厦门国科安芯科技有限公司

AS32A601 芯片设计手册

32-bit RISC-V MCU

目录

1 简介

1.1 概要

AS32A601 是采用自研高性能 E7 内核的高性能、低功耗 MCU。

- 频率最高达 180MHz
- 工作温度范围支持-40℃~+125℃
- 工作电压支持 2.7V~5.5V
- LQFP144 封装
- 支持 ASIL-B 等级的安全功能 ISO26262

1.2 模块概述

下表是 AS32A601 的模块概述。

表 1.1 AS32A601 模块概述

模块	说明
内核	自研 E7 内核，带有 FPU 与 L1Cache：16KiB 数据缓存，16KiB 指令缓存，允许零等待访问嵌入式 Flash 与外部存储，最高频率 180MHz，带有 MPU。482DMIPS/2.68DMIPS/MHz。
时钟	外部晶振（OSC）：范围在 8MHz~40MHz 内部高频振荡器（FIRC）：16MHz 内部低频振荡器（SIRC）：32KHz 系统锁相环（PLL）：最大支持 500MHz 输出
存储	512KiB 内部 SRAM（带 ECC） 16KiB ICache 和 16KiB DCache（带 ECC） 512KiB D-Flash（带 ECC） 2MiB P-Flash（带 ECC）
系统	2 个 16 通道的 DMA 模块 5 个内存保护模块（MPU）

模块	说明
	4 个时钟监测模块（CMU） 1 个错误控制模块（FCU） 1 个电源管理模块（PMU） 1 个系统管理模块（SMU） 1 个实时计数器模块（RTC） 1 个硬件加密模块（DSE），支持 AES、SM2/3/4 和 TRNG 1 个核心中断控制模块（CLINT） 1 个外部中断控制模块（PLIC） 1 个 CRC 校验模块
电源管理	四种电源管理模式：RUN，SRUN，SLEEP，DEEPSLEEP 低电压检测和复位功能（LVD/LVR） 高电压检测功能（HVD）
模拟接口	3 个 12 位的模数转换器（ADC），最多支持 48 通道模拟通路 2 个模拟比较器（ACMP） 2 个 8 位的数模转换器（DAC） 1 个温度传感器
定时器	4 个 32 位高级定时器 4 个 16 位通用定时器
通信接口	6 个 SPI 模块，支持主从模式标准 SPI 协议，速率最高可达 30MHz 4 个 CAN 模块，都支持 CANFD 8 个 USART 模块，都具有 LIN 模式、同步串口模式和 SPI 模式 1 个以太网（MAC）模块，支持 10/100M 模式、全/半双工模式 4 路 IIC 模块，支持主从模式标准 IIC 协议。 8 个通用 IO 模块（GPIO），GPIOH 支持唤醒输入
调试接口	满足 RISC-V Debug Spec 0.13.2 标准的带有 JTAG 接口的调试器

1.3 结构框图

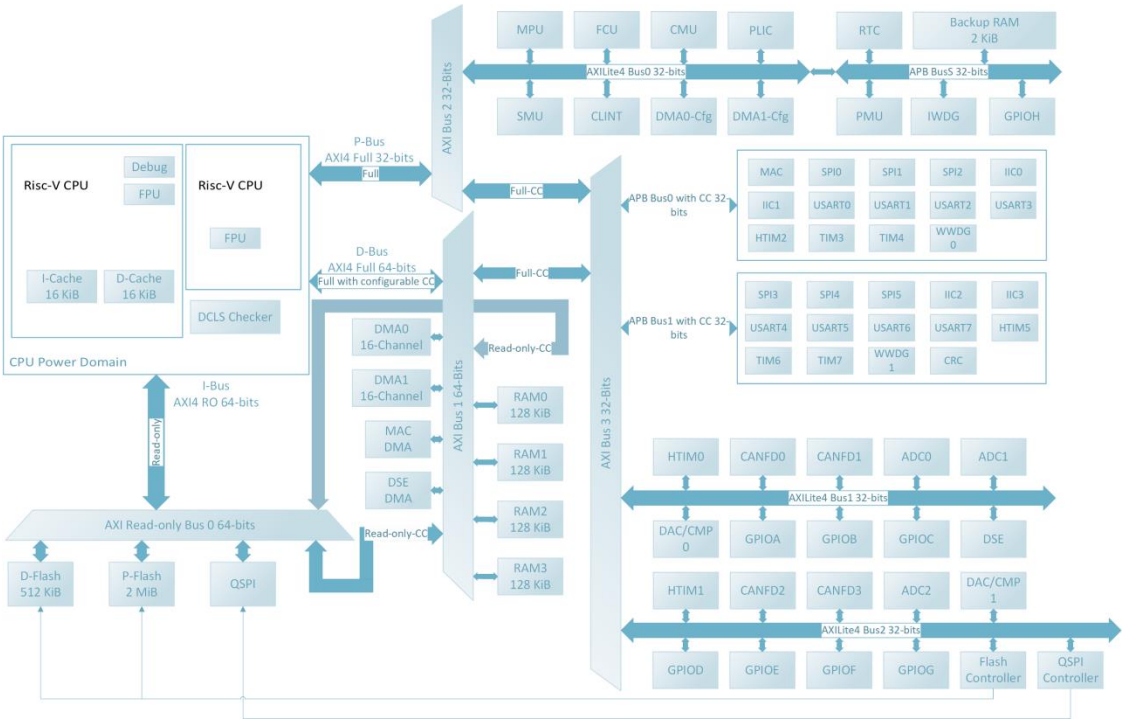


图 1.1 结构框图

1.4 名词缩略表

表 1.2 名词缩略表

缩写	描述
ADC	模拟/数字转换器
CAN	局域网控制器
DAC	数字/模拟转换器
DMA	直接内存存取控制器
EXTI	外部中断事件控制器
FLASH	闪存存储器
GPIO	通用输入输出
IIC/S	双向二线制同步串行总线
WWDG	窗口看门狗

缩写	描述
SPI	串行外设接口
TIM	通用定时器
HTIM	高级定时器
USART	通用同步异步接收发射端

2 地址空间映射

2.1 介绍

该芯片包含各种存储器与内存映射外设，被分配在 32 位连续的内存的空间中，本章描述该内存空间中的内存和外设位置。

2.2 映射表

表 2.1 地址映射表

空间	起始地址	结束地址	空间大小	说明
AXI Bus 0	0x0000_0000	0x1FFF_FFFF	512 MiB	指令总线
	0x0000_0000	0x0000_0FFF	4 KiB	BOOT-ROM
	0x0100_0000	0x011F_FFFF	2 MiB	P-FLASH Main
	0x0120_0000	0x0120_3FFF	16 KiB	P-FLASH Info
	0x0200_0000	0x0207_FFFF	512 KiB	D-FLASH Main
	0x0208_0000	0x0208_0FFF	4 KiB	D-FLASH Info
	0x1000_0000	0x1FFF_FFFF	256 MiB	QSPI-FLASH
AXI Bus 1	0x2000_0000	0x2FFF_2FFF	256 MiB	数据总线
	0x2000_0000	0x2001_FFFF	128KiB	SRAM0
	0x2002_0000	0x2003_FFFF	128KiB	SRAM1
	0x2004_0000	0x2005_FFFF	128KiB	SRAM2
	0x2006_0000	0x2007_FFFF	128KiB	SRAM3
AXi Bus 2	0x3000_0000	0x7FFF_FFFF	1.25 GiB	配置/外设总线
AxiLite4 Bus0	0x3000_0000	0x3FFF_FFFF	256MiB	设置总线
	0x3000_0000	0x3000_03FF	1 KiB	MPU0-IBUS
	0x3000_0400	0x3000_07FF	1 KiB	MPU1-DBUS
	0x3000_0800	0x3000_0BFF	1 KiB	MPU2-PBUS
	0x3000_0C00	0x3000_0FFF	1 KiB	MPU3-DMA0
	0x3000_1000	0x3000_13FF	1 KiB	MPU4-DMA1
	0x3000_2000	0x3000_23FF	1 KiB	FCU

空间	起始地址	结束地址	空间大小	说明
	0x3000_3000	0x3000_33FF	1 KiB	CMU0
	0x3000_3400	0x3000_37FF	1 KiB	CMU1
	0x3000_3800	0x3000_3BFF	1 KiB	CMU2
	0x3000_3C00	0x3000_3FFF	1 KiB	CMU3
	0x3000_4000	0x3000_43FF	1 KiB	DMA0-CFG
	0x3000_4400	0x3000_47FF	1 KiB	DMA1-CFG
	0x3000_6000	0x3000_63FF	1 KiB	SMU
	0x3001_0000	0x3001_FFFF	64 KiB	CLINT
	0x3002_0000	0x3002_FFFF	64 KiB	PLIC
APB Bus S	0x3010_0000	0x301F_FFFF	1 MiB	备份总线
	0x3010_0000	0x3010_07FF	2 KiB	BACKUP-RAM
	0x3010_1000	0x3010_13FF	1 KiB	RTC
	0x3010_2000	0x3010_23FF	1 KiB	GPIOH
	0x3010_3000	0x3010_33FF	1 KiB	PMU
	0x3010_4000	0x3010_43FF	1 KiB	IWDG
AXI Bus 3	0x4000_0000	0x7FFF_FFFF	1 GiB	外设总线
AxiLite4 Bus1	0x4100_0000	0x41FF_FFFF	16 MiB	外设 Lite 总线 1
	0x4100_0000	0x4100_03FF	1 KiB	TIM0
	0x4101_0000	0x4101_FFFF	16 KiB	CANFD0
	0x4102_0000	0x4102_FFFF	16 KiB	CANFD1
	0x4100_1000	0x4100_13FF	1 KiB	ADC0
	0x4100_2000	0x4100_23FF	1 KiB	ADC1
	0x4100_3000	0x4100_33FF	1 KiB	DAC/CMP0
	0x4100_4000	0x4100_43FF	1 KiB	GPIOA
	0x4100_5000	0x4100_53FF	1 KiB	GPIOB
	0x4100_6000	0x4100_63FF	1 KiB	GPIOC
	0x4110_0000	0x414F_FFFF	1 MiB	DSE
AxiLite4 Bus2	0x4200_0000	0x42FF_FFFF	16 MiB	外设 Lite 总线 2

空间	起始地址	结束地址	空间大小	说明
	0x4200_0000	0x4200_03FF	1 KiB	TIM1
	0x4201_0000	0x4201_FFFF	16 KiB	CANFD2
	0x4202_0000	0x4202_FFFF	16 KiB	CANFD3
	0x4200_1000	0x4200_13FF	1 KiB	ADC2
	0x4200_2000	0x4200_23FF	1 KiB	DAC/CMP1
	0x4200_3000	0x4200_33FF	1 KiB	GPIOD
	0x4200_4000	0x4200_43FF	1 KiB	GPIOE
	0x4200_5000	0x4200_53FF	1 KiB	GPIOF
	0x4200_6000	0x4200_63FF	1 KiB	GPIOG
	0x4210_0000	0x4210_03FF	1 KiB	FLASH CTRL
	0x4210_0800	0x4210_0BFF	1 KiB	QSPI CTRL
APBBus0	0x5000_0000	0x50FF_FFFF	16 MiB	外设 APB 总线 0
	0x5000_0000	0x5000_03FF	1 KiB	MAC
	0x5000_1000	0x5000_13FF	1 KiB	SPI0
	0x5000_2000	0x5000_23FF	1 KiB	SPI1
	0x5000_3000	0x5000_33FF	1 KiB	SPI2
	0x5000_4000	0x5000_43FF	1 KiB	IIC/IIS0
	0x5000_5000	0x5000_53FF	1 KiB	IIC/IIS1
	0x5000_6000	0x5000_63FF	1 KiB	USART0
	0x5000_7000	0x5000_73FF	1 KiB	USART1
	0x5000_8000	0x5000_83FF	1 KiB	USART2
	0x5000_9000	0x5000_93FF	1 KiB	USART3
	0x5000_A000	0x5000_A3FF	1 KiB	TIM2
	0x5000_B000	0x5000_B3FF	1 KiB	TIM3
	0x5000_C000	0x5000_C3FF	1 KiB	TIM4
	0x5000_D000	0x5000_D3FF	1 KiB	WWDG0
APBBus1	0x5100_0000	0x51FF_FFFF	16 MiB	外设 APB 总线 1
	0x5100_0000	0x5100_03FF	1 KiB	SPI3
	0x5100_1000	0x5100_13FF	1 KiB	SPI4

空间	起始地址	结束地址	空间大小	说明
	0x5100_2000	0x5100_23FF	1 KiB	SPI5
	0x5100_3000	0x5100_33FF	1 KiB	IIC/IIS2
	0x5100_4000	0x5100_43FF	1 KiB	IIC/IIS3
	0x5100_5000	0x5100_53FF	1 KiB	USART4
	0x5100_6000	0x5100_63FF	1 KiB	USART5
	0x5100_7000	0x5100_73FF	1 KiB	USART6
	0x5100_8000	0x5100_83FF	1 KiB	USART7
	0x5100_9000	0x5100_93FF	1 KiB	TIM5
	0x5100_A000	0x5100_A3FF	1 KiB	TIM6
	0x5100_B000	0x5100_B3FF	1 KiB	TIM7
	0x5100_C000	0x5100_C3FF	1 KiB	WWDG1
	0x5100_D000	0x5100_D3FF	1 KiB	CRC

3 平台软件 IAR For RISC-V 启动文件说明

3.1 背景

本文所介绍的启动文件基于国科环宇科技股份有限公司自研安全 MCU-AS32x 系列 进行设计，提供了一个基础的 RISC-V 架构 MCU 芯片启动流程，描述了芯片从上电到运行 C 语言程序之间的过程，用于深入理解 RISC-V 嵌入式软件开发借鉴，同时本文档所介绍的启动文件流程可直接应用于其他相同架构产品。另外，对于其他架构的芯片，流程基本相似，区别主要在于汇编指令，对应到特定汇编指令即可。

3.2 缩略语和引用文件

3.2.1 缩略语

GP global pointer 全局指针

SP stack pointer 栈指针

SMU System Manage Unit

PLIC Platform Level InterruptController

3.3 概述

MCU 在开发使用过程中，通常开发者只需要完成 C 语言代码功能，即可利用集成开发环境编译出来芯片的可执行文件，但在此过程中，编译器进行了一系列编译操作来保证 MCU 可以支持 C 程序的运行，因此，实际的完整工程代码中，总共包含了三类文件：汇编文件、C 程序代码以及链接脚本。

首先，需要明确概念，程序的编译整体包含预处理、编译、汇编、链接四个步骤。经过这四个过程之后，编译器产生 MCU 可识别的可执行文件，在编译的过程中，上述所提供的汇编指令代码文件同样被编译，同时依照链接脚本所提供的规则，穿插到 C 语言功能代码之间，如图 3-1 所示。

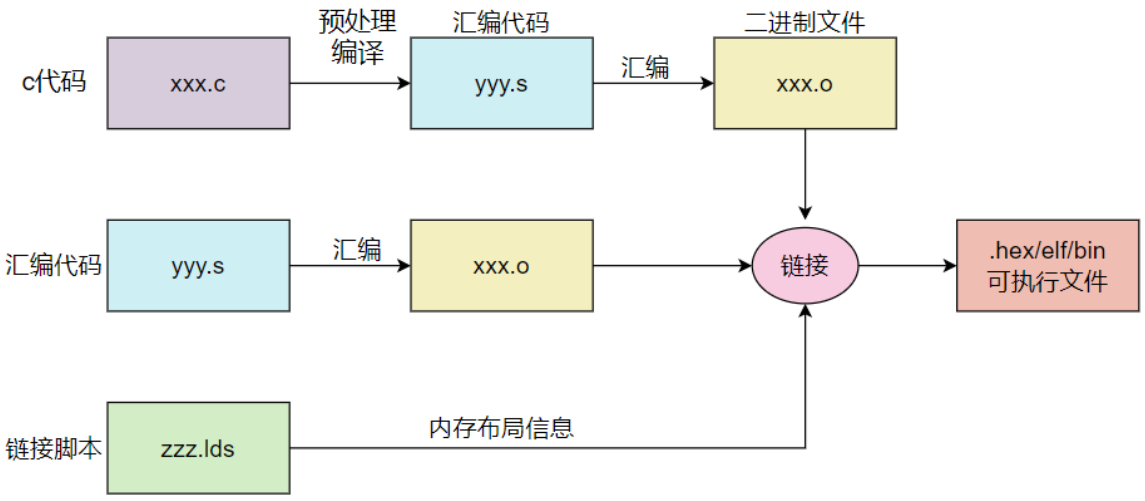


图 3.1 编译流程

3.4 详细描述

本文以 AS32x601 启动文件为例进行具体流程介绍，为了便于理解，将启动流程划分为三个阶段：第一阶段为系统启动必备阶段，此阶段主要完成指针初始化操作，配置全局指针 GP 以及栈指针 SP；第二阶段为数据搬移阶段，此过程主要完成两部分操作，搬移 data 段数据到 RAM，清空 bss 数据段；第三阶段与平台相关，主要完成的中断配置，系统时钟初始化等，此阶段需要根据不同 MCU 设计自行完成，属于非必需过程，流程图参考图 3.2：

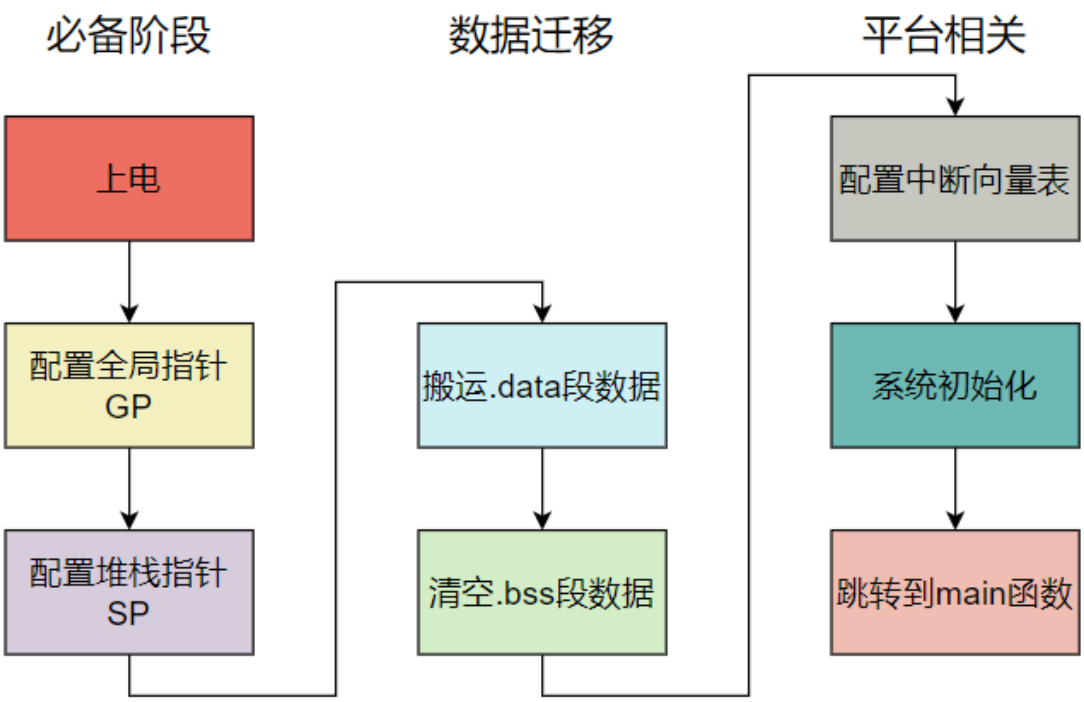

必备阶段
数据迁移
平台相关
上电
配置全局指针 GP
配置堆栈指针 SP
搬运.data段数据
清空.bss段数据
配置中断向量表
系统初始化
跳转到main函数

图 3.2 启动文件流程

```
    SECTION `.init`:CODE        //定义.init内存段
    PUBLIC  _start              //声明 _start 标签为公共符号，这意味着其他文件可以引用它;

    EXTERN   main

    EXTERN RW_DATA$$Limit       //表示可读写数据段（.data）的上限地址;
    EXTERN CSTACK$$Limit        //表示 C 栈的上限地址;
    EXTERN RW_DATA$$Base        //表示可读写数据段（.data）的起始地址;
    EXTERN RW_DATA$$Length      //表示可读写数据段的长度;
    EXTERN RW_DATA_INIT$$Base   //表示初始化数据段的起始地址;
    EXTERN RW_BSS$$Base         //表示未初始化的全局变量和静态变量段（.bss）的起始地址;
    EXTERN RW_BSS$$Length       //表示 .bss 段的长度;
    EXTERN RW_BSS$$Limit        //表示 .bss 段的上限地址;
    EXTERN HEAP$$Base           //表示堆的起始地址;
    EXTERN SystemInit           //系统初始化函数的入口地址;
    EXTERN TrapEntry            //中断和异常处理程序的入口地址
    EXTERN  __iar_static_base$$GPREL    //表示全局指针（GP）的偏移地址;

    ALIGN  2

_start:
        .option   norvc;
        la gp, __iar_static_base$$GPREL
        la sp, CSTACK$$Limit

    /* Load data section */
    la a0, RW_DATA_INIT$$Base
    la a1, RW_DATA$$Base
    la a2, RW_DATA$$Limit
    bgeu a1, a2, ClearBssSection
step1:
    lw t0, 0(a0)
    sw t0, 0(a1)
    addi a0, a0, 4
    addi a1, a1, 4
    bltu a1, a2, step1

ClearBssSection:
    /* Clear bss section */
    la a0, RW_BSS$$Base
    la a1, RW_BSS$$Limit
    bgeu a0, a1, step3
step2:
    sw zero, 0(a0)
    addi a0, a0, 4
    bltu a0, a1, step2
step3:
    /* Enable the CPU global interrupt,[3]MIE = 1, [7]MPIE = 1, [11\12]MPP = 11 ,[13\14]FS = 01 */
    lui a5,0x4
    addi a5,a5,-1912 //3888
    csrw mstatus,a5

    /* [11]MEIE enable external interrupt PLIC */
    lui a5,0x1
    addi a5,a5,-2048 //800
    csrw mie,a5

    /* Set the exception entry function */
    la t0, TrapEntry
    csrw mtvec, t0

    call SystemInit
    call main


loop:
        j loop

        END
```

图 3.3 启动代码清单

首先，利用 SECTION `.init`:CODE 指令定义.init 段，声明以下内容均包含在 init 段中，接下来定义全局标签_start，声明此全局标签后，链接文件中即可分

配.init 段地址来指定_start 为处理器上电执行的第一个操作，经过以上两个步骤，即可保证此部分内容先于所有指令，从而完成上电配置。

1. 配置全局指针 GP

第 24 行，通过 la 命令，将__iar_static_base$$GPREL 写入寄存器 gp。

其中，__iar_static_base$$GPREL 是由 IAR 提供的全局指针初始值，编译器会合理分配。此处需要注意，通常情况下，gp 指针定义在 data 区，有时候为了优化代码密度，可以根据实际情况修改 gp 指针的位置，如工程中定义了大量的初始化为 0 或未初始化的全局数组作为缓冲区，可以将 gp 指针的位置定义到 bss 段。

2. 配置栈指针 SP

第 25 行，通过 la 命令，将 CSTACK$$Limit 写入寄存器 sp。

其中，_sp 是由链接脚本提供的栈指针位置，开发者应根据数据存储器大小和程序调用层次合理分配，栈指针必须保持 4 字节对齐，否则会发生加载/存储对齐错误。

3. 加载 data 段

第 27-37 行，将 data 段从程序存储器搬运至数据存储器，作为可读可写的变量。

第 28-30 行，向寄存器 a0、a1、a2 加载必要数据，地址由链接脚本生成：

寄存器	加载值	功能
a0	程序存储器的 data 段起始地址	读指针
a1	数据存储器的 data 段起始地址	写指针
a2	数据存储器的 data 段结束地址	结束搬运的地址

第 31 行，如果 a1 大于等于 a2，表示没有需要搬运的数据，跳过以下循环，执行下面的工作。

第 32-37 行，是一个循环结构。读出 a0 指向的地址，数据写入 t0 暂存。t0 的数据写入 a1 指向的地址。a0+4，指向下一存储单元，因为 32bit 是 4 字节。a1+4，指向下一存储单元，因为 32bit 是 4 字节。如果 a1 小于 a2，表示未搬运完，跳转至 32 行，进入下一次循环。如果 a1 等于 a2，表示已经搬完了最后一个数据，退出循环，执行下面的工作。

需要说明的是，程序正常运行，还需要链接脚本的配合，对于启动文件来说，链接脚本除定义上电入口函数外，还对数据搬移过程产生影响，对于自定义的启动地址_start,则需要在链接文件中定义 initialize manually，才可以在启动文件中实现数据的搬运。

4. 清空 bss 段

第 39-47 行，工作与 data 段有点类似，但是只需要清空指定位置的数据。

第 41-42 行，向寄存器 a0、a1 加载必要数据，地址由链接脚本生成：

寄存器	加载值	功能
a0	bss 段起始地址	指针
a1	bss 段结束地址	结束清空的地址

第 43 行，如果 a0 大于等于 a1，表示没有需要清 0 的空间，跳过以下循环，执行下面的工作。

第 44-47 行，是一个循环结构。清空 a0 指向的地址 a0+4，指向 下一存储单元，因为 32bit 是 4 字节。如果 a0 小于 a1，表示清 0 未 结束，跳转至 32 行，进入下一次循环。如果 a0 等于 a1，表示已经清 0 了最后一个存储单元，退出循环，执行下面的工作。

5. 中断初始化

从此处开始，启动文件进入流程第三阶段；

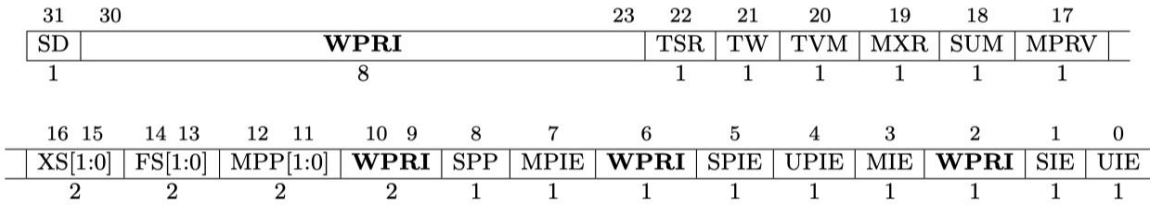


图 3.4 mstatus 寄存器

mstatus:Machine Status Register，RISC-V 架构中的一个重要控制和状态寄存器，管理和反映机器模式下的状态和控制信息。

mstatus.MIE：Machine Interrupt Enable，机器模式全局中断使能位

mstatus.MPIE：Machine Previous Interrupt Enable，机器模式先前中断使能位

mstatus.MPP：Machine Previous Privilege，机器模式之前的特权级别

mstatus.SPP：Supervisor Previous Privilege，超级模式之前的特权级别

第 49-52 行，设置 mstatus 寄存器，开启 CPU 全局中断；

15	12	11	10	9	8	7	6	5	4	3	2	1	0
0		MEIE	0	SEIE	0	MTIE	0	STIE	0	MSIE	0	SSIE	0
4		1	1	1	1	1	1	1	1	1	1	1	1

图 3.5 MIE 寄存器

MEIE：M 模式外部中断使能位

SEIE：S 模式外部中断使能位

MTIE：M 模式 timer 中断使能位

STIE：S 模式 timer 中断使能位

MSIE：M 模式软中断使能位

SSIE：S 模式软中断使能位

第 54-57 行，配置 MIE 寄存器，开启外部中断；

MXLEN-1	2	1	0
BASE[MXLEN-1:2] (**WARL**)		MODE (**WARL**)	
MXLEN-2		2	

Value	Name	Description
0	Direct	All exceptions set pc to BASE.
1	Vectored	Asynchronous interrupts set pc to BASE+4×cause.
≥2	—	*Reserved*

图 3.6 mtvec 寄存器

Direct 模式：所有的中断和异常使用同一个中断入口地址，一般都会设置为这种模式。

Vectored 模式：所有异常使用同一个入口地址，但是不同的中断使用不同的入口地址。

第 59-61 行，使用 Direct 模式，设置中断向量表入口地址 TrapEntry，再由 as32x601_trapentry.S 找到 PLIC_TrapHandler。

```
190   */
191   TrapEntry:
192       /* save context and the value of mepc register,
193       SAVE_CONTEXT
194
195       /* Call the interrupt handler */
196       call PLIC_TrapHandler
197
198       /* Restore mepc values and restore context */
199       RESTORE_CONTEXT
200
```

图 3.7 TrapEntry 代码定义

6. 系统初始化

系统初始化主要用来初始化系统时钟等一些必要的配置模块，一般芯片都会进行此部分功能模块的设计，但实际上，此部分功能为平台相关，说明此处可更改，不仅可以更改适配不同平台，同样可删除此部分代码。

第 63 行，调用 SystemInit 函数完成系统时钟初始化，此函数利用 C 语言编写，位于 as32x601_core.c 文件中，所实现的主要功能为初始化系统时钟，本芯片时钟模块命名为 SMU，设计功能包含 PLL 时钟倍频分频管理、加固以及双核锁步、管理系统复位等功能。

```
29  void SystemInit(void)
30  {
31      SMU_PLLInitTypeDef SMU_PLLInitStruct;
32      SMU_ClockInitTypeDef SMU_ClockInitStruct;
33
34      /* Bus clock configure */
35      SYS_BusClkInit();
36
37      /* Dual core lockstep enable */
38  #if DCLS == 1
39      SMU_DCLSCmd(ENABLE);
40  #endif
41
42      /* PLL clock configure */
43      SMU_PLLStructInit(&SMU_PLLInitStruct);
44      SMU_PLLInitStruct.OscillatorType = SMU_OSCILLATORTYPE_FIRC; // Select FIRC as the PLL clock
45      SMU_PLLInitStruct.FIRCOscState = ENABLE;                    // Enable FIRC
46      SMU_PLLInitStruct.FIRCCalibrationValue = 0x00;              // FIRC oscillator frequency adjustment
47      SMU_PLLInitStruct.PLLConfig.PLLState = ENABLE;              // Enable PLL
48      SMU_PLLInitStruct.PLLConfig.PLLSource = SMU_PLLCLK_FIRC;    // FIRC serves as the PLL clock
49      SMU_PLLInitStruct.PLLConfig.PLLDivR = 0x08;                 // For CAN Freq 40MHz
50      SMU_PLLInitStruct.PLLConfig.PLLDivQ = 0x02;                 // For System Clock 160MHz
51      SMU_PLLInitStruct.PLLConfig.PLLDivN = 0x10;                 // The clock after input frequency division is 1MHz
52      SMU_PLLInitStruct.PLLConfig.PLLDivF = 0x140;                // The clock output by VCO is 320MHz
53      SMU_PLLInit(&SMU_PLLInitStruct);
54
55      EFLASH_CLK_ENABLE();               // Enable EFlash clock
56      EFLASH_SET();                      // Release EFlash reset
57
58      /* EFlash unlock */
59      FLASH_UnlockCtrl();
60
61      /* Configure EFlash clock frequency */
62      FLASH_SetCLKFreq(160);
63
64      /* Bus clock configure */
65      SMU_ClockStructInit(&SMU_ClockInitStruct);
66      SMU_ClockInitStruct.SYSCLKSelect = SMU_SYSCLK_PLL;          // Select PLL as the SYSCLK
67      SMU_ClockInitStruct.AXI4Bus3CLKDiv = AXI4Bus3CLKDiv1;       // AXI4Bus3 Clock Division 1
68      SMU_ClockInitStruct.APBBus0CLKDiv = APBBus0CLKDiv1;         // APBBus0 Clock Division 1
69      SMU_ClockInitStruct.APBBus1CLKDiv = APBBus1CLKDiv1;         // APBBus1 Clock Division 1
70      SMU_ClockInitStruct.CANX2CLKDiv = CANX2CLKDiv1;             // CANX2 Clock Division 1
71      SMU_ClockInit(&SMU_ClockInitStruct);
72
73      EFLASH_CLK_UPDATE_ENABLE();        // Enable EFlash clock update
74      EFLASH_CLK_UPDATE_DISABLE();       // Disable EFlash clock update
75
76      CLINT_CLK_ENABLE();                // Enable CLINT clock
77      CLINT_SET();                       // Set CLINT
78  }
```

图 3.5 SystemInit 代码清单

至此，启动文件完成所有上电配置流程，接下来，第 64 行，通过调用 main 函数跳转到用户代码中开始执行后续操作。

7. 链接配置文件说明

AS32x601 芯片提供了最大 2MB 的 P-Flash 和 512kb 的 D-flash 及 512kb 的 SRAM。链接配置文件（.icf）作用等同于 GCC 工具链下的链接文件（.ld）,用来描述系统内存分区，定义内存大小，分配数据段存储位置，设置堆栈等，以本芯片为例，编写如下：

as32x601_trapentry.S | as32x601_plic.c | AS32I601.icf | AS32I601.menu | as32x601_

D: > 0Common Files > IAR普通版_激活使用指南 > IAR设备支持包 > 设备支持包 > AS32I601.icf

```
1  ////////////////////////////////////////////////////////////////////////////////
2  //
3  // IAR ILINK Linker configuration file for the AS32I601
4  //
5  //
6
7  build for rom;
8
9  define exported symbol _link_file_version_2 = 1;
10 keep symbol __iar_cstart_init_gp; // defined in cstartup.s
11
12 define memory mem with size = 4G;
13
14 define region ROM_region32 = mem:[from 0x10000000 size 32M];
15 define region RAM_region32 = mem:[from 0x20000000 size  32k];
16
17 initialize manually { section .data };
18 do not initialize  { section *.noinit };
19
20 define block CSTACK with alignment = 16, size = CSTACK_SIZE { };
21 define block HEAP   with alignment = 16, size = HEAP_SIZE   { };
22
23 define block RW_DATA { rw section .data};
24
25 define block RW_DATA_INIT { ro section .data_init};
26
27 define block RW_BSS {rw section .bss};
28
29 define block RW_DATA_ALL with static base GPREL { block RW_DATA, block RW_BSS };
30
31 "STARTUP" : place at start of ROM_region32 {ro section .init};
32
33 "ROM32":place in ROM_region32        { ro,
34                                        block RW_DATA_INIT };
35
36 "RAM32":place in RAM_region32        { block RW_DATA_ALL,
37                                        block HEAP,
38                                        block CSTACK };
39
```

图

第 10 行，保留符号__iar_cstart_init_gp，该符号在启动代码 cstartup.s 定义，保留符号意味着即使它未被直接引用，也不会被链接器优化掉。

第 12 行，定义总体内存空间，包含数据存储区，程序存储区，最大为 4G，可直接设置为最大，不影响实际效果；

第 14-15 行，定义 RAM、ROM 区间地址以及大小；

第 17 行，手动初始化.data 段。这意味着.data 段的内容需要在程序启动时从其他位置（如 ROM）复制到 RAM 中；

第 18 行，不对以.noinit 结尾的段进行初始化。这些段的内容在程序启动时保持不变；

第 20-21 行，定义堆栈大小以及对齐方式，此处 CSTACK_SIZE 和 HEAP_SIZE 代表此处大小由 iar 软件获取，也可直接进行固定设置，单位为字节；

第 23-29 行，定义了 RW_DATA、RW_DATA_INIT、RW_BSS 和 RW_DATA_ALL，可供启动文件调用；

第 30 行，将只读的初始化代码段.init 放置在 ROM_region32 的起始位置。这个段包含启动代码，负责初始化程序的运行环境；

第 30 行，将只读段（包括代码和常量数据）以及 RW_DATA_INIT 块放置在 ROM_region32 中。这意味着初始化数据将与代码一起存储在 ROM 中。

第 31 行，将 RW_DATA_ALL 块（包含.data 和.bss 段）、HEAP 块和 CSTACK 块放置在 RAM_region32 中。这确保了可读写数据、堆和栈在 RAM 中进行操作，以实现快速访问和修改。

4 内核与系统

4.1 内核

4.1.1 简介

带有硬件 FPU 的 E7 内核是自行研发的最新一代的嵌入式处理器。它的设计目标是低成本、低功耗、高实时性、高安全性，在这基础之上同时提供优秀的运算性能。

E7 处理器是一款高效的高性能处理器：

- 8 级双发射流水线
- 动态分支预测
- 哈弗架构的缓存（16 KiB 的 I-cache 和 16KiB 的 D-cache）
- 64 位 AXI4 总线接口 处理器支持以下的内存接口：
- 指令 AXI 接口（AXII）
- 数据 AXI 接口（AXID）
- 低延迟外设 AXI 接口（AXIP） 它内建的双浮点 FPU（浮点单元）可以加速浮点相关的软件的运行。

4.1.2 调试

E7 支持标准的 JTAG 调试，包含程序的运行控制与跟踪。它包含符合 RISC-V DEBUG 标准的标准 JTAG 调试端口。具体实现请参考调试。

4.2 总线

4.2.1 简介

MCU 的总线的 AXI Crossbar 是一个总线矩阵，用于互联 CORE 内核与系统存储器及外设模块的访问（读写）。

- Crossbar 的主机可以主动发起数据访问请求，而从机则只能被动接受访问；
- Crossbar 的每个主机/从机与总线之间都有 ECC 编解码模块；

- 每个端口都有独立的控制总线、地址总线和数据总线；
- 不同的主机可以同时访问不同的从机，从而保证 MCU 系统工作带宽；
- 主机对任意从机的保护均受 MPU 的保护；

4.2.2 互联矩阵

除 AXI Bus 1 外所有的总线皆为全互联总线。

AXI Bus 1 的总线互联图如下。

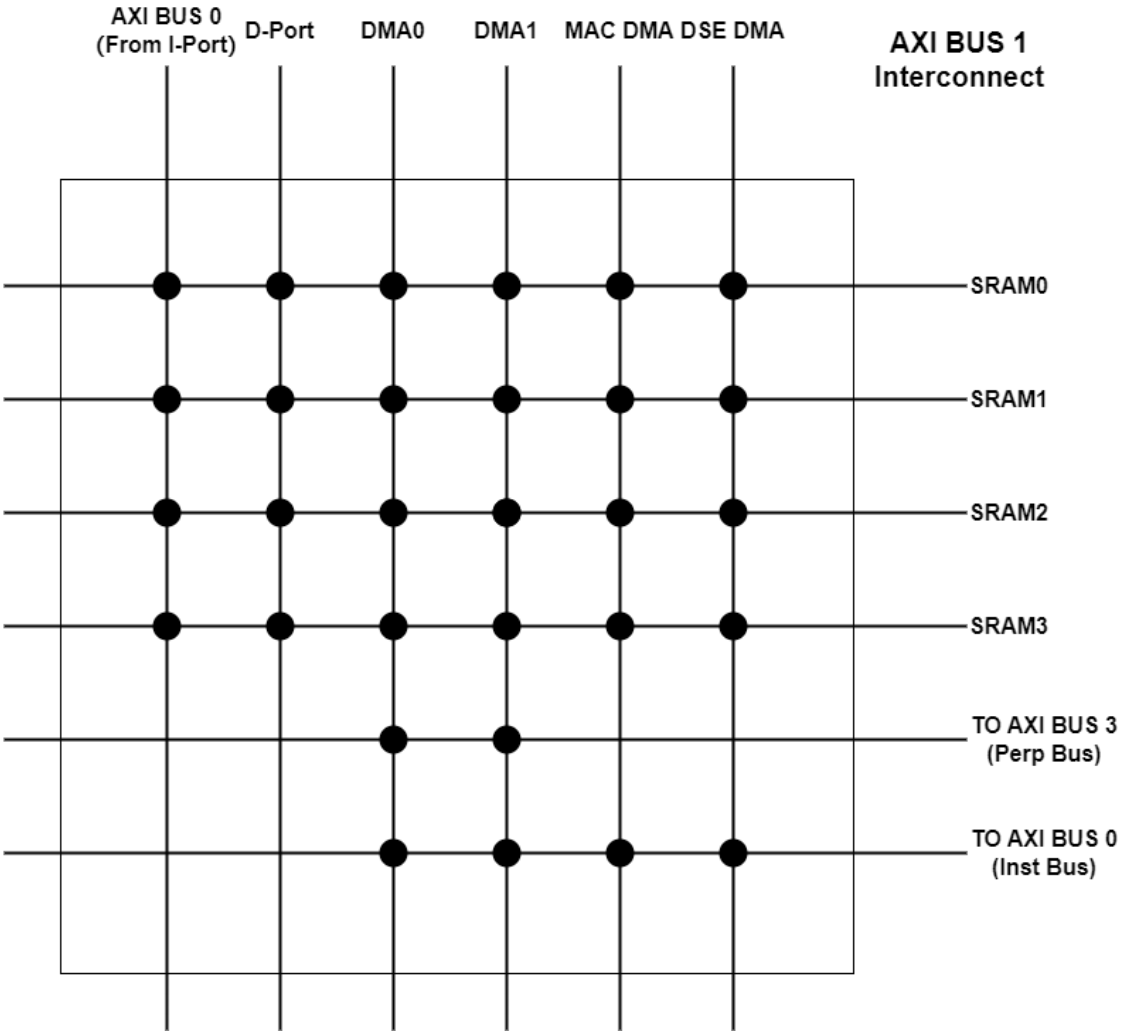


图 4.1 AXI Bus 1 总线互联图

4.3 时钟与电源

4.3.1 时钟

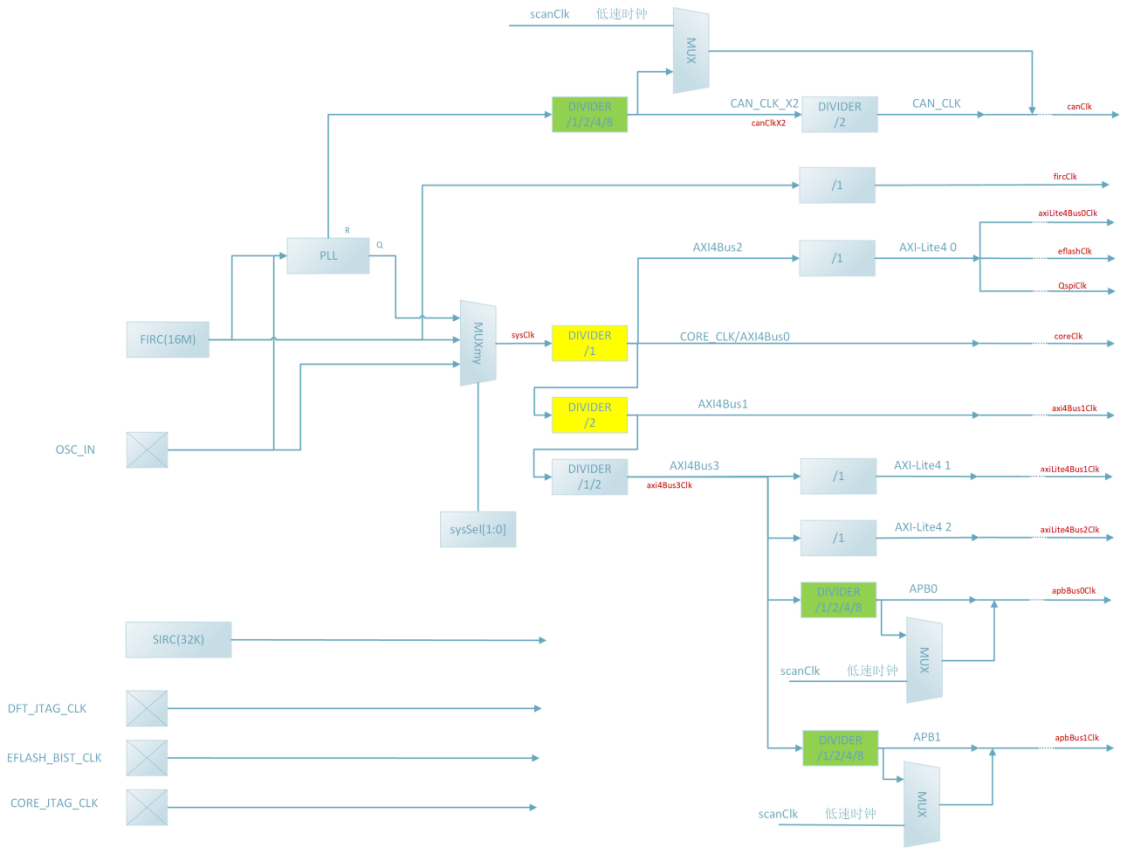


图 4.2 时钟树

系统中的时钟来源有 6 处：

1. 外部晶振：由专用引脚外接晶振提供。晶振的范围在 8-40MHz;
2. FIRC：内部振荡器，其输出频率为 16MHz，误差精度在 1%左右;
3. SIRC：内部振荡器，其输出频率为 32KHz;
4. DFT_JTAG_TCK：由 GPIO 引脚输入。用 DFT 的 SACN 和 BSD;
5. CORE_JTAG_TCK：也是由 GPIO 引脚输入。该时钟用于控制内核模块的 JTAG 状态机;
6. EFLASH_BITS_CLK：也是由 GPIO 引脚输入。

4.3.2 电源

为了有效的控制 MCU 的功耗，对模块的划分包括 AON（Always On）模块、Main 模块和 ANA 模块。

AON 模块：其供电有 RTC 模拟 IP 提供 1.2V 的电源，并提供工作的 32KHz 时钟。该模块会控制 Main 和 ANA 模块的电源开关。在深度睡眠模式下，AON 会关闭 Main 和 ANA 的电源。

Main 模块：该模块包含所有的内核逻辑和外设逻辑。其电源有 PMB 提供。

ANA 模块：该模块包含 ADC、DAC、PLL、ROSC16M、EFLASH 等模拟 IP 模块。

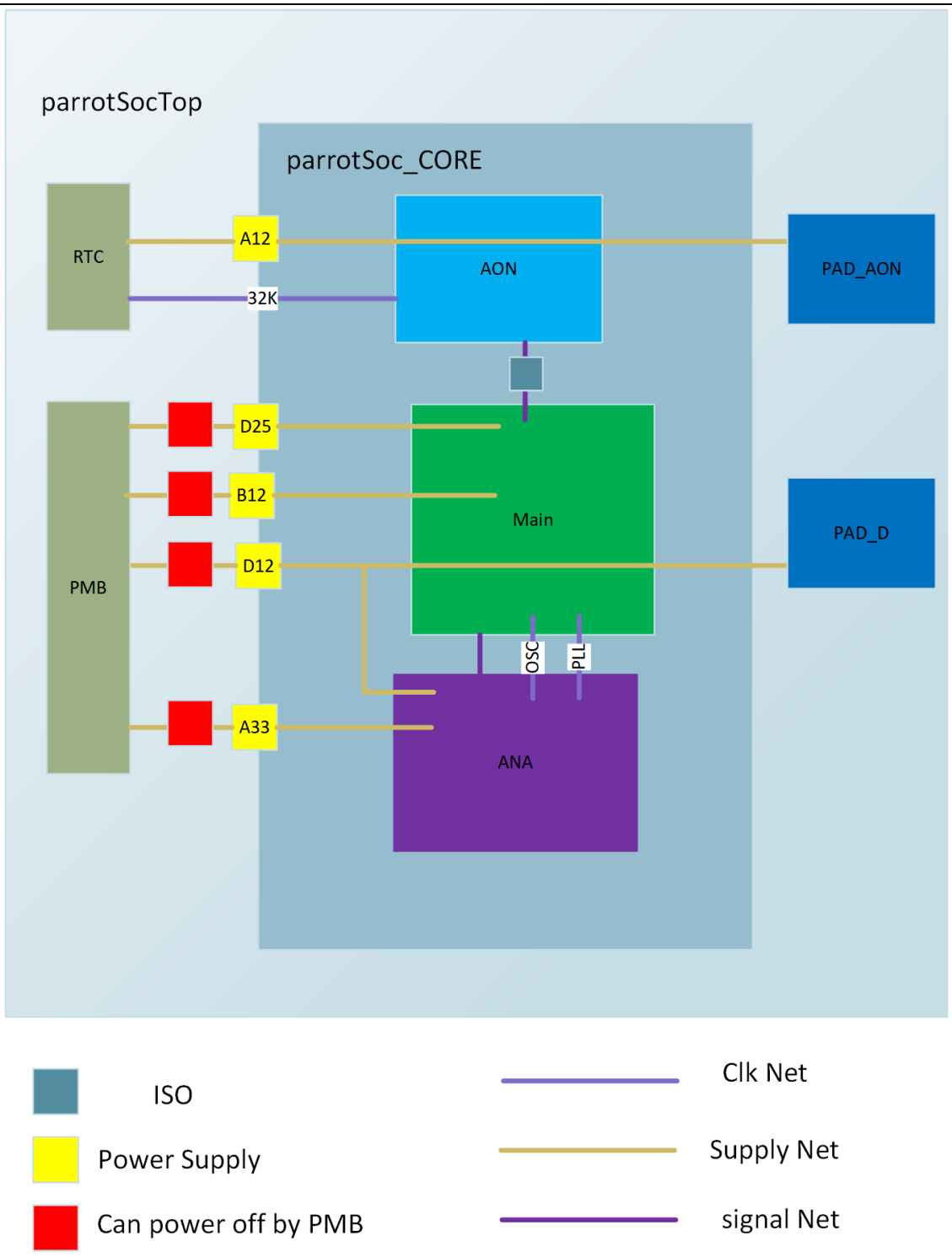

parrotSocTop
parrotSoc_CORE
RTC
A12
32K
AON
PAD_AON
PMB
D25
B12
D12
Main
PAD_D
OSC
PLL
A33
ANA
ISO
Power Supply
Can power off by PMB
Clk Net
Supply Net
signal Net

图 4.3 电源模块

4.4 工作模式

4.4.1 上电启动流程

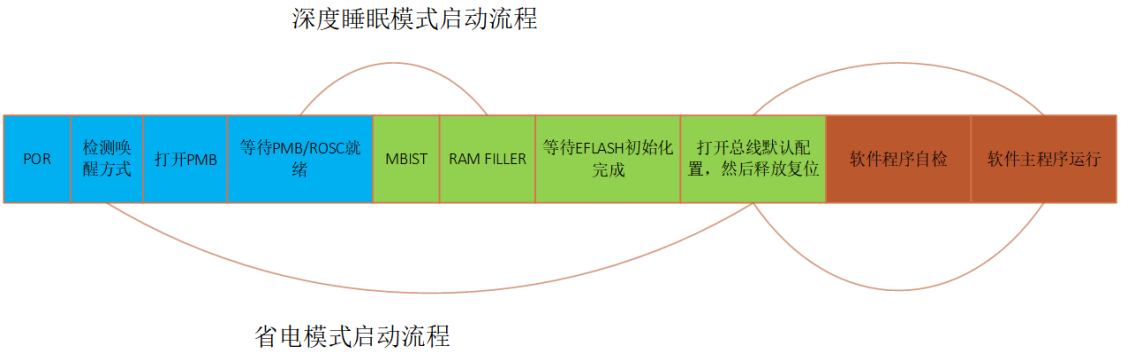


图 4.4 上电流程

4.4.2 电源管理

系统的电源管理状态有 4 种，分别是正常模式、省电模式、睡眠模式和深度睡眠模式。其四种状态之间的转换如下图所示：

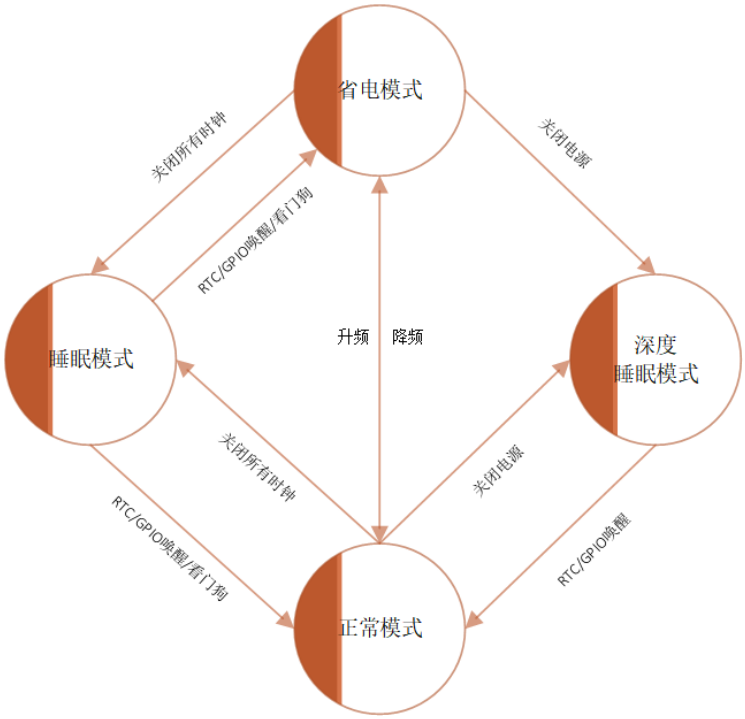


图 4.5 工作模式

在省电模式下，系统低频时钟运行（关闭 PLL，系统的工作频率由 OSC 或 FIRC 提供），而且可以关闭部分外设。用户可以在正常模式或睡眠模式下进入省

电模式。在睡眠模式下，通过 RTC/GPIO 唤醒进入到省电模式时，不需要 CORE 的干预。具体实现请参考系统管理模块。

用户也可以在省电模式下通过关闭所有时钟或关闭通常电源进入睡眠模式和深度睡眠模式。具体实现请参考电源管理模块.

低功耗的粗略运作流程如下：

1. 清空所有外设交互：CORE 需要确保所有外设不会影响到与其通信的其他芯片。也就是说外设模块不存在待发送的数据和已接收但未处理的数据。
2. 等待总线空闲：CORE 需要读取 SMU 中总线的状态信息，确保总线处理空闲状态。
3. 切换系统时钟：CORE 需要配置 SMU 的系统时钟选择模块。在正常模式下进入省电模式时：选择 FIRC 或 OSC 作为系统时钟，同时总线的分频改成 1。在省电模式进入正常模式时使用 PLL 输出的时钟作为系统时钟。在 CORE 配置完成后，SMU 模块会等待 128 个系统时钟周期，然后切换修改后的频率。
4. 保存唤醒信息：在进入深度睡眠模式时，根据用户需求，CORE 可以保存部分用户信息于 Backup RAM 中，当唤醒后可进行读取。
5. 关闭电源：在进入深度睡眠时，CORE 会通知 PMU 关闭 PMB 的电源。
6. RTC/IWDG/GPIO 唤醒：用户可以配置 RTC/IWDG，在指定时间后唤醒 CORE。

在不同模式下各个模块的工作状态如下表所示：

表 4.1 各个模块在不同模式下的工作说明表

模块	正常模式	省电模式	睡眠模式	深度睡眠模式
CORE0/1	全速运行	16M 运行	停止	掉电
AXI4BUS0/1/2/3	全速运行	16M 运行	停止	掉电
AXI4LiteBUS0/1/2	全速运行	16M 运行	停止	掉电
APB0/1	全速运行	16M 运行	停止	掉电
外设接口	全速运行	16M 运行	停止	掉电

模块	正常模式	省电模式	睡眠模式	深度睡眠模式
SMU	16M 运行	16M 运行	停止	掉电
AON	32K 运行	32K 运行	32K 运行	32K 运行
PMB	正常运行	正常运行	低功耗	掉电
PLL	正常运行	低功耗	低功耗	掉电
FIRC	正常运行	正常运行	低功耗	掉电
SIRC	正常运行	正常运行	低功耗	掉电
ADC0/1/2	正常运行	正常运行	低功耗	掉电
DAC0/1	正常运行	正常运行	低功耗	掉电

Note：

全速运行：数字逻辑运行在 PLL 输出的 CLK_Q 时钟（最高运行 180MHz）

16M 运行：数字逻辑运行在 FIRC 时钟（主频为 16MHz）

4.4.3 时钟管理

为了保证系统稳定运行，只有在工作模式切换时才会更新系统时钟配置。外设时钟的关闭和使能用户可以随时切换。其中 CORE、AxiBus0/1/2 和 AxiLiteBus0 的时钟有硬件控制，其他外设时钟由用户配置寄存器控制。

具体实现请参考系统管理模块。

4.4.4 复位管理

系统的复位由硬件复位和寄存器复位。复位的时钟结构如下图所示：

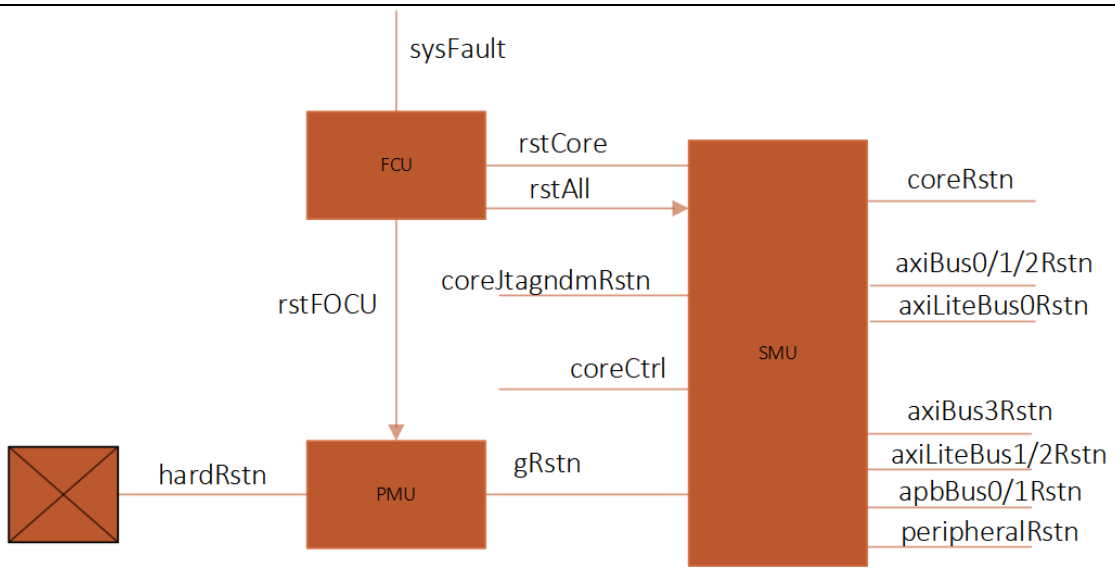


图 4.6 系统复位结构示意框图

具体实现请参考系统管理模块。

4.4.5 启动方式

AS32A601 的启动方式有 4 种。根据 BOOT 引脚确定启动方式。

表 4.2 BOOT 启动方式列表

BOOTPIN	启动方式
2’b00	P-Flash
2’b01	D-Flash
2’b10	QSPI
2’b11	SRAM

5 EFlash 控制器（EFLASH）

5.1 简介

在 AS32A601 中，片内 Flash 共包含两个存储器，分别为程序存储器（PFlash）和数据存储器（DFlash）。

Flash 控制器作为 CPU 内核与片内 Flash 之间的桥梁，可用于管理任何主设备对片内 Flash 进行的访问。Flash 控制器可对 Flash 执行读取、编程和擦除操作，并实施写保护和读保护等安全机制。

5.2 特性

- EFlash 存储器
  - PFlash 最大支持 2MB（包括 4 个 block，即 4×512KB）
  - DFlash 最大支持 512KB（包括 1 个 block）
  - 寿命：≥100,000 周期
  - 块(Block)容量:512KB/block
  - 扇区(Sector)容量：4KB/sector
  - 行(Row)容量：512B/row
- EFlash 控制器
  - 操作列表：
    - 擦除：扇区擦除、块擦除、写保护信息区擦除、全片擦除
    - 编程：行编程，编程最小单位 64-bit
    - 读：支持 8-bit/16-bit/32-bit/64-bit 宽度读数据
  - 安全措施：
    - 支持写保护功能
    - 支持读保护功能
    - 可通过安全密钥临时解锁读保护功能
- EFlash MBIST
  - 支持 EFlash MBIST 测试

- 支持 IEEE1149.7 接口
- 支持自动修复功能

5.3 结构框图

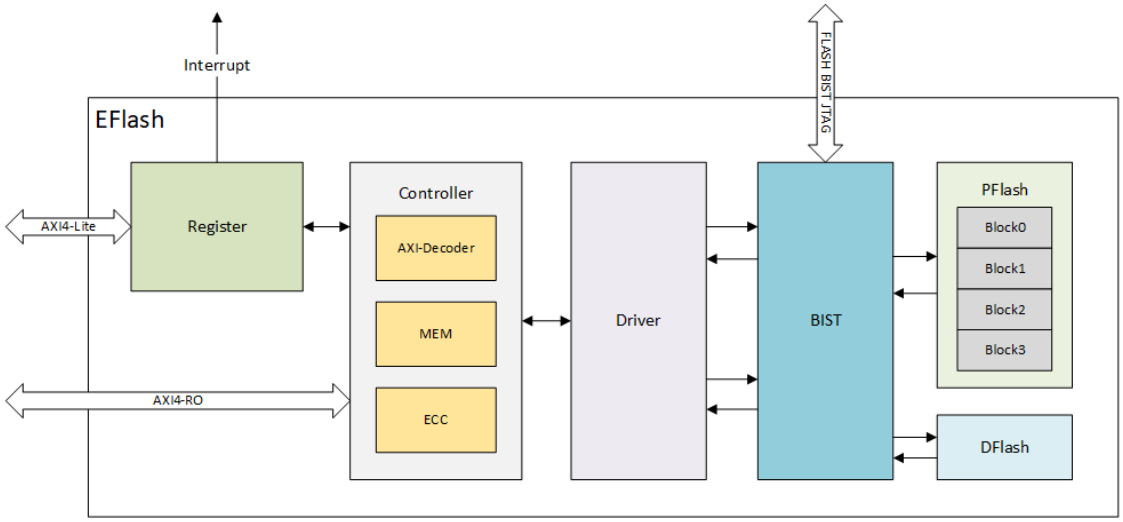


图 5.1 EFlash 结构框图

5.4 功能描述

5.4.1 EFlash 存储空间

5.4.1.1 EFlash 存储空间

表 5.1 EFlash 存储器组织结构表

存储区		空间	地址范围	扇区	起始地址	结束地址	大小
PFlash	主存储区	2MB	0x0100_0000~0x011F_FFFF	sector0	0x0100_0000	0x0100_0FFF	4KB
				sector1	0x0100_1000	0x0100_1FFF	4KB
				sector2	0x0100_2000	0x0100_2FFF	4KB
				…			
				sector511	0x011F_F000	0x011F_FFFF	4KB
	信息区	16KB		info sector0	0x0120_0000	0x0120_0FFF	4KB

存储区		空间	地址范围	扇区	起始地址	结束地址	大小
			0x0120_0000~0x0120_3FFF	info sector1	0x0120_1000	0x0120_1FFF	4KB
				info sector2	0x0120_2000	0x0120_2FFF	4KB
				info sector3	0x0120_3000	0x0120_3FFF	4KB
DFlash	主存储区	512KB	0x0200_0000~0x0207_FFFF	sector512	0x0200_0000	0x0200_0FFF	4KB
				sector513	0x0200_1000	0x0200_1FFF	4KB
				sector514	0x0200_2000	0x0200_2FFF	4KB
				…			
				sector639	0x0207_F000	0x0207_FFFF	4KB
	信息区	4KB	0x0208_0000~0x0208_0FFF	info sector4	0x0208_0000	0x0208_0FFF	4KB

5.4.1.2 EFlash 信息区组织

表 5.2 EFlash 信息区组织结构表

信息区	地址	位	大小 (Byte)	信息
info sector0	0x0120_0000	[31:0]	4	PFlash 写保护信息
	0x0120_0004	[7:0]	1	DFlash 写保护信息
info sector1	0x0120_1000	[1:0]	1	读保护使能位 11b：无读保护（默认） 其他：有读保护
		[3:2]		Security Key 使能位

信息区	地址	位	大小 (Byte)	信息
				11b：失能（默认） 其他：使能
	0x0120_1008	[63:0]	8	Security Key

5.4.2 EFlash 命令

表 5.3 EFlash 命令表

操作类型	命令	ID	描述
擦除	扇区擦除	0x01	按扇区擦除 PFlash 或 DFlash 的主存储区
	块擦除	0x02	擦除指定 Block 的主存储区
	写保护信息扇区擦除	0x03	擦除写保护信息所在扇区
	全片擦除	0x04	擦除 PFlash 的 DFlash 全部区域
编程	行编程	0x10	按照 64-bit 对齐的方式编程 PFlash 或 DFlash，一次最大可编程一行
验证	Security Key 验证	0x20	根据指定的安全秘钥验证是否和存储在信息区的秘钥匹配

5.4.3 中断说明

EFlash 共支持三个中断向量。

表 5.4 EFlash 中断列表

中断源	标志	使能位
EFlash 命令完成中断	EFLASH_STATE[FINISH]	EFLASH_CNFG[FINIE]
EFlash 操作错误中断	EFLASH_STATE[OPERR]	EFLASH_CNFG[OPEIE]
ECC 2-bit 错误中断	EFLASH_STATE[ECC2ERR]	EFLASH_CNFG[ECC2EIE]

5.4.3.1 命令完成中断

当 EFlash 命令执行完成时，命令完成标志 FINISH 置 1，若此时对应的中断使能位 FINIE 为 1，则会产生相应的中断。

中断使用方式如下：

- 判断 EFLASH_STATE 状态寄存器的 FINISH 位的状态，若 FINISH=1，则由软件写 1 将该位清零；
- 配置 EFLASH_CNFG 配置寄存器的 FINIE=1,使能对应的中断；
- 配置 EFLASH_START 命令触发寄存器的 START=1，触发执行 Flash 命令；
- 在命令完成中断处理程序中，清除中断标志位（FINISH 位）；
- 若想要禁用命令完成中断，则配置 FINIE=0。

5.4.3.2 操作错误中断

当 EFlash 命令触发执行时（即配置 EFLASH_START 寄存器的 START=1 后），控制器会对 Flash 命令、地址、数据等信息进行检查，当发生以下任意一种情况时，会将 EFLASH_STATE 状态寄存器的 OPERR 位置 1，若此时对应的中断使能位（即 EFLASH_CNFG 寄存器的 OPEIE 位）为 1，则产生响应的中断。

- EFlash 命令 ID 配置错误：
  - EFlash 配置的命令 ID 与命令 ID 表不匹配
- EFlash 地址配置错误：
  - EFLash 地址超出实际地址范围
  - EFlash 地址未对齐
  - EFlash 编程区域超出 Row 范围
- EFlash 数据错误：
  - EFlash 写入的数据量与配置的数据长度不符

中断使用方式如下：

- 判断 EFLASH_STATE 状态寄存器的 OPERR 位的状态，若 OPERR=1，则由软件写 1 将该位清零；
- 配置 EFLASH_CNFG 配置寄存器的 OPEIE=1,使能对应的中断；

- 配置 EFLASH_START 命令触发寄存器的 START=1，触发执行 Flash 命令；
- 在操作错误中断处理程序中，清除中断标志位（OPERR 位）；
- 若想要禁用操作错误中断，则配置 OPEIE=0。

5.4.3.3 ECC 2-bit 错误中断

ECC 中断相关描述请参考应用说明 ECC 章节。

5.5 应用说明

为了防止用户误操作 EFlash，配置 EFlash 寄存器前首先需要解锁，否则无法写 EFlash 寄存器，解锁方式参考如下步骤 1 所描述。

为了保证 EFlash 正常工作，务必在系统时钟初始化完成后解锁 EFlash，然后配置 EFLASH_CNFG 配置寄存器的 CLKFRQ 位，具体配置请参考该寄存器位说明。

EFlash 命令操作流程均相同，操作步骤如下：

1. 通过向 EFLASH_KEY 解锁寄存器写入正确的解锁秘钥（依次写入 0x01020304 和 0x0A0B0C0D）解锁 EFlash；
2. 通过检测 EFLASH_STATE 状态寄存器的 BUSY 位，判断当前是否有命令正在执行。如果有命令正在执行（BUSY=1）则等待当前命令执行完成后（BUSY=0 或 FINISH=1），再执行步骤 3；
3. 配置 EFlash 相关配置寄存器以及 EFLASH_CMD 命令 ID 寄存器，然后配置 EFLASH_START 命令触发寄存器的 START 位为 1，触发 EFlash 命令执行。后续章节不同的命令仅针对该步骤做详细描述，其他通用的步骤不在做描述；
4. 通过检测 EFLASH_STATE 状态寄存器的 FINISH 位来判断该命令是否执行完成，检测 OPERR 位、WPERR 位等状态位来判断该命令执行是否发生错误；
5. 配置 EFLASH_STATE 状态寄存器，清除相关状态位；

6. 操作完 EFlash 寄存器后，可通过向 EFLASH_KEY 解锁寄存器写入非正确秘钥即可锁定 EFlash（正确解锁秘钥参见步骤 1）。

操作流程如下所示：

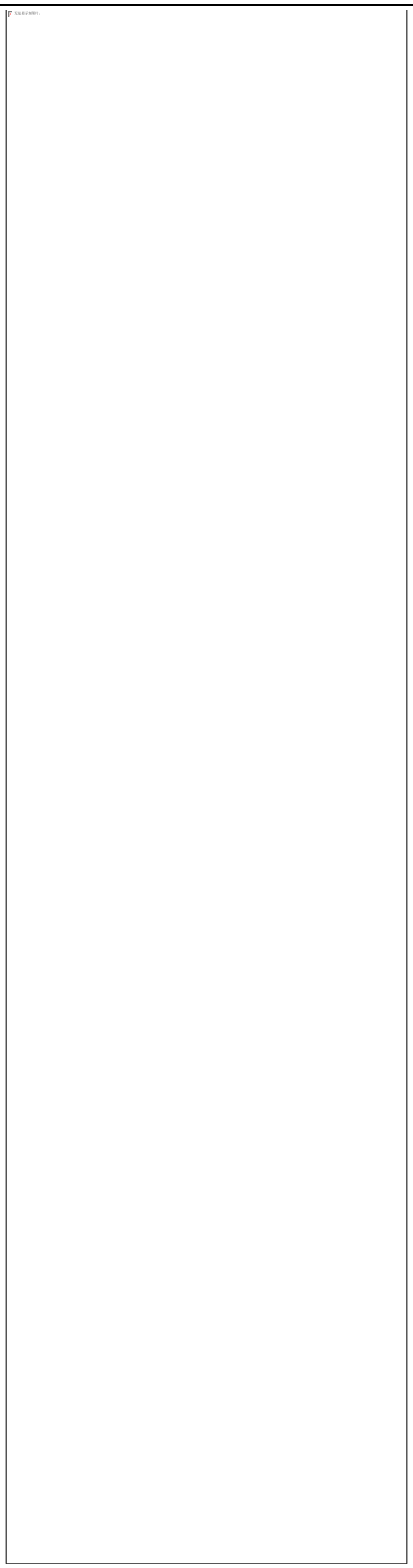


图 5.2 EFLASH 操作流程图

5.5.1 擦除操作

5.5.1.1 扇区擦除

扇区擦除命令能把指定地址所在的扇区的数据全部擦除，该命令的配置步骤如下：

1. 配置扇区擦除地址到 EFLASH_ADDR 地址寄存器，该地址需要 8 字节对齐，否则触发扇区擦除命令的执行时，会产生操作错误（即 EFLASH_STATE 寄存器的 OPERR 位置 1）；
2. 配置扇区擦除命令（0x01）到 EFLASH_CMD 命令 ID 寄存器；
3. 配置 EFLASH_START 寄存器的 START 位为 1，触发扇区擦除命令的执行。

5.5.1.2 块擦除

块擦除命令可以擦除 P-Flash 或 D-Flash 的一个 Block 块的主存储区的全部数据，该命令的配置步骤如下：

1. 配置块擦除地址到 EFLASH_ADDR 地址寄存器；
2. 配置扇区擦除命令（0x02）到 EFLASH_CMD 命令 ID 寄存器；
3. 配置 EFLASH_START 寄存器的 START 位为 1，触发块擦除命令的执行。

5.5.1.3 写保护扇区擦除

写保护扇区擦除命令仅用于擦除写保护信息所在扇区，擦除地址范围为 0x0120_0000~0x0120_0FFF，擦除完成后，复位系统将会禁用写保护，同时会同步写保护信息到写保护寄存器，该命令的配置步骤如下：

1. 配置写保护扇区擦除地址到 EFLASH_ADDR 地址寄存器；
2. 配置写保护扇区擦除命令（0x03）到 EFLASH_CMD 命令 ID 寄存器；
3. 配置 EFLASH_START 寄存器的 START 位为 1，触发写保护扇区擦除命令的执行。

5.5.1.4 全片擦除

全片擦除命令可以擦除全部 EFlash 主存储区以及信息区，该命令的配置步骤如下：

1. 配置全片擦除命令（0x04）到 EFLASH_CMD 命令 ID 寄存器；
2. 配置 EFLASH_START 寄存器的 START 位为 1，触发全片擦除命令的执行。

5.5.2 编程操作

编程命令用于编程数据到 EFlash 中，最小的编程单位为 64-bit。需要注意的是：根据 EFlash 的要求，单次编程操作不允许跨行编程（即，每次编程只允许对 EFlash 的一行进行编程数据），EFlash 一行的容量为 512B（即 64*64bit）。该命令的配置步骤如下：

1. 配置编程地址到 EFLASH_ADDR 寄存器，该地址需要 8 字节对齐，否则触发编程命令的执行时，会产生操作错误（即 EFLASH_STATE 寄存器的 OPERR 位置 1）；
2. 配置编程长度到 EFLASH_LEN 寄存器，该位的最小长度为 1（即 64-bit），最大长度为 64（即 EFlash 一行的数据长度）；
3. 配置编程命令（0x10）到 EFLASH_CMD 寄存器；
4. 配置编程数据低 32-bit 到 EFLASH_DATA0 寄存器；
5. 配置编程数据高 32-bit 到 EFLASH_DATA1 寄存器；
6. 若编程长度大于 1，则需要继续执行步骤 4,、5，直至将编程数据全部写入；
7. 配置 EFLASH_START 寄存器的 START 位为 1，触发编程命令的执行。

5.5.3 验证 Security Key

验证 Security Key 命令可以在 EFlash 数据不被擦除的前提下临时解除芯片读保护，在内核和外设不复位的前提下进行 debug，当复位后芯片又会恢复到读保护状态。因此，该命令使用在读保护生效的前提下，该命令能够正确执行需要具备以下条件：

1. 默认验证 Security Key 功能是被禁用，所以首先需要通过编程命令写地址 0x0120_1000 的 bit2 和 bit3 的值为非 11b 来使能验证 Security Key 功能；
2. 通过编程命令编程地址 0x0120_1008 从而设置信息区 Key 且复位后生效；
3. 触发验证命令执行时，EFLASH_SECKEY0 和 EFLASH_SECKEY1 组成的 64-bit Security Key 值和信息区所存在信息区（地址为 0x0120_1008）的 Key 值匹配；

当 Security Key 使能位为非 11b 时：

- 如果用于验证的 Security Key 和存储在信息区的 Key 相等，则读保护状态会被临时解除直到复位，此过程 Flash 的数据不受任何影响；
- 如果用于验证的 Security Key 是 64-bit 的全 1 或者全 0，则不会解除读保护状态；
- 如果用于验证的 Security Key 和 EFlash 信息区的 Key 不匹配，则在下次执行验证 Security Key 命令前必须复位，否则该命令不会再次被执行。

该命令的配置步骤如下：

1. 配置验证 Security Key 命令（0x20）到 EFLASH_CMD 命令 ID 寄存器；
2. 配置 Security Key 数据到 EFLASH_SECKEY0 和 EFLASH_SECKEY1 寄存器中；
3. 配置 EFLASH_START 寄存器的 START 位为 1，触发验证 Security Key 命令的执行。

5.5.4 Read while write

EFlash 控制器在编程或擦除 EFlash 其中一个 block 期间，支持对另一个 block 进行读取操作。

5.5.5 ECC

EFlash 支持 ECC 功能，相关功能如下：

1. ECC 1-bit 错误自动纠正，且会将 EFLASH_STATE 寄存器的 ECC1ERR 位置 1；
2. ECC 2-bit 错误无法纠正，且会将 EFLASH_STATE 寄存器的 ECC2ERR 位置 1，并记录发生该错误的地址，存入 EFLASH_ECC_ADDR 寄存器中；
3. 若 ECC 错误中断使能（即 EFLASH_CNFG 寄存器的 ECC2EIE 位置 1），则当检测到 ECC 发生 2-bit 错误时，会产生 ECC 中断，并将 EFLASH_STATE 的 ECC2ERR 位置 1，建议在中断中写 1 清除该中断标记。

5.5.6 写保护

5.5.6.1 使能写保护

- 默认 PFlash、DFlash 写保护失能。
- 使能读保护有两种方式：
  - 通过写保护寄存器设置写保护
  - 通过写保护信息区设置写保护

表 5.5 写保护寄存器说明表

写保护寄存器	EFLASH_PPROT	EFLASH_DPROT
寄存器地址	0x4210_0034	0x4210_0038
寄存器有效位宽	32-bit	8-bit
使能写保护方式	寄存器对应位写 0	
失能写保护方式	寄存器对应位写 1	
写保护位管控大小	64KB	
保护位和区域关系	寄存器低位保护低地址区域 寄存器高位保护高地址区域	
写保护生效时机	立即生效	

表 5.6 写保护信息区说明表

写保护信息区	P-Flash 写保护信息区	D-Flash 写保护信息区
地址	0x0120_0000	0x0120_0004
有效位宽	32-bit	8-bit
使能写保护方式	使用编程命令将对应位编程为 0	
失能写保护方式	使用写保护信息扇区擦除指令 擦除该地址所在扇区	
写保护位管控大小	64KB	
保护位和区域关系	寄存器低位保护低地址区域 寄存器高位保护高地址区域	
写保护生效时机	复位后生效	

1. P-Flash 写保护由 32-bit 控制，P-Flash 主存储区被划分成 32 个区域，例如 P-Flash 空间大小为 2MB，则每个写保护位控制 2MB/32=64KB 大小的区域；
2. D-Flash 写保护由 8-bit 控制，D-Flash 主存储区被划分成 8 个区域，例如 D-Flash 空间大小为 512KB，则每个写保护位控制 512KB/8=64KB 大小的区域；
3. 如果设置对应区域为写保护(写保护位为 0)，则擦除或编程操作该区域会失败且将 EFLASH_STATE 寄存器的 WPERR 位置 1；
4. 通过写保护寄存器设置写保护后会立即生效且不会同步到写保护信息区；
5. 通过写保护信息区设置写保护后会在复位后生效且同步写保护信息到写保护寄存器；
6. 通过写保护信息区设置写保护生效后，仍然可以通过写保护寄存器修改写保护状态，但是复位后写保护状态以写保护信息区为准直到再次更新写保护状态。

5.5.6.2 失能写保护

1. 失能写保护方式有以下几种方式：
2. 程序调用全片擦除命令(命令 ID 为 0x04)，请参考对应章节了解该命令使用流程；

3. 如果当前写保护状态是通过写保护信息区设置的，则可调用写保护信息页擦除命令失能写保护，请参考对应章节了解该命令使用流程；
4. 如果当前写保护状态是通过写保护寄存器设置的，则可通过写保护寄存器对应位为 1 即可失能写保护。

5.5.7 读保护

5.5.7.1 使能读保护

通过编程命令编程安全信息区即地址为 0x0120 1000 的 bit[1:0]为非 11b 就可以设置读保护，读保护需要复位后生效，读保护生效后，若此时处于 JTAG 工作状态下或 BOOT 选择的启动方式并非为 EFlash 时，则无法访问整个 EFlash 空间，软件可以通过读取 EFLASH_SEC 寄存器的 RPORT 位获取读保护状态。

5.5.7.2 失能读保护

失能读保护方式有以下几种方式：

1. 执行全片擦除命令(命令 ID 为 0x04)失能读保护，请参考对应章节了解该命令使用流程;
2. 执行验证 Security Key 失能读保护，请参考对应章节了解该命令使用流程。

5.5.8 时钟切换

在系统工作过程中，如需切换时钟频率，请严格按照以下步骤进行操作：

1. 通过向 EFLASH_KEY 解锁寄存器写入正确的解锁秘钥（依次写入 0x01020304 和 0x0A0B0C0D）解锁 EFlash；
2. 通过检测 EFLASH_STATE 状态寄存器的 BUSY 位，判断当前是否有命令正在执行。如果有命令正在执行（BUSY=1）则等待当前命令执行完成后（BUSY=0 或 FINISH=1），再执行后续操作；
3. 通过向 EFLASH_CNFG 配置寄存器的 CLKFRQ 位写入待切换的时钟频率，触发时钟切换逻辑(配置 CLKFRQ 位时须保证，写入的待切换时钟频率与原时钟频率不一致)；

4. 通过检测 EFLASH_STATE 状态寄存器的 CLKSWRDY 位，判断时钟切换准备工作是否完成。如果控制器未完成准备工作（CLKSWRDY=0）则需继续等待，直至完成后（CLKSWRDY=1）再执行后续操作；
5. 通过 SMU 配置时钟切换后，等待时钟稳定，再进行后续操作；
6. 配置 SMU 的 ab0CfgReg 寄存器的 bit[5]，向 EFlash 发送时钟更新使能，完成时钟切换。

注：

1. 在进行时钟切换时，请确保 EFlash 处于空闲状态，未进行编程、擦除等操作。
2. 有关系统时钟切换的相关说明，请参考 SMU 切换系统时钟流程章节。

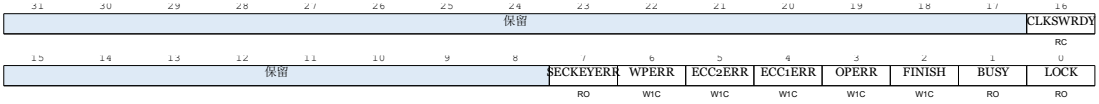
5.6 寄存器

EFlash 控制器基地址：0x4210_0000

5.6.1 EFlash 状态寄存器（EFLASH_STATE）

地址偏移：0x00

复位值：0x0000_0003



位/位域	名称	描述
31:17	保留	必须保持复位值
16	CLKSWRDY	Flash 时钟切换状态标志 0：时钟切换准备工作未完成 1：时钟切换准备工作已完成 该标志位由软件读清零，用于时钟切换时的状态检测。 注：该标志位用于判断时钟切换状态，仅当写入 CLKFRQ 寄存器的值与 CLKFRQ 寄存器原数值不相等

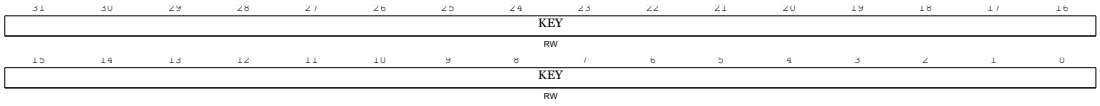
位/位域	名称	描述
		时才会触发时钟切换逻辑；若写入 CLKFRQ 寄存器的值与 CLKFRQ 寄存器原数值相等，则 CLKSWRDY 标志位不会置 1。
15:8	保留	必须保持复位值
7	SECKEYERR	Flash Security Key 验证错误标志 0：未检测到 Security Key 验证错误 1：检测到 Security Key 验证错误 该标志位仅可在 EFlash 控制器复位时硬件清除
6	WPERR	Flash 写保护错误标志 当编程/擦除有写保护的 Flash 区域时，会将该位置位 0：未检测到写保护错误 1：检测到写保护错误
5	ECC2ERR	Flash ECC 校验发生多 bit 错误 0：ECC 校验未发生多 bit 错误 1：ECC 校验发生多 bit 错误
4	ECC1ERR	Flash ECC 校验发生单 bit 错误 0：ECC 校验未发生单 bit 错误 1：ECC 校验发生单 bit 错误
3	OPERR	Flash 操作错误标志 0：未检测到操作错误 1：检测到操作错误
2	FINISH	Flash 命令完成标志

位/位域	名称	描述
		0：Flash 命令未完成 1：Flash 命令完成
1	BUSY	Flash 忙标志 0：Flash 当前未执行任何命令 1：Flash 当前正在执行命令或 Flash 处于初始化阶段
0	LOCK	Flash 锁定状态标志 0：Flash 未锁住，能写 Flash 寄存器 1：Flash 锁住，不能写 Flash 寄存器

5.6.2 EFlash 解锁寄存器（EFLASH_KEY）

地址偏移：0x04

复位值：0x0000_0000

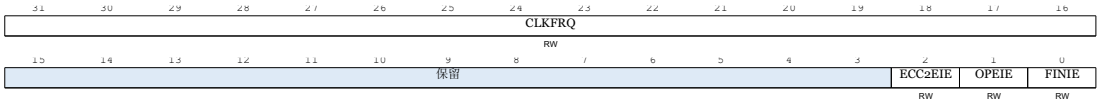


位/位域	名称	描述
31:0	KEY	Flash 解锁寄存器 当 LOCK 寄存器为 1 时，依次向解锁寄存器写入解锁秘钥 0x01020304 和 0x0A0B0C0D 可以解锁 Flash 寄存器，此时 LOCK 寄存器值变为 0。 当向解锁寄存器写入的数据非正确的解锁秘钥则会锁住 FLASH 寄存器。

5.6.3 EFlash 配置寄存器（EFLASH_CNFG）

地址偏移：0x08

复位值：0x0010_0000

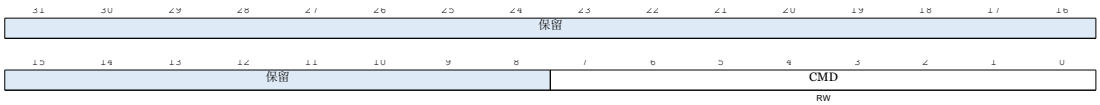


位/位域	名称	描述
31:16	CLKFRQ	时钟频率 由于 Flash 的时钟和系统时钟同步，而 Flash 硬件的编程和擦除需要周期为 1us 的脉冲，所以如果系统时钟频率为 100MHZ，则必须写入 0x64 到该寄存器
15:3	保留	必须保持复位值
2	ECC2EIE	Flash ECC 2 bit 错误中断使能 0：禁用 ECC 2 bit 错误中断 1：使能 ECC 2bit 错误中断
1	OPEIE	Flash 操作错误中断使能 0：禁用操作错误中断 1：使能操作错误中断
0	FINIE	Flash 命令完成中断使能 0：禁用命令完成中断 1：使能命令完成中断

5.6.4 EFlash 命令 ID 寄存器（EFLASH_CMD）

地址偏移：0x0C

复位值：0x0000_0000

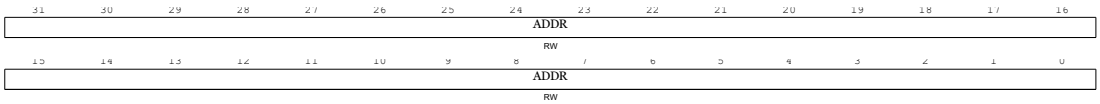


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CMD	Flash 命令 ID 寄存器 8’h01：扇区擦除 8’h02：块擦除 8’h03：写保护扇区擦除 8’h04：全片擦除 8’h10：编程 8’h20：验证 Security Key 使用其他 ID 则会置位寄存器 OPERR，产生操作错误

5.6.5 EFlash 地址寄存器（EFLASH_ADDR）

地址偏移：0x10

复位值：0x0000_0000

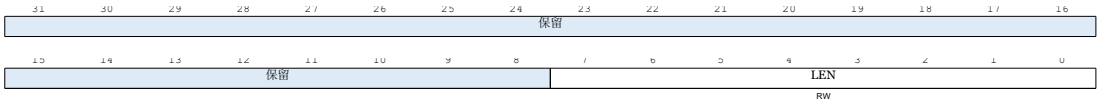


位/位域	名称	描述
31:0	ADDR	FLASH 操作地址寄存器

5.6.6 EFlash 数据长度寄存器（EFLASH_LEN）

地址偏移：0x14

复位值：0x0000_0000

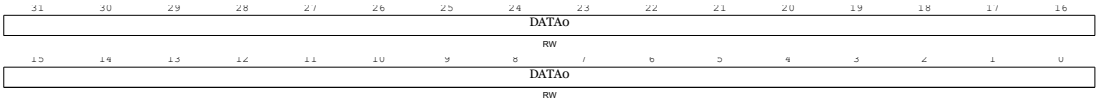


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	LEN	FLASH 数据长度寄存器

5.6.7 EFlash 数据寄存器 0（EFLASH_DATA0）

地址偏移：0x20

复位值：0x0000_0000

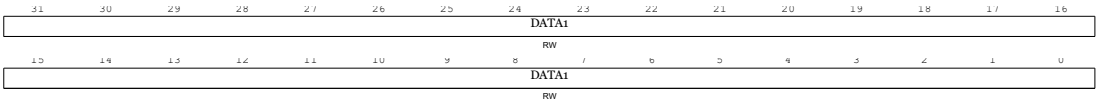


位/位域	名称	描述
31:0	DATA0	FLASH 数据寄存器低 32bit

5.6.8 EFlash 数据寄存器 1（EFLASH_DATA1）

地址偏移：0x24

复位值：0x0000_0000

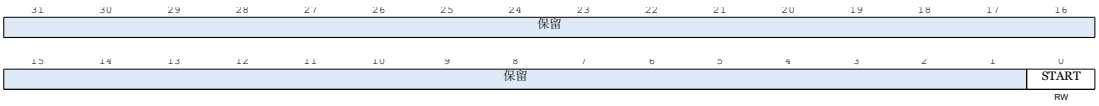


位/位域	名称	描述
31:0	DATA1	FLASH 数据寄存器高 32bit

5.6.9 EFlash 命令触发寄存器（EFLASH_START）

地址偏移：0x30

复位值：0x0000_0000

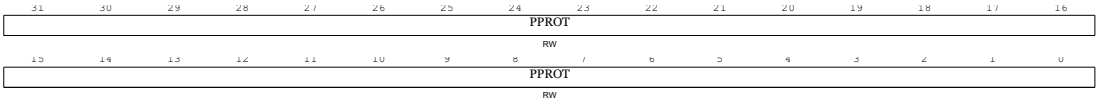


位/位域	名称	描述
31:1	保留	必须保持复位值
0	START	触发命令执行 该寄存器写 1 将会触发 Flash 命令的执行，如果当前 Flash 正处于忙碌状态则该命令将会被取消，当该命令执行完成，该寄存器会自动被清零。

5.6.10 P-Flash 写保护寄存器（EFLASH_PPROT）

地址偏移：0x34

复位值：0xFFFF_FFFF

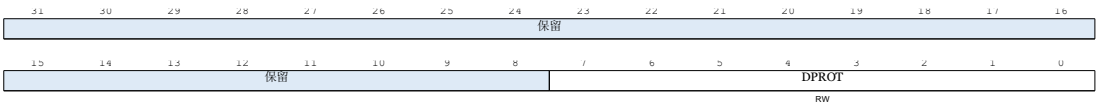


位/位域	名称	描述
31:0	PPROT	PFlash 写保护状态值 0：使能写保护 1：禁用写保护 该寄存器把 P-Flash 划分成 32 个区域，如果 P-Flash 大小是 2M 则每个区域大小为 64KB，每一个 bit 代表着一个区域的写保护状态。 复位后，该寄存器将会同步 Flash 信息区的写保护信息，通过寄存器设置的写保护立即生效。 擦除或者编程写保护的区域，将会触发写保护错误(置位 EFLASH_STATE[WPERR])。

5.6.11 D-Flash 写保护寄存器（EFLASH_DPROT）

地址偏移：0x38

复位值：0x0000_00FF



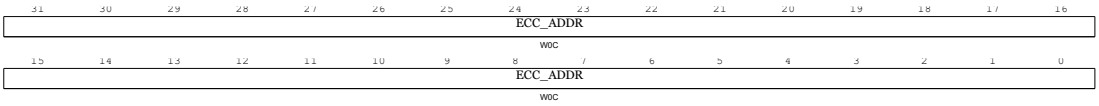
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DPROT	DFlash 写保护状态值 0：使能写保护

位/位域	名称	描述
		1：禁用写保护 该寄存器把 D-Flash 划分成 8 个区域，如果 D-Flash 大小是 512K 则每个区域大小为 64KB，每一个 bit 代表着一个区域的写保护状态。 复位后，该寄存器将会同步 Flash 信息区的写保护信息，通过寄存器设置的写保护立即生效。 擦除或者编程写保护的区域，将会触发写保护错误(置位 EFLASH_STATE[WPERR])。

5.6.12 EFlash ECC 错误地址寄存器（EFLASH_ECC_ADDR）

地址偏移：0x3C

复位值：0x0000_0000

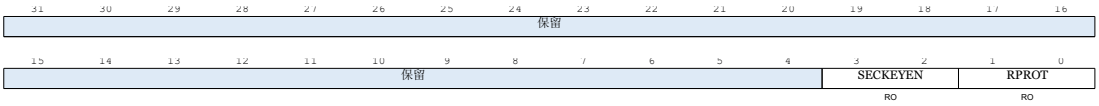


位/位域	名称	描述
31:0	ECC_ADDR	ECC 2bit 错误地址 当 Flash 产生 2-bit ECC 错误时候会将错误的地址记录在该寄存器。 该寄存器可以写 0 清除。

5.6.13 EFlash 安全寄存器（EFLASH_SEC）

地址偏移：0x40

复位值：0x0000_000F



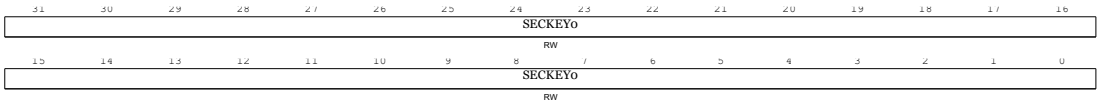
位/位域	名称	描述
31:4	保留	必须保持复位值

位/位域	名称	描述
3:2	SECKEYEN	Flash Security Key 使能标志 2‘b11：失能 其他：使能
1:0	RPROT	Flash 读保护使能标志 2‘b11：读保护失能 其他：读保护使能

5.6.14 EFlash 安全密钥寄存器 0（EFLASH_SECKEY0）

地址偏移：0x44

复位值：0x0000_0000

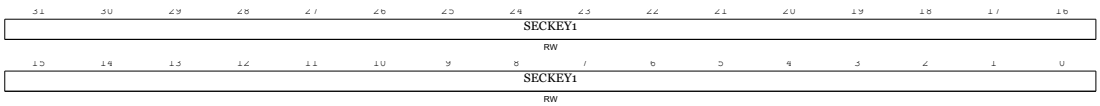


位/位域	名称	描述
31:0	SECKEY0	Flash 安全密钥低 32bit SECKEY0 与 SECKEY1 组成乘 64 位安全密钥，用于验证该密钥是否与信息区的 Security Key 相等： 若相等，则读保护状态奖杯临时解除直到复位; 若不相等，则在下次执行验证 Security Key 命令前，必须复位。

5.6.15 EFlash 安全密钥寄存器 1（EFLASH_SECKEY1）

地址偏移：0x48

复位值：0x0000_0000



位/位域	名称	描述
31:0	SECKEY1	Flash 安全密钥高 32bit SECKEY0 与 SECKEY1 组成乘 64 位安全密钥，用于验证该密钥是否与信息区的 Security Key 相等： 若相等，则读保护状态奖杯临时解除直到复位; 若不相等，则在下次执行验证 Security Key 命令前，必须复位。

6 QSPI-Flash 控制器（QSPI）

6.1 简介

QSPI Flash 控制器主要用于完成系统对外部 QSPI Flash 的访问控制，按照 QSPI 接口协议与 Flash 进行通信。该控制器主要包括以下三种工作模式：

- 间接模式：AXI Lite 总线控制寄存器实现全部操作；
- 自动轮询模式：周期性读取外部 Flash 状态寄存器，查询外部 Flash 状态，如是否完成烧写或擦除等操作；
- 内存映射模式：外部 Flash 映射到 CPU 地址空间，直接根据 CPU 地址从外部 Flash 中读取数据。

6.2 功能说明

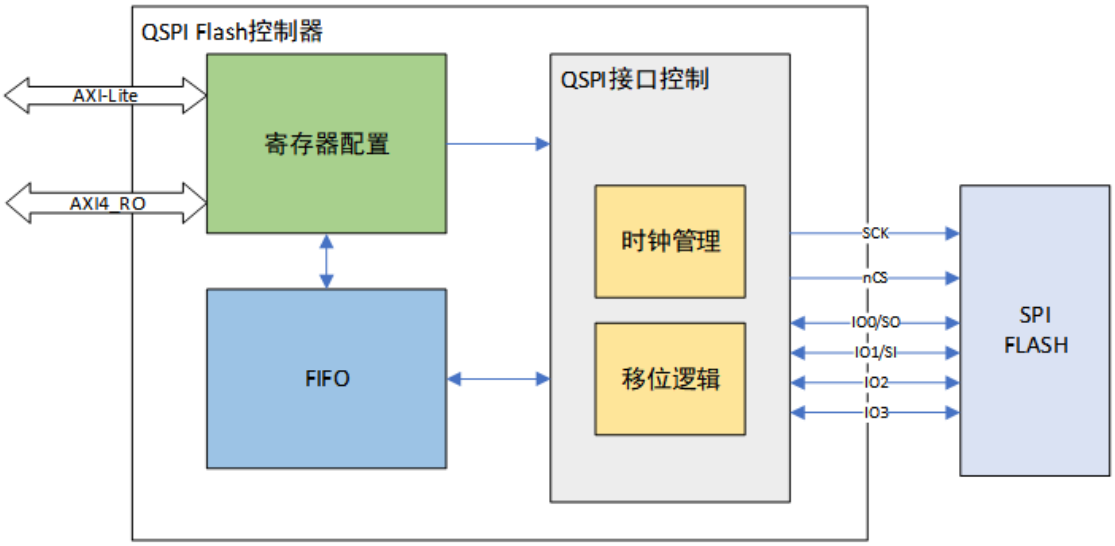


图 6.1 QSPI 控制器设计框图

6.2.1 QSPI 指令说明

QSPI Flash 控制器通过命令与外部 Flash 进行通信。由于 Flash 芯片厂商、芯片型号等方面不同，QSPI 命令存在一定差异。总体来说，QSPI 命令主要包括指令、地址、交替字节、空闲周期和数据这五个阶段，部分阶段可以跳过。

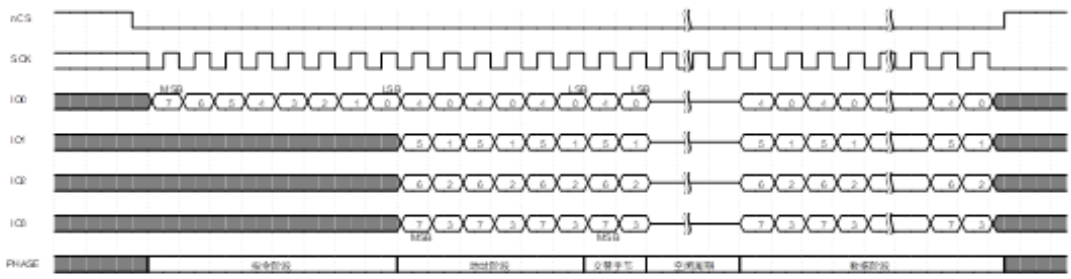


图 6.2 SDR 传输命令序列时序图

6.2.1.1 指令阶段

在指令阶段，控制器可将 QSPI_CCR 寄存器的 INST 字段中配置一条 8 位指令发送给外部 Flash。通过配置 QSPI_CCR 寄存器中的 IMODE 字段可以选择发送指令的模式，可以选择以单线、双线或四线模式进行发送。若 IMODE=00，则表示无指令，控制器不进行任何操作。

6.2.1.2 地址阶段

在地址阶段，控制器向外部 Flash 发送 1~4 字节的地址数据。通过配置 QSPI_CCR 寄存器中的 ADMODE 字段可以选择发送地址的模式，可以选择以单线、双线或四线模式进行发送。若 ADMODE=00，则表示无地址阶段，可跳过该阶段。

通过配置 QSPI_CCR 寄存器中的 ADSIZE 字段可以选择发送地址字节数。

在间接模式和自动轮询模式下，待发送的地址字节由 QSPI_AR 寄存器指定；在内存映射模式下，待发送的地址由 AXI Lite 总线直接给出。

6.2.1.3 交替字节阶段

在交替字节阶段，控制器向外部 Flash 发送 1~4 字节的交替字节数据。通过配置 QSPI_CCR 寄存器中的 ABMODE 字段可以选择发送交替字节的模式，可以选择以单线、双线或四线模式进行发送。若 ABMODE=00，则表示无交替字节阶段，可跳过该阶段。

通过配置 QSPI_CCR 寄存器中的 ABSIZE 字段可以选择发送交替字节的字节数。

待发送的交替字节由 QSPI_ABR 寄存器指定。

6.2.1.4 空闲周期阶段

在地址传输完成后的一段时间内不发送或接收任何数据，即空闲周期。目的是当时钟频率较高时，给 Flash 充足的时间准备好待传输的数据，保证传输的稳定性。

空闲周期阶段，可通过配置 QSPI_CCR 寄存器中的 DUMMY 字段开启。DUMMY 为 0 时，则跳过空闲周期阶段。

6.2.1.5 数据阶段

在数据阶段，可向外部 Flash 写入数据或从外部 Flash 读取数据。通过配置 QSPI_CCR 寄存器中的 DMODE 字段可以选择发送或接收数据的模式，可以选择以单线、双线或四线模式进行发送或接收。若 DMODE=00，则表示无数据阶段，可跳过该阶段。

在间接模式和轮询模式下，可通过对数据寄存器进行访问来获取从 Flash 读取的数据或提供写入 Flash 的数据，读取或写入 Flash 的字节数在 QSPI_DLR 数据长度寄存器中指定。在内存映射模式下，从 Flash 读取的数据通过 AXI Lite 总线直接发送回 CPU。

6.2.2 QSPI 接口协议

QSPI Flash 控制器与外部 Flash 之间的通信接口一共有 6 个，分别为 1 个片选信号输出端口、1 个时钟信号输出端口和 4 个数据信号输入/输出端口。

6.2.2.1 单线模式

单线模式下，控制器通过 SO 端口发送数据到外部 Flash，通过 SI 端口接收数据。

表 6.1 单线模式，QSPI 接口功能列表

IO0（SO）	输出（发送数据）
IO1（SI）	输入（接收数据）
IO2	输出，强制置 1（禁止写保护功能）
IO3	输出，强制置 1（禁止保持功能）

6.2.2.2 双线模式

双线模式下，控制器通过 IO0、IO1 端口同时发送或接收数据，每个时钟可串行发送或接收 2 位数据。

表 6.2 双线模式，QSPI 接口功能列表

IO0（SO）	读取数据时为输入（高阻态），其他情况为输出
IO1（SI）	读取数据时为输入（高阻态），其他情况为输出
IO2	输出，强制置 1（禁止写保护功能）
IO3	输出，强制置 1（禁止保持功能）

6.2.2.3 四线模式

四线模式下，控制器通过 IO0、IO1、IO2、IO3 端口同时发送或接收数据，每个时钟可串行发送或接收 4 位数据。

四线模式下，IO0、IO1、IO2、IO3 端口在读取数据时为输入状态（高阻态），其他情况下均为输出状态。

6.2.2.4 nCS 片选信号和 SCK 时钟信号

QSPI Flash 控制器提供片选信号（nCS）和同步时钟（SCK），控制数据传输的启动与停止。片选信号（nCS）默认为低电平有效，只有在片选有效时，外部 Flash 才能与 QSPI Flash 控制器在同步时钟的驱动下进行通信，无效时则释放 Flash，通信停止，空闲状态时，片选信号（nCS）为高电平，在通信开始前拉低，在通信完成时立即拉高。

QSPI Flash 控制器仅支持单倍数据速率模式（SDR）。在 SDR 模式下，控制器默认在 SCK 时钟的下降沿发送数据，在上升沿接收数据。通过配置 QSPI_CR 寄存器的 CKMODE 位，可以指定空闲模式下的时钟极性。CKMODE=0 表示 QSPI 工作在模式 0，CKMODE=1 表示 QSPI 工作在模式 3。

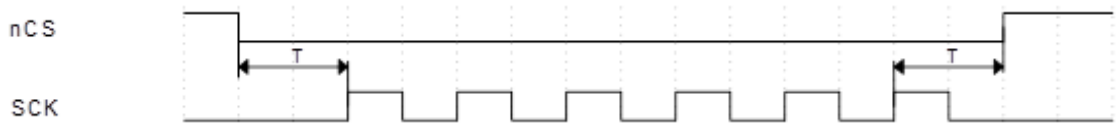


图 6.3 模式 0，nCS 和 SCK 时序图

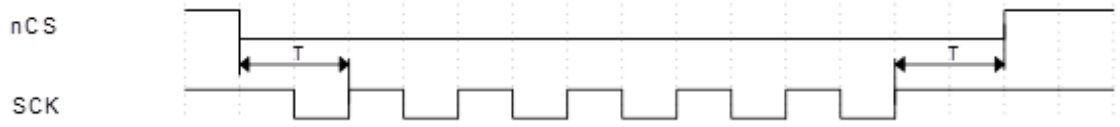


图 6.4 模式 3，nCS 和 SCK 时序图

6.2.3 QSPI 控制器功能

QSPI Flash 控制器根据不同的应用场景，可在三种不同模式下工作。

6.2.3.1 间接模式

间接模式下，可以完全通过 AXI Lite 总线访问控制器内部的寄存器完成对外部 Flash 的读、写访问。在将外部 Flash 存储器通过 QSPI 连接到 AXI Lite 总线后，需要通过间接模式发送相应的指令完成对外部 Flash 中部分寄存器的配置。在对 Flash 存储器执行读、编程或擦除等操作时，也可通过间接模式发送相应的指令，并提供必要的地址或数据信息。

QSPI_CCR 寄存器的 FMODE 字段用于选择工作模式。

若 FMODE=00，则控制器处于间接写入模式。AXI Lite 总线写入数据寄存器的数据将被发送给外部 Flash。

若 FMODE=01，则控制器处于间接读取模式。AXI Lite 总线通过读取数据寄存器获取外部 Flash 发送的数据。

传输的数据长度由 QSPI_DLR 寄存器控制。

在该模式下，只需要通过读写 QSPI 的寄存器即可完成命令序列触发、数据传输的全过程。命令序列触发原则是，只要提供了当前传输需要的所有配置信息即可出发：

1. 不需要提供地址和数据时，只提供指令即可启动；
2. 需要提供地址而不需要提供数据时，提供所需的指令和地址即可启动；
3. 需要提供地址和数据时，提供所需的指令、地址、数据即可启动。

在读写过程中，使用控制器内部一个 32 字节同步 FIFO 缓冲数据。写外部 Flash 时，AXI Lite 总线提供数据，对数据寄存器执行写操作，同时将数据写入 FIFO（支持字节、半字及字访问），在数据阶段，QSPI 接口侧从 FIFO 中读取数据，写入外部 Flash。读外部 Flash 时，QSPI 接口侧将从外部 Flash 中读取的数

据写入 FIFO，再由 AXI Lite 总线读数据寄存器即可将数据从 FIFO 中读出（支持字节、半字及字访问）。

6.2.3.2 自动轮询模式

Flash 存储器的编程或擦除等操作需要消耗大量的时间，为节约时间、节省资源，QSPI Flash 控制器启动自动轮询模式即可自动地、周期性地读取外部 Flash 的寄存器。

在自动轮询模式下，控制器周期性启动命令读取一定数量的数据，最多读取 4 字节。可屏蔽接收的字节，用于隔离某些状态位，当未屏蔽位与期望值相匹配时，则轮询匹配标志位置 1。

在自动轮询模式下，需要通过配置数据长度寄存器、轮询间隔寄存器、轮询状态屏蔽寄存器和轮询状态匹配寄存器完成首次启动，之后控制器会周期性地自动启动，直至匹配成功或发生中止（ABORT 置 1）时退出自动轮询模式。

6.2.3.3 内存映射模式

在内存映射模式下，AXI 总线上的主机可直接通过映射地址从 Flash 中读取数据。内存映射模式支持预取操作，控制器将预期下一次总线主机的访问，提前读取 Flash 连续地址上的字节并写入 FIFO。如果之后总线主机发送了访问并且地址连续，则访问可以更快完成，总线直接从 FIFO 中读取数据，若地址不连续，控制器将清空 FIFO 中的数据，重新从外部 Flash 中进行读取。若总线长时间没有有效的访问操作，为降低功耗，控制器中引入了超时计数器，当 FIFO 中写满预取的数据后，如果在设置的时钟周期内 AXI 总线没有发起访问 Flash 的操作，控制器将拉高片选，通信停止，以降低 Flash 功耗。

6.2.3.4 初始化状态说明

QSPI Flash 芯片作为非易失型存储器，主要用于存储程序、指令。芯片上电复位后，CPU 可从 Flash 中加载程序并运行。因此，QSPI 控制器上电复位时处于初始化状态。该状态下，控制器处于内存映射模式，根据 AXI Lite 总线主机提供的读操作地址进行指令的读取、加载。

QSPI 控制器在初始化状态下的配置情况如 QSPI 控制器初始化模式配置表所示。该状态下，控制器按照初始化状态下时序图所示的时序进行数据读取。

表 6.3 QSPI 控制器初始化模式配置表

偏移地址	寄存器名称	寄存器数值
0x00	控制寄存器（QSPI_CR）	0x0000_3F3D
0x04	通信配置寄存器（QSPI_CCR）	0x0680_2503
0x20	超时寄存器（QSPI_TOR）	0x0000_0014

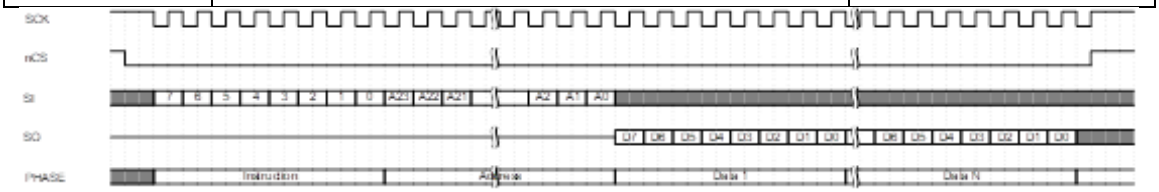


图 6.5 初始化状态下时序图

退出初始化状态的方式：

QSPI 控制器仅在 AXI Lite 总线配置 QSPI_CR 寄存器的 EN 位为 0（禁用 QSPI）或配置 QSPI_CR 寄存器的 ABORT 位为 1（中止操作）后，才能退出初始化状态，同时表 1.2.1 中所示的寄存器数值将全部自动清零。

若控制器需要切换至其他工作模式，可根据 1.3 配置说明所述，进行所需模式的配置。

6.3 配置说明

6.3.1 间接模式配置

- 配置控制寄存器（QSPI_CR）
- 使能 QSPI（EN=1）
  - 配置 CKMODE
  - 设置片选高电平时间（CSHT）
  - 设置预分频系数
- 配置交替字节寄存器（QSPI_ABR）
- 配置数据长度寄存器（QSPI_DLR）
- 配置通信配置寄存器（QSPI_CCR）
- 根据芯片手册相应指令进行配置

- FMODE 配置为 00，选择间接写入模式；配置为 01，选择间接读取模式。
  - INST 为指令字节
  - IMODE 为指令发送方式（1/2/4 线）
  - ADMODE 为地址发送方式（无/1/2/4 线）
  - ADSIZE 为地址长度（8/16/24/32 位）
  - ABMODE 为交替字节发送方式（无/1/2/4 线）
  - ABSIZE 为交替字节长度（8/16/24/32 位）
  - DUMMY 为空闲周期数
  - DMODE 为接收/发送数据方式（无/1/2/4 线）
- 配置地址寄存器（QSPI_AR）
- 配置数据寄存器（QSPI_DR）
- 根据 FMODE 模式，读取或写入数据寄存器：
- FMODE=00（间接写入模式）时，写入数据寄存器
- FMODE=01（间接读取模式）时，读取数据寄存器

Note

1. 如果 Flash 指令无需地址和数据位（即 ADMODE=00、DMODE=00），则配置 QSPI_CCR 寄存器后，该命令序列立即启动；如果 Flash 指令无需地址、需要数据时，则无需配置地址寄存器，该命令序列会在配置数据寄存器后立即启动；如果 Flash 指令需要地址、无需数据时，则无需配置数据寄存器，该命令序列会在配置地址寄存器后立即启动。
2. 交替字节寄存器和数据长度寄存器需在命令序列启动前完成配置。

6.3.2 轮询模式配置

- 配置控制寄存器（QSPI_CR）
- 使能 QSPI（EN=1）
  - 配置 CKMODE
  - 设置片选高电平时间（CSHT）

- 设置预分频系数

- 配置交替字节寄存器（QSPI_ABR）
- 配置数据长度寄存器（QSPI_DLR）

■ 该模式下，最大可配置为 3（表示读取 4 字节数据）

- 配置通信配置寄存器（QSPI_CCR）

■ 根据芯片手册相应指令进行配置

■ FMODE 配置为 10

- INST 为指令字节
- IMODE 为指令发送方式（1/2/4 线）
- ADMODE 为地址发送方式（无/1/2/4 线）
- ADSIZE 为地址长度（8/16/24/32 位）
- ABMODE 为交替字节发送方式（无/1/2/4 线）
- ABSIZE 为交替字节长度（8/16/24/32 位）
- DUMMY 为空闲周期数
- DMODE 为接收/发送数据方式（无/1/2/4 线）

- 配置地址寄存器（QSPI_AR）
- 配置轮询间隔寄存器（QSPI_PIR）
- 配置轮询状态匹配寄存器（QSPI_PSMAR）
- 配置轮询状态屏蔽寄存器（QSPI_PSMKR）

Note

该模式下，命令序列会在配置完数据长度寄存器、轮询间隔寄存器、轮询状态匹配寄存器、轮询状态屏蔽寄存器后立即启动。如需配置地址寄存器，须在命令序列启动前完成配置。

6.3.3 内存映射模式配置

- 配置控制寄存器（QSPI_CR）

■ 使能 QSPI（EN=1）

- 配置 CKMODE

- 设置片选高电平时间（CSHT）
- 设置预分频系数

- 配置通信配置寄存器（QSPI_CCR）

- 根据芯片手册相应指令进行配置
- FMODE 配置为 11
  - INST 为指令字节
  - IMODE 为指令发送方式（1/2/4 线）
  - ADMODE 为地址发送方式（无/1/2/4 线）
  - ADSIZE 为地址长度（8/16/24/32 位）
  - ABMODE 为交替字节发送方式（无/1/2/4 线）
  - ABSIZE 为交替字节长度（8/16/24/32 位）
  - DUMMY 为空闲周期数
  - DMODE 为接收/发送数据方式（无/1/2/4 线）

Note

该模式下，命令序列会在配置完通信配置寄存器（QSPI_CCR）后，AXI Lite 总线发起读取操作时，自动启动。

6.3.4 预分频系数配置流程

时钟预分频器 PRESCALER 的详细说明请参考 QSPI 控制寄存器（QSPI_CR）的 PRESCALER 位域。

预分频系数配置流程：

1. 配置 QSPI_CR 寄存器的 PRESCALER 字段；
2. 读取 QSPI_CR 寄存器数据，判断 PRESCALER 字段是否配置成功，若数据与预期不符，则需要重复执行步骤 1，直至配置成功。

Note

1. PRESCALER 时钟预分频系数仅在 BUSY=0 时，可以进行修改。

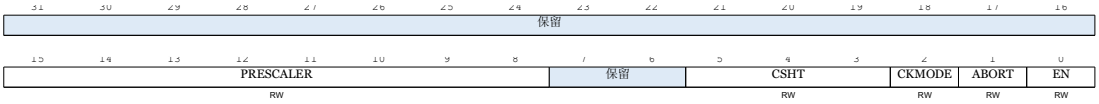
2. QSPI 控制器上电初始化状态为内存映射模式，当退出初始化状态时，配置的 PRESCALER 寄存器将会被清零。具体说明请参考应用说明 初始化状态说明。
3. 若 BOOT≠10（即非 QSPI-Flash 启动），可在步骤 1 前，执行读取 BUSY 状态位，当 BUSY=0 后，再执行步骤 1。

6.4 寄存器

6.4.1 QSPI 控制寄存器（QSPI_CR）

地址偏移：0x00

复位值：0x0000_3F3D



位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	PRESCALER	时钟预分频器 该字段定义了 QSPI 时钟 CLK 与 AXIBUS0 总线时钟的分频比： Fclk=FAXIBUS0/（PRESCALER+1），其中 PRESCALER 的范围为 1~255。 0：Fclk=FAXIBUS0/2 1：Fclk=FAXIBUS0/2 2：Fclk=FAXIBUS0/3 … 255：Fclk = FAXIBUS0/256 对于奇数时钟分频系数，QSPI 时钟信号高电平持续时间比低电平持续时间多一个周期。 注：

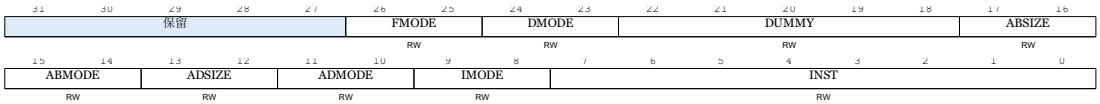
位/位域	名称	描述
		1、该字段仅可在 BUSY=0 时进行修改 2、PRESCALER 配置为 0 时，默认为 2 分频
7:6	保留	必须保持复位值
5:3	CSHT	片选高电平时间 该字段定义了发送到 QSPI Flash 的指令之间片选信号（nCS）信号必须保持高电平的最少时钟周期为（CSHT+1）。 0：在发送指令之间，nCS 信号至少保持高电平 1 个周期 1：在发送指令之间，nCS 信号至少保持高电平 2 个周期 … 7：在发送指令之间，nCS 信号至少保持高电平 8 个周期 注：该字段仅可在 BUSY=0 时进行修改
2	CKMODE	模式 0/模式 3 该位定义了 QSPI 时钟信号在指令之间的电平。 0：nCS 为高电平时，QSPI 时钟信号 CLK 保持低电平（Mode0） 1：nCS 为高电平时，QSPI 时钟信号 CLK 保持高电平（Mode3） 注：该位仅可在 BUSY=0 时进行修改
1	ABORT	中止请求 任何操作都可通过将该位置 1 来中止。

位/位域	名称	描述
		0：不中止当前传输 1：中止当前传输
0	EN	使能 使能 QSPI 0：禁止 QSPI 1：使能 QSPI

6.4.2 QSPI 通信配置寄存器（QSPI_CCR）

地址偏移：0x04

复位值：0x0680_2503



位/位域	名称	描述
31:27	保留	必须保持复位值
26:25	FMODE	工作模式 该字段定义了 QSPI 的工作模式： 00：间接写入模式 01：间接读取模式 10：自动轮询模式 11：内存映射模式 该字段仅可在 BUSY=0 时进行修改
24:23	DMODE	数据模式 该字段定义了数据阶段的操作模式： 00：无数据

位/位域	名称	描述
		01：单线传输数据 10：双线传输数据 11：四线传输数据 该字段仅可在 BUSY=0 时进行修改
22:18	DUMMY	空闲周期数 该字段定义了空闲周期的持续时间。 该字段仅可在 BUSY=0 时进行修改
17:16	ABSIZE	交替字节长度 该字段定义了交替字节长度： 00：8 位交替字节 01：16 位交替字节 10：24 位交替字节 11：32 位交替字节 该字段仅可在 BUSY=0 时进行修改
15:14	ABMODE	交替字节模式 该字段定义了交替字节阶段的操作模式： 00：无交替字节 01：单线传输交替字节 10：双线传输交替字节 11：四线传输交替字节 该字段仅可在 BUSY=0 时进行修改
13:12	ADSIZE	地址长度 该字段定义了地址长度：

位/位域	名称	描述
		00：8 位地址 01：16 位地址 10：24 位地址 11：32 位地址 该字段仅可在 BUSY=0 时进行修改
11:10	ADMODE	地址模式 该字段定义了地址阶段的操作模式： 00：无地址 01：单线传输地址 10：双线传输地址 11：四线传输地址 该字段仅可在 BUSY=0 时进行修改
9:8	IMODE	指令模式 该字段定义了指令阶段的操作模式： 00：无指令 01：单线传输指令 10：双线传输指令 11：四线传输指令 该字段仅可在 BUSY=0 时进行修改
7:0	INST	指令 该字段定义了发送到 QSPI Flash 的指令 该字段仅可在 BUSY=0 时进行修改

6.4.3 QSPI 状态寄存器（QSPI_SR）

地址偏移：0x08

复位值：0x0000_0000

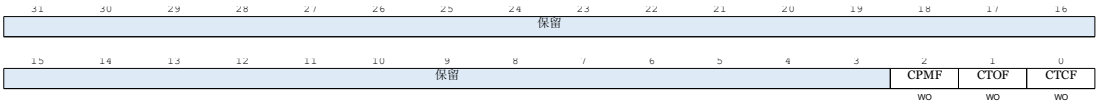
位/位域	名称	描述
31:10	保留	必须保持复位值
9	PMF	轮询匹配标志 在轮询状态下，未屏蔽的接收数据与匹配寄存器中的对应位数据相匹配，则该位置 1。可向 CPMF 写入 1，将该位清零。
8	TOF	超时标志 发生超时时，该位置 1。可向 CTOF 写入 1，将该位清零。
7	TCF	传输完成标志 在间接模式下，传输数据量达到设定值时，或在任意模式下，传输中断时，该位置 1。可向 CTCF 写入 1，将该位清零。
6:1	FLEVEL	FIFO 数据量 该字段表示 FIFO 中有效字节数。FIFO 为空时，FLEVEL=0；FIFO 写满时，FLEVEL=32。
0	BUSY	忙

位/位域	名称	描述
		QSPI Flash 控制器启动对外部 Flash 的操作时，该位置 1。 间接模式下，QSPI Flash 控制器完成了当前命令序列，且 FIFO 为空时，BUSY 位清零。 自动轮询模式下，当轮询数据匹配或发生中止时，BUSY 位清零。 内存映射模式下，在进行第一次访问后，仅在发生超时时间或中止时，BUSY 位清零。

6.4.4 QSPI 标志清除寄存器（QSPI_FCR）

地址偏移：0x0C

复位值：0x0000_0000

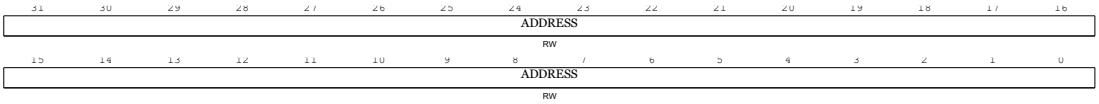


位/位域	名称	描述
31:3	保留	必须保持复位值
2	CPMF	清除轮询匹配标志 写入 1 可将状态寄存器中的 PMF 轮询匹配标志位清零。
1	CTOF	清除超时标志 写入 1 可将状态寄存器中的 TOF 超时标志位清零。
0	CTCF	清除传输完成标志 写入 1 可将状态寄存器中的 TCF 传输完成标志位清零。

6.4.5 QSPI 地址寄存器（QSPI_AR）

地址偏移：0x10

复位值：0x0000_0000

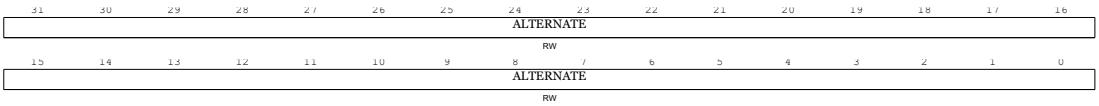


位/位域	名称	描述
31:0	ADDRESS	地址 该字段定义了发送至 QSPI Flash 的地址 FMODE=11 时，忽略对该字段的写入

6.4.6 QSPI 交替字节寄存器（QSPI_ABR）

地址偏移：0x14

复位值：0x0000_0000

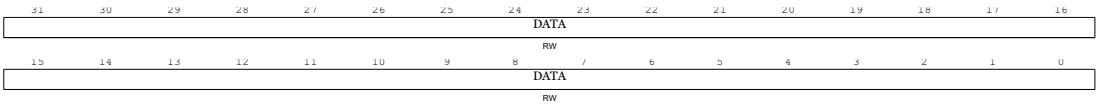


位/位域	名称	描述
31:0	ALTERNATE	交替字节 该字段定义了地址发送后需要发送至 QSPI Flash 的可选数据 该字段仅可在 BUSY=0 时进行修改

6.4.7 QSPI 数据寄存器（QSPI_DR）

地址偏移：0x18

复位值：0x0000_0000



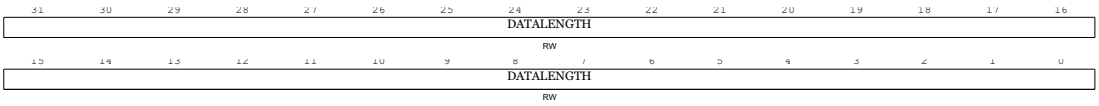
位/位域	名称	描述
31:0	DATA	数据

位/位域	名称	描述
		该字段为 QSPI Flash 通信数据 在间接写入模式下，写入该字段的数据将发送至 QSPI Flash； 在间接读取模式下，读取该字段可获得从 QSPI Flash 中接受的数据； 在轮询模式下，该字段为从 QSPI Flash 中读取的数据。

6.4.8 QSPI 数据长度寄存器（QSPI_DLR）

地址偏移：0x1C

复位值：0x0000_0000

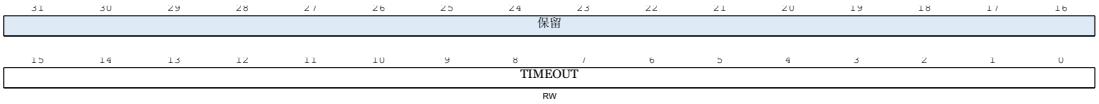


位/位域	名称	描述
31:0	DATALENGTH	数据长度 在间接模式和轮询模式下，该字段定义了与 QSPI Flash 通信时需要传输的字节数。其中，轮询模式下，该字段最大设置为 3（传输 4 字节数据）。 0x0000_0000：传输 1 个字节 0x0000_0001：传输 2 个字节 0x0000_0002：传输 3 个字节 0x0000_0003：传输 4 个字节 … 该字段仅可在 BUSY=0 时进行修改

6.4.9 QSPI 超时寄存器（QSPI_TOR）

地址偏移：0x20

复位值：0x0000_0014

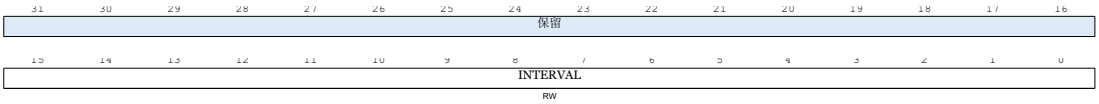


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TIMEOUT	超时时长 该字段定义了在内存映射模式下，FIFO 写满后，等待多少时钟周期后拉高 nCS 信号。 该字段仅可在 BUSY=0 时进行修改

6.4.10 QSPI 轮询间隔寄存器（QSPI_PIR）

地址偏移：0x24

复位值：0x0000_0000

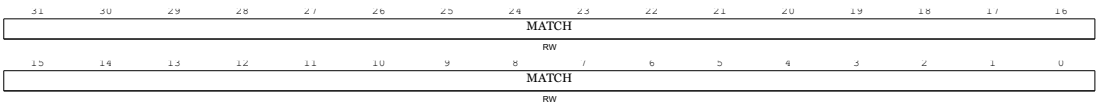


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	INTERVAL	轮询间隔 该字段定义了自动轮询模式下，读取操作之间的时钟间隔 该字段仅可在 BUSY=0 时进行修改

6.4.11 QSPI 轮询状态匹配寄存器（QSPI_PSMAR）

地址偏移：0x28

复位值：0x0000_0000

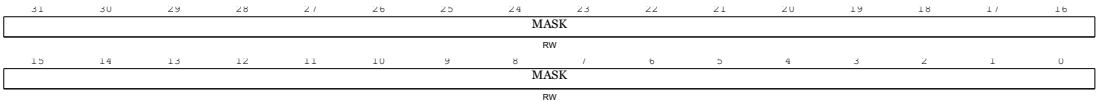


位/位域	名称	描述
31:0	MATCH	轮询状态匹配 在轮询模式下，该字段与 QSPI 传输数据进行比较（多用于比较 QSPI 状态寄存器数值） 该字段仅可在 BUSY=0 时进行修改

6.4.12 QSPI 轮询状态屏蔽寄存器（QSPI_PSMKR）

地址偏移：0x2C

复位值：0x0000_0000



位/位域	名称	描述
31:0	MASK	轮询状态屏蔽 在轮询模式下，该字段用于接收字段的屏蔽 对于位 n： 0：屏蔽在自动轮询模式下接收数据的 n 位，在比较中不考虑该位数值 1：不屏蔽在自动轮询模式下接收数据的 n 位，在比较中考虑该位数值 该字段仅可在 BUSY=0 时进行修改

7 系统管理模块（SMU）

7.1 简介

SMU 模块的主要功能是完成时钟和复位的管理。在默认状态下 SMU 工作在 IDLE 状态。只有接收到 PMU 的使能信号后才开始工作。SMU 模块会根据 PMU 的指令自动配置 COR、AXIBUS0/1/2 等总线的时钟和复位。

7.2 特征

- 支持系统时钟的切换（可以工作在 OSC 时钟、FIRC 时钟和 PLL 时钟）
- 支持总线和外设复位的单独管理
- 支持总线和外设时钟的单独使能管理
- 支持对 PLL 的单独配置
- 支持独立的总线分频器
- 支持根据 PMU 的指令自动配置 CORE 等时钟和复位

7.3 功能说明

7.3.1 时钟管理

在默认状态下，SMU 会关闭 CORE 的时钟使能，并打开 AxiBus0/1/2 和 AxiLiteBus0 的总线时钟。eFlash 控制器有时钟后开始对 eFlash 进行初始化工作。如果接收到 PMU 的上电启动指令或睡眠唤醒指令后，SMU 会自动检测 eFlash 初始化完成信号、PLL 锁定信号和 SRAM 初始化完成信号。然后开启 CORE 时钟使能。如果接收到 PMU 的进入睡眠指令或进入深度睡眠指令后，SMU 会自动关闭 CORE 时钟使能和 AxiBus0/1/2 和 AxiLiteBus0 的总线时钟使能。如果接收到用户的更新 PLL 指令后，SMU 会自动根据 PLL 的配置项配置 PLL。 为了保证系统稳定运行，只有在工作模式切换时才会更新系统时钟配置。外设时钟的关闭和使能用户可以随时切换。更新系统时钟流程如下图所示：

