

AOM62196

32 位超低功耗微处理器芯片

1. 功能特点

- 超低功耗平台
 - 1.8V 至 5.5V 供电
 - -40°C 至 85°C 温度范围
 - 0.6μA 深度睡眠模式
 - 1.0μA 深度睡眠模式 + RTC
 - 低速工作模式低至 8μA (32.768 kHz)
 - 睡眠模式低至 37μA / MHz
 - 工作模式低至 140μA / MHz
 - 5μs 唤醒时间
- 核心: ARM 32位 Cortex-M0+
 - 32kHz 至 48MHz
 - 0.95 DMIPS/MHz
- 存储器
 - 高达 256KB FLASH存储器
 - 带有奇偶校验的 32KBRAM存储器
- 多达 85 个高速GPIO引脚
- 复位及电源管理
 - 超低功耗 POR
 - 低电压探测 LVD
- 时钟源
 - 4MHz 至 32MHz 晶体振荡器
 - 经过校准的 32.768kHz RTC振荡器
 - 经过工厂校准的内部 4MHz 至 24MHz RC振荡器 (±2%)
 - 内部低功耗 38kHz RC振荡器
 - 内部低速时钟 32.8kHz / 38.4kHz
 - 用于CPU时钟的PLL
- 开发支持
 - 串行调试支持
- LCD 驱动多达 8x32 段
 - 支持对比度调节
 - 支持闪烁模式
 - 片上升压转换器
- 丰富的模拟外设
 - 多达22个通道的 1MSPS 12位ADC
 - 2 x 带有输出缓冲的12位DAC
 - 4 x 运算放大器
 - 4 通道 DMA 控制器,支持ADC, SPI, I2C, UART, LPUART, DAC, Timer
 - 18 x 外设通信接口
 - 6 x UART, 2 x LPUART
 - 4 x SPI 最高速率 12 Mbits/s
 - 4 x I2C 最高速率 1 Mbits/s
 - 1 x I2S 最高采样率 48kSPS
 - 1 x 7816
 - 2 x CAN2.0B
 - 定时器: 9x 16位BTIM, 3x 32位GTIM, 1x 高级ATIM, 1x STIM, 2x 低功耗LPTIM, 1x 可编程计数阵列 PCA, 1x 低功耗脉冲计数器 PCNT, 1x RTC, 1x IWDG, 1x WWDG
 - CRC 计算单元
 - 80-bit 全球唯一ID
 - TRNG 真随机数发生器
 - AES-128/192/256 硬件加密引擎

2. 应用

- 水电气表 工业传感器
- 医疗及健身设备
- 遥控 用户接口
- PC 外设 游戏/导航设备
- 监控系统 有线/无线传感器 视频对讲

3. 芯片概述

AOM62196 是一款集成是一款集成高集成度的 32bit RISC 微处理器, 运算能力达到 0.95 Dhrystone MIPS/MHz。内部集成了 12 位 1M sps 高精度 SARADC, 1 个 12 位 DAC 以及集成了比较器、运放、内置高性能 PWM 定时器、LCD 显示、多路 UART、SPI、I2C 等丰富的通讯外设。

AOM62196 内建 AES、TRNG 等信息安全模块, 具有高整合度、高抗干扰、高可靠性和超低功耗的特点。

该产品内核兼容 Cortex-M0+ 内核, 配合成熟的 Keil & IAR 调试开发软件, 支持 C 语言及汇编语言, 汇编指令。

4. 芯片信息

表 4-1

型号	AOM62196PXX
温度范围	-40℃ to 85℃
封装	LQFP100/64/48/QFN32 XX=100,64,48,32

5. 功能模块

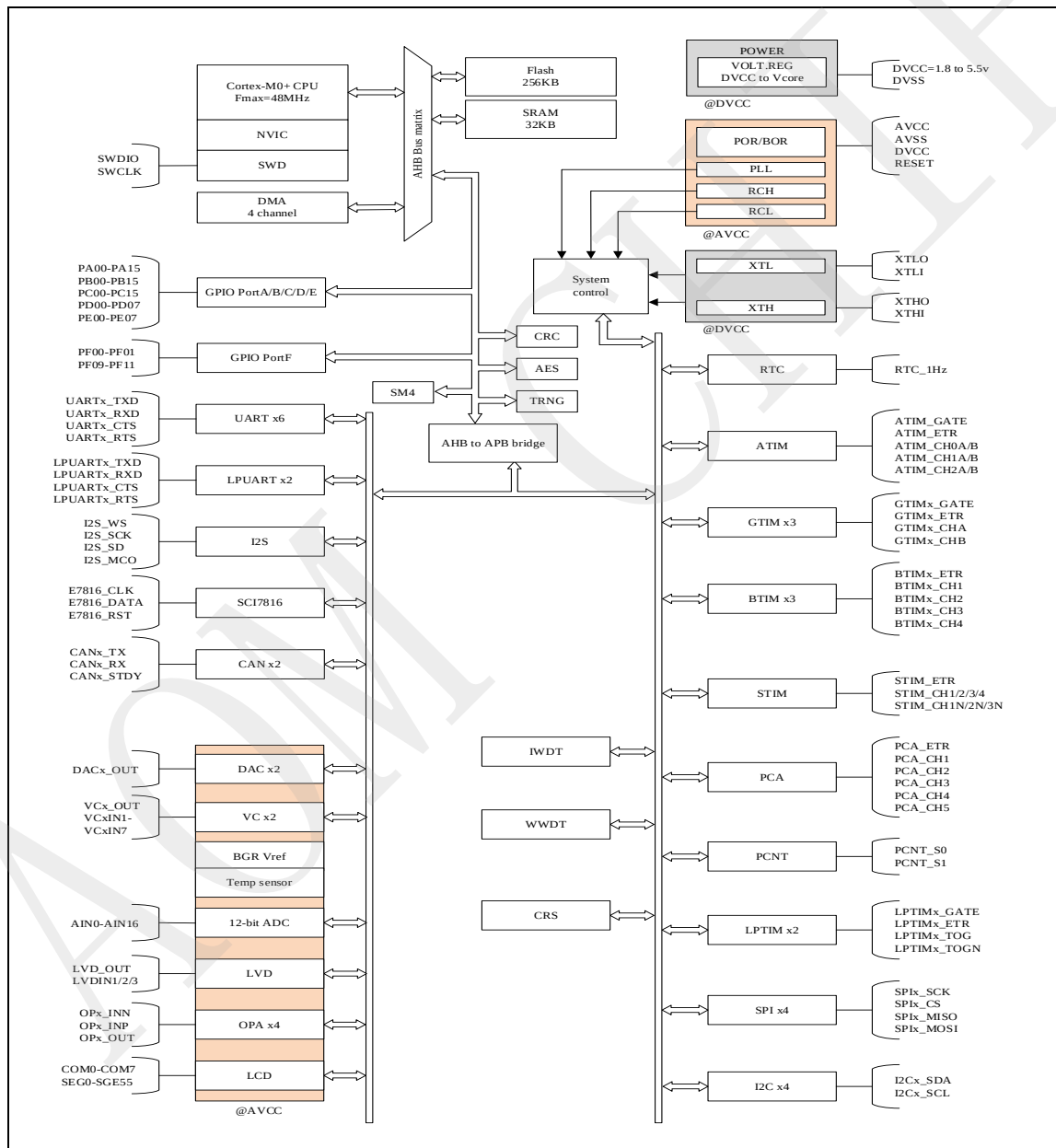


图 5-1 功能模块

5.1 32 位 CORTEX M0+ 内核

ARM® Cortex®-M0+ 处理器源于 Cortex-M0，包含了一颗 32 位 RISC 处理器，运算能力达到 0.95 Dhrystone MIPS/MHz。同时加入了多项全新设计，改进调试和追踪能力、减少每条指令循环（IPC）数量和改进 Flash 访问的两级流水线等，更纳入了节能降耗技术。Cortex-M0+ 处理器全面支持已整合 Keil & IAR 调试器。

Cortex-M0+ 包含了一个硬件调试电路，支持 2-pin 的 SWD 调试。

ARM Cortex-M0+ 特性：

指令集	Thumb / Thumb-2
流水线	2级流水线
性能效率	2.46 CoreMark / MHz
性能效率	0.95 DMIPS / MHz in Dhrystone
中断	32个快速中断
中断优先级	可配置4级中断优先级
增强指令	单周期32位乘法器
调试	Serial-wire 调试端口，支持4个硬中断（break point）以及2个观察点（watch point）

5.2 256K Byte FLASH

内建全集成 FLASH 控制器，无需外部高压输入，由全内置电路产生高压来编程。支持 ISP、IAP、ICP 功能。

5.3 32K Byte RAM

根据客户选择不同的超低功耗模式，RAM 数据都会被保留。自带硬件奇偶校验位，万一数据被意外破坏，在数据被读取时，硬件电路会立刻产生中断，保证系统的可靠性。

5.4 时钟系统

一个频率为 4M~24MHz 可配置的高精度内部时钟 RCH。在配置 24MHz 下，从低功耗模式到工作模式的唤醒时间为 5μs，全电压全温度范围内的频率偏差 < ±2.5%，无需外接昂贵的高频晶体。

- 一个频率为 4M~32MHz 的外部晶振 XTH。
- 一个频率为 32.768kHz 的外部晶振 XTL，主要提供 RTC 实时时钟。

- 一个频率为 32.8 kHz /38.4kHz 的内部时钟 RCL。
- 一个频率为 8M~48MHz 输出的 PLL。

5.5 工作模式

- 1) 运行模式(Active Mode): CPU 运行, 周边功能模块运行。
- 2) 休眠模式(Sleep Mode): CPU 停止运行, 周边功能模块运行。
- 3) 深度休眠模式(Deep sleep Mode): CPU 停止运行, 高速时钟停止, 低功耗功能模块运行。

5.6 硬件实时时钟 RTC

RTC (Real Time Counter) 是一个支持 BCD 数据的寄存器, 采用 32.768kHz 晶振作为其时钟, 能实现万年历功能, 中断周期可配置为年/月/日/小时/分钟/秒。24/12 小时时间模式, 硬件自动修正闰年。具有精确度补偿功能, 最高精度为 0.96ppm。可使用内部温度传感器或外部温度传感器进行精确度补偿, 可用软件+1/-1 调整年/月/日/小时/分钟/秒, 最小可调精度为 1 秒。

用于指示时间和日期的 RTC 日历记录器在 MCU 受外部因素影响而复位时不会清除保留值, 是需要永久高精度实时时钟的测量设备仪表的最佳选择。

5.7 通用 IO 端口

最多可提供 85 个 GPIO 端口, 其中部分 GPIO 与模拟端口复用。每个端口由独立的控制寄存器位来控制。支持边沿触发中断和电平触发中断, 可从各种超低功耗模式下把 MCU 唤醒到工作模式。支持位置位, 位清零, 位置位清零操作。支持 Push-Pull CMOS 推挽输出、Open-Drain 开漏输出。内置上拉电阻、下拉电阻, 带有施密特触发器输入滤波功能。输出驱动能力可配置, 最大支持 12mA 的电流驱动能力。所有通用 IO 可支持外部异步中断。

5.8 中断控制器

Cortex-M0+处理器内置了嵌套向量中断控制器(NVIC),支持最多 32 个中断请求(IRQ)输入;有四个中断优先级,可处理复杂逻辑,能够进行实时控制和中断处理。32 个中断入口向量地址,参考用户手册中断章节。

5.9 复位控制器

本产品具有 7 个复位信号来源,每个复位信号可以让 CPU 重新运行,绝大多数寄存器会被重新复位,程序计数器 PC 会指向起始地址。

	中断来源
[0]	上电掉电复位 POR BOR
[1]	外部 Reset Pin 复位
[2]	WDT 复位
[3]	WWDT 复位
[4]	Cortex-M0+ LOCKUP 硬件复位
[5]	Cortex-M0+ SYSRESETREQ 软件复位
[6]	LVD 复位

5.10 DMAC

DMAC(直接内存访问控制器)功能块可以不通过 CPU 高速传输数据。使用 DMAC 能提高系统性能。

- DMAC 配有独立的总线,所以即便是在使用 CPU 总线的同时,DMAC 也可进行传输操作。
- 由 4 条通道组成,能执行 4 种相互独立的 DMA 传输。
- 可设置传输目标地址、传输源地址、传输数据大小、传输请求源以及传输模式,并能控制各通道的传输操作启动、传输的强行终止以及传输的暂停。
- 可控制所有通道批量传输的启动、强行终止及暂停。
- 多通道同时操作时,可用固定方法或循环方法选择操作通道的优先级。
- 支持使用外设中断信号的硬件 DMA 传输。
- 遵从系统总线(AHB),支持 32 位地址空间(4GB)。

5.11 定时器/计数器

类型	名称	位宽	预除频	计数方向	PWM	捕获	互补输出
通用定时器	GTIM1	16/32	1/2/4/8/16/32/64/256	上计数/ 下计数/ 上下计数	2	2	1
	GTIM2	16/32	1/2/4/8/16/32/64/256	上计数/ 下计数/ 上下计数	2	2	1
	GTIM3	16/32	1/2/4/8/16/32/64/256	上计数/ 下计数/ 上下计数	2	2	1
高级定时器	ATIM	16/32	1/2/4/8/16/32/64/256	上计数/ 下计数/ 上下计数	6	6	3
	STIM	16/32	1/.../65536	上计数/ 下计数/ 上下计数	6	6	3
低功耗定时器	LPTIM0	16	1/2/4/8/16/32/64/256	上计数	无	无	无
	LPTIM1	16	1/2/4/8/16/32/64/256	上计数	无	无	无
可编程计数阵列	PCA	16	2/4/8/16/32	上计数	5	5	无
复合定时器	保留						
基本定时器	BTIM1-9	16	1/2/4/8/16/64/256/1024	上计数	无	无	无

通用定时器包含 3 个定时器 GTIM1/2/3。

通用定时器特性：

- PWM 独立输出，互补输出
- 捕获输入
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能

高级定时器 ATIM 是多通道的通用定时器,具有 GTIM1/2/3 的所有功能,可以产生 3 组 PWM 互补输出或 6 路 PWM 独立输出,最多 6 路输入捕获。具有死区控制功能。

高级定时器特性:

- 4 路 PWM 独立输出
- 4 路捕获输入
- 单脉冲模式
- 外部计数功能

低功耗定时器 LPTIM 是异步 16 位定时/计数器,在系统时钟关闭后仍然可以通过内部低速 RC 或者外部低速晶体振荡计时/计数。通过中断在低功耗模式下唤醒系统。

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可用作为一个通用的时钟计数/事件计数器的捕获/比较功能。

PCA 的每个模块都可以进行独立编程,以提供输入捕捉,输出比较或脉冲宽度调制。

超低功耗脉冲计数器 PCNT

PCNT (Pulse Counter) 模块用以对外部脉冲进行计数,支持单路以及双路(正交编码与非交叉编码)脉冲。它可以在低功耗休眠模式下无需软件参与进行计数。

脉冲计数器特性:

- 支持重载功能的 16 bit 计数器
- 单通道脉冲计数
- 双通道非交脉冲计数
- 双通道正交脉冲计数,不失码
- 加/减计数溢出中断
- 脉冲超时中断
- 4 种解码错误中断,非交脉冲模式
- 1 种方向改变中断,正交脉冲模式
- 多级脉冲宽度滤波
- 输入脉冲极性可配置
- 支持低功耗模式计数
- 支持唤醒低功耗模式下 MCU

- 支持任意脉冲沿间距不小于 1 个计数时钟周期

5.12 独立看门狗 IWDG

IWDG 是一个可配置的定时器，在 MCU 异常的情况下提供复位；内建 10kHz 低速时钟输入作为计数器时钟。调试模式下，可选择暂停或继续运行；只有写入特定序列才能重启 IWDG。

5.13 窗口看门狗 WWDG

WWDG 是一个可配置的定时器，在 MCU 异常的情况下提供复位；使用系统时钟输入作为计数器时钟。调试模式下，可选择暂停或继续运行；只有在窗口内才可以写入特定序列喂狗。

5.14 通用异步收发器 UART

6 路通用异步收发器 (Universal Asynchronous Receiver/Transmitter), UART1~UART6。

通用 UART 基本功能：

- 半双工和全双工传输
- 8/9-Bit 传输数据长度
- 硬件奇偶校验
- 1/1.5/2-Bit 停止位
- 四种不同传输模式
- 16-Bit 波特率计数器
- 多机通讯
- 硬件地址识别
- DMAC 硬件传输握手
- 硬件流控
- 支持单线模式

5.15 低功耗异步收发器 LPUART

2 路低功耗模式下可以工作的异步收发器 (Low Power Universal Asynchronous

Receiver/Transmitter)，LPUART1/LPUART2。

LPUART 基本功能：

- 传输时钟 SCLK（SCLK 可选择 XTL、RCL 以及 PCLK）
- 系统低功耗模式下收发数据
- 半双工和全双工传输
- 8/9-Bit 传输数据长度
- 硬件奇偶校验
- 1/1.5/2-Bit 停止位
- 四种不同传输模式
- 16-Bit 波特率计数器
- 多机通讯
- 硬件地址识别
- DMAC 硬件传输握手
- 硬件流控
- 支持单线模式

5.16 同步串行接口 SPI

4 路同步串行接口（Serial Peripheral Interface）

SPI 基本特性：

- 可配置为主机或者从机
- 基于两根单向数据线的全双工通信
- 基于一根单向数据线的单工通信
- 4 位到 16 位连续可调的帧宽度选择
- 支持多机模式
- 主机模式 7 种波特率可配置，最高可达 $fpclk/2$
- 从机模式波特率最高可达 $fpclk/4$
- 主机模式和从机模式都可配置 NSS 功能——硬件管脚或软件控制
- 可配置的串行时钟极性和相位

- 可配置的数据比特位顺序——先收发 MSB 或先收发 LSB
- 发送和接收标志位，支持触发中断
- SPI 总线忙标志位
- 多种错误标志位，均支持触发中断
- 发送和接收分别配有容量为一帧的支持 DMA 访问的缓冲

5.17 I2C 总线

4 路 I2C，采用串行同步时钟，可实现设备之间以不同的速率传输数据。

I2C 基本特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps) / 快速(400Kbps) / 高速(1Mbps) 三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

5.18 蜂鸣器 Buzzer

所有定时器的特定模式输出都可以为 Buzzer 提供可编程驱动频率。该蜂鸣器端口可提供 12mA 的 sink 电流，互补输出，不需要额外的三极管。

5.19 时钟校准电路

内建时钟校准电路，可以通过外部精准的晶振时钟校准内部 RC 时钟，亦可使用内部 RC 时钟去检验外部晶振时钟是否工作正常。

时钟校准基本特性：

- 自动校准模式
- 手动校准模式
- 定时器模式

5.20 唯一识别号 UID

每颗芯片出厂前具备唯一的 10 字节设备标识号，包括 wafer lot 信息，以及芯片坐标信息等。UID 地址为：(TBD)

5.21 CRC16/32 硬件循环冗余校验码

CRC16 符合 ISO/IEC13239 中给出的多项式 $x^{16}+x^{12}+x^5+1$ 。

CRC32 符合 ISO/IEC13239 中给出的多项式 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ 。

5.22 AES 硬件加密

AES (The Advanced Encryption Standard) 是美国国家标准技术研究所 (NIST) 在 2000 年 10 月 2 日正式宣布的新的数据加密标准。AES 的分组长度固定为 128 位，而密钥长度支持 128/192/256。

5.23 TRNG 真随机数发生器

TRNG 是一个真随机数发生器，用来产生真随机数。

5.24 12 Bit SARADC

单调不失码的 12 位逐次逼近型模数转换器，在 24M ADC 时钟下工作时，采样率达到 1MSPS。参考电压可选择片内精准电压 (1.5v 或 2.5v) 或从外部输入或电源电压。30 个输入通道，包括 26 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 电源电压、1 路内建 BGR 1.2V 电压、DAC 内部输出。内建可配置的输入信号放大器以检测弱信号。

SAR ADC 基本特性：

- 12 位转换精度；
- 1MSPS 转换速度；
- 22 个输入通道，包括 17 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 AVCC 电压、1 路内建 BGR 1.2V 电压、DAC 内部输出；

- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；
- ADC 的电压输入范围：0~Vref；
- 4 种转换模式：单次转换、顺序扫描连续转换、插队扫描连续转换、连续转换累加；
- 输入通道电压阈值监测；
- 软件可配置 ADC 的转换速率；
- 内置信号放大器，可转换高阻信号；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

5.25 12 Bit DAC

2 通道 12bit 500kSPS DAC, 可以进行数模转换。

5.26 电压比较器 VC

内建 2 路 VC，芯片管脚电压监测/比较电路。多个可配置的外部输入通道；多个内部负输入通道，包括 1 路内部温度传感器电压、1 路内建 BGR 2.5V 参考电压、1 路内建 BGR 1.2V 电压、1 路 64 阶电阻分压。可根据上升/下降边沿产生异步中断，从低功耗模式下唤醒 MCU。可配置的软件防抖功能。

5.27 低电压检测器 LVD

对芯片电源电压或芯片管脚电压进行检测。16 档电压监测值（1.8v ~ 3.3v）。可根据上升/下降边沿产生异步中断或复位。具有硬件迟滞电路和可配置的软件防抖功能。

LVD 基本特性：

- 4 路监测源，AVCC、PA00、PC13、PE06；
- 16 阶阈值电压，1.8V~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

5.28 运放 OPA

OPA 可以灵活配置，适用于简易滤波器和电压跟随器应用。可以作为 DAC 输出缓存器使用，也可以配置为运放使用。

5.29 LCD 驱动

LCD 控制器是一款适用于单色无源液晶显示器（LCD）的数字控制器/驱动器，最多具有 8 个公用端子（COM）和 32 个区段端子（SEG）。可以选择电容分压或电阻分压，支持内部电阻分压。内部电阻分压可以调节对比度。支持 DMA 硬件数据传输。

LCD 基本特性：

- 高度灵活的帧速率控制。
- 支持静态、1/2、1/3、1/4、1/6 和 1/8 占空比。
- 支持 1/2、1/3 偏置。
- 多达 16 个寄存器的 LCD 数据 RAM。
- 可通过软件配置 LCD 的对比度。
- 3 种驱动波形生成方式
 - 内部电阻分压、外部电阻分压，外部电容分压方式
 - 可通过软件配置内部电阻分压方式的功耗，从而匹配 LCD 面板所需的电容电荷
- 支持低功耗模式：LCD 控制器可在 Active、Sleep、DeepSleep 模式下进行显示。
- 可配置帧中断。
- 支持 LCD 闪烁功能且可配置多种闪烁频率
- 未使用的 LCD 区段和公共引脚可配置为数字或模拟功能。

5.30 嵌入式调试系统

嵌入式调试解决方案，提供全功能的实时调试器，配合标准成熟的 Keil/IAR 等调试开发软件。支持 4 个硬断点以及多个软断点。

5.31 在线编程模式

支持在线编程，将 BOOT0 (PF11) 管脚接高电平即可进入 ISP 在线烧录模式。BOOT0 管脚接低电平进入用户模式。

5.32 高安全性

加密型嵌入式调试解决方案，提供全功能的实时调试器。

6. 引脚配置及功能

6.1 封装示意图

6.1.1 LQFP100

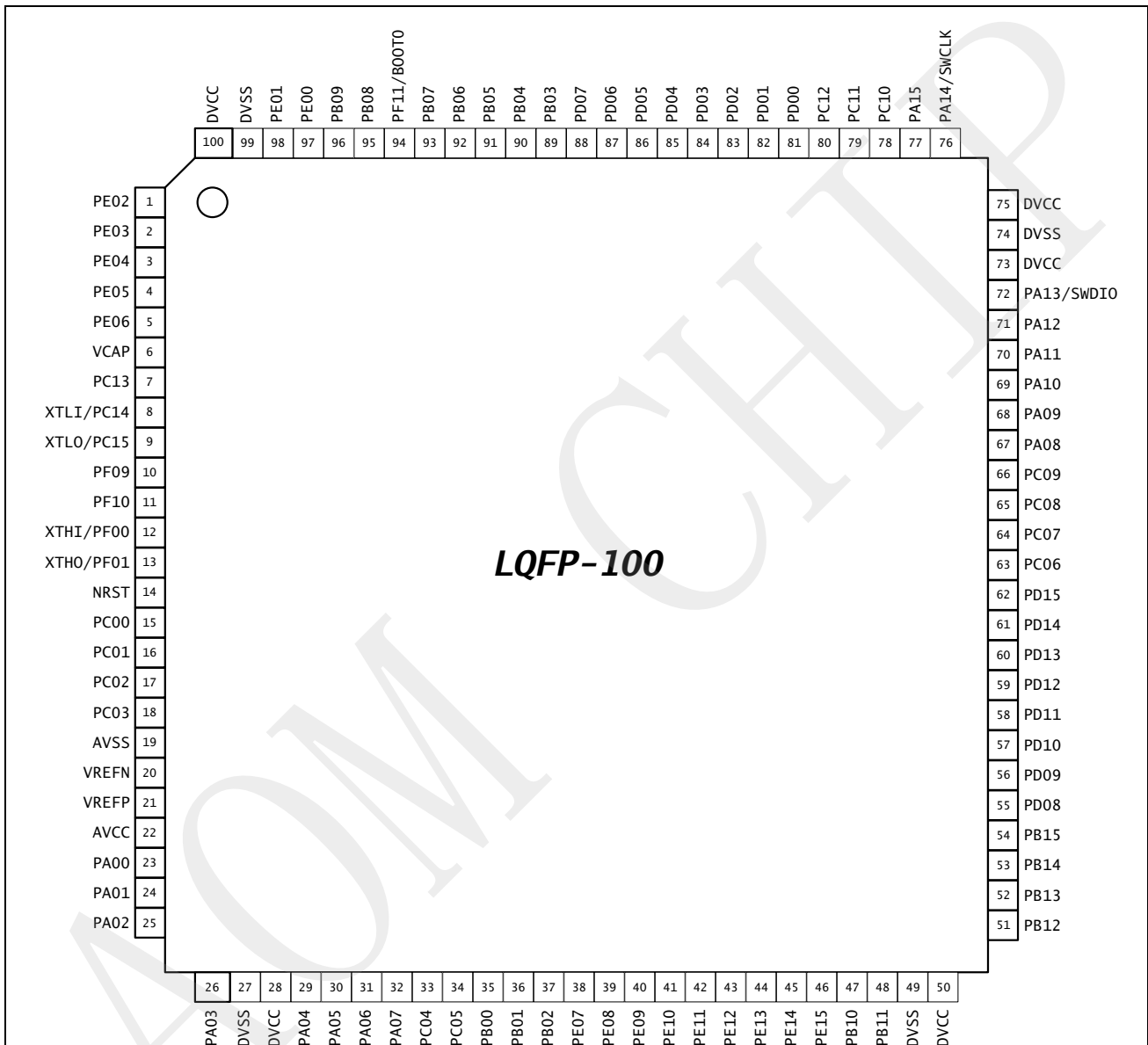
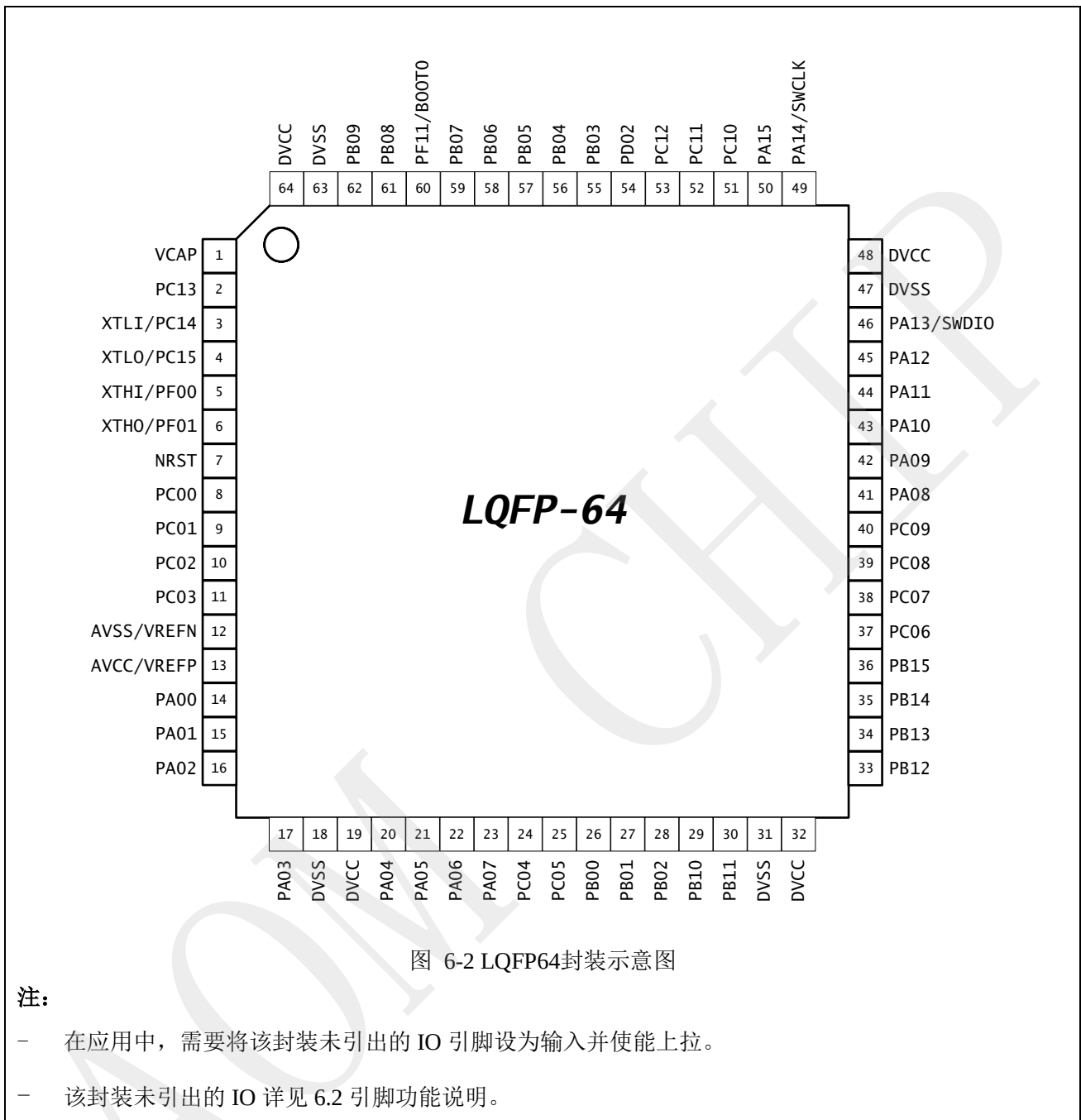


图 6-1 LQFP100封装示意图

注：

- 在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。
- 该封装未引出的 IO 详见 6.2 引脚功能说明。

6.1.2 LQFP64



6.1.3 LQFP48

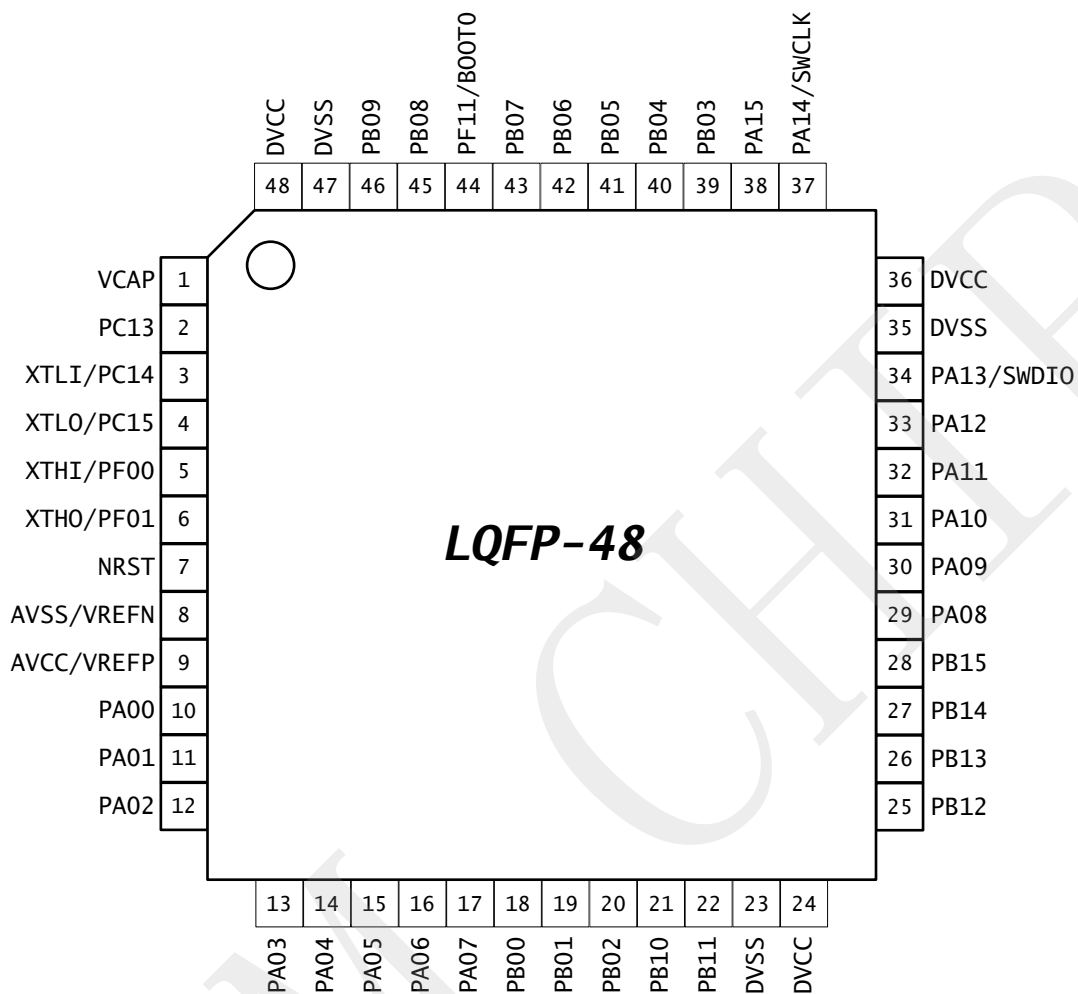


图 6-3 LQFP48封装示意图

注:

- 在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。
- 该封装未引出的 IO 详见 6.2 引脚功能说明。

6.1.4 QFN32

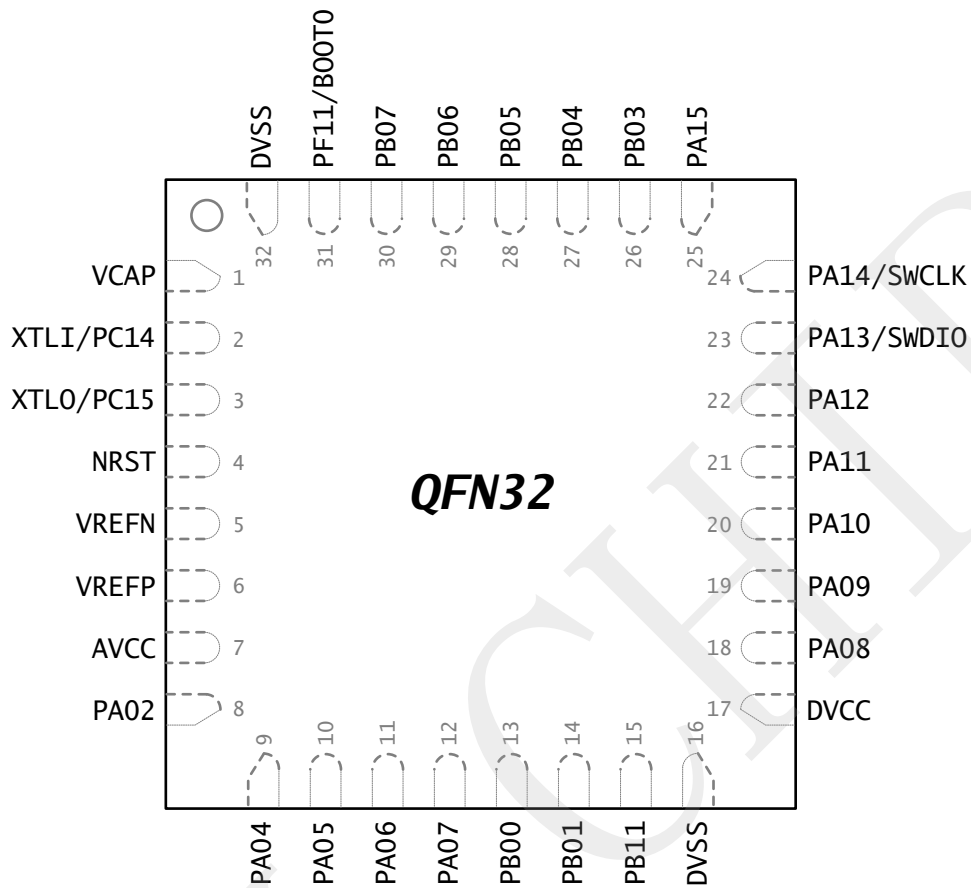


图 6-4 QFN32封装示意图

注:

- 在应用中，需要将该封装未引出的 IO 引脚设为输入并使能上拉。
- 该封装未引出的 IO 详见 6.2 引脚功能说明。

6.2 引脚功能说明

LQFP 100	LQFP64	LQFP48	QFN32	GPIO	数字复用功能	模拟复用功能
1	—	—	—	PE02	GTIM2_ETR SPI3_MISO UART6_CTS LPUART2_TX PCA_CH1 STIM_CH1	
2	—	—	—	PE03	GTIM1_CH1 SPI3_MOSI UART6_RTS LPUART2_RX PCA_CH2 STIM_CH2	
3	—	—	—	PE04	GTIM1_CH2 STIM_BKIN UART6_TX LPUART2_CTS PCA_CH3 STIM_CH3	
4	—	—	—	PE05	STIM_BKIN2 UART6_RX LPUART2_RTS PCA_CH4 STIM_CH4	
5	—	—	—	PE06	GTIM1_BKIN UART6_CK GTIM2_ETR PCA_CH5 STIM_ETR	LVD_EXT3
6	1	1	1	VCAP		
7	2	2	—	PC13	e78162_rst SPI3_SCK GTIM1_ETR PCA_ETR	LVD_EXT2
8	3	3	2	PC14		X32K_IN
9	4	4	3	PC15		X32K_OUT

10	–	–	–	PF09	I2C3_SCL SPI3_NSS GTIM1_GT LPTIM1_TB	
11	–	–	–	PF10	I2C3_SDA LPTIM1_TA	
12	5	5	–	PF00	CRS_SYNC	OSC_IN
13	6	6	–	PF01		OSC_OUT
14	7	7	4	NRST		
15	8	–	–	PC00	LPTIM1_TB GTIM2_GT LPUART1_RX I2C3_SCL ATIM_CH1	SEG20/VLCD1 ADC_IN10
16	9	–	–	PC01	LPTIM1_TA GTIM3_ETR LPUART1_TX I2C3_SDA ATIM_CH2	SEG21/VLCD2 ADC_IN11
17	10	–	–	PC02	LPTIM1_GT SPI2_MISO UART4_RTS I2C1_SCL ATIM_CH3	SEG22/VLCD3 ADC_IN12
18	11	–	–	PC03	LPTIM1_ETR SPI2_MOSI UART4_CTS I2C1_SDA ATIM_GT	SEG23/VLCD4 ADC_IN13
19	–	–	–	AVSS		
20	–	–	5	VREFN		
–	12	8	–	VREFN/ AVSS		
21	–	–	6	VREFP		
22	–	–	7	AVCC		
–	13	9	–	VREFP/ AVCC		

23	14	10	–	PA00	SPI3_NSS ATIM_CH1 VC1_OUT UART2_CTS ATIM_ETR UART4_TX	ADC_IN0 VC1_INN VC2_INN LVD_EXT1
24	15	11	–	PA01	SPI3_SCK ATIM_CH2 GTIM2_ETR UART2_RTS UART4_RX	SEG0 ADC_IN1 VC1_INN VC1_INP VC2_INP VC2_INN
25	16	12	8	PA02	SPI3_MISO ATIM_CH3 VC2_OUT UART2_TX GTIM1_GT LPUART1_TX	SEG1 ADC_IN2 OPA3N VC1_INN VC1_INP VC2_INP VC2_INN
26	17	13	–	PA03	SPI3_MOSI ATIM_GT LPTIM1_GT UART2_RX GTIM3_ETR LPUART1_RX	SEG2 ADC_IN3 OPA3P VC1_INN VC1_INP VC2_INP VC2_INN
27	18	–	–	DVSS		
28	19	–	–	DVCC		
29	20	14	9	PA04	SPI1_NSS LPTIM1_ETR I2C1_SCL UART2_CK GTIM1_ETR UART5_TX	ADC_IN4 OPA3_OUT VC1_INN VC1_INP VC2_INP VC2_INN DAC_OUT1
30	21	15	10	PA05	SPI1_SCK ATIM_ETR I2C1_SDA ATIM_CH1 GTIM2_GT UART5_RX	ADC_IN5 OPA4_OUT VC1_INN VC1_INP VC2_INP VC2_INN DAC_OUT2

31	22	16	11	PA06	SPI1_MISO STIM_BKIN LPUART1_CTS GTIM1_CH1 VC1_OUT	SEG3 ADC_IN6 OPA4N
32	23	17	12	PA07	SPI1_MOSI STIM_CH1N VC2_OUT ATIM_BK GTIM1_CH2 UART5_RTS	SEG4 ADC_IN7 OPA4P
33	24	—	—	PC04	LPTIM2_TB LPUART1_TX I2C3_SCL GTIM1_CH1 UART5_CK UART5_CTS	ADC_IN14
34	25	—	—	PC05	LPTIM2_TA LPUART1_RX I2C3_SDA GTIM1_CH2 UART6_CK SPI1_NSS	ADC_IN15
35	26	18	13	PB00	ATIM_CH1N STIM_CH2N PCNT_S1 GTIM1_GT UART6_TX SPI1_SCK	SEG5 ADC_IN8 OPA1_OUT VC1_INN VC1_INP VC2_INN VC2_INP
36	27	19	14	PB01	ATIM_CH2N STIM_CH3N CAN2_RX LPUART1_RTS UART6_RX SPI1_MISO	SEG6 ADC_IN9 OPA1_N
37	28	20	—	PB02	ATIM_CH3N STIM_CH2 LPTIM1_TA CAN2_TX PCNT_S2 LVD_OUT	OPA1_P

					SPI1_MOSI	
38	-	-	-	PE07	LPTIM1_TB UART6_CTS UART5_RTS	
39	-	-	-	PE08	ATIM_BK SPI4_NSS LPUART2_TX PCA_ETR UART6_RTS UART4_TX	
40	-	-	-	PE09	ATIM_CH1 SPI4_SCK ATIM_ETR LPUART2_RX PCA_CH1 MCO UART4_RX	
41	-	-	-	PE10	ATIM_CH2 SPI4_MISO I2C4_SCL PCA_CH2 MCO UART5_TX	
42	-	-	-	PE11	ATIM_CH3 SPI4_MOSI I2C4_SDA LPUART2_CTS PCA_CH3 UART3_TX UART5_RX	
43	-	-	-	PE12	ATIM_GT LPTIM2_TB SPI1_NSS LPUART2_RTS PCA_CH4 UART3_RX GTIM2_CH1	

44	-	-	-	PE13	ATIM_CH1N LPTIM2_TA SPI1_SCK GTIM3_CH1 PCA_CH5 UART3_CK GTIM2_CH2	
45	-	-	-	PE14	ATIM_CH2N LPTIM2_GT SPI1_MISO GTIM3_CH2 UART3_CTS GTIM2_GT	
46	-	-	-	PE15	ATIM_CH3N LPTIM2_ETR SPI1_MOSI GTIM3_GT UART3_RTS GTIM2_BKIN	
47	29	21	-	PB10	CAN1_STBY LPUART1_RX ATIM_CH3 GTIM3_BKIN LPUART1_TX SPI2_SCK I2C2_SCL	SEG7
48	30	22	15	PB11	LPUART1_TX PCNT_S1 ATIM_GT GTIM3_ETR LPUART1_RX e78162_clk I2C2_SDA	SEG8
49	31	23	16	DVSS		
50	32	24	17	DVCC		
51	33	25	-	PB12	SPI2_NSS STIM_BKIN LPUART1_RTS PCNT_S2 e78162_rst GTIM1_GT	SEG9 ADC_AIN16 OPA2_OUT VC1_INN VC1_INP VC2_INN VC2_INP

52	34	26	—	PB13	SPI2_SCK STIM_CH1N GTIM1_GT e78162_data LPUART1_CTS I2C2_SCL	SEG10 OPA2_N
53	35	27	—	PB14	SPI2_MISO STIM_CH2N RTC_OUT LPUART1_RTS I2C2_SDA	SEG11 OPA2_P
54	36	28	—	PB15	SPI2_MOSI STIM_CH3N GTIM1_GT ATIM_CH1N STIM_BKIN	SEG12
55	—	—	—	PD08	LPUART1_TX PCA_ETR I2C4_SCL I2S_CK UART6_TX ATIM_CH2N STIM_CH1	SEG28
56	—	—	—	PD09	LPUART1_RX PCA_CH1 I2C4_SDA STIM_BKIN2 UART6_RX ATIM_CH3N STIM_CH2	SEG29
57	—	—	—	PD10	PCA_CH2 CAN2_RX LPTIM2_TB UART6_CK ATIM_CH1 STIM_CH3	SEG30

58	-	-	-	PD11	LPUART1_CTS PCA_CH3 CAN2_TX LPTIM2_TA UART6_CTS ATIM_CH2 STIM_CH4	SEG31
59	-	-	-	PD12	LPUART1_RTS PCA_CH4 CAN2_STBY LPTIM2_GT UART6_RTS ATIM_CH3 STIM_ETR	
60	-	-	-	PD13	LPTIM2_TB PCA_CH5 LPTIM2_ETR ATIM_GT UART4_TX STIM_CH1N	
61	-	-	-	PD14	LPTIM2_TA SPI3_NSS MCO ATIM_ETR UART4_RX STIM_CH2N	
62	-	-	-	PD15	CRS_SYNC SPI3_SCK MCO ATIM_BK UART4_CK STIM_CH3N	
63	37	-	-	PC06	GTIM1_CH1 SPI3_MISO SPI4_NSS GTIM2_CH1 UART3_TX UART4_RTS	SEG24

64	38	—	—	PC07	GTIM1_CH2 SPI3_MOSI SPI4_MOSI GTIM2_CH2 UART3_RX UART4_CTS	SEG25
65	39	—	—	PC08	GTIM1_ETR STIM_CH1 SPI4_SCK GTIM2_GT UART3_CK	SEG26
66	40	—	—	PC09	STIM_CH2 SPI4_MISO GTIM2_BKIN UART3_CTS I2C3_SDA	SEG27
67	41	29	18	PA08	MCO STIM_CH1 CRS_SYNC LPUART2_TX UART1_CK UART3_RTS I2C3_SCL	COM0
68	42	30	19	PA09	MCO STIM_CH2 LVD_OUT LPUART2_RX UART1_TX GTIM1_BKIN I2C1_SCL	COM1
69	43	31	20	PA10	e78161_data STIM_CH3 CAN1_STBY LPUART2_CTS UART1_RX GTIM3_BKIN I2C1_SDA	COM2

70	44	32	21	PA11	SPI1_MISO STIM_CH4 CAN1_RX LPUART2_RTS UART1_CTS STIM_BKIN2 GTIM3_CH1	
71	45	33	22	PA12	SPI1_MOSI STIM_ETR CAN1_TX SPI4_NSS UART1_RTS GTIM2_CH1 GTIM3_CH2	
72	46	34	23	PA13	e78161_clk UART3_TX I2C4_SCL SPI4_MOSI GTIM3_ETR GTIM2_CH2 LPUART1_RX	SWDIO
73	–	–	–	DVCC		
74	47	35	–	DVSS		
75	48	36	–	DVCC		
76	49	37	24	PA14	e78161_rst UART3_RX I2C4_SDA SPI4_SCK UART2_TX PCA_ETR LPUART1_TX	SWCLK
77	50	38	25	PA15	SPI1_NSS UART3_CK ATIM_ETR SPI4_MISO UART2_RX ATIM_CH1 UART4_RTS	COM3

78	51	-	-	PC10	LPUART1_TX STIM_CH3 UART3_CTS PCA_CH1 I2C2_SCL UART4_TX	COM4
79	52	-	-	PC11	LPUART1_RX STIM_CH4 UART3_RTS PCA_CH2 I2C2_SDA UART4_RX	COM5
80	53	-	-	PC12	e78162_c1k UART5_TX PCA_CH3 LPUART2_TX UART4_CK	COM6
81	-	-	-	PD00	SPI2_NSS CAN1_RX PCA_CH4 LPUART2_RX MCO	
82	-	-	-	PD01	SPI2_SCK CAN1_TX PCA_CH5 LPUART2_CTS MCO	
83	54	-	-	PD02	LPUART1_RTS STIM_CH1N I2C4_SCL LPUART2_RTS UART5_RX	COM7
84	-	-	-	PD03	UART2_CTS SPI2_MISO STIM_CH2N I2C4_SDA SPI2_NSS e78162_data	

85	-	-	-	PD04	UART2_RTS SPI2_MOSI STIM_CH3N SPI2_SCK	
86	-	-	-	PD05	UART2_TX GTIM3_CH1 I2S_MCK STIM_BKIN SPI2_MISO I2S_WS	
87	-	-	-	PD06	UART2_RX GTIM3_GT I2C2_SCL GTIM2_CH1 SPI2_MOSI	
88	-	-	-	PD07	UART2_CK I2C2_SDA I2S_SD GTIM2_CH2	
89	55	39	26	PB03	SPI1_SCK I2S_MCK ATIM_CH2 CAN2_STBY GTIM3_ETR UART1_RTS UART5_TX	SEG13
90	56	40	27	PB04	SPI1_MISO I2C3_SDA GTIM3_BKIN GTIM1_CH1 UART1_CTS UART5_RX	SEG14 VC1_INN VC1_INP VC2_INN VC2_INP
91	57	41	28	PB05	SPI1_MOSI STIM_CH3 GTIM2_BKIN pcnt_S1o GTIM1_CH2 UART1_CK UART5_CK	SEG15 VC1_INN VC1_INP VC2_INN VC2_INP

92	58	42	29	PB06	UART1_TX I2C1_SCL LPTIM1_ETR GTIM2_CH2 pcnt_S2o e78161_clk	SEG16 VC1_INP VC2_INP
93	59	43	30	PB07	UART1_RX I2C1_SDA LPTIM1_GT GTIM3_CH2 I2S_WS e78161_rst UART4_CTS	SEG17 VC1_INP VC2_INP PVD_IN
94	60	44	31	PF11		
95	61	45	—	PB08	CAN1_RX I2S_SD SPI4_NSS GTIM2_CH1 I2C1_SCL GTIM1_BKIN e78161_data	SEG18
96	62	46	—	PB09	CAN1_TX I2S_CK SPI4_MOSI GTIM2_GT I2C1_SDA SPI2_NSS	SEG19
97	—	—	—	PE00	SPI4_SCK	
98	—	—	—	PE01	SPI4_MISO	
99	63	47	32	DVSS		
100	64	48	—	DVCC		

表 6-1 引脚功能说明

6.3 模块信号说明

模块	引脚名称	描述
电源	DVCC	数字电源
	AVCC	模拟电压
	DVSS	数字地
	AVSS	模拟地
	VCAP	LDO内核供电输出（仅限内部电路使用，需外接不小于1uF的去耦电容）

ISP	BOOT	BOOT为高电平进入在线编程模式，通过上位机可以进行在线编程。 BOOT0低电平为工作模式
ADC	AIN0~AIN16	ADC 输入通道0~16
	ADC_VREF	ADC外部参考电压
电压比较VC	VCIN	VC 输入
	VC1_OUT	VC1比较输出
	VC2_OUT	VC2比较输出
LVD	LVDIN0	电压侦测输入0
	LVDIN1	电压侦测输入1
	LVDIN2	电压侦测输入2
	LVD_OUT	电压侦测输出
OPA y=0~4	OPA_INN	OPA负端输入
	OPA_INP	OPA正端输入
	OPA_OUTy	OPA输出
LCD x=0~7 y=0~55 z=1, 2, 3, H	COMx	LCD 公共端输出
	SEGy	LCD区段端输出
	VLCDz	外部电阻模式，外部电容模式使用管脚
UART x=1~6	UARTx_TXD	UARTx数据发送端
	UARTx_RXD	UARTx数据接收端
	UARTx_CTS	UARTx CTS
	UARTx_RTS	UARTx RTS
LPUART x=1, 2	LPUARTx_TXD	LPUART数据发送端
	LPUARTx_RXD	LPUART数据接收端
	LPUARTx_CTS	LPUART CTS
	LPUARTx_RTS	LPUART RTS
SPI x=1~4	SPIx_MISO	SPI模块主机输入从机输出数据信号
	SPIx_MOSI	SPI模块主机输出从机输入数据信号
	SPIx_SCK	SPI模块时钟信号
	SPIx_CS	SPI 片选
I2C x=1~4	I2Cx_SDA	I2C模块数据信号
	I2Cx_SCL	I2C模块时钟信号
通用定时器 GTIMx x=1, 2, 3	GTIMx_CHA	GTimer的捕获输入比较输出A
	GTIMx_CHB	GTimer的捕获输入比较输出B
	GTIMx_ETR	GTimer的外部计数输入信号
	GTIMx_GATE	GTimer的门控信号
高级定时器 ATIM	ATIM_Chy	ATimer的捕获输入比较输出A
	ATIM_CHyN	ATimer的捕获输入比较输出B
	ATIM_ETR	ATimer的外部计数输入信号

y=1, 2, 3	ATIM_GATE	ATimer的门控信号
低功耗定时器 LPTIMx x=1, 2	LPTIMx_TA	LPTimer的翻转输出信号
	LPTIMx_TB	LPTimer的翻转输出反向信号
	LPTIMx_ETR	LPTimer的外部计数输入信号
	LPTIMx_GATE	LPTimer的门控信号
可编程计数阵列PCA	PCA_ETR	外部时钟输入信号
	PCA_CH1	捕获输入/比较输出/PWM输出 0
	PCA_CH2	捕获输入/比较输出/PWM输出 1
	PCA_CH3	捕获输入/比较输出/PWM输出 2
	PCA_CH4	捕获输入/比较输出/PWM输出 3
	PCA_CH5	捕获输入/比较输出/PWM输出 4
PCNT	PCNT_S1	PCNT 脉冲计数输入0
	PCNT_S2	PCNT 脉冲计数输入1
复合定时器	保留	

表 6-2 模块信号说明

SEL[3:0]	VC1P	VC1N	VC2P	VC2N
4'b0000	GND	BGR_1P2	GND	BGR_1P2
4'b0001	PA1	OPA1	PA1	OPA1
4'b0010	PA2	OPA2	PA2	OPA2
4'b0011	PA3	OPA3	PA3	OPA3
4'b0100	PB4	OPA4	PB4	OPA4
4'b0101	PB5	DAC1	PB5	DAC1
4'b0110	PB6	DAC2	PB6	DAC2
4'b0111	PB7	DAC6B_VC1	PB7	DAC6B_VC2
4'b1000	OPA1	vcore	OPA1	vcore
4'b1001	OPA2	temp_sensor	OPA2	temp_sensor
4'b1010	OPA3	PA0	OPA3	PA0
4'b1011	OPA4	PA1	OPA4	PA1
4'b1100	GND	PA2	GND	PA2
4'b1101	GND	PA3	GND	PA3
4'b1110	GND	PB4	GND	PB4
4'b1111	GND	PB5	GND	PB5

表 6-3 VC 输入信号列表

注意：

- IO 端口复位为输入高阻状态，休眠模式和深度休眠模式保持之前的端口状态。

7. 系统结构

7.1 概述

本产品系统由以下部分组成：

- 2 个 AHB 总线 Master:
 - Cortex-M0+
 - DMA 控制器
- 多个 AHB 总线 Slaves:
 - FLASH 存储器
 - SRAM 存储器
 - AHB to APB Bridge, 包含所有 APB 接口外设
 - AHBx, 包含所有 AHB 接口外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。如下图所示：

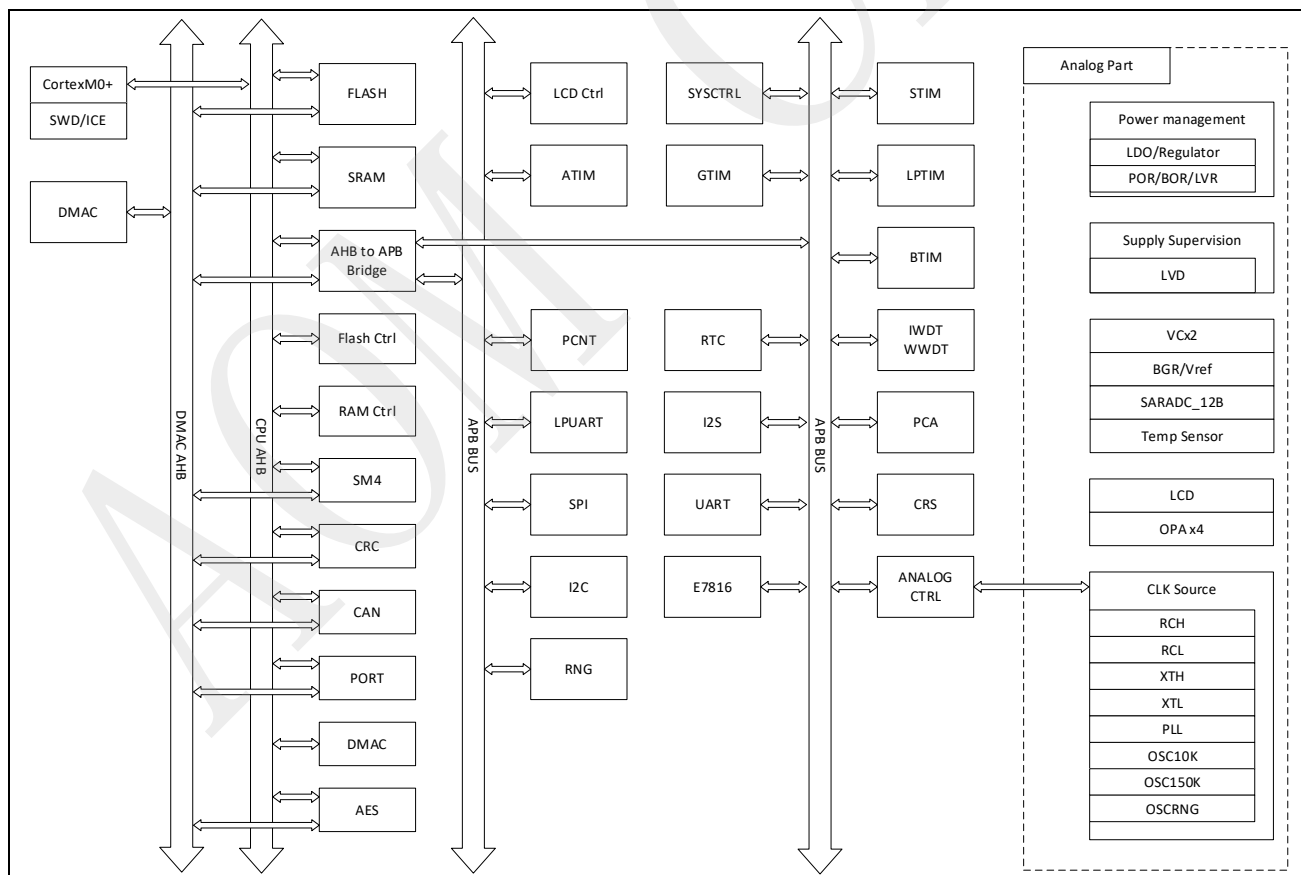


图 7-1 系统架构示意图

7.2 系统地址划分

整个 6615 系列系统的地址区域划分，如下图所示：

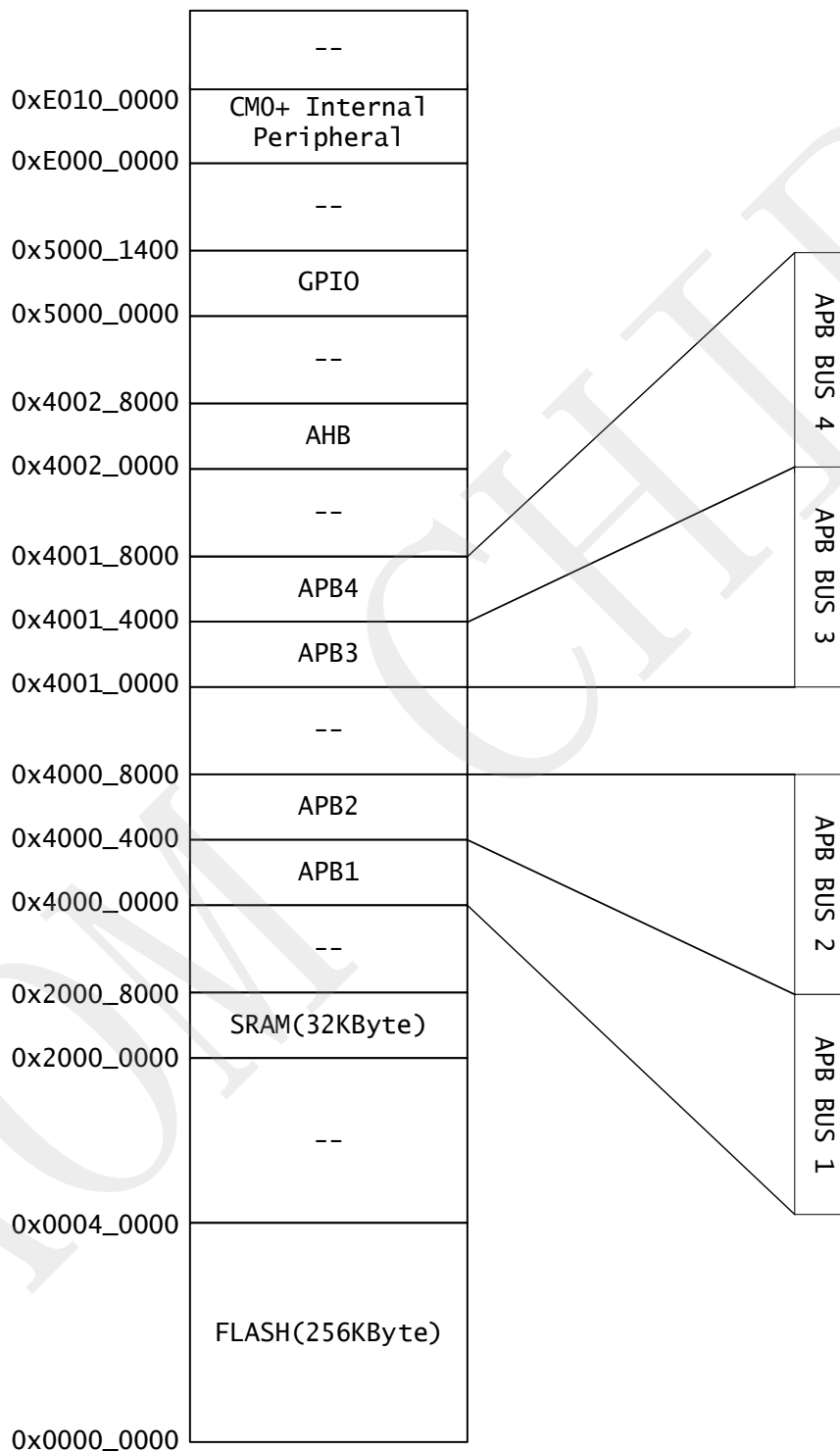


图 7-2 地址区域划分示意图

7.3 存储器和模块地址分配

0x50001400	–	0x500017FF	1KB	GPIOF	GPIO
0x50001000	–	0x500013FF	1KB	GPIOE	
0x50000C00	–	0x50000FFF	1KB	GPIOD	
0x50000800	–	0x50000BFF	1KB	GPIOC	
0x50000400	–	0x500007FF	1KB	GPIOB	
0x50000000	–	0x500003FF	1KB	GPIOA	
0x40026400	–	0x40027FFF	–	–	AHB
0x40026000	–	0x400263FF	1KB	AES	
0x40025800	–	0x40025BFF	1KB	SM4	
0x40025400	–	0x40025BFF	–	–	
0x40025000	–	0x400253FF	1KB	RNG	
0x40024000	–	0x40024FFF	–	–	
0x40023C00	–	0x40023FFF	1KB	CAN2	
0x40023800	–	0x40023BFF	1KB	CAN1	
0x40023400	–	0x400237FF	1KB	–	
0x40023000	–	0x400233FF	1KB	CRC	
0x40022C00	–	0x40022FFF	1KB	–	
0x40022800	–	0x40022BFF	1KB	–	
0x40022400	–	0x400227FF	1KB	RAM	
0x40022000	–	0x400223FF	1KB	Flash	
0x40020400	–	0x40021FFF	–	–	
0x40020000	–	0x400203FF	1KB	DMA1	
0x40015C00	–	0x40017FFF	–	–	APB4
0x40015400	–	0x400157FF	1KB	I2S	
0x40015000	–	0x400153FF	1KB	LPUART2	
0x40014C00	–	0x40014FFF	1KB	LPUART1	
0x40014800	–	0x40014BFF	1KB	BTIM789	
0x40014400	–	0x400147FF	1KB	BTIM456	
0x40014000	–	0x400143FF	1KB	BTIM123	
0x40013C00	–	0x40013FFF	1KB	UART6	APB3
0x40013800	–	0x40013BFF	1KB	UART1	
0x40013400	–	0x400137FF	1KB	–	
0x40013000	–	0x400133FF	1KB	SPI1	
0x40012C00	–	0x40012FFF	1KB	ATIM	
0x40012800	–	0x40012BFF	1KB	VC	
0x40012400	–	0x400127FF	1KB	ADC	
0x40012000	–	0x400123FF	1KB	LVD	
0x40011400	–	0x40011FFF	–	–	
0x40010C00	–	0x40010FFF	1KB	7816	

0x40010400	–	0x40010BFF	–		APB2
0x40010000	–	0x400103FF	1KB	SYSCFG	
0x40007C00	–	0x40007FFF	1KB	LPTIM1	
0x40007800	–	0x40007BFF	1KB	OPA	
0x40007400	–	0x400077FF	1KB	DAC	
0x40007000	–	0x400073FF	1KB	–	
0x40006C00	–	0x40006FFF	1KB	LPTIM2	
0x40006800	–	0x40006BFF	1KB	I2C4	
0x40006400	–	0x400067FF	1KB	–	
0x40006000	–	0x400063FF	1KB	CRS	
0x40005C00	–	0x40005FFF	1KB	I2C3	
0x40005800	–	0x40005BFF	1KB	I2C2	
0x40005400	–	0x400057FF	1KB	I2C1	
0x40005000	–	0x400053FF	1KB	UART5	
0x40004C00	–	0x40004FFF	1KB	UART4	
0x40004800	–	0x40004BFF	1KB	UART3	
0x40004400	–	0x400047FF	1KB	UART2	
0x40004000	–	0x400043FF	1KB	SPI4	
0x40003C00	–	0x40003FFF	1KB	SPI3	APB1
0x40003800	–	0x40003BFF	1KB	SPI2	
0x40003400	–	0x400037FF	1KB	–	
0x40003000	–	0x400033FF	1KB	IWDT	
0x40002C00	–	0x40002FFF	1KB	WWDG	
0x40002800	–	0x40002BFF	1KB	RTC	
0x40002400	–	0x400027FF	1KB	LCD	
0x40002000	–	0x400023FF	1KB	Reserved	
0x40001C00	–	0x40001FFF	1KB	STIM	
0x40001800	–	0x40001BFF	1KB	PCA	
0x40001000	–	0x400017FF	–	–	
0x40000C00	–	0x40000FFF	1KB	PCNT	
0x40000800	–	0x40000BFF	1KB	GTIM3	
0x40000400	–	0x400007FF	1KB	GTIM2	
0x40000000	–	0x400003FF	1KB	GTIM1	
0x00000000	–	0x001FFFFFFF	256KByte	FLASH	memory
0x00200000	–	0x1FFFFFFF	–	Reserved	
0x20000000	–	0x2003FFFFFF	32KByte	SRAM	
0x20040000	–	0x3FFFFFFF	–	Reserved	

表 7-1 地址划分表

8. 工作模式

本产品的电源管理模块负责管理本产品各种工作模式之间的切换，以及控制各工作模式下的各功能模块的工作状态。本产品的工作电压（VCC）为 1.8v ~ 5.5v。

本产品有如下几个工作模式：

- 1) 运行模式：CPU 运行，周边功能模块运行。
- 2) 休眠模式：CPU 停止运行，周边功能模块运行。
- 3) 深度休眠模式：CPU 停止运行，高速时钟停止运行。

从运行模式，通过执行软件程序，可进入其他低功耗模式。

从其他各种低功耗模式，通过中断触发，可回到运行模式。

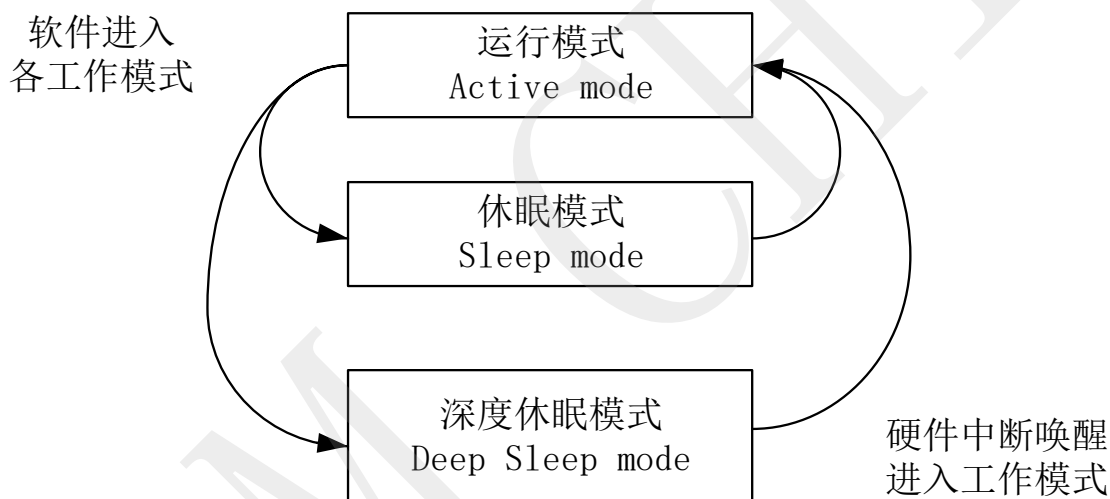


图 8-1 控制模式框图

在各模式下，CPU 可响应所有的中断参考中断章节

在各模式下，本产品可响应所有的复位类型。

	复位源	运行模式	休眠模式	深度休眠模式
[0]	上电掉电复位 POR	V	V	V
[1]	外部 Reset Pin 复位	V	V	V
[2]	LVD 复位	V	V	V
[3]	WDT 复位	V	V	V
[4]	WWDT 复位	V	V	
[5]	Cortex-M0+ LOCKUP 硬件 复位	V		
[6]	Cortex-M0+ SYSRESETREQ 软件复位	V		

8.1 运行模式

本产品运行模式（Active Mode）：

在系统在电源上电复位后，或从各低功耗唤醒后，微控制器 MCU 处于运行状态。

当 CPU 不需继续运行时，可以利用多种低功耗模式来节能，例如等待某个外部事件时。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

几种降低运行模式下芯片功耗的方法：

- 1) 在运行模式下，通过对预分频寄存器（SYSCTRL0.HCLK_PRS, SYSCTRL0.PCLK_PRS）进行编程，可以降低任意一个系统时钟(HCLK, PCLK)的速度。进入睡眠模式前，也可以利用预分频器来降低外设的时钟。
- 2) 在运行模式下，关闭不使用外设的时钟（PERIx_CLKx）来减少功耗。
- 3) 在运行模式下，关闭不使用外设的时钟（PERIx_CLKx）来减少功耗，并让系统进入睡眠模式下更多地减少功耗，并在执行 WFI 指令前关闭不使用外设的时钟（PERIx_CLKEN.x）。
- 4) 使用低功耗模式代替休眠模式，因为本产品的唤醒时间极短（~5μs），亦可满足系统的实时响应的需求。

8.2 休眠模式

本产品休眠模式（Sleep Mode）

使用 WFI 指令可以进入休眠模式，休眠模式下，CPU 停止运行，但时钟模块、系统时钟、NVIC 中断处理以及周边的功能模块仍都可以工作。

系统进入休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

- 如何进入休眠模式：

通过执行 WFI 指令进入睡眠状态。根据 Cortex-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择睡眠模式进入机制：

SLEEP-NOW: 如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入睡眠模式。

SLEEP-ON-EXIT: 如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入睡眠模式。

- 如何退出休眠模式：

如果执行 WFI 指令进入睡眠模式，任意一个高优先级嵌套向量中断控制器响应的外设中断都能将系统从睡眠模式唤醒。

使用注意：

- 1) SLEEP-ON-EXIT 该位置 1，执行完中断自动进入 sleep，程序不需要写 __wfi()；
- 2) SLEEP-ON-EXIT 该位清 0，main() 执行 __wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，执行 WFI 指令后进入 sleeping。等待后续中断触发。
- 3) SLEEP-ON-EXIT 位不影响 __wfi() 指令的执行。SLEEP-ON-EXIT=0: main() 执行 wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，继续往下执行；
- 4) 若在中断中进入 sleep，只有优先级高于此中断的中断才能唤醒，先执行高优先级，再执行低优先级；优先级低于或等于此中断的中断不能唤醒。

8.3 深度休眠模式

本产品深度休眠模式（Deep Sleep Mode）

使用 SLEEPDEEP 配合 WFI 指令可以进入深度休眠模式，在深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟可配置是否运行，部分低功耗的周边模块可配置是否允许，NVIC 中断处理仍可以工作。

- 系统从高速时钟进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度睡眠前的状态。
- 系统从低速时钟进入深度休眠模式，由于低速时钟不会自动关闭，保持运行，进入休眠模式。只有 ARM Cortex-M0+ 不运行，其他模块都运行。
- 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。
- 系统进入深度休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的

状态为休眠下的状态。

如何进入深度休眠模式：

首先设置 Cortex-M0+ 系统控制寄存器中的 SLEEPDEEP 位，通过执行 WFI 指令进入睡眠状态。根据 Cortex™-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择深度睡眠模式进入机制：

SLEEP-NOW: 如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入睡眠模式。

SLEEP-ON-EXIT: 如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入睡眠模式。

如何退出深度休眠模式：

如果执行 WFI 指令进入睡眠模式，任意一个被嵌套向量中断控制器响应的外设中断（Deep Sleep 下可运行的 周边模块中断）都能将系统从睡眠模式唤醒。

唤醒设置参考 4.4 中断唤醒控制 WIC。

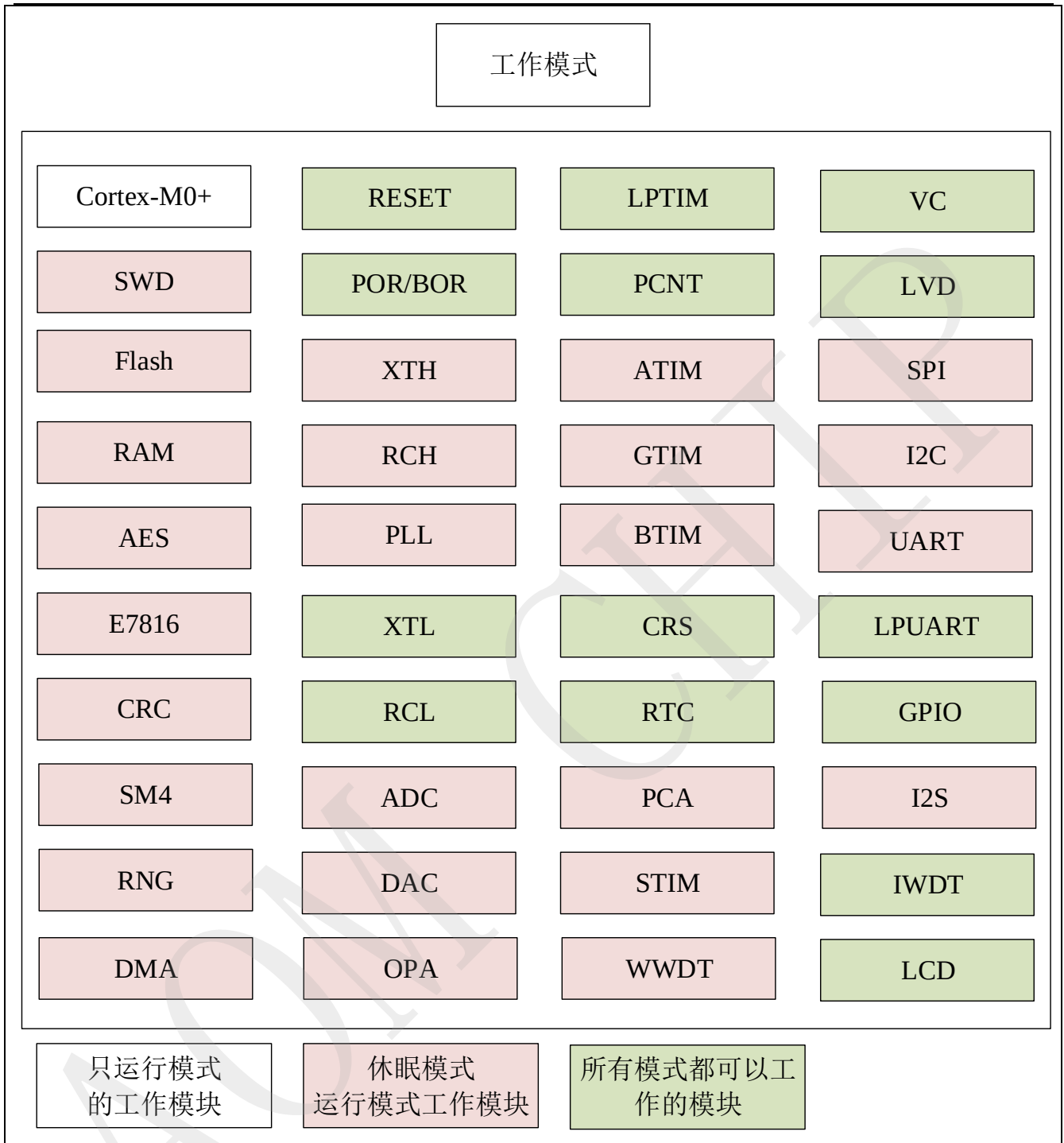


图 8-2 各种模式下可运行模块示意图

系统控制寄存器(Cortex-M0+ 内核系统控制寄存器)

地址: 0xE000ED10

复位值: 0x0000 0000

位	标记	功能描述	读写
31:5	RESERVED	保留	
4	SEVONPEND	设置为1时, 每次新的中断挂起都会产生一个事件, 如果使用了WFE 休眠, 它可用于唤醒处理器	RW
3	RESERVED	保留	
2	SLEEPDEEP	设为1时, 执行WFI进入深度休眠, 本产品进入 Deep sleep模式 设置为0时, 执行WFI进入休眠, 本产品进入 sleep/Idle 模式	RW
1	SLEEPONEXIT	设为1时, 当退出异常处理并返回程序线程时, 处理器自动进入休眠 模式(WFI) 设置为0时, 该特性就会被自动禁止	RW
0	RESERVED	保留	

进入深度休眠后, 唤醒后系统时钟有两种选择, 默认使用进入深度休眠的时钟, 配置寄存器 SYSCTRL0.wakeup_byRCH 为 1 后不管进入深度休眠前是什么时钟, 唤醒后都使用内部高速时钟 RCH。如果使用外部晶体振荡这样设置可以加速唤醒系统。

9. 系统控制器 (SYSCTRL)

9.1 系统时钟介绍

时钟控制模块主要控制系统时钟以及外设时钟, 可以配置不同的时钟源作为系统时钟、可以配置不同的系统时钟分频、可以启动或禁用外设时钟。为了确保振荡器精度, 内部时钟均具有校准功能。

本产品支持以下五个不同的时钟源作为系统时钟:

- 内部高速 RC 时钟 RCH (4M) (默认主频)
- 内部低速 RC 时钟 RCL (38.4K 与 32.8K 可配置)
- 外部高速晶振时钟 XTH
- 外部低速晶振时钟 XTL
- 锁相环时钟 PLL

注1: 切换系统时钟的时钟源时, 请严格按照操作步骤进行切换, 详见 5.2 章节。

注2: XTL 可以不接晶振, 直接从 PC14 管脚输入时钟信号。XTH 可以不接晶振, 直接从 PF00 管脚输入时钟

信号。

本产品还包含以下两个辅助时钟：

- 内部低速 10K 时钟：仅供看门狗和 CLKTRIM 模块使用。
- 内部 150K 时钟：仅供 LVD 和 VC 模块使用。

下图为本产品的时钟架构。

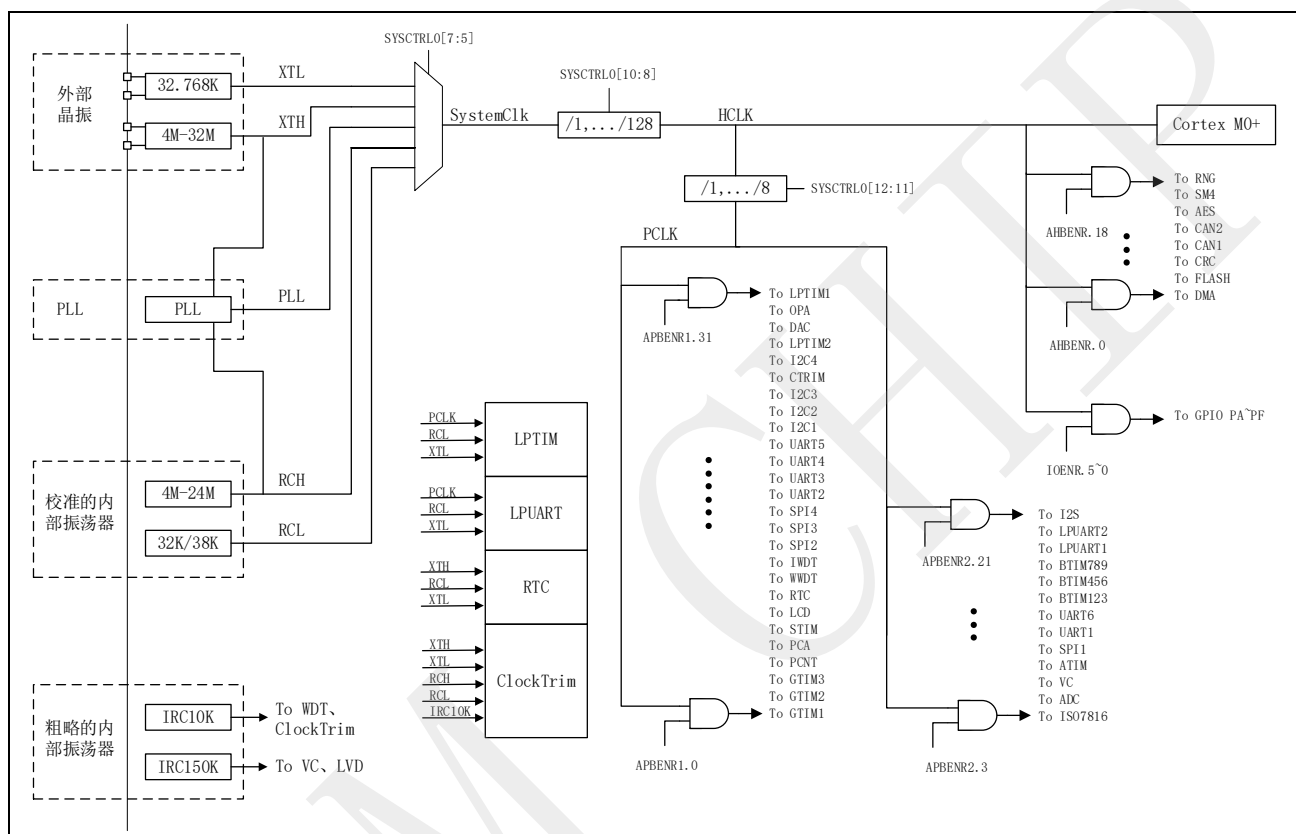


图 9-1 时钟控制模块框图

9.1.1 内部高速 RC 时钟 RCH

芯片上电或复位后的默认时钟源为内部高速时钟，频率为 4MHz。当系统进入 DeepSleep，此高速时钟会自动关闭。内部高速时钟可通过寄存器 RCH_CR[10:0]来选择其输出频率，可供选择的频率为 4MHz、8MHz、16MHz、22.12MHz、24MHz。

内部高速时钟从启动到稳定仅需 5 μ s。为了在深度休眠模式下能快速响应中断，建议进入深度休眠模式前将系统时钟切换为 RCH。

9.1.2 内部低速 RC 时钟 RCL

内部低速时钟可以通过寄存器 RCL_CR[9:0]来选择其输出频率，可供选择的频率为

38.4kHz、32.768kHz。当系统进入 DeepSleep，此低速时钟不会自动关闭，超低功耗外设模块可以选择 RCL 作为其时钟。

9.1.3 外部低速晶振时钟 XTL

外部低速晶振时钟需外接一个 32.768kHz 的低功耗晶振，具有超高精度以及超低功耗。当系统进入 Deep Sleep，此低速时钟不会自动关闭。超低功耗模式下工作的外设模块可以选择 XTL 作为其时钟。

XTL 也可以不接晶振，直接从 PC14 管脚输入时钟信号。从 PC14 输入时钟信号的方法为：配置 PC14 引脚为 GPIO 输入；设置 SYSCTRL1.EXTL_EN 为 1。

9.1.4 外部高速晶振时钟 XTH

外部高速晶振时钟需外接一个 4 MHz ~ 32MHz 的高速晶振。当系统进入 Deep Sleep，此高速时钟会自动关闭。

XTH 也可以不接晶振，直接从 PF00 管脚输入时钟信号。从 PF00 输入时钟信号的方法为：配置 PF00 引脚为 GPIO 输入；设置 SYSCTRL1.EXTH_EN 为 1。

9.1.5 锁相环时钟 PLL

内建 PLL 支持 8M ~ 48M 的时钟输出。PLL 的参考时钟源为：RCH、XTH 晶振时钟、PF00 管脚输入时钟。

9.1.6 时钟启动过程

上述五种时钟源都需要启动稳定时间，下图以外部 XTL 为例说明时钟的启动稳定过程。

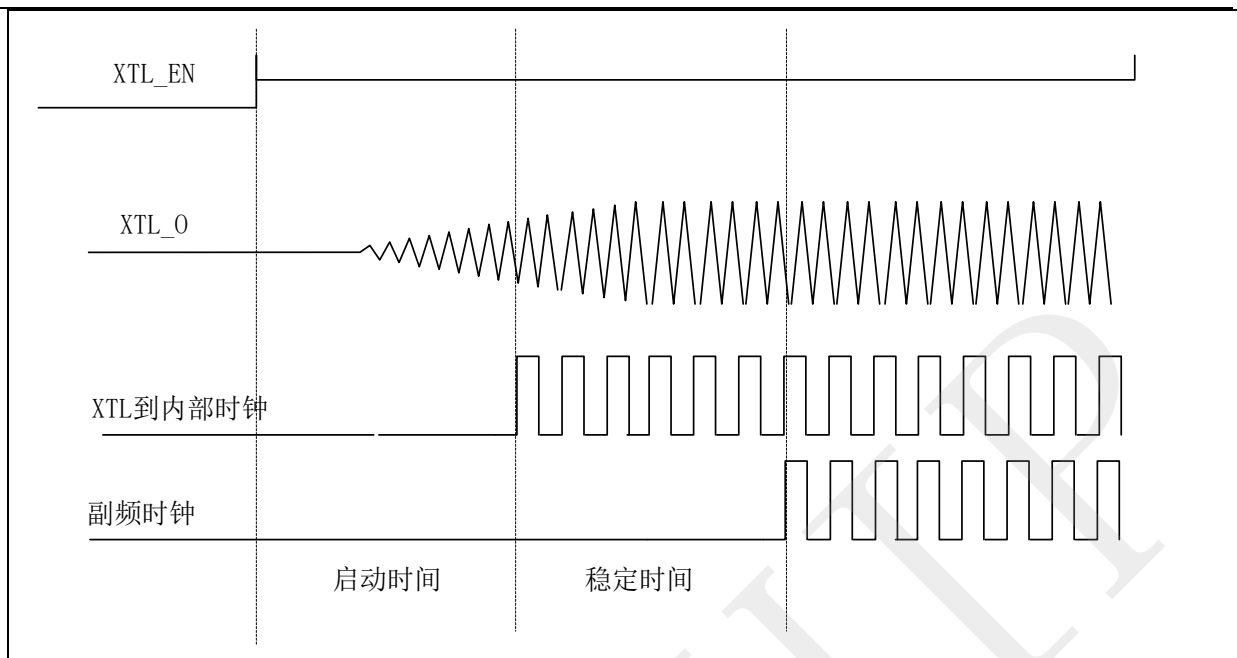


图 9-2 晶振时钟启动示意图

9.2 系统时钟切换

时钟源的切换是由寄存器 `SYSCTRL0[7:0]` 来控制。系统时钟的时钟源可通过 `SYSCTRL0[7:5]` 在 `RCH`、`RCL`、`XTH`、`XTL`、`PLL` 之间进行切换。切换时 `RCH`、`RCL`、`XTH`、`XTL` 这四个时钟源中的任意两者可以相互切换；`PLL` 只可与 `RCH`、`XTH` 这两个时钟进行切换。时钟切换操作必须按照下文所描述的 7 种时钟切换流程进行，否则可能出现异常。

时钟切换时需要同步配置 `FLASH_CR.WAIT`。时钟频率不大于 24MHz 则应设置 `FLASH_CR.WAIT` 为 0；时钟频率大于 24MHz 则应设置 `FLASH_CR.WAIT` 为 1；时钟频率大于 48MHz 则应设置 `FLASH_CR.WAIT` 为 2。

注：设置 `FLASH_CR.WAIT` 的值，需要先向 `FLASH_BYPASS` 寄存器依次写入 `0x5A5A`、`0x5A5A` 然后再对 `FLASH_CR.WAIT` 进行赋值，详见 `FLASH` 控制器章节。

9.2.1 标准的时钟切换流程

操作流程如下：

Step1：如新时钟源需要外部引脚，则将该引脚设置为适当的模式。

注：接外部晶振时需要模拟引脚；接外部时钟输入时需要 `GPIO` 输入并使能外部时钟输入。

Step2：配置新时钟源的的振荡参数。

Step3: 使能新时钟源的振荡器。

Step4: 根据当前时钟源和新时钟源两者中较高的频率, 按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step5: 等待新时钟源输出稳定的频率。

Step6: 配置 SYSCTRL0.CLKSW, 选择系统时钟的来源为新时钟源。

Step7: 根据新时钟源的频率, 按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step8: 关闭不再使用的时钟源。

9.2.2 从其它时钟切换到 XTL 示例

操作流程如下:

Step1: 设置 PCADS[14]及 PCADS.15 为 1, 配置 PC14/PC15 引脚为模拟端口。

Step2: 根据晶振特性, 配置 XTL_CR[5:0]。

Step3: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step4: 设置 SYSCTRL0.XTL_EN 为 1, 使能晶振振荡电路。

Step5: 查询等待 XTL_CR.Stable 标志变为 1, 晶振输出稳定时钟。

Step6: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step7: 设置 SYSCTRL0.CLKSW 为 3, 将系统时钟切换为 XTL。

Step8: 按 Flash 控制器章节流程设置 FLASH_CR.WAIT 为 0。

Step9: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step10: 设置 SYSCTRL0.xxx_EN 为 0, 关闭原时钟。

9.2.3 从其它时钟切换到 XTH 示例

Step1: 设置 PFADS[0]及 PFADS[1]为 1, 配置 PF00/PF01 引脚为模拟端口。

Step2: 根据晶振特性, 配置 XTH_CR[3:0]。

Step3: 设置 XTH_CR.Startup 为 3, 选择最长的晶振稳定时间。

Step4: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step5: 设置 SYSCTRL0.XTH_EN 为 1, 使能晶振振荡电路。

Step6: 根据当前时钟和 XTH 两者中较高的频率, 按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

- Step7: 查询等待 XTH_CR.Stable 标志变为 1, 晶振输出稳定时钟。
- Step8: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step9: 设置 SYSCTRL0.CLKSW 为 1, 将系统时钟切换为 XTH。
- Step10: 根据 XTH 的频率, 按 Flash 控制器章节流程配置 FLASH_CR.WAIT。
- Step11: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step12: 设置 SYSCTRL0.xxx_EN 为 0, 关闭原时钟。

9.2.4 从其它时钟切换到 RCL 示例

操作流程如下:

- Step1: 配置 RCL_CR.TRIM 及 RCL_CR.Startup。
- Step2: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step3: 设置 SYSCTRL0.RCL_EN 为 1, 使能 RCL 振荡电路。
- Step4: 查询等待 RCL_CR.Stable 标志变为 1, RCL 输出稳定时钟。
- Step5: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step6: 设置 SYSCTRL0.CLKSW 为 2, 将系统时钟切换为 RCL。
- Step7: 按 Flash 控制器章节流程设置 FLASH_CR.WAIT 为 0。
- Step8: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step9: 设置 SYSCTRL0.xxx_EN 为 0, 关闭原时钟。

9.2.5 从其它时钟切换到 RCH 示例

操作流程如下:

- Step1: 配置 RCH_CR.TRIM。
- Step2: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step3: 设置 SYSCTRL0.RCH_EN 为 1, 使能 RCH 振荡电路。
- Step4: 查询等待 RCH_CR.Stable 标志变为 1, RCH 输出稳定时钟。
- Step5: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。
- Step6: 设置 SYSCTRL0.CLKSW 为 0, 将系统时钟切换为 RCH。
- Step7: 按 Flash 控制器章节流程设置 FLASH_CR.WAIT 为 0。
- Step8: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step9: 设置 SYSCTRL0.xxx_EN 为 0，关闭原时钟。

9.2.6 PLL 与 RCH 相互切换示例，参考时钟为 RCH

从 RCH 切换到 PLL 操作流程如下：

Step1: 设置 PLL_CR.REFSEL 为 3，选择 PLL 时钟源为 RCH。

Step2: 根据 RCH 频率，配置 PLL_CR.FRSEL。

Step3: 根据 PLL 输出频率，配置 PLL_CR.DIVN 及 PLL_CR.FOSC。

Step4: 设置 PLL_CR.Startup 为 7，选择最长的 PLL 稳定时间。

Step5: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 SYSCTRL0.PLL_EN 为 1，使能 PLL 振荡电路。

Step7: 根据 PLL 输出频率，按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step8: 查询等待 PLL_CR.Stable 标志变为 1，PLL 输出稳定时钟。

Step9: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10: 设置 SYSCTRL0.CLKSW 为 4，将系统时钟切换为 PLL。

警告：切换过程中不可以更改 RCH_CR.TRIM。

从 PLL 切换到 RCH 操作流程如下：

Step1: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2: 设置 SYSCTRL0.CLKSW 为 0，将系统时钟切换为 RCH。

Step3: 按 Flash 控制器章节流程设置 FLASH_CR.WAIT 为 0。

Step4: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5: 设置 SYSCTRL0.PLL_EN 为 0，关闭 PLL。

警告：切换过程中不可以更改 RCH_CR.TRIM。

9.2.7 PLL 与 XTH 相互切换示例，参考时钟为 XTH

从 XTH 切换到 PLL 操作流程如下：

Step1: 设置 PLL_CR.REFSEL 为 0，选择 PLL 时钟源为 XTH。

Step2: 根据 XTH 频率，配置 PLL_CR.FRSEL。

Step3: 根据 PLL 输出频率，配置 PLL_CR.DIVN 及 PLL_CR.FOSC。

Step4: 设置 PLL_CR.Startup 为 7，选择最长的 PLL 稳定时间。

Step5: 向 SYCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 SYCTRL0.PLL_EN 为 1，使能 PLL 振荡电路。

Step7: 根据 PLL 输出频率，按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step8: 查询等待 PLL_CR.Stable 标志变为 1，PLL 输出稳定时钟。

Step9: 向 SYCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10: 设置 SYCTRL0.CLKSW 为 4，将系统时钟切换为 PLL。

从 PLL 切换到 XTH 操作流程如下：

Step1: 向 SYCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2: 设置 SYCTRL0.CLKSW 为 1，将系统时钟切换为 XTH。

Step3: 根据 XTH 的频率，按 Flash 控制器章节流程配置 FLASH_CR.WAIT。

Step4: 向 SYCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5: 设置 SYCTRL0.PLL_EN 为 0，关闭 PLL。

下图为时钟切换时序图：

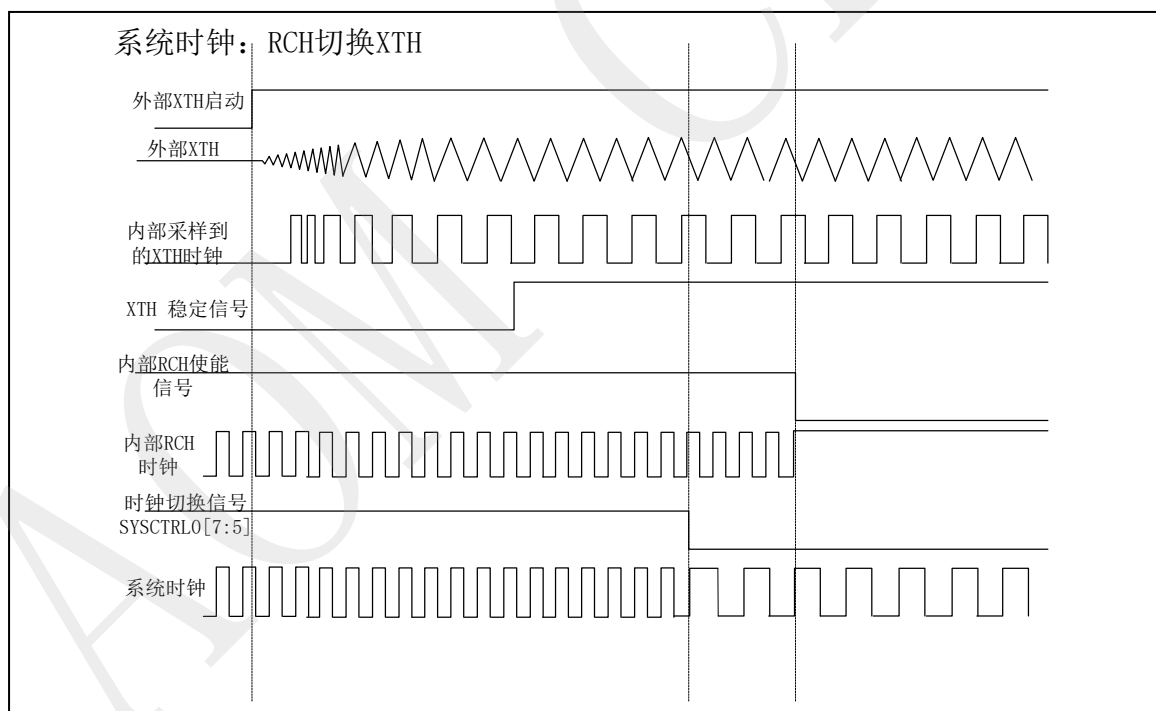


图 9-3 时钟切换示意图

9.3 时钟校准模块

本产品内嵌时钟校准电路，时钟校准参考 CRS 时钟校准章节

9.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠状态时，会停止执行指令。在休眠状态下发生了中断请求（更高优先级）且需要处理时，处理器就会被唤醒。

休眠状态下的处理器收到中断请求时的行为如下表所示：

PRIMASK 状态	WFI 行为	唤醒	ISR 执行
0	IRQ 优先级 > 当前等级	Y	Y
0	IRQ 优先级 ≤ 当前等级	N	N
1	IRQ 优先级 > 当前等级	Y	N
1	IRQ 优先级 ≤ 当前等级	N	N

9.4.1 从深度休眠模式唤醒后执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 执行 WFI 指令以进入深度休眠模式
5. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务程序

例程：

```
SCB->SCR |= 0x00000004;
while(1)
{
    __asm("WFI");
}
```

9.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 PRIMASK 为 1
4. 设置 SCB->SCR.SLEEPDEEP 为 1

5. 执行 WFI 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令
7. 清除中断标志，清除中断挂起状态
8. 执行用户定义的操作

例程：

```
__asm("CPSID I");    //Set PRIMASK

SCB->SCR |= 0x00000004u;

while(1)
{
    __asm("WFI");

    CTIMERLP_REG->TFCR_f.TFC=0;    //Clear Int Flag

    NVIC_ClearPendingIRQ(TIMERLP_IRQn); //Clear Pending Flag

    ...    //执行用户定义的操作
}
```

9.4.3 使用退出休眠特性

退出休眠（sleep-on-exit）非常适合中断驱动的应用程序。当该特性使能时，只要完成异常处理并且返回到了线程模式，处理器就会进入休眠模式。利用退出休眠特性，处理器可以尽可能多的处于休眠模式。

Cortex-M0 利用退出休眠特性进入休眠，这种情况同执行完异常退出后立即执行 WFI 的效果差不多。不过，为了下次进入异常时，不用再进行压栈操作，处理器不会执行出栈的过程。

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 设置 SCB->SCR.SLEEPONEXIT 为 1
5. 执行 WFI 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序

7. 退出中断服务时自动进入休眠模式

例程:

```
SCB->SCR |= 0x00000004;
```

```
SCB->SCR |= 0x00000002;
```

```
while(1)
```

```
{
```

```
    __asm("WFI");
```

```
}
```

9.5 寄存器

基地址 0x40010000

寄存器	偏移地址	描述
SYSCTRL0	0x000	系统控制寄存器0
SYSCTRL1	0x004	系统控制寄存器1
SYSCTRL2	0x008	系统控制寄存器2
RCH_CR	0x00C	RCH 控制寄存器
XTH_CR	0x010	XTH 控制寄存器
RCL_CR	0x014	RCL 控制寄存器
XTL_CR	0x018	XTL 控制寄存器
PLL_CR	0x03C	PLL 控制寄存器
MCO_CR	0x05C	时钟输出控制寄存器
IOENR	0x080	端口时钟控制寄存器
AHBENR	0x084	AHB外设时钟控制寄存器
APBENR1	0x088	APB外设时钟控制寄存器1
APBENR2	0x08C	APB外设时钟控制寄存器2
CIER	0x090	时钟侦测中断使能
CIFR	0x094	时钟侦测标志
CICR	0x098	时钟侦测中断清除
MUXCAP	0x09C	端口复用捕获选择寄存器
MUXGCAP	0x0A0	端口复用捕获选择寄存器
MUXETR	0x0A4	端口复用ETR输入选择寄存器
MUXGETR	0x0A8	端口复用ETR输入选择寄存器
MUXGATE	0x0AC	端口复用GATE输入选择寄存器

表 9-1 系统控制寄存器表格

9.5.1 系统控制寄存器 0 (SYSCTRL0)

偏移地址：0x000

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
wakeup_byRCH	Reserved			PCLK_PRS		HCLK_PRS									
RW				RW		RW									
7	6	5	4	3	2	1	0								
CLKSW			PLL_EN	XTL_EN	RCL_EN	XTH_EN	RCH_EN								
RW			RW	RW	RW	RW	RW								

位	标记	功能描述
31:16	Reserved	保留
15	wakeup_byRCH	1: 从 Deep Sleep 唤醒后, system clock来源为RCH, 原时钟继续使能。 0: 从 Deep Sleep 唤醒后, 不改变system clock来源。
14:13	Reserved	保留, 禁止修改
12:11	PCLK_PRS	PCLK时钟来源选择 00: HCLK 01: HCLK/2 10: HCLK/4 11: HCLK/8
10:8	HCLK_PRS	HCLK时钟来源选择 000: SystemClk 001: SystemClk/2 010: SystemClk/4 011: SystemClk/8 100: SystemClk/16 101: SystemClk/32 110: SystemClk/64 111: SystemClk/128
7:5	CLKSW	SystemClk时钟来源选择 000: 内部高速时钟 RCH 001: 外部高速晶振 XTH 010: 内部低速时钟 RCL 011: 外部低速晶振 XTL

		100: 内部 PLL
4	PLL_EN	PLL使能控制 0: 关闭 1: 使能
3	XTL_EN	外部低速晶振 XTL 使能控制 0: 关闭 1: 使能 注: 需要将PC14、PC15设置成模拟端口。 当SYSCTRL1.XTL_ALWAYS_ON=1, 一旦打开无法关闭
2	RCL_EN	内部低速时钟 RCL 使能控制 0: 关闭 1: 使能
1	XTH_EN	外部高速晶振 XTH 使能控制 0: 关闭 1: 使能 注: 当系统进入DeepSleep, 此高速时钟会自动关闭
0	RCH_EN	内部高速时钟 RCH 使能信号。 0: 关闭 1: 使能 注: 当系统进入DeepSleep, 此高速时钟会自动关闭。

注意:

- 每次改写 SYSCTRL0, SYSCTRL1 的值, 均需要先对 SYSCTRL2 依次写入 0x5A5A、0xA5A5。这样的步骤可有效防止对 SYSCTRL0, SYSCTRL1 寄存器的误操作。

9.5.2 系统控制寄存器 1 (SYSCTRL1)

偏移地址: 0x004

复位值: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved				RTC_FREQ_ADJUST				SWD_USE_IO							
				RW				RW							
7	6	5	4	3	2	1	0								
RSV	LOCKUP_EN	RTC_LPW	RSV	XTL_ALWAYS_ON	EXTL_EN	EXTH_EN	RSV								
	RW	RW		RW	RW	RW									

位	标记	功能描述
31:12	Reserved	保留位
11:9	RTC_FREQ_ADJUST	RTC 高速时钟补偿时钟频率选择 000 4M; 001 6M; 010 8M; 011 12M 100 16M; 101 20M; 110 24M; 111 32M;
8	SWD_USE_IO	SWD 端口功能配置 0: SWD 端口 1: GPIO 端口
7	RSV	保留位, 禁止修改
6	LOCKUP_EN	Cortex-M0+ LockUp 功能配置 0: 关闭 1: 使能 注: 使能后, CPU读到无效指令时会复位MCU, 可增强系统可靠性。
5	RTC_LPW	RTC 模块低功耗控制 1: 低功耗模式使能 0: 低功耗模式禁能 注: 使能后, RTC 模块进低功耗状态, 其寄存器不可以读写。
4	RSV	保留位, 禁止修改
3	XTL_ALWAYS_ON	XTL 高级使能控制 1: SYSCTRL0.XTL_EN 只可置位。 0: SYSCTRL0.XTL_EN 可置位可清零。
2	EXTL_EN	外部 XTL 时钟输入控制 1: XTL 输出时钟从 PC14 输入。 0: XTL 输出时钟由晶振产生。

		注：使用PC14 输入时钟时，需设置SYSCTRL0.XTL_EN为1。
1	EXTN_EN	外部 XTH 输入控制 1：XTH 输出时钟从 PF00输入。 0：XTH 输出时钟由晶振产生。 注：使用PF00 输入时钟时，需设置SYSCTRL0.XTH_EN为1。
0	Reserved	保留位，禁止修改

9.5.3 系统控制寄存器 2（SYSCTRL2）

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCTRL2															
WO															

位	标记	功能描述
31:16	Reserved	保留位
15:0	SYSCTRL2	寄存器 SYSCTRL0, SYSCTRL1 保护系列控制寄存器，对 SYSCTRL2 先写 0x5A5A，再写 0xA5A5，启动对于寄存器 SYSCTRL0, SYSCTRL1 的写操作，只要对寄存器 SYSCTRL0, SYSCTRL1 写操作了，这个保护位自动回复保护状态，需要重新写入系列打开保护。

9.5.4 RCH 控制寄存器 (RCH_CR)

偏移地址: 0x00C

复位值: 0x0000 0926

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				stable	TRIM										
				RO	RW										

位	标记	功能描述
31:12	Reserved	保留位
11	stable	RCH 时钟稳定标志位。 1: 代表 RCH 已经稳定, 可以被内部电路使用。 0: 代表 RCH 未稳定, 不可以被内部电路使用。
10:0	TRIM	时钟频率调整, Flash中保存了5组频率的校准值。 将Flash内的校准值读出并写入RCH_CR. TRIM即可获得精准的频率。 24M校准值地址: 0x00100A08 - 0x00100A09 22.12M校准值地址: 0x00100A06 - 0x00100A07 16M校准值地址: 0x00100A04 - 0x00100A05 8M校准值地址: 0x00100A02- 0x00100A03 4M校准值地址: 0x00100A00 - 0x00100A01

9.5.5 XTH 控制寄存器 (XTH_CR)

偏移地址: 0x010

复位值: 0x0000 7022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
xth_fsel_unstable		Driver_unstable				NF_20ENB		Reserved				TS			
RW		RW				RW						RW			
7		6		5		4		3		2		1		0	
stable		CCS		Startup				xth_fsel				Driver			
RO		RW		RW				RW				RW			

位	标记	功能描述
31:16	Reserved	保留位
15:14	xth_fsel_unstable	stable前外部晶振工作频率选择 11: 20M~32M 10: 12M~20M 01: 6M~12M 00: 4M~6M
13:12	Driver_unstable	stable前晶振驱动能力选择 11: 最强驱动能力 10: 默认驱动能力 (推荐值) 01: 弱驱动能力 00: 最弱驱动能力
11	NF_20ENB	20ns noise filter 使能控制 0: 使能 1: 关闭
10:9	Reserved	保留位, 禁止修改
8	TS	晶振时钟侦测时间间隔选择 0: 2ms 1: 8ms
7	stable	外部高速时钟 XTH 稳定标志位。 1: 代表 XTH 已经稳定, 可以被内部电路使用。 0: 代表 XTH 未稳定, 不可以被内部电路使用。
6	CCS	晶振时钟侦测使能控制 1: 使能

		0: 关闭
5:4	Startup	外部高速时钟 XTH 稳定时间选择 00: 256个周期; 01: 1024个周期; 10: 4096个周期; 11: 16384个周期; 注: 强烈建议将 XTH 的稳定时间设置为 11。如果 XTH 稳定时间不足, 在进行时钟切换时或从深度休眠唤醒时, 系统不能稳定工作。
3:2	xth_fsel	外部晶振工作频率选择 11: 20M~32M 10: 12M~20M 01: 6M~12M 00: 4M~6M
1:0	Driver	外部晶振驱动能力选择 11: 最强驱动能力 10: 默认驱动能力 (推荐值) 01: 弱驱动能力 00: 最弱驱动能力 注: 需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大; 驱动能力越弱则功耗越小。

9.5.6 RCL 控制寄存器 (RCL_CR)

偏移地址: 0x014

复位值: 0x0000 033F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			stable	Startup	TRIM										
			RO	RW	RW										

位	标记	功能描述
31:13	Reserved	保留位
12	stable	内部低速时钟 RCL 稳定标志位。 1: 代表 RCL 已经稳定, 可以被内部电路使用。 0: 代表 RCL 未稳定, 不可以被内部电路使用。
11:10	Startup	内部低速时钟 RCL 稳定时间选择 11: 256个周期; 10: 64个周期;

		01: 16个周期; 00: 4个周期;
9:0	TRIM	内部低速时钟频率调整, Flash中保存了2组频率的校准值。 将Flash内的校准值读出并写入RCL_CR. TRIM即可获得精准的频率。 38.4K校准值地址: 0x00100A0C - 0x00100A0D 32.768K校准值地址: 0x00100A0A - 0x00100A0B

9.5.7 XTL 控制寄存器 (XTL_CR)

偏移地址: 0x018

复位值: 0x0000 6021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
amp_sel_unstable		Driver_unstable				NF_20ENB		Reserved				TS			
RW		RW				RW						RW			
7		6		5		4		3		2		1		0	
stable		CCS		Startup				amp_sel				Driver			
RO		RW		RW				RW				RW			

位	标记	功能描述
31:16	Reserved	保留位
15:14	amp_sel_unstable	stable前 XTL晶振振荡幅度调整 11: 最大振幅 10: 较大振幅 (推荐值) 01: 正常振幅 00: 最小振幅
13:12	Driver_unstable	stable前 XTL晶振驱动能力选择 11: 最强驱动能力 10: 较强驱动能力 01: 一般驱动能力 (推荐值) 00: 最弱驱动能力
11	NF_20ENB	20ns noise filter 使能控制 0: 使能 1: 关闭
10:9	Reserved	保留位, 禁止修改
8	TS	晶振时钟侦测时间间隔选择 0: 2ms

		1: 8ms
7	stable	外部低速晶振 XTL 稳定标志位。 1: 代表 XTL 已经稳定, 可以被内部电路使用。 0: 代表 XTL 未稳定, 不可以被内部电路使用。
6	CCS	晶振时钟侦测使能控制 1: 使能 0: 关闭
5:4	Startup	外部低速晶振 XTL 稳定时间选择 00: 256 个周期; 01: 1024 个周期; 10: 4096 个周期; 11: 16384 个周期;
3:2	amp_sel	XTL晶振振荡幅度的调整 11: 最大振幅 10: 较大振幅 (推荐值) 01: 正常振幅 00: 最小振幅
1:0	Driver	XTL晶振驱动能力选择 11: 最强驱动能力 10: 较强驱动能力 01: 一般驱动能力 (推荐值) 00: 最弱驱动能力 注: 需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大; 驱动能力越弱则功耗越小。

9.5.8 PLL 控制寄存器 (PLL_CR)

偏移地址: 0x03C

复位值: 0x0001 0B0F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													stable	Startup	
													RO	RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Startup	FRSEL	LFSEL	IBSEL	DVIN				FOSC				REFSEL			
RW	RW	RW	RW	RW				RW				RW			

位	标记	功能描述
31:19	Reserved	保留
18	stable	PLL 稳定标志

		0: PLL未稳定 1: PLL已经稳定
17:15	Startup	PLL 稳定时间选择 000: 128个PLL周期 001: 256个PLL周期 010: 512个PLL周期 011: 1024个PLL周期 100: 2048个PLL周期 101: 4096个PLL周期 110: 8192个PLL周期 111: 16384个PLL周期
14:13	FRSEL	PLL 输入频率选择, 根据PLL输入时钟的频率进行如下配置 00: 4M~6M 01: 6M~12M 10: 12M~20M 11: 20M~24M
12:11	LFSEL	PLL 滤波器控制位, 请保持默认值
10:9	IBSEL	PLL 偏置电流选择, 请保持默认值
8:5	DIVN	PLL 输出时钟的倍频系数 PLL 输出频率与输入频率的倍数 输出频率 = DIVN*输入频率 DIVN 允许范围0X2~0XC
4:2	FOSC	PLL 输出频率范围选择, 根据PLL输出时钟频率进行如下配置 000: 8M~12M 001: 12M~18M 010: 18M~24M 011: 24M~36M 1xx: 36M~48M
1:0	REFSEL	输入时钟选择 x0: XTH晶振生成的时钟 01: XTH从管脚PF00输入的时钟。(详见5.1.4) 11: RCH时钟

9.5.9 时钟管脚输出选择控制 (MCO_CR)

偏移地址: 0x05C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	MCO_S	RSV	MCO_DIV
	RW		RW

位	标记	功能描述
31:7	Reserved	保留
6:4	MCO_S	MCO管脚时钟选择输出 000 RCH 001 RCL 010 XTH 011 XTL 100 PLL 101 SCLK 110 HCLK 111 PCLK
3	Reserved	保留
2:0	MCO_DIV	MCO管脚时钟输出除频配置 000 无输出 001 CLKS/1 010 CLKS/2 011 CLKS/4 100 CLKS/8 101 CLKS/16 110 CLKS/32 111 CLKS/64

9.5.10 IO 时钟使能控制寄存器 (IOENR)

偏移地址: 0x080

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										PF	PE	PD	PC	PB	PA
										RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:6	Reserved	保留
5	PF	端口PF时钟使能控制 1: 时钟使能 0: 无效

4	PE	端口PE时钟使能控制 1: 时钟使能 0: 无效
3	PD	端口PD时钟使能控制 1: 时钟使能 0: 无效
2	PC	端口PC时钟使能控制 1: 时钟使能 0: 无效
1	PB	端口PB时钟使能控制 1: 时钟使能 0: 无效
0	PA	端口PA时钟使能控制 1: 时钟使能 0: 无效

9.5.11 AHB 外设时钟使能控制寄存器（AHBENR）

偏移地址：0x084

复位值：0x0000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													RNG	SM4	AES
													RW	RW	RW
15		14		13		12		11		10		9		8	
CAN2		CAN1		RSV		CRC		RSV						EFLASH	
RW		RW				RW								RW	
7		6		5		4		3		2		1		0	
Reserved													DMA		
													RW		

位	标记	功能描述
31:19	Reserved	保留
18	RNG	外设RNG时钟使能控制 1: 使能 0: 禁止
17	SM4	外设SM4时钟使能控制 1: 使能 0: 禁止
16	AES	外设AES时钟使能控制 1: 使能 0: 禁止
15	CAN2	外设CAN2时钟使能控制 1: 使能 0: 禁止
14	CAN1	外设CAN1时钟使能控制 1: 使能 0: 禁止
13	Reserved	保留

12	CRC	外设CRC时钟使能控制 1: 使能 0: 禁止
11:9	Reserved	保留
8	EFLASH	外设Flash时钟使能控制 1: 使能 0: 禁止
7:1	Reserved	保留
0	DMA	外设DMA时钟使能控制 1: 使能 0: 禁止

9.5.12 APB1 外设时钟使能控制寄存器 (APBENR1)

偏移地址: 0x088

复位值: 0x0000 0000

31	30	29	28	27	26	25	24
LPTIM1	OPA	DAC	Reserved	LPTIM2	I2C4	Reserved	CTRIM
RW	RW	RW		RW	RW		RW
23	22	21	20	19	18	17	16
I2C3	I2C2	I2C1	UART5	UART4	UART3	UART2	SPI4
RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
SPI3	SPI2	Reserved	IWDT	WWDT	RTC	LCD	Reserved
RW	RW		RW	RW	RW	RW	
7	6	5	4	3	2	1	0
STIM	PCA	Reserved	Reserved	PCNT	GTIM3	GTIM2	GTIM1
RW	RW			RW	RW	RW	RW

位	标记	功能描述
31	LPTIM1	外设LPTIM1时钟使能控制 1: 使能 0: 禁止
30	OPA	外设OPA时钟使能控制 1: 使能 0: 禁止
29	DAC	外设DAC时钟使能控制 1: 使能 0: 禁止
28	Reserved	保留
27	LPTIM2	外设LPTIM2时钟使能控制 1: 使能 0: 禁止
26	I2C4	外设I2C4时钟使能控制 1: 使能 0: 禁止

25	Reserved	保留
24	CTRIM	外设clock trimming时钟使能控制 1: 使能 0: 禁止
23	I2C3	外设I2C3时钟使能控制 1: 使能 0: 禁止
22	I2C2	外设I2C2时钟使能控制 1: 使能 0: 禁止
21	I2C1	外设I2C1时钟使能控制 1: 使能 0: 禁止
20	UART5	外设UART5时钟使能控制 1: 使能 0: 禁止
19	UART4	外设UART4时钟使能控制 1: 使能 0: 禁止
18	UART3	外设UART3时钟使能控制 1: 使能 0: 禁止
17	UART2	外设UART2时钟使能控制 1: 使能 0: 禁止
16	SPI4	外设SPI4时钟使能控制 1: 使能 0: 禁止
15	SPI3	外设SPI3时钟使能控制 1: 使能 0: 禁止
14	SPI2	外设SPI2时钟使能控制 1: 使能 0: 禁止
13	Reserved	保留
12	IWDT	外设IWDT时钟使能控制 1: 使能 0: 禁止
11	WWDT	外设WWDT时钟使能控制 1: 使能 0: 禁止
10	RTC	外设RTC时钟使能控制 1: 使能 0: 禁止
9	LCD	外设LCD时钟使能控制 1: 使能 0: 禁止
8	Reserved	保留
7	STIM	外设STIM时钟使能控制 1: 使能 0: 禁止
6	PCA	外设PCA时钟使能控制 1: 使能 0: 禁止
5:4	Reserved	保留
3	PCNT	外设PCNT时钟使能控制 1: 使能 0: 禁止
2	GTIM3	外设GTIM3时钟使能控制

		1: 使能 0: 禁止
1	GTIM2	外设GTIM2时钟使能控制 1: 使能 0: 禁止
0	GTIM1	外设GTIM1时钟使能控制 1: 使能 0: 禁止

9.5.13 APB2 外设时钟使能控制寄存器（APBENR2）

偏移地址：0x08C

复位值：0x0200 0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		I2S	LPUART2	LPUART1	BTIM789	BTIM456	BTIM123
		RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
UART6	UART1	RSV	SPI1	ATIM	VC	ADC	Reserved
RW	RW		RW	RW	RW	RW	
7	6	5	4	3	2	1	0
Reserved				ISO7816	Reserved		
				RW			

位	标记	功能描述
31:22	Reserved	保留
21	I2S	外设I2S时钟使能控制 1: 使能 0: 禁止
20	LPUART2	外设LPUART2时钟使能控制 1: 使能 0: 禁止
19	LPUART1	外设LPUART1时钟使能控制 1: 使能 0: 禁止
18	BTIM789	外设BTIM789时钟使能控制 1: 使能 0: 禁止
17	BTIM456	外设BTIM456时钟使能控制 1: 使能 0: 禁止
16	BTIM123	外设BTIM123时钟使能控制 1: 使能 0: 禁止

15	UART6	外设UART6时钟使能控制 1: 使能 0: 禁止
14	UART1	外设UART1时钟使能控制 1: 使能 0: 禁止
13	Reserved	保留
12	SPI1	外设SPI1时钟使能控制 1: 使能 0: 禁止
11	ATIM	外设ATIM时钟使能控制 1: 使能 0: 禁止
10	VC	外设VC比较器时钟使能控制 1: 使能 0: 禁止
9	ADC	外设ADC时钟使能控制 1: 使能 0: 禁止
8:4	Reserved	保留
3	ISO7816	外设智能卡7816接口时钟使能控制 1: 使能 0: 禁止
2:0	Reserved	保留

9.5.14 时钟侦测中断使能（CIER）

偏移地址：0x090

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						LCCS	CCS	LFAIL	FAIL	Reserved					
						RW	RW	RW	RW						

位	标记	功能描述
31:10	Reserved	保留
9	LCCS	XTL时钟工作失效检测中断使能 1:使能 0:无效
8	CCS	XTH时钟工作失效检测中断使能 1:使能 0:无效
7	LFAIL	XTL时钟起振失效检测中断使能 1:使能 0:无效
6	FAIL	XTH时钟起振失效检测中断使能 1:使能 0:无效
5:0	Reserved	保留

9.5.15 时钟侦测标志（CIFR）

偏移地址：0x094

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						LCCS	CCS	LFAIL	FAIL	Reserved					
						R	R	R	R						

位	标记	功能描述
31:10	Reserved	保留
9	LCCS	XTL时钟工作失效检测标志 1：起振之后，正常工作下，工作失效标志 0：工作正常标志
8	CCS	XTH时钟工作失效检测标志 1：起振之后，正常工作下，工作失效标志 0：工作正常标志
7	LFAIL	XTL时钟起振失效检测标志 1：起振时，起振失效标志 0：起振正常标志
6	FAIL	XTH时钟起振失效检测标志 1：起振时，起振失效标志 0：起振正常标志
5:0	Reserved	保留

9.5.16 时钟侦测中断清除（CICR）

偏移地址：0x098

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						LCCS	CCS	LFAIL	FAIL	Reserved					
						W1	W1	W1	W1						

位	标记	功能描述

31:10	Reserved	保留
9	LCCS	XTL时钟工作失效检测标志清除；写1清除，写0无效。
8	CCS	XTH时钟工作失效检测标志清除；写1清除，写0无效。
7	LFAIL	XTL时钟起振失效检测标志清除；写1清除，写0无效。
6	FAIL	XTH时钟起振失效检测标志清除；写1清除，写0无效。
5:0	Reserved	保留

9.5.17 Pinmux 辅助选择 1 (MUXCAP)

偏移地址：0x09C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	LPUART1_RX		RSV	UART5_RX		RSV	STIM_CH4		RSV	ATIM_CH0B		RSV			
	RW			RW			RW			RW			RW		

详细参考下文选择表格

注：Pinmux 辅助寄存器写需要任何一个 port 时钟使能，任何 port 复位均复位此寄存器。

9.5.18 Pinmux 辅助选择 2 (MUXGCAP)

偏移地址：0x0A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	GTIM3_CH2		RSV	GTIM3_CH1		RSV	GTIM2_CH1		RSV	GTIM1_CH1		RSV			
	RW			RW			RW			RW			RW		

详细参考下文选择表格

注：Pinmux 辅助寄存器写需要任何一个 port 时钟使能，任何 port 复位均复位此寄存器。

9.5.19 Pinmux 辅助选择 3 (MUXETR)

偏移地址：0x0A4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	BTIM456_ETR			RSV	BTIM123_ETR			RSV	STIM_ETR			RSV	ATIM_ETR		
	RW				RW				RW				RW		

详细参考下文选择表格

注：Pinmux 辅助寄存器写需要任何一个 port 时钟使能，任何 port 复位均复位此寄存器。

9.5.20 Pinmux 辅助选择 4 (MUXGETR)

偏移地址：0x0A8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	LPTIM1_ETR			RSV	GTIM3_ETR			RSV	GTIM2_ETR			RSV	GTIM1_ETR		
	RW				RW				RW				RW		

详细参考下文选择表格

注：Pinmux 辅助寄存器写需要任何一个 port 时钟使能，任何 port 复位均复位此寄存器。

9.5.21 Pinmux 辅助选择 5 (MUXGATE)

偏移地址：0x0AC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	LPTIM1_GATE			RSV	GTIM3_GATE			RSV	GTIM2_GATE			RSV	GTIM1_GATE		
	RW				RW				RW				RW		

详细参考下文选择表格

注：Pinmux 辅助寄存器写需要任何一个 port 时钟使能，任何 port 复位均复位此寄存器。

9.5.22 Pinmux 辅助选择表格

	000	001	010	011	100	101	110	111
ETR[2:0] ATIM 外部时钟来源	ATIM_ETR	UART1_RX	UART2_RX	VC1	PE00	PA00	PB00	PC00
ETR[6:4] STIM 外部时钟来源	STIM_ETR	UART3_RX	UART4_RX	LVD	VC2	PA01	PB01	PC01
ETR[10:8] BTIM123/CTIM1 外部时钟来源	Reserved	UART5_RX	UART6_RX	PCNT_S1	PE01	PA02	PB02	PC02
ETR[14:12] BTIM456 外部时钟来源	Reserved	LPUART1_RX	LPUART2_RX	PCNT_S2	PE02	PA03	PB03	PC03
GETR[2:0] GTIM1 外部时钟来源	GTIM1_ETR	UART1_RX	UART2_RX	VC1	PE03	PA04	PB04	PC04
GETR[6:4] GTIM2 外部时钟来源	GTIM2_ETR	UART3_RX	UART4_RX	LVD	VC2	PA05	PB05	PC05
GETR[10:8] GTIM3 外部时钟来源	GTIM3_ETR	UART5_RX	UART6_RX	PCNT_S1	PE04	PA06	PB06	PC06
GETR[14:12] LPTIMER1 外部时钟来源	LPTIM1_ETR	LPUART1_RX	LPUART2_RX	PCNT_S2	PE05	PA07	PB07	PC07
CAP[2:0] ATIM 外部输入通道来源	ATIM_CH0B	UART1_RX	UART2_RX	VC1	PE06	PA08	PB08	PC08
CAP[6:4] STIM 外部输入通道来源	STIM_CH4	UART3_RX	UART4_RX	LVD	VC2	PA09	PB09	PC09
CAP[10:8]	Reserved	UART5_RX	UART6_RX	PCNT_S1	PE08	PA10	PB10	PC10
CAP[14:12]	Reserved	LPUART1_RX	LPUART2_RX	PCNT_S2	PE09	PA11	PB11	PC11
GCAP[2:0] GTIM1 外部输入通道来源	GTIM1_CH1	UART1_RX	UART2_RX	VC1	PE10	PA12	PB12	PC12
GCAP[6:4] GTIM2 外部输入通道来源	GTIM2_CH1	UART3_RX	UART4_RX	LVD	VC2	PA13	PB13	PC13
GCAP[10:8] GTIM3 外部输入通道来源	GTIM3_CH1	UART5_RX	UART6_RX	PCNT_S1	PE12	PA14	PB14	PC14
GCAP[14:12] GTIM3 外部输入通道来源	GTIM3_CH2	LPUART1_RX	LPUART2_RX	PCNT_S2	PE13	PA15	PB15	PC15
GATE[2:0] GTIM1 外部输入 Gate 来源	GTIM1_GT	UART1_RX	UART2_RX	VC1	PE14	PD00	PD01	PD02
GATE[6:4] GTIM2 外部输入 Gate 来源	GTIM2_GT	UART3_RX	UART4_RX	LVD	VC2	PD04	PD05	PD06
GATE[10:8] GTIM3 外部输入 Gate 来源	GTIM3_GT	UART5_RX	UART6_RX	PCNT_S1	PE15	PD08	PD09	PD10

GATE[14:12] LPTIMER 外部输入 Gate 来源	LPTIM1_GT	LPUART1_RX	LPUART2_RX	PCNT_S2	PD15	PD12	PD13	PD14

10. 复位控制器（RESET）

10.1 复位控制器介绍

本产品具有 7 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序会从复位向量处开始执行。

- 数字区域上电掉电复位 POR
- 外部 Reset PAD，低电平为复位信号
- IWDT 复位
- WWDT 复位
- LVD 低电压复位
- Cortex-M0+ SYSRESETREQ 软件复位
- Cortex-M0+ LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示。复位标志均由硬件置位，需要用户软件清零。芯片复位时，如果查询到 Reset_flag. POR15V 或 Reset_flag. POR5V 为 1 则为上电复位。上电复位时用户程序应当将寄存器 Reset_flag 清零，则下一次复位时可通过 Reset_flag 的相关比特判断复位来源。

下图描述各区域的复位来源。

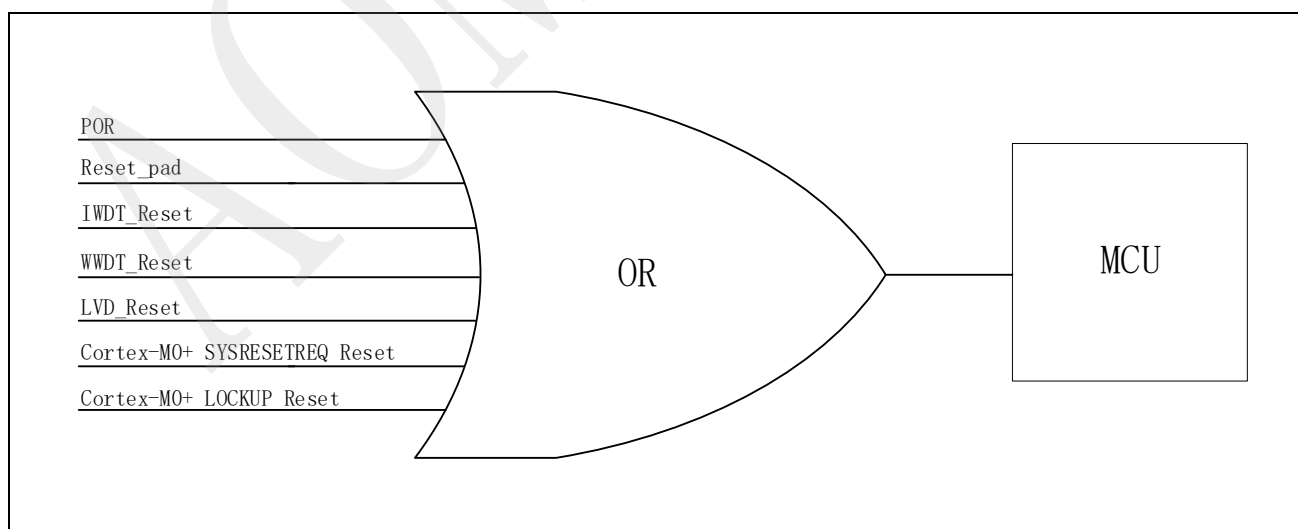


图 10-1 复位来源示意图

10.1.1 上电掉电复位 POR

本产品有两个供电区域：**VCC** 区域、**Vcore** 区域。所有的模拟模块及 **IO** 工作于 **VCC** 区域；其它模块工作于 **Vcore** 区域。

VCC 区域上电时，当 **VCC** 电压低于 **POR** 阈值电压时（典型值为 1.65V），会产生 **POR5V** 信号；**VCC** 区域下电时，当 **VCC** 电压低于 **BOR** 阈值电压时（典型值为 1.5V），会产生 **POR5V** 信号。

Vcore 区域上电时，当 **Vcore** 电压低于 **POR** 阈值电压时，会产生 **POR15V** 信号；**Vcore** 区域掉电时，当 **Vcore** 电压低于 **BOR** 阈值电压时，会产生 **POR15V** 信号。

POR5V 信号和 **POR15V** 信号均会将芯片的寄存器复位到初始化状态。

10.1.2 外部复位管脚复位

当外部复位引脚检测到低电平时会产生一个系统复位。该复位引脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于 20 μ s（典型值）的毛刺信号，因此，加到复位引脚上的低电平信号必须大于 20 μ s，才能确保芯片可靠复位。

10.1.3 IWDT 复位

看门狗复位，请参看 **IWDT** 一章说明。

10.1.4 WWDT 复位

WWDT 复位，请参看 **WWDT** 一章说明。

10.1.5 LVD 低电压复位

LVD 复位，请参考 **LVD** 一章说明。

10.1.6 Cortex-M0+ SYSRESETREQ 复位

Cortex-M0+ 软件复位

10.1.7 Cortex-M0+ LOCKUP 复位

当 Cortex-M0+ 遇到严重的异常时，它会将自己的 **PC** 指针停在当前地址处，并锁

死自己，并在几个时钟周期延时之后复位整个 CORE 区域。

10.2 寄存器

基地址 0x40010000

寄存器	偏移地址	描述
RESET_FLAG	0x01C	复位标识寄存器
IORST	0x070	端口复位控制寄存器
AHBRST	0x074	AHB外设复位控制寄存器
APBRST1	0x078	APB外设复位控制寄存器1
APBRST2	0x07C	APB外设复位控制寄存器2

表 10-1 系统控制寄存器表格

10.2.1 复位标识寄存器（RESET_FLAG）

偏移地址：0x01C

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
RSTB	SYSREQ	LOCKUP	WWDT	IWDT	LVD	POR15V	POR5V																
RW0	RW0	RW0	RW0	RW0	RW0	RW0	RW0																

位	标记	功能描述
31:8	Reserved	保留位
7	RSTB	RESETB 端口复位标志，需要软件初始化及清除 1: 发生端口复位 0: 无端口复位发生 写0清除，写1无效
6	SYSREQ	Cortex-M0+ CPU 软件复位标志，需要软件初始化及清除 1: 发生 Cortex-M0+ CPU 软件复位 0: 无 Cortex-M0+ CPU 软件复位发生 写0清除，写1无效
5	LOCKUP	Cortex-M0+ CPU Lockup 复位标志，需要软件初始化及清除 1: 发生 Cortex-M0+ CPU Lockup 复位 0: 无 Cortex-M0+ CPU Lockup 复位发生 写0清除，写1无效
4	WWDT	WWDT 复位标志，需要软件初始化及清除 1: 发生 WWDT 复位

		0: 无WWDT 复位发生 写0清除, 写1无效
3	IWDT	IWDT 复位标志, 需要软件初始化及清除 1: 发生 IWDT 复位 0: 无 IWDT 复位发生 写0清除, 写1无效
2	LVD	LVD 复位标志, 需要软件初始化及清除 1: 发生 LVD 复位 0: 无 LVD 复位发生 写0清除, 写1无效
1	POR15V	Vcore域复位标志 1: Vcore域发生复位 0: Vcore域无复位发生 写0清除, 写1无效
0	POR5V	VCC电源域复位标志 1: VCC电源域发生复位 0: VCC电源域无复位发生 写0清除, 写1无效

10.2.2 IO 复位控制寄存器 (IORST)

偏移地址: 0x070

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										PF	PE	PD	PC	PB	PA
										RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:6	Reserved	保留
5	PF	端口PF复位控制 1: 复位 0: 无效
4	PE	端口PE复位控制 1: 复位 0: 无效
3	PD	端口PD复位控制 1: 复位 0: 无效
2	PC	端口PC复位控制 1: 复位 0: 无效
1	PB	端口PB复位控制

		1: 复位 0: 无效
0	PA	端口PA复位控制 1: 复位 0: 无效

10.2.3 AHB 外设复位控制寄存器（AHBRST）

偏移地址：0x074

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													RNG	SM4	AES
													RW	RW	RW
15		14		13		12		11		10		9		8	
CAN2		CAN1		RSV		CRC		Reserved							
RW		RW				RW									
7		6		5		4		3		2		1		0	
Reserved													DMA		
													RW		

位	标记	功能描述
31:19	Reserved	保留
18	RNG	外设RNG复位控制 1: 复位 0: 无效
17	SM4	外设SM4复位控制 1: 复位 0: 无效
16	AES	外设AES复位控制 1: 复位 0: 无效
15	CAN2	外设CAN2复位控制 1: 复位 0: 无效
14	CAN1	外设CAN1复位控制 1: 复位 0: 无效
13	Reserved	保留
12	CRC	外设CRC复位控制 1: 复位 0: 无效
11:1	Reserved	保留
0	DMA	外设DMA复位控制 1: 复位 0: 无效

10.2.4 APB1 外设复位控制寄存器（APBRST1）

偏移地址：0x078

复位值：0x0000 0000

31	30	29	28	27	26	25	24
LPTIM1	OPA	DAC	Reserved	LPTIM2	I2C4	Reserved	CTRIM
RW	RW	RW		RW	RW		RW
23	22	21	20	19	18	17	16
I2C3	I2C2	I2C1	UART5	UART4	UART3	UART2	SPI4
RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
SPI3	SPI2	Reserved			RTC	LCD	Reserved
RW	RW				RW	RW	
7	6	5	4	3	2	1	0
STIM	PCA	Reserved		PCNT	GTIM3	GTIM2	GTIM1
RW	RW			RW	RW	RW	RW

位	标记	功能描述
31	LPTIM1	外设LPTIM1复位控制 1：复位 0：正常工作
30	OPA	外设OPA复位控制 1：复位 0：正常工作
29	DAC	外设DAC复位控制 1：复位 0：正常工作
28	Reserved	保留
27	LPTIM2	外设LPTIM2复位控制 1：复位 0：正常工作
26	I2C4	外设I2C4复位控制 1：复位 0：正常工作
25	Reserved	保留
24	CTRIM	外设clock trimming复位控制 1：复位 0：正常工作
23	I2C3	外设I2C3复位控制 1：复位 0：正常工作
22	I2C2	外设I2C2复位控制 1：复位 0：正常工作
21	I2C1	外设I2C1复位控制

		1: 复位 0: 正常工作
20	UART5	外设UART5复位控制 1: 复位 0: 正常工作
19	UART4	外设UART4复位控制 1: 复位 0: 正常工作
18	UART3	外设UART3复位控制 1: 复位 0: 正常工作
17	UART2	外设UART2复位控制 1: 复位 0: 正常工作
16	SPI4	外设SPI4复位控制 1: 复位 0: 正常工作
15	SPI3	外设SPI3复位控制 1: 复位 0: 正常工作
14	SPI2	外设SPI2复位控制 1: 复位 0: 正常工作
13:11	Reserved	保留
10	RTC	外设RTC复位控制 1: 复位 0: 正常工作
9	LCD	外设LCD复位控制 1: 复位 0: 正常工作
8	Reserved	保留
7	STIM	外设STIM复位控制 1: 复位 0: 正常工作
6	PCA	外设PCA复位控制 1: 复位 0: 正常工作
5:4	Reserved	保留
3	PCNT	外设PCNT复位控制 1: 复位 0: 正常工作
2	GTIM3	外设GTIM3复位控制 1: 复位 0: 正常工作
1	GTIM2	外设GTIM2复位控制 1: 复位 0: 正常工作
0	GTIM1	外设GTIM1复位控制 1: 复位 0: 正常工作

10.2.5 APB2 外设复位控制寄存器（APBRST2）

偏移地址：0x07C

复位值：0x0000 0000

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved		I2S	LPUART2	LPUART1	BTIM789	BTIM456	BTIM123
		RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8
UART6	UART1	RSV	SPI1	ATIM	VC	ADC	Reserved
RW	RW		RW	RW	RW	RW	

7	6	5	4	3	2	1	0
Reserved				ISO7816	Reserved		
				RW			

位	标记	功能描述
31:22	Reserved	保留
21	I2S	外设I2S复位控制 1: 复位 0: 正常工作
20	LPUART2	外设LPUART2复位控制 1: 复位 0: 正常工作
19	LPUART1	外设LPUART1复位控制 1: 复位 0: 正常工作
18	BTIM789	外设BTIM789复位控制 1: 复位 0: 正常工作
17	BTIM456	外设BTIM456复位控制 1: 复位 0: 正常工作
16	BTIM123	外设BTIM123复位控制 1: 复位 0: 正常工作
15	UART6	外设UART6复位控制 1: 复位 0: 正常工作
14	UART1	外设UART1复位控制 1: 复位 0: 正常工作
13	Reserved	保留
12	SPI1	外设SPI1复位控制 1: 复位 0: 正常工作
11	ATIM	外设ATIM复位控制 1: 复位 0: 正常工作
10	VC	外设VC比较器复位控制 1: 复位 0: 正常工作

9	ADC	外设ADC复位控制 1: 复位 0: 正常工作
8: 4	Reserved	保留
3	IS07816	外设智能卡7816接口复位控制 1: 复位 0: 正常工作
2: 0	Reserved	保留

11. 中断控制器（NVIC）

11.1 概述

Cortex-M0+ 处理器内置了嵌套向量中断控制器（NVIC），支持最多 32 个中断请求（IRQ）输入，以及 1 个不可屏蔽中断（NMI）输入（在本产品系统中并未使用）。

另外，处理器还支持多个内部异常。

每个异常源都有一个单独的异常编号，每种异常类型都有对应的优先级，有些异常的优先级是固定的，而有些则是可编程的。具体如下表所示：

异常编号	异常类型	优先级	描述
1	复位	-3（最高）	复位
2	NMI	-2	不可屏蔽中断（在本系统中没有使用）
3	硬件错误	-1	错误处理异常
4-10	保留	NA	...
11	SVC	可编程	通过SVC指令调用管理程序
12-13	保留	NA	...
14	PendSV	可编程	系统服务的可挂起请求
15	SysTick	可编程	SysTick定时器
16	中断#0	可编程	外部中断#0
17	中断#1	可编程	外部中断#1
...	...	...	...
47	中断#31	可编程	外部中断#31

表 11-1 Cortex-M0+ 处理器中断一览

本章节只对处理器的 32 个外部中断请求（中断#0 到中断#31）做详细介绍，处理器内部异常的具体情况可参考其他相关文档。同时，本章节只讨论处理器内核中 NVIC 的中断处理机制，外设模块自身的中断产生机制不在此处展开讨论。

11.2 中断优先级

每一个外部中断都对应一个优先级寄存器，每个优先级都是 2 位宽，并且使用中断优

优先级寄存器的最高两位，每个寄存器占 1 个字节（8 位）。在这个设定下，可以使用
的优先级为 0x00（最高）、0x40、0x80 和 0xc0（最低）。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
已使用		未使用，读出为0					

图 11-1 只使用了高两位的优先级寄存器

如果处理器已经在运行另外一个中断处理，而新中断的优先级大于正在执行的，这时就会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。

如果处理器正在运行的另外一个中断处理的优先级相同或者更高，新的中断将会等待并且进入挂起状态。挂起的中断将会一直等到当前中断等级改变，例如，当前运行的中断处理完成返回后，当前优先级降低到了比挂起中断还要小。

如果两个中断同时发生，并且它们的优先级相同，中断编号较小的中断将会首先执行。例如，如果中断#0 和中断#1 使能且具有相同的优先级，在它们同时被触发时，中断#0 会首先执行。

11.3 中断向量表

当 Cortex-M0+ 处理器要处理中断服务请求时，它需要首先确定异常处理的起始地址，所需的信息叫做向量表，如图 11-2 所示。向量表存储在存储器空间的开始位置，包含了系统中可用异常（中断）的异常（中断）向量，以及主栈指针（MSP）的初始值。

存储器地址		异常编号
0x0000004C	中断#3向量	19
0x00000048	中断#2向量	18
0x00000044	中断#1向量	17
0x00000040	中断#0向量	16
0x0000003C	SysTick向量	15
0x00000038	PendSV向量	14
0x00000034	未使用	13
0x00000030	未使用	12
0x0000002C	SVC向量	11
0x00000028	未使用	10
0x00000024	未使用	9
0x00000020	未使用	8
0x0000001C	未使用	7
0x00000018	未使用	6
0x00000014	未使用	5
0x00000010	未使用	4
0x0000000C	硬件错误异常	3
0x00000008	NMI向量	2
0x00000004	复位向量	1
0x00000000	MSP初始值	0

图 11-2 中断向量表

其中，中断向量的存储顺序同中断编号一致，由于每个向量都是 1 个字（4 字节），中断向量的地址为中断编号乘 4，每个中断向量都是中断处理的起始地址。

11.4 中断输入和挂起行为

Cortex-M0+ 处理器的 NVIC 模块中，每一个中断输入都对应着一个挂起状态寄存器，且每个寄存器只有 1 位，用于保存中断请求，而不管这个请求有没有得到确认。当处理器开始处理这个中断时，硬件将会自动清除挂起状态位。

本系统的外设使用电平触发中断输出，当中断事件发生时，由于外设连接到了 NVIC 上，中断信号会得到确认。在处理器执行中断服务并且清除外设的中断信号以前，该信号会保持高电平。在 NVIC 内部，当检测到有中断发生时，该中断的挂起状态会被置位，当处理器接收该中断并且开始执行中断服务程序后，挂起状态就会被清除。该过程如图 11-3 所示：

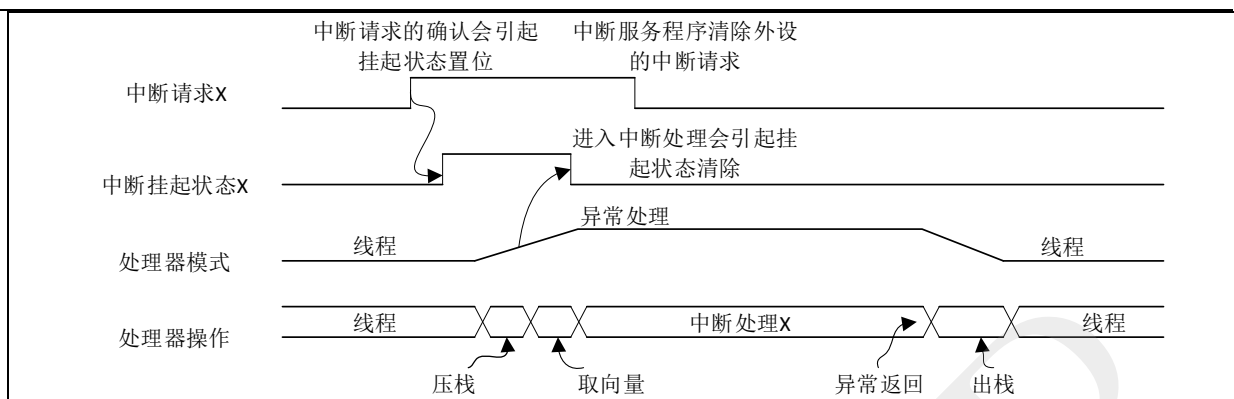


图 11-3 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除了，这样处理器会忽略掉本次请求，并且不会执行中断处理。可以通过写 `NVIC_CLRPEND` 寄存器来清除中断挂起状态，这种处理在设置外设时非常有用，因为在设置以前，该外设可能已经产生了一个中断请求。

如果在软件清除挂起状态时，外设仍然保持着中断请求，挂起状态还会立即生成。该过程如图 11-4 所示：

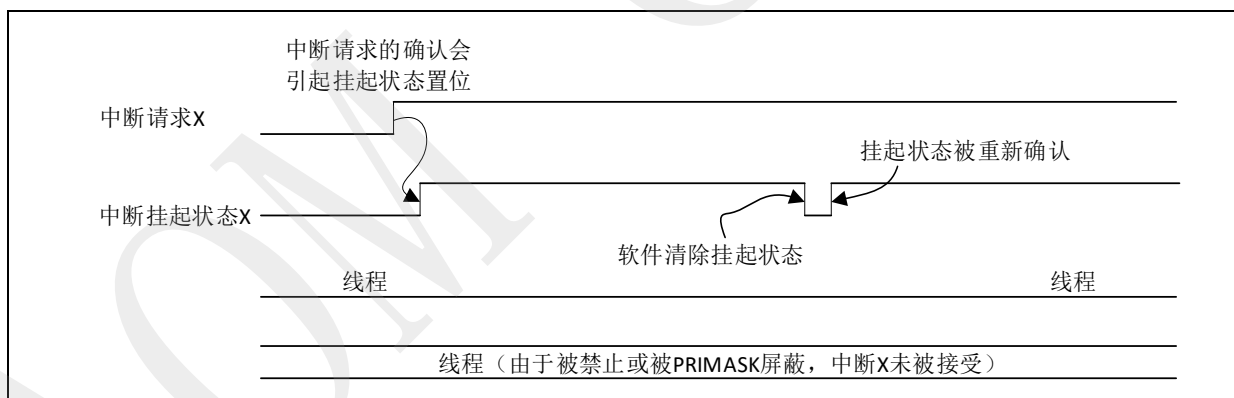


图 11-4 中断挂起状态被清除然后被重新确认

如果外设产生的中断请求在异常处理时没有被清除，异常返回后挂起状态就会被又一次激活，这样中断服务程序会再次执行。该过程如图 11-5 所示：

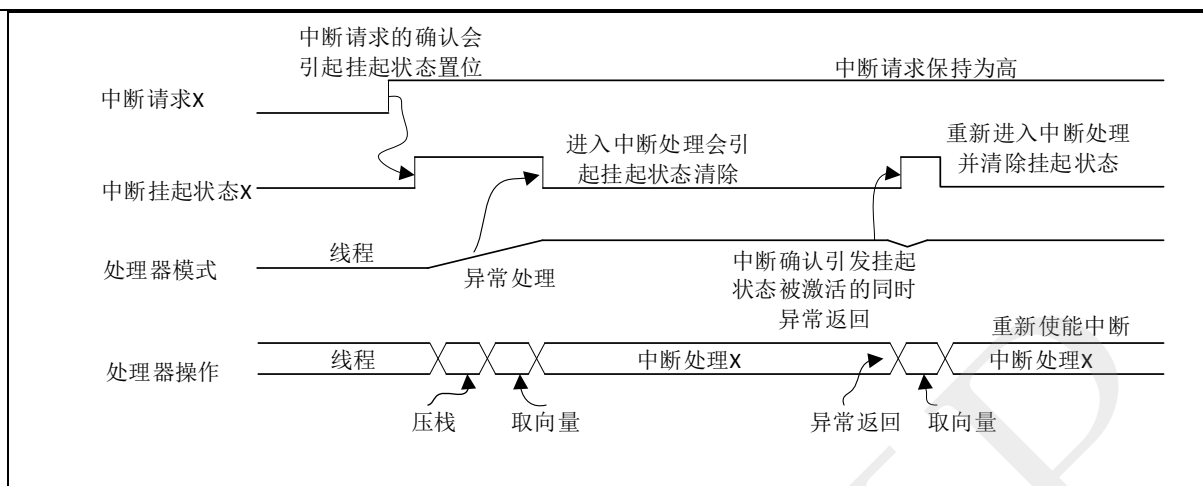


图 11-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

如果在终端服务程序执行的过程中产生外设中断请求,该请求会被当作新的中断请求,并且在本次中断退出后,还会引起中断服务程序的再次执行。该过程如图 11-6 所示:

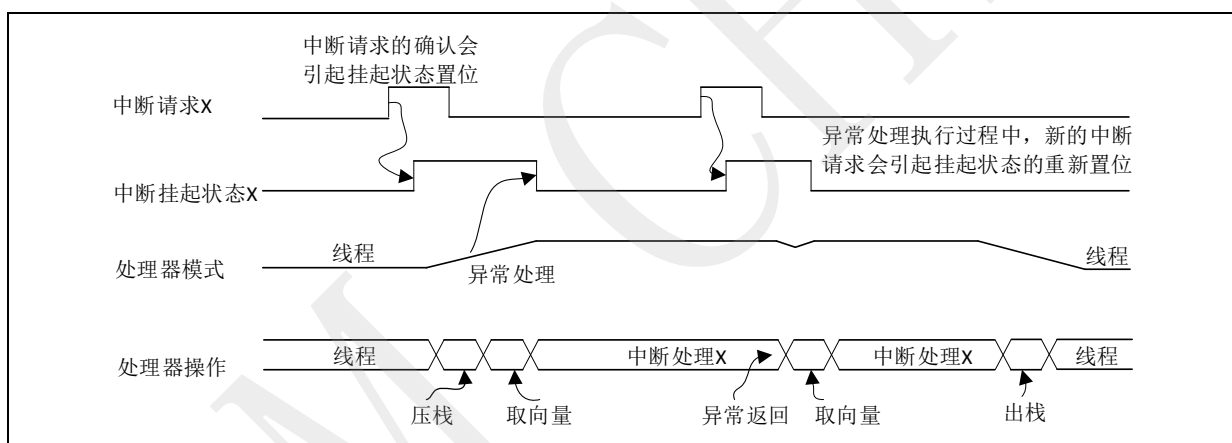


图 11-6 中断处理中产生的中断挂起也可以被确认

11.5 中断等待

通常情况下, NVIC 的中断等待时间为 16 个周期。这个等待时间从中断确认的处理时钟周期开始,一直到中断处理开始执行结束。计算中断等待需要具备以下前提:

- 该中断使能并且没有被 SCS_PRIMASK 或是其他正在执行的异常处理所屏蔽。
- 存储器系统没有任何等待状态,在中断处理、压栈、取向量或者中断处理开始时的取指都会用到总线传输,如果存储器系统需要等待,那么发生总线传输时产生的等待状态则可能使得中断延迟。

- 下面的几种情况可能会导致不同的中断等待:

- 中断的末尾连锁，如果中断返回时产生了另外一个中断请求，处理器就会跳过出栈和压栈过程，这样就减少了中断等待时间。
- 延迟到达，如果中断发生时，另外一个低优先级的中断正在进行压栈处理，由于延迟到达机制的存在，高优先级的中断会首先执行，这样也会导致高优先级中断的等待时间减小。

11.6 中断源

因为 Cortex-M0+ 处理器的 NVIC 支持最多 32 个外部中断，而在本系统中，外部中断源大于 32 个，所以部分外部中断复用在同一个 NVIC 中断输入上，其中 NMI（不可屏蔽中断）并没有使用。本系统所有外部中断源和 NVIC 中断输入的对应关系如下表所示：

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#0	WWDT/IWDT	v	v	v
中断#1	LVD	v	v	v
中断#2	RTC	v	v	v
中断#3	FLASH/RAM	v	v	—
中断#4	CRS/CCS	v	v	v
中断#5	PA	v	v	v
中断#6	PB	v	v	v
中断#7	PC/PE	v	v	v
中断#8	PD/PF	v	v	v
中断#9	DMA_CH0	v	v	v
中断#10	DMA_CH1/2	v	v	—
中断#11	DMA_CH3	v	v	—
中断#12	ADC/VC1/VC2	v	v	v
中断#13	ATIM	v	v	—
中断#14	GTIM1	v	v	—
中断#15	GTIM2	v	v	—
中断#16	GTIM3	v	v	—
中断#17	LPTIM1/DAC	v	v	v
中断#18	LPTIM2/LCD	v	v	v
中断#19	LPUART1/2	v	v	v
中断#20	BTIM123/ PCA	v	v	—
中断#21	BTIM456/ PCNT	v	v	v

中断#22	BTIM789/ STIM	v	v	—
中断#23	I2C1/I2C3	v	v	—
中断#24	I2C2/I2C4	v	v	—
中断#25	SPI1/SPI3/I2S	v	v	—
中断#26	SPI2/SPI4	v	v	—
中断#27	UART1/E7816	v	v	—
中断#28	UART2	v	v	—
中断#29	UART3/UART4/UART5/UART6	v	v	—
中断#30	CAN1/CAN2	v	v	—
中断#31	--	—	—	—

表 11-2 外部中断与 NVIC 中断输入对应关系

注意：

- 由于某些模块中断被复用于同一个 IRQ 中断源，当 CPU 进入该中断操作时，必须先判断是哪个模块产生的中断，再进行相对应的中断操作。

11.7 中断结构图

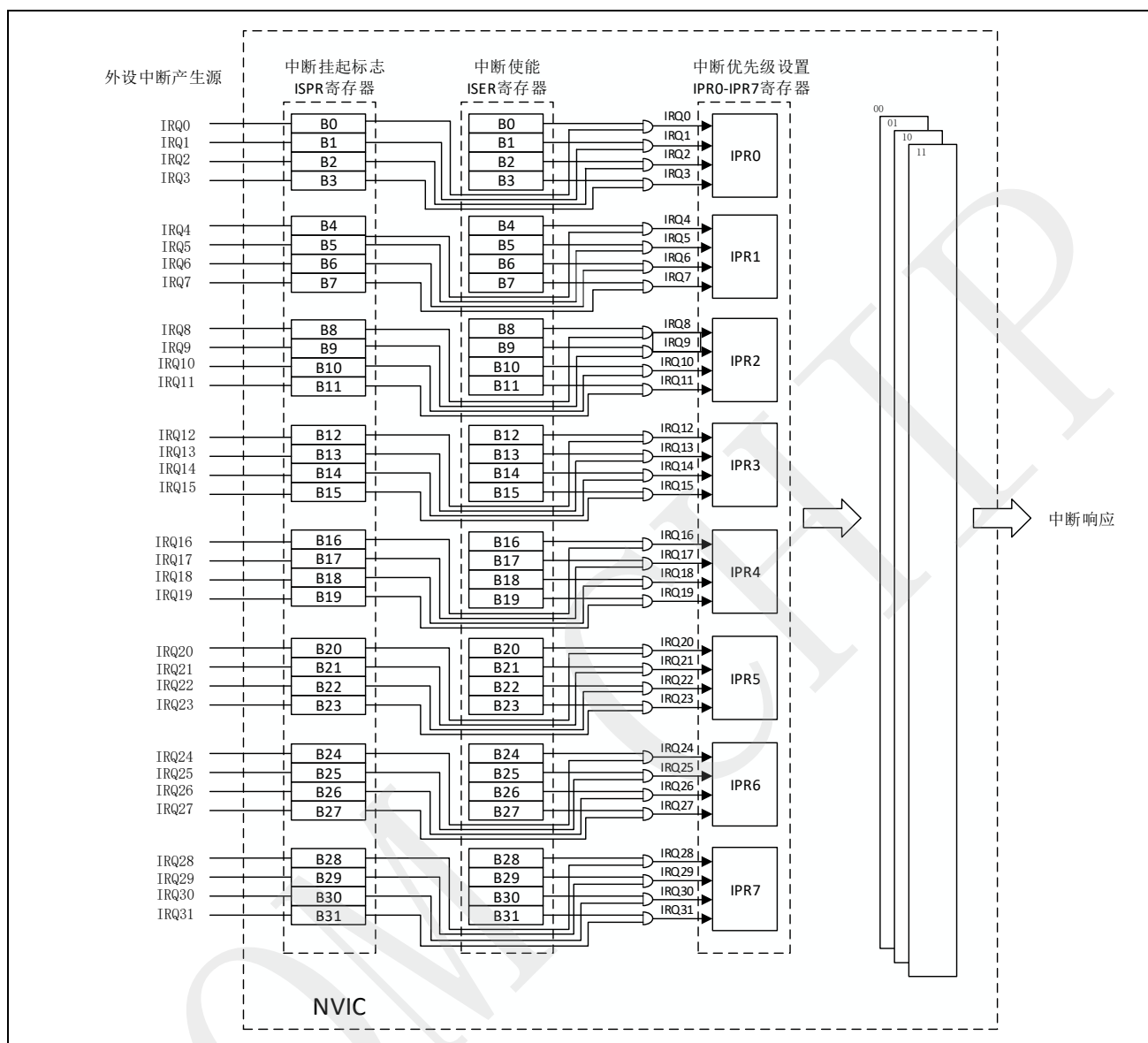


图 11-7 中断结构图

本系统的中断结构框图如图 11-7 所示。需要注意几点：

- 外设中断源各自的中断使能没有在图中标出，这里只包括外设中断产生之后的中断信号逻辑框图。
- IRQ 有多于 1 个外设中断输入复用，必须分别读取这些外设的中断标志位来判定是哪个外设的中断。
- 如果外设中断源有高电平产生，无论 NVIC 中断使能寄存器 NVIC_ISER 置位与否，中断挂起寄存器 SCS_SEPEND 都会被置位，表示相应的外设中断源有中断产生。

- 只有中断使能寄存器 NVIC_ISER 置位，相应的中断 IRQ 才会给处理器相应，执行相应的中断程序。
- 在中断程序中必须清除外设中断源高电平中断信号，中断挂起寄存器 NVIC_ISPR 由硬件自动清除。
- 中断优先级寄存器 NVIC_IPR0- NVIC_IPR7 设置了 32 个中断源的优先级，00 优先级最高，11 优先级最低。当优先级相同时，优先级由中断编号决定，编号越小优先级越高。

11.8 寄存器

基地址：0xE000 E000

寄存器	偏移地址	描述
NVIC_ISER	0x100	中断请求使能寄存器
NVIC_ICER	0x180	中断请求清除使能寄存器
NVIC_ISPR	0x200	中断设置挂起寄存器
NVIC_ICPR	0x280	中断清除挂起寄存器
NVIC_IPR0	0x400	中断#0-中断#3优先级寄存器
NVIC_IPR1	0x404	中断#4-中断#7优先级寄存器
NVIC_IPR2	0x408	中断#8-中断#11优先级寄存器
NVIC_IPR3	0x40C	中断#12-中断#15优先级寄存器
NVIC_IPR4	0x410	中断#16-中断#19优先级寄存器
NVIC_IPR5	0x414	中断#20-中断#23优先级寄存器
NVIC_IPR6	0x418	中断#24-中断#27优先级寄存器
NVIC_IPR7	0x41C	中断#28-中断#31优先级寄存器
PRIMASK	—	中断屏蔽特殊寄存器

表 11-3 NVIC 模块寄存器列表

11.8.1 中断使能设置寄存器（NVIC_ISER）

偏移地址：0x100

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETENA[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SETENA[15:0]
RW

位	标记	功能描述
31:0	SETENA[31:0]	设置使能中断#0到中断#31；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

11.8.2 中断使能清除寄存器（NVIC_ICER）

偏移地址：0x180

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA[15:0]															
RW															

位	标记	描述
31:0	CLRENA[31:0]	清除使能中断#0到中断#31；写“1”清零，写“0”无效

11.8.3 中断挂起状态设置寄存器（NVIC_ISPR）

偏移地址：0x200

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETPEND[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETPEND[15:0]															

RW

位	标记	功能描述
31:0	SETPEND[31:0]	设置中断#0到中断#31的挂起状态；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

11.8.4 中断挂起状态清除寄存器（NVIC_ICPR）

偏移地址：0x280

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRPEND[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRPEND[15:0]															
RW															

位	标记	描述
31:0	CLRPEND[31:0]	清除中断#0到中断#31的挂起状态；写“1”清零，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

11.8.5 中断优先级寄存器（NVIC_IPR0）

偏移地址：0x400

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR0[15:0]]															
RW															

位	标记	功能描述
31:0	IPR0[31:0]	中断#0到中断#3的优先级： [31:30]：中断#3 的优先级 [23:22]：中断#2 的优先级 [15:14]：中断#1 的优先级 [7:6]： 中断#0 的优先级 其中，00 优先级最高，11 优先级最低

11.8.6 中断优先级寄存器（NVIC_IPR1）

偏移地址：0x404

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR1[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR1[15:0]]															
RW															

位	标记	功能描述
31:0	IPR1[31:0]	中断#4到中断#7的优先级： [31:30]：中断#7 的优先级

		[23:22]: 中断#6 的优先级 [15:14]: 中断#5 的优先级 [7:6]: 中断#4 的优先级 其中, 00 优先级最高, 11 优先级最低

11.8.7 中断优先级寄存器 (NVIC_IPR2)

偏移地址: 0x408

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR2[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR2[15:0]															
RW															

位	标记	功能描述
31:0	IPR2[31:0]	中断#8到中断#11的优先级; [31:30]: 中断#11 的优先级 [23:22]: 中断#10 的优先级 [15:14]: 中断#9 的优先级 [7:6]: 中断#8 的优先级 其中, 00 优先级最高, 11 优先级最低

11.8.8 中断优先级寄存器 (NVIC_IPR3)

偏移地址: 0x40C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR3[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]															

RW

位	标记	功能描述
31:0	IPR3[31:0]	中断#12到中断#15的优先级; [31:30]: 中断#15 的优先级 [23:22]: 中断#14 的优先级 [15:14]: 中断#13 的优先级 [7:6]: 中断#12 的优先级 其中, 00 优先级最高, 11 优先级最低

11.8.9 中断优先级寄存器 (NVIC_IPR4)

偏移地址: 0x410

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR4[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR4[15:0]															
RW															

位	标记	功能描述
31:0	IPR4[31:0]	中断#16到中断#19的优先级; [31:30]: 中断#19 的优先级 [23:22]: 中断#18 的优先级 [15:14]: 中断#17 的优先级 [7:6]: 中断#16 的优先级 其中, 00 优先级最高, 11 优先级最低

11.8.10 中断优先级寄存器（NVIC_IPR5）

偏移地址：0x414

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR5[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR5[15:0]]															
RW															

位	标记	功能描述
31:0	IPR5[31:0]	中断#20到中断#23的优先级； [31:30]：中断#23 的优先级 [23:22]：中断#22 的优先级 [15:14]：中断#21 的优先级 [7:6]： 中断#20 的优先级 其中，00 优先级最高，11 优先级最低

11.8.11 中断优先级寄存器（NVIC_IPR6）

偏移地址：0x418

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR6[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR6[15:0]															
RW															

位	标记	功能描述
31:0	IPR6[31:0]	中断#24到中断#27的优先级： [31:30]：中断#27 的优先级 [23:22]：中断#26 的优先级 [15:14]：中断#25 的优先级 [7:6]： 中断#24 的优先级 其中，00 优先级最高，11 优先级最低

11.8.12 中断优先级寄存器（NVIC_IPR7）

偏移地址：0x41C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]]															
RW															

位	标记	功能描述
31:0	IPR7[31:0]	中断#28到中断#31的优先级： [31:30]：中断#31 的优先级 [23:22]：中断#30 的优先级 [15:14]：中断#29 的优先级 [7:6]： 中断#28 的优先级 其中，00 优先级最高，11 优先级最低

11.8.13 中断屏蔽特殊寄存器（PRIMASK）

偏移地址：---

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PRIMASK	
														RW	

BIT	符号	描述
31:1	Reserved	保留
0	PRIMASK	置位后，除了 NMI 和硬件错误异常外的其他中断都会被屏蔽掉 清零后，所有异常和中断都不会被屏蔽掉 该特殊寄存器需要通过 MSR 和 MRS 特殊寄存器操作指令访问，也可以用改变处理器状态指令 CPS 访问。在处理对时间敏感的应用时，需要操作 PRIMASK 寄存器。

11.9 软件基本操作

11.9.1 外部中断使能

在每一个外设模块内部都有各自的中断使能寄存器，在需要进行中断操作时，必须首先打开外设自己的中断使能。该使能位的操作没有在本章节讨论，请参考外设模块各自的章节描述。

11.9.2 NVIC 中断使能和清除使能

Cortex-M0+ 处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零使能位。这样就有了 32 位的中断使能寄存器 NVIC_ISER 和 32 位的清零使能寄存器 NVIC_ICER。如果想使能某一个中断，则对 NVIC_ISER 寄存器的相应位置 1。如果想清零某一个中断，则对 NVIC_ICER 寄存器的相应位置 1。

注意，这里提到的中断使能仅仅是针对处理器 NVIC 而言的，每个外设的中断生成

与否，是由外设的中断控制寄存器决定，与 NVIC_ISER 和 NVIC_ICER 无关。

11.9.3 NVIC 中断挂起和清除挂起

如果一个中断发生了，却无法立即处理，这个中断请求将会被挂起。挂起状态保存在一个寄存器中，如果处理器的当前优先级还没有降低到可以处理挂起的请求，并且没有手动清除挂起状态，该状态将会一直保持合法。

当处理器开始进入中断处理，硬件会自动引起挂起状态的清除。

可以通过操作中断设置挂起 NVIC_ISPR 和中断清除挂起 NVIC_ICPR 这两个寄存器来访问或修改中断挂起状态。中断挂起状态寄存器允许使用软件来触发中断。

11.9.4 NVIC 中断优先级

设置 NVIC_IPR0- NVIC_IPR7 寄存器决定 SCS_IRQ0- SCS_IRQ32 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优先级，这种情况的结果不可预知，并且不被 Cortex-M0+ 处理器支持。

11.9.5 NVIC 中断屏蔽

有些对时间敏感的应用，需要在一段较短的时间内禁止所有中断，可以利用中断屏蔽寄存器 SCS_PRIMASK 实现。特殊寄存器 SCS_PRIMASK 只有 1 位有效，并且在复位后默认为 0。该寄存器为 0 时，所有的中断和异常都处于允许状态；而设为 1 后，只有 NMI（本系统不支持）和硬件错误异常处于使能。实际上，当 SCS_PRIMASK 设置为 1 后，处理器的当前优先级就降到了 0（可和值的最高优先级）。

可以通过多种方法编程 SCS_PRIMASK 寄存器，使用汇编语言，可以利用 MS R 指令在设置和清除 SCS_PRIMASK 寄存器。若使用 C 语言以及 CMSIS 设备驱动库，用户可以使用以下函数来设置和清除 PRIMASK。

```
void __enable_irq(void); //清除 PRIMASK
```

```
void __disable_irq(void); //设置 PRIMASK
```

12. SWD 调试接口

6615 系列使用 ARM Cortex-M0+内核，该内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。当微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

注意：

- SWD 能把 DeepSleep 唤醒，请注意不要以 SWD 接口断开作为 DeepSleep 的状态。

12.1 SWD 调试附加功能

本产品使用了 ARM Cortex-M0+ CPU，该内核包含用于高级调试功能的硬件扩展，因此本产品所拥有的调试功能与 Cortex-M0+一致。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。当调试主机与 MCU 相连并进行调试时，将使用调试功能。

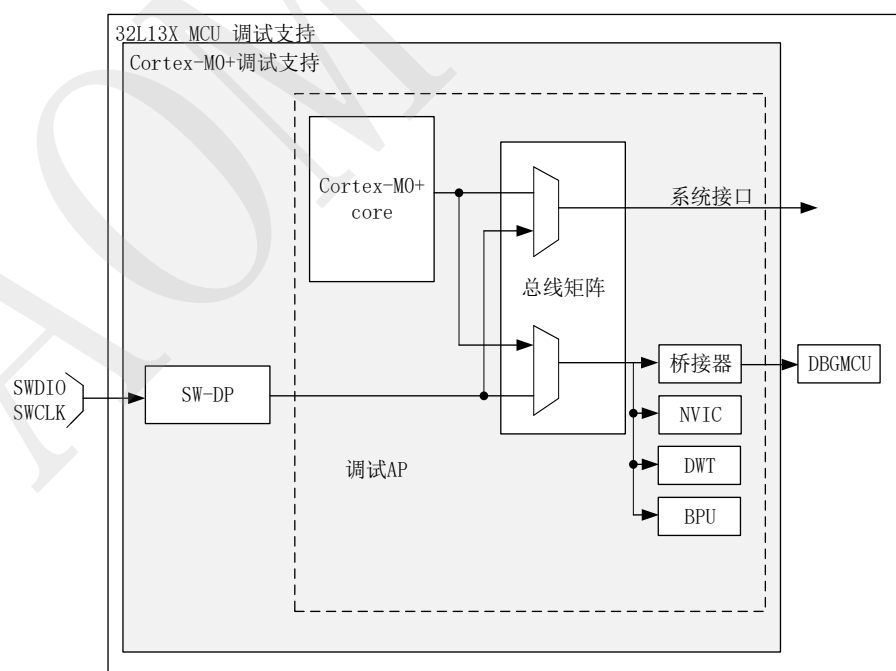


图 12-1 调试支持框图

Cortex®-M0+ 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex®-M0+内核提供集成片上调试支持。它包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发

