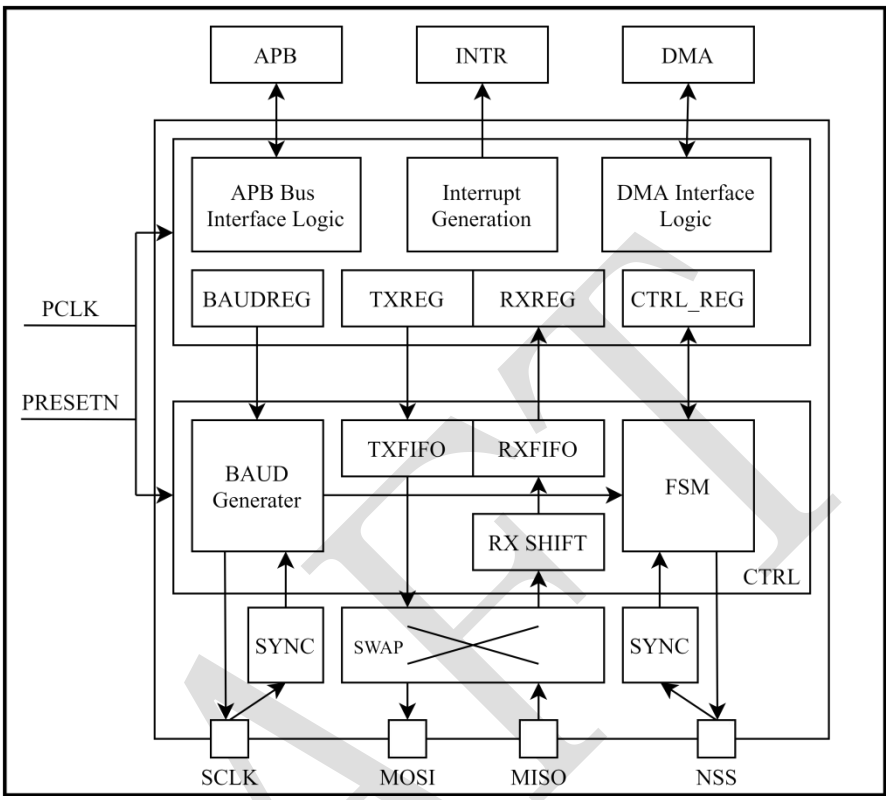


图 22-1 SPI 结构框图

22.4 功能描述

22.4.1 SPI 信号描述

SPI 属于串行同步通讯接口，在与外界交互的一共有 4 根引脚，具体如表 22-1 所示：

表 22-1 SPI 信号列表

信号	名称	详细描述
MOSI	主机输出/从机输入	在四线模式下，当 SPI 做主机时，该信号传输数据；当 SPI 做从机时，该信号接收数据。 在三线模式下，当 SPI 做主机时，该信号既传输数据也接收数据；当 SPI 做从机时，该信号无效。
MISO	主机输入/从机输出	在四线模式下，当 SPI 做从机时，该信号传输数据；当 SPI 做主机时，该信号接收数据。 在三线模式下，当 SPI 做从机时，该信号既传输数据也接收数据；当 SPI 做主机时，该信号无效。
NSS	SPI 片选信号	当 SPI 做主机时，该信号是输出片选信号；当 SPI 做从机时，该信号是输入片选信号。片选信号低有效。

SCLK	SPI 串行时钟	当 SPI 做主机时，该信号是输出串行时钟信号；当 SPI 做从机时，该信号是输入串行时钟信号。

SPI 的通信原理很简单，它以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，具体的拓扑图如图 22-2 所示。

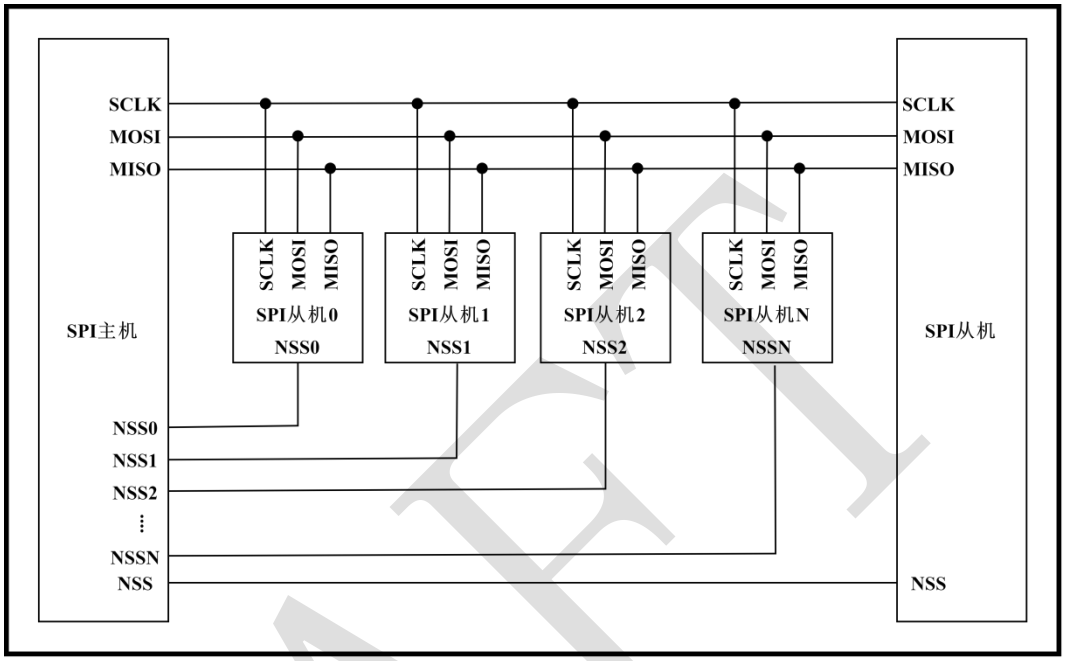


图 22-2　SPI 主从拓扑图

22.4.2　SPI 通信管理

22.4.2.1　NSS 管理

NSS 作为标准的片选信号，使得主器件和从器件可以进行通信。NSS 是由主器件传输给从器件，NSS 信号默认是低有效，在 SPI 中可以通过配置 SPI_ENABLE.CPOL 来改变 NSS 输入输出信号的有效电平，若 CPOL=0，NSS 的有效电平为低电平；若 CPOL=1，NSS 的有效电平为高电平。此外 SPI 还支持 NSS 作为主机时的软硬件输出选择和作为从机时的软硬件输入选择，具体如下：

- 主机模式（SPI_CTRL.MSTEN=1）：将 SPI_ENABLE.CSOS 设置为 1 时是软件输出模式，此时 NSS 的值只和 SPI_ENABLE.CSO 的配置值有关；设置为 0 时是硬件输出模式。
- 从机模式（SPI_CTRL.MSTEN=0）：将 SPI_ENABLE.CSIS 设置为 1 时是软件输入模式，此时 NSS 的值只和 SPI_ENABLE.CSI 的配置值有关；设置为 0 时是硬件输入模式。

从机模式下的 NSS 软件模式可以为 SPI 节省一个 GPIO。

在应用硬件输出 NSS 管理时，用户可以通过配置 SPI_ENABLE.CSSEL、SPI_TIMING.MIDI 和 SPI_TIMING.MCSI 来控制数据帧之间的 NSS 信号时序，以及插入每次事务开始时的额外延时（以分隔 NSS 和时钟启动），因为不同的从器件对处理数据速度不同，该配置十分有用。

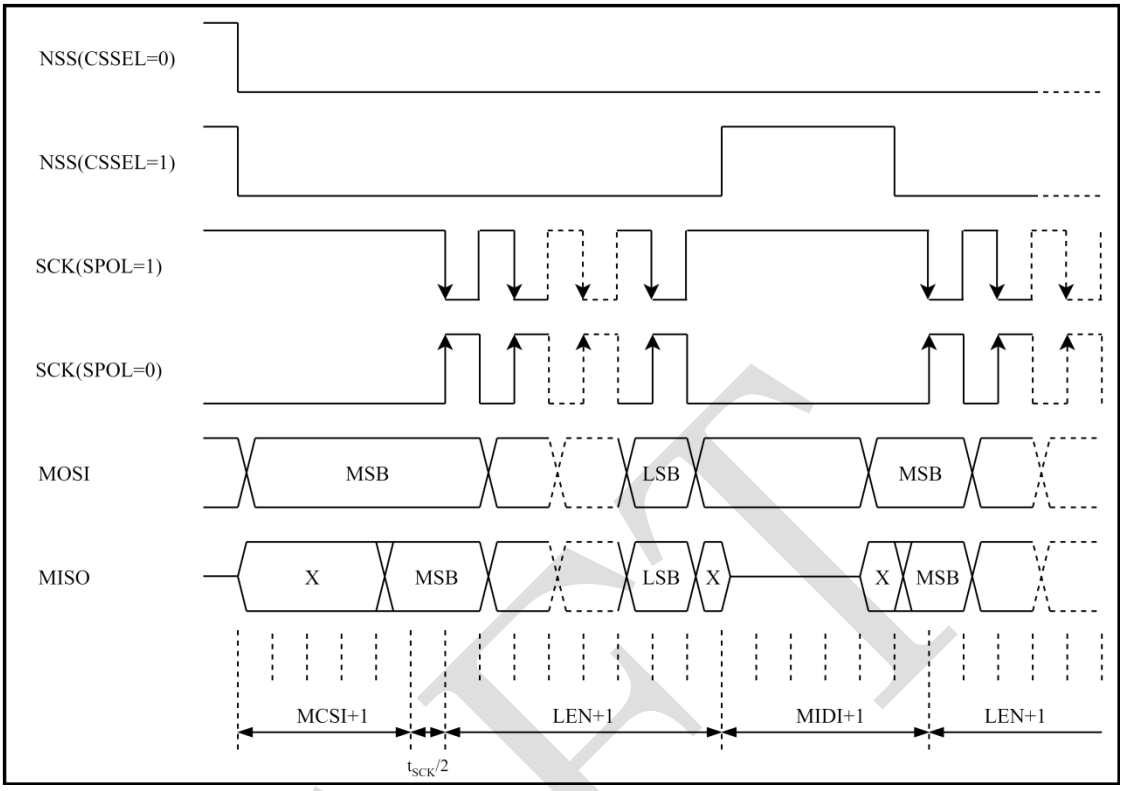


图 22-3　CPHA=0 时 NSS 管理时序图

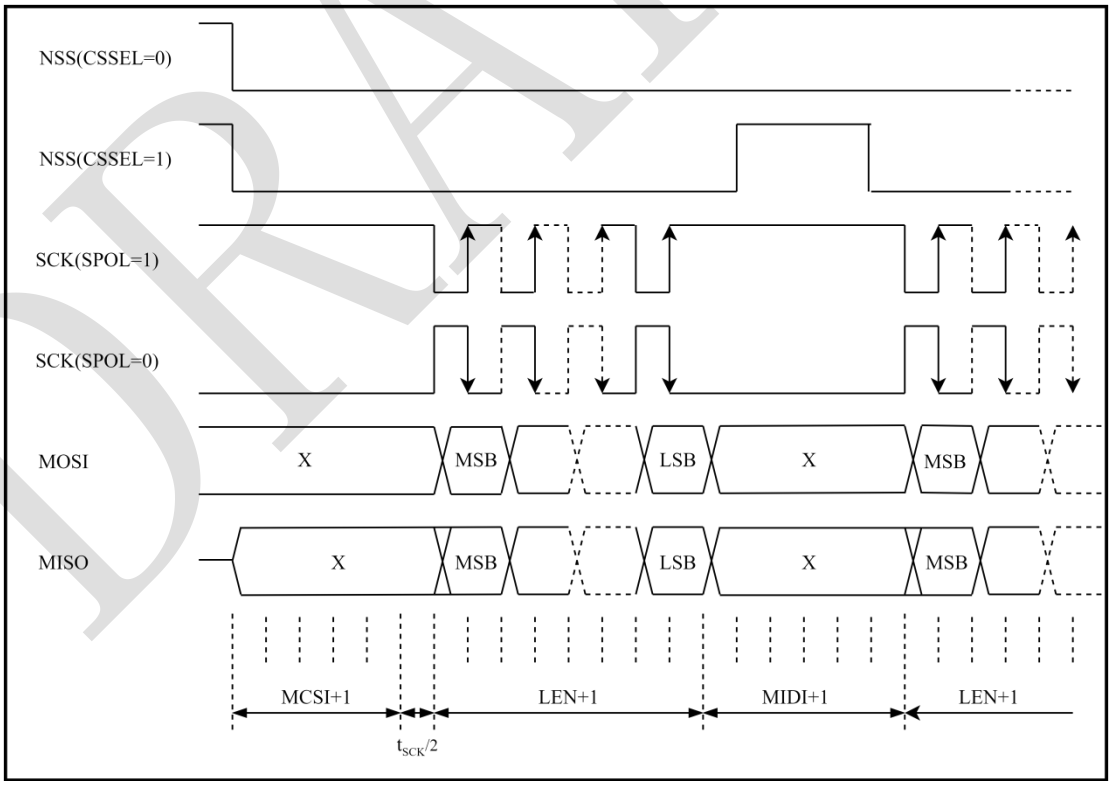


图 22-4　CPHA=1 时 NSS 管理时序图

1. MCSI 和 MIDI 的单位都是 $t_{SCK}/2$
2. 在 CSSEL=1 时，两个数据之间必然会起一个 NSS 无效时间，因此 MIDI 在 CSSEL=1 时最小值为 3，如果配置的值小于等于 3，硬件会认为是 3。

22.4.2.2 SCLK 管理

SCLK 只能由主机生成，因此波特率寄存器只有主机需要配置，如果 SPI 做从机是不需要配置的。

SPI 的波特率由 SPI_BAUD 寄存器来控制，具体配置公式如下：

$$f_{SCLK} = \frac{f_{系统时钟}}{SPI\_BAUD \times 2}$$

SPI 通信有 4 种不同的模式，不同的从设备可能在出厂是就是配置为某种模式，这是不能改变的；但我们的通信双方必须是工作在同一模式下，所以我们可以对我们的主设备的 SPI 模式进行配置，通过 CPOL（时钟极性）和 CPHA（时钟相位）来控制我们主设备的通信模式，具体如下：

- Mode0：CPOL=0，CPHA=0
- Mode1：CPOL=0，CPHA=1
- Mode2：CPOL=1，CPHA=0
- Mode3：CPOL=1，CPHA=1

时钟极性 CPOL 是用来配置 SCLK 的电平出于哪种状态时是空闲态或者有效态，时钟相位 CPHA 是用来配置数据采样是在第几个边沿：

- CPOL=0，表示当 SCLK=0 时处于空闲态，所以有效状态就是 SCLK 处于高电平时
- CPOL=1，表示当 SCLK=1 时处于空闲态，所以有效状态就是 SCLK 处于低电平时
- CPHA=0，表示数据采样是在第 1 个边沿，数据发送在第 2 个边沿
- CPHA=1，表示数据采样是在第 2 个边沿，数据发送在第 1 个边沿

例如：

- CPOL=0，CPHA=0：此时空闲态时，SCLK 处于低电平，数据采样是在第 1 个边沿，也就是 SCLK 由低电平到高电平的跳变，所以数据采样是在上升沿，数据发送是在下降沿。
- CPOL=0，CPHA=1：此时空闲态时，SCLK 处于低电平，数据发送是在第 1 个边沿，也就是 SCLK 由低电平到高电平的跳变，所以数据采样是在下降沿，数据发送是在上升沿。
- CPOL=1，CPHA=0：此时空闲态时，SCLK 处于高电平，数据采集是在第 1 个边沿，也就是 SCLK 由高电平到低电平的跳变，所以数据采集是在下降沿，数据发送是在上升沿。
- CPOL=1，CPHA=1：此时空闲态时，SCLK 处于高电平，数据发送是在第 1 个边沿，也就是 SCLK 由高电平到低电平的跳变，所以数据采集是在上升沿，数据发送是在下降沿。

关于 CPOL 和 CPHA 的时序图如图 22-5 所示：

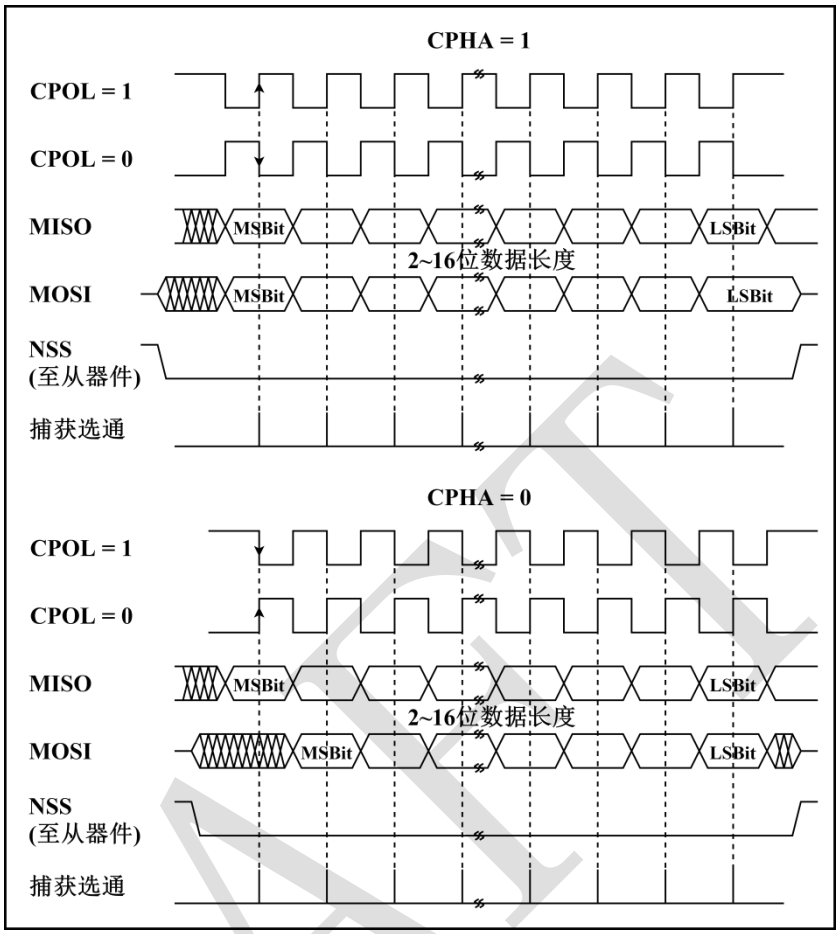


图 22-5　SPI 通信模式时序图

另外在主机接收时，主要的数据路径如下所示：

主机首先会发送 SCLK 时钟，SCLK 经过 GPIO 送往从机的 GPIO，最后送到从机，从机在接收到 SCLK 之后将数据锁存在 MISO 线上，MISO 同样的路径返回到主机被主机采样。这段路径如果超过了 SCLK 的半个周期，那么主机在采样的时候将采样不到正确的点，因此引入了 RXDLY 寄存器，对主机采样点进行延迟，每次延迟都是以一个系统时钟周期为单位，如果配置为 1，那么采样点将延迟 1 个系统时钟周期进行采样，SPI 规定了最多只能延迟 7 个系统时钟周期。具体如图 22-6 所示：

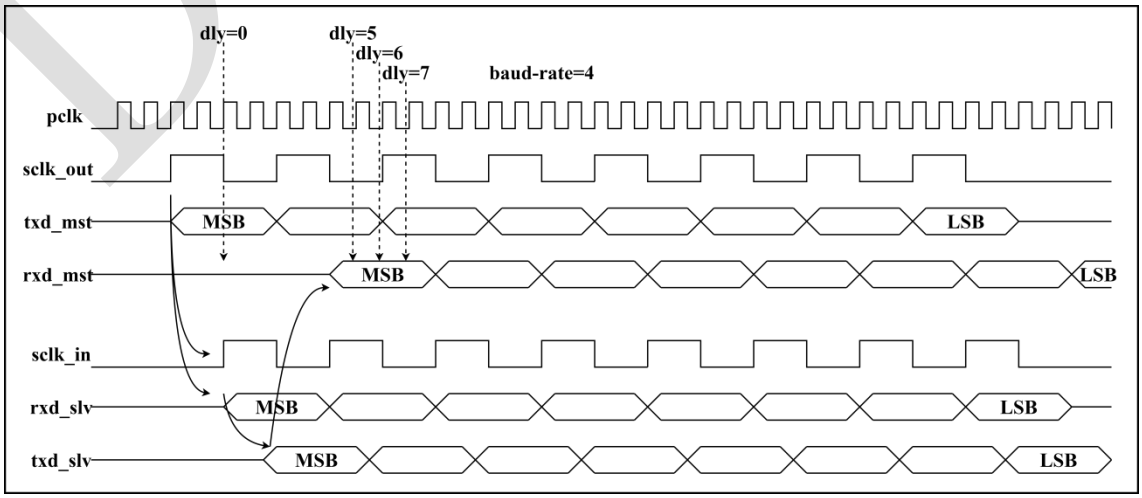


图 22-6　主机采样延迟时序图

22.4.2.3 信号管理

关于 SPI 信号的设置，有以下配置：

1. 三线四线选择

通过配置 SPI_ENABLE 的 TWE 来决定 SPI 是 3 线传输还是 4 线传输，而 SPI_ENABLE 的 SWAP 信号的意思是 MOSI 和 MISO 的引脚是否进行交换，配置为 1 就是交换，配置为 0 就是不交换。具体如表 22-2 所示

表 22-2 SPI 三线四线引脚列表

GPIO 引脚	4 线模式（TWE=0）		3 线模式（TWE=1）	
	SWAP=0	SWAP=1	SWAP=0	SWAP=1
MOSI	主机输出/从机输入	主机输入/从机输出	主机输入/主机输出	从机输入/从机输出
MISO	主机输入/从机输出	主机输出/从机输入	从机输入/从机输出	主机输入/主机输出

2. NSS 传输模式

1) 从机 NSS 输入功能

如果将 SPI_ENABLE.CSIS 位配置为 1，那么从机将不再关心 GPIO 的 NSS 的值，此时从机将使用 CSI 的值作为 NSS 的输入。

2) 软件控制输出（主机）

如果软件想要自己控制 NSS 输出，可以将 SPI_ENABLE 的 CSOS 置 1，然后 SPI_ENABLE.CSO 位的值就是 GPIO 上 NSS 的输出值，若 CSO 为 1，那么 NSS 为 1；若 CSO 为 0，那么 NSS 为 0。

3) 硬件控制输出（主机）

首先要保证 SPI_ENABLE.CSOS 位为 0 才能保证 NSS 是硬件输出模式

- NSS 极性：默认的 NSS 极性是高无效，低有效；若软件有需求，可以将 SPI_ENABLE.CPOL 位置 1，这样子 NSS 极性就改为低无效，高有效。
- NSS 模式：SPI 提供了两种 NSS 模式，可以通过配置 SPI_ENABLE.CSSEL 位的值来定，具体如图 22-7 所示

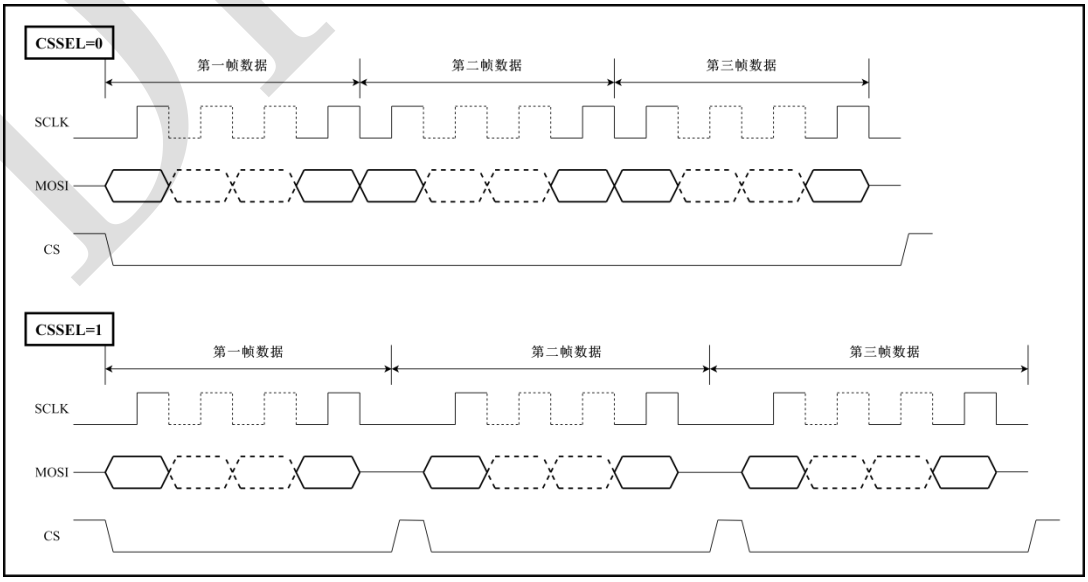


图 22-7 主机 NSS 不同模式时序图

在 CSSEL=0 时，两帧之间的 NSS 一定不会被拉高。

在 CSSEL=1 时，两帧之间的 NSS 一定会被拉高。

注意：两个数据帧之间的延长时间是可配的（SPI_TIMING.MIDI），关于 MIDI 的说明可以查看 NSS 管理章节

22.4.2.4 传输模式管理

SPI 可以通过配置 SPI_CTRL.RXEN 和 SPI_CTRL.TXEN 来决定接下来 SPI 将在哪种传输模式下运行，具体如下

1. TXEN=0，RXEN=0：屏蔽发送和接收功能，SPI 处理空闲状态
2. TXEN=1，RXEN=0：仅发送模式，会屏蔽掉接收功能
3. TXEN=0，RXEN=1：仅接收模式，会屏蔽掉发送功能
4. TXEN=1，RXEN=1：同时发送接收模式，该模式下是全双工模式

注意：当 SPI 工作在 3 线模式下，SPI 需要使用仅接收模式或者仅发送模式来确定当前的数据线是输入还是输出，若在 3 线模式下强行使用同时发送接收模式，那么会使 SPI 处于不定状态，建议不要在 3 线模式下使用同时发送接收模式。

22.4.3 SPI 配置

作为主机和作为从机的配置步骤基本一致，对于具体的模式，参考相应章节内容，若要对 SPI 进行初始化，可以执行以下步骤：

1. 对 SPI_CTRL 寄存器进行配置，配置 MSTEN（主从选择）、CPHA（串行时钟相位）、CPOL（串行时钟极性）、TXEN 和 RXEN（传输模式选择）、LSB（LSB 和 MSB 选择）、LEN（串行数据长度配置）和 RXDLY（作为主机时的采样延迟）
2. 对 SPI_FIFOCTRL 寄存器进行配置，配置 TXFTLR（TXFIFO 空中断阈值）和 RXFTLR（RXFIFO 满中断阈值）两个寄存器
3. 在 SPI 作为主机时（SPI_CTRL.MSTEN=1），还需要配置以下寄存器
   a) 对 SPI_BAUD 寄存器进行配置以确定 SPI 的波特率。
   b) 配置 SPI_TIMING 寄存器中的 MIDI 位和 MCSI 位以生成 NSS 的时序
4. 对 SPI_ENABLE 寄存器进行配置，配置引脚相关的配置，最后对 SPI_ENABLE.SPIEN 位配置为 1 以使能 SPI
5. 在 SPI 作为主机时（SPI_CTRL.MSTEN=1），还需要配置以下寄存器才能启动传输
   a) 配置 SPI_CNT.DCNT 位，告知硬件接下来要发送或者接收的数据量，若配置成 0，则为无限发送接收状态。
   b) 配置 SPI_START.START=1，启动主机传输。
6. 对 SPI_TDR 写的数据会存入 TXFIFO 中，硬件会取出后发送出去；硬件收到数据后会存

入 RXFIFO 中，此时读取 SPI_RDR 寄存器可以得到 RXFIFO 中的数据。

22.4.4 SPI 中断

在 SPI 中，不同的事件可以产生不同的中断，有关 SPI 请求的详细说明如表 22-3 示

中断名称	中断使能	置起方式	清除方式
RXFI	RXFIE	RXFIFO 中的数据量大于等于 RXFTLR 配置的值	RXFIFO 中的数据量小于 RXFTLR 配置的值
TXFI	TXFIE	TXFIFO 中的数据量小于等于 TXFTLR 配置的值	TXFIFO 中的数据量大于 TXFTLR 配置的值
RXOFI	RXOFIE	RXFIFO 已满并且有新数据即将写入 RXFIFO	对 RXOFI 位写 1 清除
RXUFI	RXUFIE	RXFIFO 已空并且软件还执行了一次读取 RXFIFO 操作	对 RXUFI 位写 1 清除
TXOFI	TXOFIE	TXFIFO 已满并且有新数据即将写入 TXFIFO	对 TXOFI 位写 1 清除
TXUFI	TXUFIE	TFIFO 已空并且软件还执行了一次读取 TFIFO 操作（从器件发送模式下溢中断）	对 TXUFI 位写 1 清除
TXDEI	TXDEIE	当 DCNT=0 时：SPI TX 或者 TRX 时，当完全发送完一个 SPI 帧且 TXFIFO 为空时起中断 当 DCNT≠0 时：SPI TX 或者 TRX 时，发送/接收完 CNT 个数据时起中断	对 TXDEI 位写 1 清除
RXDEI	RXDEIE	SPI 作为主机且模式为仅 RX 模式时，当 SPI 接收到 CNT 个数据时该位置起	对 RXDEI 位写 1 清除
TXCI	TXCIE	SPI 发送完 1 帧结束的时候该位置起	对 TXCI 位写 1 清除
RXCI	RXCIE	SPI 接收完 1 帧结束的时候该位置起	对 RXCI 位写 1 清除

22.5 寄存器定义

22.5.1 寄存器列表

SPI 基地址：

SPI0(0x40012000) SPI1(0x40013000)

偏移	实例地址	名称	默认值	描述
0x00	SPI 基地址+0x00	SPI_ENABLE	0x0000A000	SPI 使能寄存器
0x04	SPI 基地址+0x04	SPI_CTRL	0x00000000	SPI 控制寄存器
0x08	SPI 基地址+0x08	SPI_BAUD	0x00000000	SPI 波特率寄存器
0x10	SPI 基地址+0x10	SPI_FIFOCTRL	0x00000000	SPI FIFO 控制寄存器
0x14	SPI 基地址+0x14	SPI_CNT	0x00000000	SPI 主机传输数据量寄存器
0x18	SPI 基地址+0x18	SPI_RCNT	0x00000000	SPI 主机传输剩余数据量寄存器
0x20	SPI 基地址+0x20	SPI_START	0x00000000	SPI 主机传输开始寄存器
0x24	SPI 基地址+0x24	SPI_TIMING	0x00000000	SPI 时序寄存器
0x30	SPI 基地址+0x30	SPI_INTEN	0x00000000	SPI 中断使能寄存器
0x34	SPI 基地址+0x34	SPI_INT	0x00000002	SPI 中断寄存器
0x38	SPI 基地址+0x38	SPI_STATUS	0x20200000	SPI 状态寄存器
0x40	SPI 基地址+0x40	SPI_TDR	0x00000000	SPI 数据写寄存器
0x44	SPI 基地址+0x44	SPI_RDR	0x00000000	SPI 数据读寄存器
0x48	SPI 基地址+0x48	SPI_UDRDR	0x00000000	SPI 从模式下溢时寄存器

22.5.2 寄存器描述

22.5.2.1 SPI 使能寄存器（SPI_ENABLE）

- 名称：**SPI Enable Register**
- 偏移地址：**0x00**
- 默认值：**0x0000A000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI	CSIS	CSO	CSOS			DTE	DRE		SWAP	CSPOL	CSSEL	TWE	TFR	RFR	SPIEN
R/W	R/W	R/W	R/W			R/W	R/W		R/W	R/W	R/W	R/W	W	W	R/W
0x1	0x0	0x1	0x0			0x0	0x0		0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[15] CSI	**SPI NSS 软件输入（SPI Chip Select Software Input）** 0：NSS 输入 0，此时是 NSS 有效 1：NSS 输入 1，此时是 NSS 无效 Note：该位不受 CSPOL 控制 Note：该位只在 MSTEN 配置成 0 才有效
[14] CSIS	**SPI NSS 输入选择（SPI Chip Select Input Select）** 从机时选择 NSS 输入的方式 0：GPIO 的 NSS 1：CSI 的值（不受 CSPOL 控制） Note：该位只在 MSTEN 配置成 0 才有效
[13] CSO	**SPI NSS 软件输出（SPI Chip Select Output）** 0：NSS 输出有效值 1：NSS 输出无效值 Note：软件只需要配置 NSS 有效和无效值，硬件会根据 CSPOL 输出相应的值 Note：该位只在 MSTEN 配置成 1 才有效
[12] CSOS	**SPI NSS 软件输出（SPI Chip Select Output Select）** 0：使用硬件的 NSS 输出 1：使用软件的 NSS 输出 Note：该位只在 MSTEN 配置成 1 才有效
[9] DTE	**SPI 发送 DMA 使能（SPI TX DMA Enable）** 该位启用/禁用发送 FIFO DMA 通道 0：禁用发送 DMA 通道 1：使能发送 DMA 通道
[8] DRE	**SPI 接收 DMA 使能（SPI RX DMA Enable）** 该位启用/禁用接收 FIFO DMA 通道 0：禁用接收 DMA 通道 1：使能接收 DMA 通道
[6] SWAP	**SPI MOSI 和 MISO 引脚交换（SPI MOSI and MISO pins swapped）** 0：不交换 1：交换 Note：仅在 SPIEN 配置为 0 的时候才能配置
[5] CSPOL	**SPI NSS 无效状态极性选择（SPI Chip Select Polarity Select）** 0：NSS 为高无效 1：NSS 为低无效 Note：仅在 SPIEN 配置为 0 的时候才能配置
[4]	**SPI NSS 模式选择（SPI SPI Chip Select Mode Select）**

CSSEL	0：NSS 就不会拉高除非传输结束或者配置 STOP 停止传输 1：数据发送或接收的两个数据之间 NSS 都会拉高 Note：该位只在 MASTER_EN 配置成 1 才有效
[3] TWE	**SPI 三线使能（SPI three-Wire Enable）** 0：GPIO 采用 4 线接法和 SPI 交互 1：GPIO 采用 3 线接法和 SPI 交互 Note：仅在 SPIEN 配置为 0 的时候才能配置 Note：3 线模式下只能进行半双工，如果 TXEN 和 RXEN 都配置为 1(全双工模式)时，则会被认为是仅 TX 模式
[2] TFR	**SPI TXFIFO 复位（SPI TXFIFO RESET）** TXFIFO 复位信号，只写信号 0：TXFIFO 正常 1：TXFIFO 复位
[1] RFR	**SPI RXFIFO 复位（SPI RXFIFO RESET）** RXFIFO 复位信号，只写信号 0：RXFIFO 正常 1：RXFIFO 复位
[0] SPIEN	**SPI 使能位（SPI Enable）** 0：SPI 不使能 1：SPI 使能

22.5.2.2 SPI 控制寄存器（SPI_CTRL）

- **名称：SPI Control Register**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													RXDLY		
													R/W		
													0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				LEN				UDRCFG	LSB	RXEN	TXEN	SHZOE	CPOL	CPHA	MSTEN
				R/W				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0x0				0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[18:16] RXDLY	**SPI 接收数据采样延迟（SPI Master Rx Data Delay Chain ）** 主机接收时采样点的延迟，单位为 pclk 时钟周期 0x0：不延迟采样 0x1：延迟 1 个时钟周期采样 0x2：延迟 2 个时钟周期采样 ……. 0x7：延迟 7 个时钟周期采样
[11:8] LEN	**SPI 串行数据长度（SPI Serial Data Length Select）** 0x0：Reserved 0x1：2 位串行数据传输 0x2：3 位串行数据传输 0x3：4 位串行数据传输 0x4：5 位串行数据传输 0x5：6 位串行数据传输 …… 0xF：16 位串行数据传输
[7] UDRCFG	**SPI 下溢条件时从器件发送器行为 (SPI Behavior of slave transmitter at underrun condition)** 0：从器件发送一个常数，该常数由用户在 UDRDR 寄存器中定义 1：从器件重复其上一次发送的数据帧

[6] LSB	**SPI LSB 使能（LSB Enable）** 0：MSB 1：LSB Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值
[5] RXEN	**SPI 接收使能（SPI RX Enable）** 0：不使能 1：使能
[4] TXEN	**SPI 发送使能（SPI TX Enable）** 0：不使能 1：使能
[3] SHZOE	**SPI 从机输出高阻态使能位（SPI Slave High-Z Output Enable ）** 仅从机有效，当 SLV_OE 为 1，从机的输出将会始终输出高阻态。 0：从机输出正常 1：从机输出高阻态
[2] CPOL	**SPI 串行时钟极性（SPI SCLK Polarity）** 0：串行时钟无效状态为低 1：串行时钟无效状态为高 Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值
[1] CPHA	**SPI 串行时钟相位（SPI SCLK Phase）** 表示 SPI 在 SCLK 第几个边沿开始采样 0：SPI 在 SCLK 第 1 个边沿开始采样 1：SPI 在 SCLK 第 2 个边沿开始采样 Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值
[0] MSTEN	**SPI 的主机使能位（SPI Master Enable）** 0：SPI 作为从机 1：SPI 作为主机 Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值

22.5.2.3 SPI 波特率寄存器（SPI_BAUD）

- **名称：SPI Baud Rate Registers**
- **偏移地址：0x08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				BAUD											
				R/W											
				0x0											

字段	说明
[11:1] BAUD	**SPI 时钟分频器（SPI Baud Rate）** SCLK 的频率从以下公式得出： FSCLK = F 系统时钟/(BAUD*2) 其中 PCLK 是 SPI 模块的系统时钟 取值范围为：1~2047 Note：该位只在 MSTEN 配置成 1 才有效 Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值

22.5.2.4 SPI FIFO 控制寄存器（SPI_FIFOCTRL）

- 名称：SPI FIFO Control Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RXFTLR				TXFTLR			
								R/W				R/W			
								0x00				0x00			

字段	说明
[7:4] RXFTLR	**SPI RXFIFO 满中断阈值（SPI RXFIFO Full Threshold）** 0x0：FIFO 内有 1 个或 1 个以上数据就会触发 RX-FIFO 满中断 0x1：FIFO 内有 2 个或 2 个以上数据就会触发 RX-FIFO 满中断 …… 0xe：FIFO 中有 15 个或 15 个以上数据就会触发 RX-FIFO 满中断 0xf：FIFO 中有 16 个数据就会触发 RX-FIFO 满中断
[3:0] TXFTLR	**SPI TXFIFO 空中断阈值（SPI TXFIFO Empty Threshold）** 0x0：FIFO 中没有数据就会触发 TX-FIFO 空中断 0x1：FIFO 中有 1 个或 1 个以下就会触发 TXFIFO 空中断 …… 0xe：FIFO 中有 14 个或 14 个以下就会触发 TXFIFO 空中断 0xf：FIFO 中有 15 个或 15 个以下就会触发 TXFIFO 空中断

22.5.2.5 SPI 主机传输数据量寄存器（SPI_CNT）

- 名称：SPI Count Registers
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCNT															
R/W															
0x0															

字段	说明
[15:0] DCNT	**SPI 主机传输数据量（SPI Transmittion Data Count）** 设置 SPI 连续发送接收的数据帧数，配置完毕后配置 START 为 1 启动传输；若 CNT=0 时配置 START 为 1，则进入无限发送接收模式。 Note：仅当 START=0 时才可以配置 Note：该位只在 MSTEN 配置成 1 才有效

22.5.2.6 SPI 主机传输剩余数据量寄存器（SPI_RCNT）

- 名称：SPI Remain Count Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCNT															
R															
0x0															

字段	说明
[15:0] RCNT	**SPI 主机传输剩余数据量（SPI Transmittion Remain Count）** DCNT 配置不为 0 时，RCNT 初始化会等于 DCNT 的值，随着数据发送接收会递减，直到为 0 停止递减 DCNT 配置为 0 时，RCNT 会实时监测发送接收的数量，当计数到 0xffff 时会清 0 重新开始计算。 Note：该位只在 MSTEN 配置成 1 才有效

22.5.2.7 SPI 主机传输开始寄存器（SPI_START）

- 名称：SPI Count Start Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														STOP	START
														R/WAC	R/WAC
														0x0	0x0

字段	说明
[1] STOP	**SPI 主机传输停止（SPI Master Transmittion Stop）** 做主机时，写入 1 时，在发完或者接收完当前帧的数据时会停止发送接收，并且清除 START 位，同时 NSS 会被置起。如果对该位写了 1，那么软件需要等到 STOP 清 0 后才能进行下一次的传输。 Note：只有在 START=1 时才能被写入 Note：完全停止传输时会被自动清 0
[0] START	**SPI 主机传输开始（SPI Master Transmittion Start）** 在 DCNT 不等于 0 时，写入 1 启动 SPI 发送/接收，TXDEI/RXDEI 标志置位后此位清 0，发送/接收过程中也可以被 STOP 清除，同时会置起 TXDEI/RXDEI 标志。 在 DCNT 等于 0 时，写入 1 启动 SPI 发送/接收，此时 START 不会被硬件清除，可以通过配置 STOP 来清除 START Note： 1. 若 SPI 配置为全双工模式（TXEN 和 RXEN 都配置为 1）或者仅 TX 模式（TXEN 配置为 1，RXEN 配置为 0）时，则会起 TXDEI 标志； 2. 若 SPI 配置为仅 RX 模式（TXEN 配置为 0，RXEN 配置为 1）时，则会起 RXDEI 标志 Note：只有在 START=0 时才能被写入

22.5.2.8 SPI 时序寄存器（SPI_TIMING）

- **名称：SPI Timing Register**
- **偏移地址：0x24**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													MCSI		
													R/W		
													0x0		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													MIDI		
													R/W		
													0x0		

字段	说明
[20:16] MCSI	**SPI 主模式 NSS 空闲（SPI Master Chip Select Idleness)** 在主模式下额外插入到 NSS 有效边沿和第一个数据事务开始之间的额外延迟（以半个 SPI 时钟周期为时间单位）。 0x0：添加 1 个时间单位 0x1：添加 2 个时间单位 0x2：添加 3 个时间单位 ... 0x1F：添加 32 个时间单位 Note：该位只在 MSTEN 配置成 1 才有效 Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值
[4:0] MIDI	**SPI 主模式数据间空闲（SPI Master Inter-Data Idleness)** 指定主模式下在两个连续数据帧之间插入的最小时间延迟（以半个 SPI 时钟周期为时间单位）。 0x0：无额外延迟 0x1：添加了 2 个时间单位 0x2：添加了 3 个时间单位 ... 0x1F：添加了 32 个时间单位 Note：若在 CSSEL=1 时配置值小于 3 时,读取的值还是原值，但是会被硬件认为是 3 Note：该位只在 MSTEN 配置成 1 才有效 Note：在 SPI 传输过程中，硬件不会阻止软件去更改该位的值，更改后会立即使用新的配置值，当前数据帧可能会不正常，建议在 START=0 的时候再配置该值

22.5.2.9 SPI 中断使能寄存器（SPI_INTEN）

- 名称：SPI Interrupt Enable Registers
- 偏移地址：0x30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RXCIE	TXCIE	RXDEIE	TXDEIE			TXUFIE	TXOFIE	RXUFIE	RXOFIE	TXEIE	RXFIE
				R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
				0x0	0x0	0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[11] RXCIE	**SPI 接收帧完成中断使能（SPI RX Complete Interrupt Enable）** 0：不使能 1：使能
[10] TXCIE	**SPI 发送帧完成中断使能（SPI TX Complete Interrupt Enable）** 0：不使能 1：使能
[9] RXDEIE	**SPI 接收完成中断使能（SPI RX Done Interrupt Enable）** 0：不使能 1：使能
[8] TXDEIE	**SPI 发送完成中断使能（SPI TX Done Interrupt Enable）** 0：不使能 1：使能
[5] TXUFIE	**SPI 从器件发送模式下溢中断使能（SPI Slave Transmission Mode Underflow Interrupt Enable）** 0：不使能 1：使能
[4] TXOFIE	**SPI TXFIFO 溢出中断使能（SPI TXFIFO Overflow Interrupt Enable）** 0：不使能 1：使能
[3] RXUFIE	**SPI RXFIFO 下溢中断使能（SPI RXFIFO Underflow Interrupt Enable）** 0：不使能 1：使能
[2] RXOFIE	**SPI RXFIFO 溢出中断使能（SPI RXFIFO Overflow Interrupt Enable）** 0：不使能 1：使能
[1] TXEIE	**SPI TXFIFO 空阈值中断使能（SPI TXFIFO Empty Threshold Interrupt Enable）** 0：不使能 1：使能
[0] RXFIE	**SPI RXFIFO 满阈值中断使能（SPI RXFIFO Full Threshold Interrupt Enable）** 0：不使能 1：使能

22.5.2.10 SPI 中断寄存器（SPI_INT）

- 名称：SPI Interrupt Registers
- 偏移地址：0x34
- 默认值：0x00000002
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RXCI	TXCI	RXDEI	TXDEI			TXUFI	TXOFI	RXUFI	RXOFI	TXEI	RXFI
				R/W1C	R/W1C	R/W1C	R/W1C			R/W1C	R/W1C	R/W1C	R/W1C	R	R
				0x0	0x0	0x0	0x0			0x0	0x0	0x0	0x0	0x1	0x0

字段	说明
[11] RXCI	**SPI 接收帧完成中断（SPI RX Complete Interrupt）** SPI 接收完 1 帧结束的时候该位置起 0：未接收完毕 1：接收完毕 Note：对该位写 1 将清除该中断
[10] TXCI	**SPI 发送帧完成中断（SPI TX Complete Interrupt）** SPI 发送完 1 帧结束的时候该位置起 0：未发送完毕 1：发送完毕 Note：对该位写 1 将清除该中断
[9] RXDEI	**SPI 接收完成中断（SPI RX Done Interrupt ）** SPI 作为主机且模式为仅 RX 模式时，当 SPI 接收到 CNT 个数据时该位置起 0：未接收完毕 1：接收完毕 Note：该位仅在 MSTEN=1 时有效 Note：对该位写 1 将清除该中断
[8] TXDEI	**SPI 发送完成中断（SPI TX Done Interrupt ）** 当 CNT=0 时或者 MASTER_EN=0 时： SPI TX 时，当完全发送完一个 SPI 帧且 TXFIFO 为空时起中断 0：未发送完毕 1：发送完毕 当 CNT 不等于 0 时： SPI TX 或者 TRX 时，发送/接收完 CNT 个数据时起中断 0：未发送/接收完成 1：发送/接收完成 Note：对该位写 1 将清除该中断
[5] TXUFI	**SPI 从器件发送模式下溢中断（SPI Slave Transmission Mode Underflow Interrupt）** 0：无下溢错误 1：下溢错误 Note：对该位写 1 将清除 RXUFI 中断。 Note：该位仅在 MASTER_EN=0 时有效
[4] TXOFI	**SPI TXFIFO 溢出中断（SPI TXFIFO Overflow Interrupt）** 当 TXFIFO 已满并且新数据被写入 FIFO 中时，会起溢出错误中断。此时 FIFO 中的数据将保留，而接收的新数据将丢失。 0：无上溢错误 1：上溢错误 Note：对该位写 1 将清除 RXOFI 中断。
[3] RXUFI	**SPI RXFIFO 下溢中断（SPI RXFIFO Underflow Interrupt）** 当 RXFIFO 已空并且但是还是有从 FIFO 取数的操作，会起下溢错误中断。 0：无下溢错误

	1：下溢错误 Note：对该位写 1 将清除 RXUFI 中断。
[2] RXOFI	**SPI RXFIFO 溢出中断（SPI RXFIFO Overflow Interrupt）** 当 RXFIFO 已满并且新数据被接收写入 FIFO 中时，会起溢出错误中断。此时 FIFO 中的数据将保留，而接收的新数据将丢失。 0：无上溢错误 1：上溢错误 Note：对该位写 1 将清除 RXOFI 中断。
[1] TXEI	**SPI TXFIFO 空阈值中断（SPI TXFIFO Empty Threshold Interrupt ）** 与 TXFTLR 寄存器有关，当 TXFIFO 中的数据量小于或者等于预设值（TXFTLR）时置位。 0：TXFIFO 数据量大于预设值 1：TXFIFO 数据量小于等于预设值 Note：当 TXFIFO 数据量大于预设值时 TXEI 中断自动清 0
[0] RXFI	**SPI RXFIFO 满阈值中断（SPI RXFIFO Full Threshold Interrupt ）** 与 RXFTLR 寄存器有关，当 RXFIFO 中的数据量大于或等于预设值（RXFTLR）时置位。 0：RXFIFO 数据量小于预设值 1：RXFIFO 数据量大于等于预设值 Note：当 RXFIFO 数据量小于预设值时 RXFI 中断自动清 0

22.5.2.11 SPI 状态寄存器（SPI_STATUS）

- **名称：SPI Status Register**
- **偏移地址：0x38**
- **默认值：0x20200000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFF	RFE	RXFLR						TFF	TFE	TXFLR				
	R	R	R						R	R	R				
	0x00	0x01	0x00						0x00	0x01	0x00				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															BUSY
															R
															0x00

字段	说明
[30] RFF	**SPI RXFIFO 满标志（SPI RXFIFO Full Flag）** 0：接收 FIFO 未满 1：接收 FIFO 已满 Note：当 RX FIFO 不为满时，该位被清除。
[29] RFE	**SPI RXFIFO 空标志（SPI RXFIFO Empty Flag）** 0：接收 FIFO 不为空 1：接收 FIFO 为空 Note：当 RX FIFO 不为空时，该位被清除。
[28:24] RXFLR	**SPI RXFIFO 实时剩余数据量（SPI RXFIFO Level Register）** 显示当前还有多少有效数据在 RXFIFO 中
[22] TFF	**SPI TXFIFO 满标志（SPI TXFIFO Full Flag）** 0：发送 FIFO 未满 1：发送 FIFO 为满 Note：当 TX FIFO 不为满时，该位被清除。
[21] TFE	**SPI TXFIFO 空标志（SPI TXFIFO Empty Flag）** 0：发送 FIFO 不为空 1：发送 FIFO 为空 Note：当 TX FIFO 不为空时，该位被清除。
[20:16] TXFLR	**SPI TXFIFO 实时剩余数据量（SPI TXFIFO Level Register）** 显示当前还有多少有效数据在 TXFIFO 中
[0]	**SPI 状态机忙状态（SPI FSM Busy Status）**

BUSY	0：状态机空闲 1：状态机忙碌

22.5.2.12 SPI 数据写寄存器（SPI_TDR）

- **名称：SPI TXFIFO Data Registers**
- **偏移地址：0x40**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD															
W															
0x0															

字段	说明
[15:0] TD	**SPI 数据写寄存器（SPI Data Write Register）** SPI 要发送的数据 Note：TD 的实际有效位宽和 LEN 的配置有关

22.5.2.13 SPI 数据读寄存器（SPI_RDR）

- **名称：SPI RXFIFO Data Registers**
- **偏移地址：0x44**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD															
R															
0x0															

字段	说明
[15:0] RD	**SPI 数据读寄存器（SPI Data Read Register）** SPI 接收的数据 Note：RD 的实际有效位宽和 LEN 的配置有关

22.5.2.14 SPI 从模式下溢时寄存器（SPI_UDRDR）

- 名称：**SPI Underrun Data Register**
- 偏移地址：**0x48**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDRDR															
R/W															
0x0															

字段	说明
[15:0] UDRDR	**SPI 从模式下溢时的数据 (SPI Data At Slave Underflow Condition)** 从模式下如果发生 TXFIFO 下溢，如果软件将 UDRCFG 置 1，那么从机将固定发送 UDRDR 数据

23 外部并行接口（XIF）

23.1 简介

外部并行接口（XIF）是一个高速同步并行接口，它可以有效进行高效的数据传输。主要针对的是 AD76 系列芯片的数据采样和接收。

23.2 主要特性

- 支持数据位宽为 16
- 速度最高可以达到所用系统时钟的 2 分频
- 支持 DMA 传输
- 支持深度为 8 的 RXFIFO
- 所有信号的时序都做成软件可配置
- 一帧最多可以支持 255 个数据
- 支持所有的 AD76 系列的采样

23.3 结构框图

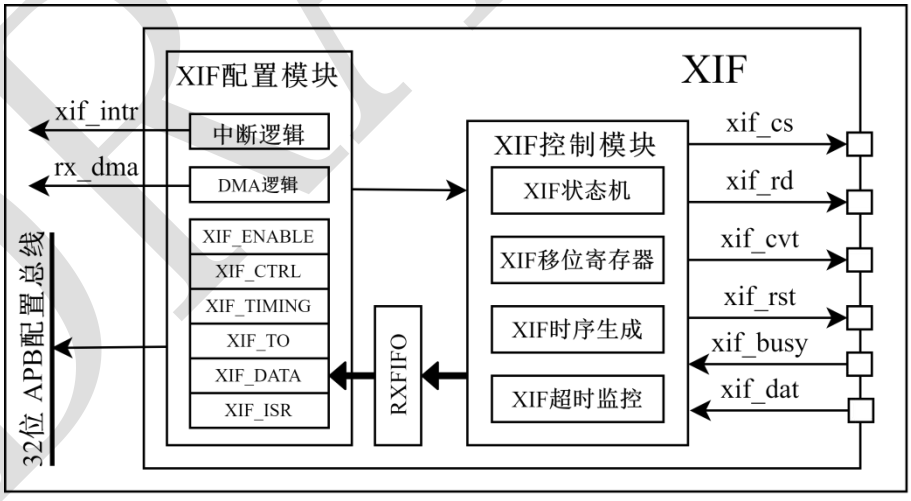


图 23-1 XIF 结构框图

23.4 功能描述

23.4.1 XIF 信号分析

表 23-1 为 XIF 线上信号分析，其中 O 代表对于 XIF 来说属于 GPIO 输出方向；I 代表对于 XIF 来说属于 GPIO 输入方向。

表 23-1 XIF 信号列表

信号名称	方向	信号描述
XIF_RST	O	复位脉冲输出。对于 AD76 系列的器件，RESET 的上升沿将会复位器件，器件应该在上电的时候收到一个 RESET 脉冲，脉冲宽度典型值为 50ns。如果在转换期间有一个 RESET 脉冲，那么转换将被中断；如果在读取期间有一个 RESET 脉冲，输出寄存器的内容将复位至全 0。
XIF_CVT	O	转换脉冲输出。对于 AD76 系列的器件，可能存在多个通道，每个通道有一个 CONVST 信号，如果需要所有输入通道同时采样，可以将所有的 CONVST 短接在一起在和 XIF_CVT 相连；如果需要对某个输入通道控制采样，则 XIF_CVT 可以和相应通道的 CONVST 相连。当 CONVST 从低电平变为高电平时，相应通道的采样会保持启动状态。
XIF_BUSY	I	繁忙脉冲输入。对于 AD76 系列的器件，所有的 CONVST 信号均达到上升沿之后，器件会将 BUSY 信号变为逻辑高电平，表示转换过程已经开始。BUSY 会保持高电平，直到所有通道的转换过程完成为止，BUSY 下降沿表示转换数据正在被锁存至输出数据寄存器，经过最少 25ns 之后可以被读取。
XIF_CS	O	片选输出。对于 AD76 系列的器件，CS 为低电平时才使能数据帧传输。只有当 CS 和 RD 均处于逻辑低电平，才会使转换数据输出在并行总线上。
XIF_RD	O	读取控制输出。对于 AD76 系列的器件，只有当 CS 和 RD 均处于逻辑低电平，才会使转换数据输出在并行总线上。
XIF_DAT[15:0]	I	读取数据输入。

23.4.2 XIF 时序分析

图 23-2 显示了读取 AD76 系列芯片数据时的时序图

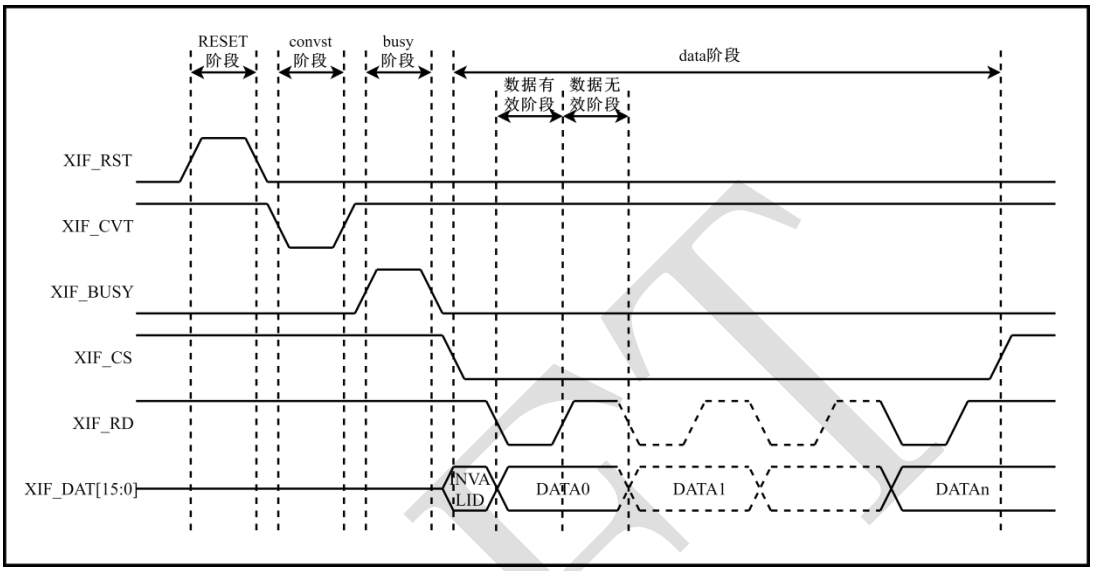


图 23-2　XIF 读取时序图

对于一个标准的 AD76 芯片读取数据时序，如图 23-2 所示一共有 4 个阶段，各个阶段说明如下所示：

1. RESET 阶段

对于 AD76 系列的器件，RESET 的上升沿将会复位器件，器件应该在上电的时候收到一个 RESET 脉冲，脉冲宽度典型值为 50ns。在 XIF 中，对于脉冲宽度由 XIF_TIMING.RSTTIME 位来配置，配置的单位为系统时钟周期，若配置为 0，则启动之后不会产生 RESET 信号，直接进入 CONVST 阶段；实际值即为配置值。

2. CONVST 阶段

对于 AD76 系列的器件，CONVST 相当于一个通道采样开始信号，最短的 CONVST 低电平为 25ns，在 CONVST 信号上升沿之后，通道的采样才会保持启动状态。在 XIF 中，对于 CONVST 低电平的持续时间由 XIF_TIMING.CONLTIME 位来配置，配置的单位为系统时钟周期，实际值等于配置值+1，也就是说，如果配置为 0，那么低电平的持续时间为 1 个系统时钟周期，如果配置为 n，那么低电平的持续时间为 n+1 个系统时钟周期。

3. BUSY 阶段

对于 AD76 系列的器件，CONVST 信号均达到上升沿之后，器件会将 BUSY 信号变为逻辑高电平，表示转换过程已经开始。BUSY 会保持高电平，直到所有通道的转换过程完成为止，BUSY 下降沿表示转换数据正在被锁存至输出数据寄存器，经过最少 25ns 之后可以被读取。该阶段 XIF 需要等到 BUSY 的下降沿才能准备下一步，因此可以设置等待超时的时间，可以通过配置 XIF_TO.PTOT 位来设置超时，若等待超时，硬件会将 XIF_ISR.BTOI 位置 1，超时之后 XIF 会回到最初的状态，需要重新对 XIF 进行初始化配置，并重新发送 CONVST。

4. DATA 阶段

该阶段 CS 固定为低，当 RD 为低时为数据有效阶段，当 RD 为高时为数据无效阶段。XIF 会在 RD 的上升沿进行采样数据。在这个阶段，RD 低电平的持续时间还有 RD 高电平的持续时间都是可配置的，RD 低电平的持续时间可以通过 XIF_TIMING.RDLTIME 位来控制，配置的单位为系统时钟周期，实际值等于配置值+1；RD 高电平的持续时间可以通过 TIMING.RDHTIME 位来控制，配置的单位为系统时钟周期，实际值等于配置值+1；

23.4.3 Reload 模式

在图 23-2 的 DATA 阶段中，硬件对 RD 的翻转不是无限的，在接收完指定数量的数据之后会触发中断，数量可以通过 XIF_CTRL.DCNT 位来配置，触发的是 RXDI 中断，触发中断之后 XIF 会进入 IDLE 状态，此时除非重新启动 XIF，否则不会继续发送 XIF 时序。

如果开启了 reload 模式（配置 XIF_ENABLE.RELOAD 位为 1）时，将不会再触发 RXDI 中断，当 DATA 阶段结束之后会重新发起 CONVST 去再收 DCNT 个数据，设置好时序后软件可以一直通过 FIFO 的空满状态取数据。

23.4.4 FIFO 模式

XIF 是工作在 FIFO 模式下的，XIF 中有一个接收 FIFO（RXFIFO），用来储存接收到的数据，因为数据的最大位宽为 16bit 的，因此 FIFO 的默认位宽是 16bit 的，最多可以存储 8 个数据。

FIFO 可以配置触发 RX 中断（XIF_ISR.RXFI）的 FIFO 阈值，阈值可以通过 XIF_CTRL.RXFTLR 位来配置，一共有以下两种情况：

1. RXFIFO 中的数据数量小于 RXFTLR 的配置值，RXFI 中断将不会置起
2. RXFIFO 中的数据数量大于等于 RXFTLR 的配置值，RXFI 中断将会被置起

此外 FIFO 还有溢出错误的中断（XIF_ISR.RXOFI），当 FIFO 已满且此时还有新的数据即将写入 FIFO，RXOFI 会被置起，此时 FIFO 中的数据将被保留，而即将写入的新数据会被丢弃。

23.4.5 XIF 中断

在 XIF 通信过程中，不同的事件可以产生不同的中断，有关 XIF 中断请求的详细说明如表 23-2 所示。

表 23-2 XIF 中断列表

中断名称	中断使能	置起方式	清除方式
RXFI	RXFIE	FIFO 中的数据量大于等于 RXFTLR 配置的值	FIFO 中的数据量小于 RXFTLR 配置的值
RXOFI	RXOFIE	FIFO 已满并且有新数据即将写入 FIFO	对 RXOFI 位写 1 清除
RXDI	RXDIE	XIF 已经接收完 DCNT 个数据	对 RXDI 位写 1 清除
BTOI	BTOIE	XIF 等待 busy 的时间超过了 PTOT 的配置值	对 BTOI 位写 1 清除

23.5 寄存器描述

23.5.1 寄存器列表

XIF 基地址：

XIF(0x40016000)

偏移	实例地址	名称	默认值	描述
0x00	XIF 基地址+0x00	XIF_ENABLE	0x00000000	XIF 使能寄存器
0x04	XIF 基地址+0x04	XIF_CTRL	0x00000000	XIF 控制寄存器
0x08	XIF 基地址+0x08	XIF_TIMING	0x00000000	XIF 时序寄存器
0x0C	XIF 基地址+0x0C	XIF_TO	0x00000000	XIF 超时设置寄存器
0x10	XIF 基地址+0x10	XIF_DATA	0x00000000	XIF 读取数据寄存器
0x14	XIF 基地址+0x14	XIF_ISR	0x00000010	XIF 中断状态寄存器

23.5.2 寄存器描述

23.5.2.1 XIF 使能寄存器（XIF_ENABLE）

- 名称：**XIF Enable Register**
- 偏移地址：**0x00**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												RELOAD	SRST	DMAEN	ENABLE
												R/W	WC	R/W	R/W
												0x00	0x00	0x00	0x00

字段	说明
[3] RELOAD	**重复模式（Reload Mode）** 0：不启动 reload 模式，接收完一次数据之后将不会继续接收，不会再发 CONVST 1：启动 reload 模式，接收完数据之后继续发送 CONVST 继续下一次接收
[2] SRST	**XIF 软复位（XIF Software Reset）** 0：写入 0 无效。该位总是读回 0。 1：强制 XIF 软复位。所有寄存器将返回其默认状态，并且所有 XIF 信号将进入无效状态。任何挂起的访问都将丢失，包括写入 FIFO 中的数据。DMA 的任何停止条件都将被释放。 Note：自清 0
[1] DMAEN	**XIF DMA 使能（XIF DMA Enable）** 0：XIF DMA 关闭，此时只能使用 CPU 的方式读 FIFO 1：XIF 使能，此时用 DMA 或者 CPU 的方式都可以读 FIFO
[0] ENABLE	**XIF 使能（XIF Enable）** 0：XIF 关闭，关闭后 XIF 将不再接收数据 1：XIF 使能，使能后 XIF 会进入工作，在此之前需要先把 XIF 配置好

23.5.2.2 XIF 控制寄存器（XIF_CTRL）

- 名称：XIF Control Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													DCS		
													R/W		
													0x00		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTOIE	RXDIE	RXOFIE	RXFIE		RXFTLR			DCNT							
R/W	R/W	R/W	R/W		R/W			R/W							
0x0	0x0	0x0	0x0		0x00			0x00							

字段	说明
[18:16] DCS	**XIF 采样时间点延后选择（XIF Delay Chain Select）** 0：不延迟 1：延迟 1 个系统时钟 2：延迟 2 个系统时钟 3：延迟 3 个系统时钟 4：延迟 4 个系统时钟 5：延迟 5 个系统时钟 6：延迟 6 个系统时钟 7：延迟 7 个系统时钟
[15] BTOIE	**BTOI 中断使能（XIF Wait BUSY Timeout Interrupt Enable）** 0：不使能 1：使能
[14] RXDIE	**RXDI 中断使能（XIF Rx Done Interrupt Enable）** 0：不使能 1：使能
[13] RXOFIE	**RXOFI 中断使能（RXFIFO Overflow Interrupt Enable）** 0：不使能 1：使能
[12] RXFIE	**RXFI 中断使能（RXFIFO Full Interrupt Enable）** 0：不使能 1：使能
[10:8] RXFTLR	**RXFIFO 满中断阈值（RXFIFO Full Watermark）** 4'b0000：FIFO 中有 1 个或 1 个以上数据就会触发 RXFIFO 满中断 4'b0001：FIFO 中有 2 个或 2 个以上数据就会触发 RXFIFO 满中断 4'b0010：FIFO 中有 3 个或 3 个以上数据就会触发 RXFIFO 满中断 4'b0011：FIFO 中有 4 个或 4 个以上数据就会触发 RXFIFO 满中断 4'b0100：FIFO 中有 5 个或 5 个以上数据就会触发 RXFIFO 满中断 4'b0101：FIFO 中有 6 个或 6 个以上数据就会触发 RXFIFO 满中断 4'b0110：FIFO 中有 7 个或 7 个以上数据就会触发 RXFIFO 满中断 4'b0111：FIFO 中有 8 个数据就会触发 RXFIFO 满中断
[7:0] DCNT	**接收的数据量（The Count of DATA）** 在接收了 DCNT 个数据之后将结束一帧的数据传输

23.5.2.3 XIF 时序寄存器（XIF_TIMING）

- 名称：XIF Timing Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDHTIME								RDLTIME							
R/W								R/W							
0x00								0x00							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONLTIME								RSTTIME							
R/W								R/W							
0x00								0x00							

字段	说明
[31:24] RDHTIME	**XIF RD 为高的时间长度（The Length Of Time For RD High）** 单位为系统时钟周期 Note：实际值为配置值+1
[23:16] RDLTIME	**XIF RD 为低的时间长度（The Length Of Time For RD Low）** 单位为系统时钟周期 Note：实际值为配置值+1
[15:8] CONLTIME	**XIF CONVST 为低的时间长度（The Length Of Time For CONVST Low）** 单位为系统时钟周期 Note：实际值为配置值+1
[7:0] RSTTIME	**XIF RESET 为高的时间长度（The Length Of Time For RESET High）** 单位为系统时钟周期 Note：实际值为配置值+1，若配置为 0，则在启动之后不会发送 RESET 信号

23.5.2.4 XIF 超时设置寄存器（XIF_TO）

- 名称：XIF Timeout Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTOT															
R/W															
0x0															

字段	说明
[15:0] PTOT	**XIF 超时设置（XIF BUSY TIMEOUT Timing）** 指的是在设置完 convst 之后等待 busy 的最大时间，单位是系统时钟周期 Note：实际值为配置值+1

23.5.2.5 XIF 读取数据寄存器（XIF_DATA）

- 名称：XIF FIFO Data Registers
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
R															
0x0															

字段	说明
[15:0] DATA	**读取数据寄存器（FIFO Read DATA Register）** 读取接收 FIFO 缓冲区数据，读取的数据会自动右对齐。

23.5.2.6 XIF 中断状态寄存器（XIF_ISR）

- 名称：XIF Interrupt And Status Register
- 偏移地址：0x14
- 默认值：0x00000010
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RXFLR						FF	FE	BTOI	RXDI	RXOFI	RXFI
				R						R	R	R/W	R/W	R/W	R
				0x00						0x00	0x1	0x0	0x0	0x0	0x0

字段	说明
[11:8] RXFLR	**RXFIFO 实时剩余数据量（RXFIFO Level Register）** RXFIFO 中剩余的数据
[5] FF	**RXFIFO 满标志（RXFIFO Full Flag）** 0：RXFIFO 未满 1：RXFIFO 已满 Note：当 RX FIFO 不再满时，该位被清除。
[4] FE	**RXFIFO 空标志（RXFIFO Empty Flag）** 0：RXFIFO 不为空 1：RXFIFO 为空 Note：当 RX FIFO 不为空时，该位被清除。
[3] BTOI	**XIF 等待 BUSY 超时中断（XIF Wait BUSY Timeout Interrupt Enable）** 超时时会将 XIF 关闭 0：无超时 1：超时 Note：对该位写 1 将清除 BUSYTOI 中断
[2] RXDI	**XIF 接收完成中断（XIF RX Done Interrupt）** XIF 接收完 CONHTIME 个数据时起中断，若 RELOAD 为 1，该中断不会起来 1：接收完毕 0：未接收完毕

	Note：对该位写 1 将清除 MSTDONEI 中断
[1] RXOFI	**RXFIFO 溢出中断（RXFIFO Overflow Interrupt）** 当 RXFIFO 已满并且新数据被接收写入 FIFO 中时，会起溢出错误中断。此时 FIFO 中的数据将保留，而接收的新数据将丢失。 0：无溢出错误 1：溢出错误 Note：对该位写 1 将清除 RXOFI 中断。
[0] RXFI	**RXFIFO 满中断（RXFIFO Full Interrupt）** 与 RXFTLR 寄存器有关，当 RXFIFO 中的数据量大于或等于预设值（RXFTLR）时置位。 1：RXFIFO 数据量大于等于预设值 0：RXFIFO 数据量小于预设值 Note：当 RXFIFO 数据量小于预设值时 RXFIFOFI 中断自动清 0

24 脉冲密度调制（PDM）

24.1 简介

本章介绍 sigma delta 滤波器模块（PDM）。PDM 是一个四通道数字滤波器，专门用于电机控制应用中的电流测量和旋转变压器位置解码。每个输入通道可以接收一个独立 delta-sigma（ΔΣ）调制器比特流。比特流由两个可单独编程的数字抽取滤波器处理。该滤波器组包括一个快速比较器（辅助滤波器），用于立即数字阈值比较，以进行过流和欠流监视以及过零检测。

24.2 主要特性

PDM 的主要功能如下：

- 支持 SPI 主从模式操作
- 支持 DMA 模式
- 支持 32k 种不同的波特率配置
- 支持四通道 PDM
- 每个 PDM 通道都有一个可配置的辅助滤波器（比较器）单元：
  - 提供四种不同的滤波器类型选择（$Sinc^1$ / $Sinc^2$ / Sincfast / $Sinc^3$/ $Sinc^4$）选项
  - 能够检测超值情况和欠值情况
  - 比较器滤波器单元（COSR）的 OSR 值，可在 1 到 32 之间编程
- 每个 PDM 通道都有一个可配置主滤波器（数据滤波器）单元：
  - 提供四种不同的滤波器类型选择（$Sinc^1$ / $Sinc^2$ / Sincfast / $Sinc^3$/ $Sinc^4$）选项
  - 数据过滤器单元（DOSR）的 OSR 值，可在 1 到 256 之间进行编程
  - 能够启用或禁用（或同时禁用）单个过滤器模块
- 主滤波器可以有 24 位和 16 位输出选择
- 可以将 TMRx/PWMx.SOCA / SOCB 配置为按每个数据过滤器通道用作 SDSYNC 源。
- 支持 16bit 数据配置

24.3 结构框图

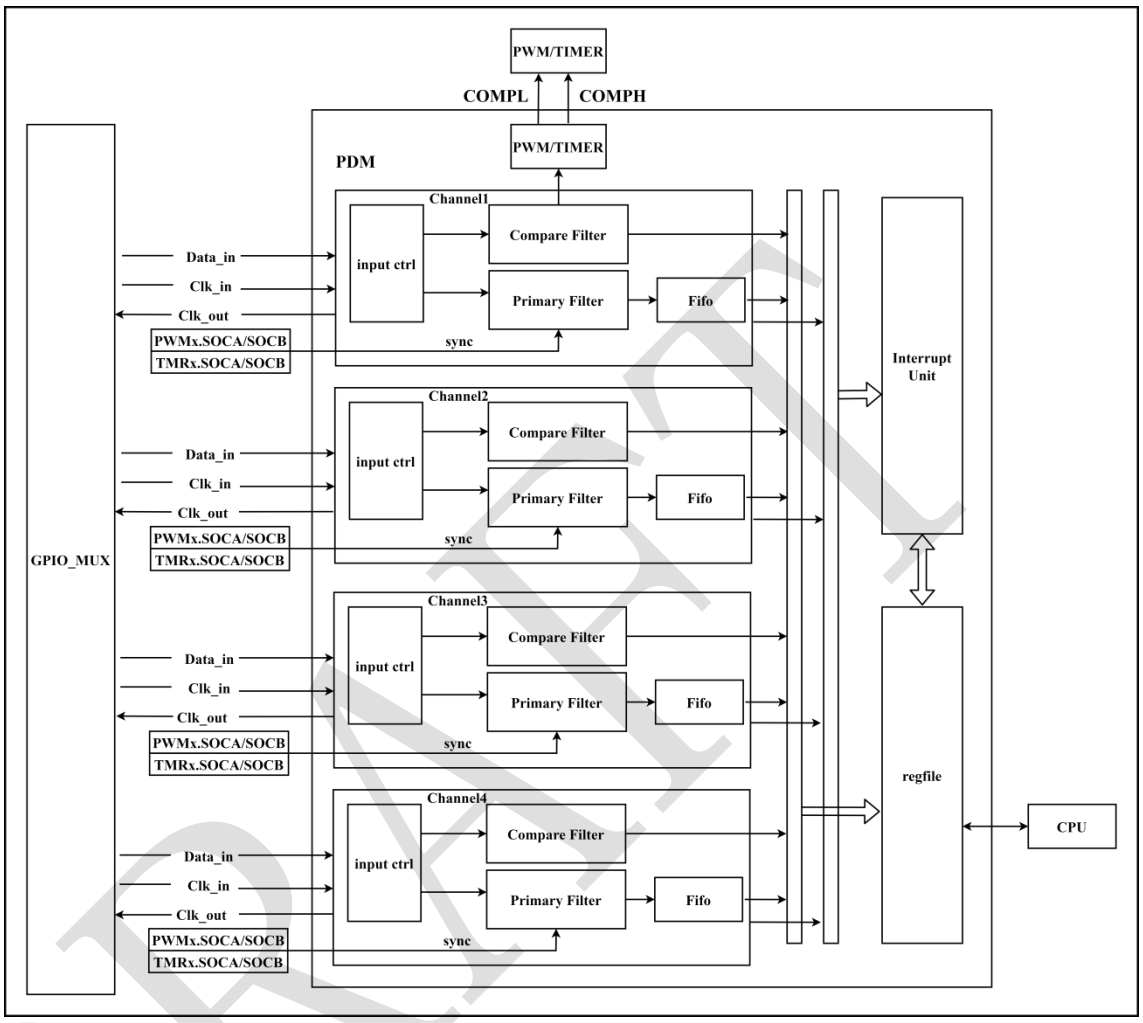


图 22-1 PDM 结构框图

24.4 功能描述

24.4.1 SPI 控制单元

PDM 和外界ΣΔ调制器的数据交互是通过 SPI 协议来实现的。端口信号如下表所示：

表 24-1 PDM 数据交互端口信息

名称	信号类型	描述
CLKOUT[1:0]	SPI 时钟输出	SPI 时钟输出，用于为外部ΣΔ调制器提供时钟信号
CLKIN[1:0]	SPI 时钟输入	SPI 时钟输入，从外部ΣΔ调制器获取的时钟信号
DATAIN[1:0]	SPI 数据输入	SPI 数据输入，从外部ΣΔ调制器获取的数据信号

PDM 可作为主机，也可作为从机，取决于 PDM_CTRL.MSTEN 寄存器位，向该位写 1 代表 PDM 作为主机，否则作为从机。

作为主机时，PDM 可以通过配置 PDM_CTRL.BAUD 寄存器来配置输入时钟的波特率，波特率公式如下所示：

$$CLKOUT = \frac{F_{SYSCLK}}{BAUD \times 2}$$

由公式可知，时钟输出的 CLKOUT 与系统总线时钟的频率有关。同时，可以配置 PDM_CTRL.SCPOL 设置输出 CLKOUT 的极性。

作为从机时，通过配置 PDM_CTRL.SAMPMODE 来决定当前 PDM 是通过上升沿采样还是下降沿采样。

因为输入时钟的不确定性，若是丢失沿的话很可能导致后续数据的失效，因此可以通过配置 PDM_CLKTO 寄存器去限制输入时钟 CLKIN 的最大翻转时长，以系统时钟为计数，如果翻转不及时，则会产生中断。

24.4.2 SINC 滤波器

PDM 包含 $Sinc^x$ 类型数字滤波器实现。此 $Sinc^x$ 滤波器执行输入数字数据流滤波，这会导致减小输出数据速率（抽取）以及增大输出数据分辨率。可对 $Sinc^x$ 数字滤波器进行配置，以达到所需输出数据速率和所需输出数据分辨率。可配置参数有：

- 滤波器阶数/类型：
  - FastSinc
  - $Sinc^1$
  - $Sinc^2$
  - $Sinc^3$
  - $Sinc^4$
- 滤波器过采样/抽取率：最大取值 256

滤波器支持以下传递函数（H 域中的冲激响应）：

- Sinc^x 滤波器类型：$H(z) = (\frac{1-z^{-OSR}}{1-z^{-1}})^x$
- FastSinc 滤波器类型： $H(z) = (\frac{1-z^{-OSR}}{1-z^{-1}})^2 \times (1 + z^{-2\times OSR})$

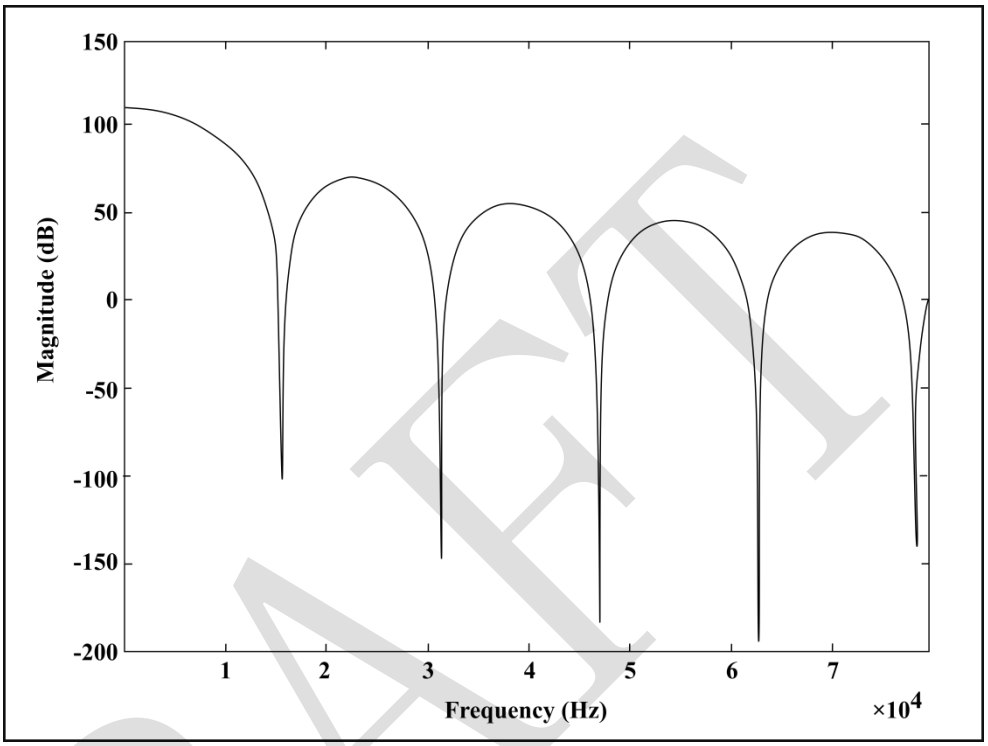


图 24-2　Sinc^3 的滤波器响应

表 24-2 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 OSR 值

OSR	Sinc^1	Sinc^2	Sinc^3	Sinc^4	FastSinc
x	+/- x	+/- x^2	+/- x^3	+/- x^4	+/- 2x^2
4	+/- 4	+/- 16	+/- 64	+/- 256	+/- 32
8	+/- 8	+/- 64	+/- 512	+/- 4096	+/- 128
32	+/- 32	+/- 1024	+/- 32768	+/- 1048576	+/- 2048
64	+/- 64	+/- 4096	+/- 262144	+/- 16777216	+/- 8192
128	+/- 128	+/- 16384	+/- 2097152	+/- 268435456	+/- 32768
256	+/- 256	+/- 65536	+/- 16777216	+/- 不支持	+/- 131072