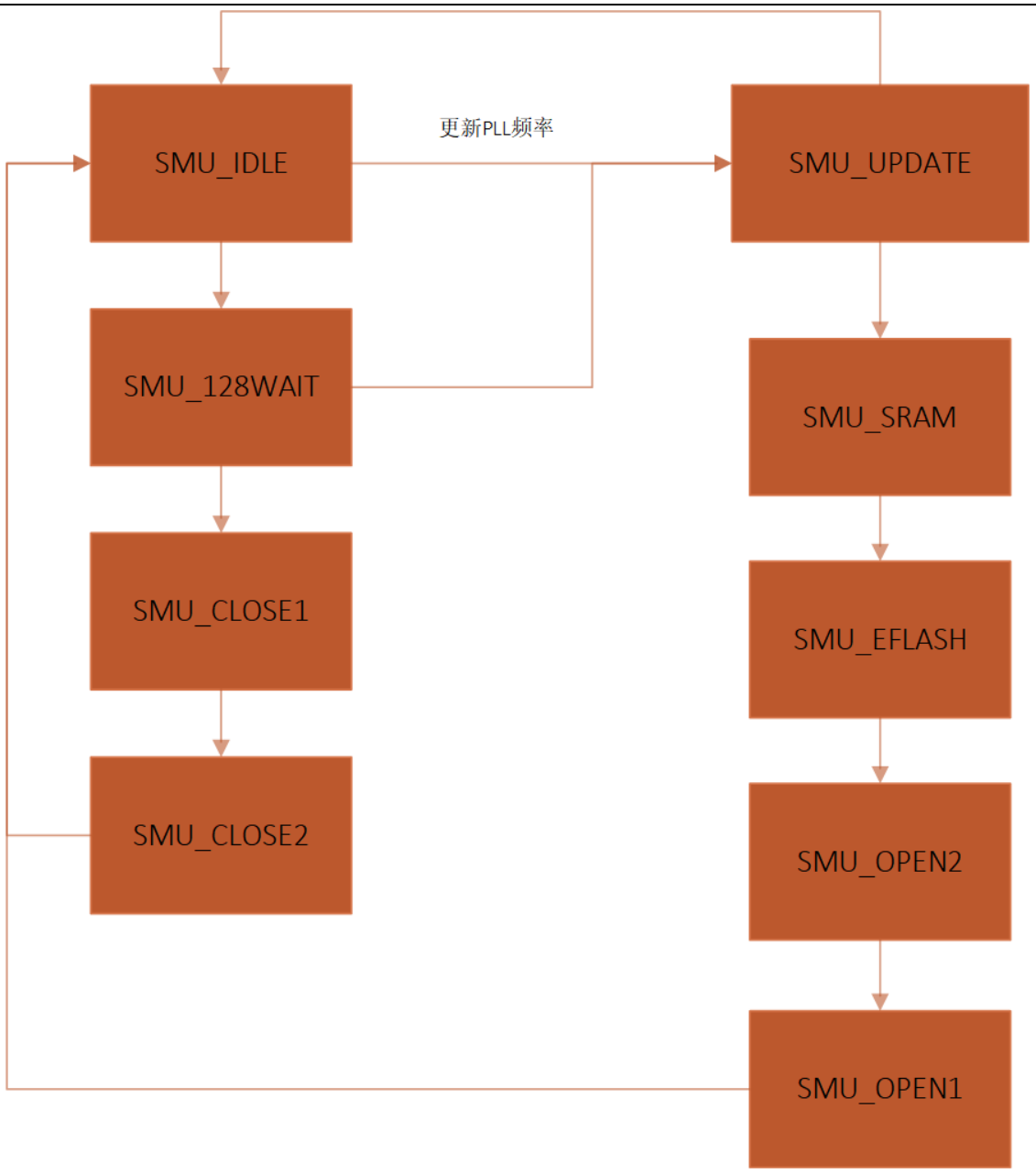


图 7.1 SMU 的状态转移图

表 7.1 SMU 状态说明

状态名称	状态说明
SMU_IDLE	SMU 的空闲状态。在此状态 SMU 会检测 AON 模块发起的工作模式信号，以及 PLL 更新使能信号。然后进行相应的状态
SMU_128WAIT	为了确保 CORE 访问外设完成，等待 128 个周期
SMU_CLOSE1	关闭 CORE 时钟

状态名称	状态说明
SMU_CLOSE2	关闭 AxiBus0/1/2 总线时钟 AxiLiteBus0 时钟。用户需要关闭其他总线总线和外设时钟。在完成时钟关闭后会给 PMU 一个完成信号
SMU_UPDATE	修改 PLL 的配置，修改总线分频器的配置在使能了 PLL 后需要检测 PLL 锁定信
SMU_SRAM	等待 SRAM 初始化完
SMU_EFLASH	MCU 的启动方式有 EFLASH、RAM 和 QSPI。在 EFLASH 启动时，需要等待其初始化完成。在 EFLASH 未初始化完成前 CORE 无法读取指令，所以需要等待其初始化完成。
SMU_OPEN2	使能 AxiBus0/1/2、AxiLiteBus0 总线时钟。
SMU_OPEN1	使能 CORE 时钟。如果是进入省电模式或者正常模式，会给 PMU 一个完成信

7.3.1.1 时钟频率范围

在 AS32A601 中，每个时钟都具有设计上的频率范围，超过范围的时钟值将不受支持。

表 7.2 内核和总线工作频率范围说明

时钟	频率下限(MHz)	频率上限(MHz)
coreClk	16	180
axi4Bus1Clk	8	90
axi4Bus3Clk	8	90
apbBus0Clk	4	45
apbBus1Clk	4	45
can	8	80
ADC	1	45
DAC	1	45

7.3.2 复位管理

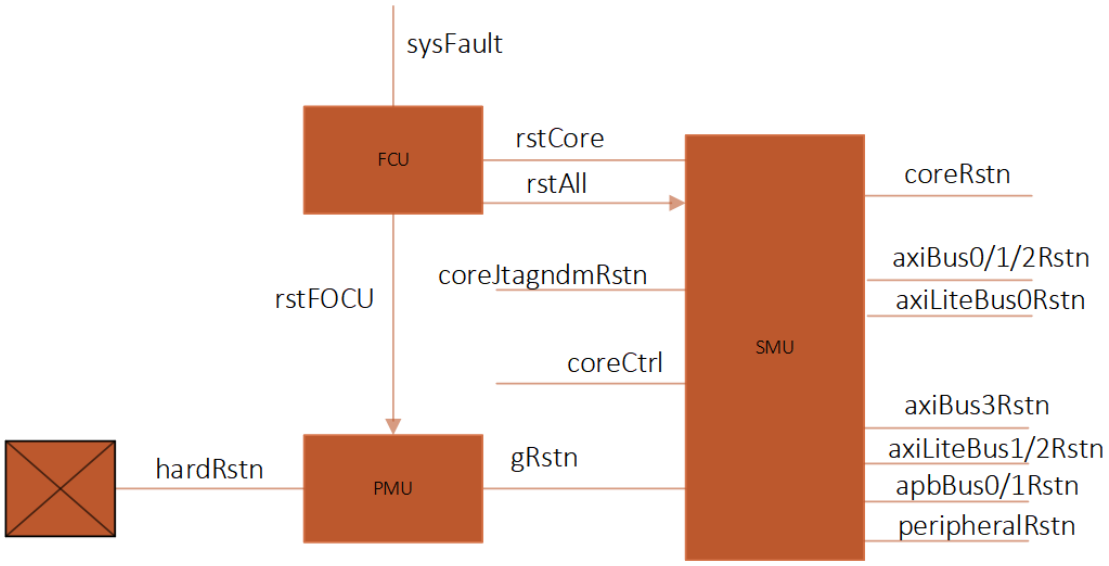


图 7.2 系统复位树示意框图

表 7.3 SMU 复位信号说明

信号名称	信号说明	备注
hardRstn	硬件复位信号，由专用引脚输入	所有模块都复位
rstFOCU	FCU 检测到系统验证严重错误后发出的复位信号	复位出 AON 外的所有模块
rstAll	CORE 接收到系统错误时会配置 FCU 发送复位信号	复位除 SMU/AON 的所有模块（高脉冲有效）
gRstn	PMU 检测到 hardRstn 后会触发 gRstn	复位除 AON 的所有模块
coreJtagndmRstn	硬件 Debug 时可以通过 JTAG 产生该复位信号	复位除 SMU/AON 的所有模块（高有效）
coreCtrl	CORE 通过总线配置 SMU 的方式产生复位信号	可以复位 AxiBus3、AxiLiteBus1/2、APBBus0/1 和所以外设

信号名称	信号说明	备注
AxiBus0/1/2Rstn、axiLiteBus0Rstn	只有 rstAll，gRstn，coreJtagndmRstn 有效（复位）时才产生复位信号	
axiBus3Rstn	由 CORE 通过配置寄存器的方式复位这两个总线	
AxiLiteBus1/2Rstn	由 CORE 通过配置寄存器的方式复位这两个总线	
axiBus3Rstn	由 CORE 通过配置寄存器的方式复位这两个总线	
apbBus0/1Rstn	由 CORE 通过配置寄存器的方式复位这两个总线	
peripheralRstn	由 CORE 通过配置寄存器的方式复位这两个总线	

注 1：在对外设进行读写时必须要保证上一级（或上上级），以及本设备所在总线的时钟使能有效，且复位处于无效状态。例如：访问 HTIM2 时，必须要保证 AxiBus2、AxiBus3、Axi 转 APB 桥以及 APBBus0 模块上有时钟，且释放复位。因为 CORE 会先访问 AxiBus2，然后访问 AxiBus3，然后访问 APBBus0，最后把读写数据时序传到 HTIM2。

注 2：为了保证用户能读取到复位的源头，在 rstAll 复位有效时不复位 FCU 模块。如果如需复位，需要单独处理。

7.4 应用说明

7.4.1 配置 PLL 流程说明

PLL 的输入源有：OSC 和 FIRC（默认）。可以通过配置 SMU_BUSCLKDIVREG 寄存器中的 pllClkSel 字段来选择 PLL 的输入源。为了保证 PLL 正常运行需要保证 PLL 的输入时钟频率经 PLL_DIVN 分频后的范围在

0.95MHz 到 2.1MHz 之间。经过分频后的时钟给到 VCO 进行向上变频。通过 PLL_DIVF 确定变频的值，其值必须在 100MHz 和 480MHz 之间。PLL 输出的时钟频率有 PLL_DIVQ 和 PLL_DIVR 确定。其范围说明参考 SMU_PLLCFGREG 寄存器。PLL 的典型配置值可以参考下表：

表 7.4 PLL 配置参考表

PLLIN	PLL_CLK_Q	PLL_CLK_R	PLL_DIVN	PLL_DIVF	PLL_DIVQ	PLL_DIVR
16MHz	180MHz	60MHz	16	360	2	6
16MHz	160MHz	80MHz	16	320	2	4
16MHz	160MHz	80MHz	16	160	1	2
16MHz	60MHz	60MHz	16	240	4	4
16MHz	60MHz	60MHz	16	120	2	2

用户可根据具体需求配置 SMU_PLLCFG 寄存器，实现更新 PLL 状态。如关闭 PLL，切换 PLL 输出频率等。

7.4.2 切换系统时钟流程说明

系统的工作时钟可以通过配置 SMU_BUSCLKDIV 寄存器中的 sysClkSel 字段来切换。系统工作可以工作在 OSCIN 时钟、FIRC 时钟和 PLL 时钟。默认状态下 sysClkSel 的值为 0，也即选择 FIRC 作为系统工作时钟。用户切换时钟时需要判断相应的状态，以保证系统时钟切换成功。

1. 如果用户想把系统时钟切换到 OSC 时钟，需要确定芯片外接晶振；
2. 如果用户想把系统时钟切换到 FIRC 时钟，需要确保 SMU_FIRCCFG 寄存器的 fircRdy 位为 1；
3. 如果用户想把系统时钟切换到 PLL 时钟，需要确保 SMU_STATUS 寄存器的 pllLock 位为 1。

例：如果当前系统工作时钟为 PLL 输出的 80MHz。然后切换到 PLL 输出的 180MHz

1. 判断 SMU_FIRCCFG 寄存器的 fircRdy 位为 1，然后把系统工作时钟切换到 FIRC（配置 SMU_BUSCLK 寄存器的 sysClkSel 为 0x00）；

2. 如果 OSC 时钟有效，用户也可以配置 SMU_BUSCLK 寄存器选择 OSC 作为系统时钟；
3. 配置 PLL 的输出频率为 180MHz。然后等待 PLL 锁定（配置 SMU_PLLCFG 寄存器）；
4. 切换系统时钟到 PLL 输出的 180MHz。（配置 SMU_BUSCLKDIV 寄存器的 sysClkSel 为 0x01）

Warning

在改变 AxiBus3、AxiLiteBus1/2、APBBus0/1 的时钟频率时，需要先关闭总线时钟。等待分频后的时钟稳定后再开启总线时钟。

7.4.3 启动 CORE 的时钟配置流程说明

系统上电状态和深度唤醒状态下：

1. 用户发起睡眠唤醒（如果是上电状态，则不需要），PMU 会自动向 SMU 发起上电启动指令；
2. SMU 自动检测 PLL 锁定信号；
3. SMU 自动检测 SRAM 初始化完成信号；
4. 如果是 eFlash 启动，SMU 自动等到 eFlash 初始化完成信号；如果是 QSPI 启动，SMU 则会直接进入打开总线状态；
5. SMU 自动打开总线时钟，然后打开 CORE 时钟；
6. CORE 根据 BOOT 选择启动模式，运行应用程序。

睡眠状态下：

1. 用户发起睡眠唤醒（如果是上电状态，则不需要），PMU 会自动向 SMU 发起上电启动指令；
2. SMU 自动检测 PLL 锁定信号（此时必须要是能 PLL）；
3. SMU 自动检测 SRAM 初始化完成信号；
4. 如果是 eFlash 启动，SMU 自动等到 eFlash 初始化完成信号；如果是 QSPI 启动，SMU 则会直接进入打开总线状态；
5. SMU 自动打开总线时钟，然后打开 CORE 时钟；

6. CORE 根据 BOOT 选择启动模式，运行应用程序。

7.4.4 部分总线和外设的时钟开关说明

目前 CORE、AxiBus0/1/2 总线和 AxiLiteBus0 总线时钟是有 SMU 控制开启或关闭。而 AxiBus3、AxiLiteBus1/2、APBBus0/1 总线为用户控制开启或关闭（默认状态下，时钟处于关闭）。用户需要根据总线的隶属关系开启相应的时钟。以开启 USART0 的时钟的示例如下：

1. 配置 SMU_AB3CFG 寄存器，打开 AxiBus3 时钟使能；
2. 配置 SMU_AB3CFG 寄存器，打开 AXI4 转 APB 桥 0 的时钟使能；
3. 配置 SMU_APB0CFG 寄存器，打开 APBBus0 总线使能;
4. 配置 SMU_APB0CFG 寄存器，打开 USART0 时钟使能。

Note

总线的时钟使能的优先级要比外设的优先级高。也就是说，如果关闭总线时钟使能，外设的时钟也会被关闭。要同时开启总线时钟使能和外设时钟使能，该外设才会有时钟。

7.4.5 系统复位流程说明

用户可以通过 hardRstn 实现系统级复位，且优先级最高。

7.4.6 外设复位流程说明

用户可以通过配置寄存器控制 AxiBus3、AxiLiteBus1/2、APBBus0/1 总线以及其外设的复位。用户需要根据总线的隶属关系释放相应总线的复位。以释放 USART0 的复位为例：

1. 配置 SMU_AB3CFG 寄存器，释放 AxiBus3 的复位；
2. 配置 SMU_AB3CFG 寄存器，释放 AXI4 转 APB 桥 0 的复位
3. 配置 SMU_APB0CFG 寄存器，释放 APBBus0 总线复位；
4. 配置 SMU_APB0CFG 寄存器，释放 USART0 的复位。

Note

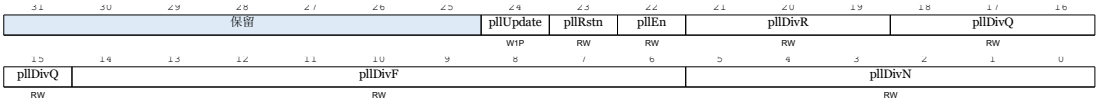
与总线时钟使能流程类似，想要外设正常工作，必须要释放相应总线的复位。

7.5 寄存器

7.5.1 SMU PLL 配置寄存器(SMU_PLLCFG)

地址偏移：0x0

复位值：0x00e15a10



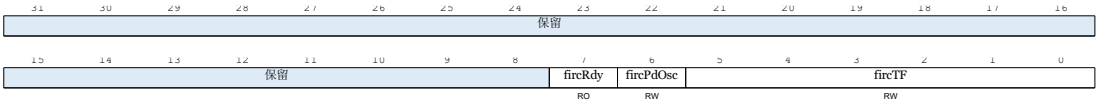
位/位域	名称	描述
31:25	保留	必须保持复位值
24	pllUpdate	更新 PLL 频率使能 0: 不更新 1: 更新 PLL 频率
23	pllRstn	PLL 时钟复位信号 0: 复位 PLL 1:释放 PLL 复位
22	pllEn	PLL 时钟使能 0: 不使能 PLL 1:使能 PLL
21:19	pllDivR	CLK_R 输出分频控制信号 PLL 输出的 R 时钟。其输出范围在 30MHz 到 240MHz 之间
18:15	pllDivQ	CLK_Q 的输出分频控制信号 PLL 输出的 Q 时钟。其输出范围在 16MHz 到 480MHz 之间
14:6	pllDivF	反馈时钟控制信号 反馈时钟，即 VCO 输出的时钟。其范围应控制在 100MHz 到 480MHz 之间
5:0	pllDivN	输入分频控制信号

位/位域	名称	描述
		输入时钟经过分频后应该在 0.95MHz 到 2.1MHz 之间

7.5.2 SMU FIRC 配置寄存器(SMU_FIRCCFG)

地址偏移：0x4

复位值：0x00000020

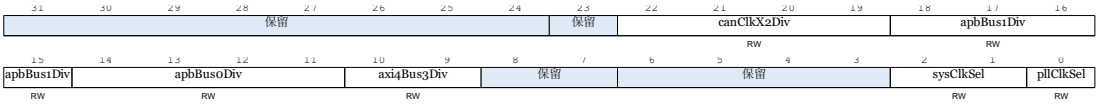


位/位域	名称	描述
31:8	保留	必须保持复位值
7	fircRdy	FIRC 振荡器准备位，高有效
6	fircPdOsc	FIRC 振荡器开关，低打开，高关闭
5:0	fircTF	FIRC 振荡器频率调整

7.5.3 SMU 总线时钟分频寄存器(SMU_BUSCLKDIV)

地址偏移：0x8

复位值：0x00091308



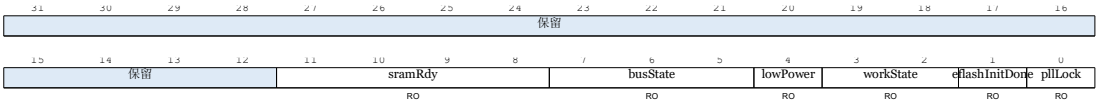
位/位域	名称	描述
31:24	保留	必须保持复位值
23	保留	必须保持复位值
22:19	canClkX2Div	Can 时钟分频比 4’b0001: 不分频 4’b0010: 时钟进行 2 分频 4’b0100: 时钟进行 4 分频 4’b1000: 时钟进行 8 分频 其他值：时钟进行 8 分频
18:15	apbBus1Div	ApbBus1 时钟分频比

位/位域	名称	描述
		4’b0001: 不分频 4’b0010: 时钟进行 2 分频 4’b0100: 时钟进行 4 分频 4’b1000: 时钟进行 8 分频 其他值：时钟进行 8 分频
14:11	apbBus0Div	ApbBus0 时钟分频比 4’b0001: 不分频 4’b0010: 时钟进行 2 分频 4’b0100: 时钟进行 4 分频 4’b1000: 时钟进行 8 分频 其他值：时钟进行 8 分频
10:9	axi4Bus3Div	Axi4Bus3 时钟分频比 2’b01: 不分频 2’b10: 时钟进行 2 分频 其他值：时钟进行 2 分频
8:7	保留	必须保持复位值
6:3	保留	必须保持复位值
2:1	sysClkSel	系统主时钟的选择 2’b00: FIRC 作为系统时钟 2’b01: PLL 输出的 CLK_Q 作为系统时钟 2’b10: OSC 作为系统时钟
0	pllClkSel	PLL 输入时钟选择 0: PLL 输入的时钟为 FIRC 时钟 1: PLL 输入的时钟为 OSC 时钟

7.5.4 SMU 状态寄存器(SMU_STATUS)

地址偏移：0xC

复位值：0x00000000



位/位域	名称	描述
31:12	保留	必须保持复位值
11:8	sramRdy	SRAM 准备就绪位。在 SRAM 从低功耗模式恢复时需要等待 SRAM 准备就绪才能对其读写 Bit3:SRAM3 准备就绪标志 Bit2:SRAM2 准备就绪标志 Bit1:SRAM1 准备就绪标志 Bit0:SRAM0 准备就绪标志 1:表示 SRAM 准备就绪，可以读写 0:表示 SRAM 未准备就绪，不可以读写
7:5	busState	Bus 总线状态 Bit2: AXI4Bus2 总线状态 Bit1: AXI4Bus1 总线状态 Bit0: AXI4Bus0 总线状态 0：表示总线空闲 1: 表示总线忙
4	lowPower	省电模式标志 此位只读。有硬件置位和清零 在用户配置 FIRC 作为系统为系统时钟时，即写 sysClkSel 为 00，此位自动置 1 在用户配置 PLL 输出时钟作为系统时钟时，即写 sysClkSel 为 01，此位自动清零 0:性能模式 1:省电模式

位/位域	名称	描述
3:2	workState	MCU 启动方式 2’b00:上电启动 2’b01:深度睡眠启动 2’b10:睡眠启动
1	eflashInitDone	EFLASH 初始化完成信号 0:初始化未完成 1:初始化完成
0	pllLock	PLL 时钟锁定位，高有效

7.5.5 SMU AXI 总线 0 配置寄存器(SMU_AB0CFG)

地址偏移：0x10

复位值：0x0000000f

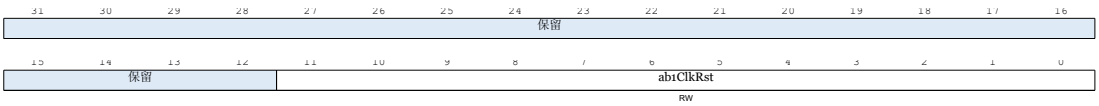
位/位域	名称	描述
31:6	保留	必须保持复位值
5:0	ab0ClkRst	AxiBus0 总线外设的时钟使能和复位信号 Bit5:EFLASH 时钟更新使能。EFLASH 控制器检测到上升沿时更新分频器 Bit4:QSPI 时钟更新使能。QSPI 控制器检测到上升沿时更新分频器 Bit3:EFLASH 控制器的复位 Bit2:EFLASH 控制器的时钟使能 Bit1:QSPI 控制器的复位 Bit0:QSPI 控制器的时钟使能 0:如果是复位信号，则表示复位有效

位/位域	名称	描述
		如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能

7.5.6 SMU AXI 总线 1 配置寄存器(SMU_AB1CFG)

地址偏移：0x14

复位值：0x00000fff



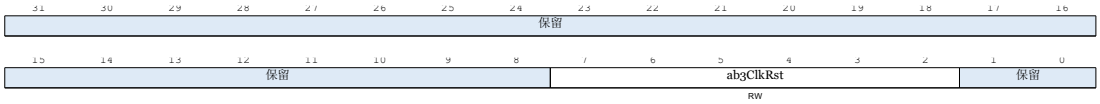
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	ab1ClkRst	AxiBus1 总线外设的时钟使能和复位信号 Bit11:SRAM3 的复位 Bit10:SRAM3 的时钟使能 Bit9:SRAM2 的复位 Bit8:SRAM2 的时钟使能 Bit7:SRAM1 的复位 Bit6:SRAM1 的时钟使能 Bit5:SRAM0 的复位 Bit4:SRAM0 的时钟使能 Bit3:DMA 控制器 1 的复位 Bit2:DMA 控制器 1 的时钟使能 Bit1:DMA 控制器 0 的复位 Bit0:DMA 控制器 0 的时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。

位/位域	名称	描述
		如果是时钟使能信号，则表示时钟使能

7.5.7 SMU AXI 总线 3 配置寄存器(SMU_AB3CFG)

地址偏移：0x18

复位值：0x000000aa

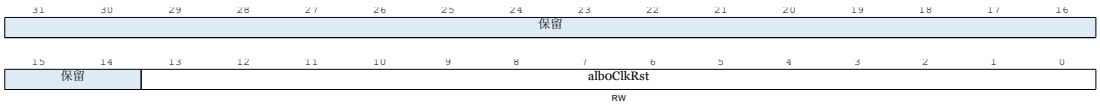


位/位域	名称	描述
31:8	保留	必须保持复位值
7:2	ab3ClkRst	AxiBus3 总线外设的时钟使能和复位信号 Bit7:AxiBus3 的复位 Bit6:AxiBus3 的时钟使能 Bit5:AXI4 转 APB 桥 1 的复位 Bit4:AXI4 转 APB 桥 1 的时钟使能 Bit3:AXI4 转 APB 桥 0 的复位 Bit2:AXI4 转 APB 桥 0 时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能
1:0	保留	必须保持复位值

7.5.8 SMU AXILITE 总线 0 配置寄存器(SMU_ALB0CFG)

地址偏移：0x1C

复位值：0x00002aaa

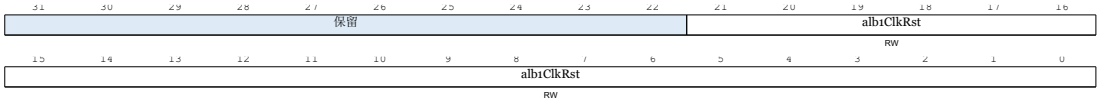


位/位域	名称	描述
31:14	保留	必须保持复位值
13:0	alb0ClkRst	AxiLiteBus0 总线外设的时钟使能和复位信号 Bit13: CLINT 的复位 Bit12: CLINT 的时钟使能 Bit11:PLIC 的复位 Bit10:PLIC 的时钟使能 Bit9:CMU3 的复位 Bit8:CMU3 的时钟使能 Bit7:CMU2 的复位 Bit6:CMU2 的时钟使能 Bit5:CMU1 的复位 Bit4:CMU1 的时钟使能 Bit3:CMU0 的复位 Bit2:CMU0 的时钟使能 Bit1:FCU 的复位 Bit0:FCU 的时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能

7.5.9 SMU AXILITE 总线 1 配置寄存器(SMU_ALB1CFG)

地址偏移：0x20

复位值：0x002aaaaa



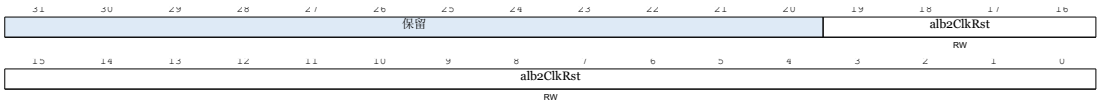
位/位域	名称	描述
31:22	保留	必须保持复位值
21:0	alb1ClkRst	AxiLiteBus1 总线外设的时钟使能和复位信号 Bit21:AxiLiteBus1 的复位 Bit20:AxiLiteBus1 的时钟使能 Bit19:DSE 的复位 Bit18:DSE 的时钟使能 Bit17:GPIOC 的复位 Bit16:GPIOC 的时钟使能 Bit15:GPIOB 的复位 Bit14:GPIOB 的时钟使能 Bit13:GPIOA 的复位 Bit12:GPIOA 的时钟使能 Bit11:DAC0 的复位 Bit10:DAC0 的时钟使能 Bit9:ADC1 的复位 Bit8:ADC1 的时钟使能 Bit7:ADC0 的复位 Bit6:ADC0 的时钟使能 Bit5:CANFD1 的复位 Bit4:CANFD1 的时钟使能 Bit3:CANFD0 的复位 Bit2:CANFD0 的时钟使能 Bit1:HTIM0 的复位 Bit0:HTIM0 的时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能

位/位域	名称	描述
		1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能

7.5.10 SMU AXILITE 总线 2 配置寄存器(SMU_ALB2CFG)

地址偏移：0x24

复位值：0x000aaaaa



位/位域	名称	描述
31:20	保留	必须保持复位值
19:0	alb2ClkRst	AxiLiteBus2 总线外设的时钟使能和复位信号 Bit19:AxiLiteBus2 的复位 Bit18:AxiLiteBus2 的时钟使能 Bit17:GPIOG 的复位 Bit16:GPIOG 的时钟使能 Bit15:GPIOF 的复位 Bit14:GPIOF 的时钟使能 Bit13:GPIOE 的复位 Bit12:GPIOE 的时钟使能 Bit11:GPIOD 的复位 Bit10:GPIOD 的时钟使能 Bit9:DAC1 的复位 Bit8:DAC1 的时钟使能 Bit7:ADC2 的复位 Bit6:ADC2 的时钟使能 Bit5:CANFD3 的复位 Bit4:CANFD3 的时钟使能

位/位域	名称	描述
		Bit3:CANFD2 的复位 Bit2:CANFD2 的时钟使能 Bit1:HTIM1 的复位 Bit0:HTIM1 的时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能

7.5.11 SMU APB 总线 0 配置寄存器(SMU_APB0CFG)

地址偏移：0x28

复位值：0xaaaaaaaa

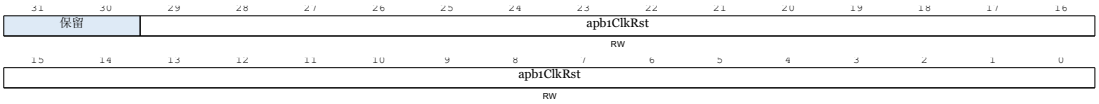
位/位域	名称	描述
31:0	apb0ClkRst	ApbBus0 总线外设的时钟使能和复位信号 Bit31:APBBus0 的复位 Bit30:APBBus0 的时钟使能 Bit29:28: 保留 Bit27:WWDG0 的复位 Bit26:WWDG0 的时钟使能 Bit25:TIM4 的复位 Bit24:TIM4 的时钟使能 Bit23:TIM3 的复位 Bit22:TIM3 的时钟使能 Bit21:HTIM2 的复位 Bit20:HTIM2 的时钟使能

位/位域	名称	描述
		Bit19:USART3 的复位 Bit18:USART3 的时钟使能 Bit17:USART2 的复位 Bit16:USART2 的时钟使能 Bit15:USART1 的复位 Bit14:USART1 的时钟使能 Bit13:USART0 的复位 Bit12:USART0 的时钟使能 Bit11:IIC1 的复位 Bit10:IIC1 的时钟使能 Bit9:IIC0 的复位 Bit8:IIC0 的时钟使能 Bit7:SPI2 的复位 Bit6:SPI2 的时钟使能 Bit5:SPI1 的复位 Bit4:SPI1 的时钟使能 Bit3:SPI0 的复位 Bit2:SPI0 的时钟使能 Bit1:MAC 的复位 Bit0:MAC 的时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能

7.5.12 SMU APB 总线 1 配置寄存器(SMU_APB1CFG)

地址偏移：0x2C

复位值：0x2aaaaaaa



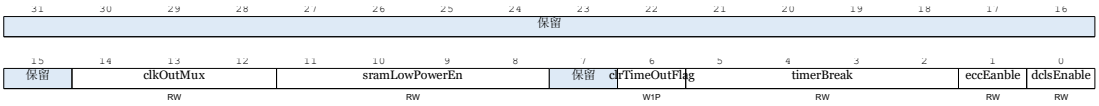
位/位域	名称	描述
31:30	保留	必须保持复位值
29:0	apb1ClkRst	ApbBus1 总线外设的时钟使能和复位信号 Bit29:APBBus1 的复位 Bit28:APBBus1 的时钟使能 Bit27:CRC 的复位 Bit26:CRC 的时钟使能 Bit25:WWDG1 的复位 Bit24:WWDG1 的时钟使能 Bit23:TIM7 的复位 Bit22:TIM7 的时钟使能 Bit21:TIM6 的复位 Bit20:TIM6 的时钟使能 Bit19:HTIM5 的复位 Bit18:HTIM5 的时钟使能 Bit17:USART7 的复位 Bit16:USART7 的时钟使能 Bit15:USART6 的复位 Bit14:USART6 的时钟使能 Bit13:USART5 的复位 Bit12:USART5 的时钟使能 Bit11:USART4 的复位 Bit10:USART4 的时钟使能 Bit9:IIC3 的复位 Bit8:IIC3 的时钟使能

位/位域	名称	描述
		Bit7:IIC2 的复位 Bit6:IIC2 的时钟使能 Bit5:SPI5 的复位 Bit4:SPI5 的时钟使能 Bit3:SPI4 的复位 Bit2:SPI4 的时钟使能 Bit1:SPI3 的复位 Bit0:SPI3 的时钟使能 0:如果是复位信号，则表示复位有效 如果是时钟使能信号，则表示时钟不使能 1:如果是复位信号，则表示复位无效。 如果是时钟使能信号，则表示时钟使能

7.5.13 SMU 其它项配置寄存器(SMU_OTHCFG)

地址偏移：0x30

复位值：0x00000003



位/位域	名称	描述
31:15	保留	必须保持复位值
14:12	clkOutMux	系统时钟经过 8 分频后输出 000: CoreClk 的 8 分频时钟 001:AxiBus1Clk 的 8 分频时钟 010:AxiBus3Clk 的 8 分频时钟 011:ApbBus0Clk 的 8 分频时钟 100:ApbBus1Clk 的 8 分频时钟 101:FircClk 的 8 分频时钟

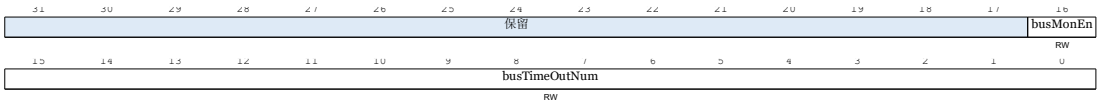
位/位域	名称	描述
		110:OscClk 的 8 分频时钟 111:SircClk
11:8	sramLowPowerEn	SRAM 低功耗使能信号。在省电模式下可以关闭部分 SRAM，以节省功耗。 Bit3:SRAM3 低功耗使能位 Bit2:SRAM2 低功耗使能位 Bit1:SRAM1 低功耗使能位 Bit0:SRAM0 低功耗使能位 1：SRAM 进入 RETENTION 模式。0：SRAM 工作在正常模式
7	保留	必须保持复位值
6	clrTimeOutFlag	清除超时标志。该标志可能有 PLL 或 ELFASH 未在规定时间内锁定或初始化完成。 该寄存器为写 1 产生脉冲寄存器 1:清除超时标志
5:2	timerBreak	高级定时器刹车使能 Bit3:TIMER5 的刹车使能信号 Bit2:TIMER2 的刹车使能信号 Bit1:TIMER1 的刹车使能信号 Bit0:TIMER0 的刹车使能信号 0:刹车使能无效 1:刹车使能有效 此位完全由软件控制
1	eccEanble	ECC 使能 0:旁路全局 ECC，1：使能全局 ECC
0	dclsEnable	双核使能

位/位域	名称	描述
		0:不是能双核 1: 使能双核

7.5.14 SMU 总线检测配置寄存器(SMU_BUSMON)

地址偏移：0x34

复位值：0x0001ffff



位/位域	名称	描述
31:17	保留	必须保持复位值
16	busMonEn	总线超时使能 1：使能总线超时功能； 0：不使能总线超时功能
15:0	busTimeOutNum	总线超时计数器 AxiBus0/1/2 工作该寄存器。当总线在规定时间内没有检测到响应，就会产生总线超时，并上报 FCU 注：该总线超时计数器的最小值为 30000

8 电源管理模块（PMU）

8.1 简介

电源管理模块 PMU（Power Management Unit）为软件提供了电源方面的系统管理功能，包含睡眠/深度睡眠/唤醒功能的控制，电源监测功能（LVR/LVD/HVD）。

8.2 特性

- 支持低功耗模式（SLEEP/DEEP_SLEEP/WAKEUP）的切换
- 支持唤醒源的配置（外部 IO/RTC/IWDG）
- 支持 FIRC 的开启与关闭

8.3 功能说明

8.3.1 睡眠模式

进入睡眠模式的流程如下图所示：

图 8.1 睡眠模式的工作流程示意图

在睡眠模式下，用户需要复位/禁止 PLL 输出、禁止 FIRC 输出时钟、ADC、DAC 等处于低功耗模式。

8.3.2 深度睡眠模式

进入深度睡眠模式的流程如下图所示：

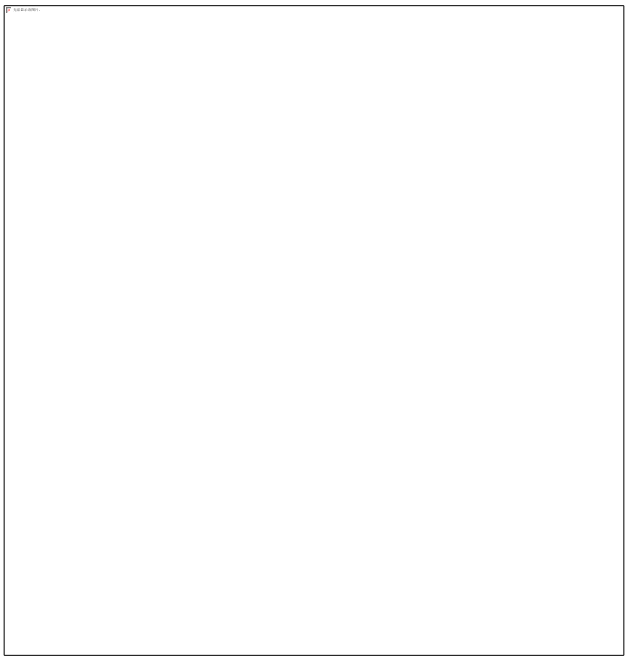


图 8.2 深度睡眠模式的工作流程示意图

在深度睡眠模式下，需要 PMU 关闭 PMB 输出，使 Main 和 ANA 进入掉电状态。此时唤醒只能通过 RTC 或外部引脚唤醒进入正常工作模式。

8.4 应用说明

8.4.1 内部高速振荡器（FIRC）电源控制

FIRC 默认处于开启状态。需要时，通过配置 PMU_CA_ENRC16M，打开或关闭。

8.4.2 电压检测（LVR/LVD/HVD）

8.4.3 LVR(Low Voltage Reset)低电压复位

- 当芯片 VDD 欠压至 LVR 阈值以下，触发 LVR 复位。
- LVR 触发电平 2.6V。 需要时，通过配置 PMU_VD_EN_LVR，打开或关闭 LVR 功能。

8.4.4 LVD（Low Voltage Detect）低电压检测

- 当芯片 VDD 欠压至 LVD 阈值以下，触发 LVD 错误，并通报给 FCU 处理。
- LVD 阈值可通过 PMU_VD_ADJ_LVD 进行配置。 需要时，通过配置 PMU_VD_EN_LVD，打开或关闭 LVD 功能。

8.4.5 HVD（High Voltage Detect）低电压检测

- 当芯片 VDD 过压至 HVD 阈值以上，触发 HVD 错误，并通报给 FCU 处理。
- HVD 阈值可通过 PMU_VD_ADJ_HVD 进行配置。 需要时，通过配置 PMU_VD_EN_HVD，打开或关闭 HVD 功能。

8.4.6 电源模式切换

要进入睡眠/深度睡眠模式，请遵循以下步骤：

1. 读取唤醒信息：读取 PMU_WKP 寄存器，若有位为 1 则写 1 清除该位，若皆为 0 则进入下一步操作；
2. 配置唤醒通道：将 PMU_WKEN 寄存器的对应唤醒通道配置为 1，其他通道配置为 0；
3. 保存备份信息：将需要备份的信息存储到 Bkp RAM 内；
4. 配置睡眠模式：配置 PMU_MODE 寄存器的值，配置 1 进入深度睡眠模式，配置 2 进入睡眠模式；
5. 系统将自动睡眠。 唤醒后，可通过 PMU_WKP 寄存器读取唤醒信息，获取唤醒源。

8.4.7 唤醒通道说明

唤醒通道根据下表进行连接：

表 8.1 唤醒通道表

唤醒通道	唤醒功能
0	IWDG

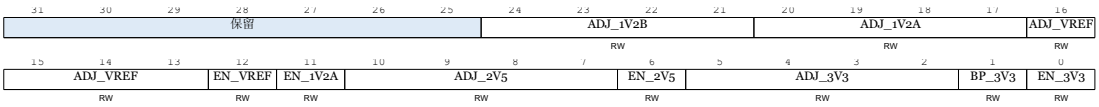
唤醒通道	唤醒功能
1~2	保留
3	RTC
4~13	PH0~PH9

8.5 寄存器

8.5.1 PMU 电源调节寄存器（PMU_PW）

地址偏移：0x0

复位值：0x01911420



位/位域	名称	描述
31:25	保留	必须保持复位值
24:21	ADJ_1V2B	1.2V 备份 LDO 电压调节 0000:1.04V 0001:1.06V … … 1000:1.20V(默认) … … 1111:1.34V
20:17	ADJ_1V2A	1.2V LDO 电压调节 0000:1.04V 0001:1.06V … … 1000:1.20V(默认) … …

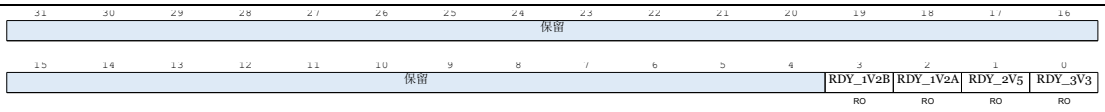
位/位域	名称	描述
		1111:1.34V
16:13	ADJ_VREF	1.2V 参考电压源调节 0000:1.04V 0001:1.06V … … 1000:1.20V(默认) … … 1111:1.34V
12	EN_VREF	1.2V Flash 参考电压源使能 1:使能 1.2V 参考电压源 0:关闭 1.2V 参考电压源
11	EN_1V2A	1.2V LDO 使能 1:使能 1.2V LDO 0:关闭 1.2V LDO
10:7	ADJ_2V5	2.5V LDO 电压调节 0000:2.18V 0001:2.22V … … 1000:2.50V(默认) … … 1111:2.785V

位/位域	名称	描述
6	EN_2V5	2.5V LDO 使能 1:使能 2.5V LDO 0:关闭 2.5V LDO
5:2	ADJ_3V3	3.3V LDO 电压调节 0000: 2.90V 0001: 2.95V … … 1000: 3.30V(默认) … … 1111: 3.65V
1	BP_3V3	3.3V LDO Bypass 1:将 VDD 直接接到 3.3V 电源域上，在 3.3V 电源输入时使用 0:将 3.3V LDO 的输出接到 3.3V 电源域上，在 5V 电源时使用
0	EN_3V3	3.3V LDO 使能 1:使能 3.3V LDO 0:关闭 3.3V LDO

8.5.2 PMU 电源状态寄存器（PMU_PS）

地址偏移：0x4

复位值：0x00000000

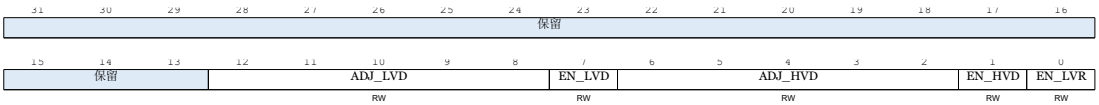


位/位域	名称	描述
31:4	保留	必须保持复位值
3	RDY_1V2B	1.2V 备份电源就绪标志位 1:备份 1.2V LDO 就绪 0:备份 1.2V LDO 未就绪
2	RDY_1V2A	1.2V 通用 LDO 就绪标志位 1:1.2V LDO 就绪 0:1.2V LDO 未就绪
1	RDY_2V5	2.5V LDO 就绪标志位 1:2.5V LDO 就绪 0:2.5V LDO 未就绪
0	RDY_3V3	3.3V LDO 就绪标志位 1:3.3V LDO 就绪 0:3.3V LDO 未就绪

8.5.3 PMU 电源监测寄存器（PMU_VD）

地址偏移：0x8

复位值：0x0000057c



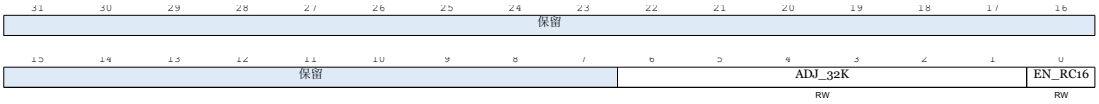
位/位域	名称	描述
31:13	保留	必须保持复位值
12:8	ADJ_LVD	低压检测阈值电压调节

位/位域	名称	描述
		00000:2.4V 00001:2.5V … … 00101:2.9V（默认） … … 11111:5.5V
7	EN_LVD	低压检测使能 1:低压检测启动 0:低压检测关闭
6:2	ADJ_HVD	高压检测阈值电压调节 00000:2.4V 00001:2.5V … … 11111:5.5V（默认）
1	EN_HVD	高压检测使能 1:高压检测启动 0:高压检测关闭
0	EN_LVR	低压复位使能 1:低压复位检测启动 0:低压复位检测关闭

8.5.4 PMU 振荡器调节寄存器（PMU_CA）

地址偏移：0xC

复位值：0x00000041

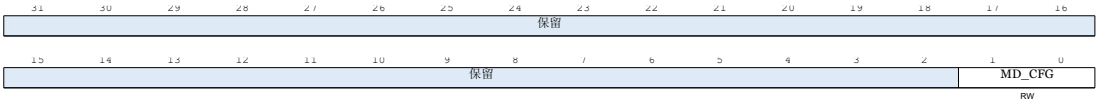


位/位域	名称	描述
31:7	保留	必须保持复位值
6:1	ADJ_32K	RC32K 时钟调节 000000:22.282KHz 000001:22.609KHz … … 100000:32.768KHz（默认） … … 111111:42.926KHz
0	EN_RC16	内部 16M 晶振使能 1:内部 16M 晶振启动 0:内部 16M 晶振关闭

8.5.5 PMU 模式寄存器（PMU_MODE）

地址偏移：0x10

复位值：0x00000000



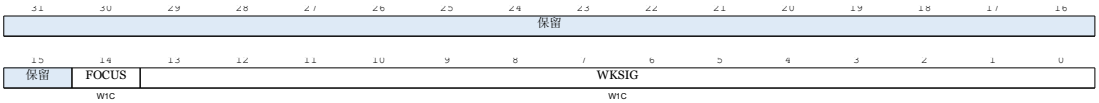
位/位域	名称	描述
31:2	保留	必须保持复位值
1:0	MD_CFG	电源模式配置

位/位域	名称	描述
		0:启动初始模式 1:深度睡眠模式 2:睡眠模式 3:普通模式 注：不要手动写入 0 或 3

8.5.6 PMU 唤醒寄存器（PMU_WKP）

地址偏移：0x14

复位值：0x00000000

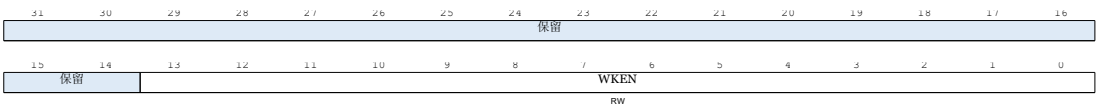


位/位域	名称	描述
31:15	保留	必须保持复位值
14	FOCUS	FOCU 复位标志位 1:上次复位为 FOCU 复位 0:上次复位不为 FOCU 复位 写入 1 清零
13:0	WKSIG	睡眠唤醒位，写 1 清除 通道信息请查看”唤醒通道说明”

8.5.7 PMU 唤醒使能寄存器（PMU_WKEN）

地址偏移：0x18

复位值：0x00000000



位/位域	名称	描述
31:14	保留	必须保持复位值
13:0	WKEN	睡眠唤醒通道使能位 1:使能对应唤醒通道 0:关闭对应唤醒通道 通道信息请查看”唤醒通道说明”

9 调试（DEBUG）

9.1 简介

AS32A601 基于 RISC-V DEBUG Spec v0.13.2 文档标准的调试器。外部调试器通过调试接口访问寄存器和存储器，控制程序运行/停止/复位等操作。

9.2 特性

- 基于 RISC-V DEBUG Spec v0.13.2 文档标准的 JTAG 接口
- 16 个硬件断点
- 调试安全性

9.3 功能描述

9.3.1 调试安全性

使能 EFlash 读保护后，调试端口的功能将受限，Debug 模块不能访问整个 MCU 资源，这样能有效保护 EFlash 数据。

9.3.2 低功耗调试

睡眠模式/深度睡眠模式下会断开 Debug 连接，不支持调试访问寄存器和内存。

10 内核本地中断控制器（CLINT）

10.1 简介

CLINT 模块产生 CSR 寄存器的定时器中断、软件中断和计时器。

10.2 主要特征

- AXILite 接口
- 支持定时器中断、软件中断和计时器输出

10.3 结构框图

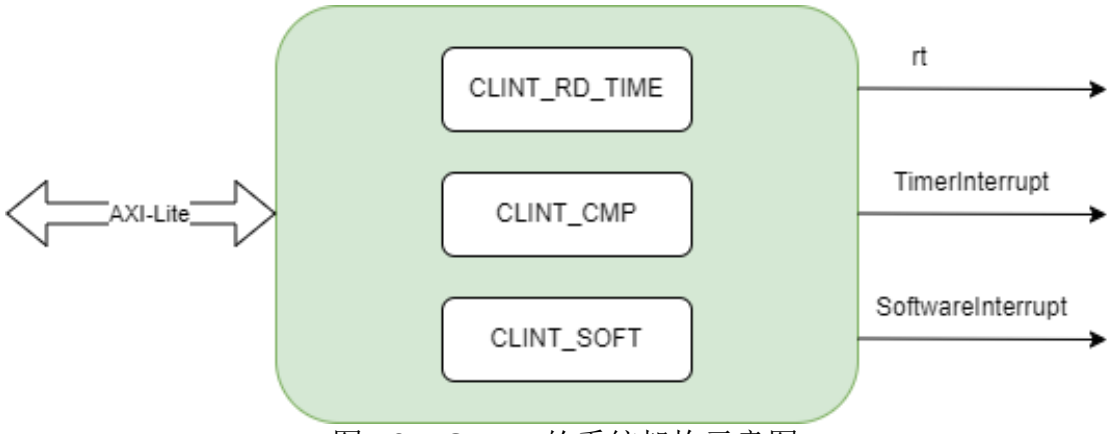


图 10.1 CLINT 的系统架构示意图

10.4 功能说明

CLINT 由软件中断触发器、定时器中断触发器、计数器组成。 软件中断触发器由一个寄存器构成，直连 CPU 的机器级软件中断。 计数器负责提供 CPU 的 TIME 与 TIMEH，并参与定时器中断的触发。 定时器中断触发器用于定时器中断的触发，当计数器值大于配置的值时会触发定时器中断。

10.5 应用说明

10.5.1 软件中断

1. 配置好 CPU 中断相关配置，并打开软件中断；
2. CLINT_SOFT 寄存器 SOFT 位写入 1，触发软件中断；

3. 触发过后，CLINT_SOFT 寄存器的 SOFT 位写入 0，清除软件中断。

10.5.2 定时器中断

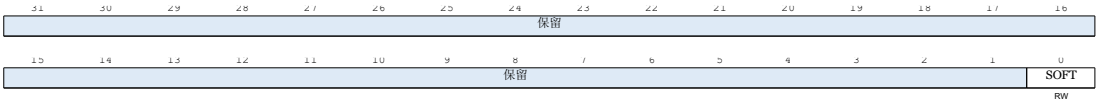
1. 配置好 CPU 中断相关配置，并打开定时器中断；
2. 分别将触发值写入 CLINT_CMP_L 与 CLINT_CMP_H。

10.6 寄存器

10.6.1 CLINT 软中断控制寄存器（CLINT_SOFT）

地址偏移：0x0000

复位值：0x00000000

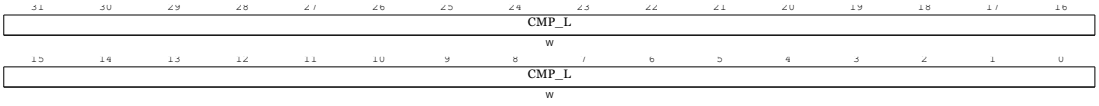


位/位域	名称	描述
31:1	保留	必须保持复位值
0	SOFT	软中断触发位 0：清除中断 1：触发中断

10.6.2 CLINT 比较寄存器低 32 位（CLINT_CMP_L）

地址偏移：0x4000

高低 32 位两个寄存器都需要配置;

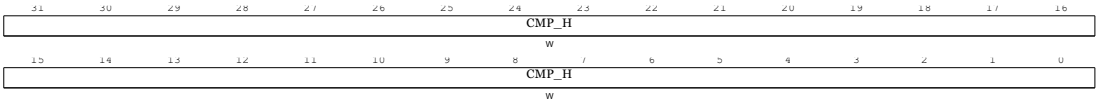


位/位域	名称	描述
31:0	CMP_L	比较器值低 32 位，与 CMP_H 共同组成 CMP。 当实时计数器大于等于 CMP 时产生定时器中断，重新配置大于计数器的 CMP 清除中断

10.6.3 CLINT 比较寄存器高 32 位（CLINT_CMP_H）

地址偏移：0x4004

高低 32 位两个寄存器都需要配置;

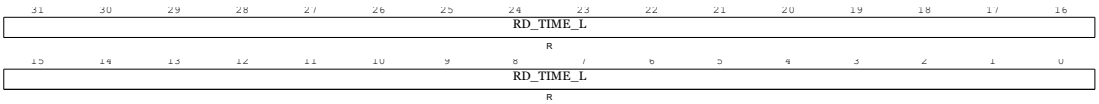


位/位域	名称	描述
31:0	CMP_H	比较器值高 32 位，与 CMP_L 共同组成 CMP。 当实时计数器大于等于 CMP 时产生定时器中断，重新配置大于计数器的 CMP 清除中断

10.6.4 CLINT 实时计数器低 32 寄存器（CLINT_RD_TIME_L）

地址偏移：0xBFF8

复位值：0x00000000

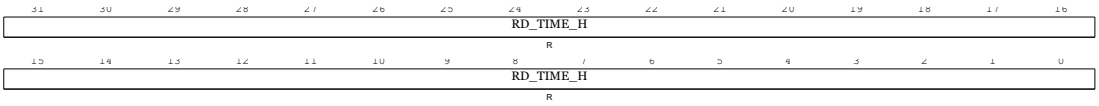


位/位域	名称	描述
31:0	RD_TIME_L	计数器低 32 位 计数器为 64 位的寄存器，读时需要分地址分别读取高 32 位和低 32 位 复位后一直递增,并循环计数

10.6.5 CLINT 实时计数器高 32 寄存器（CLINT_RD_TIME_H）

地址偏移：0xBFFC

复位值：0x00000000



位/位域	名称	描述
31:0	RD_TIME_L	计数器高 32 位

11 平台中断控制器（PLIC）

11.1 简介

平台中断控制器（Platform Level Interrupt Controller，PLIC）是 CPU 的中断控制器，主要对中断源进行采样，优先级仲裁和分发。各外设中断统一连到 PLIC，PLIC 统一管理并输出中断请求到内核。

11.2 主要特点

- 支持 64 个中断源
- 每个中断源分配一个 ID，从 1 开始，0 代表无中断
- 支持设置每个中断源优先级，共计 64 级，可以设置为相同优先级，63 级最高，0 级最低，优先级一样时，ID 号的先执行
- 支持设置中断使能位，注意区别于接口私有的中断使能，这两种需要全部开启
- 输出单周期中断请求到内核，相同时间只有一个最高优先级中断输出，执行中断函数时不响应相同中断
- 中断源上升沿触发和高电平触发
- 支持中断嵌套（保存现场和恢复现场时不支持中断嵌套）
- 软件方式保存恢复现场

11.3 设计框图

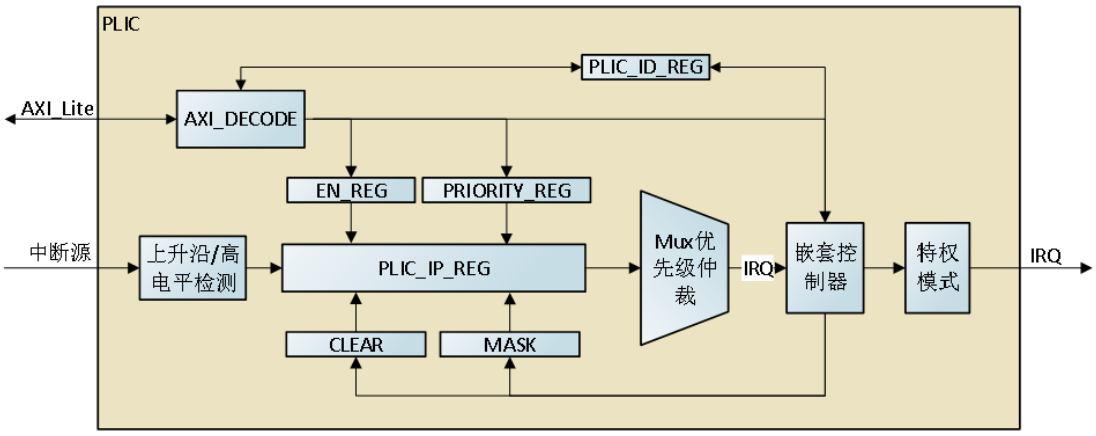


图 11.1 PLIC 架构示意图

PLIC 主要进行中断源的派发，当有多个中断源时，选出优先级最高的中断，并更新 ID 寄存器为优先级最高的中断源的 ID，之后硬件自动使能对应的中断源的屏蔽位，防止再次触发，并挂起对应的 IP 位，当用户写此 ID 时，表示中断函数结束，硬件自动清零 IP 位，并清零屏蔽位，之后可以再次响应相同中断。

以下为中断处理流程：

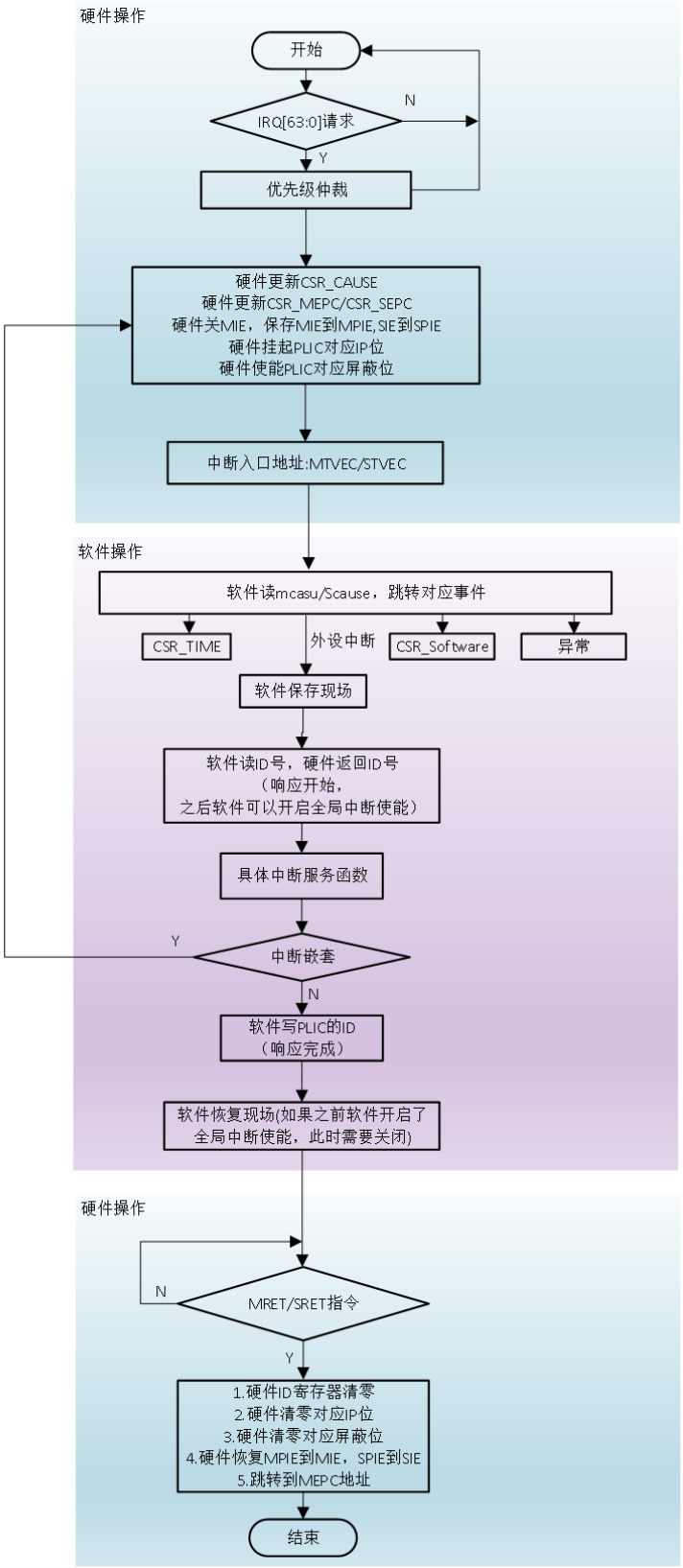


图 11.2 中断处理流程示意图

11.4 中断 ID 表

中断向量如下表所示：

表 11.1 中断向量表

中断源标号	中断源
1	EFLASH
2	FCU
3	DMA0
4	DMA1
5	TIM0
6	CANFD0
7	CANFD1
8	ADC0
9	ADC1
10	GPIOA
11	GPIOB
12	GPIOC
13	PKA
14	SPACC
15	TRNG
16	TIM1
17	CANFD2
18	CANFD3
19	ADC2
20	GPIOD
21	GPIOE
22	GPIOF
23	GPIOG
24	MAC
25	SPI0

中断源标号	中断源
26	SPI1
27	SPI2
28	IIC0
29	IIC1
30	USART0
31	USART1
32	USART2
33	USART3
34	TIM2
35	TIM3
36	TIM4
37	SPI3
38	SPI4
39	SPI5
40	IIC2
41	IIC3
42	USART4
43	USART5
44	USART6
45	USART7
46	TIM5
47	TIM6
48	TIM7
49	GPIOH
50	RTC
51	NONE
52	NONE
53	NONE
54	NONE

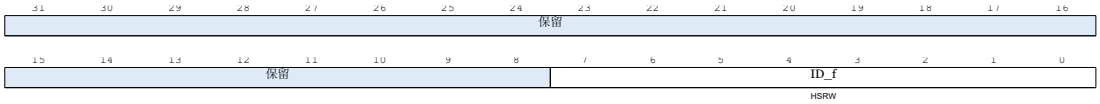
中断源标号	中断源
55	NONE
56	NONE
57	NONE
58	NONE
59	NONE
60	NONE
61	NONE
62	NONE
63	NONE

11.5 寄存器

11.5.1 PLIC ID 寄存器(PLIC_ID)

地址偏移：0x0

复位值：0x00000000

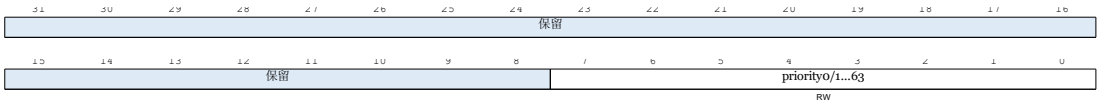


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	ID_f	中断 ID 号

11.5.2 PLIC 优先级寄存器(PLIC_PRIORITY)

地址偏移：0x4 0x8 … 0x100

复位值：0x00000000

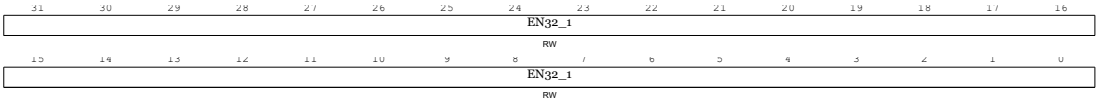


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	priority0/1…63	第 1/2…64 中断优先级

11.5.3 PLIC 使能寄存器 0(PLIC_EN0)

地址偏移：0x104

复位值：0x00000000

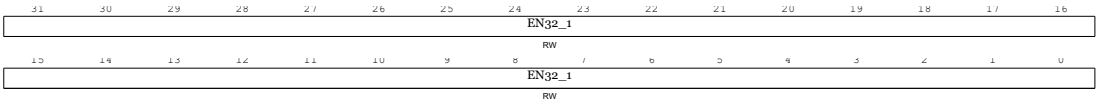


位/位域	名称	描述
31:0	EN32_1	第 32~1 的中断使能

11.5.4 PLIC 使能寄存器 1(PLIC_EN1)

地址偏移：0x108

复位值：0x00000000



位/位域	名称	描述
31:0	EN32_1	第 64~33 的中断使能

12 功能安全综述

12.1 简介

AS32A601 是根据 GB/T 34590 与 ISO 26262 标准开发的，具有针对 ISO26262 ASIL-B 完整性级别的综合安全概念。

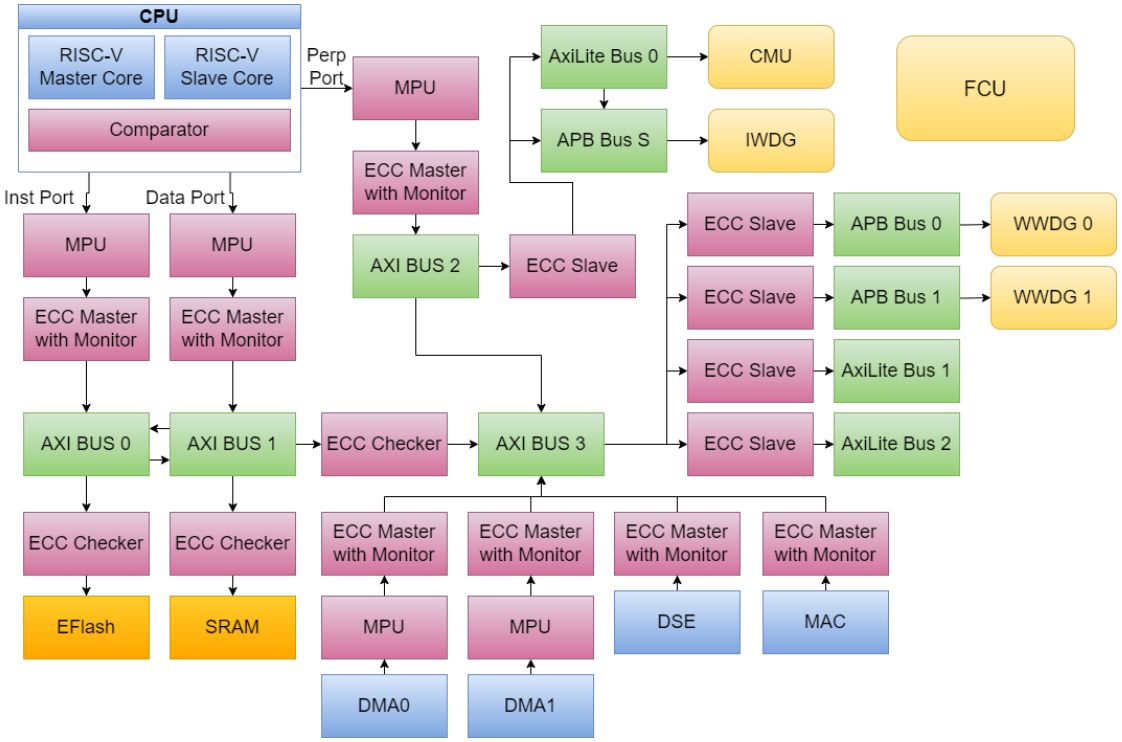


图 12.1 功能安全结构框图

12.2 安全描述

12.3 双核锁步（DCLS）

AS32A601 的 CPU 与 DMA 采用双核锁步设计，对其对总线接口/RAM 的访问进行监控与比较，用于检测其中产生的错误，在内核域满足 ASIL-D 的安全要求。

12.3.1 MPU

对于需要满足 ASIL-B 的应用程序，AS32A601 提供了多个内存保护单元（MPU）用于访问保护。MPU 负责分配访问权限并确保只有授权的访问单元才可以访问对应资源。

更多细节请参考内存保护单元（MPU）章节。

12.3.2 总线 ECC

AS32A601 在 CPU CACHE、EFlash、SRAM、总线上各个主机接口、总线连接处提供了 ECC 保护功能。在对应监控区域出现单位/多位 ECC 错误时，ECC 模块将把错误上报给 FCU 模块。在单位错误时，ECC 模块可以自动进行纠正，多位错误时可以进行检测。

ECC 单元具体信号及其部署位置如下：

表 12.1 ECC 单元部署表

单元名称	单元说明	部署位置
cacheEcc	CPU D-Cache 与 I-Cache 的 SRAM 的 ECC	CPU 内部
cpuPortEcc	CPU I/D/P 接口的 ECC	CPU I/D/P 接口处
flashPortEcc	EFlash 总线读取接口 ECC	EFlash Axi Bus 0 总线读取接口处
eflashEcc	EFlash 存储的 ECC	EFlash 内部
dma0Ecc	DMA0 主机接口 ECC	DMA0 Axi Bus 1 主机接口处
dma1Ecc	DMA1 主机接口 ECC	DMA1 Axi Bus 1 主机接口处
sramEcc	SRAM0、SRAM1、SRAM2、SRAM3 的读写 ECC	SRAM 于 Axi Bus 1 的总线接口及其内部
apb0Ecc	APB0 转接口 ECC	Axi Bus 3 到 APB0 的转接口处
apb1Ecc	APB1 转接口 ECC	Axi Bus 3 到 APB1 的转接口处
axilite0Ecc	Axilite0 主机接口 ECC	Axi Bus 2 到 Axilite0 的转接口处

单元名称	单元说明	部署位置
axilite1Ecc	Axilite1 主机接口 ECC	Axi Bus 3 到 Axilite1 的转接口处
axilite2Ecc	Axilite2 主机接口 ECC	Axi Bus 3 到 Axilite2 的转接口处

错误控制细节请参考 FCU 部分的说明

12.3.3 FCU

错误控制单元（FCU）用于从硬件功能安全模块接受错误相关的信号，从而产生中断或复位请求。

更多细节请参考错误控制单元章节。

12.3.4 时钟监控

时钟监控单元（CMU）用于对内部 16M 振荡器时钟（FIRC）、PLL 输出时钟（PLL_Q、PLL_R）进行监控。

其中，FIRC、PLL_Q、PLL_R 时钟使用 OSC 时钟进行监控，OSC 使用 SIRC 时钟进行监控。

更多细节请参考时钟监控单元章节。

12.3.5 CRC

CRC 通过计算数据的 CRC 校验值并将其与之前计算好的 CRC 进行比较，可用于检测传输中数据或存储的数据是否被意外篡改。

更多细节请参考循环冗余校验章节。

12.3.6 电源监控

AS32A601 提供了电源监控（POR、LVR、LVD、HVD）对电源的检测，其中 POR 常开，LVR、LVD、HVD 用户选配。POR、LVR 在电源电压 VDD 过低时会触发整个芯片的复位，LVD、HVD 在电源电压 VDD 超出配置的范围时会上报 FCU 处理。这样在 VDD 异常时，电源监控可以让芯片处于安全状态。

12.3.7 总线超时监控

AS32A601 在每个总线主机接口上提供了超时检测单元，以确保设备不会因总线/个别外设问题整体卡死。具体配置

12.3.8 看门狗

看门狗提供了程序运行时间监控以及外部电路监控功能。AS32A601 提供两类看门狗：

- 独立看门狗（IWDG）：采用内部 32K 时钟，确保整个系统按照计划执行；
- 窗口看门狗（WWDG）：单个四独立通道设计，带有窗口功能，可用于软件监控。 更多细节请参考独立看门狗和窗口看门狗章节。

12.3.9 系统资源的多样性

功能安全设计通常会在系统中进行冗余设计，AS32A601 提供多种系统资源来支持用户的冗余设计。

- 数字输入可以被复制以获得冗余的安全输入；
- 对至关重要的数字输出可以冗余的写入，或者与回读操作相结合；
- ADC 可以用过采样的方法来检测瞬时误差的影响。在测量功能安全相关的信号时，可以通过两个 ADC 同时测量同一个模拟信号来得到冗余的转换结果；
- 芯片提供了 SPI、IIC、USART、LIN、CAN 等多样的通信模块来支持系统设计出于安全原因的通信通道冗余。

13 错误收集模块（FCU）

13.1 简介

错误控制单元（FCU）作为系统功能安全的重要部分，其负责收集各个模块的错误信息，并对这些错误信息进行处理。 对于不同的错误，FCU 可以采取不同的行为进行处理，中断行为将交由 PLIC 进行中断操作，复位行为将交由 SMU 进行复位操作。

13.2 特性

- 多种不同的单点故障与多点故障的监测与处理
- 收到硬件错误时，根据配置产生中断或复位请求
- 软件错误可计数触发
- 可屏蔽的错误处理

13.3 详细设计

13.3.1 整体框图

共有 42 个通道，在此画出一个以供示例。

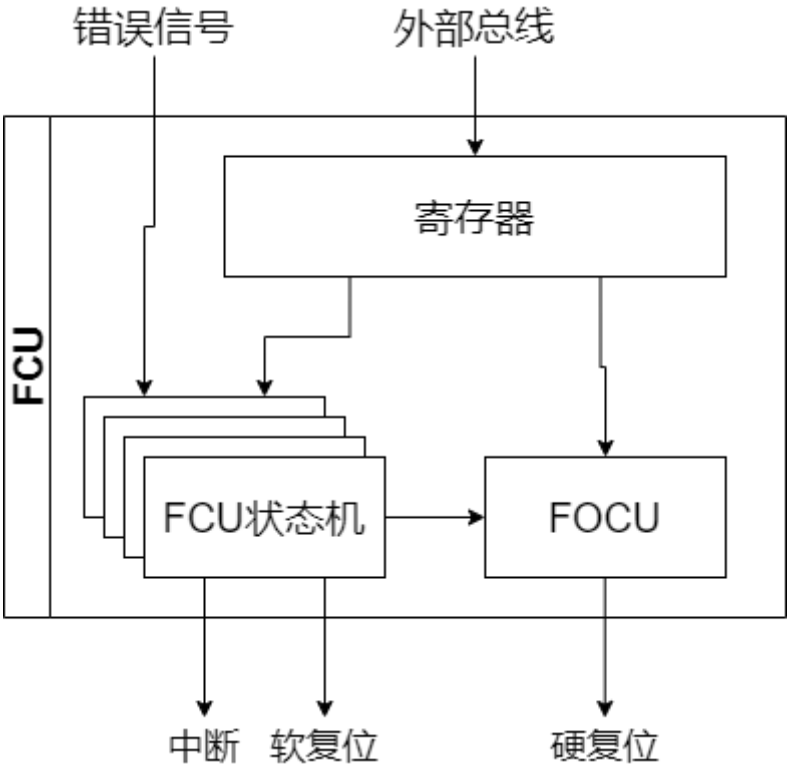


图 13.1 FCU 设计示意图

13.3.2 FCU 链接说明

FCU 与外部错误信号的连接关系如下：

表 13.1 FCU 信号源连接表

通道	信号源
0	cpuDcls
1	mpu_0
2	mpu_1
3	mpu_2
4	cacheEcc
5	cpuPortEcc
6	flashPortEcc
7	eflashEcc
8	cpuIBusTimeout
9	dma0Dcls

通道	信号源
10	dma1Dcls
11	mpuDma_0
12	mpuDma_1
13	dma0Ecc
14	dma1Ecc
15	sramEcc
16	cpuDBusTimeout
17	macBusTimeout
18	dma0BusTimeout
19	dma1BusTimeout
20	dseBusTimeout
21	cpuPBusTimeout
22	apb0Ecc
23	apb1Ecc
24	axilite0Ecc
25	cmu_0
26	cmu_1
27	cmu_2
28	cmu_3
29	axilite1Ecc
30	axilite2Ecc
31	wwdg0
32	wwdg1
33	lvd
34	hvd
35	iwdg
36	软件源 0
37	软件源 1
38	软件源 2

通道	信号源
39	软件源 3
40	外部错误 0
41	外部错误 1

13.3.3 应用说明

13.3.3.1 硬件触发通道配置

硬件触发通道应按照如下顺序进行配置：

1. 配置 FCU_CFGx 寄存器的 LEVELx 位。若错误为 ECC 错误，在配置为 1 时，单比特错误触发 Alarm，多比特错误触发 Fault，在配置为 2 时，单比特与多比特错误都会触发 Fault。若错误为其他错误，当配置为 1 时，错误信号会触发 Alarm，配置为 2 时，错误信号会触发 Fault；
2. 配置 FCU_CFGx 寄存器的 AACTx 位与 FACTx 位决定 FCU 对于对应通道错误的处理行为；
3. 配置 FCU_CFGx 寄存器的 ATF_CNTx 与 FOCU_CNTx 位以配置 Alarm 转 Fault 计数器与 FOCU 计数器；
4. 配置 FCU_CFGx 寄存器的 MASKx = 0，ENx = 1 以使能通道。

13.3.3.2 软件触发通道配置

软件触发通道应按照如下顺序进行配置：

1. 配置 SFCNT 寄存器的 CNTx 位以配置对应软件通道的触发阈值；
2. 配置 CFGx 寄存器的 LEVELx 位。若错误为 ECC 错误，在配置为 1 时，单比特错误触发 Alarm，多比特错误触发 Fault，在配置为 2 时，单比特与多比特错误都会触发 Fault。若错误为其他错误，当配置为 1 时，错误信号会触发 Alarm，配置为 2 时，错误信号会触发 Fault；
3. 配置 CFGx 寄存器的 AACTx 位与 FACTx 位决定 FCU 对于对应通道错误的处理行为；
4. 配置 CFGx 寄存器的 ATF_CNTx 与 FOCU_CNTx 位以配置 Alarm 转 Fault 计数器与 FOCU 计数器；

5. 配置 CFGx 寄存器的 MASKx = 0，ENx = 1 以使能通道。

13.3.3.3 硬件触发通道中断处理

在硬件触发通道中断处理时，步骤如下：

1. 查询 ALARMy 与 FAULTy 的 Ax 位与 Fx 位，找到触发的错误位置；
2. 配置 CFGx 寄存器的 MASKx = 1 以保证通道不会继续跳变与计数；
3. 清除对应硬件模块的 Fault 与 Alarm，如果为 ECC 或总线错误，通过写 FCU 的 OCLR 寄存器的对应位清除；
4. CLEARy 寄存器的 Cx 位写入 1 以清除 FCU 通道的错误状态；
5. 配置 CFGx 寄存器的 MASKx = 0 以保证通道继续运行；

13.3.3.4 软件触发通道中断处理

在软件触发通道中断处理时，步骤如下：

1. 查询 ALARMy 与 FAULTy 的 Ax 位与 Fx 位，找到触发的错误位置；
2. 配置 CFGx 寄存器的 MASKx = 1 以保证通道不会继续跳变与计数；
3. 写入 SFCLR 的 CLRx 位以清除软件通道的错误；
4. CLEARy 寄存器的 Cx 位写入 1 以清除 FCU 通道的错误状态；
5. 配置 CFGx 寄存器的 MASKx = 0 以保证通道继续运行；

13.3.3.5 硬件/软件触发通道复位处理

在硬件触发通道复位处理后，系统会进行软全局复位。在有配置复位处理的通道的软件处理中，应在配置系统信息前做如下步骤：

1. 查询 FAULTy 的 Fx 位，确定复位是否由错误产生，若无错误则进行初始化配置，若有错误执行步骤 2；
2. 根据用户程序定义对问题对外报告；
3. CLEARy 寄存器的 Cx 位写入 1 以清除 FCU 通道的错误状态；

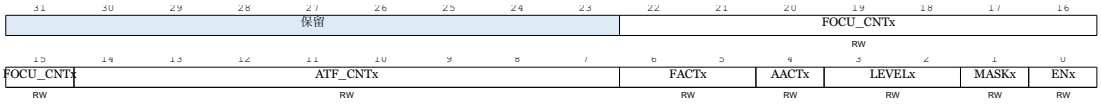
13.4 寄存器

FCU 基地址：0x30002000

13.4.1 FCU 通道 x 配置寄存器（FCU_CFGx）

地址偏移：0x0 + x*0x4

复位值：0x00000000



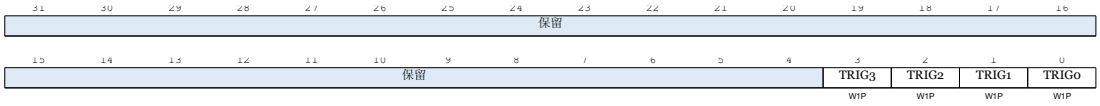
位/位域	名称	描述
31:23	保留	必须保持复位值
22:15	FOCU_CNTx	通道 xFOCU 计数值配置,超过计数值未处理错误信息则强制复位整个系统
14:7	ATF_CNTx	通道 xALARM 转 FAULT 计数值配置 0:关闭计数器 其他值:超过计数值周期未处理 ALARM 则通道状态变为 FAULT
6:5	FACTx	通道 xFAULT 状态行为 0:触发中断 2:全局软复位 1、3:保留
4	AACTx	通道 xALARM 状态行为 0:无行为 1:触发中断
3:2	LEVELx	通道 x 通道错误等级配置 0:CONFIG，设置模式时任何系统性报错皆被忽略 1:ALARM，报错后状态机将被置为 ALARM，超时后转为 FAULT

位/位域	名称	描述
		2:FAULT，报错后状态机将被置位为 FAULT 3:保留
1	MASKx	通道 x 通道 MASK 配置 0:关闭 MASK 模式,错误信息正常输出 1:开启 MASK 模式,错误信息屏蔽
0	ENx	通道 x 使能 0:关闭通道 1:通道使能

13.4.2 FCU 软件错误触发寄存器（FCU_SFTR）

地址偏移：0xA8

复位值：0x00000000



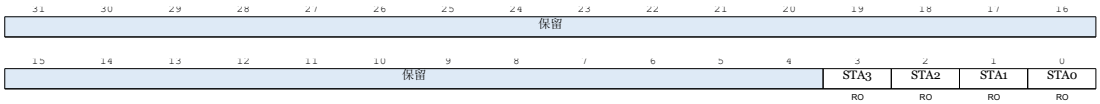
位/位域	名称	描述
31:4	保留	必须保持复位值
3	TRIG3	软件通道 3 错误触发位 0:无效果 1:计数一次软件错误
2	TRIG2	软件通道 2 错误触发位 0:无效果 1:计数一次软件错误

位/位域	名称	描述
1	TRIG1	软件通道 1 错误触发位 0:无效果 1:计数一次软件错误
0	TRIG0	软件通道 0 错误触发位 0:无效果 1:计数一次软件错误

13.4.3 FCU 软件错误状态寄存器（FCU_SFSTA）

地址偏移：0xAC

复位值：0x00000000



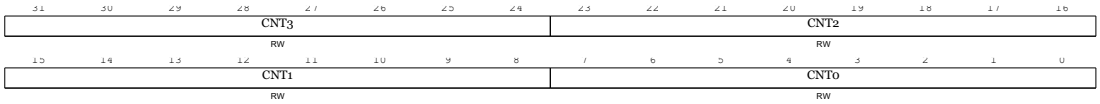
位/位域	名称	描述
31:4	保留	必须保持复位值
3	STA3	软件通道 3 错误位 0:未触发错误 1:错误触发
2	STA2	软件通道 2 错误位 0:未触发错误 1:错误触发
1	STA1	软件通道 1 错误位 0:未触发错误 1:错误触发

位/位域	名称	描述
0	STA0	软件通道 0 错误位 0:未触发错误 1:错误触发

13.4.4 FCU 软件错误计数寄存器（FCU_SFCNT）

地址偏移：0xB0

复位值：0x00000000

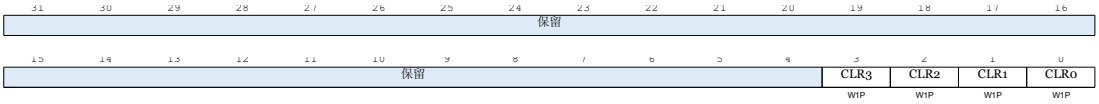


位/位域	名称	描述
31:24	CNT3	软件通道 3 错误计数 0:不计数，触发一次直接报错 其他值:触发到对应次数报错
23:16	CNT2	软件通道 2 错误计数 0:不计数，触发一次直接报错 其他值:触发到对应次数报错
15:8	CNT1	软件通道 1 错误计数 0:不计数，触发一次直接报错 其他值:触发到对应次数报错
7:0	CNT0	软件通道 0 错误计数 0:不计数，触发一次直接报错 其他值:触发到对应次数报错

13.4.5 FCU 软件错误清除寄存器（FCU_SFCLR）

地址偏移：0xB4

复位值：0x00000000

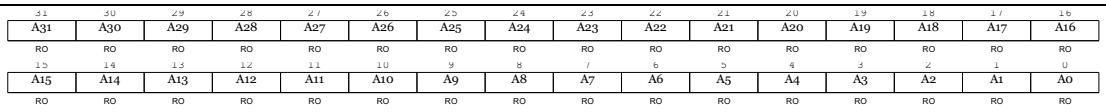


位/位域	名称	描述
31:4	保留	必须保持复位值
3	CLR3	软件通道 3 错误清除位 0:无效果 1:清除软件错误
2	CLR2	软件通道 2 错误清除位 0:无效果 1:清除软件错误
1	CLR1	软件通道 1 错误清除位 0:无效果 1:清除软件错误
0	CLR0	软件通道 0 错误清除位 0:无效果 1:清除软件错误

13.4.6 FCU 警报寄存器 0（FCU_ALARM0）

地址偏移：0xB8

复位值：0x00000000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

位/位域	名称	描述
31	A31	FCU 通道 31ALARM 位 0:通道 31 未触发 ALARM 1:通道 31 已触发 ALARM
30	A30	FCU 通道 30ALARM 位 0:通道 30 未触发 ALARM 1:通道 30 已触发 ALARM
29	A29	FCU 通道 29ALARM 位 0:通道 29 未触发 ALARM 1:通道 29 已触发 ALARM
28	A28	FCU 通道 28ALARM 位 0:通道 28 未触发 ALARM 1:通道 28 已触发 ALARM
27	A27	FCU 通道 27ALARM 位 0:通道 27 未触发 ALARM 1:通道 27 已触发 ALARM
26	A26	FCU 通道 26ALARM 位 0:通道 26 未触发 ALARM 1:通道 26 已触发 ALARM
25	A25	FCU 通道 25ALARM 位

位/位域	名称	描述
		0:通道 25 未触发 ALARM 1:通道 25 已触发 ALARM
24	A24	FCU 通道 24ALARM 位 0:通道 24 未触发 ALARM 1:通道 24 已触发 ALARM
23	A23	FCU 通道 23ALARM 位 0:通道 23 未触发 ALARM 1:通道 23 已触发 ALARM
22	A22	FCU 通道 22ALARM 位 0:通道 22 未触发 ALARM 1:通道 22 已触发 ALARM
21	A21	FCU 通道 21ALARM 位 0:通道 21 未触发 ALARM 1:通道 21 已触发 ALARM
20	A20	FCU 通道 20ALARM 位 0:通道 20 未触发 ALARM 1:通道 20 已触发 ALARM
19	A19	FCU 通道 19ALARM 位 0:通道 19 未触发 ALARM 1:通道 19 已触发 ALARM

位/位域	名称	描述
18	A18	FCU 通道 18ALARM 位 0:通道 18 未触发 ALARM 1:通道 18 已触发 ALARM
17	A17	FCU 通道 17ALARM 位 0:通道 17 未触发 ALARM 1:通道 17 已触发 ALARM
16	A16	FCU 通道 16ALARM 位 0:通道 16 未触发 ALARM 1:通道 16 已触发 ALARM
15	A15	FCU 通道 15ALARM 位 0:通道 15 未触发 ALARM 1:通道 15 已触发 ALARM
14	A14	FCU 通道 14ALARM 位 0:通道 14 未触发 ALARM 1:通道 14 已触发 ALARM
13	A13	FCU 通道 13ALARM 位 0:通道 13 未触发 ALARM 1:通道 13 已触发 ALARM
12	A12	FCU 通道 12ALARM 位 0:通道 12 未触发 ALARM 1:通道 12 已触发 ALARM

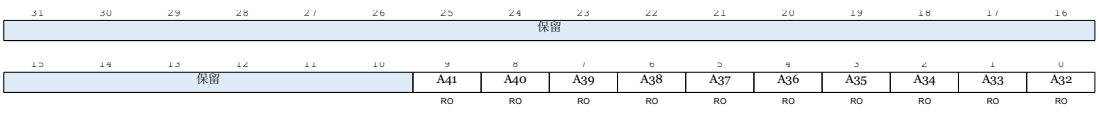
位/位域	名称	描述
11	A11	FCU 通道 11ALARM 位 0:通道 11 未触发 ALARM 1:通道 11 已触发 ALARM
10	A10	FCU 通道 10ALARM 位 0:通道 10 未触发 ALARM 1:通道 10 已触发 ALARM
9	A9	FCU 通道 9ALARM 位 0:通道 9 未触发 ALARM 1:通道 9 已触发 ALARM
8	A8	FCU 通道 8ALARM 位 0:通道 8 未触发 ALARM 1:通道 8 已触发 ALARM
7	A7	FCU 通道 7ALARM 位 0:通道 7 未触发 ALARM 1:通道 7 已触发 ALARM
6	A6	FCU 通道 6ALARM 位 0:通道 6 未触发 ALARM 1:通道 6 已触发 ALARM
5	A5	FCU 通道 5ALARM 位 0:通道 5 未触发 ALARM

位/位域	名称	描述
		1:通道 5 已触发 ALARM
4	A4	FCU 通道 4ALARM 位 0:通道 4 未触发 ALARM 1:通道 4 已触发 ALARM
3	A3	FCU 通道 3ALARM 位 0:通道 3 未触发 ALARM 1:通道 3 已触发 ALARM
2	A2	FCU 通道 2ALARM 位 0:通道 2 未触发 ALARM 1:通道 2 已触发 ALARM
1	A1	FCU 通道 1ALARM 位 0:通道 1 未触发 ALARM 1:通道 1 已触发 ALARM
0	A0	FCU 通道 0ALARM 位 0:通道 0 未触发 ALARM 1:通道 0 已触发 ALARM

13.4.7 FCU 警报寄存器 1（FCU_ALARM1）

地址偏移：0xBC

复位值：0x00000000



位/位域	名称	描述
31:10	保留	必须保持复位值
9	A41	FCU 通道 41ALARM 位 0:通道 41 未触发 ALARM 1:通道 41 已触发 ALARM
8	A40	FCU 通道 40ALARM 位 0:通道 40 未触发 ALARM 1:通道 40 已触发 ALARM
7	A39	FCU 通道 39ALARM 位 0:通道 39 未触发 ALARM 1:通道 39 已触发 ALARM
6	A38	FCU 通道 38ALARM 位 0:通道 38 未触发 ALARM 1:通道 38 已触发 ALARM
5	A37	FCU 通道 37ALARM 位 0:通道 37 未触发 ALARM 1:通道 37 已触发 ALARM
4	A36	FCU 通道 36ALARM 位 0:通道 36 未触发 ALARM 1:通道 36 已触发 ALARM
3	A35	FCU 通道 35ALARM 位 0:通道 35 未触发 ALARM

位/位域	名称	描述
		1:通道 35 已触发 ALARM
2	A34	FCU 通道 34ALARM 位 0:通道 34 未触发 ALARM 1:通道 34 已触发 ALARM
1	A33	FCU 通道 33ALARM 位 0:通道 33 未触发 ALARM 1:通道 33 已触发 ALARM
0	A32	FCU 通道 32ALARM 位 0:通道 32 未触发 ALARM 1:通道 32 已触发 ALARM

13.4.8 FCU 错误寄存器 0（FCU_FAULT0）

地址偏移：0xC0

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F31	F30	F29	F28	F27	F26	F25	F24	F23	F22	F21	F20	F19	F18	F17	F16
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

位/位域	名称	描述
31	F31	FCU 通道 31FAULT 位 0:通道 31 未触发 FAULT 1:通道 31 已触发 FAULT
30	F30	FCU 通道 30FAULT 位 0:通道 30 未触发 FAULT

位/位域	名称	描述
		1:通道 30 已触发 FAULT
29	F29	FCU 通道 29FAULT 位 0:通道 29 未触发 FAULT 1:通道 29 已触发 FAULT
28	F28	FCU 通道 28FAULT 位 0:通道 28 未触发 FAULT 1:通道 28 已触发 FAULT
27	F27	FCU 通道 27FAULT 位 0:通道 27 未触发 FAULT 1:通道 27 已触发 FAULT
26	F26	FCU 通道 26FAULT 位 0:通道 26 未触发 FAULT 1:通道 26 已触发 FAULT
25	F25	FCU 通道 25FAULT 位 0:通道 25 未触发 FAULT 1:通道 25 已触发 FAULT
24	F24	FCU 通道 24FAULT 位 0:通道 24 未触发 FAULT 1:通道 24 已触发 FAULT
23	F23	FCU 通道 23FAULT 位

位/位域	名称	描述
		0:通道 23 未触发 FAULT 1:通道 23 已触发 FAULT
22	F22	FCU 通道 22FAULT 位 0:通道 22 未触发 FAULT 1:通道 22 已触发 FAULT
21	F21	FCU 通道 21FAULT 位 0:通道 21 未触发 FAULT 1:通道 21 已触发 FAULT
20	F20	FCU 通道 20FAULT 位 0:通道 20 未触发 FAULT 1:通道 20 已触发 FAULT
19	F19	FCU 通道 19FAULT 位 0:通道 19 未触发 FAULT 1:通道 19 已触发 FAULT
18	F18	FCU 通道 18FAULT 位 0:通道 18 未触发 FAULT 1:通道 18 已触发 FAULT
17	F17	FCU 通道 17FAULT 位 0:通道 17 未触发 FAULT 1:通道 17 已触发 FAULT

位/位域	名称	描述
16	F16	FCU 通道 16FAULT 位 0:通道 16 未触发 FAULT 1:通道 16 已触发 FAULT
15	F15	FCU 通道 15FAULT 位 0:通道 15 未触发 FAULT 1:通道 15 已触发 FAULT
14	F14	FCU 通道 14FAULT 位 0:通道 14 未触发 FAULT 1:通道 14 已触发 FAULT
13	F13	FCU 通道 13FAULT 位 0:通道 13 未触发 FAULT 1:通道 13 已触发 FAULT
12	F12	FCU 通道 12FAULT 位 0:通道 12 未触发 FAULT 1:通道 12 已触发 FAULT
11	F11	FCU 通道 11FAULT 位 0:通道 11 未触发 FAULT 1:通道 11 已触发 FAULT
10	F10	FCU 通道 10FAULT 位 0:通道 10 未触发 FAULT 1:通道 10 已触发 FAULT

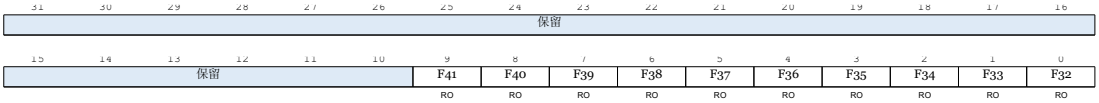
位/位域	名称	描述
9	F9	FCU 通道 9FAULT 位 0:通道 9 未触发 FAULT 1:通道 9 已触发 FAULT
8	F8	FCU 通道 8FAULT 位 0:通道 8 未触发 FAULT 1:通道 8 已触发 FAULT
7	F7	FCU 通道 7FAULT 位 0:通道 7 未触发 FAULT 1:通道 7 已触发 FAULT
6	F6	FCU 通道 6FAULT 位 0:通道 6 未触发 FAULT 1:通道 6 已触发 FAULT
5	F5	FCU 通道 5FAULT 位 0:通道 5 未触发 FAULT 1:通道 5 已触发 FAULT
4	F4	FCU 通道 4FAULT 位 0:通道 4 未触发 FAULT 1:通道 4 已触发 FAULT
3	F3	FCU 通道 3FAULT 位 0:通道 3 未触发 FAULT

位/位域	名称	描述
		1:通道 3 已触发 FAULT
2	F2	FCU 通道 2FAULT 位 0:通道 2 未触发 FAULT 1:通道 2 已触发 FAULT
1	F1	FCU 通道 1FAULT 位 0:通道 1 未触发 FAULT 1:通道 1 已触发 FAULT
0	F0	FCU 通道 0FAULT 位 0:通道 0 未触发 FAULT 1:通道 0 已触发 FAULT

13.4.9 FCU 错误寄存器 1（FCU_FAULT1）

地址偏移：0xC4

复位值：0x00000000



位/位域	名称	描述
31:10	保留	必须保持复位值
9	F41	FCU 通道 41FAULT 位 0:通道 41 未触发 FAULT 1:通道 41 已触发 FAULT
8	F40	FCU 通道 40FAULT 位

位/位域	名称	描述
		0:通道 40 未触发 FAULT 1:通道 40 已触发 FAULT
7	F39	FCU 通道 39FAULT 位 0:通道 39 未触发 FAULT 1:通道 39 已触发 FAULT
6	F38	FCU 通道 38FAULT 位 0:通道 38 未触发 FAULT 1:通道 38 已触发 FAULT
5	F37	FCU 通道 37FAULT 位 0:通道 37 未触发 FAULT 1:通道 37 已触发 FAULT
4	F36	FCU 通道 36FAULT 位 0:通道 36 未触发 FAULT 1:通道 36 已触发 FAULT
3	F35	FCU 通道 35FAULT 位 0:通道 35 未触发 FAULT 1:通道 35 已触发 FAULT
2	F34	FCU 通道 34FAULT 位 0:通道 34 未触发 FAULT 1:通道 34 已触发 FAULT

位/位域	名称	描述
1	F33	FCU 通道 33FAULT 位 0:通道 33 未触发 FAULT 1:通道 33 已触发 FAULT
0	F32	FCU 通道 32FAULT 位 0:通道 32 未触发 FAULT 1:通道 32 已触发 FAULT

13.4.10 FCU 警报/错误清除寄存器 0（FCU_CLEAR0）

地址偏移：0xC8

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C31	C30	C29	C28	C27	C26	C25	C24	C23	C22	C21	C20	C19	C18	C17	C16
W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P	W1P

位/位域	名称	描述
31	C31	FCU 通道 31ALARM/FAULT 清除位 0:无效果 1:通道 31ALARM 位与 FAULT 位清除
30	C30	FCU 通道 30ALARM/FAULT 清除位 0:无效果 1:通道 30ALARM 位与 FAULT 位清除
29	C29	FCU 通道 29ALARM/FAULT 清除位 0:无效果 1:通道 29ALARM 位与 FAULT 位清除

位/位域	名称	描述
28	C28	FCU 通道 28ALARM/FAULT 清除位 0:无效果 1:通道 28ALARM 位与 FAULT 位清除
27	C27	FCU 通道 27ALARM/FAULT 清除位 0:无效果 1:通道 27ALARM 位与 FAULT 位清除
26	C26	FCU 通道 26ALARM/FAULT 清除位 0:无效果 1:通道 26ALARM 位与 FAULT 位清除
25	C25	FCU 通道 25ALARM/FAULT 清除位 0:无效果 1:通道 25ALARM 位与 FAULT 位清除
24	C24	FCU 通道 24ALARM/FAULT 清除位 0:无效果 1:通道 24ALARM 位与 FAULT 位清除
23	C23	FCU 通道 23ALARM/FAULT 清除位 0:无效果 1:通道 23ALARM 位与 FAULT 位清除
22	C22	FCU 通道 22ALARM/FAULT 清除位 0:无效果 1:通道 22ALARM 位与 FAULT 位清除

位/位域	名称	描述
21	C21	FCU 通道 21ALARM/FAULT 清除位 0:无效果 1:通道 21ALARM 位与 FAULT 位清除
20	C20	FCU 通道 20ALARM/FAULT 清除位 0:无效果 1:通道 20ALARM 位与 FAULT 位清除
19	C19	FCU 通道 19ALARM/FAULT 清除位 0:无效果 1:通道 19ALARM 位与 FAULT 位清除
18	C18	FCU 通道 18ALARM/FAULT 清除位 0:无效果 1:通道 18ALARM 位与 FAULT 位清除
17	C17	FCU 通道 17ALARM/FAULT 清除位 0:无效果 1:通道 17ALARM 位与 FAULT 位清除
16	C16	FCU 通道 16ALARM/FAULT 清除位 0:无效果 1:通道 16ALARM 位与 FAULT 位清除
15	C15	FCU 通道 15ALARM/FAULT 清除位 0:无效果

位/位域	名称	描述
		1:通道 15ALARM 位与 FAULT 位清除
14	C14	FCU 通道 14ALARM/FAULT 清除位 0:无效果 1:通道 14ALARM 位与 FAULT 位清除
13	C13	FCU 通道 13ALARM/FAULT 清除位 0:无效果 1:通道 13ALARM 位与 FAULT 位清除
12	C12	FCU 通道 12ALARM/FAULT 清除位 0:无效果 1:通道 12ALARM 位与 FAULT 位清除
11	C11	FCU 通道 11ALARM/FAULT 清除位 0:无效果 1:通道 11ALARM 位与 FAULT 位清除
10	C10	FCU 通道 10ALARM/FAULT 清除位 0:无效果 1:通道 10ALARM 位与 FAULT 位清除
9	C9	FCU 通道 9ALARM/FAULT 清除位 0:无效果 1:通道 9ALARM 位与 FAULT 位清除
8	C8	FCU 通道 8ALARM/FAULT 清除位

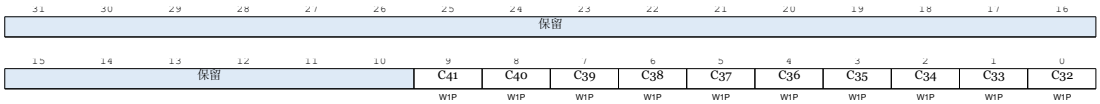
位/位域	名称	描述
		0:无效果 1:通道 8ALARM 位与 FAULT 位清除
7	C7	FCU 通道 7ALARM/FAULT 清除位 0:无效果 1:通道 7ALARM 位与 FAULT 位清除
6	C6	FCU 通道 6ALARM/FAULT 清除位 0:无效果 1:通道 6ALARM 位与 FAULT 位清除
5	C5	FCU 通道 5ALARM/FAULT 清除位 0:无效果 1:通道 5ALARM 位与 FAULT 位清除
4	C4	FCU 通道 4ALARM/FAULT 清除位 0:无效果 1:通道 4ALARM 位与 FAULT 位清除
3	C3	FCU 通道 3ALARM/FAULT 清除位 0:无效果 1:通道 3ALARM 位与 FAULT 位清除
2	C2	FCU 通道 2ALARM/FAULT 清除位 0:无效果 1:通道 2ALARM 位与 FAULT 位清除

位/位域	名称	描述
1	C1	FCU 通道 1ALARM/FAULT 清除位 0:无效果 1:通道 1ALARM 位与 FAULT 位清除
0	C0	FCU 通道 0ALARM/FAULT 清除位 0:无效果 1:通道 0ALARM 位与 FAULT 位清除

13.4.11 FCU 警报/错误清除寄存器 1（FCU_CLEAR1）

地址偏移：0xCC

复位值：0x00000000



位/位域	名称	描述
31:10	保留	必须保持复位值
9	C41	FCU 通道 41ALARM/FAULT 清除位 0:无效果 1:通道 41ALARM 位与 FAULT 位清除
8	C40	FCU 通道 40ALARM/FAULT 清除位 0:无效果 1:通道 40ALARM 位与 FAULT 位清除
7	C39	FCU 通道 39ALARM/FAULT 清除位 0:无效果 1:通道 39ALARM 位与 FAULT 位清除

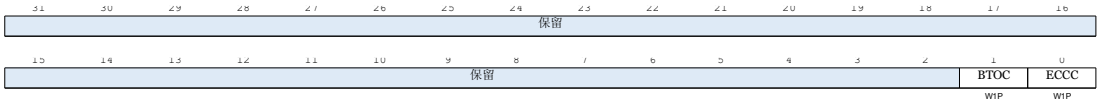
位/位域	名称	描述
6	C38	FCU 通道 38ALARM/FAULT 清除位 0:无效果 1:通道 38ALARM 位与 FAULT 位清除
5	C37	FCU 通道 37ALARM/FAULT 清除位 0:无效果 1:通道 37ALARM 位与 FAULT 位清除
4	C36	FCU 通道 36ALARM/FAULT 清除位 0:无效果 1:通道 36ALARM 位与 FAULT 位清除
3	C35	FCU 通道 35ALARM/FAULT 清除位 0:无效果 1:通道 35ALARM 位与 FAULT 位清除
2	C34	FCU 通道 34ALARM/FAULT 清除位 0:无效果 1:通道 34ALARM 位与 FAULT 位清除
1	C33	FCU 通道 33ALARM/FAULT 清除位 0:无效果 1:通道 33ALARM 位与 FAULT 位清除
0	C32	FCU 通道 32ALARM/FAULT 清除位 0:无效果

位/位域	名称	描述
		1:通道 32ALARM 位与 FAULT 位清除

13.4.12 FCU 杂项错误清除寄存器（FCU_OCLR）

地址偏移：0xD0

复位值：0x00000000



位/位域	名称	描述
31:2	保留	必须保持复位值
1	BTOC	总线超时计数器信号清除器 0:无效果 1:清除总线超时信号
0	ECCC	ECC 控制器信号清除器 0:无效果 1:清除 ECC 信号

14 内存保护模块（MPU）

14.1 简介

AS32A601 为用户提供了 MPU 以对特定地址的内存/外设进行保护。MPU 可用于禁止程序破坏关键任务（例如操作系统内核）的数据，锁定 SRAM 内存区域（读保护+写保护）以防止代码注入攻击等操作来保护整个嵌入式系统，进而提升系统的鲁棒性与安全性。

14.2 主要特征

- 任意保护宽度，特定阈值间的任意保护范围；
- 16 个（CPU I/D/P MPU）/8 个（DMA MPU）互相独立的内存保护通道，各个通道之间的保护区域可叠加；
- 单周期快速异常处理；
- 在命中时偏移访问地址以防止出现非法访问情况；
- 支持三种保护模式：读保护，写保护，或锁定区域（读保护+写保护）。

14.3 功能说明

14.3.1 MPU 操作

MPU 通过读取其控制接口的地址位，基于 MPU_SADDRx 与 MPU_EADDRx 的地址，进行比对，来确定是否触发保护机制。

MPU 触发公式如下，

$$\mathrm{SADDRx} \leq \mathrm{MADDR} \leq \mathrm{EADDRx}$$

其中 MADDR 为总线发送地址，在配置时，MPU_SADDRx 应大于 MPU_EADDRx，否则会导致通道失效。

14.3.2 通道配置

要对特定的内存区域进行保护，建议遵循以下步骤进行操作：

- 读取 MPU_PTEN 寄存器的 ENx 位，如果为 1（通道已使能），清零该位。当 ENx 为 0 时，请按照下列步骤配置 MPU 开始内存保护的配置。
- 通过 MPU_RPTEN 寄存器的 ENx 位与 MPU_WPTEN 寄存器的 ENx 位配置读、写保护模式（CPU I-Port MPU 不存在写通道，故无写保护相关配置）。
- 通过 MPU_SADDRx 寄存器配置保护下限地址。
- 通过 MPU_EADDRx 寄存器配置保护上限地址。
- 将 MPU_EN 寄存器的 ENx 位置 1，开启对应通道的内存保护。
- 触发错误。

Note

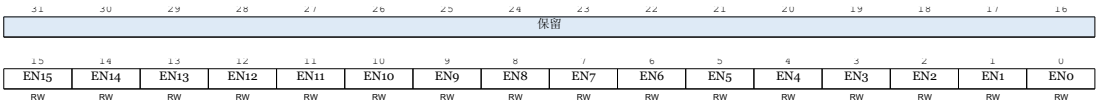
错误通道的具体配置请参考错误收集模块（FCU）部分

14.4 CPU I-Port MPU 寄存器

14.4.1 MPU 保护通道使能寄存器（MPU_PTEN）

地址偏移：0x0

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	EN15	通道 15 使能信号 0:MPU 通道关闭 1:MPU 通道开启
14	EN14	通道 14 使能信号 0:MPU 通道关闭 1:MPU 通道开启

位/位域	名称	描述
13	EN13	通道 13 使能信号 0:MPU 通道关闭 1:MPU 通道开启
12	EN12	通道 12 使能信号 0:MPU 通道关闭 1:MPU 通道开启
11	EN11	通道 11 使能信号 0:MPU 通道关闭 1:MPU 通道开启
10	EN10	通道 10 使能信号 0:MPU 通道关闭 1:MPU 通道开启
9	EN9	通道 9 使能信号 0:MPU 通道关闭 1:MPU 通道开启
8	EN8	通道 8 使能信号 0:MPU 通道关闭 1:MPU 通道开启
7	EN7	通道 7 使能信号 0:MPU 通道关闭

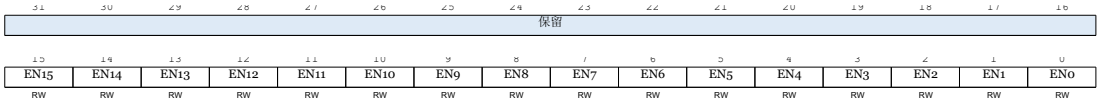
位/位域	名称	描述
		1:MPU 通道开启
6	EN6	通道 6 使能信号 0:MPU 通道关闭 1:MPU 通道开启
5	EN5	通道 5 使能信号 0:MPU 通道关闭 1:MPU 通道开启
4	EN4	通道 4 使能信号 0:MPU 通道关闭 1:MPU 通道开启
3	EN3	通道 3 使能信号 0:MPU 通道关闭 1:MPU 通道开启
2	EN2	通道 2 使能信号 0:MPU 通道关闭 1:MPU 通道开启
1	EN1	通道 1 使能信号 0:MPU 通道关闭 1:MPU 通道开启
0	EN0	通道 0 使能信号

位/位域	名称	描述
		0:MPU 通道关闭 1:MPU 通道开启

14.4.2 MPU 读保护使能寄存器（MPU_RPTEN）

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	EN15	通道 15 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
14	EN14	通道 14 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
13	EN13	通道 13 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
12	EN12	通道 12 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

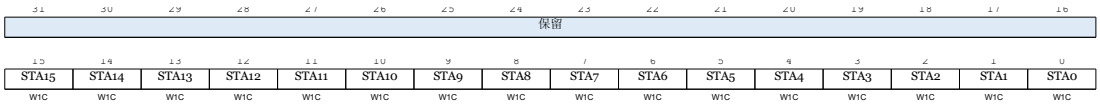
位/位域	名称	描述
11	EN11	通道 11 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
10	EN10	通道 10 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
9	EN9	通道 9 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
8	EN8	通道 8 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
7	EN7	通道 7 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
6	EN6	通道 6 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
5	EN5	通道 5 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

位/位域	名称	描述
4	EN4	通道 4 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
3	EN3	通道 3 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
2	EN2	通道 2 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
1	EN1	通道 1 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
0	EN0	通道 0 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

14.4.3 MPU 读保护状态标志寄存器（MPU_RPTS）

地址偏移：0x8

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	STA15	通道 15 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
14	STA14	通道 14 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
13	STA13	通道 13 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
12	STA12	通道 12 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
11	STA11	通道 11 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
10	STA10	通道 10 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
9	STA9	通道 9 读保护状态标志位,写入 1 清零 0:通道读保护未触发

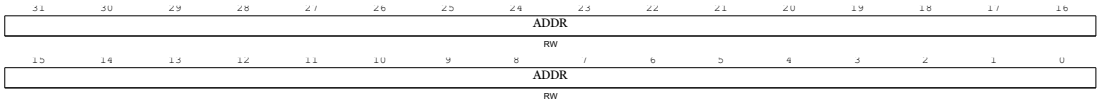
位/位域	名称	描述
		1:通道读保护触发
8	STA8	通道 8 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
7	STA7	通道 7 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
6	STA6	通道 6 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
5	STA5	通道 5 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
4	STA4	通道 4 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
3	STA3	通道 3 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
2	STA2	通道 2 读保护状态标志位,写入 1 清零

位/位域	名称	描述
		0:通道读保护未触发 1:通道读保护触发
1	STA1	通道 1 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
0	STA0	通道 0 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发

14.4.4 MPU 保护通道 x 起始地址寄存器（MPU_SADDRx）

地址偏移：0xC + x*0x8

复位值：0x00000000

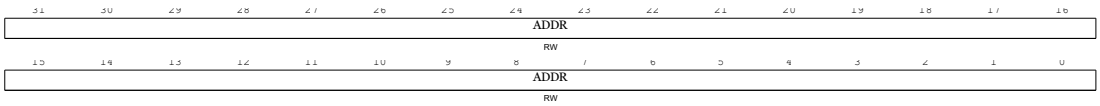


位/位域	名称	描述
31:0	ADDR	通道 x 保护开始地址 大于该地址（不包括该地址）的地址域将被 MPU 保护

14.4.5 MPU 保护通道 x 结束地址寄存器（MPU_EADDRx）

地址偏移：0x10 + x*0x8

复位值：0x00000000



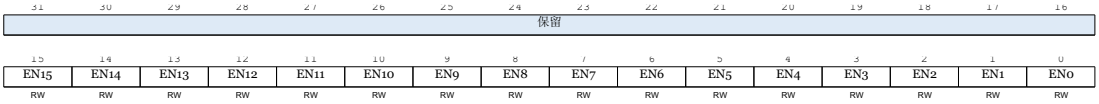
位/位域	名称	描述
31:0	ADDR	通道 x 保护结束地址 小于该地址（不包括该地址）的地址域将被 MPU 保护

14.5 CPU D-Port/P-Port MPU 寄存器

14.5.1 MPU 保护通道使能寄存器 MPU_PTEN

地址偏移：0x0

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	EN15	通道 15 使能信号 0:MPU 通道关闭 1:MPU 通道开启
14	EN14	通道 14 使能信号 0:MPU 通道关闭 1:MPU 通道开启
13	EN13	通道 13 使能信号 0:MPU 通道关闭 1:MPU 通道开启
12	EN12	通道 12 使能信号 0:MPU 通道关闭 1:MPU 通道开启

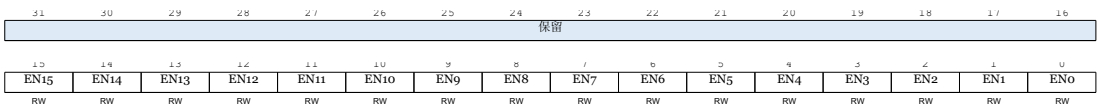
位/位域	名称	描述
11	EN11	通道 11 使能信号 0:MPU 通道关闭 1:MPU 通道开启
10	EN10	通道 10 使能信号 0:MPU 通道关闭 1:MPU 通道开启
9	EN9	通道 9 使能信号 0:MPU 通道关闭 1:MPU 通道开启
8	EN8	通道 8 使能信号 0:MPU 通道关闭 1:MPU 通道开启
7	EN7	通道 7 使能信号 0:MPU 通道关闭 1:MPU 通道开启
6	EN6	通道 6 使能信号 0:MPU 通道关闭 1:MPU 通道开启
5	EN5	通道 5 使能信号 0:MPU 通道关闭

位/位域	名称	描述
		1:MPU 通道开启
4	EN4	通道 4 使能信号 0:MPU 通道关闭 1:MPU 通道开启
3	EN3	通道 3 使能信号 0:MPU 通道关闭 1:MPU 通道开启
2	EN2	通道 2 使能信号 0:MPU 通道关闭 1:MPU 通道开启
1	EN1	通道 1 使能信号 0:MPU 通道关闭 1:MPU 通道开启
0	EN0	通道 0 使能信号 0:MPU 通道关闭 1:MPU 通道开启

14.5.2 MPU 写保护使能寄存器（MPU_WPTEN）

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	EN15	通道 15 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
14	EN14	通道 14 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
13	EN13	通道 13 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
12	EN12	通道 12 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
11	EN11	通道 11 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
10	EN10	通道 10 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
9	EN9	通道 9 写保护使能信号 0:通道写保护关闭

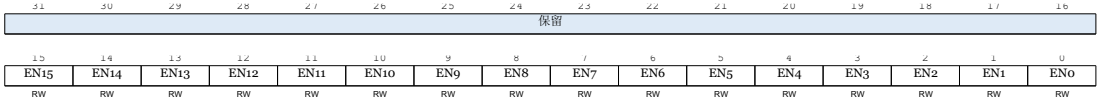
位/位域	名称	描述
		1:通道写保护打开
8	EN8	通道 8 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
7	EN7	通道 7 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
6	EN6	通道 6 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
5	EN5	通道 5 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
4	EN4	通道 4 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
3	EN3	通道 3 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
2	EN2	通道 2 写保护使能信号

位/位域	名称	描述
		0:通道写保护关闭 1:通道写保护打开
1	EN1	通道 1 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
0	EN0	通道 0 写保护使能信号 0:通道写保护关闭 1:通道写保护打开

14.5.3 MPU 读保护使能寄存器（MPU_RPTEN）

地址偏移：0x8

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	EN15	通道 15 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
14	EN14	通道 14 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

位/位域	名称	描述
13	EN13	通道 13 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
12	EN12	通道 12 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
11	EN11	通道 11 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
10	EN10	通道 10 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
9	EN9	通道 9 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
8	EN8	通道 8 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
7	EN7	通道 7 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

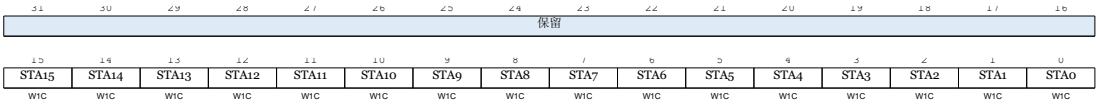
位/位域	名称	描述
6	EN6	通道 6 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
5	EN5	通道 5 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
4	EN4	通道 4 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
3	EN3	通道 3 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
2	EN2	通道 2 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
1	EN1	通道 1 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
0	EN0	通道 0 读保护使能通道 0:通道读保护关闭

位/位域	名称	描述
		1:通道读保护打开

14.5.4 MPU 写保护状态标志寄存器（MPU_WPTS）

地址偏移：0xC

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	STA15	通道 15 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
14	STA14	通道 14 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
13	STA13	通道 13 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
12	STA12	通道 12 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
11	STA11	通道 11 写保护状态标志位,写入 1 清零

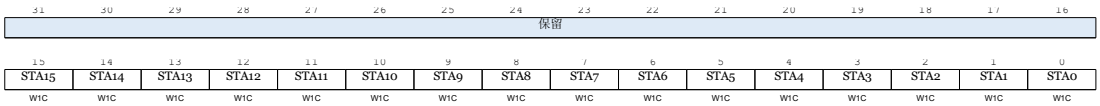
位/位域	名称	描述
		0:通道写保护未触发 1:通道写保护触发
10	STA10	通道 10 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
9	STA9	通道 9 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
8	STA8	通道 8 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
7	STA7	通道 7 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
6	STA6	通道 6 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
5	STA5	通道 5 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发

位/位域	名称	描述
4	STA4	通道 4 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
3	STA3	通道 3 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
2	STA2	通道 2 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
1	STA1	通道 1 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
0	STA0	通道 0 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发

14.5.5 MPU 读保护状态标志寄存器（MPU_RPTS）

地址偏移：0x10

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	STA15	通道 15 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
14	STA14	通道 14 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
13	STA13	通道 13 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
12	STA12	通道 12 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
11	STA11	通道 11 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
10	STA10	通道 10 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
9	STA9	通道 9 读保护状态标志位,写入 1 清零 0:通道读保护未触发

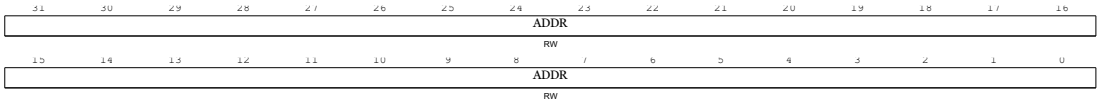
位/位域	名称	描述
		1:通道读保护触发
8	STA8	通道 8 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
7	STA7	通道 7 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
6	STA6	通道 6 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
5	STA5	通道 5 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
4	STA4	通道 4 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
3	STA3	通道 3 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
2	STA2	通道 2 读保护状态标志位,写入 1 清零

位/位域	名称	描述
		0:通道读保护未触发 1:通道读保护触发
1	STA1	通道 1 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
0	STA0	通道 0 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发

14.5.6 MPU 保护通道 x 起始地址寄存器（MPU_SADDRx）

地址偏移：0x14 + x*0x8

复位值：0x00000000

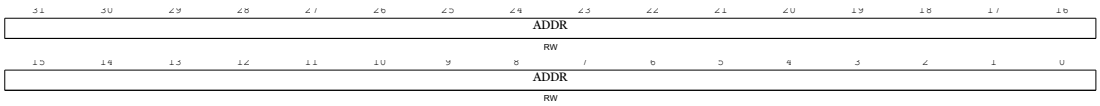


位/位域	名称	描述
31:0	ADDR	通道 x 保护开始地址 大于该地址（不包括该地址）的地址域将被 MPU 保护

14.5.7 MPU 保护通道 x 结束地址寄存器（MPU_EADDRx）

地址偏移：0x18 + x*0x8

复位值：0x00000000



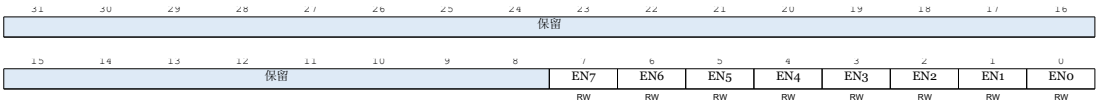
位/位域	名称	描述
31:0	ADDR	通道 x 保护结束地址 小于该地址（不包括该地址）的地址域将被 MPU 保护

14.6 DMA0/DMA1 MPU 寄存器

14.6.1 MPU 保护通道使能寄存器（MPU_PTEN）

地址偏移：0x0

复位值：0x00000000



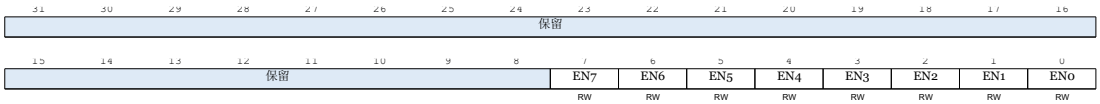
位/位域	名称	描述
31:8	保留	必须保持复位值
7	EN7	通道 7 使能信号 0:MPU 通道关闭 1:MPU 通道开启
6	EN6	通道 6 使能信号 0:MPU 通道关闭 1:MPU 通道开启
5	EN5	通道 5 使能信号 0:MPU 通道关闭 1:MPU 通道开启
4	EN4	通道 4 使能信号 0:MPU 通道关闭 1:MPU 通道开启

位/位域	名称	描述
3	EN3	通道 3 使能信号 0:MPU 通道关闭 1:MPU 通道开启
2	EN2	通道 2 使能信号 0:MPU 通道关闭 1:MPU 通道开启
1	EN1	通道 1 使能信号 0:MPU 通道关闭 1:MPU 通道开启
0	EN0	通道 0 使能信号 0:MPU 通道关闭 1:MPU 通道开启

14.6.2 MPU 写保护使能寄存器（MPU_WPTEN）

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:8	保留	必须保持复位值
7	EN7	通道 7 写保护使能信号 0:通道写保护关闭

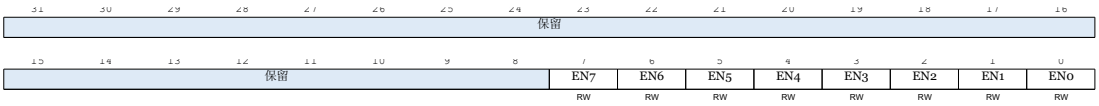
位/位域	名称	描述
		1:通道写保护打开
6	EN6	通道 6 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
5	EN5	通道 5 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
4	EN4	通道 4 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
3	EN3	通道 3 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
2	EN2	通道 2 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
1	EN1	通道 1 写保护使能信号 0:通道写保护关闭 1:通道写保护打开
0	EN0	通道 0 写保护使能信号

位/位域	名称	描述
		0:通道写保护关闭 1:通道写保护打开

14.6.3 MPU 读保护使能寄存器（MPU_RPTEN）

地址偏移：0x8

复位值：0x00000000



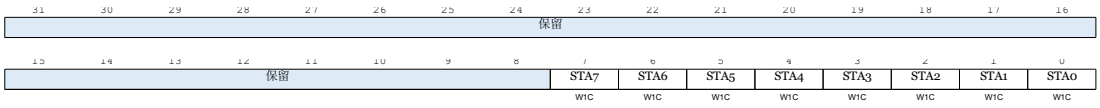
位/位域	名称	描述
31:8	保留	必须保持复位值
7	EN7	通道 7 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
6	EN6	通道 6 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
5	EN5	通道 5 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
4	EN4	通道 4 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

位/位域	名称	描述
3	EN3	通道 3 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
2	EN2	通道 2 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
1	EN1	通道 1 读保护使能通道 0:通道读保护关闭 1:通道读保护打开
0	EN0	通道 0 读保护使能通道 0:通道读保护关闭 1:通道读保护打开

14.6.4 MPU 写保护状态标志寄存器（MPU_WPTS）

地址偏移：0xC

复位值：0x00000000



位/位域	名称	描述
31:8	保留	必须保持复位值
7	STA7	通道 7 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发

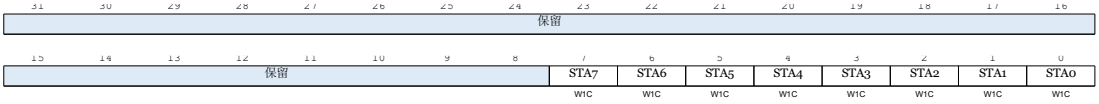
位/位域	名称	描述
6	STA6	通道 6 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
5	STA5	通道 5 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
4	STA4	通道 4 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
3	STA3	通道 3 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
2	STA2	通道 2 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
1	STA1	通道 1 写保护状态标志位,写入 1 清零 0:通道写保护未触发 1:通道写保护触发
0	STA0	通道 0 写保护状态标志位,写入 1 清零 0:通道写保护未触发

位/位域	名称	描述
		1:通道写保护触发

14.6.5 MPU 读保护状态标志寄存器（MPU_RPTS）

地址偏移：0x10

复位值：0x00000000



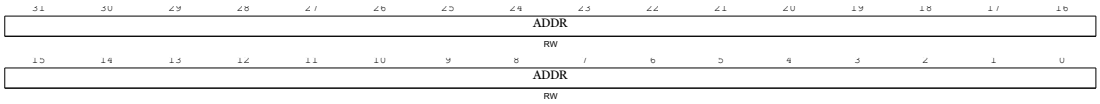
位/位域	名称	描述
31:8	保留	必须保持复位值
7	STA7	通道 7 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
6	STA6	通道 6 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
5	STA5	通道 5 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
4	STA4	通道 4 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
3	STA3	通道 3 读保护状态标志位,写入 1 清零

位/位域	名称	描述
		0:通道读保护未触发 1:通道读保护触发
2	STA2	通道 2 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
1	STA1	通道 1 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发
0	STA0	通道 0 读保护状态标志位,写入 1 清零 0:通道读保护未触发 1:通道读保护触发

14.6.6 MPU 保护通道 X 起始地址寄存器（MPU_SADDRx）

地址偏移：0x14 + x*0x8

复位值：0x00000000

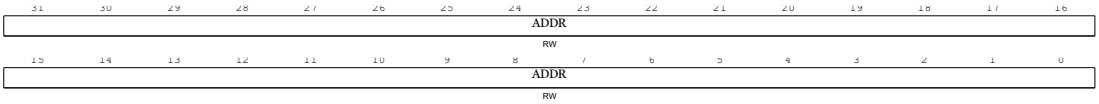


位/位域	名称	描述
31:0	ADDR	通道 x 保护开始地址 大于该地址（不包括该地址）的地址域将被 MPU 保护

14.6.7 MPU 保护通道 x 结束地址寄存器（MPU_EADDRx）

地址偏移：0x18 + x*0x8

复位值：0x00000000



位/位域	名称	描述
31:0	ADDR	通道 x 保护结束地址 小于该地址（不包括该地址）的地址域将被 MPU 保护

15 独立看门狗（IWDG）

15.1 简介

独立看门狗（IWDG）是一个独立定时器，一般用来检测系统软件程序是否按预期运行。如果看门狗模块没有被按时刷新，看门狗会产生错误上报给 FCU。比如程序中存在死循环，但因某些原因没有跳出，或者使用多任务操作系统，喂狗任务没有按时得到执行，都会由于看门狗模块没有被刷新而产生错误，一般用于高安全性 场合。

15.2 特性

- 内部晶振独立时钟源
- 可编程 8 位时钟预分频器
- 可编程 16 位时钟分频器
- 可编程 16 位超时时间计数
- 锁定安全机制

15.3 结构框图

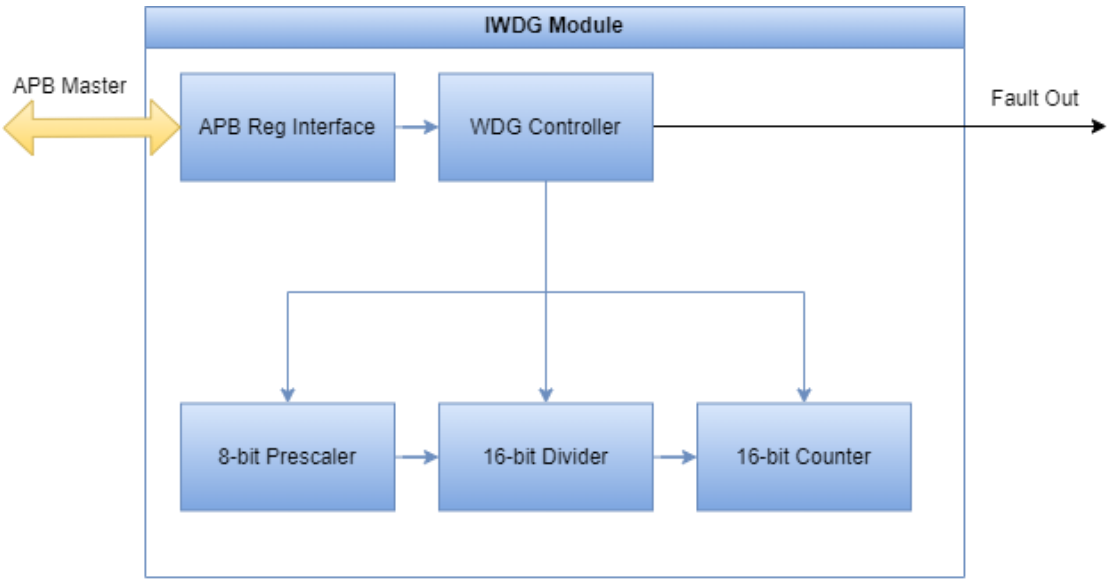


图 15.1 IWDG 模块结构框图

15.4 功能描述

15.4.1 基本看门狗

看门狗使用一个 16 位的向上计数器以及一个 8 位的预分频器、16 位的分频器进行计数。

看门狗使能后，开始计数，若计数到 BVAL 值，会产生错误信号，在计数达到 BVAL 值之前，刷新看门狗会使计数器复位并重新开始计数。

15.4.2 看门狗超时行为

看门狗在计数器的值超过 BVAL 后，会产生错误信号，用户可通过配置 FCU 模块对看门狗的报错处理行为进行约束。

15.5 应用说明

15.5.1 配置看门狗

当需要配置独立看门狗时，应按照如下步骤操作：

1. 配置预分频器：通过配置 IWDG_PSC 寄存器的 PSCNUM 位配置预分频系数，**系数最低为 2**，在配置完成后，配置 IWDG_PSC 的 EN 位为 1；
2. 配置分频器：通过配置 IWDG_DIV 寄存器的 DIVNUM 位配置分频系数，**系数最低为 2**；
3. 配置超时周期：配置 IWDG_TO 寄存器的 BVAL；
4. 启动看门狗：配置 IWDG_CFG 寄存器的 EN 位为 1，看门狗开始计数。