注：

- 有关 ARM® Cortex®-M0+ 内核支持的调试功能的详细信息，请参见 Cortex®-M0+ 技术参考手册。

12.2 ARM® 参考文档

- Cortex®-M0+技术参考手册 (TRM)
可从 www.infocenter.arm.com 获取。
- ARM® 调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册

12.3 调试端口引脚

12.3.1 SWD 端口引脚

6615 系列的 SWD 接口，需要用到 2 个引脚，如下表所示。

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	PA14
SWDIO	串行数据输入/输出	PA13

12.3.2 SW-DP 引脚分配

如果烧录程序时使能了【加密芯片】选项，则上电后 SWD 调试功能被禁止。如果烧录程序时没有使能【加密芯片】选项，则上电后 PA13/PA14 引脚均被初始化为可被调试器使用的专用引脚。用户可设置 SYSCTRL1.SWD_USE_IO 寄存器来禁止 SWD 引脚的调试功能，SWD 引脚将被释放以用作普通 GPIO。

SWD 引脚的配置与功能小结如下表所示：

【加密芯片】选项	SWD_USE_IO 配置	PA13/PA14 功能
加密	0	NA
加密	1	GPIO
不加密	0	SWD
不加密	1	GPIO

12.3.3 SWD 引脚上的内部上拉

用户软件释放 SW I/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入上拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

12.4 SWD 端口

12.4.2 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（ARM® 建议采用 100 k）。这些上拉电阻可在内部配置。无需外部上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

12.4.3 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

位	名称	说明
0	启动	必须为1
1	APnDP	0: DP 访问; 1: AP 访问
2	RnW	0: 写请求; 1: 读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M0+ TRM。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动。

位	名称	说明
0	ACK	001: FAULT 010: WAIT 100: OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

位	名称	说明
0:31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

12.4.4 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 ARM®代码，设置为 0x0BB11477（相当于 Cortex®-M0+）。

注意：

- 在目标读取此 ID 代码前，SW-DP 状态机是不工作的。

- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机必须首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 Cortex®-M0+ TRM 和 CoreSight 设计套件 r1p0TRM。

12.4.5 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。
- 每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作

（在内核上电后才有效的操作）会立即执行，这将导致失败。

12.4.6 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器：

A[3:2]	RW	SELECT寄存器的 CTRLSEL位	寄存器	注释
00	读取		IDCODE	制造商代码设置为Cortex [®] -M0+的默认ARM [®] 代码。 0x0BB11477 （标识SW-DP）
00	写		ABORT	
01	读/写	0	DP-CTRL/STAT	目的： - 请求系统或调试上电 - 配置AP访问的传输操作 - 控制比较和验证操作 - 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRECONTROL	用于配置物理串行端口协议（如转换时间的 持续时间）
10	读取		READ RESEND	允许从已损坏的调试软件传输中恢复读取数 据，无 需重复执行原始AP传输。
10	写		SELECT	用于选择当前访问端口和活动的4字寄存器 窗口
11	读/写		READ BUFFER	由于已发出AP访问，因此该读缓冲区非常有 用（在执行 下个AP事务时提供读取AP请求 的结果）。 此读取缓冲区捕获AP中的数据，显示为

前一次读取的结果，无需启动新操作。

12.4.7 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器：

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

地址	A[3:2] 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： - 请求系统或调试上电 - 配置 AP 访问的传输操作 - 控制比较和验证操作 - 读取一些状态标志（上溢和上电确认）

0x8	10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 - 位 31:24: APSEL：选择当前 AP (select the current AP) - 位 23:8: 保留 - 位 7:4: APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 - 位 3:0: 保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）

12.5 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0+ TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C_DEBUGEN)

12.6 BPU（断点单元）

Cortex®-M0+ BPU 实现提供四个断点寄存器。

12.6.2 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见

ARMv6-M ARM®和 ARM® CoreSight 组件技术参考手册。

12.7 DWT（数据观察点）

Cortex®-M0+ DWT 实现提供了两个观察点寄存器组。

12.7.2 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 ARMv6-M ARM® 中所述。

12.7.3 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器(DWT_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 ARMv6-M ARM®。

Cortex®-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

12.8 MCU 调试组件 (DBG)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

12.8.2 对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

12.8.3 对定时器、看门狗的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

12.9 调试模式模块工作状态控制 (DEBUG_STOP)

复位值 0x00000000(仅在 SWD 调试模式下, 此寄存器设置才起作用)

地址: 0x4000_2038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
TICK	CRS	RTC	WWDT	IWDT	PCA	STIM	LPTIM2								
RW	RW	RW	RW	RW	RW	RW	RW								
7	6	5	4	3	2	1	0								
LPTIM1	BTIM789	BTIM456	BTIM123	ATIM	GTIM2	GTIM1	GTIM0								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:16	Reserved	保留
15	TICK	调试时, SYSTICK 使用参考时钟计数功能配置 1: 在 SWD 调试界面下, 暂停 SYSTICK 计数功能 0: 在 SWD 调试界面下, SYSTICK 正常计数功能
14	CRS	调试时, CRS 使用参考时钟计数功能配置 1: 在 SWD 调试界面下, 暂停 CRS 计数功能 0: 在 SWD 调试界面下, CRS 正常计数功能
13	RTC	调试时, RTC 计数功能配置 1: 在 SWD 调试界面下, 暂停 RTC 计数功能 0: 在 SWD 调试界面下, RTC 正常计数功能
12	WWDT	调试时, WWDT 计数功能配置 1: 在 SWD 调试界面下, 暂停 WWDT 计数功能 0: 在 SWD 调试界面下, WWDT 正常计数功能
11	IWDT	调试时, IWDT 计数功能配置 1: 在 SWD 调试界面下, 暂停 IWDT 计数功能 0: 在 SWD 调试界面下, IWDT 正常计数功能
10	PCA	调试时, PCA 计数功能配置 1: 在 SWD 调试界面下, 暂停 PCA 计数功能 0: 在 SWD 调试界面下, PCA 正常计数功能
9	STIM	调试时, STIM 计数功能配置 1: 在 SWD 调试界面下, 暂停 STIM 计数功能 0: 在 SWD 调试界面下, STIM 正常计数功能
8	LPTIM2	调试时, LPTIM2 计数功能配置

		1: 在 SWD 调试界面下, 暂停 LPTIM2 计数功能 0: 在 SWD 调试界面下, LPTIM2 正常计数功能
7	LPTIM1	调试时, LPTIM1 计数功能配置 1: 在 SWD 调试界面下, 暂停 LPTIM1 计数功能 0: 在 SWD 调试界面下, LPTIM1 正常计数功能
6	BTIM789	调试时, BTIM789 计数功能配置 1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能
5	BTIM456	调试时, BTIM456 计数功能配置 1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能
4	BTIM123	调试时, BTIM123 计数功能配置 1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能
3	ATIM	调试时, ATIM 计数功能配置 1: 在 SWD 调试界面下, 暂停 ATIM 计数功能 0: 在 SWD 调试界面下, ATIM 正常计数功能
2	GTIM3	调试时, GTIM3 计数功能配置 1: 在 SWD 调试界面下, 暂停 GTIM3 计数功能 0: 在 SWD 调试界面下, GTIM3 正常计数功能
1	GTIM2	调试时, GTIM2 计数功能配置 1: 在 SWD 调试界面下, 暂停 GTIM2 计数功能 0: 在 SWD 调试界面下, GTIM2 正常计数功能
0	GTIM1	调试时, GTIM1 计数功能配置 1: 在 SWD 调试界面下, 暂停 GTIM1 计数功能 0: 在 SWD 调试界面下, GTIM1 正常计数功能

13. FLASH 控制器（FLASH）

13.1 概述

本系统包含一块 64k/128k/192k/256k 字节（Byte）容量的 FLASH 存储器，共划分为 128/256/384/512 个页（Sector），每个页（Sector）的容量为 512 字节（Byte）。FLASH 控制器支持对 FLASH 存储器的擦除、编程以及读操作。本控制器还支持对 FLASH 存储器擦写保护，以及控制寄存器的写保护。

13.2 FLASH 容量划分

地址	序号		地址	序号
0x0E00 - 0x0FFF	Sector7		0x3FE00 - 0x3FFFF	Sector511
0x0C00 - 0x0DFF	Sector6		0x3FC00 - 0x3DFF	Sector510
0x0A00 - 0x0BFF	Sector5		0x3FA00 - 0x3FBFF	Sector509
0x0800 - 0x09FF	Sector4		0x3F800 - 0x3F9FF	Sector508
0x0600 - 0x07FF	Sector3		0x3F600 - 0x3F7FF	Sector507
0x0400 - 0x05FF	Sector2		0x3F400 - 0x3F5FF	Sector506
0x0200 - 0x03FF	Sector1		0x3F200 - 0x3F3FF	Sector505
0x0000 - 0x01FF	Sector0		0x3F000 - 0x3F1FF	Sector504

表 13-1 FLASH 容量划分

注意：

- 64kByte FLASH 由 Sector0-127 组成；
- 128kByte FLASH 由 Sector0-255 组成；
- 192kByte FLASH 由 Sector0-383 组成；
- 256kByte FLASH 由 Sector0-511 组成。

13.3 功能描述

本控制器支持对 FLASH 的 Byte（8 位）、half-word（16 位）和 word（32 位）三种位宽的数据读写操作。注意，Byte 操作的目标地址必须按字节对齐，half-word 操作的目标地址必须按半字对齐（地址最低位为 1'b0），word 操作的地址必须按字对齐（地址最低两位为 2'b00）。如果目标地址没有按照位宽对齐，则该操作无效，并且 CPU 会进入 Hard Fault 出错中断。

13.3.1 页擦除（Sector Erase）

页擦除每次可以擦除用户指定的一个页（Sector）。擦除操作完成后，页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH_CR.BUSY 变为 0）；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR.BUSY 变为 0）。

Sector 擦除操作步骤如下：

Step1: 配置 FLASH 擦写参数，详见 13.4 章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 2，设置 Flash 操作模式为 Sector 擦除。

Step4: 检查 FLASH_CR.OP 是否为 2，如不为 2 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除该 Sector 的擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8: 对待擦除的 Sector 内的任意地址进行写入任意数据，触发 Sector 擦除。

例：* ((unsigned char *) 0x00000200) = 0x00。

Step9: 等待 FLASH_CR.BUSY 变为 0，Sector 擦除操作完成。

Step10: 如需擦除其它 Sector，重复 Step5 - Step9。

13.3.2 全片擦除（Chip Erase）

全片擦除可以一次性擦除全部的页（Sector）。擦除操作完成后，所有页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则该操作会被禁止。因为

该操作会擦除当前 PC 所在的程序段。如发生这种情况，出错标志位会被置起；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR.BUSY 变为 0）。

Chip 擦除操作步骤如下：

Step1: 配置 FLASH 擦写参数，详见 13.4 章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 3，设置 Flash 操作模式为 Chip 擦除。

Step4: 检查 FLASH_CR.OP 是否为 3，如不为 3 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK0 为 0xFFFFFFFF，去除所有 Sector 的擦写保护。

Step7: 检查 FLASH_SLOCK0 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到则跳转到 Step5。

Step8: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9: 设置 FLASH_SLOCK1 为 0xFFFFFFFF，去除所有 Sector 的擦写保护。

Step10: 检查 FLASH_SLOCK1 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到 则跳转到 Step8。

Step11: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12: 设置 FLASH_SLOCK2 为 0xFFFFFFFF，去除所有 Sector 的擦写保护。

Step13: 检查 FLASH_SLOCK2 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到 则跳转到 Step11。

Step14: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step15: 设置 FLASH_SLOCK3 为 0xFFFFFFFF，去除所有 Sector 的擦写保护。

Step16: 检查 FLASH_SLOCK3 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到 则跳转到 Step14。

Step17: 对待擦除的 Chip 内的任意地址进行写操作，触发 Chip 擦除。

例：* ((unsigned char *) 0x00000000) = 0x00。

Step18: 等待 FLASH_CR.BUSY 变为 0，Chip 擦除操作完成。

13.3.3 写操作 (Program)

写操作只能将 FLASH 内的比特数据由 1 写成 0，故写入数据前应确保待写入的地址内的数据为 0xFF。支持写入 3 种数据长度：Byte (8bits)、Half-word (16bits)、Word (32bits)，写入的数据以小端模式存放在 FLASH 中，即低地址存放数据的低字节。如果写操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成 (FLASH_CR.BUSY 变为 0)；如果该写操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成 (FLASH_CR.BUSY 变为 0)。

Byte 写操作步骤如下：

Step1: 配置 FLASH 擦写参数，详见 13.4 章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入。

Step4: 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8: 对待写入的目标地址进行 Byte 写操作，触发写入操作。

例：* ((unsigned char *) 0x00001231) = 0x5A。

Step9: 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

Step10: 如需写 Byte 到其它地址，重复 Step8 - Step9。

Half-word 写操作步骤如下：

Step1: 配置 FLASH 擦写时间，详见 13.4 章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入。

Step4: 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8: 对待写入的目标地址进行 Half-word 写操作，触发写入操作。

例: $* ((\text{unsigned short int } *) 0x00001232) = 0xABCD$ 。

Step9: 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

Step10: 如需写 Half-word 到其它地址，重复 Step8 - Step9。

Word 写操作步骤如下：

Step1: 配置 FLASH 擦写参数，详见 13.4 章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入。

Step4: 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转 Step5。

Step8: 对待写入的目标地址进行 Word 写操作，触发写入操作。

例: $* ((\text{unsigned int } *) 0x00001234) = 0x55667788$ 。

Step9: 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

Step10: 如需写 Word 到其它地址，重复 Step8 - Step9。

13.3.4 读操作 (Read)

支持读出 3 种数据长度：Byte (8bits)、Half-word (16bits)、Word (32bits)，读出的数据为小端模式，即低地址存放数据的低字节。读操作无需操作步骤，任何时刻都可以读出 FLASH 内的数据。

例：

Byte 读操作: $temp = * ((\text{unsigned char } *) 0x00001231)$

Half-word 读操作 $temp = * ((\text{unsigned short int } *) 0x00001232)$

Word 读操作 $temp = * ((\text{unsigned int } *) 0x00001234)$

13.4 擦写时间

FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。上电时默认装载了 HCLK 为 4MHz 时的擦写参数；如果对 Flash 进行擦写时的 HCLK 频率不为 4MHz，则用户程序应装载 HCLK 频率所对应的擦写参数。对 FLASH 进行操作时，支持的最高 HCLK 为 48MHz。

擦写时序参数相关的寄存器为：FLASH_TNVS、FLASH_TPGS、FLASH_TPROG、FLASH_TSERASE、FLASH_TMERASE、FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV。如果 HCLK 由默认 4MHz 升高为 8MHz，则上述 FLASH_Tx 寄存器的值应该设为默认值的 2 倍，即保持当前 $T_{sysclk} * FLASH\_Tx$ 的结果与默认值相等即可。

以下表格为不同频率下相应的 FLASH 擦写时间参数：

Parameter	4M	8M	16M	24M	32M	48M
TNVS	0x20	0x40	0x80	0xC0	0x100	0x180
TPGS	0x17	0x2E	0x5C	0x8A	0xB8	0x114
TPROG	0x1B	0x36	0x6C	0xA2	0xD8	0x144
TSERASE	0x4650	0x8CA0	0x11940	0x1A5E0	0x23280	0x34BC0
TMERASE	0x222E0	0x445C0	0x88B80	0xCD140	0x111700	0x19A280
TPRCV	0x18	0x30	0x60	0x90	0xC0	0x120
TSRCV	0xF0	0x1E0	0x3C0	0x5A0	0x780	0xB40
TMRCV	0x3E8	0x7D0	0xFA0	0x1770	0x1F40	0x2EE0

表 13-2 不同频率下 FLASH 擦写时间参数

配置系统频率为 8MHz 时擦写参数的操作步骤如下：

Step1: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step2: 向 FLASH_TNVS 寄存器写入 0x40，如读出的该寄存器值不为 0x40，则跳转到上一步。

Step3: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step4: 向 FLASH_TPGS 寄存器写入 0x2E，如读出的该寄存器值不为 0x2E，则跳转到上一步。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 向 FLASH_TPROG 寄存器写入 0x36，如读出的该寄存器值不为 0x36，则跳转到上一步。

Step7: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step8: 向 FLASH_TSERASE 寄存器写入 0x8CA0，如读出的该寄存器值不为 0x8CA0，则跳转到上一步。

Step9: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step10: 向 FLASH_TMERASE 寄存器写入 0x445C0，如读出的该寄存器值不为 0x445C0，则跳转到上一步。

Step11: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step12: 向 FLASH_TPRCV 寄存器写入 0x30，如读出的该寄存器值不为 0x30，则跳转到上一步。

Step13: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step14: 向 FLASH_TSRCV 寄存器写入 0x1E0，如读出的该寄存器值不为 0x1E0，则跳转到上一步。

Step15: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step16: 向 FLASH_TMRCV 寄存器写入 0x7D0，如读出的该寄存器值不为 0x7D0，则跳转到上一步。

13.5 读等待周期

本设备内置的 FLASH 支持的最快取指频率为 24MHz。当 HCLK 频率超过 24MHz 小于 48MHz 时，必须为 FLASH 的读取时间插入等待周期，即设置 FLASH_CR.WAIT 为 1。当插入等待周期时，FLASH 每两个周期才会完成一次读取操作。

13.6 擦写保护

13.6.1 擦写保护位

整个 64k/128k/192k/256k 字节 FLASH 存储器被划分为 128/256/384/512 个页，每 4 个页共用一个擦写保护位。当页被保护时，对该页进行的擦写操作均无效并产生报警

标志位和中断信号。当 FLASH 存储器中的任意页被保护时，对该 FLASH 的全片擦写无效，并产生报警标志位和中断信号。

13.6.2 PC 地址擦写保护

CPU 在 FLASH 中运行程序时，如果当前 PC 指针正好落在待擦写的页地址范围之内，那么该擦写操作无效并产生报警标志位和中断信号。

13.7 寄存器写保护

本模块的重要控制器屏蔽普通的写操作，必须用写序列方式才能修改。

需要通过写序列方式才能更改的寄存器如下所示：

FLASH_TNVS 、 FLASH_TPGS 、 FLASH_TPROG 、 FLASH_TSERASE 、
FLASH_TMERASE、FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV、FLASH_CR、
FLASH_SLOCK0、FLASH_SLOCK1、FLASH_SLOCK2、FLASH_SLOCK3。

不需要通过写序列方式即可更改的寄存器如下所示：

FLASH_ICLR、FLASH_BYPASS。

通过写序列方式修改寄存器值的 具体操作步骤如下所示：

Step1: 向 FLASH_BYPASS 寄存器写入 0x5A5A。

Step2: 向 FLASH_BYPASS 寄存器写入 0xA5A5。

Step3: 对待修改的寄存器写入目标值。

Step4: 验证待修改的寄存器当前值是否与目标相同，如不则跳转到 Step1。

Step5: 执行其他操作。

注意：

- 写 0x5A5A 、 0xA5A5、写目标寄存器，这三步操作之间不可插入本模块内的任何寄存器写操作，否则该 Bypass 序列会失效，需要重新写入 0x5A5A-0xA5A5 序列。

13.8 寄存器

基地址：0x4002 2000

寄存器	偏移地址	描述
FLASH_TNVS	0x00	Tnvs 时间参数
FLASH_TPGS	0x04	Tpgs 时间参数
FLASH_TPROG	0x08	Tprog 时间参数
FLASH_TSERASE	0x0C	Tserase 时间参数
FLASH_TMERASE	0x10	Tmerase 时间参数
FLASH_TPRCV	0x14	Tprcv 时间参数
FLASH_TSRCV	0x18	Tsrcv 时间参数
FLASH_TMRCV	0x1C	Tmrcv 时间参数
FLASH_CR	0x20	控制寄存器
FLASH_IFR	0x24	中断标志寄存器
FLASH_ICLR	0x28	中断标志清除寄存器
FLASH_BYPASS	0x2C	0x5A5A-0xA5A5 Bypass 序列寄存器
FLASH_SLOCK0	0x40	Sector0-127 擦写保护寄存器
FLASH_SLOCK1	0x44	Sector128-255 擦写保护寄存器
FLASH_SLOCK2	0x48	Sector256-383 擦写保护寄存器
FLASH_SLOCK3	0x4C	Sector384-512 擦写保护寄存器

表 13-3 FLASH 寄存器表格

13.8.1 TNVS 参数寄存器（FLASH_TNVS）

偏移地址：0x00

复位值：0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TNVS							
								RW							

位	标记	功能描述
31:9	Reserved	保留
8:0	TNVS	计算公式：TNVS = 8 * HCLK，HCLK 的单位为 MHz。 4MHz 示例：TNVS = 8 * 4 = 32。

13.8.2 TPGS 参数寄存器 (FLASH_TPGS)

偏移地址: 0x04

复位值: 0x0000 0017

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TPGS							
								RW							

位	标记	功能描述
31:9	Reserved	保留
8:0	TPGS	计算公式: $TPGS = 5.75 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TPGS = 5.75 * 4 = 23$ 。

13.8.3 TPROG 参数寄存器 (FLASH_TPROG)

偏移地址: 0x08

复位值: 0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TPROG							
								RW							

位	标记	功能描述
31:9	Reserved	保留
8:0	TPROG	计算公式: $TPROG = 6.75 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TPROG = 6.75 * 4 = 27$ 。

13.8.4 TSERASE 寄存器 (FLASH_TSERASE)

偏移地址: 0x0C

复位值: 0x0000 4650

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

Reserved	TSERASE
	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSERASE															
RW															

位	标记	功能描述
31:18	Reserved	保留
17:0	TSERASE	计算公式: $TSERASE = 4500 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TSERASE = 4500 * 4 = 18000 (0x4650)$ 。

13.8.5 TMERASE 参数寄存器 (FLASH_TMERASE)

偏移地址: 0x10

复位值: 0x000222E0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMERASE															
RW															

位	标记	功能描述
31:21	Reserved	保留
20:0	TMERASE	计算公式: $TMERASE = 35000 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TMERASE = 35000 * 4 = 140000$ 。

13.8.6 TPRCV 参数寄存器 (FLASH_TPRCV)

偏移地址: 0x14

复位值: 0x0000 0018

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TPRCV											

	RW

位	标记	功能描述
31:12	Reserved	保留
11:0	TPRCV	计算公式: $TPRCV = 6 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TPRCV = 6 * 4 = 24$ 。

13.8.7 TSRCV 参数寄存器 (FLASH_TSRCV)

偏移地址: 0x18

复位值: 0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TSRCV											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	TSRCV	计算公式: $TSRCV = 60 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TSRCV = 60 * 4 = 240$ 。

13.8.8 TMRCV 参数寄存器 (FLASH_TMRCV)

偏移地址: 0x1C

复位值: 0x0000 03E8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TMRCV											
				RW											

位	标记	功能描述
31:14	Reserved	保留
13:0	TMRCV	计算公式: $TMRCV = 250 * HCLK$, HCLK的单位为MHz。 4MHz示例: $TMRCV = 250 * 4 = 1000$ 。

13.8.9 CR 寄存器 (FLASH_CR)

偏移地址: 0x20

复位值: 0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DPSTB_EN	Reserved	IE	BUSY	WAIT	OP				
						RW		RW	RO	RW	RW				

位	标记	功能描述
31:10	Reserved	保留
9	DPSTB_EN	FLASH dpstb使能Mask位; 0: 当系统进入deepsleep模式, FLASH不进入低功耗模式; 1: 当系统进入 deepsleep 模式, FLASH 进入低功耗模式;
8:7	Reserved	保留
6:5	IE	IE[6]: FLASH 擦写被保护地址中断使能; 0: 不使能; 1: 使能 IE[5]: FLASH 擦写 PC 值中断使能; 0: 不使能; 1: 使能
4	BUSY	空闲/忙标志位; 0: 空闲状态; 1: 忙状态;
3:2	WAIT	读FLASH 周期; 0~24MHz: 00/11, 1个周期; 24~48MHz: 01: 2个周期; 48~72MHz: 10:3个周期; (本系列产品最高时钟为48MHz)
1:0	OP	FLASH 操作; 00: 读 (read); 01: 写 (program); 10: 页擦除 (sector erase); 11: 全片擦除 (chip erase)

13.8.10 IFR 寄存器 (FLASH_IFR)

偏移地址: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IF1	IF0
														RO	RO

位	标记	描述
31:2	Reserved	保留
1	IF1	擦写保护报警中断标志位
0	IF0	擦写 PC 地址报警中断标志位

13.8.11 ICLR 寄存器 (FLASH_ICLR)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8																																
Reserved																																																							
7							6							5							4							3							2							1							0						
Reserved																		ICLR1							ICLR0																														
																		R1W0							R1W0																														

位	标记	功能描述
31:4	Reserved	保留
3:2	Reserved	写无效, 读为0x3
1	ICLR1	清除保护报警中断标志位; 写0清除; 写1无效;
0	ICLR0	清除 PC 地址报警中断标志位; 写0清除; 写1无效;

13.8.12 BYPASS 寄存器 (FLASH_BYPASS)

偏移地址: 0x2C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYSEQ[15:0]															
WO															

位	标记	功能描述
31:16	Reserved	保留
15:0	BYSEQ	在修改本模块寄存器之前, 必须对 BYSEQ[15:0] 寄存器写入 0x5a5a-0xa5a5 序列。每次写入该 Bypass 序列后, 只可以修改一次寄存器。如需再次修改寄存器, 必须再次输入 0x5a5a-0xa5a5 序列。

13.8.13 SLOCK0 寄存器 (FLASH_SLOCK0)

偏移地址: 0x40

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK0[15:0]															
RW															

位	标记	功能描述
31:0	SLOCK0	Sector 擦写保护位; 0: 不允许擦写; 1: 允许擦写 Bit[0] 对应: Sector0-1-2-3 Bit [1] 对应: Sector4-5-6-7 Bit [2] 对应: Sector8-9-10-11 Bit [3] 对应: Sector12-13-14-15 Bit [31] 对应: Sector124-125-126-127

13.8.14 SLOCK1 寄存器 (FLASH_SLOCK1)

偏移地址: 0x44

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK1[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK1[15:0]															
RW															

位	标记	功能描述
31:0	SLOCK1	Sector 擦写保护位; 0: 不允许擦写; 1: 允许擦写 Bit [0] 对应: Sector128-129-130-131 Bit [1] 对应: Sector132-133-134-135 Bit [2] 对应: Sector136-137-138-139 Bit [3] 对应: Sector140-141-142-143

		 Bit [31] 对应: Sector252-253-254-255

13.8.15 SLOCK2 寄存器 (FLASH_SLOCK2)

偏移地址: 0x48

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK2[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK2[15:0]															
RW															

位	标记	功能描述
31:0	SLOCK2	Sector 擦写保护位; 0: 不允许擦写; 1: 允许擦写 Bit[0] 对应: Sector256-257-258-259 Bit [1] 对应: Sector260-261-262-263 Bit [2] 对应: Sector264-265-266-267 Bit [3] 对应: Sector268-269-270-271 Bit [31] 对应: Sector380-381-382-383

13.8.16 SLOCK3 寄存器 (FLASH_SLOCK3)

偏移地址: 0x4C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK3[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK3[15:0]															
RW															

位	标记	功能描述
31:0	SLOCK3	Sector 擦写保护位; 0: 不允许擦写; 1: 允许擦写 Bit[0] 对应: Sector384-385-386-387 Bit [1] 对应: Sector388-389-390-391

		Bit [2] 对应: Sector392-393-394-395 Bit [3] 对应: Sector396-397-398-399 Bit [31] 对应: Sector508-509-510-511

14. RAM 控制器 (RAM)

14.1 概述

本系统中包含一块容量为 8k/16k/24k/32k 字节 (Byte) 的 SRAM, 支持字节 (8 位)、半字 (16 位 bits)、字 (32 位) 三种读写操作。可在系统时钟频率下进行读写操作, 无须等待周期。此外, 本控制器还支持奇偶校验, 可对每个字节 (Byte) 的 SRAM 数据进行奇偶校验, 并产生奇偶校验出错中断。

14.2 功能描述

14.2.1 RAM 地址范围

RAM 在系统映射中的地址范围如下表所示:

地址范围	大小	Memory类型
0x2000_0000 - 0x2000_7FFF	32kByte	SRAM

表 14-1 RAM 地址映射

14.2.2 读写位宽

本控制器支持 字节 (8 位)、半字 (16 位)、字 (32 位) 三种位宽的读写操作。字节操作的地址必须按字节对齐, 半字操作的目标地址必须按半字对齐 (地址最低位为 1'b0), 字操作的地址必须按字对齐 (地址最低两位为 2'b00)。如果读写操作的目标地址没有按照位宽规定对齐, 该操作无效, 并且系统会产生 Hard Fault 出错中断。

14.2.3 奇偶校验

本控制器支持 SRAM 数据的奇偶校验。当对 SRAM 写数据时, 对每个字节的数据做奇偶校验, 并把 1bit 校验值和 8bits 数据一起存入 SRAM 中。当对 SRAM 读数据时, 控制器会读取 8bits 数据和 1bit 校验值, 并做奇偶校验, 如果校验出错, 则置位奇偶校验出错标志位, 在中断使能情况下, 会产生出错中断。

注意：在奇偶校验使能时，在读取 SRAM 数据前，必须初始化 SRAM，否则可能会误触发奇偶校验报警标志位或者中断。

14.3 寄存器

基地址：0x4002 2400

寄存器	偏移地址	描述
RAM_CR	0x00	控制寄存器
RAM_ERRADDR	0x04	出错地址寄存器
RAM_IFR	0x08	出错中断标志寄存器
RAM_ICLR	0x0C	出错中断标志清除寄存器

表 14-2 寄存器基地址

14.3.1 控制寄存器（RAM_CR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IE	RSV
														RW	

位	标记	功能描述
31:2	Reserved	保留
1	IE	出错报警中断使能信号；1：使能报警中断，0：关闭报警中断；
0	Reserved	保留

14.3.2 奇偶校验出错地址寄存器（RAM_ERRADDR）

偏移地址：0x04

复位值：0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	ERRADDR														

	RO

位	标记	功能描述
31:15	Reserved	固定为0x4000
14:0	ERRADDR	奇偶校验出错Byte地址；中断标志位清除后，该地址同时被清除；

14.3.3 出错中断标志寄存器（RAM_IFR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ERR	
														RO	

位	标记	功能描述
31:1	Reserved	保留
0	ERR	奇偶校验出错标志位

14.3.4 出错中断标志清除寄存器（RAM_ICLR）

偏移地址：0x0C

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved														ERRCLR									
														R1W0									

位	标记	功能描述
31:1	Reserved	
0	ERRCLR	出错中断标志清除位；写1：无效，写0：清零

15. DMA 控制器（DMAC）

15.1 概述

DMAC（直接内存访问控制器）功能块可以不通过 CPU 高速传输数据。使用 DMAC 能提高系统性能。

DMAC 概述：

- DMAC 配有独立的总线，所以即便是在使用 CPU 总线的时候，DMAC 也可进行传输操作。
- 由 4 条通道组成，能执行 4 种相互独立的 DMA 传输。
- 可设置传输目标地址、传输源地址、传输数据大小、传输请求源以及传输模式，并能控制各通道的传输操作启动、传输的强行终止以及传输的暂停。
- 可控制所有通道批量传输的启动、强行终止及暂停。
- 多通道同时操作时，可用固定方法或循环方法选择操作通道的优先级。
- 支持使用外设中断信号的硬件 DMA 传输。
- 遵从系统总线(AHB)，支持 32 位地址空间(4GB)。

各通道功能概述：

- 可递增或固定传输源和传输目标地址。
- 具有传输源和传输目标地址重装功能（即传输完成后回到原设置值的功能）。
- 传输数据的大小可以从以下三种规格中选择：
 - 1) 传输数据宽度：（从字节/半字/字中选择）
 - 2) 设置块数：（从 1 至 16 中选择）
 - 3) 设置传输长度：（从 1 至 65536 中选择）
- 可指定是否发出传输成功完成和传输出错中断。
- 可选择以下四种传输模式：
 - 1) 软件-Block 传输
 - 2) 软件-Burst 传输
 - 3) 硬件-Block 传输
 - 4) 硬件-Burst 传输

传输模式概述:

软件传输是通过 CPU 直接指令启动 DMAC 的方法。

硬件传输是外设发出传输请求时将外设的请求信号作为 DMAC 的传输请求信号以直接启动 DMAC 的方法。外设请求可来自于 UART1/2, LPUART1/2, SPI1/2, TIM1/2/3/4/5/6, ADC, LCD。当这些外设具有数据传输请求时,数据可以通过当前通道预先的传输配置来实现传输,而不用 CPU 介入。

15.2 功能框图

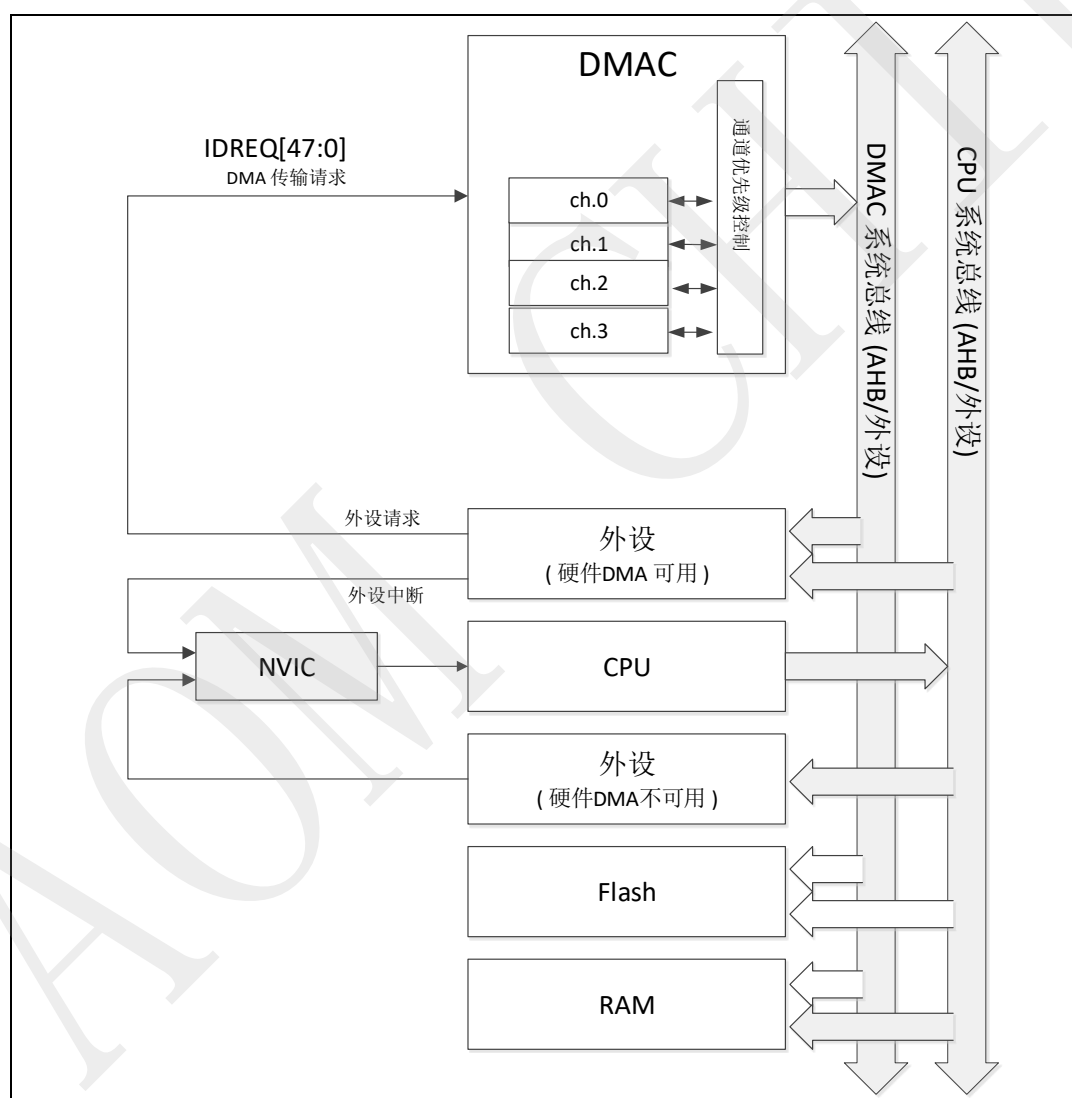


图 15-1 功能框图

框图说明

- DMAC

DMAC 最多配置 4 条通道。各通道执行独立传输。通道之间发生冲突时，优先级控制器控制这些通道的传输操作。

- 系统连接

DMAC 通过系统总线连接至 CPU、FLASH、RAM 以及各种外设。DMAC 自身有独立于 CPU 总线的总线，可在访问 CPU 总线时执行传输操作。通过指定各通道传输目标地址和传输源地址以进行存储器与外设之间传输数据，系统中的任何地址区域均可访问。由于部分区域不能从 DMAC 访问，需检查内存映射。

- 硬件传输请求连接

被中断控制器（指图 14-1 中的 DRQSEL）所选择支持硬件传输的外设中断信号，可以作为发送至 CPU 的中断信号或者同时作为发送至 DMAC 的 DMA 传输请求信号。

根据硬件请求执行 DMA 传输时，通过设置 DRQSEL，连接各外设的中断信号，作为提前发给 DMAC 的传输请求信号。不支持硬件传输的外设中断信号不能作为 DMA 传输请求信号。

输入 DMAC 的 DMA 传输请求信号有 48 个。硬件传输时，DMAC 各通道从上述 40 条传输请求信号中选择 1 条信号执行操作。使用 CONFA: TRI_SEL 寄存器进行选择。详见以下表格：

IDREQ 编号	对应外设传输请求
0	SPI1 RX
1	SPI1 TX
2	SPI2 RX
3	SPI2 TX
4	ADC JRQ0 RX
5	ADC SRQ0 RX
6	LCD TX

7	/
8	UART1_RX
9	UART1_TX
10	UART2_RX
11	UART2_TX
12	LPUART1_RX
13	LPUART1_TX
14	LPUART2_RX
15	LPUART2_TX
16	DAC1
17	DAC2
18	GTIM1_CHA
19	GTIM1_CHB
20	GTIM2_CHA
21	GTIM2_CHB
22	GTIM3_CHA
23	GTIM3_CHB
24	ADVTIM_CHA
25	ADVTIM_CHB
26	STIM_overflow
27	STIM_CH1
28	STIM_CH2
29	STIM_CH3
30	STIM_CH4
31	STIM_CCT
32	UART3_RX
33	UART3_TX

34	UART4_RX
35	UART4_TX
36	I2S_L
37	I2S_R
38	/
39	/
40	SPI3_RX
41	SPI3_TX
42	SPI4_RX
43	SPI4_TX
44	UART5_RX
45	UART5_TX
46	UART6_RX
47	UART6_TX

表 15-1 传输请求和对应外设请求信号

- DMAC 中断

尽管图 14-1 未作说明，用于传输完成通知的中断信号连接至 NVIC。各通道有 8 个中断源，合并成 1 个输出。

15.3 基本功能

本节说明各传输模式下 DMAC 的操作。为了加速各模块间的数据搬运，PCLK 要与 HCLK 同频率下，DMA 才能发挥性能优势。

15.3.1 软件-块（Block）传输

本节说明软件-块传输。

图 14-2 所示为软件-块传输操作示例。示例中应用以下设置：

- 传输模式： 软件请求块传输 (ST=1, TRI_SEL[5:0]=000000, MODE=00)
- 传输源启动地址： SA(SRC_ADR=SA)

- 传输源地址控制：可递增和重装 (FS=0, RS=1)
- 传输目标启动地址：DA(DST_ADR=DA)
- 传输目标地址控制：不可递增和重装 (FD=0, RD=0)
- 传输数据大小：半字 (16 位)，块数= 2，传输长度= 3 (WIDTH=01, BC=1, TC=2)
- BC/TC 重装：可重装 (RC=1)

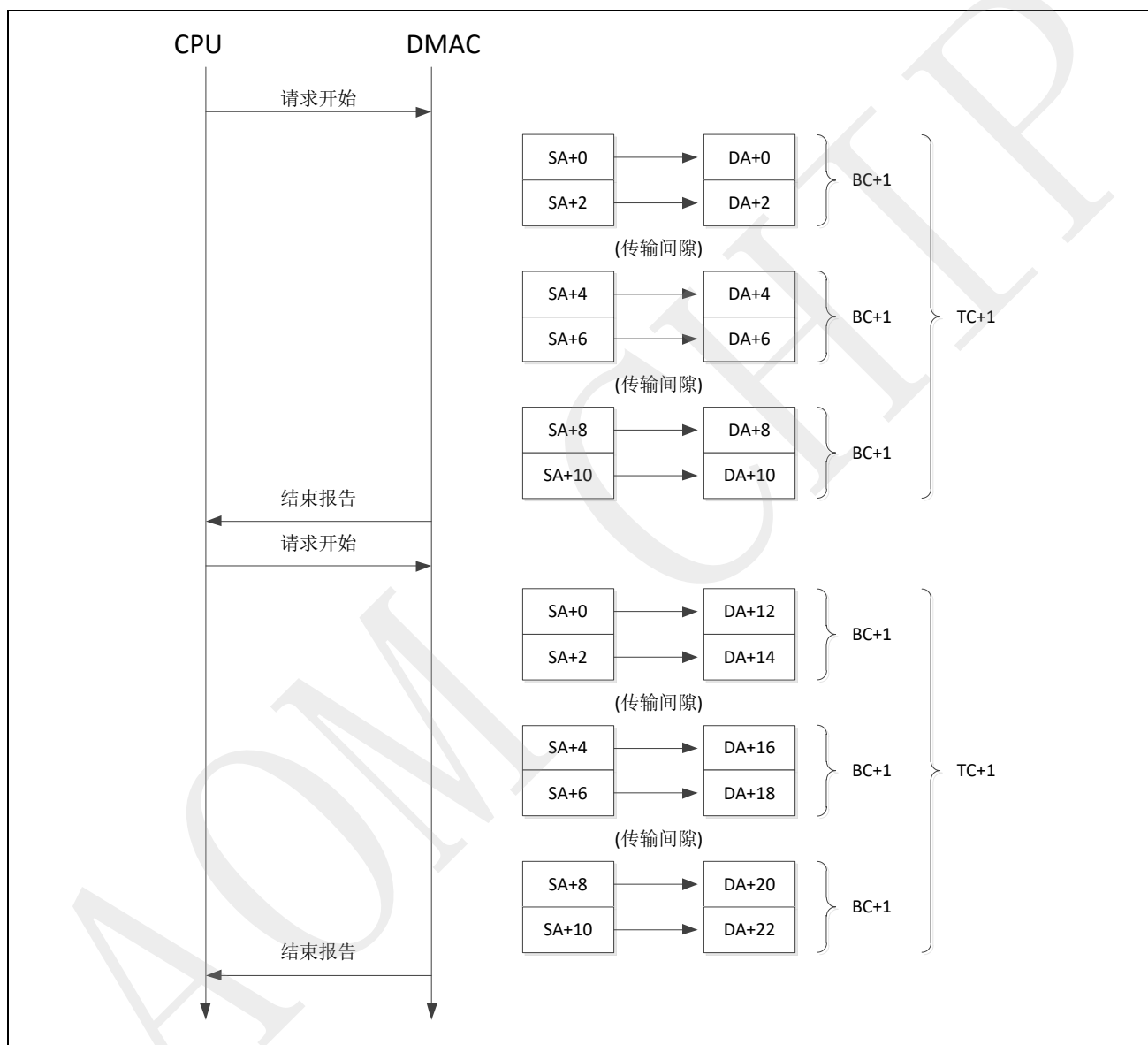


图 15-2 软件-块传输操作示例

通过 CPU 设置传输内容并指示启动传输后，DMAC 执行以下操作：

- 由于指定了传输数据位宽，传输以半字 (16 位) 执行。
- 根据传输源及传输目标的启动地址、数据位宽以及递增/固定的规定，在源地址至目的地址区域执行块数 (=BC+1) 传输。

- 块传输时，每完成一次块传输则产生一个传输间隔。
- DMAC 根据传输数量 ($=TC+1$)和传输块数 ($=BC+1$)执行数据传输。根据 CPU 传输请求所传输的数据量大小等于“数据位宽 (WIDTH) $\times$ 块数 ($BC+1$) $\times$ 传输数量 ($TC+1$)”。
- 传输完成后，DMAC 通知 CPU 传输完成。
- 如传输完成后再次指示启动传输，因传输源地址已设置为重装 ($RS=1$)，将从之前的传输源地址重新启动传输。如果未将传输目标地址设置为重装 ($RD=0$)，将从先前传输结束地址的下一地址开始传输。同样，如果指定了 BC/TC 重装，下次传输将重装与上次传输相同的块数和传输数量值。
- 传输间隔期间不执行传输，插入传输间隔的目的是防止某一条 DMAC 通道占用系统总线的访问权。如多条通道均有传输请求，DMAC 切换在传输间隔计时期间执行传输操作的通道。通过调节 BC 及 TC 的设置可控制传输间隔生成的频率。

此外，传输间隔计时期间总线访问权也转至 CPU。此产品的系统总线为多层配置，带有专用于 DMA 的特殊系统总线。因此，如 CPU 与访问目标之间无冲突，CPU 操作的同时可执行传输。即使 CPU 与访问目标之间发生冲突，只要 DMAC 传输处在不同的地址段 (RAM 和外设，或 FLASH 和 RAM 等)，CPU 操作几乎不受影响。但如传输处在相同的地址段 (RAM 和 RAM 等)，CPU 性能可能会受影响，具体取决于所用块数，对此必须引起注意。

15.3.2 软件-突发 (Burst) 传输

本节说明软件-突发传输。

图 14-3 所示为软件-突发传输操作示例。示例中应用以下设置：

- 传输模式： 软件请求突发传输 ($ST=1$, $TRI\_SEL[5:0]=000000$, $MODE=01$)
- 传输源启动地址： SA($SRC\_ADR=SA$)
- 传输源地址： 固定和可重装 ($FS=1$, $RS=1$)
- 传输目标启动地址： DA($DST\_ADR=DA$)
- 传输目标地址： 递增和不可重装 ($FD=0$, $RD=0$)

- 传输数据大小： 字（32 位），块数=3，传输数量=2(WIDTH=10,BC=2,TC=1)
- 传输数量重装： 重装传输数量 (RC=1)

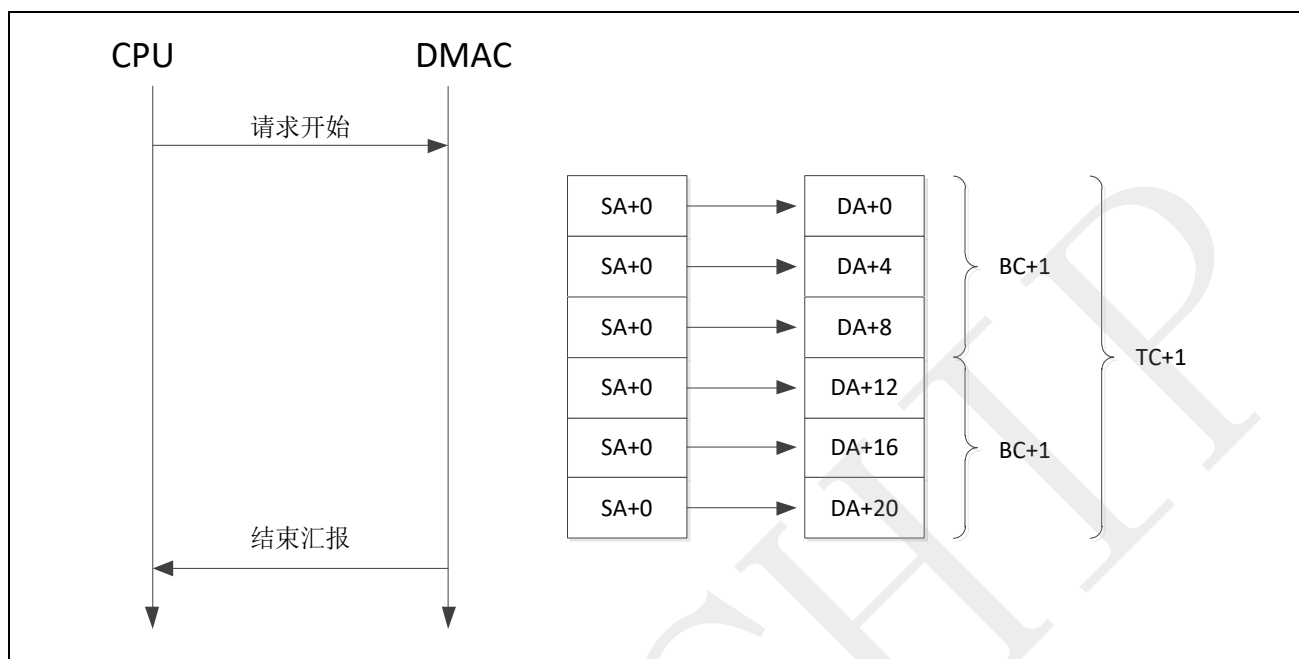


图 15-3 软件-突发传输操作示例

通过 CPU 设置传输内容并指示启动传输后，DMAC 执行以下操作：

- 由于指定了传输数据位宽，传输以字（32 位）执行。
- 根据传输源及传输目标的启动地址、数据位宽以及递增/固定的规定，在地址 SA 至地址 DA 区域执行块数 (=BC+1) 传输。由于指定了固定的传输源地址，传输源地址与传输源启动地址 (SA+0) 相同。
- 突发传输时，连续执行传输，不产生传输间隔。
- DMAC 根据传输数量 (=TC+1) 执行块数 (=BC+1) 数据传输。根据 CPU 传输请求所传输的数据大小等于"数据位宽 (WIDTH) × 块数 (BC+1) × 传输数量 (TC+1)。"
- 传输完成后，DMAC 通知 CPU 传输完成。

与块传输不同的是突发传输不产生传输间隔。由于受控通道占用了系统总线访问权，该通道可作为优先通道。

15.3.3 硬件-块（Block）传输

本节说明硬件-块传输。

图 14-4 所示为硬件-块传输操作示例。示例中应用以下设置。其中省略了传输源及传

输目标的地址设置以及传输数据位宽的设置。

- 传输模式： 硬件-块传输（ST=0，TRI_SEL= 传输请求源的外设，MODE=00）
- 传输数据大小： 块数= 4，传输数量= 3 (BC=3,TC=2)

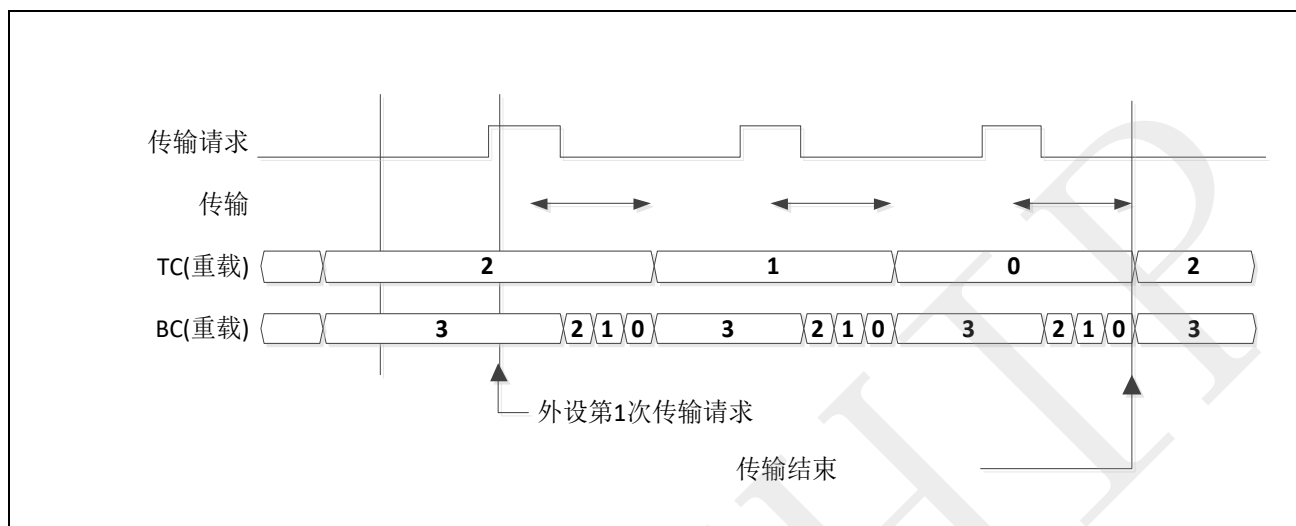


图 15-4 硬件-块传输操作示例

硬件-块传输操作如下：

通过 CPU 指定传输内容，指令启动操作。DMAC 等待外设的传输请求。收到传输请求后，DMAC 执行块数 (=BC+1) 传输，然后等待下一次传输请求。等待时产生传输间隔。总共收到 (TC+1) 次传输请求，DMAC 执行 (TC+1) 次传输操作。执行的传输总数为 (BC+1) × (TC+1)。外设发出的传输请求必须与 DMAC 传输数量 (TC+1) 相匹配。完成所有传输后，DMAC 通知 CPU 传输完成。

15.3.4 硬件-突发 (Burst) 传输

本节说明硬件-突发传输。

图 14-5 所示为硬件-突发传输操作示例。示例中应用以下设置。其中省略了传输源及传输目标的地址设置以及传输数据位宽的设置。

- 传输模式： 输请求源的外设，MODE=01）
- 传输数据大小： 块数= 4，传输数量= 5 (BC=3,TC=4)

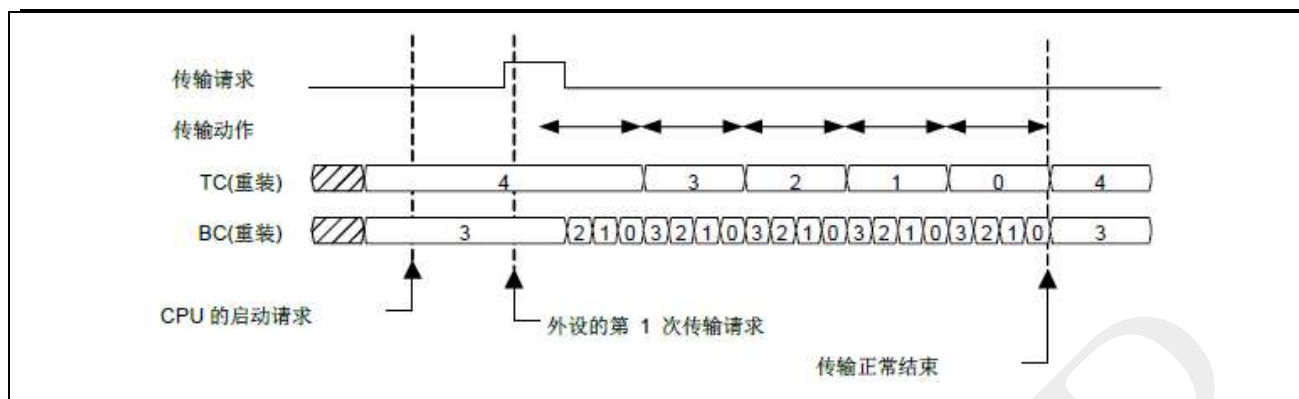


图 15-5 硬件-突发传输操作示例

硬件-突发传输操作如下所述：

通过 CPU 指定传输内容，指令启动操作。DMAC 等待外设的传输请求。收到第一次传输请求后，DMAC 执行通过 $(BC+1) \times (TC+1)$ 计算出的所有次数的传输。硬件-突发传输过程不产生传输间隙。完成所有传输后，DMAC 通知 CPU 传输完成。

15.3.5 通道优先级控制

如 2 个通道均有传输请求，各通道传输间隔计时期间 DMAC 切换其中要传输的通道。此时根据优先控制确定执行传输的下一条通道。可通过 PRIO 从固定优先或循环优先中选择优先控制。

固定优先模式操作 (PRIO=0)

固定优先模式下，所有接到传输请求的通道中最小编号通道优先执行传输操作。（优先顺序：CH0>CH1）

首先执行最高优先级通道的传输（CH0）。最高优先级通道在传输间隔计时期间停止传输操作时，执行第二优先通道的传输操作（CH1）。因此，第一优先通道和第二优先通道交替执行传输操作。

循环优先模式操作 (PRIO=1)

循环优先模式下，所有通道均等地执行其传输操作。

15.4 寄存器

DMAC 基地址：0x4002 0000

寄存器	偏移地址	控制通道	描述
DMAC_CONF	0x00	ALL	所有通道配置寄存器

DMAC_CONFA0	0x10	CH0	通道0配置A寄存器
DMAC_CONFBO	0x14	CH0	通道0配置B寄存器
DMAC_SRCADRO	0x18	CH0	通道0传输源地址寄存器
DMAC_DSTADRO	0x1C	CH0	通道0传输目的地址寄存器
DMAC_CONFA1	0x20	CH1	通道1配置A寄存器
DMAC_CONFBI	0x24	CH1	通道1配置B寄存器
DMAC_SRCADR1	0x28	CH1	通道1传输源地址寄存器
DMAC_DSTADR1	0x2C	CH1	通道1传输目的地址寄存器
DMAC_CONFA2	0x30	CH2	通道2配置A寄存器
DMAC_CONFBI2	0x34	CH2	通道2配置B寄存器
DMAC_SRCADR2	0x38	CH2	通道2传输源地址寄存器
DMAC_DSTADR2	0x3C	CH2	通道2传输目的地址寄存器
DMAC_CONFA3	0x40	CH3	通道3配置A寄存器
DMAC_CONFBI3	0x44	CH3	通道3配置B寄存器
DMAC_SRCADR3	0x48	CH3	通道3传输源地址寄存器
DMAC_DSTADR3	0x4C	CH3	通道3传输目的地址寄存器

表 15-2 DMAC 寄存器列表

15.4.1 DMAC_CONF

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	ST	RSV	PRI0	HALT[3:0]				Reserved							
RW	RW		RW	RW											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

位	标记	功能描述
31	EN	该位控制所有通道的使能和禁止传输操作。 当该位置“1”时，使能所有通道的操作，每个通道按照各自设置工作。 当该位清“0”时，所有通道操作都被禁止，直到该位重新置“1”。正在传输的通道强制停止传输。
30	ST	该位指示所有通道的传输状态。 如果下列任何情况在传输中发生，DMAC 将该位置“1”。 - 当“0”写入 CONF:EN 位，并且所有通道传输结束。 - 当除“0000”外的值写入 CONF:HALT 位，并且所有通道传输暂停。 当 CONF:EN=1 和 CONF:HALT=0000 有效，并且所有的通道使能操作，DMAC 将本位清零。 尽管本位的属性是 RW，CPU 写“1”不影响 DMAC 的操作。然而如果 CONF 寄存器需要更新而又不想改变当前位，应该首先读出这个位，然后重新写入同样的值。

29	Reserved	
28	PRI0	PRI0控制各个通道的传输优先级。 当该位置“0”，各通道优先级固定。CH0 优先级 > CH1 优先级。 当该位置“1”，各通道优先级采用轮询方式决定。
27:24	HALT[3:0]	HALT位控制所有通道的传输暂停或取消。 当这些位设置为“0000”以外的值时，所有正在传输的通道进入暂停状态。当设置为“0000”，所有传输继续。 即便来自外部或者外设的传输请求有效，暂停状态的通道忽略传输请求。如果是块传输和突发传输，即便暂停清除，相关的通道也不启动传输。为了完成传输过程中暂停的传输，需要在暂停取消后发动另外的传输请求。 这些位可以用于暂停传输而不复位该通道的配置寄存器。
23:0	Reserved	

15.4.2 DMAC_CONFA0/1/2/3

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENS	PAS	ST	TRI_SEL[6:0]							Reserved	BC[3:0]				
RW	RW	RW	RW								RW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC[15:0]															
RW															

位	标记	功能描述
31	ENS	ENS位控制单独通道的使能和禁止功能。 当该位置“1”，相关通道使能，等待触发，开始传输功能(CONF:EN 必须置“1”)。 如果 MSK 位(CONFB[0])位清“0”，DMAC 在完成传输后清除本位，即“0”。 当该位为“0”，相关通道禁止，不进行数据传输。同时，如果正好在传输过程中，那么停止传输。 该位可用于强制停止正在传输操作的相关通道，同时复位配置寄存器。
30	PAS	PAS位控制各自通道的传输暂停或取消。 当该位置“1”，相关正在传输的通道进入暂停状态。清除该位到0，从新开始传输。 该位清除为“0”当通道的传输结束后。 即便外部或外设传输请求有效，暂停状态通道忽略传输请求。如果是块传输和突发传输，相关通道不会开始传输，即使是清除了暂停位。为了完成暂停的传输，在清除暂停后发送另外的传输请求。 该位可用于暂停传输而不复位相关通道的配置寄存器。
29	ST	ST 位用于产生单独通道的软件传输请求。 当该位置“1”，软件传输触发产生，相关通道开始传输。传输完成后，DMAC 清除该位至“0”。 当该位在传输中置“0”，传输停止。
28:22	TRI_SEL[6:0]	TRI_SEL[6:0]位选择传输请求的触发源。 设置 IS[6:0]为：

TRI_SEL[6:0]	对应外设传输请求
"0000000"	传输触发设定为软件请求
"1000000"	SPI1 RX
"1000001"	SPI1 TX
"1000010"	SPI2 RX
"1000011"	SPI2 TX
"1000100"	ADC_JRQ0 RX
"1000101"	ADC_SRQ0 RX
"1000110"	LCD TX
"1000111"	/
"1001000"	UART1 RX
"1001001"	UART1 TX
"1001010"	UART2 RX
"1001011"	UART2 TX
"1001100"	LPUART1 RX
"1001101"	LPUART1 TX
"1001110"	LPUART2 RX
"1001111"	LPUART2 TX
"1010000"	DAC1
"1010001"	DAC2
"1010010"	GTIM1_CHA
"1010011"	GTIM1_CHB
"1010100"	GTIM2_CHA
"1010101"	GTIM2_CHB
"1010110"	GTIM3_CHA
"1010111"	GTIM3_CHB
"1011000"	ADVTIM_CHA
"1011001"	ADVTIM_CHB
"1011010"	STIM_overflow
"1011011"	STIM_CH1
"1011100"	STIM_CH2
"1011101"	STIM_CH3
"1011110"	STIM_CH4
"1011111"	STIM_CCT
"1100000"	UART3 RX
"1100001"	UART3 TX
"1100010"	UART4 RX
"1100011"	UART4 TX
"1100100"	I2S_L
"1100101"	I2S_R

		<table><tr><td>"1100110"</td><td>/</td></tr><tr><td>"1100111"</td><td>/</td></tr><tr><td>"1101000"</td><td>SPI3_RX</td></tr><tr><td>"1101001"</td><td>SPI3_TX</td></tr><tr><td>"1101010"</td><td>SPI4_RX</td></tr><tr><td>"1101011"</td><td>SPI4_TX</td></tr><tr><td>"1101100"</td><td>UART5_RX</td></tr><tr><td>"1101101"</td><td>UART5_TX</td></tr><tr><td>"1101110"</td><td>UART6_RX</td></tr><tr><td>"1101111"</td><td>UART6_TX</td></tr></table>	"1100110"	/	"1100111"	/	"1101000"	SPI3_RX	"1101001"	SPI3_TX	"1101010"	SPI4_RX	"1101011"	SPI4_TX	"1101100"	UART5_RX	"1101101"	UART5_TX	"1101110"	UART6_RX	"1101111"	UART6_TX
"1100110"	/																					
"1100111"	/																					
"1101000"	SPI3_RX																					
"1101001"	SPI3_TX																					
"1101010"	SPI4_RX																					
"1101011"	SPI4_TX																					
"1101100"	UART5_RX																					
"1101101"	UART5_TX																					
"1101110"	UART6_RX																					
"1101111"	UART6_TX																					
21:20	Reserved																					
19:16	BC[3:0]	BC[3:0]位指定块/突发传输的个数。 设置为请求传输时，设置"0000"。 设置"BC[3:0]=块数量 - 1"。最大允许传输的块数为 16。 这几位在传输的工程中可以去读。通常情况，当一个源地址访问，或者一个目的地址的访问成功完成，BC[3:0] 减去 1。 - 在 CONFB:RC=1 的情况下： 传输开始时设置的值，在传输完成后重载。 - 在 CONFB:RC=0 的情况下： 成功完成传输后，该位清零，同时在传输暂停引起的传输非成功完成时，该值保持不变。																				
15:0	TC[15:0]	TC[15:0] 位用于指定块/突发/请求传输的传输次数。 设定值"TC = 传输数量 - 1"。最大传输数为 65536。 传输时，这些值是可以读的。正常情况，当一次块传输完成，TC 减去 1。 - 在 CONFB:RC=1 的情况下 传输开始时设置的值，在传输完成后重载。 - 在 CONFB:RC=0 的情况下 成功完成传输后，该位清零，同时在传输暂停引起的传输非成功完成时，该值保持不变。																				

15.4.3 DMAC_CONFB0/1/2/3

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		MODE[1:0]		WIDTH[1:0]		FS	FD	RC	RS	RD	ERR_IE	FIS_IE	STAT[2:0]		
		RW		RW		RW	RW	RW	RW	R	RW	RW	RW		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														MSK	
														RW	

位	标记	功能描述
31:30	Reserved	

29:28	MODE[1:0]	该位控制传输模式： 00: Block传输（初始值） 01: Burst传输 10: Reserved 11: Reserved
27:26	WIDTH[1:0]	该位指示传输数据的宽度： 00: 字节（8-Bit）（初始值） 01: 半字（16-Bit） 10: 字（32-Bit） 11: Reserved
25	FS	该位用于指定源地址自增或者固定： 0: 按 CONFB:WIDTH[1:0]的数据位宽自增传输源地址（初始值） 1: 固定传输源地址
24	FD	该位用于指定目的地址自增或者固定： 0: 按 CONFB:WIDTH[1:0]的数据位宽自增传输目的地址（初始值） 1: 固定传输目的地址
23	RC	该位控制CONFA:BC[3:0]和CONFA:TC[15:0]的重载功能。 当该位置”1”时，CONFA:BC[3:0]和 CONFA:TC[15:0]在传输完成后重新加载传输开始时的设定值。 0: 禁止重新加载 BC/TC 值（初始值） 1: 使能重新加载 BC/TC 值
22	RS	该位控制传输源地址的重载功能 0: 禁止重新加载传输源地址（初始值） 1: 使能重新加载传输源地址
21	RD	该位控制传输目的地址的重载功能 0: 禁止重新加载传输目的地址（初始值） 1: 使能重新加载传输目的地址
20	ERR_IE	该位使能或禁止在传输未成功完成中断的产生。 0: 禁止传输出错时产生中断（初始值） 1: 使能传输出错时产生中断 当该位设置为”1”时，在如下情况下，传输出错中断发出。 — 地址溢出 CONFB:STAT[2:0]=”001” — 由外设发出的停止传输引起的停止，或者设置 DMAC_CONF:EN、DDMAC_CONFA:ENS 引起的禁止传输 CONFB:STAT[2:0]=010 — 传输源地址访问错误 CONFB:STAT[2:0]=011 — 传输目的地址访问错误 CONFB:STAT[2:0]=100
19	FIS_IE	该位使能或禁止传输成功完成中断的产生。 0: 禁止传输成功完成时产生中断（初始值） 1: 使能传输成功完成时产生中断 当该位置”1”时，如果 CONF:STAT 在传输完成时为成功完成传输，中断发生。
18:16	STAT[2:0]	该位使用3-bit代码指示传输的当前状态。 如果一个成功传输完成中断或者传输非成功完成中断发生，中断信号可以通过写”000”到该位

		来清除当前状态。 000 - 初始值 001 - 传输错误引起中止（地址溢出） 010 - 传输停止请求引起中止（外设停止请求引起的停止或者 EB/DE 位引起的禁止传输） 011 - 传输错误引起中止（传输源地址访问错误） 100 - 传输错误引起中止（传输目的地地址访问错误） 101 - 成功传输完成 110 - 保留 111 - 传输暂停
15:1	Reserved	
0	MSK	该位用于在传输完成时，屏蔽来自 DMAC 的 EB 位 (CONFA[31]) 清除功能。 - 如果 MSK=0， DMAC 在传输完成时清除 CONFA:ENS 位。 - 如果 MSK=1， DMAC 在传输完成时不清除 CONFA:ENS 位。这个功能允许连续传输而不需要 CPU 干预。 这个功能用于硬件传输。使用这个功能时，可同时使能重载功能 CONFB:RC，CONFB:RS 和 CONFB:RD 位。

15.4.4 DMAC_SRCADR0/1/2/3

偏移地址：0x18（DMAC_SRCADR0）

0x28（DMAC_SRCADR1）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCADR[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCADR[15:0]															
RW															

位	标记	功能描述
31:0	SRCADR[31:0]	SRCADR[31:0] 位指定传输源地址。 不能设置相对于传输数据位宽（CONFB:WIDTH[1:0]）非对齐的地址。传输时可读取这些位的值。 如果 CONFB:FS=“1”，传输源地址设定为固定值，并且不会改变。 如果 CONFB:FS=“0”，且 CONFB:RS=“0”，传输时源地址根据 CONFB:WIDTH[1:0] 的数据位宽自增。如果传输成功完成，这个值为传输完成地址的下一地址。如果传输未成功完成，这个值为暂停时所设定的值。 如果 CONFB:FS=“0”且 CONFB:RS=“1”，传输时源地址根据 CONFB:WIDTH[1:0] 的数据位宽自增。传输完成后，传输开始时设定的源地址重新加载到本寄存器位中。

15.4.5 DMAC_DSTADR0/1/2/3

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTADR[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTADR[15:0]															
RW															

位	符号	描述
31:0	DSTADR[31:0]	DSTADR[31:0] 位指定传输目的地址。 不能设置相对于传输数据位宽 (CONFB:WIDTH[1:0]) 非对齐的地址。传输时可读取这些位的值。 如果CONFB:FD= “1” , 传输目的地址设置为固定地址, 并且不改变。 如果CONFB:FD= “0” , 且CONFB:RD= “0” , 传输时目的地址根据CONFB:WIDTH[1:0]的数据位宽自增。如果传输成功完成, 这个值为传输完成地址的下一地址。如果传输未成功完成, 这个值为暂停时所设定的值。 如果CONFB:FD= “0” 且CONFB:RD= “1” , 传输时目的地址根据CONFB:WIDTH[1:0]的数据位宽自增。传输完成后, 传输开始时设定的目的地址重新加载到本寄存器位中。

16. 端口控制器 (GPIO)

16.1 端口控制器简介

6615 系列有 85 个数字通用输入输出端口 PA[15:0], PB[15:0], PC[15:0], PD[15:0], PE[15:0], PF[11:9], PF[1:0]。100PIN 封装支持 85 个 GPIO, 未封出的 GPIO 需要配置成输入, 并使能上拉; 其他封装未封出的 GPIO 也应该做相应配置。

模拟模块 ADC/VC/LVD/LCD 的输入输出信号, 各功能模块 (如 SPI, UART, I2C, Timer 等) 的输入输出信号以及测试调试功能的输入输出信号都可以和数字通用输入输出端口复用。每个端口都可以配置成内部拉高(pull up)/拉低(pull down)的输入, 高阻输入 (floating input), 推挽输出(CMOS output), 开漏输出(open drain output), 增强驱动能力输出。芯片复位后端口为高阻输入, 目的是防止芯片被异常复位时, 对外部器件产生异常动作。但为了避免高阻输入而产生的漏电, 用户要在芯片启动之后对端口进行相应的配置 (配置成内部拉高/拉低输入或者输出)。数字端口被配置成模拟端口后, 数字功能被隔离, 不能输出数字“1”和“0”, CPU 读取端口输入值寄存器的结果

为“0”。

每个数字端口被配置为输入时，都可以提供外部中断，中断类型可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发 4 种，查询 PxSTAT[n]的中断标志位即可查出相应的中断触发端口。另外，每个数字端口的中断都可以把芯片从睡眠模式/深度睡眠模式唤醒到工作模式。

16.2 端口控制器主要特性

端口控制器支持以下特性：

- 端口输入值/输出值寄存器支持 AHB 总线读写
- 其他寄存器支持 AHB 总线接口读写
- 模拟功能管脚/SWD 调试管脚/数字通用管脚/数字功能管脚复用
- 支持上拉/下拉/增强驱动/开漏输出功能选择
- 支持工作模式/睡眠模式/深度睡眠模式下中断
- 支持高电平/低电平/上升沿/下降沿触发中断
- 支持位置位，位清零，位置位清零功能

16.3 端口控制器功能描述

16.3.1 端口配置功能

每一个端口都可以根据系统需求通过配置寄存器(PxADS)把端口配置成模拟端口或数字端口。当配置为数字端口时，还可以配置寄存器实现以下特性。

1. 内部上拉（PxPU）/下拉（PxPU）
2. 增强驱动输出(PxDR)
3. 开漏输出（PxOD）
4. 方向选择（PxDIR）
5. 输出高低电平选择(PxOUT)，可通过 AHB 总线访问
6. 输入电平状态（PxIN）可通过 AHB 总线访问
7. 清零（PxBCLR）
8. 置位清零（PxBSETCLR）

注：以上特性在配置为模拟端口时无效。

端口的状态和寄存器配置的关系如下表：

PxADS	PxDIR	PxOUT	PxIN	PxBCLR	PxBSETCLR	PxPU	PxPD	PxOD	PxDR	PxAFRx	IO 方向	IO 状态
1	W	W	0	W	W	W	W	W	W	W	输入/输出	模拟
0	1	W	X	W	W	0	0	W	W	0	输入	浮空
0	1	W	0	W	W	0	1	W	W	0	输入	下拉
0	1	W	1	W	W	1	0	W	W	0	输入	上拉
0	1	W	1	W	W	1	1	W	W	0	输入	上拉
0	1	W	1	W	W	W	W	W	W	0	输入	1
0	1	W	0	W	W	W	W	W	W	0	输入	0
0	0	1	1	0	0	W	W	0	W	0	输出	1
0	0	0	0	0	0	W	W	0	W	0	输出	0
0	0	W	1	0	0	W	W	0	W	0	输出	1
0	0	W	0	1	0	W	W	0	W	0	输出	0
0	0	W	(SET)1/(CLR)0	0	1	W	W	0	W	0	输出	(SET)1/(CLR)0
0	0	0	0	0	0	W	W	1	W	0	输出	0
0	0	1	X	0	0	0	0	1	W	0	输出	Z
0	0	1	0	0	0	0	1	1	W	0	输出	0
0	0	1	1	0	0	1	0	1	W	0	输出	1
0	0	1	1	0	0	1	1	1	W	0	输出	1

注：

0 - Logic low

1 - Logic high

W - whatever

0 or 1

X - unknow state

Z - high impedance

表 16-1 端口状态真值表

16.3.2 端口数字复用功能

端口数字复用是端口控制器的主要功能之一。通过配置寄存器，可以灵活的把端口配置成测试调试端口/数字通用端口/数字功能端口。

如下表所示：AFRH/AFRL 寄存器用于数字通用端口/数字功能端口切换，每一个端口都可以独立配置成系统所需的功能端口。调试端口的配置信息请参考相关章节。

PA00	BTIM6_TB	SPI3_NSS	ATIM_CH1	VC1_OUT	UART2_CTS	ATIM_ETR	UART4_TX
PA01	BTIM6_TB	SPI3_SCK	ATIM_CH2	GTIM2_ETR	UART2_RTS	BTIM456_ETR	UART4_RX
PA02	BTIM4_TA	SPI3_MISO	ATIM_CH3	VC2_OUT	UART2_TX	GTIM1_GT	LPUART1_TX
PA03	BTIM4_TB	SPI3_MOSI	ATIM_GT	LPTIM1_GT	UART2_RX	GTIM3_ETR	LPUART1_RX
PA04	SPI1_NSS	BTIM789_ETR	LPTIM1_ETR	I2C1_SCL	UART2_CK	GTIM1_ETR	UART5_TX
PA05	SPI1_SCK	BTIM9_TA	ATIM_ETR	I2C1_SDA	ATIM_CH1	GTIM2_GT	UART5_RX
PA06	SPI1_MISO	STIM_BKIN	BTIM1_TA	BTIM9_TB	LPUART1_CTS	GTIM1_CH1	VC1_OUT
PA07	SPI1_MOSI	STIM_CH1N	BTIM1_TB	VC2_OUT	ATIM_BK	GTIM1_CH2	UART5_RTS
PA08	MCO	STIM_CH1	CRS_SYNC	LPUART2_TX	UART1_CK	UART3_RTS	I2C3_SCL
PA09	MCO	STIM_CH2	LVD_OUT	LPUART2_RX	UART1_TX	GTIM1_BKIN	I2C1_SCL
PA10	e78161_data	STIM_CH3	CAN1_STBY	LPUART2_CTS	UART1_RX	GTIM3_BKIN	I2C1_SDA
PA11	SPI1_MISO	STIM_CH4	CAN1_RX	LPUART2_RTS	UART1_CTS	STIM_BKIN2	GTIM3_CH1
PA12	SPI1_MOSI	STIM_ETR	CAN1_TX	SPI4_NSS	UART1_RTS	GTIM2_CH1	GTIM3_CH2
PA13	e78161_clk	UART3_TX	I2C4_SCL	SPI4_MOSI	GTIM3_ETR	GTIM2_CH2	LPUART1_RX
PA14	e78161_rst	UART3_RX	I2C4_SDA	SPI4_SCK	UART2_TX	PCA_ETR	LPUART1_TX
PA15	SPI1_NSS	UART3_CK	ATIM_ETR	SPI4_MISO	UART2_RX	ATIM_CH1	UART4_RTS
PB00	ATIM_CH1N	STIM_CH2N	BTIM2_TA	PCNT_S1	GTIM1_GT	UART6_TX	SPI1_SCK
PB01	ATIM_CH2N	STIM_CH3N	BTIM2_TB	CAN2_RX	LPUART1_RTS	UART6_RX	SPI1_MISO
PB02	ATIM_CH3N	STIM_CH2	LPTIM1_TA	CAN2_TX	PCNT_S2	LVD_OUT	SPI1_MOSI
PB03	SPI1_SCK	I2S_MCK	ATIM_CH2	CAN2_STBY	GTIM3_ETR	UART1_RTS	UART5_TX
PB04	SPI1_MISO	I2C3_SDA	BTIM1_TA	GTIM3_BKIN	GTIM1_CH1	UART1_CTS	UART5_RX
PB05	SPI1_MOSI	STIM_CH3	GTIM2_BKIN	pcnt_S1o	GTIM1_CH2	UART1_CK	UART5_CK
PB06	UART1_TX	I2C1_SCL	LPTIM1_ETR	GTIM2_CH2	pcnt_S2o	e78161_clk	
PB07	UART1_RX	I2C1_SDA	LPTIM1_GT	GTIM3_CH2	I2S_WS	e78161_rst	UART4_CTS
PB08	CAN1_RX	I2S_SD	SPI4_NSS	GTIM2_CH1	I2C1_SCL	GTIM1_BKIN	e78161_data
PB09	CAN1_TX	I2S_CK	SPI4_MOSI	GTIM2_GT	I2C1_SDA	SPI2_NSS	
PB10	CAN1_STBY	LPUART1_RX	ATIM_CH3	GTIM3_BKIN	LPUART1_TX	SPI2_SCK	I2C2_SCL
PB11	LPUART1_TX	PCNT_S1	ATIM_GT	GTIM3_ETR	LPUART1_RX	e78162_clk	I2C2_SDA
PB12	SPI2_NSS	STIM_BKIN	LPUART1_RTS	PCNT_S2		e78162_rst	GTIM1_GT
PB13	SPI2_SCK	STIM_CH1N	GTIM1_GT	e78162_data	LPUART1_CTS	I2C2_SCL	BTIM5_TA

PB14	SPI2_MISO	STIM_CH2N	RTC_OUT		LPUART1_RTS	I2C2_SDA	BTIM5_TB
PB15	SPI2_MOSI	STIM_CH3N		GTIM1_GT		ATIM_CH1N	STIM_BKIN
PC00	LPTIM1_TB	BTIM7_TA	BTIM3_TA	GTIM2_GT	LPUART1_RX	I2C3_SCL	ATIM_CH1
PC01	LPTIM1_TA	BTIM7_TB	BTIM3_TB	GTIM3_ETR	LPUART1_TX	I2C3_SDA	ATIM_CH2
PC02	LPTIM1_GT	BTIM8_TA	SPI2_MISO		UART4_RTS	I2C1_SCL	ATIM_CH3
PC03	LPTIM1_ETR	BTIM8_TB	SPI2_MOSI		UART4_CTS	I2C1_SDA	ATIM_GT
PC04	LPTIM2_TB	BTIM3_TA	LPUART1_TX	I2C3_SCL	GTIM1_CH1	UART5_CK	UART5_CTS
PC05	LPTIM2_TA	BTIM3_TB	LPUART1_RX	I2C3_SDA	GTIM1_CH2	UART6_CK	SPI1_NSS
PC06	GTIM1_CH1	SPI3_MISO	BTIM1_TA	SPI4_NSS	GTIM2_CH1	UART3_TX	UART4_RTS
PC07	GTIM1_CH2	SPI3_MOSI	BTIM1_TB	SPI4_MOSI	GTIM2_CH2	UART3_RX	UART4_CTS
PC08	GTIM1_ETR	STIM_CH1	BTIM2_TA	SPI4_SCK	GTIM2_GT	UART3_CK	BTIM789_ET R
PC09	BTIM456_ET R	STIM_CH2	BTIM2_TB	SPI4_MISO	GTIM2_BKIN	UART3_CTS	I2C3_SDA
PC10	LPUART1_TX	STIM_CH3	UART3_CTS	BTIM6_TA	PCA_CH1	I2C2_SCL	UART4_TX
PC11	LPUART1_RX	STIM_CH4	UART3_RTS	BTIM6_TB	PCA_CH2	I2C2_SDA	UART4_RX
PC12	e78162_clk		UART5_TX	BTIM456_ETR	PCA_CH3	LPUART2_TX	UART4_CK
PC13	e78162_rst	SPI3_SCK	GTIM1_ETR			PCA_ETR	
PC14							
PC15							
PD00	BTIM5_TA	SPI2_NSS		CAN1_RX	PCA_CH4	LPUART2_RX	MCO
PD01	BTIM5_TB	SPI2_SCK		CAN1_TX	PCA_CH5	LPUART2_CTS	MCO
PD02	LPUART1_RTS	BTIM4_TB	BTIM123_ETR	STIM_CH1N	I2C4_SCL	LPUART2_RTS	UART5_RX
PD03	UART2_CTS		SPI2_MISO	STIM_CH2N	I2C4_SDA	SPI2_NSS	e78162_dat a
PD04	UART2_RTS	SPI2_MOSI		STIM_CH3N	BTIM9_TA	SPI2_SCK	BTIM123_ET R
PD05	UART2_TX	GTIM3_CH1	I2S_MCK	STIM_BKIN	BTIM9_TB	SPI2_MISO	I2S_WS
PD06	UART2_RX	GTIM3_GT	I2C2_SCL	BTIM456_ETR	GTIM2_CH1	SPI2_MOSI	
PD07	UART2_CK	BTIM5_TB	I2C2_SDA	I2S_SD	GTIM2_CH2		
PD08	LPUART1_TX	PCA_ETR	I2C4_SCL	I2S_CK	UART6_TX	ATIM_CH2N	STIM_CH1
PD09	LPUART1_RX	PCA_CH1	I2C4_SDA	STIM_BKIN2	UART6_RX	ATIM_CH3N	STIM_CH2
PD10		PCA_CH2	CAN2_RX	LPTIM2_TB	UART6_CK	ATIM_CH1	STIM_CH3
PD11	LPUART1_CTS	PCA_CH3	CAN2_TX	LPTIM2_TA	UART6_CTS	ATIM_CH2	STIM_CH4
PD12	LPUART1_RTS	PCA_CH4	CAN2_STBY	LPTIM2_GT	UART6_RTS	ATIM_CH3	STIM_ETR
PD13	LPTIM2_TB	PCA_CH5	BTIM789_ETR	LPTIM2_ETR	ATIM_GT	UART4_TX	STIM_CH1N
PD14	LPTIM2_TA	SPI3_NSS	BTIM6_TA	MCO	ATIM_ETR	UART4_RX	STIM_CH2N
PD15	CRS_SYNC	SPI3_SCK	BTIM6_TB	MCO	ATIM_BK	UART4_CK	STIM_CH3N

PE00			SPI4_SCK				
PE01			SPI4_MISO				
PE02	GTIM2_ETR	SPI3_MISO	BTIM123_ETR	UART6_CTS	LPUART2_TX	PCA_CH1	STIM_CH1
PE03	GTIM1_CH1	SPI3_MOSI	BTIM1_TA	UART6_RTS	LPUART2_RX	PCA_CH2	STIM_CH2
PE04	GTIM1_CH2	STIM_BKIN	BTIM1_TB	UART6_TX	LPUART2_CTS	PCA_CH3	STIM_CH3
PE05	BTIM5_TA	STIM_BKIN2	BTIM2_TA	UART6_RX	LPUART2_RTS	PCA_CH4	STIM_CH4
PE06	BTIM5_TB	GTIM1_BKIN	BTIM2_TB	UART6_CK	GTIM2_ETR	PCA_CH5	STIM_ETR
PE07			LPTIM1_TB			UART6_CTS	UART5_RTS
PE08	ATIM_BK	SPI4_NSS	BTIM123_ETR	LPUART2_TX	PCA_ETR	UART6_RTS	UART4_TX
PE09	ATIM_CH1	SPI4_SCK	ATIM_ETR	LPUART2_RX	PCA_CH1	MCO	UART4_RX
PE10	ATIM_CH2	SPI4_MISO	I2C4_SCL		PCA_CH2	MCO	UART5_TX
PE11	ATIM_CH3	SPI4_MOSI	I2C4_SDA	LPUART2_CTS	PCA_CH3	UART3_TX	UART5_RX
PE12	ATIM_GT	LPTIM2_TB	SPI1_NSS	LPUART2_RTS	PCA_CH4	UART3_RX	GTIM2_CH1
PE13	ATIM_CH1N	LPTIM2_TA	SPI1_SCK	GTIM3_CH1	PCA_CH5	UART3_CK	GTIM2_CH2
PE14	ATIM_CH2N	LPTIM2_GT	SPI1_MISO	GTIM3_CH2	BTIM6_TA	UART3_CTS	GTIM2_GT
PE15	ATIM_CH3N	LPTIM2_ETR	SPI1_MOSI	GTIM3_GT	BTIM6_TB	UART3_RTS	GTIM2_BKIN
PF00	CRS_SYNC						
PF01							
PF09	I2C3_SCL	SPI3_NSS	GTIM1_GT	LPTIM1_TB			
PF10	I2C3_SDA			LPTIM1_TA			
PF11							

表 16-2 端口数字复用表

16.3.3 端口中断功能

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，分别对应的中断使能寄存器为高电平中断使能寄存器(PxHIE)/低电平中断使能寄存器(PxLIE)/上升沿中断使能寄存器(PxRIE)/下降沿中断使能寄存器(PxFIE)。当中断触发时，通过查询中断状态寄存器(PxSTAT)就可以判断是哪一个端口触发了中断，通过清零中断清除寄存器(PxICLR)就可以清除对应的中断状态标志位。

16.3.4 端口配置操作流程

端口复用配置为模拟端口操作流程

- 设置寄存器 PxADS[n]为 1

端口复用配置为数字通用端口操作流程

- 设置寄存器 PxADS[n] 为 0
- 设置寄存器 PxAFRx[n] 为 0
- 设置寄存器 PxDIR[n] 为 1: 端口方向为输入, CPU 可以读取端口的状态 PxIN[n]。
- 设置寄存器 PxDIR[n] 为 0: 端口方向为输出
- 设置寄存器 PxOUT[n] 为 1: 端口输出高电平
- 设置寄存器 PxOUT[n] 为 0: 端口输出低电平

端口复用配置为数字功能端口操作流程

- 设置寄存器 PxADS[n] 为 0
- 设置寄存器 PxAFRx[n] 为 1~7 (根据系统需求, 参考端口复用表)
- 设置寄存器 PxDIR[n] (根据系统需求)
- 设置寄存器 PxPU[n]/PxPD[n]/PxOD[n] (根据系统需求)

端口复用配置为调试测试端口操作流程

参考测试调试相关章节。

端口高电平中断操作流程

- 设置寄存器 PxADS[n] 为 0
- 设置寄存器 PxAFRx[n] 为 0
- 设置寄存器 PxDIR[n] 为 1
- 设置寄存器 PxHIE[n] 为 1
- 中断触发后读取中断状态寄存器 PxSTAT[n]
- 设置寄存器 PxICLR[n] 为 0 清除中断状态寄存器 PxSTAT[n]

端口低电平中断操作流程

- 设置寄存器 PxADS[n] 为 0
- 设置寄存器 PxAFRx[n] 为 0
- 设置寄存器 PxDIR[n] 为 1
- 设置寄存器 PxLIE[n] 为 1

- e) 中断触发后读取中断状态寄存器 PxSTAT[n]
- f) 设置寄存器 PxICLR[n]为 0 清除中断状态寄存器 PxSTAT[n]

端口上升沿中断操作流程

- a) 设置寄存器 PxADS[n]为 0
- b) 设置寄存器 PxAFRx[n]为 0
- c) 设置寄存器 PxDIR[n]为 1
- d) 设置寄存器 PxRIE[n]为 1
- e) 中断触发后读取中断状态寄存器 PxSTAT[n]
- f) 设置寄存器 PxICLR[n]为 0 清除中断状态寄存器 PxSTAT[n]

端口下降沿中断操作流程

- a) 设置寄存器 PxADS[n]为 0
- b) 设置寄存器 PxAFRx[n]为 0
- c) 设置寄存器 PxDIR[n]为 1
- d) 设置寄存器 PxFIE[n]为 1
- e) 中断触发后读取中断状态寄存器 PxSTAT[n]
- f) 设置寄存器 PxICLR[n]为 0 清除中断状态寄存器 PxSTAT[n]

端口上拉使能配置操作流程

- a) 设置寄存器 PxPU[n]为 1

端口下拉使能配置操作流程

- a) 设置寄存器 PxPU[n]为 0
- b) 设置寄存器 PxPD[n]为 1

注：当 PxPU[n],PxPD[n]同时置 1 时，PxPU[n]优先级高，PxPD[n]无效。

端口增强驱动配置操作流程

- a) 设置寄存器 PxDR[n]为 0

端口开漏输出配置操作流程

- a) 设置寄存器 PxOD[n]为 1

端口位置位操作流程

- a) 设置寄存器 PxBSET[n]为 1

端口位清零操作流程

- a) 设置寄存器 PxBCLR[n]为 1

端口位置位清零操作流程

- a) 设置寄存器 PxBSETCLR[n]为 1

16.4 端口控制器寄存器描述

基地址

PA 0x50000000; PB 0x50000400; PC 0x50000800

PD 0x50000C00; PE 0x50001000; PF 0x50001400

寄存器	偏移地址	描述
PxDIR	0x000	端口方向控制寄存器
PxOD	0x004	端口开漏输出控制寄存器
PxSPEED	0x008	端口速度控制寄存器
PxPD	0x00C	端口下拉电阻控制寄存器
PxPU	0x010	端口上拉电阻控制寄存器
PxAFRH	0x014	功能复用控制寄存器
PxAFRL	0x018	功能复用控制寄存器
PxADS	0x01C	数字模拟功能选择控制寄存器
PxDR	0x020	端口驱动控制寄存器
PxRIE	0x024	端口上升沿中断控制寄存器
PxFIE	0x028	端口下降沿中断控制寄存器
PxHIE	0x02C	端口高电平中断控制寄存器
PxLIE	0x030	端口低电平中断控制寄存器
PxSTAT	0x034	端口中断标志寄存器
PxICLR	0x038	端口中断清除控制寄存器
PxLOCK	0x040	端口锁定控制寄存器
PxIN	0x050	端口输入值寄存器
PxOUT	0x054	端口输出控制寄存器
PxBCLR	0x058	端口位清除控制寄存器
PxBSETCLR	0x05C	端口位清除与置位控制寄存器
PxINV	0x060	端口输出反相控制寄存器

表 16-3 GPIO 寄存器列表

16.4.1 端口 Px 输入输出配置寄存器(PxDIR) (x = A,B,C,D,E,F)

地址偏移量: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxDIR[15:0]															

RW		

位	标记	功能描述
31:16	Reserved	保留
15:0	PxDIR	端口Px输入输出配置寄存器（对应Px15-Px00） 1：配置成输入 0：配置成输出 注：每个bit对应一个端口，例如：PxDIR[15]对应端口Px15

16.4.2 端口 Px 开漏输出配置寄存器(PxOD) (x = A,B,C,D,E,F)

地址偏移量：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxOD[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxOD	端口Px开漏输出配置寄存器 1：使能 0：禁止

16.4.3 端口 Px 速度配置寄存器(PxSPEED) (x = A,B,C,D,E,F)

地址偏移量：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxSPEED[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留

15:0	PxSPEED	端口Px速度控制 1: 使能快速 0: 禁止快速

16.4.4 端口 Px 下拉使能配置寄存器(PxPD) (x = A,B,C,D,E,F)

地址偏移量: 0x00C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxPD[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxPD	端口Px下拉使能配置寄存器 1: 使能 0: 禁止

16.4.5 端口 Px 上拉使能配置寄存器(PxPU) (x = A,B,C,D,E,F)

地址偏移量: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxPU[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxPU	端口Px上拉使能配置寄存器 1: 使能 0: 禁止

16.4.6 端口 Px 复用功能配置寄存器(PxAFRH) (x = A,B,C,D,E,F)

地址偏移量: 0x014

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV	Px15_SEL			RSV	Px14_SEL			RSV	Px13_SEL			RSV	Px12_SEL		
	RW				RW				RW				RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	Px11_SEL			RSV	Px10_SEL			RSV	Px09_SEL			RSV	Px08_SEL		
	RW				RW				RW				RW		

位	标记	功能描述
31	Reserved	保留
30:28	Px15_SEL	Px15端口功能复用选择控制寄存器, 参考端口数字复用功能表格
...	...	...
2:0	Px08_SEL	Px08端口功能复用选择控制寄存器, 参考端口数字复用功能表格

16.4.7 端口 Px 复用功能配置寄存器(PxAFRL) (x = A,B,C,D,E,F)

地址偏移量: 0x018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV	Px07_SEL			RSV	Px06_SEL			RSV	Px05_SEL			RSV	Px04_SEL		
	RW				RW				RW				RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	Px03_SEL			RSV	Px02_SEL			RSV	Px01_SEL			RSV	Px00_SEL		
	RW				RW				RW				RW		

位	标记	功能描述
31	Reserved	保留
30:28	Px07_SEL	Px07端口功能复用选择控制寄存器, 参考端口数字复用功能表格
...	...	...
2:0	Px00_SEL	Px00端口功能复用选择控制寄存器, 参考端口数字复用功能表格

16.4.8 端口 Px 数模配置寄存器(PxADS) (x = A,B,C,D,E,F)

地址偏移量: 0x01C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxADS[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxADS	端口Px数模配置寄存器 (对应Px15-Px00) 1: 配置为模拟端口 0: 配置为数字端口

16.4.9 端口 Px 驱动能力配置寄存器(PxDR) (x = A,B,C,D,E,F)

地址偏移量: 0x020

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxDR[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxDR	端口Px驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力

16.4.10 端口 Px 上升沿中断使能配置寄存器(PxRIE) (x = A,B,C,D,E,F)

地址偏移量: 0x024

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxRIE[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxRIE	端口Px上升沿中断使能配置寄存器 1: 使能 0: 禁止

16.4.11 端口 Px 下降沿中断使能配置寄存器(PxFIE) (x = A,B,C,D,E,F)

地址偏移量: 0x028

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxFIE[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxFIE	端口Px下降沿中断使能配置寄存器 1: 使能 0: 禁止

16.4.12 端口 Px 高电平中断使能配置寄存器(PxHIE) (x = A,B,C,D,E,F)

地址偏移量: 0x02C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxHIE[15:0]															

RW		

位	标记	功能描述
31:16	Reserved	保留
15:0	PxHIE	端口Px高电平中断使能配置寄存器 1: 使能 0: 禁止

16.4.13 端口 Px 低电平中断使能配置寄存器(PxLIE) (x = A,B,C,D,E,F)

地址偏移量: 0x030

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxLIE[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxLIE	端口Px低电平中断使能配置寄存器 1: 使能 0: 禁止

16.4.14 端口 Px 中断状态寄存器(PxSTAT) (x = A,B,C,D,E,F)

地址偏移量: 0x034

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxSTAT[15:0]															
RO															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxSTAT	端口Px中断状态寄存器

		1: 中断触发 0: 无中断触发

16.4.15 端口 Px 中断清除寄存器(PxICLR) (x = A,B,C,D,E,F)

地址偏移量: 0x038

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxICLR[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxICLR	端口Px中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位

16.4.16 端口 Px 锁定控制寄存器(PxLOCK) (x = A,B,C,D,E,F)

地址偏移量: 0x040

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxLOCK[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxLOCK	端口Px控制寄存器锁定控制 1: 锁定, 控制寄存器不可修改 0: 解锁状态, 控制寄存器可修改 锁定的控制寄存器有DIR/OD/SPEED/PD/PU/AFRH/ARFL/ADS/DR 写LOCK寄存器高16位为0X5A5A才可以写入

16.4.17 端口 Px 输入值寄存器(PxIN) (x = A,B,C,D,E,F)

地址偏移量: 0x050

复位值: 0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxIN[15:0]															
RO															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxIN	端口Px输入值寄存器 (对应Px15-Px00) 1: 输入为高电平 0: 输入为低电平

16.4.18 端口 Px 输出值配寄存器(PxOUT) (x = A,B,C,D,E,F)

地址偏移量: 0x054

复位值: 0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxOUT[15:0]															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxOUT	端口Px输出值配置寄存器 (对应Px15-Px00) 1: 输出高电平。如果配置成开漏输出, 则需外部上拉电阻来拉高。 0: 输出低电平。

16.4.19 端口 Px 位清零寄存器(PxBCLR) (x = A,B,C,D,E,F)

地址偏移量: 0x058

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBCLR[15:0]															
W1															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxBCLR	端口Px位清零寄存器（对应Px15-Px00） 1：清零 0：保持

16.4.20 端口 Px 位置位清零寄存器(PxBSETCLR) (x = A,B,C,D,E,F)

地址偏移量：0x05c

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PxBCLR[15:0]															
W1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxBSET[15:0]															
W1															

位	标记	功能描述
31:16	PxBCLR	端口Px位清零寄存器 1：清零 0：保持
15:0	PxBSET	端口Px位置位寄存器 1：置位 0：保持

注意：

- PxBSET 和 PxBCLR 相同位同时置 1 时，PxBCLR 具有高优先级。即该端口被清零。

16.4.21 端口 Px 位反相寄存器(PxINVx = A,B,C,D,E,F)

地址偏移量：0x060

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxINV															
W															

位	标记	功能描述
31:16	Reserved	保留
15:0	PxINV	端口Px位输出反向寄存器（对应Px15-Px00） 1: 反相 0: 保持

17. I2C 总线（I2C）

17.1 I2C 简介

I2C 是双线双向的串行总线，它为设备之间数据交换提供了一种简单高效的方法。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时请求控制总线时发生数据冲突。

I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。I2C 总线使用连接设备的"SCL"(串行时钟总线)和"SDA"(串行数据总线)来传送信息。数据在主机与从机之间通过 SCL 时钟线控制在 SDA 数据线上实现一个字节一个字节的同步传输，每个字节为 8 位长度，一个 SCL 时钟脉冲传输一个数据位，数据由最高位 MSB 开始传输，每个传输字节后跟随一个应答位，每个位在 SCL 为高时采样；因此，SDA 线只有在 SCL 为低时才可以改变，在 SCL 为高时 SDA 保持稳定。当 SCL 为高时，SDA 线上的跳变视为命令中断（START 或 STOP）。

I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器（I2Cx_STAT）能反映 I2C 总线控制器和 I2C 总线的状态。

17.2 I2C 主要特性

I2C 控制器支持以下特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps) / 快速(400Kbps) / 高速(1Mbps) 三种工作速率

- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

17.3 I2C 协议描述

通常标准 I2C 传输协议包含四个部分：

- 1) 起始信号或重复起始信号
- 2) 从机地址传输和读写位传输
- 3) 数据传输
- 4) 停止信号

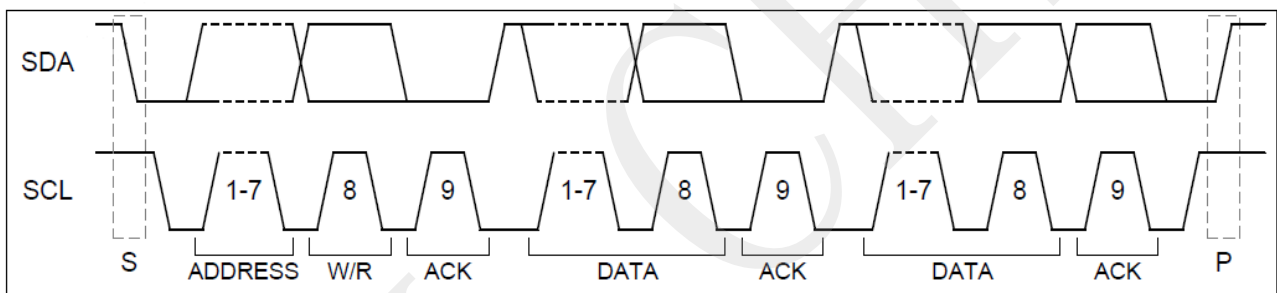


图 17-1 I2C 传输协议

17.3.1 I2C 总线上数据传输

主机发出从机接收 7 位地址（一个字节）传输方向未改变。

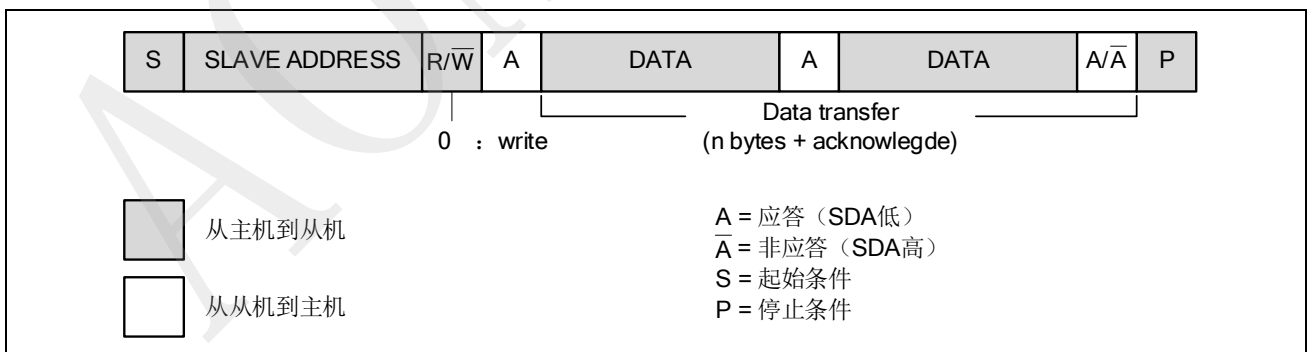


图 17-2 主机向从机传输数据

第一个字节后主机紧接着由从机读取数据(内容为从机地址) 传输方向改变。

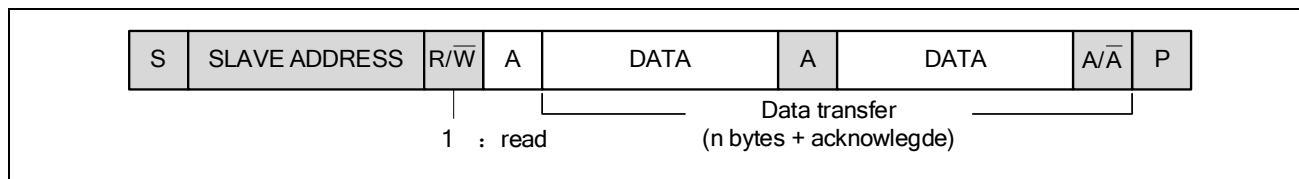


图 17-3 主机由从机读取地址

17.3.2 起始位或重复起始信号

当总线处于空闲状态下，说明没有主机对总线发起传输请求（SCL 和 SDA 线同时为高），主机可以通过发送一个 START 信号来发起传输请求。

起始信号：通常表示为 S-bit。当 SCL 线为高时，SDA 线上信号由高至低，标示总线上产生起始信号，新的传输开始。

重复起始信号 (Sr)：即在两个 START 信号之间没有 STOP 信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信（例如：从写入设备到从设备读出）而不释放总线。

STOP 信号：主机向总线发出停止信号结束数据传送。停止信号，通常用 P-bit 表示，当 SCL 线为高时，SDA 线上出现由低到高的信号，被定义为停止信号。

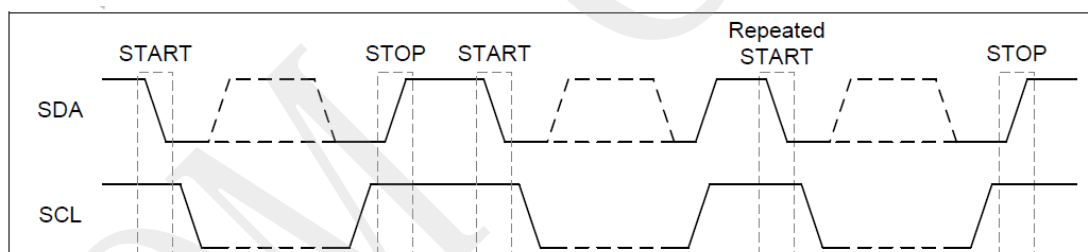


图 17-4 START 和 STOP 条件

17.3.3 从机地址传输

START 信号是从机地址时，主机立即传输数据的第一位。这是一个跟随有一个读写位的 7 位调用地址，读写位控制从机的信号传输方向。系统中没有两个从机有相同的地址，只有被主机寻址的从机会通过在第 9 个 SCL 时钟周期将 SDA 置为低电平作为应答。

17.3.4 数据传输

当从机地址被成功识别，就可以根据读写所决定的方向，开始一字节一字节的数据传

输，每个传输字节最后带一个第 9 时钟周期上的响应信号，如果从机上产生无响应信号(NACK)，主机可以产生停止信号来退出数据传输，或者产生重复起始信号开始新一轮的数据传输。

当主机作为接收器件时，发生无响应信号(NACK)，从机释放 SDA 线，使主机产生停止信号或重复起始信号。

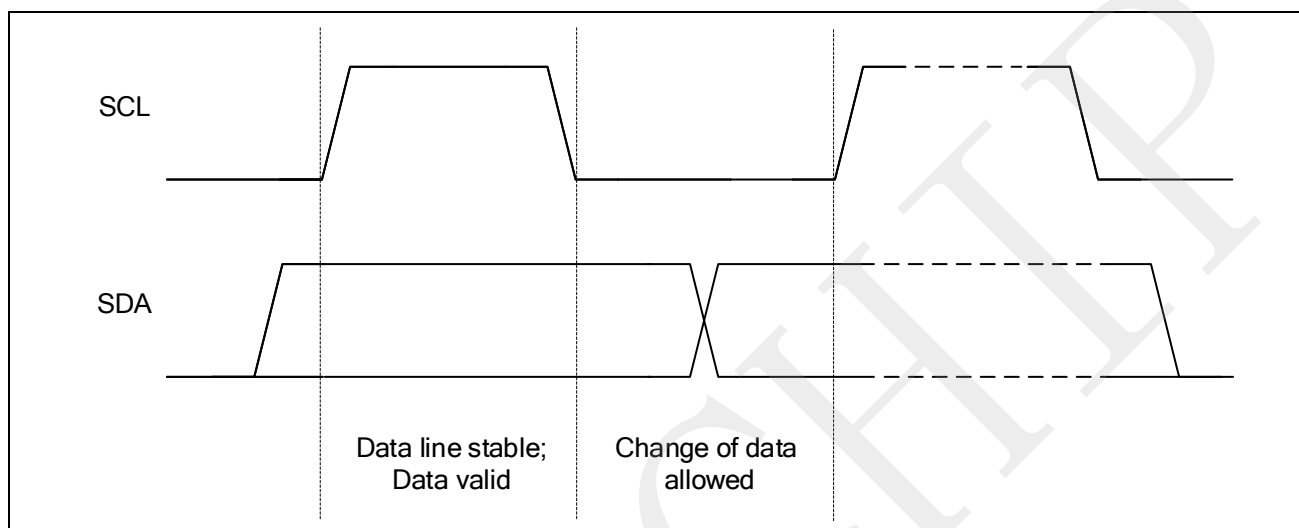


图 17-5I2C 总线上位传输

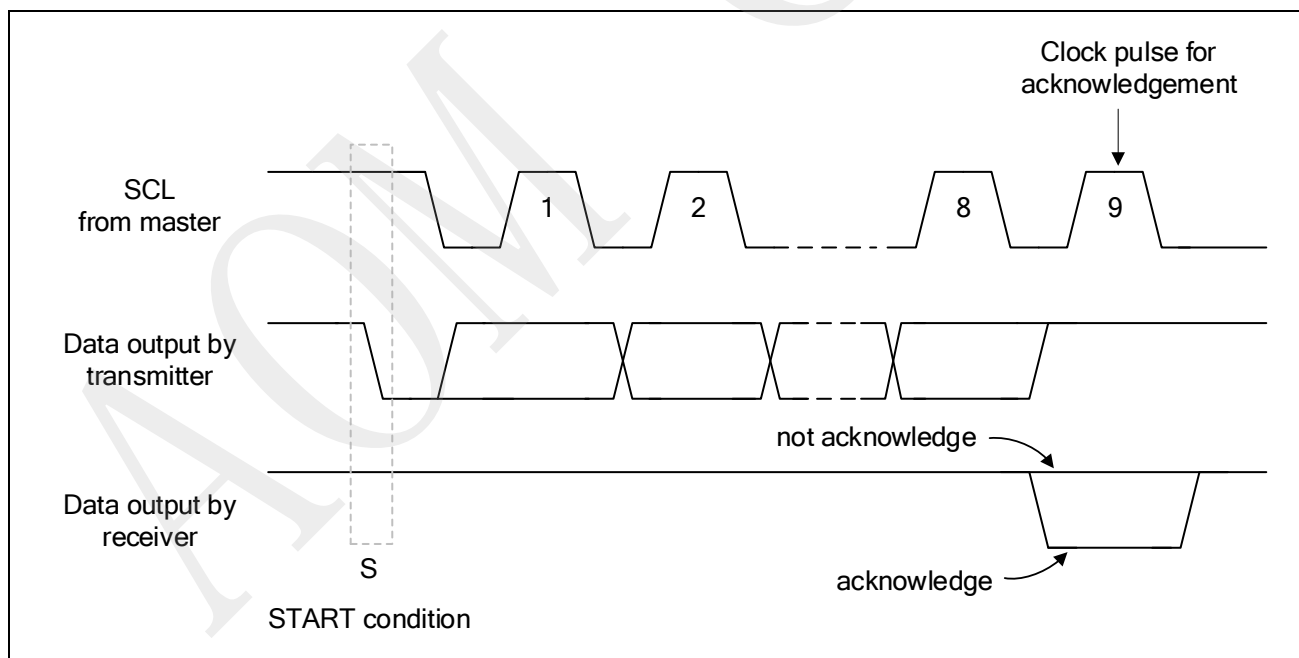


图 17-6I2C 总线上应答信号

17.4.1 I2C 工作模式

I2C 组件可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100Kbits/s 而在高速模式下可达 400Kbits/s，在高速模式下可达 1Mbits/s，并且可以在以下四种模式下工作：

- 1) 主机发送模式：当“SCL”输出串行时钟信号时“SDA”输出串行数据。
- 2) 主机接收模式：当“SCL”输出串行时钟信号时串行数据通过“SDA”接收。
- 3) 从机接收模式：串行数据和串行时钟分别通过“SDA”和“SCL”接收。
- 4) 从机发送模式：当串行时钟从“SCL”口输入时串行数据通过“SDA”口发送。

17.4.2 仲裁与同步逻辑

在主发送模式中，仲裁逻辑检查每个发送的逻辑 1 是否真正出现在总线上。如果总线的另一个器件撤消了一个逻辑 1 并将 SDA 线拉低，仲裁丢失，I2C 模块立刻由主发送器变为从接收器。I2C 模块将继续输出时钟脉冲（在 SCL 上），直至发送完当前的串行字节。

仲裁也可能在主接收模式中丢失。这种情况只在 I2C 模块正在向总线返回一个“非应答：（逻辑 1）”时出现。当总线的另一个器件将信号拉低时仲裁丢失。由于它只在串行字节结束时出现，因此 I2C 模块不会再产生时钟脉冲。

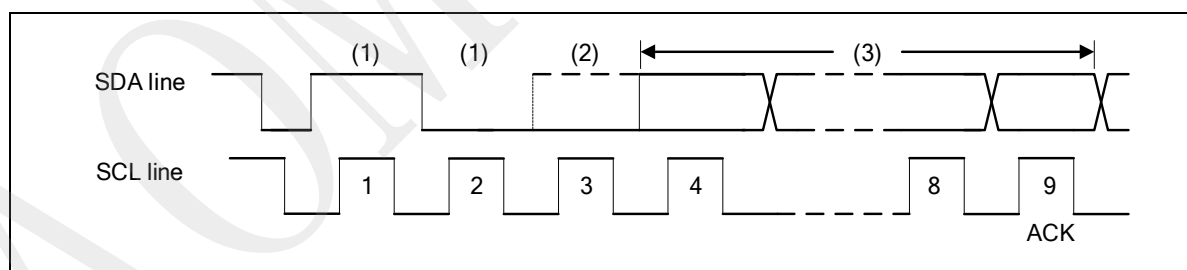


图 17-8 I2C 总线上的仲裁

仲裁丢失及处理过程：

- 1) 另一器件发送串行数据；
- 2) 另一器件通过拉低 SDA 先撤消了该 I2C 主机发送的一个逻辑 1(虚线)。仲裁丢失，I2C 进入从接收模式；
- 3) 此时 I2C 处于从接收模式，但仍产生时钟脉冲，直至发送完当前字节。I2C 将不为

下个字节的传输产生时钟脉冲。一旦赢得仲裁，SDA 上的数据传输由新的主机来启动。

同步逻辑使得串行时钟发生器与另一个器件 SCL 线上的时钟脉冲同步。如果 2 个或更多主器件产生时钟脉冲，则高电平周期取决于产生最短高电平时间的器件；低电平周期取决于产生最长低电平时间的器件。

17.4.3 串行时钟发生器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 pclk 信号的频率关系为 $F_{scl} = F_{pclk}/8*(N+1)$ ($N > 0$)

下面的表格表示 pclk 为各种频率时，分频系数为 1-7 时，SCL 信号的频率值。

分频系数 频率 (kHz)	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

表 17-1 I2C 时钟信号波特率

17.4.4 输入滤波器

输入信号与时钟信号 (pclk) 同步，低于 2 个时钟周期的尖峰脉冲信号会被滤除。每个滤波器由 3 个触发器组成。第一个触发器用来直接锁存输入信号，并将数据载入由另外两个构成的移位寄存器中。当第二和第三个触发器的状态是"11"或"00"时，内部的滤除信号会各自被置 1 或置 0。

17.4.5 地址比较器

I2C 比较器将自己的从机地址与接收到的 7 位从机地址做比较。它可使用 "I2Cx_ADDR" 寄存器对自己的从机地址进行编程。并且会根据 "I2Cx_ADDR" 寄存器的 "GC" 位与首次接收到的 8 位字节与通用调用地址 (0x00) 相比较。如果任何一者相同, "I2Cx_CR" 寄存器的 "si" 位会被置 1 并产生一个中断请求。

17.4.6 中断产生器

I2C 模块的所有四种模式都被使用时, 则有 26 种可能的总线状态。当 I2C 进入 26 种状态的 25 种状态时, "I2Cx_CR" 寄存器的 "si" 标志位会被硬件置 1。"si" 位唯一不会被置 1 的状态是 F8h, 这表明没有有效的相关状态信息。"si" 标志位必须通过软件清零。为了清除 "si" 位, 必须把 0 写入此位。若在 "si" 里写 1 不会改变 "si" 的值。为了确定中断的实际中断源, 中断服务程序在清除 "si" 标志位之前, 会对 I2C 状态寄存器进行查询。

17.4.7 I2C 主机发送模式

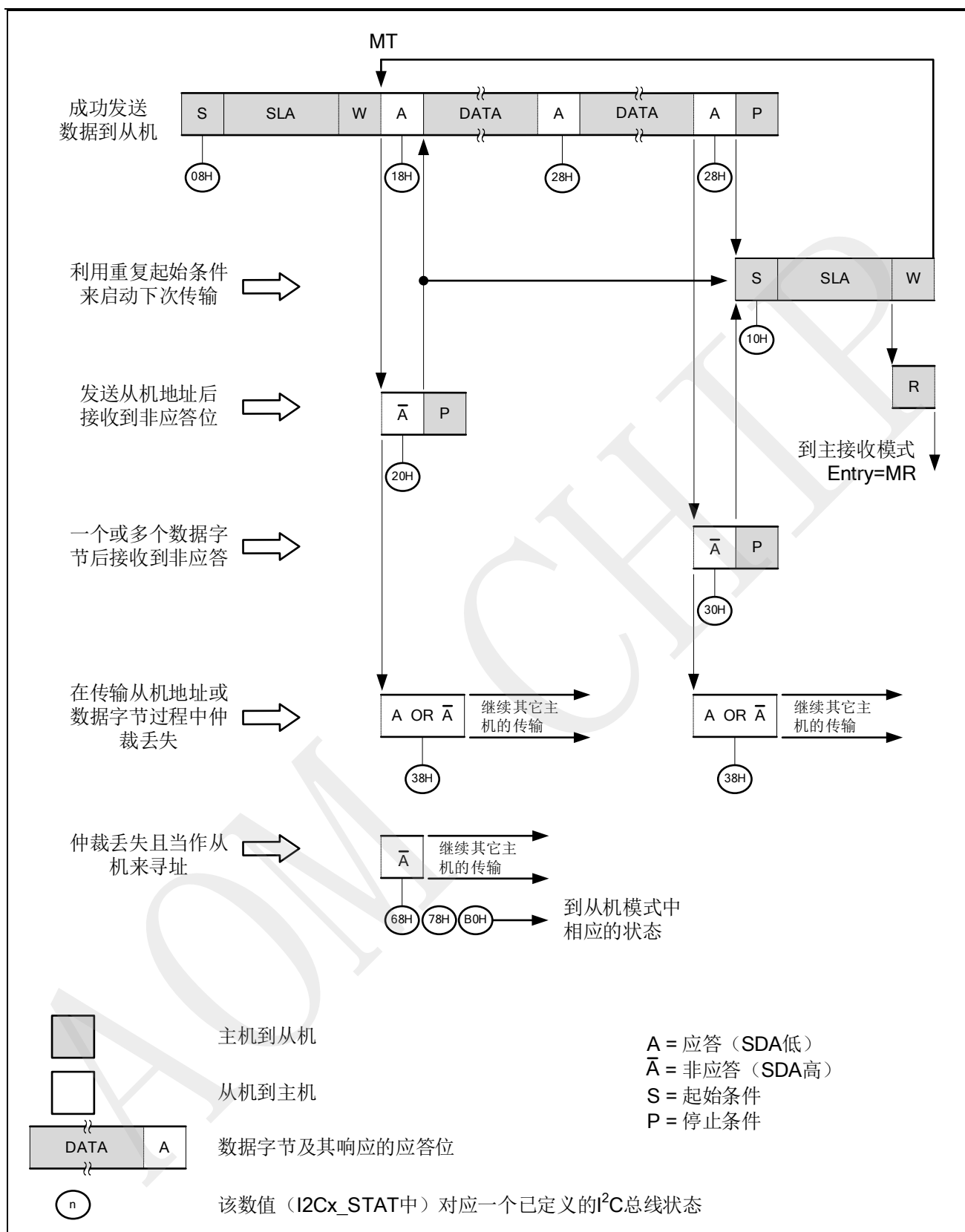
必须将 `ens` 置“1”来使能 I2C 模块。如果 `aa` 位复位, 当另一个器件正变成总线主机时, I2C 模块将不会应答其自身的从机地址或通用调用地址。换句话说, 如果 `aa` 位复位, I2C 接口就不能进入从机模式。

此时, 可通过置位 `sta` 位进入主发送模式。一旦总线空闲, I2C 逻辑会马上测试 I2C 总线并产生一个起始条件。当发送起始条件时, 串行中断标志 (`si`) 置位, 状态寄存器 (I2Cx_STAT) 中的状态代码为 0x08。中断服务程序利用该状态代码进入相应的状态服务程序, 将从机地址和数据方向位 (`SLA+W`) 装入 I2Cx_DATA。复位 I2Cx_CR 的 `si` 位后串行传输继续进行。

当发送完从机地址和方向位且接收到一个应答位时, 串行中断标志 (`si`) 再次置位, I2Cx_STAT 中可能是一系列不同的状态代码。主机模式下为 0x18, 0x20 或 0x38。每个状态代码对应的操作在下表中详细介绍。在发送完重复起始条件 (状态 0x10) 后, I2C 模块通过将 `SLA+R` 装入 I2Cx_DATA 切换到主接收模式。

状态 代码	I2C 总线和硬件状态	应用软件响应					I2C硬件执行的下一个动作
		读/写 I2Cx_DATA	写 I2Cx_CR				
			sta	sto	si	aa	
08H	已发送起始条件	装入SLA+W	X	0	0	X	将发送SLA+W，接收ACK
10H	已发送重复起始条件	装入SLA+W	X	0	0	X	同上
		装入SLA+R	X	0	0	X	将发送SLA +R，I2C自动切换到主接收模式
18H	已发送SLA+W 已接收ACK	装入数据字节	0	0	0	X	将发送数据字节，将接收ACK
		无I2Cx_DATA 动作	1	0	0	X	将发送重复起始条件
		无I2Cx_DATA 动作	0	1	0	X	将发送停止条件，sto标志位复位
		无I2Cx_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件，sto标志位复位
20H	已发送SLA+W 已接收非ACK	装入数据字节	0	0	0	X	将发送数据字节，将接收ACK
		无I2Cx_DATA 动作	1	0	0	X	将发送重复起始条件
		无I2Cx_DATA 动作	0	1	0	X	将发送停止条件，sto标志位复位
		无I2Cx_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件，sto标志位复位
28H	已发送I2Cx_DATA中的数据；已接收ACK	装入数据字节	0	0	0	X	将发送数据字节，将接收ACK
		无I2Cx_DATA 动作	1	0	0	X	将发送重复起始条件
		无I2Cx_DATA 动作	0	1	0	X	将发送停止条件，sto标志位复位
		无I2Cx_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件，sto标志位复位
30H	已发送I2Cx_DATA中的数据；	装入数据字节	0	0	0	X	将发送数据字节，将接收 ACK
		无I2Cx_DATA 动作	1	0	0	X	将发送重复起始条件
		无I2Cx_DATA 动作	0	1	0	X	将发送停止条件，sto标志位复位
		无I2Cx_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件，sto标志位复位
38H	在SLA+读写或写数据字节时丢失仲裁	无 I2Cx_DATA 动作	0	0	0	X	I2C总线被释放。进入不可寻址从模式。
		无 I2Cx_DATA 动作	1	0	0	X	当I2C总线空闲时发送起始条件

表 17-2 I2C 主机发送模式状态



17.4.8 I2C 主机接收模式

在主接收模式中，主机所接收的数据字节来自从发送器。按主发送模式中的方法初始化传输。当发送完起始条件后，中断服务程序必须把 7 位从机地址和数据方向位（SLA+R）装入 I2Cx_DATA。清除 I2Cx_CR 中的 si 位后串行传输才会继续执行。当发送完从机地址和数据方向位且接收到一个应答位时，串行中断标志 si 再次置位，这时，I2Cx_STAT 中可能是一系列不同的状态代码。主机模式下为 0x40, 0x48 或 0x38。每个状态代码对应的操作详见下表。在发送完重复起始条件（状态 0x10）后，I2C 模块通过将 SLA+W 装入 I2Cx_DATA 切换到主发送模式。

状态 代码	I2C 总线和硬件状 态	应用软件响应					I2C硬件执行的下一个动作
		读/写 I2Cx_DATA	写 I2Cx_CR				
			sta	sto	si	aa	
08H	已发送起始条件	装入SLA+R	X	0	0	X	将发送SLA+R，接收ACK
10H	已发送重复起始条 件	装入SLA+R	X	0	0	X	同上
		装入SLA+W	X	0	0	X	将发送SLA +W，I2C自动切换到主发送模式
38H	在非ACK中丢失仲 裁	无I2Cx_DATA 动作	0	0	0	X	I2C总线将被释放；进入从模式
		无I2Cx_DATA 动作	1	0	0	X	当总线空闲时发起起始条件
40H	已发送SLA +R 已 接收ACK	无I2Cx_DATA 动作	0	0	0	0	将接收数据字节，将返回非ACK
		无I2Cx_DATA 动作	0	0	0	1	将接收数据字节，将返回ACK
48H	已发送SLA +R 已 接收非ACK	无I2Cx_DATA 动作	1	0	0	X	将发送重复起始条件
		无I2Cx_DATA 动作	0	1	0	X	将发送停止条件，sto标志位复位
		无I2Cx_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件， sto 标志位复位
50H	已接收数据字节， ACK已返回	读取数据字节	0	0	0	0	将接收数据字节，将返回非 ACK
		读取数据字节	0	0	0	1	将接收数据字节，将返回ACK
58H	已接收数据字节， 非ACK已返回	读取数据字节	1	0	0	X	将发送重复起始条件
		读取数据字节	0	1	0	X	将发送停止条件，sto标志位复位

表 17-3 I2C 主机接收模式状态表

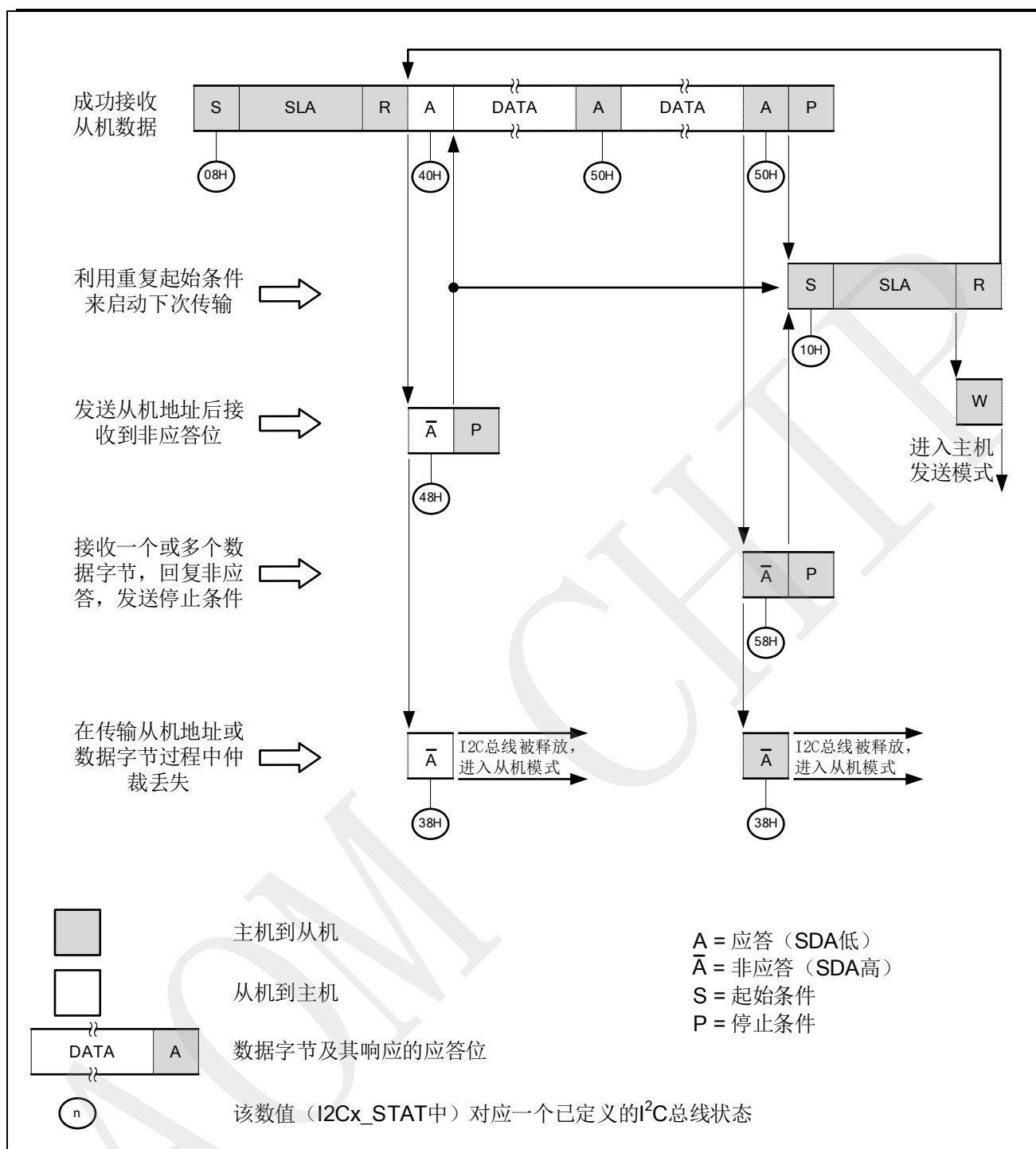


图 17-10 I2C 主机接收状态图

17.4.9 I2C 从机接收模式

在从接收模式中，从机接收的数据字节来自主发送器。寄存器 I2Cx_ADDR 的高 7 位是主机寻址时 I2C 模块响应的地址。如果 LSB (GC) 被置位，I2C 模块将响应通用调用地址 (0x00)；否则忽略通用调用地址。

I2C 总线速率的设置不影响从机模式中的 I2C 模块。必须置位 ena 来使能 I2C 模块。

aa 位必须置位以使能 I2C 模块来应答其自身从机地址或通用调用地址。sta, sto 和 si 必须复位。

当 I2Cx_ADDR 和 I2Cx_CR 完成初始化后, I2C 模块一直等待, 直至被从机地址寻址, 之后是数据方向位寻址, 为了工作在从接收模式中, 数据方向位必须为“0” (W)。接收完其自身的从机地址和 W 位后, 串行中断标志 (si) 置位, 可从 I2Cx_STAT 中读出一个有效的状态代码。该状态代码用作状态服务程序的向量。每个状态代码的对应操作见下表。如果 I2C 模块在主机模式中仲裁丢失, 也可进入从接收模式 (请参考状态 0x68 和 0x78 的描述)。

如果 aa 位在传输过程中复位, 则在接收完下一个数据字节后 I2C 模块将向 SDA 返回一个非应答 (逻辑 1)。当 aa 复位时, I2C 模块不响应其自身的从机地址或通用调用地址。但是, I2C 总线仍被监控, 而且, 地址识别可随时通过置位 aa 来恢复。这就意味着 aa 位可临时将 I2C 模块从 I2C 总线上分离出来。

状态 代码	I2C总线和硬件状 态	应用软件响应					I2C硬件执行的下一个动作
		读/写I2Cx_DATA	写I2Cx_CR				
			sta	sto	si	aa	
60H	已接收自身的SLA+W; 已发送ACK	无I2Cx_DATA动作	X	0	0	0	将接收数据字节, 将返回非ACK
		无I2Cx_DATA动作	X	0	0	1	将接收数据字节, 将返回ACK
68H	主控时在SLA+读写丢失仲裁; 已接收自身的SLA+W; 已返回ACK;	无I2Cx_DATA动作	X	0	0	0	将接收数据字节, 将返回非ACK
		无I2Cx_DATA动作	X	0	0	1	将接收数据字节, 将返回ACK
70H	已接收通用调用地址 (0x00); 已返回ACK;	无I2Cx_DATA动作	X	0	0	0	将接收数据字节, 将返回非ACK
		无I2Cx_DATA动作	X	0	0	1	将接收数据字节, 将返回ACK
78H	主控时在SLA+读写中丢失仲裁; 已接收通用调用地址; 已返回ACK;	无 I2Cx_DATA 动作	X	0	0	0	将接收数据字节, 将返回非ACK
		无I2Cx_DATA动作	X	0	0	1	将接收数据字节, 将返回ACK
80H	前一次寻址使用自身从地址; 已接收数据字节; 已返回ACK;	无 I2Cx_DATA 动作	X	0	0	0	将接收数据字节, 将返回非ACK
		无I2Cx_DATA动作	X	1	0	1	将接收数据字节, 将返回ACK
88H	前一次寻址使用自身从地址; 已接收数据字节; 已返回	读取数据字节	0	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		读取数据字节	0	0	0	1	切换到不可寻址从模式;

	非ACK;						不识别自身从地址或通用地址;
		读取数据字节	1	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		读取数据字节	1	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;
90H	前一次寻址使用通用调用地址; 已接收数据; 已返回ACK;	读取数据字节	1	0	0	X	将接收数据字节, 将返回非ACK
		读取数据字节	0	1	0	X	将接收数据字节, 将返回ACK
98H	前一次寻址使用通用调用地址; 已接收数据; 已返回非ACK;	读取数据字节	0	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		读取数据字节	0	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		读取数据字节	1	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;
		读取数据字节	1	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;
A0H	当使用从接收/从发送模式中静态寻址时, 接收到停止条件或重复起始条件	无I2Cx_DATA动作	0	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		无I2Cx_DATA动作	0	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		无I2Cx_DATA动作	1	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;
		无I2Cx_DATA动作	1	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;

表 17-4 I2C 从机接收模式状态表

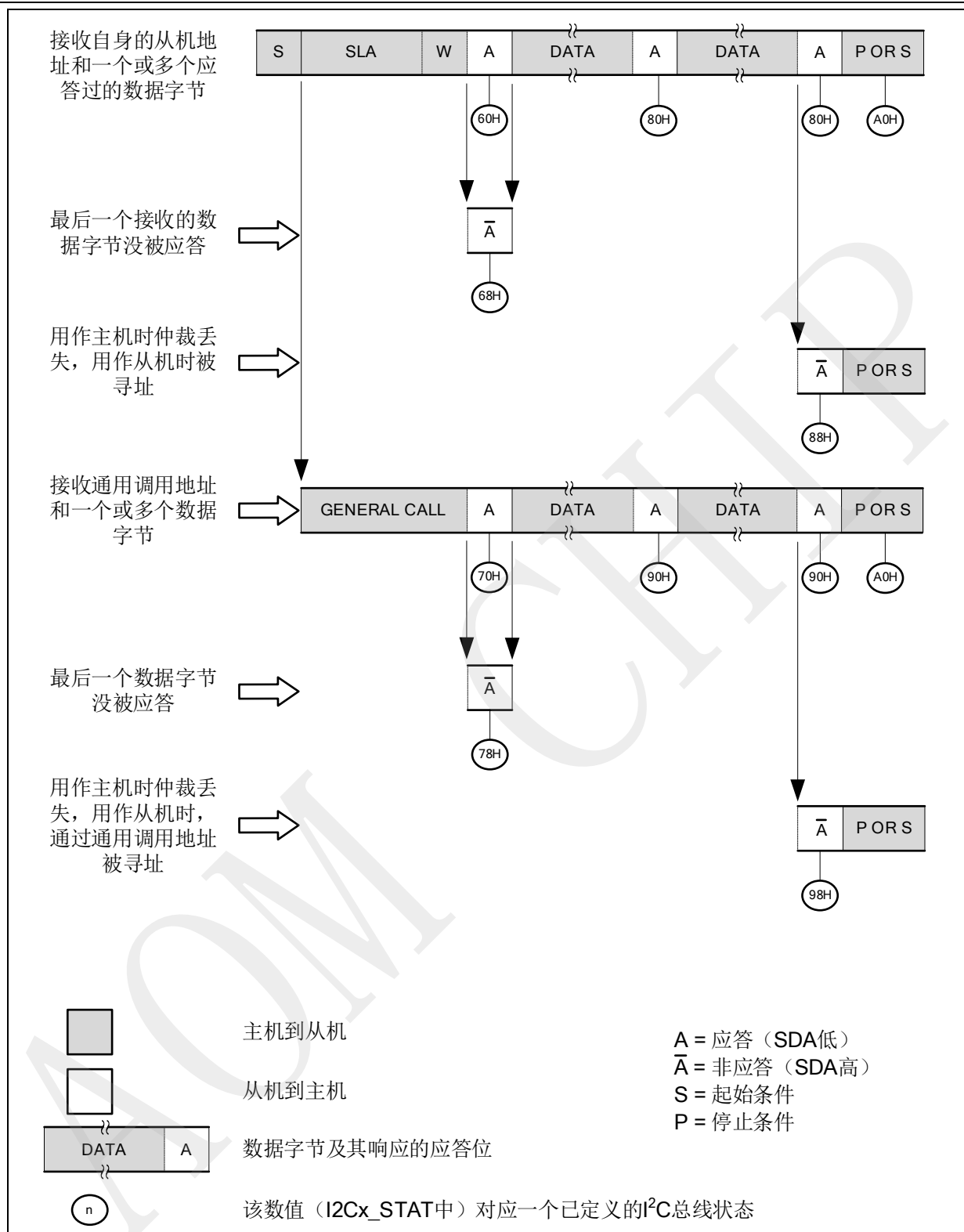


图 17-11 从机接收状态图

17.4.10 I2C 从机发送模式

在从发送模式中，向主接收器发送数据字节。数据传输按照从接收模式中的情况初

始化。当初始化 I2Cx_ADDR 和 I2Cx_CR 后, I2C 模块一直等待, 直至被自身的从机地址寻址, 之后是数据方向位, 该数据方向位必须为“1”(R), 以便 I2C 模块工作在从发送模式下。接收完其自身的从机地址和 R 位后, 串行中断标志 (si) 置位, 并且可从 I2Cx_STAT 中读取一个有效的状态代码。该状态代码用作状态服务程序的向量, 每个状态代码的对应操作见下表所示。如果 I2C 模块在主机模式下时仲裁丢失, 则可进入从发送模式 (见状态 0xB0)。

如果 aa 位在传输过程中复位, 则 I2C 模块将发送最后一个字节并进入状态 0xC0 或 0xC8。I2C 模块切换到非寻址的从机模式, 如果继续传输, 它将忽略主接收器。因此主接收器接收所有 1 作为串行数据。当 aa 复位时, I2C 模块不响应其自身的从机地址或通用调用地址。但是, I2C 总线仍被监控, 而且, 地址识别可随时通过置位 aa 来恢复。这就意味着 aa 位可用来暂时将 I2C 模块从 I2C 总线上分离出来。

状态 代码	I2C总线和硬件状 态	应用软件响应					I2C硬件执行的下一个动作
		读/写 I2Cx_DATA	写 I2Cx_CR				
			sta	sto	si	aa	
A8H	已接收自身的 SLA+R; 已返回 ACK	装入数据字节	X	0	0	0	将发送最后一个数据字节; 将接收非ACK;
		装入数据字节	X	0	0	1	将发送一个数据字节; 将接收ACK;
B0H	当主控时在SLA+ 读写中丢失仲裁; 已接收自身 SLA+R; 已返回非 ACK;	装入数据字节	X	0	0	0	将发送最后一个数据字节; 将接收非ACK;
		装入数据字节	X	0	0	1	将发送一个数据字节; 将接收ACK;
B8H	已发送数据; 已接 收ACK;	装入数据字节	X	0	0	0	将发送最后一个数据字节; 将接收非ACK;
		装入数据字节	X	0	0	1	将发送一个数据字节; 将接收ACK;
C0H	已发送数据字节; 已接收非ACK;	无I2Cx_DATA动 作	0	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		无I2Cx_DATA动 作	0	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址;
		无I2Cx_DATA动 作	1	0	0	0	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;
		无I2Cx_DATA动 作	1	0	0	1	切换到不可寻址从模式; 不识别自身从地址或通用地址; 当总线空闲后发送起始条件;

情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

I2Cx_STAT= 0x00:

该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时 si 置位。要从总线错误中恢复，sto 标志必须置位，si 必须被清除。这使得 I2C 模块进入“非寻址的”从机模式（已定义的状态）并清除 sto 标志（I2Cx_CR 中的其它位不受影响）。SDA 和 SCL 线被释放（不发送停止条件）。

状态 代码	I2C 总线和硬件状态	应用软件响应				I2C硬件执行的下一个动作	
		读/写 I2Cx_DATA	写 I2Cx_CR				
			sta	sto	si		aa
F8H	无可用的相关状态信息； si=0;	无I2Cx_DATA动作	无I2Cx_DATA动作				等待或执行当前传输
00H	由于非法的起始或停止条件的出现，在主机或被选中的从机将出现总线错误； 当外部干扰使I2C进入未定义的状态时也会出0x00状态	无I2Cx_DATA动作	0	1	0	X	只有在主机或被寻址的从机模式中，内部硬件受影响。一般情况下，总线被释放，I2C模块切换到非寻址的从机模式。sto复位。

表 17-6 其他杂项状态表

17.5 I2C 操作模式

17.5.1 初始化程序

将 I2C 接口初始化用作从机和/或主机的例子。

- 将自身的从机地址装入 I2Cx_ADDR，使能通用调用识别（如果需要的话）；
- 使能 I2C 中断；

- c) 向寄存器 I2Cx_CR 写入 0x44 来置位 ens 和 aa 位，并使能从机功能。对于主机功能，可向寄存器 I2Cx_CR 写入 0x40。

17.5.2 启动主机发送功能

通过建立缓冲区、指针和数据计数然后发起起始条件便可执行主发送操作。

- a) 初始化主机数据计数器；
- b) 建立数据将被发送到的从机地址，并且添加写位；
- c) 置位 I2Cx_CR.sta；
- d) 在主发送缓冲区内建立要发送的数据；
- e) 初始化主机数据计数器来匹配正在发送的信息长度；
- f) 退出。

17.5.3 启动主机接收功能

通过建立缓冲区、指针和数据计数然后发起起始条件便可执行主接收操作。

- a) 初始化主机数据计数器；
- b) 建立数据将被发送到的从机地址，并且添加读位；
- c) 置位 I2Cx_CR.sta；
- d) 在主接收缓存区内存放接收到的数据；
- e) 初始化主机数据计数器来匹配正在发送的信息长度；
- f) 退出。

17.5.4 I2C 中断程序

确定 I2C 的状态和处理该状态的状态程序。

- a) 从 I2Cx_STAT 中读出 I2C 的状态；
- b) 使用状态值跳转到 26 个可能状态程序中的一个。

17.5.5 无指定模式的状态

- 1) 状态：0x00 总线错误。

进入非寻址的从机模式并释放总线。

- a) 置位 I2Cx_CR.sto 和 I2Cx_CR.aa;
- b) 清除 I2Cx_CR.si;
- c) 退出。

2) 主机状态

状态 0x08 和 0x10 适用于主发送模式和主接收模式。读写位决定了下一个状态是在主发送模式中还是在主接收模式中。

3) 状态：0x08

已发送起始条件。即将发送从机地址+读写位和接收 ACK 位。

- a) 向 I2Cx_DATA 写入从机地址和读写位;
- b) 置位 I2Cx_CR.aa;
- c) 清除 I2Cx_CR.si;
- d) 建立主发送模式数据缓冲区;
- e) 建立主接收模式数据缓冲区;
- f) 初始化主机数据计数器;
- g) 退出。

4) 状态：0x10

已发送重复起始条件。即将发送从机地址+读写位和接收 ACK 位。

- a) 向 I2Cx_DATA 写入从机地址和读写位;
- b) 置位 I2Cx_CR.aa;
- c) 清除 I2Cx_CR.si;
- d) 建立主发送模式数据缓冲区;
- e) 建立主接收模式数据缓冲区;
- f) 初始化主机数据计数器;
- g) 退出。

17.5.6 主发送器状态

1) 状态：0x18

之前状态为 0x08 或 0x10 表示已发送从机地址和写操作位，并接收了应答。即将发送第一个数据字节和接收 ACK 位。

- a) 将主发送缓冲区的第一个数据字节装入 I2Cx_DATA;
- b) 清除 I2Cx_CR.si;
- c) 主发送缓冲区指针加 1;
- d) 退出。

2) 状态: 0x20

已发送从机地址和写操作位并接收了非应答。即将发送停止条件。

- a) 置位 I2Cx_CR.sto;
- b) 清除 I2Cx_CR.si;
- c) 退出。

3) 状态: 0x28

已发送数据并接收了 ACK。如果发送的数据是最后一个数据字节则发送一个停止条件，否则发送下一个数据字节。

- a) 主机数据计数器减 1，如果发送的不是最后一个数据字节就跳至第 e) 步;
- b) 置位 I2Cx_CR.sto;
- c) 清除 I2Cx_CR.si;
- d) 退出;
- e) 将主发送缓冲区的下一个数据字节装入 I2Cx_DATA;
- f) 清除 I2Cx_CR.si;
- g) 主机发送缓冲区指针加 1;
- h) 退出。

4) 状态: 0x30

已发送数据并接收到非应答。即将发送停止条件;

- a) 置位 I2Cx_CR.sto;
- b) 清除 I2Cx_CR.si;
- c) 退出。

5) 状态: 0x38

仲裁已在发送从机地址和写操作位或数据的过程中丢失。总线已被释放且进入非寻址的从机模式。当总线再次空闲时将发送一个新的起始条件。

- a) 置位 I2Cx_CR.sta;
- b) 清除 I2Cx_CR.si;
- c) 退出。

17.5.7 主接收器状态

1) 状态: 0x40

前面的状态是 0x08 或 0x10 表示已发送从机地址和读操作位，并接收到 ACK。将接收数据和返回 ACK。

- a) 置位 I2Cx_CR.aa;
- b) 清除 I2Cx_CR.si;
- c) 退出。

2) 状态: 0x48

已发送从机地址和读操作位，并接收到非应答。将发送停止条件。

- a) 置位 I2Cx_CR.sto 和 I2Cx_CR.aa;
- b) 清除 I2Cx_CR.si;
- c) 退出。

3) 状态: 0x50

已接收到数据，并返回 ACK。将从 I2Cx_DATA 读取数据。将接收其它的数据。如果这是最后一个数据字节，则返回非应答，否则返回 ACK。

- a) 读取 I2Cx_DATA 中的数据字节，存放到主机接收缓冲区;
- b) 主机数据计数器减 1，如果不是最后一个数据字节就跳到第 f) 步;
- c) 置位 I2Cx_CR.aa;
- d) 清除 I2Cx_CR.si;
- e) 退出;
- f) 置位 I2Cx_CR.aa;
- g) 清除 I2Cx_CR.si;

- h) 主机接收缓冲区指针加 1;
 - i) 退出。
- 4) 状态: 0x58
- 已接收到数据, 已返回非应答。将从 I2Cx_DATA 中读取数据和发送停止条件。
- a) 读取 I2Cx_DATA 中的数据字节, 存放到主机接收缓冲区;
 - b) 置位 I2Cx_CR.sto 和 I2Cx_CR.aa;
 - c) 清除 I2Cx_CR.si;
 - d) 退出。

17.5.8 从接收器状态

- 1) 状态: 0x60
- 已接收到自身从机地址和写操作位, 已返回 ACK。将接收数据和返回 ACK。
- a) 置位 I2Cx_CR.aa;
 - b) 清除 I2Cx_CR.si;
 - c) 建立从接收模式数据缓冲区;
 - d) 初始化从机数据计数器;
 - e) 退出。
- 2) 状态: 0x68
- 用作总线主机时仲裁已在传输从机地址和读写位时丢失。已接收到自身从机地址和写操作位, 并已返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 sta 来重启主机模式。
- a) 置位 I2Cx_CR.sta 和 I2Cx_CR.aa;
 - b) 清除 I2Cx_CR.si;
 - c) 建立从接收模式数据缓冲区;
 - d) 初始化从机数据计数器;
 - e) 退出。
- 3) 状态: 0x70
- 已接收到通用调用和返回 ACK。将接收数据和返回 ACK。

- a) 置位 I2Cx_CR.aa;
- b) 清除 I2Cx_CR.si;
- c) 建立从接收模式数据缓冲区;
- d) 初始化从机数据计数器;
- e) 退出。

4) 状态: 0x78

用作总线主机时仲裁已在传输从机地址和读写位时丢失。已接收到通用调用和返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 sta 来重启主机模式。

- a) 置位 I2Cx_CR.sta 和 I2Cx_CR.aa;
- b) 清除 I2Cx_CR.si;
- c) 建立从接收模式数据缓冲区;
- d) 初始化从机数据计数器;
- e) 退出。

5) 状态: 0x80

之前寻址自身从机地址。已接收到数据并返回 ACK。将读取其它数据。

- a) 读取 I2Cx_DATA 的数据字节, 存放到从机接收缓冲区。
- b) 从机数据计数器减 1, 如果不是最后一个数据字节就跳到第 f) 步;
- c) 清除 I2Cx_CR.aa;
- d) 清除 I2Cx_CR.si;
- e) 退出;
- f) 置位 I2Cx_CR.aa;
- g) 清除 I2Cx_CR.si;
- h) 从机接收缓冲区指针加 1;
- i) 退出。

6) 状态: 0x88

之前寻址自身从机地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

- a) 置位 I2Cx_CR.aa;

b) 清除 I2Cx_CR.si;

c) 退出。

7) 状态：0x90

之前寻址通用调用地址。已接收到数据并返回 ACK。将保存接收到的数据。只接收第一个数据字节并返回 ACK。接收其它数据字节后返回非应答。

a) 读取 I2Cx_DATA 的数据字节，并放入从机接收缓冲区；

b) 清除 I2Cx_CR.aa;

c) 清除 I2Cx_CR.si;

d) 退出。

8) 状态：0x98

之前寻址通用调用地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

a) 置位 I2Cx_CR.aa;

b) 清除 I2Cx_CR.si;

c) 退出。

9) 状态：0xA0

已接收停止条件或重复起始条件，但仍作为从机寻址。不保存接收到的数据。进入非寻址的从机模式。

a) 置位 I2Cx_CR.aa;

b) 清除 I2Cx_CR.si;

c) 退出。

17.5.9 从发送器状态

1) 状态：0xA8

已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。

a) 将从机发送缓冲区的第一个数据字节装入 I2Cx_DATA;

b) 置位 I2Cx_CR.aa;

c) 清除 I2Cx_CR.si;

- d) 建立从发送模式数据缓冲区;
- e) 从机发送缓冲区指针加 1;
- f) 退出。

2) 状态: 0xB0

用作总线主机时, 在传输从机地址和读写位时丢失仲裁。已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。当总线再次空闲后置位 sta 来重启主机模式。

- a) 将从机发送缓冲区的第一个数据字节装入 I2Cx_DATA;
- b) 置位 I2Cx_CR.sta 和 I2Cx_CR.aa;
- c) 清除 I2Cx_CR.si;
- d) 建立从发送模式数据缓冲区;
- e) 从机发送缓冲区指针加 1;
- f) 退出。

3) 状态: 0xB8

已发送数据并接收到 ACK。将发送数据和接收 ACK 位。

- a) 将从机发送缓冲区的数据字节装入 I2Cx_DATA;
- b) 置位 I2Cx_CR.aa;
- c) 清除 I2Cx_CR.si;
- d) 从机发送缓冲区指针加 1;
- e) 退出。

4) 状态: 0xC0

已发送数据并接收到非应答。进入非寻址的从机模式。

- a) 置位 I2Cx_CR.aa;
- b) 清除 I2Cx_CR.si;
- c) 退出。

5) 状态: 0xC8

已发送最后一个数据字节并接收到 ACK。进入非寻址的从机模式。

- a) 置位 I2Cx_CR.aa;

- b) 清除 I2Cx_CR.si;
- c) 退出。

17.6 I2C 寄存器描述

寄存器列表

I2C1 基地址: 0x40005400

I2C2 基地址: 0x40005800

I2C3 基地址: 0x40005C00

I2C4 基地址: 0x40006800

寄存器名称	偏移量	访问	寄存器描述
I2Cx_TMRUN	0x00	RW	I2C 波特率计数器使能寄存器
I2Cx_TM	0x04	RW	I2C 波特率计数器配置寄存器
I2Cx_CR	0x08	RW	I2C 配置寄存器
I2Cx_DATA	0x0C	RW	I2C 数据寄存器
I2Cx_ADDR	0x10	RW	I2C 地址寄存器
I2Cx_STAT	0x14	RO	I2C 状态寄存器

表 17-7 寄存器列表

17.6.1 I2C 波特率计数器使能寄存器(I2Cx_TMRUN)

地址偏移量: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														tme	
														RW	

位	标记	功能描述
31: 1	Reserved	保留
0	tme	波特率计数器使能 0: 禁止 1: 使能

17.6.2 I2C 波特率计数器配置寄存器(I2Cx_TM)

地址偏移量：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								tm							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	tm	tm:波特率计数器配置值 $F_{scl} = F_{pclk}/8*(tm+1)$, $tm > 0$

17.6.3 I2C 配置寄存器(I2Cx_CR)

地址偏移量: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ens	sta	sto	si	aa	RSV	h1m
									RW	RW	RW	RW	RW		RW

位	标记	功能描述
31:7	Reserved	保留
6	ens	I2C 模块使能 0: 禁止 1: 使能
5	sta	开始标志使能 0: 禁止 1: 使能
4	sto	停止标志使能 0: 禁止 1: 使能
3	si	I2C 中断标志位
2	aa	应答标志使能 0: 禁止 1: 使能
1	Reserved	保留
0	h1m	I2C 高速 1Mbps 模式使能 0: 禁止 1: 使能 当 I2Cx_TM<9 时必须使能

17.6.4 I2C 数据寄存器(I2Cx_DATA)

地址偏移量: 0x0c

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cdat							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	i2cdat	I2C 数据寄存器 在 I2C 发送模式下, 写发送数据到这个寄存器 在 I2C 接收模式下, 读接收数据从这个寄存器

17.6.5 I2C 地址寄存器(I2Cx_ADDR)

地址偏移量: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cadr						GC	
								RW						RW	

位	标记	功能描述
31:8	Reserved	保留
7:1	i2cadr	I2C 从机模式地址
0	GC	广播地址应答使能 0: 禁止 1: 使能

17.6.6 I2C 状态寄存器(I2Cx_STAT)

地址偏移量: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								STAT							
								RO							

位	标记	功能描述
31:8	Reserved	保留
7:0	STAT	I2C 状态寄存器

18. 串行外设接口（SPI）

18.1 SPI 简介

SPI（Serial Peripheral Interface）接口用于 MCU 和外部器件之间进行同步串行通信，支持 MCU 作为主机或从机与外部器件进行全双工、半双工或单工的通信。

18.2 SPI 主要特性

- 可配置为主机或者从机
- 基于两根单向数据线的全双工通信
- 基于一根单向数据线的单工通信
- 4 位到 16 位连续可调的帧宽度选择
- 支持多机模式
- 主机模式 7 种波特率可配置，最高可达 $fpclk/2$
- 从机模式波特率最高可达 $fpclk/4$
- 主机模式和从机模式都可配置 NSS 功能——硬件管脚或软件控制
- 可配置的串行时钟极性和相位
- 可配置的数据比特位顺序——先收发 MSB 或先收发 LSB
- 发送和接收标志位，支持触发中断
- SPI 总线忙标志位
- 多种错误标志位，均支持触发中断
- 发送和接收分别配有容量为一帧的支持 DMA 访问的缓冲

18.3 SPI 功能描述

18.3.1 概述

SPI 支持多种通信方式和多种帧格式，可满足不同配置外设的兼容需求。

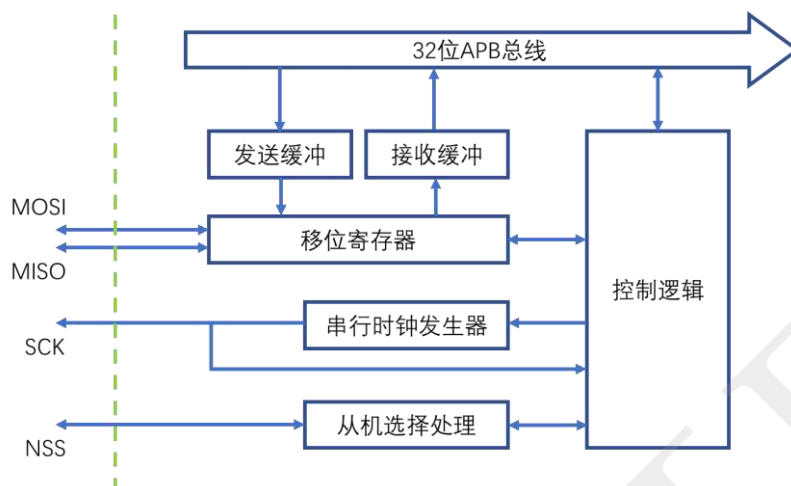


图 18-1 SPI 框图

SPI 接口一般涉及到以下管脚：

NSS：从机选择。根据配置，可用于选中从机以开始通信，或检测多机是否有总线冲突。

SCK：串行时钟。由主机产生（输出）、从机接收（输入），是主从器件同步收发数据的基准。

MISO：主机输入/从机输出的串行数据。一般用来在主机模式下接收数据或者在从机模式下发送数据。

MOSI：主机输出/从机输入的串行数据。一般用来在主机模式下发送数据或者在从机模式下接收数据。

18.3.2 通信格式和时序

SPI 是串行同步通信接口，主从机都根据串行时钟进行数据的采样和移位。其通信格式取决于串行时钟极性、串行时钟相位和数据帧格式。数据帧格式又取决于数据宽度和数据比特位顺序。SPI 接口正常通信要求所有的主机和从机必须配置为相同的通信格式。

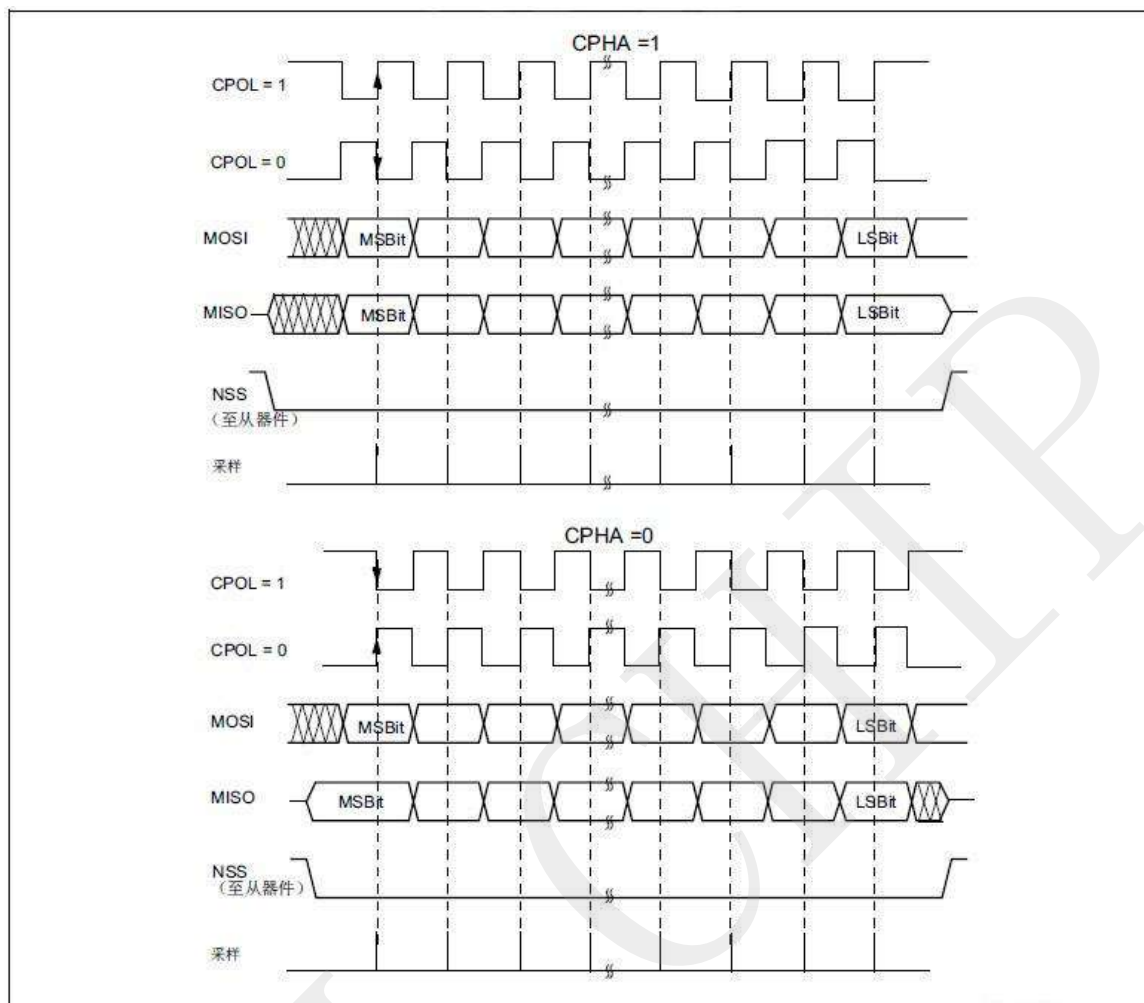


图 18-2SPI 通信格式时序图

串行时钟极性，指 SPI 不传输数据的空闲状态下串行时钟的电平状态，由寄存器 SPIx_CR1.CPOL 进行配置。当 SPIx_CR1.CPOL=0 时，串行时钟 SCK 在空闲时为低电平；当 SPIx_CR1.CPOL=1 时，串行时钟 SCK 在空闲时为高电平。

串行时钟相位，指 SPI 的主从机根据串行时钟 SCK 进行采样和移位的先后顺序，由寄存器 SPIx_CR1.CPHA 进行配置。当 SPIx_CR1.CPHA=0 时，主从机在前边沿（即发生在前的、SCK 由空闲态转为非空闲态的边沿）采样数据，后边沿（即发生在后的、SCK 由非空闲态转为空闲态的边沿）移位数据；当 SPIx_CR1.CPHA=1 时，主从机在前边沿移位数据，后边沿采样数据。

数据宽度，指每一帧（一次不可再分割的最小单元的传输）所包含的比特位数，由寄存器 SPIx_CR1.WIDTH 进行配置。SPI 通信常用的数据宽度为 8bit 或 16bit，本模块支持配置为 4bit 到 16bit 之中的任一宽度。

数据比特位顺序,指每一帧数据的收发是最高有效位优先还是最低有效位优先,由寄存器 SPIx_CR1.LSBF 进行配置。当 SPIx_CR1.LSBF=0 时,每帧数据的收发从最高有效位开始,依比特位顺序降序到最低有效位结束;当 SPIx_CR1.LSBF=1 时,每帧数据的收发从最低有效位开始,依比特位顺序升序到最高有效位结束。

18.3.3 一般状态标志和错误标志

SPI 提供多种状态标志位来指示当前的工作状态,包括一般状态标志和错误标志。一般状态标志用来指示 SPI 当前的一般工作状态。错误标志用来指示 SPI 是否发生了特定的错误。

从机选择输入状态

从机选择输入状态指示当前输入的从机选择信号的电平,对应寄存器 SPIx_SR.SSLVL。当 SPIx_SR.SSLVL=0 时,从机选择输入是低电平;当 SPIx_SR.SSLVL=1 时,从机选择输入是高电平。

需要注意的是,从机选择输入状态在主机模式且 SSM=1 时无意义,因为此配置下没有从机选择输入。

总线忙标志位

总线忙标志位指示当前 SPI 接口处于忙碌还是空闲状态,对应寄存器 SPIx_SR.BUSY。当 SPIx_SR.BUSY=0 时, SPI 接口处于空闲状态,当前没有数据传输;当 SPIx_SR.BUSY=1 时, SPI 接口处于忙碌状态,当前正在准备进行或正在进行数据传输。

从机选择输入上升沿标志位

从机选择输入上升沿标志位指示当前输入的从机选择信号是否出现了上升沿,对应寄存器 SPIx_SR.SSR。当 SPIx_SR.SSR=0 时,从机选择输入未出现上升沿;当 SPIx_SR.SSR=1 时,从机选择输入出现了上升沿。

需要注意的是,从机选择输入上升沿标志位在主机模式且 SSM=1 时无意义,因为此

配置下没有从机选择输入。

从机选择输入下降沿标志位

从机选择输入下降沿标志位指示当前输入的从机选择信号是否出现了下降沿, 对应寄存器 SPIx_SR.SSF。当 SPIx_SR.SSF=0 时, 从机选择输入未出现下降沿; 当 SPIx_SR.SSF=1 时, 从机选择输入出现了下降沿。

需要注意的是, 从机选择输入下降沿标志位在主机模式且 SSM=1 时无意义, 因为此配置下没有从机选择输入。

发送缓冲空标志位

发送缓冲空标志位指示当前 SPI 的发送缓冲是否已装载待发送数据, 对应寄存器 SPIx_SR.TXE。当 SPIx_SR.TXE=0 时, SPI 的发送缓冲已装载待发送数据; 当 SPIx_SR.TXE=1 时, SPI 的发送缓冲处于空状态, 可以将下一个待发送的数据写入。

接收缓冲非空标志位

接收缓冲非空标志位指示当前 SPI 的接收缓冲是否存在已接收的数据, 对应寄存器 SPIx_SR.RXNE。当 SPIx_SR.RXNE=0 时, SPI 的接收缓冲尚未装载有效的接收数据; 当 SPIx_SR.RXNE=1 时, SPI 的接收缓冲装载了已接收的数据, 可供读出。

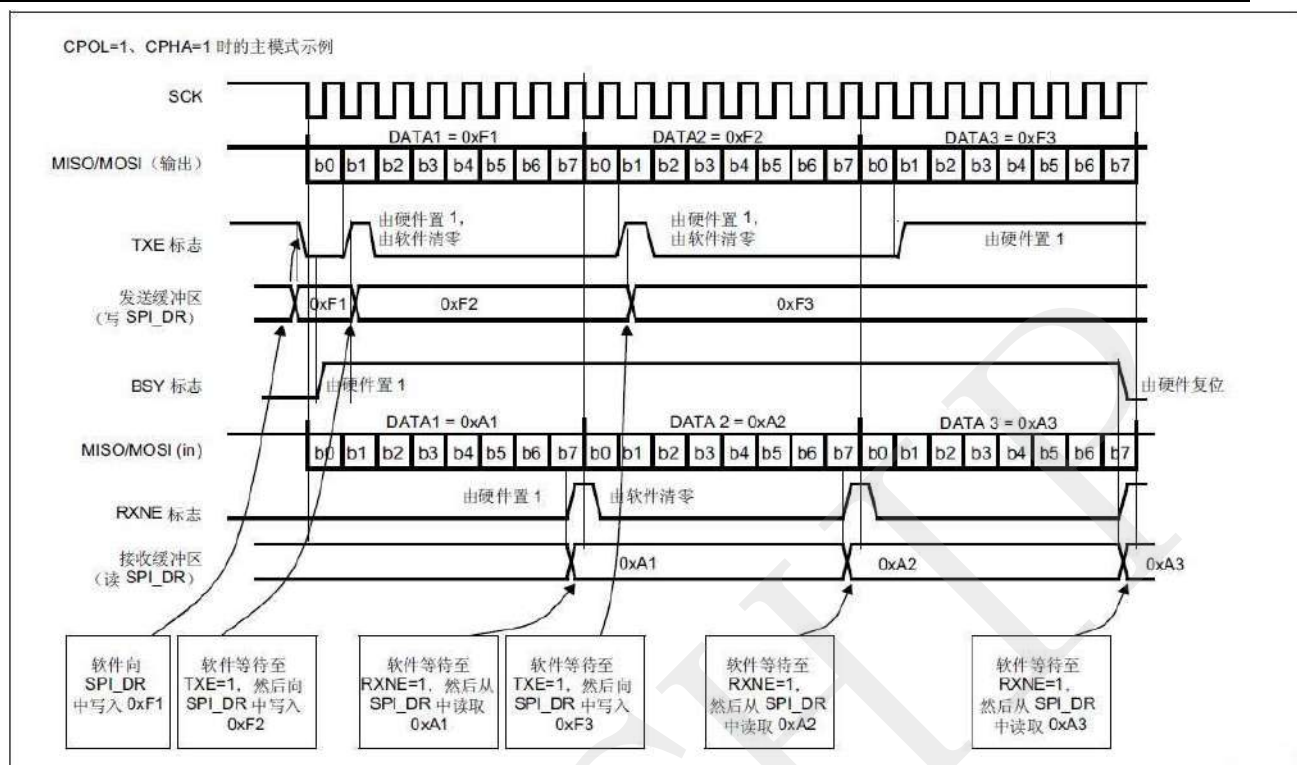


图 18-3SPI 主机模式一般标志位说明图

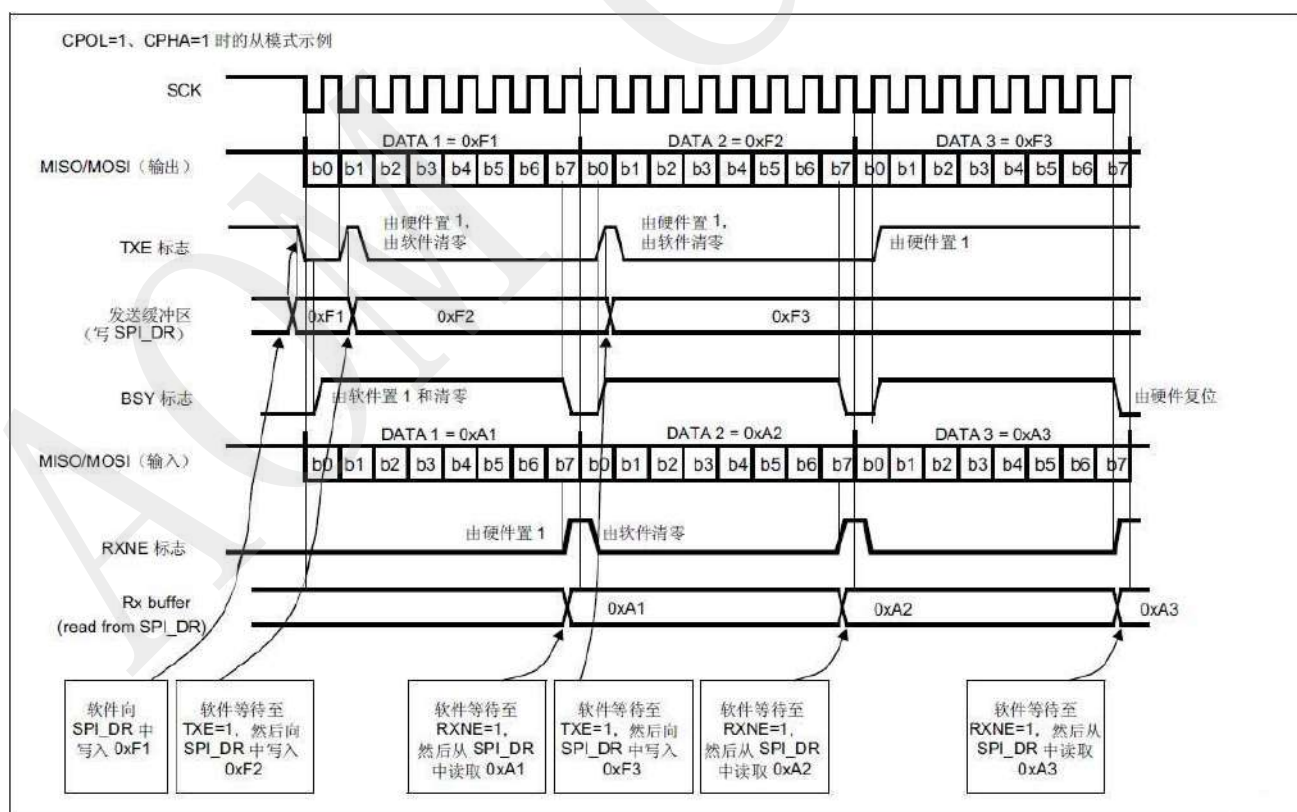


图 18-4SPI 从机模式一般标志位说明图

从机模式下的从机选择错误标志位

当 SPI 工作于从机模式且正在进行有效的数据传输时，输入的从机选择信号要保持选中状态（低电平）来保证正常的通信，如果从机选择在此期间被拉高，就会发生从机选择错误，对应的寄存器 SPIx_SR.SSERR 会置位。