24.4.3　主滤波器

数据过滤器是可配置的 Sinc 过滤器，它支持以下过滤器类型：Sinc^1，Sinc^2，Sinc^3，Sinc^4 和 SincFast。数据过滤器 OSR（DOSR）设置可以配置为 1 到 256，并且与比较器过滤器无关。数据滤波器

的有效分辨率取决于数据滤波器类型，DOSR 和 sigma-delta 调制器频率。默认情况下，数据过滤器是不启动的。数据过滤器以 24 位或者 16 位有符号整数格式输出。该滤波器单元将低输入信号转换为“-1”，将高输入信号转换为“1”。结果计算为数据滤波器的输出给出了正值和负值。

24.4.3.1 24bit 或者 16bit 输出

数据过滤器输出可以 24 位或 16 位格式表示。

24 位数据过滤器表示形式：

- PDM_DFCR[13]的 DFDR=1 时，数据过滤器输出以 24 位格式表示。在这种配置下，写入 PDM_DFCR[19:16]的 SHIFT 位对数据滤波器的输出没有任何影响。

16 位数据过滤器表示形式：

- 默认情况下，数据过滤器输出以 16 位格式表示
- PDM_DFCR[13]的 DFDR=0 时，数据过滤器输出以 16 位格式表示。但是，用户需要配置相应的 PDM_DFCR[19:16]的 SHIFT 位，以控制将 24 位字的哪个 16 位部分作为输出。

例如，对于以下数据过滤器配置：

- 过滤器类型= $Sinc^3$
- OSR = 128
- PDM_DFCR[13]的 DFDR=0

上述配置对于带符号输出值的数据过滤器的范围可以在-2097152 到 2097152 之间。但是，16 位带符号输出仅支持从-32768 到 32767 的值。因此，需要将移位控制位（PDM_DFCR[19:16]的 SHIFT）配置为 7，以正确表示 16 位格式的数据过滤器输出。下表显示了不同 OSR 和滤波器类型的移位控制位的最小推荐配置设置。

表 24-3 SHIFT 位的最小推荐配置

OSR	$Sinc^1$	$Sinc^2$	FastSinc	$Sinc^3$	$Sinc^4$
1~13	0	0	0	0	0
14~15	0	0	0	0	1
16~19	0	0	0	0	2
20~21	0	0	0	0	3
22~25	0	0	0	0	4
26~31	0	0	0	0	5
32~38	0	0	0	1	6
39~40	0	0	0	1	7
41~45	0	0	0	2	7
46~50	0	0	0	2	8
51~53	0	0	0	3	8
54~63	0	0	0	3	9
64~76	0	0	0	4	10
77~80	0	0	0	4	不支持
81~101	0	0	0	5	不支持

102~127	0	0	0	6	不支持
128~161	0	0	1	7	不支持
162~181	0	0	1	8	不支持
182~203	0	1	2	8	不支持
204~255	0	1	2	9	不支持
256	0	2	3	10	不支持

若 shift 配置得比表格中的数值小，可能会导致 16 位输出数据有误。

24.4.3.2 FIFO

每个主（数据）过滤器通道都有一个深度为 8 的 24 位 FIFO。

等待同步功能

FIFO 等待同步功能可用于忽略数据过滤器中的数据写入事件，直到触发 SDSYNC（来自 PWM 或者 TIMER）事件为止。

当 PDM_SDSYNC[6]的 WTSYNCEN=0 时，数据滤波器中的数据将会一直写入 FIFO 中，直到 FIFO 被填满。

当 PDM_SDSYNC[6]的 WTSYNCEN=1 时，那么只有在接收到 SDSYNC 事件后，FIFO 才会接收数据滤波器的数据写入事件，直到 FIFO 被填满或者 FIFINTR 中断起来。

当 PDM_SDSYNC[9]的 FFSYNCCLREN=1 时，接收到 SDSYNC 事件后会把 FIFO 中的数据清除。

24.4.3.3 SDSYNC 事件

主（数据）滤波器可以相对于 PWM 或者 TIMER 事件（称为 SDSYNC 事件）进行同步。来自 PWM 或者 TIMER 模块的 SDSYNC 信号会复位 DOSR 计数器。默认情况下，此功能是禁用的，可以通过设置 PDM_DFCR [14]的 SDSYNCEN=1 来启用。每个主滤波器都可以与任何可用的 PWMx 或者 TIMERx 同步。PDM_SDSYNC[5:0]的 SDSYNCSEL 位允许用户配置哪个 PWM 或者 TIMER 信号向主滤波器提供 SDSYNC 脉冲，如下表所示：

表 24-4 SDSYNCSEL 配置对应的 PWM 的信号表

	SDSYNCSEL[1:0]			
SDSYNCSEL[5:2]	00	01	10	11
0	HRPWM_ADC_TRG0	HRPWM_ADC_TRG1	TMR7_TRGO	TMR8_TRGO
1	HRPWM_ADC_TRG2	HRPWM_ADC_TRG3	TMR0_CC0	TMR0_TRGO
2	HRPWM_ADC_TRG4	HRPWM_DAC_TRG5	TMR1_CC0	TMR1_TRGO
3	HRPWM_ADC_TRG6	HRPWM_ADC_TRG7	TMR2_CC0	TMR2_TRGO
4	HRPWM_ADC_TRG8	HRPWM_ADC_TRG9	TMR3_CC0	TMR3_TRGO
5	HRPWM_DAC_TRG0	HRPWM_DAC_TRG0	TMR4_CC0	TMR4_TRGO
6	HRPWM_DAC_TRG1	HRPWM_DAC_TRG1	TMR9_CC0	TMR9_TRGO
7	HRPWM_DAC_TRG2	HRPWM_DAC_TRG2	TMR10_CC0	TMR10_TRGO

8~15	Reserved	Reserved	Reserved	Reserved

由于 Sinc 滤波器的固有体系结构（Sinc[1]，Sinc[2]，Sinc[3]，Sinc[4]，SincFast），根据滤波器类型的不同，前几个样本将是错误的。表 22-5 显示在以下情况下不正确的样本数：

1. 首次启用/配置 Sinc 过滤器时。
2. 在操作过程中禁用/重新启用或重新配置了 Sinc 过滤器时。
3. 数据过滤器从 PWM 或者 TIMER 接收到 SDSYNC 事件时。

表 24-5 启动并配置滤波器后不正确的样本数

滤波器类型	启用并配置过滤器后不正确的样本数
Sinc[1]	没有错误的样本
Sinc[2]	第一个样本是错误的
Sinc[3]	前两个样本是错误的
Sinc[4]	前两个样本是错误的
FastSinc	前两个样本是错误的

24.4.4 比较滤波器

大多数控制系统要求在电流或电压超出范围时通过使 PWM 跳闸来保护系统。次级（比较器）滤波器的主要目的是允许用户以快速的建立时间监视输入条件。这使用户可以使 PWM 跳闸，以保护系统免受潜在损害。

比较器滤波器是可配置的 Sinc 滤波器，它支持以下滤波器类型：Sinc[1]，Sinc[2]，Sinc[3] 和 SincFast。比较器 OSR（COSR）设置可以配置为 1 到 32，并且与数据过滤器无关。比较器滤波器的有效分辨率取决于比较器滤波器的类型，COSR 和 sigma-delta 调制器频率。默认情况下，比较器滤波器被禁用。比较器滤波器的输出以 16 位无符号格式表示。该滤波器单元将低输入信号转换为“0”，将高输入信号转换为“1”。计算结果仅给出比较器滤波器输出的正值。

表 24-6. 比较器滤波器可以使用不同的 OSR 存储的不同的满量程值

OSR	Sinc[1]	Sinc[2]	Sinc[3]	Sinc[4]	FastSinc
x	0~x	0~x^2	0~x^3	0~x^4	0~$2x^2$
4	0~4	0~16	0~64	0~256	0~32
8	0~8	0~64	0~512	0~4096	0~128
32	0~32	0~1024	0~32768	0~1048576	0~2048

24.4.4.1 阈值比较器

通过 PDM_CMPH 寄存器配置高阈值比较器，比较器可用于检测数据高出阈值的情形。当比较滤波器数据≥高阈值寄存器（PDM_CMPH.HLT）时，将产生高阈值事件。高阈值事件可以产生 CPU 中断，也可以生成输出信号 PDMx_CMPH，该信号给到 PWM 以及 Timer，可以作为关断 PWM 和 Timer 的事件。

通过 PDM_CMPL 寄存器配置低阈值比较器，比较器可用于检测数据低于阈值的情形。当比较器数据≤下阈值寄存器（PDM_CMP.LLT）时，将产生一个低阈值事件。低阈值事件可以产生

CPU 中断，也可以生成输出信号 PDMx_CMPL，该信号给到 PWM 以及 Timer，可以作为关断 PWM 和 Timer 的事件。

PDMx_CMPH 与 PDMx_CMPL 信号反映了高阈值比较器/低阈值比较器的比较结果，PDMx_CMPH 为 1 表示高阈值事件，PDMx_CMPL 为 1 表示低阈值事件。PDMx_CMPH 可以配置为高阈值+低阈值事件，通过 PDM_CFCRx 寄存器的 COMPSEL 选择。

24.4.5 中断单元

比较器低阈值中断

比较滤波器的计算结果小于等于预设值 LLT 时触发比较器低阈值中断。四个比较器滤波模块中的任何一个比较器低阈值中断都可以触发 CPU 中断。

比较器高阈值中断

比较滤波器的计算结果大于等于预设值 HLT 时触发比较器高阈值中断。四个比较器滤波模块中的任何一个比较器高阈值中断都可以触发 CPU 中断。

FIFO 满中断

FIFO 中的数据大于预设值（PDMFLT）时触发 FIFO 满中断，该中断在 FIFO 的数据小于等于预设值时自动清除。四个滤波模块中的任何一个 FIFO 满中断都可以触发 CPU 中断。

FIFO 上溢中断

如果 FIFO 已满（FIFO 深度为 8），且当前 FIFO 中还有一个数据写入，此时触发 FIFO 上溢中断。四个比较器滤波模块中的任何一个 FIFO 上溢中断都可以触发 CPU 中断。

FIFO 下溢中断

如果 FIFO 已空，且当前 FIFO 中还有一个数据读取，此时触发 FIFO 下溢中断。四个比较器滤波模块中的任何一个 FIFO 下溢中断都可以触发 CPU 中断。

数据完成中断

比较滤波器计算完成并将数据写入到 PDM_CDAT 寄存器中，此时触发数据完成中断。四个比较器滤波模块中的任何一个数据完成中断都可以触发 CPU 中断。

数据溢出中断

数据完成中断已经触发，软件没读取 PDM_CDAT，也没有对 DFINTR 位写 1 去清除 DFINTR，此时比较滤波器完成了一个新的计算结果并覆盖 PDM_CDAT 寄存器时，会触发数据溢出中断。四个比较器滤波模块中的任何一个数据溢出中断都可以触发 CPU 中断。

时钟超时中断

PDM 作为从机时，输入使能翻转的时长超过了 CLKTO 寄存器的值时会触发时钟超时中断。

24.5 寄存器定义

24.5.1 寄存器列表

PDM 基地址：

PDM0(0x40017000) PDM1(0x40017100) PDM2(0x40017200) PDM3(0x40017300)

偏移	实例地址	名称	默认值	描述
0x00	PDM 基地址+0x00	PDMx_ENABLE	0x00000000	PDM 使能寄存器
0x04	PDM 基地址+0x04	PDMx_CTRL	0x00000000	PDM 控制寄存器
0x08	PDM 基地址+0x08	PDMx_DFCR	0x00000000	PDM 主滤波器控制寄存器
0x0C	PDM 基地址+0x0C	PDMx_CFCR	0x00000000	PDM 比较滤波器控制寄存器
0x10	PDM 基地址+0x10	PDMx_FCSR	0x00010000	PDM FIFO 控制寄存器
0x14	PDM 基地址+0x14	PDMx_IER	0x00000000	PDM 中断使能寄存器
0x18	PDM 基地址+0x18	PDMx_ISR	0x00000000	PDM 中断状态寄存器
0x1C	PDM 基地址+0x1C	PDMx_DDAT	0x00000000	PDM 主滤波器数据寄存器
0x20	PDM 基地址+0x20	PDMx_CDAT	0x00000000	PDM 比较滤波器数据寄存器
0x24	PDM 基地址+0x24	PDMx_FDAT	0x00000000	PDM FIFO 数据寄存器
0x28	PDM 基地址+0x28	PDMx_CMPH	0x0000FFFF	PDM 高阈值比较寄存器
0x2C	PDM 基地址+0x2C	PDMx_CMPL	0x00000000	PDM 低阈值比较寄存器
0x30	PDM 基地址+0x30	PDMx_CLKTO	0x0003FFFF	PDM 时钟翻转超时寄存器
0x34	PDM 基地址+0x34	PDMx_SDSYNC	0x00000400	PDM 同步控制寄存器

24.5.2 寄存器描述

24.5.2.1 PDM 使能寄存器（PDMx_ENABLE）

- 名称：**PDM Enable Register**
- 偏移地址：**0x00**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															PDMEN
															R/W
															0x0

字段	说明
[0] PDMEN	**PDM 使能位（PDM Enable）** 1：PDM 使能 0：PDM 不使能

24.5.2.2 PDM 控制寄存器（PDMx_CTRL）

- 名称：**PDM Control Register**
- 偏移地址：**0x04**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDLY											DMAEN	SAMPMODE	SCPOL	MSTEN
	R/W											R/W	R/W	R/W	R/W
	0x0											0x0	0x0	0x0	0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				BAUD											
				R/W											
				0x0											

字段	说明
[30:28] RXDLY	**SPI 接收采样延迟（SPI RX Delay）** 作为主机时，这个寄存器代表需要延后多少个系统周期进行采样
[19] DMAEN	**DMA 使能位（DMA Enable）** 1：使能 0：不使能
[18] SAMPMODE	**从机时采样模式选择（Slave Sample Mode）** 1：时钟上升沿采样 0：时钟下降沿采样
[17] SCPOL	**串行时钟极性（Serial Clock Polarity）** 0：串行时钟的无效状态为低 1：串行时钟的无效状态为高
[16]	**PDM 主机使能位（PDM Master Enable）**

MSTEN	0：PDM 作为从机 1：PDM 作为主机
[11:0] BAUD	**波特率配置（Baud Rate Configuration）** 为 0 时不向外发送时钟；输出时钟的频率如下 Foutclk = Fpclk / (baudr*2)

24.5.2.3 PDM 主滤波器控制寄存器（PDMx_DFCR）

- **名称：PDM Main Filter Control Register**
- **偏移地址：0x08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												SHIFT			
												R/W			
												0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSYNCEN	DFDR	DFBYPASS	DFSST			DFEN	DOSR							
	R/W	R/W	R/W	R/W			R/W	R/W							
	0x0	0x0	0x0	0x0			0x0	0x0							

字段	说明
[19:16] SHIFT	**移位控制选择位（Shift Control Select）** 最大值为 13，超过 13 的值均以 13 为准。只在 DFDR=0 时有效。 0：取计算结果的[15:0]作为输出。 1：取计算结果的[16:1]作为输出。 2：取计算结果的[17:2]作为输出。 …… 13：取计算结果的[28:13]作为输出。 14~15：Reserved
[14] SDSYNCEN	**SDSYNC 信号使能位（SDSYNC Enable）** PDM 是否接收 PWM 的 SDSYNC 作为同步信号。 1：接收 0：不接收
[13] DFDR	**主滤波器输出数据位宽选择（Data Filter Output Data Length）** 1：24bit 输出 0：16bit 输出
[12] DFBYPASS	**主滤波器 bypass 使能（Data Filter Bypass Enable）** 使能时，滤波器输出会直接等于滤波器输入。 1：bypass 使能 0：bypass 不使能
[11:9] DFSST	**主滤波器类型（Data Filter Structure）** 000：FastSinc 001：Sinc1 010：Sinc2 011：Sinc3 100：Sinc4
[8] DFEN	**主滤波器使能位（Data Filter Enable）** 1：使能 0：不使能
[7:0] DOSR	**主滤波器过采样值（Data filter Oversampling ratio）** 配置的真实值会比写入的值+1。例如需要跑过采样 256，那么需要向该寄存器写入 255。 注：DOSR=0,PDM 不工作

24.5.2.4 PDM 比较滤波器控制寄存器（PDMx_CFCR）

- 名称：PDM Compare Filter Control Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					COMPSEL	CFBYPASS	CFSST			CFEN	COSR				
					R/W	R/W	R/W			R/W	R/W				
					0x0	0x0	0x0			0x0	0x0				

字段	说明
[10] COMPSEL	**比较滤波器输出选择（Comparator Filter Output Select）** 比较滤波器 COMPL 和 COMPH 输出选择，COMPL 指的是比较滤波器输出低于预设值(PDMx_CMPL)，COMPH 指的是比较滤波器输出高于预设值(PDMx_CMPH) 1：COMPL 和 COMPH 或起来输出 0：COMPL 和 COMPH 分别输出
[9] CFBYPASS	**比较滤波器 bypass 使能位（Comparator Filter Bypass Enable）** 使能时，滤波器输出会直接等于滤波器输入。 1：bypass 使能 0：bypass 不使能
[8:6] CFSST	**比较滤波器类型（Comparator Filter Structure）** 000：FastSinc 001：Sinc1 010：Sinc2 011：Sinc3 100：Sinc4
[5] CFEN	**比较滤波器使能位（Comparator Filter Enable）** 1：使能 0：不使能
[4:0] COSR	**比较滤波器过采样值（Comparator filter Oversampling ratio）** 配置的真实值会比写入的值+1。例如需要跑过采样 32，那么需要向该寄存器写入 31。 注：COSR=0,PDM 不工作

24.5.2.5 PDM FIFO 控制寄存器（PDMx_FCSR）

- 名称：**PDM FIFO Control Register**
- 偏移地址：**0x10**
- 默认值：**0x00010000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													FIFORST	FIFOFULL	FIFOEMPTY
													W	R	R
													0x0	0x0	0x1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				PDMFST									PDMFIL		
				R									R/W		
				0x0									0x0		

字段	说明
[18] FIFORST	**FIFO 复位（FIFO Reset）** 写 1 复位，自动清 0。 0：无作用 1：FIFO 复位
[17] FIFOFULL	**FIFO 满标志（FIFO Full Flag）** 0：FIFO 未满 1：FIFO 满
[16] FIFOEMPTY	**FIFO 空标志（FIFO Empty Flag）** 1：FIFO 空 0：FIFO 非空
[11:8] PDMFST	**FIFO 状态（FIFO Status）** 0000：FIFO 中没有数据 0001：FIFO 中有 1 个数据 0010：FIFO 中有 2 个数据 0011：FIFO 中有 3 个数据 …… 1000：FIFO 中有 8 个数据
[2:0] PDMFIL	**FIFO 满中断阈值选择。** 000：FIFO 中有 1 个或 1 个以上数据时触发 FIFO 满中断 001：FIFO 中有 2 个或 2 个以上数据时触发 FIFO 满中断 010：FIFO 中有 3 个或 3 个以上数据时触发 FIFO 满中断 011：FIFO 中有 4 个或 4 个以上数据时触发 FIFO 满中断 …… 111：FIFO 中有 8 个数据时触发 FIFO 满中断

24.5.2.6 PDM 中断使能寄存器（PDMx_IER）

- 名称：PDM Interrupt Enable Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FIUIE	CTIE	FFIE	FIOIE	DOFIE	DFIE	LLIE	HLIE
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] FIUIE	**FIUINTR 中断使能（FIUINTR Interrupt Enable）** 1：使能 0：不使能
[6] CTIE	**CTOINTR 中断使能（CTOINTR Interrupt Enable）** 1：使能 0：不使能
[5] FFIE	**FIFINTR 中断使能（FIFINTR Interrupt Enable）** 1：使能 0：不使能
[4] FIOIE	**FIOINTR 中断使能（FIOINTR Interrupt Enable）** 1：使能 0：不使能
[3] DOFIE	**DOINTR 中断使能（DOINTR Interrupt Enable）** 1：使能 0：不使能
[2] DFIE	**DFINTR 中断使能（DFINTR Interrupt Enable）** 1：使能 0：不使能
[1] LLIE	**LLINTR 中断使能（LLINTR Interrupt Enable）** 1：使能 0：不使能
[0] HLIE	**HLINTR 中断使能（HLINTR Interrupt Enable）** 1：使能 0：不使能

24.5.2.7 PDM 中断状态寄存器（PDMx_ISR）

- 名称：PDM Interrupt State Register
- 偏移地址：0x18
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FIUINTR	CTOINTR	FIFINTR	FIOINTR	DOINTR	DFINTR	LLINTR	HLINTR
								R/W1C	R/W1C	R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] FIUINTR	**FIFO 下溢中断（FIFO Underflow Interrupt）** FIFO 已空，且当前 FIFO 中还有一个数据读取操作，造成 FIFO 下溢并触发中断。 1：起中断 0：无效果 Note：对该位写 1 会清除 FIUINTR 中断
[6] CTOINTR	**时钟超时中断（Clock Timeout Interrupt）** PDM 作为从机时，输入使能翻转的时长超过了 CLKTO 寄存器的值时会触发中断。 1：起中断。 0：无效果 Note：对该位写 1 会清除 CTOINTR 中断
[5] FIFINTR	**FIFO 满中断（FIFO Full Interrupt）** FIFO 中的数据大于预设值（PDMFLT）时会触发中断。 1：起中断 0：无效果 Note：FIFO 中的数据小于等于预设值（PDMFLT）时会自动清除 FIFINTR 中断
[4] FIOINTR	**FIFO 上溢中断（FIFO Overflow Interrupt）** FIFO 已满，且当前 FIFO 中还有一个数据写入操作，造成 FIFO 上溢并触发中断。 1：起中断 0：无效果 Note：对该位写 1 会清除 FIOINTR 中断
[3] DOINTR	**数据溢出中断（Data Overflow Interrupt）** DFINTR 中断已经触发，软件没读取 PDMx_CDAT 也没有对 DFINTR 位写 1 去清除 DFINTR，此时比较滤波器完成了一个新的计算结果并覆盖 PDMx_CDAT 寄存器时，会触发 DOINTR 中断。 1：起中断 0：无效果 Note：对该位写 1 会清除 DOINTR 中断
[2] DFINTR	**数据完成中断（Data Finish Interrupt）** 比较滤波器完成计算并将数据写入到 PDMx_CDAT 寄存器中，此时触发中断代表 PDMx_CDAT 有效可读。 1：起中断 0：无效果 Note：读取 PDMx_CDAT 寄存器或者对该位写 1 都可以清除 DFINTR 中断。
[1] LLINTR	**低阈值中断（Low Level Interrupt）** 比较滤波器的计算结果小于等于预设值（LLT）时触发中断。 1：起中断 0：无效果 Note：对该位写 1 会清除 LLINTR 中断
[0] HLINTR	**高阈值中断（High Level Interrupt）** 比较滤波器的计算结果大于等于预设值（HLT）时触发中断。 1：起中断 0：无效果

Note：对该位写 1 会清除 HLINTR 中断

24.5.2.8 PDM 主滤波器数据寄存器（PDMx_DDAT）

- 名称：**PDM Main Filter Result Register**
- 偏移地址：**0x1C**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								DDAT							
								R							
								0x0							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAT															
R															
0x0															

字段	说明
[23:0] DDAT	**主滤波器数据寄存器（Data Filter Data Register）** 主滤波器计算结果

24.5.2.9 PDM 比较滤波器数据寄存器（PDMx_CDAT）

- 名称：**PDM Compare Filter Result Register**
- 偏移地址：**0x20**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDAT															
R															
0x0															

字段	说明
[15:0] CDAT	**比较滤波器数据寄存器（Comparator Filter Data Register）** 比较滤波器计算结果

24.5.2.10 PDM FIFO 数据寄存器（PDMx_FDAT）

- 名称：**PDM FIFO Result Register**
- 偏移地址：**0x24**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								FDAT							
								R							
								0x0							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDAT															
R															
0x0															

字段	说明
[23:0] FDAT	**FIFO 数据寄存器（FIFO Data Register）** 读取 FIFO 中接收到的主滤波器计算结果

24.5.2.11 PDM 高阈值比较寄存器（PDMx_CMPH）

- 名称：**PDM High-Level Compare Register**
- 偏移地址：**0x28**
- 默认值：**0x0000FFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLT															
R/W															
0xffff															

字段	说明
[15:0] HLT	**比较滤波器的高阈值（Comparator High-Level Data）**

24.5.2.12 PDM 低阈值比较寄存器（PDMx_CMPL）

- 名称：PDM Low-Level Compare Register
- 偏移地址：0x2C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLT															
R/W															
0x0															

字段	说明
[15:0] LLT	**比较滤波器的低阈值（Comparator Low-Level Data）**

24.5.2.13 PDM 时钟翻转超时寄存器（PDMx_CLKTO）

- 名称：PDM Clock Timeout Register
- 偏移地址：0x30
- 默认值：0x0003FFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														CLKTOT	
														R/W	
														0x3ffff	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKTOT															
R/W															
0x3ffff															

字段	说明
[17:0] CLKTOT	**PDM 时钟超时寄存器（Clock Timeout Register）** PDM 输入时钟翻转超时阈值，以系统时钟为单位

24.5.2.14 PDM 同步控制寄存器（PDMx_SDSYNC）

- 名称：PDM Synchronization Control Register
- 偏移地址：0x34
- 默认值：0x00000400
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					WTSCLREN	FFSYNCCLREN	WTSYNCLR	WTSYNFLG	WTSYNCEN	SYNCSEL					
					R/W	R/W	W	R	R/W	R/W					
					0x1	0x0	0x0	0x0	0x0	0x0					

字段	说明
[10] WTSCLREN	**WTSYNFLG 清除使能位（WTSYNFLG Clear Enable）** 1：当 FIFO_FULL_INTR 中断触发或者对 WTSYNCLR 写 1 时 WTSYNFLG 会清 0 0：当对 WTSYNCLR 写 1 时 WTSYNFLG 会清 0
[9] FFSYNCCLREN	**SDSYNC 信号复位 FIFO 使能位（FIFO Reset By SDSYNC）** 1：SDSYNC 信号会触发 FIFO 复位 0：SDSYNC 信号不会触发 FIFO 复位
[8] WTSYNCLR	**WTSYNFLG 清除位（WTSYNFLG Clear Enable）** 1：清除 WTSYNFLG 0：无作用
[7] WTSYNFLG	**收到 SDSYNC 标志（Receive SDSYNC Flag）** PDM 收到 SDSYNC 信号时会使 WTSYNFLG 置 1。
[6] WTSYNCEN	**FIFO 在 WTSYNFLG 标志下的表现（The performance of FIFO under the WTSYNFLG）** 1：FIFO 只在 WTSYNFLG 为 1 时才能写入，在 WTSYNFLG 为 0 时不能写入 0：FIFO 在任何时候都可以写入
[5:0] SYNCSEL	**SDSYNC 选择（SDSYNC Select）** PWM 有 16 个信号会输出给 PDM，PDM 只选择其中一个信号作为 SDSYNC。

25 正交编码器接口（QEI）

25.1 简介

QEI 模块通常配合编码器来获取位置、方向及转速信息，主要包含如下几个单元：正交解码单元（QDU）；位置计数器及控制单元（PCCU）；边沿捕获单元（QCAP）；定时器基准单元（UTIME）。

25.2 主要特性

QEI 模块的输入信号主要有如下 3 路：

- QEPA/XCLK 与 QEPB/XDIR
  - 正交时钟模式
    在正交时钟模式下，QEI 提供两路互差 90°的脉冲信号 QEPA 及 QEPB，用两者之间的相位关系可判断旋转方向，脉冲信号的频率可判断转速。
  - 方向计数模式
    在方向计数模式下，方向以及脉冲信号分别由 XDIR 及 XCLK 单独提供。
- QEPI
  编码器通过索引脉冲信号 QEPI 来表明绝对起始地址，这路信号在每个旋转周期内用来复位芯片内部 QEI 模块的计数器。

25.3 结构框图

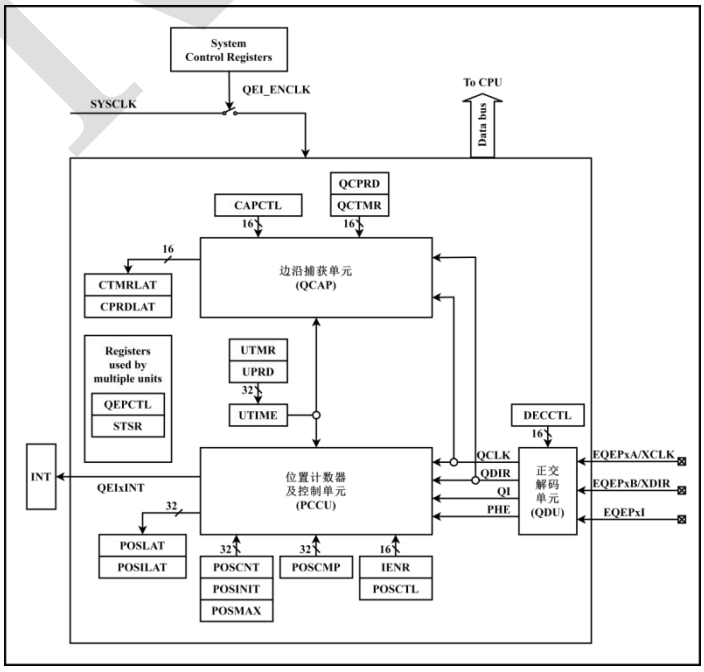


图 25-1 QEI 结构框图

25.4 功能描述

25.4.1.1 位置计数器的输入模式

位置计数器的输入模式由 QEI_DECCTL.QSRC 来决定，包括以下 4 种模式：正交计数模式、方向计数模式、增计数模式及减计数模式，以下将对这几种技术模式做详细的介绍。

正交计数模式

在正交计数模式下，方向判断逻辑电路通过判断 QEPA 及 QEPB 之间的相位关系来获得方向，并储存在 QEI_STSR.QDS 中。表 25-1 和图 25-2 分别以真值表和状态机方式来说明方向判断逻辑电路的工作过程。

表 25-1 方向判断逻辑电路真值表

上次边沿	当前边沿	QDIR	QPOSCNT
QA↑	QB↑	UP	增加
	QB↓	DOWN	减小
	QA↓	TOGGLE	增加或减小
QA↓	QB↓	UP	增加
	QB↑	DOWN	减小
	QA↑	TOGGLE	增加或减小
QB↑	QA↑	DOWN	减小
	QA↓	UP	增加
	QB↓	TOGGLE	增加或减小
QB↓	QA↓	DOWN	减小
	QA↑	UP	增加
	QB↑	TOGGLE	增加或减小

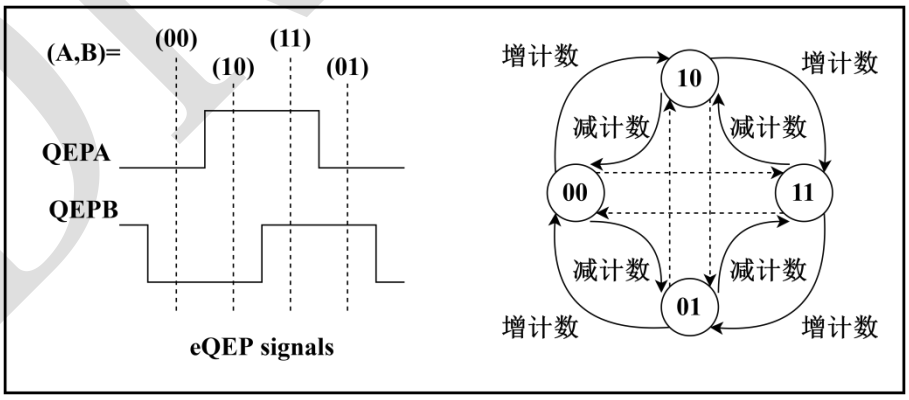


图 25-2 正交解码状态机

QEPA 及 QEPB 的上升沿和下降沿都将作为位置计数器的触发事件，因此 QEI 逻辑产生的计数脉冲的频率是每个输入脉冲的 4 倍，图 25-2 给出了 QEI 模块计数时钟及方向的解码逻辑。

正常情况下 QEPA 与 QEPB 之间的相位将相差 90°，当系统检测到两者之间的相位同步时，会将相位错误标志位 QEI_STSR.PHE 置位，同时会产生中断事件。

正常情况下 QEPA 连接到解码器的输入端 QA，QEPB 连接到解码器的输入端 QB，可通过设定 QEI_DECCTL.SWAP 位将两者交换，即 QEPA 接到 QB，而 QEPB 接到 QA。

方向计数模式

有些编码器直接提供方向信号以及计数时钟，在这种情况下可使用方向计数模式，QEPA 直接为位置计数器提供计数脉冲，QEPB 为位置计数器提供方向信息。当 QEPB 为高时，位置计数器在每个计数时钟的上升沿增加。当 QEPB 为低时，位置计数器在每个计数时钟的上升沿减小。

增计数模式

计数器的方向直接被硬件设定为增计数模式，位置计数器用来测量 QEPA 输入信号的频率。将 QEI_DECCTL.XCR 置位将使能 QEPA 输入的 2 个边沿都产生计数脉冲，从而将检测精度提高一倍。

减计数模式

计数器的方向直接被硬件设定为减计数模式，位置计数器用来测量 QEPA 输入信号的频率。将 QEI_DECCTL.XCR 置位将使能 QEPA 输入的 2 个边沿都产生计数脉冲，从而将检测精度提高一倍。

25.4.1.2 QEI 输入极性选择

QEI 模块的每路输入信号都可通过 QEI_DECCTL[3:0]位来控制输入极性，可以配置对应的位来对输入信号进行取反，例如，将 QEI_DECCTL.QIP 置位会将索引脉冲取反。

25.4.1.3 位置比较同步输出功能

QEI 模块包括一个位置比较单元，当位置计数寄存器 QEI_POSCNT 的值与位置比较寄存器 QEI_POSCMP 的值相等时会产生一个同步信号。

25.4.2 位置计数器及控制单元

位置计数器及控制单元（PCCU）通过两个寄存器 QEI_QEPCTL 与 QEI_POSCTL 来设定位置计数器的运行模式、初始化/锁存模式以及位置比较同步信号的产生。

25.4.2.1 位置计数器的运行模式

在一些系统中，位置计数器在多个旋转周期内连续累加，提供了相对初始位置的位移量。例如，将正交编码器安装在打印机的电机上，每次将打印机机头移动到初始位置时，位置计数器复位，位置计数器中的值随机头的移动而增加，从而记录了打印机机头相对初始位置移动的绝对距离。在其他系统中，位置计数器的值在每个旋转周期内由索引脉冲复位，位置计数

器的值提供了相对索引脉冲位置的角度。

位置计数器可配置成如下 4 种运行模式：

1. 位置计数器在索引脉冲到来时发生复位
2. 位置计数器在达到最大计数值时复位，
3. 位置计数器仅在第一个索引脉冲到来时复位
4. 位置计数器在单位时间输出事件时复位（频率测量）

以上所有运行模式中，计数器在增加到 QEI_POSMAX 时，如果下个计数脉冲到来，将复位到 0；计数器在减计数到 0 时，如果下个计数脉冲到来，将复位到 QEI_POSMAX。

位置计数器在索引脉冲到来时发生复位（QEI_QEPCTL.PCRM=00）

正向运行时，如果出现索引脉冲信号，位置计数器在下一个 QEI 时钟到来时被复位到 0；反向运行时，如果出现索引脉冲信号，位置计数器在下一个 QEI 时钟到来时被复位为 QEI_POSMAX 寄存器中的值。

将第一个索引脉冲边沿到来后的正交信号的边沿定义为索引标志时刻，QEI 模块记录第一个索引标志的发生（QEI_STSR.QDF）以及第一个索引事件发生时的方向（QEI_STSR.QDI），还记录第一个索引标志对应的正交信号边沿，从而使用这个相同的正交边沿完成复位操作。例如，如果第一次复位操作发生在正向运行过程中的 QEPB 的下降沿，那么以后所有正向运行的复位操作都将发生在 QEPB 的下降沿，而反向运行的复位操作发生在 QEPB 的上升沿。

每个索引事件发生时，位置计数器的值被锁存到 QEI_POSILAT 寄存器中，运行方向也被记录到 QEI_STSR.QDS 位中。如果 QEI_POSILAT 中的值不等于 0 或 QEI_POSMAX，那么位置计数器错位标志位（QEI_STSR.PCE）将置位。位置计数器错位标志位在每次索引脉冲事件发生时进行更新，而中断标志位则必须由软件清除。

位置计数器在达到最大数值时复位（QEI_QEPCTL.PCRM=01）

正向运行时，如果位置计数器的值到达 QPOSMAX 寄存器中的值，在下一个 QEI 时钟信号到来时将位置计数器复位为 0，并且将位置计数器上溢标志位置位；反向运行时，如果位置计数器的值到达 0，在下一个 QEI 时钟信号到来时将位置计数器复位到 QPOSMAX，并将位置计数器下溢标志位置位。

位置计数器仅在第一个索引脉冲到来时复位（QEI_QEPCTL.PCRM=10）

正向运行时，如果出现索引脉冲信号，位置计数器在下一个 QEI 时钟到来时被复位到 0，反向运行时，如果出现索引脉冲信号，位置计数器在下一个 QEI 时间到来时被复位为 QEI_POSMAX 寄存器中的值。但需要注意的是，以上复位操作只发生在第一次索引事件到来时，接下来的索引事件不能将位置计数器复位，位置计数器以后的复位操作与第二种情况描述的相同。

位置计数器在 Timer 事件时复位（QEI_QEPCTL.PCRM=11）

在该模式下，当一次 Timer 事件发生时，QEI_POSCNT 的值被锁存到 QEI_POSLAT 寄存器中，并且 QEI_POSCNT 被复位到 0 或 QEI_POSMAX（这由方向控制位决定），该模式可用于频

率的测量。

25.4.2.2 位置计数器的锁存

QEI 模块的索引输入（Index input）可以将位置计数器的锁存到 QEI_POSILAT 寄存器中。

索引事件锁存

许多应用中，必须在每个索引事件发生时将位置计数器的值复位且要将位置计数器运行在 32 位模式下（QEI_QEPCTL.PCRM=01 或 10）。在这种情况下，可在每个索引事件发生时将位置计数器的值以及方向进行锁存，具有如下 3 种选择。

1. 上升沿锁存（QEI_QEPCTL.IEL=01）
   位置计数器的当前值（QEI_POSCNT）在每次索引信号的上升沿被锁存到 QEI_POSILAT 寄存器中。
2. 下降沿锁存（QEI_QEPCTL.IEL=10）
   位置计数器的当前值（QEI_POSCNT）在每次索引信号的下降沿被锁存到 QEI_POSILAT 寄存器中。
3. 索引事件标志时刻锁存（QEI_QEPCTL.IEL=11）
   索引事件的标志时刻定义为索引脉冲第一个边沿后的正交信号的边沿，在这个边沿将位置计数器的当前值（QEI_POSCNT）锁存到 QEI_POSILAT 寄存器中。

位置计数器的值被锁存后，锁存事件中断标志位 QEI_STSR.IEL 被置位。

25.4.2.3 位置计数器的初始化

位置计数器可采用如下 2 种初始化方法。

1. 使用索引事件初始化
   在索引脉冲的上升沿或下降沿可对位置计数器进行初始化。如果 QEI_QEPCTL.IEI=2，将在索引脉冲的上升沿将 QEI_POSINIT 寄存器中的值装载到位置计数器 QEI_POSCNT 中；如果 QEI_QEPCTL.IEI=3，初始化过程将发生在下降沿。
2. 软件初始化
   通过向 QEI_QEPCTL.SWI 中写 1 将对位置计数器发起一次软件初始化过程。QEI_QEPCTL.SWI 位并不会自动清零，但当再次向其写 1 时会发起另一次初始化过程。

25.4.2.4 QEI 位置比较单元

QEI 模块包含一个位置比较单元，当匹配时用来产生同步输出信号和/或中断信号。

当 QEI_POSCNT=QEI_POSCMP 时即产生一次比较匹配事件，将 QEI_STSR.PCM 置位，并输出脉冲宽度可调的同步脉冲与触发外部器件。例如，如果 QEI_POSCMP=2，增计数时匹配事件将发生从 1 到 2 的跳变过程，减计数匹配事件将发生从 3 到 2 的跳变过程。

位置比较单元的脉冲扩展功能可在匹配事件发生时产生脉冲宽度可调的同步脉冲信号，如果先前输出的同步脉冲仍有效，新的匹配事件又到来，那么脉宽扩展功能将允许根据新的匹配事件产生同步脉冲信号。

25.4.3 边沿捕获单元

模块内部集成了一个边沿捕获单元，用来测量单位位移量之间的时间，利用这个模块可用下式测量低速段的转速：

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{x}{\Delta T} \qquad (3)$$

式中，单位位移量×定义为 N 个正交脉冲数。

捕获定时器 QCTMR 的计数脉冲由 QEI_CAPCTL.CCPS 位对系统总线时钟分频而来，每次 EVENT 触发脉冲都会将捕获定时器 QCTMR 中的值锁存到捕获周期寄存器 QEI_QCPRD 中，然后捕获定时器复位，QEI_STSR.CDS 置位用来表明 QEI_QCPRD 中以锁存一个新值，在软件读取捕获周期寄存器 QEI_QCPRD 的值之前可首先检查此位，向此位写 1 将对其清零。

如果满足以下两个条件，则单位位移量所经历的时间△T 测量正确。

1. 捕获定时器的值没有超过 65535
2. 而在两次事件间隔内，无转动方向的改变

如果捕获定时器的值出现上溢，上溢错误标志位 QEI_STSR.COE 将置位，如果在 2 次 EVENT 事件间隔内出现方向的改变，则错误标志位将置位。

捕获定时器 QCTMR 及捕获周期寄存器 QEI_QCPRD 的值可在如下 2 个事件发生时被锁存：

1. CPU 读 QEI_POSCNT 寄存器
2. 定时器基准单元超时事件。

定时器基准单元由一个 32 位的定时器及一个周期寄存器组成，定时器的计数时钟为系统总线时钟，当定时器的值等于周期寄存器的值时，将会产生一次超时事件，将 QEI_STSR.UTO 置位。

如果 QEI_CAPCTL.CMD=0，在 CPU 读取 QEI_POSCNT 值时，捕获定时器及捕获周期寄存器的值将会分别锁存到 QEI_CTMRLAT 及 QEI_CPRDLAT 寄存器中，如果 QEI_CAPCTL.CMD=1，在定时器基准单元超时事件发生时将把位置计数器、捕获定时器及捕获周期寄存器的值分别锁存到 QEI_POSLAT、QEI_CTMRLAT 及 QEI_CPRDLAT 寄存器中。利用此锁存功能，可用下式测量高速段的转速：

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta x}{T} \qquad (4)$$

式 3，式 4 中变量所对应的硬件寄存器如表 25-2 所示。

表 25-2 硬件寄存器变量

变量	对应的硬件寄存器
T	单元周期寄存器 QUPRD
△x	增加的位移量=QPOSLAT(k)-QPOSLAT(k-1)

x	由 CAPCTL[UPPS]位定义的固定位移量
△T	捕获周期寄存器 QCPRDLAT

25.4.4 中断结构

QEI 模块可产生 11 路中断信号：PCE、PHE、QDC、PCU、PCO、PCM、IEL 及 UTO。中断控制寄存器 QEI_IENR 用来使能/禁止相应的中断事件，中断标志寄存器 QEI_STSR 用来表明各中断事件发生的情况，并且包括全局中断标志位。

25.5 寄存器定义

25.5.1 寄存器列表

QEI 基地址：

QEI0(0x4002D000) QEI1(0x4002E000) QEI2(0x4002F000)

偏移	实例地址	名称	默认值	描述
0x00	QEI 基地址+0x00	QEI_POSCNT	0x00000000	QEI 位置计数寄存器
0x04	QEI 基地址+0x04	QEI_POSINIT	0x00000000	QEI 位置初始化寄存器
0x08	QEI 基地址+0x08	QEI_POSMAX	0xFFFFFFFF	QEI 位置最大值寄存器
0x0C	QEI 基地址+0x0C	QEI_POSCMP	0x00000000	QEI 位置比较值寄存器
0x10	QEI 基地址+0x10	QEI_POSILAT	0x00000000	QEI 位置索引锁存寄存器
0x14	QEI 基地址+0x14	QEI_POSLAT	0x00000000	QEI 位置定时锁存寄存器
0x20	QEI 基地址+0x20	QEI_UTMR	0x00000000	QEI 定时计数寄存器
0x24	QEI 基地址+0x24	QEI_UPRD	0xFFFFFFFF	QEI 定时周期寄存器
0x30	QEI 基地址+0x30	QEI_DECCTL	0x00000000	QEI Decoder 控制寄存器
0x34	QEI 基地址+0x34	QEI_QEPCTL	0x00000000	QEI 控制寄存器
0x38	QEI 基地址+0x38	QEI_POSCTL	0x00000000	QEI 位置计数比较寄存器
0x3C	QEI 基地址+0x3C	QEI_CAPCTL	0x00000000	QEI 捕获控制寄存器
0x40	QEI 基地址+0x40	QEI_QCTMR	0x00000000	QEI 捕获计数寄存器
0x44	QEI 基地址+0x44	QEI_QCPRD	0x00000000	QEI 捕获周期寄存器
0x48	QEI 基地址+0x48	QEI_CTMRLAT	0x00000000	QEI 捕获计数锁存寄存器
0x4C	QEI 基地址+0x4C	QEI_CPRDLAT	0x00000000	QEI 捕获周期锁存寄存器
0x50	QEI 基地址+0x50	QEI_IENR	0x00000000	QEI 中断使能寄存器
0x54	QEI 基地址+0x54	QEI_STSR	0x00000000	QEI 中断状态寄存器

25.5.2 寄存器描述

25.5.2.1 QEI 位置计数寄存器（QEI_POSCNT）

- 名称：QEI Position Counter Register
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POSCNT															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSCNT															
R/W															
0x0															

字段	说明
[31:0] POSCNT	**QEI 位置计数寄存器(QEI Position Counter Register)** QEI 位置计数器根据旋转方向进行递增或者递减计数，QEI 位置计数器根据不同模式进行计数(详细参考模式介绍寄存器)，会根据不同触发事件进行初始化操作； Note：QEI 位置计数器在使能前完成初始化操作，开始计数后不能修改；

25.5.2.2 QEI 位置初始化寄存器（QEI_POSINIT）

- 名称：QEI Position Counter Initial Register
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POSINIT															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSINIT															
R/W															
0x0															

字段	说明
[31:0] POSINIT	**QEI 位置初始化寄存器(QEI Position Counter Initial Register)** 位置初始化寄存器用于在外边 Index 或者 Software 事件触发下，对 QEI 位置计数器完成初始化操作；

25.5.2.3 QEI 位置最大值寄存器（QEI_POSMAX）

- 名称：QEI Position Counter Max Register
- 偏移地址：0x08
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POSMAX															
R/W															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSMAX															
R/W															
0xFFFFFFFF															

字段	说明
[31:0] POSMAX	**QEI 位置最大值寄存器(QEI Position Counter Max Register)** 位置计数器计数上限设置，当位置计数器递增计数到 Max 上限会复位到 0；或者递减计数到 0 会复位到 Max 值；

25.5.2.4 QEI 位置比较值寄存器（QEI_POSCMP）

- 名称：QEI Position Counter Compare Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POSCMP															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSCMP															
R/W															
0x0															

字段	说明
[31:0] POSCMP	**QEI 位置比较值寄存器 (QEI Position Counter Compare Register)** 位置计数器比较值，当位置计数器计数到 Compare 值会产生一个比较中断；

25.5.2.5 QEI 位置索引锁存寄存器（QEI_POSILAT）

- 名称：QEI Position Counter Index Latch Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POSILAT															
R															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSILAT															
R															
0x0															

字段	说明
[31:0] POSILAT	**QEI 位置索引锁存寄存器 (QEI Position Counter Index Latch Register)** 位置计数器 Index 事件触发锁存寄存器；

25.5.2.6 QEI 位置定时锁存寄存器 （QEI_POSLAT）

- 名称：QEI Position Counter Timer Latch Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POSLAT															
R															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSLAT															
R															
0x0															

字段	说明
[31:0] POSLAT	**QEI 位置定时锁存寄存器 (QEI Position Counter Timer Latch Register)** 位置计数器 Timer 事件触发锁存寄存器；

25.5.2.7 QEI 定时计数寄存器（QEI_UTMR）

- 名称：QTIMER Counter Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UTMR															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UTMR															
R/W															
0x0															

字段	说明
[31:0] UTMR	**QEI 定时计数寄存器（QEI Timer Counter Register）** Timer 计数器计数值等于 Compare 值时会产生中断与事件； Note：启动 Timer 前可以设置初始值，开始计数后不能修改；

25.5.2.8 QEI 定时周期寄存器 （QEI_UPRD）

- 名称：QTIMER Counter Period Register
- 偏移地址：0x24
- 默认值：0xFFFFFFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPRD															
R/W															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UPRD															
R/W															
0xFFFFFFFF															

字段	说明
[31:0] UPRD	**QEI 定时周期寄存器（QEI Timer Counter Period Register）** Timer 计数比较值，当计数值等于 Period 值时会产生中断与事件；

25.5.2.9 QEI Decoder 控制寄存器（QEI_DECCTL）

- 名称：QEI Decoder Control Register
- 偏移地址：0x30
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								DBC							
								R/W							
								0x0							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCM	QSRC						DEN	XCR	IGATE	SWAP	QBP	QAP		QIP
	R/W	R/W						R/W	R/W	R/W	R/W	R/W	R/W		R/W
	0x0	0x0						0x0	0x0	0x0	0x0	0x0	0x0		0x0

字段	说明
[23:16] DBC	**QEI 去抖窗口设置（QEI Debounce Value）** 1：1 个 SYSCLK …… n：n 个 SYSCLK
[14] DCM	**QEI 方向计数模式（QEI Direction Counter Mode）** 0：模式 0 - Counter 在 QEA 边沿递增，在 QEB 边沿递减(X1 和 X2 模式)； 1：模式 1 - Counter 在 QEA 边沿增减，方向依赖于 QEB 信号，高增低减； Note： 1）仅在 QEI WorkMode 选择方向计数模式下才有效； 2）在模式 0 下，X1 模式在 QEA 上升沿递增，在 QEB 上升沿递减；X2 模式在 QEA 双沿递增，在 QEB 双沿递减； 3）在模式 1 下，X1 模式在 QEA 上升沿并且 QEB 为高递增，为低递减；X2 在 QEA 双沿并且 QEB 为高递增，为低递减；
[13:12] QSRC	**QEI 位置计数器源选择（QEI Position-counter Source Selection）** 00：正交计数模式-X1 模式在 QA 上下边沿计数，X2 模式在 QA/QB 上下边沿计数 01：方向计数模式-参考[14]位描述 10：向上计数模式- Counter 在 QEA 边沿递增，和 QEB 无关(建议不配 QEB IO 复用) 11：向下计数模式- Counter 在 QEA 边沿递减，和 QEB 无关(建议不配 QEB IO 复用) Note：采用单边沿还是双边沿依据选择 X1 还是 X2 模式；
[7] DEN	**QEI 去抖使能（QEI Debounce Enable）** 0：Bypass 1：Enable
[6] XCR	**QEI 时钟速率控制（QEI Clock Rate Control）** 0：X1 模式 1：X2 模式 Note： 1）正交计数模式，X1 模式在 QA 上下边沿计数，X2 模式在 QA/QB 上下边沿计数； 2）方向计数模式，X1 模式在 QA 上升沿计数，X2 模式在 QA 上下边沿计数；
[5] IGATE	**QEI 索引信号使能（QEI Index Gate Enable）** 0：索引信号有效 1：索引信号无效（Gate 生效）
[4] SWAP	**QEI 正交时钟交换控制（QEI QEA/B SWAP Enable）** 0：内部不做交换处理 1：内部实现 A/B 交换
[3] QBP	**QEI QEB 信号输入极性选择（QEI QEB Input Polarity）** 0：Default 为低电平输入，高电平有效； 1：取反输入；
[2] QAP	**QEI QEA 信号输入极性选择（QEI QEA Input Polarity）** 0：Default 为低电平输入，高电平有效； 1：取反输入；
[0] QIP	**QEI Index 信号输入极性选择（QEI Index Input Polarity）**

0：Default 为低电平输入，高电平有效；
1：取反输入；

25.5.2.10 QEI 控制寄存器（QEI_QEPCTL）

- **名称：QEI Contorl Register**
- **偏移地址：0x34**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		PCRM					SWI	IEI		IEL				UTE	QPE
		R/W					R/W1C	R/W		R/W				R/W	R/W
		0x0					0x0	0x0		0x0				0x0	0x0

字段	说明
[13:12] PCRM	**QEI 位置计数器复位选择（QEI Position Counter Reset Selection）** 00：位置计数器在 Index 脉冲后的第一个正交信号边沿复位(区分正反方向)； 01：位置计数器在正向计数时计数到最大值复位，方向计数时计数到最小值复位； 10：位置计数器仅在第一个 Index 脉冲后的第一个正交信号边沿复位； 11：位置计数器在 Timer 事件时复位； Note：每种模式下均包含位置计数器出现上溢与下溢时触发复位；
[8] SWI	**QEI 位置计数器初始化使能（QEI Position Counter Software Initial Enable）** 0：Disable 1：写 1 发起初始化操作，将位置计数器初始化(初始化为 INIT 的值-0x4 寄存器)； Note：该 Bit 为写 1 自清，写 1 发起软件 Initial 操作，写 0 无效；
[7:6] IEI	**QEI 位置计数器索引初始选择（QEI Position Counter Index Initial Selection）** 0x：Disable 10：在 Index 信号的上升沿将位置计数器初始化为 Initial 值； 11：在 Index 信号的下降沿将位置计数器初始化为 Initial 值；
[5:4] IEL	**QEI 位置计数器索引锁存选择（QEI Position Counter Index Latch Selection）** 00：Disable 01：在 Index 信号的上升沿将位置计数器的值锁存； 10：在 Index 信号的下降沿将位置计数器的值锁存； 11：在 Index 信号有效后的第一个正交信号边沿锁存(区分正反方向)；
[1] UTE	**QEI Timer 使能（QEI Timer Enable）：** 0：不使能 1：使能
[0] QPE	**QEI 使能（QEI Enable）：** 0：不使能 1：使能

25.5.2.11 QEI 位置计数比较寄存器（QEI_POSCTL）

- 名称：QEI Position Contorl Register
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														PSE	CCE
														R/W	R/W
														0x0	0x0

字段	说明
[1] PSE	**QEI 位置计数比较影子寄存器使能（QEI Position Counter Compare Shadow Enable）** 0：关闭 Compare 影子寄存器功能 1：使能 Compare 影子寄存器功能
[0] CCE	**QEI 位置计数比较使能（QEI Position Counter Compare Enable）** 0：关闭位置计数器比较功能 1：使能位置计数器比较功能

25.5.2.12 QEI 捕获控制寄存器（QEI_CAPCTL）

- 名称：QEI Capture Contorl Register
- 偏移地址：0x3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				UPPS					CCPS					CMD	CEN
				R/W					R/W					R/W	R/W
				0x0					0x0					0x0	0x0

字段	说明
[11:8] UPPS	**QEI 捕获事件分频（QEI Capture Position Event Division）** 0x0: QCLK/1 0x1: QCLK/2 0x2: QCLK/4 …… 0xB: QCLK/2048
[6:4] CCPS	**QEI 捕获定时器分频（QEI Capture Timer Clock Division）** 0x0: SYSCLK/1 0x1: SYSCLK/2 0x2: SYSCLK/4 …… 0x7: SYSCLK/128
[1] CMD	**QEI 捕获锁存模式（QEI Capture Latch Mode）用于 M 法测速** 0：在 CPU 读取位置计数器(0x00 寄存器)时锁存 1：在定时器超时事件时锁存（QEI TIMER 定时器）
[0] CEN	**QEI 捕获使能（QEI Capture Enable）** 0：关闭捕获功能

1：使能捕获功能

25.5.2.13 QEI 捕获计数寄存器（QEI_QCTMR）

- **名称：QEI Capture Counter Register**
- **偏移地址：0x40**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTMR															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTMR															
R/W															
0x0															

字段	说明
[31:0] CTMR	**QEI 捕获计数寄存器（QEI Capture Counter Register）** 捕获定时计数器，基于 Capture Time Clock 进行计时，即基于分频后的时钟进行计时； Note：每次捕获更新事件会对 Counter 清 0，重新下一次计时；

25.5.2.14 QEI 捕获周期寄存器（QEI_QCPRD）

- **名称：QEI Capture Period Register**
- **偏移地址：0x44**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPRD															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPRD															
R/W															
0x0															

字段	说明
[31:0] CPRD	**QEI 捕获周期寄存器（QEI Capture Period Register）** 每次捕获更新事件会将 Capture Counter 计数值锁存，并产生中断；

25.5.2.15 QEI 捕获计数锁存寄存器（QEI_CTMRLAT）

- 名称：QEI Capture Counter Latch Register
- 偏移地址：0x48
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTMRLAT															
R															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTMRLAT															
R															
0x0															

字段	说明
[31:0] CTMRLAT	**QEI 捕获计数锁存寄存器（QEI Compare Counter Latch Register）** 捕获定时器的值在两种事件下可以锁存到寄存器中： 1）CPU 读取位置计数器的值； 2）TMR 定时器超时事件；

25.5.2.16 QEI 捕获周期锁存寄存器（QEI_CPRDLAT）

- 名称：QEI Capture Period Latch Register
- 偏移地址：0x4C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPRDLAT															
R															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPRDLAT															
R															
0x0															

字段	说明
[31:0] CPRDLAT	**QEI 捕获周期锁存寄存器（QEI Capture Period Latch Register）** 捕获定时器的值在两种事件下可以锁存到寄存器中： 1）CPU 读取位置计数器的值； 2）TMR 定时器超时事件；

25.5.2.17 QEI 中断使能寄存器（QEI_IENR）

- 名称：QEI Interrupt Enable Register
- 偏移地址：0x50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					CDE	PRE	PIE	UTO	IEL	PCM	PCO	PCU	QDC	QPE	PCE
					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[10] CDE	**QEI Capture 捕获成功中断使能（QEI Capture Done Interrupt Enable）** 0：关闭 1：使能
[9] PRE	**QEI 位置计数器复位中断使能（QEI Position Counter Reset Interrupt Enable）** 0：关闭 1：使能
[8] PIE	**QEI 位置计数器初始化中断使能（QEI Position Counter Initial Interrupt Enable）** 0：关闭 1：使能
[7] UTO	**QEI 定时器溢出中断使能（QEI Timer Overflow Interrupt Enable）** 0：关闭 1：使能 Position Counter Unit Timer 比较匹配中断；
[6] IEL	**QEI 位置计数器锁存中断使能（QEI Position Counter Latch Interrupt Enable）** 0：关闭 1：使能
[5] PCM	**QEI 位置计数器比较中断使能（QEI Position Counter Compare Interrupt Enable）** 0：关闭 1：使能
[4] PCO	**QEI 位置计数器上溢中断使能（QEI Position Counter Overflow Interrupt Enable）** 0：关闭 1：使能 位置计数器上溢中断；
[3] PCU	**QEI 位置计数器下溢中断使能（QEI Position Counter Underflow Interrupt Enable）** 0：关闭 1：使能
[2] QDC	**QEI 正交方向变化中断使能（QEI Direction Change Interrupt Enable）** 0：关闭 1：使能
[1] QPE	**QEI 相位错误中断使能（QEI Phase Error Interrupt Enable）** 0：关闭 1：使能
[0] PCE	**QEI 位置计数器错误中断使能（QEI Position Counter Error Interrupt Enable）** 0：关闭 1：使能

25.5.2.18 QEI 中断状态寄存器（QEI_STSR）

- 名称：QEI Interrupt Status Register
- 偏移地址：0x54
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															QDF
															R
															0x0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QDI	QDS	COE	CDE	FIS	CDS	PRS	PIS	UTO	IEL	PCM	PCO	PCU	QDC	PHE	PCE
R	R	R/W1C	R/W1C	R	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[16] QDF	**QEI 首次电机转向标志位（QEI Direction First Index Status）** 0：首次索引正交脉冲有效时电机方向为反转 1：首次索引正交脉冲有效时电机方向正转
[15] QDI	**QEI 每次电机转向标志位（QEI Direction Index Latch Status）** 0：每次索引正交脉冲有效时电机方向为反转 1：每次索引正交脉冲有效时电机方向为正转
[14] QDS	**QEI 实时电机转向标志位（QEI Direction Status）** 0：反转 1：正转
[13] COE	**QEI 捕获上溢错误标志位（QEI Capture Overflow Error Status）** 0：无错误 1：捕获计数器出现上溢错误
[12] CDE	**QEI 捕获方向错误标志位（QEI Capture Direction Error Status）** 0：无错误 1：在两次捕获事件间隔内方向发生了变化
[11] FIS	**QEI 出现首次索引标志位（QEI The First Index Status）** 0：没出现首次索引正交脉冲 1：有出现首次索引正交脉冲
[10] CDS	**QEI Capture 捕获成功状态标志位（QEI Capture Done Interrupt Status）** 0：Capture 捕获未成功 1：Capture 捕获成功标志 Note: Capture 捕获成功状态标志位；
[9] PRS	**QEI 位置计数器复位标志位（QEI Position Counter Reset Interrupt Status）** 0：位置计数器复位未触发 1：位置计数器复位触发标志 Note: 位置计数器出现复位操作，其中包括多种复位方式；
[8] PIS	**QEI 位置计数器初始化标志位（QEI Position Counter Initial Interrupt Status）** 0：位置计数器初始化未触发 1：位置计数器初始化触发标志 Note: 位置计数器出现初始化操作，其中包括多种初始化方式；
[7] UTO	**QEI 定时器溢出标志位（QEI Timer Overflow Interrupt Status）** 0：Timer 比较计数未完成 1：Timer 比较计数完成标志 Note: Position Counter Unit Timer 比较匹配中断；
[6] IEL	**QEI 位置计数器锁存标志位（QEI Position Counter Latch Interrupt Status）** 0：计数器锁存未触发 1：计数器锁存触发标志 Note: 位置计数器触发锁存操作，包括多种 Latch 方式；
[5] PCM	**QEI 位置计数器比较成功标志位（QEI Position Counter Compare Interrupt Status）** 0：位置计数器比较匹配未成功 1：位置计数器比较匹配成功标志

[4] PCO	**QEI 位置计数器上溢标志位（QEI Position Counter Overflow Interrupt Status）** 0：位置计数器未上溢 1：位置计数器上溢标志
[3] PCU	**QEI 位置计数器下溢标志位（QEI Position Counter Underflow Interrupt Status）** 0：位置计数器未下溢 1：位置计数器下溢标志
[2] QDC	**QEI 正交方向变化标志位（QEI Direction Change Interrupt Status）** 0：正交方向未变化 1：正交方向变化标志 正交方向变化中断；
[1] PHE	**QEI 相位错误标志位（QEI Phase Error Interrupt Status）** 0：相位未错误 1：相位错误标志 Note：QEA/QEB 信号的相位同时变化为相位错误；
[0] PCE	**QEI 位置计数器错误标志位（QEI Position Counter Error Interrupt Status）** 0：位置计数器未错误 1：位置计数器错误标志 Note: 1）位置计数值大于 Max 值; 2）位置计数值从 0 计数到全 F(并且 Max 不为全 F)

26 独立看门狗（IWDG）

26.1 简介

独立看门狗具有安全性高、使用灵活的特点。可用于检测并解决软件错误导致的故障，在定时器到达指定的超时值时触发系统复位。

独立看门狗（IWDG）由其专用低速时钟（LSI）驱动，因此即便在主时钟发生故障时仍然保持工作状态。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。

26.2 主要特性

IWDG 主要具有以下特性：

- 自由运行递减计数器
- 16 位重载计数值寄存器，计数时钟支持 4 到 512 分频
- 时钟由独立 RC 振荡器（LSI）提供（可在待机模式下独立运行）
- 当递减计数器值达到 0x0 时产生复位（如果看门狗已激活）

26.3 结构框图

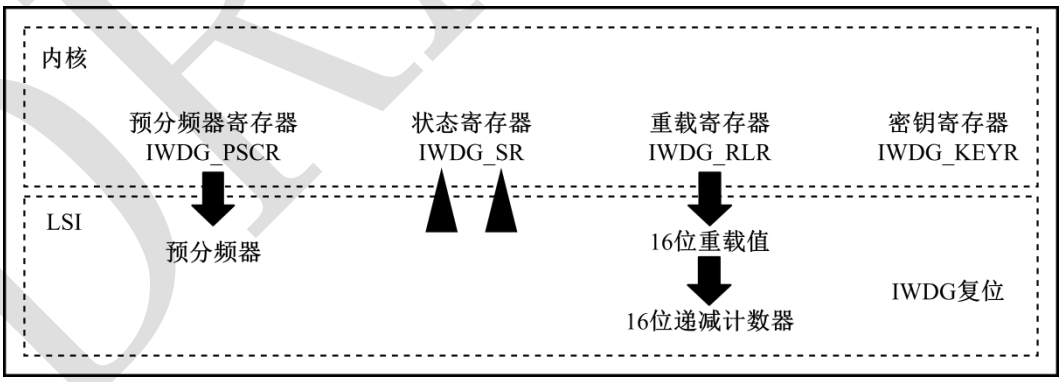


图 26-1　IWDG 架构示意图

26.4 功能描述

26.4.1 IWDG 键值功能

将键值 0xCCCC 写到键寄存器（IWDG_KEYR[15:0]）中，将启动独立看门狗。看门狗内部计数器开始从复位值 0xFFFF 递减计数，当内部计数器计数到终值（0x0000）时将会产生一个系统复位或触发中断。

任何时候将键值 0xAAAA 写到键寄存器（IWDG_KEYR[15:0]）中，重载寄存器（IWDG_RLR）的值就会被重载到计数器，从而避免产生看门狗复位。

将键值 0xDDDD 写到键寄存器（IWDG_KEYR[15:0]）中，将停止独立看门狗。

将键值 0x5003 写到键寄存器（IWDG_KEYR[15:0]）中，将解除 IWDG 的寄存器访问保护，详见“13.4.2IWDG 寄存器访问保护”说明。

注意：重载操作（将键值 0xAAAA 写到 IWDG_KEYR[15:0]）必须在启动 IWDG 后，且 IWDG_SR 寄存器中的 RLVUPD 和 PSCUPD 位都为 0 时才可以进行。

26.4.2 IWDG 寄存器访问保护

IWDG 预分频寄存器（IWDG_PSCR）、重加载寄存器（IWDG_RLR）及控制寄存器（IWDG_CR）具有写访问保护。

若要修改上述寄存器，首*先需*要将 0x3FAC 写入键寄存器（IWDG_KEYR[15:0]）解*除*上述寄存器的访问保护；写入其他值，则使寄存器访问保护重新生效，下次修改则需要重新解除访问保护。这意味着重装载操作（写 0xAAAA）也会启动写保护功能。

状态寄存器指示预分频值和递减计数器是否正在被更新。

26.4.3 IWDG 调试模式

当芯片进入调试模式时（Cortex™-M4 内核停止），IWDG 计数器会根据 SYSCTRL_SYSDCR 寄存器中的 IWDGDEN 位（SYSCTRL_SYSDCR. IWDGDEN）选择继续正常工作或者停止工作。

26.4.4 IWDG 中断模式和复位模式

IWDG 支持两种工作模式：中断模式和复位模式。通过 IWDG_CR 寄存器的 MODE 位选择 IWDG 的工作模式。

中断模式下，IWDG 在未重载导致超时后，IWDG_SR 寄存器中的 TOIF 将会置位，并触发中断（如果使能 IWDG 中断），且不会导致系统复位。

复位模式下，IWDG 在未重载导致超时后，触发一个系统复位信号，如果 RCU_SRSTSR 中的

IWRSTE 位（RCU_SRSTSR.IWRSTE）已经使能，则会引发系统复位（如果 IWRSTE 位未使能，则同样不会导致系统复位，直到该位使能）。

26.4.5 IWDG 计时

表 26-1 32 kHz 频率条件下 IWDG 超时周期的最小值/最大值

预分频器	IWDG_PSCR[2:0]	最短超时（ms） IWDG_RLR = 0x0	最长超时（ms） IWDG_RLR = 0xFFFF
/4	0	0.125	8192
/8	1	0.25	16384
/16	2	0.5	32768
/32	3	1	65536
/64	4	2	131072
/128	5	4	262144
/256	6	8	524288
/512	7	16	1048576

注意：

- *这些时间均依据LSI-32KHz 时钟给出。实际上芯片内部的LSI 时钟频率会在16kHz 到48kHz 之间变化。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的；*
- *配置 IWDG_RLR 及 IWDG_PSCR 都需要从 APB 时钟域同步到 LSI 时钟域下，会一定程度影响 IWDG 的精度。*

26.4.6 IWDG 中断和状态

IWDG_SR 寄存器中定义了下述几个中断标志位和状态标志位。

标志名称	中断使能	标志位	备注
超时中断	TOIE	TOIF	IWDG 处于中断模式下时，未重载导致看门狗超时。
重载值更新状态	-	RLVUPD	置位时表示重载值正在更新，完成后硬件自动清 0。
分频值更新状态	-	PSCUPD	置位时表示分频值正在更新，完成后硬件自动清 0。

注意：IWDG 配置重载值寄存器（IWDG_RLR）和预分频寄存器（IWDG_PSCR）后，应当检查 RLVUPD 和 PSCUPD 标志，只有更新动作完成后，才可对看门狗进行重载操作。

26.5 寄存器定义

26.5.1 寄存器列表

IWDG 基地址：

IWDG(0x4000C000)

偏移	实例地址	名称	默认值	描述
0x00	IWDG 基地址+0x00	IWDG_KEYR	0x00000000	IWDG 键寄存器
0x04	IWDG 基地址+0x04	IWDG_CR	0x00000000	IWDG 控制寄存器
0x08	IWDG 基地址+0x08	IWDG_RLR	0x0000FFFF	IWDG 重载寄存器
0x0C	IWDG 基地址+0x0C	IWDG_PSCR	0x00000000	IWDG 预分频寄存器
0x10	IWDG 基地址+0x10	IWDG_SR	0x00000000	IWDG 状态寄存器

26.5.2 寄存器描述

26.5.2.1 IWDG 键寄存器（IWDG_KEYR）

- 名称：**IWDG Key Register**
- 偏移地址：**0x00**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY															
R/W															
0x0															

字段	说明
[15:0] KEY	**IWDG 键值（IWDG Key Value）** 1. 必须每隔一段时间便通过软件对这些位写入键值 0xAAAA 来执行喂狗操作，否则当计数器计数到 0 时，看门狗会产生复位； 2. 写入键值 0x3FAC 可使能对 CR、RLV 和 PSC 寄存器的访问； 3. 写入键值 0xAAAA 可发起 RLV 值的重装； 4. 写入键值 0xCCCC 可使能看门狗； 5. 写入键值 0xDDDD 可关闭看门狗； Note： 1. 当对 CR、RLV 及 PSC 寄存器的写操作，必须先向 KEY 寄存器写入 0x3FAC 完成解锁，然后开始写操作，否则配置操作无效。 2. 解锁后，建议在完成操作后执行加锁操作，即将 KEY 值写为无效值，起到防误操作。

26.5.2.2 IWDG 控制寄存器（IWDG_CR）

- 名称：**IWDG Control Register**
- 偏移地址：**0x04**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														TOIE	MODE
														R/W	R/W
														0x0	0x0

字段	说明
[1] TOIE	**IWDG 超时中断使能位（IWDG Timeout Interruput Enable）** 需要由软件解锁，写访问前对 KEY 寄存器写入值 0x3FAC； 0: 关闭 1: 使能 Note：

	中断模式下，使能 TOIE 才会产生中断请求，TOIE 不使能只会产生 Pengding，需要软件进行查询；
[0] MODE	**IWDG 模式配置位（IWDG Mode Config）** 需要由软件解锁，写访问前对 KEY 寄存器写入值 0x3FAC； 0: 复位模式 1: 中断模式 Note： 1. 中断模式下，不会产生复位； 2. 复位模式下，为了调试方便，可以开启 IWDG 的调试使能位进行 Debug 调试，避免调试中复位； 3. 复位模式下，当产生复位后，可查看复位状态标志位，IWDG 产生的复位逻辑不会影响到复位状态寄存器，便于查询复位的来源。

26.5.2.3 IWDG 重载寄存器（IWDG_RLR）

- **名称：IWDG Reload Register**
- **偏移地址：0x08**
- **默认值：0x0000FFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RLV															
R/W															
0xFFFF															

字段	说明
[15:0] RLV	**IWDG 重载值（IWDG Reload Value）** 1. RLV 区域受写访问保护，写操作前需要由软件解锁，对 KEY 寄存器写入值 0x3FAC 完成解锁； 2. 对 KEY 寄存器写入值 0xAAAA 时, RLV 域的值就会重装载到看门狗计数器中，看门狗计数器便从该装载的值开始递减计数。 Note： 对 KEY 寄存器写入值 0xAAAA 时， RLV 域的值就会重装载到看门狗计数器中，RLVUPD 状态位会硬件置 1，软件必须等待 RLVUPD 位置 0，表示 RLV 值更新成功，才能执行下一笔操作。

26.5.2.4 IWDG 预分频寄存器（IWDG_PSCR）

- 名称：IWDG Prescaler Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													PSC		
													R/W		
													0x0		

字段	说明
[2:0] PSC	**IWDG 预分频器（IWDG Prescaler Divider）** 000: 4 分频 001: 8 分频 010: 16 分频 011: 32 分频 100: 64 分频 101: 128 分频 110: 256 分频 111: 512 分频 Note: 1. PSC 区域受写访问保护，写操作前需要由软件解锁，对 KEY 寄存器写入值 0x3FAC 完成解锁； 2. 当软件更改 PSC 域的值后，PSCUPD 状态位会硬件置 1，软件必须等待 PSCUPD 位置 0，表示 PSC 值更新成功，才能执行下一笔操作。

26.5.2.5 IWDG 状态寄存器（IWDG_SR）

- **名称：IWDG Status Register**
- **偏移地址：0x10**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TOIF	RLVUPD	PSCUPD
													R/W1C	R	R
													0x0	0x0	0x0

字段	说明
[2] TOIF	**IWDG 超时中断标志位（IWDG Timeout Interrupt Flag）** 0: No Pending 1: Irq Pending Note: 该位由软件写 1 清 0，写 0 无效
[1] RLVUPD	**IWDG 计数器重载值更新（IWDG Reload Value Update）** 当软件 KEY 寄存器写入值 0xAAAA 时，RLV 域的值就会重装载到看门狗计数器中，硬件会自动将该位置 1 以指示重载值正在更新，当完成重载值更新操作后，硬件会自动将该位置 0。 Note：软件执行 RLV 重装操作(喂狗)后，必须等待该位为置 0，表示重载值更新完成，方可执行下一笔操作。
[0] PSCUPD	**IWDG 预分频器值更新（IWDG Prescaler Update）** 当软件对 PSC 进行写操作时，硬件会自动将该位置 1 以指示预分频器值正在更新，当完成预分频器值更新操作后，硬件会自动将该位置 0。 Note：软件执行 PSC 的写操作后，必须等待该位为置 0，表示预分频器值更新完成，方可执行下一笔操作。

27 窗口看门狗（WWDG）

27.1 简介

窗口看门狗（WWDG）时钟由 APB0 时钟经预分频后提供，窗口看门狗通常被用来监测，通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非递减计数器的值在 T6 位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生一个系统复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的递减计数器值，也会产生复位。这意味着必须在限定的时间窗口内刷新计数器。

27.2 主要特性

WWDG 主要具有以下特性：

- 自由运行递减计数器
- 复位条件（当 RCU_SRSTSR 寄存器的 WWRSTE 位置 1，允许 WWDG 系统复位功能，详见 RCU_SRSTSR 寄存器介绍）
  - 当递减计数器值小于 0x40 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）
- 提前唤醒中断（EWI）：当递减计数器减至 0x40 时触发（如果已使能且看门狗已激活），可用于在中断内重载计数器以避免 WWDG 复位

27.3 结构框图

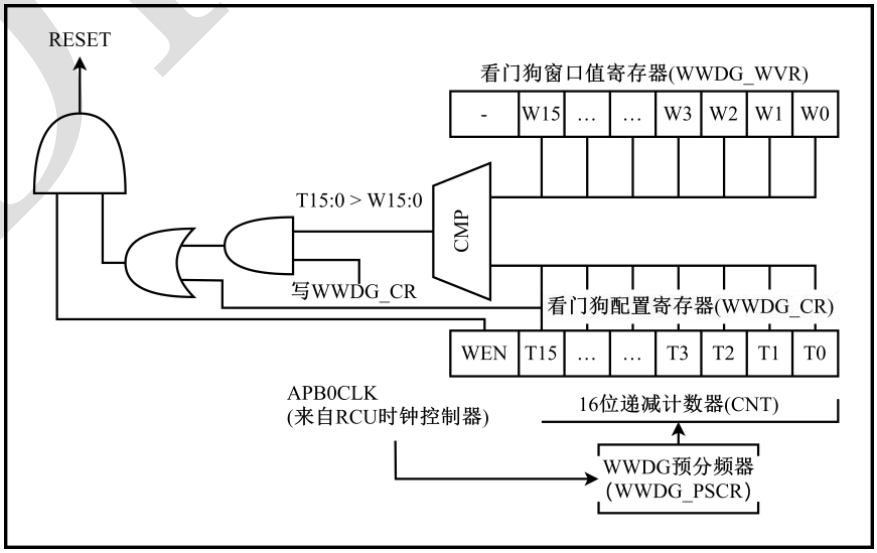


图 27-1 WWDG 架构框图

27.4 功能描述

如果激活看门狗（WWDG_CR 寄存器的 WEN 位置 1），当 16 位递减计数器从 0x40 滚动到 0x3F 时会引发复位信号；如果软件在计数器值（WWDG_CVR）大于窗口寄存器（WWDG_WVR）中所存储的值时重载计数器，也会产生复位信号。

如果 RCU_SRSTSR 寄存器的 WWRSTE 位置 1，则 WWDG 的复位信号最终导致系统复位。

应用程序在正常运行过程中必须定期地写 WWDG_CVR 寄存器以防止发生复位。只有当计数器值低于窗口寄存器值时，才能执行此操作。存储在 WWDG_CVR 寄存器中的值须介于 0xFFFF 和 0x40 之间（如果设置为 0x40 将无法触发提前唤醒中断 Early Wakeup Interrupt），用户应当根据实际应用需求合理配置。

27.4.1 使能看门狗

在系统复位后，看门狗处于关闭状态。可通过设置窗口看门狗控制寄存器（WWDG_CR）中的 WEN 位来使能看门狗，也可以按需随时关闭窗口看门狗功能。

27.4.2 控制递减计数器

在使能窗口看门狗之后递减计数器（WWDG_CVR[15:0]）就开始运行，每个计数周期执行一次递减操作。使能看门狗前，用户需要根据自己的应用需求确保以下限制：

- 递减计数器的值不应当配置 0x40 以下，否则启动窗口狗后将立即引发系统复位信号。
- 如果需要启用提前唤醒中断（EWI）功能，递减计数器的值不应当配置在 0x41 以下（仅在递减计数器从 0x41 递减至 0x40 时触发提前唤醒中断）

在 WWDG 开始工作后，为了防止系统复位，当递减计数器的值低于窗口寄存器（WWDG_WVR[15:0]）的值，且大于 0x3F 时，必须对递减计数器进行重载（向 WWDG_CVR[15:0]写入新值）。重载后递减计数器从新值重新开始按计数周期递减。