超时时间的计算方法如下：

$$T = （PSCNUM + 1） * DIVNUM * BVAL/32768$$

在配置时，可根据系统安全需要锁定预分频器（即向 IWDG_PSC 寄存器的 LOCK 位写入 1）、锁定看门狗（向 IWDG_CFG 寄存器的 LOCK 位写入 1）。

15.5.2 刷新看门狗

在使能看门狗后，程序应在看门狗超时前向 IWDG_CLR 寄存器写入 0x2f3e5d8a 来刷新看门狗的计数值，当刷新计数值之后，计数值归零，看门狗将开启新一轮的计数。

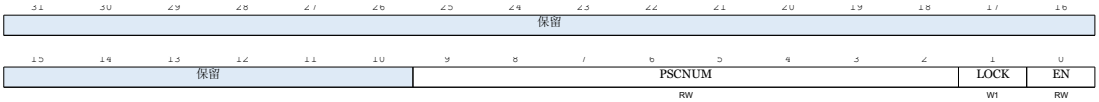
15.6 寄存器

独立看门狗基地址：0x30104000

15.6.1 IWDG 预分频寄存器（IWDG_PSC）

地址偏移：0x0

复位值：0x00000000

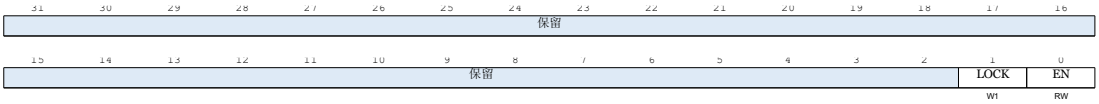


位/位域	名称	描述
31:10	保留	必须保持复位值
9:2	PSCNUM	独立看门狗预分频器分频系数配置 最小配置为 2
1	LOCK	独立看门狗预分频器锁定位 0:无效果 1:锁定预分频器寄存器
0	EN	独立看门狗预分频器使能 0:关闭预分频器 1:开启预分频器

15.6.2 IWDG 配置寄存器（IWDG_CFG）

地址偏移：0x4

复位值：0x00000000

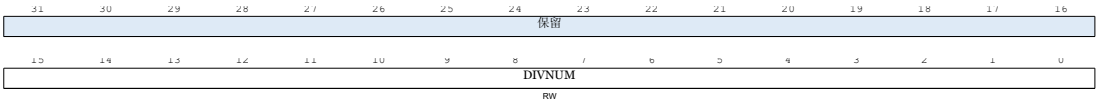


位/位域	名称	描述
31:2	保留	必须保持复位值
1	LOCK	独立看门狗锁定位 0:无效果 1:锁定对应的配置寄存器
0	EN	独立看门狗使能 0:不使能看门狗 1:使能看门狗

15.6.3 IWDG 分频寄存器（IWDG_DIV）

地址偏移：0x8

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DIVNUM	独立看门狗分频器配置 最小配置为 2

15.6.4 IWDG 超时寄存器（IWDG_TO）

地址偏移：0xC

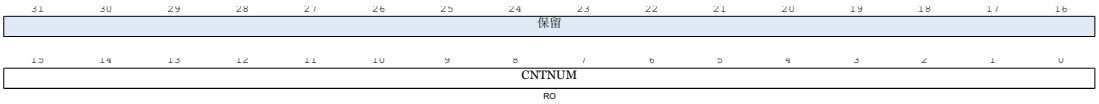
复位值：0x00000000

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BVAL	独立看门狗超时周期配置

15.6.5 IWDG 计数寄存器（IWDG_CNT）

地址偏移：0x10

复位值：0x00000000

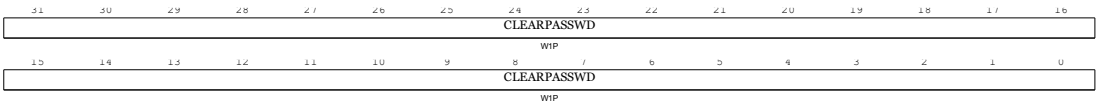


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNTNUM	独立看门狗计数器数值

15.6.6 IWDG 清除寄存器（IWDG_CLR）

地址偏移：0x14

复位值：0x00000000



位/位域	名称	描述
31:0	CLEARPASSWD	看门狗刷新位 必须写入 0x2f3e5d8a 才可刷新看门狗

16 窗口看门狗（WWDG）

16.1 简介

窗口看门狗（WWDG）是一个独立定时器，一般用来检测系统软件程序是否按预期运行。如果看门狗模块没有及时或过早刷新，看门狗会产生错误上报给FCU。比如程序中存在死循环，但因某些原因没有跳出，或者使用多任务操作系统，喂狗任务没有按时得到执行，都会由于看门狗模块没有被刷新而产生错误，一般用于高安全性场合。

16.2 特性

- 每个控制器 4 个独立通道
- 可编程 8 位时钟预分频器
- 可编程 16 位时钟分频器
- 可编程 16 位超时时间计数
- 可编程 16 位窗口时间计数
- 锁定安全机制

16.3 结构框图

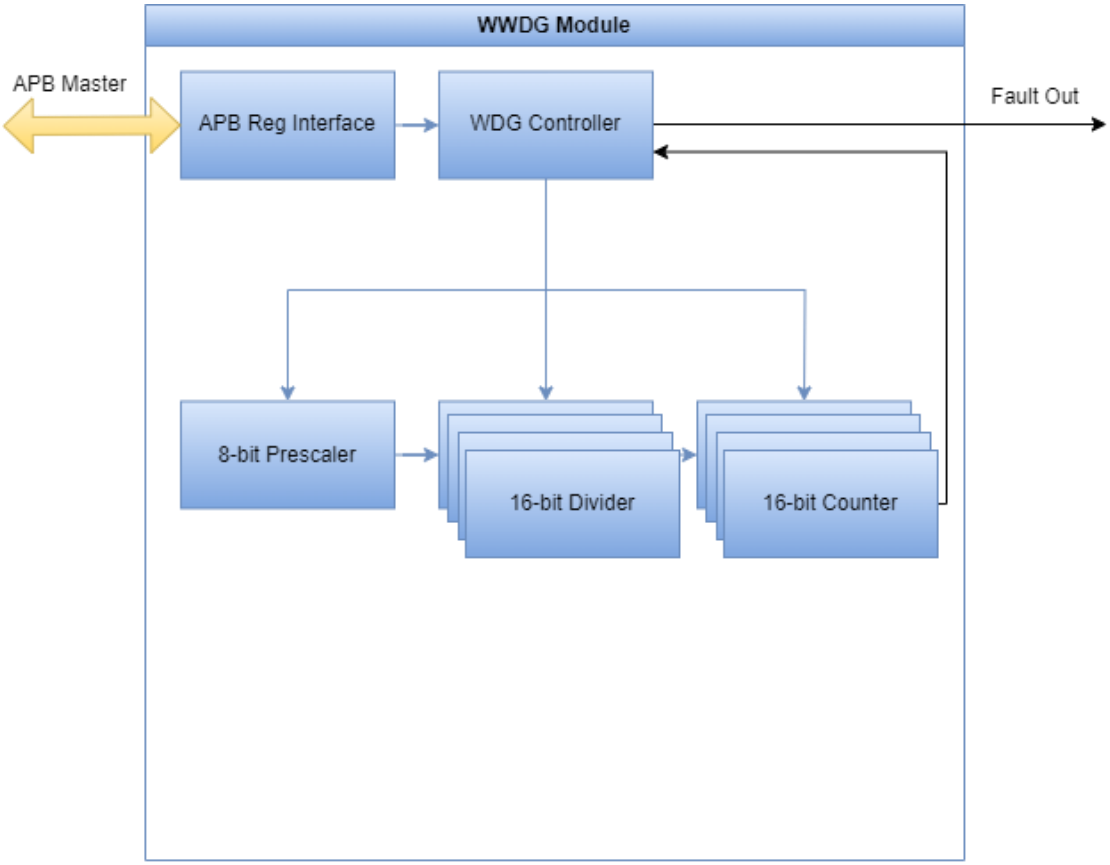


图 16.1 WWDG 模块结构框图

16.4 功能描述

16.4.1 基本看门狗

看门狗使用一个 16 位的向上计数器以及一个 8 位的预分频器、16 位的分频器进行计数。

看门狗使能后，开始计数，若：

- 计数到 BVAL 值
- 计数到 WDCNT 值之前刷新看门狗 窗口看门狗会产生错误信号，在计数器值在窗口值与 BVAL 值之间，刷新看门狗会使计数器复位并重新开始计数。

16.4.2 看门狗超时行为

看门狗在计数器的值超过 BVAL 后或计数到 WDCNT 值之前刷新看门狗时，会产生错误信号，用户可通过配置 FCU 模块对看门狗的报错处理行为进行约束。

16.5 应用说明

16.5.1 配置看门狗

当需要配置窗口看门狗通道时，应按照如下步骤操作：

1. 配置预分频器：通过配置 WWDG_PSC 寄存器的 PSCNUM 位配置预分频系数，**系数最低为 2**，在配置完成后，配置 WWDG_PSC 的 EN 位为 1；
2. 配置分频器：通过配置 WWDG_DIVx 寄存器的 DIVNUM 位配置分频系数，**系数最低为 2**；
3. 配置超时周期：配置 WWDG_TOx 寄存器的 BVAL；
4. 配置窗口值：若有窗口配置需要，通过配置 WWDG_WDx 寄存器的 WDCNT 位配置窗口计数系数，**配置值必须小于 BVAL**，在配置完成后，配置 WWDG_WDx 的 WDEN 位为 1；
5. 启动看门狗：配置 WWDG_CFGx 寄存器的 EN 位为 1，看门狗开始计数。

超时时间的计算方法如下：

$$T = (PSCNUM + 1) * DIVNUM * BVAL / f_b us$$

在配置时，可根据系统安全需要锁定预分频器（即向 WWDG_PSC 寄存器的 LOCK 位写入 1）、锁定对应看门狗通道（向 WWDG_CFGx 寄存器的 LOCK 位写入 1）。

16.5.2 刷新看门狗

在使能看门狗后，程序应在窗口超时后、看门狗超时前向 WWDG_CLRx 寄存器写入 0x2f3e5d8a 来刷新看门狗的计数值，当刷新计数值之后，计数值归零，看门狗将开启新一轮的计数。

16.6 寄存器

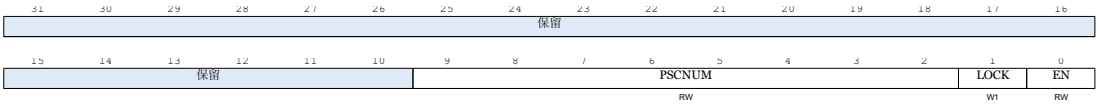
WWDG0 基地址：0x5000_D000

WWDG1 基地址：0x5100_C000

16.6.1 WWDG 预分频寄存器（WWDG_PSC）

地址偏移：0x0

复位值：0x00000000

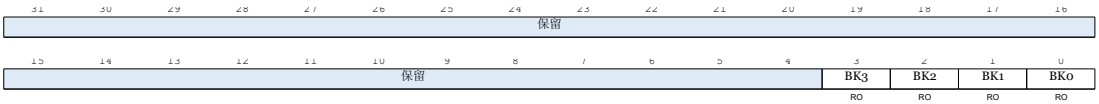


位/位域	名称	描述
31:10	保留	必须保持复位值
9:2	PSCNUM	窗口看门狗预分频器分频系数配置 最小配置为 2
1	LOCK	窗口看门狗预分频器锁定位 0:无效果 1:锁定预分频器寄存器
0	EN	窗口看门狗预分频器使能 0:关闭预分频器 1:开启预分频器

16.6.2 WWDG 状态寄存器（WWDG_DSTA）

地址偏移：0x4

复位值：0x00000000

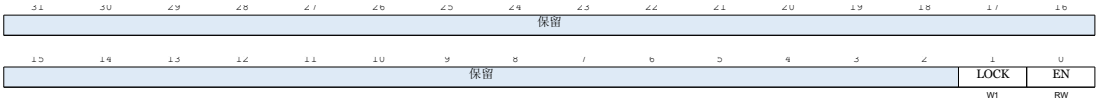


位/位域	名称	描述
31:4	保留	必须保持复位值
3	BK3	窗口看门狗通道 3 狗叫标志位
2	BK2	窗口看门狗通道 2 狗叫标志位
1	BK1	窗口看门狗通道 1 狗叫标志位
0	BK0	窗口看门狗通道 0 狗叫标志位

16.6.3 WWDG 通道 x 配置寄存器（WWDG_CFGx）

地址偏移：0x8 + x*0x18

复位值：0x00000000

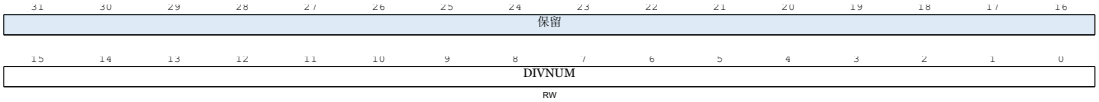


位/位域	名称	描述
31:2	保留	必须保持复位值
1	LOCK	窗口看门狗 x 锁定位 0:无效果 1:锁定对应的配置寄存器
0	EN	窗口看门狗 x 使能 0:不使能看门狗 1:使能看门狗

16.6.4 WWDG 通道 x 分频寄存器（WWDG_DIVx）

地址偏移：0xC + x*0x18

复位值：0x00000000

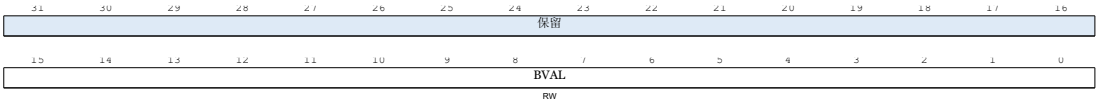


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DIVNUM	窗口看门狗 x 分频器配置 最小配置为 2

16.6.5 WWDG 通道 x 超时寄存器（WWDG_TOx）

地址偏移：0x10 + x*0x18

复位值：0x00000000

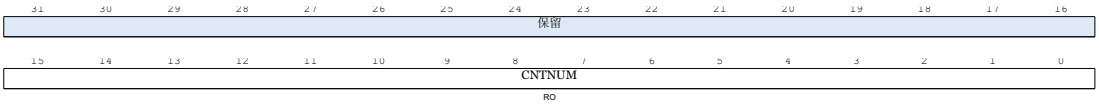


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BVAL	窗口看门狗 x 超时周期配置

16.6.6 WWDG 通道 x 计数寄存器（WWDG_CNTx）

地址偏移：0x14 + x*0x18

复位值：0x00000000

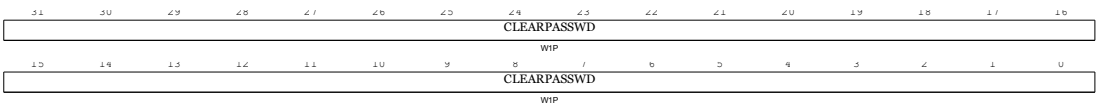


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNTNUM	窗口看门狗 x 计数器数值

16.6.7 WWDG 通道 x 清除寄存器（WWDG_CLRx）

地址偏移：0x18

复位值：0x00000000

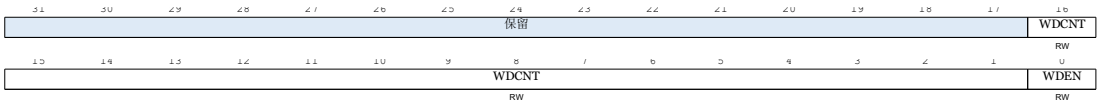


位/位域	名称	描述
31:0	CLEARPASSWD	看门狗 x 刷新位 必须写入 0x2f3e5d8a 才可刷新看门狗

16.6.8 WWDG 通道 x 窗口寄存器（WWDG_WDx)

地址偏移：0x1C + x*0x18

复位值：0x00000000



位/位域	名称	描述
31:17	保留	必须保持复位值
16:1	WDCNT	窗口看门狗通道 x 窗口周期配置 配置值必须小于超时周期
0	WDEN	窗口看门狗通道 x 窗口使能 0:关闭窗口计数 1:打开窗口计数

17 循环冗余校验（CRC）

17.1 简介

循环冗余校验（Cyclic Redundancy Check，CRC）是一种根据网络数据包或计算机文件等数据产生简短固定位数校验码的一种信道编码技术，主要用来检测或校验数据传输或者保存后可能出现的错误。它是利用除法及余数的原理来作错误侦测的。

CRC（循环冗余校验）计算单元使用一个多项式发生器从 8 位/16 位/32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

17.2 主要特征

- 使用位数可预设定的（7 位、8 位、16 位和 32 位）完全可编程多项式；
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 APB 时钟周期（PCLK）内完成
- 8 位通用寄存器（可用于临时存储）
- I/O 数据的可逆性选项
- 支持异或计算

17.3 功能框图

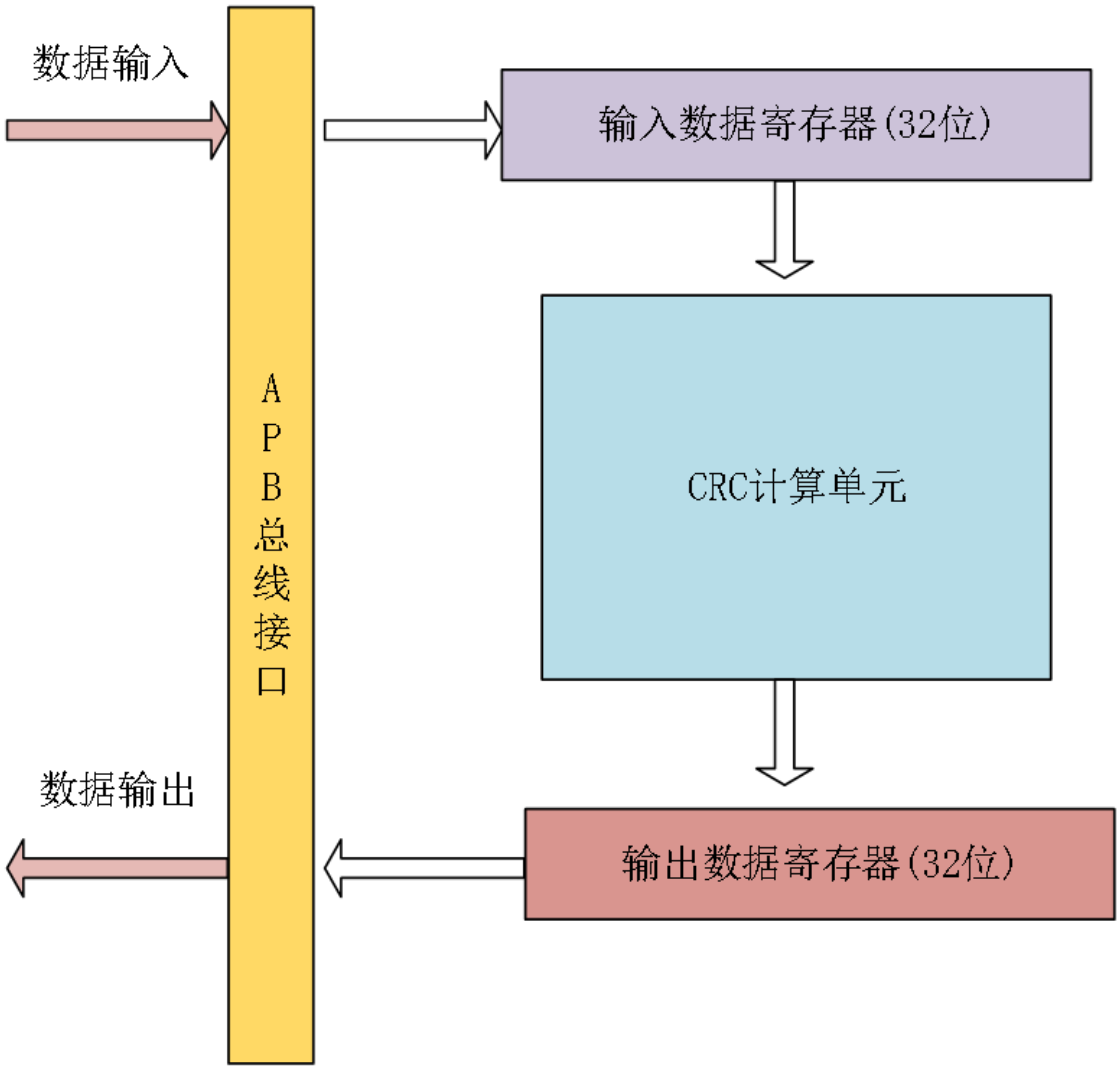


图 17.1 CRC 结构框图

17.4 功能说明

17.4.1 使用说明

CRC 计算单元具有单个 32 位读/写数据寄存器（CRC_DR）。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其它寄存器，只允许进行 32 位访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 APB 时钟周期
- 16 位数据需要 2 个 APB 时钟周期
- 8 位数据需要 1 个 APB 时钟周期

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待任何等待状态。

可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

输入数据的顺序可反转，以管理各种数据存放方式（双字/单字/字节，大端/小端等等）。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0]位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

- 按字节执行位反转的 0x58D43CB2
- 按半字执行位反转的 0xD458B23C
- 按全字执行位反转的 0xB23CD458

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转

该操作按位进行：例如，输出数据 0x11223344 将转换为 0x22CC4488。

使用 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写访问时会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中的 RESET 位影响。

多项式编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0]位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶数多项式。

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读操作。

默认的多项式值为 CRC-32（以太网）多项式：0x4C11DB7

17.4.2 使用例程

单次计算

- 控制寄存器配置，初始化配置
- 初始化完成配置初始向量寄存器
- 配置生成多项式寄存器
- 配置数据寄存器（写入待计算的数据）
- 读出数据- 连续计算
- 控制寄存器配置，初始化配置
- 初始化完成配置初始向量寄存器
- 配置生成多项式寄存器
- 配置数据寄存器（写入第一个的数据）
- 配置数据寄存器（写入第二个的数据）
- 配置数据寄存器（写入第三个的数据）
- ……
- 配置数据寄存器（写入第 n 个的数据）
- 读出数据

17.4.3 常用多项式

表 17.1 CRC 生成多项式

名称	多项式	表示法
CRC-8	$x^8+x^2+x^1+1$	0x107
CRC-12	$x^{12}+x^{11}+x^3+x^2+x^1+1$	0x180F
CRC-16	$x^{16}+x^{15}+x^2+1$	0x18005
CRC-CCITT	$x^{16}+x^{15}+x^5+1$	0x11021
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	0x104C11DB7
CRC-32C	$x^{32}+x^{28}+x^{27}+x^{25}+x^{23}+x^{22}+x^{21}+x^{20}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^9+x^8+x^6+1$	0x11EDC6F41

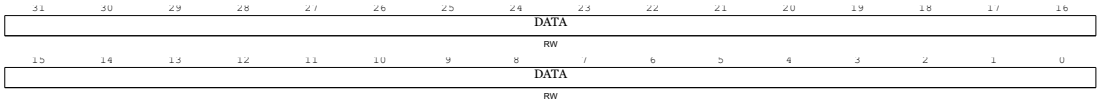
17.5 寄存器

CRC 基地址：0x5100d000

17.5.1 CRC 数据寄存器(CRC_DR)

地址偏移：0x0

复位值：0x00000000

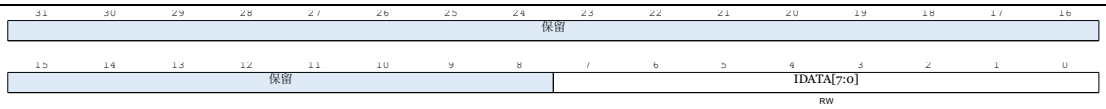


位/位域	名称	描述
31:0	DATA	CRC 计算结果位 软件可读可写 该寄存器用于接收待计算的新数据，直接将其写入即可。写入的数据可立即读出

17.5.2 CRC 独立数据寄存器(CRC_IDR)

地址偏移：0x4

复位值：0x00000000

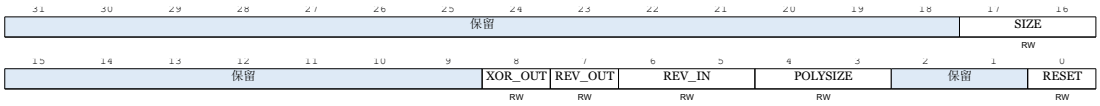


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	IDATA[7:0]	通用的 8 位数据寄存器位 这些位可用作一个字节的临时存储单元。 此寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响

17.5.3 CRC 控制寄存器(CRC_CR)

地址偏移：0x8

复位值：0x00000000



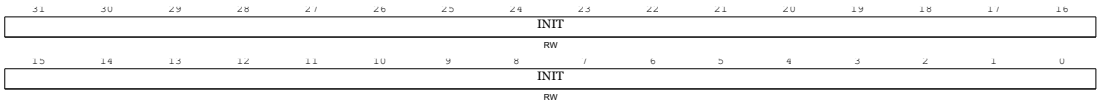
位/位域	名称	描述
31:18	保留	必须保持复位值
17:16	SIZE	计算数据的大小 2’b00：8bit 2’b01：16bit 2’b10：32bit
15:9	保留	必须保持复位值
8	XOR_OUT	CRC计算结果与异或寄存器（CRC_XOR）进行异或使能位 0：不异或 1：异或
7	REV_OUT	反转输出数据（Reverse output data） 该位用于控制输出数据位顺序的反转。 0：不影响位顺序

位/位域	名称	描述
		1：位反转输出格式
6:5	REV_IN	反转输入数据（Reverse input data） 这些位用于控制输入数据位顺序的反转。 00：不影响位顺序 01：按字节执行位反转 10：按半字执行位反转 11：按字执行位反转
4:3	POLYSIZE	多项式大小 00：32 位多项式 01：16 位多项式 10：8 位多项式 11：7 位多项式
2:1	保留	必须保持复位值
0	RESET	此位由软件置 1，用于复位 CRC 计算单元并将数据寄存器 DR 设置为存储在 CRC_INIT 寄存器中的值。 此位只能置 1，将由硬件自动进行清零

17.5.4 CRC 初始向量寄存器(CRC_INIT)

地址偏移：0xc

复位值：0xffffffff

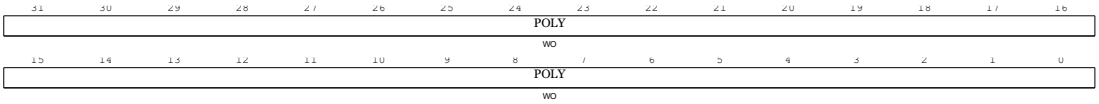


位/位域	名称	描述
31:0	INIT	初始向量寄存器 如果 CRC 多项式不是 32bit，则低位有效

17.5.5 CRC 生成多项式寄存器(CRC_POLY)

地址偏移：0x10

复位值：0x00000000

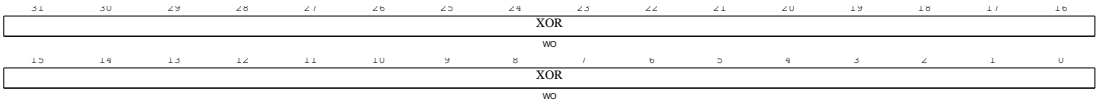


位/位域	名称	描述
31:0	POLY	生成多项式寄存器 如果 CRC 多项式不是 32bit，则低位有效

17.5.6 CRC 异或多项式寄存器(CRC_XOR)

地址偏移：0x14

复位值：0x00000000



位/位域	名称	描述
31:0	XOR	异或寄存器 如果 CRC 多项式不是 32bit，则低位有效（高位为 0）

18 时钟监控模块（CMU）

18.1 检测

时钟监控单元(Clock Monitor Unit, CMU)主要有两大功能：

- 检测时钟频率的偏差是否超出最大的允许范围；
- 检测时钟是否丢失。

18.2 主要特征

- 如果被监控的频率大于参考时钟的最大值（HFREF），产生一个频率大于高阈值（FHH）事件
- 如果被监控的频率小于参考时钟的最小值（LFREF），产生一个频率小于低阈值（FLL）事件
- 当被监控的时钟丢失时，会产生 FLL 事件
- 参考时钟计数的窗口时间支持软件配置

18.3 功能说明

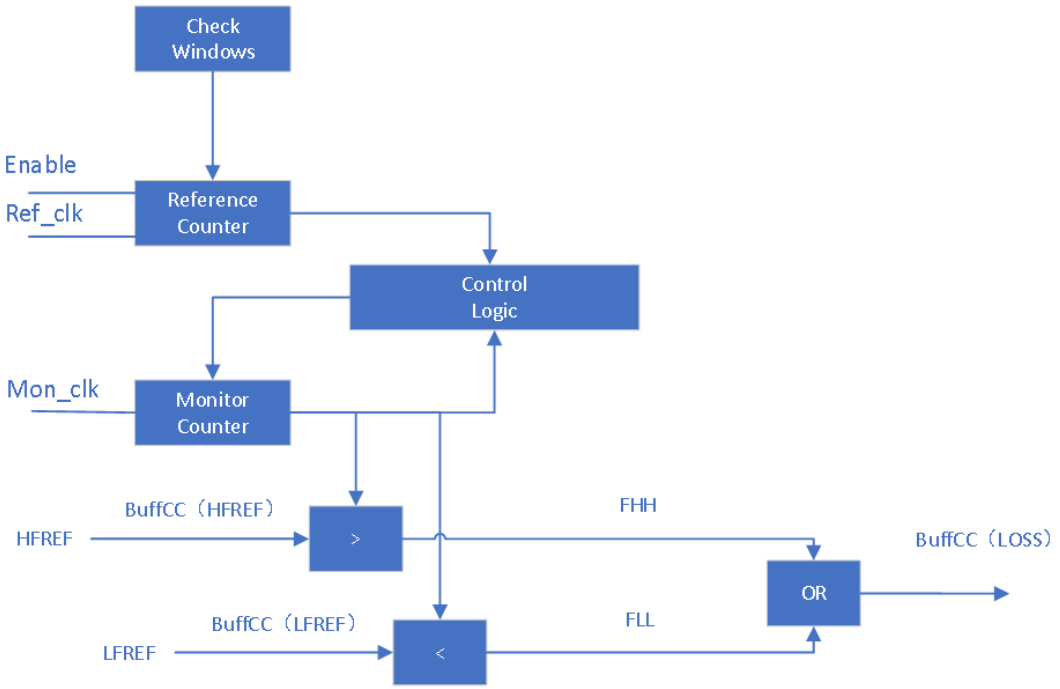


图 18.1 结构框图

该模块的主要输入信号有两个：参考时钟 cmuClk_ref 和被监测时钟 cmuClk_mon。CMU 模块会把监控的实时计数值和用户配置的 2 个阈值进行比较，然后产生 FHH（实际计数值大于配置的高阈值的值）或 FLL（实际计数值小于配置的低阈值的值）事件。

18.4 应用说明

1. 配置 CMU 寄存器 RCRR，计数器配置寄存器
2. 配置 CMU 寄存器 HTCR，高阈值配置寄存器
3. 配置 CMU 寄存器 LTCR，低阈值配置寄存器
4. 配置 CMU 寄存器 CR[2:0]为非 0，表示启动 CMU 模块
5. 等待中断，如果出现中断，这表示 FHH 时间或 FLL 事件触发
6. 读取 CMU 寄存器 SR，并判断中断的类型 7 .配置 CMU 就寄存器 CR[2:0]为 0，停止 CMU 工作。

假设：

1. 要求检测时间为 5us，
2. 参考时钟为 5MHz，则 Tref = 1000us/5MHz
3. 被监测时钟为 24MHz，则 Tmon=1000us/24MHz。

18.4.1 窗口计数器的计算

WINLEN=5us/Tref=25。参考时钟域下的计数器的计数值为 25。窗口计数器的最小值为 10。

18.4.2 阈值的计算

高低阈值的配置需要考虑下面的因素：

1. 被监控的时钟频率可容忍的偏差
2. 参考时钟频率的偏差
3. CMU 模块本身的偏差。

理想情况下被监测时钟域的计数器的值为：Monitor Counter=5us/Tmon=120。考虑到参考时钟和被监控时钟的误差为正负 5%。所以：

最小阈值为 Monitor Counter*（100% -5%）=114

最大阈值为 Monitor Counter*（100%+5%）=126

18.5 寄存器

CMU_FIRC 基地址：0x30003000

CMU_PLLQ 基地址：0x30003400

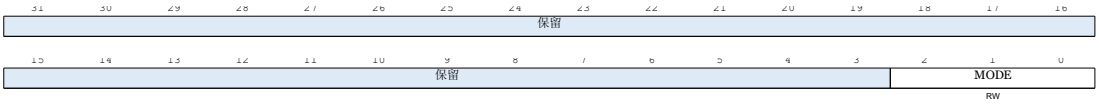
CMU_PLLR 基地址：0x30003800

CMU_OSC 基地址：0x30003C00

18.5.1 CMU 控制寄存器(CMU_CR)

地址偏移：0x0

复位值：0x00000000

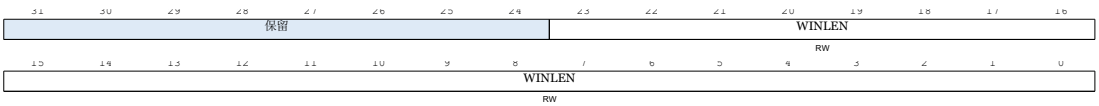


位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	MODE	时钟监控使能 3’b000:不是能时钟检测 非 0 值:时钟监测使能有效

18.5.2 CMU 参考计数配置寄存器(CMU_RCCR)

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	WINLEN	参考计数窗口长度 检测运行的参考时钟计数器的数据目。定义参考平率检查窗口的时序时间

18.5.3 CMU 高阈值配置寄存器(CMU_HTCR)

地址偏移：0x8

复位值：0x00000000

位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	HFREF	高频参考阈值 被检测时钟的时钟频率的高参考阈值。在监控的时间窗口内，如果检测的时钟计数大于该阈值，则产生 FHH 时间

18.5.4 CMU 低阈值配置寄存器(CMU_LTCR)

地址偏移：0xC

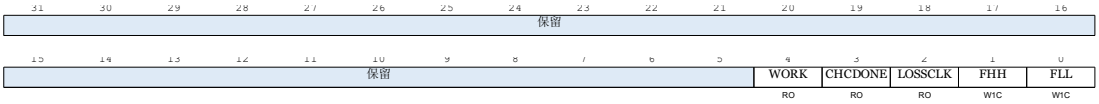
复位值：0x00000000

位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	LFREF	低频参考阈值 被检测时钟的时钟频率的低参考阈值。在监控的时间窗口内，如果检测的时钟计数小于该阈值，则产生 FLL 事件

18.5.5 CMU 状态寄存器(CMU_SR)

地址偏移：0x10

复位值：0x00000000



位/位域	名称	描述
31:5	保留	必须保持复位值
4	WORK	工作状态 0:时钟检测未工作 1:时钟监测正在工作
3	CHCDONE	状态检测 0:模块未完成一次时钟监测 1:模块完成至少一次时钟监测
2	LOSSCLK	时钟丢失状态 0:时钟没有丢失 时钟已丢失 说明，FFH 和 FLL 时间产生，都会触发时钟丢失状态置位
1	FHH	频率大于高频参考阈值状态 0:无 FHH 时间 1:FHH 时间已产生
0	FLL	频率小于低频率参考阈值状态 0:无 FLL 事件 1:FLL 时间已产生

19 通用型输入输出接口（GPIO）

19.1 简介

片上设备用 GPIO 来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。外设 GPIO 引脚上都有其独立的中断控制。GPIO 端口和其他的备用功能（Afs）共用引脚，在特定的封装下获得最大的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能引脚，备用功能输入/输出都可。 每个 GPIO 引脚可以由软件配置为输出（推挽或开漏）、输入、外设的备用功能或者模拟模式。 每个 GPIO 引脚都可以配置为上拉、下拉或无上拉/下拉。

19.2 主要特征

- 输入/输出方向控制
- 触发器输入功能使能控制
- 每个引脚都具有弱上拉/下拉功能
- 推挽/开漏输出使能控制
- 置位/复位输出使能
- 可编程沿触发/电平触发的中断
- 模拟输入/输出配置
- 备用功能输入/输出配置

19.3 功能描述

每个通用 I/O 端口都可以通过一个 32 位的控制寄存器（GPIOx_CTRL）和一个 16 位的读写模式控制器（GPIOx_MODE）配置为 8 种模式：模拟输入，浮空输入，上拉输入，下拉输入，推挽输出，开漏输出，复用推挽输出和复用开漏输出。详情请见下表。

表 19.1 GPIO 配置表

配置模式		CTRL[1:0]	MODE
输入	模拟	00	0

配置模式		CTRL[1:0]	MODE
	浮空输入	01	0
	下拉输入	10	0
	上拉输入	11	0
普通输出	推挽	00	1
	开漏	01	1
复用输出	推挽	10	1
	开漏	11	1

下图为标准 I/O 口味的基本结构图。

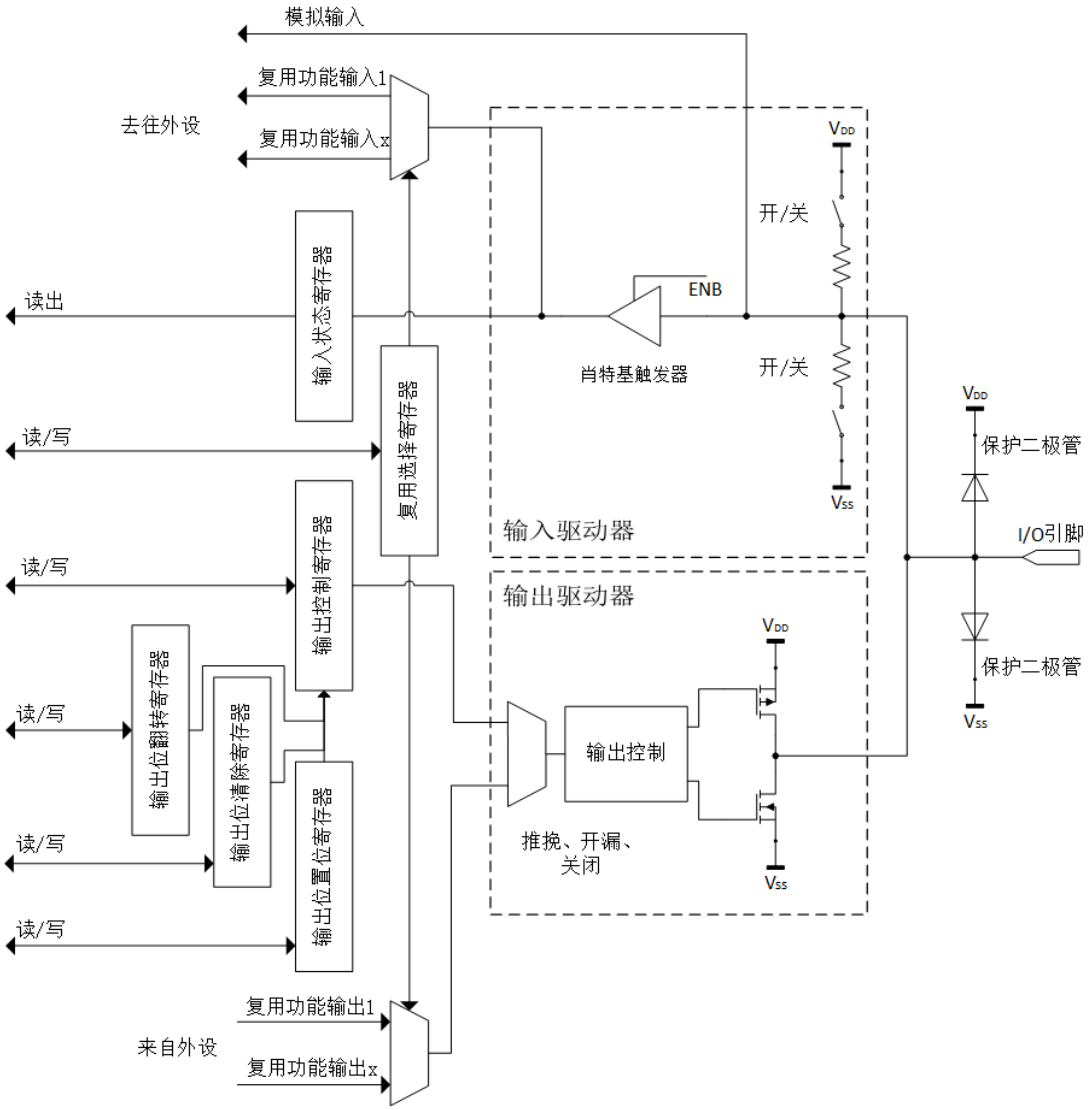


图 19.1 标准 I/O 端口位的基础结构

下图为 GPIO 控制器的结构图。

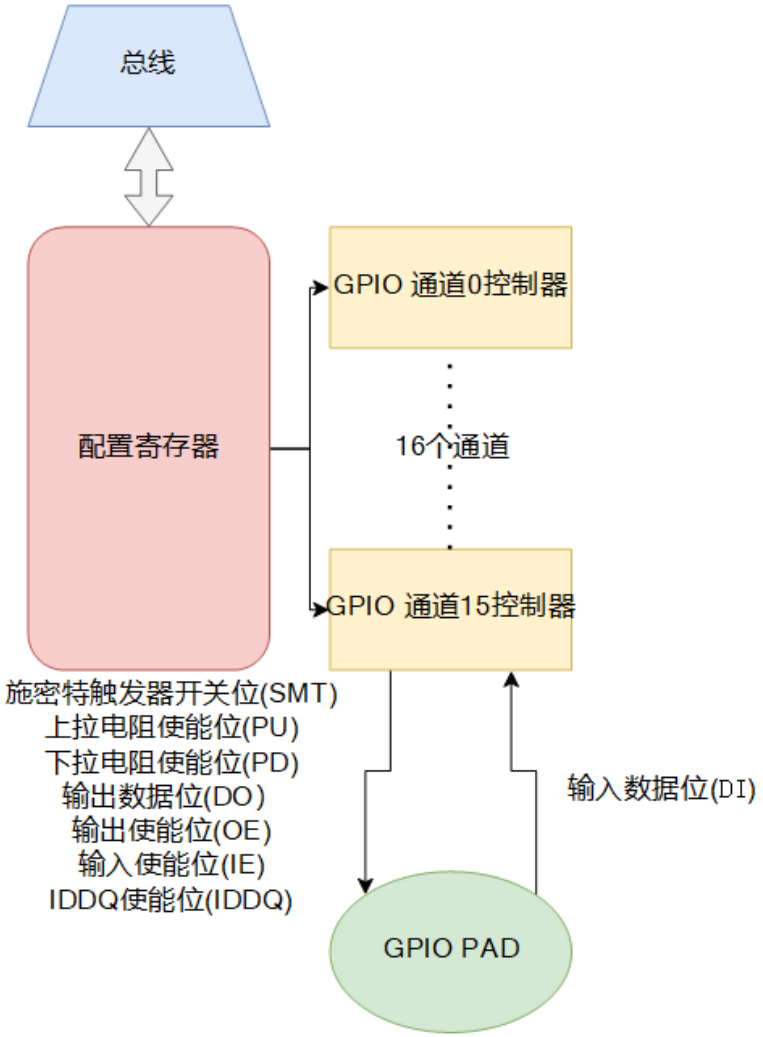


图 19.2 标准 I/O 端口位的基础结构

19.4 GPIO 引脚配置

在复位期间或复位之后，默认配置为复用功能 0，此时 GPIO 被配置为浮空输入模式。 GPIO 引脚可以配置为输入或输出模式，当 GPIO 引脚配置为输入引脚时，所有的 GPIO 引脚内部都有一个可选择的弱上拉和弱下拉电阻。外部引脚上的数据在每个时钟周期都会装载到读端口寄存器（GPIOx_RD）。 当 GPIO 引脚配置为输出引脚，用户可以配置端口的输出速度和选择输出驱动模式：推挽或开漏模式，输出控制寄存器（GPIOx_OC）的值将会从相应 I/O 引脚上输出。

19.4.1 端口复用

当端口设置为复用模式（设置 GPIOx_PM 寄存器中的 pmy 值为非“0b000”，GPIOx_RW 寄存器中的 rwy 值为 0）时，该端口用作外设复用引脚。

19.4.2 输入配置

当 GPIO 引脚配置为输入时：

- 触发器输入使能；
- 可选择的弱上拉和下拉电阻；
- 当前 I/O 引脚上的数据在每个时钟周期都会被采样并存入端口状态寄存器；
- 输出缓冲器禁用。

19.4.3 输出配置

当 GPIO 引脚配置为输出时：

- 触发器输入使能；
- 弱上拉和下拉电阻禁用；
- 输出缓冲器使能；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应引脚输出高电平；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值；
- 对端口输入状态寄存器进行读操作，将获得当前 I/O 口的状态。

19.4.4 模拟配置

当 GPIO 引脚用于模拟模式时：

- 弱上拉和上拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发输入禁用；

- 端口输入状态寄存器相应位为“0”。

19.4.5 端口复用配置

为适应不同的器件封装，GPIO 端口支持软件配置将一些备用功能应用到其他引脚上。 当引脚配置为备用功能时：

- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 在输入配置时，可选择弱上拉/下拉电阻；
- I/O 引脚上的数据在每个外设时钟周期采样。

19.5 应用说明

每个 I/O 引脚都可以配置很多功能，每个功能都是通过对 GPIO 寄存器配置实现的。

19.5.1 输入配置

当需要将一个 GPIO 引脚配置为输入时，请遵循以下操作：

1. 将 GPIOx_RW 寄存器的 rwy 寄存器设置为“0”；
2. 根据上下拉需求，配置 GPIOx_IM 的 imy 位，浮空输入配置为“0”，上拉输入配置为“1”，下拉输入配置为“2”；
3. 通过读取 GPIOx_RD 寄存器的 rdy 位读取引脚输入值；

19.5.2 输出配置

当需要将一个 GPIO 引脚配置为输出时，请遵循以下操作：

1. 将 GPIOx_RW 寄存器的 rwy 寄存器设置为“1”；
2. 根据输出需求配置 GPIOx_OM 寄存器的 omy 位，开漏输出设置为“0”，推挽输出配置为“1”；
3. 根据需求配置 GPIOx_OSTR 寄存器的 ostry 位；
4. 通过配置 GPIOx_OBS、GPIOx_OBC、GPIOx_OBT、GPIOx_OC 寄存器来控制 IO 输出；

19.5.3 数字复用配置

当 GPIO 引脚需要配置为数字外设复用时，请遵循以下操作：

1. 根据输入/输出需求配置输入输出寄存器，具体配置参考输入配置的 1~2 步，输出配置的 1~3 步；
2. 配置 GPIOx_PM 寄存器的 pmy 位选择对应外设，具体配置请参考 IO 复用表。

19.5.4 模拟复用配置

当 GPIO 引脚需要配置为模拟外设复用时，将对应引脚 GPIOx_RW 寄存器的 rwy 位配置为”2”即可切换为模拟复用模式；

19.5.5 中断控制

每个 I/O 引脚都有独立的中断控制单元。通过配置 GPIOx_ITYPE 中的相应 itypey 位可以选择不同的中断模式，具体情况参照下表。

表 19.2 中断控制位配置

中断类型	配置位
关闭中断	0b00
上升沿触发	0b01
下降沿触发	0b10
所有沿触发	0b11

具体配置参考以下步骤：

1. 清除并配置 PLIC；
2. 读取 GPIOx_INT_RAW 寄存器的 giyraw 位，若为 1 则写 1 清除后重复执行步骤 1，若为 0 则继续以下步骤；
3. 配置 GPIOx_ITYPE 中的 itypey 位；
4. 在配置中断模式后，将 GPIOx_INT_MASK 的 giymask 位配置为 0，开启中断；
5. 触发中断后，读取 GPIOx_INT_RAW 寄存器判断中断引脚；

6. 处理中断后，通过向 GPIOx_INT_RAW 的 giyraw 位写入 1 清除中断。

19.6 寄存器（GPIOA~GPIOG）

GPIOA 基地址：0x41004000

GPIOB 基地址：0x41005000

GPIOC 基地址：0x41006000

GPIOD 基地址：0x42003000

GPIOE 基地址：0x42004000

GPIOF 基地址：0x42005000

GPIOG 基地址：0x42006000

19.6.1 GPIOx 读写模式控制寄存器（GPIOx_RW）

地址偏移：0x0

GPIOA 复位值：0x05550000

GPIOB 复位值：0x00001000

GPIOC 复位值：0x40000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00055555

GPIOG 复位值：0x00000000

位/位域	名称	描述
31:30	rw15	通道 15 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。

位/位域	名称	描述
29:28	rw14	通道 14 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
27:26	rw13	通道 13 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
25:24	rw12	通道 12 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
23:22	rw11	通道 11 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
21:20	rw10	通道 10 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
19:18	rw9	通道 9 读写模式控制，读写，

位/位域	名称	描述
		0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
17:16	rw8	通道 8 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
15:14	rw7	通道 7 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
13:12	rw6	通道 6 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
11:10	rw5	通道 5 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
9:8	rw4	通道 4 读写模式控制，读写， 0：数字输入模式，此时 IM 有效；

位/位域	名称	描述
		1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
7:6	rw3	通道 3 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
5:4	rw2	通道 2 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
3:2	rw1	通道 1 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
1:0	rw0	通道 0 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。

19.6.2 GPIOx 输入模式控制寄存器（GPIOx_IM）

地址偏移：0x4

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

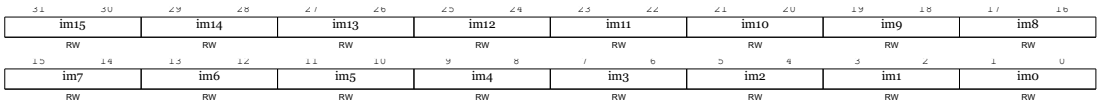
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:30	im15	通道 15 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
29:28	im14	通道 14 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
27:26	im13	通道 13 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
25:24	im12	通道 12 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式；

位/位域	名称	描述
		2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
23:22	im11	通道 11 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
21:20	im10	通道 10 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
19:18	im9	通道 9 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
17:16	im8	通道 8 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
15:14	im7	通道 7 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式；

位/位域	名称	描述
		3：KEEP 模式（上下拉全拉）。
13:12	im6	通道 6 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
11:10	im5	通道 5 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
9:8	im4	通道 4 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
7:6	im3	通道 3 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
5:4	im2	通道 2 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。

位/位域	名称	描述
3:2	im1	通道 1 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
1:0	im0	通道 0 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。

19.6.3 GPIOx 输出模式控制器（GPIOx_OM）

地址偏移：0x8

GPIOA 复位值：0x00000300

GPIOB 复位值：0x00000040

GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x000003f0

GPIOG 复位值：0x00000000

位/位域	名称	描述
31:16	保留	必须保持复位值
15	om15	通道 15 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）；

位/位域	名称	描述
		1：推挽输出（低电平高电平都输出）。
14	om14	通道 14 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
13	om13	通道 13 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
12	om12	通道 12 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
11	om11	通道 11 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
10	om10	通道 10 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
9	om9	通道 9 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
8	om8	通道 8 输出模式控制，读写，

位/位域	名称	描述
		0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
7	om7	通道 7 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
6	om6	通道 6 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
5	om5	通道 5 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
4	om4	通道 4 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
3	om3	通道 3 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
2	om2	通道 2 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）；

位/位域	名称	描述
		1：推挽输出（低电平高电平都输出）。
1	om1	通道 1 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
0	om0	通道 0 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。

19.6.4 GPIOx 输出驱动强度寄存器（GPIOx_OSTR）

地址偏移：0xC

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

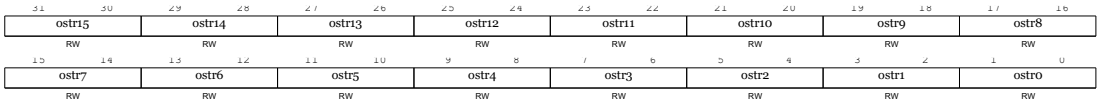
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:30	ostr15	通道 15 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA；

位/位域	名称	描述
		3：输出最大驱动电流 18mA；
29:28	ostr14	通道 14 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
27:26	ostr13	通道 13 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
25:24	ostr12	通道 12 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
23:22	ostr11	通道 11 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ；

位/位域	名称	描述
		2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
21:20	ostr10	通道 10 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
19:18	ostr9	通道 9 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
17:16	ostr8	通道 8 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
15:14	ostr	通道 7 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ；

位/位域	名称	描述
		1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
13:12	ostr6	通道 6 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
11:10	ostr5	通道 5 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
9:8	ostr4	通道 4 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
7:6	ostr3	通道 3 输出强度控制，GPIO 数字输出模式/复用时有效，读写，

位/位域	名称	描述
		0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
5:4	ostr2	通道 2 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
3:2	ostr1	通道 1 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
1:0	ostr0	通道 0 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；

19.6.5 GPIOx 复用接口选择寄存器 0（GPIOx_PM0）

地址偏移：0x10

GPIOA 复位值：0x09000000

GPIOB 复位值：0x09248000

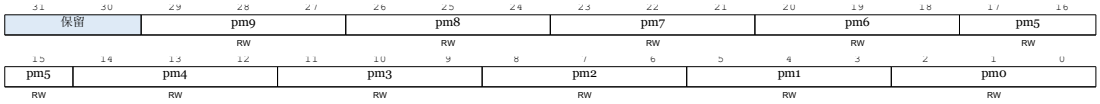
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x09249249

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:30	保留	必须保持复位值
29:27	pm9	通道 9 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
26:24	pm8	通道 8 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
23:21	pm7	通道 7 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）；

位/位域	名称	描述
		注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
20:18	pm6	通道 6 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
17:15	pm5	通道 5 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
14:12	pm4	通道 4 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
11:9	pm3	通道 3 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
8:6	pm2	通道 2 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）；

位/位域	名称	描述
		注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
5:3	pm1	通道 1 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
2:0	pm0	通道 0 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。

19.6.6 GPIOx 复用接口选择寄存器 1（GPIOx_PM1）

地址偏移：0x14

GPIOA 复位值：0x00000249

GPIOB 复位值：0x00000000

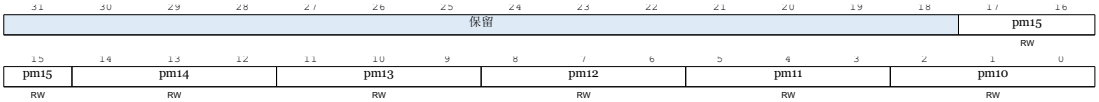
GPIOC 复位值：0x00009200

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:18	保留	必须保持复位值
17:15	pm15	通道 15 复用接口选择，只写，

位/位域	名称	描述
		0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
14:12	pm14	通道 14 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
11:9	pm13	通道 13 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
8:6	pm12	通道 12 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
5:3	pm11	通道 11 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
2:0	pm10	通道 10 复用接口选择，只写， 0：GPIO 输出；

位/位域	名称	描述
		1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。

19.6.7 GPIOx 输出置位寄存器（GPIOx_OBS）

地址偏移：0x18

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000

位/位域	名称	描述
31:16	保留	必须保持复位值
15	obs15	通道 15 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
14	obs14	通道 14 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
13	obs13	通道 13 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
12	obs12	通道 12 输出置位，只写，

位/位域	名称	描述
		0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
11	obs11	通道 11 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
10	obs10	通道 10 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
9	obs9	通道 9 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
8	obs8	通道 8 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
7	obs7	通道 7 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
6	obs6	通道 6 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
5	obs5	通道 5 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
4	obs4	通道 4 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。

位/位域	名称	描述
3	obs3	通道 3 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
2	obs2	通道 2 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
1	obs1	通道 1 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
0	obs0	通道 0 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。

19.6.8 GPIOx 输出清除寄存器（GPIOx_OBC）

地址偏移：0x1C

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

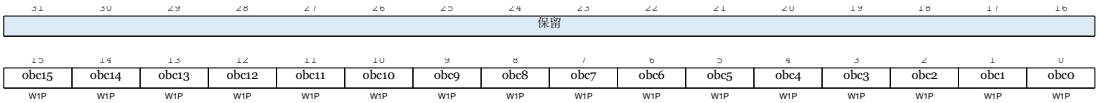
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	obc15	通道 15 输出清除，只写，

位/位域	名称	描述
		0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
14	obc14	通道 14 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
13	obc13	通道 13 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
12	obc12	通道 12 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
11	obc11	通道 11 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
10	obc10	通道 10 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
9	obc9	通道 9 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
8	obc8	通道 8 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
7	obc7	通道 7 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。

位/位域	名称	描述
6	obc6	通道 6 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
5	obc5	通道 5 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
4	obc4	通道 4 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
3	obc3	通道 3 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
2	obc2	通道 2 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
1	obc1	通道 1 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
0	obc0	通道 0 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。

19.6.9 GPIOx 输出翻转寄存器（GPIOx_OBT)

地址偏移：0x20

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

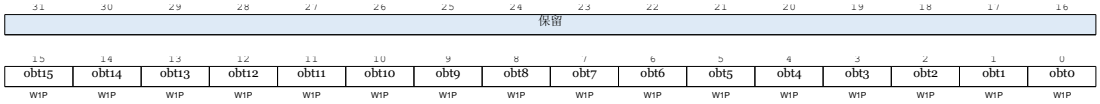
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	obt15	通道 15 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
14	obt14	通道 14 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
13	obt13	通道 13 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
12	obt12	通道 12 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
11	obt11	通道 11 输出翻转，只写， 0：保持当前状态，

位/位域	名称	描述
		1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
10	obt10	通道 10 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
9	obt9	通道 9 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
8	obt8	通道 8 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
7	obt7	通道 7 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
6	obt6	通道 6 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
5	obt5	通道 5 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。

位/位域	名称	描述
4	obt4	通道 4 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
3	obt3	通道 3 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
2	obt2	通道 2 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
1	obt1	通道 1 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
0	obt0	通道 0 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。

19.6.10 GPIOx 输出控制寄存器（GPIOx_OC）

地址偏移：0x24

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

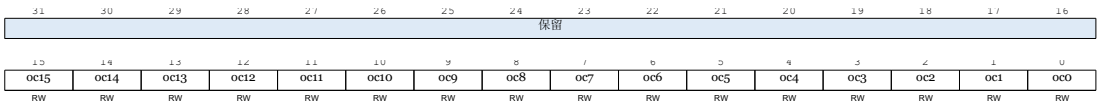
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	oc15	通道 15 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
14	oc14	通道 14 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
13	oc13	通道 13 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
12	oc12	通道 12 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
11	oc11	通道 11 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
10	oc10	通道 10 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
9	oc9	通道 9 输出控制，读写， 0：当前输出设置为低电平；

位/位域	名称	描述
		1：当前输出设置为高电平。
8	oc8	通道 8 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
7	oc7	通道 7 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
6	oc6	通道 6 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
5	oc5	通道 5 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
4	oc4	通道 4 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
3	oc3	通道 3 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
2	oc2	通道 2 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
1	oc1	通道 1 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
0	oc0	通道 0 输出控制，读写，

位/位域	名称	描述
		0：当前输出设置为低电平； 1：当前输出设置为高电平。

19.6.11 GPIOx 输入读取寄存器（GPIOx_RD）

地址偏移：0x28

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

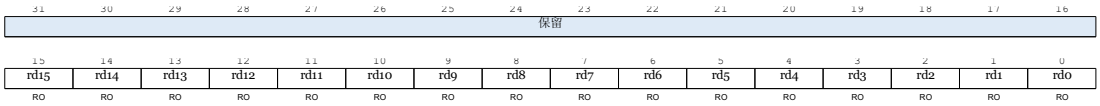
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	rd15	通道 15 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
14	rd14	通道 14 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
13	rd13	通道 13 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
12	rd12	通道 12 输入读取，只读， 0：当前输入为低电平；

位/位域	名称	描述
		1：当前输入为高电平。
11	rd11	通道 11 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
10	rd10	通道 10 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
9	rd9	通道 9 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
8	rd8	通道 8 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
7	rd7	通道 7 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
6	rd6	通道 6 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
5	rd5	通道 5 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
4	rd4	通道 4 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
3	rd3	通道 3 输入读取，只读，

位/位域	名称	描述
		0：当前输入为低电平； 1：当前输入为高电平。
2	rd2	通道 2 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
1	rd1	通道 1 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
0	rd0	通道 0 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。

19.6.12 GPIOx 中断类型选择寄存器（GPIOx_ITYPE）

地址偏移：0x2C

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

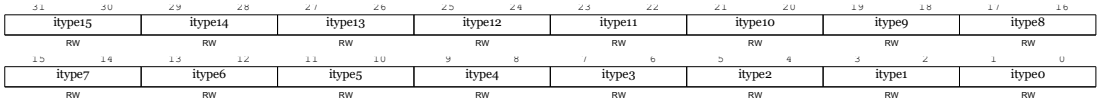
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:30	itype15	通道 15 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断；

位/位域	名称	描述
		2：下降沿触发中断； 3：上升沿或下降沿触发中断。
29:28	itype14	通道 14 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
27:26	itype13	通道 13 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
25:24	itype12	通道 12 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
23:22	itype11	通道 11 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
21:20	itype10	通道 10 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断；

位/位域	名称	描述
		3：上升沿或下降沿触发中断。
19:18	itype9	通道 9 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
17:16	itype8	通道 8 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
15:14	itype7	通道 7 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
13:12	itype6	通道 6 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
11:10	itype5	通道 5 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。

位/位域	名称	描述
9:8	itype4	通道 4 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
7:6	itype3	通道 3 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
5:4	itype2	通道 2 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
3:2	itype1	通道 1 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
1:0	itype0	通道 0 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。

19.6.13 GPIOx 中断原始数据寄存器（GPIOx_INT_RAW）

地址偏移：0x30

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

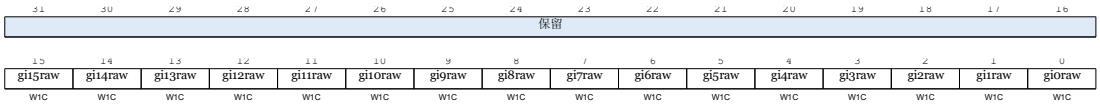
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	gi15raw	通道 15 中断原始位，触发 itype15 条件后置 1，写 1 清 0
14	gi14raw	通道 14 中断原始位，触发 itype14 条件后置 1，写 1 清 0
13	gi13raw	通道 13 中断原始位，触发 itype13 条件后置 1，写 1 清 0
12	gi12raw	通道 12 中断原始位，触发 itype12 条件后置 1，写 1 清 0
11	gi11raw	通道 11 中断原始位，触发 itype11 条件后置 1，写 1 清 0
10	gi10raw	通道 10 中断原始位，触发 itype10 条件后置 1，写 1 清 0
9	gi9raw	通道 9 中断原始位，触发 itype9 条件后置 1，写 1 清 0
8	gi8raw	通道 8 中断原始位，触发 itype8 条件后置 1，写 1 清 0
7	gi7raw	通道 7 中断原始位，触发 itype7 条件后置 1，写 1 清 0

位/位域	名称	描述
6	gi6raw	通道 6 中断原始位，触发 itype6 条件后置 1，写 1 清 0
5	gi5raw	通道 5 中断原始位，触发 itype5 条件后置 1，写 1 清 0
4	gi4raw	通道 4 中断原始位，触发 itype4 条件后置 1，写 1 清 0
3	gi3raw	通道 3 中断原始位，触发 itype3 条件后置 1，写 1 清 0
2	gi2raw	通道 2 中断原始位，触发 itype2 条件后置 1，写 1 清 0
1	gi1raw	通道 1 中断原始位，触发 itype1 条件后置 1，写 1 清 0
0	gi0raw	通道 0 中断原始位，触发 itype0 条件后置 1，写 1 清 0

19.6.14 GPIOx 中断强制赋值寄存器（GPIOx_INT_FORCE）

地址偏移：0x34

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

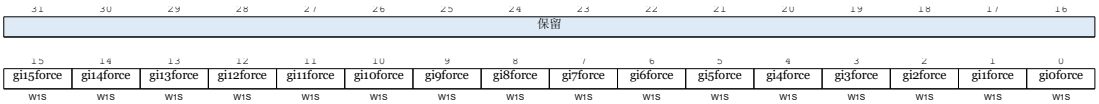
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	gi15force	通道 15 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
14	gi14force	通道 14 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
13	gi13force	通道 13 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1

位/位域	名称	描述
12	gi12force	通道 12 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
11	gi11force	通道 11 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
10	gi10force	通道 10 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
9	gi9force	通道 9 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
8	gi8force	通道 8 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
7	gi7force	通道 7 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
6	gi6force	通道 6 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
5	gi5force	通道 5 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
4	gi4force	通道 4 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
3	gi3force	通道 3 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
2	gi2force	通道 2 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
1	gi1force	通道 1 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1
0	gi0force	通道 0 中断强制赋值位，用于测试通道连通性，写 1 将 gi15raw 赋 1

19.6.15 GPIOx 中断遮罩寄存器（GPIOx_INT_MASK）

地址偏移：0x38

GPIOA 复位值：0x0000ffff

GPIOB 复位值：0x0000ffff

GPIOC 复位值：0x0000ffff

GPIOD 复位值：0x0000ffff

GPIOE 复位值：0x0000ffff

GPIOF 复位值：0x0000ffff

GPIOG 复位值：0x0000ffff

位/位域	名称	描述
31:16	保留	必须保持复位值
15	gi15mask	通道 15 中断遮罩位 1:中断关闭 0:中断使能
14	gi14mask	通道 14 中断遮罩位 1:中断关闭 0:中断使能
13	gi13mask	通道 13 中断遮罩位 1:中断关闭 0:中断使能
12	gi12mask	通道 12 中断遮罩位 1:中断关闭 0:中断使能
11	gi11mask	通道 11 中断遮罩位 1:中断关闭

位/位域	名称	描述
		0:中断使能
10	gi10mask	通道 10 中断遮罩位 1:中断关闭 0:中断使能
9	gi9mask	通道 9 中断遮罩位 1:中断关闭 0:中断使能
8	gi8mask	通道 8 中断遮罩位 1:中断关闭 0:中断使能
7	gi7mask	通道 7 中断遮罩位 1:中断关闭 0:中断使能
6	gi6mask	通道 6 中断遮罩位 1:中断关闭 0:中断使能
5	gi5mask	通道 5 中断遮罩位 1:中断关闭 0:中断使能
4	gi4mask	通道 4 中断遮罩位 1:中断关闭 0:中断使能
3	gi3mask	通道 3 中断遮罩位 1:中断关闭 0:中断使能
2	gi2mask	通道 2 中断遮罩位

位/位域	名称	描述
		1:中断关闭 0:中断使能
1	gi1mask	通道 1 中断遮罩位 1:中断关闭 0:中断使能
0	gi0mask	通道 0 中断遮罩位 1:中断关闭 0:中断使能

19.6.16 GPIOx 中断状态寄存器（GPIOx_INT_STATUS）

地址偏移：0x3C

GPIOA 复位值：0x00000000

GPIOB 复位值：0x00000000

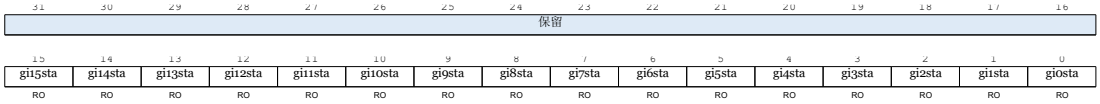
GPIOC 复位值：0x00000000

GPIOD 复位值：0x00000000

GPIOE 复位值：0x00000000

GPIOF 复位值：0x00000000

GPIOG 复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	gi15sta	通道 15 中断状态位 1:中断触发 0:中断未触发
14	gi14sta	通道 14 中断状态位 1:中断触发

位/位域	名称	描述
		0:中断未触发
13	gi13sta	通道 13 中断状态位 1:中断触发 0:中断未触发
12	gi12sta	通道 12 中断状态位 1:中断触发 0:中断未触发
11	gi11sta	通道 11 中断状态位 1:中断触发 0:中断未触发
10	gi10sta	通道 10 中断状态位 1:中断触发 0:中断未触发
9	gi9sta	通道 9 中断状态位 1:中断触发 0:中断未触发
8	gi8sta	通道 8 中断状态位 1:中断触发 0:中断未触发
7	gi7sta	通道 7 中断状态位 1:中断触发 0:中断未触发
6	gi6sta	通道 6 中断状态位 1:中断触发 0:中断未触发
5	gi5sta	通道 5 中断状态位

位/位域	名称	描述
		1:中断触发 0:中断未触发
4	gi4sta	通道 4 中断状态位 1:中断触发 0:中断未触发
3	gi3sta	通道 3 中断状态位 1:中断触发 0:中断未触发
2	gi2sta	通道 2 中断状态位 1:中断触发 0:中断未触发
1	gi1sta	通道 1 中断状态位 1:中断触发 0:中断未触发
0	gi0sta	通道 0 中断状态位 1:中断触发 0:中断未触发

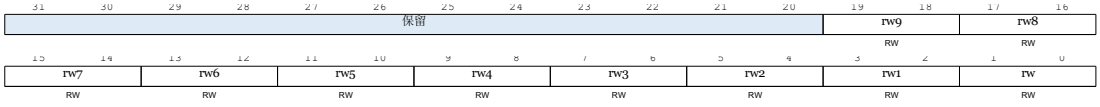
19.7 寄存器（GPIOH）

基地址：0x30102000

19.7.1 GPIOH 读写模式控制寄存器（GPIOH_RW）

地址偏移：0x0

复位值：0x00000000



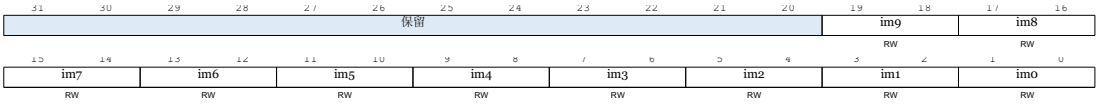
位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	rw9	通道 9 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
17:16	rw8	通道 8 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
15:14	rw7	通道 7 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
13:12	rw6	通道 6 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
11:10	rw5	通道 5 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。

位/位域	名称	描述
9:8	rw4	通道 4 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
7:6	rw3	通道 3 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
5:4	rw2	通道 2 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
3:2	rw1	通道 1 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。
1:0	rw0	通道 0 读写模式控制，读写， 0：数字输入模式，此时 IM 有效； 1：数字输出模式，此时 OM 有效； 2：模拟模式（即什么都不开模式）； 3：IDDQ 模式（一种测试模式）。

19.7.2 GPIOH 输入模式控制寄存器（GPIOH_IM）

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	im9	通道 9 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
17:16	im8	通道 8 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
15:14	im7	通道 7 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
13:12	im6	通道 6 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。

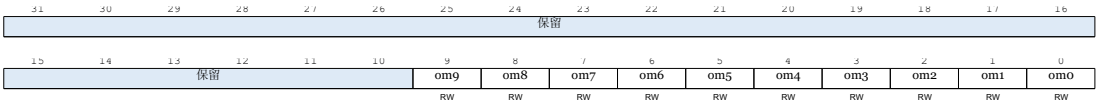
位/位域	名称	描述
11:10	im5	通道 5 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
9:8	im4	通道 4 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
7:6	im3	通道 3 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
5:4	im2	通道 2 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
3:2	im1	通道 1 输入模式控制，读写， 0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。
1:0	im0	通道 0 输入模式控制，读写，

位/位域	名称	描述
		0：浮空输入模式； 1：上拉输入模式； 2：下拉输入模式； 3：KEEP 模式（上下拉全拉）。

19.7.3 GPIOH 输出模式控制寄存器（GPIOH_OM）

地址偏移：0x8

复位值：0x00000000



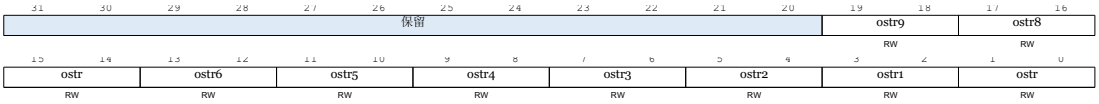
位/位域	名称	描述
31:10	保留	必须保持复位值
9	om9	通道 9 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
8	om8	通道 8 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
7	om7	通道 7 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
6	om6	通道 6 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）；

位/位域	名称	描述
		1：推挽输出（低电平高电平都输出）。
5	om5	通道 5 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
4	om4	通道 4 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
3	om3	通道 3 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
2	om2	通道 2 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
1	om1	通道 1 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。
0	om0	通道 0 输出模式控制，读写， 0：开漏输出（输出高电平时不拉读使能以实现高阻态输出）； 1：推挽输出（低电平高电平都输出）。

19.7.4 GPIOH 输出驱动强度控制寄存器（GPIOH_OSTR0）

地址偏移：0xC

复位值：0x00000000



位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	ostr9	通道 9 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
17:16	ostr8	通道 8 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
15:14	ostr	通道 7 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；

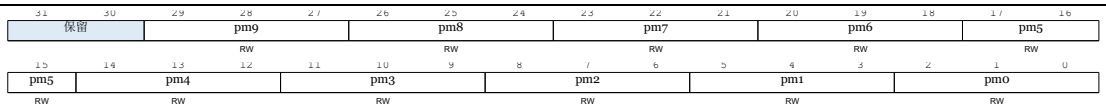
位/位域	名称	描述
13:12	ostr6	通道 6 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
11:10	ostr5	通道 5 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
9:8	ostr4	通道 4 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA； 3：输出最大驱动电流 18mA；
7:6	ostr3	通道 3 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA； 1：输出最大驱动电流 9mA； 2：输出最大驱动电流 13.5mA；

位/位域	名称	描述
		3：输出最大驱动电流 18mA；
5:4	ostr2	通道 2 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
3:2	ostr1	通道 1 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；
1:0	ostr	通道 0 输出强度控制，GPIO 数字输出模式/复用时有效，读写， 0：输出最大驱动电流 4.5mA ； 1：输出最大驱动电流 9mA ； 2：输出最大驱动电流 13.5mA ； 3：输出最大驱动电流 18mA；

19.7.5 GPIOH 复用接口选择寄存器（GPIOH_PM）

地址偏移：0x10

复位值：0x00000009



位/位域	名称	描述
31:30	保留	必须保持复位值
29:27	pm9	通道 9 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
26:24	pm8	通道 8 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
23:21	pm7	通道 7 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
20:18	pm6	通道 6 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
17:15	pm5	通道 5 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）；

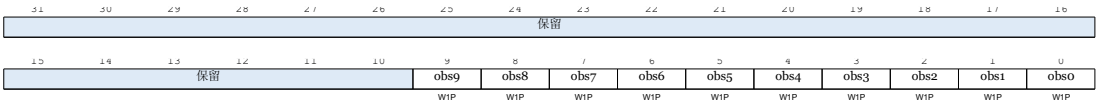
位/位域	名称	描述
		注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
14:12	pm4	通道 4 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
11:9	pm3	通道 3 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
8:6	pm2	通道 2 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
5:3	pm1	通道 1 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）； 注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。
2:0	pm0	通道 0 复用接口选择，只写， 0：GPIO 输出； 1~7：GPIO 使用复用功能 0~6（默认）；

位/位域	名称	描述
		注：在设为复用时，请将接口 GPIO 模式调整为浮空输入或推挽输出。

19.7.6 GPIOH 输出置位寄存器（GPIOH_OBS）

地址偏移：0x14

复位值：0x00000000



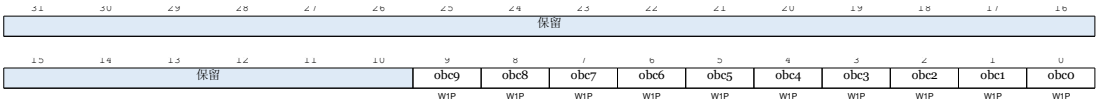
位/位域	名称	描述
31:10	保留	必须保持复位值
9	obs9	通道 9 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
8	obs8	通道 8 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
7	obs7	通道 7 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
6	obs6	通道 6 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
5	obs5	通道 5 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
4	obs4	通道 4 输出置位，只写， 0：保持当前状态；

位/位域	名称	描述
		1：将 GPIO 的输出设置为高电平。
3	obs3	通道 3 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
2	obs2	通道 2 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
1	obs1	通道 1 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。
0	obs0	通道 0 输出置位，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为高电平。

19.7.7 GPIOH 输出清除寄存器（GPIOH_OBC）

地址偏移：0x18

复位值：0x00000000



位/位域	名称	描述
31:10	保留	必须保持复位值
9	obc9	通道 9 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
8	obc8	通道 8 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。

位/位域	名称	描述
7	obc7	通道 7 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
6	obc6	通道 6 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
5	obc5	通道 5 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
4	obc4	通道 4 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
3	obc3	通道 3 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
2	obc2	通道 2 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
1	obc1	通道 1 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。
0	obc0	通道 0 输出清除，只写， 0：保持当前状态； 1：将 GPIO 的输出设置为低电平。

19.7.8 GPIO 输出翻转寄存器（GPIOH_OBT）

地址偏移：0x1C

复位值：0x00000000

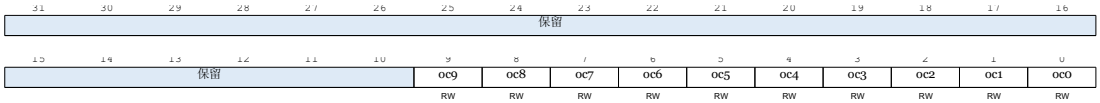
位/位域	名称	描述
31:10	保留	必须保持复位值
9	obt9	通道 9 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
8	obt8	通道 8 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
7	obt7	通道 7 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
6	obt6	通道 6 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
5	obt5	通道 5 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
4	obt4	通道 4 输出翻转，只写， 0：保持当前状态，

位/位域	名称	描述
		1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
3	obt3	通道 3 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
2	obt2	通道 2 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
1	obt1	通道 1 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。
0	obt0	通道 0 输出翻转，只写， 0：保持当前状态， 1：将 GPIO 的输出电平翻转，当前为高电平则输出低电平，当前为低电平则输出高电平。

19.7.9 GPIOH 输出控制寄存器（GPIOH_OC）

地址偏移：0x20

复位值：0x00000000



位/位域	名称	描述
31:10	保留	必须保持复位值
9	oc9	通道 9 输出控制，读写，

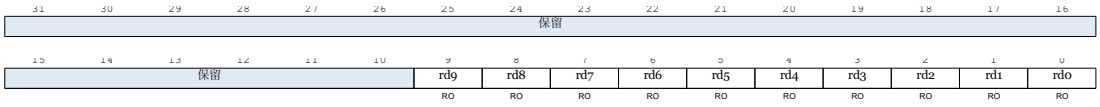
位/位域	名称	描述
		0：当前输出设置为低电平； 1：当前输出设置为高电平。
8	oc8	通道 8 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
7	oc7	通道 7 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
6	oc6	通道 6 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
5	oc5	通道 5 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
4	oc4	通道 4 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
3	oc3	通道 3 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
2	oc2	通道 2 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。
1	oc1	通道 1 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。

位/位域	名称	描述
0	oc0	通道 0 输出控制，读写， 0：当前输出设置为低电平； 1：当前输出设置为高电平。

19.7.10 GPIOH 输入读取寄存器（GPIOH_RD）

地址偏移：0x24

复位值：0x00000000



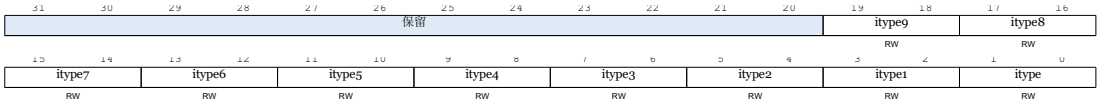
位/位域	名称	描述
31:10	保留	必须保持复位值
9	rd9	通道 9 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
8	rd8	通道 8 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
7	rd7	通道 7 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
6	rd6	通道 6 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
5	rd5	通道 5 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
4	rd4	通道 4 输入读取，只读，

位/位域	名称	描述
		0：当前输入为低电平； 1：当前输入为高电平。
3	rd3	通道 3 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
2	rd2	通道 2 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
1	rd1	通道 1 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。
0	rd0	通道 0 输入读取，只读， 0：当前输入为低电平； 1：当前输入为高电平。

19.7.11 GPIOH 中断类型选择寄存器（GPIOH_ITYPE）

地址偏移：0x28

复位值：0x00000000



位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	itype9	通道 9 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。

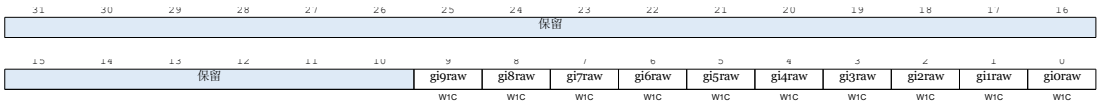
位/位域	名称	描述
17:16	itype8	通道 8 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
15:14	itype7	通道 7 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
13:12	itype6	通道 6 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
11:10	itype5	通道 5 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
9:8	itype4	通道 4 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
7:6	itype3	通道 3 中断类型选择，读写，

位/位域	名称	描述
		0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
5:4	itype2	通道 2 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
3:2	itype1	通道 1 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。
1:0	itype	通道 0 中断类型选择，读写， 0：无中断触发； 1：上升沿触发中断； 2：下降沿触发中断； 3：上升沿或下降沿触发中断。

19.7.12 GPIOH 中断原始数据寄存器（GPIOH_INT_RAW）

地址偏移：0x2C

复位值：0x00000000



位/位域	名称	描述
31:10	保留	必须保持复位值

位/位域	名称	描述
9	gi9raw	通道 9 中断原始位，触发 itype0 条件后置 1，写 1 清 0
8	gi8raw	通道 8 中断原始位，触发 itype0 条件后置 1，写 1 清 0
7	gi7raw	通道 7 中断原始位，触发 itype0 条件后置 1，写 1 清 0
6	gi6raw	通道 6 中断原始位，触发 itype0 条件后置 1，写 1 清 0
5	gi5raw	通道 5 中断原始位，触发 itype0 条件后置 1，写 1 清 0
4	gi4raw	通道 4 中断原始位，触发 itype0 条件后置 1，写 1 清 0
3	gi3raw	通道 3 中断原始位，触发 itype0 条件后置 1，写 1 清 0
2	gi2raw	通道 2 中断原始位，触发 itype0 条件后置 1，写 1 清 0
1	gi1raw	通道 1 中断原始位，触发 itype0 条件后置 1，写 1 清 0
0	gi0raw	通道 0 中断原始位，触发 itype0 条件后置 1，写 1 清 0

19.7.13 GPIOH 中断强制赋值寄存器（GPIOH_INT_FORCE）

地址偏移：0x30

复位值：0x00000000

位/位域	名称	描述
31:10	保留	必须保持复位值
9	gi9force	通道 9 中断强制赋值位，用于测试通道连通性，写 1 将 gi9raw 赋 1
8	gi8force	通道 8 中断强制赋值位，用于测试通道连通性，写 1 将 gi8raw 赋 1
7	gi7force	通道 7 中断强制赋值位，用于测试通道连通性，写 1 将 gi7raw 赋 1
6	gi6force	通道 6 中断强制赋值位，用于测试通道连通性，写 1 将 gi6raw 赋 1

位/位域	名称	描述
5	gi5force	通道 5 中断强制赋值位，用于测试通道连通性，写 1 将 gi5raw 赋 1
4	gi4force	通道 4 中断强制赋值位，用于测试通道连通性，写 1 将 gi4raw 赋 1
3	gi3force	通道 3 中断强制赋值位，用于测试通道连通性，写 1 将 gi3raw 赋 1
2	gi2force	通道 2 中断强制赋值位，用于测试通道连通性，写 1 将 gi2raw 赋 1
1	gi1force	通道 1 中断强制赋值位，用于测试通道连通性，写 1 将 gi1raw 赋 1
0	gi0force	通道 0 中断强制赋值位，用于测试通道连通性，写 1 将 gi0raw 赋 1

19.7.14 GPIOH 中断遮罩寄存器（GPIOH_INT_MASK）

地址偏移：0x34

复位值：0x000003ff

位/位域	名称	描述
31:10	保留	必须保持复位值
9	gi9mask	通道 9 中断遮罩位 1:中断关闭 0:中断使能
8	gi8mask	通道 8 中断遮罩位 1:中断关闭 0:中断使能
7	gi7mask	通道 7 中断遮罩位

位/位域	名称	描述
		1:中断关闭 0:中断使能
6	gi6mask	通道 6 中断遮罩位 1:中断关闭 0:中断使能
5	gi5mask	通道 5 中断遮罩位 1:中断关闭 0:中断使能
4	gi4mask	通道 4 中断遮罩位 1:中断关闭 0:中断使能
3	gi3mask	通道 3 中断遮罩位 1:中断关闭 0:中断使能
2	gi2mask	通道 2 中断遮罩位 1:中断关闭 0:中断使能
1	gi1mask	通道 1 中断遮罩位 1:中断关闭 0:中断使能
0	gi0mask	通道 0 中断遮罩位 1:中断关闭 0:中断使能

19.7.15 GPIOH 中断状态寄存器（GPIOH_INT_STATUS）

地址偏移：0x38

复位值：0x00000000

位/位域	名称	描述
31:10	保留	必须保持复位值
9	gi9sta	通道 9 中断状态位 1:中断触发 0:中断未触发
8	gi8sta	通道 8 中断状态位 1:中断触发 0:中断未触发
7	gi7sta	通道 7 中断状态位 1:中断触发 0:中断未触发
6	gi6sta	通道 6 中断状态位 1:中断触发 0:中断未触发
5	gi5sta	通道 5 中断状态位 1:中断触发 0:中断未触发
4	gi4sta	通道 4 中断状态位 1:中断触发 0:中断未触发
3	gi3sta	通道 3 中断状态位 1:中断触发 0:中断未触发
2	gi2sta	通道 2 中断状态位 1:中断触发

位/位域	名称	描述
		0:中断未触发
1	gi1sta	通道 1 中断状态位 1:中断触发 0:中断未触发
0	gi0sta	通道 0 中断状态位 1:中断触发 0:中断未触发

20 同步/异步串行通信接口（USART）

20.1 简介

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器实现了多种波特率。 USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）和调制解调器操作 (CTS/RTS) 通过配置多个缓冲区使用 DMA（直接存储器访问）可实现高速数据通信

20.2 主要特征

- 全双工异步通信/单线半双工通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，从而在速度容差与时钟容差之间取得最佳平衡
- 波特率发生器系统
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- 数据字长度可编程（6 位、7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 通用可编程收发波特率
- 自动波特率检测
- 两个用于收发数据的内部 FIFO。每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志
- 用于同步通信的同步主/从模式和时钟输出/输入
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 调制解调器和 RS-485 收发器的硬件流控制

- 通信控制/错误检测标志
  - 奇偶校验控制
  - 发送奇偶校验位
- 检查接收的数据字节的奇偶性
- 具有标志的中断源

20.3 扩展特征

- LIN 主模式同步中断发送功能和 LIN 从模式中断检测功能

20.4 功能说明

20.4.1 波特率生成器

接收器和发送器的波特率均由 USART_BRR 寄存器中的值确定。无论是 16 倍过采样还是 8 倍过采样其波特率的计算方式统一为

$$\text{Tx/Rx Baud} = \frac{PCLK}{USASRTDIV}$$

20.4.2 发送功能

发送器可以发送 6 位、7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位（TE）置 1。发送移位寄存器中的数据在 TX 引脚输出。如果使能同步主模式。其相应的时钟脉冲在 SCLK 引脚输出。

20.4.2.1 字符发送

USART 发送的数据格式为起始位、数据位、奇偶校验位和停止位。下图位 8 位数据位和 1 位停止位，不包含奇偶校验位的格式

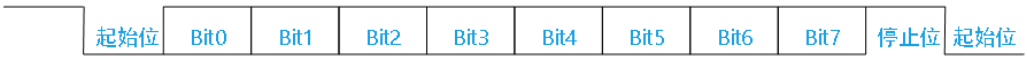


图 20.1 字符格式

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。在该模式下，USART_TDR 寄存器的缓冲区（TDR）位于内部总线和发送移位寄存器之间。使能 FIFO 模式时，写入到发送数据寄存器（USART_TDR）中的数据

会在 TXFIFO 中排队。每个字符前面都要一个起始位，其对应于一个位周期的逻辑低电平。

20.4.2.2 可配置停止位

可以通过 USART_CR2.STOP 位设置串口发送停止位的数量

- 1 个停止位：默认值
- 2 个停止位：推荐在 USART 正常模式、半双工模式和调制解调器模式中使用。

20.4.2.3 奇偶校验

USART 支持奇校验方式和偶校验方式。奇校验是指每帧数据中，包括数据位和奇偶校验位的全部数据位中 1 的个数必须为奇数；偶校验是指每帧数据中，包括数据位和奇偶校验位的全部数据位中 1 的个数必须为偶数

20.4.2.4 Break 字符

如果把 SBKRQ 位置 1，则 USART 会发送一个 break 字符。Break 字符的长度取决于 USART_CR2.BKNUM 位（Break 字符的长度）。通过写操作将 SBKF 位置 1 并在 break 字符发送完成时，该位由硬件复位（置 0）。USART 在 break 字符末尾的两位持续时间内插入一个逻辑“1”信号（STOP），以确保识别下一个帧的起始位

如果发送当前数据未完成就置位 USART_RQR.SBKRQ 位，则会等数据发送完成才发送 Break 字符。

在使能 FIFO 模式时，即使 TXFIFO 已满，发送 Break 字符的优先级也高于发送数据的优先级。

20.4.2.5 空闲字符

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧（起始位数据位奇偶校验位停止位均为 1）

20.4.2.6 单字节通信操作说明

1. 对 USART_CR1 中的 M 位进行编程以定义字长；
2. 使用 USART_BRR 寄存器选择所需波特率；
3. 对 USART_CR2 中停止位数量进行编程；

4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART；
5. 如果必须进行多缓冲区通信，选择 USART_CR3 中的 DMA 使能（DMAT）；
6. 将 USART_CR1 中的 TE 位置 1 以便首次发送时发送一个空闲帧；
7. 在 USART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤；
   - 禁止 FIFO 模式时，向 USART_TDR 写入数据会将 TXE 标志清零；
   - 使能 FIFO 模式时，向 USART_TDR 写入数据会为 TXFIFO 增添一个数据。当 TXFNF 标志置 1 时，会对 USART_TDR 执行写操作。该标志会保持置 1，直到 TXFIFO 已满；
8. 将最后一个数据写入 USART_TDR 寄存器后，等待 TC=“1”
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空

20.4.3 接收功能

USART 可以接收 6 位、7 位、8 位和 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

20.4.3.1 起始位检测

接收数据时可以采用 16 倍或 8 倍过采样。通过判断是否接收到特定序列来检测下降沿。检测下降沿的序列为 111_0XX，其中 XX 中必须有一个为 0。

后续判断起始位的方式也是通过特定序。在 16 倍过采样情况下，如果检测到序列 XXX_XXX_XXX（其中，每个“XXX”中必须包含 2 个“0”），则认为是起始位。如果检测到“XXX”中检测到 2 个 1，则认为接收到错误帧。在 8 倍过采样情况下，如果检测序列 XXX（其中，“XXX”中必须包含 2 个 0，则认为是检测到了起始位。同样，如果“XXX”中检测到 2 个 1，则认为接收到错误帧。

如果检测出的序列不完成，起始位检测将终止，接收器将返回空闲状态，等待下一个下降沿。 通过指定序列确定是否为起始位，还可以通过过滤的方式确定是否为起始位

20.4.3.2 Break 字符

在使能 LIN 的模式下，如果接收的数据全为 0，且停止位也为 0，则认为接收的是 Break 字符。此时会拉起 LIN 中断检测标志。如果是能了 LIN 中断检测使能，则会产生中断信号。 在不使能 LIN 模式下，如果接收的数据全为 0，且停止位也为 0，则认为是帧错误。

20.4.3.3 空闲状态

如果在两个字符内没有检测到下降沿，则认为线路空闲。如果是能 USART_CR1.IDLEIE 位，则会产生空闲中断。

20.4.3.4 噪声检测

在接收整个串口帧的过程中，如果三个连续采样点“XXX”出现非“111”和“000”时，则视为出现一个噪声点。三个连续的采样点为第 4/5/6 个。整个串口帧中如果出现三个噪声点，则会产生一个噪声中断。

Warning

在自动检测波特率时，可能会出现噪声错误。用户可以忽略此错误。

20.4.3.5 帧错误检测

在接收停止位时如果停止位为低，则认为是出现了帧错误（LIN 模式除外）。

20.4.3.6 奇偶校验错误检测

在使能奇偶校验模式下，如果线路上的奇偶校验位的值和计算的奇偶校验的值不同时会产生奇偶校验错误。如果是能了 USART_CR3.EIE 位，则产生奇偶校验错误中断。

20.4.3.7 上溢错误

- 禁止 FIFO 模式
- 如果在 RXNE 未清零时接收到字符，则会发生上溢错误。每接收到一个字节后，RXNE 标志都将置 1。当 RXNE 标志位是 1 时，如果再接收到一个数据，则会发生上溢错误。此时 USART_ISR.ORE 位会置 1。如果如果 USART_CR1.RXNEIF/RXFNEIE 为 1 或 USART_CR3.EIE 为 1 时，则会产生中断。发生上溢错误时，旧的数据会被新的数据覆盖。
- 使能 FIFO 模式

- 如果在 RXFNE 为高时接收到字符，则会发送上溢错误。此时 USART_ISR.ORE 位会置 1。如果如果 USART_CR1.RXNEIF/RXFNEIE 为 1 或 USART_CR3.EIE 为 1 时，则会产生中断。与禁止 FIFO 模式不同的是，使能 FIFO 模式时，发生上溢错误后接收的数据会丢失。

20.4.4 FIFO 模式

USART 可以工作在 FIFO 模式下。USART 具有一个发送 FIFO（TXFIFO）和一个接收 FIFO（RXFIFO）。可以通过 USART_CR1 寄存器中的 FIFOEN 置 1 使能 FIFO 模式。仅 UART、SPI 模式下支持该模式。

可以在配置触发 Tx 和 Rx 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR3 控制寄存器进行编程。在这种情况下：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 位域中编程的阈值时，会生成 Rx 中断。此时，USART_ISR 寄存器中的 RXFT 标志置 1。这表示已接收到 RXFTCFG 个数据。USART_RDR 中有 1 个数据，RXFIFO 中有（RXFTCFG-1）个数据。例如，如果将 RXFTCFG 编程为“100”，即 RXFIFO 的阈值为其深度的 7/8。 当接收深度的 7/8 个数据后，此时 RXFT 标志置 1。
- 当 TXFIFO 中的空位置数达到在 TXFTCFG 位域中编程的阈值时，会生成 Tx 中断

AS32A601 低功耗版本中 FIFO 的深度为 64

20.4.5 极性控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）

20.4.5.1 奇/偶校验

对于奇偶校验位进行计算，使数据位和奇偶校验位中“1”的数量为偶数。对于偶校验位进行计算是，使数据位和奇偶校验位中“1”的数量为奇数。例如，如果数据位为“01011010”，则其中“1”的数量为 4。当选择偶校验时，其奇偶校验位为“0”。在 TX 上发送的完整串口帧为“0_01011010_0_1”（分别为起始位、数据位、

奇偶校验位和停止位）。当选择奇校验时，其奇偶校验位为“1”。在 TX 上发送的完成串口帧为“0_01011010_1_1”。

20.4.5.2 接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会发生中断。通过软件将 USART_ICR 寄存器中 PECF 位写 1，来实现清零 PE 标志。

20.4.5.3 发送时进行奇偶校验位生成

如果选择偶校验（PS 为 0），则“1”的数量为偶数；如果选择奇校验（PS 为 1），则“1”的数量为奇数。发送模块会根据 PS 的值进行奇偶校验的填充。

表 20.1 数据位数表

M 位	PCE 位	USART 帧
00	0	-SB-8 位数据-STB
00	1	-SB-8 位数据-PB-STB
01	0	-SB-9 位数据-STB
01	1	-SB-9 位数据-PB-STB
10	0	-SB-7 位数据-STB
10	1	-SB-7 位数据-PB-STB
11	0	-SB-6 位数据-STB
11	1	-SB-6 位数据-PB-STB

表注： SB：起始位， STB：停止位， PB：奇偶校验位。

20.4.6 同步功能

20.4.6.1 主模式

要实现同步主模式，必须要设置如下寄存器

- USART_CR2 寄存器的 SLVEN 位为 0；
- USART_CR2 寄存器的 CLKEN 位为 1；
- USART_CR2 寄存器的 LINEN 位为 0；

- USART_CR3 寄存器的 HDSEL 位为 0； 在此模式下，USART 可用于在主模式下控制双向同步串口通信。SCLK 引脚是 USART 发送器时钟输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，如果 USART_CR2 寄存器中 LBCL 位为 1，则会发送生成时钟脉冲。如果 LBCL 位为 0，则不会发送生成时钟脉冲。

Warning

如果 LBCL 寄存器设置为 0，用户需要把发送和接收的字长设置为 7 位

在同步主模式下，USART 发送的工作方式与异步模式下完全相同。但是，由于 SCLK 与 TX 同步，因此 TX 上的数据是同步的。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置 1，则数据在 SCLK 边沿采样（根据 CPHA 和 CPOL 的值），而不进行任何过采样。此时必须保证给定建立时间和保持时间。

在主模式下，SCLK 引脚与 TX 引脚结合使用。因此，仅当使能发送器且正在发送数据时，才会提供时钟。这就意味着，没有发送数据时无法接收同步数据。发送数据时也可以复用 NSS 实现输出偏选使能。低有效。

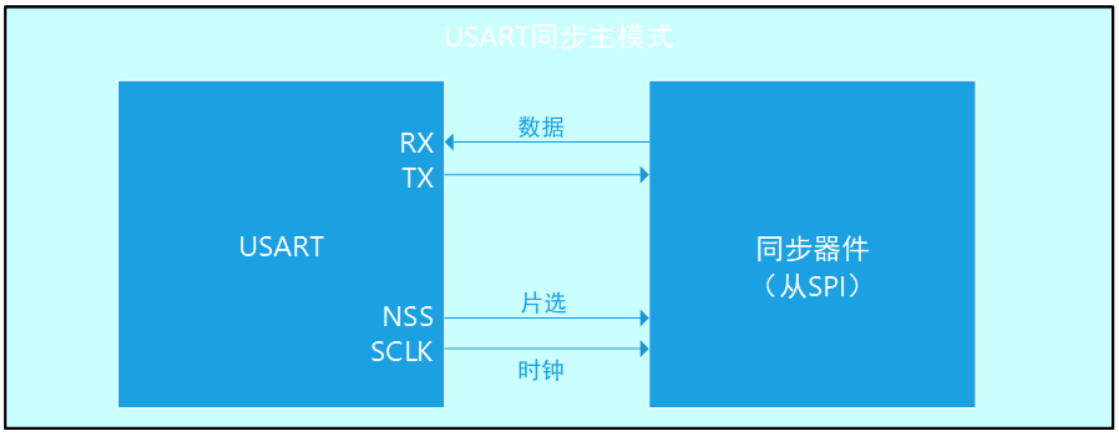


图 20.2 USART 同步主模式工作示意图

USART 在主同步模式下会根据 CPOL 和 CPHA 的值确定发送和接收数的时序。

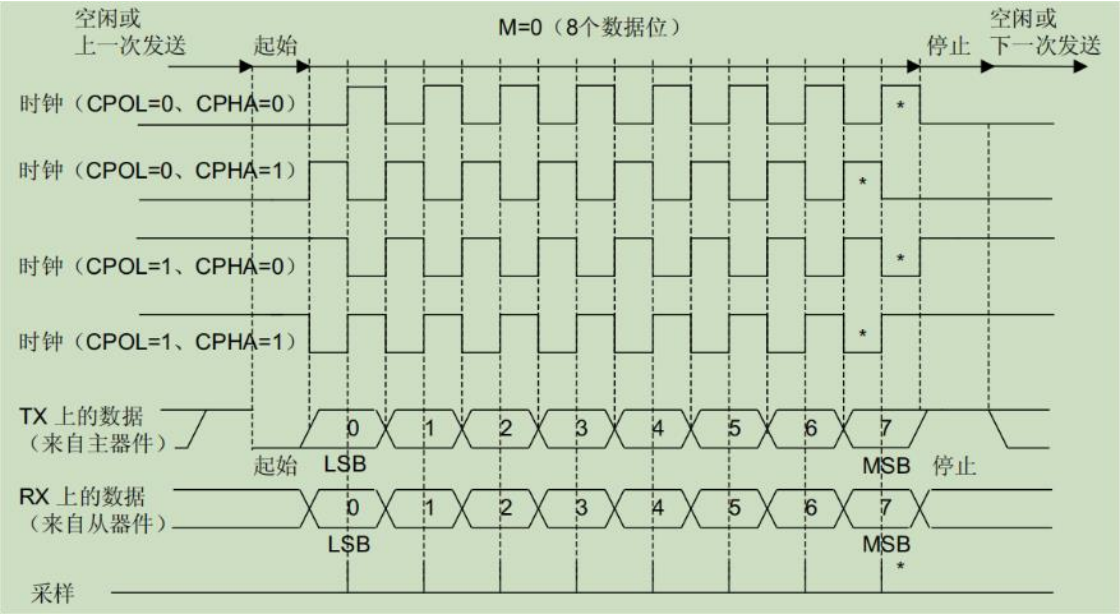


图 20.3 USART 同步主模式下的数据和时钟相位关系

20.4.6.2 从模式

要实现同步从模式，必须要设置如下寄存器

- USART_CR2 寄存器的 SLVEN 位为 1；
- USART_CR2 寄存器的 CLKEN 位为 1；
- USART_CR2 寄存器的 LINEN 位为 0；
- USART_CR3 寄存器的 HDSEL 位为 0；

在此模式下，USART 可用于在从模式下的双向同步串口通信。在从模式下，SCLK 引脚是 USART 的输入。从模式下引脚的连接关系如下图所示：

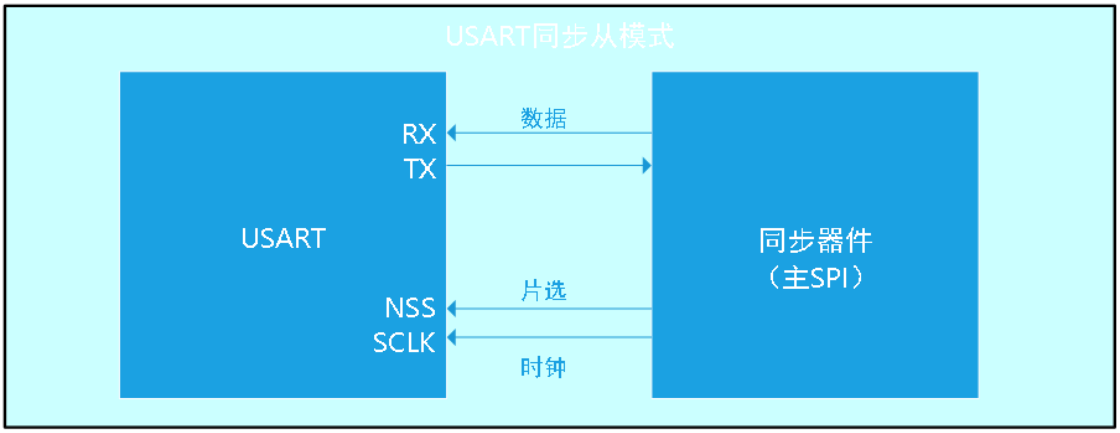


图 20.4 USART 同步从模式工作示意图

作为同步从模式时器件选择（NSS）引脚为输入模式。可通过 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从器件选择管理。

- 软件 NSS 管理（DIS_NSS 为 1）。将时钟选择 SPI 从器件并忽略 NSS 输入引脚
- 硬件 NSS 管理（DIS_NSS 为 0）。SPI 从器件选择取决于 NSS 输入引脚。NSS 为低电平时选择从器件，NSS 为高电平是取消选择从器件。

20.4.7 单线半双工

要实现单线半双工模式，必须要设置如下寄存器

- USART_CR2 寄存器的 SLVEN 位为 0；
- USART_CR2 寄存器的 CLKEN 位为 0；
- USART_CR2 寄存器的 LINEN 位为 0；
- USART_CR3 寄存器的 HDSEL 位为 1； USART 可以配置位遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 USART_CR3 寄存器中的控制位 HDSEL 进行半双工通信和全双工通信的选择。HDSEL 位置 1 后：
- TX 和 RX 线路从内部相连接
- 不能再使用 RX 引脚 无数据传输时，TX 引脚时钟处于释放状态。因此，它在空闲状态或者接收过程中用作标准 I/O。这就意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻

20.4.8 LIN 模式

要实现 LIN 模式，必须要设置如下寄存器

- USART_CR2 寄存器的 SLVEN 位为 0；
- USART_CR2 寄存器的 CLKEN 位为 0；
- USART_CR2 寄存器的 LINEN 位为 1；

- USART_CR3 寄存器的 HDSEL 位为 0； 在 LIN 模式下，建议使能 FIFO 模式、M 位配置字长为 8，且不是能奇偶校验。如果不失能 FIFO，则需要用户及时处理数据发送和数据接收。

20.4.8.1 LIN 主模式

在 LIN 主模式下的写数据工作流程为：

1. 配置 USART 为 LIN 模式、使能 FIFO 以及相应的设置。
2. 配置 USART_RQR.SBKRQ 位，以发送 Break 字符
3. 向 USART_TXR 寄存器写同步头 55、PID、数据和校验和等

在 LIN 主模式下的读数据流程为：

1. 配置 USART 为 LIN 模式、使能 FIFO 以及相应的设置；
2. 配置 USART_RQR.SBKRQ 位，以发送 Break 字符；
3. 向 USART_TXR 寄存器写同步头 55、PID；
4. 读取 USART_RDR 中接收的数据。

20.4.8.2 LIN 从模式

在 LIN 从模式下的写数据工作流程为：

1. 配置 USART 为 LIN 模式、使能 FIFO 以及相应的设置。
2. 检测 Break 中断，然后接收 1 个 BYTE,并判断其值是否为 55，如果是，则进入接收 PID 流程；
3. 接收 PID 值，并进行校验；
4. 向 USART_TXR 寄存器数据和校验和等。

20.5 中断

在 USART 通过过程中会产生不同的中断。每个中断源和中断处理方式有所不同，具体中断如下表所示：

表 20.2 USART 中断源

中断事件	事件标志	使能控制位	中断清除方法
发送完成	TC	TCIE	TCCF 位置 1 清零或写 TDR 寄存器清零

中断事件	事件标志	使能控制位	中断清除方法
线路空闲	LINE_IDLE	IDLEIE	IDLECF 位置 1 清零
接收寄存器不为空	RXNE	RXNEIE	读取 RDR 寄存器清零或写 RXFRQ 位置 1 清零
接收 FIFO 非空	RXFNE	RXFNEIE	读取 RDR 寄存器清零或写 RXFRQ 位置 1 清零
接收 FIFO 满	RXFF	RXFFIE	读取 RDR 寄存器清零或写 RXFRQ 位置 1 清零
发送 FIFO 空	TXFE	TXFEIE	写 TXFECF 位置 1 清零
发送 FIFO 阈值中断	TXFTH	TXFTHIE	写 TXFTCF 位置 1 清零
接收 FIFO 阈值中断	RXFFTH	RXFFTHIE	写 RXFTCF 位置 1 清零
发送寄存器空	TXE	TXEIE	写 TDR 寄存器清零或写 TXFRQ 位置 1 清零
发送 FIFO 非满 发送 FIFO 从满变成不满时会产生中断	TXFNF	TXFNFIE	写 TXFRQ 位置 1 清零
CTS 状态变化中断	CTS	CTSIE	写 CTSCF 位置 1 清零
LIN BREAK 字符检标志	LBDF	LBDIE	写 LBDCF 位置 1 清零

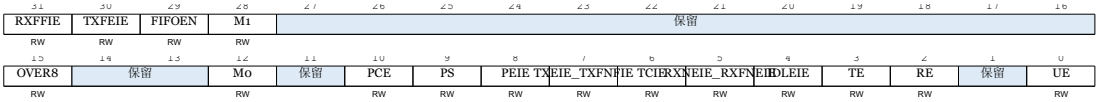
中断事件	事件标志	使能控制位	中断清除方法
奇偶校验错误	PE	PEIE	写 PECF 位置 1 清零
噪声错误	NE	EIE	写 NECF 位置 1 清零
帧错误	FE	EIE	写 FECF 位置 1 清零
溢出错误	ORE	EIE	写 ORECF 位置 1 清零

20.6 寄存器

20.6.1 USART 控制寄存器 1(USART_CR1)

地址偏移：0x0

复位值：0x00000000



位/位域	名称	描述
31	RXFFIE	RXFIFO 为满时发起中断使能 0:RXFIFO 为满时发起中断使能无效 1: RXFIFO 为满时发起中断使能有效
30	TXFEIE	TXFIFO 为空时发起中断使能 0:TXFIFO 为空时发起中断使能无效 1: TXFIFO 为空时发起中断使能有效
29	FIFOEN	FIFO 模式使能 0:FIFO 模式无效 1:FIFO 模式有效 只有在 UE 为 0 时才能写入此位

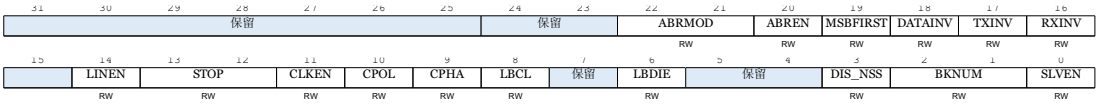
位/位域	名称	描述
28	M1	字长高位 M[1:0] = 00 : 8 个数据位 M[1:0] = 01 : 9 个数据位 M[1:0] = 10 : 7 个数据位 M[1:0] = 11 : 6 个数据位 只有在 UE 为 0 时才能写入此位
27:16	保留	必须保持复位值
15	OVER8	过采样模式 0: 16 被过采样模式 1: 8 被过采样模式 只有在 UE 为 0 时才能写入此位
14:13	保留	必须保持复位值
12	M0	字长低位 只有在 UE 为 0 时才能写入此位
11	保留	必须保持复位值
10	PCE	奇偶校验控制使能（Parity Control Enable） 0:不使能奇偶校验 1:使能奇偶校验 只有在 UE 为 0 时才能写入此位
9	PS	奇偶校验选择（Parity Select） 0:选择偶校验 1:选择奇校验 只有在 UE 为 0 时才能写入此位
8	PEIE	奇偶校验错误中断使能 0:奇偶校验错误中断使能无效 1:奇偶校验错误中断使能有效

位/位域	名称	描述
7	TXEIE_TXFNFIE	发送数据寄存器为空（TXE）/TXFIFO 未满（TXFE）中断使能 0:TXE/TXFE 中断使能无效 TXE/TXFE 中断使能有效
6	TCIE	传输完成中断使能 0:数据传输完成中断使能无效 1:数据传输完成中断使能有效
5	RXNEIE_RXFNEIE	接收数据寄存器非空（RXNE）/RXFIFO 非空（RXFNE）中断使能 0:不使能 RXNE/RXFNE 中断 1:使能 RXNE/RXFNE 中断
4	IDLEIE	接收空闲中断使能 0:不使能接收空闲中断 1:使能接收空闲中断
3	TE	USART 发送器使能 0:USART 发送器无效 1:USART 发送器有效
2	RE	USART 接收器使能 0:USART 接收器无效 1:USART 接收器有效
1	保留	必须保持复位值
0	UE	USART 使能 0:不使能 USART 1:使能 USART

20.6.2 USART 控制寄存器 2(USART_CR2)

地址偏移：0x4

复位值：0x00000100



位/位域	名称	描述
31:23	保留	必须保持复位值
22:21	ABRMOD	自动波特率检测模式 目前只支持检测 0x55。需要在 7bit 模式下完成
20	ABREN	自动波特率使能 0: 禁止自动波特率检测 1: 使能自动波特率检测
19	MSBFIRST	最高有效位在前。 0: 先发送 bit0，最后发送 bit6/7/8/9 1: 先发送 bit6/7/8/9，最后发送 bit0 只有在 UE 为 0 时才能写入此位
18	DATAINV	二进制数据反向（Binary Data Inversion） 0: 发送数据和接收数据不取反 1: 发送数据和接收数据取反 只有在 UE 为 0 时才能写入此位
17	TXINV	TX 引脚有效电平反向（TX Pin Active Level Inversion） 0: TX 引脚为高电压时，对应数据位逻辑 1 1: TX 引脚为高电压时，对应数据位逻辑 0 允许在 TX 线路上使用外部反相器 只有在 UE 为 0 时才能写入此位
16	RXINV	RX 引脚有效电平反向（RX Pin Active Level Inversion） 0: RX 引脚为高电压时，对应数据位逻辑 1 1: RX 引脚为高电压时，对应数据位逻辑 0

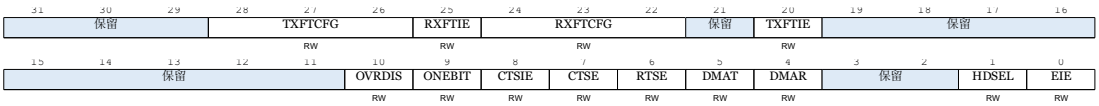
位/位域	名称	描述
		允许在 RX 线路上使用外部反相器 只有在 UE 为 0 时才能写入此位
15	保留	必须保持复位值
14	LINEN	LIN 模式使能 0: LIN 模式无效 1: LIN 模式有效 只有在 UE 为 0 时才能写入此位
13:12	STOP	停止位个数（STOP Bits） 00: 1 个停止位 10: 2 个停止位 只有在 UE 为 0 时才能写入此位
11	CLKEN	时钟使能（Clock Enable） 0: 禁止 SCLK 引脚 1: 使能 SCLK 引脚 注：如果使能时钟，则 USART 会进入同步主模式或同步从模式 只有在 UE 为 0 时才能写入此位
10	CPOL	时钟极性（Clock Polarity） 0: 空闲是 SCLK 引脚为低电平 1: 空闲是 SCLK 引脚为高电平
9	CPHA	时钟相位（Clock Phase） 0: 从第一个时钟边沿开始采样数据 1: 从第二个边沿开始采样数据
8	LBCL	最后一个位时钟脉冲（Last bit clock pulse）。在同步从模式下改为必须为 1 0: 最后一个数据为的时钟脉冲不在 SCLK 引脚上输出

位/位域	名称	描述
		1：最后一个数据位的时钟在 SCLK 引脚上输出
7	保留	必须保持复位值
6	LBDIE	LIN 中断检测中断使能 0: 禁止中断 1: 在 LIN 模式下，如果检测到 LIN 中断，则 USART_ISR.LBDF 为 1 时就会产生中断
5:4	保留	必须保持复位值
3	DIS_NSS	忽略 NSS 引脚输入 0: SPI 从器件选择信号取决于 NSS 输入引脚 1: 时钟选择 SPI 从器件，忽略 NSS 输入引脚
2:1	BKNUM	发送 Break 字符的长度 00: 发送 Break 的字符长度为 13 个数据位 01: 发送 Break 的字符长度为 14 个数据位 10: 发送 Break 的字符长度为 12 个数据位 11: 发送 Break 的字符长度为 11 个数据位
0	SLVEN	同步从模式使能 0:同步从模式无效 1:同步从模式有效 注：如果使能 USART_CR2.CLKEN,则会打开同步模式。主模式还是从模式取决于本位

20.6.3 USART 控制寄存器 3(USART_CR3)

地址偏移：0x8

复位值：0x00000000



位/位域	名称	描述
31:29	保留	必须保持复位值
28:26	TXFTCFG	接收 FIFO 阈值配置 000：TXFIFO 达到其深度的 1/8 001：TXFIFO 达到其深度的 1/4 010：TXFIFO 达到其深度的 1/2 011：TXFIFO 达到其深度的 3/4 100：TXFIFO 达到其深度的 7/8 101：TXFIFO 变空 其余组合：保留
25	RXFTIE	RXFIFO 阈值中断使能 0: 禁止中断 当 RXFIFO 达到 RXFTCFG 中编程的阈值时，会产生 USART 中断
24:22	RXFTCFG	接收 FIFO 阈值配置 000：RXFIFO 达到其深度的 1/8 001：RXFIFO 达到其深度的 1/4 010：RXFIFO 达到其深度的 1/2 011：RXFIFO 达到其深度的 3/4 100：RXFIFO 达到其深度的 7/8 101：RXFIFO 变满 其余组合：保留
21	保留	必须保持复位值
20	TXFTIE	TXFIFO 阈值中断使能 0: 禁止中断 当 TXFIFO 达到 TXFTCFG 中编程的阈值时，会产生 USART 中断

位/位域	名称	描述
19:11	保留	必须保持复位值
10	OVRDIS	上溢禁止 0: 接收新数据前未读取已经接收的数据时，会产生 ORE 1: 禁止上溢功能，如果在 RXNE 为 1 时，仍能接收新的数据
9	ONEBIT	一个采样位方法使能 0: 3 个采样位方法 1: 1 个采样位方法 当用户选择 1 个采样位方法时，将禁止噪声检测。
8	CTSIE	CTS 中断使能 0: 禁止中断 1: 当 USART_ISR.CTSIF 为 1 时会产生中断
7	CTSE	CTS 使能 0: 禁止 CTS 硬件流控 1: 使能 CTS 模式，仅当 nCTS 输入有效时才发送数据。如果在发送数据时使能 nCTS 无效。如果使能 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效
6	RTSE	RTS 使能，硬件流控 0: 禁止 RTS 硬件流控 1: 使能 RTS 输出，仅当接收缓存区中有空间时才会请求数据，发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉低为 0）
5	DMAT	DMA 发送使能 0: 禁止使用 DMA 发送数据

位/位域	名称	描述
		1: 使能 DMA 发送数据
4	DMAR	DMA 接收使能 0: 禁止使用 DMA 接收数据 1: 使能 DMA 接收数据
3:2	保留	必须保持复位值
1	HDSEL	半双工选择 0: 禁止半双工模式 1: 使能半双工模式 在半双工模式，只有 TX 引脚可用
0	EIE	错误中断使能（Error Interrupt Enable） 0: 禁止中断 1: USART_ISR.FE、USART_ISR.NE 为 1 时，记出现帧错误或噪声错误时，会产生中断

20.6.4 USART 波特率寄存器(USART_BRR)

地址偏移：0xC

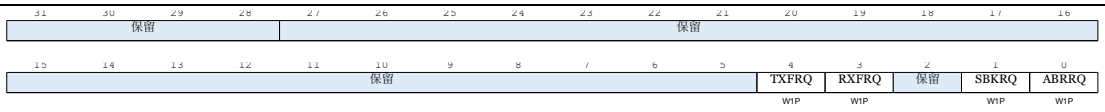
复位值：0x00000000

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BRR	USART 波特率寄存器 无论是 16 倍过采样还是 8 被过采样，其波特率的算方式一致

20.6.5 USART 请求寄存器(USART_RQR)

地址偏移：0x18

复位值：0x00000000



位/位域	名称	描述
31:28	保留	必须保持复位值
27:5	保留	必须保持复位值
4	TXFRQ	发送数据刷新请求（Transmit Data Flush Request） 禁止 FIFO 模式时：向此位写入 1，将 TXE 标志位置 1
3	RXFRQ	接收数据刷新请求（Receive Data Flush Request） 向此位写入 1，将清空整个接收 FIFO，RXFNE 位同时清零 此操作可能导致数据丢失
2	保留	必须保持复位值
1	SBKRQ	发送中断请求（Send Break Request） 向此位写入 1，可将 USART_ISR.SBKF 位置 1，并在发送单元可用后立刻发送 BREAK
0	ABRRQ	自动波特率请求（Auto Baud Rate Request） 向此位写 1，可以复位 USART_ISR.ABRF 位，并请求对下一个接收到的数据帧进行自动波特率测量

20.6.6 USART 中断状态寄存器(USART_ISR)

地址偏移：0x1C

复位值：0x00800090

位/位域	名称	描述
31:28	保留	必须保持复位值
27	ISR_TXFT	TXFIFO 阈值标志（TXFIFO Threshold Flag）

位/位域	名称	描述
		达到 USART_CR3.TXFTCFG 中编程的阈值时，该位由硬件置 1 如果 USART_CR3.TXFTIE 为 1，则产生中断 0:发送 FIFO 未达到编程阈值 1：发送 FIFO 达到编程阈值
26	ISR_RXFT	RXFIFO 阈值标志（RXFIFO Threshold Flag） 达到 USART_CR3.RXFTCFG 中编程的阈值时，该位由硬件置 1 如果 USART_CR3.RXFTIE 为 1，则产生中断 0:接收 FIFO 未达到编程阈值 1：接收 FIFO 达到编程阈值
25	保留	必须保持复位值
24	ISR_RXFF	RXFIFO 已满（RXFIFO Full） 当接收 FIFO 已满时，该位由硬件置 1 如果 USART_CR1.RXFFIE 位为 1，则产生中断 0:RXFIFO 未满 1：RXFIFO 已满
23	ISR_TXFE	TXFIFO 为空（TXFIFO Empty） 当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 USART_RQR.TXFRQ（写入“1”将 TXFE 标志置 1。 如果 USART_CR1.TXFEIE 为 1，则会生成中断 0:TXFIFO 非空 1:TXFIFO 为空
22:19	保留	必须保持复位值
18	ISR_SBKF	发送中断标志（Send Break Flag）

位/位域	名称	描述
		此位指示请求发送 Break 字符。此位由软件置 1，方法时向 USART_CR3.SBKRQ 位写入 1。此位在中断发送完成后自动复位 0:不发送 Break 字符 1:将发送 Break 字符
17	保留	必须保持复位值
16	ISR_BUSY	忙标志（Busy Flag） 此位由硬件置 1 和复位。当 RX 线路上正在进行通信时有效。检测到起始位时拉高，接收完成时拉低 0:USART 处于空闲状态 1:正在接收
15	ISR_ABRF	自动波特率标志（Auto Baud Rate Flag） 已设置自动比特率，或者自动波特率操作未完成时，此位由硬件置 1 可以想 USART_RQR.ABRRQ 写 1 清零此位
14	ISR_ABRE	自动波特率错误（Auto Baud Rate Error） 如果波特率测量失败（波特率超出范围或者字符比较失败），此位由硬件置 1 此位由软件清零，方法时向 USART_CR3.ABRRQ 位写 1
13:11	保留	必须保持复位值
10	ISR_CTS	CTS 标志（CTS Flag） 此位由硬件置 1,此位是对 nCTS 输入信教的状态取反 0:nCTS 线为 1 1:nCTS 线为 0
9	ISR_CTSIF	CTS 中断标志 （CTS Interrupt Flag）

位/位域	名称	描述
		如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。此位由软件清零，方法时向 USART_ICR.CTSCF 位写 1 如果 USART_CR3.CTSIE 为 1，会产生中断 0:nCTS 状态线上未发生变化 1:nCTS 状态线上发送变化
8	ISR_LBDF	LIN 中断检测标志（LIN Break Detection Flag） 检测到 LIN 中断时，该位由硬件置 1.此位由软件清零，方法时向 USART_ICR.LBDCF 写 1 如果 USART_CR2.LBDIE 为 1，会产生中断 0:未检测到 LIN 中断 1:检测到 LIN 中断
7	ISR_TXE	发送数据寄存器为空/TXFIFO 未满（Transmit Data Register Empty/ TXFIFO Not FUll 禁止 FIFO 模式：当 USART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1.通过对 USART_TDR 寄存器执行写操作将该位清零。也可以通过向 USART_RQR.TXFRQ 写入 1 将该为位置 1，并清除中断 使能 FIFO 模式：TXFNF 会在 TXFIFO 未满时由硬件置 1，表示可以向 USART_TDR 中写入数据。当 TXFIFO 已满时，该位清零，表示不能想 USART_TDR 写入数据 如果 USART_CR1.TXEIE/TXFNFIE 为 1 时，会产生中断 0:数据寄存器已满/发送 FIFO 已满

位/位域	名称	描述
		1:数据寄存器未满/发送 FIFO 未满
6	ISR_TC	发送完成（Transmission Complete） 该位这是写入 USART_TDR 中的最后一个数据已从移位寄存器中发出 如果已完成对包含数据的帧的发送并且 TXE/TXFE 置 1，则此位由硬件置 1 如果 USART_CR1.TCIE 为 1，则会产生中断 TC 位由软件清零，方法是向 USART_ICR.TCCF 写入 1 或向 USART_TDR 执行写操作 0:传输未完成 1:传输已完成
5	ISR_RXNE	读取数据寄存器非空/RXFIFO 非空 通过对 USART_RDR 寄存器执行读操作将该位清零。也可以通过向 USART_RQR.RXFRQ 写如 1 将 RXNE 清零 如果 USART_CR1.RXNEIE/RXFNEIE 为 1，则会生成中断 0：未接收到数据 1：已准好读取接收到的数据
4	ISR_IDLE	在接收完数据后一段时间如果没有检测到下降沿就认为是空闲状态 当检测到线路空闲是，该位由硬件置 1。如果 USART_CR1.IDLEIE 为 1，则会产生中断。此位由软件清零。方法时向 USART_ICR.IDLECF 写入 1 0：未检测到空闲线路 1：检测空闲线路

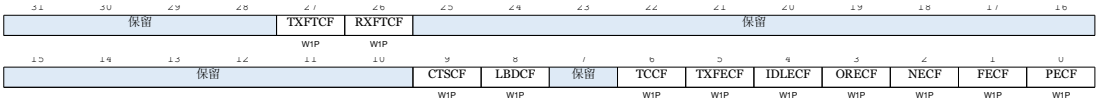
位/位域	名称	描述
3	ISR_ORE	溢出错误（Overrun Error） 当使能 FIFO 时：接收 FIFO 已满，且再次接收到数据，就会触发溢出错误 当禁止 FIFO 时：接收数据已缓存到 USART_RDR 寄存器，且再次接收到数据，也会触发溢出错误 软件清零的方式是向 USART_ICR.ORECF 写入 1 如果 USART_CR1.RXNEIF/RXFNEIE 为 1 或 USART_CR3.EIE 为 1 时，则会产生中断 0：无溢出错误 1：检测到溢出错误
2	ISR_NE	噪声检测标志（Noise Detection Flag） 当在接收的帧上检测到噪声时，该位由硬件置 1。由软件清零，方式是向 USART_ICR.NFCF 位写 1 0：未检测到噪声 1：检测到噪声 注：只有在异步串口模式下噪声错误才有效
1	ISR_FE	帧错误（Framing Error） 当检测到过度的噪声或 break 字符时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR.FECF 位写 1 如果 USART_CR.EIE 为 1，则会产生中断 0：未检测到帧错误 1：检测到帧错误或 Break 字符 注：只有在异步串口模式下帧错误才有效
0	ISR_PE	奇偶校验错误（Parity Error）

位/位域	名称	描述
		当在接收数据过程中发生奇偶校验错误时，该位由硬件置 1.此位由软件清零，方法是向 USART_ICR.PECF 写 1 如果 USART_CR1。PEIE 为 1，则会生成中断 0:无奇偶校验错误 1:奇偶校验错误 注：只有在异步串口模式下奇偶校验错误才有效

20.6.7 USART 中断清除寄存器(USART_ICR)

地址偏移：0x20

复位值：0x00000000



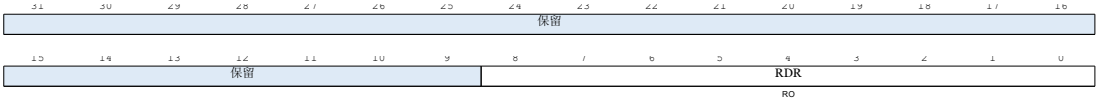
位/位域	名称	描述
31:28	保留	必须保持复位值
27	TXFTCF	发送 FIFO 阈值状态清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.TXFT 标志将清零
26	RXFTCF	接收 FIFO 阈值状态清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.RXFT 标志将清零
25:10	保留	必须保持复位值
9	CTSCF	CTS 清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.CTSIF 标志将清零
8	LBDCF	LIN 中断检测清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.LBDF 标志将清零
7	保留	必须保持复位值
6	TCCF	发送完成清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.TC 标志将清零

位/位域	名称	描述
5	TXFECF	TXFIFO 为空清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.TXFE 标志将清零
4	IDLECF	检测到空闲线路清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.IDLE 标志将清零
3	ORECF	上溢错误清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.ORE 标志将清零
2	NECF	检测到噪声清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.NE 标志将清零
1	FECF	帧错误清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.FE 标志将清零
0	PECF	奇偶校验错误清零标志。此位为写 1 产生脉冲位 向此位写入 1 时，USART_ISR.PE 标志将清零

20.6.8 USART 接收数据寄存器(USART_RDR)

地址偏移：0x24

复位值：0x00000000

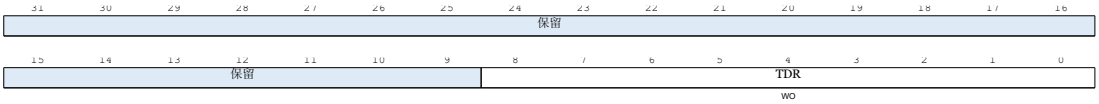


位/位域	名称	描述
31:9	保留	必须保持复位值
8:0	RDR	接收数据值

20.6.9 USART 发送数据寄存器(USART_TDR)

地址偏移：0x28

复位值：0x00000000



位/位域	名称	描述
31:9	保留	必须保持复位值
8:0	TDR	发送数据值

21 控制器局域网络（CAN）

21.1 简介

控制器局域网络（CAN）根据 ISO11898-1：2015 和 Bosch CAN FD 规范进行通信。连接到物理层需要额外的收发器硬件。

所有有关处理消息的函数都由接收处理程序和发送处理程序实现。接收处理程序管理消息接受过滤，将接收到的消息从 CAN 核心发送到消息 RAM，以及提供接收消息状态信息。发送处理程序负责将发送消息从消息 RAM 发送到 CAN 核心，并提供发送状态信息。