发生从机选择错误，会造成对应的数据传输出错，SPI 接口也会因从机选择的拉高而进入未选中状态。从机选择错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

需要注意的是，从机选择错误 SSERR 是只出现于从机模式下的错误，且仅当有效数据尚在传输时从机选择被拉高才会触发，帧与帧之间没有有效数据传输时正常的从机选择拉高并不触发该错误。

接收缓冲上溢错误标志位

SPI 的接收缓冲器的大小为一帧，当接收缓冲中的数据没有被读出且新一帧的数据已经接收完成时，发生的就是接收缓冲上溢错误，对应的寄存器 SPIx_SR.OVF 会置位。

发生接收缓冲上溢错误，缓冲器中原有数据会被新接收的数据覆盖，造成数据的丢失。接收缓冲上溢错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

从机模式下的发送缓冲下溢错误标志位

SPI 的发送缓冲器的大小为一帧，当从机模式下发送缓冲已空且新一帧的传输开始时，发生的就是发送缓冲下溢错误，对应的寄存器 SPIx_SR.UDF 会置位。

发生从机模式下发送缓冲下溢错误，意味着发送缓冲器中没有及时填入数据，错过了给主机发送数据。从机模式下缓冲下溢错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

主机模式下的模式错误标志位

当 SPI 工作于主机模式且从机选择引脚被配置为输入时,输入的从机选择信号要一直保持非选中状态(高电平)来保证正常的通信,如果从机选择被拉低,就会发生模式错误,对应的寄存器 SPIx_SR.MODF 会置位。

发生模式错误,会造成 SPI 强制返回未使能状态(即 SPIx_CR1.EN 被清零),主从模式选择寄存器(SPIx_CR1.MSTR)也会被清零(即切换到从机模式)。模式错误发生后,对应的错误标志位置位并不阻止用户程序重新使能 SPI 并进行正常的数据传输,但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

需要注意的是,模式错误 MODF 是只出现于主机模式特定配置下的错误,仅当从机选择被配置为输入且被拉高才会触发,其他情况不会触发该错误。

18.3.4 中断的产生和清除

SPI 支持从不同的事件产生中断请求,具体信息如下:

中断事件	对应标志位 (SPIx_SR)	对应中断使能 (SPIx_CR2)	清除中断标志的方法
从机选择输入上升沿	SSR	SSRIE	向 SPIx_ICR.SSRC 写 0
从机选择输入下降沿	SSF	SSFIE	向 SPIx_ICR.SSFC 写 0
发送缓冲空	TXE	TXEIE	向数据寄存器 DR 中写数据
接收缓冲非空	RXNE	RXNEIE	从数据寄存器 DR 中读数据 或 向 SPIx_ICR.RXNEC 写 0
从机模式下从机选择错误	SSERR	SSERRIE	向 SPIx_ICR.SSERRC 写 0
接收缓冲上溢错误	OVF	OVFIE	向 SPIx_ICR.OVFC 写 0
从机模式发送缓冲下溢错误	UDF	UDFIE	向 SPIx_ICR.UDFC 写 0
主机模式下模式错误	MODF	MODFIE	向 SPIx_ICR.MODFC 写 0

表 18-1SPI 中断

如上表,每种中断事件都有独立的中断使能控制,其中由错误事件产生的中断也都有独立的中断清除寄存器。对于上述的中断,无论中断是否使能,都可以使用表中所述的方法来清除中断标志。

18.3.5 从机选择（NSS）引脚的配置

SPI 提供了对从机选择 NSS 引脚的多种配置来适配不同的应用场景。

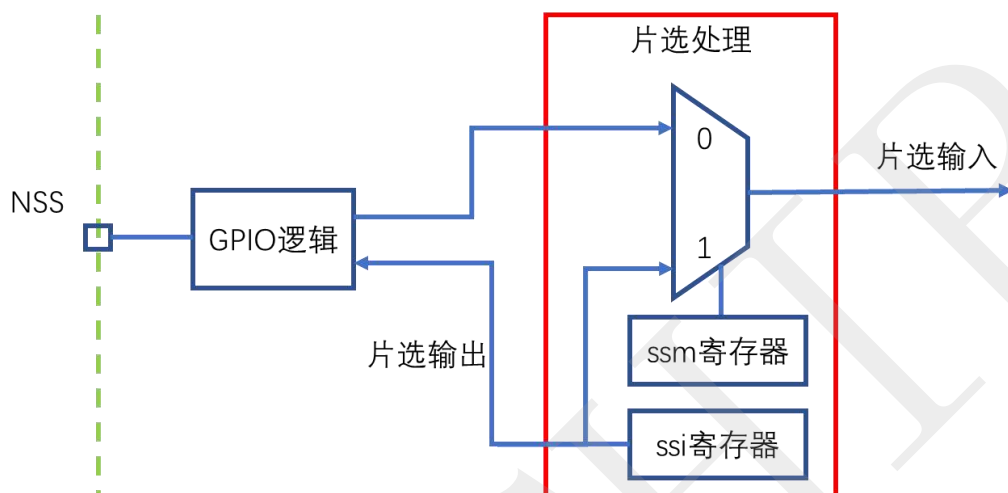


图 18-5 从机选择 NSS 引脚配置

主机模式且 $SPIx\_CR1.SSM=0$ 时，从机选择 NSS 是输入，其状态由 NSS 引脚的电平决定。这种配置一般用在多机系统中，从机选择 NSS 引脚作为输入被用来检测多机通信中可能出现的总线冲突，从机选择输出则需要借助 GPIO 来实现。

主机模式且 $SPIx\_CR1.SSM=1$ 时，从机选择 NSS 是输出，NSS 引脚的输出电平由内部寄存器 $SPIx\_SSI.SSI$ 决定。当 $SPIx\_SSI.SSI=0$ 时，NSS 引脚输出低电平；当 $SPIx\_SSI.SSI=1$ 时，NSS 引脚输出高电平。需要注意的是，当不需要从机选择引脚输出时，该引脚可供其他功能使用，但是用户程序仍然需要按照正常流程操作内部寄存器 $SPIx\_SSI.SSI$ （通信开始前必须先向 SSI 写 0 来选中从机，通信结束后可以向 SSI 写 1 来取消选中从机）来保证功能正常。

从机模式且 $SPIx\_CR1.SSM=0$ 时，从机选择 NSS 是输入，其状态由 NSS 引脚的电平决定。当 NSS 引脚输入为低电平时被选中；当 NSS 引脚输入为高电平时取消选中。这种配置是从机模式常用的配置。

从机模式且 $SPIx\_CR1.SSM=1$ 时，从机选择 NSS 对应的引脚未被 SPI 使用，供其他功能使用，SPI 根据内部寄存器 $SPIx\_SSI.SSI$ 决定是否被选中。当 $SPIx\_SSI.SSI=0$ 时被选中；当 $SPIx\_SSI.SSI=1$ 时取消选中。这种配置允许用户程序以软件方式来模拟从机选择输入，使 SPI 在从机模式下节省了从机选择引脚。

18.4 SPI 软件操作流程

SPI 提供丰富的配置和状态寄存器以及对应的中断，支持寄存器查询/DMA 等多种数据收发方式，方便软件的操作，也提高了灵活性和兼容性。

18.4.1 初始化配置

在开始使用 SPI 时，需要对 SPI 进行初始化的配置以保证正常的通信。初始化配置遵循以下步骤：

1. 使能 GPIO 和 SPI 的时钟并释放其复位（参见系统控制章节）
2. 配置 SPI 所用到的引脚（参见 GPIO 章节）
3. 配置 SPIx_CR1 寄存器
 - 1) 根据设定的主从机角色配置 SPIx_CR1.MSTR;
 - 2) 如果是主机，还要根据设定的波特率配置 SPIx_CR1.BRR，如果是从机则忽略 SPIx_CR1.BRR;
 - 3) 根据设定的通信模式（单工/半双工/双工）配置 SPIx_CR1.CM;
 - 4) 根据设定的从机选择模式（硬件引脚或软件）配置 SPIx_CR1.SSM;
 - 5) 根据设定的串行时钟格式（极性和相位）配置 SPIx_CR1.CPOL、SPIx_CR1.CPHA;
 - 6) 根据数据帧格式（数据宽度以及比特位顺序）配置 SPIx_CR1.WIDTH 和 SPIx_CR1.LSBF;

需要注意的是，SPIx_CR1 包含了 SPI 的通信格式配置寄存器，可以随时进行改写，但是不建议用户程序在 SPI 运行时进行改写，以避免不可预测的严重错误。如果需要在 SPI 使能的状态下进行改写，请保证 SPI 处于空闲状态（SPIx_SR.BUSY=0）、收发缓存已经没有用户关注的数​​据且与之关联的 DMA 和中断请求已处理完毕，在这种条件下，先禁用 SPI（SPIx_CR1.EN 清零），然后进行改写。

4. 置位 SPIx_CR1.EN 以使能 SPI

需要注意的是，上一步步骤 3 和本步骤 4 中对 SPIx_CR1 的配置，包括 SPIx_CR1.EN，可以在一次写操作中一起完成。

5. 配置 SPIx_CR2 寄存器

- 1) 如果用到 DMA，那么根据 DMA 数据传输的方向选择置位 SPIx_CR2.DMATXEN 和（或）SPIx_CR2.DMARXEN。
- 2) 同理，如果需要使能某些中断，那么根据需要使能的中断事件选择置位各个中断使能控制位。

注意：

- SPIx_CR2 包含了 SPI 的 DMA 请求和中断请求使能的寄存器，用户程序可以根据应用需求随时进行改写。

18.4.2 寄存器查询方式进行通信

全双工通信

1. 按上文步骤对 SPI 进行初始化配置；
2. 如果是主机，要将输出的从机选择拉低来选中对应的从机；如果是从机，则要保证已经被选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉低或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 0；
3. 循环查询发送缓冲空标志 SPIx_SR.TXE，直到其为 1 时，将待发送数据写入数据寄存器 SPIx_DR.DR；
4. 循环查询接收缓冲非空标志 SPIx_SR.RXNE 来判断是否完成该帧的传输并接收到数据，直到其为 1 时，可以从数据寄存器 SPIx_DR.DR 中读出接收数据；
5. 多帧数据传输，重复步骤 3 到步骤 4，直到全部数据收发结束；
6. （可选）如果是主机，可以将输出的从机选择拉高来取消选中；如果是从机，输入的从机选择可以拉高来取消选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉高或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 1；
7. 如果需要进行新的数据传输，重复步骤 1 至步骤 6；

半双工通信

1. 按上文步骤对 SPI 进行初始化配置；
2. 如果是主机，要将输出的从机选择拉低来选中对应的从机；如果是从机，则要保证已经被选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉低或

SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 0;

3. 根据收发方向修改半双工输出使能寄存器 SPIx_HDOE.HDOE: 发送则将 HDOE 置位, 接收则将 HDOE 清零;
4. 如果是发送, 则循环查询发送缓冲空标志 SPIx_SR.TXE, 直到其为 1 时, 将待发送数据写入数据寄存器 SPIx_DR.DR; 如果是接收, 则先循环查询发送缓冲空标志 SPIx_SR.TXE, 直到其为 1 时, 将虚拟数据 (dummy data) 写入数据寄存器 SPIx_DR.DR 启动传输, 然后循环查询接收缓冲非空标志 SPIx_SR.RXNE 来判断是否接收到数据, 直到其为 1 时, 可以从数据寄存器 SPIx_DR.DR 中读出接收数据;
5. 多帧数据传输, 重复步骤 3 到步骤 4, 直到全部数据收发结束;
6. (可选) 如果是主机, 可以将输出的从机选择拉高来取消选中; 如果是从机, 输入的从机选择可以拉高来取消选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉高或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 1;
7. 如果需要进行新的数据传输, 重复步骤 1 至步骤 6;

注意:

- SPI 工作在主机半双工模式时, 如果需要接收数据, 就必须向数据寄存器中写入虚拟数据 (dummy data) 来发起一次传输, 虚拟数据的数值不影响传输结果。因此, 即使是主机接收, 也需要向发送缓冲填充虚拟数据来启动传输, 以此实现数据的接收。

单工通信

1. 按上文步骤对 SPI 进行初始化配置;
2. 如果是主机, 要将输出的从机选择拉低来选中对应的从机; 如果是从机, 则要保证已经被选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉低或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 0;
3. 如果是发送, 则循环查询发送缓冲空标志 SPIx_SR.TXE, 直到其为 1 时, 将待发送数据写入数据寄存器 SPIx_DR.DR; 如果是接收, 则先循环查询发送缓冲空标志 SPIx_SR.TXE, 直到其为 1 时, 将虚拟数据 (dummy data) 写入数据寄存器 SPIx_DR.DR 启动传输, 然后循环查询接收缓冲非空标志 SPIx_SR.RXNE 来判断

是否接收到数据，直到其为 1 时，可以从数据寄存器 SPIx_DR.DR 中读出接收数据；

4. 多帧数据传输，重复步骤 3 到步骤 4，直到全部数据收发结束；
5. （可选）如果是主机，可以将输出的从机选择拉高来取消选中；如果是从机，输入的从机选择可以拉高来取消选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉高或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 1；
6. 如果需要进行新的数据传输，重复步骤 1 至步骤 5；

注意：

- SPI 工作在主机单工模式且只接收数据时，就必须向数据寄存器中写入虚拟数据（dummy data）来发起一次传输，虚拟数据的数值不影响传输结果。因此，即使是主机接收，也需要向发送缓冲填充虚拟数据来启动传输，以此实现数据的接收。

18.4.3 DMA 方式进行通信

全双工通信

1. 根据数据传输需求配置 DMA 控制器（参见 DMA 控制器章节）；
2. 使能 DMA 控制器（参见 DMA 控制器章节）；
3. 按上文步骤对 SPI 进行初始化配置，其中 DMA 请求使能暂时不要置位；
4. 如果是主机，要将输出的从机选择拉低来选中对应的从机；如果是从机，则要保证已经被选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉低或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 0；
5. 置位 DMA 请求使能（SPIx_CR2.DMATXEN 和 SPIx_CR2.DMARXEN）；
6. 数据传输开始，收发的数据由 DMA 直接进行搬运，无需软件参与；
7. 通过查询 DMA 的完成状态来判断数据是否传输完毕，也可以借助 DMA 控制器的完成中断来达成；
8. 数据传输结束，关闭 SPI 的 DMA 请求使能；
9. （可选）如果是主机，可以将输出的从机选择拉高来取消选中；如果是从机，输入的从机选择可以拉高来取消选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉高或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 1；

10. 如果需要进行新的数据传输，重复步骤 1 至步骤 9；

注意：

- SPI 支持的 DMA 传输类型为数据块大小为 1 的 DMA 块传输（block），每帧的数据宽度小于等于 8bit 时建议使用字节（byte）宽度，大于 8bit 时建议使用半字（half-word）宽度。

半双工通信

1. 根据数据传输需求配置 DMA 控制器（参见 DMA 控制器章节）；
 2. 使能 DMA 控制器（参见 DMA 控制器章节）；
 3. 按上文步骤对 SPI 进行初始化配置，其中 DMA 请求使能暂时不要置位；
 4. 如果是主机，要将输出的从机选择拉低来选中对应的从机；如果是从机，则要保证已经被选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉低或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 0；
 5. 根据收发方向修改半双工输出使能寄存器 SPIx_HDOE.HDOE：发送则将 HDOE 置位，接收则将 HDOE 清零；
 6. 置位 DMA 请求使能：主机发送/从机发送时要置位 SPIx_CR2.DMATXEN；从机接收时需要置位 SPIx_CR2.DMARXEN；主机接收时要置位 SPIx_CR2.DMATXEN 和 SPIx_CR2.DMARXEN；
 7. 数据传输开始，收发的数据由 DMA 直接进行搬运，无需软件参与；
 8. 通过查询 DMA 的完成状态来判断数据是否传输完毕，也可以借助 DMA 控制器的完成中断来达成；
 9. 数据传输结束，关闭 SPI 的 DMA 请求使能；
 10. （可选）如果是主机，可以将输出的从机选择拉高来取消选中；如果是从机，输入的从机选择可以拉高来取消选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉高或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 1；
 11. 如果需要进行新的数据传输，重复步骤 1 至步骤 10；
- 注意：
- SPI 支持的 DMA 传输类型为数据块大小为 1 的 DMA 块传输（block），每帧的数据宽度小于等于 8bit 时建议使用字节（byte）宽度，大于 8bit 时建议使用半字（half-word）宽度。

- SPI 工作在主机半双工模式时，如果需要接收数据，就必须向数据寄存器中写入虚拟数据（dummy data）来发起一次传输，虚拟数据的数值不影响传输结果。因此，即使是主机接收，也需要将发送 DMA 使能，向发送缓冲填充虚拟数据来启动传输，以此实现数据的接收。
- 由于半双工模式的收发切换需要通过额外的寄存器改写来实现，所以此模式下，发送接收穿插交错在单次 DMA 任务中无法实现，而要通过多次不同方向的 DMA 任务来实现。

单工通信

1. 根据数据传输需求配置 DMA 控制器（参见 DMA 控制器章节）；
2. 使能 DMA 控制器（参见 DMA 控制器章节）；
3. 按上文步骤对 SPI 进行初始化配置，其中 DMA 请求使能暂时不要置位；
4. 如果是主机，要将输出的从机选择拉低来选中对应的从机；如果是从机，则要保证已经被选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉低或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 0；
5. 置位 DMA 请求使能：主机发送/从机发送时要置位 SPIx_CR2.DMATXEN；从机接收时需要置位 SPIx_CR2.DMARXEN；主机接收时要置位 SPIx_CR2.DMATXEN 和 SPIx_CR2.DMARXEN；
6. 数据传输开始，收发的数据由 DMA 直接进行搬运，无需软件参与；
7. 通过查询 DMA 的完成状态来判断数据是否传输完毕，也可以借助 DMA 控制器的完成中断来达成；
8. 数据传输结束，关闭 SPI 的 DMA 请求使能；
9. （可选）如果是主机，可以将输出的从机选择拉高来取消选中；如果是从机，输入的从机选择可以拉高来取消选中——SPIx_CR1.SSM=0 时输入的从机选择引脚拉高或 SPIx_CR1.SSM=1 时内部从机选择寄存器 SPIx_SSI.SSI 写 1；
10. 如果需要进行新的数据传输，重复步骤 1 至步骤 9；

注意：

- SPI 支持的 DMA 传输类型为数据块大小为 1 的 DMA 块传输（block），每帧的数据宽度小于等于 8bit 时建议使用字节（byte）宽度，大于 8bit 时建议使用半字（half-word）宽度。

- SPI 工作在主机单工模式且只接收数据时，就必须向数据寄存器中写入虚拟数据（dummy data）来发起一次传输，虚拟数据的数值不影响传输结果。因此，此种情况下也需要将发送 DMA 使能，向发送缓冲填充虚拟数据来启动传输，以此实现数据的接收。

18.5 SPI 常见使用场景

18.5.1 单主机单从机通信

全双工通信

全双工通信的方式，允许主从机依托两根单向的数据线进行同时收发的双向的通信。

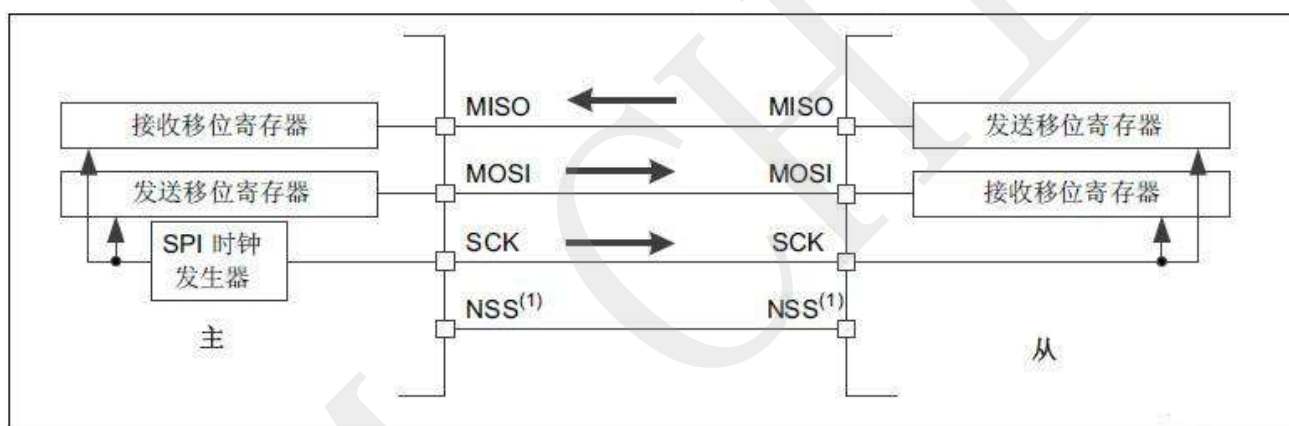


图 18-6 SPI 全双工通信

半双工通信

半双工通信的方式，允许主从机依托一根双向的数据线进行分时收发的双向的通信。作为主机时使用的数据线是 MOSI，作为从机时使用的数据线是 MISO。未用到的引脚不参与 SPI 通信，可供其他功能使用。

当主从机之间进行半双工通信时，由于收发的切换不同步，它们对于双向的共用数据线可能出现驱动冲突。为避免此类情况对器件可能产生的损害，建议在数据线上串入限流电阻以保护芯片引脚。

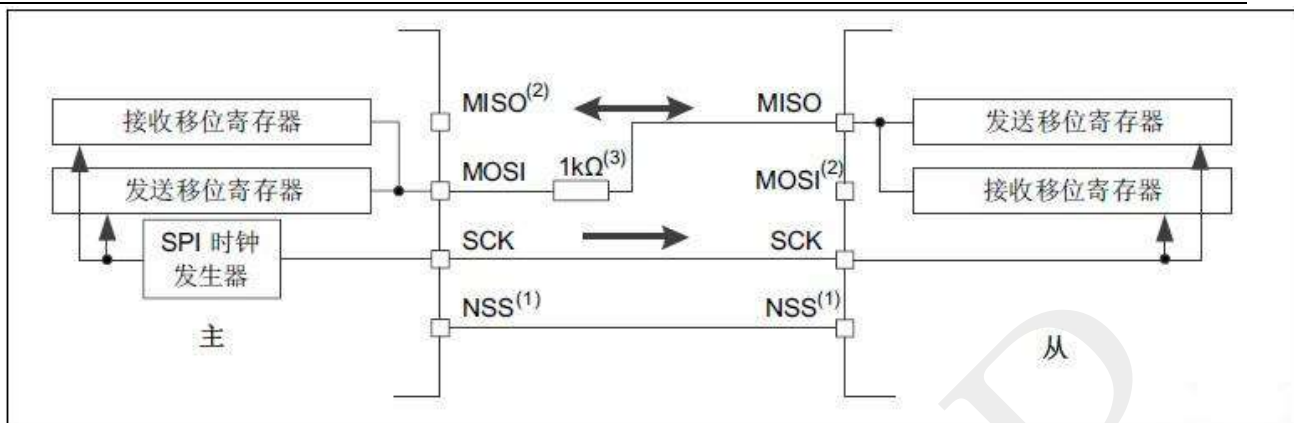
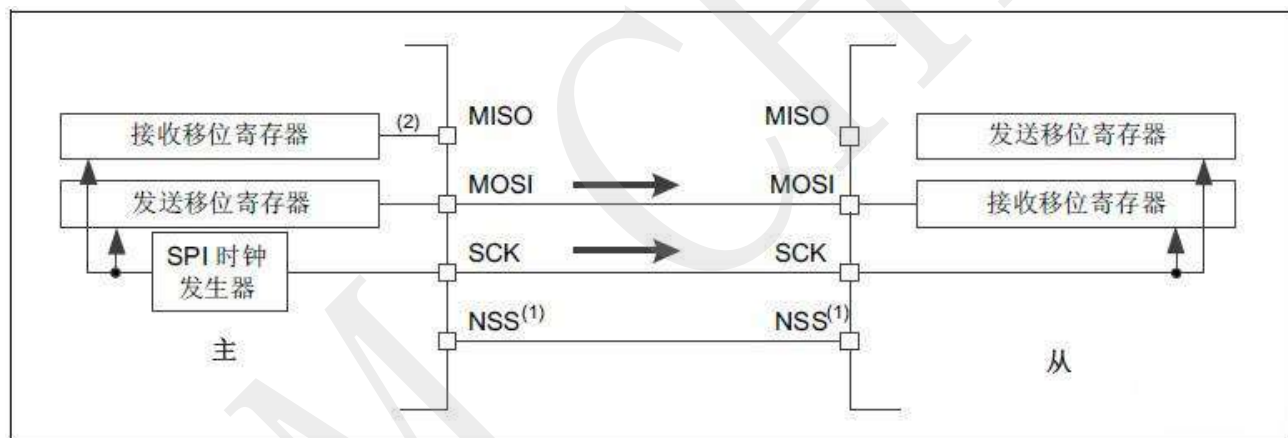


图 18-7 SPI 半双工通信

单工通信

单工通信的方式，允许主从机依托一根单向的数据线进行固定接收或固定发送的单向的通信。未用到的引脚不参与 SPI 通信，可供其他功能使用。



1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见[第 14.3.5 节，从机选择（NSS）引脚的配置](#)。
2. 在发送器 Rx 移位寄存器的输入上捕获意外输入信息。标准只发送模式下必须忽略与发送器接收流相关的所有事件（例如 OVF 标志）。
3. 在这种配置下，两个 MISO 引脚均可用作 GPIO。

图 18-8 SPI 单工通信

18.5.2 多机通信

多机的系统中，主机的从机选择 NSS 引脚配置为硬件输入（SPIx_CR1.SSM=0），由 GPIO 负责输出从机选择信号，且主机的 NSS 输入都由另一台主机的 GPIO 驱动。主机在尝试获取 SPI 总线控制权时，能结合模式错误 MODF 来检测总线是否发生冲突。

多个从机的系统中，每台从机共享串行时钟和数据线，但是需要各自独立的从机选择信号，这就要求主机通过不同的 GPIO 向每个从机输出对应的从机选择信号。

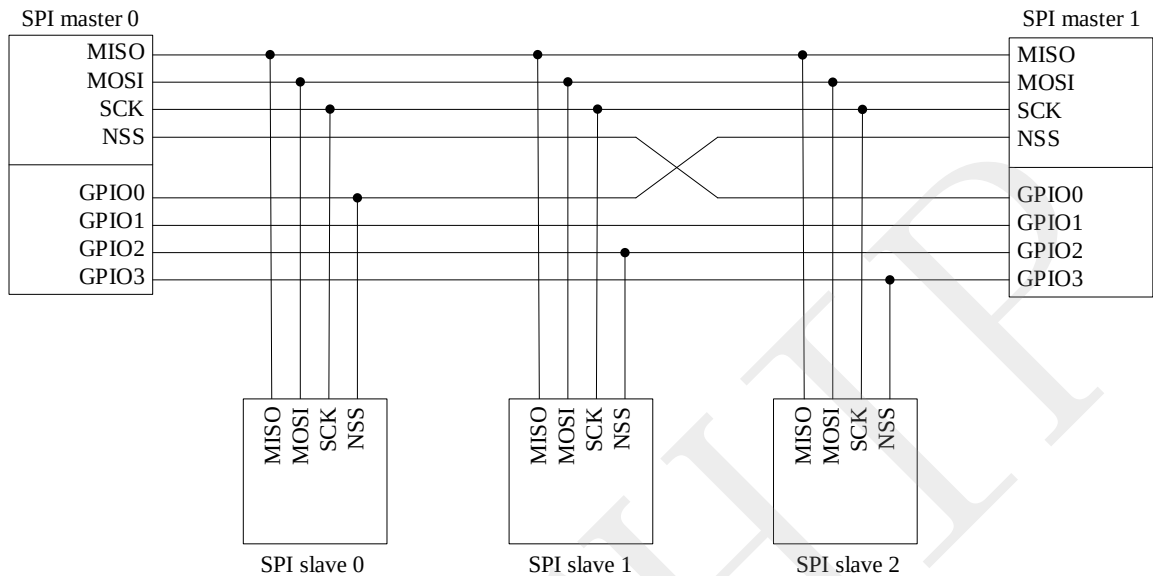


图 18-9 SPI 多机系统

18.6 SPI 寄存器描述

SPI1 基地址： 0x4001_3000

SPI2 基地址： 0x4000_3800

SPI3 基地址： 0x4000_3C00

SPI4 基地址： 0x4000_4000

寄存器名称	偏移量	访问	寄存器描述
SPIx_CR1	0x00	RW	SPIx 控制寄存器 1
SPIx_CR2	0x04	RW	SPIx 控制寄存器 2
SPIx_HDOE	0x08	RW	SPIx 半双工输出使能寄存器
SPIx_SSI	0x0C	RW	SPIx 内部从机选择寄存器
SPIx_SR	0x10	RO	SPIx 状态寄存器
SPIx_ICR	0x14	R1W0	SPIx 中断清除寄存器
SPIx_DR	0x18	RW	SPIx 数据寄存器

表 18-2 SPI 寄存器列表

18.6.1 SPI 控制寄存器 1(SPIx_CR1)

地址偏移量: 0x00

复位值: 0x00000700

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
RSV	CM			SSM	WIDTH										
	RW			RW	RW										
7	6	5	4	3	2	1	0								
LSBF	MSTR	CPOL	CPHA	BRR			EN								
RW	RW	RW	RW	RW			RW								

位	标记	功能描述
31:15	Reserved	保留
14:13	CM	通信方式选择 00: 全双工双向 01: 单工只发 10: 单工只收 11: 单线半双工
12	SSM	从机选择配置（主/从机模式下效果不同） 1: （主）SSI 寄存器决定从机选择输出值/（从）SSI 寄存器决定从机选择输入值 0: （主）管脚决定从机选择输出值/（从）管脚决定从机选择输入值
11:8	WIDTH	每帧的数据宽度 0000/0001/0010 禁用，写入禁用的值会使寄存器的值被改为默认值 0111 0011: 4bit 0100: 5bit 0101: 6bit ... 1110: 15bit 1111: 16bit
7	LSBF	高低位顺序选择 1: 最低有效位 LSB 收发在前 0: 最高有效位 MSB 收发在前
6	MSTR	主机/从机模式选择 1: 主机模式 0: 从机模式

5	CPOL	串行时钟极性选择 1: 待机时高电平 0: 待机时低电平
4	CPHA	串行时钟相位选择 1: 前边沿移位/后边沿采样 0: 前边沿采样/后边沿移位
3:1	BRR	主机模式波特率 000: fpc1k/2 001: fpc1k/4 010: fpc1k/8 011: fpc1k/16 ... 110: fpc1k/128 111: 保留（不要使用）
0	EN	模块使能 1: 使能 0: 禁用

18.6.2 SPI 控制寄存器 2(SPIx_CR2)

地址偏移量: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved						DMATXEN	DMARXEN								
						RW	RW								
7	6	5	4	3	2	1	0								
MODFIE	SSERRIE	OVFIE	UDFIE	SSRIE	SSFIE	RXNEIE	TXEIE								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:10	Reserved	保留
9	DMATXEN	发送 DMA 使能 1: 使能 0: 禁用
8	DMARXEN	接收 DMA 使能 1: 使能

		0: 禁用
7	MODFIE	模式错误中断使能 1: 使能 0: 禁用
6	SSERRIE	从机模式下的从机选择错误中断使能 1: 使能 0: 禁用
5	OVFIE	接收缓冲上溢错误中断使能 1: 使能 0: 禁用
4	UDFIE	从机模式下发送缓冲下溢错误中断使能 1: 使能 0: 禁用
3	SSRIE	从机选择输入上升沿中断使能 1: 使能 0: 禁用
2	SSFIE	从机选择输入下降沿中断使能 1: 使能 0: 禁用
1	RXNEIE	接收缓冲非空中断使能 1: 使能 0: 禁用
0	TXEIE	发送缓冲空中断使能 1: 使能 0: 禁用

18.6.3 SPI 半双工输出使能寄存器(SPIx_HDOE)

地址偏移量: 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														HDOE	
														RW	

位	标记	功能描述
31:1	Reserved	保留

0	HDOE	半双工通信时的收发切换寄存器，仅在半双工通信时有效 0：只收 1：只发

18.6.4 SPI 内部从机选择寄存器(SPIx_SSI)

地址偏移量：0x0C

复位值：0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SSI	
															RW

位	标记	功能描述
31:1	Reserved	保留
0	SSI	内部从机选择，当 SSM 寄存器为 1 时有效（主机模式下寄存器决定从机选择输出值/从机模式下寄存器决定从机选择输入值）

18.6.5 SPI 状态寄存器(SPIx_SR)

地址偏移量：0x10

复位值：0x0000 0201

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved										SSLVL		BUSY			
										RO		RO			
7	6	5	4	3	2	1	0								
MODF	SSERR	OVF	UDF	SSR	SSF	RXNE	TXE								
RO	RO	RO	RO	RO	RO	RO	RO								

位	标记	功能描述
31:10	Reserved	保留

9	SSLVL	从机选择输入的状态 1: 从机选择输入是高电平 0: 从机选择输入是低电平 注意: 当主机模式且 SSM=1 时, 没有从机选择输入, 本 bit 无意义。
8	BUSY	总线忙标志位 1: 总线忙 0: 总线空闲
7	MODF	模式错误标志位 (主机模式下从机选择输入为低则触发 MODF) 1: 出错 0: 正常
6	SSERR	从机模式下的从机选择错误标志位 1: 出错 0: 正常
5	OVF	接收缓冲上溢错误标志位 1: 出错 0: 正常
4	UDF	从机模式下的发送缓冲下溢错误标志位 1: 出错 0: 正常
3	SSR	从机选择输入上升沿标志位 1: 出现了上升沿 0: 未出现上升沿 注意: 当主机模式且 SSM=1 时, 没有从机选择输入, 本 bit 无意义。
2	SSF	从机选择输入下降沿标志位 1: 出现了下降沿 0: 未出现下降沿 注意: 当主机模式且 SSM=1 时, 没有从机选择输入, 本 bit 无意义。
1	RXNE	接收缓冲非空标志位 1: 接收缓冲非空 0: 接收缓冲空
0	TXE	发送缓冲空标志位 1: 发送缓冲空 0: 发送缓冲非空

18.6.6 SPI 中断清除寄存器(SPIx_ICR)

地址偏移量: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
MODFC		SSERRC		OVFC		UDFC		SSRC		SSFC		RXNEC		TXES									
R1W0		R1W0		R1W0		R1W0		R1W0		R1W0		R1W0		R1W0									

位	标记	功能描述
31:8	Reserved	保留
7	MODFC	模式错误清除（读出总为 1，写 0 清除 MODF，写 1 无效果）
6	SSERRC	从机选择错误清除（读出总为 1，写 0 清除 SSERR，写 1 无效果）
5	OVFC	接收缓冲上溢错误清除（读出总为 1，写 0 清除 OVF，写 1 无效果）
4	UDFC	从机模式下的发送缓冲下溢错误清除（读出总为 1，写 0 清除 UDF，写 1 无效果）
3	SSRC	从机选择输入上升沿标志清除（读出总为 1，写 0 清除 SSR，写 1 无效果）
2	SSFC	从机选择输入下降沿标志清除（读出总为 1，写 0 清除 SSF，写 1 无效果）
1	RXNEC	接收缓冲非空标志清除（读出总为 1，写 0 清除 RXNE，写 1 无效果）
0	TXES	发送缓冲空标志置位（读出总为 1，写 0 置位 TXE，写 1 无效果） 注意：本 bit 的功能是让软件能根据需要置位 TXE（使 TXE=1，即忽略当前发送缓冲的内容，将发送缓冲的状态改为空），触发 TXE 中断，而不是清除 TXE 中断。这与本寄存器内其他 bit 不同，在使用时需特别注意。

18.6.7 SPI 数据寄存器(SPIx_DR)

地址偏移量：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	DR	数据寄存器（将待发送数据写入本寄存器，或从本寄存器读出接收到的数据） 注意：当每帧的数据宽度不足 16bit 时，数据寄存器存放的是被零扩展到 16bit 的接收数据，用户程序读取后只需关心有效位即可。

19. 音频接口（I2S）

I2S 特性

- 支持 Philip/ MSB/LSB /PCM 模式
- 支持 MCK 输出
- 支持 48K,44.1k,32k,16k,8k 等不同的音频采样率
- 支持 数据长度 16,24,32
- 支持帧长度 16/32
- 支持 DMA 数据传输
- 支持 master 发送，master 接收
- 支持 slave 发送，slave 接收

19.1 功能说明

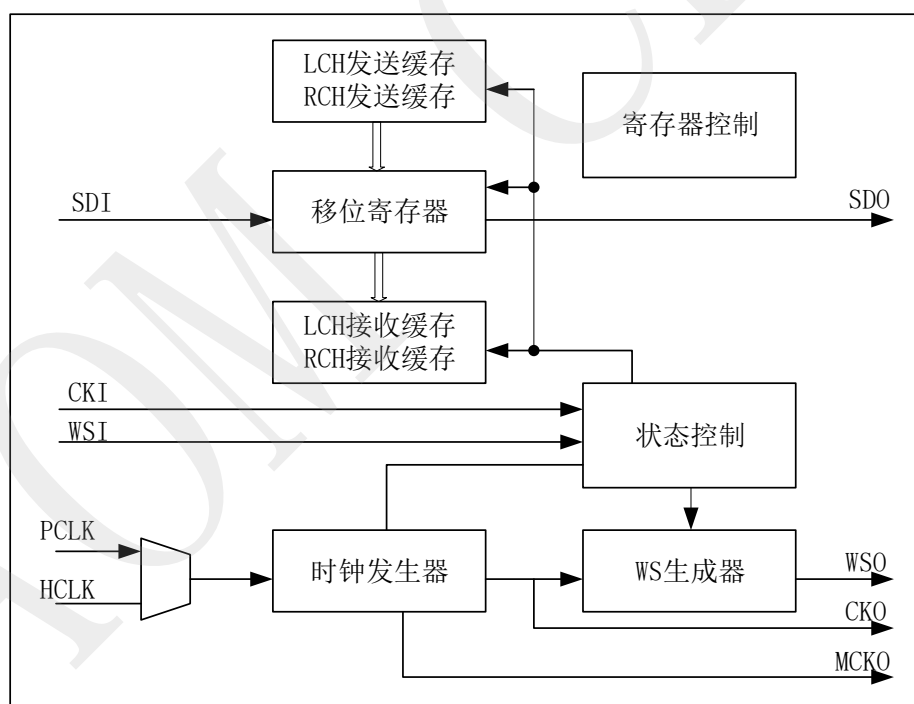


图 19-1 功能框图

19.2 I2S 音频协议说明

I2S 需要处理通常在左右两个通道上时分复用的音频数据。但是，有两个 16 位寄存

器进行发送或接收。所以，需由软件将与相应通道对应的值写入通道数据寄存器，以及从相应通道数据寄存器中读取相应数据。始终先发送左通道数据，而后再发送右通道数据（对于 PCM 协议来说，右通道数据没有意义）。

数据和帧格式组合有四种，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中
- 将 16 位数据封装在 32 位帧中
- 将 24 位数据封装在 32 位帧中
- 将 32 位数据封装在 32 位帧中

当使用 32 位数据包中的 16 位数据时，只需要读写 16 位有效数据，16 位无效数据被强制清零，无需任何软件操作或 DMA 请求（只需一个读/写操作）。

24 位和 32 位数据帧需要对 I2S_DR 寄存器执行两次 CPU 读取或写入操作，或者需要两次 DMA 操作。对于 24 位数据帧，硬件会在自动填充 8 个 0 扩展到 32 位。对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 优先）。

I2S 接口支持四种音频标准，可使用 I2S_I2SCFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

19.2.1 I2S Philips 标准

使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位(MSB) 之前的一个时钟开始有效。

发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

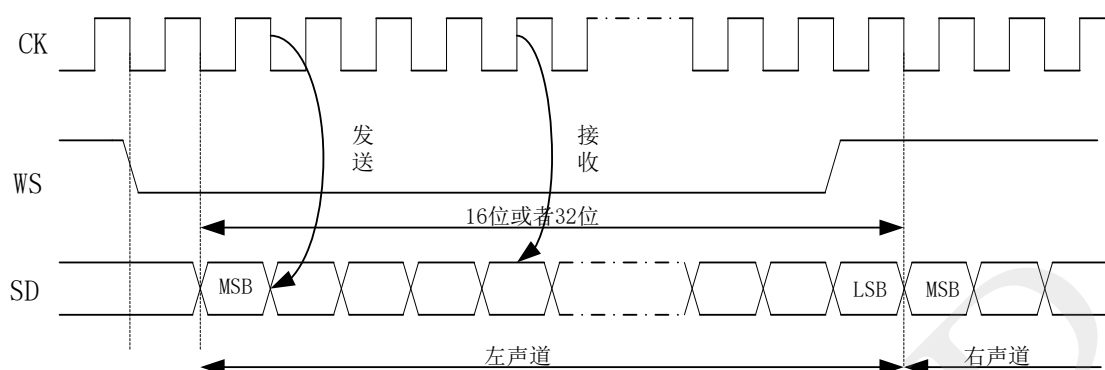


图 19-2 Philip协议波形（16/32位全精度）

该模式需要对 I2S_DRL/ I2S_DRR 寄存器执行两次写入或读取操作。

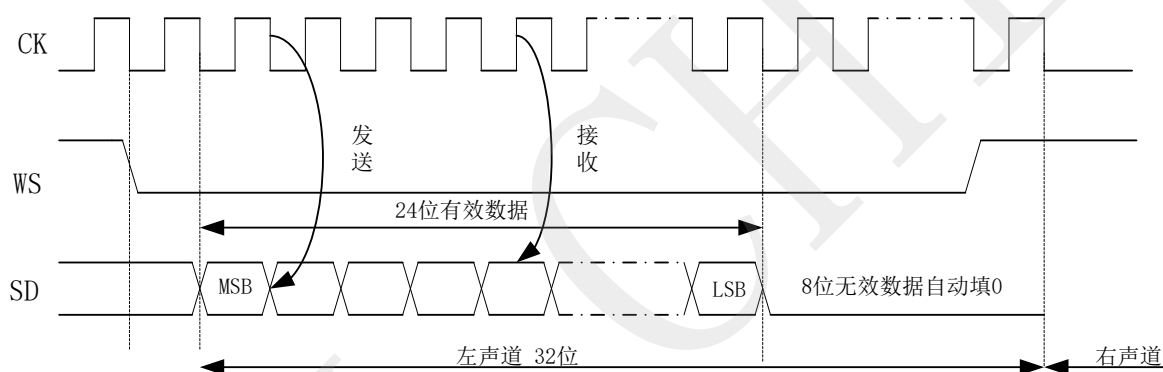


图 19-3 Philip 协议波形（24 位帧）

该模式需要对 I2S_DRL/ I2S_DRR 寄存器执行两次写入或读取操作。

在发送模式下，如果需要发送 0x89ABCD,0x8FEDCA（24 位）：

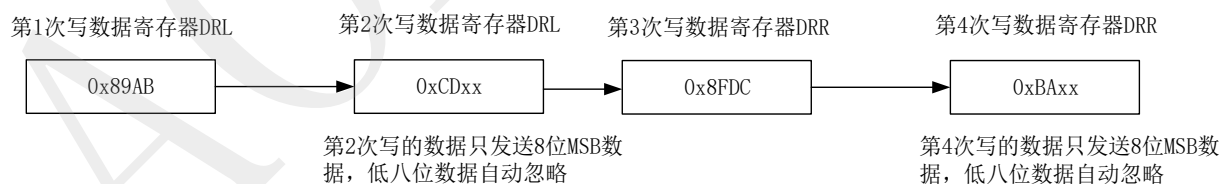


图 19-4 24 位数据发送写操作

在接收模式下，如果接收数据 0x123456,0x89ABCD：

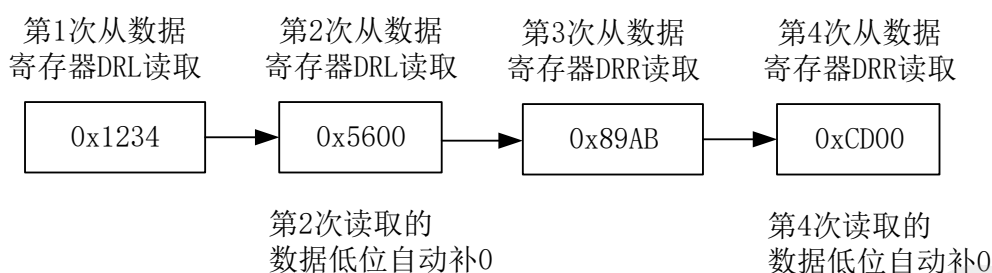


图 19-5 24 位数据模式接收读操作

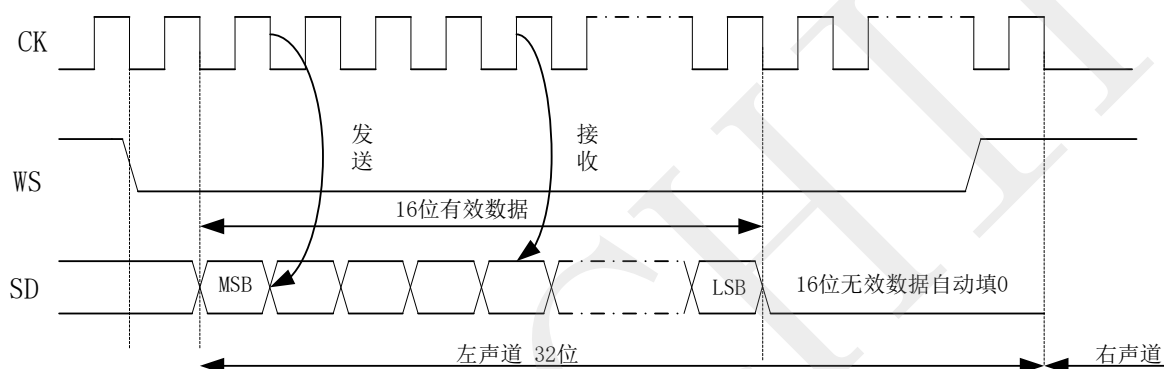


图 19-6 Philip 协议 16 位扩展到 32 位数据帧

如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 I2S_DR 寄存器。另外 16 位将有硬件自动填 0，不需要软件参与

如果要发送的数据或已接收的数据为 0xCDEF (0xCDEF0000 扩展为 32 位)，则需要执行下图中显示的操作。

只需要一次数据寄存器DRL/DRR读写

0xCDEF

图 19-7 16 位数据读写操作

发送时，每次将 MSB 写入数据寄存器 I2S_DRL/R，数据转移到移位寄存器后，相应通道的 TXE 标志就会置 1，并在中断使能的情况下触发中断，以将要发送的新数据加载到 I2S_DR 寄存器。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

接收时，接收到第一个半字（高 16 位），则硬件将相应通道的 RXNE 标志置 1，

并在中断使能的情况下触发中断。

19.2.2 MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位，WS 左声道对应高电平，右声道对应高电平。

发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

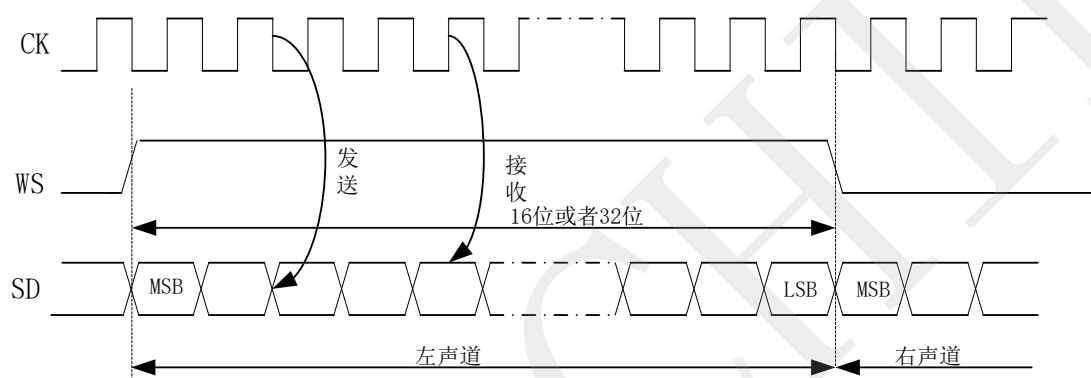


图 19-8 MSB 协议波形 (16/32 位全精度)

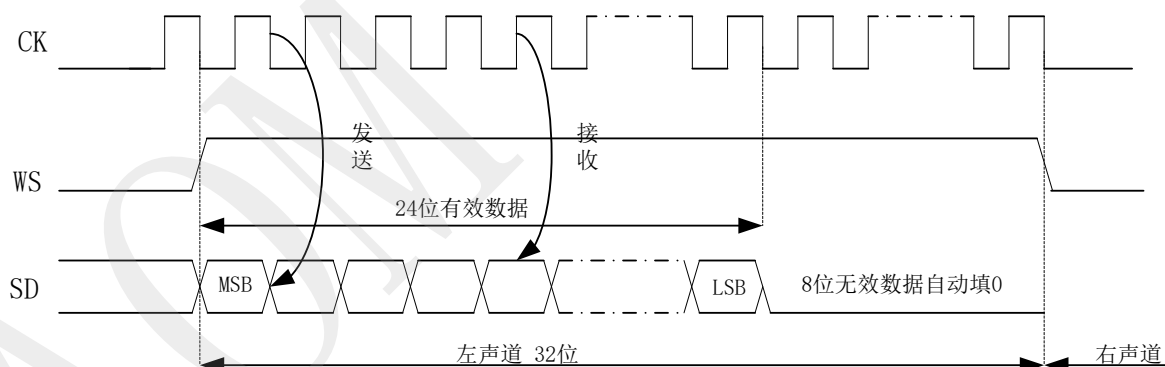


图 19-9 MSB 协议波形 (24 位帧)

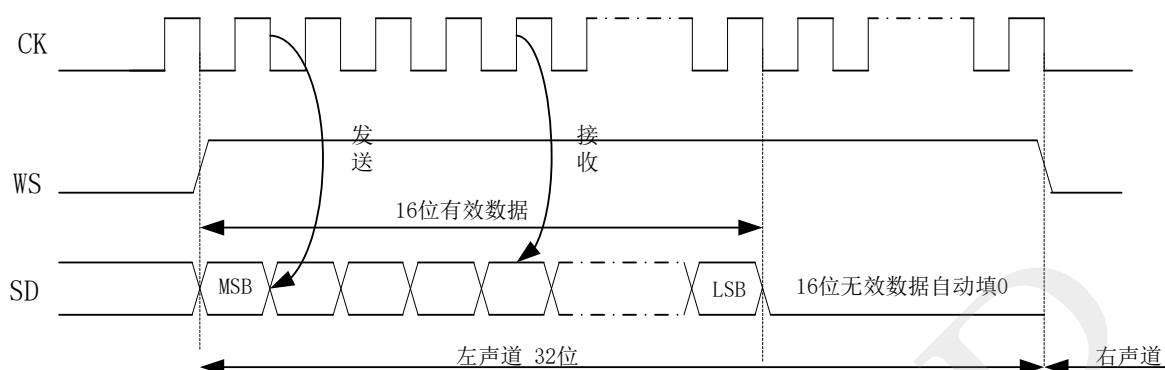


图 19-10 MSB 协议 16 位扩展到 32 位数据帧

19.2.3 LSB 对齐标准

该标准与 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

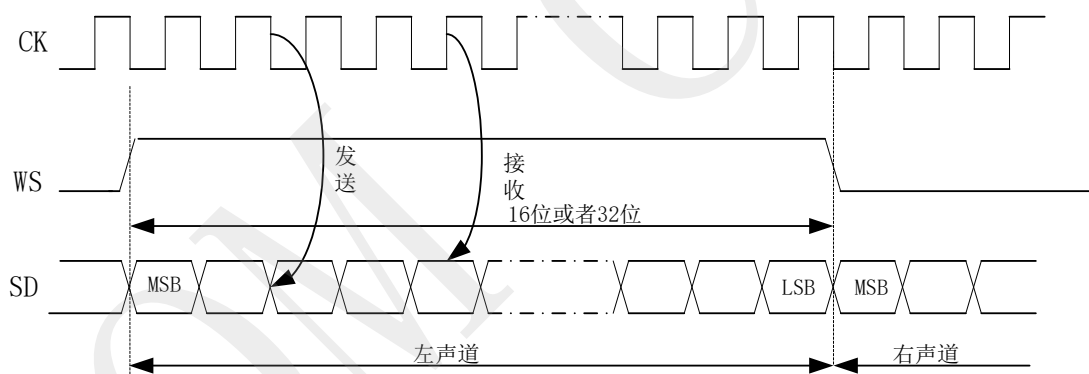


图 19-11 LSB协议波形（16/32位全精度）

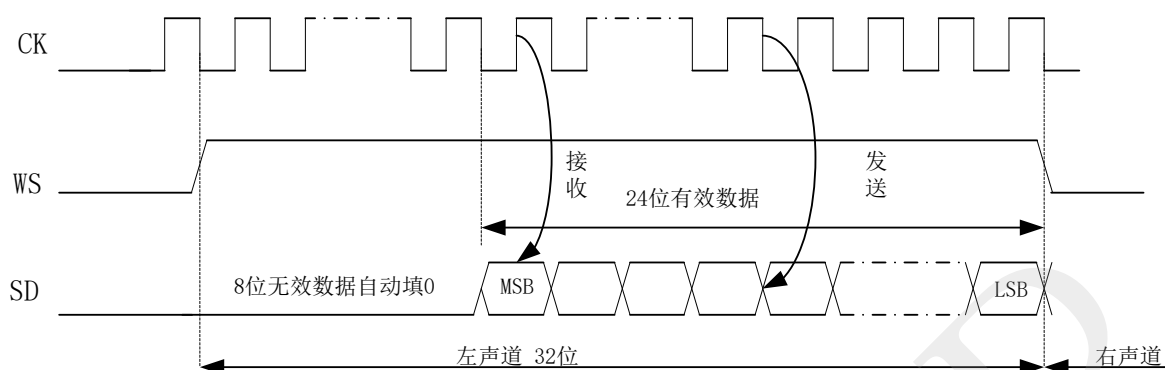


图 19-12 LSB 协议波形（24 位帧）

在发送模式下，如果需要发送数据 0x89ABCD，则需要通过软件或 DMA 对 I2S_DRL/ I2S_DRR 寄存器执行两次写入操作。下面给出了这些操作。

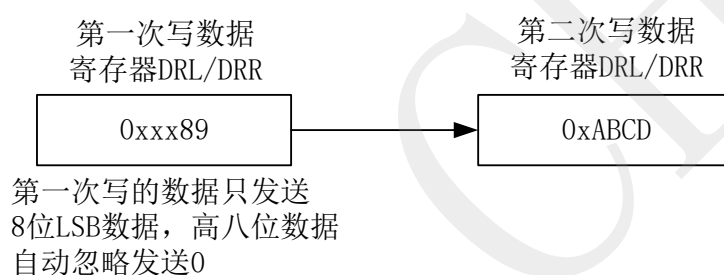


图 19-13 LSB 24 位数据发送写操作

在接收模式下，如果接收到数据 0x123456，则在每个 RXNE 事件时需要对 I2S_DR 寄存器执行两次连续的读取操作。

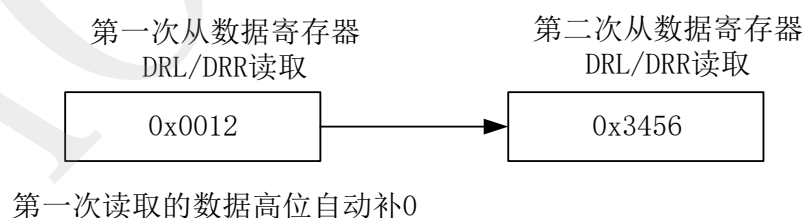


图 19-14 LSB 24 位数据模式接收读操作

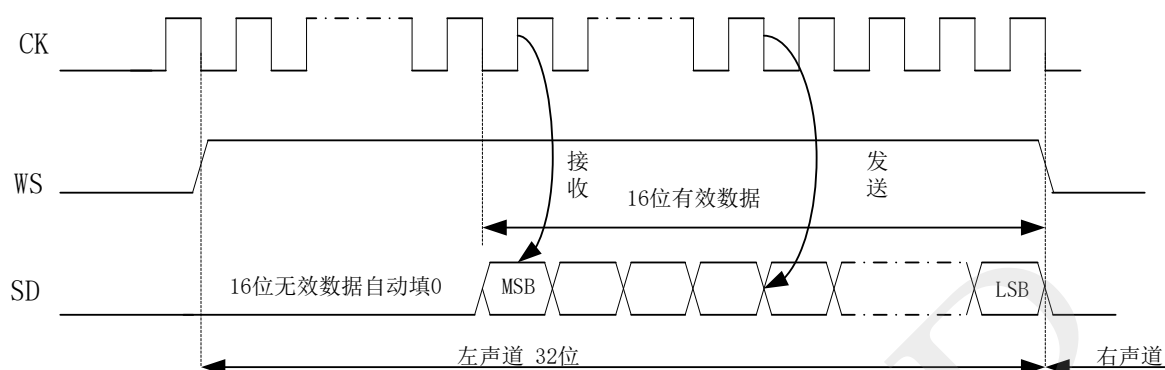


图 19-15 LSB 协议 16 位扩展到 32 位数据帧

如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 I2S_DRL/I2S_DRR 寄存器。扩展到 32 位中，高半字（16 位 MSB）被硬件置为 0x0000。

如果要发送的数据或已接收的数据为 0x7893（0x0000 7893 是扩展 32 位的数据），则需要执行下中显示的操作。

只需要一次数据
寄存器DRL/DRR
读写

0x7893

图 19-16 16 位数据读写操作

在发送模式下，相应通道发生 TXE 事件时，应用程序需要在相应通道写入要发送的数据（此例中，为 0x7893）。

首先发送 0x0000 字段（扩展到 32 位）。相应通道的有效数据 (0x7893) 转移到移位寄存器后，相应通道的 TXE 标志会被再次置 1。

在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位相应通道的 RXNE。

19.2.4 PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 I2S_I2SCFGR 寄存器中的 PCMSYNC 位来配置。PCM 发送数据使用上升沿，

接收数据使用下降沿。

发送数据与接收数据只需要使用 I2S_DRL 数据寄存器，I2S_DRR 数据寄存器在这种模式下无效。

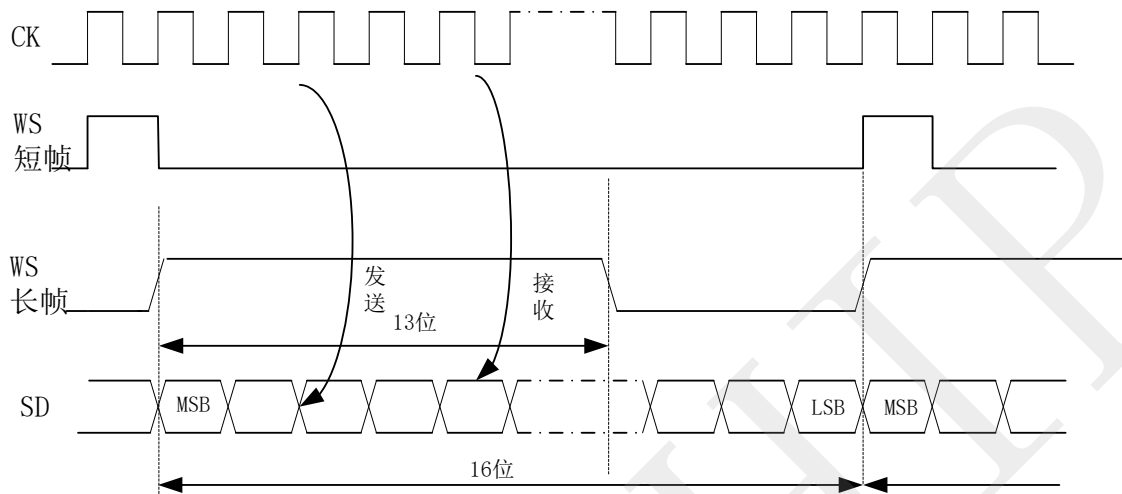


图 19-17 PCM 协议波形（16 位）

对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

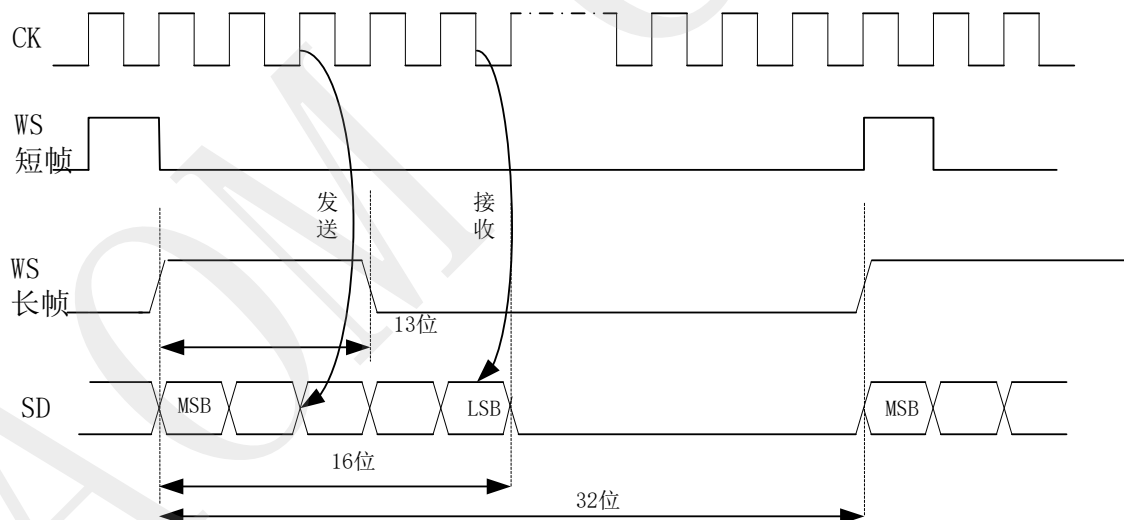


图 19-18 PCM 协议波形（16 位数据扩展到 32 位）

注：对于两种模式（主/从模式）和两种同步（短/长同步），即使在从模式下，也需要指定两组连续数据（以及两个同步信号）之间位的个数（I2S_I2SCFGR 寄存器中的 DATLEN 位和 CHLEN 位）。

19.3 I2S 时钟发生器

I2S 比特率用来确定 I2S 数据线上的数据流和 I2S 时钟信号频率。

I2S 比特率 = 每个通道的位数 × 通道数 × 音频采样频率

对于 16 位双通道音频，I2S 比特率的计算公式如下：

I2S 比特率 = $16 \times 2 \times f_s$

如果数据包为 32 位宽，则 I2S 比特率 = $32 \times 2 \times f_s$ 。

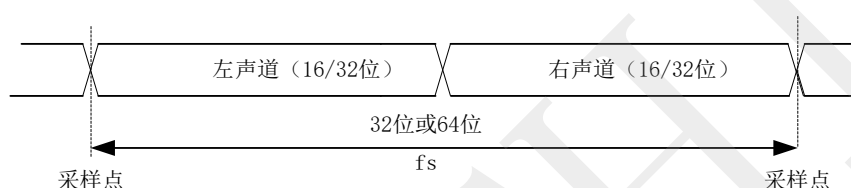


图 19-19 音频采样频率定义

配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

下图展示了通信时钟架构。I2S 时钟可选择 PCLK 或者 HCLK。

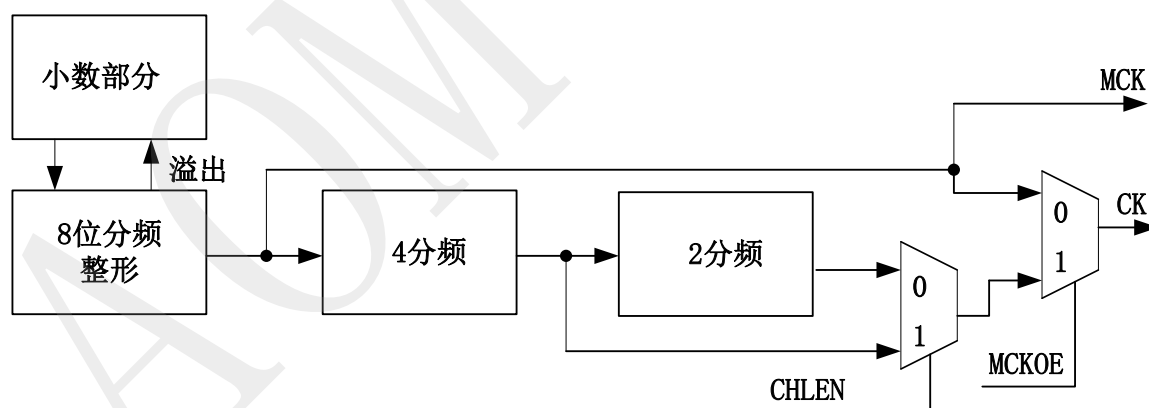


图 19-20 时钟产生

音频采样频率可以是 96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz 或 8 kHz（或此范围内的任何其它值）。为达到所需频率，需要根据以下公式对线性分频器进行编程：

小数分频 FRACT 等于 0x00 时：

输出主时钟（I2S_I2SPR 寄存器中的 MCKOE 置 1）时：

$$f_s = I2SCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8] \quad (\text{通道帧宽度为 16 位时})$$

$$f_s = I2SCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4] \quad (\text{通道帧宽度为 32 位时})$$

禁止主时钟输出（MCKOE 位清零）时：

$$f_s = I2SCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)] \quad (\text{通道帧宽度为 16 位时})$$

$$f_s = I2SCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)] \quad (\text{通道帧宽度为 32 位时})$$

小数分频 FRACT 不等于 0x00 时：

输出主时钟（I2S_I2SPR 寄存器中的 MCKOE 置 1）时：

$$f_s = I2SCLK / \{ (16 \times 2) \times [2 \times (I2SDIV + FRACT/64)] \times 8 \} \quad (\text{通道帧宽度为 16 位时})$$

$$f_s = I2SCLK / \{ [(32 \times 2) \times [2 \times (I2SDIV + FRACT/64)] \times 4 \} \quad (\text{通道帧宽度为 32 位时})$$

禁止主时钟输出（MCKOE 位清零）时：

$$f_s = I2SCLK / \{ (16 \times 2) \times [2 \times (I2SDIV + FRACT/64)] \} \quad (\text{通道帧宽度为 16 位时})$$

$$f_s = I2SCLK / \{ (32 \times 2) \times [2 \times (I2SDIV + FRACT/64)] \} \quad (\text{通道帧宽度为 32 位时})$$

下表提供了时钟 48M 时钟针对不同采样率的设置及精度值（不使用小数分频）。

I2S CLK	通道 长度	I2SDIV	I2SODD	MCLK	采样率	实际 频率	误差%	freq(mck)
48000000	16	8	0	无输出	96000	93750	-2.34	0
48000000	32	4	0	无输出	96000	93750	-2.34	0
48000000	16	15	1	无输出	48000	48387	0.81	0
48000000	32	8	0	无输出	48000	46875	-2.34	0
48000000	16	17	0	无输出	44100	44118	0.04	0
48000000	32	8	1	无输出	44100	44118	0.04	0
48000000	16	23	1	无输出	32000	31915	-0.27	0
48000000	32	11	1	无输出	32000	32609	1.9	0
48000000	16	34	0	无输出	22050	22059	0.04	0
48000000	32	17	0	无输出	22050	22059	0.04	0
48000000	16	47	0	无输出	16000	15957	-0.27	0
48000000	32	23	1	无输出	16000	15957	-0.27	0
48000000	16	68	0	无输出	11025	11029	0.04	0
48000000	32	34	0	无输出	11025	11029	0.04	0
48000000	16	94	0	无输出	8000	7979	-0.26	0
48000000	32	47	0	无输出	8000	7979	-0.26	0
48000000	16	2	0	输出	48000	46875	-2.34	12000000
48000000	32	2	0	输出	48000	46875	-2.34	12000000
48000000	16	2	0	输出	44100	46875	6.29	12000000
48000000	32	2	0	输出	44100	46875	6.29	12000000
48000000	16	3	0	输出	32000	31250	-2.34	8000000
48000000	32	3	0	输出	32000	31250	-2.34	8000000
48000000	16	4	1	输出	22050	20833	-5.52	5333248
48000000	32	4	1	输出	22050	20833	-5.52	5333248
48000000	16	6	0	输出	16000	15625	-2.34	4000000
48000000	32	6	0	输出	16000	15625	-2.34	4000000
48000000	16	8	1	输出	11025	11029	0.04	2823424
48000000	32	8	1	输出	11025	11029	0.04	2823424
48000000	16	11	1	输出	8000	8152	1.9	2086912
48000000	32	11	1	输出	8000	8152	1.9	2086912

下表提供了时钟 48M 时钟使用小数分频针对不同采样率的设置及精度值。

I2S CLK	通道长度	I2SDIV	FRACT	MCLK	采样率	实际频率	误差%	freq(mck)
48000000	16	7	52	无输出	96000	96000	0	0
48000000	32	3	58	无输出	96000	96000	0	0
48000000	16	15	40	无输出	48000	48000	0	0
48000000	32	7	52	无输出	48000	48000	0	0
48000000	16	17	0	无输出	44100	44118	0.04	0
48000000	32	8	32	无输出	44100	44118	0.04	0
48000000	16	23	28	无输出	32000	32000	0	0
48000000	32	11	46	无输出	32000	32000	0	0
48000000	16	34	1	无输出	22050	22049	0	0
48000000	32	17	0	无输出	22050	22059	0.04	0
48000000	16	46	56	无输出	16000	16000	0	0
48000000	32	23	28	无输出	16000	16000	0	0
48000000	16	68	2	无输出	11025	11024	-0.01	0
48000000	32	34	1	无输出	11025	11024	-0.01	0
48000000	16	93	48	无输出	8000	8000	0	0
48000000	32	46	56	无输出	8000	8000	0	0
48000000	16	1	61	输出	48000	48000	0	12288000
48000000	32	1	61	输出	48000	48000	0	12288000
48000000	16	2	8	输出	44100	44118	0.04	11294208
48000000	32	2	8	输出	44100	44118	0.04	11294208
48000000	16	2	60	输出	32000	31915	-0.27	8170240
48000000	32	2	60	输出	32000	31915	-0.27	8170240
48000000	16	4	16	输出	22050	22059	0.04	5647104
48000000	32	4	16	输出	22050	22059	0.04	5647104
48000000	16	5	55	输出	16000	16000	0	4096000
48000000	32	5	55	输出	16000	16000	0	4096000
48000000	16	8	32	输出	11025	11029	0.04	2823424
48000000	32	8	32	输出	11025	11029	0.04	2823424
48000000	16	11	46	输出	8000	8000	0	2048000
48000000	32	11	46	输出	8000	8000	0	2048000

19.4 I2S 工作模式

19.4.1 I2S 主模式

I2S 可配置为主模式。这意味着将在 CK 引脚输出串行时钟，在 WS 引脚生成字选信号。主时钟 (MCK) 可以输出，也可以不输出，具体由 I2S_I2SPR 寄存器中的 MCKOE 位控制。

步骤

1. 设置 I2S_I2SPR 寄存器的 I2SDIV[7:0] 位，以定义串行时钟波特率，从而达到相应的音频采样频率。I2S_I2SPR 寄存器的 ODD 与 FRACT 位根据需要进行设置。
2. 如果需要为外部 DAC/ADC 音频组件提供主时钟 MCK，则将 I2S_I2SPR 寄存器的 MCKOE 位置 1 (I2SDIV 和 ODD (FRACT) 值应根据 MCK 输出的状态进行计算。有关详细信息，请参见时钟发生器。
3. 通过 I2SSTD[1:0] 和 PCMSYNC 位选择 I2S 标准，通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。此外，通过 I2S_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择 I2S 主模式和方向 (发送器或接收器)。
4. 如果需要，通过对 I2S_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
5. I2S_I2SCFGR 寄存器的 I2SE 位必须置 1。

WS 和 CK 的 IO 配置为输出模式。如果 I2S_I2SPR 的 MCKOE 位置 1，则 MCK 也需要配置为输出。

发送序列

将半字写入发送缓冲区后，发送序列随即开始。

假设写入发送缓冲区的左声道第一个数据写入数据寄存器 L；写入发送缓冲区的右声道第一个数据写入数据寄存器 R。数据从发送缓冲区传输到移位寄存器时，相应通道的 TXE 置 1，并且必须将对应于相应通道的数据写入相应通道的发送缓冲区。

一个完整帧表示先进行左通道数据发送再进行右通道数据发送。不存在仅发送左通道的部分帧。

首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，相应通道的 TXE 标志都将置 1，如果 I2S_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关各种 I2S 标准模式的写操作的更多详细信息，请参见 I2S 音频协议说明。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送的数据写入 I2S_DRL/DRR 寄存器。

要通过将 I2SE 清零来关闭 I2S，必须等待 TXE = 1 且 BSY = 0。

接收序列

此工作模式与发送模式相同，只有第 3 点存在不同（请参见 I2S 主模式所述的步骤），即通过 I2SCFG[1:0] 位设置主器件接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收相应通道缓冲区有接收到数据时，相应通道的 RXNE 标志即置 1，并且如果 I2S_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可通过一次或两次接收操作进入相应通道的接收缓冲区，具体取决于数据和通道长度配置。

读取 I2S_DRL/DRR 寄存器即会使相应通道的 RXNE 位清零。

有关各种 I2S 标准模式中读操作的更多详细信息，请参见 I2S 音频协议说明。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误并将 OVR 标志置 1。数据接收寄存的值保持先前的值，当前接收的数据丢失。

如果 I2S_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要关闭 I2S，需要执行特定操作来确保 I2S 正确完成传输周期而不启动新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 LSB 对齐模式（I2SSTD = 10）

- a) 等待倒数第二个 RXNE = 1 ($n - 1$)
- b) 然后等待 17 个 I2S 时钟周期（使用软件循环）

c) 禁止 I2S (I2SE = 0)

32 位通道长度上扩展的 16 位数据长度 (DATLEN = 00 且 CHLEN = 1), 使用 MSB 对齐、I2S 或 PCM 模式 (分别为 I2SSTD = 00、I2SSTD = 01 或 I2SSTD = 11)

a) 等待最后一个 RXNE

b) 然后等待 1 个 I2S 时钟周期 (使用软件循环)

c) 禁止 I2S (I2SE = 0)

对于 DATLEN 和 CHLEN 的所有其它组合, 无论通过 I2SSTD 位选择何种音频模式, 都将执行以下序列来关闭 I2S:

a) 等待倒数第二个 RXNE = 1 ($n - 1$)

b) 然后等待一个 I2S 时钟周期 (使用软件循环)

c) 禁止 I2S (I2SE = 0)

注: 传输期间, BSY 标志保持低电平。

19.4.2 I2S 从模式

对于从配置而言, I2S 可配置为发送模式或接收模式。

此工作模式所遵循的规则与 I2S 主模式配置基本相同。在从模式下, I2S 接口不产生时钟。时钟和 WS 信号从 I2S 接口所连接的外部主器件输入。这样, 用户便不需要配置时钟。

应遵循如下配置步骤:

1. 通过 I2SSTD[1:0] 位选择 I2S 标准, 通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择帧中每个通道的位数。此外, 通过 I2S_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择从器件的模式 (发送或接收)。
2. 如果需要, 通过对 I2S_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
3. I2S_I2SCFGR 寄存器的 I2SE 位必须置 1。

发送序列

当外部主器件发送时钟并且通过 I2S_WS 信号请求传输数据时，发送序列开始。必须首先使能从器件，然后外部主器件才能开始通信。主器件开始通信前，还必须加载 I2S 数据寄存器。

对于 I2S、MSB 对齐和 LSB 对齐模式，要发送的第一个数据为左声道数据，分别在左右声道数据寄存器写入相应通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。相应通道 TXE 标志随即置 1，以请求将相应通道的下一个数据写入 I2S 相应通道数据寄存器。

从器件需要首先为发送第一个数据做好准备，然后主器件才能产生时钟。WS 置位意味着首先发送左通道数据。

注：必须要在主器件发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE。

首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，相应通道的 TXE 标志都将置 1，如果 I2S_CR2 寄存器的 TXEIE 位置 1，将产生中断。

请注意，仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。

有关各种 I2S 标准模式中写操作的更多详细信息，请参见 I2S 音频协议说明。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 I2S_DRL/R 寄存器。如果在数据尚未写入 I2S_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。如果 I2S_CR2 寄存器的 ERRIE 位置 1，则当 I2S_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。这种情况下，必须关闭 I2S 并从左通道开始重新启动数据传输。

要通过将 I2SE 位清零来关闭 I2S，必须等待 TXE = 1 且 BSY = 0。

接收序列

此工作模式与发送模式相同，只有第 1 点存在不同（请参见 I2S 从模式所述的步骤），即通过 I2S_I2SCFGR 寄存器的 I2SCFG[1:0] 位设置从器件接收模式。

无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区有数据时，I2S_SR 寄存器中的相应通道的 RXNE 标志即置 1，并且如果 I2S_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入相应通道的接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收的数据要从相应通道的数据寄存器读取，读取 I2S_DR 寄存器即使 RXNE 位清零。

有关各种 I2S 标准模式的读操作的更多详细信息，请参见 I2S 音频协议说明。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误，OVR 标志将置 1。如果 I2S_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要在接收模式下关闭 I2S，必须在接收到最后一个 RXNE = 1 后立即将 I2SE 清零。

注：外部主器件应能够通过音频通道以 16 位或 32 位数据包发送/接收数据。

19.5 I2S 状态标志

应用程序可通过三个状态标志来全面监视 I2S 总线的状态。

19.5.1 忙标志 (BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。该标志表示 I2S 通信层的状态。

BSY 置 1 时，表示 I2S 正在忙于通信。在主接收模式 (I2SCFG = 11) 中，BSY 标志的情况例外，该标志在接收期间仍保持低电平。

如果软件需要禁止 I2S，可使用 BSY 标志检测传输是否结束。这可以避免损坏最后传输的数据。为此，必须严格遵循下述步骤。

在传输开始时（I2S 处于主接收器模式时除外），将 BSY 标志置 1。

出现以下情况时，BSY 标志被硬件清零：

- 传输完成时（主发送模式除外，在该模式下通信是连续的）
- 禁止 I2S 时

当通信连续时：

- 在主发送模式下，BSY 标志在所有传输期间均保持高电平
- 在从模式下，BSY 标志在每次传输之间变为低电平并持续一个 I2S 时钟周期

注：请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。

19.5.2 发送缓冲区为空 (TXE_L TXE_R)

如果此标志置 1，表示发送缓冲区为空，可将要发送的数据加载到其中。发送缓冲区已包含要发送的数据时，TXE 标志复位。禁止 I2S（I2SE 位复位）时，该标志也会复位。

19.5.3 接收缓冲区非空 (RXNE_L RXNE_R)

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 I2S_DRL/R 寄存器时，该标志复位。

19.5.4 下溢标志 (UDR)

在发送模式下，如果在软件尚未将任何值加载到 I2S_DR 之前出现第一个数据发送时钟，此标志将置 1。如果 I2S_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

UDR 位通过 I2S_ICLR 进行清零。

19.5.5 上溢标志 (OVR)

如果在尚未从 I2S_DR 寄存器读取上一个数据时又接收到新数据，此标志将置 1。因此，传入的数据将丢失。如果 I2S_CR2 寄存器中的 ERRIE 位置 1，可产生中断。这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 I2S_DR 寄存器执行的读操作将返回先前正确接收的数据。主器件后续发送的所有其它半字都将丢失。要将 OVR 位清零，应首先对 I2S_DR 寄存器执行读操作，然后再对 I2S_SR 寄存器进行读访问。

19.5.6 帧错误标志 (FRE)

仅当 I2S 配置为从模式时，此标志才可由硬件置 1。如果外部主器件没有按照从器件期望的那样改变 WS 线，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主器件与

I2S 从器件重新同步，需执行下列步骤：

1. 禁止 I2S。
2. 在 WS 线上检测到正确的电平时将其重新使能（WS 线在 I2S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主器件与从器件之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步信号线上存在噪音干扰。如果 ERRIE 位置 1，可产生错误中断。同步失效标志 (FRE) 由软件操作 I2S_ICLR 进行清零。

19.6 I2S 中断 DMA

下表为 I2S 中断列表

中断事件	事件标志	使能控制
发送缓存空	TXE_L TXE_R	TXEIE
接收缓存非空	RXNE_RXNE_R	RXNEIE
上溢错误	OVR	ERRIE
下溢错误	UDR	
帧错误	FRE	

DMA 特性

- 在 I2S 模式下，发送数据和接收的数据可以使用 DMA 进行搬运。

19.7 I2S 寄存器

基地址：0x4001_5400

寄存器	偏移地址	描述
I2S_CR	0x004	控制寄存器
I2S_SR	0x008	中断和状态寄存器

I2S_CFGR	0x01C	配置寄存器
I2S_PR	0x020	主模式分频寄存器
I2S_ICR	0x024	中断标志清除寄存器
I2S_DRL	0x028	左声道发送数据与接收数据寄存器
I2S_DRR	0x02C	右声道发送数据与接收数据寄存器

表 19-1 I2S 寄存器

19.7.1 I2S 控制寄存器（CSR_CR）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
TXEIE		RXNEIE		ERRIE		RSV								RDMAEN		LDMAEN							
RW		RW		RW										RW		RW							

位	标记	功能描述
31:8	Reserved	保留
7	TXEIE	发送缓冲区空中断使能 (Tx buffer empty interrupt enable) 0: 屏蔽 TXE 中断 1: 使能 TXE 中断。TXE 标志置 1 时产生中断请求。
6	RXNEIE	接收缓冲区非空中断使能 (RX buffer not empty interrupt enable) 0: 屏蔽 RXNE 中断 1: 使能 RXNE 中断 RXNE 标志置 1 时产生中断请求。
5	ERRIE	错误中断使能 (Error interrupt enable) 此位用于在出现错误条件 (UDR、OVR 和 FRE) 时控制中断的生成。 0: 屏蔽错误中断 1: 使能错误中断
4:2	Reserved	保留
1	RDMAEN	右声道缓冲区 DMA 数据发送使能 (Right buffer DMA enable) 当此位置 1 时，每当右声道有数据发送接收标志置 1 时，即产生 DMA 请求。 0: 禁止右声道缓冲区 DMA 1: 使能右声道缓冲区 DMA
0	LDMAEN	左声道缓冲区 DMA 声道数据发送使能 (Left buffer DMA enable) 当此位置 1 时，每当左声道有数据发送接收标志置 1 时，即产生 DMA 请求。 0: 禁止左声道缓冲区 DMA 1: 使能左声道缓冲区 DMA

19.7.2 I2S 状态寄存器 (CSR_SR)

偏移地址: 0x08

复位值: 0x0000 8002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
TXE_R	RXNE_R	OVR_R	RSV										FRE		
RO	RO	RO											RO		
7		6		5		4		3		2		1		0	
BSY	OVR_L	RSV				UDR_R		UDR_L		TXE_L		RXNE_L			
RO	RO					RO		RO		RO		RO			

位	标记	功能描述
31:16	Reserved	保留
15	TXE_R	右通道发送缓冲区为空 (Transmit buffer empty) 0: 发送缓冲区非空 1: 发送缓冲区为空
14	RXNE_R	右通道接收缓冲区非空 (Receive buffer not empty) 0: 发送缓冲区为空 1: 接收缓冲区非空
13	OVR_R	右通道上溢标志 (Overrun flag) 0: 未发生上溢 1: 发生上溢 此标志由硬件置 1, 由软件清零
12:9	Reserved	保留
8	FRE	帧错误 (Frame Error) 0: 无帧错误 1: 发生帧错误 该位由硬件置 1, 由软件清零。 无论选择哪种音频协议, 它会在从模式下的非预期时间检测 WS 线的变化, 通知外部主器件和从器件之间的同步失效情况。
7	BSY	BSY: 忙标志 (Busy flag) 0: I2S 不繁忙 1: I2S 忙于通信或者发送缓冲区不为空 此标志由硬件置 1 和清零。
6	OVR_L	左通道上溢标志 (Overrun flag)

		0: 未发生上溢 1: 发生上溢 此标志由硬件置 1, 由软件清零
5:4	Reserved	保留
3	UDR_R	右通道下溢标志 (Underrun flag) 0: 未发生下溢 1: 发生下溢 此标志由硬件置 1, 由软件清零
2	UDR_L	左通道下溢标志 (Underrun flag) 0: 未发生下溢 1: 发生下溢 此标志由硬件置 1, 由软件清零
1	TXE_L	左通道发送缓冲区为空 (Transmit buffer empty) 0: 发送缓冲区非空 1: 发送缓冲区为空
0	RXNE_L	左通道接收缓冲区非空 (Receive buffer not empty) 0: 发送缓冲区为空 1: 接收缓冲区非空

19.7.3 I2S 配置寄存器 (I2S_CFGR)

偏移地址: 0x1C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved										I2SE	I2SCFG				
										RW	RW				
7	6	5	4	3	2	1	0								
PCMSYNC	CKSEL	I2SSTD			CKPOL	DATLEN			CHLEN						
RW	RW	RW			RW	RW			RW						

位	标记	功能描述
31:11	Reserved	保留
10	I2SE	I2S 使能 (I2S Enable) 0: 禁止 I2S 1: 使能 I2S
9:8	I2SCFG	I2S 配置模式 (I2S configuration mode) 00: 从模式 - 发送 01: 从模式 - 接收 10: 主模式 - 发送 11: 主模式 - 接收
7	PCMSYNC	PCM 帧同步 (PCM frame synchronization) 0: 短帧同步 1: 长帧同步 注: 只有在 I2SSTD = 11 (使用 PCM 标准) 时, 此位才有意义
6	CKSEL	主模式下 I2S 时钟选择 0 PCLK 1 HCLK
5:4	I2SSTD	I2S 标准选择 (I2S standard selection) 00: I2S Philips 标准。 01: MSB 对齐标准 (左对齐) 10: LSB 对齐标准 (右对齐) 11: PCM 标准 注: 为确保正确运行, 应在 I2S 禁止时配置这些位
3	CKPOL	
2:1	DATLEN	要传输的数据长度 (Data length to be transferred)

		00: 16 位数据长度 01: 24 位数据长度 10: 32 位数据长度 11: 不允许 注: 为确保正确运行, 应在 I2S 禁止时配置这些位。
0	CHLEN	CHLEN: 通道长度(每个音频通道的位数)(Channel length (number of bits per audio channel)) 0: 16 位 1: 32 位 只有在 DATLEN = 00 时, 此位的写0操作才有意义, 在其他数据长度下此位被强制写1。 注: 为确保正确运行, 应在 I2S 禁止时配置此位。

19.7.4 I2S 预分频寄存器 (I2S_PR)

偏移地址: 0x20

复位值: 0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACT						MCKOE	ODD	I2SDIV							
RW						RW	RW	RW							

位	标记	功能描述
31:16	Reserved	保留
15:10	FRACT	预分频器小数分频设置, 0x00时ODD设置有效, 小数分频为0, 非零时, ODD设置无效 实际分频值为 $= (I2SDIV + FRACT/64) * 2$ 注: 应在 I2S 禁止时配置此位。
9	MCKOE	主时钟输出使能 (Master clock output enable) 0: 禁止主时钟输出 1: 使能主时钟输出 注: 应在 I2S 禁止时配置此位。
8	ODD	预分频器的奇数因子 (Odd factor for the prescaler) 0: 实际分频值为 $= I2SDIV * 2$ 1: 实际分频值为 $= (I2SDIV * 2) + 1$ 详细参见预分频设置 注: 应在 I2S 禁止时配置此位。
7:0	I2SDIV	I2SDIV: I2S 线性预分频器 (I2S Linear prescaler) I2SDIV [7:0] = 0 为禁用值。 详细参见预分频设置 注: 应在 I2S 禁止时配置这些位。

19.7.5 I2S 中断状态清除寄存器 (I2S_ICR)

偏移地址: 0x24

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FRE	RSV	OVR	RSV		UDF	RSV		
							W0		W0			W0			

位	标记	功能描述
31:9	Reserved	保留
8	FRE	帧错误 (Frame Error) 标志清除 写0清除 写1无效
7	Reserved	保留
6	OVR	上溢标志 (Overflow flag) 写0清除 写1无效
5:4	Reserved	保留
3	UDF	下溢标志 (Underrun flag) 写0清除 写1无效
2:0	Reserved	保留

19.7.6 I2S 数据寄存器 L (I2S_DRL)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRL															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	DRL	数据寄存器 (Data register) 已接收或者要发送的左声道数据与PCM的数据。 数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。 对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。

19.7.7 I2S 数据寄存器 R (I2S_DRR)

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRR															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	DRR	数据寄存器 (Data register) 已接收或者要发送的右声道数据。PCM模式下无效。 数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。 对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。

20. 通用同步异步收发器（UART）

20.1 概述

通用 UART 模块（UART1/2/3/4/5/6）支持以下基本功能：

- 半双工和全双工传输
- 单线半双工传输
- 8/9-Bit 传输数据长度
- 硬件奇偶校验
- 1/1.5/2-Bit 停止位
- 四种不同传输模式
- 16-Bit 波特率计数器
- 多机通讯
- 硬件地址识别
- DMAC 硬件传输握手
- 硬件流控
- 支持单线模式

20.2 功能框图

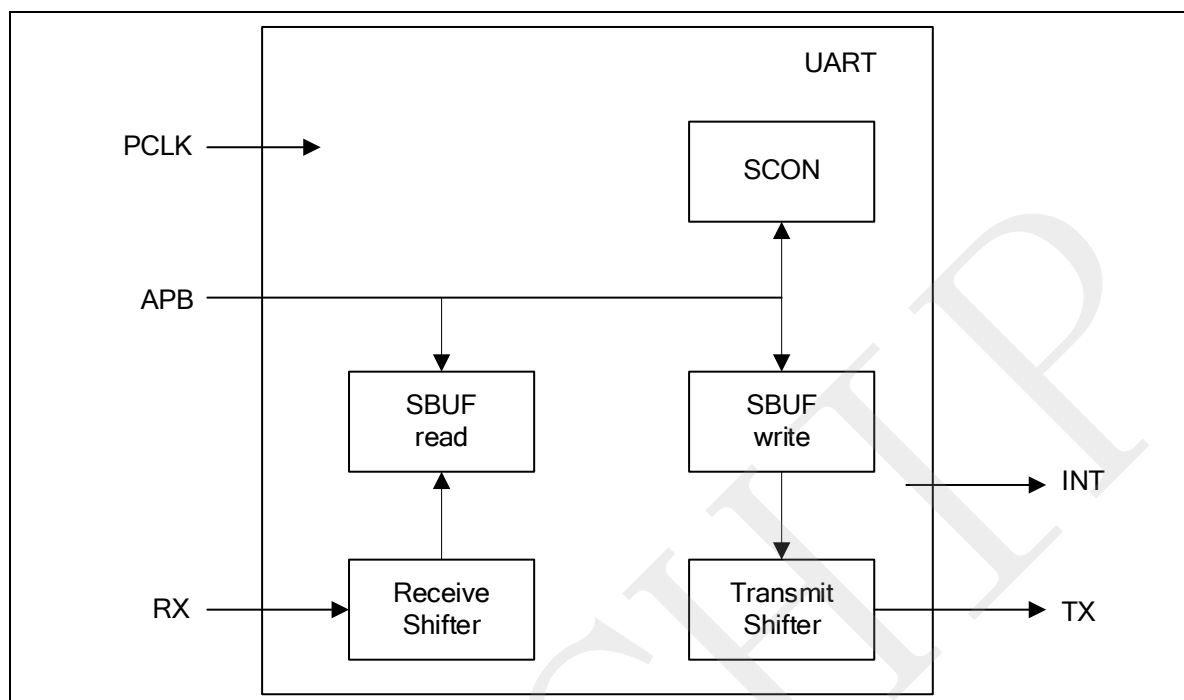


图 20-1 结构框图

20.3 工作模式

20.3.1 Mode0（同步模式，半双工）

当工作在 Mode0 时，UART 为同步传输模式，其波特率固定为 PCLK 时钟频率的 1/12。UART 收发数据都由 RXD 输入输出，RXD 此时为双向端口。同步移位时钟由 TXD 输出，TXD 此时为输出端口。注意，本模式只能作为主机发送 UART 同步移位时钟，不可以作为从机接收外部输入的 UART 同步移位时钟。Mode0 时，传输数据位宽为 8-Bit，没有起始位和结束位。

当 UARTx_SCON.SM[1:0]=2'b00，可进入 Mode0 工作模式。

20.3.1.1 发送数据

发送数据时，清除 UARTx_SCON.REN 位，并将数据写入 UARTx_SBUF 寄存器中。此时，发送数据将从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

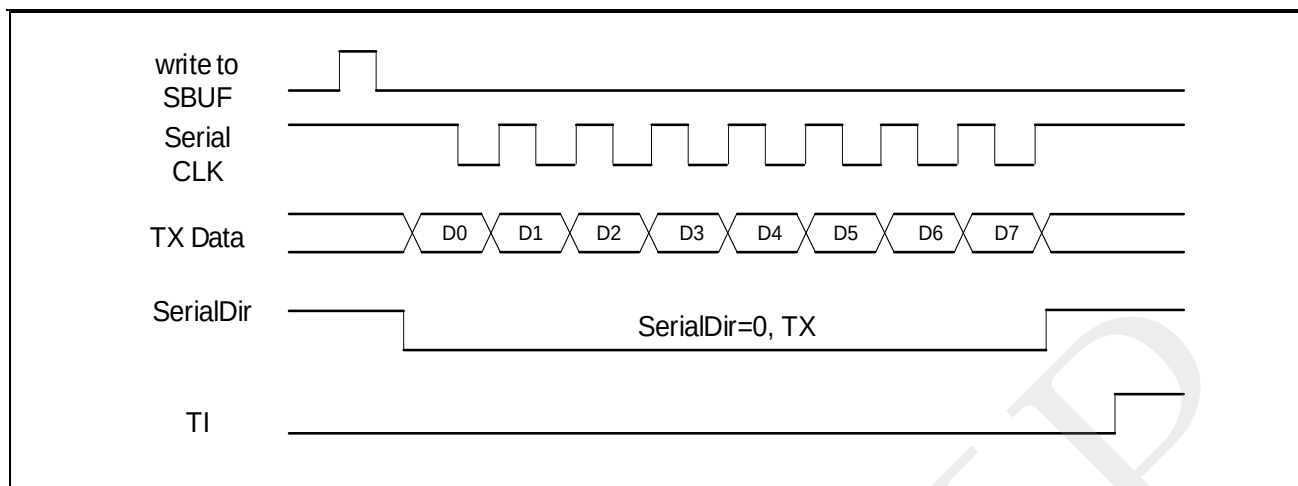


图 20-2 Mode0 发送数据

20.3.1.2 接收数据

接收数据时，将 `UARTx_SCON.REN` 位置 1，并将 `UARTx_ISR.RC` 位清零。当接收结束，数据可从 `UARTx_SBUF` 寄存器读取。此时，接收数据从 `RXD` 输入（低位在先，高位在后），同步移位时钟从 `TXD` 输出。

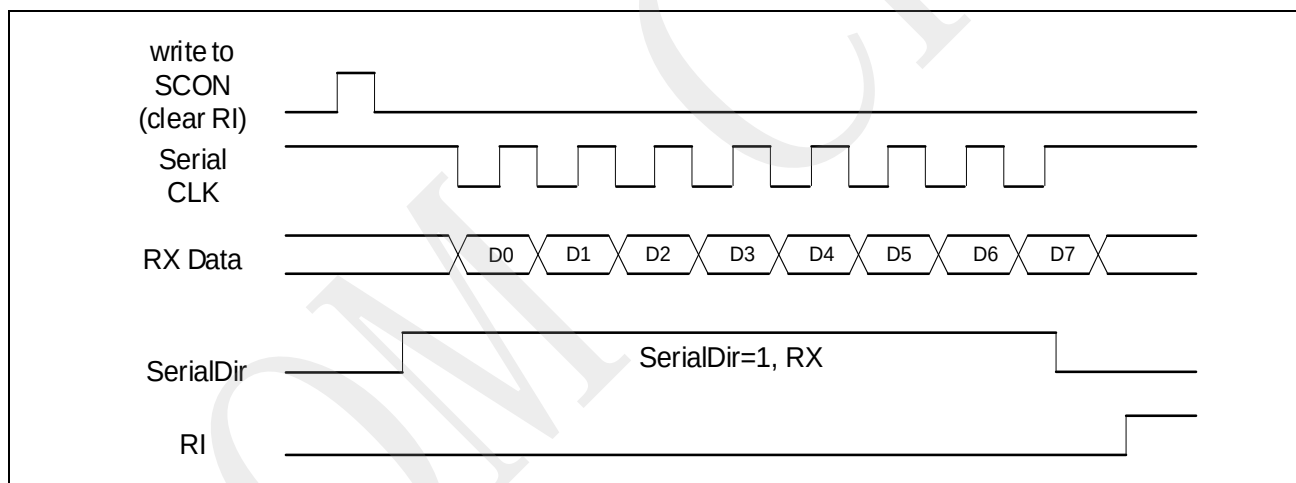


图 20-3 Mode0 接收数据

20.3.2 Mode1（异步模式，全双工）

当工作在 Mode1 时，发送数据通过 `TXD` 发送，接收数据通过 `RXD` 接收。该数据帧由以下比特位组成：1-Bit 起始位“0”开始，紧接着 8-Bit 数据位（低位在先，高位在后），最后是 1/1.5/2-Bit 结束位“1”。结束位长度由 `UARTx_SCON.STOPBIT[1:0]` 选择。该模式下，波特率由内部 16-Bit 可编程波特率发生器产生。波特率计算方式请见后面波特率编程章节。

当 `UARTx_SCON.SM[1:0]=2'b01` 时，可进入 Mode1 工作模式。

20.3.2.1 发送数据

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

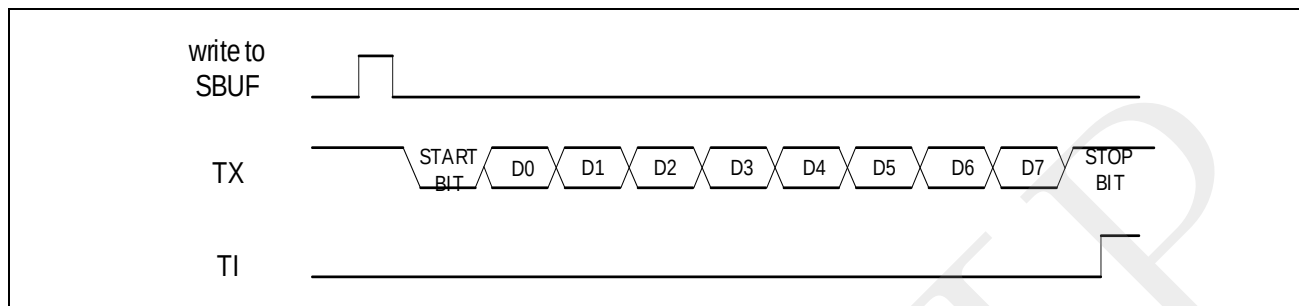


图 20-4 Mode1 发送数据

20.3.2.2 接收数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读出。

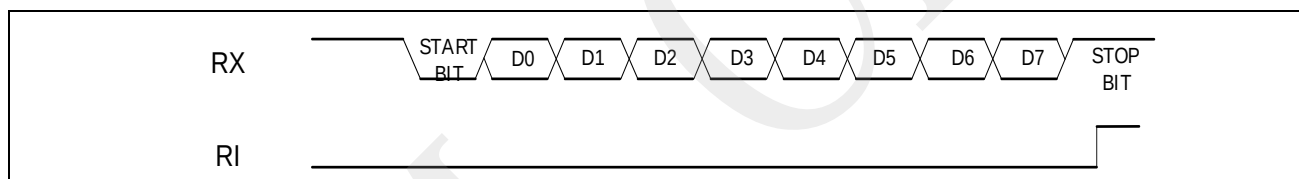


图 20-5 Mode1 接收数据

20.3.3 Mode2（异步模式，全双工）

当工作在 Mode2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由以下比特位组成：1-Bit 起始位“0”开始，接着是 8-Bit 数据位，1-Bit TB8 位和 1/1.5/2-Bit 结束位。结束位长度由 UARTx_SCON.STOPBIT[1:0] 选择。额外的 TB8 位用于在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位或者数据位来使用。该模式下，波特率为 PCLK 时钟的固定分频，波特率计算方式请见后面波特率编程章节。

当 UARTx_SCON.SM[1:0]=2'b10，可进入 Mode2 工作模式。

20.3.3.1 发送数据

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

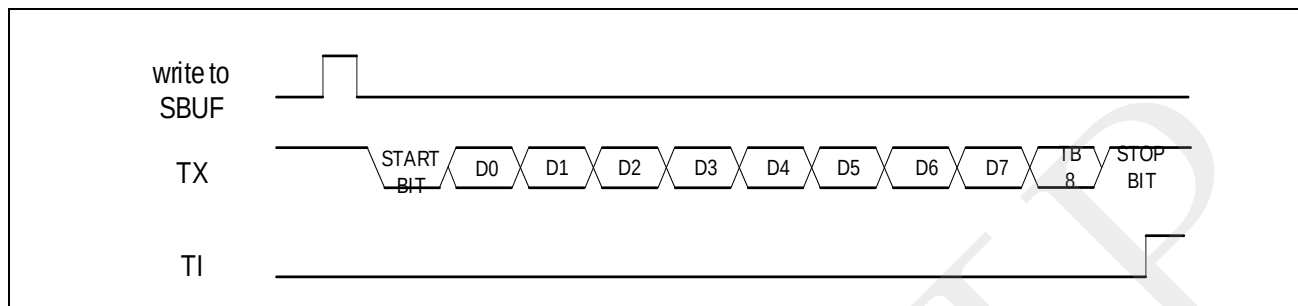


图 20-6 Mode2 发送数据

20.3.3.2 接收数据

接收数据时，需将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上的数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读取。

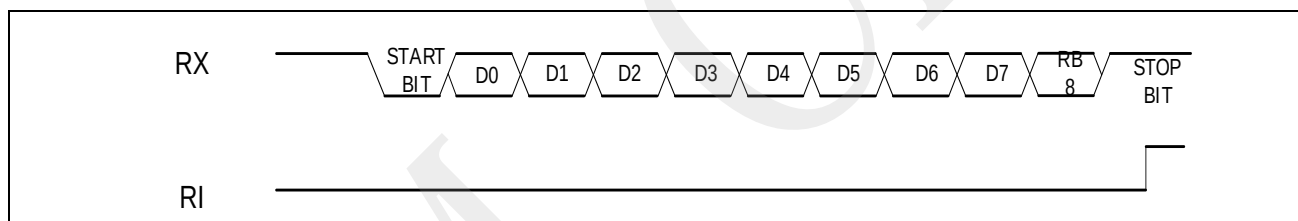


图 20-7 Mode2 接收数据

20.3.4 Mode3（异步模式，全双工）

当工作在 Mode3 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由以下比特位组成组成：1-Bit 起始位“0”开始，接着是 8-Bit 数据位，1-Bit TB8 位和 1/1.5/2-Bit 结束位。结束位长度由 UARTx_SCON.STOPBIT[1:0]选择。额外的 TB8 位用于在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位或者数据位来使用。

该模式下，波特率由内部 16-Bit 可编程波特率发生器产生。波特率计算方式请见后面波特率编程章节。

当 UARTx_SCON.SM[1:0]=2'b11，可进入 Mode3 工作模式。

20.3.4.1 发送数据

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

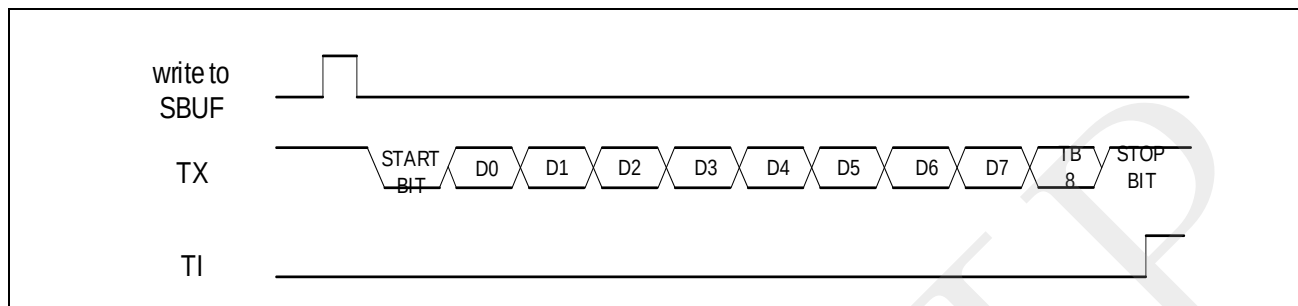


图 20-8 Mode3 发送数据

20.3.4.2 接收数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 UARTx_SBUF 寄存器读出。

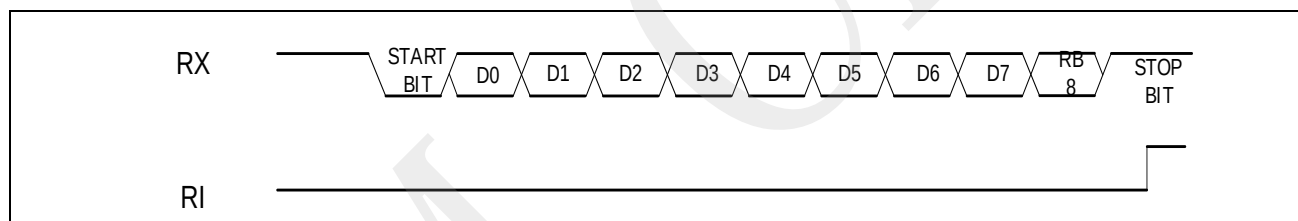


图 20-9 Mode3 接收数据

20.3.5 Single-wire Half-duplex（单线半双工）

当 UARTx_SCON.HDSEL=1'b1，可进入单线半双工传输模式。在该模式下，TX 与 RX 信号线在模块内部相连。RX 信号不再使用，接收发送的数据都经由 TX 信号完成。TX 信号的方向控制由硬件逻辑完成，无需软件控制。当发送缓存为空时，TX 信号始终为输入（接收状态）。当向发送缓存填入一个数据，TX 信号变为输出（发送状态）。当发送完成，发送缓存变为空，TX 信号又回到输入（接收状态）。

注意：

1. 即使模块处于接收过程中，只要发送缓存被填入数据，TX 信号立即变为输出（发送状态）。接收过程将被打断，并得到一个错误的接收数据。
2. 单线半双工模式并不是 Mode0 模式，虽然两者都是用一根信号分时复用地完成了

数据的收发功能。但是 Mode0 为同步传输，有同步时钟。而单线半双工模式为异步传输，所有的波特率计算还是遵照 Mode1-2-3 的波特率公式，仅仅是收发功能复用到了 TX 这一根信号上面。

20.4 波特率编程

20.4.1 Mode0

当工作在 Mode0 时，UART 同步移位时钟频率固定为 PCLK 的 1/12。具体公式如下所示：

$$BaudRate = \frac{Freq}{12}$$

其中，Freq 为 PCLK 的时钟频率。

20.4.2 Mode1/3

当工作在 Mode1 或者 Mode3 时，波特率由可编程波特率发生器产生。具体公式如下所示：

$$BaudRate = \frac{Freq}{OVER * SCNT}$$

其中，Freq 为 PCLK 的时钟频率；OVER 可以为 16 或者 8，由 UARTx_SCON.OVER 的值决定；SCNT 为 16-Bit 波特率计数器 UARTx_SCNT 的计数值。

以下表格为常用系统主频和波特率的对照表。该表格同样适用于 LPUART1/2 的波特率计算。

注意：LPUART1/2 支持 OVER4/OVER8/OVER16。UART1/2 仅支持 OVER8/OVER16。

波特率	PCLK = 4 MHz								
	OVER4 (仅 LPUART1/2 支持)			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%	104	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%	52	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%	26	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%	13	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%	7	35714.29	-6.99%
57600	17	58823.53	2.12%	9	55555.56	-3.55%	4	62500.00	8.51%
76800	13	76923.08	0.16%	7	71428.57	-6.99%	3	83333.33	8.51%
115200	9	111111.11	-3.55%	4	125000.00	8.51%	2	125000.00	8.51%

表 20-1 PCLK=4MHz 波特率计算表

波特率	PCLK = 8 MHz								
	OVER4 (仅 LPUART1/2 支持)			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	833	2400.96	0.04%	417	2398.08	-0.08%	208	2403.85	0.16%
4800	417	4796.16	-0.08%	208	4807.69	0.16%	104	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%	52	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%	26	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%	13	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%	9	55555.56	-3.55%
76800	26	76923.08	0.16%	13	76923.08	0.16%	7	71428.57	-6.99%
115200	17	117647.06	2.12%	9	111111.11	-3.55%	4	125000.00	8.51%

表 20-2 PCLK=8MHz 波特率计算表

波特率	PCLK = 16 MHz								
	OVER4 (仅 LPUART1/2 支持)			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%	417	2398.08	-0.08%
4800	833	4801.92	0.04%	417	4796.16	-0.08%	208	4807.69	0.16%
9600	417	9592.33	-0.08%	208	9615.38	0.16%	104	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%	52	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%	26	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%	17	58823.53	2.12%
76800	52	76923.08	0.16%	26	76923.08	0.16%	13	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%	9	111111.11	-3.55%
128000	31	129032.26	0.81%	16	125000.00	-2.34%	8	125000.00	-2.34%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%	4	250000.00	-2.34%

表 20-3 PCLK=16MHz 波特率计算表

波特率	PCLK = 24 MHz								
	OVER4 (仅 LPUART1/2 支持)			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	2500	2400.00	0.00%	1250	2400.00	0.00%	625	2400.00	0.00%
4800	1250	4800.00	0.00%	625	4800.00	0.00%	313	4792.33	-0.16%
9600	625	9600.00	0.00%	313	9584.66	-0.16%	156	9615.38	0.16%
19200	313	19169.33	-0.16%	156	19230.77	0.16%	78	19230.77	0.16%
38400	156	38461.54	0.16%	78	38461.54	0.16%	39	38461.54	0.16%
57600	104	57692.31	0.16%	52	57692.31	0.16%	26	57692.31	0.16%
76800	78	76923.08	0.16%	39	76923.08	0.16%	20	75000.00	-2.34%
115200	52	115384.62	0.16%	26	115384.62	0.16%	13	115384.62	0.16%
128000	47	127659.57	-0.27%	23	130434.78	1.90%	12	125000.00	-2.34%
256000	23	260869.57	1.90%	12	250000.00	-2.34%	6	250000.00	-2.34%

表 20-4 PCLK=24MHz 波特率计算表

波特率	PCLK = 32 MHz								
	OVER4 (仅 LPUART1/2 支持)			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	3333	2400.24	0.01%	1667	2399.52	-0.02%	833	2400.96	0.04%
4800	1667	4799.04	-0.02%	833	4801.92	0.04%	417	4796.16	-0.08%
9600	833	9603.84	0.04%	417	9592.33	-0.08%	208	9615.38	0.16%
19200	417	19184.65	-0.08%	208	19230.77	0.16%	104	19230.77	0.16%
38400	208	38461.54	0.16%	104	38461.54	0.16%	52	38461.54	0.16%
57600	139	57553.96	-0.08%	69	57971.01	0.64%	35	57142.86	-0.79%
76800	104	76923.08	0.16%	52	76923.08	0.16%	26	76923.08	0.16%
115200	69	115942.03	0.64%	35	114285.71	-0.79%	17	117647.06	2.12%

表 20-5 PCLK=32MHz 波特率计算表

波特率	PCLK = 48 MHz								
	OVER4 (仅 LPUART1/2 支持)			OVER8			OVER16		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	5000	2400.00	0.00%	2500	2400.00	0.00%	1250	2400.00	0.00%
4800	2500	4800.00	0.00%	1250	4800.00	0.00%	625	4800.00	0.00%
9600	1250	9600.00	0.00%	625	9600.00	0.00%	313	9584.66	-0.16%
19200	625	19200.00	0.00%	313	19169.33	-0.16%	156	19230.77	0.16%
38400	313	38338.66	-0.16%	156	38461.54	0.16%	78	38461.54	0.16%
57600	208	57692.31	0.16%	104	57692.31	0.16%	52	57692.31	0.16%
76800	156	76923.08	0.16%	78	76923.08	0.16%	39	76923.08	0.16%
115200	104	115384.62	0.16%	52	115384.62	0.16%	26	115384.62	0.16%

表 20-6 PCLK=48MHz 波特率计算表

20.4.3 Mode2

当工作在 Mode2 时，波特率为固定值，具体公式如下所示：

$$\text{BaudRate} = \frac{\text{Freq}}{\text{OVER}}$$

其中，Freq 为 PCLK 的时钟频率；OVER 可以为 32 或者 16，由 UARTx_SCON.OVER 的值决定。

20.5 传输数据结构

在不同工作模式下，传输数据的组成有所不同，请参考以下表格：

工作模式	传输位宽	数据组成
Mode0	8-Bit	<u>8-Bit Data</u>
Mode1	10-Bit	<u>1-Bit Start Bit</u> + <u>8-Bit Data</u> + <u>(1/1.5/2)-Bit Stop Bit</u>
Mode2	11-Bit	<u>1-Bit Start Bit</u> + <u>8-Bit Data</u> + <u>1-Bit B8</u> + <u>(1/1.5/2)-Bit Stop Bit</u>
Mode3	11-Bit	<u>1-Bit Start Bit</u> + <u>8-Bit Data</u> + <u>1-Bit B8</u> + <u>(1/1.5/2)-Bit Stop Bit</u>

表 20-7 传输数据组成结构

B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

应用场景	UARTx_SCON. ADDRDET	UARTx_SCON. B8CONT[1:0]	B8 数据含义
奇偶校验	--	01/10	接收时，B8 是所收到的 8-Bit 数据的奇偶校验位； 发送时，B8 是所发送的 8-Bit 数据的奇偶校验位；
多机通讯	1	--	B8=1，代表当前是地址帧； B8=0，代表当前是数据帧；
其他	0	00/11	接收/发送的 DATA[8]

表 20-8 B8 数据含义

注意：

- 当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制；

20.6 帧错误检测

当工作在 Mode1/2/3 时，UART 具有帧错误检测功能，硬件会自动检测接收到的帧数

据是否带有效的 Stop 位。如果没有收到有效 Stop 位，则 UARTx_ISR.FE 置 1。UARTx_ISR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有效 Stop 位，也不会把 UARTx_ISR.FE 标志清 0。

20.7 多机通讯

当工作在 Mode2/3 时，可以打开多机通讯功能，为此在其帧格式中增加了 1 位 TB8/RB8。将 UARTx_SCON.ADRDET 位置“1”，可开启多机通讯功能。

当开启多机通讯后，主机可以通过 UARTx_SBUF[8]来区分当前的发送帧是地址帧（UARTx_SBUF[8]=1）还是数据帧（UARTx_SBUF[8]=0）。从机会忽略 RB8 位（接收到的 Bit8 数据位）为“0”的接收帧，该帧数据不会存入到从机的 UARTx_SBUF 寄存器中，从机也不会产生接收中断。当接收帧的 RB8 位（接收到的 Bit8 数据位）为“1”，表明其是地址帧，从机会继续判断接收到的地址与其自身地址是否相等。如果匹配，则从机会对 UARTx_SBUF[8]置“1”，同时将该地址帧存入到 UARTx_SBUF 寄存器中。并对 UARTx_ISR.RC 置“1”，以表明该帧为地址帧并且地址已经匹配。从机软件看到 UARTx_SBUF[8]=1 并且 UARTx_ISR.RC=1 后，先把 UARTx_SCON.ADEDET 位清“0”，然后准备接收给它的数据帧。如果地址不相等，表明主机并不是寻址该从机，从机硬件保持 UARTx_SBUF[8]和 UARTx_ISR.RC 为“0”，软件保持 UARTx_SCON.ADRDET 位为“1”，从机继续处于地址监听状态。

20.7.1 自动地址识别

当开启多机通讯后（UARTx_SCON.ADRDET 位置“1”），自动地址识别功能也将开启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收从机会给出 UARTx_ISR.RC 接收标志。如果地址不匹配，则接收从机不会给出任何接收标志。

如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 Stop 位代替。当从机接收到匹配的地址帧和有效的 Stop 位时，UARTx_ISR.RC 会被置“1”。

20.7.2 给定地址

UART 设备的 UARTx_SADDR 寄存器用来表示自己的设备给定地址，

UARTx_SADEN 寄存器是地址掩码，可以用来定义地址中的无关位。当 UARTx_SADEN 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，UARTx_SADEN 寄存器必须设为 8'hFF。给定地址公式如下所示：

$$\text{GivenAddr} = \text{SADDR} \& \text{SADEN}$$

20.7.3 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 8'hFF。

$$\text{BroadcastAddr} = \text{SADDR} | \text{SADEN}$$

20.7.4 举例

假设某从机的 UARTx_SADDR 和 UARTx_SADEN 配置如下：

SADDR: 8'b01101001

SADEN: 8'b11111011

那么其给定地址和广播地址如下：

Given: 8'b01101x01

Broadcast: 8'b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

8'b01101001 和 8'b01101101 (given address)

8'b11111011 和 8'b11111111 (broadcast address)。

20.8 DMAC 硬件握手

UART 模块支持 DMAC 的硬件握手逻辑。

将 UARTx_SCON.DMACTXEN 设置为 1，可以打开 UART TX 的 DMAC 硬件握手逻辑。当发送缓存为空时，UART 会向 DMAC 发出数据搬运请求 TX REQ。DMAC 收到该 TX REQ 信号，则从 DMAC 源地址搬运一个帧的发送数据到 UARTx_SBUF 中。

上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

将 UARTx_SCON.DMACRXEN 设置为 1，可以打开 UART RX 的 DMAC 硬件握手逻辑。

之前，先判断 nCTS 的高低电平。如果 nCTS 为低电平（低电平表示对端设备可以接收下一帧数据），则 UART 发送下一帧数据。如果 nCTS 为高电平（高电平表示对端设备不可以接收下一帧数据），则 UART 暂停发送下一帧数据。nCTS 拉高不会影响当前发送帧的传输，只会暂停下一帧的发送，直到 nCTS 重新变低为止。

当 nCTS 流控使能时，一旦 nCTS 信号发生翻转，会把 UARTx_SFLAG.CTSIF 置 1。如果使能 CTSIE 中断（UARTx_SCON.CTSIE 设置为 1），则会产生中断。同时，nCTS 信号的高低电平会记录在 UARTx_SFLAG.CTS 标志位中。

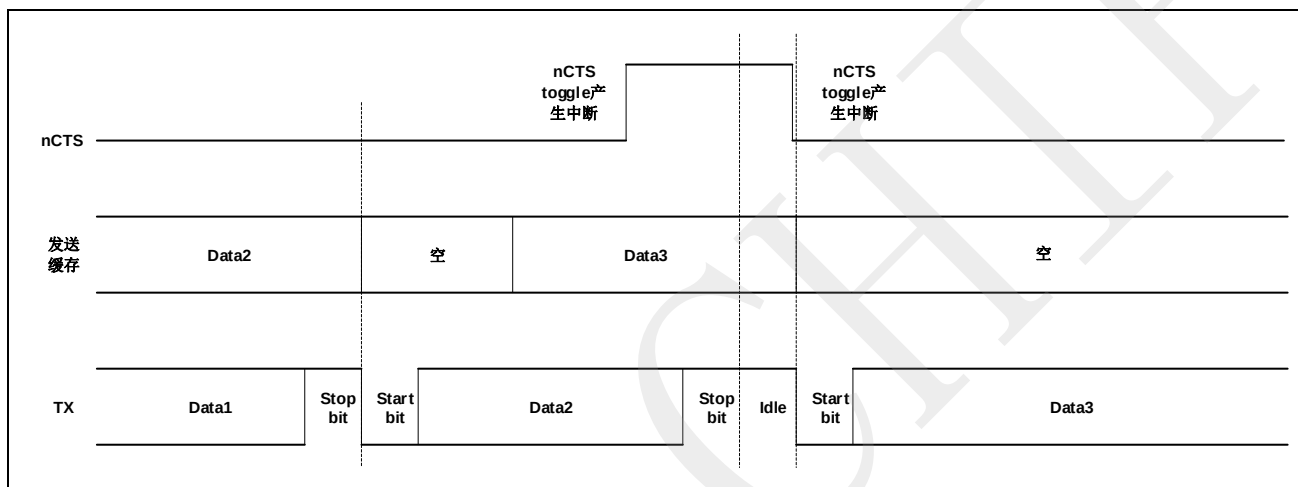


图 20-12 nCTS 硬件流控信号

20.10 收发端缓存

20.10.1 接收缓存

UART 模块接收端有一个帧(8/9-Bit)的接收缓存，也就是说当一帧数据接收完毕后，接收缓存中的数据会被一直保持，直到下一帧数据的 Stop 位接收完毕后，接收缓存才会更新到这一帧所收到的数据。

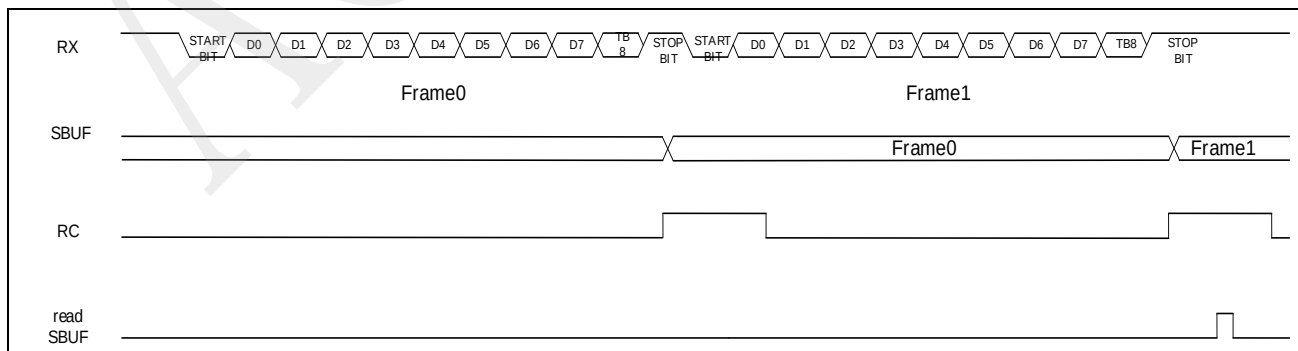


图 20-13 接收缓存

20.10.2 发送缓存

UART 模块发送端有一个帧(8/9-Bit)的发送缓存，当 UART 在发送当前帧时，软件可以对 UARTx_SBUF 写入下一个发送数据。UARTx_ISR.TXE 位表示了当前发送缓存的空满状态，当 UARTx_ISR.TXE=0 时，表明当前发送缓存满，软件不能对 UARTx_SBUF 写入下一个发送数据。如果软件继续写入 UARTx_SBUF，则该笔写入数据会硬件丢弃。当 UARTx_ISR.TXE=1 时，表明当前发送缓存空，软件可以写入下一个需要发送的数据帧。该笔发送数据会保存在发送缓存中，直到当前发送数据完成，硬件自动把发送缓存中的数据装载入移位寄存器中发送出去。

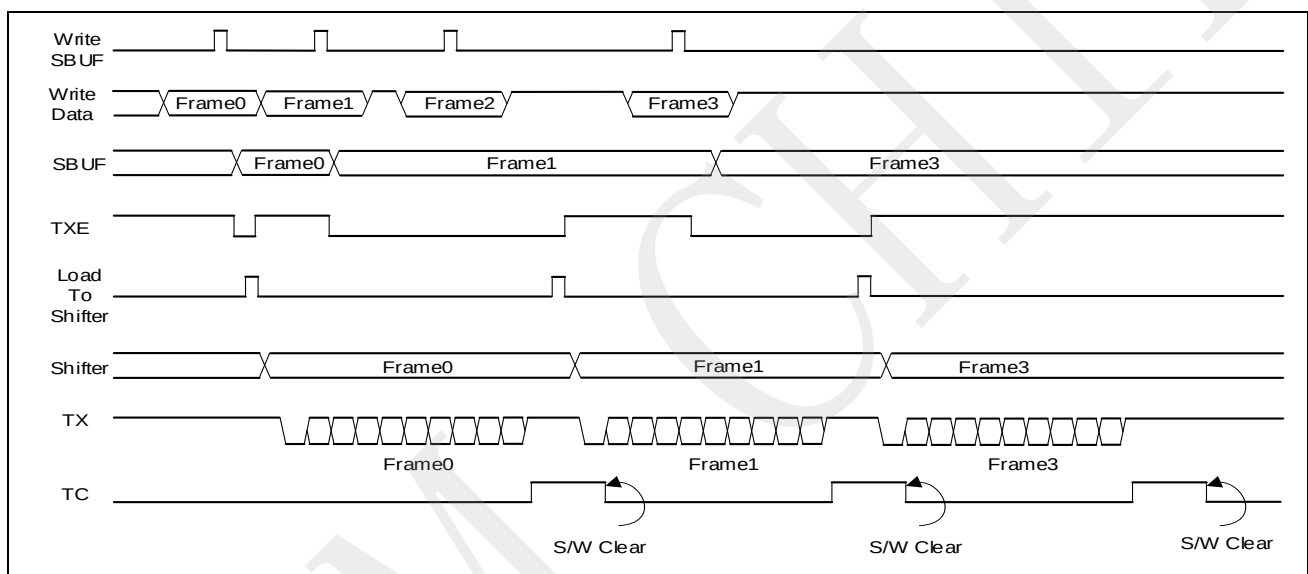


图 20-14 发送缓存

20.11 寄存器

UART1 基地址: 0x4001 3800

UART2 基地址: 0x4000 4400

UART3 基地址: 0x4000 4800

UART4 基地址: 0x4000 4C00

UART5 基地址: 0x4000 5000

UART6 基地址: 0x4001 3C00

寄存器	偏移地址	描述

UARTx_SBUF	0x00	数据寄存器
UARTx_SCON	0x04	控制寄存器
UARTx_SADDR	0x08	地址寄存器
UARTx_SADEN	0x0C	地址掩码寄存器
UARTx_ISR	0x10	中断标志位寄存器
UARTx_ICR	0x14	中断标志位清除寄存器
UARTx_SCNT	0x18	波特率寄存器

表 20-9UART 寄存器列表

20.11.1 数据寄存器（UARTx_SBUF）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DATA[8]	DATA[7:0]							
								RW							

位	标记	功能描述
31:9	Reserved	保留
8	DATA[8]	在Mode0/1下，读取该位为0，写入该位无效； 在Mode2/3下，该位表示Bit8数据位，分以下两种情况： （1）当硬件奇偶校验位开启时，接收时该位为接收数据奇偶校验位，校验由硬件进行，如校验出错，校验错误标志位PE置1；发送时该位无效，发送数据奇偶校验位由硬件计算并发送； （2）当硬件奇偶校验位关闭时，接收时该位为接收数据Bit8；发送时该位为发送数据Bit8； 注意：当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受B8CONT控制；
7:0	DATA[7:0]	发送数据时，将发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读取。

20.11.2 控制寄存器 (UARTx_SCON)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RSV	HDSEL	FEIE	CTSIE	CTSEN	RTSEN	DMATXEN	DMARXEN
	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
STOPBIT		PEIE	Reserved			OVER	TXEIE
RW		RW				RW	RW
7	6	5	4	3	2	1	0
SM		ADRDET	REN	B8CONT		TCIE	RCIE
RW		RW	RW	RW		RW	RW

位	标记	功能描述
31:23	Reserved	保留
22	HDSEL	单线半双工模式使能 0: 关闭单线半双工模式 1: 打开单线半双工模式
21	FEIE	帧错误中断使能 0: 关闭中断 1: 打开中断
20	CTSIE	CTS信号翻转中断使能 0: 关闭中断 1: 打开中断
19	CTSEN	硬件flow control信号使能位 0: 关闭flow control信号 1: 打开flow control信号
18	RTSEN	
17	DMATXEN	TX DMAC的硬件握手信号使能位 0: 关闭硬件握手信号 1: 打开硬件握手信号
16	DMARXEN	RX DMAC的硬件握手信号使能位 0: 关闭硬件握手信号 1: 打开硬件握手信号
15:14	STOPBIT	stop bit长度选择

		00: 1-bit 01: 1.5-bit 10: 2-bit 11: reserved 注意: Mode0 时虽然没有 Stop Bit, 但仍需把 STOPBIT[1:0]保持为 2' b00;
13	PEIE	奇偶校验错误中断使能位 0: 奇偶校验错误中断关闭 1: 奇偶校验错误中断打开 数据接收标志产生于数据停止位接收完成, 硬件奇偶校验与数据接收中断由于停止位设置不同, 奇偶校验错误中断会提前于数据接收中断。使用此中断使能时请注意。 建议如下: 方法1: 关闭PEIE中断使能, 接收数据中断后软件判断ISR. PE奇偶校验是否正确。 方法2: 关闭PEIE中断使能, 接收数据中断通过接收SBUF. BIT8软件判断奇偶校验是否正确。
12:10	Reserved	保留
9	OVER	Mode0: 无效 Mode1/3: 0: 16采样分频; 1: 8采样分频; Mode2: 0: 32采样分频; 1: 16采样分频;
8	TXEIE	TX空中断使能位 0: TX Buffer空中断关闭 1: TX Buffer空中断打开
7:6	SM	工作模式 00: mode0 01: mode1 10: mode2 11: mode3
5	ADRDET	多机通讯地址自动识别使能位 0: 关闭 1: 打开
4	REN	Mode0: 0: 发送; 1: 接收; Mode1/2/3: 0: 发送; 1: 接收/发送;
3:2	B8CONT	Bit8数据控制位 00: 由软件读写SBUF[8]来决定 01: 硬件偶校验 10: 硬件奇校验 11: 保留;
1	TCIE	发送中断使能位 0: 发送中断关闭 1: 发送中断打开
0	RCIE	接收中断使能位 0: 接收中断关闭 1: 接收中断打开

20.11.3 地址寄存器（UARTx_SADDR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADDR							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	SADDR	从机设备地址

20.11.4 地址掩码寄存器（UARTx_SADEN）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADEN							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	SADEN	从机设备地址掩码

20.11.5 标志位寄存器 (UARTx_ISR)

偏移地址: 0x10

复位值: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
RSV		CTS		CTSIF		PE		TXE		FE		TC		RC									
		RO		RO		RO		RO		RO		RO		RO									

位	符号	描述
31:7	Reserved	保留
6	CTS	CTS信号标志位; 硬件置1, 硬件清零。 0: CTS信号为低电平 1: CTS信号为高电平
5	CTSIF	CTS中断标志位; 硬件置1, 软件清零。 0: CTS信号没有发生反转 1: CTS信号发生反转
4	PE	奇偶校验错误标志位; 硬件置1, 软件清零。 0: 无奇偶校验错误 1: 奇偶校验错误
3	TXE	Tx Buffer空标志位; 硬件置1, 硬件清零。 0: Tx Buffer非空 1: Tx Buffer空
2	FE	帧错误标志位; 硬件置1, 软件清零。
1	TC	发送完毕中断标志位; 硬件置1, 软件清零。 0: 发送未完成 1: 发送完毕
0	RC	接收完毕中断标志位; 硬件置1, 软件清零。 0: 接收未完成 1: 接收完成

20.11.6 标志位清除寄存器 (UARTx_ICR)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8																	
Reserved																																								
7							6							5							4			3			2			1			0							
Reserved							CTSIFCF							PECF							RSV			FECF							TCCF							RCCF		
							R1W0							R1W0										R1W0							R1W0							R1W0		

21. 低功耗同步异步收发器（LPUART）

21.1 概述

LPUART 模块（LPUART1/2）支持以下基本功能：

- 配置时钟 PCLK
- 传输时钟 SCLK（SCLK 可选择 XTL、RCL 以及 PCLK）
- 系统低功耗模式下收发数据
- 半双工和全双工传输
- 单线半双工传输
- 8/9-Bit 传输数据长度
- 硬件奇偶校验
- 1/1.5/2-Bit 停止位
- 四种不同传输模式
- 16-Bit 波特率计数器
- 多机通讯
- 硬件地址识别
- DMAC 硬件传输握手
- 硬件流控
- 支持单线模式

21.2 功能框图

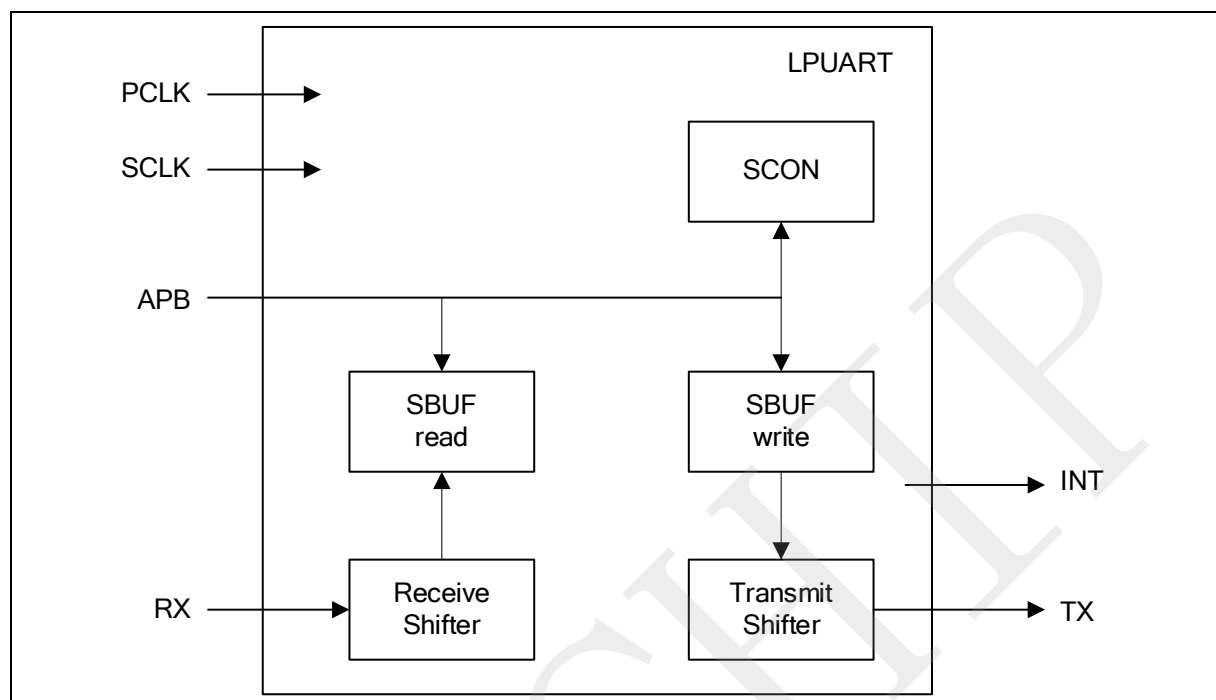


图 21-1 结构框图

21.3 配置时钟和传输时钟

LPUART 模块有两个时钟：配置时钟 PCLK 和传输时钟 SCLK。

21.3.1 配置时钟

配置时钟（PCLK）用于系统 APB 总线对 LPUART 模块进行寄存器配置，配置时钟固定为 APB 总线时钟 PCLK。当系统进入深度睡眠（DeepSleep）模式，PCLK 时钟将会停止。

21.3.2 传输时钟

传输时钟（SCLK）用于 LPUART 数据收发逻辑工作，可选择外部低速晶振时钟（XTL）、内部低速 RC 时钟（RCL）以及 PCLK 时钟。

当系统进入深度睡眠（DeepSleep）模式，如果 SCLK 选择外部低速晶振时钟（XTL）或者内部低速 RC 时钟（RCL）。LPUART 仍旧可以进行正常的数据收发，而不受系统深度睡眠（DeepSleep）模式的影响。

21.4 工作模式

21.4.1 Mode0（同步模式，半双工）

当工作在 Mode0 时，LPUART 为同步传输模式，其波特率固定为 SCLK 时钟频率的 1/12。LPUART 收发数据都由 RXD 输入输出，RXD 此时为双向端口。同步移位时钟由 TXD 输出，TXD 此时为输出端口。注意，本模式只能作为主机发送 LPUART 同步移位时钟，不可以作为从机接收外部输入的 LPUART 同步移位时钟。Mode0 时，传输数据位宽为 8-Bit，没有起始位和结束位。

当 LPUARTx_SCON.SM[1:0]=2'b00，可进入 Mode0 工作模式。

21.4.1.1 发送数据

发送数据时，清除 LPUARTx_SCON.REN 位，并将数据写入 LPUARTx_SBUF 寄存器中。此时，发送数据将从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

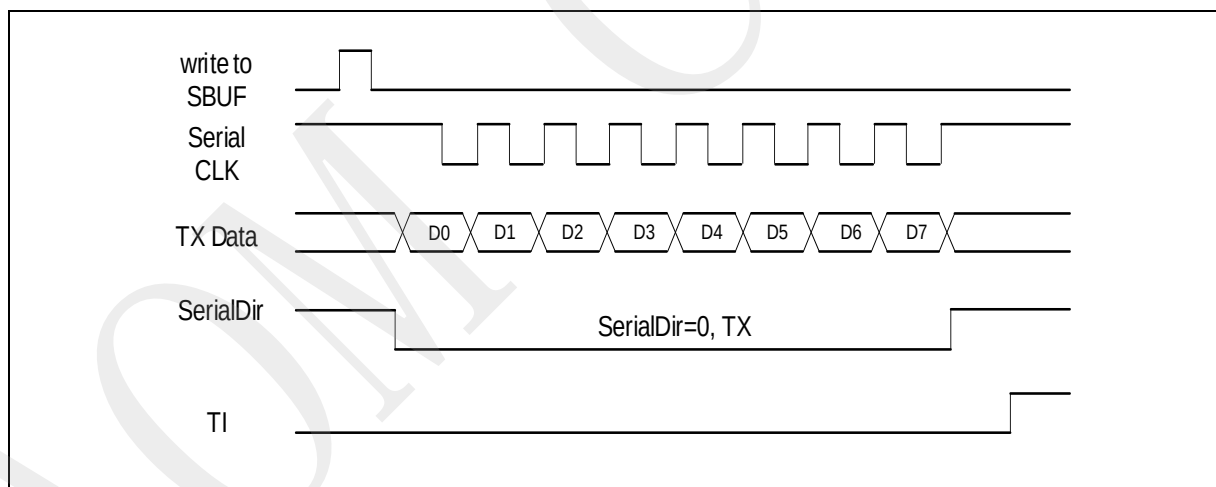


图 21-2 Mode0 发送数据

21.4.1.2 接收数据

接收数据时，将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清零。当接收结束，数据可从 LPUARTx_SBUF 寄存器读取。此时，接收数据从 RXD 输入（低位在先，高位在后），同步移位时钟从 TXD 输出。

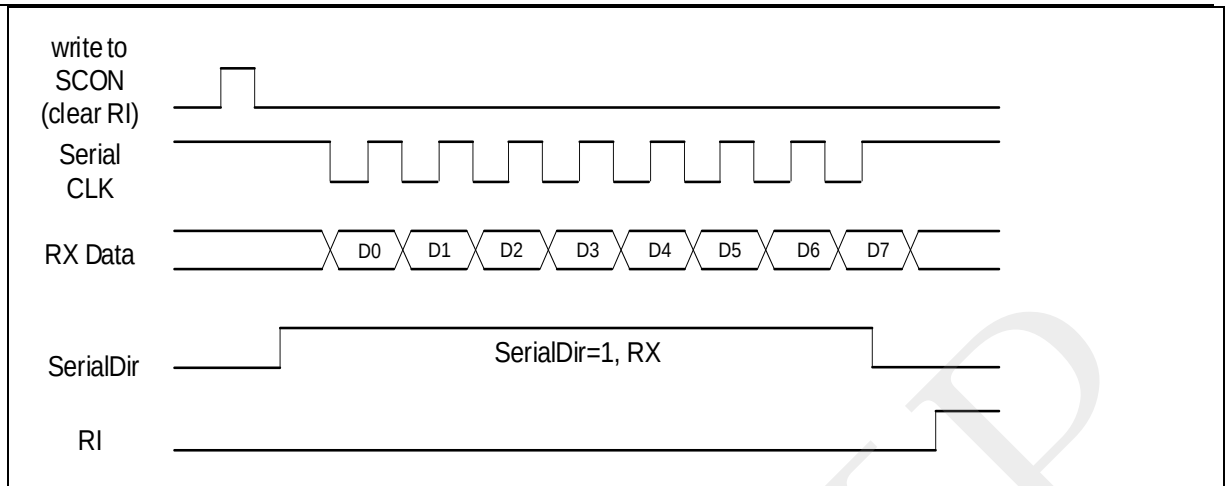


图 21-3 Mode0 接收数据

21.4.2 Mode1（异步模式，全双工）

当工作在 Mode1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据帧由以下比特位组成：1-Bit 起始位“0”开始，紧接着 8-Bit 数据位（低位在先，高位在后），最后是 1/1.5/2-Bit 结束位“1”。结束位长度由 LPUARTx_SCON.STOPBIT[1:0] 选择。

该模式下，波特率由内部 16-Bit 可编程波特率发生器产生。波特率计算方式请见后面波特率编程章节。

当 LPUARTx_SCON.SM[1:0]=2'b01 时，可进入 Mode1 工作模式。

21.4.2.1 发送数据

发送数据时，与 LPUARTx_SCON.REN 的值无关，将所发送数据写入 LPUARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

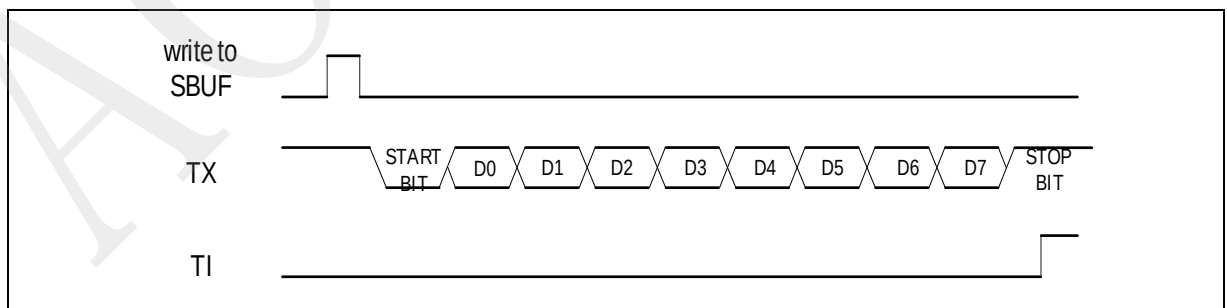


图 21-4 Mode1 发送数据

21.4.2.2 接收数据

接收数据时，将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清 0。开

始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 LPUARTx_SBUF 寄存器读出。

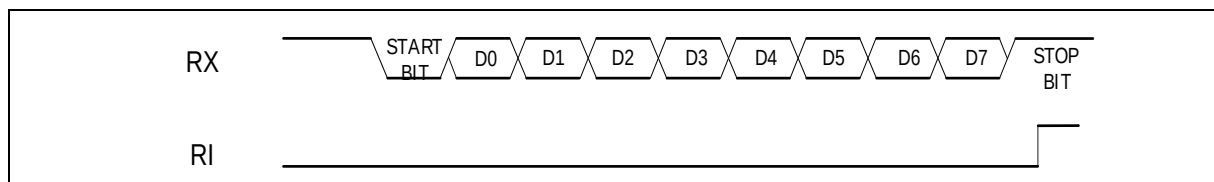


图 21-5 Mode1 接收数据

21.4.3 Mode2（异步模式，全双工）

当工作在 Mode2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由以下比特位组成：1-Bit 起始位“0”开始，接着是 8-Bit 数据位，1-Bit TB8 位和 1/1.5/2-Bit 结束位。结束位长度由 LPUARTx_SCON.STOPBIT[1:0]选择。额外的 TB8 位用于在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位或者数据位来使用。该模式下，波特率为传输时钟 SCLK 的固定分频，波特率计算方式请见后面波特率编程章节。

当 LPUARTx_SCON.SM[1:0]=2'b10，可进入 Mode2 工作模式。

21.4.3.1 发送数据

发送数据时，与 LPUARTx_SCON.REN 的值无关，将所发送数据写入 LPUARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

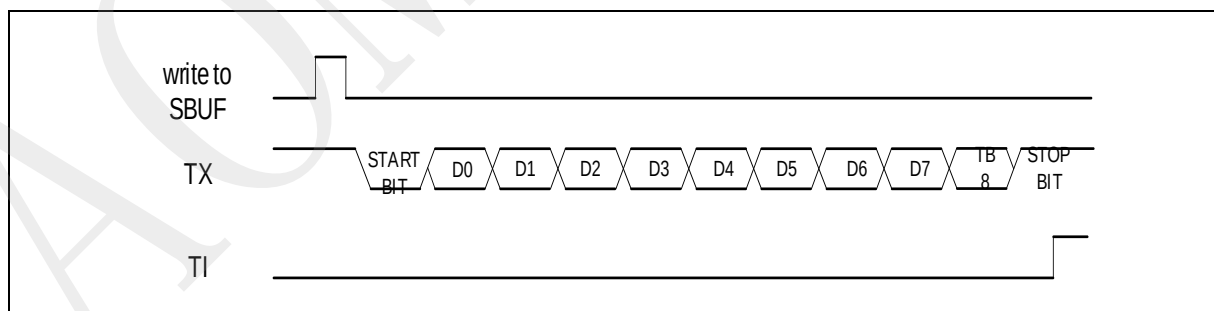


图 21-6 Mode2 发送数据

21.4.3.2 接收数据

接收数据时，需将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清 0。

开始接收 RXD 上的数据（低位在先，高位在后），当接收完毕，可以从 LPUARTx_SBUF 寄存器读取。

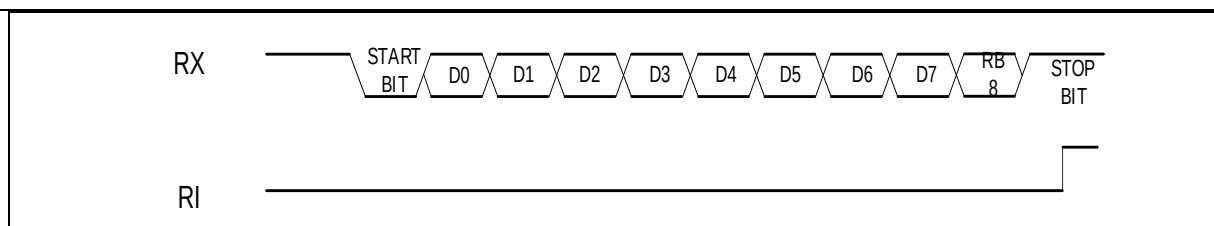


图 21-7 Mode2 接收数据

21.4.4 Mode3（异步模式，全双工）

当工作在 Mode3 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由以下比特位组成：1-Bit 起始位“0”开始，接着是 8-Bit 数据位，1-Bit TB8 位和 1/1.5/2-Bit 结束位。结束位长度由 LPUARTx_SCON.STOPBIT[1:0]选择。额外的 TB8 位用于在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位或者数据位来使用。该模式下，波特率由内部 16-Bit 可编程波特率发生器产生。波特率计算方式请见后面波特率编程章节。

当 LPUARTx_SCON.SM[1:0]=2'b11，可进入 Mode3 工作模式。

21.4.4.1 发送数据

发送数据时，与 LPUARTx_SCON.REN 的值无关，将所发送数据写入 LPUARTx_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

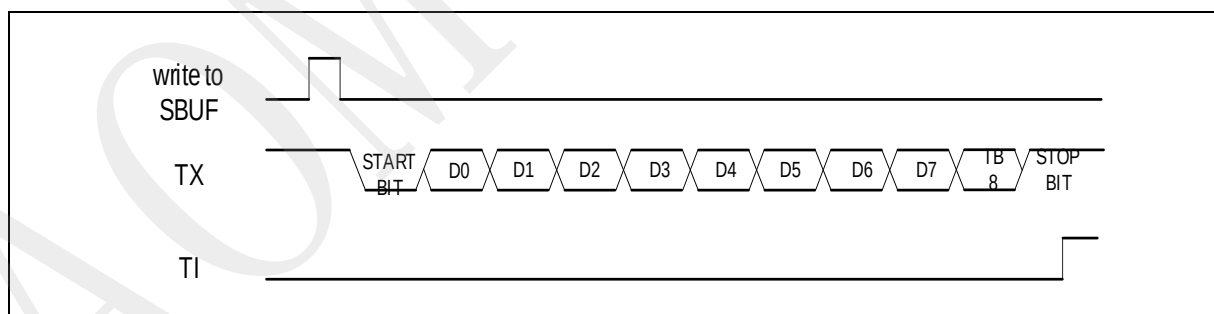


图 21-8 Mode3 发送数据

21.4.4.2 接收数据

接收数据时，将 LPUARTx_SCON.REN 位置 1，并将 LPUARTx_ISR.RC 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 LPUARTx_SBUF 寄存器读出。

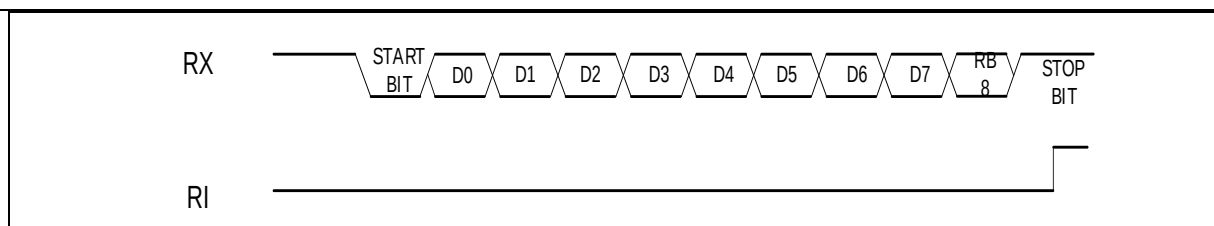


图 21-9 Mode3 接收数据

21.4.5 Single-wire Half-duplex（单线半双工）

当 LPUARTx_SCON.HDSEL=1'b1，可进入单线半双工传输模式。在该模式下，TX 与 RX 信号线在模块内部相连。RX 信号不再使用，接收发送的数据都经由 TX 信号完成。TX 信号的方向控制由硬件逻辑完成，无需软件控制。当发送缓存为空时，TX 信号始终为输入（接收状态）。当向发送缓存填入一个数据，TX 信号变为输出（发送状态）。当发送完成，发送缓存变为空，TX 信号又回到输入（接收状态）。

注意：

- 即使模块处于接收过程中，只要发送缓存被填入数据，TX 信号立即变为输出（发送状态）。接收过程将被打断，并得到一个错误的接收数据。

单线半双工模式并不是 Mode0 模式，虽然两者都是用一根信号分时复用地完成了数据的收发功能。但是 Mode0 为同步传输，有同步时钟。而单线半双工模式为异步传输，所有的波特率计算还是遵照 Mode1-2-3 的波特率公式，仅仅是收发功能复用到了 TX 这一根信号上面。

21.5 波特率编程

21.5.1 Mode0

当工作在 Mode0 时，LPUART 同步移位时钟频率固定为 SCLK 的 1/12。具体公式如下所示：

$$\text{BaudRate} = \frac{\text{Freq}}{12}$$

其中，Freq 为 SCLK 的时钟频率。

21.5.2 Mode1/3

当工作在 Mode1 或者 Mode3 时，波特率由可编程波特率发生器产生。具体公式如下

所示：

$$\text{BaudRate} = \frac{\text{Freq}}{\text{OVER} * \text{SCNT}}$$

其中，Freq 为 SCLK 的时钟频率；OVER 可以为 16、8 或者 4，由 LPUARTx_SCON.OVER 的值决定；SCNT 为 16-Bit 波特率计数器 LPUARTx_SCNT 的计数值。

21.5.3 Mode2

当工作在 Mode2 时，波特率为固定值，具体公式如下所示：

$$\text{BaudRate} = \frac{\text{Freq}}{\text{OVER}}$$

其中，Freq 为 SCLK 的时钟频率；OVER 可以为 32、16 或者 8，由 LPUARTx_SCON.OVER 的值决定。

21.6 传输数据结构

在不同工作模式下，传输数据的组成有所不同，请参考以下表格：

工作模式	传输位宽	数据组成
Mode0	8-Bit	<u>8-Bit Data</u>
Mode1	10-Bit	<u>1-Bit Start Bit</u> + <u>8-Bit Data</u> + <u>(1/1.5/2)-Bit Stop Bit</u>
Mode2	11-Bit	<u>1-Bit Start Bit</u> + <u>8-Bit Data</u> + <u>1-Bit B8</u> + <u>(1/1.5/2)-Bit Stop Bit</u>
Mode3	11-Bit	<u>1-Bit Start Bit</u> + <u>8-Bit Data</u> + <u>1-Bit B8</u> + <u>(1/1.5/2)-Bit Stop Bit</u>

表 21-1 传输数据组成结构

B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

应用场景	LPUARTx_SCON. ADDRDET	LPUARTx_SCON. B8CONT[1:0]	B8 数据含义
奇偶校验	--	01/10	接收时，B8 是所收到的 8-Bit 数据的奇偶校验位； 发送时，B8 是所发送的 8-Bit 数据的奇偶校验位；
多机通讯	1	--	B8=1，代表当前是地址帧； B8=0，代表当前是数据帧；

其他	0	00/11	接收/发送的 DATA[8]

表 21-2 B8 数据含义

注意：

- 当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制。

21.7 帧错误检测

当工作在 Mode1/2/3 时，LPUART 具有帧错误检测功能，硬件会自动检测接收到的帧数据是否带有效的 Stop 位。如果没有收到有效 Stop 位，则 LPUARTx_ISR.FE 置 1。LPUARTx_ISR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有效 Stop 位，也不会把 LPUARTx_ISR.FE 标志清 0。

21.8 多机通讯

当工作在 Mode2/3 时，可以打开多机通讯功能，为此在其帧格式中增加了 1 位 TB8/RB8。将 LPUARTx_SCON.ADRDET 位置“1”，可开启多机通讯功能。

当开启多机通讯后，主机可以通过 LPUARTx_SBUF[8]来区分当前的发送帧是地址帧（LPUARTx_SBUF[8]=1）还是数据帧（LPUARTx_SBUF[8]=0）。从机会忽略 RB8 位（接收到的 Bit8 数据位）为“0”的接收帧，该帧数据不会存入到从机的 LPUARTx_SBUF 寄存器中，从机也不会产生接收中断。当接收帧的 RB8 位（接收到的 Bit8 数据位）为“1”表明其是地址帧，从机会继续判断接收到的地址与其自身地址是否相等。如果匹配，则从机会对 LPUARTx_SBUF[8]置“1”，同时将该地址帧存入到 LPUARTx_SBUF 寄存器中。并对 LPUARTx_ISR.RC 置“1”，以表明该帧为地址帧并且地址已经匹配。从机软件看到 LPUARTx_SBUF[8]=1 并且 LPUARTx_ISR.RC=1 后，先把 LPUARTx_SCON.ADEDET 位清“0”，然后准备接受给它的数据帧。如果地址不相等，表明主机并不是寻址该从机，从机硬件保持 LPUARTx_SBUF[8] 和 LPUARTx_ISR.RC 为“0”，软件保持 LPUARTx_SCON.ADRDET 位为“1”，从机继续处于地址监听状态。

21.8.1 自动地址识别

当开启多机通讯后（LPUARTx_SCON.ADRDET 位置“1”），自动地址识别功能也将开

启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收从机会给出 LPUARTx_ISR.RC 接收标志。如果地址不匹配，则接收从机不会给出任何接收标志。

如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 Stop 位代替。当从机接收到匹配的地址帧和有效的 Stop 位时，LPUARTx_ISR.RC 会被置“1”。

21.8.2 给定地址

LPUART 设备的 LPUARTx_SADDR 寄存器用来表示自己的设备给定地址，LPUARTx_SADEN 寄存器是地址掩码，可以用来定义地址中的无关位。当 LPUARTx_SADEN 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，LPUARTx_SADEN 寄存器必须设为 8'hFF。给定地址公式如下所示：

$$\text{GivenAddr} = \text{SADDR} \& \text{SADEN}$$

21.8.3 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 8'hFF。

$$\text{BroadCastAddr} = \text{SADDR} | \text{SADEN}$$

21.8.4 举例

假设某从机的 LPUARTx_SADDR 和 LPUARTx_SADEN 配置如下：

SADDR: 8'b01101001

SADEN: 8'b11111011

那么其给定地址和广播地址如下：

Given: 8'b01101x01

Broadcast: 8'b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

8'b01101001 和 8'b01101101 (given address)

8'b11111011 和 8'b11111111 (broadcast address)。

21.9 DMAC 硬件握手

LPUART 模块支持 DMAC 的硬件握手逻辑。

将 LPUARTx_SCON.DMACTXEN 设置为 1，可以打开 LPUART TX 的 DMAC 硬件握手逻辑。当发送缓存为空时，LPUART 会向 DMAC 发出数据搬运请求 TX REQ。DMAC 收到该 TX REQ 信号，则从 DMAC 源地址搬运一个帧的发送数据到 LPUARTx_SBUF 中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

将 LPUARTx_SCON.DMACRXEN 设置为 1，可以打开 LPUART RX 的 DMAC 硬件握手逻辑。当一帧接收完成，LPUART 会向 DMAC 发出数据搬运请求 RX REQ。DMAC 收到该 RX REQ 信号，则从 LPUARTx_SBUF 中把接收数据搬运至 DMAC 的目标地址中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

21.10 硬件流控

通过增加 nCTS 和 nRTS 信号可以实现 LPUART 硬件流控的功能，即 LPUART 硬件模块根据 nCTS 和 nRTS 的高低电平自动控制数据的收发，而无需通过软件来判断。

两个 LPUART 模块之间的硬件流控示意图如下所示：

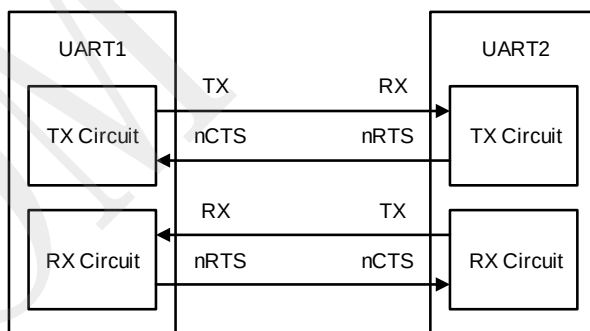


图 21-10 LPUART 硬件流控

21.10.1 nRTS 流控

当 nRTS 流控使能时（LPUARTx_SCON.RTSEN 设置为 1），当 LPUART 接收缓存空时，nRTS 信号拉低（低表示可以接收数据），通知对端设备可以发送下一帧数据。当接收缓存满时，nRTS 信号拉高（高表示不可以接收数据），通知对端设备在当前发送帧传输完成后，暂停发送下一帧数据。

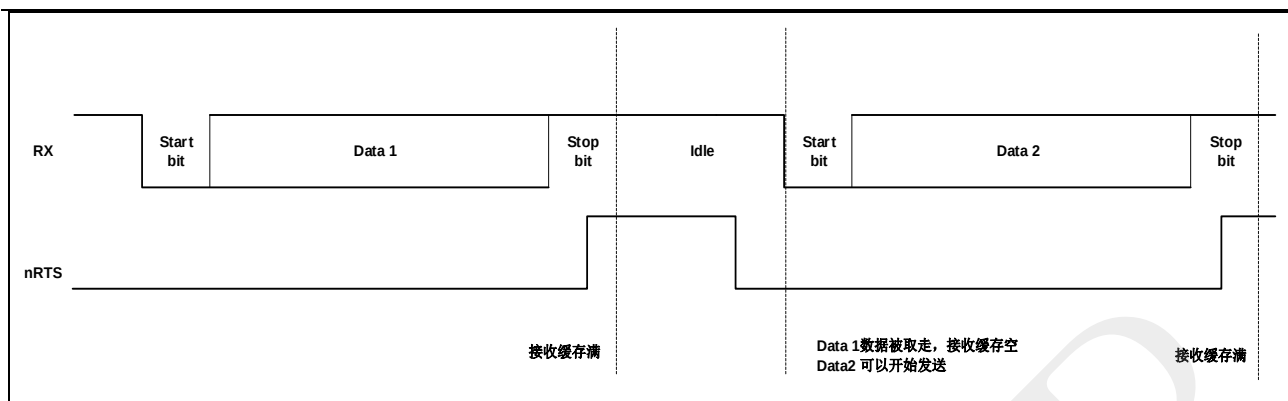


图 21-11 nRTS 硬件流控信号

21.10.2 CTS 流控

当 nCTS 流控使能时 (LPUARTx_SCON.CTSEN 设置为 1)，在 LPUART 发送下一帧数据之前，先判断 nCTS 的高低电平。如果 nCTS 为低电平（低电平表示对端设备可以接收下一帧数据），则 LPUART 发送下一帧数据。如果 nCTS 为高电平（高电平表示对端设备不可以接收下一帧数据），则 LPUART 暂停发送下一帧数据。nCTS 拉高不会影响当前发送帧的传输，只会暂停下一帧的发送，直到 nCTS 重新变低为止。

当 nCTS 流控使能时，一旦 nCTS 信号发生翻转，会把 LPUARTx_SFLAG.CTSIF 置 1。

如果使能 CTSIE 中断 (LPUARTx_SCON.CTSIE 设置为 1)，则会产生中断。同时，nCTS 信号的高低电平会记录在 LPUARTx_SFLAG.CTS 标志位中。

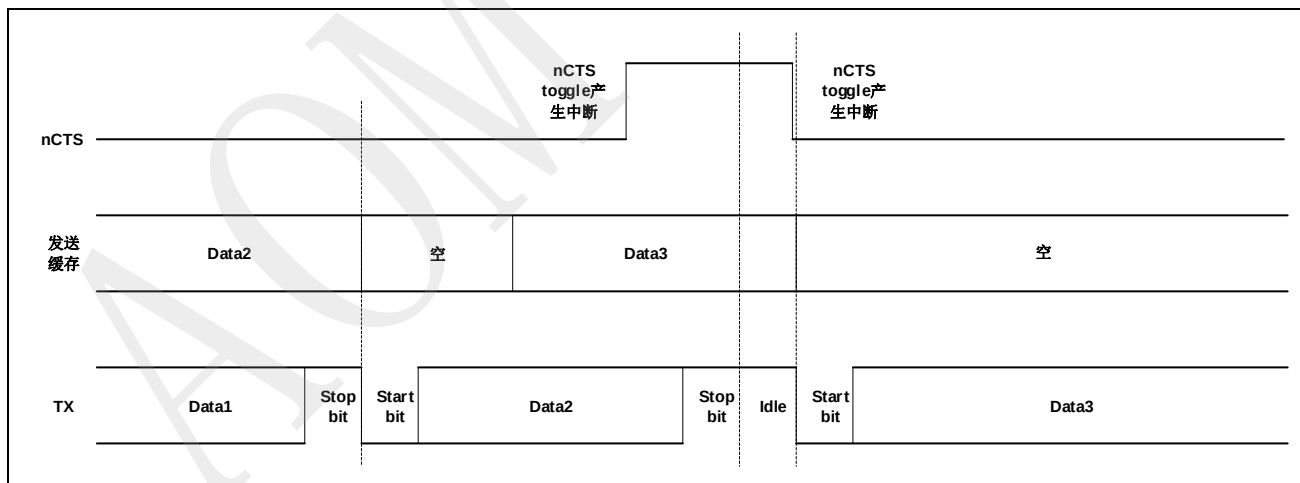


图 21-12 nCTS 硬件流控信号

21.11 收发端缓存

21.11.1 接收缓存

LPUART 模块接收端有一个帧（8/9-Bit）的接收缓存，也就是说当一帧数据接收完毕后，接收缓存中的数据会被一直保持，直到下一帧数据的 Stop 位接收完毕后，接收缓存才会更新到这一帧所收到的数据。

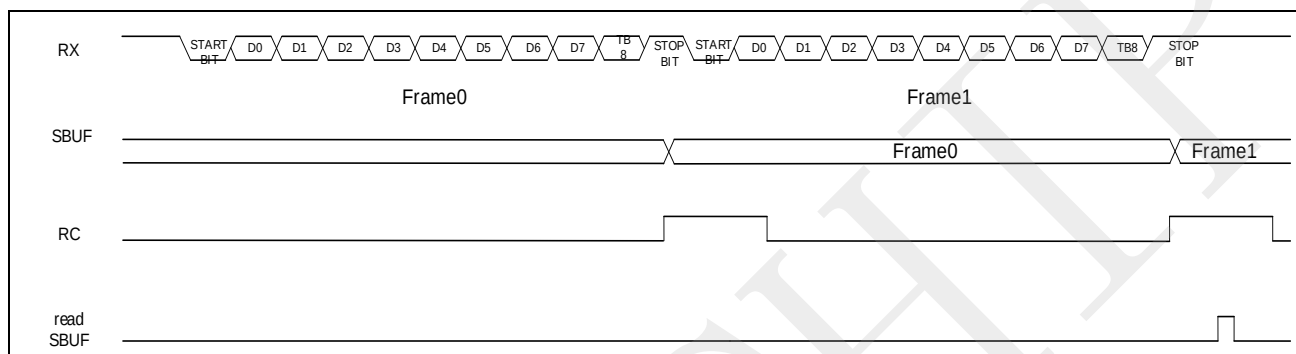


图 21-13 接收缓存

21.11.2 发送缓存

LPUART 模块发送端有一个帧(8/9-Bit)的发送缓存，当 LPUART 在发送当前帧时，软件可以对 LPUARTx_SBUF 写入下一个发送数据。LPUARTx_ISR.TXE 位表示了当前发送缓存的空满状态，当 LPUARTx_ISR.TXE=0 时，表明当前发送缓存满，软件不能对 LPUARTx_SBUF 写入下一个发送数据。如果软件继续写入 LPUARTx_SBUF，则该笔写入数据会硬件丢弃。当 LPUARTx_ISR.TXE=1 时，表明当前发送缓存空，软件可以写入下一个需要发送的数据帧。该笔发送数据会保存在发送缓存中，直到当前发送数据完成，硬件自动把发送缓存中的数据装载入移位寄存器中发送出去。

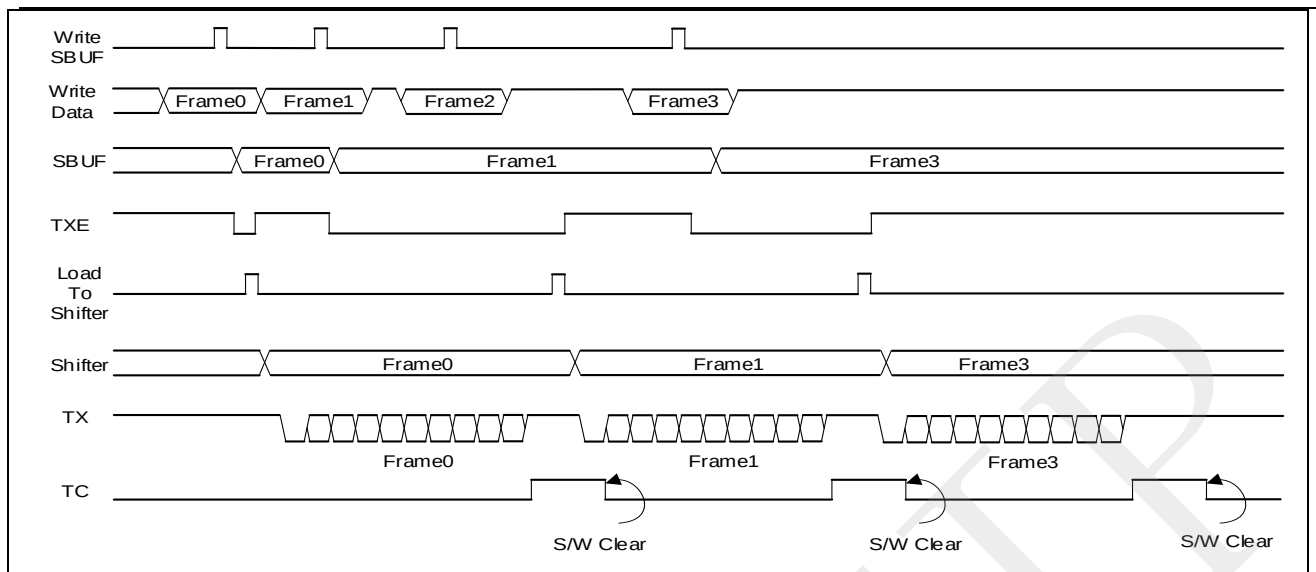


图 21-14 发送缓存

21.12 寄存器

LPUART1 基地址: 0x4001 4C00

LPUART2 基地址: 0x4001 5000

寄存器	偏移地址	描述
LPUARTx_SBUF	0x00	数据寄存器
LPUARTx_SCON	0x04	控制寄存器
LPUARTx_SADDR	0x08	地址寄存器
LPUARTx_SADEN	0x0C	地址掩码寄存器
LPUARTx_ISR	0x10	中断标志位寄存器
LPUARTx_ICR	0x14	中断标志位清除寄存器
LPUARTx_SCNT	0x18	波特率寄存器

表 21-3LPUART 寄存器列表

21.12.1数据寄存器（LPUARTx_SBUF）

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DATA[8]	DATA[7:0]							
								RW							

位	标记	功能描述
31:9	Reserved	保留
8	DATA[8]	在Mode0/1下，读取该位为0，写入该位无效； 在Mode2/3下，该位表示Bit8数据位，分以下两种情况： （1）当硬件奇偶校验位开启时，接收时该位为接收数据奇偶校验位，校验由硬件进行，如校验出错，校验错误标志位PE置1；发送时该位无效，发送数据奇偶校验位由硬件计算并发送； （2）当硬件奇偶校验位关闭时，接收时该位为接收数据Bit8；发送时该位为发送数据Bit8； 注意：当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受B8CONT控制；
7:0	DATA[7:0]	发送数据时，将发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读取。

21.12.2控制寄存器（LPUARTx_SCON）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RSV	HDSEL	FEIE	CTSIE	CTSEN	RTSEN	DMATXEN	DMARXEN
	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
STOPBIT		PEIE	SCLKSEL			OVER	TXEIE
RW		RW	RW			RW	RW
7	6	5	4	3	2	1	0
SM		ADRDET	REN	B8CONT		TCIE	RCIE
RW		RW	RW	RW		RW	RW

位	标记	功能描述
31:23	Reserved	保留
22	HDSEL	单线半双工模式使能 0：关闭单线半双工模式 1：打开单线半双工模式
21	FEIE	帧错误中断使能 0：关闭中断 1：打开中断
20	CTSIE	CTS信号翻转中断使能 0：关闭中断 1：打开中断

19	CTSEN	硬件flow control信号使能位 0: 关闭flow control信号
18	RTSEN	1: 打开flow control信号
17	DMATXEN	TX DMAC的硬件握手信号使能位 0: 关闭硬件握手信号 1: 打开硬件握手信号
16	DMARXEN	RX DMAC的硬件握手信号使能位 0: 关闭硬件握手信号 1: 打开硬件握手信号
15:14	STOPBIT	stop bit长度选择 00: 1-bit 01: 1.5-bit 10: 2-bit 11: reserved 注意: Mode0 时虽然没有 Stop Bit, 但仍需把 STOPBIT[1:0]保持为 2' b00;
13	PEIE	奇偶校验错误中断使能位; 0: 奇偶校验错误中断关闭; 1: 奇偶校验错误中断打开; 在系统时钟使用高速时钟与LPUART使用低速时钟情况下, 由于需要不同时钟域的同步, 硬件奇偶校验与数据接收中断由于停止位设置不同, 奇偶校验错误中断会提前或落后于数据接收中断。使用奇偶校验中断使能时请注意。 建议如下: 关闭PEIE中断使能, 接收数据中断后通过接收的数据位SBUF.BIT8软件判断是否出现奇偶校验错误。
12:11	SCLKSEL	传输时钟选择位: 00, 01: PCLK 10: XTL 11: RCL
10:9	OVER	Mode0: 无效; Mode1/3: 00: 16采样分频; 01: 8采样分频; 10: 4采样分频; 11: 保留; Mode2: 00: 32采样分频; 01: 16采样分频; 10: 8采样分频; 11: 保留;
8	TXEIE	TX空中断使能位 0: TX Buffer空中断关闭 1: TX Buffer空中断打开
7:6	SM	工作模式 00: mode0 01: mode1 10: mode2 11: mode3
5	ADRDET	多机通讯地址自动识别使能位 0: 关闭 1: 打开
4	REN	Mode0: 0: 发送; 1: 接收;