27.4.3 看门狗中断高级特性

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断（EWI）。通过设置控制寄存器（WWDG_CR）中 EWIE 位使能该中断。当递减计数器的值从 0x41 递减至 0x40 时，将生成中断。在 WWDG 递减计数器继续递减到 0x3F 之前，可以使用相应的中断服务程序来触发特定操作（例如通信或数据记录）或重载递减计数器避免复位。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序可用来重载 WWDG 递减计数器以避免 WWDG 复位，然后再触发所需操作。

通过对中断状态寄存器（WWDG_ISR）写 1 对 EWIF 清 0，即清除提前中断。

注意：在提前唤醒中断的处理中，如果处理时间超过一个计数周期未重载递减计数器，计数

器从 0x40 递减至 0x3F，WWDG 将发出复位信号，引发系统复位（如果使能复位）。因此所有操作必须在这一个计数周期内完成，否则应当先关闭 WWDG 或重载递减计数器再执行其他操作。

27.4.4 复位使能和调试模式

WWDG 的复位功能，还需要在 RCU_SRSTSR 寄存器的 WWRSTE 位置 1，使能 WWDG 的系统复位功能，否则 WWDG 发出的复位信号并不会引发实际的系统复位，直到 RCU_SRSTSR 寄存器的 WWRSTE 位置 1。

还可以通过将 SYSCTRL_SYSDCR 寄存器的 WWDGDEN 位置 1，开启 WWDG 的调试模式。开启后，当 MCU 进入调试模式，并且处于 Halt 状态（Cortex™-M4 内核停止），WWDG 的递减定时器也会停止工作，直到退出调试模式或调试处于非 Halt 状态。

27.4.5 看门狗设置

图 27-2 为 WWDG 计数时序图。

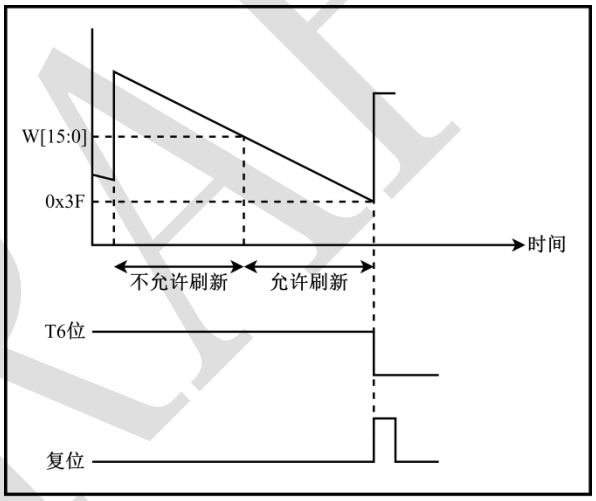


图 27-2 WWDG 计数时序图

超时值的计算公式如下：

$$t_{WWDG} = t_{APB0CLK} \times （PSC + 1）\times （CV + 1\text{-}64）\ ms$$

$$t_{WWDG} = t_{APB0CLK} \times （2\text{^}SCALE）\times （T\ [15:0] + 1）$$

其中：

PSC ： WWDG_PSCR 窗口看门狗分频寄存器

CV ： WWDG_CVR 窗口看门狗递减计数器

t_{WWDG} ： WWDG 超时时间

$t_{APB0CLK}$ ： APB0 时钟周期，以 ms 为测量单位

例如，假设 APB0 频率为 50MHz（$t_{APB0CLK}$=1/50000 ms），PSC 值为 99，CV 值为 999：

$$t_{WWDG} =（1/50000）\times（99 + 1）\times（999 + 1\ -64）= 2\ ms$$

27.5 寄存器定义

27.5.1 寄存器列表

WWDG 基地址：

WWDG(0x4000D000)

偏移	实例地址	名称	默认值	描述
0x00	WWDG 基地址+0x00	WWDG_CR	0x00000000	WWDG 控制寄存器
0x04	WWDG 基地址+0x04	WWDG_WVR	0x0000FFFF	WWDG 窗口值寄存器
0x08	WWDG 基地址+0x08	WWDG_CVR	0x0000FFFF	WWDG 计数器值寄存器
0x0C	WWDG 基地址+0x0C	WWDG_PSCR	0x00000000	WWDG 预分频寄存器
0x10	WWDG 基地址+0x10	WWDG_ISR	0x00000000	WWDG 状态寄存器

27.5.2 寄存器描述

27.5.2.1 WWDG 控制寄存器（WWDG_CR）

- 名称：**WWDG Control Register**
- 偏移地址：**0x00**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														EWIE	WEN
														R/W	R/W
														0x0	0x0

字段	说明
[1] EWIE	**WWDG 提前唤醒中断使能位（WWDG Early Wakeup Interrupt Enable）** 使能 Early 中断，在计数到 0x40 产生中断，用于提醒喂狗. 0: 关闭 1: 使能
[0] WEN	**WWDG 使能位（WWDG Enable）** 0: 关闭看门狗 1: 使能看门狗

27.5.2.2 WWDG 窗口值寄存器（WWDG_WVR）

- 名称：**WWDG Window Register**
- 偏移地址：**0x04**
- 默认值：**0x0000FFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WV															
R/W															
0xFFFF															

字段	说明
[15:0] WV	**WWDG 窗口比较值（WWDG Window Value）** Note: 用于与递减计数器进行比较的窗口值，只有当计数器值低于窗口寄存器值时，才能重载计数器值，否则也会产生复位操作。

27.5.2.3 WWDG 计数器值寄存器（WWDG_CVR）

- 名称：WWDG Counter Register
- 偏移地址：0x08
- 默认值：0x0000FFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CV															
R/W															
0xFFFF															

字段	说明
[15:0] CV	**WWDG 递减计数值（WWDG Counter Value）** Note: 1、用来存储看门狗计数器的值，它每隔 N(取决于分频系统)个 PCLK 周期递减一次，当它从 0x40 滚动到 0x3F 时会产生复位。 2、正常运行过程中必须定期地写该寄存器以防止芯片发生复位，只有当计数器值低于窗口比较值时，才能执行此操作，否则也会产生复位操作。

27.5.2.4 WWDG 预分频寄存器（WWDG_PSCR）

- 名称：WWDG Prescaler Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC															
R/W															
0x0															

字段	说明
[15:0] PSC	**WWDG 预分频器（WWDG Prescaler divider）** 每次发生更新事件时要装载到实际预分频器寄存器的值； 0:不分频，系统时钟 1:2 分频 2:3 分频 3:4 分频 …… Note：分频关系 = PSC+1 分频，例如：(0~65535)

27.5.2.5 WWDG 状态寄存器（WWDG_ISR）

- 名称：**WWDG Status Register**
- 偏移地址：**0x10**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															EWIF
															R/W1C
															0x0

字段	说明
[0] EWIF	**WWDG 提前唤醒中断标志位（WWDG Early Wakeup Interrupt Flag）** 0: No Pending 1: Irq Pending 软件写 1 清 0，写 0 无效

28 基本定时器（TMR7/TMR8）

28.1 简介

定时器 TMR7 和 TMR8 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。基础定时器不仅可用作通用定时器以生成时基，还可以专门用于数模转换器(DAC)的触发驱动。实际上，此类定时器内部连接到 DAC 并能通过其触发驱动 DAC。

28.2 主要特性

TMR7/8 主要具有以下特性：

- 16 位的自动重载向上计数器。
- 16 位预分频器，用于对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间
- 16 位的计数比较值和周期值寄存器。
- 两种工作模式：
  - 循环模式，计数完成后自动重载并继续计数
  - 单次模式，计数完成后自动关闭定时器的使能
- 支持 Master 主模式，可用于触发其他定时器及 DAC 的同步电路

28.3 结构框图

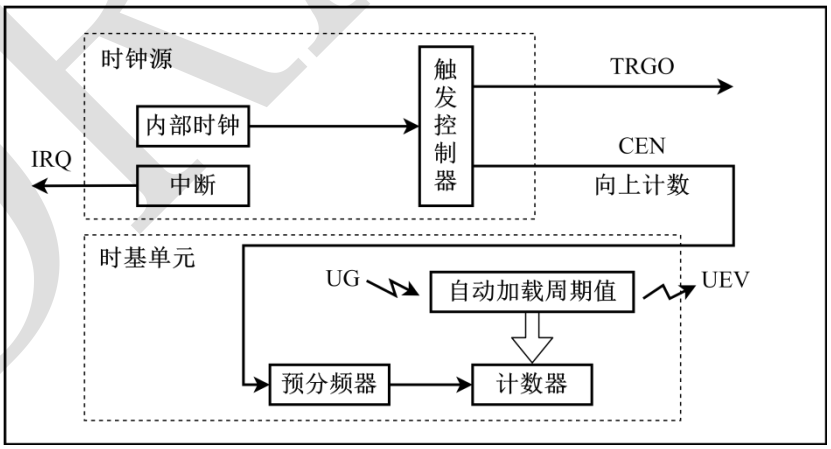


图 28-1 TMR 结构框图

28.4 功能描述

28.4.1 内部信号

下表主要介绍基础定时器的中断及触发事件的描述：

表 28-1 TMRx 内部信号说明

Signal Name	Signal Type	Description
PCLK	Input	TMRx APB Clock
TRGO	Output	TMRx 内部触发输出信号，可用于触发其他外设
IRQ	Output	TMRx 更新事件中断

28.4.2 时钟控制

TMRx 模块总线时钟为 APB 总线时钟，内部计数器时钟也由 APB 时钟驱动。

对于定时器的计数器时钟，TMRx_CR0 寄存器中的 CEN 位和 TMRx_UGR 寄存器中的 UG 位用作控制位，当对 CEN 位写 1 时，预分频器的时钟就由内部时钟 CLK_IN 提供；通过软件对 UG 位写 1，则会使预分频器清零并重新开始计数。

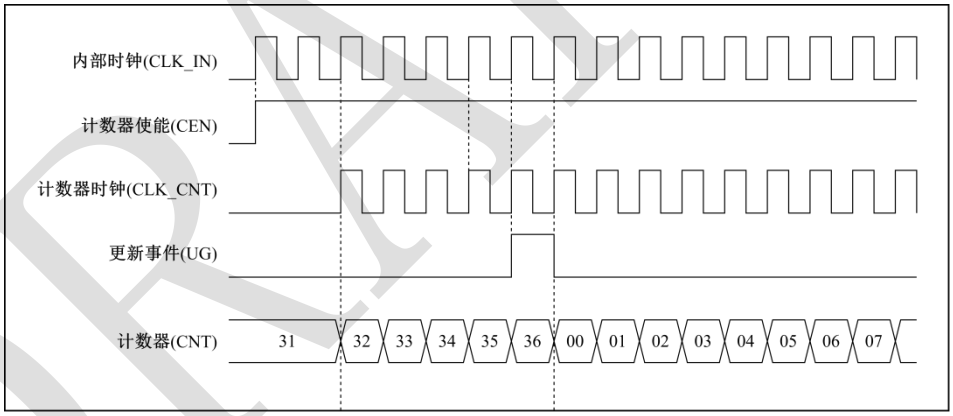


图 28-2　正常模式下的控制时序图

28.4.3 时基单元

基础定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成，计数器采用递增方式计数，计数器的时钟可通过预分频器进行分频。

基础定时器的自动重载计数器(TMRx_CNTR)、预分频寄存器(TMRx_PSCR)及计数周期寄存器(TMRx_CPR)可通过软件进行读写，即使在计数器运行中也可执行读写操作。

时基单元包含：

- 自动重载计数器(TMRx_CNTR)
- 预分频寄存器(TMRx_PSCR)
- 计数周期寄存器(TMRx_CPR)

其中预分频寄存器(TMRx_PSCR)、计数周期寄存器(TMRx_CPR)都是可预装载的。对预装载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器立即生效，也可以在每次发生更新事件时更新到影子寄存器，这取决于定时器 TMRx_CR0 控制寄存器中的 ARE 位（自动重载预装载使能位）。

定时器计数器计数到 TMRx_CPR 计数周期寄存器设定的值，且控制寄存器 TMRx_CR0 中的 UDIS 位为 0 时，将产生更新事件。

更新事件相关的详细介绍，请参看“28.4.6.1 更新事件”章节。

定时器的计数器由时钟源经过预分频器分频后提供的时钟驱动，且仅当 TMRx_CR0 控寄存器中的 CEN 位(计数器使能)置 1 时，才会启动计数器计数。

注意：计数器将在 TMRx_CR 控制寄存器中的位 CEN(计数器使能)置 1 时刻后的一个时钟周期开始计数。

预分频器说明

预分频器用于对计数器的计数时钟频率进行分频，分频系数介于 1 和 65536 之间。通过配置预分频器寄存器(TMRx_PSCR)来设定预分频的分频系数，计数器计数频率的计算公式如下：

$$f_{CLK\_CNT} = f_{CLK\_SRC} / (PSC + 1)$$

该寄存器具有预加载缓存功能，当启动预加载功能后，可以对预分频器进行实时修改，而新的预分频系数将在下一个更新事件发生时被采用。

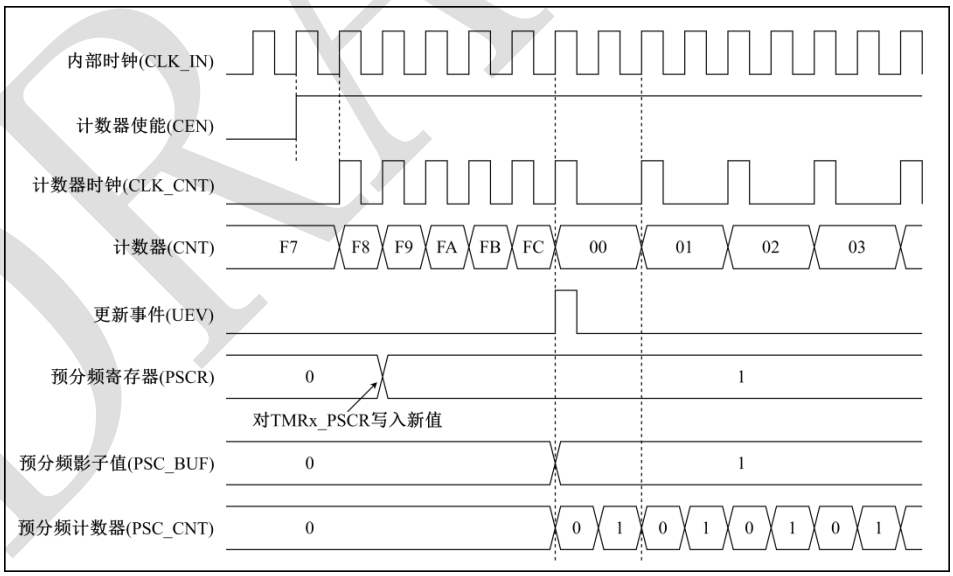


图 28-3　实时修改预分频器的分频值从 1 变为 2 的时序图

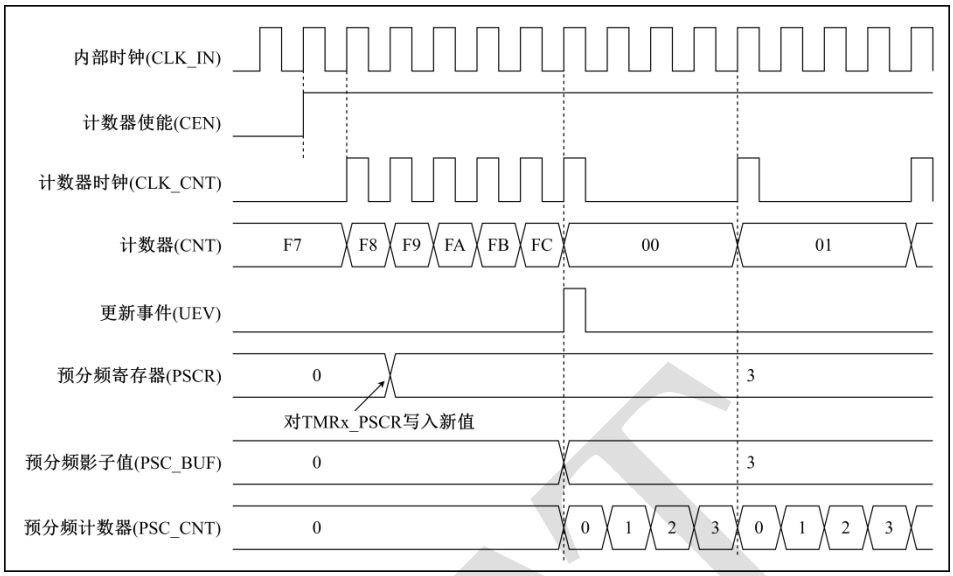


图 28-4　实时修改预分频器的分频值从 1 变为 4 的时序图

28.4.4　计数器模式

基础定时器仅支持向上递增计数模式，计数器从 0 开始计数到计数周期值(TMRx_CPR)，支持单次和连续计数模式：单次模式下，计数完成后将自动关闭计数器使能(CEN 自动置 0)，即停止计数；连续模式下，每次计数完成后，自动从 0 重新开始计数，并生成计数器更新事件 UEV。

每次发生计数器上溢时会生成更新事件，或将 TMRx_UGR 寄存器中的 UG 位置 1（通过软件）也可以生成更新事件。

定时器发生更新事件(UEV)时，将更新相关寄存器的值，并且将更新中断标志位(UIF)置 1：

- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图 28-5 ~图 28-10 将通过一些示例说明当计数器等于 0x36 时，不同时钟频率下表现的行为。

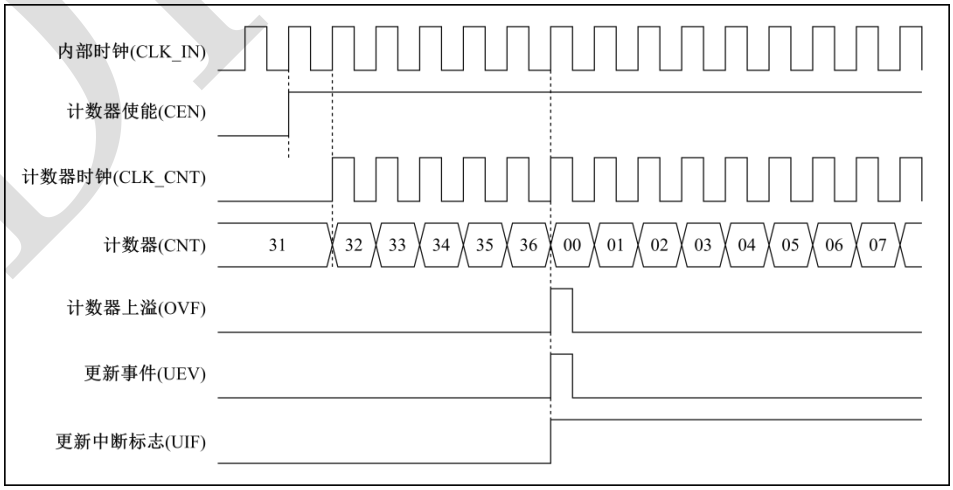


图 28-5　计数器时序图，1 分频内部时钟

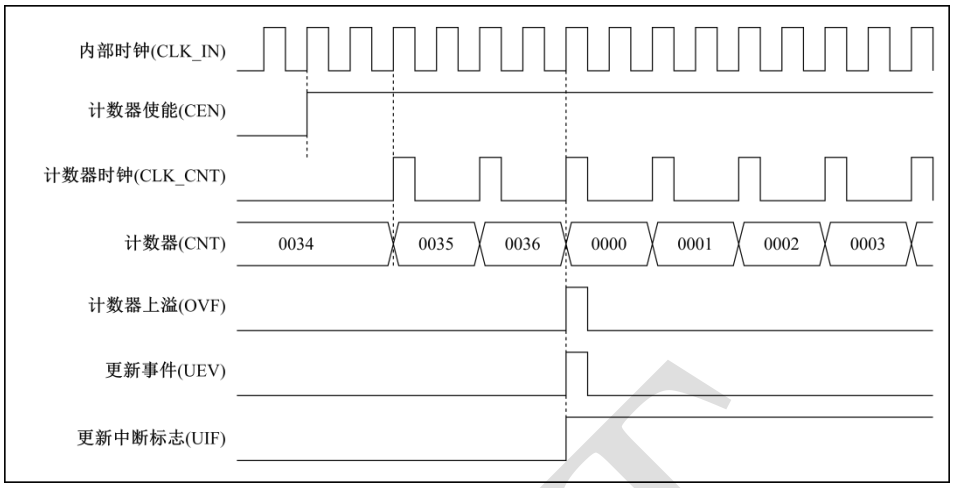


图 28-6　计数器时序图，2 分频内部时钟

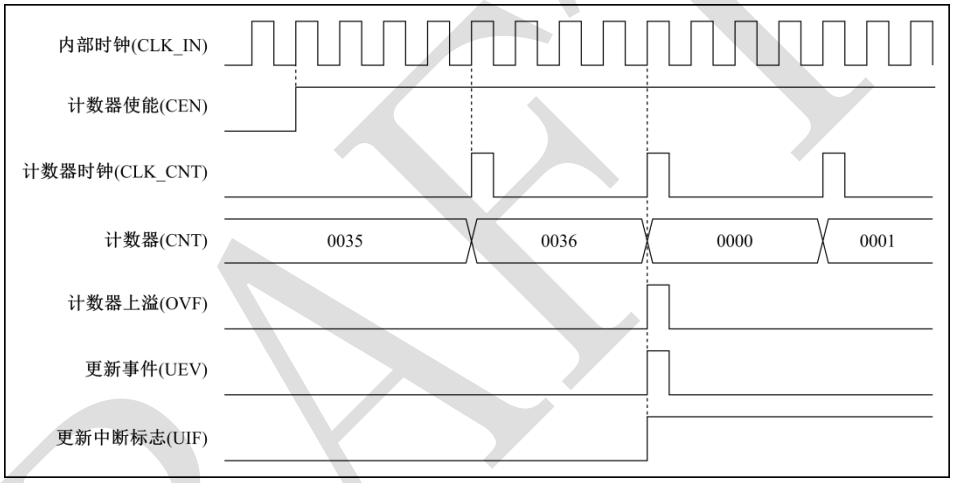


图 28-7　计数器时序图，4 分频内部时钟

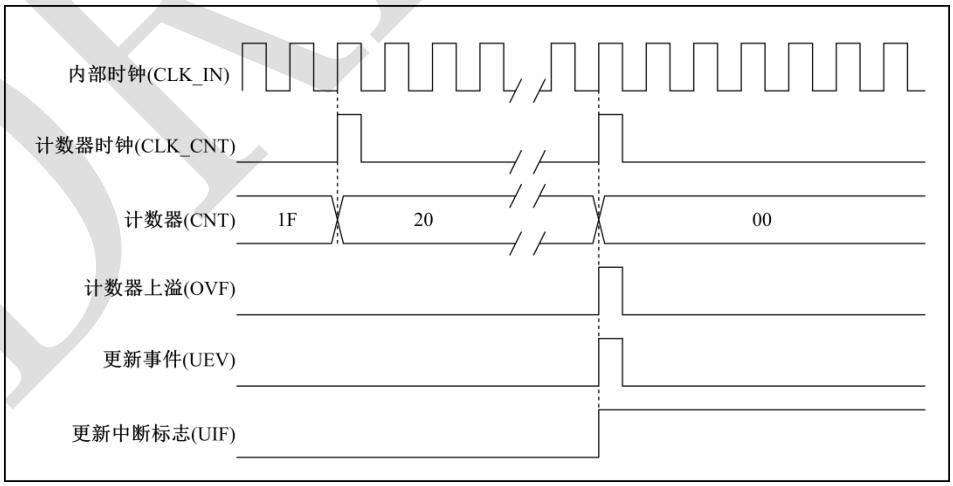


图 28-8　计数器时序图，N 分频内部时钟

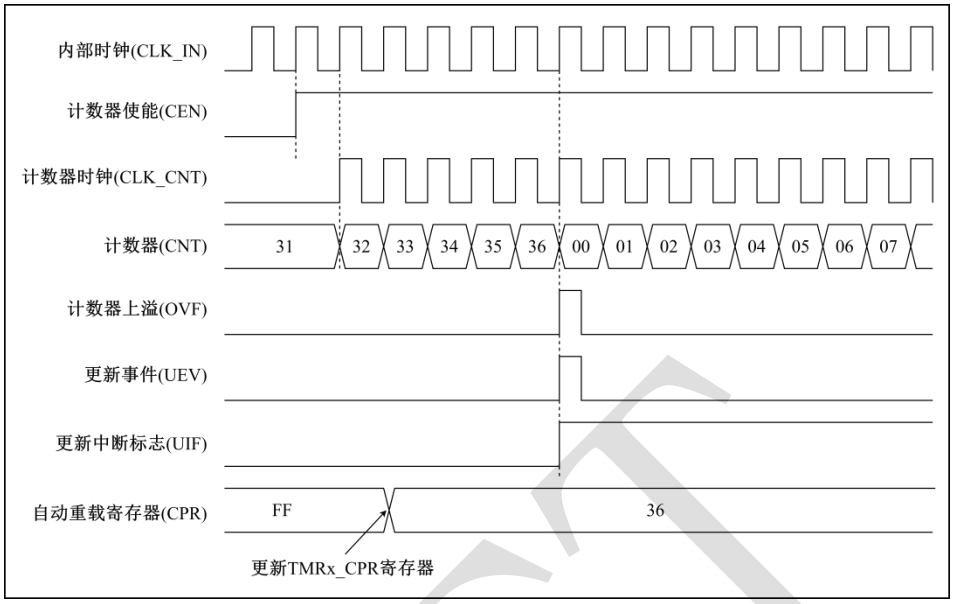


图 28-9　计数器时序图，运行中更新重载值（自动重载使能关闭-实时生效）

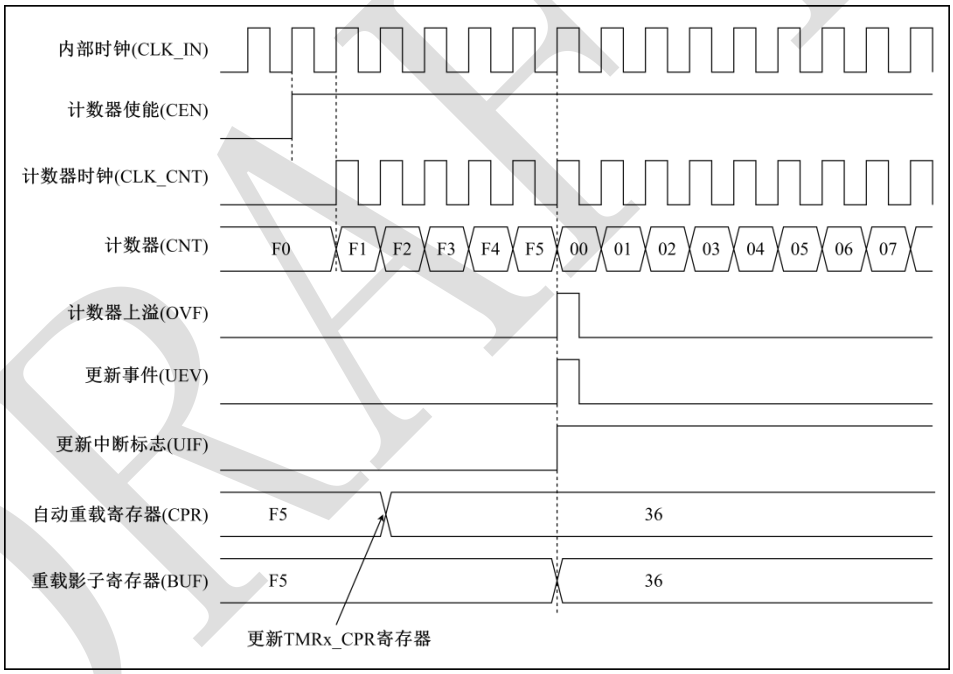


图 28-10　计数器时序图，运行中更新重载值（自动重载使能使能-更新事件生效）

28.4.5　调试模式

当内核进入调试模式（Cortex™-M4 内核停止）时，定时器内部计数器会根据 SYSCTRL 模块中 SYSCTRL_SYSDCR 系统调试配置寄存器中的 TMRxDEN 位来使能/关闭调试模式，可以选择继续工作或者停止工作。

28.4.6　事件与中断

基础定时器在计数完成时产生更新事件，事件主要用于更新相应的影子寄存器以及产生相应

的中断。

基础定时器内只有以下一种事件：

- 定时器上溢事件，简称上溢事件 OVEV(Overflow Event)

中断包含以下一种中断：

- 定时器上溢中断，发生上溢事件后，上溢中断标志位(OVIF)同时被置位

28.4.6.1 更新事件（Update Event）

更新事件(UEV)由以下情况产生：

- 计数器寄存器(TMRx_CNTR)的计数值到达计数周期寄存器(TMRx_CPR)设定的值。

更新事件的作用

计数器到达上溢后，计数器寄存器会立即重新加载计数值起始寄存器(TMRx_CSVR)的值，重新开始计数(如果为连续模式)。

事件产生后，中断状态寄存器(TMRx_ISR)中的上溢中断标志位(OVIF)将会置 1，并且会触发定时器中断(如果 OVIE = 1)。

28.4.6.2 中断

中断标志位与中断使能之间的关系如下表描述所示，详细内容请参考“28.4.6 事件与中断”章节描述：

表 28-2 中断标志位与中断使能之间的关系

中断名称	中断使能	中断标志	清除方式	备注
上溢中断	OVIE	OVIF	W1C	计数器计数上溢后产生

28.5 寄存器定义

28.5.1 寄存器列表

TMRx 基地址：

TMR7(0x40008000) TMR8(0x40009000)

偏移	实例地址	名称	默认值	描述
0x00	TMRx 基地址+0x00	TMRx_CR0	0x00000000	TMRx 控制寄存器 0
0x04	TMRx 基地址+0x04	TMRx_CR1	0x00000000	TMRx 控制寄存器 1
0x0C	TMRx 基地址+0x0C	TMRx_IER	0x00000000	TMRx 中断使能寄存器
0x10	TMRx 基地址+0x10	TMRx_SR	0x00000000	TMRx 状态寄存器
0x14	TMRx 基地址+0x14	TMRx_UGR	0x00000000	TMRx 更新事件寄存器
0x2C	TMRx 基地址+0x2C	TMRx_TCR	0x00000007	TMRx 脉冲控制寄存器
0x30	TMRx 基地址+0x30	TMRx_CPR	0x0000FFFF	TMRx 计数周期寄存器
0x38	TMRx 基地址+0x38	TMRx_PSCR	0x00000000	TMRx 预分频寄存器
0x3C	TMRx 基地址+0x3C	TMRx_CNTR	0x00000000	TMRx 计数寄存器

28.5.2 寄存器描述

28.5.2.1 TMRx 控制寄存器 0（TMRx_CR0）

- 名称：TMRx Control Register0
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ARE				OPM	URS	UDIS	CEN
								R/W				R/W	R/W	R/W	R/W
								0x0				0x0	0x0	0x0	0x0

字段	说明
[7] ARE	**自动重载使能(AutoReload Enable)** AutoReload 功能使能，是否开启影子寄存器加载功能； 0: 关闭重载功能 1: 开启重载功能 Note： 1. 预加载功能涉及的寄存器有：计数终止值寄存器、预分频寄存器； 2. 如果不开启预加载功能，则相关寄存器写入新值将立刻生效； 3. 如果开启预加载功能，则相关寄存器写入新值后，在下一个更新事件（UEV）到来后生效。
[3] OPM	**单脉冲模式(One Pulse Mode)** 0: 计数器在发生更新事件时不会停止计数； 1: 计数器在发生下一更新事件时停止计数(将 CEN 位清零)； Note：配置为单次模式后，计数器溢出后，将自动停止计数。
[2] URS	**更新事件源(Update Request Source)** 此位由软件置 1 和清零，用以选择更新事件源。 0: 当配置为 0 时，所有以下事件都会生成更新事件: — 计数器上溢 — 将 UG 位置 1 1: 当配置为 1 时，只有计数器上溢会生成更新事件。
[1] UDIS	**更新禁止位(Update Disable)** 此位由软件置 1 和清零，用以使能/禁止更新事件的生成； 0:更新使能，更新(UEV)事件可通过以下事件之一生成: — 计数器上溢 — 将 UG 位置 1 1:更新禁止 Note: 更新禁止后，不会生成更新事件(不会产生更新中断)，各影子寄存器的值保持不变，如果 UG 位置 1，则会重新初始化计数器和预分频器。
[0] CEN	**计数器使能位(Counter Enable):** 1:开始计数器 0:关闭计数器 Note: 单次模式下，当发生更新事件时会自动关闭 CEN；

28.5.2.2 TMRx 控制寄存器 1（TMRx_CR1）

- 名称：TMRx Control Register1
- 偏移地址：0x04
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														MMS	
														R/W	
														0x0	

字段	说明
[1:0] MMS	**主模式选择(Master Mode Selection):** 这些位用于选择主模式下将要发送到从定时器的同步的信息 (TRGO)，组合描述如下： 00: 复位 - 寄存器(UGR)中的 UG 位用作触发输出，如果从模式控制器配置为复位模式，则 TRGO 上的信号用于从定时器的复位操作； 01: 使能 - 计数器使能信号用作触发输出(TRGO)，该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器； 10: 更新 - 选择更新事件作为触发输出(TRGO)，例如：主定时器可用作从定时器的预分频器；

28.5.2.3 TMRx 中断使能寄存器（TMRx_IER）

- 名称：TMRx Interrupt Enable Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						OVIE	UIE								
						R/W	R/W								
						0x0	0x0								

字段	说明
[9] OVIE	**计数上溢中断使能(Overflow Interrupt Enable)** 1：使能中断 0：禁止中断
[8] UIE	**更新中断使能(Update Interrupt Enable)** 1：使能中断 0：禁止中断

28.5.2.4 TMRx 状态寄存器（TMRx_SR）

- 名称：TMRx Status Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						OVIF	UIF								
						R/W1C	R/W1C								
						0x0	0x0								

字段	说明
[9] OVIF	**计数器上溢中断标志位(Counter Overflow Interrupt Flag)** 0: 未发生上溢 1: 计数器上溢
[8] UIF	**更新中断标志(Update Interrupt Flag)** 该位在发生更新事件时通过硬件置 1,但需要通过软件清零。 0: 未发生更新 1: 更新中断挂起 该位在以下情况下更新寄存器时由硬件置 1: — 计数器发生上溢且当 CR 寄存器中 UDIS="0"时； — 当 CR 寄存器中 URS="0"且 UDIS="0"，通过软件使用 UGR 寄存器中的 UG 位重新初始化 Counter 时；

28.5.2.5 TMRx 更新事件寄存器（TMRx_UGR）

- 名称：TMRx Update Generation Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															UG
															R/W1C
															0x0

字段	说明
[0] UG	**更新事件生成(Update Generation)** 此位由软件置 1 生成事件，并由硬件自动清零; 0: 不执行任何操作 1: 重新初始化计数器并生成寄存器更新事件，预分频器计数器也将清零(但预分频比不受影响)，计数器清零

28.5.2.6 TMRx 脉冲控制寄存器（TMRx_TCR）

- 名称：TMRx Trigger Control Register
- 偏移地址：0x2C
- 默认值：0x00000007
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TOW			
												R/W			
												0x7			

字段	说明
[3:0] TOW	**TRGO 脉冲宽度控制(Trigger Output Width)** 0: 1 个 Cycle 1: 2 个 Cycle …… Note： 1. TRGO 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle； 2. TRGO 脉冲宽度必须小于 Timer 的 Period 值(CPV).

28.5.2.7 TMRx 计数周期寄存器（TMRx_CPR）

- 名称：TMRx Counter Period Register
- 偏移地址：0x30
- 默认值：0x0000FFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CPV							
								R/W							
								0xFFFF							

字段	说明
[15:0] CPV	**计数周期值(Counter Period Value)** 通过配置此寄存器设置定时器结束计数的值。此外，该寄存器支持自动预装载（Auto-Reload）功能： 1. 不使能自动预装载功能（ARE=0） 对该寄存器的设置将立即作用到内部影子寄存器内，本次计数周期内即生效； 2. 使能自动预装载功能（ARE=1） 对该寄存器的设置将在更新事件（UEV）产生后，再更新至内部影子寄存器内生效。

28.5.2.8 TMRx 预分频寄存器（TMRx_PSCR）

- 名称：TMRx Prescaler Register
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC															
R/W															
0x0															

字段	说明
[15:0] PSC	**预分频寄存器（Prescaler Value)** 该寄存器支持自动预装载(Auto-Reload)功能，每次发生更新事件时会装载到实际预分频器寄存器的值； Note：计数器时钟频率 CLK_CNT 等于 FCLK/(PSC+1)，其中 PSC 范围为 0~65535： 0:不分频，系统时钟 1:2 分频 2:3 分频 ……

28.5.2.9 TMRx 计数寄存器（TMRx_CNTR）

- 名称：TMRx Counter Register
- 偏移地址：0x3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															
0x0															

字段	说明
[15:0] CNT	**定时器计数值(Counter Value)** 通过该寄存器可对定时器计数器的值进行读取/写入。 需注意，因为系统与外设之间存在一定的总线延迟，对该寄存器的读取/写入操作可能存在一定的偏差。

29 通用定时器（TMR0/TMR1/TMR2）

29.1 简介

TMR0/TMR1/TMR2 定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

通用定时器可用于多种用途，包括最基本的定时功能（基础计时）以及测量输入信号的脉冲宽度（输入捕获）、生成 PWM 输出波形（输出比较）等多种灵活配置的功能。

使用定时器预分频器和 RCU 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

TMR0/TMR1/TMR2 定时器完全彼此独立，不共享任何资源，它们可以同步操作。

29.2 主要特性

TMR0/1/2 主要具有以下特性：

- 16 位的自动重载向上计数器
- 16 位的预分频器，用于对计数器的时钟源进行预分频，分频系数介于 1 和 65536 之间
- 16 位的计数比较值和周期值寄存器
- 两种工作模式：
  - 循环模式，计数完成后自动重载并继续计数
  - 单次模式，计数完成后自动关闭定时器的使能
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿模式）
  - 单脉冲模式输出
- 带可编程死区时间的互补输出
- 支持主从模式，可以实现多个定时器的互连
- 支持断路保护功能，在发生异常时将 PWM 输出置于复位或已知状态
- 发生如下事件时产生中断：
  - 计数器上溢(Overflow Event)
  - 更新事件(Update Event) (需使能)：计数器上溢产生更新事件，也可软件产生(UG)
  - 输入捕获(Capture Event)、重复捕获(Over Capture Event)
  - 输出比较(Compare Event)
  - 断路输入(Break Event)

29.3 结构框图

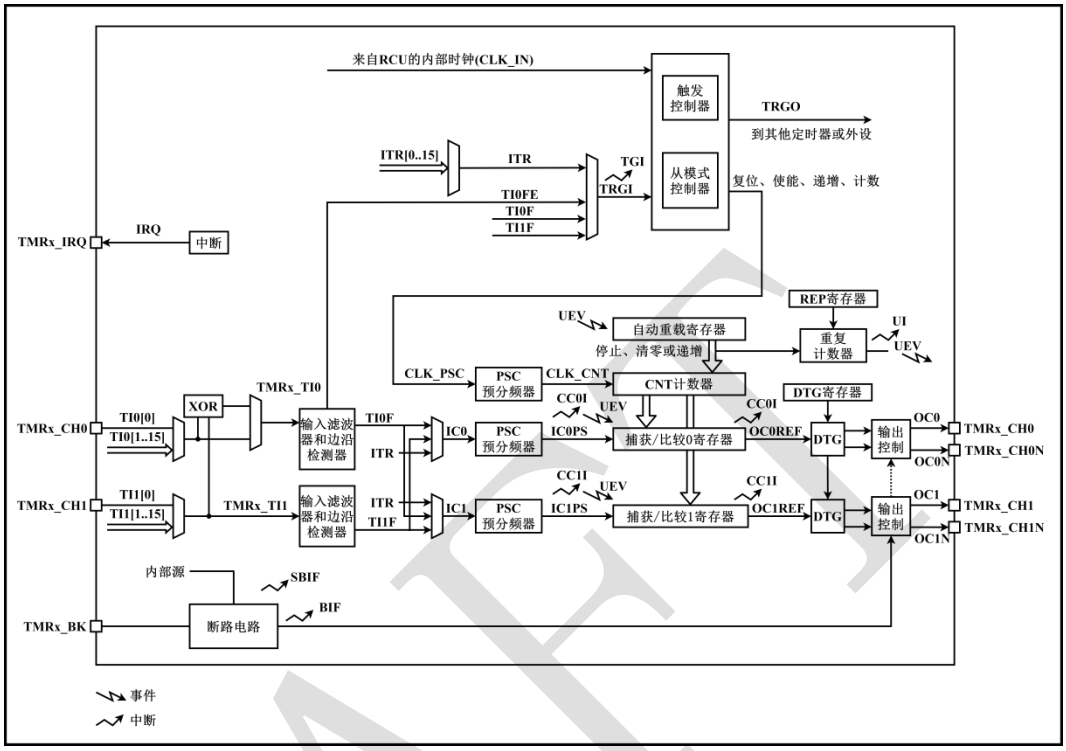


图 29-1　TMR 结构框图

29.4 功能描述

29.4.1 内部信号

表 29-1 列出了 TMRx 定时器主要的输入/输出信号及简要描述：

表 29-1 TMRx 内部信号

Signal Name	Signal Type	Description
PCLK	Input	TMRx APB Clock
TMRx_CH0/1	Input/Output	TMRx 多功能复用通道，可以用作 Compare、Capture 及 PWM，还可以用于外部时钟或 Trigger 触发输入。
TMRx_BK	Input	TMRx Break 输入信号
TRGO	Output	TMRx 内部触发输出信号，可用于触发其他外设
IRQ	Output	TMRx 全局中断，包括 Capture/Compare/Update/Break 中断请求。

表 29-2 列出了 TMRx 定时器 TMRx_TI 输入信号的连接关系：

表 29-2 TMRx_TI 信号内部互连

TMRx_TIx 输入	Sources					
	TMR0		TMR1		TMR2	
	事件名称(CH0)	事件名称(CH1)	事件名称(CH0)	事件名称(CH1)	事件名称(CH0)	事件名称(CH1)
TMRx_TI_0	CMP0_OUT	HSE	CMP0_OUT	HSE	CMP0_OUT	HSE

TMRx_TI_1	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP1_OUT
TMRx_TI_2	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT
TMRx_TI_3	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT
TMRx_TI_4	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT
TMRx_TI_5	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT
TMRx_TI_6	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT
TMRx_TI_7	CMP7_OUT	CMP7_OUT	CMP7_OUT	CMP7_OUT	CMP7_OUT	CMP7_OUT
TMRx_TI_8	HSI	CMP8_OUT	HSI	CMP8_OUT	HSI	CMP8_OUT
TMRx_TI_9	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
TMRx_TI_10	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
TMRx_TI_11	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
TMRx_TI_12	TMR0_CH0	TMR0_CH0	TMR1_CH0	TMR1_CH0	TMR2_CH0	TMR2_CH0
TMRx_TI_13	TMR0_CH1	TMR0_CH1	TMR1_CH1	TMR1_CH1	TMR2_CH1	TMR2_CH1
TMRx_TI_14	TMR0_CH0N	TMR0_CH0N	TMR1_CH0N	TMR1_CH0N	TMR2_CH0N	TMR2_CH0N
TMRx_TI_15	TMR0_CH1N	TMR0_CH1N	TMR1_CH1N	TMR1_CH1N	TMR2_CH1N	TMR2_CH1N

表 29-3 列出了 TMRx 定时器内部 ITR 输入信号的连接关系：

表 29-3 TMRx_ITR 信号内部互连

TMRx_ITRx 输入	**Sources**		
	TMR0	**TMR1**	**TMR2**
TMRx_ITR_0	TMR7_TRGO	TMR7_TRGO	TMR7_TRGO
TMRx_ITR_1	TMR8_TRGO	TMR8_TRGO	TMR8_TRGO
TMRx_ITR_2	TMR1_TRGO	TMR0_TRGO	TMR0_TRGO
TMRx_ITR_3	TMR2_TRGO	TMR2_TRGO	TMR1_TRGO
TMRx_ITR_4	TMR3_TRGO	TMR3_TRGO	TMR3_TRGO
TMRx_ITR_5	TMR4_TRGO	TMR4_TRGO	TMR4_TRGO
TMRx_ITR_6	TMR1_OC0	TMR0_OC0	TMR0_OC0
TMRx_ITR_7	TMR2_OC0	TMR2_OC0	TMR1_OC0
TMRx_ITR_8	TMR3_OC0	TMR3_OC0	TMR3_OC0
TMRx_ITR_9	TMR4_OC0	TMR4_OC0	TMR4_OC0
TMRx_ITR_10	TMR9_TRGO	TMR9_TRGO	TMR9_TRGO
TMRx_ITR_11	TMR9_OC0	TMR9_OC0	TMR9_OC0
TMRx_ITR_12	TMR9_OC1	TMR9_OC1	TMR9_OC1
TMRx_ITR_13	TMR10_TRGO	TMR10_TRGO	TMR10_TRGO
TMRx_ITR_14	TMR10_OC0	TMR10_OC0	TMR10_OC0
TMRx_ITR_15	HRPWM_SYNCO	HRPWM_SYNCO	HRPWM_SYNCO

表 29-4 列出了 TMRx 定时器内部 Break 输入信号的连接关系：

表 29-4 TMRx Break 信号内部互连

TMRx_BKx 输入	**Sources**		
	TMR0	**TMR1**	**TMR2**
TMRx_BK_0	CMP0_OUT	CMP0_OUT	CMP0_OUT
TMRx_BK_1	CMP1_OUT	CMP1_OUT	CMP1_OUT
TMRx_BK_2	CMP2_OUT	CMP2_OUT	CMP2_OUT

TMRx_BK_3	CMP3_OUT	CMP3_OUT	CMP3_OUT
TMRx_BK_4	CMP4_OUT	CMP4_OUT	CMP4_OUT
TMRx_BK_5	CMP5_OUT	CMP5_OUT	CMP5_OUT
TMRx_BK_6	CMP6_OUT	CMP6_OUT	CMP6_OUT
TMRx_BK_7	CMP7_OUT	CMP7_OUT	CMP7_OUT
TMRx_BK_8	CMP8_OUT	CMP8_OUT	CMP8_OUT
TMRx_BK_9	TMR0_BK	TMR1_BK	TMR2_BK
TMRx_BK_10	ADC0_AWD0	ADC0_AWD0	ADC0_AWD0
TMRx_BK_11	ADC1_AWD0	ADC1_AWD0	ADC1_AWD0
TMRx_BK_12	PDM0_CMPH	PDM0_CMPH	PDM0_CMPH
TMRx_BK_13	PDM1_CMPH	PDM1_CMPH	PDM1_CMPH
TMRx_BK_14	PDM2_CMPH	PDM2_CMPH	PDM2_CMPH
TMRx_BK_15	PDM3_CMPH	PDM3_CMPH	PDM3_CMPH

表 29-5 列出了 TMRx 定时器内部系统 Break 输入信号的连接关系：

表 29-5 TMRx 系统 Break 信号内部互连

TMRx_SBK	TMR0/1/2
TMRx_SBK_0	Cortex™-M4 With FPU LOCKUP
TMRx_SBK_1	Low Voltage Detector (LVD)
TMRx_SBK_2	FLASH ECC Error
TMRx_SBK_3	QEI0/1/2 Error
TMRx_SBK_4	Clock Security System（CSS）

注意：TMRx 系统 Break 输入信号由以上 5 个源或起来得到，没有优先级或选择控制。

29.4.2 时基单元

通用定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成，计数器采用递增计数的方式，计数器的时钟可通过预分频器进行分频。

通用定时器的自动重载计数器(TMRx_CNTR)、计数周期寄存器(TMRx_CPR) 、重复计数寄存器(TMRx_CRR)和预分频寄存器(TMRx_PSCR)可通过软件进行读写，即使在计数器运行中也可执行读写操作。

时基单元包含：

- 自动重载计数器(TMRx_CNTR)
- 预分频寄存器(TMRx_PSCR)
- 计数周期寄存器(TMRx_CPR)
- 重复计数寄存器(TMRx_CRR)

其中预分频寄存器(TMRx_PSCR)、计数周期寄存器(TMRx_CPR)、重复计数寄存器(TMRx_CPR)都是可预装载的。对预装载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器立即生效，也可以在每次发生更新事件时更新到影子寄存器，这取决于定时器 TMRx_CR0 控制寄存器中的 ARE 位（自动重载预装载使能位）。

定时器计数器计数到 TMRx_CPR 计数周期寄存器设定的值，且控制寄存器 TMRx_CR0 中的 UDIS 位为 0 时，将产生更新事件。此外，该更新事件也可通过软件发起，通过对定时器事件生成寄存器 TMRx_UGR 中的 UG 位置 1。

更新事件相关的详细介绍，请参看“12.4.9.2 更新事件”章节。

定时器的计数器由时钟源经过预分频器分频后提供的时钟驱动，且仅当 TMRx_CR0 控寄存器中的 CEN 位(计数器使能)置 1 时，才会启动计数器计数。

注意：计数器将在 TMRx_CR0 控制寄存器中的位 CEN(计数器使能)置 1 时刻后的一个时钟周期开始计数。

预分频器说明

预分频器用于对计数器的计数时钟频率进行分频，分频系数介于 1 和 65536 之间。通过配置预分频器寄存器(TMRx_PSCR)来设定预分频的分频系数，计数器计数频率的计算公式如下：

$$f_{CLK\_CNT} = f_{CLK\_SRC} / (PSC + 1)$$

该寄存器具有预加载缓存功能，当启动预加载功能后，可以对预分频器进行实时修改，而新的预分频系数将在下一个更新事件发生时被采用，也可通过软件设置 TMRx_UGR 寄存器中的 UG 位，立即产生一个更新事件（UEV）。

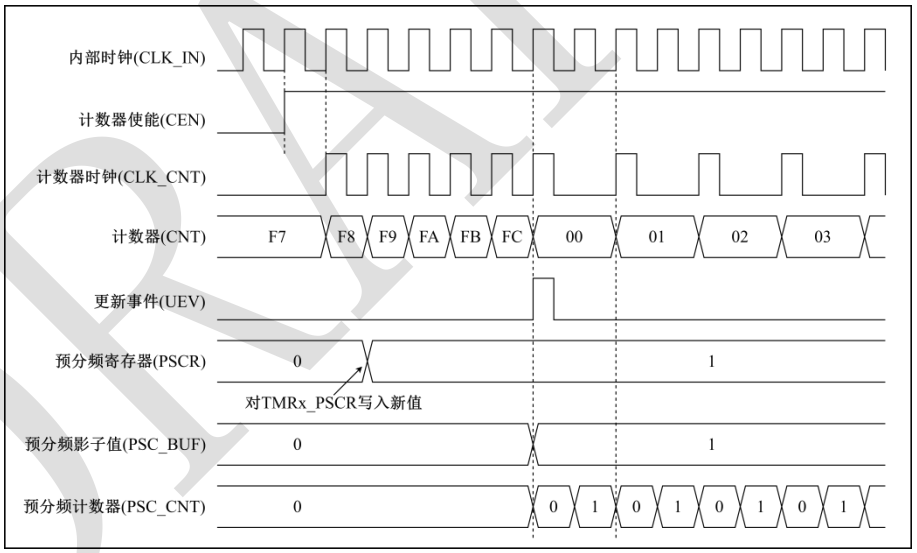


图 29-2　实时修改预分频器的分频值从 1 变为 2 的时序图

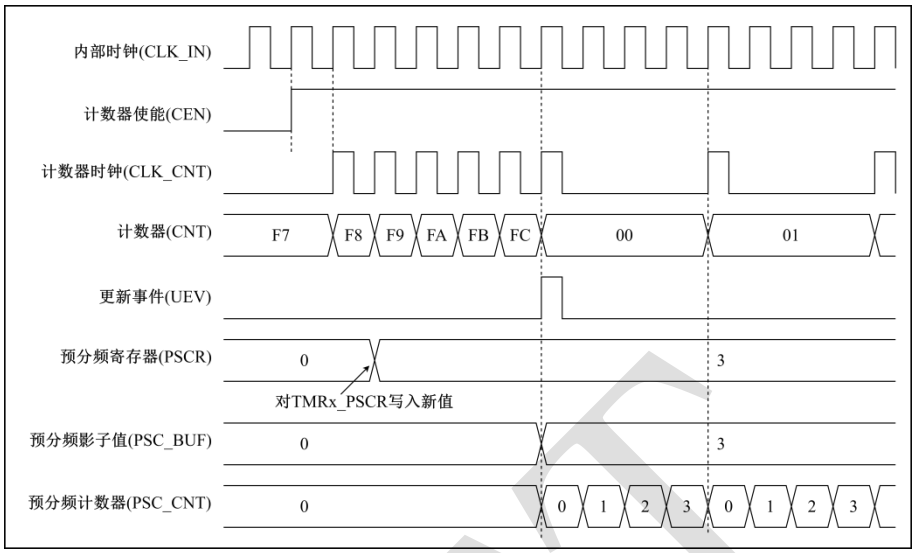


图 29-3　实时修改预分频器的分频值从 1 变为 4 的时序图

29.4.3　计数器模式

递增计数模式

通用定时器仅支持向上递增计数模式，计数器从 0 开始计数到计数周期值(TMRx_CPR)，支持单次和连续计数模式：单次模式下，计数完成后将自动关闭计数器使能(CEN 自动置 0)，即停止计数；连续模式下，每次计数完成后，自动从 0 重新开始计数，并生成计数器上溢事件 OVEV 和更新事件 UEV(如果使能更新事件)。

每次计数器发生上溢时会生成更新事件，或将 TMRx_UGR 寄存器中的 UG 位置 1（通过软件）也可以生成更新事件。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过计数器及预分频计数器仍然能够在上溢事件 OVEV 后，重新从 0x0 开始计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器等于 0x36 时，不同时钟频率下表现的行为。

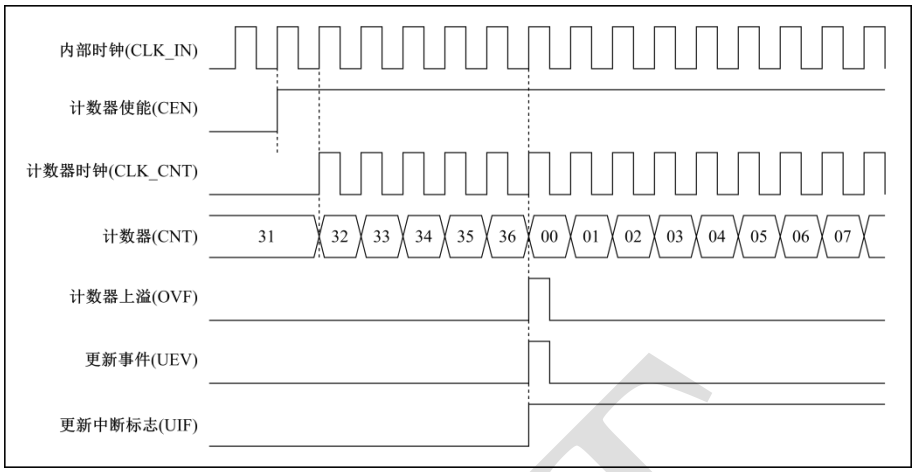


图 29-4　计数器时序图，1 分频内部时钟

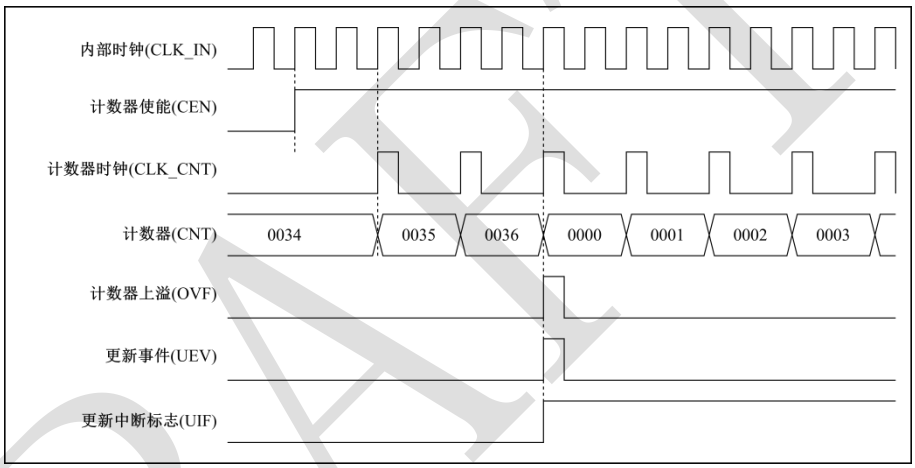


图 29-5　计数器时序图，2 分频内部时钟

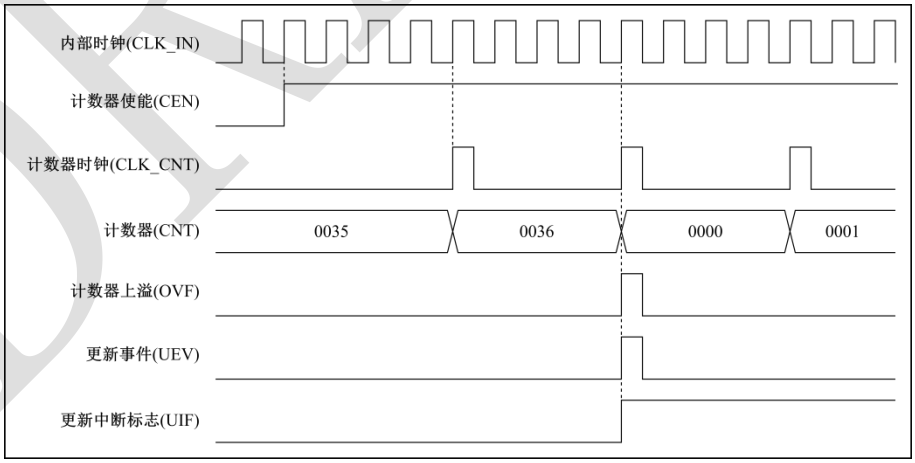


图 29-6　计数器时序图，4 分频内部时钟

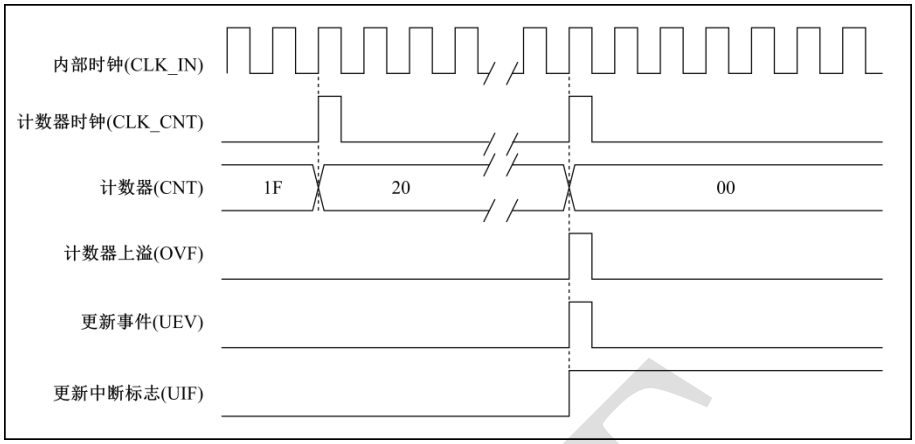


图 29-7　计数器时序图，N 分频内部时钟

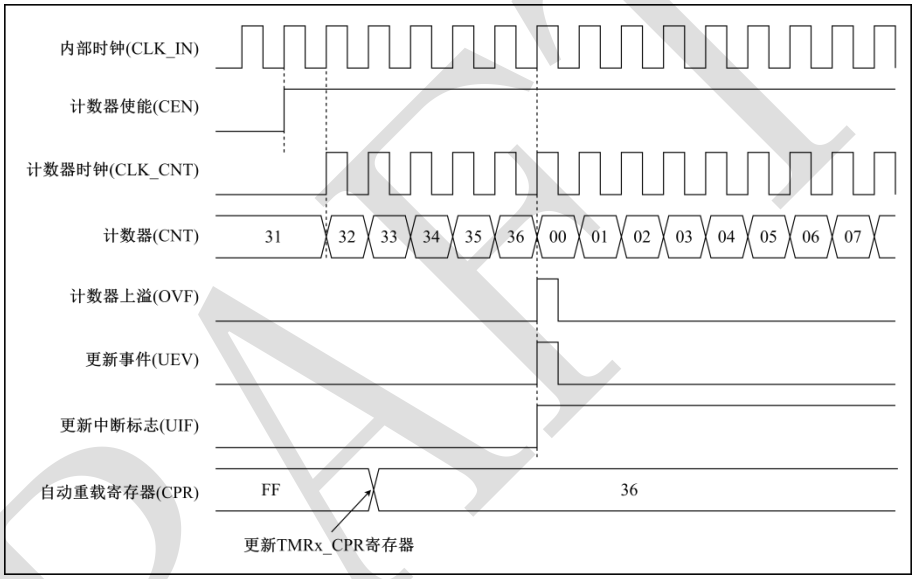


图 29-8　计数器时序图，运行中更新重载值（自动重载使能关闭-实时生效）

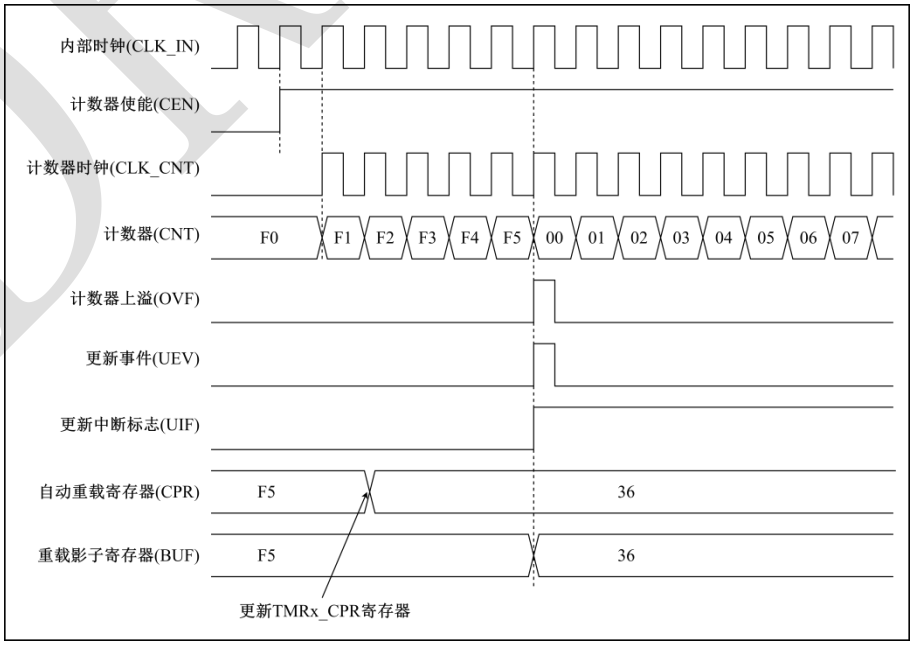


图 29-9　计数器时序图，运行中更新重载值（自动重载使能使能-更新事件生效）

29.4.4 重复计数器

在“1.4.2 章节：时基单元”中，介绍了计数器上溢时会生成更新事件（UEV）。当引入重复计数功能后，只有当重复计数器达到零时，才会生成更新事件（UEV），其对 PWM 信号的生成很有用。

这意味着，仅当定时器发生 N+1 个计数器上溢（N 为 TMRx_CRR 重复计数寄存器中的值）才生成一次更新事件（UEV），将触发重复计数影子寄存器（从 TMRx_CRR 加载）、计数周期影子寄存器（从 TMRx_CPR 加载）及预分频器影子寄存器（从 TMRx_PSCR 加载）的自动重载。

重复计数器是自动重载类型，其重复加载频率为 TMRx_CRR 寄存器中定义的值（请参见下图）。当更新事件由软件（通过将 TMRx_UGR 寄存器的 UG 位置 1）或硬件生成（通过从模式控制器）时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TMRx_CRR 寄存器的值。

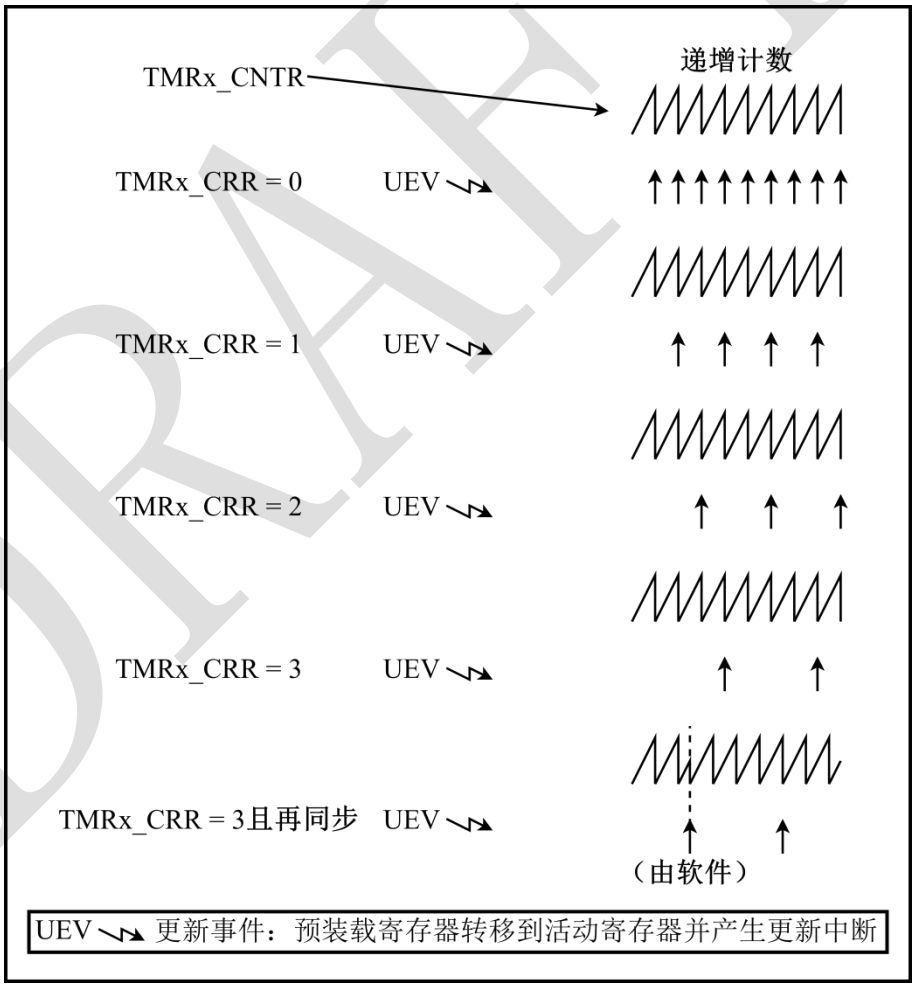


图 29-10　TMRx_CRR 重复计数器设置下的更新频率示例

29.4.5 时钟选择

定时器的计数器时钟源分为以下几类：

- 内部时钟源（CLK_IN）：系统时钟上升沿
- 外部时钟模式：通过外部输入引脚（TIx）
- 内部触发输入（ITRx）：连接其他定时器的触发输出

内部时钟源（CLK_IN）

内部时钟源是 TMRx 定时器的默认时钟源。

将 TMRx_SCR 寄存器中的 SMS 位置 0x0，则禁止从模式控制器，定时器默认选择内部时钟源作为计数时钟。然后，将 TMRx_CR0 寄存器中的 CEN 位和 TMRx_UGR 寄存器中的 UG 位用作控制位，当对 CEN 位写 1 时，预分频器的时钟就由内部时钟 CLK_IN 提供；通过软件对 UG 位写 1，则会使预分频器清零并重新开始计数。

下图为正常模式下内部控制逻辑及计数器计数的行为（预分频设置为 0，即在没有预分频情况下）。

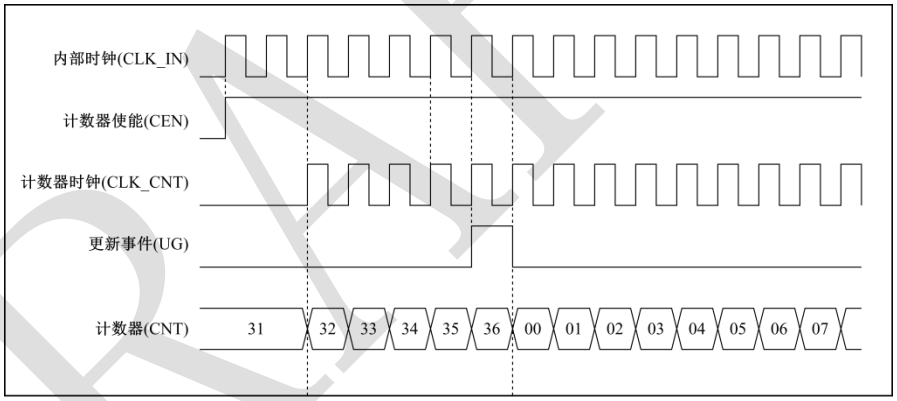


图 29-11　正常模式下的控制时序图（没有预分频情况）

外部时钟源模式

当 TMRx_SCR 寄存器中的 SMS 位为“111”时，则选择外部时钟源模式。在该模式下，计数器将在选定输入信号的上升沿计数。

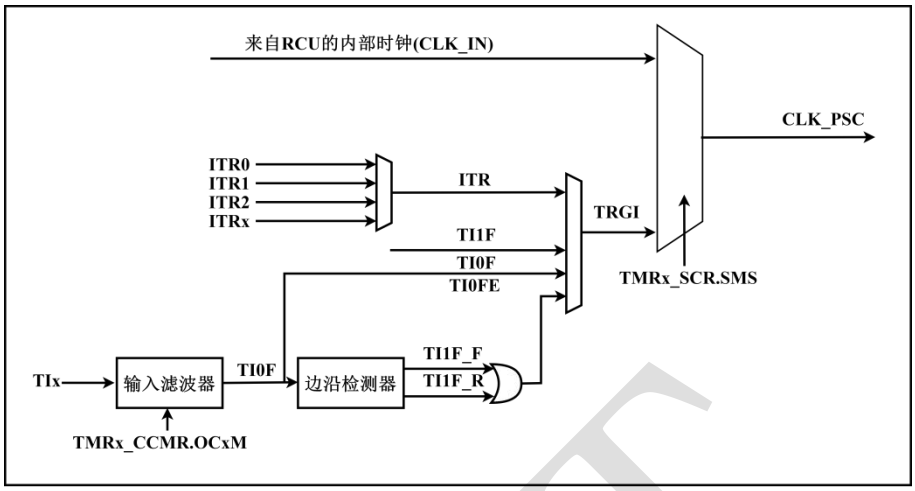


图 29-12　外部时钟连接示意图

定时器可选取外部时钟源作为计数器的时钟源，外部时钟源可来源于外部输入引脚或内部触发输入，外部输入引脚先经过内部滤波电路，然后通过内部边沿检测电路后作为计数器的触发时钟。

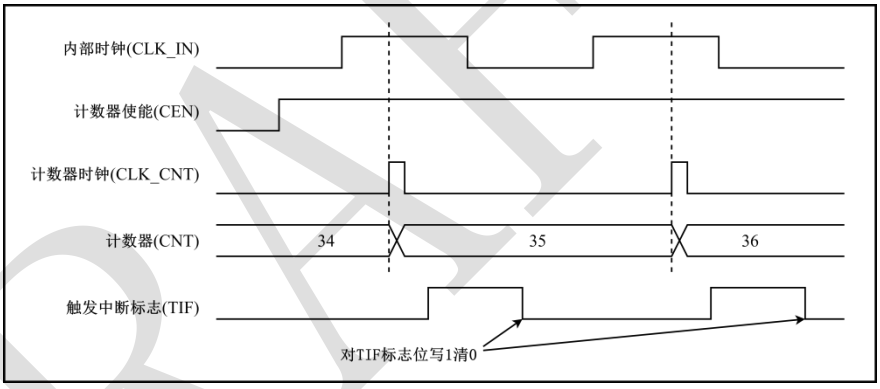


图 29-13　外部时钟模式下的控制电路

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 触发标志位置 1，TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入经过同步电路引起的。

29.4.6　捕获/比较通道

通用定时器的每个捕获/比较通道均基于一个捕获/比较寄存器(包括影子寄存器)、一个捕获输入(滤波器、极性边沿检测、多路复用和预分频器)和一个比较输出(比较器和输出控制)组成。

输入捕获是对相应的输入信号 TIx 进行采样，经过滤波器后输出 TIxF，然后，再经过极性选择、边沿检测生成一个触发信号（TIxFE），该信号作为从模式控制器的输入，也可用作捕获信号输入。该信号先经过预分频电路（ICxPS），然后进入到捕获寄存器，如下图。

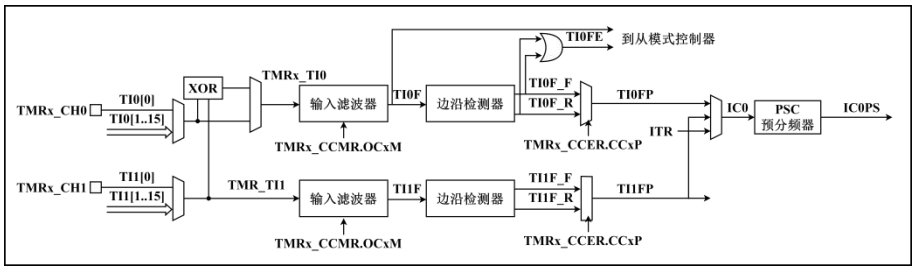


图 29-14　输入捕获通道示意图

输入捕获用于在捕获到指定边沿发生时，锁存计数器的准确数值。通过这种机制可实现对外部信号的准确捕获和测量，由此可实现对信号的脉宽测量、周期测量及占空比计算等功能。

输出比较是利用计数器计数，并与预先配置好的比较值进行对比，当计数值与比较值匹配时，根据所设定的比较输出模式进行相应的波形输出，从而实现例如 PWM、反转、强制极性输出等功能。

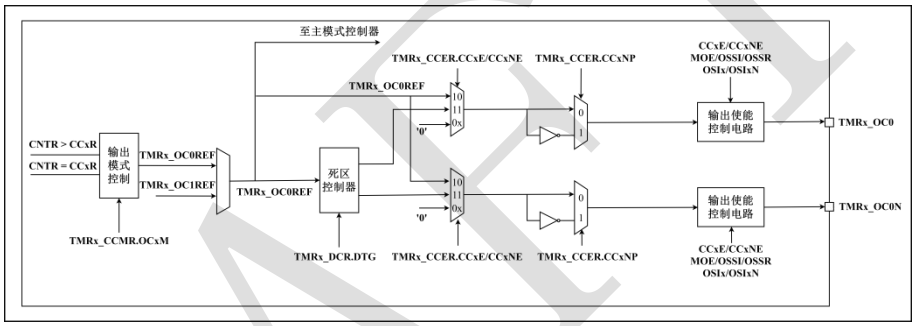


图 29-15　比较输出通道示意图

- 在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

29.4.7　输入捕获

在输入捕获模式下，当在对应的 ICx 信号上检测到边沿跳变后，会将计数器的值锁存到捕获/比较寄存器(TMRx_CCxR)中。当发生捕获事件时，会将 TMRx_SR 寄存器中相应的 CCxMIF 标志位置 1，并触发定时器中断(如果对应中断已使能)。如果在发生捕获事件时 CCxMIF 标志位已经被置位，则会将 TMRx_SR 寄存器中相应的 CCxOIF 重复捕获标志位置 1。可通过软件对捕获标志位 CCxMIF 及重复捕获标志位 CCxOIF 写 1 来清零，或读取存储在 TMRx_CCxR 寄存器中的捕获数据时。

通过配置 TMRx_CCMR 寄存器中的 CCxS 位来选择捕获输入通道，并将 TMRx_CCER 寄存器中的 CCxE 位置 1，就可以使定时器工作在输入捕获模式下。

通过配置 TMRx_CCER 寄存器中的 CCxP 位来选择捕获模式下的触发边沿，配置 TMRx_CIR 寄存器的 CxTIS 位可选择捕获的引脚输入源或比较器模块(CMPx_OUT)的内部输出信号。

当定时器工作于输入捕获模式下，在检测到捕获输入上产生边沿跳变时，定时器会将当前计数器的值锁存捕获到捕获/比较寄存器(TMRx_CCxR)内，同时将 TMRx_SR 寄存器中的输入捕

获中断标志位(CCxMIF)置 1，并触发定时器中断(如果对应中断已使能)，软件上需要及时清除输入捕获中断标志 (对 TMRx_SR 寄存器中的 CCxMIF 写 1，或读取 TMRx_CCxR 寄存器，硬件将自动清除标志位)。如果在清除该标志位之前，再次捕获到输入跳变沿，会将新的计数值锁存并覆盖之前的值，同时会将 TMRx_SR 寄存器中的输入重复捕获标志位(CCxOIF)置 1，并触发定时器中断(如果对应中断已使能)。因此在处理输入捕获时，建议在捕获状态标志位置起后应当及时读取数据，否则一旦发生重复捕获，新捕获的数据将覆盖之前的数据，导致捕获信息丢失。

29.4.8 比较输出

配置 TMRx_CCMR 寄存器中的 CCxS 位来选择比较输出模式，配置 TMRx_CCER 寄存器中的 CCxE 位来使能比较输出功能，使定时器工作在输出比较模式下。

配置 TMRx_CCMR 寄存器中的 OCxM 位来选择比较输出的工作模式，配置 TMRx_CCMR 寄存器中的 CCxPE 位来使能比较输出的预装载功能，配置 TMRx_CCER 寄存器中的 CCxP 位来选择比较模式下有效电平的极性。

当定时器的计数器(TMRx_CNTR)值与捕获/比较寄存器(TMRx_CCxR)值相匹配时，根据比较输出的工作模式在引脚上输出相应的波形，同时会将 TMRx_SR 寄存器中的 CCxMIF 位置 1，并触发定时器中断(如果对应中断已使能)，软件上可通过对 CCxMIF 位写 1 进行清除。

输出工作模式

定时器支持的比较输出工作模式有以下几种，通过 TMRx_CCMR 寄存器中的 OCxM 位进行配置：

- 0000：冻结，输出比较寄存器与计数器进行比较不会对输出造成任何影响（保持原有状态）
- 0001：当计数器与捕获/ 比较寄存器匹配时，输出有效电平（高电平）
- 0010：当计数器与捕获/ 比较寄存器匹配时，输出无效电平（低电平）
- 0011：当计数器与捕获/ 比较寄存器匹配时，发生翻转
- 0100：强制变为无效电平（低电平）
- 0101：强制变为有效电平（高电平）
- 0110：PWM 模式 1 - 当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态（高有效）
- 0111：PWM 模式 2 - 当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态（高有效）
- 1100：组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同，参考是 OC1REF 或 OC2REF 的结果
- 1101：组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同，参考是 OC1REF 和 OC2REF 的结果。

配合定时器中的单次/连续计数模式，还可实现更多组合功能，例如利用单次计数模式+输出比较 PWM 模式可实现单脉冲输出功能，更多特别模式说明将在后续章节进行更详细介绍。

自动装载功能

在比较输出模式中，软件可通过修改捕获/比较寄存器(TMRx_CCxR)来调整比较值，通过配置TMRx_CCMR 寄存器中的 OCxPE 位来使能比较预装载功能，自动装载的作用如下：

- 不开启预装载功能，向 TMRx_CCxR 寄存器写入新值，将立即更新到内部影子寄存器并生效。
- 开启预装载功能，向 TMRx_CCxR 寄存器写入新值，并不会立即生效，而是在下一个更新事件之后，再将 TMRx_CCxR 寄存器值更新到影子寄存器并生效。更详细的捕获/比较事件描述，请参考“29.4.14.5 捕获/比较事件”章节说明。

29.4.8.1 强制输出模式

在比较输出模式（TMRx_CCMR 寄存器中的 CCxS 位等于 0x0）下，可直接通过软件将比较输出信号（OCxREF）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的比较结果。

配置强制输出模式，只需将 TMRx_CCMR 寄存器中的 OCxM 位配置为 100 或 101，如对该位写入 101 后，OCxREF 将强制为高电平（OCxREF 始终为高电平有效），而输出 OCx 会根据 CCxP 极性位选择合适的有效电平。

在强制输出模式下，虽然定时器引脚输出固定电平，但定时器的计数器值与比较值的匹配行为依然在执行，在匹配时依然会引发相应的标志位并触发中断。

- 强制输出无效电平 (OCxM=100)时：
  - CCxP=0（有效电平为高电平），将比较强制输出为低电平
  - CCxP=1（有效电平为低电平），将比较强制输出为高电平
- 强制输出有效电平 (OCxM=101)时：
  - CCxP=0（有效电平为高电平），将比较强制输出为高电平
  - CCxP=1（有效电平为低电平），将比较强制输出为低电平