接收过滤通过最多 32 个过滤器元素的组合来实现，其中每个元素可以配置为范围、位掩码或专用 ID 过滤器。

21.2 主要特征

- 设计符合 ISO11898-1/2015 规范；
- 支持 CAN 和 CAN FD 帧；
- 支持 ISO11899：2015 规范中指定的 CAN FD 帧格式；
- 支持 64 字节 CAN FD 帧；
- 支持可变数据速率高达 8Mb/s；
- 支持正常数据速率 1Mb/s；
- 支持高达三个数据位的发射器延迟补偿；
- 支持可配置的发送和和接受邮箱缓冲区；
- 支持两个深度为 64 帧消息的缓冲与 32 个 ID 过滤掩码；
- 支持低 ID 的消息优先发送；
- 支持待发送消息消除；
- 支持快速数据速率时的单独错误记录。

21.3 功能说明

21.3.1 帧结构说明

表 21.1 帧结构表

名称	位宽	说明
ID	32	帧的 ID、帧类型等信息
DLC	32	帧长度信息、是否为 CAN FD 帧，是否加速等信息
DATA0	32	数据字段，高字节有效。 如果只发送一个字节则发送 DATA0[31:24]。 如果发送两个字节则先发送 DATA0[31:24]，再发送 DATA0[23:16]。以此类推
DATA1-15	32	同上

表 21.2 ID 的具体字段说明

位	名称	说明
31：21	ID[28:18]	标准帧 ID，11bit。应用于标准 CAN 和标准 CAN FD 帧格式
20	SRR/RTR/RRS	代替远程远程请求标志: 扩展 CAN 和扩展 CAN FD 帧时该位为 1 标准 CAN FD 帧时该位为 0 标准 CAN 帧时为 0 表示该帧为数据帧；为 1 表示该帧为远程帧
19	IDE	帧扩展标志位。1：表示扩展帧；0：表示标准帧
18:1	ID[17:0]	扩展帧 ID，29bit。应用于扩展 CAN 和扩展 CAN FD 帧格式
0	RTR/RRS	远程请求标志:

表 21.3 DLC 的具体字段说明

位	名称	说明
31:28	DLC[3:0]	数据长度编码
27	FDF/EDL	CAN FD 帧标志位。 1：表示 CAN FD 帧；0：表示 CAN 帧
26	BRS	CAN FD 帧的快速数据速率标志位。 1：切换速率；0：不切换速率
25:0	RSV	预留位

21.3.2 工作模式

CAN FD 工作有多种工作模式：休眠模式、回环模式、监听模式和正常模式。

- 休眠模式：使能 MSR 寄存器的 SLEEP 位，且无效 LBACK 和 SNOOP 位后，CAN 核会进入休眠模式。
- 回环模式：用户写 SRR 寄存器的 CEN 位为 0，然后使能 MSR 寄存器的 LBACK 位，然后再写 CEN 位为 1，即可进入回环模式。在此模式下，用户可以直接发送数据，同时接收端也会接收到刚刚发送的数据。回环模式下，不需要用户把收发短接。
- 监听模式：用户需要写 SRR 寄存器的 CEN 位为 0，然后使能 MSR 寄存器的 SNOOP 位，然后再写 CEN 位为 1，即可进入监听模式。在此模式下，发送端会发送隐性位到 CAN 总线。接收其他发送端发送的消息，但是不回复 ACK 信号。把符合接收过滤器规则的接收消息存储到 RAM 中。
- 在此模式下完成对数据的收发功能。

21.4 应用说明

21.4.1 初始化流程

在正常使用 CAN FD 前需要对其寄存器进行初始化。首先需要对 CAN FD 进行复位，然后再对其他寄存器进行操作。具体流程如下图所示：

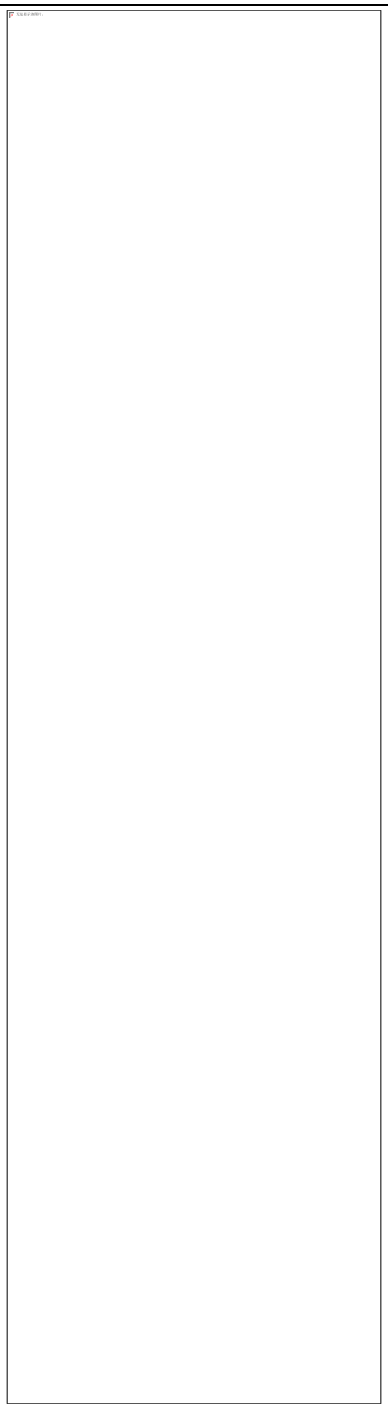


图 21.1 CAN 初始化流程示意图

在初始化时配置的 CAN 波特率寄存器和 CAN FD 波特率寄存器时 SRR 寄存器 CEN 的值必须为 0。在配置完波特率寄存器后，写入 SRR 寄存器 CEN 位为 1。然后读取状态寄存器 SR，并判断是否进入相应的状态。如果进行相应的状态就表示完成了初始化。

对于 CAN 总线来说，波特率和线缆的长度是成反比。波特率越高，线缆的长度越短。只有波特率一致才能实现 CAN 之间的相互同性。CAN 的波特率由 BRP、TS1 和 TS2 的值确定。其计算公式为：

$$F_{canfd} = \frac{F_{sysclk}/(BRP+1)}{1+(TS1+1)+(TS2+1)}$$

在初始化完成后开始配置中断使能和接收过滤器。两者的初始化没有先后顺序。CAN FD 支持 32 个接收过滤器同时有效。可以通过接收过滤控制寄存器（AFR）控制。

21.4.2 发送数据流程

发送数据时首先把数据写入待发送 FIFO 缓存区，然后向发送缓存准备就绪请求寄存器（TRR）写入发送指令。CAN FD 接收到发送指令开始组织发送数据。具体的流程如图下图所示：

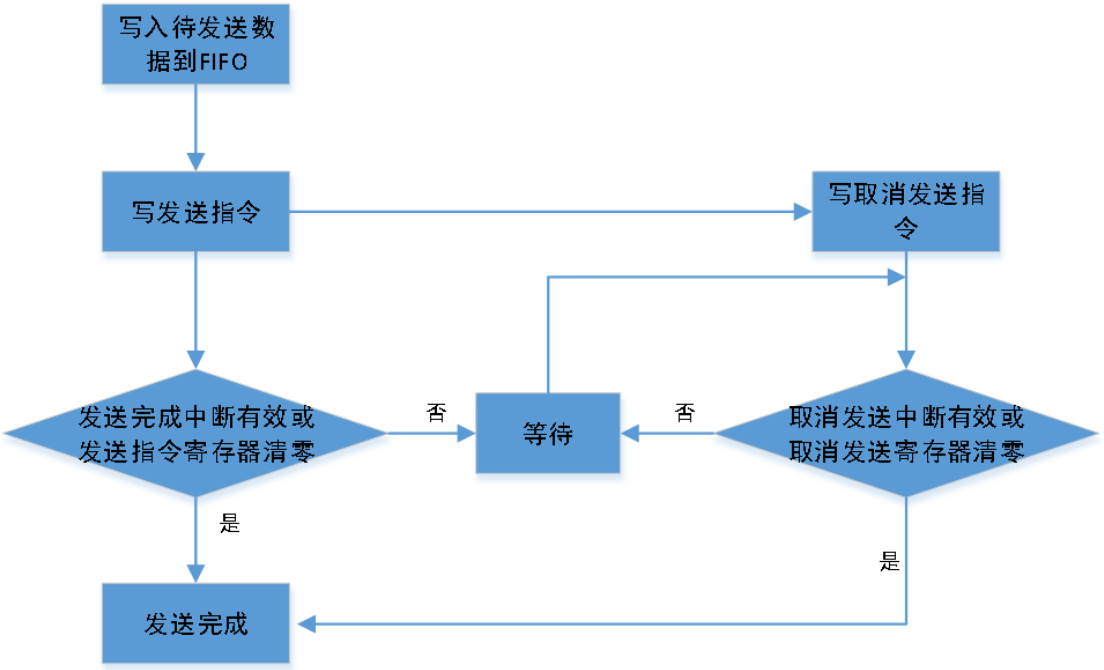


图 21.2 发送数据流程示意图

在写待发送数据时需要正确配置每一帧的 ID、RTR、DLC 等信息。如果发送多帧消息，可以直接向发送请求寄存器（TRR）写发送多帧指令。如发送 TB0 和 TB4，则可以向寄存器 0x90 写 0x00000011。CAN FD 会根据 ID 的优先级发送帧

数据。ID 值越小，优先级越高。CAN FD 完成数据发送后会自动清零发送请求寄存器（TRR）的相应位。如果使能了发送中断位，则发送完成后中断信号有效。

如果发送数据过程中想取消发送某些帧数据发送，则需要向寄存器 0x98 写入相应的指令。如取消发送 TB4，则向 0x98 写入 0x00000010。待 CAN FD 成功取消发送，取消发送寄存器（TCR）的相应位会自动清零。如果使能了取消发送中断位，则中断有效。用户可以通过轮询 TCR 或者检测中断判断是否成功取消发送。

21.4.3 接收数据流程

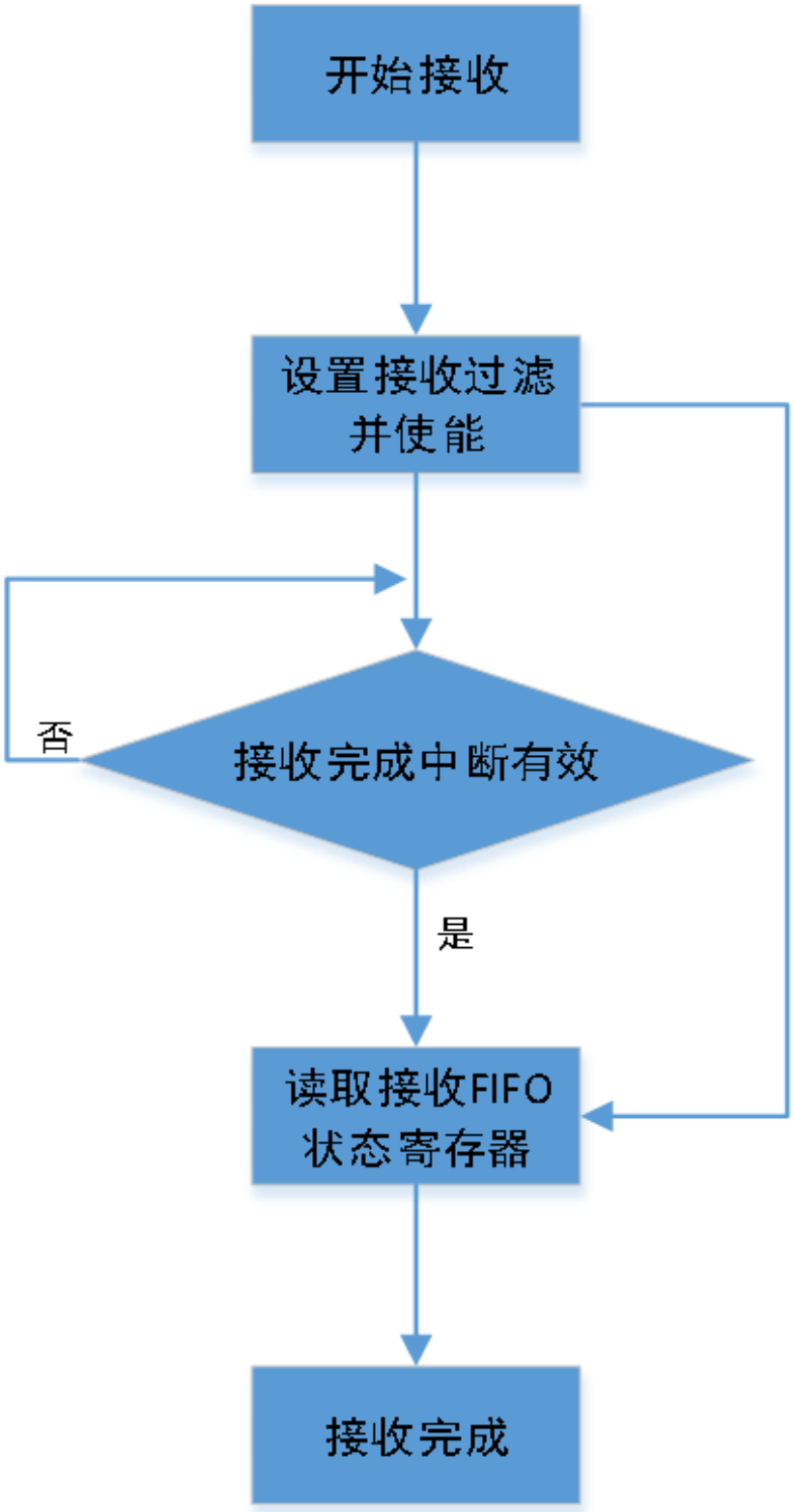


图 21.3 接收数据流程图

在接收数据前需要先设置接收过滤器，并使能接收过滤器使能寄存器（AFR）的相应位。如设置 AFMR0 为 0xFFE00000，AFIR 为 22E00000，且写入 AFR 寄存器的 UAF0 位为 1，则表示只有符合 ID 为 0x117 的消息才能被接收，并存到缓存区。如果使能了接收中断，则接收完数据后中断信号就会拉高。

CAN FD 接收完数据后，用户需要读取接收 FIFO 状态寄存器（RFSR），并根据其值判断接收数据存放的位置。其计算规则见 FRSR 寄存器

21.5 寄存器

CANFD 和需要 32KB 的存储空间。对于空间的地址分配如下表所示

表 21.4 CAN FD 寄存器空间地址分配

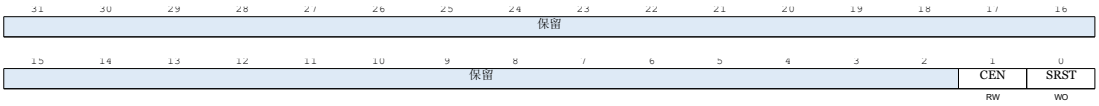
起始地址	结束地址	名称	说明
0x0000	0x00FF	控制寄存器空间	该空间由触发器实现
0x0100	0x09FF	在顺序缓存模式下的发送消息空间	最大支持同时发送 32 帧消息
0x0A00	0x0AFF	接收过滤器掩码寄存器空间	最大同时支持 32 个过滤器
0x0B00	0x1FFF	保留	保留
0x2000	0x20FF	发送事件 FIFO 寄存器空间	只读空间
0x2100	0x29FF	在顺序缓存模式下接收数据空间	最大支持同时接收 32 帧消息
0x2A00	0x7FFF	保留	保留

21.5.1 控制寄存器说明

21.5.1.1 CAN 软件复位寄存器(CAN_SRR)

地址偏移：0x0

复位值：0x00000000

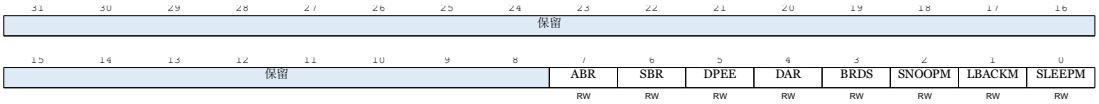


位/位域	名称	描述
31:2	保留	必须保持复位值
1	CEN	CAN 使能 1: CAN 正工作在 LBACK、SLEEP 或者正常模式 0: CAN 工作在配置模式
0	SRST	软件复位 1：复位 CAN。当软件把该位写 1 后，所有的配置寄存器都会被复位。 读取该位时返回 0 0：无影响

21.5.1.2 CAN 模式选择寄存器(CAN_MSR)

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:8	保留	必须保持复位值
7	ABR	BUS-OFF 自动恢复请求 1：BUS-OFF 自动恢复请求有效 0：无 BUS-OFF 自动恢复请求 在 CEN 为 0 时设置该位 如果该位有效，则无论 SBR 位是否设置都无法影响节点 BUS-OFF 恢复。
6	SBR	开始 BUS-OFF 恢复请求 1：开始 BUS-OFF 恢复请求有效 0：无 BUS-OFF 恢复请求

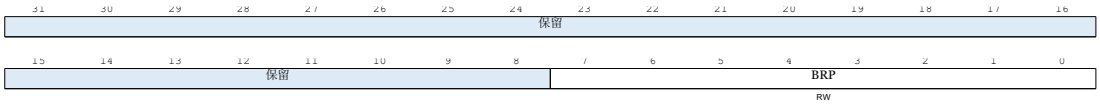
位/位域	名称	描述
		当节点处于 BUS-OFF 状态时，设置此位为 1，则节点完成 BUS-OFF 恢复 在节点完成 BUS-OFF 或者软复位/硬复位后该位自动清零
5	DPEE	无效协议异常时间的检测和产生 1：不检测 CAN FD 帧的 res 位为 1 的情况，CAN FD 接收端也不会产生格式错误 0：如果检测 CAN FD 帧的 res 位为 1，则 CAN FD 接收端进入总线空闲状态。
4	DAR	无效自动重传 1：无效自动重传 0：使能自动重传
3	BRDS	CAN FD 位速率开关 1：CAN FD 帧只能以正产的位速率发送数据 0：通过 CAN FD 帧中 BRS 位判断发送的位速率
2	SNOOPM	SNOOP 模式请求 1: 请求 CAN 处于 SNOOP 模式 0: 无此请求 在 CEN 位为 0 时设置该位。LBACK 和 SLEEP 位必须设置为 0 1、CAN 发送端以“隐性位”在 CAN 总线上； 2、接收其他发送节点的数据但是不返回 ACK； 3、无效错误计数和错误检测 0
1	LBACKM	回环模式请求 1: 请求 CAN 处于回环模式 0: 无此请求

位/位域	名称	描述
		在 CEN 位为 0 时设置该位。SLEEP 和 SNOOP 位必须设置为 0
0	SLEEPM	休眠模式请求 1: 请求 CAN 处于休眠模式 0: 无此请求 在 CEN 位为 0 时设置该位。LBACK 和 SNOOP 位必须设置为 0

21.5.1.3 CAN 仲裁域波特率预分频器(CAN_APBRPR)

地址偏移：0x8

复位值：0x00000000

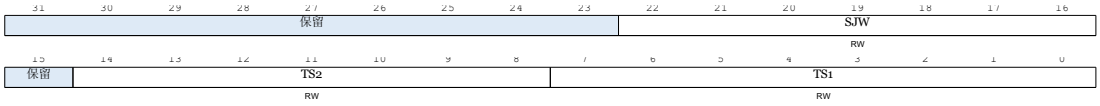


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	BRP	仲裁域正常波特率比例设置

21.5.1.4 CAN 仲裁域位时序寄存器(CAN_APBTR)

地址偏移：0xC

复位值：0x00000000



位/位域	名称	描述
31:23	保留	必须保持复位值
22:16	SJW	同步跳转宽度 在 CEN 位为 0 时设置该位
15	保留	必须保持复位值
14:8	TS2	时序段 2

位/位域	名称	描述
		在 CEN 位为 0 时设置该位
7:0	TS1	时序段 1 在 CEN 位为 0 时设置该位

21.5.1.5 CAN 错误计数寄存器(CAN_ECR)

地址偏移：0x10

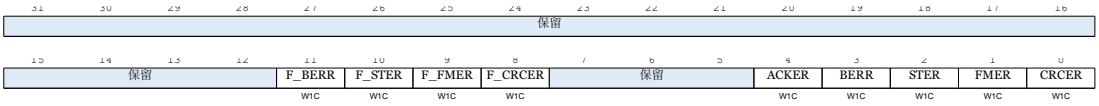
复位值：0x00000000

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	REC	接收错误计数器
7:0	TEC	发送错误计数器

21.5.1.6 CAN 错误状态寄存器(CAN_ESR)

地址偏移：0x14

复位值：0x00000000



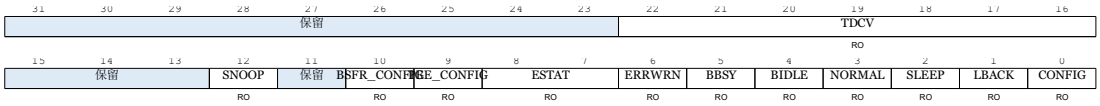
位/位域	名称	描述
31:12	保留	必须保持复位值
11	F_BERR	CAN FD 位错误
10	F_STER	CAN FD 填充位错误
9	F_FMER	CAN FD 格式错误

位/位域	名称	描述
8	F_CRCER	CAN FD CRC 错误
7:5	保留	必须保持复位值
4	ACKER	ACK 错误
3	BERR	位错误
2	STER	填充位错误
1	FMER	格式错误
0	CRCER	CRC 错误

21.5.1.7 CAN 状态寄存器(CAN_SR)

地址偏移：0x18

复位值：0x00000000



位/位域	名称	描述
31:23	保留	必须保持复位值
22:16	TDCV	发送延迟补偿的值
15:13	保留	必须保持复位值
12	SNOOP	SNOOP 模式
11	保留	必须保持复位值
10	BSFR_CONFIG	BUS-OFF 恢复模式指示

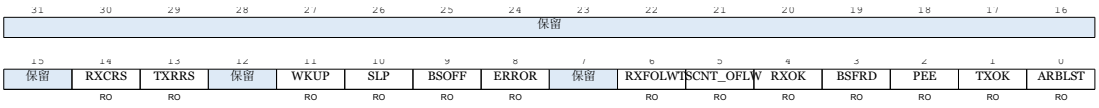
位/位域	名称	描述
9	PEE_CONFIG	PEE 模式
8:7	ESTAT	错误状态 00：配置模式，错误状态未定义 01：主动错误状态 11：被动错误状态 10：BUS-OFF 状态
6	ERRWRN	错误警告 1：一个或多个错误计数器的值大于 96 0：没有错误计数器的值大于 96
5	BBSY	CAN 总线状态指示 1：出接收或发送消息 0：CAN 处于配置状态或者空闲状态
4	BIDLE	总线空闲 1：总线处于空闲状态 0：CAN 处于配置状态或总是处于繁忙
3	NORMAL	正常模式 1：CAN 处于正常模式 0：CAN 处于非正常模式
2	SLEEP	休眠模式 1：CAN 处于休眠模式 0：CAN 处于非休眠模式
1	LBACK	回环模式 1：CAN 处于回环模式 0：CAN 处于非回环模式

位/位域	名称	描述
0	CONFIG	配置模式 1：CAN 处于配置模式 0：CAN 处于非配置模式

21.5.1.8 CAN 中断状态寄存器(CAN_ISR)

地址偏移：0x1C

复位值：0x00000000



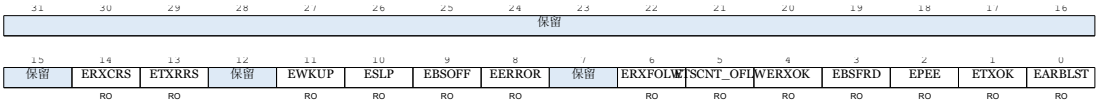
位/位域	名称	描述
31:15	保留	必须保持复位值
14	RXCRS	发送取消发送请求中断
13	TXRRS	发送缓存准备就绪请求中断
12	保留	必须保持复位值
11	WKUP	唤醒中断
10	SLP	休眠中断
9	BSOFF	BUS-OFF 中断
8	ERROR	错误中断
7	保留	必须保持复位值
6	RXFOLW	接收 FIFO0 溢出中断
5	TSCNT_OFLW	时间戳计数器溢出中断

位/位域	名称	描述
4	RXOK	接收消息中断
3	BSFRD	BUS-OFF 恢复完成中断
2	PEE	协议异常事件中断
1	TXOK	发送完成中断
0	ARBLST	仲裁失败中断

21.5.1.9 CAN 中断使能寄存器(CAN_IER)

地址偏移：0x20

复位值：0x00000000



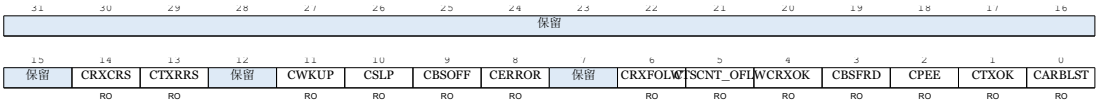
位/位域	名称	描述
31:15	保留	必须保持复位值
14	ERXCRS	使能发送取消发送请求中断
13	ETXRRS	使能发送缓存准备就绪请求中断
12	保留	必须保持复位值
11	EWKUP	使能唤醒中断
10	ESLP	使能休眠中断

位/位域	名称	描述
9	EBSOFF	使能 BUS-OFF 中断
8	EERROR	使能错误中断
7	保留	必须保持复位值
6	ERXFOLW	使能接收 FIFO 溢出中断
5	ETSCNT_OFLW	使能发送完成中断
4	ERXOK	使能接收消息中断
3	EBSFRD	使能 BUS-OFF 恢复完成中断
2	EPEE	使能协议异常事件中断
1	ETXOK	使能发送完成中断
0	EARBLST	使能仲裁失败中断

21.5.1.10 CAN 中断清除寄存器(CAN_ICR)

地址偏移：0x24

复位值：0x00000000



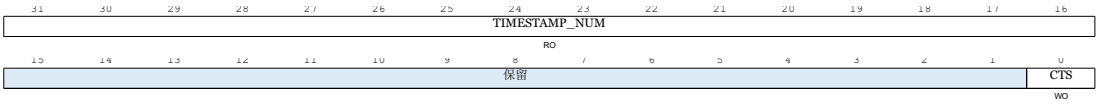
位/位域	名称	描述
31:15	保留	必须保持复位值
14	CRXCRS	清除发送取消发送请求中断

位/位域	名称	描述
13	CTXRRS	清除发送缓存准备就绪请求中断
12	保留	必须保持复位值
11	CWKUP	清除唤醒中断
10	CSLP	使能休眠中断
9	CBSOFF	清除 BUS-OFF 中断
8	CERROR	清除错误中断
7	保留	必须保持复位值
6	CRXFOLW	清除接收 FIFO 溢出中断
5	CTSCNT_OFLW	清除发送完成中断
4	CRXOK	清除接收消息中断
3	CBSFRD	清除 BUS-OFF 恢复完成中断
2	CPEE	清除协议异常事件中断
1	CTXOK	清除发送完成中断
0	CARBLST	清除仲裁失败中断

21.5.1.11 CAN 时间戳寄存器(CAN_TSR)

地址偏移：0x28

复位值：0x00000000



位/位域	名称	描述
31:16	TIMESTAMP_NUM	时间戳计数器的值
15:1	保留	必须保持复位值
0	CTS	清除时间戳计数器

21.5.1.12 CAN 数据域波特率预分频器寄存器(CAN_DPBRPR)

地址偏移：0x88

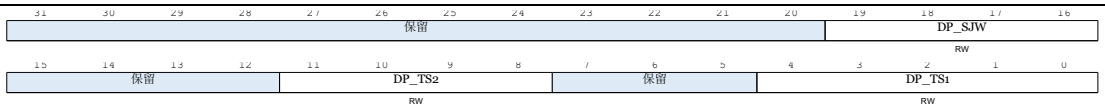
复位值：0x00000000

位/位域	名称	描述
31:17	保留	必须保持复位值
16	TDC	发送延迟补偿使能
15:14	保留	必须保持复位值
13:8	TDCOFF	发送延迟补偿偏移 偏移的单位是 CAN 时钟周期。
7:0	DP_BRP	CAN FD 帧的数据域波特率刻度

21.5.1.13 CAN 数据域位时序寄存器(CAN_DPBTR)

地址偏移：0x8C

复位值：0x00000000

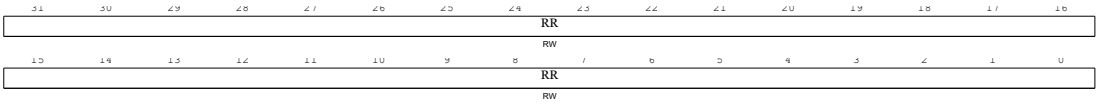


位/位域	名称	描述
31:20	保留	必须保持复位值
19:16	DP_SJW	数据域同步跳转宽度 在 CEN 位为 0 时设置该位
15:12	保留	必须保持复位值
11:8	DP_TS2	数据域时序段 2 CAN FD 标准下的数据位时序，在 CEN 位为 0 时设置该位
7:5	保留	必须保持复位值
4:0	DP_TS1	数据域时序段 1 CAN FD 标准下的数据位时序，在 CEN 位为 0 时设置该位

21.5.1.14 CAN 发送缓存就绪请求寄存器(CAN_TRR)

地址偏移：0x90

复位值：0x00000000

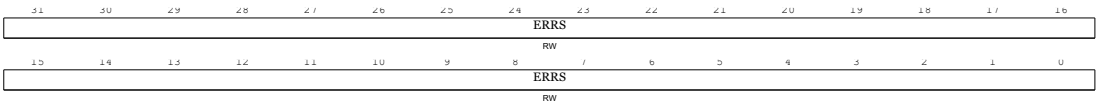


位/位域	名称	描述
31:0	RR	发送缓存 buffer 准备好请求 用户写 1，核发送完数据后自动清除 每 1bit 对应一个发送消息缓存。如 RR[0]为 1，表示 TB0 消息准备好请求发送。

21.5.1.15 CAN 发送缓存就绪请求服务/清除中断使能寄存器(CAN_ERRSR)

地址偏移：0x94

复位值：0x00000000

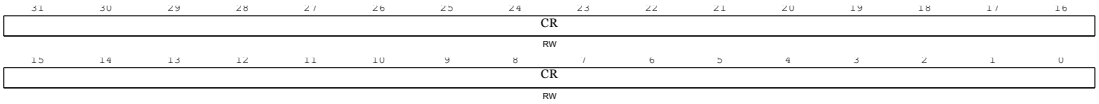


位/位域	名称	描述
31:0	ERRS	中断使能发送缓存 buffer 准备好请求 1：当 TRR 的 RR0 位清零时，ISR 寄存器的 TXRRS 位有效 0：当 TRR 的 RR0 位清零时，ISR 寄存器的 TXRRS 位不会有效

21.5.1.16 CAN 发送缓存取消请求寄存器(CAN_CRR)

地址偏移：0x98

复位值：0x00000000

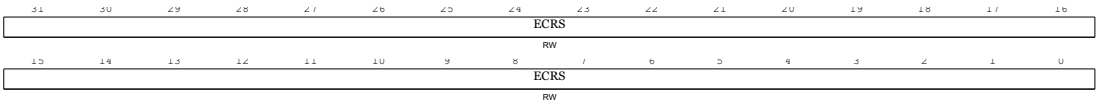


位/位域	名称	描述
31:0	CR	发送缓存取消请求 CR0 置位表示请求取消 TB0 的发送。如果 TB0，已经或者正在发送，则无法取消。 如果 TB0 尚未发送，则可以取消发送。 如果 TB0 发送成功或者取消发送成功 RR0 和 CR0 都会自动清零

21.5.1.17 CAN 发送缓存取消服务/清除中断使能寄存器(CAN_ECRSR)

地址偏移：0x9C

复位值：0x00000000



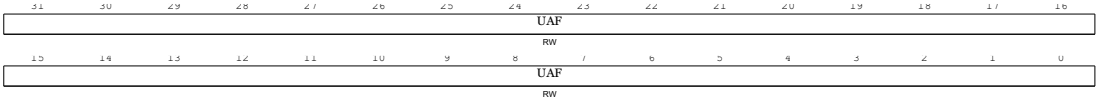
位/位域	名称	描述
31:0	ECRS	发送缓存取消请求中断使能 1：当 TCR 的 CR0 位清零时，ISR 寄存器的 TXCSR 位有效

位/位域	名称	描述
		0：当 TCR 的 CR0 位清零时，ISR 寄存器的 TXCSR 位不有效

21.5.1.18 CAN 接收过滤器寄存器(CAN_AFR)

地址偏移：0xE0

复位值：0x00000000

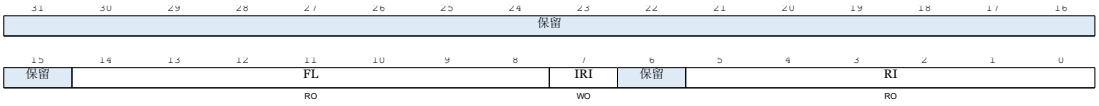


位/位域	名称	描述
31:0	UAF	是否使用接收过滤掩码对 1：使用。使用接收过滤 ID 寄存器。如果 UAF0 有效，则使用 AFMR0 和 AFID0。 0：不使用。

21.5.1.19 CAN 接收 FIFO 状态寄存器(CAN_RFSR)

地址偏移：0xE8

复位值：0x00000000



位/位域	名称	描述
31:15	保留	必须保持复位值
14:8	FL	存储接收的消息个数。其值范围为 0-64。 如果 FL=0x5 RI=0x2，表示有 5 个已经接收的消息尚未读取。
7	IRI	读取一个接收的消息后需要把此位写 1。写 1 后 FL 减 1，RI 加 1
6	保留	必须保持复位值
5:0	RI	接收 FIFO 读取索引，其值范围为 0-63。

位/位域	名称	描述
		如果 RI=0x0，下一个未读的消息地址在 0x2100 如果 RI=0x1，下一个未读的消息地址在 0x2148 RI 在软复位/硬复位时清零

21.5.2 发送消息寄存器说明

发送消息的一帧数据占 18 个 32bit。最多可以存储 32 个消息数据。 该数据存在 RAM 中。具体的划分如下表所示

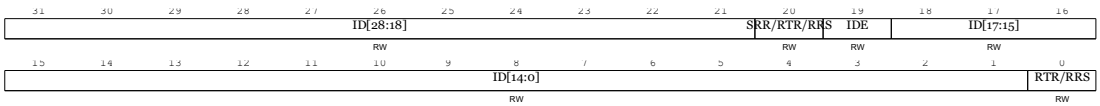
发送消息地址说明

偏移地址	名称	说明
0x0100	TB0-ID	发送消息的 ID 信息
0x0104	TB0-DLC	发送消息的数据长度
0x0108	TB0-DW0	发送消息的数据双字 0
0x010C	TB0-DW1	发送消息的数据双字 1
……	……	……
0x0144	TB0-DW15	发送消息的数据双字 15
0x0148- 0x018C	TB1	发送消息的第二帧 TB1
……	……	……
0x09B8- 0x09FC	TB31	发送消息的第 32 帧 TB31

21.5.2.1 TB*-ID 寄存器

地址偏移：0x0100，0x0148……

复位值：0x00000000



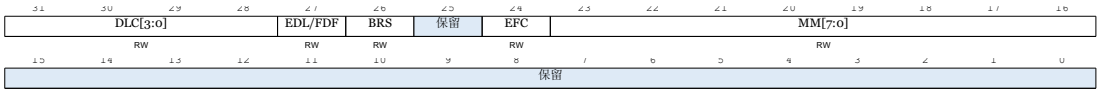
位/位域	名称	说明
31:21	ID[28:18]	标准消息的 ID。CAN 和 CAN FD 标准扩展帧都有效

位/位域	名称	说明
20	SRR/RTR/RRS	代替远程传输请求 对于扩展 CAN 帧和扩展 CAN FD 帧该位为 SRR，必须设置为 1 对于标准 CAN FD 帧为 RRS，必须设置为 0 对于标准 CAN 帧 1：表示 CAN 远程请求帧 0：表示 CAN 数据帧
19	IDE	扩展帧标志 1：表示扩展帧 0：表示标准帧
18:1	ID[17:0]	扩展消息的 ID 仅对扩展帧有效
0	RTR/RRS	远程帧标志位 该位用于区分 CAN 远程帧和数据帧 1：CAN 远程帧 0：CAN 数据帧 对于扩展 CAN FD 帧该位为 RRS，必须设置为 0 对于标准 CAN 和 CAN FD 帧，该位必须写 0

21.5.2.2 TB*-DLC 寄存器

地址偏移：0x0104，0x014C……

复位值：0x00000000



位/位域	名称	说明
31:28	DLC[3:0]	数据长度码，取值范围为 0-15，表示数据长度为 0-64 字节

位/位域	名称	说明
27	EDL/FDF	扩展数据帧标志 1：CAN FD 格式帧 0：CAN 格式帧
26	BRS	位速率开关 在 MSR 寄存器的 BRSD 位为 0 时，此位有效 1：CAN FD 帧的数据域改变速率。 0：CAN FD 帧的数据域不改变速率。
25:24	保留	必须保持复位值
23:16	MM[7:0]	在配置发送缓存是由用户写入发送消息的状态
15:0	保留	必须保持复位值

21.5.2.3 TB*-DW0 寄存器

地址偏移：0x0108，……，0x0150，……

复位值：0x00000000

位/位域	名称	说明
31:24	Data bytes0[7:0]	数据字节 0
23:16	Data bytes1[7:0]	数据字节 1
15:8	Data bytes2[7:0]	数据字节 2
7:0	Data bytes3[7:0]	数据字节 3

21.5.3 接收过滤寄存器说明

支持 32 个接收过滤器。每个接收过滤器可以通过寄存器 AFR 单独使能。接收过滤掩码寄存器（AFMR）包含用于接收过滤器的掩码位。将消息帧的传入消息

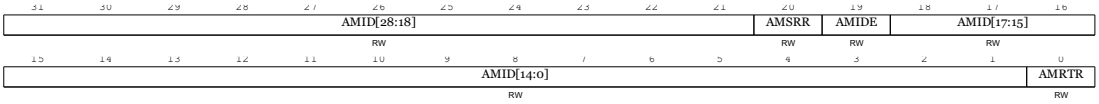
标识符部分与存储在接收过滤器 ID 寄存器的消息标识符进行比较。掩码位定义将存储在接收过滤器 ID 寄存器中的哪些标识符位与 CAN 或 CAN FD 帧传入消息标识符进行比较。对于扩展帧，需要定义所有位字段（AMID[28:18]、AMSSR、AMIDE、AMID[17:0]和 AMRTR）。标准帧只需要定义 AMID[28:18]、AMSRR 和 AMIDE。AMID[17:0]和 AMRTR 应该写 0。

偏移地址	名称	说明
0x0A00	AFMR0	接收过滤器掩码寄存器 0
0x0A04	AFIR0	接收过滤 ID 寄存器 0
0x0A08	AFMR1	接收过滤器掩码寄存器 1
0x0A0C	AFIR1	接收过滤 ID 寄存器 1
0x0A10-0x0AFC	AFMR2-AFMR31 AFIR2-AFIR31	接收过滤器掩码寄存器 2-31 接收过滤 ID 寄存器 2-31

21.5.3.1 AFMR*寄存器(AFMR)

地址偏移：0x0A00,0x0A08，……

复位值：0x00000000



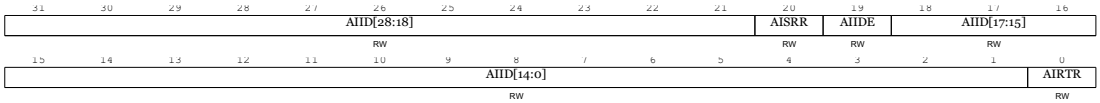
位/位域	名称	说明
31:21	AMID[28:18]	标准消息的 ID 掩码 1：表示在比较传入的消息标识符时使用了接收掩码 ID 寄存器中相应的位。 0：表示没有使用
20	AMSRR	远程帧标志符掩码 1：表示在比较传入的消息标识符时使用了接收掩码 ID 寄存器中相应的位。 0：表示没有使用

位/位域	名称	说明
19	AMIDE	扩展标志位掩码 1：表示在比较传入的消息标识符时使用了接收掩码 ID 寄存器中相应的位。 0：表示没有使用 如果 AMIDE = 1 且相应接受标识符寄存器中的 AIIDE 位为 0，则此掩码仅适用于标准帧。 如果 AMIDE = 1 且相应接受标识符寄存器中的 AIIDE 位为 1，则此掩码仅适用于扩展帧。 如果 AMIDE = 0，则此掩码同时适用于标准帧和扩展帧。
18:1	AMID[18:1]	扩展消息的 ID 掩码 1：表示在比较传入的消息标识符时使用了接收掩码 ID 寄存器中相应的位。 0：表示没有使用
0	AMRTR	扩展帧中的远程标志位掩码 1：表示在比较传入的消息标识符时使用了接收掩码 ID 寄存器中相应的位。 0：表示没有使用

21.5.3.2 AFIR*寄存器(AFIR)

地址偏移：0x0A04,0x0A0C，……

复位值：0x00000000



位/位域	名称	说明
31:21	AIID[28:18]	标准消息的 ID
20	AISRR	远程帧标志符
19	AIIDE	扩展标志位

位/位域	名称	说明
18:1	AIID[18:1]	扩展消息的 ID
0	AIRTR	扩展帧中的远程标志位

21.5.4 接收消息寄存器说明

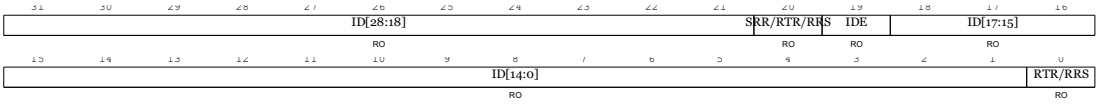
接收消息地址说明

偏移地址	名称	说明
0x2100	RB0-ID	发送消息的 ID 信息
0x2104	RB0-DLC	发送消息的 DLC 信息
0x2108	RB0-DW0	接收消息的数据双字 0
0x210C	RB0-DW1	接收消息的数据双字 1
……	……	……
0x2144	RB0-DW15	接收消息的数据双字 15
0x2148-0x218C	RB1	接收消息 RB1
……	……	……
0x2190-0x29FC	RB2-RB31	接收消息 RB2-RB31

21.5.4.1 RB*-ID 寄存器（RB*-ID）

地址偏移：0x2100，0x2148……

复位值：0x00000000



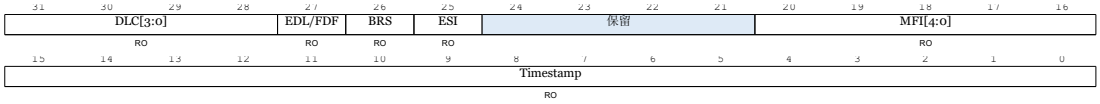
位/位域	名称	说明
31:21	ID[28:18]	标准消息的 ID。CAN 和 CAN FD 标准扩展帧都有效
20	SRR/RTR/RRS	代替远程传输请求 对于扩展 CAN 帧和扩展 CAN FD 帧该位为 SRR，必须设置为 1

位/位域	名称	说明
		对于标准 CAN FD 帧为 RRS，必须设置为 0 对于标准 CAN 帧 1：表示 CAN 远程请求帧 0：表示 CAN 数据帧
19	IDE	扩展帧标志 1：表示扩展帧 0：表示标准帧
18:1	ID[17:0]	扩展消息的 ID 仅对扩展帧有效
0	RTR/RRS	远程帧标志位 该位用于区分 CAN 远程帧和数据帧 1：CAN 远程帧 0：CAN 数据帧 对于扩展 CAN FD 帧该位为 RRS，必须设置为 0 对于标准 CAN 和 CAN FD 帧，该位必须写 0

21.5.4.2 RB*-DLC 寄存器（RB*-DLC

地址偏移：0x2104，0x214C……

复位值：0x00000000



位/位域	名称	说明
31:28	DLC[3:0]	数据长度码，取值范围为 0-15，表示数据长度为 0-64 字节
27	EDL/FDF	扩展数据帧标志 1：CAN FD 格式帧 0：CAN 格式帧

位/位域	名称	说明
26	BRS	位速率开关 在 MSR 寄存器的 BRSD 位为 0 时，此位有效 1：CAN FD 帧的数据域改变速率。 0：CAN FD 帧的数据域不改变速率。
25	ESI	错误指示 1：接收的帧所在的发送端处于被动错误状态 0：接收的帧所在的发送端处于主动错误状态
24:21	保留	必须保持复位值
20:16	MFI[4:0]	在配置发送缓存是由用户写入发送消息的状态(Matched_Filter_Index)
15:0	Timestamp	检测到 SOF 位后，如果过成功接收到消息，则记录下时间戳

21.5.4.3 RB*-DW0 寄存器（RB*-DW0）

地址偏移：0x2108，……，0x2150，……

复位值：0x00000000

位/位域	名称	说明
31:24	Data bytes0[7:0]	数据字节 0
23:16	Data bytes1[7:0]	数据字节 1
15:8	Data bytes2[7:0]	数据字节 2
7:0	Data bytes3[7:0]	数据字节 3

22 集成电路总线（IIC）

22.1 简介

IIC 模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 IIC 设备的通讯。IIC 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。 IIC 接口模块实现了 IIC 协议的标速模式和快速模式，支持多主机 IIC 总线架构。

22.2 主要特征

- 同一接口既可实现主机功能又可实现从机功能
- 主从机之间的双向数据传输
- 支持 7 位的地址寻址模式
- 支持 IIC 多主机模式
- 支持标速（最高 100kHz），快速（最高 400kHz）
- 从机模式下可配置的 SCL 主动拉低
- 三种中断：发送中断、接收中断与传输步骤中断

22.3 功能描述

IIC 接口的内部结构如下图所示。

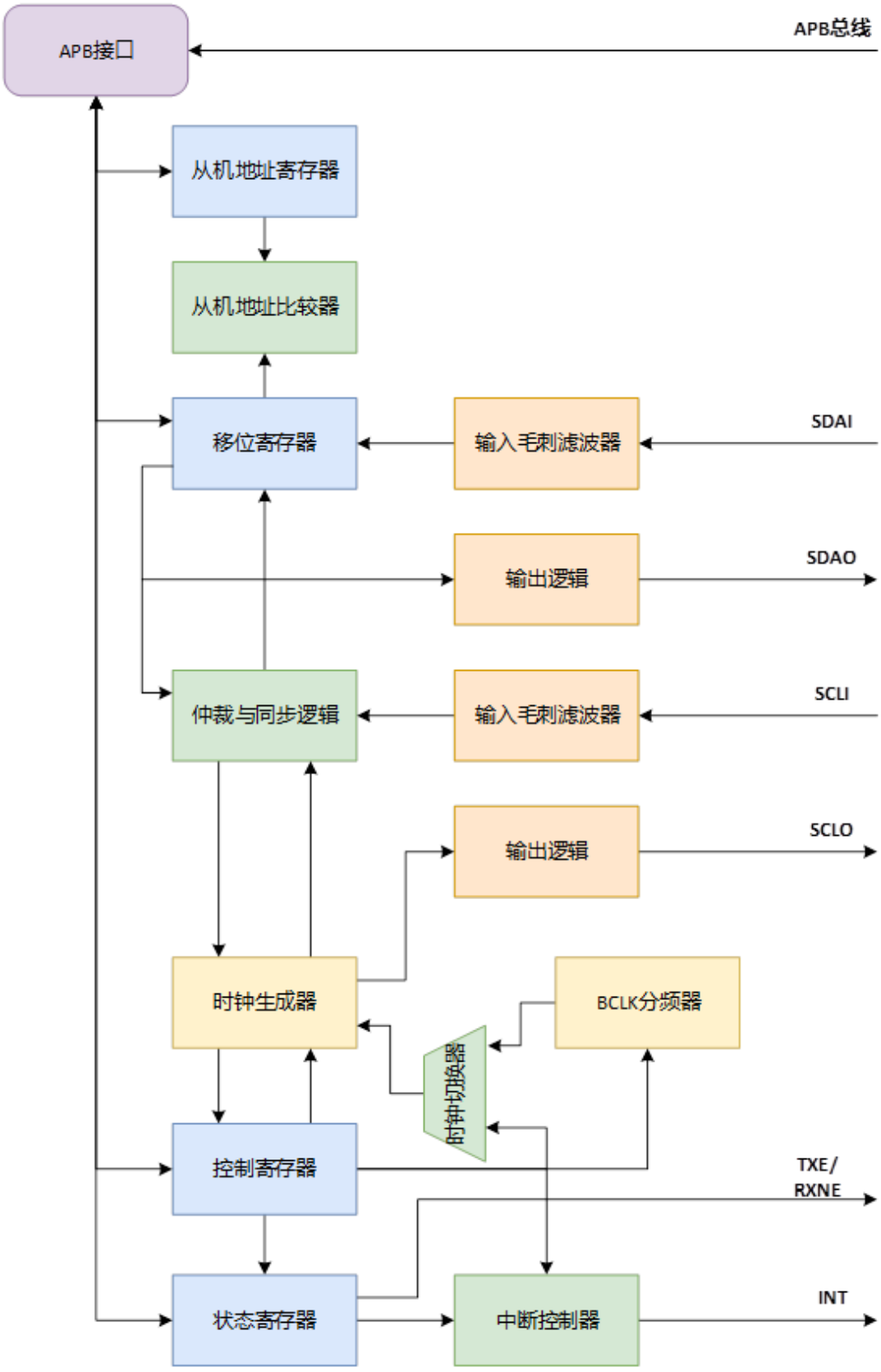


图 22.1 IIC 的内部结构图

22.3.1 SDA 线和 SCL 线

IIC 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。两个接口的 GPIO 必须都配置为复用开漏输出，以提供线与功能。IIC 总线上的数据在标准模式下可以达到 100Kbit/s，在快速模式下可以达到 400Kbit/s。

22.3.2 软件编程模型

一个 IIC 设备例如 LCD 驱动器可能只是作为一个接收器，但是一个存储器既可以接收数据，也能发送数据。除了按照发送/接收方来区分，IIC 设备也分为数据传输的主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。 不管 IIC 设备是主机还是从机，都可以发送或接收数据，因此，IIC 设备有以下 4 种运行模式：

- 主机发送方；
- 主机接收方；
- 从机发送方；
- 从机接收方。

IIC 模块支持以上四种模式。IIC 在使能后默认工作在从机模式下。通过软件配置使 IIC 在总线上发送一个 START 起始位之后，IIC 变为主机模式。

22.3.2.1 从机软件流程

在从机模式下，IIC 将在总线上监听自己的地址。如果检测到自己的地址，内核将切换到寻址从属模式并将 IIC_CTL 的 IICI 位置 1。之后 IIC 可以作为从机发送方或从机接收方工作。

22.3.2.1.1 从机发送流程

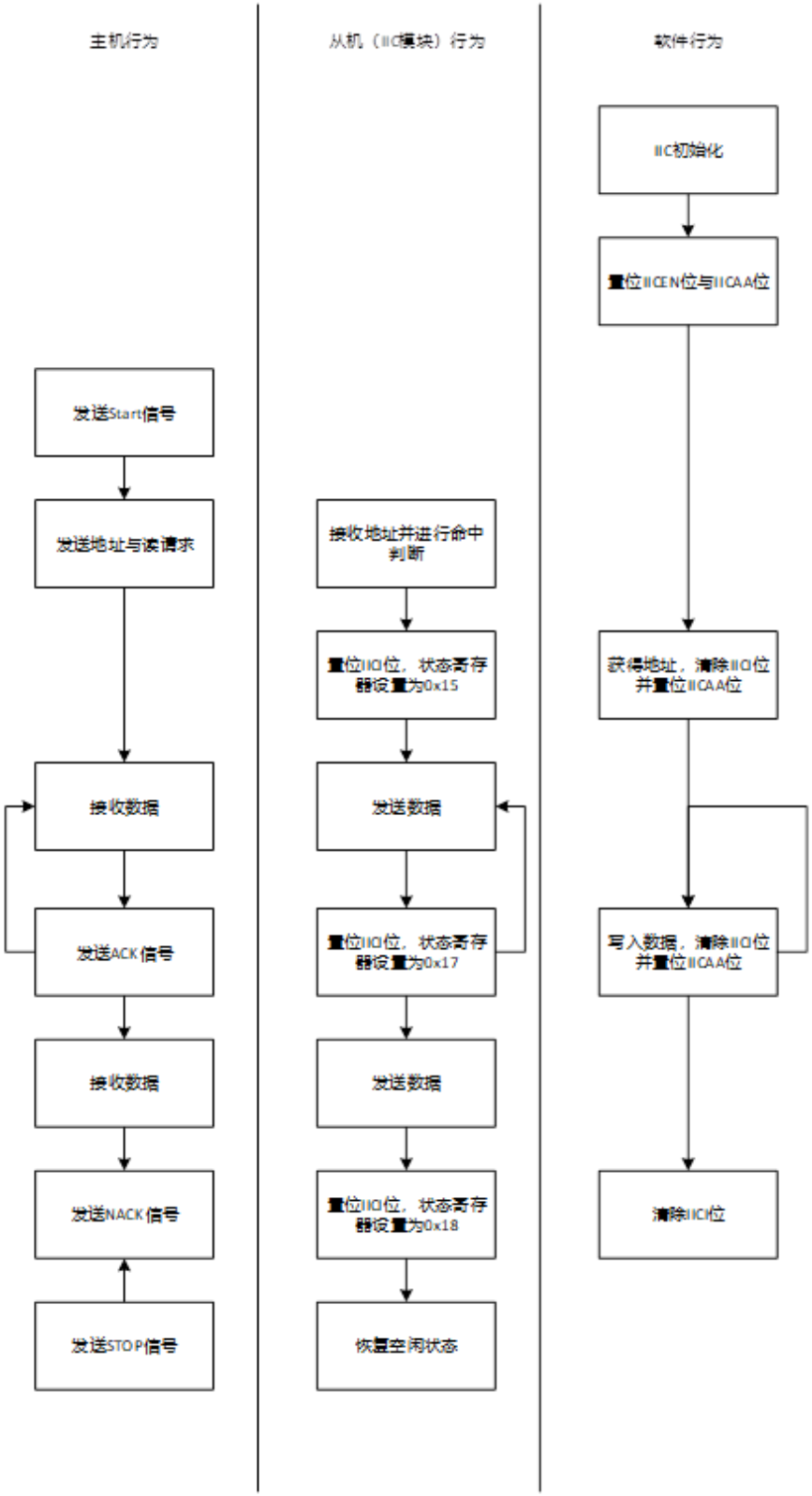


图 22.2 从机发送流程图

1. 软件设置 IICEN 和 IICAA 位；
2. IIC 接收自己的地址并将传输方向位设置为发送；
3. IIC 将 IIC_CTL 的 IICI 位置 1；IIC_STAT 的 STATE 位置为 0x15；
4. 软件写入要发送的数据数据，然后清除 IIC_CTL 的 IICI 位并置位 IICAA 位；
5. IIC 发送下一个数据字节，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位可能为 0x17 或 0x18，若为 0x17 则返回步骤 4 继续发送；
6. 清除 IICI 位，结束此次传输；
7. IIC 在接到 STOP 后恢复空闲状态。

22.3.2.1.2 从机接收流程

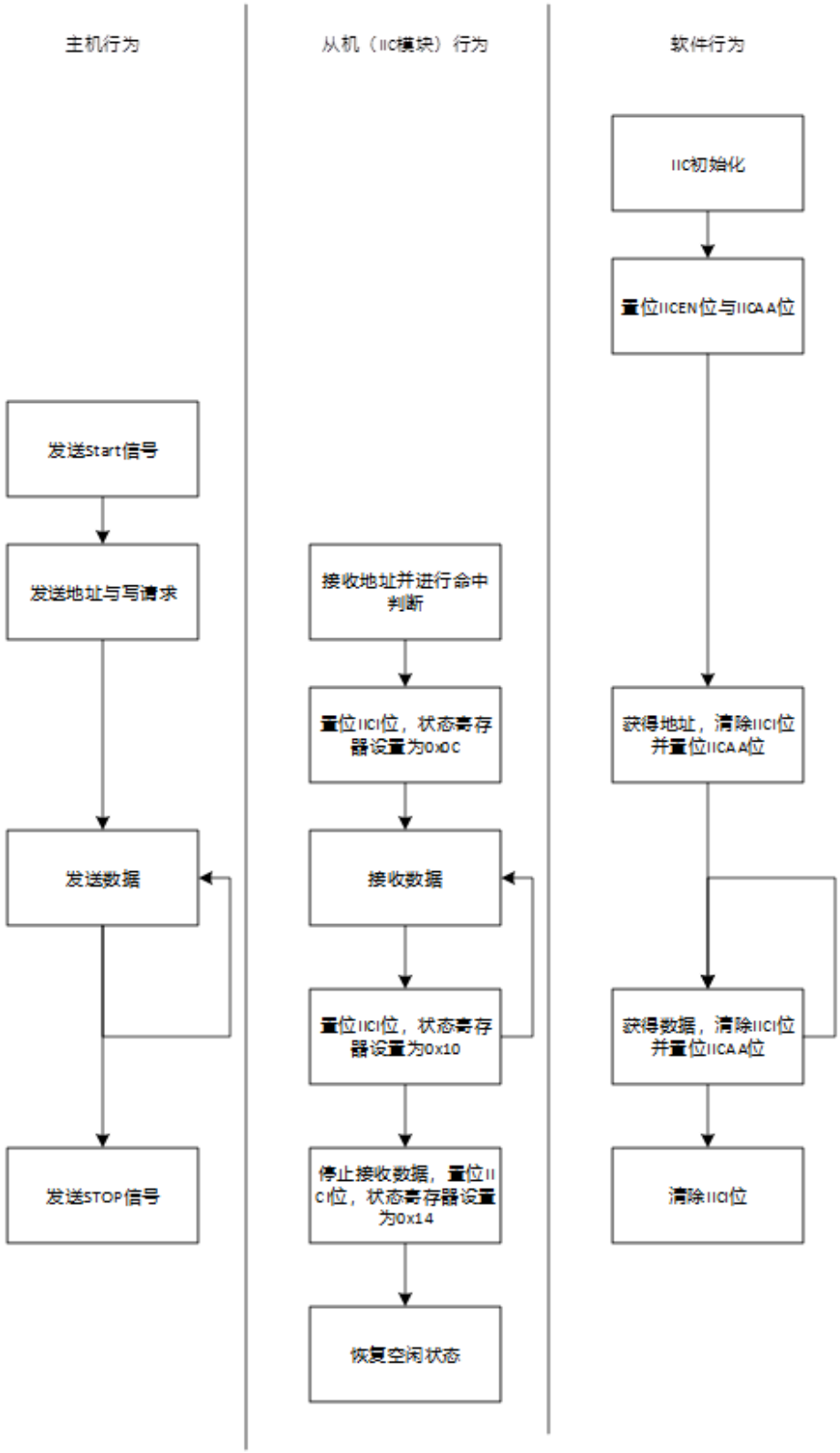


图 22.3 从机接收流程图

1. 软件设置 IICEN 和 IICAA 位；
2. IIC 接收自己的地址并将传输方向位设置为接收；
3. IIC 将 IIC_CTL 的 IICI 位置 1；IIC_STAT 的 STATE 位置为 0x0C；
4. 软件准备接收数据，然后清除 IIC_CTL 的 IICI 位并置位 IICAA 位；
5. IIC 接收下一个数据字节，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位可能为 0x10 或 0x14，若为 0x10 则返回步骤 4 继续接收；
6. 清除 IICI 位，结束传输。

22.3.2.2 主机软件流程

切换为主模式之前，IIC 将等待总线空闲。当总线空闲时，IIC 将发送一个 START 信号，并发送从机地址（希望控制的从机）和传输方向位。根据传输方向位，IIC 可以作为主发射机或主接收机工作。

22.3.2.2.1 主机发送流程

图 22.4 主机发送流程图

1. 软件设置 IICEN 和 IICSTA 位；
2. IIC 发送 START 信号，然后将 IIC_CTL 的 IICI 位置 1；IIC_STAT 的 STATE 位置为 0x01。
3. 软件写入要操作的从机的地址，然后清除 IIC_CTL 的 IICI 位；
4. IIC 发送数据寄存器的地址，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位置为 0x03；
5. 软件写入数据并清除 IICI 位。
6. IIC 发送数据，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位置为 0x05，若数据传输未完成，则返回步骤 5；
7. 将 IICSTO 位置 1 并清除 IICI 位，结束传输。

22.3.2.2.2 主机接收流程

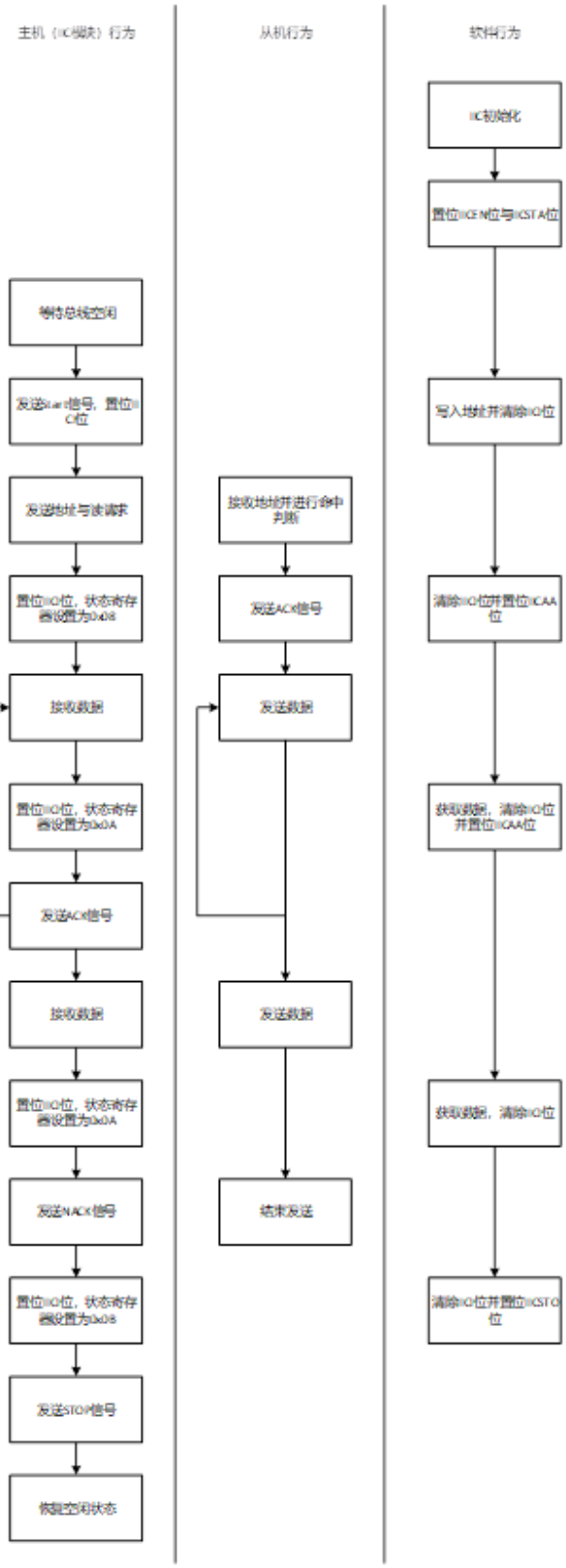


图 22.5 主机接收流程图

1. 软件设置 IICEN 和 IICSTA 位；
2. IIC 发送 START 信号，然后将 IIC_CTL 的 IICI 位置 1；IIC_STAT 的 STATE 位置为 0x01；
3. 软件写入要操作的从机的地址，然后清除 IIC_CTL 的 IICI 位；
4. IIC 发送地址，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位置为 0x08；
5. 软件清除 IICI 位并置位 IICAA 位；
6. IIC 接收数据，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位置为 0x0A；
7. 软件读取数据，清除 IICI 位，若继续读取，置位 IICAA 位；
8. IIC 发送 NACK，然后将 IIC_CTL 的 IICI 位置 1。IIC_STAT 的 STATE 位置为 0x0B；
9. 软件清除 IICI 位并置位 IICSTO 位；
10. IIC 发送 STOP 信号，结束此次传输。

22.4 寄存器

22.4.1 IIC 控制寄存器（IIC_CTLR）

地址偏移：0x00

复位值：0x00

该寄存器只支持按字节（8 位）访问

7	6	5	4	3	2	1 0
IICLK[2]	IICEN	IICSTA	IICSTO	IICI	IICAA	IICLK[1:0]
RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
7,1:0	IICLK[2:0]	时钟选择位 000：PCLK/256 001：PCLK/224 010：PCLK/192 011：PCLK/160

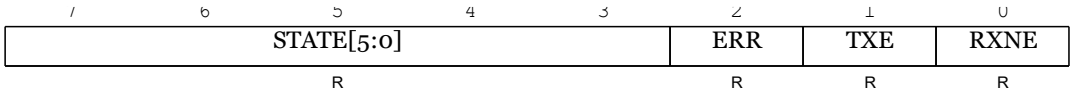
位/位域	名称	描述
		100：PCLK/960 101：PCLK/120 110：PCLK/60 111: PCLK /8/DIV DIV 为 16 位数值，通过 IIC_DIV_H 和 IIC_DIV_L 寄存器设置
6	IICEN	IIC 使能位 0：sda 和 scl 输出处于高阻抗状态，sda 和 scl 输入信号被忽略 1：IIC 启用
5	IICSTA	开始标志发送位 0：无动作 1：IIC 模块检查总线的状态，并在总线空闲时生成 START 信号
4	IICSTO	停止标志发送位 0：无动作 1：IIC 处于主模式时，将 STOP 信号传输至总线。当总线上出现 STOP 标志位时，该位自动清除
3	IICI	逐次操作中断位 只要 IIC_STA 中有可维护的更改，IICI 标志位就会被置位。寄存器更新后，APB 主机必须清除 IICI 位。当作为从发送器/接收器工作时，延迟清除该位会实现时钟拉伸
2	IICAA	接收确认标志发送位 0：主/从模式下，接收到一个数据时发送 NACK 1：主/从模式下，接收到一个数据时发送 ACK

22.4.2 IIC 状态寄存器（IIC_STAT）

地址偏移：0x04

复位值：0xF8

该寄存器只支持按字节（8 位）访问



位/位域	名称	描述
7:3	STATE[5:0]	IIC 运行状态位，具体状态参见 IIC 状态真值表
2	ERR	传输错误状态位 STATE 位为 0x04，0x06，0x09，0x0B，0x07，0x18，0x09 时该位置 1 完成错误处理操作后清零
1	TXE	发送空闲状态位 STATE 位为 0x03，0x05，0x15，0x16，0x17 时该位置 1 完成数据发送操作后清零
0	RXNE	接收空闲状态位 STATE 位为 0x0A，0x0B，0x10，0x12 时该位置 1 完成数据接收操作后清零

22.4.3 IIC 数据寄存器（IIC_DATA）

地址偏移：0x08

复位值：0x00

该寄存器只支持按字节（8 位）访问

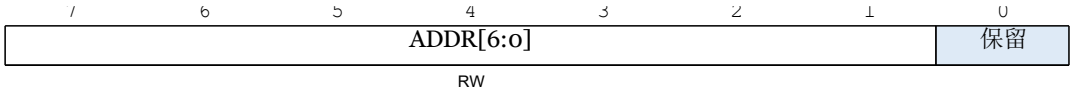
位/位域	名称	描述
7:0	DATA[7:0]	IIC 数据位，写入 IIC 发送的数据或读取 IIC 接收的数据 在发送地址时，第 7 到 1 位为地址位，第 0 位为读写位，写为 0 读为 1

22.4.4 IIC 从机地址寄存器（IIC_ADDR）

地址偏移：0x0C

复位值：0x00

该寄存器只支持按字节（8 位）访问



位/位域	名称	描述
7:1	ADDR[6:0]	IIC 从机地址位，当 IIC 作为从机时的地址
0	保留	必须保持复位值

22.4.5 IIC 中断控制寄存器（IIC_IRQC）

地址偏移：0x14

复位值：0x00

该寄存器只支持按字节（8 位）访问

位/位域	名称	描述
7:4	保留	必须保持复位值
3	ERRN	传输错误中断使能位 如果该位置 1，IIC_STAT 寄存器中 ERR 被置位时产生中断
2	SIEN	逐次操作中断使能位

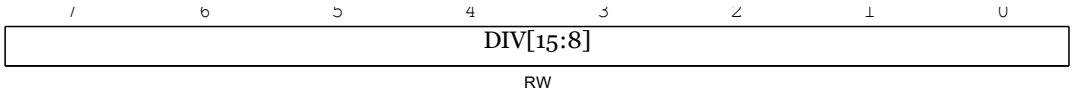
位/位域	名称	描述
		如果该位置 1，IIC_CTLR 寄存器中 IICI 被置位时产生中断
1	TXEN	发送空闲中断使能位 如果该位置 1，IIC_STAT 寄存器中 TXE 被置位时产生中断
0	RXNEN	接收空闲中断使能位 如果该位置 1，IIC_STAT 寄存器中 RXNE 被置位时产生中断

22.4.6 IIC 分频系数寄存器 H（IIC_DIV_H）

地址偏移：0x18

复位值：0x00

该寄存器只支持按字节（8 位）访问



位/位域	名称	描述
7:0	DIV[15:8]	分频系数高 8 位 用于自定义 IIC 输出时钟，与 IIC_CTLR 寄存器配合使用

22.4.7 IIC 分频系数寄存器 L（IIC_DIV_L）

地址偏移：0x10

复位值：0x02

该寄存器只支持按字节（8 位）访问

位/位域	名称	描述
7:0	DIV[7:0]	分频系数低 8 位

位/位域	名称	描述
		用于自定义 IIC 输出时钟，与 IIC_CTLR 寄存器配合使用

22.4.8 IIC 状态寄存器真值表

表 22.1 主机发送模式

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x01	发送一个 START 信号	写入地址与写信号	—	0	0	—	将会发送地址与写信号；预计收到从机的 ACK 信号
0x02	重复发送一个 START 信号	写入地址与写信号	—	0	0	—	将会发送地址与写信号；预计收到从机的 ACK 信号
		写入地址与读信号	—	0	0	—	将会发送地址与读信号；切换到主机接收模式
0x1c	发送一个 STOP 信号	没有动作	—	—	—	—	没有动作
0x03	向从机发送地址与写信号完成，收到 ACK 信号	写入要发送的数据	0	0	0	—	将会发送数据；预计收到从机的 ACK 信号

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
		没有动作	1	0	0	—	将会重复发送一遍 START 信号
		没有动作	0	1	0	—	将会发送 STOP 信号；将自动清零 IICSTO
		没有动作	1	1	0	—	将会连续发送一个 STOP 信号与一个 START 信号；将自动清零 IICSTO
0x04	向从机发送地址与写信号完成，收到 NACK 信号	没有动作	1	0	0	—	将会重复发送一遍 START 信号
		没有动作	0	1	0	—	将会发送 STOP 信号；将自动清零 IICSTO
		没有动作	1	1	0	—	将会连续发送一个 STOP 信号与一个 START 信号；将自动清零 IICSTO
0x05	数据寄存器的数据发送已完成，收到 ACK 信号	写入要发送的数据	0	0	0	—	将会发送数据；预计收到从机的 ACK 信号

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
		没有动作	1	0	0	—	将会重复发送一遍 START 信号
		没有动作	0	1	0	—	将会发送 STOP 信号；将自动清零 IICSTO
		没有动作	1	1	0	—	将会连续发送一个 STOP 信号与一个 START 信号；将自动清零 IICSTO
0x06	数据寄存器的数据发送已完成，收到 NACK 信号	没有动作	1	0	0	—	将会重复发送一遍 START 信号
		没有动作	0	1	0	—	将会发送 STOP 信号；将自动清零 IICSTO
		没有动作	1	1	0	—	将会连续发送一个 STOP 信号与一个 START 信号；将自动清零 IICSTO
0x07	发送地址或数据时仲裁丢失	没有动作	0	0	0	—	将会释放总线；切换到从机模式
		没有动作	1	0	0	—	将会在总线空闲时发送 START 信号

表 22.2 主机接收模式

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x01	发送一个 START 信号	写入地址与读信号	—	0	0	—	将会发送地址与读信号；预计收到从机的 ACK 信号
0x02	重复发送一个 START 信号	写入地址与读信号	—	0	0	—	将会发送地址与读信号；预计收到从机的 ACK 信号
		写入地址与写信号	—	0	0	—	将会发送地址与写信号；切换到主机发送模式
0x07	仲裁丢失	没有动作	0	0	0	—	将会释放总线；切换到从机模式
		没有动作	1	0	0	—	将会在总线空闲时发送 START 信号
0x08	向从机发送地址与读信号完成，收到 ACK 信号	没有动作	0	0	0	0	将会接收从机的数据；发送 NACK 信号
		没有动作	0	0	0	1	将会接收从机的数据；发送 ACK 信号

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x09	向从机发送地址与读信号完成，收到 NACK 信号	没有动作	1	0	0	—	将会重复发送一遍 START 信号
		没有动作	0	1	0	—	将会发送 STOP 信号；将自动清零 IICSTO
		没有动作	1	1	0	—	将会连续发送一个 STOP 信号与一个 START 信号；将自动清零 IICSTO
0x0A	数据寄存器的数据接收已完成，已回复 ACK 信号	读取数据	0	0	0	0	将会接收从机的数据；发送 NACK 信号
		读取数据	0	0	0	1	将会接收从机的数据；发送 ACK 信号
0x0B	数据寄存器的数据接收已完成，已回复 NACK 信号	读取数据	1	0	0	—	将会重复发送一遍 START 信号
		读取数据	0	1	0	—	将会发送 STOP 信号；将自动清零 IICSTO
		读取数据	1	1	0	—	将会连续发送一个 STOP 信号与一个 START 信号；将自动清零 IICSTO

表 22.3 从机接收模式

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x0C	接收到主机发送的地址与写信号；已回复 ACK 信号	没有动作	—	0	0	0	将会接收主机的数据，之后会向主机回复 NACK 信号
		没有动作	—	0	0	1	将会接收主机的数据，之后会向主机回复 ACK 信号
0x0D	作为主机发送地址与读写位后仲裁丢失，且接收到其他主机发送的地址与写信号；已回复 ACK 信号	没有动作	—	0	0	0	将会接收主机的数据，之后会向主机回复 NACK 信号
		没有动作	—	0	0	1	将会接收主机的数据，之后会向主机回复 ACK 信号
0x0E	收到广播寻址（0x00）；已回复 ACK 信号	没有动作	—	0	0	0	将会接收主机的数据，之后会向主机回复 NACK 信号
		没有动作	—	0	0	1	将会接收主机的数据，之后会向

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
							主机回复 ACK 信号
0x0F	作为主机发送地址与读写位后仲裁丢失，且接收到其他主机发送的广播寻址；已回复 ACK 信号	没有动作	—	0	0	0	将会接收主机的数据，之后会向主机回复 NACK 信号
		没有动作	—	0	0	1	将会接收主机的数据，之后会向主机回复 ACK 信号
0x10	之前被主机成功寻址；数据已接收；已回复 ACK 信号	读取数据	—	0	0	0	将会接收主机的数据，之后会向主机回复 NACK 信号
		读取数据	—	0	0	1	将会接收主机的数据，之后会向主机回复 ACK 信号
0x11	之前被主机成功寻址；数据已接收；已回复 NACK 信号	读取数据	0	0	0	0	进入未被寻址的从机模式；将会不识别自身从机地址

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
		读取数据	1	0	0	0	进入未被寻址的从机模式；不识别自身从机地址；将会在总线空闲时发送 START 信号
		读取数据	0	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址
		读取数据	1	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址；将会在总线空闲时发送 START 信号
0x12	之前被广播寻址；数据已接收；已回复 ACK 信号	读取数据	—	0	0	0	将会接收主机的数据，之后会向主机回复 NACK 信号
		读取数据	—	0	0	1	将会接收主机的数据，之后会向

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
							主机回复 ACK 信号
0x13	之前被广播寻址；数据已接收；已回复 NACK 信号	读取数据	0	0	0	0	进入未被寻址的从机模式；将会不识别自身从机地址
		读取数据	1	0	0	0	进入未被寻址的从机模式；不识别自身从机地址；将会在总线空闲时发送 START 信号
		读取数据	0	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址
		读取数据	1	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址；将会在总线空闲时发送 START 信号

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x14	接收到 STOP 信号或重复的 START 信号	读取数据	0	0	0	0	进入未被寻址的从机模式；将会不识别自身从机地址
		读取数据	1	0	0	0	进入未被寻址的从机模式；不识别自身从机地址；将会在总线空闲时发送 START 信号
		读取数据	0	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址
		读取数据	1	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址；将会在总线空闲时发送 START 信号

表 22.4 从机发送模式

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x15	接收到主机发送的地址与读信号；已回复 ACK 信号	写入要发送的数据	—	0	0	0	将会向主机发送最后一个数据；预计收到主机的 NACK 信号
		写入要发送的数据	—	0	0	1	将会向主机发送数据；预计收到主机的 ACK 信号
0x16	作为主机发送地址与读写位后仲裁丢失，且接收到其他主机发送的地址与读信号；已回复 ACK 信号	写入要发送的数据	—	0	0	0	将会向主机发送最后一个数据；预计收到主机的 ACK 信号
		写入要发送的数据	—	0	0	1	将会向主机发送数据；预计收到主机的 ACK 信号
0x17	数据发送完成；收到 ACK 信号	写入要发送的数据	—	0	0	0	将会向主机发送最后一个数据；预计收到主机的 NACK 信号

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
		写入要发送的数据	—	0	0	1	将会向主机发送数据；预计收到主机的 ACK 信号
0x18	数据发送完成；收到 NACK 信号	读取数据	0	0	0	0	进入未被寻址的从机模式；将会不识别自身从机地址
		读取数据	1	0	0	0	进入未被寻址的从机模式；不识别自身从机地址；将会在总线空闲时发送 START 信号
		读取数据	0	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址
		读取数据	1	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址；将会在总线空闲时发送 START 信号

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
0x19	最后一个数据发送完成；收到 ACK 信号	读取数据	0	0	0	0	进入未被寻址的从机模式；将会不识别自身从机地址
		读取数据	1	0	0	0	进入未被寻址的从机模式；不识别自身从机地址；将会在总线空闲时发送 START 信号
		读取数据	0	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址
		读取数据	1	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址；将会在总线空闲时发送 START 信号
0x14	接收到 STOP 信号或重复的 START 信号	读取数据	0	0	0	0	进入未被寻址的从机模式；将会不识别自身从机地址
		读取数据	1	0	0	0	进入未被寻址的从机模式；不识别自

状态位	状态	数据寄存器动作	控制寄存器动作				IIC 执行动作
			IICSTA	IICSTO	IICI	IICAA	
							身从机地址；将会在总线空闲时发送 START 信号
		读取数据	0	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址
		读取数据	1	0	0	1	进入未被寻址的从机模式；将会识别并响应自身从机地址；将会在总线空闲时发送 START 信号

23 串行外围设备接口（SPI）

23.1 简介

SPI 全称是 serial peripheral interface，串行外围设备接口，支持全双工的同步串行通信。该接口可配置为主机或从机模式，配置为主机模式时，它可为外部从器件提供通信时钟（SCK），6 个 SPI 每个都支持 4 个从机。

23.2 主要特征

- 支持 Motorola 模式串行外设接口（SPI）
- 4~32 位传输帧格式选择
- SPI 时钟速率可配置
- 主模式或从模式操作
- 主模式时最高频率 PCLK/2
- 从模式时最高频率 PCLK/8
- 数据发送顺序 MSB 在前
- 可编程的时钟极性和相位
- 支持中断

23.3 功能说明

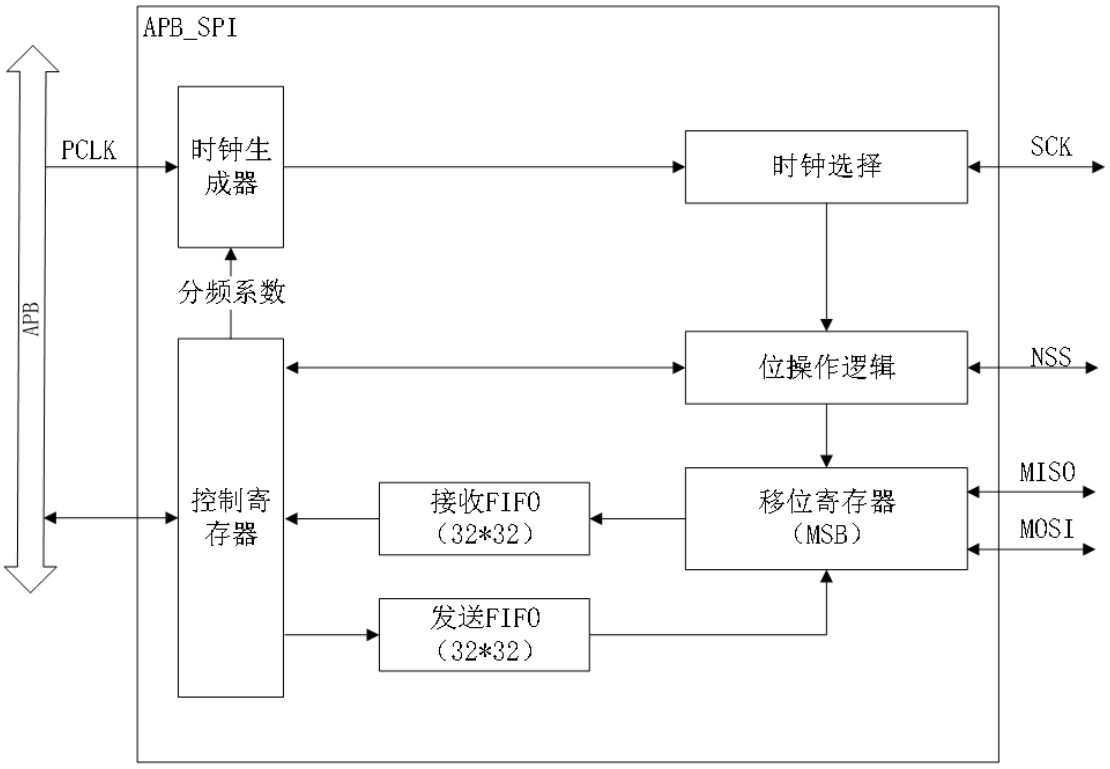


图 23.1 系统框图

上图中的 SPI 时钟生成器根据用户配置的分频系数产生 SPI 的发送时钟，配置为主机模式时，最高频率支持 PCLK/2，配置为从机时，采样时钟为 PCLK，支持的 SPI 频率最高到 PCLK/8。发送和接收的 FIFO 的位宽和深度都为 32，每次发送或者接收都占用一个字，如果设置的帧大小不足 32 位，则 FIFO 的对应字的剩余位无效，例如，设置的帧大小为 8 位，则 FIFO 的每个字的[31:8]位为无效的。选择的帧大小对发送和接收都有效。SPI 只支持 MSB，发送和接收时第一位为最高位。

23.3.1 Motorola SPI 时序和数据帧格式

SPI 模式寄存器中的模式选择位决定了 SPI 时钟和数据信号的时序。CFG_MOT_MODE 位决定了空闲状态时 SCK 的电平以及第一个或第二个时钟跳变沿为有效采样边沿。

- 模式 0 时，SCK 默认为低电平，在第一个时钟跳变沿采样。

- 模式 1 时，SCK 默认为低电平，在第二个时钟跳变沿采样。
- 模式 2 时，SCK 默认为高电平，在第一个时钟跳变沿采样。
- 模式 3 时，SCK 默认为高电平，在第二个时钟跳变沿采样。

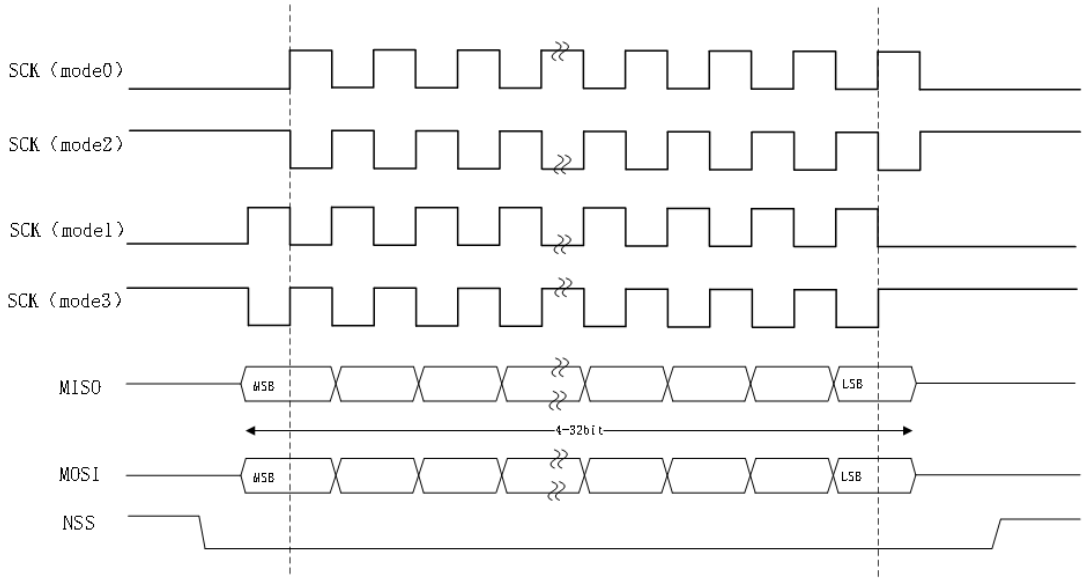


图 23.2 Motorola SPI 时序图

23.3.2 主机模式

SPI 配置为主机时，CPU 写发送数据寄存器时，数据被写入到发送 FIFO 中，当发送 FIFO 非空时即自动启动发送流程，直到发送 FIFO 空，并产生对应的发送完成中断。

SPI 配置为主机时，SCK 由主机产生，频率由时钟分频配置寄存器配置。下图为主机模式时一个主机和一个从机的结构图。

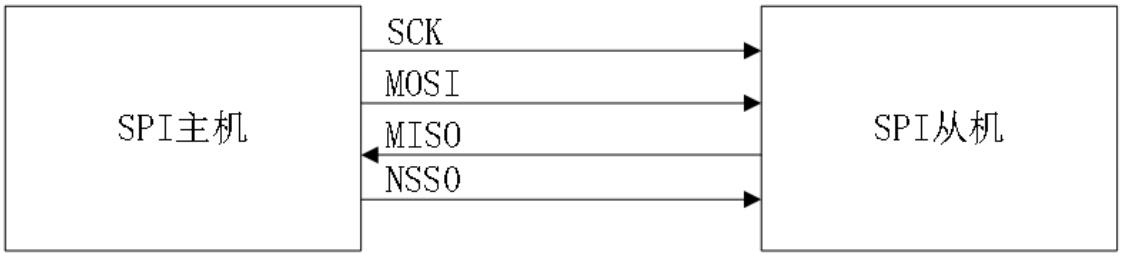


图 23.3 单主单从结构图

主机通过不同的片选选择通信的从机。一个主机和 4 个从机的结构图如下图所示：

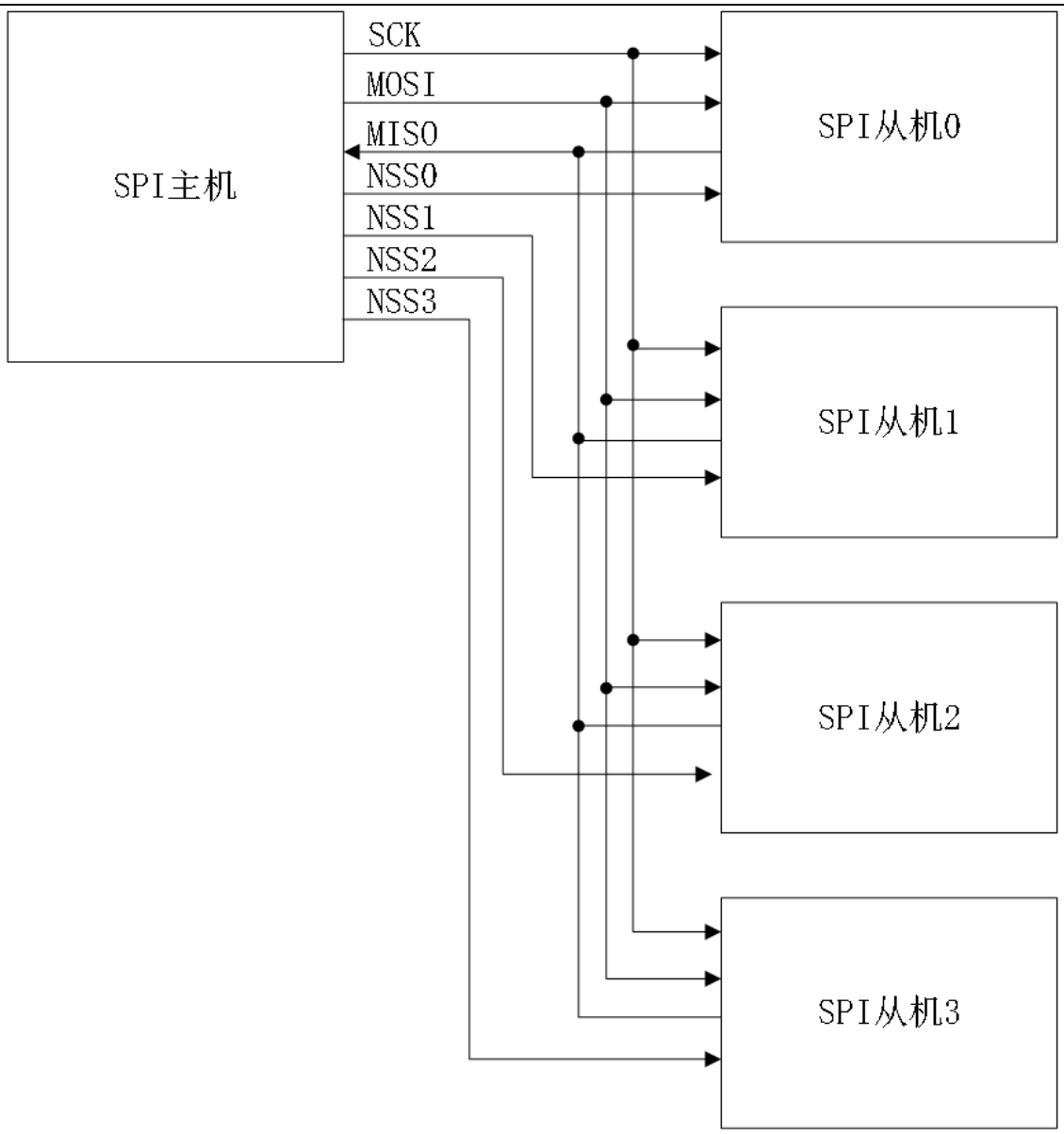


图 23.4 单主多从结构图

23.3.3 从机模式

配置为从机模式时，接收 FIFO 中非空或者溢出时，产生接收 FIFO 非空或溢出中断，通知 CPU 进行后续操作。

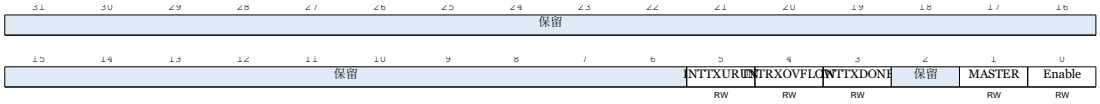
SCK 引脚用于接收到从主设备来的串行时钟。此时，时钟分频配置寄存器的设置不影响数据传输速率。

23.4 寄存器

23.4.1 SPI 控制寄存器 0（SPI_CTRL0）

地址偏移：0x0

复位值：0x00000000

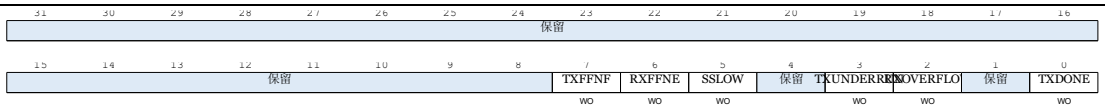


位/位域	名称	描述
31:6	保留	必须保持复位值
5	INTTXURUN	发送缓冲区空中断使能 0:禁用 1:使能
4	INTRXOVFLOW	接收缓冲区溢出中断使能 0:禁用 1:使能
3	INTTXDONE	发送完成中断使能 0:禁用 1:使能
2	保留	必须保持复位值
1	MASTER	主从模式选择 0:从机模式 1:主机模式
0	Enable	使能 SPI 0:禁用，此时 SPI 不会响应外部信号 1:使能

23.4.2 SPI 中断清除寄存器（SPI_INTCLEAR）

地址偏移：0x4

复位值：0x00000000

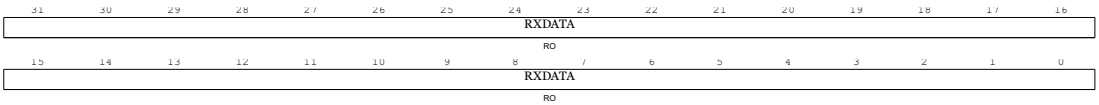


位/位域	名称	描述
31:8	保留	必须保持复位值
7	TXFFNF	写 1 清除 TXFFNF 中断
6	RXFFNE	写 1 清除 DATA_RX 中断
5	SSLOW	写 1 清除 SSEND 中断
4	保留	必须保持复位值
3	TXUNDERRUN	写 1 清除 TXUNDERRUN 中断
2	RXOVERFLOW	写 1 清除 RX 溢出中断
1	保留	必须保持复位值
0	TXDONE	写 1 清除 TXDONE 中断

23.4.3 SPI 接收数据寄存器（SPI_RXDATA）

地址偏移：0x8

复位值：0x00000000

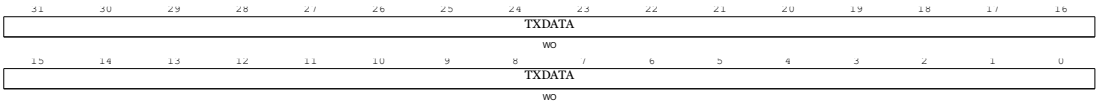


位/位域	名称	描述
31:0	RXDATA	接收数据寄存器， 从该寄存器读取数据 SPI 会从接收缓冲区中读取一帧数据

23.4.4 SPI 发送数据寄存器（SPI_TXDATA）

地址偏移：0xC

复位值：0x00000000

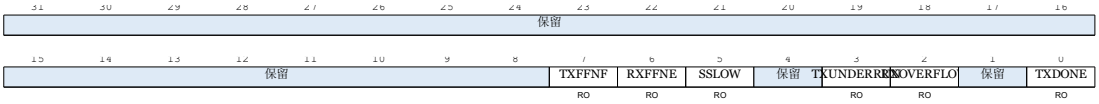


位/位域	名称	描述
31:0	TXDATA	发送数据寄存器， 写入该寄存器会将一帧数据写入发送缓冲区

23.4.5 SPI 中断状态寄存器（SPI_INT_SR）

地址偏移：0x10

复位值：0x00000000

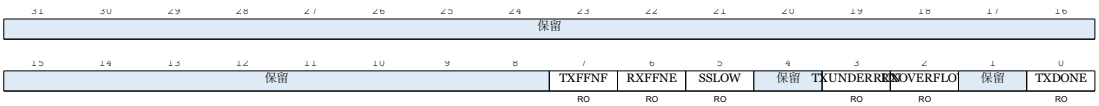


位/位域	名称	描述
31:8	保留	必须保持复位值
7	TXFFNF	发送 FIFO 非满
6	RXFFNE	接收 FIFO 非空
5	SSLOW	片选信号为低中断
4	保留	必须保持复位值
3	TXUNDERRUN	表示在从机模式下，发送缓冲区中需要的数据不可用
2	RXOVERFLOW	接收缓冲区数据溢出
1	保留	必须保持复位值
0	TXDONE	表示所有帧，包括最后一帧都已完成发送

23.4.6 SPI 中断原始状态寄存器（SPI_INT_RAW）

地址偏移：0x14

复位值：0x00000080



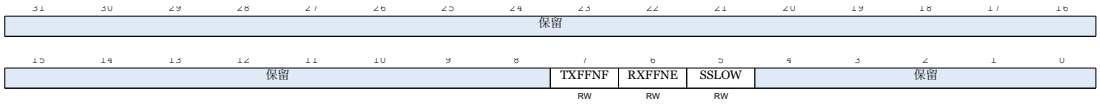
位/位域	名称	描述
31:8	保留	必须保持复位值

位/位域	名称	描述
7	TXFFNF	发送 FIFO 非满
6	RXFFNE	接收 FIFO 非空
5	SSLOW	片选信号为低中断
4	保留	必须保持复位值
3	TXUNDERRUN	表示在从机模式下，发送缓冲区中需要的数据不可用
2	RXOVERFLOW	接收缓冲区数据溢出
1	保留	必须保持复位值
0	TXDONE	表示所有帧，包括最后一帧都已完成发送

23.4.7 SPI 控制寄存器 1（SPI_CTRL1）

地址偏移：0x18

复位值：0x00000000



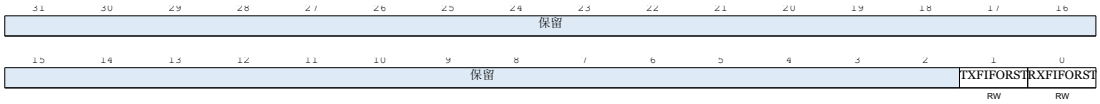
位/位域	名称	描述
31:8	保留	必须保持复位值
7	TXFFNF	发送 FIFO 非满中断使能 0:禁用 1:使能
6	RXFFNE	接收 FIFO 非空中断使能 0:禁用 1:使能
5	SSLOW	片选高电平中断使能 0:禁用 1:使能

位/位域	名称	描述
4:0	保留	必须保持复位值

23.4.8 SPI 缓冲区复位控制寄存器（SPI_COMMAND）

地址偏移：0x1C

复位值：0x00000000

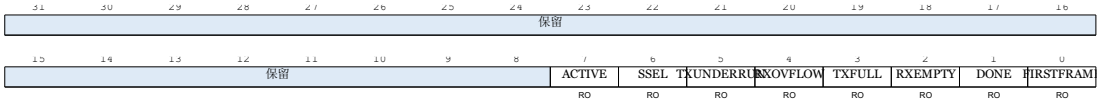


位/位域	名称	描述
31:2	保留	必须保持复位值
1	TXFIFORST	复位发送缓冲区使能 0:禁用 1:使能
0	RXFIFORST	复位接收缓冲区使能 0:禁用 1:使能

23.4.9 SPI 状态寄存器（SPI_STAT）

地址偏移：0x20

复位值：0x00000044



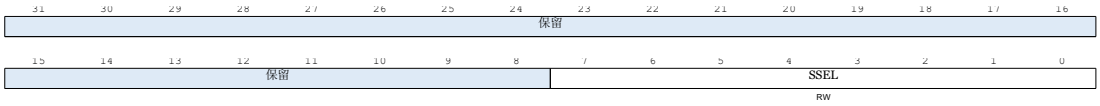
位/位域	名称	描述
31:8	保留	必须保持复位值
7	ACTIVE	总线工作状态, 0:空闲, 1:忙,此时正在发送或者接收数据
6	SSEL	片选信号的状态 0:使能

位/位域	名称	描述
		1:无效
5	TXUNDERRUN	发送缓冲区为空标志 0:无效 1:有效
4	RXOVFLOW	接收缓冲区溢出标志 0:无效 1:有效
3	TXFULL	发送缓冲区已满标志，即没有空间容纳更多新的数据 0:无效 1:有效
2	RXEMPTY	接收缓冲区为空，即没有数据可供读取
1	DONE	工作结束标志 0:无效 1:有效,此时片选输出拉高
0	FIRSTFRAME	片选有效后，接收缓冲区中的下一帧被首次接收

23.4.10 SPI 从机选择寄存器（SPI_SSEL）

地址偏移：0x24

复位值：0x00000000



位/位域	名称	描述
31:4	保留	必须保持复位值
3:0	SSEL	从机片选， 指定所选的从机。默认值为 0（未选择任何从机）。向每个位写入 1 以选择一个或多个从机。从机选择输出低电平有效。

位/位域	名称	描述
		0:禁用 1:使能

23.4.11 SPI 发送结束数据寄存器（SPI_TXDATA_LAST）

地址偏移：0x28

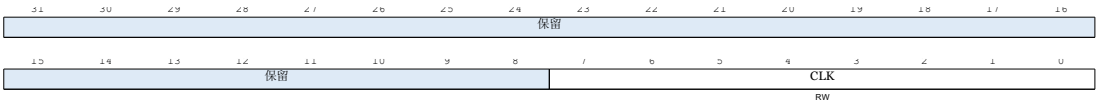
复位值：0x00000000

位/位域	名称	描述
31:0	TXDATA_LAST	发送数据寄存器

23.4.12 SPI 时钟分频配置寄存器（SPI_CFG_CLK）

地址偏移：0x2C

复位值：0x00000007

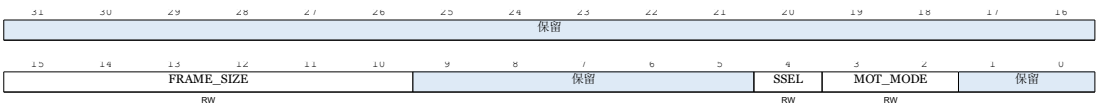


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CLK	SPI 时钟分频 生成 SPI 主时钟的时钟速率由公式确定: SPICLK=PCLK/（2*（CLK+1））

23.4.13 SPI 模式寄存器（SPI_MODE）

地址偏移：0x30

复位值：0x00000400



位/位域	名称	描述
31:16	保留	必须保持复位值

位/位域	名称	描述
15:10	FRAME_SIZE	SPI 帧大小，单位为位。要求>=4
9:5	保留	必须保持复位值
4	SSEL	传输模式 0:正常传输 1:背对背传输 在主机模式中，如果传输已经完成并且 SPI 数据寄存器中的新数据是有效的，该字节会立即发送，而不需要最小空闲时间
3:2	MOT_MODE	Motorola 模式选择 0:Mode0 1:Mode1 2:Mode2 3:Mode3
1:0	保留	必须保持复位值

24 高级定时器（TIM0、TIM1、TIM2、TIM5）

24.1 简介

高级定时器包含一个 32 位自动重载计数器，该计数器由可编程预分频器驱动，支持递增、递减、中心计数、编码器模式等计数方式。

高级定时器具有 6 个独立通道，可实现测量输入信号的脉冲宽度、可编程 PWM 输出、带死区插入的互补 PWM 等功能。

24.2 主要特征

- 32 位递增、递减、递增/递减自动重装载计数器
- 32 位可编程预分频器，用于对计数器时钟频率进行分频
- 多达 6 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成
  - 单脉冲模式输出
- 可编程死区的互补输出
- 使用外部信号控制定时器，可实现多个定时器互联
- 16 位重复计数器
- 断路输入，用于将定时器的输出信号置于用户可选的安全配置中
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较

24.3 功能说明

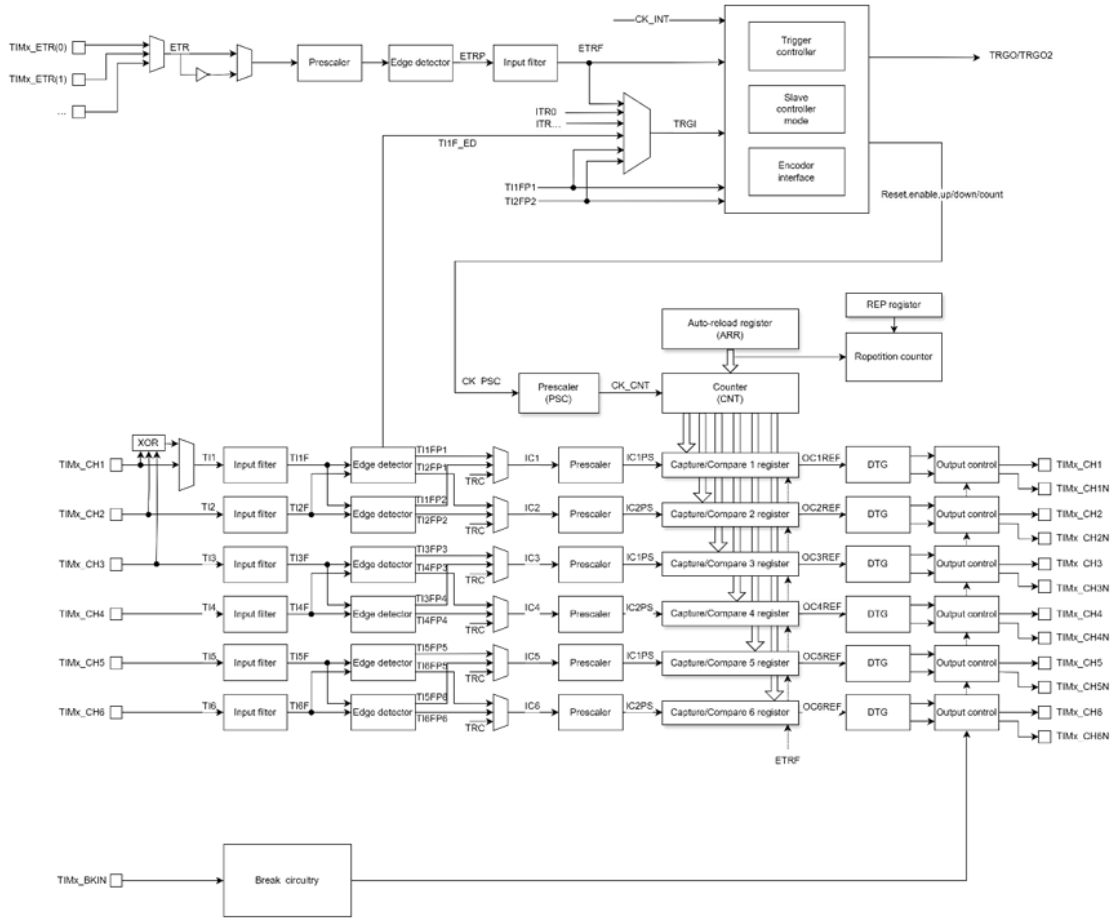


图 24.1 高级定时器结构框图

24.3.1 计数时钟

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚 TIx（x=1~6）
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入（ITRx）

24.3.1.1 内部时钟

内部时钟 CK_INT 来自定时器总线时钟。

如果禁止从模式控制器（SMS=0000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进

行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

使用内部时钟驱动计数器计数的操作步骤如下：

- 配置 TIMx_SMCR 寄存器的 SMS=0000，禁止从模式；
- 配置 TIMx_PSC 寄存器，设置计数器计数时钟频率；
- 配置 TIMx_ARR 寄存器，设置计数器计数周期；
- 配置 TIMx_CR1 寄存器的 CMS 位，以选择计数方式，若选择边沿对齐模式，则还需要配置 TIMx_CR1 寄存器的 DIR 位，以选择计数方向；
- 配置 TIMx_CR1 寄存器的 CEN 位，使能计数器。

24.3.1.2 外部时钟模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的触发信号（TRGI）出现上升沿或下降沿时计数。

使用外部时钟模式 1 驱动计数器计数的操作步骤如下：

- 配置 TIMx_SMCR 寄存器的 TS 位，选择触发信号（TRGI）的信号源；
  - TS=00100 时，选择 TI1 的边沿检测器（TI1F_ED）（即 TI1 经过滤波器后，信号的边沿检测信号）作为 TRGI；若选择该信号源，还需要配置 TIMx_CCMR1 寄存器的 IC1F 输入捕获 1 滤波器。
  - TS=00101 时，选择滤波器和边沿检测后的定时器输入 1（TI1FP1）作为 TRGI。若选择该信号源，还需要配置 TIMx_CCMR1 寄存器的 IC1F 输入捕获 1 滤波器和 TIMx_CCER 寄存器的 CC1P 和 CC1NP 配置捕获极性。
  - TS=00110 时，选择滤波器和边沿检测后的定时器输入 2（TI2FP2）作为 TRGI。若选择该信号源，还需要配置 TIMx_CCMR1 寄存器的 IC2F 输入捕获 2 滤波器和 TIMx_CCER 寄存器的 CC2P 和 CC2NP 配置捕获极性。
- 配置 TIMx_SMCR 寄存器的 SMS=0111，选择外部时钟模式 1；
- 配置 TIMx_PSC 寄存器，设置计数器计数时钟频率；

- 配置 TIMx_ARR 寄存器，设置计数器计数周期；
- 配置 TIMx_CR1 寄存器的 CMS 位，以选择计数方式，若选择边沿对齐模式，则还需要配置 TIMx_CR1 寄存器的 DIR 位，以选择计数方向；
- 配置 TIMx_CR1 寄存器的 CEN 位，使能计数器。

注：由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。

以下给出了要使递增计数器在 TI2 输入出现上升沿时计数的操作示例：

1. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=01 来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIMx_CCMR1 寄存器中写入 IC2F 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
3. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。（有关计数器的配置已省略）

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

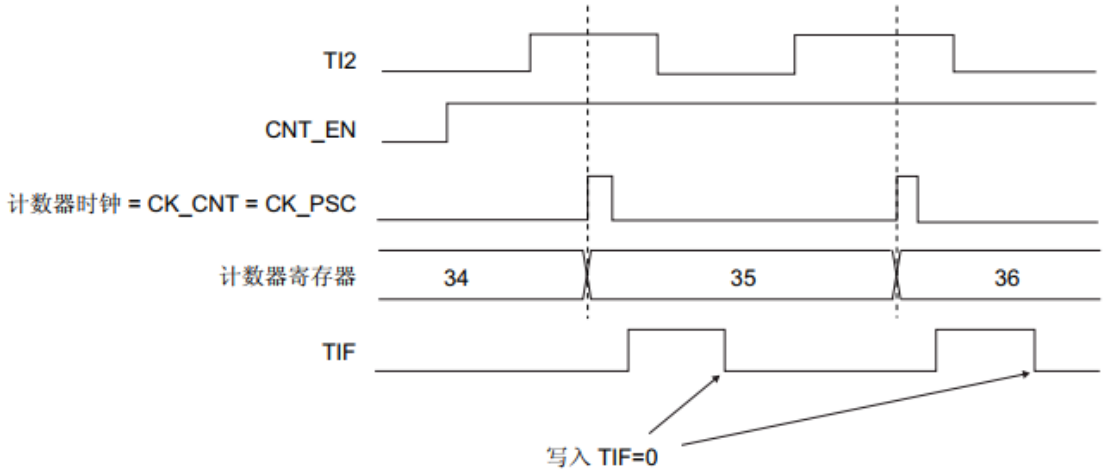


图 24.2 外部时钟模式 1 的计数器时序图

24.3.1.3 外部时钟模式 2

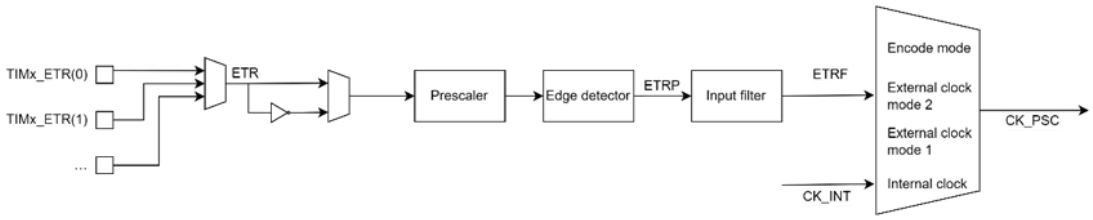


图 24.3 外部触发输入模块电路图

以下是针对外部时钟模式 2 电路的功能介绍：

（1）时钟信号输入引脚：

当使用外部时钟模式 2 时，时钟信号来自于高级定时器的特定输入通道 TIMx_ETR，ETR 的输入源可通过 TIMx_AF1 寄存器的 ETRSEL 配置。

（2）外部触发极性：

来自 ETR 引脚输入的信号可通过配置 TIMx_SMCR 寄存器的 ETP 位选择为上升沿或下降沿有效。

（3）外部触发预分频器：

当触发信号的频率过高时，可通过配置 TIMx_SMCR 寄存器的 ETPS 位进行预分频。

（4）外部触发滤波器：

若 ETRP 的信号频率过高或混杂了高频干扰信号，可通过配置 TIMx_SMCR 寄存器的 ETF 位，对 ETRP 信号进行重新采样，以达到降频或去除高频干扰的目

的。ETF 寄存器描述中的 fDTS 是由内部失踪 CK_INT 分频产生的，具体由 TIMx_CR1 寄存器的 CKD 配置。

（5）从模式选择：

经过滤波器滤波后的 ETRF 信号，可作为外部时钟模式 2 的触发信号，最终输出 CK_PSC 作为时基单元的 PSC 分频器时钟，最终驱动计数器计数。外部时钟模式 2 的选择可通过 TIMx_SMCR 寄存器的 ECE 位进行配置。

使用外部时钟模式 2 驱动计数器计数的操作步骤如下：

- 配置 TIMx_AF1 寄存器的 ETRSEL，选择 ETR 信号源；
- 配置 TIMx_SMCR 寄存器的 ETP，以选择外部触发极性；
- 配置 TIMx_SMCR 寄存器的 ETPS，以选择外部触发预分频器；
- 配置 TIMx_SMCR 寄存器的 ETF，以选择外部触发滤波器；
- 配置 TIMx_SMCR 寄存器的 ECE，以选择外部时钟模式 2；
- 配置 TIMx_CR1 寄存器的 CEN，使能计数器。

注：通过该流程选择外部时钟模式 2，等效于外部时钟模式 1 选择外部触发输入 ETRF 作为 TRGI（即 TIMx_SMCR 寄存器中的 SMS 选择外部时钟模式 1，同时 TS 选择外部触发输入（ETRF））。

以下给出了使递增计数器在 ETR 每出现 2 个上升沿时计数的操作示例：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS=01 来设置预分频器。
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。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。

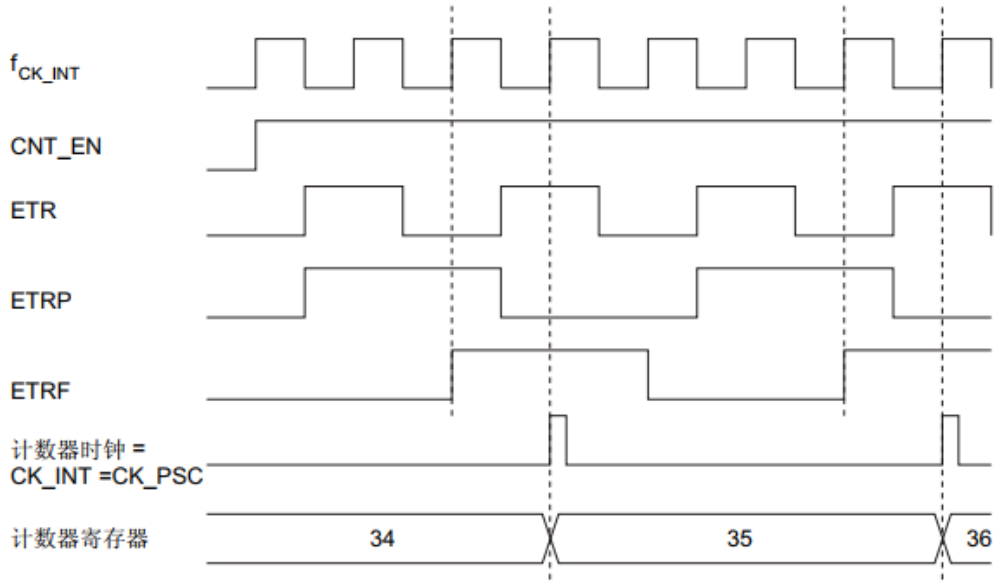


图 24.4 外部时钟模式 2 的计数器时序图

24.3.1.4 内部触发输入

定时器之间支持互联同步，因此一个定时器的时钟可由另一个定时器的输出信号 TIMx_TRGO 提供。配置 TIMx_SMCR 寄存器的 TS=00000~00011，即可选择内部触发信号启动计数器计数。

使用内部触发输入驱动计数器计数的操作步骤如下：

- 配置 TIMx_SMCR 寄存器的 TS，选择内部触发源；
- 配置 TIMx_SMCR 寄存器的 SMS=0111，选择外部时钟模式 1；
- 配置 TIMx_CR1 寄存器的 CEN，使能计数器。

24.3.1.5 触发源说明

高级定时器的外部触发源和内部触发源连接详见下表：

表 24.1 高级定时器触发源列表

TIMx	外部触发源 0 ETR(0)	外部触发源 1 ETR(1)	内部触发源 0 ITR(0)	内部触发源 1 ITR(1)
TIM0	ADC0.AMO	ADC1.AMO	TIM1.TRGO	—
TIM1	ADC2.AMO	—	TIM0.TRGO	—
TIM2	—	—	TIM4.TRGO	TIM3.TRGO
TIM5	—	—	TIM7.TRGO	TIM6.TRGO

24.3.2 时基单元

时基单元是定时器的核心单元，其包括一个 32 位的计数器、一个 32 位的预分频器、一个 32 位的自动重载计数器和一个 16 位的重复计数器。计数器可根据设置进行递增计数、递减计数或中心计数（递增递减计数交替进行）。可编程预分频器模块为计数器提供计数时钟，实现计数频率控制。自动重载寄存器为计数器提供重载值。重复计数器可控制更新事件的产生时间。

24.3.2.1 预分频器

预分频器对时钟源进行分频，分频后的时钟作为计数器时钟。预分频系数由预分频寄存器（TIMx_PSC）设定。由于预分频寄存器具有缓冲功能（存在影子寄存器），因此预分频器可以实现实时更改，新的预分频比可以随时写入预分频寄存器中，只有在下一个更新事件 UEV 发生时才会被采用（发生 UEV 时写入影子寄存器中）。

预分频系数实时发生变化时，计数器的时序图如下。

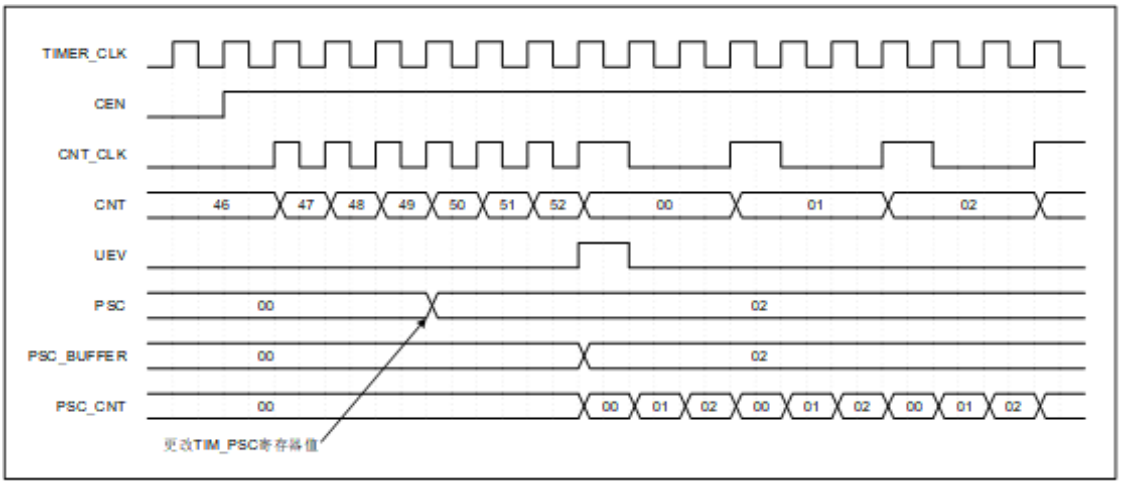


图 24.5 预分频系数由 1 变为 3 时的计数器时序图

24.3.2.2 自动重载寄存器

自动重载寄存器用于设置计数器计数的重载值，其自动重载功能由 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE）控制。预装载使能（ARPE=1）时，自动重载值将被缓存，并在下一个更新事件 UEV 发生时被采用（发生 UEV 时写入影子寄存器中）；预装载未使能（ARPE=0）时，自动重载值会被立即采用（立即写入影子寄存器中）。

24.3.2.3 计数器

24.3.2.3.1 递增计数模式

在递增计数模式下，计数器从 0 开始时递增计数到自动重载值（TIM_ARR 寄存器数值），一旦计数器计数到自动重载值，则会重新从 0 开始计数并产生计数器上溢事件。

下图分别为预分频系数为 1、2 时，递增计数模式下计数器时序图。

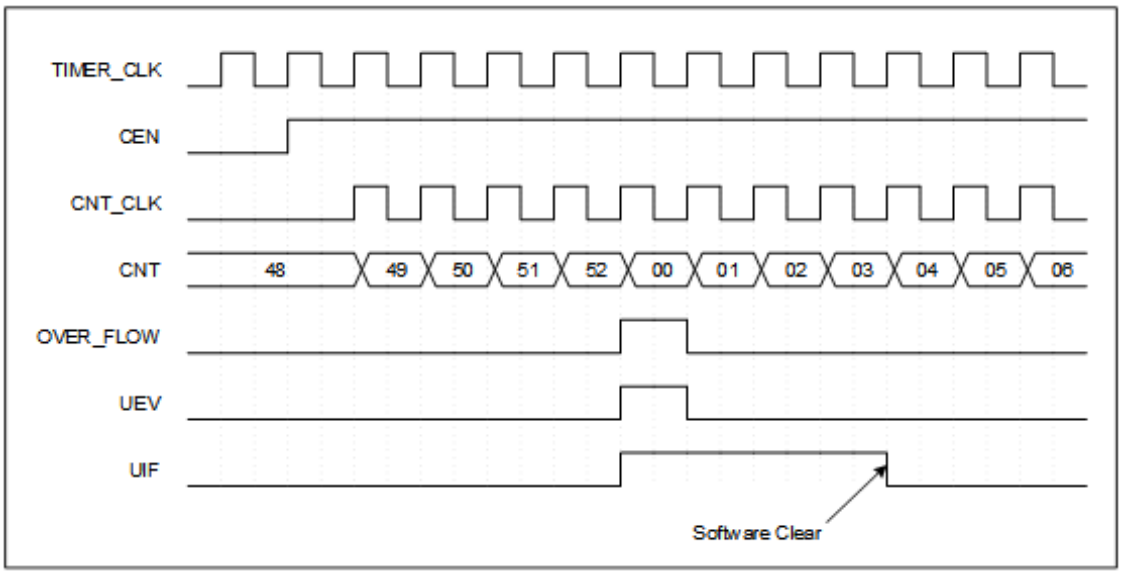


图 24.6 递增计数时序图（1 分频）

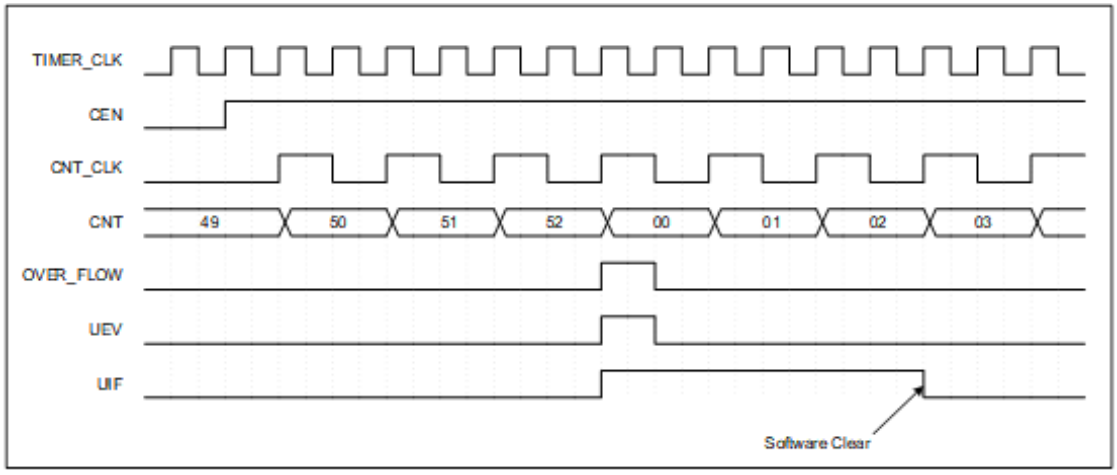


图 24.7 递增计数时序图（2 分频）

预装载未使能（ARPE=0）时，计数器在递增计数过程中，改变 TIM_ARR 寄存器值的时序图如下图所示。

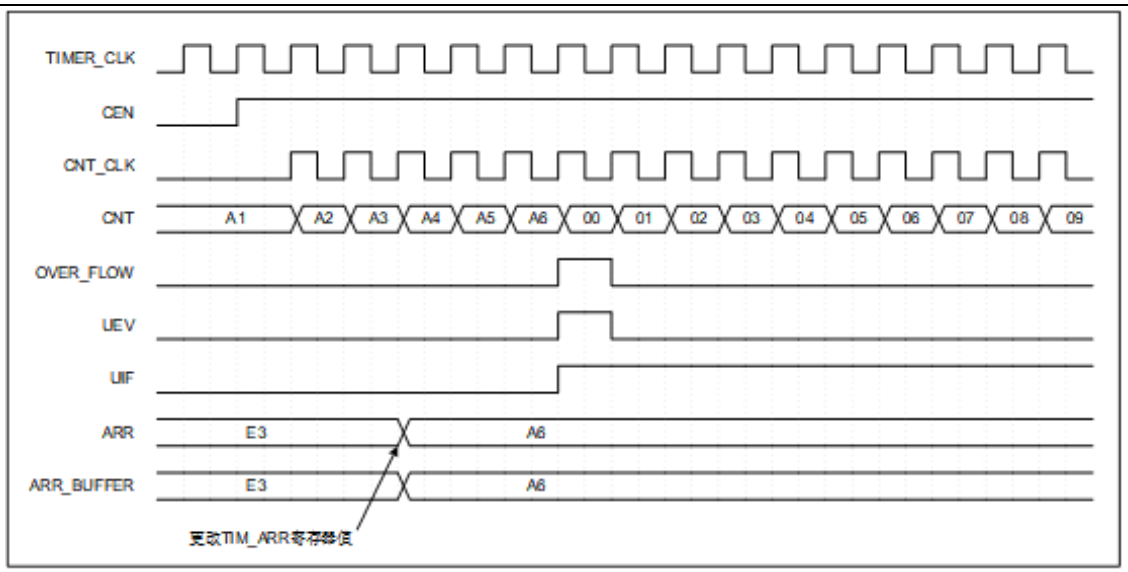


图 24.8 计数器时序图（ARPE=0 时，递增计数时改变 TIM_ARR 寄存器值）

预装载使能（ARPE=1）时，计数器在递增计数过程中，改变 TIM_ARR 寄存器值的时序图如下图所示。

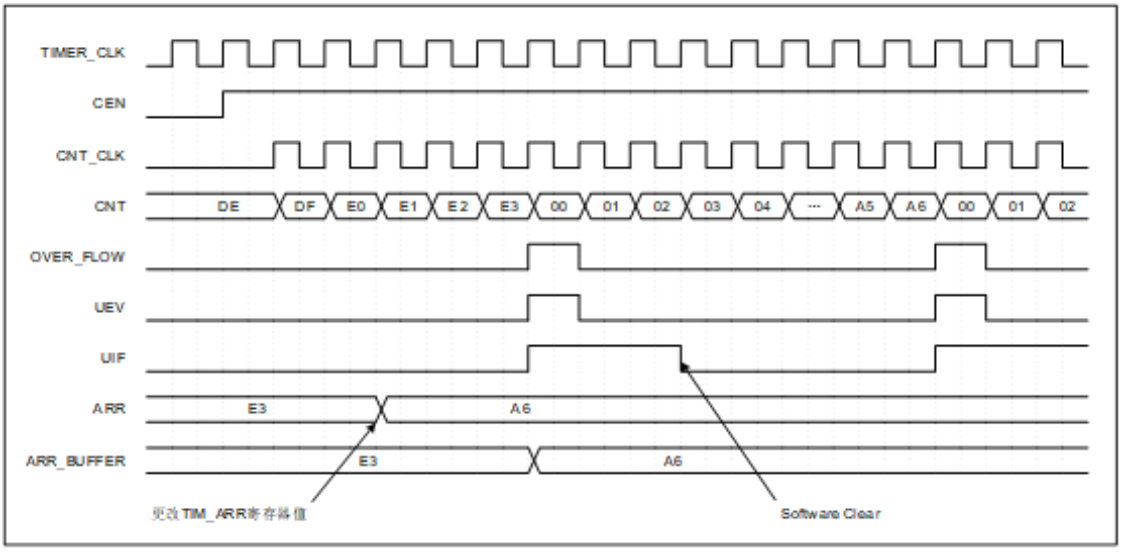


图 24.9 计数器时序图（ARPE=1 时，递增计数时改变 TIM_ARR 寄存器值）

24.3.2.3.2 递减计数模式

在递减计数模式下，计数器从自动重载值（TIM_ARR 寄存器数值）开始时递减计数到 0，一旦计数器计数到 0，则会重新从自动重载值开始计数并产生计数器下溢事件。

下图分别为预分频系数为 1、2 时，递减计数模式下计数器时序图。

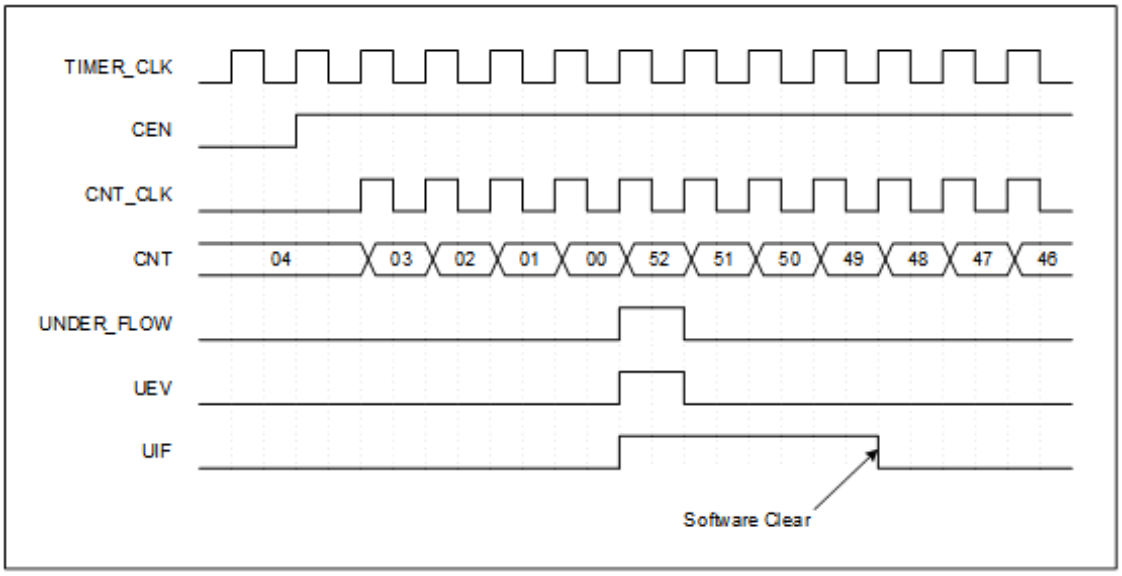


图 24.10 递减计数时序图（1 分频）

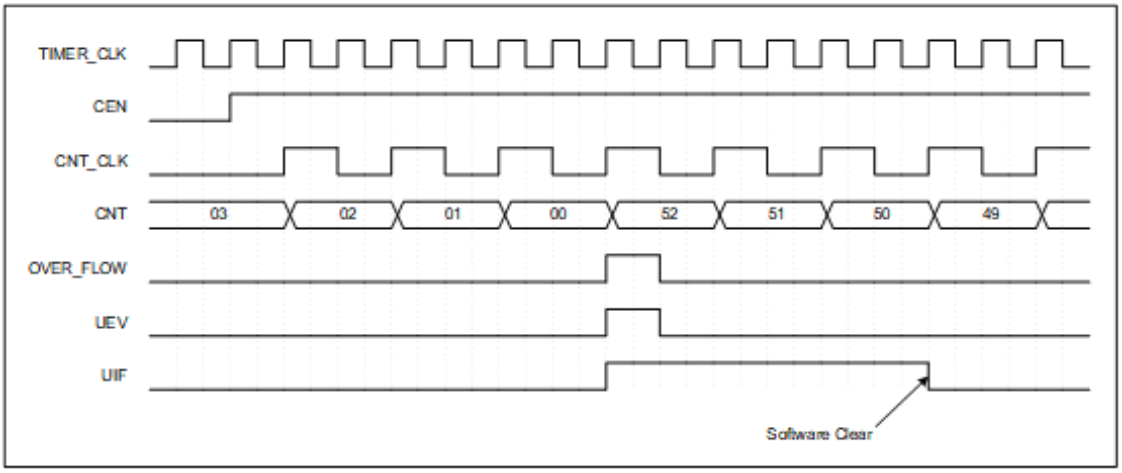


图 24.11 递减计数时序图（2 分频）

预装载未使能（ARPE=0）时，计数器在递减计数过程中，改变 TIM_ARR 寄存器值的时序图如下图所示。

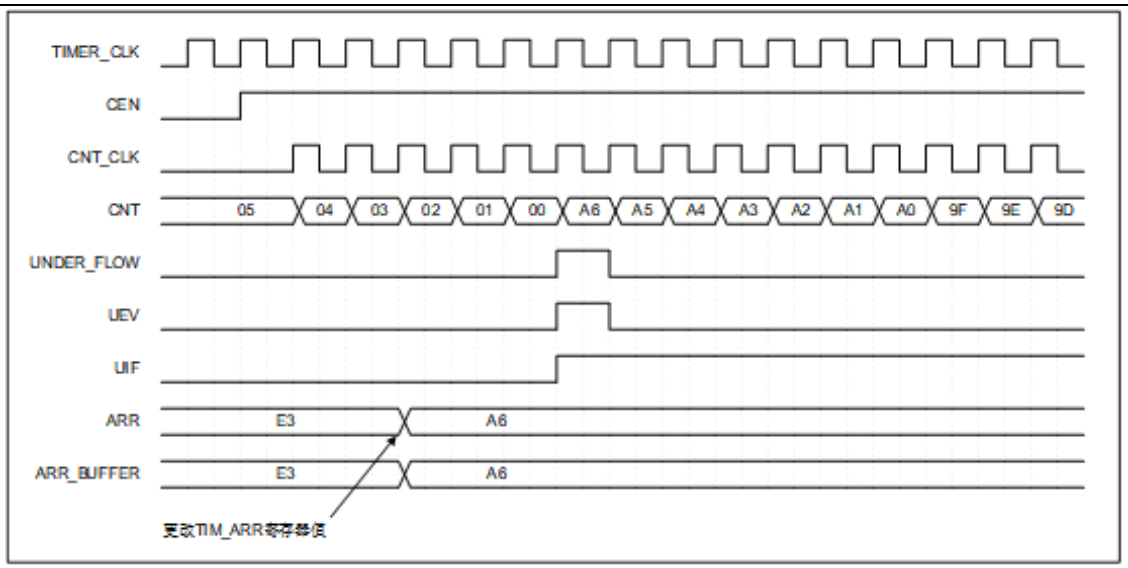


图 24.12 计数器时序图（ARPE=0 时，递减计数时改变 TIM_ARR 寄存器值）

预装载使能（ARPE=1）时，计数器在递减计数过程中，改变 TIM_ARR 寄存器值的时序图如图下图所示。

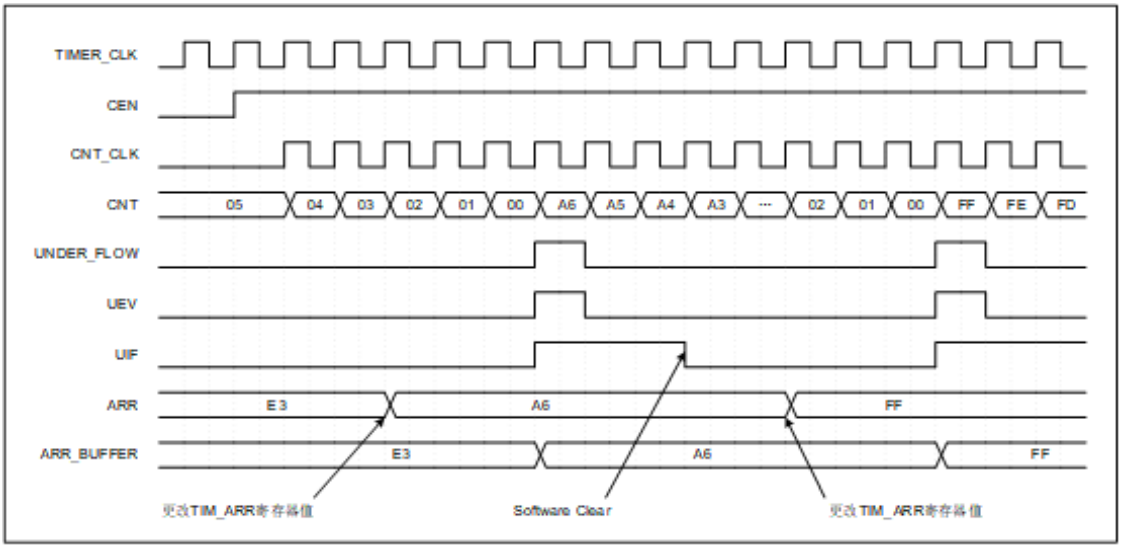


图 24.13 计数器时序图（ARPE=1 时，递减计数时改变 TIM_ARR 寄存器值）

24.3.2.3.3 递增/递减计数模式（中心对齐模式）：

在中心对齐模式下，计数器交替进行递增、递减计数，从 0 开始递增计数到自动重载值，然后再递减计数到 0。在递增计数中，计数器计数到自动重载值-1（TIM_ARR-1）时，会产生一个上溢事件；在递减计数中，计数器计数到 1 时，会产生一个下溢事件。在中心对齐模式下，TIM_CR1 寄存器的方向位（DIR）是只读的，计数方向由硬件控制。