		Mode1/2/3: 0: 发送; 1: 接收/发送;
3:2	B8CONT	Bit8数据控制位; 00: 由软件读写SBUF[8]来决定; 01: 硬件偶校验; 10: 硬件奇校验; 11: 保留;
1	TCIE	发送中断使能位 0: 发送中断关闭 1: 发送中断打开
0	RCIE	接收中断使能位 0: 接收中断关闭 1: 接收中断打开

21.12.3地址寄存器（LPUARTx_SADDR）

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADDR							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	SADDR	从机设备地址

21.12.4地址掩码寄存器（LPUARTx_SADEN）

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADEN							
								RW							

位	标记	功能描述
31:8	Reserved	保留

7:0	SADEN	从机设备地址掩码

21.12.5 标志位寄存器 (LPUARTx_ISR)

偏移地址: 0x10

复位值: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
RSV		CTS		CTSIF		PE		TXE		FE		TC		RC									
		RO		RO		RO		RO		RO		RO		RO									

位	符号	功能描述
31:7	Reserved	保留
6	CTS	CTS信号标志位; 硬件置1, 硬件清零。 0: CTS信号为低电平 1: CTS信号为高电平
5	CTSIF	CTS中断标志位; 硬件置1, 软件清零。 0: CTS信号没有发生反转 1: CTS信号发生反转
4	PE	奇偶校验错误标志位; 硬件置1, 软件清零。 0: 无奇偶校验错误 1: 奇偶校验错误
3	TXE	Tx Buffer空标志位; 硬件置1, 硬件清零。 0: Tx Buffer非空 1: Tx Buffer空
2	FE	帧错误标志位; 0: 硬件置, 软件清零。
1	TC	发送完毕中断标志位; 硬件置1, 软件清零。 0: 发送未完成 1: 发送完毕
0	RC	接收完毕中断标志位; 硬件置1, 软件清零。 0: 接收未完成 1: 接收完成

21.12.6 标志位清除寄存器 (LPUARTx_ICR)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7				6			5		4		3		2		1		0						
Reserved						CTSIFCF		PECF		RSV		FECF		TCCF		RCCF							
						R1W0		R1W0				R1W0		R1W0		R1W0							

位	标记	功能描述
31:6	Reserved	保留
5	CTSIFCF	CTSIF标志清除位; 写0清除, 写1无效。
4	PECF	PE标志清除位; 写0清除, 写1无效。
3	Reserved	保留
2	FECF	FE标志清除位; 写0清除, 写1无效。
1	TCCF	TC标志清除位; 写0清除, 写1无效。
0	RCCF	RC标志清除位; 写0清除, 写1无效。

21.12.7 波特率寄存器 (LPUARTx_SCNT)

偏移地址: 0x18

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCNT															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	SCNT	波特率计数器

22. 控制器局域网（CAN）

22.1 简介

CAN（Controller Area Network）总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。本模块遵循 CAN 总线协议 2.0A 和 2.0B 协议并向上兼容 CAN-FD。CAN 总线控制器可以处理总线上的数据收发，在本产品中，CAN 具有 8 组筛选器。筛选器用于为应用程序选择要接收的消息。

应用程序通过 1 个高优先级的主发送缓冲器（Primary Transmit Buffer，以下简称 PTB）和 4 个辅发送缓冲器（Secondary Transmit Buffer，以下简称 STB）将发送数据送至总线，由发送调度器决定邮箱发送顺序。通过 10 个接收缓冲器（Receive Buffer，以下简称 RB）获取总线数据。4 个 STB 以及 10 个 RB 可以理解为一个 4 级 FIFO 和一个 10 级 FIFO，FIFO 完全由硬件控制。

CAN 总线控制器同时也可以支持时间触发 CAN 通信（Time-trigger communication）。

CAN 主要特性：

- 完全支持 CAN2.0A/CAN2.0B 协议。
- 向上兼容 CAN-FD 协议。
- 支持最高通信波特率 1Mbit/s
- 支持 1~1/256 的波特率预分频，灵活配置波特率。
- 10 个接收缓冲器
 - FIFO 方式
 - 错误或者不被接收的数据不会覆盖存储的消息
- 1 个高优先主发送缓冲器 PTB
- 4 个副发送缓冲器 STB
 - FIFO 方式
 - 优先级仲裁方式
- 8 组独立的筛选器
 - 支持 11 位标准 ID 和 29 位扩展 ID
 - 可编程 ID CODE 位以及 MASK 位

- PTB/STB 均支持支持单次发送模式
- 支持静默模式
- 支持回环模式
- 支持捕捉传输的错误种类以及定位仲裁失败位置
- 可编程的错误警告值
- 支持 ISO11898-4 规定时间触发 CAN 以及接收时间戳

22.2 CAN 系统框图

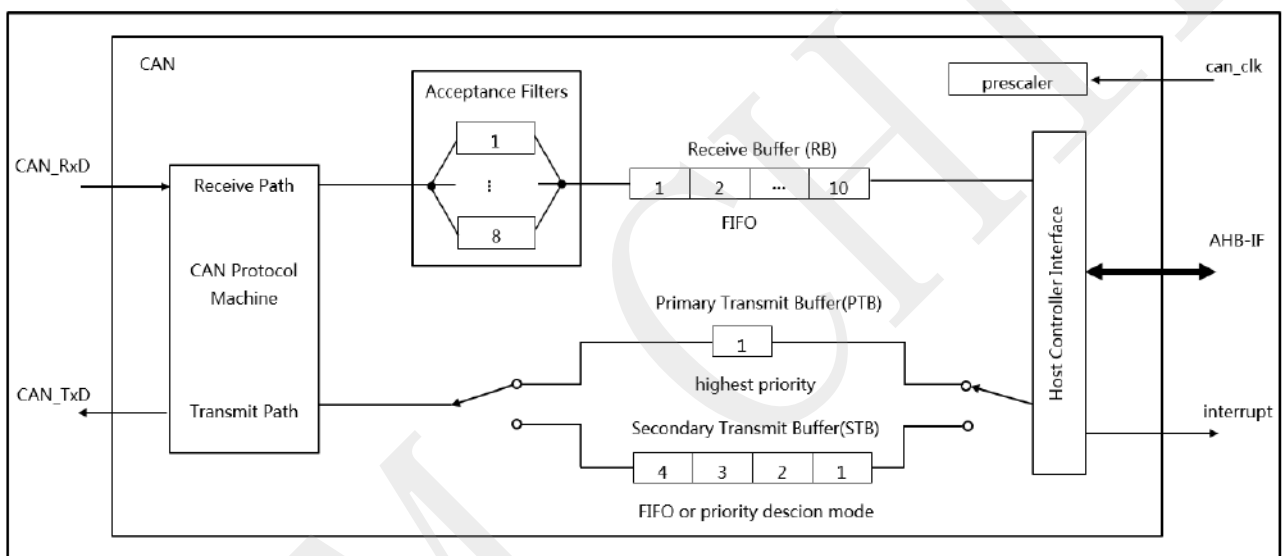


图 22-1CAN 系统框图

22.3 管脚说明

管脚名	方向	功能描述
CAN_RxD	输入	CAN接收数据信号
CAN_TxD	输出	CAN发送数据信号

表 22-1 CAN 管脚说明

22.4 功能说明

22.4.1 波特率设定

CAN 通信使用时钟 can_clk 的时钟源为外部高速振荡器，使用 CAN 模块之前，需要

在 CMU 章节设定 CAN 通信时钟。

下图给出 CAN 位时间定义图，虚线上部分为 CAN 协议规定的位时间，虚线下部分为本 CAN 控制器 CAN-CTRL 定义的位时间。其中 segment1 和 segment2 可以通过寄存器 BT 设定。BT 寄存器只能在 CFG_STAT.RESET=1 即 CAN 软件复位时设定。

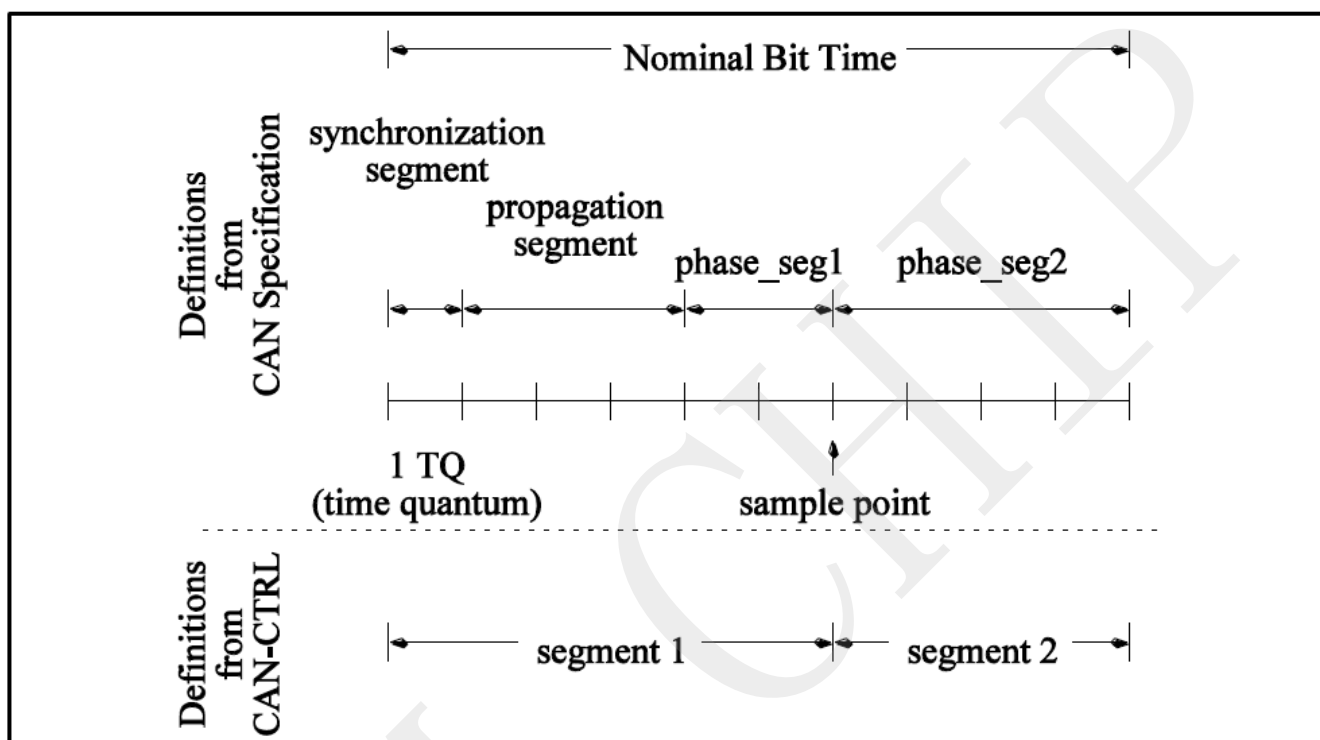


图 22-2 CAN 位时间定义图

TQ 计算方法请参考以下公式，其中 PRESC 通过 BT 寄存器的 PRESC 位设定。 f_{can_clk} 为 CAN 通信时钟频率。

$$TQ = \frac{PRESC + 1}{f_{can\_clk}}$$

位时间计算方法请参考以下公式，其中 SEG_1 和 SEG_2 通过 BT 寄存器的 SEG_1 位和 SEG_2 位设定。

$$BT = t_{SEG1} + t_{SEG2} = ((SEG\_1 + 2) + (SEG\_2 + 1)) \times TQ$$

位	设定范围	规则
BT寄存器的SEG_1位	0~63	SEG_1 ≥ SEG_2 + 1 SEG_2 ≥ SJW
BT寄存器的SEG_2位	0~7	
BT寄存器的SJW位	0~7	

表 22-2 CAN 位时间设定规则

22.4.2 发送缓冲器

CAN_CTRL 提供两种发送缓冲器用于发送数据，主发送数据缓冲器 PTB 和副发送缓冲器 STB。PTB 具有最高的优先级，但只能缓冲一帧数据。STB 优先级比 PTB 低，但可以缓冲 4 帧数据，且 STB 内 4 帧数据可以工作在 FIFO 模式或者优先级仲裁模式。

STB 中的 4 帧数据可以通过 TCMD 寄存器的 TSALL 位设定为 1 全部发送，在 FIFO 模式下，最先写入的数据先发送，在优先级模式下，ID 小的数据先发送。

PTB 中的数据具有最高优先级，所以 PTB 发送能推迟 STB 发送，但是已经赢得仲裁并开始发送的 STB 不能够被 PTB 发送推迟。

PTB 和 STB 中的一帧数据需占用 4 个字，可以通过 TBUF 寄存器进行访问。通过 TCMD 寄存器的 TBSEL 位选择 PTB 或者 STB，TBSEL=0，选择 PTB，TBSEL=1，选择 STB。通过 TCTRL 寄存器的 TSNEXT 位选择 STB 中的下一个 SLOT。对应关系如下图所示：

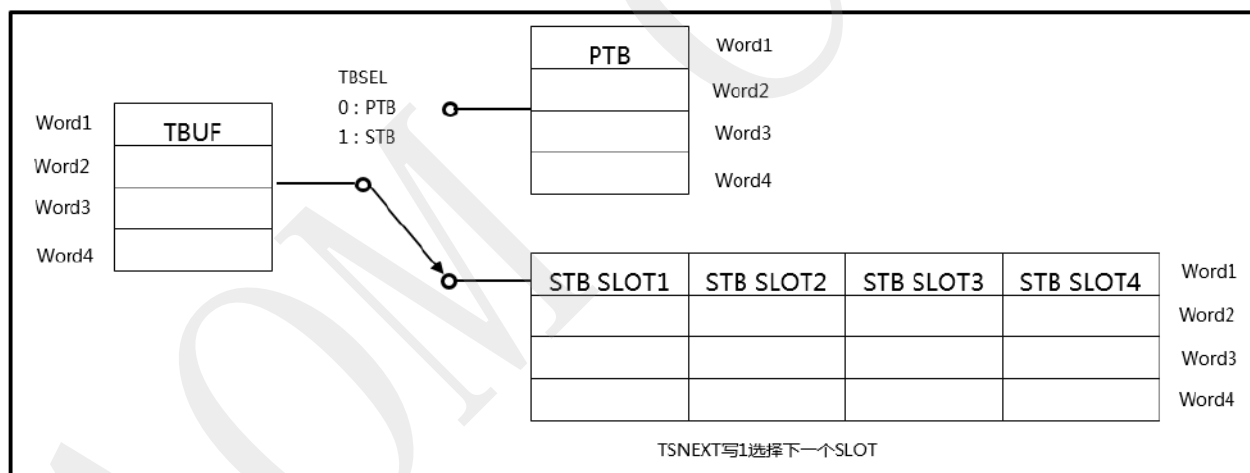


图 22-3 CAN TBUF 寄存器写发送缓冲器和示意图

22.4.3 接收缓冲器

CAN_CTRL 提供 10 个 SLOT 的接收缓冲器用于存储接收到的数据，该 10SLOT 的接收缓冲器工作在 FIFO 模式。每个 RB SLOT 需占用 4 个字，通过 RBUF 寄存器来读取接收到的数据，总是最先读取最早接收到数据，并通过 RCTRL 寄存器的 RREL 设置为 1 释放已经读取的 RB SLOT，并指向下一个 RB SLOT。

通过 RBUF 读取 RB SLOT 示意图如下。



图 22-4 CAN RBUF 寄存器读接收缓冲器示意图

22.4.4 接收筛选寄存器组

CAN_CTRL 提供 8 组 32 位筛选器用于过滤接收到的数据从而降低 CPU 负荷，筛选器可以支持标准格式 11 位 ID 或者扩展格式 29 位 ID。每组筛选器有一个 32 位 ID CODE 寄存器和一个 32 位 ID MASK 寄存器，ID CODE 寄存器用于比较接收到 CAN ID，而 ID MASK 寄存器用于选择比较的 CAN ID 位。对应的 ID MASK 位为 1 时，不比较该位的 ID CODE。

接收到的数据只要通过 8 组筛选器的任意一组，则被接收，接收到的数据存储在 RB 中，否则数据不被接收，也不被存储。

每组筛选器通过 ACFEN 寄存器使能或者禁止。ID CODE 和 ID MASK 通过 ACFCTRL 寄存器的 SELMASK 位设定，SELMASK=0 时，指向 ID CODE，SELMASK=1 时，指向 ID MASK。筛选器通过 ACFCTRL 寄存器的 ACFADR 位选择。ID CODE 和 ID MASK 通过 ACF 寄存器访问且只能在 CFG_STAT.RESET=1 即 CAN 软件复位时设定。ACF 寄存访问筛选寄存器组的方式请参考下图。

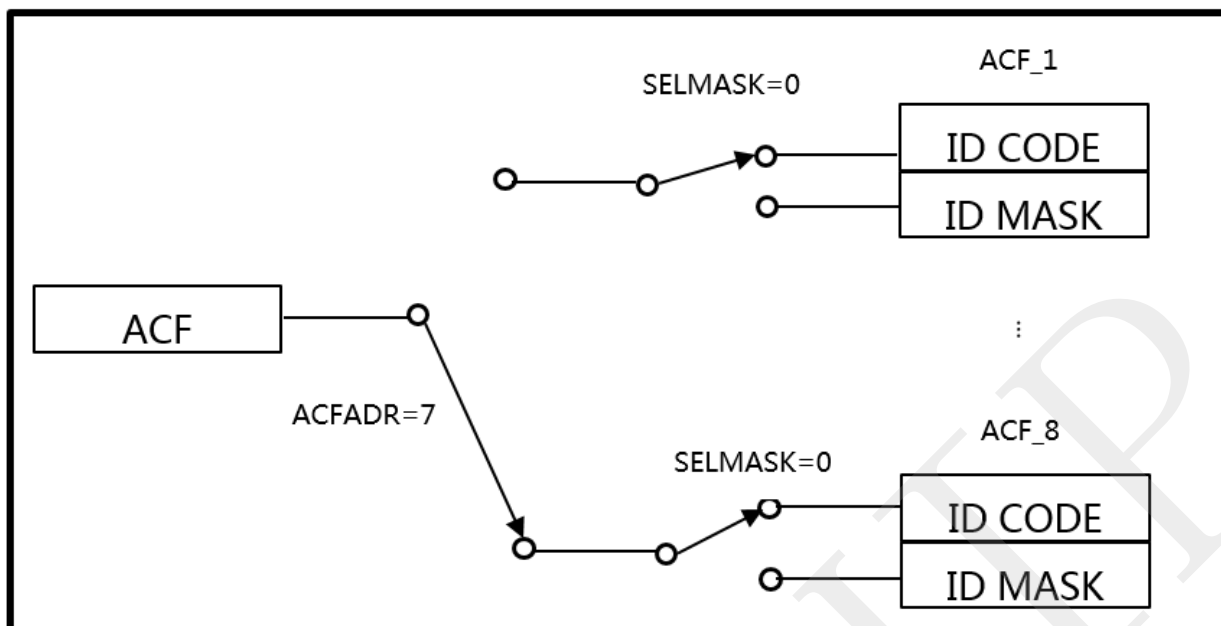


图 22-5 CAN ACF 寄存器访问筛选器组示意图

22.4.5 数据发送

在开始发送前必须保证 PTB 或者 STB 中至少有一帧数据已被装载，PTB 发送过程中 TPE 被锁定，STB 的填充情况可以通过 TSSTAT 位确认。发送数据设定步骤如下：

1. 设定 TBSEL 从 PTB 和 STB 中选择发送 BUF
2. 通过 TBUF 寄存器写需要发送的数据。
3. 如果选择的时 STB，设置 TSNEXT=1 以完成 STB SLOT 的装载。
4. 发送使能
 - PTB 发送使用 TPE
 - STB 发送使用 TSALL 或者 TSONE
5. 发送完成状态确认
 - PTB 发送完成使用 TPIF，TPIE 用于使能 TPIF
 - STB 采用 TSONE 发送完成时使用 TSIF，TSIE 用于使能 TSIF
 - STB 采用 TSALL 发送完成时使用 TSIF，此时需要设定的全部 STB SLOT 数据发送完成后，TSIF 才置位，TSIE 用于使能 TSIF

22.4.6 单次数据发送

不需要使用自动重新发送功能时，可以通过寄存器设定为单次发送模式，CFG_STAT

寄存器的 TPSS 位用于 PTB 的单次发送模式设定, TSSS 位用于 STB 的单次发送模式。数据成功发送时单次发送和正常发送模式时动作相同。但是数据没有成功发送时会出现以下结果:

- TPIF 置位 (TPIE=1), 对应的 BUF SLOT 数据会被清除。
- 有错误发送时, KOER 更新, BEIF 置位 (BEIE=1)。
- 仲裁失败, ALIF 置位 (ALIE=1)。

单次发送模式, 不能单独依靠 TPIF 来判断发送完成, 需要同 BEIF 和 ALIF 一起判断发送是否完成。

22.4.7 取消数据发送

可以通过 TPA 或者 TSA 取消已请求但还没有被执行的数据发送。取消数据发送会出现以下几种情况:

- 仲裁中
 - 节点仲裁失败, 则取消数据发送。
 - 节点仲裁成功, 则继续发送。
- 数据发送中
 - 成功发送数据且收到 ACK, 对应的标志和状态正常置位。数据发送不取消。
 - 成功发送数据但没有收到 ACK, 数据发送取消, 错误计数器增加。
 - TSALL=1 设定的发送数据, 正在发送的 STB SLOT 数据正常发送, 没有开始发送的 STB SLOT 被取消。

取消数据发送的结果有以下两种情况。

- TPA 释放 PTB, 且使 TPE=0。
- TSA 释放一个 STB SLOT 或者全部 STB SLOT 取决是 TSONE 还是 TSALL 使能的发送。

22.4.8 数据接收

接收筛选器组可以过滤掉不需要的接收数据, 减少中断的发生和 RB 的读取, 从而降低 CPU 负荷。接收数据设定步骤如下:

1. 设定筛选器组。

2. 设定 RFIE, RAIE 和 AFWL。
3. 等待 RFIF 或者 RAIF。
4. 通过 RBUF 从 RB FIFO 中读取最早接收到的数据。
5. 设置 RREL=1, 选择下一个 RB SLOT。
6. 重复 4, 5 直到通过 RSTAT 确认 RB 为空。

22.4.9 错误处理

CAN_CTRL 一方面可以自动处理部分错误, 比如自动重发数据或者丢弃接收到含有错误的帧, 另一方面通过中断将错误向 CPU 报告。

CAN 节点有以下三种错误状态:

- 错误主动: 节点检测到错误时自动发送主动错误标志。
- 错误被动: 节点检测到错误时自动发送被动错误标志。
- 节点关闭: 关闭状态下此节点不再影响整个 CAN 网络。

CAN_CTRL 提供 TECNT 和 RECNT 两个计数器用于计数错误。TECNT 和 RECNT 计数器按照 CAN2.0B 协议规定的规则进行增减。另外提供可编程的 CAN 错误警告 LIMIT 寄存器用于产生错误中断通知 CPU。

CAN 通信过程中有以下 5 种错误类型, 错误类型可以通过 EALCAP 寄存器的 KOER 位识别。

- 位错误
- 形式错误
- 填充错误
- 应答错误
- CRC 错误

22.4.10 节点关闭

当发送错误数大于 255 时, CAN 节点自动进入节点关闭状态从而不参与 CAN 通信, 直到返回到错误主动状态。可以通过 CFG_STAT 寄存器的 BUSOFF 位确认 CAN 节点关闭状态。BUSOFF 被置位的同时 EIF 中断产生。

CAN 从节点关闭状态状态恢复到错误主动状态有以下两种方法:

- 上电复位
- 接收到连续 128 个 11 位的隐形位序列（恢复序列）

节点关闭状态下，TECNT 值保持不变，RECNT 用于计数恢复序列。从节点关闭状态恢复后，TECNT 和 RECNT 被复位为 0。

节点关闭标志 BUSOFF 置位的同时，CFG_STAT 寄存器的 RESET 位也被置位。

22.4.11 仲裁失败位置捕捉

CAN_CTRL 能够精确捕捉到仲裁失败位的位置并反映到 ALC 寄存器中。ALC 寄存器中保存着最近一次仲裁失败位的位置，如果节点赢得仲裁，则 ALC 位不更新。

ALC 值定义如下：

SOF 位后，第一个 ID 数据位 ALC 为 0，第二个 ID 数据位 ALC 为 1，依次类推。因为仲裁只发生在仲裁场内，所以 ALC 的最大值为 31。比如一个标准格式远程帧和一个扩展帧仲裁，扩展帧在 IDE 位失败，则 ALC=12。

22.4.12 回环模式

CAN_CTRL 支持以下两种回环模式：

- 内部回环
- 外部回环

两种回环模式都可以接收自己发出的数据帧，主要用于测试用途。

内部回环模式，模块内部将接收数据线连接到发送数据线，并且发送数据不输出。

内部回环模式下，节点会生成自应答信号以避免 ACK 错误。

外部回环模式保持和收发器的连接因此发送的数据仍能出现在 CAN 总线上，在收发器的帮助下，CAN 能收到自己发送的数据。外部回环模式可以通过 RCTRL 寄存器的 SACK 位来决定是否生成自应答信号，SACK=0 时，不生成自应答信号，SACK=1 时，生成自应答信号。

外部回环模式，SACK=0 时，会出现以下两种情况：

- 其它节点也收到本节点发送的数据帧并发送应答信号，该情况下本节点能够成功收发数据。
- 如果没有其它节点返回应答信号，则会产生应答错误，会重新发送数据并增加错

误计数器。此时推荐采用单次发送模式。

从回环模式返回到正常模式时,除了清除模式位以外,还需要软件复位 CAN_CTRL。

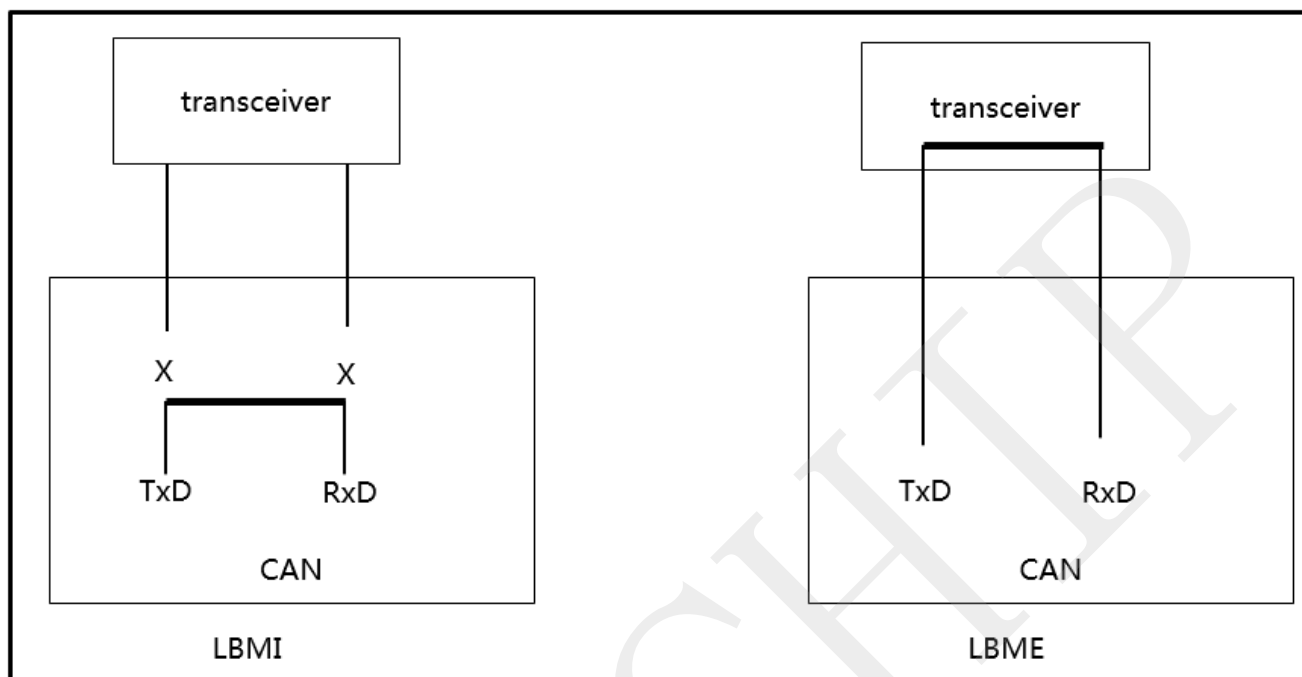


图 22-6 CAN LBMI 和 LBME 示意图

22.4.13 静默模式

静默模式可以用来监控 CAN 网络数据。在静默模式下,可以从 CAN 总线接收数据,不向总线发送任何数据。将 TCMD 寄存器中的 LOM 置 1,使 CAN 总线控制器进入静默模式,将其清 0 可以离开静默模式。

外部回环模式可以和静默模式组合成外部回环静默模式,此时 CAN 可以认为一个安静的接收者,但在有必要的时候可以发送数据。外部回环静默模式下,帧包含自应答信号允许被发送,但是该节点不会产生错误标志和过载帧。

22.4.14 软件复位功能

通过设定寄存器 CFG_STAT 寄存器的 RESET 位为 1,实现软件复位功能,软件复位功能的复位范围如下表所示。

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ACFADR	否	-	EWL	是	-
ACODE	否	只能在软件复位时可以写	KOER	是	-
AE_x	否	-	LBME	是	-
AFWL	否	-	LBMI	是	-
AIF	是	-	RACTIVE	是	接收立即停止，并不生成ACK
ALC	是	-	RAFIE	否	-
ALIE	否	-	RAFIF	是	-
ALIF	是	-	RBALL	是	-
AMASK	否	只能在软件复位时可以写	RBUF	是	所有RB被标记为空,数值不定
BEIE	否	-	REF_ID	否	-
BEIF	是	-	REF_IDE	否	-
BUSOFF	否	通过写1清除	RFIE	否	-
EIE	否	-	RFIF	是	-
EIF	否	-	RIE	否	-
EPASS	否	-	RIF	是	-
EPIE	否	-	ROIE	否	-
EPIF	是	-	ROIF	是	-
EWARN	否	否	ROM	否	-

寄存器位名	软件复位	备注	寄存器位名	软件复位	备注
ROV	是	-	TSNEXT	是	-
RREL	是	-	TSONE	是	-
PRESC	否	只能在软件复位时可以写	TPIE	否	-
RSTAT	是	-	TPIF	是	-
SACK	是	-	TPSS	是	-
SELMASK	否	-	TSFF	是	所有STB SLOT被标记为空
SEG_1	否	只能在软件复位时可以写	TSIE	否	-
SEG_2	否	只能在软件复位时可以写	TSIF	是	-
SJW	否	只能在软件复位时可以写	TSSS	是	-
TACTIVE	是	发送立即停止	TSSTAT	是	所有STB SLOT被标记为空
TBE	是	-	TTEN	是	-
TBF	否	-	TTIF	是	-
TBPTR	否	-	TTIE	否	-
TBSEL	是	-	TPPTR	否	-
TBUF	是	所有的STB被标记为空, 指向PTB	TTTBM	否	-
TECNT	否	通过BUSOFF=1清除	TTYPE	否	-
TEIF	是	-	TT_TRIG	否	-
TPA	是	-	TT_WTRIG	否	-
TPE	是	-	T_PRESC	否	-
TSA	是	-	WTIE	否	-
TSALL	是	-	WTIF	是	-
TSMODE	否	-			

表 22-3 软件复位范围表

22.4.15 向上兼容 CAN-FD 功能

CAN-CTRL 即使在包含 CAN-FD 网络中收到 CAN-FD 的帧, 接收器会自动忽略这些帧, 不返回 ACK, 等到总线空闲时, 再发送或者接收下一个 CAN2.0B 的帧。

22.4.16 时间触发 TTCAN

CAN-CTRL 为 ISO11898-4 规定的时间触发通信方式提供部分(lever 1)硬件支持。本章节从以下 5 个部分介绍 TTCAN 功能。

22.4.16.1 TTCAN 模式下的 TBUF 行为

TTTBM=1

TTTBM=1 时, PTB 和 STB SLOT 一样组成 TB SLOT, 通过 TBPTR 寄存器指定发送 BUF, 其中 TBPTR=0 时, 指向 PTB, TBPTR=1 是指向 STB SLOT1, 依次类推。主机可以通过 TPE 和 TPF 寄存器来标记发送 BUF SLOT。此时 TBSEL 和 TSNEXT 寄存器无任何意义从而可以被忽略。

TTTBM=1 时, PTB 不具有任何特殊的属性, 和 STB SLOT 一样, 传送完成标志也采用 TSIF。

TTCAN 模式时, 发送 BUF 没有 FIFO 模式和优先级仲裁模式, 同时也只有一个选定的 SLOT 可以发送数据。

TTCAN 模式下, 传输开始需要采用时间触发方式, TPE, TSONE, TSALL, TPSS 和 TPA 被固定为 0 且被忽略。

TTTBM=0

TTTBM=0 时, 组合使用事件驱动通信和接收时间戳功能。在该模式下, PTB 和 STB 的功能和 TTEN=0 时一致, 因此 PTB 始终具有最高的优先级, 而 STB 可以工作在 FIFO 模式或者仲裁模式。

22.4.16.2 TTCAN 功能描述

上电后, Time Master 需要根据 ISO 11898-4 协议进行初始化。一个 CAN 网络中, 最多可以有 8 个潜在的 Time Master。每一个 Time Master 都具有自己的参考消息 ID(ID 最后 3 位)。这些潜在的 Time Master 根据自己的优先级发送各自的参考消息。

TTEN=1 后, 16 位的计数器开始工作, 当参考消息被成功接收或者 Time Master 成功发送参考消息时, CAN 控制器将 Sync_Mark 拷贝给 Ref_Mark, Ref_Mark 将 cycle time 设置为 0。成功接收参考消息置位 RIF 标志而成功发送参考消息置位 TPIF 标志或者 TSIF 标志。此时主机需要准备下一个动作的触发条件。

触发条件可以是接收触发。该触发仅触发中断可用于检测期待的消息没有被收到。

触发条件也可以是发送触发。该触发开始发送通过 TTPTR 寄存器指定的 TBUF SLOT 里的数据。如果选定的 TBUF SLOT 被标记为空, 则不开始发送, 但置位中断标志。

22.4.16.3 TTCAN 时序

CAN_CTRL 支持 ISO11898-4 level 1。包含的一个 16 位计数器工作在为 PRESC, SEG_1, SET_2 定义的位时间下。如果 TTEN=1, 则有一个额外的预分频器 T_PRESC。一帧数据的 SOF 时, 计数器的值为 Sync_Mark。如果该帧数据为参考消息, 则将 Sync_Mark 拷贝给 Ref_Mark。cycle time 等于计数器的值减去 Ref_Mark。该时间用作接收消息的时间戳或者发送消息的触发时间基准。

22.4.16.4 TTCAN 触发方式

通过 TTYPE 寄存器定义 TTCAN 的触发方式, TTPTR 寄存器指定发送 SLOT, 而 TT_TRIG 指定触发器的 cycle time。

包含以下五种触发方式:

- 立即触发
- 时间触发
- 单次发送触发
- 发送开始触发
- 发送停止触发

除了立即触发方式外, 所有的触发器都使用 TTIF 标志。TTTBM=1 时, 只支持时间触发方式。

立即触发

通过写 TT_TRIG 的高位（不在意写入的值），启动触发器。此模式下，TTPTR 选定的 TBUF SLOT 内的数据会立即发送。TTIF 不置位。

时间触发

时间触发方式仅通过置位 TTIF 标志产生中断，并无其他功能。如果一个节点期待在特定的时间窗口内收到期待的数据，则可以使用时间触发方式。如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

单次发送触发

单次发送触发方式用于在执行时间窗口内发送数据。此时，忽略 TSSS 位。

通过 TEW 位设定 ISO11898-4 规定的最多 16 个 cycle time 的 Tick，设定范围为 1~16。

如果在规定的发送使能时间窗口内数据没有开始发送，则帧被丢弃。对应的发送 BUF SLOT 被标记为空，并且置位 AIF，对应的发送 BUF 内的数据不会被改写，因为可以通过置位 TPF 再次发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送开始触发

发送开始触发方式用于仲裁时间窗口内，参与仲裁。TSSS 用于决定是否自动重发或者单次发送模式。如果指定的消息没有被成功发送，可以使用发送停止触发来停止该发送。

如果 TT_TRIG 值小于实际的 cycle time，则 TEIF 置位且无其它动作。

发送停止触发

发送停止触发方式用于停止通过发送开始触发方式已经开始的发送。如果发送被停止，则发送帧被舍弃，置位 AIF 并将选定的 TBUF SLOT 标记为空，但 TBUF SLOT 内的数据不会被改写，可以通过置位 TPF 就可以再次发送。

如果 TT_TRIG 值小于实际的 cycle time 则 TEIF 置位且执行停止。

22.4.16.5 TTCAN 触发看门时间

TTCAN 触发看门时间功能类似于看门狗功能，在 TTTBM=1 时使用。用来看门从上次成功接收到参考消息开始的时间。参考消息可以在周期 cycle time 中或者一个事件后被接收，应用程序应该根据具体情况设定合适的看门时间。

如果 cycle count 等于 TT_WTRIG，则置位 WTIF。通过 WTIE 写 0，关闭看门触发。

如果 TT_WTRIG 比实际的 cycle time 小，则 TEIF 置位。

22.4.17 中断

中断标志	描述
RIF	接收中断
ROIF	接收上溢中断
ROIF	接收BUF满中断
RAFIF	接收BUF将满中断
TPIF	PTB发送中断
TSIF	STB发送中断
EIF	错误中断
AIF	取消发送中断
EPIE	错误被动中断
ALIF	仲裁失败中断
BEIF	总线错误中断
WTIF	触发看门中断
TEIF	触发错误中断
TTIF	时间触发中断

表 22-4 CAN 中断表

22.5 寄存器说明

基地址

CAN1 :0x4002_3800

CAN2 :0x4002_3C00

寄存器名	符号	偏移地址	位宽	复位值
CAN接收BUF寄存器	CAN_RBUF	0x00~0x0F	128	0xFFFF XXXX
CAN发送BUF寄存器	CAN_TBUF	0x50~0x5F	128	0xFFFF XXXX
CAN配置和状态寄存器	CAN_CFG_STAT	0xA0	8	0x80
CAN命令寄存器	CAN_TCMD	0xA1	8	0x00
CAN发送控制寄存器	CAN_TCTRL	0xA2	8	0x90
CAN接收控制寄存器	CAN_RCTRL	0xA3	8	0x00
CAN接收和发送中断使能寄存器	CAN_RTIE	0xA4	8	0xFE
CAN接收和发送中断标志寄存器	CAN_RTIF	0xA5	8	0x00
CAN错误中断使能和标志寄存器	CAN_ERRINT	0xA6	8	0x00
CAN警告限定寄存器	CAN_LIMIT	0xA7	8	0x1B
CAN位时序寄存器	CAN_BT	0xA8	32	0x0102 0203
CAN错误和仲裁失败捕捉寄存器	CAN_EALCAP	0xB0	8	0x00
CAN接收错误计数器寄存器	CAN_RECNT	0xB2	8	0x00
CAN发送错误计数器寄存器	CAN_TECNT	0xB3	8	0x00
CAN筛选器组控制寄存器	CAN_ACFCTRL	0xB4	8	0x00
CAN筛选器组使能寄存器	CAN_ACFEN	0xB6	8	0x01
CAN筛选器组code和mask寄存器	CAN_ACF	0xB8	32	0xFFFF XXXX
TTCAN TB slot指针寄存器	CAN_TB SLOT	0xBE	8	0x00
TTCAN 时间触发配置寄存器	CAN_TTCFG	0xBF	8	0x90
TTCAN 参考消息寄存器	CAN_REF_MSG	0xC0	32	0xFFFF XXXX
TTCAN 触发配置寄存器	CAN_TRG_CFG	0xC4	16	0x0000
TTCAN 触发时间寄存器	CAN_TRG_TRIG	0xC6	16	0x0000
TTCAN 触发看门时间寄存器	CAN_TRG_WTRIG	0xC8	16	0x0000

表 22-5 CAN 寄存器一览表

地址	BYTE访问	HALFWORD访问		WORD访问			
0x00~0x0F	CAN_RBUF	CAN_RBUF		CAN_RBUF			
0x50~0x5F	CAN_TBUF	CAN_TBUF		CAN_TBUF			
0xA0	CAN_CFG_STAT	CAN_TCMD	CAN_CFG_STAT	CAN_RCTRL	CAN_TCTRL	CAN_TCMD	CAN_CFG_STAT
0xA1	CAN_TCMD	-		-			
0xA2	CAN_TCTRL	CAN_RCTRL	CAN_TCTRL	-			
0xA3	CAN_RCTRL	-		-			
0xA4	CAN_RTIE	CAN_RTIF	CAN_RTIE	CAN_LIMIT	CAN_ERRINT	CAN_RTIF	CAN_RTIE
0xA5	CAN_RTIF	-		-			
0xA6	CAN_ERRINT	CAN_LIMIT	CAN_ERRINT	-			
0xA7	CAN_LIMIT	-		-			
0xA8	CAN_BT[7:0]	CAN_BT[15:0]		CAN_BT			
0xA9	CAN_BT[15:8]			-			
0xAA	CAN_BT[23:16]	-		-			
0xAB	CAN_BT[31:24]	CAN_BT[31:16]		-			
0xB0	-	-		CAN_TECNT	CAN_RECNT	-	
0xB1	-	-		-			
0xB2	CAN_RECNT	CAN_TECNT	CAN_RECNT	-			
0xB3	CAN_TECNT	-		-			
0xB4	CAN_ACFCTRL[7:0]	CAN_ACFCTRL		CAN_ACFEN		CAN_ACFCTRL	
0xB5	CAN_ACFCTRL[15:8]	-		-			
0xB6	CAN_ACFEN[7:0]	CAN_ACFEN		-			
0xB7	CAN_ACFEN[15:8]	-		-			
0xB8	CAN_ACF	CAN_ACF		CAN_ACF			
0xBC	-	-		CAN_TTCFG	CAN_TBSLOT	-	-
0xBD	-	-		-			
0xBE	CAN_TBSLOT	CAN_TTCFG	CAN_TBSLOT	-			
0xBF	CAN_TTCFG			-			
0xC0	CAN_REF_MSG[7:0]	CAN_REF_MSG[15:0]		CAN_REF_MSG			
0xC1	CAN_REF_MSG[15:8]	-		-			
0xC2	CAN_REF_MSG[23:16]	CAN_REF_MSG[31:16]		-			
0xC3	CAN_REF_MSG[31:24]	-		-			
0xC4	CAN_TRG_CFG[7:0]	CAN_TRG_CFG		CAN_TRG_TRIG		CAN_TRG_CFG	
0xC5	CAN_TRG_CFG[15:8]	-		-			
0xC6	CAN_TRG_TRIG[7:0]	CAN_TRG_TRIG		-			
0xC7	CAN_TRG_TRIG[15:8]	-		-			
0xC8	CAN_TRG_WTRIG[7:0]	CAN_TRG_WTRIG		-		CAN_TRG_WTRIG	
0xC9	CAN_TRG_WTRIG[15:8]	-		-			

表 22-6 CAN 寄存器 BYTE/HALFWORD/WORD 访问排布表

22.5.1 CAN 接收 BUF 寄存器 (CAN_RBUF)

CAN Receive Buffer Registers

偏移地址: 0x50

复位值: 0xFFFF XXXX

RBUF 寄存器指向最早接收到的 CAN 邮箱的 RB SLOT 地址, RBUF 寄存器可以按照任意顺序读取。

KOER 位即为寄存器 EALCAP.KOER, 仅在 RBALL=1 时有意义。

TX 位表示在回环模式下接受到自己发送的邮箱。

CYCLE_TIME 位仅在 TTCAN 模式时有效, 表示 SOF 开始时的 cycle time。

CAN 接收邮箱的数据格式如下:

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	-					ID[10:8]			ID
RBUF+2	-								ID
RBUF+3	-								ID
RBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX		-			Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
RBUF+11	DATA4								Data
RBUF+12	DATA5								Data
RBUF+13	DATA6								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 22-7 标准格式 CAN 接收邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
RBUF	ID[7:0]								ID
RBUF+1	ID[15:8]								ID
RBUF+2	ID[23:16]								ID
RBUF+3	-			ID[28:24]					ID
RBUF+4	IDE=1	RTR	0	0	DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	-				Status
RBUF+6	CYCLE_TIME[7:0]								TTCAN
RBUF+7	CYCLE_TIME[15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
RBUF+11	DATA4								Data
RBUF+12	DATA5								Data
RBUF+13	DATA6								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

表 22-8 扩展格式 CAN 接收邮箱格式

控制位含义如下：

IDE(Identifier Extension):

0: 标准格式

1: 扩展格式

RTR(Remote Transmission Request)

0: 数据帧

1: 远程帧

DLC(Data Length Code):

数据长度码，设定范围为 0~8，对应数据长度为 0Byte~8Byte

22.5.2 CAN 发送 BUF 寄存器 (CAN_TBUF)

CAN Transmit Buffer Registers

偏移地址: 0x00

复位值: 0xFFFF XXXX

TBUF 寄存器指向下一个空的 CAN 发送 BUF SLOT, TBUF 寄存器可以按照任意顺序读取。通过软件将 TSNEXT 写 1 来标记对应的 TBUF SLOT 已经写入数据, 从而指向下一个 TBUF SLOT。

TBUF 只能 WORD 访问。

CAN 发送邮箱的数据格式如下:

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	-					ID[10:8]			ID
TBUF+2	-								ID
TBUF+3	-								ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 22-9 标准格式 CAN 发送邮箱格式

地址	b7	b6	b5	b4	b3	b2	b1	b0	功能
TBUF	ID[7:0]								ID
TBUF+1	ID[15:8]								ID
TBUF+2	ID[23:16]								ID
TBUF+3	-			ID[28:24]					ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

表 22-10 扩展格式 CAN 发送邮箱格式

控制位含义如下：

IDE(IDentifier Extension):

0: 标准格式

1: 扩展格式

RTR(Remote Transmission Request)

0: 数据帧

1: 远程帧

DLC(Data Length Code):

数据长度码，设定范围为 0~8，对应数据长度为 0Byte~8Byte

22.5.3 CAN 配置和状态寄存器 (CAN_CFG_STAT)

CAN Configuration and Status Register

偏移地址: 0xA0

复位值: 0x80

7	6	5	4	3	2	1	0
RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF
R/W	R/W	R/W	R/W	R/W	R	R	R/W

位	符号	功能描述
7	RESET	复位请求位 0: 不请求局部复位 1: 请求局部复位 部分寄存器只能在RESET=1时进行写操作, 具体请参考软件复位功能, 当该节点进入BUS OFF状态时, 硬件自动将RESET位置1。请注意, 当RESET=0后需要11个CAN bit time该节点才能参与通信。
6	LBME	外部回环模式使能位 0: 禁止外部回环模式 1: 使能外部回环模式 注意: 通信中禁止设定该位。
5	LBMI	内部回环模式使能位 0: 禁止内部回环模式 1: 使能内部回环模式 注意: 通信中禁止设定该位。
4	TPSS	PTB单次传输模式 0: 禁止PTB单次传输模式 1: 使能PTB单次传输模式
3	TSSS	STB单次传输模式 0: 禁止STB单次传输模式 1: 使能STB单次传输模式
2	RACTIVE	接收中状态信号 0: 非接收中 1: 接收中
1	TACTIVE	发送中状态信号 0: 非发送中 1: 发送中
0	BUSOFF	总线关闭状态 0: 总线有效状态 1: 总线关闭状态 注意: 写1可以清零TECNT和RECENT寄存器, 但仅限用于调试用途。

22.5.4 CAN 命令寄存器（CAN_TCMD）

CAN Command Register

偏移地址：0xA1

复位值：0x00

7	6	5	4	3	2	1	0
TBSEL	LOM	RSV	TPE	TPA	TSONE	TSALL	TSA
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

位	符号	功能描述
7	TBSEL	发送BUF选择位（Transmit Buffer Select） 0：PTB 1：STB 当TTEN=1&TTBM=1时，TBSEL被复位成复位值。 注意：写TBUF寄存器或者TSNEXT位时，此位需要保持定值。
6	LOM	静默模式使能位（Listen Only Mode） 0：禁止静默模式 1：使能静默模式 LOM=1&LBME=0时禁止发送。 LOM=1&LBME=1时禁止应答相应接收到的帧以及错误帧，但可以发送数据。 注意：通信中禁止设定该位。
5	Reserved	必须保持复位值。
4	TPE	PTB发送使能位（Transmit Primary Enable） 0：禁止PTB发送 1：使能PTB发送 此位使能后，PTB中的Mailbox将在下一个可以发送的位置被发送。已经开始的STB发送将继续，但是下一个等待的STB发送会被延迟到PTB发送完成后再进行。 该位写1后将保持为1直到PTB发送完成或者通过TPA取消发送。软件不能通过写0清除该位。 以下情况TPE被硬件复位成复位值： - RESET=1 - BUSOFF=1 - LOM=1&LBME=0 TTEN=1&TTBM=1
3	TPA	PTB发送取消位（Transmit Primary Abort） 0：不取消 1：取消已经通过TPE置1请求但还未开始的PTB发送 该位软件写1但是通过硬件清零。通过写1可以清零TPE位，因此和TPE同时写1。 以下情况TPE被硬件复位成复位值： - RESET=1 - BUSOFF=1

		TTEN=1&TTTBM=1
2	TSONE	发送一帧STB数据设定位 (Transmit Secondary ONE frame) 0: 不发送 1: 发送一帧STB数据 FIFO模式中, 发送最早写入的数据, 优先级模式里发送最高优先级的数据 该位写1后将保持为1直到STB发送完成或者通过TSA取消发送。软件不能通过写0清除该位。 以下情况TSONE被硬件复位成复位值: - RESET=1 - BUFOFF=1 - LOM=1&LBME=0 TTEN=1&TTTBM=1
1	TSALL	发送所有的STB数据设定位 (Transmit Secondary ALL frame) 0: 不发送 1: 发送STB中所有的数据 该位写1后将保持为1直到STB发送完成或者通过TSA取消发送。软件不能通过写0清除该位。 以下情况TSALL被硬件复位成复位值: - RESET=1 - BUSOFF=1 - LOM=1&LBME=0 TTEN=1&TTTBM=1
0	TSA	STB发送取消位 (Transmit Secondary Abort) 0: 不取消 1: 取消已经通过TSONE或者TSALL置1请求但还未开始的STB发送 该位通过软件写1但是通过硬件清零。写1可以清零TSONE或者TSALL位。 以下情况TSA被硬件复位成复位值: - RESET=1 - BUSOFF=1

22.5.5 CAN 发送控制寄存器 (CAN_TCTRL)

CAN Transmit Control Register

偏移地址: 0xA2

复位值: 0x90

7	6	5	4	3	2	1	0
RSV	TSNEXT	TSMODE	TTTBM	Reserved		TSSTAT[1:0]	
R	R/W	R/W	R/W	R		R	

位	符号	功能描述
7	Reserved	必须保持复位值。
6	TSNEXT	下一个STB (Transmit buffer Secondary NEXT) 0: 无动作 1: 当前STB SLOT已填充, 指向下一个SLOT 应用程序将TBUF中的数据写完后, 通过置位TSNEXT位标识当前STB SLOT已经被填充, 从而硬件将TBUF指向下一个STB SLOT。 被TSNEXT位标识的STB SLOT中的数据可以通过TSONE或者TSALL位发送。 该位通过应用程序写1, 硬件清零。 所有的STB SLOT被填满后, TSNEXT保持为1直到有STB SLOT被释放。 注意: TTCAN模式时此位固定为0。
5	TSMODE	STB发送模式 (Transmit buffer Secondary operation MODE) 0: FIFO模式 1: 优先级模式 FIFO模式根据数据帧写入的先后顺序发送。 优先级模式根据ID自动判断, ID越小, 优先级越高。 无论何种模式, PTB具有最高的优先级。 注意: TSMODE位只能在STB空时设定。
4	TTTBM	TTCAN BUF模式 (TTCAN Transmit Buffer Mode) TTEN=0时, TTTBM被忽略。 0: TSMODE决定, PTB和STB 1: 通过TBPTR和TTPTR设定 TTCAN模式时, 只需要接收时间戳功能时, 此位可以设置为0, 通过TSMODE决定使用PTB还是STB。 注意: TSMODE位只能在STB空时设定。
3:2	Reserved	必须保持复位值。
1:0	TSSTAT	STB状态 (Transmission Secondary STATus bits) TTEN=0 00: STB空 01: STB小于等于半满 10: STB大于半满 11: STB满 TTEN=0&TTTBM=1 00: PTB和STB空 01: PTB和STB非满 10: 保留 11: PTB和STB满

22.5.6 CAN 接收控制寄存器 (CAN_RCTRL)

CAN Receive Control Register

偏移地址：0xA3

复位值：0x00

7	6	5	4	3	2	1	0
SACK	ROM	ROV	RREL	RBALL	RSV	RSSTAT[1:0]	
R/W	R/W	R	R/W	R/W	R	R	

位	符号	功能描述
7	SACK	自应答 (Self-ACKnowledge) 0: 无自应答 1: LBME=1时, 使能自应答功能
6	ROM	接收BUF上溢模式设定位 (Receive buffer Overflow Mode) 0: 最早接收到的数据被覆盖 1: 新接收到的数据不被存储
5	ROV	接收BUF上溢标志位 (Receive buffer Overflow) 0: 无上溢 1: 上溢, 最少有一个数据丢失 通过写RREL为1清零。
4	RREL	释放接收BUF (Receive buffer RElease) 0: 不释放 1: 表示该接收BUF已经被读取过, RBUF寄存器指向下一个RB SLOT。
3	RBALL	接收BUF数据存储素所有的数据帧 (Receive Buffer stores ALL data frames) 0: 正常模式 1: 存储所有的数据包括有错误的的数据。
2	Reserved	必须保持复位值。
1:0	RSTAT	接收BUF状态 (Receive buffer STATus) 00: RBUF空 01: RBUF非空但小于AFWL编程值 10: RBUF大于等于AFWL编程值但未满 11: 满 (上溢时保持此值)

22.5.7 CAN 接收和发送中断使能寄存器 (CAN_RTIE)

CAN Receive and Transmit Interrupt Enable Register

偏移地址：0xA4

复位值：0xFE

7	6	5	4	3	2	1	0
RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

位	符号	功能描述
7	RIE	接收中断使能 (Receive Interrupt Enable) 0: 禁止 1: 使能
6	ROIE	接收上溢中断使能 (Receive Overrun Interrupt Enable) 0: 禁止 1: 使能
5	RFIE	接收BUF满中断使能 (RB Full Interrupt Enable) 0: 禁止 1: 使能
4	RAFIE	接收BUF将满中断使能 (RB Almost Full Interrupt Enable) 0: 禁止 1: 使能
3	TPIE	PTB发送中断使能 (Transmission Primary Interrupt Enable) 0: 禁止 1: 使能
2	TSIE	STB发送中断使能 (Transmission Secondary Interrupt Enable) 0: 禁止 1: 使能
1	EIE	错误中断使能 (Error Interrupt Enable) 0: 禁止 1: 使能
0	TSFF	TTEN=0 or TTTBM=0: STB满标志 (Transmit Secondary buffer Full Flag) 0: STB SLOT没有被全部填充 1: STB SLOT被全部填充 TTEN=1 and TTTBM=1: TB满标志 (Transmit buffer Full Flag) 0: TBPTR选择的发送BUF没有被全部填充 1: TBPTR选择的发送BUF被全部填充

22.5.8 CAN 接收和发送中断状态寄存器 (CAN_RTIF)

CAN Receive and Transmit Interrupt Status Register

偏移地址: 0xA5

复位值: 0x00

7	6	5	4	3	2	1	0
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	符号	功能描述

7	RIF	接收中断标志 (Receive Interrupt Flag) 0: 未收到数据帧 1: 接收到有效的数据帧或者远程帧 通过应用程序写1清0。
6	ROIF	接收上溢中断标志 (Receive Overrun Interrupt Flag) 0: 未发生接收上溢 1: 发生接收上溢 上溢时ROIF和RFIF同时置1。 通过应用程序写1清0。
5	RFIF	接收BUF满中断标志 (RB Full Interrupt Flag) 0: RB FIFO未满 1: RB FIFO满 通过应用程序写1清0。
4	RAFIF	接收BUF将满中断标志 (RB Almost Full Interrupt Flag) 0: 被填充的RB SLOT数目小于AFWL设定值 1: 被填充的RB SLOT数目大于等于AFWL设定值 通过应用程序写1清0。
3	TPIF	PTB发送中断标志 (Transmission Primary Interrupt Flag) 0: 没有PTB发送完成 1: 请求的PTB发送成功完成 通过应用程序写1清0。 注意: TTCAN模式时, TPIF无效, 仅适用TSIF标志
2	TSIF	STB发送中断标志 (Transmission Secondary Interrupt Flag) 0: 没有STB发送完成 1: 请求的STB发送成功完成 通过应用程序写1清0。 注意: TTCAN模式时, TPIF无效, 仅使用TSIF标志
1	EIF	错误中断标志 (Error Interrupt Flag) 0: BUSOFF位未发生变化, 或者错误计数器的值与ERROR warning limit设定值的相对关系未发生变化。 1: BUSOFF位发生变化, 或者错误计数器的值与ERROR warning limit设定值的相对关系发生变化。 比如错误计数器的值从小于设定值变为大于设定值, 或者从大于设定值变为小于设定值。 通过应用程序写1清0。
0	AIF	取消发送中断标志 (Abort Interrupt Flag) 0: 未取消发送数据 1: 通过TPA和TSA请求的发送消息被成功取消。 通过应用程序写1清0。

22.5.9 CAN 错误中断使能和标志寄存器 (CAN_ERRINT)

CAN ERRor INTerrupt Enable and Flag Register

偏移地址: 0xA6

复位值：0x00

7	6	5	4	3	2	1	0
EWARN	EPASS	WPIE	EPIF	ALIE	ALIF	BEIE	BEIF
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

位	符号	功能描述
7	EWARN	到达设定的ERROR WARNING LIMIT (Error WARNING limit reached) 0: RECNT或者TECNT小于EWL设定值 1: RECNT或者TECNT大于等于EWL设定值 通过应用程序写1清0。
6	EPASS	错误被动 (Error Passive mode active) 0: 节点是主动错误节点 1: 节点时被动错误节点 通过应用程序写1清0。
5	EPIE	错误被动中断使能 (Error Passive Interrupt Enable) 0: 禁止 1: 使能
4	EPIF	错误被动中断标志 (Error Passive Interrupt Flag) 0: 未发生错误主动到错误被动或者错误被动到错误主动的变化 1: 发生错误主动到错误被动或者错误被动到错误主动的变化 通过应用程序写1清0。
3	ALIE	仲裁失败中断使能 (Arbitration Lost Interrupt Enable) 0: 禁止 1: 使能
2	ALIF	仲裁失败中断标志 (Arbitration Lost Interrupt Flag) 0: 仲裁成功 1: 仲裁失败 通过应用程序写1清0。
1	BEIE	总线错误中断使能 (Bus Error Interrupt Enable) 0: 禁止 1: 使能
0	BEIF	总线错误中断标志 (Bus Error Interrupt Flag) 0: 仲裁成功 1: 仲裁失败 通过应用程序写1清0。

22.5.10 CAN 位时序寄存器 (CAN_BT)

CAN Bit Timing Register

偏移地址：0xA8

复位值：0x0102 0203

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[7:0]								RSV	SJW[6:0]						
R/W								R	R/W						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	SEG_2[6:0]							SEG_1[7:0]							
R	R/W							R/W							

位	符号	功能描述
31:24	PRESC	预分频设定 (Prescaler) 此寄存器设定值时将模块BCLK设定位 (PRESC+1) 分频作为TQ的时钟。
23	Reserved	必须保持复位值
22:16	SJW	再同步补偿宽度时间设定 (Bit Timing Segment 2) 再同步补偿宽度时间=(SJW+1)*TQ
15	Reserved	必须保持复位值
14:8	SEG_2	位段2时间单元设定 (Bit Timing Segment 2) 位段2时间=(SEG_2+1)*TQ
7:0	SEG_1	位段1时间单元设定 (Bit Timing Segment 1) 位段1时间=(SEG_1+2)*TQ

22.5.11 CAN 错误和仲裁失败捕捉寄存器 (CAN_EALCAP)

CAN Error and Arbitration Lost Capture Register

偏移地址：0xB0

复位值：0x00

7	6	5	4	3	2	1	0
KOER[2:0]				ALC[4:0]			
R				R/W			

位	符号	功能描述
7:5	KOER	错误类别 (Kind Of Error) 000: 无错误 001: 位错误 010: 形式错误 011: 填充错误 100: 应答错误 101: CRC错误

		110: 其他错误 111: 保留 有错误时KOER位更新，正常发送接收时KOER位保持不变。
4:0	ALC	仲裁失败位置捕捉 (Arbitration Lost Capture) 仲裁失败时ALC记录一帧数据中仲裁失败时的位置。

22.5.12 CAN 警告限定寄存器 (CAN_AFWL)

CAN Warning Limits Register

偏移地址: 0xA7

复位值: 0x00

7	6	5	4	3	2	1	0
AFWL[3:0]				EWL[3:0]			
R/W				R/W			

位	符号	功能描述
7:4	AFWL	接收BUF将满Warning Limit (receive buffer Almost Full Warning Limit) 设定值范围为1~10。 AFWL=0无意义，当做AFWL=1处理。
3:0	EWL	Error Waring Limit编程值 (Programmable Error Warning Limit) Error Waring Limit= (EWL+1) *8。 该寄存器设定值影响EIF标志。

22.5.13 CAN 接收错误计数器寄存器 (CAN_RECNT)

CAN Receive Error CouNT Register

偏移地址: 0xB2

复位值: 0x00

7	6	5	4	3	2	1	0
RECNT[7:0]							
R/W							

位	符号	功能描述
7:0	RECNT	接收错误计数器 (Receive Error CouNT) 接收错误计数器根据CAN协议规定的错误计数增加或者减少。该计数器不存在上溢，255为最大值。

22.5.14 CAN 发送错误计数器寄存器 (CAN_TECNT)

CAN Transmit Error CouNT Register

偏移地址: 0xB3

复位值: 0x00

7	6	5	4	3	2	1	0
TECNT[7:0]							
R/W							

位	符号	功能描述
7:0	TECNT	发送错误计数器 (Transmit Error CouNT) 发送错误计数器根据CAN协议规定的错误计数增加或者减少。该计数器不存在上溢, 255为最大值。

22.5.15 CAN 筛选器组控制寄存器 (CAN_ACFCTRL)

CAN Acceptance Filter Control Register

偏移地址: 0xB4

复位值: 0x00

7	6	5	4	3	2	1	0
RSV		SELMASK	RSV	ACFADR			
R		R/W	R				

位	符号	功能描述
7:6	Reserved	必须保持复位值。
5	SELMASK	选择筛选器的屏蔽寄存器 (SElect acceptance MASK) 0: ACF指向筛选器ID寄存器 1: ACF指向筛选器MASK寄存器 通过ACFADR选择具体的筛选寄存器组
4	Reserved	必须保持复位值。
3:0	ACFADR	筛选器地址 (acceptance filter address) ACFADR指向具体的筛选器, 通过SELMASK去区分ID和MASK。 0000: 指向ACF_1 0001: 指向ACF_2 0010: 指向ACF_3 0011: 指向ACF_4 0100: 指向ACF_5 0101: 指向ACF_6 0110: 指向ACF_7

		0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111: 指向ACF_8

22.5.16 CAN 筛选器组使能寄存器 (CAN_ACFEN)

CAN Acceptance Filter Enable Register

偏移地址: 0xB6

复位值: 0x01

7	6	5	4	3	2	1	0
AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

位	符号	功能描述
7	AE_8	ACF_8使能 (Acceptance Filter 8 Enable) 0: 禁止 1: 使能
6	AE_7	ACF_7使能 (Acceptance Filter 7 Enable) 0: 禁止 1: 使能
5	AE_6	ACF_6使能 (Acceptance Filter 6 Enable) 0: 禁止 1: 使能
4	AE_5	ACF_5使能 (Acceptance Filter 5 Enable) 0: 禁止 1: 使能
3	AE_4	ACF_4使能 (Acceptance Filter 4 Enable) 0: 禁止 1: 使能
2	AE_3	ACF_3使能 (Acceptance Filter 3 Enable) 0: 禁止 1: 使能
1	AE_2	ACF_2使能 (Acceptance Filter 2 Enable) 0: 禁止

		1: 使能
0	AE_1	ACF_1使能 (Acceptance Filter 1 Enable) 0: 禁止 1: 使能

22.5.17 CAN 筛选器组 code 和 mask 寄存器 (CAN_ACF)

CAN Acceptance Filter code and mask Register

偏移地址: 0xB8

复位值: 0XXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV	AIDEE	AIDE	ACODE[28:16] or AMASK[28:16]												
R	R/W	R/W	R/W												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACODE[15:0] or AMASK[15:0]															
R/W															

位	符号	功能描述
31	Reserved	读出值为不定
30	AIDEE	ID位比较使能 (Acceptance mask IDE bit check enable) 尽在SELMASK=1时有效 0: 筛选器接收标准格式和扩展格式帧 1: 筛选器接收AIDE位定义的标准格式或者扩展格式帧
29	AIDE	ID位MASK 0: 筛选器仅接收标准格式 1: 筛选器仅接收扩展格式
28:0	ACODE/ AMASK	筛选器CODE (acceptance filter code) 通过ACFADR指向具体的筛选器。 SELMASK=0时表示筛选器的CODE。 标准格式时使用位10~位0, 扩展格式时使用位29~位0。 筛选器CODE (acceptance filter mask) 通过ACFADR指向具体的筛选器。 SELMASK=1时表示筛选器的MASK。 标准格式时使用位10~位0, 扩展格式时使用位29~位0。

22.5.18 TTCAN TB slot 指针寄存器 (CAN_TB SLOT)

TTCAN TB Slot Pointer Register

偏移地址：0xBE

复位值：0x00

7	6	5	4	3	2	1	0
TBE	TBF	Reserved			TBPTR[2:0]		
R/W	R/W	R			R/W		

位	符号	功能描述
7	TBE	设置TB为空 (set TB slot to “empty”) 0: 无操作 1: 被TBPTR选择的SLOT被标记为空 当SLOT被标记为空并且TSFF=0时, TBE自动复位为0。 如果设定此位为1时, 被选定的SLOT中存在数据正在发送状态则TBE=1, 则等到发送完成、发送错误或者发送取消后TBE复位为0。 TBE优先级高于TBF。
6	TBF	设置TB已填充 (set TB slot to “Filled”) 0: 无操作 1: 被TBPTR选择的SLOT被标记为已填充 当SLOT被标记为已填充并且TSFF=1时, TBE自动复位为0。
5:3	Reserved	必须保持复位值。
2:0	TBPTR	TB SLOT指针 (Pointer to a TB message slot) 000: 指向PTB 001: 指向STB SLOT1 010: 指向STB SLOT2 011: 指向STB SLOT3 100: 指向STB SLOT4 其他: 设定禁止 被指向的TB SLOT可以通过TBUF进行读写访问, 并且可以通过TBE和TBF来标记是否已经被填充。 TTCAN模式时, TBSEL和TSNEXT寄存器无效。 注意: 仅可以在TSFF=0时对该位进行写操作。

22.5.19 TTCAN 时间触发配置寄存器 (CAN_TTCFG)

TTCAN TB Slot Pointer Register

偏移地址：0xBF

复位值：0x90

7	6	5	4	3	2	1	0
WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC[1:0]		TTEN
R/W	R/W	R	R/W	R/W	R/W		R/W

位	符号	功能描述
7	WTIE	触发看门中断使能 (Watch Trigger Interrupt Enable) 0: 禁止 1: 使能
6	WTIF	触发看门中断标志 (Watch Trigger Interrupt Flag) 当CYCLE_COUNT值=TT_WTRIG设定值时且WTIE=1时, WTIF置位。 通过应用程序写1清0。
5	TEIF	触发错误中断标志 (Trigger Error Interrupt Flag) TT_TTIG设定值小于实际的CYCLE_TIME时, TEIF置位。
4	TTIE	时间触发中断使能 (Time Trigger Interrupt Enable) 0: 禁止 1: 使能
3	TTIF	时间触发中断标志 (Time Trigger Interrupt Flag) 当CYCLE_COUNT值=TT_TRIG设定值时且TTIE=1时, TTIF置位。 如果TT_TRIG没有更新, 则TTIF只置位1次, 下一个基本CYCLE不置位。 通过应用程序写1清0。
2:1	T_PRESC	TTCAN计数器预分频 (TTCAN Timer PRESCaler) 00: BT寄存器设定的位时间的1分频 01: BT寄存器设定的位时间的2分频 10: BT寄存器设定的位时间的4分频 11: BT寄存器设定的位时间的8分频 注意: T_PRESC可在TTEN=0时进行写操作或者写TTEN=1时同时操作。
0	TTEN	TTCAN使能 (Time Trigger Enable) 0: 禁止 1: 使能TTCAN, 计数器开始计数。

22.5.20TTCAN 参考消息寄存器 (CAN_REF_MSG)

TTCAN Reference Message Register

偏移地址: 0xC0

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REF_IDE	Reserved	REF_ID[28:16]													
R/W	R	R/W													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REF_ID[15:0]															
R/W															

位	符号	功能描述
31	REF_IDE	参考消息的IDE位 (REference message IDE bit) 0: 标准格式 1: 扩展格式
30:29	Reserved	读出值为不定。
28:0	REF_ID	参考消息的ID位 (REference message IDentifier) REF_ID=0: REF_ID[28:0]有效 REF_ID=1: REF_ID[28:0]有效 REF_ID用于检测参考消息, 适用于发送和接收。 检测到参考消息后, 当前帧的Sync_Mark则变成Ref_Mark。 REF_ID[2:0]固定为0, 并不检查其值, 这样最多可以支持8个潜在的time master。 当REF_MSG的最高字节写操作后, 则需要等待6个CAN时钟周期以完成REF_MSG向CAN时钟域的传递。

22.5.21 TTCAN 触发配置寄存器 (CAN_TRG_CFG)

TTCAN Reference Message Register

偏移地址: 0xC4

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEW[3:0]				RSV	TTYPE[2:0]			Reserved					TTPTR[2:0]		
R/W				R	R/W			R					R/W		

位	符号	功能描述
15:12	TEW	发送使能窗口 (Transmit Enable Window) 用于TTCAN的单次发送触发模式 (Single Shot Transmit Trigger), 可以设定TEW+1个cycle time的窗口, 发送仅在此窗口内被允许。
11	Reserved	必须保持复位值。
10:8	TTYPE	触发类型 (Trigger Type) 000: 立即触发 (Immediate Trigger for immediate transmission) 001: 时间触发 (Time Trigger for receive triggers) 010: 单次发送触发 (Single Shot Transmit Trigger for exclusive time windows) 011: 发送开始触发 (Transmit Start Trigger for merged arbitrating time windows) 100: 发送停止触发 (Transmit Stop Trigger for merged arbitrating time windows) 其他: 保留 触发时间通过TT_TRIG寄存器设定, TB Slot通过TTPTR选择。
7:3	Reserved	必须保持复位值。
2:0	TTPTR	发送触发器TB slot指针 (Transmit Trigger TB slot Pointer) 000: 指向PTB 001: 指向STB SLOT1 010: 指向STB SLOT2 011: 指向STB SLOT3 100: 指向STB SLOT4 其他: 设定禁止 如果指向的TB SLOT被标记为空, 当到达触发时间后, TEIF置位。

22.5.22 TTCAN 触发时间寄存器 (CAN_TRG_TRIG)

TTCAN Reference Message Register

偏移地址: 0xC6

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT_TRIG[15:0]															
R/W															

位	符号	功能描述
15:0	TT_TRIG	触发时间 (Trigger Time) 用于指定触发器的cycle time, 对于发送触发器来说发送SOF时间大约是TT_TRIG设定值+1。 当TT_TRIG的最高字节写操作后, TT_TRIG值开始向CAN时钟域的传递。因此如果BYTE操作, 需先写低字节再写高字节。

22.5.23 TTCAN 触发看门时间寄存器 (TRG_WTRIG)

TTCAN Watch Trigger Time Register

偏移地址: 0xC8

复位值: 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT_WTRIG[15:0]															
R/W															

位	符号	功能描述
15:0	TT_WTRIG	触发时间 (Trigger Time) 用于指定看门触发器的cycle time。 当TT_WTRIG的最高字节写操作后, TT_WTRIG值开始向CAN时钟域的传递。因此如果BYTE操作, 需先写低字节再写高字节。

23. 循环冗余校验（CRC）

23.1 概述

在许多应用中，都需要用 CRC 算法来校验数据的完整性和正确性。尤其是在数据传输中，CRC 校验更是被广泛运用。本模块实现 CRC16 和 CRC32 两种算法对数据进行运算和校验。

23.2 功能框图

以下示意了 CRC 算法在数据传输中的一个最典型的应用：

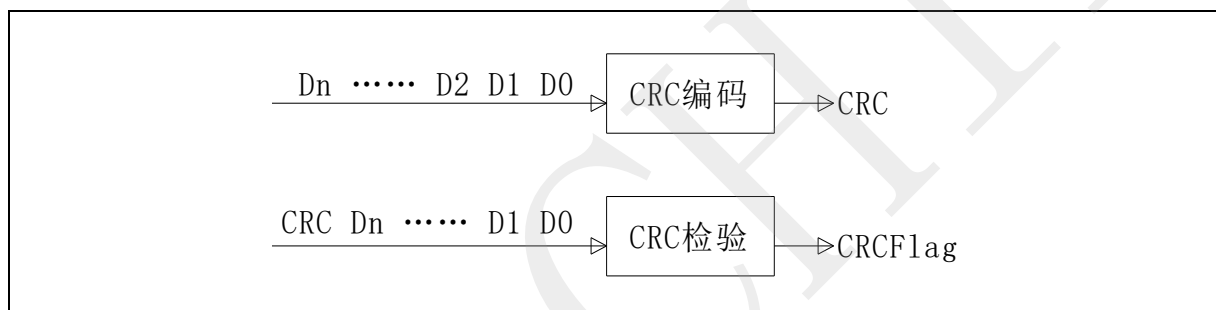


图 23-1 CRC 应用示意图

23.3 功能描述

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 32 位和 16 位的 CRC。其中，CRC32 的生成多项式为 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ ，32 位初值为“0xFFFFFFFF”。CRC16 的生成多项式为 $x^{16}+x^{12}+x^5+1$ ，16 位初值为“0xFFFF”。

本模块功能包括：

- CRC 编码和 CRC 校验；
- 3 种位宽访问方式 8 位、16 位、32 位：
 - 8 位位宽下输入数据示例为 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77；
 - 16 位位宽下输入数据示例为 0x1100, 0x3322, 0x5544, 0x7766；
 - 32 位位宽下输入数据示例为 0x33221100, 0x77665544；

23.4 软件基本操作

23.4.1 CRC16 编码模式

编码模式可以对原始数据编码以计算其 CRC 值，操作流程如下所示：

- Step 1: 向 CRC_CR.CR 写入 1'b0，选择 CRC16。
- Step 2: 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算。
- Step 3: 将待编码的原始数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）
- Step 4: 读取 CRC_RESULT[15:0]，即为 CRC 值。
- Step 5: 如果需要计算其他数据，重复 step1~step4。

23.4.2 CRC16 检验模式

检验模式可以检验已编码的数据是否被篡改，操作流程如下所示：

- Step 1: 向 CRC_CR.CR 写入 1'b0，选择 CRC16。
- Step 2: 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算。
- Step 3: 将已编码的数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）
- Step 4: 读取 CRC_CR.FLAG，以判定编码的数据是否被篡改。当 CRC_FLAG=0，表示校验失败；当 CRC_FLAG=1，表示校验成功。
- Step 5: 如果需要检验其他数据，重复 step1~step4。

23.4.3 CRC32 编码模式

编码模式可以对原始数据编码以计算其 CRC 值，操作流程如下所示：

- Step 1: 向 CRC_CR.CR 写入 1，选择 CRC32。
- Step 2: 向 CRC_RESULT[31:0]写入 0xFFFF FFFF，初始化 CRC 计算。
- Step 3: 将待编码的原始数据按 8 位/16 位/32 位的组织方式，依次写入 CRC_DATA 寄存器。（注：数据写入的具体方式参见 CRC_DATA 寄存器的说明）
- Step 4: 读取 CRC_RESULT[31:0]，即为 CRC 值。

Step 5: 如果需要计算其他数据, 重复 step1~step4。

23.4.4 CRC32 检验模式

检验模式可以检验已编码的数据是否被篡改, 操作流程如下所示:

Step 1: 向 CRC_CR.CR 写入 1, 选择 CRC32。

Step 2: 向 CRC_RESULT[31:0]写入 0xFFFF FFFF, 初始化 CRC 计算。

Step 3: 将已编码的数据按 8 位/16 位/32 位的组织方式, 依次写入 CRC_DATA 寄存器。(注: 数据写入的具体方式参见 CRC_DATA 寄存器的说明)

Step 4: 读取 CRC_CR.FLAG, 以判定编码的数据是否被篡改。当 CRC_CR.FLAG=0, 表示校验失败; 当 CRC_CR.FLAG=1, 表示校验成功。

Step 5: 如果需要检验其他数据, 重复 step1~step4。

23.5 寄存器

基地址：0x4002 3000

寄存器	偏移地址	描述
CRC_CR	0x00	CRC 控制寄存器，选择CRC16/32，读取校验结果标志位。
CRC_RESULT	0x04	CRC 结果寄存器，计算完成后对该寄存器读取即获得结果。
CRC_DATA	0x80-0xFF	CRC 数据寄存器，用于输入需要运算的数据。

表 23-1CRC 寄存器列表

23.5.1 控制寄存器（CRC_CR）

偏移地址：0x00

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													FLAG	CR	
R													RO	RW	

位	符号	功能描述
31:2	Reserved	保留
1	FLAG	CRC校验结果标志位；0：当前校验错误；1：当前校验正确；
0	CR	CRC功能选择；0：CRC16；1：CRC32

23.5.2 结果寄存器（CRC_RESULT）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESULT[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT[15:0]															
RW															

位	符号	描述
31:0	RESULT	CRC结果寄存器；选择CRC16时，取RESULT[15:0]；选择CRC32时，取RESULT[31:0]；

23.5.3 数据寄存器（CRC_DATA）

偏移地址：0x80-0xFF

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
WO															

位	符号	功能描述
31:0	DATA	本寄存器用于输入需要运算的数据；本寄存器的地址是一个范围（0x80-0xFF），对该地址范围内的任何地址进行操作，都会认为对本寄存器进行操作。这样定义的目的就是方便软件可以用STM指令对本寄存器进行连续的32位数据写入操作，以加快运算速度。同时本寄存器也支持8/16位的输入方式。需要特别注意的是，无论是计算还是检验，对本寄存器进行的数据写入操作，必须符合“位宽一致、先低后高”的原则，即“每次写入数据，都必须写入到与本次有效数据位宽相等的寄存器中，并且较低位数据的写入先于较高位数据”。举例如下： 如果待写入数据为0xFEDCBA9876543210，那么 按字节（单次8位）写入时： *((unsigned char *) 0x40020980) = 0x10 ; *((unsigned char *) 0x40020981) = 0x32 ;

		<pre>*((unsigned char *) 0x40020986) = 0xDC ; *((unsigned char *) 0x40020987) = 0xFE ; 按半字（单次16位）写入时： *((unsigned short *) 0x40020980) = 0x3210 ; *((unsigned short *) 0x40020982) = 0x7654 ; *((unsigned short *) 0x40020984) = 0xBA98 ; *((unsigned short *) 0x40020986) = 0xFEDC ; 按全字（单次32位）写入时： *((unsigned int *) 0x40020980) = 0x76543210 ; *((unsigned int *) 0x40020984) = 0xFEDCBA98 ;</pre>

24. 真随机数发生器（TRNG）

24.1 概述

真随机数模块产生 64 位真随机数。

24.2 功能框图

以下示意了 TRNG 模块的数据流：

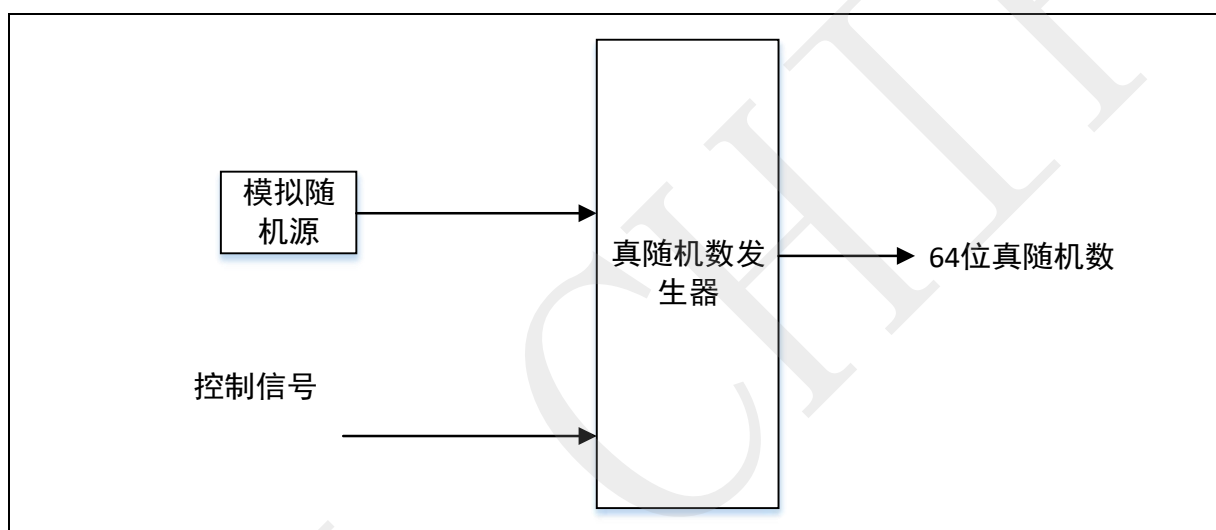


图 24-1 TRNG 数据流

24.3 功能描述

本模块采用内部的模拟随机源，每次启动都可以产生 64bits 真随机数。此外，还可以对真随机数生成的方式进行软件配置，详细内容可查看寄存器描述章节。生成的 64 位真随机数分别存放在 DATA0 和 DATA1 寄存器中。

24.4 软件基本操作

24.4.1 生成 64bits 真随机数的操作流程（上电第一次）

上电第一次生成 64bits 真随机数，需要经过以下操作：

Step1: 打开随机源电路：对真随机数控制寄存器的 Bit0（RNG_CR.RNGcir_En）写入“1”，启动随机源电路，开始输出串行真随机数。

Step2: 选择重新装载初始值: 将真随机数模式寄存器的 Bit0 (RNG_MODE.RNG_LOAD) 设置为“1”, 使新产生的真随机数的初始值从随机源获得。

Step3: 选择 PRNG64 的直接反馈的方式: 将真随机数模式寄存器的 Bit1 (RNG_MODE.RNG_FDBK) 设置为“1”, 将反馈信号与随机源异或后输入 PRNG 中。

Step4: 选择 PRNG64 的移位次数: 设置真随机数模式寄存器的 Bit4 – Bit2 (RNG_MODE.RNG_CNT) 为“110”, 选择移位 256 次。

Step5: 生成真随机数: 软件将“1”写入真随机数控制寄存器的 Bit1 (RNG_CR.RNG_RUN), 硬件根据真随机数生成配置进行操作, 在操作完成后, 硬件自动将 Bit1 清为“0”。

Step6: 选择不重新装载初始值: 将真随机数模式寄存器的 Bit0 (RNGModeReg.RNG_Load) 设置为“0”。

Step7: 选择 PRNG64 的直接反馈的方式: 将真随机数模式寄存器的 Bit1 (RNG_MODE.RNG_FDBK) 设置为“0”, 将反馈信号直接输入 PRNG 中。

Step8: 选择 PRNG64 的移位次数: 设置真随机数模式寄存器的 Bit4 – Bit2 (RNG_MODE.RNG_CNT) 为“100”, 选择移位 64 次。

Step9: 生成真随机数: 软件将“1”写入真随机数控制寄存器的 Bit1 (RNG_CR.RNG_RUN), 硬件根据真随机数生成配置进行操作, 在操作完成后, 硬件自动将 Bit1 清为“0”。

Step10: 读取真随机数: 软件在查询到真随机数控制寄存器的 Bit1 (RNG_CR.RNG_RUN) 变为“0”后, 通过读取真随机数数据寄存器 0 (RNG_DATA0) 和真随机数数据寄存器 1 (RNG_DATA1), 得到 64Bits 真随机数。

Step11: 完成真随机数的生成后, 推荐选择关闭随机源电路, 节省功耗: 对真随机数控制寄存器的 Bit0 (RNG_CR.RNGcir_En) 写入“0”, 关闭随机源电路。

24.4.2 生成 64bits 真随机数的操作流程 (非上电第一次生成)

非上电第一次生成 64bits 真随机数, 需要经过以下操作:

Step1: 打开随机源电路: 对真随机数控制寄存器的 Bit0 (RNG_CR.RNGcir_En) 写

入“1”，启动随机源电路，开始输出串行真随机数。

Step2：选择不重新装载初始值：将真随机数模式寄存器的 Bit0（RNG_MODE.RNG_LOAD）设置为“0”。

Step3：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（RNG_MODE.RNG_FDBK）设置为“1”，将反馈信号与随机源异或后输入 PRNG 中。

Step4：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4—Bit2（RNG_MODE.RNG_CNT）为“110”，选择移位 256 次。

Step5：生成真随机数：软件将“1”写入真随机数控制寄存器的 Bit1（RNG_CR.RNG_RUN），硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 Bit1 清为“0”。

Step6：选择 PRNG64 的直接反馈的方式：将真随机数模式寄存器的 Bit1（RNG_MODE.RNG_FDBK）设置为“0”，将反馈信号直接输入 PRNG 中。

Step7：选择 PRNG64 的移位次数：设置真随机数模式寄存器的 Bit4—Bit2（RNG_MODE.RNG_CNT）为“100”，选择移位 64 次。

Step8：读取真随机数：软件在查询到真随机数控制寄存器的 Bit1（RNG_CR.RNG_RUN）变为“0”后，通过读取真随机数数据寄存器 0（RNG_Data0）和真随机数数据寄存器 1（RNG_Data1），得到 64Bits 真随机数。

如果需要继续生成新的真随机数，那么回到 Step2，直到满足要求。

Step9：完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：对真随机数控制寄存器的 Bit0（RNG_CR.RNGcir_En）写入“0”，关闭随机源电路。

24.5 寄存器

基地址：0x40025000

寄存器	偏移地址	描述
TRNG_CR	0x00	控制寄存器
TRNG_MODE	0x04	模式寄存器
TRNG_DATA0	0x0C	数据寄存器0
TRNG_DATA1	0x10	数据寄存器1

24.5.1 控制寄存器（TRNG_CR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved							RNG_RUN		RNGcir_EN														
							RW		RW														

位	标记	功能描述
31:2	Reserved	
1	RNG_RUN	软件写入“1”，开始产生新的64bits随机数；运行完毕后，硬件清零； 0：随机数产生完成； 1：写1启动随机数产生，读1表示随机数正在产生；
0	RNGcir_EN	随机源电路使能位： 0：关闭随机源； 1：打开随机源；

24.5.2 模式寄存器（TRNG_MODE）

偏移地址：0x04

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved				RNG_CNT			RNG_FDBK	RNG_LOAD															
				RW			RW	RW															

位	标记	描述
31:5	Reserved	
4:2	RNG_CNT	64bits RNG的反馈移位次数 3' b000:移位0次（即输出随机源的采样值） 3' b001:移位8次 3' b010:移位16次 3' b011:移位32次 3' b100:移位64次 3' b101:移位128次 3' b110:移位256次 3' b111:Reserved
1	RNG_FDBK	在移位操作时，64bits RNG的反馈信号是否与随机源进行异或操作 0：不进行异或操作； 1：进行异或操作；
0	RNG_LOAD	在产生新的随机数时，64bits RNG是否从随机源获得新的初始值 0：不装载新的初始值（产生伪随机数）； 1：装载新的初始值（产生真随机数）；

24.5.3 数据寄存器 0 (TRNG_DATA0)

偏移地址: 0x0C

复位值: ---

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA0[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[15:0]															
RO															

位	标记	功能描述
31:0	DATA0	软件对本寄存器读取将得到低32位的随机数

24.5.4 数据寄存器 1 (TRNG_DATA1)

偏移地址: 0x10

复位值: ---

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[15:0]															
RO															

位	标记	功能描述
31:0	DATA1	软件对本寄存器读取将得到高32位的随机数

25. 高级加密标准模块（AES）

25.1 功能定义

25.1.1 AES 算法简述

AES（The Advanced Encryption Standard）是美国国家标准技术研究所（NIST）在 2000 年 10 月 2 日正式宣布的新的数据加密标准。

AES 的分组长度固定为 128 位，而密钥长度支持 128、192 和 256 位。对于加密来说，其输入是一个明文分组和一个密钥，输出是一个密文分组；对解密而言，输入是一个密文分组和一个密钥，而输出是一个明文分组。此过程如图 25-1 所示：

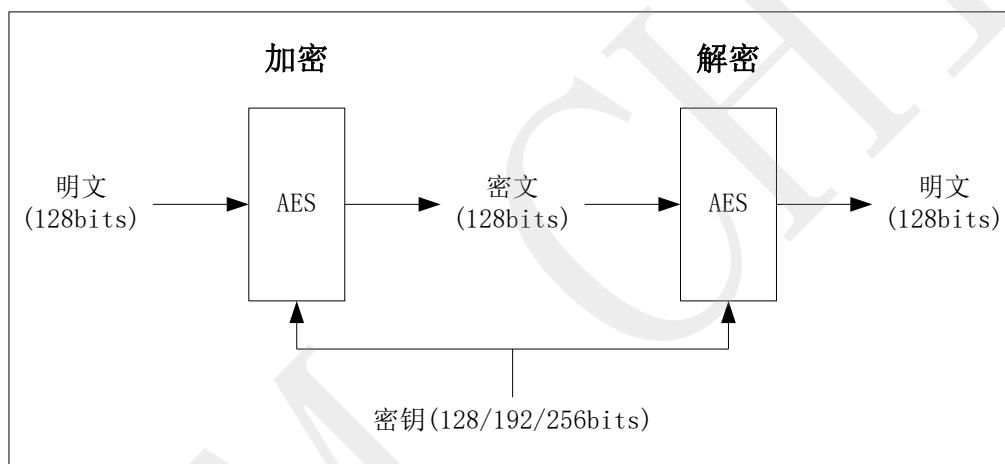


图 25-1 AES 的加解密示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个 4×4 的矩阵中，称为状态（state），AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分组密码，包含了轮变换对状态的重复作用。AES 的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 包括求每个字节在 $GF(2^8)$ 中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey，状态中的各字节与轮密钥进行逐位异或操作。AES 的加密流程如图 25-2 所示：

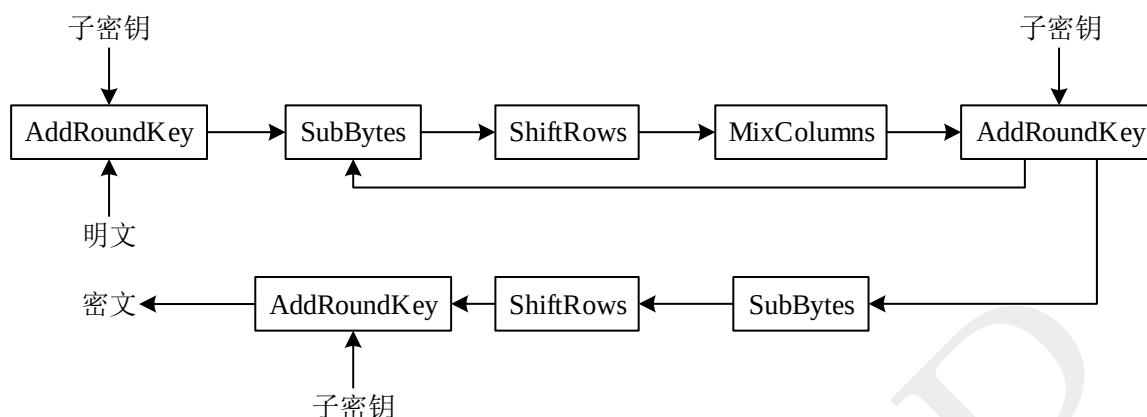


图 25-2 AES 的加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。

由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》

25.1.2 AES 模块功能描述

- 执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合《FIPS PUB 197》对算法原理的描述；

- 支持 128、192 和 256 位密钥。

25.2 异常机制

- 只支持 32 位访问，其它位宽的访问会导致系统异常，进入硬件异常中断。
- 访问 AES 模块的偏移地址大于等于 0x80 的地址，会导致系统异常，进入硬件异常中断。

25.3 本模块操作说明

本模块共有两个功能：加密、解密。对两个功能的操作有一些共同的特点，下面先介绍其共同点，再分别介绍每个功能的标准操作流程。

25.3.1 IP 操作的共同点

1. 在 AES 加解密过程中，数据寄存器会改变，如果下次运算的被操作数据就是本次运算的结果，那么就无需重新写入数据了。
2. 支持 128、192 和 256 位密钥，128 位密钥写入偏移地址：0x20~0x2C，192 位密钥写入偏移地址 0x20~0x34，256 位密钥写入偏移地址 0x20~0x3C。
3. 判断模块运算结束的方法：不断读取 AES_CR.Start，如果其值变为 0，则表示运算结束。

25.3.2 加密操作流程

Step 1: 将待加密的 128 位数据写入数据寄存器（AES_DATA）中。

Step 2: 将加密密钥写入密钥寄存器（AES_KEY）中。

Step 3: 根据密钥长度设置 AES_CR.Keysize

Step 4: 将 AES_CR.Mode 设置为 0，启动加密模式。

Step 5: 向控制寄存器中的 AES_CR.Start 写入 1，启动模块进行运算。

Step 3、Step 4 和 Step 5 可同时进行。

Step 6: 等待 AES_CR.Start 的值恢复为 0，模块运算结束。

Step 7: 读取数据寄存器（AES_DATA），获得 128 位密文。

25.3.3 解密操作流程

Step 1: 将待解密的 128 位数据写入数据寄存器（AES_DATA）中。

Step 2: 将解密密钥写入密钥寄存器（AES_KEY）中。

Step 3: 根据密钥长度设置 AES_CR.Keysize

Step 4: 将 AES_CR.Mode 设置为 1，启动解密模式。

Step 5: 向控制寄存器中的 AES_CR.Start 写入 1，启动模块进行运算。

Step 3、Step 4 和 Step 5 可同时进行。

Step 6: 等待 AES_CR.Start 的值恢复为 0，模块运算结束。

Step 7: 读取数据寄存器（AES_DATA），获得 128 位明文。

25.3.4 数据示例

128 位明文: 0xFFEEDDCCBBAA99887766554433221100

128 位密钥: 0x0F0E0D0C0B0A09080706050403020100

128 位密文: 0x5AC5B47080B7CDD830047B6AD8E0C469

表 25-1 128 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
Key0	0x03020100	Data0	0xD8E0C469
Key1	0x07060504	Data1	0x30047B6A
Key2	0x0B0A0908	Data2	0x80B7CDD8
Key3	0x0F0E0D0C	Data3	0x5AC5B470

128 位明文: 0xFFEEDDCCBBAA99887766554433221100

192 位密钥: 0x17161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 25-2 192 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
Key0	0x03020100	Data0	0xA47CA9DD
Key1	0x07060504	Data1	0xE0DF4C86
Key2	0x0B0A0908	Data2	0xA070AF6E
Key3	0x0F0E0D0C	Data3	0x91710DEC
Key4	0x13121110		
Key5	0x17161514		

128 位明文：0xFFEEDDCCBBAA99887766554433221100

256 位密钥：0x1F1E1D1C1B1A191817161514131211100F0E0D0C0B0A090807060504
03020100

128 位密文：0x5AC5B47080B7CDD830047B6AD8E0C469

表 25-3 192 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
Key0	0x03020100	Data0	0xCAB7A28E
Key1	0x07060504	Data1	0xBF456751
Key2	0x0B0A0908	Data2	0x9049FCAE
Key3	0x0F0E0D0C	Data3	0x8960494B
Key4	0x13121110		
Key5	0x17161514		
Key6	0x1B1A1918		
Key7	0x1F1E1D1C		

25.4 运行时间说明

本模块从启动一次运算（AES_CR.Start 写入 1）到该次运算结束（AES_CR.Start 恢复到 0）所需时间如

表 25-4 所示：

表 25-4 AES 加解密运行时间

	128 位密钥	192 位密钥	256 位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

25.5 模块寄存器说明

AES 基地址 0x40026000

寄存器	偏移地址	描述
AES_CR	0x00或0x40	控制寄存器
AES_Data	0x10~0x1C	数据寄存器
AES_Key	0x20~0x3C	密钥寄存器

表 25-5 寄存器列表

25.5.1 控制寄存器（AES_CR）

偏移地址：0x00 或 0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											Keysize	Reserved	Mode	Start	
R											RW	R	RW	RW	

位	符号	功能描述
31:5	Reserved	保留位，读为0
4:3	Keysize	2' b00: 密钥长度为128位 2' b01: 密钥长度为192位 2' b10: 密钥长度为256位 2' b11: 密钥长度为128位
2	Reserved	保留位，读为0
1	Mode	0: 加密运算 1: 解密运算
0	Start	0: 本模块运算结束或未被启动 1: 启动本模块进行运算

说明：

1. AES_CR.Start 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。
2. 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES_CR.Start = 0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

25.5.2 数据寄存器 (AES_Data)

偏移地址: 0x10~0x1C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data[15:0]															
RW															

位	符号	功能描述
31:0	Data	存放AES算法的128比特明文/密文

说明:

1. 数据寄存器由四个 32 位的寄存器组成 128 位数据, 用于在模块运算前存放需要被加密的明文或者需要被解密的密文, 并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128 位明文	128 位密文	128 位密文	128 位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据, 读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下:

数据举例: 0xFFEEDDCBBAA99887766554433221100

偏移地址	寄存器名称	填入数据
0x10	AES_Data0	0x33221100
0x14	AES_Data1	0x77665544
0x18	AES_Data2	0xBBAA9988
0x1C	AES_Data3	0xFFEEDDCC

2. 对于本寄存器的写入只能在本模块没有处于运算状态时(即 AES_CR.Start = 0 时)才能进行, 否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时(即 AES_CR.Start = 0 时)才能进行, 否则对本寄存器的读取将得到全 0。

25.5.3 密钥寄存器 (AES_Key)

偏移地址: 0x20~0x3C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Key[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Key[15:0]															
RW															

位	符号	描述
31:0	Key	存放AES算法的128、192、256比特密钥

1. 密钥寄存器由 8 个 32 位的寄存器组成，存放输入的初始密钥。写操作时需要分别对 8 个 32 位的寄存器进行操作。对应的操作顺序如下：

128 位密钥举例: 0x0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C

192 位密钥举例: 0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908
0x2C	AES_Key3	0x0F0E0D0C
0x30	AES_Key4	0x13121110
0x34	AES_Key5	0x17161514

256 位密钥举例: 0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

偏移地址	寄存器名称	填入数据
0x20	AES_Key0	0x03020100
0x24	AES_Key1	0x07060504
0x28	AES_Key2	0x0B0A0908

0x2C	AES_Key3	0x0F0E0D0C
0x30	AES_Key4	0x13121110
0x34	AES_Key5	0x17161514
0x38	AES_Key6	0x1B1A1918
0x3C	AES_Key7	0x1F1E1D1C

2. 对于本寄存器的写入只能在本模块没有处于运算状态时(即 AES_CR.Start = 0 时)才能进行, 否则硬件将自动忽略对本寄存器的写操作。
3. 对于本寄存器的读取只能在本模块没有处于运算状态时(即 AES_CR.Start = 0 时)才能进行, 否则对本寄存器的读取将得到全 0。

26. 国密 SM4 模块

26.1 概述

提供外部以 AMBA2.0 AHB 总线(8 位, 16 位和 32 位) 小端方式访问; 本模块的运算数据和密钥的输入输出格式为大端格式; 实现 SM4 标准加密和解密。

26.2 功能描述

本模块提供以下功能:

- 提供外部以 AMBA2.0 AHB 总线(8 位, 16 位和 32 位) 小端方式访问, 同时也支持系统对本模块的权限控制
- 执行 SM4 算法标准的加密流程和解密流程, 其执行结果完全符合《无线局域网产品使用的 SMS4 密码算法》对算法原理的描述
- 提供芯片测试模式和芯片应用模式两种操作模式:
 - 在芯片测试模式下可以修改 SKEY 寄存器中的密钥
 - 在芯片应用模式下则禁止 SKEY 寄存器进行修改
- 提供一个安全的硬件实现架构, 能够有效抵御 Side Channel Attack 中的差异功耗分析 (DPA)

26.3 模块接口定义



图表 26-1 SM4 算法模块接口定义

表格 26-1 SM4 算法模块接口信号说明

NAME	I/O	SAMPLED BY	FUNCTION
Reset signal			
hresetn	I	hclk	AHB 异步复位信号，低电平有效
Clock signals			
hclk	I		AHB 时钟信号
IO interface signals			
mpufail_i	I	hclk	来自 MPU 的非法访问信号： 0: 合法访问；1: 非法访问
privctrl_i	I	hclk	来自系统控制的权限控制信号： 0: 不允许用户模式访问本模块； 1: 允许用户模式访问本模块
polarity_i	I	hclk	写入数据的极性位 0: hwddata 为正逻辑值； 1: hwddata 为反逻辑值
test_enb_i	I	hclk	系统工作模式 0: 芯片测试模式；1: 芯片应用模式
polarity_o	O	hclk	读出数据的极性位 0: hrdata 为正逻辑值； 1: hrdata 为反逻辑值
Bus interface signals			
hsel	I	hclk	本模块选择信号，高电平有效

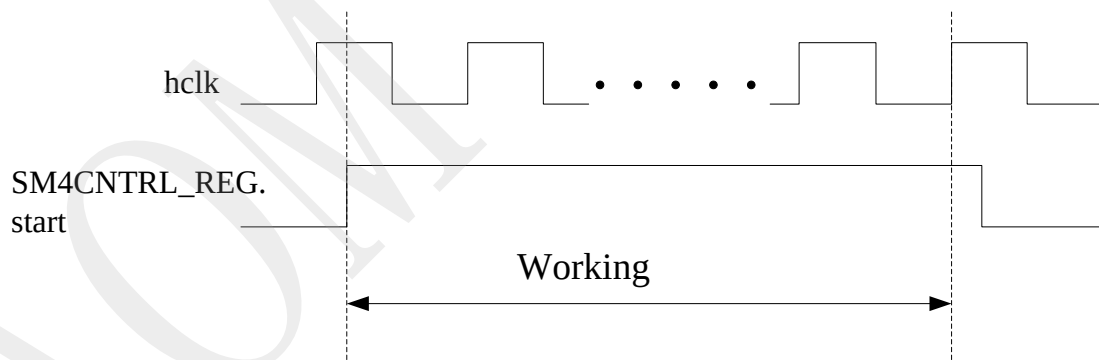
haddr	I	hclk	地址总线
hsize	I	hclk	传输位宽信号: 3'b000: 8 位; 3'b001: 16 位; 3'b010: 32 位; others: RFU
htrans	I	hclk	传输类型信号: 2'b00: 空闲; 2'b01: 忙; 2'b10: 非连续; 2'b11: 连续 本模块忽略空闲和忙传输类型
hwrite	I	hclk	传输方向信号: 高电平表示写操作; 低电平表示读操作
hwdata	I	hclk	写数据总线, 32Bits
hready	I	hclk	全局 ready 信号, 高电平有效
hprot	I	hclk	安全控制信号(在权限控制中有进一步的描述)
hready_resp	O	hclk	响应的 ready, 高电平表示访问完成
hresp	O	hclk	传输响应信号 2'b00: OKAY; 2'b01: ERROR; 2'b10: RETRY; 2'b11: SPLIT
hrdata	O	hclk	读数据总线, 32Bits

注: hwdata 信号说明

hsize	接口位宽	haddr [1:0]	说明
3'b000	8 位	2'b00	hwdata [7:0]有效
		2'b01	hwdata [15:8]有效
		2'b10	hwdata [23:16]有效
		2'b11	hwdata [31:24]有效
3'b001	16 位	2'b0x	hwdata [15:0]有效
		2'b1x	hwdata [31:16]有效
3'b010	32 位	2'bxx	hwdata [31:0]有效
others	无效	2'bxx	hwdata [31:0]忽略

26.4 性能要求

26.4.1 运行时间说明



图表 26-2 SM4 运算启停时序

运行时间如下表所示：

运行时间	
加密时间 (cycle)	67
密钥扩展时间 (cycle)	33

标准解密时间 (cycle)	100
简化解密时间 (cycle)	67

26.5 软件操作说明

本模块共有四个功能：加密、标准解密、密钥扩展和简化解密。对四个功能的操作有一些共同的特点，下面先介绍其共同点，再分别介绍每个功能的标准操作流程。

26.5.1 IP 操作的共同点

- 1) 上电后，首先必须通过端口 `hresetn` 对本模块进行一次异步复位操作。时钟 `hclk` 必须在复位脱离前稳定有效，并且在后续运行中持续稳定。
- 2) 在 SM4 加解密过程中，数据寄存器会改变，如果下次运算操作的数据就是本次运算的结果，那么就无需重新写入数据了。
- 3) 在用相同的密钥加解密大量数据的情况下，不需要重复写入密钥，且在该密钥进行过正向扩展（密钥扩展或标准解密）的情况下，可启动简化解密流程进行解密。
- 4) SM4 内部可同时保存简化解密密钥和加密密钥。即在用有效的解密密钥进行过密钥扩展或标准解密后，简化解密密钥就被保存了下来，如果输入新的密钥进行加密也不会覆盖简化解密密钥，这样就可以在不重复输入加密密钥和解密密钥的情况下，加密流程和简化解密流程也可交替进行。
- 5) 为了有效抵御差异功耗分析，在每次异步复位后必须输入新的随机数，并建议在每次启动本模块运算前都输入新的随机数，每次输入的 64bits 随机数必须保证任何一个字节不等于 0。当然，即使不输入新的随机数也不会影响模块本身运算的正确性，只是可能会降低抵御差异功耗分析的能力。
- 6) 判断模块运算结束的方法：不断读取 `CNTRL_REG.start`，如果其值变为 0，则表示运算结束。

26.5.2 加密流程

- 1) 将 64bits 随机数写入 **rand_reg** 中（在有抗 DPA 要求的应用场合下，每次本模块复位后的第一次操作必须写入新的随机数，其余场合可以选择省略本步骤）
- 2) 如果是在芯片测试模式下（**test_enb_i=0**），根据应用需求将系统密钥写入系统密钥寄存器(**skey_reg**)中，除非复位或者重新写入，否则 **skey_reg** 中的值将被保持，如果多次运算的系统密钥相同，则在第一次写入后即可忽略本步骤；如果是在芯片应用模式下（**test_enb_i=1**），忽略本步骤。
- 3) 根据应用需求将加密密钥写入加密密钥寄存器(**mkey_reg**)中，除非复位或者重新写入，否则 **mkey_reg** 中的值将被保持。如果多次运算的密钥相同，则在第一次写入后即可忽略本步骤。
- 4) 将 128bits 明文写入数据寄存器(**data_reg**)中。
- 5) 设置 **cntrl_reg** 中的各位，包括
 - a. 将 **cntrl_reg.func_sel** 设置为 2'b00;
 - b. 保持 **cntrl_reg.step_en=0**;
 - c. 向控制寄存器中的 **cntrl_reg.start** 写入 1，启动模块进行运算以上的步骤 a，b，c 可同时进行。
- 6) 判断模块运算是否结束。

不断读取 **cntrl_reg.start**，如果其值变为 0，则表示运算结束。
- 7) 读取数据寄存器(**data_reg**)，获得 128 位密文。

如果要继续进行新的运算，回到步骤 1)。

26.5.3 标准解密流程

- 1) 将 64bits 随机数写入 `rand_reg` 中（在有抗 DPA 要求的应用场合下，每次本模块复位后的第一次操作必须写入新的随机数，其余场合可以选择省略本步骤）
- 2) 如果是在芯片测试模式下（`test_enb_i=0`），根据应用需求将系统密钥写入系统密钥寄存器(`skey_reg`)中，除非复位或者重新写入，否则 `skey_reg` 中的值将被保持，如果多次运算的系统密钥相同，则在第一次写入后即可忽略本步骤；如果是在芯片应用模式下（`test_enb_i=1`），忽略本步骤。
- 3) 根据应用需求将加密密钥写入加密密钥寄存器(`mkey_reg`)中，除非复位或者重新写入，否则 `mkey_reg` 中的值将被保持。如果多次运算的密钥相同，则在第一次写入后即可忽略本步骤。
- 4) 将 128bits 密文写入数据寄存器(`data_reg`)中。
- 5) 设置 `cntrl_reg` 中的各位，包括
 - a. 将 `cntrl_reg.func_sel` 设置为 2'b01;
 - b. 保持 `cntrl_reg.step_en=0`;
 - c. 向控制寄存器中的 `cntrl_reg.start` 写入 1，启动模块进行运算。

以上的步骤 a, b, c 可同时进行。

- 6) 判断模块运算是否结束。

不断读取 `cntrl_reg.start`，如果其值变为 0，则表示运算结束。

- 7) 读取数据寄存器(`data_reg`)，获得 128 位运算结果。

如果要继续进行新的运算，回到步骤 1)。

26.5.4 密钥扩展流程

- 1) 将 64bits 随机数写入 `rand_reg` 中（在有抗 DPA 要求的应用场合下，每次本模块复位后的第一次操作必须写入新的随机数，其余场合可以选择省略本步骤）
- 2) 如果是在芯片测试模式下（`test_enb_i=0`），根据应用需求将系统密钥写入系统密钥寄存器(`skey_reg`)中，除非复位或者重新写入，否则 `skey_reg` 中的值将被保持，如果多次运算的系统密钥相同，则在第一次写入后即可忽略本步骤；如果是在芯片应用模式下（`test_enb_i=1`），忽略本步骤。
- 3) 根据应用需求将加密密钥写入加密密钥寄存器(`mkey_reg`)中，除非复位或者重新写入，否则 `mkey_reg` 中的值将被保持。如果多次运算的密钥相同，则在第一次写入后即可忽略本步骤。
- 4) 设置 `cntrl_reg` 中的各位，包括
 - a. 将 `cntrl_reg.func_sel` 设置为 2'b10;
 - b. 保持 `cntrl_reg.step_en=0`;
 - c. 向控制寄存器中的 `cntrl_reg.start` 写入 1，启动模块进行运算。

以上的步骤 a, b, c 可同时进行。

- 5) 判断模块运算是否结束。

不断读取 `cntrl_reg.start`，如果其值变为 0，则表示运算结束。

密钥扩展后的结果保存在解密密钥寄存器中。

26.5.5 简化解密流程

- 1) 将 64bits 随机数写入 `rand_reg` 中（在有抗 DPA 要求的应用场合下，每次本模块复位后的第一次操作必须写入新的随机数，其余场合可以选择省略本步骤）
- 2) 将 128bits 密文写入数据寄存器(`data_reg`)中。
- 3) 设置 `cntrl_reg` 中的各位，包括
 - a. 将 `cntrl_reg.func_sel` 设置为 2'b11;
 - b. 保持 `cntrl_reg.step_en=0`;
 - c. 向控制寄存器中的 `cntrl_reg.start` 写入 1，启动模块进行运算。以上的步骤 a, b, c 可同时进行。
- 4) 判断模块运算是否结束。
不断读取 `cntrl_reg.start`，如果其值变为 0，则表示运算结束。
- 5) 读取数据寄存器(`data_reg`)，获得 128 位运算结果。
如果要继续进行新的运算，回到步骤 1)。

26.5.6 单步测试流程

在芯片测试模式下(`test_enb_i=0`)，可以启动单步测试流程对每个功能进行测试，每启动一次运算一个 cycle。

26.6 寄存器说明

SM4 模块的寄存器列表见表 26-1。

ADDR	名 称	说 明	极性	复位值
SM4_Base + 0x00	控制寄存器 SM4_CNTRL	包括启动、加解密和密钥扩展功能选择，以及密钥选择	不带极性存储	0
SM4_Base + 0x04	RFU	-	-	-
SM4_Base + 0x08 ~ 0x0C	随机数寄存器 SM4_RAND0 ~ 1	存放本模块安全功能所需的64比特随机数	不带极性存储	某一固定值
SM4_Base + 0x10 ~ 0x1F	数据寄存器 SM4_DATA0 ~ 3	存放SM4算法的128比特明文/密文	带极性存储	0
SM4_Base + 0x20 ~ 0x2F	加密密钥寄存器 SM4_MKEY0 ~ 3	存放SM4算法的128比特加密密钥	带极性存储	0
SM4_Base + 0x30 ~ 0x3F	系统密钥寄存器 SM4_SKEY0 ~ 3	存放SM4算法的128比特系统密钥	带极性存储	

表 26-1SM4 模块的寄存器列表

SM4_Base = 0x40025800

说明：

1. 对所有寄存器的写操作只在本模块处于空闲状态 (SM4_CNTRL.start=0) 时有效，否则忽略写操作；
2. 对除控制寄存器 cntrl_reg 以外的寄存器的读操作只有在本模块处于空闲状态 (SM4_CNTRL.start=0) 时才能读出有效数据，否则读出全 0 的数据；对控制寄存器 SM4_CNTRL 的读操作可随时进行，都可读出有效数据。

26.6.1 SM4 控制寄存器 (SM4_CNTRL)

地址: SM4_BASE_ADDR + 0x00

读写: 可读/可写

极性: 本寄存器不带极性存储, 读出的极性取决于输入的极性信号 polarity_i。

功能: 详见下表

*参考 table 后说明。

BIT	定义	说明	复位值	Chip Mode	
				Test	Apply
[0]	启动/结束 (start)	0: 本模块运算结束或未被启动 1: 启动本模块进行运算	1'b0	WR	WR
[2] – [1]	功能选择 (func_sel)	00: 加密运算 01: 标准解密运算 10: 密钥扩展运算 11: 简化解密运算	2'b00	WR	WR
[3]	单步使能 (step_en)	0: 关闭单步测试功能 1: 打开单步测试功能	1'b0	WR	RO (固定为0)
[31] – [4]	RFU	-	-	-	-

说明:

- 1) 本寄存器中的启动 (start) 位的操作方法是: 软件对本位写入 1 后, 本模块将启动运行, 本次运行结束后本模块硬件会自动将本位清 0, 软件查询到本位为 0 即表示本次运行完成。
- 2) 单步功能只有在芯片测试模式下 (test_enb_i=0) 才能打开, 在芯片应用模式下 (test_enb_i=1) 始终处于关闭状态。

26.6.2 SM4 随机数寄存器 (SM4_RAND)

地址: SM4_BASE_ADDR + 0x08 – RAND_REG[31: 0]

SM4_BASE_ADDR + 0x0C – RAND_REG[63: 32]

读写: 可读/可写

极性: 本寄存器不带极性存储, 读出的极性取决于输入极性信号 polarity_i。

功能: 详见说明

复位值：某一固定值

说明：

1. 随机数寄存器 rand_reg 由两个 32 位的寄存器组成 64 位 LSFR，用于在加解密过程中随机的选择 4 个 SBOX 的组合以及数据和密钥的 mask。为了有效抵御差异功耗分析，在每次异步复位后必须输入新的随机数，并建议在每次启动运算前都输入新的随机数，每次输入的 64bits 随机数必须保证任何一个字节不等于 0。

26.6.3 SM4 数据寄存器（SM4_DATA）

地址： SM4_BASE_ADDR +0x10 – DATA[31:0]

SM4_BASE_ADDR +0x14 – DATA[63:32]

SM4_BASE_ADDR +0x18 – DATA[95:64]

SM4_BASE_ADDR +0x1C – DATA[127:96]

读写： 可读可写

极性： 本寄存器带极性存储，读出的极性取决于存储时的极性（运算过程中会反复存取，因此启动运算之后的极性由最后一次运算存储时的极性决定）。

功能： 由 4 个 32 位寄存器拼接而成的 128 位数据寄存器。用于在模块运算前写入需要被加密的明文或者需要被解密的密文，并且，运算完成后通过读取本寄存器得到加密后的密文或者解密后的明文。

**参考 table 后说明.*

BIT	名称	说明	复位值
[127:0]	数据寄存器 (DATA_REG)	128位明文/密文	0

说明：

数据寄存器由四个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且，运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128 位明文	128 位密文	128 位密文	128 位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下：

数据举例：

128'hA15_A14_A13_A12_A11_A10_A9_A8_A7_A6_A5_A4_A3_A2_A1_A0

	data[31:0]	data[63:32]	data[95:64]	data[127:96]
0	A15	A11	A7	A3
7	A14	A10	A6	A2
15	A13	A9	A5	A1
23	A12	A8	A4	A0
31				

每个 32 位的寄存器成为了矩阵的一列，每一列由 4 个字节组成。

26.6.4 SM4 加密密钥寄存器（SM4_MKEY）

地址： SM4_BASE_ADDR +0x20 – MKEY[31:0]

SM4_BASE_ADDR +0x24 – MKEY[63:32]

SM4_BASE_ADDR +0x28 – MKEY[95:64]

SM4_BASE_ADDR +0x2C– MKEY[127:96]

读写：可读可写

极性：本寄存器带极性存储，读出的极性取决于存储时的极性。

功能：由 4 个 32 位寄存器拼接而成的 128 位加密密钥寄存器。用于存放本次运算所需的加密密钥（MKEY）。

*参考 table 后说明。

BIT	名称	说明	复位值
[127:0]	加密密钥寄存器 (MKEY_REG)	存放128位加密密钥	0

说明：

- 1) 加密密钥寄存器由 4 个 32 位的寄存器组成，存放输入的加密密钥。
- 2) 写操作时需要分别对 4 个 32 位的寄存器进行操作。对应的操作顺序与数据寄存器一致。

26.6.5 SM4 系统密钥寄存器 (SM4_SKEY)

地址：SM4_BASE_ADDR +0x30 – SKEY[31:0]

SM4_BASE_ADDR +0x34 – SKEY[63:32]

SM4_BASE_ADDR +0x38 – SKEY[95:64]

SM4_BASE_ADDR +0x3C – SKEY[127:96]

读写：可读可写（芯片测试模式）/ 不可见（芯片应用模式）

极性：本寄存器带极性存储，读出的极性取决于存储时的极性。

功能：由 4 个 32 位寄存器拼接而成的 128 位系统密钥寄存器。

*参考 table 后说明。

BIT	名称	说明	复位值	Chip Mode	
				Test	Apply
[127:0]	系统密钥寄存器 (SKEY_REG)	存放128位系统密钥	某一固定值	RW	-

说明：

- 1) 系统密钥寄存器由 4 个 32 位的寄存器组成，存放输入的系统密钥或初始系统密钥。写操作时需要分别对 4 个 32 位的寄存器进行操作。对应的操作顺序与数据寄存器一致。
- 2) 本寄存器的读写受到端口 test_enb_i 控制。当 test_enb_i 为低电平时，允许进行读写操作，否则硬件将自动屏蔽对本寄存器的读写操作。
- 3) 本寄存器的复位受到端口 test_enb_i 控制。当 test_enb_i 为低电平时，允许对本寄

寄存器进行异步复位（通过 hresetn），否则硬件将自动屏蔽对本位的异步复位操作。

27. 7816 控制器

27.1 功能概述

SCI7816 为异步半双工串行通讯端口，符合 ISO/IEC 7816-3 标准，功能和特性如下：

- 支持 T=0 和 T=1 的传输模式；
- 收发自动转换；
- 支持 Retry 次数设置；
- 支持额外的 ETU 设置；
- 支持 ETU counter 计数；
- 支持硬件 LRC/CRC 计算；
- 起始位采用 16 次采样判决算法；
- 每一个 Bit 数据三次采样并遵从择多判决算法；
- 支持 SCI7816 接口软复位；
- 支持接收数据未取走情况下，被新数据覆盖（overrun）标志
- 支持 SCI7816 接口状态（busy/idle）标志

SCI7816 接口有以下中断源：

- 发送/接收数据完成（包括成功和失败）
- ETU counter 中断

27.1.1 SCI7816 通信速率设置

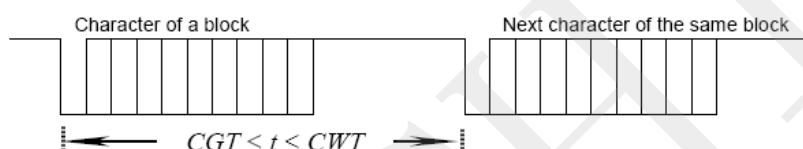
SCI7816 模块通信速率由 SBDR 寄存器控制。该寄存器的值为一个 ETU 所对应的外部时钟周期数。具体寄存器设置与 7816 模块通信速率的对应关系如下表所示：

Fi/Di	SBDR[16:0]	Baud Rate(3.57MHz)	Baud Rate(5MHz)
372	0x0174	9.6bps	13.4kbps

64	0x0040	57.6kbps	80.7kbps
32	0x001F	115.2kbps	161.3kbps
16	0x0010	230.4kbps	320.6kbps

27.1.2 SCI7816 字符等待时间

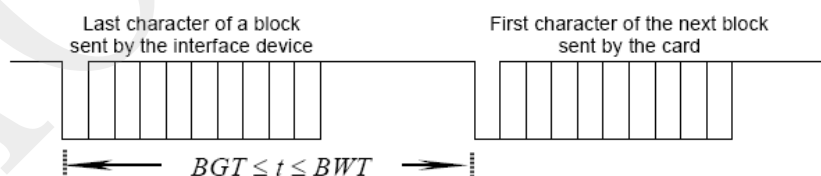
字符等待时间 CWT (Character Waiting Time) 表示相同传输方向上连续两个字符间起始沿的最大延迟。



在 T=0 模式下, 字符等待时间 $WT = WI * 960 * \frac{F}{f}$, (WI=0~255, 默认为 10), 该默认值为 9600 etu; 在 T=1 模式下, 字符等待时间 $CWT = 11etu + 2^{CWI}$, (CWI=0~15, 默认为 13), 该默认值为 8203 etu;

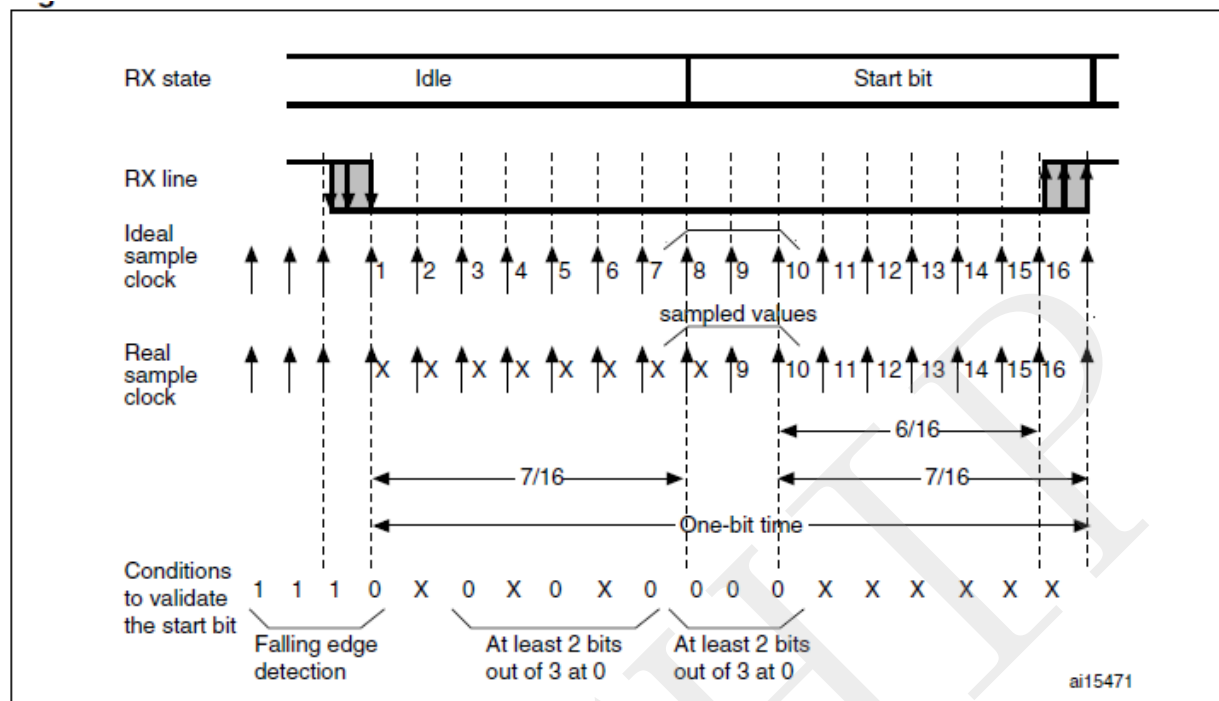
27.1.3 SCI7816 块保护时间

块保护时间 BGT (Block Guard Time) 表示在 T=1 模式下两个反向字符第一个沿之间的最小延迟。如下图所示:



27.1.4 起始位采样

起始位采样采用 16 次采样, 采样图形如下:



采样判决方式说明如下：

- 两组采样点中，全部 6 个采样点均为 0（第 3、5、7 采样点处为 0，第 8、9、10 采样点处为 0），则起始位采样有效，不置起 RX_NE 标志。
- 对两组采样点中，两组里的 3 个采样点有 2 个为 0（第一组采样点：第 3、5、7 采样点；第二组采样点：第 8、9、10 采样点），起始位采样有效，并置起 RX_NE 标志。
- 如果以上采样条件不满足，退出起始位采样，并使 7816 接口回到等待状态。

27.2 软件操作流程

27.2.1 初始化配置

- 1) 清除 SCI_STATUS 寄存器里的中断标志;
- 2) 设置 SCI_MODE 寄存器;
- 3) 配置中断优先级, 使能 7816 中断;
- 4) 设置 SCI_CON 寄存器, 使能 SCI7816 模块, 及配置其他功能;
- 5) 配置 SCI_SBDR 寄存器;
- 6) 配置 SCI_ETU_CNT 寄存器。

27.2.2 发送（中断方式，无 FIFO）

- 1) 初始化配置;
- 2) 将待发送数据写入 SCI_DATA 寄存器中;
- 3) 当数据发送成功或失败（超过重试设置），则会产生中断，则软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功;
- 4) 如继续发送，则重复步骤 2-3;
- 5) 数据发送过程结束后，硬件自动把 7816 模块设成硬件接收模式，不需要软件设置。

27.2.3 发送（轮询方式，无 FIFO）

- 1) 初始化配置;
- 2) 将待发送数据写入 SCI_DATA 寄存器中;
- 3) 软件查询 SCI_STATUS 标志，若数据传输成功则硬件会置 TX_FIN 标志，若失败则置 TX_ERR 标志；程序查询到两标志之一置起后，需把相应标志清 0，并读取 SCI_STATUS 标志寄存器确认清除成功;
- 4) 如继续发送，重复步骤 2-3;
- 5) 数据发送过程结束后，硬件自动把 7816 模块设成硬件接收模式，不需要软件设置。

27.2.4 接收（中断方式，无 FIFO）

- 1) 初始化配置；
- 2) 当接收了一个字节数据成功或失败之后，会产生中断请求，若数据接收成功则硬件会置 `RX_FIN` 标志，若失败则置 `RX_ERR` 标志；软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功；
- 3) 若接收成功，则软件应从 `SCI_DATA` 中读取收到的数据；若接收失败，则进行出错处理；
- 4) 继续接收重复步骤 2-3。

27.2.5 接收（轮询方式，无 FIFO）

- 1) 初始化配置；
- 2) 软件查询 `SCI_STATUS` 标志，当接收了一个字节数据成功或失败之后，会产生中断请求，若数据接收成功则硬件会置 `RX_FIN` 标志，若失败则置 `RX_ERR` 标志；软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功；
- 3) 若接收成功，则软件应从 `SCI_DATA` 中读取收到的数据；若接收失败，则进行出错处理；
- 4) 继续接收重复步骤 2-3。

27.2.6 发送（中断方式，有 FIFO）

- 1) 初始化配置；
- 2) 将待发送数据写入 `SCI_DATA` 寄存器中；
- 3) 每向 `SCI_DATA` 寄存器写一次数据，查询一次 `SCI_FIFO_STATUS` 标志，如果没有置位，继续写入，如果置位，停止写入；
- 4) 当 FIFO 中的数据全部发送成功，则会产生中断，软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功；
- 5) 数据发送失败（超过重试设置），也会产生中断，软件应首先读取 `SCI_STATUS`，判断中断是否是 `TX_ERR`，之后读取 `SCI_FIFO_DEPTH`，计算还有几个数没发出去，然后清空 FIFO，之后清除中断标志，并读取中断标志寄存器确认中断清除

成功;

- 6) 如继续发送, 则重复步骤 2-5;
- 7) 数据发送过程结束后, 硬件自动把 7816 模块设成硬件接收模式, 不需要软件设置。

27.2.7 发送 (轮询方式, 有 FIFO)

- 1) 初始化设置;
- 2) 将待发送数据写入 SCI_DATA 寄存器中;
- 3) 每向 SCI_DATA 寄存器写一次数据, 查询一次 SCI_FIFO_STATUS 寄存器的 TX_FUL 标志, 如果没有置位, 继续写入, 如果置位, 停止写入;
- 4) 软件查询 SCI_STATUS 标志, 若 FIFO 中数据全部传输成功则硬件会置 TX_FIN 标志, 程序查询到标志置起后, 需把相应标志清 0, 并读取 SCI_STATUS 标志寄存器确认清除成功;
- 5) 软件查询 SCI_STATUS 标志, 如果查询到错误标志, 软件应首先读取 SCI_STATUS, 判断中断是否是 TX_ERR, 之后读取 SCI_FIFO_DEPTH, 计算还有几个数没发出去, 然后清空 FIFO, 之后清除中断标志, 并读取中断标志寄存器确认中断清除成功;
- 6) 如继续发送, 重复步骤 2-5;
- 7) 数据发送过程结束后, 硬件自动把 7816 模块设成硬件接收模式, 不需要软件设置。

27.2.8 接收 (中断方式, 有 FIFO)

- 1) 初始化配置;
- 2) 当接收了一个字节数据成功或失败之后, 会产生中断请求, 若数据接收成功则硬件会置 RX_FIN 标志, 若失败则置 RX_ERR 标志; 软件应首先清除中断标志, 并读取中断标志寄存器确认中断清除成功;
- 3) 若接收成功, 则软件应从 SCI_DATA 中读取收到的数据; 若接收失败, 则进行出错处理;
- 4) 继续接收重复步骤 2-3。

27.2.9 接收（轮询方式，有 FIFO）

- 1) 初始化设置；
- 2) 软件查询 SCI_STATUS 标志，当接收了一个字节数据成功或失败之后，会产生中断请求，若数据接收成功则硬件会置 RX_FIN 标志，若失败则置 RX_ERR 标志；软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功；
- 3) 若接收成功，则软件应从 SCI_DATA 中读取收到的数据；若接收失败，则进行出错处理；
- 4) 继续接收重复步骤 2-3。

27.2.10 ETU 计数器的使用

SCI7816 提供 ETU 计数器的功能，软件可以根据需要设置 SCI_ETU_CNT，产生中断后做相应的处理。

使用 ETU 计数器的情况有以下几种：

- 1) 计算 CWT 超时等待时间
- 2) 计算 BGT 等待时间
- 3) 计算额外的保护时间（EXTRA ETU）

CWT 应用：软件按需要设置 SCI_ETU_CNT 为 CWT 的值，如果接收两个字节数据之间间隔超过 CWT，则会产生 ETU 计数器中断，软件可以在 ETU 计数器中断服务函数中做相应的超时处理。

GT 应用：T=1 模式下，软件按需要设置 SCI_ETU_CNT 为 BGT 的值，在接收到一个包数据后，必须等 ETU 计数器中断产生后，才可以发送数据包，用以满足块保护时间的要求。

EXTRA ETU 应用：SCI7816 提供 0/1/2/4 四种 EXTRA ETU 的可选配置，但如果有其他需要，可以通过设置 SCI_ETU_CNT 并等待中断的方法，满足特殊的 EXTRA ETU 需求。

27.3 寄存器定义

SCI7816 一共包含 12 个特殊寄存器，寄存器列表如下：

基地址：0x40010C00

名称	位宽	读写	复位值	描述
SCI_CON	32	R/W	32' h31	SCI7816控制寄存器
SCI_MODE	32	R/W	32' h0	SCI7816模式配置寄存器
SCI_STATUS	32	R/W	32' h0	SCI7816状态寄存器
SCI_FIFO_STATUS	32	R	32' h0	SCI7816 FIFO状态寄存器
SCI_FIFO_DEPTH	32	R	32' h0	SCI7816 FIFO深度寄存器
SCI_DATA	32	R/W	32' h0	SCI7816收发数据寄存器
EDC_DATA	32	R/W	32' h0	SCI7816校验数据寄存器
SBDR	32	R/W	32' d372	SCI7816波特率配置寄存器
ETU_CNT	32	R/W	32' d9600	SCI7816 ETU计数寄存器

表 27-1 7816 控制器寄存器列表

27.3.1 控制寄存器（SCI_CON）

地址偏移量：0x00

复位值：0x31

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8								
Reserved																															
7				6				5				4				3				2				1				0			
TX_FIFO_CLR				RX_FIFO_CLR				ETU_CNT_EN				EDC_EN				RETRY_EN				PARITY_CHK_EN				F1_EN				SCI_EN			
RW				RW				RW				RW				RW				RW				RW				RW			

		注：仅在 T=1 模式下有效。在该功能使能后，在接收到校验正确的数据，或者在发送数据时，硬件会自动计算 EDC 的值并保存在 EDC_DATA 寄存器中。
3	RETRY_EN	SCI7816 自动重收发功能使能位。 1：使能 0：关闭
2	PARITY_CHK_EN	SCI7816 校验位检查使能位。 1：使能 0：关闭
1	FI_EN	SCI7816 强制输出高电平使能位。 1：使能 0：关闭 注：此功能使能后，接收数据时，若发生奇偶校验出错，在请求重发的一个 ETU 低电平之前和之后，SCI7816 硬件会强制输出一个时钟周期的高电平；发送数据时，在起始位开始前和数据位之后，SCI7816 硬件会强制输出一个时钟周期的高电平。在该功能不使能的情况下，IO PAD 在输入状态时，只能靠上拉保持高电平，因此电平转换的上升时间或下降时间有可能超出规范（具体数据依赖于系统设计方案）；使能该功能，可以在电平转换时先将 IO PAD 电平强制拉高，之后切换为上拉控制，这样可以改善电平转换时间。
0	SCI_EN	SCI7816 模块使能位。 1：使能 0：关闭 注：SCI_EN 清零后，所有状态位恢复默认值。

注：T=0 模式及 T=1 模式下的校验检查及重发机制说明。

➤ T=0 模式：

RETRY_EN	PARITY_CHK_EN	功能
0	0	不做校验检查，不重试，不会产生错误标志。在发送数据时，若在11ETU处采样到对方的重发请求，硬件不重发，不产生发送错误标志/中断；在接收数据时，不校验接收的数据，硬件不请求重发，不产生接收错误标志/中断。
0	1	做校验检查，不重试，如校验出错则产生错误标志。在发送数据时，若在11ETU处采样到对方的重发请求，硬件不重发，并产生发送错误标志/中断；在接收数据时，若接收到奇偶校验错误的字节，硬件不请求重发，并产生接收错误标志/中断。
1	0	-
1	1	做校验检查，出错重试，如重试次数超出设置，则产生错误标志。在发送数据时，若在11ETU处采样到对方的重发请求（且未超过重试次数设置），则硬件自动重发之前发送的最后一个字节数据，且重试次数加1；若超过重试次数设置，则硬件不再重发，并产生发送错误标志/中断；在接收数据时，若接收到奇偶校验错误的字节（且未超过重试次数设置），硬件自动在10.5-11.5 ETU处将IO拉低一个ETU的时间，用来请求重发，且重试次数加1；若超过重试次数设置，则硬件不再请求重发，并产生接收错误标志/中断。

➤ T=1 模式：

（T=1 下不允许将 RETRY_EN 置 1，RETRY_EN 控制位无效，根据 7816-3 协议奇偶校验出错不会重新收发每个字节的数据）

PARITY_CHK_EN	功能
0	在发送数据时，不产生发送错误标志/中断；在接收数据时，不产生接收错误标志/中断。
1	在发送数据时，不会产生发送错误标志/中断；在接收数据时，若接收到奇偶校验错误的字节，据协议不会产生重发请求，在10.5个ETU处产生接收错误标志/中断。

27.3.2 模式配置寄存器（SCI_MODE）

地址偏移量：0x04

复位值：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
RETRY_TIMES				EDC_MODE	PAD_MODE	CDCVT	PARITY	T_MODE															
RW				RW	RW	RW	RW	RW															

位	标记	功能描述
31:8	RESERVED	保留
7:5	RETRY_TIMES	自动重收发次数。在PARITY_CHK_EN=1且RETRY_EN=1时有效。 000：允许无限次自动重收发，出错标志不会置位 001：允许一次自动重收发，如仍有奇偶校验错则停止收发，出错标志被置位 010：允许二次自动重收发，如仍有奇偶校验错则停止收发，出错标志被置位 111：允许七次自动重收发，如仍有奇偶校验错则停止收发，出错标志被置位
4	EDC_MODE	校验模式选择位 0：LRC校验 1：CRC校验 注：仅T=1模式下有效。LRC校验为1字节，存在寄存器EDC_DATA的低8位中；CRC校验为2字节，存在寄存器EDC_DATA中。CRC16校验算法参照ISO/IEC 13239，算法多项式为 $x^{16}+x^{12}+x^5+1$ ，初始值为全FFFF，输出取反。
3	PAD_MODE	IO PAD模式选择位 0：IO PAD为上拉模式 1：IO PAD为开漏模式
2	CDCVT	传输方向选择位 0：正向传输 1：反向传输
1	PARITY	校验机制选择位 0：偶校验 1：奇校验
0	T_MODE	7816模式选择位 0：T=0模式 1：T=1模式 注：T=1模式下，收发数据字符间隔默认为11ETU，在T=0模式下为12ETU；T=1模式下，RETRY功能不可用。

27.3.3 状态寄存器 (SCI_STATUS)

地址偏移量: 0x08

复位值: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
RX_NE		RX_OVR		SCI_STAT		EDC_CNT_FIN		TX_ERR		TX_FIN		RX_ERR		RX_FIN									
RW		RW		RW		RW		RW		RW		RW		RW									

位	标记	功能描述
31:8	Reserved	保留
7	RX_NE	接收起始位噪声标志位。 1: 接收起始位存在噪声 0: 接收起始位不存在噪声 注: 接收数据开始后, 起始位采样有效, 但6个采样点不是全部为0, 则会置位接收起始位噪声标志, 可做查询使用。在下一一次接收数据的起始位后, 如果起始位无效, 或者6个采样点全部为0 (无噪声), 则该标志自动清零。
6	RX_OVR	接收数据被覆盖标志位。 1: 接收数据被覆盖 0: 接收数据未被覆盖 注: 在接收数据时, 若没有及时取走buffer中的数据, 在正确接到新的数据时, 在接收起始位下降沿开始11.5个ETU时自动置位, 表明有数据被覆盖; 置位RX_FIFO_CLR可将此位清零。
5	SCI_STAT	SCI7816工作状态标志位。 1: 在工作 0: 空闲 注: 在接收数据时, 从接收起始位下降沿起, 12个ETU内一直为1, 12个ETU后自动清零, 直到接收新的字节, 再次被硬件置位; 在发送数据时, 从发送起始位下降沿起, 12个ETU内一直为1, 12个ETU后自动清零, 直到发送新的字节开始, 再次被硬件置位。
4	EDC_CNT_FIN	ETU计数器结束标志位。 1: ETU计数器达到SCI_ETU_CN设置的值 0: ETU计数器未达到SCI_ETU_CN设置的值 注: 置位后如果有新的数据收发, 则该标志位被清零, ETU计数器重新开始计数; 也可以由软件写0清除, 软件写1无效。
3	TX_ERR	发送错误标志位。 1: 发送错误 0: 无发送错误

		注：T=0模式下，因奇偶校验错，发送重试次数超过设置的次数，则此标志在最后一次发送起始位下降沿后11.5 ETU时被置位，须由软件写0清除，软件写1无效；T=1模式下，该标志位无效。
2	TX_FIN	发送一个字节结束标志位。 1：发送结束 0：发送还未结束 注：若成功发送一个字节数据，在起始位下降沿后11.5 ETU（T=0）/10.5 ETU（T=1）时被置位，由软件写0清除，软件写1无效。
1	RX_ERR	接收错误标志位。 1：接收错误 0：无接收错误 注：T=0模式下，因奇偶校验错，接收重试次数超过设置的次数，则此标志在最后一次接收起始位下降沿后11.5 ETU时被置位，须由软件写0清除，软件写1无效；T=1模式下，该标志位无效。
0	RX_FIN	接收一个字节结束标志位。 1：接收结束 0：接收还未结束 注：若成功接收一个字节数据，在起始位下降沿后11.5 ETU（T=0）/10.5 ETU（T=1）时被置位，由软件写0清除，软件写1无效。

27.3.4 FIFO 状态寄存器 (SCI_FIFO_STATUS)

地址偏移量: 0x0C

复位值: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7				6				5				4				3				2			
TX_OVR				TX_UDR				TX_FUL				TX_EPT				RX_OVR				RX_UDR			
R/W				R/W				R/W				R/W				R/W				R/W			

位	符号	功能描述
31:8	RESERVED	保留
7	TX_OVR	发送FIFO溢出 (overflow) 标志位。 1: 溢出 0: 未溢出 注: 置位TX_FIFO_CLR可将此位清零。
6	TX_UDR	发送FIFO下溢 (underrun) 标志位。 1: 下溢 0: 未下溢 注: 置位TX_FIFO_CLR可将此位清零。
5	TX_FUL	发送FIFO满 (FULL) 标志位。 1: 满 2: 未满 注: 置位TX_FIFO_CLR可将此位清零。
4	TX_EPT	发送FIFO空 (EMPTY) 标志位。 1: 空 2: 未空 注: 置位TX_FIFO_CLR可将此位清零。
3	RX_OVR	接收FIFO溢出 (overflow) 标志位。 1: 溢出 0: 未溢出 注: 置位RX_FIFO_CLR可将此位清零。
2	RX_UDR	接收FIFO下溢 (underrun) 标志位。 1: 下溢 0: 未下溢 注: 置位RX_FIFO_CLR可将此位清零。
1	RX_FUL	接收FIFO满 (FULL) 标志位。 1: 满 2: 未满

		注：置位RX_FIFO_CLR可将此位清零。
0	RX_EPT	接收FIFO空（EMPTY）标志位。 1：空 2：未空 注：置位RX_FIFO_CLR可将此位清零。

注：对 SCI_DATA 寄存器进行读写可能影响 SCI_FIFO_STATUS 所有状态位，请注意以下操作：

- 在 RX_EPT=0 时，不可以读取 SCI_DATA 寄存器；
- 在 RX_FUL=1 时，需要立刻读取 SCI_DATA 寄存器中的数据；
- 在 TX_FUL=1 时，不可以写入 SCI_DATA 寄存器；
- 在 RX_UDR/RX_OVR/TX_UDR/TX_OVR 任何一位被置位后，软件需要立刻置位 RX_FIFO_CLR 或 TX_FIFO_CLR，清空 FIFO 将所有状态位恢复默认值；

对 SCI_DATA 寄存器进行读写，影响 SCI_FIFO_STATUS 状态位有一定延时（硬件需要做跨时钟同步），软件需要限制单次操作 SCI_DATA 寄存器的次数，不可超过 FIFO 本身的深度，具体操作如下：

- 查询 RX_EMP 从 0 变 1 时，只可以读出一个字节数据；
- 查询 RX_EMP=0 时，可以通过查询 RX_FIFO_DEPTH 得到接收 FIFO 中存储的字节数，再全部读出；
- 查询 RX_FUL=1 时，最多读出 X 字节数据（X=FIFO 大小）；
- 查询 TX_EMP=0 时，最多可以写入 X 字节数据（X=FIFO 大小）；
- 查询 TX_FUL=0 时，可以通过查询 TX_FIFO_DEPTH 得到发送 FIFO 中存储的字节数，继续写入 N 字节数据（N=X-TX_FIFO_DEPTH）；

27.3.5 FIFO 深度寄存器 (SCI_FIFO_DEPTH)

地址偏移量: 0x10

复位值: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8								
Reserved																															
7				6				5				4				3				2				1				0			
TX_FIFO_DEPTH												RX_FIFO_DEPTH																			
R												R																			

位	符号	功能描述
31:8	Reserved	保留
7:4	TX_FIFO_DEPTH	发送FIFO深度, 表示发送FIFO中存储的待发送数据字节数。
3:0	RX_FIFO_DEPTH	接收FIFO深度, 表示接收FIFO中存储的已接收数据字节数。

27.3.6 收发数据寄存器 (SCI_DATA)

地址偏移量: 0x14

复位值: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8								
Reserved																															
7				6				5				4				3				2				1				0			
SCI_DATA																															
RW																															

位	符号	功能描述
31:8	Reserved	保留
7:0	SCI_DATA	数据缓存寄存器。存放待发送的数据字节, 以及接收到的数据; 二者地址映射在一起。

27.3.7 校验数据寄存器 (EDC_DATA)

地址偏移量: 0x18

复位值: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDC_DATA															
RW															

位	符号	功能描述
31:16	Reserved	保留
15:0	EDC_DATA	校验数据寄存器。T=1模式下，存储接收到或待发送的CRC校验值或者LRC校验值。操作流程如下：发送时写入SCI_DATA的数据，硬件会自动进行计算，并将计算结果随时更新至EDC_DATA寄存器中。软件可以将整个block数据（除EDC字节外）全部写入SCI_DATA寄存器，再读取EDC_DATA寄存器，将其值写入SCI_DATA寄存器，此时EDC_DATA寄存器将自动恢复默认值0；接收时所有接收到的数据，硬件会自动进行计算，并将计算结果随时更新至EDC_DATA寄存器中。如果所有数据均接收正确，在整个block数据接收完成后，EDC_DATA寄存器最终计算结果应该为0，软件可以在完成一个block数据接收后查询EDC_DATA来校验数据，也可以在接收过程中查询EDC_DATA的值并与软件计算结果进行对比。

27.3.8 波特率配置寄存器（SCI_SBD R）

地址偏移量：0x1C

复位值：0x0174

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTRA_ETU		SBD R													
RW		RW													

位	符号	功能描述
31:16	Reserved	保留
15:14	EXTRA_ETU	发送模式下增加额外ETU个数配置位。 00：无额外ETU； 01：增加一个额外ETU； 10：增加二个额外 ETU； 11：增加四个额外ETU。 注：无额外ETU 情况下，T=0默认为12ETU，T=1默认为11ETU。
13:0	SBD R	波特率配置寄存器。SBD R = Fi/D，默认值0x174，表示1个ETU单位为372 CLK。若SCI7816输入时钟不是3.57MHz，需要根据频率比例计算相应的SBD R值。

27.3.9 ETU 计数寄存器 (SCI_ETU_CNT)

地址偏移量: 0x20

复位值: 0x2580

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCI_ETU_CNT															
RW															

位	符号	功能描述
31:16	Reserved	保留
15:0	SCI_ETU_CNT	ETU计数器的设定值。单位为1个ETU时间，默认值0x2580，表示9600 ETU。

27.3.10 时钟复位及中断控制寄存器 (SCI_CLKRSTINT_CON)

地址偏移量: 0x24

复位值: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved										RxSofIntEn		EtuFinIntEn			
RW										RW		RW			
7	6	5	4	3	2	1	0								
TxErrIntEn	TxFinIntEn	RxErrIntEn	RxFinIntEn	clk_sel			eclk_en		erst_ctrl						
RW	RW	RW	RW	RW			RW		RW						

位	符号	功能描述
31:10	Reserved	保留
9	RxSofIntEn	接收起始位中断使能 1: 使能 0: 关闭
8	EtuFinIntEn	ETU结束中断使能 1: 使能

		0: 关闭
7	TxErrIntEn	发送错误中断使能 1: 使能 0: 关闭
6	TxFinIntEn	发送结束中断使能 1: 使能 0: 关闭
5	RxErrIntEn	接收错误中断使能 1: 使能 0: 关闭
4	RxFinIntEn	接收结束中断使能 1: 使能 0: 关闭
3:2	clk_sel	7816时钟选择位 00: hclk/2 01: hclk/4 10: hclk/8 11: hclk/16
1	ec1k_en	7816输出时钟使能位 1: 使能 0: 关闭
0	erst_ctrl	7816输出rst控制位 1: 7816 rst端口输出1 0: 7816 rst端口输出0

28. 高级定时器（ATIM）

28.1 高级定时器简介

ATIM 是由 1 个计数单元和 6 个比较单元组成的定时器，支持 6 个独立 PWM 输出或 3 对互补 PWM 输出。支持 6 个捕获输入。

使用定时器预分频、系统预分频器和系统时钟选择，可灵活的调整脉冲宽度和波形周期；可以方便的测量脉冲宽度。

28.1.1 基本特性(ATIM)

- 6 路独立 PWM 输出 CH0A,CH0B, CH1A,CH1B, CH2A,CH2B(端口对应 CH1,CH1N, CH2,CH2N, CH3,CH3N)
- 3 路互补 PWM 输出 (CHxA,CHxB) +1 路独立 PWM 输出 (gate)

28.2.2 定时计数器

可编程通用定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

28.2.3 定时器预除频

使用 TCLK 做为 Timer 定时器时钟时，可以使用预分频。预分频设置如下：

PRS	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	256

预分频器没有预装载缓存，因此对预分频进行更改，预分频将会立刻生效。

28.2.4 模式 0 计数定时器功能

这种模式下，计数器向上计数。计数器支持两种计数模式，重载模式与自由计数模式，可以选择外部时钟 ETR 计数或系统时钟计数。门控 Gate 可以控制计数屏蔽。计数到最大值溢出产生中断。翻转输出端口 CHA,CHB，这种模式下 CHA,CHB 反相。

重载模式计数范围从 ARR 向上计数到 0xFFFF 溢出，然后从 ARR 再开始计数，计数周期为 0xFFFF-ARR+1；自由计数模式从设置计数值计数到 0xFFFFFFFF 后溢出，溢出后计数值从 0x0 重新开始计数。

计数器可用通过外部门控功能控制计数器是否计数。控制关系如下

M0CR.GATE	M0CR.GATEP	端口 GATE 输入	M0CR.CTEN	计数器
x	x	x	0	不计数
0	x	x	1	计数
1	0	高电平	1	计数
1	0	低电平	1	不计数
1	1	高电平	1	不计数
1	1	低电平	1	计数

28.2.4.1 功能框图

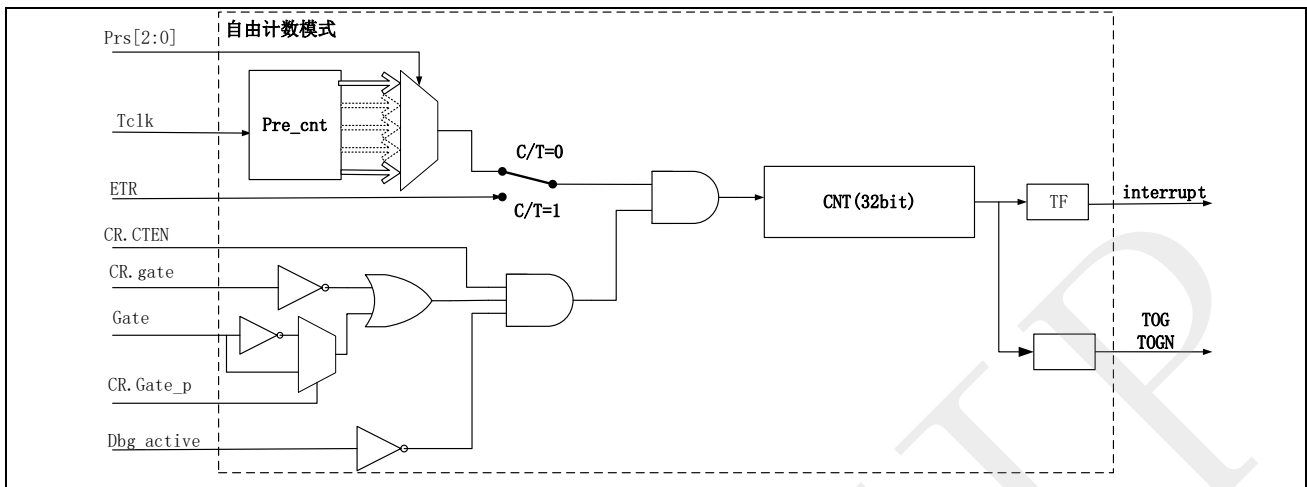


图 28-2 自由计数框图

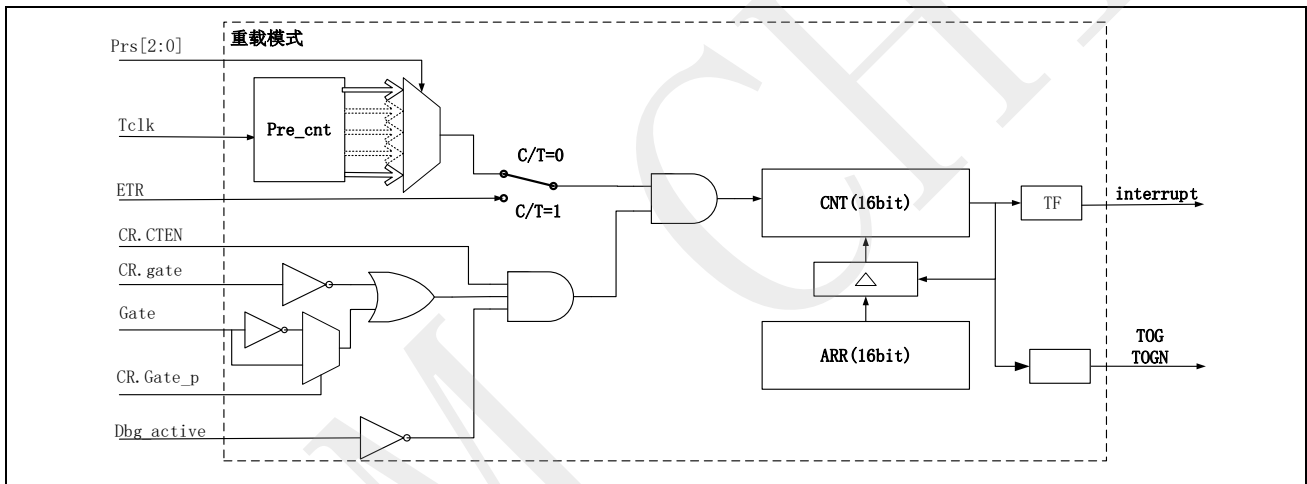


图 28-3 重载计数波形

28.2.4.2 计数波形

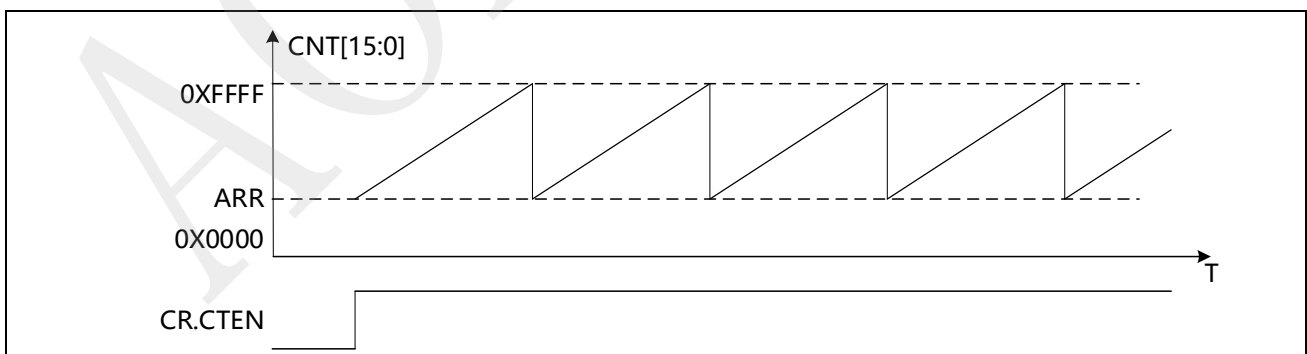


图 28-4 16位重载计数波形

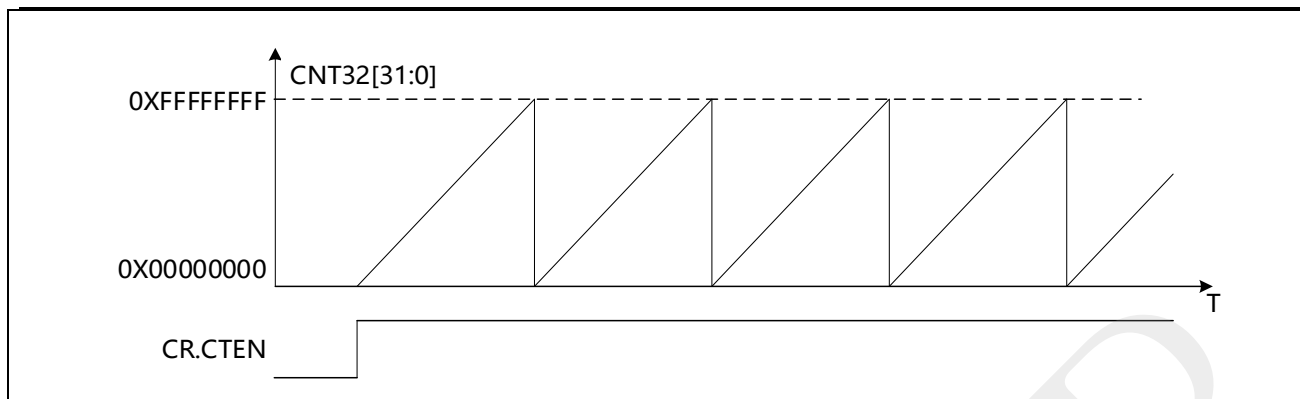


图 28-5 32位自由计数波形

28.2.4.3 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 `tclk` 采样，因此外部输入时钟频率不能超过系统的 `tclk` 时钟。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

28.2.4.4 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

28.2.4.5 时序图

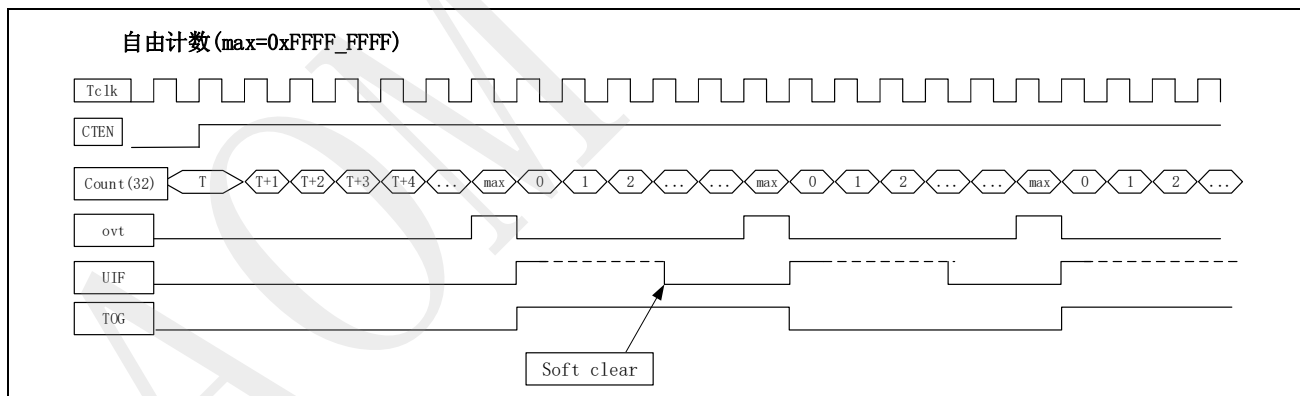


图 28-6 自由计数时序图

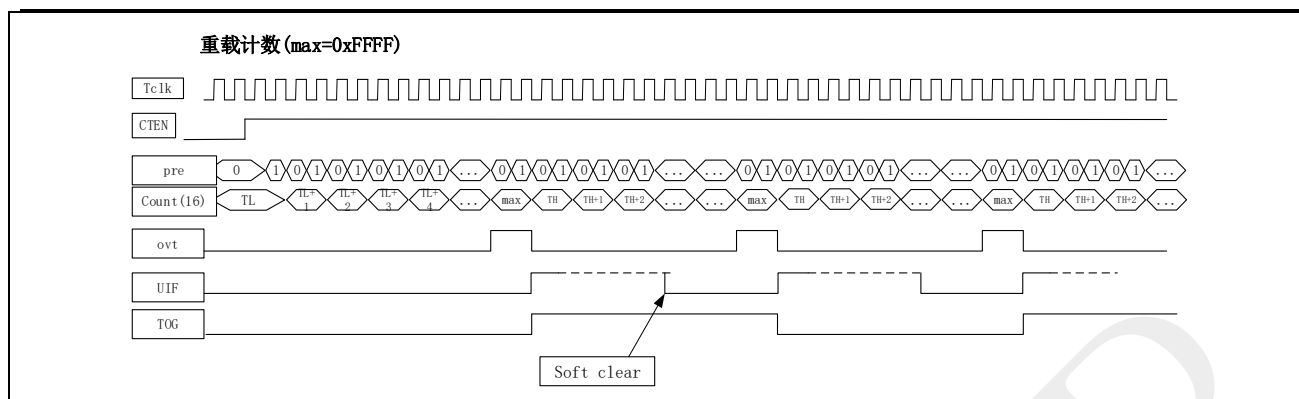


图 28-7 重载计数时序图（预分频设置为2）

28.2.4.6 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。使用 toggle 输出需要使能 DTR.MOE 控制位。设置 M0CR.TOGEN 为 0 可以同时设置端口 CHA,CHB 输出为 0。

在计数时钟为 4M 情况下 Buzzer 输出不同频率的 timer 重载模式配置如下：

Buzzer 频率	计数器周期	计数器计数值	计数器重载值	CNTL 初始值	CNTH 重载值
1000Hz	0.5ms	2000	63536	0XF830	0XF830
2000Hz	0.25ms	1000	64536	0XFC18	0XFC18
4000Hz	0.125ms	500	65036	0XFE0C	0XFE0C

28.2.4.7 设置示例

重载定时器设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置装载值 ARR
3. 设置计数器初值 CNT
4. 清除中断标志
5. 使能中断 M0CR.UIE
6. 使能重载模式 M0CR.MD
7. 开启定时器 M0CR.CTEN

门控外部时钟自由计数设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置计数器初值 CNT

3. 使能门控功能 M0CR.GATE
4. 选择门控有效电平 M0CR.GATEP
5. 清除中断标志
6. 使能中断 M0CR.UIE
7. 使能外部时钟模式 M0CR.CT
8. 开启定时器 M0CR.CTEN

BUZZER 输出控制

1. 根据输出频率设置合适的 ARR 值
2. 设置定时器为重载模式，参考重载定时器设置
3. 使能输出使能 DTR.MOE
4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出。

28.2.5 模式 1 脉宽测量 PWC

这种模式下可以自动测量输入脉冲的高电平低电平或者周期宽度。

第一个有效边沿计数器初始化为 0x0001，第二个有效边沿将停止计数，并将当前计数值存入 CMAR，并且产生捕获中断 CAF，如果计数器发生溢出，会产生溢出标志。

设置溢出中断使能会产生溢出中断。

M1CR.edg1st	0	0	1	1
M1CR.edg2nd	0	1	0	1
脉宽测量	上沿~上沿 周期宽度	上沿~下沿 高电平宽度	下沿~上沿 低电平宽度	下沿~下沿 周期宽度

周期测量时，会间隔一个周期测量一个周期。

28.2.5.1 PWC 功能框图

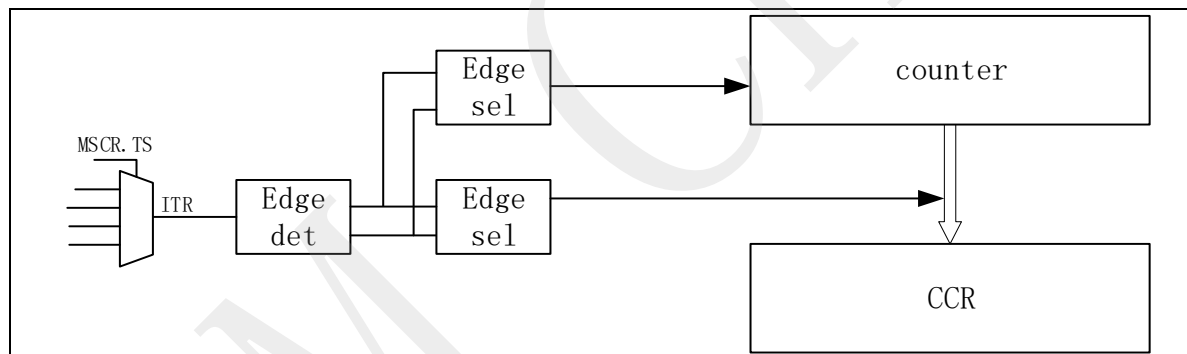


图 28-8 PWC测量框图

MSCR.TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP; 001: 内部互联信号 ITR0; 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 无效; 110: 端口CH0A的滤波后的信号IAFP（极性选择在脉宽测量模式下无效）; 111: 端口CH0B的滤波后的信号IBFP（极性选择在脉宽测量模式下无效）。

28.2.5.2 PWC 波形测量时序图

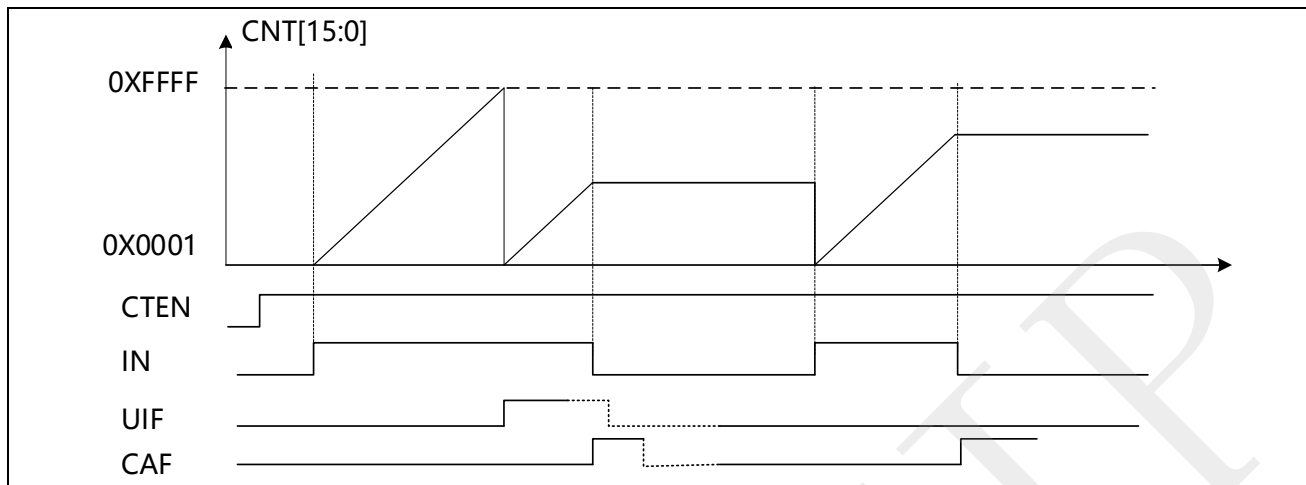


图 28-9 高电平脉冲宽度测量

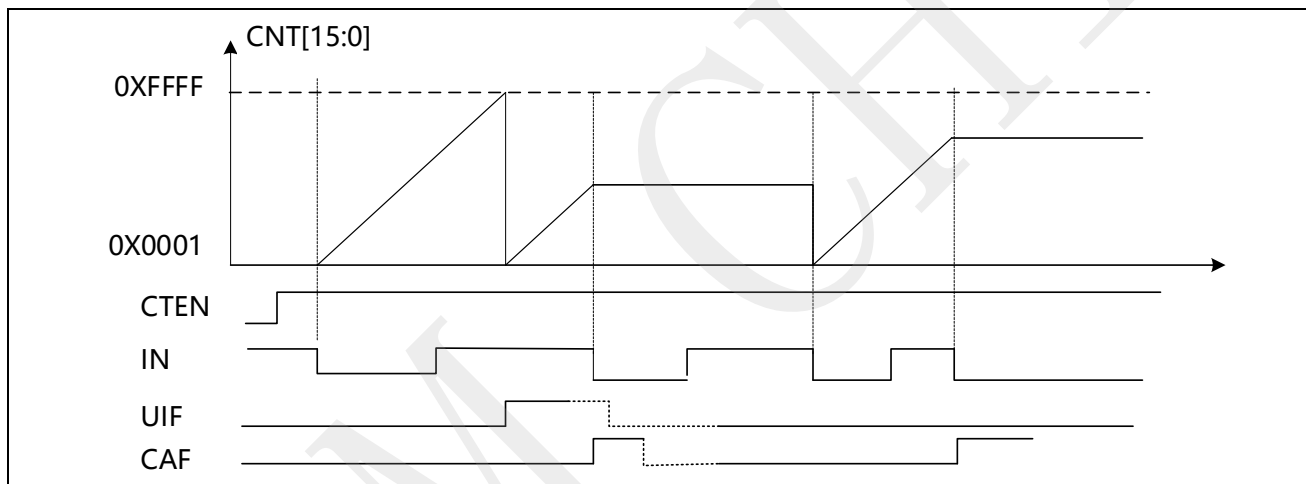


图 28-10 下降沿到下降沿周期测量

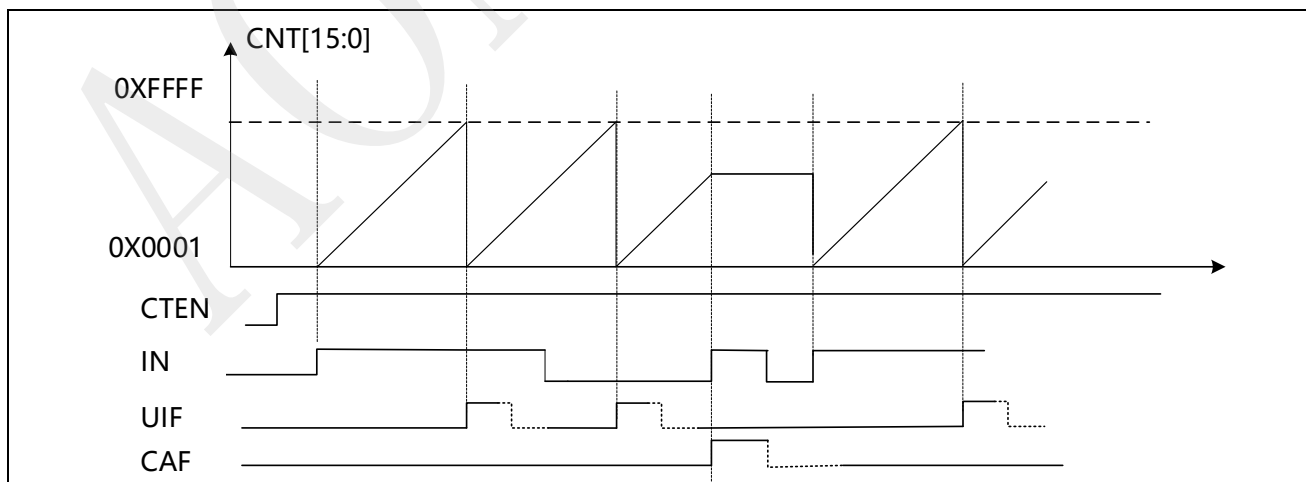


图 28-11 上升沿到上升沿周期测量

通过寄存 MSCR.TS 选择测量信号源。

000 ETFP: ETR 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

001 ITR0: Timer 内部互联信号 0, 其他 timer 的 TRGO 输出,参考 timer 互联部分

010 ITR1: Timer 内部互联信号 1, 其他 timer 的 TRGO 输出

011 ITR2: Timer 内部互联信号 2, 其他 timer 的 TRGO 输出

100 ITR3: Timer 内部互联信号 3, 其他 timer 的 TRGO 输出

101 IA0ED: 无效

110 IAFP: CH0A 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

111 IBFP: CH0B 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

注: 关于 TRGO 输出参考寄存器描述。

28.2.5.3 PWC 单次触发模式

设置 M1CR.ONESHOT=1 可以设置 PWC 单次测量, 测量完成后 CTEN 将被清除。

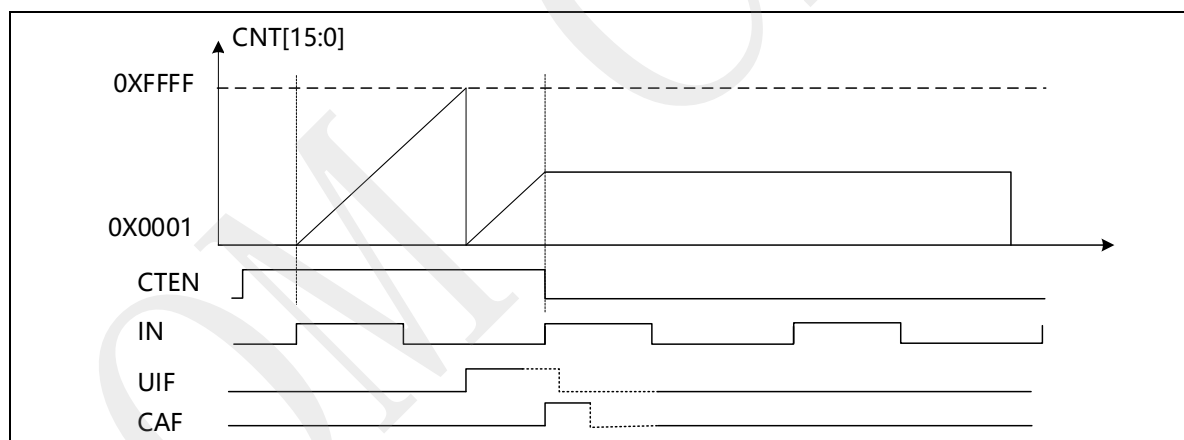


图 28-12 上升沿到上升沿周期测量单次模式

28.2.5.4 设置示例

脉冲低电平测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置 M1CR.edg2dn=0,M1CR.edg1st=1 选择测量低电平
4. 清除中断标志
5. 使能溢出中断 M1CR.UIE