29.4.8.2 匹配输出模式

匹配输出模式可用于控制输出波形，或指示时间周期的变化。

输出引脚的电平由匹配输出模式（TMRx_CCMR 寄存器中的 OCxM 位）和输出极性（TMRx_CCER 寄存器中的 CCxP 位）定义。当定时器的计数器值与比较值相匹配时，输出比较匹配功能：

- 输出引脚即可保持原有电平（OCxM=0000）
- 配置为有效电平（OCxM=0001）
- 配置为无效电平（OCxM=0010）
- 配置为进行翻转（OCxM=0011）

匹配时，会将 TMRx_SR 中断状态寄存器中的 CCxMIF 置位，并触发定时器中断(如果对应中断已使能)。

通过配置 TMRx_CCMR 寄存器中的 CCxPE 位，可选择 TMRx_CCxR 寄存器的自动加载功能的开启。匹配输出模式也可用作为单脉冲输出（在单脉冲模式下）。

29.4.8.3 PWM 输出模式

比较输出的 PWM 模式，可以用于生成一个调制信号，该信号频率周期由 TMRx_CPR 周期值寄存器决定，而占空比则由捕获/比较寄存器(TMRx_CCxR)设定的值决定，比较输出的 PWM 模式有两种：

- PWM 模式 1（OCM=0110）
  当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态（高有效）。
- PWM 模式 2（OCM=0111）
  当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态（高有效）。

OCx 极性可通过 TMRx_CCER 寄存器的 CCxP 位来设置，既可以设为高电平有效，也可以设为低电平有效，OCx 输出通过将 TMRx_CCER 寄存器中的 CCxE 位置 1 来使能，有关其详细信息，请参考 TMRx_CCER 寄存器说明。

因为计数器采用的递增方式计数，所以定时器为边沿对齐模式下生成 PWM。

边沿对齐模式

下图以 PWM 模式 1 为例，只要 TMRx_CNTR 计数器值小于 TMRx_CCxR 比较值，PWM 参考信号 OCxREF 为高电平，否则为低电平。如果 TMRx_CCxR 比较值大于 TMRx_CPR 重载周期值，则 PWM 参考信号 OCxREF 为高电平。如果 TMRx_CCxR 比较值为 0，则 PWM 参考信号 OCxREF 保持低电平。下图为边沿对齐的 PWM 波形，其中 TMRx_CPR 值为 8。

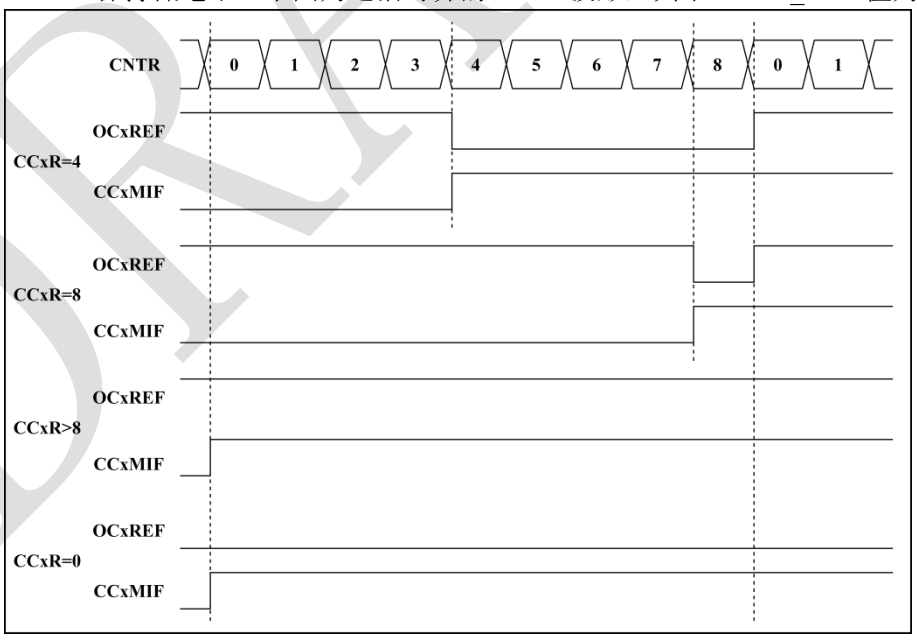


图 29-16　PWM 输出（CPR=8，CCxR=4，CCxP=0 高电平有效）

29.4.8.4 单脉冲模式

单脉冲模式(One-Pulse Mode)是单次计数模式+PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号下启动，并在一段可编程的延时后输出一个可编程宽度的单脉冲信号。

可通过从模式控制器启动计数器，在匹配输出模式或 PWM 模式下生成波形，将 TMRx_CR0 寄存器中的 OPM 位置 1，即可使能单脉冲模式。这样，在发生下一个更新事件 UEV 时，计数器将自动停止。

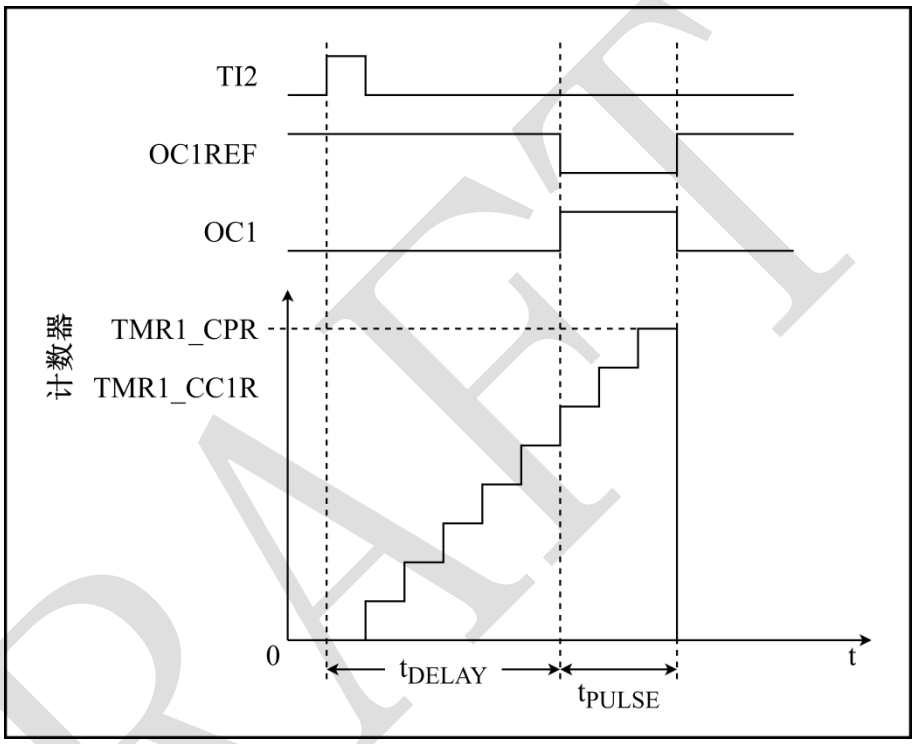


图 29-17 单脉冲模式示例

上图效果为：在 TI2 输入引脚上检测到上升沿时，经过 t_{DELAY} 的延迟后，将在 OC1 上输出一个长度为 t_{PULSE} 的正脉冲波形。

单脉冲的波形可通过调整 TMRx_CCxR 比较寄存器的值来定义（需考虑时钟频率及计数器预分频）：

- t_{DELAY} 时间由写入 TMRx_CCxR 的值来定义
- t_{PULSE} 时间由重载周期值与比较值（TMRx_CPR-TMRx_CCxR）之差来定义

当 TMRx_CR0 寄存器中的 OPM 位写入“1”，即使能单脉冲模式，选择合适的输出工作模式（匹配输出或 PWM 模式 1/2），对捕获/比较寄存器(TMRx_CCxR)写入合适的比较值，根据需求配置有效电平的极性，计数器在发生下一个更新事件（计数器从重载周期值返回到 0）时计数器将停止计数，并输出一段可控宽度的单脉冲。当 TMRx_CR0 寄存器中的 OPM 位写入“0”，即选择重复连续模式。

29.4.9 互补输出

定时器可以输出两路互补信号，并可通过死区控制器来管理比较输出的关与开。对于死区时间的设置，用户需根据外部器件及其特性（电平转换器的固有延迟、开关器件产生的延迟等）来调整死区时间的长短。

每路输出都可独立选择输出极性（主输出 OCx 或互补输出 OCxN），通过配置 TMRx_CCER 寄存器中的 CCxP 和 CCxNP 位来选择输出极性。其中，互补信号 OCx 和 OCxN 是通过以下多个控制位的组合来控制：TMRx_CCER 寄存器中的 CCxE 和 CCxNE 位、TMRx_DCR 寄存器中的 MOE、OSSI 和 OSSR 位及 TMRx_CR1 寄存器中的 OISx、OISxN 位。更多详细信息，请参考“表 29-6：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位”。

当 TMRx_CCER 寄存器中的 CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1 时，将使能死区插入功能。其中，TMRx_DCR 寄存器中的 DTG 位用于控制所有通道的死区时间设置。死区控制器将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，其上升沿相对参考上升沿存在延迟
- 输出信号 OCxN 与参考信号相反，其上升沿相对参考下降沿存在延迟

如果死区时间大于有效输出（OCx 或 OCxN）的宽度，则不会生成相应的脉冲。

下图为死区发生器的输出信号与参考信号 OCxREF 之间的关系（示例中，设置 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

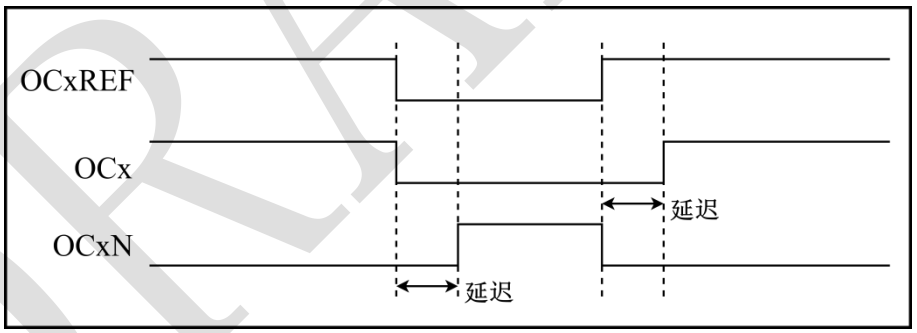


图 29-18 带死区插入的互补输出

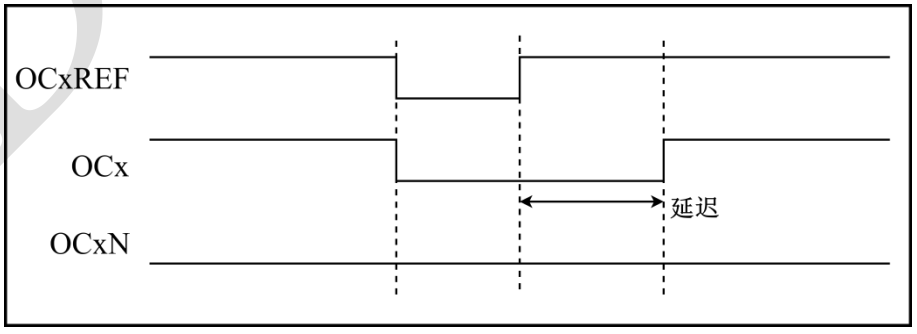


图 29-19 死区时间大于负脉冲宽度的波形

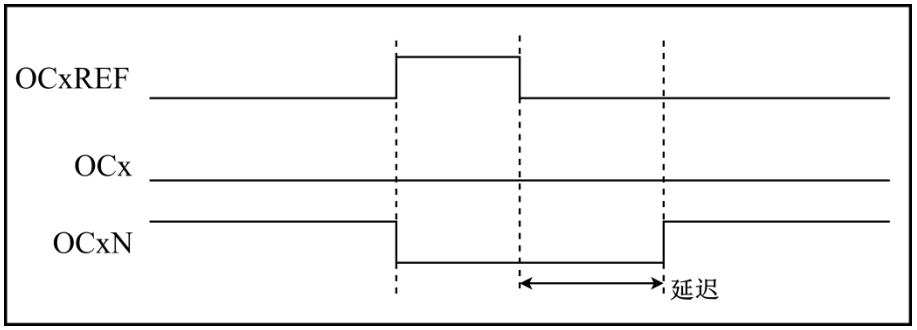


图 29-20　死区时间小于正脉冲宽度的波形

死区时间的设置作用于所有通道并均相同，可通过 TMRx_DCR 寄存器中的 DTG 位进行配置。有关延迟时间的计算信息，请参见 TMRx_DCR 寄存器说明。

表 29-6. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	0	1	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	OCxREF+极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		0	1	0	OCxREF+极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	1	1	OCREF+极性+死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+极性+死区 OCxN_EN=1
		1	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP、OCx_EN=1	OCxREF+极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		1	1	0	OCxREF+极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF+极性+死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+极性+死区 OCxN_EN=1
0	0	X	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
	0		0	1	禁止输出（不由定时器驱动）	
	0		1	0	异步：OCx=CCxP、OCx_EN=0、OCxN=CCxNP、OCxN_EN=0	
	0		1	1	如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应。	
	1		0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
	1		0	1	禁止输出（不由定时器驱动）	
	1		1	0	异步：OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1	

	1		1	1	如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应。

29.4.10 断路保护

使用断路功能时，根据 TMRx_DCR 寄存器中的 MOE 位、OSSI 位和 OSSR 位及 TMRx_CR1 寄存器中的 OISx 位、OISxN 位来控制输出使能及输出电平。任何情况下，OCx 和 OCxN 输出都不能同时设为有效电平。更多详细信息，请参考“表 29-6：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位”。

断路功能的输入源可以是断路输入引脚，也可以是时钟故障事件，后者由时钟复位控制器 RCU 中的时钟安全系统 (CSS) 生成。有关时钟安全系统的详细信息，请参考“第 7.2.6 节：时钟安全系统 (CSS)”。

芯片上电复位后，断路功能默认处于禁止状态，MOE 位为低电平。将 TMRx_DCR 寄存器中的 BKE 位置 1 可使能断路功能，断路输入的极性可通过 TMRx_DCR 寄存器中的 BKP 位来选择，BKE 和 BKP 位可同时修改。

当发生断路（断路输入上出现所设定的电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态或复位状态（通过 OSSI 位进行选择）。
- MOE=0 时，将以 TMRx_CR1 寄存器中的 OISx 位的电平来驱动每个通道的输出。如果 OSSI=0，则定时器将关闭输出使能，否则输出使能始终保持高电平。
- 将断路状态标志（TMRx_SR 寄存器中的 BIF 位）置 1，如果 TMRx_IER 寄存器中的 BIE 位置 1，则可产生中断。
- 如果 TMRx_DCR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件(UEV)时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入“1”。这种情况下，这一特性可用于确保安全。可以切断功率器件、温度传感器或任何安全元件，保证其不被损坏。

注意：断路输入为电平有效，因此，当断路输入为有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将断路状态标志 BIF 清零。

断路事件可由外部 TMRx_BK 引脚输入生成，输入引脚的极性是可配置，其使能由 TMRx_DCR 寄存器中的 BKE 位来控制。

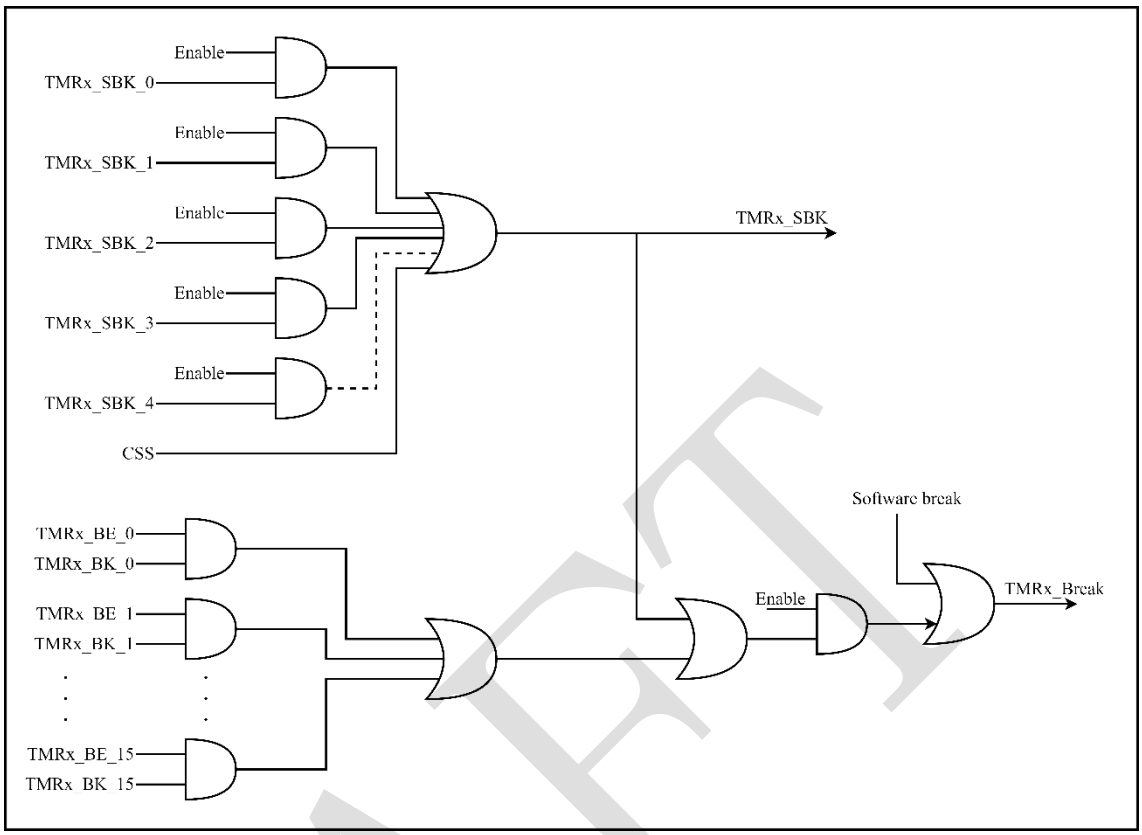


图 29-21　TMRx Break 输入控制示意图

断路事件有以下两种生成方案：

- 使用 TMRx_BK 引脚生成
- 由软件通过 TMRx_UGR 寄存器中的 BG 位生成

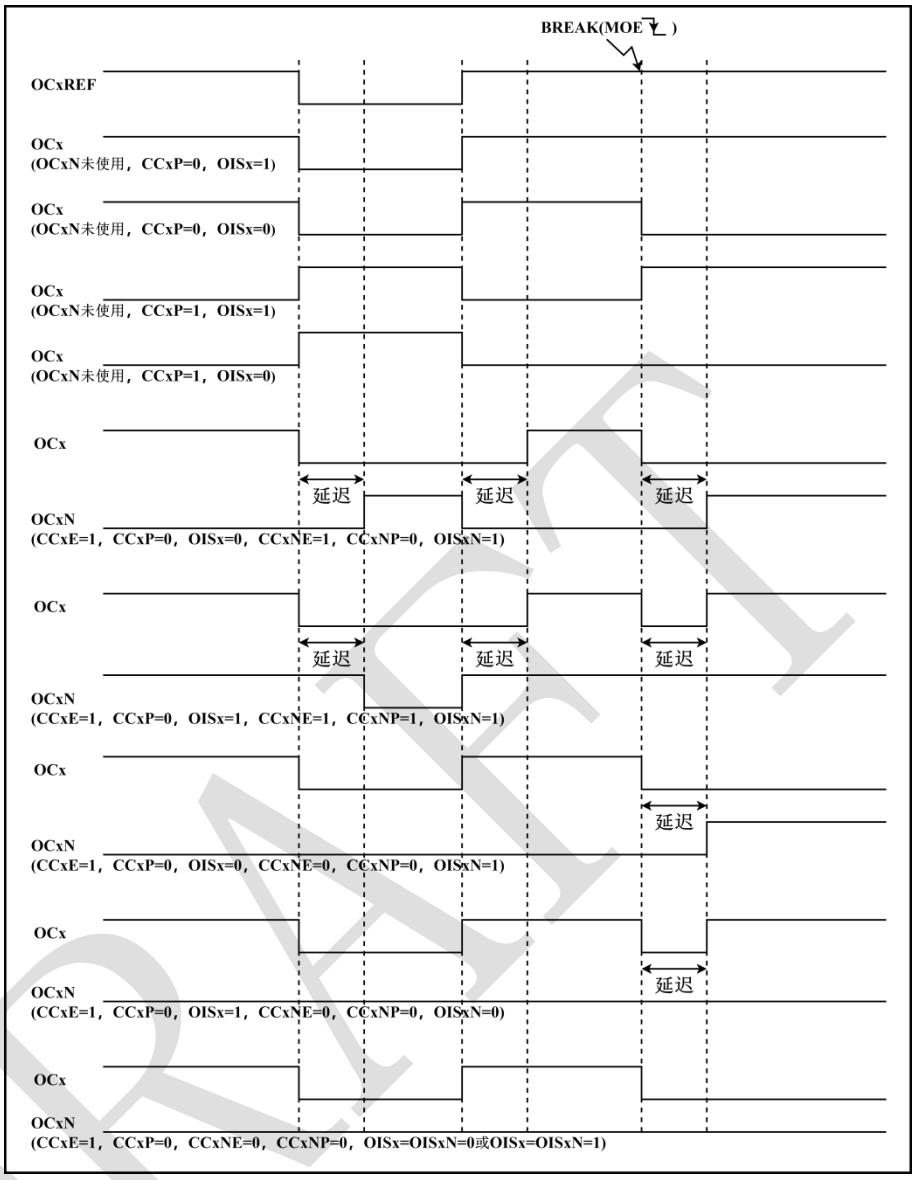


图 29-22　输出的断路响应行为

29.4.11　从模式

定时器可通过外部触发实现以下功能的同步：复位模式、门控模式和触发模式。

从模式：复位模式

当输入触发信号产生上升沿跳变时，计数器及其预分频器将被重新初始化。如果 TMRx_CR0 寄存器中的 URS 位为 0 时，则还会生成更新事件 UEV。然后，所有预装载寄存器 TMRx_CPR 和 TMRx_CCxR 都将更新。

以下示例中，在 TI1 输入信号上出现上升沿时，递增计数器清零：

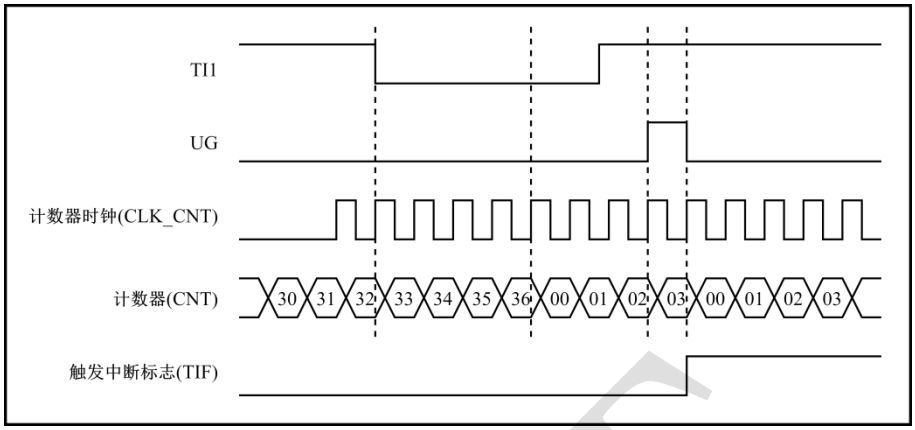


图 29-23　复位模式下的控制电路

计数器使用内部时钟计数，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发 TMRx_SR 寄存器中的 TIF 标准位置 1，使能对应的中断后，还可发送中断请求。

上图显示了自动重载寄存器 TMRx_CPR=0x36 时的行为，TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：门控模式

通过触发输入信号的电平来使能计数器，当输入信号为高电平时，计数器开始计数，当输入信号为低电平时，计数器保持不变。

下图中，递增计数器仅在 TI1 输入为高电平时计数：

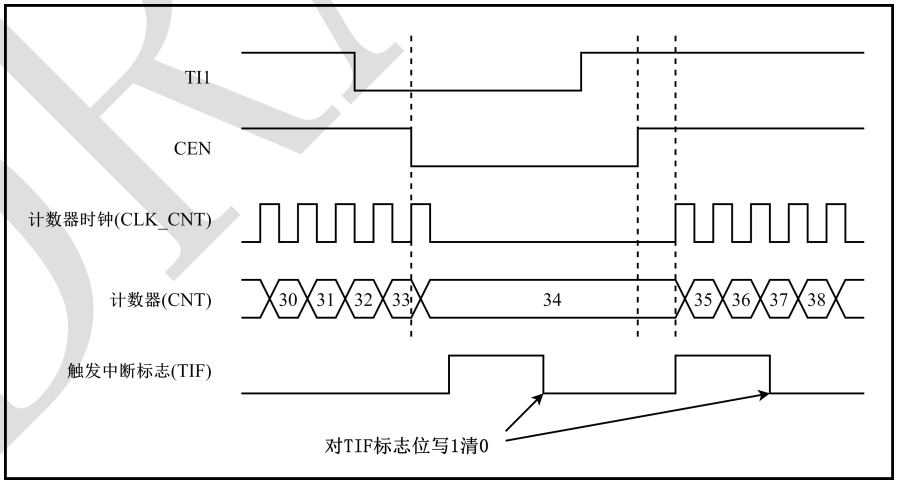


图 29-24　门控模式下的控制电路

只要 TI1 为高电平，计数器就开始根据内部时钟计数，当 TI1 变为低电平时停止计数。当计数器启动或停止时，TMRx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：触发模式

通过输入信号的上升沿来启动计数器，计数器开始计数器。

下图中，递增计数器在 TI1 输入上升沿时，启动递增计数器：

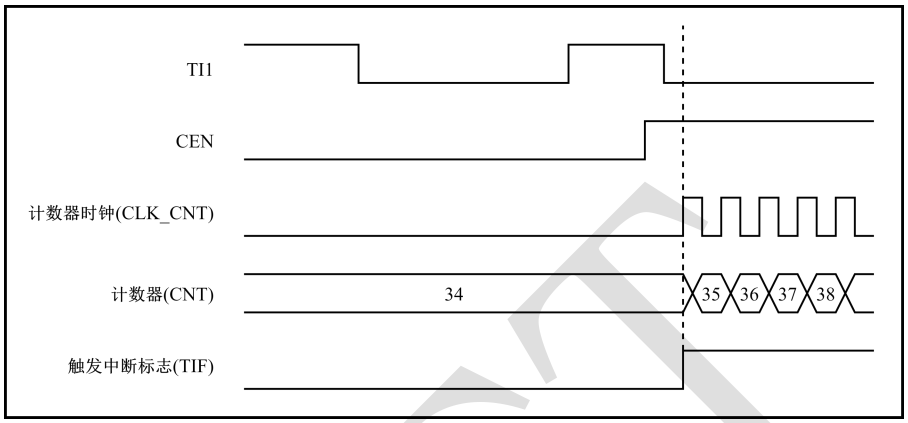


图 29-25　触发模式下的控制电路

当 TI1 出现上升沿时，计数器根据内部时钟开始计数，并且 TIF 标志位置 1。

TI1 的上升沿与实际计数器启动之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：外部时钟模式 2+触发模式

外部时钟模式 2 可与另一种从模式结合使用。这种情况下，ETR 信号作为外部时钟输入，在复位模式、门控模式或触发模式下，可选另一个输入作为触发输入，不建议通过 TMRx_SCR 寄存器中的 TS 位来选择 ETR 作为 TRGI 的输入源。
在下图示例中，当 TI1 出现上升沿时将使能计数器，同时将 TIF 标志位置 1，然后计数器即会在 ETR 信号的每个上升沿递增计数。

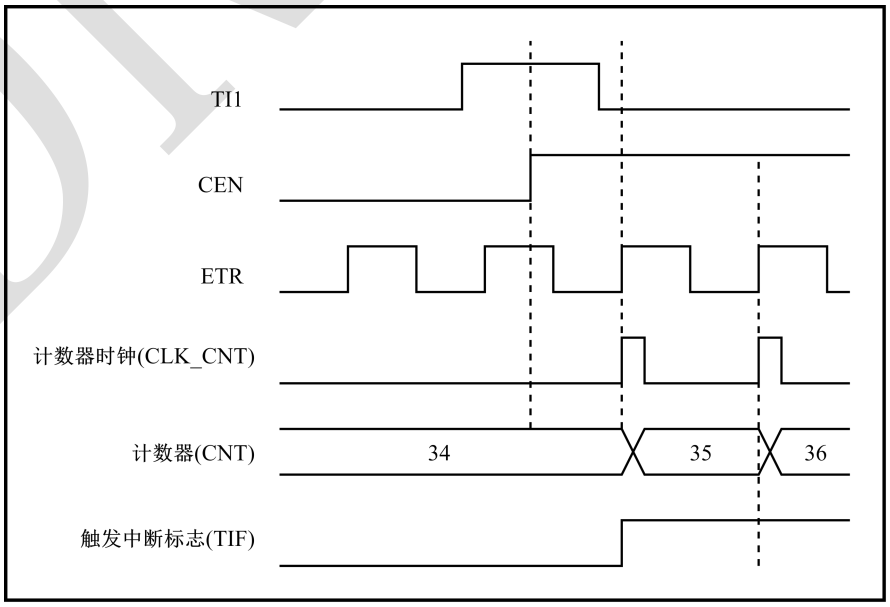


图 29-26　外部时钟模式 2+触发模式下的控制电路

ETR 信号的上升沿与实际计数器之间的延迟是由于 ETRP 输入的重新同步电路引起的。

29.4.12 定时器同步

定时器内部是连在一起，以实现定时器之间的同步与级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

下图简要介绍了从模式的触发选择及主模式选择框图，将一个定时器用作另一个定时器的预分频器：

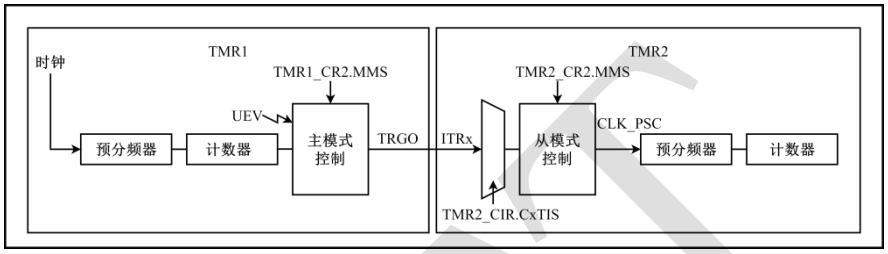


图 29-27　主/从定时器示例

注意：如果选择定时器 1 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动定时器 2 的计数器。

下图使用一个定时器 1 的输出来使能另一个定时器 2 的示例，相关的连接图如下，仅当定时器 1 的 OC1REF 位高电平时，定时器 2 才会根据分频后的时钟进行计数，两个计数器的时钟频率都是基于 3 分频。

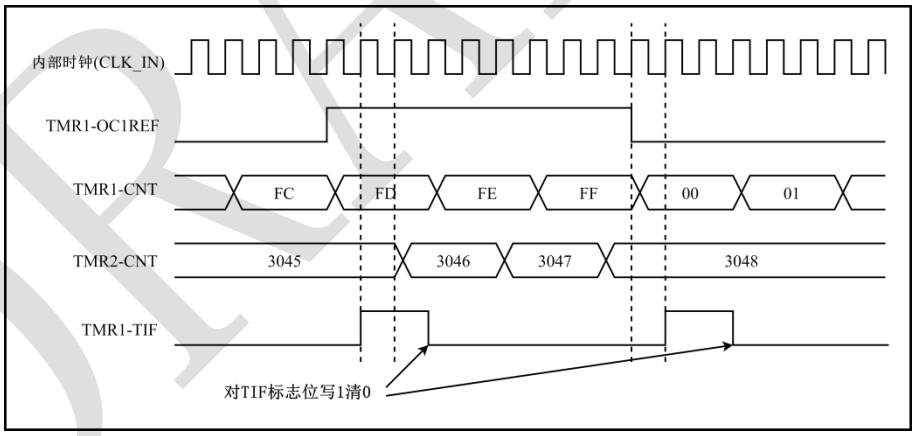


图 29-28 使用定时器 1 的输出来对定时器 2 实施门控控制

29.4.13 调试模式

当内核进入调试模式（Cortex™-M4 内核停止）时，定时器内部计数器会根据 SYSCTRL 模块 SYSDCR 系统调试配置寄存器中的 TMRxDEN 位来选择继续正常工作或者停止工作。

29.4.14 事件与中断

通用定时器在多种情况下会产生相应的事件，事件主要用于更新相应的影子寄存器、执行特

定功能以及产生相应中断。

通用定时器内主要有以下几种事件：

- 定时器上溢事件，简称上溢事件 OVEV(Overflow Event)
- 定时器更新事件，简称更新事件 UEV (Update Event)
  需要使用更新事件，必须将定时器控制寄存器(TMRx_CR0)中的禁止更新位(UDIS)置 0
- 定时器输入捕获/输出比较事件，简称捕获/比较事件 CCEV (Capture/Compare Event)
- 定时器触发事件，简称触发事件 TEV (Trigger Event)
- 定时器断路事件，简称断路事件 BEV (Break Event)
  断路事件源包括断路引脚输入或系统故障输入。

中断主要包含以下五种中断触发：

- 定时器上溢中断，发生上溢事件后，上溢中断标志位(OVIF)同时被置位
- 定时器更新中断，发生更新事件后，更新中断标志位(UIF)同时被置位
- 定时器触发中断，发生触发事件后，更新触发标志位(TIF)同时被置位
- 定时器断路中断，发生断路事件后，更新断路标志位(BIF 或 SBIF)同时被置位

特别说明：如果更新请求源选择位(URS)置 1，即仅上溢事件会产生更新事件，那么软件上如果通过设置更新产生位(UG)，将只会产生更新事件用于更新预加载相关的寄存器，但不会触发定时器更新中断。

- 定时器输入捕获中断，当定时器为输入捕获模式并捕获到相应的边沿时，输入捕获中断标志位（CCxMIF）将被置位。
- 定时器重复捕获中断，当输入捕获中断标志位已经置位(CCxMIF =1)，再次捕获到相应边沿，则输入捕获重复捕获中断标志位（CCxOIF）将被置位。
- 定时器输出比较中断，当定时器位输出比较模式，并且计数值与比较值相匹配时，输出比较中断标志位(CCxMIF)将被置位。

29.4.14.1 上溢事件（Overflow Event）

上溢事件(OVEV)由以下情况产生：

- 计数器 (TMRx_CNTR)值到达计数周期寄存器(TMRx_CPR)设定的值。

上溢事件的作用

计数器到达上溢后，计数器寄存器会立即重新从 0 开始计数 (在连续模式下)。

事件产生后，中断状态寄存器(TMRx_SR)中的上溢中断标志位(OVIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

29.4.14.2 更新事件（Update Event）

更新事件(UEV)可以由以下两种情况产生：

- 计数器上溢：计数器上溢事件产生的同时会产生更新事件。
- 软件置位：软件将定时器事件产生寄存器(TMRx_UGR)中的更新标志位(UG) 置 1。

更新事件使用的注意事项

- 如果需要使用更新事件(UEV)，必须将定时器控制寄存器(TMRx_CR0)中的更新禁止位(UDIS)置 0，使能更新事件功能，否则更新事件将被禁止。
- 如果需要软件置位功能，还必须要配置定时器控制寄存器(TMRx_CR0)中的更新请求位(URS)置 0。否则只有计数器上溢才会产生更新事件。

更新事件的作用

事件产生后，中断状态寄存器(TMRx_SR)中的更新中断标志位(UIF)将会置 1，并且会触发定时器中断 (如果对应的中断已使能)。

特别的，如果在计数器计数的过程中(未达到计数器上溢)，通过软件设置 UG 位产生更新事件后，计数器寄存器也会立即重新从 0 开始计数(在连续模式下)。不同之处在于：通过软件更新标志位(UG)设置产生的更新事件，不会触发上溢事件的中断。

更新事件对自动装载预加载(Auto-Reload Preload)功能的作用

当定时器开启自动装载预加载功能（TMRx_CR0 寄存器内 ARE=1），更新事件产生后，才会将以下几个寄存器的值，更新到内部影子寄存器内使之生效：

- 计数周期寄存器(TMRx_CPR)
- 预分频寄存器(TMRx_PSCR)
- 捕获/比较寄存器(TMRx_CCxR)

注意：软件置位(UG=1)产生的更新事件，不会影响捕获/比较寄存器(TMRx_CCxR)。

如果定时器未开启自动装载预加载功能（TMRx_CR0 寄存器内 ARE=0），则上述寄存器写入新值将立即生效，更新事件将不影响上述寄存器的配置。

29.4.14.3 触发事件（Trigger Event）

触发事件(TEV)由以下情况产生：

- 在从模式下，输入信号上检测到预期的跳变事件
- 在从模式下，软件将事件产生寄存器(TMRx_UGR)中的事件产生标志位(TG)置 1

触发事件的作用

在从模式下，输入信号上检测到预期的跳变事件，中断状态寄存器(TMRx_SR)中的 Trigger 中断标志位(TIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

29.4.14.4 断路事件（Break Event）

断路事件(BEV)由以下情况产生：

- 在比较输出模式下，输入断路信号出现预定的电平信号
- 在从模式下，软件将事件产生寄存器(TMRx_UGR)中的事件产生标志位(BG)置 1

断路事件的作用

在比较输出模式下，输入断路信号出现预定的电平信号（断路功能已使能），即事件触发后，比较输出会根据用户设定的方式进行保护输出。

事件产生后，中断状态寄存器(TMRx_SR)中的上溢中断标志位(BIF 或 SBIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

29.4.14.5 捕获/比较事件（Capture/Compare Event）

捕获/比较事件(CCEV)，可以由以下三种情况产生：

- 比较成功：计数器 (TMRx_CNTR)的值与捕获/比较寄存器(TMRx_CCxR)的值比较成功时(仅针对输出比较模式下)
- 捕获成功：外部输入一个与 MRx_CCER 寄存器中 CCxP 位相符的边沿信号时（仅针对输入捕获模式下）
- 软件产生：将事件产生寄存器(TMRx_UGR)中的事件产生标志位(CCxUG)置 1

捕获/比较事件的作用

在不同模式下捕获/比较事件的作用不相同：

- 对于比较输出模式：
  - 捕获/比较事件产生后，中断状态寄存器(TMRx_SR)中的比较成功中断(CCxMIF)位将会置 1，并触发定时器中断(如果中断已使能)。
  - 如果使能了比较器预加载功能(TMRx_CCMR 寄存器中的 CCxPE 位置 1)，事件产生后(特指比较成功所产生事件)，定时器将会执行以下动作：
    - 定时器的输出，将根据 TMRx_CCMR 寄存器中的比较输出模式 OCxM 位，对输出状态进行相应的改变；
    - 将捕获/比较寄存器(TMRx_CCxR)的值重新加载到内部影子寄存器，使之生效。
    - 将定时器的计数器值(TMRx_CNTR)重新从 0 开始计数(连续循环模式)。
  - 如果没有使能比较器预加载功能(TMRx_CCMR 寄存器中的 CCxPE 位置 0)，事件产生后，定时器的动作与上述情况不同之处在于：对捕获/比较寄存器(TMRx_CCxR)写入新值将立即生效，捕获/比较事件不会影响该寄存器值何时生效。

注意：如果在到达比较值之前(未达到比较成功)，通过软件对 CCxUG 置 1 产生事件，与上述描述基本一致，不同之处在于：不会导致比较输出模式(OCM[2:0])所设定的输出变化，因为本次计数器值并没有与触发捕获/比较事件前所设定的比较值相匹配。

- 对于输入捕获模式：
  - 捕获/比较事件产生后，定时器将当前计数器 (TMRx_CNTR)的值捕获到捕获/比较寄存器(TMRx_CCxR)内；
  - 中断状态寄存器(TMRx_SR)中的捕获中断标志(CCxMIF)位将会置 1，并触发定时器中断(如果对应的中断已使能)。如果本次捕获成功产生捕获/比较事件时，前一次捕获中断标志未及时清除(CCxMIF=1)，则中断状态寄存器(TMRx_SR)中的重复捕获中断标志(CCxOIF)位将会置 1，并且触发定时器中断(如果对应的中断已使能)。

注意：如果通过软件对 CCxUG 置 1 产生的捕获/比较事件，则立刻执行上述动作，而不用外部产生符合捕获成功的信号。

29.4.14.6 中断

中断标志位与中断使能之间的关系如下表描述所示，详细内容请参考“29.4.14 事件与中断”章节描述：

表 29-7 中断标志位与中断使能之间的关系

中断名称	中断使能	中断标志	清除方式	备注
上溢中断	OVIE	OVIF	W1C	计数器计数上溢后产生
更新中断	UIE	UIF	W1C	计数器更新事件触发更新中断
比较捕获中断	CxMIE	CCxMIF	W1C 或读 CCxR	比较捕获模式下，计数值与比较值匹配或捕获到目标边沿
重复捕获中断	CxPIE	CCxOIF	W1C 或读 CCxR	输入捕获中断未及时清除，再次发生捕获
触发中断	TIE	TIF	W1C	在从模式下，输入信号出现目标触发边沿
断路中断	BIE	BIF	W1C	在比较输出下，断路信号出现有效电平
系统断路中断	SBIE	SBIF	W1C	在比较输出下，系统断路信号出现有效电平

29.5 寄存器定义

29.5.1 寄存器列表

TMRx 基地址：

TMR0(0x40018000) TMR1(0x40019000) TMR2(0x4001A000)

偏移	实例地址	名称	默认值	描述
0x00	TMRx 基地址+0x00	TMRx_CR0	0x00000000	TMRx 控制寄存器 0
0x04	TMRx 基地址+0x04	TMRx_CR1	0x00000000	TMRx 控制寄存器 1
0x08	TMRx 基地址+0x08	TMRx_SCR	0x00000000	TMRx Slave 控制寄存器
0x0C	TMRx 基地址+0x0C	TMRx_IER	0x00000000	TMRx 中断使能寄存器
0x10	TMRx 基地址+0x10	TMRx_SR	0x00000000	TMRx 状态寄存器
0x14	TMRx 基地址+0x14	TMRx_UGR	0x00000000	TMRx 更新事件生成寄存器
0x20	TMRx 基地址+0x20	TMRx_CCMR	0x00000000	TMRx 捕获/比较模式寄存器
0x24	TMRx 基地址+0x24	TMRx_CCER	0x00000000	TMRx 捕获/比较使能寄存器
0x28	TMRx 基地址+0x28	TMRx_DCR	0x00000000	TMRx 死区控制寄存器
0x2C	TMRx 基地址+0x2C	TMRx_TTCR	0x00000077	TMRx 触发周期寄存器
0x30	TMRx 基地址+0x30	TMRx_CPR	0x0000FFFF	TMRx 计数周期寄存器
0x38	TMRx 基地址+0x38	TMRx_PSCR	0x00000000	TMRx 预分频寄存器
0x3C	TMRx 基地址+0x3C	TMRx_CNTR	0x00000000	TMRx 计数寄存器
0x40	TMRx 基地址+0x40	TMRx_CRR	0x00000000	TMRx 重复计数寄存器
0x50	TMRx 基地址+0x50	TMRx_CC0R	0x00000000	TMRx 捕获/比较寄存器 0
0x54	TMRx 基地址+0x54	TMRx_CC1R	0x00000000	TMRx 捕获/比较寄存器 1
0x60	TMRx 基地址+0x60	TMRx_CIR	0x00000000	TMRx 捕获输入寄存器
0x64	TMRx 基地址+0x64	TMRx_BPR	0x00000000	TMRx Break 输入极性寄存器
0x68	TMRx 基地址+0x68	TMRx_BER	0x00000000	TMRx Break 使能寄存器

29.5.2 寄存器描述

29.5.2.1 TMRx 控制寄存器 0（TMRx_CR0）

- 名称：TMRx Control Register0
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TI0S	DCD		ARE				OPM	URS	UDIS	CEN
					R/W	R/W		R/W				R/W	R/W	R/W	R/W
					0x0	0x0		0x0				0x0	0x0	0x0	0x0

字段	说明
[10] TI0S	**TI0 输入选择(TI0 Input Selection)** 0： CH0 引脚连接到 TI0 输入 1： CH0~CH1 引脚异或后连接到 TI0 输入
[9:8] DCD	**时钟分频(Clock Division)** 此位指示定时器时钟频率与死区发生器及滤波发生器的时钟频率之间分频比； 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频
[7] ARE	**自动重载使能(AutoReload Enable)** AutoReload 功能使能，是否开启影子寄存器加载功能； 0: 关闭重载功能 1: 开启重载功能 Note： 1. 预加载功能涉及的寄存器有：计数终止值寄存器、预分频寄存器； 2. 如果不开启预加载功能，则相关寄存器写入新值将立刻生效； 3. 如果开启预加载功能，则相关寄存器写入新值后，在下一个更新事件（UEV）到来后生效。
[3] OPM	**单脉冲模式(One Pulse Mode)** 0: 计数器在发生更新事件时不会停止计数； 1: 计数器在发生下一更新事件时停止计数(将 CEN 位清零)； Note：配置为单次模式后，计数器溢出后，将自动停止计数。
[2] URS	**更新事件源(Update Request Source)** 此位由软件置 1 和清零，用以选择更新事件源。 0: 当配置为 0 时，所有以下事件都会生成更新事件: — 计数器上溢 — 将 UG 位置 1 — 通过从模式生成的更新事件 1: 当配置为 1 时，只有计数器上溢会生成更新事件。
[1] UDIS	**更新禁止位(Update Disable)** 此位由软件置 1 和清零，用以使能/禁止更新事件的生成； 0:更新使能，更新(UEV)事件可通过以下事件之一生成: — 计数器上溢 — 将 UG 位置 1 — 通过从模式生成的更新事件 1:更新禁止 Note: 更新禁止后，不会生成更新事件(不会产生更新中断)，各影子寄存器的值保持不变，如果 UG 位置

	1，则会重新初始化计数器和预分频器。
[0] CEN	**计数器使能位(Counter Enable):** 1:开始计数器 0:关闭计数器 Note: 单次模式下，当发生更新事件时会自动关闭 CEN；

29.5.2.2 TMRx 控制寄存器 1（TMRx_CR1）

- **名称：TMRx Control Register1**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OIS1N	OIS1	OIS0N	OIS0						MMS		
				R/W	R/W	R/W	R/W						R/W		
				0x0	0x0	0x0	0x0						0x0		

字段	说明
[11] OIS1N	**CH1N 空闲状态输出电平(CH1N Output Idle-State):** 0: 当 MOE=0 时，OC0N=0 1: 当 MOE=0 时，OC0N=1
[10] OIS1	**CH1 空闲状态输出电平(CH1 Output Idle-State):** 0: 当 MOE=0 时，OC0=0 1: 当 MOE=0 时，OC0=1
[9] OIS0N	**CH0N 空闲状态输出电平(CH0N Output Idle-State):** 0: 当 MOE=0 时，OC0N=0 1: 当 MOE=0 时，OC0N=1
[8] OIS0	**CH0 空闲状态输出电平(CH0 Output Idle-State):** 0: 当 MOE=0 时，OC0=0 1: 当 MOE=0 时，OC0=1
[2:0] MMS	**主模式选择(Master Mode Selection):** 这些位用于选择主模式下将要发送到从定时器的同步的信息(TRGO)，组合描述如下： 000: 复位 - 寄存器(UGR)中的 UG 位用作触发输出，如果从模式控制器配置为复位模式，则 TRGO 上的信号用于从定时器的复位操作； 001: 使能 - 计数器使能信号用作触发输出(TRGO)，该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器； 010: 更新 - 选择更新事件作为触发输出(TRGO)。例如：主定时器可用作从定时器的预分频器； 011: 捕获比较事件 - 一旦发生输入捕获或比较匹配事件，当 CC0IF 标志被置 1 时，触发输出都会发送一个正脉冲； 100: 比较输出 - OC0 信号用作触发输出(TRGO)； 101: 比较输出 - OC1 信号用作触发输出(TRGO)；

29.5.2.3 TMRx Slave 控制寄存器（TMRx_SCR）

- 名称：**TMRx Slave Control Register**
- 偏移地址：**0x08**
- 默认值：**0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TS					FE				SMS			
			R/W					R/W				R/W			
			0x0					0x0				0x0			

字段	说明
[12:8] TS	**触发选择(Trigger Selection)** 此位域可选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) ..... 01111: 内部触发 15(ITR15) 10000: 滤波后的定时器输入 0(CH0-FE-双边沿) 10001: 滤波后的定时器输入 0(CH0-F) 10010: 滤波后的定时器输入 1(CH1-F)
[7] FE	**主从快速同步使能(Master/Slave Fast-Sync Enable)** 此位域用于同步多个定时器完美同步计数。 0：不执行任何操作 1：避免当前定时器的触发输入事件(TRGI)的延迟，以使当前定时器与其主定时器实现完美同步。
[3:0] SMS	**从模式选择(Slave Mode Selection):** 选择外部触发信号(TRGI)的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器说明）； 0000: 禁止从模式 0001: 保留 0010: 保留 0011: 保留 0100: 复位模式 - 在出现所选触发输入(TRGI)上升沿时，重新初始化计数器并生成一个寄存器更新事件； 0101: 门控模式 - 触发输入(TRGI)为高电平时使能计数器时钟，只要触发输入变为低电平，计数器立即停止计数（但不复位），计数器的启动和停止都被控制； 0110: 触发模式 - 触发信号 TRGI 出现上升沿时启动计数器（但不复位），只控制计数器的启动，可使用 FE 来快速同步使能； 0111: 外部时钟模式 - 由所选触发信号(TRGI)的上升沿提供计数器时钟； 1000: 组合复位+触发模式 - 在出现所选触发输入(TRGI)上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器； 1001: 组合门控+复位模式 - 在触发输入(TRGI)为高电平时使能计数器时钟，只要触发输入变为低电平，计数器立即停止计数并复位，计数器的启动和停止都被控制；

29.5.2.4 TMRx 中断使能寄存器（TMRx_IER）

- 名称：TMRx Interrupt Enable Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				BIE	TIE	OVIE	UIE					C1OIE	C1MIE	C0OIE	C0MIE
				R/W	R/W	R/W	R/W					R/W	R/W	R/W	R/W
				0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0

字段	说明
[11] BIE	**断路中断使能(Break Interrupt Enable)** 0：关闭 1：使能
[10] TIE	**触发中断使能(Trigger Interrupt Enable)** 0：关闭 1：使能
[9] OVIE	**计数上溢中断使能(Overflow Interrupt Enable)** 1：使能中断 0：禁止中断
[8] UIE	**更新中断使能(Update Interrupt Enable)** 1：使能中断 0：禁止中断
[3] C1OIE	**CH1 重复捕获中断使能(CH1 OverCapture Interrupt Enable):** 1: 中断使能 0: 中断关闭
[2] C1MIE	**CH1 比较中断使能(CH1 Capture/Compare Interrupt Enable):** 1: 中断使能 0: 中断关闭
[1] C0OIE	**CH0 重复捕获中断使能(CH0 OverCapture Interrupt Enable):** 1: 中断使能 0: 中断关闭
[0] C0MIE	**CH0 比较中断使能(CH0 Capture/Compare Interrupt Enable):** 1: 中断使能 0: 中断关闭

29.5.2.5 TMRx 状态寄存器（TMRx_SR）

- 名称：TMRx Status Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SBIF	BIF	TIF	OVIF	UIF					CC1OIF	CC1MIF	CC0OIF	CC0MIF
			R/W1C	R/W1C	R/W1C	R/W1C	R/W1C					R/W1C	R/W1C	R/W1C	R/W1C
			0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0

字段	说明
[12] SBIF	**系统断路中断标志位(System Fault Break Interrupt Flag)** 只要断路输入变为有效状态，此标志便由硬件置 1，断路输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路输入上检测到有效电平
[11] BIF	**断路中断标志位(Break Interrupt Flag)** 只要断路输入变为有效状态，此标志便由硬件置 1，断路输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路输入上检测到有效电平
[10] TIF	**触发中断标志(Trigger Interrupt Flag)** 该标志在发生触发事件时由硬件置 1，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将置 1，需要通过软件清零。 0: 未发生触发事件 1: 触发中断挂起
[9] OVIF	**计数器上溢中断标志位(Counter Overflow Interrupt Flag)** 0: 未发生上溢 1: 计数器上溢
[8] UIF	**更新中断标志(Update Interrupt Flag)** 该位在发生更新事件时通过硬件置 1,但需要通过软件清零。 0: 未发生更新 1: 更新中断挂起 该位在以下情况下更新寄存器时由硬件置 1: — 计数器发生上溢且当 CR 寄存器中 UDIS="0"时； — 当 CR 寄存器中 URS="0"且 UDIS="0"，通过软件使用 UGR 寄存器中的 UG 位重新初始化计数器 Counter 时；
[3] CC1OIF	**CC1 重复捕获成功中断标志位(CC1 OverCapture Interrupt Flag)** 参考 CC0 的描述
[2] CC1MIF	**CC1 比较成功中断标志位(CC1 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[1] CC0OIF	**CC0 重复捕获成功中断标志位(CC0 OverCapture Interrupt Flag)** 0: 未检测到重复捕获 1: 当 CPIF=1，同时 Caputre 再次捕获到匹配的边沿，将置位该位(重复捕获成功有效) Note: 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1，通过软件写 1 可将该位清零
[0] CC0MIF	**CC0 比较成功中断标志位(CC0 Capture/Compare Interrupt Flag)** 配置为输入: 此位将在发生捕获事件时由硬件置 1，通过软件写 1 清 0 或读取 Capture 值寄存器时自动清 0。 0: 未发生输入捕获事件。 1: Compare 预加载寄存器中已捕获到计数器值（检测到与所选极性匹配的边沿） 配置为输出: 当计数器与比较值匹配时，此标志由硬件置 1，但需要通过软件写 1 清零。 0: 不匹配

1: 计数器的值与 Compare 寄存器的值匹配，当 Compare 值大于 Counter 周期值时， 将在计数器发生上溢时变为高电平；当 UGR 寄存器 UG 位软件置 1，也会触发 Compare 中断；

29.5.2.6 TMRx 更新事件生成寄存器（TMRx_UGR）

- **名称：TMRx Update Generation Register**
- **偏移地址：0x14**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										CC1UG	CC0UG		BG	TG	UG
										R/W1C	R/W1C		R/W1C	R/W1C	R/W1C
										0x0	0x0		0x0	0x0	0x0

字段	说明
[5] CC1UG	**CC1 捕获/比较更新事件生成(CC1 Capture/Compare Update Generation)** 参考 CC0 的描述
[4] CC0UG	**CC0 捕获/比较更新事件生成(CC0 Capture/Compare Update Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零 0: 不执行任何操作 1: 重生 Capture/Compare 更新事件 配置为输出: 使能后，Compare 中断标志置 1 并发送相应的中断，生成更新事件，更新 Compare 值； 配置为输入: 将捕获到计数器当前值写入 CCxR 寄存器中，使能后，Capture 中断标志置 1 并发送相应中断，如果 Capture 中断标志已为高电平，重复标志将置 1
[2] BG	**断路事件生成(Break Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: SR 寄存器中的 BIF 标志置 1，生成 Break 事件。
[1] TG	**触发事件生成(Trigger Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: SR 寄存器中的 TIF 标志置 1，生成 Trigger 事件。
[0] UG	**更新事件生成(Update Generation)** 此位由软件置 1 生成事件，并由硬件自动清零; 0: 不执行任何操作 1: 重新初始化计数器并生成寄存器更新事件，预分频器计数器也将清零(但预分频比不受影响)，计数器清零

29.5.2.7 TMRx 捕获/比较模式寄存器（TMRx_CCMR）

- **名称：TMRx Capture Compare Mode Register**
- **偏移地址：0x20**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC1M				CC1PE		CC1S		OC0M				CC0PE		CC0S	
R/W				R/W		R/W		R/W				R/W		R/W	
0x0				0x0		0x0		0x0				0x0		0x0	

字段	说明
[15:12] OC1M	**CC1 输入滤波/输出比较模式选择(CC1 Input Filter/Output Compare Mode)** 参考 CC0 的描述
[11:10] CC1PE	**CC1 输入捕获分频/比较自动加载使能(CC1 Compare Auto-Realod Enable)** 参考 CC0 的描述
[9:8] CC1S	**CC1 捕获/比较方向选择(CC1 Capture Compare Selection)** 参考 CC0 的描述
[7:4] OC0M	**CC0 输入滤波/输出比较模式选择(CC0 Input Filter/Output Compare Mode)** 配置为输出，OCxRef 的输出模式（有效电平为高电平） 0000: 冻结，输出比较寄存器与计数器进行比较不会对输出造成任何影响(保持原有状态) 0001: 当计数器与捕获/ 比较寄存器匹配时，输出有效电平(高电平) 0010: 当计数器与捕获/ 比较寄存器匹配时，输出无效电平(低电平) 0011: 当计数器与捕获/ 比较寄存器匹配时，发生翻转 0100: 强制变为无效电平(低电平) 0101: 强制变为有效电平(高电平) 0110: PWM 模式 1 - 当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态(高有效) 0111: PWM 模式 2 - 当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态(高有效) 1100: 组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同，参考是 OC1REF 或 OC2REF 的结果。 1101: 组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同，参考是 OC1REF 和 OC2REF 的结果。 配置为输入 0: 无滤波 1: Ttds x1 2: Ttds x2 ……
[3:2] CC0PE	**CC0 输入捕获分频/比较自动加载使能(CC0 Compare Auto-Realod Enable)** 配置为输出 0: 禁止自动预装载，可随时向预装载寄存器写入数据，写入后将立即使用新值 1: 使能自动预装载，可读/写访问预装载寄存器，而预装载值在每次生成更新事件时都会装载到活动寄存器中 配置为输入 此位域定义 CC0 输入(IC0)的预分频比。 00：无预分频器，捕获输入上每检测到一个边沿便执行捕获 01：每发生 2 个事件便执行一次捕获 10：每发生 4 个事件便执行一次捕获 11：每发生 8 个事件便执行一次捕获
[1:0] CC0S	**CC0 捕获/比较方向选择(CC0 Capture Compare Selection)** 此位定义对应通道的方向（输入/输出）。 00: CC0 通道配置为输出。 01: CC0 通道配置为输入，输入映射到 CH0 上 10: CC0 通道配置为输入，输入映射到 CH1 上 11: CC0 通道配置为输入，输入映射到 ITR 上，此模式仅在通过 TS 位选择内部触发输入时有效；

29.5.2.8 TMRx 捕获/比较使能寄存器（TMRx_CCER）

- 名称：TMRx Capture Compare Enable Register
- 偏移地址：0x24
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CC1P		CC1NE	CC1E	CC0P		CC0NE	CC0E
								R/W		R/W	R/W	R/W		R/W	R/W
								0x0		0x0	0x0	0x0		0x0	0x0

字段	说明
[7:6] CC1P	**CC1 捕获/比较极性选择(CC1 Compare/Capture Polarity)** 参考 CC0 的描述
[5] CC1NE	**CC1N 捕获/比较互补通道使能(CC1N Capture/Compare Enable)** 参考 CC0 的描述
[4] CC1E	**CC1 捕获/比较通道使能(CC1 Capture/Compare Enable)** 参考 CC0 的描述
[3:2] CC0P	**CC0 捕获/比较极性选择(CC0 Compare/Capture Polarity)** 配置为输出(bit3 和 bit2 分别对应 CC0N 和 CC0 的输出极性选择) 0: 高电平有效 1: 低电平有效 配置为输入: 00: 关闭 01: 上升沿 10: 下降沿 11: 上升沿和下降沿
[1] CC0NE	**CC0N 捕获/比较互补通道使能(CC0N Capture/Compare Enable)** 配置为输出: 0: 关闭 1: 开启
[0] CC0E	**CC0 捕获/比较通道使能(CC0 Capture/Compare Enable)** 配置为输出: 0: 关闭 1: 开启 配置为输入: 0: 禁止捕获 1: 使能捕获

29.5.2.9 TMRx 死区控制寄存器（TMRx_DCR）

- 名称：TMRx Dead-Time Contorl Register
- 偏移地址：0x28
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										BKF					
										R/W					
										0x0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKF		BKP	BKE	MOE	AOE	OSSR	OSSI	DTG							
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W							
0x0		0x0	0x0	0x0	0x0	0x0	0x0	0x0							

字段	说明
[21:14] BKF	**断路滤波器(Break Filter)** 0: 无滤波 1: Ttds x 1 2: Ttds x 2 ……
[13] BKP	**断路输入极性(Break Polarity)** 0：断路输入为高电平有效 1：断路输入为低电平有效
[12] BKE	**断路控制使能(Break Enable)** 0：禁止断路输入 1：使能断路输入
[11] MOE	**主路输出使能(Main Output Enable)** 只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1，此位仅对配置为输出的通道有效。 0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。 1：如果 OC 和 OCN 输出的相应使能位（CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。
[10] AOE	**自动输出使能(Automatic Output Enable)** 0： MOE 只能由软件置 1 1： MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果断路输入无效）
[9] OSSR	**运行模式下关闭状态选择(Off-State Selection for Run-Mode)** 此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。 0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。
[8] OSSI	**空闲模式下关闭状态选择(Off-State Selection for Idle-Mode)** 此位在 MOE=0 时作用于配置为输出的通道。 0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0） 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平，然后设置 OC/OCN 使能输出信号=1
[7:0] DTG	**配置死区时间选择(Dead-Time Generator Setup)** 此位域定义插入到互补输出之间的死区持续时间。 DTG[7:5]=0xx => DT=DTG[7:0] x Tdtg，　　其中 Tdtg=Ttds。 DTG[7:5]=10x => DT=(64+DTG[5:0]) x Tdtg，其中 Tdtg=2xTtds。 DTG[7:5]=110 => DT=(32+DTG[4:0]) x Tdtg，其中 Tdtg=8xTtds。 DTG[7:5]=111 => DT=(32+DTG[4:0]) x Tdtg，其中 Tdtg=16xTtds。 示例：如果 Ttds=125ns(8MHz)，则可能的死区值为： 0 到 15875ns（步长为 125ns） 16us 到 31750ns（步长为 250ns） 32us 到 63us　（步长为 1us） 64us 到 126us　（步长为 2us）

29.5.2.10 TMRx 触发周期寄存器（TMRx_TTCR）

- 名称：TMRx Trigger Cycles Register
- 偏移地址：0x2C
- 默认值：0x00000077
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TC1E	TC0E	TCW				TOW			
						R/W	R/W	R/W				R/W			
						0x0	0x0	0x7				0x7			

字段	说明
[9] TC1E	**TRGO CH1 输出使能(Trigger CH1 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[8] TC0E	**TRGO CH0 输出使能(Trigger CH0 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[7:4] TCW	**TRGCC 脉冲宽度控制(Trigger Compare/Caputre Width)** 0: 1 个 Cycle 1: 2 个 Cycle …… Note： 1. TRGCC 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle： 2. TRGCC 脉冲宽度必须小于 Timer 的 Period 值(CPV).
[3:0] TOW	**TRGO 脉冲宽度控制(Trigger Output Width)** 0: 1 个 Cycle 1: 2 个 Cycle …… Note： 1. TRGO 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle： 2. TRGO 脉冲宽度必须小于 Timer 的 Period 值(CPV).

29.5.2.11 TMRx 计数周期寄存器（TMRx_CPR）

- 名称：TMRx Counter Period Register
- 偏移地址：0x30
- 默认值：0x0000FFFF
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPV															
R/W															
0xFFFF															

字段	说明
[15:0] CPV	**计数周期值(Counter Period Value)** 通过配置此寄存器设置定时器结束计数的值。此外，该寄存器支持自动预装载（Auto-Reload）功能： 1. 不使能自动预装载功能（ARE=0） 对该寄存器的设置将立即作用到内部影子寄存器内，本次计数周期内即生效； 2. 使能自动预装载功能（ARE=1） 对该寄存器的设置将在更新事件（UEV）产生后，再更新至内部影子寄存器内生效。

29.5.2.12 TMRx 预分频寄存器（TMRx_PSCR）

- 名称：TMRx Prescaler Register
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC															
R/W															
0x0															

字段	说明
[15:0] PSC	**预分频寄存器（Prescaler Value)** 该寄存器支持自动预装载(Auto-Reload)功能，每次发生更新事件时会装载到实际预分频器寄存器的值； Note：计数器时钟频率 CLK_CNT 等于 FCLK/(PSC+1)，其中 PSC 范围为 0~65535： 0:不分频，系统时钟 1:2 分频 2:3 分频 ……

29.5.2.13 TMRx 计数寄存器（TMRx_CNTR）

- 名称：TMRx Counter Register
- 偏移地址：0x3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															
0x0															

字段	说明
[15:0] CNT	**定时器计数值(Counter Value)** 通过该寄存器可对定时器计数器的值进行读取/写入。 需注意，因为系统与外设之间存在一定的总线延迟，对该寄存器的读取/写入操作可能存在一定的偏差。

29.5.2.14 TMRx 重复计数寄存器（TMRx_CRR）

- 名称：TMRx Counter Repeat Register
- 偏移地址：0x40
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								REP							
								R/W							
								0x0							

字段	说明
[7:0] REP	**重复计数值(Counter Repeat Value)** 使能预装载寄存器之前，用户可通过这些位设置比较寄存器的更新频率（从预装载寄存器向活动寄存器周期性传输数据），使能更新中断时，也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。 由于只有生成重复更新事件时才会重载 REP 值，因此在生成下一重复更新事件之前，无论向该寄存器写入何值都无影响。 这意味着在 PWM 模式(REP+1)下对应于边沿对齐模式的 PWM 周期数。

29.5.2.15 TMRx 捕获/比较寄存器 0（TMRx_CC0R）

- 名称：TMRx Capture Compare Register0
- 偏移地址：0x50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC0V															
R/W															
0x0															

字段	说明
[15:0] CC0V	**CC0 捕获/比较值(CC0 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

29.5.2.16 TMRx 捕获/比较寄存器 1（TMRx_CC1R）

- 名称：TMRx Capture Compare Register1
- 偏移地址：0x54
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1V															
R/W															
0x0															

字段	说明
[15:0] CC1V	**CC1 捕获/比较值(CC1 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次

	2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

29.5.2.17 TMRx 捕获输入寄存器（TMRx_CIR）

- **名称：TMRx Capture Input Register**
- **偏移地址：0x60**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				C1TIS								C0TIS			
				R/W								R/W			
				0x0								0x0			

字段	说明
[11:8] C1TIS	**CH1 触发输入选择(CH1 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[3:0] C0TIS	**CH0 触发输入选择(CH0 Trigger Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……

29.5.2.18 TMRx Break 输入极性寄存器（TMRx_BPR）

- 名称：TMRx Break Polarity Register
- 偏移地址：0x64
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BPOL															
R/W															
0x0															

字段	说明
[15:0] BPOL	**Break 输入极性选择(Break Input Polarity Selection)** 0：高电平有效 1：低电平有效

29.5.2.19 TMRx Break 使能寄存器（TMRx_BER）

- 名称：TMRx Break Enable Register
- 偏移地址：0x68
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIEN															
R/W															
0x0															

字段	说明
[15:0] BIEN	**Break 输入使能控制(Break Input Enable)** 0：关闭 1：使能

30 通用定时器（TMR3/TMR4）

30.1 简介

TMR3/TMR4 定时器包含一个 32 位自动重载计数器，该计数器由可编程预分频器驱动。

通用定时器可用于多种用途，包括最基本的定时功能（基础计时）以及测量输入信号的脉冲宽度（输入捕获）、生成 PWM 输出波形（输出比较）、外部脉冲计数(ETR)等多种灵活配置的功能。

使用定时器预分频器和 RCU 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

TMR3/TMR4 定时器之间彼此完全独立，不共享任何资源，可以同步操作。

30.2 主要特性

TMR3/4 主要具有以下特性：

- 32 位递增、递减和递增/递减自动重载计数器
- 16 位的预分频器，用于对计数器的时钟源进行预分频，分频系数介于 1 和 65536 之间
- 32 位的计数比较值和周期值寄存器
- 两种工作模式：
  - 循环模式，计数完成后自动重载并继续计数
  - 单次模式，计数完成后自动关闭定时器的使能
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 支持主从模式，可以实现多个定时器的互连
- 发生如下事件时产生中断：
  - 计数器上溢(Overflow Event)
  - 更新事件(Update Event) (需使能)：计数器上溢产生更新事件，也可软件产生(UG)
  - 输入捕获(Capture Event)、重复捕获(Over Capture Event)
  - 输出比较(Compare Event)

30.3 结构框图

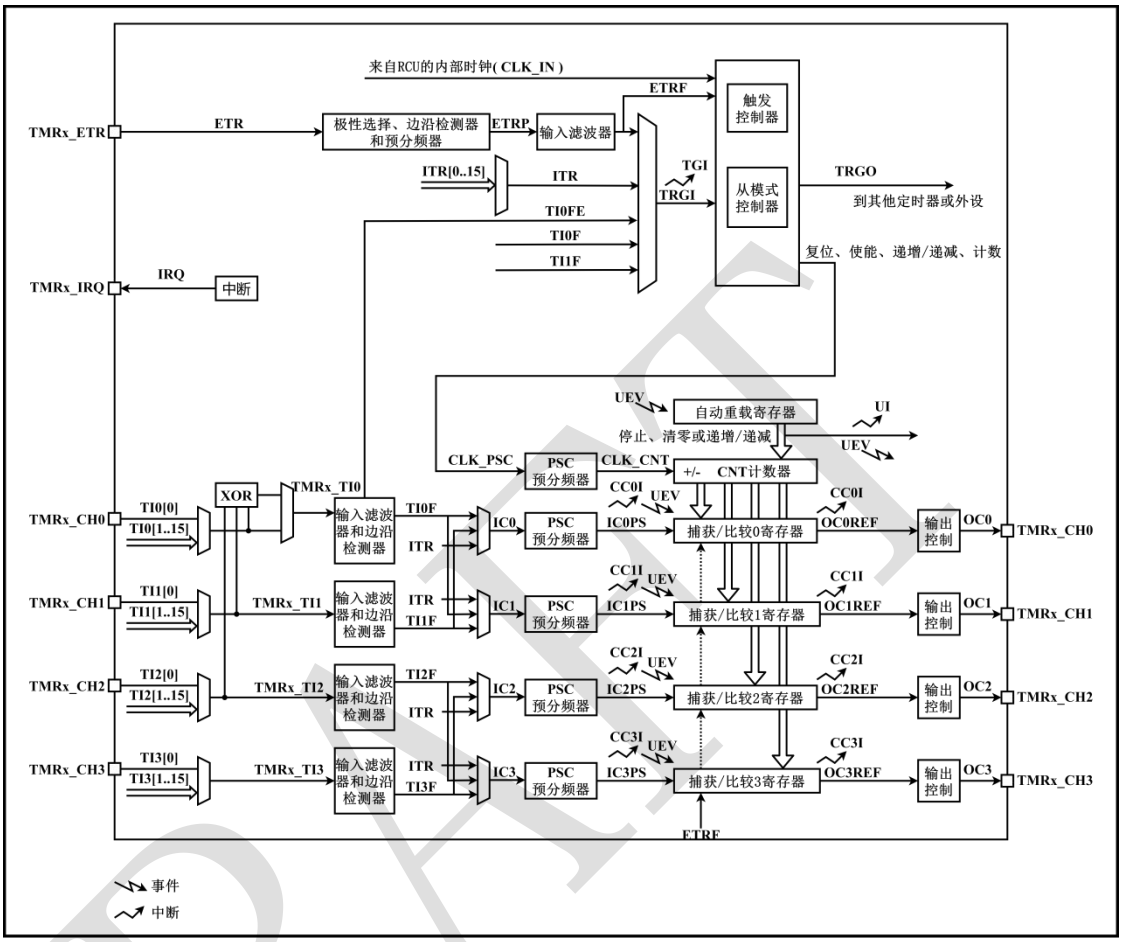


图 30-1 TMR 结构框图

30.4 功能描述

30.4.1 内部信号

表 30-1 列出了 TMRx 定时器主要的输入/输出信号及简要描述：

表 30-1 TMRx 内部信号

Signal Name	Signal Type	Description
HCLK	Input	TMRx AHB Clock
TMRx_CH0/1/2/3	Input/Output	TMRx 多功能复用通道，可以用作 Compare、Capture 及 PWM，还可以用于外部时钟或 Trigger 触发输入。
TMRx_ETR	Input	TMRx 外部时钟源 2 输入引脚
TRGO	Output	TMRx 内部触发输出信号，可用于触发其他外设
IRQ	Output	TMRx 全局中断，包括 Capture/Compare/Update 中断请求。

表 30-2 列出了 TMRx 定时器 TMRx_TI 输入信号的连接关系：

表 31-2 TMRx_TI 信号内部互连

表 a. TMR3_TI 信号内部互连

TMRx_TIx 输入	Sources			
	事件名称(CH0)	事件名称(CH1)	事件名称(CH2)	事件名称(CH3)
TMRx_TI_0	CMP0_OUT	CMP0_OUT	CMP0_OUT	CMP0_OUT
TMRx_TI_1	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP1_OUT
TMRx_TI_2	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT
TMRx_TI_3	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT
TMRx_TI_4	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT
TMRx_TI_5	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT
TMRx_TI_6	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT
TMRx_TI_7	CMP7_OUT	CMP7_OUT	CMP7_OUT	CMP7_OUT
TMRx_TI_8	HSI	HSE	LSI	CMP8_OUT
TMRx_TI_9	Reserved	Reserved	Reserved	Reserved
TMRx_TI_10	Reserved	Reserved	Reserved	Reserved
TMRx_TI_11	Reserved	Reserved	Reserved	Reserved
TMRx_TI_12	TMR3_CH0	TMR3_CH0	TMR3_CH0	TMR3_CH0
TMRx_TI_13	TMR3_CH1	TMR3_CH1	TMR3_CH1	TMR3_CH1
TMRx_TI_14	TMR3_CH2	TMR3_CH2	TMR3_CH2	TMR3_CH2
TMRx_TI_15	TMR3_CH3	TMR3_CH3	TMR3_CH3	TMR3_CH3

表 b. TMR4_TI 信号内部互连

TMRx_TIx 输入	Sources			
	事件名称(CH0)	事件名称(CH1)	事件名称(CH2)	事件名称(CH3)
TMRx_TI_0	CMP0_OUT	CMP0_OUT	CMP0_OUT	CMP0_OUT
TMRx_TI_1	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP1_OUT
TMRx_TI_2	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT
TMRx_TI_3	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT

TMRx_TI_4	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT
TMRx_TI_5	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT
TMRx_TI_6	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT
TMRx_TI_7	CMP7_OUT	CMP7_OUT	CMP7_OUT	CMP7_OUT
TMRx_TI_8	HSI	HSE	LSI	CMP8_OUT
TMRx_TI_9	Reserved	Reserved	Reserved	Reserved
TMRx_TI_10	Reserved	Reserved	Reserved	Reserved
TMRx_TI_11	Reserved	Reserved	Reserved	Reserved
TMRx_TI_12	TMR4_CH0	TMR4_CH0	TMR4_CH0	TMR4_CH0
TMRx_TI_13	TMR4_CH1	TMR4_CH1	TMR4_CH1	TMR4_CH1
TMRx_TI_14	TMR4_CH2	TMR4_CH2	TMR4_CH2	TMR4_CH2
TMRx_TI_15	TMR4_CH3	TMR4_CH3	TMR4_CH3	TMR4_CH3

表 30-3 列出了 TMRx 定时器内部 ITR 输入信号的连接关系：

表 30-3 TMRx_ITR 信号内部互连

TMRx_ITRx 输入	Sources	
	TMR3	TMR4
TMRx_ITR_0	TMR7_TRGO	TMR7_TRGO
TMRx_ITR_1	TMR8_TRGO	TMR8_TRGO
TMRx_ITR_2	TMR0_TRGO	TMR0_TRGO
TMRx_ITR_3	TMR1_TRGO	TMR1_TRGO
TMRx_ITR_4	TMR2_TRGO	TMR2_TRGO
TMRx_ITR_5	TMR4_TRGO	TMR3_TRGO
TMRx_ITR_6	TMR0_OC0	TMR0_OC0
TMRx_ITR_7	TMR1_OC0	TMR1_OC0
TMRx_ITR_8	TMR2_OC0	TMR2_OC0
TMRx_ITR_9	TMR4_OC0	TMR3_OC0
TMRx_ITR_10	TMR9_TRGO	TMR9_TRGO
TMRx_ITR_11	TMR9_OC0	TMR9_OC0
TMRx_ITR_12	TMR9_OC1	TMR9_OC1
TMRx_ITR_13	TMR10_TRGO	TMR10_TRGO
TMRx_ITR_14	TMR10_OC0	TMR10_OC0
TMRx_ITR_15	HRPWM_SYNCO	HRPWM_SYNCO

30.4.2 时基单元

通用定时器的主要模块由一个 32 位计数器及其相关的自动重载寄存器组成，计数器采用递增计数的方式，计数器的时钟可通过预分频器进行分频。

通用定时器的自动重载计数器(TMRx_CNTR)、计数周期寄存器(TMRx_CPR) 、重复计数寄存器(TMRx_CRR)和预分频寄存器(TMRx_PSCR)可通过软件进行读写，即使在计数器运行中也可执行读写操作。

时基单元包含：

- 自动重载计数器(TMRx_CNTR)
- 预分频寄存器(TMRx_PSCR)
- 计数周期寄存器(TMRx_CPR)
- 重复计数寄存器(TMRx_CRR)

其中预分频寄存器(TMRx_PSCR)、计数周期寄存器(TMRx_CPR)、重复计数寄存器(TMRx_CPR)都是可预装载的。对预装载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器立即生效，也可以在每次发生更新事件时更新到影子寄存器，这取决于定时器 TMRx_CR0 控制寄存器中的 ARE 位（自动重载预装载使能位）。

定时器计数器计数到 TMRx_CPR 计数周期寄存器设定的值，且控制寄存器 TMRx_CR0 中的 UDIS 位为 0 时，将产生更新事件。此外，该更新事件也可通过软件发起，通过对定时器事件生成寄存器 TMRx_UGR 中的 UG 位置 1。

更新事件相关的详细介绍，请参看“30.4.12.2 更新事件”章节。

定时器的计数器由时钟源经过预分频器分频后提供的时钟驱动，且仅当 TMRx_CR0 控寄存器中的 CEN 位(计数器使能)置 1 时，才会启动计数器计数。

注意：计数器将在 TMRx_CR0 控制寄存器中的位 CEN(计数器使能)置 1 时刻后的一个时钟周期开始计数。

预分频器说明

预分频器用于对计数器的计数时钟频率进行分频，分频系数介于 1 和 65536 之间。通过配置预分频器寄存器(TMRx_PSCR)来设定预分频的分频系数，计数器计数频率的计算公式如下：

$$f_{CLK\_CNT} = f_{CLK\_SRC} / (PSC + 1)$$

该寄存器具有预加载缓存功能，当启动预加载功能后，可以对预分频器进行实时修改，而新的预分频系数将在下一个更新事件发生时被采用，也可通过软件设置 TMRx_UGR 寄存器中的 UG 位，立即产生一个更新事件（UEV）。