下图为在中心对齐模式下的计数器的时序图。

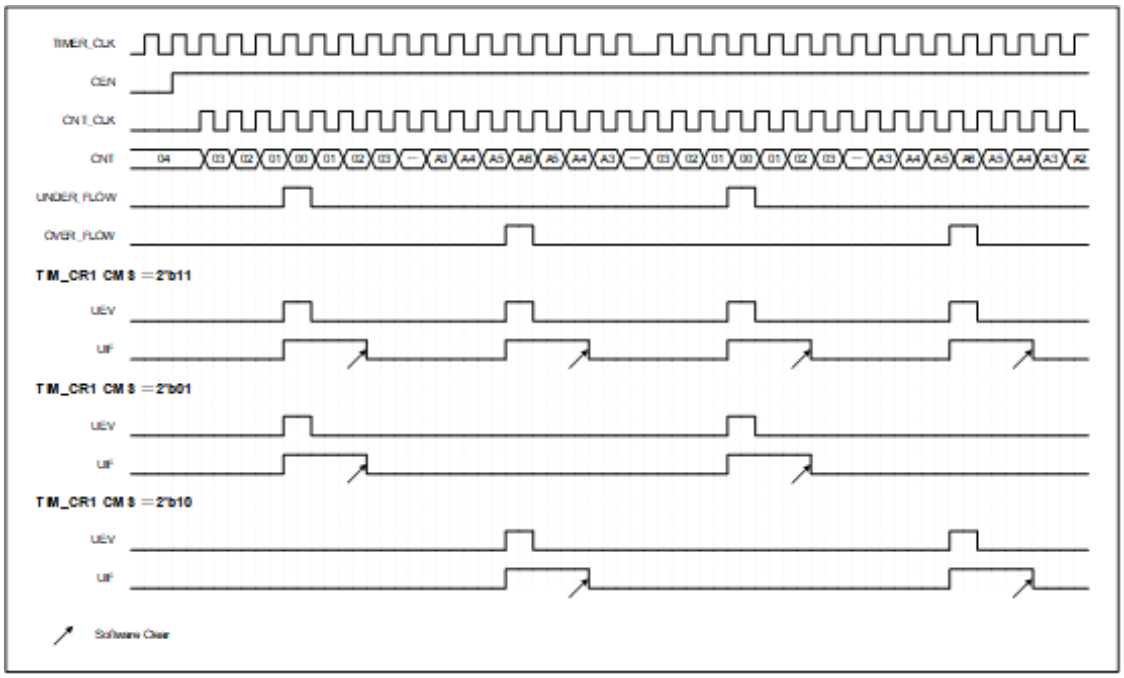


图 24.14 中心对齐模式，计数器时序图

24.3.2.3.4 重复计数模式

TIMx_RCR 寄存器用于配置重复计数器计数周期，当 TIMx_RCR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每（REP+1）次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器值等于 0 时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

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。
- 中心对齐模式下的每个计数器上溢和计数器下溢。

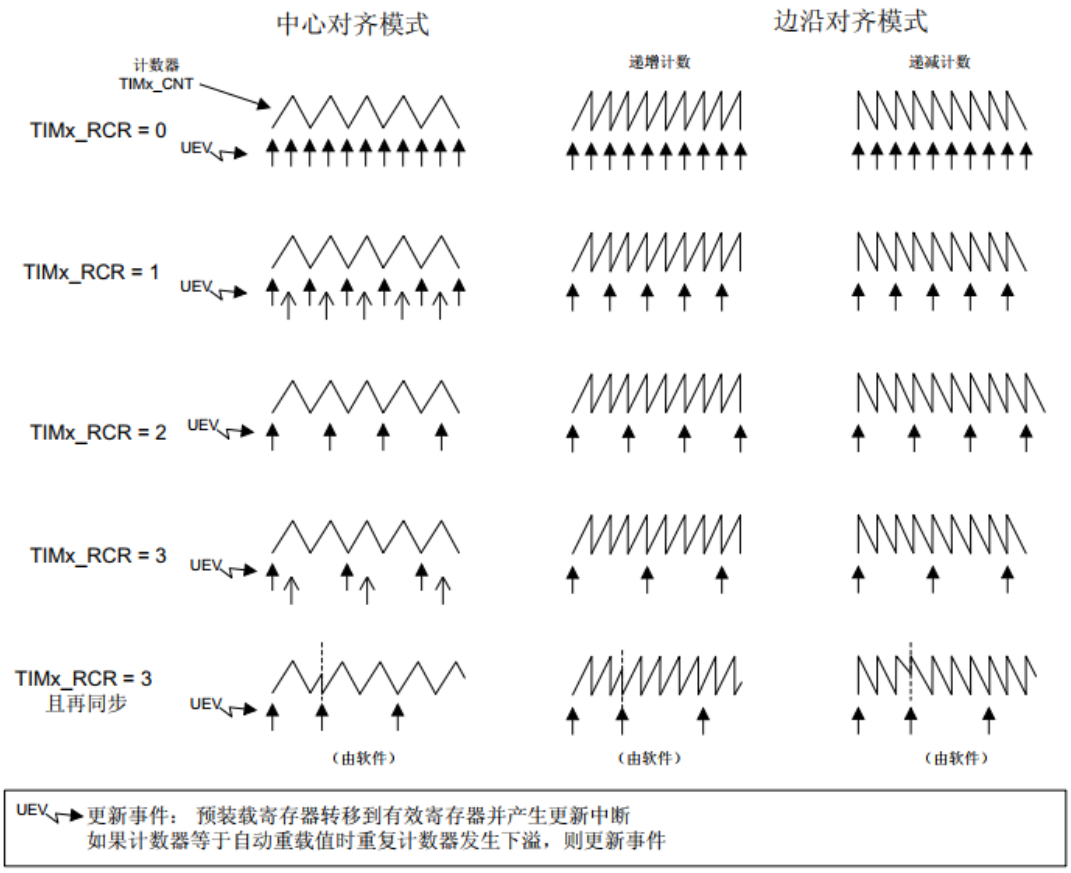


图 24.15 不同模式、不同 TIMx_RCR 寄存器配置下的更新频率示意图

24.3.2.3.5 编码器模式

编码器模式下需提供两组输入信号 TIMx_CH1 和 TIMx_CH2，根据一组输入信号电平值，计数器在另一组输入信号边沿向上或向下计数。计数方向由 DIR 值指示。

- 编码器模式 1：SMS=0001,计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
- 编码器模式 2：SMS=0010,计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
- 编码器模式 3：SMS=0011,计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。

若要使用编码器模式可按下面步骤配置：

- 配置 TIMx_CC1M 寄存器的 IC1F 位，设置通道 1 输入信号滤波；配置 TIMx_CCER 寄存器的 CC1P 位，设置通道 1 输入信号有效电平。
- 配置 TIMx_CC1M 寄存器的 IC2F 位，设置通道 2 输入信号滤波；配置 TIMx_CCER 寄存器的 CC2P 位，设置通道 2 输入信号有效电平。
- 配置 TIMx_CC1M 寄存器的 CC1S 位，设置通道 1 为输入模式；配置 TIMx_CC1M 寄存器的 CC2S 位，设置通道 2 为输入模式；
- 配置 TIMx_SMCR 寄存器的 SMS 位，选择编码器模式 1、编码器模式 2 或编码器模式 3;
- 配置 TIMx_ARR 寄存器的 ARR 位，设置计数器自动重装载值；
- 配置 TIMx_CR1 寄存器的 CEN 位，使能计数器。

编码模式下计数器计数方向如下表所示：

表 24.2 编码器模式下信号与计数器方向关系表

有效边沿	计数边沿相对信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

下图以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=01（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。

- CC1P=0，CC1NP=0（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=0，CC2NP=0（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2=TI2）。
- SMS=0011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=1（TIMx_CR1 寄存器，使能计数器）。

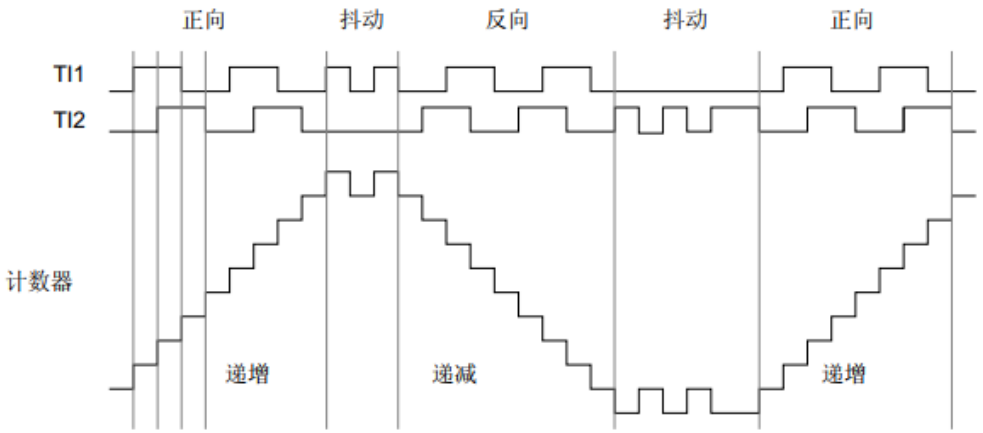


图 24.16 编码器模式下计数器工作示意图

24.3.2.3.6 更新事件 UEV 产生

更新事件 UEV 是否产生由更新禁止 UDIS 位控制（TIM_CR1 寄存器）。UDIS=0 时，使能 UEV，允许产生更新事件 UEV；UDIS=1 时，禁止 UEV，不会产生更新事件 UEV。

使能时，发生计数器上溢或下溢、生成更新 UG=1（TIM_EGR 寄存器）时，将会产生更新事件 UEV，并且更新影子寄存器的值（自动重载影子寄存器和预分频影子寄存器）。

禁止时，不会产生更新事件，影子寄存器保持不变。只有在 UG 置 1 时，初始化计数器和预分频器。

24.3.2.3.7 更新中断 UIF 产生

更新中断 UIF 的事件源由更新请求源 URS 位控制（TIM_CR1 寄存器）。URS=0 时，发生计数器上溢或下溢、UG 位置 1 时都会产生更新中断 UIF；URS=0 时，只有发生计数器上溢或下溢才会产生更新中断 UIF。

24.3.3 输入捕获

高级定时器拥有 6 个独立通道，每个通道可配置为输入捕获或输出比较模式。下图展示了通道输入部分的主要电路。

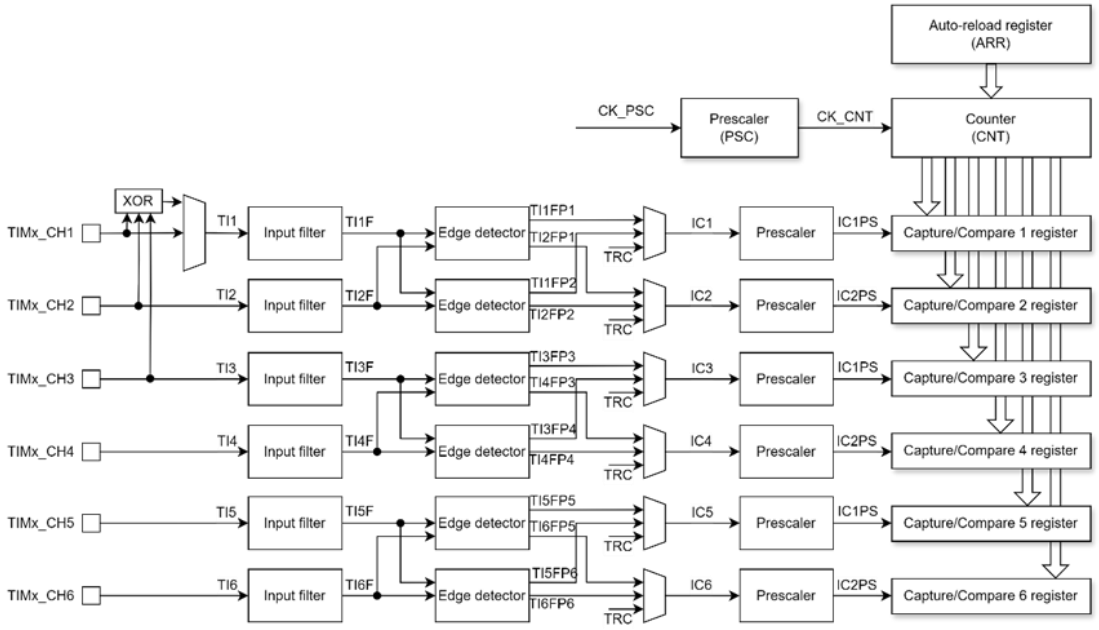


图 24.17 输入捕获电路图

（1）输入通道：

待测信号从定时器的外部引脚 TIMx_CH1-TIMx_CH6 进入，经过预处理输出 TIx。 其中，TI1 可通过 TIMx_CR2 的 TI1S 选择由 TIMx_CH1 输入或选择 TIMx_CH1、TIMx_CH2、TIMx_CH3 异或。TI2-TI6 则分别连接 TIMx_CH2~TIMx_CH6。

（2）输入滤波器：

TIx 信号经过数字滤波器，输出滤波后的信号 TIxF。 输入滤波器内置两级同步器，输入信号经过同步器被定时器时钟信号同步后，输入至数字滤波器。当输入的信号存在高频干扰的时候，可通过配置 TIMx_CR1 寄存器的 CKD 位和

TIMx_CCMR1/2/3 寄存器的 ICxF 位，对输入信号 TIx 进行滤波处理，即进行重新采样，根据采样定律，采样的频率必须大于等于两倍的输入信号。 通过 ICxF 位介绍可知，采样频率 fSAMPLE 可由 fCK_INT 和 fDTS 分频后的时钟提供，其中 fCK_INT 是内部时钟，即定时器总线时钟；fDTS 是由 fCK_INT 分频后产生，分频系数由 CKD 位进行配置。

（3）边沿检测器：

TIxF 信号经过边沿检测器，输出边沿选择后的信号 TIxFPx。 边沿检测器可通过配置 TIMx_CCER 寄存器的 CCxP 和 CCxNP 位，选择在输入信号的上升沿、下降沿或双边沿发生捕获事件。 以 TI1FPx 为例，TI1FP1 表示来自通道 1 的 TI1F 信号经过通道 1 的边沿检测器（即由 CC1P 和 CC1NP 位共同控制）处理后的信号；TI1FP2 表示来自通道 1 的 TI1F 信号经过通道 2 的边沿检测器（即由 CC2P 和 CC2NP 位共同控制）处理后的信号。

（4）捕获信号选择器：

TIxFPx 信号和 TRC 信号经过输入捕获信号选择器，输出选择后的信号 ICx。捕获信号选择器可通过配置 TIMx_CCMR1/2/3 寄存器的 CCxS 位，选择捕获信号。 以通道 1 为例，CC1S=01 时，IC1 选择 TI1FP1；CC1S=10 时，IC1 选择 TI2FP1；CC1S=11 时，IC1 选择 TRC。

（5）预分频器：

ICx 信号经过预分频器，输出预分频后的信号 ICxPS。 预分频器可通过配置 TIMx_CCMR1/2/3 寄存器的 ICxPSC 位，对 ICx 信号进行预分频，即选择发生多少个事件后进行一次捕获，若希望捕获信号的每一个边沿，则不分频。

（6）捕获寄存器：

输入捕获通道最终以经过预分频后的信号 ICxPS 进行捕获，当发生捕获时，计数器 CNT 的数值将会被锁存到捕获寄存器 CCRx 中。

24.3.3.1 输入模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器（TIMx_CCRx）来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发

生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入 0，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
- 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 fDTS 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。

- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注：通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。

24.3.3.2 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入 0（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=10（下降沿有效）。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。

- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入 1。

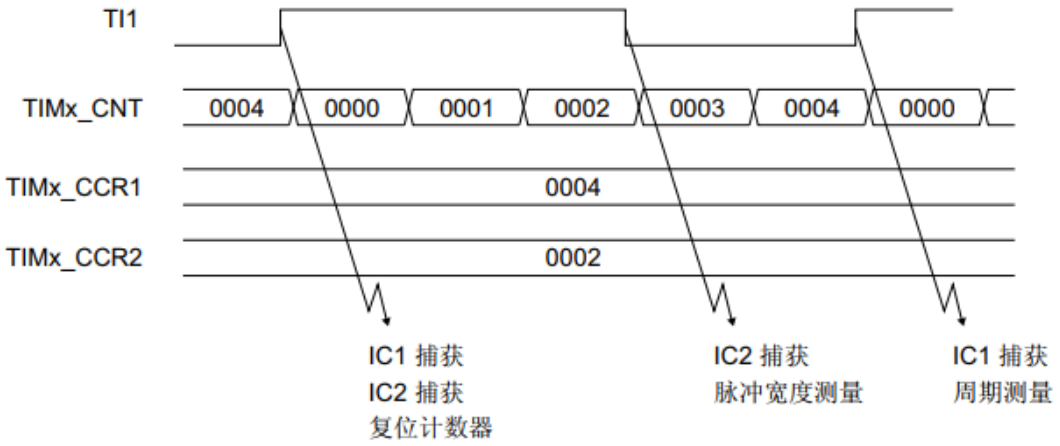


图 24.18 PWM 输入模式时序图

24.3.3.3 输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔，如下图所示。

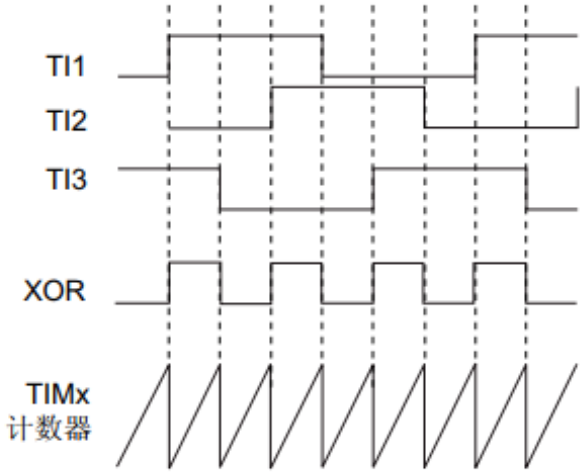


图 24.19 测量 3 输入信号上边沿的时间间隔示意图

24.3.4 输出比较

下图展示了通道输出部分的主要电路。

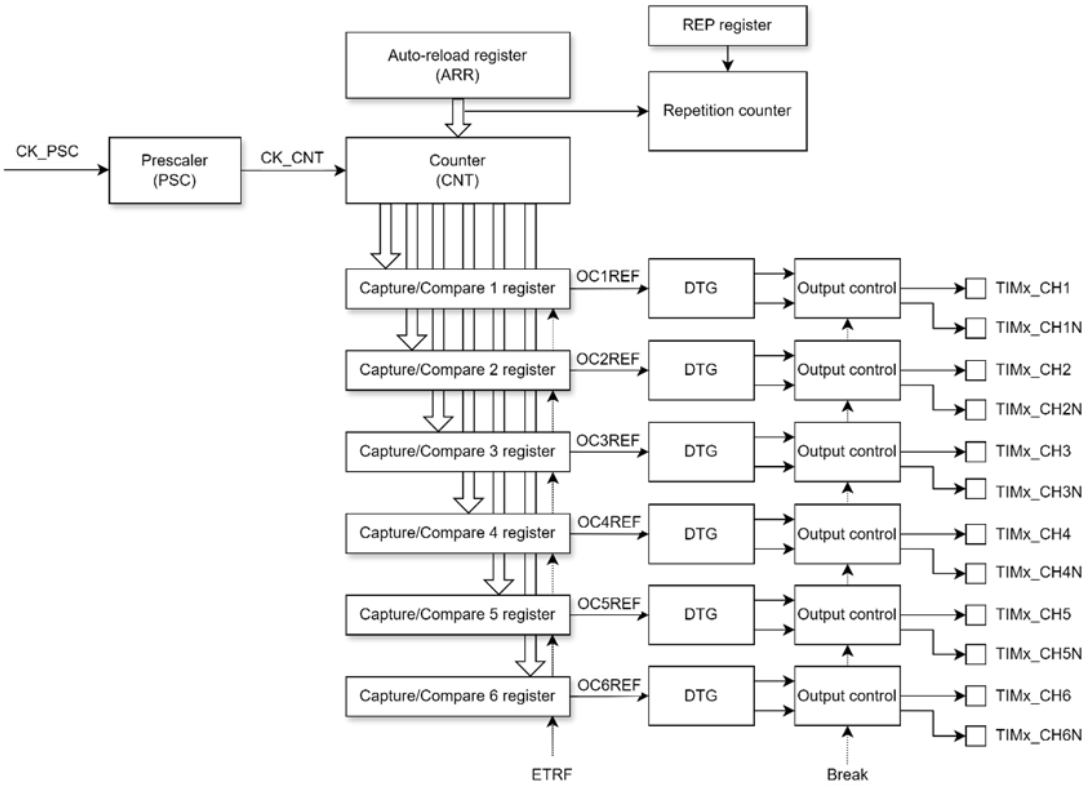


图 24.20 输出比较电路图

（1）比较寄存器

比较寄存器的主要功能是，把计数器 CNT 的数值与比较寄存器 CCRx 的值进行比较，根据 TIMx_CCMR1/2/3 寄存器的 OCxM 位的配置，输出参考信号 OCxREF，其中 OCxREF=1（高电平）称为有效电平，OCxREF=0（低电平）称为无效电平。 在使能比较中断时，比较寄存器还会产生中断信号，相应的标志位 TIMx_SR 寄存器的 CCxIF 位置 1。

（2）死区发生器

在生成的参考信号 OCxREF 的基础上，可插入死区事件，生成两路带死区的互补信号 OCx_DT 和 OCxN_DT。死区时间的大小由 TIMx_BDTR 寄存器的 DTG 位配置。

（3）输出控制器

输出控制器的主要功能是，根据配置的输出模式，控制信号的极性反转，输出信号 OCx 和 OCxN。 参考信号 OCxREF 在经过死区发生器后会产生两路带死区

的互补信号 OCx_DT 和 OCxN_DT，这两路信号将会进入输出控制器。若没有加入死区控制，那么直接将 OCxREF 信号输入至输出控制器。 进入输出控制器的信号将会根据 TIMx_CCER 寄存器的 CCxP 和 CCxNP 的配置，进行极性选择。经过极性选择的信号是否由 OCx 和 OCxN 引脚输出至外部引脚，则由 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位进行控制。 若定时器加入了断路功能，则 TIMx_BDTR 寄存器的 MOE、OSSI 和 OSSR 位也将共同影响输出的信号。

24.3.4.1 输出模式

通道输出比较模式由 CCxS 位进行控制，当 CCxS=00 时，该通道被配置为输出比较模式。

24.3.4.1.1 输出逻辑

输出比较各模式描述如下：其中输出信号 OCx 的有效电平取决于 CCxP 位。CCxP=0 时，OCx 的有效电平为高电平；CCxP=1 时，OCx 的有效电平为低电平。其中输出信号 OCxN 的有效电平取决于 CCxNP 位。CCxNP=0 时，OCxN 的有效电平为高电平；CCxNP=1 时，OCxN 的有效电平为低电平。

表 24.3 输出模式介绍表

通道状态	输出形式	OCxM[3:0]	描述
输出	冻结	0000	时基： 捕获/比较寄存器（CCRx）与计数器（CNT）进行比较不会对输出信号 OCx 造成任何影响
	比较输出	0001	匹配时设置为有效电平： 当计数器（CNT）与捕获/比较寄存器（CCRx）匹配时，输出信号 OCx 强制变为有效电平
		0010	匹配时设置为无效电平： 当计数器（CNT）与捕获/比较寄存器（CCRx）匹配时，输出信号 OCx 强制变为无效电平

通道状态	输出形式	OCxM[3:0]	描述
		0011	匹配时翻转： 当计数器（CNT）与捕获/比较寄存器（CCRx）匹配时，输出信号 OCx 发生翻转
	强制输出	0100	输出信号 OCx 强制变为有效电平
		0101	输出信号 OCx 强制变为无效电平
	PWM 输出	0110	PWM 模式 1： 在递增计数时，若 CNT＜CCRx，则输出有效电平，否则输出无效电平 在递减计数时，若 CNT＞CCRx，则输出无效电平，否则输出有效电平
		0111	PWM 模式 2： 在递增计数时，若 CNT＜CCRx，则输出无效电平，否则输出有效电平 在递减计数时，若 CNT＞CCRx，则输出有效电平，否则输出无效电平
	可再触发 OPM	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有效状态。 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：

通道状态	输出形式	OCxM[3:0]	描述
			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。
	组合 PWM	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。
	不对称 PWM	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。

通道状态	输出形式	OCxM[3:0]	描述
	单脉冲模式		单脉冲模式为一种特殊的输出比较模式。 当配置为单脉冲模式时，计数器在发生更新事件时，停止计数（CEN 位清零）

捕获/比较通道的中断触发条件如下表所示。

除中心对齐模式外，计数器 CNT 与比较值 CCRx 匹配时，CCxIF 比较中断标志置 1，并产生对应中断。若比较值 CCRx 大于自动重载值 ARR，则 CCxIF 位将在计数器发生上溢（递增计数模式和中心对齐模式）或下溢（递减计数模式）时置 1。

表 24.4 捕获/比较通道中断产生条件

CCx 通道模式	CCxIF 中断条件		
输入捕获	发生输入捕获事件		
输出比较	边沿对齐模式（CMS=00）	CCRx≤ARR	CNT=CCRx
		CCRx＞ARR	dir=0（递增计数），计数器发生上溢事件
			dir=1（递减计数），计数器发生下溢事件
	中心对齐模式 1（CMS=01）	CCRx≤ARR	dir=1（递减计数），CNT=CCRx
		CCRx＞ARR	计数器发生上溢事件
	中心对齐模式 2（CMS=10）	CCRx≤ARR	dir=0（递增计数），CNT=CCRx
		CCRx＞ARR	计数器发生上溢事件
	中心对齐模式 3（CMS=11）	CCRx≤ARR	CNT=CCRx
		CCRx＞ARR	计数器发生上溢事件

比较输出模式：

下图为三种输出比较模式（匹配时置高电平/低电平/翻转）的时序图。

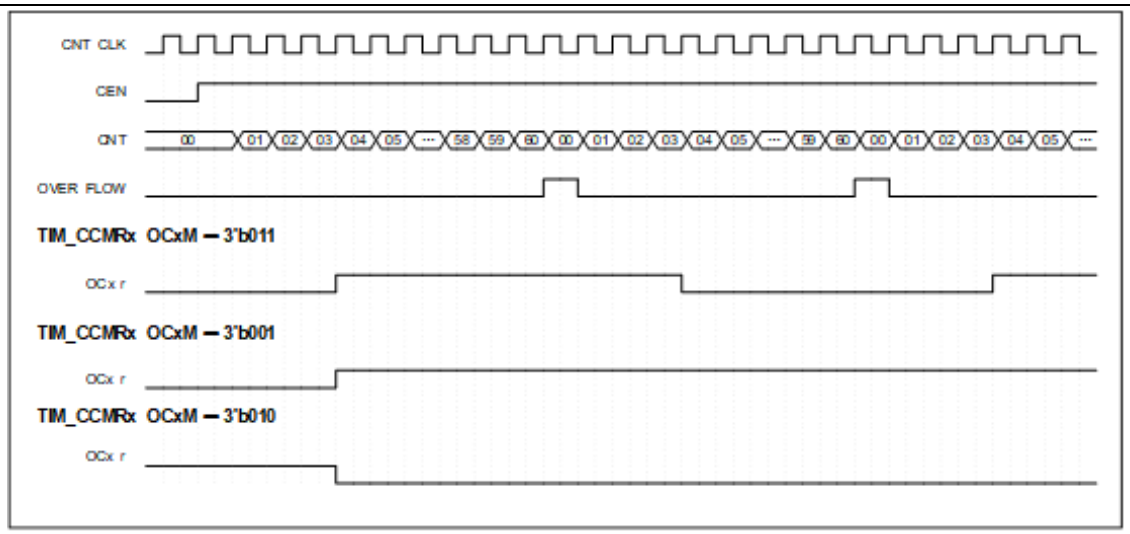


图 24.21 三种输出比较模式时序图

PWM 输出模式：

下图为递增计数模式下，输出 PWM 波形的时序图。在 PWM1 模式下，递增计数时，若 CNT＜CCRx，则输出无效电平；若 CNT≤CCRx，则输出有效电平。在 PWM2 模式下，递增计数时，若 CNT＜CCRx，则输出无效电平；若 CNT≥CCRx，则输出有效电平。

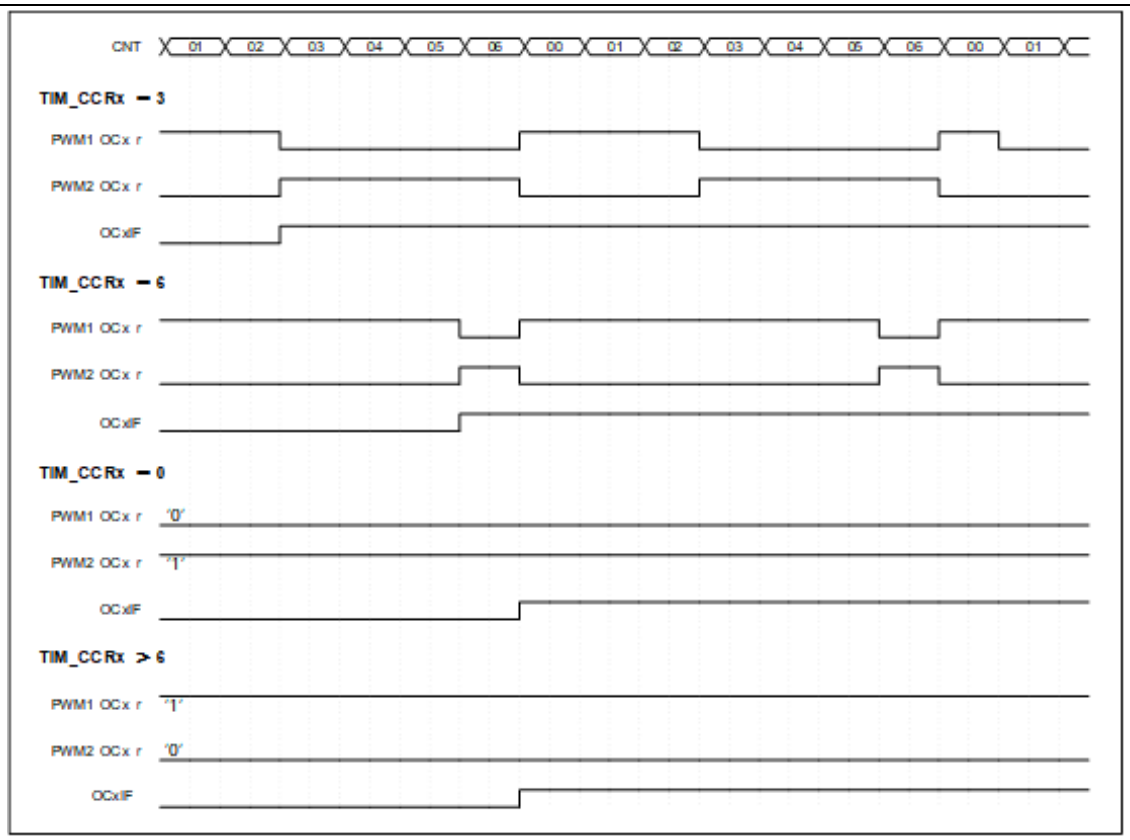


图 24.22 递增计数模式，输出 PWM 波形（ARR=06）

下图为递减计数模式下，输出 PWM 波形的时序图。在 PWM1 模式下，递减计数时，若 CNT＞CCRx，则输出无效电平；若 CNT≤CCRx，则输出有效电平。在 PWM2 模式下，递减计数时，若 CNT＞CCRx，则输出有效电平；若 CNT≤CCRx，则输出无效电平。

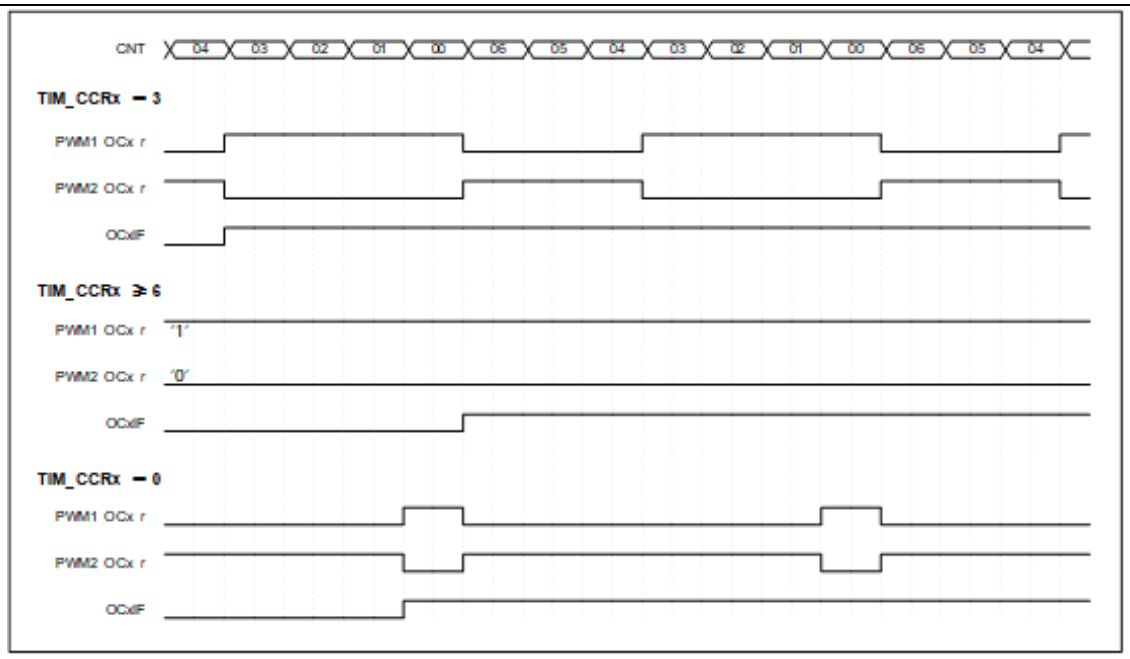


图 24.23 递减计数模式，输出 PWM 波形（ARR=06）

下图为中心对齐模式下，输出 PWM 波形的时序图。

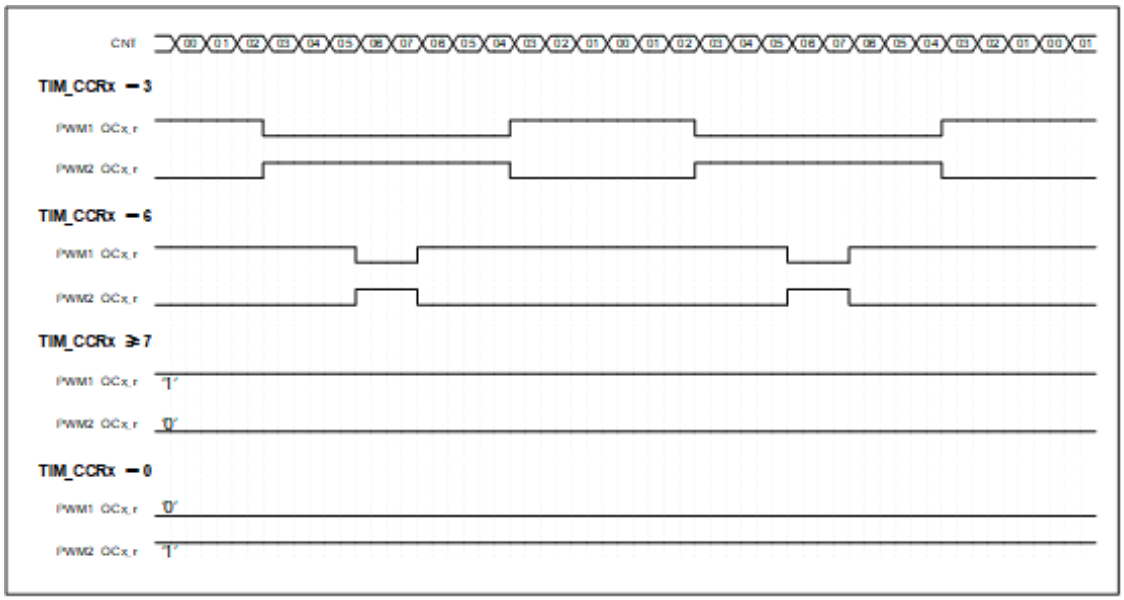


图 24.24 中心对齐模式，输出 PWM 波形（ARR=07）

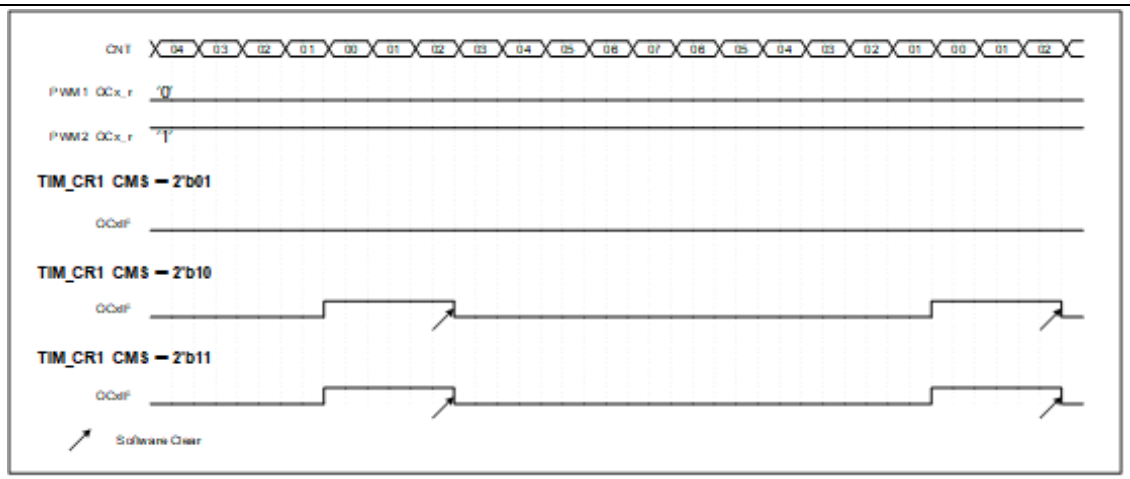


图 24.25 中心对齐模式，中断标志时序图（ARR=07、CCRx=00）

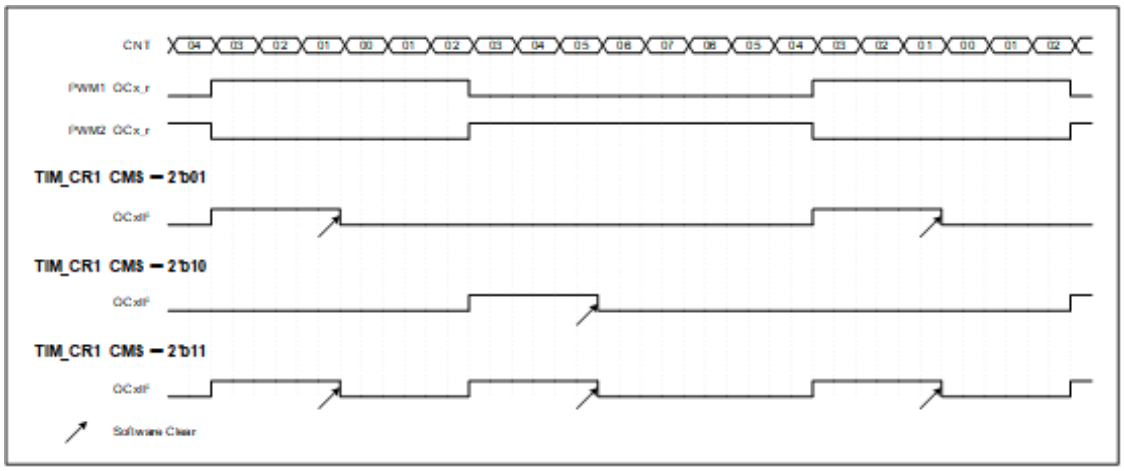


图 24.26 中心对齐模式，中断标志时序图（ARR=07、CCRx=03）

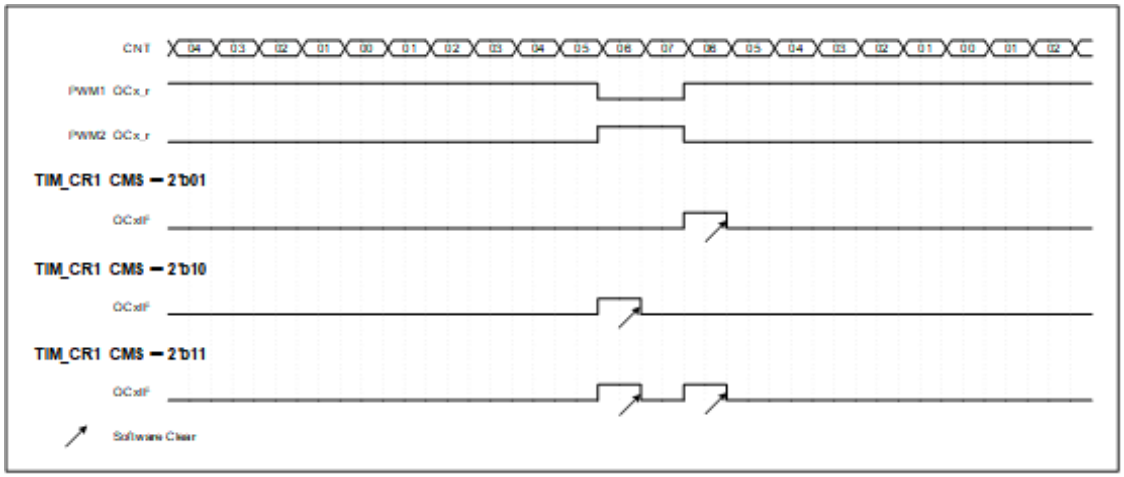


图 24.27 中心对齐模式，中断标志时序图（ARR=07、CCRx=06）

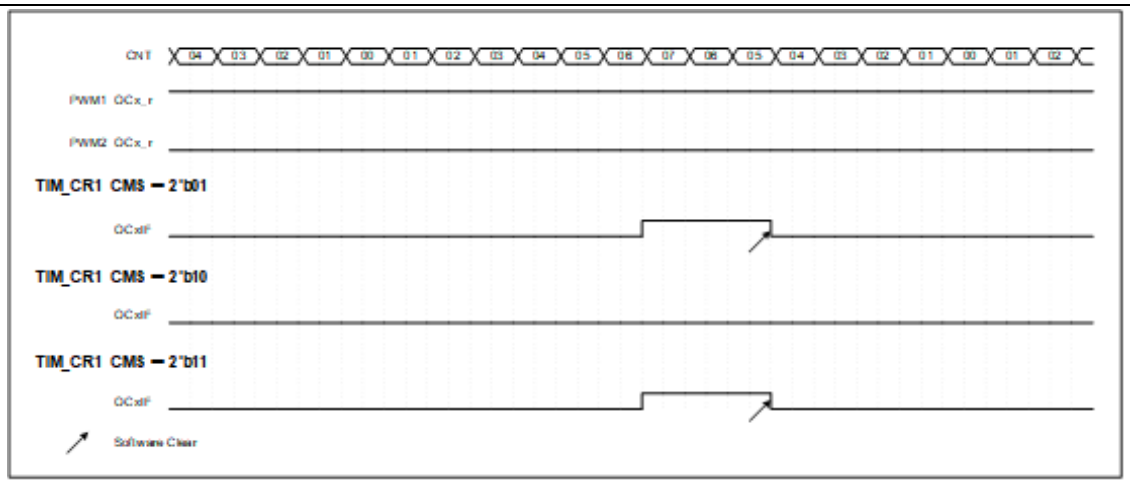


图 24.28 中心对齐模式，中断标志时序图（ARR=07、CCRx=07）

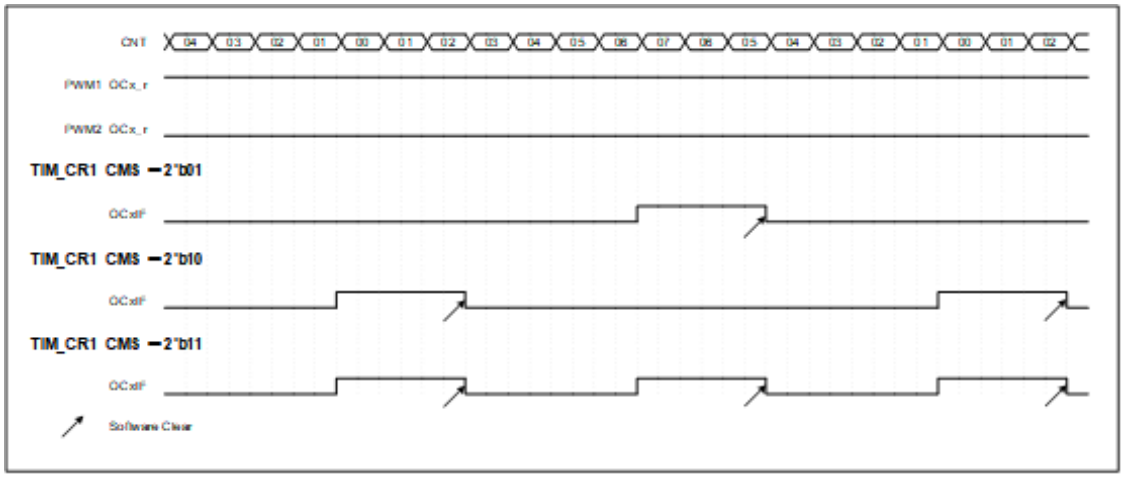


图 24.29 中心对齐模式，中断标志时序图（ARR=07、CCRx＞07）

不对称 PWM 模式：

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入 1110（不对称 PWM 模式 1）或 1111（不对称 PWM 模式 2）。

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

下图显示了不对称 PWM 模式下可以产生的信号示例。

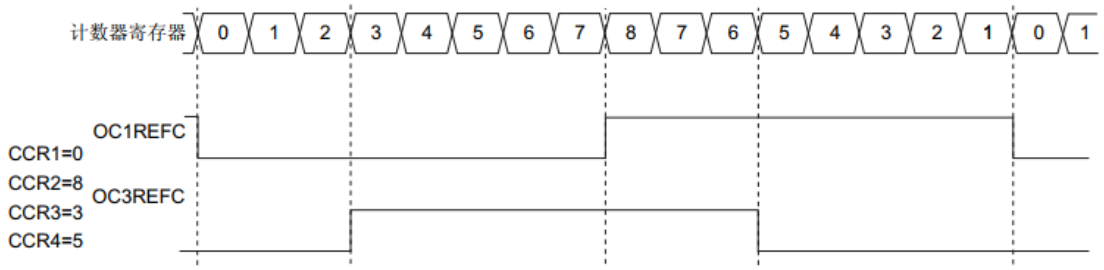


图 24.30 50% 占空比时产生的 2 个相移 PWM 信号

组合 PWM 模式：

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入 1100（组合 PWM 模式 1）或 1101（组合 PWM 模式 2）。

当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

下图显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。

- 通道 3 在组合 PWM 模式 2 下配置。
- 通道 4 在 PWM 模式 1 下配置。

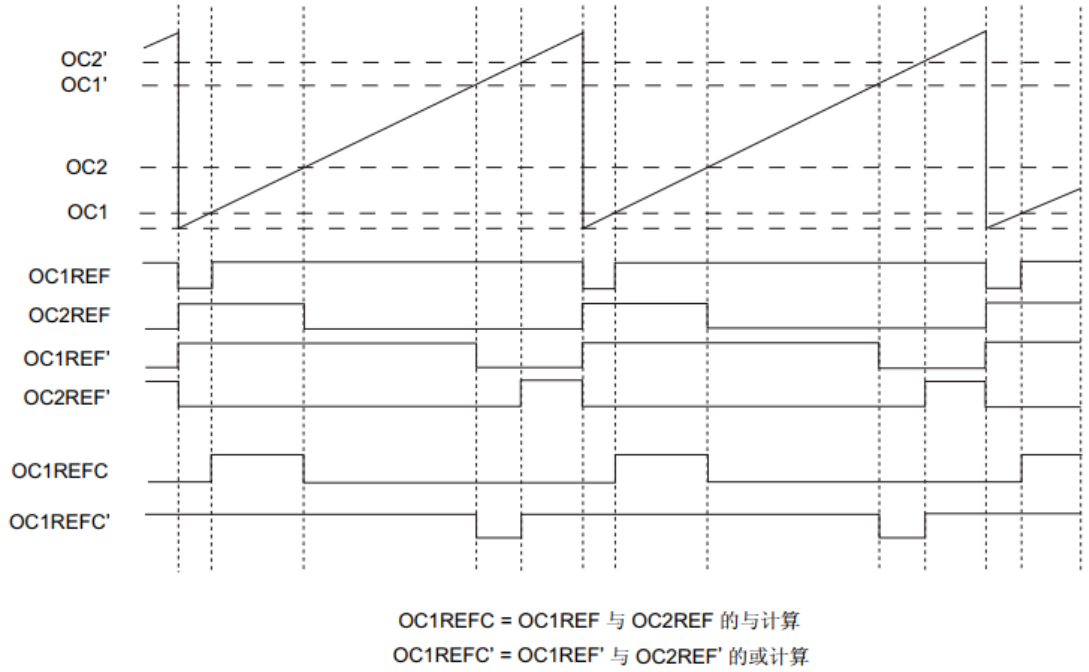


图 24.31 通道 1 和通道 3 上的组合 PWM 模式

24.3.4.1.2 比较寄存器预装载

比较寄存器具有预装载功能，其预装载功能由 TIM_CCMRx 寄存器的 OCxPE 位进行控制。预装载使能时（OCxPE=1）时，CCRx 寄存器进行缓冲，CCRx 可随时被写入，但其预装载值将会在每次发生更新事件时装载到对应的影子寄存器中，并用于比较；预装载未使能（OCxPE=0）时，CCRx 寄存器不进行缓冲，CCRx 可随时被写入，且写入后立即装载到影子寄存器中，并立即被用于比较。

下图给出了输出比较在预装载使能和未使能时的不同的时序图。当 OCxPE=0 时，输出比较预装载未使能，此时修改 CCRx 寄存器值，比较值将会被立即缓存并生效。当 OCxPE=1 时，输出比较预装载使能，此时修改 CCRx 寄存器值，比较值不会立即生效，而是等待计数器产生更新事件时才会进行缓存并生效。

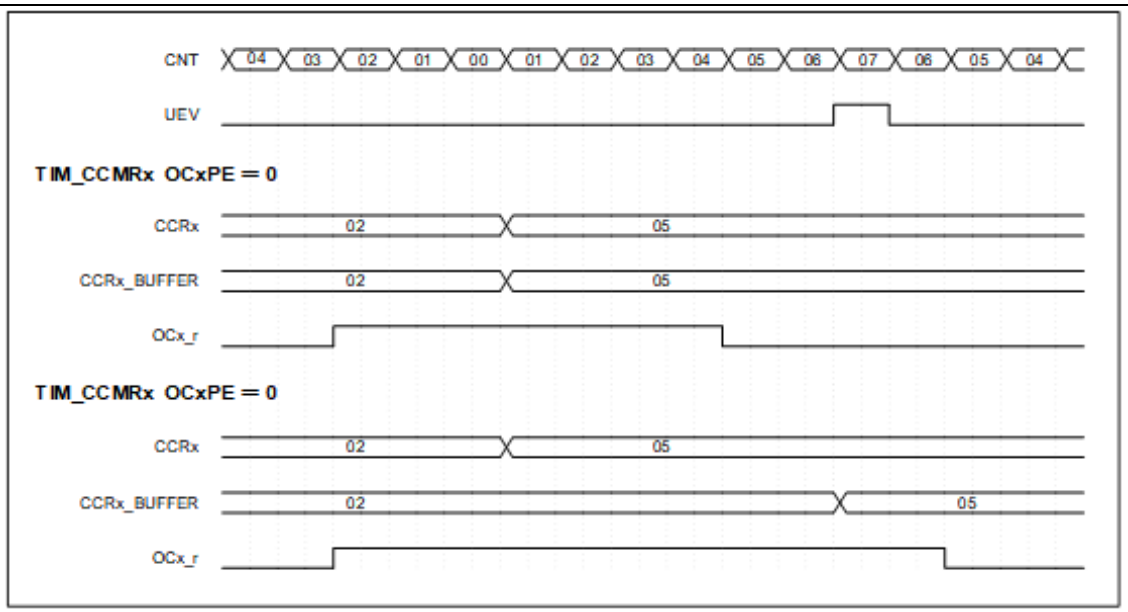


图 24.32 输出比较预装载时序图

24.3.4.2 互补输出和死区插入

高级定时器可以输出互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟）来调整死区时间。

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在断路）时，将使能死区插入。每个通道有一个 10 位死区发生器。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，只是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，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

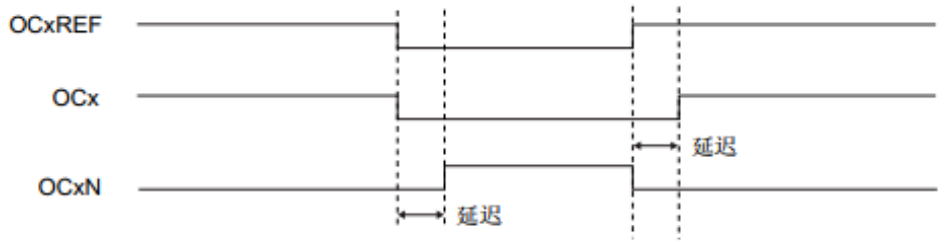


图 24.33 带死区插入的互补输出波形图

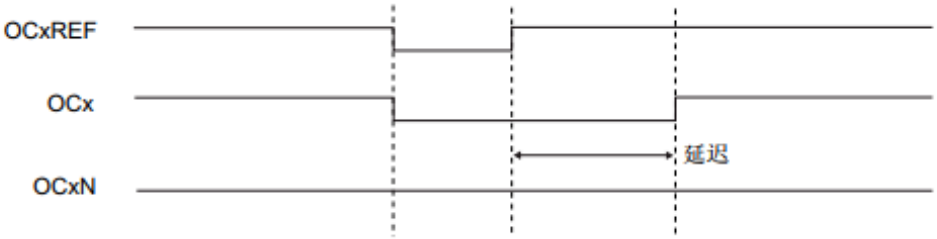


图 24.34 延迟时间大于负脉冲宽度的死区波形图

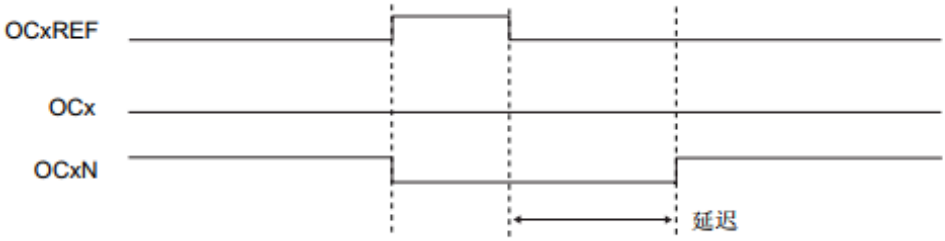


图 24.35 延迟时间大于正脉冲宽度的死区波形图

死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。

24.3.5 断路功能

断路期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生断路时复位。

- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 TIMx_BDTR 寄存器中的 BKE 位置 1，可使能断路功能。可通过配置同一寄存器中的 BKP 位来选择断路输入的极性。BKEx 和 BKPx 位可同时修改。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。

发生断路之一（其中一个断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，定时器将禁止输出，否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于无效状态（取决于极性）。
  - 在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。
  - 如果 OSSI=0，定时器将禁止输出，否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 SBIF、BIF 和 B2IF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，则会产生中断。如果 TIMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。
- 如果 TIMx_BDTR 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。例如，这可用于执行调节。否则，MOE 将始终保持低电平，直到应用将其再次置 1。

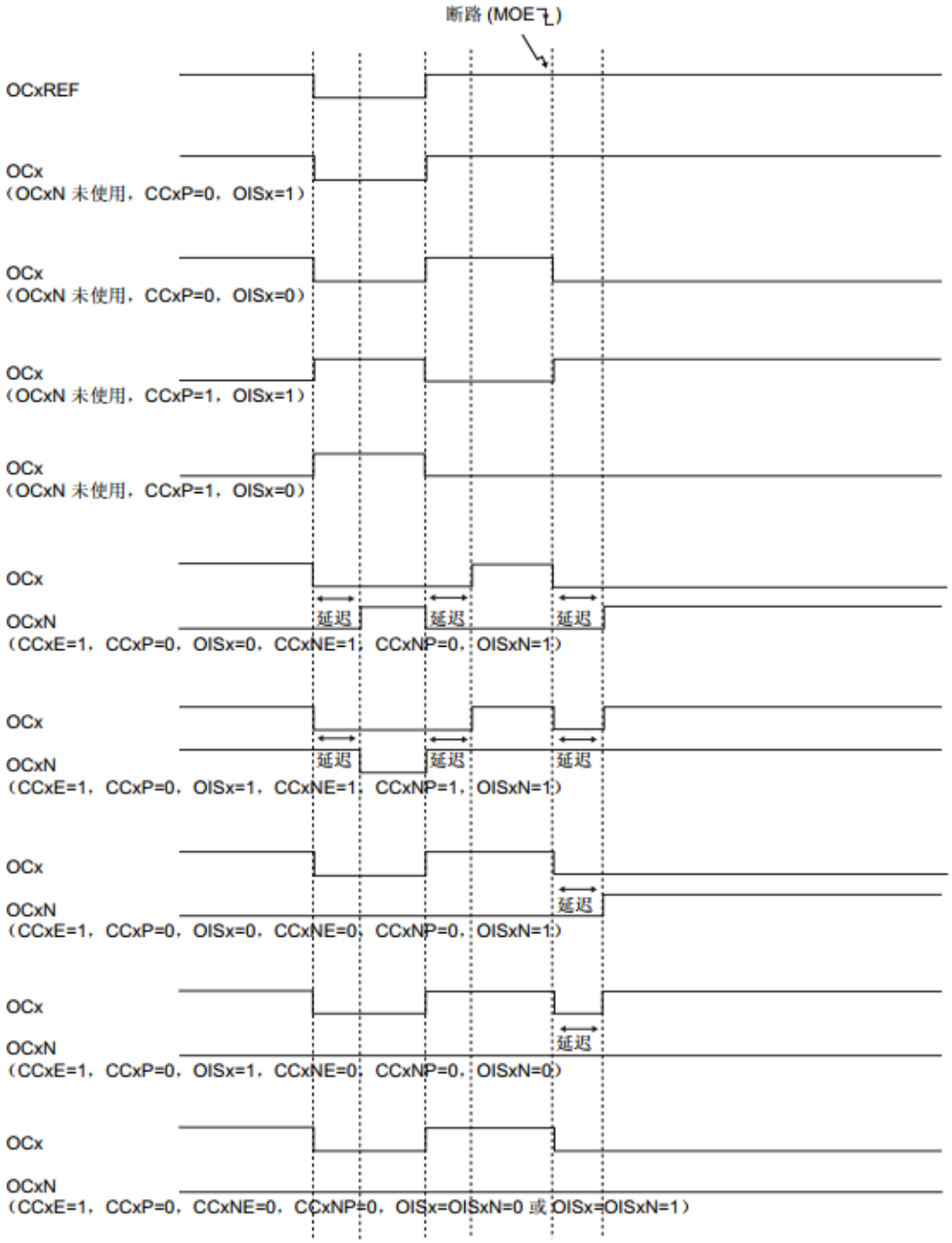


图 24.36 响应 BRK 上的断路事件的不同输出行为

下表描述了具有断路功能的互补通道 OCx 和 OCxN 的输出控制逻辑。

表 24.5 断路功能输出控制逻辑表

控制位					输出状态	
MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态

<table>
<tr><td colspan="5">控制位</td><td colspan="2">输出状态</td></tr>
<tr><td rowspan="7">1</td><td rowspan="7">x</td><td>x</td><td>0</td><td>0</td><td colspan="2">禁止输出 OCx=0、OCxN=0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>禁止输出 OCx=0</td><td>OCxREF+极性
OCxN=OCxREF 异或
CCxNP</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>OCxREF+极性
OCx=OCxREF 异或
CCxP</td><td>禁止输出 OCxN=0</td></tr>
<tr><td>x</td><td>1</td><td>1</td><td>OCxREF+极性+死区</td><td>OCxREF 互补项（对 OCxREF 取反）+极性+死区</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>关闭状态 OCx=CCxP</td><td>OCxREF+极性
OCxN=OCxREF 异或
CCxNP</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>OCxREF+极性
OCx=OCxREF 异或
CCxP</td><td>关闭状态 OCxN=CCxNP</td></tr>
<tr><td rowspan="4">0</td><td>0</td><td rowspan="4">x</td><td>x</td><td>x</td><td colspan="2" rowspan="2">禁止输出 OCx=0、OCxN=0</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td colspan="2" rowspan="3">关闭状态 异步：OCx=CCxP、OCxN=CCxNP 若时钟存在：经过一个死区时间后，OCx=OISx、OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 和 OCxN 的有效电平</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>

24.3.6 定时器同步

24.3.6.1 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP='0'，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，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，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。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。

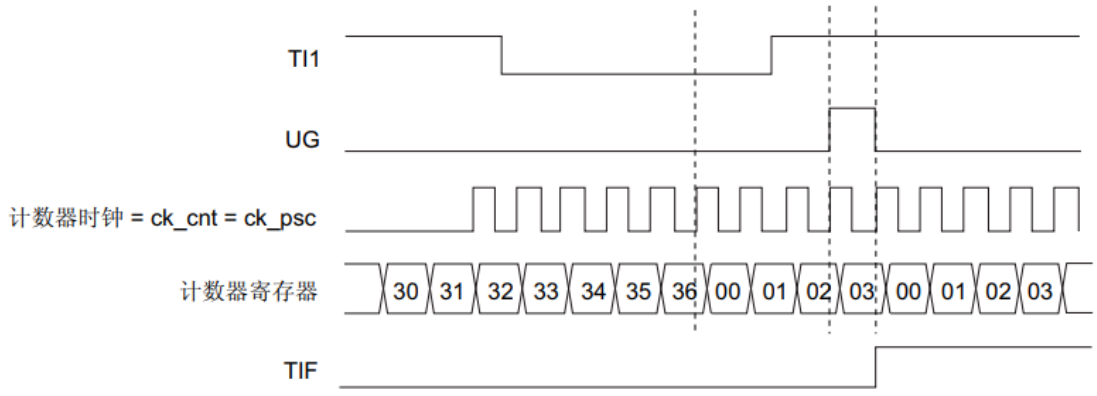


图 24.37 复位模式下的计数器时序图

24.3.6.1.1 从模式：门控模式

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_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。

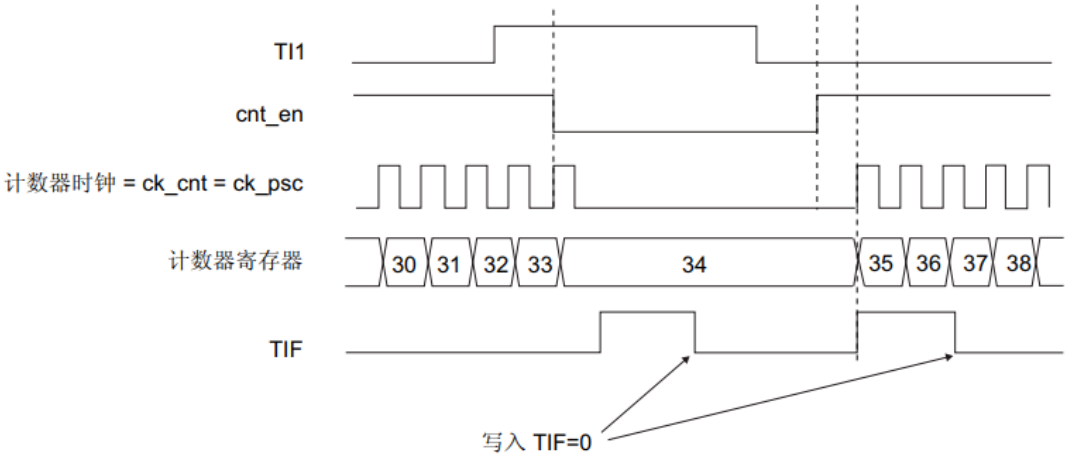


图 24.38 门控模式下的计数器时序图

24.3.6.1.2 从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 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。

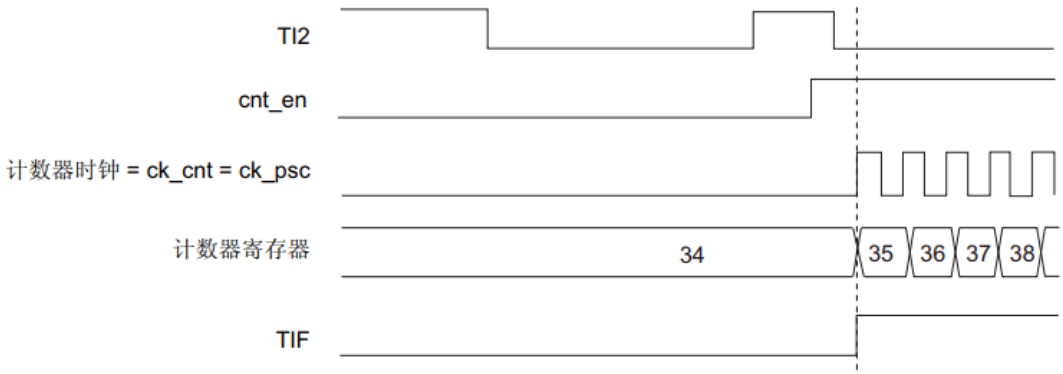


图 24.39 触发模式下的计数器时序图

24.3.6.1.3 从模式：组合复位+触发模式

在这种情况下，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。

该模式用于单脉冲模式。

24.3.6.1.4 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（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF=0000：无滤波器。
   - ETPS=00：禁止预分频器。
   - ETP=0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP=0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

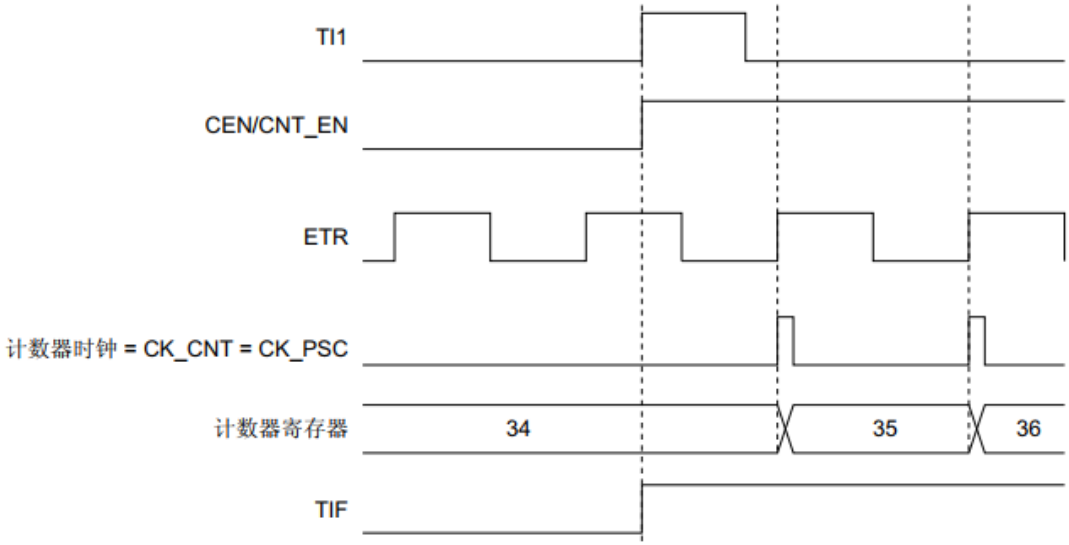


图 24.40 外部触发模式 2+触发模式下的计数器时序图

24.3.6.2 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4REF 的上升沿和下降沿
- OC5REF 上的上升沿或 OC6REF 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，它们可通过 TIMx_CR2 寄存器中的 MMS2 位选择。

24.3.7 DMA

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x=2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量=3。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：DBL=3 次传输，DBA=0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

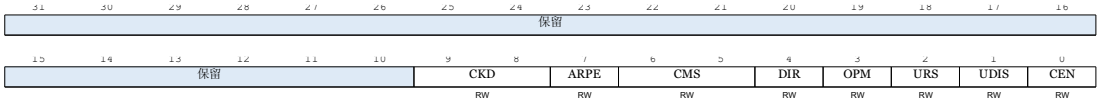
本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间 data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

24.4 寄存器

24.4.1 TIMx 控制寄存器 1（TIMx_CR1）

地址偏移：0x00

复位值：0x0000_0000



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKD	时钟分频 此位域指示定时器时钟（CK_INT）频率与死区发生器以及数字滤波器（ETR、TIx）所使用的死区及采样时钟（tDTS）之间的分频比， 00：tDTS=tCK_INT 01：tDTS=2*tCK_INT* *10：tDTS=4*tCK_INT 11：保留，不要设置成此值
7	ARPE	自动重载预装载使能 0：TIMx_ARR 寄存器不进行缓冲 1：TIMx_ARR 寄存器进行缓冲
6:5	CMS	中心对齐模式选择 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

位/位域	名称	描述
		（TIMx_CCMRx 寄存器中的 CCxS=00）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（TIMx_CCMRx 寄存器中的 CCxS=00）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（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。 注：只要计数器处于使能状态（CEN=1），就不得从边沿对齐模式切换为中心对齐模式。
4	DIR	方向 0：计数器递增计数 1：计数器递减计数 注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。
3	OPM	单脉冲模式 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）
2	URS	更新请求源 此位由软件置 1 和清零，用以选择 UEV 事件源。

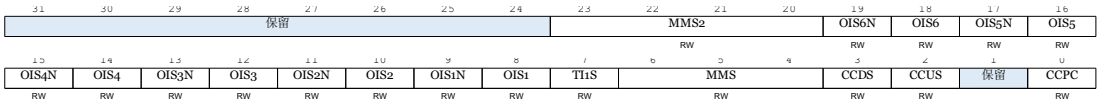
位/位域	名称	描述
		0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括： — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件 1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。
1	UDIS	更新禁止 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0：使能 UEV。更新（UEV）事件可通过以下事件之一生成： — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件 然后更新影子寄存器的值 1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。
0	CEN	计数器使能 0：禁止计数器 1：使能计数器

位/位域	名称	描述
		注：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

24.4.2 TIMx 控制寄存器 2（TIMx_CR2）

地址偏移：0x04

复位值：0x0000_0000



位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	MMS2	主模式选择 2 这些位可选择将发送到 ADC 以实现同步的信息(TRGO2)。这些位的组合如下： 0000：复位——TIMx_EGR 寄存器中的 UG 位用作触发输出(TRGO2)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO2 上的信号相比实际复位会有延迟。 0001：使能——计数器使能信号 CNT_EN 用作触发输出(TRGO2)。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。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。 0010：更新——选择更新事件作为触发输出(TRGO2)。

位/位域	名称	描述
		0011：比较脉冲——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出(TRGO2)都会发送一个正脉冲。 0100：比较——OC1REF 信号用作触发输出(TRGO2) 0101：比较——OC2REF 信号用作触发输出(TRGO2) 0110：比较——OC3REF 信号用作触发输出(TRGO2) 0111：比较——OC4REF 信号用作触发输出(TRGO2) 1000：比较——OC5REF 信号用作触发输出(TRGO2) 1001：比较——OC6REF 信号用作触发输出(TRGO2) 1010：比较脉冲——OC4REF 上升沿或下降沿时，TRGO2 上生成脉冲 1011：比较脉冲——OC6REF 上升沿或下降沿时，TRGO2 上生成脉冲 1100：比较脉冲——OC4REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲 1101：比较脉冲——OC4REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲 1110：比较脉冲——OC5REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲 1111：比较脉冲——OC5REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲 注：必须先使能 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改 ADC 的时钟。
19	OIS6N	输出空闲状态 6（OC6N 输出）

位/位域	名称	描述
		请参见 OIS1N 位
18	OIS6	输出空闲状态 6（OC6 输出） 请参见 OIS1 位
17	OIS5N	输出空闲状态 5（OC5N 输出） 请参见 OIS1N 位
16	OIS5	输出空闲状态 5（OC5 输出） 请参见 OIS1 位
15	OIS4N	输出空闲状态 4（OC4N 输出） 请参见 OIS1N 位
14	OIS4	输出空闲状态 4（OC4 输出） 请参见 OIS1 位
13	OIS3N	输出空闲状态 3（OC3N 输出） 请参见 OIS1N 位
12	OIS3	输出空闲状态 3（OC3 输出） 请参见 OIS1 位
11	OIS2N	输出空闲状态 2（OC2N 输出） 请参见 OIS1N 位
10	OIS2	输出空闲状态 2（OC2 输出）

位/位域	名称	描述
		请参见 OIS1 位
9	OIS1N	输出空闲状态 1（OC1N 输出） 0：当 MOE=0 时，经过死区时间后 OC1N=0 1：当 MOE=0 时，经过死区时间后 OC1N=1
8	OIS1	输出空闲状态 1（OC1 输出） 0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0 1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1
7	TI1S	TI1 选择 0：TIMx_CH1 引脚连接到 TI1 输入 1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）
6:4	MMS	主模式选择 这些位可选择主模式下将要发送到从定时器以实现同步的信息(TRGO)。这些位的组合如下： 000：复位——TIMx_EGR 寄存器中的 UG 位用作触发输出(TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。 001：使能——计数器使能信号 CNT_EN 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。计数器使能信号由

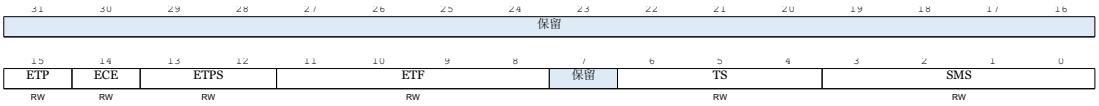
位/位域	名称	描述
		CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟。 010：更新——选择更新事件作为触发输出(TRGO)。例如，主定时器可用作从定时器的预分频器。 011：比较脉冲——旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高），触发输出都会发送一个正脉冲。(TRGO)。 100：比较——OC1REF 信号用作触发输出(TRGO) 101：比较——OC2REF 信号用作触发输出(TRGO) 110：比较——OC3REF 信号用作触发输出(TRGO) 111：比较——OC4REF 信号用作触发输出(TRGO) 注：必须先使能从定时器的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器的时钟。
3	CCDS	捕获/比较 DMA 选择 0：发生 CCx 事件时发送 CCx DMA 请求 1：发生更新事件时发送 CCx DMA 请求
2	CCUS	捕获/比较控制更新选择 0：如果捕获/比较控制位进行预装载（CCPC=1），仅通过将 COMG 位置 1 来对这些位进行更新 1：如果捕获/比较控制位进行预装载（CCPC=1），可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

位/位域	名称	描述
		注：此位仅对具有互补输出的通道有效。
1	保留	必须保持复位值
0	CCPC	捕获/比较预装载控制 0：CCxE、CCxNE 和 OCxM 位未进行预装载 1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件（COM）（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。 注：此位仅对具有互补输出的通道有效。

24.4.3 TIMx 从模式控制寄存器（TIMx_SMCR）

地址偏移：0x08

复位值：0x0000_0000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 此位可选择将 ETR 还是 ETR 用于触发操作 0：ETR 未反相，高电平或上升沿有效。 1：ETR 反相，低电平或下降沿有效。
14	ECE	外部时钟使能 此位可使能外部时钟模式 2。 0：禁止外部时钟模式 2

位/位域	名称	描述
		1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。 注： 1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。 2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。 3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。
13:12	ETPS	外部触发预分频器 外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。 00：预分频器关闭 01：2 分频 ETRP 频率 10：4 分频 ETRP 频率 11：8 分频 ETRP 频率
11:8	ETF	外部触发滤波器 此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿： 0000：无滤波器，按 fDTS 频率进行采样 0001：fSAMPLING=fCK_INT，N=2

位/位域	名称	描述
		0010：fSAMPLING=fCK_INT，N=4 0011：fSAMPLING=fCK_INT，N=8 0100：fSAMPLING=fDTS/2，N=6 0101：fSAMPLING=fDTS/2，N=8 0110：fSAMPLING=fDTS/4，N=6 0111：fSAMPLING=fDTS/4，N=8 1000：fSAMPLING=fDTS/8，N=6 1001：fSAMPLING=fDTS/8，N=8 1010：fSAMPLING=fDTS/16，N=5 1011：fSAMPLING=fDTS/16，N=6 1100：fSAMPLING=fDTS/16，N=8 1101：fSAMPLING=fDTS/32，N=5 1110：fSAMPLING=fDTS/32，N=6 1111：fSAMPLING=fDTS/32，N=8
7	保留	必须保持复位值
6:4	TS	触发选择 此位域与 TS[4:3]位组合在一起。 此位域可选择将要用于同步计数器的触发输入。 00000：内部触发 0(ITR0) 00001：内部触发 1(ITR1) 00010：内部触发 2(ITR2) 00011：内部触发 3(ITR3) 00100：TI1 边沿检测器(TI1F_ED) 00101：滤波后的定时器输入 1(TI1FP1) 00110：滤波后的定时器输入 2(TI2FP2)

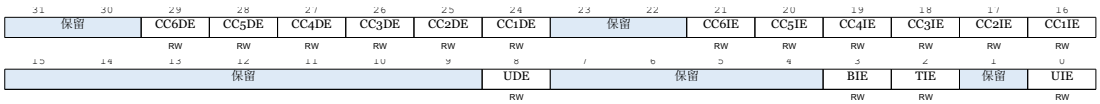
位/位域	名称	描述
		00111：外部触发输入(ETRF) 其它值：保留 注：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。
3:0	SMS	从模式选择 选择外部信号时，触发信号(TRGI)的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。 0000：禁止从模式——如果 CEN=1，预分频器时钟直接由内部时钟提供。 0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。 0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。 0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。 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。 0111：外部时钟模式 1——由所选触发信号(TRGI)的上升沿提供计数器时钟。

位/位域	名称	描述
		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。 1000 以上的代码：保留。 注：如果将 TI1F_ED 选作触发输入(TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。 注：必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

24.4.4 TIMx DMA/中断使能寄存器（TIMx_DIER）

地址偏移：0x0C

复位值：0x0000_0000



位/位域	名称	描述
31:30	保留	必须保持复位值
29	CC6DE	捕获/比较 6 DMA 请求使能 0：禁止 CC6 DMA 请求 1：使能 CC6 DMA 请求
28	CC5DE	捕获/比较 5 DMA 请求使能 0：禁止 CC5 DMA 请求 1：使能 CC5 DMA 请求

位/位域	名称	描述
27	CC4DE	捕获/比较 4 DMA 请求使能 0：禁止 CC4 DMA 请求 1：使能 CC4 DMA 请求
26	CC3DE	捕获/比较 3 DMA 请求使能 0：禁止 CC3 DMA 请求 1：使能 CC3 DMA 请求
25	CC2DE	捕获/比较 2 DMA 请求使能 0：禁止 CC2 DMA 请求 1：使能 CC2 DMA 请求
24	CC1DE	捕获/比较 1 DMA 请求使能 0：禁止 CC1 DMA 请求 1：使能 CC1 DMA 请求
23:22	保留	必须保持复位值
21	CC6IE	捕获/比较 6 中断使能 0：禁止 CC6 中断 1：使能 CC6 中断
20	CC5IE	捕获/比较 5 中断使能 0：禁止 CC5 中断 1：使能 CC5 中断
19	CC4IE	捕获/比较 4 中断使能

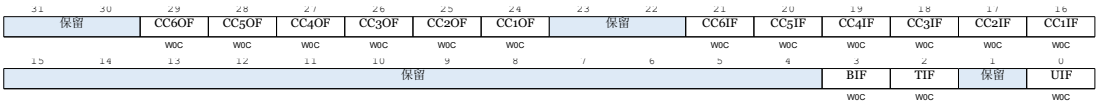
位/位域	名称	描述
		0：禁止 CC4 中断 1：使能 CC4 中断
18	CC3IE	捕获/比较 3 中断使能 0：禁止 CC3 中断 1：使能 CC3 中断
17	CC2IE	捕获/比较 2 中断使能 0：禁止 CC2 中断 1：使能 CC2 中断
16	CC1IE	捕获/比较 1 中断使能 0：禁止 CC1 中断 1：使能 CC1 中断
15:9	保留	必须保持复位值
8	UDE	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:4	保留	必须保持复位值
3	BIE	断路中断使能 0：禁止断路中断 1：使能断路中断
2	TIE	触发中断使能 0：禁止触发中断

位/位域	名称	描述
		1：使能触发中断
1	保留	必须保持复位值
0	UIE	更新中断使能 0：禁止更新中断 1：使能更新中断

24.4.5 TIMx 状态寄存器（TIMx_SR）

地址偏移：0x10

复位值：0x0000_0000



位/位域	名称	描述
31:30	保留	必须保持复位值
29	CC6OF	捕获/比较 6 重复捕获标志 请参见 CC1OF 说明
28	CC5OF	捕获/比较 5 重复捕获标志 请参见 CC1OF 说明
27	CC4OF	捕获/比较 4 重复捕获标志 请参见 CC1OF 说明
26	CC3OF	捕获/比较 3 重复捕获标志 请参见 CC1OF 说明
25	CC2OF	捕获/比较 2 重复捕获标志

位/位域	名称	描述
		请参见 CC1OF 说明
24	CC1OF	捕获/比较 1 重复捕获标志 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。通过软件写入 0 可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。
23:22	保留	必须保持复位值
21	CC6IF	捕获/比较 6 中断标志 请参见 CC1IF 说明
20	CC5IF	捕获/比较 5 中断标志 请参见 CC1IF 说明
19	CC4IF	捕获/比较 4 中断标志 请参见 CC1IF 说明
18	CC3IF	捕获/比较 3 中断标志 请参见 CC1IF 说明
17	CC2IF	捕获/比较 2 中断标志 请参见 CC1IF 说明
16	CC1IF	捕获/比较 1 中断标志

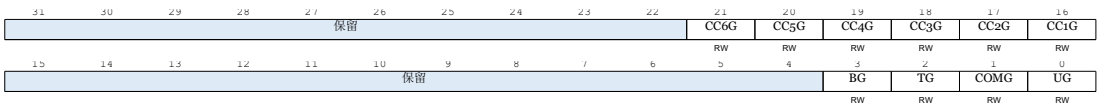
位/位域	名称	描述
		如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。 0：不匹配。 1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。 如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。 0：未发生输入捕获事件 1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）
15:4	保留	必须保持复位值
3	BIF	断路中断标志 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。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。
2	TIF	触发中断标志

位/位域	名称	描述
		在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。 0：未发生触发事件。 1：触发中断挂起。
1	保留	必须保持复位值
0	UIF	更新中断标志 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： ——TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器=0 时更新）。 ——TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。 ——TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时

24.4.6 TIMx 事件生成寄存器（TIMx_EGR）

地址偏移：0x14

复位值：0x0000_0000



位/位域	名称	描述
31:22	保留	必须保持复位值
21	CC6G	捕获/比较 6 生成 请参见 CC1G 说明
20	CC5G	捕获/比较 5 生成 请参见 CC1G 说明
19	CC4G	捕获/比较 4 生成 请参见 CC1G 说明
18	CC3G	捕获/比较 3 生成 请参见 CC1G 说明
17	CC2G	捕获/比较 2 生成 请参见 CC1G 说明
16	CC1G	捕获/比较 1 生成 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：通道 1 上生成捕获/比较事件： 如果通道 CC1 配置为输出： 使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果通道 CC1 配置为输入：

位/位域	名称	描述
		TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。
15:4	保留	必须保持复位值
3	BG	断路生成 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：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。
2	TG	触发生成 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：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。
1	COMG	捕获/比较控制更新生成 该位可通过软件置 1，并由硬件自动清零 0：不执行任何操作 1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位 注： 此位仅对具有互补输出的通道有效。
0	UG	更新生成 该位可通过软件置 1，并由硬件自动清零。 0：不执行任何操作

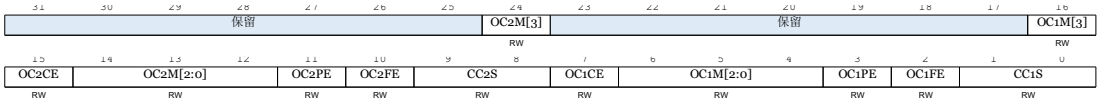
位/位域	名称	描述
		1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

24.4.7 TIMx 捕获/比较模式寄存器 1（TIMx_CCMR1）

地址偏移：0x18

复位值：0x0000_0000

输出比较：



位/位域	名称	描述
31:25	保留	必须保持复位值
24	CC2M[3]	输出比较 2 模式——位 3 请参见 OC2M 说明
23:17	保留	必须保持复位值
16	CC1M[3]	输出比较 1 模式——位 3 请参见 OC1M 说明
15	OC2CE	输出比较 2 清零使能
14:12	OC2M[2:0]	输出比较 2 模式
11	OC2PE	输出比较 2 预装载使能
10	OC2FE	输出比较 2 快速使能
9:8	CC2S	捕获/比较 2 选择

位/位域	名称	描述
		此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上 10：CC2 通道配置为输入，IC2 映射到 TI1 上 11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC2E=0），才可向 CC2S 位写入数据。
7	OC1CE	输出比较 1 清零使能 0：OC1Ref 不受 ETRF 输入影响 1：ETRF 输入上检测到高电平时，OC1Ref 立即清零
6:4	OC1M	输出比较 1 模式 这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。 0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

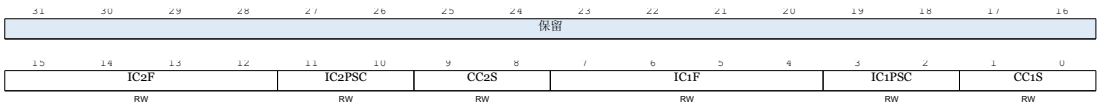
位/位域	名称	描述
		0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。 0011 ： 翻 转 ——TIMx_CNT=TIMx_CCR1 时 ，OC1REF 发生翻转。 0100：强制变为无效电平——OC1REF 强制变为低电平。 0101：强制变为有效电平——OC1REF 强制变为高电平。 0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，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 TIMx_CNT < TIMx_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 TIMx_CNT>TIMx_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有效状态。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。

位/位域	名称	描述
		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。 1010：保留。 1011：保留。 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。 注： 在 PWM 模式下，仅当比较结果发生改变或输出比较模式由冻结模式切换到 PWM 模式时， OCREF 电平才会发生更改。

位/位域	名称	描述
		注： 此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。
3	OC1PE	输出比较 1 预装载使能 0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。 1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。 2：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。
2	OC1FE	输出比较 1 快速使能 此位用于加快触发输入事件对 CC 输出的影响。 0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。 1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。
1:0	CC1S	捕获/比较 1 选择

位/位域	名称	描述
		此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上 11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC1E=0），才可向 CC1S 位写入数据。

输入捕获：



位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC2F	输入捕获 2 滤波器
11:10	IC2PSC	输入捕获 2 预分频器
9:8	CC2S	捕获/比较 2 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上 10：CC2 通道配置为输入，IC2 映射到 TI1 上

位/位域	名称	描述
		11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC2E=0），才可向 CC2S 位写入数据。
7:4	IC1F	输入捕获 1 滤波器 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿： 0000：无滤波器，按 fDTS 频率进行采样 0001：fSAMPLING=fCK_INT，N=2 0010：fSAMPLING=fCK_INT，N=4 0011：fSAMPLING=fCK_INT，N=8 0100：fSAMPLING=fDTS/2，N=6 0101：fSAMPLING=fDTS/2，N=8 0110：fSAMPLING=fDTS/4，N=6 0111：fSAMPLING=fDTS/4，N=8 1000：fSAMPLING=fDTS/8，N=6 1001：fSAMPLING=fDTS/8，N=8 1010：fSAMPLING=fDTS/16，N=5 1011：fSAMPLING=fDTS/16，N=6 1100：fSAMPLING=fDTS/16，N=8 1101：fSAMPLING=fDTS/32，N=5 1110：fSAMPLING=fDTS/32，N=6 1111：fSAMPLING=fDTS/32，N=8

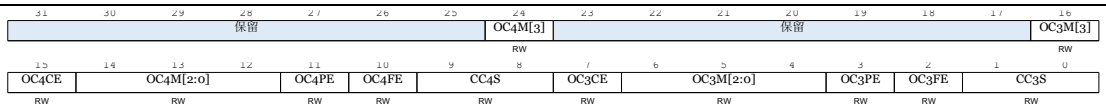
位/位域	名称	描述
3:2	IC1PSC	输入捕获 1 预分频器 此位域定义 CC1 输入(IC1)的预分频比。只要 CC1E=0（TIMx_CCER 寄存器），预分频器便立即复位。 00：无预分频器，捕获输入上每检测到一个边沿便执行捕获 01：每发生 2 个事件便执行一次捕获 10：每发生 4 个事件便执行一次捕获 11：每发生 8 个事件便执行一次捕获
1:0	CC1S	捕获/比较 1 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上 11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC1E=0），才可向 CC1S 位写入数据。

24.4.8 TIMx 捕获/比较模式寄存器 2（TIMx_CCMR2）

地址偏移：0x1C

复位值：0x0000_0000

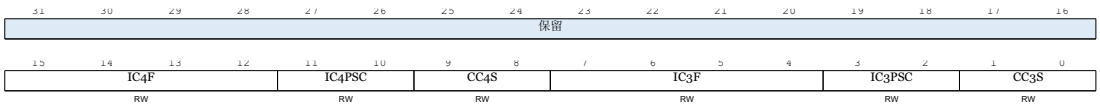
输出比较：



位/位域	名称	描述
31:25	保留	必须保持复位值
24	CC4M[3]	输出比较 4 模式——位 3 请参见 OC2M 说明
23:17	保留	必须保持复位值
16	CC3M[3]	输出比较 3 模式——位 3 请参见 OC1M 说明
15	OC4CE	输出比较 4 清零使能
14:12	OC4M[2:0]	输出比较 4 模式
11	OC4PE	输出比较 4 预装载使能
10	OC4FE	输出比较 4 快速使能
9:8	CC4S	捕获/比较 4 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 上 10：CC4 通道配置为输入，IC4 映射到 TI3 上 11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC4E=0），才可向 CC4S 位写入数据。
7	OC3CE	输出比较 3 清零使能

位/位域	名称	描述
6:4	OC3M	输出比较 3 模式
3	OC3PE	输出比较 3 预装载使能
2	OC3FE	输出比较 3 快速使能
1:0	CC3S	捕获/比较 3 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 上 10：CC3 通道配置为输入，IC3 映射到 TI4 上 11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC3E=0），才可向 CC3S 位写入数据。

输入捕获：



位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC4F	输入捕获 4 滤波器
11:10	IC4PSC	输入捕获 4 预分频器
9:8	CC4S	捕获/比较 4 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC4 通道配置为输出

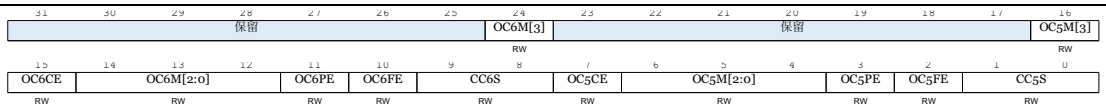
位/位域	名称	描述
		01：CC4 通道配置为输入，IC4 映射到 TI4 上 10：CC4 通道配置为输入，IC4 映射到 TI3 上 11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC4E=0），才可向 CC4S 位写入数据。
7:4	IC3F	输入捕获 3 滤波器
3:2	IC3PSC	输入捕获 3 预分频器
1:0	CC3S	捕获/比较 3 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 上 10：CC3 通道配置为输入，IC3 映射到 TI4 上 11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC3E=0），才可向 CC3S 位写入数据。

24.4.9 TIMx 捕获/比较模式寄存器 3（TIMx_CCMR3）

地址偏移：0x20

复位值：0x0000_0000

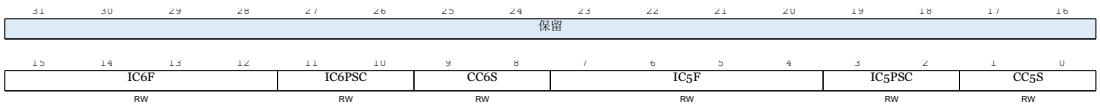
输出比较：



位/位域	名称	描述
31:25	保留	必须保持复位值
24	CC6M[3]	输出比较 6 模式——位 3 请参见 OC2M 说明
23:17	保留	必须保持复位值
16	CC5M[3]	输出比较 5 模式——位 3 请参见 OC1M 说明
15	OC6CE	输出比较 6 清零使能
14:12	OC6M[2:0]	输出比较 6 模式
11	OC6PE	输出比较 6 预装载使能
10	OC6FE	输出比较 6 快速使能
9:8	CC6S	捕获/比较 6 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC6 通道配置为输出 01：CC6 通道配置为输入，IC6 映射到 TI6 上 10：CC6 通道配置为输入，IC6 映射到 TI5 上 11：CC6 通道配置为输入，IC6 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC6E=0），才可向 CC6S 位写入数据。
7	OC5CE	输出比较 5 清零使能

位/位域	名称	描述
6:4	OC5M	输出比较 5 模式
3	OC5PE	输出比较 5 预装载使能
2	OC5FE	输出比较 5 快速使能
1:0	CC5S	捕获/比较 5 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC5 通道配置为输出 01：CC5 通道配置为输入，IC5 映射到 TI5 上 10：CC5 通道配置为输入，IC5 映射到 TI6 上 11：CC5 通道配置为输入，IC5 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC5E=0），才可向 CC5S 位写入数据。

输入捕获：



位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC6F	输入捕获 6 滤波器
11:10	IC6PSC	输入捕获 6 预分频器
9:8	CC6S	捕获/比较 6 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC6 通道配置为输出

位/位域	名称	描述
		01：CC6 通道配置为输入，IC6 映射到 TI6 上 10：CC6 通道配置为输入，IC6 映射到 TI5 上 11：CC6 通道配置为输入，IC6 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC6E=0），才可向 CC6S 位写入数据。
7:4	IC5F	输入捕获 5 滤波器
3:2	IC5PSC	输入捕获 5 预分频器
1:0	CC5S	捕获/比较 5 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC5 通道配置为输出 01：CC5 通道配置为输入，IC5 映射到 TI5 上 10：CC5 通道配置为输入，IC5 映射到 TI6 上 11：CC5 通道配置为输入，IC5 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC5E=0），才可向 CC5S 位写入数据。

24.4.10 TIMx 捕获/比较使能寄存器（TIMx_CCER）

地址偏移：0x24

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CC6NP	CC6NE	CC6P	CC6E	CC5NP	CC5NE	CC5P	CC5E
								RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:24	保留	必须保持复位值
23	CC6NP	捕获/比较 6 互补输出极性 请参见 CC1NP 说明
22	CC6NE	捕获/比较 6 互补输出使能 请参见 CC1NE 说明
21	CC6P	捕获/比较 6 输出极性 请参见 CC1P 说明
20	CC6E	捕获/比较 6 输出使能 请参见 CC1E 说明
19	CC5NP	捕获/比较 5 互补输出极性 请参见 CC1NP 说明
18	CC5NE	捕获/比较 5 互补输出使能 请参见 CC1NE 说明
17	CC5P	捕获/比较 5 输出极性 请参见 CC1P 说明
16	CC5E	捕获/比较 5 输出使能 请参见 CC1E 说明
15	CC4NP	捕获/比较 4 互补输出极性

位/位域	名称	描述
		请参见 CC1NP 说明
14	CC4NE	捕获/比较 4 互补输出使能 请参见 CC1NE 说明
13	CC4P	捕获/比较 4 输出极性 请参见 CC1P 说明
12	CC4E	捕获/比较 4 输出使能 请参见 CC1E 说明
11	CC3NP	捕获/比较 3 互补输出极性 请参见 CC1NP 说明
10	CC3NE	捕获/比较 3 互补输出使能 请参见 CC1NE 说明
9	CC3P	捕获/比较 3 输出极性 请参见 CC1P 说明
8	CC3E	捕获/比较 3 输出使能 请参见 CC1E 说明
7	CC2NP	捕获/比较 2 互补输出极性 请参见 CC1NP 说明
6	CC2NE	捕获/比较 2 互补输出使能

位/位域	名称	描述
		请参见 CC1NE 说明
5	CC2P	捕获/比较 2 输出极性 请参见 CC1P 说明
4	CC2E	捕获/比较 2 输出使能 请参见 CC1E 说明
3	CC1NP	捕获/比较 1 互补输出极性 CC1 通道配置为输出： 0：OC1N 高电平有效。 1：OC1N 低电平有效。 CC1 通道配置为输入： 此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。 注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。
2	CC1NE	捕获/比较 1 互补输出使能 0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。 1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

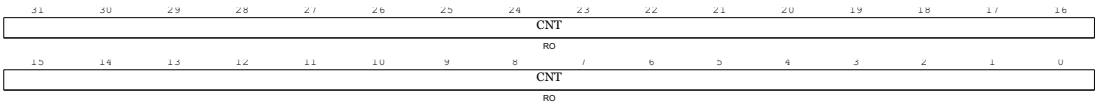
位/位域	名称	描述
		注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。
1	CC1P	捕获/比较 1 输出极性 CC1 通道配置为输出： 0：OC1 高电平有效 1：OC1 低电平有效 CC1 通道配置为输入：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。 00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。 01：反相/下降沿触发。电路对 TIxFP1 下升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。 10：保留，不使用此配置。 11：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

位/位域	名称	描述
		注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。
0	CC1E	捕获/比较 1 输出使能 CC1 通道配置为输出： 0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。 1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。 CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1(TIMx_CCR1)中。 0：禁止捕获。 1：使能捕获。 注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

24.4.11 TIMx 计数器（TIMx_CNT）

地址偏移：0x28

复位值：0x0000_0000

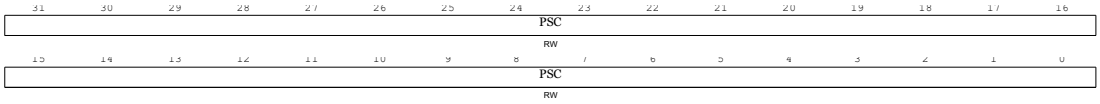


位/位域	名称	描述
31:0	CNT	计数器值

24.4.12 TIMx 预分频器（TIMx_PSC）

地址偏移：0x2C

复位值：0x0000_0000

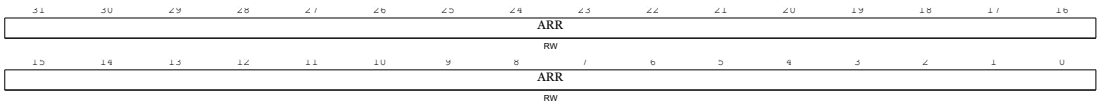


位/位域	名称	描述
31:0	PSC	预分频器值 计数器时钟频率 (CK_CNT) 等于 fCK_PSC / (PSC + 1)。 PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为复位模式时通过触发控制器清零时）时要装载到有效预分频器寄存器的值。

24.4.13 TIMx 自动重载寄存器（TIMx_ARR）

地址偏移：0x30

复位值：0xFFFF_FFFF

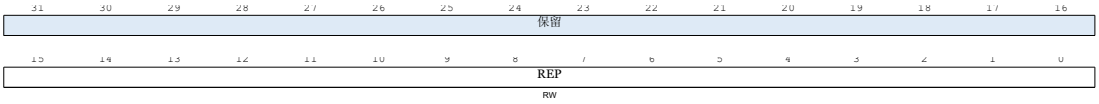


位/位域	名称	描述
31:0	ARR	自动重载值 ARR 为要装载到实际自动重载寄存器的值。 有关 ARR 更新和行为的更多详细信息，请参见时基单元。 当自动重载值为空时，计数器不工作。

24.4.14 TIMx 重复计数器寄存器（TIMx_RCR）

地址偏移：0x34

复位值：0x0000_0000

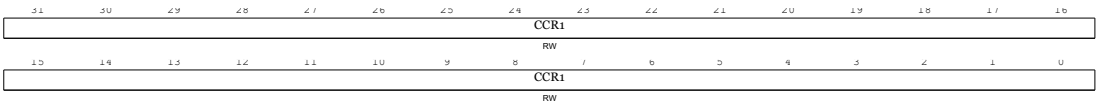


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	REP	重复计数器值 使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向有效寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。 由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。 这意味着 PWM 模式下 (REP+1) 相当于： 边沿对齐模式下的 PWM 周期数。 中心对齐模式下的 PWM 半周期数。

24.4.15 TIMx 捕获/比较寄存器 1（TIMx_CCR1）

地址偏移：0x38

复位值：0x0000_0000

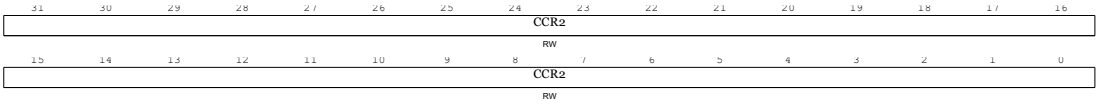


位/位域	名称	描述
31:0	CCR1	捕获/比较 1 值 如果通道 CC1 配置为输出：CCR1 为要装载到有效捕获/比较 1 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。 如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

24.4.16 TIMx 捕获/比较寄存器 2（TIMx_CCR2）

地址偏移：0x3C

复位值：0x0000_0000



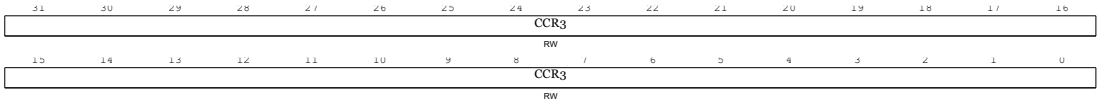
位/位域	名称	描述
31:0	CCR2	捕获/比较 2 值 如果通道 CC2 配置为输出：CCR2 为要装载到有效捕获/比较 2 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。

位/位域	名称	描述
		有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。 如果通道 CC2 配置为输入：CR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

24.4.17 TIMx 捕获/比较寄存器 3（TIMx_CCR3）

地址偏移：0x40

复位值：0x0000_0000

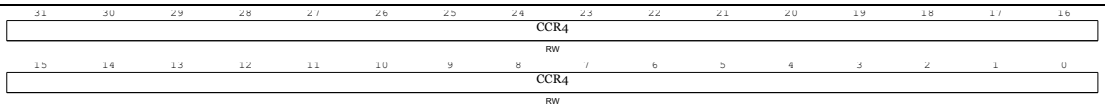


位/位域	名称	描述
31:0	CCR3	捕获/比较 3 值 如果通道 CC3 配置为输出：CCR3 为要装载到有效捕获/比较 3 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR3 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。 如果通道 CC3 配置为输入：CR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

24.4.18 TIMx 捕获/比较寄存器 4（TIMx_CCR4）

地址偏移：0x44

复位值：0x0000_0000

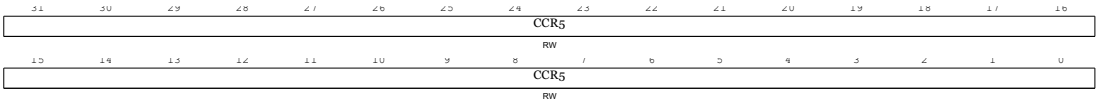


位/位域	名称	描述
31:0	CCR4	捕获/比较 4 值 如果通道 CC4 配置为输出：CCR4 为要装载到有效捕获/比较 4 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR4 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。 如果通道 CC4 配置为输入：CR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

24.4.19 TIMx 捕获/比较寄存器 5（TIMx_CCR5）

地址偏移：0x48

复位值：0x0000_0000



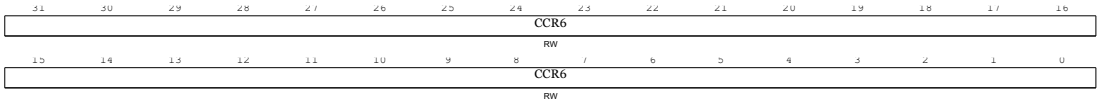
位/位域	名称	描述
31:0	CCR5	捕获/比较 5 值 如果通道 CC5 配置为输出：CCR5 为要装载到有效捕获/比较 5 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR5 寄存器中的 OC5PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 5）。

位/位域	名称	描述
		有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。 如果通道 CC5 配置为输入：CR5 为上一个输入捕获 5 事件 (IC5) 发生时的计数器值。只能读取 TIMx_CCR5 寄存器，无法对其进行编程。

24.4.20 TIMx 捕获/比较寄存器 6（TIMx_CCR6）

地址偏移：0x4C

复位值：0x0000_0000

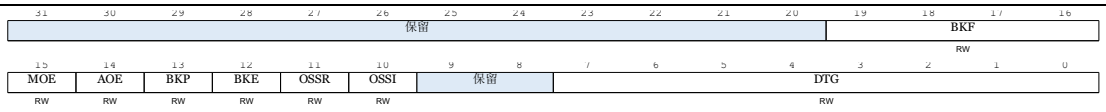


位/位域	名称	描述
31:0	CCR6	捕获/比较 6 值 如果通道 CC6 配置为输出：CCR6 为要装载到有效捕获/比较 6 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR6 寄存器中的 OC6PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 6）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。 如果通道 CC6 配置为输入：CR6 为上一个输入捕获 6 事件 (IC6) 发生时的计数器值。只能读取 TIMx_CCR6 寄存器，无法对其进行编程。

24.4.21 TIMx 断路和死区寄存器（TIMx_BDTR）

地址偏移：0x50

复位值：0x0000_0000



位/位域	名称	描述
31:20	保留	必须保持复位值
19:16	BKF	断路滤波器 此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿： 0000：无滤波器，BRK 异步工作 0001：fSAMPLING=fCK_INT，N=2 0010：fSAMPLING=fCK_INT，N=4 0011：fSAMPLING=fCK_INT，N=8 0100：fSAMPLING=fDTS/2，N=6 0101：fSAMPLING=fDTS/2，N=8 0110：fSAMPLING=fDTS/4，N=6 0111：fSAMPLING=fDTS/4，N=8 1000：fSAMPLING=fDTS/8，N=6 1001：fSAMPLING=fDTS/8，N=8 1010：fSAMPLING=fDTS/16，N=5 1011：fSAMPLING=fDTS/16，N=6 1100：fSAMPLING=fDTS/16，N=8 1101：fSAMPLING=fDTS/32，N=5 1110：fSAMPLING=fDTS/32，N=6 1111：fSAMPLING=fDTS/32，N=8
15	MOE	主输出使能

位/位域	名称	描述
		只要断路输入（BRK 或 BRK2）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：响应断路事件（2 个）。禁止 OC 和 OCN 输出。 响应断路事件或向 MOE 写入 0 时：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（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。 有关详细信息，请参见 OC/OCN 使能说明（捕获/比较使能寄存器(TIMx_CCER)）。
14	AOE	自动输出使能 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 BRK 和 BRK2 均无效）
13	BKP	断路极性 0：断路输入 BRK 为低电平有效 1：断路输入 BRK 为高电平有效 注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
12	BKE	断路使能 该位可使能完整的断路保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源）。

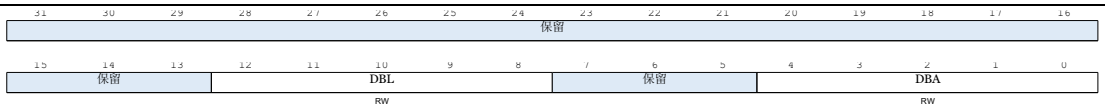
位/位域	名称	描述
		0：禁止断路功能 1：使能断路功能 注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
11	OSSR	运行模式下的关闭状态选择 此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。 有关详细信息，请参见 OC/OCN 使能说明（捕获/比较使能寄存器(TIMx_CCER)）。 0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。
10	OSSI	空闲模式下的关闭状态选择 当由于断路事件或软件写操作而使 MOE=0 时，此位作用于配置为输出的通道。 有关详细信息，请参见 OC/OCN 使能说明（捕获/比较使能寄存器 (TIMx_CCER)）。 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）。

位/位域	名称	描述
		1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。
9:8	保留	必须保持复位值
7:0	DTG	配置死区发生器 此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。 DTG[7:5]=0xx => DT=DTG[7:0]x tdtg，其中 tdtg=tDTS。 DTG[7:5]=10x => DT=(64+DTG[5:0])xtdtg，其中 Tdtg=2xtDTS。 DTG[7:5]=110 => DT=(32+DTG[4:0])xtdtg，其中 Tdtg=8xtDTS。 DTG[7:5]=111 => DT=(32+DTG[4:0])xtdtg，其中 Tdtg=16xtDTS。 示例：如果 TDTS=125ns (8MHz)，则可能的死区值为： 0 到 15875 ns（步长为 125 ns） 16 us 到 31750 ns（步长为 250 ns） 32 us 到 63us（步长为 1 us） 64 us 到 126 us（步长为 2 us）

24.4.22 TIMx DMA 控制寄存器（TIMx_DCR）

地址偏移：0x54

复位值：0x0000_0000



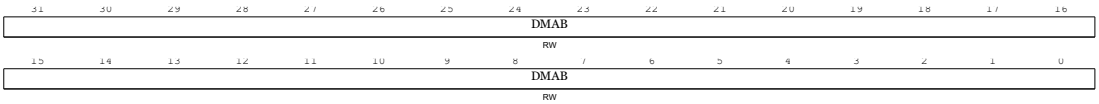
位/位域	名称	描述
31:13	保留	必须保持复位值
12:8	DBL	DMA 连续传送长度 该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。 00000：1 次传送 00001：2 次传送 00010：3 次传送 … 10001：18 次传送 示例：以下面的传送为例：DBL = 7 字节且 DBA = TIM2_CR1。 – 如果 DBL = 7 字节且 DBA = TIM2_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址： (TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL 在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。 在这种情况下，将向自以下地址开始的 7 个寄存器传送数据：(TIMx_CR1 地址) + DBA 根据 DMA 数据大小的配置，可能发生下面几种情况： – 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。

位/位域	名称	描述
		– 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。
7:5	保留	必须保持复位值
4:0	DBA	DMA 基址 该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。 示例： 00000：TIMx_CR1， 00001：TIMx_CR2， 00010：TIMx_SMCR， …

24.4.23 TIMx 全传输 DMA 地址寄存器（TIMx_DMAR）

地址偏移：0x58

复位值：0x0000_0000



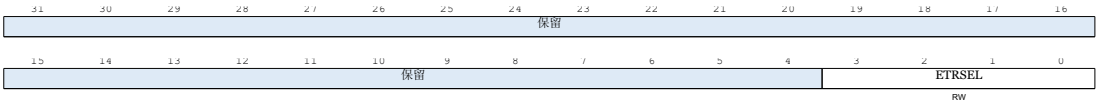
位/位域	名称	描述
31:0	DMAB	DMA 连续传送寄存器 对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

位/位域	名称	描述
		其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

24.4.24 TIMx 复用功能选项寄存器 1（TIMx_AF1）

地址偏移：0x5C

复位值：0x0000_0000



位/位域	名称	描述
31:4	保留	必须保持复位值
3:0	ETRSEL	ETR 源选择 这些位选择 ETR 输入源。 0000：ADC0 AMO 0001：ADC1 AMO 其它值：保留

25 通用定时器（TIM3、TIM4、TIM6、TIM7）

25.1 简介

通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动，支持递增、递减、中心计数、编码器模式等计数方式。

通用定时器具有 4 个独立通道，可实现测量输入信号的脉冲宽度、可编程 PWM 输出等功能。

25.2 主要特征

- 16 位递增、递减、递增/递减自动重装载计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成
  - 单脉冲模式输出
- 使用外部信号控制定时器，可实现多个定时器互联
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较

25.3 功能说明

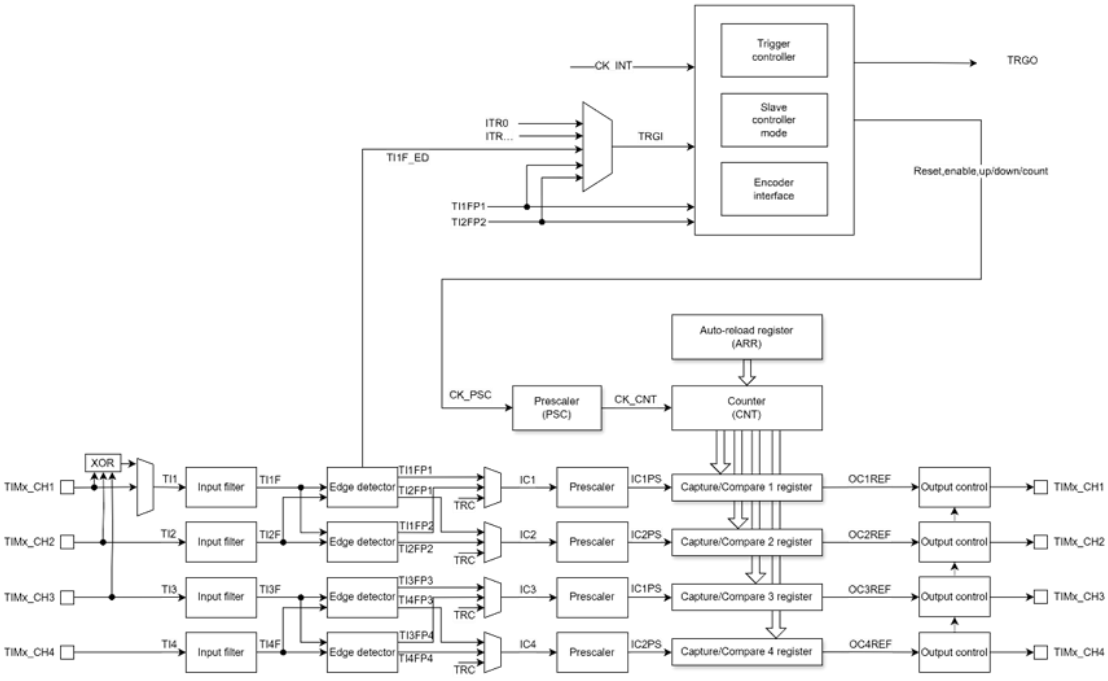


图 25.1 通用定时器结构框图

25.3.1 计数时钟

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚 TIx（x=1~6）
- 内部触发输入（ITRx）

25.3.1.1 内部时钟

内部时钟 CK_INT 来自定时器总线时钟。

如果禁止从模式控制器（SMS=0000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

使用内部时钟驱动计数器计数的操作步骤如下：

- 配置 TIMx_SMCR 寄存器的 SMS=0000，禁止从模式；
- 配置 TIMx_PSC 寄存器，设置计数器计数时钟频率；

- 配置 TIMx_ARR 寄存器，设置计数器计数周期；
- 配置 TIMx_CR1 寄存器的 CMS 位，以选择计数方式，若选择边沿对齐模式，则还需要配置 TIMx_CR1 寄存器的 DIR 位，以选择计数方向；
- 配置 TIMx_CR1 寄存器的 CEN 位，使能计数器。

25.3.1.2 外部时钟模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的触发信号（TRGI）出现上升沿或下降沿时计数。

使用外部时钟模式 1 驱动计数器计数的操作步骤如下：

- 配置 TIMx_SMCR 寄存器的 TS 位，选择触发信号（TRGI）的信号源；
  - TS=00100 时，选择 TI1 的边沿检测器（TI1F_ED）（即 TI1 经过滤波器后，信号的边沿检测信号）作为 TRGI；若选择该信号源，还需要配置 TIMx_CCMR1 寄存器的 IC1F 输入捕获 1 滤波器。
  - TS=00101 时，选择滤波器和边沿检测后的定时器输入 1（TI1FP1）作为 TRGI。若选择该信号源，还需要配置 TIMx_CCMR1 寄存器的 IC1F 输入捕获 1 滤波器和 TIMx_CCER 寄存器的 CC1P 和 CC1NP 配置捕获极性。
  - TS=00110 时，选择滤波器和边沿检测后的定时器输入 2（TI2FP2）作为 TRGI。若选择该信号源，还需要配置 TIMx_CCMR1 寄存器的 IC2F 输入捕获 2 滤波器和 TIMx_CCER 寄存器的 CC2P 和 CC2NP 配置捕获极性。
- 配置 TIMx_SMCR 寄存器的 SMS=0111，选择外部时钟模式 1；
- 配置 TIMx_PSC 寄存器，设置计数器计数时钟频率；
- 配置 TIMx_ARR 寄存器，设置计数器计数周期；
- 配置 TIMx_CR1 寄存器的 CMS 位，以选择计数方式，若选择边沿对齐模式，则还需要配置 TIMx_CR1 寄存器的 DIR 位，以选择计数方向；
- 配置 TIMx_CR1 寄存器的 CEN 位，使能计数器。

注：由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。

以下给出了要使递增计数器在 TI2 输入出现上升沿时计数的操作示例：

1. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=01 来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIMx_CCMR1 寄存器中写入 IC2F 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
3. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。（有关计数器的配置已省略）

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

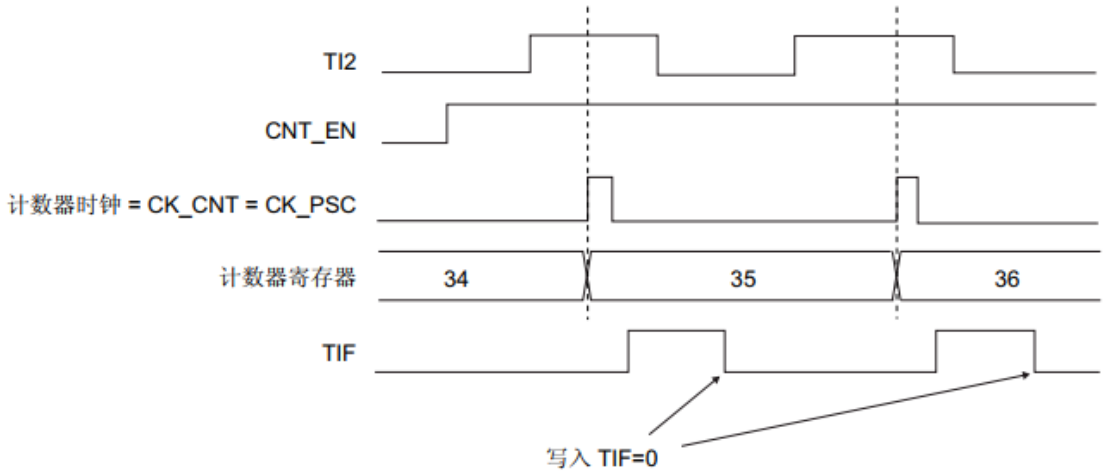


图 25.2 外部时钟模式 1 的计数器时序图

25.3.1.3 触发源说明

通用定时器的外部触发源和内部触发源连接详见下表：

表 25.1 通用定时器触发源列表

TIMx	内部触发源 0 ITR(0)	内部触发源 1 ITR(1)
TIM3	TIM4.TRGO	TIM2.TRGO
TIM4	TIM3.TRGO	TIM2.TRGO
TIM6	TIM7.TRGO	TIM5.TRGO
TIM7	TIM6.TRGO	TIM5.TRGO

25.3.2 时基单元

时基单元是定时器的核心单元，其包括一个 16 位的计数器、一个 16 位的预分频器、一个 16 位的自动重载计数器。计数器可根据设置进行递增计数、递减计数或中心计数（递增递减计数交替进行）。可编程预分频器模块为计数器提供计数时钟，实现计数频率控制。自动重载寄存器为计数器提供重载值。重复计数器可控制更新事件的产生时间。

25.3.2.1 预分频器

预分频器对时钟源进行分频，分频后的时钟作为计数器时钟。预分频系数由预分频寄存器（TIMx_PSC）设定。由于预分频寄存器具有缓冲功能（存在影子寄存器），因此预分频器可以实现实时更改，新的预分频比可以随时写入预分频寄存器中，只有在下一个更新事件 UEV 发生时才会被采用（发生 UEV 时写入影子寄存器中）。

预分频系数实时发生变化时，计数器的时序图如下。

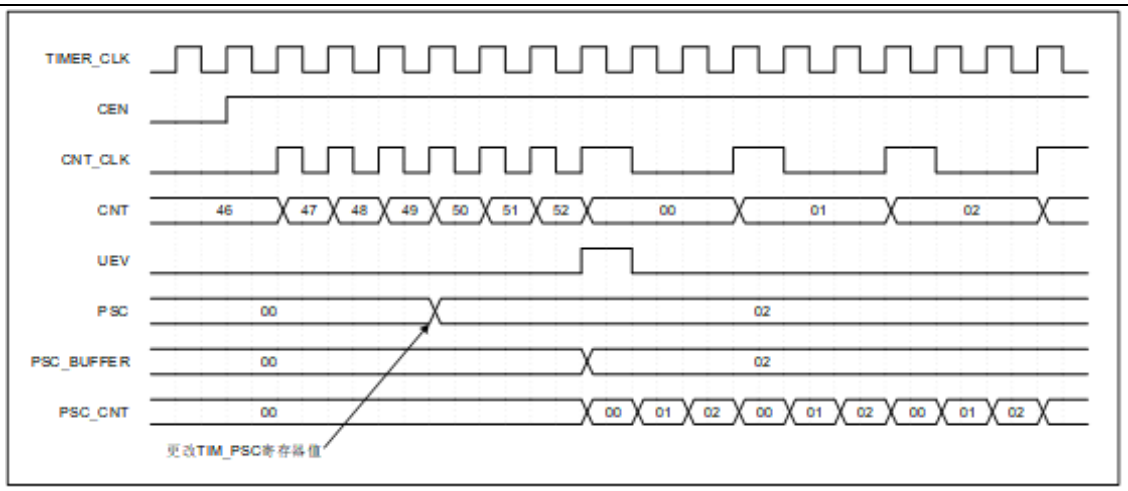


图 25.3 预分频系数由 1 变为 3 时的计数器时序图

25.3.2.2 自动重载寄存器

自动重载寄存器用于设置计数器计数的重载值，其自动重载功能由 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE）控制。预装载使能（ARPE=1）时，自动重载值将被缓存，并在下一个更新事件 UEV 发生时被采用（发生 UEV 时写入影子寄存器中）；预装载未使能（ARPE=0）时，自动重载值会被立即采用（立即写入影子寄存器中）。

25.3.2.3 计数器

25.3.2.3.1 递增计数模式

在递增计数模式下，计数器从 0 开始时递增计数到自动重载值（TIM_ARR 寄存器数值），一旦计数器计数到自动重载值，则会重新从 0 开始计数并产生计数器上溢事件。