6. 使能测量结束中断 CR0.CIEA
7. 使能定时器 M1CR.CTEN
8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
9. 等待下次测量

脉冲高电平单次测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置脉冲单次测量模式 M1CR.ONESHOT=1
4. 设置 M1CR.edg2dn=1, M1CR.edg1st=0 选择测量低电平
5. 清除中断标志
6. 使能溢出中断 M1CR.UIE
7. 使能测量结束中断 CR0.CIEA
8. 使能定时器 M1CR.CTEN
9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
10. 测量结束

28.2.6 模式 2/3 比较捕获模式

28.2.6.1 计数器

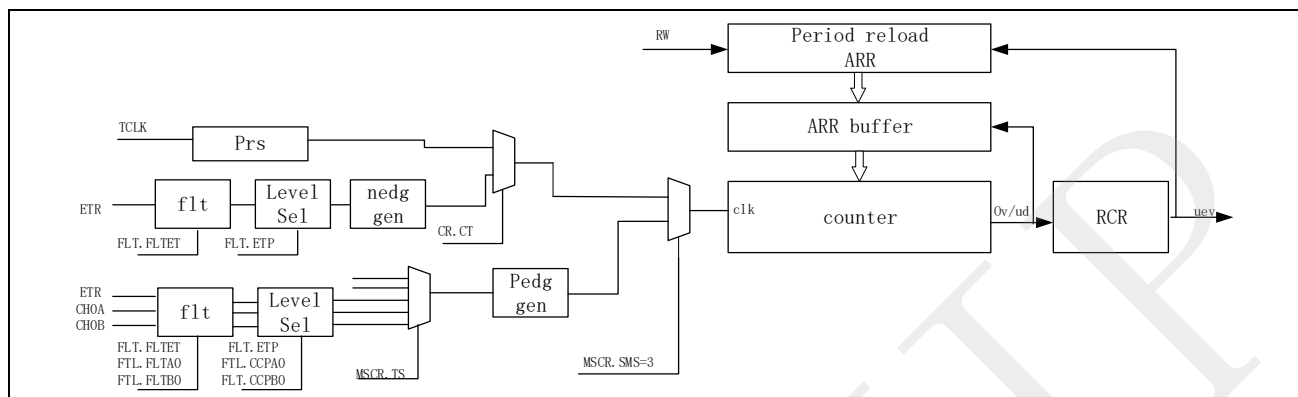


图 28-13 计数器框图

计数器主要部分是一个 16 位计数器与相关的自动装载寄存器。这个计数器可以向上计数（模式 2），向下计数（模式 2）或向上向下双向计数（模式 3）。计数器的时钟可以由预分频器 PRS 分频得到，也可以选择 ETR 输入外部时钟或者通过 MSCR.TS 选择的外部输入信号和内部互联信号。

计数器基本单元包括：

- 计数器寄存器 CNT
- 预分频寄存器 M23CR.PRS
- 自动装载寄存器 ARR
- 重复次数寄存器 RCR
- 时钟选择控制寄存器 FLT,CR0,MSCR, M23CR

自动装载寄存器具有缓存功能，计数器产生事件更新后重载值从缓存寄存器更新到计数器。当计数器处于停止状态或者缓存功能关闭状态，自动装载寄存器立刻更新到缓存寄存器。当定时器处于运行状态并行缓存功能有效时，写入到自动装载寄存器的值不会立刻更新到缓存寄存器，当事件更新后才有自动装载寄存器更新到缓存寄存器。时钟选择及门控功能，触发功能，复位功能参考模式 2/3 从模式章节。

28.2.6.2 计数器波形

模式 2 为锯齿波计数波形，通过设置 M23CR.DIR 可以更改计数方向。

设置 M23CR.DIR 为 0 时，计数器为递增计数模式，这种模式下，计数器从 0 计数到自动重载值（ATIM_ARR），然后重新从 0 开始计数并生成计数器上溢事件。如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次（ATIM_RCR+1）后，将生成更新事件（UEV）。否则，将在每次计数器上溢时产生更新事件。

将 M23CR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

发生更新事件时，将更新所有寄存器且将更新标志（ATIM_IFR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 自动重载缓存值将以 ARR 寄存器值进行更新
- 比较缓存值将以比较寄存器 CCRxy 进行更新

以下图示显示 ARR=0x2C 时不同计数方向的计数器波形

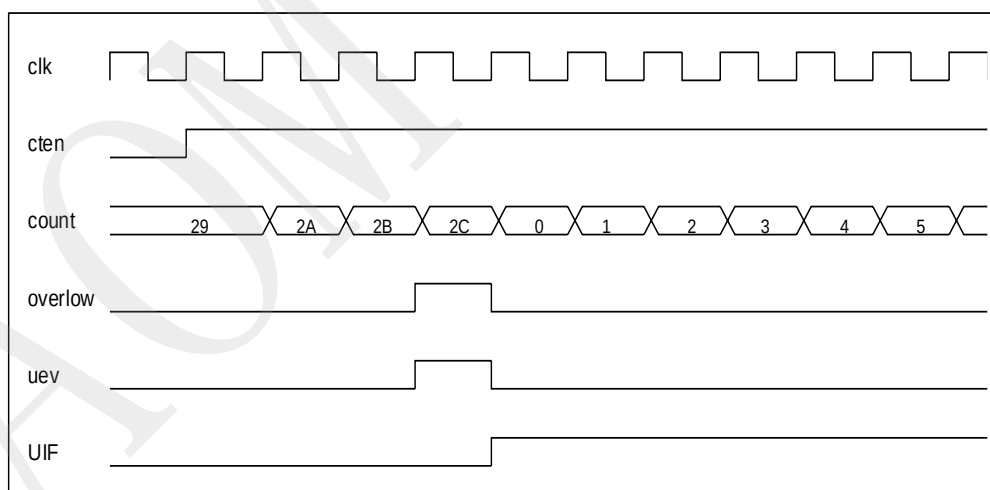


图 28-14 无预分频的向上计数

计数溢出周期时钟个数为 ARR+1

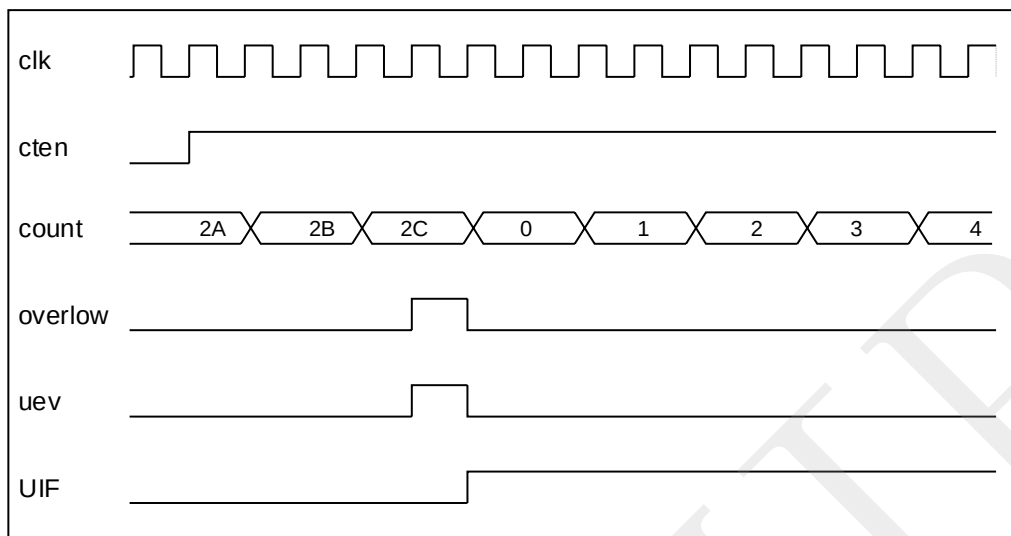


图 28-15 带预分频的上计数

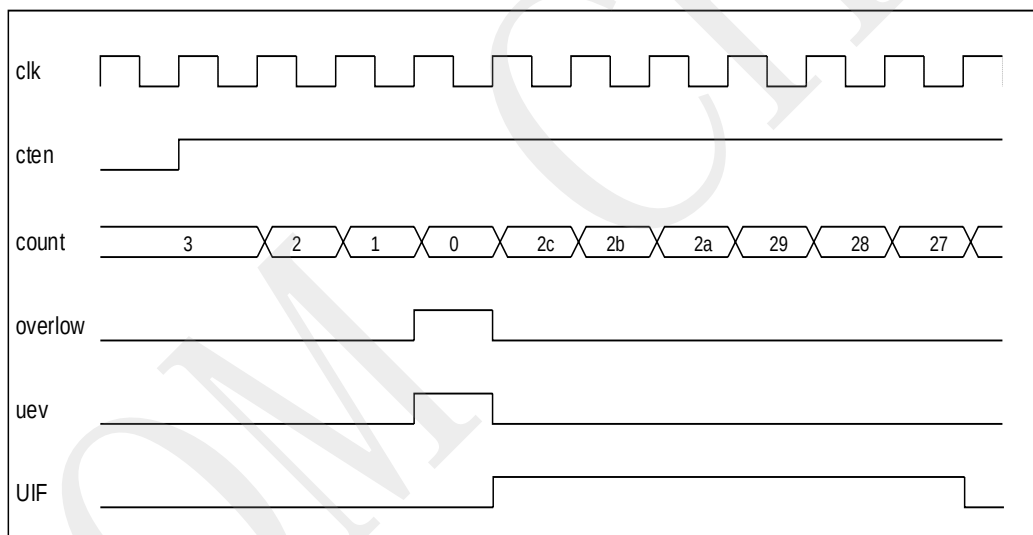


图 28-16 不带预分频的下计数

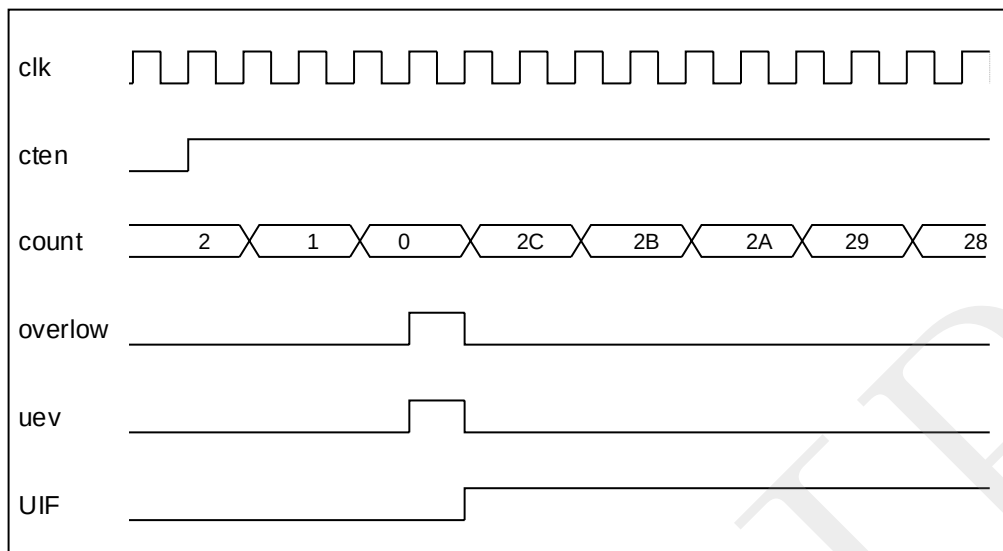


图 28-17 带预分频的下计数

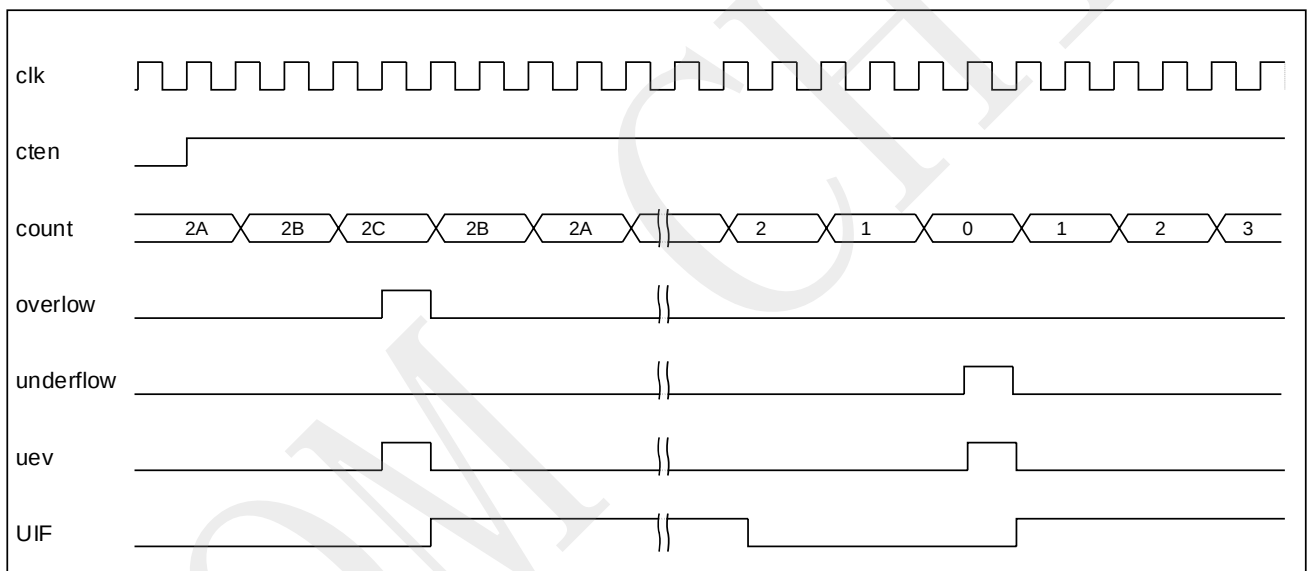


图 28-18 带预分频的上下计数

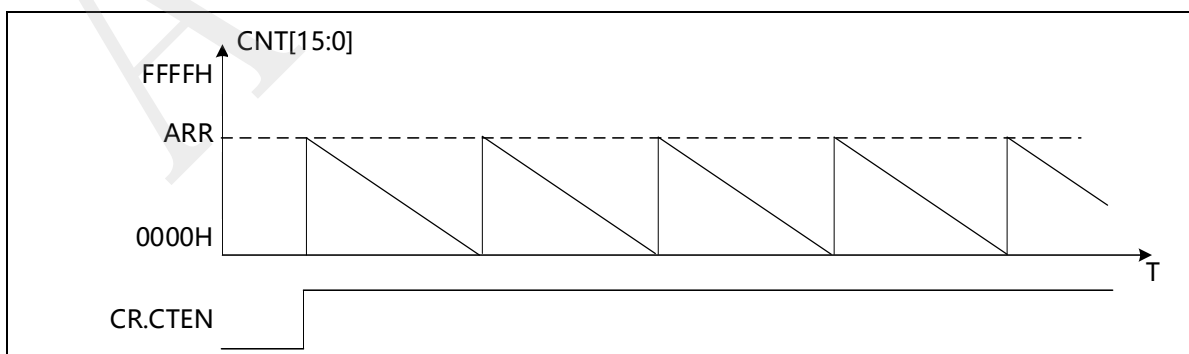


图 28-19 边沿对齐计数器波形(DIR =1)

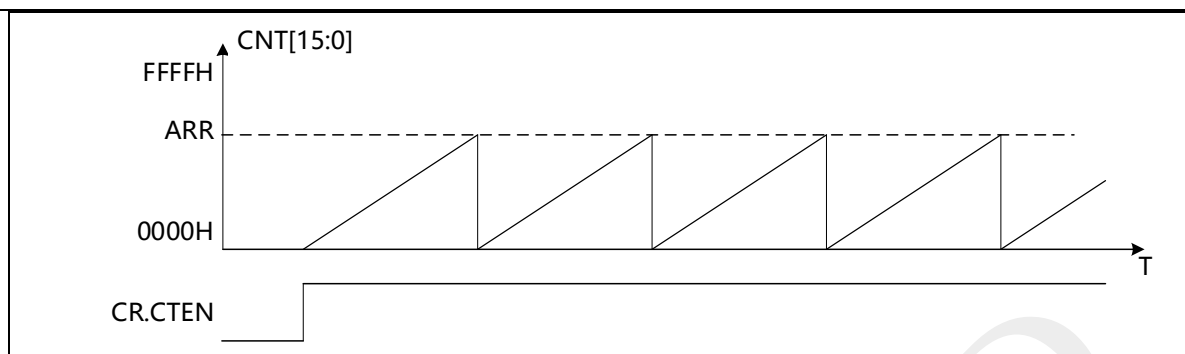


图 28-20 边沿对齐计时器波形(DIR =0)

模式 3 为三角波计数波形，计数方向控制位只读，不可以更改计数方向。

中心对齐（三角波）模式下 M23CR.DIR 方向位是只读的，写值无效。从其他模式切换到中心对齐模式 DIR 自动清 0。软件事件更新和从模式外部触发复位模式 DIR 自动清零。

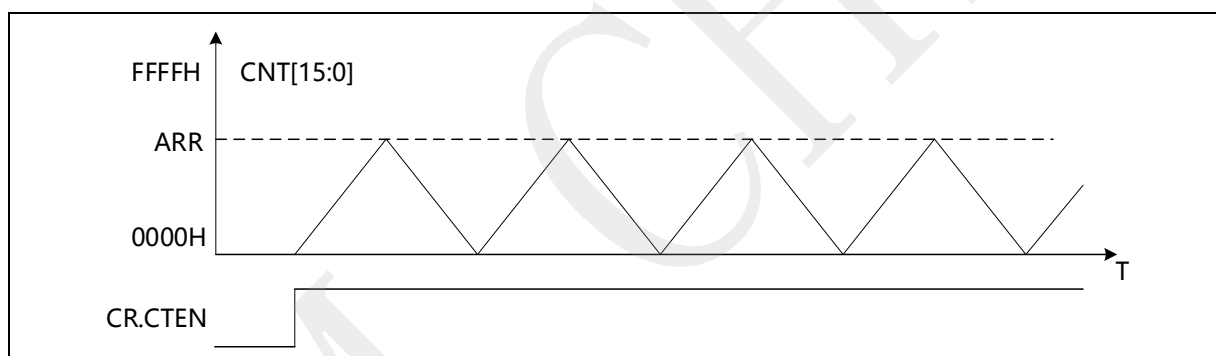


图 28-21 中心对齐计数器波形

28.2.6.3 重复计数

重复计数器使用计数器的溢出进行向下计数。计数到 0 时，即计数器发生重复寄存器设置的值加一次溢出时。当缓存寄存器使能时，周期重载寄存器更新到周期缓存寄存器。比较模式下比较寄存的值更新到比较缓存寄存中。

重复计数器在下面条件成立时递减：

- 上计数模式下每次计数器溢出时
- 下计数模式下每次计数器下溢时
- 三角波模式下每次上溢出和每次下溢出时

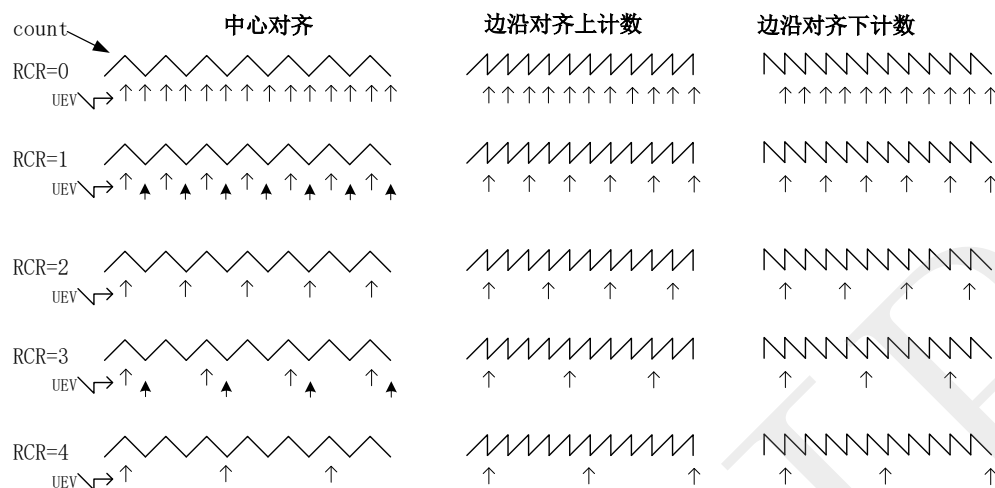


图 28-22 重复计数器产生更新时序

除了上下溢出通过重复计数器可以产生事件更新 UEV 外，还可以通过写寄存器 M23CR.UG 产生软件及从模式复位事件更新 UEV；这时需要配置 M23CR.URS。

28.2.6.4 数据缓存

自动重载数据 ARR 与比较寄存器都可以配置缓存功能，当缓存功能有效时，当发生 UEV 事件更新时，写入的周期值 ARR 与比较值 CCR 才会生效。

自动重载值在不同计数模式下的更新时序图如下

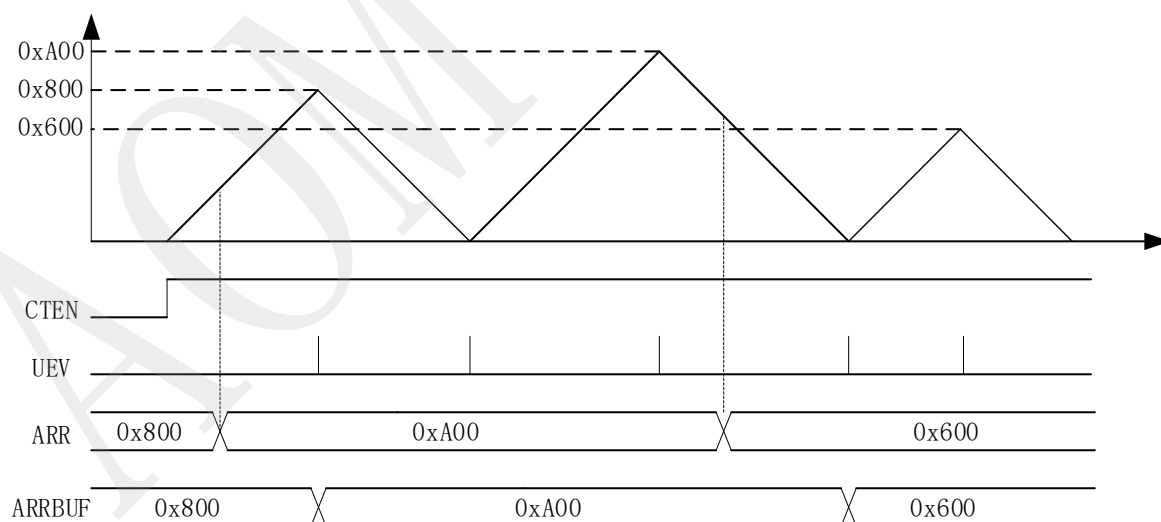


图 28-23 三角波模式下缓存使能

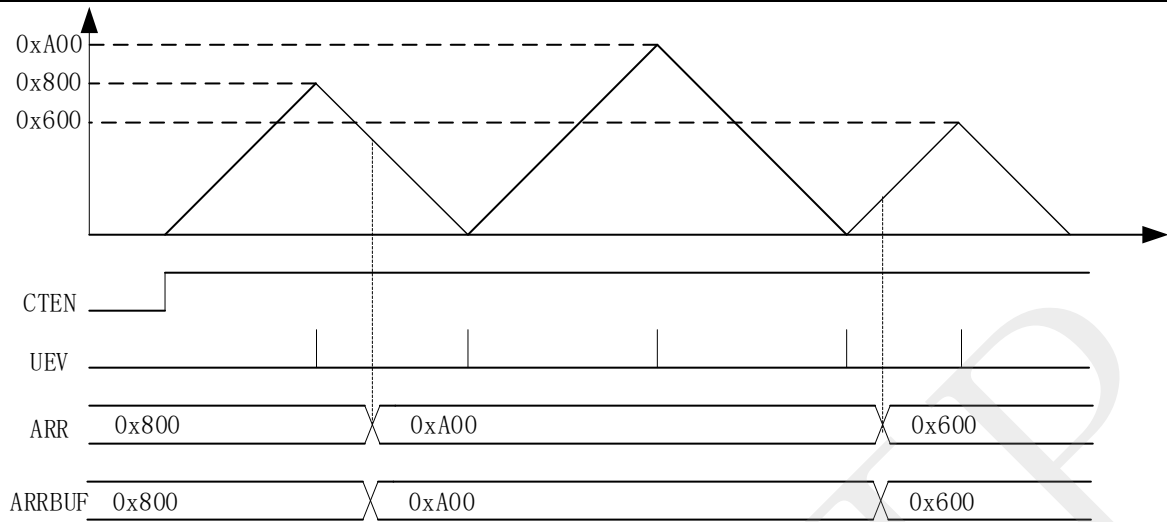


图 28-24 三角波模式下缓存无效

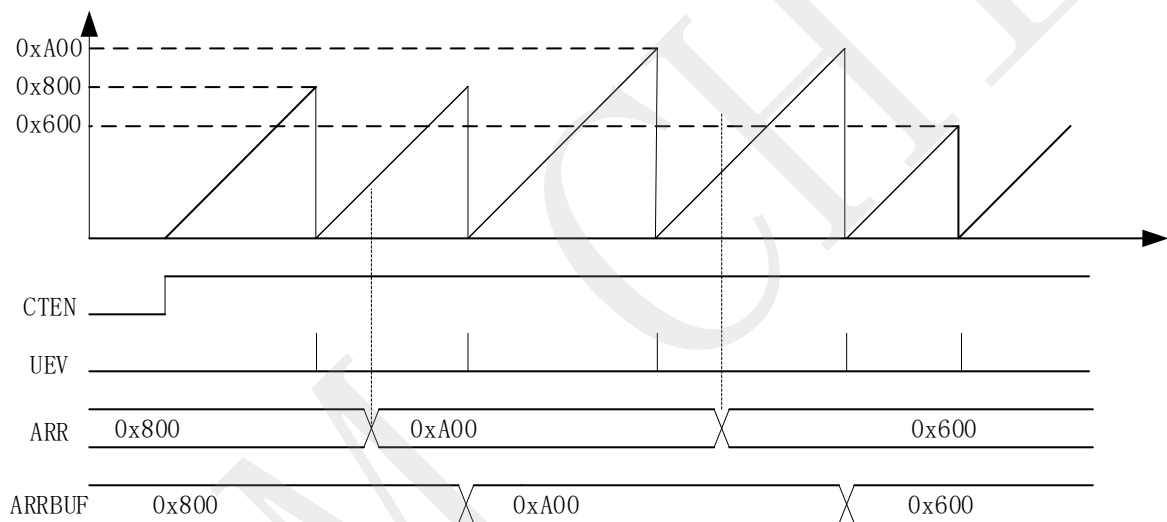


图 28-25 锯齿波模式下上计数缓存使能

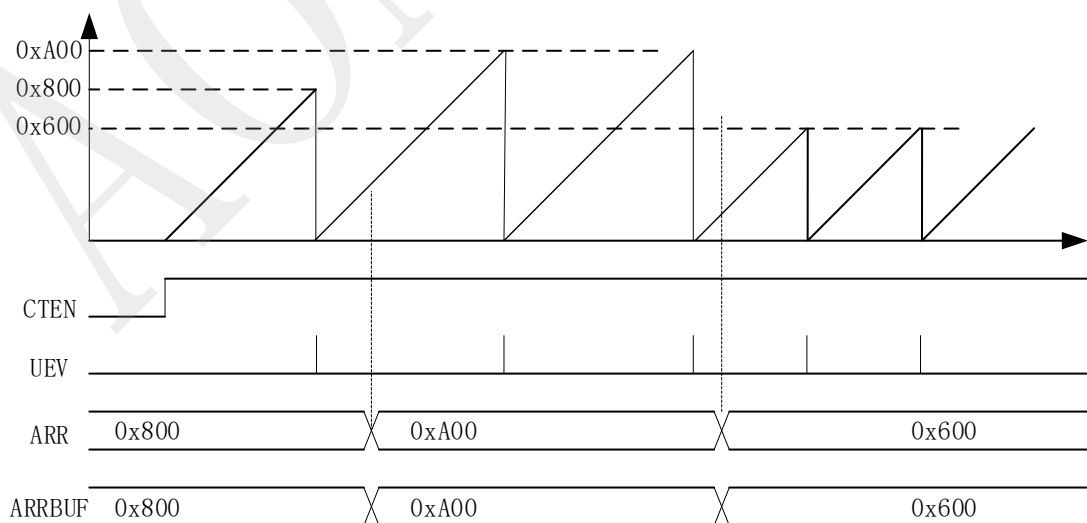


图 28-26 锯齿波模式下上计数缓存无效

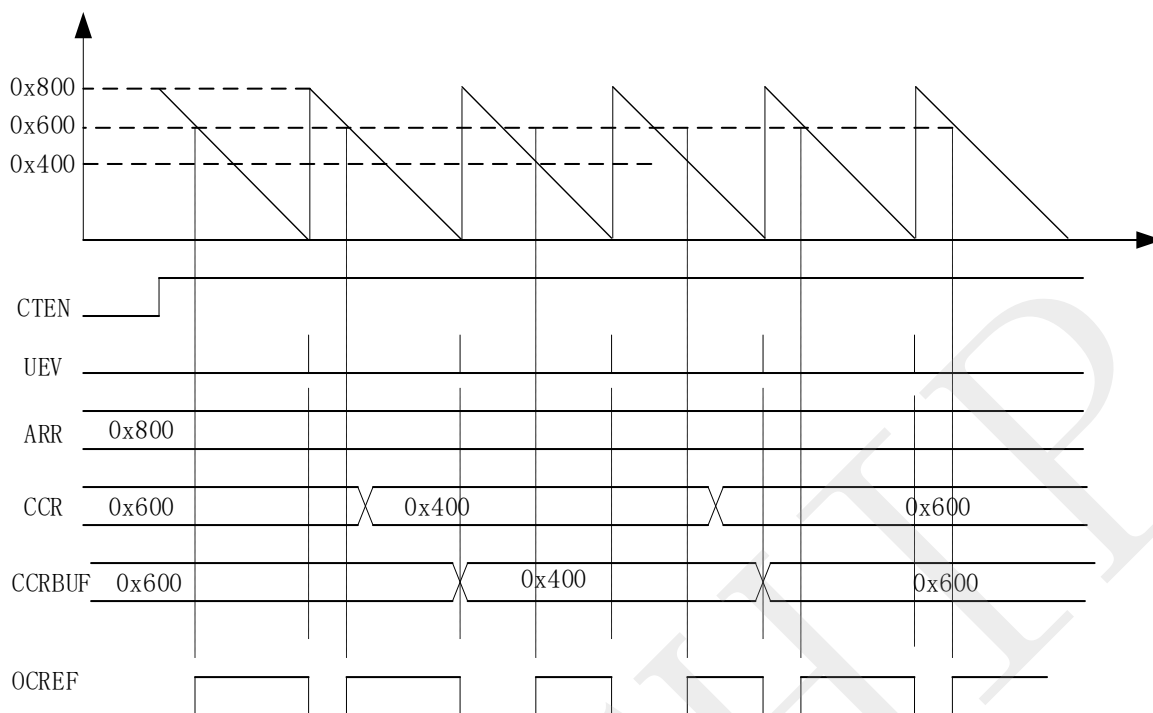


图 28-29 锯齿波模式下计数比较缓存使能

28.2.6.5 比较输出 OCREF

比较输出 OCREFA 可以配置为单点比较，使用比较寄存器 CCRA 控制 OCREFA 的输出；OCREFA 的比较输出也可以配置为双点比较，使用比较寄存器 CCRA,CCRB 一起控制 OCREFA 的比较输出。

OCREFB 的比较输出只能使用单点比较，使用比较寄存器 CCRB 控制 OCREFB 的比较输出。

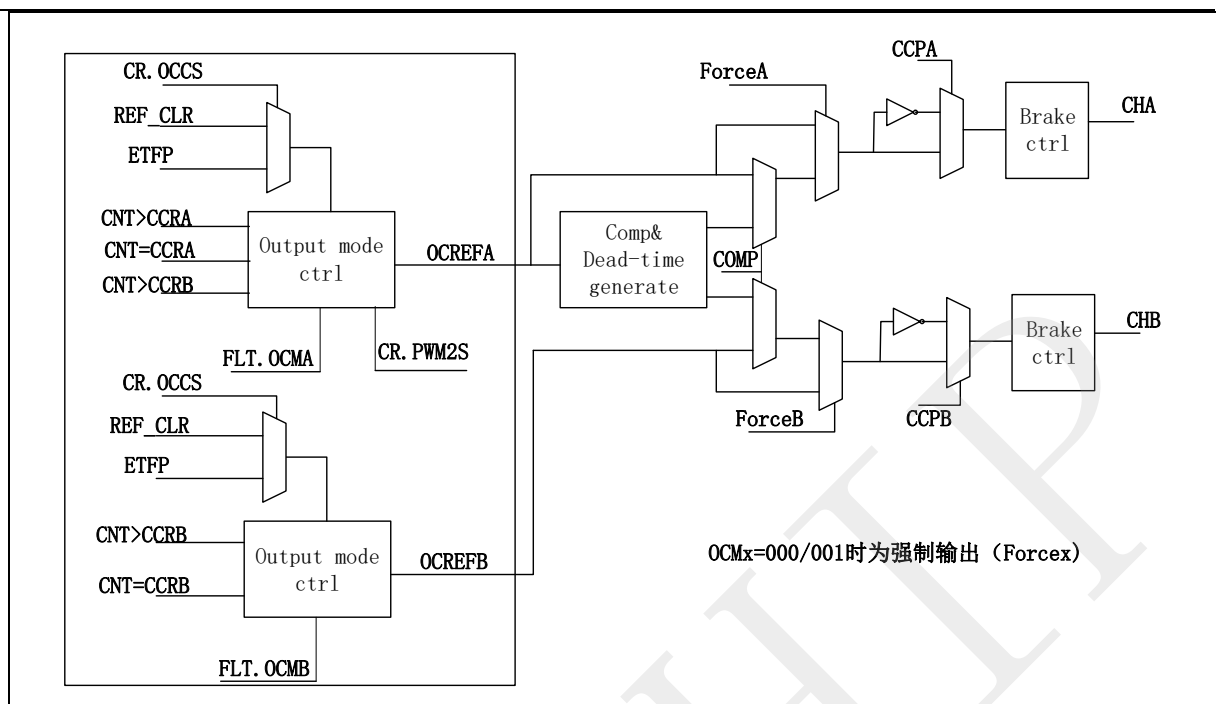


图 28-30 OCREF 输出框图

OCREF 输出使用 OCMx 选择

000: 强制为 0

001: 强制为 1

010: 比较匹配时强制为 0

011: 比较匹配时强制为 1

100: 比较匹配时翻转

101: 比较匹配时输出一个计数周期的高电平

110: PWM 模式 1

单点比较:

上计数时 $CNT < CCR_{xy}$ 输出高, 下计数时 $CNT > CCR_{xy}$ 输出为低电平

双点比较:

1) 锯齿波上计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为低电平

2) 锯齿波下计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为高电平

3) 三角波上计数 $CNT < CCR_{xA}$ 输出高, 下计数 $CNT > CCR_{xB}$ 为低电平

111: PWM 模式 2

单点比较:

上计数时 $CNT < CCR_{xy}$ 输出低，下计数时 $CNT > CCR_{xy}$ 输出为高电平
双点比较：

- 1) 锯齿波上计数 $CCR_{xA} \leq CNT < CCR_{xB}$ 输出为高电平
- 2) 锯齿波下计数 $CCR_{xA} \leq CNT < CCR_{xB}$ 输出为低电平
- 3) 三角波上计数 $CNT < CCR_{xA}$ 输出低，下计数 $CNT > CCR_{xB}$ 为高电平

注：强制输出有高优先级，当强制输出有效时，互补输出控制无效。

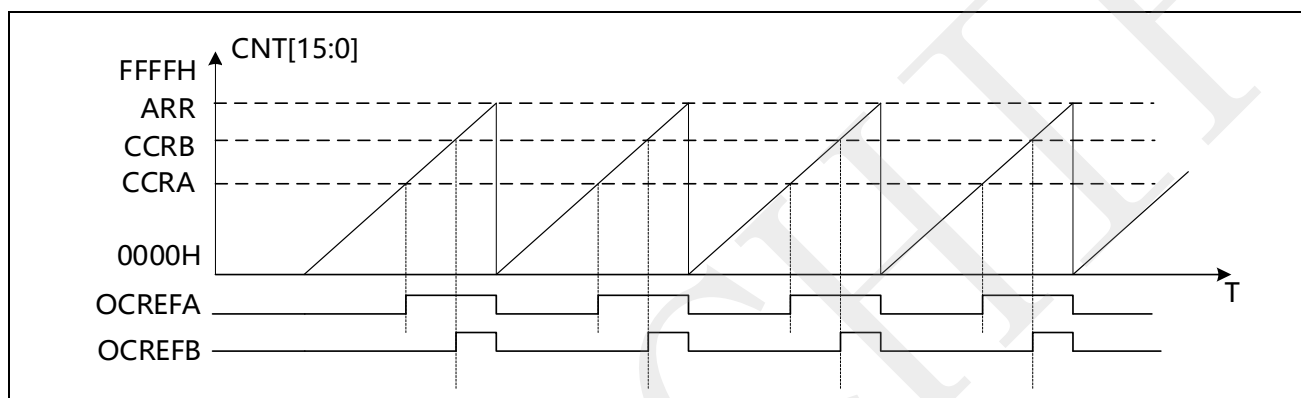


图 28-31 锯齿波计数单点比较 OCREF 输出波形 (OCM_x=111)

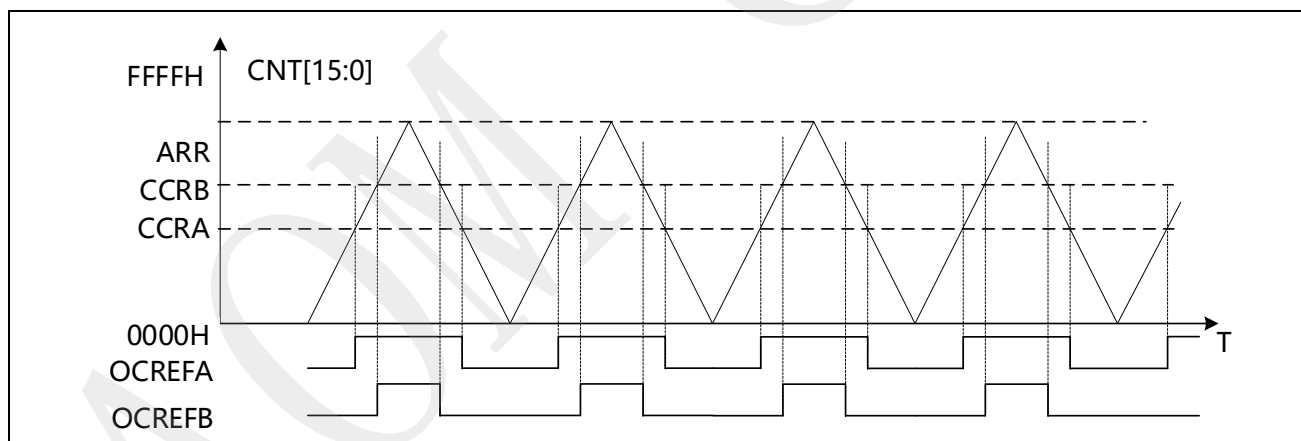


图 28-32 三角波计数单点比较 OCREF 输出波形 (OCM_x=111)

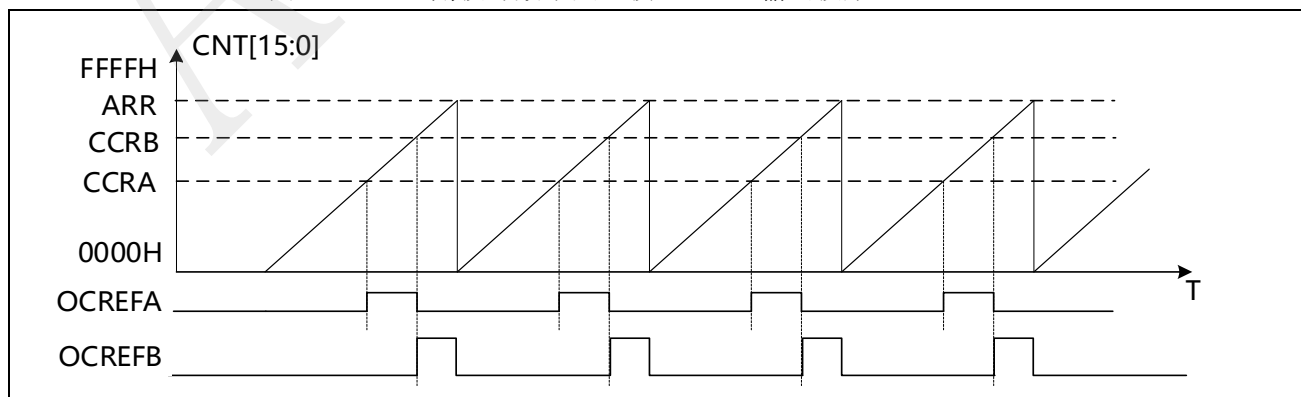


图 28-33 锯齿波计数双点比较 OCREF 输出 (OCMx=111)

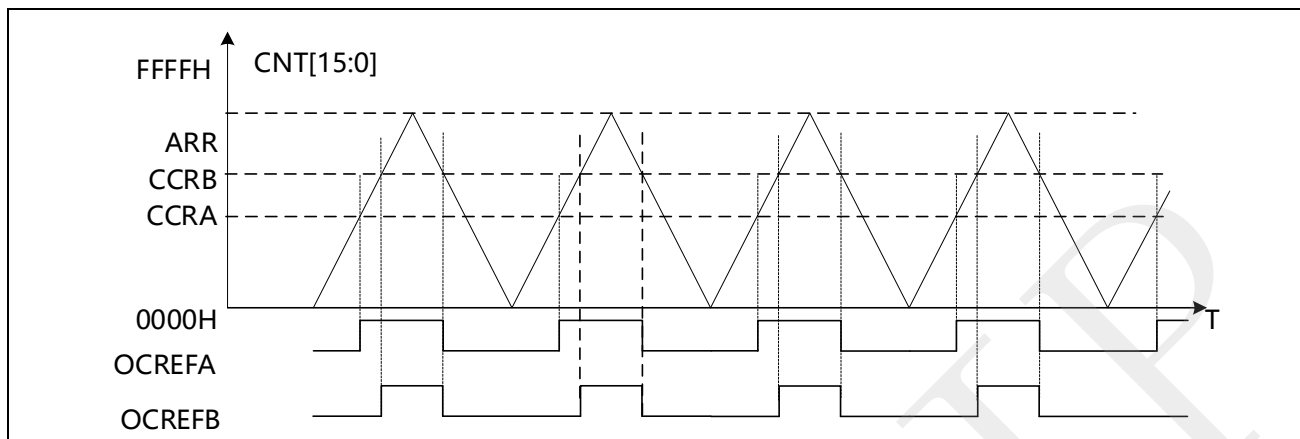


图 28-34 三角波计数双点比较 OCREF 输出 (OCMx=111)

28.2.6.6 独立 PWM 输出

由 OCREFA 控制 CHA 的输出, OCREFB 控制 CHB 的输出。通过 CRCHx.CCPA, CRCHx.CCPB 可以控制 CHA, CHB 输出的反向。

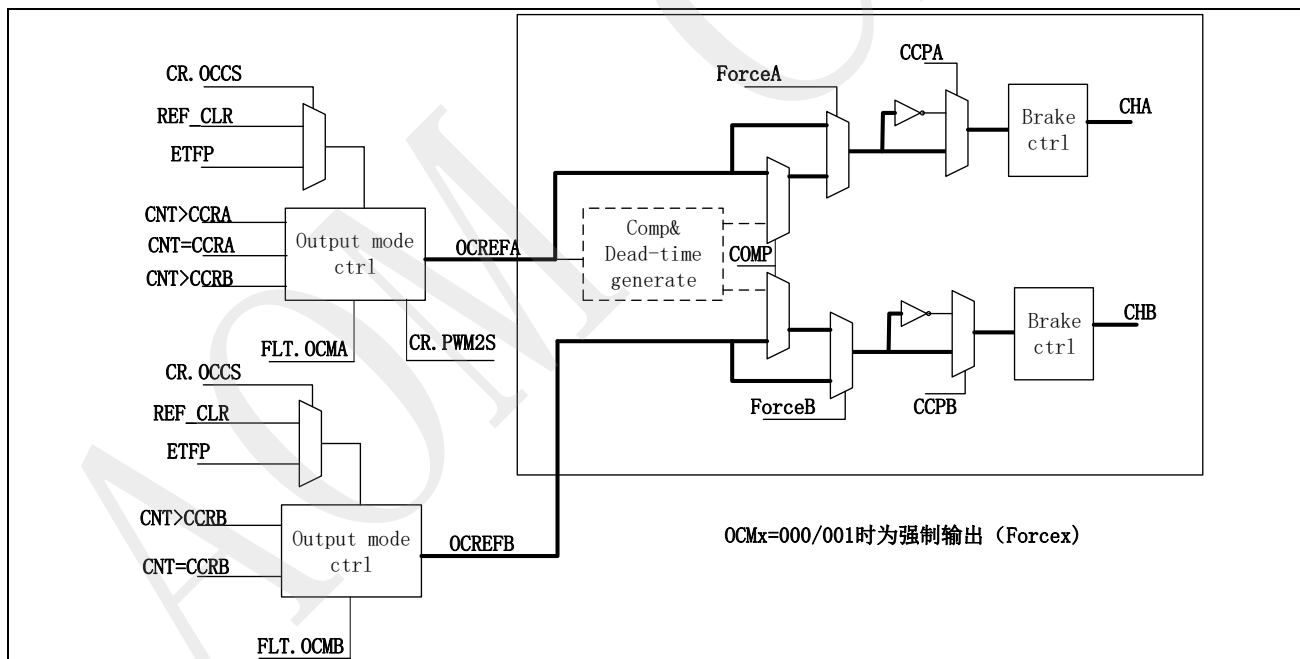


图 28-35 独立 PWM 输出框图

PWM 输出与 OCREF 关系

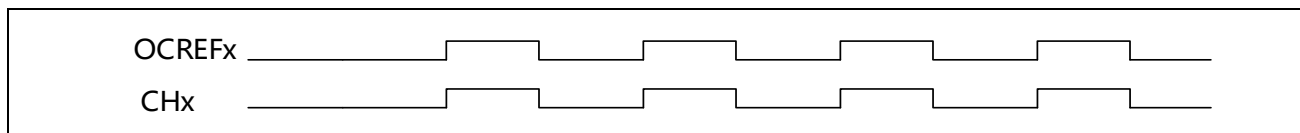


图 28-36 CCPx=0时PWM输出波形

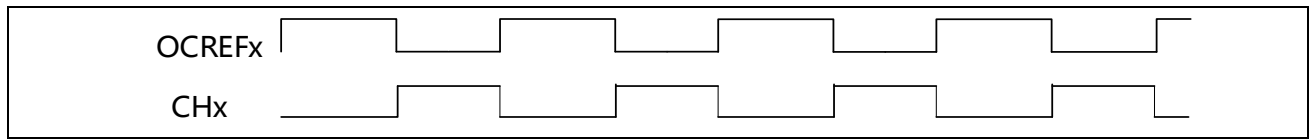


图 28-37 CCPx=1 时 PWM 输出波形

28.2.6.7 互补 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFA 同时控制 CHB 的输出。比较寄存器 CCRxB 可以作为专用比较控制 ADC 触发。

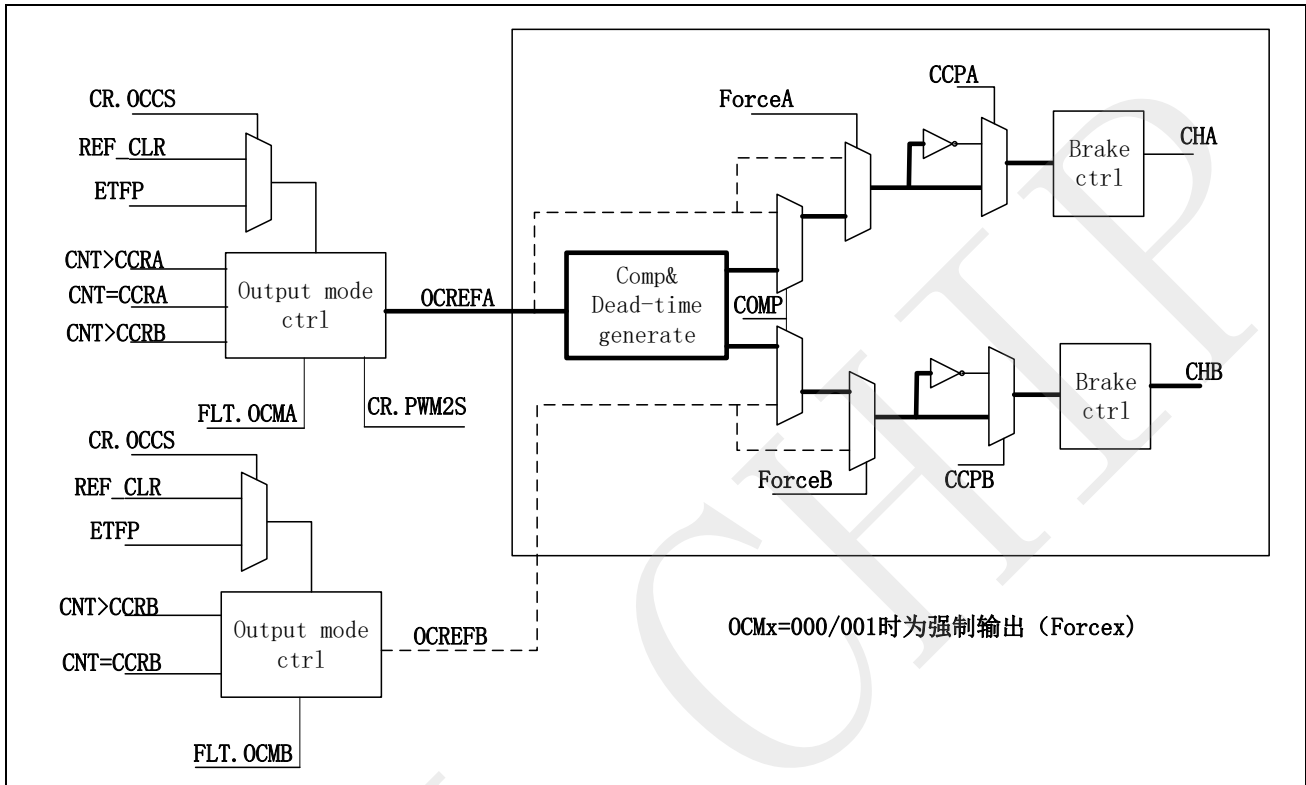


图 28-38 互补 PWM 输出框图

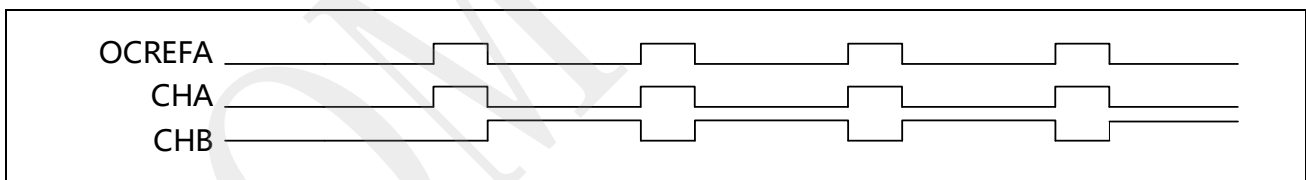


图 28-39 互补PWM输出波形图

28.2.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

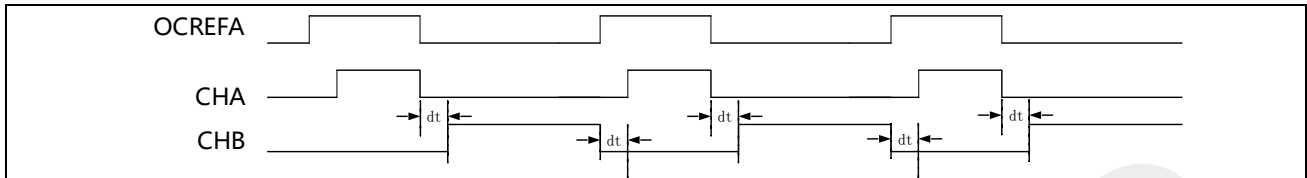
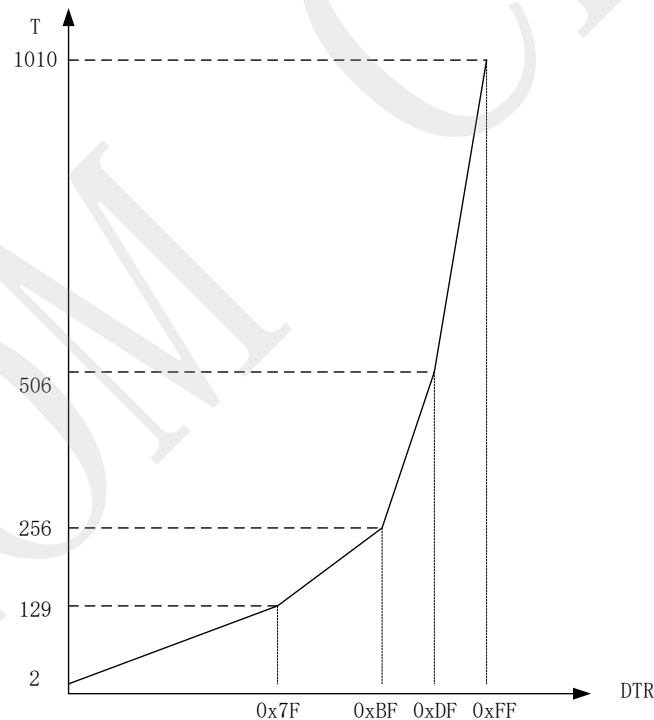


图 28-40 互补PWM输出波形图

死区时间使用 8 位 DTR 控制，死区时间 dt 与 DTR 的关系如下

DTR[7] =0	$T=DTR[6:0]+2$	2-129	step=1
DTR[7:6] =10	$T=\{DTR[5:0]+64\}*2+2$	130-256	step=2
DTR[7:5] =110	$T=\{DTR[4:0]+32\}*8+2$	258-506	step=8
DTR[7:5] =111	$T=\{DTR[4:0]+32\}*16+2$	514-1010	step=16



100M clock 死区时间由2.56us调整为10.1us

图 28-41 死区时间

28.2.6.9 单脉冲输出

单脉冲模式 (ONE SHOT) 是上述 PWM 模式的一个特例。在这种模式下, 计数器可以在一个激励信号的触发下启动, 并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 ATIM_M23CR 寄存器中的 ONESHOT 位置 1, 即可选择单脉冲模式。这样, 发生下一更新事件 UEV 时, 计数器将自动停止。

单脉冲模式在锯齿波下计数模式计数器初值不要设置为 0, 上计数模式计数值不要设置到大于等于 ARR。

只有当比较值与计数器初始值不同时, 才能正确产生输出脉冲。启动前 (定时器等待触发时), 必须进行如下配置:

- 递增计数时: $CNT < CCR_{xy} \leq ARR$ (特别注意, $0 < CCR_{xy}$),
- 递减计数时: $CNT > CCR_{xy}$ 。

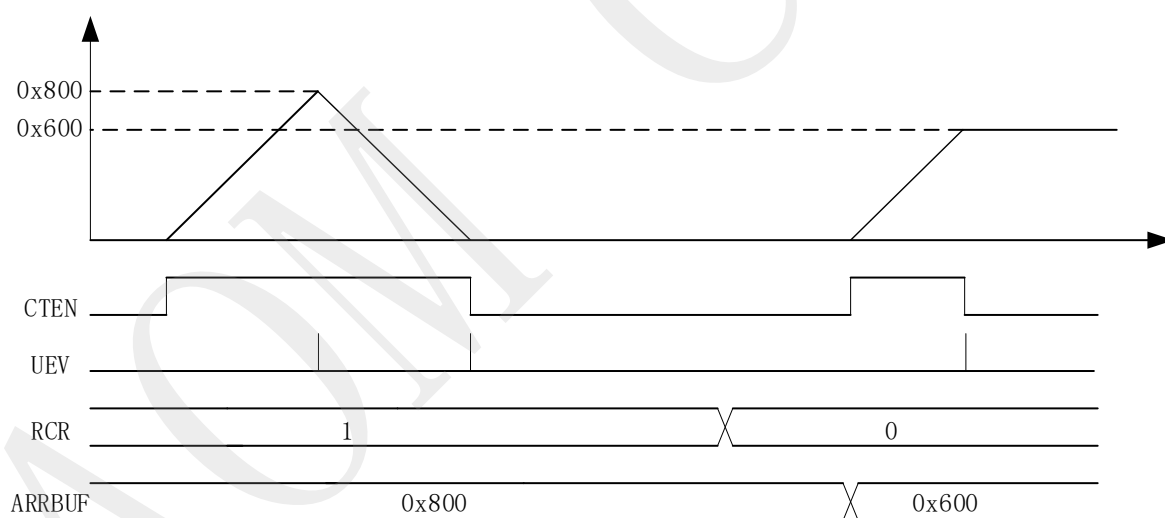


图 28-42 三角波模式单脉冲计数

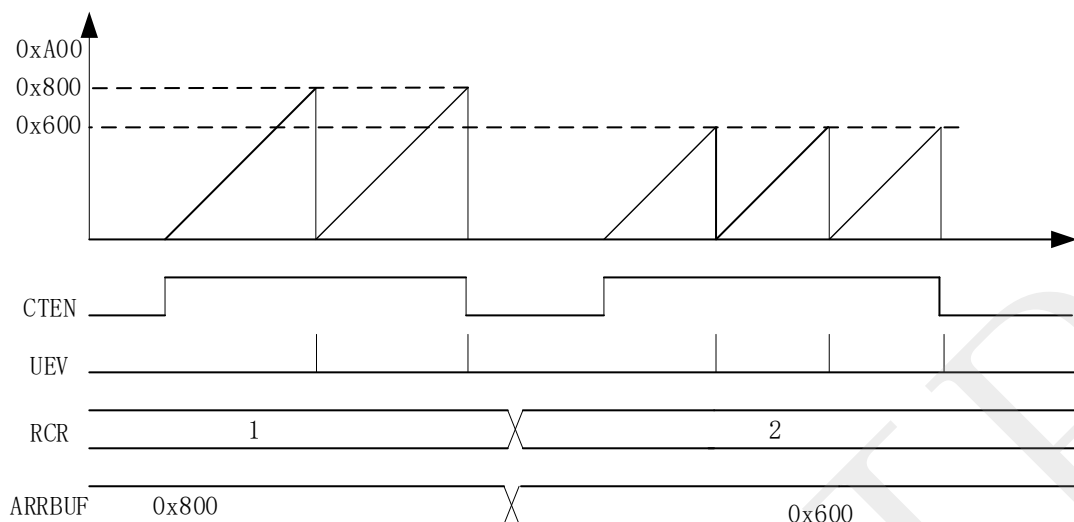


图 28-43 锯齿波上计数单脉冲模式

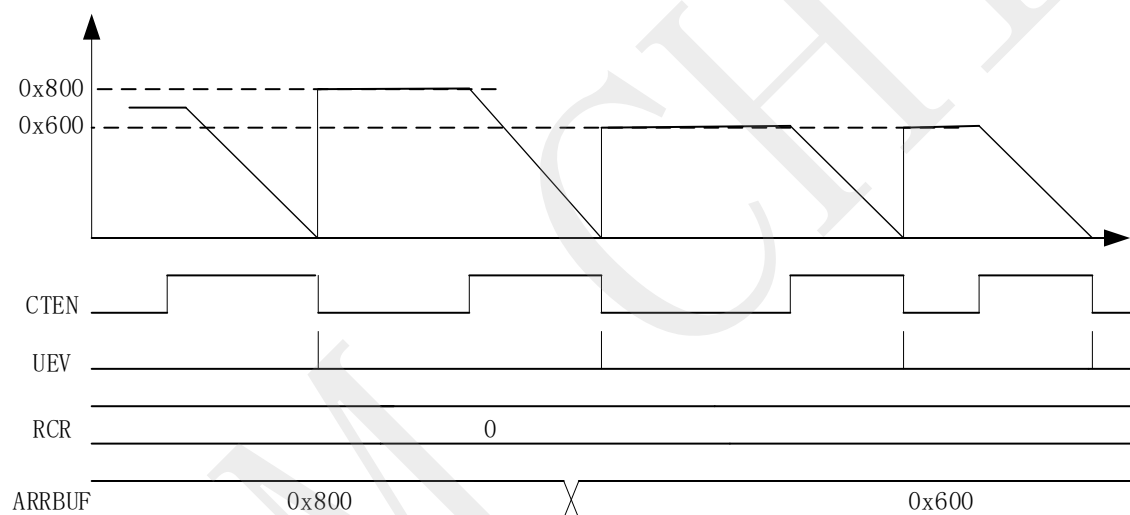


图 28-44 锯齿波下计数单脉冲模式

28.2.6.10 比较中断

锯齿波比较匹配会将 IFR 寄存器相应的标志置 1，如果中断使能 CRCHx.CIEy (x=0,1,2;y=A,B)将会触发中断。

三角波比较匹配可单独选择上升计数比较匹配，下降计数比较匹配或者两者比较都匹配。

比较 A 比较匹配当计数值与 CCRxA 相等时，统一使用 M23CR.CIS 控制。

M23CR.CIS=2'b00 时 比较匹配无输出

M23CR.CIS=2'b01 时 上计数时比较匹配

M23CR.CIS=2'b10 时 下计数时比较匹配

M23CR.CIS=2'b11 时 上下计数时比较都匹配

比较 B 比较匹配当计数值与 CCRxB 相等时，不同通道可以使用 CRCHx.CISB 单独控制。

CRCHx.CISB=2'b00 时 通道 x 比较匹配无输出

CRCHx.CISB=2'b01 时 通道 x 上计数时比较匹配

CRCHx.CISB=2'b10 时 通道 x 下计数时比较匹配

CRCHx.CISB=2'b11 时 通道 x 上下计数时比较都匹配

B 通道比较匹配单独控制为了更灵活的触发 ADC。详见 Timer 触发 ADC 章节。

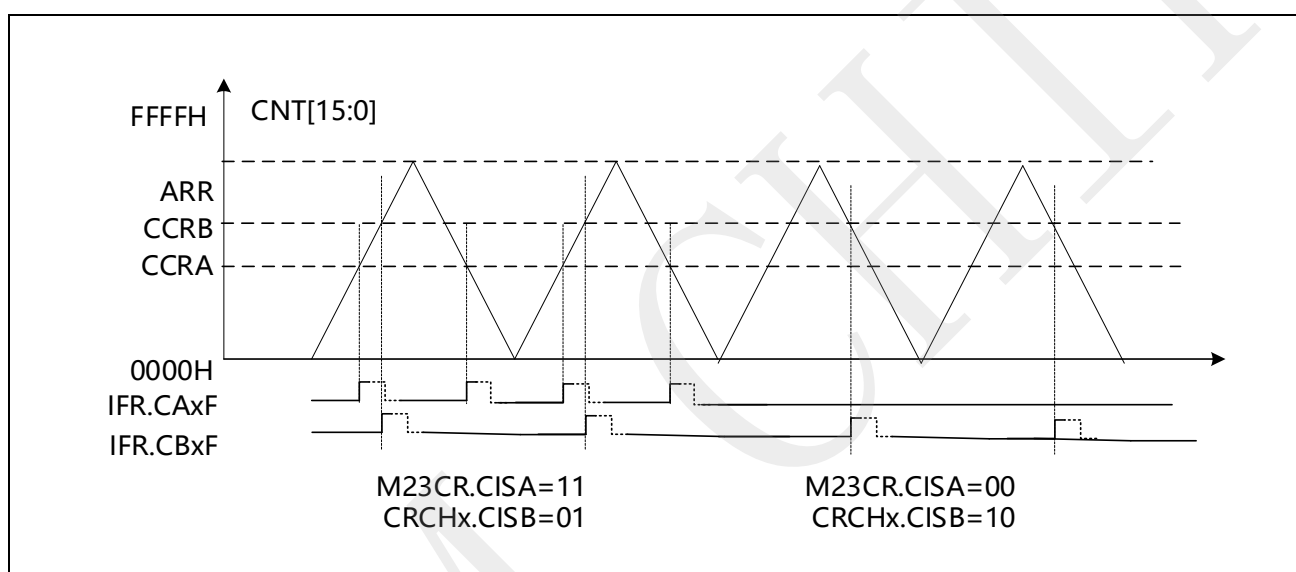


图 28-45 中断示意图

28.2.6.11 捕获输入

M23CR.MODE=2/3

在三角波计数或者锯齿波计数模式下都可以设置捕获功能，可以设置捕获的电平边沿，当发生捕获时捕获的值存入比较捕获寄存器并产出捕获中断。

每个通道的比较捕获功能可以单独设置，通过寄存器 CRCHx.CSA/CSB 选择。

每个通道的捕获边沿可以单独设置，通过寄存器 CRCHx.CFy/CRY(x=0/1/2;y=A/B)选择捕获触发的边沿。

当捕获发生后捕获标志未清除前再次发生捕获动作，会产生捕获数据覆盖标志。

定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

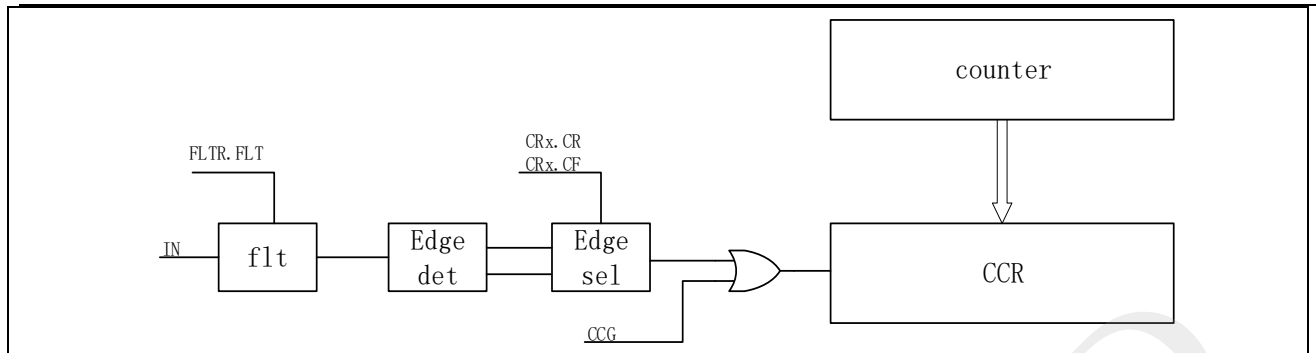


图 28-46 捕获功能框图

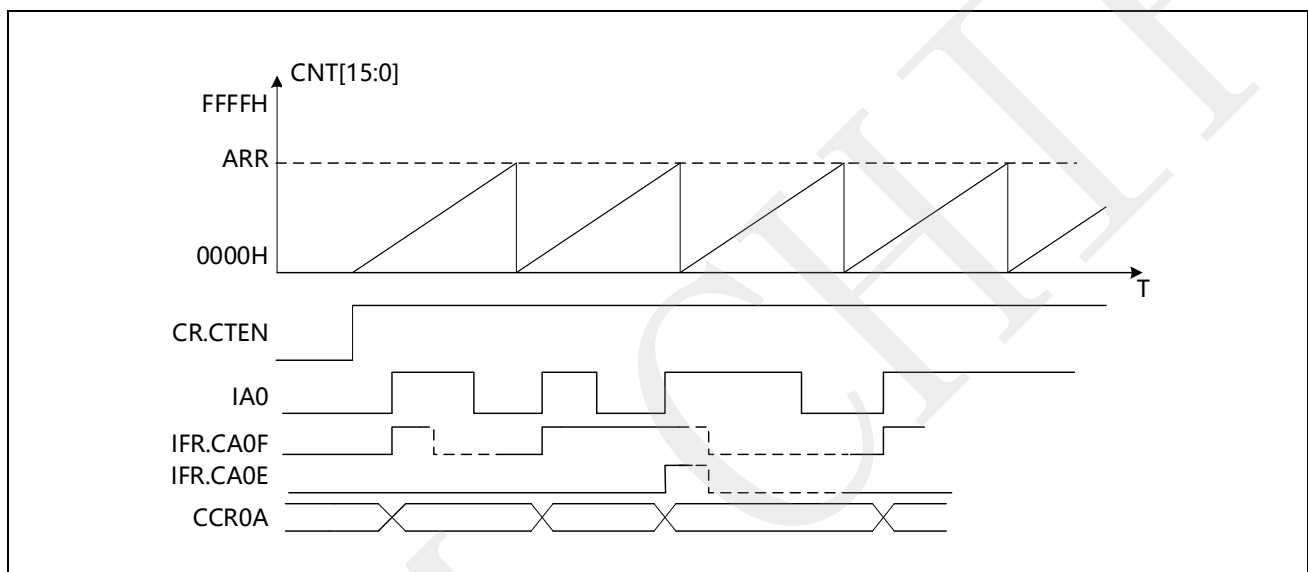


图 28-47 捕获时序图

CH0A 的捕获输入通过 MSCR.IA0S 选择 CH0A 输入还是 CH0A, CH1A, CH2A 的异或输入。

IA0S	0	1
ATIM	CHA0	CHA0 CHA1 CHA2异或输入

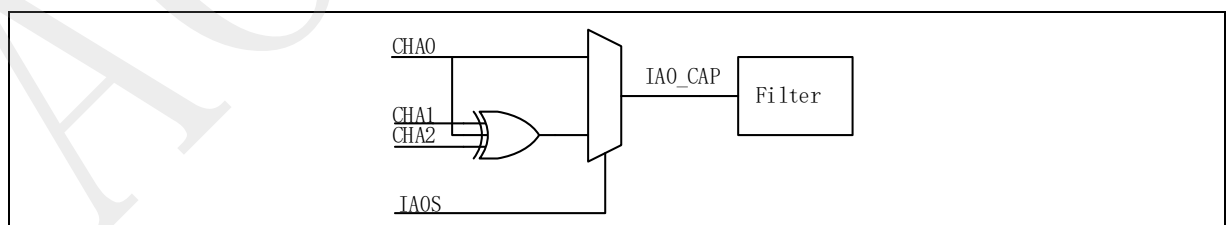


图 28-48 CHA 端口选择

CH0B 的捕获输入通过 MSCR. IB0S 选择 CH0B 输入还是内容 MSCR.TS 选择的信号。

IB0S	0	1																		
Timer0/1/2/3	CHB0	内部触发MSCR.TS选择信号																		
		<table><tr><th>MSCR.TS</th><th>通道B捕获输入</th></tr><tr><td>000</td><td>ETR 滤波相位选择输出信号，滤波相位选择可配</td></tr><tr><td>001</td><td>内部互联ITR0</td></tr><tr><td>010</td><td>内部互联ITR1</td></tr><tr><td>011</td><td>内部互联ITR2</td></tr><tr><td>100</td><td>内部互联ITR3</td></tr><tr><td>101</td><td>CH0A 边沿</td></tr><tr><td>110</td><td>CH0A 滤波输出信号，滤波功能可配</td></tr><tr><td>111</td><td>CH0B 滤波输出信号，滤波功能可配</td></tr></table>	MSCR.TS	通道B捕获输入	000	ETR 滤波相位选择输出信号，滤波相位选择可配	001	内部互联ITR0	010	内部互联ITR1	011	内部互联ITR2	100	内部互联ITR3	101	CH0A 边沿	110	CH0A 滤波输出信号，滤波功能可配	111	CH0B 滤波输出信号，滤波功能可配
		MSCR.TS	通道B捕获输入																	
		000	ETR 滤波相位选择输出信号，滤波相位选择可配																	
		001	内部互联ITR0																	
		010	内部互联ITR1																	
		011	内部互联ITR2																	
		100	内部互联ITR3																	
		101	CH0A 边沿																	
		110	CH0A 滤波输出信号，滤波功能可配																	
111	CH0B 滤波输出信号，滤波功能可配																			

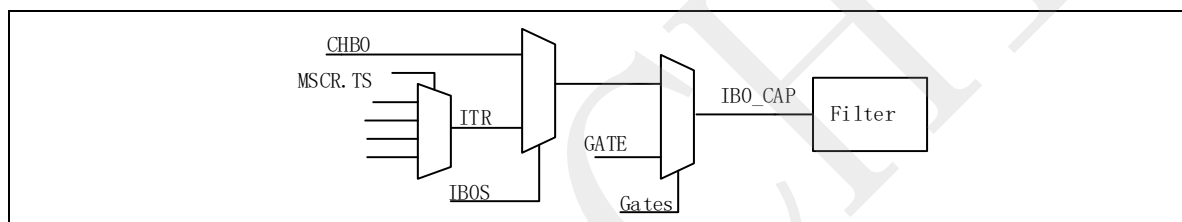


图 28-49 CHB 端口选择

Gates 信号为选择 PWM 互补输出功能生成的信号，当 PWM 互补输出时，比较捕获寄存器 CCR0B 没有使用，自动切换到使用 GATE 作为捕获/比较输出。

28.2.6.12 设置示例

边沿对齐独立 PWM 输出设置

1. 设置模式 M23CR.MODE=2
2. 设置锯齿波计数方向 M23CR.DIR
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA,CCRxB
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断

9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 M23CR.CTEN

中心对齐互补 PWM 输出设置

1. 设置模式 M23CR.MODE=3
2. 设置互补输出 M23CR.COMP
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA(CCRxB 不需要设置，PWM 输出与这个寄存器无关)
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 M23CR.CTEN

三角波非中心对齐带死区互补 PWM 输出设置

1. 设置模式 M23CR.MODE=3
2. 设置互补输出 M23CR.COMP
3. 设置两个比较使能 M23CR.PWM2S
4. 设置 PWM 周期值 ARR
5. 设置计数器初值（初值必须小于周期值）
6. 设置 PWM 比较值 CCRxA,CCRxB,上计数比较点为 CCRxA,下计数比较点为 CCRxB
7. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
8. 清除相关中断标志
9. 使能相应的中断

10. 设置输出极性 FLT.CCPAx FLT.CCPBx
11. 设置死区使能 DTR.DTEN
12. 设置死区时间 DTR.DTR
13. 使能输出 DTR.MOE
14. 使能定时器 M23CR.CTEN

捕获功能设置

设置 CH0B 为上升沿捕获功能

1. 选择计数器计数方式, 设置 M23CR.MODE =2
2. 选择 CH0B 为捕获模式 CRCHx.CSB=1
3. 根据需求选择输入滤波 FLT.FLTB0
4. 选择捕获的边沿 CRCHx.CRB=1
5. 设置 ARR 改变周期值
6. 启动定时器 M23CR.CTEN
7. 清除相关中断标志
8. 清除捕获标志, 使能相应中断
9. 查询到捕获标志后, 读取 CCR0B 获得捕获值

注: 定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

互补 PWM 输出时使用 GATE 作为 PWM 输出功能

1. 设置为 PWM 互补输出模式, 参考“中心对齐互补 PWM 输出设置”
2. 设置 CCR0B 设置 GATE PWM 比较值
3. 设置 FLT.OCMB0=6/7 设置 PWM 输出模式
4. 使能 PWM 输出 DTR.MOE

互补 PWM 输出时使用 GATE 下降沿作为捕获输入功能

1. 设置为 PWM 互补输出模式, 参考“中心对齐互补 PWM 输出设置”
2. 设置 GATE 作为捕获输入 M23CR.CSG

3. 选择 GATE 捕获输入边沿设置 M23CR.CFG(GATE 作为捕获输入 B 通道滤波设置无效)
4. 启动定时器 M23CR.CTEN
5. 清除相关中断标志
6. 清除捕获标志, 使能相应中断
7. 查询到捕获标志后, 读取 CCR0B 获得捕获值

28.2.7 模式 2/3 从模式

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

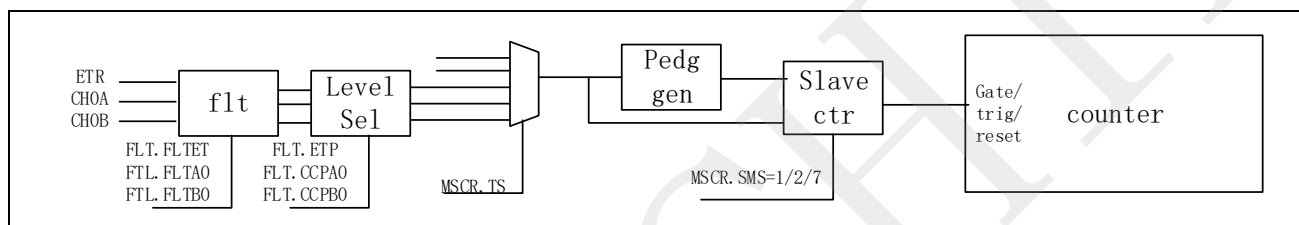


图 28-50 从模式示意图

从模式功能选择通过 MSCR.SMS 选择;

MSCR.SMS	从模式功能选择 001: 复位功能; 010: 触发模式; 111: 门控功能;
MSCR.TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP, 滤波功能可配置; 001: 内部互联信号 ITR0; 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 端口CH0A的边沿信号; 110: 端口CH0A的滤波相位选择后的信号IAFP, 滤波功能可配置; 111: 端口CH0B的滤波相位选择后的信号IBFP, 滤波功能可配置。

28.2.7.1 门控计数

按照选中的输入端电平使能计数器。在如下的例子中, 计数器只在 CH0A 为低电平时向上计数:

- 配置通道 CH0A 以检测 CH0A 上的低电平。配置输入滤波器带宽(本例中, 不需要滤波, 所以保持 ATIM_FLTR.FLTA0=000)。置 ATIM_FLTR 寄存器中 CCPA0=1 以

确定极性(只检测低电平)。

- 置 MSCR 寄存器中 SMS=111,配置定时器为门控模式;置 MSCR 寄存器中 TS=110,选择 CH0A 作为输入源。
- 置 M23CR 寄存器中 CTEN=1,启动计数器。在门控模式下,如果 CTEN=0,则计数器不能启动,不论触发输入电平如何。只要 CH0A 为低,计数器开始依据内部时钟计数,一旦 CH0A 变高则停止计数。

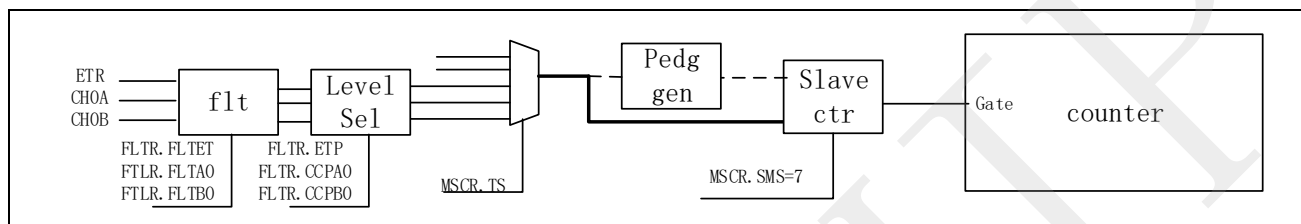


图 28-51 门控功能

28.2.7.2 触发功能

使用外部触发 (CH0A、CH0B、ETR) 可以使定时器同步启动,使用定时器内部互联信号结合 MSCR.MSM 也可以配置定时器同步启动。触发信号为输入信号的上升沿。也可以使用软件写 M23CR.TG 启动软件触发功能。

输入端上选中的事件使能计数器。在下面的例子中,计数器在 CH0B 输入的上升沿开始向上计数:

- 配置通道 CH0B 检测 CH0B 的上升沿。配置输入滤波器带宽(本例中,不需要任何滤波器,保持 FLT.FLTB0=000)。设 FLTR 寄存器中 CCPB0=0 以确定极性(不反向)。
- 置 MSCR 寄存器中 SMS=010,配置定时器为触发模式;置 MSCR 寄存器中 TS=111,选择 CH0B 作为输入源。当 CH0B 出现一个上升沿时,计数器开始在内部时钟驱动下计数,同时设置 TIF 标志。CH0B 上升沿和计数器启动计数之间的延时,取决于 CH0B 输入端的重同步电路。

注意: 如果使用下降沿触发,先选择触发极性,然后再选择模式,否则会产生误触发。

28.2.7.3 复位计数

在发生一个触发输入事件时,计数器和它的预分频器能够重新被初始化;同时,如果 M23CR 寄存器的 URS 位为低,还产生一个更新事件 UEV;然后所有的预装载寄存

器(Arr, CCRx)都被更新了。

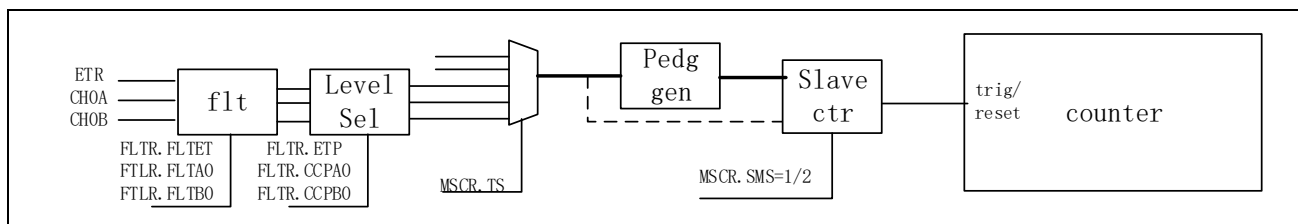


图 28-52 触发和复位功能

28.2.8 正交编码计数功能

MSCR.SMS=4/5/6，对应正交编码模式的模式 1/2/3。这时计数器根据 IAFP, IBFP 的相位关系进行编码计数。IAFP,IBFP 为端口输入 CH0A, CH0B 的滤波相位选择的信号。

模式 1 使用 CH0A 的边沿计数。模式 2 使用 CH0B 的边沿计数。模式 3 使用 CH0A,CH0B 的边沿都计数。

为了保证计数相位的正确性，需要保证 A/B 输入的相位大于一个脉冲宽度的相位差，A/B 输入脉冲宽度需要大于两个脉冲宽度。

			IAFP		IBFP	
	IBFP	IAFP	Rising	falling	Rising	falling
MOD1	High		Down	Up	–	–
	Low		up	Down	–	–
MOD2		High	–	–	up	Down
		Low	–	–	Down	Up
MOD3	High	High	Down	Up	up	Down
	Low	Low	up	Down	Down	Up

CHAF/CHBF 为端口 CH0A/CH0B 滤波的信号，IAFP/IBFP 为端口滤波相位选择后的信号。

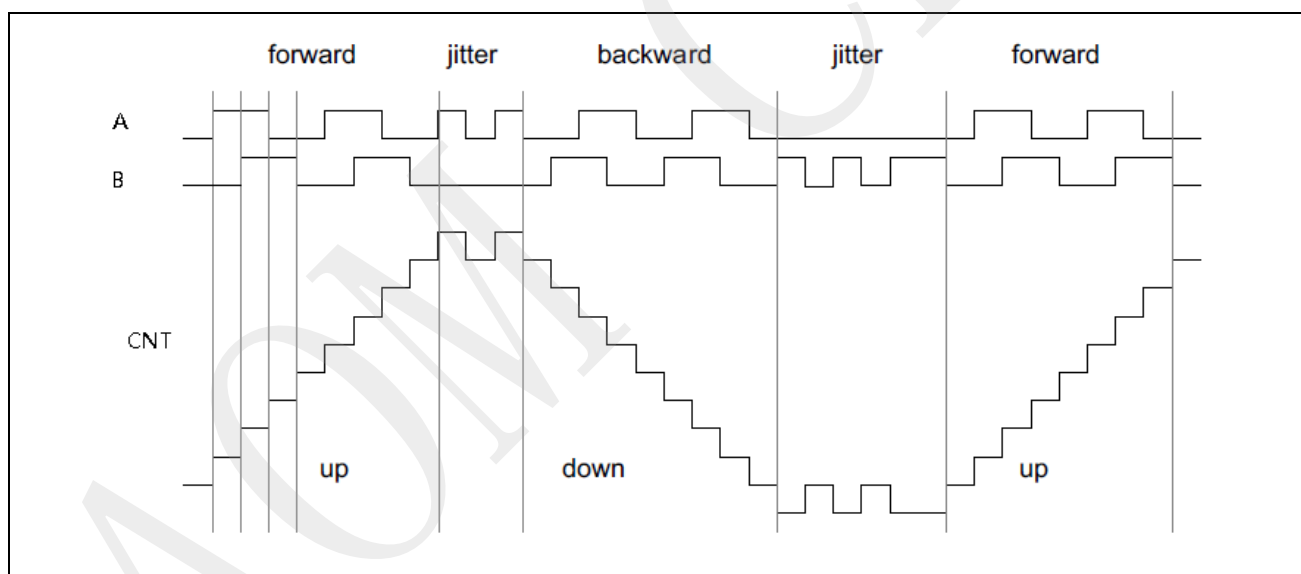
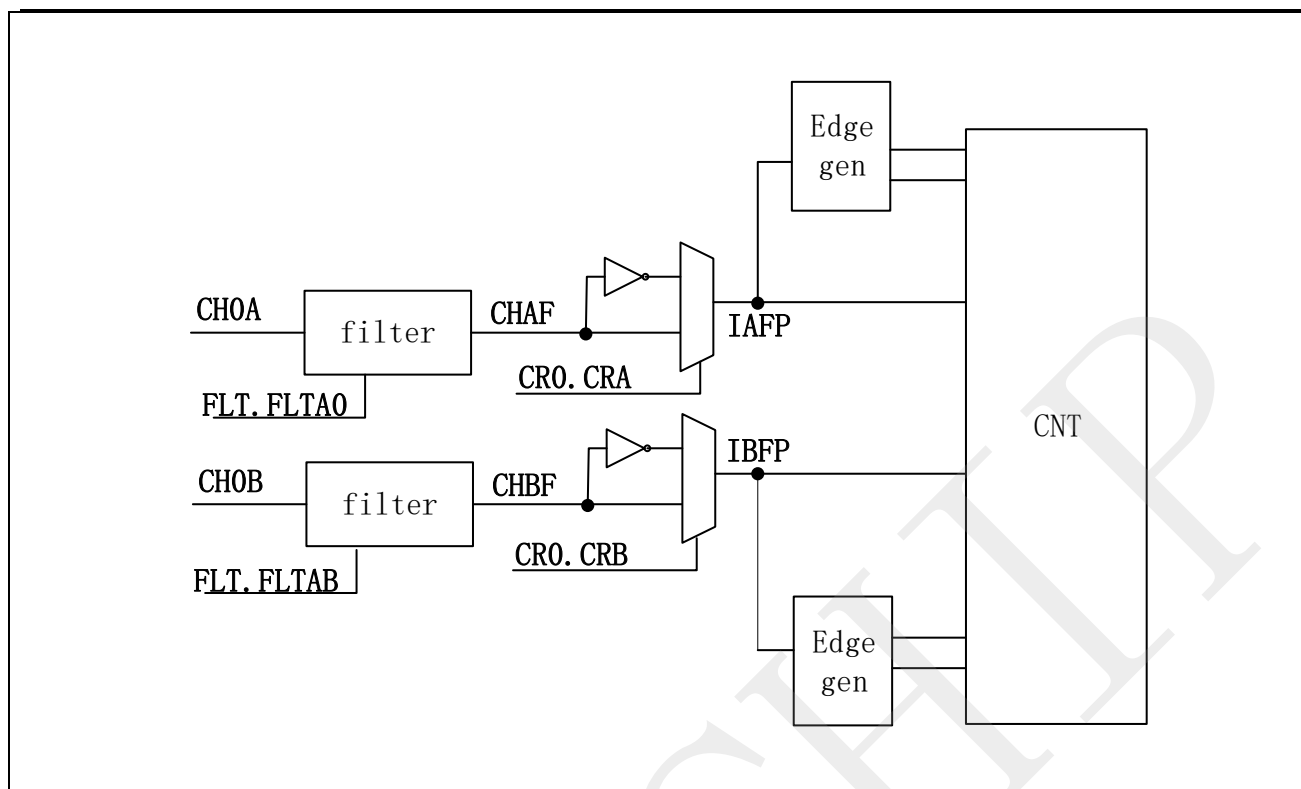


图 28-53 编码计数

28.2.9 Timer 触发 ADC

CCR0A, CCR1A, CCR2A 比较匹配可以配置触发 ADC，中心对齐 PWM 时只能通过控制寄存器 M23CR.CIS 可以选择在上升匹配触发或者下降匹配触发。

CCR0B, CCR1B, CCR2B 比较匹配可以配置触发 ADC，中心对齐 PWM 时，可以通过寄存器 CRx.CISB 分别控制三个匹配触发点（上升，下降，上升下降）。

28.2.12 ETR 输入互联

ETR 输入可以从端口直接输入,也可通过端口功能寄存器 MUXETR 选择可以连通到其他模块或端口。

当 ATIM_ETR=0x0 时,外部时钟 ETR 输入为 PxAFR 选择的端口输入,当 ATIM_ETR=0x1~0x7 时,连接其他模块的输入或输出。

0	1	2	3	4	5	6	7
ATIM_ETR	UART1_RX	UART2_RX	VC1	PE00	PA00	PB00	PC00

28.2.13 CHx 捕获输入互联

ATIM 的 CH0A/CH0B 输入可以从端口直接输入,也可通过 MUXCAP 寄存器选择可以连通到其他模块或端口。

当 MUXCAP[2:0]=0x0 时,捕获输入是 PxAFR 选择的端口输入,当 MUXCAP[2:0]=0x1~0x7 时,连接其他模块的输入或输出。

0	1	2	3	4	5	6	7
ATIM_CH0B	UART1_RX	UART2_RX	VC1	PE06	PA08	PB08	PC08

28.2.14 DMA

ATIM 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器,或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求,A 请求触发源可选择比较捕获 A,外部触发;B 请求触发源可选择比较捕获 B,事件更新。

IDREQ	Interrupt Signal of Peripheral
24	ATIMA
25	ATIMB

ATIMA 的触发源可选择比较捕获 A,ETRIG

ATIMB 的触发源可选择比较捕获 B,UEV

28.2.14.1 设置示例

捕获数据 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考数据捕获流程设置数据捕获
9. 使能 CRCHx.CDy 使能捕获数据触发 DAM 传输。

Note: 源地址设置为捕获通道寄存器，源地址固定，目标地址加。

脉冲调整 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考脉冲调整输出设置脉冲调制输出
9. 根据需要选择硬件触发 DMA 的条件（UEV 触发或者比较匹配触发）

Note: 目标地址设置为捕获通道寄存器，源地址变化，目标地址固定。

28.3 Timer 寄存器描述

ATIM 基地址: 0x40012C00

寄存器	偏移地址	描述
ATIM_ARR	0x000	Timer重载寄存器/周期
ATIM_CNT	0x004	Timer 16位模式计数寄存器
ATIM_CNT32	0x008	Timer 32位模式计数寄存器 注: ATIM_CNT32是TIMx_ARR与TIMx_CNT的重映射寄存器。为了方便32位定时器写操作。
ATIM_M0CR	0x00C	Timer模式0控制寄存器 (按不同模式进行描述)
ATIM_M1CR	0x00C	Timer模式1控制寄存器 (按不同模式进行描述)
ATIM_M23CR	0x00C	Timer模式23控制寄存器 (按不同模式进行描述)
ATIM_IFR	0x010	Timer中断标志
ATIM_ICLR	0x014	Timer中断清除寄存器
ATIM_MSCR	0x018	主从模式控制
ATIM_FLTR	0x01C	滤波控制
ATIM_ADTR	0x020	ADC触发控制
ATIM_CR0	0x024	比较单元0控制寄存器
ATIM_CRCH1	0x028	比较单元1控制寄存器
ATIM_CRCH2	0x02C	比较单元2控制寄存器
ATIM_DTR	0x030	死区寄存器
ATIM_RCR	0x034	重复计数寄存器
ATIM_CCR0A	0x03C	比较0A寄存器
ATIM_CCR0B	0x040	比较0B寄存器
ATIM_CCR1A	0x044	比较1A寄存器
ATIM_CCR1B	0x048	比较1B寄存器
ATIM_CCR2A	0x04C	比较2A寄存器
ATIM_CCR2B	0x050	比较2B寄存器

表 28-1 Timer 寄存器列表

28.4 模式 0 定时器寄存器描述

28.4.1 16 位模式重载寄存器（ATIM_ARR）

偏移地址：0x000

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	功能描述
31:16	Reserved	保留位
15:0	ARR	16位重载定时器重载值

28.4.2 16 位模式计数寄存器（ATIM_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	功能描述
31:16	Reserved	保留位
15:0	CNT	16位重载定时器计数值

28.4.3 32 位模式计数寄存器 (ATIM_CNT32)

偏移地址: 0x008

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT32[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT32[15:0]															
RW															

位	符号	功能描述
31:0	CNT32	32位定时器计数值 注: 仅在模式0 32位定时器自由计数模式下有效, 其他模式禁止写该寄存器

28.4.4 控制寄存器 (ATIM_M0CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
Reserved				Mode				Reserved	UIE		GATEP		GATE		
				RW					RW		RW		RW		
7		6		5		4		3		2		1		0	
Reserved		PRS						TOGEN		CT		MD		CTEN	
		RW						RW		RW		RW		RW	

位	符号	功能描述
31:14	Reserved	保留位
13:12	MODE	工作模式 00: 定时器模式0; 01: PWC模式; 10: 锯齿波模式; 11: 三角波模式;
11	Reserved	保留位
10	UIE	中断使能控制, 写1后使能中断

9	GATEP	端口 GATE 极性控制 0: 端口GATE高电平有效 1: 端口GATE低电平有效
8	GATE	定时器门控使能 0: 无门控, CTEN=1 时定时器工作 1: 门控使能, 端口 GATE 有效并且CTEN=1时定时器才工作
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;
3	TOGEN	模式0下翻转输出使能 1: 翻转输出使能 0: 翻转输出关闭CHA, CHB输出为低电平
2	CT	计数时钟选择 0: 内部计数时钟TCLK 1: 外部计数时钟ETR
1	MD	模式选择32定时/16定时模式选择 0: 32位自由计数 1: 16位重载计数
0	CTEN	定时器使能 0: 定时器停止 1: 定时器使能

28.4.5 中断标志寄存器（ATIM_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														UIF	
														RO	

位	符号	功能描述
31:1	RSV	保留位
0	UIF	溢出中断标志

28.4.6 中断标志清除寄存器（ATIM_ICLR）

偏移地址：0x014

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														UIF	
														R1W0	

位	符号	功能描述
31:1	REV	保留位
0	UIF	溢出中断标志清除位，写0清除

28.4.7 死区时间寄存器 (ATIM_DTR)

偏移地址: 0x030

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			MOE	Reserved											
			RW												

位	符号	功能描述
31:13	Reserved	保留位
12	MOE	翻转输出使能 0: 翻转输出为输入状态 1: 翻转端口为输出状态
11:0	Reserved	保留位

28.5 脉冲宽度测量 PWC 寄存器描述

28.5.1 16 位模式计数寄存器 (ATIM_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	16位定时器计数值

28.5.2 控制寄存器（ATIM_M1CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved																	
15				14		13		12		11		10		9		8	
Reserved		Oneshot		Mode				Reserved		UIE		Edg2nd		Edg1st			
		RW		RW						RW		RW		RW			
7				6		5		4		3		2		1		0	
Reserved		PRS						Reserved		CT		Reserved		CTEN			
		RW								RW				RW			

位	符号	功能描述				
31:15	Reserved	保留位				
14	Oneshot	单次触发模式选择 1：完成一次脉冲测量自动结束，再次测量需要重新使能CTEN 0：循环测量				
13:12	MODE	工作模式 00：定时器模式0； 01：PWC模式； 10：锯齿波模式； 11：三角波模式；				
11	Reserved	保留位				
10	UIE	中断使能控制，写1后使能中断 计数到0xFFFF才会溢出并产生溢出标志				
9	Edg2nd	脉冲宽度测量结束边沿选择				
8	Edg1st	脉冲宽度测量开始边沿选择				
		edg2nd Edg1st	00	01	10	11
		测量	上沿-上沿周期	低电平宽度	高电平宽度	下沿-下沿周期
7	Reserved	保留位				
6:4	PRS	内部时钟分频选择 000：1； 001：2； 010：4； 011：8； 100：16； 101：32； 110：64； 111：256；				
3	Reserved	保留位				
2	CT	计数时钟选择 0：内部计数时钟TCLK 1：外部计数时钟ETR				
1	Reserved	保留位				

0	CTEN	脉冲宽度测量使能 0: 禁止 1: 使能

28.5.3 中断标志寄存器（ATIM_IFR）

偏移地址：0x010

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CA0F	RSV	UIF
													RO		RO

位	符号	功能描述
31:3	RSV	保留位
2	CA0F	脉冲宽度测量中断标志
1	RSV	保留位
0	UIF	溢出中断标志

28.5.4 中断标志清除寄存器（ATIM_ICLR）

偏移地址：0x014

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CA0F	RSV	UIF
													R1W0		R1W0

位	符号	功能描述
31:3	RSV	保留位
2	CA0F	脉冲宽度测量中断标志清除，写0清除
1	RSV	保留位
0	UIF	溢出中断标志清除，写0清除

28.5.5 主从模式控制寄存器（ATIM_MSCR）

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			IB0S	IA0S	Reserved			TS		Reserved					
			RW	RW				RW							

位	符号	功能描述
31:13	RSV	保留位
12	IB0S	CH0B 输入选择 0: CH0B 1: 内部触发TS选择信号 注：当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0: CH0A; 1: CH0A CH1A CH2A XOR(ATIM) 0: CH0A; 1: CH0A ETR GATE XOR(TIM0) 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	RSV	保留位
7:5	TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1 011: 内部互联信号 ITR2 100: 内部互联信号 ITR3 101: 端口CH0A的边沿信号 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口CH0B的滤波相位选择后的信号IBFP
4:0	RSV	保留位

28.5.6 输出控制滤波（ATIM_FLTR）

偏移地址：0x01C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET			Reserved											
RW	RW														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FLTB0		Reserved	FLTA0			
									RW			RW			

位	符号	功能描述
31	ETP	ETR 输入相位选择 0: 同相位 1: 反向输入
30:28	FLTET	ETR滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效
27:7	Reserved	保留位
6:4	FLTB0	CHB 输入滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效
3	Reserved	保留位
2:0	FLTA0	CHA 输入滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效

28.5.7 控制寄存器 (ATIM_CR0)

偏移地址: 0x024

复位值: 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							CIEA	Reserved							
							RW								

位	符号	功能描述
31:9	Reserved	保留位
8	CIEA	CIEA 脉冲宽度测量完成中断使能 0: 禁止 1: 使能
7:0	Reserved	保留位

28.5.8 比较捕获寄存 (ATIM_CCR0A)

偏移地址: 0x03C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0A															
R															

位	符号	功能描述
31:16	Reserved	保留位
15:0	CCR0A	脉冲宽度测量结果

28.6 模式 2,3 寄存器描述

28.6.1 16 位模式重载寄存器（ATIM_ARR）

偏移地址：0x000(0X038 dummy address)

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	功能描述
31:16	Reserved	保留位
15:0	ARR	重载寄存器/周期寄存器，具有缓存功能 在计数器未使能或者缓存没有使能时，缓存寄存器可以立刻更新

28.6.2 16 位模式计数寄存器（ATIM_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW															

位	符号	功能描述
31:16	Reserved	保留位
15:0	CNT	重载定时器计数值

注：使用 PWM 比较输出时，初始化 CNT 值需要小于 ARR 的值。

28.6.3 控制寄存器（ATIM_M23CR）

偏移地址：0x00C

复位值： 0x0060 0008

31	30	29	28	27	26	25	24
Reserved				DIR	BG	UG	TG
				RW/RO	W1	W1	W1
23	22	21	20	19	18	17	16
OCCE	CIS		BIE	TIE	TDE	URS	OCCS
RW	RW		RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
CSG	Oneshot	Mode		UDE	UIE	CFG	CRG
RW	RW	RW		RW	RW	RW	RW
7	6	5	4	3	2	1	0
BUFPEN	PRS			Pwm2s	CT	Comp	CTEN
RW	RW			RW	RW	RW	RW

位	符号	功能描述
31:28	Reserved	保留位
27	DIR	计数方向，只有在锯齿波模式下可以写。其他模式下只读，写无效 0：上计数 1：下计数 注：从其他模式切换到中心对齐模式DIR自动清0。软件事件更新和从模式外部触发复位模式DIR自动清零。
26	BG	软件刹车，自动清零 写1：产生软件刹车 写0：无效
25	UG	软件更新，自动清零 写1：产生软件更新 写0：无效 初始化计数器并更新缓存寄存器到相应寄存器（缓存使能），预除频计数器也会被清零。
24	TG	软件触发，自动清零，需要在触发模式SMS=2且mode=2/3下都可以触发。 写1：产生软件触发 写0：无效
23	OCCE	OCREF 清除使能 1：OCREF_CLR信号可以清除OCREF输出

		0: OCREF输出不受OCREF_CLR影响
22:21	CIS	中心对齐A比较中断模式(B比较中断单独控制在CRx寄存器中CISB) 00: 无中断 01: 上沿中断 10: 下沿中断 11: 上下沿都中断
20	BIE	刹车中断使能 1: 中断使能 0: 中断禁止
19	TIE	触发中断使能 1: 中断使能 0: 中断禁止
18	TDE	触发DMA使能 1: 中断使能 0: 中断禁止
17	URS	更新源 0: 上溢出/下溢出/软件更新UG/从模式复位 1: 上溢出/下溢出
16	OCCS	OCREF 清除源选择 0: 电压比较器VC输出, VC选择在VCx_OUTCFG寄存器设置 1: ETR端口滤波相位选择后的信号 当OCCE有效时, OC_clr可以清除OCREF的比较输出信号为零, (当OCMx>1时有效), 下一个uev事件后继续比较输出
15	CSG	GATE 在PWM互补模式下捕获/比较选择(只有在PWM互补输出时有效) 使用CCR0B作为GATE的比较或捕获通道 1: 捕获 0: 比较
14	Oneshot	单次触发模式选择 1: 发生事件更新后定时器停止 0: 循环计数
13:12	MODE	工作模式 00: 定时器模式0; 01: PWC模式; 10: 锯齿波模式; 11: 三角波模式;
11	UDE	更新DMA使能 1: 使能更新触发DMA 0: 禁止更新触发DMA
10	UIE	UIE 更新中断使能 1: 使能更新中断 0: 禁止更新中断
9	CFG	GATE作为捕获输入时, 下沿捕获有效控制(只有在PWM互补输出时有效) 1: 下沿捕获有效

		0: 下沿捕获无效
8	CRG	GATE作为捕获输入时, 上沿捕获有效控制 (只有在PWM互补输出时有效) 1: 上沿捕获有效 0: 上沿捕获无效
7	BUFPEN	重载缓存使能 1: 周期缓存使能, 写入后下个事件更新后才影响到周期值 0: 周期缓存无效, 写入后立刻影响周期值
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;
3	PWM2S	OCREFA双点比较选择(缺省值为1) 0: 双点比较使能, 使用CCRA, CCRB比较控制OCREFA输出 1: 单点比较使能, 只使用CCRA比较控制OCREFA输出 注: OCREFB不受影响, 仍然使用CCRB控制OCREFB输出
2	CT	计数时钟选择 0: 内部计数时钟TCLK 1: 外部计数时钟ETR
1	Comp	PWM互补输出模式选择 0: 独立PWM输出 1: 互补PWM输出
0	CTEN	定时器使能 0: 禁止 1: 使能 可以外部触发使能, 在oneshot模式下结束时该位自动清零

28.6.4 中断标志寄存器（ATIM_IFR）

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E								
RO	RO	RO	RO	RO	RO	RO	RO								
7	6	5	4	3	2	1	0								
CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	RSV	UIF								
RO	RO	RO	RO	RO	RO		RO								

位	符号	功能描述
31:16	Reserved	保留位
15	TIF	触发中断标志
14	BIF	刹车中断标志
13	CB2E	通道CH2B捕获数据丢失标志 0：无数据丢失 1：数据丢失（仅TIM3存在）
12	CB1E	通道CH1B捕获数据丢失标志 0：无数据丢失 1：数据丢失（仅TIM3存在）
11	CB0E	通道CH0B捕获数据丢失标志 0：无数据丢失 1：数据丢失
10	CA2E	通道CH2A捕获数据丢失标志 0：无数据丢失 1：数据丢失（仅TIM3存在）
9	CA1E	通道CH1A捕获数据丢失标志 0：无数据丢失 1：数据丢失（仅TIM3存在）
8	CA0E	通道CH0A捕获数据丢失标志 0：无数据丢失 1：数据丢失
7	CB2F	通道CH2B发生捕获/比较匹配标志 0：没有发生

		1: 发生 (仅TIM3存在)
6	CB1F	通道CH1B发生捕获/比较匹配标志 0: 没有发生 1: 发生 (仅TIM3存在)
5	CB0F	通道CH0B发生捕获/比较匹配标志 0: 没有发生 1: 发生
4	CA2F	通道CH2A发生捕获/比较匹配标志 0: 没有发生 1: 发生 (仅TIM3存在)
3	CA1F	通道CH1A发生捕获/比较匹配标志 0: 没有发生 1: 发生 (仅TIM3存在)
2	CA0F	通道CH0A发生捕获/比较匹配标志 0: 没有发生 1: 发生
1	RSV	保留位
0	UIF	事件更新中断标志 0: 没有发生 1: 发生

28.6.5 中断标志清除寄存器 (ATIM_ICLR)

偏移地址: 0x014

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1A	CA0E								
R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0								
7	6	5	4	3	2	1	0								
CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	RSV	UIF								
R1W0	R1W0	R1W0	R1W0	R1W0	R1W0		R1W0								

位	符号	功能描述
31:16	REV	保留位
15	TIF	触发中断标志清除, 写0清除
14	BIF	刹车中断标志清除, 写0清除

13	CB2E	通道CH2B捕获数据丢失标志清除，写0清除（仅TIM3存在）
12	CB1E	通道CH1B捕获数据丢失标志清除，写0清除（仅TIM3存在）
11	CB0E	通道CH0B捕获数据丢失标志清除，写0清除
10	CA2E	通道CH2A捕获数据丢失标志清除，写0清除（仅TIM3存在）
9	CA1E	通道CH1A捕获数据丢失标志清除，写0清除（仅TIM3存在）
8	CA0E	通道CH0A捕获数据丢失标志清除，写0清除
7	CB2F	通道CH2B捕获/比较匹配标志清除，写0清除（仅TIM3存在）
6	CB1F	通道CH1B捕获/比较匹配标志清除，写0清除（仅TIM3存在）
5	CB0F	通道CH0B捕获/比较匹配标志清除，写0清除
4	CA2F	通道CH2A捕获/比较匹配标志清除，写0清除（仅TIM3存在）
3	CA1F	通道CH1A捕获/比较匹配标志清除，写0清除（仅TIM3存在）
2	CA0F	通道CH0A捕获/比较匹配标志清除，写0清除
1	RSV	保留位
0	UIF	事件更新中断标志清除，写0清除

28.6.6 主从模式控制寄存器（ATIM_MSCR）

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			IB0S	IA0S	SMS			TS			MSM	CCDS	MMS		
			RW	RW	RW			RW			RW	RW	RW		

位	符号	功能描述
31:13	RSV	保留位
12	IB0S	CH0B 输入选择 0: CH0B 1: 内部触发TS选择信号 注：当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0: CH0A; 1: CH0A CH1A CH2A XOR(ATIM) 0: CH0A; 1: CH0A ETR GATE XOR(TIM0) 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	SMS	从模式功能选择 000: 使用内部时钟 001: 复位功能

		010: 触发模式 011: 外部时钟模式 100: 正交编码计数模式1 101: 正交编码计数模式2 110: 正交编码计数模式3 111: 门控功能		
7:5	TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1 011: 内部互联信号 ITR2 100: 内部互联信号 ITR3 101: 端口CH0A的边沿信号 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口CH0B的滤波相位选择后的信号IBFP		
4	MSM	主从选择 0: 无延时 1: 延时使能, 使主送计数器同时启动 注: 使用触发模式时, 从模式设置为0, 主模式设置为1, 可以使主从计数同时启动		
3	CCDS	比较模式下 DMA 比较触发选择 0: 比较匹配触发DMA 1: 比较匹配不触发DMA, 事件更新代替比较匹配触发DMA		
2:0	MMS	主模式输出选择, 用于内部互联, 连接到其他定时器的ITRx <table><tr><td>定时器0/1/2</td></tr><tr><td>000: 软件更新UG, 写CR. UG 001: 定时器使能CTEN 010: 定时器事件更新UEV 011: 比较匹配选择输出CMPSO 100: 定时器比较参数输出OCREF0A 101: 定时器比较参数输出OCREF0B 110: 定时器比较参数输出OCREF0B 111: 定时器比较参数输出OCREF0B</td></tr></table>	定时器0/1/2	000: 软件更新UG, 写CR. UG 001: 定时器使能CTEN 010: 定时器事件更新UEV 011: 比较匹配选择输出CMPSO 100: 定时器比较参数输出OCREF0A 101: 定时器比较参数输出OCREF0B 110: 定时器比较参数输出OCREF0B 111: 定时器比较参数输出OCREF0B
定时器0/1/2				
000: 软件更新UG, 写CR. UG 001: 定时器使能CTEN 010: 定时器事件更新UEV 011: 比较匹配选择输出CMPSO 100: 定时器比较参数输出OCREF0A 101: 定时器比较参数输出OCREF0B 110: 定时器比较参数输出OCREF0B 111: 定时器比较参数输出OCREF0B				

28.6.7 输出控制/输入滤波 (ATIM_FLTR)

偏移地址: 0x01C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET			BKP	FLTBK			CCPB2	OCMB2			CCPA2	OCMA2		
-	-			-	-			-	FLTB2			-	FLTA2		
RW	RW			RW	RW			RW	RW			RW	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCPB1	OCMB1			CCPA1	OCMA1			CCPB0	OCMB0			CCPA0	OCMA0		
-	FLTB1			-	FLTA1			CCPB0	FLTB0			CCPA0	FLTA0		
RW	RW			RW	RW			RW	RW			RW	RW		

位	符号	功能描述
31	ETP	ETR 输入相位选择 0: 同相位 1: 反向输入
30:28	FLTET	ETR滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效
27	BKP	刹车BK输入相位选择 0: 同相位 1: 反向输入
26:24	FLTBK	刹车输入滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效 注: 为了保证OCMB0的PWM输出设置, GATE 在PWM互补模式下作为捕获输入, 滤波设置无效。
23	CCPB2	比较功能: CH2B通道比较输出相位控制 0: 正常输出 1: 反向输出
22:20	OCMB2	比较功能: CH2B通道比较控制; 参考OCMB0

	FLTB2	捕获功能：CH2B输入通道滤波设置，参考FLTBK
19	CCPA2	比较功能：CH2A通道比较输出相位控制 0：正常输出 1：反向输出
18:16	OCMA2 FLTA2	比较功能：CH2A通道比较控制；参考OCMB0 捕获功能：CH2A输入通道滤波设置，参考FLTBK
15	CCPB1	比较功能：CH1B通道比较输出相位控制 0：正常输出 1：反向输出
14:12	OCMB1 FLTB1	比较功能：CH1B通道比较控制，参考OCMB0 捕获功能：CH1B输入通道滤波设置，参考FLTBK
11	CCPA1	比较功能：CH1A通道比较输出相位控制 0：正常输出 1：反向输出
10:8	OCMA1 FLTA1	比较功能：CH1A通道比较控制；参考OCMB0 捕获功能：CH1A输入通道滤波设置，参考FLTBK
7	CCPB0	比较功能：输出比较模式 CCPBx比较输出CHBx端口极性控制 0：正常输出 1：反向输出 编码计数与从模式门控功能：输入相位控制 CCPB0从模式门控，复位，外部触发，外部时钟使用CH0B端口输入极性控制 0：正常输入 1：反向输入
6:4	OCMB0 FLTBO	比较功能：CH0B通道比较控制 000：强制为0 001：强制为1 010：比较匹配时强制为0 011：比较匹配时强制为1 100：比较匹配时翻转 101：比较匹配时输出一个计数周期的高电平 110：PWM 模式1 单点比较： 上计数时 $CNT < CCR_{xy}$ 输出高，下计数时 $CNT > CCR_{xy}$ 输出为低电平 双点比较： 1) 锯齿波上计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为低电平 2) 锯齿波下计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为高电平 3) 三角波上计数 $CNT < CCR_{xA}$ 输出高，下计数 $CNT > CCR_{xB}$ 为低电平 111 PWM 模式2 单点比较： 上计数时 $CNT < CCR_{xy}$ 输出低，下计数时 $CNT > CCR_{xy}$ 输出为高电平 双点比较：

		1) 锯齿波上计数 $CCRxA \leq CNT < CCRxB$ 输出为高电平 2) 锯齿波下计数 $CCRxA \leq CNT < CCRxB$ 输出为低电平 3) 三角波上计数 $CNT < CCRxA$ 输出低, 下计数 $CNT > CCRxB$ 为高电平 捕获功能: CH0B输入通道滤波设置, 参考FLTBK
3	CCPA0	比较功能: CCPAx比较输出CHAx端口极性控制 0: 正常输出 1: 反向输出 编码计数与从模式门控功能: 输入相位控制 CCPA0从模式门控, 复位, 外部触发, 外部时钟使用CH0A端口输入极性控制 0: 正常输入 1: 反向输入
2:0	OCMA0 FLTA0	比较功能: A通道比较控制, 参考OCMB0 捕获功能: CH0A输入通道滤波设置, 参考FLTBK

28.6.8 ADC 触发控制寄存器 (ATIM_ADTR)

偏移地址: 0x020

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
ADTE	CMB2E	CMB1E	CMB0E	CMA2E	CMA1E	CMA0E	UEVE																
RW	RW	RW	RW	RW	RW	RW	RW																

位	符号	功能描述
31:8	Reserved	保留位
7	ADTE	使能ADC触发全局控制 1: 使能 0: 禁止
6	CMB2E	通道2B比较匹配触发ADC使能 1: 使能 0: 禁止
5	CMB1E	通道1B比较匹配触发ADC使能 1: 使能 0: 禁止
4	CMB0E	通道0B比较匹配触发ADC使能 1: 使能 0: 禁止
3	CMA2E	通道2A比较匹配触发ADC使能

		1: 使能 0: 禁止
2	CMA1E	通道1A比较匹配触发ADC使能 1: 使能 0: 禁止
1	CMAOE	通道0A比较匹配触发ADC使能 1: 使能 0: 禁止
0	UEVE	事件更新触发ADC使能 1: 使能 0: 禁止

28.6.9 通道 0 控制寄存器 (ATIM_CRCH0)

偏移地址: 0x024

复位值: 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	BKSB		BKSA		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		

位	符号	功能描述
31:16	Reserved	保留位
15	CCGB	捕获比较B软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12		B通道比较匹配设置 00: 无匹配 01: 上升匹配 10: 下降匹配 11: 双匹配
11	CDEB	B捕获比较触发DMA使能 0: 禁止 1: 使能

10	CDEA	A捕获比较触发DMA使能 0: 禁止 1: 使能
9	CIEB	B捕获比较触发中断使能 0: 禁止 1: 使能
8	CIEA	A捕获比较触发中断使能 0: 禁止 1: 使能
7	BUFEB	比较功能: B比较缓存使能控制 0: 禁止 1: 使能
6	BUFEA	比较功能: A比较缓存使能控制 0: 禁止 1: 使能
5	CSB	B通道捕获/比较功能选择 0: 比较模式 1: 捕获模式
4	CSA	A通道捕获/比较功能选择 0: 比较模式 1: 捕获模式
3	CFB	B通道下降沿捕获使能 0: 禁止 1: 使能
2	CRB	B通道上升沿捕获使能 0: 禁止 1: 使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00: 高阻输出 01: 对输出无影响 10: 强制输出低电平 11: 强制输出高电平
1	CFA	A通道下降沿捕获使能 0: 禁止 1: 使能
0	CRA	A通道上升沿捕获使能 0: 禁止 1: 使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00: 高阻输出 01: 对输出无影响 10: 强制输出低电平

11: 强制输出高电平

28.6.10 通道 1/2 控制寄存器 (ATIM_CRCH1/2)

偏移地址:

ATIM_CRCH1: 0x028

ATIM_CRCH2: 0x02C

复位值: 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	BKSB		BKSA		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		

位	符号	功能描述
31:16	Reserved	保留位
15	CCGB	捕获比较B软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00: 无匹配 01: 上升匹配 10: 下降匹配 11: 双匹配
11	CDEB	B捕获比较触发DMA使能 0: 禁止 1: 使能
10	CDEA	A捕获比较触发DMA使能 0: 禁止 1: 使能
9	CIEB	B捕获比较触发中断使能 0: 禁止 1: 使能
8	CIEA	A捕获比较触发中断使能

		0: 禁止 1: 使能
7	BUFEB	比较功能: B比较缓存使能控制 0: 禁止 1: 使能
6	BUFEA	比较功能: A比较缓存使能控制 0: 禁止 1: 使能
5	CSB	B通道捕获/比较功能选择 0: 比较模式 1: 捕获模式
4	CSA	A通道捕获/比较功能选择 0: 比较模式 1: 捕获模式
3	CFB	B通道下降沿捕获使能 0: 禁止 1: 使能
2	CRB	B通道上升沿捕获使能 0: 禁止 1: 使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00: 高阻输出 01: 保持之前输出1 10: 强制输出低电平 11: 强制输出高电平
1	CFA	A通道下降沿捕获使能 0: 禁止 1: 使能
0	CRA	A通道上升沿捕获使能 0: 禁止 1: 使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00: 高阻输出 01: 保持之前输出1 10: 强制输出低电平 11: 强制输出高电平

28.6.12 重复周期设置值 (ATIM_RCR)

偏移地址: 0x034

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RCR							
								RW							

位	符号	功能描述
31:8	Reserved	保留位
7:0	RCR	重复周期计数值 设置RCR+1个周期个上溢出下溢出后产生事件更新, 计数器上溢出或下溢出时内部RCR_CNT减1, 当计数到零后RCR_CNT重载RCR的值, 并且产生事件更新UEV信号。

28.6.13 通道 0 比较捕获寄存器 (ATIM_CCR0A/B)

偏移地址:

ATIM_CCR0A: 0x03C

ATIM_CCR0B: 0x040

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0y															
RW															

位	符号	功能描述
31:16	Reserved	保留位
15:0	CCR0y	比较捕获寄存器, 比较具有缓存功能(y=A, B)

28.6.14 通道 1/2 比较捕获寄存器 (ATIM_CCR1/2 A/B)

偏移地址:

ATIM_CCR1A: 0x044

ATIM_CCR1B: 0x048

ATIM_CCR2A: 0x04C

ATIM_CCR2B: 0x050

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCRxy															
RW															

位	符号	功能描述
31:16	Reserved	保留位
15:0	CCRxy	比较捕获, 比较具有缓存功能 (x=1, 2; y=A, B)

29. 通用定时器（GTIM1/2/3）

29.1 通用定时器简介

GTIM1/2/3 是分别由 1 个计数单元和 2 个比较单元组成的定时器。每个定时器支持 2 个独立 PWM 输出或 1 对互补 PWM 输出和 1 个独立 PWM 输出。支持 2 个捕获输入。

GTIM1/2/3 可以组成 3 对互补 PWM 输出。

使用定时器预分频、系统预分频器和系统时钟选择，可灵活的调整脉冲宽度和波形周期；可以方便的测量脉冲宽度。

29.1.1 基本特性(GTIM1/2/3)

- 2 路独立 PWM 输出 CHA,CHB, (端口对应 CH1,CH2)
- 1 路互补 PWM 输出 (CHA,CHB) +1 路独立 PWM 输出 (gate)
- 1 路互补 PWM 输出 (CHA,CHB) +1 路捕获功能(gate)
- 最多 2 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发

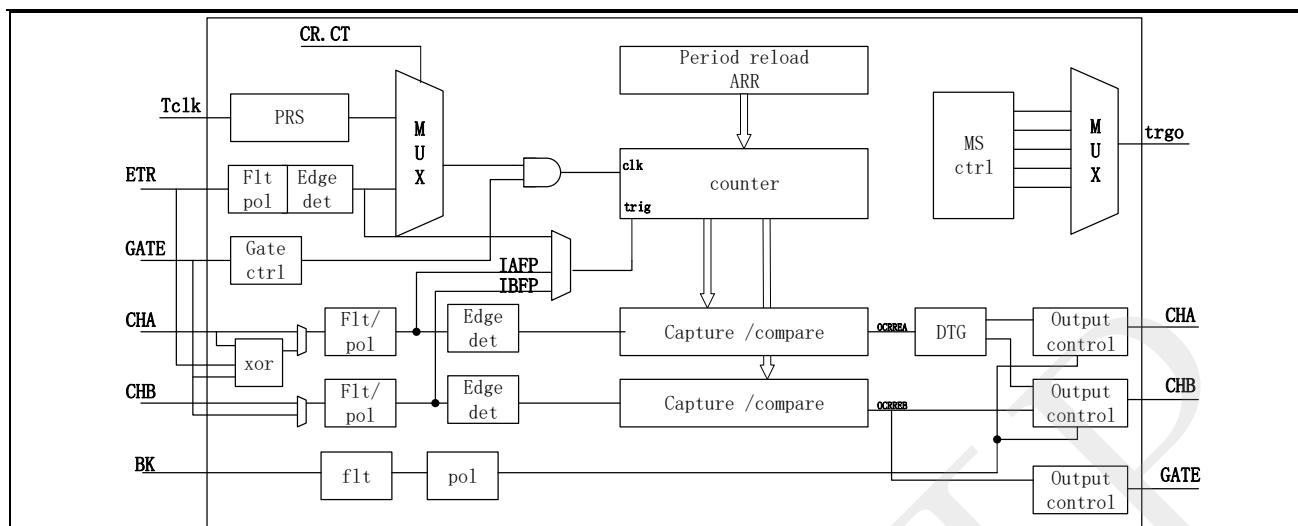


图 29-1 GTIM1/2/3 框图

29.2 Timer 功能描述

29.2.1 定时器时钟

定时器使用 PCLK 作为定时器时钟，下文定时器时钟使用 TCLK 标示。

29.2.2 定时计数器

可编程通用定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

29.2.3 定时器预除频

使用 TCLK 做为 Timer 定时器时钟时，可以使用预分频。预分频设置如下：

PRS	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	256

预分频器没有预装载缓存，因此对预分频进行更改，预分频将会立刻生效。

29.2.4 模式 0 计数定时器功能

这种模式下,计数器向上计数。计数器支持两种计数模式,重载模式与自由计数模式,可以选择外部时钟 ETR 计数或系统时钟计数。门控 Gate 可以控制计数屏蔽。计数到最大值溢出产生中断。翻转输出端口 CHA,CHB,这种模式下 CHA,CHB 反相。

重载模式计数范围从 ARR 向上计数到 0xFFFF 溢出,然后从 ARR 再开始计数,计数周期为 0xFFFF-ARR+1;自由计数模式从设置计数值计数到 0xFFFFFFFF 后溢出,溢出后计数值从 0x0 重新开始计数。

计数器可用通过外部门控功能控制计数器是否计数。控制关系如下

MOCR. GATE	MOCR. GATEP	端口 GATE 输入	MOCR. CTEN	计数器
x	x	x	0	不计数
0	x	x	1	计数
1	0	高电平	1	计数
1	0	低电平	1	不计数
1	1	高电平	1	不计数
1	1	低电平	1	计数

29.2.4.1 功能框图

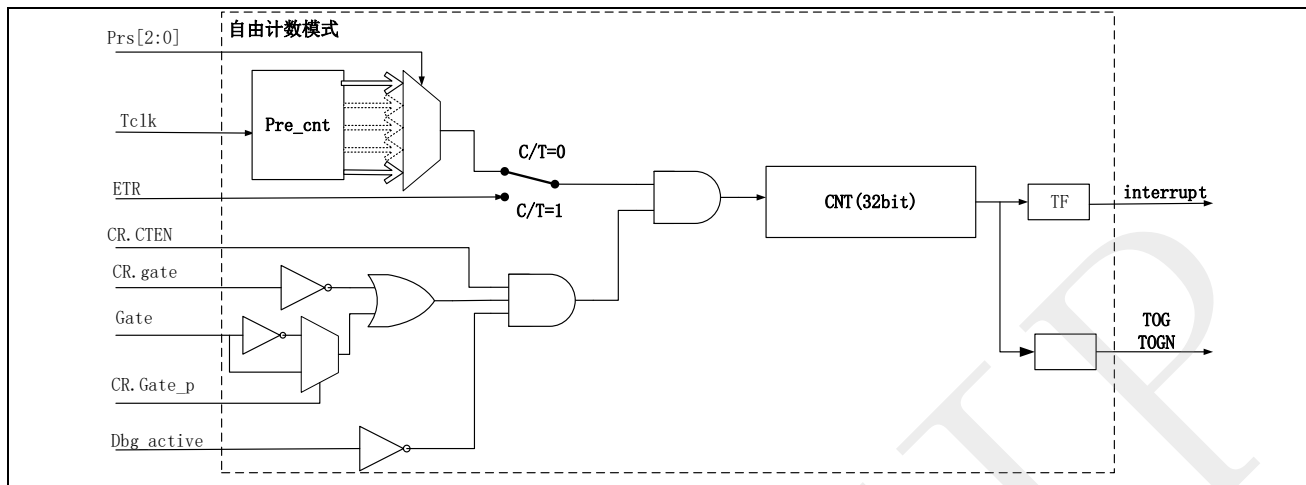


图 29-2 自由计数框图

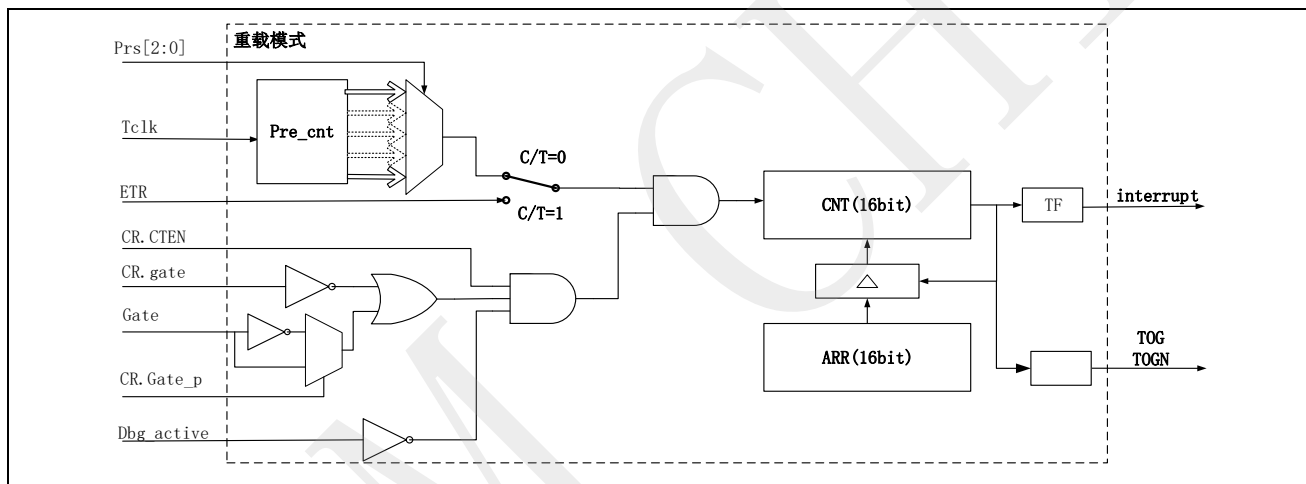


图 29-3 重载计数波形

29.2.4.2 计数波形

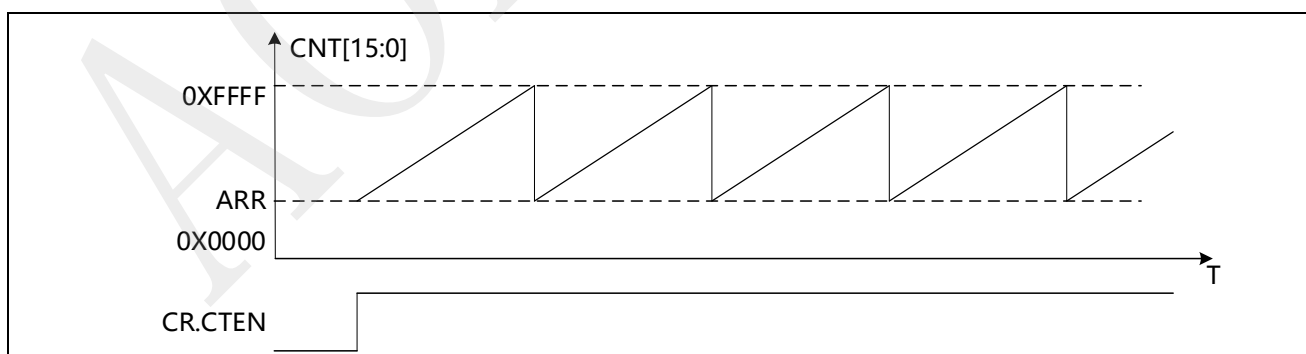


图 29-4 16位重载计数波形

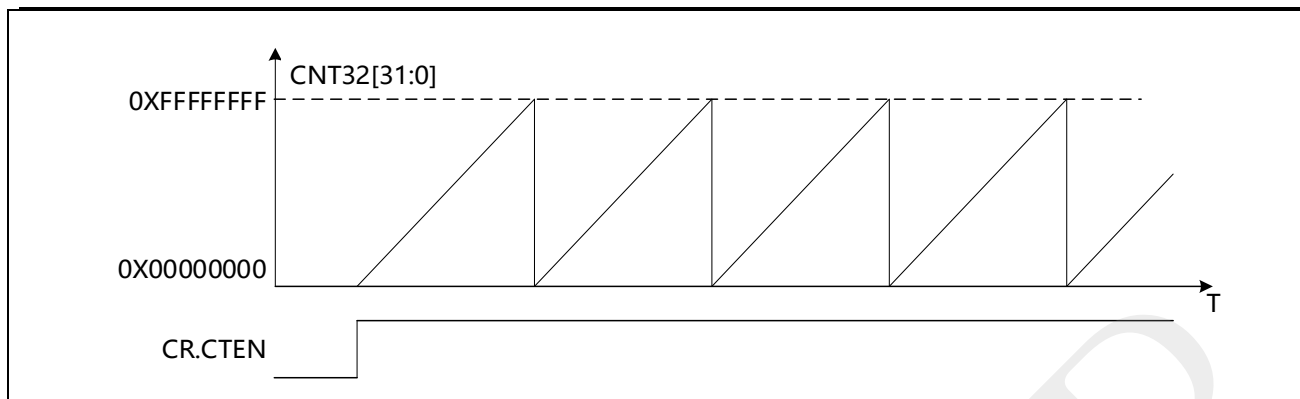


图 29-5 32位自由计数波形

29.2.4.3 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 tclk 采样，因此外部输入时钟频率不能超过系统的 tclk 时钟。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

29.2.4.4 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

29.2.4.5 时序图

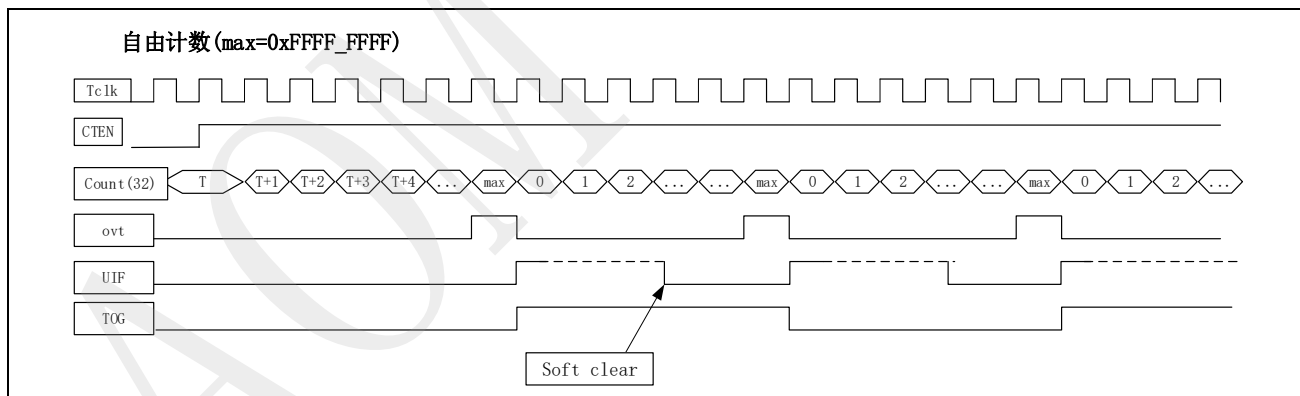


图 29-6 自由计数时序图

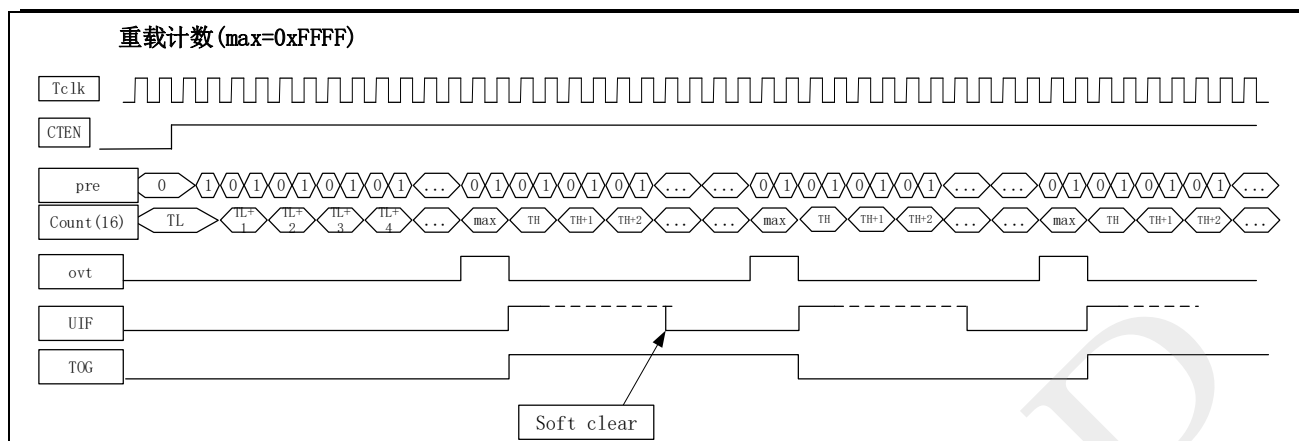


图 29-7 重载计数时序图 (预分频设置为2)

29.2.4.6 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。使用 toggle 输出需要使能 DTR.MOE 控制位。设置 CR.TOG_EN 为 0 可以同时设置端口 CHA,CHB 输出为 0。在计数时钟为 4M 情况下 Buzzer 输出不同频率的 timer 重载模式配置如下：

Buzzer 频率	计数器周期	计数器计数值	计数器重载值	CNTL 初始值	CNTH 重载值
1000Hz	0.5ms	2000	63536	0XF830	0XF830
2000Hz	0.25ms	1000	64536	0XFC18	0XFC18
4000Hz	0.125ms	500	65036	0XFE0C	0XFE0C

29.2.4.7 设置示例

重载定时器设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置装载值 ARR
3. 设置计数器初值 CNT
4. 清除中断标志
5. 使能中断 M0CR.UIE
6. 使能重载模式 M0CR.MD
7. 开启定时器 M0CR.CTEN

门控外部时钟自由计数设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置计数器初值 CNT

3. 使能门控功能 M0CR.GATE
4. 选择门控有效电平 M0CR.GATEP
5. 清除中断标志
6. 使能中断 M0CR.UIE
7. 使能外部时钟模式 M0CR.CT
8. 开启定时器 M0CR.CTEN

BUZZER 输出控制

1. 根据输出频率设置合适的 ARR 值
2. 设置定时器为重载模式，参考重载定时器设置
3. 使能输出使能 DTR.MOE
4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出。

29.2.5 模式 1 脉宽测量 PWC

这种模式下可以自动测量输入脉冲的高电平低电平或者周期宽度。

第一个有效边沿计数器初始化为 0x0001，第二个有效边沿将停止计数，并将当前计数值存入 CMAR，并且产生捕获中断 CAF，如果计数器发生溢出，会产生溢出标志。

设置溢出中断使能会产生溢出中断。

M1CR.edg1st	0	0	1	1
M1CR.edg2nd	0	1	0	1
脉宽测量	上沿~上沿 周期宽度	上沿~下沿 高电平宽度	下沿~上沿 低电平宽度	下沿~下沿 周期宽度

周期测量时，会间隔一个周期测量一个周期。

29.2.5.1 PWC 功能框图

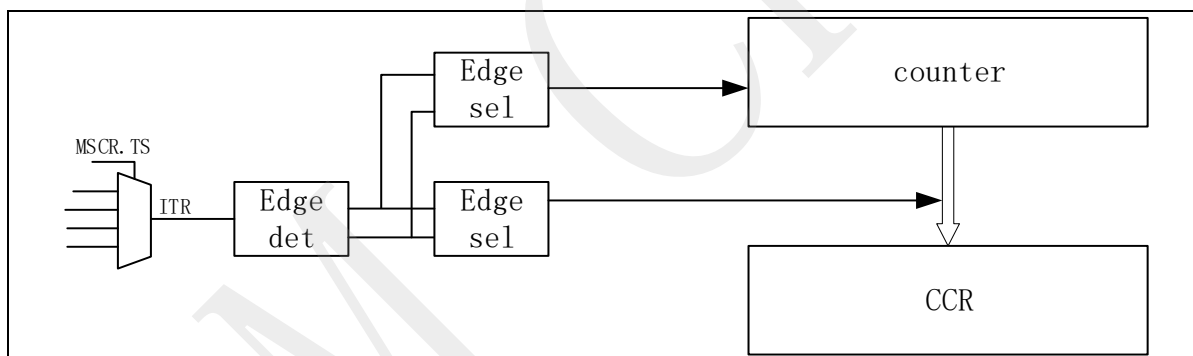


图 29-8 PWC测量框图

MSCR.TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP; 001: 内部互联信号 ITR0; 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 无效; 110: 端口CH0A的滤波后的信号IAFP（极性选择在脉宽测量模式下无效）; 111: 端口CH0B的滤波后的信号IBFP（极性选择在脉宽测量模式下无效）。

29.2.5.2 PWC 波形测量时序图

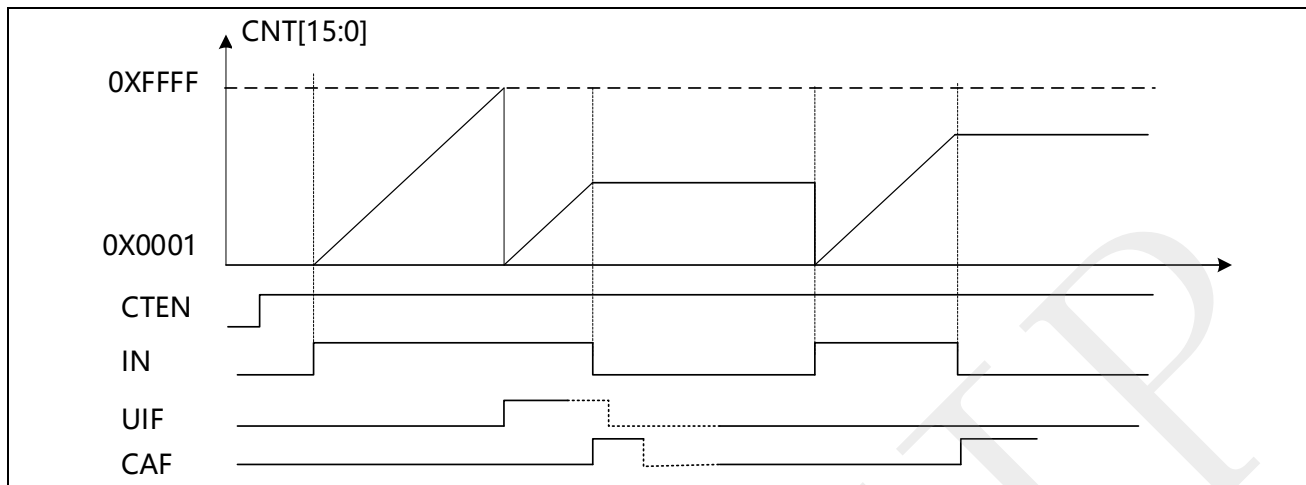


图 29-9 高电平脉冲宽度测量

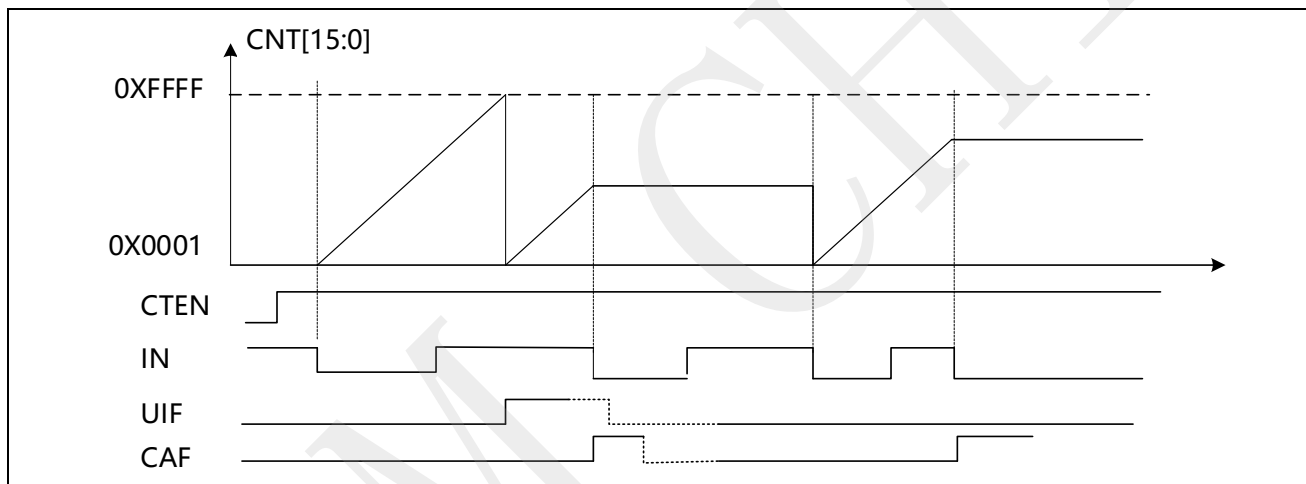


图 29-10 下降沿到下降沿周期测量

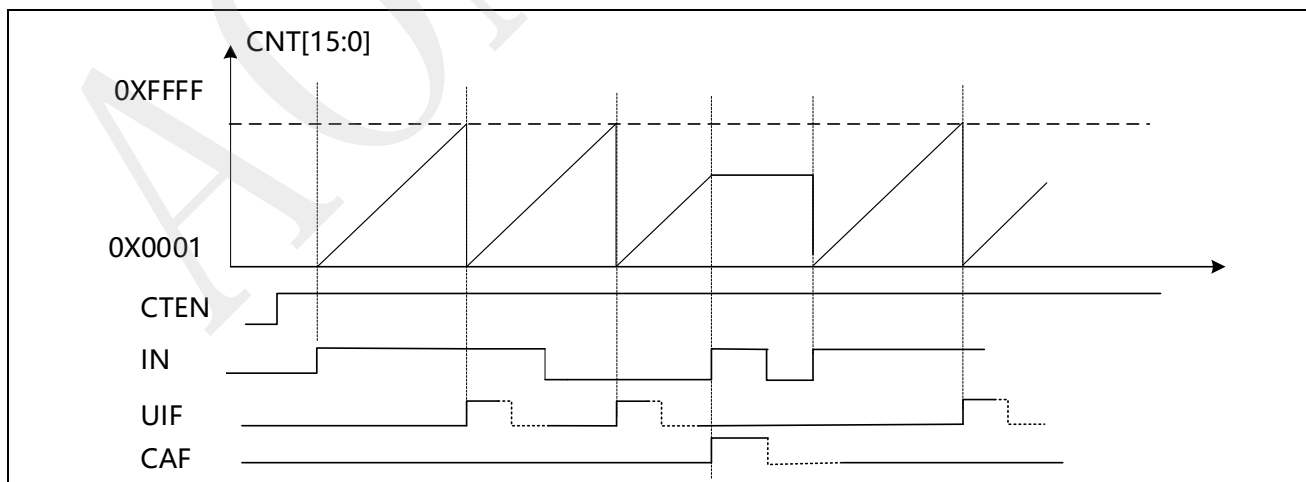


图 29-11 上升沿到上升沿周期测量

通过寄存 MSCR.TS 选择测量信号源。

000 ETFP: ETR 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

001 ITR0: Timer 内部互联信号 0, 其他 timer 的 TRGO 输出, 参考 timer 互联部分

010 ITR1: Timer 内部互联信号 1, 其他 timer 的 TRGO 输出

011 ITR2: Timer 内部互联信号 2, 其他 timer 的 TRGO 输出

100 ITR3: Timer 内部互联信号 3, 其他 timer 的 TRGO 输出

101 IA0ED: 无效

110 IAFP: CH0A 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

111 IBFP: CH0B 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

注: 关于 TRGO 输出参考寄存器描述。

29.2.5.3 PWC 单次触发模式

设置 M1CR.ONESHOT=1 可以设置 PWC 单次测量, 测量完成后 CTEN 将被清除。

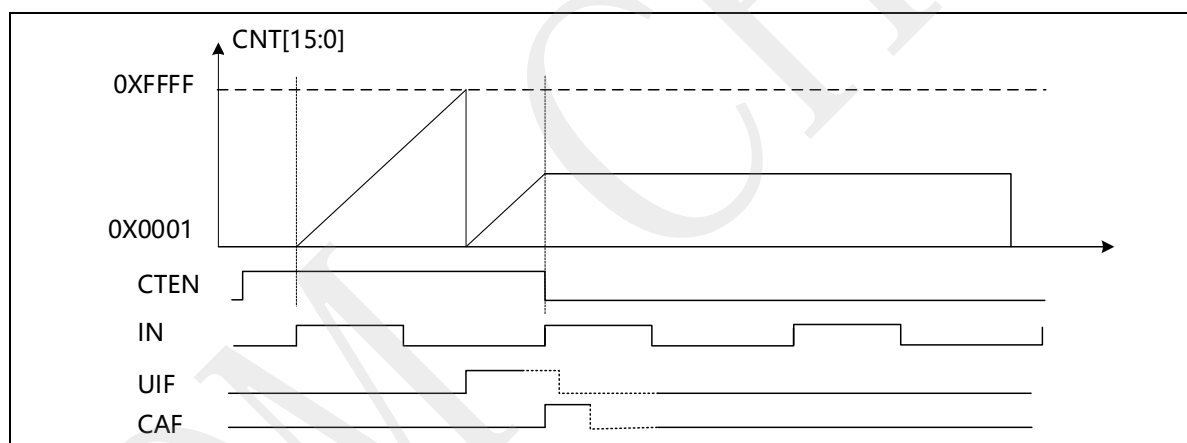


图 29-12 上升沿到上升沿周期测量单次模式

29.2.5.4 设置示例

脉冲低电平测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置 M1CR.edg2dn=0,M1CR.edg1st=1 选择测量低电平
4. 清除中断标志
5. 使能溢出中断 M1CR.UIE
6. 使能测量结束中断 CR0.CIEA
7. 使能定时器 M1CR.CTEN

8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
9. 等待下次测量

脉冲高电平单次测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置脉冲单次测量模式 M1CR.ONESHOT=1
4. 设置 M1CR.edg2dn=1, M1CR.edg1st=0 选择测量低电平
5. 清除中断标志
6. 使能溢出中断 M1CR.UIE
7. 使能测量结束中断 CR0.CIEA
8. 使能定时器 M1CR.CTEN
9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
10. 测量结束

29.2.6 模式 2/3 比较捕获模式

29.2.6.1 计数器

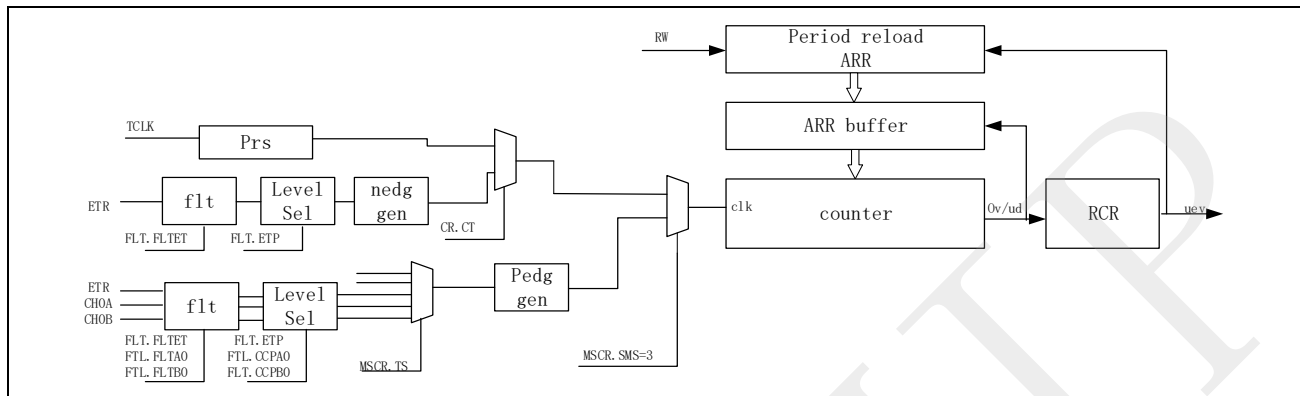


图 29-13 计数器框图

计数器主要部分是一个 16 位计数器与相关的自动装载寄存器。这个计数器可以向上计数（模式 2），向下计数（模式 2）或向上向下双向计数（模式 3）。计数器的时钟可以由预分频器 PRS 分频得到，也可以选择 ETR 输入外部时钟或者通过 MSCR.TS 选择的外部输入信号和内部互联信号。

- 计数器基本单元包括：
- 计数器寄存器 CNT
- 预分频寄存器 CR.PRS
- 自动装载寄存器 ARR
- 重复次数寄存器 RCR
- 时钟选择控制寄存器 FLT,CR0,MSCR,CR

自动装载寄存器具有缓存功能，计数器产生事件更新后重载值从缓存寄存器更新到计数器。当计数器停止状态或者缓存功能关闭状态，自动装载寄存器立刻更新到缓存寄存器。当定时器处于运行状态并行缓存功能有效时，写入到自动装载寄存器的值不会立刻更新到缓存寄存器，当事件更新后才有自动装载寄存器更新到缓存寄存器。

时钟选择及门控功能，触发功能，复位功能参考模式 2/3 从模式章节。

29.2.6.2 计数器波形

模式 2 为锯齿波计数波形，通过设置 CR.DIR 可以更改计数方向。

设置 CR.DIR 为 0 时，计数器为递增计数模式，这种模式下，计数器从 0 计数到自动重载值 (GTIMx_ARR)，然后重新从 0 开始计数并生成计数器上溢事件。如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次 (GTIMx_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 GTIMx_CR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

发生更新事件时，将更新所有寄存器且将更新标志 (GTIMx_IFR 寄存器中的 UIF 位) 置 1（取决于 URS 位）：

- 自动重载缓存值将以 ARR 寄存器值进行更新
- 比较缓存值将以比较寄存器 CCRxy 进行更新

以下图示显示 ARR=0X2C 时不同计数方向的计数器波形

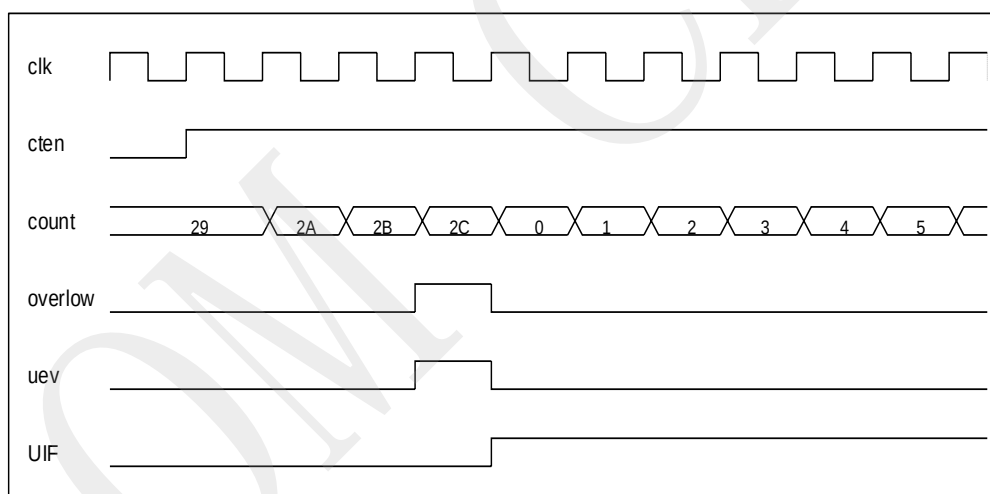


图 29-14 无预分频的向上计数

计数溢出周期时钟个数为 ARR+1，

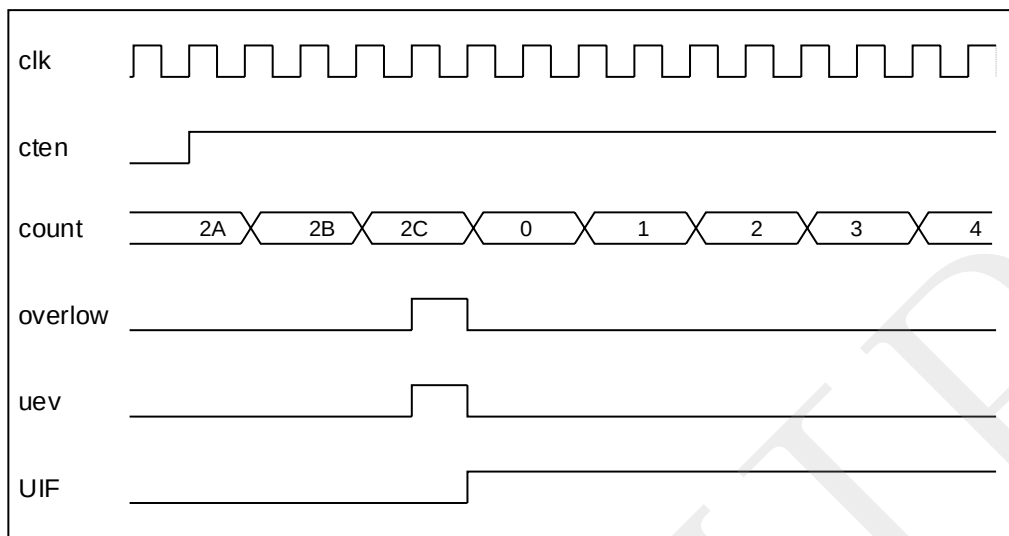


图 29-15 带预分频的上计数

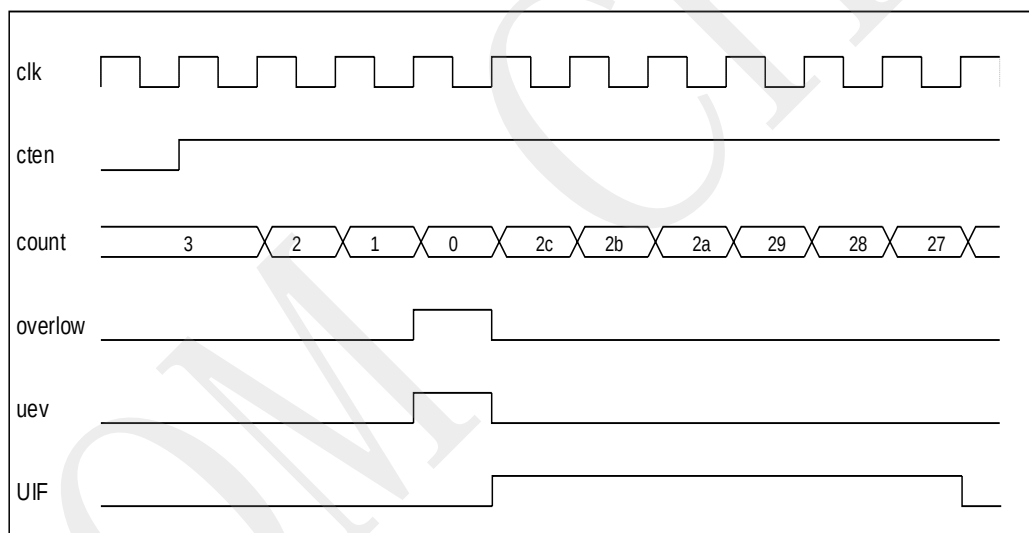


图 29-16 不带预分频的下计数

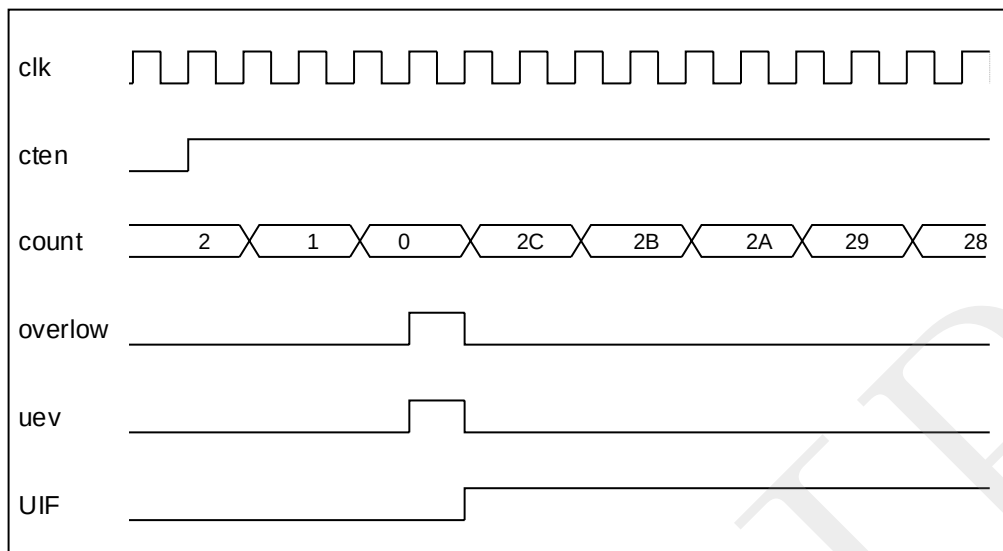


图 29-17 带预分频的下计数

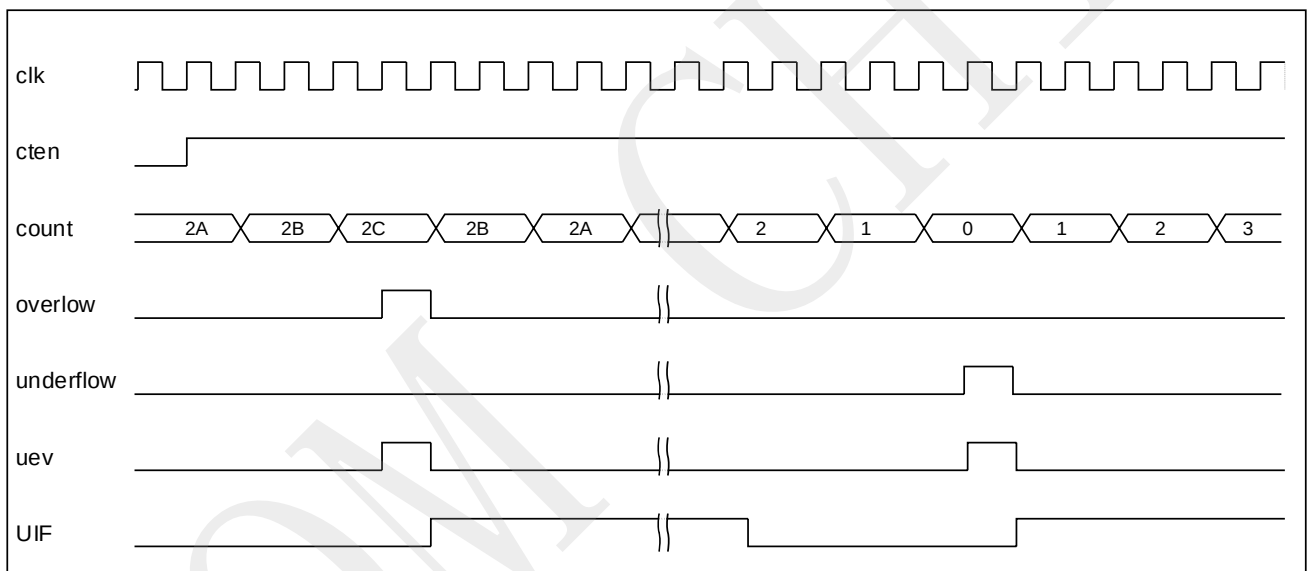


图 29-18 带预分频的上下计数

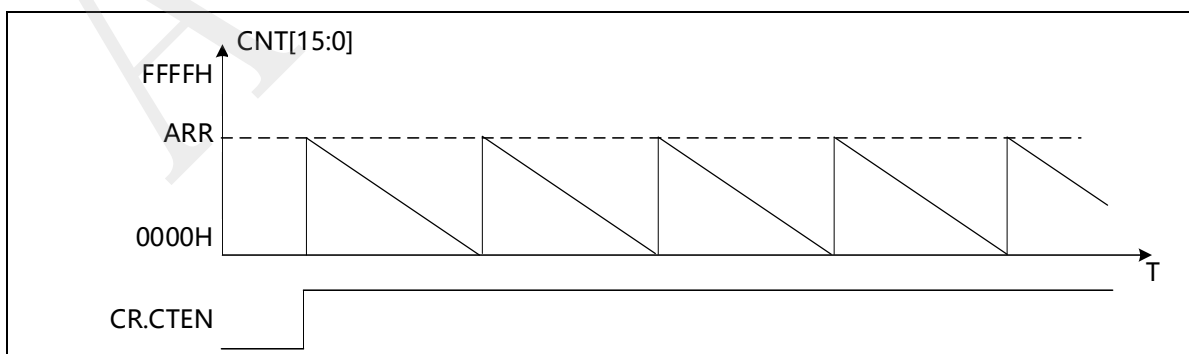


图 29-19 边沿对齐计数器波形(DIR =1)

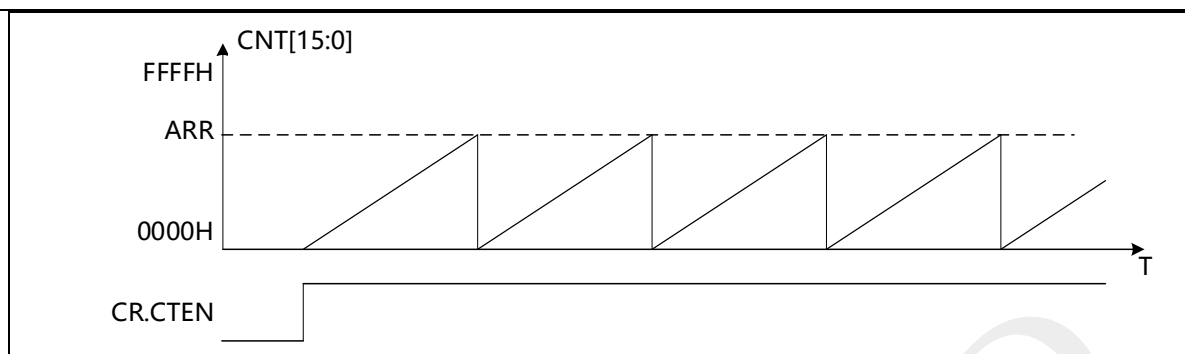


图 29-20 边沿对齐计时器波形(DIR =0)

模式 3 为三角波计数波形，计数方向控制位只读，不可以更改计数方向。

中心对齐（三角波）模式下 CR.DIR 方向位是只读的。写值无效。从其他模式切换到中心对齐模式 DIR 自动清 0。软件事件更新和从模式外部触发复位模式 DIR 自动清零。

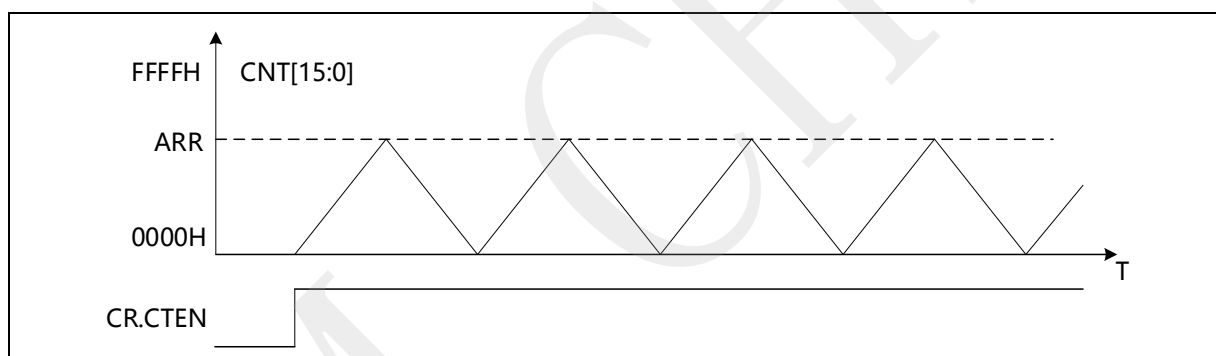


图 29-21 中心对齐计数器波形

29.2.6.3 重复计数

重复计数器使用计数器的溢出进行向下计数。计数到 0 时，即计数器发生重复寄存器设置的值加一次溢出时。当缓存寄存器使能时，周期重载寄存器更新到周期缓存寄存器。比较模式下比较寄存的值更新到比较缓存寄存中。