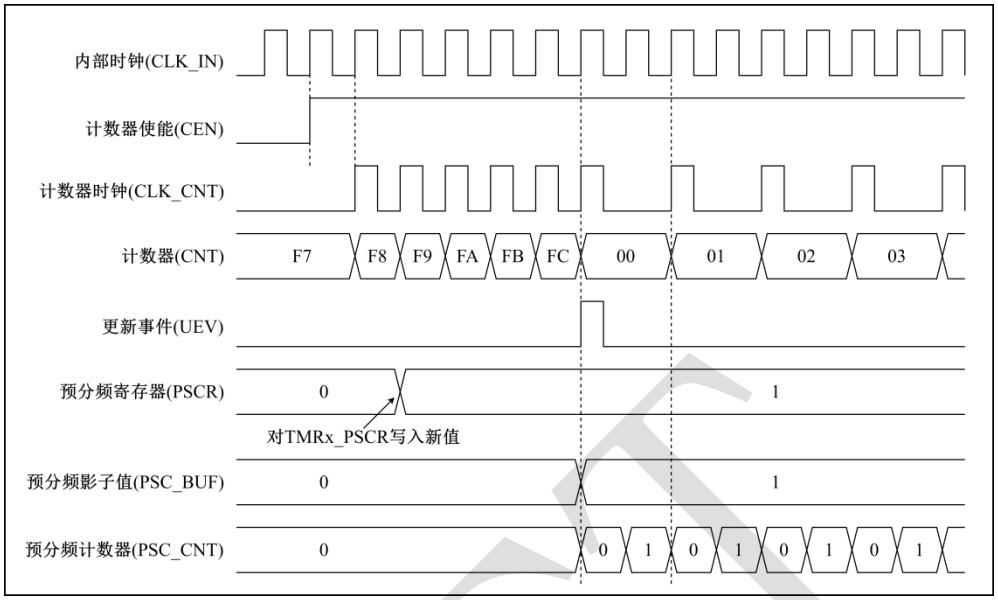


图 30-2　实时修改预分频器的分频值从 1 变为 2 的时序图

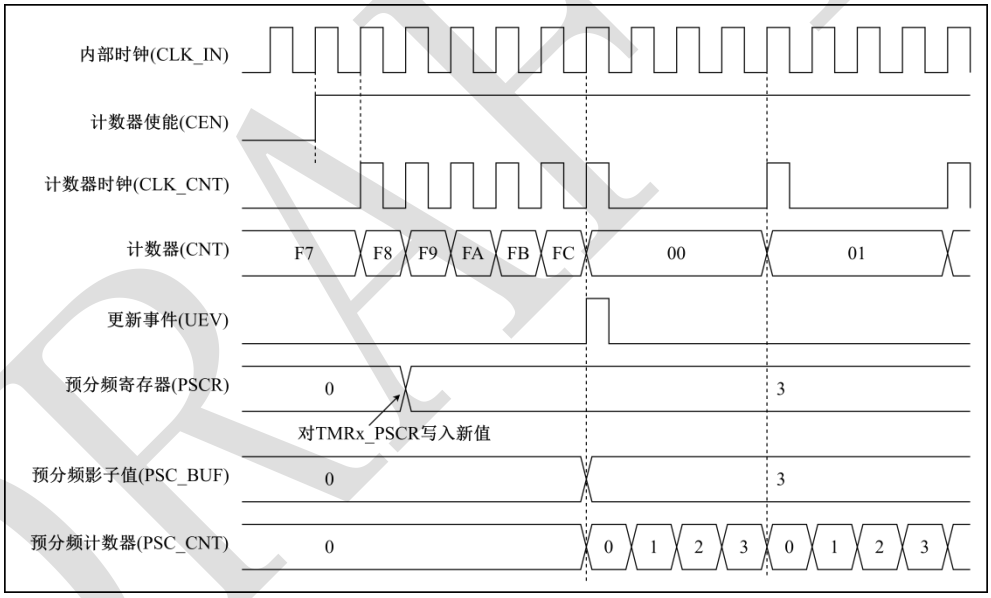


图 30-3　实时修改预分频器的分频值从 1 变为 4 的时序图

30.4.3　计数器模式

递增计数模式

通用定时器支持向上递增计数模式，计数器从 0 开始计数到计数周期值(TMRx_CPR)，支持单次和连续计数模式：单次模式下，计数完成后将自动关闭计数器使能(CEN 自动置 0)，即停止计数；连续模式下，每次计数完成后，自动从 0 重新开始计数，并生成计数器上溢事件 OVF 和更新事件 UEV(如果使能更新事件)。

每次发生计数器上溢时会生成更新事件，或将 TMRx_UGR 寄存器中的 UG 位置 1（通过软件）也可以生成更新事件。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过在上溢事件 OVEV 后，计数器重新从 0x0 开始计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器等于 0x36 时，不同时钟频率下表现的行为。

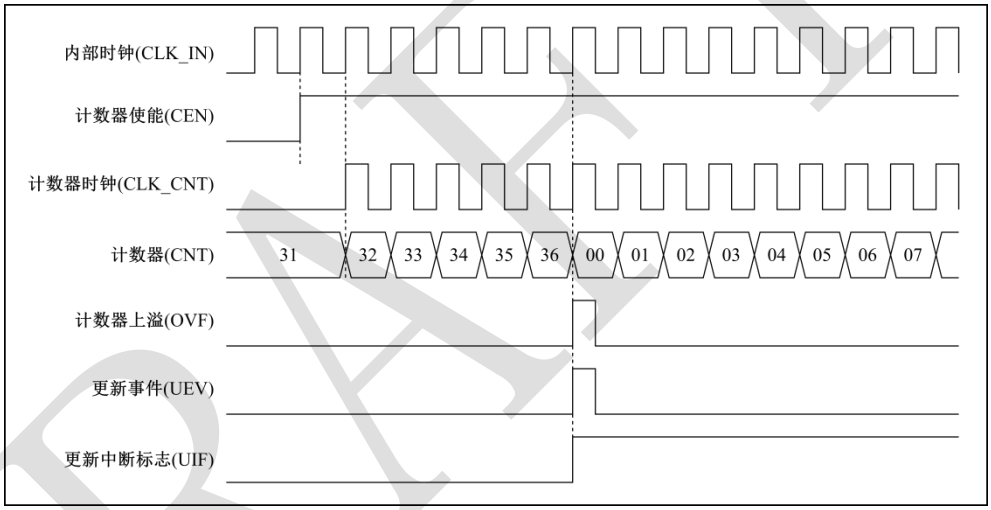


图 30-4　计数器时序图，1 分频内部时钟

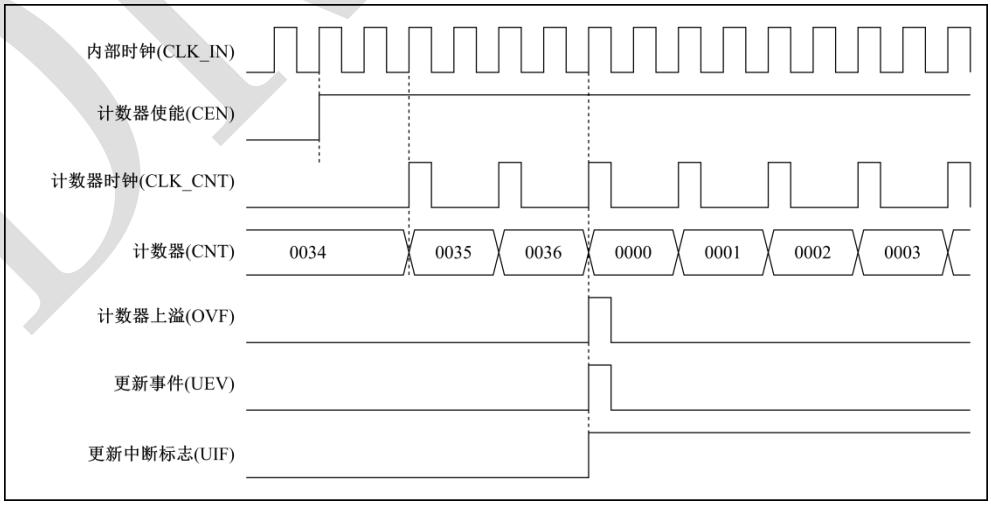


图 30-5　计数器时序图，2 分频内部时钟

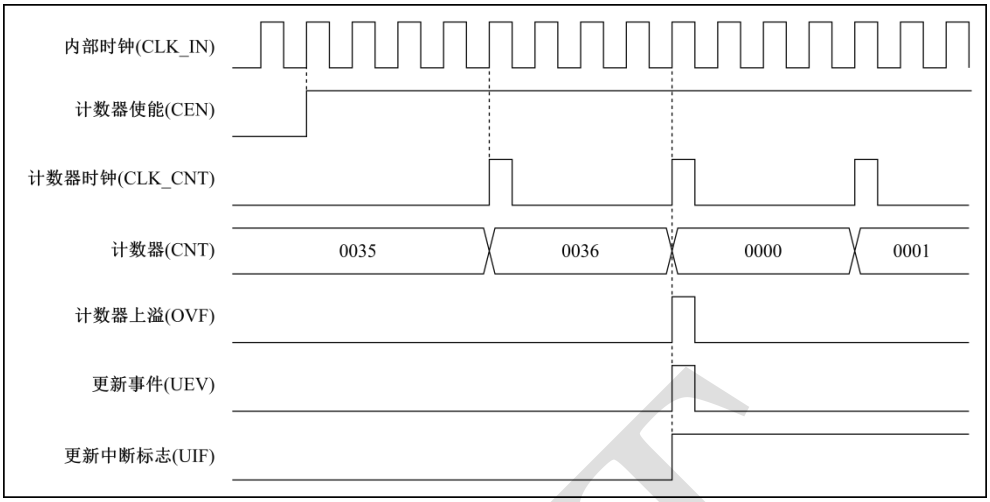


图 30-6　计数器时序图，4 分频内部时钟

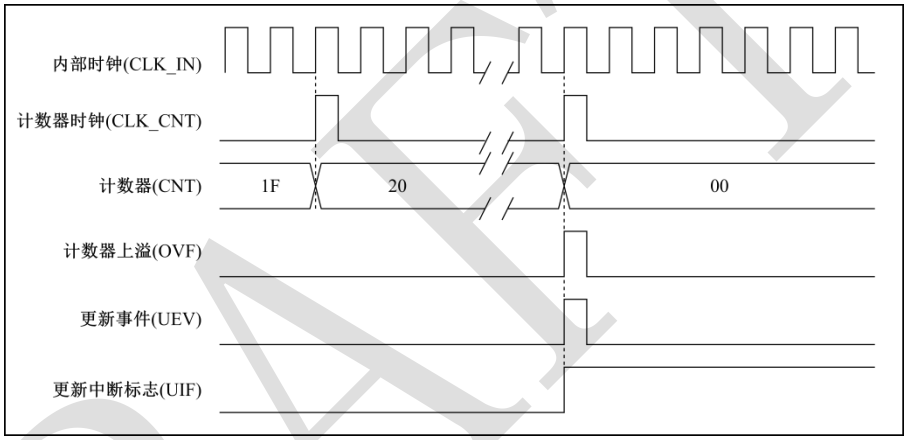


图 30-7　计数器时序图，N 分频内部时钟

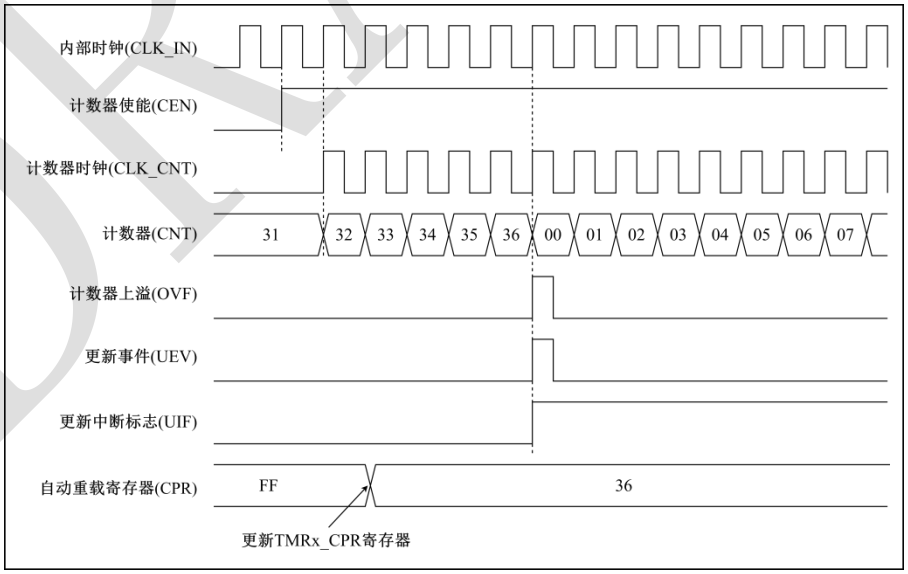


图 30-8　计数器时序图，运行中更新重载值（自动重载使能关闭-实时生效）

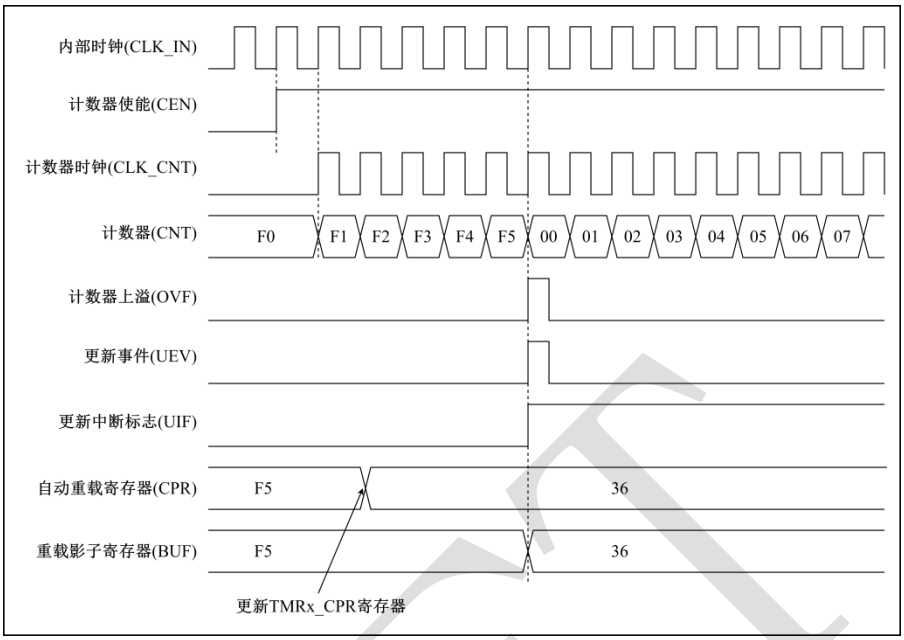


图 30-9　计数器时序图，运行中更新重载值（自动重载使能使能-更新事件生效）

递减计数模式

通用定时器支持向下递减计数模式，计数器从计数周期值(TMRx_CPR)开始递减到 0，支持单次和连续计数模式：单次模式下，计数完成后将自动关闭计数器使能(CEN 自动置 0)，即停止计数；连续模式下，每次计数完成后，自动从计数周期值(TMRx_CPR)重新开始递减计数，并生成计数器下溢事件 OVEV 和更新事件 UEV(如果使能更新事件)。

每次计数器发生下溢时会生成更新事件，或将 TMRx_UGR 寄存器中的 UG 位置 1（通过软件）也可以生成更新事件。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过计数器及预分频计数器仍然能够在下溢事件 OVEV 后，重新从 0x0 开始计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器等于 0x36 时，不同时钟频率下表现的行为。

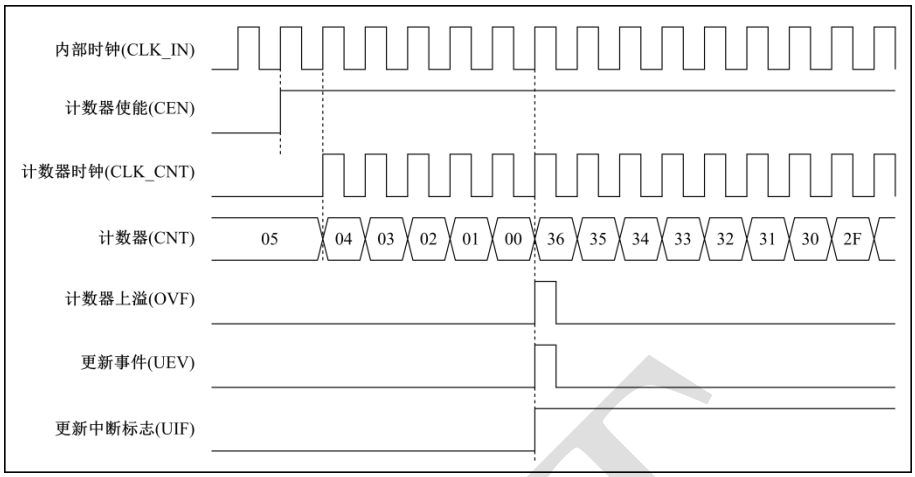


图 30-10　计数器时序图，1 分频内部时钟

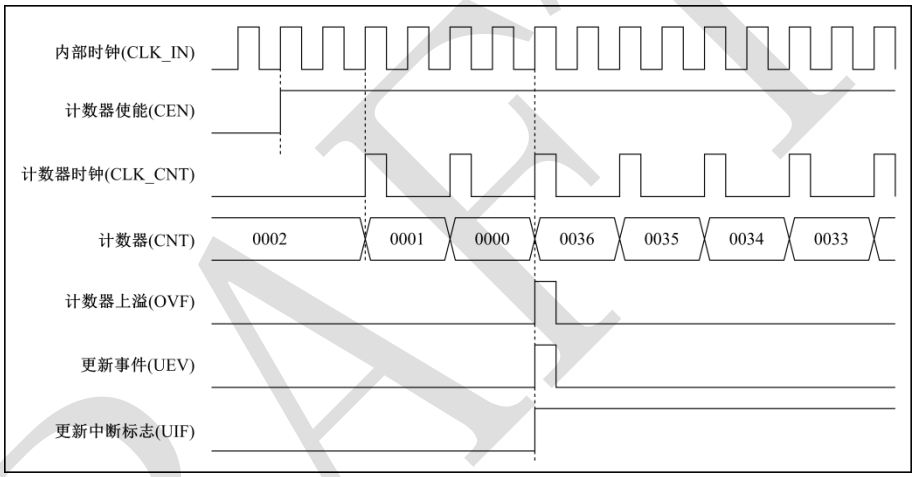


图 30-11　计数器时序图，2 分频内部时钟

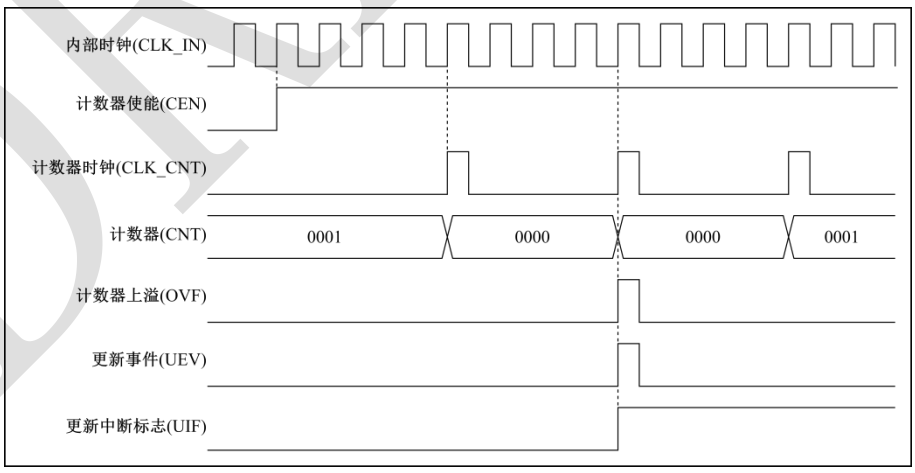


图 30-12　计数器时序图，4 分频内部时钟

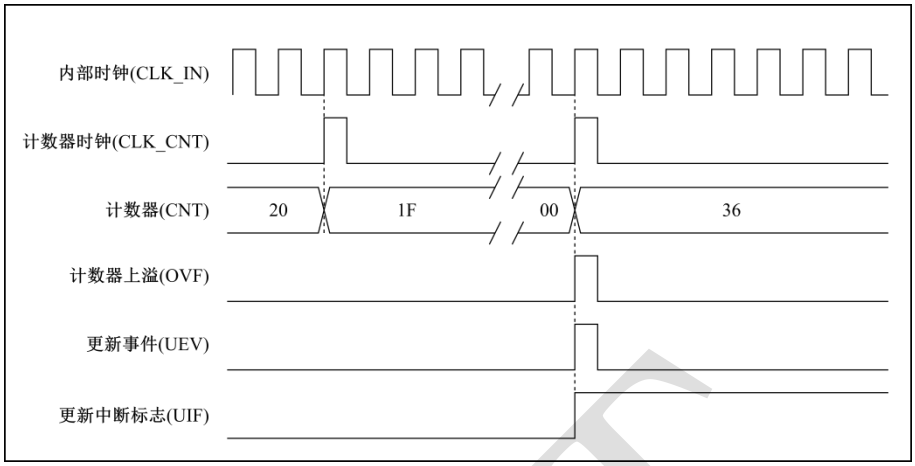


图 30-13　计数器时序图，N 分频内部时钟

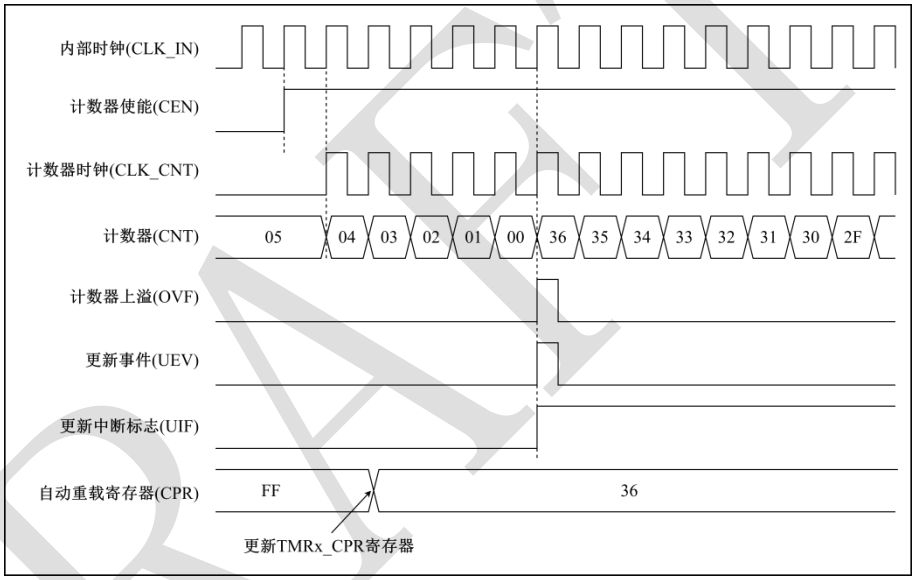


图 30-14　计数器时序图，运行中更新重载值（自动重载使能关闭-实时生效）

递增/递减计数模式

在中心对齐模式下，计数器从 0 开始计数到计数周期值(TMRx_CPR)–1，生成计数器上溢事件；然后从计数周期值(TMRx_CPR)开始向下计数到 1 并生成计数器下溢事件，之后又从 0 开始重新计数。

用户可以通过配置 TMRx_CR0 寄存器中的 CMS 位来选择中心对齐模式。在通道配置为输出模式时，比较输出的中断标志位会在下面各种模式下置 1：

- 仅在计数器递减计数（中心对齐模式 1，CMS=“01”）
- 仅在计数器递增计数（中心对齐模式 2，CMS =“10”）
- 在计数器递增/递减计数（中心对齐模式 3，CMS =“11”）

在中心对齐模式下，将无法修改方向位（TMRx_CR0 寄存器中的 DIR 位），该位是由硬件更新并指示当前计数器的方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TMRx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数

器将重新从 0 开始计数。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过计数器及预分频计数器仍然能够在上溢事件 OVEV 后，计数器会根据当前重载周期值进行递增和递减计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器在不同时钟频率下表现的行为。

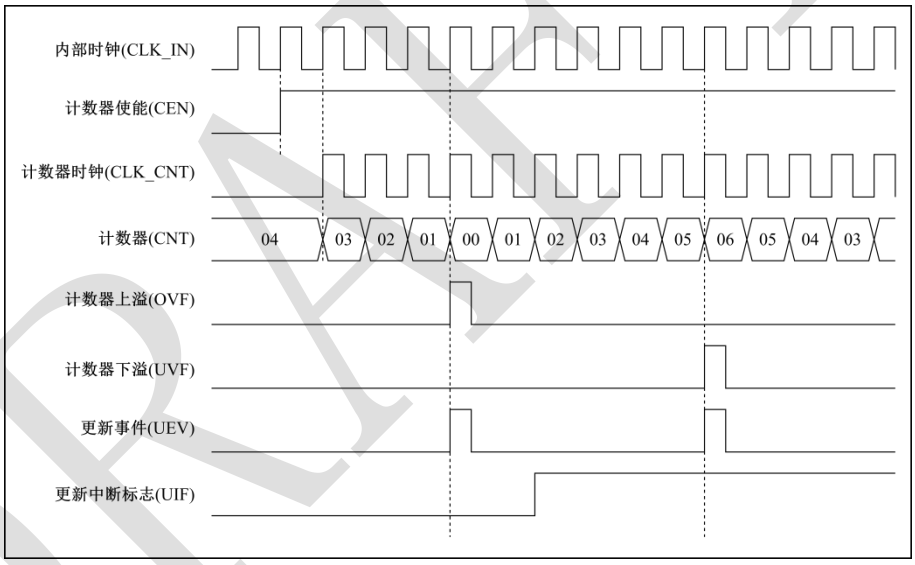


图 30-15　计数器时序图，1 分频内部时钟

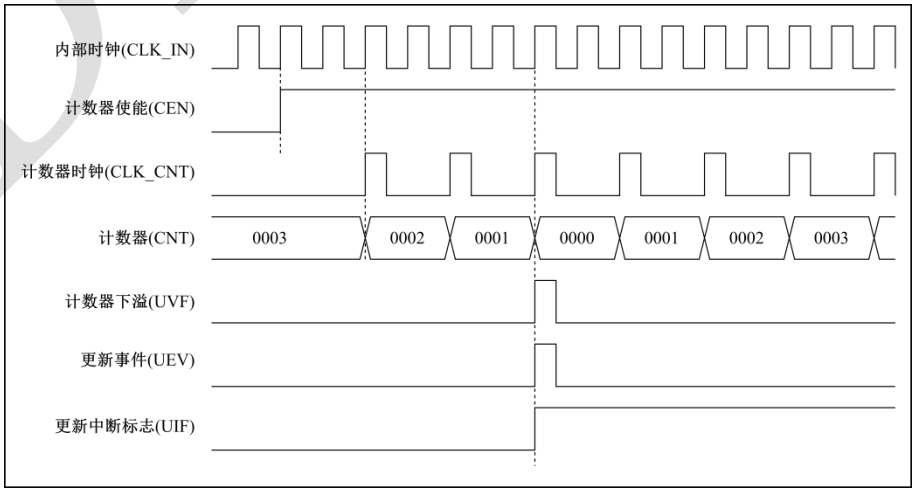


图 30-16　计数器时序图，2 分频内部时钟

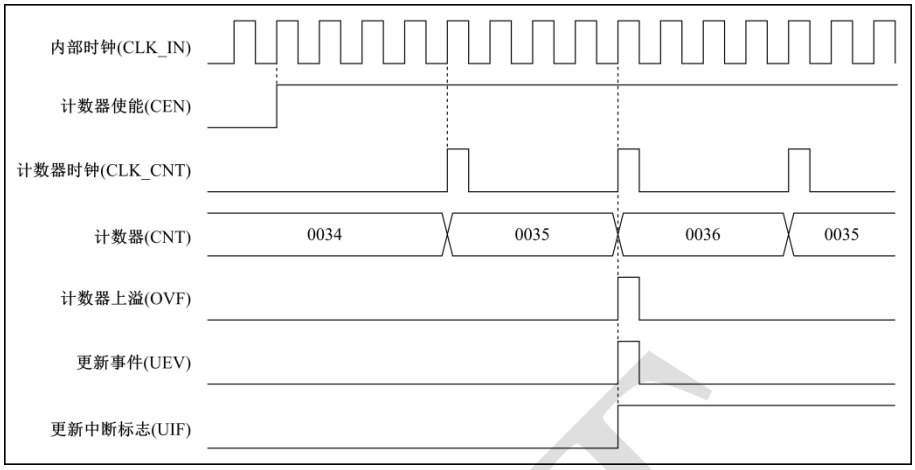


图 30-17　计数器时序图，4 分频内部时钟

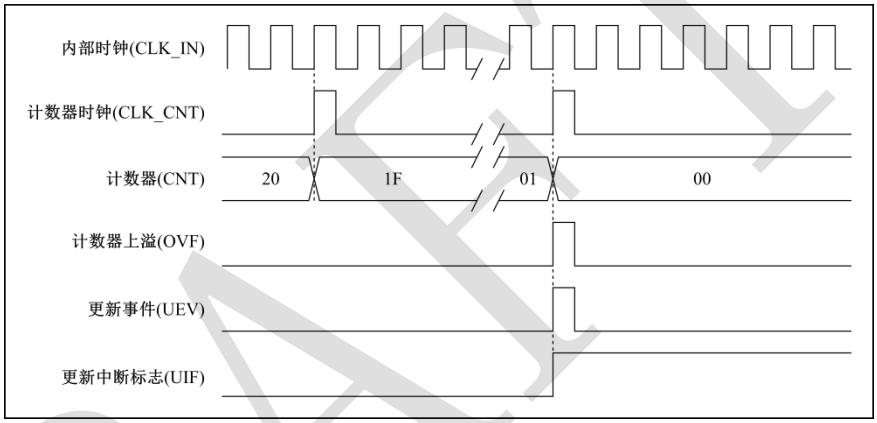


图 30-18　计数器时序图，N 分频内部时钟

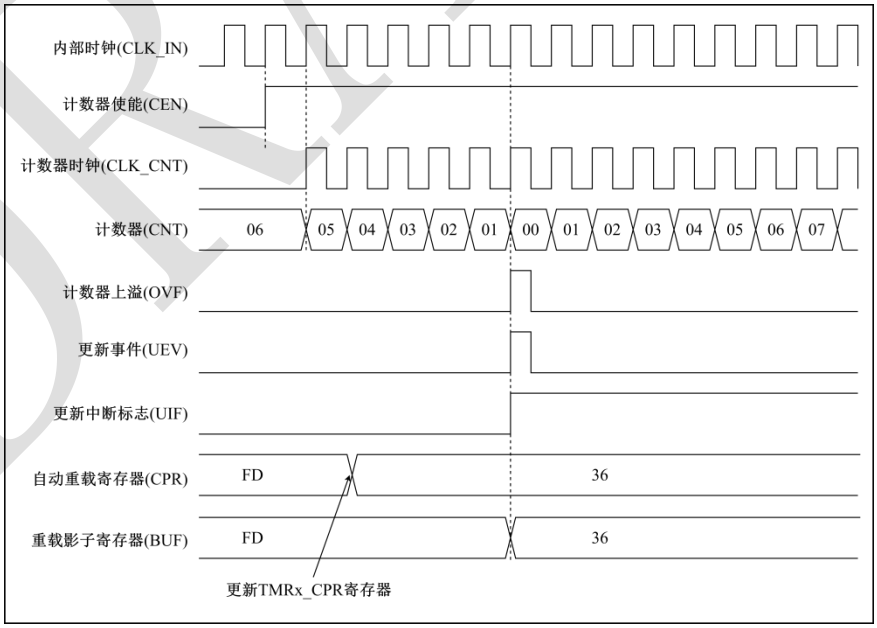


图 30-19 计数器时序图，运行中更新重载值（重载已使能-更新事件生效-计数器下溢）

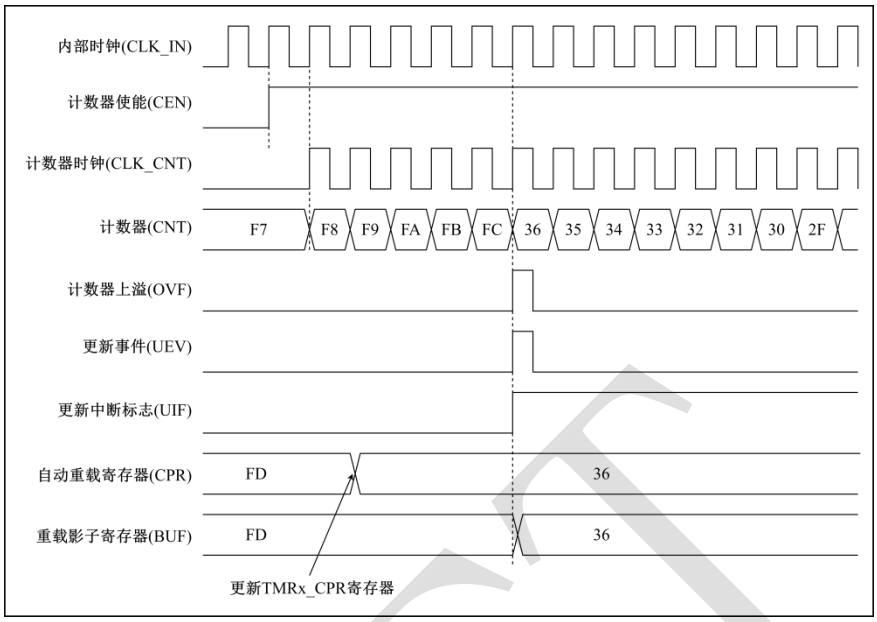


图 30-20　计数器时序图，运行中更新重载值（重载已使能-更新事件生效-计数器上溢）

30.4.4　重复计数器

在“1.4.2 章节：时基单元”中，介绍了计数器上溢时会生成更新事件（UEV）。当引入重复计数功能后，只有当重复计数器达到零时，才会生成更新事件（UEV），其对 PWM 信号的生成很有用。

这意味着，仅当定时器发生 N+1 个计数器上溢（N 为 TMRx_CRR 重复计数寄存器中的值）才生成一次更新事件（UEV），将触发重复计数影子寄存器（从 TMRx_CRR 加载）、计数周期影子寄存器（从 TMRx_CPR 加载）及预分频器影子寄存器（从 TMRx_PSCR 加载）的自动重载。

重复计数器在下列情况下递减：

- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中心对齐模式下每个计数器上溢和计数器下溢。

重复计数器是自动重载类型，其重复加载频率为 TMRx_CRR 寄存器中定义的值（请参见下图）。当更新事件由软件（通过将 TMRx_UGR 寄存器的 UG 位置 1）或硬件生成（通过从模式控制器）时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TMRx_CRR 寄存器的值。

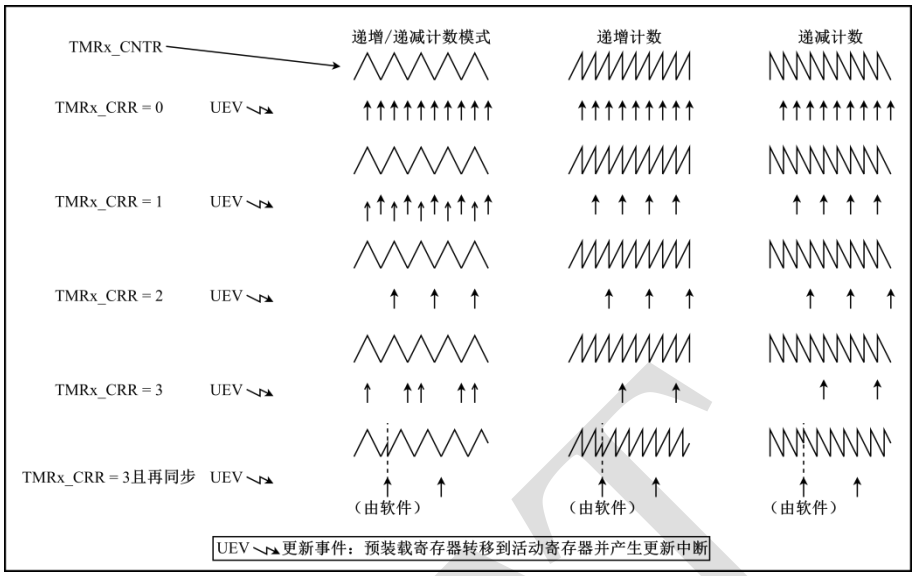


图 30-21　TMRx_CRR 重复计数器设置下的更新频率示例

30.4.5　时钟选择

定时器的计数器时钟源分为以下几类：

- 内部时钟源（CLK_IN）：系统时钟上升沿
- 外部时钟模式：通过外部输入引脚（TIx）
- 内部触发输入（ITRx）：连接其他定时器的触发输出

内部时钟源（CLK_IN）

内部时钟源是 TMRx 定时器的默认时钟源。

将 TMRx_SCR 寄存器中的 SMS 位置 0x0，则禁止从模式控制器，定时器默认选择内部时钟源作为计数时钟。然后，将 TMRx_CR0 寄存器中的 CEN 位和 TMRx_UGR 寄存器中的 UG 位用作控制位，当对 CEN 位写 1 时，预分频器的时钟就由内部时钟 CLK_IN 提供；通过软件对 UG 位写 1，则会使预分频器清零并重新开始计数。

下图为正常模式下内部控制逻辑及计数器计数的行为（预分频设置为 0，即在没有预分频情况下）。

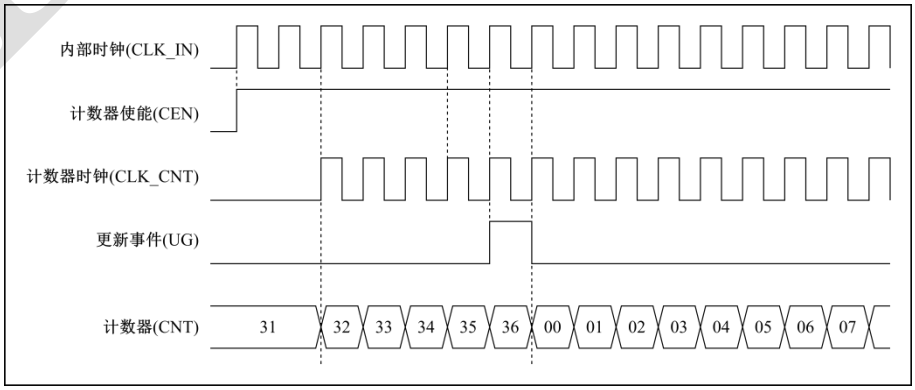


图 30-22　正常模式下的控制时序图（没有预分频情况）

外部时钟源模式 1

当 TMRx_SCR 寄存器中的 SMS 位为“111”时，则选择外部时钟源模式。在该模式下，计数器将在选定输入信号的上升沿计数。

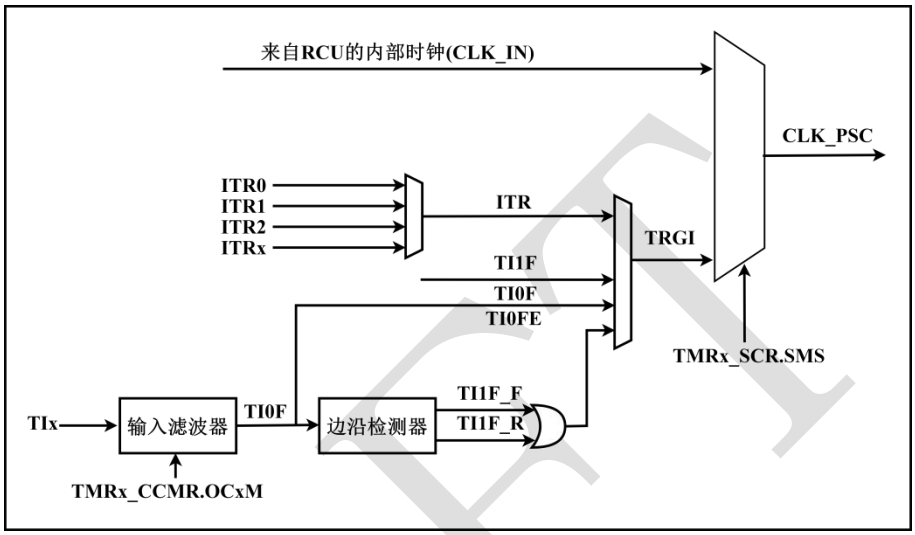


图 30-23　外部时钟连接示意图 1

定时器可选取外部时钟源作为计数器的时钟源，外部时钟源可来源于外部输入引脚或内部触发输入，外部输入引脚先经过内部滤波电路，然后通过内部边沿检测电路后作为计数器的触发时钟。

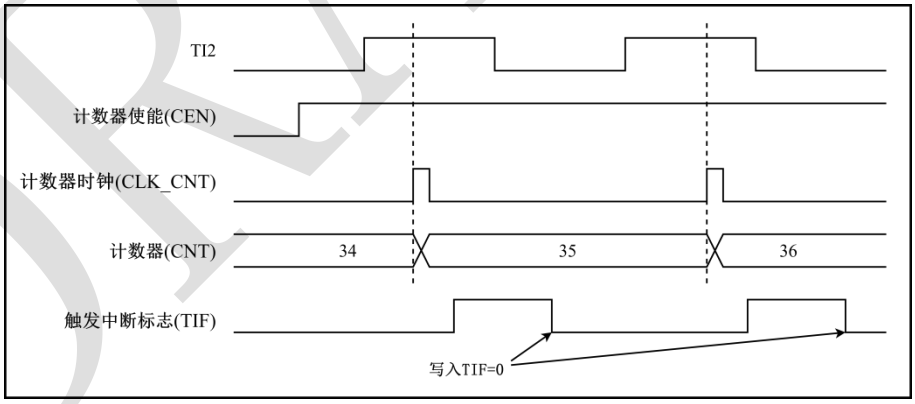


图 30-24　外部时钟模式下的控制电路

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 触发标志位置 1，TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入经过同步电路引起的。

外部时钟源模式 2

通过将 TMRx_SCR 寄存器中的 EE 位置 1 来选择外部时钟源模式 2，在这种模式下，计数器在外部输入 ETR 的边沿进行计数，通过 TMRx_SCR 寄存器中的 EMS 位来设定触发边沿，分为以下几种模式：

- 上升沿计数
- 下降沿计数
- 双边沿计数

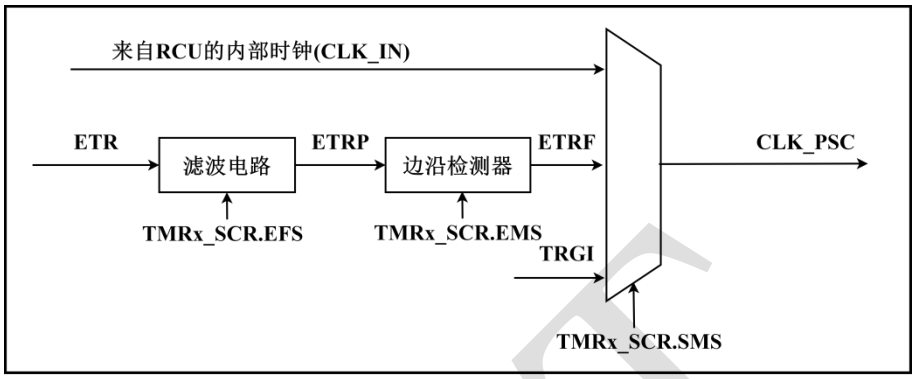


图 30-25　外部时钟连接示意图 2

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的，如下图所示。

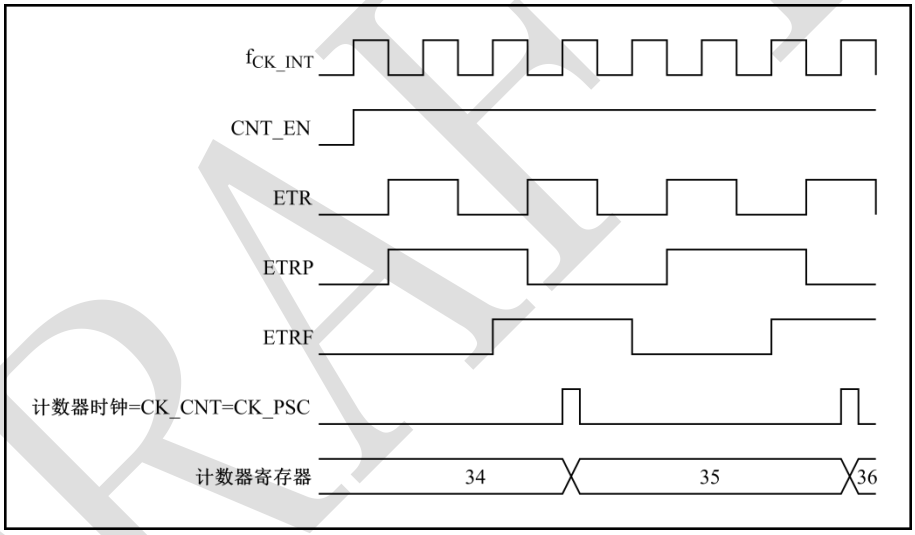


图 30-26　外部时钟模式 2 下的控制电路

注意：外部输入时钟源从功能引脚输入，可通过 GPIO 滤波器或 ETR 自有滤波器来实现消抖，GPIO 滤波器的消抖使能在 GPIO 模块中的“去抖使能寄存器(GPIOx_DER)”中设置。开启滤波器功能后，只有当输入电平宽度达到至少 4 个消抖时钟周期才会被送出，否则会被当作信号抖动被滤波器滤掉。

30.4.6　捕获/比较通道

通用定时器的每个捕获/比较通道均基于一个捕获/比较寄存器(包括影子寄存器)、一个捕获输入(滤波器、极性边沿检测、多路复用和预分频器)和一个比较输出(比较器和输出控制)组成。

输入捕获是对相应的输入信号 TIx 进行采样，经过滤波器后输出 TIxF，然后，再经过极性选择、边沿检测生成一个触发信号（TIxFE），该信号作为从模式控制器的输入，也可用作捕获信号输入。该信号先经过预分频电路（ICxPS），然后进入到捕获寄存器，如下图。

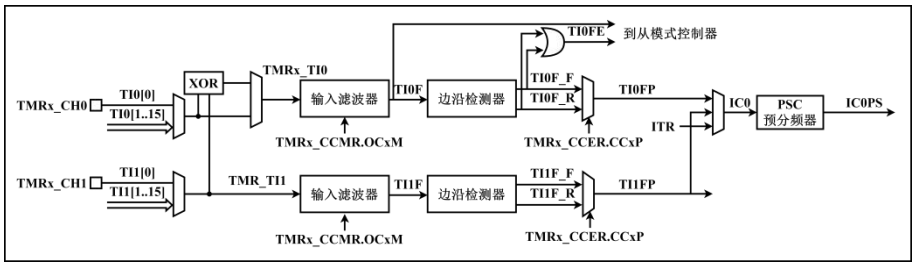


图 30-27　输入捕获通道示意图

输入捕获用于在捕获到指定边沿发生时，锁存计数器的准确数值。通过这种机制可实现对外部信号的准确捕获和测量，由此可实现对信号的脉宽测量、周期测量及占空比计算等功能。

输出比较是利用计数器计数，并与预先配置好的比较值进行对比，当计数值与比较值匹配时，根据所设定的比较输出模式进行相应的波形输出，从而实现例如 PWM、反转、强制极性输出等功能。

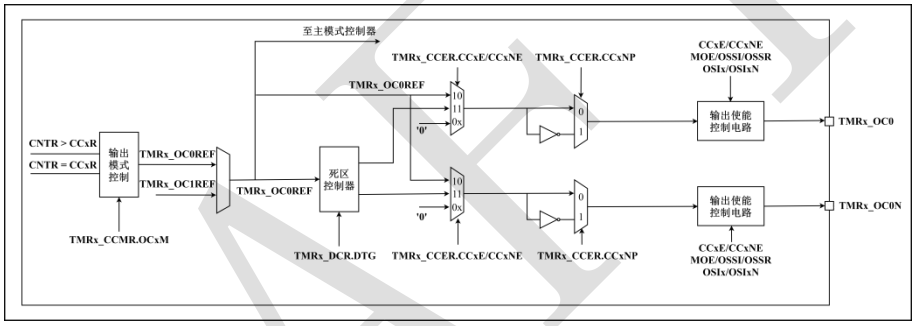


图 30-28　比较输出通道示意图

- 在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

30.4.7　输入捕获

在输入捕获模式下，当在对应的 ICx 信号上检测到边沿跳变后，会将计数器的值锁存到捕获/比较寄存器(TMRx_CCxR)中。当发生捕获事件时，会将 TMRx_SR 寄存器中相应的 CCxMIF 标志位置 1，并触发定时器中断(如果对应中断已使能)。如果在发生捕获事件时 CCxMIF 标志位已经被置位，则会将 TMRx_SR 寄存器中相应的 CCxOIF 重复捕获标志位置 1。可通过软件对捕获标志位 CCxMIF 及重复捕获标志位 CCxOIF 写 1 来清零，或读取存储在 TMRx_CCxR 寄存器中的捕获数据时。

通过配置 TMRx_CCMR 寄存器中的 CCxS 位来选择捕获输入通道，并将 TMRx_CCER 寄存器中的 CCxE 位置 1，就可以使定时器工作在输入捕获模式下。

通过配置 TMRx_CCER 寄存器中的 CCxP 位来选择捕获模式下的触发边沿，配置 TMRx_CIR 寄存器的 CxTIS 位可选择捕获的引脚输入源或比较器模块(CMPx_OUT)的内部输出信号。

当定时器工作于输入捕获模式下，在检测到捕获输入上产生边沿跳变时，定时器会将当前计数器的值锁存捕获到捕获/比较寄存器(TMRx_CCxR)内，同时将 TMRx_SR 寄存器中的输入捕

获中断标志位(CCxMIF)置 1，并触发定时器中断(如果对应中断已使能)，软件上需要及时清除输入捕获中断标志 (对 TMRx_SR 寄存器中的 CCxMIF 写 1，或读取 TMRx_CCxR 寄存器，硬件将自动清除标志位)。如果在清除该标志位之前，再次捕获到输入跳变沿，会将新的计数值锁存并覆盖之前的值，同时会将 TMRx_SR 寄存器中的输入重复捕获标志位(CCxOIF)置 1，并触发定时器中断(如果对应中断已使能)。因此在处理输入捕获时，建议在捕获状态标志位置起后应当及时读取数据，否则一旦发生重复捕获，新捕获的数据将覆盖之前的数据，导致捕获信息丢失。

30.4.8 比较输出

配置 TMRx_CCMR 寄存器中的 CCxS 位来选择比较输出模式，配置 TMRx_CCER 寄存器中的 CCxE 位来使能比较输出功能，使定时器工作在输出比较模式下。

配置 TMRx_CCMR 寄存器中的 OCxM 位来选择比较输出的工作模式，配置 TMRx_CCMR 寄存器中的 CCxPE 位来使能比较输出的预装载功能，配置 TMRx_CCER 寄存器中的 CCxP 位来选择比较模式下有效电平的极性。

当定时器的计数器(TMRx_CNTR)值与捕获/比较寄存器(TMRx_CCxR)值相匹配时，根据比较输出的工作模式在引脚上输出相应的波形，同时会将 TMRx_SR 寄存器中的 CCxMIF 位置 1，并触发定时器中断(如果对应中断已使能)，软件上可通过对 CCxMIF 位写 1 进行清除。

输出工作模式

定时器支持的比较输出工作模式有以下几种，通过 TMRx_CCMR 寄存器中的 OCxM 位进行配置：

- 0000：冻结，输出比较寄存器与计数器进行比较不会对输出造成任何影响（保持原有状态）
- 0001：当计数器与捕获/ 比较寄存器匹配时，输出有效电平（高电平）
- 0010：当计数器与捕获/ 比较寄存器匹配时，输出无效电平（低电平）
- 0011：当计数器与捕获/ 比较寄存器匹配时，发生翻转
- 0100：强制变为无效电平（低电平）
- 0101：强制变为有效电平（高电平）
- 0110：PWM 模式 1 - 当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态（高有效）
- 0111：PWM 模式 2 - 当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态（高有效）
- 1000：可再触发 OPM 模式 1 - 在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。
- 1001：可再触发 OPM 模式 2 - 在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到

触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。

- 1100：组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同，参考是 OC1REF 或 OC2REF 的结果
- 1101：组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同，参考是 OC1REF 和 OC2REF 的结果。
- 1110：不对称 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。
- 1111：不对称 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

配合定时器中的单次/连续计数模式，还可实现更多组合功能，例如利用单次计数模式+输出比较 PWM 模式可实现单脉冲输出功能，更多特别模式说明将在后续章节进行更详细介绍。

自动装载功能

在比较输出模式中，软件可通过修改捕获/比较寄存器(TMRx_CCxR)来调整比较值，通过配置 TMRx_CCMR 寄存器中的 OCxPE 位来使能比较预装载功能，自动装载的作用如下：

- 不开启预装载功能，向 TMRx_CCxR 寄存器写入新值，将立即更新到内部影子寄存器并生效。
- 开启预装载功能，向 TMRx_CCxR 寄存器写入新值，并不会立即生效，而是在下一个更新事件之后，再将 TMRx_CCxR 寄存器值更新到影子寄存器并生效。更详细的捕获/比较事件描述，请参考“30.4.12.4 捕获/比较事件”章节说明。

30.4.8.1 强制输出模式

在比较输出模式（TMRx_CCMR 寄存器中的 CCxS 位等于 0x0）下，可直接通过软件将比较输出信号（OCxREF）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的比较结果。

配置强制输出模式，只需将 TMRx_CCMR 寄存器中的 OCxM 位配置为 100 或 101，如对该位写入 101 后，OCxREF 将强制为高电平（OCxREF 始终为高电平有效），而输出 OCx 会根据 CCxP 极性位选择合适的有效电平。

在强制输出模式下，虽然定时器引脚输出固定电平，但定时器的计数器值与比较值的匹配行为依然在执行，在匹配时依然会引发相应的标志位并触发中断。

- 强制输出无效电平 (OCxM=100)时：
  - CCxP=0（有效电平为高电平），将比较强制输出为低电平
  - CCxP=1（有效电平为低电平），将比较强制输出为高电平
- 强制输出有效电平 (OCxM=101)时：
  - CCxP=0（有效电平为高电平），将比较强制输出为高电平
  - CCxP=1（有效电平为低电平），将比较强制输出为低电平

30.4.8.2 匹配输出模式

匹配输出模式可用于控制输出波形，或指示时间周期的变化。

输出引脚的电平由匹配输出模式（TMRx_CCMR 寄存器中的 OCxM 位）和输出极性（TMRx_CCER 寄存器中的 CCxP 位）定义。当定时器的计数器值与比较值相匹配时，输出比较匹配功能：

- 输出引脚即可保持原有电平（OCxM=0000）
- 配置为有效电平（OCxM=0001）
- 配置为无效电平（OCxM=0010）
- 配置为进行翻转（OCxM=0011）

匹配时，会将 TMRx_SR 中断状态寄存器中的 CCxMIF 置位，并触发定时器中断(如果对应中断已使能)。

通过配置 TMRx_CCMR 寄存器中的 CCxPE 位，可选择 TMRx_CCxR 寄存器的自动加载功能的开启。匹配输出模式也可用作为单脉冲输出（在单脉冲模式下）。

30.4.8.3 PWM 输出模式

比较输出的 PWM 模式，可以用于生成一个调制信号，该信号频率周期由 TMRx_CPR 周期值寄存器决定，而占空比则由捕获/比较寄存器(TMRx_CCxR)设定的值决定，比较输出的 PWM 模式有两种：

- PWM 模式 1（OCM=0110）
  当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态（高有效）。
- PWM 模式 2（OCM=0111）
  当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态（高有效）。

OCx 极性可通过 TMRx_CCER 寄存器的 CCxP 位来设置，既可以设为高电平有效，也可以设为低电平有效，OCx 输出通过将 TMRx_CCER 寄存器中的 CCxE 位置 1 来使能，有关其详细信息，请参考 TMRx_CCER 寄存器说明。

因为计数器采用的递增方式计数，所以定时器为边沿对齐模式下生成 PWM。

边沿对齐模式

下图以 PWM 模式 1 为例，只要 TMRx_CNTR 计数器值小于 TMRx_CCxR 比较值，PWM 参考信号 OCxREF 为高电平，否则为低电平。如果 TMRx_CCxR 比较值大于 TMRx_CPR 重载周期值，则 PWM 参考信号 OCxREF 为高电平。如果 TMRx_CCxR 比较值为 0，则 PWM 参考信号 OCxREF 保持低电平。下图为边沿对齐的 PWM 波形，其中 TMRx_CPR 值为 8。

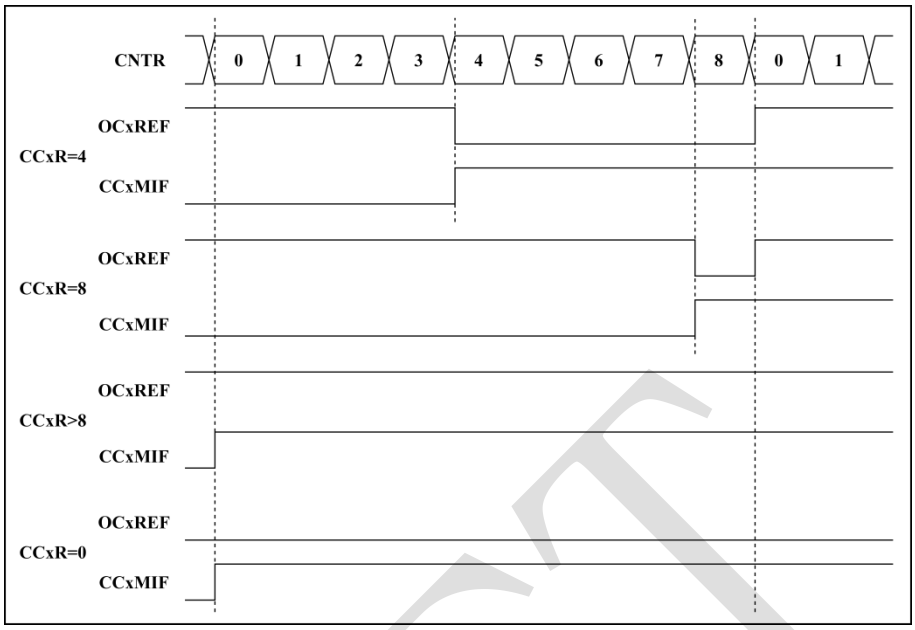


图 30-29　PWM 输出（CPR=8，CCxR=4，CCxP=0 高电平有效）

中心对齐模式

通过配置 TMRx_CR0 寄存器中的 CMS 位来选择中心对齐模式，在通道配置为输出模式时，根据 CMS 位的配置，在计数器递减计数、计数器递增计数或计数器同时递增/递减计数时将比较输出的中断标志位置 1，并且 TMRx_CR0 寄存器中的 DIR 方向位由硬件更新，软件不能修改。

下图显示了中心对齐模式下的 PWM 波形，其中，TMRx_CPR=8，PWM 模式配置为 PWM 模式 1，根据 TMRx_CR0 寄存器中的 CMS 位而选择中心对齐模式 1，当计数器递减计数时，比较标志位置 1。

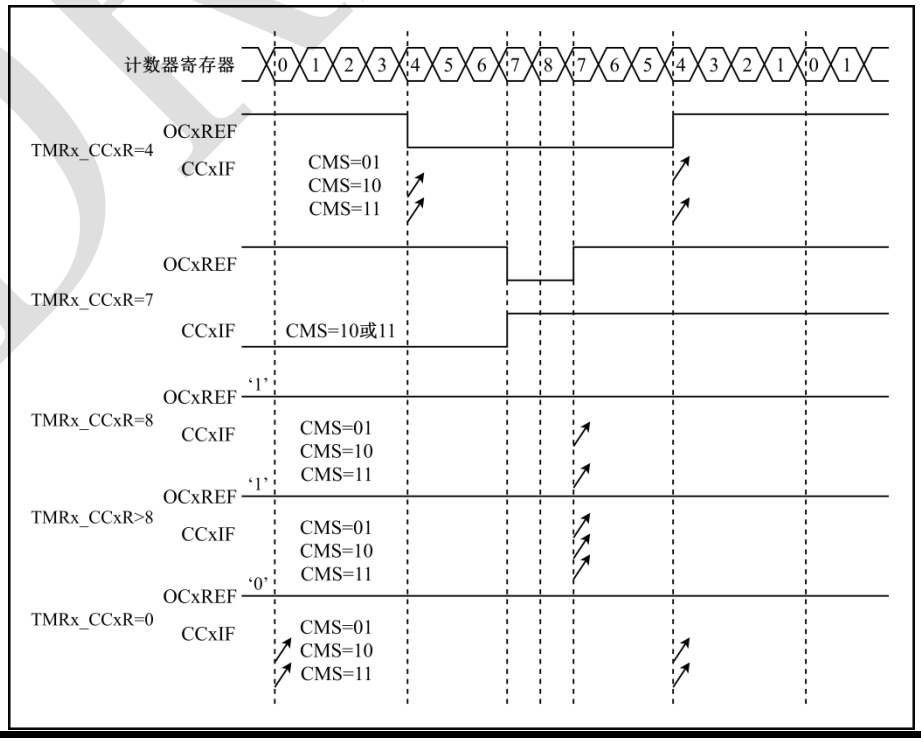


图 30-30　中心对齐模式 PWM 波形（CPR=8）

中心对齐模式使用建议：

- 在中心对齐模式下，TMRx_CR0 寄存器中的 DIR 方向位由硬件更新，软件不能修改。
- 在中心对齐模式运行过程中，不建议对计数器执行写操作，否则将发生意想不到的结果，尤其是：

  - 如果写入计数器中的值大于计数周期重载值 (TMRx_CNTR>TMRx_CPR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TMRx_CPR 的值，计数方向会更新。

30.4.8.4　单脉冲模式

单脉冲模式(One-Pulse Mode)是单次计数模式+PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号下启动，并在一段可编程的延时后输出一个可编程宽度的单脉冲信号。

可通过从模式控制器启动计数器，在匹配输出模式或 PWM 模式下生成波形，将 TMRx_CR0 寄存器中的 OPM 位置 1，即可使能单脉冲模式。这样，在发生下一个更新事件 UEV 时，计数器将自动停止。

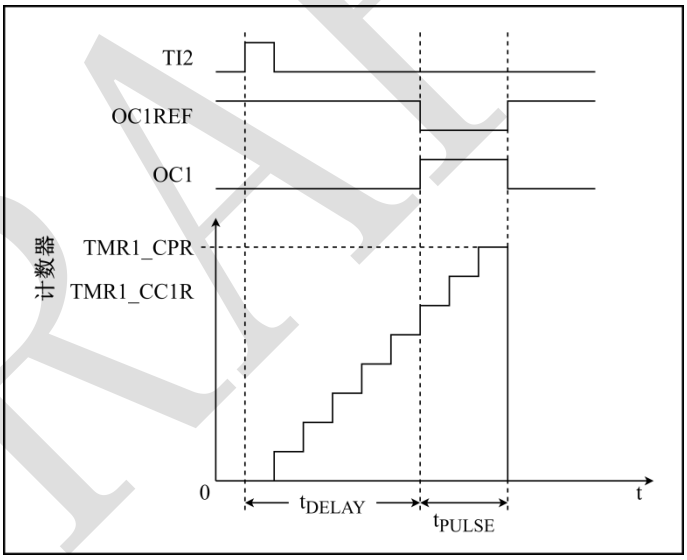


图 30-31　单脉冲模式示例

上图效果为：在 TI2 输入引脚上检测到上升沿时，经过 t_{DELAY} 的延迟后，将在 OC1 上输出一个长度为 t_{PULSE} 的正脉冲波形。

单脉冲的波形可通过调整 TMRx_CCxR 比较寄存器的值来定义（需考虑时钟频率及计数器预分频）：

- t_{DELAY} 时间由写入 TMRx_CCxR 的值来定义
- t_{PULSE} 时间由重载周期值与比较值（TMRx_CPR-TMRx_CCxR）之差来定义

当 TMRx_CR0 寄存器中的 OPM 位写入“1”，即使能单脉冲模式，选择合适的输出工作模式（匹配输出或 PWM 模式 1/2），对捕获/比较寄存器(TMRx_CCxR)写入合适的比较值，根据需求配置有效电平的极性，计数器在发生下一个更新事件（计数器从重载周期值返回到 0）时计数器将停止计数，并输出一段可控宽度的单脉冲。当 TMRx_CR0 寄存器中的 OPM 位写入“0”，

即选择重复连续模式。

30.4.9 从模式

定时器可通过外部触发实现以下功能的同步：复位模式、门控模式和触发模式。

从模式：复位模式

当输入触发信号产生上升沿跳变时，计数器及其预分频器将被重新初始化。如果 TMRx_CR0 寄存器中的 URS 位为 0 时，则还会生成更新事件 UEV。然后，所有预装载寄存器 TMRx_CPR 和 TMRx_CCxR 都将更新。

以下示例中，在 TI1 输入信号上出现上升沿时，递增计数器清零：

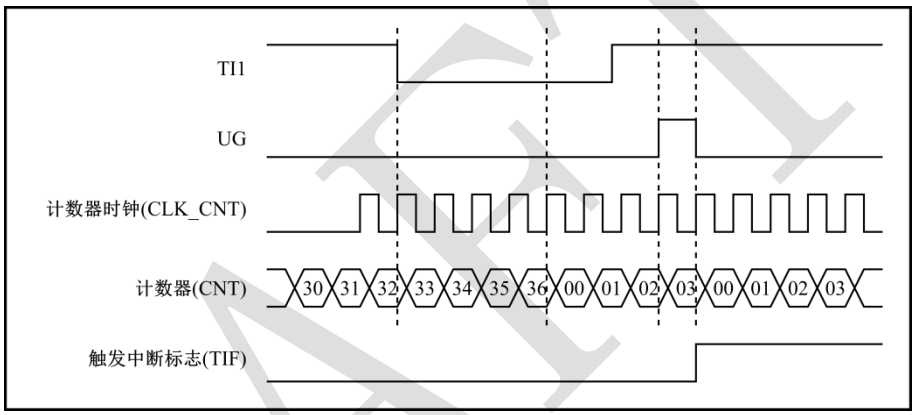


图 30-32　复位模式下的控制电路

计数器使用内部时钟计数，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发 TMRx_SR 寄存器中的 TIF 标准位置 1，使能对应的中断后，还可发送中断请求。

上图显示了自动重载寄存器 TMRx_CPR=0x36 时的行为，TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：门控模式

通过触发输入信号的电平来使能计数器，当输入信号为高电平时，计数器开始计数，当输入信号为低电平时，计数器保持不变。

下图中，递增计数器仅在 TI1 输入为高电平时计数：

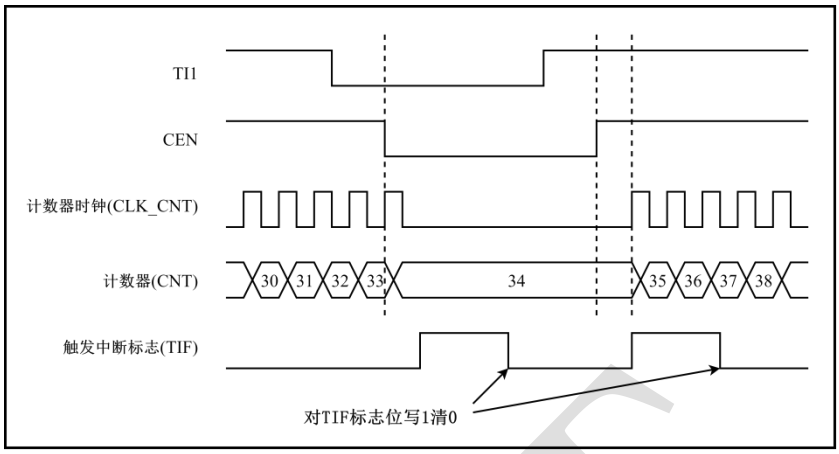


图 30-33　门控模式下的控制电路

只要 TI1 为高电平，计数器就开始根据内部时钟计数，当 TI1 变为低电平时停止计数。当计数器启动或停止时，TMRx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：触发模式

通过输入信号的上升沿来启动计数器，计数器开始计数器。

下图中，递增计数器在 TI2 输入上升沿时，启动递增计数器：

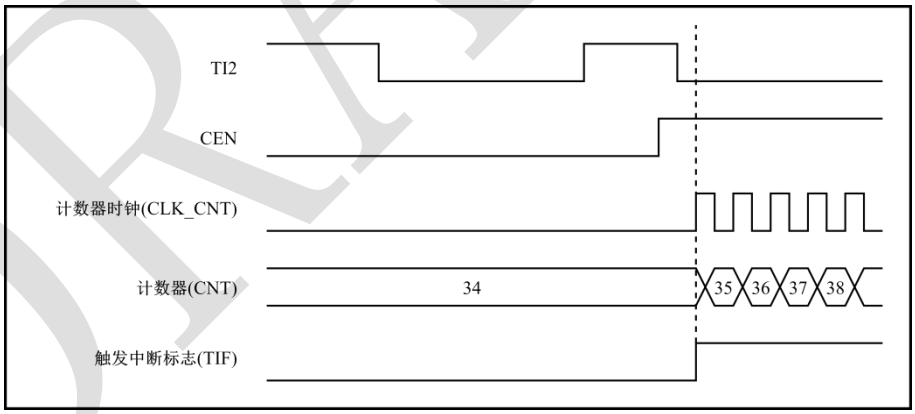


图 30-34　触发模式下的控制电路

当 TI2 出现上升沿时，计数器根据内部时钟开始计数，并且 TIF 标志位置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

30.4.10　定时器同步

定时器内部是连在一起，以实现定时器之间的同步与级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

下图简要介绍了从模式的触发选择及主模式选择框图，将一个定时器用作另一个定时器的预

分频器：

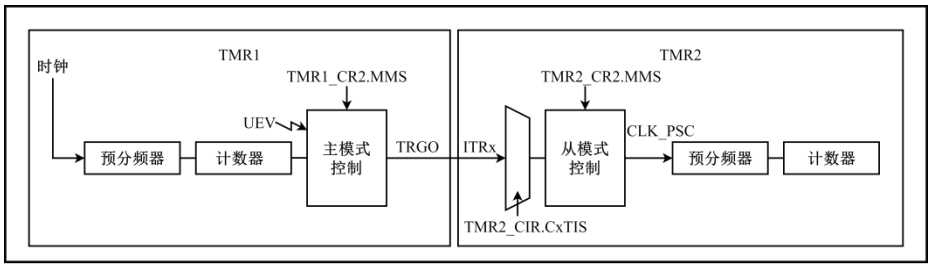


图 30-35　主/从定时器示例

注意：如果选择定时器 1 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动定时器 2 的计数器。

下图使用一个定时器 1 的输出来使能另一个定时器 2 的示例，相关的连接图如下，仅当定时器 1 的 OC1REF 位高电平时，定时器 2 才会根据分频后的时钟进行计数，两个计数器的时钟频率都是基于 3 分频。

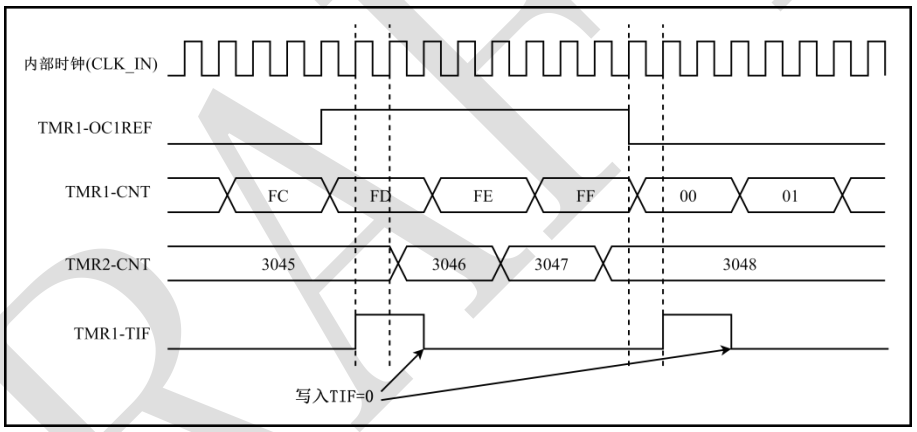


图 30-36　使用定时器 1 的输出来对定时器 2 实施门控控制

30.4.11　调试模式

当内核进入调试模式（Cortex™-M4 内核停止）时，定时器内部计数器会根据 SYSCTRL 模块 SYSDCR 系统调试配置寄存器中的 TMRxDEN 位来选择继续正常工作或者停止工作。

30.4.12　事件与中断

通用定时器在多种情况下会产生相应的事件，事件主要用于更新相应的影子寄存器、执行特定功能以及产生相应中断。

通用定时器内主要有以下几种事件：

- 定时器上溢事件，简称上溢事件 OVEV(Overflow Event)
- 定时器更新事件，简称更新事件 UEV (Update Event)

需要使用更新事件，必须将定时器控制寄存器(TMRx_CR0)中的禁止更新位(UDIS)置 0

- 定时器输入捕获/输出比较事件，简称捕获/比较事件 CCEV (Capture/Compare Event)
- 定时器触发事件，简称触发事件 TEV (Trigger Event)

中断主要包含以下五种中断触发：

- 定时器上溢中断，发生上溢事件后，上溢中断标志位(OVIF)同时被置位
- 定时器更新中断，发生更新事件后，更新中断标志位(UIF)同时被置位
- 定时器触发中断，发生触发事件后，更新触发标志位(TIF)同时被置位

特别说明：如果更新请求源选择位(URS)置 1，即仅上溢事件会产生更新事件，那么软件上如果通过设置更新产生位(UG)，将只会产生更新事件用于更新预加载相关的寄存器，但不会触发定时器更新中断。

- 定时器输入捕获中断，当定时器为输入捕获模式并捕获到相应的边沿时，输入捕获中断标志位（CCxMIF）将被置位。
- 定时器重复捕获中断，当输入捕获中断标志位已经置位(CCxMIF =1)，再次捕获到相应边沿，则输入捕获重复捕获中断标志位（CCxOIF）将被置位。
- 定时器输出比较中断，当定时器位输出比较模式，并且计数值与比较值相匹配时，输出比较中断标志位(CCxMIF)将被置位。

30.4.12.1 上溢事件（Overflow Event）

上溢事件(OVEV)由以下情况产生：

- 计数器 (TMRx_CNTR)值到达计数周期寄存器(TMRx_CPR)设定的值。

上溢事件的作用

计数器到达上溢后，计数器寄存器会立即重新从 0 开始计数 (在连续模式下)。

事件产生后，中断状态寄存器(TMRx_SR)中的上溢中断标志位(OVIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

30.4.12.2 更新事件（Update Event）

更新事件(UEV)可以由以下两种情况产生：

- 计数器上溢：计数器上溢事件产生的同时会产生更新事件。
- 软件置位：软件将定时器事件产生寄存器(TMRx_UGR)中的更新标志位(UG) 置 1。

更新事件使用的注意事项

- 如果需要使用更新事件(UEV)，必须将定时器控制寄存器(TMRx_CR0)中的更新禁止位(UDIS)置 0，使能更新事件功能，否则更新事件将被禁止。
- 如果需要软件置位功能，还必须要配置定时器控制寄存器(TMRx_CR0)中的更新请求位(URS)置 0。否则只有计数器上溢才会产生更新事件。

更新事件的作用

事件产生后，中断状态寄存器(TMRx_SR)中的更新中断标志位(UIF)将会置 1，并且会触发定时器中断 (如果对应的中断已使能)。

特别的，如果在计数器计数的过程中(未达到计数器上溢)，通过软件设置 UG 位产生更新事件后，计数器寄存器也会立即重新从 0 开始计数(在连续模式下)。不同之处在于：通过软件更新标志位(UG)设置产生的更新事件，不会触发上溢事件的中断。

更新事件对自动装载预加载(Auto-Reload Preload)功能的作用

当定时器开启自动装载预加载功能（TMRx_CR0 寄存器内 ARE=1），更新事件产生后，才会将以下几个寄存器的值，更新到内部影子寄存器内使之生效：

- 计数周期寄存器(TMRx_CPR)
- 预分频寄存器(TMRx_PSCR)
- 捕获/比较寄存器(TMRx_CCxR)

注意：软件置位(UG=1)产生的更新事件，不会影响捕获/比较寄存器(TMRx_CCxR)。

如果定时器未开启自动装载预加载功能（TMRx_CR0 寄存器内 ARE=0），则上述寄存器写入新值将立即生效，更新事件将不影响上述寄存器的配置。

30.4.12.3 触发事件（Trigger Event）

触发事件(TEV)由以下情况产生：

- 在从模式下，输入信号上检测到预期的跳变事件
- 在从模式下，软件将事件产生寄存器(TMRx_UGR)中的事件产生标志位(TG)置 1

触发事件的作用

在从模式下，输入信号上检测到预期的跳变事件，中断状态寄存器(TMRx_SR)中的 Trigger 中断标志位(TIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

30.4.12.4 捕获/比较事件（Capture/Compare Event）

捕获/比较事件(CCEV)，可以由以下三种情况产生：

- 比较成功：计数器 (TMRx_CNTR)的值与捕获/比较寄存器(TMRx_CCxR)的值比较成功时(仅针对输出比较模式下)
- 捕获成功：外部输入一个与 MRx_CCER 寄存器中 CCxP 位相符的边沿信号时（仅针对输入捕获模式下）
- 软件产生：将事件产生寄存器(TMRx_UGR)中的事件产生标志位(CCxUG)置 1

捕获/比较事件的作用

在不同模式下捕获/比较事件的作用不相同：

- 对于比较输出模式：
  - 捕获/比较事件产生后，中断状态寄存器(TMRx_SR)中的比较成功中断(CCxMIF)位将会置 1，并触发定时器中断(如果中断已使能)。
  - 如果使能了比较器预加载功能(TMRx_CCMR 寄存器中的 CCxPE 位置 1)，事件产生后(特指比较成功所产生事件)，定时器将会执行以下动作：
    - 定时器的输出，将根据 TMRx_CCMR 寄存器中的比较输出模式 OCxM 位，对输出状态进行相应的改变；
    - 将捕获/比较寄存器(TMRx_CCxR)的值重新加载到内部影子寄存器，使之生效。
    - 将定时器的计数器值(TMRx_CNTR)重新从 0 开始计数(连续循环模式)。
  - 如果没有使能比较器预加载功能(TMRx_CCMR 寄存器中的 CCxPE 位置 0)，事件产生后，定时器的动作与上述情况不同之处在于：对捕获/比较寄存器(TMRx_CCxR)写入新值将立即生效，捕获/比较事件不会影响该寄存器值何时生效。

注意：如果在到达比较值之前(未达到比较成功)，通过软件对 CCxUG 置 1 产生事件，与上述描述基本一致，不同之处在于：不会导致比较输出模式(OCM[2:0])所设定的输出变化，因为本次计数器值并没有与触发捕获/比较事件前所设定的比较值相匹配。

- 对于输入捕获模式：
  - 捕获/比较事件产生后，定时器将当前计数器 (TMRx_CNTR)的值捕获到捕获/比较寄存器(TMRx_CCxR)内；
  - 中断状态寄存器(TMRx_SR)中的捕获中断标志(CCxMIF)位将会置 1，并触发定时器中断(如果对应的中断已使能)。如果本次捕获成功产生捕获/比较事件时，前一次捕获中断标志未及时清除(CCxMIF=1)，则中断状态寄存器(TMRx_SR)中的重复捕获中断标志(CCxOIF)位将会置 1，并且触发定时器中断(如果对应的中断已使能)。

注意：如果通过软件对 CCxUG 置 1 产生的捕获/比较事件，则立刻执行上述动作，而不用外部产生符合捕获成功的信号。

30.4.12.5 中断

中断标志位与中断使能之间的关系如下表描述所示，详细内容请参考“30.4.12 事件与中断”章节描述：

表 30-4 中断标志位与中断使能之间的关系

中断名称	中断使能	中断标志	清除方式	备注
上溢中断	OVIE	OVIF	W1C	计数器计数上溢后产生
更新中断	UIE	UIF	W1C	计数器更新事件触发更新中断
比较捕获中断	CxMIE	CCxMIF	W1C 或读 CCxR	比较捕获模式下，计数值与比较值匹配或捕获到目标边沿
重复捕获中断	OCxIE	CCxOIF	W1C 或读 CCxR	输入捕获中断未及时清除，再次发生捕获
触发中断	TIE	TIF	W1C	在从模式下，输入信号出现目标触发边沿

30.5 寄存器定义

30.5.1 寄存器列表

TMRx 基地址：

TMR3(0x4002A000) TMR4(0x4002B000)

偏移	实例地址	名称	默认值	描述
0x00	TMRx 基地址+0x00	TMRx_CR0	0x00000000	TMRx 控制寄存器 0
0x04	TMRx 基地址+0x04	TMRx_CR1	0x00000000	TMRx 控制寄存器 1
0x08	TMRx 基地址+0x08	TMRx_SCR	0x00000000	TMRx Slave 控制寄存器
0x0C	TMRx 基地址+0x0C	TMRx_IER	0x00000000	TMRx 中断使能寄存器
0x10	TMRx 基地址+0x10	TMRx_SR	0x00000000	TMRx 状态寄存器
0x14	TMRx 基地址+0x14	TMRx_UGR	0x00000000	TMRx 更新事件生成寄存器
0x20	TMRx 基地址+0x20	TMRx_CCMR	0x00000000	TMRx 捕获/比较模式寄存器
0x24	TMRx 基地址+0x24	TMRx_CCER	0x00000000	TMRx 捕获/比较使能寄存器
0x2C	TMRx 基地址+0x2C	TMRx_TTCR	0x00000077	TMRx 触发周期寄存器
0x30	TMRx 基地址+0x30	TMRx_CPR	0xFFFFFFFF	TMRx 计数周期寄存器
0x38	TMRx 基地址+0x38	TMRx_PSCR	0x00000000	TMRx 预分频寄存器
0x3C	TMRx 基地址+0x3C	TMRx_CNTR	0x00000000	TMRx 计数寄存器
0x50	TMRx 基地址+0x50	TMRx_CC0R	0x00000000	TMRx 捕获/比较寄存器 0
0x54	TMRx 基地址+0x54	TMRx_CC1R	0x00000000	TMRx 捕获/比较寄存器 1
0x58	TMRx 基地址+0x58	TMRx_CC2R	0x00000000	TMRx 捕获/比较寄存器 2
0x5C	TMRx 基地址+0x5C	TMRx_CC3R	0x00000000	TMRx 捕获/比较寄存器 3
0x60	TMRx 基地址+0x60	TMRx_CIR	0x00000000	TMRx 捕获输入寄存器