下图分别为预分频系数为 1、2 时，递增计数模式下计数器时序图。

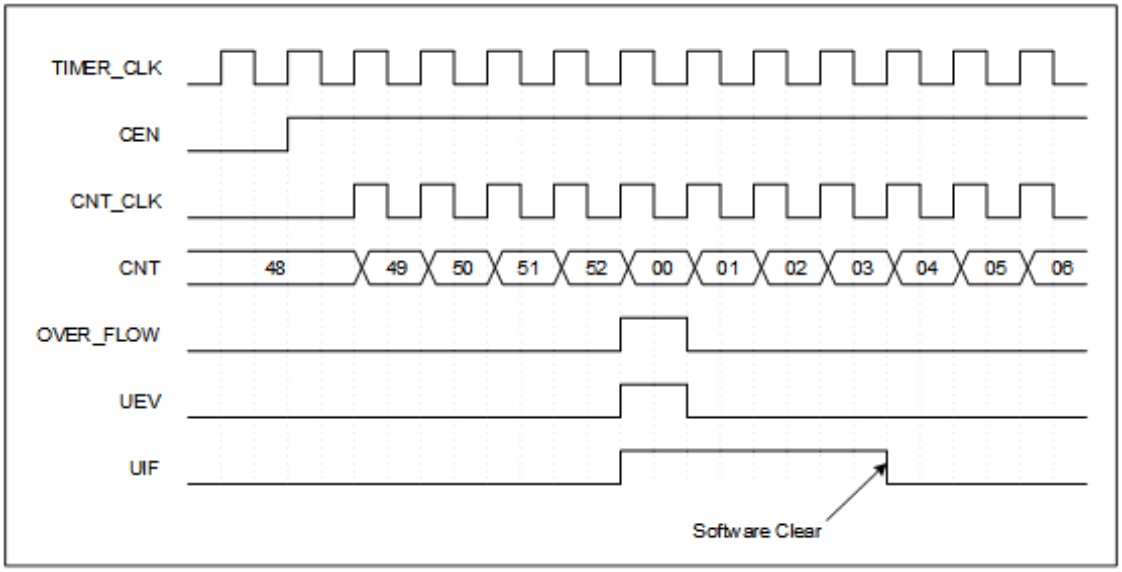


图 25.4 递增计数时序图（1 分频）

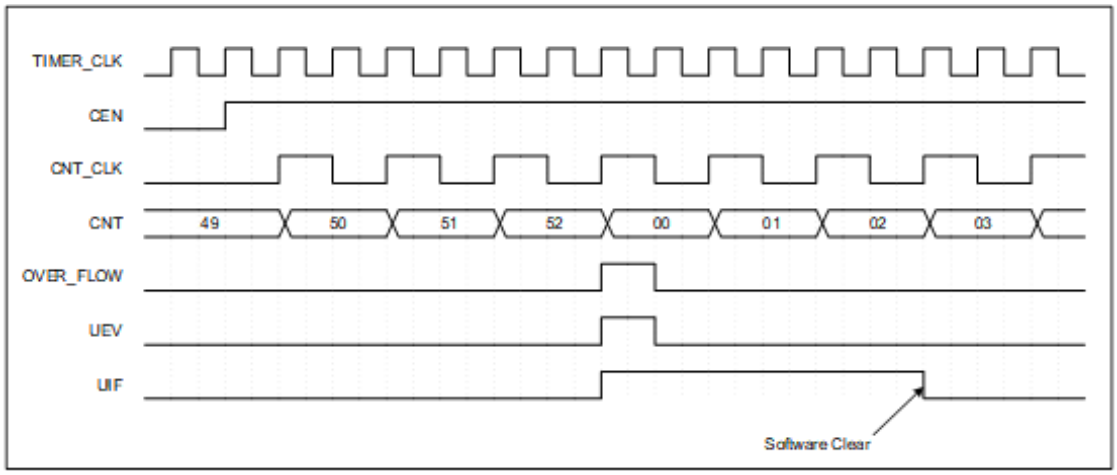


图 25.5 递增计数时序图（2 分频）

预装载未使能（ARPE=0）时，计数器在递增计数过程中，改变 TIM_ARR 寄存器值的时序图如下图所示。

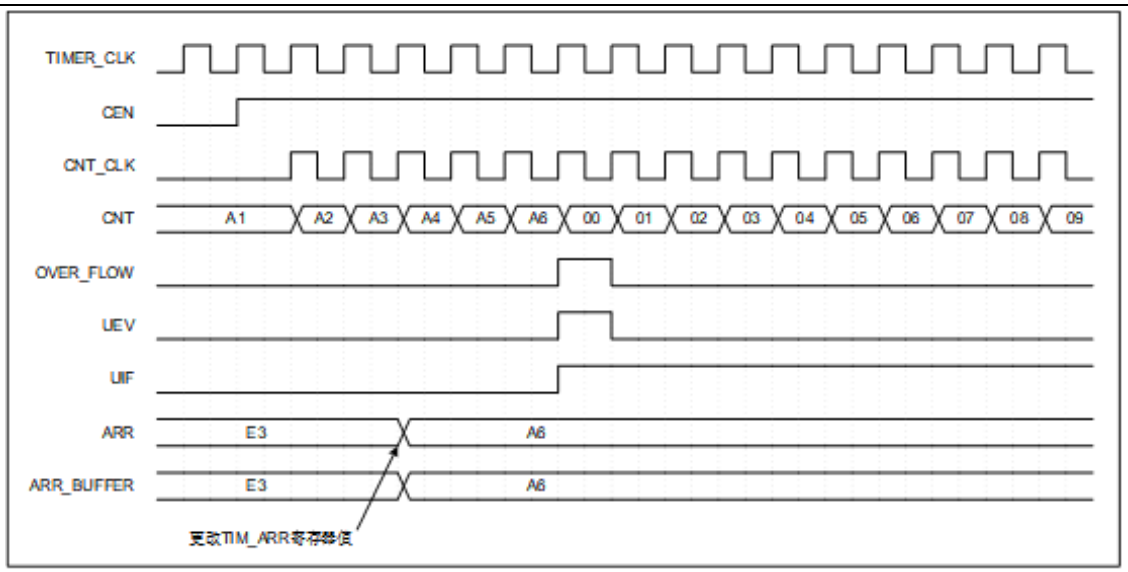


图 25.6 计数器时序图（ARPE=0 时，递增计数时改变 TIM_ARR 寄存器值）

预装载使能（ARPE=1）时，计数器在递增计数过程中，改变 TIM_ARR 寄存器值的时序图如下图所示。

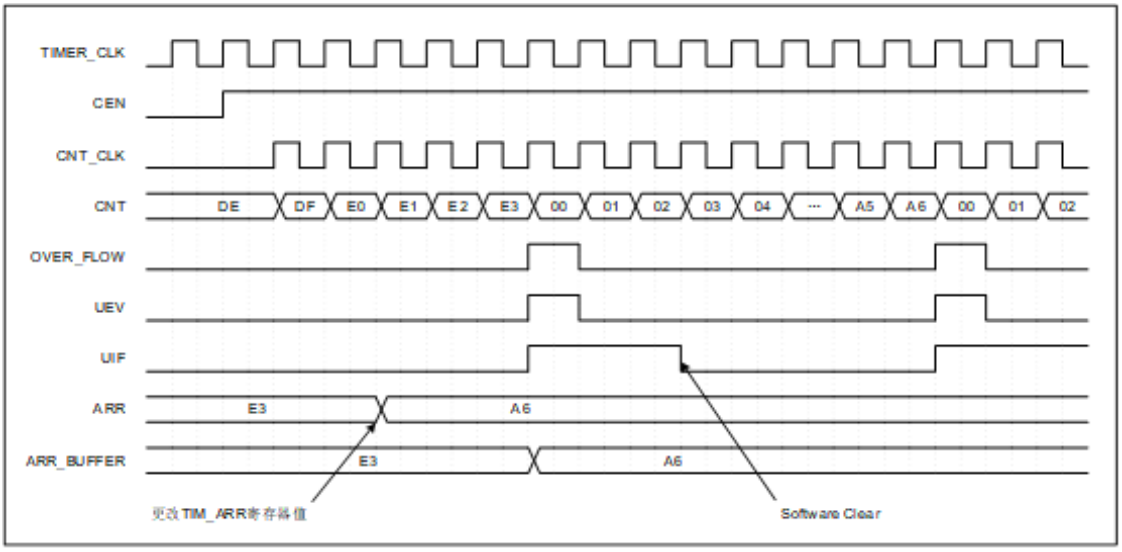


图 25.7 计数器时序图（ARPE=1 时，递增计数时改变 TIM_ARR 寄存器值）

25.3.2.3.2 递减计数模式

在递减计数模式下，计数器从自动重载值（TIM_ARR 寄存器数值）开始时递减计数到 0，一旦计数器计数到 0，则会重新从自动重载值开始计数并产生计数器下溢事件。

下图分别为预分频系数为 1、2 时，递减计数模式下计数器时序图。

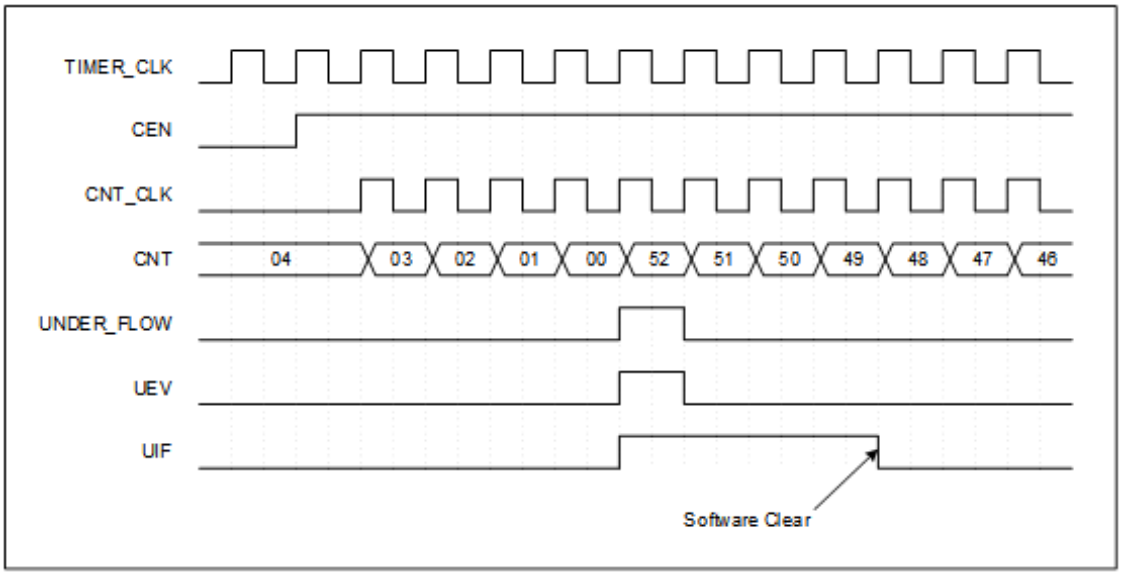


图 25.8 递减计数时序图（1 分频）

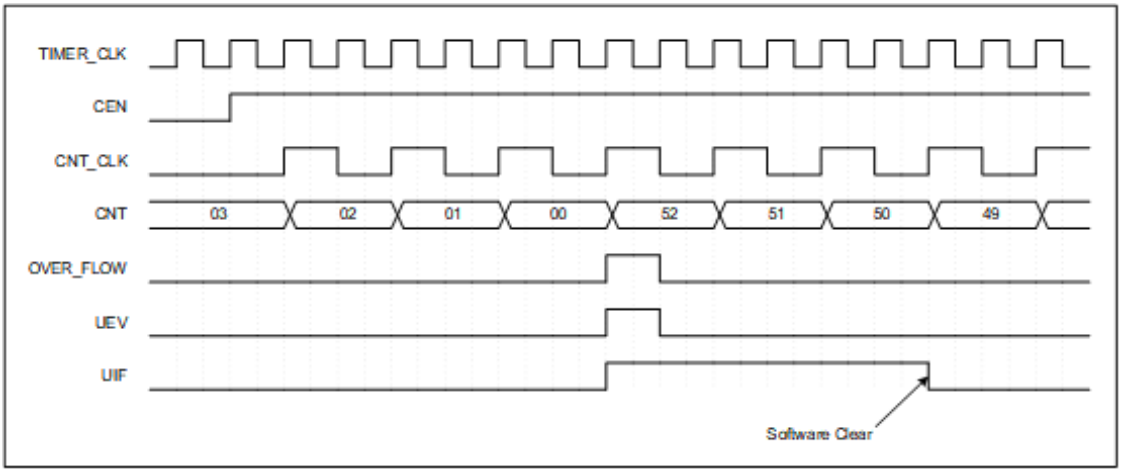


图 25.9 递减计数时序图（2 分频）

预装载未使能（ARPE=0）时，计数器在递减计数过程中，改变 TIM_ARR 寄存器值的时序图如下图所示。

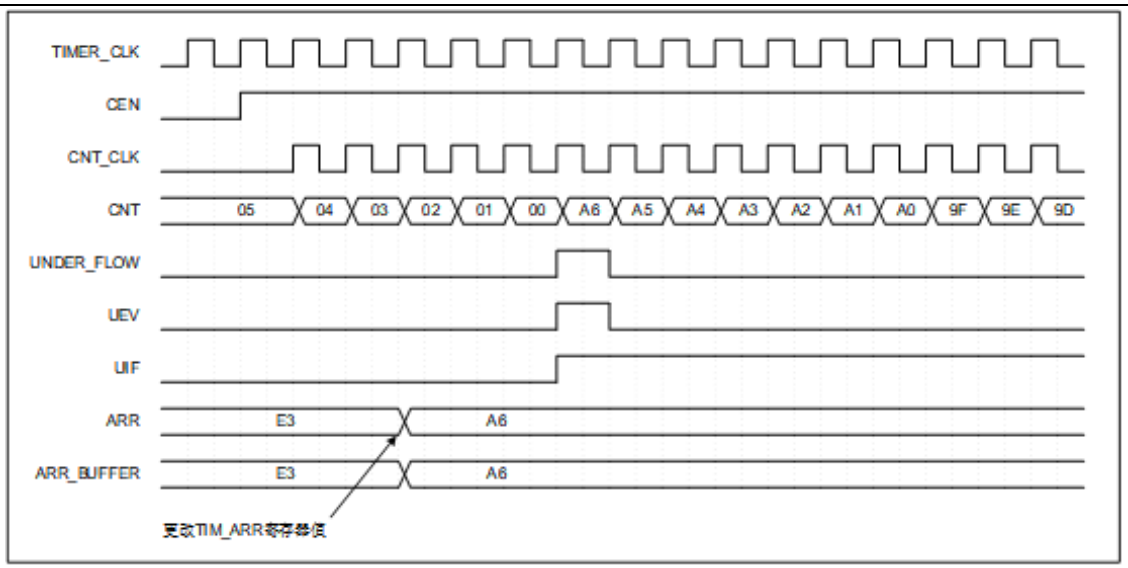


图 25.10 计数器时序图（ARPE=0 时，递减计数时改变 TIM_ARR 寄存器值）

预装载使能（ARPE=1）时，计数器在递减计数过程中，改变 TIM_ARR 寄存器值的时序图如图下图所示。

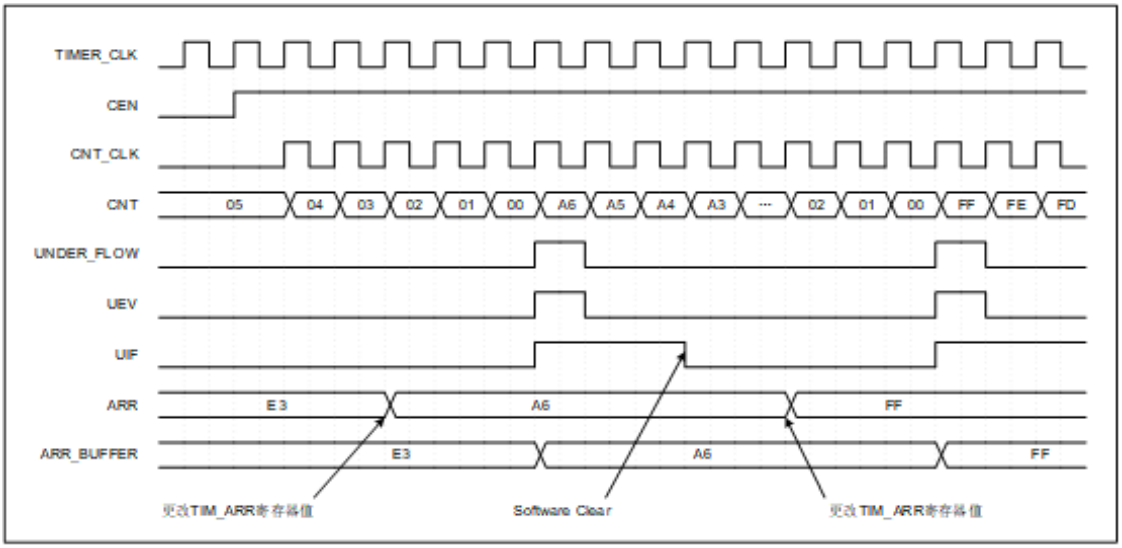


图 25.11 计数器时序图（ARPE=1 时，递减计数时改变 TIM_ARR 寄存器值）

25.3.2.3.3 递增/递减计数模式（中心对齐模式）：

在中心对齐模式下，计数器交替进行递增、递减计数，从 0 开始递增计数到自动重载值，然后再递减计数到 0。在递增计数中，计数器计数到自动重载值-1（TIM_ARR-1）时，会产生一个上溢事件；在递减计数中，计数器计数到 1 时，会产生一个下溢事件。在中心对齐模式下，TIM_CR1 寄存器的方向位（DIR）是只读的，计数方向由硬件控制。

下图为在中心对齐模式下的计数器的时序图。

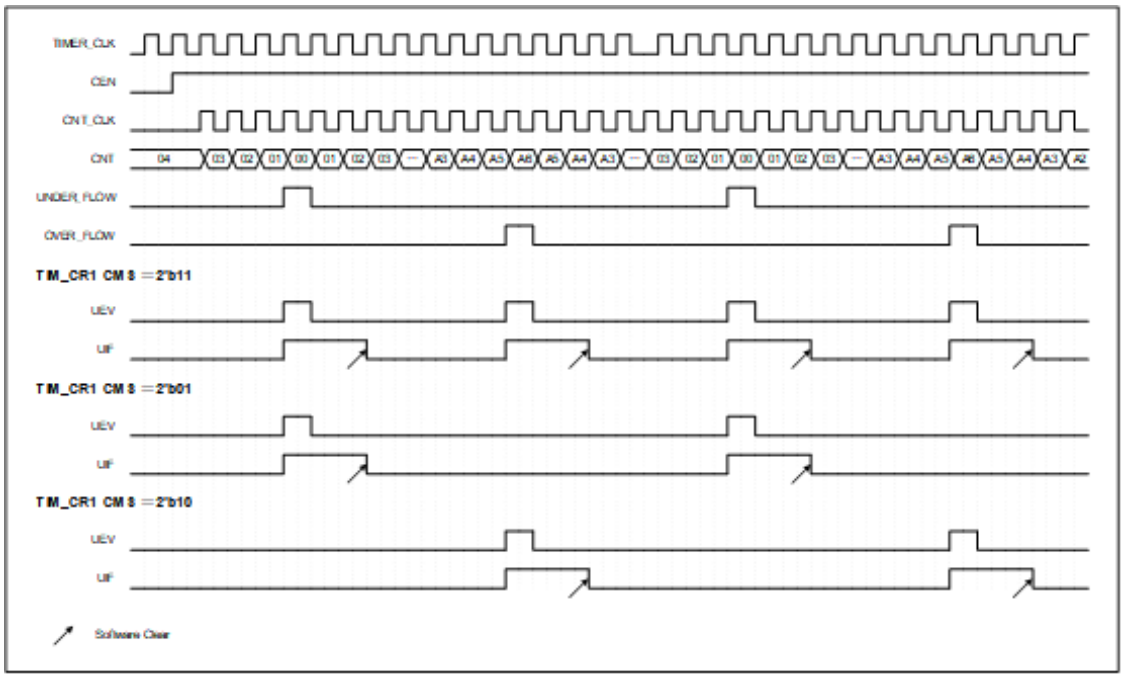


图 25.12 中心对齐模式，计数器时序图

25.3.2.3.4 编码器模式

编码器模式下需提供两组输入信号 TIMx_CH1 和 TIMx_CH2，根据一组输入信号电平值，计数器在另一组输入信号边沿向上或向下计数。计数方向由 DIR 值指示。

- 编码器模式 1：SMS=0001,计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
- 编码器模式 2：SMS=0010,计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
- 编码器模式 3：SMS=0011,计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。

若要使用编码器模式可按下面步骤配置：

- 配置 TIMx_CC1M 寄存器的 IC1F 位，设置通道 1 输入信号滤波；配置 TIMx_CCER 寄存器的 CC1P 位，设置通道 1 输入信号有效电平。
- 配置 TIMx_CC1M 寄存器的 IC2F 位，设置通道 2 输入信号滤波；配置 TIMx_CCER 寄存器的 CC2P 位，设置通道 2 输入信号有效电平。

- 配置 TIMx_CC1M 寄存器的 CC1S 位，设置通道 1 为输入模式；配置 TIMx_CC1M 寄存器的 CC2S 位，设置通道 2 为输入模式；
- 配置 TIMx_SMCR 寄存器的 SMS 位，选择编码器模式 1、编码器模式 2 或编码器模式 3;
- 配置 TIMx_ARR 寄存器的 ARR 位，设置计数器自动重装载值；
- 配置 TIMx_CR1 寄存器的 CEN 位，使能计数器。

编码模式下计数器计数方向如下表所示：

表 25.2 编码器模式下信号与计数器方向关系表

有效边沿	计数边沿相对信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

下图以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=01（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。
- CC1P=0，CC1NP=0（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=0，CC2NP=0（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2=TI2）。

- SMS=0011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=1（TIMx_CR1 寄存器，使能计数器）。

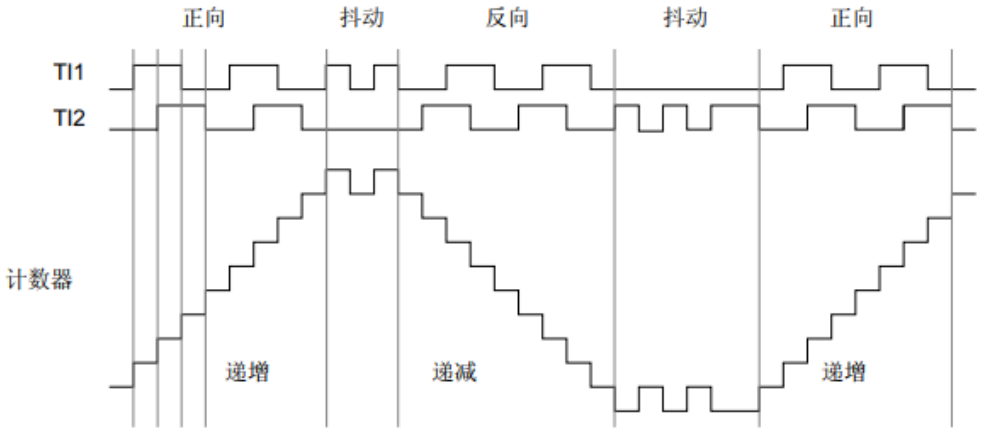


图 25.13 编码器模式下计数器工作示意图

25.3.2.3.5 更新事件 UEV 产生

更新事件 UEV 是否产生由更新禁止 UDIS 位控制（TIM_CR1 寄存器）。UDIS=0 时，使能 UEV，允许产生更新事件 UEV；UDIS=1 时，禁止 UEV，不会产生更新事件 UEV。

使能时，发生计数器上溢或下溢、生成更新 UG=1（TIM_EGR 寄存器）时，将会产生更新事件 UEV，并且更新影子寄存器的值（自动重载影子寄存器和预分频影子寄存器）。

禁止时，不会产生更新事件，影子寄存器保持不变。只有在 UG 置 1 时，初始化计数器和预分频器。

25.3.2.3.6 更新中断 UIF 产生

更新中断 UIF 的事件源由更新请求源 URS 位控制（TIM_CR1 寄存器）。URS=0 时，发生计数器上溢或下溢、UG 位置 1 时都会产生更新中断 UIF；URS=0 时，只有发生计数器上溢或下溢才会产生更新中断 UIF。

25.3.3 输入捕获

通用定时器拥有 4 个独立通道，每个通道可配置为输入捕获或输出比较模式。下图展示了通道输入部分的主要电路。

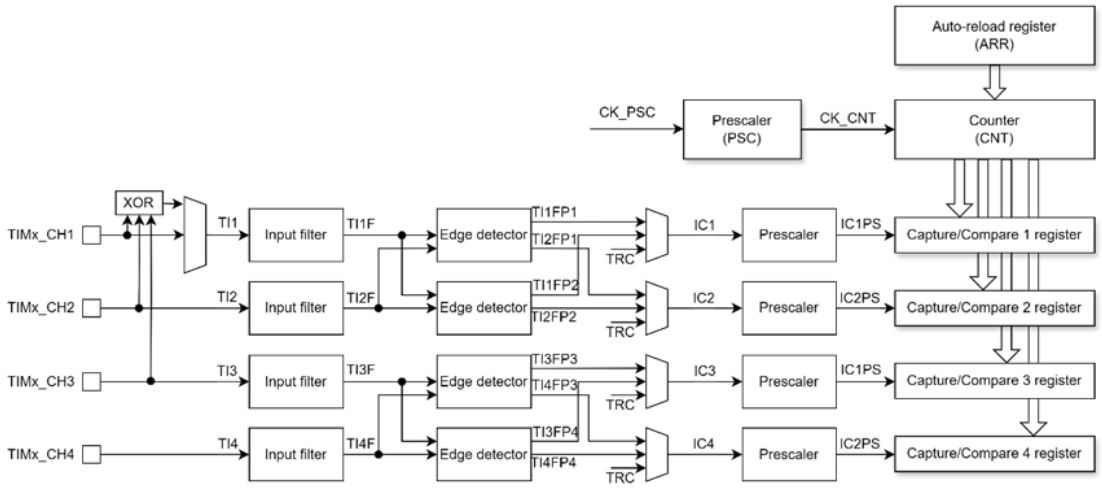


图 25.14 输入捕获电路图

（1）输入通道：

待测信号从定时器的外部引脚 TIMx_CH1-TIMx_CH6 进入，经过预处理输出 TIx。 其中，TI1 可通过 TIMx_CR2 的 TI1S 选择由 TIMx_CH1 输入或选择 TIMx_CH1、TIMx_CH2、TIMx_CH3 异或。TI2-TI6 则分别连接 TIMx_CH2~TIMx_CH6。

（2）输入滤波器：

TIx 信号经过数字滤波器，输出滤波后的信号 TIxF。 输入滤波器内置两级同步器，输入信号经过同步器被定时器时钟信号同步后，输入至数字滤波器。当输入的信号存在高频干扰的时候，可通过配置 TIMx_CR1 寄存器的 CKD 位和 TIMx_CCMR1/2/3 寄存器的 ICxF 位，对输入信号 TIx 进行滤波处理，即进行重新采样，根据采样定律，采样的频率必须大于等于两倍的输入信号。 通过 ICxF 位介绍可知，采样频率 fSAMPLE 可由 fCK_INT 和 fDTS 分频后的时钟提供，其中 fCK_INT 是内部时钟，即定时器总线时钟；fDTS 是由 fCK_INT 分频后产生，分频系数由 CKD 位进行配置。

（3）边沿检测器：

TIxF 信号经过边沿检测器，输出边沿选择后的信号 TIxFPx。 边沿检测器可通过配置 TIMx_CCER 寄存器的 CCxP 和 CCxNP 位，选择在输入信号的上升沿、下降沿或双边沿发生捕获事件。 以 TI1FPx 为例，TI1FP1 表示来自通道 1 的 TI1F

信号经过通道 1 的边沿检测器（即由 CC1P 和 CC1NP 位共同控制）处理后的信号；TI1FP2 表示来自通道 1 的 TI1F 信号经过通道 2 的边沿检测器（即由 CC2P 和 CC2NP 位共同控制）处理后的信号。

（4）捕获信号选择器：

TIxFPx 信号和 TRC 信号经过输入捕获信号选择器，输出选择后的信号 ICx。捕获信号选择器可通过配置 TIMx_CCMR1/2/3 寄存器的 CCxS 位，选择捕获信号。 以通道 1 为例，CC1S=01 时，IC1 选择 TI1FP1；CC1S=10 时，IC1 选择 TI2FP1；CC1S=11 时，IC1 选择 TRC。

（5）预分频器：

ICx 信号经过预分频器，输出预分频后的信号 ICxPS。 预分频器可通过配置 TIMx_CCMR1/2/3 寄存器的 ICxPSC 位，对 ICx 信号进行预分频，即选择发生多少个事件后进行一次捕获，若希望捕获信号的每一个边沿，则不分频。

（6）捕获寄存器：

输入捕获通道最终以经过预分频后的信号 ICxPS 进行捕获，当发生捕获时，计数器 CNT 的数值将会被锁存到捕获寄存器 CCRx 中。

25.3.3.1 输入模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器（TIMx_CCRx）来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入 0，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。

- 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 fDTS 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注：通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。

25.3.3.2 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入 0（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=10（下降沿有效）。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入 1。

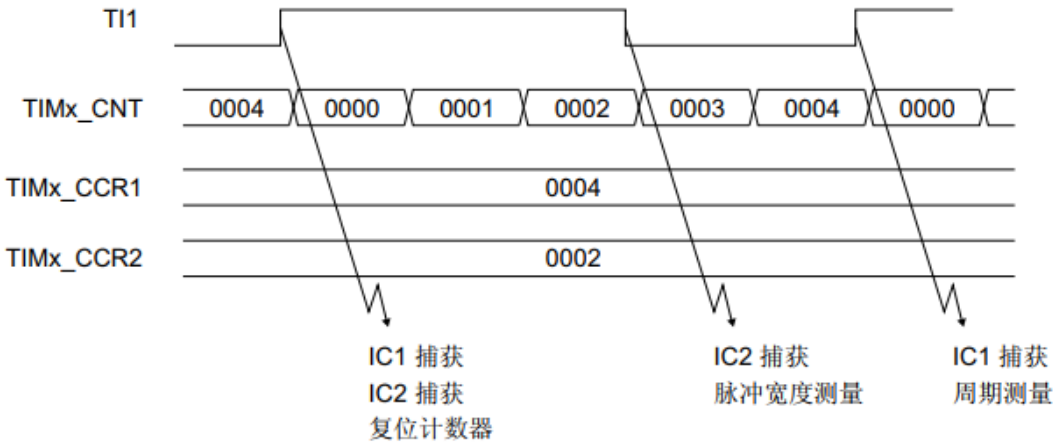


图 25.15 PWM 输入模式时序图

25.3.3.3 输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔，如下图所示。

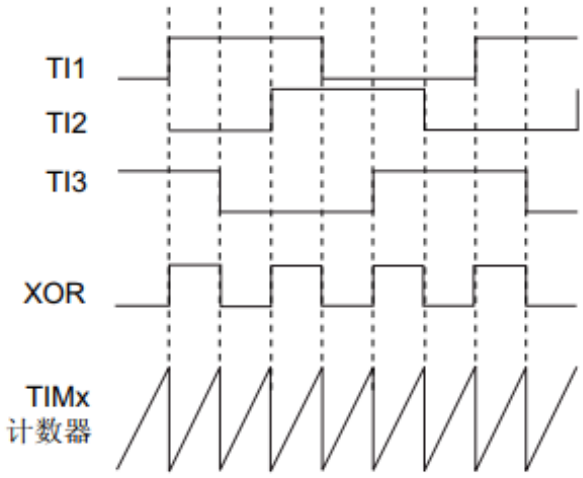


图 25.16 测量 3 输入信号上边沿的时间间隔示意图

25.3.4 输出比较

下图展示了通道输出部分的主要电路。

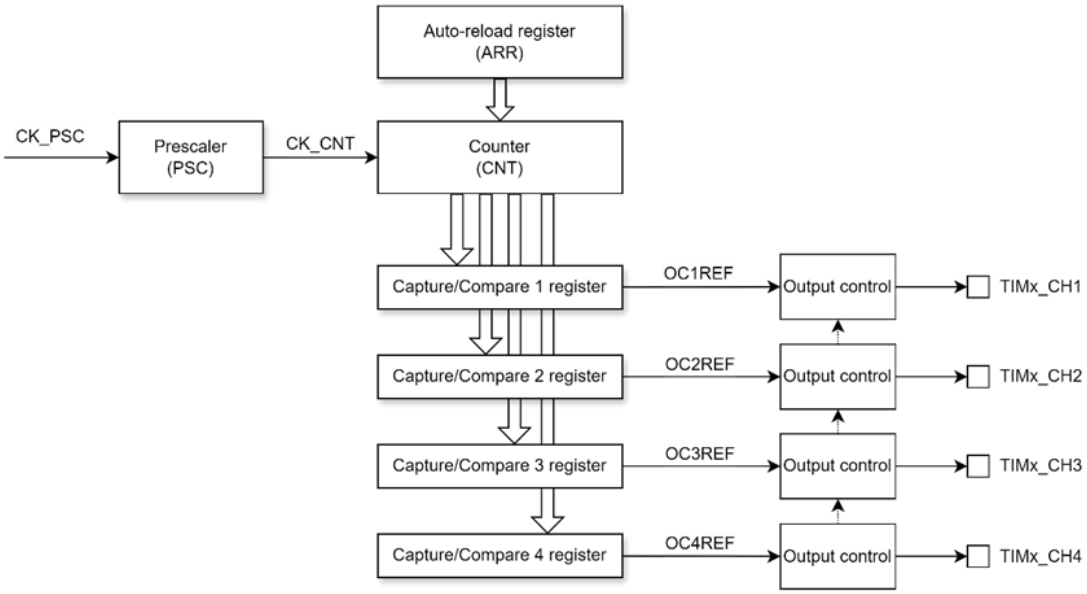


图 25.17 输出比较电路图

（1）比较寄存器

比较寄存器的主要功能是，把计数器 CNT 的数值与比较寄存器 CCRx 的值进行比较，根据 TIMx_CCMR1/2/3 寄存器的 OCxM 位的配置，输出参考信号 OCxREF，其中 OCxREF=1（高电平）称为有效电平，OCxREF=0（低电平）称为无效电平。 在使能比较中断时，比较寄存器还会产生中断信号，相应的标志位 TIMx_SR 寄存器的 CCxIF 位置 1。

（2）输出控制器

输出控制器的主要功能是，根据配置的输出模式，控制信号的极性反转，输出信号 OCx。 进入输出控制器的信号将会根据 TIMx_CCER 寄存器的 CCxP 的配置，进行极性选择。经过极性选择的信号是否由 OCx 引脚输出至外部引脚，则由 TIMx_CCER 寄存器的 CCxE 位进行控制。

25.3.4.1 输出模式

通道输出比较模式由 CCxS 位进行控制，当 CCxS=00 时，该通道被配置为输出比较模式。

25.3.4.1.1 输出逻辑

输出比较各模式描述如下：

输出信号 OCx 的有效电平取决于 CCxP 位。CCxP=0 时，OCx 的有效电平为高电平；CCxP=1 时，OCx 的有效电平为低电平。

表 25.3 输出模式介绍表

通道状态	输出形式	OCxM[3:0]	描述
输出	冻结	0000	时基： 捕获/比较寄存器（CCRx）与计数器（CNT）进行比较不会对输出信号 OCx 造成任何影响
	比较输出	0001	匹配时设置为有效电平： 当计数器（CNT）与捕获/比较寄存器（CCRx）匹配时，输出信号 OCx 强制变为有效电平
		0010	匹配时设置为无效电平： 当计数器（CNT）与捕获/比较寄存器（CCRx）匹配时，输出信号 OCx 强制变为无效电平
		0011	匹配时翻转： 当计数器（CNT）与捕获/比较寄存器（CCRx）匹配时，输出信号 OCx 发生翻转
	强制输出	0100	输出信号 OCx 强制变为有效电平
		0101	输出信号 OCx 强制变为无效电平
	PWM 输出	0110	PWM 模式 1： 在递增计数时，若 CNT＜CCRx，则输出有效电平，否则输出无效电平 在递减计数时，若 CNT＞CCRx，则输出无效电平，否则输出有效电平
		0111	PWM 模式 2：

通道状态	输出形式	OCxM[3:0]	描述
			在递增计数时，若 CNT＜CCRx，则输出无效电平，否则输出有效电平 在递减计数时，若 CNT＞CCRx，则输出有效电平，否则输出无效电平
	可再触发 OPM	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有效状态。 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。
	组合 PWM	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。

通道状态	输出形式	OCxM[3:0]	描述
		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。
	不对称 PWM	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。
	单脉冲模式	单脉冲模式为一种特殊的输出比较模式。 当配置为单脉冲模式时，计数器在发生更新事件时，停止计数（CEN 位清零）	

捕获/比较通道的中断触发条件如下表所示。

除中心对齐模式外，计数器 CNT 与比较值 CCRx 匹配时，CCxIF 比较中断标志置 1，并产生对应中断。若比较值 CCRx 大于自动重载值 ARR，则 CCxIF 位将在计数器发生上溢（递增计数模式和中心对齐模式）或下溢（递减计数模式）时置 1。

表 25.4 捕获/比较通道中断产生条件

CCx 通道模式	CCxIF 中断条件		
输入捕获	发生输入捕获事件		
		CCRx≤ARR	CNT=CCRx

CCx 通道模式	CCxIF 中断条件		
输出比较	边沿对齐模式（CMS=00）	CCRx＞ARR	dir=0（递增计数），计数器发生上溢事件
			dir=1（递减计数），计数器发生下溢事件
	中心对齐模式 1（CMS=01）	CCRx≤ARR	dir=1（递减计数），CNT=CCRx
		CCRx＞ARR	计数器发生上溢事件
	中心对齐模式 2（CMS=10）	CCRx≤ARR	dir=0（递增计数），CNT=CCRx
		CCRx＞ARR	计数器发生上溢事件
	中心对齐模式 3（CMS=11）	CCRx≤ARR	CNT=CCRx
		CCRx＞ARR	计数器发生上溢事件

比较输出模式：

下图为三种输出比较模式（匹配时置高电平/低电平/翻转）的时序图。

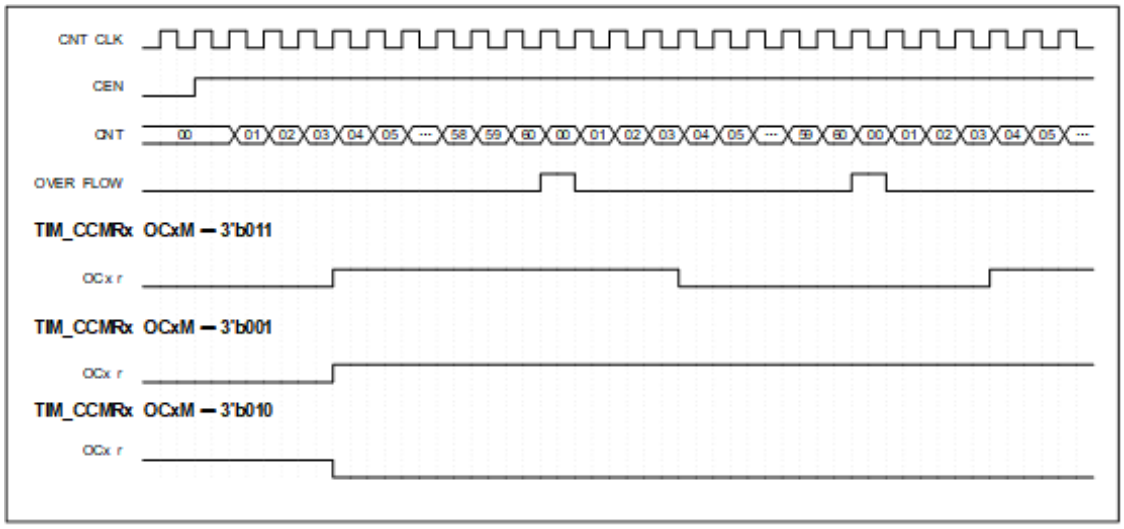


图 25.18 三种输出比较模式时序图

PWM 输出模式：

下图为递增计数模式下，输出 PWM 波形的时序图。在 PWM1 模式下，递增计数时，若 CNT＜CCRx，则输出无效电平；若 CNT≤CCRx，则输出有效电平。在 PWM2 模式下，递增计数时，若 CNT＜CCRx，则输出无效电平；若 CNT≥CCRx，则输出有效电平。

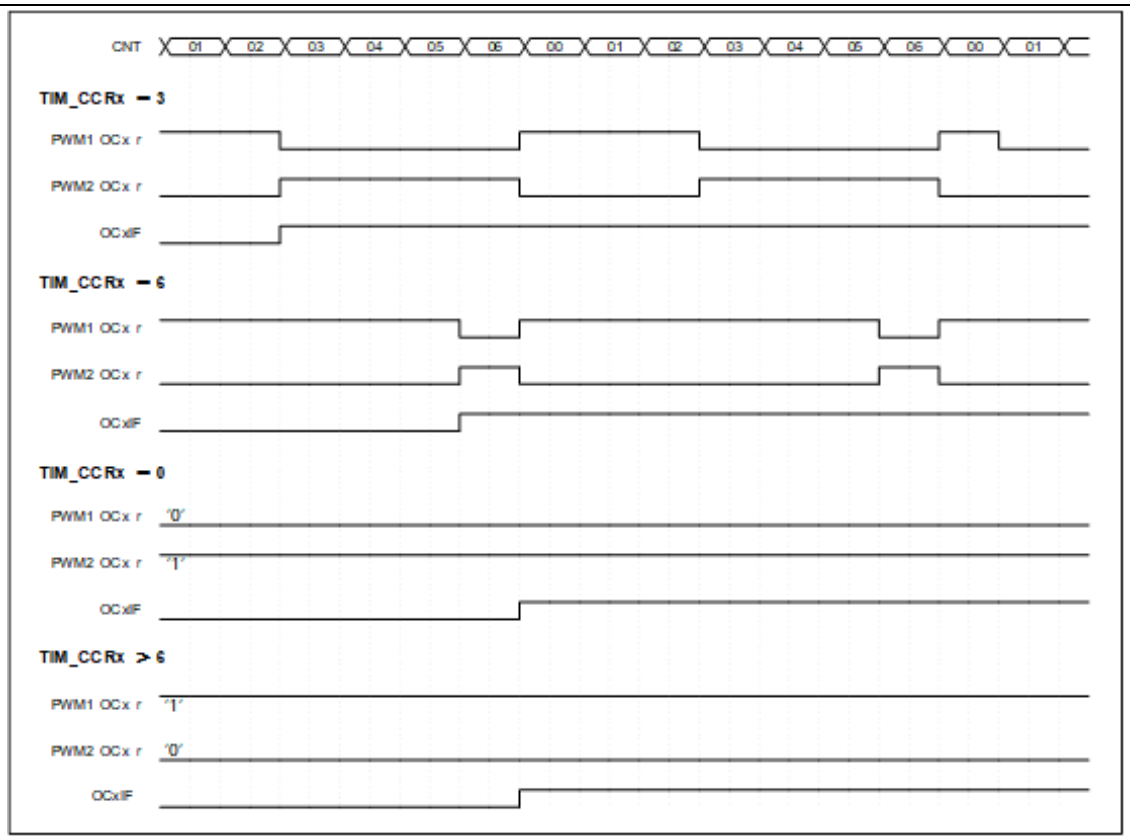


图 25.19 递增计数模式，输出 PWM 波形（ARR=06）

下图为递减计数模式下，输出 PWM 波形的时序图。在 PWM1 模式下，递减计数时，若 CNT＞CCRx，则输出无效电平；若 CNT≤CCRx，则输出有效电平。在 PWM2 模式下，递减计数时，若 CNT＞CCRx，则输出有效电平；若 CNT≤CCRx，则输出无效电平。

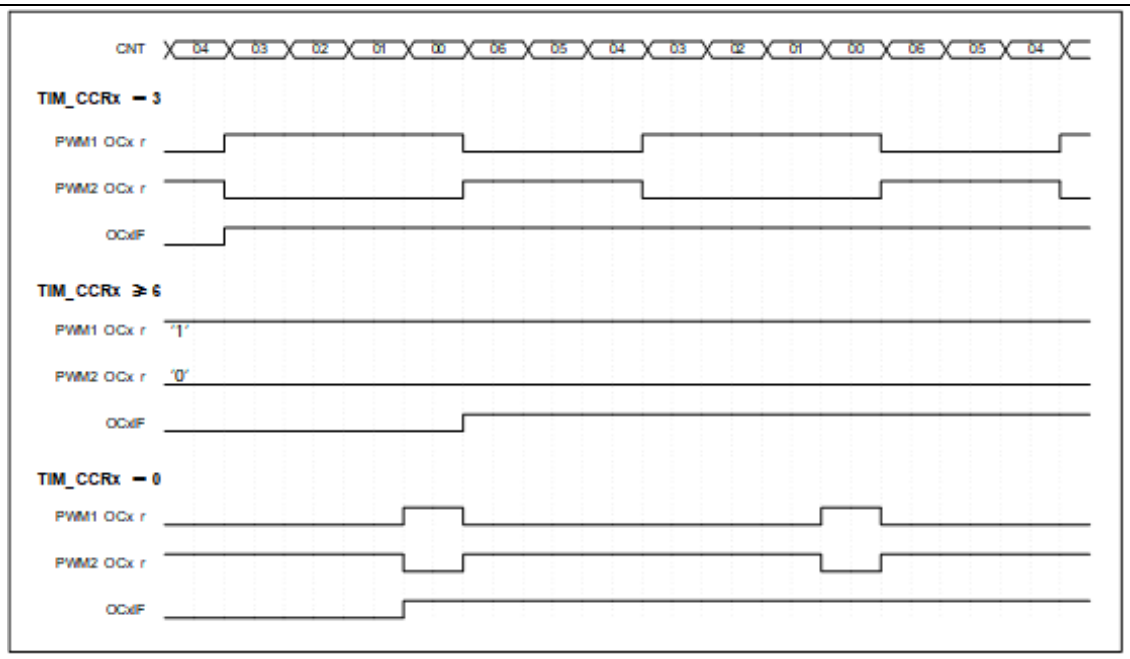


图 25.20 递减计数模式，输出 PWM 波形（ARR=06）

下图为中心对齐模式下，输出 PWM 波形的时序图。

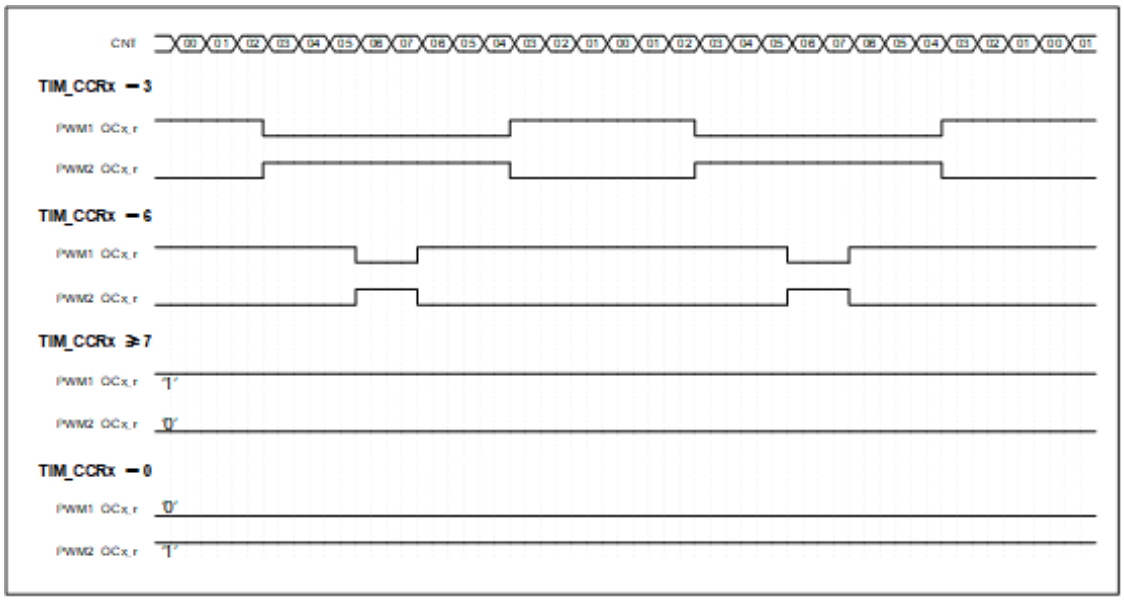


图 25.21 中心对齐模式，输出 PWM 波形（ARR=07）

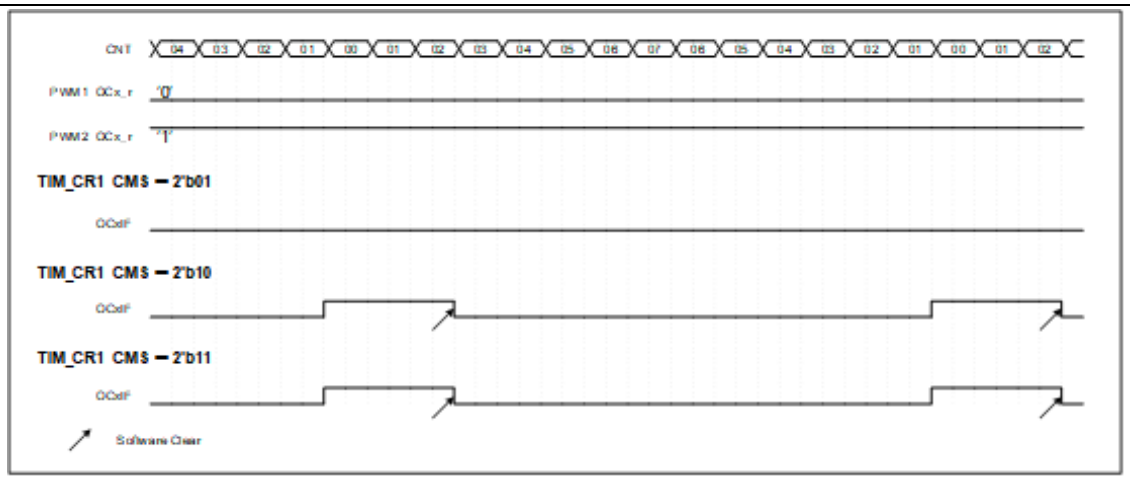


图 25.22 中心对齐模式，中断标志时序图（ARR=07、CCRx=00）

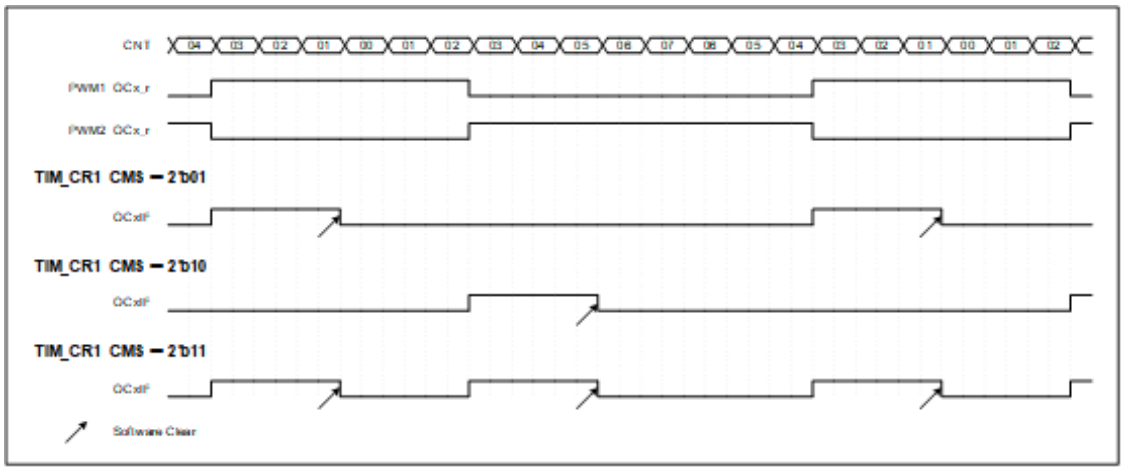


图 25.23 中心对齐模式，中断标志时序图（ARR=07、CCRx=03）

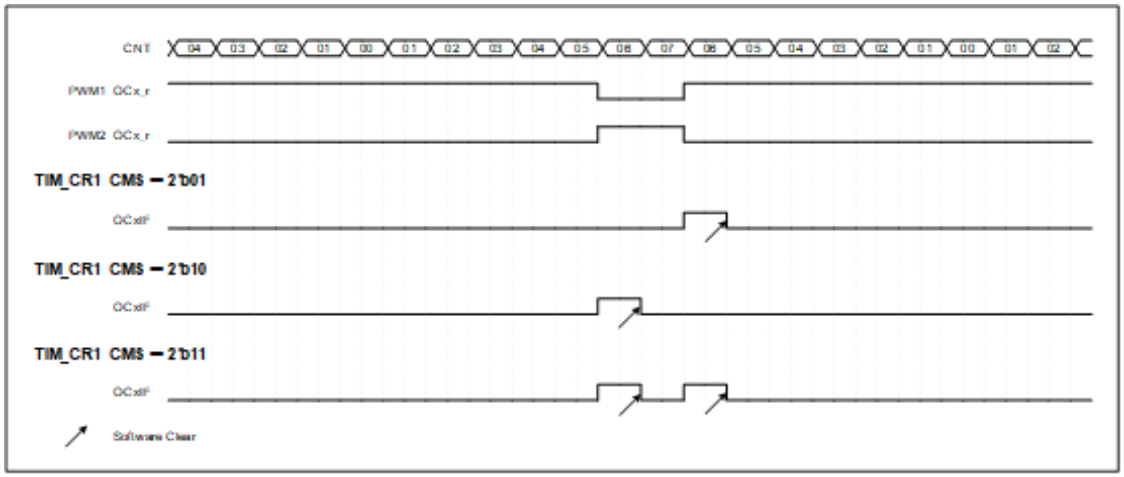


图 25.24 中心对齐模式，中断标志时序图（ARR=07、CCRx=06）

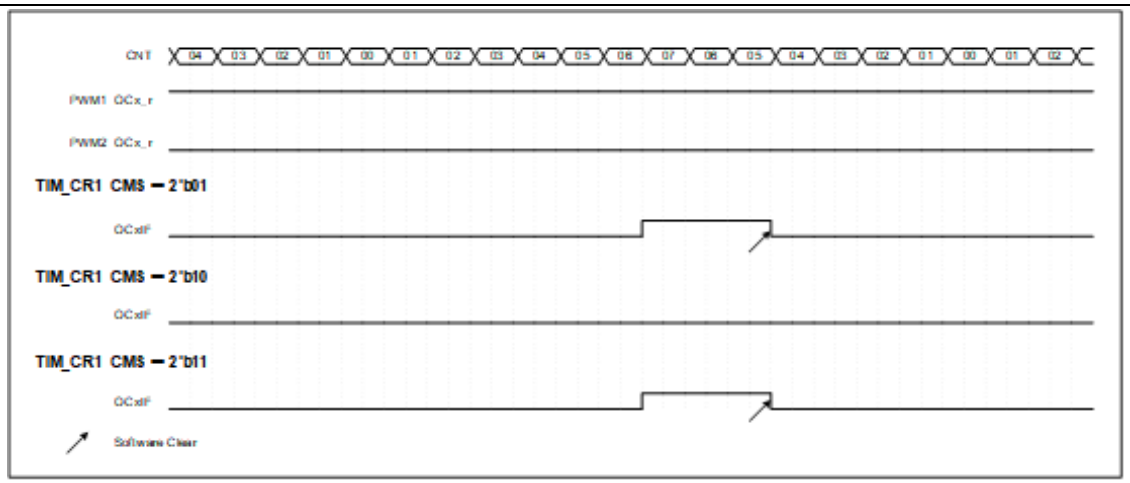


图 25.25 中心对齐模式，中断标志时序图（ARR=07、CCRx=07）

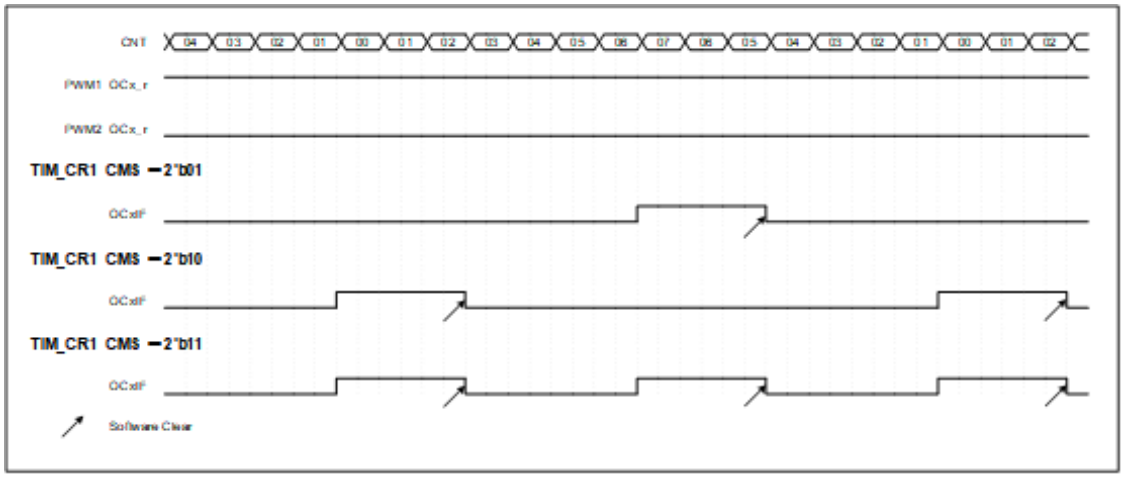


图 25.26 中心对齐模式，中断标志时序图（ARR=07、CCRx＞07）

不对称 PWM 模式：

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入 1110（不对称 PWM 模式 1）或 1111（不对称 PWM 模式 2）。

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

下图显示了不对称 PWM 模式下可以产生的信号示例。

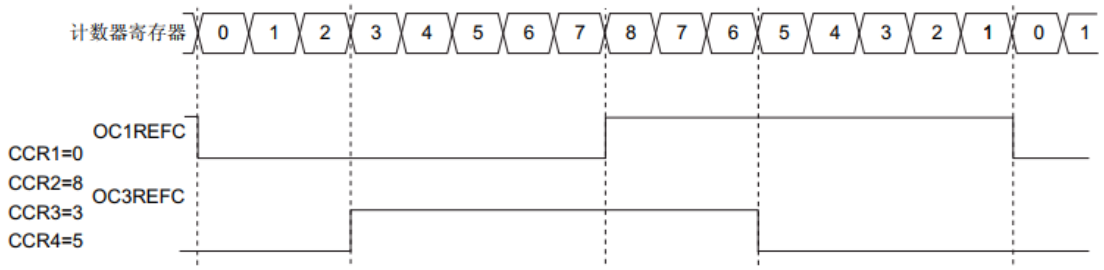


图 25.27 50% 占空比时产生的 2 个相移 PWM 信号

组合 PWM 模式：

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入 1100（组合 PWM 模式 1）或 1101（组合 PWM 模式 2）。

当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

下图显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。

- 通道 3 在组合 PWM 模式 2 下配置。
- 通道 4 在 PWM 模式 1 下配置。

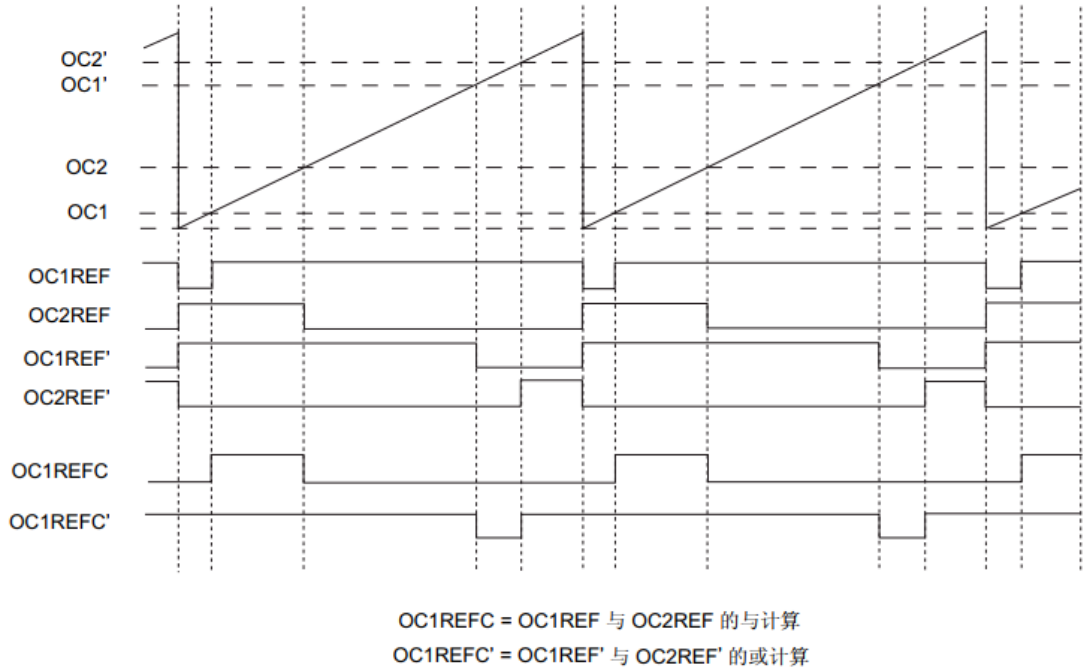


图 25.28 通道 1 和通道 3 上的组合 PWM 模式

25.3.4.1.2 比较寄存器预装载

比较寄存器具有预装载功能，其预装载功能由 TIM_CCMRx 寄存器的 OCxPE 位进行控制。预装载使能时（OCxPE=1）时，CCRx 寄存器进行缓冲，CCRx 可随时被写入，但其预装载值将会在每次发生更新事件时装载到对应的影子寄存器中，并用于比较；预装载未使能（OCxPE=0）时，CCRx 寄存器不进行缓冲，CCRx 可随时被写入，且写入后立即装载到影子寄存器中，并立即被用于比较。

下图给出了输出比较在预装载使能和未使能时的不同的时序图。当 OCxPE=0 时，输出比较预装载未使能，此时修改 CCRx 寄存器值，比较值将会被立即缓存并生效。当 OCxPE=1 时，输出比较预装载使能，此时修改 CCRx 寄存器值，比较值不会立即生效，而是等待计数器产生更新事件时才会进行缓存并生效。

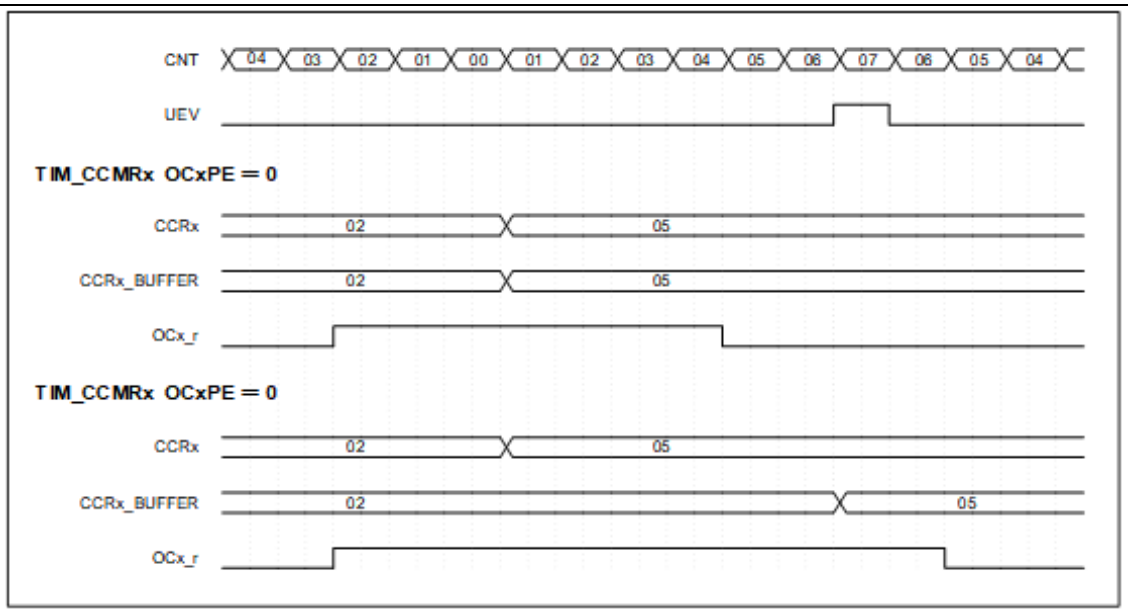


图 25.29 输出比较预装载时序图

25.3.5 定时器同步

25.3.5.1 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

25.3.5.1.1 从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP='0'，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。

- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，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，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。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。

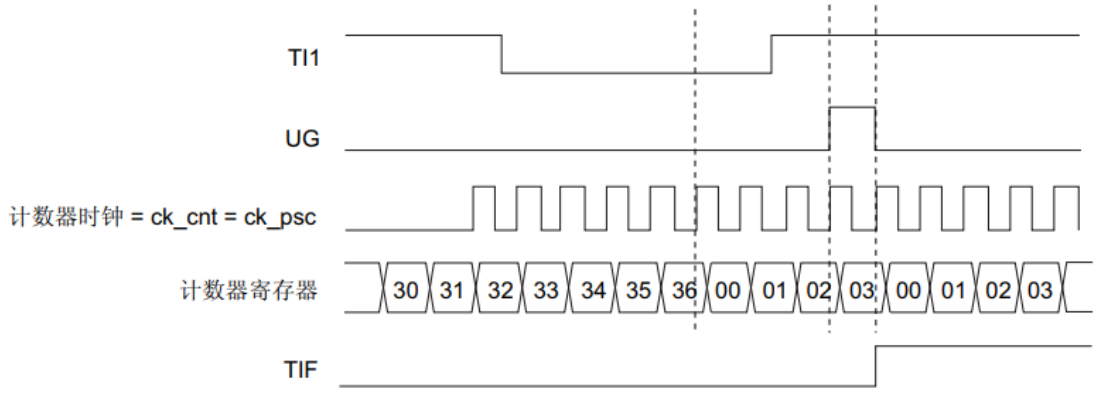


图 25.30 复位模式下的计数器时序图

25.3.5.1.2 从模式：门控模式

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_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。

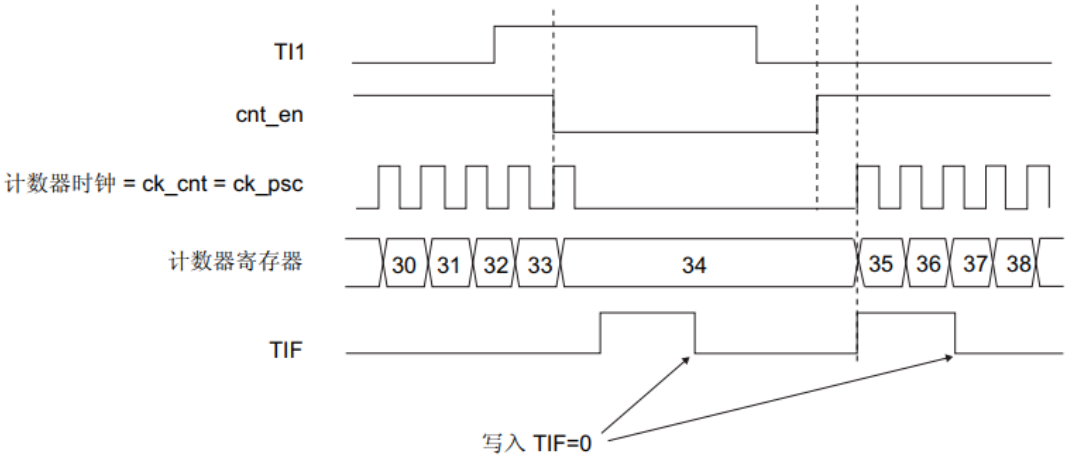


图 25.31 门控模式下的计数器时序图

25.3.5.1.3 从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 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。

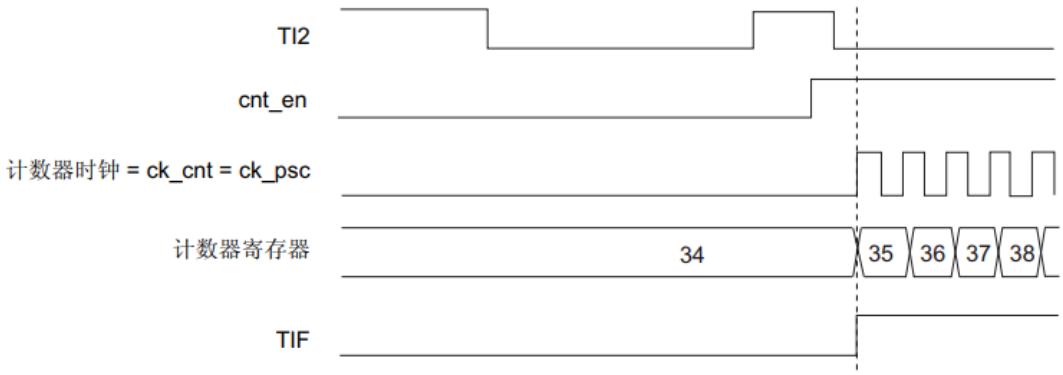


图 25.32 触发模式下的计数器时序图

25.3.5.1.4 从模式：组合复位+触发模式

在这种情况下，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。

该模式用于单脉冲模式。

25.3.6 DMA

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x=2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量=3。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：DBL=3 次传输，DBA=0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

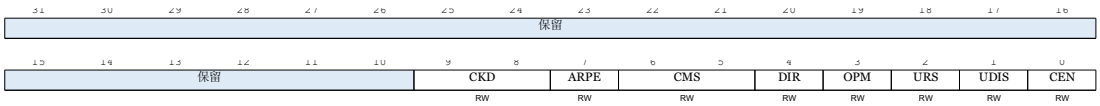
本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间 data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

25.4 寄存器

25.4.1 TIMx 控制寄存器 1（TIMx_CR1）

地址偏移：0x00

复位值：0x0000_0000



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKD	时钟分频 此位域指示定时器时钟（CK_INT）频率与死区发生器以及数字滤波器（ETR、TIx）所使用的死区及采样时钟（tDTS）之间的分频比， 00：tDTS=tCK_INT 01：tDTS=2*tCK_INT* *10：tDTS=4*tCK_INT 11：保留，不要设置成此值
7	ARPE	自动重载预装载使能 0：TIMx_ARR 寄存器不进行缓冲 1：TIMx_ARR 寄存器进行缓冲
6:5	CMS	中心对齐模式选择 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（TIMx_CCMRx 寄存器中的 CCxS=00）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（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。

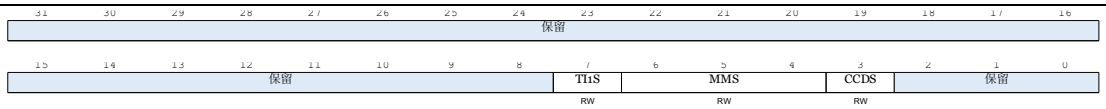
位/位域	名称	描述
		11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。 注：只要计数器处于使能状态（CEN=1），就不得从边沿对齐模式切换为中心对齐模式。
4	DIR	方向 0：计数器递增计数 1：计数器递减计数 注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。
3	OPM	单脉冲模式 0：计数器在发生更新事件时不会停止计数 1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）
2	URS	更新请求源 此位由软件置 1 和清零，用以选择 UEV 事件源。 0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括： — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件

位/位域	名称	描述
		1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。
1	UDIS	更新禁止 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0：使能 UEV。更新（UEV）事件可通过以下事件之一生成： — 计数器上溢/下溢 — 将 UG 位置 1 — 通过从模式控制器生成的更新事件 然后更新影子寄存器的值 1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。
0	CEN	计数器使能 0：禁止计数器 1：使能计数器 注：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

25.4.2 TIMx 控制寄存器 2（TIMx_CR2）

地址偏移：0x04

复位值：0x0000_0000



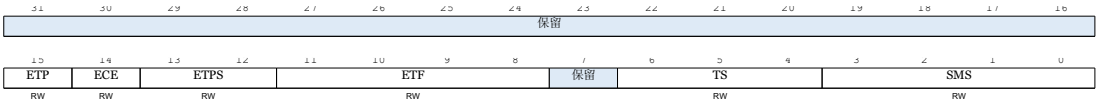
位/位域	名称	描述
31:8	保留	必须保持复位值
7	TI1S	TI1 选择 0：TIMx_CH1 引脚连接到 TI1 输入 1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）
6:4	MMS	主模式选择 这些位可选择主模式下将要发送到从定时器以实现同步的信息(TRGO)。这些位的组合如下： 000：复位——TIMx_EGR 寄存器中的 UG 位用作触发输出(TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。 001：使能——计数器使能信号 CNT_EN 用作触发输出(TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟。 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：比较脉冲——旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高），触发输出都会发送一个正脉冲。(TRGO)。

位/位域	名称	描述
		100：比较——OC1REF 信号用作触发输出(TRGO) 101：比较——OC2REF 信号用作触发输出(TRGO) 110：比较——OC3REF 信号用作触发输出(TRGO) 111：比较——OC4REF 信号用作触发输出(TRGO) 注：必须先使能从定时器的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器的时钟。
3	CCDS	捕获/比较 DMA 选择 0：发生 CCx 事件时发送 CCx DMA 请求 1：发生更新事件时发送 CCx DMA 请求
2:0	保留	必须保持复位值

25.4.3 TIMx 从模式控制寄存器（TIMx_SMCR）

地址偏移：0x08

复位值：0x0000_0000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 此位可选择将 ETR 还是 ETR 用于触发操作 0：ETR 未反相，高电平或上升沿有效。 1：ETR 反相，低电平或下降沿有效。
14	ECE	外部时钟使能

位/位域	名称	描述
		此位可使能外部时钟模式 2。 0：禁止外部时钟模式 2 1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。 注： 1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。 2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。 3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。
13:12	ETPS	外部触发预分频器 外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。 00：预分频器关闭 01：2 分频 ETRP 频率 10：4 分频 ETRP 频率 11：8 分频 ETRP 频率
11:8	ETF	外部触发滤波器 此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

位/位域	名称	描述
		0000：无滤波器，按 fDTS 频率进行采样 0001：fSAMPLING=fCK_INT，N=2 0010：fSAMPLING=fCK_INT，N=4 0011：fSAMPLING=fCK_INT，N=8 0100：fSAMPLING=fDTS/2，N=6 0101：fSAMPLING=fDTS/2，N=8 0110：fSAMPLING=fDTS/4，N=6 0111：fSAMPLING=fDTS/4，N=8 1000：fSAMPLING=fDTS/8，N=6 1001：fSAMPLING=fDTS/8，N=8 1010：fSAMPLING=fDTS/16，N=5 1011：fSAMPLING=fDTS/16，N=6 1100：fSAMPLING=fDTS/16，N=8 1101：fSAMPLING=fDTS/32，N=5 1110：fSAMPLING=fDTS/32，N=6 1111：fSAMPLING=fDTS/32，N=8
7	保留	必须保持复位值
6:4	TS	触发选择 此位域与 TS[4:3]位组合在一起。 此位域可选择将要用于同步计数器的触发输入。 00000：内部触发 0(ITR0) 00001：内部触发 1(ITR1) 00010：内部触发 2(ITR2) 00011：内部触发 3(ITR3) 00100：TI1 边沿检测器(TI1F_ED)

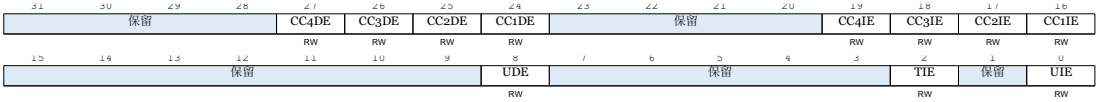
位/位域	名称	描述
		00101：滤波后的定时器输入 1(TI1FP1) 00110：滤波后的定时器输入 2(TI2FP2) 00111：外部触发输入(ETRF) 其它值：保留 注：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。
3:0	SMS	从模式选择 选择外部信号时，触发信号(TRGI)的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。 0000：禁止从模式——如果 CEN=“1”，预分频器时钟直接由内部时钟提供。 0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。 0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。 0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。 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。

位/位域	名称	描述
		0111：外部时钟模式 1——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。 1000 以上的代码：保留。 注：如果将 TI1F_ED 选作触发输入(TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。 注：必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

25.4.4 TIMx DMA/中断使能寄存器（TIMx_DIER）

地址偏移：0x0C

复位值：0x0000_0000



位/位域	名称	描述
31:28	保留	必须保持复位值
27	CC4DE	捕获/比较 4 DMA 请求使能 0：禁止 CC4 DMA 请求 1：使能 CC4 DMA 请求
26	CC3DE	捕获/比较 3 DMA 请求使能

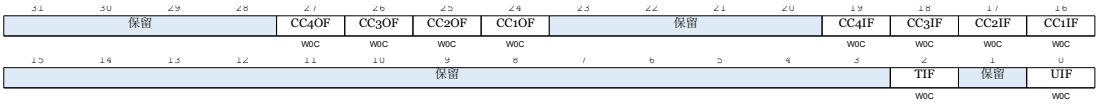
位/位域	名称	描述
		0：禁止 CC3 DMA 请求 1：使能 CC3 DMA 请求
25	CC2DE	捕获/比较 2 DMA 请求使能 0：禁止 CC2 DMA 请求 1：使能 CC2 DMA 请求
24	CC1DE	捕获/比较 1 DMA 请求使能 0：禁止 CC1 DMA 请求 1：使能 CC1 DMA 请求
23:20	保留	必须保持复位值
19	CC4IE	捕获/比较 4 中断使能 0：禁止 CC4 中断 1：使能 CC4 中断
18	CC3IE	捕获/比较 3 中断使能 0：禁止 CC3 中断 1：使能 CC3 中断
17	CC2IE	捕获/比较 2 中断使能 0：禁止 CC2 中断 1：使能 CC2 中断
16	CC1IE	捕获/比较 1 中断使能 0：禁止 CC1 中断 1：使能 CC1 中断

位/位域	名称	描述
15:9	保留	必须保持复位值
8	UDE	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:3	保留	必须保持复位值
2	TIE	触发中断使能 0：禁止触发中断 1：使能触发中断
1	保留	必须保持复位值
0	UIE	更新中断使能 0：禁止更新中断 1：使能更新中断

25.4.5 TIMx 状态寄存器（TIMx_SR）

地址偏移：0x10

复位值：0x0000_0000



位/位域	名称	描述
31:28	保留	必须保持复位值
27	CC4OF	捕获/比较 4 重复捕获标志 请参见 CC1OF 说明
26	CC3OF	捕获/比较 3 重复捕获标志

位/位域	名称	描述
		请参见 CC1OF 说明
25	CC2OF	捕获/比较 2 重复捕获标志 请参见 CC1OF 说明
24	CC1OF	捕获/比较 1 重复捕获标志 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。通过软件写入“0”可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。
23:20	保留	必须保持复位值
19	CC4IF	捕获/比较 4 中断标志 请参见 CC1IF 说明
18	CC3IF	捕获/比较 3 中断标志 请参见 CC1IF 说明
17	CC2IF	捕获/比较 2 中断标志 请参见 CC1IF 说明
16	CC1IF	捕获/比较 1 中断标志 如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见

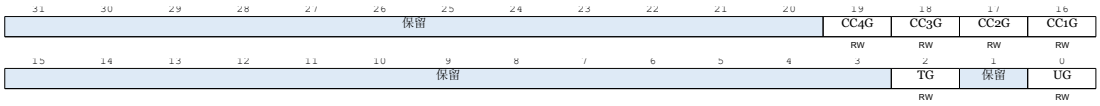
位/位域	名称	描述
		TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。 0：不匹配。 1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。 如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。 0：未发生输入捕获事件 1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）
15:3	保留	必须保持复位值
2	TIF	触发中断标志 在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。 0：未发生触发事件。 1：触发中断挂起。
1	保留	必须保持复位值
0	UIF	更新中断标志

位/位域	名称	描述
		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： ——TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器=0 时更新）。 ——TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。 ——TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时

25.4.6 TIMx 事件生成寄存器（TIMx_EGR）

地址偏移：0x14

复位值：0x0000_0000



位/位域	名称	描述
31:20	保留	必须保持复位值
19	CC4G	捕获/比较 4 生成 请参见 CC1G 说明
18	CC3G	捕获/比较 3 生成 请参见 CC1G 说明
17	CC2G	捕获/比较 2 生成

位/位域	名称	描述
		请参见 CC1G 说明
16	CC1G	捕获/比较 1 生成 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：通道 1 上生成捕获/比较事件： 如果通道 CC1 配置为输出： 使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果通道 CC1 配置为输入： TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。 如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。
15:3	保留	必须保持复位值
2	TG	触发生成 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：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。
1	保留	必须保持复位值
0	UG	更新生成 该位可通过软件置 1，并由硬件自动清零。 0：不执行任何操作

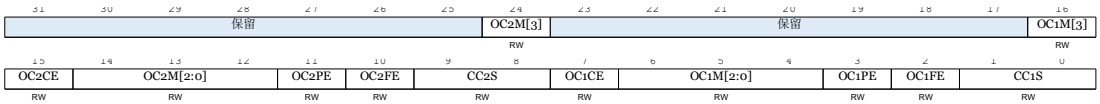
位/位域	名称	描述
		1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

25.4.7 TIMx 捕获/比较模式寄存器 1（TIMx_CCMR1）

地址偏移：0x18

复位值：0x0000_0000

输出比较：



位/位域	名称	描述
31:25	保留	必须保持复位值
24	CC2M[3]	输出比较 2 模式——位 3 请参见 OC2M 说明
23:17	保留	必须保持复位值
16	CC1M[3]	输出比较 1 模式——位 3 请参见 OC1M 说明
15	OC2CE	输出比较 2 清零使能
14:12	OC2M[2:0]	输出比较 2 模式
11	OC2PE	输出比较 2 预装载使能
10	OC2FE	输出比较 2 快速使能
9:8	CC2S	捕获/比较 2 选择

位/位域	名称	描述
		此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上 10：CC2 通道配置为输入，IC2 映射到 TI1 上 11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。
7	OC1CE	输出比较 1 清零使能 0：OC1Ref 不受 ETRF 输入影响 1：ETRF 输入上检测到高电平时，OC1Ref 立即清零
6:4	OC1M	输出比较 1 模式 这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。 0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

位/位域	名称	描述
		0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。 0011 ： 翻 转 ——TIMx_CNT=TIMx_CCR1 时 ，OC1REF 发生翻转。 0100：强制变为无效电平——OC1REF 强制变为低电平。 0101：强制变为有效电平——OC1REF 强制变为高电平。 0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT < TIMx_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 TIMx_CNT>TIMx_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 TIMx_CNT < TIMx_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 TIMx_CNT>TIMx_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有效状态。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

位/位域	名称	描述
		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。 1010：保留。 1011：保留。 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。

位/位域	名称	描述
		注： 在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时， OCREF 电平才会发生更改。 注： 此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。
3	OC1PE	输出比较 1 预装载使能 0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。 1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。 2：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。
2	OC1FE	输出比较 1 快速使能 此位用于加快触发输入事件对 CC 输出的影响。 0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。 1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为

位/位域	名称	描述
		3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。
1:0	CC1S	捕获/比较 1 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上 11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC1E=“0”），才可向 CC1S 位写入数据。

输入捕获：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC2F	输入捕获 2 滤波器
11:10	IC2PSC	输入捕获 2 预分频器
9:8	CC2S	捕获/比较 2 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上

位/位域	名称	描述
		10：CC2 通道配置为输入，IC2 映射到 TI1 上 11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。
7:4	IC1F	输入捕获 1 滤波器 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿： 0000：无滤波器，按 fDTS 频率进行采样 0001：fSAMPLING=fCK_INT，N=2 0010：fSAMPLING=fCK_INT，N=4 0011：fSAMPLING=fCK_INT，N=8 0100：fSAMPLING=fDTS/2，N=6 0101：fSAMPLING=fDTS/2，N=8 0110：fSAMPLING=fDTS/4，N=6 0111：fSAMPLING=fDTS/4，N=8 1000：fSAMPLING=fDTS/8，N=6 1001：fSAMPLING=fDTS/8，N=8 1010：fSAMPLING=fDTS/16，N=5 1011：fSAMPLING=fDTS/16，N=6 1100：fSAMPLING=fDTS/16，N=8 1101：fSAMPLING=fDTS/32，N=5 1110：fSAMPLING=fDTS/32，N=6

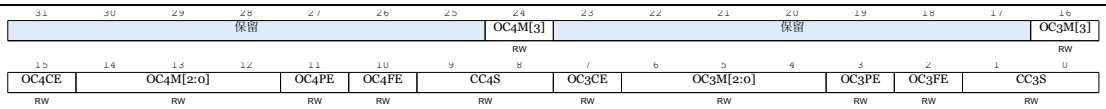
位/位域	名称	描述
		1111：fSAMPLING=fDTS/32，N=8
3:2	IC1PSC	输入捕获 1 预分频器 此位域定义 CC1 输入(IC1)的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。 00：无预分频器，捕获输入上每检测到一个边沿便执行捕获 01：每发生 2 个事件便执行一次捕获 10：每发生 4 个事件便执行一次捕获 11：每发生 8 个事件便执行一次捕获
1:0	CC1S	捕获/比较 1 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上 11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC1E=“0”），才可向 CC1S 位写入数据。

25.4.8 TIMx 捕获/比较模式寄存器 2（TIMx_CCMR2）

地址偏移：0x1C

复位值：0x0000_0000

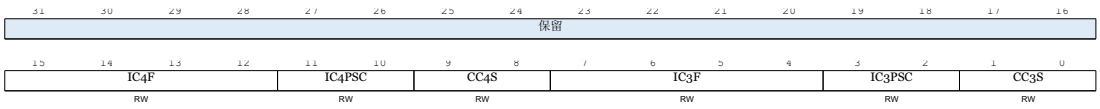
输出比较：



位/位域	名称	描述
31:25	保留	必须保持复位值
24	CC4M[3]	输出比较 4 模式——位 3 请参见 OC2M 说明
23:17	保留	必须保持复位值
16	CC3M[3]	输出比较 3 模式——位 3 请参见 OC1M 说明
15	OC4CE	输出比较 4 清零使能
14:12	OC4M[2:0]	输出比较 4 模式
11	OC4PE	输出比较 4 预装载使能
10	OC4FE	输出比较 4 快速使能
9:8	CC4S	捕获/比较 4 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 上 10：CC4 通道配置为输入，IC4 映射到 TI3 上 11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC4E=“0”），才可向 CC4S 位写入数据。
7	OC3CE	输出比较 3 清零使能

位/位域	名称	描述
6:4	OC3M	输出比较 3 模式
3	OC3PE	输出比较 3 预装载使能
2	OC3FE	输出比较 3 快速使能
1:0	CC3S	捕获/比较 3 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 上 10：CC3 通道配置为输入，IC3 映射到 TI4 上 11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注： 仅当通道关闭时（TIMx_CCER 中的 CC3E =“0”），才可向 CC3S 位写入数据。

输入捕获：



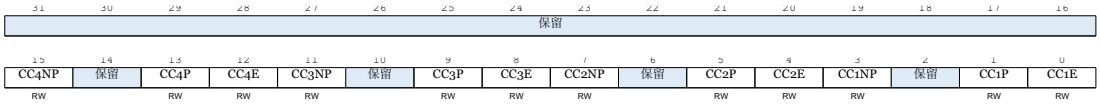
位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC4F	输入捕获 4 滤波器
11:10	IC4PSC	输入捕获 4 预分频器
9:8	CC4S	捕获/比较 4 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC4 通道配置为输出

位/位域	名称	描述
		01：CC4 通道配置为输入，IC4 映射到 TI4 上 10：CC4 通道配置为输入，IC4 映射到 TI3 上 11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC4E=“0”），才可向 CC4S 位写入数据。
7:4	IC3F	输入捕获 3 滤波器
3:2	IC3PSC	输入捕获 3 预分频器
1:0	CC3S	捕获/比较 3 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 上 10：CC3 通道配置为输入，IC3 映射到 TI4 上 11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC3E=“0”），才可向 CC3S 位写入数据。

25.4.9 TIMx 捕获/比较使能寄存器（TIMx_CCER）

地址偏移：0x24

复位值：0x0000_0000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	CC4NP	捕获/比较 4 互补输出极性 请参见 CC1NP 说明
14	保留	必须保持复位值
13	CC4P	捕获/比较 4 输出极性 请参见 CC1P 说明
12	CC4E	捕获/比较 4 输出使能 请参见 CC1E 说明
11	CC3NP	捕获/比较 3 互补输出极性 请参见 CC1NP 说明
10	保留	必须保持复位值
9	CC3P	捕获/比较 3 输出极性 请参见 CC1P 说明
8	CC3E	捕获/比较 3 输出使能 请参见 CC1E 说明
7	CC2NP	捕获/比较 2 互补输出极性 请参见 CC1NP 说明
6	保留	必须保持复位值
5	CC2P	捕获/比较 2 输出极性

位/位域	名称	描述
		请参见 CC1P 说明
4	CC2E	捕获/比较 2 输出使能 请参见 CC1E 说明
3	CC1NP	捕获/比较 1 互补输出极性 CC1 通道配置为输出： 0：OC1N 高电平有效。 1：OC1N 低电平有效。 CC1 通道配置为输入： 此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。 注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。
2	保留	必须保持复位值
1	CC1P	捕获/比较 1 输出极性 CC1 通道配置为输出： 0：OC1 高电平有效 1：OC1 低电平有效 CC1 通道配置为输入：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。 00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或

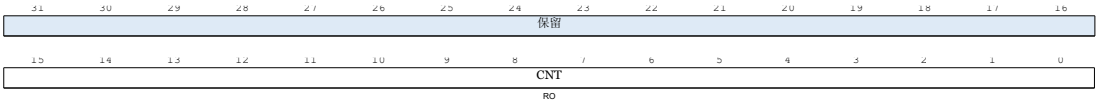
位/位域	名称	描述
		触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。 01：反相/下降沿触发。电路对 TIxFP1 下升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。 10：保留，不使用此配置。 11：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。 注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。
0	CC1E	捕获/比较 1 输出使能 CC1 通道配置为输出： 0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。 1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。 CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1(TIMx_CCR1)中。

位/位域	名称	描述
		0：禁止捕获。 1：使能捕获。 注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

25.4.10 TIMx 计数器（TIMx_CNT）

地址偏移：0x28

复位值：0x0000_0000

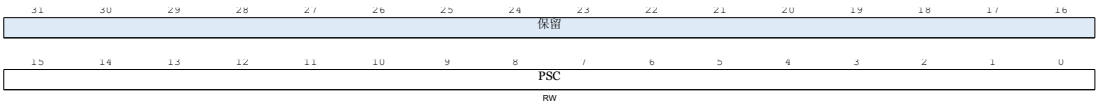


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT	计数器值

25.4.11 TIMx 预分频器（TIMx_PSC）

地址偏移：0x2C

复位值：0x0000_0000



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC	预分频器值 计数器时钟频率 (CK_CNT) 等于 fCK_PSC / (PSC[15:0] + 1)。 PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复

位/位域	名称	描述
		位模式”时通过触发控制器清零时）时要装载到有效预分频器寄存器的值。

25.4.12 TIMx 自动重载寄存器（TIMx_ARR）

地址偏移：0x30

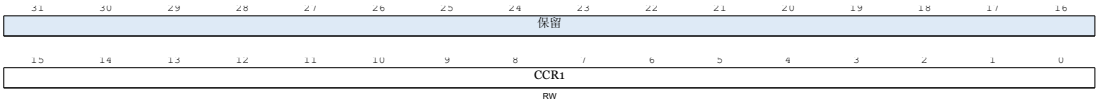
复位值：0x0000_FFFF

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ARR	自动重载值 ARR 为要装载到实际自动重载寄存器的值。 有关 ARR 更新和行为的更多详细信息，请参见时基单元。 当自动重载值为空时，计数器不工作。

25.4.13 TIMx 捕获/比较寄存器 1（TIMx_CCR1）

地址偏移：0x38

复位值：0x0000_0000



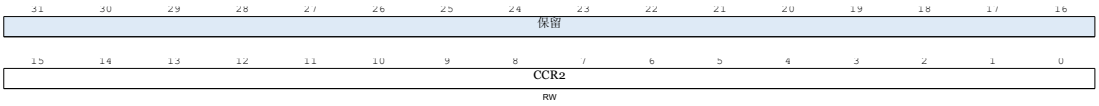
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR1	捕获/比较 1 值 如果通道 CC1 配置为输出：CCR1 为要装载到有效捕获/比较 1 寄存器的值（预装载值）。

位/位域	名称	描述
		如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。 如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

25.4.14 TIMx 捕获/比较寄存器 2（TIMx_CCR2）

地址偏移：0x3C

复位值：0x0000_0000



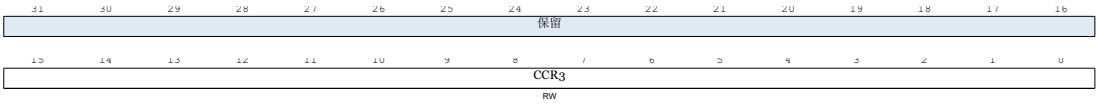
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR2	捕获/比较 2 值 如果通道 CC2 配置为输出：CCR2 为要装载到有效捕获/比较 2 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

位/位域	名称	描述
		如果通道 CC2 配置为输入：CR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

25.4.15 TIMx 捕获/比较寄存器 3（TIMx_CCR3）

地址偏移：0x40

复位值：0x0000_0000

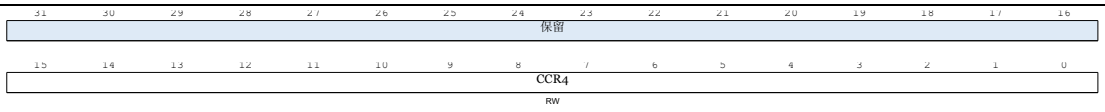


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR3	捕获/比较 3 值 如果通道 CC3 配置为输出：CCR3 为要装载到有效捕获/比较 3 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR3 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。 如果通道 CC3 配置为输入：CR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

25.4.16 TIMx 捕获/比较寄存器 4（TIMx_CCR4）

地址偏移：0x44

复位值：0x0000_0000

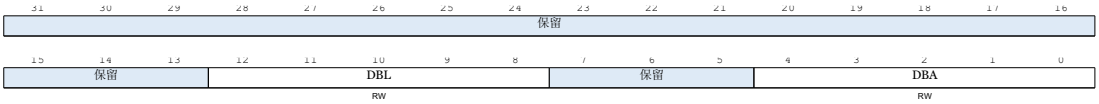


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR4	捕获/比较 4 值 如果通道 CC4 配置为输出：CCR4 为要装载到有效捕获/比较 4 寄存器的值（预装载值）。 如果没有通过 TIMx_CCMR4 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。 如果通道 CC4 配置为输入：CR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

25.4.17 TIMx DMA 控制寄存器（TIMx_DCR）

地址偏移：0x54

复位值：0x0000_0000



位/位域	名称	描述
31:13	保留	必须保持复位值
12:8	DBL	DMA 连续传送长度 该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次

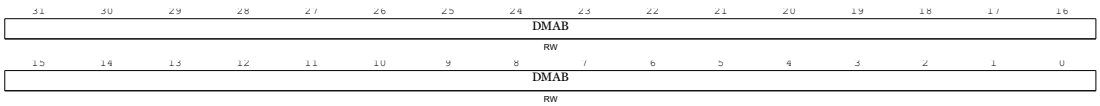
位/位域	名称	描述
		连续传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。 00000：1 次传送 00001：2 次传送 00010：3 次传送 … 10001：18 次传送 示例：以下面的传送为例：DBL = 7 字节且 DBA = TIM2_CR1。 – 如果 DBL = 7 字节且 DBA = TIM2_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址： (TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL 在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。 在这种情况下，将向自以下地址开始的 7 个寄存器传送数据：(TIMx_CR1 地址) + DBA 根据 DMA 数据大小的配置，可能发生下面几种情况： – 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。 – 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。
7:5	保留	必须保持复位值
4:0	DBA	DMA 基址

位/位域	名称	描述
		该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。 示例： 00000：TIMx_CR1， 00001：TIMx_CR2， 00010：TIMx_SMCR， …

25.4.18 TIMx 全传输 DMA 地址寄存器（TIMx_DMAR）

地址偏移：0x58

复位值：0x0000_0000



位/位域	名称	描述
31:0	DMAB	DMA 连续传送寄存器 对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4 其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

26 以太网接口（EMAC）

26.1 简介

MAC 接口它实现了由 IEEE 802.3 定义的媒体访问控制（MAC）通过以太网连接的碰撞检测（CSMA/CD）算法。与外部主机的通信是通过一组控制和状态寄存器和针对外部共享 RAM 的直接内存访问（DMA）控制器来实现的。MAC 接口通过 DMA 方式与主机实现数据发送。它自动获取发送缓冲区数据和存储接收缓冲区数据到外部 RAM。它通过对接收和发送描述符链表的管理，实现多种内存分配方案。MAC 接口内部 RAM 被用作可配置的 FIFO 存储器块，并且有单独的用于发送和接收进程的存储器块。

26.2 主要特征

- 支持外部 PHY 接口实现 10/100Mbit/s 数据传输速率；
- 通过符合 IEEE 802.3 的 MII 接口与外部快速以太网 PHY 进行通信；
- 支持全双工和半双工操作；
- 报头和帧起始数据（SFD）在发送路径中插入、在接收路径中删除；
- 可逐帧控制 CRC 和 PAD 自动生成；
- 接收帧时可自动去除 PAD/CRC；
- 支持多种灵活的地址过滤模式；
- 两组缓存 FIFO（接收 FIFO 和发送 FIFO）；
- 支持 DMA 在内存和 MAC 的缓存区进行数据搬移

26.3 功能说明

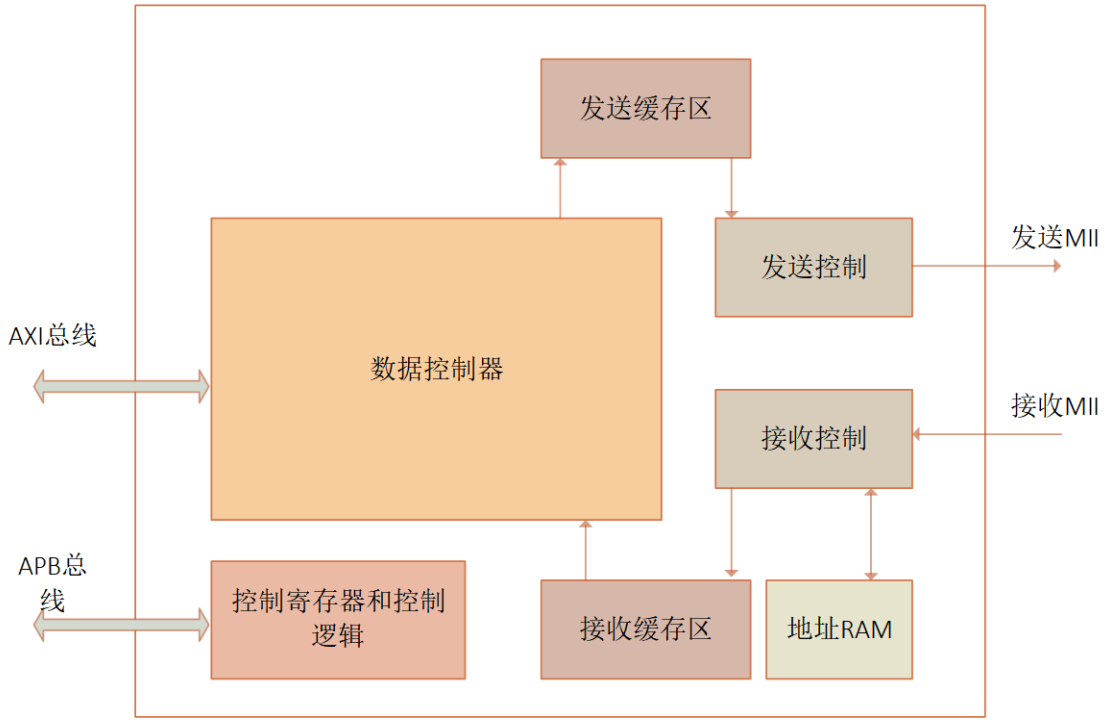


图 26.1 以太网 MAC 结构示意框图

26.3.1 介质独立接口：MII

介质独立接口（MII）定义了 10Mbit/s 和 100Mbit/s 的数据传输速率下 MAC 子层与 PHY 之间的互连。

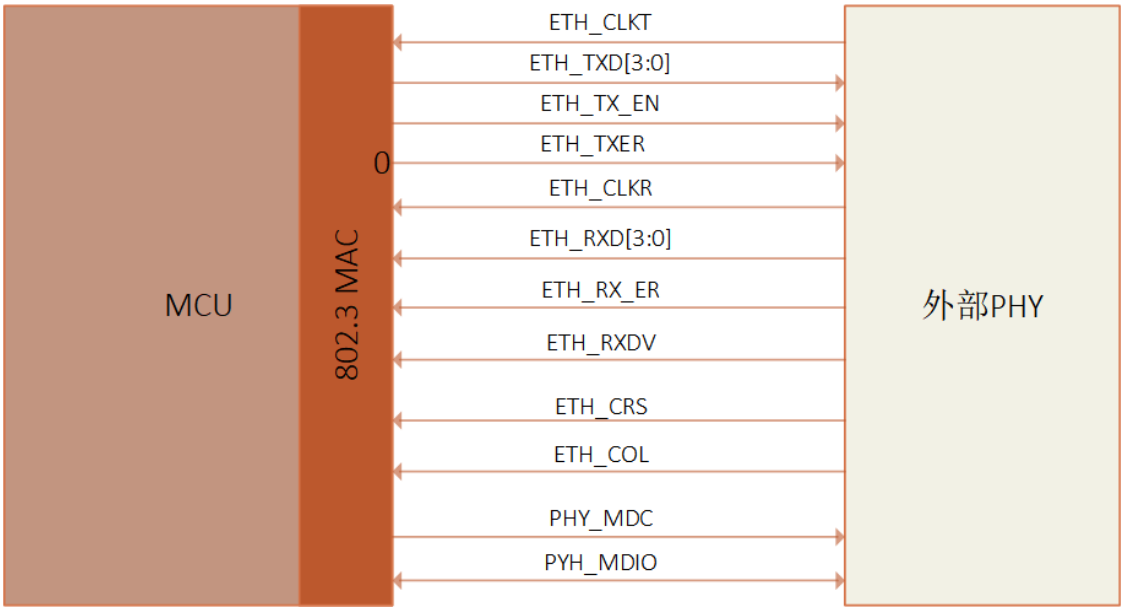


图 26.2 介质独立接口示意框图

- ETH_CLKT：连续时钟信号。

该信号提供进行 TX 数据传输时的参考时序。标称频率为：速率为 10Mbit/s 时为 2.5MHz；速率为 100Mbit/s 时为 25MHz

- ETH_CLKR：连续时钟信号。

该信号提供进行 RX 数据传输时的参考时序。标称频率为：速率为 10Mbit/s 时为 2.5MHz；速率为 100Mbit/s 时为 25MHz。

- ETH_TX_EN：发送使能信号。

该信号表示 MAC 当前正针对 MII 发送半字节。该信号必须与报头的前半字节进行同步（ETH_CLKT），并在所有待发送的半字节均发送到 MII 时必须保持同步。

- ETH_TXD[3:0]：数据发送信号。

该信号是 4bit 组成的数据信号，由 MAC 子层同步驱动，在 ETH_TX_EN 信号为高时数据有效。MII_TXD[0]为最低有效位，MII_TXD[3]为最高有效位。禁止 ETH_TX_EN 时，发送数据不会对 PHY 产生任何影响。

- ETH_TXER：发送错误。

当前版本该信号为静态信号，也即固定为 0（没有发送错误）。

- ETH_CRS：载波侦听信号。

当发送或接收介质处于非空闲状态时，由 PHY 使能该信号。发送和接收介质均处于空闲状态时，由 PHY 禁止该信号。PHY 必须确保 ETH_CRS 信号在冲突条件下保持有效状态。该信号无需与 TX 和 RX 时钟保持同步。在全双工模式下，该信号没意义。

- ETH_COL：冲突检测信号。

检测到介质上存在冲突后，PHY 立即使能冲突检测信号，并且只要存在冲突条件，冲突检测信号必须保持有效状态。该信号无需与 TX 和 RX 时钟保持同步。在全双工模式下，该信号没意义。

- ETH_RXD[3:0]：数据接收信号。

该信号是 4bit 组成的数据信号，由 PHY 同步驱动，在 ETH_RXDV 信号为高时数据有效。ETH_RXD[0]为最低有效位，ETH_RXD[3]为最高有效位。当 ETH_RXDV 禁止、ETH_RX_ER 使能时，特定的 ETH_RXD[3:0]值用于传输来自 PHY 的特定信息。

- ETH_RXDV：接收数据有效信号。

该信号表示 PHY 当前正针对 MII 接收已恢复并解码的半字节。该信号必须与恢复帧的头半字节进行同步（ETH_CLKR），并且一直保持同步到恢复帧的最后半字节。该信号必须在最后半字节随后的第一个时钟周期之前禁止。为了正确地接收帧，ETH_RXDV 信号必须在时间范围上涵盖要接收的帧，其开始时间不得迟于 SFD 字段出现的时间

- ETH_RX_ER：接收错误信号。

该信号必须保持一个或多个周期（ETH_CLKR），从而向 MAC 子层指示在帧的某处检测到错误。该错误条件必须通过 ETH_RXDV 验证

26.3.2 MAC 帧格式

图 26.3 MAC 帧格式

帧格式说明如下

- 报头：7 字节字段，用于同步通信 十六进制值：55-55-55-55-55-55-55 位模式：01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101（从右到左进行位发送）
- 起始帧定界符（SFD）：1 字节字段，用于指示帧的开始。十六进制值：D5 位模式：11010101（从右到左进行位发送）
- 目标地址和源地址字段：MAC 地址字段（6 字节），指示目标站和源站地具体如下：
  - 每个地址的长度为 48 位
  - 目标地址字段中的第一个 LSB 位（I/G）用于指示单个地址（I/G=0）或组地址（I/G=1）。一个组地址可以标识“无”、一个或多个，或所有连接至 LAN 的站。在源地址中，第一位保留并复位为 0。
  - 第二位（U/L）用于区分局部（U/L=1）或全局（U/L=0）管理地址。对于广播地址，该位同样为 1。
  - 每个地址字段的各个字节必须最先发送最低有效位

- MAC 客户端长度/类型：2 字节字段，具有不同含义（互斥），具体取决于其值：
  - 如果该值小于或等于 maxValidFrame（1500），则该字段表示 802.3 帧的后续数据字段中所包含的 MAC 客户端数据字节的数量（长度解析）。
  - 如果该值大于或等于 MinTypeValue（1536），则该字段表示与以太网帧相关的 MAC 客户端协议的种类（类型解析）。
  - 无论长度/类型字段的解析结果为何，如果数据字段的长度小于协议正确运行所需的最小长度，则将在数据字段之后、FCS（帧检查序列）字段之前添加一个 PAD 字段。发送和接收长度/类型字段时，高位字节在前。
  - 对于在 maxValidLength 到 minTypeValue 范围内（不包括边界）的长度/类型字段值，未指定 MAC 子层的行为：MAC 子层可能传递、也可能不传递这些值。
  - 数据和 PAD 字段：n 字节数据字段。其数据完全透明，这意味着数据字段中可能出现任意顺序的字节值。PAD 的大小（如果存在）由数据字段的大小决定。数据和 PAD 字段的最大和最小长度为：
  - 最大长度=1500 字节
  - 无标记的 MAC 帧的最小长度=46 字节
  - 带标记的 MAC 帧的最小长度=42 字节
  - 当数据字段的长度小于要求的最小长度时，将添加 PAD 字段以匹配最小长度（带标记的帧为 42 字节，无标记的帧为 46 字节）。
  - 帧检查序列：包含循环冗余校验（CRC）值的 4 字节字段。CRC 计算基于下列字段：源地址、目标地址、QTag 前缀、长度/类型、LLC 数据和 PAD（即，除报头、SFD 字段以外的所有字段）。

地址指定基于以下类型：

- 单个地址：与网络中的特殊站关联的物理地址

- 组地址：与给定网络中一个或多个站关联的多目标地址。有两种多播地址：
  - 多播组地址：与一组逻辑相关站关联的地址。
  - 广播地址：一个特殊的预定义多播地址（目标地址字段中全为“1”），该地址总是表示给定 LAN 上的所有站。
- QTag 前缀：在源地址字段和 MAC 客户端长度/类型字段中插入的 4 字节字段。该字段是对基本帧（未标记）的扩展，用于获得标记的 MAC 帧。未标记的 MAC 帧不包括该字段。扩展的标记如下：
  - 2 字节常量长度/类型字段值（符合“类型”解析，大于 0x0600），等于 802.1Q 变量协议类型的值（十六进制 0x8100）。该常量字段用于区分标记和未标记的 MAC 帧。
  - 包含变量控制信息的 2 字节字段可以再分为：一个 3 位用户优先级、一个 1 位标准格式指示符（CFI）和一个 12 位 VLAN 标识符。标记 MAC 帧的长度通过 QTag 前缀扩展 4 个字节

26.3.3 DMA 描述符说明

主机和 MAC 之间的数据交换是通过驻留在系统共享 RAM 中的描述符列表和数据缓冲区来执行的。这些缓冲区保存着 MAC 要发送或接收的主机数据。描述符充当指向这些 buffers 的指针。每个描述符列表应该由主机在共享内存区域中构建，大小可以是任意的。发送和接收进程都有一个单独的描述符列表。

第一个描述符在描述符列表中的位置由（接收描述符基地址寄存器）描述为接收列表，（发送描述符基地址寄存器）描述为发送列表。描述符既可以排列成链式结构，也可以排列成环状结构。在链式结构中，每个描述符都包含一指向列表中下一个描述符的指针。在一个环结构中，下一个描述符的地址由控制寄存器（DSL-descriptor skip length）决定。每个描述符最多可以指向两个数据 buffers。当使用描述符链接时，第二个 buffers 的地址被用作指向下一个描述符的指针；因此，只有一个缓冲区可用。一个帧可以占用一个或多个数据描述符和 buffers，但一个描述符不能超过一个帧。在环形结构中，如果只使用一个描述符，则描述符操作可能

会被破坏。此外，在环结构中，主机必须设置至少两个描述符。在发送进程中，主机可以将第一个描述符的所有权授予 MAC，并使由第一个描述符指定的数据被发送。同时，主机拥有它自己的第二个或最后一个描述符的所有权。这样做是为了防止 MAC 在主机准备好发送在第二个描述符中指定的数据之前获取下一帧。在一个接收进程中，所有可用的描述符的所有权，除非它被主机挂起处理，必须给 MAC。

在发送数据 FIFO 中最多可以存储两个帧，包括在发送数据 FIFO 内等待的帧，从数据接口发送到发送数据 FIFO 的帧，以及通过 MII 接口从发送数据 FIFO 发送出的帧。

在接收数据 FIFO 中存储最多四个帧，包括在接收数据 FIFO 内等待的帧，该帧从接收数据 FIFO 传输到数据接口，以及通过 MII 接口接收到接收数据 FIFO 中的帧。

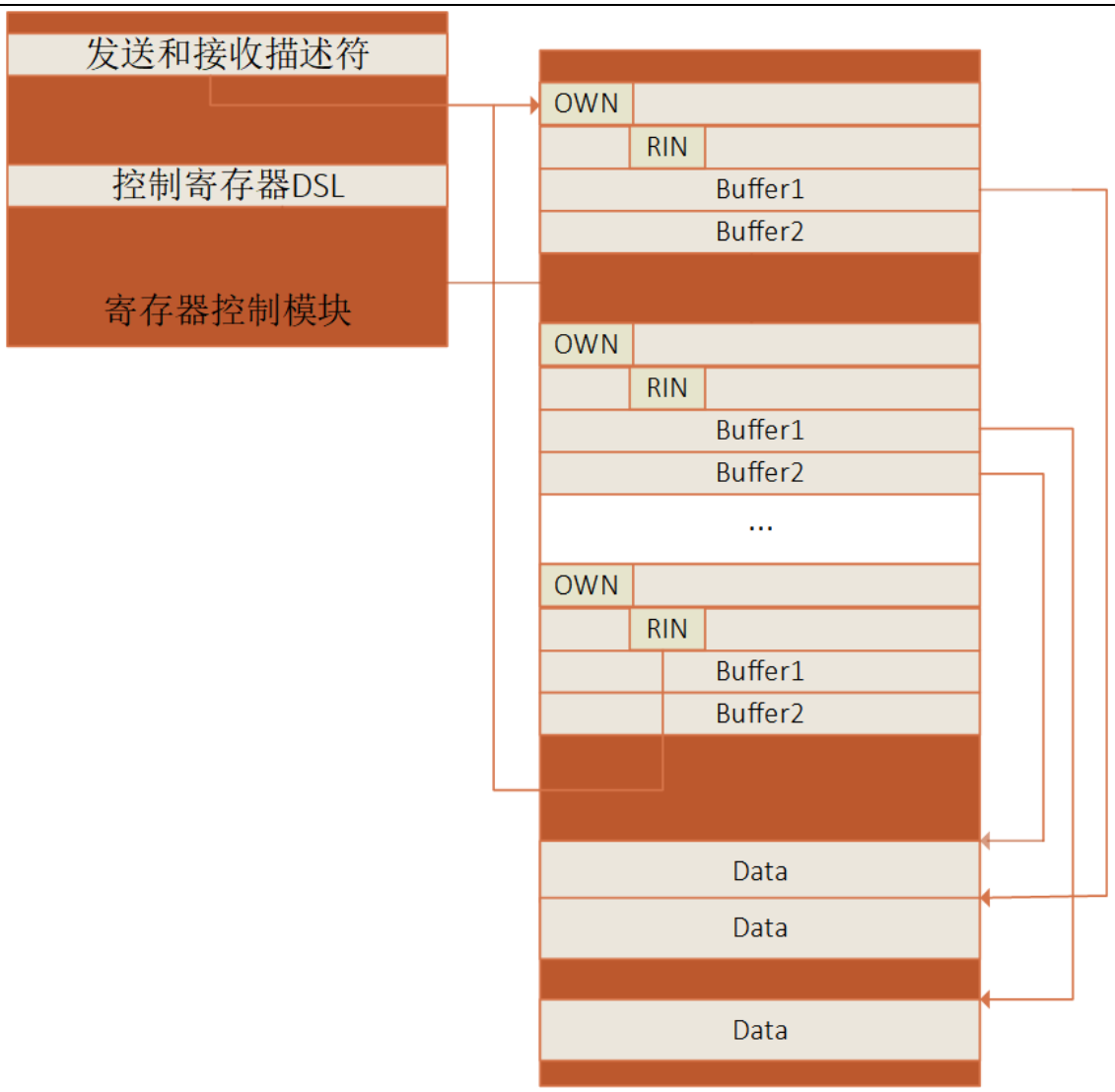


图 26.4 环形结构描述符

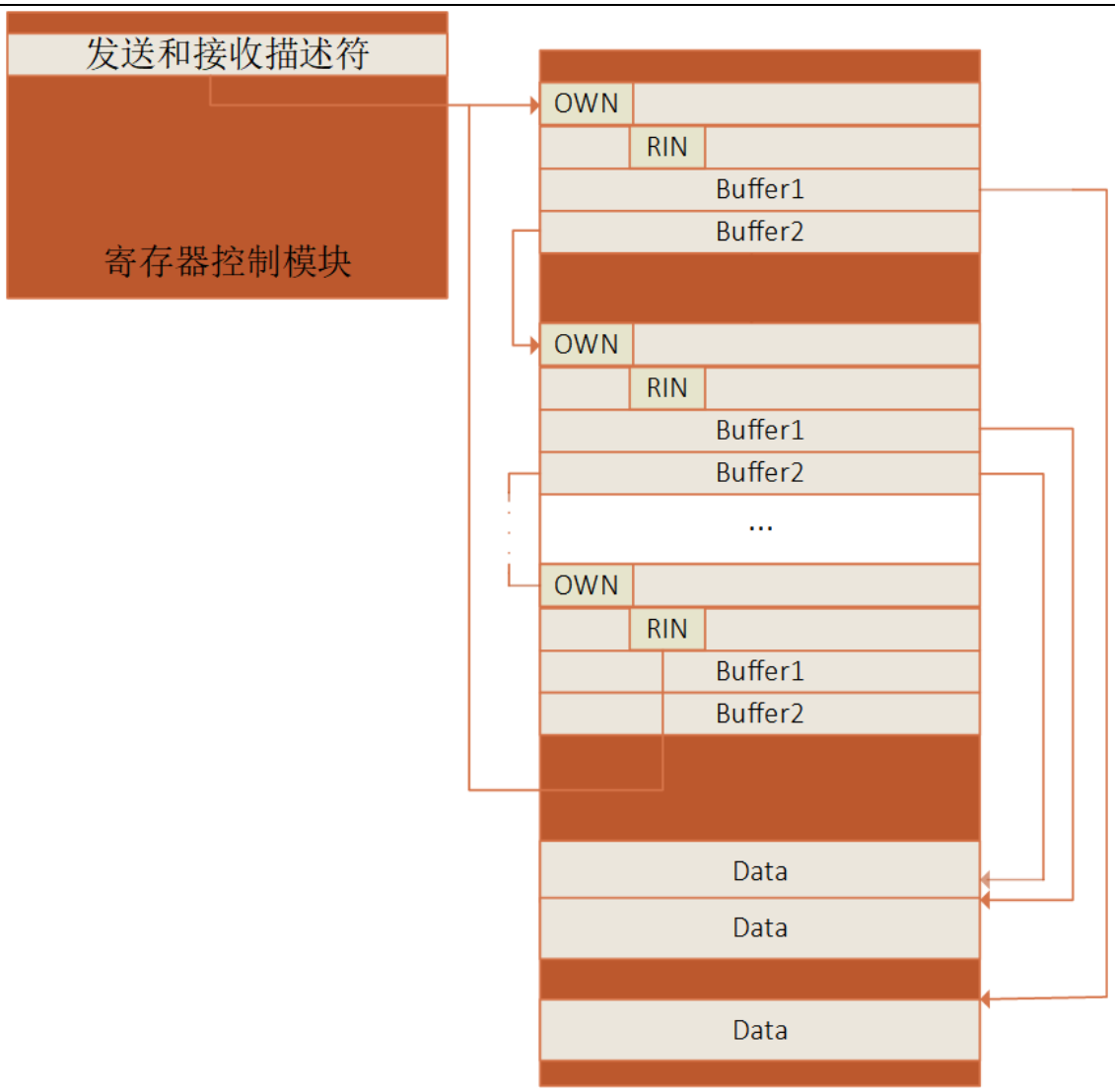


图 26.5 链式结构描述符

26.3.3.1 接收描述符说明

接收描述符包含 16 个字节，其组成如下图所示：

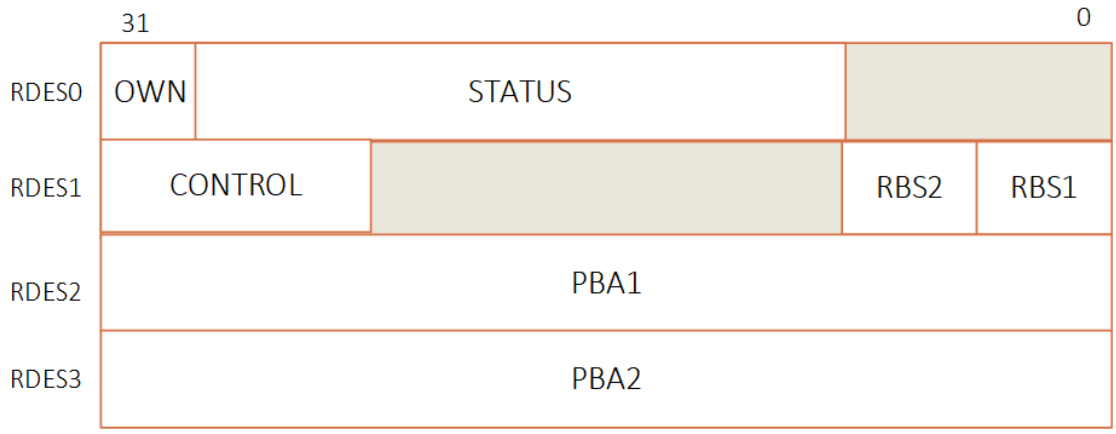
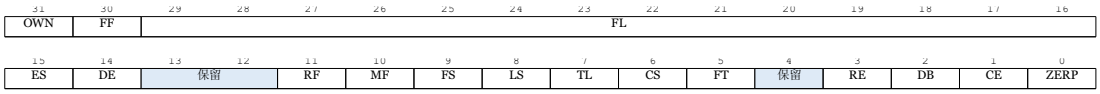


图 26.6 接收描述符的字节组成示意图

26.3.3.1.1 接收描述符-RDES0

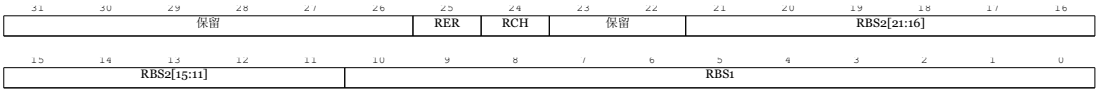


位/位域	名称	描述
31	OWN	所有关系位（Own bit） 该位置 1 时，表示描述符为 DMA 所有。该位复位时，表示描述符为 CPU 所有。DMA 可在完成帧发送时或在描述符分配的缓冲区全部读取完成后将该位清零。将属于同一个帧的所有后续描述符置 1 后，必须将帧的第一个描述符的所有位置 1。
30	FF	过滤失败。设置后，表示接收到的帧没有通过地址识别过程。当设置了模式寄存器 bit30（接收所有）位并且帧至少 64 字节长时，这个位只对帧的最后一个描述符有效（设置为 RDES0.8）。
29:16	FL	帧长度。表示给定帧发送到主机内存中的数据的长度（以字节为单位）这个位只有在 RDES0.8（最后一个描述符）被设置并且 RDES0.14（描述符错误）被清除时才有效。
15	ES	错误摘要

位/位域	名称	描述
		此位是以下位的逻辑或： RDES0.1：CRC error RDES0.6：Collision seen RDES0.7：Frame too long RDES0.11：Runt frame RDES0.14：描述符错误 此位仅在设置 RDES0.8（最后一个描述符）时有效。
14	DE	描述符错误 当试图存储接收到的数据时，没有可用的接收缓冲区时，由 MAC 设置。 此位仅在设置 RDES0.8（最后一个描述符）时有效。
13:12	保留	必须保持复位值
11	RF	短帧 设置后，表示帧被碰撞损坏或在碰撞窗口结束之前被提前终止。 此位仅在设置 RDES0.8（最后一个描述符）时有效。
10	MF	组播帧 设置后，表示该帧有一个组播地址。此位仅在设置 RDES0.8（最后一个描述符）时有效。
9	FS	第一个描述符块 设置后，表示这是帧的第一个描述符。
8	LS	最后描述符 设置后，表示这是一个帧的最后一个描述符。
7	TL	帧过长 设置后，表示当前帧大于 802.3 规定的最大帧大小 1518 字节。

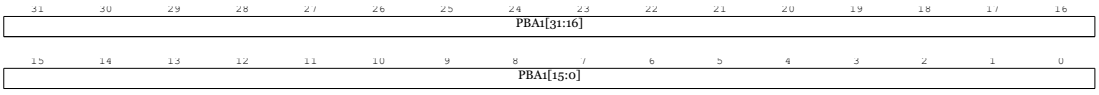
位/位域	名称	描述
		当接收到的帧超过 1518 字节时，在接收描述符中设置了 TL（帧过长）。 当一个帧使用多个描述符时，此标志在所有接收描述符中有效。
6	CS	设置后，表示看到了一个后期的冲突（在 SFD 之后的 64 字节之后的冲突）。 此位仅在设置 RDES0.8（最后一个描述符）时有效。
5	FT	帧类型 设置后，表示该帧的字段长度大于 1500（以太网类型的帧）。 清除时表示 802.3 类型的帧。 此位仅在设置 RDES0.8（最后一个描述符）时有效。 此外，FT 对于小于 14 字节的矮帧无效。
4	保留	必须保持复位值
3	RE	报告 MII 错误 设置后，表示通过 MII 接口连接的物理层芯片检测到错误。 此位仅在设置 RDES0.8（最后一个描述符）时有效。
2	DB	设置时，表示帧不是字节对齐的。此位仅在设置 RDES0.8（最后一个描述符）时有效。
1	CE	设置该值，表示接收的帧发生了 CRC 错误。此位仅在设置 RDES0.8（最后一个描述符）时有效。此外，CE 在接收到的帧是一个矮帧时是无效的。
0	ZERO	对于具有合法长度的帧，该位将被重置。

26.3.3.1.2 接收描述符-RDES1



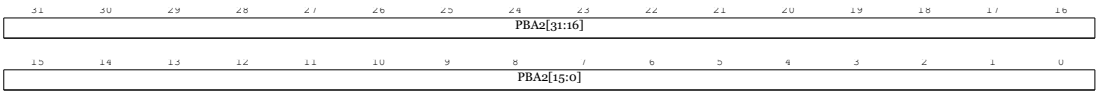
位/位域	名称	描述
31:26	保留	必须保留复位
25	RER	设置后，表示这是接收描述符环中的最后一个描述符。DMA 返回到环中的第一个描述符，由寄存器指定（接收列表地址的开始）。
24	RCH	设置后，表示第二个缓冲区的地址指向下一个描述符，而不是数据缓冲区。请注意，RER 优先于 RCH。
23:22	保留	必须保持复位值
21:11	RBS2	指示第二个数据缓冲区使用的内存空间的大小（以字节为单位）。这个数字必须是 4 的倍数。如果是 0，MAC 将忽略第二个数据缓冲区，并获取下一个数据描述符。该值仅在清除 RDES1.24（第二个地址链）时有效。
10:0	RBS1	指示第一个数据缓冲区使用的内存空间的大小（以字节为单位）。这个数字必须是 4 的倍数。如果为 0，则 MAC 忽略第一个数据缓冲区，使用第二个数据缓冲区。

26.3.3.1.3 接收描述符-RDES2



位/位域	名称	描述
31:0	PBA1	接收缓冲区 1 地址

26.3.3.1.4 接收描述符-RDES3



位/位域	名称	描述
31:0	PBA2	接收缓冲区 2 地址

26.3.3.2 发送描述符说明

与接收描述符一样，发送描述符也是由 16 个字节组成。

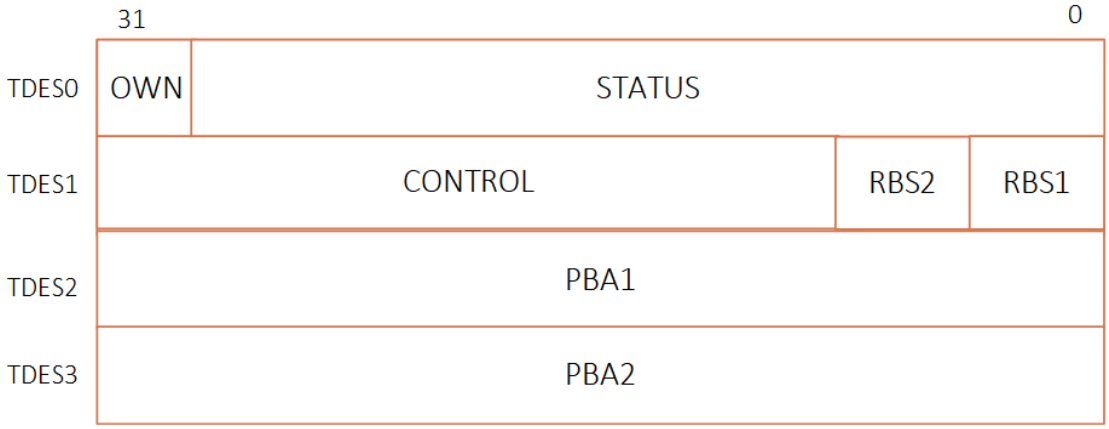
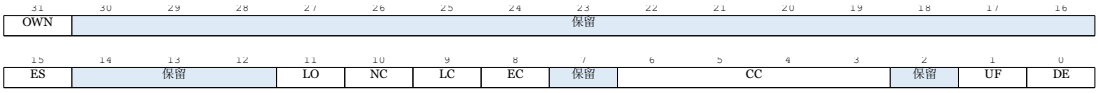


图 26.7 发送描述符的字节组成示意图

26.3.3.2.1 发送描述符-TDES0

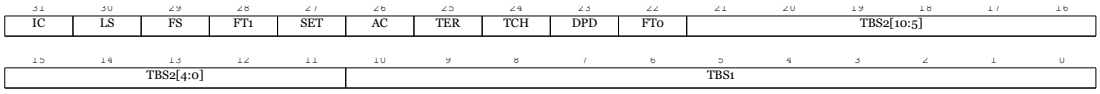


位/位域	名称	描述
31	OWN	所有关系位（Own bit） 该位置 1 时，表示描述符为 DMA 所有。该位复位时，表示描述符为 CPU 所有。DMA 可在完成帧发送时或在描述符分配的缓冲区全部读取完成后将该位清零。将属于同一个帧的所有后续描述符置 1 后，必须将帧的第一个描述符的所有位置 1。
30:16	保留	必须保持复位值
15	ES	错误摘要 这个位是下列位的逻辑或： TDES0.1：下溢错误 TDES0.8：过度碰撞错误

位/位域	名称	描述
		TDES0.9：延迟碰撞 TDES0.10：没有载波讯号 TDES0.11：损失的载体 这个位仅在设置 TDES1.30（最后一个描述符）时有效。
14:12	保留	必须保持复位值
11	LO	载波丢失 设置后，表示发送过程中载波丢失。这个位仅在设置 TDES1.30（最后一个描述符）时有效。
10	NC	没有载波 设置后，表示该载波在发送过程中没有被外部收发机断言。这个位仅在设置 TDES1.30（最后一个描述符）时有效。
9	LC	检测到冲突 设置后，表示发送 64 字节后检测到冲突。当设置 TDES0.1（下溢错误）时，此位无效。 这个位仅在设置 TDES1.30（最后一个描述符）时有效。
8	EC	设置后，表示发送在重试 16 次后终止。这个位仅在设置 TDES1.30（最后一个描述符）时有效。
7	保留	必须保持复位值
6:3	CC	冲突计数器 更新碰撞计数这个字段表示在帧发送结束之前发生的冲突数。当设置 TDES0.8（过度碰撞位）时，此值无效。这个位仅在设置 TDES1.30（最后一个描述符）时有效。

位/位域	名称	描述
2	保留	必须保持复位值
1	UF	下溢错误 设置该值，表示发送帧时 FIFO 为空。这个位仅在设置 TDES1.30（最后一个描述符）时有效。
0	DE	延迟 设置后，表示帧在发送之前被延迟。如果在发送准备开始时检测到载波，则发生延迟。 这个位仅在设置 TDES1.30（最后一个描述符）时有效。

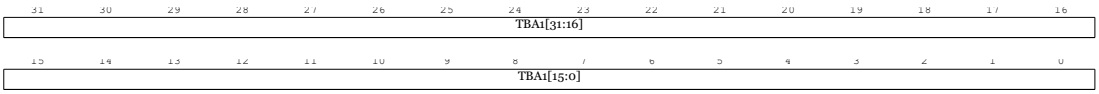
26.3.3.2.2 发送描述符-TDES1



位/位域	名称	描述
31	IC	设置这个标志指示 DMA 在处理当前帧后立即设置状态寄存器 bit0（发送中断）。 这个位在设置 TDES1.30（最后一个描述符）或设置包时有效。
30	LS	当设置时，表示帧的最后一个描述符。
29	FS	当设置时，表示帧的第一个描述符。
28	FT1	该位与 TDES0.22（FT0）一起控制电流滤波模式。此位仅对设置帧有效。
27	SET	当设置时，表示这是一个设置帧描述符。
26	AC	设置后，DMA 不会在帧的末尾追加 CRC 值。当帧小于 64 字节并且启用了自动填充字节时例外。在这种情况下，添加 CRC 字段，不管 AC 标志的状态如何
25	TER	设置后，表示描述符环中的最后一个描述符。