重复计数器在下面条件成立时递减

- 上计数模式下每次计数器溢出时
- 下计数模式下每次计数器下溢时
- 三角波模式下每次上溢出和每次下溢出时

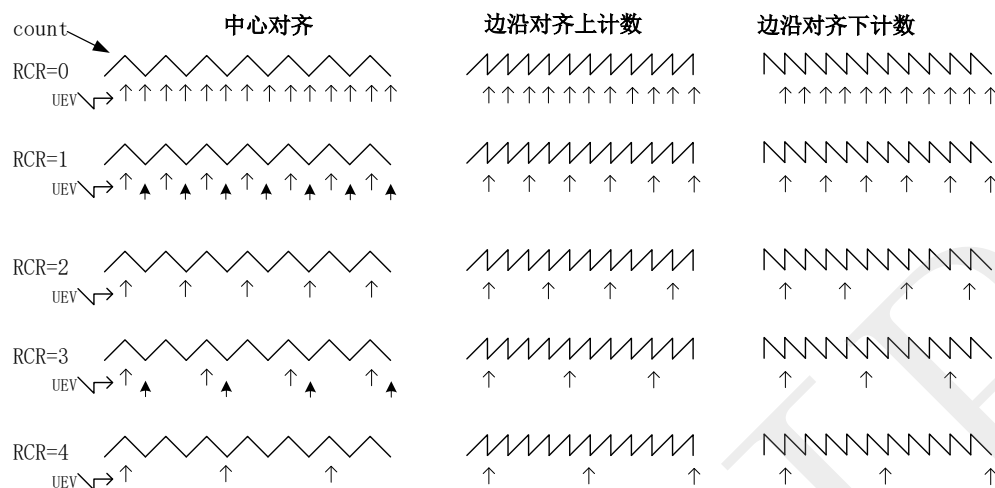


图 29-22 重复计数器产生更新时序

除了上下溢出通过重复计数器可以产生事件更新 UEV 外，还可以通过写寄存器 CR.UG 产生软件及从模式复位事件更新 UEV；这时需要配置 CR.URS。

29.2.6.4 数据缓存

自动重载数据 ARR 与比较寄存器都可以配置缓存功能，当缓存功能有效时，当发生 UEV 事件更新时，写入的周期值 ARR 与比较值 CCR 才会生效。

自动重载值在不同计数模式下的更新时序图如下

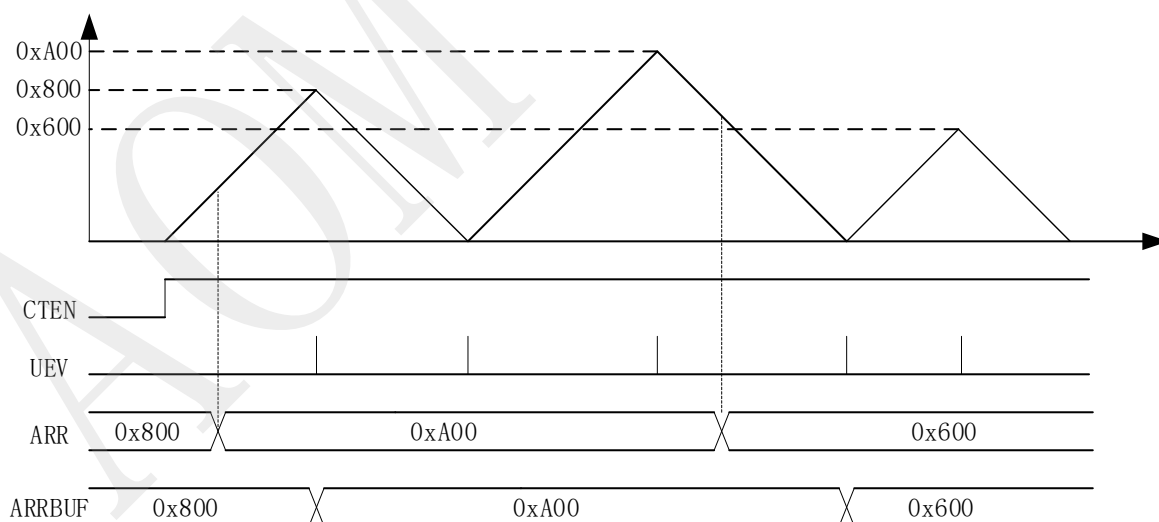


图 29-23 三角波模式下缓存使能

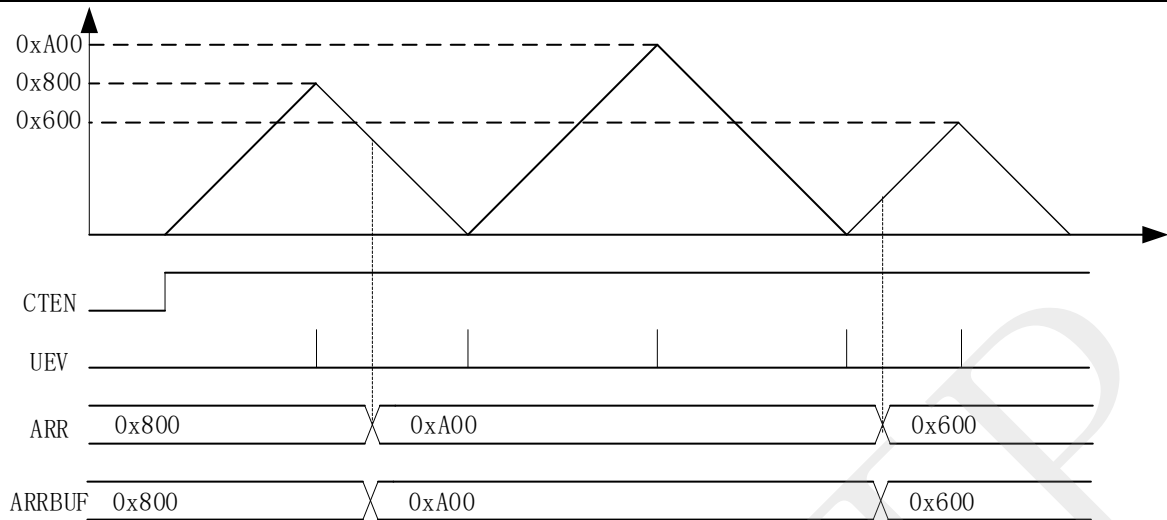


图 29-24 三角波模式下缓存无效

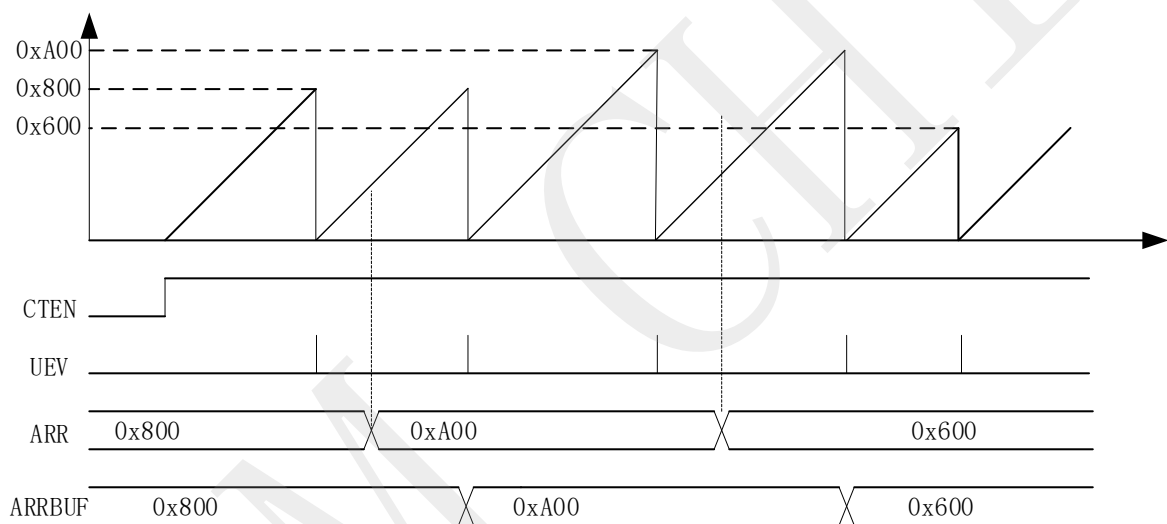


图 29-25 锯齿波模式下上计数缓存使能

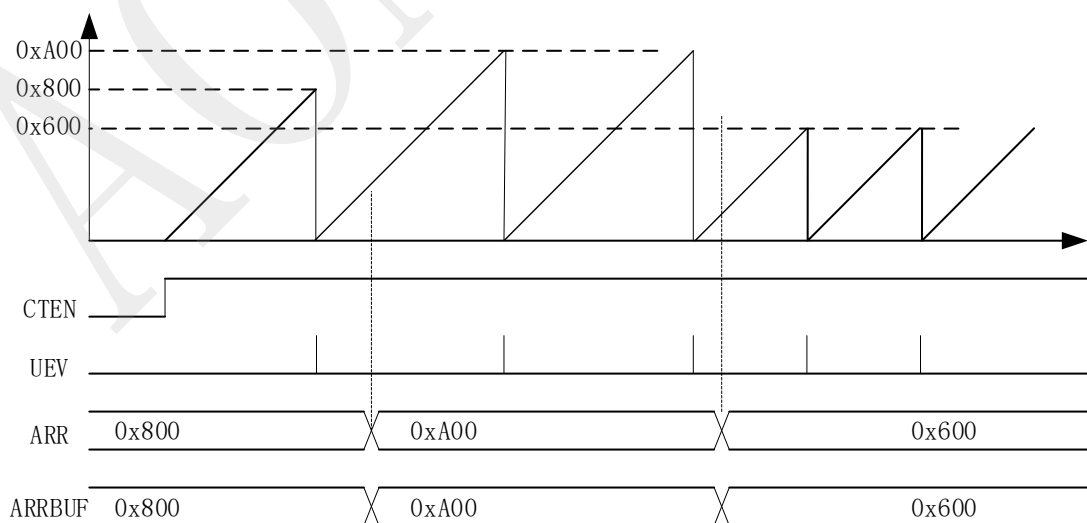


图 29-26 锯齿波模式下上计数缓存无效

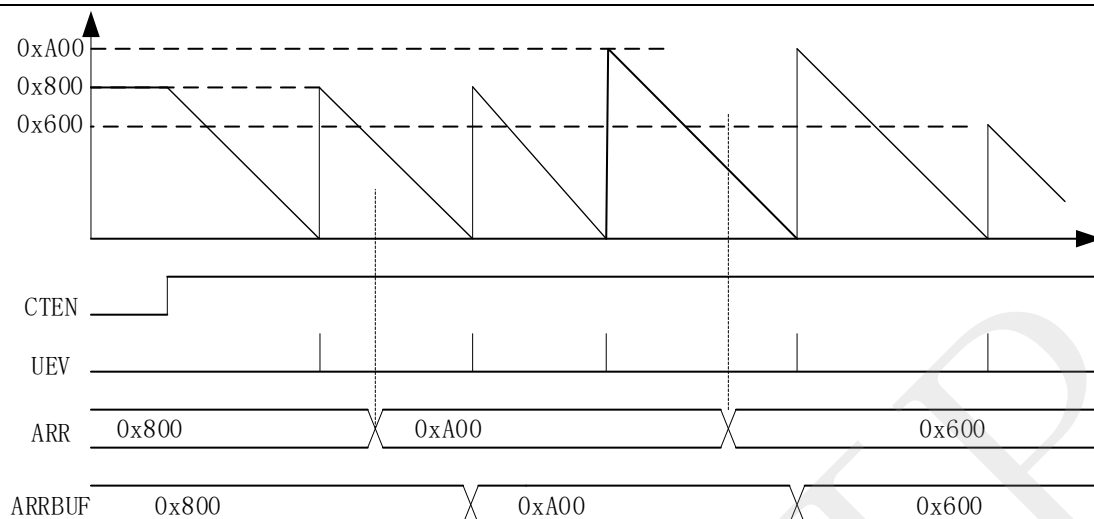


图 29-27 锯齿波模式下计数缓存使能

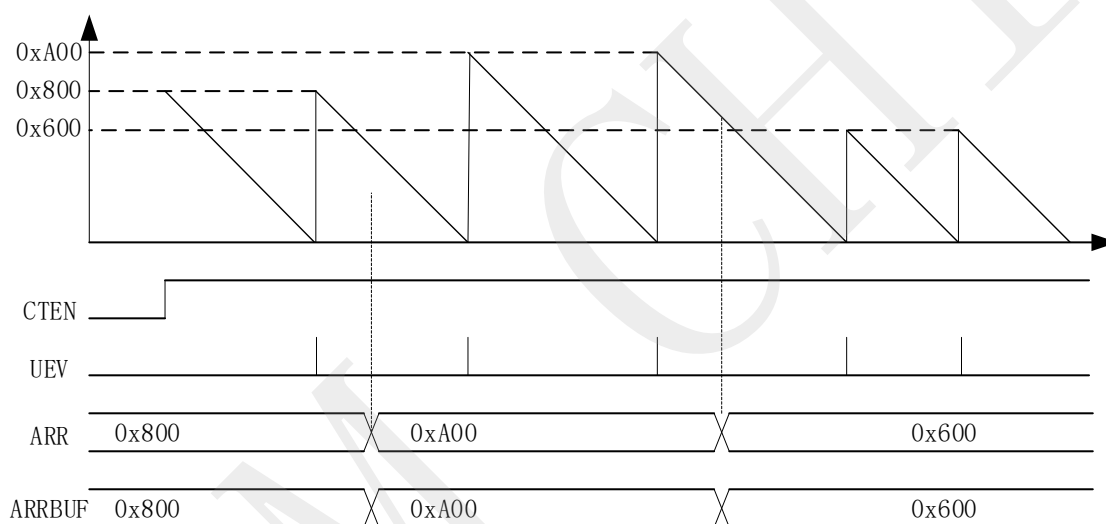


图 29-28 锯齿波模式下计数缓存无效

在三角波模式与锯齿波上计数模式时，如果缓存不使能，更改的 ARR 时，当前计数器的值要小于要更改的 ARR 周期值，否则当前周期会计数到 0xFFFF。

比较缓存与周期缓存更新状态一致，这里不一一列出时序图。

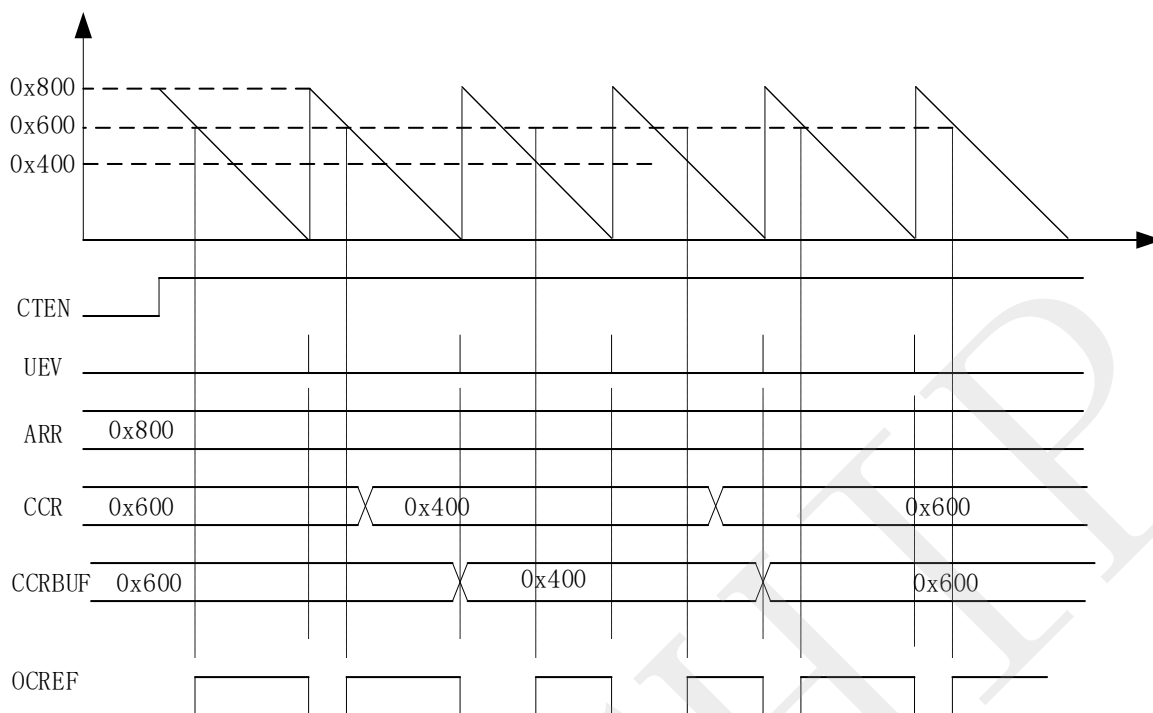


图 29-29 锯齿波模式下计数比较缓存使能

29.2.6.5 比较输出 OCREF

比较输出 OCREFA 可以配置为单点比较，使用比较寄存器 CCRA 控制 OCREFA 的输出；OCREFA 的比较输出也可以配置为双点比较，使用比较寄存器 CCRA,CCRB 一起控制 OCREFA 的比较输出。

OCREFB 的比较输出只能使用单点比较，使用比较寄存器 CCRB 控制 OCREFB 的比较输出。

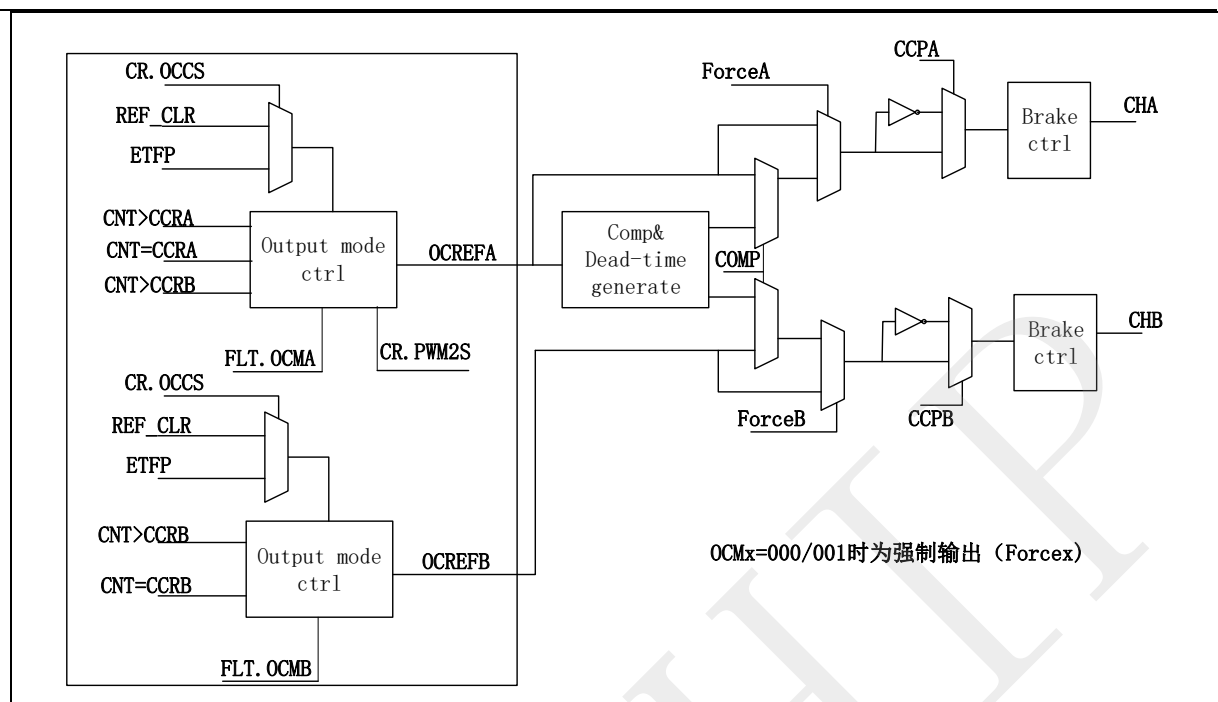


图 29-30 OCREF 输出框图

OCREF 输出使用 OCMx 选择

000: 强制为 0

001: 强制为 1

010: 比较匹配时强制为 0

011: 比较匹配时强制为 1

100: 比较匹配时翻转

101: 比较匹配时输出一个计数周期的高电平

110: PWM 模式 1

单点比较:

上计数时 $CNT < CCR_{xy}$ 输出高, 下计数时 $CNT > CCR_{xy}$ 输出为低电平

双点比较:

1) 锯齿波上计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为低电平

2) 锯齿波下计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为高电平

3) 三角波上计数 $CNT < CCR_{xA}$ 输出高, 下计数 $CNT > CCR_{xB}$ 为低电平

111: PWM 模式 2

单点比较:

上计数时 $CNT < CCR_{xy}$ 输出低，下计数时 $CNT > CCR_{xy}$ 输出为高电平
双点比较：

- 1) 锯齿波上计数 $CCR_{xA} \leq CNT < CCR_{xB}$ 输出为高电平
- 2) 锯齿波下计数 $CCR_{xA} \leq CNT < CCR_{xB}$ 输出为低电平
- 3) 三角波上计数 $CNT < CCR_{xA}$ 输出低，下计数 $CNT > CCR_{xB}$ 为高电平

注：强制输出有高优先级，当强制输出有效时，互补输出控制无效。

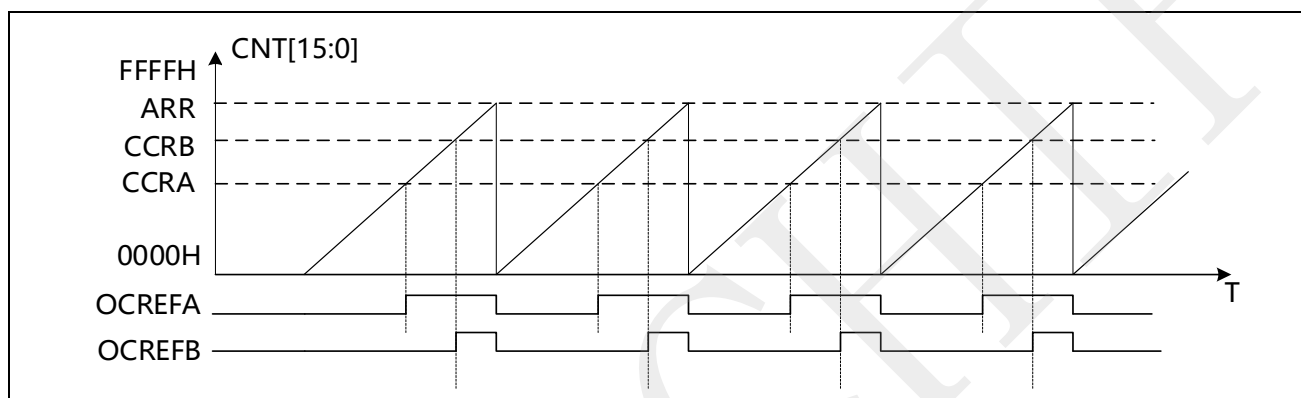


图 29-31 锯齿波计数单点比较 OCREF 输出波形 (OCM_x=111)

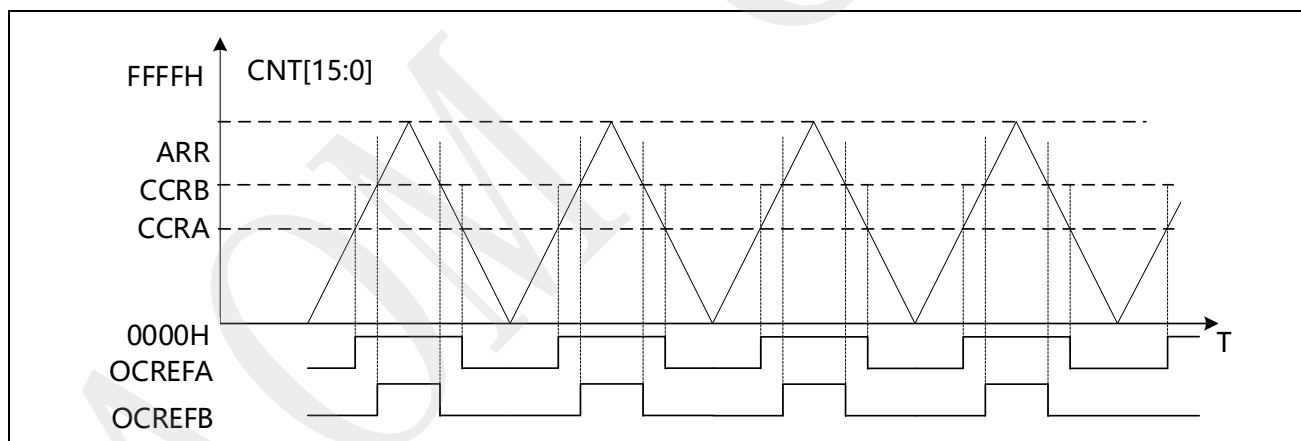


图 29-32 三角波计数单点比较 OCREF 输出波形 (OCM_x=111)

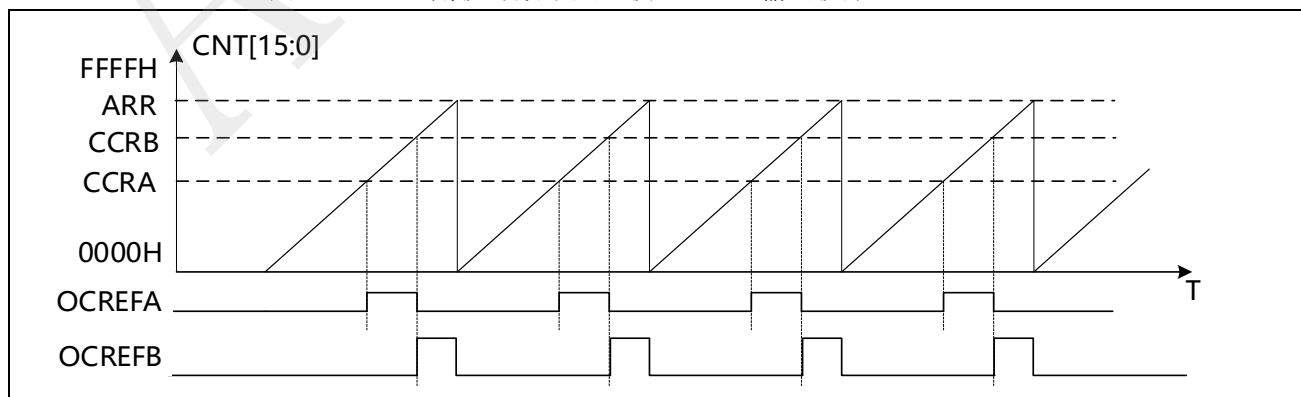


图 29-33 锯齿波计数双点比较 OCREF 输出 (OCMx=111)

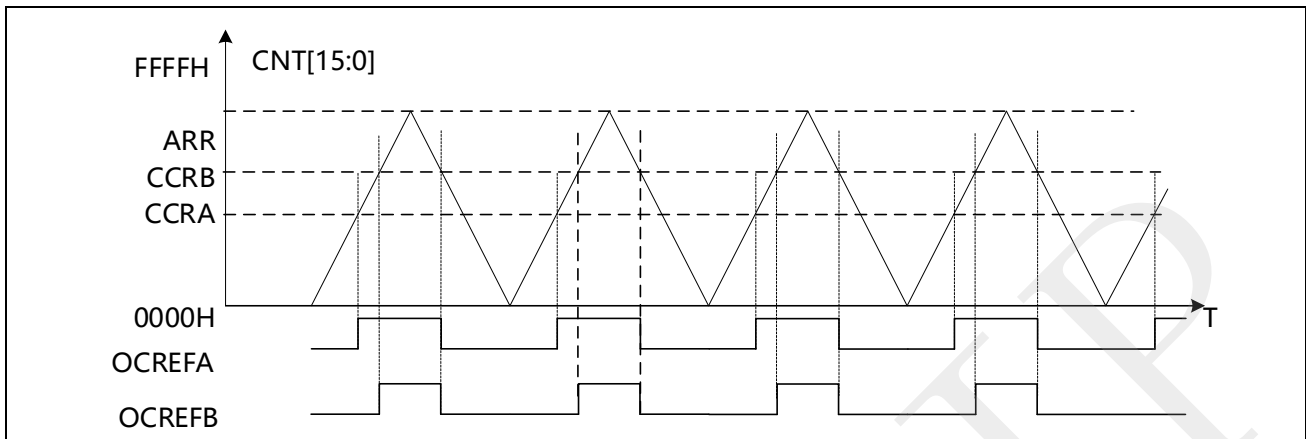


图 29-34 三角波计数双点比较 OCREF 输出 (OCMx=111)

29.2.6.6 独立 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFB 控制 CHB 的输出。通过 CRCHx.CCPA, CRCHx.CCPB 可以控制 CHA, CHB 输出的反向。

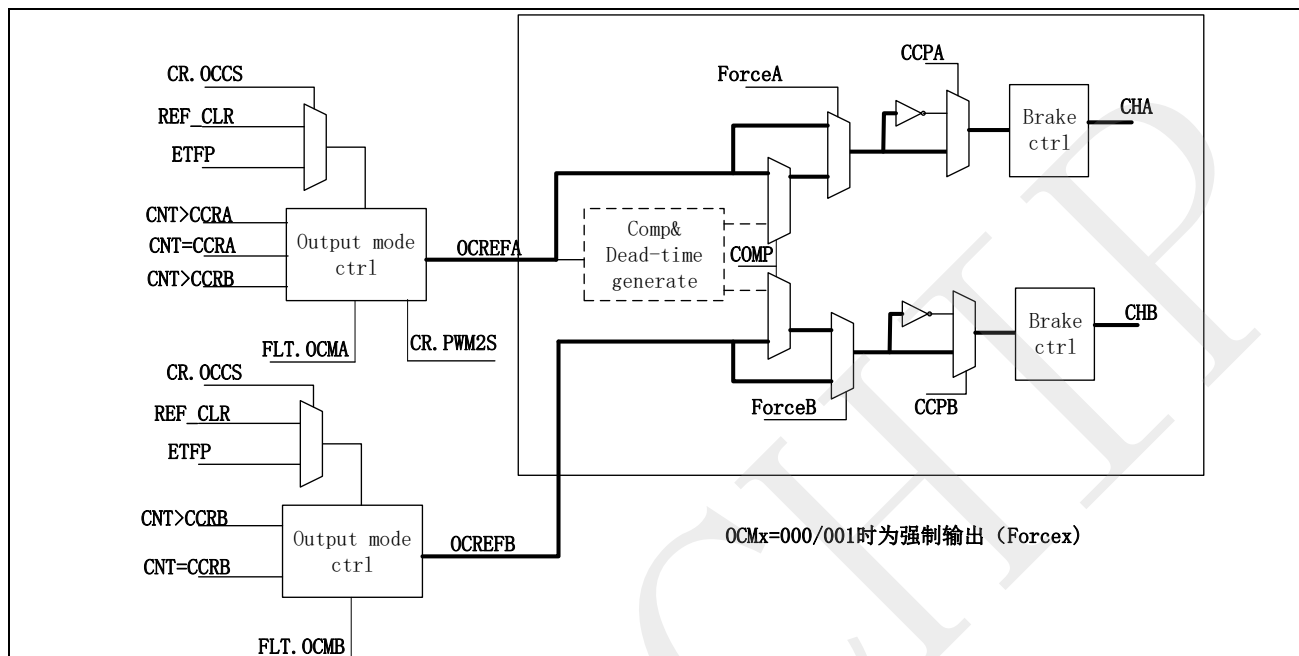


图 29-35 独立 PWM 输出框图

PWM 输出与 OCREF 关系

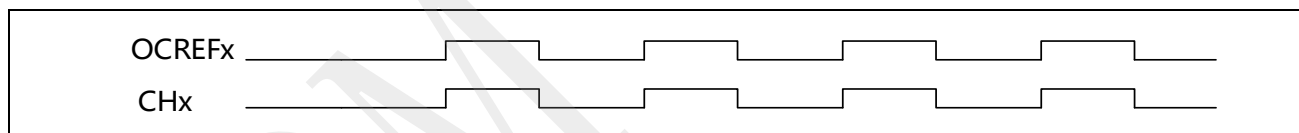


图 29-36 CCPx=0时PWM输出波形

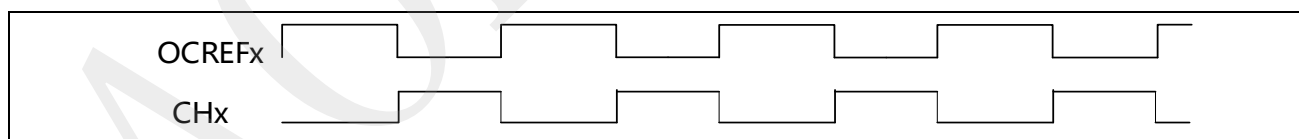


图 29-37 CCPx=1 时 PWM 输出波形

29.2.6.7 互补 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFA 同时控制 CHB 的输出。比较寄存器 CCRxB 可以作为专用比较控制 ADC 触发。

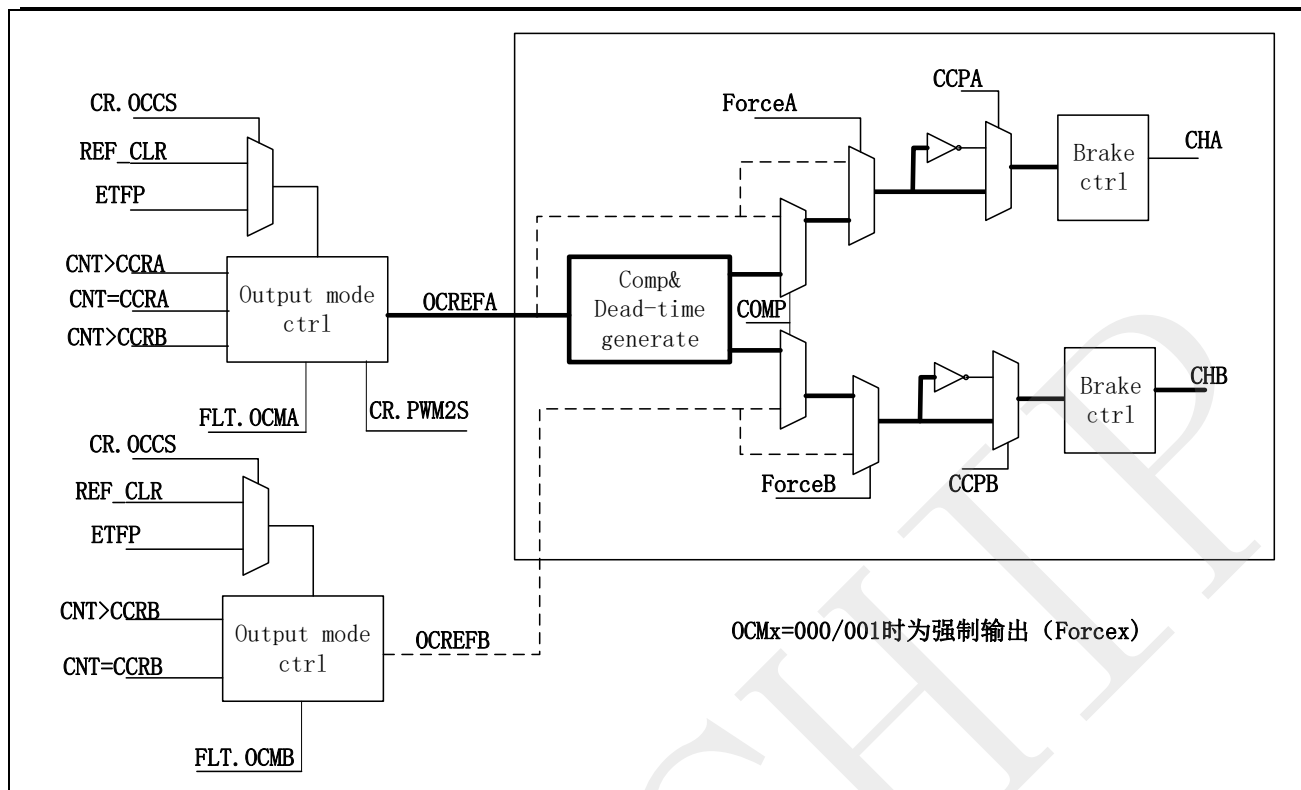


图 29-38 互补 PWM 输出框图

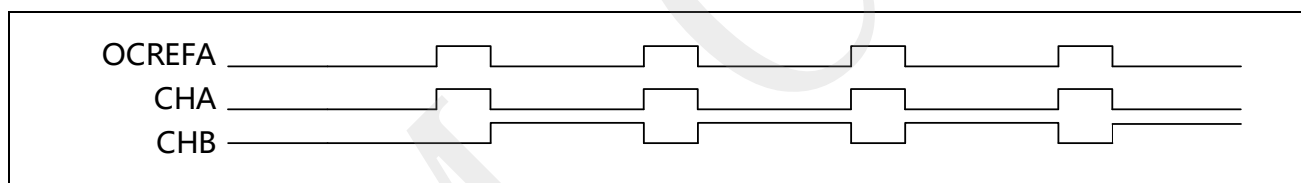


图 29-39 互补PWM输出波形图

29.2.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

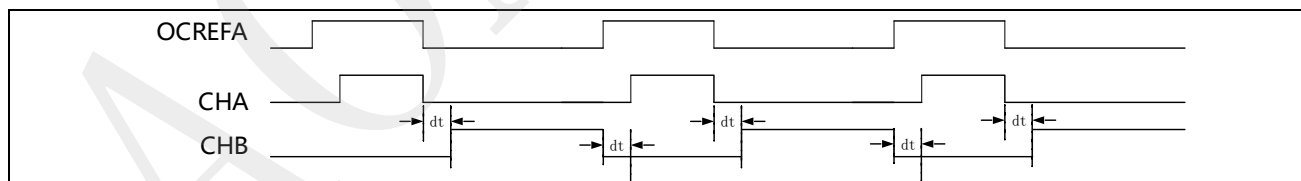
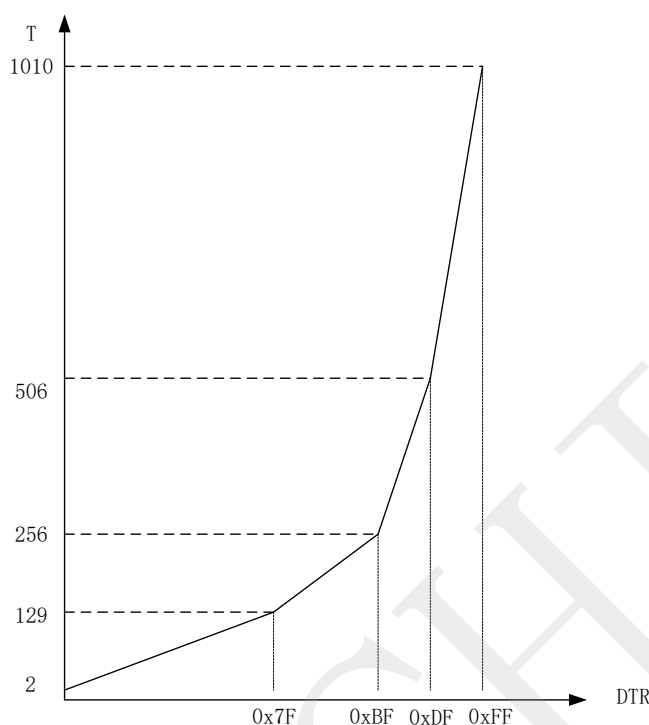


图 29-40 互补PWM输出波形图

死区时间使用 8 位 DTR 控制，死区时间 dt 与 DTR 的关系如下

DTR[7] = 0	$T = DTR[6:0] + 2$	2-129	step=1
DTR[7:6] = 10	$T = \{DTR[5:0] + 64\} * 2 + 2$	130-256	step=2
DTR[7:5] = 110	$T = \{DTR[4:0] + 32\} * 8 + 2$	258-506	step=8
DTR[7:5] = 111	$T = \{DTR[4:0] + 32\} * 16 + 2$	514-1010	step=16



100M clock 死区时间由2.56us调整为10.1us

图 29-41 死区时间

29.2.6.9 单脉冲输出

单脉冲模式 (ONE SHOT) 是上述 PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 GTIMx_M23CR 寄存器中的 ONESHOT 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

单脉冲模式在锯齿波下计数模式计数器初值不要设置为 0，上计数模式计数值不要设置到大于等于 ARR。

只有当比较值与计数器初始值不同时，才能正确产生输出脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数时：CNT < CCRxy ≤ ARR（特别注意，0 < CCRxy），
- 递减计数时：CNT > CCRxy。

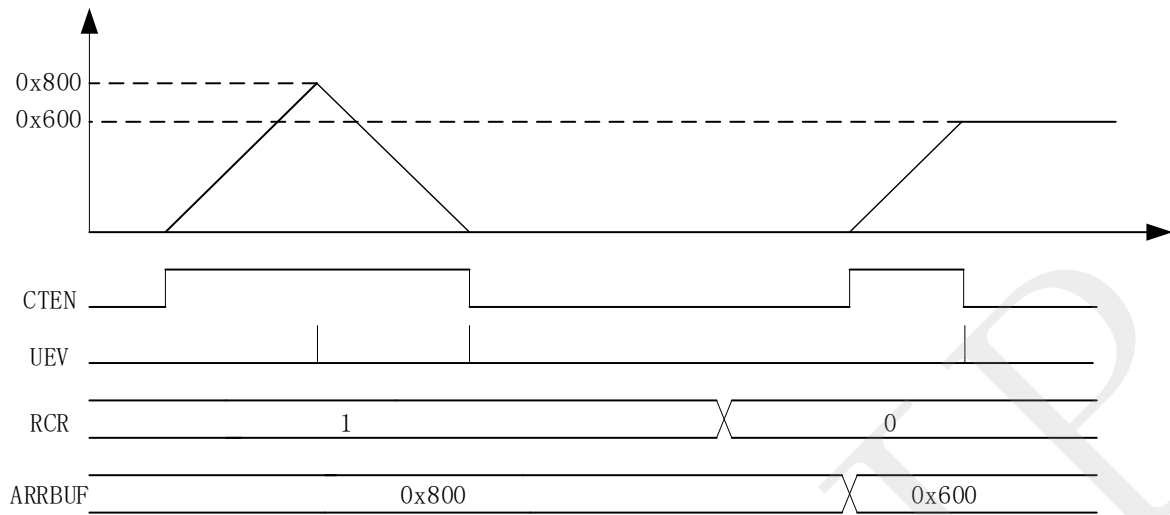


图 29-42 三角波模式单脉冲计数

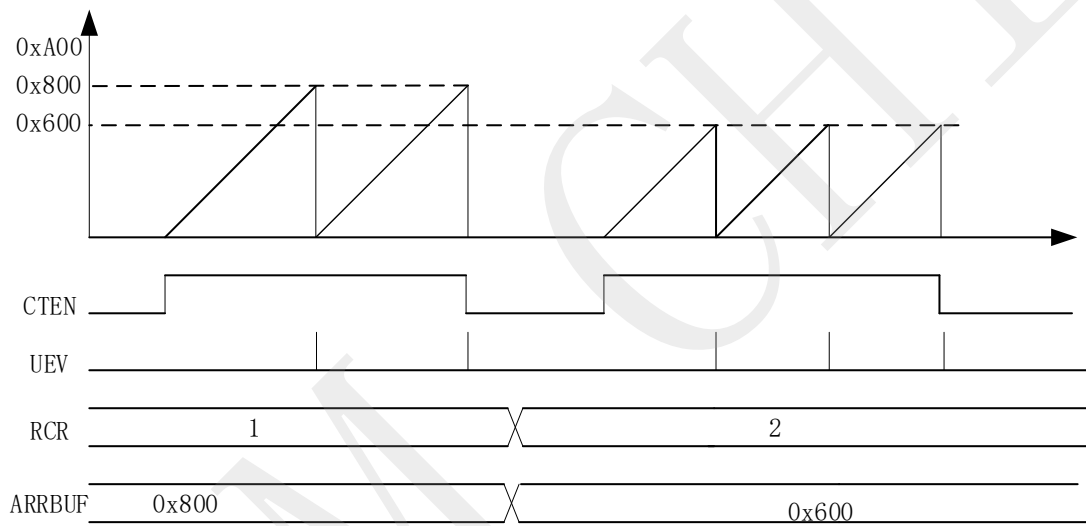


图 29-43 锯齿波上计数单脉冲模式

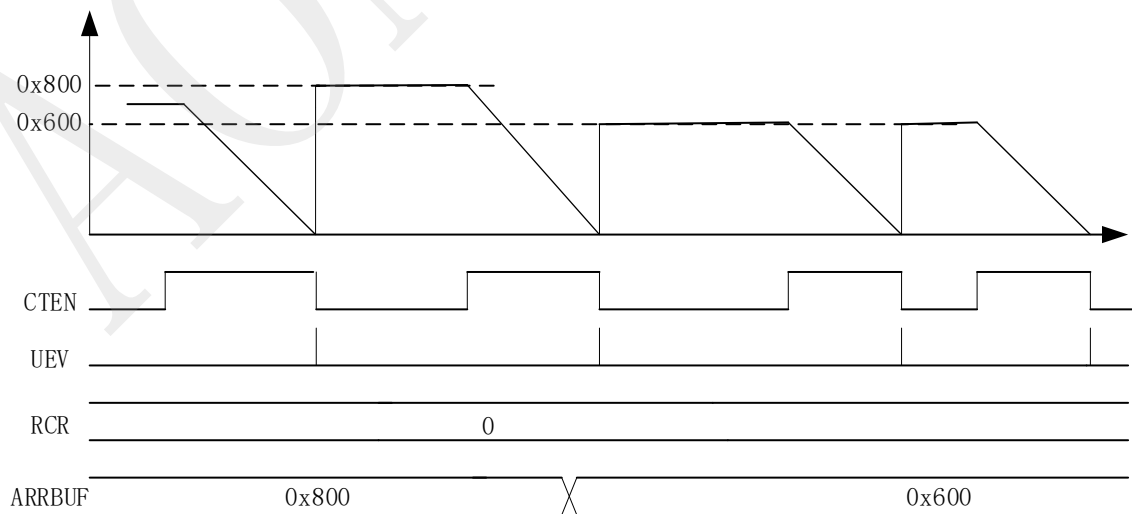


图 29-44 锯齿波下计数单脉冲模式

29.2.6.10 比较中断

锯齿波比较匹配会将 IFR 寄存器相应的标志置 1，如果中断使能 CRCHx.CIEy (x=0,1,2;y=A,B)将会触发中断。

三角波比较匹配可单独选择上升计数比较匹配，下降计数比较匹配或者两者比较都匹配。

比较 A 比较匹配当计数值与 CCRxA 相等时，统一使用 CR.CIS 控制。

CR.CIS=2'b00 时 比较匹配无输出

CR.CIS=2'b01 时 上计数时比较匹配

CR.CIS=2'b10 时 下计数时比较匹配

CR.CIS=2'b11 时 上下计数时比较都匹配

比较 B 比较匹配当计数值与 CCRxB 相等时，不同通道可以使用 CRCHx.CISB 单独控制。

CRCHx.CISB=2'b00 时 通道 x 比较匹配无输出

CRCHx.CISB=2'b01 时 通道 x 上计数时比较匹配

CRCHx.CISB=2'b10 时 通道 x 下计数时比较匹配

CRCHx.CISB=2'b11 时 通道 x 上下计数时比较都匹配

B 通道比较匹配单独控制为了更灵活的触发 ADC。详见 Timer 触发 ADC 章节。

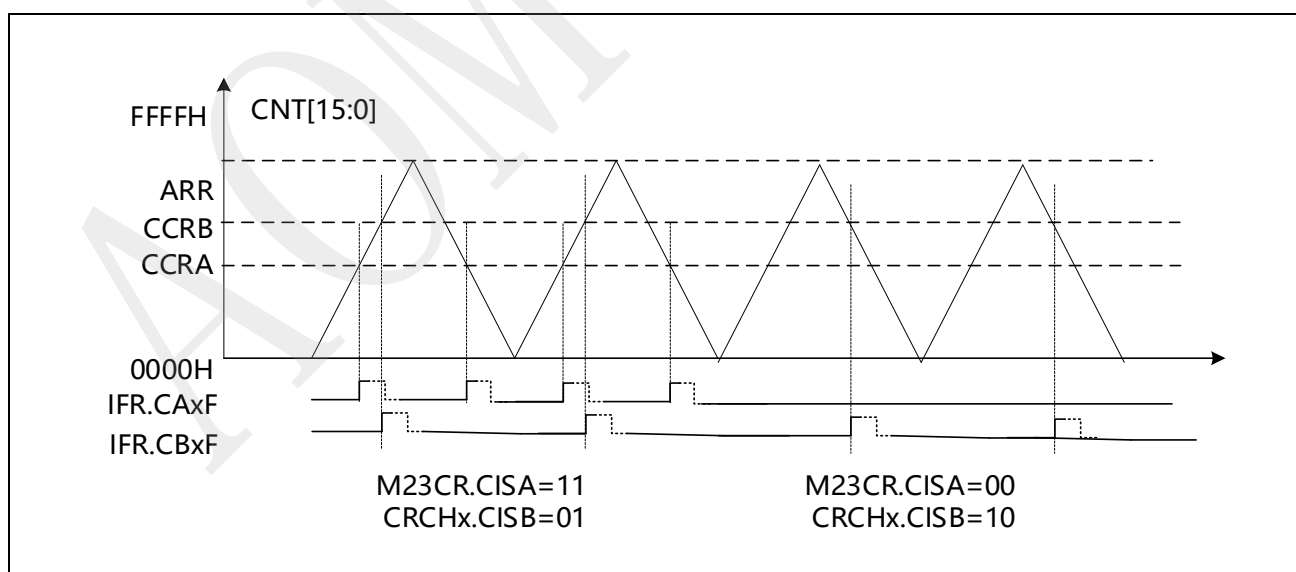


图 29-45 中断示意图

29.2.6.11 捕获输入

CR.MODE=2/3

在三角波计数或者锯齿波计数模式下都可以设置捕获功能，可以设置捕获的电平边沿，当发生捕获时捕获的值存入比较捕获寄存器并产出捕获中断。

每个通道的比较捕获功能可以单独设置，通过寄存器 CRCHx.CSA/CSB 选择。

每个通道的捕获边沿可以单独设置，通过寄存器 CRCH0.CFy/CRy;(y=A/B)选择捕获触发的边沿。

当捕获发生后捕获标志未清除前再次发生捕获动作，会产生捕获数据覆盖标志。

定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

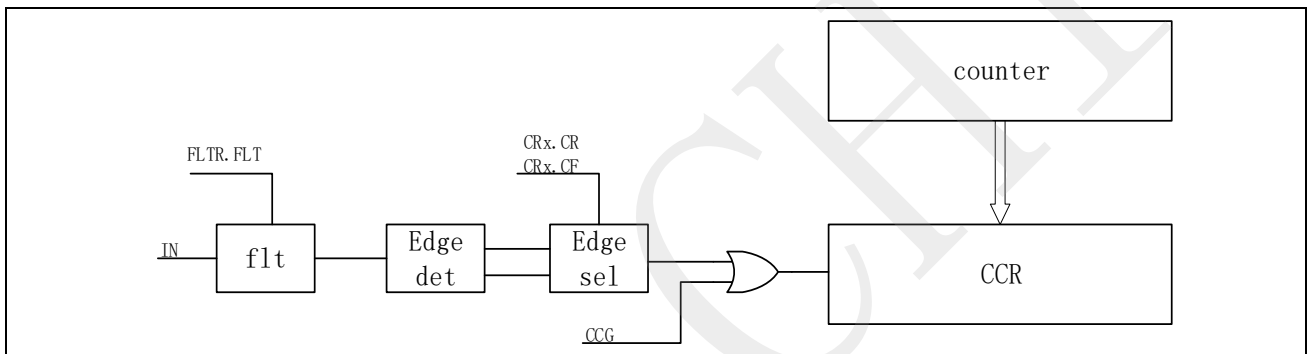


图 29-46 捕获功能框图

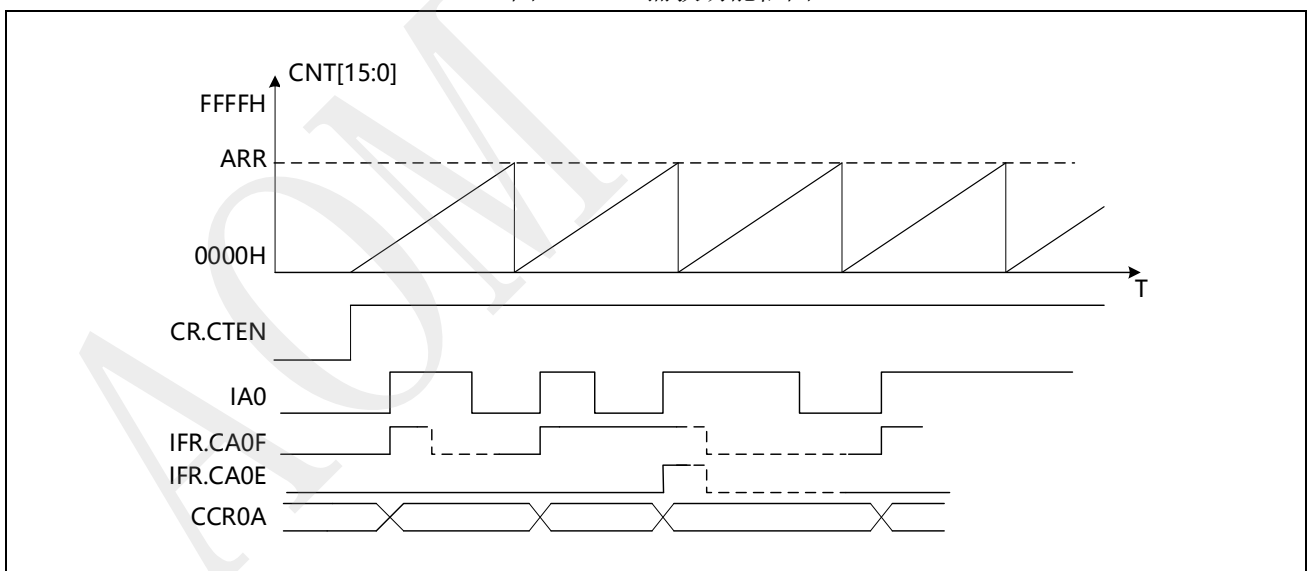


图 29-47 捕获时序图

CH0A 的捕获输入通过 MSCR.IA0S 选择 CH0A 输入还是 CH0A, ETR, GATE 的异或输入。

IA0S	0	1
GTIM	CHA0	CHA0 ETR GATE异或输入

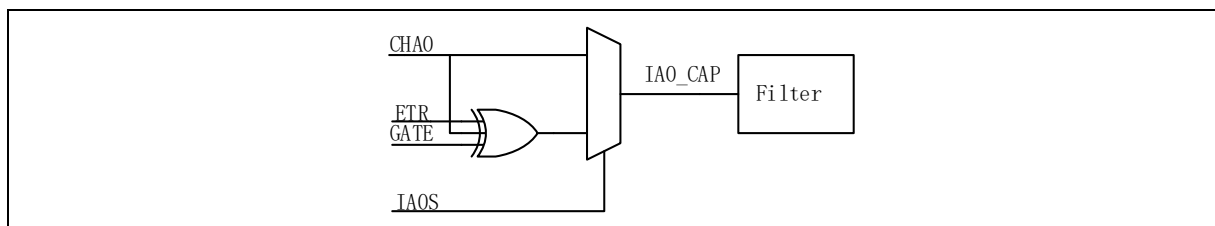


图 29-48 CHA 端口选择

CH0B 的捕获输入通过 MSCR. IB0S 选择 CH0B 输入还是内容 MSCR.TS 选择的信号。

IB0S	0	1																		
GTIM1/2/3	CHB0	内部触发MSCR.TS选择信号																		
		<table><tr><th>MSCR.TS</th><th>通道B捕获输入</th></tr><tr><td>000</td><td>ETR 滤波相位选择输出信号，滤波相位选择可配</td></tr><tr><td>001</td><td>内部互联ITR0</td></tr><tr><td>010</td><td>内部互联ITR1</td></tr><tr><td>011</td><td>内部互联ITR2</td></tr><tr><td>100</td><td>内部互联ITR3</td></tr><tr><td>101</td><td>CH0A 边沿</td></tr><tr><td>110</td><td>CH0A 滤波输出信号，滤波功能可配</td></tr><tr><td>111</td><td>CH0B 滤波输出信号，滤波功能可配</td></tr></table>	MSCR.TS	通道B捕获输入	000	ETR 滤波相位选择输出信号，滤波相位选择可配	001	内部互联ITR0	010	内部互联ITR1	011	内部互联ITR2	100	内部互联ITR3	101	CH0A 边沿	110	CH0A 滤波输出信号，滤波功能可配	111	CH0B 滤波输出信号，滤波功能可配
		MSCR.TS	通道B捕获输入																	
		000	ETR 滤波相位选择输出信号，滤波相位选择可配																	
		001	内部互联ITR0																	
		010	内部互联ITR1																	
		011	内部互联ITR2																	
		100	内部互联ITR3																	
		101	CH0A 边沿																	
		110	CH0A 滤波输出信号，滤波功能可配																	
111	CH0B 滤波输出信号，滤波功能可配																			

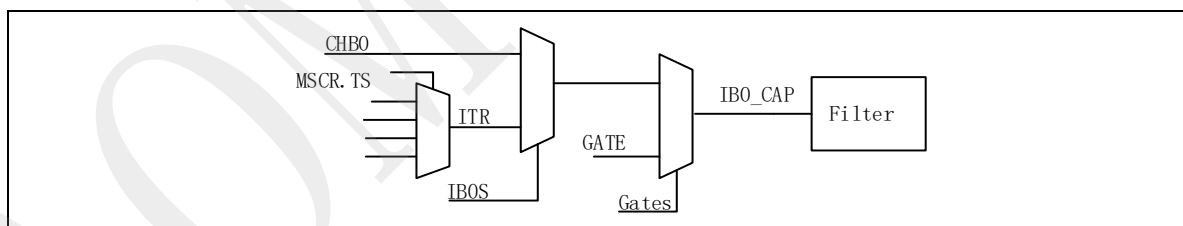


图 29-49 CHB 端口选择

Gates 信号为选择 PWM 互补输出功能生成的信号，当 PWM 互补输出时，比较捕获寄存器 CCR0B 没有使用，自动切换到使用 GATE 作为捕获/比较输出。

29.2.6.12 设置示例

边沿对齐独立 PWM 输出设置

1. 设置模式 CR.MODE=2

2. 设置锯齿波计数方向 CR.DIR
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA,CCRxB
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 CR.CTEN

中心对齐互补 PWM 输出设置

1. 设置模式 CR.MODE=3
2. 设置互补输出 CR.COMP
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA(CCRxB 不需要设置，PWM 输出与这个寄存器无关)
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 CR.CTEN

三角波非中心对齐带死区互补 PWM 输出设置

1. 设置模式 CR.MODE=3
2. 设置互补输出 CR.COMP
3. 设置两个比较使能 CR.PWM2S

4. 设置 PWM 周期值 ARR
5. 设置计数器初值（初值必须小于周期值）
6. 设置 PWM 比较值 CCRxA,CCRxB,上计数比较点为 CCRxA,下计数比较点为 CCRxB
7. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
8. 清除相关中断标志
9. 使能相应的中断
10. 设置输出极性 FLT.CCPAx FLT.CCPBx
11. 设置死区使能 DTR.DTEN
12. 设置死区时间 DTR.DTR
13. 使能输出 DTR.MOE
14. 使能定时器 CR.CTEN

捕获功能设置

1. 设置 CH0B 为上升沿捕获功能
2. 选择计数器计数方式，设置 mode=2
3. 选择 CH0B 为捕获模式 CRCHx.CSB=1
4. 根据需要进行输入滤波 FLT.FLTB0
5. 选择捕获的边沿 CRCHx.CRB=1
6. 设置 ARR 改变周期值
7. 启动定时器 CR.CTEN
8. 清除相关中断标志
9. 清除捕获标志，使能相应中断
10. 查询到捕获标志后，读取 CCR0B 获得捕获值

注：定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

互补 PWM 输出时使用 GATE 作为 PWM 输出功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCR0B 设置 GATE PWM 比较值

3. 设置 FLT.OCMB0=6/7 设置 PWM 输出模式
4. 使能 PWM 输出 DTR.MOE

互补 PWM 输出时使用 GATE 下降沿作为捕获输入功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 GATE 作为捕获输入 CR.CSG
3. 选择 GATE 捕获输入边沿设置 CR.CFG(GATE 作为捕获输入 B 通道滤波设置无效)
4. 启动定时器 CR.CTEN
5. 清除相关中断标志
6. 清除捕获标志，使能相应中断
7. 查询到捕获标志后，读取 CCR0B 获得捕获值

29.2.7 模式 2/3 从模式

定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

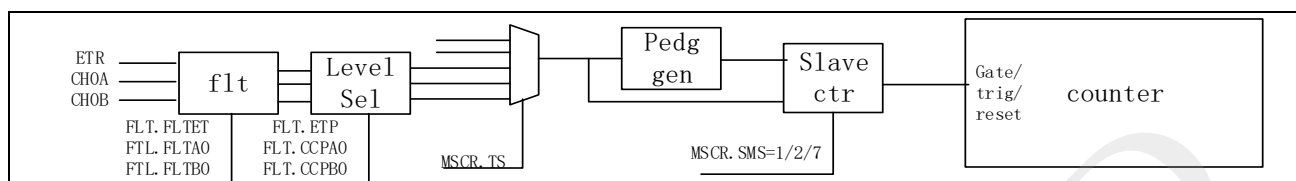


图 29-50 从模式示意图

从模式功能选择通过 MSCR.SMS 选择；

MSCR.SMS	从模式功能选择 001: 复位功能; 010: 触发模式; 111: 门控功能
MSCR.TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP, 滤波功能可配置 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 端口CH0A的边沿信号; 110: 端口CH0A的滤波相位选择后的信号IAFP, 滤波功能可配置 111: 端口CH0B的滤波相位选择后的信号IBFP, 滤波功能可配置

29.2.7.1 门控计数

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 CH0A 为低电平时向上计数：

- 配置通道 CH0A 以检测 CH0A 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 FLT.FLTA0=000)。置 FLTR 寄存器中 CCPA0=1 以确定极性(只检测低电平)。
- 置 SMCR 寄存器中 SMS=111, 配置定时器为门控模式; 置 SMCR 寄存器中 TS=110, 选择 CH0A 作为输入源。
- 置 CR 寄存器中 CTEN=1, 启动计数器。在门控模式下，如果 CTEN=0, 则计数器不能启动，不论触发输入电平如何。只要 CH0A 为低，计数器开始依据内部时钟计数，一旦 CH0A 变高则停止计数。

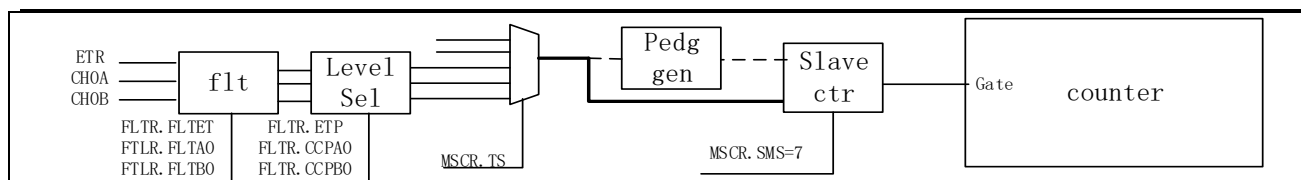


图 29-51 门控功能

29.2.7.2 触发功能

使用外部触发（CH0A、CH0B、ETR）可以使定时器同步启动，使用定时器内部互联信号结合 MSCR.MSM 也可以配置定时器同步启动。触发信号为输入信号的上升沿。也可以使用软件写 CR.TG 启动软件触发功能。

输入端上选中的事件使能计数器。在下面的例子中，计数器在 CH0B 输入的上升沿开始向上计数：

- 配置通道 CH0B 检测 CH0B 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 FLT.FLTB0=000)。设 FLTR 寄存器中 CCPB0=0 以确定极性(不反向)。
- 置 SMCR 寄存器中 SMS=010，配置定时器为触发模式；置 SMCR 寄存器中 TS=111，选择 CH0B 作为输入源。当 CH0B 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。CH0B 上升沿和计数器启动计数之间的延时，取决于 CH0B 输入端的重同步电路。

注意：如果使用下降沿触发，先选择触发极性，然后再选择模式，否则会产生误触发。

29.2.7.3 复位计数

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

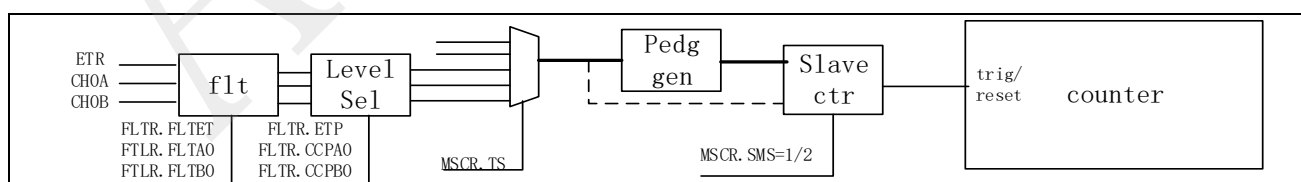


图 29-52 触发和复位功能

29.2.8 正交编码计数功能

MSCR.SMS=4/5/6，对应正交编码模式的模式 1/2/3。这时计数器根据 IAFP, IBFP 的

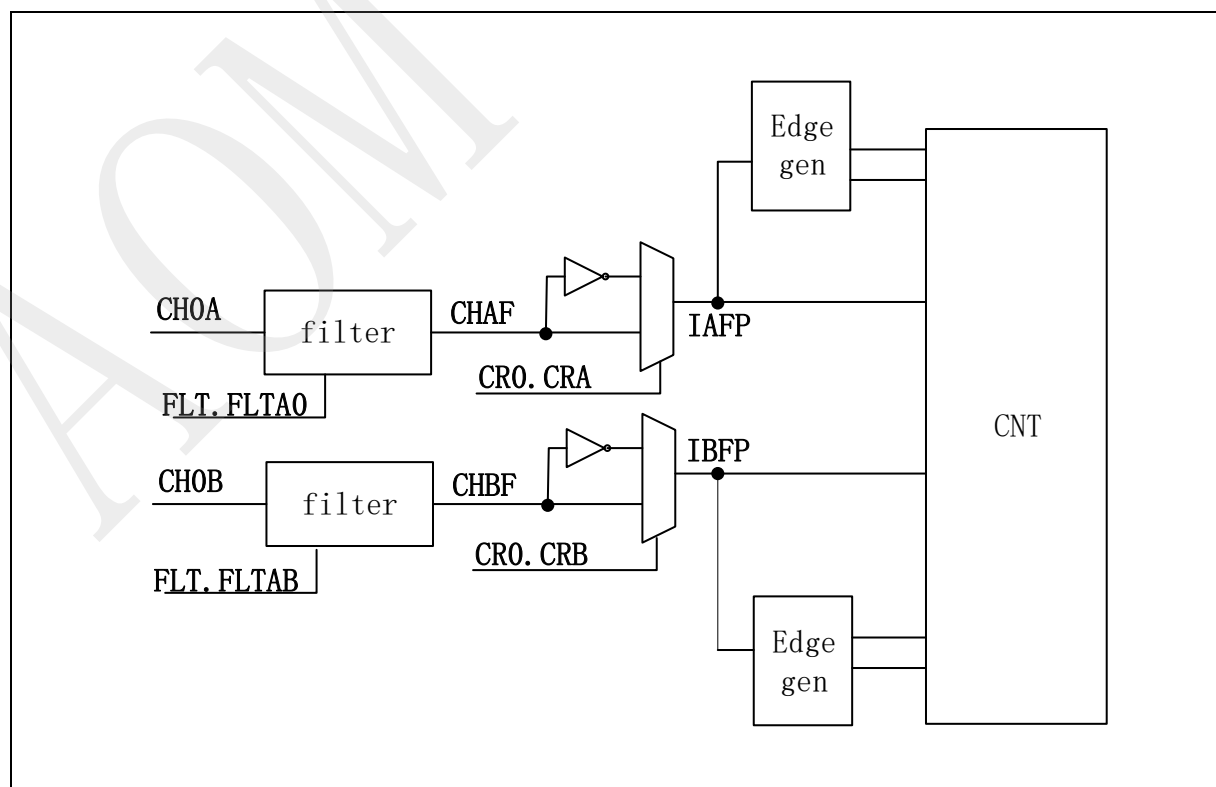
相位关系进行编码计数。IAFP,IBFP 为端口输入 CH0A, CH0B 的滤波相位选择的信号。

模式1使用CH0A的边沿计数。模式2使用CH0B的边沿计数。模式3使用CH0A,CH0B的边沿都计数。

为了保证计数相位的正确性,需要保证 A/B 输入的相位大于一个脉冲宽度的相位差, A/B 输入脉冲宽度需要大于两个脉冲宽度。

			IAFP		IBFP	
	IBFP	IAFP	Rising	falling	Rising	falling
MOD1	High		Down	Up	–	–
	Low		up	Down	–	–
MOD2		High	–	–	up	Down
		Low	–	–	Down	Up
MOD3	High	High	Down	Up	up	Down
	Low	Low	up	Down	Down	Up

CHAF/CHBF 为端口 CH0A/CH0B 滤波的信号, IAFP/IBFP 为端口滤波相位选择后的信号。



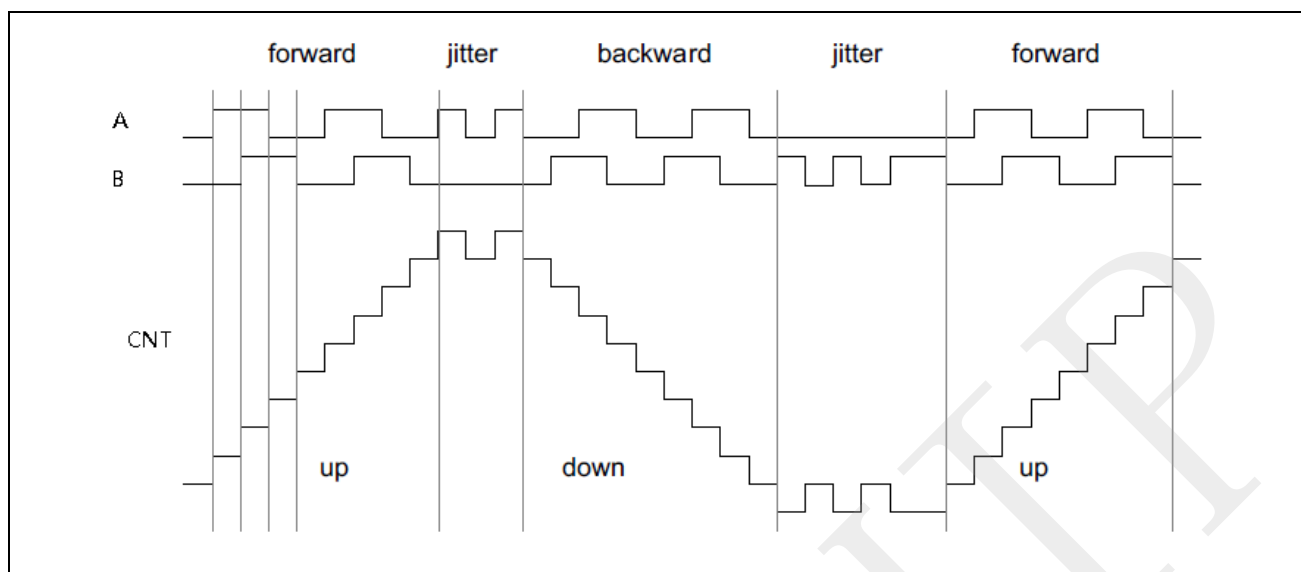


图 29-53 编码计数

29.2.9 Timer 触发 ADC

CCMR0A,CCMR1A,CCMR2A 比较匹配可以配置触发 ADC，中心对齐 PWM 时只能通过控制寄存器 CR.CIS 可以选择在上升匹配触发或者下降匹配触发。

CCMR0B,CCMR1B,CCMR2B 比较匹配可以配置触发 ADC，中心对齐 PWM 时，可以通过寄存器 CRx.CISB 分别控制三个匹配触发点（上升，下降，上升下降）。

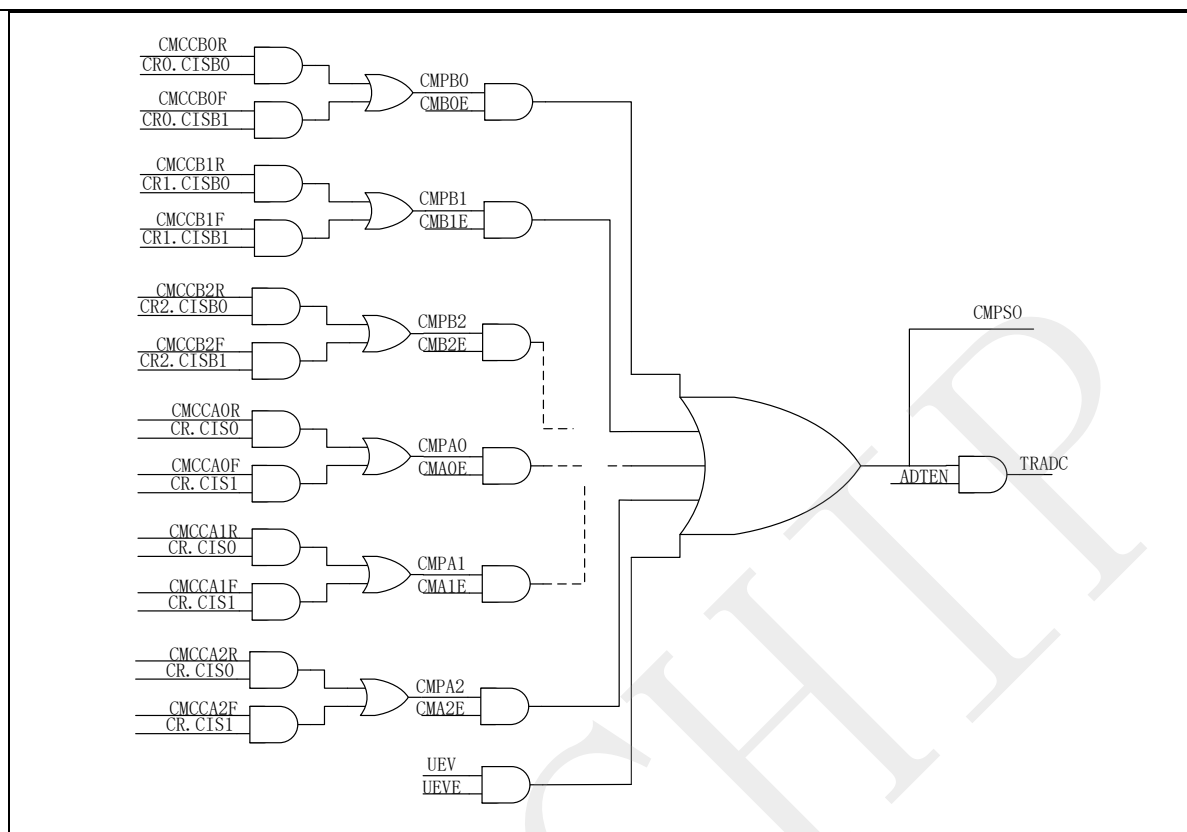


图 29-54 ADC 触发

互补 PWM 输出时 B 比较作为 ADC 触发功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCRxB 设置 ADC 触发比较点
3. 设置 CRCHx.CISB 选择上计数下计数比较匹配（锯齿波不需要选择）
4. 选择 ADC 触发比较的源 ADTR
5. 使能 ADC 触发 ADTR.ADTE

29.2.10 刹车控制

VC 比较输出可以控制刹车功能，外部 BK 端口输入可控制刹车，系统 fail 可以控制刹车功能。通过 CR.BG 可以实现软件刹车功能，控制输出端口到设定的状态。

两通道的输出的 GTIM1/2/3 可以选择使用 GTIM1 的刹车端口控制 GTIM2/3 的刹车功能，也可以使用各自的刹车输入控制各自的刹车功能。当 DTR.BKSEL 设置为 1 时，GTIM2/3 共用 GTIM1 的刹车输入。

29.2.11 Timer 互联

TRGO 输出信号可以连接到其他定时器的 ITR 信号。连接关系如下：

	ITR0	ITR1	ITR2	ITR3
GTIM1	STIM_TRGO	GTIM2_TRGO	GTIM3_TRGO	ATIM_TRGO
GTIM2	GTIM1_TRGO	保留	GTIM3_TRGO	ATIM_TRGO
GTIM3	GTIM1_TRGO	GTIM2_TRGO	保留	ATIM_TRGO

定时器 GTIM1/2/3 同时启动 PWM 输出设置（使用 TIM0 的 CTEN 触发 TIM1/2）
示例

1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
2. 设置 GTIM1 为主模式 MSCR.MSM=1
3. 设置 GTIM1 的 CTEN 触发另外两个定时器 MSCR.MMS=1
4. 保持 GTIM2/3 的 MSCR.MSM=0
5. 设置 GTIM2/3 为触发模式 MSCR.SMS=2
6. 选择 GTIM2/3 的触发源为 GTIM1 的 TRGO, MSCR.TS=1
7. 最后使能定时器 GTIM1，启动定时器 CR.CTEN=1

定时器 GTIM1/2/3 同时启动 PWM 输出设置（使用 GTIM2 的 UG 触发 GTIM1/3）
示例

1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
2. 设置 GTIM2 为触发主模式 MSCR.MSM=1
3. 设置 GTIM2 的 UG 触发另外两个定时器 MSCR.MMS=0
4. 保持 GTIM1/3 的 MSCR.MSM=0
5. 设置 GTIM1/3 为触发模式 MSCR.SMS=2
6. 选择 GTIM1/3 的触发源为 TIM0 的 TRGO, MSCR.TS=2
7. 最后使能定时器 GTIM2，启动定时器 CR.UG=1

29.2.12 GATE 输入互联

GATE 输入可以从端口通过 PxAFR 选择直接输入，也可通过端口功能寄存器 GPIO

MUX_GATE 选择可以连通到其他模块或端口。

当 GTIMx_GATE=0x0 时，门控 GATE 输入为 PxAFR 选择的端口输入，当 GTIMx_GATE=0x1~0x7 时，连接其他模块的输入或输出。

0	1	2	3	4	5	6	7
GTIM1_GT	UART1_RX	UART2_RX	VC1	PE14	PD00	PD01	PD02
GTIM2_GT	UART3_RX	UART4_RX	LVD	VC2	PD04	PD05	PD06
GTIM3_GT	UART5_RX	UART6_RX	PCNT_S1	PE15	PD08	PD09	PD10

29.2.13 ETR 输入互联

ETR 输入可以从端口直接输入，也可通过端口功能寄存器 MUXETR 选择可以连通到其他模块或端口。

当 GTIMx_ETR=0x0 时，外部时钟 ETR 输入为 PxAFR 选择的端口输入，当 GTIMx_ETR=0x1~0x7 时，连接其他模块的输入或输出。

0	1	2	3	4	5	6	7
GTIM1_ETR	UART1_RX	UART2_RX	VC1	PE03	PA04	PB04	PC04
GTIM2_ETR	UART3_RX	UART4_RX	LVD	VC2	PA05	PB05	PC05
GTIM3_ETR	UART5_RX	UART6_RX	PCNT_S1	PE04	PA06	PB06	PC06

29.2.14 CHx 捕获输入互联

GTIM1/2/3 的 CHA，Timer3 的 CH0A/CH0B 输入可以从端口直接输入，也可通过端口功能寄存器 GPIO_TIMCPS 选择可以连通到其他模块或端口。

当 GTIMx_CHy=0x0 时，捕获输入是 PxAFR 选择的端口输入，当 GTIMx_CHy=0x1~0x7 时，连接其他模块的输入或输出。

0	1	2	3	4	5	6	7
GTIM1_CH1	UART1_RX	UART2_RX	VC1	PE10	PA12	PB12	PC12
GTIM2_CH1	UART3_RX	UART4_RX	LVD	VC2	PA13	PB13	PC13
GTIM3_CH1	UART5_RX	UART6_RX	PCNT_S1	PE12	PA14	PB14	PC14
GTIM3_CH2	LPUART1_RX	LPUART2_RX	PCNT_S2	PE13	PA15	PB15	PC15

29.2.15DMA

Timer 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器，或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求，A 请求触发源可选择比较捕获 A，外部触发，B 请求触发源可选择比较捕获 B，事件更新。

IDREQ	Interrupt Signal of Peripheral
18	GTIM1A
19	GTIM1B
20	GTIM2A
21	GTIM2B
22	GTIM3A
23	GTIM3B

GTIMxA 的触发源可选择比较捕获 A,ETRIG

GTIMxB 的触发源可选择比较捕获 B,UEV

29.2.15.1 设置示例

捕获数据 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考数据捕获流程设置数据捕获
9. 使能 CRCHx.CDy 使能捕获数据触发 DAM 传输。

Note: 源地址设置为捕获通道寄存器，源地址固定，目标地址加。

脉冲调整 DMA 数据传输

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择定时器 DMA 的通道
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 参考脉冲调整输出设置脉冲调制输出
9. 根据需要选择硬件触发 DMA 的条件（UEV 触发或者比较匹配触发）

Note: 目标地址设置为捕获通道寄存器，源地址变化，目标地址固定。

29.3 Timer 寄存器描述

Timer	基地址	描述
GTIM1	0x40000000	GTIM1基地址
GTIM2	0x40000400	GTIM2基地址
GTIM3	0x40000800	GTIM3基地址

寄存器	偏移地址	描述
GTIMx_ARR	0x000	Timer重载寄存器/周期
GTIMx_CNT	0x004	Timer 16位模式计数寄存器
GTIMx_CNT32	0x008	Timer 32位模式计数寄存器 注：GTIMx_CNT32是GTIMx_ARR与GTIMx_CNT的重映射寄存器。为了方便32位定时器写操作。
GTIMx_MOCR	0x00C	Timer模式0控制寄存器（按不同模式进行描述）
GTIMx_M1CR	0x00C	Timer模式1控制寄存器（按不同模式进行描述）
GTIMx_M23CR	0x00C	Timer模式23控制寄存器（按不同模式进行描述）
GTIMx_IFR	0x010	Timer中断标志
GTIMx_ICLR	0x014	Timer中断清除寄存器
GTIMx_MSCR	0x018	主从模式控制
GTIMx_FLTR	0x01C	滤波控制
GTIMx_ADTR	0x020	ADC触发控制
GTIMx_CRO	0x024	比较单元0控制寄存器
GTIMx_DTR	0x030	死区寄存器
GTIMx_RCR	0x034	重复计数寄存器
GTIMx_ARRDM	0x038	Timer重载寄存器/周期映射地址
GTIMx_CCROA	0x03C	比较0A寄存器
GTIMx_CCROB	0x040	比较0B寄存器

表 29-1 GTIM 寄存器列表

29.4 模式 0 定时器寄存器描述

29.4.1 16 位模式重载寄存器（GTIMx_ARR）

偏移地址：0x000

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	16位重载定时器重载值

29.4.2 16 位模式计数寄存器（GTIMx_CNT）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	16位重载定时器计数值

29.4.3 32 位模式计数寄存器 (GTIMx_CNT32)

偏移地址: 0x008

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT32[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT32[15:0]															
RW															

位	符号	描述
31:0	CNT32	32位定时器计数值 注: 仅在模式0 32位定时器自由计数模式下有效, 其他模式禁止写该寄存器

29.4.4 控制寄存器 (GTIMx_M0CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
Reserved				Mode				Reserved	UIE		GATEP		GATE		
				RW					RW		RW		RW		
7		6		5		4		3		2		1		0	
Reserved		PRS						TOGEN		CT		MD		CTEN	
		RW						RW		RW		RW		RW	

位	符号	描述
31:14	Reserved	保留位
13:12	MODE	工作模式 00: 定时器模式0; 01: PWC模式; 10: 锯齿波模式; 11: 三角波模式;
11	Reserved	保留位
10	UIE	中断使能控制, 写1后使能中断

9	GATEP	端口 GATE 极性控制 0: 端口GATE高电平有效 1: 端口GATE低电平有效
8	GATE	定时器门控使能 0: 无门控, CTEN=1 时定时器工作 1: 门控使能, 端口 GATE 有效并且CTEN=1时定时器才工作
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;
3	TOGEN	模式0下翻转输出使能 1: 翻转输出使能 0: 翻转输出关闭CHA, CHB输出为低电平
2	CT	计数时钟选择 0: 内部计数时钟TCLK 1: 外部计数时钟ETR;
1	MD	模式选择32定时/16定时模式选择 0: 32位自由计数 1: 16位重载计数
0	CTEN	定时器使能 0: 定时器停止 1: 定时器使能

29.4.5 中断标志寄存器 (GTIMx_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														UIF	
														RO	

位	符号	描述
31:1	RSV	保留位
0	UIF	溢出中断

29.4.6 中断标志清除寄存器 (GTIMx_ICLR)

偏移地址: 0x014

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														UIF	
														R1W0	

位	符号	描述
31:1	RSV	保留位
0	UIF	溢出中断清除, 写0清除

29.4.7 死区时间寄存器 (GTIMx_DTR)

偏移地址: 0x030

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			MOE	Reserved											
			RW												

位	符号	描述
31:13	Reserved	保留位
12	MOE	翻转输出使能 0: 翻转输出为输入状态 1: 翻转端口为输出状态
11:0	Reserved	保留位

29.5 脉冲宽度测量 PWC 寄存器描述

29.5.1 16 位模式计数寄存器 (GTIMx_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	计数值寄存器

29.5.2 控制寄存器 (GTIMx_M1CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
Reserved	Oneshot		Mode		Reserved		UIE		Edg2nd		Edg1st				
	RW		RW				RW		RW		RW				
7		6		5		4		3		2		1		0	
Reserved	PRS					Reserved		CT		Reserved		CTEN			
	RW							RW				RW			

位	符号	描述
31:15	Reserved	保留位
14	Oneshot	单次触发模式选择 1: 完成一次脉冲测量自动结束, 再次测量需要重新使能CTEN 0: 循环测量
13:12	MODE	工作模式

		00: 定时器模式0; 01: PWC模式; 10: 锯齿波模式; 11: 三角波模式;				
11	Reserved	保留位				
10	UIE	中断使能控制，写1后使能中断 计数到0XFFFF才会溢出并产生溢出标志				
9	Edg2nd	脉冲宽度测量结束边沿选择				
8	Edg1st	脉冲宽度测量开始边沿选择				
		edg2nd Edg1st	00	01	10	11
		测量	上沿-上沿周期	低电平宽度	高电平宽度	下沿-下沿周期
7	Reserved	保留位				
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;				
3	Reserved	保留位				
2	CT	计数时钟选择 0: 内部计数时钟TCLK 1: 外部计数时钟ETR				
1	Reserved	保留位				
0	CTEN	脉冲宽度测量使能 0: 禁止 1: 使能				

29.5.3 中断标志寄存器 (GTIMx_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CA0F	RSV	UIF	
												RO		RO	

位	符号	描述
31:3	RSV	保留位
2	CA0F	脉冲宽度测量中断标志
1	RSV	保留
0	UIF	溢出中断标志

29.5.4 中断标志清除寄存器 (GTIMx_ICLR)

偏移地址: 0x014

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CA0F	RSV	UIF	
												R1W0		R1W0	

位	符号	描述
31:3	RSV	保留位
2	CA0F	脉冲宽度测量中断标志清除, 写0清除
1	RSV	保留
0	UIF	溢出中断标志清除, 写0清除

29.5.5 主从模式控制寄存器 (GTIMx_MSCR)

偏移地址: 0x018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			IB0S	IA0S	Reserved			TS		Reserved					
			RW	RW				RW							

位	符号	描述
31:13	RSV	保留位
12	IB0S	CH0B 输入选择 0: CH0B 1: 内部触发TS选择信号 注: 当PWM互补输出时自动选择GATE端口作为CH0B的输入
11	IA0S	IA0 输入选择 0: CH0A 1: CH0A ETR GATE XOR(TIM0)

		注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	RSV	保留位
7:5	TS	触发选择 000：端口ETR的滤波相位选择后的信号ETFP 001：内部互联信号 ITR0 010：内部互联信号 ITR1 011：内部互联信号 ITR2 100：内部互联信号 ITR3 101：端口CH0A的边沿信号 110：端口CH0A的滤波相位选择后的信号IAFP 111：端口CH0B的滤波相位选择后的信号IBFP
4:0	RSV	保留位

29.5.6 输出控制滤波（GTIMx_FLTR）

偏移地址：0x01C

复位值：0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FLTB0		Reserved	FLTA0			
									RW			RW			

位	符号	描述
31	ETP	ETR 输入相位选择 0：同相位 1：反向输入
30:28	FLTET	ETR滤波控制 滤波设置 0xx：滤波无效 100：pc1k 3个连续有效 101：pc1k/4 3个连续有效 110：pc1k/16 3个连续有效 111：pc1k/64 3个连续有效
27:7	Reserved	保留位
6:4	FLTB0	CHB 输入滤波控制 滤波设置 0xx：滤波无效

		100: pc1k 3个连续有效 101: pc1k/4 3个连续有效 110: pc1k/16 3个连续有效 111: pc1k/64 3个连续有效
3	Reserved	保留位
2:0	FLTA0	CHA 输入滤波控制 滤波设置 0xx: 滤波无效 100: pc1k 3个连续有效 101: pc1k/4 3个连续有效 110: pc1k/16 3个连续有效 111: pc1k/64 3个连续有效

29.5.7 控制寄存器（GTIMx_CR0）

偏移地址：0x024

复位值：0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							CIEA	Reserved							
							RW								

位	符号	描述
31:9	Reserved	保留位
8	CIEA	CIEA 脉冲宽度测量完成中断使能 0: 禁止 1: 使能
7:0	Reserved	保留位

29.5.8 比较捕获寄存 (GTIMx_CCR0A)

偏移地址: 0x03C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0A															
R															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0A	脉冲宽度测量结果

29.6 模式 2,3 寄存器描述

29.6.1 16 位模式重载寄存器 (GTIMx_ARR)

偏移地址: 0x000(0x038 dummy address)

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	重载寄存器/周期寄存器, 具有缓存功能 在计数器未使能或者缓存没有使能时, 缓存寄存器可以立刻更新

29.6.2 16 位模式计数寄存器 (GTIMx_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	重载定时器计数值

注: 使用 PWM 比较输出时, 初始化 CNT 值需要小于 ARR 的值。

29.6.3 控制寄存器 (GTIMx_M23CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24
Reserved				DIR	BG	UG	TG
				RW/RO	W1	W1	W1
23	22	21	20	19	18	17	16
OCCE	CIS	BIE	TIE	TDE	URS	OCCS	
RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8
CSG	Oneshot	Mode	UDE	UIE	CFG	CRG	
RW	RW	RW	RW	RW	RW	RW	RW
7	6	5	4	3	2	1	0
BUFPEN	PRS	Pwm2s	CT	Comp	CTEN		
RW	RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:28	Reserved	保留位
27	DIR	计数方向，只有在锯齿波模式下可以写。其他模式下只读，写无效 0：上计数 1：下计数 注：从其他模式切换到中心对齐模式DIR自动清0。软件事件更新和从模式外部触发复位模式DIR自动清零。
26	BG	软件刹车，自动清零 写1产生软件刹车 写0无效
25	UG	软件更新，自动清零 写1产生软件更新 写0无效 初始化计数器并更新缓存寄存器到相应寄存器（缓存使能），预除频计数器也会被清零。
24	TG	软件触发，自动清零，需要在触发模式SMS=2且mode=2/3下都可以触发。 写1产生软件触发 写0无效
23	OCCE	OCREF清除使能 1：OCREF_CLR信号可以清除OCREF输出 0：OCREF输出不受OCREF_CLR影响
22:21	CIS	中心对齐A比较中断模式(B比较中断单独控制在CRx寄存器中CISB) 00：无中断 01：上沿中断 10：下沿中断 11：上下沿都中断
20	BIE	刹车中断使能 1：中断使能 0：中断禁止
19	TIE	触发中断使能 1：中断使能 0：中断禁止
18	TDE	触发DMA使能 1：中断使能 0：中断禁止
17	URS	更新源 0：上溢出/下溢出/软件更新UG/从模式复位 1：上溢出/下溢出
16	OCCS	OCREF 清除源选择 0：电压比较器VC输出，VC选择在VCx_OUTCFG寄存器设置 1：ETR端口滤波相位选择后的信号 当OCCE有效时，OC_clr可以清除OCREF的比较输出信号为零，（当OCMx>1时有效），下一个uev事件后继续比较输出

15	CSG	GATE 在PWM互补模式下捕获/比较选择；（只有在PWM互补输出时有效） 使用CCR0B作为GATE的比较或捕获通道 1：捕获 0：比较
14	Oneshot	单次触发模式选择 1：发生事件更新后定时器停止 0：循环计数
13:12	MODE	工作模式 00：定时器模式0； 01：PWC模式； 10：锯齿波模式； 11：三角波模式；
11	UDE	更新DMA使能 1：使能更新触发DMA 0：禁止更新触发DMA
10	UIE	UIE 更新中断使能 1：使能更新中断 0：禁止更新中断
9	CFG	GATE作为捕获输入时，下沿捕获有效控制（只有在PWM互补输出时有效） 1：下沿捕获有效 0：下沿捕获无效
8	CRG	GATE作为捕获输入时，上沿捕获有效控制（只有在PWM互补输出时有效） 1：上沿捕获有效 0：上沿捕获无效
7	BUFPEN	重载缓存使能 1：周期缓存使能，写入后下个事件更新后才影响到周期值 0：周期缓存无效，写入后立刻影响周期值
6:4	PRS	内部时钟分频选择 000：1； 001：2； 010：4； 011：8； 100：16； 101：32； 110：64； 111：256；
3	PWM2S	OCREFA双点比较选择(缺省值为1) 0：双点比较使能，使用CCRA, CCRB比较控制OCREFA输出 1：单点比较使能，只使用CCRA比较控制OCREFA输出 注：OCREFB不受影响，仍然使用CCRB控制OCREFB输出
2	CT	计数时钟选择 0：内部计数时钟TCLK 1：外部计数时钟ETR
1	Comp	PWM互补输出模式选择 0：独立PWM输出 1：互补PWM输出
0	CTEN	定时器使能 0：禁止 1：使能 可以外部触发使能，在oneshot模式下结束时该位自动清零

29.6.4 中断标志寄存器 (GTIMx_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF	BIF	Reserved		CBOE	Reserved		CAOE	Reserved		CBOF	Reserved		CAOF	RSV	UIF
RO	RO			RO			RO			RO			RO		RO

位	符号	描述
31:16	Reserved	保留位
15	TIF	触发中断标志
14	BIF	刹车中断标志
13:12	Reserved	保留位
11	CBOE	通道CH0B捕获数据丢失标志 0: 无数据丢失 1: 数据丢失
10:9	Reserved	保留位
8	CAOE	通道CH0A捕获数据丢失标志 0: 无数据丢失 1: 数据丢失
7:6	Reserved	保留位
5	CBOF	通道CH0B发生捕获/比较匹配标志 0: 没有发生 1: 发生
4:3	Reserved	保留位
2	CAOF	通道CH0A发生捕获/比较匹配标志 0: 没有发生 1: 发生
1	RSV	保留位
0	UIF	事件更新中断标志 0: 没有发生 1: 发生

29.6.5 中断标志清除寄存器 (GTIMx_ICLR)

偏移地址: 0x014

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF	BIF	RSV		CB0E	RSV		CA0E	RSV		CB0F	RSV		CA0F	RSV	UIF
R1W0	R1W0			R1W0			R1W0			R1W0			R1W0		R1W0

位	符号	描述
31:16	RSV	保留位
15	TIF	触发中断标志清除, 写0清除
14	BIF	刹车中断标志清除, 写0清除
13:12	RSV	保留位
11	CB0E	通道CH0B捕获数据丢失标志清除, 写0清除
10:9	RSV	保留位
8	CA0E	通道CH0A捕获数据丢失标志清除, 写0清除
7:6	RSV	保留位
5	CB0F	通道CH0B捕获/比较匹配标志清除, 写0清除
4:3	RSV	保留位
2	CA0F	通道CH0A捕获/比较匹配标志清除, 写0清除
1	RSV	保留
0	UIF	事件更新中断标志清除, 写0清除

29.6.6 主从模式控制寄存器（GTIMx_MSCR）

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			IBOS	IAOS	SMS			TS			MSM	CCDS	MMS		
			RW	RW	RW			RW			RW	RW	RW		

位	符号	描述
31:13	RSV	保留位
12	IBOS	CHOB 输入选择 0: CHOB 1: 内部触发TS选择信号 注：当PWM互补输出时自动选择GATE端口作为CHOB的输入
11	IAOS	IAO 输入选择 0: CHOA 1: CHOA ETR GATE XOR 注：设置为1后端口的任意一个端口变化将导致输入变化
10:8	SMS	从模式功能选择 000: 使用内部时钟 001: 复位功能 010: 触发模式 011: 外部时钟模式 100: 正交编码计数模式1 101: 正交编码计数模式2 110: 正交编码计数模式3 111: 门控功能
7:5	TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1 011: 内部互联信号 ITR2 100: 内部互联信号 ITR3 101: 端口CH0A的边沿信号 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口CH0B的滤波相位选择后的信号IBFP
4	MSM	主从选择

		0: 无延时 1: 延时使能, 使主送计数器同时启动 注: 使用触发模式时, 从模式设置为0, 主模式设置为1, 可以使主从计数同时启动	
3	CCDS	比较模式下 DMA 比较触发选择 0: 比较匹配触发DMA 1: 比较匹配不触发DMA, 事件更新代替比较匹配触发DMA	
2:0	MMS	主模式输出选择, 用于内部互联, 连接到其他定时器的ITRx	
		定时器0/1/2	定时器3
		000: 软件更新UG, 写CR.UG 001: 定时器使能CTEN 010: 定时器事件更新UEV 011: 比较匹配选择输出CMPSO 100: 定时器比较参数输出OCREF0A 101: 定时器比较参数输出OCREF0B 110: 定时器比较参数输出OCREF0B 111: 定时器比较参数输出OCREF0B	000: 软件更新UG, 写CR.UG 001: 定时器使能CTEN 010: 定时器事件更新UEV 011: 比较匹配选择输出CMPSO 100: 定时器比较参数输出OCREF0A 101: 定时器比较参数输出OCREF1A 110: 定时器比较参数输出OCREF2A 111: 定时器比较参数输出OCREF0B

29.6.7 输出控制/输入滤波 (GTIMx_FLTR)

偏移地址: 0x01C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET			BKP	FLTBK			RSV							
-	-			-	-										
RW	RW			RW	RW										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV								CCPB0	OCMB0		CCPA0	OCMA0			
								CCPB0	FLTB0		CCPA0	FLTA0			
								RW	RW		RW	RW			

位	符号	描述
31	ETP	ETR 输入相位选择 0: 同相位 1: 反向输入
30:28	FLTET	ETR滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效
27	BKP	刹车BK输入相位选择 0: 同相位 1: 反向输入
26:24	FLTBK	刹车输入滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效 111: pclk/64 3个连续有效 注: 为了保证OCMB0的PWM输出设置, GATE 在PWM互补模式下作为捕获输入, 滤波设置无效。
23:8	RSV	保留位
7	CCPB0	比较功能: 输出比较模式 CCPBx比较输出CHBx端口极性控制 0: 正常输出

		1: 反向输出 编码计数与从模式门控功能: 输入相位控制 CCPB0从模式门控, 复位, 外部触发, 外部时钟使用CH0B端口输入极性控制 0: 正常输入 1: 反向输入
6:4	OCMB0 FLTB0	比较功能: CH0B通道比较控制 000: 强制为0 001: 强制为1 010: 比较匹配时强制为0 011: 比较匹配时强制为1 100: 比较匹配时翻转 101: 比较匹配时输出一个计数周期的高电平 110: PWM 模式1 单点比较: 上计数时 $CNT < CCR_{xy}$ 输出高, 下计数时 $CNT > CCR_{xy}$ 输出为低电平 双点比较: 1) 锯齿波上计数 $CCR_xA < CNT \leq CCR_xB$ 输出为低电平 2) 锯齿波下计数 $CCR_xA < CNT \leq CCR_xB$ 输出为高电平 3) 三角波上计数 $CNT < CCR_xA$ 输出高, 下计数 $CNT > CCR_xB$ 为低电平 111 PWM 模式2 单点比较: 上计数时 $CNT < CCR_{xy}$ 输出低, 下计数时 $CNT > CCR_{xy}$ 输出为高电平 双点比较: 1) 锯齿波上计数 $CCR_xA \leq CNT < CCR_xB$ 输出为高电平 2) 锯齿波下计数 $CCR_xA \leq CNT < CCR_xB$ 输出为低电平 3) 三角波上计数 $CNT < CCR_xA$ 输出低, 下计数 $CNT > CCR_xB$ 为高电平 捕获功能: CH0B输入通道滤波设置, 参考FLTBK
3	CCPA0	比较功能: CCPAx比较输出CHAx端口极性控制 0: 正常输出 1: 反向输出 编码计数与从模式门控功能: 输入相位控制 CCPA0从模式门控, 复位, 外部触发, 外部时钟使用CH0A端口输入极性控制 0: 正常输入 1: 反向输入
2:0	OCMA0 FLTA0	比较功能: A通道比较控制; 参考OCMB0 捕获功能: CH0A输入通道滤波设置, 参考FLTBK

29.6.8 ADC 触发控制寄存器 (GTIMx_ADTR)

偏移地址: 0x020

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
ADTE		CMB2E		CMB1E		CMB0E		CMA2E		CMA1E		CMA0E		UEVE									
RW		RW		RW		RW		RW		RW		RW		RW									

位	符号	描述
31:8	Reserved	保留位
7	ADTE	使能ADC触发全局控制 1: 使能 0: 禁止
6	CMB2E	通道2B比较匹配触发ADC使能, 只有TIM3存在 1: 使能 0: 禁止
5	CMB1E	通道1B比较匹配触发ADC使能, 只有TIM3存在 1: 使能 0: 禁止
4	CMBOE	通道0B比较匹配触发ADC使能 1: 使能 0: 禁止
3	CMA2E	通道2A比较匹配触发ADC使能, 只有TIM3存在 1: 使能 0: 禁止
2	CMA1E	通道1A比较匹配触发ADC使能, 只有TIM3存在 1: 使能 0: 禁止
1	CMA0E	通道0A比较匹配触发ADC使能 1: 使能 0: 禁止
0	UEVE	事件更新触发ADC使能 1: 使能 0: 禁止

29.6.9 通道 0 控制寄存器 (GTIMx_CRCH0)

偏移地址: 0x024

复位值: 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	BKSB		BKSA		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

位	符号	描述
31:16	Reserved	保留位
15	CCGB	捕获比较B软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00: 无匹配 01: 上升匹配 10: 下降匹配 11: 双匹配
11	CDEB	B捕获比较触发DMA使能 0: 禁止 1: 使能
10	CDEA	A捕获比较触发DMA使能 0: 禁止 1: 使能
9	CIEB	B捕获比较触发中断使能 0: 禁止 1: 使能
8	CIEA	A捕获比较触发中断使能 0: 禁止 1: 使能
7	BUFEB	比较功能: B比较缓存使能控制 0: 禁止 1: 使能

6	BUFEA	比较功能：A比较缓存使能控制 0：禁止 1：使能
5	CSB	B通道捕获/比较功能选择 0：比较模式 1：捕获模式
4	CSA	A通道捕获/比较功能选择 0：比较模式 1：捕获模式
3	CFB	B通道下降沿捕获使能 0：禁止 1：使能
2	CRB	B通道上升沿捕获使能 0：禁止 1：使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00：高阻输出 01：对输出无影响 10：强制输出低电平 11：强制输出高电平
1	CFA	A通道下降沿捕获使能 0：禁止 1：使能
0	CRA	A通道上升沿捕获使能 0：禁止 1：使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00：高阻输出 01：对输出无影响 10：强制输出低电平 11：强制输出高电平

29.6.11 重复周期设置值 (GTIMx_RCR)

偏移地址: 0x034

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RCR							
								RW							

位	符号	描述
31:8	Reserved	保留位
7:0	RCR	重复周期计数值 设置RCR+1个周期个上溢出下溢出后产生事件更新, 计数器上溢出或下溢出时内部RCR_CNT减1, 当计数到零后RCR_CNT重载RCR的值, 并且产生事件更新UEV信号

29.6.12 通道比较捕获寄存器 (GTIMx_CCR0A/B)

偏移地址:

GTIMx_CCR0A : 0x03C

GTIMx_CCR0B : 0x040

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0y															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0y	比较捕获, 比较具有缓存功能(y=A,B)

30. 基本定时器（BTIM）

30.1 概述

BTIM 是一组具有基本定时功能的定时计数器，每组包含 3 个 16 位的具有重载功能的定时器。本产品含有三组基本定时器：BTIM1/2/3、BTIM4/5/6、BTIM7/8/9。

30.2 主要特性

- 9 个 16bit 定时器
- 每个定时器 2 个翻转输出
- 单脉冲模式
- 外部计数功能
- 门控功能
- 外部触发功能
- 级联功能

30.3 功能描述

30.3.1 功能框图

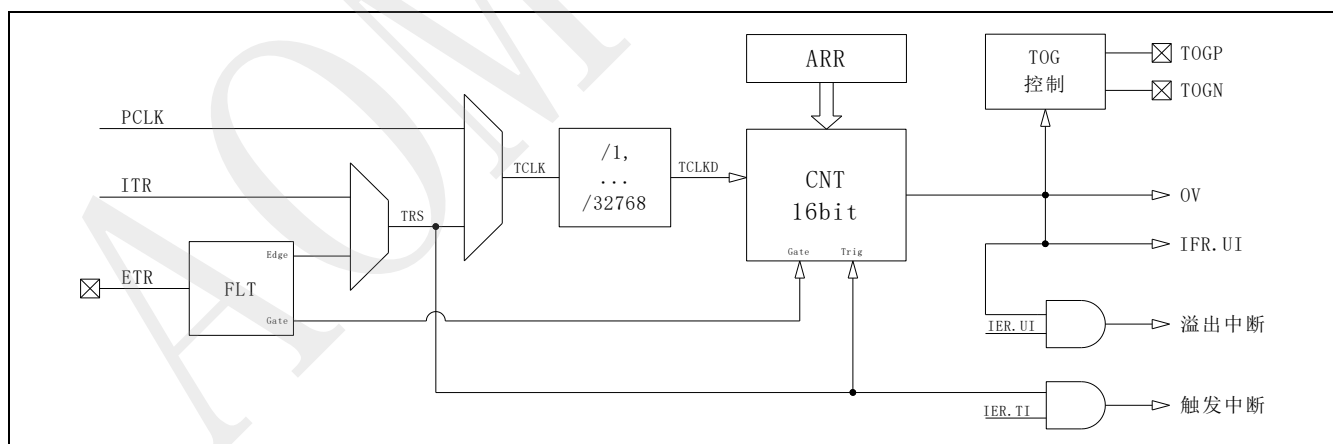


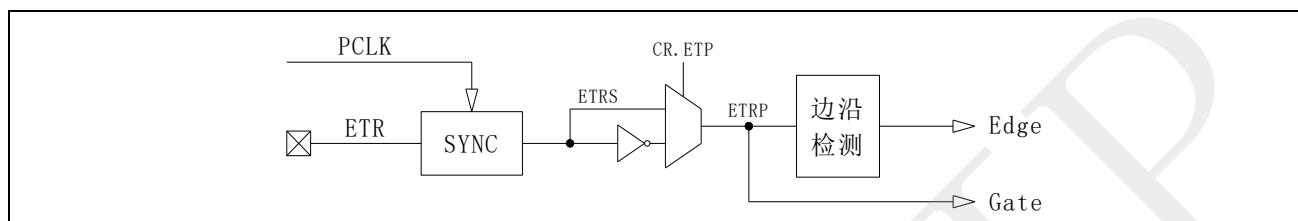
图 30-1 BTIM 框图

注：

- BTIM1/2/3 共用一个 ETR, BTIM4/5/6 共用一个 ETR, BTIM7/8/9 共用一个 ETR。
- ITR 信号来源详见级联功能章节。

30.3.2 滤波单元

外部管脚输入的 ETR 信号经滤波、极性选择、边沿检测后，输出门控信号及边沿信号供计数单元及触发单元使用。由于 ETR 信号采用 PCLK 及其分频进行同步滤波，故输入的 ETR 信号的频率应低于 PCLK 的频率的二分之一。



30.3.3 计数单元

在每个 TCLKD 信号的上升沿，内部计数器 CNT 自动加 1。计数到 ARR 后产生溢出信号 OV（宽度为一个 PCLK），并复位内部计数值为 0 并再次开始加计数。计数器的计数范围为 0 至 ARR，计数单元的计数周期为 ARR+1。

定时器支持 2 种计数模式，One-Shot 模式与连续计数模式。当设置 BTIMx_CR.OST 为 1 时，定时器工作于 One-Shot 模式；发生溢出后定时器立即停止计数，硬件自动清除 CEN。当设置 BTIMx_CR.OST 为 0 时，定时器工作于连续计数模式；只要 BTIMx_CR.CEN 为 1，定时器会一直进行计数。

每次溢出时，BTIMx_IFR.UI 会被硬件置 1，若使能相应中断则可以产生溢出中断，用户应在中断服务程序中清除该溢出标志。

若 BTIMx_CR.TOGEN 为 1，则每溢出一次 TOGP/TOGN 输出电平翻转一次；用户可通过 GPIO 的辅助功能将 TOGP/TOGN 信号输出到管脚用于驱动外部电路。

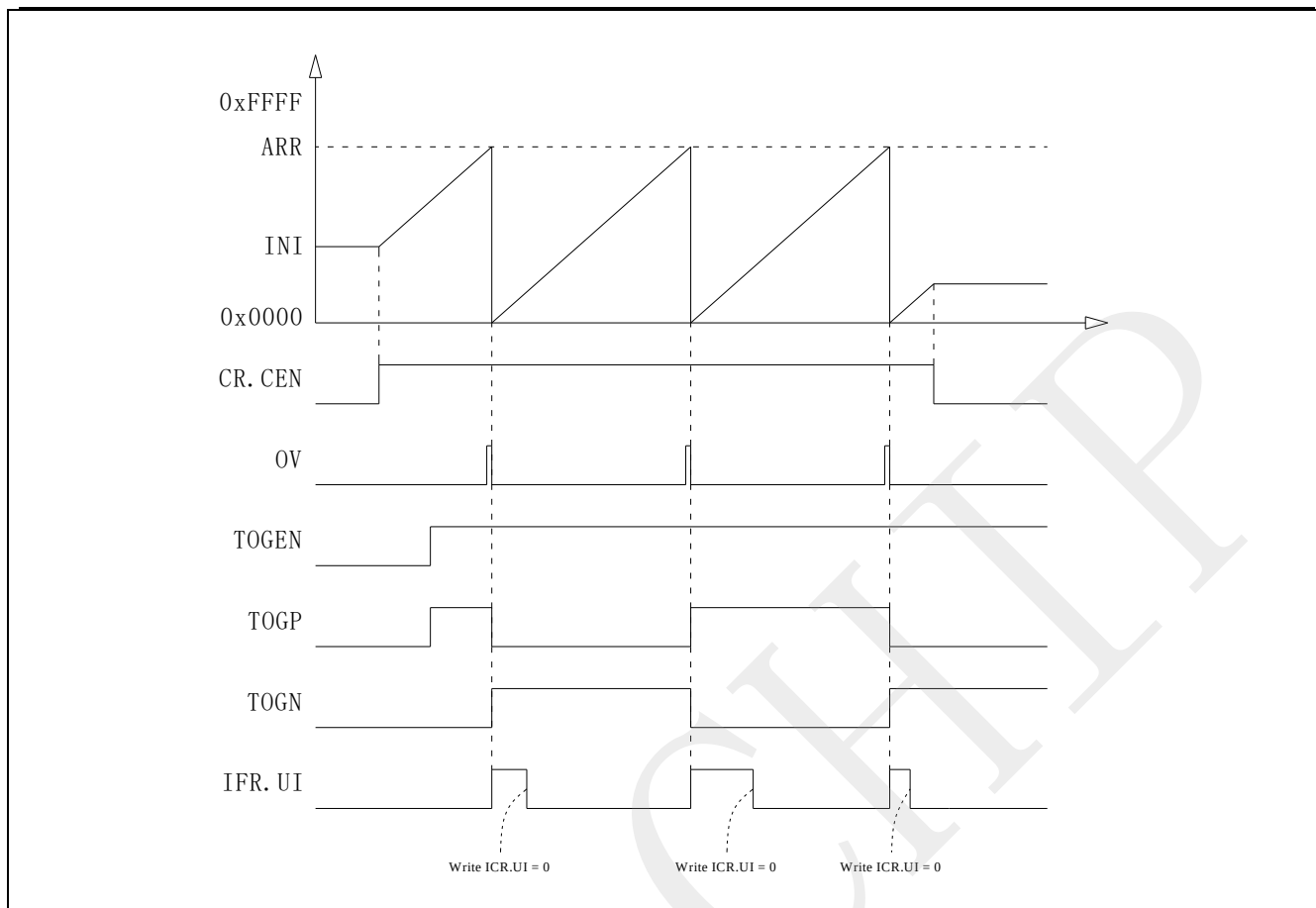


图 30-2 计数器波形

30.3.4 定时器模式

当设置 BTIMx_CR.MD 为 0x00 时，BTIMx 工作于定时器模式。在该模式下，计数时钟为分频后的 PCLK，当设置 BTIMx_CR.CEN 为 1 后，定时器即从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数。

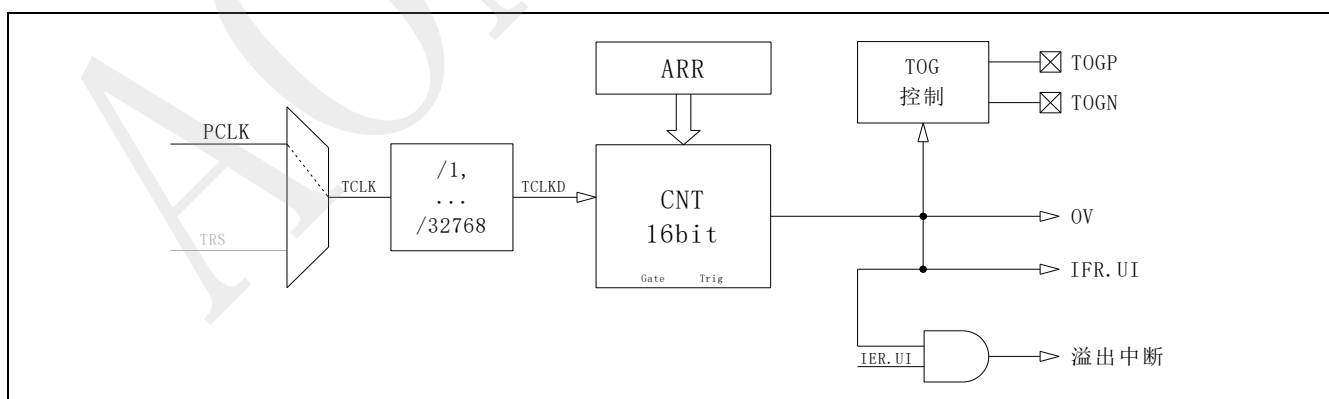


图 30-3 定时模式框图

30.3.5 计数器模式

当设置 BTIMx_CR.MD 为 0x01 时, BTIMx 工作于计数器模式。在该模式下, 计数时钟为分频后的 ITR 或 ETR 管脚输入的信号。当设置 BTIMx_CR.CEN 为 1 后, 计数器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 BTIMx_CR.TRS 可以选择计数时钟源为 ITR 或 ETR; 通过 BTIMx_CR.ETP 可以选择对 ETR 管脚输入信号的上升沿或下降沿进行计数。

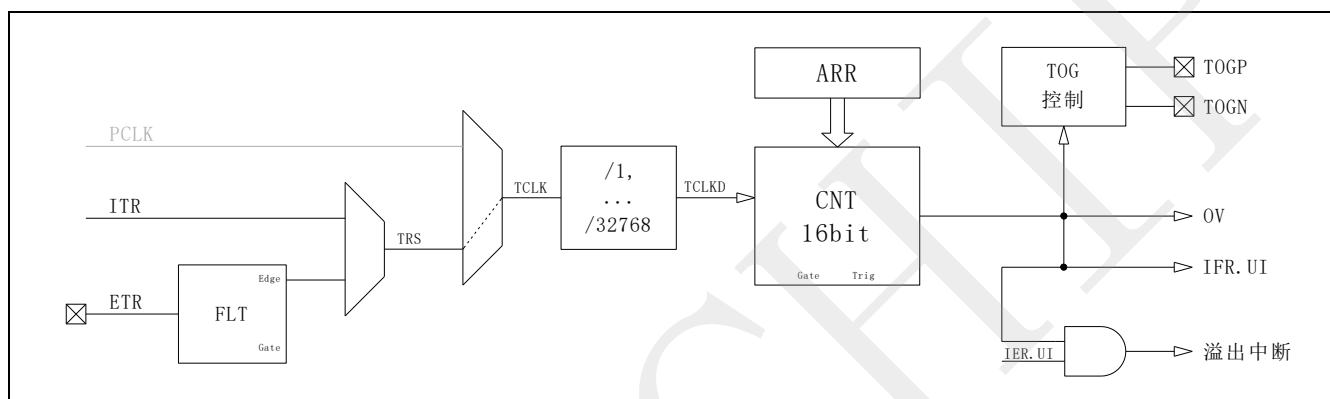


图 30-4 计数器模式框图

30.3.6 触发启动模式

当设置 BTIMx_CR.MD 为 0x02 时, BTIMx 工作于触发启动模式, 计数时钟为分频后的 PCLK。该模式下定时器具有两种启动方法: 设置 BTIMx_CR.CEN 为 1 或 TRS 信号上符合预期的电平变化。

启动后, 计数器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数; 在任意时刻设置 BTIMx_CR.CEN 为 0, 则可以停止定时器。

通过 BTIMx_CR.TRS 选择 TRS 信号来原为 ETR 或 ITR。

通过 BTIMx_CR.ETP 可以选择 ETR 管脚输入信号的上升沿或下降沿启动定时器。

每次检测到符合预期的触发信号时, BTIMx_IFR.TI 会被硬件置 1, 若使能相应中断则可以产生触发中断, 用户应在中断服务程序中清除该触发标志。

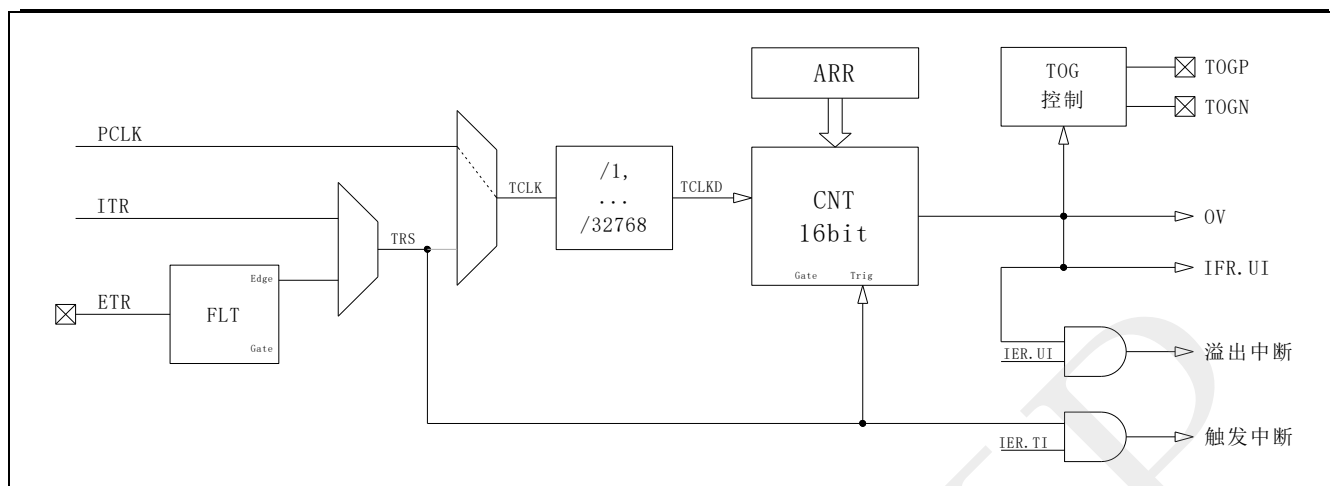


图 30-5 触发启动器模式框图

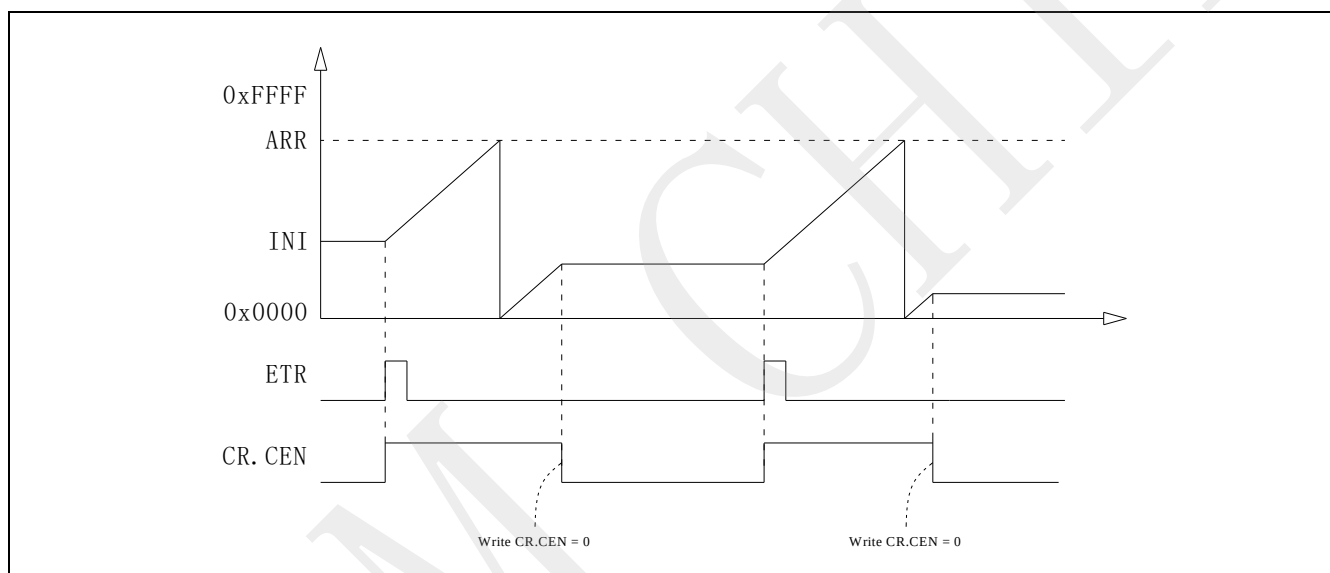


图 30-6 触发启动模式计数示意图

30.3.7 门控模式

当设置 BTIMx_CR.MD 为 0x03 时，BTIMx 工作于门控模式，计数时钟为分频后的 PCLK。该模式下定时器需要满足两个条件才会进行计数：BTIMx_CR.CEN 为 1、ETR 管脚上的电平符合预期。定时器运行时，计数器从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 BTIMx_CR.ETP 可以选择 ETR 管脚输入信号的高电平或低电平为计数电平。

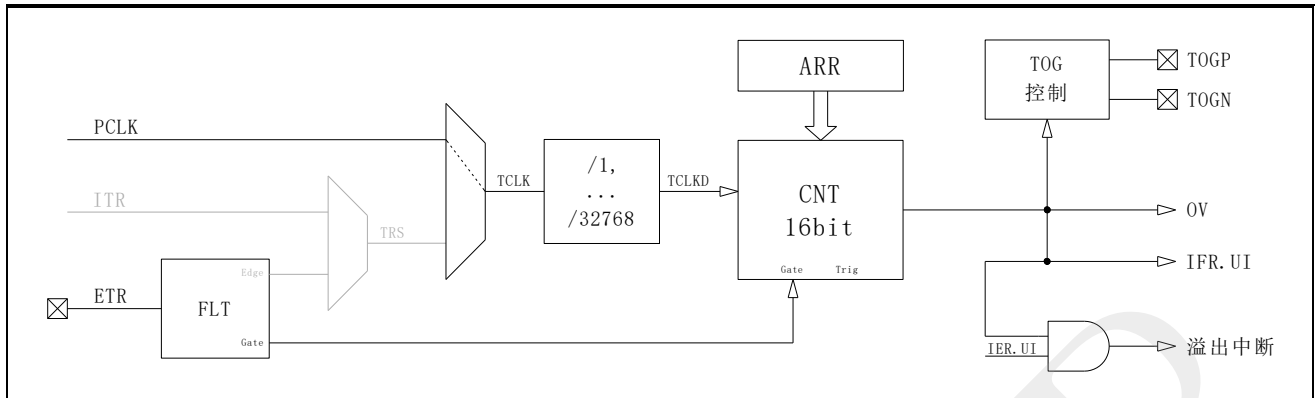


图 30-7 门控模式框图

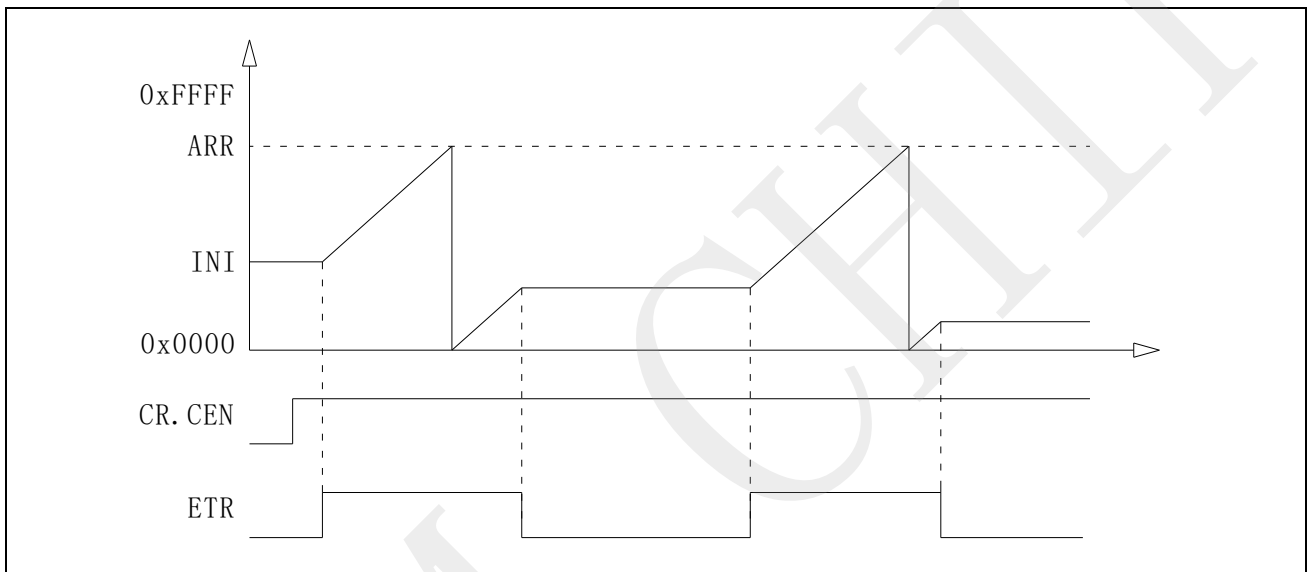


图 30-8 门控模式计数示意图

30.4 BTIM 寄存器描述

Timer	基地址	描述
BTIM1	0x40014000	BTIM1基地址
BTIM2	0x40014100	BTIM2基地址
BTIM3	0x40014200	BTIM3基地址
BTIM4	0x40014400	BTIM4基地址
BTIM5	0x40014500	BTIM5基地址
BTIM6	0x40014600	BTIM6基地址
BTIM7	0x40014800	BTIM7基地址
BTIM8	0x40014900	BTIM8基地址
BTIM9	0x40014A00	BTIM9基地址

BTIMx_ARR	0x000	重载寄存器/周期
BTIMx_CNT	0x004	16位模式计数寄存器
BTIMx_CR	0x010	控制寄存器
BTIMx_IER	0x014	中断使能
BTIMx_IFR	0x018	中断标志
BTIMx_ICR	0x01C	中断清除寄存器
BTIMx_ACR	0x00C	总控寄存器
BTIMx_AIER	0x044	总控中断使能
BTIMx_AIFR	0x048	总控中断标志
BTIMx_AICR	0x04C	总控中断清除寄存器

表 30-1 BTIM 寄存器列表

30.4.116 位模式重载寄存器 (BTIMx_ARR) (x=1-9)

偏移地址: 0x000

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	16位重载定时器重载值

30.4.216 位模式计数寄存器 (BTIMx_CNT) (x=1-9)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	16位重载定时器计数值

30.4.3 控制寄存器（BTIMx_CR）(x=1-9)

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ETP	TRS	OST	PRS			OEN	MD		CEN		
				RW	RW	RW	RW			RW	RW		RW		

位	符号	描述
31:12	Reserved	保留位
11	ETP	ETR电平/边沿选择
10:9	TRS	触发源选择 00: ETR 01: 保留 1x: 保留
8	OST	ONE-SHOT
7:4	PRS	预除频，参考定时器预分频章节
3	TOGEN	翻转输出使能，PWM输出时此控制位需要设置为0
2:1	MD	模式选择(bit1= cks bit2=ens) 00: 内部时钟 01: 触发源作为外部时钟 10: 触发源作为触发启动定时器 11: ETR作为定时器门控
0	CEN	定时器使能 0: 定时器停止 1: 定时器使能

30.4.4 中断使能（BTIMx_IER）(x=1-9)

偏移地址：0x014

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														EIE	UIE
														RW	RW

位	符号	描述
31:2	RSV	保留位
1	EIF	触发中断使能
0	UIF	溢出中断使能

30.4.5 中断标志寄存器（BTIMx_IFR）(x=1-9)

偏移地址：0x018

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														EIF	UIF
														RO	RO

位	符号	描述
31:2	RSV	保留位
1	EIF	触发中断
0	UIF	溢出中断

30.4.6 中断标志清除寄存器 (BTIMx_ICR) (x=1-9)

偏移地址: 0x01C

复位值: 0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														EIF	UIF
														R1W0	R1W0

位	符号	描述
31:2	RSV	保留位
1	EIF	触发中断清除, 写0清除
0	UIF	溢出中断清除, 写0清除

30.4.7 CR 寄存器 (BTIMxyz_ACR) (xyz = 123 / 456 / 789)

偏移地址: 0x00C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									EFLT		RSV	CENz	CENy	CENx	
									RW			RW	RW	RW	

位	符号	描述
31:7	Reserved	保留位
6:4	ETRFLT	BTIMxyz ETR Filter 000: 无滤波 001: $F_{\text{sample}} = \text{pclk} N=2$ 010: $F_{\text{sample}} = \text{pclk} N=4$ 011: $F_{\text{sample}} = \text{pclk} N=6$ 100: $F_{\text{sample}} = \text{pclk}/4 N=4$ 101: $F_{\text{sample}} = \text{pclk}/4 N=6$ 110: $F_{\text{sample}} = \text{pclk}/16 N=4$

		111: $F_{\text{sample}} = \text{pclk}/16 \text{ N}=6$
3	RSV	保留位
2	CENz	BTIMz使能
1	CENy	BTIMy使能
0	CENx	BTIMx使能

30.4.8 中断使能 (BTIMxyz_AIER) (xyz = 123 / 456 / 789)

偏移地址: 0x044

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EIEz	UIEz	EIEy	UIEy	EIEx	UIEx
										RW	RW	RW	RW	RW	RW

位	符号	描述
31:6	RSV	保留位
5	EIEz	BTIMz触发中断使能
4	UIEz	BTIMz溢出中断使能
3	EIEy	BTIMy触发中断使能
2	UIEy	BTIMy溢出中断使能
1	EIEx	BTIMx触发中断使能
0	UIEx	BTIMx溢出中断使能

30.4.9 中断标志寄存器（BTIM_{xyz}_AIFR）（xyz = 123 / 456 / 789）

偏移地址：0x048

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EIFz	UIFz	EIFy	UIFy	EIFx	UIFx
										RO	RO	RO	RO	RO	RO

位	符号	描述
31:6	RSV	保留位
5	EIFz	BTIMz触发中断
4	UIFz	BTIMz溢出中断
3	EIFy	BTIMy触发中断
2	UIFy	BTIMy溢出中断
1	EIFx	BTIMx触发中断
0	UIFx	BTIMx溢出中断

30.4.10 中断标志清除寄存器（BTIM_{xyz}_AICR）（xyz = 123 / 456 / 789）

偏移地址：0x04C

复位值： 0x0000 003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EIFz	UIFz	EIFy	UIFy	EIFx	UIFx
										R1W0	R1W0	R1W0	R1W0	R1W0	R1W0

位	符号	描述
31:6	RSV	保留位
5	EIFz	BTIMz触发中断清除，写0清除
4	UIFz	BTIMz溢出中断清除，写0清除
3	EIFy	BTIMy触发中断清除，写0清除
2	UIFy	BTIMy溢出中断清除，写0清除
1	EIFx	BTIMx触发中断清除，写0清除
0	UIFx	BTIMx溢出中断清除，写0清除

31. 低功耗定时器（LPTIM）

31.1 LPTimer 简介

LPTimer 是异步 16 位定时/计数器, 在系统时钟关闭后仍然可以通过内部低速 RC 或者外部低速晶体振荡计时/计数。通过中断在低功耗模式下唤醒系统。

LPTimer 支持最大 256 的预分频, 系统内有两个低功耗定时器, 可以通过内部进行级联。

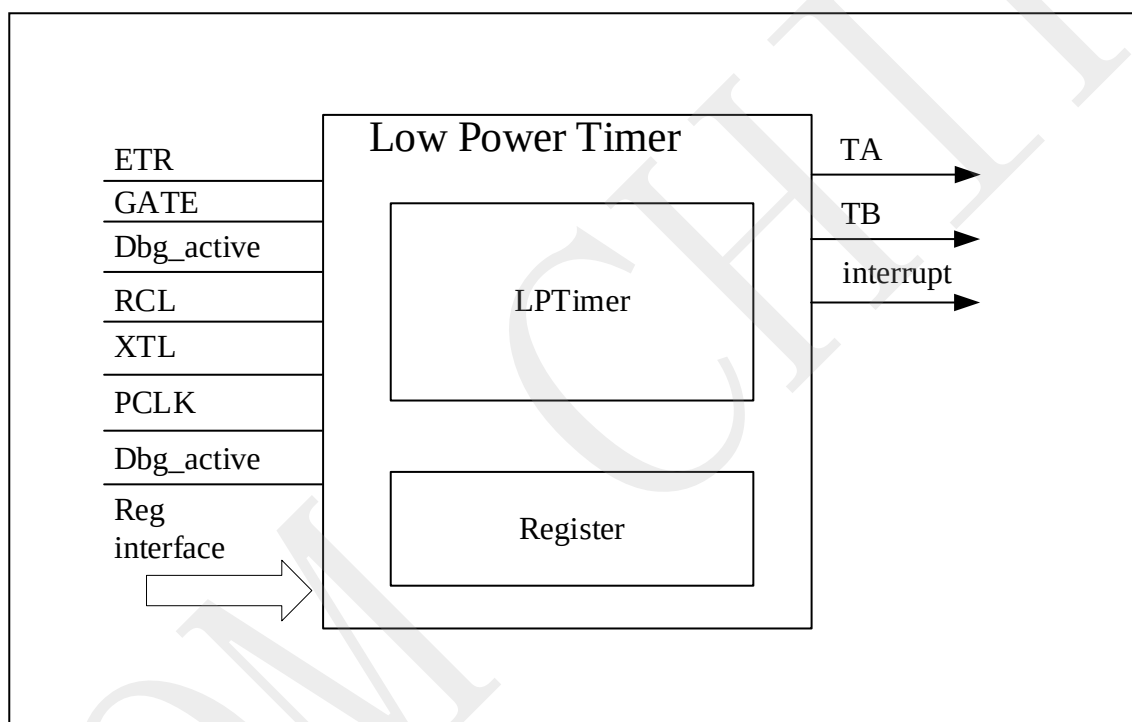


图 31-1 LPTimer 框图

31.2 LPTimer 功能描述

LPTimer 支持 2 种工作模式, 每个定时器/计数器都有独立的控制启动信号, 以及外部输入时钟, 门控信号。

LPTimer 使用 EXT, GATE 来进行计数功能, EXT 用于计数器的外部输入时钟信号, Gate 用于有效电平计数使能信号。

LPTimer 的定时器支持两种工作模式, 通过设置定时器控制寄存器 (CR) 中 MD 选择工作模式。模式 1 为 16 位自由计数模式。模式 2 是 16 位重载模式。

LPTimer 启动时会自动装载重载寄存器 ARR 的值到计数器中。

LPTimer 可选三种时钟作为定时器时钟，通过控制寄存器 CR.TCK_SEL 选择。默认选择 PCLK。时钟选择如表：

TCK_SEL	00	01	10	11
定时器时钟	PCLK	PCLK	XTL	RCL
读定时器计数值	读经过同步	无同步	读经过同步	读经过同步

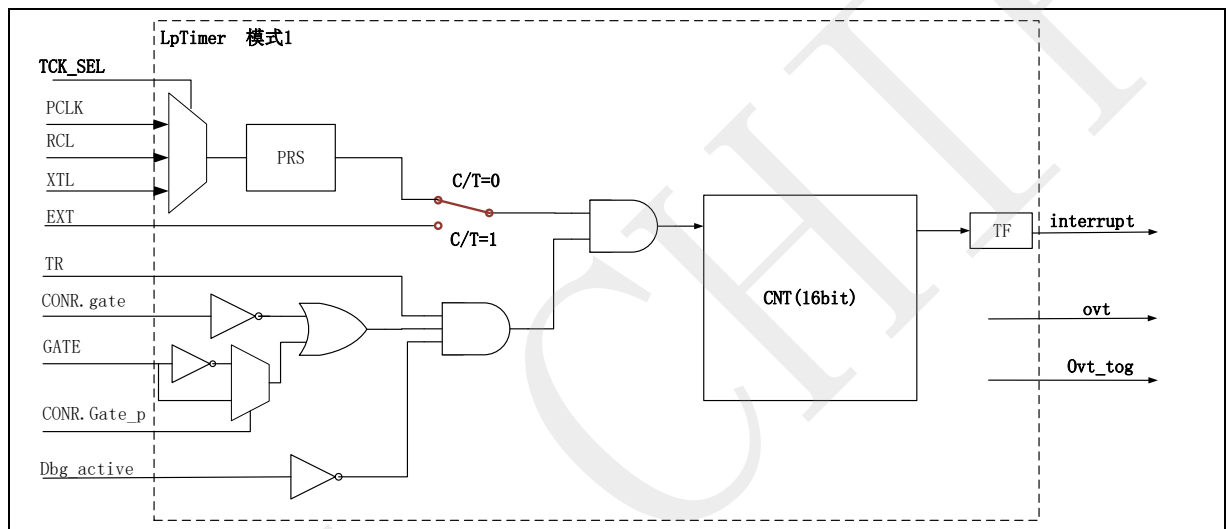


图 31-2 LPTimer 模式 1

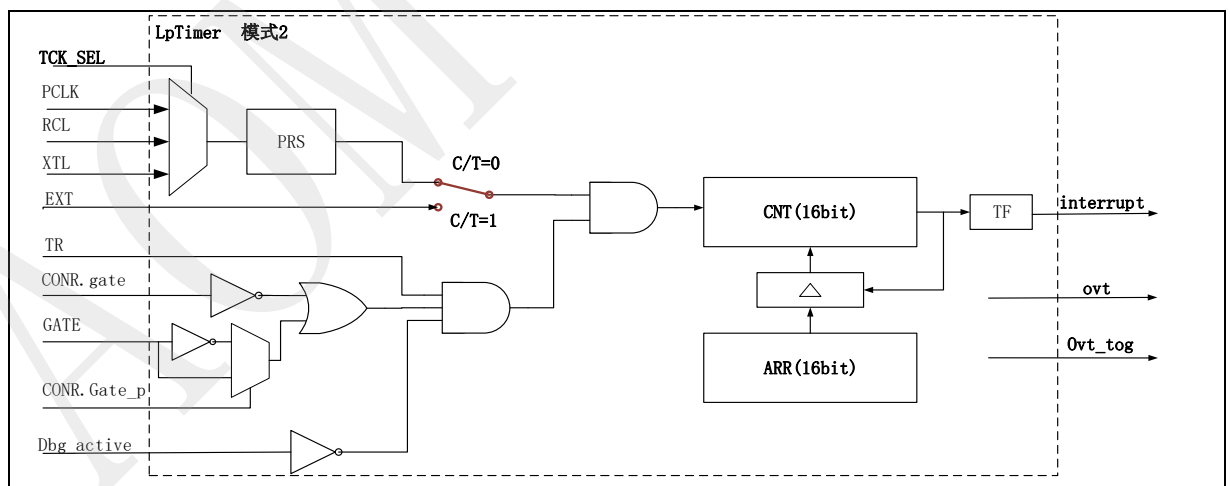


图 31-3 LPTimer 模式 2

当设置对应定时器 TR 为 1 后定时器开始运行。计数器从设定的值开始计数，计数到最大 0xFFFF 后产生溢出中断。模式 1 产生中断后计数器清零继续向上计数，模式 2 产生中断后重载寄存器 ARR 的值到计数器中，向上计数。计数器 CNT 的初值

在启动定时器时由 ARR 自动装载。

31.2.1 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的计数时钟采样，因此外部输入时钟频率不能超过系统的计数时钟。计数到最大值会溢出并且产生中断。

31.2.2 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器一个时钟累加一次，计数到最大值会溢出并且产生中断。

定时时钟周期数为 $(0xFFFF-ARR+0X0001)$

31.3 LPTimer 互连

31.3.1 GATE 互联

GATE 输入可以从端口通过 PxAFR 选择直接输入，也可通过端口功能寄存器 MUXGATE 选择可以连通到其他模块或端口。

当 LPTIM1_GATE=0x0 时，门控 GATE 输入为 PxAFR 选择的端口输入，当 LPTIM1_GATE=0x1~0x7 时，连接其他模块的输入或输出。

000	001	010	011	100	101	110	111
LPTIM1_GT	LPUART1_RX	LPUART2_RX	PCNT_S2	PD15	PD12	PD13	PD14

31.3.2 ETR 互联

EXT 输入可以从端口直接输入，也可通过端口功能寄存器 MUXETR 选择可以连通到其他模块或端口。

当 LPTIM1_ETR=0x0 时，外部时钟 EXT 输入为 PxAFR 选择的端口输入，当 LPTIM1_ETR=0x1~0x7 时，连接其他模块的输入或输出。

000	001	010	011	100	101	110	111
LPTIM1_ETR	LPUART1_RX	LPUART2_RX	PCNT_S2	PE05	PA07	PB07	PC07

31.3.3 Toggle 输出互联

LPTimer 的翻转输出到端口上，可以驱动 Buzzer 实现蜂鸣器的控制。

31.4 LPTimer 寄存器描述

LPTIMER1 基地址：0x40007C00

LPTIMER2 基地址：0x40006C00

寄存器	偏移地址	描述
LPTIMx_CNT	0x000	LPTimer 计数值只读寄存器
LPTIMx_ARR	0x004	LPTimer 重载寄存器
LPTIMx_CR	0x00C	LPTimer 控制寄存器
LPTIMx_IFR	0x010	LPTimer 中断标志
LPTIMx_ICLR	0x014	LPTimer 中断清除寄存器

表 31-1 LPTimer 寄存器列表

31.4.1 计数器计数值寄存器（LPTIMx_CNT）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RO															

位	符号	描述
31:16	Reserved	保留位，读为0
15:0	CNT	低功耗定时器计数值

31.4.2 重载寄存器 (LPTIMx_ARR)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
RW															

位	符号	描述
31:16	Reserved	保留位, 读为0
15:0	ARR	低功耗定时器重载寄存器 写 ARR 前需要读取 CR.WT_FLAG, 当且尽当 WT_FLAG 为1时, 写入时才能写入数据。写ARR 寄存器后 WT2_FLAG 会变低。 注: 在定时器启动时或者重载时这个值会装载到定时器计数器中。定时器计数器值不可软件更改。

31.4.3 控制寄存器 (LPTIMx_CR)

偏移地址: 0x00C

复位值: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	PRS		IE	GATE_P		GATE	WT_FLAG		Reserved	TCK_SEL		TOG_EN		CT	MD	TR
	RW		RW	RW		RW	R			RW		RW		RW	RW	RW

位	符号	描述
31:14	Reserved	保留位, 读为0
13:11	PRS	预除频 (在定时器运行时不可以更改) 000: 1; 001: 2; 010: 4; 011: 8;

		100: 16; 101: 32; 110: 64; 111: 256;
10	IE	中断使能控制, 写1后使能中断
9	GATE_P	GATE 极性控制, 默认高电平 gate 有效, 设置为1后低电平有效
8	GATE	定时器门控 0: 无门控, TR=1 时定时器工作; 1: 只有门控为1 并且 TR=1 时才工作;
7	WT_FLAG	WT, 写同步标志 0: 正在同步, 此时写ARR无效 1: 同步完成, 此时可更改ARR
6	Reserved	保留位, 读为0
5:4	TCK_SEL	LPTimer 时钟选择 00 PCLK; 10:XTL; 11 ,RCL
3	TOG_EN	TOG 输出使能 0: TOG, TOGN 同时输出0 1: TOG, TOGN 输出相位相反的信号。可供 buzzer 使用。
2	CT	计数器/定时器功能选择 0: 定时器功能, 定时器使用 TCK_SEL 选择的时钟进行计数。 1: 计数器功能, 计数器使用外部输入的下降沿进行计数。采样时钟使用 TCK_SEL选择的时钟, 外部输入时钟要低于 1/2 采样时钟。
1	MD	定时器工作模式 0: 模式1无重载模式16位计数器/定时器 1: 模式2自动重载16位计数器/定时器
0	TR	定时器运行控制位 0: 定时器停止 1: 定时器运行

31.4.4 中断标志寄存器 (LPTIMx_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TF	
														RO	

位	符号	描述
31:1	Reserved	保留位, 读为0
0	TF	中断标志, 硬件置位。写清除寄存器清零

31.4.5 中断标志清除寄存器 (LPTIMx_ICLR)

偏移地址: 0x014

复位值: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TFC	
														R1W0	

位	符号	描述
31:1	Reserved	保留位, 读为0
0	TFC	中断标志清除, 写0清除, 写1无效

32. 可编程计数阵列 (PCA)

32.1 PCA 简介

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可用作为一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个模块都可以进行独立编程, 以提供输入捕捉, 输出比较或脉冲宽度调制。另外模块 4 有额外的看门狗定时器模式。

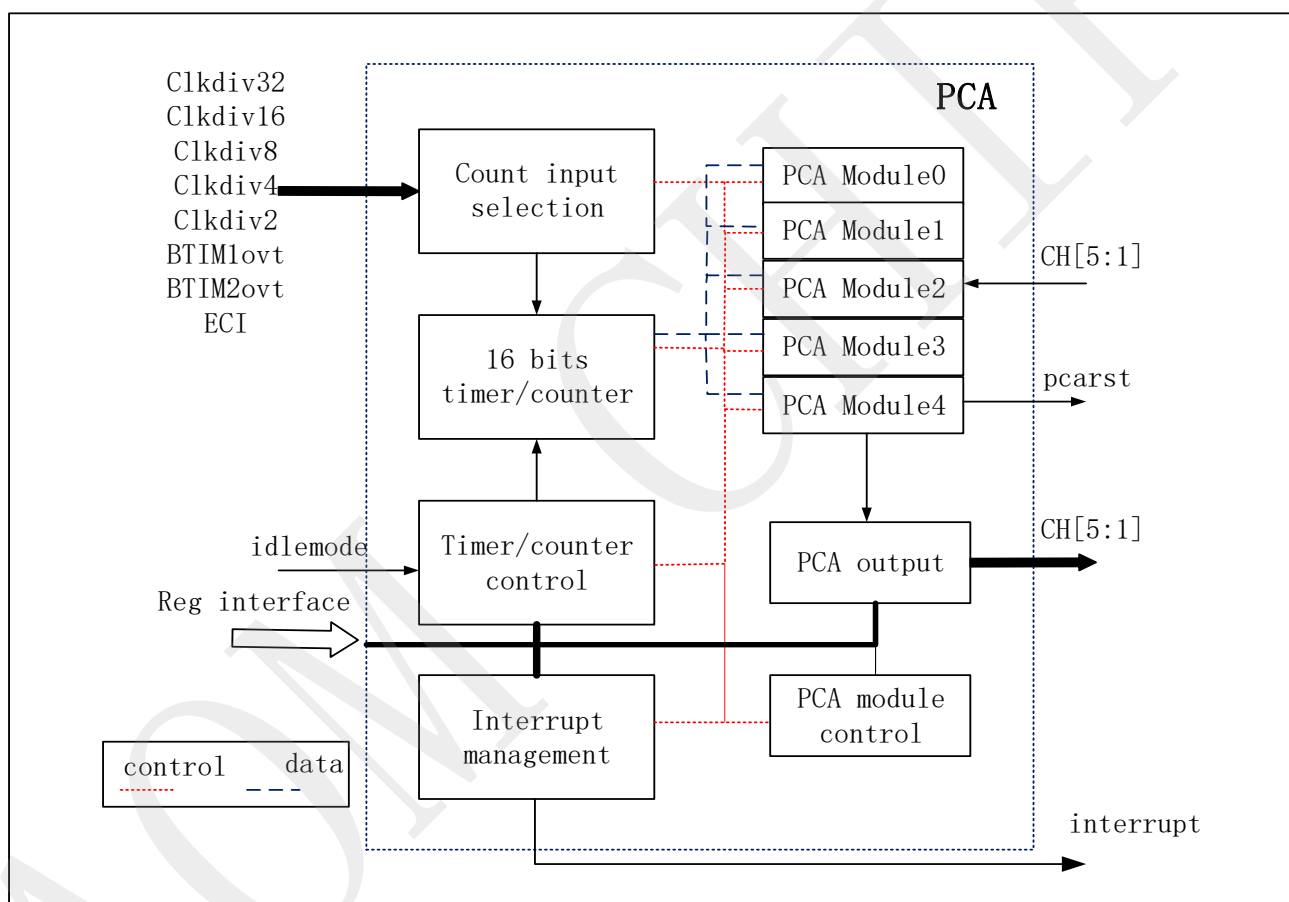


图 32-1 PCA 整体框图

32.2 PCA 功能描述

五个模块都可被配置为独立工作, 有三种工作方式: 边沿触发捕捉、输出比较、8/16 位脉宽调制。每个模块在系统控制器中都有属于自己的功能寄存器, 这些寄存器用于配置模块的工作方式和与模块交换数据。

每组比较/捕获模块是由一个比较/捕获寄存器组(CCAPx), 以及 1 个 16 位比较器, 和

各种逻辑门控制组成。寄存器组用来存储时间或次数，针对外部触发捕获条件，或内部触发比较条件。在 8 位 PWM 模式下，寄存器(CCAPxL)用来控制输出波形的占空比，CCAPxH 为 8 位比较缓存。16 位 PWM 模式下 CARR 用于控制 PWM 输出的周期，CCAPx 寄存器控制占空比。

每个模块都可以独立编程的操作在任何以下模式：

- 16 位捕获模式的上升沿，下降沿或任意沿触发。
- 比较模式：16 位高速输出，16 位看门狗定时器（模块 4）或 16/8 位脉冲宽度调制。
- 关闭状态。

比较/捕获模块模式寄存器（CCAPMx）确定相应的工作模式。对于比较/捕获模块进行编程时，他们是基于共同的时间计数。定时器/计数器打开和关闭通过 CCON.CR 位即可控制 PCA 定时/计数器的运行。在一个比较/捕获模块捕获，软件定时器，高速输出，设置模块的比较/捕获标志（CCON.CCFx），并产生 PCA 中断请求，如果相应的使能位在 CCAPMx 寄存器设置。CPU 可以在任何时候读写 CCAPx 寄存器。

32.2.1 PCA 定时/计数器

CNT 的这组特殊功能寄存器可作为一个 16 位定时器/计数器。这是一个 16 位向上计数的计数器。如果 CMOD.CFIE 位被置“1”时，当 CNT 溢出时硬件自动设置 PCA 溢出标志（CCON.CF）并产生 PCA 中断请求。CMOD.CPS[2: 0]三位选择八个信号输入到定时器/计数器。

- 系统时钟 PCLK 的 32 分频。
- 系统时钟 PCLK 的 16 分频。
- 系统时钟 PCLK 的 8 分频。
- 系统时钟 PCLK 的 4 分频。
- 系统时钟 PCLK 的 2 分频。
- 定时器 0 的溢出。每次定时器 0 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- 定时器 1 的溢出。每次定时器 1 计数溢出后，CNT 就递增，这样提供了 PCA 的

可变编程频率输入。

- ECI。CPU 每过 4 个 PCLK 时钟周期就对 PCA ECI 进行采样，当每次采样结果从高变低时，CL 自动加 1，因此最高的 ECI 输入频率不能高于系统时钟 PCLK 的 1/8,以满足采样需求。

设置运行控制器 (CCON.CR) 启动 PCA 定时/计数器。当 CMOD.CIDL 置“1”后，PCA 定时器/计数器可以继续运行在空闲模式下。CPU 可以随时读取 CNT 的数值，但当计数启动后 (CCON.CR=1) 时，为了防止计数错误，CNT 是禁止写入的。

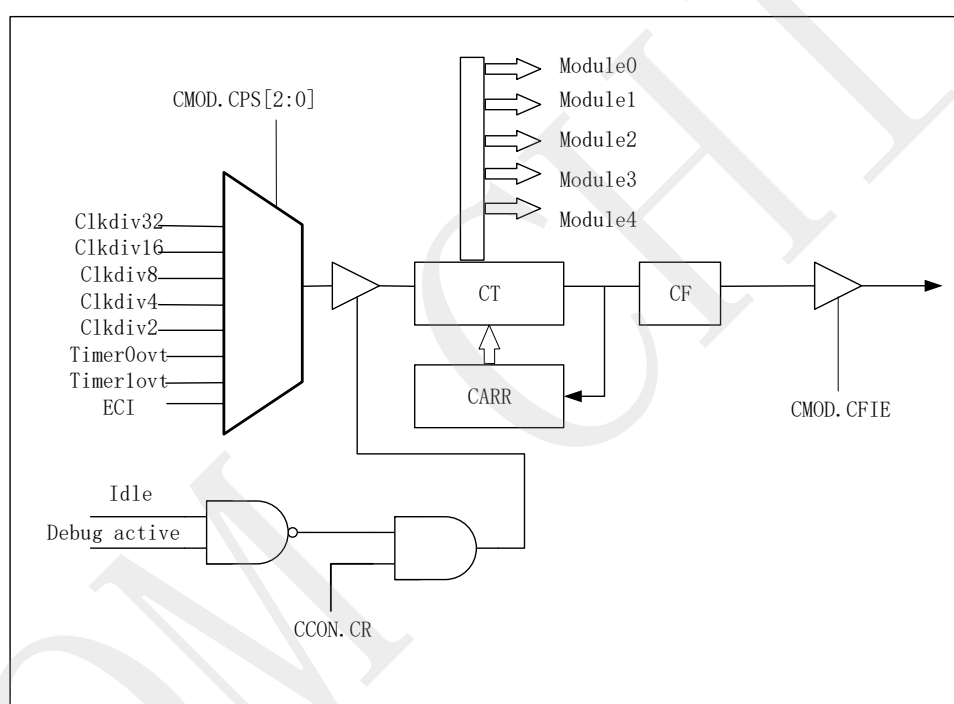


图 32-2 PCA 计数器框图

32.2.1.1 16 位自由计数模式

计数器计数到最大值 0xFFFF 溢出后计数器的值变为 0，重新开始向上计数，如果需要更改计数周期值可以停止 PCA 更改计数器的初值然后继续计数。可用于 PCA 捕获模式，8 位 PWM 模式。

设置流程

1. 保持 PCA_EPWM 的 EPMW 为 0
2. 设置 PCA_CM0D.CPS 选择计数时钟
3. 根据需要设置 PCA_CM0D.CFIE 设置计数溢出中断使能

4. 时钟 PCA_CCON.CR 启动 PCA 计数器
5. 溢出更改计数器初值需要停止 PCA 计数器

32.2.1.2 16 位重载计数模式

计数器计数到与寄存器 CARR 的值相同时溢出后计数器的值变为 0, 继续开始向上计数, 可用于 PCA 捕获模式, 16 位 PWM 模式。

设置流程

1. 设置 PCA_EPWM 的 EPMW 为 1
2. 设置 PCA_CARR 设置计数周期值
3. 设置 PCA_CMODO.CPS 选择计数时钟
4. 根据需要设置 PCA_CMODO.CFIE 设置计数溢出中断使能
5. 时钟 PCA_CCON.CR 启动 PCA 计数器

32.2.2 PCA 捕获功能

PCA 捕获模式提供了 5 路 PCA 测量脉冲周期, 脉冲宽度, 占空比和相位差的功能。引脚上出现的电平跳变导致 PCA 捕捉 PCA 计数器/定时器的值并将其装入到对应模块的 16 位捕捉/比较寄存器 (CCAPx)。CCAPMx.CAPP 以及 CCAPMx.CAPN 位用于选择触发捕捉的电平变化类型: 低电平到高电平 (正沿)、高电平到低电平 (负沿) 或任何变化 (正沿或负沿)。当捕捉发生时, CCON 中的捕捉/比较标志 (CCFn) 被置为逻辑‘1’并产生一个中断请求 (如果 CCF 中断被允许)。当 CPU 转向中断服务程序时, CCFn 位不能被硬件自动清除, 用户软件写 INTCL 寄存器清除此标志位。如果 CCPMx.CAPP 以及 CCAPMx.CAPN 位都被设置为逻辑‘1’, 可以通过直接读对应端口引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。

分辨率等于定时器/计数器的时钟。输入信号必须在高电平或低电平期间至少保持 2 个时钟周期, 以保证输入信号能够被硬件识别。

CPU 可以在任何时候读取或写入 CCAPx 的寄存器。

捕获设置:

- 当需要在外部上升沿进行捕获, CCPMx.CAPP = “1” 以及 CCAPMx.CAPN = “0”
- 当需要在外部下降沿进行捕获, CCPMx.CAPP = “0” 以及 CCAPMx.CAPN = “1”

- 当需要在外部上升、下降沿进行捕获, $CCPMx.CAPP = "1"$ 以及 $CCAPMx.CAPN = "1"$
- 根据需要配置捕获中断及中断处理程序。
- 按照定时/计数器配置 PCA 计数器启动。

注意:

- 随后由同一模块的捕获值会覆盖现有捕获的值。为了保持捕获的值, 在中断服务程序中将它保存在 RAM 里面, 这个操作必须在下一次事件出现之前完成, 否则就会丢失前面一次捕获采样值。

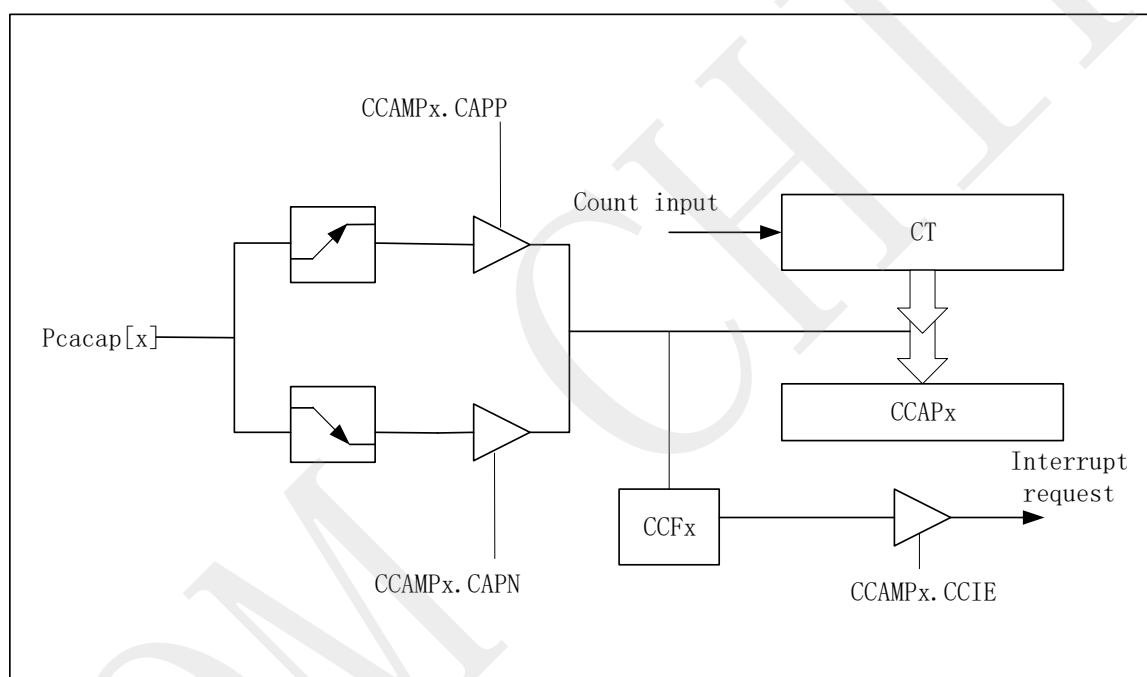


图 32-3 PCA 捕获功能框图

32.2.3 PCA 比较功能

比较功能提供如下功能, 高速输出模式, WDT 模式, 16 为 PWM 模式和 8 位 PWM 模式。在前三个功能中, 比较/捕获模块比较 16 位 PCA 定时器/计数器的值与预先加载到该模块的 $CCAPx$ 寄存器中的 16 位值。在 8 位 PWM 模式下, PCA 模块不断地将 PCA 定时器/计数器低字节寄存器 (CNT) 与一个在 $CCAPxL$ 模块寄存器 8 位的值进行比较。每 4 个时钟周期比较一次, 即与最快的 PCA 定时器/计数器的时钟速率相匹配。

设置 CCAPMx.ECOM 位选择该模块的比较功能。

若要正确使用在比较模式下的模块，请遵守以下的一般程序：

- 选择 PCA 模块的操作模式。
- 选择 PCA 定时器/计数器的输入信号。
- 比较值加载到模块的比较/捕获寄存器对。
- 设置 PCA 定时器/计数器运行控制位。
- 匹配后产生中断，清除模块的比较/捕获标志。

32.2.3.1 比较翻转输出模式

在比较翻转输出模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器（CCAPx）发生匹配时，模块 PCA 的 CH[x]引脚上的逻辑电平将发生变化。这可以提供比切换 IO 输出有更高精度，因为这个输出不会被中断响应而影响输出频率，靠 CPU 来切换 IO 输出的话，功耗，精度都有所欠缺。

要设定一个比较/捕获模块的比较翻转输出模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位。PCA 定时器/计数器和比较/捕获的寄存器（CCAPx）之间的匹配切换 PCA 的 CH[x]信号，并设置模块的比较/捕获标志（CCON.CCFx）。通过软件设置或清除 PCA 的 CH[x]信号，用户可以选择匹配切换信号从低到高或高到低。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位（CCAPMx.CCIE）当匹配发生时，即可产生中断请求。由于硬件无法清除的比较/捕获标志中断，用户必须在软件中清除这个标志位。如果用户在中断程序中不去改变比较/捕获寄存器，PCA 并重新计数比较值，如相匹配则发生下一次翻转。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器（CCAPx）。

注意：

- 为了防止无效的匹配，而更新这些寄存器，用户软件应该写 CCAPxL 的首先然后 CCAPxH。写到 CCAPxL 清除禁用比较功能 ECOM 位，而写到 CCAPxH 设置的 ECOM 位，重新启用比较功能。

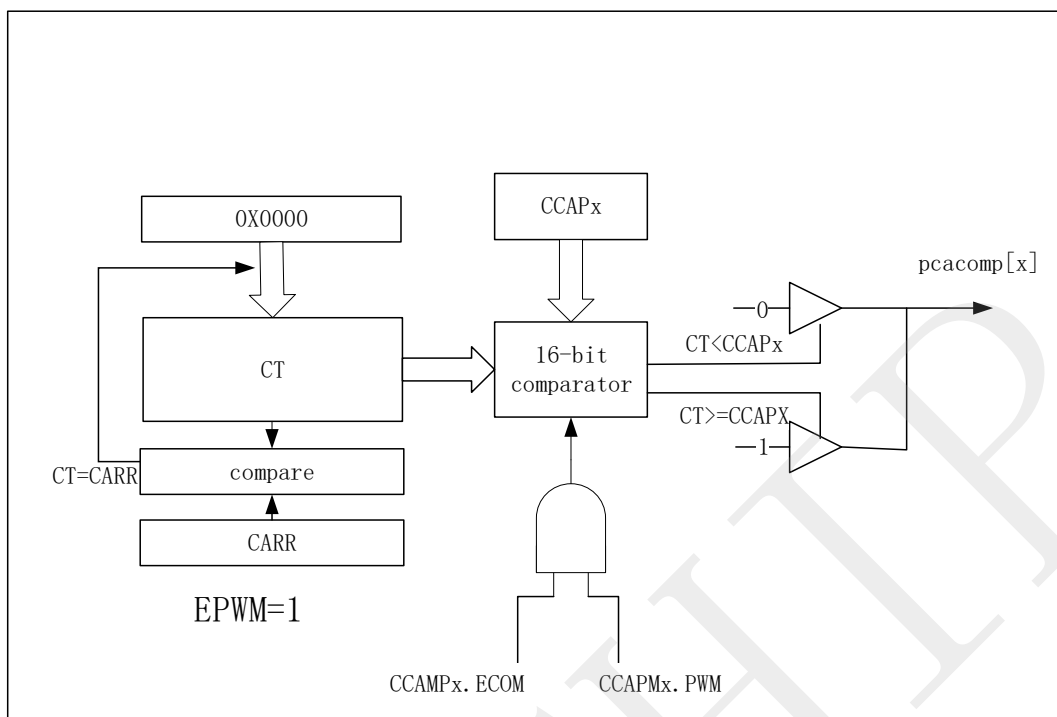


图 32-5 PCA 16 位 PWM 功能框图

32.2.3.3 PCA 8 位 PWM 功能

脉宽调制是一种使用程序来控制波形占空比，周期，相位的技术。5 个 PCA 模块都可以被独立地用于在对应 PCA 的 CH[x]引脚产生脉宽调制（PWM）输出，脉冲宽度为 8 位分辨率。PWM 输出的频率取决于 PCA 计数器/定时器的时基。使用模块的捕捉/比较寄存器 CCAPxL 来改变 PWM 输出信号的占空比。当 PCA 计数器/定时器的低字节（CL）与 CCAPxL 中的值相等时，PCA 的 CH[x]引脚上的输出被置“1”；当 CL 中的计数值溢出时，PCA 的 CH[x]输出被复位“0”。当计数器/定时器的低字节 CL 溢出时（从 0xFF 到 0x00），保存在 CCAPxH 中的值被自动装入到 CCAPxL，不需软件干预。

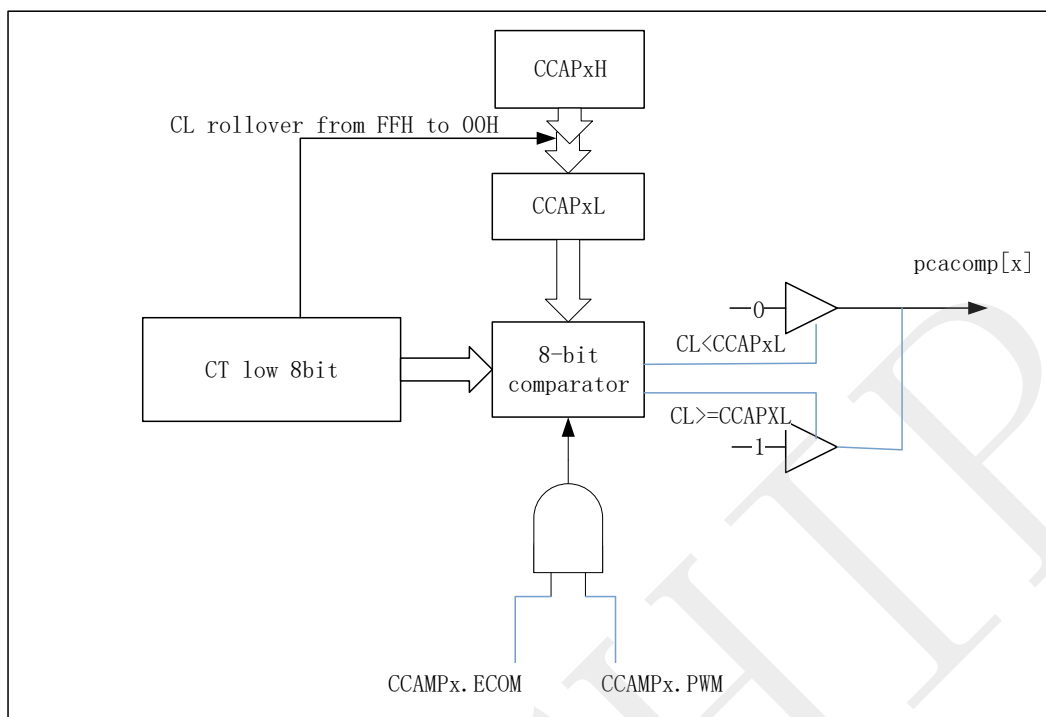


图 32-6 PCA PWM 功能框图

在这种模式下，PCA 定时器/计数器（CL）的低字节中的值是不断在低字节比较/捕获寄存器（CCAPxL）的值相比。当 $CL < CCAPxL$ ，输出波形为低。当两者匹配时（ $CL = CCAPxL$ ），输出波形去到高，直到 CL 溢出从 FFH 到 00H，结束期间仍然很高。在溢出时，在 CCAPxH 的值自动装载到 CCAPxL 内，一个新的周期的开始。在 CCAPxL 的值决定当前波形的占空比。在 CCAPxH 的值确定下一个波形的占空比。改变 CCAPxL 中的值即可更改的脉冲宽度调制。正如图所示，8 位值在 CCAPxL 可以从 0（100% 占空比），到 255（0.4% 占空比）。要改变 CCAPxL 值而不会产生毛刺，需要在高字节寄存器（CCAPxH）写入一个新值。当 CL 超过 FFH 滚动到 00h，这个值是由硬件自动加载到 CCAPxL。

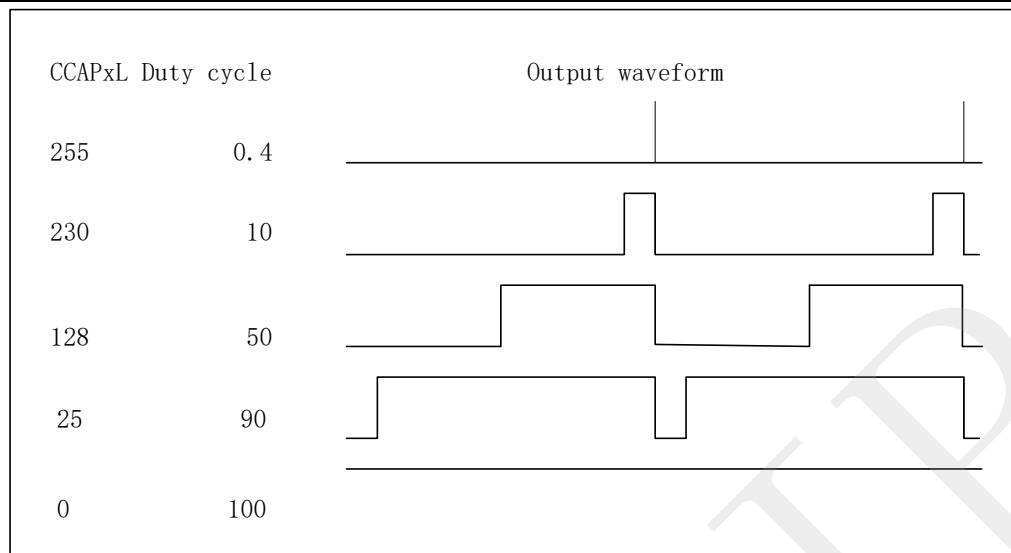


图 32-7 PCA PWM 输出波形

要设定一个比较/捕获模块在 PWM 模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.PWM 位。另外 PCA 定时器/计数器由编程 CMOD.CSP[2:0]可以选择输入计数信号频率。在 CCAPxL 输入一个 8 位的值指定第一个 PWM 波形的占空比。在 CCAPxH 输入一个 8 位的值会指定第二个 PWM 波形的占空比。设置定时器/计数器运行控制位 (CCON.CR) 启动 PCA 定时器/计数器。

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	EPWM	工作方式
X	1	0	0	0	0	X	0	用正沿触发捕获
X	0	1	0	0	0	X	0	用负沿触发捕获
X	1	1	0	0	0	X	0	用跳变沿触发捕获
1	0	0	1	0	0	X	0	软件定时器
1	0	0	1	1	0	X	0	高速输出
1	0	0	0	0	1	X	0	8位脉冲宽度调制器
1	0	0	1	1	0	X	1	16位脉冲宽度调制器

表 32-1 PCA 比较捕获功能模块设置

32.3 PCA 寄存器描述

基地址 0x40001800

寄存器	偏移地址	描述
PCA_CCON	0x000	PCA 控制寄存器
PCA_CM0D	0x004	PCA 模式寄存器
PCA_CNT	0x008	PCA 计数寄存器
PCA_INT_CL	0x00C	PCA 中断清除寄存器
PCA_CCAPM0	0x010	PCA 比较/捕获模块0模式寄存器
PCA_CCAPM1	0x014	PCA 比较/捕获模块1模式寄存器
PCA_CCAPM2	0x018	PCA 比较/捕获模块2模式寄存器
PCA_CCAPM3	0x01C	PCA 比较/捕获模块3模式寄存器
PCA_CCAPM4	0x020	PCA 比较/捕获模块4模式寄存器
PCA_CCAP0H	0x024	PCA 比较/捕获模块0高8位寄存器
PCA_CCAP0L	0x028	PCA 比较/捕获模块0低8位寄存器
PCA_CCAP1H	0x02C	PCA 比较/捕获模块1高8位寄存器
PCA_CCAP1L	0x030	PCA 比较/捕获模块1低8位寄存器
PCA_CCAP2H	0x034	PCA 比较/捕获模块2高8位寄存器
PCA_CCAP2L	0x038	PCA 比较/捕获模块2低8位寄存器
PCA_CCAP3H	0x03C	PCA 比较/捕获模块3高8位寄存器
PCA_CCAP3L	0x040	PCA 比较/捕获模块3低8位寄存器
PCA_CCAP4H	0x044	PCA 比较/捕获模块4高8位寄存器
PCA_CCAP4L	0x048	PCA 比较/捕获模块4低8位寄存器
PCA_CCAP0	0x04C	PCA PWM与高速输出标志寄存器
PCA_CCAP0	0x050	PCA 比较/捕获模块0 的16位寄存器
PCA_CCAP1	0x054	PCA 比较/捕获模块1 的16位寄存器
PCA_CCAP2	0x058	PCA 比较/捕获模块2 的16位寄存器
PCA_CCAP3	0x05C	PCA 比较/捕获模块3 的16位寄存器
PCA_CCAP4	0x060	PCA 比较/捕获模块4 的16位寄存器
PCA_CARR	0x064	PCA 周期装载寄存器
PCA_EPWM	0x068	PCA PWM增强寄存器

表 32-2 PCA 寄存器列表

32.3.1 控制寄存器（PCA_CCON）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CF	CR	Reserved	CCF4	CCF3	CCF2	CCF1	CCF0
								RO	RW		RO	RO	RO	RO	RO

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志（写无效） 当PCA计数溢出时，CF由硬件置位，如果CMOD寄存器的CFIE位为1，则CF标志可以产生中断 1：发生计数器溢出；0：无溢出；
6	CR	PCA计数器运行控制位 1：启动PCA计数器计数 0：关闭PCA计数器计数
5	Reserved	保留位
4	CCF4	PCA计数器模块4 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM4.CCIE置位时，这个标志位会产生一个PCA中断
3	CCF3	PCA计数器模块3 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM3.CCIE置位时，这个标志位会产生一个PCA中断
2	CCF2	PCA计数器模块2 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM2.CCIE置位时，这个标志位会产生一个PCA中断
1	CCF1	PCA计数器模块1 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM1.CCIE置位时，这个标志位会产生一个PCA中断
0	CCF0	PCA计数器模块0 比较/捕获标志位 当出现匹配或捕获时，该位由硬件置位。（写无效） 当CCAPM0.CCIE置位时，这个标志位会产生一个PCA中断