30.5.2 寄存器描述

30.5.2.1 TMRx 控制寄存器 0（TMRx_CR0）

- 名称：TMRx Control Register0
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TI0S	DCD		ARE	CMS		DIR	OPM	URS	UDIS	CEN
					R/W	R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W
					0x0	0x0		0x0	0x0		0x0	0x0	0x0	0x0	0x0

字段	说明
[10] TI0S	**TI0 输入选择(TI0 Input Selection)** 0：CH0 引脚连接到 TI0 输入 1：CH0~CH2 引脚异或连接到 TI0 输入
[9:8] DCD	**时钟分频(Clock Division)** 此位指示定时器时钟频率与死区发生器及滤波发生器的时钟频率之间分频比； 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频
[7] ARE	**自动重载使能(AutoReload Enable)** AutoReload 功能使能，是否开启影子寄存器加载功能； 0: 关闭重载功能 1: 开启重载功能 Note： 1. 预加载功能涉及的寄存器有：计数终止值寄存器、预分频寄存器； 2. 如果不开启预加载功能，则相关寄存器写入新值将立刻生效； 3. 如果开启预加载功能，则相关寄存器写入新值后，在下一个更新事件（UEV）到来后生效。
[6:5] CMS	**中心对齐模式(Center-Aligned Mode Selection)** 00：边沿对齐模式，计数器根据方向位 (DIR) 递增计数或递减计数。 01：中心对齐模式 1，计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道的输出比较中断标志才置 1。 10：中心对齐模式 2，计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道的输出比较中断标志才置 1。 11：中心对齐模式 3，计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道的输出比较中断标志都会置 1。
[4] DIR	**计数方向(Counter Direction)** 0: 计数器递增计数； 1: 计数器递减计数；
[3] OPM	**单脉冲模式(One Pulse Mode)** 0: 计数器在发生更新事件时不会停止计数； 1: 计数器在发生下一更新事件时停止计数(将 CEN 位清零)； Note：配置为单次模式后，计数器溢出后，将自动停止计数。
[2] URS	**更新事件源(Update Request Source)** 此位由软件置 1 和清零，用以选择更新事件源。 0: 当配置为 0 时，所有以下事件都会生成更新事件: — 计数器上溢或下溢

	— 将 UG 位置 1 — 通过从模式生成的更新事件 1: 当配置为 1 时，只有计数器上溢或下溢时生成更新事件。
[1] UDIS	**更新禁止位(Update Disable)** 此位由软件置 1 和清零，用以使能/禁止更新事件的生成； 0:更新使能，更新(UEV)事件可通过以下事件之一生成: — 计数器上溢或下溢 — 将 UG 位置 1 — 通过从模式生成的更新事件 1:更新禁止 Note: 更新禁止后，不会生成更新事件(不会产生更新中断)，各影子寄存器的值保持不变，如果 UG 位置 1，则会重新初始化计数器和预分频器。
[0] CEN	**计数器使能位(Counter Enable)** 1:开始计数器 0:关闭计数器 Note: 单次模式下，当发生更新事件时会自动关闭 CEN；

30.5.2.2 TMRx 控制寄存器 1（TMRx_CR1）

- **名称：TMRx Control Register1**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													MMS		
													R/W		
													0x0		

字段	说明
[2:0] MMS	**主模式选择(Master Mode Selection)** 这些位用于选择主模式下将要发送到从定时器的同步的信息(TRGO)，组合描述如下： 000: 复位 - 寄存器(UGR)中的 UG 位用作触发输出，如果从模式控制器配置为复位模式，则 TRGO 上的信号用于从定时器的复位操作； 001: 使能 - 计数器使能信号用作触发输出(TRGO)，该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器； 010: 更新 - 选择更新事件作为触发输出(TRGO)。例如：主定时器可用作从定时器的预分频器； 011: 捕获比较事件 - 一旦发生输入捕获或比较匹配事件，当 CC0IF 标志被置 1 时，触发输出都会发送一个正脉冲； 100: 比较输出 - OC0 信号用作触发输出(TRGO)； 101: 比较输出 - OC1 信号用作触发输出(TRGO)； 110: 比较输出 - OC2 信号用作触发输出(TRGO)； 111: 比较输出 - OC3 信号用作触发输出(TRGO)；

30.5.2.3 TMRx Slave 控制寄存器（TMRx_SCR）

- 名称：TMRx Slave Control Register
- 偏移地址：0x08
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									EE	EMS		EFS			
									R/W	R/W		R/W			
									0x0	0x0		0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TS									SMS			
			R/W									R/W			
			0x0									0x0			

字段	说明
[22] EE	**ETR 控制模式选择(ETR Mode Selection)** 0: 外部时钟模式 1 1: 外部时钟模式 2 Note: 当选择外部时钟模式 2，将在上下双边沿进行计数，并且禁止从模式选择 ETR 作为源；
[21:20] EMS	**ETR 模式选择(ETR Mode Selection)** 00: 关闭 01: 上升沿 10: 下降沿 11: 上下沿
[19:16] EFS	**ETR 滤波选择(ETR Filter Selection)** 0: 无滤波 1: Ttds x1 2: Ttds x2 ……
[12:8] TS	**触发选择(Trigger Selection)** 此位域可选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) ..... 01111: 内部触发 15(ITR15) 10000: 滤波后的定时器输入 0(CH0-FE-双边沿) 10001: 滤波后的定时器输入 1(CH0-F) 10010: 滤波后的定时器输入 2(CH1-F) 10011: 滤波后的定时器输入 3(CH2-F) 10100: 滤波后的定时器输入 4(CH3-F) 10101: 边沿后的定时器输入 5(ETR-FE) 10110: 滤波后的定时器输入 6(ETR-F)
[3:0] SMS	**从模式选择(Slave Mode Selection)** 选择外部触发信号(TRGI)的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器说明）； 0000: 禁止从模式 0001: 保留 0010: 保留 0011: 保留 0100: 复位模式 - 在出现所选触发输入(TRGI)上升沿时，重新初始化计数器并生成一个寄存器更新事件； 0101: 门控模式 - 触发输入(TRGI)为高电平时使能计数器时钟，只要触发输入变为低电平，计数器立即停止计数（但不复位），计数器的启动和停止都被控制； 0110: 触发模式 - 触发信号 TRGI 出现上升沿时启动计数器（但不复位），只控制计数器的启动，可使用 FE 来快速同步使能； 0111: 外部时钟模式 - 由所选触发信号(TRGI)的上升沿提供计数器时钟； 1000: 组合复位+触发模式 - 在出现所选触发输入(TRGI)上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器； 1001: 组合门控+复位模式 - 在触发输入(TRGI)为高电平时使能计数器时钟，只要触发输入变为低电平，

计数器立即停止计数并复位，计数器的启动和停止都被控制；

30.5.2.4 TMRx 中断使能寄存器（TMRx_IER）

- 名称：TMRx Interrupt Enable Register
- 偏移地址：0x0C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TIE	OVIE	UIE	C3OIE	C3MIE	C2OIE	C2MIE	C1OIE	C1MIE	C0OIE	C0MIE
					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[10] TIE	**触发中断使能(Trigger Interrupt Enable)** 0：关闭 1：使能
[9] OVIE	**计数上溢中断使能(Overflow Interrupt Enable)** 1：使能中断 0：禁止中断
[8] UIE	**更新中断使能(Update Interrupt Enable)** 1：使能中断 0：禁止中断
[7] C3OIE	**CH3 重复捕获中断使能(CH3 OverCapture Interrupt Enable)** 1: 中断使能 0: 中断关闭
[6] C3MIE	**CH3 比较中断使能(CH3 Capture/Compare Interrupt Enable)** 1: 中断使能 0: 中断关闭
[5] C2OIE	**CH2 重复捕获中断使能(CH2 OverCapture Interrupt Enable)** 1: 中断使能 0: 中断关闭
[4] C2MIE	**CH2 比较中断使能(CH2 Capture/Compare Interrupt Enable)** 1: 中断使能 0: 中断关闭
[3] C1OIE	**CH1 重复捕获中断使能(CH1 OverCapture Interrupt Enable)** 1: 中断使能 0: 中断关闭
[2] C1MIE	**CH1 比较中断使能(CH1 Capture/Compare Interrupt Enable)** 1: 中断使能 0: 中断关闭
[1] C0OIE	**CH0 重复捕获中断使能(CH0 OverCapture Interrupt Enable)** 1: 中断使能 0: 中断关闭
[0] C0MIE	**CH0 比较中断使能(CH0 Capture/Compare Interrupt Enable)** 1: 中断使能 0: 中断关闭

30.5.2.5 TMRx 状态寄存器（TMRx_SR）

- 名称：TMRx Status Register
- 偏移地址：0x10
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TIF	OVIF	UIF	CC3OIF	CC3MIF	CC2OIF	CC2MIF	CC1OIF	CC1MIF	CC0OIF	CC0MIF
					R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[10] TIF	**触发中断标志(Trigger Interrupt Flag)** 该标志在发生触发事件时由硬件置 1，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将置 1，需要通过软件清零。 0: 未发生触发事件 1: 触发中断挂起
[9] OVIF	**计数器上溢中断标志位(Counter Overflow Interrupt Flag)** 0: 未发生上溢 1: 计数器上溢
[8] UIF	**更新中断标志(Update Interrupt Flag)** 该位在发生更新事件时通过硬件置 1,但需要通过软件清零。 0: 未发生更新 1: 更新中断挂起 该位在以下情况下更新寄存器时由硬件置 1: — 计数器发生上溢且当 CR 寄存器中 UDIS="0"时； — 当 CR 寄存器中 URS="0"且 UDIS="0"，通过软件使用 UGR 寄存器中的 UG 位重新初始化 Counter 时；
[7] CC3OIF	**CC3 重复捕获成功中断标志位(CC3 OverCapture Interrupt Flag)** 参考 CC0 的描述
[6] CC3MIF	**CC3 比较成功中断标志位(CC3 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[5] CC2OIF	**CC2 重复捕获成功中断标志位(CC2 OverCapture Interrupt Flag)** 参考 CC0 的描述
[4] CC2MIF	**CC2 比较成功中断标志位(CC2 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[3] CC1OIF	**CC1 重复捕获成功中断标志位(CC1 OverCapture Interrupt Flag)** 参考 CC0 的描述
[2] CC1MIF	**CC1 比较成功中断标志位(CC1 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[1] CC0OIF	**CC0 重复捕获成功中断标志位(CC0 OverCapture Interrupt Flag)** 0: 未检测到重复捕获 1: 当 CPIF=1，同时 Caputre 再次捕获到匹配的边沿，将置位该位(重复捕获成功有效) Note: 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1，通过软件写 1 可将该位清零
[0] CC0MIF	**CC0 比较成功中断标志位(CC0 Capture/Compare Interrupt Flag)** 配置为输入: 此位将在发生捕获事件时由硬件置 1，通过软件写 1 清 0 或读取 Capture 值寄存器时自动清 0。 0: 未发生输入捕获事件。 1: Compare 预加载寄存器中已捕获到计数器值（检测到与所选极性匹配的边沿） 配置为输出: 当计数器与比较值匹配时，此标志由硬件置 1，但需要通过软件写 1 清零。 0: 不匹配

1: 计数器的值与 Compare 寄存器的值匹配，当 Compare 值大于 Counter 周期值时， 将在计数器发生上溢时变为高电平；当 UGR 寄存器 UG 位软件置 1，也会触发 Compare 中断；

30.5.2.6 TMRx 更新事件生成寄存器（TMRx_UGR）

- **名称：TMRx Update Generation Register**
- **偏移地址：0x14**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CC3UG	CC2UG	CC1UG	CC0UG			TG	UG
								R/W1C	R/W1C	R/W1C	R/W1C			R/W1C	R/W1C
								0x0	0x0	0x0	0x0			0x0	0x0

字段	说明
[7] CC3UG	**CC3 捕获/比较更新事件生成(CC3 Capture/Compare Update Generation)** 参考 CC0 的描述
[6] CC2UG	**CC2 捕获/比较更新事件生成(CC2 Capture/Compare Update Generation)** 参考 CC0 的描述
[5] CC1UG	**CC1 捕获/比较更新事件生成(CC1 Capture/Compare Update Generation)** 参考 CC0 的描述
[4] CC0UG	**CC0 捕获/比较更新事件生成(CC0 Capture/Compare Update Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零 0: 不执行任何操作 1: 重生 Capture/Compare 更新事件 配置为输出: 使能后，Compare 中断标志置 1 并发送相应的中断，生成更新事件，更新 Compare 值； 配置为输入: 将捕获到计数器当前值写入 CCxR 寄存器中，使能后，Capture 中断标志置 1 并发送相应中断，如果 Capture 中断标志已为高电平，重复标志将置 1
[1] TG	**触发事件生成(Trigger Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: SR 寄存器中的 TIF 标志置 1，生成 Trigger 事件。
[0] UG	**更新事件生成(Update Generation)** 此位由软件置 1 生成事件，并由硬件自动清零; 0: 不执行任何操作 1: 重新初始化计数器并生成寄存器更新事件，预分频器计数器也将清零(但预分频比不受影响)，计数器清零

30.5.2.7 TMRx 捕获/比较模式寄存器（TMRx_CCMR）

- 名称：TMRx Capture Compare Mode Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OC3M				CC3PE		CC3S		OC2M				CC2PE		CC2S	
R/W				R/W		R/W		R/W				R/W		R/W	
0x0				0x0		0x0		0x0				0x0		0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC1M				CC1PE		CC1S		OC0M				CC0PE		CC0S	
R/W				R/W		R/W		R/W				R/W		R/W	
0x0				0x0		0x0		0x0				0x0		0x0	

字段	说明
[31:28] OC3M	**CC3 输入滤波/输出比较模式选择(CC3 Input Filter/Output Compare Mode)** 参考 CC0 描述；
[27:26] CC3PE	**CC3 输入捕获分频/比较自动加载使能(CC3 Compare Auto-Realod Enable)** 参考 CC0 描述；
[25:24] CC3S	**CC3 捕获/比较方向选择(CC3 Capture Compare Selection)** 参考 CC0 描述；
[23:20] OC2M	**CC2 输入滤波/输出比较模式选择(CC2 Input Filter/Output Compare Mode)** 参考 CC0 描述；
[19:18] CC2PE	**CC2 输入捕获分频/比较自动加载使能(CC2 Compare Auto-Realod Enable)** 参考 CC0 描述；
[17:16] CC2S	**CC2 捕获/比较方向选择(CC2 Capture Compare Selection)** 参考 CC0 描述；
[15:12] OC1M	**CC1 输入滤波/输出比较模式选择(CC1 Input Filter/Output Compare Mode)** 参考 CC0 描述；
[11:10] CC1PE	**CC1 输入捕获分频/比较自动加载使能(CC1 Compare Auto-Realod Enable)** 参考 CC0 描述；
[9:8] CC1S	**CC1 捕获/比较方向选择(CC1 Capture Compare Selection)** 参考 CC0 描述；
[7:4] OC0M	**CC0 输入滤波/输出比较模式选择(CC0 Input Filter/Output Compare Mode)** 配置为输出，OCxRef 的输出模式（有效电平为高电平） 0000: 冻结, 输出比较寄存器与计数器进行比较不会对输出造成任何影响(保持原有状态) 0001: 当计数器与捕获/比较寄存器匹配时，输出有效电平(高电平) 0010: 当计数器与捕获/比较寄存器匹配时，输出无效电平(低电平) 0011: 当计数器与捕获/比较寄存器匹配时，发生翻转 0100: 强制变为无效电平(低电平) 0101: 强制变为有效电平(高电平) 0110：PWM 模式 1 - 在递增计数模式下，只要 TMRx_CNT<TMRx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TMRx_CNT>TMRx_CCR1，通道 1 便为无效状态(OC1REF=0)，否则为有效状态(OC1REF=1)。 0111：PWM 模式 2 - 在递增计数模式下，只要 TMRx_CNT<TMRx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TMRx_CNT>TMRx_CCR1，通道 1 便为有效状态，否则为无效状态。 1000：可再触发 OPM 模式 1 - 在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。 1001：可再触发 OPM 模式 2 - 在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。 1100: 组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF

	的逻辑或运算结果。 1101：组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。 1110：不对称 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。 1111：不对称 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。 配置为输入 0: 无滤波 1: Ttds x1 2: Ttds x2 ……
[3:2] CC0PE	**CC0 输入捕获分频/比较自动加载使能(CC0 Compare Auto-Realod Enable)** 配置为输出 0: 禁止自动预装载，可随时向预装载寄存器写入数据，写入后将立即使用新值 1: 使能自动预装载，可读/写访问预装载寄存器，而预装载值在每次生成更新事件时都会装载到活动寄存器中 配置为输入 此位域定义 CC0 输入 (IC0) 的预分频比。 00：无预分频器，捕获输入上每检测到一个边沿便执行捕获 01：每发生 2 个事件便执行一次捕获 10：每发生 4 个事件便执行一次捕获 11：每发生 8 个事件便执行一次捕获
[1:0] CC0S	**CC0 捕获/比较方向选择(CC0 Capture Compare Selection)** 此位定义对应通道的方向（输入/输出）。 00: CC0 通道配置为输出。 01: CC0 通道配置为输入，输入映射到 CH0 上 10: CC0 通道配置为输入，输入映射到 CH1 上 11: CC0 通道配置为输入，输入映射到 ITR 上，此模式仅在通过 TS 位选择内部触发输入时有效；

30.5.2.8 TMRx 捕获/比较使能寄存器（TMRx_CCER）

- **名称：TMRx Capture Compare Enable Register**
- **偏移地址：0x24**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC3P			CC3E	CC2P			CC2E	CC1P			CC1E	CC0P			CC0E
R/W			R/W	R/W			R/W	R/W			R/W	R/W			R/W
0x0			0x0	0x0			0x0	0x0			0x0	0x0			0x0

字段	说明
[15:14] CC3P	**CC3 捕获/比较极性选择(CC3 Compare/Capture Polarity)** 参考 CC0 的描述
[12] CC3E	**CC3 捕获/比较通道使能(CC3 Capture/Compare Enable)** 考 CC0 的描述
[11:10] CC2P	**CC2 捕获/比较极性选择(CC2 Compare/Capture Polarity)** 参考 CC0 的描述
[8] CC2E	**CC2 捕获/比较通道使能(CC2 Capture/Compare Enable)** 考 CC0 的描述
[7:6]	**CC1 捕获/比较极性选择(CC1 Compare/Capture Polarity)**

CC1P	参考 CC0 的描述
[4] CC1E	**CC1 捕获/比较通道使能(CC1 Capture/Compare Enable)** 考 CC0 的描述
[3:2] CC0P	**CC0 捕获/比较极性选择(CC0 Compare/Capture Polarity)** 配置为输出(bit2 对应 CC0 的输出极性选择) 0: 高电平有效 1: 低电平有效 配置为输入: 00: 关闭 01: 上升沿 10: 下降沿 11: 上升沿和下降沿
[0] CC0E	**CC0 捕获/比较通道使能(CC0 Capture/Compare Enable)** 配置为输出: 0: 关闭 1: 开启 配置为输入: 该位决定了输入捕获/比较寄存器 1 实际捕获到计数器的值 0: 禁止捕获 1: 使能捕获

30.5.2.9 TMRx 触发周期寄存器（TMRx_TTCR）

- **名称：TMRx Trigger Cycles Register**
- **偏移地址：0x2C**
- **默认值：0x00000077**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				TC3E	TC2E	TC1E	TC0E	TCW				TOW			
				R/W	R/W	R/W	R/W	R/W				R/W			
				0x0	0x0	0x0	0x0	0x7				0x7			

字段	说明
[11] TC3E	**TRGO CH3 输出使能(Trigger CH3 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[10] TC2E	**TRGO CH2 输出使能(Trigger CH2 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[9] TC1E	**TRGO CH1 输出使能(Trigger CH1 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[8] TC0E	**TRGO CH0 输出使能(Trigger CH0 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[7:4]	**TRGCC 脉冲宽度控制(Trigger Compare/Caputre Width)**

TCW	0: 1 个 Cycle 1: 2 个 Cycle …… Note： 1. TRGCC 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle： 2. TRGCC 脉冲宽度必须小于 Timer 的 Period 值(CPV).
[3:0] TOW	**TRGO 脉冲宽度控制(Trigger Output Width)** 0: 1 个 Cycle 1: 2 个 Cycle …… Note： 1. TRGO 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle： 2. TRGO 脉冲宽度必须小于 Timer 的 Period 值(CPV).

30.5.2.10 TMRx 计数周期寄存器（TMRx_CPR）

- **名称：TMRx Counter Period Register**
- **偏移地址：0x30**
- **默认值：0xFFFFFFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPV															
R/W															
0xFFFFFFFF															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPV															
R/W															
0xFFFFFFFF															

字段	说明
[31:0] CPV	**计数周期值(Counter Period Value)** 通过配置此寄存器设置定时器结束计数的值。此外，该寄存器支持自动预装载（Auto-Reload）功能： 1. 不使能自动预装载功能（ARPE=0） 对该寄存器的设置将立即作用到内部影子寄存器内，本次计数周期内即生效； 2. 使能自动预装载功能（ARPE=1） 对该寄存器的设置将在更新事件（UEV）产生后，再更新至内部影子寄存器内生效。

30.5.2.11 TMRx 预分频寄存器（TMRx_PSCR）

- 名称：TMRx Prescaler Register
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC															
R/W															
0x0															

字段	说明
[15:0] PSC	**预分频寄存器（Prescaler Value)** 该寄存器支持自动预装载(Auto-Reload)功能，每次发生更新事件时会装载到实际预分频器寄存器的值； Note：计数器时钟频率 CLK_CNT 等于 FCLK/(PSC+1)，其中 PSC 范围为 0~65535： 0:不分频，系统时钟 1:2 分频 2:3 分频 ……

30.5.2.12 TMRx 计数寄存器（TMRx_CNTR）

- 名称：TMRx Counter Register
- 偏移地址：0x3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															
0x0															

字段	说明
[31:0] CNT	**定时器计数值(Counter Value)** 通过该寄存器可对定时器计数器的值进行读取/写入。 需注意，因为系统与外设之间存在一定的总线延迟，对该寄存器的读取/写入操作可能存在一定的偏差。

30.5.2.13 TMRx 捕获/比较寄存器 0（TMRx_CC0R）

- 名称：TMRx Capture Compare Register0
- 偏移地址：0x50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CC0V															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC0V															
R/W															
0x0															

字段	说明
[31:0] CC0V	**CC0 捕获/比较值(CC0 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

30.5.2.14 TMRx 捕获/比较寄存器 1（TMRx_CC1R）

- 名称：TMRx Capture Compare Register1
- 偏移地址：0x54
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CC1V															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1V															
R/W															
0x0															

字段	说明
[31:0] CC1V	**CC1 捕获/比较值(CC1 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次

	2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

30.5.2.15 TMRx 捕获/比较寄存器 2（TMRx_CC2R）

- **名称：TMRx Capture Compare Register2**
- **偏移地址：0x58**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CC2V															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC2V															
R/W															
0x0															

字段	说明
[31:0] CC2V	**CC2 捕获/比较值(CC2 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

30.5.2.16 TMRx 捕获/比较寄存器 3（TMRx_CC3R）

- 名称：TMRx Capture Compare Register3
- 偏移地址：0x5C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CC3V															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC3V															
R/W															
0x0															

字段	说明
[31:0] CC3V	**CC3 捕获/比较值(CC3 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

30.5.2.17 TMRx 捕获输入寄存器（TMRx_CIR）

- 名称：TMRx Capture Input Register
- 偏移地址：0x60
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				C3TIS								C2TIS			
				R/W								R/W			
				0x0								0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				C1TIS								C0TIS			
				R/W								R/W			
				0x0								0x0			

字段	说明
[27:24] C3TIS	**CH3 触发输入选择(CH3 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[19:16] C2TIS	**CH2 触发输入选择(CH2 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[11:8] C1TIS	**CH1 触发输入选择(CH1 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[3:0] C0TIS	**CH0 触发输入选择(CH0 Trigger Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……

31 高级控制定时器（TMR9/TMR10）

31.1 简介

TMR9/TMR10 高级控制定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

高级控制定时器可用于多种用途，包括测量输入信号的脉冲长度（输入捕获）或生成输出波形（输出比较、PWM、带死区插入的互补 PWM）。

使用定时器预分频器和 RCU 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

TMR9/TMR10 定时器之间彼此完全独立，不共享任何资源，可以同步操作。

31.2 主要特性

TMR9/10 主要具有以下特性：

- 16 位递增、递减和递增/递减自动重载计数器
- 16 位的预分频器，用于对计数器的时钟源进行预分频，分频系数介于 1 和 65536 之间
- 16 位的计数比较值和周期值寄存器
- 两种工作模式：
  - 循环模式，计数完成后自动重载并继续计数
  - 单次模式，计数完成后自动关闭定时器的使能
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 带可编程死区时间的互补输出
- 支持主从模式，可以实现多个定时器的互连
- 支持断路保护功能，在发生异常时将 PWM 输出置于复位或已知状态
- 发生如下事件时产生中断：
  - 计数器上溢(Overflow Event)
  - 更新事件(Update Event) (需使能)：计数器上溢产生更新事件，也可软件产生(UG)
  - 输入捕获(Capture Event)、重复捕获(Over Capture Event)
  - 输出比较(Compare Event)
  - 断路输入(Break Event)
- 外部时钟或逐周期的触发输入

31.3 结构框图

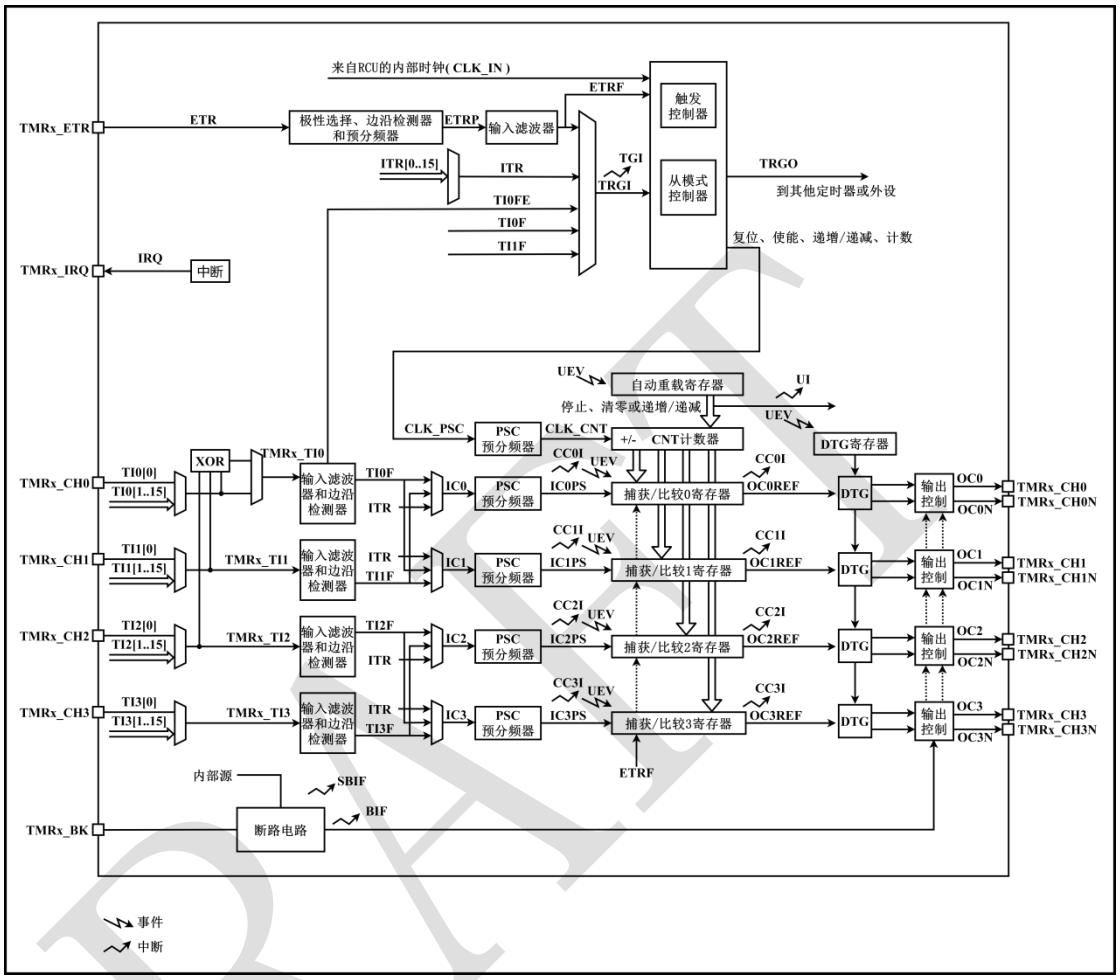


图 31-1 TMR 结构框图

31.4 功能描述

31.4.1 内部信号

表 31-1 列出了 TMRx 定时器主要的输入/输出信号及简要描述：

表 31-1 TMRx 内部信号

Signal Name	Signal Type	Description
HCLK	Input	TMRx AHB Clock
TMRx_CH0/1/2/3	Input/Output	TMRx 多功能复用通道，可以用作 Compare、Capture 及 PWM，还可以用于外部时钟或 Trigger 触发输入。
TMRx_ETR	Input	TMRx 外部时钟源 2 输入引脚
TMRx_BKx	Input	TMRx Break 输入信号
TRGO	Output	TMRx 内部触发输出信号，可用于触发其他外设

IRQ	Output	TMRx 全局中断，包括 Capture/Compare/Update/Break 中断请求。

表 31-2 列出了 TMRx 定时器 TMRx_TI 输入信号的连接关系：

表 31-2 TMRx_TI 信号内部互连

表 a. TMR9_TI 信号内部互连

TMRx_TIx 输入	Sources			
	事件名称(CH0)	事件名称(CH1)	事件名称(CH2)	事件名称(CH3)
TMRx_TI_0	CMP0_OUT	CMP0_OUT	CMP0_OUT	CMP8_OUT
TMRx_TI_1	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP1_OUT
TMRx_TI_2	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT
TMRx_TI_3	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT
TMRx_TI_4	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT
TMRx_TI_5	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT
TMRx_TI_6	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT
TMRx_TI_7	HSI	HSE	LSI	CMP7_OUT
TMRx_TI_8	TMR9_CH0	TMR9_CH0	TMR9_CH0	TMR9_CH0
TMRx_TI_9	TMR9_CH1	TMR9_CH1	TMR9_CH1	TMR9_CH1
TMRx_TI_10	TMR9_CH2	TMR9_CH2	TMR9_CH2	TMR9_CH2
TMRx_TI_11	TMR9_CH3	TMR9_CH3	TMR9_CH3	TMR9_CH3
TMRx_TI_12	TMR9_CH0N	TMR9_CH0N	TMR9_CH0N	TMR9_CH0N
TMRx_TI_13	TMR9_CH1N	TMR9_CH1N	TMR9_CH1N	TMR9_CH1N
TMRx_TI_14	TMR9_CH2N	TMR9_CH2N	TMR9_CH2N	TMR9_CH2N
TMRx_TI_15	TMR9_CH3N	TMR9_CH3N	TMR9_CH3N	TMR9_CH3N

表 b. TMR10_TI 信号内部互连

TMRx_TIx 输入	Sources			
	事件名称(CH0)	事件名称(CH1)	事件名称(CH2)	事件名称(CH3)
TMRx_TI_0	CMP1_OUT	CMP1_OUT	CMP1_OUT	CMP0_OUT
TMRx_TI_1	CMP2_OUT	CMP2_OUT	CMP2_OUT	CMP2_OUT
TMRx_TI_2	CMP3_OUT	CMP3_OUT	CMP3_OUT	CMP3_OUT
TMRx_TI_3	CMP4_OUT	CMP4_OUT	CMP4_OUT	CMP4_OUT
TMRx_TI_4	CMP5_OUT	CMP5_OUT	CMP5_OUT	CMP5_OUT
TMRx_TI_5	CMP6_OUT	CMP6_OUT	CMP6_OUT	CMP6_OUT
TMRx_TI_6	CMP7_OUT	CMP7_OUT	CMP7_OUT	CMP7_OUT
TMRx_TI_7	HSI	HSE	LSI	CMP8_OUT
TMRx_TI_8	TMR10_CH0	TMR10_CH0	TMR10_CH0	TMR10_CH0
TMRx_TI_9	TMR10_CH1	TMR10_CH1	TMR10_CH1	TMR10_CH1
TMRx_TI_10	TMR10_CH2	TMR10_CH2	TMR10_CH2	TMR10_CH2
TMRx_TI_11	TMR10_CH3	TMR10_CH3	TMR10_CH3	TMR10_CH3
TMRx_TI_12	TMR10_CH0N	TMR10_CH0N	TMR10_CH0N	TMR10_CH0N
TMRx_TI_13	TMR10_CH1N	TMR10_CH1N	TMR10_CH1N	TMR10_CH1N
TMRx_TI_14	TMR10_CH2N	TMR10_CH2N	TMR10_CH2N	TMR10_CH2N
TMRx_TI_15	TMR10_CH3N	TMR10_CH3N	TMR10_CH3N	TMR10_CH3N

表 31-3 列出了 TMRx 定时器内部 ETR 输入信号的连接关系：

表 31-3 TMRx_ETR 信号内部互连

TMRx_ETRx 输入	Sources	
	TMR9	TMR10
TMRx_ETR_0	TMR9_ETR	TMR10_ETR
TMRx_ETR_1	CMP0_OUT	CMP2_OUT
TMRx_ETR_2	CMP1_OUT	CMP3_OUT
TMRx_ETR_3	CMP2_OUT	CMP4_OUT
TMRx_ETR_4	CMP3_OUT	CMP5_OUT
TMRx_ETR_5	CMP4_OUT	CMP6_OUT
TMRx_ETR_6	CMP5_OUT	CMP7_OUT
TMRx_ETR_7	CMP6_OUT	CMP8_OUT
TMRx_ETR_8	PDM0_CMPH	PDM2_CMPH
TMRx_ETR_9	PDM1_CMPH	PDM3_CMPH
TMRx_ETR_10	ADC0_AWD0	ADC2_AWD0
TMRx_ETR_11	ADC0_AWD1	ADC2_AWD1
TMRx_ETR_12	HSE	HSE
TMRx_ETR_13	ADC1_AWD0	ADC3_AWD0
TMRx_ETR_14	ADC1_AWD1	ADC3_AWD1
TMRx_ETR_15	HSI	HSI

表 31-4 列出了 TMRx 定时器内部 ITR 输入信号的连接关系：

表 31-4 TMRx_ITR 信号内部互连

TMRx_ITRx 输入	Sources	
	TMR9	TMR10
TMRx_ITR_0	TMR7_TRGO	TMR7_TRGO
TMRx_ITR_1	TMR8_TRGO	TMR8_TRGO
TMRx_ITR_2	TMR0_TRGO	TMR0_TRGO
TMRx_ITR_3	TMR1_TRGO	TMR1_TRGO
TMRx_ITR_4	TMR2_TRGO	TMR2_TRGO
TMRx_ITR_5	TMR3_TRGO	TMR3_TRGO
TMRx_ITR_6	TMR4_TRGO	TMR4_TRGO
TMRx_ITR_7	TMR0_OC0	TMR0_OC0
TMRx_ITR_8	TMR1_OC0	TMR1_OC0
TMRx_ITR_9	TMR2_OC0	TMR2_OC0
TMRx_ITR_10	TMR3_OC0	TMR3_OC0
TMRx_ITR_11	TMR4_OC0	TMR4_OC0
TMRx_ITR_12	TMR10_TRGO	TMR9_TRGO
TMRx_ITR_13	TMR10_OC0	TMR9_OC0
TMRx_ITR_14	TMR10_OC1	TMR9_OC1
TMRx_ITR_15	HRPWM_SYNCO	HRPWM_SYNCO

表 31-5 和表 31-6 分别列出了 TMRx 定时器内部 Break0/1 输入信号的连接关系：

表 31-5 TMRx Break0 信号内部互连

TMRx_BKx 输入	Sources	
	TMR9	TMR10
TMRx_BK0_0	CMP0_OUT	CMP0_OUT
TMRx_BK0_1	CMP1_OUT	CMP1_OUT
TMRx_BK0_2	CMP2_OUT	CMP2_OUT
TMRx_BK0_3	CMP3_OUT	CMP3_OUT
TMRx_BK0_4	CMP4_OUT	CMP4_OUT
TMRx_BK0_5	CMP5_OUT	CMP5_OUT
TMRx_BK0_6	CMP6_OUT	CMP6_OUT
TMRx_BK0_7	CMP7_OUT	CMP7_OUT
TMRx_BK0_8	CMP8_OUT	CMP8_OUT
TMRx_BK0_9	TMR9_BK0	TMR10_BK0
TMRx_BK0_10	ADC0_AWD0	ADC0_AWD0
TMRx_BK0_11	ADC1_AWD0	ADC1_AWD0
TMRx_BK0_12	PDM0_CMPH	PDM0_CMPH
TMRx_BK0_13	PDM1_CMPH	PDM1_CMPH
TMRx_BK0_14	PDM2_CMPH	PDM2_CMPH
TMRx_BK0_15	PDM3_CMPH	PDM3_CMPH

表 31-6 TMRx Break1 信号内部互连

TMRx_BKx 输入	Sources	
	TMR9	TMR10
TMRx_BK1_0	CMP0_OUT	CMP0_OUT
TMRx_BK1_1	CMP1_OUT	CMP1_OUT
TMRx_BK1_2	CMP2_OUT	CMP2_OUT
TMRx_BK1_3	CMP3_OUT	CMP3_OUT
TMRx_BK1_4	CMP4_OUT	CMP4_OUT
TMRx_BK1_5	CMP5_OUT	CMP5_OUT
TMRx_BK1_6	CMP6_OUT	CMP6_OUT
TMRx_BK1_7	CMP7_OUT	CMP7_OUT
TMRx_BK1_8	CMP8_OUT	CMP8_OUT
TMRx_BK1_9	TMR9_BK1	TMR10_BK1
TMRx_BK1_10	ADC0_AWD0	ADC0_AWD0
TMRx_BK1_11	ADC1_AWD0	ADC1_AWD0
TMRx_BK1_12	PDM0_CMPH	PDM0_CMPH
TMRx_BK1_13	PDM1_CMPH	PDM1_CMPH
TMRx_BK1_14	PDM2_CMPH	PDM2_CMPH
TMRx_BK1_15	PDM3_CMPH	PDM3_CMPH

表 31-7 列出了 TMRx 定时器内部系统 Break 输入信号的连接关系：

表 31-7 TMRx 系统 Break 信号内部互连

TMRx_SBK	TMR9/10
TMRx_SBK_0	Cortex™-M4 With FPU LOCKUP
TMRx_SBK_1	Low Voltage Detector (LVD)
TMRx_SBK_2	FLASH ECC Error
TMRx_SBK_3	QEI0/1/2 Error
TMRx_SBK_4	Clock Security System（CSS）

注意：TMRx 系统 Break 输入信号由以上 5 个源通过或运算得到，没有优先级或选择控制。

31.4.2 时基单元

定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成，计数器采用递增计数的方式，计数器的时钟可通过预分频器进行分频。

定时器的自动重载计数器(TMRx_CNTR)、计数周期寄存器(TMRx_CPR) 、重复计数寄存器(TMRx_CRR)和预分频寄存器(TMRx_PSCR)可通过软件进行读写，即使在计数器运行中也可执行读写操作。

时基单元包含：

- 自动重载计数器(TMRx_CNTR)
- 预分频寄存器(TMRx_PSCR)
- 计数周期寄存器(TMRx_CPR)
- 重复计数寄存器(TMRx_CRR)

其中预分频寄存器(TMRx_PSCR)、计数周期寄存器(TMRx_CPR)、重复计数寄存器(TMRx_CPR)都是可预装载的。对预装载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器立即生效，也可以在每次发生更新事件时更新到影子寄存器，这取决于定时器 TMRx_CR0 控制寄存器中的 ARE 位（自动重载预装载使能位）。

定时器计数器计数到 TMRx_CPR 计数周期寄存器设定的值，且控制寄存器 TMRx_CR0 中的 UDIS 位为 0 时，将产生更新事件。此外，该更新事件也可通过软件发起，通过对定时器事件生成寄存器 TMRx_UGR 中的 UG 位置 1。

更新事件相关的详细介绍，请参看“31.4.14.2 更新事件”章节。

定时器的计数器由时钟源经过预分频器分频后提供的时钟驱动，且仅当 TMRx_CR0 控寄存器中的 CEN 位(计数器使能)置 1 时，才会启动计数器计数。

注意：计数器将在 TMRx_CR0 控制寄存器中的位 CEN(计数器使能)置 1 时刻后的一个时钟周期开始计数。

预分频器说明

预分频器用于对计数器的计数时钟频率进行分频，分频系数介于 1 和 65536 之间。通过配置预分频器寄存器(TMRx_PSCR)来设定预分频的分频系数，计数器计数频率的计算公式如下：

$$f_{CLK\_CNT} = f_{CLK\_SRC} / (PSC + 1)$$

该寄存器具有预加载缓存功能，当启动预加载功能后，可以对预分频器进行实时修改，而新的预分频系数将在下一个更新事件发生时被采用，也可通过软件设置 TMRx_UGR 寄存器中的 UG 位，立即产生一个更新事件（UEV）。

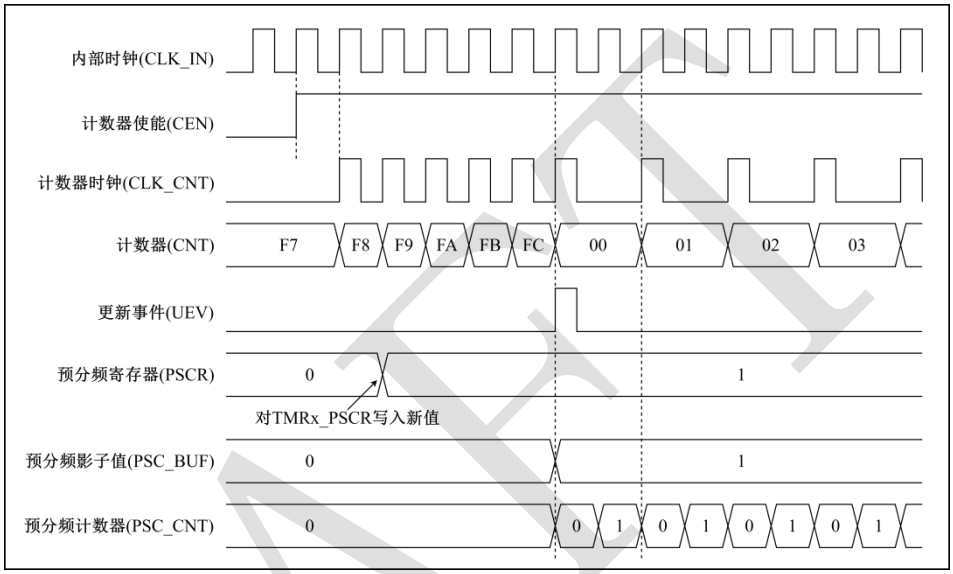


图 31-2　实时修改预分频器的分频值从 1 变为 2 的时序图

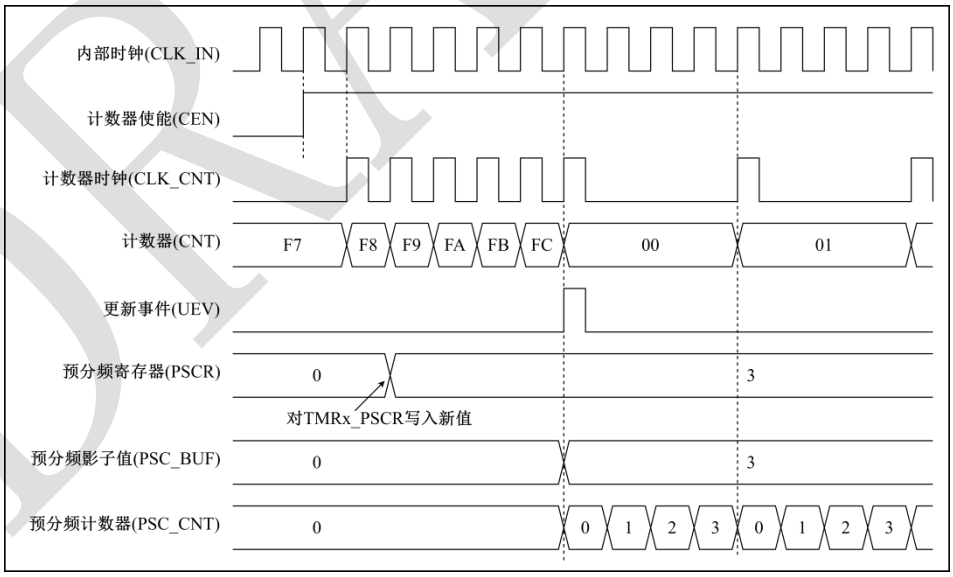


图 31-3　实时修改预分频器的分频值从 1 变为 4 的时序图

31.4.3　计数器模式

递增计数模式

定时器支持向上递增计数模式，计数器从 0 开始计数到计数周期值(TMRx_CPR)，支持单次和

连续计数模式：单次模式下，计数完成后将自动关闭计数器使能(CEN 自动置 0)，即停止计数；连续模式下，每次计数完成后，自动从 0 重新开始计数，并生成计数器上溢事件 OVF 和更新事件 UEV(如果使能更新事件)。

每次发生计数器上溢时会生成更新事件，或将 TMRx_UGR 寄存器中的 UG 位置 1（通过软件）也可以生成更新事件。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过计数器及预分频计数器仍然能够在上溢事件 OVEV 后，重新从 0x0 开始计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器等于 0x36 时，不同时钟频率下表现的行为。

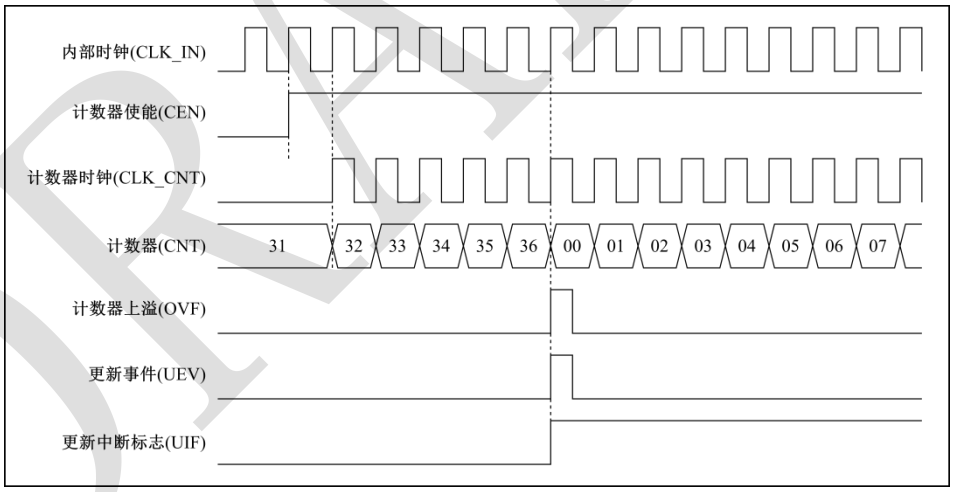


图 31-4　计数器时序图，1 分频内部时钟

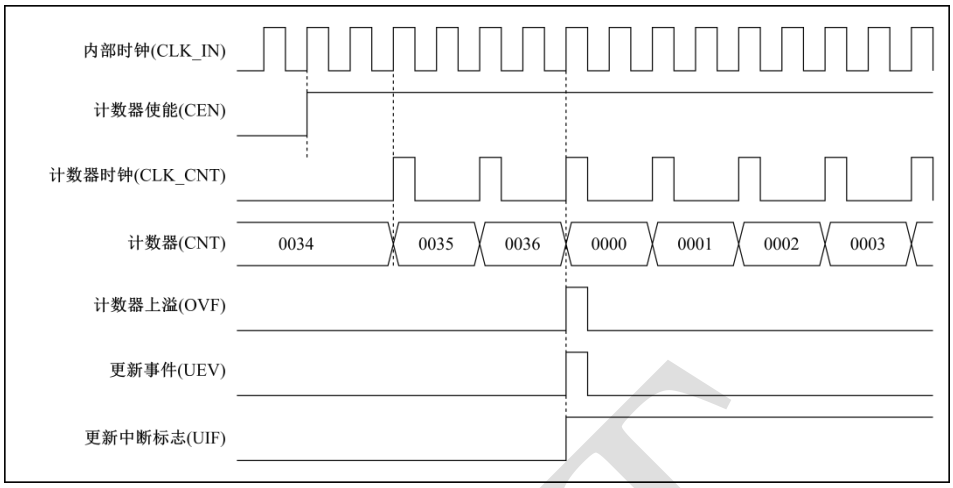


图 31-5　计数器时序图，2 分频内部时钟

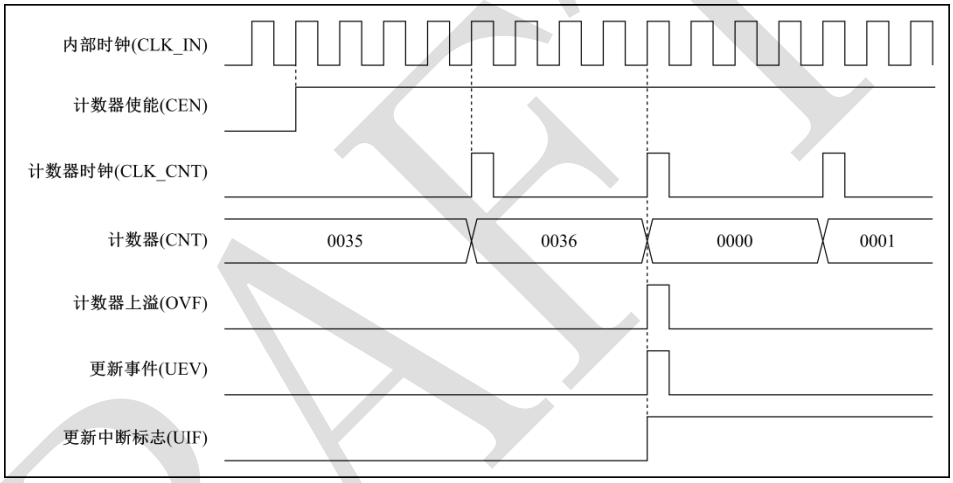


图 31-6　计数器时序图，4 分频内部时钟

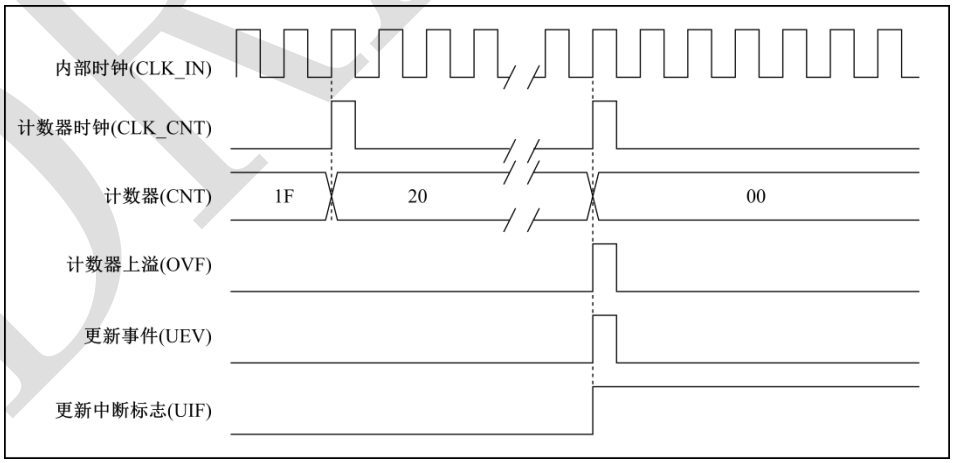


图 31-7　计数器时序图，N 分频内部时钟

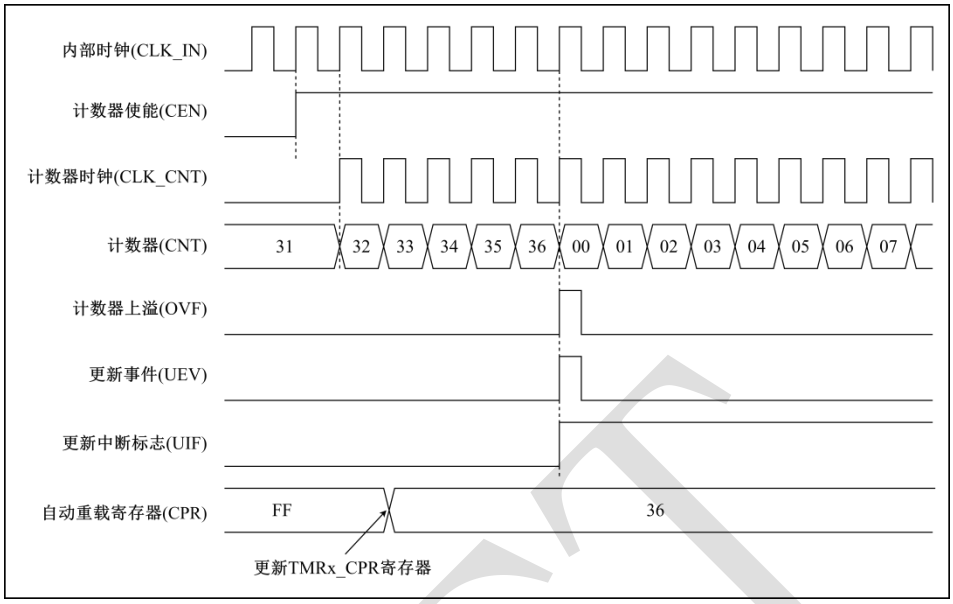


图 31-8　计数器时序图，运行中更新重载值（自动重载使能关闭-实时生效）

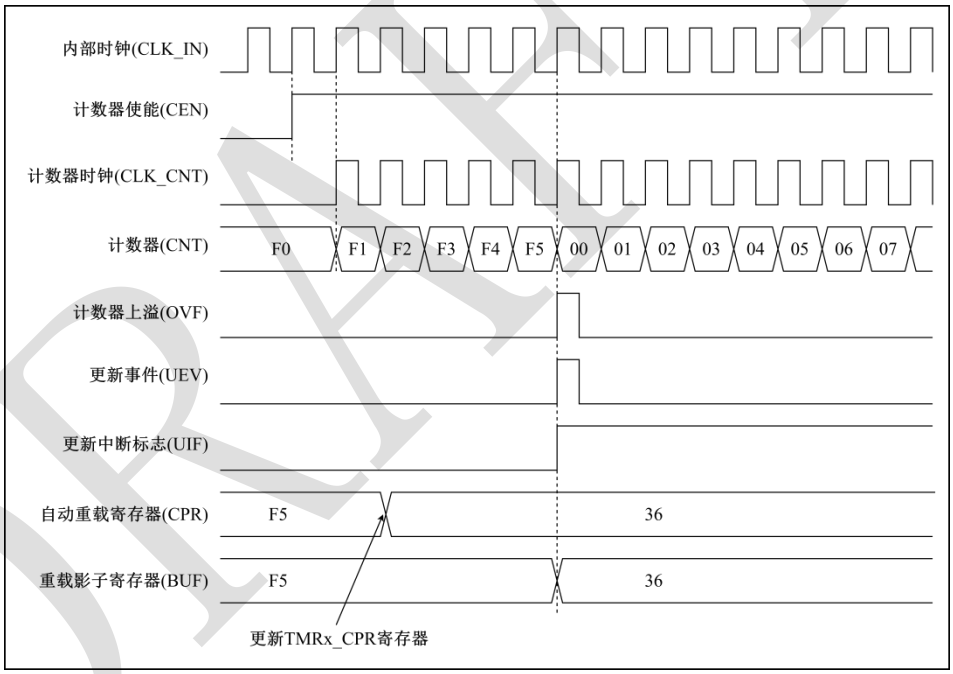


图 31-9　计数器时序图，运行中更新重载值（自动重载使能使能-更新事件生效）

递减计数模式

定时器支持向下递减计数模式，计数器从计数周期值(TMRx_CPR)开始递减到 0，支持单次和连续计数模式：单次模式下，计数完成后将自动关闭计数器使能(CEN 自动置 0)，即停止计数；连续模式下，每次计数完成后，自动从计数周期值(TMRx_CPR)重新开始递减计数，并生成计数器下溢事件 OVEV 和更新事件 UEV(如果使能更新事件)。

每次计数器发生下溢时会生成更新事件，或将 TMRx_UGR 寄存器中的 UG 位置 1（通过软件）也可以生成更新事件。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过计数器及预分频计数器仍然能够在下溢事件 OVEV 后，重新从 0x0 开始计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器等于 0x36 时，不同时钟频率下表现的行为。

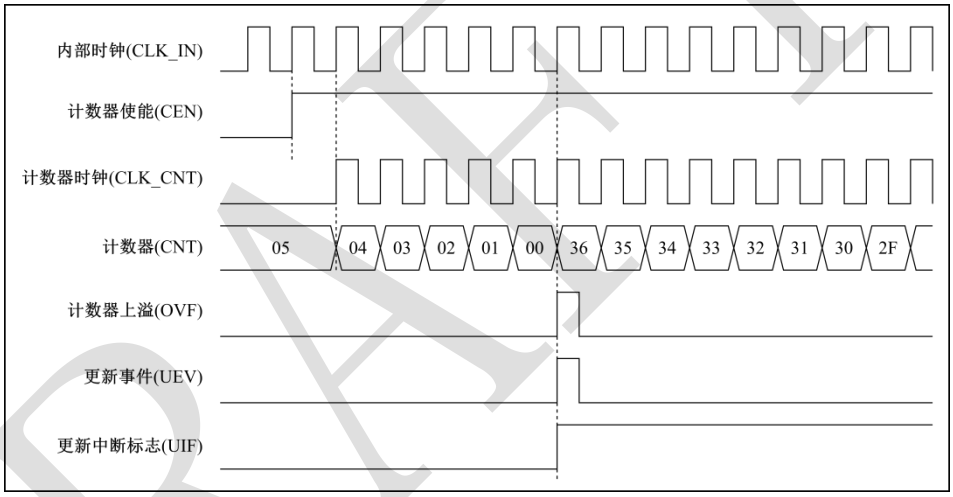


图 31-10　计数器时序图，1 分频内部时钟

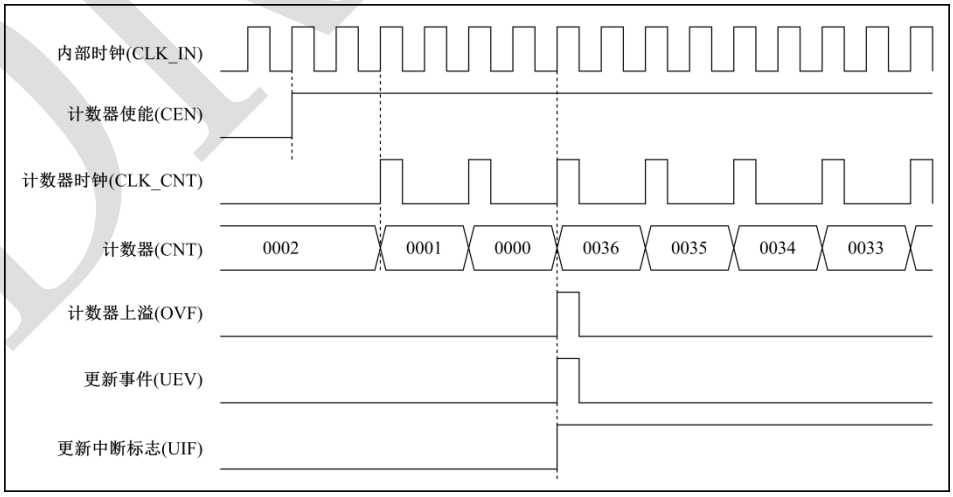


图 31-11　计数器时序图，2 分频内部时钟

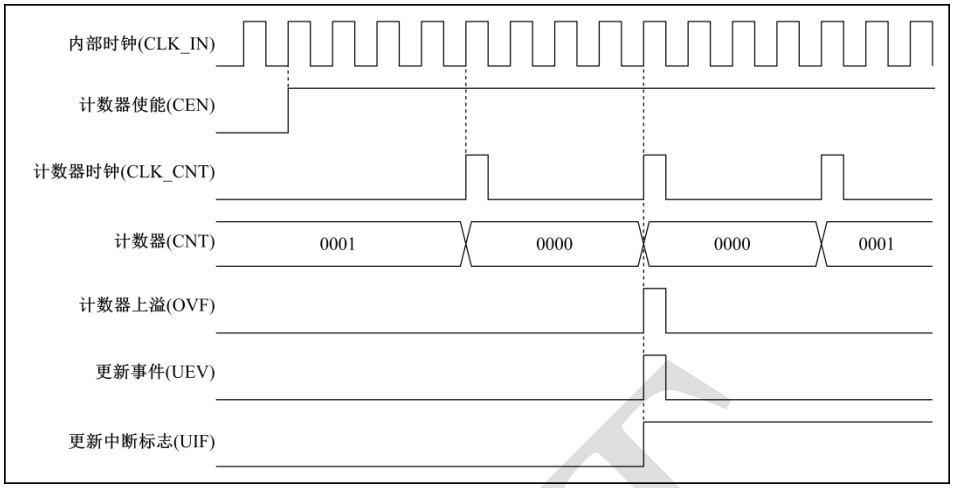


图 31-12　计数器时序图，4 分频内部时钟

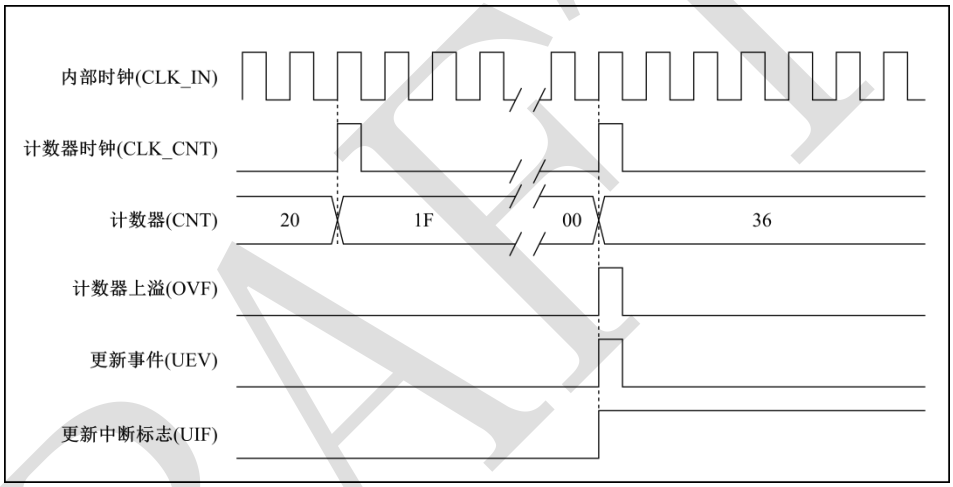


图 31-13　计数器时序图，N 分频内部时钟

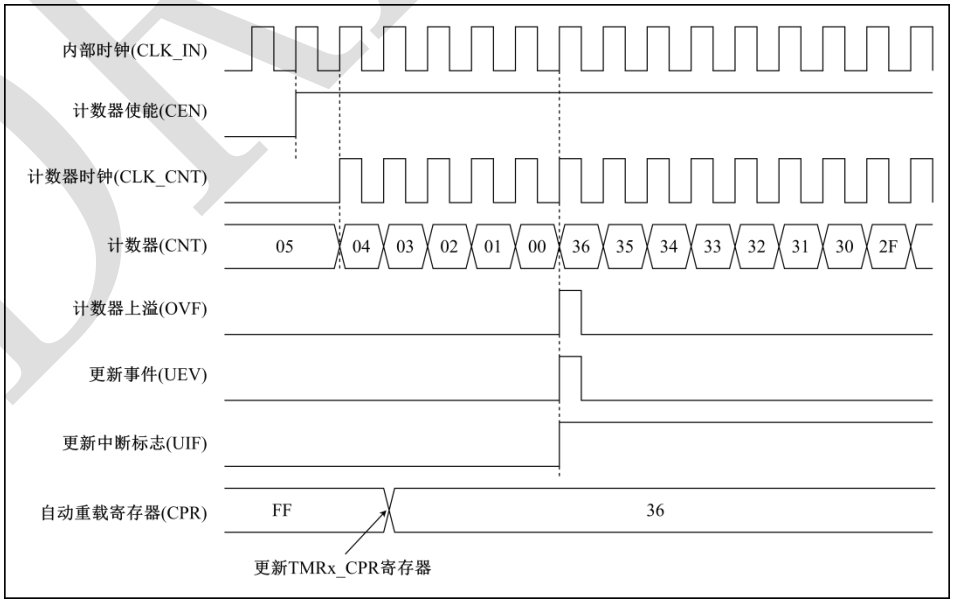


图 31-14　计数器时序图，运行中更新重载值（自动重载使能关闭-实时生效）

递增/递减计数模式

在中心对齐模式下，计数器从 0 开始计数到计数周期值(TMRx_CPR)–1，生成计数器上溢事件；然后从计数周期值(TMRx_CPR)开始向下计数到 1 并生成计数器下溢事件，之后又从 0 开始重新计数。

用户可以通过配置 TMRx_CR0 寄存器中的 CMS 位来选择中心对齐模式。在通道配置为输出模式时，比较输出的中断标志位会在下面各种模式下置 1：

- 仅在计数器递减计数（中心对齐模式 1，CMS=“01”）
- 仅在计数器递增计数（中心对齐模式 2，CMS =“10”）
- 在计数器递增/递减计数（中心对齐模式 3，CMS =“11”）

在中心对齐模式下，将无法修改方向位（TMRx_CR0 寄存器中的 DIR 位），该位是由硬件更新并指示当前计数器的方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TMRx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TMRx_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件，禁止之后将不会产生任何的更新事件，直到禁止更新位重新置 0。这样可避免向预装载寄存器写入新值时更新影子寄存器，不过在上溢事件 OVEV 后，计数器仍会根据当前重载周期值进行递增和递减计数，而预分频系数保持不变。

此外，如果 TMRx_CR0 寄存器中 UDIS 位置 1，然后通过软件将定时器 TMRx_UGR 寄存器中的 UG 位置 1，此时只会重新初始化计数器和预分频器，各影子寄存器内的值保持不变，也不会将更新中断标志位(UIF)置 1，因此不会发起任何中断。

定时器发生更新事件(UEV)时，将更新相关寄存器的值并将更新中断标志位(UIF)置 1(这同时取决于 URS 位)：

- 重复计数影子寄存器将更新为预装载寄存器(TMRx_CRR)的值
- 计数周期影子寄存器将更新为预装载寄存器(TMRx_CPR)的值
- 预分频器影子寄存器将更新为预装载寄存器(TMRx_PSCR)的值

下图将通过一些示例说明当计数器在不同时钟频率下表现的行为。

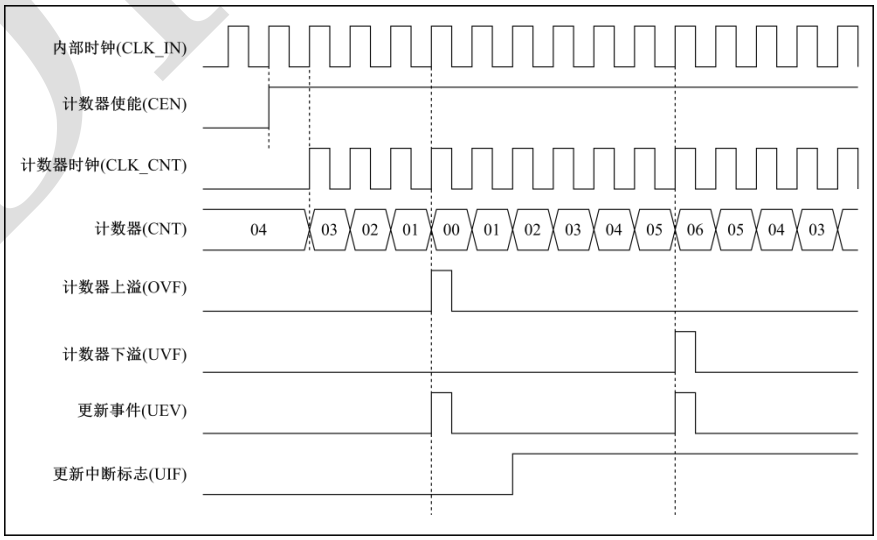


图 31-15　计数器时序图，1 分频内部时钟

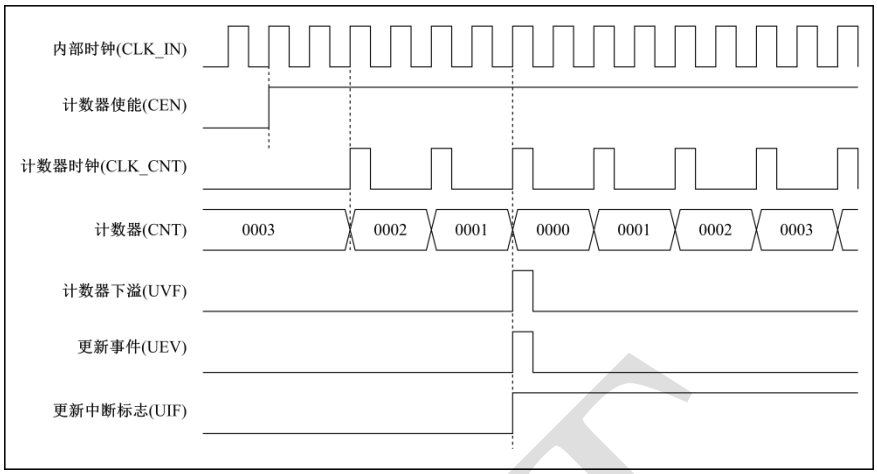


图 31-16　计数器时序图，2 分频内部时钟

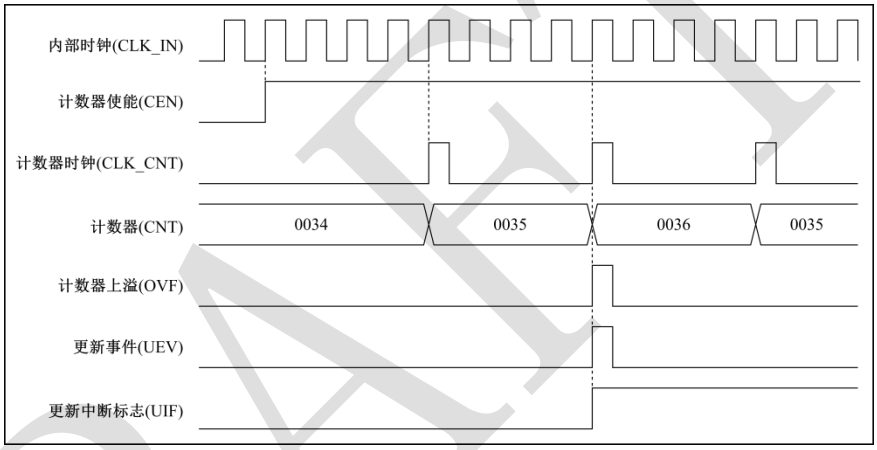


图 31-17　计数器时序图，4 分频内部时钟

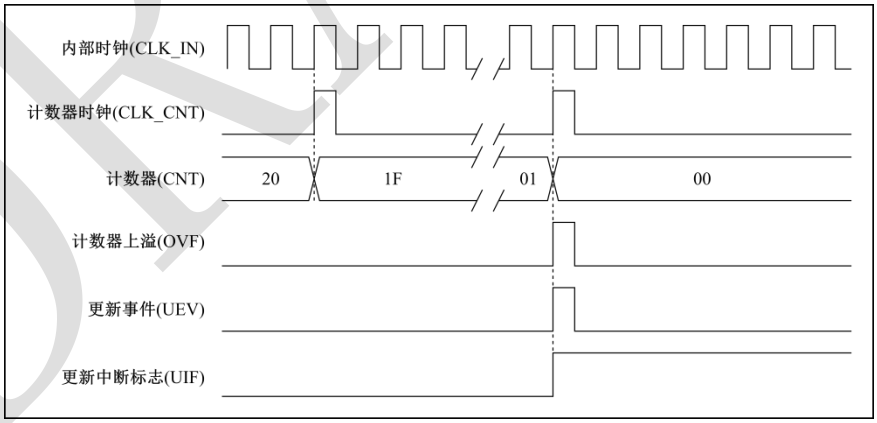


图 31-18　计数器时序图，N 分频内部时钟

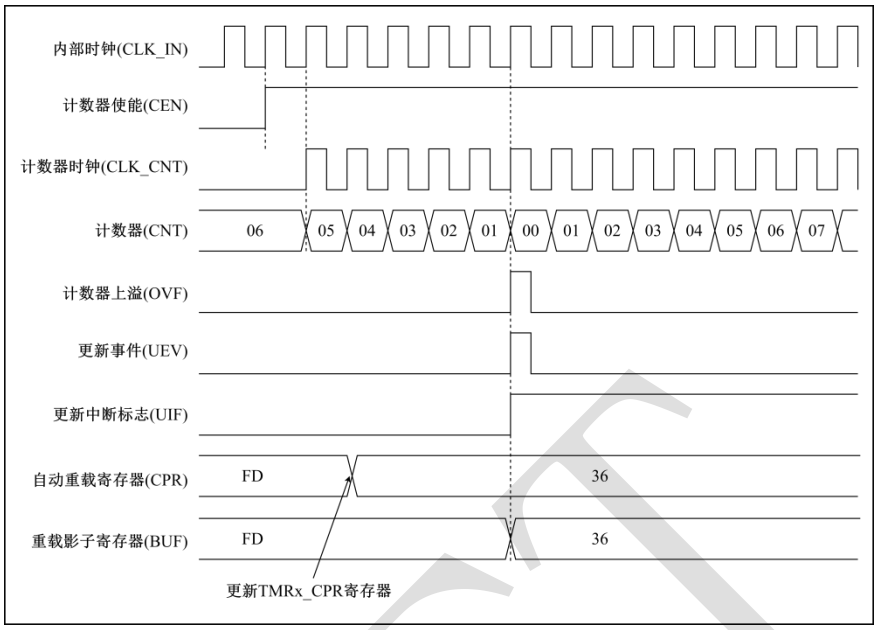


图 31-19 计数器时序图，运行中更新重载值（重载已使能-更新事件生效-计数器下溢）

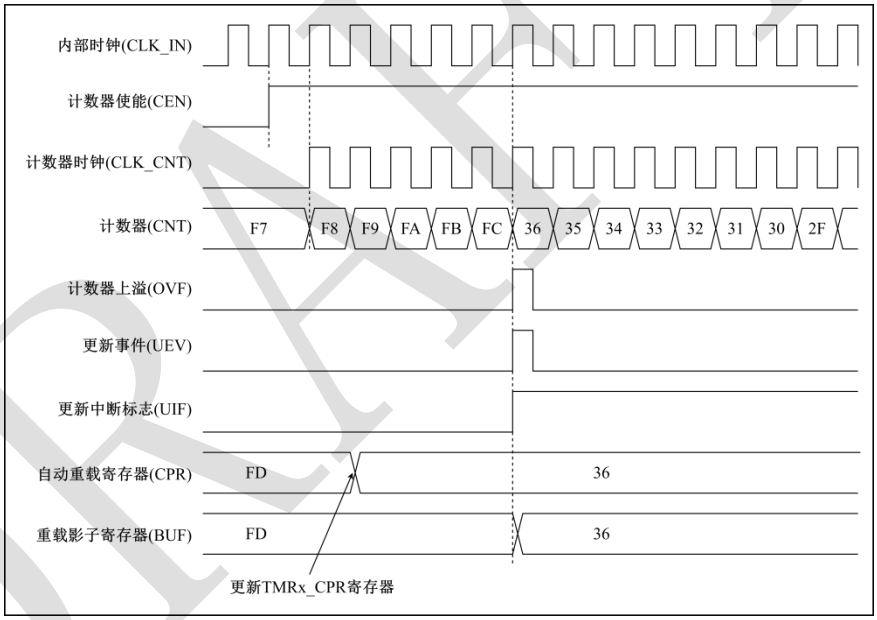


图 31-20 计数器时序图，运行中更新重载值（重载已使能-更新事件生效-计数器上溢）

31.4.4 重复计数器

在“31.4.2 章节：时基单元”中，介绍了计数器上溢时会生成更新事件（UEV）。当引入重复计数功能后，只有当重复计数器达到零时，才会生成更新事件（UEV），其对 PWM 信号的生成很有用。

这意味着，仅当定时器发生 N+1 个计数器上溢（N 为 TMRx_CRR 重复计数寄存器中的值）才生成一次更新事件（UEV），将触发重复计数影子寄存器（从 TMRx_CRR 加载）、计数周期影子寄存器（从 TMRx_CPR 加载）及预分频器影子寄存器（从 TMRx_PSCR 加载）的自动重载。

重复计数器在下列情况下递减：

- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中心对齐模式下每个计数器上溢和计数器下溢。

重复计数器是自动重载类型，其重复加载频率为 TMRx_CRR 寄存器中定义的值（请参见下图）。当更新事件由软件（通过将 TMRx_UGR 寄存器的 UG 位置 1）或硬件生成（通过从模式控制器）时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TMRx_CRR 寄存器的值。

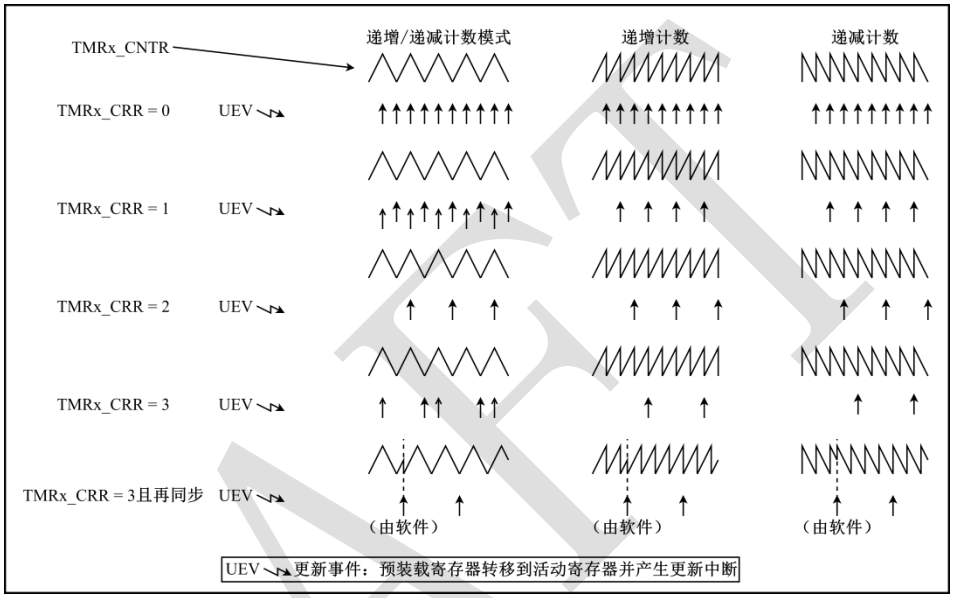


图 31-21　TMRx_CRR 重复计数器设置下的更新频率示例

31.4.5　时钟选择

定时器的计数器时钟源分为以下几类：

- 内部时钟源（CLK_IN）：系统时钟上升沿
- 外部时钟模式：通过外部输入引脚（TIx）
- 内部触发输入（ITRx）：连接其他定时器的触发输出

内部时钟源（CLK_IN）

内部时钟源是 TMRx 定时器的默认时钟源。

将 TMRx_SCR 寄存器中的 SMS 位置 0x0，则禁止从模式控制器，定时器默认选择内部时钟源作为计数时钟。然后，将 TMRx_CR0 寄存器中的 CEN 位和 TMRx_UGR 寄存器中的 UG 位用作控制位，当对 CEN 位写 1 时，预分频器的时钟就由内部时钟 CLK_IN 提供；通过软件对 UG 位写 1，则会使预分频器清零并重新开始计数。