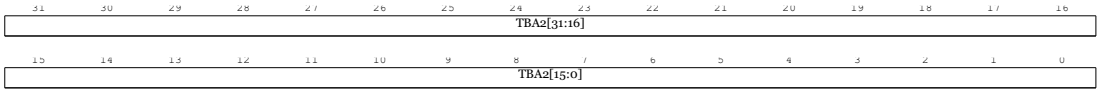
位/位域	名称	描述
24	TCH	设置后，表示第二个描述符的地址指向下一个描述符，而不是指向数据缓冲区。 该位仅在 TDES1.25（发送描述符环的最后一个描述符）复位时有效。
23	DPD	当设置时，自动字节填充被禁用。DMA 通常在实际帧大小小于 64 字节时，在 INFO 字段后面附加 PAD 字段。填充字节后，CRC 字段也被插入，不管 AC 标志的状态如何。当设置 DPD 时，不追加填充字节
22	FT0	这个位和 TDES0.28（FT1）一起控制当前的滤波模式。 该位仅在设置了 TDES1.27（SET）位时有效。
21:11	TBS2	指示第二个数据缓冲区使用的内存空间的大小（以字节为单位）。 如果为 0，DMA 将忽略第二个数据缓冲区，并获取下一个数据描述符。 该位仅在 TDES1.24（第二个地址链）被清除时有效。
10:0	TBS1	指示第一个数据缓冲区使用的内存空间的大小（以字节为单位）。 如果为 0，则 DMA 忽略第一个数据缓冲区，使用第二个数据缓冲区。

26.3.3.2.3 发送描述符-TDES2



位/位域	名称	描述
31:0	TBA1	发送缓冲区 1 地址

26.3.3.2.4 发送描述符-TDES3



位/位域	名称	描述
31:0	TBA2	发送缓冲区 1 地址

26.3.3.3 MAC 发送进程

发送进程可以在三种模式中运行：运行、停止或挂起。在软件或硬件复位后，或在发送停止命令后，发送进程处于停止状态。

当处于运行状态时，发送进程执行描述符/缓冲区处理。当操作处于挂起或停止状态时，发送进程保留下一个描述符的位置，也就是说，最后一个被关闭的描述符后面的描述符地址。进入运行状态后，该位置将用于下一个描述符的获取。唯一的例外是当主机写发送时。

描述符基址寄存器。在这种情况下，将重置描述符地址，并将获取定向到列表中的第一个位置。在写入寄存器之前，MAC 必须处于停止状态。当运行在停止状态时，发送进程已停止（Transmit Process Stopped，简称 TPS）输出是高。这个输出可以用来禁用 MAC 外部的 ETH_CLKT 时钟信号。当发送进程已停止(TPS)和接收进行已停止（Receive Process Stopped，简称 RPS）输出都是高时，除 PCLK 外的所有时钟信号都可以在 MAC 外部被禁用。 发送进程将一直运行，直到发生以下事件之一：

1. 发出硬件或软件复位。设置控制寄存器（SWR）位可以执行软件复位。复位后，所有的内部寄存器都返回到它们的默认状态。当前描述符在发送描述符列表中的位置丢失。
2. 主机发出一个停止发送命令。这可以通过将 0 写入模式寄存器（ST）位来执行。当前描述符的位置被保留。
3. 找到主机拥有的描述符。当前描述符的位置被保留。
4. 检测到发送 FIFO 下溢错误。当发送帧的 FIFO 为空时，会产生下溢错误。当它发生时，发送进程进入暂停状态。发送自动轮询在内部被禁用，即使主机通过写入 TAP 位来启用它。当前描述符的位置被保留。

在以下情况之一，可以离开暂停状态：

1. 发出发送轮询请求命令。这可以通过将发送轮询寄存器写入一个非零值来执行。发送轮询请求命令也可以在开启发送自动轮询功能时自动生成。只有当控制寄存器 bit0（19..17）（TAP）位被写入一个非零值，并且在进入暂停状态之前没有下溢错误时，发送自动轮询才会被启用。
2. 主机发出一个停止发送命令。这可以通过将 0 写入模式寄存器 bit13（ST）位来执行。当前描述符的位置被保留。

发送过程的典型数据流如图所示。发送过程中的事件通常按照以下顺序发生：

1. 主机需要设置操作模式、中断等设置（配置控制寄存器，模式寄存器，中断寄存器等）
2. 主机在共享 RAM 中设置发送描述符/数据
3. 主机发起发送启动命令（设置模式寄存器 bit13）
4. MAC 开始获取发送描述符
5. MAC 的 DMA 开始从共享 RAM 中发送数据到发送数据 RAM 发送数据 RAM 是 MAC 内部的数据缓冲区，共享 RAM 是外部指针指向的 ram 空间）。
6. MAC 开始在 MII 上发送数据。

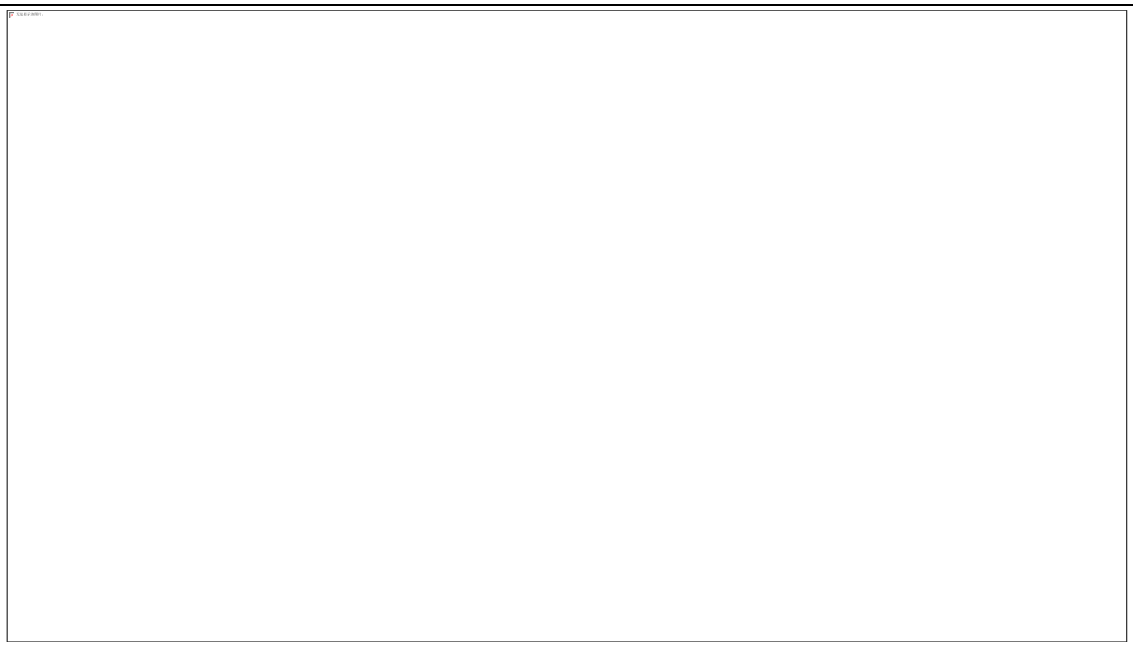


图 26.8 MAC 发送数据状态转换示意图

26.3.3.4 MAC 接收进程

接收进程可以在三种模式中运行：运行、停止或挂起。当软件或硬件复位或执行了停止接收命令后，接收进程处于停止状态。只有在启动接收命令之后，接收进程才能离开停止状态。

在运行状态下，接收器执行描述符/缓冲区处理。在运行状态下，接收方从接收描述符列表中获取数据。无论链接上是否有帧，它都会执行这个获取操作。当没有等待的帧时，接收进程读取描述符并等待帧。当一个有效的帧被识别时，接收进程开始填充由当前描述符指向的内存缓冲区。当帧结束时，或者当内存缓冲区被完全填满时，当前帧描述符被关闭（所有权位清除）。立即，以同样的方式获取列表中的下一个描述符，以此类推。

当操作处于挂起或停止状态时，接收进程保留下一个描述符的位置（最后一个关闭的描述符后面的描述符地址）。进入运行状态后，保留的位置将用于下一个描述符的获取。唯一的例外是当主机写入接收描述符基址寄存器（0x18）时。在这种情况下，描述符地址被重置，并且获取被指向列表中的第一个位置。在写入寄存器之前，MAC 必须处于停止状态。

当运行在停止状态时，RPS 输出为高。这个输出允许在外部关闭接收时钟 ETH_CLKR。当 RPS 和 TPS 输出都是高时，除 PCLK 外的所有时钟都可以从外部关闭。

接收流程将一直运行，直到发生以下事件之一：

1. 硬件或软件复位是由主机发出的。软件复位可以通过设置控制寄存器 bit0（SWR）位来执行。复位后，所有内部寄存器返回到默认状态。当前描述符在接收描述符列表中的位置丢失。
2. 主机发出一个停止接收命令。这可以通过将 0 写入模式寄存器 bit1（SR）位来执行。当前描述符的位置被保留。
3. 主机拥有的描述符在获取描述符期间由 MAC 找到。当前描述符的位置被保留。 在以下情况之一，可以离开暂停状态：
4. 接收轮询命令由主机发出。这可以通过向接收轮询寄存器写入非零值来实现。
5. MAC 在接收链路上检测到一个新帧。
6. 主机发出一个停止接收命令。这可以通过将 0 写入模式寄存器 bit1（SR）位来执行。当前描述符的位置被保留。 如果给出了一个停止接收命令，接收状态机将在当前帧完成后进入停止状态。它不会立即进入停止状态 接收流程中的典型数据流如图所示。接收流程的事件通常按以下顺序发生：
7. 主机为操作模式、中断等设置相应的寄存器。
8. 主机在共享 RAM 中设置接收描述符
9. 主机发送开始接收命令。
10. MAC 的 DMA 开始获取发送描述符。
11. MAC 等待 MII 上的接收数据
12. MAC 核将接收到的数据发送到接收数据 FIFO
13. MAC 的 DMA 将接收到的数据从接收数据 FIFO 发送到共享 RAM

图 26.9 MAC 接收数据状态转换示意图

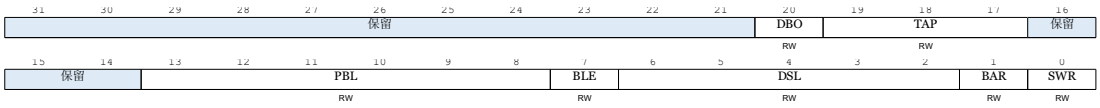
26.4 寄存器

26.4.1 总线模式寄存器（MAC_BUS_MODE）

地址偏移：0x00

复位值：0xFE000000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31:21	保留	必须保持复位值
20	DBO	描述符字节排序模式： 1：用于数据描述符的大端模式 0：用于数据描述符的小端模式
19:17	TAP	发送自动轮询 如果 TAP 被写入一个非零值，MAC 在运行在挂起状态时将执行一个自动发送描述符轮询。当描述符可用时，

位/位域	名称	描述
		发送进程进入运行状态。当描述符被标记为由主机拥有时，发送进程将保持挂起。 轮询总是在当前的发送描述符列表位置执行。两个连续轮询之间的时间间隔如下所示。 Bits 10Mbps 100Mbps 000 TAB 禁用 001 819 微秒 81.9 微秒 010 2450 微秒 24.5 微秒 011 5730 微秒 57.3 微秒 100 51.2 微秒 5.12 微秒 101 102.4 微秒 10.24 微秒 110 153.6 微秒 15.36 微秒 111 358.4 微秒 35.84 微秒
16:14	保留	必须保持复位值
13:8	PBL	可编程脉冲长度 指定一个 DMA 事务中可以发送的最大字数。取值范围为：0、1、2、4、8、16、32。当值 0 被写入时，突发只被内部 FIFO 的阈值水平限制。 注意，PBL 只对数据缓冲区有效。
7	BLE	大/小端 选择数据缓冲区使用的字节顺序模式 1：用于数据缓冲区的大端模式 0：用于数据缓冲区的小端模式
6:2	DSL	描述符跳过长度 指定一个环结构中两个连续描述符之间的长字数（间隔大小）。
1	BAR	总线仲裁方案 1：发送和接收进程在访问总线时具有同等的优先级

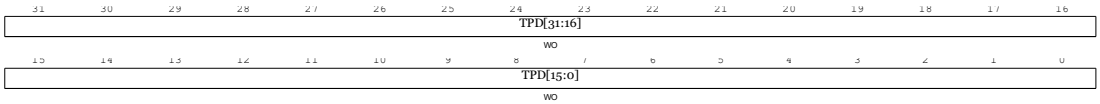
位/位域	名称	描述
		0：智能仲裁，接收进程比发送进程具有优先级。
0	SWR	软复位 设置此位将重置所有内部触发器，处理器应将 1 写入该位，然后等待读取返回 0，表示复位已完成。该位将在几个时钟周期内保持设置。

26.4.2 发送轮询请求寄存器（MAC_TX_POLL）

地址偏移：0x08

复位值：0x00000000

该寄存器只支持按字（32 位）访问



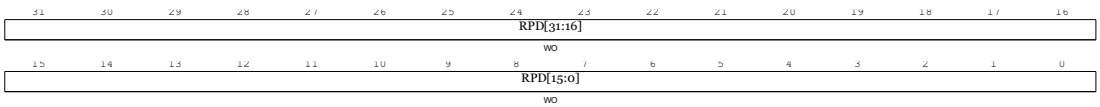
位/位域	名称	描述
31:0	RPD	使用任何值写入此字段将指示 MAC 检查要发送的帧。 此操作仅在发送进程挂起时有效。 如果没有可用的描述符，发送过程将保持挂起状态。 当描述符可用时，发送进程进入运行状态。 该寄存器读取返回值为 0。

26.4.3 接收轮询请求寄存器（MAC_RX_POLL）

地址偏移：0x10

复位值：0x00000000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31:0	RPD	使用任何值写入此字段将指示 MAC 检查要获取的接收描述符。

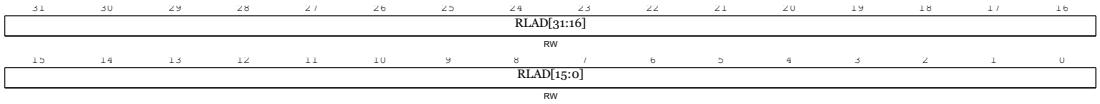
位/位域	名称	描述
		此操作仅在接收进程挂起时有效。 如果没有可用的描述符，接收进程将保持挂起状态。 当描述符可用时，接收进程进入运行状态。 该寄存器读取返回值为 0。

26.4.4 接收描述符基地址寄存器（MAC_RDES_BASE_ADDR）

地址偏移：0x18

复位值：0xFFFFFFFF

该寄存器只支持按字（32 位）访问



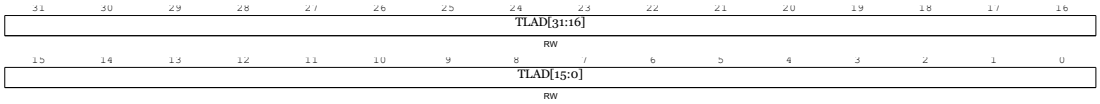
位/位域	名称	描述
31:0	RLAD	接收列表地址的开始 包含接收描述符列表中第一个描述符的地址。此地址必须是长字对齐的

26.4.5 发送描述符基地址寄存器（MAC_TDES_BASE_ADDR）

地址偏移：0x20

复位值：0xFFFFFFFF

该寄存器只支持按字（32 位）访问



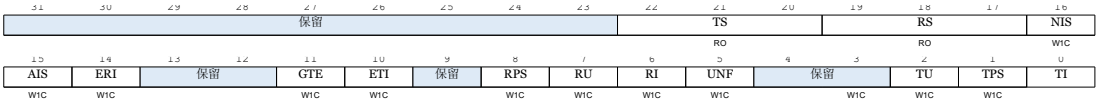
位/位域	名称	描述
31:0	RLAD	发送列表地址的开始 包含发送描述符列表中第一个描述符的地址。此地址必须是长字对齐的

26.4.6 状态寄存器（MAC_STATUS）

地址偏移：0x28

复位值：0xF0000000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31:23	保留	必须保持复位值
22:20	TS	发送进程状态 指示发送进程的当前状态： 000：停止；发出重置或停止发送命令 001：正在运行，获取发送描述符 010：正在运行，等待发送结束 011：运行，将数据缓冲区从主机内存发送到 FIFO 100：保留 101：正在运行，设置数据包 110：暂停；FIFO 下溢或不可用描述符 111：正在运行，正在关闭发送描述符
19:17	RS	接收进程状态 指示接收进程的当前状态： 000：停止；发出重置或停止接收命令 001：正在运行，获取接收描述符 010：正在运行，在预取下一个描述符之前等待接收结束包 011：正在运行，等待接收数据包 100：暂停，接收缓冲区不可用 101：正在运行，正在关闭接收描述符 110：保留 111：运行，将数据从 FIFO 发送到主机内存

位/位域	名称	描述
16	NIS	正常中断摘要 该位是以下位的逻辑 OR： bit0：发送中断 bit2：发送缓冲区不可用 bit6：接收中断 bit11：通用定时器溢出 bit14：早期接收中断 只有未屏蔽位影响正常中断摘要位。 用户可以通过写入 1 来清除该位。写入 0 无效。
15	AIS	异常中断摘要 该位是以下位的逻辑 OR bit1：发送进程已停止 bit5：发送下溢。 bit7：接收缓冲区不可用 bit8：接收进程已停止 bit10：早期发送中断 只有未屏蔽位影响异常中断摘要位。 用户可以通过写入 1 来清除该位。写入 0 无效。
14	ERI	早期接收中断 当 MAC 填充第一个描述符的数据缓冲区时设置。 用户可以通过写入 1 来清除该位。写入 0 无效。
13:12	保留	必须保持复位值
11	GTE	通用定时器到期

位/位域	名称	描述
		在通用计时器达到零值时设置。 用户可以通过写入 1 来清除该位。写入 0 无效。
10	ETI	早期发送中断 指示要发送的数据包已完全发送到 FIFO。 用户可以通过写入 1 来清除该位。写入 0 无效。
9	保留	必须保持复位值
8	RPS	接收进程已停止 RPS 是在接收进程进入停止状态时设置的。 用户可以通过写入 1 来清除该位。写入 0 无效。
7	RU	接收缓冲区不可用 设置后，表示下一个接收描述符归主机所有，对于 MAC 不可用。当 RU 被设置时，MAC 进入一个挂起状态，当主机改变描述符的所有权时返回接收描述符处理。要么发出一个 receive-poll-demand 命令，要么 MAC 识别一个新帧。 用户可以通过写入 1 来清除该位。写入 0 无效。
6	RI	接收中断 表示接收帧的结束。整个帧已经被传送到接收缓冲区。RI 位的断言可以使用接收中断缓解计数器/定时器（0x58.NRP/RT）延迟。 用户可以通过写入 1 来清除该位。写入 0 无效。
5	UNF	发送下溢 表示发送过程中发送 FIFO 为空。发送进程进入暂停状态。 用户可以通过写入 1 来清除该位。写入 0 无效。

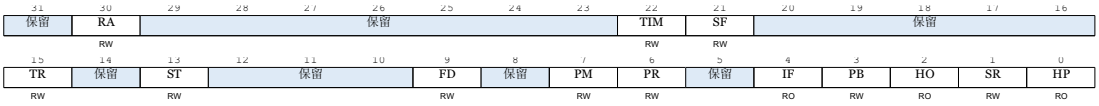
位/位域	名称	描述
4:3	保留	必须保持复位值
2	TU	发送缓冲区不可用 设置后，TU 表示主机拥有发送描述符列表中的下一个描述符；因此，MAC 无法使用。设置了 TU 后，发送进程进入挂起状态，当主机改变描述符的所有权时，可以恢复正常的描述符处理。要么执行 transmit-poll-demand 命令，要么启用发送自动轮询。 用户可以通过写入 1 来清除该位。写入 0 无效。
1	TPS	发送进程已停止 当发送进程进入停止状态时设置 TPS。 用户可以通过写入 1 来清除该位。写入 0 无效。
0	TI	发送中断 表示帧发送过程的结束。TI 位的断言可以使用发送中断缓解计数器/定时器延迟。 用户可以通过写入 1 来清除该位。写入 0 无效

26.4.7 操作模式寄存器（MAC_OPT_MODE）

地址偏移：0x30

复位值：0x32000040

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31	保留	必须保持复位值
30	RA	接收所有

位/位域	名称	描述
		设置后，无论目的地址是什么，所有传入的帧都会被接收。执行地址检查，并将检查结果写入接收描述符（RDES0.30）。
29:23	保留	必须保持复位值
22	TTM	发送阈值模式 1：发送 FIFO 阈值设置为 100Mbps 模式 0：发送 FIFO 阈值设置为 10Mbps 模式 在 RMII 模式下，该位还用于选择 2.5MHz 和 25MHz 之间的发射和接收时钟频率。 只有当发送进程处于停止状态时，才能更改此位。
21	SF	存储和转发 设置此阈值后，无论当前 FIFO 的阈值是多少，发送 FIFO 的报文写满后才开始发送。 此位只能在发送进程处于停止状态时更改。
20:16	保留	必须保持复位值
15	TR	阈值控制位 这些位，连同 TTM，SF 和 PS，控制发送 FIFO 的阈值水平。
14	保留	必须保持复位值
13	ST	启动/停止发送命令 当发送进程处于停止状态时设置此位将导致转换到运行状态。在运行状态下，MAC 在当前描述符列表位置检查发送描述符。如果 MAC 拥有描述符，那么数据开始从内存发送到内部发送 FIFO。如果主机拥有描述符，MAC 将进入挂起状态。

位/位域	名称	描述
		当发送进程处于运行或挂起状态时清除该位将指示 MAC 进入停止状态。 MAC 清除 ST 位后不会立即进入停止状态；它将完成与当前描述符对应的帧数据的发送，然后进入停止状态。应该读取状态寄存器的状态位，以检查实际的发送操作状态。在给出停止发送命令之前，可以检查状态寄存器中的发送状态机。如果发送状态机处于挂起状态，可以发出停止发送命令，保证 MAC 完成帧发送。
12:10	保留	必须保持复位值
9	FD	全双工模式 0：半双工模式 1：强制全双工模式 只有当发送端和接收端进程都处于停止状态时，才允许改变这个位。
8	保留	必须保持复位值
7	PM	通过所有组播 设置后，无论地址检查结果如何，都将接收所有具有组播目的地址的帧。
6	PR	混合模式 设置后，无论地址检查结果如何，所有帧都将被接收。不执行地址检查。
5	保留	必须保持复位值
4	IF	逆滤波（只读） 如果在完美过滤模式下设置此位，则接收端在地址检查过程中进行反过滤。 设置帧的“过滤类型”位决定了该位的状态。

位/位域	名称	描述
3	PB	通过坏帧 设置后，无论接收错误与否，MAC 接口都将所有帧发送到数据缓冲区。这允许接收短帧、碰撞碎片和截断的帧。
2	HO	仅哈希过滤模式（只读） 设置后，Core10100_AHBAPB 对组播和物理地址都执行不完美的过滤。 设置帧的“过滤类型”位决定了该位的状态。
1	SR	启动/停止接收命令 设置此位使 MAC 能够接收帧，并将帧写入接收 FIFO。如果没有使能该位，则帧不写入接收 FIFO。 当接收进程处于停止状态时设置此位将导致转换到运行状态。在运行状态下，MAC 在当前描述符列表位置检查接收描述符。如果 MAC 拥有描述符，它可以处理传入的帧。 当主机拥有描述符时，接收端进入暂停状态，同时设置状态寄存器 bit7（接收缓冲区不可用）位。 当接收进程处于运行或挂起状态时清除该位将指示 MAC 在接收到当前帧后进入停止状态。 MAC 清除 SR 位后不会立即进入停止状态。MAC 将在进入停止状态之前完成所有挂起的接收操作。应该读取状态寄存器的状态位，以检查实际的接收操作状态。
0	HP	哈希/完美接收过滤模式（只读） 0：根据设置帧中指定的物理地址，对传入帧进行完美的过滤。

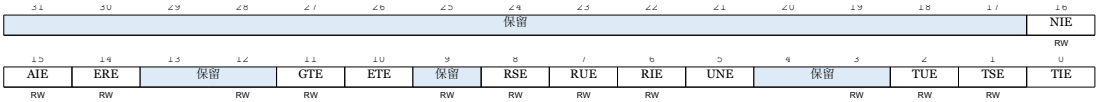
位/位域	名称	描述
		1：根据设置帧中指定的哈希表，对具有组播地址的帧进行不完美的过滤。 根据 HO（仅哈希）位执行物理地址检查。当 HO 和 HP 位都被设置时，对所有的地址执行一个不完美的过滤。 设置帧的“过滤类型”位决定了该位的状态。

26.4.8 中断使能寄存器（MAC_INT_ENABLE）

地址偏移：0x38

复位值：0xF3FE0000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31:17	保留	必须保持复位值
16	NIE	正常中断汇总启用 设置后，正常中断被启用。正常中断列出如下： 状态寄存器 bit0：发送中断 状态寄存器 bit2：发送缓冲区不可用 状态寄存器 bit6：接收中断 状态寄存器 bit11：通用计时器过期 状态寄存器 bit14：早期的接收中断。
15	AIE	设置后，将启用异常中断。异常中断列表如下： 0x28 bit1：发送过程中停止 0x28 bit5：发送下溢 0x28 bit7：接收缓冲区不可用 0x28 bit8：接收过程停止 0x28 bit10：早期发送中断

位/位域	名称	描述
14	ERE	早期接收中断启用 当同时设置 ERE 位和 NIE 位时，使能早期接收中断。
13:13	保留	必须保持复位值
11	GTE	通用定时器溢出启用 当 GTE 位和 NIE 位同时设置时，使能通用定时器溢出中断。
10	ETE	早期中断启动 当设置了 ETE 位和 AIE 位时，早期中断被启用。
9	保留	必须保持复位值
8	RSE	接收停止中断使能 当同时设置 RSE 位和 AIE 位时，使能接收停止中断。
7	RUE	接收缓冲区不可用中断使能 当 RUE 位和 AIE 位同时设置时，接收缓冲区不可用。
6	RIE	接收中断使能 当同时设置 RIE 位和 NIE 位时，接收中断被启用。
5	UNE	发送下溢中断使能 当同时设置了 UNE 位和 AIE 位时，发送下溢中断被启用。
4:3	保留	必须保持复位值
2	TUE	发送缓冲区不可用中断使能 当 TUE 位和 NIE 位同时设置时，发送缓冲区不可用中断被启用。
1	TSE	发送停止中断使能 当 TSE 位和 AIE 位同时设置时，发送过程停止中断被启用。
0	TIE	发送中断使能

位/位域	名称	描述
		当 TIE 位和 NIE 位同时设置时，使能发送中断。

26.4.9 漏帧和计数器下溢状态寄存器（MAC_FRAME_CNT）

地址偏移：0x40

复位值：0xE0000000

该寄存器只支持按字（32 位）访问

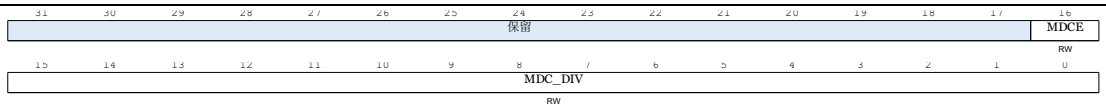
位/位域	名称	描述
31:29	保留	必须保持复位值
28	OCO	溢出计数器溢出（只读） 在 FIFO 溢出计数器溢出时设置。 当读取高字节（bits31:24）时重置。
27:17	FOC	FIFO 溢出计数器（只读） 计数由于接收 FIFO 溢出而未接受的帧数。 当读取高字节（bits31:24）时，计数器重置。
16	MFO	漏帧溢出（只读） 当漏帧计数器溢出时设置。 当读取高字节（bits31:24）时，计数器重置。
15:0	MFC	漏帧计数器（只读） 计算由于接收描述符不可用而未被接受的帧数。 当读取高字节（bits31:24）时，计数器重置。当内部帧缓存已满且描述符不可用时，缺失的帧计数器会增加。

26.4.10 MDIO 配置寄存器（MAC_MDIO_CTRL0）

地址偏移：0x48

复位值：0x00000032

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31:17	保留	必须保持复位值
16	MDCE	MDIO 使能位。当改位有效时 MDC 输出时钟
15:0	MDC_DIV	MDC 的分频器，根据 MAC 所在总线进行分频。当设置的值小于 10 时，硬件强制改为 10。 分频的时钟频率为：PCLK/MDC_DIV。

Warning

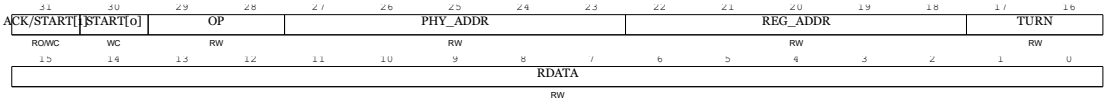
MDC 为 MDIO 接口时钟信号，用来对 MDIO 的数据采样。MDC 可以是非周期性信号。协议对 MDC 的高低电平的最大宽度并没有限制，仅限制了 MDC 的最小高低电平宽度不能小于 160ns，最小周期不能小于 400ns。这意味着 MDC 最高频率不能超过 2.5MHz

26.4.11 MDIO 配置寄存器（MAC_MDIO_CTRL1）

地址偏移：0x50

复位值：0x3FFF 0000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31	ACK/START[1]	当写该寄存器时，其含义为 START 的高位。当读该寄存器时，其含义为读取数据的 ACK 信号。
30	START[0]	当写该寄存器时，其含义为 START 的低位。当读该寄存器时，读出的值为 0。
29:28	RDWR	读写标志 10:表示读操作，01:表示写操作
27:23	PHY_ADDR	PHY 芯片的地址数据

位/位域	名称	描述
22:18	REG_ADDR	PHY 芯片的寄存器地址
17:16	TURN	当执行写操作是该值必须写 10。当执行 MDIO 读操作时可以写 00，也可以写 10。
15:0	RDATA	MDIO 读取的数据信息

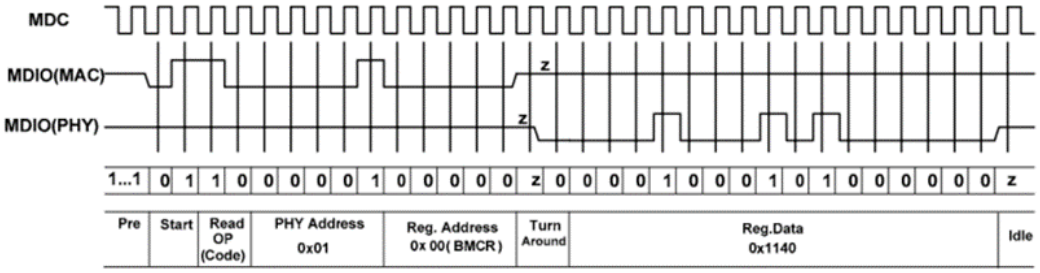


图 26.10 MDC/MDIO 的读数据时序图

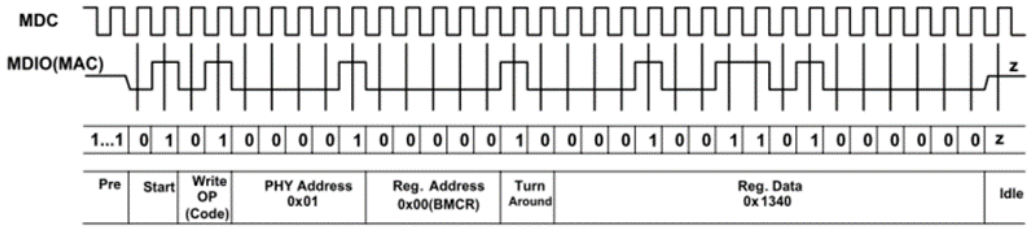


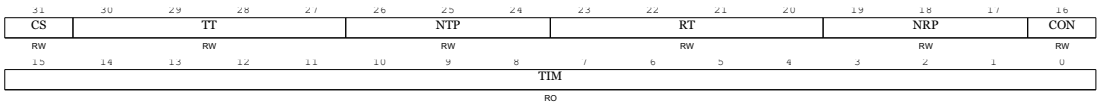
图 26.11 MDC/MDIO 的写数据时序图

26.4.12 通用定时器和中断抑制控制寄存器（MAC_TIMER_CTRL）

地址偏移：0x58

复位值：0xFFFE0000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31	CS	周期大小 根据以下方法控制发送和接收计时器的时间单位： 1： MII 100Mbps 模式：5.12μs

位/位域	名称	描述
		MII 10Mbps 模式：51.2μs 0： MII 100Mbps 模式：81.92μs MII 10Mbps 模式：819.2μs
30:27	TT	发送计时器 控制从发送操作结束到状态寄存器状态位设置之间必须经过的最长时间。 TI（发送中断）位。 这个时间等于 TT*（16*CS）。 当写入非零值时，发送计时器被启用。在每一帧发送后，如果定时器还没有启动，则开始倒数。它会在每一发送帧后重新加载。 向此字段写入 0 将禁用发送中断缓解机制上的定时器效应。 读取该字段将给出计时器的实际计数值。
26:24	NTP	发送报文数 在设置状态寄存器状态位之前控制发送的最大帧数。TI（发送中断）位。 当写入非零值时，启用发送计数器。它在每一发送帧之后递减。它会在设置状态寄存器 TI 位之后重新加载。 向此字段写入 0 将禁用发送中断缓解机制上的反效果。 读取此字段将给出计数器的实际计数值。
23:20	RT	接收计时器 控制从接收操作结束到状态寄存器状态位设置之间必须经过的最长时间。RI（接收中断）位。 这个时间等于 RT*CS。

位/位域	名称	描述
		当使用非零值写入时，接收定时器被启用。每接收一帧后，如果计时器还没有开始计时，则开始倒数。它会在每一个接收到的帧后重新加载。 向该字段写入 0 将禁用接收中断缓解机制上的定时器效应。 读取该字段将给出计时器的实际计数值。
19:17	NRP	接收报文数 在设置状态寄存器之前控制接收帧的最大数量。RI（接收中断）位。 使用非零值写入时启用接收计数器。它在每接收到一帧后递减。它在设置状态寄存器 RI 位后重新加载。 向该字段写入 0 将禁用接收中断缓解机制上的定时器效应。 读取此字段将给出计数器的实际计数值。
16	CON	连续方式 1：通用定时器工作在连续模式 0：通用定时器工作在一次性模式 这个位必须总是在写入计时器值之前被写入。
15:0	TIM	计时器值 包含通用计时器的迭代次数。每次迭代持续时间如下： MII 100Mbps 模式—81.92μs MII 10Mbps 模式—819.2μs

27 直接存储访问（DMA）

27.1 简介

直接存储器访问(Direct Memory Access, DMA)控制器用于在外设和存储器之间，或存储器与存储器之间的高速数据传输，无需 CPU 干预，数据可以通过 DMA 快速地移动，提升了微处理器的性能。 单个 DMA 有 16 个通用通道用于不同类型的外设，支持 USART,IIC,SPI,ADC,TIM。还有一个轮询仲裁器来协调通道间的优先级。

27.2 特性

- 16 个独立的可配置通道
- 每个 DMA 通道都支持 USART,IIC,SPI,ADC,TIM 的 DMA 请求，每个通道上都同样支持存储器到存储器搬移操作。该配置可通过软件完成
- 在同一个 DMA 模块上，多个请求间的优先级可以通过软件编程设置(4 个等级分别为：很高，高，中等和低)，优先级设置相等时由硬件决定(通道 0 优先级最高，通道 15 优先级最低，依次类推)
- 独立数据源和目标数据区的传输大小(字节，半字，字)，源/目标地址必须按数据传输宽度对齐
- 每个通道都支持 DMA 传输完成和 DMA 传输错误等状态标志位和中断，这些状态标志逻辑或成为一个单独的中断请求
- 存储器和存储器间的传输，外设和外设间的传输
- 外设和存储器、存储器和外设间的传输
- SRAM、外设、Flash(只读)均可作为访问的源和目标
- 可编程的数据传输字节长度：最大为 32767 bytes
- 可选的双核锁步的安全备份
- 内置超时看门狗

27.3 结构框图

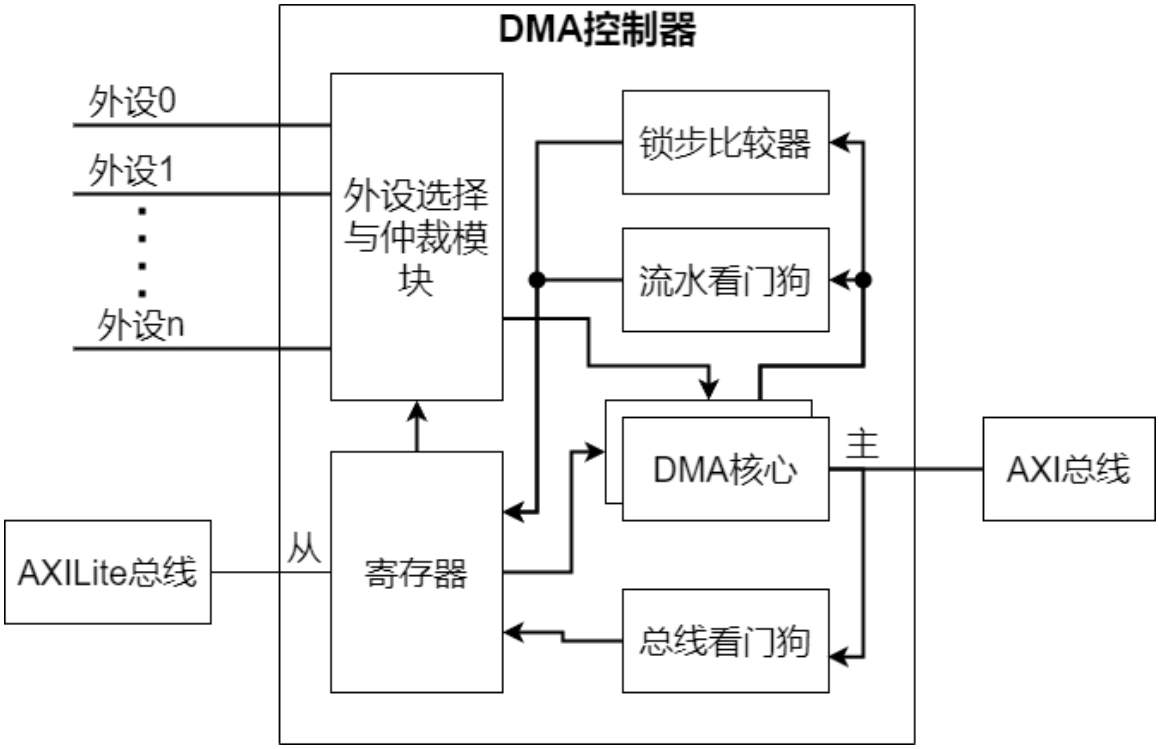


图 27.1 DMA 设计架构图

27.4 功能说明

27.4.1 DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA_CHxPADDR、DMA_CHxMADDR、DMA_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA_CHxCNT 寄存器用于控制传输的次数。DMA_CHxCTL 寄存器的 PWIDTH 和 MWIDTH 位域决定每次发送和接收的字节数（字节/半字/字）。 假设 DMA_CHxCNT 寄存器的值为 4，并且 PNAGA 和 MNAGA 位均置位。结合 PWIDTH 和 MWIDTH 的各种配置，DMA 传输的操作详见下表。

表 27.1 DMA 传输操作

传输宽度		传输操作	
源	目标	源	目标

传输宽度		传输操作	
32bits	32bits	1：Read B3B2B1B0[31:0] @0x0 2：Read B7B6B5B4[31:0] @0x4 3：Read BBBAB9B8[31:0] @0x8 4：Read BFBEBDBC[31:0] @0xC	1：Write B3B2B1B0[31:0] @0x0 2：Write B7B6B5B4[31:0] @0x4 3：Write BBBAB9B8[31:0] @0x8 4：Write BFBEBDBC[31:0] @0xC
32bits	16bits	1：Read B3B2B1B0[31:0] @0x0 2：Read B7B6B5B4[31:0] @0x4 3：Read BBBAB9B8[31:0] @0x8 4：Read BFBEBDBC[31:0] @0xC	1：Write B1B0[15:0] @0x0 2：Write B5B4[15:0] @0x2 3：Write B9B8[15:0] @0x4 4：Write BDBC[15:0] @0x6
32bits	8bits	1：Read B3B2B1B0[31:0] @0x0 2：Read B7B6B5B4[31:0] @0x4 3：Read BBBAB9B8[31:0] @0x8 4：Read BFBEBDBC[31:0] @0xC	1：Write B0[7:0] @0x0 2：Write B4[7:0] @0x1 3：Write B8[7:0] @0x2 4：Write BC[7:0] @0x3
16bits	32bits	1：Read B1B0[15:0] @0x0 2：Read B3B2[15:0] @0x2 3：Read B5B4[15:0] @0x4	1：Write 0000B1B0[31:0] @0x0 2：Write 0000B3B2[31:0] @0x4 3：Write 0000B5B4[31:0] @0x8

传输宽度		传输操作	
		4：Read B7B6[15:0] @0x6	4：Write 0000B7B6[31:0] @0xC
16bits	16bits	1：Read B1B0[15:0] @0x0 2：Read B3B2[15:0] @0x2 3：Read B5B4[15:0] @0x4 4：Read B7B6[15:0] @0x6	1：Write B1B0[15:0] @0x0 2：Write B3B2[15:0] @0x2 3：Write B5B4[15:0] @0x4 4：Write B7B6[15:0] @0x6
16bits	8bits	1：Read B1B0[15:0] @0x0 2：Read B3B2[15:0] @0x2 3：Read B5B4[15:0] @0x4 4：Read B7B6[15:0] @0x6	1：Write B0[7:0] @0x0 2：Write B2[7:0] @0x1 3：Write B4[7:0] @0x2 4：Write B6[7:0] @0x3
8bits	32bits	1：Read B0[7:0] @0x0 2：Read B1[7:0] @0x1 3：Read B2[7:0] @0x2 4：Read B3[7:0] @0x3	1：Write 000000B0[31:0] @0x0 2：Write 000000B1[31:0] @0x4 3：Write 000000B2[31:0] @0x8 4：Write 000000B3[31:0] @0xC
8bits	16bits	1：Read B0[7:0] @0x0 2：Read B1[7:0] @0x1 3：Read B2[7:0] @0x2 4：Read B3[7:0] @0x3	1：Read 00B0[15:0] @0x0 2：Read 00B1[15:0] @0x1 3：Read 00B2[15:0] @0x2 4：Read 00B3[15:0] @0x3
8bits	8bits	1：Read B0[7:0] @0x0 2：Read B1[7:0] @0x1 3：Read B2[7:0] @0x2 4：Read B3[7:0] @0x3	1：Write B0[7:0] @0x0 2：Write B1[7:0] @0x1 3：Write B2[7:0] @0x2 4：Write B3[7:0] @0x3

DMA_CHxCNT 寄存器的 CNT 位域必须在 CHEN 位置位前被配置，其控制传输的次数。在传输过程中，CNT 位域的值表示还有多少次数据传输将被执行。 将 DMA_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输。

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：

- 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输。
- 在重新使能 DMA 通道前，对任意相关寄存器进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

27.4.2 仲裁

当 DMA 控制器在同一时间接收到多个外设请求时,仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级,优先级规则如下:

- 软件优先级:分为 4 级,低,中,高和极高。可以通过寄存器 DMA_CHxCTL 的 PRIO 位域来配置。
- 硬件优先级:当通道具有相同的软件优先级时,编号低的通道优先级高。例:通道 0 和通道 2 配置为相同的软件优先级时,通道 0 的优先级高于通道 2。

27.4.3 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器 DMA_CHxCTL 的 PNAGA 和 MNAGA 位用来设置存储器和外设的地址生成算法。在固定模式中，地址一直固定为初始化的基地址（DMA_CHxPADDR，DMA_CHxMADDR）。 在增量模式中，下一次传输数据的地址是当前地址加 1（或 2、4），这个值取决于数据传输宽度。

27.4.4 循环模式

循环模式用来处理连续的外设请求（如 ADC 扫描模式）。将 DMA_CHxCTL 寄存器的 CMEN 位置位可以使能循环模式。 在循环模式中，当每次 DMA 传输完成后，CNT 值会被重新载入，且传输完成标志位会被置 1。DMA 会一直响应外设的请求，直到通道使能位（DMA_CHxCTL 寄存器的 CHEN 位）被清 0。

27.4.5 存储器到存储器模式

将 DMA_CHxCTL 寄存器的 DIR 位与 M2M 位同时置位可以使能存储器到存储器模式。在此模式下，DMA 通道传输数据时不依赖外设的请求信号。一旦 DMA_CHxCTL 寄存器的 CHEN 位被置 1，DMA 通道就立即开始传输数据，直到 DMA_CHxCNT 寄存器达到 0，DMA 通道才会停止。

27.5 应用说明

27.5.1 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 寄存器的 CHx 位，如果为 1（通道已使能），清零该位。当 CHx 为 0 时，请按照下列步骤配置 DMA 开始新的传输。
2. 配置 DMA_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式。
3. 配置 DMA_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式。
4. 配置 DMA_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级。
5. 通过 DMA_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法。
6. 通过 DMA_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位。
7. 通过 DMA_CHxPADDR 寄存器配置外设数据寄存器地址。
8. 通过 DMA_CHxMADDR 寄存器配置存储器基地址。
9. 通过 DMA_CHxCNT 寄存器配置数据传输总量。
10. 配置 DMA 请求映射，若为 M2M 模式则无需配置。
11. 将 DMA_CHEN 寄存器的 ENx 位置 1，使能 DMA 通道。 注意：对于大多数关于 DMA 的应用（包括内存到内存，关于 UART、SPI、ADC、TIM 的外设操作）需对外设进行适当配置并根据上述步骤配置 DMA。

27.5.2 双核锁步

DMA 控制器本身为一双核锁步的控制单元，该设计保证 DMA 作为总线主机的功能安全，可根据需求进行配置。 要启动双核锁步，建议遵循以下步骤进行操作：

1. 读取 CHEN 寄存器，如果不是为 0，清零该寄存器。当 CHEN 寄存器为 0 时，请按照下列步骤配置双核锁步。
2. 将 DMA_CCR 寄存器的 DEN 位置 1。
3. 读取 DMA_CCR 寄存器的 DRDY 位，如果不为 1，继续轮询直到为 1。如果为 1，双核锁步配置完成，根据通道配置说明配置通道进行 DMA 传输。

27.5.3 内置看门狗

DMA 控制器内置一看门狗，以防止 DMA 因硬件或软件原因卡死。 要配置内置看门狗，建议遵循以下步骤进行操作：

1. 读取 CCR 寄存器的 WDEN 位，如果为 1（看门狗已使能），清零该位。当 WDEN 为 1 时，如下继续配置。
2. 配置 CCR 寄存器的 WDCNT 位，选择看门狗的狗叫等待时间。
3. 将 CCR 寄存器的 WDEN 位置 1，使能看门狗。

27.5.4 中断

每个 DMA 通道都有一个专用的中断。中断事件有四种类型：传输完成，半传输完成，传输错误和看门狗超时错误。每一个通道的中断事件有专用的原始数据位、标志位、强制赋值位与遮罩位。其对应关系如下表：

表 27.2 DMA 中断事件

位类型	中断类型	对应标志
遮罩位	传输完成	FTFMASK
	半传输完成	HTFMASK
	传输错误	ERRMASK

位类型	中断类型	对应标志
	看门狗触发	WDGMASK
原始数据位	传输完成	FTFRAW
	半传输完成	HTFRAW
	传输错误	ERRRAW
	看门狗触发	WDGRAW
状态位	传输完成	FTFSTA
	半传输完成	HTFSTA
	传输错误	ERRSTA
	看门狗触发	WDGSTA

27.5.5 DMA 请求映射

通过修改 PTCH 寄存器的 CHxPT 位，可配置外设将 DMA 请求映射到相应的通道。

27.5.5.1 DMA0 连接表

表 27.3 DMA0 连接表

信号源标号	外设信号源
0	TIM0
1	ADC0
2	ADC1
3	SPI0
4	SPI1
5	SPI2
6	保留
7	保留
8	USART0
9	USART1
10	USART2

信号源标号	外设信号源
11	USART3
12	TIM2
13	TIM3
14	TIM4

27.5.5.2 DMA1 连接表

表 27.4 DMA1 连接表

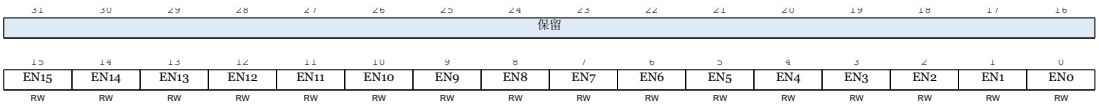
信号源标号	外设信号源
0	TIM1
1	ADC2
2	SPI3
3	SPI4
4	SPI5
5	保留
6	保留
7	USART4
8	USART5
9	USART6
10	USART7
11	TIM5
12	TIM6
13	TIM7

27.6 寄存器

27.6.1 DMA 通道使能寄存器（DMA_CHEN）

地址偏移：0x0

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	EN15	通道 15 使能 0:关闭通道 15 1:使能通道 15
14	EN14	通道 14 使能 0:关闭通道 14 1:使能通道 14
13	EN13	通道 13 使能 0:关闭通道 13 1:使能通道 13
12	EN12	通道 12 使能 0:关闭通道 12 1:使能通道 12
11	EN11	通道 11 使能 0:关闭通道 11 1:使能通道 11
10	EN10	通道 10 使能 0:关闭通道 10 1:使能通道 10
9	EN9	通道 9 使能 0:关闭通道 9 1:使能通道 9
8	EN8	通道 8 使能 0:关闭通道 8 1:使能通道 8
7	EN7	通道 7 使能 0:关闭通道 7 1:使能通道 7
6	EN6	通道 6 使能 0:关闭通道 6 1:使能通道 6
5	EN5	通道 5 使能 0:关闭通道 5 1:使能通道 5
4	EN4	通道 4 使能 0:关闭通道 4 1:使能通道 4
3	EN3	通道 3 使能 0:关闭通道 3 1:使能通道 3
2	EN2	通道 2 使能 0:关闭通道 2 1:使能通道 2
1	EN1	通道 1 使能 0:关闭通道 1 1:使能通道 1
0	EN0	通道 0 使能 0:关闭通道 0 1:使能通道 0

27.6.2 DMA 通道 x 控制寄存器（DMA_CHxCTL）

地址偏移：0x4 + 0x10*x

复位值：0x00000000

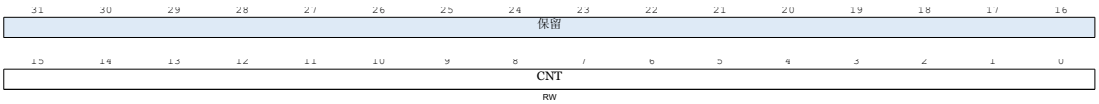
位/位域	名称	描述
31:11	保留	必须保持复位值
10:9	PRIO	软件优先级 00:低优先级 01:中优先级 10:高优先级 11:极高优先级 CHEN_EN0 为 1 时，该位不能被配置

位/位域	名称	描述
8:7	MWIDTH	内存的数据传输宽度 00:8-bit 01:16-bit 10:32-bit 11:保留 CHEN_EN0 为 1 时，该位不能被配置
6:5	PWIDTH	外设的数据传输宽度 00:8-bit 01:16-bit 10:32-bit 11:保留 CHEN_EN0 为 1 时，该位不能被配置
4	MNAGA	内存的地址生成算法 0:固定地址模式 1:增量地址模式 CHEN_EN0 为 1 时，该位不能被配置
3	PNAGA	外设的地址生成算法 0:固定地址模式 1:增量地址模式 CHEN_EN0 为 1 时，该位不能被配置
2	CMEN	循环模式使能 0:禁止循环模式 1:使能循环模式 CHEN_EN0 为 1 时，该位不能被配置
1	DIR	传输方向 0:从外设读出并写入存储器 1:从存储器读出并写入外设/M2M 模式 CHEN_EN0 为 1 时，该位不能被配置
0	M2M	内存到内存使能 0:禁止存储器到存储器模式 1:使能存储器到存储器模式 CHEN_EN0 为 1 时，该位不能被配置

27.6.3 DMA 通道 x 计数寄存器（DMA_CHxCNT）

地址偏移：0x8 + 0x10*x

复位值：0x00000000



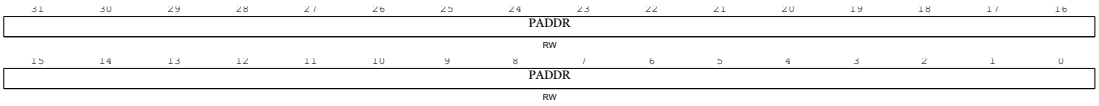
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PRIO	传输计数 CHEN_ENx 为 1 时，该位不能被配置 该寄存器标明还有多少数据等待被传输。一旦通道使能，该寄存器为只读，并在每个 DMA 传输之后值减 1。

位/位域	名称	描述
		如果该寄存器的值为 0，无论通道开启与否，都不会有数据传输。如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会被自动重装载为初始设置值。

27.6.4 DMA 通道 x 外设地址寄存器（DMA_CHxPADDR）

地址偏移：0xC + 0x10*x

复位值：0x00000000

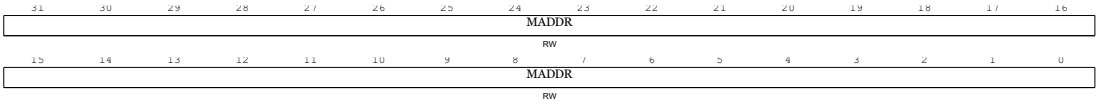


位/位域	名称	描述
31:0	PADDR	外设基地址 CHEN_ENx 为 1 时，该位不能被配置 当 PWIDTH 位域的值为 01（16-bit），PADDR[0]被忽略，访问自动与 16 位地址对齐。 当 PWIDTH 位域的值为 10（32-bit），PADDR[1:0]被忽略，访问自动与 32 位地址对齐。

27.6.5 DMA 通道 x 内存地址寄存器（DMA_CHxMADDR）

地址偏移：0x10 + 0x10*x

复位值：0x00000000



位/位域	名称	描述
31:0	MADDR	内存基地址 CHEN_ENx 为 1 时，该位不能被配置 当 MWIDTH 位域的值为 01（16-bit），MADDR[0]被忽略，访问自动与 16 位地址对齐。 当 MWIDTH 位域的值为 10（32-bit），MADDR[1:0]被忽略，访问自动与 32 位地址对齐。

27.6.6 DMA 通道外设信号分配寄存器 0（DMA_PTCH0）

地址偏移：0x104

复位值：0x00000000

位/位域	名称	描述
31:28	CH7PT	通道 7 外设接口 配置数字为 DMA 表格上对应的外设
27:24	CH6PT	通道 6 外设接口 配置数字为 DMA 表格上对应的外设
23:20	CH5PT	通道 5 外设接口 配置数字为 DMA 表格上对应的外设
19:16	CH4PT	通道 4 外设接口 配置数字为 DMA 表格上对应的外设
15:12	CH3PT	通道 3 外设接口 配置数字为 DMA 表格上对应的外设
11:8	CH2PT	通道 2 外设接口 配置数字为 DMA 表格上对应的外设
7:4	CH1PT	通道 1 外设接口 配置数字为 DMA 表格上对应的外设
3:0	CH0PT	通道 0 外设接口 配置数字为 DMA 表格上对应的外设

27.6.7 DMA 通道外设信号分配寄存器 1（DMA_PTCH1）

地址偏移：0x108

复位值：0x00000000

位/位域	名称	描述
31:28	CH15PT	通道 15 外设接口 配置数字为 DMA 表格上对应的外设
27:24	CH14PT	通道 14 外设接口 配置数字为 DMA 表格上对应的外设
23:20	CH13PT	通道 13 外设接口 配置数字为 DMA 表格上对应的外设
19:16	CH12PT	通道 12 外设接口 配置数字为 DMA 表格上对应的外设
15:12	CH11PT	通道 11 外设接口 配置数字为 DMA 表格上对应的外设
11:8	CH10PT	通道 10 外设接口 配置数字为 DMA 表格上对应的外设
7:4	CH9PT	通道 9 外设接口 配置数字为 DMA 表格上对应的外设

位/位域	名称	描述
3:0	CH8PT	通道 8 外设接口 配置数字为 DMA 表格上对应的外设

27.6.8 DMA 安全配置寄存器（DMA_FCR）

地址偏移：0x10C

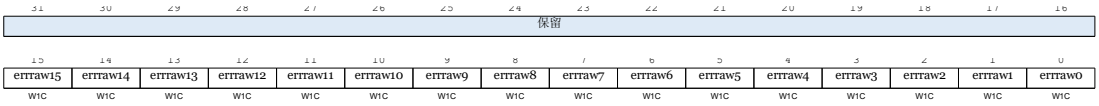
复位值：0x00000000

位/位域	名称	描述
31:20,2:0	保留	必须保持复位值
19:4	WDCNT	看门狗计数阈值 在对应周期看门狗未解除通道锁定后，看门狗报错
3	WDEN	看门狗使能 0:关闭 DMA 看门狗 1:打开 DMA 看门狗

27.6.9 DMA 错误中断原始数据寄存器（DMA_ERRINT_RAW）

地址偏移：0x110

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	errraw15	错误中断 15 记录原始位，当出现错误时置 1，写 1 清 0
14	errraw14	错误中断 14 记录原始位，当出现错误时置 1，写 1 清 0
13	errraw13	错误中断 13 记录原始位，当出现错误时置 1，写 1 清 0
12	errraw12	错误中断 12 记录原始位，当出现错误时置 1，写 1 清 0
11	errraw11	错误中断 11 记录原始位，当出现错误时置 1，写 1 清 0
10	errraw10	错误中断 10 记录原始位，当出现错误时置 1，写 1 清 0
9	errraw9	错误中断 9 记录原始位，当出现错误时置 1，写 1 清 0

位/位域	名称	描述
8	errraw8	错误中断 8 记录原始位，当出现错误时置 1，写 1 清 0
7	errraw7	错误中断 7 记录原始位，当出现错误时置 1，写 1 清 0
6	errraw6	错误中断 6 记录原始位，当出现错误时置 1，写 1 清 0
5	errraw5	错误中断 5 记录原始位，当出现错误时置 1，写 1 清 0
4	errraw4	错误中断 4 记录原始位，当出现错误时置 1，写 1 清 0
3	errraw3	错误中断 3 记录原始位，当出现错误时置 1，写 1 清 0
2	errraw2	错误中断 2 记录原始位，当出现错误时置 1，写 1 清 0
1	errraw1	错误中断 1 记录原始位，当出现错误时置 1，写 1 清 0
0	errraw0	错误中断 0 记录原始位，当出现错误时置 1，写 1 清 0

27.6.10 DMA 错误中断强制赋值寄存器（DMA_ERRINT_FORCE）

地址偏移：0x114

复位值：0x00000000

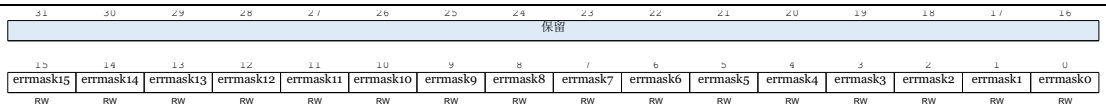
位/位域	名称	描述
31:16	保留	必须保持复位值
15	errforce15	错误中断 15 强制赋值位，用于测试通道连通性，写 1 将 errraw15 赋 1
14	errforce14	错误中断 14 强制赋值位，用于测试通道连通性，写 1 将 errraw14 赋 1
13	errforce13	错误中断 13 强制赋值位，用于测试通道连通性，写 1 将 errraw13 赋 1
12	errforce12	错误中断 12 强制赋值位，用于测试通道连通性，写 1 将 errraw12 赋 1
11	errforce11	错误中断 11 强制赋值位，用于测试通道连通性，写 1 将 errraw11 赋 1

位/位域	名称	描述
10	errforce10	错误中断 10 强制赋值位，用于测试通道连通性，写 1 将 errraw10 赋 1
9	errforce9	错误中断 9 强制赋值位，用于测试通道连通性，写 1 将 errraw9 赋 1
8	errforce8	错误中断 8 强制赋值位，用于测试通道连通性，写 1 将 errraw8 赋 1
7	errforce7	错误中断 7 强制赋值位，用于测试通道连通性，写 1 将 errraw7 赋 1
6	errforce6	错误中断 6 强制赋值位，用于测试通道连通性，写 1 将 errraw6 赋 1
5	errforce5	错误中断 5 强制赋值位，用于测试通道连通性，写 1 将 errraw5 赋 1
4	errforce4	错误中断 4 强制赋值位，用于测试通道连通性，写 1 将 errraw4 赋 1
3	errforce3	错误中断 3 强制赋值位，用于测试通道连通性，写 1 将 errraw3 赋 1
2	errforce2	错误中断 2 强制赋值位，用于测试通道连通性，写 1 将 errraw2 赋 1
1	errforce1	错误中断 1 强制赋值位，用于测试通道连通性，写 1 将 errraw1 赋 1
0	errforce0	错误中断 0 强制赋值位，用于测试通道连通性，写 1 将 errraw0 赋 1

27.6.11 DMA 错误中断遮罩寄存器（DMA_ERRINT_MASK）

地址偏移：0x118

复位值：0x0000ffff



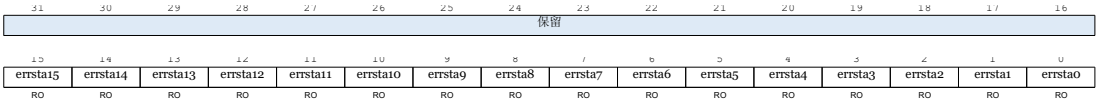
位/位域	名称	描述
31:16	保留	必须保持复位值
15	errmask15	错误中断 15 遮罩位 1:中断关闭 0:中断使能
14	errmask14	错误中断 14 遮罩位 1:中断关闭 0:中断使能
13	errmask13	错误中断 13 遮罩位 1:中断关闭 0:中断使能
12	errmask12	错误中断 12 遮罩位 1:中断关闭 0:中断使能
11	errmask11	错误中断 11 遮罩位 1:中断关闭 0:中断使能
10	errmask10	错误中断 10 遮罩位 1:中断关闭 0:中断使能
9	errmask9	错误中断 9 遮罩位 1:中断关闭 0:中断使能
8	errmask8	错误中断 8 遮罩位 1:中断关闭

位/位域	名称	描述
		0:中断使能
7	errmask7	错误中断 7 遮罩位 1:中断关闭 0:中断使能
6	errmask6	错误中断 6 遮罩位 1:中断关闭 0:中断使能
5	errmask5	错误中断 5 遮罩位 1:中断关闭 0:中断使能
4	errmask4	错误中断 4 遮罩位 1:中断关闭 0:中断使能
3	errmask3	错误中断 3 遮罩位 1:中断关闭 0:中断使能
2	errmask2	错误中断 2 遮罩位 1:中断关闭 0:中断使能
1	errmask1	错误中断 1 遮罩位 1:中断关闭 0:中断使能
0	errmask0	错误中断 0 遮罩位 1:中断关闭 0:中断使能

27.6.12 DMA 错误中断状态寄存器（DMA_ERRINT_STA）

地址偏移：0x11C

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	errsta15	错误中断 15 状态位 1:中断触发 0:中断未触发
14	errsta14	错误中断 14 状态位 1:中断触发 0:中断未触发
13	errsta13	错误中断 13 状态位 1:中断触发 0:中断未触发
12	errsta12	错误中断 12 状态位 1:中断触发 0:中断未触发
11	errsta11	错误中断 11 状态位 1:中断触发 0:中断未触发
10	errsta10	错误中断 10 状态位 1:中断触发 0:中断未触发
9	errsta9	错误中断 9 状态位 1:中断触发

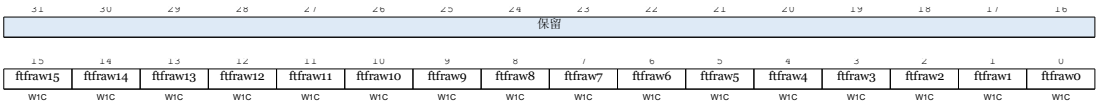
位/位域	名称	描述
		0:中断未触发
8	errsta8	错误中断 8 状态位 1:中断触发 0:中断未触发
7	errsta7	错误中断 7 状态位 1:中断触发 0:中断未触发
6	errsta6	错误中断 6 状态位 1:中断触发 0:中断未触发
5	errsta5	错误中断 5 状态位 1:中断触发 0:中断未触发
4	errsta4	错误中断 4 状态位 1:中断触发 0:中断未触发
3	errsta3	错误中断 3 状态位 1:中断触发 0:中断未触发
2	errsta2	错误中断 2 状态位 1:中断触发 0:中断未触发
1	errsta1	错误中断 1 状态位 1:中断触发 0:中断未触发
0	errsta0	错误中断 0 状态位

位/位域	名称	描述
		1:中断触发 0:中断未触发

27.6.13 DMA 全满中断原始数据寄存器（DMA_FTFINT_RAW）

地址偏移：0x120

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	ftfraw15	全满中断 15 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
14	ftfraw14	全满中断 14 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
13	ftfraw13	全满中断 13 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
12	ftfraw12	全满中断 12 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
11	ftfraw11	全满中断 11 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
10	ftfraw10	全满中断 10 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
9	ftfraw9	全满中断 9 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
8	ftfraw8	全满中断 8 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0

位/位域	名称	描述
7	ftfraw7	全满中断 7 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
6	ftfraw6	全满中断 6 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
5	ftfraw5	全满中断 5 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
4	ftfraw4	全满中断 4 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
3	ftfraw3	全满中断 3 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
2	ftfraw2	全满中断 2 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
1	ftfraw1	全满中断 1 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0
0	ftfraw0	全满中断 0 记录原始位，当通道 CNT 归 0 时置 1，写 1 清 0

27.6.14 DMA 全满中断强制赋值寄存器（DMA_FTFINT_FORCE）

地址偏移：0x124

复位值：0x00000000

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ftfforce15	全满中断 15 强制赋值位，用于测试通道连通性，写 1 将 ftfraw15 赋 1

位/位域	名称	描述
14	ftfforce14	全满中断 14 强制赋值位，用于测试通道连通性，写 1 将 ftfraw14 赋 1
13	ftfforce13	全满中断 13 强制赋值位，用于测试通道连通性，写 1 将 ftfraw13 赋 1
12	ftfforce12	全满中断 12 强制赋值位，用于测试通道连通性，写 1 将 ftfraw12 赋 1
11	ftfforce11	全满中断 11 强制赋值位，用于测试通道连通性，写 1 将 ftfraw11 赋 1
10	ftfforce10	全满中断 10 强制赋值位，用于测试通道连通性，写 1 将 ftfraw10 赋 1
9	ftfforce9	全满中断 9 强制赋值位，用于测试通道连通性，写 1 将 ftfraw9 赋 1
8	ftfforce8	全满中断 8 强制赋值位，用于测试通道连通性，写 1 将 ftfraw8 赋 1
7	ftfforce7	全满中断 7 强制赋值位，用于测试通道连通性，写 1 将 ftfraw7 赋 1
6	ftfforce6	全满中断 6 强制赋值位，用于测试通道连通性，写 1 将 ftfraw6 赋 1
5	ftfforce5	全满中断 5 强制赋值位，用于测试通道连通性，写 1 将 ftfraw5 赋 1
4	ftfforce4	全满中断 4 强制赋值位，用于测试通道连通性，写 1 将 ftfraw4 赋 1
3	ftfforce3	全满中断 3 强制赋值位，用于测试通道连通性，写 1 将 ftfraw3 赋 1
2	ftfforce2	全满中断 2 强制赋值位，用于测试通道连通性，写 1 将 ftfraw2 赋 1

位/位域	名称	描述
1	ftfforce1	全满中断 1 强制赋值位，用于测试通道连通性，写 1 将 ftfraw1 赋 1
0	ftfforce0	全满中断 0 强制赋值位，用于测试通道连通性，写 1 将 ftfraw0 赋 1

27.6.15 DMA 全满中断遮罩寄存器（DMA_FTFINT_MASK）

地址偏移：0x128

复位值：0x0000ffff

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ftfmask15	全满中断 15 遮罩位 1:中断关闭 0:中断使能
14	ftfmask14	全满中断 14 遮罩位 1:中断关闭 0:中断使能
13	ftfmask13	全满中断 13 遮罩位 1:中断关闭 0:中断使能
12	ftfmask12	全满中断 12 遮罩位 1:中断关闭 0:中断使能
11	ftfmask11	全满中断 11 遮罩位 1:中断关闭 0:中断使能

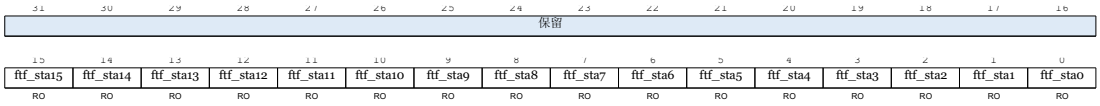
位/位域	名称	描述
10	ftfmask10	全满中断 10 遮罩位 1:中断关闭 0:中断使能
9	ftfmask9	全满中断 9 遮罩位 1:中断关闭 0:中断使能
8	ftfmask8	全满中断 8 遮罩位 1:中断关闭 0:中断使能
7	ftfmask7	全满中断 7 遮罩位 1:中断关闭 0:中断使能
6	ftfmask6	全满中断 6 遮罩位 1:中断关闭 0:中断使能
5	ftfmask5	全满中断 5 遮罩位 1:中断关闭 0:中断使能
4	ftfmask4	全满中断 4 遮罩位 1:中断关闭 0:中断使能
3	ftfmask3	全满中断 3 遮罩位 1:中断关闭 0:中断使能
2	ftfmask2	全满中断 2 遮罩位 1:中断关闭

位/位域	名称	描述
		0:中断使能
1	ftfmask1	全满中断 1 遮罩位 1:中断关闭 0:中断使能
0	ftfmask0	全满中断 0 遮罩位 1:中断关闭 0:中断使能

27.6.16 DMA 全满中断状态寄存器（DMA_FTFINT_STA）

地址偏移：0x12C

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	ftfsta15	全满中断 15 状态位 1:中断触发 0:中断未触发
14	ftfsta14	全满中断 14 状态位 1:中断触发 0:中断未触发
13	ftfsta13	全满中断 13 状态位 1:中断触发 0:中断未触发
12	ftfsta12	全满中断 12 状态位 1:中断触发 0:中断未触发

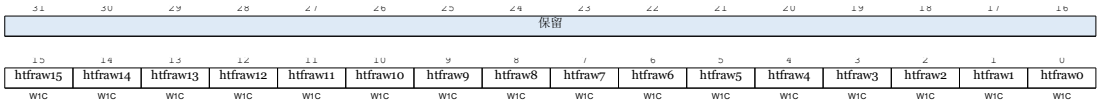
位/位域	名称	描述
11	ftfsta11	全满中断 11 状态位 1:中断触发 0:中断未触发
10	ftfsta10	全满中断 10 状态位 1:中断触发 0:中断未触发
9	ftfsta9	全满中断 9 状态位 1:中断触发 0:中断未触发
8	ftfsta8	全满中断 8 状态位 1:中断触发 0:中断未触发
7	ftfsta7	全满中断 7 状态位 1:中断触发 0:中断未触发
6	ftfsta6	全满中断 6 状态位 1:中断触发 0:中断未触发
5	ftfsta5	全满中断 5 状态位 1:中断触发 0:中断未触发
4	ftfsta4	全满中断 4 状态位 1:中断触发 0:中断未触发
3	ftfsta3	全满中断 3 状态位 1:中断触发

位/位域	名称	描述
		0:中断未触发
2	ftfsta2	全满中断 2 状态位 1:中断触发 0:中断未触发
1	ftfsta1	全满中断 1 状态位 1:中断触发 0:中断未触发
0	ftfsta0	全满中断 0 状态位 1:中断触发 0:中断未触发

27.6.17 DMA 半满中断原始数据寄存器（DMA_HTFINT_RAW）

地址偏移：0x130

复位值：0x00000000



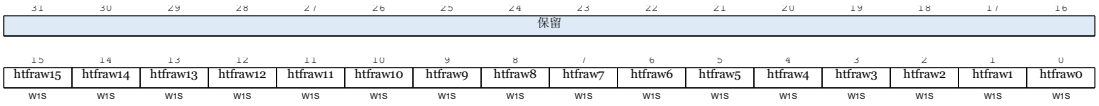
位/位域	名称	描述
31:16	保留	必须保持复位值
15	htfraw15	半满中断 15 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
14	htfraw14	半满中断 14 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
13	htfraw13	半满中断 13 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
12	htfraw12	半满中断 12 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0

位/位域	名称	描述
11	htfraw11	半满中断 11 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
10	htfraw10	半满中断 10 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
9	htfraw9	半满中断 9 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
8	htfraw8	半满中断 8 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
7	htfraw7	半满中断 7 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
6	htfraw6	半满中断 6 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
5	htfraw5	半满中断 5 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
4	htfraw4	半满中断 4 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
3	htfraw3	半满中断 3 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
2	htfraw2	半满中断 2 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
1	htfraw1	半满中断 1 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0
0	htfraw0	半满中断 0 记录原始位，当通道 CNT 计数到一半时置 1，写 1 清 0

27.6.18 DMA 半满中断强制赋值寄存器（DMA_HTFINT_FORCE）

地址偏移：0x134

复位值：0x00000000



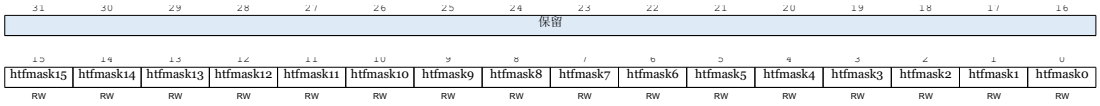
位/位域	名称	描述
31:16	保留	必须保持复位值
15	htfforce15	半满中断 15 强制赋值位，用于测试通道连通性，写 1 将 htfraw15 赋 1
14	htfforce14	半满中断 14 强制赋值位，用于测试通道连通性，写 1 将 htfraw14 赋 1
13	htfforce13	半满中断 13 强制赋值位，用于测试通道连通性，写 1 将 htfraw13 赋 1
12	htfforce12	半满中断 12 强制赋值位，用于测试通道连通性，写 1 将 htfraw12 赋 1
11	htfforce11	半满中断 11 强制赋值位，用于测试通道连通性，写 1 将 htfraw11 赋 1
10	htfforce10	半满中断 10 强制赋值位，用于测试通道连通性，写 1 将 htfraw10 赋 1
9	htfforce9	半满中断 9 强制赋值位，用于测试通道连通性，写 1 将 htfraw9 赋 1
8	htfforce8	半满中断 8 强制赋值位，用于测试通道连通性，写 1 将 htfraw8 赋 1
7	htfforce7	半满中断 7 强制赋值位，用于测试通道连通性，写 1 将 htfraw7 赋 1
6	htfforce6	半满中断 6 强制赋值位，用于测试通道连通性，写 1 将 htfraw6 赋 1
5	htfforce5	半满中断 5 强制赋值位，用于测试通道连通性，写 1 将 htfraw5 赋 1

位/位域	名称	描述
4	htfforce4	半满中断 4 强制赋值位，用于测试通道连通性，写 1 将 htfraw4 赋 1
3	htfforce3	半满中断 3 强制赋值位，用于测试通道连通性，写 1 将 htfraw3 赋 1
2	htfforce2	半满中断 2 强制赋值位，用于测试通道连通性，写 1 将 htfraw2 赋 1
1	htfforce1	半满中断 1 强制赋值位，用于测试通道连通性，写 1 将 htfraw1 赋 1
0	htfforce0	半满中断 0 强制赋值位，用于测试通道连通性，写 1 将 htfraw0 赋 1

27.6.19 DMA 半满中断遮罩寄存器（DMA_HTFINT_MASK）

地址偏移：0x138

复位值：0x0000ffff



位/位域	名称	描述
31:16	保留	必须保持复位值
15	htfmask15	半满中断 15 遮罩位 1:中断关闭 0:中断使能
14	htfmask14	半满中断 14 遮罩位 1:中断关闭 0:中断使能
13	htfmask13	半满中断 13 遮罩位 1:中断关闭 0:中断使能

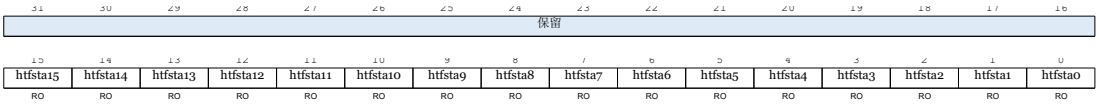
位/位域	名称	描述
12	htfmask12	半满中断 12 遮罩位 1:中断关闭 0:中断使能
11	htfmask11	半满中断 11 遮罩位 1:中断关闭 0:中断使能
10	htfmask10	半满中断 10 遮罩位 1:中断关闭 0:中断使能
9	htfmask9	半满中断 9 遮罩位 1:中断关闭 0:中断使能
8	htfmask8	半满中断 8 遮罩位 1:中断关闭 0:中断使能
7	htfmask7	半满中断 7 遮罩位 1:中断关闭 0:中断使能
6	htfmask6	半满中断 6 遮罩位 1:中断关闭 0:中断使能
5	htfmask5	半满中断 5 遮罩位 1:中断关闭 0:中断使能
4	htfmask4	半满中断 4 遮罩位 1:中断关闭

位/位域	名称	描述
		0:中断使能
3	htfmask3	半满中断 3 遮罩位 1:中断关闭 0:中断使能
2	htfmask2	半满中断 2 遮罩位 1:中断关闭 0:中断使能
1	htfmask1	半满中断 1 遮罩位 1:中断关闭 0:中断使能
0	htfmask0	半满中断 0 遮罩位 1:中断关闭 0:中断使能

27.6.20 DMA 半满中断状态寄存器（DMA_HTFINT_STA）

地址偏移：0x13C

复位值：0x00000000



位/位域	名称	描述
31:16	保留	必须保持复位值
15	htfsta15	半满中断 15 状态位 1:中断触发 0:中断未触发
14	htfsta14	半满中断 14 状态位 1:中断触发 0:中断未触发

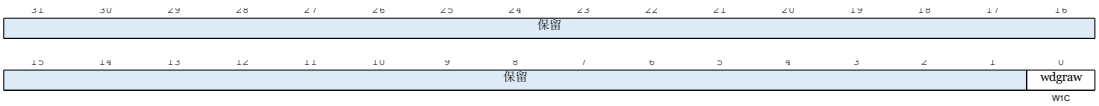
位/位域	名称	描述
13	htfsta13	半满中断 13 状态位 1:中断触发 0:中断未触发
12	htfsta12	半满中断 12 状态位 1:中断触发 0:中断未触发
11	htfsta11	半满中断 11 状态位 1:中断触发 0:中断未触发
10	htfsta10	半满中断 10 状态位 1:中断触发 0:中断未触发
9	htfsta9	半满中断 9 状态位 1:中断触发 0:中断未触发
8	htfsta8	半满中断 8 状态位 1:中断触发 0:中断未触发
7	htfsta7	半满中断 7 状态位 1:中断触发 0:中断未触发
6	htfsta6	半满中断 6 状态位 1:中断触发 0:中断未触发
5	htfsta5	半满中断 5 状态位 1:中断触发

位/位域	名称	描述
		0:中断未触发
4	htfsta4	半满中断 4 状态位 1:中断触发 0:中断未触发
3	htfsta3	半满中断 3 状态位 1:中断触发 0:中断未触发
2	htfsta2	半满中断 2 状态位 1:中断触发 0:中断未触发
1	htfsta1	半满中断 1 状态位 1:中断触发 0:中断未触发
0	htfsta0	半满中断 0 状态位 1:中断触发 0:中断未触发

27.6.21 DMA 看门狗中断原始数据寄存器（DMA_WDGINT_RAW）

地址偏移：0x140

复位值：0x00000000

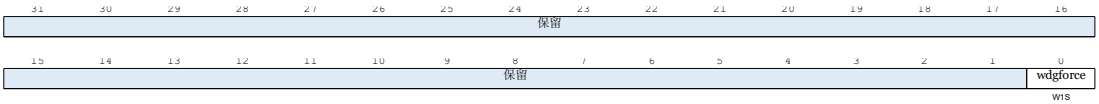


位/位域	名称	描述
31:1	保留	必须保持复位值
0	wdgraw	看门狗中断记录原始位，看门狗超时时置 1，写 1 清 0

27.6.22 DMA 看门狗中断强制赋值寄存器（DMA_WDGINT_FORCE）

地址偏移：0x144

复位值：0x00000000

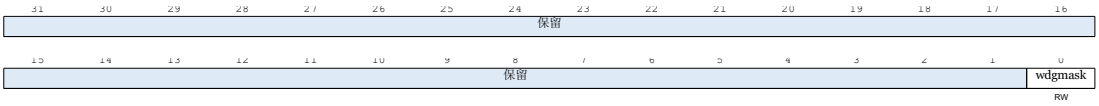


位/位域	名称	描述
31:1	保留	必须保持复位值
0	wdgforce	看门狗强制赋值位，用于测试通道连通性，写 1 将 wdgraw 赋 1

27.6.23 DMA 看门狗中断遮罩位寄存器（DMA_WDGINT_MASK）

地址偏移：0x148

复位值：0x00000001



位/位域	名称	描述
31:1	保留	必须保持复位值
0	wdgmask	看门狗中断遮罩位 1:中断关闭 0:中断使能

27.6.24 DMA 看门狗中断状态寄存器（DMA_WDGINT_STA）

地址偏移：0x14C

复位值：0x00000000

位/位域	名称	描述
31:1	保留	必须保持复位值

位/位域	名称	描述
0	wdgsta	看门狗中断状态位 1:中断触发 0:中断未触发

28 实时时钟（RTC）

28.1 简介

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。RTC 还包含具有中断功能的周期性可编程唤醒标志。两个 32 位寄存器包含 BCD 格式的十分之一秒、秒、分钟、小时、星期几、日期、月份和年份。此外,还可提供二进制格式的亚秒值。系统可以自动将月份的天数补偿为 28、29(闰年)、30 和 31 天。

28.2 主要特征

- 包含十分之一秒、秒、分钟、小时、星期几、日期、月份和年份的日历；
- 具有中断功能的可编程闹钟。可通过任意日历字段的组合触发闹钟；
- 自动唤醒单元,可周期性地生成标志以触发自动唤醒中断。

28.3 模块设计

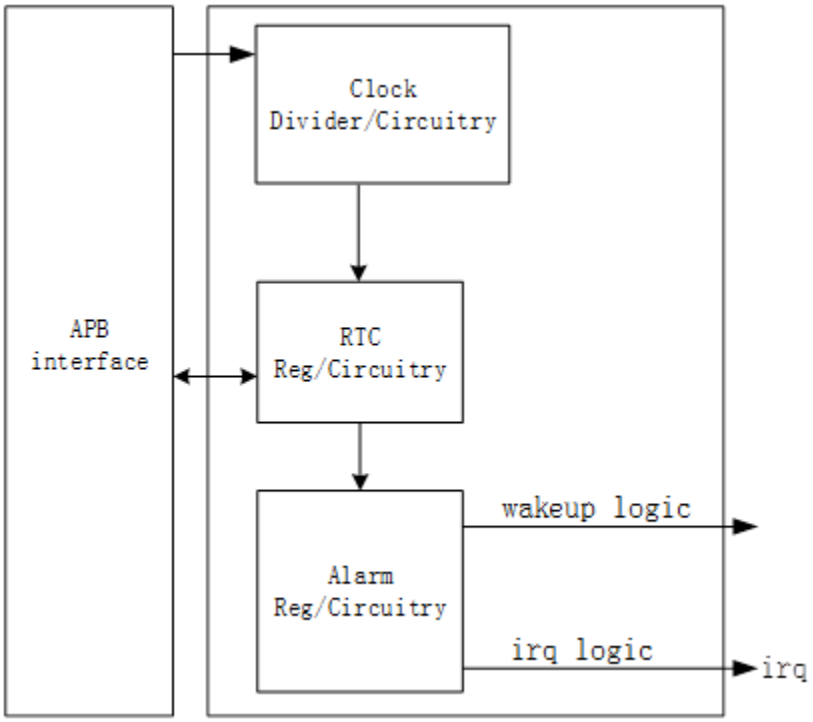


图 28.1 RTC 设计示意图

28.4 应用说明

28.4.1 硬件复位

复位输入异步复位 RTC 核心中的所有寄存器和计数器。中断请求信号被屏蔽，RTC 正常操作停止。

28.4.2 初始化/改变时间和日期

当 RTC 启用时，在第一次写入任何时间/日期计数器后，用户有十分之一秒的时间来初始化/更改所有时间/日期计数器。这保证了十分之一秒的计数器翻转不会发生，因此可能会翻转其他计数器。

28.4.3 时间和日期计数器操作

时间和日期计数器计数时间和日期在 BCD 格式。时间计数器使用 24 小时格式和日期计数器支持闰年计算。 对于正常操作，时间和日期计数器必须通过设置位 RRTC_CTRL[EN]来启用，当配置完所需寄存器后，立刻置位 RRTC_CTRL[EN]，RTC 正常运行。如果 RRTC_CTRL[BTOS]位被清除，它们的操作精度为十分之一秒。

实时时钟被分割因子 RRTC_CTRL[DIV]。当该值为 0 时，表示不分割实时时钟频率。

28.4.4 CPU 唤醒功能

如果设置了 RRTC_CTRL_2 的 WAKEUP_EN 位，到达设置的唤醒时间时，会立刻唤醒 CPU。

28.4.5 告警

闹钟可以设置为世纪、年、月、日、星期、小时、分钟、秒或十分之一秒的任意组合。要设置警报，将所需的值写入相应的警报寄存器，并使能对警报寄存器中所需的比特组进行比较。 当时间和日期计数器匹配所有使能的告警组位时，设置 RRTC_CTRL[ALRM]位。软件必须稍后清除这个位，以便能够检测到未来的警报。

28.4.6 告警中断

如果设置了 RRTC_CTRL[INTE]位，则告警触发后会产生告警中断。软件应该清除 RRTC_CTRL[ALRM]位，以清除挂起的告警中断。

Warning

软件清除 RRTC_CTRL[ALRM]位前，必须清除告警的十分一秒、秒、分钟、小时等告警使能位。

28.4.7 软件配置

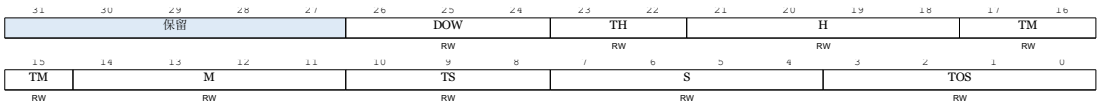
28.5 寄存器

28.5.1 RTC 当前时间寄存器（RRTC_TIME）

地址偏移：0x00

复位值：0x00000000

该寄存器只支持按字（32 位）访问



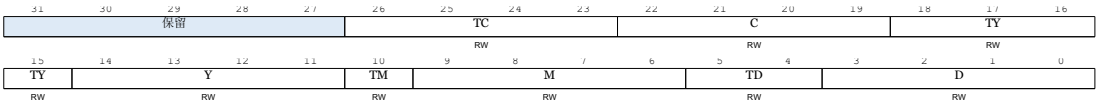
位/位域	名称	描述
31:27	保留	必须保持复位值
26:24	DOW	星期数，有效值从 1(星期日)到 7(星期六)
23:22	TH	小时数十位，24 小时模式下有效值 0 ~ 2；12 小时模式下有效值 0 ~ 1
21:18	H	小时数个位，有效值 0 ~ 9
17:15	TM	分钟数十位，有效值 0 ~ 5
14:11	M	分钟数个位，有效值 0 ~ 9
10:8	TS	秒数十位，有效值 0 ~ 5
7:4	S	秒数个位，有效值 0 ~ 9
3:0	TOS	十分之一秒，有效值 0 ~ 9

28.5.2 RTC 当前日期寄存器（RRTC_DATE）

地址偏移：0x04

复位值：0x00000000

该寄存器只支持按字（32 位）访问



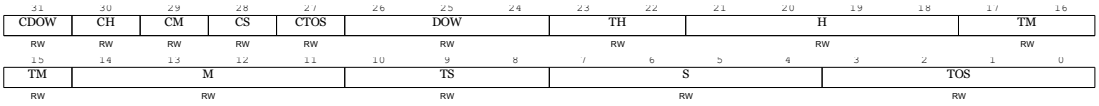
位/位域	名称	描述
31:27	保留	必须保持复位值
26:23	TC	世纪数十位，有效值 0 ~ 9
22:19	C	世纪数个位，有效值 0 ~ 9
28:15	TY	年数十位，有效值 0 ~ 9
14:11	Y	年数个位，有效值 0 ~ 9
10	TM	月数十位，有效值 0 或 1
9:6	M	月数个位，有效值 0 ~ 9
5:4	TD	天数十位，有效值 0 ~ 3
3:0	D	天数个位，有效值 0 ~ 9

28.5.3 RTC 闹钟时间寄存器（RRTC_TALRM）

地址偏移：0x08

复位值：0x00000000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31	CDOW	星期位比较使能位 0:关闭星期位比较 1:开启星期位比较
30	CH	小时位比较使能位

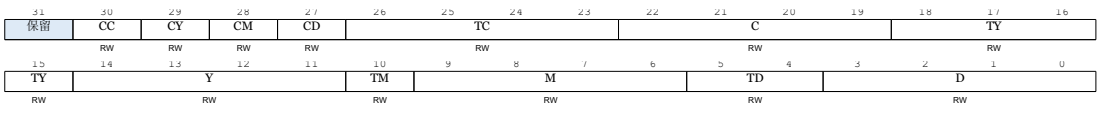
位/位域	名称	描述
		0:关闭小时位比较 1:开启小时位比较
29	CM	分钟位比较使能位 0:关闭分钟位比较 1:开启分钟位比较
28	CS	秒位比较使能位 0:关闭秒位比较 1:开启秒位比较
27	CTOS	十分之一秒比较使能位 0:关闭十分之一秒比较 1:开启十分之一秒比较
26:24	DOW	星期数，有效值从 1(星期日)到 7(星期六)
23:22	TH	小时数十位，24 小时模式下有效值 0 ~ 2；12 小时模式下有效值 0 ~ 1
21:18	H	小时数个位，有效值 0 ~ 9
17:15	TM	分钟数十位，有效值 0 ~ 5
14:11	M	分钟数个位，有效值 0 ~ 9
10:8	TS	秒数十位，有效值 0 ~ 5
7:4	S	秒数个位，有效值 0 ~ 9
3:0	TOS	十分之一秒，有效值 0 ~ 9

28.5.4 RTC 闹钟日期寄存器（RRTC_DALRM）

地址偏移：0x0C

复位值：0x00000000

该寄存器只支持按字（32 位）访问



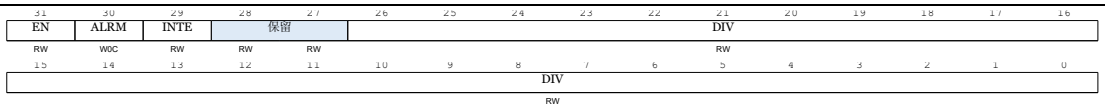
位/位域	名称	描述
31	保留	必须保持复位值
30	CC	世纪数比较使能位 0:关闭世纪比较 1:开启世纪比较
29	CY	年数比较使能位 0:关闭年数比较 1:开启年数比较
28	CM	月数比较使能位 0:关闭月数比较 1:开启月数比较
27	CD	天数比较使能位 0:关闭天数比较 1:开启天数比较
26:23	TC	世纪数十位，有效值 0 ~ 9
22:19	C	世纪数个位，有效值 0 ~ 9
28:15	TY	年数十位，有效值 0 ~ 9
14:11	Y	年数个位，有效值 0 ~ 9
10	TM	月数十位，有效值 0 或 1
9:6	M	月数个位，有效值 0 ~ 9
5:4	TD	天数十位，有效值 0 ~ 3
3:0	D	天数个位，有效值 0 ~ 9

28.5.5 RTC 控制寄存器（RRTC_CTRL）

地址偏移：0x10

复位值：0x00000000

该寄存器只支持按字（32 位）访问



位/位域	名称	描述
31	EN	RTC 运行使能位 0:关闭 RTC，所有配置将不产生效果 1:开启 RTC
30	ALRM	闹钟触发位，由硬件置位，当符合配置的闹钟状态时该位置位，写 0 清除
29	INTE	中断使能位 0:关闭中断 1:开启中断，在 ALRM 置位时触发中断
28:27	保留	必须保持复位值
26:0	DIV	分频器系数，若 BTOS 为 0，建议配为 1638，若 BTOS 为 1，建议配为 16384

28.5.6 RTC 唤醒控制寄存器（RRTC_WKPC）

地址偏移：0x14

复位值：0x00000000

该寄存器只支持按字（32 位）访问

位/位域	名称	描述
31:1	保留	必须保持复位值
0	WKPE	唤醒使能 0:关闭唤醒功能 1:开启唤醒功能，触发闹钟时将唤醒系统，具体配置参考电源管理模块

29 模拟/数字转换控制器（ADC）

29.1 简介

在 AS32A601 中共有 3 个 ADC 模块，其 ADC 采用的是逐次逼近型模拟数字转换器，分辨率为 12bit，拥有最多 16 路外部通道。

该 ADC 支持单次、连续、扫描或间接转换等多种工作模式；支持模拟监控器功能，可以监测输入电压是否超过用户设定的电压范围，并可在超出范围时发送中断。ADC 转换结果可按照左对齐或右对齐的方式存储在 16 位数据寄存器中。

29.2 主要特征

- 控制器特性
  - 3 个独立 ADC 模块
  - 转换序列分为规则组（regular group）和注入组（injection group）
  - 8 种工作模式
  - 通过内部软件触发或外部硬件触发启动 ADC
  - 模拟监控器功能
  - DMA 访问，仅用于规则组通道
- ADC 特性
  - 模拟供电：3.3V
  - 数字供电：1.2V
  - 内置参考电压：2.5V/2.0V/1.5V±0.5%
  - 12 位分辨率
  - 高达 2Msps
  - 采样通道：
    - 16 组模拟通道（支持 8 组差分输入）
    - 内置 1 路温度传感器（精度为±2℃，温度范围：-40~125℃）

- SNDR：
  - 10KHz/1Msps：65 dB
  - 10KHz/2Msps：62 dB
- THD：
  - 10KHz/1Msps：-72 dB
  - 10KHz/2Msps：-68 dB
- INL：
  - 1Msps：±2 LSB
  - 2Msps：±2.2 LSB
- DNL：
  - 1Msps：±1 LSB
  - 2Msps：±1.3 LSB

29.3 功能说明

29.3.1 上电时序

在开始所有功能之前，需配置 ADC_CTRL1 寄存器的 ADON 位为 1，使 ADC 上电，然后等在 tSTART 后，配置 ADC_CTRL1 寄存器的 RESET 位为 1，使 ADC 退出复位状态。

在 EN_BUF 为 0 时，tSTART 不应低于 2us；在 EN_BUF 为 1 时，tSTART 不应低于 8us。

29.3.2 工作模式

根据实际应用可以灵活使用不同的模式，上电和有效触发后 ADC 工作于以下模式之一。

表 29.1 ADC 工作模式配置表

工作模式	MODE_BITS	触发源	转换序列
mode1	5’b0000x	规则触发	规则组单通道单次转换
mode2	5’b0100x	规则触发	规则组单通道连续转换

工作模式	MODE_BITS	触发源	转换序列
mode3 (注入组扫描模式)	5'b10000 (INTERVAL=0)	规则/注入触发	规则组扫描+注入组扫描模式多通道单次转换
mode3 (注入组间隔模式)	5'b10000 (INTERVAL=1)	规则/注入触发	规则组扫描+注入组间隔模式多通道单次转换
mode4	5'b10001	规则触发+自动注入触发	规则组扫描+注入组扫描模式多通道单次转换
mode5 (注入组扫描模式)	5'b11000 (INTERVAL=0)	规则/注入触发	规则组扫描+注入组扫描模式多通道连续转换
mode5 (注入组间隔模式)	5'b11000 (INTERVAL=1)	规则/注入触发	规则组扫描+注入组间隔模式多通道连续转换
mode6	5'b11001	规则触发+自动注入触发	规则组扫描+注入组扫描模式多通道连续转换
mode7	5'b1x10x	规则触发	规则组子组扫描模式转换
mode8	5'b1x01x	注入触发	注入组子组扫描模式转换

Note

注：MODE_BITS={SCAN,CONT,DISCEN,IDISEN,IAUTO}，对应 ADC_CTRL0 寄存器的[4:0]位。

在描述每个模式操作流程之前，有必要介绍一些术语，例如规则组，注入组等。

ADC 的输入通道中，编号 0~15 是外部输入通道，本文中称为 ch0~ch15，其他编号为内部通道，请参考 ADC_RSQRx 寄存器或 ADC_ISQRx 寄存器中 SEQ 的编号定义。

ADC 模块包含一个规则组和一个注入组可配置。规则组和注入组均可配置转换零个或多个 ADC 输入通道。同一组或不同组中的输入通道配置，除了最大数量外，没有额外限制，即任意一个输入通道，如 ch0，可以配置到规则组或注入组，或两个组都配置，甚至一个组中配置多个 ch0 都可以。

规则组和注入组的区别，除了最大通道数，并没有区别，只是在不同模式下能够转换的组，以及转换方式有区别。详细内容可参考本章节后续各个工作模式的描述。

当 ADC 配置为可使用规则组和注入组进行采样的模式时，如果某个组被触发，会根据对应的序列寄存器，按顺序进行转换。如果是规则组，则会按照 ADC_RSQRx（x=0~15）寄存器的顺序，依次转换寄存器中配置的通道。与规则组相应，注入组则是按照 ADC_ISRQx（x=0~3）寄存器的顺序，依次转换寄存器中配置的通道。

例如，如果 RSQ0~RSQ11 分别设置为 9、8、12、1、5、4、7、3、13、2、0、0，且规则组长度为 12，则规则组将按规则组序列图所示顺序依次进行转换。

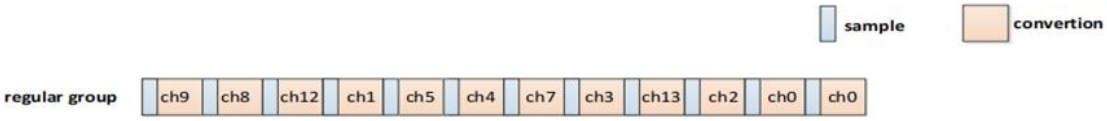


图 29.1 规则组序列图

如果 RSQL 设置为 8（Length=9），则最后 3 个通道将无效且无法转换。因此，有效的规则组序列如有效规则组序列图所示。

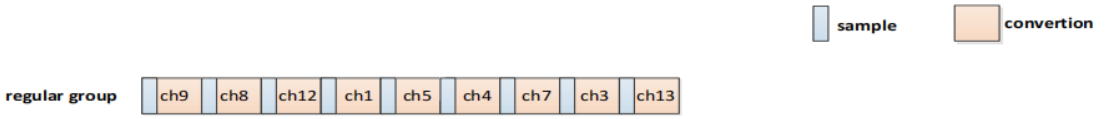


图 29.2 有效规则组序列图

以同样的方式，注入组由最多 4 个通道组成，顺序依次为 ISQ0 至 ISQ3。

例如，如果 ISQ0~ISQ3 分别设置为 12、7、13、2，则将注入组按注入组序列序列图所示进行排列。

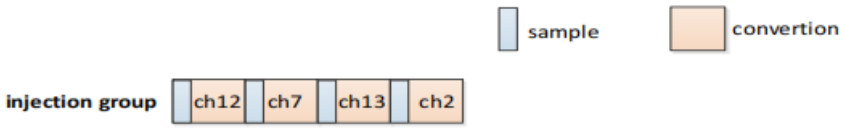


图 29.3 注入组序列序列图

如果 ISQL 设置为 2（Length=3），则最后 1 个通道将无效并且不会被转换。因此，有效注入组序列如有效注入组序列图所示。

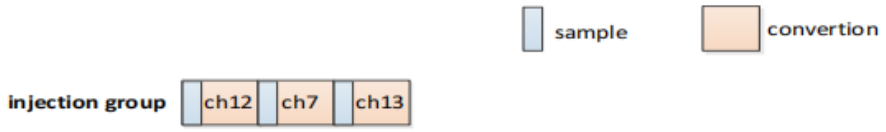


图 29.4 有效注入组序列图

显然，规则组触发和注入组触发是开始转换规则组和注入组序列的相应信号。该触发源自 ADC 内部 ADC_CTRL0 的 SWSTART、ISWSTART 或外部触发源。当 ADC 处于规则组通道转换过程中时，规则触发无效。基于此基本介绍，每种模式的详细描述如本节后续所述。

29.3.2.1 mode1

此模式下，无论 RSQL 为何值，ADC 仅转换规则组的第一个通道。模式按 ADC 工作模式配置表进行配置后，有效触发可使 ADC 工作在此模式。

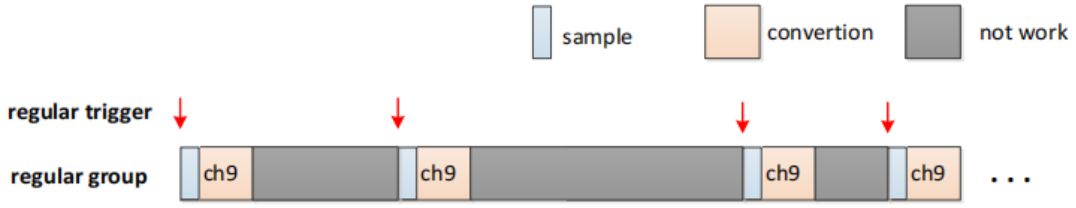


图 29.5 mode1 工作流程图

如 mode1 工作流程图所示，规则组中的第一个通道在有效的规则触发后转换一次。然后 ADC 进入空闲状态，直到下一次有效规则触发带来的下一次转换。

29.3.2.2 mode2

此模式下，无论 RSQL 为何值，只会连续转换规则组中的第一个通道。模式按 ADC 工作模式配置表进行配置后，有效触发可以使 ADC 在此模式下工作。

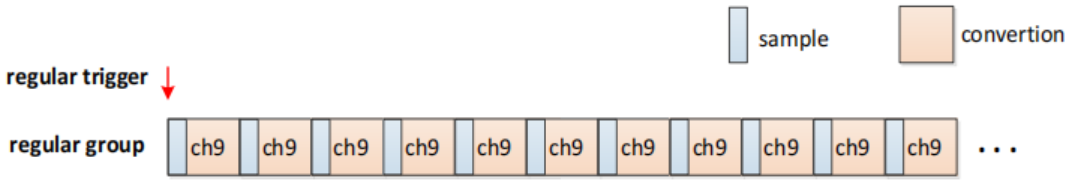


图 29.6 mode2 工作流程图

如 mode2 工作流程图所示，在有效的规则触发后，规则组第一个通道将不断转换，除非，断电、复位或者更改 ADC 工作模式。

29.3.2.3 mode3

29.3.2.3.1 interval = 0，注入组为扫描模式

此模式转换规则组通道和注入组通道。有效的规则和注入组通道长度分别由 ADC_SQL 中的 RSQL 和 ISQL 决定。使用 ADC 工作模式配置表中的模式配置，有效触发可使 ADC 在此模式下工作。例如，RSQL 设置为 6（Length=7），ISQL 设置为 2（Length=3），一个典型操作如 mode3 注入组扫描模式工作流程图所示。

第一次规则触发会触发 ADC 转换规则组中的 7 个通道。当 ADC 正在转换规则组中的 ch1 时，此时产生注入触发，会不等待 ch1 的转换完成，而立刻切换至转换 3 个注入组通道，在所有注入组通道转换完成后，自动切换回规则组通道，重新开始 ch1 的转换，完成全部的规则通道转换后，ADC 将运行至空闲状态，直至下一次触发到来。

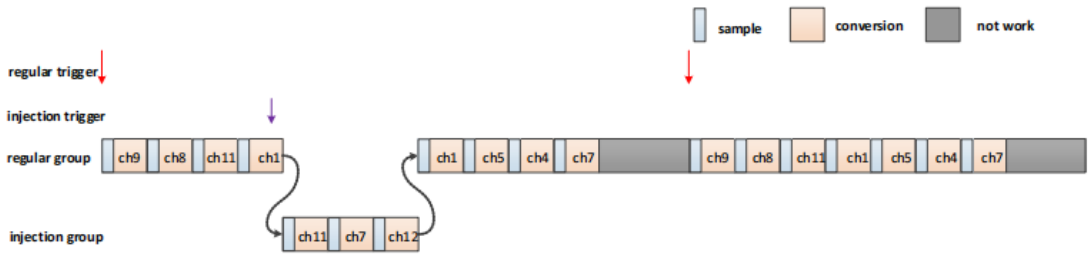


图 29.7 mode3 注入组扫描模式工作流程图

如果在 ADC 空闲时发生注入触发，ADC 将完成有效注入组通道的转换，如 mode3 在 ADC 空闲状态下具有注入触发的工作流程图所示。

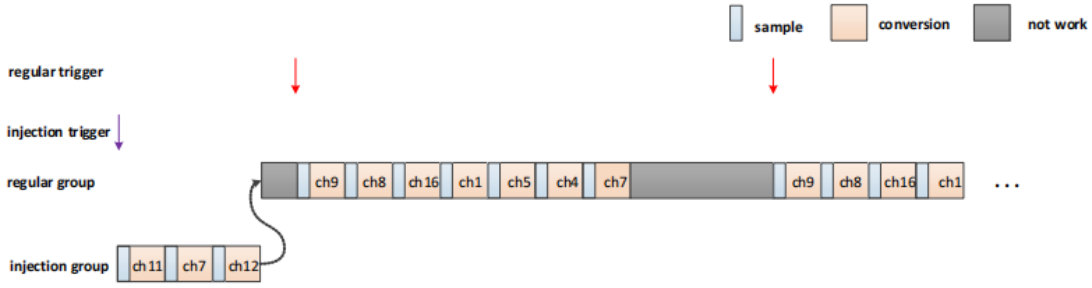


图 29.8 mode3 在 ADC 空闲状态下具有注入触发的工作流程图

29.3.2.3.2 interval = 1，注入组为间隔模式

与 mode3 注入组扫描模式工作流程图的区别在于，产生一次注入触发只会转换注入组序列的一个通道，下一次再发生注入触发，注入组序列的下一通道进行转换，如 mode3 注入组间隔模式工作流程图所示。

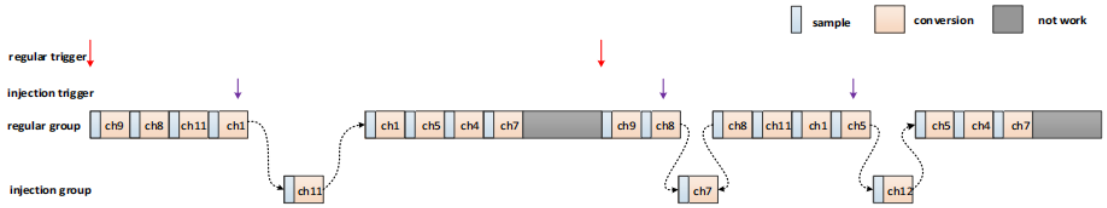


图 29.9 mode3 注入组间隔模式工作流程图

如果该模式下在 ADC 空闲时发生注入触发，ADC 将完成一个有效注入组通道的转换，如 mode3 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程图所示：

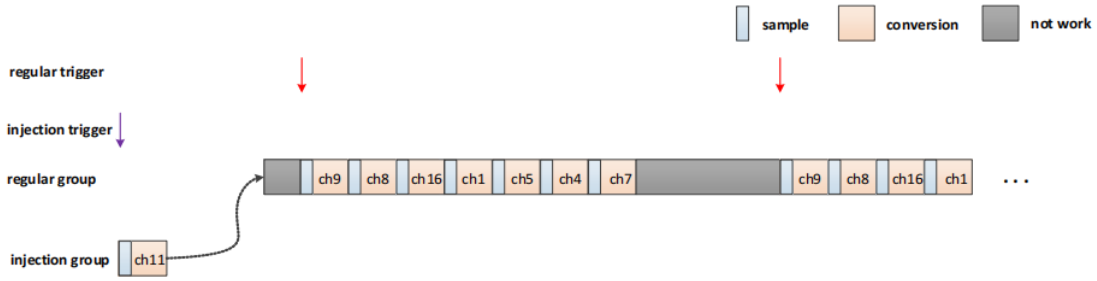


图 29.10 mode3 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程图

29.3.2.4 mode4

此模式下收到规则触发后将自动按照先转换规则组通道，后转换注入组通道的顺序进行转换。有效的规则组通道和注入组通道分别由 RSQL 和 ISQL 决定。使用 ADC 工作模式配置表中的模式配置，有效触发可使 ADC 在此模式下工作。例如，RSQL 设置为 6（Length=7），ISQL 设置为 2（Length=3）。典型操作如 mode4 工作流程图所示。规则触发器开始转换前 7 个规则组通道，然后自动开始转换 3 个注入组通道。在总共 10 个通道均完成完全转换后，ADC 将运行至空闲状态，直到下一个有效的规则触发。

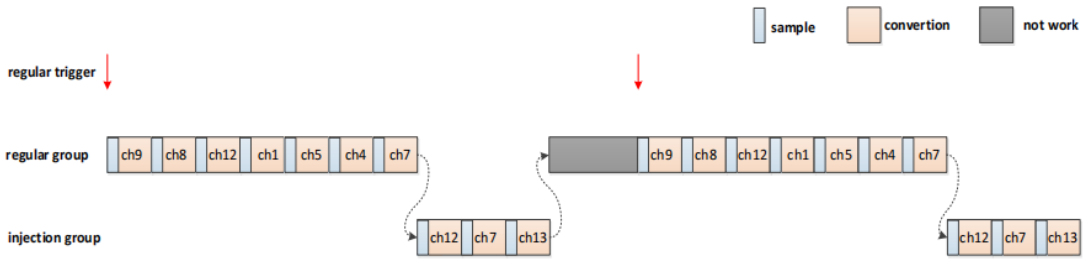


图 29.11 mode4 工作流程图

29.3.2.5 mode5

29.3.2.5.1 interval = 0，注入组为扫描模式

与 mode3 区别在于此模式为规则组连续转换。使用 ADC 工作模式配置表中的模式配置，有效触发可使 ADC 在此模式下工作。此模式的一个关键特性是单个规则触发可以使 ADC 始终工作，除了掉电、复位或模式更改。例如，RSQL 设置为 6（Length=7），ISQL 设置为 2（Length=3）。一个典型操作如 mode5 注入组扫描模式工作流程图所示。在规则触发后，ADC 按规则组通道顺序工作，如果发生注入触发，则在注入组通道上工作。

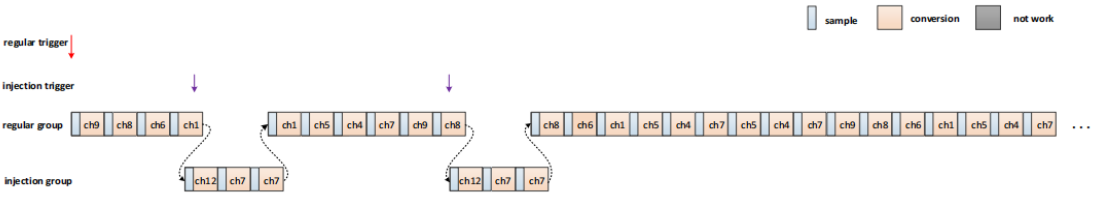


图 29.12 mode5 注入组扫描模式工作流程图

特别地，如果在 ADC 空闲时发生注入触发，ADC 将首先完成有效注入组通道的转换，如 mode5 注入组扫描模式在 ADC 空闲状态下具有注入触发的工作流程图所示。

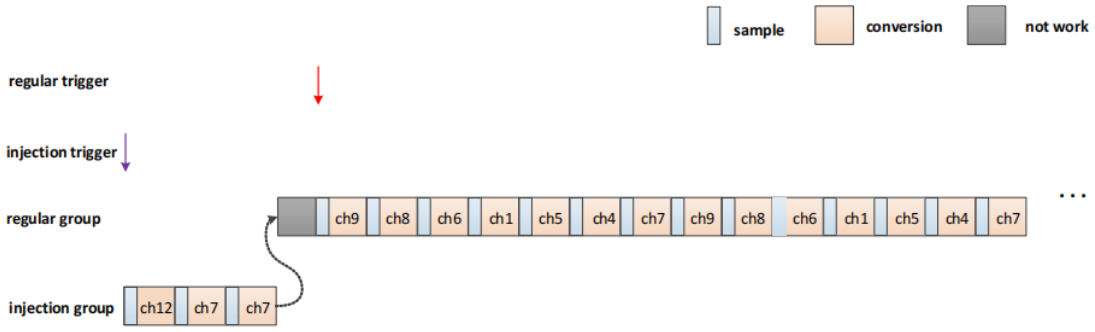


图 29.13 mode5 注入组扫描模式在 ADC 空闲状态下具有注入触发的工作流程图

29.3.2.5.2 interval = 1，注入组为间隔模式

与 mode5 注入组扫描模式工作流程图的区别在于，产生一次注入触发只会转换注入组序列的一个通道，下一次再发生注入触发，注入组序列的下一通道进行转换。工作流程如 mode5 注入组间隔模式工作流程图所示。

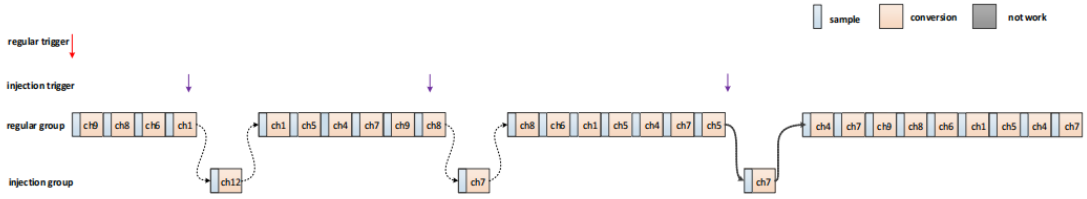


图 29.14 mode5 注入组间隔模式工作流程图

如果该模式下在 ADC 空闲时发生注入触发，ADC 将完成一个有效注入组通道的转换，如 mode5 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程图所示：

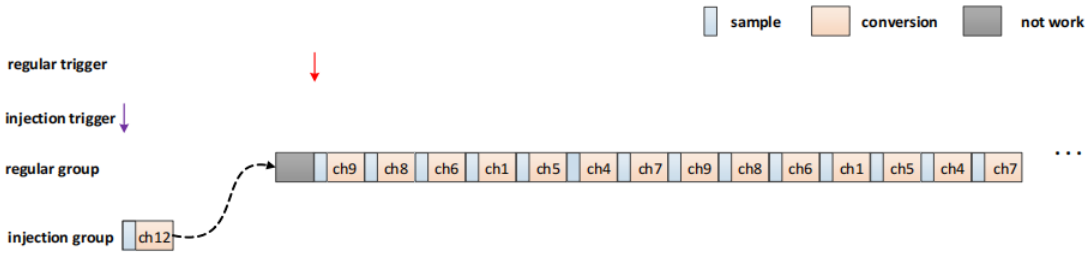


图 29.15 mode5 注入组间隔模式在 ADC 空闲状态下具有注入触发的工作流程图

29.3.2.6 mode6

与 mode4 区别在于此模式为连续转换。使用 ADC 工作模式配置表中的模式配置，有效的规则触发可以使 ADC 在此模式下工作。此模式的一个关键特性是单个规则触发可以使 ADC 始终工作，除了掉电、复位或模式更改。例如，RSQL 设置为 6（Length=7），ISQL 设置为 2（Length=3），操作流程如 mode6 工作流程图所示。ADC 在规则组通道上按顺序工作，然后在规则组转换完成后转换注入组通道。

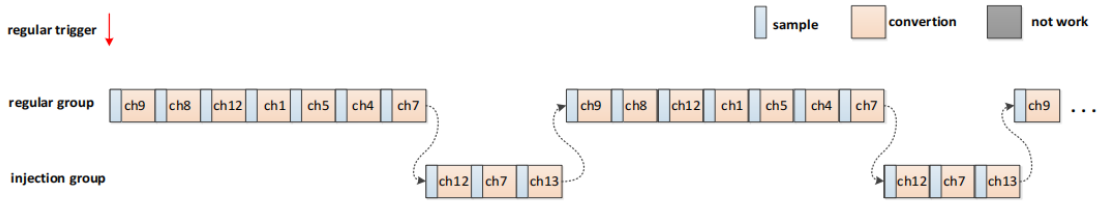


图 29.16 mode6 工作流程图

29.3.2.7 mode7

此模式仅转换规则组通道。有效的规则组通道由 RSQL 决定。使用 ADC 工作模式配置表中的模式配置，ADC 可以在此模式下工作。依据 ADC_CTRL0[DISCNUM]将有效的规则通道分成若干子组。

例如，RSQL 设置为 6（Length=7），DISCNUM 设置为 1（Length=2）。

第一次规则触发：ch9、ch8；

第二次规则触发：ch12、ch1；

第三次规则触发：ch5、ch4；

第四次规则触发：ch7；

因此，实际的转换流程如 mode7 工作流程图所示。

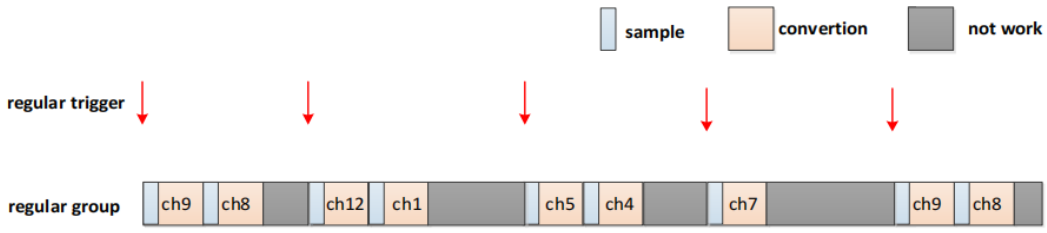


图 29.17 mode7 工作流程图

29.3.2.8 mode8

该模式仅转换注入组通道。有效的注入通道组通道由 ISQL 决定。使用 ADC 工作模式配置表中的模式配置，ADC 可以在此模式下工作。每次触发只转换一个通道。例如，ISQL 设置为 2（Length=3）。

第一次注入触发：ch12；

第二次注入触发：ch7；

第三次注入触发：ch13；

第四次注入触发：ch12；

…

因此，实际的转换流程如 mode8 工作流程图所示。

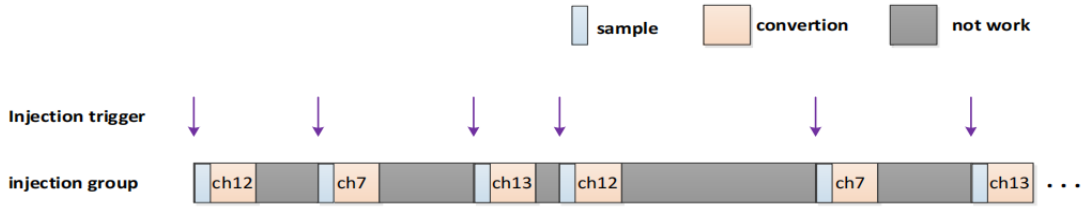


图 29.18 mode8 工作流程图

29.3.3 触发说明

29.3.3.1 触发方式

表 29.2 不同触发方式下的响应行为

触发方式	响应行为
触发规则组	规则组转换
触发注入组	注入组转换
在规则组转换期间产生规则触发	规则组持续转换，第二次触发事件不响应
在规则组转换期间产生注入触发	不等待当前转换通道完成转换，在小于一个转换周期的时间内立刻切换到注入组，注入组转换完成后切换到原来的规则组中被打断的通道继续执行（规则组序列未转换完的情况下）
在注入组转换期间产生规则触发	注入组持续转换，第二次注入触发事件不响应
在注入组转换期间产生注入触发	在注入转换期间产生一次规则事件，注入转换不会被打断，但是规则序列将在注入序列结束后被执行
规则触发和注入触发同一时刻产生	注入组先转换，完成后在转换规则组。 在规则组正在进行转换时，此时的行为可认为是收到一次注入触发和一次冗余规则触发。 在注入组正在进行转换时，此时的行为可认为是收到一次冗余的注入触发和一次规则触发。

29.3.3.2 触发源

ADC 外部触发源详见下表：

表 29.3 外部触发源说明

	iexttrig			exttrig		
	0	1	2	0	1	2
ADC0/1/2	TIM0.TRGO2	TIM0.CHx(0)	TIM0.CHx(1)	TIM0.CHx(2)	TIM0.CHx(3)	TIM0.TRGO2

该触发信号经由内部硬件连接，由 TIM 输出至 ADC。其中，TIM0.TRGO2 为 TIM0 输出的 ADC 同步信号，该信号配置说明详见 TIM0 的 TIMx_CR2 寄存器的 MMS2 位域的说明。其中，TIM0.CHx(0~3)为 TIM0 的通道 0~3 的输出信号。

ADC 内部将对触发信号的上升沿进行检测，检测到上升沿后则完成一个触发，按照 ADC 工作模式完成采样。

29.3.4 采样通道

ADC 采样通道由 ADC_RSQRx 寄存器和 ADC_ISQRx 寄存器中的 SEQ 位进行配置，由 ADC_CONFIG 寄存器的 SELVI_HD_LS 位选择单端/差分模式。ADC 实际采样通道如下表所示。

表 29.4 ADC 采样通道说明

ADC_CONFIG.SELVI_HD_LS	SEQ	ADC 采样通道	
		Positive Input	Negative Input
x	1x000	VTEMP (Temperature sensor)	VREFN
x	1x001	VBAT(VSS)	VREFN
x	1x010	VR_EX1(VSS)	VREFN
x	1x011	VR_EX2(VSS)	VREFN
x	1x100	VR_EX3(VSS)	VREFN
0	00000	VIN1	VREFN
0	00001	VIN2	VREFN
0	00010	VIN3	VREFN

ADC_CONFIG.SELVI_HD_LS	SEQ	ADC 采样通道	
		Positive Input	Negative Input
0	00011	VIN4	VREFN
0	00100	VIN5	VREFN
0	00101	VIN6	VREFN
0	00110	VIN7	VREFN
0	00111	VIN8	VREFN
0	01000	VIN9	VREFN
0	01001	VIN10	VREFN
0	01010	VIN11	VREFN
0	01011	VIN12	VREFN
0	01100	VIN13	VREFN
0	01101	VIN14	VREFN
0	01110	VIN15	VREFN
0	01111	VIN16	VREFN
1	0x000	VIN1	VIN9
1	0x001	VIN2	VIN10
1	0x010	VIN3	VIN11
1	0x011	VIN4	VIN12
1	0x100	VIN5	VIN13
1	0x101	VIN6	VIN14
1	0x110	VIN7	VIN15
1	0x111	VIN8	VIN16

29.3.5 模拟监控器

模拟监控器用于监控通道的电压是否超出用户设定的阈值范围。模拟监控器需要监控的通道可通过 ADC_AMOCR 的 AMOEN、IAMOEN、AMOSGL 和 AMOCH 位来配置，其中：

AMOEN 为 1 表示模拟监控器会对规则组通道进行监控，为 0 则不监控规则组通道；

IAMOEN 为 1 表示模拟监控器会对注入组通道进行监控，为 0 则不监控注入组通道；

AMOSGL 为 1 表示模拟监控器会对指定的通道进行监控，此时通过 AMOCH 来指定监控的通道，为 0 则不对单个通道进行监控。

由于模拟监控器需要指定被监控的通道，如果当前 ADC 未对监控通道进行转换，则模拟监控器无法发挥作用。关于 ADC 工作模式及模拟监控器监控配置的关系可参考。

表 29.5 模拟监控通道配置

模拟监控通道	AMO_MODE	工作模式	注释
无	3’b00x	所有模式	
所有注入组通道	3’b010	mode3/4/5/6/8	
所有规则组通道	3’b100	除了 mode8	
所有通道	3’b110	所有模式	
单注入组通道	3’b011	mode3/4/5/6/8	转换序列必须包括由 AMOCH 指定的注入通道
单规则组通道	3’b101	mode1~7	转换序列必须包括由 AMOCH 指定的规则通道
单规则组或注入组通道	3’b111	所有通道	转换序列必须包括由 AMOCH 指定的规则或注入通道

注：AMO_MODE={AMOEN，IAMOEN，AMOSGL}，对应 ADC_AMOCR[7:5]。

阈值通过 ADC_AMOHR 和 ADC_AMOLR 寄存器进行配置。阈值比较使用的是原始数据进行比较，设置数据对齐或注入组偏置不影响比较结果。

如果被监控通道当前的转换结果大于阈值 ADC_AMOHR 或小于 ADC_AMOLR，则模拟监控器将 AMO 标志设置为 1，如果 AMOIE 中断使能被配置为 1，则产生中断。

29.3.6 数据符号转换与对齐方式

ADC 可通过配置 ADC_CTRL0 寄存器的 SELDO 位对 ADC 输出数据类型进行选择，输出有符号数或无符号数，并通过内部逻辑将有符号数的补码类型转换为原码类型存入数据寄存器中。

数据对齐功能主要用于控制 ADC 转换结果在数据寄存器中的排布方式，可选择左对齐或右对齐的方式。ADC 分辨率不同，数据存储位置也会存在差异。另外，注入组具有数据偏置功能，因此注入组数据寄存器的符号位会受其影响。

	规则组数据															
SELDO=0	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
SELDO=1	0	0	0	0	Sign	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	注入组数据															
	Sign	Sign	Sign	Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	ALIGN=0															
	规则组数据															
SELDO=0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
SELDO=1	Sign	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
	注入组数据															
	Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0
	ALIGN=1															

图 29.19 ADC 数据排布（分辨率：12bit）

	规则组数据															
SELDO=0	0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
SELDO=1	0	0	0	0	0	0	Sign	D8	D7	D6	D5	D4	D3	D2	D1	D0
	注入组数据															
	Sign	Sign	Sign	Sign	Sign	Sign	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	ALIGN=0															
	规则组数据															
SELDO=0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
SELDO=1	Sign	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
	注入组数据															
	Sign	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0
	ALIGN=1															

图 29.20 ADC 数据排布（分辨率：10bit）

	规则组数据															
SELDO=0	0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
SELDO=1	0	0	0	0	0	0	0	0	Sign	D6	D5	D4	D3	D2	D1	D0
	注入组数据															
	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	D7	D6	D5	D4	D3	D2	D1	D0
	ALIGN=0															
	规则组数据															
SELDO=0	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0
SELDO=1	Sign	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0
	注入组数据															
	Sign	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0
	ALIGN=1															

图 29.21 ADC 数据排布（分辨率：8bit）

	规则组数据															
SELDO=0	0	0	0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0
SELDO=1	0	0	0	0	0	0	0	0	0	0	Sign	D4	D3	D2	D1	D0
	注入组数据															
	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	D5	D4	D3	D2	D1	D0
	ALIGN=0															
	规则组数据															
SELDO=0	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0	0	0
SELDO=1	Sign	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0	0	0
	注入组数据															
	Sign	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0	0
	ALIGN=1															

图 29.22 ADC 数据排布（分辨率：6bit）

29.3.7 采样转换时间

ADC 使用若干个 ADCCLK 周期对输入电压采样，即对 ADC 内部电路进行充电，使其达到外部输入信号的电平，完成采样之后才能进行模拟到数字的转换。采样周期个数可通过 ADC_SPTx(x=0~2)寄存器中的 SPT[3:0]位配置。每个通道可以分别用不同的时间进行采样。

表 29.6 采样转换时间表

SPT[3:0]	采样时间 TSAMP（单位：1/FADC_CLK）
0000	3
0001	5
0010	7
0011	10
0100	13
0101	16
0110	20
0111	30
1000	60
1001	80
1010	100

SPT[3:0]	采样时间 TSAMP（单位：1/FADC_CLK）
1011	120
1100	160
1101	320
1110	480
1111	640

29.3.8 温度传感器

温度传感器可以用来测量器件周围的温度，与 ADC 内部直接连接，通过 ADC 把传感器输出电压转换成数字量。

温度计算公式：TJ = (VVTEMP – VOS)/KT

TJ：Junction Temperature

VVTEMP：当前温度电压数值

VOS：0℃时的电压数值

KT：温度传感器的平均斜率（单位为 mV/℃）

29.3.9 中断

ADC 有两种转换状态标志位：EOC 和 AMO。这两种状态标志均有单独的中断使能配置位，当使能相应的中断请求使能时，在标志置起的同时，会触发系统中断。

ADC 的每个规则组和注入组通道均有单独的 EOC 中断标志。EOC 标志表示规则组或注入组中某个通道的转换结束，产生的 EOC 状态位会在当前完成转换的序列寄存器中置起。

AMO 标志表示是否发生模拟监控器事件。

29.3.10 DMA

当进行规则组的转换时，可通过 DMA 将 ADC 数据保存到内存中。需要注意，在配置 DMA 时，需根据当前 ADC 的配置，设置 DMA 的源地址和结束地址，比如当前使用 mode3、RSQL 设置为 3（Length=4），则 DMA 的源地址为 ADC_RDR0，结束地址为 ADC_RDR4 的地址。

注：在使用 DMA 功能时，由于 DMA 读取 ADC_RDR 的速度较快，EOC 状态和中断请求会被迅速清除，可能导致软件和中断控制器无法正常读到 EOC 状态和检测到中断。

29.4 应用说明

29.4.1 上电配置说明

在 ADC 工作前，需要完成 ADC 上电配置，详见上电时序章节。配置流程如下：

- 配置 ADC 控制寄存器 1 的 ADON 位，使 ADC 上电；
- 延时；
- 配置 ADC 控制寄存器 1 的 PSC 位，根据总线频率配置 ADC 时钟分频系数；
- 配置 ADC 控制寄存器 1 的 RESET 位，使 ADC 解除复位。

29.4.2 ADC 触发配置说明

ADC 控制器的规则组和注入组都具备两种触发方式：

1. 内部软件触发；
2. 外部硬件触发。

通过配置规则组和注入组触发源选择寄存器可以完成对触发方式的选择，完成配置后，配置规则组和注入组触发使能寄存器，使能对应的触发。

若选择软件触发，则需要向规则组和注入组的软件触发寄存器写 1，触发规则组和注入组转换，该位由硬件清零。

若选择硬件触发，则硬件触发信号由定时器提供。

29.4.3 ADC 基础配置说明

- 依据工作模式章节的说明，配置对应寄存器；
- 配置规则组和/或注入组转换序列；
- 配置各个通道对应的采样时间；
- 配置规则组和/或注入组的转换长度；

- 规则组和/或注入组触发 ADC 工作；
- 规则组和/或注入组转换完成后，EOC 标志位硬件置 1，若使能对应的中断使能，则产生相关中断；
- 读取对应序列的数据寄存器。