32.3.2 模式寄存器 (PCA_CMOD)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CIDL	WDTE	Reserved		CPS[2:0]		CFIE	
								RW	RW			RW		RW	

位	符号	描述
31:8	Reserved	保留位
7	CIDL	空闲模式IDLE下, PCA是否停止工作 1: 休眠模式 (sleep) 下, PCA停止工作 0: 休眠模式 (sleep) 下, PCA继续工作
6	WDTE	PCA WDT功能使能控制位 1: 启动PCA模块4 WDT功能 0: 关闭PCA模块4 WDT功能
5:4	Reserved	保留位
3:1	CPS[2:0]	时钟分频选择及时钟源选择 000: PCLK/32 001: PCLK/16 010: PCLK/8 011: PCLK/4 100: PCLK/2 101: timer0 overflow 110: timer1 overflow 111: ECI 外部时钟, 时钟PCLK四分频采样
0	CFIE	PCA计数器中断使能控制信号 1: 使能中断 0: 关闭中断

32.3.3 计数寄存器（PCA_CNT）

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	定时器计数器的值 只有在PCA停止状态CNT才可以写入。否则写入无效

32.3.4 中断清除寄存器（PCA_ICLR）

偏移地址：0x00C

复位值：0x0000 009F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CF	Reserved		CCF4	CCF3	CCF2	CCF1	CCF0
								R1W0			R1W0	R1W0	R1W0	R1W0	R1W0

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志清除（软件写0清零，写1无效），读出值为1
6:5	RSV	保留位
4	CCF4	PCA计数器模块4 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
3	CCF3	PCA计数器模块3 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
2	CCF2	PCA计数器模块2 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
1	CCF1	PCA计数器模块1 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1
0	CCF0	PCA计数器模块0 比较/捕获标志位清除（软件写0清零，写1无效），读出值为1

32.3.5 比较捕获模式寄存器 (PCA_CCAPM0~4)

偏移地址

CCAPM0: 0x010; CCAPM1: 0x014; CCAPM2: 0x018;

CCAPM3: 0x01C; CCAPM4: 0x020;

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ECOM	CAPP	CAPN	MAT	TOG	PWM	CCIE
									RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:7	Reserved	保留位
6	ECOM	允许比较器功能控制位 1: 允许比较器功能; 0: 禁止比较器功能; 当PCA用于软件计数器, 高速输出, PWM模式, WDT模式, 要置位ECOM 写CCAMPHx或CCAMPx寄存器会自动置位ECOM; 写CCAMPLx寄存器会自动清除ECOM位
5	CAPP	正沿捕获控制位 1: 允许上升沿捕获; 0: 禁止上升沿捕获
4	CAPN	负沿捕获控制位 1: 允许下降沿捕获; 0: 禁止下降沿捕获
3	MAT	允许匹配控制位 1: PCA计数值与模块的比较/捕获寄存器的值一旦匹配, 将置位CCON寄存的中断标志CCFx (x=0-4) 0: 禁止匹配功能
2	TOG	翻转控制位 1: 工作在PCA高速输出模式, PCA计数器的值与模块的比较/捕获寄存器的值一旦匹配, CCPx引脚翻转 0: 禁止翻转功能
1	PWM	脉宽调制控制位 1: 允许CCPx引脚作为PWM输出 0: 禁止PWM脉宽调制功能 只有CCAPMx[6:0]=100_0010时, PWM功能才有效
0	CCIE	PCA使能中断 1: 使能比较/捕获中断 0: PCA比较/捕获功能中断禁止

32.3.6 比较捕获数据寄存器高 8 位 (PCA_CCAP0~4H)

偏移地址

CCAP0H: 0x024; CCAP1H: 0x02C; CCAP2H: 0x034;

CCAP3H: 0x03C; CCAP4H: 0x044;

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CCAPx[15:8]							
								RW							

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[15:8]	比较/捕获模式高8位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值的高8位；写CCAPxH寄存器会自动置位寄存器CCAPMx的ECOM位。 当PCA模式用于PWM模式时，用于控制输出占空比装载寄存器，在计数器低8位溢出时，装载寄存器会自动更新到PWM比较寄存器

32.3.7 比较捕获数据寄存器低 8 位 (PCA_CCAP0~4L)

偏移地址

CCAP0L: 0x028; CCAP1L: 0x030; CCAP2L: 0x038;

CCAP3L: 0x040; CCAP4L: 0x048;

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CCAPx[7:0]							
								RW							

位	符号	描述
31:8	Reserved	保留位

7:0	CCAPx[7:0]	比较/捕获模式低8位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值的低8位；写CCAPxL寄存器会自动清除寄存器CCAPMx的ECOM位。 当PCA模式用于PWM模式时，用于控制输出占空比较寄存器，在PWM模式，计数器的低8位的值小于CCAPx[7:0]的值PWM输出低电平，否则PWM输出高电平。

32.3.8 比较捕获 16 位寄存器（PCA_CCAP0~4）

偏移地址

CCAP0: 0x050; CCAP1: 0x054; CCAP2: 0x058;

CCAP3: 0x05C; CCAP4: 0x060;

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCAPx[15:0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCAPx[15:0]	比较/捕获模式16位寄存器 当PCA模式用于比较/捕获模式时，用于保存16位捕获计数值；写CCAPx寄存器会置位寄存器CCAPMx的ECOM位。 写CCAPX寄存器相当于写CCAPxL及CCAPxH这两个8位寄存器。在比较/捕获模式下可以直接读写这个寄存器，在PWM模式下，使用CCAPxL及CCAPxH寄存器

32.3.9 比较高速输出标志寄存器 (PCA_CCAPO)

偏移地址: 0x04C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7				6				5				4				3				2			
Reserved				CCAPO4				CCAPO3				CCAPO2				CCAPO1				CCAPO0			
				RW				RW				RW				RW				RW			

位	符号	描述
31:5	Reserved	保留位
4	CCAPO4	比较模块4的输出值
3	CCAPO3	比较模块3的输出值
2	CCAPO2	比较模块2的输出值
1	CCAPO1	比较模块1的输出值
0	CCAPO0	比较模块0的输出值

32.3.10 周期寄存器 (PCA_CARR)

偏移地址: 0x64

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15															
CARR[15:0]															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CARR	计数周期重载寄存器

32.3.11 增强 PWM 控制 (PCA_EPWM)

偏移地址: 0x68

复位值: 0x0000 0000

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

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														EPWM	
														RW	

位	符号	描述
31:1	Reserved	保留位
0	EPWM	16 bit PWM 使能

33. 实时时钟（RTC）

33.1 实时时钟简介

实时时钟/日历提供秒、分、时、日、周、月、年的信息，每月的天数和闰年的天数可自动调整。时钟操作可通过 AM/PM 寄存器位，决定采用 24 或 12 小时格式。表 33-1 所示是其基本特性。具有最小周期为 0.5S 的周期中断，如果需要更短的中断周期请使用 LPTIM 产生中断。

时钟源	片外低速晶振 XTL (32.768kHz) 片内低速振荡器 RCL (32kHz, 1%精度) 片外高速晶振 XTH
基本功能	可计算 00~99 年之间的秒、分、时、日、周、月、年
	可自动进行闰年调整
	可配置为 24 或 12 小时格式
	可程序控制启动或停止
	具有闹钟功能
	具有高精度 1Hz 方波信号输出
中断	具有最小周期为0.5S的周期中断
	具有精确到秒的闹钟中断

表 33-1 RTC 的基本特性

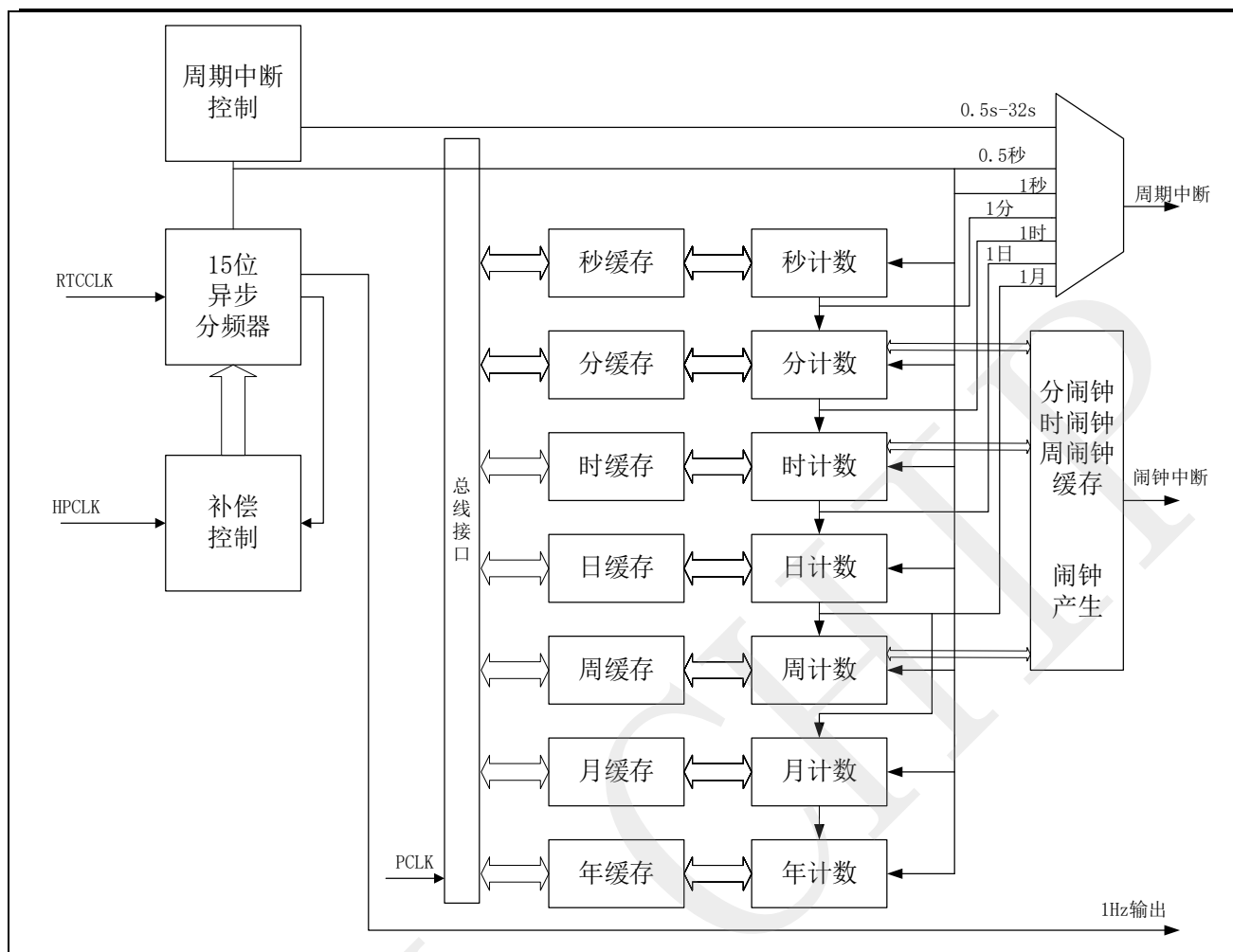


图 33-1 RTC 框图

33.2 实时时钟功能描述

实时时钟的时钟源可配置为外部低速晶振，外部高速晶振，内部低速 RC；默认使用外部低速晶振。控制寄存器 RTC_CR0、RTC_CR1 与 RTC_COMPEN 只受上电复位控制，其他复位源不能复位这三个控制寄存器。其他数据寄存器上电状态不定，上电后需要初始化，不受任何复位影响。

所有软件写入和读取的日期时间值都为 BCD 码，无须十六进制转换为十进制。

任何无效的日期时间将无法写入，比如 32 日，25 时，70 秒，B 月等。

33.2.1 上电设定

RTC 在上电之后复位一次，在系统不掉电的情况下，外部各种复位请求都不能复位 RTC，RTC 会一直处于计数状态。在上电之后，设定日历初始值、闹钟设置、误差补偿、中断等之后，启动 RTC。

33.2.2 RTC 计数开始设定

1. 设定 `RTC_CR0.START=0`，计数停止；
2. 设定 `RTC_CR0.AMPM` 和 `RTC_CR0.PRDS`，`RTC_CR0.PRDX` 设定定时制和中断周期；
3. 设定 `RTC_CR1.CKSEL` 选择 RTC 的计时时钟；
4. 设定秒，分，时，周，日，月，年的日历计数寄存器；
5. 需要进行时钟误差补偿时，设定计数时钟误差补偿寄存器 `RTC_COMPEN`；
6. 清除中断标志位 `RTC_CR1.ALMF`，`RTC_CR1.PRDF`，并使能中断；
7. 设定 `RTC_CR0.START=1`，计数开始。

33.2.3 系统低功耗模式切换

在 RTC 计数开始后，系统如果立即切换为低功耗模式时，请执行下列任意一种确认后再进行模式切换。

控制寄存器在系统控制寄存器 `SYSCTRL1.RTC_LPW`

在 `RTC_CR0.START=1` 设定后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。

在 `RTC_CR0.START=1` 设定后，设定 `RTC_CR1.WAIT=1`，查询 `RTC_CR1.WAITF=1`。再设定 `RTC_CR1.WAIT=0`，查询 `RTC_CR1.WAITF=0` 后再进行模式切换。

在 RTC 低功耗模式下，RTC 寄存器不能读写。在低功耗模式下，RTC 消耗更少的电流。

RTC 运行时切换低功耗模式不需要等待。

33.2.4 读出计数寄存器

有三种读取计数寄存器的方式：

方式 1：任意时刻读取方式 1

1. 设定 `RTC_CR1.WAIT=1`，停止日历寄存器计数，进入读写模式；
2. 查询直到 `RTC_CR1.WAITF=1`；
3. 读出秒，分，时，周，日，月，年计数寄存器值；
4. 设定 `RTC_CR1.WAIT=0`，计数器计数；

5. 查询直到 RTC_CR1.WAITF=0。

方式 2：任意时刻读取方式 2

1. 读出分，时，周，日，月，年计数寄存器值；
2. 读出秒计数寄存器值；
3. 再次读出秒计数寄存器值；
4. 判断两次秒的读出值是否相同，不同重新从第一步开始，相同读取结束。

方式 3：中断读取方式

在 RTC 周期中断服务中读取秒，分，时，周，日，月，年计数寄存器值。因为中断发生后到下数据改变至少 0.5s 的时间。

33.2.5 写入计数寄存器

1. 设定 RTC_CR1.WAIT=1，停止日历寄存器计数，进入读写模式；
2. 查询直到 RTC_CR1.WAITF=1；
3. 写入秒，分，时，周，日，月，年计数寄存器值；
4. 设定 RTC_CR1.WAIT=0，计数器重新开始计数。注意：须在 1 秒内完成所有写操作；
5. 查询直到 RTC_CR1.WAITF=0。

在 RTC 未启动模式下写秒，分，时，周，日，月，年计数寄存器不需要等 WAIT。
注意：

- 在计数模式下更改秒寄存器会复位秒计数，写分，时，周，日，月，年计数寄存器值不会影响 RTC 计数。

33.2.6 闹钟设定

1. 设定 RTC_CR1.ALMEN=0，闹钟禁止；
2. 设定 RTC_CR1.ALMIE=1，闹钟中断许可；
3. 秒闹钟 RTC_ ALMSEC，分闹钟 RTC_ ALMMIN，时闹钟 RTC_ ALMHOUR，周闹钟 RTC_ ALM WEEK 设定；
4. 设定 RTC_CR1.ALMEN=1，闹钟许可；
5. 等待发生中断；

6. 由于闹钟中断和定周期中断共用中断请求信号，则当 RTC_CR1.ALMF=1 时，进入闹钟中断处理；否则进入定周期中断处理。

33.2.7 1Hz 输出

RTC 可选择输出一般精度，较高精度和高精度 3 种 1Hz 时钟。当时钟误差补偿功能有效时输出较高精度的 1Hz 时钟；当使用不同频率的 PCLK 时输出高精度的 1Hz 时钟。需要根据 PCLK 频率配置系统控制寄存器，其中，

一般精度的 1Hz 输出设定如下：（无时钟补偿）

1. 设定 RTC_CR0.START=0，计数停止；
2. RTC 输出引脚设定；
3. RTC_CR0.1HZOE=1，时钟输出许可；
4. 设定 RTC_CR0.START=1，计数开始；
5. 等待 2 个计数周期以上；
6. 1Hz 输出开始。

较高精度的 1Hz 输出设定如下：（低速补偿）

1. 设定 RTC_CR0.START=0，计数停止；
2. RTC 输出引脚设定；
3. RTC_CR0.1HZOE=1，时钟输出许可；
4. 时钟误差补偿寄存器 RTC_COMPEN.CR 补偿数设定；
5. 时钟误差补偿寄存器 RTC_COMPEN.EN=1，误差补偿有效；
6. 设定 RTC_CR0.START=1，计数开始；
7. 等待 2 个计数周期以上；
8. 1Hz 输出开始。

当高精度的 1Hz 输出时，需要在较高精度输出的基础上，为 RTC 提供 4M,6M,8M,12M,16M,20M, 24M,32MHz 的高速 PCLK 时钟，输出设定如下：

1. 设定 RTC_CR0.START=0，计数停止；
2. RTC 输出引脚设定；
3. RTC_CR0.1HZOE=1，时钟输出许可；

4. RTC_CR0.1HZSEL=1, 选择输出高精度 1Hz 时钟;
5. 配置高速时钟补偿时钟 SYSCTRL1.RTC_FREQ_ADJUST
6. 时钟误差补偿寄存器 RTC_COMPEN.CR[8:0] 补偿数设定;
7. 时钟误差补偿寄存器 RTC_COMPEN.EN=1, 精度补偿有效;
8. 设定 RTC_CR0.START=1, 计数开始;
9. 等待 2 个计数周期以上;
10. 1Hz 输出开始。

33.2.8 时钟误差补偿

由于外部晶振存在误差, 在需要得到高精度的计数结果时, 需要对该误差进行补偿。补偿方法分为两种: 第一种, 基于自身时钟的误差补偿; 第二种, 基于高速时钟的误差补偿。

基于自身时钟的误差补偿原理与计算:

由于计数器采用 32.768kHz 的时钟计数, 如果需要对每秒精度进行补偿时, 只能按照 32.768kHz 的整数周期补偿, 则每秒补偿的最小单位为 $(1/32768) \times 10^6 = 30.5\text{ppm}$, 无法满足高精度的要求。

那么要在 32.768kHz 的计数时钟下实现精度较高的时钟补偿时, 需要在算法上做调整, 将最大补偿周期扩大 32 倍。则在只能补偿的最小单位为 30.5ppm 的情况下, 平均到每秒的补偿单位变为 $30.5\text{ppm}/32 = 0.96\text{ppm}$ 。满足了精度较高的时钟补偿要求。而且补偿发生在每 32 秒内比较均匀的范围。所以, 该寄存器中引入了 5 位小数的设定。

例 1:

当默认状态下直接输出 1Hz 时钟, 通过测定该时钟的精度, 计算补偿目标值。

假设实际测定值为 0.9999888Hz, 则:

$$\text{实际发振频率} = 32768 \times 0.9999888 \approx 32767.63$$

$$\begin{aligned}\text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} \times 10^6 \\ &= (32767.96 - 32768) / 32768 \times 10^6 \\ &= -11.29\text{ppm}\end{aligned}$$

根据

$$CR[8:0] = \left(\frac{\text{补偿目标值[ppm]} \times 2^{15}}{10^6} \right)_{\text{取 2 的补码}} + 0001.00000 \text{ B}$$

如果补偿目标值为 -11.29ppm，计算相应的寄存器值如下：

$$\begin{aligned} CR[8:0] &= (-11.29 \times 215/106)_{\text{取 2 的补码}} + 0001.00000\text{B} \\ &= (-0.37)_{\text{取 2 的补码}} + 0001.00000\text{B} \\ &= 1111.10101\text{B} + 0001.00000\text{B} \\ &= 0000.10101\text{B} \end{aligned}$$

基于高速 24MHz 时钟的误差补偿原理与计算：

该方式的计算方法与基于自身时钟的误差补偿相同。由于引入了 4M-32MHz 高速时钟，本来需要在最多 32 秒内累计的 1/32768 秒误差可分散到每 1 秒，针对每 1 秒进行最小 0.96ppm（23 个 24MHz 时钟周期）的补偿，实现平均的每秒高精度 1Hz 时钟输出。

33.3 RTC 中断

RTC 支持两种中断类型。闹钟中断、定周期中断。闹钟中断与定周期中断共用一个中断信号。

33.3.1 RTC 闹钟中断

当 RTC_CR1.ALMIE=1 时，若当前日历时间与秒闹钟寄存器（RTC_ALMSEC）、分闹钟寄存器（RTC_ALMMIN）、时闹钟寄存器（RTC_ALMHOUR）、周闹钟寄存器（RTC_ALMWEEK）相等时，触发闹钟中断。

33.3.2 RTC 周期中断

控制寄存器 1（RTC_CR1）的 ALMIE=1 时，选择的周期发生后，触发定周期唤醒中断，由于闹钟和定周期共用中断，通过标志寄存器位来区分。

33.4 RTC 寄存器描述

基地址 0x40002800

寄存器	偏移地址	描述
RTC_CR0	0x000	控制寄存器0
RTC_CR1	0x004	控制寄存器1
RTC_SEC	0x008	秒计数寄存器
RTC_MIN	0x00C	分计数寄存器
RTC_HOUR	0x010	时计数寄存器
RTC_WEEK	0x014	周计数寄存器
RTC_DAY	0x018	日计数寄存器
RTC_MON	0x01C	月计数寄存器
RTC_YEAR	0x020	年计数寄存器
RTC_ALMMIN	0x024	分闹钟寄存器
RTC_ALMHOUR	0x028	时闹钟寄存器
RTC_ALM WEEK	0x02C	周闹钟寄存器
RTC_COMPEN	0x030	时钟误差补偿寄存器
RTC_ALMSEC	0x034	秒闹钟寄存器

表 33-2 RTC 寄存器列表

33.4.1 控制寄存器 0 (RTC_CR0)

*只有上电对该寄存器复位有效

地址偏移量: 0x000

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
RSV		PRDSEL		PRDX											
		R/W		R/W											
7		6		5		4		3		2		1		0	
START		HZ1SEL		HZ1OE		RSV		AMPM		PRDS					
R/W		R/W		R/W				R/W		R/W					

位	符号	功能描述
31:15	Reserved	-
14	PRDSEL	0: 使用 PRDS 所设定的周期中断时间间隔 1: 使用 PRDX 所设定的周期中断时间间隔
13:8	PRDX	设置产生周期中断的时间间隔, 可设定的范围为0.5秒到32秒, 步进为0.5秒。 000000: 0.5秒 000001: 1秒 111110: 31.5秒 111111: 32秒
7	START	0: 停止 RTC 计数器 1: 使能 RTC 计数器
6	HZ1SEL	0: 普通精度 1Hz 输出 1: 高精度 1Hz 输出
5	HZ1OE	0: 禁止 1Hz 输出 1: 使能 1Hz 输出
4	Reserved	-
3	AMPM	0: 12小时制 1: 24小时制
2:0	PRDS	设置产生中断的时间间隔: 000: 不产生周期中断 001: 0.5秒 010: 1秒

		011: 1分钟 100: 1小时 101: 1天 11x: 1月 注意: 如需要在 START=1 时写入更改周期中断的时间间隔操作步骤如下: step1, 在 NVIC 中关闭 RTC 中断; step2, 更改周期中断的时间间隔; step3, 清除 RTC 中断标志; step4, 使能 RTC 中断。

33.4.2 控制寄存器 1 (RTC_CR1)

*只有上电对该寄存器复位有效

地址偏移量: 0x004

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
Reserved										CKSEL					
										R/W					
7		6		5		4		3		2		1		0	
ALMEN		ALMIE		RSV		ALMF		PRDF		RSV		WAITF		WAIT	
R/W		R/W				R/W		R/W				R/W		R/W	

位	符号	功能描述
31:11	Reserved	-
10:8	CKSEL	RTC 时钟选择 00x: XTL 32.768k 01x: RCL 32k 100: XTH/128 (晶振为 4M 时选择此项) 101: XTH/256 (晶振为 8M 时选择此项) 110: XTH/512 (晶振为 16M 时选择此项) 111: XTH/1024 (晶振为 32M 时选择此项)
7	ALMEN	0: 禁止闹钟 1: 使能闹钟 注意: 在 START=1 日历计数过程中并且 ALMIE=1 中断许可的情况下使能ALMEN 时, 为防止误动作请将系统中断关闭。使能后请将 ALMF 标志位清除。

6	ALMIE	0: 禁止闹钟中断 1: 使能闹钟中断
5	Reserved	–
4	ALMF	0: 未发生闹钟中断 1: 已发生闹钟中断 注意: 该位仅在 ALMEN=1 时有效。闹钟匹配时, 32.7689kHz 一个时钟后置1。写0时清除标志, 写1无效。
3	PRDF	0: 未发生周期中断 1: 已发生周期中断 注意: 发生周期中断后, 该位置1。写0时清除该标志, 写1无效。
2	Reserved	–
1	WAITF	0: 非写入/读出状态 1: 写入/读出状态 注意: WAIT 位设定是否有效标志。在写入/读出前请确认该位是否为“1”。计数过程中, 在 WAIT 位清”0“后等待写入完成后该位才清”0“。
0	WAIT	0: 正常计数模式 1: 写入/读出模式 注意: 在写入/读出时请将该位置“1”, 由于计数器在连续计数, 请在1秒的时间内完成写入/读出操作并将该位清“0”。

33.4.3秒计数寄存器 (RTC_SEC)

地址偏移量: 0x008

复位值: 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SECH		SECL			
										R/W		R/W			

位	符号	功能描述
31:7	Reserved	–
6:4	SECH	秒计数十位值
3:0	SECL	秒计数个位值

表示 0-59 秒, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

33.4.4 分计数寄存器（RTC_MIN）

地址偏移量：0x00C

复位值：不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										MINH		MINL			
										R/W		R/W			

位	符号	功能描述
31:7	Reserved	–
6:4	MINH	分计数十位值
3:0	MINL	分计数个位值

表示 0-59 分，采用十进制计数。请写入十进制 0-59 的 BCD 码，写入错误值时，写入值将被忽略。

33.4.5 时计数寄存器（RTC_HOUR）

地址偏移量：0x010

复位值：不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										HOURH		HOURL			
										R/W		R/W			

位	符号	功能描述
31:6	Reserved	–
5:4	HOURH	时计数十位值
3:0	HOURL	时计数个位值

24 小时制时，表示 0-23 小时。12 小时制时，b5=0 表示 AM，则 01:12 表示上午；b5=1 表示 PM，则 21:32 表示下午。

请根据控制为 AMPM 的值，设定正确十进制的 0:23 或者 01:12,21:32 的 BCD 码。

写入超出范围的值将被忽略。

具体时间表示参考下表：

24小时制	AMPM=1	12小时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
00时	00H	AM 12时	12H
01时	01H	AM 01时	01H
02时	02H	AM 02时	02H
03时	03H	AM 03时	03H
04时	04H	AM 04时	04H
05时	05H	AM 05时	05H
06时	06H	AM 06时	06H
07时	07H	AM 07时	07H
08时	08H	AM 08时	08H
09时	09H	AM 09时	09H
10时	10H	AM 10时	10H
11时	11H	AM 11时	11H
12时	12H	PM 12时	32H
13时	13H	PM 01时	21H
14时	14H	PM 02时	22H
15时	15H	PM 03时	23H
16时	16H	PM 04时	24H
17时	17H	PM 05时	25H
18时	18H	PM 06时	26H
19时	19H	PM 07时	27H
20时	20H	PM 08时	28H
21时	21H	PM 09时	29H
22时	22H	PM 10时	30H
23时	23H	PM 11时	31H

33.4.6 日计数寄存器（RTC_DAY）

地址偏移量：0x018

复位值：不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										DAYH		DAYL			
										R/W		R/W			

位	符号	功能描述
31:6	Reserved	–
5:4	DAYH	日计数十位值
3:0	DAYL	日计数个位值

十进制表示 1:31 日，自动计算闰年和月份。具体表示如下：

月份	日计数表示
2月（普通年）	01:28
2月（闰年）	01:29
4、6、9、11月	01:30
1、3、5、7、8、10、12月	01:31

33.4.7 周计数寄存器（RTC_WEEK）

地址偏移量：0x014

复位值：不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													WEEK		
													R/W		

位	符号	功能描述
31:3	Reserved	–
2:0	WEEK	周计数值

十进制 0:6 表示周日:周六。请写入正确的十进制 0:6 的 BCD 码，写入其他值，将

被忽略。周计数值对应关系如下：

周	周计数表示
周日	00H
周一	01H
周二	02H
周三	03H
周四	04H
周五	05H
周六	06H

33.4.8 月计数寄存器（RTC_MON）

地址偏移量：0x01C

复位值：不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											MON				
											R/W				

位	符号	功能描述
31:5	Reserved	-
4:0	MON	月计数值

十进制 1:12 表示 1:12 月。请写入正确的十进制 1:12 的 BCD 码，写入其他值，将被忽略。

33.4.9 年计数寄存器 (RTC_YEAR)

地址偏移量: 0x020

复位值: 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								YEARH				YEARL			
								R/W				R/W			

位	符号	功能描述
31:8	Reserved	–
7:4	YEARH	年计数十位值
3:0	YEARL	年个位计数值

十进制 0:99 表示 0:99 年。根据月进位计数。自动计算闰年如: 00、04、08、...、92、96 等。请写入正确的十进制年计数值, 写入错误值将被忽略。

33.4.10 秒闹钟寄存器 (RTC_ALMSEC)

地址偏移量: 0x034

复位值: 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ALMSECH				ALMSECL			
								R/W				R/W			

位	符号	功能描述
31:7	Reserved	–
6:4	ALMSECH	秒闹钟匹配值十位
3:0	ALMSECL	秒闹钟匹配值个位

请设定十进制 0:59 的 BCD 码。写入其他值, 不会发生闹钟匹配。

33.4.11 分闹钟寄存器 (RTC_ALMMIN)

地址偏移量: 0x024

复位值: 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ALMMINH		ALMMINL				
									R/W		R/W				

位	符号	功能描述
31:7	Reserved	–
6:4	ALMMINH	分闹钟匹配值十位
3:0	ALMMINL	分闹钟匹配值个位

请设定十进制 0:59 的 BCD 码。写入其他值, 不会发生闹钟匹配。

33.4.12 时闹钟寄存器 (RTC_ALMHOUR)

地址偏移量: 0x028

复位值: 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										ALMHOURH		ALMHOURL			
										R/W		R/W			

位	符号	功能描述
31:6	Reserved	–
5:4	ALMHOURH	时闹钟十位匹配值
3:0	ALMHOURL	时闹钟个位匹配值

请根据时制设定正确的闹钟匹配值, 否则不会发生时闹钟匹配。

33.4.13 周闹钟寄存器 (RTC_ALMWEED)

地址偏移量: 0x02C

复位值: 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										ALM WEEK					
										R/W					

位	符号	功能描述
31:7	Reserved	—
6:0	ALM WEEK	周闹钟匹配值。 b0:b6 分别对应周日:周六, 对应为置“1”时, 代表每周该日闹钟有效。如, b0=1, b5=1 代表周日和周五闹钟设定有效。

请根据时制设定正确的闹钟匹配值, 否则不会发生时闹钟匹配。

33.4.14 时钟误差补偿寄存器 (RTC_COMPEN)

地址偏移量: 0x030

*只有上电对该寄存器复位有效, 复位值: 0x00000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	Reserved							CR							
R/W								R/W							

位	符号	功能描述																																																	
31:16	Reserved	-																																																	
15	EN	补偿使能 0: 禁止时钟误差补偿 1: 使能时钟误差补偿																																																	
14:9	Reserved	-																																																	
8:0	CR	补偿值 通过补偿值设定, 可针对每秒进行 $\pm 0.96\text{ppm}$ 的精度补偿。补偿值为9位带小数点的2的补码, 后5位为小数部分。可补偿范围 -274.6ppm:212.6ppm。最小微分误差 $\pm 0.48\text{ppm}$。最小分辨率 0.96ppm。具体补偿精度请参考下表: <table><tr><th colspan="10">补偿值设定</th><th rowspan="5">补偿数</th></tr><tr><th>EN</th><th colspan="9">CR[8:0]</th></tr><tr><td rowspan="3">1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr></table>	补偿值设定										补偿数	EN	CR[8:0]									1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	:	:	:	:	:	:	:	:	:
补偿值设定										补偿数																																									
EN	CR[8:0]																																																		
1	1	0	0	0	0	0	0	0	0																																										
	1	0	0	0	0	0	0	0	1																																										
	:	:	:	:	:	:	:	:	:																																										

			0	0	0	0	1	1	1	1	1	-0.95ppm	
			0	0	0	1	0	0	0	0	0	0ppm	
			:	:	:	:	:	:	:	:	:	:	
			0	1	1	1	1	1	1	1	0	+211.7ppm	
			0	1	1	1	1	1	1	1	1	+212.6ppm	
		0	X	X	X	X	X	X	X	X	X	无补偿	

补偿原理说明与计算：

由于计数器采用 32.768kHz 的时钟计数，如果需要对每秒精度进行补偿时，只能按照 32.768kHz 的整数周期补偿，则每秒补偿的最小单位为 $(1/32768) \times 106 = 30.5\text{ppm}$ ，无法满足高精度的要求。

那么要在 32.768kHz 的计数时钟下实现精度较高的时钟补偿时，需要在算法上做调整，将最大补偿周期扩大 32 倍。则在只能补偿的最小单位为 30.5ppm 的情况下，平均每秒的补偿单位变为 $30.5\text{ppm}/32 = 0.96\text{ppm}$ 。满足了精度较高的时钟补偿要求。而且补偿发生在每 32 秒内比较均匀的范围。所以，该寄存器中引入了 5 位小数的设定。

设定值计算如下：

$$\text{CR}[8 : 0] = \left(\frac{\text{补偿目标值}[\text{ppm}] \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.00000 \text{ B}$$

如果补偿目标值为 +20.6ppm，计算相应的寄存器值如下：

$$\begin{aligned} \text{CR}[8:0] &= (20.3 \times 215/106)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= (0.6651904)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= 0000.10101\text{B} + 0001.00000\text{B} \\ &= 0001.10101\text{B} \end{aligned}$$

如果补偿目标值为-20.6ppm，计算相应的寄存器值如下：

$$\begin{aligned} \text{CR}[8:0] &= (-20.3 \times 215/106)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= (-0.6651904)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= 1111.01011\text{B} + 0001.00000\text{B} \\ &= 0000.01011\text{B} \end{aligned}$$

34. 脉冲计数器（PCNT）

34.1 脉冲计数器简介

脉冲计数器（PCNT）模块可以对输入的脉冲进行计数，支持三种脉冲模式：单通道脉冲、双通道正交脉冲、双通道非交脉冲。无需软件参与，可在低功耗模式下正确计数。

34.2 脉冲计数器主要特性

脉冲计数器支持以下特性：

- 支持重载功能的 16 bit 计数器
- 单通道脉冲计数
- 双通道非交脉冲计数
- 双通道正交脉冲计数，不失码
- 加/减计数溢出中断
- 脉冲超时中断
- 4 种解码错误中断，非交脉冲模式
- 1 种方向改变中断，正交脉冲模式
- 多级脉冲宽度滤波
- 输入脉冲极性可配置
- 支持低功耗模式计数
- 支持唤醒低功耗模式下 MCU
- 支持任意脉冲沿间距不小于 1 个计数时钟周期
- 支持 IO 输入或者 VC 输出作为脉冲输入信号
- 支持双路脉冲信号整形之后通过 IO 观测输出 PCNT_S0FO/PCNT_S1FO

34.3 脉冲计数器功能描述

34.3.1 整体框图

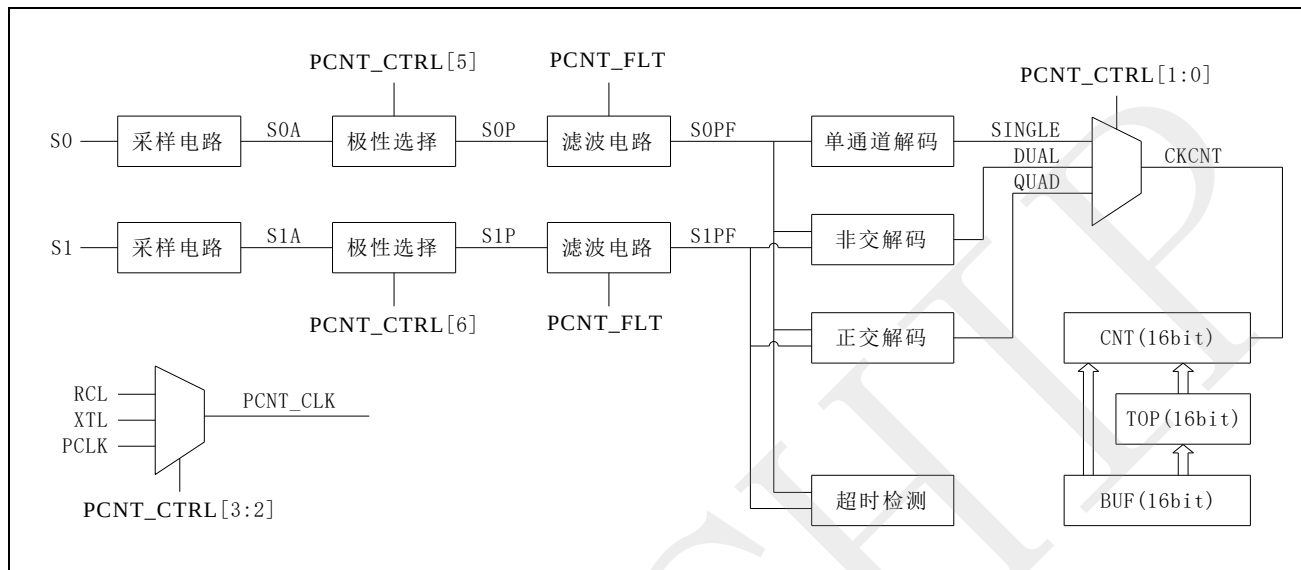


图 34-1 整体框图

34.3.2 信号说明

以下以输入信号 S0 为例说明信号在计数器模块中的处理流程。

- S0 信号经采样模块后输出 S0A 信号，采样时钟为 PCNT_CLK。
- S0A 信号经极性选择模块后输出 SOP 信号。
- SOP 信号经脉冲宽度滤波模块后输出 S0PF 信号，滤波模块的时钟为分频后的 PCNT_CLK。
- S0PF 信号经单通道解码模块、双通道非交解码模块、双通道正交解码模块解码后输出相应的 SIGNAL、DUAL、QUAD 信号。
- SIGNAL、DUAL、QUAD 经多路选择器后输出 CKCNT 信号以驱动 CNT 模块进行计数。

34.3.3 计数模式

本模块支持 3 种计数模式：单通道脉冲、双通道正交脉冲、双通道非交脉冲。

34.3.3.1 单通道脉冲计数模式（Single Mode）

当配置 PCNT_CTRL.Mode 为 0x00 或 0x01 时，脉冲计数器工作于单通道脉冲计数模式。该模式下解码模块只对 S0 脉冲信号进行计数。当 PCNT_CLK 时钟采样到 S0PF 信号的下降沿时，计数器按 PCNT_CTRL.DIR 的配置进行一次递增或递减操作。

计数器的计数范围为 0x00 至计数上阈值(PCNT_TOP)。

在加计数状态，当计数值等于 PCNT_TOP 且采样到 S0PF 信号的下降沿时，计数器发生上溢出，计数值回到 0 并置位 PCNT_IFR.OF，该中断标志需要使用软件进行清零。下图为 PCNT 加计数的时序图，图中 PCNT_TOP 的值为 99。

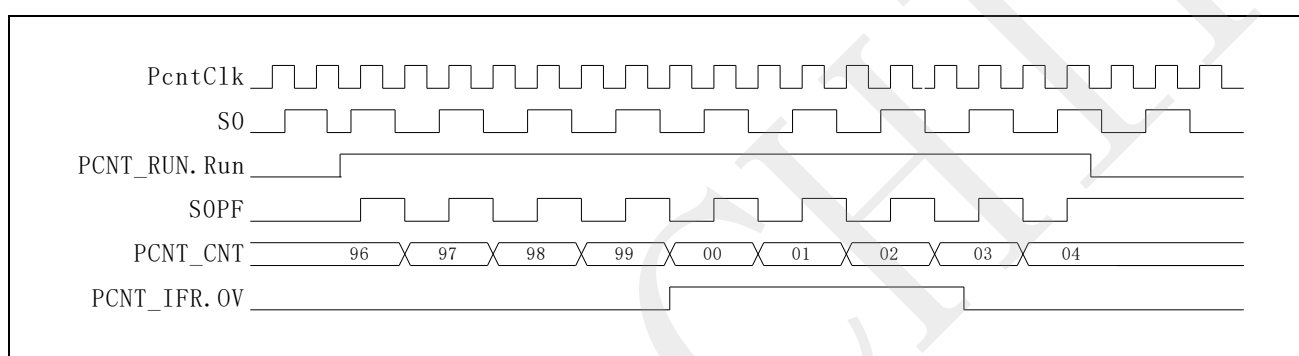


图 34-2 单通道脉冲计数模式加计数波形

在减计数状态，当计数值等于 0x00 且采样到 S0PF 信号的下降沿时，计数器发生下溢出，计数值回到 PCNT_TOP 并置位 PCNT_IFR.UF，该中断标志需要使用软件进行清零。下图为 PCNT 减计数的时序图，图中 PCNT_TOP 的值为 99。

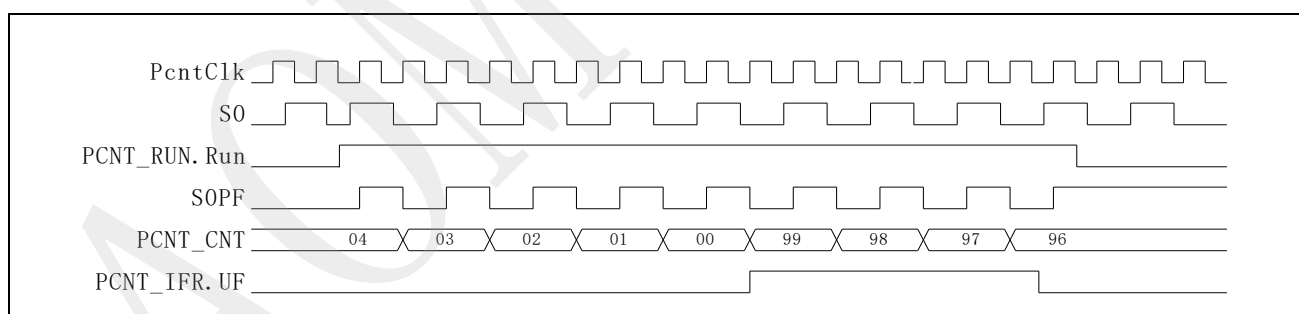


图 34-3 单通道脉冲计数模式减计数波形

34.3.3.2 双通道非交脉冲计数模式（Dual Mode）

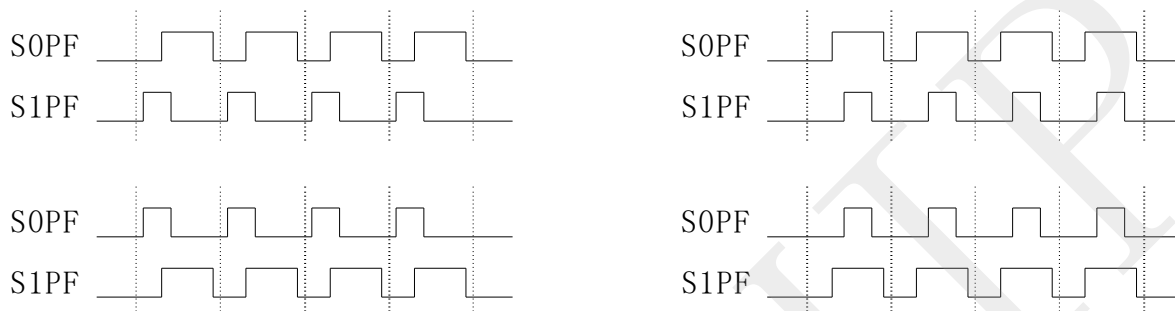
当配置 PCNT_CTRL.Mode 为 0x02 时，脉冲计数器工作于双通道非交脉冲计数模式。

该模式下解码模块对 S0 和 S1 脉冲信号进行解码计数。当 S0PF 和 S1PF 依次出现两个正脉冲，则计数器按 PCNT_CTRL.DIR 的配置进行一次递增或递减操作。

解码模块能正确计数的两种波形如下所示，S0PF 与 S1PF 没有同时为高的情况。



解码模块不能正确计数的几种波形如下所示，S0PF 与 S1PF 出现同时为高的情况。



计数器的计数范围为 0x00 至计数上阈值(PCNT_TOP)。

在加计数状态，当计数值等于 PCNT_TOP 且 S0PF、S1PF 依次出现两个正脉冲时，计数器发生上溢出，计数值回到 0 并置位 PCNT_IFR.OF，该中断标志需要使用软件进行清零。下图为 PCNT 加计数的时序图，图中 PCNT_TOP 的值为 99。

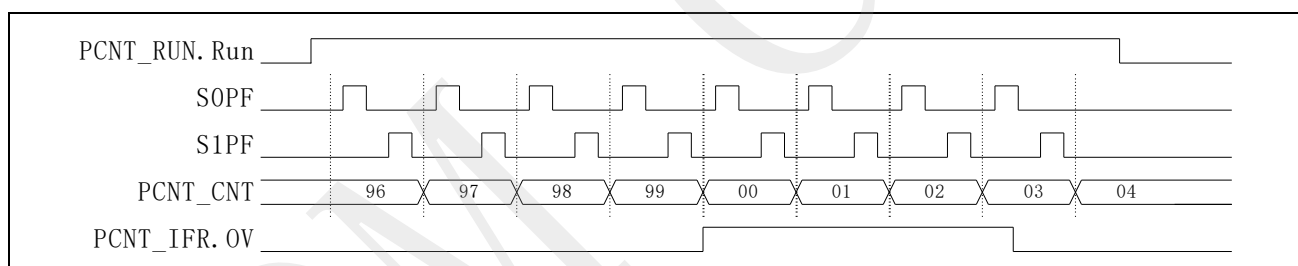


图 34-4 双通道非交脉冲计数模式加计数波形

在减计数状态，当计数值等于 0x00 且 S0PF、S1PF 依次出现两个正脉冲时，计数器发生下溢出，计数值回到 PCNT_TOP 并置位 PCNT_IFR.UF，该中断标志需要使用软件进行清零。下图为 PCNT 减计数的时序图，图中 PCNT_TOP 的值为 99。

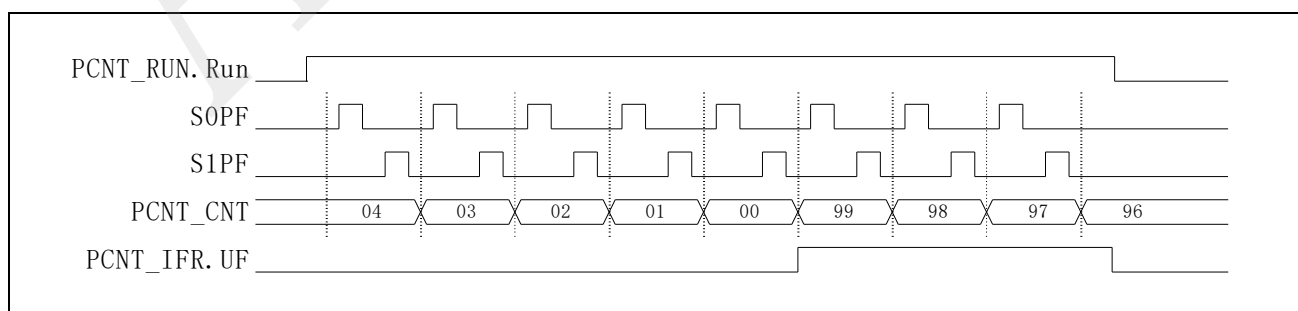


图 34-5 双通道非交脉冲计数模式减计数波形

34.3.3.3 双通道正交脉冲记数模式（Quad Mode）

当配置 PCNT_CTRL.Mode 为 0x03 时,脉冲计数器工作于双通道正交脉冲计数模式。

该模式下解码模块需要 S0 和 S1 两路脉冲信号进行解码计数。计数器根据 S0PF 和 S1PF 脉冲的相位关系自动判断计数的方向,物理设备任意正转、反转均能正确计数。

当 S0PF 和 S1PF 完成一个周期的变化时,计数器按计数方向进行一次递增或递减操作。

解码模块判断计数方向的波形如下所示:



计数器的计数范围为 0x00 至计数上阈值(PCNT_TOP)。

在加计数状态,当计数值等于 PCNT_TOP 且 S0PF 和 S1PF 完成一个周期的变化时,计数器发生上溢出,计数值回到 0 并置位 PCNT_IFR.OF,该中断标志需要使用软件进行清零。下图为 PCNT 加计数的时序图,图中 PCNT_TOP 的值为 99。

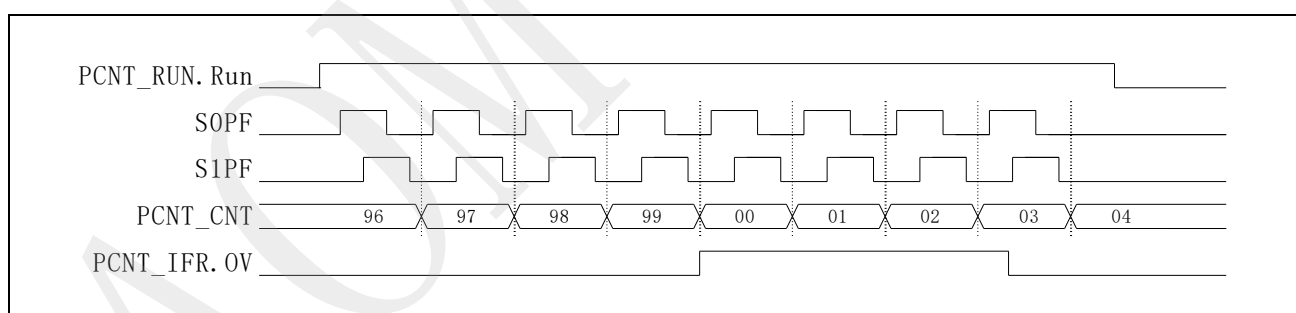


图 34-6 双通道正交脉冲计数模式加计数波形

在减计数状态,当计数值等于 0x00 且 S0PF 和 S1PF 完成一个周期的变化时,计数器发生下溢出,计数值回到 PCNT_TOP 并置位 PCNT_IFR.UF,该中断标志需要使用软件进行清零。下图为 PCNT 减计数的时序图,图中 PCNT_TOP 的值为 99。

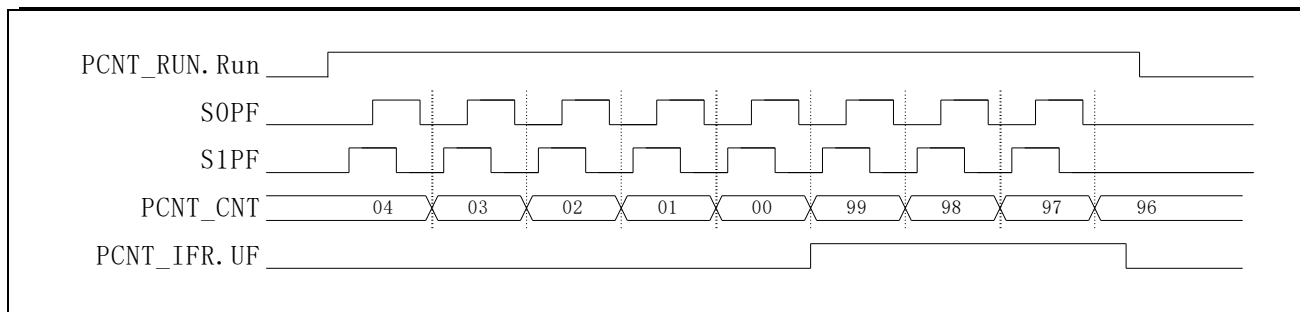


图 34-7 双通道正交脉冲计数模式减计数波形

34.3.4 脉冲宽度滤波

该脉冲宽度滤波器使用计数时钟分频作为滤波时钟，并以此计数从而实现脉冲去抖动目的。其中滤波时钟的分频系数从 1~8096 倍配置寄存器可选（FLT.ClkDiv），保持电平超过（FLT.DebTop）（1~7）个滤波时钟视为非抖动，可通过滤波器。

该滤波功能可由配置寄存器（FLT.EN）打开或关闭。

同时仅当启动 PCNT 之后（Run=1），滤波计数器才启动。

34.3.5 超时

打开超时功能（TOCR.EN）将使用滤波时钟对脉冲高电平时间计数，采样到低电平将使计数器回到 0，若一个高电平计数值达到（TOCR.TH）将产生中断标志位 IF.TO。

同时仅当启动 PCNT 之后（Run=1），超时计数器才启动。

34.3.6 脉冲输入选择

相较于 6615 系列在 IO_MUX 中选择 IO 的输入作为 PCNT 的脉冲输入，在 6615 系列中增加了 VC 的输出作为 PCNT 的脉冲输入，通过 IO_CTRL7 寄存器来进行选择，详见 IO_CTRL 模块的寄存器描述。

34.3.7 Debug 观测输出

为确认进入脉冲解码模块的信号符合预期，可以通过寄存器选择三组整形后的脉冲通过 IO 输出作为 PCNT_S0FO/PCNT_S1FO 观测信号，分别是：

- S0A/S1A
- S0P/S1P
- S0PF/S1PF

通过观测 PCNT_S0FO/PCNT_S1FO 输出脉冲信号，可以更直观的调整极性选择和脉冲宽度滤波器的配置，避免解码电路的输入信号不符合预期。

34.4 PCNT 寄存器描述

寄存器列表

基地址：0x40000C00

偏移量	寄存器名称	访问	同步/异步	寄存器描述
0x00	PCNT_RUN	RW	异步	PCNT 启动寄存器
0x04	PCNT_CTRL	RW	异步长信号	PCNT 控制寄存器
0x08	PCNT_FLT	RW	异步长信号	PCNT 滤波控制寄存器
0x0C	PCNT_TOCR	RW	异步长信号	PCNT 超时控制寄存器
0x10	PCNT_CMD	W01	异步	PCNT 命令寄存器
0x14	PCNT_SR1	RO	同步	PCNT 状态寄存器 1
0x18	PCNT_CNT	RO	同步	PCNT 计数寄存器
0x1C	PCNT_TOP	RO	同步	PCNT 计数溢出寄存器
0x20	PCNT_BUF	RW	异步	PCNT 计数溢出缓存寄存器
0x24	PCNT_IFR	RO	同步	PCNT 中断标识寄存器
0x28	PCNT_ICR	W00	异步	PCNT 中断清除寄存器
0x2C	PCNT_IEN	RW	异步长信号	PCNT 中断使能寄存器
0x30	PCNT_SR2	RO	异步	PCNT 状态寄存器 2
0x34	PCNT_DBG	RW	异步长信号	PCNT PCNT_S0F0/PCNT_S1F0 输出选择寄存器

表 34-1 寄存器列表

34.4.1 PCNT 启动寄存器(PCNT_RUN)

地址偏移量：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														Run	
														RW	

位	标记	功能描述
31:1	Reserved	保留
0	Run	PCNT启动/停止控制。 0: 停止 1: 启动

注意：

- 当前模块将 pclk 时钟域的配置寄存器视作长信号，未进行到 pcnt_clk 时钟域的同步，所以当 Run=1 启动 PCNT 后，配置寄存器**不允许**进行修改以避免发生未知错误。所有配置寄存器的修改务必在 PCNT 停止的状态（Run=0）下进行。
- 对 Run 位写入一个值后，需要经过一个同步的过程才可启动/停止 PCNT 计数，在同步未完成之前读取 Run 位将返回写入之前的 Run 状态值。所以如果要停止 PCNT 之后再启动 PCNT，需对 Run 位写入 0，查询到 Run 位=0 之后才可继续后续其他操作（包括对 Run 位写入 1 以重新启动 PCNT）。

34.4.2 PCNT 控制寄存器(PCNT_CTRL)

地址偏移量：0x04

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									S1P	S0P	DIR	ClkSel	Mode		
									RW	RW	RW	RW	RW		

位	标记	功能描述
31:7	Reserved	保留
6	S1P	S1通道输入极性选择 0: 不取反 1: 取反
5	S0P	S0 通道输入极性选择 0: 不取反 1: 取反
4	DIR	计数方向选择 0: 加计数 1: 减计数 注：仅对单通道模式和双通道非交模式有效。
3:2	ClkSel	PCNT_CLK计数时钟选择 0: PCLK 1: PCLK 2: XTL 3: RCL
1:0	Mode	脉冲计数模式选择 0: 单通道脉冲计数模式 1: 单通道脉冲计数模式 2: 双通道非交脉冲计数模式 3: 双通道正交脉冲计数模式

34.4.3 PCNT 滤波控制寄存器(PCNT_FLT)

地址偏移量：0x08

复位值：0x00002000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DebTop			ClkDiv												
RW			RW												

位	标记	功能描述
31:17	Reserved	保留
16	EN	脉冲宽度滤波器使能控制。 0: 不使能 1: 使能
15:13	DebTop	计数器阈值 使用滤波时钟对输入脉冲连续采样，当连续电平采样计数值达到阈值则视为正常脉冲，可通过滤波器。 0: 为非法值
12:0	ClkDiv	滤波时钟分频系数 系数 = ClkDiv + 1

34.4.4 PCNT 超时控制寄存器(PCNT_TOCR)

地址偏移量: 0x0C

复位值: 0x00000FFF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TH											
RW				RW											

位	标记	功能描述
31:17	Reserved	保留
16	EN	超时功能使能控制。 0: 不使能 1: 使能
15:12	Reserved	保留
11:0	TH	超时阈值。 当脉冲高电平被滤波时钟连续采样计数达到阈值, 产生超时中断标识, 计数器重新开始计数。

34.4.5 PCNT 命令寄存器(PCNT_CMD)

地址偏移量: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													B2C	B2T	T2C
													WO1	WO1	WO1

位	标记	功能描述
31:3	Reserved	保留

2	B2C	写1，立即将BUF中的值同步到CNT。 同步过程中PCNT_SR2的相应位为1，此时不应应对BUF进行写操作，不应应对CNT进行读操作。
1	B2T	写1，立即将BUF中的值同步到TOP。 同步过程中PCNT_SR2的相应位为1，此时不应应对BUF进行写操作，不应应对TOP进行读操作。
0	T2C	写1，立即将TOP中的值同步到CNT。 同步过程中PCNT_SR2的相应位为1，此时不应应对B2T写1，不应应对CNT进行读操作。

34.4.6 PCNT 状态寄存器 1(PCNT_SR1)

地址偏移量：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														DIR	
														RO	

位	标记	功能描述
31:1	Reserved	保留
0	DIR	双通道正交脉冲计数方向指示 0：加计数 1：减计数

34.4.7 PCNT 计数寄存器(PCNT_CNT)

地址偏移量: 0x18

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RO															

位	标记	功能描述
31:16	Reserved	保留
15:0	CNT	计数器计数值。

34.4.8 PCNT 计数溢出寄存器(PCNT_TOP)

地址偏移量: 0x1C

复位值: 0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP															
RO															

位	标记	功能描述
31:16	Reserved	保留
15:0	TOP	计数器上溢出阈值。 计数器下溢出重载值。

34.4.9 PCNT 计数溢出缓存寄存器(PCNT_BUF)

地址偏移量: 0x20

复位值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUF															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	BUF	CNT寄存器、TOP寄存器的缓存。

34.4.10 PCNT 中断标识寄存器(PCNT_IFR)

地址偏移量：0x24

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								S1E	S0E	BB	FE	DIR	TO	OV	UF
								RO	RO	RO	RO	RO	RO	RO	RO

位	标记	功能描述
31:8	Reserved	保留
7	S1E	脉冲解码错误中断标识。 在一个非交编码完整采样周期内，S1通道未发生变化。 仅在双通道非交编码模式有效。
6	S0E	脉冲解码错误中断标识。 在一个非交编码完整采样周期内，S0通道未发生变化。 仅在双通道非交编码模式有效。
5	BB	脉冲解码错误中断标识。 双通道脉冲分别背靠背采样为高，中间并无低状态。 仅在双通道非交编码模式有效。
4	FE	脉冲解码错误中断标识。 在一个非交编码完整采样周期内，采样到的不是一个正确的非交编码帧。 仅在双通道非交编码模式有效。

3	DIR	正交脉冲方向改变中断标识。 正交脉冲被解码器判定为计数方向发生改变。 仅在双通道正交脉冲有效。
2	TO	超时中断标识。 在3种模式均有效。
1	OV	上溢出中断标识。 在3种模式均有效。
0	UF	下溢出中断标识。 在3种模式均有效。

34.4.11 PCNT 中断清除寄存器(PCNT_ICR)

地址偏移量: 0x28

复位值: 0x000000FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								S1E	S0E	BB	FE	DIR	TO	OV	UF
								WO0	WO0	WO0	WO0	WO0	WO0	WO0	WO0

对每个 bit 写 0 可对 34.4.10 中 PCNT_IFR 各对应中断标识清除。

每 bit 详细解释参见 34.4.10。

34.4.12 PCNT 中断使能寄存器(PCNT_IEN)

地址偏移量: 0x2C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								S1E	S0E	BB	FE	DIR	TO	OV	UF
								RW	RW	RW	RW	RW	RW	RW	RW

对每个 bit 对应 34.4.10 中 PCNT_IFR 各对应中断的输出使能位。

1: 使能。 0: 不使能。

每 bit 详细解释参见 34.4.10。

34.4.13 PCNT 同步状态寄存器(PCNT_SR2)

地址偏移量: 0x30

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													B2C	B2T	T2C
													RO	RO	RO

位	寄存器名	描述
31:3	Reserved	保留
2	B2C	将BUF中数值同步到CNT时的同步状态位。 1: 正在进行同步, 不可读取CNT寄存器 0: 同步已完成
1	B2T	将BUF中数值同步到TOP时的同步状态位。 1: 正在进行同步, 不可读取TOP寄存器 0: 同步已完成
0	T2C	将TOP中数值同步到CTN时的同步状态位。 1: 正在进行同步, 不可读取CNT寄存器 0: 同步已完成

34.4.14 PCNT debug 观测输出选择寄存器(PCNT_DBG)

地址偏移量: 0x34

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														DBG	
														RW	

Bits	寄存器名	描述
31:2	Reserved	保留

1:0	DBG	选择脉冲PCNT_S0F0/PCNT_S1F0观测输出的来源。 0: 固定为0 1: 脉冲同步之后的输出 (S0A/S1A) 2: 脉冲极性选择之后的输出 (S0P/S1P) 3: 脉冲滤波之后的输出 (S0PF/S1PF)

35. 时钟校准定时器 (CRS)

时钟校准定时器可以调整校准 RCH/RCL 时钟频率，不使用时钟校准功能时，也可以作为一个普通低功耗定时器来使用。

35.1 特性

- 支持手动 TRIM
- 自动选择 RCH, RCL 逼近式与 2 分法自动 TRIM
- 自动校准初始步进可设置
- 支持频率跟踪校准
- 可以作为低功耗定时器

35.2 时钟校准功能描述

时钟校准模块可以自动校准 RC 振荡时钟频率，也可以手动校准 RC 振荡时钟频率。对于高精度的应用环境，还可以实时跟踪时钟频率进行校准。防止环境变化造成振荡频率漂移。

35.2.1 系统框图

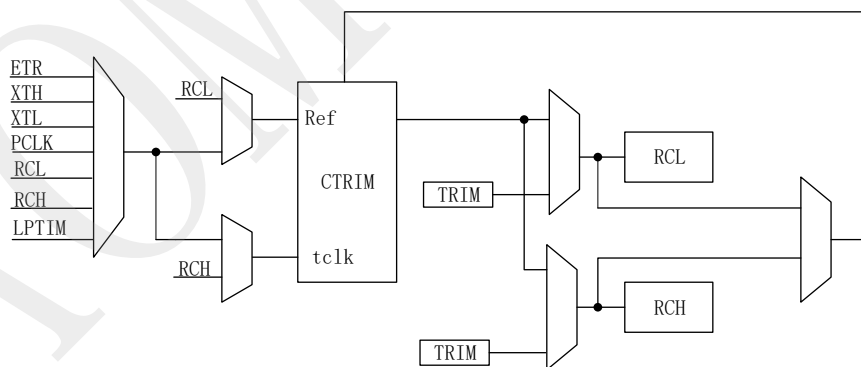


图 35-1 CRS 模块系统框图

35.2.2 CTRIM 模块框图

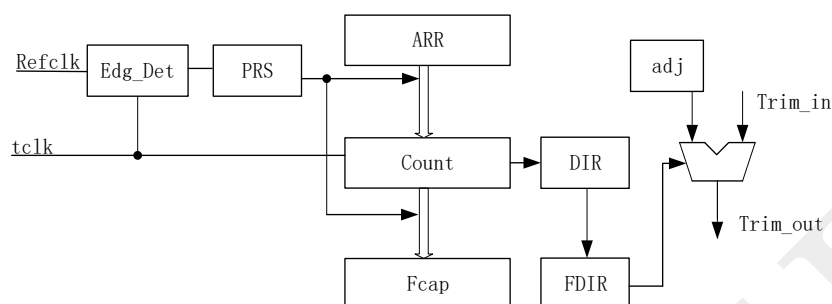


图 35-2 CTRIM 模块框图

基准参考时钟使用计数时钟同步后可以设置分频选择(计数时钟频率高于参考时钟频率的 2 倍以上), 选择范围为 $1 \sim 2^{15}$ 。分频后的信号为参考源信号。根据参考时钟频率选择合适的预分频生成参考源信号。

参考时钟分频后的周期内, 计数器从 ARR 向下计数, 计数到零后更改计数方向向上计数。

时钟自动校准有两种校准方式。

- 1) 参考时钟为精准时钟, 校准计数时钟。
- 2) 计数时钟为精准时钟, 校准参考时钟。

精准时钟可以使用内部的晶体振荡, 也可以使用外部输入的精准时钟源。

35.2.3 频率误差计数器

频率误差计数器是一个 16 位递减/递增计数器, 会在参考时钟周期结束时重载 ARR 值。它开始递减计数, 直至达到零值, 此时会生成 UDF (预期下溢出) 事件。随后它将递增计数到 ARR 限值, 这种情况下最终将停止计数 (如果未接收到参考源 SYNC 信号) 并生成 MISS 事件。

当检测到参考源 SYNC 事件时, 频率误差计数器的实际值及其方向存储在寄存器 FCAP 和 ISR. FDIR (频率误差方向) 里。

当校准时钟为计数时钟时 (校准 RCH 时, 时钟选择 CFGR.CLKSEL 选择一个精准时钟作为参考时钟), 当在递减计数阶段 (达到零值之前) 期间检测到参考源 SYNC 事

件时，意味着实际频率小于目标频率；当在递增计数期间检测到参考源 SYNC 事件时，意味着实际频率大于目标频率。

当校准时钟为参考时钟时（校准 RCL 时，时钟选择 CFGR.CLKSEL 选择一个精准时钟作为计数时钟），当在递减计数阶段（达到零值之前）期间检测到参考源 SYNC 事件时，意味着实际频率大于目标频率；当在递增计数期间检测到参考源 SYNC 事件时，意味着实际频率小于目标频率。

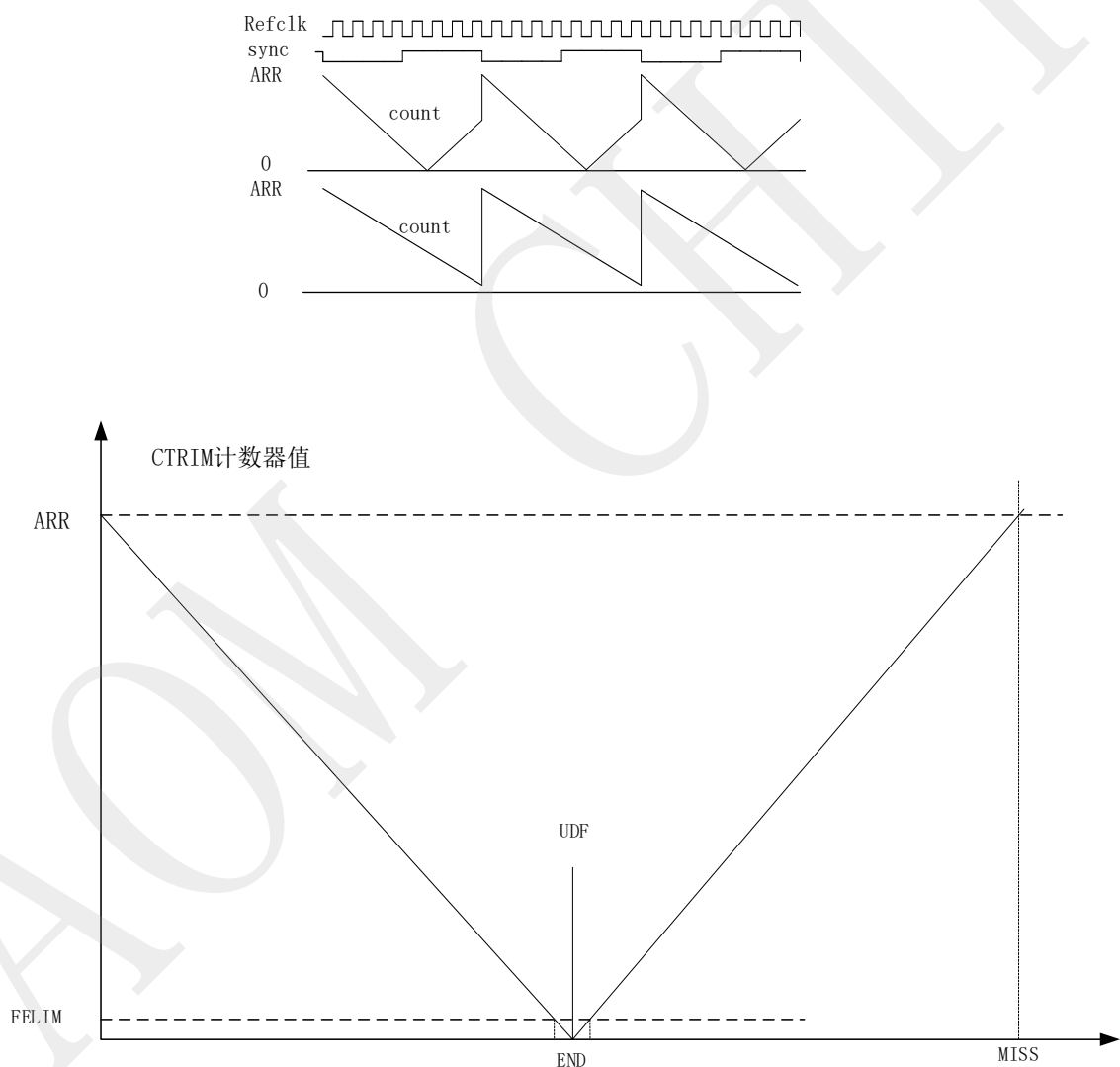


图 35-3 CTRIM 计数器

35.2.4 频率误差校准

- 当频率误差低于容差限值时，意味着 TRIM 字段中的实际微调值是最优值，因此无需任何微调操作。
 - 指示 END 状态，软件清除 CEN 寄存器，TRIM 停止
- 当频率误差与目标频率差值小于 2 倍时，TRIM 值初始步进使用 2 分法继续调整
 - 无状态指示
- 当计数器计数到 0 后再次向上计数到 ARR 时没有检测到参考时钟结束信号。
 - 指示 MISS 状态

35.2.5 CTRIM 初始化和配置

ARR 值

ARR 值应根据计数频率与分频后参考同步源频率之比进行选择。以在零值时达到预期同步。具体公式如下：

$$ARR = (f_{TARGET} / f_{SYNC}) - 1$$

如果计数时钟频率为 4MHz，参考时钟同步 sync 时钟频率为 1kHz，则 ARR 的值应该设置为 3999；

FLIM 值

FLIM 值的选择与振荡器的特性及其典型微调步长紧密相关。最优值对应于微调步长的一半。以 RCH 振荡器为计数时钟。可使用以下公式：

$$FLIM > (f_{TARGET} / f_{SYNC}) * STEP[\%] / 100\% / 2 \text{ (STEP 振荡 TRIM 值改变的精度)}$$

结果应始终舍入为最接近的整数值以获得最佳微调响应。如果应用中不需要频繁的微调操作，可通过稍微增大 FLIM 值来增加微调滞后。

校准后的目标频率偏差为 $(2 * FLIM) / ARR * 100\%$

35.2.6 自动校准说明

选择一个精准时钟作为参考时钟或者计数时钟，对于参考时钟选择合适的分频比，让计数时钟在参考时钟参数的同步 sync 信号中计数达到一定值。建议计数值不小于 200。根据计数时钟为待校准时钟还是参考时钟为计数时钟设置合适的模式。根据当前振荡

频率与目标频率的差选择合适的初始步进 step，能加速校准过程。

35.2.7 手动校准说明

选择一个精准时钟作为参考时钟或者计数时钟，对于参考时钟选择合适的分频比，让计数时钟在参考时钟参数的同步 sync 信号中计数达到一定值。建议计数值不小于 200。在单次处理完成子程序中根据 FDIR 与 FCAP 的值修改 TRIM 值进行振荡的调整。直到到达目标频率。

35.2.8 频率跟踪与校准

自动频率校准时，设置 CR.OST=1，校准频率达到目标频率或者 step 值变为 1 时，设置 END 标志位为 1，并结束自动校准校准。检测 OKF 标志看是否在允许频率范围内。CR.OST=0 时，校准频率达到目标频率时，设置 END 标志位为 1，不停止自动校准，继续进行校准不断的产生 END 中断标志。

35.3 振荡校准设置流程

35.3.1 RCH 时钟自动校准

- 1、设置 CR.MD=0,选择校准 RCH 时钟
- 2、设置 CFGR.clkSel=0/2 选择 外部输入或者低速晶体作为时钟输入(输入时钟频率低于 RCH 时钟频率的 1/2)。
- 3、设置 ARR=3905,FLIM=10
- 4、设置参考时钟预分频（选择 32768 时钟时，CR.PRS=5;32 分频）1024Hz 的同步 sync 信号
- 5、设置 CFGR.STEP=4 选择自动校准步进为 0x10
- 6、设置 CR.ATEN=1 使能自动校准
- 7、设置 CR.CEN 使能时钟校准模块
- 8、等待 ISR.END 标志等待自动校准结束，读取 TVAL 即是校准后的校准值。
- 9、关闭时钟校准，将 TVAL 的值写入 RCH 频率校准寄存器

35.3.2 RCL 时钟自动校准

计数时钟为 8M 时钟，校准 RCL 到 32K

- 1、设置 CFGR.ClkSel=1 或 2 选择外部时钟或者高速晶体 XTH 作为计数时钟
- 2、设置 CR.MD=1,选择校准 RCL 时钟(输入时钟频率高于 RCL 时钟频率的 2 倍)。
- 3、设置 ARR=8000,FLIM=20
- 4、设置参考时钟预分频 (CR.PRS=5;32 分频) 1ms 的同步 sync 信号 (RCL 校准后频率计算 1ms 同步信号)
- 5、设置 CFGR.STEP=4 选择自动校准步进为 0x10
- 6、设置 CR.ATEN=1 使能自动校准
- 7、设置 CR.CEN 使能时钟校准模块
- 8、等待 ISR.END 标志等待自动校准结束，读取 TVAL 即是校准后的 RCL 校准值。
- 9、关闭时钟校准，将 TVAL 的值写入 RCL 频率校准寄存器

校准高速时钟 RCH 时钟可以选择端口输入，或者内部其他精准时钟源
校准低速时钟 RCL 时，，时钟选择为精准的内部时钟或外部输入精准时钟。

35.3.3 手动校准过程

- 1、设置 CR.md 和 CFGR.ClkSel 选择参考时钟和计数时钟
- 2、设置 CR.PSR 设置合理的预分频
- 3、设置 ARR 与 FLIM 到合适的值
- 4、使能 PSF 中断
- 5、在中断服务子程序中清除 PSF 中断标志，判断 FCAP 值是否达到目标频率，达到结束进入 7，不到进入 6
- 6、中断服务子程序中根据 ISR.DIR 标志调整 TRIM 值
- 7、清除使能，写 CR.CEN 为 0。

35.4 定时器功能说明

35.4.1 框图

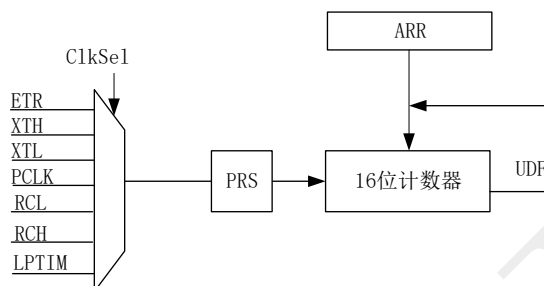


图 35-4 定时器模块框图

定时器特性

- 16 位下计数自动重载低功耗定时器
- 多种时钟源可配
- 1/2/4/8/16/32/64/128~32768 预分频可配置
- 定时器唤醒功能
- 支持溢出中断

35.4.2 定时器时钟选择

定时器可以选择 RCH, RCL, XTH, XTL, PCLK 等内部高速时钟, 也可以选择外部端口输入。

35.4.3 定时器

定时器计数值从 ARR 减计数, 当计数到 0 后产生下溢出, 如果中断使能会产生下溢出中断, 计数周期为 $ARR+1$ 。

35.5 CTRIM 寄存器

基地址 0x40006000

寄存器	偏移地址	描述
CTRIM_ARR	0x000	重装值
CTRIM_CNT	0x004	计数器值
CTRIM_CFGR	0x00C	配置寄存器
CTRIM_CR	0x010	控制寄存器
CTRIM_IER	0x014	中断使能控制寄存器
CTRIM_ISR	0x018	中断和状态寄存器
CTRIM_ICR	0x01C	中断标志清除寄存器
CTRIM_FCAP	0x020	TRIM周期结束时计数器的值
CTRIM_TVAL	0x024	TRIM值寄存器，TRIM过程软件读需要读两次相同有效
CTRIM_FLIM	0x028	误差允许值

表 35-1 CTRIM 寄存器

35.5.1 CTRIM 配置寄存器（CTRIM_ARR）

偏移地址：0x000

复位值：0x0000FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	ARR	计数器重载值

35.5.2 CTRIM 配置寄存器 (CTRIM_CNT)

偏移地址: 0x004

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
RO															

位	标记	功能描述
31:16	Reserved	保留
15:0	CNT	计数器计数值

35.5.3 CTRIM 配置寄存器 (CTRIM_CFGR)

偏移地址: 0x00C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV					clkssel		RSV	REFFLT			RSV	tstep			
					RW			RW				RW			

位	标记	功能描述
31:11	RSV	保留
10:8	clkssel	000: ETR 001: XTH 010: XTL 011: PCLK 100: RCL 101: RCH 110: LPTIMO TOG
7	RSV	保留

6:4	ETRFLT	外部输入滤波，滤波时钟为PCLK
3	RSV	保留
2:0	tstep	自动TRIM步进初值设置 000: 0x10 001: 0x02 010: 0x04 011: 0x08 100: 0x10 101: 0x20 110: 0x40 111: 0x80

注：这个寄存器只有在 CEN 为 0 时才可以更改。

35.5.4 CTRIM 控制寄存器 (CTRIM_CR)

偏移地址：0x010

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							OST	PRS			ATEN	MD		CEN	
							RW	RW			RW	RW		RW	

位	标记	功能描述
31:8	Reserved	保留
8	OST	0 实时校准模式, 自动TRIM不会自动结束 1 单次校准模式, 校准频率达到目标频率或者step从其他值变为1时, 设置END标志位为1, 并结束自动校准校准。
7:4	PRS	参考时钟预分频 TRIM同步信号分频器 / 定时器时钟预分频

		0000:不分频（校准时不支持） 0001: 2分频 0010: 4分频 0011: 8分频 0100:16分频 0101: 32分频 0110: 64分频 0111:128分频	1000:256分频 1001: 512分频 1010: 1024分频 1011: 2048分频 1100:4096分频 1101: 8192分频 1110: 16384分频 1111:32768 分频
3	ATEN	自动微调使能 1: 使能; 0: 禁止	
2:1	MD	功能选择 00: 校准RCH, CFGR.CLKSEL需要选择低速精准时钟自动作为参数时钟; 01: 校准RCL, CFGR.CLKSEL需要选择高速精准时钟自动作为计数时钟; 1x: 定时器功能	
0	CEN	频率误差计数器/定时器使能	

35.5.5 CTRIM 控制寄存器 (CTRIM_IER)

偏移地址: 0x014

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										OKF	OVF	MISS	PSF	END	UDF
										RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:6	Reserved	保留
5	OK_EN	自动Trim结束并且频率在FLIM范围内
4	OV_EN	TRIM值上溢或下溢中断使能
3	MISS_EN	校准参考同步丢失中断使能
2	PS_EN	程序TRIM单次计数调整中断使能
1	END_EN	自动Trim结束中断使能
0	UD_EN	计数器下溢出中断使能。

35.5.6 CTRIM 中断和状态寄存器 (CTRIM_ISR)

偏移地址: 0x018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DIR	OKF	OVF	MISS	PSF	END	UDF
									RO	RO	RO	RO	RO	RO	RO

位	标记	功能描述
31:6	Reserved	保留
6	DIR	频率误差方向，显示实际频率低于目标频率还是高于目标频率 0: 递增计数方向，计数时钟频率实际频率高于目标频率 1: 递减计数方向，计数时钟频率实际频率低于目标频率
5	OKF	自动Trim结束结果判断 1: 满足FLIM要求 0: 不满足FLIM要求
4	OVF	TRIM值上溢或下溢时，该标志由硬件置1。如果TRIM值修改到范围内该位会自动清除。也可以通过ICR寄存器清除该标志。
3	MISS	校准参考同步丢失(频率过快并超范围)，计数到0后计数值递增计数到0xFFFF没有检测到同步信号。
2	PSF	程序TRIM单次计数调整标志
1	END	自动Trim结束标志；CEN保持不变，需要软件清除。 Trim结束条件1: 已达到设定的目标频率范围内 Trim结束条件2: trim step从其他值变成1
0	UDF	计数器下溢出标志，频率误差计数器计数到0时，该位硬件置1。

35.5.7 CTRIM 中断状态清除寄存器 (CTRIM_ICR)

偏移地址: 0x01C

复位值: 0x0000001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7							6							5							4		
3							2							1							0		
Reserved							OKF							OVF							MISS		
							R1W0							R1W0							R1W0		
														PSF							END		
														R1W0							R1W0		
																					UDF		
																					R1W0		

位	标记	功能描述
31:6	Reserved	保留
5	OKF	自动Trim结束结果标志清除。写0清除
4	OVF	TRIM值上溢或下溢中断标志清除。写0清除
3	MISS	校准参考同步丢失中断标志清除。写0清除
2	PSF	程序TRIM单次计数调整中断标志清除。写0清除
1	END	自动Trim结束中断标志清除。写0清除
0	UDF	计数器下溢出中断标志清除。写0清除

35.5.8 CTRIM 误差计数值 (CTRIM_FCAP)

偏移地址: 0x020

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCAP															
RO															

位	标记	功能描述
31:16	RSV	保留位
15:0	FCAP	频率误差捕获值

35.5.9 CTRIM VALUE 寄存器 (CTRIM_TVAL)

偏移地址: 0x024

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TVAL															
RO															

位	标记	功能描述
31:16	RSV	保留位
15:0	TVAL	自动TRIM结果, 自动TRIM结束需要将TVAL的值写入相应的TRIM寄存器中

35.5.10 CTRIM FLIM 寄存器 (CTRIM_FLIM)

偏移地址: 0x028

复位值: 0x0000 0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FLIM							
								RW							

位	标记	功能描述
31:8	RSV	保留位
7:0	FLIM	误差允许值

36. 看门狗 (IWDG)

36.1 简介

此器件具有一个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。此独立看门狗外设可用于检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

独立看门狗 (IWDG) 由其专用低速时钟 (RC10K) 驱动，因此即便在主时钟发生故障时仍然保持工作状态。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低场合。

36.1.1 基本特性

- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供
- 复位条件
 - 当递减计数器值小于 0x000 时复位（如果看门狗已激活）
 - 在窗口之外重载递减计数器时复位（如果看门狗已激活）

36.2 功能描述

36.2.1 框图

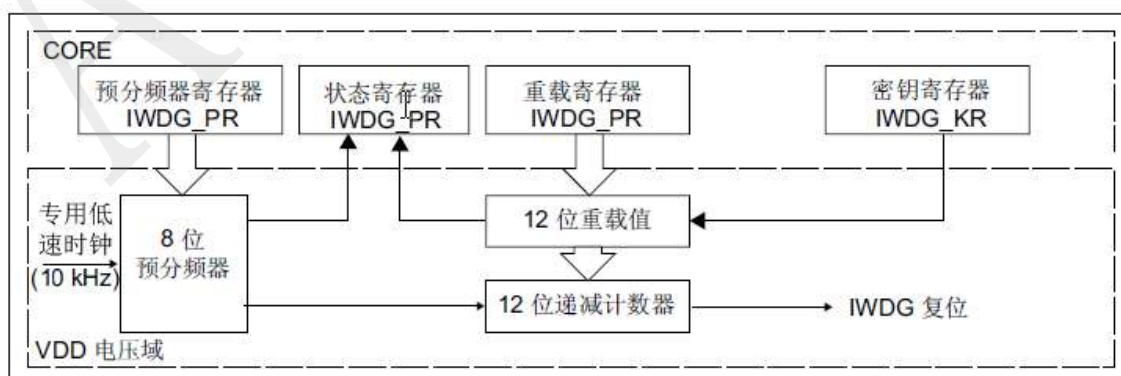


图 36-1 独立看门狗框图

通过向键寄存器 (IWDT_KR) 中写入值 0x0000 CCCC 来启动独立看门狗时, 计数器开始从复位值 0xFFFF 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号 (IWDT 复位)。

任何时候将键值 0x0000 AAAA 写到 IWDT_KR 寄存器中, IWDT_RLR 的值就会被重载到计数器, 从而避免产生看门狗复位。

36.2.2 IWDT 窗口功能

通过在 IWDT_WINR 寄存器中设置合适的窗口, IWDT 也可以用作窗口看门狗。

当计数器值大于窗口寄存器 (IWDT_WINR) 中存储的值时, 如果执行重载操作, 则会产生复位。

IWDT_WINR 的默认值为 0x0000 0FFF, 因此, 如果不更新此默认值, 将禁止窗口选项。

窗口值一经更改, 便执行重载操作, 以便将递减计数器复位为 IWDT_RLR 值, 并方便计算周期数以生成下一次重载。

使能窗口选项时配置 IWDT

1. 通过在 IWDT_KR 寄存器中写入 0x0000 CCCC 来使能 IWDT。
2. 通过在 IWDT_KR 寄存器中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 IWDT_CR 编程为 0~7 中的数值来配置 IWDT 预分频器。
4. 写重载寄存器 (IWDT_RLR)。
5. 等待寄存器更新 (IWDT_SR = 0x0000 0000)。
6. 写窗口寄存器 IWDT_WINR, 将自动刷新计数器值为 IWDT_RLR 的值。

禁止窗口选项时配置 IWDT

不使用窗口选项时, 可按以下步骤配置 IWDT:

1. 通过在 IWDT_KR 寄存器中写入 0x0000 CCCC 来使能 IWDT。
2. 通过在 IWDT_KR 寄存器中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 IWDT_CR 编程为 0~7 中的数值来配置 IWDT 预分频器。
4. 写重载寄存器 (IWDT_RLR)。
5. 等待寄存器更新 (IWDT_SR = 0x0000 0000)。

6. 刷新计数器值为 IWDT_RLR 的值 (IWDT_KR = 0x0000 AAAA)

36.2.3 硬件看门狗

如果通过器件选项位使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向键寄存器写入相应的值，或者在窗口内部重载了递减计数器，则系统会产生复位。

36.2.4 寄存器访问保护

IWDT_CR、IWDT_RLR 和 IWDT_WINR 寄存器具有写访问保护。若要修改寄存器，必须首先对 IWDT_KR 寄存器写入代码 0x0000 5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这意味着重装操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

36.2.5 停止和待机模式下的行为

- 1) 通过写序列 0x1234,0x5678 可以关闭 IWDT
- 2) 深度休眠模式可以通过 CR.DSDIS 位控制 IWDT 是否运行。

36.2.6 调试模式

当微控制器进入调试模式时（内核停止），IWDT 计数器会根据 DBG 模块中的 DBG_IWDT_STOP 配置位选择继续正常工作或者停止工作。

DBG_IWDT_STOP=0 时 debug mode wdt 运行

DBG_IWDT_STOP=1 时 debug mode wdt 停止

36.3 IWDT 寄存器描述

IWDT 基地址：0x40003000

寄存器	偏移地址	描述
IWDT_KR	0x000	IWDT键值寄存器
IWDT_CR	0x004	IWDT控制寄存器
IWDT_RLR	0x008	IWDT重载寄存器
IWDT_SR	0x00C	IWDT状态寄存器
IWDT_WINR	0x010	IWDT窗口寄存器

表 36-1 IWDT 寄存器列表

36.3.1 键值寄存器（IWDT_KR）

偏移地址：0x000

复位值： 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY															
WO															

位	符号	描述
31:16	Reserved	保留位
15:0	KEY	序列 必须每隔一段时间便通过软件对这些位写入键值 0xAAAA重载WDT计数值，否则当计数器计数到 0 时，看门狗会产生复位。 写入键值 0x5555 可使能对 IWDT_PR、IWDT_RLR 和 IWDT_WINR IWDT_CR寄存器的访问。 写入键值 0xCCCC 可启动看门狗（选中硬件看门狗选项的情况除外）。

36.3.2 控制寄存器（IWDT_CR）

偏移地址：0x004

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DSDIS	SINT	INTEN	PR			
									RW	RW	RW	RW			

位	符号	描述
31:6	Reserved	保留位
6	DSDIS	Deepsleep 暂停控制 0: deepsleep模式下IWDT运行 1: deepsleep模式下IWDT暂停
5	SINT	中断功能选择，默认复位
4	INTEN	中断使能
3:0	PRS	111: 512分频 110: 256分频 101: 128分频 100: 64分频 011: 32分频 010: 16分频 001: 8分频 000: 4分频

36.3.3 重载寄存器 (IWDT_RLR)

偏移地址: 0x008

复位值: 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ARR											
				RW											

位	符号	描述
31:12	Reserved	保留位
11:0	ARR	WDT重载值

36.3.4 状态寄存器 (IWDT_SR)

偏移地址: 0x00C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											RUNF	INTF	WVU	RVU	PVU
											RO	RO	RO	RO	RO

位	符号	描述
31:5	RSV	保留位
4	RUNF	WDT运行标志
3	INTF	中断标志, 写0清除
2	WVU	WIN 寄存器写更新完成标志 该位由硬件置 1 以指示窗口值正在更新。当完成重载值更新操作后 (需要多达5 个 WDTCLK 周期), 会通过硬件将该位复位。 窗口值只有在 WVU 位为 0 时才可更新。 此位只有在通用“窗口”= 1 时才生成
1	RVU	RLR 寄存器写更新完成标志 该位由硬件置 1 以指示重载值正在更新。当完成重载值更新操作后 (需要多达5 个 WDTCLK 周

		期)，会通过硬件将该位复位。 重载值只有在 RVU 位为 0 时才可更新。
0	PVU	CR寄存器的PRS 寄存器写更新完成标志

36.3.5 窗口寄存器 (IWDT_WINR)

偏移地址：0x010

复位值： 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WINR											
				RW											

位	符号	描述
31:12	Reserved	保留位
11:0	WIN	窗口比较器比较值

这些位受写访问保护，这些位包含用于与递减计数器进行比较的窗口值上限。

为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x0 时必须重载
若要更改重载值，IWDT_SR 中的 WVU 位必须为 0。

37. 窗口看门狗 (WWDT)

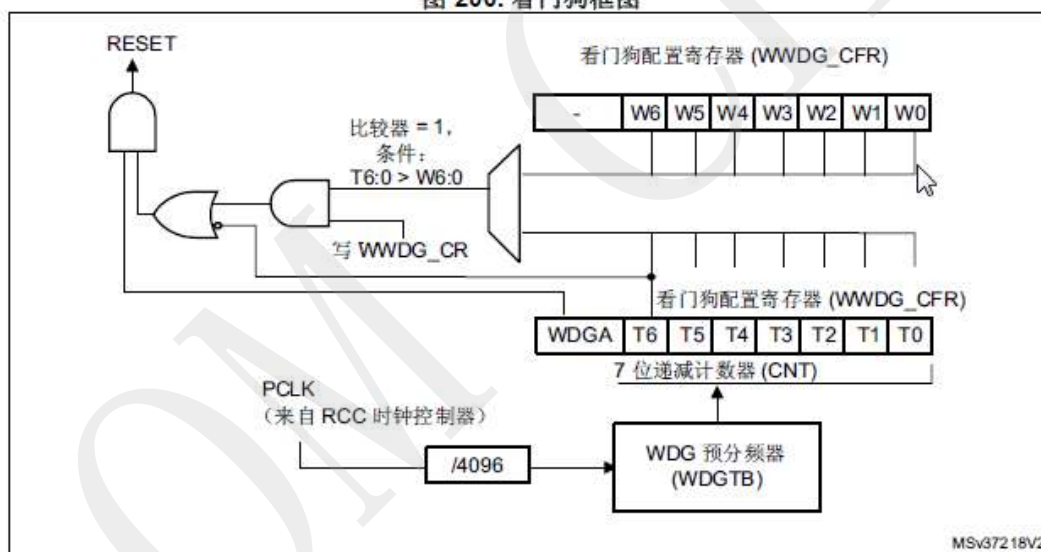
37.1 简介

系统窗口看门狗 (WWDG) 通常被用来监测, 由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非程序在 T6 位变成 0 前刷新递减计数器的值, 否则看门狗电路在达到预置的时间周期时, 会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值, 也会产生 MCU 复位。这意味着必须在限定的时间窗口内刷新计数器。

WWDG 时钟由 APB1 时钟经预分频后提供, 通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用程序。

图 200. 看门狗框图



37.1.1 基本特性

- 可编程的自由运行递减计数器
- 复位条件
- 当递减计数器值小于 0x40 时复位 (如果看门狗已激活)
- 在窗口之外重载递减计数器时复位 (如果看门狗已激活)
- 提前唤醒中断 (EWI): 当递减计数器等于 0x40 时触发 (如果已使能且看门狗已激活)

37.2 功能描述

如果激活看门狗 (WWDG_CR 寄存器中的 WDGA 位置 1)，则当 7 位递减计数器 (T[6:0]位) 从 0x40 递减到 0x3F (T6 已清零) 时会引发复位。当计数器值大于窗口寄存器中所存储的值时，如果软件重载计数器，则会产生复位。

37.2.1 使能 WWDT

在系统复位后，看门狗总是处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

37.2.2 控制递减计数器

递减计数器处于自由运行状态，当使能看门狗时，必须将 T6 位置 1，以防止立即复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频器的状态是未知的（请参见图 201）。配置寄存器 (WWDG_CGR) 包含窗口的上限：为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x3F 时必须重载。图 201 介绍了窗口看门狗的工作过程。

37.2.3 中断特性

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断(ENI)。通过设置 WWDG_CGR 寄存器中的 ENI 位使能 ENI 中断。当递减计数器的值为 0x40 时，将生成中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序 (ISR) 可用来重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

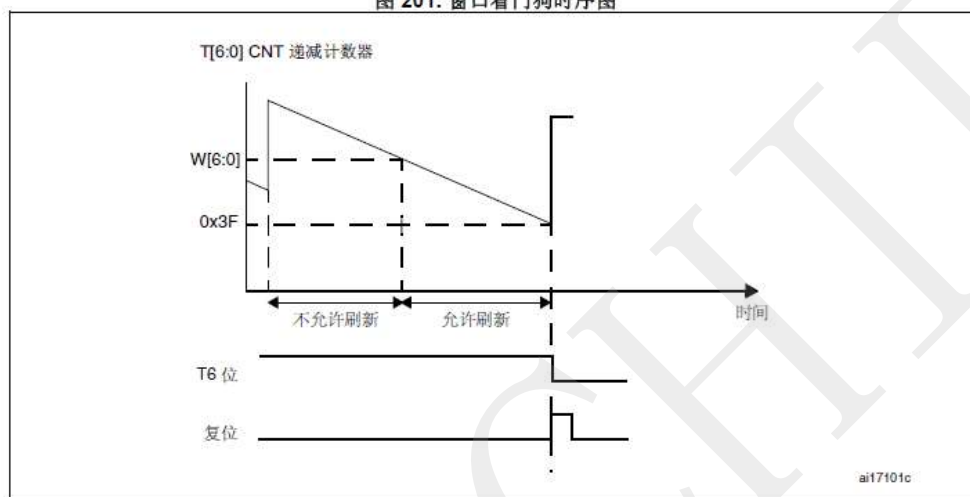
通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

37.2.4 设置看门狗超时

可以使用下图中的公式来计算 WWDG 超时。

警告： 写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。

图 201. 窗口看门狗时序图



超时值的计算公式如下：

$$t_{\text{WWDG}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{WDGTB}[1:0]} \times (\text{T}[5:0] + 1) \quad (\text{ms})$$

其中：

t_{WWDG} : WWDG 超时

t_{PCLK1} : APB1 时钟周期，以 ms 为测量单位

4096: 对应于内部分频器的值

例如，假设 APB1 频率等于 32 MHz，将 WDGTB[1:0] 设置为 3，将 T[5:0] 设置为 63：

$$t_{\text{WWDG}} = 1/32000 \times 4096 \times 2^3 \times (63 + 1) = 65.54 \text{ ms}$$

有关 t_{WWDG} 的最小值和最大值，请参见数据手册。

37.2.5 调试模式

当微控制器进入调试模式 (Cortex®-M0+ 内核停止) 时，WWDG 计数器会根据 DBG 模块中的 DBG_WWDG_STOP 配置位选择继续正常工作或者停止工作。

37.3 WWDT 寄存器描述

WWDT 基地址：0x40002C00

寄存器	偏移地址	描述
WWDT_CR	0x000	WWDT控制寄存器
WWDT_CFR	0x004	WWDT配置寄存器
WWDT_SR	0x008	WWDT状态寄存器

表 37-1 WWDT 寄存器列表

37.3.1 控制寄存器（WWDT_CR）

偏移地址：0x000

复位值：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EN	WCNT						
								RW	RW						

位	符号	描述
31:8	Reserved	保留位
7	EN	WDT 使能，只能使能不能软件关闭 1：使能 0：禁止
6:0	WCNT	计数器值

37.3.2 配置寄存器（WWDT_CFR）

偏移地址：0x004

复位值： 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PRS				RSV	IEN	RSV	WVAL							
	RW					RW		RW							

位	符号	描述
31:14	Reserved	保留位，读为0
13:11	PRS	WDT计数器预分频 000 4096*1 001 4096*2 010 4096*4 011 4096* 8 100 4096*16 101 4096*32 110 4096*64 111 4096* 128
10	RSV	保留位，读为0
9	IEN	WWDT中断使能 1：使能 0：禁止
8:7	RSV	保留位，读为0
6:0	WVAL	窗口比较器比较值

37.3.3 配置寄存器（WWDT_SR）

偏移地址：0x008

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															IF
															RW

位	符号	描述
31:1	RSV	保留位
0	IF	中断标志，硬件置位，软件写0清0

38. 高级定时器 (STIM)

38.1 STIM 简介

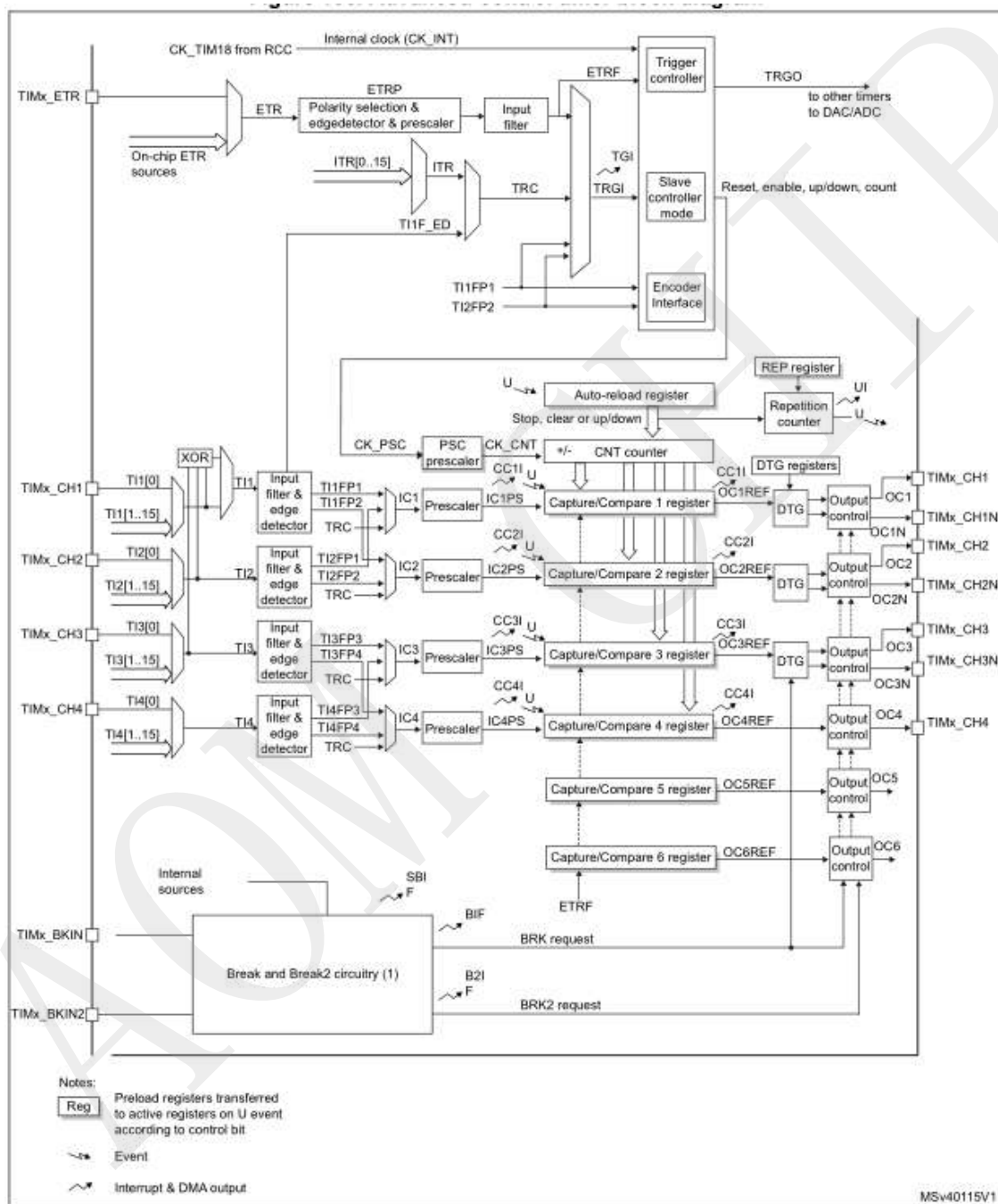
STIM 是一个 16 位与 ST TIM1 兼容的定时器。高级控制定时器由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。

使用定时器预分频器和时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

38.2 STIM 主要特性

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1~65535 之间的任意数值
- 多达 4 个独立通道：
 - 输入捕获
 - 输出比较
 - PWM 生成(边缘或中间对齐模式)
 - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA：
 - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
 - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获

- 输出比较
- 刹车信号输入
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理



38.3 STIM 功能描述

38.3.1 STIM 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器(TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)
- 重复次数寄存器 (TIMx_RCR)

自动装载寄存器是预先装载的，写或读自动重载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件(向下计数时的下溢条件)并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。注意，在设置了 TIMx_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个(在 TIMx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图51 当预分频器的参数从1变到2时，计数器的时序图

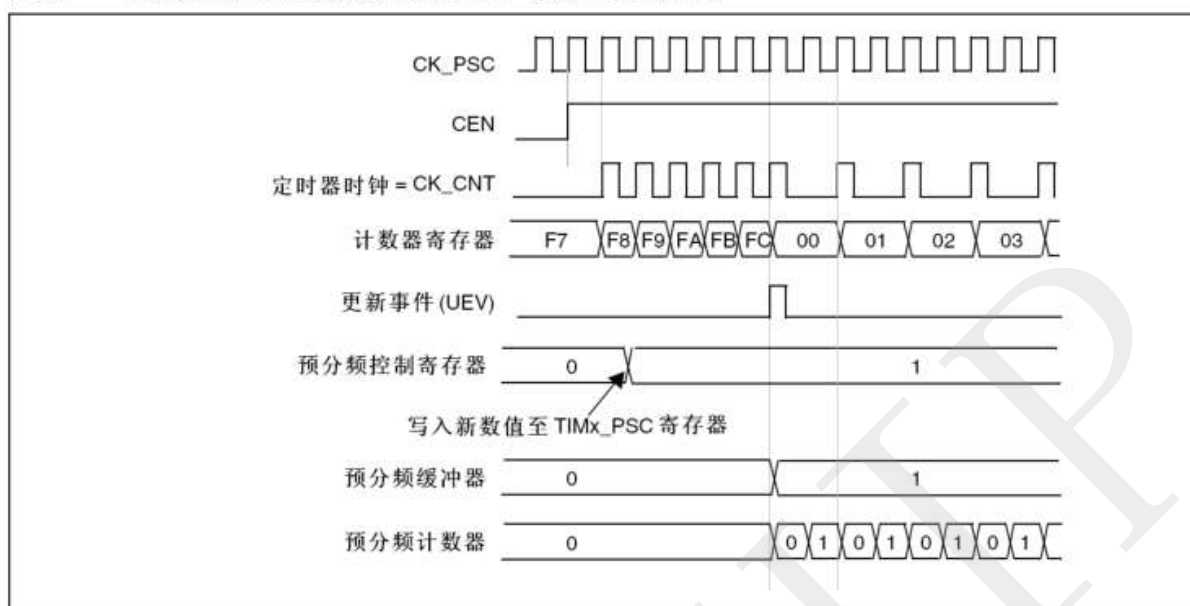
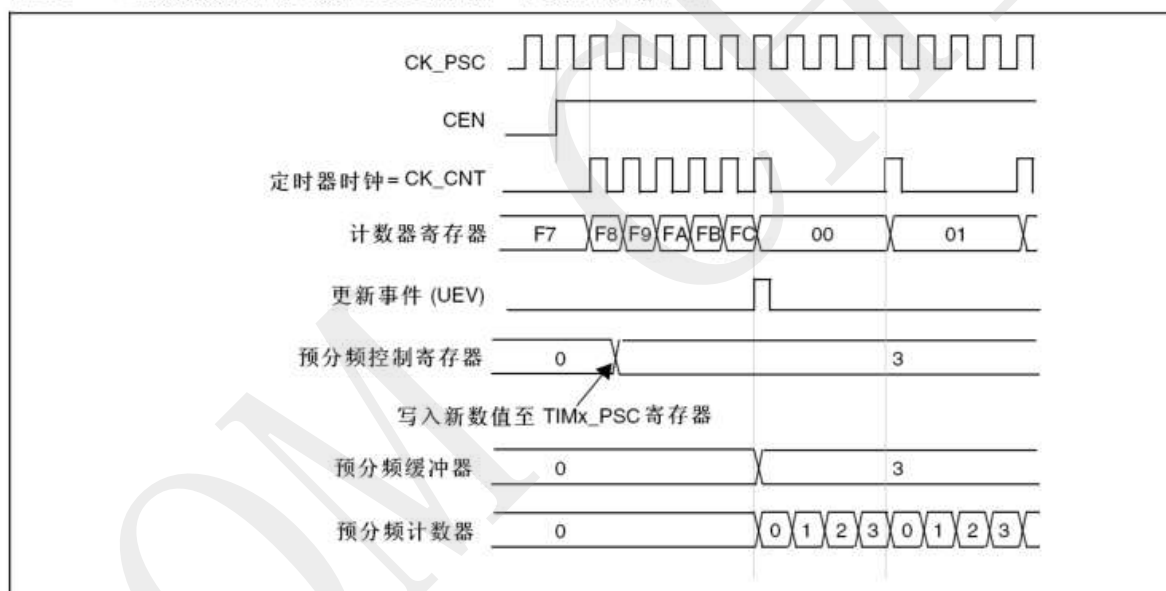


图52 当预分频器的参数从1变到4时，计数器的时序图



38.3.2 计数器模式

向上计数模式

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

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数(TIMx_RCR)时，产生更新事件(UEV)；否则每次计数器溢出时才产生更新事件。

在 TIMx_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位也同样

可以产生一个更新事件。

设置 TIMx_CR1 寄存器中的 UDIS 位, 可以禁止更新事件; 这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清 0 之前, 将不产生更新事件。但是在应该产生更新事件时, 计数器仍会被清 0, 同时预分频器的计数也被清 0(但预分频器的数值不变)。此外, 如果设置了 TIMx_CR1 寄存器中的 URS 位(选择更新请求), 设置 UG 位将产生一个更新事件 UEV, 但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时, 同时产生更新和捕获中断。

当发生一个更新事件时, 所有的寄存器都被更新, 硬件同时(依据 URS 位)设置更新标志位(TIMx_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIMx_RCR 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值(TIMx_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值(TIMx_PSC 寄存器的内容)。

下图给出一些例子, 当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

图53 计数器时序图, 内部时钟分频因子为1

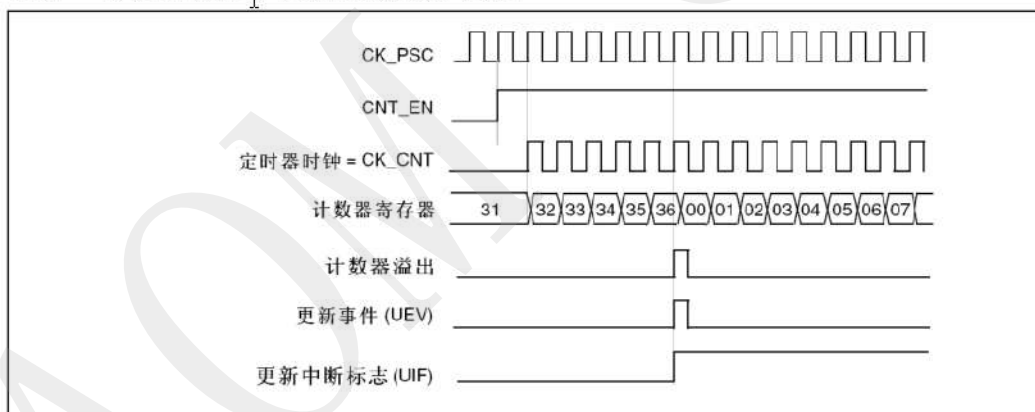


图54 计数器时序图，内部时钟分频因子为2

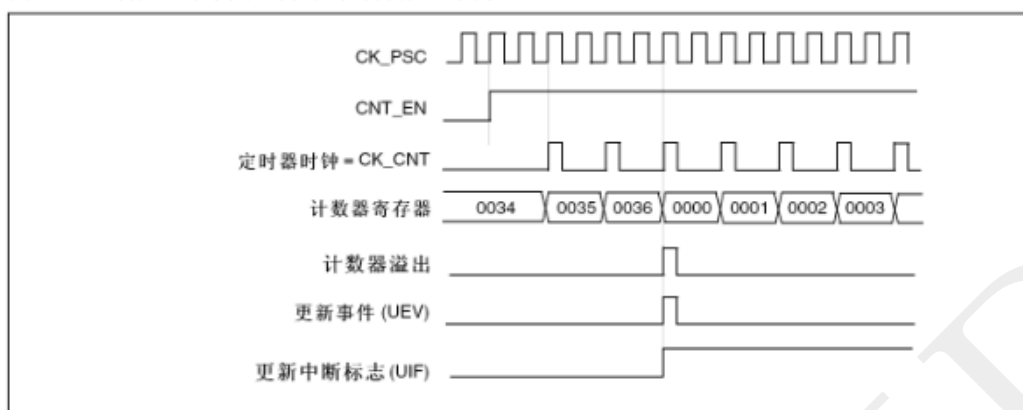


图55 计数器时序图，内部时钟分频因子为4

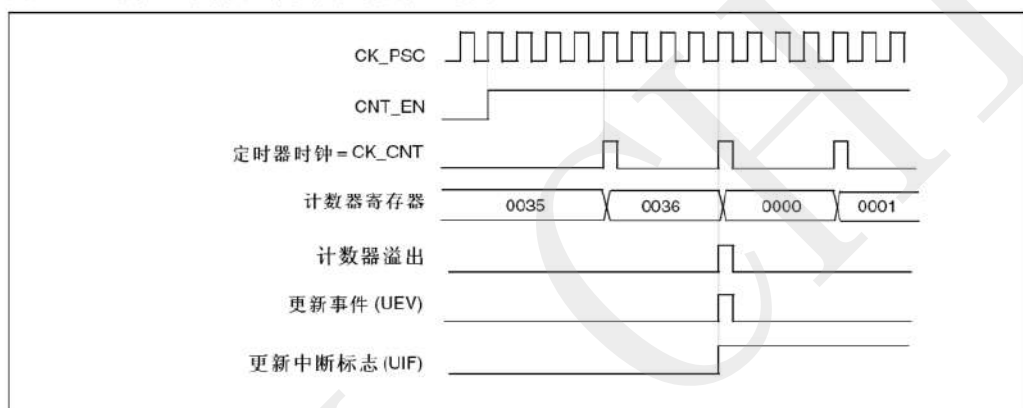


图56 计数器时序图，内部时钟分频因子为N

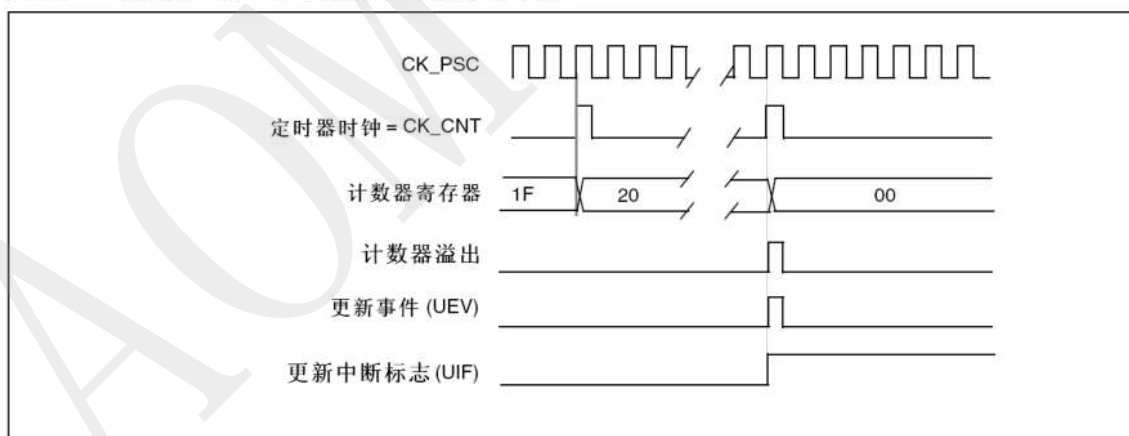


图57 计数器时序图，当ARPE=0时的更新事件(TIMx_ARR没有预装入)

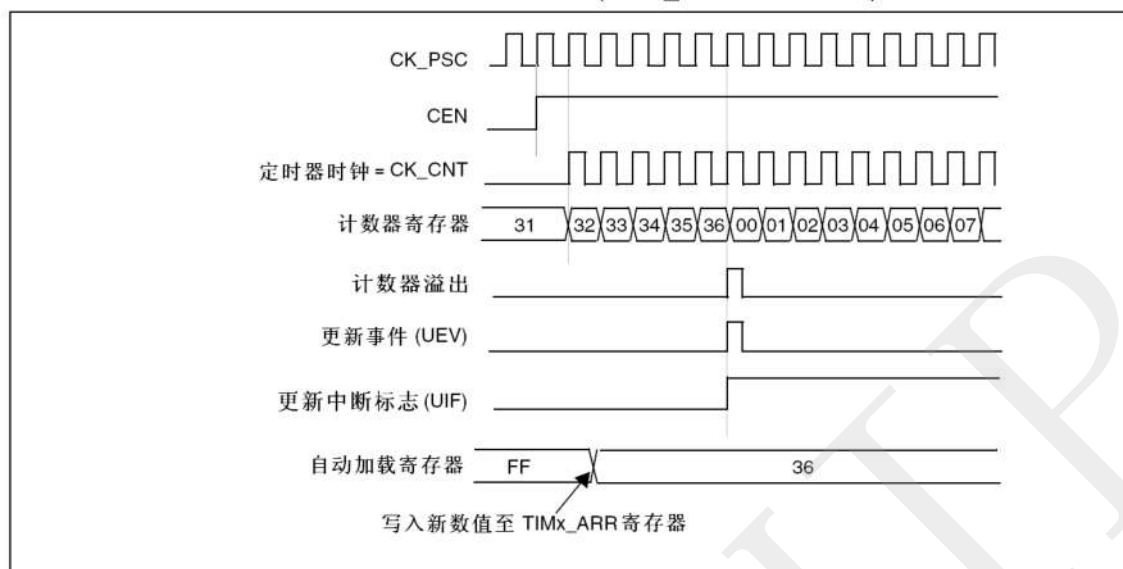
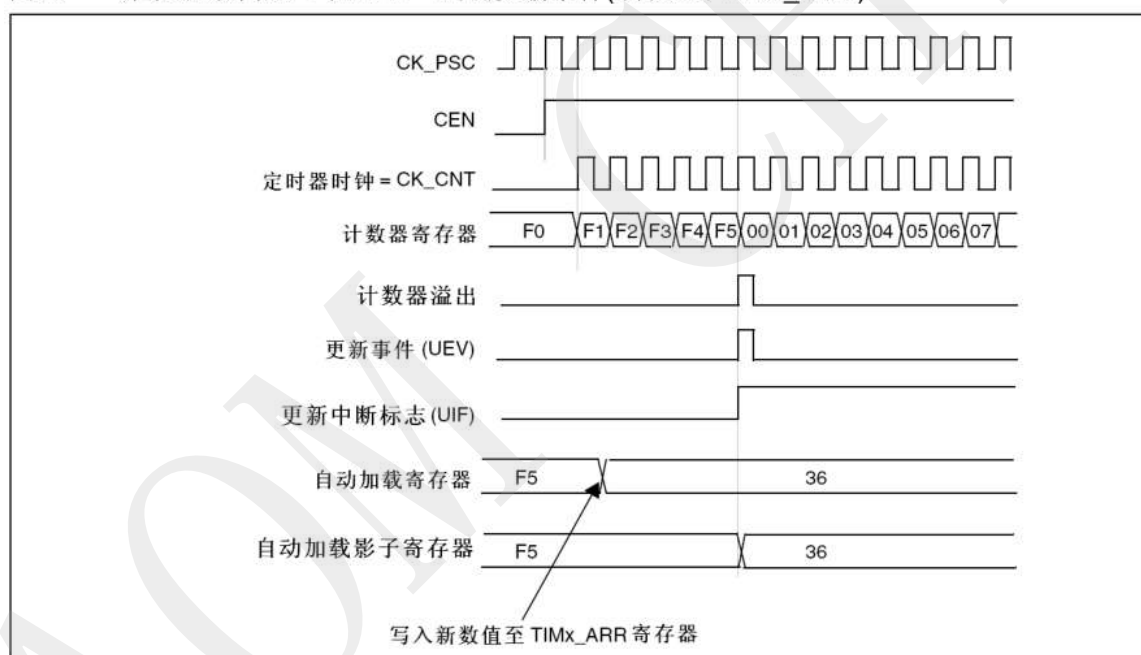


图58 计数器时序图，当ARPE=1时的更新事件(预装入了TIMx_ARR)



向下计数模式

在向下模式中，计数器从自动装入的值(TIMx_ARR 计数器的值)开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

如果使用了重复计数器，当向下计数重复了重复计数寄存器(TIMx_RCR)中设定的次数后，将产生更新事件(UEV)，否则每次计数器下溢时才产生更新事件。

在 TIMx_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位，也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始(但预分频系数不变)。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且(根据 URS 位的设置)更新标志位(TIMx_SR 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 TIMx_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载的值(TIMx_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值(TIMx_ARR 寄存器中的内容)。注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIMx_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

图59 计数器时序图，内部时钟分频因子为1

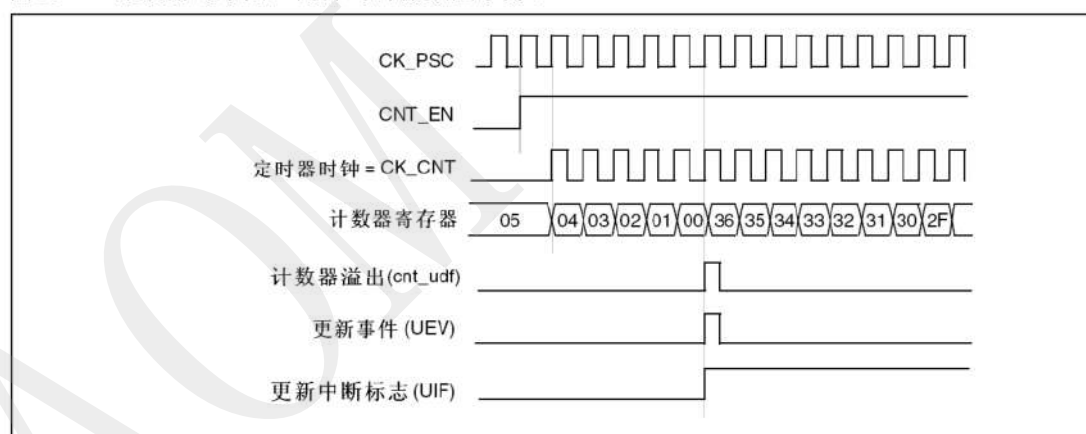


图60 计数器时序图，内部时钟分频因子为2

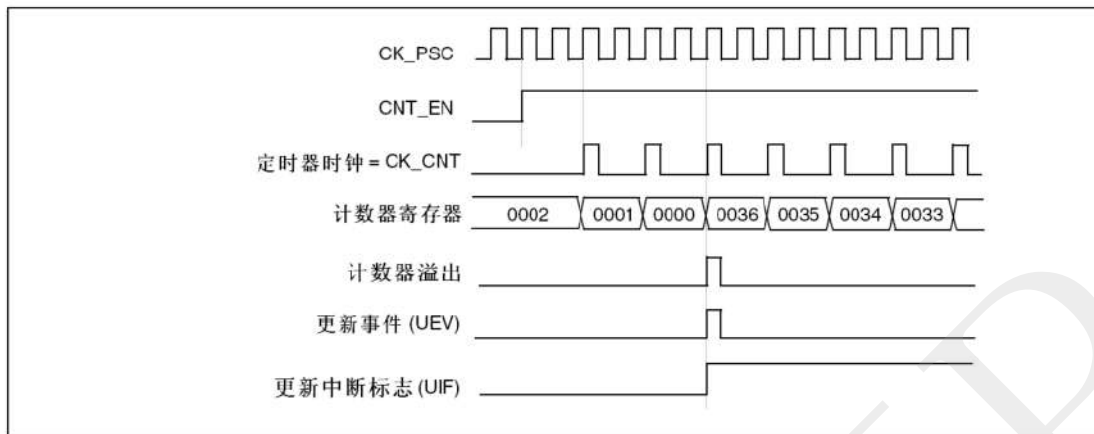


图61 计数器时序图，内部时钟分频因子为4

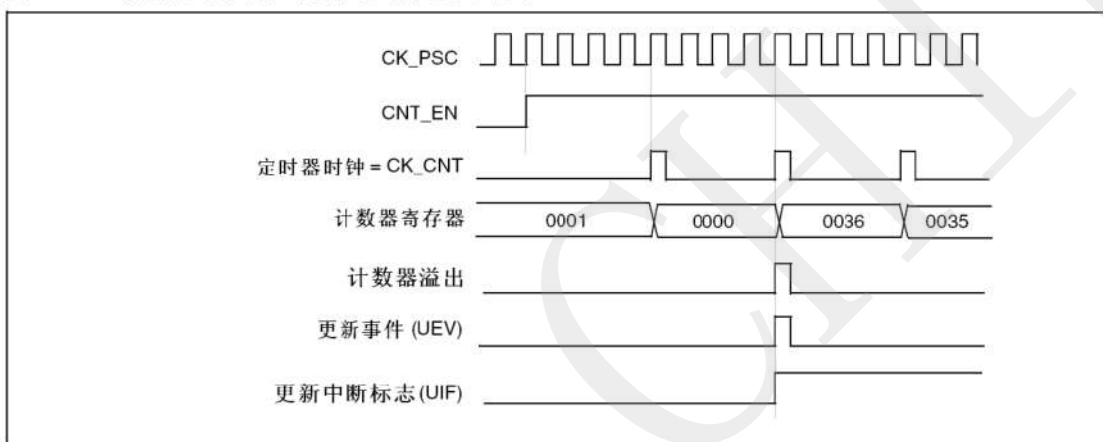


图62 计数器时序图，内部时钟分频因子为N

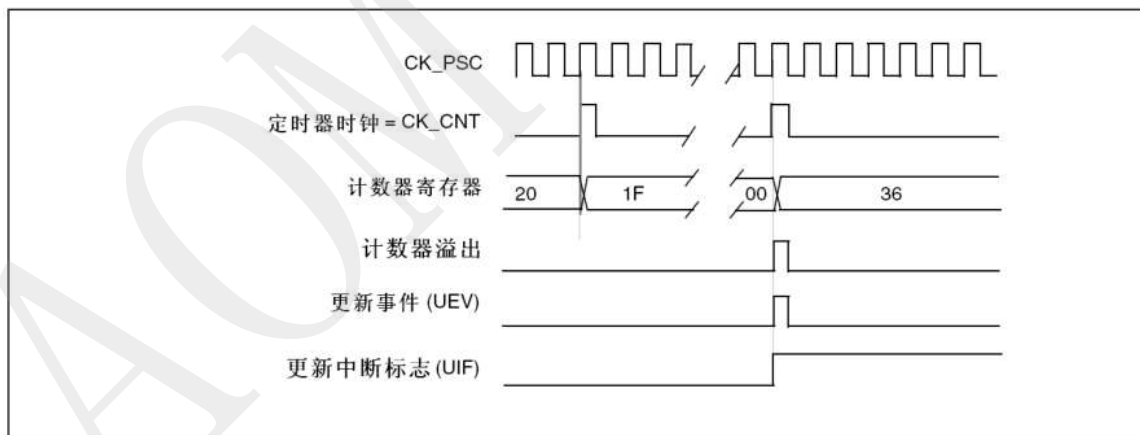
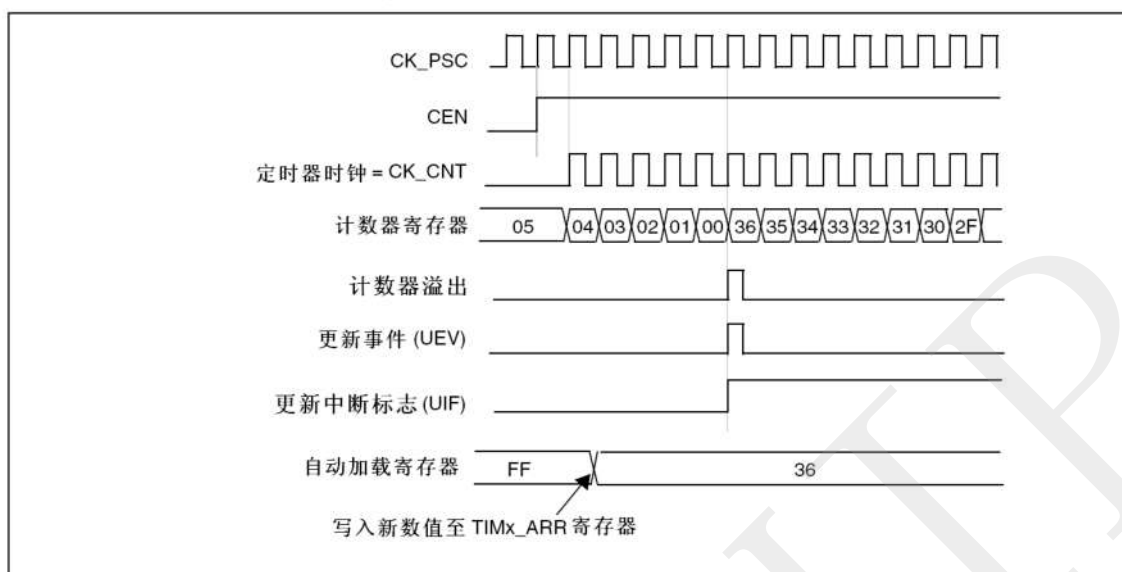


图63 计数器时序图，当没有使用重复计数器时的更新事件



中央对齐模式(向上/ 向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值(TIMx_ARR 寄存器)-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过(软件或者使用从模式控制器)

设置 TIMx_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且(根据 URS 位的设置)更新标志位(TIMx_SR 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 TIMx_RCR 寄存器中的内容

- 预分频器的缓存器被加载为预装载(TIMx_PSC 寄存器)的值。
- 当前的自动加载寄存器被更新为预装载值(TIMx_ARR 寄存器中的内容)。

注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值(计数器被装载为新的值)。

以下是一些计数器在不同时钟频率下的操作的例子：

图64 计数器时序图，内部时钟分频因子为1，TIMx_ARR=0x6

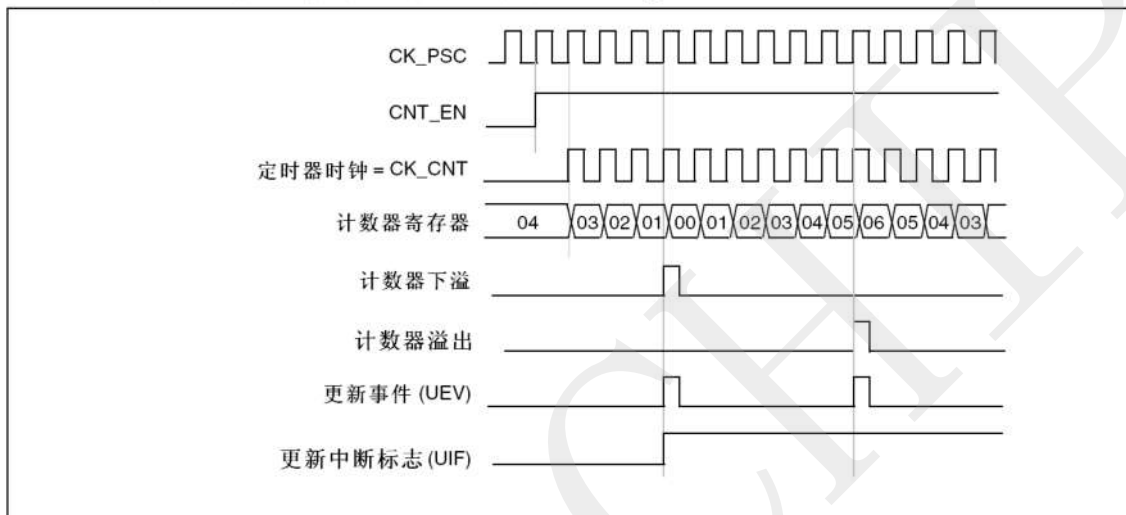


图65 计数器时序图，内部时钟分频因子为2

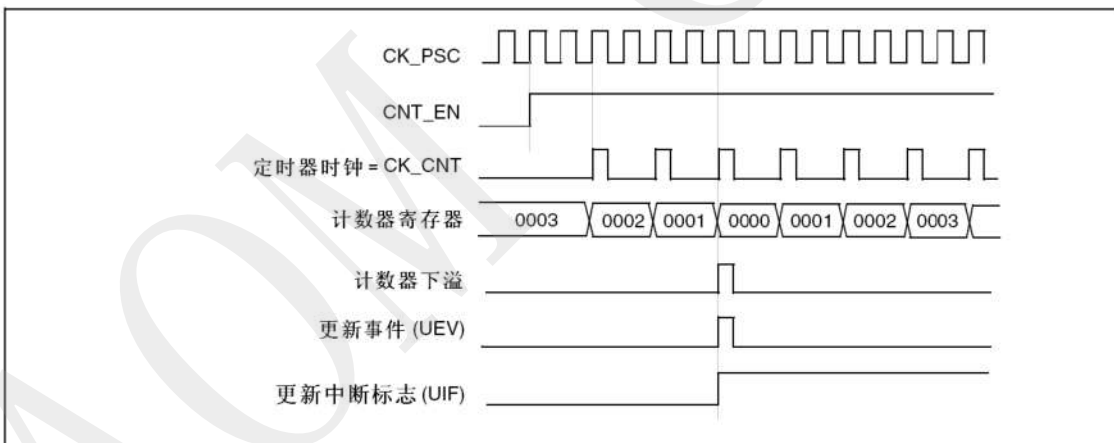


图66 计数器时序图，内部时钟分频因子为4，TIMx_ARR=0x36

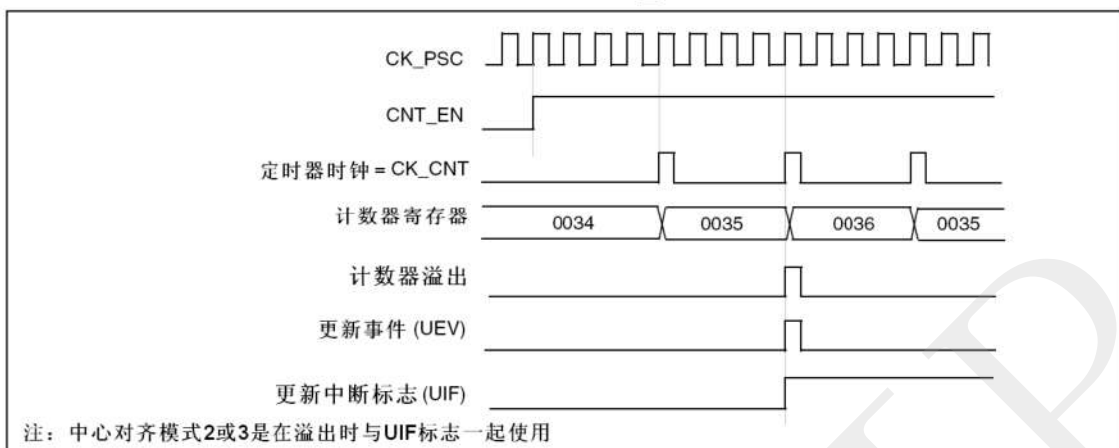


图67 计数器时序图，内部时钟分频因子为N

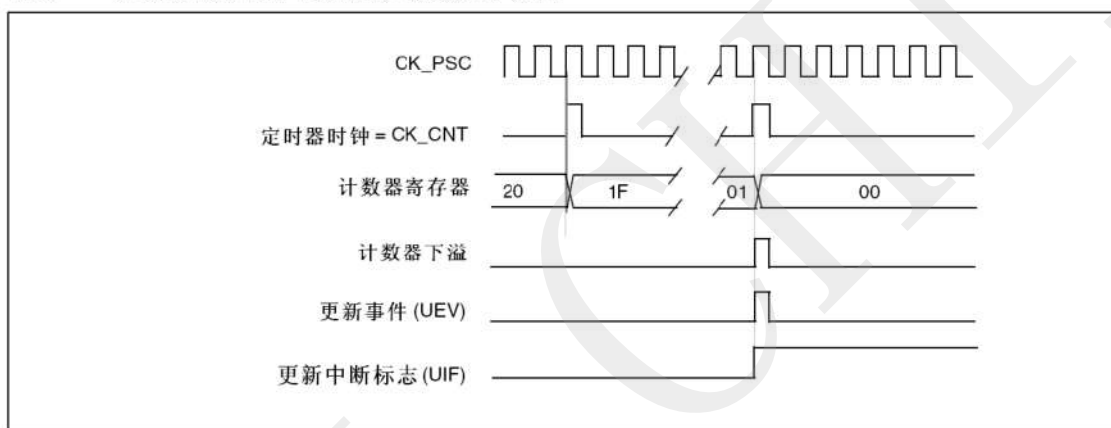


图68 计数器时序图，ARPE=1时的更新事件(计数器下溢)

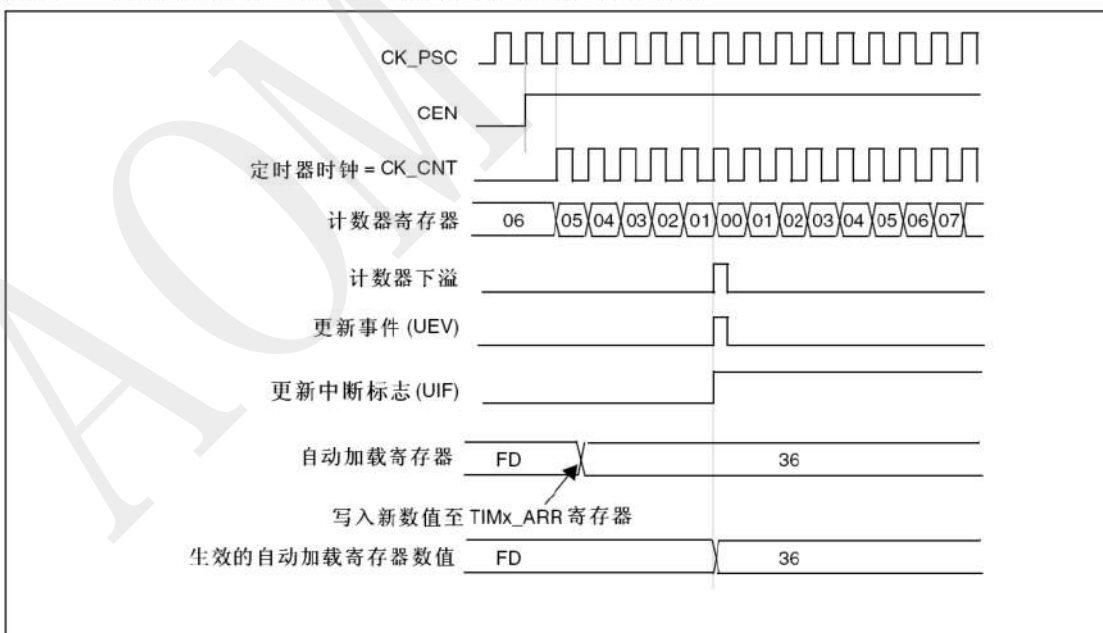
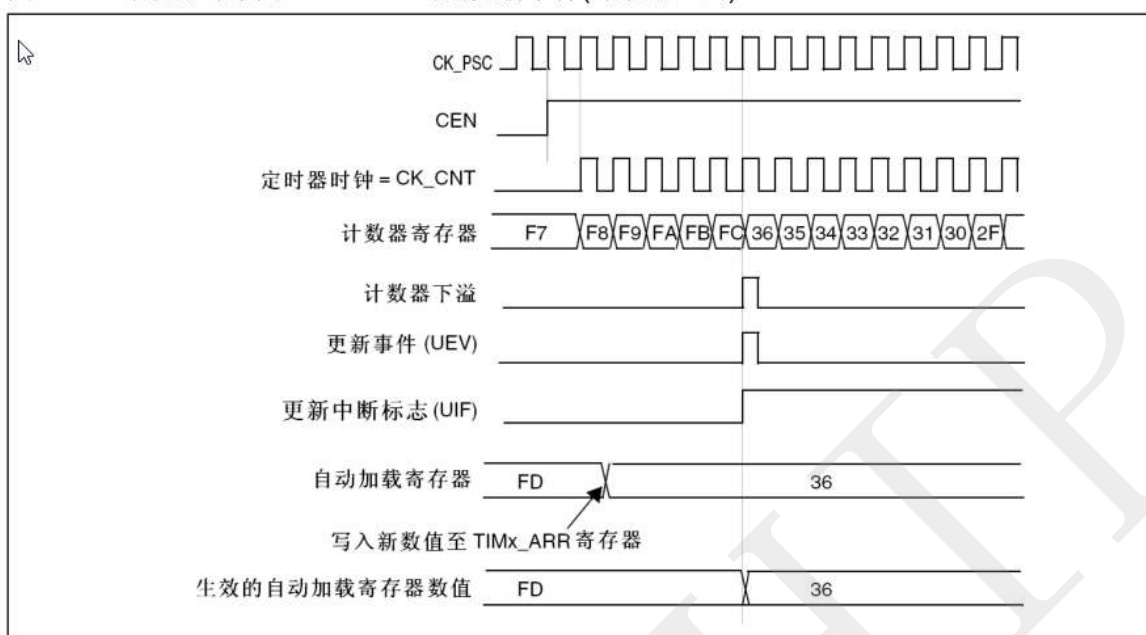


图69 计数器时序图, ARPE=1时的更新事件(计数器溢出)



38.3.3 重复计数器

35.3.1 节“时基单元”解释了计数器上溢/下溢时更新事件(UEV)是如何产生的,然而事实上它只能在重复计数达到0的时候产生。这个特性对产生PWM信号非常有用。

这意味着在每 N 次计数上溢或下溢时,数据从预装载寄存器传输到影子寄存器 (TIMx_ARR 自动重载寄存器, TIMx_PSC 预装载寄存器,还有在比较模式下的捕获/比较寄存器 TIMx_CCRx), N 是 TIMx_RCR 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减:

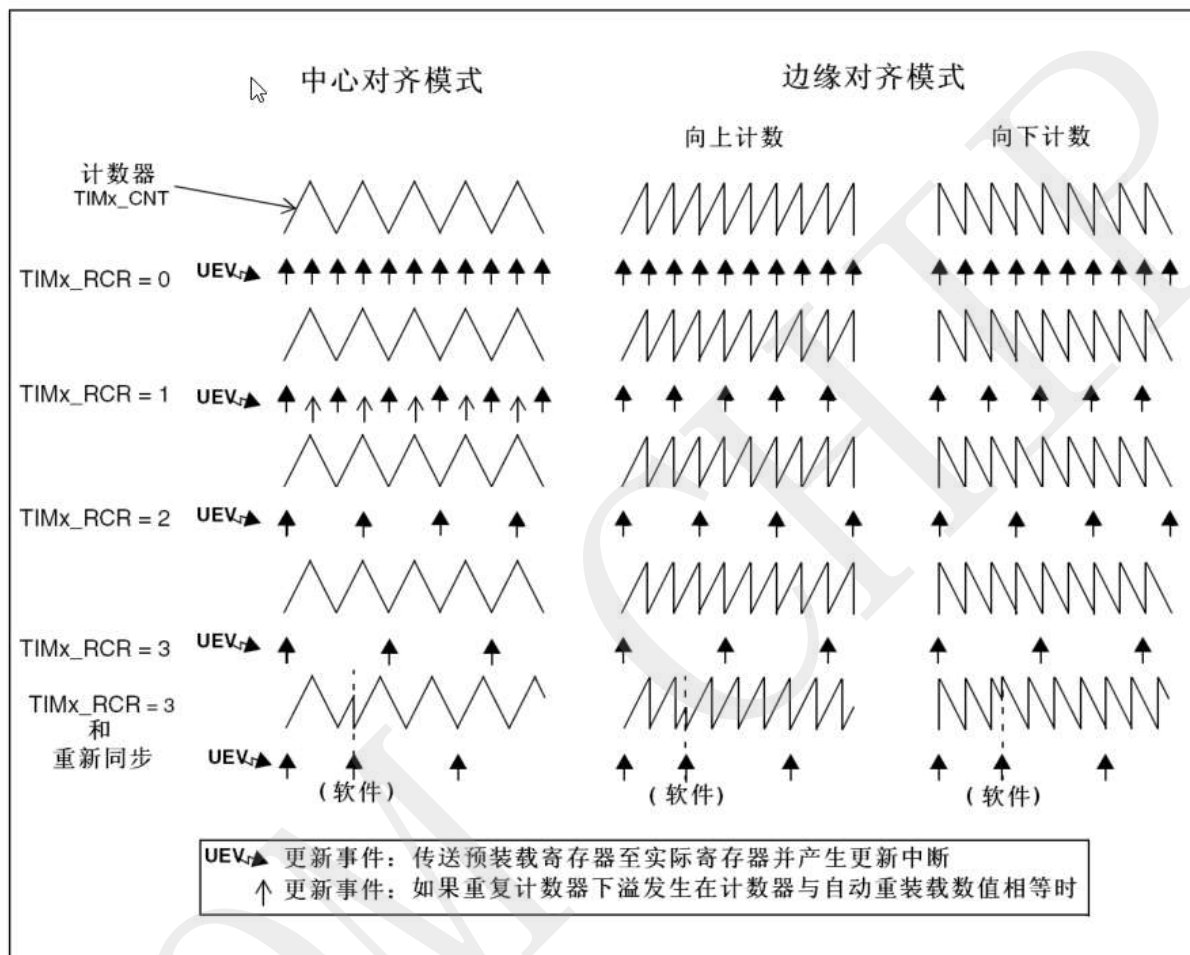
- 向上计数模式下每次计数器溢出时,
- 向下计数模式下每次计数器下溢时,
- 中央对齐模式下每次上溢和每次下溢时。

虽然这样限制了 PWM 的最大循环周期为 128,但它能够在每个 PWM 周期 2 次更新占空比。在中央对齐模式下,因为波形是对称的,如果每个 PWM 周期中仅刷新一次比较寄存器,则最大的分辨率为 $2 \times T_{ck}$ 。

图 70 重复计数器是自动加载的,重复速率是由 TIMx_RCR 寄存器的值定义(参看)。

当更新事件由软件产生(通过设置 TIMx_EGR 中的 UG 位)或者通过硬件的从模式控制器产生, 则无论重复计数器的值是多少, 立即发生更新事件, 并且 TIMx_RCR 寄存器中的内容被重载入到重复计数器。

图70 不同模式下更新速率的例子, 及TIMx_RCR的寄存器设置



38.3.4 时钟选择

计数器时钟可由下列时钟源提供:

- 内部时钟(CK_INT 使用 PCLK)
- 外部时钟模式 1: 外部输入引脚
- 外部时钟模式 2: 外部触发输入 ETR
- 内部触发输入(ITRx): 使用一个定时器作为另一个定时器的预分频器。

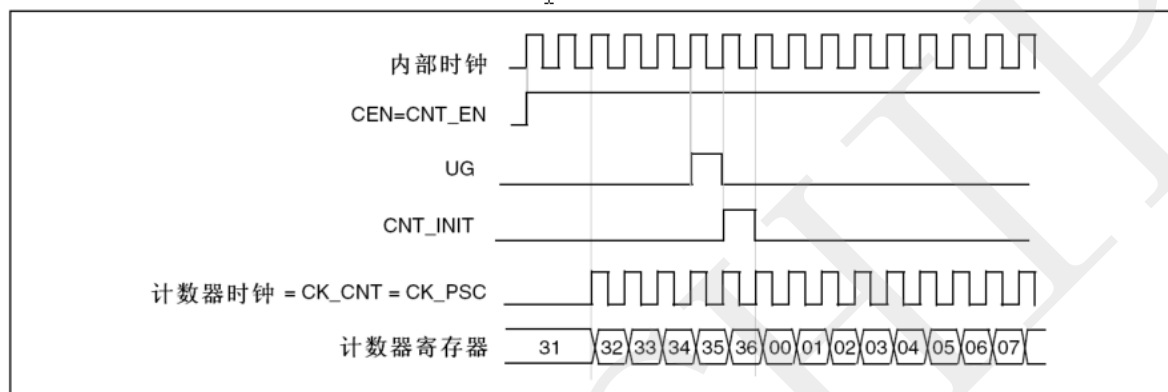
内部时钟源 内部时钟源(CK_INT)

如果禁止了从模式控制器(SMS=000)，则 CEN、DIR(TIMx_CR1 寄存器)和 UG 位(TIMx_EGR 寄存器)是事实上的控制位,并且只能被软件修改(UG 位仍被自动清除)。

只要 CEN 位被写成 1，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

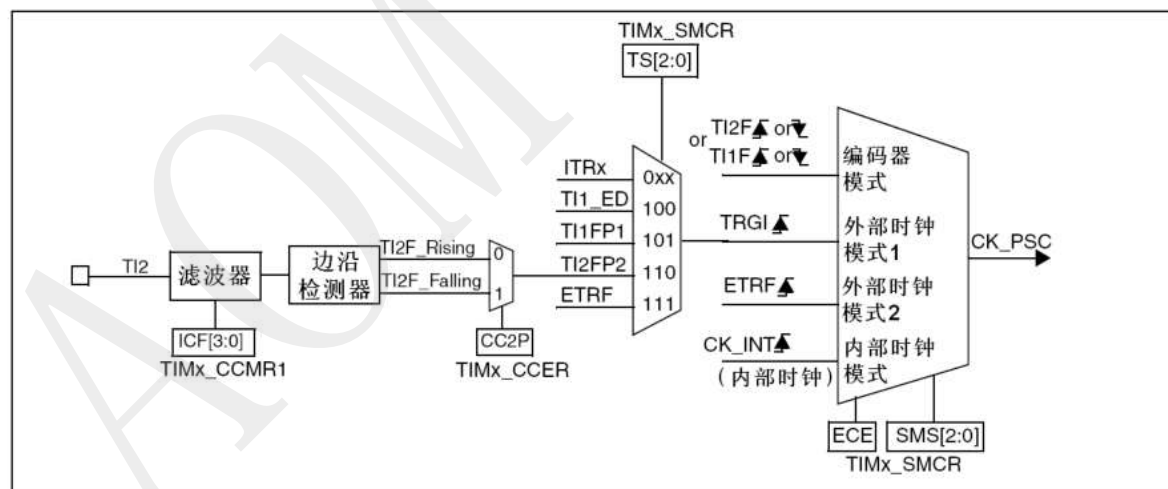
图71 一般模式下的控制电路，内部时钟分频因子为1



外部时钟源模式 外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图72 TI2外部时钟连接例子



例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

1. 配置 TIMx_CCMR1 寄存器 CC2S=01，配置通道 2 检测 TI2 输入的上升沿
2. 配置 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持 IC2F=0000)

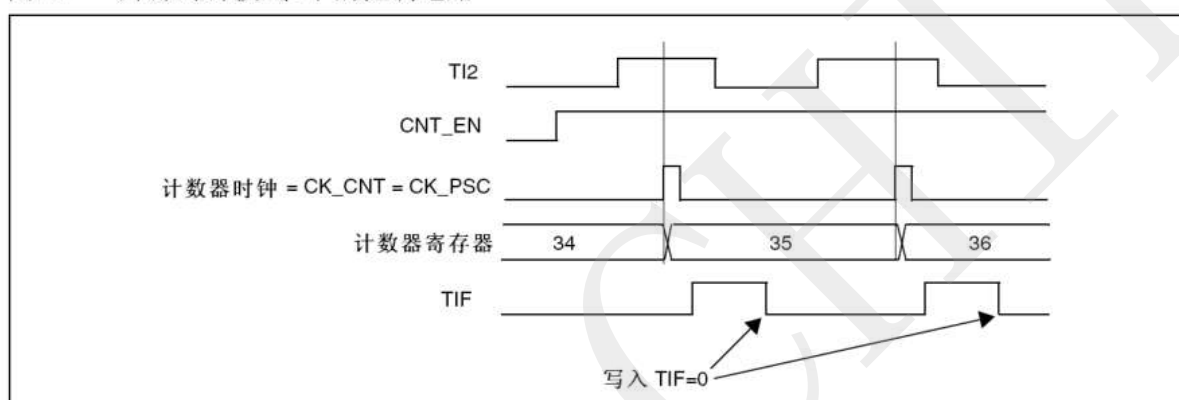
3. 配置 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性
4. 配置 TIMx_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1
5. 配置 TIMx_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源
6. 设置 TIMx_CR1 寄存器的 CEN=1，启动计数器

注：捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

图73 外部时钟模式1下的控制电路



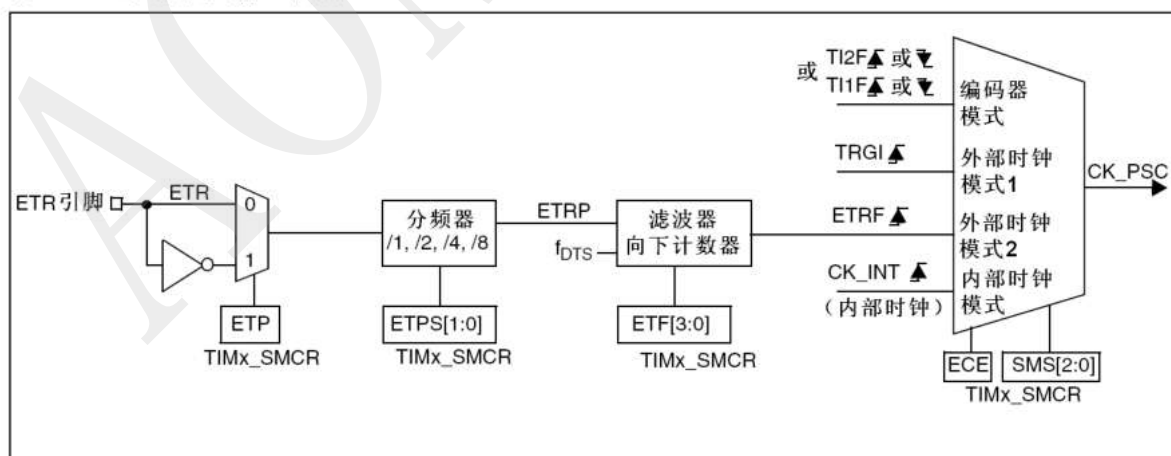
外部时钟源模式 外部时钟源模式 2

选定此模式的方法为：令 TIMx_SMCR 寄存器中的 ECE=1

计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图

图74 外部触发输入框图



例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

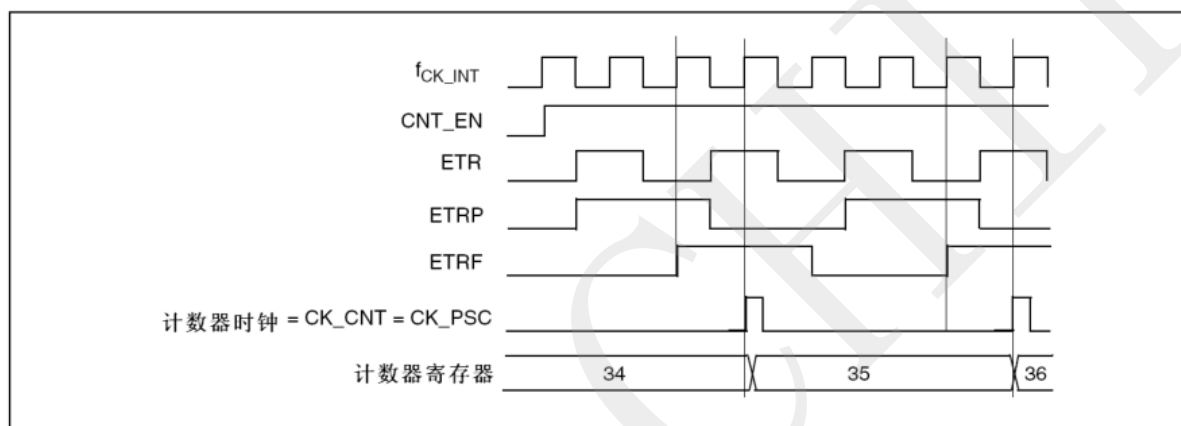
1. 本例中不需要滤波器，置 TIMx_SMCR 寄存器中的 ETF[3:0]=0000

2. 设置预分频器，置 TIMx_SMCR 寄存器中的 ETPS[1:0]=01
3. 选择 ETR 的上升沿检测，置 TIMx_SMCR 寄存器中的 ETP=0
4. 开启外部时钟模式 2，写 TIMx_SMCR 寄存器中的 ECE=1
5. 启动计数器，写 TIMx_CR1 寄存器中的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

图75 外部时钟模式2下的控制电路



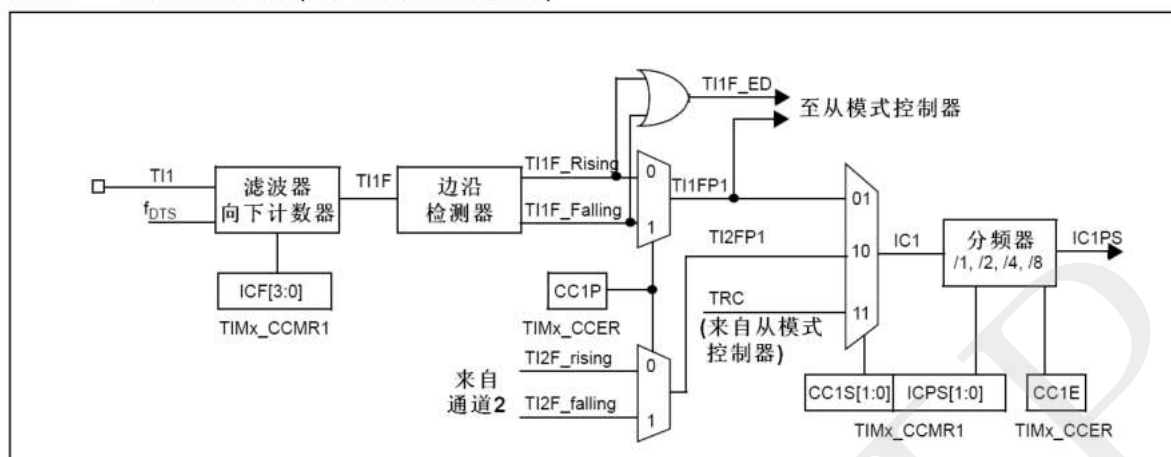
38.3.5 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器(包含影子寄存器)，包括捕获的输入部分(数字滤波、多路复用和预分频器)，和输出部分(比较器和输出控制)。

图 76 至图 79 是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号(TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器(ICxPS)。

图76 捕获/比较通道(如: 通道1输入部分)



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

图77 捕获/比较通道1的主电路

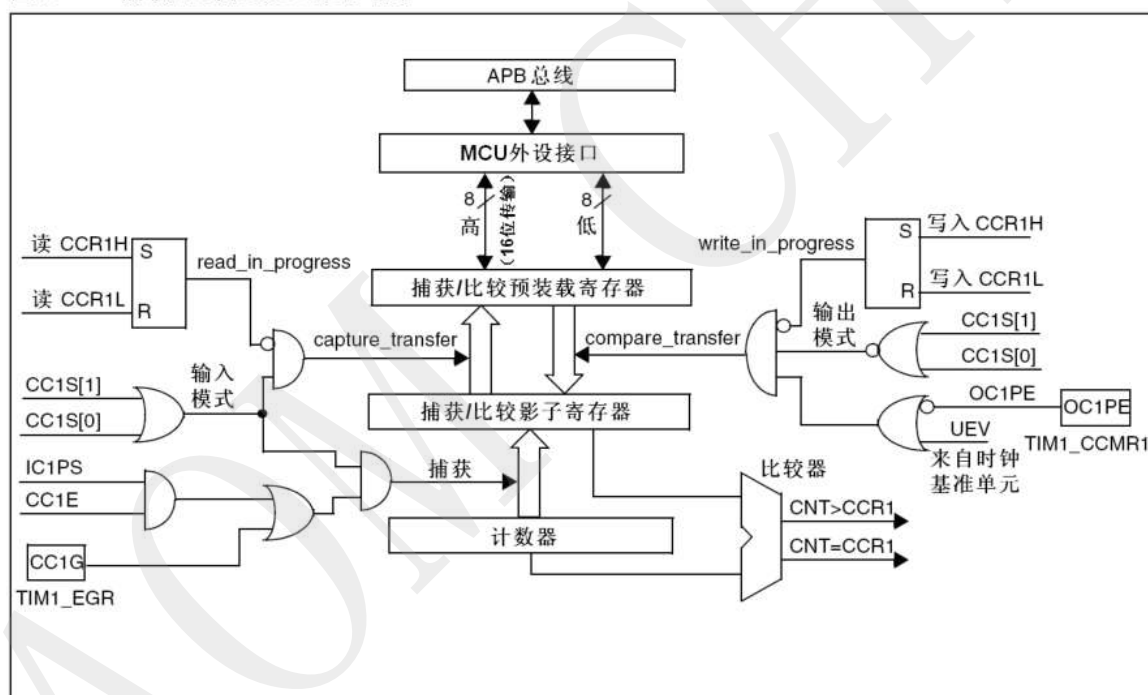


图78 捕获/比较通道的输出部分(通道1至3)

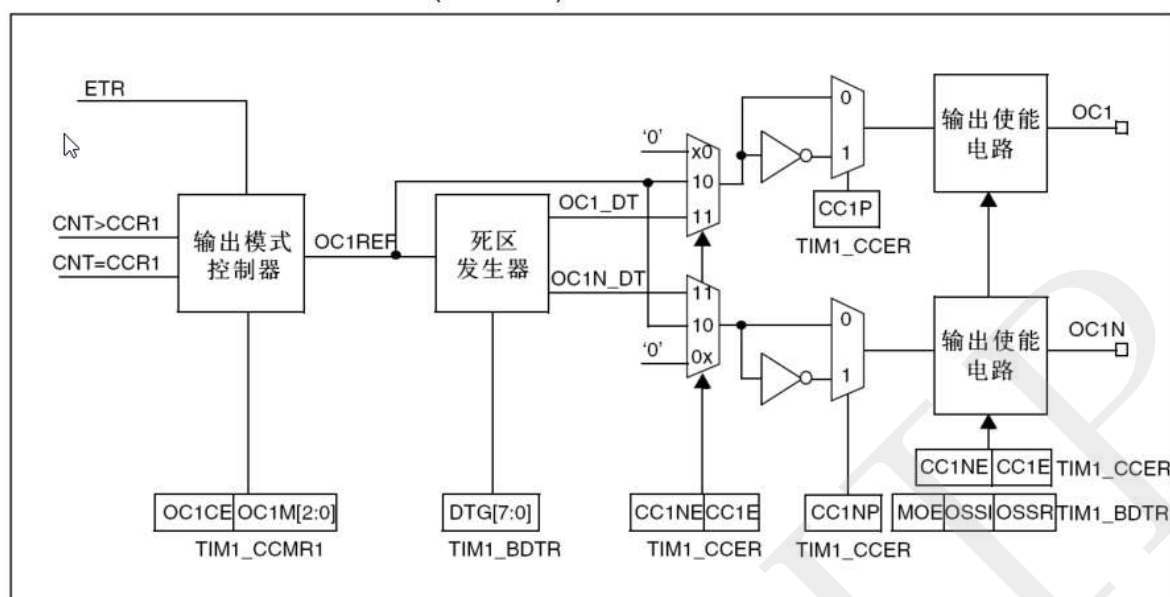
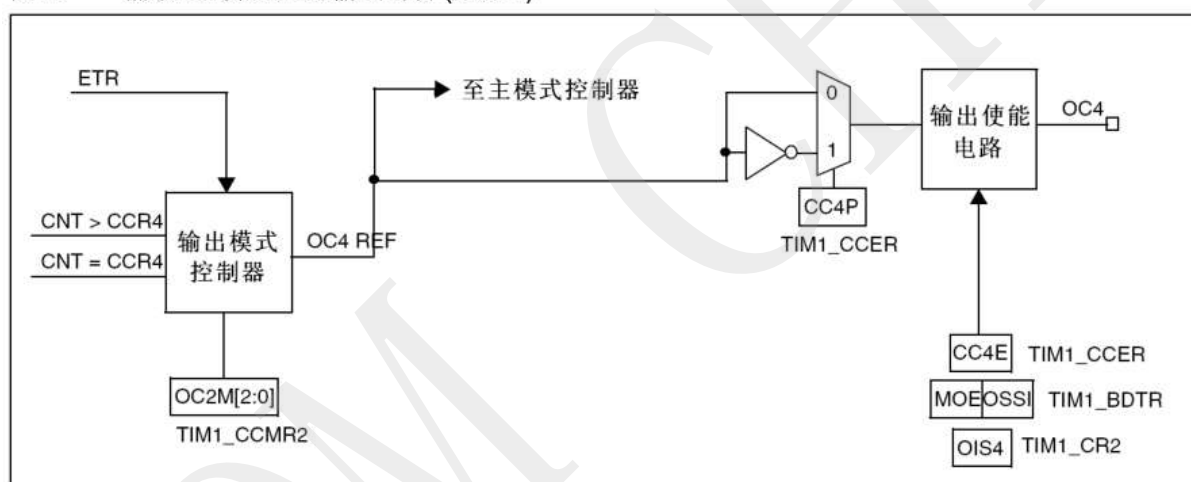


图79 捕获/比较通道的输出部分(通道4)



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

38.3.6 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器(TIMx_CCRx)中。当发生捕获事件时，相应的 CCxIF 标志(TIMx_SR 寄存器)被置 1，如果开放了中断或者 DMA 操作，则将产生中断或者 DMA

请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIMx_SR 寄存器)被置 1。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

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

- 选择有效输入端：TIMx_CCR1 必须连接到 TI1 输入，所以写入 TIMx_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为'00'，通道被配置为输入，并且 TIMx_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 Tl_x 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以(以 f_{DTS} 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0(上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIMx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

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

注：设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

38.3.7 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

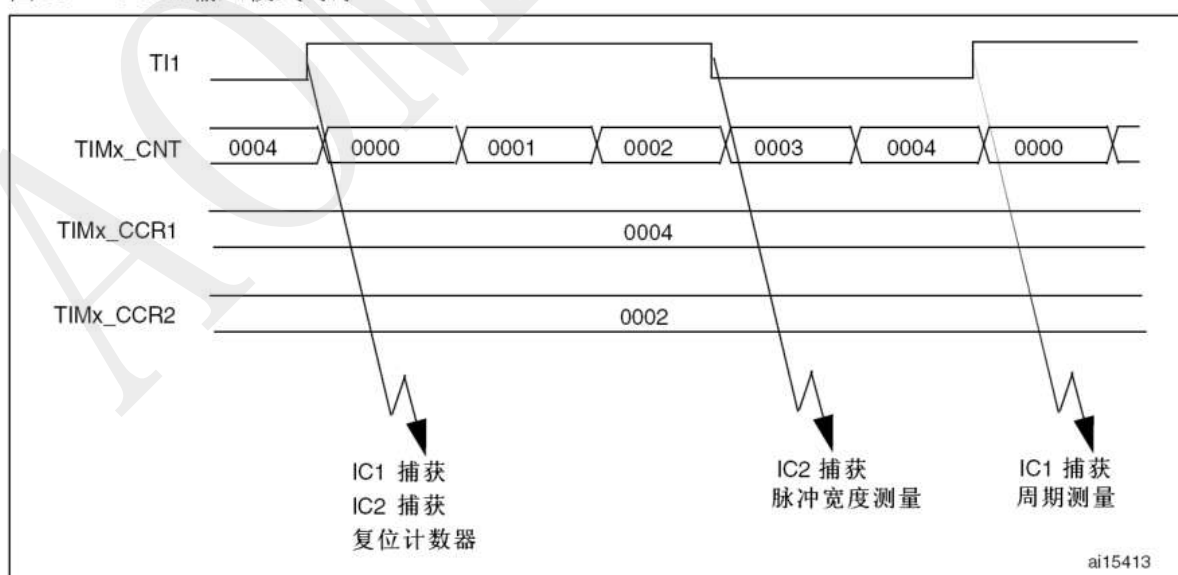
- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的长度(TIMx_CCR1 寄存器)和占空比(TIMx_CCR2 寄存器)，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 选择 TI1FP1 的有效极性(用来捕获数据到 TIMx_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到 TIMx_CCR2)：置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器,所以 PWM 输入模式只能使用 TIMx_CH1/TIMx_CH2 信号。

图80 PWM输入模式时序



38.3.8 强置输出模式

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

置 TIMx_CCMRx 寄存器中相应的 OCxM=101, 即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效), 同时 OCx 得到 CCxP 极性相反的信号。

例如: CCxP=0(OCx 高电平有效), 则 OCx 被强置为高电平。

置 TIMx_CCMRx 寄存器中的 OCxM=100, 可强置 OCxREF 信号为低。

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

38.3.9 输出比较模式

此项功能是用来控制一个输出波形, 或者指示一段给定的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时, 输出比较功能做如下操作:

- 将输出比较模式(TIMx_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时, 输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMx_DIER 寄存器中的 CCxIE 位), 则产生一个中断。
- 若设置了相应的使能位(TIMx_DIER 寄存器中的 CCxDE 位, TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能), 则产生一个 DMA 请求。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。

在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也

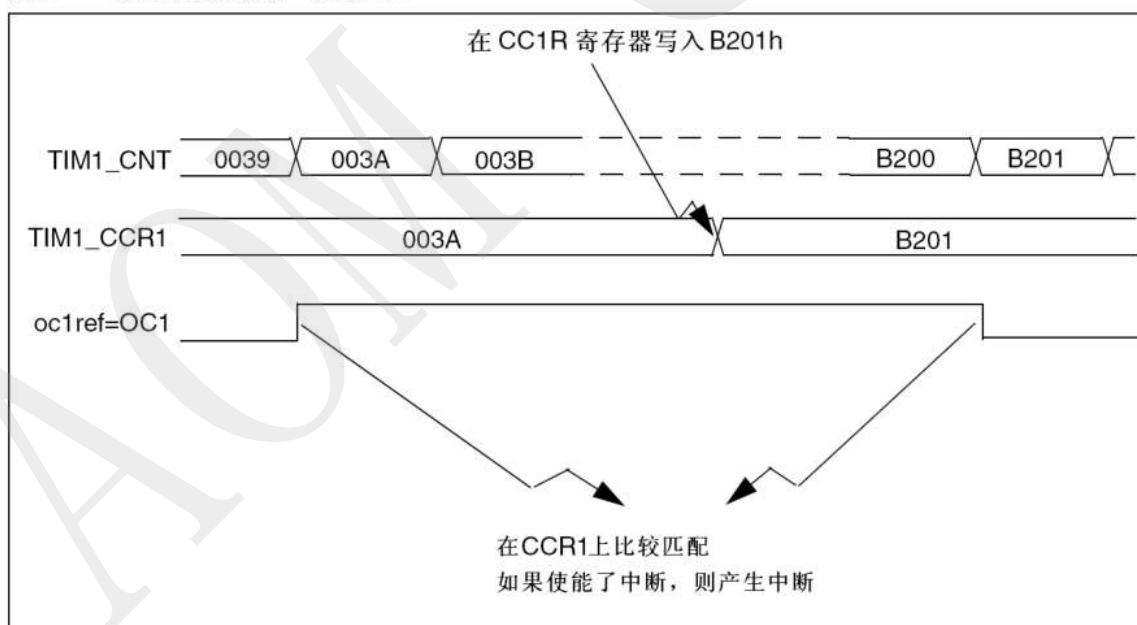
能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
 - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
 - 置 OCxPE = 0 禁用预装载寄存器
 - 置 CCxP = 0 选择极性为高电平有效
 - 置 CCxE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OCxPE=0，否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图81 输出比较模式，翻转OC1



38.3.10 PWM 模式

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

在 TIMx_CCMRx 寄存器中的 OCxM 位写入 '110' (PWM 模式 1) 或 '111' (PWM 模式 2), 能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器, 最后还要设置 TIMx_CR1 寄存器的 ARPE 位, (在向上计数或中心对称模式中) 使能自动重载的预装载寄存器。

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

OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置, 它可以设置为高电平有效或低电平有效。OCx 的输出使能通过 (TIMx_CCER 和 TIMx_BDTR 寄存器中) CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下, TIMx_CNT 和 TIMx_CCRx 始终在进行比较, (依据计数器的计数方向) 以确定是否符合 $TIMx\_CCRx \leq TIMx\_CNT$ 或者 $TIMx\_CNT \leq TIMx\_CCRx$ 。