下图为正常模式下内部控制逻辑及计数器计数的行为（预分频设置为 0，即在没有预分频情况下）。

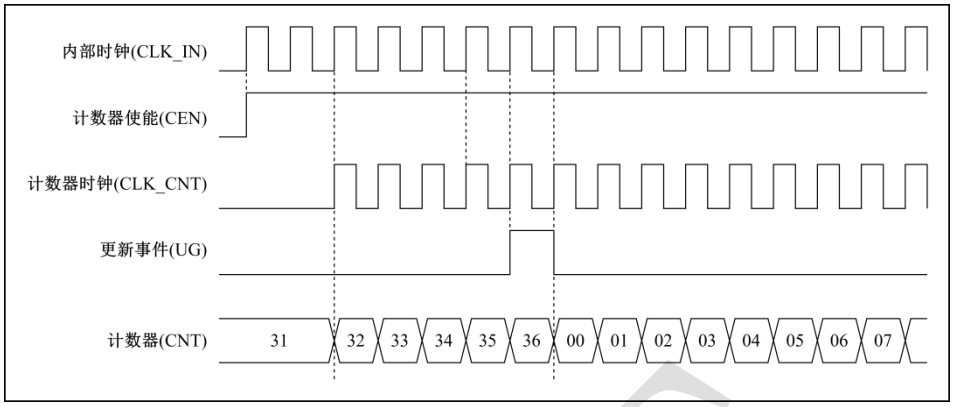


图 31-22　正常模式下的控制时序图（没有预分频情况）

外部时钟源模式 1

当 TMRx_SCR 寄存器中的 SMS 位为“111”时，则选择外部时钟源模式。在该模式下，计数器将在选定输入信号的上升沿或下降沿计数。

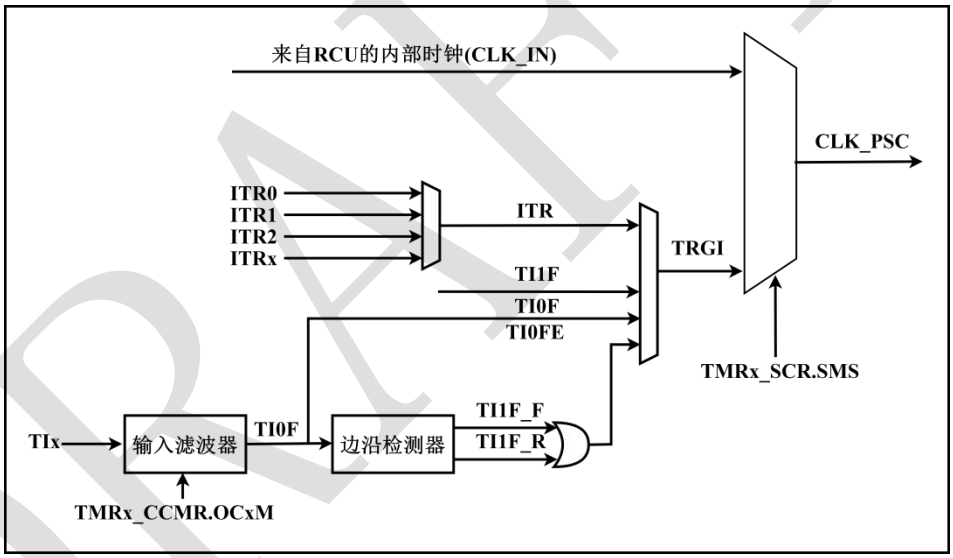


图 31-23　外部时钟连接示意图 1

定时器可选取外部时钟源作为计数器的时钟源，外部时钟源可来源于外部输入引脚或内部触发输入，外部输入引脚先经过内部滤波电路，然后通过内部边沿检测电路后作为计数器的触发时钟。

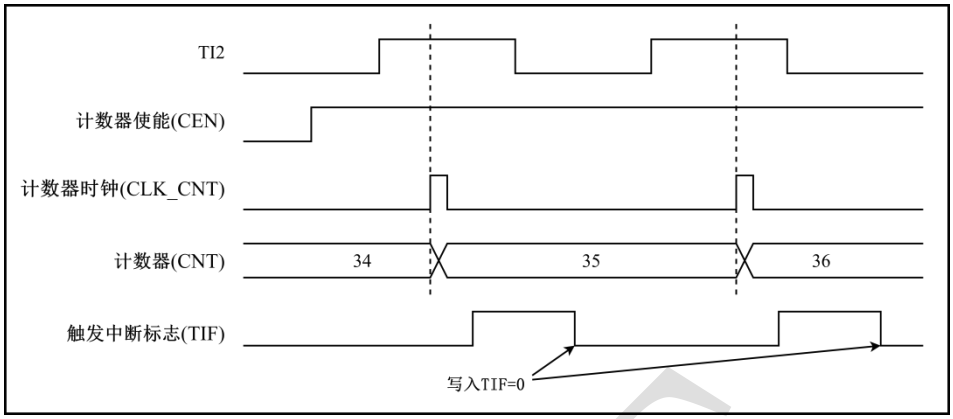


图 31-24　外部时钟模式下的控制电路

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 触发标志位置 1，TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入经过同步电路引起的。

外部时钟源模式 2

通过将 TMRx_SCR 寄存器中的 EE 位置 1 来选择外部时钟源模式 2，在这种模式下，计数器在外部输入 ETR 的边沿进行计数，通过 TMRx_SCR 寄存器中的 EMS 位来设定触发边沿，分为以下几种模式：

- 上升沿计数
- 下降沿计数
- 双边沿计数

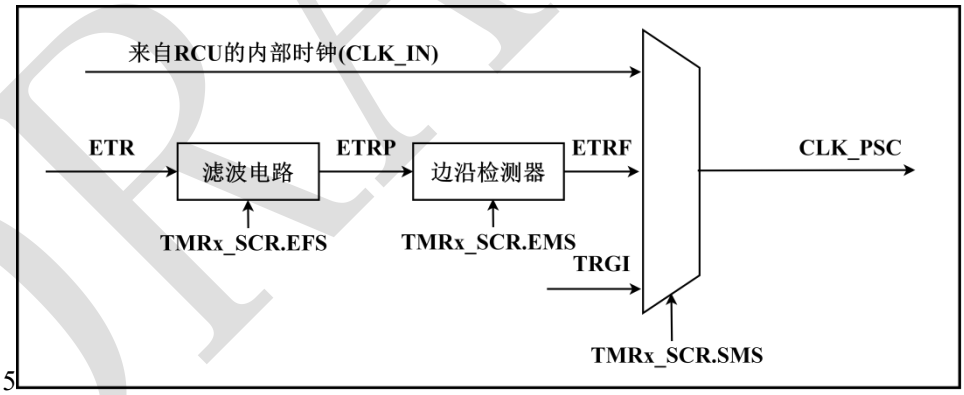


图 31-25　外部时钟连接示意图 2

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的，如下图所示。

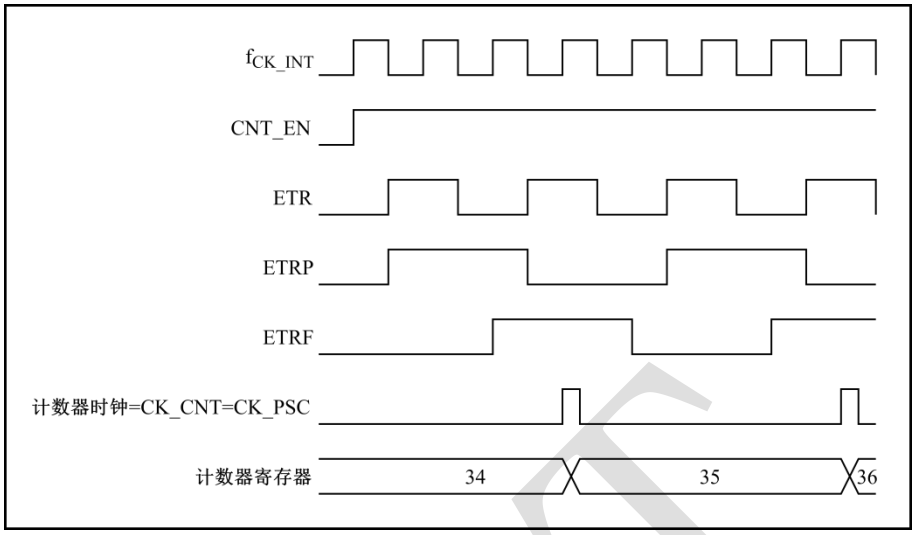


图 31-26　外部时钟模式 2 下的控制电路

注意：外部输入时钟源从功能引脚输入，可通过 GPIO 滤波器或 ETR 自有滤波器来实现消抖，GPIO 滤波器的消抖使能在 GPIO 模块中的“去抖使能寄存器(GPIOx_DER)”中设置。开启滤波器功能后，只有当输入电平宽度达到至少 4 个消抖时钟周期才会被送出，否则会被当作信号抖动被滤波器滤掉。

31.4.6　捕获/比较通道

定时器的每个捕获/比较通道均基于一个捕获/比较寄存器(包括影子寄存器)、一个捕获输入(滤波器、极性边沿检测、多路复用和预分频器)和一个比较输出(比较器和输出控制)组成。

输入捕获是对相应的输入信号 TIx 进行采样，经过滤波器后输出 TIxF，然后，再经过极性选择、边沿检测生成一个触发信号（TIxFE），该信号作为从模式控制器的输入，也可用作捕获信号输入。该信号先经过预分频电路（ICxPS），然后进入到捕获寄存器，如下图。

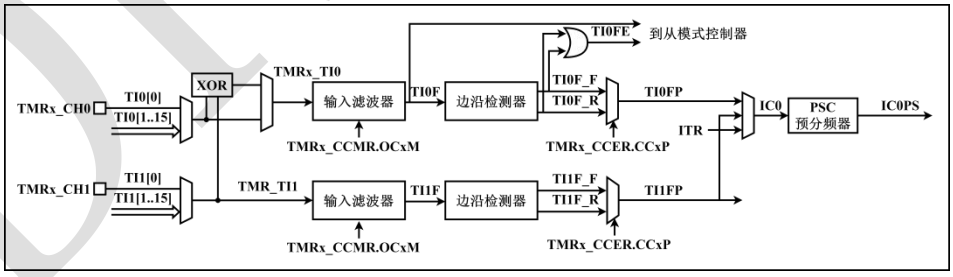


图 31-27　输入捕获通道示意图

输入捕获用于在捕获到指定边沿发生时，锁存计数器的准确数值。通过这种机制可实现对外部信号的准确捕获和测量，由此可实现对信号的脉宽测量、周期测量及占空比计算等功能。

输出比较是利用计数器计数，并与预先配置好的比较值进行对比，当计数值与比较值匹配时，根据所设定的比较输出模式进行相应的波形输出，从而实现例如 PWM、反转、强制极性输出等功能。

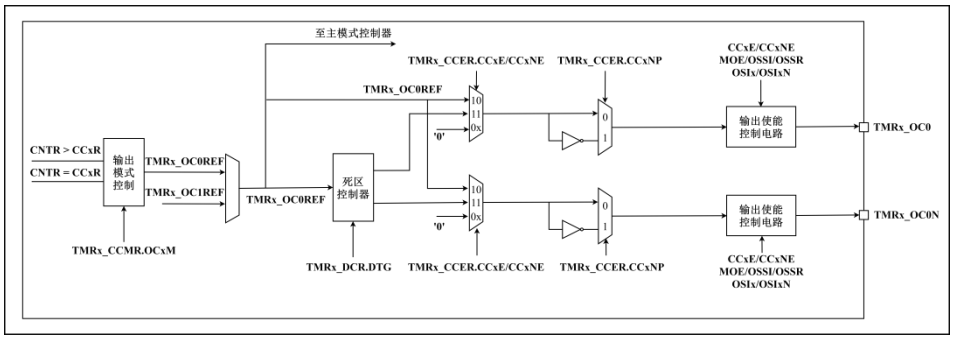


图 31-28　比较输出通道示意图

- 在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

31.4.7　输入捕获

在输入捕获模式下，当在对应的 ICx 信号上检测到边沿跳变后，会将计数器的值锁存到捕获/比较寄存器(TMRx_CCxR)中。当发生捕获事件时，会将 TMRx_SR 寄存器中相应的 CCxMIF 标志位置 1，并触发定时器中断(如果对应中断已使能)。如果在发生捕获事件时 CCxMIF 标志位已经被置位，则会将 TMRx_SR 寄存器中相应的 CCxOIF 重复捕获标志位置 1。可通过软件对捕获标志位 CCxMIF 及重复捕获标志位 CCxOIF 写 1 来清零，或读取存储在 TMRx_CCxR 寄存器中的捕获数据时。

通过配置 TMRx_CCMR 寄存器中的 CCxS 位来选择捕获输入通道，并将 TMRx_CCER 寄存器中的 CCxE 位置 1，就可以使定时器工作在输入捕获模式下。

通过配置 TMRx_CCER 寄存器中的 CCxP 位来选择捕获模式下的触发边沿，配置 TMRx_CIR 寄存器的 CxTIS 位可选择捕获的引脚输入源或比较器模块(CMPx_OUT)的内部输出信号。

当定时器工作于输入捕获模式下，在检测到捕获输入上产生边沿跳变时，定时器会将当前计数器的值锁存捕获到捕获/比较寄存器(TMRx_CCxR)内，同时将 TMRx_SR 寄存器中的输入捕获中断标志位(CCxMIF)置 1，并触发定时器中断(如果对应中断已使能)，软件上需要及时清除输入捕获中断标志 (对 TMRx_SR 寄存器中的 CCxMIF 写 1，或读取 TMRx_CCxR 寄存器，硬件将自动清除标志位)。如果在清除该标志位之前，再次捕获到输入跳变沿，会将新的计数值锁存并覆盖之前的值，同时会将 TMRx_SR 寄存器中的输入重复捕获标志位(CCxOIF)置 1，并触发定时器中断(如果对应中断已使能)。因此在处理输入捕获时，建议在捕获状态标志位置起后应当及时读取数据，否则一旦发生重复捕获，新捕获的数据将覆盖之前的数据，导致捕获信息丢失。

31.4.8　比较输出

配置 TMRx_CCMR 寄存器中的 CCxS 位来选择比较输出模式，配置 TMRx_CCER 寄存器中的 CCxE 位来使能比较输出功能，使定时器工作在输出比较模式下。

配置 TMRx_CCMR 寄存器中的 OCxM 位来选择比较输出的工作模式，配置 TMRx_CCMR 寄

存器中的 CCxPE 位来使能比较输出的预装载功能，配置 TMRx_CCER 寄存器中的 CCxP 位来选择比较模式下有效电平的极性。

当定时器的计数器(TMRx_CNTR)值与捕获/比较寄存器(TMRx_CCxR)值相匹配时，根据比较输出的工作模式在引脚上输出相应的波形，同时会将 TMRx_SR 寄存器中的 CCxMIF 位置 1，并触发定时器中断(如果对应中断已使能)，软件上可通过对 CCxMIF 位写 1 进行清除。

输出工作模式

定时器支持的比较输出工作模式有以下几种，通过 TMRx_CCMR 寄存器中的 OCxM 位进行配置：

- 0000：冻结，输出比较寄存器与计数器进行比较不会对输出造成任何影响（保持原有状态）
- 0001：当计数器与捕获/ 比较寄存器匹配时，输出有效电平（高电平）
- 0010：当计数器与捕获/ 比较寄存器匹配时，输出无效电平（低电平）
- 0011：当计数器与捕获/ 比较寄存器匹配时，发生翻转
- 0100：强制变为无效电平（低电平）
- 0101：强制变为有效电平（高电平）
- 0110：PWM 模式 1 - 当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态（高有效）
- 0111：PWM 模式 2 - 当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态（高有效）
- 1000：可再触发 OPM 模式 1 - 在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。
- 1001：可再触发 OPM 模式 2 - 在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。
- 1100：组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同，参考是 OC1REF 或 OC2REF 的结果
- 1101：组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同，参考是 OC1REF 和 OC2REF 的结果。
- 1110：不对称 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。
- 1111：不对称 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

配合定时器中的单次/连续计数模式，还可实现更多组合功能，例如利用单次计数模式+输出比较 PWM 模式可实现单脉冲输出功能，更多特别模式说明将在后续章节进行更详细介绍。

自动装载功能

在比较输出模式中，软件可通过修改捕获/比较寄存器(TMRx_CCxR)来调整比较值，通过配置 TMRx_CCMR 寄存器中的 OCxPE 位来使能比较预装载功能，自动装载的作用如下：

- 不开启预装载功能，向 TMRx_CCxR 寄存器写入新值，将立即更新到内部影子寄存器并生效。
- 开启预装载功能，向 TMRx_CCxR 寄存器写入新值，并不会立即生效，而是在下一个更新事件之后，再将 TMRx_CCxR 寄存器值更新到影子寄存器并生效。更详细的捕获/比较事件描述，请参考“31.4.14.5 捕获/比较事件”章节说明。

31.4.8.1 强制输出

在比较输出模式（TMRx_CCMR 寄存器中的 CCxS 位等于 0x0）下，可直接通过软件将比较输出信号（OCxREF）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的比较结果。

配置强制输出模式，只需将 TMRx_CCMR 寄存器中的 OCxM 位配置为 100 或 101，如对该位写入 101 后，OCxREF 将强制为高电平（OCxREF 始终为高电平有效），而输出 OCx 会根据 CCxP 极性位选择合适的有效电平。

在强制输出模式下，虽然定时器引脚输出固定电平，但定时器的计数器值与比较值的匹配行为依然在执行，在匹配时依然会引发相应的标志位并触发中断。

- 强制输出无效电平 (OCxM=100)时：
  - CCxP=0（有效电平为高电平），将比较强制输出为低电平
  - CCxP=1（有效电平为低电平），将比较强制输出为高电平
- 强制输出有效电平 (OCxM=101)时：
  - CCxP=0（有效电平为高电平），将比较强制输出为高电平
  - CCxP=1（有效电平为低电平），将比较强制输出为低电平

31.4.8.2 匹配输出模式

匹配输出模式可用于控制输出波形，或指示时间周期的变化。

输出引脚的电平由匹配输出模式（TMRx_CCMR 寄存器中的 OCxM 位）和输出极性（TMRx_CCER 寄存器中的 CCxP 位）定义。当定时器的计数器值与比较值相匹配时，输出比较匹配功能：

- 输出引脚即可保持原有电平（OCxM=0000）
- 配置为有效电平（OCxM=0001）
- 配置为无效电平（OCxM=0010）
- 配置为进行翻转（OCxM=0011）

匹配时，会将 TMRx_SR 中断状态寄存器中的 CCxMIF 置位，并触发定时器中断(如果对应中断已使能)。

通过配置 TMRx_CCMR 寄存器中的 CCxPE 位，可选择 TMRx_CCxR 寄存器的自动加载功

能的开启。匹配输出模式也可用作为单脉冲输出（在单脉冲模式下）。

31.4.8.3 PWM 输出模式

比较输出的 PWM 模式，可以用于生成一个调制信号，该信号频率周期由 TMRx_CPR 周期值寄存器决定，而占空比则由捕获/比较寄存器(TMRx_CCxR)设定的值决定，比较输出的 PWM 模式有两种：

- PWM 模式 1（OCM=0110）
  当计数 Counter 值小于 Compare 值时，输出有效状态，否则输出无效状态（高有效）。
- PWM 模式 2（OCM=0111）
  当计数 Counter 值小于 Compare 值时，输出无效状态，否则输出有效状态（高有效）。

OCx 极性可通过 TMRx_CCER 寄存器的 CCxP 位来设置，既可以设为高电平有效，也可以设为低电平有效，OCx 输出通过将 TMRx_CCER 寄存器中的 CCxE 位置 1 来使能，有关其详细信息，请参考 TMRx_CCER 寄存器说明。

因为计数器采用的递增方式计数，所以定时器为边沿对齐模式下生成 PWM。

边沿对齐模式

下图以 PWM 模式 1 为例，只要 TMRx_CNTR 计数器值小于 TMRx_CCxR 比较值，PWM 参考信号 OCxREF 为高电平，否则为低电平。如果 TMRx_CCxR 比较值大于 TMRx_CPR 重载周期值，则 PWM 参考信号 OCxREF 为高电平。如果 TMRx_CCxR 比较值为 0，则 PWM 参考信号 OCxREF 保持低电平。下图为边沿对齐的 PWM 波形，其中 TMRx_CPR 值为 8。

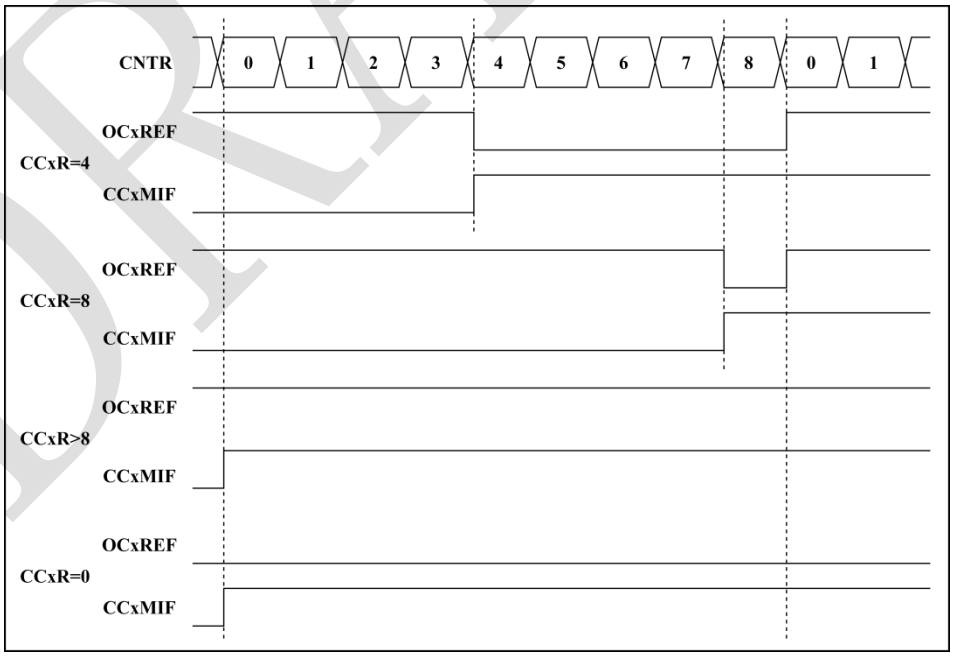


图 31-29　PWM 输出（CPR=8，CCxR=4，CCxP=0 高电平有效）

中心对齐模式

通过配置 TMRx_CR0 寄存器中的 CMS 位来选择中心对齐模式，在通道配置为输出模式时，

根据 CMS 位的配置，在计数器递减计数、计数器递增计数或计数器同时递增/递减计数时将比较输出的中断标志位置 1，并且 TMRx_CR0 寄存器中的 DIR 方向位由硬件更新，软件不能修改。

下图显示了中心对齐模式下的 PWM 波形，其中，TMRx_CPR=8，PWM 模式配置为 PWM 模式 1，根据 TMRx_CR0 寄存器中的 CMS 位而选择中心对齐模式 1，当计数器递减计数时，比较标志位置 1。

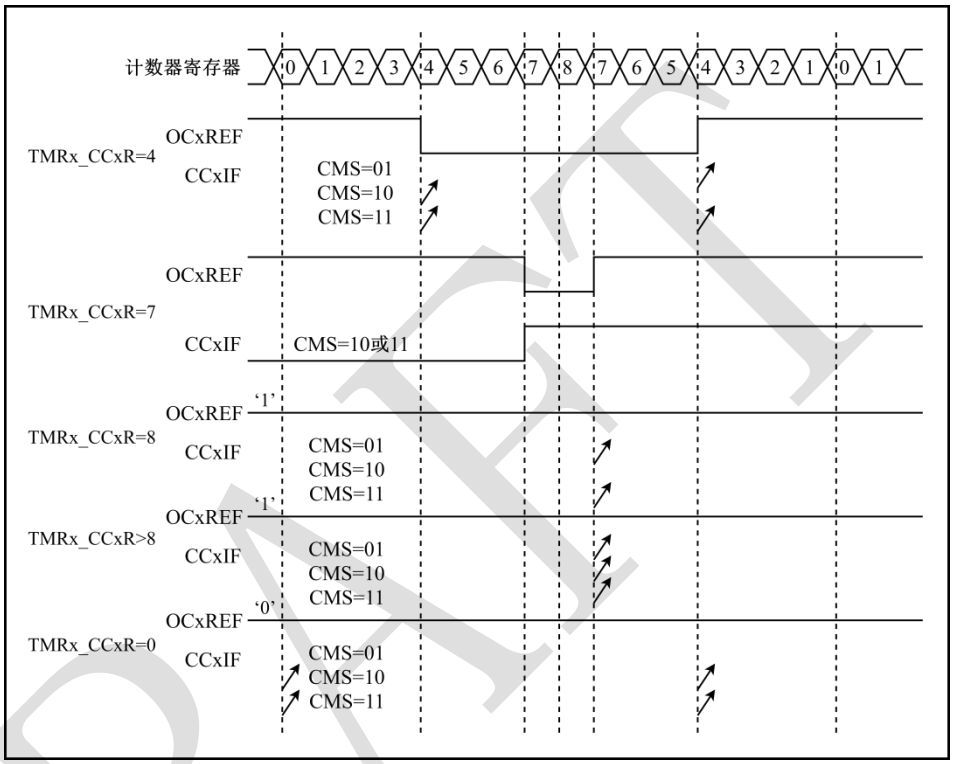


图 31-30　中心对齐模式 PWM 波形（CPR=8）

中心对齐模式使用建议：

- 在中心对齐模式下，TMRx_CR0 寄存器中的 DIR 方向位由硬件更新，软件不能修改。
- 在中心对齐模式运行过程中，不建议对计数器执行写操作，否则将发生意想不到的结果，尤其是：
  - 如果写入计数器中的值大于计数周期重载值 (TMRx_CNTR>TMRx_CPR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TMRx_CPR 的值，计数方向会更新。

31.4.8.4　单脉冲模式

单脉冲模式(One-Pulse Mode)是单次计数模式+PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号下启动，并在一段可编程的延时后输出一个可编程宽度的单脉冲信号。

可通过从模式控制器启动计数器，在匹配输出模式或 PWM 模式下生成波形，将 TMRx_CR0 寄存器中的 OPM 位置 1，即可使能单脉冲模式。这样，在发生下一个更新事件 UEV 时，计数器将自动停止。

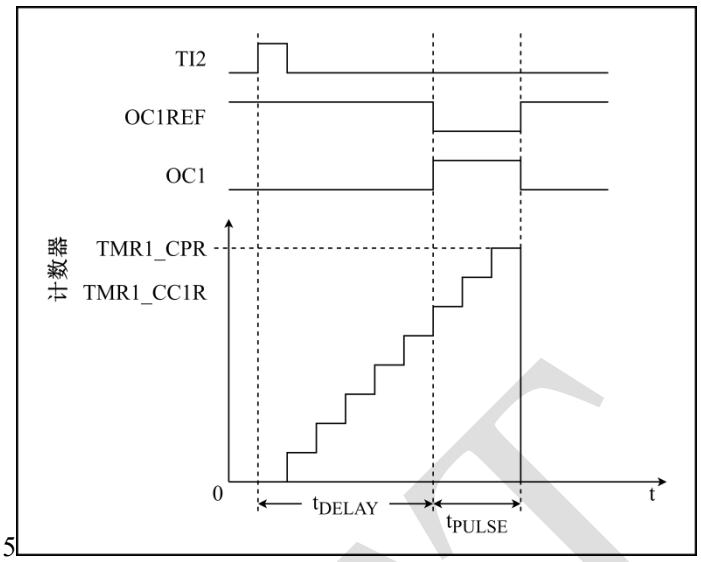


图 31-31　单脉冲模式示例

上图效果为：在 TI2 输入引脚上检测到上升沿时，经过 t_{DELAY} 的延迟后，将在 OC1 上输出一个长度为 t_{PULSE} 的正脉冲波形。

单脉冲的波形可通过调整 TMRx_CCxR 比较寄存器的值来定义（需考虑时钟频率及计数器预分频）：

- t_{DELAY} 时间由写入 TMRx_CCxR 的值来定义
- t_{PULSE} 时间由重载周期值与比较值（TMRx_CPR-TMRx_CCxR）之差来定义

当 TMRx_CR0 寄存器中的 OPM 位写入“1”，即使能单脉冲模式，选择合适的输出工作模式（匹配输出或 PWM 模式 1/2），对捕获/比较寄存器(TMRx_CCxR)写入合适的比较值，根据需求配置有效电平的极性，计数器在发生下一个更新事件（计数器从重载周期值返回到 0）时计数器将停止计数，并输出一段可控宽度的单脉冲。当 TMRx_CR0 寄存器中的 OPM 位写入“0”，即选择重复连续模式。

31.4.9　互补输出

定时器可以输出两路互补信号，并可通过死区控制器来管理比较输出的关与开。对于死区时间的设置，用户需根据外部器件及其特性（电平转换器的固有延迟、开关器件产生的延迟等）来调整死区时间的长短。

每路输出都可独立选择输出极性（主输出 OCx 或互补输出 OCxN），通过配置 TMRx_CCER 寄存器中的 CCxP 和 CCxNP 位来选择输出极性。其中，互补信号 OCx 和 OCxN 是通过以下多个控制位的组合来控制：TMRx_CCER 寄存器中的 CCxE 和 CCxNE 位、TMRx_DCR 寄存器中的 MOE、OSSI 和 OSSR 位及 TMRx_CR1 寄存器中的 OISx、OISxN 位。更多详细信息，请参考“表 31-8：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位”。

当 TMRx_CCER 寄存器中的 CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1 时，将使能死区插入功能。其中，TMRx_DCR 寄存器中的 DTG 位用于控制所有通道的死区时间设置。死区控制器将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，其上升沿相对参考上升沿存在延迟
- 输出信号 OCxN 与参考信号相反，其上升沿相对参考下降沿存在延迟

如果死区时间大于有效输出（OCx 或 OCxN）的宽度，则不会生成相应的脉冲。

下图为死区发生器的输出信号与参考信号 OCxREF 之间的关系（示例中，设置 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

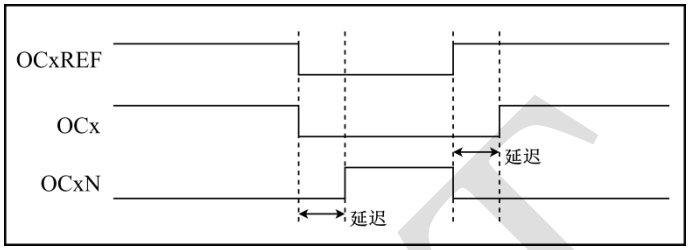


图 31-32　带死区插入的互补输出

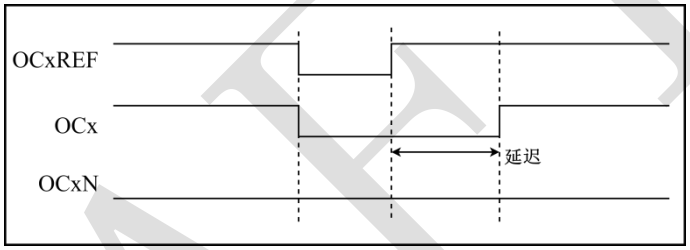


图 31-33　死区时间大于负脉冲宽度的波形

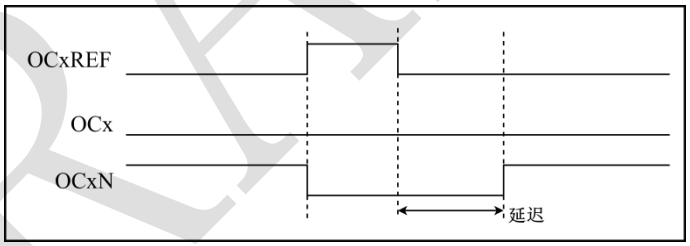


图 31-34 死区时间小于正脉冲宽度的波形

死区时间的设置作用于所有通道并均相同，可通过 TMRx_DCR 寄存器中的 DTG 位进行配置。有关延迟时间的计算信息，请参见 TMRx_DCR 寄存器说明。

表 31-8. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	0	1	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	OCxREF+极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		0	1	0	OCxREF+极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	1	1	OCREF+极性+死区	OCREF 互补项（而非 OCREF）+

					OCx_EN=1	极性+死区 OCxN_EN=1
		1	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP、OCx_EN=1	OCxREF+极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		1	1	0	OCxREF+极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF+极性+死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+极性+死区 OCxN_EN=1
0	0	X	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
	0		0	1	禁止输出（不由定时器驱动） 异步：OCx=CCxP、OCx_EN=0、OCxN=CCxNP、OCxN_EN=0 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应。	
	0		1	0		
	0		1	1		
	1		0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
	1		0	1	禁止输出（不由定时器驱动） 异步：OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应。	
	1		1	0		
	1		1	1		

31.4.10 断路保护

使用断路功能时，根据 TMRx_DCR 寄存器中的 MOE 位、OSSI 位和 OSSR 位及 TMRx_CR1 寄存器中的 OISx 位、OISxN 位来控制输出使能及输出电平。任何情况下，OCx 和 OCxN 输出都不能同时设为有效电平。更多详细信息，请参考“表 31-8：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位”。

断路功能的输入源可以是断路输入引脚，也可以是时钟故障事件，后者由时钟复位控制器 RCU 中的时钟安全系统 (CSS) 生成。有关时钟安全系统的详细信息，请参考“第 7.2.6 节：时钟安全系统 (CSS)”。

芯片上电复位后，断路功能默认处于禁止状态，MOE 位为低电平。将 TMRx_DCR 寄存器中的 BKxE 位置 1 可使能断路功能，断路输入的极性可通过 TMRx_DCR 寄存器中的 BKxP 位来选择，BKxE 和 BKxP 位可同时修改。

当发生断路（断路输入上出现所设定的电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态或复位状态（通过 OSSI 位进行选择）。
- MOE=0 时，将以 TMRx_CR1 寄存器中的 OISx 位的电平来驱动每个通道的输出。如果 OSSI=0，则定时器将关闭输出使能，否则输出使能始终保持高电平。
- 将断路状态标志（TMRx_SR 寄存器中的 BIF 位）置 1，如果 TMRx_IER 寄存器中的 BIE 位置 1，则可产生中断。

● 如果 TMRx_DCR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件(UEV)时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入“1”。这种情况下，这一特性可用于确保安全。可以切断功率器件、温度传感器或任何安全元件，保证其不被损坏。

注意：断路输入为电平有效，因此，当断路输入为有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将断路状态标志 BIF 清零。

断路事件可由外部 TMRx_BK 引脚输入生成，输入引脚的极性是可配置，其使能由 TMRx_DCR 寄存器中的 BKxE 位来控制。

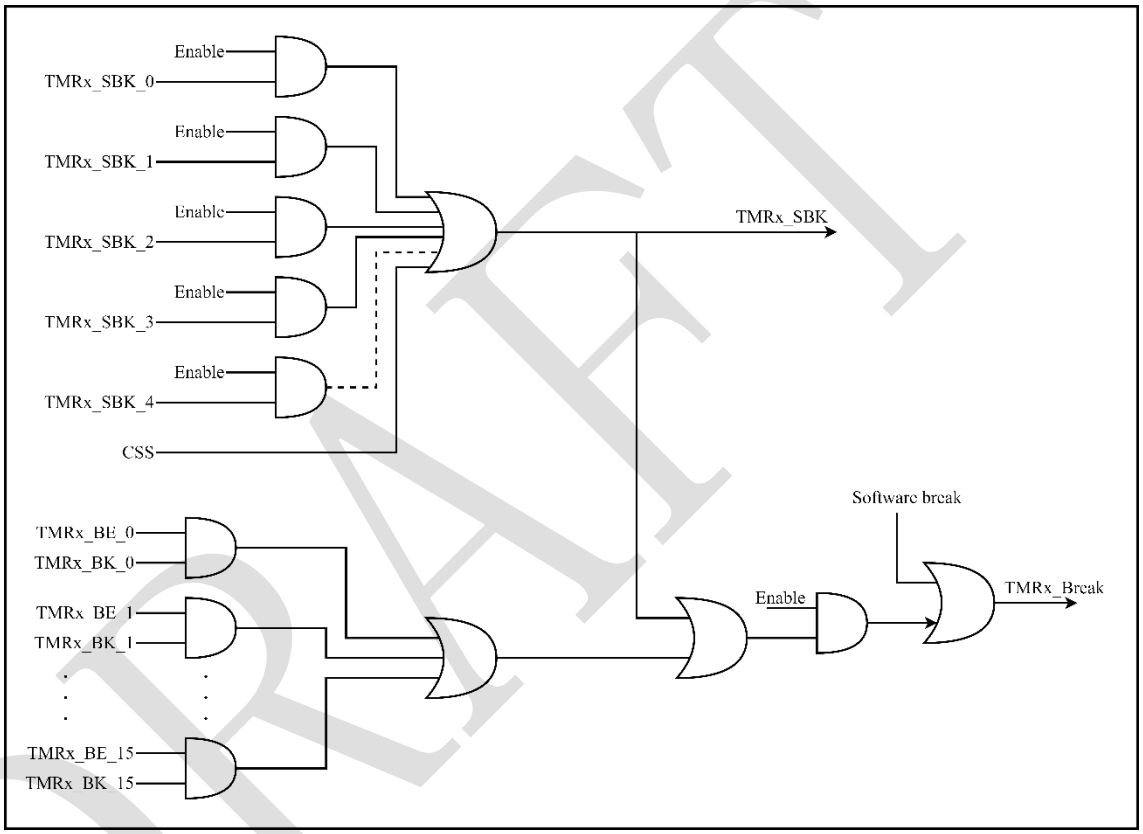


图 31-35　TMRx Break 输入控制示意图

断路事件有以下两种生成方案：

- 使用 TMRx_BK 引脚生成
- 由软件通过 TMRx_UGR 寄存器中的 BG 位生成

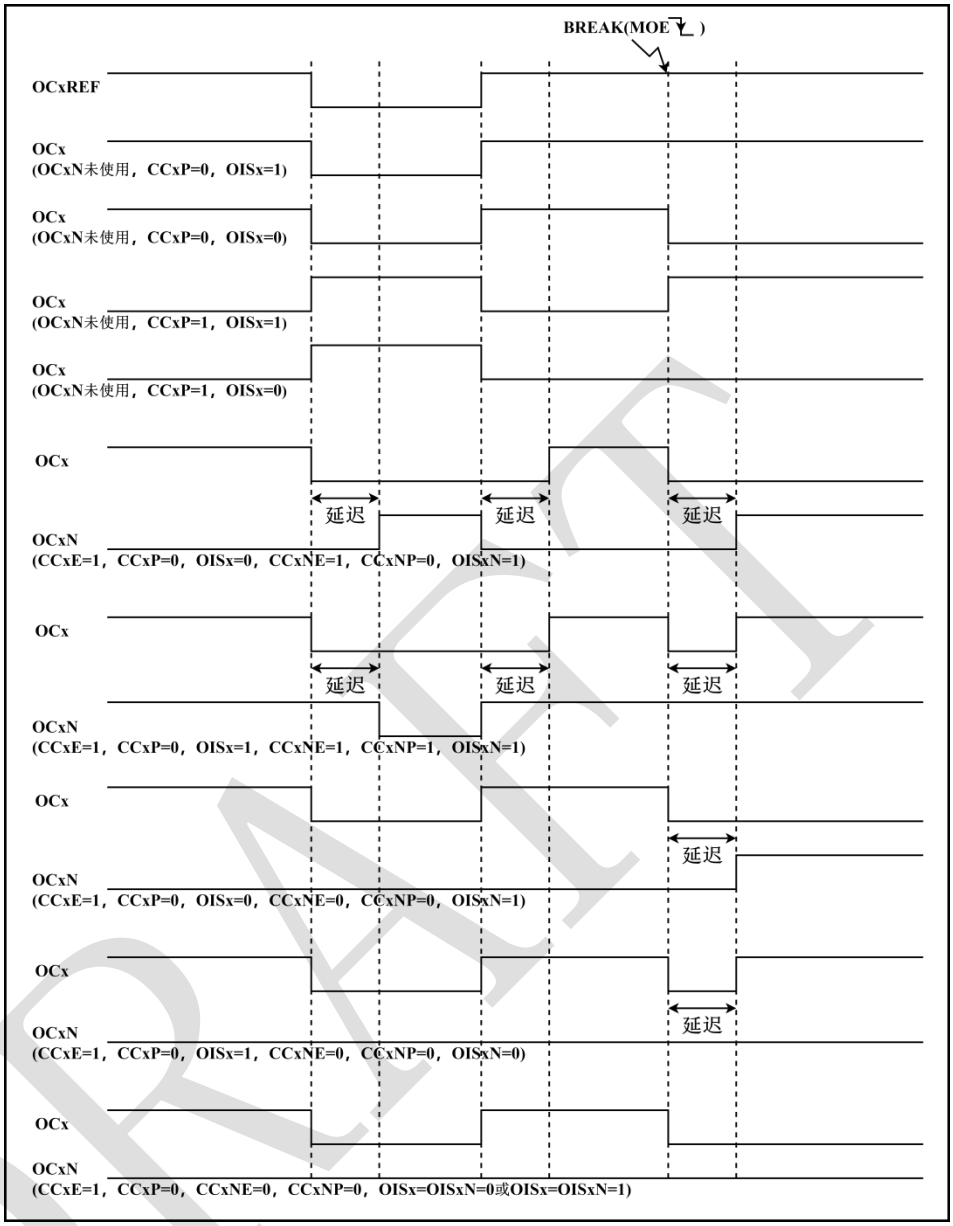


图 31-36　输出的断路响应行为

31.4.11　从模式

定时器可通过外部触发实现以下功能的同步：复位模式、门控模式和触发模式。

从模式：复位模式

当输入触发信号产生上升沿跳变时，计数器及其预分频器将被重新初始化。如果 TMRx_CR0 寄存器中的 URS 位为 0 时，则还会生成更新事件 UEV。然后，所有预装载寄存器 TMRx_CPR 和 TMRx_CCxR 都将更新。

以下示例中，在 TI1 输入信号上出现上升沿时，递增计数器清零：

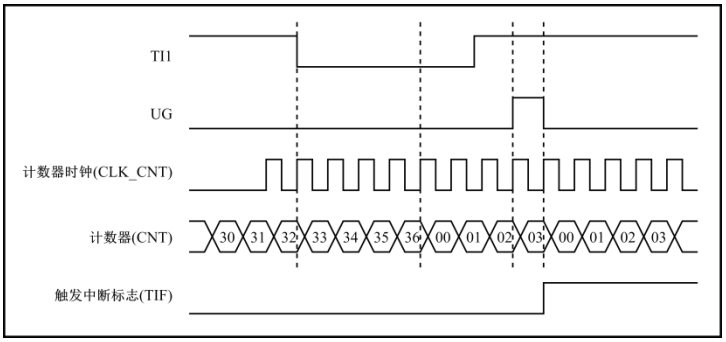


图 30-37　复位模式下的控制电路

计数器使用内部时钟计数，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发 TMRx_SR 寄存器中的 TIF 标准位置 1，使能对应的中断后，还可发送中断请求。

上图显示了自动重载寄存器 TMRx_CPR=0x36 时的行为，TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：门控模式

通过触发输入信号的电平来使能计数器，当输入信号为高电平时，计数器开始计数，当输入信号为低电平时，计数器保持不变。

下图中，递增计数器仅在 TI1 输入为高电平时计数：

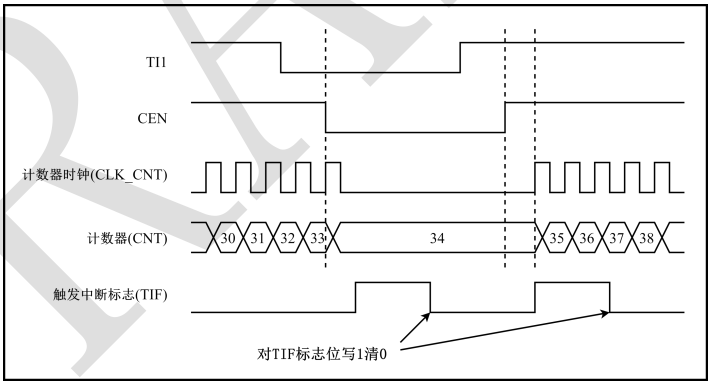


图 31-38　门控模式下的控制电路

只要 TI1 为高电平，计数器就开始根据内部时钟计数，当 TI1 变为低电平时停止计数。当计数器启动或停止时，TMRx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：触发模式

通过输入信号的上升沿来启动计数器，计数器开始计数器。

下图中，递增计数器在 TI1 输入上升沿时，启动递增计数器：

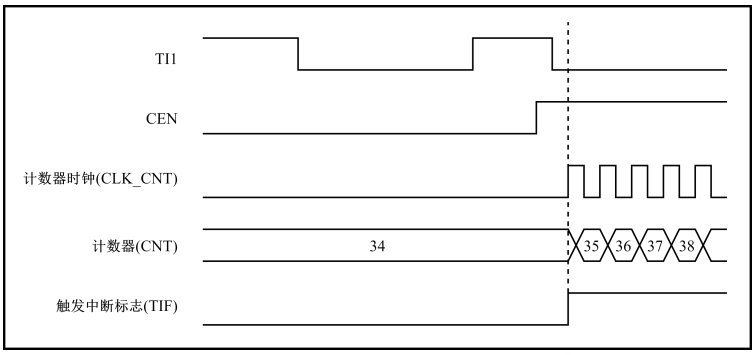


图 31-39　触发模式下的控制电路

当 TI1 出现上升沿时，计数器根据内部时钟开始计数，并且 TIF 标志位置 1。

TI1 的上升沿与实际计数器启动之间的延迟是由于 TI1 输入的重新同步电路引起的。

从模式：外部时钟模式 2+触发模式

外部时钟模式 2 可与另一种从模式结合使用。这种情况下，ETR 信号作为外部时钟输入，在复位模式、门控模式或触发模式下，可选另一个输入作为触发输入，不建议通过 TMRx_SCR 寄存器中的 TS 位来选择 ETR 作为 TRGI 的输入源。
在下图示例中，当 TI1 出现上升沿时将使能计数器，同时将 TIF 标志位置 1，然后计数器即会在 ETR 信号的每个上升沿递增计数。

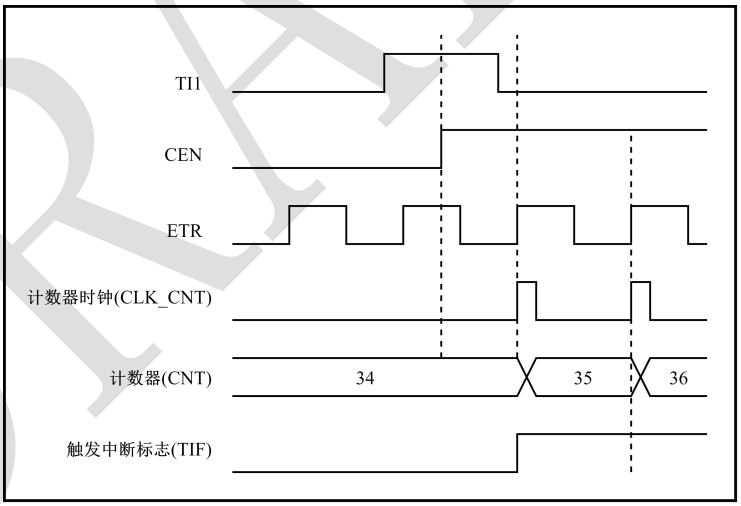


图 31-40 外部时钟模式 2+触发模式下的控制电路

ETR 信号的上升沿与实际计数器之间的延迟是由于 ETRP 输入的重新同步电路引起的。

31.4.12　定时器同步

定时器内部是连在一起，以实现定时器之间的同步与级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

下图简要介绍了从模式的触发选择及主模式选择框图，将一个定时器用作另一个定时器的预分频器：

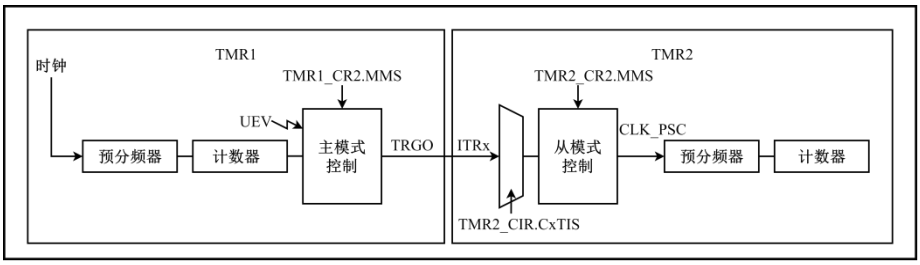


图 31-41　主/从定时器示例

注意：如果选择定时器 1 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动定时器 2 的计数器。

下图使用一个定时器 1 的输出来使能另一个定时器 2 的示例，相关的连接图如下，仅当定时器 1 的 OC1REF 位高电平时，定时器 2 才会根据分频后的时钟进行计数，两个计数器的时钟频率都是基于 3 分频。

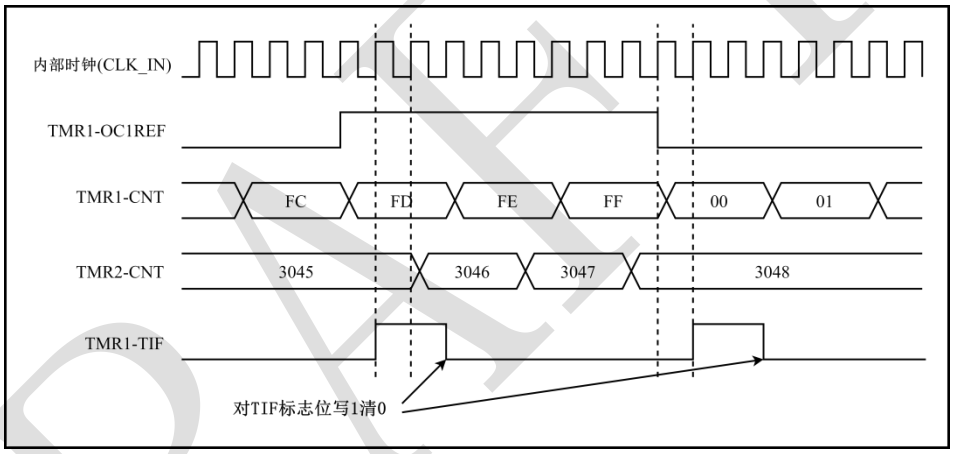


图 31-42　使用定时器 1 的输出来对定时器 2 实施门控控制

31.4.13　调试模式

当内核进入调试模式（Cortex™-M4 内核停止）时，定时器内部计数器会根据 SYSCTRL 模块 SYSDCR 系统调试配置寄存器中的 TMRxDEN 位来选择继续正常工作或者停止工作。

31.4.14　事件与中断

定时器在多种情况下会产生相应的事件，事件主要用于更新相应的影子寄存器、执行特定功能以及产生相应中断。

定时器内主要有以下几种事件：

- 定时器上溢事件，简称上溢事件 OVEV(Overflow Event)
- 定时器更新事件，简称更新事件 UEV (Update Event)
  需要使用更新事件，必须将定时器控制寄存器(TMRx_CR0)中的禁止更新位(UDIS)置 0
- 定时器输入捕获/输出比较事件，简称捕获/比较事件 CCEV (Capture/Compare Event)

- 定时器触发事件，简称触发事件 TEV (Trigger Event)
- 定时器断路事件，简称断路事件 BEV (Break Event)
  断路事件源包括断路引脚输入或系统故障输入。

中断主要包含以下五种中断触发：

- 定时器上溢中断，发生上溢事件后，上溢中断标志位(OVIF)同时被置位
- 定时器更新中断，发生更新事件后，更新中断标志位(UIF)同时被置位
- 定时器触发中断，发生触发事件后，更新触发标志位(TIF)同时被置位
- 定时器断路中断，发生断路事件后，更新断路标志位(BIF 或 SBIF)同时被置位

特别说明：如果更新请求源选择位(URS)置 1，即仅上溢事件会产生更新事件，那么软件上如果通过设置更新产生位(UG)，将只会产生更新事件用于更新预加载相关的寄存器，但不会触发定时器更新中断。

- 定时器输入捕获中断，当定时器为输入捕获模式并捕获到相应的边沿时，输入捕获中断标志位（CCxMIF）将被置位。
- 定时器重复捕获中断，当输入捕获中断标志位已经置位(CCxMIF =1)，再次捕获到相应边沿，则输入捕获重复捕获中断标志位（CCxOIF）将被置位。
- 定时器输出比较中断，当定时器位输出比较模式，并且计数值与比较值相匹配时，输出比较中断标志位(CCxMIF)将被置位。

31.4.14.1 上溢事件（Overflow Event）

上溢事件(OVEV)由以下情况产生：

- 计数器 (TMRx_CNTR)值到达计数周期寄存器(TMRx_CPR)设定的值。

上溢事件的作用

计数器到达上溢后，计数器寄存器会立即重新从 0 开始计数 (在连续模式下)。

事件产生后，中断状态寄存器(TMRx_SR)中的上溢中断标志位(OVIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

31.4.14.2 更新事件（Update Event）

更新事件(UEV)可以由以下两种情况产生：

- 计数器上溢：计数器上溢事件产生的同时会产生更新事件。
- 软件置位：软件将定时器事件产生寄存器(TMRx_UGR)中的更新标志位(UG) 置 1。

更新事件使用的注意事项

- 如果需要使用更新事件(UEV)，必须将定时器控制寄存器(TMRx_CR0)中的更新禁止位(UDIS)置 0，使能更新事件功能，否则更新事件将被禁止。
- 如果需要软件置位功能，还必须要配置定时器控制寄存器(TMRx_CR0)中的更新请求位(URS)置 0。否则只有计数器上溢才会产生更新事件。

更新事件的作用

事件产生后，中断状态寄存器(TMRx_SR)中的更新中断标志位(UIF)将会置 1，并且会触发定时器中断 (如果对应的中断已使能)。

特别的，如果在计数器计数的过程中(未达到计数器上溢)，通过软件设置 UG 位产生更新事件后，计数器寄存器也会立即重新从 0 开始计数(在连续模式下)。不同之处在于：通过软件更新标志位(UG)设置产生的更新事件，不会触发上溢事件的中断。

更新事件对自动装载预加载(Auto-Reload Preload)功能的作用

当定时器开启自动装载预加载功能（TMRx_CR0 寄存器内 ARE=1），更新事件产生后，才会将以下几个寄存器的值，更新到内部影子寄存器内使之生效：

- 计数周期寄存器(TMRx_CPR)
- 预分频寄存器(TMRx_PSCR)
- 捕获/比较寄存器(TMRx_CCxR)

注意：软件置位(UG=1)产生的更新事件，不会影响捕获/比较寄存器(TMRx_CCxR)。

如果定时器未开启自动装载预加载功能（TMRx_CR0 寄存器内 ARE=0），则上述寄存器写入新值将立即生效，更新事件将不影响上述寄存器的配置。

31.4.14.3 触发事件（Trigger Event）

触发事件(TEV)由以下情况产生：

- 在从模式下，输入信号上检测到预期的跳变事件
- 在从模式下，软件将事件产生寄存器(TMRx_UGR)中的事件产生标志位(TG)置 1

触发事件的作用

在从模式下，输入信号上检测到预期的跳变事件，中断状态寄存器(TMRx_SR)中的 Trigger 中断标志位(TIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

31.4.14.4 断路事件（Break Event）

断路事件(BEV)由以下情况产生：

- 在比较输出模式下，输入断路信号出现预定的电平信号
- 在从模式下，软件将事件产生寄存器(TMRx_UGR)中的事件产生标志位(BG)置 1

断路事件的作用

在比较输出模式下，输入断路信号出现预定的电平信号（断路功能已使能），即事件触发后，比较输出会根据用户设定的方式进行保护输出。

事件产生后，中断状态寄存器(TMRx_SR)中的上溢中断标志位(BIF 或 SBIF)将会置 1，并且会触发定时器中断(如果对应的中断已使能)。

31.4.14.5 捕获/比较事件（Capture/Compare Event）

捕获/比较事件(CCEV)，可以由以下三种情况产生：

- 比较成功：计数器 (TMRx_CNTR)的值与捕获/比较寄存器(TMRx_CCxR)的值比较成功时(仅针对输出比较模式下)
- 捕获成功：外部输入一个与 MRx_CCER 寄存器中 CCxP 位相符的边沿信号时（仅针对输入捕获模式下）
- 软件产生：将事件产生寄存器(TMRx_UGR)中的事件产生标志位(CCxUG)置 1

捕获/比较事件的作用

在不同模式下捕获/比较事件的作用不相同：

- 对于比较输出模式：
  - 捕获/比较事件产生后，中断状态寄存器(TMRx_SR)中的比较成功中断(CCxMIF)位将会置 1，并触发定时器中断(如果中断已使能)。
  - 如果使能了比较器预加载功能(TMRx_CCMR 寄存器中的 CCxPE 位置 1)，事件产生后(特指比较成功所产生事件)，定时器将会执行以下动作：
    - 定时器的输出，将根据 TMRx_CCMR 寄存器中的比较输出模式 OCxM 位，对输出状态进行相应的改变；
    - 将捕获/比较寄存器(TMRx_CCxR)的值重新加载到内部影子寄存器，使之生效。
    - 将定时器的计数器值(TMRx_CNTR)重新从 0 开始计数(连续循环模式)。
  - 如果没有使能比较器预加载功能(TMRx_CCMR 寄存器中的 CCxPE 位置 0)，事件产生后，定时器的动作与上述情况不同之处在于：对捕获/比较寄存器(TMRx_CCxR)写入新值将立即生效，捕获/比较事件不会影响该寄存器值何时生效。

注意：如果在到达比较值之前(未达到比较成功)，通过软件对 CCxUG 置 1 产生事件，与上述描述基本一致，不同之处在于：不会导致比较输出模式(OCM[2:0])所设定的输出变化，因为本次计数器值并没有与触发捕获/比较事件前所设定的比较值相匹配。

- 对于输入捕获模式：
  - 捕获/比较事件产生后，定时器将当前计数器 (TMRx_CNTR)的值捕获到捕获/比较寄存器(TMRx_CCxR)内；
  - 中断状态寄存器(TMRx_SR)中的捕获中断标志(CCxMIF)位将会置 1，并触发定时器中断(如果对应的中断已使能)。如果本次捕获成功产生捕获/比较事件时，前一次捕获中断标志未及时清除(CCxMIF=1)，则中断状态寄存器(TMRx_SR)中的重复捕获中断标志(CCxOIF)位将会置 1，并且触发定时器中断(如果对应的中断已使能)。

注意：如果通过软件对 CCxUG 置 1 产生的捕获/比较事件，则立刻执行上述动作，而不用外部产生符合捕获成功的信号。

31.4.14.6 中断

中断标志位与中断使能之间的关系如下表描述所示，详细内容请参考“31.4.14 事件与中断”章节描述：

表 31-9 中断标志位与中断使能之间的关系

中断名称	中断使能	中断标志	清除方式	备注
上溢中断	OVIE	OVIF	W1C	计数器计数上溢后产生
更新中断	UIE	UIF	W1C	计数器更新事件触发更新中断
比较捕获中断	CxMIE	CCxMIF	W1C 或读 CCxR	比较捕获模式下，计数值与比较值匹配或捕获到目标边沿
重复捕获中断	CxOIE	CCxOIF	W1C 或读 CCxR	输入捕获中断未及时清除，再次发生捕获
触发中断	TIE	TIF	W1C	在从模式下，输入信号出现目标触发边沿
断路中断	BIE	BIF	W1C	在比较输出下，断路信号出现有效电平
系统断路中断	SBIE	SBIF	W1C	在比较输出下，系统断路信号出现有效电平

31.5 寄存器定义

31.5.1 寄存器列表

TMRx 基地址：

TMR9(0x40030000) TMR10(0x40031000)

偏移	实例地址	名称	默认值	描述
0x00	TMRx 基地址+0x00	TMRx_CR0	0x00000000	TMRx 控制寄存器 0
0x04	TMRx 基地址+0x04	TMRx_CR1	0x00000000	TMRx 控制寄存器 1
0x08	TMRx 基地址+0x08	TMRx_SCR	0x00000000	TMRx Slave 控制寄存器
0x0C	TMRx 基地址+0x0C	TMRx_IER	0x00000000	TMRx 中断使能寄存器
0x10	TMRx 基地址+0x10	TMRx_SR	0x00000000	TMRx 状态寄存器
0x14	TMRx 基地址+0x14	TMRx_UGR	0x00000000	TMRx 更新事件生成寄存器
0x20	TMRx 基地址+0x20	TMRx_CCMR	0x00000000	TMRx 捕获/比较模式寄存器
0x24	TMRx 基地址+0x24	TMRx_CCER	0x00000000	TMRx 捕获/比较使能寄存器
0x28	TMRx 基地址+0x28	TMRx_DCR	0x00000000	TMRx 死区控制寄存器
0x2C	TMRx 基地址+0x2C	TMRx_TTCR	0x00000077	TMRx 触发周期寄存器
0x30	TMRx 基地址+0x30	TMRx_CPR	0x0000FFFF	TMRx 计数周期寄存器
0x38	TMRx 基地址+0x38	TMRx_PSCR	0x00000000	TMRx 预分频寄存器
0x3C	TMRx 基地址+0x3C	TMRx_CNTR	0x00000000	TMRx 计数寄存器
0x40	TMRx 基地址+0x40	TMRx_CRR	0x00000000	TMRx 重复计数寄存器
0x50	TMRx 基地址+0x50	TMRx_CC0R	0x00000000	TMRx 捕获/比较寄存器 0
0x54	TMRx 基地址+0x54	TMRx_CC1R	0x00000000	TMRx 捕获/比较寄存器 1
0x58	TMRx 基地址+0x58	TMRx_CC2R	0x00000000	TMRx 捕获/比较寄存器 2
0x5C	TMRx 基地址+0x5C	TMRx_CC3R	0x00000000	TMRx 捕获/比较寄存器 3
0x60	TMRx 基地址+0x60	TMRx_CIR	0x00000000	TMRx 捕获输入寄存器
0x64	TMRx 基地址+0x64	TMRx_BPR	0x00000000	TMRx Break 输入极性寄存器
0x68	TMRx 基地址+0x68	TMRx_BER	0x00000000	TMRx Break 输入使能寄存器

31.5.2 寄存器描述

31.5.2.1 TMRx 控制寄存器 0（TMRx_CR0）

- 名称：TMRx Control Register0
- 偏移地址：0x00
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TI0S	DCD		ARE	CMS		DIR	OPM	URS	UDIS	CEN
					R/W	R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W
					0x0	0x0		0x0	0x0		0x0	0x0	0x0	0x0	0x0

字段	说明
[10] TI0S	**TI0 输入选择(TI0 Input Selection)** 0：CH0 引脚连接到 TI0 输入 1：CH0~CH2 引脚异或后连接到 TI0 输入
[9:8] DCD	**时钟分频(Clock Division)** 此位指示定时器时钟频率与死区发生器及滤波发生器的时钟频率之间分频比； 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频
[7] ARE	**自动重载使能(AutoReload Enable)** AutoReload 功能使能，是否开启影子寄存器加载功能； 0: 关闭重载功能 1: 开启重载功能 Note： 1. 预加载功能涉及的寄存器有：计数终止值寄存器、预分频寄存器； 2. 如果不开启预加载功能，则相关寄存器写入新值将立刻生效； 3. 如果开启预加载功能，则相关寄存器写入新值后，在下一个更新事件（UEV）到来后生效。
[6:5] CMS	**中心对齐模式(Center-Aligned Mode Selection)** 00：边沿对齐模式，计数器根据方向位 (DIR) 递增计数或递减计数。 01：中心对齐模式 1，计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道的输出比较中断标志才置 1。 10：中心对齐模式 2，计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道的输出比较中断标志才置 1。 11：中心对齐模式 3，计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道的输出比较中断标志都会置 1。
[4] DIR	**计数方向(Counter Direction)** 0: 计数器递增计数； 1: 计数器递减计数；
[3] OPM	**单脉冲模式(One Pulse Mode)** 0: 计数器在发生更新事件时不会停止计数； 1: 计数器在发生下一更新事件时停止计数(将 CEN 位清零)； Note：配置为单次模式后，计数器溢出后，将自动停止计数。
[2] URS	**更新事件源(Update Request Source)** 此位由软件置 1 和清零，用以选择更新事件源。 0: 当配置为 0 时，所有以下事件都会生成更新事件: — 计数器上溢或下溢 — 将 UG 位置 1

	— 通过从模式生成的更新事件 1: 当配置为 1 时，只有计数器上溢或下溢时生成更新事件。
[1] UDIS	**更新禁止位(Update Disable)** 此位由软件置 1 和清零，用以使能/禁止更新事件的生成； 0:更新使能，更新(UEV)事件可通过以下事件之一生成: — 计数器上溢或下溢 — 将 UG 位置 1 — 通过从模式生成的更新事件 1:更新禁止 Note: 更新禁止后，不会生成更新事件(不会产生更新中断)，各影子寄存器的值保持不变，如果 UG 位置 1，则会重新初始化计数器和预分频器。
[0] CEN	**计数器使能位(Counter Enable):** 1:开始计数器 0:关闭计数器 Note: 单次模式下，当发生更新事件时会自动关闭 CEN；

31.5.2.2 TMRx 控制寄存器 1（TMRx_CR1）

- **名称：TMRx Control Register1**
- **偏移地址：0x04**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	OIS0N	OIS0						MMS		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						R/W		
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0						0x0		

字段	说明
[15] OIS3N	**CH3N 空闲状态输出电平(CH3N Output Idle-State):** 0: 当 MOE=0 时，OC0N=0 1: 当 MOE=0 时，OC0N=1
[14] OIS3	**CH3 空闲状态输出电平(CH3 Output Idle-State):** 0: 当 MOE=0 时，OC0=0 1: 当 MOE=0 时，OC0=1
[13] OIS2N	**CH2N 空闲状态输出电平(CH2N Output Idle-State):** 0: 当 MOE=0 时，OC0N=0 1: 当 MOE=0 时，OC0N=1
[12] OIS2	**CH2 空闲状态输出电平(CH2 Output Idle-State):** 0: 当 MOE=0 时，OC0=0 1: 当 MOE=0 时，OC0=1
[11] OIS1N	**CH1N 空闲状态输出电平(CH1N Output Idle-State):** 0: 当 MOE=0 时，OC0N=0 1: 当 MOE=0 时，OC0N=1
[10] OIS1	**CH1 空闲状态输出电平(CH1 Output Idle-State):** 0: 当 MOE=0 时，OC0=0 1: 当 MOE=0 时，OC0=1
[9] OIS0N	**CH0N 空闲状态输出电平(CH0N Output Idle-State):** 0: 当 MOE=0 时，OC0N=0 1: 当 MOE=0 时，OC0N=1
[8] OIS0	**CH0 空闲状态输出电平(CH0 Output Idle-State):** 0: 当 MOE=0 时，OC0=0 1: 当 MOE=0 时，OC0=1

[2:0] MMS	**主模式选择(Master Mode Selection):** 这些位用于选择主模式下将要发送到从定时器的同步的信息(TRGO)，组合描述如下： 000: 复位 - 寄存器(UGR)中的 UG 位用作触发输出，如果从模式控制器配置为复位模式，则 TRGO 上的信号用于从定时器的复位操作； 001: 使能 - 计数器使能信号用作触发输出(TRGO)，该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器； 010: 更新 - 选择更新事件作为触发输出(TRGO)。例如：主定时器可用作从定时器的预分频器； 011: 捕获比较事件 - 一旦发生输入捕获或比较匹配事件，当 CC0IF 标志被置 1 时，触发输出都会发送一个正脉冲； 100: 比较输出 - OC0 信号用作触发输出(TRGO)； 101: 比较输出 - OC1 信号用作触发输出(TRGO)； 110: 比较输出 - OC2 信号用作触发输出(TRGO)； 111: 比较输出 - OC3 信号用作触发输出(TRGO)；

31.5.2.3 TMRx Slave 控制寄存器（TMRx_SCR）

- **名称：TMRx Slave Control Register**
- **偏移地址：0x08**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				ETS					EE	EMS		EFS			
				R/W					R/W	R/W		R/W			
				0x0					0x0	0x0		0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TS					FE				SMS			
			R/W					R/W				R/W			
			0x0					0x0				0x0			

字段	说明
[27:24] ETS	**ETR 输入源选择(ETR Input Shource Selection)** 0x0: ETR - PAD 0x1~0x7: CMP0-6 0x8-0x9: PDM0-1 0xa-0xc: ADC0WDGx 0xd-0xf: ADC0WDGx
[22] EE	**ETR 控制模式选择(ETR Mode Selection)** 0: 外部时钟模式 1 1: 外部时钟模式 2 Note: 当选择外部时钟模式 2，将在上下双边沿进行计数，并且禁止从模式选择 ETR 作为源；
[21:20] EMS	**ETR 边沿模式选择(ETR Edge-Mode Selection)** 00: 关闭 01: 上升沿 10: 下降沿 11: 上下沿
[19:16] EFS	**ETR 滤波选择(ETR Filter Selection)** 0: 无滤波 1: Ttds x1 2: Ttds x2 ……
[12:8] TS	**触发选择(Trigger Selection)** 此位域可选择用于同步计数器的触发输入。 00000: 内部触发 0(ITR0) ..... 01111: 内部触发 15(ITR15) 10000: 滤波后的定时器输入 0(CH0-FE-双边沿) 10001: 滤波后的定时器输入 1(CH0-F)

	10010: 滤波后的定时器输入 2(CH1-F) 10011: 滤波后的定时器输入 3(CH2-F) 10100: 滤波后的定时器输入 4(CH3-F) 10101: 边沿后的定时器输入 5(ETR-FE) 10110: 滤波后的定时器输入 6(ETR-F)
[7] FE	**主从快速同步使能(Master/Slave Fast-Sync Enable)** 0：不执行任何操作 1：避免当前定时器的触发输入事件 (TRGI) 的延迟，以使当前定时器与其主定时器实现同步。
[3:0] SMS	**从模式选择(Slave Mode Selection):** 选择外部触发信号(TRGI)的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器说明）； 0000: 禁止从模式 0001: 保留 0010: 保留 0011: 保留 0100: 复位模式 - 在出现所选触发输入(TRGI)上升沿时，重新初始化计数器并生成一个寄存器更新事件； 0101: 门控模式 - 触发输入(TRGI)为高电平时使能计数器时钟，只要触发输入变为低电平，计数器立即停止计数（但不复位），计数器的启动和停止都被控制； 0110: 触发模式 - 触发信号 TRGI 出现上升沿时启动计数器（但不复位），只控制计数器的启动，可使用 FE 来快速同步使能； 0111: 外部时钟模式 - 由所选触发信号(TRGI)的上升沿提供计数器时钟； 1000: 组合复位+触发模式 - 在出现所选触发输入(TRGI)上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器； 1001: 组合门控+复位模式 - 在触发输入(TRGI)为高电平时使能计数器时钟，只要触发输入变为低电平，计数器立即停止计数并复位，计数器的启动和停止都被控制；

31.5.2.4 TMRx 中断使能寄存器（TMRx_IER）

- **名称：TMRx Interrupt Enable Register**
- **偏移地址：0x0C**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				BIE	TIE	OVIE	UIE	C3OIE	C3MIE	C2OIE	C2MIE	C1OIE	C1MIE	C0OIE	C0MIE
				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[11] BIE	**断路中断使能(Break Interrupt Enable)** 0：关闭 1：使能 Note: 包括 System Fault 及 Break0、Break1 事件；
[10] TIE	**触发中断使能(Trigger Interrupt Enable)** 0：关闭 1：使能
[9] OVIE	**计数上溢中断使能(Overflow Interrupt Enable)** 1：使能中断 0：禁止中断
[8] UIE	**更新中断使能(Update Interrupt Enable)** 1：使能中断 0：禁止中断
[7]	**CH3 重复捕获中断使能(CH3 OverCapture Interrupt Enable):**

C3OIE	1：中断使能 0：中断关闭
[6] C3MIE	**CH3 比较中断使能(CH3 Capture/Compare Interrupt Enable):** 1：中断使能 0：中断关闭
[5] C2OIE	**CH2 重复捕获中断使能(CH2 OverCapture Interrupt Enable):** 1：中断使能 0：中断关闭
[4] C2MIE	**CH2 比较中断使能(CH2 Capture/Compare Interrupt Enable):** 1：中断使能 0：中断关闭
[3] C1OIE	**CH1 重复捕获中断使能(CH1 OverCapture Interrupt Enable):** 1：中断使能 0：中断关闭
[2] C1MIE	**CH1 比较中断使能(CH1 Capture/Compare Interrupt Enable):** 1：中断使能 0：中断关闭
[1] C0OIE	**CH0 重复捕获中断使能(CH0 OverCapture Interrupt Enable):** 1：中断使能 0：中断关闭
[0] C0MIE	**CH0 比较中断使能(CH0 Capture/Compare Interrupt Enable):** 1：中断使能 0：中断关闭

31.5.2.5 TMRx 状态寄存器（TMRx_SR）

- **名称：TMRx Status Register**
- **偏移地址：0x10**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		B1IF	SBIF	B0IF	TIF	OVIF	UIF	CC3OIF	CC3MIF	CC2OIF	CC2MIF	CC1OIF	CC1MIF	CC0OIF	CC0MIF
		R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[13] B1IF	**断路中断标志位(Break1 Interrupt Flag)** 只要断路输入变为有效状态，此标志便由硬件置 1，断路输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路输入上检测到有效电平
[12] SBIF	**系统断路中断标志位(System Fault Break Interrupt Flag)** 只要断路输入变为有效状态，此标志便由硬件置 1，断路输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路输入上检测到有效电平
[11] B0IF	**断路中断标志位(Break0 Interrupt Flag)** 只要断路输入变为有效状态，此标志便由硬件置 1，断路输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路输入上检测到有效电平
[10] TIF	**触发中断标志(Trigger Interrupt Flag)** 该标志在发生触发事件时由硬件置 1，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将置 1，需要通过软件清零。

	0: 未发生触发事件 1: 触发中断挂起
[9] OVIF	**计数器上溢中断标志位(Counter Overflow Interrupt Flag)** 0: 未发生上溢 1: 计数器上溢
[8] UIF	**更新中断标志(Update Interrupt Flag)** 该位在发生更新事件时通过硬件置 1,但需要通过软件清零。 0: 未发生更新 1: 更新中断挂起 该位在以下情况下更新寄存器时由硬件置 1: — 计数器发生上溢且当 CR 寄存器中 UDIS="0"时； — 当 CR 寄存器中 URS="0"且 UDIS="0"，通过软件使用 UGR 寄存器中的 UG 位重新初始化 Counter 时；
[7] CC3OIF	**CC3 重复捕获成功中断标志位(CC3 OverCapture Interrupt Flag)** 参考 CC0 的描述
[6] CC3MIF	**CC3 比较成功中断标志位(CC3 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[5] CC2OIF	**CC2 重复捕获成功中断标志位(CC2 OverCapture Interrupt Flag)** 参考 CC0 的描述
[4] CC2MIF	**CC2 比较成功中断标志位(CC2 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[3] CC1OIF	**CC1 重复捕获成功中断标志位(CC1 OverCapture Interrupt Flag)** 参考 CC0 的描述
[2] CC1MIF	**CC1 比较成功中断标志位(CC1 Capture/Compare Interrupt Flag)** 参考 CC0 的描述
[1] CC0OIF	**CC0 重复捕获成功中断标志位(CC0 OverCapture Interrupt Flag)** 0: 未检测到重复捕获 1: 当 CPIF=1，同时 Caputre 再次捕获到匹配的边沿，将置位该位(重复捕获成功有效) Note: 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1，通过软件写 1 可将该位清零
[0] CC0MIF	**CC0 比较成功中断标志位(CC0 Capture/Compare Interrupt Flag)** 配置为输入: 此位将在发生捕获事件时由硬件置 1，通过软件写 1 清 0 或读取 Capture 值寄存器时自动清 0。 0: 未发生输入捕获事件。 1: Compare 预加载寄存器中已捕获到计数器值（检测到与所选极性匹配的边沿） 配置为输出: 当计数器与比较值匹配时，此标志由硬件置 1，但需要通过软件写 1 清零。 0: 不匹配 1: 计数器的值与 Compare 寄存器的值匹配，当 Compare 值大于 Counter 周期值时，将在计数器发生上溢时变为高电平；当 UGR 寄存器 UG 位软件置 1，也会触发 Compare 中断；

31.5.2.6 TMRx 更新事件生成寄存器（TMRx_UGR）

- 名称：TMRx Update Generation Register
- 偏移地址：0x14
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CC3UG	CC2UG	CC1UG	CC0UG	B1G	B0G	TG	UG
								R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
[7] CC3UG	**CC3 捕获/比较更新事件生成(CC3 Capture/Compare Update Generation)** 参考 CC0 的描述
[6] CC2UG	**CC2 捕获/比较更新事件生成(CC2 Capture/Compare Update Generation)** 参考 CC0 的描述
[5] CC1UG	**CC1 捕获/比较更新事件生成(CC1 Capture/Compare Update Generation)** 参考 CC0 的描述
[4] CC0UG	**CC0 捕获/比较更新事件生成(CC0 Capture/Compare Update Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零 0: 不执行任何操作 1: 重生 Capture/Compare 更新事件 配置为输出: 使能后，Compare 中断标志置 1 并发送相应的中断，生成更新事件，更新 Compare 值； 配置为输入: 将捕获到计数器当前值写入 CCxR 寄存器中，使能后，Capture 中断标志置 1 并发送相应中断，如果 Capture 中断标志已为高电平，重复标志将置 1
[3] B1G	**断路 1 事件生成(Break1 Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: SR 寄存器中的 B1IF 标志置 1，生成 Break1 事件。
[2] B0G	**断路 0 事件生成(Break0 Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: SR 寄存器中的 B0IF 标志置 1，生成 Break0 事件。
[1] TG	**触发事件生成(Trigger Generation)** 此位由软件置 1 以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: SR 寄存器中的 TIF 标志置 1，生成 Trigger 事件。
[0] UG	**更新事件生成(Update Generation)** 此位由软件置 1 生成事件，并由硬件自动清零; 0: 不执行任何操作 1: 重新初始化计数器并生成寄存器更新事件，预分频器计数器也将清零(但预分频比不受影响)，计数器清零

31.5.2.7 TMRx 捕获/比较模式寄存器（TMRx_CCMR）

- 名称：TMRx Capture Compare Mode Register
- 偏移地址：0x20
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OC3M				CC3PE		CC3S		OC2M				CC2PE		CC2S	
R/W				R/W		R/W		R/W				R/W		R/W	
0x0				0x0		0x0		0x0				0x0		0x0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC1M				CC1PE		CC1S		OC0M				CC0PE		CC0S	
R/W				R/W		R/W		R/W				R/W		R/W	
0x0				0x0		0x0		0x0				0x0		0x0	

字段	说明
[31:28] OC3M	**CC3 输入滤波/输出比较模式选择(CC3 Input Filter/Output Compare Mode)** 参考 CC0 描述；
[27:26] CC3PE	**CC3 输入捕获分频/比较自动加载使能(CC3 Compare Auto-Realod Enable)** 参考 CC0 描述；
[25:24] CC3S	**CC3 捕获/比较方向选择(CC3 Capture Compare Selection)** 参考 CC0 描述；
[23:20] OC2M	**CC2 输入滤波/输出比较模式选择(CC2 Input Filter/Output Compare Mode)** 参考 CC0 描述；
[19:18] CC2PE	**CC2 输入捕获分频/比较自动加载使能(CC2 Compare Auto-Realod Enable)** 参考 CC0 描述；
[17:16] CC2S	**CC2 捕获/比较方向选择(CC2 Capture Compare Selection)** 参考 CC0 描述；
[15:12] OC1M	**CC1 输入滤波/输出比较模式选择(CC1 Input Filter/Output Compare Mode)** 参考 CC0 描述；
[11:10] CC1PE	**CC1 输入捕获分频/比较自动加载使能(CC1 Compare Auto-Realod Enable)** 参考 CC0 描述；
[9:8] CC1S	**CC1 捕获/比较方向选择(CC1 Capture Compare Selection)** 参考 CC0 描述；
[7:4] OC0M	**CC0 输入滤波/输出比较模式选择(CC0 Input Filter/Output Compare Mode)** 配置为输出，OCxRef 的输出模式（有效电平为高电平） 0000: 冻结, 输出比较寄存器与计数器进行比较不会对输出造成任何影响(保持原有状态) 0001: 当计数器与捕获/比较寄存器匹配时，输出有效电平(高电平) 0010: 当计数器与捕获/比较寄存器匹配时，输出无效电平(低电平) 0011: 当计数器与捕获/比较寄存器匹配时，发生翻转 0100: 强制变为无效电平(低电平) 0101: 强制变为有效电平(高电平) 0110：PWM 模式 1 - 在递增计数模式下，只要 TMRx_CNT<TMRx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TMRx_CNT>TMRx_CCR1，通道 1 便为无效状态(OC1REF=0)，否则为有效状态(OC1REF=1)。 0111：PWM 模式 2 - 在递增计数模式下，只要 TMRx_CNT<TMRx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TMRx_CNT>TMRx_CCR1，通道 1 便为有效状态，否则为无效状态。 1000：可再触发 OPM 模式 1 - 在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。 1001：可再触发 OPM 模式 2 - 在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。 1100：组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF

	的逻辑或运算结果。 1101：组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。 1110：不对称 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。 1111：不对称 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。 配置为输入 0: 无滤波 1: Ttds x1 2: Ttds x2 ……
[3:2] CC0PE	**CC0 输入捕获分频/比较自动加载使能(CC0 Compare Auto-Realod Enable)** 配置为输出 0: 禁止自动预装载，可随时向预装载寄存器写入数据，写入后将立即使用新值 1: 使能自动预装载，可读/写访问预装载寄存器，而预装载值在每次生成更新事件时都会装载到活动寄存器中 配置为输入 此位域定义 CC0 输入 (IC0) 的预分频比。 00：无预分频器，捕获输入上每检测到一个边沿便执行捕获 01：每发生 2 个事件便执行一次捕获 10：每发生 4 个事件便执行一次捕获 11：每发生 8 个事件便执行一次捕获
[1:0] CC0S	**CC0 捕获/比较方向选择(CC0 Capture Compare Selection)** 此位定义对应通道的方向（输入/输出）。 00: CC0 通道配置为输出。 01: CC0 通道配置为输入，输入映射到 CH0 上 10: CC0 通道配置为输入，输入映射到 CH1 上 11: CC0 通道配置为输入，输入映射到 ITR 上，此模式仅在通过 TS 位选择内部触发输入时有效；

31.5.2.8 TMRx 捕获/比较使能寄存器（TMRx_CCER）

- 名称：TMRx Capture Compare Enable Register
- 偏移地址：0x24
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC3P		CC3NE	CC3E	CC2P		CC2NE	CC2E	CC1P		CC1NE	CC1E	CC0P		CC0NE	CC0E
R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W
0x0		0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0

字段	说明
[15:14] CC3P	**CC3 捕获/比较极性选择(CC3 Compare/Capture Polarity)** 参考 CC0 的描述
[13] CC3NE	**CC3N 捕获/比较互补通道使能(CC3N Capture/Compare Enable)** 参考 CC0 的描述
[12] CC3E	**CC3 捕获/比较通道使能(CC3 Capture/Compare Enable)** 考 CC0 的描述
[11:10] CC2P	**CC2 捕获/比较极性选择(CC2 Compare/Capture Polarity)** 参考 CC0 的描述
[9] CC2NE	**CC2N 捕获/比较互补通道使能(CC2N Capture/Compare Enable)** 参考 CC0 的描述
[8] CC2E	**CC2 捕获/比较通道使能(CC2 Capture/Compare Enable)** 考 CC0 的描述
[7:6] CC1P	**CC1 捕获/比较极性选择(CC1 Compare/Capture Polarity)** 参考 CC0 的描述
[5] CC1NE	**CC1N 捕获/比较互补通道使能(CC1N Capture/Compare Enable)** 参考 CC0 的描述
[4] CC1E	**CC1 捕获/比较通道使能(CC1 Capture/Compare Enable)** 考 CC0 的描述
[3:2] CC0P	**CC0 捕获/比较极性选择(CC0 Compare/Capture Polarity)** 配置为输出(bit3 和 bit2 分别对应 CC0N 和 CC0 的输出极性选择) 0: 高电平有效 1: 低电平有效 配置为输入: 00: 关闭 01: 上升沿 10: 下降沿 11: 上升沿和下降沿
[1] CC0NE	**CC0N 捕获/比较互补通道使能(CC0N Capture/Compare Enable)** 配置为输出: 0: 关闭 1: 开启
[0] CC0E	**CC0 捕获/比较通道使能(CC0 Capture/Compare Enable)** 配置为输出: 0: 关闭 1: 开启 配置为输入: 该位决定了输入捕获/比较寄存器 1 实际捕获到计数器的值 0: 禁止捕获 1: 使能捕获

31.5.2.9 TMRx 死区控制寄存器（TMRx_DCR）

- 名称：TMRx Dead-Time Contorl Register
- 偏移地址：0x28
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BK1F								BK1P	BK1E	BK0F					
R/W								R/W	R/W	R/W					
0x0								0x0	0x0	0x0					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BK0F		BK0P	BK0E	MOE	AOE	OSSR	OSSI	DTG							
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W							
0x0		0x0	0x0	0x0	0x0	0x0	0x0	0x0							

字段	说明
[31:24] BK1F	**断路滤波器(Break1 Filter)** 0: 无滤波 1: Ttds x 1 2: Ttds x 2 ……
[23] BK1P	**断路输入极性(Break1 Polarity)** 0：断路输入为高电平有效 1：断路输入为低电平有效
[22] BK1E	**断路控制使能(Break1 Enable)** 0：禁止断路输入 1：使能断路输入
[21:14] BK0F	**断路滤波器(Break0 Filter)** 0: 无滤波 1: Ttds x 1 2: Ttds x 2 ……
[13] BK0P	**断路输入极性(Break0 Polarity)** 0：断路输入为高电平有效 1：断路输入为低电平有效
[12] BK0E	**断路控制使能(Break0 Enable)** 0：禁止断路输入 1：使能断路输入
[11] MOE	**主路输出使能(Main Output Enable)** 只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1，此位仅对配置为输出的通道有效。 0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。 1：OC 和 OCN 输出的相应使能位(CCxE 和 CCxNE 位)均置 1，则使能 OC 和 OCN 输出。
[10] AOE	**自动输出使能(Automatic Output Enable)** 0：MOE 只能由软件置 1 1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果断路输入无效）
[9] OSSR	**运行模式下关闭状态选择(Off-State Selection for Run-Mode)** 此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。 0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。
[8] OSSI	**空闲模式下关闭状态选择(Off-State Selection for Idle-Mode)** 此位在 MOE=0 时作用于配置为输出的通道。 0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0） 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平，然后设置

	OC/OCN 使能输出信号=1
[7:0] DTG	**配置死区时间选择(Dead-Time Generator Setup)** 此位域定义插入到互补输出之间的死区持续时间。 DTG[7:5]=0xx => DT=DTG[7:0] x Tdtg，　　其中 Tdtg=Ttds。 DTG[7:5]=10x => DT=(64+DTG[5:0]) x Tdtg，其中 Tdtg=2xTtds。 DTG[7:5]=110 => DT=(32+DTG[4:0]) x Tdtg，其中 Tdtg=8xTtds。 DTG[7:5]=111 => DT=(32+DTG[4:0]) x Tdtg，其中 Tdtg=16xTtds。 示例：如果 Ttds=125ns(8MHz)，则可能的死区值为： 0 到 15875ns（步长为 125ns） 16us 到 31750ns（步长为 250ns） 32us 到 63us　（步长为 1us） 64us 到 126us　（步长为 2us）

31.5.2.10 TMRx 触发周期寄存器（TMRx_TTCR）

- 名称：**TMRx Trigger Cycles Register**
- 偏移地址：**0x2C**
- 默认值：**0x00000077**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				TC3E	TC2E	TC1E	TC0E	TCW				TOW			
				R/W	R/W	R/W	R/W	R/W				R/W			
				0x0	0x0	0x0	0x0	0x7				0x7			

字段	说明
[11] TC3E	**TRGO CH3 输出使能(Trigger CH3 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[10] TC2E	**TRGO CH2 输出使能(Trigger CH2 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[9] TC1E	**TRGO CH1 输出使能(Trigger CH1 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[8] TC0E	**TRGO CH0 输出使能(Trigger CH0 Enable)** 0: 关闭 1: 使能 Note：TRGO 输出由各个 CHx 或操作后输出；
[7:4] TCW	**TRGCC 脉冲宽度控制(Trigger Compare/Caputre Width)** 0: 1 个 Cycle 1: 2 个 Cycle …… Note： 1. TRGCC 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle： 2. TRGCC 脉冲宽度必须小于 Timer 的 Period 值(CPV).
[3:0] TOW	**TRGO 脉冲宽度控制(Trigger Output Width)** 0: 1 个 Cycle 1: 2 个 Cycle ……

Note：
1. TRGO 脉冲默认宽度为一个时钟 Cycle，配置输出宽度等于(TOW+1)时钟 Cycle：
2. TRGO 脉冲宽度必须小于 Timer 的 Period 值(CPV).