29.5 寄存器

ADC0 控制器基地址：0x4100_1000

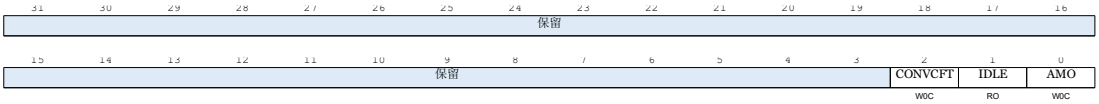
ADC1 控制器基地址：0x4100_2000

ADC2 控制器基地址：0x4200_1000

29.5.1 ADC 状态寄存器（ADC_STATE）

地址偏移：0x00

复位值：0x0000_0000

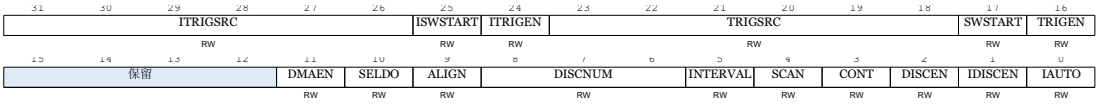


位/位域	名称	描述
31:3	保留	必须保持复位值
2	CONVCFT	触发冲突标志 0：没有触发冲突 1：发生触发冲突，写 0 清除
1	IDLE	ADC 空闲状态标志 0：ADC 处于非空闲状态 1：ADC 处于空闲状态
0	AMO	模拟监控异常事件发生标志 0：没有异常事件 1：发生异常事件，写 0 清除

29.5.2 ADC 控制寄存器 0（ADC_CTRL0）

地址偏移：0x04

复位值：0x0000_0000



位/位域	名称	描述
31:26	ITRIGSRC	注入组触发源选择 000000：内部（软件触发） 000001：外部注入组触发源 0 000010：外部注入组触发源 1 000011：外部注入组触发源 2 其他：保留
25	ISWSTART	注入组软件触发 写 1 触发，读为 0
24	ITRIGEN	注入组触发使能 0：禁用 1：使能
23:18	TRIGSRC	规则组触发源选择 000000：内部（软件触发） 000001：外部规则组触发源 0 000010：外部规则组触发源 1 000011：外部规则组触发源 2 其他：保留
17	SWSTART	规则组软件触发

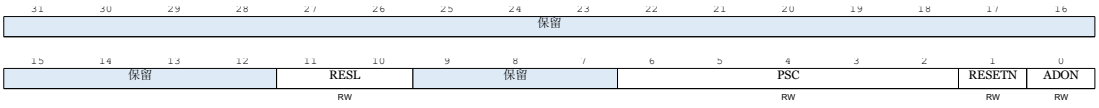
位/位域	名称	描述
		写 1 触发，读为 0
16	TRIGEN	规则组触发使能 0：禁用 1：使能
15:12	保留	必须保持复位值
11	DMAEN	DMA 功能使能 0：禁用 DMA 1：使能 DMA
10	SELDO	ADC 输出数据类型选择 0：无符号数 1：有符号数
9	ALIGN	数据对齐 0：右对齐 1：左对齐
8:6	DISCNUM	通道的不连续转换长度 0~7：mode 7 规则组子组长度为 1~8，即 DISCNUM+1
5	INTERVAL	间隔模式 0：注入组为扫描模式 1：注入组为间隔模式
4	SCAN	ADC 工作模式-bit4

位/位域	名称	描述
3	CONT	ADC 工作模式-bit3
2	DISCEN	ADC 工作模式-bit2
1	IDISCEN	ADC 工作模式-bit1
0	IAUTO	ADC 工作模式-bit0

29.5.3 ADC 控制寄存器 1（ADC_CTRL1）

地址偏移：0x08

复位值：0x0000_007c



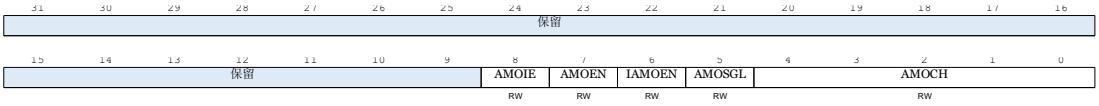
位/位域	名称	描述
31:12	保留	必须保持复位值
11:10	RESL	分辨率选择 00：12bit 01：10bit 10：8bit 11：6bit
9:7	保留	必须保持复位值
6:2	PSC	分频系数 00000：不分频 00001：2 分频 00010：4 分频 … 11111：62 分频
1	RESETN	ADC 复位

位/位域	名称	描述
		0：ADC 复位 1：ADC 不复位
0	ADON	ADC 上电 0：ADC 断电 1：ADC 上电

29.5.4 ADC 模拟监控器配置寄存器（ADC_AMOCR）

地址偏移：0x0C

复位值：0x0000_0000



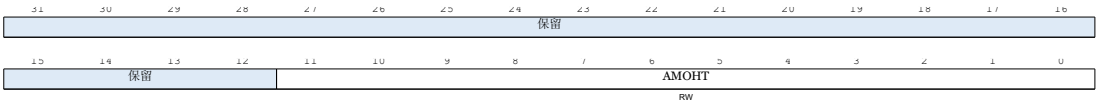
位/位域	名称	描述
31:9	保留	必须保持复位值
8	AMOIE	模拟监控中断使能 0：禁用 1：使能
7	AMOEN	模拟监控模式-bit2
6	IAMOEN	模拟监控模式-bit1
5	AMOSGL	模拟监控模式-bit0
4:0	AMOCH	模拟监控通道 当模拟监控器配置为仅检测单个通道时，指定被监测的通道。

位/位域	名称	描述
		该寄存器域的通道编码与 ADC_RSQRx 中 SEQ 域的通道编码相同。

29.5.5 ADC 高阈值寄存器（ADC_AMOHR）

地址偏移：0x10

复位值：0x0000_0000

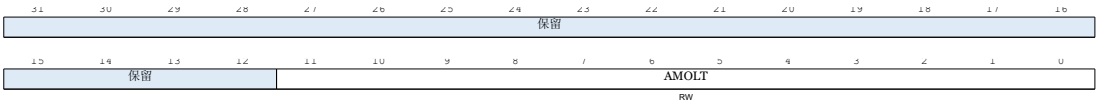


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	AMOHT	模拟控制器的高阈值 定义高阈值

29.5.6 ADC 低阈值寄存器（ADC_AMOLR）

地址偏移：0x14

复位值：0x0000_0000

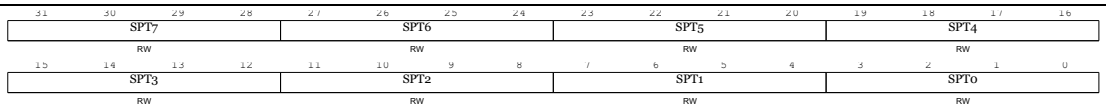


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	AMOLT	模拟控制器的低阈值 定义低阈值

29.5.7 ADC 采样时间寄存器 0（ADC_SPT0）

地址偏移：0x18

复位值：0x0000_0000



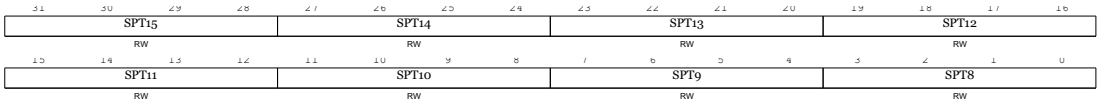
位/位域	名称	描述
31:28	SPT7	通道 7 采样时间 参考 SPT0
27:24	SPT6	通道 6 采样时间 参考 SPT0
23:20	SPT5	通道 5 采样时间 参考 SPT0
19:16	SPT4	通道 4 采样时间 参考 SPT0
15:12	SPT3	通道 3 采样时间 参考 SPT0
11:8	SPT2	通道 2 采样时间 参考 SPT0
7:4	SPT1	通道 1 采样时间 参考 SPT0
3:0	SPT0	通道 0 采样时间 0000：3 ADCCLK 0001：5 ADCCLK

位/位域	名称	描述
		0010：7 ADCCLK 0011：10 ADCCLK 0100：13 ADCCLK 0101：16 ADCCLK 0110：20 ADCCLK 0111：30 ADCCLK 1000：60 ADCCLK 1001：80 ADCCLK 1010：100 ADCCLK 1011：120 ADCCLK 1100：160 ADCCLK 1101：320 ADCCLK 1110：480 ADCCLK 1111：640 ADCCLK

29.5.8 ADC 采样时间寄存器 1（ADC_SPT1）

地址偏移：0x1C

复位值：0x0000_0000



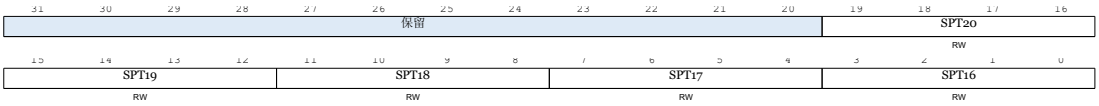
位/位域	名称	描述
31:28	SPT15	通道 15 采样时间 参考 SPT0
27:24	SPT14	通道 14 采样时间 参考 SPT0

位/位域	名称	描述
23:20	SPT13	通道 13 采样时间 参考 SPT0
19:16	SPT12	通道 12 采样时间 参考 SPT0
15:12	SPT11	通道 11 采样时间 参考 SPT0
11:8	SPT10	通道 10 采样时间 参考 SPT0
7:4	SPT9	通道 9 采样时间 参考 SPT0
3:0	SPT8	通道 8 采样时间 参考 SPT0

29.5.9 ADC 采样时间寄存器 2（ADC_SPT2）

地址偏移：0x20

复位值：0x0000_0000



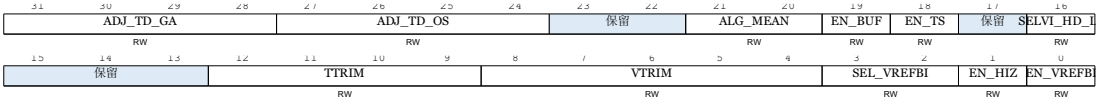
位/位域	名称	描述
31:20	保留	必须保持复位值
19:16	SPT20	通道 20 采样时间

位/位域	名称	描述
		参考 SPT0
15:12	SPT19	通道 19 采样时间 参考 SPT0
11:8	SPT18	通道 18 采样时间 参考 SPT0
7:4	SPT17	通道 17 采样时间 参考 SPT0
3:0	SPT16	通道 16 采样时间 参考 SPT0

29.5.10 ADC 配置寄存器（ADC_CONFIG）

地址偏移：0x24

复位值：0x8804_0ee1



位/位域	名称	描述
31:28	ADJ_TD_GA	ADC 温度传感器增益调整 温度传感器增益调整信号。默认 1000
27:24	ADJ_TD_OS	ADC 温度传感器偏移调整 温度传感器偏移调节信号。默认 1000
23:22	保留	必须保持复位值

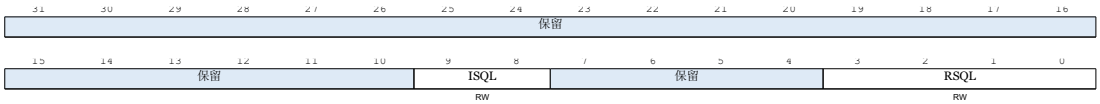
位/位域	名称	描述
21:20	ALG_MEAN	ADC 温度采样点选择 平均算法的温度采样点的选择 00：32 01：64 10：128 11：256
19	EN_BUF	ADC 外部参考输入缓冲区启用 检测到外部参考输入缓冲区启用信号，高活动。 当 EN_BUF 为 0 时，缓冲区输出为 0。 当 ADC_EN 为 1 时，如果 EN_BUF 设置为 0 到 1，缓冲区的开始时间为 3us@Typ，8us@Max。当 ADC_EN 为 0 时，控制信号 EN_BUF=为 0。
18	EN_TS	ADC 温度传感器使能 温度传感器启用信号，处于高激活状态。 当 ADC_EN 为 1 时，如果 EN_TS 设置为 0 到 1，则温度传感器的开始时间为 6us@Typ，10us@Max。 当 ADC_EN 为 0 时，控制信号 EN_TS=为 0。
17	保留	必须保持复位值
16	SELVI_HD_LS	ADC 输入类型选择 0：单端输入 1：差分输入
15:13	保留	必须保持复位值
12:9	TTRIM	ADC 内置参考电压温度比控制

位/位域	名称	描述
		内置参考电压温度比控制（默认=0111） 0110~0000：正温度趋势调整 1000~1111：负温度趋势调整
8:4	VTRIM	ADC 内置参考电压输出控制 内置参考电压输出控制（默认=01110）
3:2	SEL_VREFBI	ADC 内置参考输出电压选择 内置参考输出电压选择信号 00：1.5V 01：2.0V 10/11：2.5V
1	EN_HIZ	ADC 高阻态控制信号 当内置参考被关闭时的高电阻状态控制信号，高有效。 EN_HIZ 不受 ADC_EN 的控制
0	EN_VREFBI	ADC 内置参考电压使能 内置的参考电压启用信号，高有效。EN_VREFBI 不受 ADC_EN 的控制

29.5.11 ADC 转换长度寄存器（ADC_SQL）

地址偏移：0x28

复位值：0x0000_0000

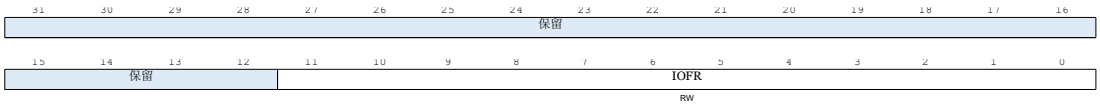


位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	ISQL	注入组长度 0~3：规则组长度为 1~4
7:4	保留	必须保持复位值
3:0	RSQL	规则组长度 0~15：规则组长度为 1~16

29.5.12 ADC 注入组偏移寄存器 x（ADC_IOFRx）

地址偏移：0x30+x*4（x=0~3)

复位值：0x0000_0000

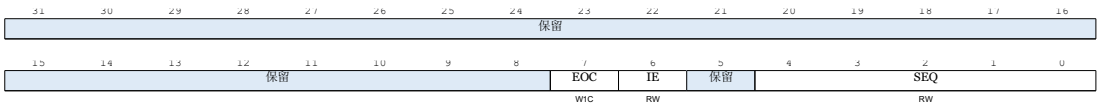


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	IOFR	注入组偏移值 注入组通道转换的数据值 IDR 已经减去了在 ADC_IOFR 寄存器中定义的偏移量。

29.5.13 ADC 规则组序列配置寄存器 x（ADC_RSQRx）

地址偏移：0x40+x*4（x=0~15)

复位值：0x0000_0000



位/位域	名称	描述
31:8	保留	必须保持复位值

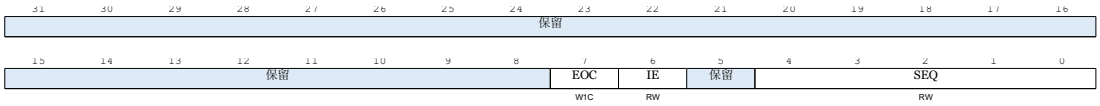
位/位域	名称	描述
7	EOC	转换结束标志 0：转换未结束 1：转换结束 注：该位除了可通过写 1 清除外，还可以通过读取相应规则组数据寄存器来清除。比如序列号为 0 的 ADC_RSQR0 的 EOC 位可通过读取 ADC_RDR0 来清除，ADC_RSQR1 的 EOC 位可通过读取 ADC_RDR1 来清除。
6	IE	中断使能 0：不使用 EOC 中断 1：使用 EOC 中断 注：该位仅用于配置该通道转换结束后是否触发中断。
5	保留	必须保持复位值
4:0	SEQ	规则组第 x 次转换的通道编号 00000：VIN1 00001：VIN2 00010：VIN3 00011：VIN4 00100：VIN5 00101：VIN6 00110：VIN7 00111：VIN8 01000：VIN9

位/位域	名称	描述
		01001：VIN10 01010：VIN11 01011：VIN12 01100：VIN13 01101：VIN14 01110：VIN15 01111：VIN16 1x000：VTEMP(Temperaturesensor) 1x001：VBAT 1x010：VR_EX1 1x011：VR_EX2 1x100：VR_EX3

29.5.14 ADC 注入组序列配置寄存器 x（ADC_ISQRx）

地址偏移：0x80+x*4（x=0~3)

复位值：0x0000_0000



位/位域	名称	描述
31:8	保留	必须保持复位值
7	EOC	转换结束标志 0：转换未结束 1：转换结束 注：该位除了可通过写 1 清除外，还可以通过读取相应注入组数据寄存器来清除。比如序列号为 0 的 ADC_ISQR0 的 EOC 位可通过读取 ADC_IDR0 来清

位/位域	名称	描述
		除，ADC_ISQR1 的 EOC 位可通过读取 ADC_IDR1 来清除。
6	IE	中断使能 0：不使用 EOC 中断 1：使用 EOC 中断 注：该位仅用于配置该通道转换结束后是否触发 EOC 中断。
5	保留	必须保持复位值
4:0	SEQ	注入组第 x 次转换的通道编号 00000：VIN1 00001：VIN2 00010：VIN3 00011：VIN4 00100：VIN5 00101：VIN6 00110：VIN7 00111：VIN8 01000：VIN9 01001：VIN10 01010：VIN11 01011：VIN12 01100：VIN13 01101：VIN14 01110：VIN15

位/位域	名称	描述
		01111：VIN16 1x000：VTEMP (Temperature sensor) 1x001：VBAT 1x010：VR_EX1 1x011：VR_EX2 1x100：VR_EX3

29.5.15 ADC 规则组数据寄存器 x（ADC_RDRx）

地址偏移：0x90+x*4（x=0~15)

复位值：0x0000_0000

位/位域	名称	描述
31:23	保留	必须保持复位值
22	PARITY	转换结果的偶校验值 软件可将[15:0]的数据寄存器结果进行偶校验计算，结果可与该 bit 值进行比较。若结果相同，说明软件读到的 ADC 数据与硬件提供的值相同。
21	保留	必须保持复位值
20:16	CH	转换结果对应的通道 该值与对应 ADC_RSQRx 中的 SEQ 的值相同，可供软件快速获取该寄存器 RDR 值对应的通道编号。
15:0	RDR	规则组数据寄存器

29.5.16 ADC 注入组数据寄存器 x（ADC_IDRx）

地址偏移：0xD0+x*4（x=0~3)

复位值：0x0000_0000

位/位域	名称	描述
31:23	保留	必须保持复位值
22	PARITY	转换结果的偶校验值 软件可将[15:0]的数据寄存器结果进行偶校验计算，结果可与该 bit 值进行比较。若结果相同，说明软件读到的 ADC 数据与硬件提供的值相同。
21	保留	必须保持复位值
20:16	CH	转换结果对应的通道 该值与对应 ADC_ISQRx 中的 SEQ 的值相同，可供软件快速获取该寄存器 IDR 值对应的通道编号。
15:0	IDR	注入组数据寄存器

30 数字/模拟转换控制器（DAC）

30.1 简介

DAC（包含模拟比较器 ACMP，以下称 DAC）为外采 IP， CPU 通过 AxiLite 总线控制 DAC 功能。

30.2 主要特点

- 8 位分辨率
- 集成非线性:±2 LSB
- 差分非线性:±1 LSB
- 模拟电源操作:2.97V ~ 3.63V
- 设置时间:≤1us
- 支持数字断电保护

30.3 系统框图

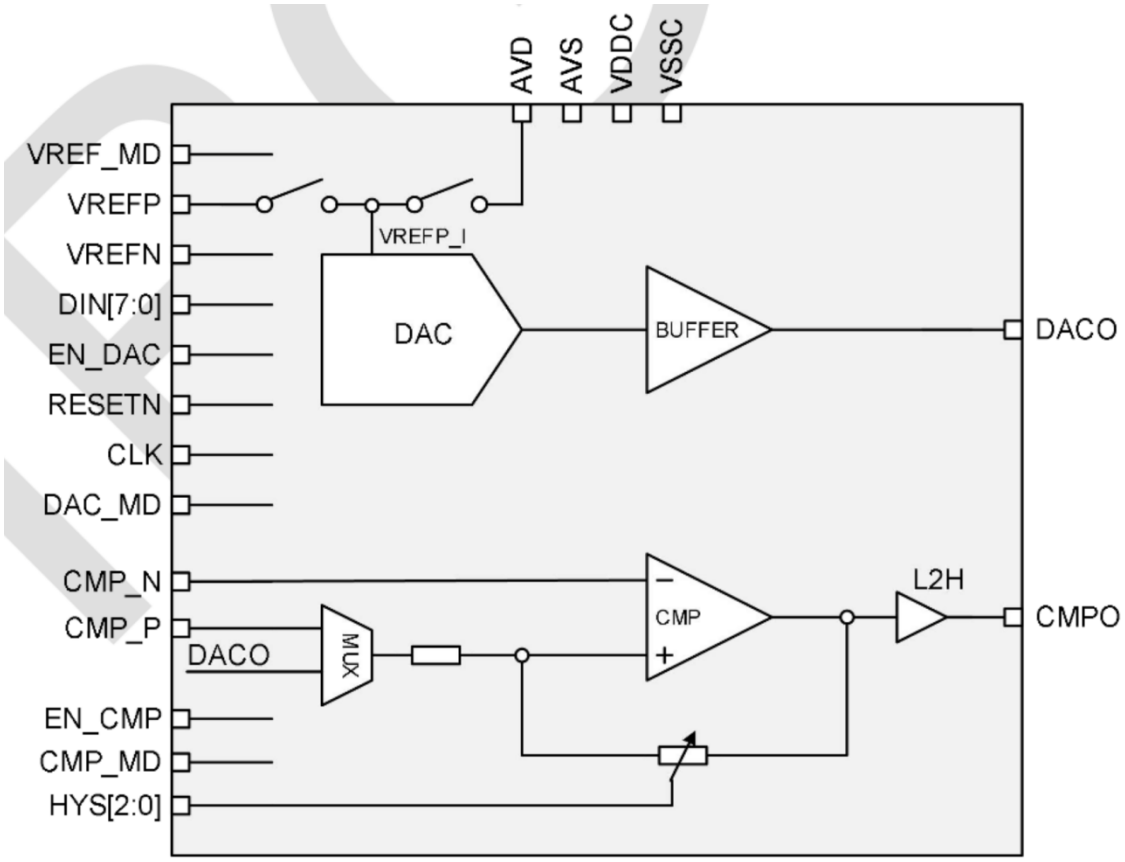


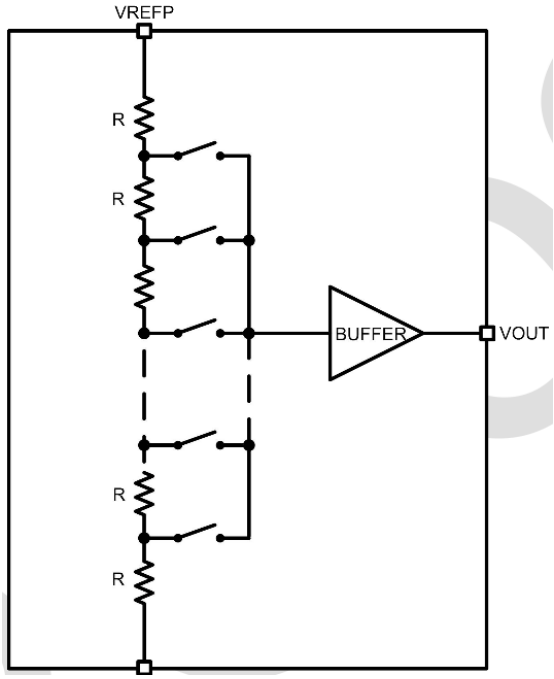
图 30.1 DAC 结构框图

DAC 的接口信号可以通过 AxiLite 接口直接控制。配置完 DAC_CLK_DIV 寄存器后时钟就会输入到 DAC 模块。

30.4 功能说明

30.4.1 数字模拟转换功能

DAC 模块是有 DAC 电阻网络和一个输出放大器组成如下图所示：

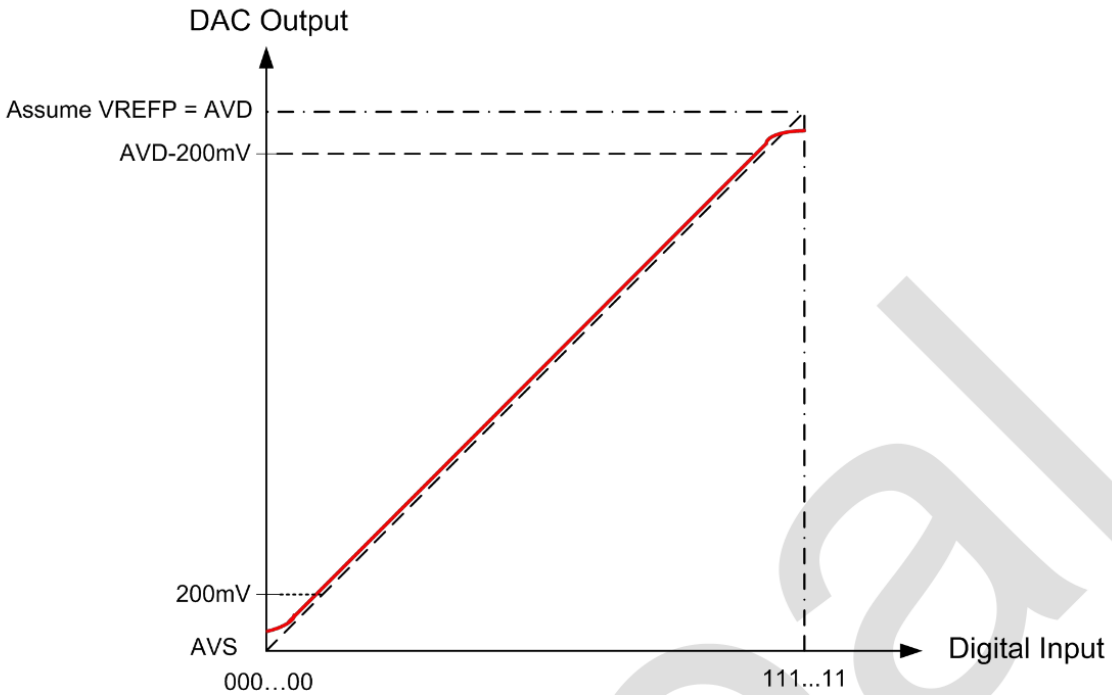


DAC 架构示意图

这个缓存器是一个轨到轨输出缓冲放大器，也就是说输出电压可以非常接近电源电压。当 DAC 正常工作且无负载时，输出电压可以摆动到电源电压和地。但是由于放大器本身的编译误差，输出在接近地是可能会有 0 到 15mV 的偏差。当 DAC 正常工作并连接一个 10KΩ的负载电阻时，其转换函数在一下输出范围内是线性的。

- 最小线性输出：FIX[0.2*256/VREFP]
- 最大线性输出：FIX[(AVD-0.2)*256/VREFP]

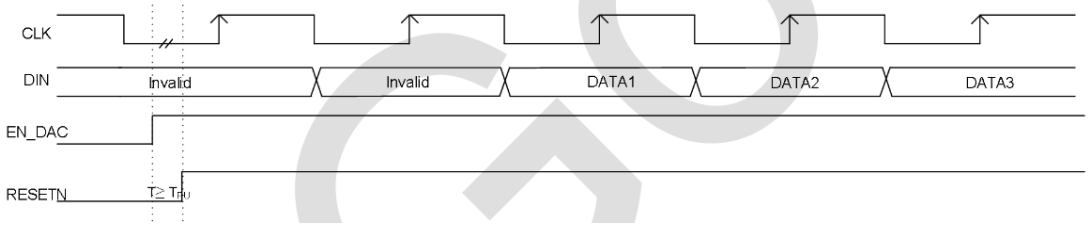
通常情况下，在 DAC 输出曲线的两个端点（接近地和电源电压）会出现线性度下降的现象。这是由于输出放大器在这些电压极端值辅警进入饱和状态，无法精确跟随输入数字码的变化，从而导致非线性。在实际应用中，只有当 DAC 的输出电压处于 200mV 到 AVD-200mV 的范围内时，才能保证 DAC 的线性指标规格要求。



DAC 线性输出示意图

DAC 的工作流程

1. 配置 DAC_CLK_DIV 寄存器，确保工作时钟低于 5MHz；
2. 在 RESETN 有效时配置 DAC_CTRL 寄存器。配置完寄存器后至少延迟 15ms，等待 DAC 获取配置信息；
3. 配置 DAC_CTRL 寄存器的 RESETN；
4. 配置 DAC_DOUT 寄存器，控制 DAC 输出电压。更新 DAC_OUT 的频率要低于工作时钟频率的 1/2。如果工作时钟是 5MHz，更新 DAC_OUT 的时间间隔只在 400ns 以上。



DAC 的工作配置时序图

30.4.2 模拟比较器功能

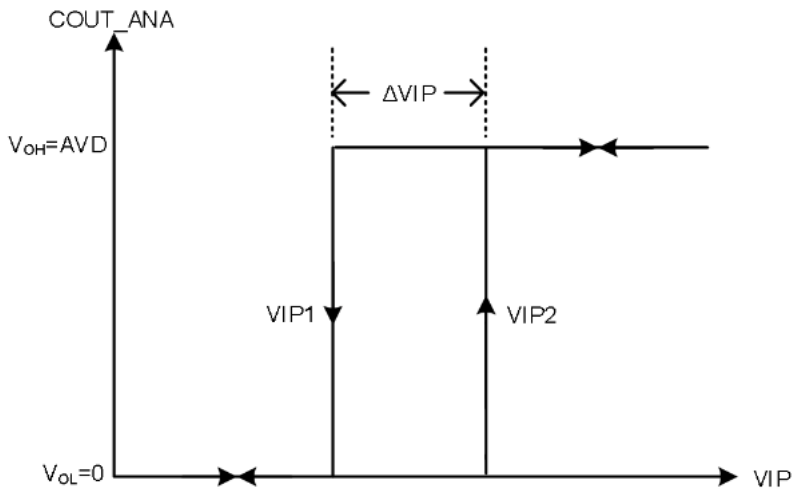
基础比较：当 HYS 为 0 时为基础的比较模式。当 VIP 电压高于 VIN 时输出高，也即 CPMO 为 1，反之 CMPO 为 0。

迟滞比较：当 HYS 为非 0 时为迟滞比较模式。当比较器的两个输入电压非常接近时，即使是很小的噪声也可能导致输出产生振荡或不稳定切换。这在实际系统中尤其成问题，因为噪声是不可避免的。输入信号（VIP）连接到比较器的非反相输入端，通过反馈电阻网络（R1 和 R2）提供正反馈，从而产生迟滞。

输出从高电平切换到低电平时的阈值计算为：VIP1 = [VIN (R1 + R2) \u2013 AVD X R1] / R2

输出从低电平切换到高电平时的阈值计算为：VIP2 = [VIN X (R1 + R2)]/ R2

迟滞电压（ΔVIP）计算为： ΔVIP = VIP2- VIP1 = AVD X R1/ R2



迟滞比较器

30.5 寄存器

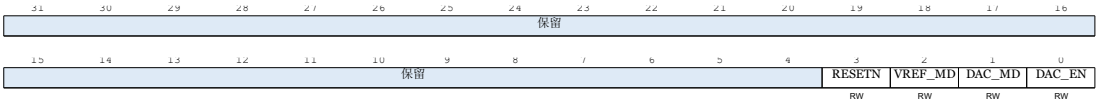
DAC0 基地址：0x41003000

DAC1 基地址：0x42002000

30.5.1 DAC 控制寄存器(DAC_CTRL)

地址偏移：0x0

复位值：0x00000000

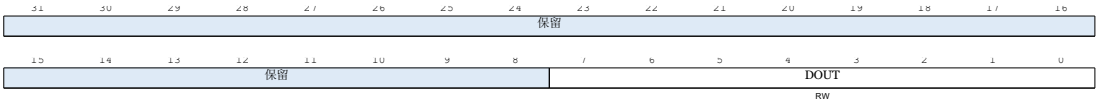


位/位域	名称	描述
31:4	保留	必须保持复位值
3	RESETN	DAC 复位 0:复位 1:取消复位
2	VREF_MD	DAC 参考电压 0:VREFP 1:AVD
1	DAC_MD	DACO 输出模式 0:Buffer 输出，用于增强驱动 1:Bypass 输出，直接输出
0	DAC_EN	DACO 使能 0:禁止，DACO 输出高阻 1:使能

30.5.2 DAC 数据寄存器(DAC_DOUT)

地址偏移：0x4

复位值：0x00000000

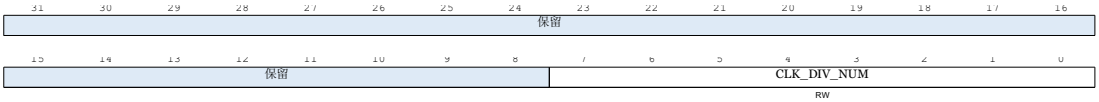


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DOUT	DAC 输出数据，8 位分辨率， Vout = VREFP_I / 255 * DOUT V

30.5.3 DAC 分频寄存器(DAC_CLK_DIV)

地址偏移：0x8

复位值：0x00000000

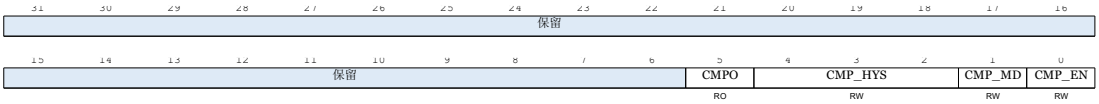


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CLK_DIV_NUM	DAC 时钟分频系数

30.5.4 CMP 控制寄存器(DAC_CMP_CTRL)

地址偏移：0xc

复位值：0x00000000



位/位域	名称	描述
31:6	保留	必须保持复位值
5	CMPO	CMP 比较结果
4:2	CMP_HYS	比较器迟滞电压选择信号 Hysteresis voltage(mV) @AVD=5V 000: 0 001: 3.3 010: 16.5 011: 33 100: 66 101: 165 110: 264 111: 330
1	CMP_MD	比较器正输入选择信号

位/位域	名称	描述
		0:DACO 1:CMP_P
0	CMP_EN	CMP 使能信号,CMP 不受 EN_DAC&RESETN 控制 0:无效 1:使能

31 硬件加密引擎（DSE）

硬件加密引擎的地址分配如下：

表 31.1 硬件加密引擎地址分配表

起始地址	结束地址	空间说明
4110_0000	4113_FFFF	SPACC
4114_0000	4114_7FFF	PKA
4115_0000	4115_0FFF	TRNG
4116_0000	4116_03FF	REG

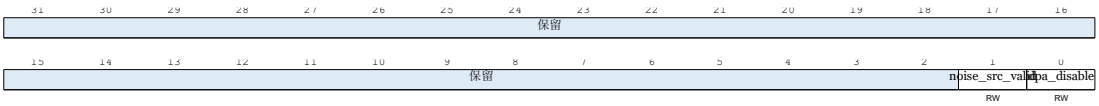
31.1 寄存器

DSE 基地址：0x41160000

31.1.1 PKA 接口控制寄存器(PKA_IF_CTRL)

地址偏移：0x0

复位值：0x00000001

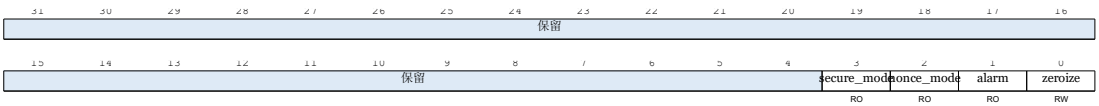


位/位域	名称	描述
31:2	保留	必须保持复位值
1	noise_src_valid	TA/DPA 噪声源启动信号
0	dpa_disable	TA/DPA 禁用信号

31.1.2 TRNG 接口控制寄存器(TRNG_IF_CTRL)

地址偏移：0x4

复位值：0x00000000



位/位域	名称	描述
31:4	保留	必须保持复位值
3	secure_mode	安全模式
2	nonce_mode	随机数模式
1	alarm	警报
0	zeroize	归零

31.2 对称加密与哈希算法加速引擎 （SPACC）

31.2.1 简介

大部分安全传输协议都有类似操作需求：1，需要加密算法用于对数据包做加解密操作; 2，需要哈希算法对加密后数据做完整性验证。SPACC 用于执行并行哈希算法和加密操作，可满足上述两点需求。

Note

SPACC 能实现各种对称加密算法，对于非对称加密算法需要由 PKA 模块负责。

SPAcc 体系结构包括：

1. Sequencer: 命令序列器
2. Cipher Core: 对称加密算法加速核。
3. Hash Core: 哈希算法加速核。
4. Data Buffer: 运算缓存器。
5. Key Buffer: 密钥及其他信息存储器。
6. SDMA：scatter-gather DMA 引擎，用于将加解密或者哈希处理后的数据 DMA 回 HOST 端的内存。

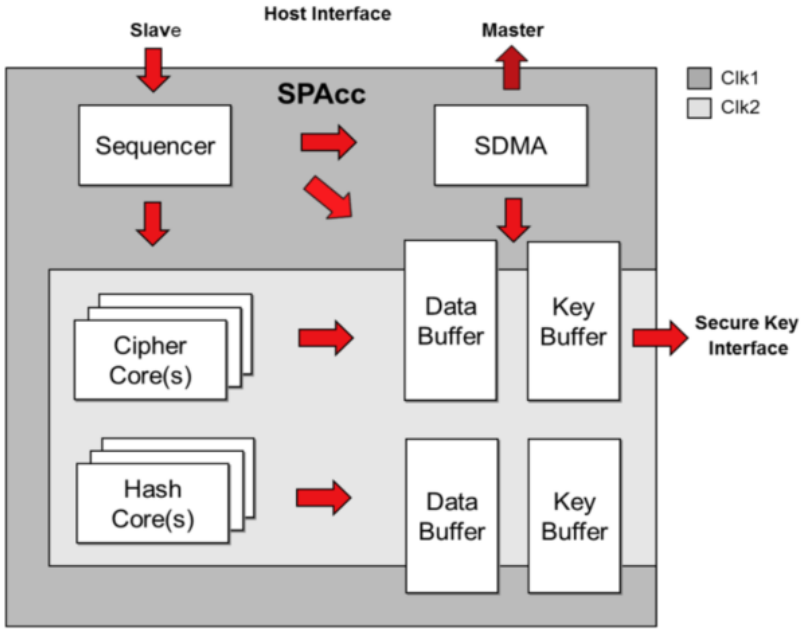


图 31.1 SPACC 整体框架图

31.2.2 主要特性

- 支持 AES128，AES192, AES256。
- 支持 SM4。
- AES 和 SM4 模式均支持 CBC，CCM，CFB, CMAC, CTR, ECB, F8, GCM, OFB, XCBC, XTS。
- 支持 SM3。
- 支持 DES3-CBC，DES3-ECB，DES-CBC，DES-ECB。

Note

SPACC 也支持 SHAKE，CSHAKE，HMAC，KASUMI，SHA，SHA3，CRC 等其他算法，为了文档简洁，后续不展开介绍。

31.2.3 操作描述

31.2.3.1 上下文管理

这里的上下文代表密钥，初始值等算法所需的常量。上下文存储在上下文存储器中(也被称为 context buffer 或 key buffer）。SPACC 有两个上下文存储器：加

密和哈希操作各一个。每个上下文存储器都以 PAGE 形式排列。每个 PAGE 都分配了一个索引。

当上下文被导入到某个 PAGE 后，HOST 端的软件需要将上下文的大小写到对应的 KEY_SZ 寄存器。KEY_SZ 寄存器有多个，一个 KEY_SZ 寄存器对应一个 PAGE 索引。

有些算法（比如 AE,SM4 等）有密钥扩展环节。密钥扩展就是将原密钥通过散列处理，以创建更长，更复杂的密钥。如果需要做密钥扩展操作，在 SPACC 中，第一包用于做密钥扩展，需要提前将 CTRL.KEY_EXP 标志位写 1。第一包处理完后，将 CTRL.KEY_EXP 清 0，然后处理接下去的数据包。

SPACC 也支持将扩展后的密钥提前写入上下文存储器中，这样接下去做加密算法的时候，可以不需要再用第一包来做密钥扩展操作。对于这种操作方法，需要将如下寄存器都写 0：CTRL.KEY_EXP，PRE_AAD_LEN,POST_AAD_LEN,PROC_LEN。

PAGE 内的数据结构依赖于对应的算法。每个算法有对应的 PAGE 内数据结构。

当对一个数据包同时进行哈希和加密算法处理时，哈希和加密的上下文必须被存储在相同索引的两个 PAGE 中。

加密和哈希算法的 IV 值，可以被存储在上下文存储器，或者可以被嵌入在输入 SPACC 的待处理数据包中。IV 值嵌入在数据包中，需要提前设置 IV_OFFSET 寄存器。如果需要同时做加密和哈希算法处理，则 IV 值必须被嵌入在数据包中。加密算法的 IV 置于哈希算法的 IV 之前。

31.2.3.1.1 AES 和 SM4 的上下文映射

下图是 AES 和 SM4 的上下文（PAGE）的数据结构，包含：

1. 地址 0x0 - 0x1f: 共 32 字节，存放 KEY。
2. 地址 0x20 - 0x2f: 共 16 字节，存放 IV，CTR，Y0。
3. 地址 0x30 - 0x3f： 共 16 字节，f8 Salt Mask, XTS Key。

Note

- KEY 是密钥。

- IV 是 CBC/CFB/OFB 的 IV 值。在完成 CBC 操作后，IV 这个区域会被上下文的 XOR 值自动替代。
- CTR 是 CCM 模式的初始计数器值。
- Y0 用于 GCM,GMAC 模式。
- f8 Salt Mask 用于 f8 模式的的 key-salt mask。
- XTS Key 用于存放 AES-XTS 模式的 KEY2。

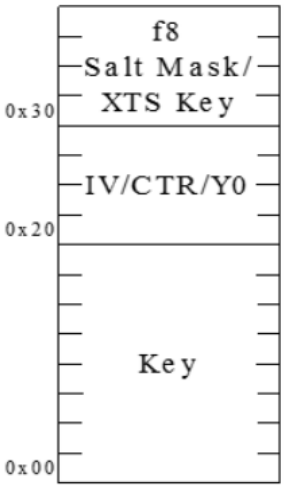


图 31.2 AES 和 SM4 的上下文映射

31.2.3.1.2 DES 的上下文映射

1. 地址 0x0 - 0x7: 共 8 字节，存放 IV。
2. 地址 0x8 - 0xf: 共 8 字节，存放 Key0。
3. 地址 0x10 - 0x17: 共 8 字节，存放 Key1。
4. 地址 0x18 - 0x1f: 共 8 字节，存放 Key2。

Note

- KEY0,KEY1,KEY2 是支持三重 DES 加密对应的三个密钥，如果只做单次 DES 加密，则只需填写 KEY0。
- IV 是 CBC 的 IV 值。在完成 CBC 操作后，IV 这个区域会被上下文的 XOR 值自动替代。SPACC 的 DES 支持 CBC 和 ECB 模式。

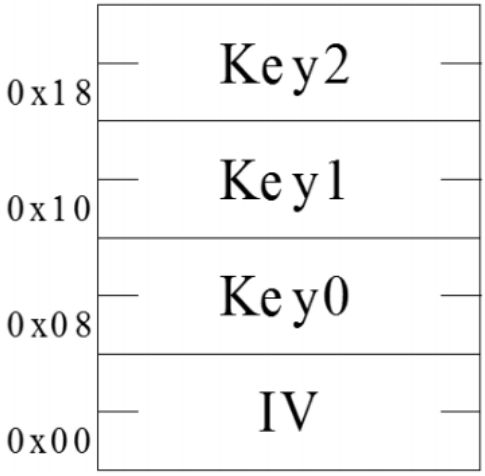


图 31.3 DES 的上下文映射

31.2.3.2 命令和状态寄存器

SPACC 包含命令 FIFO 和状态 FIFO。

为了发送一个包数据处理任务到 SPACC，需要设置 SRC_PTR, DST_PTR, OFFSET, PRE_AAD_LEN, POST_AAD_LEN, PROC_LEN, ICV_LEN, ICV_OFFSET, IV_OFFSET, SW_CTRL, AUX_INFO, and CTRL。CTRL 寄存器必须最后一个被设置，因为设置 CTRL 寄存器，将引起待处理的任务被 PUSH 进命令 FIFO 中。如果 FIFO_STAT.CMDx_FULL 标志位是 1 时，代表 FIFO 已满，不能再写 CTRL 寄存器。

当数据包处理结束，处理结果被写入状态 FIFO 中。HOST 端软件可以通过轮询读取 FIFO_STAT 寄存器或者通过中断来判断处理是否完成。如果确定已处理结束，HOST 端软件可以写 STAT_POP 寄存器来将 SPACC 处理结果从状态 FIFO 中读出，并自动转移到 STATUS 寄存器。

31.2.3.2.1 命令 FIFO 优先级

SPACC 包含 3 个命令 FIFO，优先级最高的是 FIFO0，最低的是 FIFO2。优先级最高的 FIFO 中的处理任务会被优先处理。HOST 端软件可以通过往不同命令 FIFO 中写任务，来调节命令的优先级。

Note

因为命令 FIFO 存在优先级，导致处理结果并不是按发起任务的时间先后来返回。HOST 端软件可以通过 SW_CTRL.SW_ID 来识别状态 FIFO 中的处理结果是来源于哪个任务。

31.2.3.2.2 中断产生

SPACC 可以开启相关中断位，用于产生命令 FIFO 和状态 FIFO 中断。当 FIFO 达到特定设定条件的时候，将会产生中断。可以通过 IRQ_CTRL 寄存器来设置 FIFO 的中断条件。典型的使用场景是：HOST 端软件开启 2 个线程，一个线程用于不停的往 SPACC 喂待处理的任务，直到命令 FIFO 满，线程可以开始 SLEEP。如果命令 FIFO 中断产生，将唤醒线程。另一个线程开始默认处于 SLEEP 状态，如果有接收到状态 FIFO 的中断，则唤醒线程，并从状态 FIFO 中读取处理结果，直到状态 FIFO 又为空时，重新进入 SLEEP。

STAT_WD 中断用于 HOST 及时 flush 状态 FIFO.当使能 STAT_WD 中断后，如果状态 FIFO 不为空时，SPACC 内部计数器将开始计时。一旦计数器的值达到 STAT_WD_CTRL 中设定的阈值，则将触发中断，HOST 端软件须及时从状态 FIFO 中读取相应的处理结果。

当发生的中断事件时，SPACC 会设置 IRQ_STAT 寄存器中相应的中断标志位。往中断标志位写 1，可以清除中断位。中断信号 当以下等式成立时，则产生中断：

IRQ_PIN = IRQ_EN.GLBL_EN & ((IRQ_EN.CMD0_EN & IRQ_STAT.CMD0) | (IRQ_EN.CMD1_EN & IRQ_STAT.CMD1) | (IRQ_EN.CMD2_EN & IRQ_STAT.CMD2) | (IRQ_EN.STAT_EN & IRQ_STAT.STAT) | (IRQ_EN.RC4_DMA_EN & IRQ_STAT.RC4_DMA) | (IRQ_EN.STAT_WD_EN & IRQ_STAT.STAT_WD));

31.2.3.3 包格式

数据包存储在主机内存空间中，如果启用了 DDT 模式 DMA，则数据包可能会分段。SPAcc 利用指向存储在主机内存空间中的 DDT 结构的指针收集来自 SRC 缓冲区的数据，并将算法处理结果片段的写回 DST 缓冲区。

通用数据包的格式如下图所示。

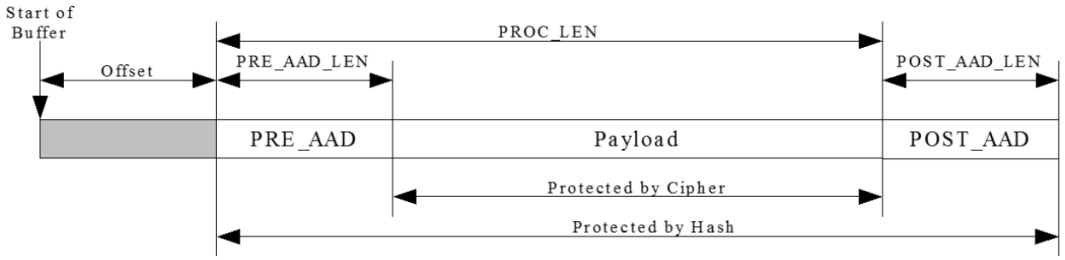


图 31.4 数据包格式定义图

- **AAD**：Additional Authentication Data，附加身份验证信息，不同加密场景，可以代表不同含义。比如 IPsec-ESP 的 AAD 为 ESP Header，IPsec-AH 的 AAD 为 Mod Header。
- **Payload**：数据包本体。
- **PRE_AAD**：Payload 前面的 AAD，和 Payload 合一起做加/解密处理。
- **POST_AAD**：Payload 后面的 AAD，不参与加/解密处理。当存在 ICV，且 ICV 由密文计算出时，POST_AAD 的长度需固定为 0。
- **ICV**：Integrity Check Value。完整性校验值，可由明文或者密文计算出。如果 ICV 需要和 Payload 一起参与加/解密过程，则 ICV 必须由明文计算出，且需固定放在 Payloay 后面（不能由 ICV_OFFSE 指引）。如果 ICV 由密文计算出，则 ICV 位置由 ICV_OFFSET 决定。上图未列出 ICV，因为 ICV 不一定存在。
- **PROC_LEN**：参与加/解密处理的数据长度，包含 Payload 和 PRE_AAD，也可能包含 ICV。
- **Offset**：PRE_ADD 起始地址相对 DDT 中定义的 SRC/DST 地址的偏移。
- **POST_AAD_LEN**：包含任何不参与加解密的数据段。如果 Payload 长度为 0，则 POST_AAD_LEN 也必须为 0。如果 ICV_ENC 寄存器被设置，则 POST_AAD_LEN 也必须为 0。

31.2.3.3.1 ICV 处理

ICV_OFFSET 是数据包数据结构中到 ICV 开头的字节偏移量。此偏移量为相对于数据包缓冲区的起始位置。它指的是解密时和源缓冲区的 OFFSET 或者加密时和目标缓冲区的 OFFSET。如果 CTRL.ICV_APPEND 字段已设置，则 ICV_OFFSET 被忽略。

ICV_APPEND 寄存器用于选择 ICV 是否出现在有效载荷的末尾。设置了这个寄存器会覆盖 ICV_OFFSET 值。

ICV_PT 寄存器用于选择是通过明文有效载荷还是密文有效载荷来计算出 ICV。如果设置此寄存器，则通过明文计算 ICV。如果清除，则通过密文计算 ICV。

ICV 在加密和解密操作中的处理方式不同：

- 在加密过程中，ICV 是通过 AAD 和有效载荷计算的。数据包完成加密，被写到目标缓冲区的时候会同时将 ICV 插入 ICV_OFFSET 定义的位置。如果设置了 ICV_ENC 标志，则 ICV 将附加到有效载荷的末尾，并且在放入目标缓冲区之前进行加密。如果设置了 ICV_APPEND（没有 ICV_ENC） ICV 写入有效载荷末尾的目标缓冲区。
- 解密过程中，ICV 通过 AAD 和有效载荷进行计算。源缓冲区中存有主机端提前写入的 ICV，具体起始地址由 ICV_OFFSET 寄存器提供。从源缓存区提取的 ICV 与计算出的 ICV 进行比较，并将比较结果写入状态 FIFO 中。如果设置 ICV_APPEND（没有 ICV_ENC），则 ICV 存在紧邻源缓冲区中的 POST_AAD 的后面字节位置。

31.2.3.4 DMA 选项

数据描述符表 （DDT） 格式是一种描述符，旨在通过提供以 NULL 结尾的 pointer-length 对列表。通过 pointer 获得每个分段数据的起始地址。通过 length 获得每个分段数据的长度。每个数据段的长度以字节为单位。DDT 的地址本身必须是 8 字节对齐的。

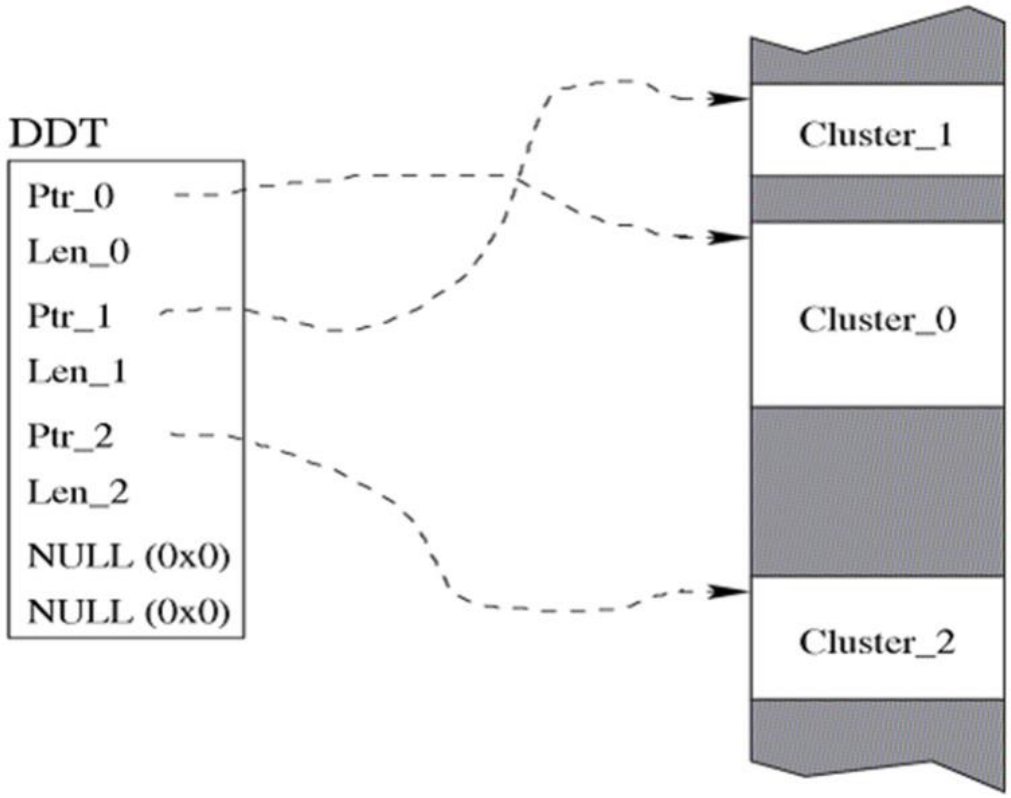


图 31.5 DDT 帧格式图

HOST 端软件需要建立两个 DDT 结构体，分别用于描述源缓冲区和目的缓冲区的 pointer-length 链表。然后往 SRC_PTR 和 DST_PTR 寄存器写 DDT 本身的物理地址。当 SPACC 开始处理数据包时，会用 DMA 的方式主动从源缓冲区中读取数据，处理完后，将处理后的结果再以 DMA 的方式写入目的缓冲区。

31.2.4 操作流程

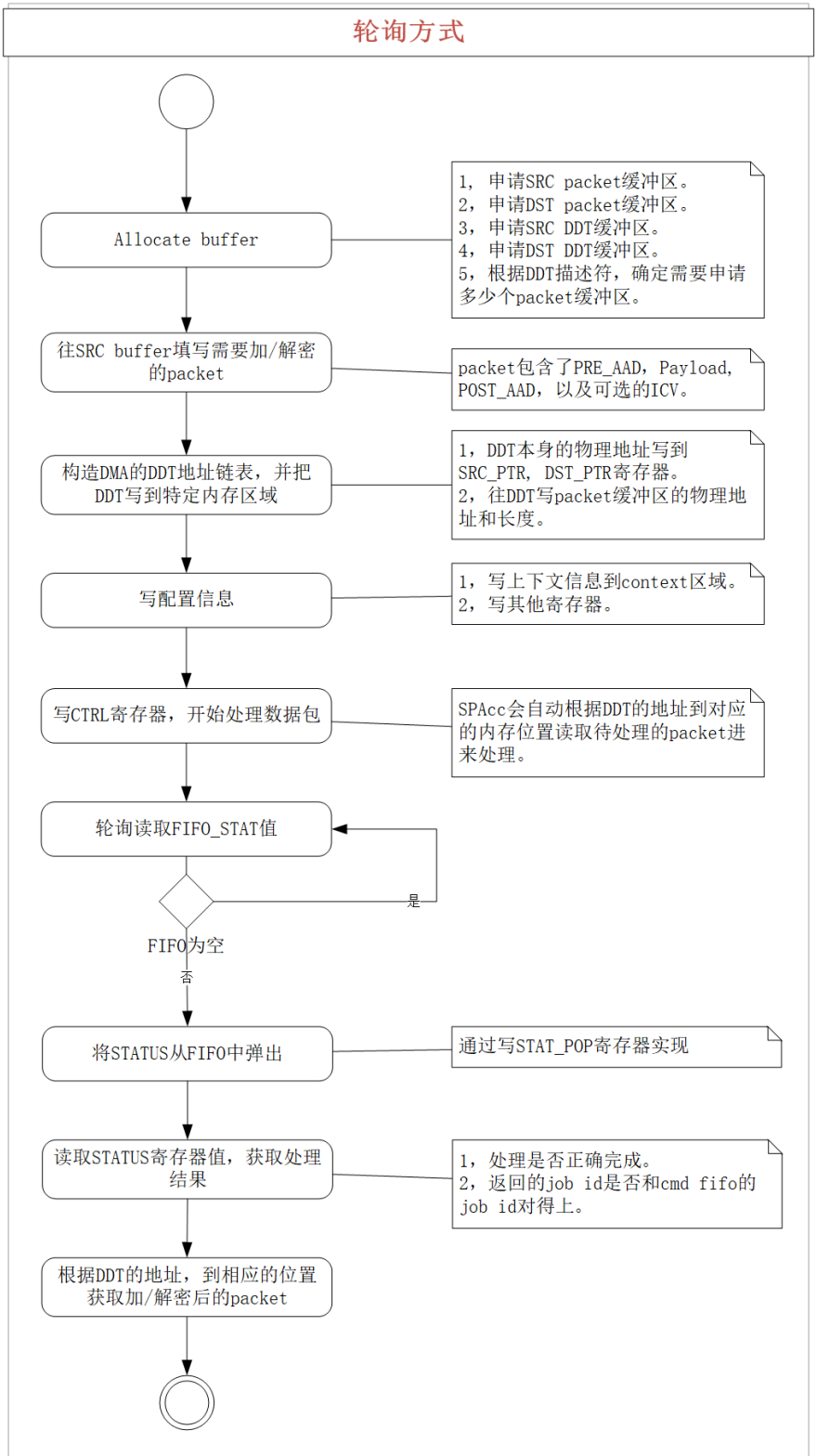


图 31.6 操作流程图

Note

中断模式需要设置 IRQ_EN, IRQ_CTRL 寄存器

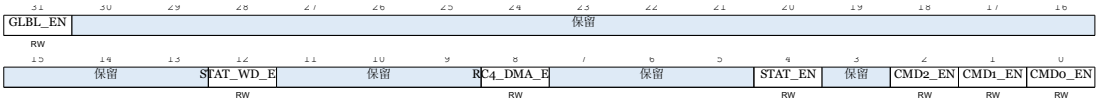
31.2.5 寄存器

SPACC 基地址：0x41100000

31.2.5.1 中断使能寄存器(IRQ_EN)

地址偏移：0x0

复位值：0x00000000



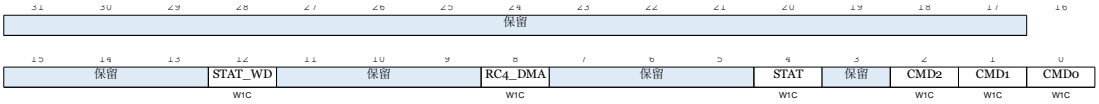
位/位域	名称	描述
31	GLBL_EN	中断源的全局使能 0x0 (DIS): 禁用所有中断源 0x1 (ENA): 使能所有中断源
30:13	保留	必须保持复位值
12	STAT_WD_EN	状态看门狗的中断使能 0x0 (DIS): 禁用此中断 0x1 (ENA): 使能此中断
11:9	保留	必须保持复位值
8	RC4_DMA_EN	RC4 context DMA 的中断使能 0x0 (DIS): 禁用此中断 0x1 (ENA): 使能此中断
7:5	保留	必须保持复位值
4	STAT_EN	状态 FIFO 的中断使能 0x0 (DIS): 禁用此中断 0x1 (ENA): 使能此中断
3	保留	必须保持复位值
2	CMD2_EN	优先级为 2 的命令 FIFO 中断使能

位/位域	名称	描述
		0x0 (DIS): 禁用此中断 0x1 (ENA): 使能此中断
1	CMD1_EN	优先级为 1 的命令 FIFO 中断使能 0x0 (DIS): 禁用此中断 0x1 (ENA): 使能此中断
0	CMD0_EN	优先级为 0 的命令 FIFO 中断使能 0x0 (DIS): 禁用此中断 0x1 (ENA): 使能此中断

31.2.5.2 中断状态寄存器(IRQ_STAT)

地址偏移：0x4

复位值：0x00000000



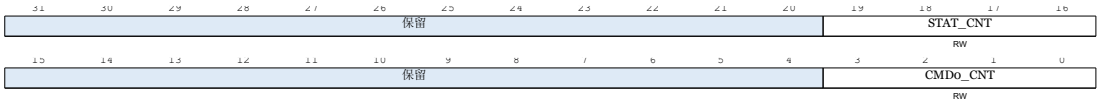
位/位域	名称	描述
30:13	保留	必须保持复位值
12	STAT_WD	状态看门狗的中断状态 0x0 (INACTIVE): 中断无效 0x1 (ACTIVE): 中断有效
11:9	保留	必须保持复位值
8	RC4_DMA	RC4 context DMA 的中断状态 0x0 (INACTIVE): 中断无效 0x1 (ACTIVE): 中断有效
7:5	保留	必须保持复位值
4	STAT	状态 FIFO 的中断状态 0x0 (INACTIVE): 中断无效 0x1 (ACTIVE): 中断有效

位/位域	名称	描述
3	保留	必须保持复位值
2	CMD2	优先级为 2 的命令 FIFO 的中断状态 0x0 (INACTIVE): 中断无效 0x1 (ACTIVE): 中断有效
1	CMD1	优先级为 1 的命令 FIFO 的中断状态 0x0 (INACTIVE): 中断无效 0x1 (ACTIVE): 中断有效
0	CMD0	优先级为 0 的命令 FIFO 的中断状态 0x0 (INACTIVE): 中断无效 0x1 (ACTIVE): 中断有效

31.2.5.3 中断控制寄存器(IRQ_CTRL)

地址偏移：0x8

复位值：0x00000000

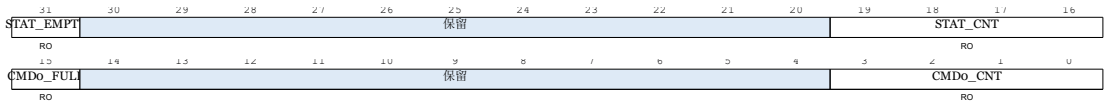


位/位域	名称	描述
31:20	保留	必须保持复位值
19:16	STAT_CNT	设置状态 FIFO 中存储个数的阈值，用于触发一个状态 FIFO 中断。
15:4	保留	必须保持复位值
3:0	CMD0_CNT	设置优先级为 0 的命令 FIFO 中存储个数的阈值，用于触发一个命令 FIFO 中断。

31.2.5.4 FIFO 状态寄存器(FIFO_STAT)

地址偏移：0xc

复位值：0x80000000

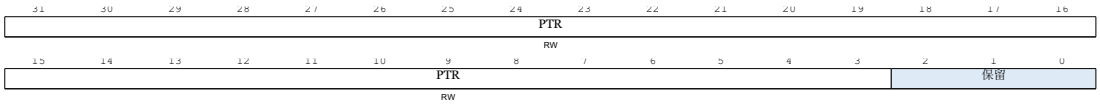


位/位域	名称	描述
31	STAT_EMPTY	状态 FIFO 空标志位 0x0 (INACTIVE): 当前为不空 0x1 (ACTIVE): 当前为空
30:20	保留	必须保持复位值
19:16	STAT_CNT	当前存储在状态 FIFO 中的数据个数。
15	CMD0_FULL	优先级为 0 的命令 FIFO 的满标志位 0x0 (INACTIVE): 当前为不满 0x1 (ACTIVE): 当前为满
14:4	保留	必须保持复位值
3:0	CMD0_CNT	当前存储在 CMD0 FIFO 中的数据个数

31.2.5.5 源指针寄存器(SRC_PTR)

地址偏移：0x20

复位值：0x00000000

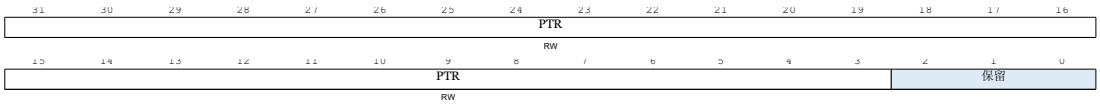


位/位域	名称	描述
31:3	PTR	存储源数据包的系统内存首地址
2:0	保留	必须保持复位值

31.2.5.6 目的指针寄存器(DST_PTR)

地址偏移：0x24

复位值：0x00000000

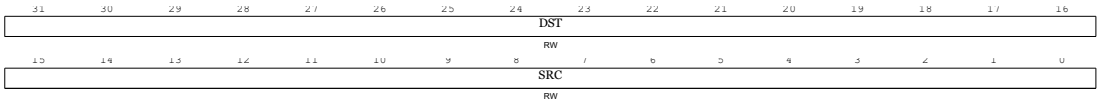


位/位域	名称	描述
31:3	PTR	存储目的数据包的系统内存首地址
2:0	保留	必须保持复位值

31.2.5.7 偏移寄存器(OFFSET)

地址偏移：0x28

复位值：0x00000000

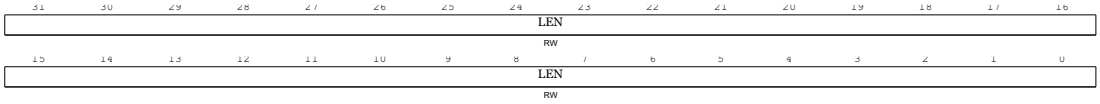


位/位域	名称	描述
31:16	DST	目的缓冲地址偏移值
15:0	SRC	源缓冲地址偏移值

31.2.5.8 Prefixed AAD 长度寄存器(PRE_AAD_LEN)

地址偏移：0x2c

复位值：0x00000000

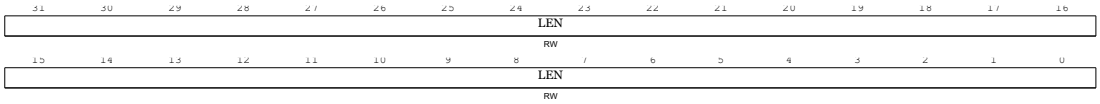


位/位域	名称	描述
31:0	LEN	Prefixed AAD 字节长度

31.2.5.9 Postfixed AAD 长度寄存器(POST_AAD_LEN)

地址偏移：0x30

复位值：0x00000000

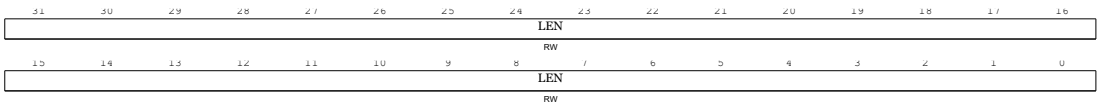


位/位域	名称	描述
31:0	LEN	Postfixed AAD 字节长度

31.2.5.10 包长度寄存器(PROC_LEN)

地址偏移：0x34

复位值：0x00000000

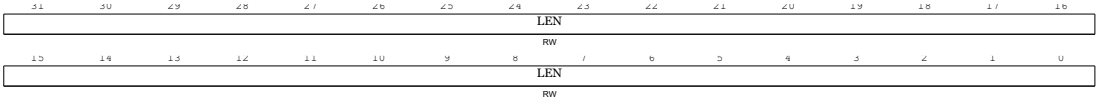


位/位域	名称	描述
31:0	LEN	包长度寄存器，包含 PRE_AAD_LEN 和 payload 的包长度

31.2.5.11 ICV 长度寄存器(ICV_LEN)

地址偏移：0x38

复位值：0x00000000

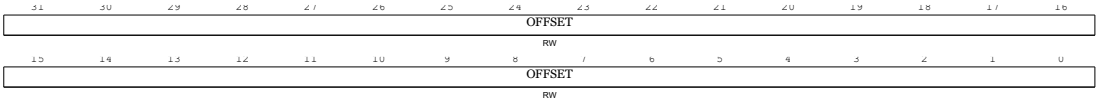


位/位域	名称	描述
31:0	LEN	ICV 长度寄存器。如果设置为 0，采用默认长度。

31.2.5.12 ICV 偏移寄存器(ICV_OFFSET)

地址偏移：0x3c

复位值：0x00000000

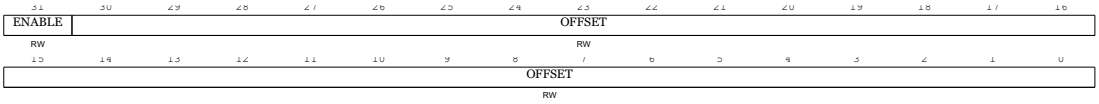


位/位域	名称	描述
31:0	OFFSET	ICV 相对于 packet data 的地址偏移，指向解密流程中的源缓冲和加密流程中的目的缓冲。

31.2.5.13 IV 偏移寄存器(IV_OFFSET)

地址偏移：0x40

复位值：0x00000000

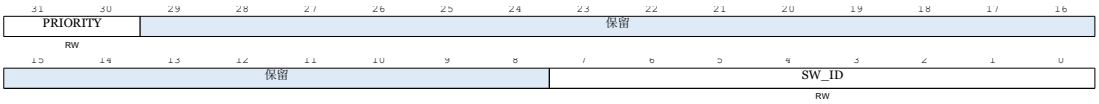


位/位域	名称	描述
31	ENABLE	控制 IV 值的位置 0x0 (IV_SRC0)：IV 值从 context page 中获取。 0x1 (IV_SRC1)：IV 值从源缓冲中获取
30:0	OFFSET	IV 值在源缓冲中的字节偏移值值。

31.2.5.14 软控制寄存器(SW_CTRL)

地址偏移：0x44

复位值：0x00000000



位/位域	名称	描述
31:30	PRIORITY	选择用哪个优先级的命令 FIFO 用于发送命令
29:8	保留	必须保持复位值
7:0	SW_ID	软件 job ID，在写入命令 FIFO 的时候会自动加 1，并从状态 FIFO 中返回。

31.2.5.15 Auxiliary 信息寄存器(AUX_INFO)

地址偏移：0x48

复位值：0x00000000

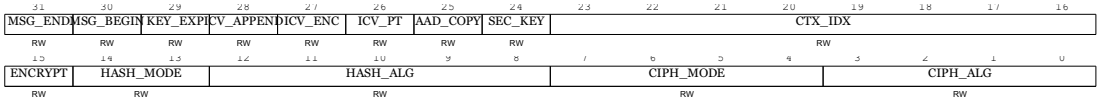
位/位域	名称	描述
31:18	保留	必须保持复位值
17:16	CBC_CS_SEL	为 CBC 算法选择 CTS 模式。（只用于 AES 和 SM4） 0x0 (NONE): Mode = none. 0x1 (CS1): Mode = CBC-CS1. 0x2 (CS2): Mode = CBC-CS2. 0x3 (CS3): Mode = CBC-CS3.
15:4	保留	必须保持复位值
3	CRC_INV_OUT_BA2	此域的含义需要基于选择的哈希算法 CRC_INV_OUT: 对于 CRC-32 操作，对输出的所有 bit 做取反 BA[2]: BA[2:0]的 Bit2.对于 KASUMI-f9，SNOW 3G-UIA2 和 ZUC 128-EIA3 算法，BA[2:0]用于指示包里最

位/位域	名称	描述
		后一个字节的有效 bit 数目,如果 BA[2:0]设置为 0，代表最后一个字节有 8bit 有效数据
2	CRC_REF_OUT_BA1	此域的含义需要基于选择的哈希算法 CRC_REF_OUT: 对于 CRC-32 操作，对 32 位输出结果做 bit reflected. BA[1]: BA[2:0]的 Bit1.对于 KASUMI-f9，SNOW 3G-UIA2 和 ZUC 128-EIA3 算法，BA[2:0]用于指示包里最后一个字节的有效 bit 数目,如果 BA[2:0]设置为 0，代表最后一个字节有 8bit 有效数据
1	CRC_REF_IN_BA0	此域的含义需要基于选择的哈希算法 CRC_REF_IN: 对于 CRC-32 操作，此位用于对输入做 bit reflected。 BA[0]: BA[2:0]的 Bit0.对于 KASUMI-f9，SNOW 3G-UIA2 和 ZUC 128-EIA3 算法，BA[2:0]用于指示包里最后一个字节的有效 bit 数目,如果 BA[2:0]设置为 0，代表最后一个字节有 8bit 有效数据
0	DIR_ARB_LEN	此域的含义需要基于选择的哈希算法 DIR: 对于 KASUMI-f9，此域用于设置方向位（uplink/downlink） ARB_LEN: 对于 KMAC，如果此 bit 为 1，代表 arbitrary-length 输出前缀编码被使用。 0x0 （DIS_UPLINK）：对于 KMAC：选择固定长度。对于 KASUMI-f9：选择上行链路。 0x1 （ENA_DOWNLINK）：对于 KMAC：选择任意长度。对于 KASUMI-f9：选择下行链路。

31.2.5.16 主控制寄存器(CTRL)

地址偏移：0x4c

复位值：0xc0000000



位/位域	名称	描述
31	MSG_END	Partical packet 处理结束标志位。 0x0 (non-last): 指示不是包的最后一段 0x1 (last): 指示是包的最后一段
30	MSG_BEGIN	Partical packet 处理开始标志位。 0x0 (non-first): 指示不是包的第一段 0x1 (first): 指示是包的第一段
29	KEY_EXP	在当前上下文（context）中扩展密钥 0x0 (DIS): 不扩展密钥 0x1 (EN): 扩展密钥
28	ICV_APPEND	The ICV appears at the end of the packet. Overrides the ICV_OFFSET register. 用于指示从哪里获取 ICV 值 Values: 0x0 (offset): ICV location indicated by ICV OFFSET. 从 ICV_OFFSET 寄存器中获取 ICV 值。 0x1 (append): ICV located at end of packet. 从包的最后一段获取 ICV 值。
27	ICV_ENC	指示 ICV 是否需要被加密。如果被加密，则 ICV_PT 和 ICV_APPEND 标志位需要被设置为 1，而且 POST_AAD_LEN 也需要设为 0. 0x0 (DIS): ICV 不加密 0x1 (EN): ICV 加密

位/位域	名称	描述
26	ICV_PT	用于指示 ICV 通过何种方式被计算出来。 0x0 (PLAIN): ICV 通过明文被计算出来。 0x1 (CIPHER): ICV 通过密文被计算出来。
25	AAD_COPY	拷贝 PRE_AAD 到目的缓冲区 0x0 (DIS): 不拷贝 0x1 (EN): 拷贝
24	SEC_KEY	用于指示加密密钥信息从哪里获取 0x0 (CTX): 从 context 中获取 0x1 (SKP): 从 SKP 接口中获取
23:16	CTX_IDX	Context 页码设置
15	ENCRYPT	设置加密还是解密 0x0 (DEC): Decrypt. 解密 0x1 (ENC): Encrypt. 加密
14:13	HASH_MODE	哈希算法模式设置 0x0 (RAW_SHAKE): 对于 MD5, SM3 和 SHA-1/2/3，设置 Mode=Raw. 对于基于 XOF 的 SHAKE，设置 Mode=SHAKE 0x1 (SSLMAC_cSHAKE): 对于 MD5, SM3 和 SHA-1/2/3，设置 Mode=SSLMAC. 对于基于 XOF 的 SHAKE，设置 Mode= cSHAKE 0x2 (HMAC_KMAC): 对于 MD5, SM3 和 SHA-1/2/3，设置 Mode=HMAC. 对于基于 XOF 的 SHAKE，设置 Mode=KMAC
12:8	HASH_ALG	选择哪款哈希算法。注意：如果选择 AEAD 加密算法，则此域需要选择 NULL。 0x0 (NULL): Hash algorithm is NULL. 0x1 (MD5): Hash algorithm is MD5.

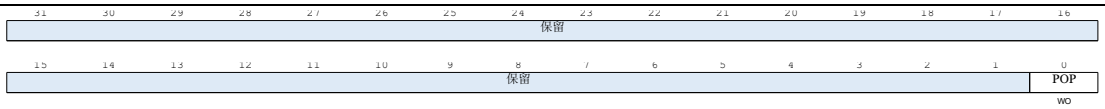
位/位域	名称	描述
		0x2 (SHA_1): Hash algorithm is SHA-1. 0x3 (SHA_224): Hash algorithm is SHA-224. 0x4 (SHA_256): Hash algorithm is SHA-256. 0x5 (SHA_384): Hash algorithm is SHA-384. 0x6 (SHA_512): Hash algorithm is SHA-512. 0x7 (AES_XCBC_MAC): Hash algorithm is AES-XCBCMAC. 0x8 (AES_CMAC): Hash algorithm is AES-CMAC. 0x9 (KASUMI_F9): Hash algorithm is KASUMI-f9. 0xa (SNOW_3G_UIA2): Hash algorithm is SNOW 3GUIA2. 0xb (CRC_32_IEEE_802_3): Hash algorithm is CRC-32-IEEE 802.3. 0xc (ZUC_128_EIA3): Hash algorithm is ZUC 128-EIA3. 0xd (SHA_512_224): Hash algorithm is SHA-512/224. 0xe (SHA_512_256): Hash algorithm is SHA-512/256. 0xf (MICHAEL): Hash algorithm is Michael. 0x10 (SHA3_224): Hash algorithm is SHA3-224. 0x11 (SHA3_256): Hash algorithm is SHA3-256. 0x12 (SHA3_384): Hash algorithm is SHA3-384. 0x13 (SHA3_512): Hash algorithm is SHA3-512. 0x14 (SHAKE128): Hash algorithm is SHAKE128. 0x15 (SHAKE256): Hash algorithm is SHAKE256. 0x16 (POLY1305): Hash algorithm is Poly1305. 0x17 (SM3): Hash algorithm is SM3. 0x18 (SM4_XCBC_MAC): Hash algorithm is SM4-XCBCMAC. 0x19 (SM4_CMAC): Hash algorithm is SM4-CMAC.

位/位域	名称	描述
7:4	CIPH_MODE	密码算法模式设置 0x0 (ECB): Block cipher mode is ECB. 0x1 (CBC): Block cipher mode is CBC. 0x2 (CTR_STREAM): Block cipher mode is CTR or ChaCha20 mode is Stream Cipher. 0x3 (CCM): Block cipher mode is CCM. 0x5 (GCM_AEAD): Block cipher mode is GCM or ChaCha20 mode is AEAD. 0x7 (OFB): Block cipher mode is OFB. 0x8 (CFB): Block cipher mode is CFB. 0x9 (F8): Block cipher mode is f8. 0xa (XTS): Block cipher mode is XTS.
3:0	CIPH_ALG	密码算法选择 0x0 (NULL): NULL Algorithm is selected. 0x1 (DES): DES Algorithm is selected. 0x2 (AES): AES Algorithm is selected. 0x3 (RC4): RC4 Algorithm is selected. 0x4 (MULTI2): MULTI2 Algorithm is selected. 0x5 (KASUMI): KASUMI Algorithm is selected. 0x6 (SNOW_3G_UEA2): SNOW-3G-UEA2 Algorithm is selected. 0x7 (ZUC_128_EE3): ZUC 128-EE3 Algorithm is selected. 0x8 (CHACHA20): ChaCha20 Algorithm is selected. 0x9 (SM4): SM4 Algorithm is selected.

31.2.5.17 FIFO 读寄存器(STAT_POP)

地址偏移：0x50

复位值：0x00000000

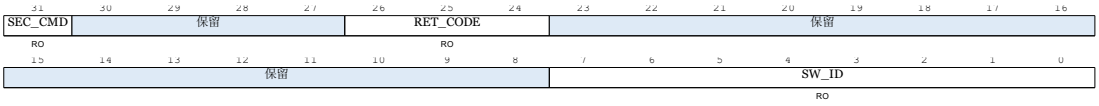


位/位域	名称	描述
31:1	保留	必须保持复位值
0	POP	往此寄存器写任意值，将会从状态 FIFO 中读取一个新的结果。在 FIFO_STAT.STAT_EMPTY 有效时，不允许写此寄存器

31.2.5.18 状态寄存器(STATUS)

地址偏移：0x54

复位值：0x00000000

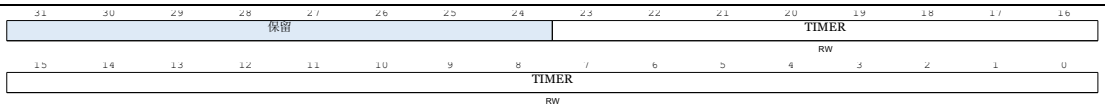


位/位域	名称	描述
31	SEC_CMD	用于指示命令被执行的模式 0x0 (NORMAL): 正常模式 0x0 (SECURE): 安全模式
30:27	保留	必须保持复位值
26:24	RET_CODE	操作结果指示 0x0 (OK): Result = OK. 0x1 (ICV_FAIL): Result = ICV Fail. 0x2 (MEM_ERROR): Result = Memory Error. 0x3 (BLK_ERROR): Result = Block Error. 0x4 (SEC_ERROR): Result = Security Error.
23:8	保留	必须保持复位值
7:0	SW_ID	软件 job ID 值

31.2.5.19 看门狗状态寄存器(STAT_WD_CTRL)

地址偏移：0x80

复位值：0x00262144

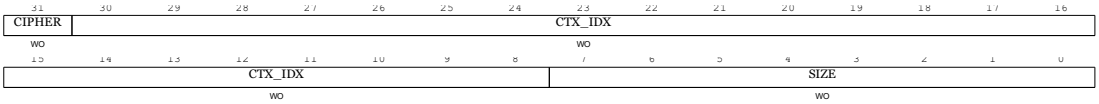


位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	TIMER	用于设置状态看门狗计算器的时钟周期数，最小值为1。

31.2.5.20 密钥寄存器(KEY_SZ)

地址偏移：0x100

复位值：0x00000000



位/位域	名称	描述
31	CIPHER	用于指示密钥是密码密钥还是哈希密钥 0x0 (HASH): 哈希密钥 0x1 (CIPHER): 密码密钥
30:8	CTX_IDX	用于设置 Context 的页码值
7:0	SIZE	用于设置密钥的字节长度

31.3 非对称加密算法加速引擎 （PKA）

31.3.1 简介

非对称加密需要对非常大的数字（从 160 到 4096 位或更多）做复杂的数学运算。PKA 是一个专用于 RSA 和 ECC（椭圆曲线加密）算法的高计算能力的协处理器。CPU 只需加载操作数，选择算法，然后开始操作。CPU 可以通过轮询或者中断的方式来获取执行结果。

因为 RSA 和 ECC 算法的密钥宽度和椭圆曲线的种类非常多，如果纯硬件方式实现，将需要非常大的电路面积。PKA 以类似哈佛结构的处理器方式来构建。PKA 包含一个程序存储器、一个指令执行单元 （IEU）、一个大位宽 ALU、几个操作数寄存器组，以及控制和状态接口。

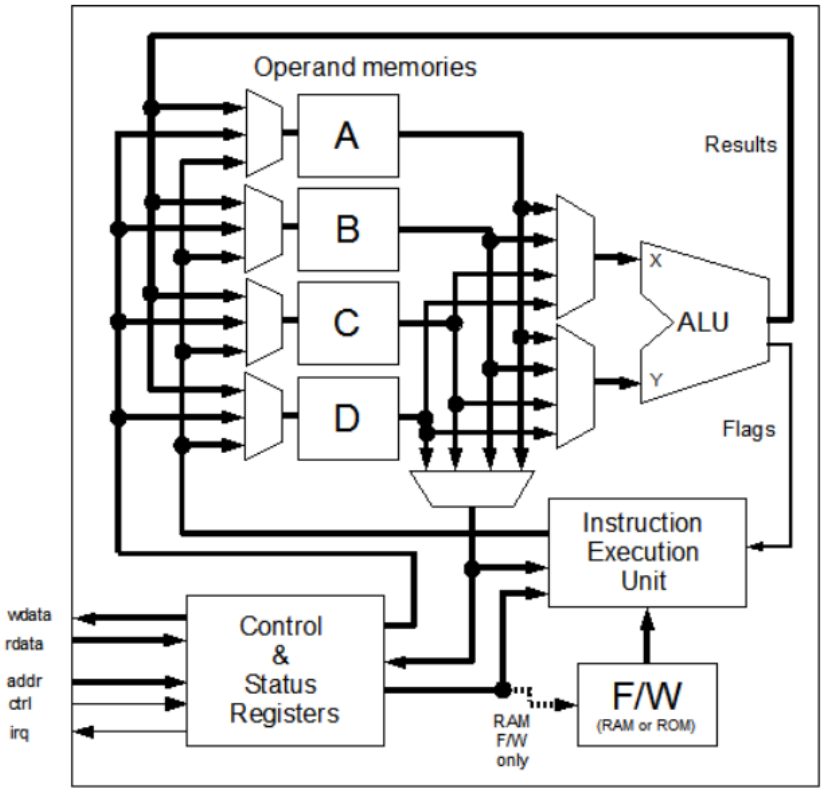


图 31.7 PKA 整体框架图

31.3.2 特性

PKA 指令集包含了 RSA 和 ECC 算法所需的基本算术操作，如 PMULT（Elliptic Curve Point Multiplication），MODMULT（Modular Multiplication），CALC_MP（Montgomery m' Precalculation）等。基于这些数学运算功能，PKA 可以完成如下的非对称加密算法：

支持的密钥位宽

- RSA (with or without CRT): 256, 512, 768, 1024, 1536 and 2048-bit
- ECC-GF(p): 160, 192, 224, 256, 320, 384, 512 and 521-bit

支持的 ECC 椭圆曲线

- 160-bit: brainpool160r1, brainpool160t1, secp160r1, secp160r2, secp160k1, wtls_9_160
- 192-bit: brainpool192r1, brainpool192t1, secp192r1, secp192k1, fips186p192

- 224-bit: brainpool224r1, brainpool224k1, secp224r1, secp224k1, fips186p224
- 255-bit: curve25519, ed25519
- 256-bit: brainpool256r1, brainpool256t1, secp256r1, secp256k1, fips186p256, ansix9p256r1, sm2
- 320-bit: brainpool320r1, brainpool320t1
- 384-bit: brainpool384r1, brainpool384t1, secp384r1, fips186p384
- 512-bit: brainpool512t1, brainpool512r1
- 521-bit: secp521r1

Note

除了以上列出的椭圆曲线，其他曲线如果满足下面 1，2 点，则也能被支持。

1. (y2 = x3 + ax + b mod p) 。
2. 位宽符合{160, 192, 224, 256, 320, 512}。

31.3.3 操作描述

31.3.3.1 存储映射

PKA 模块的主机软件接口被映射到如下的存储区域。该区域包括状态寄存器和控制寄存器，操作数寄存器组，F/W 区域（存储指令）。需要说明的是操作数寄存器组实际电路为 SRAM 电路，而非寄存器。

表 31.2 PKA 整体存储地址映射图

区域	起始地址	长度（字节）
控制/状态寄存器	0x0	384
操作数内存区域 A	0x400	2048
操作数内存区域 B	0x800	1024
操作数内存区域 C	0xC00	1024
操作数内存区域 D	0x1000	1024
F/W 区域	0x4000	8192

主机软件不允许也不需要进行部分字节访问，主机端必须以 32 位的形式对 PKA 进行读写操作。

当 PKA 引擎正在执行指令时，主机对操作数区域的访问被锁定以防止无意中更改了操作数值。但是对状态和控制寄存器的读操作任何时候都是被允许。

31.3.3.2 操作数寄存器映射

操作数存储区域基于的不同位宽，被动态的分割为一系列操作数寄存器组。操作数寄存器组分为 A,B,C,D 组。每组内又分为标号 0 到 7 的 8 个寄存器。当主机端选择不同密钥宽度的算法时，操作数寄存器需要根据密钥位宽来重新排列。这里重新排列的意思是逻辑上重新排列，下表中的分布图，详细的列出了如何根据密钥位宽来动态调整寄存器组的逻辑排列，并确定排列后的操作数寄存器地址。

表 31.3 操作数地址映射图

操作数寄存器	ECC 操作数地址			RSA 操作数地址		
	256-bit	512-bit	1024-bit	512-bit	1024-bit	2048-bit
A0	0x400	0x400	0x400	0x400	0x400	0x400
A1	0x420	0x440	0x480	0x440	0x480	0x500
A2	0x440	0x480	0x500	–	–	–
A3	0x460	0x4C0	0x580	–	–	–
A4	0x480	0x500	0x600	–	–	–
A5	0x4A0	0x540	0x680	–	–	–
A6	0x4C0	0x580	0x700	–	–	–
A7	0x4E0	0x5C0	0x780	–	–	–
B0	0x800	0x800	0x800	0x800	0x800	0x800
B1	0x820	0x840	0x880	0x840	0x880	0xA00
B2	0x840	0x880	0x900	–	–	–
B3	0x860	0x8C0	0x980	–	–	–
B4	0x880	0x900	0xA00	–	–	–
B5	0x9A0	0x940	0xA80	–	–	–
B6	0x9C0	0x980	0xB00	–	–	–
B7	0x9E0	0x9C0	0xB80	–	–	–
C0	0xC00	0xC00	0xC00	0xC00	0xC00	0xC00
C1	0xC20	0xC40	0xC80	0xC40	0xC80	0xD00

操作数寄存器	ECC 操作数地址			RSA 操作数地址		
	256-bit	512-bit	1024-bit	512-bit	1024-bit	2048-bit
C2	0xC40	0xC80	0xD00	–	–	–
C3	0xC60	0xCC0	0xD80	–	–	–
C4	0xC80	0xD00	0xE00	–	–	–
C5	0xCA0	0xD40	0xE80	–	–	–
C6	0xCC0	0xD80	0xF00	–	–	–
C7	0xCE0	0xDC0	0xF80	–	–	–
D0	0x1000	0x1000	0x1000	0x1000	0x1000	0x1000
D1	0x1020	0x1040	0x1080	0x1040	0x1080	0x1100
D2	0x1040	0x1080	0x1100	0x1080	0x1100	0x1200
D3	0x1060	0x10C0	0x1180	0x10C0	0x1180	0x1300
D4	0x1080	0x1100	0x1200	–	–	–
D5	0x10A0	0x1140	0x1280	–	–	–
D6	0x10C0	0x1180	0x1300	–	–	–
D7	0x10E0	0x11C0	0x1380	–	–	–

Note

注：以上表格中只列出了位宽是 2 的 N 次方的情况。如果不是 2 的 N 次方的位宽，则用下一个能满足 2 的 N 次方的位宽的排列。 比如 160-bit 和 224-bit 被映射到上面表格中的 256-bit。384-bit 被映射到 512-bit。521-bit 被映射到 1024-bit。

31.3.3.3 操作数小端排列

PKA 的操作数都是大于 32 位，且默认为小端模式，所以往 PKA 写操作数默认是以小端排列的 4 字节为单位的数据串形式，如下图例子，一个 32 个 Word 的操作数，在存储器中的排列。

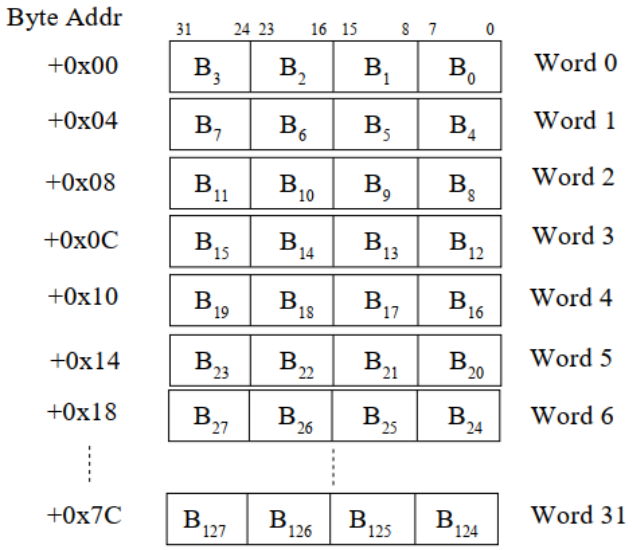


图 31.8 操作数小端排列定义图

31.3.3.4 双精度模式

PKA 的操作数支持单精度或者双精度模式。双精度操作数需 2 个单精度操作数寄存器来拼接。访问地址只会是上表中偶数标号的地址，比如 A0,A2,A4,A6,B2,D4 等。对于双精度模式的 A0 寄存器实际覆盖了 A0,A1 两个单精度操作数寄存器，其他也类似。

31.3.4 操作举例

31.3.4.1 功能入口

下表列出了 PKA 模块支持的数学运算操作，将功能入口地址写到 ENTRY_PNT 寄存器，可以控制 PKA 做相应运算。

表 31.4 支持的运算操作及入口地址图

名字	功能入口地址	描述
bit_serial_mod	0x14	Calculate y/x mod m
bit_serial_mod_dp	0x13	双精度 Calculate y/x mod m
calc_mp	0x10	Montgomery m’ Precalculation
calc_r_inv	0x11	Calculate Montgomery precomputation values: r_inv = 1/r mod m

名字	功能入口地址	描述
		mp = ((r * r-1)-1)/m r_sqr = (r * r) mod m
calc_r_sqr	0x12	Montgomery r2 mod m Precalculation
crt	0x18	Calculate CRT
crt_key_setup	0x17	Calculate CRT key setup coefficients
modadd	0xb	Calculate x + y mod m
moddiv	0xd	Calculate y/x mod m
modexp	0x16	Modular Exponentiation
modinv	0xe	Calculate 1/x mod m
modmult	0xa	Calculate x * y mod m
modmult_521	0x29	Calculate x * y mod m mod p, where p = 2^521-1
modsub	0xc	Calculate x - y mod m
mult	0x15	Calculate a * b
m_521_montmult	0x28	Calculate x * y mod m, where p is an arbitrary modulus (usually the order-of-the-base-point)
padd	0x1c	Calculate R = P + Q
padd_521	0x26	Calculate R = P + Q mod p, where p = 2^521-1
pdbl	0x1a	Calculate Q = 2P
pdbl_521	0x25	Calculate Q = 2P mod p, where p = 2^521-1
pmult	0x19	Calculate Q = kP
pmult_521	0x24	Calculate Q = kP mod p, where p = 2^521-1
pver	0x1e	Calculate y2 == x3 + ax + b mod p
pver_521	0x27	Calculate y2 == x3 + ax + b mod p, where p = 2^521-1

名字	功能入口地址	描述
reduce	0xf	Calculate x mod m
shamir	0x23	Calculate R = kP + lQ
shamir_521	0x2a	Calculate R = kP + lQ

31.3.4.2 运行 192-bit EC Point Multiply 示例

本示例以小端模式呈现，演示如何运行一个椭圆曲线点乘算法。

Note

注：更多细节可以参考 PKA 的固件手册《DWC_pka_firmware_user》。

椭圆曲线采用 FIPS-186p192，曲线参数如下：

Gx = 0x188DA80E_B03090F6_7CBF20EB_43A18800_F4FF0AFD_82FF101212

Gy = 0x07192B95_FFC8DA78_631011ED_6B24CDD5_73F977A1_1E794811

a = 0xFFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFE_FFFFFFFF_FFFFFFFC

b = 0x64210519_E59C80E7_0FA7E9AB_72243049_FEB8DEEC_C146B9B1

p = 0xFFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFE_FFFFFFFF_FFFFFFFF

p’ = 0x00000000_FFFFFFFF_FFFFFFFF_00000000_00000001

r2 mod p = 0x00000001_00000000_00000002_00000000_00000001

o = 0xFFFFFFFF_FFFFFFFF_FFFFFFFF_99DEF836_146BC9B1_B4D22831

k = 0xFFFFFFFF_FFFFFFFF_FFFFFFFF_99DEF836_146BBC5D_21A05277

算法输入输出的寄存器映射如下表：

表 31.5 寄存器映射举例图

参数	方向	操作寄存器	描述
Px	Input	A2	Point affine x-coordinate, typically base-point Gx but may be any arbitrary point on the curve
Py	Input	B2	Point affine y-coordinate, typically base-point Gy but may be any arbitrary point on the curve
a	Input	A6	Curve parameter, a
k	Input	D7	Key (2 < key < order of base point of curve)
p	Input	D0	Modulus, p
p’	Input	D1	Montgomery precomputed value, p’
r2 mod p	Input	D3	Montgomery precomputed value, r2 mod p
Qx	Output	A2	Result affine x-coordinate
Qy	Output	B2	Result affine y-coordinate

使用 256-bit 寄存器映射将数据写入正确的内存地址位置（192-bit 需要提到 256-bit 的操作数寄存器地址映射）。

1. 通过写 CONFIG 寄存器，确保 PKA 处于小端模式：
   - 0x00000000 => *(+0x001C)
2. 将 Px 写到 A2：
   - 0x82FF1012 => *(+0x0440)
   - 0xF4FF0AFD => *(+0x0444)
   - 0x43A18800 => *(+0x0448)
   - 0x7CBF20EB => *(+0x044C)
   - 0xB03090F6 => *(+0x0450)
   - 0x188DA80E => *(+0x0454)
3. 将 Py 写到 B2:
   - 0x1E794811 => *(+0x0840)

- 0x73F977A1 => *(+0x0844)
- 0x6B24CDD5 => *(+0x0848)
- 0x631011ED => *(+0x084C)
- 0xFFC8DA78 => *(+0x0850)
- 0x07192B95 => *(+0x0854)

4. 将 a 写到 A6:
   - 0xFFFFFFFC => *(+0x04C0)
   - 0xFFFFFFFF => *(+0x04C4)
   - 0xFFFFFFFE => *(+0x04C8)
   - 0xFFFFFFFF => *(+0x04CC)
   - 0xFFFFFFFF => *(+0x04D0)
   - 0xFFFFFFFF => *(+0x04D4)
5. 将 k 写到 D7。 对于 PMult 操作, 192 bit 的 key 被要求用 0 填充扩展到 256 bit。所以需要再填充 64 bit 的 0 到 D7。
   - 0x21A05277 => *(+0x10E0)
   - 0x146BBC5D => *(+0x10E4)
   - 0x99DEF836 => *(+0x10E8)
   - 0xFFFFFFFF => *(+0x10EC)
   - 0xFFFFFFFF => *(+0x10F0)
   - 0xFFFFFFFF => *(+0x10F4)
   - 0x00000000 => *(+0x10F8)
   - 0x00000000 => *(+0x10FC)
6. 将 p 写到 D0:
   - 0xFFFFFFFF => *(+0x1000)
   - 0xFFFFFFFF => *(+0x1004)
   - 0xFFFFFFFE => *(+0x1008)
   - 0xFFFFFFFF => *(+0x100C)
   - 0xFFFFFFFF => *(+0x1010)
   - 0xFFFFFFFF => *(+0x1014)

7. 将 p’写到 D1。需要注意的是，在这个例子中，p’只有 128 bit，需要用 0 扩展到 192 bit。
   - 0x00000001 => *(+0x1020)
   - 0x00000000 => *(+0x1024)
   - 0xFFFFFFFF => *(+0x1028)
   - 0xFFFFFFFF => *(+0x102C)
   - 0x00000000 => *(+0x1030)
   - 0x00000000 => *(+0x1034)
8. 将“r2 mod p” 写 D3 (也需要做 0 扩展):
   - 0x00000001 => *(+0x1060)
   - 0x00000000 => *(+0x1064)
   - 0x00000002 => *(+0x1068)
   - 0x00000000 => *(+0x106C)
   - 0x00000001 => *(+0x1070)
   - 0x00000000 => *(+0x1074)
9. 清除所有残留标志:
   - 0x00000000 => *(+0x0024)
10. 清除 F-Stack:
    - 0x00000000 => *(+0x0010)
11. 写功能入口地址:
    - 0x00000020 => *(+0x0004)
12. 禁掉中断，采用轮询模式：
    - 0x00000000 => *(+0x0040)
13. PKA 开始运算：
    - 0x80000206 => *(+0x0000)
14. 通过 STAT.DONE 查看 PKA 运算是否完成：
    - while((*(+0x0020) & 0x40000000) == 0)
15. 清除运算完成标志位 STAT.DONE : 0x40000000 => *(+0x0020)
16. 查看 RTN_CODE.STOP_REASON 是否等于 0，确认运算过程是否有出现错误:

- if((*(+0x0008) & 0x00FF0000) != 0)

17. 从 A2 读取运算结果 Qx：
    - *(+0x0440) => 0xB8640A98
    - *(+0x0444) => 0x28AA931C
    - *(+0x0448) => 0xD6388B5B
    - *(+0x044C) => 0xBB055E5C
    - *(+0x0450) => 0x37E5B499
    - *(+0x0454) => 0x95D70F44
18. 从 B2 读取运算结果 Qy：
    - *(+0x0840) => 0x987DFCB6
    - *(+0x0844) => 0xB4F21D38
    - *(+0x0848) => 0x4773EC2A
    - *(+0x084C) => 0xC361B7A6
    - *(+0x0850) => 0x5B10026A
    - *(+0x0854) => 0xDBAF49EF

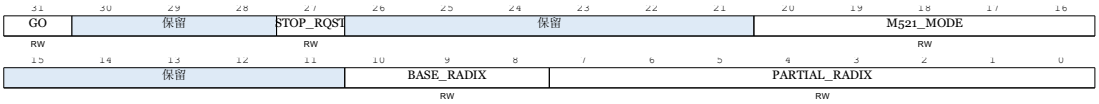
31.3.5 寄存器

PKA 基地址：0x41140000

31.3.5.1 主控制寄存器(CTRL)

地址偏移：0x0

复位值：0x00000000



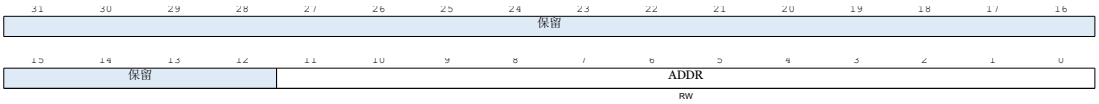
位/位域	名称	描述
31	GO	启动对当前数据和命令集进行操作的 PKA。 PKA 将开始执行来自 ENTRY_PNT.ADDR 字段中指示的固件地址的指令。
30:28	保留	必须保持复位值
27	STOP_RQST	请求 PKA 在下一个指令边界处停止

位/位域	名称	描述
26:21	保留	必须保持复位值
20:16	M521_MODE	Mersenne M-521 模式设置。 将 M-521 模式的字段设置为 9。对于所有其他操作大小，设置为 0！ 此字段仅存在于特定配置中。
15:11	保留	必须保持复位值
10:8	BASE_RADIX	设置基本操作数大小 （radix） 的大小。 此字段的最大大小由实际参数化硬件配置决定。 2 = 256-bit 3 = 512-bit 4 = 1024-bit 5 = 2048-bit 6 = 4096-bit others = reserved
7:0	PARTIAL_RADIX	从基数中选择部分字段。 这些位作为一种掩码运行，允许执行完整基数的子部分。 具有 32 位内存字的实例的示例；如果 CTRL_BASE_RADIX 位设置为 2 以表示 256 位操作数大小，PARTIAL_RADIX 位设置为 5，则将处理 8 位操作数中的 5 个字。实际上，将操作数基数限制为 5/8 * 256 或 160 位。 PARTIAL_RADIX 位是该 register 中唯一对配置的 ALU 宽度敏感的位。 ALU 宽度为 32 位，将此字段设置为 17 以启用 Mersenne M-521 模式 （ceiling（521/32） = 17）。

31.3.5.2 入口指针寄存器(ENTRY_PNT)

地址偏移：0x4

复位值：0x00000000

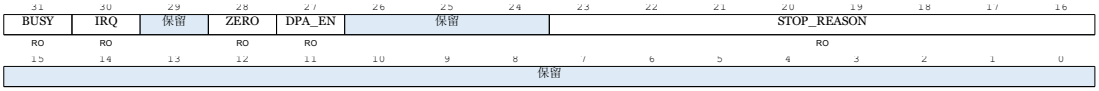


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	ADDR	指示要执行的函数的启动指令。 此 register 只能在 PKA 停止时写入。 PKA 将忽略任何其他时间的写入。 此 register 将在 PKA 执行时更新，并在 PKA 停止时指向要获取的下一条指令。这可以与 CTRL.STOP_RQST 位或看门狗寄存器以调试模式运行 PKA。 地址是相对于固件内存开头的字索引。

31.3.5.3 返回代码寄存器(RTN_CODE)

地址偏移：0x8

复位值：0x00000000



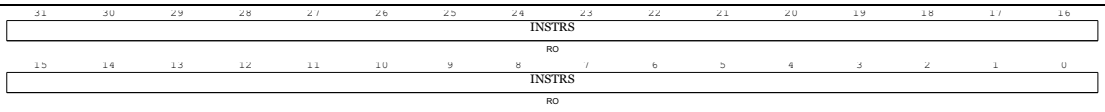
位/位域	名称	描述
31	BUSY	指示 PKA 正在主动执行程序
30	IRQ	指示 PKA 何时完成其任务。它是 STAT.IRQ 的 R/O 反映。
29	保留	必须保持复位值
28	ZERO	指示 sequencer 中 zero 标志的状态。
27	DPA_EN	表示内核能够在 TA 和/或 DPA 模式下运行。
26:24	保留	必须保持复位值

位/位域	名称	描述
23:16	STOP_REASON	指示 PKA 停止的原因。 如果 PKA 因正常的 HALT 指令而停止，则将显示为 0。 任何其他原因都将收到非 0 值。 在 PKA 完成执行程序后，此字段将是静态且稳定的，并将保持此状态，直到 PKA 再次重新启动。 如果 PKA 已完成 STOP_REASON 非 0 的操作，则表示发生了异常停止情况。在异常停止条件下，在开始任何新操作之前清除 STACK_PNTR 寄存器非常重要。此外，如果存在任何可选的 bankswitch 寄存器，则在异常停止条件后也必须将其设置为 0。 注意：在加载任何新数据之前，必须完成清除所有现有 bankswitch 寄存器！ 0 = 正常停止 1 = 无效的操作码 2 = 堆栈下溢 3 = 堆栈溢出 4 = 看门狗 5 = 主机请求 （CTRL.STOP_RQST） 6 = 保留 7 = 保留 8 = 内存端口冲突 9 = 保留 others = 保留用于固件
15:0	保留	必须保持复位值

31.3.5.4 指令执行数量寄存器(INSTR_SINCE_GO)

地址偏移：0x14

复位值：0x00000000

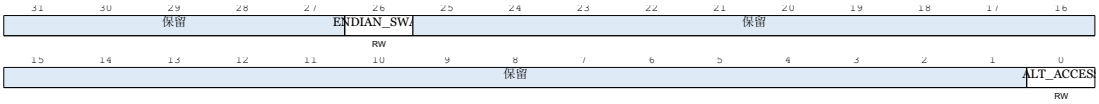


位/位域	名称	描述
31:0	INSTRS	从 CTRL.GO 有效后指令执行的条数

31.3.5.5 配置寄存器(CONFIG)

地址偏移：0x1c

复位值：0x00000000

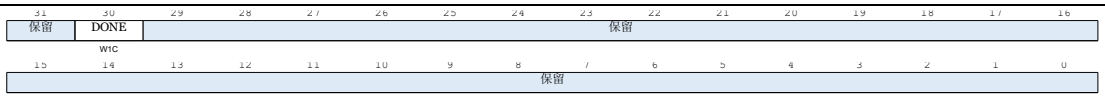


位/位域	名称	描述
31:27	保留	必须保持复位值
26	ENDIAN_SWAP	设置数据存储器区域在传入或传出内核时进行字节序交换。 0 = 无字节通道交换（小端） 1 = 字节通道交换（大端）
25:1	保留	必须保持复位值
0	ALT_ACCESS	用于防止 synthesizer 删除原本是 W/O 内存的内容。 此 register bit 永远不会在正常操作模式下设置，也不打算由最终用户写入。 这个 bit 仅存在于可选的 DPA-hardened-configurations 中，主要用于 synthesis 目的。 仅供试用！ 0 = 正常 C 内存访问 1 = 将 C 内存读/写访问插槽切换到 DPA 影子内存。

31.3.5.6 中断状态寄存器(STAT)

地址偏移：0x20

复位值：0x00000000

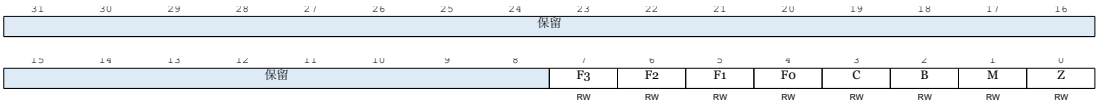


位/位域	名称	描述
31	保留	必须保持复位值
30	DONE	W1 = 确认现有中断 W0 = NOP R1 = 存在未确认的中断 R0 = 不存在未确认的中断
29:0	保留	必须保持复位值

31.3.5.7 PKA 标志寄存器(FLAGS)

地址偏移：0x24

复位值：0x00000000

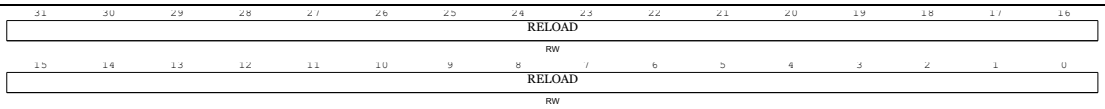


位/位域	名称	描述
31:8	保留	必须保持复位值
7	F3	用户标志 3
6	F2	用户标志 2
5	F1	用户标志 1
4	F0	用户标记 0
3	C	Carry 标志
2	B	Borrow 标志
1	M	内存测试标志
0	Z	零标志

31.3.5.8 看门狗寄存器(WATCHDOG)

地址偏移：0x28

复位值：0x00000000

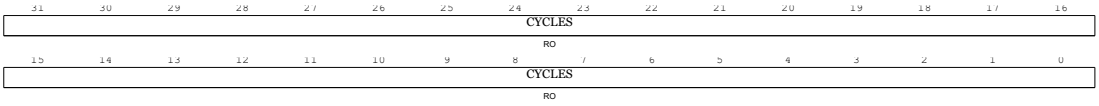


位/位域	名称	描述
31:0	RELOAD	看门狗重新加载值

31.3.5.9 时钟周期数量寄存器(CYCLES_SINCE_GO)

地址偏移：0x2c

复位值：0x00000000

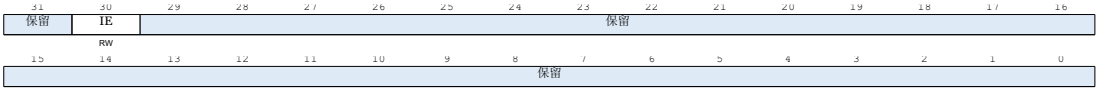


位/位域	名称	描述
31:0	CYCLES	自 CTRL.GO 以来的时钟周期

31.3.5.10 中断使能寄存器(IRQ_EN)

地址偏移：0x40

复位值：0x00000000



位/位域	名称	描述
31	保留	必须保持复位值
30	IE	0 = 在 O_irq 时禁用中断信号 1 = 在 O_irq 启用 STAT.IRQ
29:0	保留	必须保持复位值

31.4 真随机数发生 （TRNG）

31.4.1 简介

TRNG 用于生成统计上均匀分布的真随机数。该电路包括符合 NIST SP800-90B 标准的噪声源、NIST SP800-90B 标准的调节组件和 NIST SP800-90A 认证的确定性随机位发生器 （DRBG）。TRNG 也兼容德国 BSI AIS 20/31 标准。

31.4.2 主要特性

- 支持内部产生随机种子
- 支持由外部 HOST 导入随机种子
- 支持 128 位随机数生成
- 支持内置连续基本运行状况测试
- 符合 NIST SP800-90A/B/c 和 BSI AIS 20/31 标准。
- 使用块密码 AES-CBC 对熵源调节
- 6 个独立环形振荡器的比特发生器模块

31.4.3 功能描述

TRNG 的总体架构图如下图所示，包含：

- 实时熵源
- 伪随机数发生器
- 寄存器模块，包含控制，状态，数据寄存器
- 健康测试块

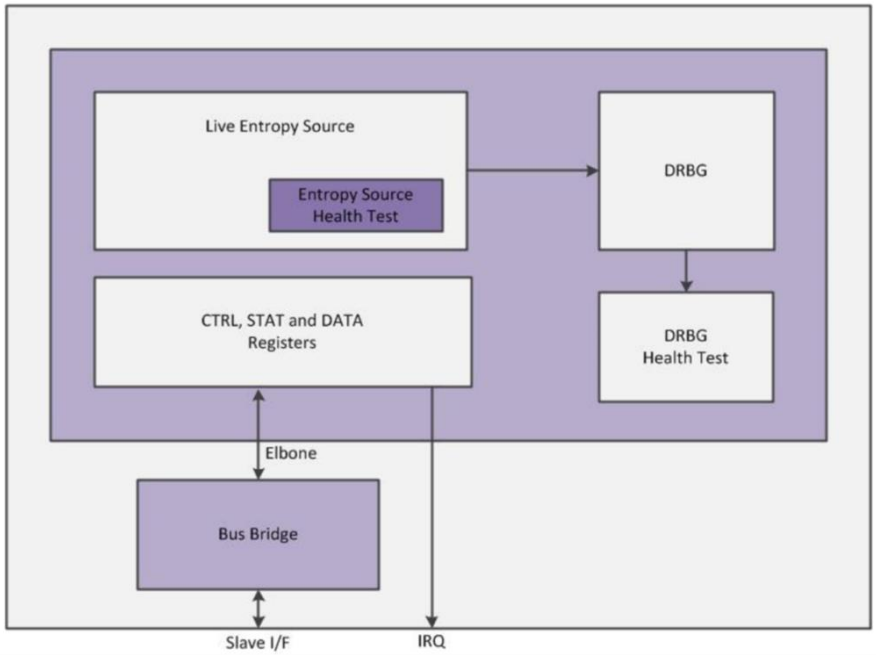


图 31.9 TRNG 整体框架图

31.4.3.1 实时熵源

实时熵源用于为 DRBG 生成种子。熵源符合 NIST SP800-90B 和 BSI AIS 20/31 标准。它生成的随机数在统计上等价于统一分布式随机数据流。实时熵源包含:

- **噪声源：**基于环形振荡器的种子发生器用作 TRNG 的噪声源。噪声源为熵源中的推导函数提供非确定性随机位。随机种子发生器由多个环形振荡器对组成，每个环形振荡器对驱动一个采样寄存器。这套逻辑电路（环形振荡器、采样寄存器和测量）统称为随机位发生器 （BG）。 噪声源包含 6 个独立的 BG。每个 BG 产生随机时间间隔的 0，1 系列值。
- **调节组件：**熵源使用分组密码推导函数用于对噪声源的输出进行后处理。可减少数据的偏差并增加熵。调节组件主要模块是 AES core，它是 NIST SP800-90B 标准批准的分组密码算法之一。该模块采用 512 位随机流并收集 128 位真随机数流进入 SEEDx 寄存器。

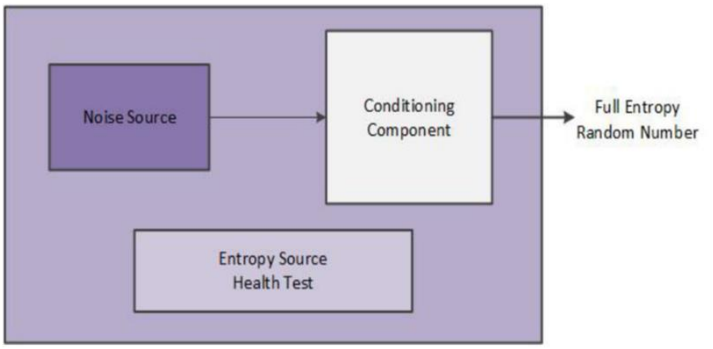


图 31.10 熵源结构图

31.4.3.2 DRBG

DRBG 是一款符合 NIST SP800-90A 标准的伪随机数发生器。它产生随机序列需要一个初始值（也就是种子）。种子来源于实时熵源或主机。用户程序应定期进行重新播种，以保证种子的新鲜。

31.4.3.3 健康测试

- **熵源健康测试：**TRNG 能够对实时熵源执行满足 NIST SP800-90B 和 AIS 20/31 标准的健康测试。 TRNG 复位后会自动对调节组件进行一轮 KAT 测试和对噪声源进行一轮完整的统计测试。健康测试模块接收来自噪

声源的比特流，并进行连续的统计性测试。对主机提供的任何随机数都应用相同的测试。连续性健康测试符合 NIST SP800-90B 标准，分别是：

- Repetition Count
- Adaptive Proportion

Note

使用以上两项测试足以满足 BSI AIS20/31 标准。

- **DRBG 健康测试：**TRNG 在启动时对 DRBG 执行 KAT(Known Answer TestS)测试。测试向量为 NIST SP800-90A 标准提供的 Vector0, Vector1,Vector2。
- **错误处理选项：**当健康测试失败时，无论是统计测试还是 KAT，无论是在启动期间，还是连续统计检查或按需测试时，会激发不可屏蔽的警报，TRNG 内部状态自行归零。原因可以从 alarm 寄存器中读取故障类型。在启动测试进行时，TRNG 处于 busy 状态，此时不可被使用。如果在启动时测试失败，TRNG 会发出警报并自行归零。但是，默认情况下，TRNG 会从 busy 状态中跳出，并随时可以被使用。所以应用程序需要负责在 TRNG 自测失败时采取适当的措施。建议应用程序重置 TRNG 以启动新一轮启动测试。

31.4.4 TRNG 操作流程

下图是 TRNG 允许的操作状态机。

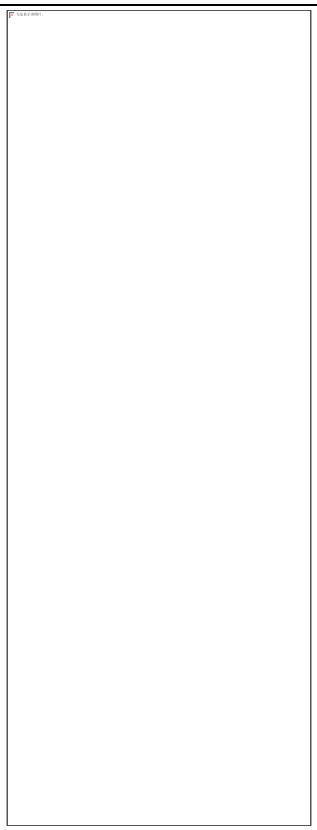


图 31.11

- 种子生成操作

确保 SMODE.NONCE == 0

当使用轮询模式：

1. 往 CTRL.CMD 写 1，TRNG 开始种子生成操作。
2. 轮询读取 ISTAT.DONE 位，直到 ISTAT.DONE 等于 1。
3. 写 1 到 ISTAT.DONE，代表 HOST 已响应，并清除 DONE 标志位。

当使用中断模式:

1. 往 IE.GLBL 和 IE.DONE 分别写 1，使能中断。
2. 往 CTRL.CMD 写 1，TRNG 开始种子生成操作。
3. 等待中断发生。
4. 在 HOST 的应用程序中的中断处理函数里，往 ISTAT.DONE 写 1。

- 创建新 State

当使用轮询模式：

1. 往 CTRL.CMD 写 3 。
2. 轮询读取 ISTAT.DONE 位，直到 ISTAT.DONE 等于 1。
3. 写 1 到 ISTAT.DONE，代表 HOST 已响应，并清除 DONE 标志位。

当使用中断模式：

1. 往 IE.GLBL 和 IE.DONE 分别写 1，使能中断。
2. 往 CTRL.CMD 写 3。
3. 等待中断发生。
4. 在 HOST 的应用程序中的中断处理函数里，往 ISTAT.DONE 写 1。

- 随机数生成

当使用轮询模式：

1. 往 CTRL.CMD 写 6，TRNG 开始随机数产生操作。
2. 轮询读取 ISTAT.DONE 位，直到 ISTAT.DONE 等于 1。
3. 写 1 到 ISTAT.DONE，代表 HOST 已响应，并清除 DONE 标志位。
4. 从 RANx 寄存器读取 128bit 的随机数。
5. 重复 1 到 4 步骤，获取所需位宽的随机数。

当使用中断模式：

1. 往 IE.GLBL 和 IE.DONE 分别写 1，使能中断。
2. 往 CTRL.CMD 写 6，TRNG 开始随机数生成操作。
3. 等待中断发生。
4. 在 HOST 的应用程序中的中断处理函数里，往 ISTAT.DONE 写 1。
5. 从 RANx 寄存器读取 128bit 的随机数
6. 重复 1 到 4 步骤，获取所需位宽的随机数。

- 归零操作

当使用轮询模式：

1. 往 CTRL.CMD 写 15 。
2. 轮询读取 ISTAT.ZEROIZED 位，直到 ISTAT.ZEROIZED 等于 1。
3. 写 1 到 ISTAT.ZEROIZED，代表 HOST 已响应，并清除 DONE 标志位。

当使用中断模式：

1. 往 IE.GLBL 和 IE.DONE 分别写 1，使能中断。
2. 往 CTRL.CMD 写 15。
3. 等待中断发生。
4. 在 HOST 的应用程序中的中断处理函数里，往 ISTAT.ZEROIZED 写 1。

31.4.5 处理时间

表 31.6 处理时间表

操作	处理时钟周期
创建新 State	60
随机数生成	24

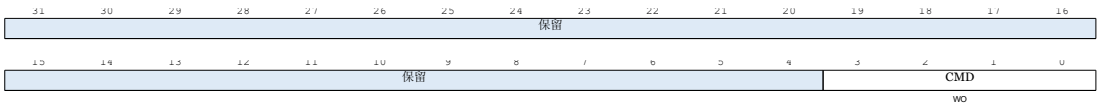
31.4.6 寄存器

TRNG 基地址：0x41150000

31.4.6.1 控制寄存器(CTRL)

地址偏移：0x0

复位值：0x00000000



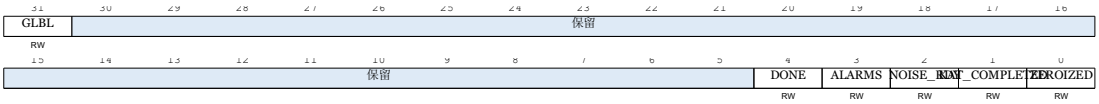
位/位域	名称	描述
31:4	保留	必须保持复位值
3:0	CMD	执行一条命令。 未列出的枚举值为 'reserved'。 0x0 （NOP）：执行 NOP 0x1 （GEN_NOISE）：从噪声生成全熵种子 0x2 （GEN_NONCE）：从主机写入的随机数生成种子 0x3 （CREATE_STATE）：移动 DRBG 以创建状态 0x4 （RENEW_STATE）：将 DRBG 移动到更新状态

位/位域	名称	描述
		0x5 （REFRESH_ADDIN）：将 DRBG 移动到刷新插件 0x6 （GEN_RANDOM）：生成一个随机数 0x7 （ADVANCE_STATE）：高级 DRBG 状态 0x8 （RUN_KAT）：在 DRBG 或熵源上运行 KAT 0xf （ZEROIZE）：归零

31.4.6.2 中断使能寄存器(IE)

地址偏移：0x10

复位值：0x00000000



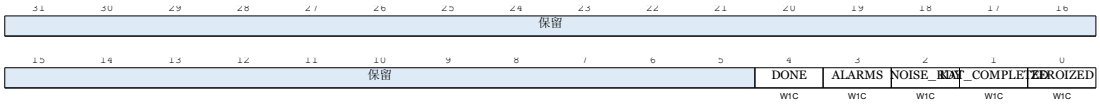
位/位域	名称	描述
31	GLBL	TRNG 的全局中断使能信号 0x0 （GLBL_DIS）：禁用 GLBL 中断 0x1 （GLBL_EN）：启用 GLBL 中断
30:5	保留	必须保持复位值
4	DONE	包括或排除 DONE 中断 0x0 （DONE_DIS）：禁用 DONE 中断 0x1 （DONE_EN）：启用 DONE 中断
3	ALARMS	包括或排除 ALARMS 中断 0x0 （ALARMS_DIS）：禁用 ALARMS 中断 0x1 （ALARMS_EN）：启用 ALARMS 中断
2	NOISE_RDY	包括或排除 NOISE_RDY 中断 0x0 （NOISE_RDY_DIS）：禁用 NOISE_RDY 中断 0x1 （NOISE_RDY_EN）：启用 NOISE_RDY 中断

位/位域	名称	描述
1	KAT_COMPLETED	包括或排除 KAT_COMPLETED 中断 0x0（KAT_COMPLETED_DIS）：禁用 KAT_COMPLETED 中断 0x1（KAT_COMPLETED_EN）：启用 KAT_COMPLETED 中断
0	ZEROIZED	包括或排除 ZEROIZED 中断 0x0（ZEROIZED_DIS）：禁用 ZEROIZED 中断 0x1（ZEROIZED_EN）：启用 ZEROIZED 中断

31.4.6.3 中断状态寄存器(ISTAT)

地址偏移：0x14

复位值：0x00000000



位/位域	名称	描述
31:5	保留	必须保持复位值
4	DONE	DONE 位允许用户轮询除 RUN_KAT 和 ZEROIZE 之外的所有命令的完成情况，它们有自己的中断。 0x0（DONE_R0）：R0：没有未确认的命令完成 0x1（DONE_R1）：R1：未确认的命令完成 0x0（DONE_W0）：W0：NOP 0x1（DONE_W1）：W1：清除 DONE 标志
3	ALARMS	ALARMS 位允许用户轮询失败。发生警报时，会自动归零。清除此中断也会清除 O_alarm 引脚。 0x0（ALARMS_R0）： R0： 无未确认的警报 0x1（ALARMS_R1）： R1： 未确认的警报 0x0（ALARMS_W0）：W0：NOP

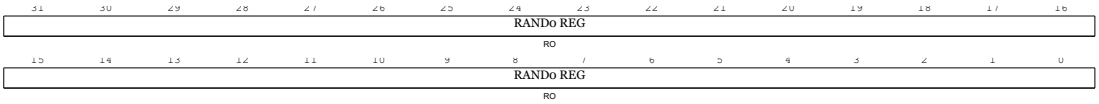
位/位域	名称	描述
		0x1 （ALARMS_W1）：W1：清除 ALARMS 标志
2	NOISE_RDY	当 DWC TRNG NIST 在自种子模式下生成全熵种子时，SMODE.MISSION_MODE 为 0 （TEST 模式） 和 SMODE.NOISE_COLLECT 设置为 1，则 NOISE_RDY 位会在生成 512 位噪声时通知用户。在 MISSION 操作模式下，此中断永远不会发生。 值： 0x0 （NOISE_RDY_R0）：R0：没有未确认的噪声生成完成 0x1 （NOISE_RDY_R1）：R1：未确认的噪声生成完成 0x0 （NOISE_RDY_W0）：W0：NOP 0x1 （NOISE_RDY_W1）：W1：清除 NOISE_RDY 标志
1	KAT_COMPLETED	指示 RUN_KAT 命令的完成。 0x0 （KAT_COMPLETED_R0）：R0：无未确认的 KAT_COMPLETED 0x1 （KAT_COMPLETED_R1）：R1：未确认的 KAT_COMPLETED 0x0 （KAT_COMPLETED_W0）：W0：NOP 0x1 （KAT_COMPLETED_W1）：W1：透明 KAT_COMPLETED 标志
0	ZEROIZED	指示 ZEROIZE 操作完成。 0x0 （ZEROIZED_R0）：R0：没有未确认的归零 0x1 （ZEROIZED_R1）：R1：未确认的 ZEROIZED 0x0 （ZEROIZED_W0）：W0：NOP

位/位域	名称	描述
		0x1 （ZEROIZED_W1）：W1：清除 ZEROIZED 标志

31.4.6.4 随机数寄存器 0(RAND0)

地址偏移：0x24

复位值：0x00000000

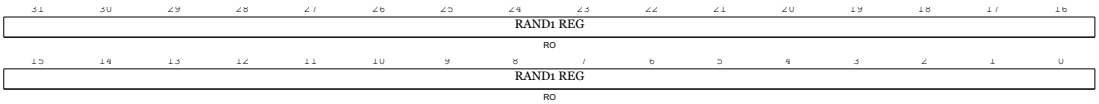


位/位域	名称	描述
31:0	RAND0 REG	随机数据字 0。

31.4.6.5 随机数寄存器 1(RAND1)

地址偏移：0x28

复位值：0x00000000

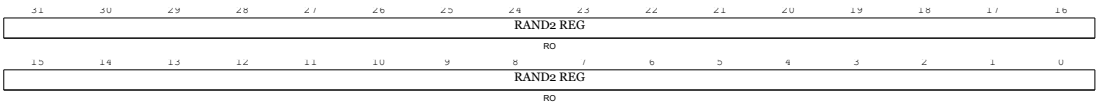


位/位域	名称	描述
31:0	RAND1 REG	随机数据字 1。

31.4.6.6 随机数寄存器 2(RAND2)

地址偏移：0x2c

复位值：0x00000000

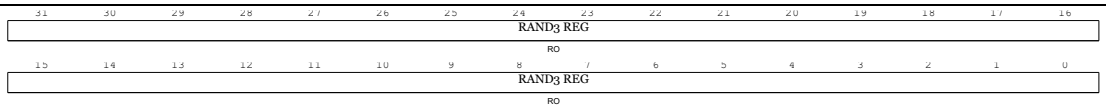


位/位域	名称	描述
31:0	RAND2 REG	随机数据字 2。

31.4.6.7 随机数寄存器 3(RAND3)

地址偏移：0x30

复位值：0x00000000



位/位域	名称	描述
31:0	RAND3 REG	随机数据字 3。

32 修订历史

版本号	修订内容	修订时间
V1.0	初始版本	2024.10
V1.1	1: 章节 1.2 的表 1.1 中，通信接口的 IIC 路数由 2 路修订为 4 路； 2: 章节 13.3.1 中，通道数量由 40 个修订为 42 个； 3: 章节 21.5.3.1 补充了 AFMR * 寄存器位 19 的相关功能描述； 4: 章节 23.1 中，每个 SPI 支持的从机数量由 8 个修订为 5 个； 5: 章节 29.2 补充了 ADC 的相关特性； 6: 章节 30.4.1 补充了数字模拟转换功能的说明； 7: 修订了章节 30.5.2 对 DAC 数据寄存器位 7:0 的公式描述。 8: 修改 GPIOD 基地址为 0x4200300	2025.09