31.5.2.11 TMRx 计数周期寄存器（TMRx_CPR）

- **名称：TMRx Counter Period Register**
- **偏移地址：0x30**
- **默认值：0x0000FFFF**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPV															
R/W															
0xFFFF															

字段	说明
[15:0] CPV	**计数周期值(Counter Period Value)** 通过配置此寄存器设置定时器结束计数的值。此外，该寄存器支持自动预装载（Auto-Reload）功能： 1. 不使能自动预装载功能（ARPE=0） 对该寄存器的设置将立即作用到内部影子寄存器内，本次计数周期内即生效； 2. 使能自动预装载功能（ARPE=1） 对该寄存器的设置将在更新事件（UEV）产生后，再更新至内部影子寄存器内生效。

31.5.2.12 TMRx 预分频寄存器（TMRx_PSCR）

- 名称：TMRx Prescaler Register
- 偏移地址：0x38
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC															
R/W															
0x0															

字段	说明
[15:0] PSC	**预分频寄存器（Prescaler Value)** 该寄存器支持自动预装载(Auto-Reload)功能，每次发生更新事件时会装载到实际预分频器寄存器的值； Note：计数器时钟频率 CLK_CNT 等于 FCLK/(PSC+1)，其中 PSC 范围为 0~65535： 0:不分频，系统时钟 1:2 分频 2:3 分频 ……

31.5.2.13 TMRx 计数寄存器（TMRx_CNTR）

- 名称：TMRx Counter Register
- 偏移地址：0x3C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															
0x0															

字段	说明
[15:0] CNT	**定时器计数值(Counter Value)** 通过该寄存器可对定时器计数器的值进行读取/写入。 需注意，因为系统与外设之间存在一定的总线延迟，对该寄存器的读取/写入操作可能存在一定的偏差。

31.5.2.14 TMRx 重复计数寄存器（TMRx_CRR）

- 名称：TMRx Counter Repeat Register
- 偏移地址：0x40
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								REP							
								R/W							
								0x0							

字段	说明
[7:0] REP	**重复计数值(Counter Repeat Value)** 使能预装载寄存器之前，用户可通过这些位设置比较寄存器的更新频率（从预装载寄存器向活动寄存器周期性传输数据），使能更新中断时，也可设置更新中断的生成速率。 与 REP 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件时才会重载 REP 值，因此在生成下一重复更新事件之前，无论向该寄存器写入何值都无影响。 这意味着在 PWM 模式(REP+1)下对应于边沿对齐模式的 PWM 周期数。

31.5.2.15 TMRx 捕获/比较寄存器 0（TMRx_CC0R）

- 名称：TMRx Capture Compare Register0
- 偏移地址：0x50
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC0V															
R/W															
0x0															

字段	说明
[15:0] CC0V	**CC0 捕获/比较值(CC0 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

31.5.2.16 TMRx 捕获/比较寄存器 1（TMRx_CC1R）

- 名称：TMRx Capture Compare Register1
- 偏移地址：0x54
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1V															
R/W															
0x0															

字段	说明
[15:0] CC1V	**CC1 捕获/比较值(CC1 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

31.5.2.17 TMRx 捕获/比较寄存器 2（TMRx_CC2R）

- 名称：TMRx Capture Compare Register2
- 偏移地址：0x58
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC2V															
R/W															
0x0															

字段	说明
[15:0] CC2V	**CC2 捕获/比较值(CC2 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次 2:3 次 …… 配置为输入: 为上一个输入捕获事件发生时的计数器值

31.5.2.18 TMRx 捕获/比较寄存器 3（TMRx_CC3R）

- 名称：TMRx Capture Compare Register3
- 偏移地址：0x5C
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC3V															
R/W															
0x0															

字段	说明
[15:0] CC3V	**CC3 捕获/比较值(CC3 Capture/Compare Value)** 配置为输出: 装载到实际比较寄存器的值（预装载值）。 如果没有通过 CCMR 寄存器中的 CCxPE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中，否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器中），实际比较寄存器中包含要与计数器进行比较并输出上发出信号的值； 计数实例:(1~65536) 0:1 次 1:2 次

2:3 次
……
配置为输入:
为上一个输入捕获事件发生时的计数器值

31.5.2.19 TMRx 捕获输入寄存器（TMRx_CIR）

- **名称：TMRx Capture Input Register**
- **偏移地址：0x60**
- **默认值：0x00000000**
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				C3TIS								C2TIS			
				R/W								R/W			
				0x0								0x0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				C1TIS								C0TIS			
				R/W								R/W			
				0x0								0x0			

字段	说明
[27:24] C3TIS	**CH3 触发输入选择(CH3 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[19:16] C2TIS	**CH2 触发输入选择(CH2 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[11:8] C1TIS	**CH1 触发输入选择(CH1 Capture Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……
[3:0] C0TIS	**CH0 触发输入选择(CH0 Trigger Input Selection)** 0000: TMRx_IN0 0001: TMRx_IN1 ……

31.5.2.20 TMRx Break 输入极性寄存器（TMRx_BPR）

- 名称：TMRx Break Polarity Register
- 偏移地址：0x64
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B1POL															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B0POL															
R/W															
0x0															

字段	说明
[31:16] B1POL	**Break1 输入极性选择(Break1 Input Polarity Selection)** 0：高电平有效 1：低电平有效
[15:0] B0POL	**Break0 输入极性选择(Break0 Input Polarity Selection)** 0：高电平有效 1：低电平有效

31.5.2.21 TMRx Break 输入使能寄存器（TMRx_BER）

- 名称：TMRx Break Enable Register
- 偏移地址：0x68
- 默认值：0x00000000
- 寄存器列表

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B1IEN															
R/W															
0x0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B0IEN															
R/W															
0x0															

字段	说明
[31:16] B1IEN	**Break1 输入使能控制(Break1 Input Enable)** 0：关闭 1：使能
[15:0] B0IEN	**Break0 输入使能控制(Break0 Input Enable)** 0：关闭 1：使能

32 调试接口

32.1 主要特性

芯片使用 Cortex™-M4 的内核，该内核内含硬件调试模块，支持复杂的调试操作。硬件调试模块允许内核在取指（指令断点）或访问数据（数据断点）时停止，当内核停止时，内核的内部状态和系统的外部状态都是可以查询的。完成查询后，内核和外设可以被复原，程序将继续执行。

芯片支持 SWD 调试接口，当芯片连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

ARM Cortex™-M4 内核提供集成的片上调试功能。它由以下部分组成：

- SWD-DP：串行调试端口
- ITM：执行跟踪单元
- FPB：闪存指令断点
- DWT：数据触发

有关调试接口的更多信息请参考 ARM 官方手册“Cortex™-M4 Technical Reference Manual”，（汉译版本为“Cortex™-M4 技术参考手册”），以及 ARM 开发工具集技术参考手册。

32.2 SWD 调试接口

芯片内核集成了串行调试接口（SWD），串行调试接口（SWD）为 AHP-AP 模块提供 2 个引脚接口（时钟＋数据）：

- SWCLK：从主机到目标的时钟
- SWDIO：双向数据接口

对于 SWDIO，必须在电路板上对线路进行上拉（ARM 建议采用 100 K）。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

芯片的 2 个普通 I/O 口用作为 SWD 接口的引脚，这些引脚在所有的封装里都存在。

表 32-1 SWD 调试端口引脚

SWD 端口引脚名称	SWD 端口类型	SWD 端口描述	芯片内部状态	引脚分配
SWCLK	输入	时钟	集成下拉	PA14
SWDIO	输入/输出	数据	集成上拉	PA13
SWO	输出		-	

33 设备电子签名

电子签名存储在 FLASH 内部，可以使用 SWD 或 CPU 对其进行读取。它包含出厂前编程的标识数据。

33.1 唯一设备标识（128 位）

唯一设备标识的用途：

- 用于产品序列号（例如 USB 字符串序列号或其它终端应用程序）
- 激活安全自举过程等

128 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。

128 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

33.2 唯一设备标识寄存器

33.2.1 寄存器描述

基地址: SYSCTRL（0x40021000）

33.2.1.1 UID Control register0

- **名称**：UID Control register0
- **复位值**：0xXXXX XXXX（只读，其中 X 是出厂前编程的）
- **偏移地址**：0x50

位	31:0
名称	UID0
访问	R/W
复位值	0x0

字段	说明
[31:0] UID0	设备标识 0

33.2.1.2 UID Control register1

- **名称**：UID Control register1
- **复位值**：0xXXXX XXXX（只读，其中 X 是出厂前编程的）
- **偏移地址**：0x54

位	31:0
名称	UID1
访问	R/W
复位值	0x0

字段	说明
[31:0] UID1	设备标识 1

33.2.1.3 UID Control register2

- **名称**：UID Control register2
- **复位值**：0xXXXX XXXX（只读，其中 X 是出厂前编程的）
- **偏移地址**：0x58

位	31:0
名称	UID2
访问	R/W
复位值	0x0

字段	说明
[31:0] UID2	设备标识 2

33.2.1.4 UID Control register3

- **名称**：UID Control register3
- **复位值**：0xXXXX XXXX（只读，其中 X 是出厂前编程的）
- **偏移地址**：0x5C

位	31:0
名称	UID3
访问	R/W
复位值	0x0

字段	说明
[31:0] UID3	设备标识 3

34 电气特性

34.1 参数条件

除非特别说明，所有电压都是以 VSS 作为参考。

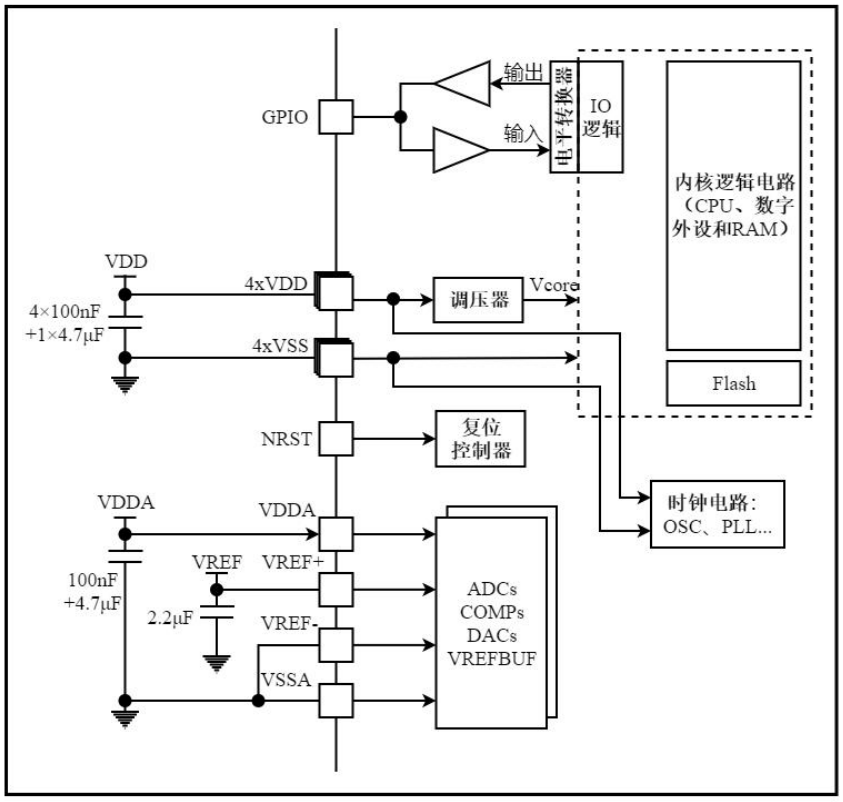
34.2 最大值和最小值

除非特别说明，最大值和最小值是在最差环境温度、供电电压以及频率下的测试结果。

34.3 典型值

除非特别说明，典型值都是在 TA=25℃，V_{DD}=V_{DDA}=3.3V 的条件下获取的，仅作为设计参考。

34.4 电源供电



注意：建议按上图推荐给每个电源添加退耦电容。建议给每个电源引脚添加一个 100nF 小电容滤除高频噪声。

34.5 绝对最大额定值

超过下表所示绝对最大额定值的电压值、电流或温度可能导致芯片永久性的损坏。

34.5.1 电压特性

Symbol	Description	Min	Max	Unit
VDD -VSS	片外供电电压，包括 VDD、VDDA。	-0.3	4.0	V
VDD - VDDA	允许的 VDD 和 VDDA 电压差值	-0.4	0.4	V
VIN	TT_x 的输入电压	VSS-0.3	4.0	V
	FT_x 的输入电压	VSS-0.3	5.5	
ΔVDDx	不同 VDD 引脚的电压差值	-	50	mV
ΔVSSx	不同 VSS 引脚的电压差值（包括 VSSA）	-	50	mV

34.5.2 电流特性

Symbol	Description	Max.	Unit
ΣI_{VDD}	所有 VDD 引脚输入的总电流	150	mA
ΣI_{VDDA}	所有 VDDA 引脚输入的总电流	100	mA
I_{VDD}	单个 VDD 引脚输入的总电流	100	mA
I_{VDDA}	单个 VDDA 引脚输入的总电流	100	mA
I_{IO}	单个 GPIO 输入/输出的电流	20	mA
ΣI_{IO}	所有 GPIO 输入/输出的总电流	100	mA

34.5.3 温度特性

Symbol	Description	Max.	Unit
T_{STRG}	芯片储存温度范围	–65 to +150	℃
T_j	芯片结温	150	℃

34.6 典型工作条件

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
F_{hclk}	芯片内部 AHB 时钟频率		0		180	MHz
F_{pclk0}	芯片内部 APB0 时钟频率		0		90	MHz
F_{pclk1}	芯片内部 APB1 时钟频率		0		90	MHz
V_{DD}	数字 IO 工作电压		3.1	3.3	3.63	V
V_{DDA}	模拟供电电压	与 V_{DD} 电压相同	3.1	3.3	3.63	V
V_{IN}	IO 输入电压	TT_xx	-0.3		V_{DD}+0.3	V

		FT_xx	-0.3		MIN(MIN(V_{DD}, V_{DDA})+3.6, 5.5)	V
PD	整体功耗	温度 TA=85℃，工作频率 180MHz，无 IO 负载			TBD	mW
T_J	芯片结温		-40		150	℃

34.7 芯片上电/掉电特性

（VDD=VDDA=3.3V，TA=25℃，除非另有说明）

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
VDD_{POR}*	VDD 上电、掉电复位阈值电压	下降沿		1.4		V
		上升沿		1.7		V
$VDD_{PORhyst}$*	VDD POR 迟滞			300		mV
VDD_{OK}*	VDD 上电 OK 阈值	下降沿		2.25		V
		上升沿		2.4		V
VDD_{LVD}*	VDD 低压检测阈值	2bit 可配置为 2.4V~3V	2.4		3	V
T_{rst}*	上电复位时间					ms

注：*-由设计保证，不在量产测试中测试。

34.8 片内电源 V_{DDA} 监测特性

Symble	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{LVD0}	V_{DDA} 电压监测阈值 0	Rising edge		2.55		V
		Falling edge				
V_{LVD1}	V_{DDA} 电压监测阈值 1	Rising edge		2.7		V
		Falling edge				
V_{LVD2}	V_{DDA} 电压监测阈值 2	Rising edge		2.85		V
		Falling edge				
V_{LVD3}	V_{DDA} 电压监测阈值 3	Rising edge		3.0		V
		Falling edge				
V_{hyst_LVD}	V_{DDA} 电压监测迟滞		-	10	-	mV

34.9 片内参考电压

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{REF+}	参考电压	(V_{DDA} >3.1V)，VREFSEL =0		2.9		V
		(V_{DDA} <3.1V)，VREFSEL =1		2.5		V
Iload*	负载电流能力				10	mA
TC*	参考电压温度系数	-40℃~150℃，VREFSEL =1	27.5	36.62	42.13	ppm/℃
		-40℃~150℃，VREFSEL =0	26.9	40	49.1	ppm/℃
LNR*	线性调整率	V_{DDA} >3.1V，VREFSEL =0	0.437	0.607	1.18	mV/V

		V_{DDA} <3.1V，VREFSEL =1	0.374	0.522	0.969	mV/V
LDR*	负载调整率	V_{DDA} =3.3V，VREFSEL =0	0.246	0.334	0.583	V/A
		V_{DDA} =3V，VREFSEL =1	0.196	0.262	0.446	V/A
Output Noise*	参考电压输出噪声	I_Load = 2mA，VREFSEL =0，CL=2.2uF	39.8	45.2	55.6	uVrms
		I_Load = 2mA，VREFSEL =1，CL=2.2uF	39.8	45.1	55.6	uVrms
PSRR*	参考电压电源抑制比	1kHz，I_Load = 2mA，VREFSEL =0	-75.86	-75	-73.76	dB
		1kHz，I_Load = 2mA，VREFSEL =1	-71.4	-69.77	-67.81	dB
t_{Setup}*	内部参考建立时间	CL=2.2uF	56	72.36	98	uS
V_{Trim_step}*	校准步长		-11.11	-8.5	-6.571	mV/step
I_VSSA*		-40℃ < TA < 150℃，TRIM 后，VREFSEL=1	45	150	585	uA
		-40℃ < TA < 150℃，TRIM 后，VREFSEL=0	78	285	1150	uA
I_VSS*				15		uA

（V_{DDA}=3.3V，TA=25℃）

注：*-由设计保证，不在量产测试中测试。

34.10 时钟特性

34.10.1 片外晶振特性（HSE）

（VDD= 3.3V，TA=25℃）

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
F_{xosc}	外部输入时钟频率或晶振频率		4	8	26	MHz
VXI_H*	XI 输入高电平					V
VXI_L*	XI 输入低电平		0	-	0.45VDD	V
Duty	外部输入时钟占空比		45	50	55	%
R_{FB}	反馈电阻		40	50	60	kΩ
I_{VDDA}	起振电路功耗	起振期间			2	mA
		Rs=30，CL=10pF @ 8MHz		0.4		
		Rs=45，CL=10pF @ 8MHz		0.5		
		Rs=30，CL=5pF @ 32MHz		0.8		
		Rs=30，CL=10pF @ 32MHz		1		
		Rs=30，CL=20pF @ 32MHz		1.5		
gm*	振荡器跨导	起振期间，驱动强度可调	2		20	mA/V
T_{su}*	启动时间	从软件使能到输出稳定的时钟		0.5	2	ms

注：*-由设计保证，不在量产测试中测试。

34.10.2 片内高速 RC 振荡器特性（HSI）

（VDD= 3.3V，TA=25℃）

symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
F_{HSI}**	Frequency		5	8	13	MHz
DuCy	Duty Cycle		45	50	55	%
ACC*	振荡器的频率精度	TA=-40～105℃		±1		%
		TA=-10～85℃		±0.6		
		TA=0～85℃				
T_{su}*	启动时间	从软件使能到震荡到目标频率			12	us
I_{VDD}	振荡器功耗	TA=25℃ @ 8MHz		1.2		mA

注：*-振荡器频率在 TA=25℃测量并存储到 FLASH 中，振荡器的频率稳定性由设计保证，量产芯片中并未全部测量。

注：**-振荡器频率不校正，通过调节 PLL 分频系数进行 PLL 输出频率校正，绝对精度<±1%。

34.10.3 片内低速 RC 振荡器特性（LSI）

（VDD= 3.3V，TA=25℃）

symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
F_{LSI}**	Frequency		24	32	40	kHz
DuCy	Duty Cycle		45	50	55	%
ACC*	振荡器的频率精度	TA=-40～105℃	-1.25	±0.35		%

注：*-由设计保证，不在量产测试中测试。

注：**-振荡器频率无校正，FLASH 存储其 TA=25℃的频率值，可通过查询 FLASH NVR 区域获得实际振荡频率。

34.10.4 锁相环特性

（VDD= 3.3V，TA=25℃）

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{PLL_IN}	PLL 输入时钟	4		26	MHz
F_{PLL_OUT}	PLL 输出时钟	200	400	500	MHz
T_{LOCK}*	PLL 锁定时间		0.5	2	ms
T_{Jitter}*	Cycle to Cycle Jitter			300	ps

注：*-由设计保证，不在量产测试中测试。

34.11 Flash 存储器特性

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
T_{PROG}	Program time		8	8	10	us
T_{SECTOR_ERASE}	Sector Erase time		3.2	3.2	4	ms

T_{CHIP_ERASE}	Chip Erase time		16	16	20	ms
N_{END}	FLASH Endurance	TA=-40~150℃	100			kCycles
T_{RET}	Data Retention	TA=85℃ with 10k sector endurance	20			Years
		TA=85℃ with 100k sector endurance	5			

34.12 EMC 特性

34.12.1 电磁敏感特性（EMS）

Symbol	Parameter	Conditions	Min.
VFESD	Voltage limits to be applied on any I/O pin to induce a functional disturbance	VDD = 3.3 V, TA = +25 °C,fHCLK= 200 MHz, conforming to IEC 61000-4-2	
VEFTB	Fast transient voltage burst limits to be applied through 100 pF on VDD and VSS pins to induce a functional disturbance	VDD = 3.3 V, TA = +25 °C,fHCLK = 200 MHz, conforming to IEC 61000-4-4	

34.12.2 电磁干扰特性（EMI）

Symbol	Parameter	Conditions	Monitored frequency band	Max vs. [f_{HSE}/f_{HCLK}] 8 MHz / 200 MHz	Unit
S_{EMI}	Peak level	VDD = 3.6 V, TA = 25 °C, LQFP100 package compliant with IEC 61967-2	0.1 MHz to 30 MHz		dBµV
			30 MHz to 130 MHz		
			130 MHz to 1 GHz		
			1 GHz to 2 GHz		
			EMI Level		-

34.13 电灵敏度特性

34.13.1 ESD

Symbol	Ratings	Conditions		Class	Max	Unit
VESD(HBM)	ectrostatic discharge voltage (human body model)	TA = +25 °C, conforming to ANSI/ESDA/JEDEC JS-001				V
VESD(CDM)	Electrostatic discharge voltage (charge device model)	TA = +25 °C, conforming to ANSI/ESDA/JEDEC JS- 002	LQFP100			V
			LQFP80			
			Other packages			

34.13.2 Latch-Up

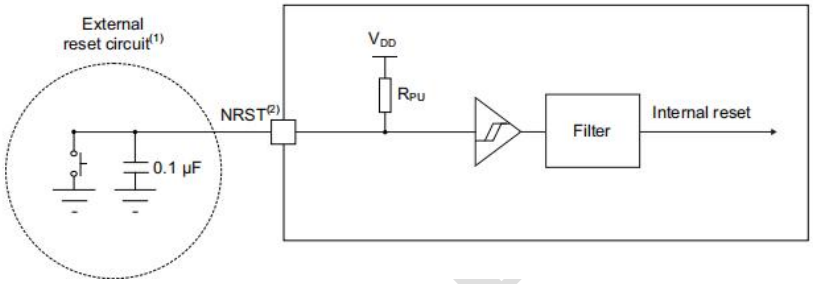
Symbol	Ratings	Conditions	Class
LU	Static latch-up class	TA = +150 °C conforming to JESD78E	

34.14 通用输入/输出端口特性

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{IL}	Low level input voltage	GPIO Hysteresis On			0.48*VDD-0.12	V
V_{IH}	High level input voltage	GPIO Hysteresis On	0.54*VDD+0.08			
C_{IO}	IO pin capacitance	Total Capacitance should add Package Cap:1.5pF		2		pF
R_{PU}	weak pull-up equivalent resistor		8.3	10	12	kΩ
R_{PD}	weak pull-down equivalent resistor		8.3	10	12	kΩ
V_{OL}	Low level output voltage	IIO=+6mA, DS=0	-	-	0.57	V
		IIO=+20mA, DS=1	-	-	0.66	
V_{OH}	High level output voltage	IIO=+6mA, DS=0	VDD-0.59	-	-	V
		IIO=+20mA, DS=1	VDD-0.68	-	-	
R_{NRST}	NRST PIN Pull-up resistor		75	100	125	kΩ

34.15 NRST 引脚特性

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
VIL(NRST)	NRST input low level voltage	-	-	-	0.3_xVDD	V
VIH(NRST)	NRST input high level voltage	-	0.7_xVDD	-	-	
Vhys(NRST)	NRST Schmitt trigger voltage hysteresis	-	-	200	-	V
RPU	Weak pull-up equivalent resistor	VIN = VSS	25	40	55	V
VF(NRST)	NRST input filtered pulse	-	-	-	70	V
VNF(NRST)	NRST input not filtered pulse	1.71 V ≤ VDD ≤ 3.6 V	350	-	-	pF



34.16 高精度 PWM 特性

（VDD= 3.3V，TA=25°C）**HRPWM Characteristics**

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
F_{HRPWM}			120	-	180	MHz
T_{HRPWM}			5.56	-	8.33	ns
T_{RES}	high-resolution step size	F_{HRPWM}=180MHz, TA=-40~125°C	-	174	-	ps
ResHRPWM	Timer resolution		-	-	16	bit
T_{DTG}	Dead time generator clock period		1/32	-	8	T_{HRPWM}
		F_{HRPWM}=180 MHz	0.174	-	44.44	ns
$\|T_{DTR}\|$ / $\|T_{DTF}\|$ max	Dead time range (absolute value)		-	-	1023	t_{DTG}
		F_{HRPWM}=180 MHz	-	-	45.46	µs
fCHPFRQ	Chopper stage clock frequency		1/256	-	1/16	F_{HRPWM}
		F_{HRPWM}=180 MHz	0.703	-	11.25	MHz
T_{1STPW}	Chopper first pulse length		16	-	256	T_{HRPWM}
		F_{HRPWM}=180 MHz	0.089	-	1.422	µs

（VDD= 3.3V，TA=25℃）**HRPWM output response to fault protection**

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
$T_{LAT(DFLT)}$	Digital fault response latency	Propagation delay from HRPWM_FLTx digital input to HRPWM_OUTxy output pin	-	8.5	18.89	ns
$T_{W(FLT)}$	Minimum Fault pulse width		6.67	-	-	
$T_{LAT(AFLT)}$	Analog fault response latency	Propagation delay from comparator CMPx_INP input pin to HRPWM_OUTxy output pin	-	15.11	29.28	

（VDD= 3.3V，TA=25℃）**HRPWM output response to external events 0 to 9 (Fast mode)**

Symbol	Parameter	Condition	Min	Typ(2)	Max.	Unit
$T_{LAT(DEVT)}$	Digital external event response latency	Propagation delay from HRPWM_EVTx digital input to HRPWM_OUTxy output pin	-	11.33	21.72	ns
$T_{W(EVT)}$	Minimum external event pulse width		6.67	-	-	
$T_{LAT(AEVT)}$	Analog external event response latency	Propagation delay from comparator CMPx_INP input pin to HRPWM_CHxy output pin	-	17.94	29.27	

（VDD= 3.3V，TA=25℃）**HRPWM output response to external events 0 to 9 (Slow mode)**

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
$T_{LAT(DEVT)}$	Digital external event response latency	Propagation delay from HRPWM1_EVTx digital input to HRPWM_CHxy output pin	-	52.89	62.33	ns
$T_{LAT(AEVT)}$	Analog external event response latency	Propagation delay from COMPx_INP input pin to HRPWM_CHxy output pin	-	58.56	71.78	ns
$T_{W(EVT)}$	Minimum external event pulse width		6.67	-	-	ns
$T_{JIT(EVT)}$	External event response jitter	Jitter of the delay from HRPWM1_EVTx digital input or COMPx_INP to HRPWM_CHxy output pin	-	-	1	T_{HRPWM}

（VDD= 3.3V，TA=25℃）**HRPWM synchronization input / output**

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
$T_{W(SYNCIN)}$	Minimum pulse width on SYNCIN inputs, including HRPWM_SCIN		2	-	-	T_{HRPWM}
$T_{RES(ESR)}$	Response time to external synchronization request		-	-	3	T_{HRPWM}
$T_{W(SYNCOUT)}$	Pulse width on HRPWM_SCOUT output		-	16	-	T_{HRPWM}
		F_{HRPWM}=180 MHz	-	88.89	-	ns

34.17 模拟外设特性

34.17.1 模数转换器特性

（测试条件：V_{DDA}=3.3V，TA=25℃，F_ADC=60MHz，除非另有说明）

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DDA}	模拟电源电压		2.7	3.3	3.6	V
Resolution	ADC 分辨率			13		bit
V_{ref+}	ADC 正参考电压		2		V_{DDA}	V
V_{ref-}	ADC 负参考电压，芯片内部连接到 VSSA			0		V
V_{AIN}	ADC 输入电压范围		VSSA		V_{REF+}	V
C_{IN}	ADC 输入电容			1.6		pF
F_{ADC}	ADC 工作时钟			60		MHz
F_S	ADC 采样率				5	MSps
T_{sample}	ADC 采样时间，ADC 时钟周期数		2			Tadc
T_{con}	总转换时间	Tcon=Tsample（2~10 Tadc）+Tconv（10Tadc）				Tadc
DNL*	微分非线性			TBD		LSB
INL*	积分非线性			TBD		LSB
I_{VDDA}	ADC 功耗	单端/差分，5MSPS		2		mA
DR*	ADC 动态范围	差分输入，-40dBFS 输入测量，10kHz 正弦信号				dB
		单端输入，-40dBFS 输入测量，10kHz 正弦信号				dB
SNDR*	ADC 信号与噪声失真比	差分输入，-6dBFS 输入测量，10kHz 正弦信号				dB
		单端输入，-6dBFS 输入测量，10kHz 正弦信号				dB

注：*-由设计保证，不在量产测试中测试。

34.17.2 数模转换器特性

（测试条件为：V_{DDA}=3.3V，TA=25℃，除非另有说明）

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DDA}	模拟电源电压		2.7	3.3	3.6	V
Resolution	DAC 分辨率			12		bit
V_{ref+}	DAC 正参考电压，片内连接到 V_{DDA}				V_{DDA}	V
V_{ref-}	DAC 负参考电压，片内连接到 V_{SSA}			V_{SSA}		V
V_{DACOUT}	DAC 输出电压范围		V_{SSA}		V_{REF+}	V
$t_{setting}$*	DAC 建立时间	DAC Data 0->4095 DAC Data 4095->0 建立到 1LSB 精度， CL=50pF			3	us
DNL*	微分非线性					LSB

INL*	积分非线性					LSB
I_{VDDA}	DAC 功耗					mA
Update Rate	DAC 数据更新速率					MS/s

注：*-由设计保证，不在量产测试中测试。

34.17.3 比较器特性

（V_{DDA}=3.3V，TA=25℃）

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DDA}	模拟电源电压		3.1	3.3	3.6	V
V_{in}	比较器输入电压范围		0		V_{DDA}	V
V_{offset}*	比较器失调电压			±5		mV
V_{hyst}*	比较器迟滞电压	0~30mV 可调	0		30	mV
T_{DLY}*	比较器延时	从比较器输入引脚到输出引脚，CL=30pF		15	20	ns

注：*-由设计保证，不在量产测试中测试。

34.17.4 温度传感器特性

（VDDA=3.3V，TA=25℃）

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
Slope*	温度曲线斜率		-1.96	-1.94	-1.86	mV/℃
V_0	0 摄氏度电压			688		mV
T_{START}	启动时间		4		10	us

注：为了保证温度传感器测量精度，可开启 ADC 过采样。

注：*-由设计保证，不在量产测试中测试。

34.18 计时器特性

Symbol	Parameter	Conditions	Min.	Max.	Unit
$t_{res(TMR)}$	计时器分辨率时间	-	1	-	$t_{TMRxCLK}$
		$f_{TMRxCLK}$=180 MHz	5.55	-	ns
f_{EXT}	CH1至CH4上的定时器外部时钟频率	-	0	$f_{TMRxCLK}$/2	MHz
		$f_{TMRxCLK}$=180 MHz	0	90	MHz
Res_{TMR}	计时器分辨率	TMRx（TMR3和TMR4除外）	-	16	bit
		TMR3和TMR4	-	32	
$t_{COUNTER}$	16位计数器时钟周期	-	1	65536	$t_{TMRxCLK}$
		$f_{TMRxCLK}$=180 MHz	0.00555	363.7	us
t_{MAX_COUNT}	32位计数器的最大可能计数	-	-	65536x65536	$t_{TMRxCLK}$
		$f_{TMRxCLK}$=180 MHz	-	23.837	s
$t_{W(INDEX)}$	ETR输入上的索引脉冲宽度	-	2	-	Tck
$t_{w(TI1,TI2)}$	最小脉冲宽度在除方向时钟x1外的所有	-	2	-	Tck

	编码器模式下的TI1和TI2输入上				
	最小脉冲宽度在方向时钟x1中的TI1和TI2输入上	-	3	-	Tck

1. TMRx 是一个通用术语，其中 x 代表 0,1,2,3,4,,7,8,9 或 10。
2. 设计保证。

IWDG min/max timeout period at 32 kHz (LSI)(1)(2)				
Prescaler divider	**PR[2:0] bits**	**Min timeout RL[11:0]=0x000**	**Max timeout RL[11:0]=0xFFF**	**Unit**
/4	0	0.125	512	ms
/8	1	0.250	1024	
/16	2	0.500	2048	
/32	3	1.0	4096	
/64	4	2.0	8192	
/128	5	4.0	16384	
/256	6或7	8.0	32768	

1.设计保证。
2.准确的定时仍然取决于 APB 接口时钟与 LSI 时钟的相位，因此始终存在一个完整的 RC 不确定性周期。

34.19 通信接口特性

34.19.1 I2C 特性

I2C 接口满足 I2C 总线规范的时序要求：

- 标准模式（Sm）：比特率高达 100 kbit/s
- 快速模式（Fm）：比特率高达 400 kbit/s
- 快速模式 Plus（Fm+）：比特率高达 1 Mbit/s。

当 I2C 外围设备正确配置时，I2C 定时要求通过设计得到保证。

Minimum I2CCLK frequency in all I2C modes					
Symbol	**Parameter**	**Conditing**		**Min**	**Unit**
f（I2CCLK）	I2C CLK频率	标准模式		2	MHz
		快速模式	模拟过滤器开启DNF=0	8	
			模拟过滤器关闭DNF=1	9	
		快速模式Plus	模拟过滤器开启DNF=0	17	
			模拟过滤器关闭DNF=1	16	

所有 I2C SDA 和 SCL I/O 均嵌入模拟滤波器。模拟滤波器特性见下表：

analog filter characteristics(1)				
Symbol	**Parameter**	**Condition**	**Min**	**Unit**
t_{AF}	模拟滤波器抑制的峰值的最大脉冲宽度	50(2)	90(3)	ns

1. 设计保证。

34.19.2 SPI 特性

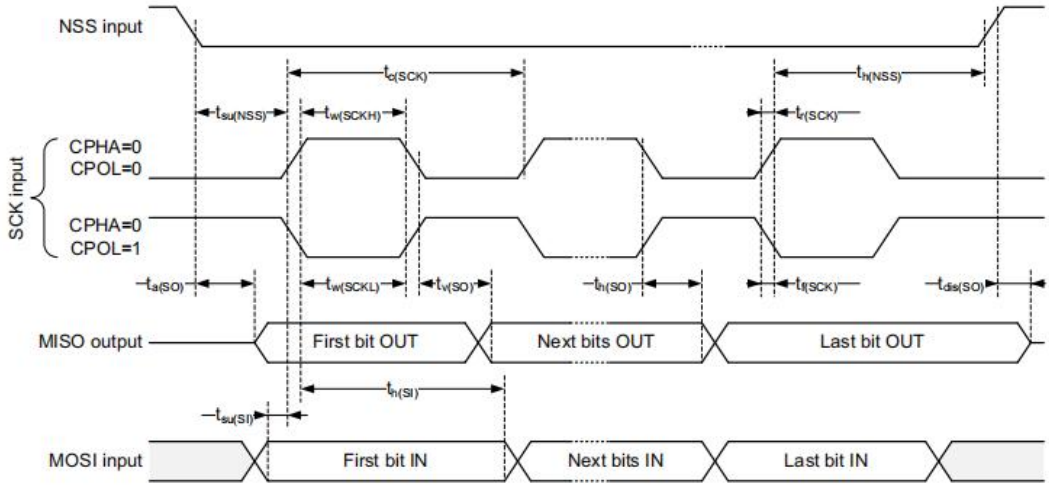
除非另有规定，下表中给出的 SPI 参数来自于在环境温度、fPCLKx 频率和电源电压条件下进行的测试。

- 输出速度设置为 DR=1
- 电容负载 C=30 pF
- 测量点在 CMOS 水平上进行：0.5 x VDD

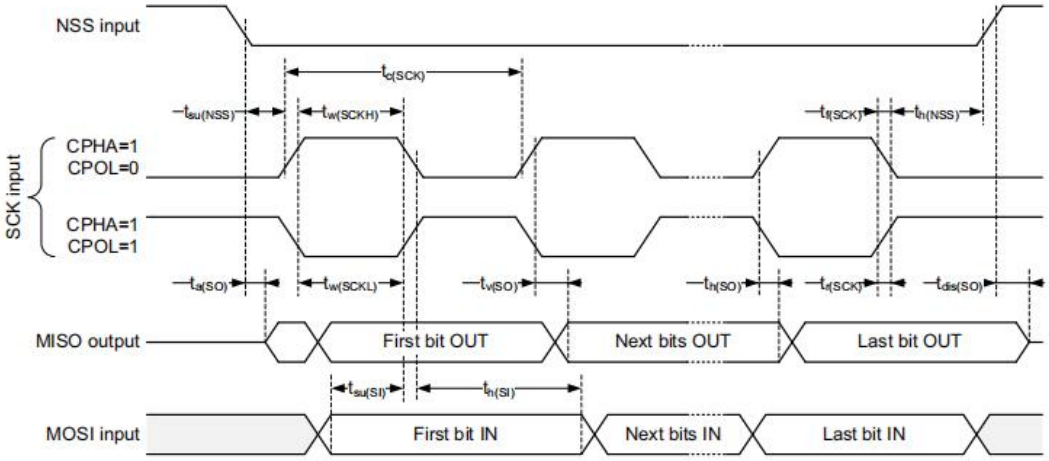
有关输入/输出备用功能特性（NSS、SCK、MOSI、MISO 用于 SPI）的更多详细信息，请参阅 I/O 端口特性。

Symbol	Parameter	Conditing	Min	Typ	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI时钟频率	主模式	-	-	75	MHz
		主发射机模式			50	
		从属接收器模式			50	
		从模式发射机/全双工			41	
$t_{su(NSS)}$	NSS设置时间	从模式，SPI预分频器=2	$4*T_{pclk}$	-	-	-
$t_{h(NSS)}$	NSS保持时间	从模式，SPI预分频器=2	$2*T_{pclk}$	-	-	-
$t_{W(SCKH)}$ $t_{w(SCKL)}$	SCK高和低时间	主模式	Tpclk-1			-
$t_{su（MI）}$	数据输入设置时间	主模式	4	-	-	ns
$t_{su(SI)}$		从属模式	3	-	-	
$t_{h(MI)}$	数据输入保持时间	主模式	4	-	-	ns
$t_{h(SI)}$		从属模式	1	-	-	
$t_{a(MO)}$	数据输出访问时间	从属模式	9	-	34	ns
$T_{dis(SO)}$	数据输出禁用时间	从属模式	9	-	16	ns
$t_{h(SO)}$	数据输出有效时间	从属模式	-	9	12	ns
$t_{V(MO)}$		主模式	-	3.5	4.5	
$t_{h(SO)}$	数据输出保持时间	从模式	6	-	-	ns
$t_{h(MO)}$		主模式	2	-	-	

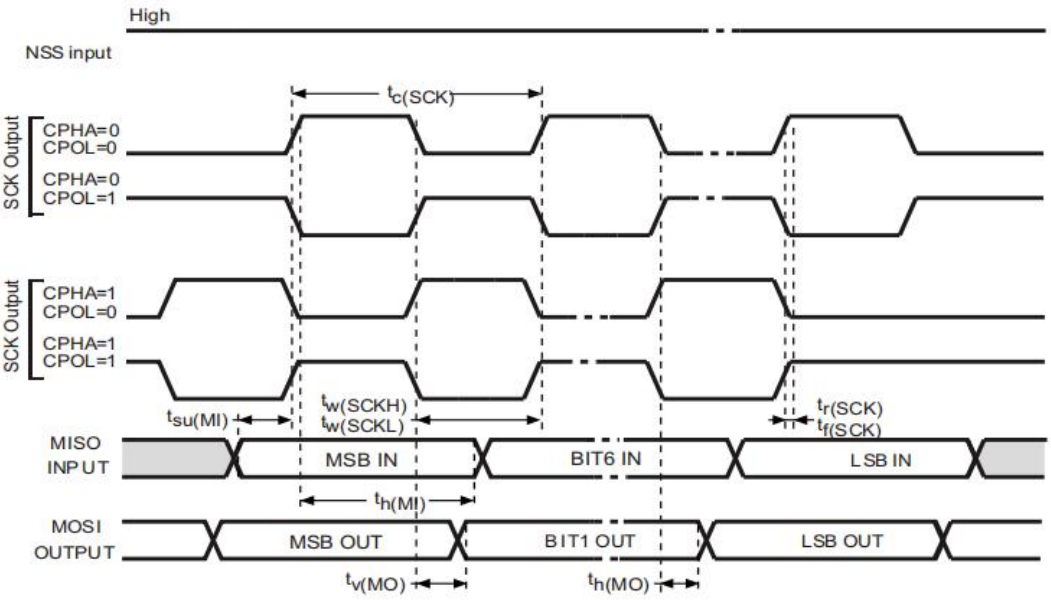
- SPI 时序图-从模式和 CPHA=0



- SPI 时序图-从模式和 CPHA=1



- SPI 时序图-主模式



34.19.3 USB 特性

设备 USB 接口符合 USB 规范 2.0 版，用于全速设备操作。

Symbol	Parameter	Conditing	Min	Max	Unit
V_{DD}	USB收发器工作电压	2.7	-	3.6	V
R_{PUI}	闲置期间的片内USB_DP上拉值	900	1250	1500	Ω
R_{PUR}	接收期间的片内USB_DP上拉值	1400	2300	3200	
Z_{sDRV}	输出驱动器阻抗	28	36	44	Ω

34.19.4 UART 接口特性

除非另有规定，表中给出的 UART 参数是根据表中总结的环境温度、fPCLKx 频率和 VDD 电源电压条件下进行的测试得出的，配置如下：

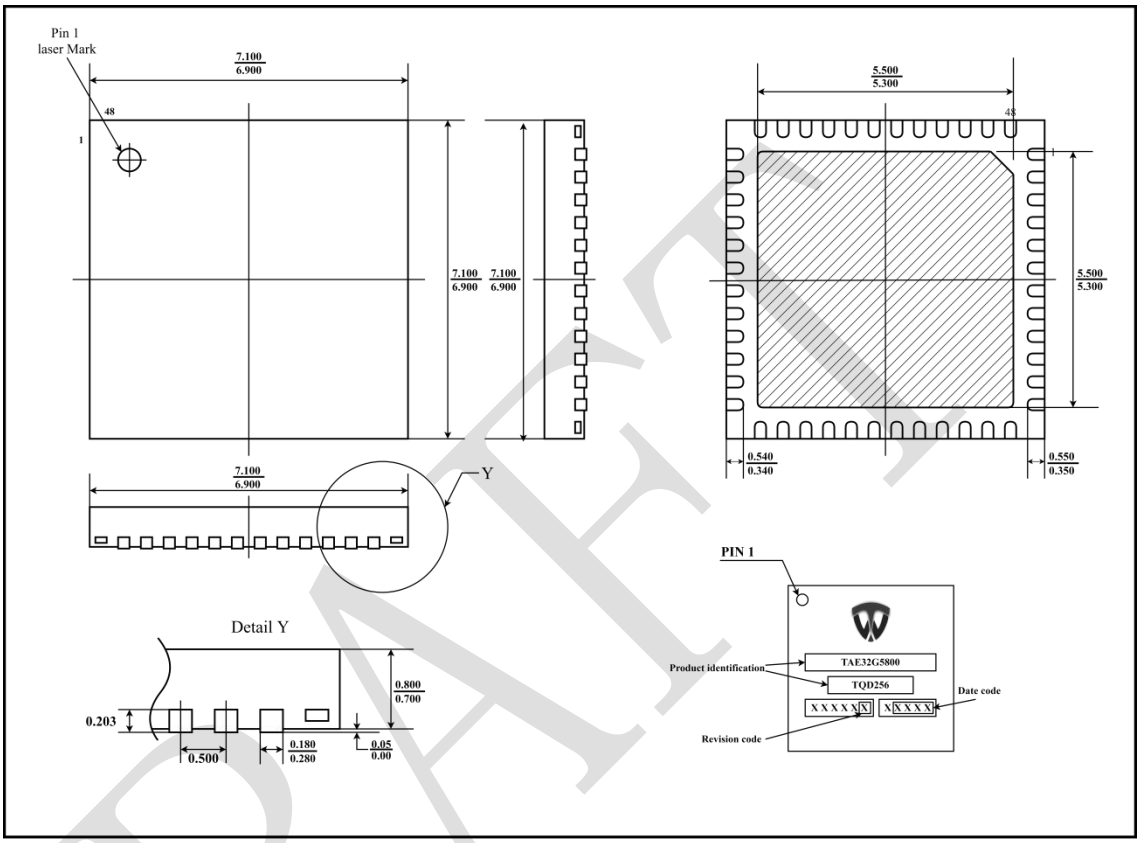
- 输出速度设置为 DR=0
- 电容负载 C=30 pF
- 测量点在 CMOS 水平下进行：0.5 VDD

有关输入/输出备用功能特性（NSS、CK、TX、RX 用于 UART）的更多详细信息，请参阅 I/O 端口特性。

UART electrical characteristics						
Symbol	**Parameter**	**Conditing**	**Min**	**Typ**	**Max**	**Unit**
f_{CK}	UART时钟频率	主模式	-	-	21	MHz
		从属模式	-	-	22	
$t_{su\ (NSS)}$	NSS设置时间	从属模式	t_{ker}+2	-	-	ns
$t_{h\ (NSS)}$	NSS保持时间	从属模式	2	-	-	
$t_{w\ (CKH)}$ $t_{w\ (CKDL)}$	CK高低时间	主模式	$1/f_{ck}/2-1$	$1/f_{ck}/2$	$1/f_{ck}/2+1$	ns
$t_{su\ (RX)}$	数据输入设置时间	主模式	t_{ker}+2	-	-	ns
		从属模式	2	-	-	
$t_{SU\ (RX)}$	数据输入保持时间	主模式	1	-	-	
		从属模式	0.5	-	-	
$t_{V\ (TX)}$	数据输出有效时间	主模式	-	0.5	1.5	ns
		从属模式	-	10	22	
$t_{h\ (RX)}$	数据输出保持时间	主模式	0	-	-	
		从属模式	7	-	-	

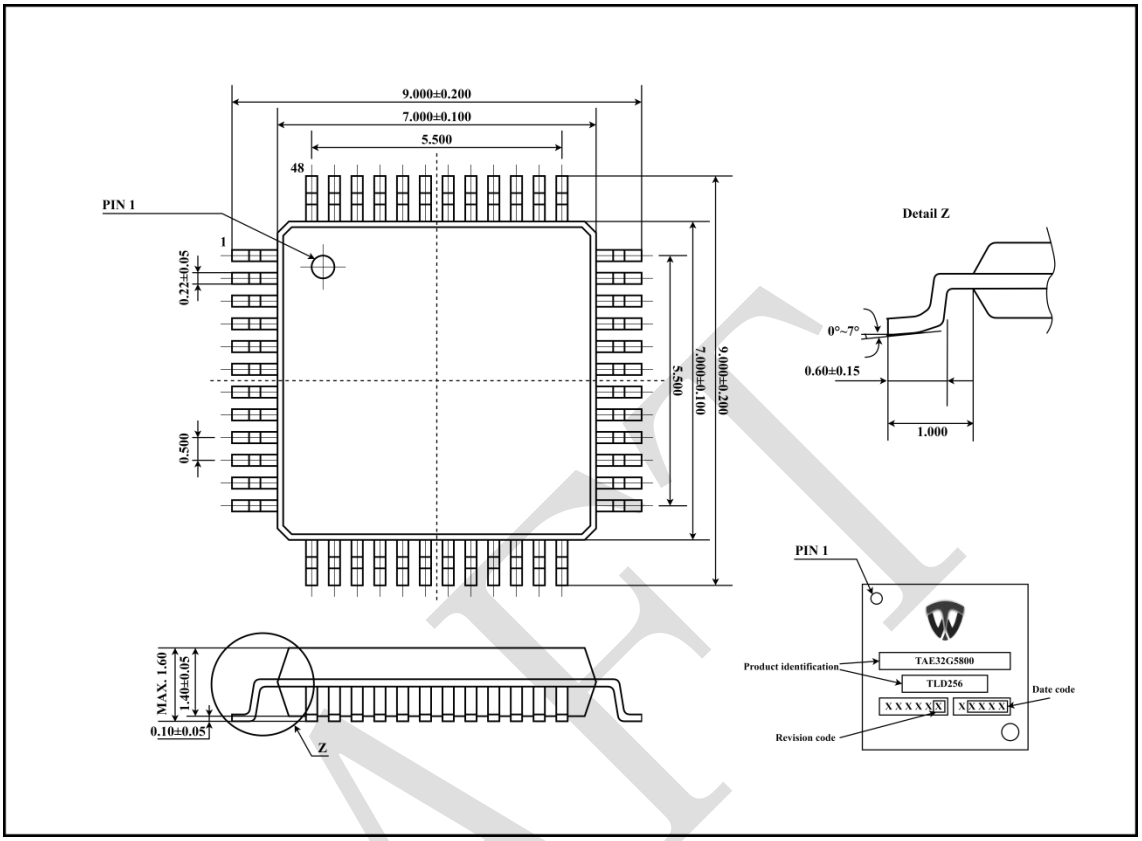
35 封装信息

QFN48:



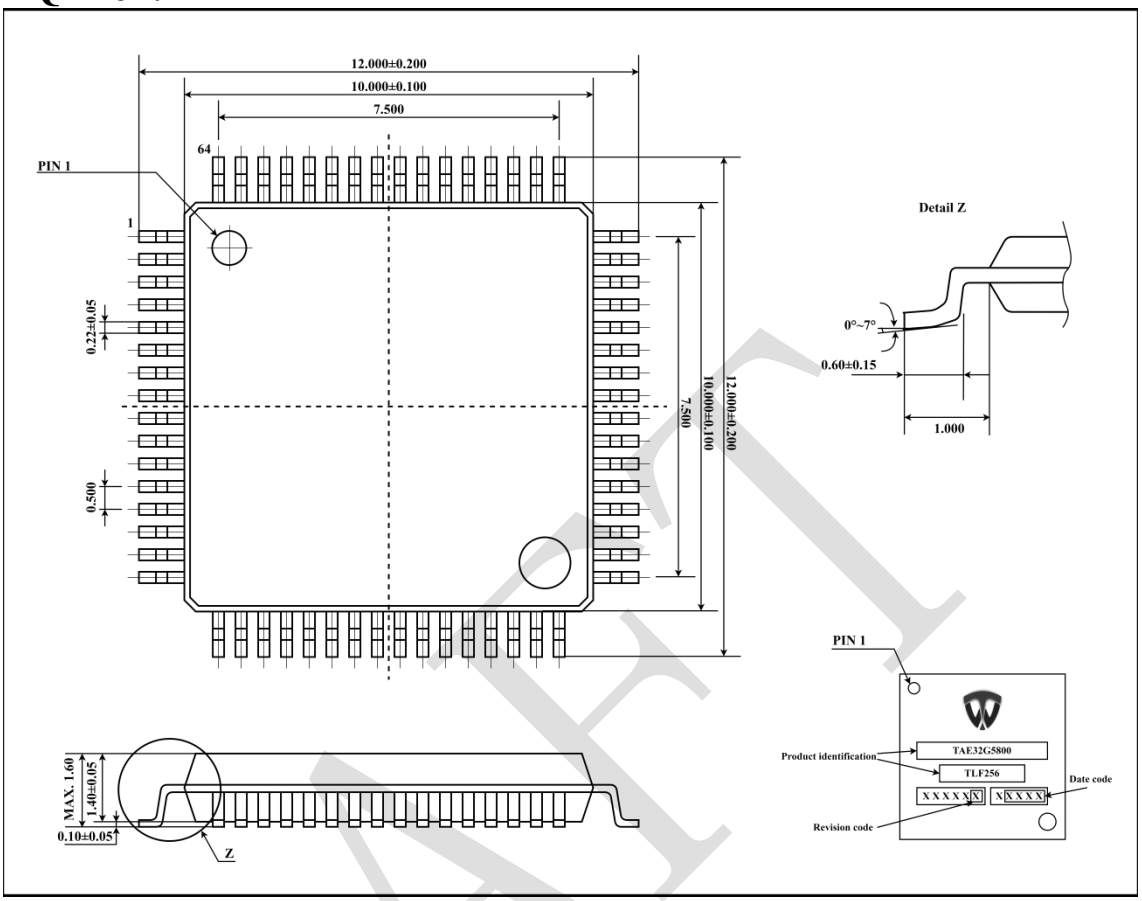
NOTE：UNITS OF MEASURE = MILLIMETER

LQFP48:



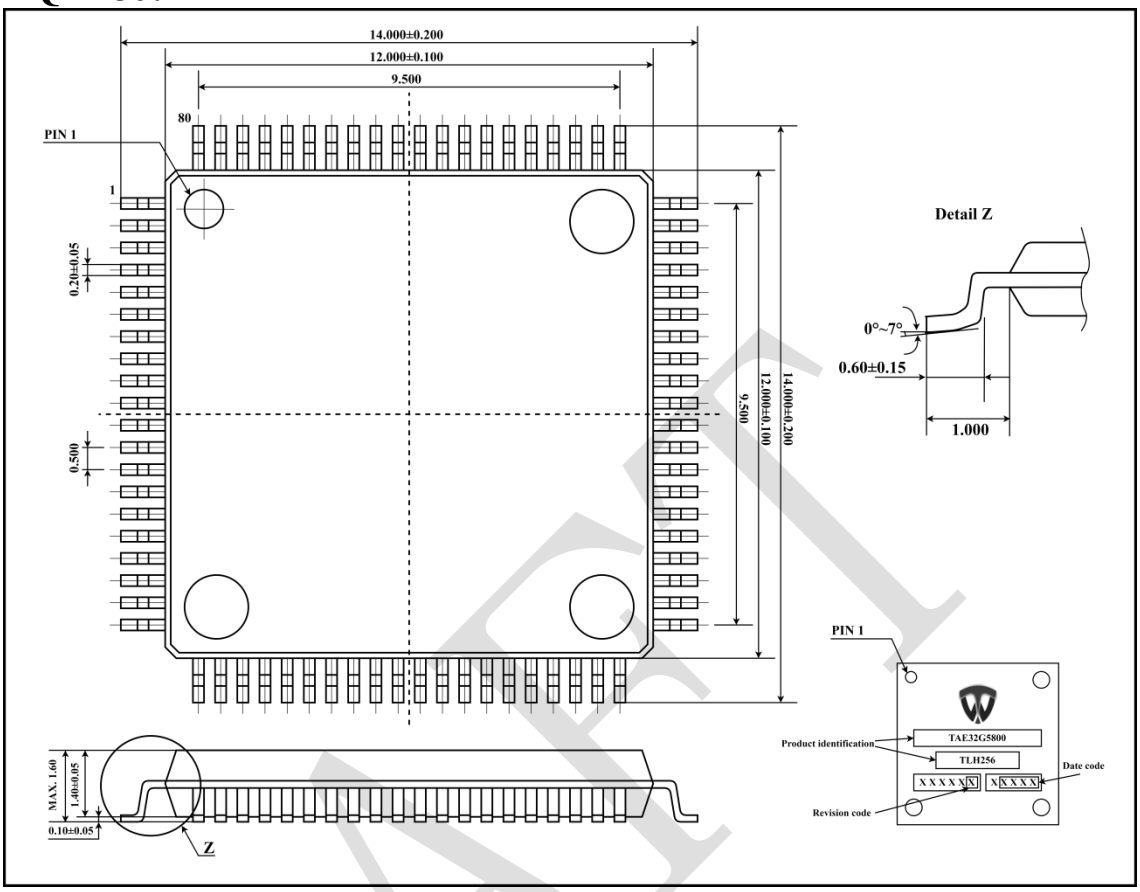
NOTE：UNITS OF MEASURE = MILLIMETER

LQFP64:



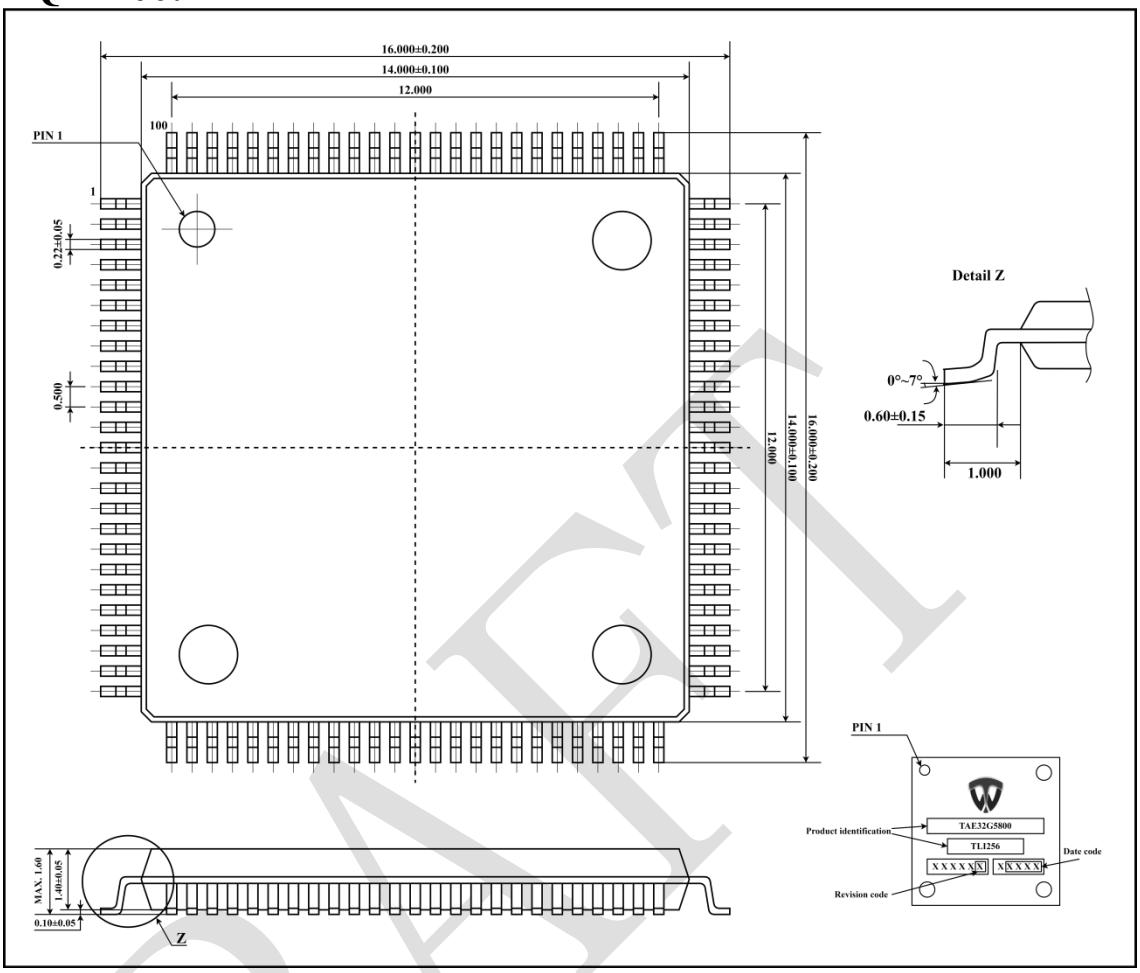
NOTE：UNITS OF MEASURE = MILLIMETER

LQFP80:



NOTE：UNITS OF MEASURE = MILLIMETER

LQFP100:



NOTE：UNITS OF MEASURE = MILLIMETER

36 订购信息

Example:

TAE32 G5800 T L I 256

产品序列

TAE32 = ARM Cortex-M4 32bit系列

产品型号

G5800 = 产品型号

结温范围

G = 工业级温度范围为-40℃至+105℃
T = 工业级温度范围为-40℃至+150℃

封装类型

E = 封装类型：eLQFP
L = 封装类型：LQFP
W = 封装类型：WQFN
Q = 封装类型：QFN

引脚数

D = 引脚数48引脚
F = 引脚数64引脚
H = 引脚数80引脚
I = 引脚数100引脚

Flash大小

256 = FLASH 256KB
128 = FLASH 128KB

版本历史

日期	版本	版本记录
2022/11/17	V0.1	初始版本
2022/12/07	V0.11	修改寄存器描述及格式调整
2022/12/21	V0.12	ADC 修改更新 1. 修改主要特性中高性能特性描述 2. 修改 表 14-2 3. 修改 图 14-2 4. 修改 14.4.3 单端和差分输入通道 内容 5. 修改 14.4.6 通道选择 内容 HRPWM 修改更新 1. 修改 表 17-1 hrpwm_hclk/hrpwm_clk 的信号描述 2. 修改 17.4.3 时钟 内容 I2C 修改更新 1. 修改部分寄存器描述
2023/1/4	V0.13	CMP 修改更新 1. 修改 表 16-2 2. 修改 CMPx_CR.INM 的描述 HRPWM 修改更新 1. 修改 图 17-69 2. 修改 DAC 锯齿波触发事件内容
2023/1/7	V0.14	PLL 修改更新： 1. 修改图 7-3 自举配置修改： 1. 修改表 4-2 TIMER 修改更新： 1. 修改寄存器 TMRx_CCMR.CC0S 描述 SYSCTRL 修改更新： 1. 修改 SYSCTRL_PLCR.AVCCLVL 电压阈值描述 FLASH 修改更新： 1. 修改特性描述，增加 FLASH 128K/256K 单双 Bank Sector 大小 CMP 修改更新： 1. 修改 CMP 负端选源表格，DAC7/DAC8 互换
2023/1/16	V0.15	ADC 修改更新： 1. 修改 14.4.1：ADC 时钟描述 2. 修改 14.4.4：增加使能限制描述 3. 修改 14.4.18：增益补偿/偏置补偿部分 4. 修改 14.4.18：出厂校准/温度传感器部分 DAC 修改更新： 1. 修改 15.4.2：增加使能限制描述

2023/2/1	V0.2	SPI 修改更新 1. 修改 SPI_CTRL[5:4]的 STM 拆分为 RXEN(BIT5)和 TXEN(BIT4) CAN 修改更新 1. 修改 AFWL、EWL、TSNEXT、TBSEL 的寄存器描述 2. 增加时钟域交叉章节，描述 CAN 中系统时钟和功能时钟分别控制的功能模块 3. 增加图 20-2 时钟域交叉示意图 I2C 修改更新： 1. 修改图 18-1 HRPWM 修改更新 1. 修改了 HRPWM_EEFxRx 寄存器滤波位的描述内容 RCU 修改更新 1. 修改寄存器 RCU_CCR.P1PSC 描述与默认值 2. 修改寄存器 RCU_PLL0CR.BDS 描述
2023/2/10	V0.21	修改 QFN48 封装图
2023/2/28	V0.22	修改表 3-1 引脚定义缩写说明 1. 修改_a 定义 2. 增加_fa 定义 CORDIC 修改更新 1. 修改表 12-12 名称与描述 2. 修改表 12-13 名称 3. 修改表 12-14 描述 PDM 修改更新 1. 修改寄存器 PDM_ENABLEx.INMOD 描述 2. 修改寄存器 PDM_DFCRx.DOSR 描述 3. 修改寄存器 PDM_CFCRx.COSR 描述 I2C 修改更新 修改表 18-3 HRPWM 修改更新 1. 修改寄存器 HRPWM_BMTRGR0.PER4EVT8 描述 2. 修改寄存器 HRPWM_BMTRGR0.PER0EVT7 描述 CMP 修改更新 1. 修改寄存器 CMPx_CR.INM 描述
2023/3/2	V0.23	DMA 更新 1. 修改寄存器 DMA_CTR.PRI 描述 TMRx 修改更新 1. 修改表 30-2 2. 修改表 31-2
2023/3/7	V0.24	HRPWM 修改更新 1. 修改寄存器 HRPWM_RSTxR/HRPWM_RSTxER 描述 2. 修改寄存器 HRPWM_CAPAxCR/HRPWM_CAPAxCER 描述 3. 修改寄存器 HRPWM_CAPBxCR/HRPWM_CAPBxCER 描述
2023/3/9	V0.25	HRPWM 修改更新 1. 修改寄存器 HRPWM_RSTxR/HRPWM_RSTxER 描述

		2. 修改寄存器 HRPWM_CAPAxCR/HRPWM_CAPAxCER 描述 3. 修改寄存器 HRPWM_CAPBxCR/HRPWM_CAPBxCER 描述 CORDIC 修改更新 1. 修改表 12-7：q1.15 精度 2. 修改双曲函数的 q1.15 精度 3. 修改表 12-12：q1.15 精度 4. 修改表 12-14：q1.15 精度
2023/3/17	V0.26	DAC 修改更新： 1. 修改 DACx_CR.TGTRIG 寄存器中三角波触发事件描述 ADC 修改更新 1. 修改 14.4.5 内容描述 2. 修改 14.4.18 内容描述
2023/4/7	V0.3	DAC 修改更新 1. 修改 15.4.5 表格描述 2. 修改 DACx_CR 寄存器描述，修改如下域描述（STINCTRIG/STRSTTRIG/TGTRIG） I2C 修改更新 1. 修改 18.4.6 章节内容描述 2. 修改 18.4.2 中，I2C_INTR.RXATI→I2C_INTR.RXADI TMRx 修改更新 1. 修改 TMR0/1/2 的 TMRx_BPR、TMRx_BER 描述 2. 修改 TMR9/10 的 TMRx_BPR、TMRx_BER 描述 3. 增加图 29-21、图 31-35 4. 修改 TMR0/1/2 的 TMRx_CIR 描述 5. 修改 TMR3/4 的 TMRx_CIR 描述 6. 修改 TMR9/10 的 TMRx_CIR 描述 HRPWM 修改更新 1. 修改 HRPWM_DLLCR.DLLSTART 描述 USB 修改更新 1. 更新 USB 寄存器描述 GPIO 修改更新 1. 修改 GPIOx_IMR 寄存器描述 电气特性修改更新 1. 修改 34.17 计时器特性 内容
2023/4/28	V0.4	HRPWM 修改更新 1. 修改 HRPWM_CR2 寄存器描述 2. 更新 HRPWM 比较值最大值使用限制 3. 修改 HRPWM 表格 17-15 增加 BM 寄存器 DAC 修改更新 1. 修改 DAC 计算公式 ADC 修改更新 1. 更新 ADC_LR/ADC_JLR 寄存器描述 寄存器描述更新 1. RCU、SYSCTRL、FLASH、TMR0-2、TMR3-4、TMR9-10 寄存器描

		述均更新
2023/5/6	V0.41	HRPWM 修改更新 1. 修改 HRPWM Event 来源表格及寄存器描述 2. 修改 HRPWM Fault 来源表格及寄存器描述 CMP 修改更新 1. 修改 CMP 消抖宽度描述 所有寄存器描述更新同步
2023/7/7	V0.42	SYSCTRL 修改更新 1. 修改 SYSCTRL_PLCR.CPULKEN，SYSCTRL_PLCR.FLSECCEN I2C 修改更新 1. 修改 I2C_INTREN，I2C_INTR 描述 TMRx 修改更新 1. 修改完善图 29-21，图 31-35 GPIO 修改更新 1. 修改图 8-13 ADC 修改更新 1. 修改 SYSCTRL_ATCR.VBFEN 描述 2. 修改表 14-2 V_{REF+}说明 CAN 修改更新 1. 修改图 20-1 2. 修改 CAN_CTRL.AFWL 描述 SPI 修改更新 1. SPI_ENABLE.CSSEL UART 修改更新 1. 修改 UART_INT.RUIF，UART_INT.ROIF 描述 HRPWM 修改更新 1. 修改 HRPWM_RSTxR/ HRPWM_RSTxER/HRPWM_CAPAxCR 寄存器描述 2. 修改图 17-24 FLASH 修改更新 1. 修改表 5-2
2023/7/12	V0.43	封装信息修改更新 1. 所有封装图增加丝印示意图
2023/8/10	V0.44	UART 修改更新 1. 修改图 19-8 HRPWM 修改更新 1. 修改 HRPWM_BMTRGR0.CMPB3 描述 I2C 修改更新 1. 修改 I2C_ENABLE.I2CEN/I2C_CTRL.PECBYTE/I2C_INTR.MOHI 寄存器描述 RCU 修改更新 1. 修改 RCU_SRSTSR.SQRST/ RCU_SRSTSR.SQRSTE 默认值 SYSCTRL 修改更新 1. 修改 SYSCTRL_PECR.VDDOCBE/ SYSCTRL_PECR.VDDOCRE/

		SYSCTRL_PECR.VDDOCIE 寄存器描述 FLASH 修改更新 1. 修改 FLASH_LPR.LW 寄存器描述 USB 修改更新 1. 修改 USB_TX0CTRL.SENTSTA 寄存器描述 DAC 修改更新 1. 修改 DACx_CR.PEN 寄存器描述 2. 修改 DACx_SR 寄存器名称 CAN 修改更新 1. 修改 CAN_BITTIME.S_SJW 名称 2. 修改主要特性中“支持 2 帧数据的接收缓冲区（RB）大小”为“支持 1 帧数据的接收缓冲区（RB）大小” 3. 20.4.5 中“它最大支持存储 2 帧 CAN 报文”改为“它最大支持存储 1 帧 CAN 报文” QEI 修改更新 1. 修改 QEI_QCTMR.CTMR 寄存器描述
2023/8/16	V0.45	HRPWM 寄存器修改更新 1. 修改触发事件长度相关描述 封装信息修改更新 1. 更新封装图数据信息 WWDG 修改更新 1. 修改超时值的计算公式 FLASH 修改更新 1. 修改主要特性描述 UART 修改更新 1. 19.4.3 UART 字符说明中奇偶检验位配置描述更新 2. 19.4.7 UART 接收器描述更新
2023/9/13	V0.46	引脚定义修改更新 1. TMR3_CH0→TMR3_CH0/ETR 2. TMR4_CH0→TMR4_CH0/ETR 3. TMR0_BK→TMR0_BK0 CAN 修改更新 1. 20.4.5 中“但是如果在 RB 已满且新的报文被接收前将 CAN_CTRL.RREL 置 1，则不会覆盖任何报文，新的报文将会被丢弃”改为“如果在 RB 已满且新的报文被接收前将 CAN_CTRL.RREL 置 1，则不会覆盖任何报文，也不会产生 ROIF 中断，但最早的报文将被丢弃”。 HRPWM 修改更新 1. 修改 HRPWM_RSTxER/HRPWM_CAPAxCER/ HRPWM_CAPBxCER 寄存器描述，SLVx 对应后面描述中的 PWMx 2. 修改图 17-43，图 17-44 3. 更新加窗模式描述 TMR 修改更新 1. 31.4.5 时钟选择中“外部时钟源模式 1”描述更新 “在该模式下，计数器将在选定输入信号的上升沿计数”改为“在该模式下，

		计数器将在选定输入信号的上升沿或下降沿计数”。 I2C 修改更新 1. 修改图 18-12，18-13，18-17，18-18 2. 修改 I2C_INTR.SEXTOI/I2C_INTR.MEXTOI 寄存器描述
2023/11/06	V0.47	I2C 修改更新 1. 修改 I2C_TIMING 寄存器描述 2. 修改表 18-4 SEXTOI 描述部分 FLASH 修改更新 1. 修改同步 FLASH_OPDR.BMD/FLASH_OPDR.BORLV 默认值 2. 更新 FLASH_OPDR 描述 TMRx 修改更新 1. 修改表 29-2 2. 修改表 30-2 3. 修改表 31-2 / 添加表 31-3 CAN 修改更新 1. 增加寄存器描述 CAN_CTRL.MUXSEL 封装信息修改更新 1. 更新所有丝印示意图
2023/11/28	V0.5	所有寄存器描述更新同步 RCU 修改更新 1. 修改同步版本内容描述