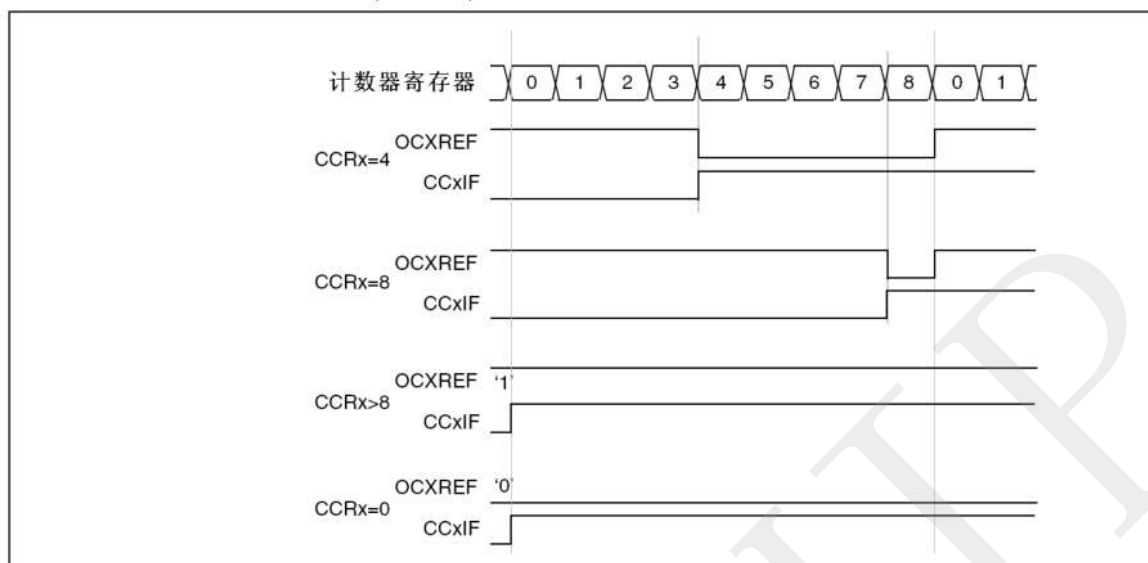
根据 TIMx_CR1 寄存器中 CMS 位的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

38.3.10.1 PWM 边沿对齐模式

向上计数配置

35.3.2 节, 当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看下面是一个 PWM 模式 1 的例子。当 $TIMx\_CNT < TIMx\_CCRx$ 时, PWM 参考信号 OCxREF 为高, 否则为低。如果 TIMx_CCRx 中的比较值大于自动重载值 (TIMx_ARR), 则 OCxREF 保持为 1。如果比较值为 0, 则 OCxREF 保持为 0。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

图82 边沿对齐的PWM波形(ARR=8)



向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。参看 35.3.2 节。

在 PWM 模式 1, 当 TIMx_CNT>TIMx_CCRx 时参考信号 OCxREF 为低, 否则为高。

如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值, 则 OCxREF 保持为 1。该模式下不能产生 0% 的 PWM 波形。

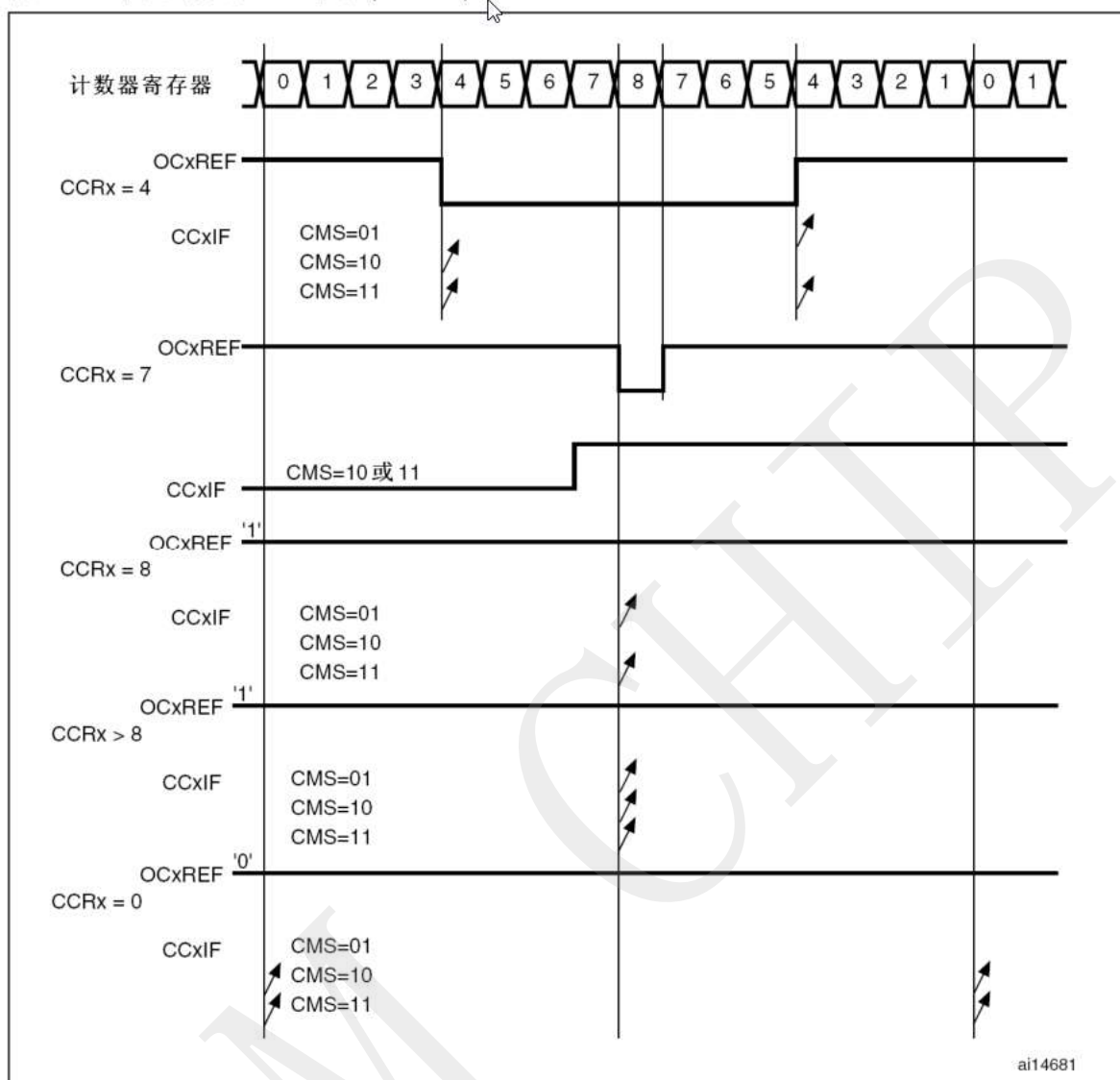
38.3.10.2 PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为'00'时为中央对齐模式(所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置, 比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位(DIR)由硬件更新, 不要用软件修改它。参看 13.3.2 节的中央对齐模式。

下图给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR=8
- PWM 模式 1
- TIMx_CR1 寄存器的 CMS=01, 在中央对齐模式 1 下, 当计数器向下计数时设置比较标志。

图83 中央对齐的PWM波形(APR=8)



使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIM_x_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
 - 如果写入计数器的值大于自动重加载的值(TIM_x_CNT>TIM_x_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
 - 如果将 0 或者 TIM_x_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设

置 TIMx_EGR 位中的 UG 位), 并且不要在计数进行过程中修改计数器的值。

38.3.11 PWM 互补输出和死区插入

高级控制定时器能够输出互补信号, 并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区, 用户应该根据连接的输出器件和它们的特性(电平转换的延时、电源开关的延时等)来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位, 可以为每一个输出独立地选择极性(主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制: TIMx_CCER 寄存器的 CCxE 和 CCxNE 位, TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位, 详见表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是, 在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区, 如果存在刹车电路, 则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效:

- OCx 输出信号与参考信号相同, 只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反, 只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度(OCx 或者 OCxN), 则不会产生相应的脉冲。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

图84 带死区插入的互补输出

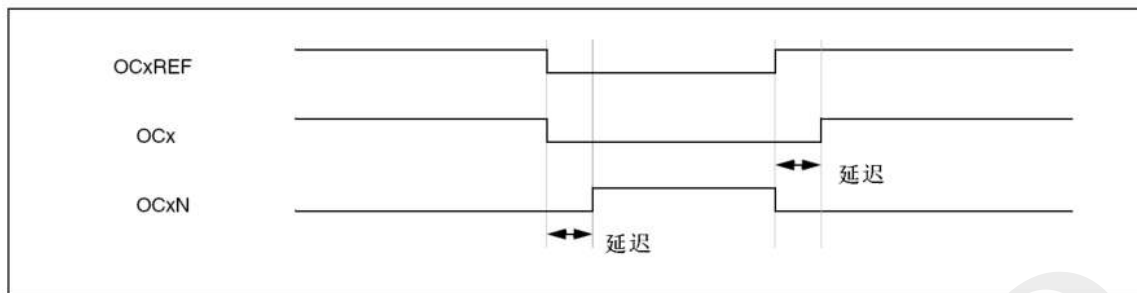


图85 死区波形延迟大于负脉冲

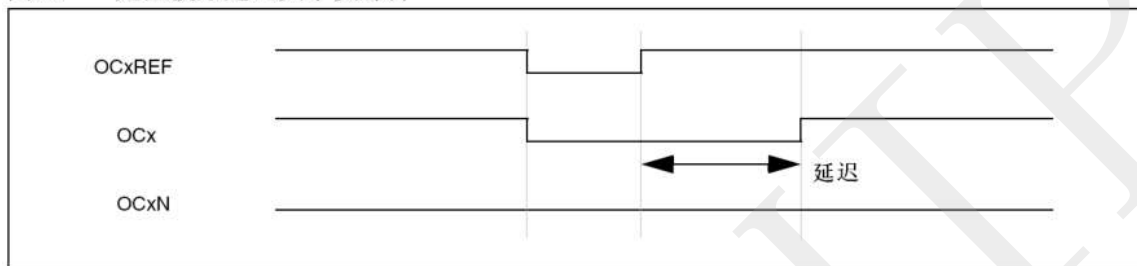
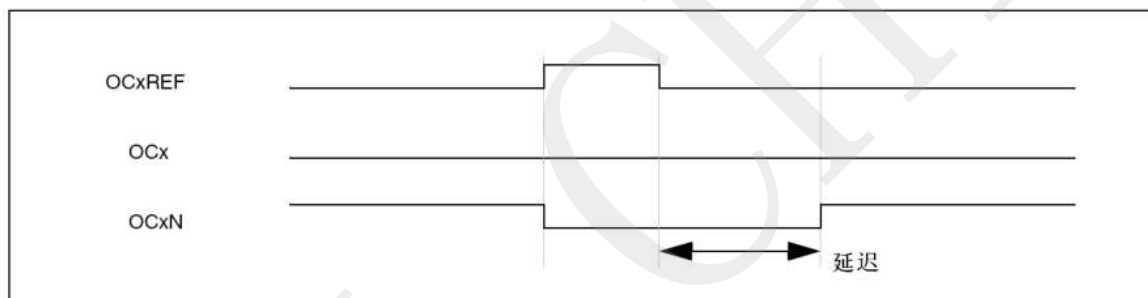


图86 死区波形延迟大于正脉冲



每一个通道的死区延时都是相同的,是由 TIMx_BDTR 寄存器中的 DTG 位编程配置。
详见 35.4.18 节 TIM1 和 TIM8 刹车和死区寄存器(TIMx_BDTR)中的延时计算。

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下(强置、输出比较或 PWM),通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位,OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时,在某个输出上送出一个特殊的波形(例如 PWM 或者静态有效电平)。另一个作用是,让两个输出同时处于无效电平,或处于有效电平和带死区的互补输出。

注:当只使能 OCxN(CCxE=0, CCxNE=1)时,它不会反相,当 OCxREF 有效时立即变高。例如,如果 CCxNP=0,则 OCxN=OCxREF。另一方面,当 OCx 和 OCxN 都被使能时(CCxE=CCxNE=1),

当 OCxREF 为高时 OCx 有效;而 OCxN 相反,当 OCxREF 低时 OCxN 变为有效。

38.3.12 使用刹车功能

设当使用刹车功能时，依据相应的控制位（TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OISx 和 OISxN 位），输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。

详见表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。

刹车源既可以是刹车输入引脚又可以是一个时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统产生。

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

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

当发生刹车时（在刹车输入端出现选定的电平），有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
 - 输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
 - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些(大

约 2 个 `ck_tim` 的时钟周期)。

— 如果 `OSSI=0`，定时器释放使能输出，否则保持使能输出；或一旦 `CCxE` 与 `CCxNE` 之一变高时，使能输出变为高。

- 如果设置了 `TIMx_DIER` 寄存器中的 `BIE` 位，当刹车状态标志(`TIMx_SR` 寄存器中的 `BIF` 位)为 1 时，则产生一个中断。如果设置了 `TIMx_DIER` 寄存器中的 `BDE` 位，则产生一个 DMA 请求。

- 如果设置了 `TIMx_BDTR` 寄存器中的 `AOE` 位，在下一个更新事件 `UEV` 时 `MOE` 位被自动置位；

例如，这可以用来进行整形。否则，`MOE` 始终保持低直到被再次置 1；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

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

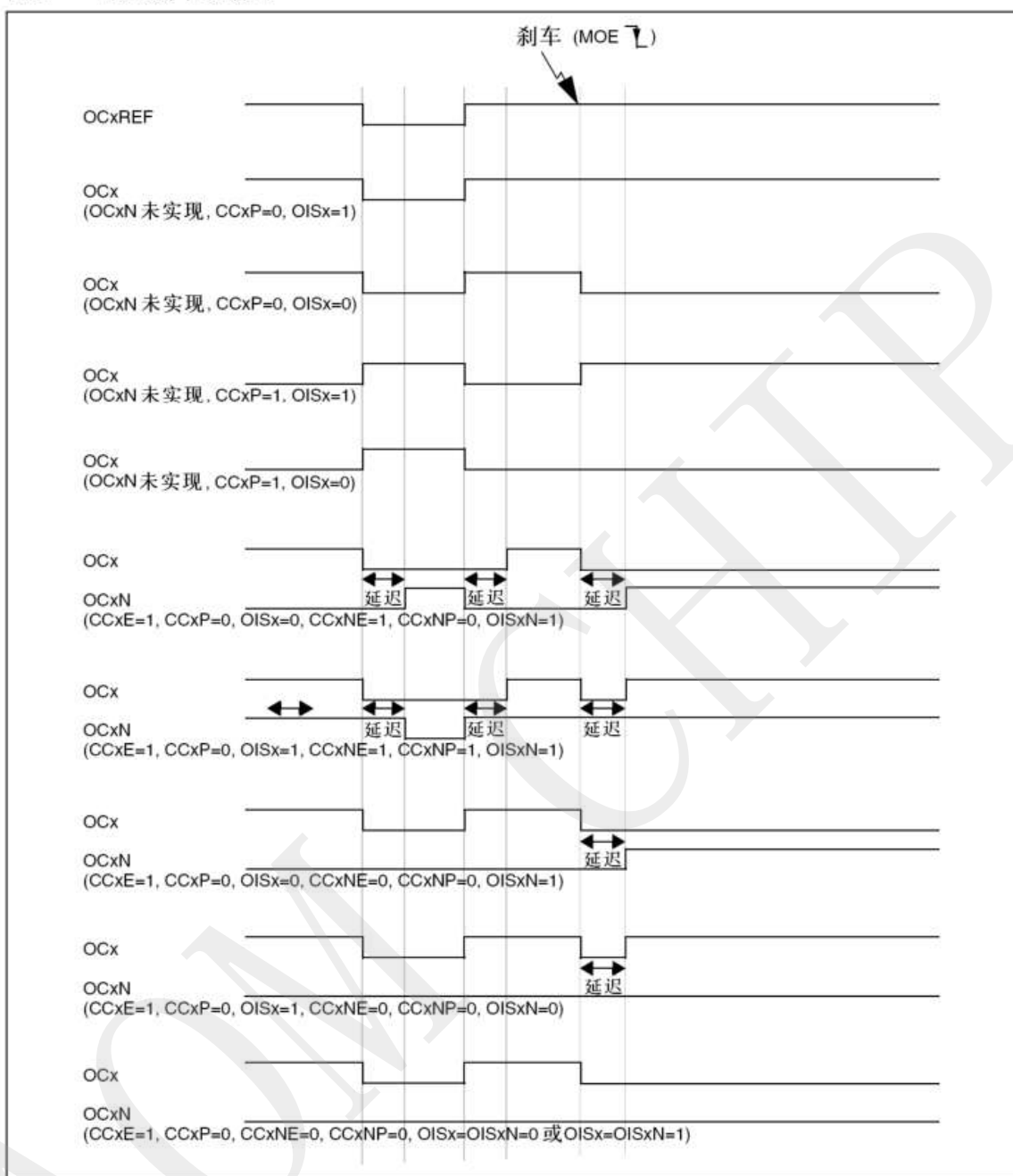
刹车由 `BRK` 输入产生，它的有效极性是可编程的，且由 `TIMx_BDTR` 寄存器中的 `BKE` 位开启。

除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，`OCx/OCxN` 极性和被禁止的状态，`OCxM` 配置，刹车使能和极性)。

用户可以通过 `TIMx_BDTR` 寄存器中的 `LOCK` 位，从三级保护中选择一种，参看 13.4.18 节刹车和死区寄存器(`TIMx_BDTR`)。在 MCU 复位后 `LOCK` 位只能被修改一次。

下图显示响应刹车的输出实例。

图87 响应刹车的输出



38.3.13 在外部事件时清除 OCxREF

对于一个给定的通道，设置 TIMx_CCMRx 寄存器中对应的 OCxCE 位为 1，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

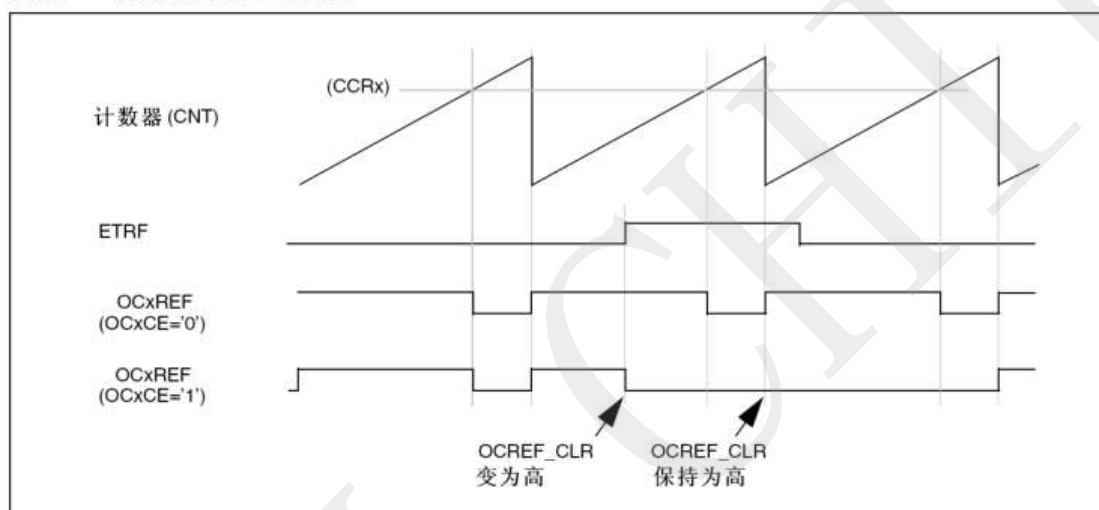
该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIMx_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE=0。
3. 外部触发极性(ETP)和外部触发滤波器(ETF)可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。
在这个例子中，定时器 TIMx 被置于 PWM 模式。

图88 清除TIMx的OCxREF



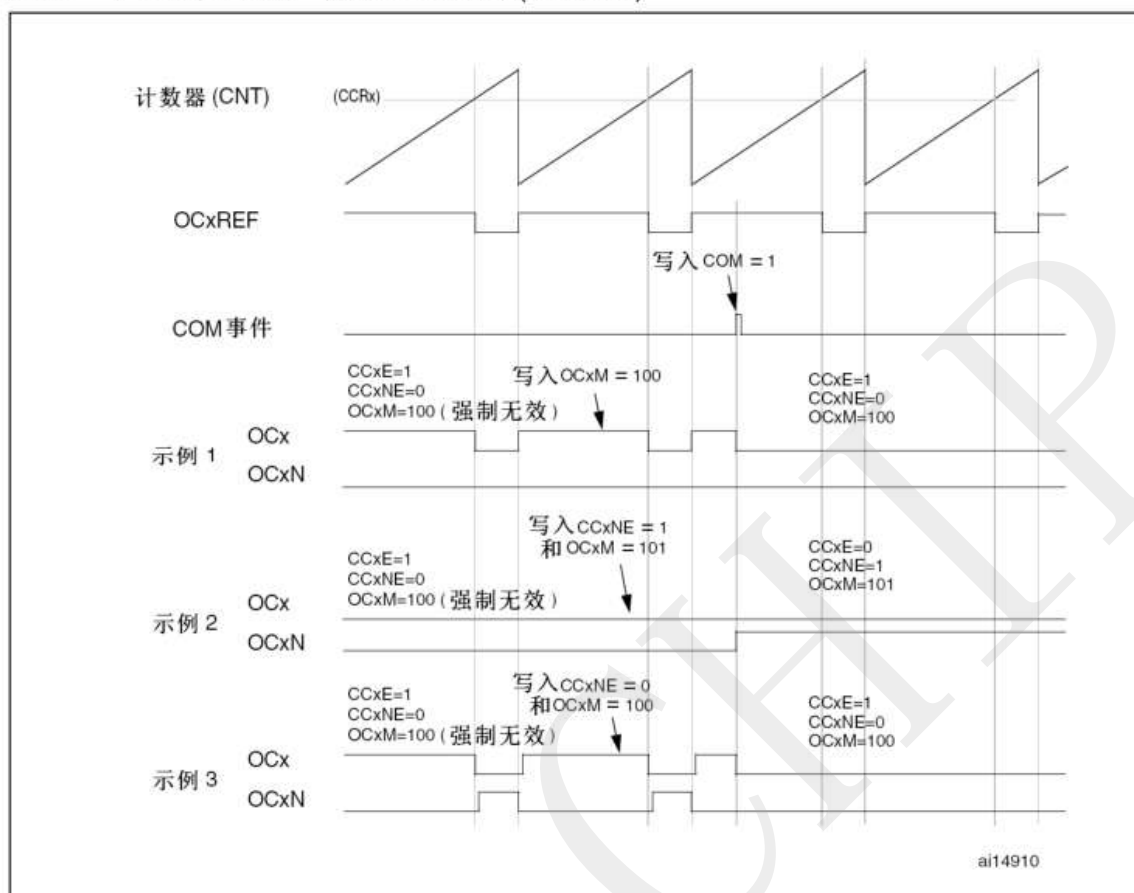
38.3.14产生六步 PWM 输出

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

当发生 COM 事件时会设置一个标志位(TIMx_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIMx_DIER 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIMx_DIER 寄存器的 COMDE 位，则产生一个 DMA 请求。

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

图89 产生六步PWM，使用COM的例子(OSSR=1)



38.3.15 单脉冲模式

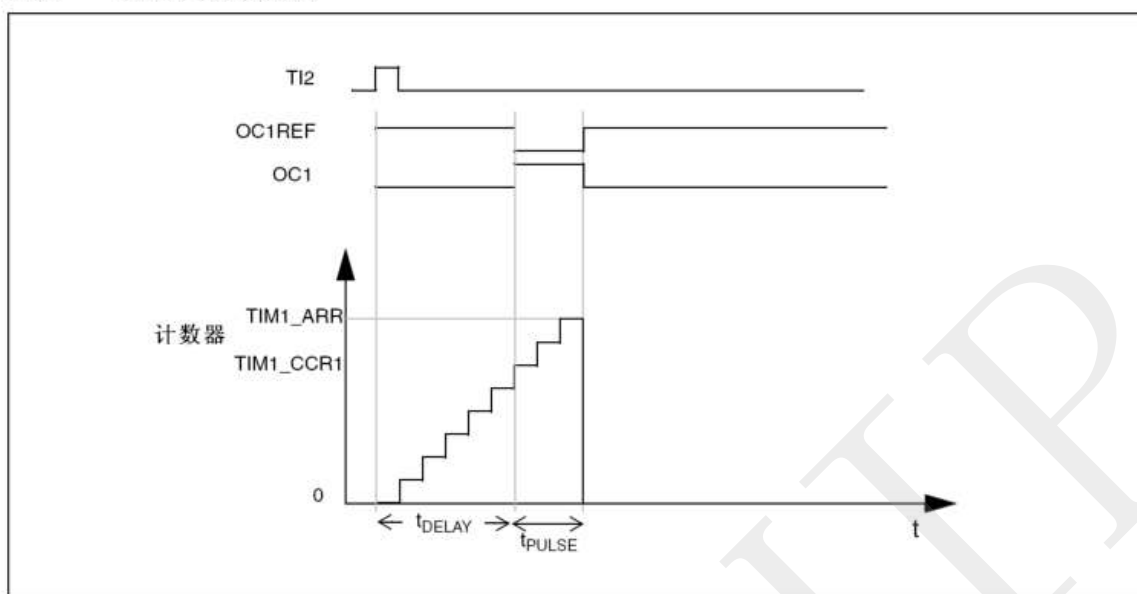
单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIM_x_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前(当定时器正在等待触发)，必须如下配置：

- 向上计数方式：计数器 $CNT < CCR_x \leq ARR$ (特别地, $0 < CCR_x$),
- 向下计数方式：计数器 $CNT > CCR_x$ 。

图90 单脉冲模式的例子



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

假定 $TI2FP2$ 作为触发 1:

- 置 $TIMx\_CCMR1$ 寄存器中的 $CC2S=01$, 把 $TI2FP2$ 映像到 $TI2$ 。
- 置 $TIMx\_CCER$ 寄存器中的 $CC2P=0$, 使 $TI2FP2$ 能够检测上升沿。
- 置 $TIMx\_SMCR$ 寄存器中的 $TS=110$, $TI2FP2$ 作为从模式控制器的触发($TRGI$)。
- 置 $TIMx\_SMCR$ 寄存器中的 $SMS=110$ (触发模式), $TI2FP2$ 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)

- t_{DELAY} 由 $TIMx\_CCR1$ 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义($TIMx\_ARR - TIMx\_CCR1$)。
- 假定当发生比较匹配时要产生从 0 到 1 的波形,当计数器达到预装载值时要产生一个从 1 到 0 的波形;首先要置 $TIMx\_CCMR1$ 寄存器的 $OC1M=111$, 进入 PWM 模式 2; 根据需要有选择地使能预装载寄存器: 置 $TIMx\_CCMR1$ 中的 $OC1PE=1$ 和 $TIMx\_CR1$ 寄存器中的 $ARPE$; 然后在 $TIMx\_CCR1$ 寄存器中填写比较值, 在 $TIMx\_ARR$ 寄存器中填写自动装载值, 设置 UG 位来产生一个更新事件, 然后等待在 $TI2$ 上的一个外部触发事件。本例中, $CC1P=0$ 。

在这个例子中, $TIMx\_CR1$ 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲, 所以必须设置 $TIMx\_CR1$ 寄存器中的 $OPM=1$, 在下一个更新

事件(当计数器从自动装载值翻转到 0)时停止计数。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 tDELAY。

如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx)直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

38.3.16 编码器接口模式

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

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

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看表 73，假定计数器已经启动(TIMx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应的设置。

不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数(根据方向,或是 0 到 ARR 计数,或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIMx_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

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

表73 计数方向与编码器信号的关系

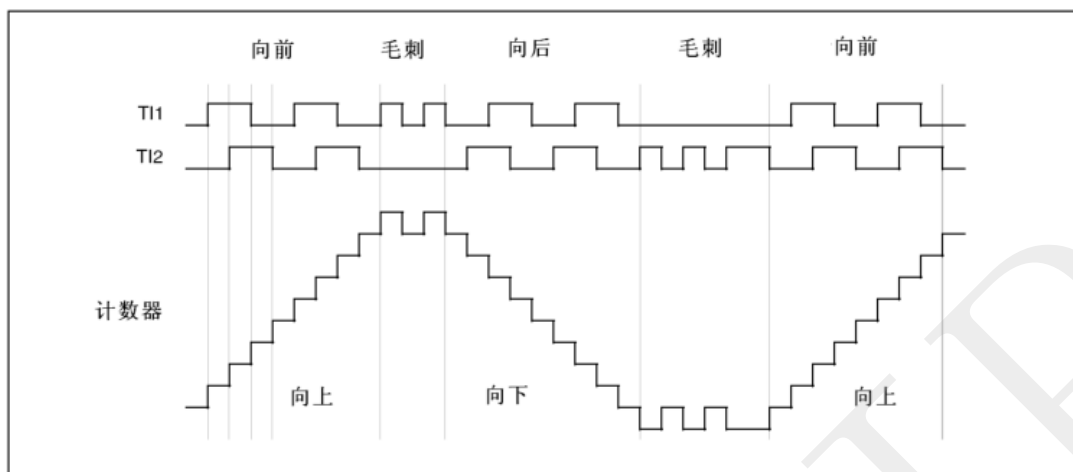
有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

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

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

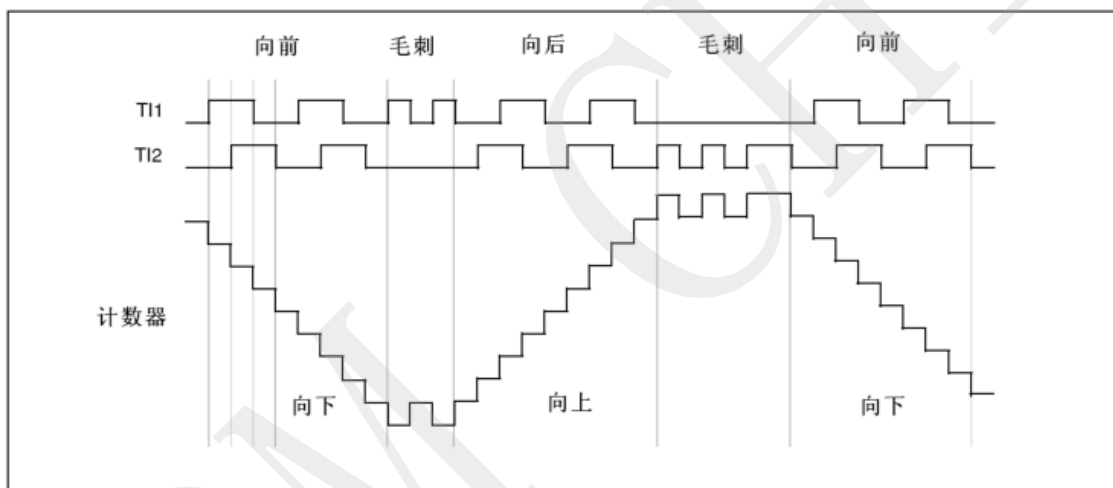
- CC1S='01' (TIMx_CCMR1 寄存器, IC1FP1 映射到 TI1)
- CC2S='01' (TIMx_CCMR2 寄存器, IC2FP2 映射到 TI2)
- CC1P=0 (TIMx_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- CC2P=0 (TIMx_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)
- SMS='011' (TIMx_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效).
- CEN=1 (TIMx_CR1 寄存器, 计数器使能)

图91 编码器模式下的计数器操作实例



下图为当 IC1FP1 极性反相时计数器的操作实例(CC1P=1, 其他配置与上例相同)

图92 IC1FP1反相的编码器接口模式实例



当定时器配置成编码器接口模式时, 提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器, 可以测量两个编码器事件的间隔, 获得动态的信息(速度, 加速度, 减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔, 可以按照固定的时间读出计数器。如果可能的话, 你可以把计数器的值锁存到第三个输入捕获寄存器(捕获信号必须是周期的并且可以由另一个定时器产生); 也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

38.3.17 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位, 允许通道 1 的输入滤波器连接到一个异或门的输出端, 异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。下节 13.3.18 给出了此特性用于连接霍尔传感器的例子。

38.3.18 与霍尔传感器的接口

使用高级控制定时器(TIM1 或 TIM8)产生 PWM 信号驱动马达时，可以用另一个通用 TIMx(TIM2、TIM3、TIM4 或 TIM5)定时器作为“接口定时器”来连接霍尔传感器，见图 93，3 个定时器输入脚(CC1、CC2、CC3)通过一个异或门连接到 TI1 输入通道(通过设置 TIMx_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置于复位模式，从输入是 TI1F_ED。每当 3 个输入之一变化时，计数器从新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC(见图 76)。捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级定时器 TIM1 或 TIM8 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。

因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1 或 TIM8。

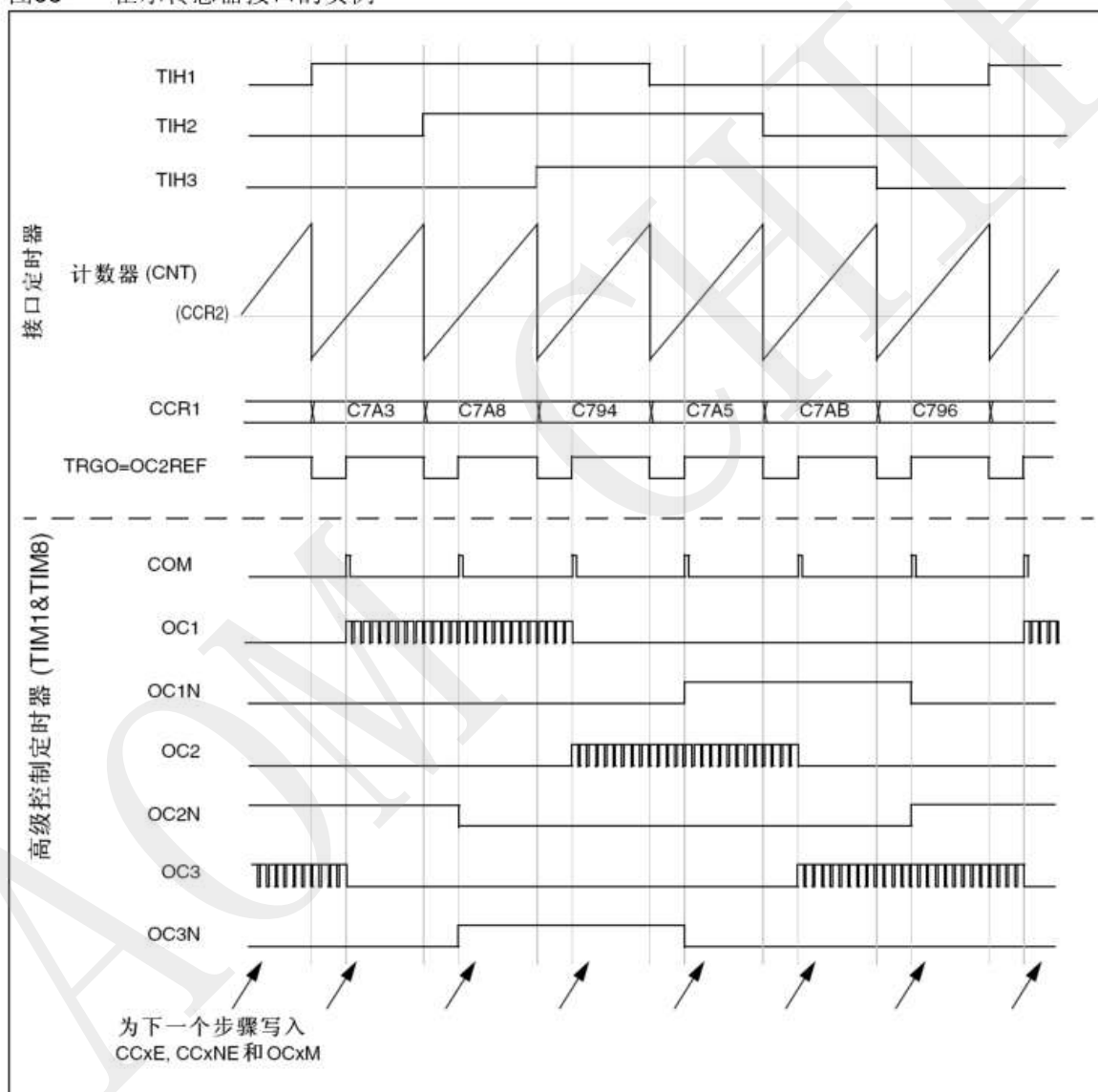
举例：霍尔输入连接到 TIMx 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIMx 的 PWM 配置。

- 置 TIMx_CR2 寄存器的 TI1S 位为 1，配置三个定时器输入逻辑或到 TI1 输入。
- 时基编程：置 TIMx_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式(选中 TRC)：置 TIMx_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIMx_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIMx_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIMx_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件(TIMx_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一步的 PWM 控制位(CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个实例

图93 霍尔传感器接口的实例



38.3.19TIMx 定时器和外部触发的同步

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

从模式：复位模式

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

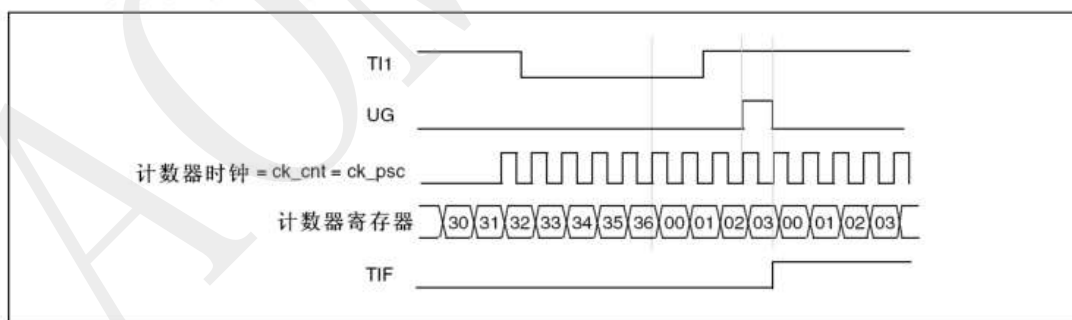
在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIMx_SR 寄存器中的 TIF 位)被设置，根据 TIMx_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。

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

图94 复位模式下的控制电路



从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，

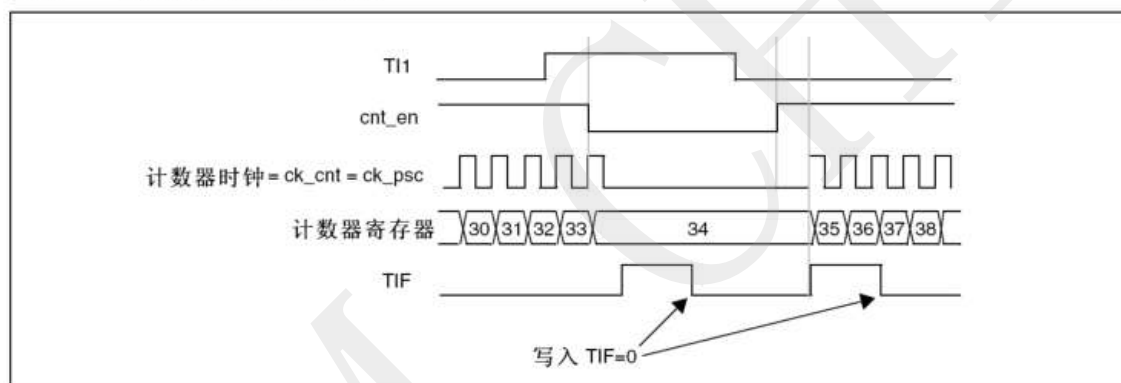
所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。

- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标志。

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

图95 门控模式下的控制电路



从模式：触发模式

输入端上选中的事件使能计数器。

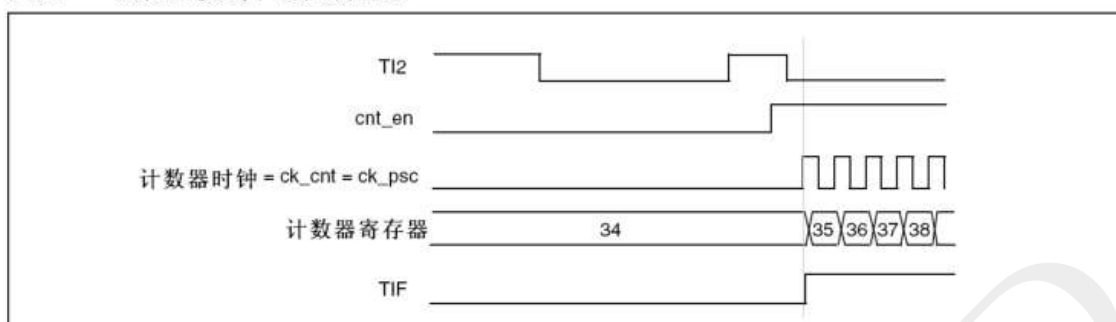
在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

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

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

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

图96 触发器模式下的控制电路



从模式：外部时钟模式 从模式：外部时钟模式 2 + 触发模式

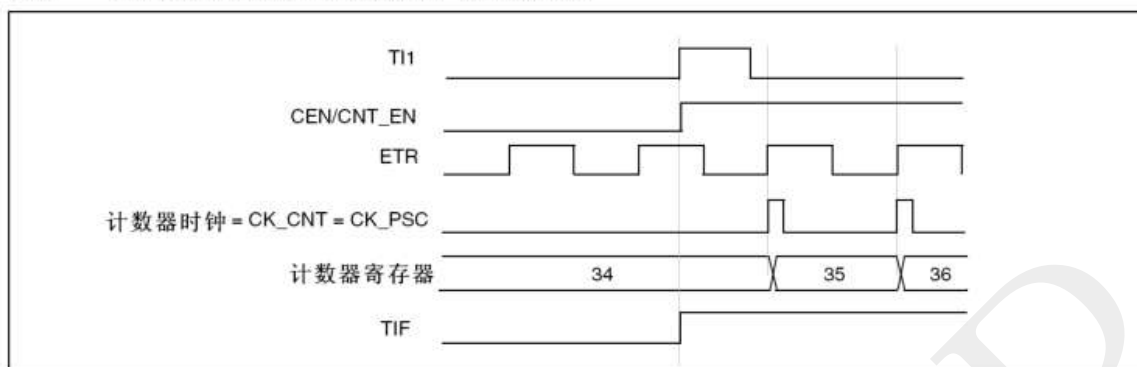
外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次：

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

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

图97 外部时钟模式2+触发模式下的控制电路



38.3.20 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。

38.3.21 调试模式

当微控制器进入调试模式时 (Cortex-M0+ 核心停止)，根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。详见相关章节。

38.4 寄存器

STIM 基地址: 0x40001C00

偏移量	寄存器名称	访问	寄存器描述
0x00	STIM_CR1	RW	控制寄存器1
0x04	STIM_CR2	RW	控制寄存器2
0x08	STIM_SMCRR	RW	从模式控制寄存器
0x0C	STIM_DIER	RW	DMA/中断使能寄存器
0x10	STIM_SR	RW	状态寄存器
0x14	STIM_EGR	RW	事件产生寄存器
0x18	STIM_CCMR1	RW	捕获比较模式寄存器1
0x1C	STIM_CCMR2	RW	捕获比较模式寄存器2
0x20	STIM_CCER	RW	捕获比较使能寄存器
0x24	STIM_CNT	RW	计数器
0x28	STIM_PSC	RW	预分配寄存器
0x2C	STIM_ARR	RW	重载值寄存器
0x30	STIM_RCR	RW	重复计数寄存器
0x34	STIM_CCR1	RW	比较捕获寄存器1
0x38	STIM_CCR2	RW	比较捕获寄存器2
0x3C	STIM_CCR3	RW	比较捕获寄存器3
0x40	STIM_CCR4	RW	比较捕获寄存器4
0x44	STIM_BDTR	RW	刹车死区控制寄存器
0x48	NC		
0x4C	NC		
0x50	STIM_OR1	RW	Option寄存器
0x54	STIM_CCMR3	RW	捕获比较模式寄存器3
0x58	STIM_CCR5	RW	比较寄存器5
0x5C	STIM_CCR6	RW	比较寄存器6
0x60	STIM_AF1	RW	辅助功能寄存器1
0x64	STIM_AF2	RW	辅助功能寄存器2
0x68	STIM_TISEL	RW	T1选择寄存器

表 38-1 寄存器列表

38.4.1 控制寄存器 1(TIMx_CR1)

地址偏移量: 0x00

复位值:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD	PE	CMS	DIR	OPM	URS	DIS	CEN		
						RW	RW	RW	RW	RW	RW	RW	RW		

位	标记	功能描述
31:10	RSV	保留位
9:8	CKD	时钟分频因子 (Clock division) 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例。 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 \times t_{CK\_INT}$ 10: $t_{DTS} = 4 \times t_{CK\_INT}$ 11: 保留, 不要使用这个配置
7	PE	自动重载预装载允许位 0: TIMx_ARR 寄存器没有缓冲 1: TIMx_ARR 寄存器被装入缓冲器
6:5	CMS	选择中央对齐模式 00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向下计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被设置。 注: 在计数器开启时(CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	方向 0: 计数器向上计数; 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读
3	OPM	单脉冲模式 0: 在发生更新事件时, 计数器不停止; 1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。

2	URS	更新请求源 软件通过该位选择 UEV 事件的源 0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	DIS	禁止更新 软件通过该位允许/禁止 UEV 事件的产生。 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(译注: 更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、CCR_x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。
0	CEN	使能计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

38.4.2 控制寄存器 2(TIMx_CR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
RSV	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1								
	RW	RW	RW	RW	RW	RW	RW								
7	6	5	4	3	2	1	0								
TI1S	MMS[2:0]			CCDS	CCUS	RSV	CCPC								
RW	RW			RW	RW		RW								

位	标记	功能描述
31:15	RSV	保留位
14	OIS4	输出空闲状态 4(OC4 输出)。参见 OIS1 位。
13	OIS3N	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。
12	OIS3	输出空闲状态 3(OC3 输出)。参见 OIS1 位。
11	OIS2N	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。
10	OIS2	输出空闲状态 2(OC2 输出)。参见 OIS1 位。
9	OIS1N	输出空闲状态 1(OC1N 输出) (Output Idle state 1) 0: 当 MOE=0 时, 死区后 OC1N=0; 1: 当 MOE=0 时, 死区后 OC1N=1。 注: 已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
8	OIS1	输出空闲状态 1(OC1 输出) (Output Idle state 1) 0: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=0; 1: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=1。 注: 已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
7	TI1S	TI1 选择 (TI1 selection) 0: TIMx_CH1 引脚连到 TI1 输入; 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。
6:4	MMS[2:0]	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: 000: 复位 - TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能 - 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)。 010: 更新 - 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从

		定时器的预分频器。 011: 比较脉冲 - 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。 100: 比较 - OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 - OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较 - OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较 - OC4REF 信号被用于作为触发输出(TRGO)。。
3	CCDS	捕获/比较的 DMA 选择 (Capture/compare DMA selection) 0: 当发生 CCx 事件时, 送出 CCx 的 DMA 请求; 1: 当发生更新事件时, 送出 CCx 的 DMA 请求。
2	CCUS	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1), 只能通过设置 COM 位更新它们; 1: 如果捕获/比较控制位是预装载的(CCPC=1), 可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注: 该位只对具有互补输出的通道起作用。
1	RSV	位 1 保留, 始终读为 0。
0	CCPC	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位不是预装载的; 1: CCxE, CCxNE 和 OCxM 位是预装载的; 设置该位后, 它们只在设置了 COM 位后被更新。

38.4.3 从模式控制寄存器 (TIMx_SMCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			RSV	SMS[2:0]		
RW	RW	RW		RW				RW	RW			RW	RW		

位	标记	功能描述
31:16	RSV	保留位
15	ETP	外部触发极性 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相, 高电平或上升沿有效; 1: ETR 被反相, 低电平或下降沿有效。
14	ECE	外部时钟使能位 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2; 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。 注 1: 设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF (SMS=111 和 TS=111) 具有相同功效。 注 2: 下述从模式可以与外部时钟模式 2 同时使用: 复位模式, 门控模式和触发模式; 但是, 这时 TRGI 不能连到 ETRF (TS 位不能是 '111')。 注 3: 外部时钟模式 1 和外部时钟模式 2 同时被使能时, 外部时钟的输入是 ETRF。
13:12	ETPS[1:0]	外部触发预分频 外部触发信号 ETRP 的频率必须最多是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。 00: 关闭预分频; 01: ETRP 频率除以 2; 10: ETRP 频率除以 4; 11: ETRP 频率除以 8。
11:8	ETF[3:0]	外部触发滤波 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产生一个输出的跳变。 0000: 无滤波器, 以 f_{DTS} 采样 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8, N = 6$ 0001: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N = 2$ 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8, N = 8$ 0010: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N = 4$ 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16, N = 5$ 0011: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N = 8$

		1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16, N = 6$ 0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2, N = 6$ 1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16, N = 8$ 0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2, N = 8$ 1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N = 5$ 0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4, N = 6$ 1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N = 6$ 0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4, N = 8$ 1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N = 8$
7	MSM	主/从模式 0: 无作用; 1: 触发输入 (TRGI) 上的事件被延迟了, 以允许在当前定时器 (通过 TRG0) 与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
6:4	TS[2:0]	触发选择 这 3 位选择用于同步计数器的触发输入。 000: 内部触发 0 (ITR0) 100: TI1 的边沿检测器 (TI1F_ED) 001: 内部触发 1 (ITR1) 101: 滤波后的定时器输入 1 (TI1FP1) 010: 内部触发 2 (ITR2) 110: 滤波后的定时器输入 2 (TI2FP2) 011: 内部触发 3 (ITR3) 111: 外部触发输入 (ETRF) 注: 这些位只能在未用到 (如 SMS=000) 时被改变, 以避免在改变时产生错误的边沿检测。
3	RSV	保留, 始终读为 0。
2:0	SMS[2:0]	从模式选择 当选择了外部信号, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明) 000: 关闭从模式 - 如果 CEN=1, 则预分频器直接由内部时钟驱动。 001: 编码器模式 1 - 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数。 010: 编码器模式 2 - 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。 011: 编码器模式 3 - 根据另一个信号的输入电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 100: 复位模式 - 选中的触发输入 (TRGI) 的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。 101: 门控模式 - 当触发输入 (TRGI) 为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止 (但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 - 计数器在触发输入 TRGI 的上升沿启动 (但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式 1 - 选中的触发输入 (TRGI) 的上升沿驱动计数器。 注: 如果 TI1F_EN 被选为触发输入 (TS=100) 时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。

38.4.4DMA/中断使能寄存器(TIMx_DIER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
RSV		TDE		COMDE		CC4DE		CC3DE		CC2DE		CC1DE		UDE	
		RW		RW		RW		RW		RW		RW		RW	
7		6		5		4		3		2		1		0	
BIE		TIE		COMIE		CC4IE		CC3IE		CC2IE		CC1IE		UIE	
RW		RW		RW		RW		RW		RW		RW		RW	

位	标记	功能描述
31:15	RSV	保留位
14	TDE	允许触发 DMA 请求 0: 禁止触发 DMA 请求; 1: 允许触发 DMA 请求。
13	COMDE	允许 COM 的 DMA 请求 0: 禁止 COM 的 DMA 请求; 1: 允许 COM 的 DMA 请求。 1: 允许捕获/比较 4 的 DMA 请求。
12	CC4DE	允许捕获/比较 4 的 DMA 请求 0: 禁止捕获/比较 4 的 DMA 请求;
11	CC3DE	允许捕获/比较 3 的 DMA 请求 0: 禁止捕获/比较 3 的 DMA 请求; 1: 允许捕获/比较 3 的 DMA 请求。
10	CC2DE	允许捕获/比较 2 的 DMA 请求 0: 禁止捕获/比较 2 的 DMA 请求; 1: 允许捕获/比较 2 的 DMA 请求。
9	CC1DE	允许捕获/比较 1 的 DMA 请求 0: 禁止捕获/比较 1 的 DMA 请求; 1: 允许捕获/比较 1 的 DMA 请求。
8	UDE	允许更新的 DMA 请求 0: 禁止更新的 DMA 请求; 1: 允许更新的 DMA 请求。
7	BIE	允许刹车中断 0: 禁止刹车中断; 1: 允许刹车中断。
6	TIE	触发中断使能

		0: 禁止触发中断; 1: 使能触发中断。
5	COMIE	允许 COM 中断 0: 禁止 COM 中断; 1: 允许 COM 中断。
4	CC4IE	允许捕获/比较 4 中断 0: 禁止捕获/比较 4 中断; 1: 允许捕获/比较 4 中断。
3	CC3IE	允许捕获/比较 3 中断 0: 禁止捕获/比较 3 中断; 1: 允许捕获/比较 3 中断。
2	CC2IE	允许捕获/比较 2 中断 0: 禁止捕获/比较 2 中断; 1: 允许捕获/比较 2 中断。
1	CC1IE	允许捕获/比较 1 中断 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断
0	UIE	允许更新中断 0: 禁止更新中断; 1: 允许更新中断。

38.4.5 状态寄存器(TIMx_SR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														CC6IF	CC5IF
														RW	RW
15		14		13		12		11		10		9		8	
Reserved						CC4OF		CC3OF		CC2OF		CC1OF		B2IF	
						RW		RW		RW		RW		RW	
7		6		5		4		3		2		1		0	
BIF		TIF		COMIF		CC4IF		CC3IF		CC2IF		CC1IF		UIF	
RW		RW		RW		RW		RW		RW		RW		RW	

位	标记	功能描述
31:18	RSV	保留位
17	CC6IF	比较 6 中断标记
16	CC5IF	比较 5 中断标记
15:13	RSV	保留位
12	CC4OF	捕获/比较 4 重复捕获标记
11	CC3OF	捕获/比较 3 重复捕获标记
10	CC2OF	捕获/比较 2 重复捕获标记
9	CC1OF	捕获/比较 1 重复捕获标记 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置 1。写 0 可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到 TIMx_CCR1 寄存器时, CC1IF 的状态已经为 1
8	B2IF	刹车 2 中断标记 一旦刹车输入有效, 由硬件对该位置 1。如果刹车输入无效, 则该位可由软件清 0。 0: 无刹车事件产生; 1: 刹车输入上检测到有效电平。
7	BIF	刹车中断标记 一旦刹车输入有效, 由硬件对该位置 1。如果刹车输入无效, 则该位可由软件清 0。 0: 无刹车事件产生; 1: 刹车输入上检测到有效电平。
6	TIF	触发器中断标记 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时, 在 TRGI 输入端检测到有效边沿, 或门控模式下的任一边沿)时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生; 1: 触发中断等待响应。
5	COMIF	COM 中断标记

		一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置 1。它由软件清 0。 0: 无 COM 事件产生; 1: COM 中断等待响应。
4	CC4IF	捕获/比较 4 中断标记
3	CC3IF	捕获/比较 3 中断标记
2	CC2IF	捕获/比较 2 中断标记
1	CC1IF	捕获/比较1中断标记 如果通道 如果通道 CC1 配置为输出模式: 当计数器值与比较值匹配时该位由硬件置1, 但在中心对称模式下除外(参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清0。 0: 无匹配发生; 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时, 在向上或向上/下计数模式时计数器溢出, 或向下计数模式时的计数器下溢条件下, CC1IF 位变高。 如果通道 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置1, 它由软件清0或通过读 TIMx_CCR1 清0。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIMx_CCR1 (在 IC1 上检测到与所选极性相同的边沿)
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置 1: – 若 TIMx_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 – 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当计数器 CNT 被触发事件重新初始化时。(参考 38.4.3 从模式控制寄存器 (TIMx_SMCR))。

38.4.6 事件产生寄存器(TIMx_EGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:9	RSV	保留位
8	B2G	Bk2 产生刹车事件 该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0。 0：无动作； 1：产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
7	BG	BK 产生刹车事件 该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0。 0：无动作； 1：产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
6	TG	产生触发事件 该位由软件置 1，用于产生一个触发事件，由硬件自动清 0。 0：无动作； 1：TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
5	COMG	捕获/比较事件，产生控制更新 该位由软件置 1，由硬件自动清 0。 0：无动作； 1：当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。 注：该位只对拥有互补输出的通道有效。
4	CC4G	产生捕获/比较 4 事件
3	CC3G	产生捕获/比较 3 事件
2	CC2G	产生捕获/比较 2 事件
1	CC1G	产生捕获/比较 1 事件 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0：无动作； 1：在通道 CC1 上产生一个捕获/比较事件： 若通道 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。

0	UG	产生更新事件 该位由软件置 1，由硬件自动清 0。 0：无动作； 1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0；若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。

38.4.7 捕获比较模式寄存器 1(TIMx_CCMR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
OC2CE		OC2M[2:0]						OC2PE		OC2FE		CC2S[1:0]			
RW		RW						RW		RW		RW			
7		6		5		4		3		2		1		0	
OC1CE		OC1M[2:0]						OC1PE		OC1FE		CC1S[1:0]			
RW		RW						RW		RW		RW			

位	标记	功能描述
31:16	RSV	保留位
15	OC2CE	输出比较 2 清 0 使能 (Output Compare 2 clear enable)
14: 12	OC2M[2:0]	输出比较 2 模式 (Output Compare 2 mode)
11	OC2PE	输出比较 2 预装载使能 (Output Compare 2 preload enable)
10	OC2FE	输出比较 2 快速使能 (Output Compare 2 fast enable)
9:8	CC2S[1:0]	捕获/比较 2 选择。(Capture/Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出； 01：CC2 通道被配置为输入，IC2 映射在 TI2 上； 10：CC2 通道被配置为输入，IC2 映射在 TI1 上； 11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7	OC1CE	输出比较 1 清 0 使能 (Output Compare 1 clear enable) 0：OC1REF 不受 ETRF 输入的影响； 1：一旦检测到 ETRF 输入高电平，清除 OC1REF=0。
6:4	OC1M[2:0]	输出比较 1 模式 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；

		001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR1) 相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1(TIMx_CCR1) 相同时, 强制 OC1REF 为低。 011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00(该通道配置成输出) 则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。
3	OC1PE	输出比较 1 预装载使能 (Output Compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能, 可随时写入 TIMx_CCR1 寄存器, 并且新写入的数值立即起作用。 1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00(该通道配置成输出) 则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMx_CR1 寄存器的 OPM=1), 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定。
2	OC1FE	输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S[1:0]	捕获/比较 1 选择。(Capture/Compare 1 selection) 位 1:0 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

位	标记	功能描述

31:16	RSV	保留位
15:12	IC2F[3:0]	输入捕获 2 滤波器 (Input capture 2 filter)
11:10	IC2PSC[1:0]	输入/捕获 2 预分频器 (Input capture 2 prescaler)
9:8	CC2S[1:0]	捕获/比较 2 选择 (Capture/Compare 2 selection) 位 9:8 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2 通道被配置为输出; 01: CC2 通道被配置为输入, IC2 映射在 TI2 上; 10: CC2 通道被配置为输入, IC2 映射在 TI1 上; 11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F[3:0]	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器, 以 f_{DTS} 采样 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8, N = 6$ 0001: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N = 2$ 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8, N = 8$ 0010: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N = 4$ 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16, N = 5$ 0011: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N = 8$ 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16, N = 6$ 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2, N = 6$ 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16, N = 8$ 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2, N = 8$ 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32, N = 5$ 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4, N = 6$ 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32, N = 6$ 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4, N = 8$ 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32, N = 8$
3:2	IC1PSC[1:0]	输入/捕获 1 预分频器 (Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1E=0(TIMx_CCER 寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每 2 个事件触发一次捕获; 10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获
1:0	CC1S[1:0]	捕获/比较 1 选择 (Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上;

		11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。

38.4.8 捕获比较模式寄存器 2(TIMx_CCMR2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
C4CE	OC4M[2:0]				OC4PE	OC4FE	CC4S[1:0]								
RW	RW				RW	RW	RW								
7	6	5	4	3	2	1	0								
OC3CE	OC3M[2:0]				OC3PE	OC3FE	CC3S[1:0]								
RW	RW				RW	RW	RW								

位	标记	功能描述
31:16	RSV	保留位
15	C4CE	输出比较 4 清 0 使能 (Output compare 4 clear enable)
14:12	OC4M[2:0]	输出比较 4 模式 (Output compare 4 mode)
11	OC4PE	输出比较 4 预装载使能 (Output compare 4 preload enable)
10	OC4FE	输出比较 4 快速使能 (Output compare 4 fast enable)
9:8	CC4S[1:0]	捕获/比较 4 选择 (Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 通道被配置为输出; 01: CC4 通道被配置为输入, IC4 映射在 TI4 上; 10: CC4 通道被配置为输入, IC4 映射在 TI3 上; 11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
7	OC3CE	输出比较 3 清 0 使能 (Output compare 3 clear enable)
6:4	OC3M[2:0]	输出比较 3 模式 (Output compare 3 mode)
3	OC3PE	输出比较 3 预装载使能 (Output compare 3 preload enable)
2	OC3FE	输出比较 3 快速使能 (Output compare 3 fast enable)
1:0	CC3S[1:0]	捕获/比较 3 选择 (Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3 通道被配置为输出; 01: CC3 通道被配置为输入, IC3 映射在 TI3 上; 10: CC3 通道被配置为输入, IC3 映射在 TI4 上; 11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由

		TIMx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC3E=0) 才是可写的。

位	标记	功能描述
31:16	RSV	保留位
15:12	IC4F[3:0]	输入捕获 4 滤波器 (Input capture 4 filter)
11: 10	IC4PSC[1:0]	输入/捕获 4 预分频器 (Input capture 4 prescaler)
9:8	CC4S[1:0]	捕获/比较 4 选择 (Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4 通道被配置为输出； 01: CC4 通道被配置为输入，IC4 映射在 TI4 上； 10: CC4 通道被配置为输入，IC4 映射在 TI3 上； 11: CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC4S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC4E=0) 才是可写的。
7:4	IC3F[3:0]	输入捕获 3 滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获 3 预分频器 (Input capture 3 prescaler)
1:0	CC3S[1:0]	捕获/比较 3 选择 (Capture/compare 3 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3 通道被配置为输出； 01: CC3 通道被配置为输入，IC3 映射在 TI3 上； 10: CC3 通道被配置为输入，IC3 映射在 TI4 上； 11: CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC3E=0) 才是可写的。

38.4.9 捕获/ 比较使能寄存器(TIMx_CCER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E								
		RW	RW	RW	RW	RW	RW								
7	6	5	4	3	2	1	0								
CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:14	RSV	保留位
13	CC4P	输入/捕获 4 输出极性
12	CC4E	输入/捕获 4 输出使能
11	CC3NP	输入/捕获 3 互补输出极性
10	CC3NE	输入/捕获 3 互补输出使能
9	CC3P	输入/捕获 3 输出极性
8	CC3E	输入/捕获 3 输出使能
7	CC2NP	输入/捕获 2 互补输出极性
6	CC2NE	输入/捕获 2 互补输出使能
5	CC2P	输入/捕获 2 输出极性
4	CC2E	输入/捕获 2 输出使能
3	CC1NP	输入/捕获 1 互补输出极性 0: OC1N 高电平有效; 1: OC1N 低电平有效。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S=00(通道配置为输出)则 该位不能被修改。
2	CC1NE	输入/捕获 1 互补输出使能 0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位 的值。 1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	输入/捕获 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:

		该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相: 捕获发生在 IC1 的上升沿; 当用作外部触发器时, IC1 不反相。 1: 反相: 捕获发生在 IC1 的下降沿; 当用作外部触发器时, IC1 反相。 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位) 设为 3 或 2, 则该位不能被修改。
0	CC1E	输入/捕获 1 输出使能 CC1 通道配置为输出: 0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启— OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入: 该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 0: 捕获禁止; 0: 捕获使能。

表75 带刹车功能的互补输出通道OCx和OCxN的控制位

控制位					输出状态 ⁽¹⁾	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止(与定时器断开) OCx=0, OCx_EN=0	输出禁止(与定时器断开) OCxN=0, OCxN_EN=0
		0	0	1	输出禁止(与定时器断开) OCx=0, OCx_EN=0	OCxREF + 极性, OCxN= OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + 极性, OCx= OCxREF xor CCxP, OCx_EN=1	输出禁止(与定时器断开) OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF反相 + 极性 + 死区, OCxN_EN=1
		1	0	0	输出禁止(与定时器断开) OCx=CCxP, OCx_EN=0	输出禁止(与定时器断开) OCxN=CCxNP, OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP, OCx_EN=1	OCxREF + 极性, OCxN= OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性, OCx= OCxREF xor CCxP, OCx_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF反相 + 极性 + 死区, OCxN_EN=1
0	X	0	0	0	输出禁止(与定时器断开)	异步地: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0; 若时钟存在: 经过一个死区时间后 OCx=OISx, OCxN=OISxN, 假设OISx与OISxN并不都对应OCx和OCxN的有效电平。
		0	0	1		
		0	1	0		
		0	1	1		
		1	0	0	关闭状态(输出使能且为无效电平)	异步地: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1; 若时钟存在: 经过一个死区时间后 OCx=OISx, OCxN=OISxN, 假设OISx与OISxN并不都对应OCx和OCxN的有效电平。
		1	0	1		
		1	1	0		
		1	1	1		

1. 如果一个通道的2个输出都没有使用(CCxE = CCxNE = 0), 那么OISx, OISxN, CCxP和CCxNP都必须清零。
引脚连接到互补的OCx和OCxN通道的外部I/O引脚的状态, 取决于OCx和OCxN通道状态和GPIO以及AFIO寄存器。

38.4.10 计数器(TIMx_CNT)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
RW															

位	标记	功能描述
31:16	RSV	保留位
15:0	CNT	计数器的值 (Counter value)

38.4.11 预分配寄存器(TIMx_PSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC [15:0]															
RW															

位	标记	功能描述
31:16	RSV	保留位
15:0	PSC	预分频器的值 (Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ 。 PSC 包含了每次当更新事件产生时, 装入当前预分频器寄存器的值; 更新事件包括计数器被 TIM_EGR 的 UG 位清 0 或被工作在复位模式的从控制器清 0。

38.4.12 重载值寄存器(TIMx_ARR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR [15:0]															
RW															

位	标记	功能描述
31:16	RSV	保留位
15:0	ARR[15:0]	自动重载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重载寄存器的值。 当自动重载的值为空时, 计数器不工作。

38.4.13 重复计数寄存器 (TIMx_RCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCR [15:0]															
RW															

位	标记	功能描述
31:16	RSV	保留位
15:0	RCR	重复计数器的值 (Repetition counter value) 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。 每次向下计数器 REP_CNT 达到 0, 会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值, 因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在 PWM 模式中, (REP+1) 对应着: - 在边沿对齐模式下, PWM 周期的数目; - 在中心对称模式下, PWM 半周期的数目。

38.4.14 比较捕获寄存器 x (TIMx_CCRx) (x=1,2,3,4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR _x [15:0]															
RW															

位	标记	功能描述
31:16	RSV	保留位
15:0	CCR	CCR _x [15:0]: 捕获/比较通道 x 的值 (Capture/Compare x value) 位 15:0 若 CC _x 通道配置为输出: CCR _x 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIM _x _CCMR 寄存器(OCxPE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 x 寄存器中。当前捕获/比较寄存器参与同计数器 TIM _x _CNT 的比较, 并在 OC _x 端口上产生输出信号。 若 CC _x 通道配置为输入: CCR _x 包含了由上一次输入捕获 x 事件(IC _x)传输的计数器值。

38.4.15 刹车死区寄存器(TIMx_BDTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	SSR	OSSI	LOCK[1:0]		UTG[7:0]							
RW	RW	RW	RW	RW	RW	RW		RW							

位	标记	功能描述
31:16	RSV	保留位
15	MOE	主输出使能:一旦刹车输入有效, 该位被硬件异步清 0。根据 AOE 位的设置值, 该位可以由软件清 0 或被自动置 1。它仅对配置为输出的通道有效。 0: 禁止 OC 和 OCN 输出或强制为空闲状态; 1: 如果设置了相应的使能位(TIM _x _CCER 寄存器的 CCxE、CCxNE 位), 则开启 OC 和 OCN 输出。 有关 OC/OCN 使能的细节, 参见 13.4.9 捕获/比较使能寄存器(TIM _x _CCER)。
14	AOE	自动输出使能 (Automatic output enable) 0: MOE 只能被软件置 1; 1: MOE 能被软件置 1 或在下一个更新事件被自动置 1(如果刹车输入无效)。

		注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位) 设为 1，则该位不能被修改。
13	BKP	刹车输入极性 (Break polarity) 0：刹车输入低电平有效； 1：刹车输入高电平有效。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位) 设为 1，则该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
12	BKE	刹车功能使能 (Break enable) 0：禁止刹车输入(BRK 及 CCS 时钟失效事件)； 1：开启刹车输入(BRK 及 CCS 时钟失效事件)。 注：当设置了 LOCK 级别 1 时(TIMx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。 注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
	SSR	运行模式下“关闭状态”选择 (Off-state selection for Run mode) 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。 参考 OC/OCN 使能的详细说明 (0 捕获/ 比较使能寄存器(TIMx_CCER))。 0：当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OC/OCN 使能输出信号=1。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位) 设为 2，则该位不能被修改。
	OSSI	空闲模式下“关闭状态”选择 (Off-state selection for Idle mode) 该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说 (0 捕获/ 比较使能寄存器(TIMx_CCER))。 0：当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 首先输出其空闲电平，然后置 OC/OCN 使能输出信号=1。 注：一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位) 设为 2，则该位不能被修改。
	LOCK[1:0]	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 00：锁定关闭，寄存器无写保护； 01：锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位； 10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出，CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位； 11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出，CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位)； 注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMx_BDTR 寄存器，则其内容冻结直至复位。
	UTG[7:0]	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间： $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times T_{dtg}, T_{dtg} = T_{DTS}$; $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times T_{dtg}, T_{dtg} = 2 \times T_{DTS}$; $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times T_{dtg}, T_{dtg} = 8 \times T_{DTS}$;

		$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times T_{dtg}$, $T_{dtg} = 16 \times T_{DTS}$; 例: 若 $T_{DTS} = 125ns$ (8MHZ), 可能的死区时间为: 0 到 15875ns, 若步长时间为 125ns; 16 μs 到 31750ns, 若步长时间为 250ns; 32 μs 到 63 μs , 若步长时间为 1 μs ; 64 μs 到 126 μs , 若步长时间为 2 μs ; 注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位) 设为 1、2 或 3, 则不能修改这些位。

38.4.16 Option 寄存器 1(TIMx_OR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														OCREF_CLR	
														RW	

位	标记	功能描述
31:1	RSV	保留位
0	OCREF_CLR	OCREF_CLR 源选择 0: VC1 滤波输出 1: VC2 滤波输出

38.4.17 比较模式寄存器(TIMx_CCMR3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6CE	OC6M[2:0]		OC6PE	OC6FE	RSV		OC5CE	OC5M[2:0]		OC5PE	OC5FE	RSV			
RW	RW		RW	RW			RW	RW		RW	RW				

位	标记	功能描述
31:16	RSV	保留位
15	OC6CE	输出比较 6 清 0 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	RSV	保留位
7	OC5CE	输出比较 5 清 0 使能

6:4	OC5M[2:0]	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	RSV	保留位

38.4.18 比较寄存器(TIMx_CCRx) (x=5,6)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR _x [15:0]															
RW															

位	标记	功能描述
31:16	RSV	保留位
15:0	CCR _x	比较值

38.4.19 辅助功能寄存器 1(TIMx_AF1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														ETRSEL[3:2]	
														RW	
15		14		13		12		11		10		9		8	
ETRSEL[1:0]		Reserved				BKVC2P		BKVC1P		BKINP		RSV			
RW						RW		RW		RW					
7		6		5		4		3		2		1		0	
Reserved										BKVC2E		BKVC1E		BKINE	
										RW		RW		RW	

位	标记	功能描述
31:16	RSV	保留位
17:14	ETRSEL	ETR 输入源选择 0000: 端口 0001: VC1 滤波输出 0010: VC2 滤波输出 0011: ADC 低阈值比较中断

		0100: ADC 高阈值比较中断 0101: ADC 区间阈值比较中断 其他: 保留
13:12	RSV	保留位
11	BKVC2P	VC2 刹车输入极性选择 0: 高电平有效 1: 低电平有效
10	BKVC1P	VC1 刹车输入极性选择 0: 高电平有效 1: 低电平有效
9	BKINP	端口刹车输入极性选择 0: 高电平有效 1: 低电平有效
8:3	RSV	保留位
2	BKVC2E	VC2 输入使能 1: 使能 0: 无效
1	BKVC1E	VC1 输入使能 1: 使能 0: 无效
0	BKINE	BKIN 输入使能 1: 使能 0: 无效

38.4.20 辅助功能寄存器 2(TIMx_AF2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11		10		9		8					
Reserved				BK2VC2P		BK2VC1P		BK2INP		Reserved					
				RW		RW		RW							
7	6	5	4	3		2		1		0					
Reserved						BK2VC2E		BK2VC1E		BK2INE					
						RW		RW		RW					

位	标记	功能描述
31:12	RSV	保留位
11	BK2VC2P	VC2 刹车输入极性选择 0: 高电平有效 1: 低电平有效
10	BK2VC1P	VC1 刹车输入极性选择 0: 高电平有效 1: 低电平有效
9	BK2INP	端口刹车输入极性选择

		0: 高电平有效 1: 低电平有效
8:3	RSV	保留位
2	BK2VC2E	VC2 输入使能 1: 使能 0: 无效
1	BK2VC1E	VC1 输入使能 1: 使能 0: 无效
0	BK2INE	BK2IN 输入使能 1: 使能 0: 无效

38.4.21 TI 选择寄存器 (TIMx_TISEL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TI2SEL				Reserved				TI1SEL			
				RW								RW			

位	标记	功能描述
31:12	Reserved	保留位
11:8	TI2SEL	0000: CH2 输入 0001: VC2 输出 其他: 保留
7:4	Reserved	保留位
3:0	TI1SEL	0000: CH1 输入 0001: VC1 输出 其他: 保留

39. 液晶控制器（LCD）

39.1 LCD 简介

LCD 控制器是一款适用于单色无源液晶显示器(LCD)的数字控制器/驱动器，最多具有 8 个公用端子 (COM) 和 32 个区段端子 (SEG)，用以驱动(4x32)或(8x32)个 LCD 图像元素。端子的确切数量取决于数据手册中所述的器件引脚。

LCD 由若干区段（像素或完整符号）组成，这些区段均可点亮或熄灭。每个区段都包含一层在两根电极之间对齐的液晶分子。当向液晶施加高于阈值电压的电压时，相应的区段可见。区段电压必须为交流，以避免液晶中出现电泳效应（这将影响显示效果）。之后，必须在区段两端生成波形以避免出现直流。

词汇表

液晶（LCD）：无源显示面板，带有直接引向区段的端子。

公用（COM）：连接到多个区段的电气连接端子。

偏置（BIAS）：驱动 LCD 时使用的电压等级，定义为 $1/(\text{驱动 LCD 显示的电压等级数}-1)$ 。

区段（SEG）：最小可视单元（LCD 显示器上的最小组成元素，线条或点）。

占空比（DUTY）：定义为 $1/(\text{LCD 显示器上的公用端子数})$ 的数字。

帧：写入区段的波形的一个周期。

帧速率：每秒帧数，即每秒激励 LCD 区段的次数。

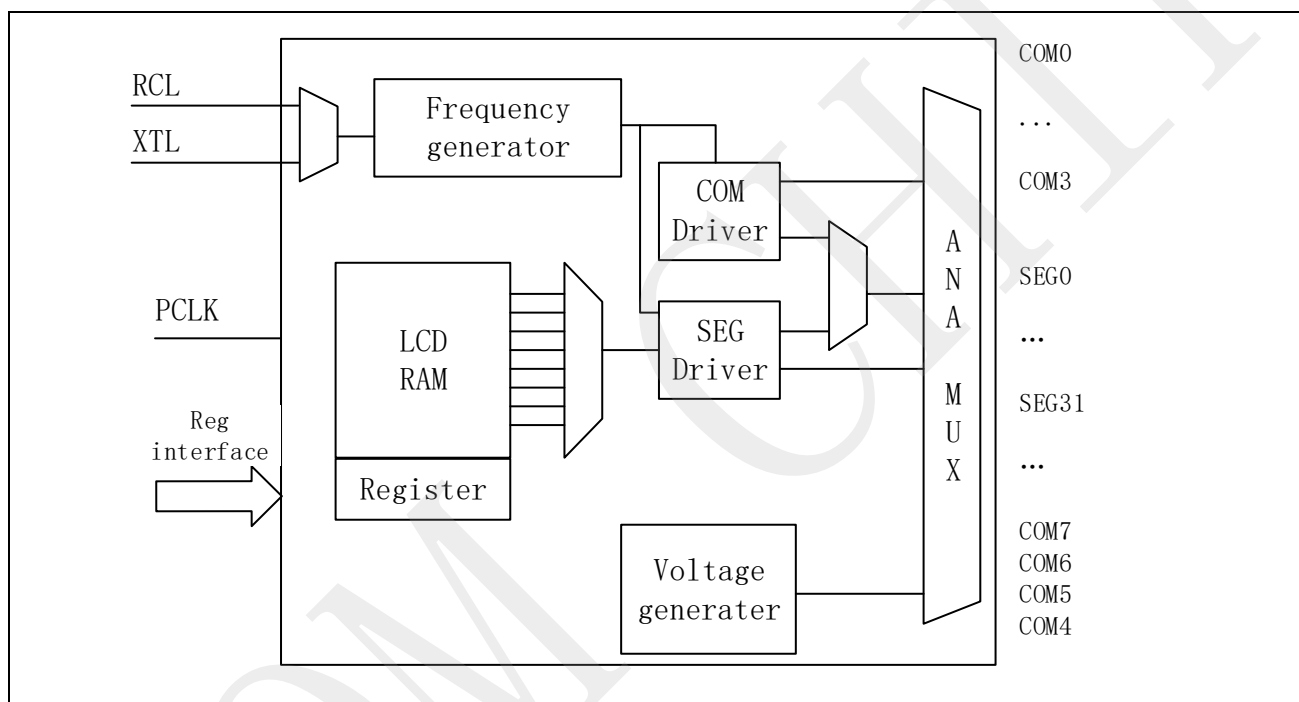
39.2 LCD 主要特性

- 高度灵活的帧速率控制。
- 支持静态、1/2、1/3、1/4、1/6 和 1/8 占空比。
- 支持 1/2、1/3 偏置。
- 多达 8 个寄存器的 LCD 数据 RAM。
- 可通过软件配置 LCD 的对比度。
- 3 种驱动波形生成方式
 - 内部电阻分压、外部电阻分压，外部电容分压方式

- 可通过软件配置内部电阻分压方式的功耗,从而匹配 LCD 面板所需的电容电荷

- 支持低功耗模式: LCD 控制器可在 Active、Sleep、DeepSleep 模式下进行显示。
- 可配置帧中断。
- 支持 LCD 闪烁功能且可配置多种闪烁频率
- 未使用的 LCD 区段和公共引脚可配置为数字或模拟功能。

39.3 LCD 框图



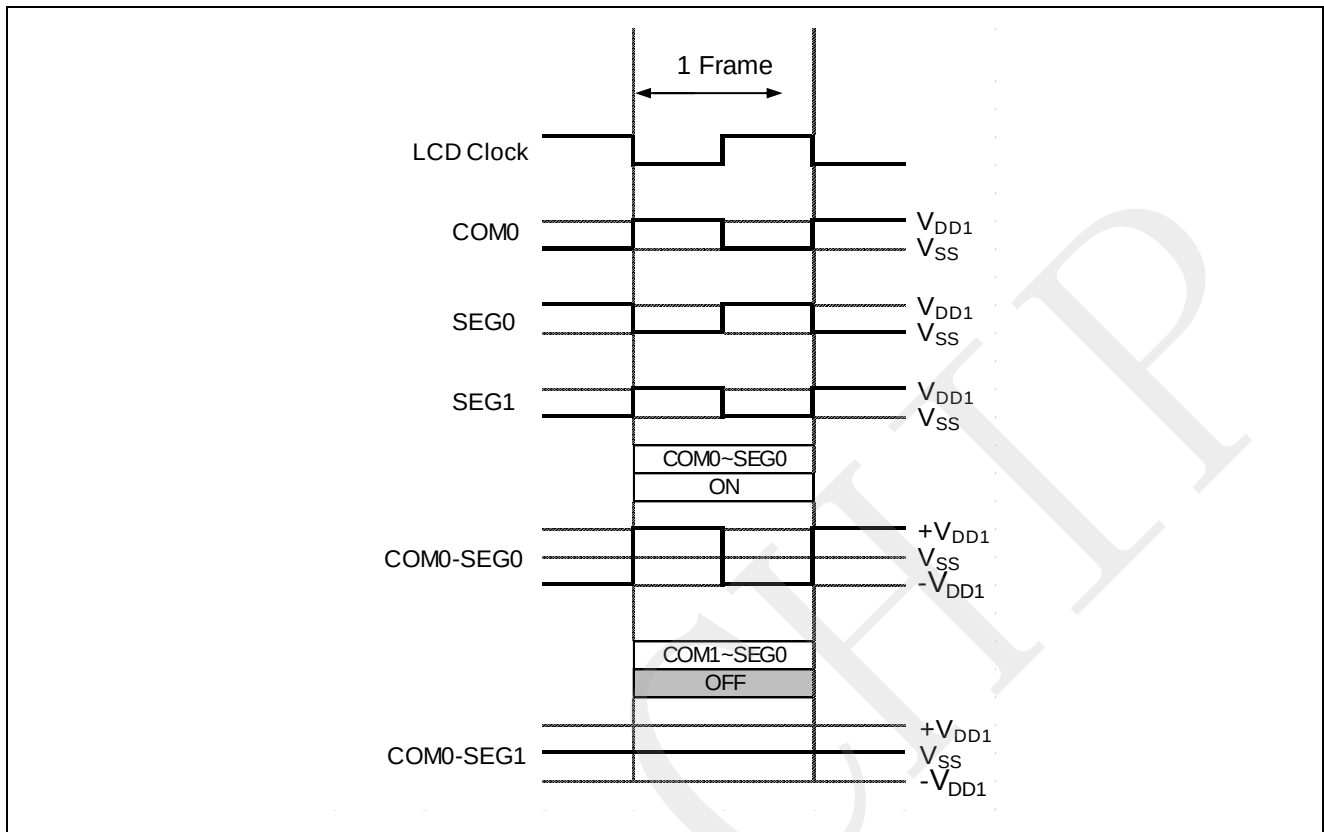
39.4 LCD 驱动波形

LCD 支持 5 种占空比 (Duty) 的驱动波形: 静态、1/2、1/3、1/4、1/6 和 1/8, 由 LCD_CR0.Duty 进行设置。LCD 支持 2 种偏置 (Bias) 的驱动波形: 1/2、1/3, 由 LCD_CR0.Bias 进行设置。建议的组合方式如下表所示:

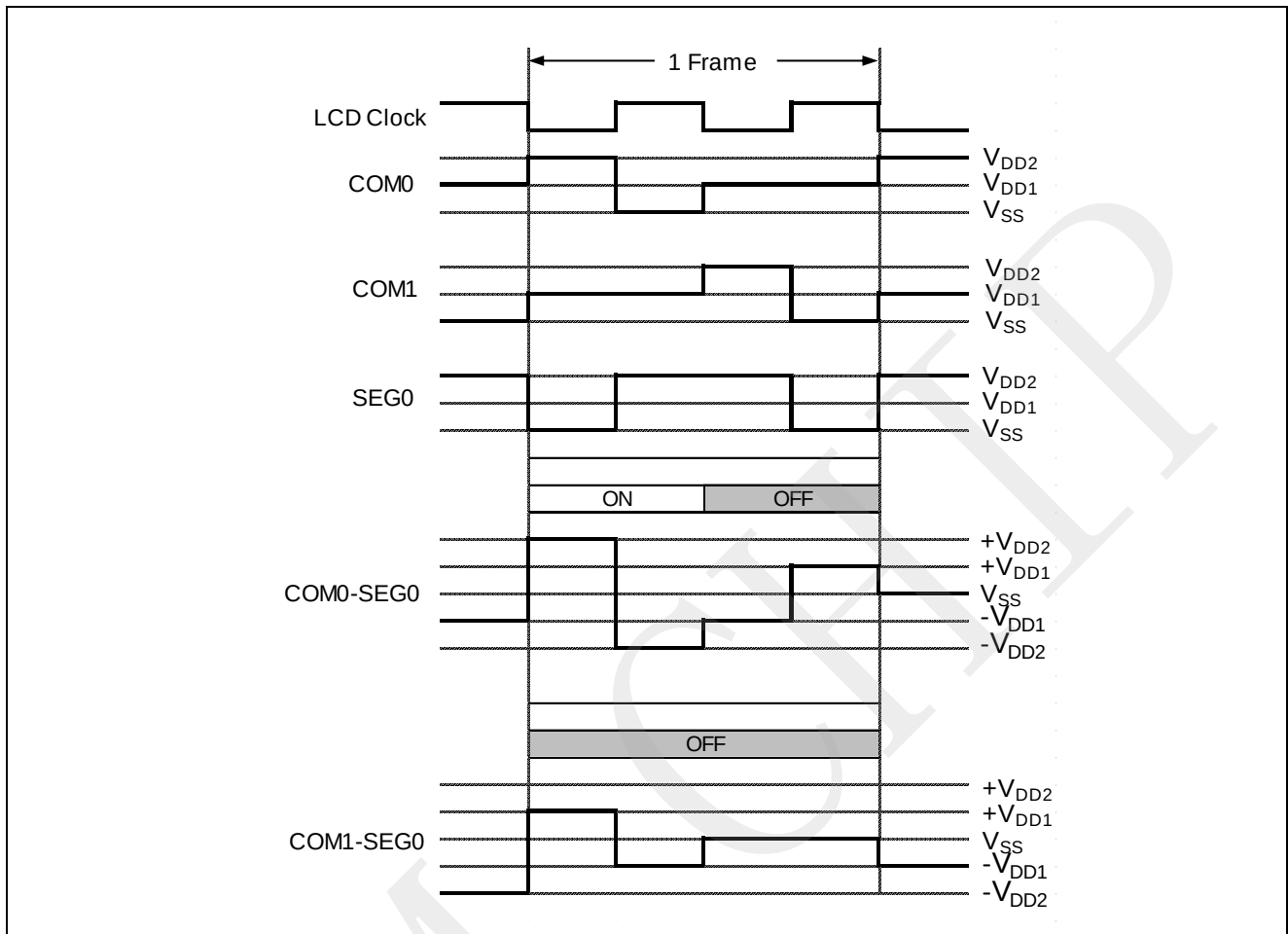
	1/2 Duty	1/3 Duty	1/4 Duty	1/6 Duty	1/8 Duty
1/2 Bias	√	√	不推荐	不推荐	不推荐
1/3 Bias	不推荐	不推荐	√	√	√

各模式下的驱动波形如下方所示:

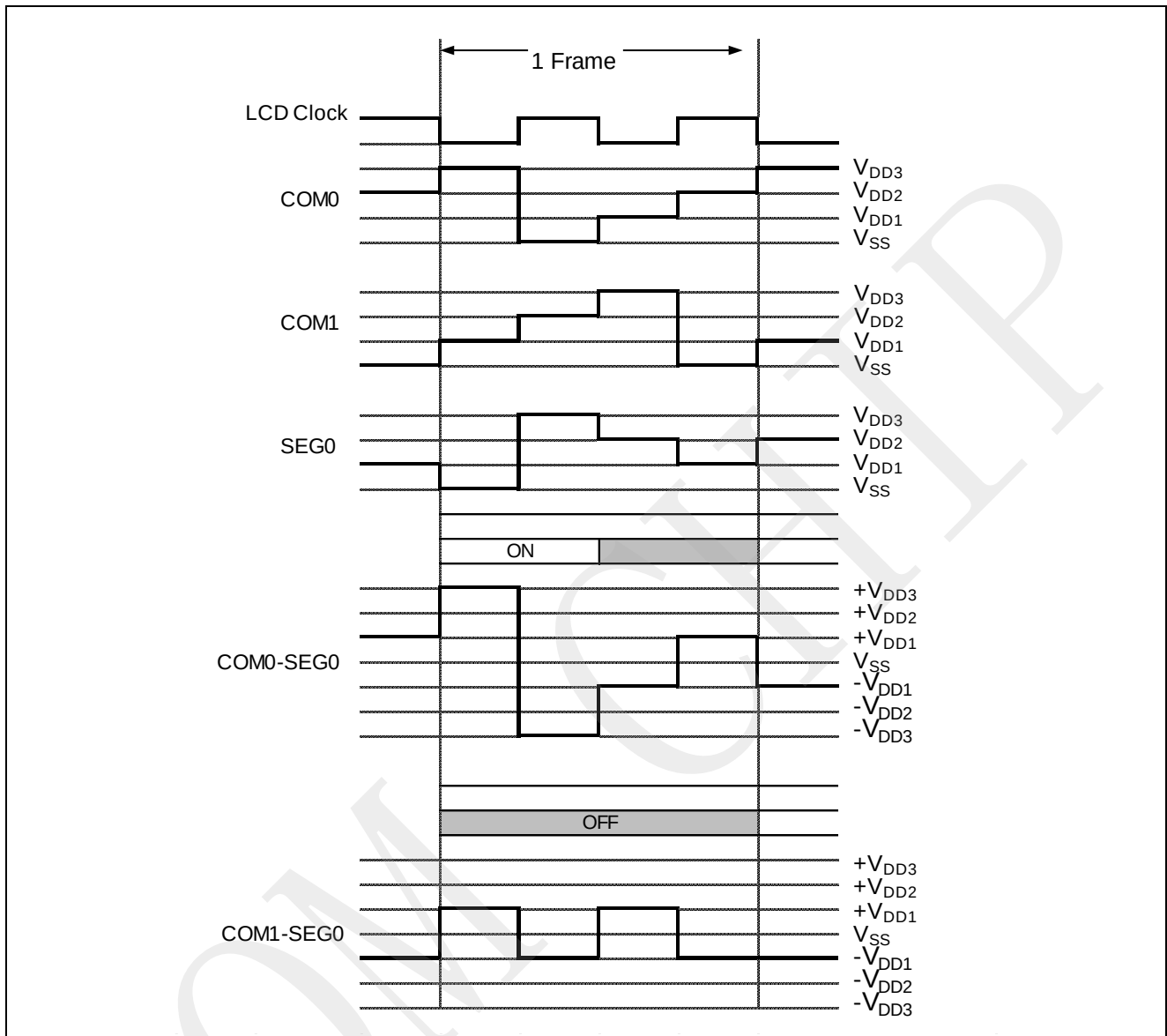
39.4.1 静态驱动波形



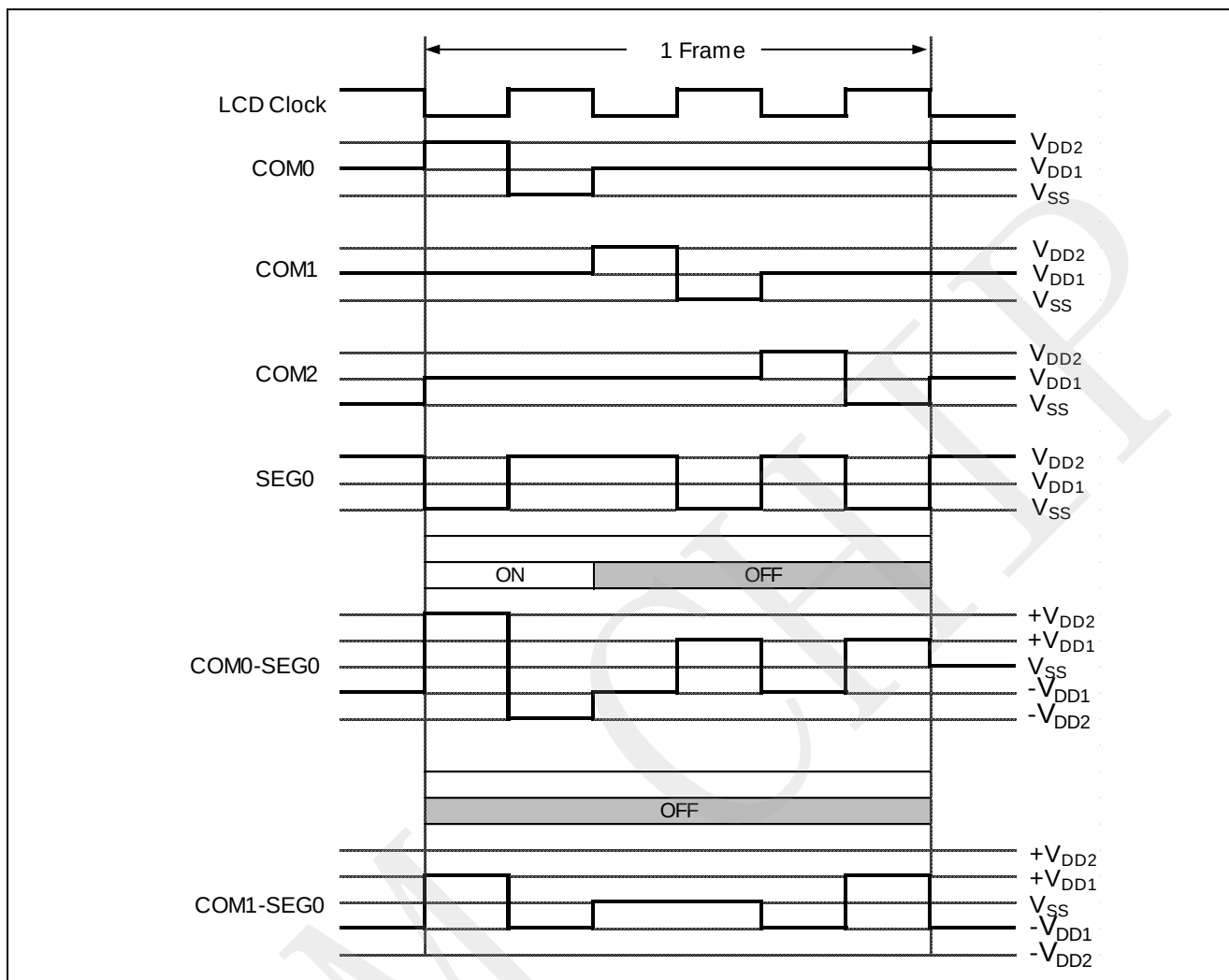
39.4.2 1/2Duty 1/2Bias 驱动波形



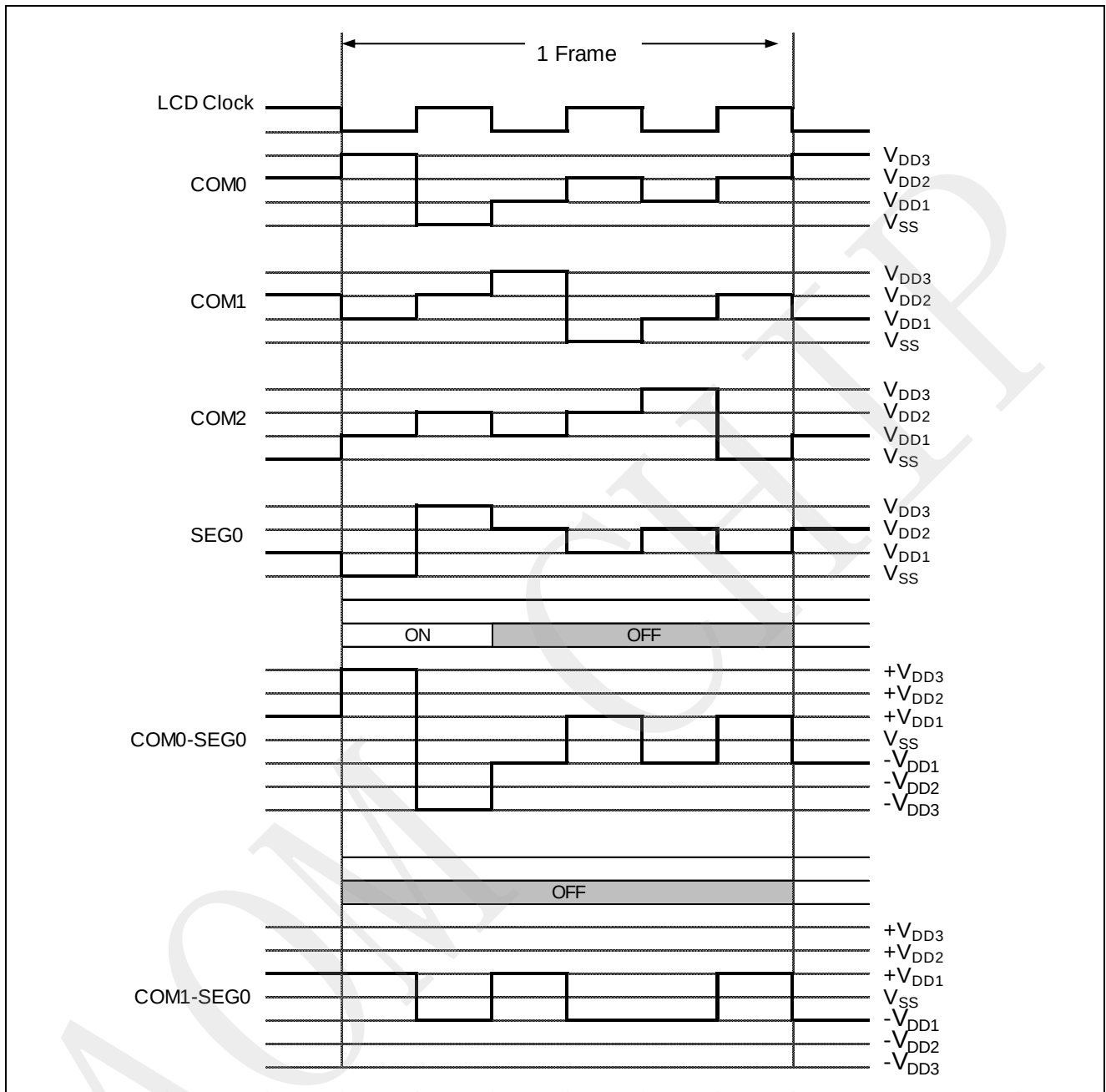
39.4.3 1/2Duty 1/3Bias 驱动波形



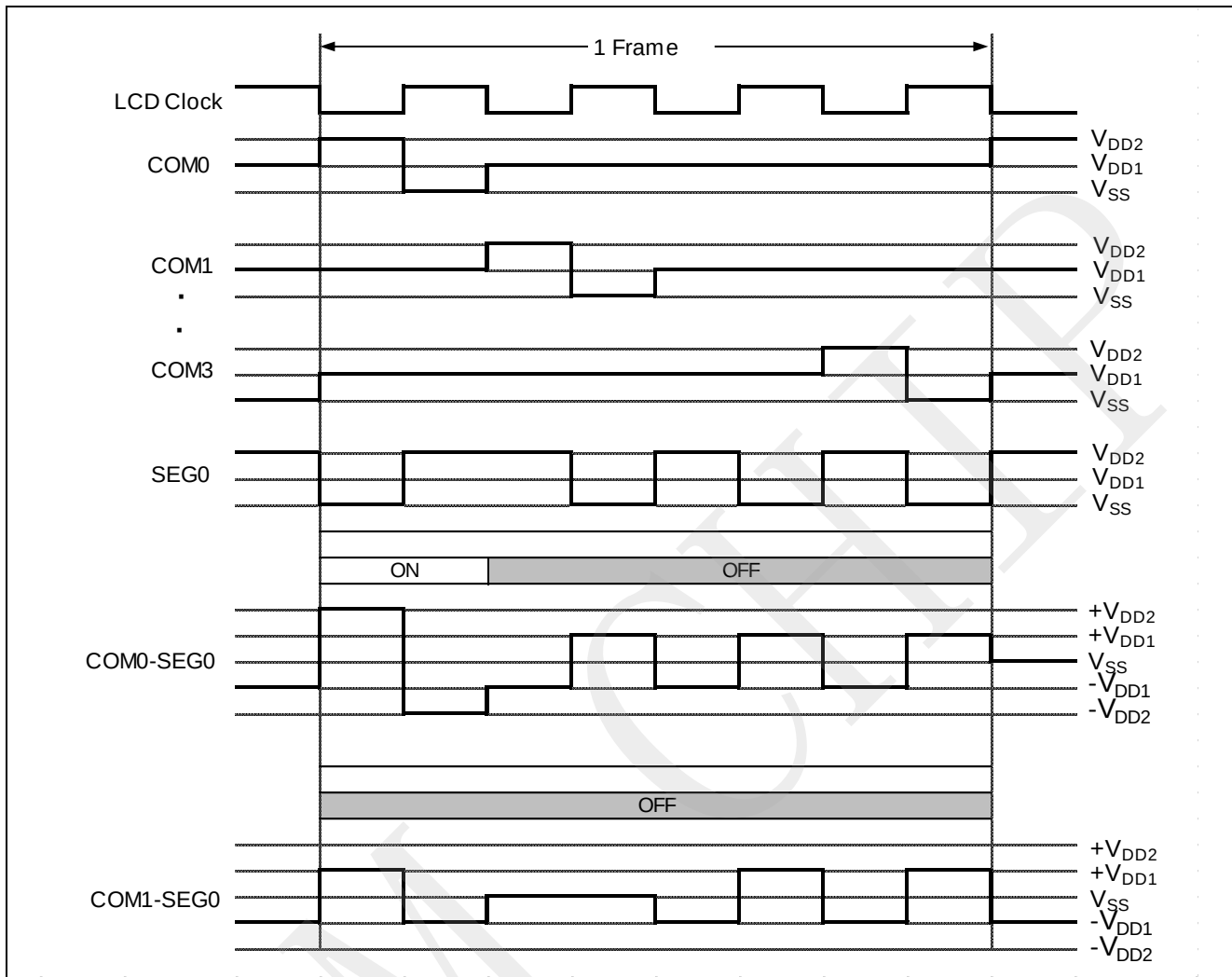
39.4.41/3Duty 1/2Bias 驱动波形



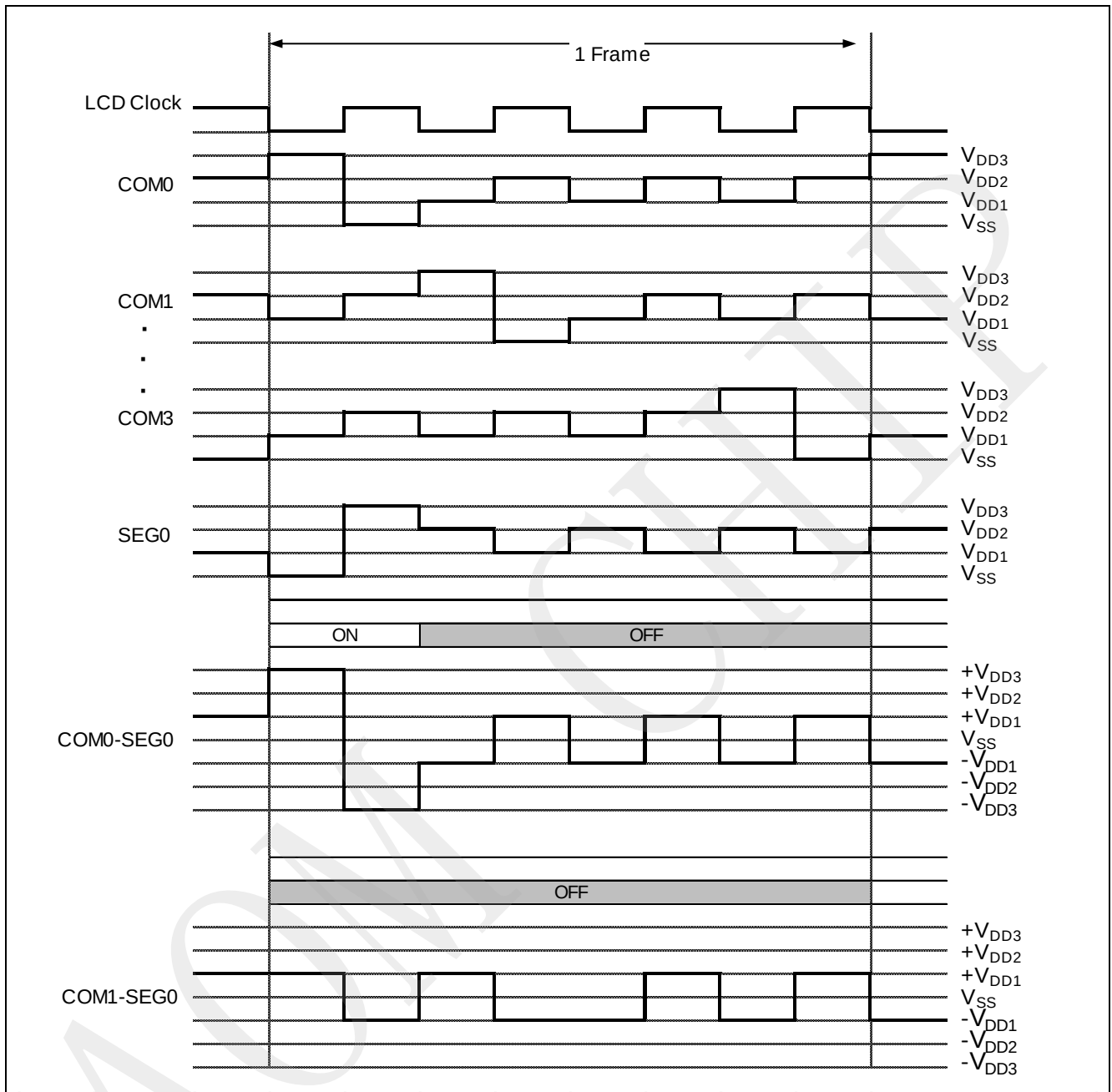
39.4.5 1/3Buty 1/3Bias 驱动波形



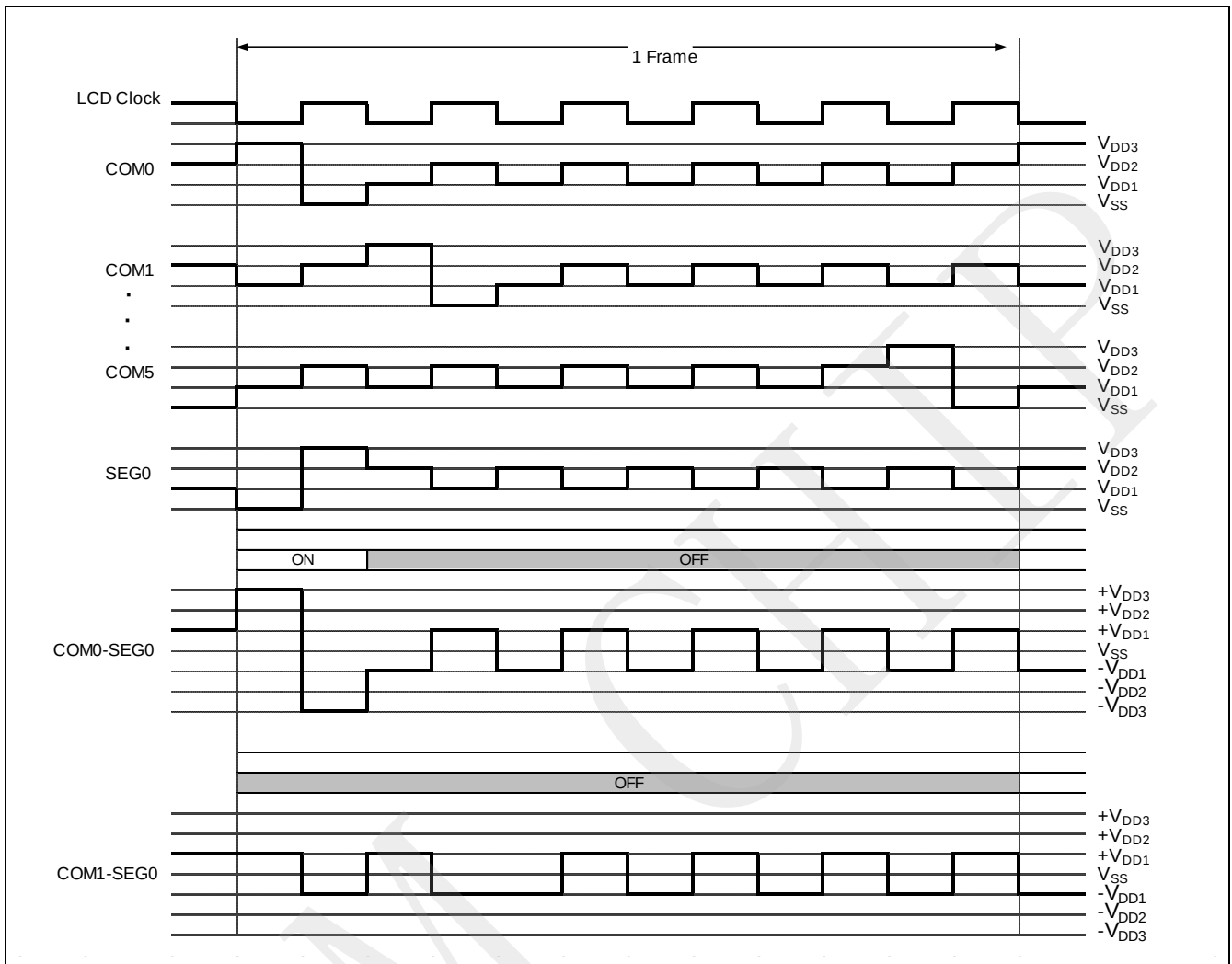
39.4.6 1/4Duty 1/2Bias 驱动波形



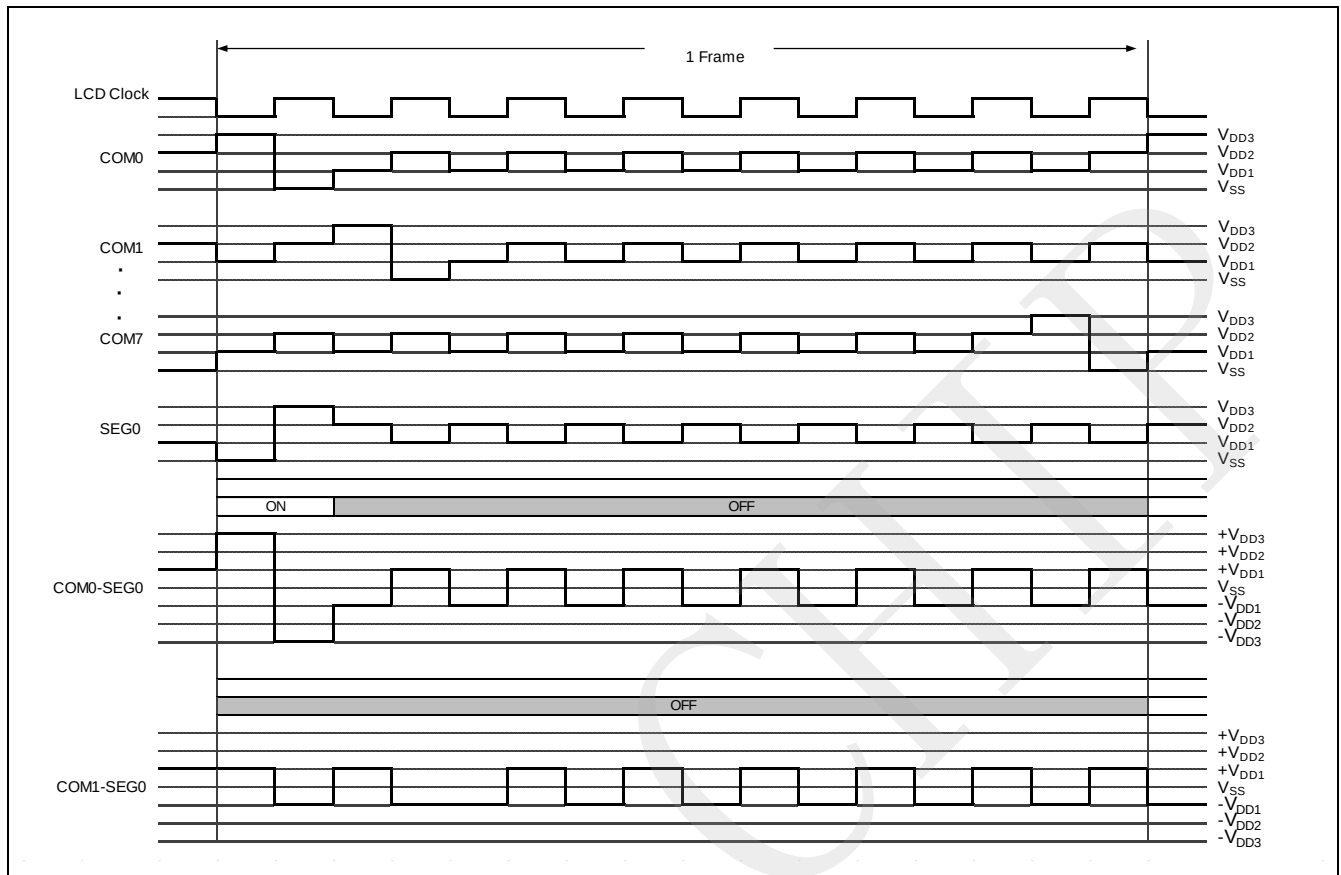
39.4.7 1/4Duty 1/3Bias 驱动波形



39.4.81/6Duty 1/3Bias 驱动波形



39.4.9 1/8Duty 1/3Bias 驱动波形



39.5 LCD Bias 产生电路

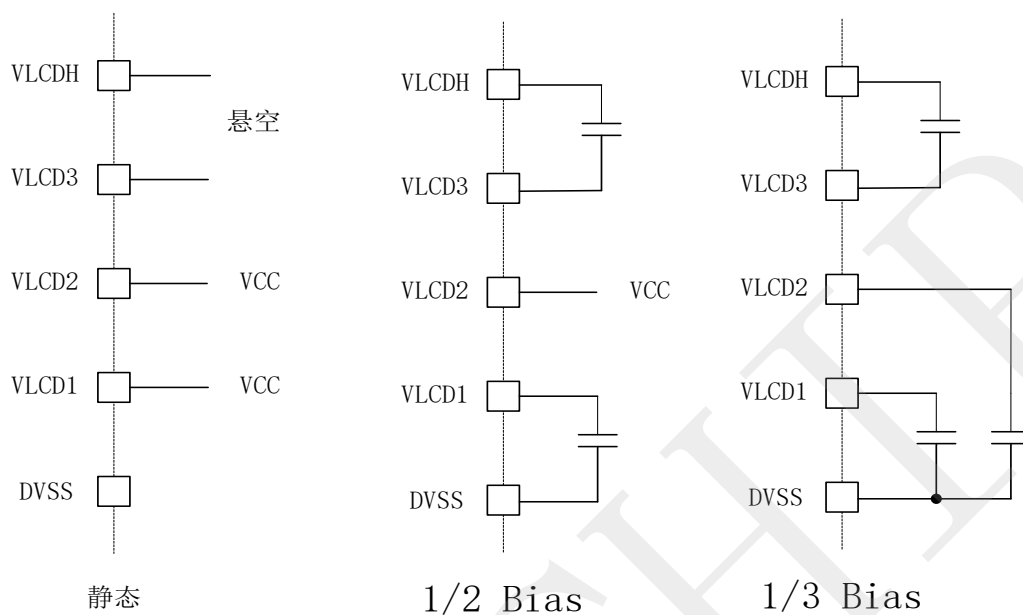
LCD 的 Bias 电压具有 3 种来源：内部电阻分压、外部电阻分压、外部电容分压。当选择内部电阻分压时，芯片会自动切换内部的电路以产生符合 Bias 和 Duty 的电压。当选择外部电阻分压或外部电容分压时，需要用户在芯片的外围引脚搭建相关电路。

39.5.1 内部电阻模式

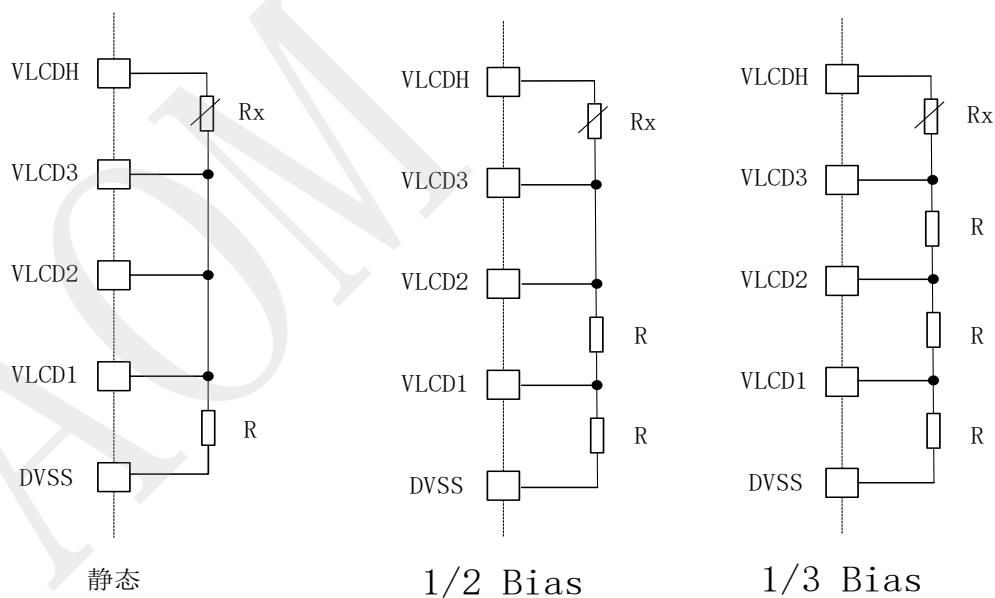
内部电阻模式 VLCDH,VLCD1~VLCD3 可以作为 LCD SEG 输出或者 IO 端口使用。内部电阻模式，LCD 的驱动电压由 CR0.Contrast 控制，如下表所示：

CR0.Contrast	VLCD(1/3 bias)	VLCD(1/2 bias)
0	1.00 * VCC	1.00 * VCC
1	0.94 * VCC	0.92 * VCC
2	0.9 * VCC	0.85 * VCC
3	0.85 * VCC	0.8 * VCC
4	0.81 * VCC	0.75 * VCC
5	0.78 * VCC	0.70 * VCC
6	0.75 * VCC	0.66 * VCC
7	0.72 * VCC	0.63 * VCC
8	0.70 * VCC	0.61 * VCC
9	0.67 * VCC	0.58 * VCC
10	0.65 * VCC	0.55 * VCC
11	0.63 * VCC	0.53 * VCC
12	0.61 * VCC	0.51 * VCC
13	0.59 * VCC	0.48 * VCC
14	0.57 * VCC	0.47 * VCC
15	0.55 * VCC	0.45 * VCC

39.5.2 外部电容模式



39.5.3 外部电阻模式



注:

- Rx 为可调电阻，用于调节 LCD 显示对比度。
- 根据使用 LCD 屏幕选择合适的电阻 R。

39.6 DMA

LCD 支持软件和硬件触发 DMA 数据传输，可以将需要显示的内容从 RAM 或 ROM 中自动搬到 LCD 显示 RAM 中。硬件触发使用的是帧中断信号。LCD 使用的 DMA 通道号为[6]。

DMA 数据传送配置流程：

1. 使能 DMA
2. 使能 DMA 通道使能
3. 选择 LCD DMA
4. 设置传输类型、传输长度、传输方式
5. 设置源起始地址，目标起始地址
6. 设置源地址、目标地址的递增方式
7. 根据需要使能 DMA 中断
8. 使能 LCD DMA 触发

39.7 中断

当 LCD 设置有效时，LCD 中断可以配置为帧数产生中断。

39.8 LCD 显示模式

LCD 支持两种显示模式。一种以 COM 为显示单元，同一个 SEG 的所有 COM 段在同一字节中（模式 0）。另外一种为同一个 COM 的不同 SEG 在同一个字节中（模式 1）。根据 LCD 面板选择合适的显示方式可以简化程序操作。

39.8.1 LCD 显示

1/8 Duty

Bit00	Bit01	Bit02	Bit03	Bit04	Bit05	Bit06	Bit07	Bit08	Bit09	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG29								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7

1/6 Duty

Bit00	Bit01	Bit02	Bit03	Bit04	Bit05	Bit06	Bit07	Bit08	Bit09	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31		
COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7		
SEG0								SEG1								SEG2								SEG3									LCDRAM0
SEG4								SEG5								SEG6								SEG7									LCDRAM1
SEG8								SEG9								SEG10								SEG11									LCDRAM2
SEG12								SEG13								SEG14								SEG15									LCDRAM3
SEG16								SEG17								SEG18								SEG19									LCDRAM4
SEG20								SEG21								SEG22								SEG23									LCDRAM5
SEG24								SEG25								SEG26								SEG29									LCDRAM6
SEG28								SEG29								SEG30								SEG31									LCDRAM7

1/4 Duty

Bit00	Bit01	Bit02	Bit03	Bit04	Bit05	Bit06	Bit07	Bit08	Bit09	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31		
COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7		
SEG0								SEG1								SEG2								SEG3									LCDRAM0
SEG4								SEG5								SEG6								SEG7									LCDRAM1
SEG8								SEG9								SEG10								SEG11									LCDRAM2
SEG12								SEG13								SEG14								SEG15									LCDRAM3
SEG16								SEG17								SEG18								SEG19									LCDRAM4
SEG20								SEG21								SEG22								SEG23									LCDRAM5
SEG24								SEG25								SEG26								SEG29									LCDRAM6
SEG28								SEG29								SEG30								SEG31									LCDRAM7

1/3 Duty 1/2 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31		
COM0	COM1	COM2						COM0	COM1	COM2						COM0	COM1	COM2						COM0	COM1	COM2							LCDRAM0
SEG0								SEG1								SEG2								SEG3									LCDRAM1
SEG4								SEG5								SEG6								SEG7									LCDRAM2
SEG8								SEG9								SEG10								SEG11									LCDRAM3
SEG12								SEG13								SEG14								SEG15									LCDRAM4
SEG16								SEG17								SEG18								SEG19									LCDRAM5
SEG20								SEG21								SEG22								SEG23									LCDRAM6
SEG24								SEG25								SEG26								SEG29									LCDRAM7
SEG28								SEG29								SEG30								SEG31									

39.9 LCD 寄存器

基地址 0x40002400

寄存器	偏移地址	描述
LCD_CR0	0x000	LCD 配置寄存器0
LCD_CR1	0x004	LCD 配置寄存器1
LCD_INTCLR	0x008	LCD 中断清除寄存器
LCD_SOEN1	0x00C	LCD 输出配置寄存器
LCD_COEN	0x014	LCD 输出配置寄存器
LCD_RAM0	0x040	LCD RAM0
LCD_RAM1	0x044	LCD RAM1
LCD_RAM2	0x048	LCD RAM2
LCD_RAM3	0x04C	LCD RAM3
LCD_RAM4	0x050	LCD RAM4
LCD_RAM5	0x054	LCD RAM5
LCD_RAM6	0x058	LCD RAM6
LCD_RAM7	0x05C	LCD RAM7

表 39-1 LCD 寄存器

39.9.1 配置寄存器 0 (LCD_CR0)

偏移地址: 0x000

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Contrast				INRS			Duty			RSV	Bias	bump	CpClk		EN
RW				RW			RW			RW	RW	RW	RW		RW

位	标记	功能描述
31:16	Reserved	保留
15:12	Contrast	LCD对比度调整 注: 仅当Bias电压来源选择内部电阻分压时有效。 Constrast值越大, LCD波形的幅度越小。 0X0时, LCD波形幅度最大, 对比度最大; 0XF时, LCD波形幅度最小, 对比度最小;

11:9	INRS	内部电阻选择 100: 内部电阻分压, 大功耗模式 010: 内部电阻分压, 中功耗模式 001: 内部电阻分压, 小功耗模式 000: 外部电阻模式, 需要外部电路配合																															
8:6	Duty	LCD duty配置 000: 静态 001: 1/2 duty 010: 1/3 duty 011: 1/4 duty 100: Reserved 101: 1/6 duty 110: Reserved 111: 1/8 duty <table><tr><td>duty</td><td>PAD</td><td>FUNCTION</td></tr><tr><td rowspan="4">1/4 duty</td><td>COM4SEG51PAD</td><td>SEG51</td></tr><tr><td>COM5SEG50PAD</td><td>SEG50</td></tr><tr><td>COM6SEG49PAD</td><td>SEG49</td></tr><tr><td>COM7SEG48PAD</td><td>SEG48</td></tr><tr><td rowspan="4">1/6 duty</td><td>COM4SEG51PAD</td><td>COM4</td></tr><tr><td>COM5SEG50PAD</td><td>COM5</td></tr><tr><td>COM6SEG49PAD</td><td>SEG49</td></tr><tr><td>COM7SEG48PAD</td><td>SEG48</td></tr><tr><td rowspan="4">1/8 duty</td><td>COM4SEG51PAD</td><td>COM4</td></tr><tr><td>COM5SEG50PAD</td><td>COM5</td></tr><tr><td>COM6SEG49PAD</td><td>COM6</td></tr><tr><td>COM7SEG48PAD</td><td>COM7</td></tr></table>		duty	PAD	FUNCTION	1/4 duty	COM4SEG51PAD	SEG51	COM5SEG50PAD	SEG50	COM6SEG49PAD	SEG49	COM7SEG48PAD	SEG48	1/6 duty	COM4SEG51PAD	COM4	COM5SEG50PAD	COM5	COM6SEG49PAD	SEG49	COM7SEG48PAD	SEG48	1/8 duty	COM4SEG51PAD	COM4	COM5SEG50PAD	COM5	COM6SEG49PAD	COM6	COM7SEG48PAD	COM7
duty	PAD	FUNCTION																															
1/4 duty	COM4SEG51PAD	SEG51																															
	COM5SEG50PAD	SEG50																															
	COM6SEG49PAD	SEG49																															
	COM7SEG48PAD	SEG48																															
1/6 duty	COM4SEG51PAD	COM4																															
	COM5SEG50PAD	COM5																															
	COM6SEG49PAD	SEG49																															
	COM7SEG48PAD	SEG48																															
1/8 duty	COM4SEG51PAD	COM4																															
	COM5SEG50PAD	COM5																															
	COM6SEG49PAD	COM6																															
	COM7SEG48PAD	COM7																															
5	RSV	保留位																															
4	Bias	LCD Bias配置 00: 1/3 bias 01: 1/2 bias																															
3	bump	电荷泵模式选择																															
2:1	CpClk	电压泵时钟频率选择 00: 2k Hz 01: 4k Hz 10: 8k Hz 11: 16k Hz																															
0	EN	LCD使能控制 1: 使能 0: 禁止																															

39.9.2 配置寄存器 1 (LCD_CR1)

偏移地址: 0x004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved						INTF	IE	DmaEn	LcdFS						
						RO	RW	RW	RW						
7	6	5	4	3	2	1	0								
ClkCS	BlinkEn	BlinkCnt													
RW	RW	RW													

位	标记	功能描述
31:13	Reserved	保留
12	INTF	中断标志 1: 中断 0: 无中断
11	IE	中断使能 1: 使能 0: 禁止
10	DmaEn	DMA硬件触发使能 1: 使能LCD中断触发DMA, 0: 禁止LCD中断触发DMA
9:8	LcdFS	LCD扫描频率选择 00: 128Hz 01: 256Hz 10: 512Hz 11: 64Hz 注: LCD帧频率 = LCD扫描频率×Duty
7	ClkCS	LCD时钟源选择 1: XTL 0: RCL
6	BlinkEn	LCD闪屏配置 1: 使能 0: 禁止
5:0	BlinkCnt	闪屏频率与LCD中断间隔设置 注: LCD闪烁频率为 = LCD帧频率 / (BlinkCnt+1) LCD中断间隔= (BlinkCnt+1)*(1/LCD 帧频率)

39.9.3 中断清除寄存器 (LCD_INTCLR)

偏移地址: 0x008

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				INTF	Reserved										
				R1W0											

位	标记	功能描述
31:12	Reserved	保留
11	INTF	中断标志清除, 写0清除, 写1无效
10:0	Reserved	保留

39.9.4 输出配置寄存器 1 (LCD_SOEN1)

偏移地址: 0x00C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:0	Sx	Segx输出控制位 1: SEG输出使能 0: SEG输出关闭, 可以使用其他功能, 如 IO, 模拟输入输出

39.9.5 输出配置寄存器 3 (LCD_COEN)

偏移地址: 0x014

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved							MUX								
							RW								
7	6	5	4	3	2	1	0								
C7S48	C6S49	C5D50	C4S51	C3	C2	C1	C0								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:9	RSV	保留
8	MUX	SEG23~SEG20端口功能选择，详细参考选择表格
7:4	SxCy	Segx/COMy输出控制位 1: SEG/COM输出使能 0: SEG/COM输出关闭，可以使用其他功能，如 IO, 模拟输入输出 SEG COM 管脚功能选择由CR0.DUTY决定
3:0	Cx	COMx输出控制位 1: COM输出使能 0: COM输出关闭，可以使用其他功能，如 IO, 模拟输入输出

VLCDXSEGXPAD 状态描述			寄存器如何配置(已更新)								
			MUX	S<23:20>	INRS						
VLCDXSEGXPAD 选择 GPIO, LCD disable(LCD_ON=0)时			0	0000	X	X	X				
								VLCDHSEG23=IO			
								VLCD3SEG22=IO			
								VLCD2SEG21=IO			
VLCDXSEGXPAD 选择 IO, LCD enable(LCD_ON=1)时, 这时只能选择内部电阻工作模式			0	0000	1	1	0				
								VLCDHSEG23=IO			
								VLCD3SEG22=IO			
								VLCD2SEG21=IO			
			中电阻(中电流)模式			0	0000	0	1	0	
											VLCDHSEG23=IO
											VLCD3SEG22=IO
											VLCD2SEG21=IO
			大电阻(小电流)模式			0	0000	1	0	0	
											VLCDHSEG23=IO
											VLCD3SEG22=IO
											VLCD2SEG21=IO
VLCD1SEG20=IO											
选择外部电阻工作模式, 内部电阻断路			1	1111	0	0	0				
								VLCDHSEG23=VOUT			
								VLCD3SEG22=VLCD3			
								VLCD2SEG21=VLCD2			
选择内部电阻工作模式			1	XXXX	1	1	0				
								VLCDHSEG23=SEG23/IO			
								VLCD3SEG22=SEG22/IO			
								VLCD2SEG21=SEG21/IO			
			中电阻(中电流)模式			1	XXXX	0	1	0	
											VLCDHSEG23=SEG23/IO
											VLCD3SEG22=SEG22/IO
											VLCD2SEG21=SEG21/IO
			大电阻(小电流)模式			1	XXXX	1	0	0	
											VLCDHSEG23=SEG23/IO
											VLCD3SEG22=SEG22/IO
											VLCD2SEG21=SEG21/IO
VLCD1SEG20=SEG20/IO											
选择外部电容工作模式, 内部电阻断路			1	0000	0	0	1				
								VLCDHSEG23=DH1			
								VLCD3SEG22=DH2			
								VLCD2SEG21=VLCD2			
VLCD1SEG20=VLCD1											

39.9.6 LCD_RAM0~7

偏移地址：0x040~0x5c

复位值：无

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:0	Dx	LCD点输出，显示参考LCD显示模式 0：对应的SEG COM交叉点不亮； 1：对应的SEG COM交叉点亮；

40. 模数转换器（ADC）

40.1 模块简介

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。内部集成了一个 12 位高精度、高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 12 位转换精度；
- 1MSPS 转换速度；
- 22 路输入通道，包括 17 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 AVCC 电压、1 路内建 BGR 1.2V 电压、2 路 DAC 内部输出；
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；
- ADC 的电压输入范围：0~Vref；
- 4 种转换模式：单次转换、顺序扫描连续转换、插队扫描连续转换、连续转换累加；
- 输入通道电压阈值监测；
- 软件可配置 ADC 的转换速率；
- 内置信号跟随器，可转换高阻信号；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

40.3 转换时序及转换速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由转换过程及逐次比较过程组成。

其中转换过程需要 4~12 个 ADCCLK，由 ADC_CR0.SAM 配置；逐次比较过程需要 16 个 ADCCLK。所以，一次 ADC 转换共需要 20~28 个 ADCCLK。

ADC 转换速度的单位为 SPS，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：ADCCLK 的频率 / 一次 ADC 转换所需要的 ADCCLK 的个数。

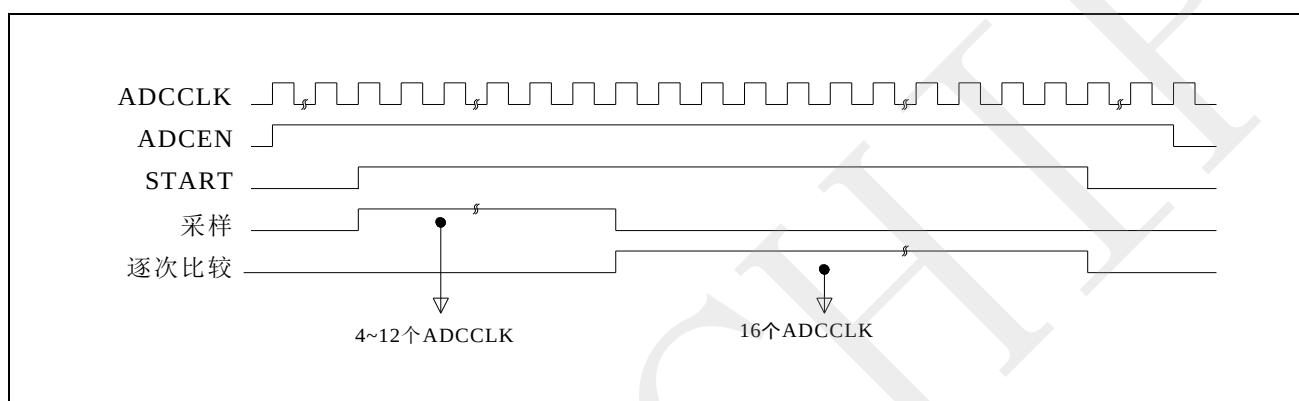


图 40-2ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 AVCC 电压相关，最高转换速度如下表所示：

ADC 参考电压	AVCC 电压	最高转换速度	最大 ADCCLK 频率
内部 1.5V	1.8V~5.5V	200kSPS	4MHz
内部 2.5V	2.8V~5.5V	200kSPS	4MHz
AVCC / ExRef	1.8V~2.4V	200kSPS	4MHz
AVCC / ExRef	2.4V~2.7V	500kSPS	16MHz
AVCC / ExRef	2.7V~5.5V	1MSPS	24MHz

40.4 单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 22 路 ADC 通道进行转换。该模式既可通过设置 ADC_SglStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_IFR.SGLIF 位会自动置 1，转换结果保存在 ADC_Result 寄存器中。

通过 ADC_SglStart.Start 位启动 ADC 单次转换操作流程：

Step1: 配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5: 延时 20 μ s，等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step7: 配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8: 设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10: 配置 ADC_CR0.SGLMux，选择待转换的通道。

Step11: 设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

Step12: 设置 ADC_SglStart.Start 为 1，启动 ADC 单次转换。

Step13: 等待 ADC_IFR.SGLIF 变为 1，读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step14: 如需对其它通道进行转换，重复执行 Step10~Step13。

Step15: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 单次转换操作流程:

Step1: 配置 PAADS~PEADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1, 将 ADC 外部参考电压引脚配置为模拟端口。

注: 如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1, 使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1, 使能 ADC 模块。

Step5: 延时 20 μ s, 等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 0, 选择单次转换模式。

Step7: 设置 ADC_CR0.IE 为 1, 使能 ADC 中断。

Step8: 使能 NVIC 中断向量表中的 ADC 中断。

Step9: 配置 ADC_CR0.Ref, 选择 ADC 的参考电压。

Step10: 设置 ADC_CR0.InRefEn 为 1, 使能 ADC 内部参考电压。

注: 如果 ADC 参考电压不选择内部参考电压, 则可以略过本步骤。

Step11: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv, 设置 ADC 的转换速度。

Step12: 配置 ADC_CR0.SGLMux, 选择待转换的通道。

Step13: 设置 ADC_IFR 为 0x0, 清除 ADC 中断标志。

Step14: 配置 ADC_ExtTrigger0, 选择外部触发条件。

Step15: 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step16: 如需对其它通道进行转换, 重复执行 Step12~Step15。

Step17: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

40.5 扫描转换模式

扫描转换模式分为顺序扫描转换和插队扫描转换两种模式。两种模式各自单独工作时，均可连续对多个通道进行多次转换；当两种模式同时工作时，则优先对插队扫描配置的通道进行转换。

40.5.1 顺序扫描转换模式

顺序扫描转换模式最多可进行 16 次连续转换，转换的总次数由 ADC_SQR2.CNT 进行配置；可配置所有 22 个通道进行转换，待转换通道由 ADC_SQRx.CHxMux 进行配置。该模式既可通过设置 ADC_SqrStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。启动转换后，ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.SQRIF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_SqrResultx~ADC_SqrResult0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 8 次转换的顺序扫描转换过程。其中顺序扫描转换通道 7，4，1 配置为 AIN1，转换通道 6，3，0 配置为 AIN0，转换通道 5，2 配置为 AIN5。ADC_SqrStart.Start 置 1 后，ADC 模块会依次对顺序扫描转换通道 7~0 进行转换。



图 40-3 ADC 顺序扫描转换过程示例

通过 ADC_SqrStart.Start 位启动 ADC 顺序扫描转换操作流程：

Step1: 配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5: 延时 20 μ s，等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7: 配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8: 设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10: 配置 ADC_SQRx.CHxMux，选择顺序扫描转换通道。

Step11: 配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数。

注：总转换次数需与 Step10 中配置的转换通道数保持一致。

Step12: 设置 ADC_ICR.SQRIC 为 0，清除 ADC_IFR.SQRIF 标志。

Step13: 设置 ADC_SqrStart.Start 为 1，启动 ADC 顺序扫描转换。

Step14: 等待 ADC_IFR.SQRIF 变为 1，读取 ADC_SqrResultx ~ ADC_SqrResult0 寄存器以获取相应通道的转换结果。

Step15: 如需对其它通道进行转换，重复执行 Step10~Step14。

Step16: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 顺序扫描转换操作流程：

Step1: 配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5: 延时 20 μ s, 等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 1, 选择扫描转换模式。

Step7: 设置 ADC_CR0.IE 为 1, 使能 ADC 中断。

Step8: 使能 NVIC 中断向量表中的 ADC 中断。

Step9: 配置 ADC_CR0.Ref, 选择 ADC 的参考电压。

Step10: 设置 ADC_CR0.InRefEn 为 1, 使能 ADC 内部参考电压。

注: 如果 ADC 参考电压不选择内部参考电压, 则可以略过本步骤。

Step11: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv, 设置 ADC 的转换速度。

Step12: 配置 ADC_SQRx.CHxMux, 选择顺序扫描转换通道。

Step13: 配置 ADC_SQR2.CNT, 选择顺序扫描转换的总转换次数。

注: 总转换次数需与 Step12 中配置的转换通道数保持一致。

Step14: 设置 ADC_IFR 为 0x0, 清除 ADC 中断标志。

Step15: 配置 ADC_ExtTrigger0, 选择外部触发条件。

Step16: 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_SqrResultx ~ ADC_SqrResult0 寄存器以获取相应通道的转换结果。

Step17: 如需对其它通道进行转换, 重复执行 Step12~Step16。

Step18: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

40.5.2 插队扫描转换模式

插队扫描转换模式最多可进行 4 次连续转换, 转换的总次数由 ADC_JQR.CNT 进行配置; 可配置所有 22 个通道进行转换, 待转换通道由 ADC_JQR.CHxMux 进行配置。该模式既可通过设置 ADC_JqrStart.Start 位启动也可通过设置 ADC_ExtTrigger1 的外部触发启动。启动转换后, ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后, ADC_IFR.JQRIF 位会自动置 1, 转换结果保存在转换通道所对应的 ADC_JqrResultx~ ADC_JqrResult0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 4 次转换的插队扫描转换的过程。其中插队扫描转换通道 3, 2, 1, 0 分别设置为 AIN5, AIN0, AIN1, AIN5。ADC_JqrStart.Start

置 1 后，ADC 模块会依次对插队扫描转换通道 3~0 进行转换。

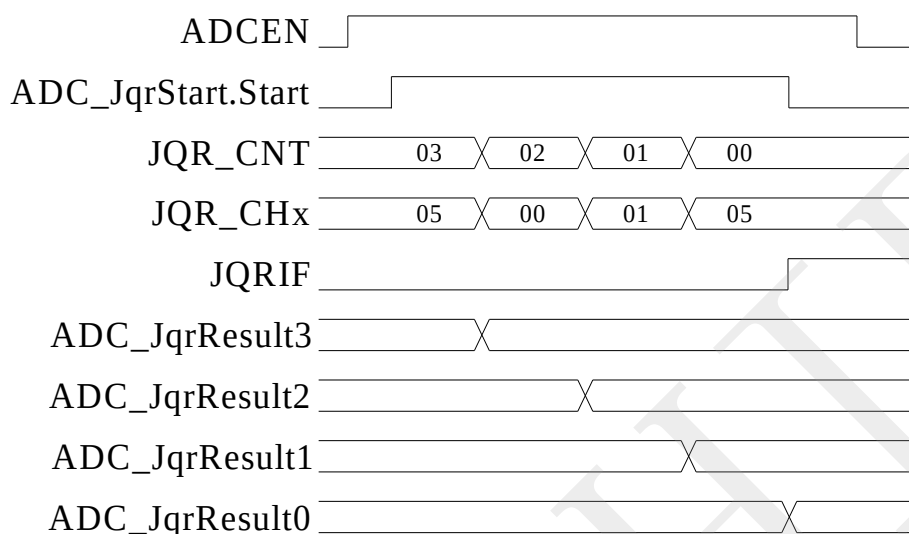


图 40-4 ADC 插队扫描转换过程示例

通过 **ADC_JqrStart.Start** 位启动 ADC 插队扫描转换操作流程：

Step1: 配置 PAADS~PEADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step5: 延时 20μs，等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 1，选择扫描转换模式。

Step7: 配置 ADC_CR0.Ref，选择 ADC 的参考电压。

Step8: 设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

注：如果 ADC 参考电压不选择内部参考电压，则可以略过本步骤。

Step9: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step10: 配置 ADC_JQR.CHxMux，选择插队扫描转换通道。

Step11: 配置 ADC_JQR.CNT，选择插队扫描转换的总转换次数。

注：总转换次数需与 Step10 中配置的转换通道数保持一致。

Step12: 设置 ADC_ICR.JQRIC 为 0, 清除 ADC_IFR.JQRIF 标志。

Step13: 设置 ADC_JqrStart.Start 为 1, 启动 ADC 插队扫描转换。

Step14: 等待 ADC_IFR.JQRIF 变为 1, 读取 ADC_JqrResultx~ ADC_JqrResult0 寄存器以获取相应通道的转换结果。

Step15: 如需对其它通道进行转换, 重复执行 Step10~Step14。

Step16: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 插队扫描转换操作流程:

Step1: 配置 PAADS~PEADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1, 将 ADC 外部参考电压引脚配置为模拟端口。

注: 如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1, 使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1, 使能 ADC 模块。

Step5: 延时 20μs, 等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 1, 选择扫描转换模式。

Step7: 设置 ADC_CR0.IE 为 1, 使能 ADC 中断。

Step8: 使能 NVIC 中断向量表中的 ADC 中断。

Step9: 配置 ADC_CR0.Ref, 选择 ADC 的参考电压。

Step10: 设置 ADC_CR0.InRefEn 为 1, 使能 ADC 内部参考电压。

注: 如果 ADC 参考电压不选择内部参考电压, 则可以略过本步骤。

Step11: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv, 设置 ADC 的转换速度。

Step12: 配置 ADC_JQR.CHxMux, 选择插队扫描转换通道。

Step13: 配置 ADC_JQR.CNT, 选择插队扫描转换的总转换次数。

注: 总转换次数需与 Step12 中配置的转换通道数保持一致。

Step14: 设置 ADC_IFR 为 0x0, 清除 ADC 中断标志。

Step15: 配置 ADC_ExtTrigger1, 选择外部触发条件。

Step16: 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_JqrResultx~ ADC_JqrResult0 寄存器以获取

相应通道的转换结果。

Step17: 如需对其它通道进行转换, 重复执行 Step12~Step16。

Step18: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

插队扫描转换的执行优先级高于顺序扫描转换, 当顺序扫描转换正在进行时, 如果插队扫描转换启动, 则顺序扫描在完成当前通道的转换后暂停, 等到插队扫描转换完成后, 再继续执行剩下的通道转换。下图演示了对 AIN0, AIN1, AIN5, AIN8 进行顺序扫描转换时, 启动对 AIN2, AIN6 进行插队扫描转换的过程。其中顺序扫描转换通道 3, 2, 1, 0 分别设置为 AIN1, AIN0, AIN5, AIN8, 插队扫描转换通道 1, 0 分别设置为 AIN6, AIN2。在顺序扫描进行 AIN0 转换的过程中, 启动了插队扫描, 顺序扫描会将当前 AIN0 的转换完成然后暂停, 等到插队扫描完成 AIN6 和 AIN2 的转换后, 顺序扫描再进行 AIN5 和 AIN8 的转换。

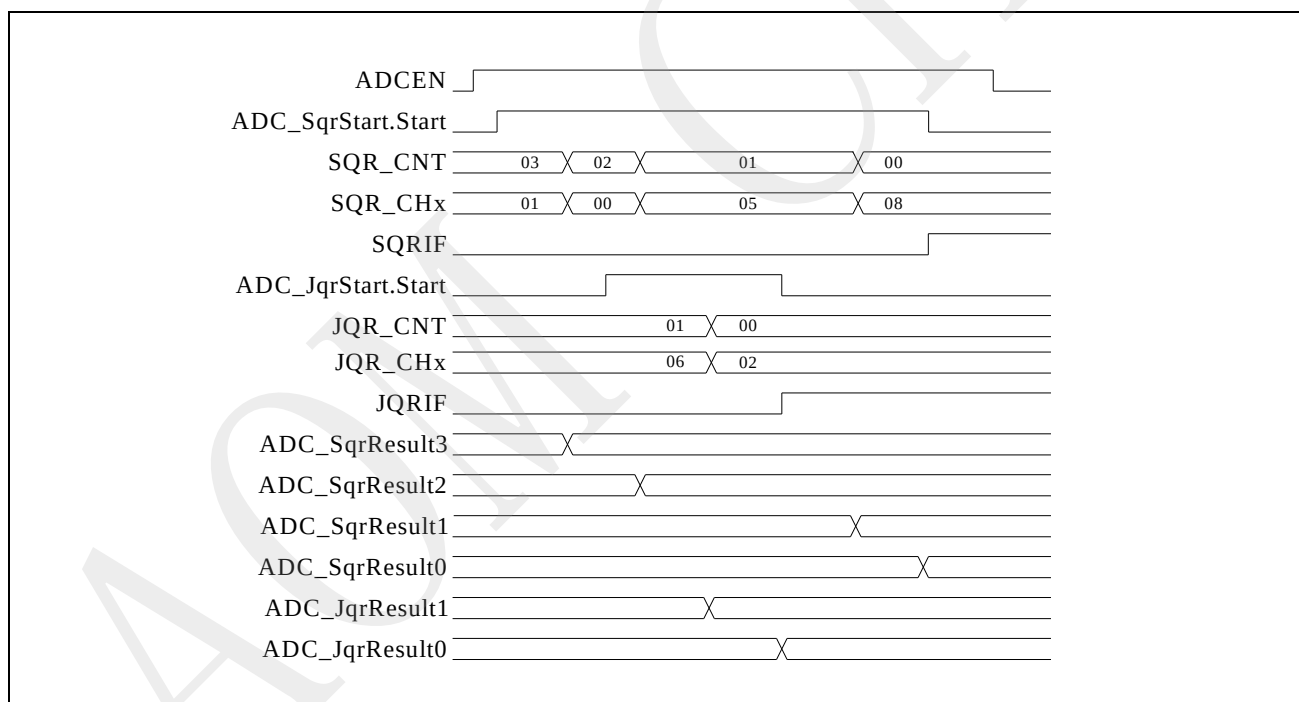


图 40-5 ADC 顺序扫描过程中进行插队扫描示例

40.5.3 扫描转换触发 DMA 读取

顺序扫描和插队扫描转换完成后,可自动触发 DMA 读取转换结果。顺序扫描模式下,通过配置 ADC_CR1.DmaSqr 为 1 使能该功能;插队扫描模式下,通过配置 ADC_CR1.DmaJqr 为 1 使能该功能。

40.6 连续转换累加模式

在连续转换累加模式下,启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加;可配置所有 22 个通道进行转换。ADC 转换的总次数由 ADC_SQR2.CNT 进行配置;待转换通道由 ADC_SQRx.CHxMux 进行配置。该模式既可通过设置 ADC_SqrStart.Start 位启动也可通过设置 ADC_ExtTrigger0 的外部触发启动。启动连续转换后,ADC 模块依次转换 CHxMux~CH0Mux 中配置的通道直到总转换次数完成。ADC 模块完成总转换次数后,ADC_IFR.SQRIF 位会自动置 1,转换结果的累加值保存在 ADC_ResultAcc 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。顺序扫描转换通道 9, 6, 3, 0 配置为 AIN0, 转换通道 8, 5, 2 配置为 AIN1, 转换通道 7, 4, 1 配置为 AIN5。ADC_SqrStart.Start 置 1 后,ADC 模块会按照顺序扫描转换通道配置依次进行转换,直到 SQR_CNT 的计数值变为 0。每次转换完成时,ADC_ResultAcc 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

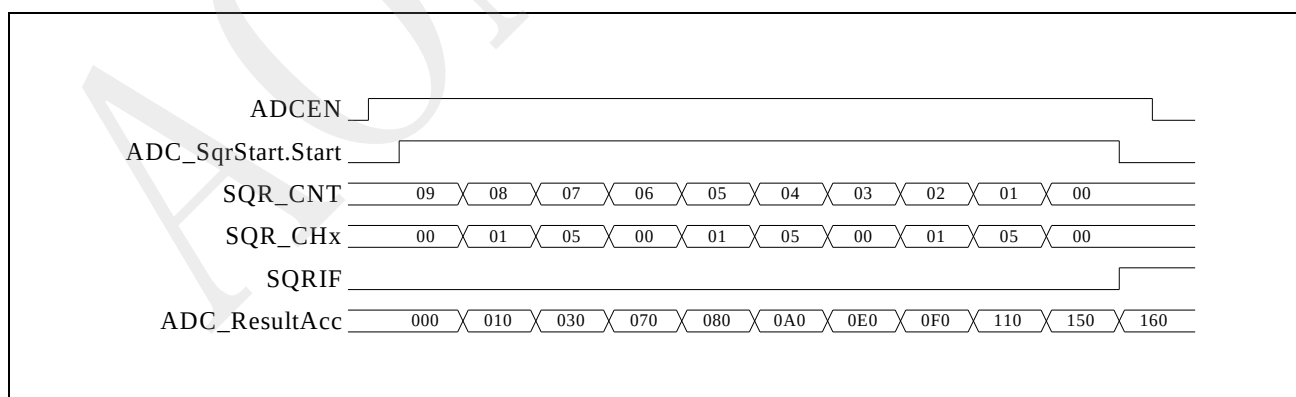


图 40-6 ADC 连续转换累加过程示例

通过 ADC_SqrStart.Start 位启动 ADC 连续转换累加操作流程:

Step1: 配置 PAADS~PEADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1, 将 ADC 外部参考电压引脚配置为模拟端口。

注: 如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1, 使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1, 使能 ADC 模块。

Step5: 延时 20 μ s, 等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 1, 选择扫描转换模式。

Step7: 设置 ADC_CR1.RAccEn 为 1, 使能 ADC 转换自动累加功能。

Step8: 配置 ADC_CR0.Ref, 选择 ADC 的参考电压。

Step9: 设置 ADC_CR0.InRefEn 为 1, 使能 ADC 内部参考电压。

注: 如果 ADC 参考电压不选择内部参考电压, 则可以略过本步骤。

Step10: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv, 设置 ADC 的转换速度。

Step11: 配置 ADC_SQRx.CHxMux, 选择顺序扫描转换通道。

Step12: 配置 ADC_SQR2.CNT, 选择顺序扫描转换的总转换次数。

注: 总转换次数需与 Step11 中配置的转换通道数保持一致。

Step13: 设置 ADC_ICR.SQRIC 为 0, 清除 ADC_IFR.SQRIF 标志。

Step14: 设置 ADC_CR1.RAccClr 为 0, 清零 ADC_ResultAcc 寄存器。

Step15: 设置 ADC_SqrStart.Start 为 1, 启动 ADC 顺序扫描转换。

Step16: 等待 ADC_IFR.SQRIF 变为 1, 读取 ADC_ResultAcc 寄存器以获取转换结果累加值。

Step17: 如需对其它通道进行转换, 重复执行 Step11~Step16。

Step18: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 连续转换累加操作流程:

Step1: 配置 PAADS~PEADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 PBADS.bit1 为 1, 将 ADC 外部参考电压引脚配置为模拟端口。

注: 如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1, 使能 BGR 模块。

Step4: 设置 ADC_CR0.En 为 1, 使能 ADC 模块。

Step5: 延时 20 μ s, 等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.Mode 为 1, 选择扫描转换模式。

Step7: 设置 ADC_CR0.IE 为 1, 使能 ADC 中断。

Step8: 使能 NVIC 中断向量表中的 ADC 中断。

Step9: 设置 ADC_CR1.RAccEn 为 1, 使能 ADC 转换自动累加功能。

Step10: 配置 ADC_CR0.Ref, 选择 ADC 的参考电压。

Step11: 设置 ADC_CR0.InRefEn 为 1, 使能 ADC 内部参考电压。

注: 如果 ADC 参考电压不选择内部参考电压, 则可以略过本步骤。

Step12: 配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv, 设置 ADC 的转换速度。

Step13: 配置 ADC_SQRx.CHxMux, 选择顺序扫描转换通道。

Step14: 配置 ADC_SQR2.CNT, 选择顺序扫描转换的总转换次数。

注: 总转换次数需与 Step13 中配置的转换通道数保持一致。

Step15: 设置 ADC_IFR 为 0x0, 清除 ADC 中断标志。

Step16: 设置 ADC_CR1.RAccClr 为 0, 清零 ADC_ResultAcc 寄存器。

Step17: 配置 ADC_ExtTrigger0, 选择外部触发条件。

Step18: 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_ResultAcc 寄存器以获取转换结果累加值。

Step19: 如需对其它通道进行转换, 重复执行 Step13~Step18。

Step20: 设置 ADC_CR0.En 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

40.7 DC 转换外部触发源

ADC 转换既可通过软件配置启动，也可通过外部触发启动。

配置 ADC_ExtTrigger0 寄存器可设置 ADC 单次转换或顺序扫描转换的外部触发源。

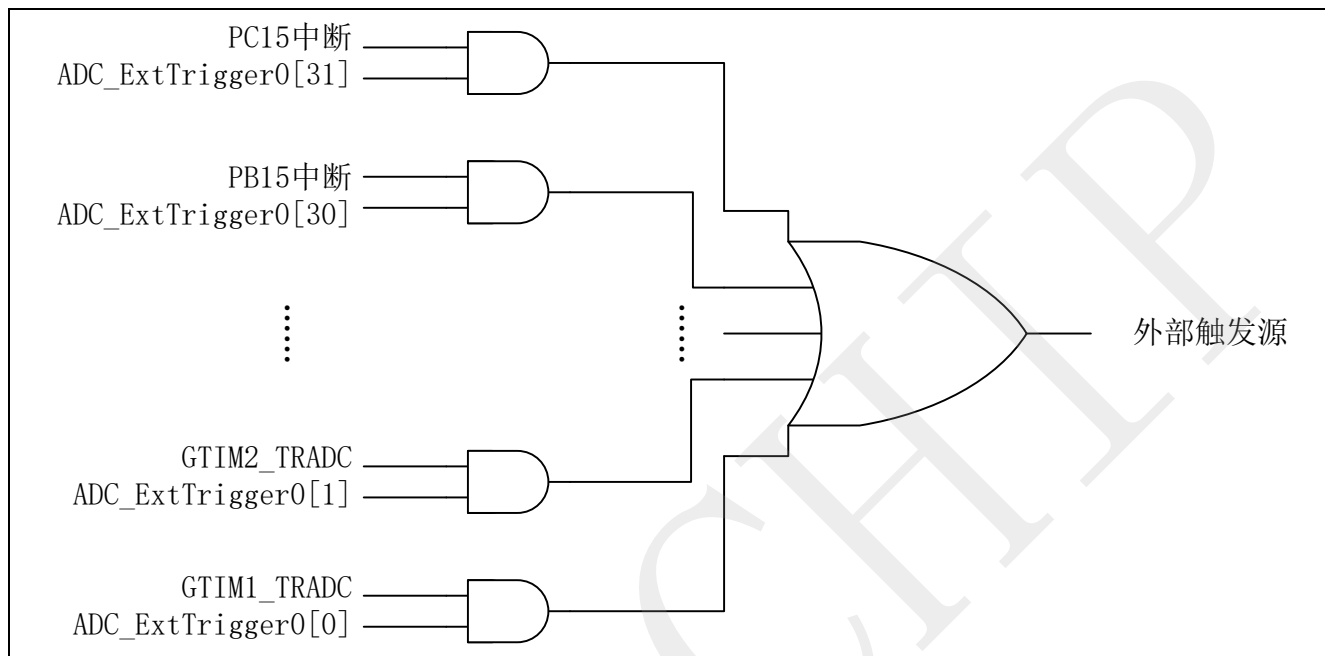


图 40-7 ADC 单次转换或顺序扫描转换外部触发源示意图

配置 ADC_ExtTrigger1 寄存器可设置 ADC 插队扫描转换的外部触发源。

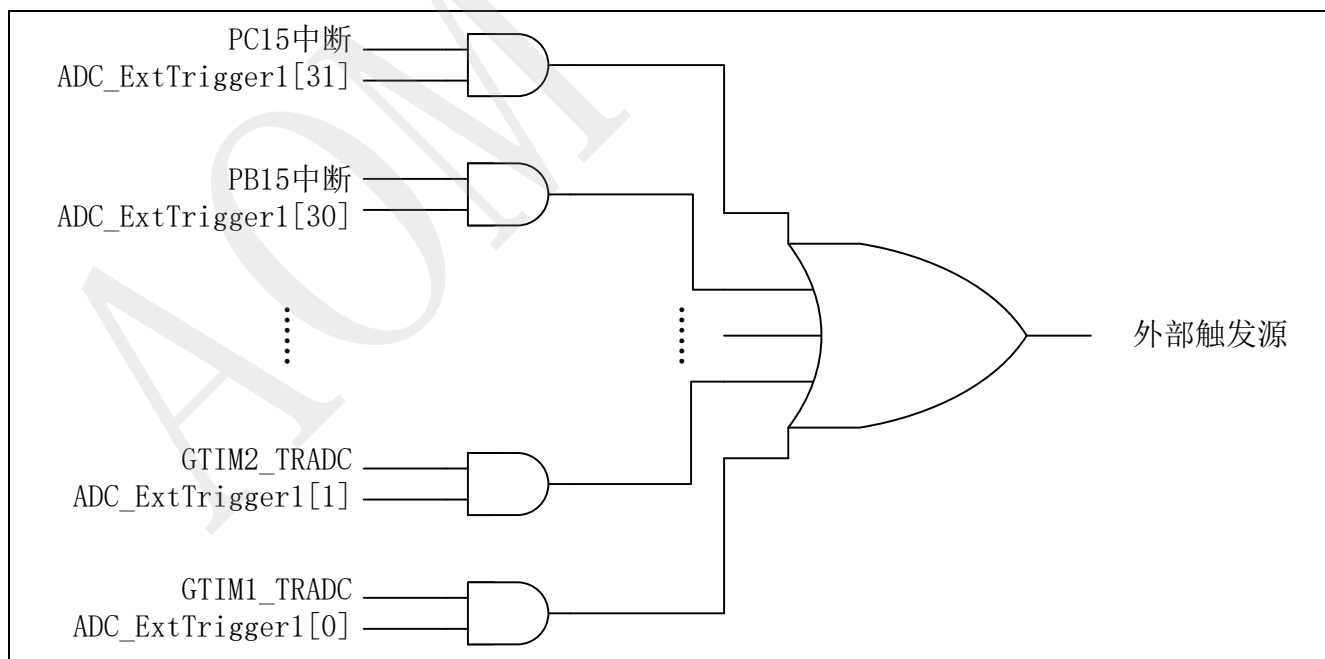


图 40-8 ADC 插队扫描转换外部触发源示意图

40.8 ADC 转换结果比较

ADC 转换完成时,ADC 转换结果可以与用户设定的阈值进行比较,支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 HtCmp、LtCmp、RegCmp 置 1。该功能可实现对模拟量的自动监测,直到 ADC 转换结果符合用户预期时才产生中断申请用户程序介入。监测通道选择通过 ADC_CR1.ThCh 进行配置。

上阈值比较:当 ADC 转换结果位于[ADC_HT, 4095]区间内则 ADC_IFR.HTIF 置 1;向 ADC_ICR.HTIC 写入 0 则清零 ADC_IFR.HTIF。

下阈值比较:当 ADC 转换结果位于[0,ADC_LT)区间内则 ADC_IFR.LTIF 置 1;向 ADC_ICR.LTIC 写入 0 则清零 ADC_IFR.LTIF。

区间值比较:当 ADC 转换结果位于[ADC_LT, ADC_HT)区间内则 ADC_IFR.REGIF 置 1;向 ADC_ICR.REGIC 写入 0 则清零 ADC_IFR.REGIF。

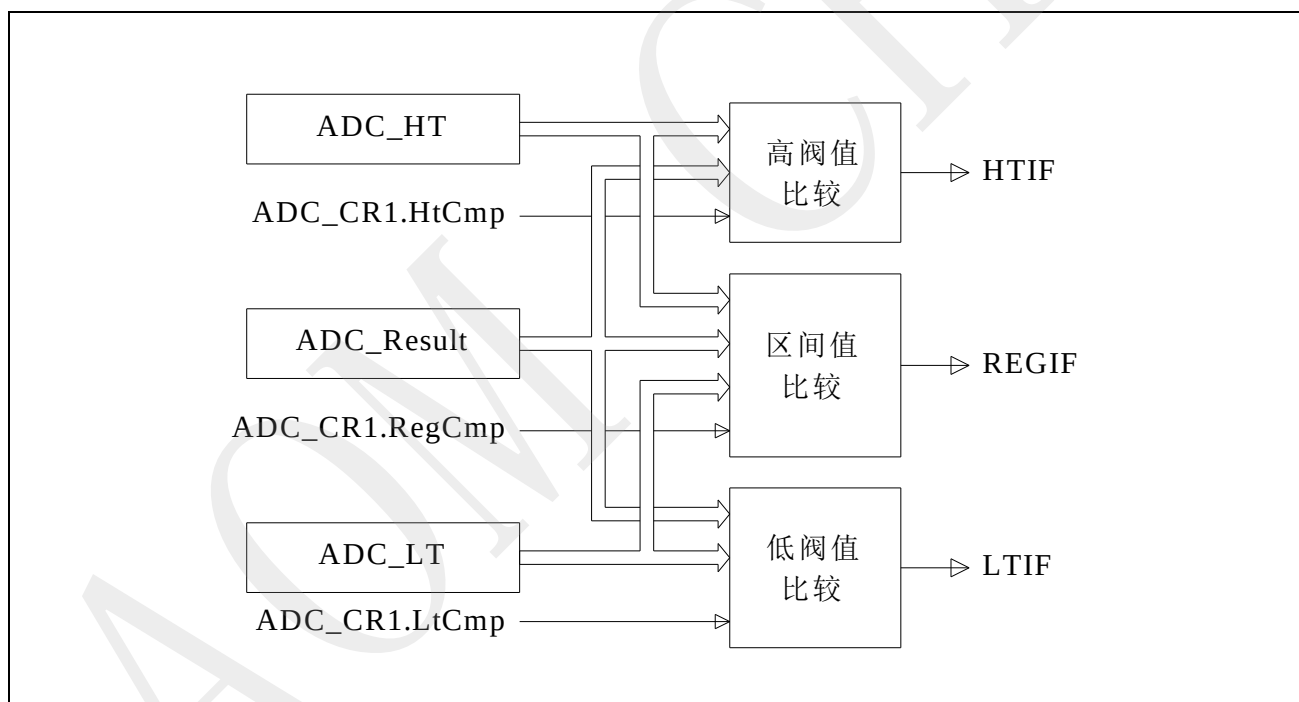


图 40-9 ADC 转换结果比较

40.9 ADC 中断

ADC 中断请求如下表所示：

中断源	中断标志	中断使能
ADC 插队扫描转换完成	ADC_IFR. JQRIF	ADC_CR0. IE
ADC 顺序扫描转换完成	ADC_IFR. SQRIF	
ADC 转换结果位于区间值区域 (大于等于下阈值小于上阈值)	ADC_IFR. REGIF	
ADC 转换结果位于上阈值区域 (大于等于上阈值)	ADC_IFR. HTIF	
ADC 转换结果比较下阈值区域 (小于下阈值)	ADC_IFR. LTIF	
ADC 单次转换完成	ADC_IFR. SGLIF	

表 40-1 ADC 中断请求

40.10 使用温度传感器测量环境温度

温度传感器的输出电压会随环境温度的改变而变化，故根据温度传感器的输出电压即可计算出相应的环境温度。当 ADC 模块的测量通道选择温度传感器的输出电压时，即可测量环境温度。

计算公式如下：

$$\text{环境温度} = 25 + 0.0839 \times V_{\text{ref}} \times (\text{AdcValue} - \text{Trim})$$

其中：Vref 为当前 ADC 模块的参考电压，取值为 1.5 或 2.5。

AdcValue 为 ADC 模块测量温度传感器输出电压的结果，取值为 0~4095。

Trim 为 16Bits 的校准值，计算时需要从 Flash 存储器中读出，其存放地址详见下表。

ADC 参考电压	校准值存放地址	校准值精度
内部 1.5V	0x00100A12	±3°C
内部 2.5V	0x00100A14	±3°C

计算示例如下：

条件 1：Vref=2.5、AdcValue=0x7E5、Trim=0x76C：

温度 1： $25 + 0.0839 \times 2.5 \times (0x7E5 - 0x76C) = 50^{\circ}\text{C}$ 。

条件 2：Vref=1.5、AdcValue=0x72D、Trim=0x76C：

温度 3： $25 + 0.0839 \times 1.5 \times (0x72D - 0x76C) = 17^{\circ}\text{C}$ 。

通过 ADC 测量环境温度操作流程：

Step1：设置 BGR_CR 为 3，使能 BGR 模块和温度传感器模块。

Step2：设置 ADC_CR0.En 为 1，使能 ADC 模块。

Step3：延时 20 μs ，等待 ADC 及 BGR 模块启动完成。

Step4：设置 ADC_CR1.Mode 为 0，选择单次转换模式。

Step5：设置 ADC_CR0.InRefEn 为 1，使能 ADC 内部参考电压。

Step6：配置 ADC_CR0.Ref，选择 ADC 的参考电压为内部 1.5V 或内部 2.5V。

Step7：配置 ADC_CR0.SAM 及 ADC_CR0.CkDiv，设置 ADC 的转换速度。

Step8：设置 ADC_CR0.SGLMux 为 0x14，选择待转换的通道为温度传感器的输出。

Step9：设置 ADC_CR0.Buf 为 1，使能输入信号放大器。

Step10：设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

Step11：设置 ADC_SglStart.Start 为 1，启动 ADC 单次转换。

Step12：等待 ADC_IFR.SGLIF 变为 1，读取 ADC_Result 寄存器以获取 ADC 转换结果。

Step13：设置 ADC_CR0.En 及 BGR_CR 为 0，关闭 ADC 模块、BGR 模块、温度传感器模块。

Step14：读取温度传感器校准值，根据公式计算当前的环境温度。

40.11 ADC 模块寄存器

基地址：0x40012400

寄存器	偏移地址	描述
ADC_CR0	0x000	ADC 基本配置寄存器0
ADC_CR1	0x004	ADC 基本配置寄存器1
ADC_SQR0	0x010	ADC 顺序扫描转换通道配置寄存器0
ADC_SQR1	0x014	ADC 顺序扫描转换通道配置寄存器1
ADC_SQR2	0x018	ADC 顺序扫描转换通道配置寄存器2
ADC_JQR	0x01C	ADC 插队扫描转换通道配置寄存器
ADC_SqrResult0	0x020	ADC 顺序扫描转换通道0转换结果
ADC_SqrResult1	0x024	ADC 顺序扫描转换通道1转换结果
ADC_SqrResult2	0x028	ADC 顺序扫描转换通道2转换结果
ADC_SqrResult3	0x02C	ADC 顺序扫描转换通道3转换结果
ADC_SqrResult4	0x030	ADC 顺序扫描转换通道4转换结果
ADC_SqrResult5	0x034	ADC 顺序扫描转换通道5转换结果
ADC_SqrResult6	0x038	ADC 顺序扫描转换通道6转换结果
ADC_SqrResult7	0x03C	ADC 顺序扫描转换通道7转换结果
ADC_SqrResult8	0x040	ADC 顺序扫描转换通道8转换结果
ADC_SqrResult9	0x044	ADC 顺序扫描转换通道9转换结果
ADC_SqrResult10	0x048	ADC 顺序扫描转换通道10转换结果
ADC_SqrResult11	0x04C	ADC 顺序扫描转换通道11转换结果
ADC_SqrResult12	0x050	ADC 顺序扫描转换通道12转换结果
ADC_SqrResult13	0x054	ADC 顺序扫描转换通道13转换结果
ADC_SqrResult14	0x058	ADC 顺序扫描转换通道14转换结果
ADC_SqrResult15	0x05C	ADC 顺序扫描转换通道15转换结果
ADC_JqrResult0	0x060	ADC 插队扫描转换通道0转换结果
ADC_JqrResult1	0x064	ADC 插队扫描转换通道1转换结果
ADC_JqrResult2	0x068	ADC 插队扫描转换通道2转换结果
ADC_JqrResult3	0x06C	ADC 插队扫描转换通道3转换结果
ADC_Result	0x070	ADC 转换结果
ADC_ResultAcc	0x074	ADC 转换结果累加值
ADC_HT	0x078	ADC 比较上阈值
ADC_LT	0x07C	ADC 比较下阈值
ADC_IFR	0x080	ADC 中断标志寄存器
ADC_ICR	0x084	ADC 中断清除寄存器
ADC_ExtTrigger0	0x088	ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器
ADC_ExtTrigger1	0x08C	ADC 插队扫描转换外部中断触发源配置寄存器
ADC_SglStart	0x090	ADC单次转换启动控制寄存器

ADC_SqrStart	0x094	ADC顺序扫描转换启动控制寄存器
ADC_JqrStart	0x098	ADC插队扫描转换启动控制寄存器
ADC_allstart	0x09C	ADC一直转换启动停止控制寄存器
BGR_CR	0x0B0	BGR控制寄存器

表 40-2 ADC 寄存器

40.11.1 ADC 基本配置寄存器 0 (ADC_CR0)

偏移地址: 0x000

复位值: 0x0000 67F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
IE	InRefEn	SAM			Buf	Ref		SGLMux							
RW	RW	RW			RW	RW		RW							
7	6	5	4	3	2	1	0								
SGLMux					CkDiv			RSV	En						
RW					RW				RW						

位	标记	功能描述
31:16	Reserved	保留
15	IE	ADC中断控制 1: 使能中断 0: 禁止中断
14	InRefEn	ADC内部参考电压使能 1: 使能内部参考电压 0: 禁止内部参考电压
13:12	SAM	ADC采样周期选择 00: 4个转换周期 01: 6个转换周期 10: 8个转换周期 11: 12个转换周期
11	Buf	ADC输入信号放大器控制 0: 关闭放大器, 外部输入信号与ADC直接相连。 1: 打开放大器, 外部输入信号通过放大器放大后与ADC相连, 用于高阻信号。以下几种情况需要打开BUF 功能, 使用BUF功能时, 最大速率200k bps 1) 外部驱动很弱信号 2) 测量1/3AVCC 3) 测量temp sensor 4) 测量VREF1P2
10:9	Ref	ADC参考电压选择 00: 内部1.5V 01: 内部2.5V

		10: 外部参考电压ExRef (VREFP / VREFN) 11: AVCC电压
8:4	SGLMux	单次转换模式转换通道选择 00000: AIN0 (PA00) 00001: AIN1 (PA01) 00010: AIN2 (PA02) 00011: AIN3 (PA03) 00100: AIN4 (PA04) 00101: AIN5 (PA05) 00110: AIN6 (PA06) 00111: AIN7 (PA07) 01000: AIN8 (PB00) 01001: AIN9 (PB01) 01010: AIN10 (PC00) 01011: AIN11 (PC01) 01100: AIN12 (PC02) 01101: AIN13 (PC03) 01110: AIN14 (PC04) 01111: AIN15 (PC05) 10000: AIN16 (PB12) 10001: DAC1输出 注: ADC_CR0.Buf 必须为1 10010: DAC2输出 注: ADC_CR0.Buf 必须为1 10011: 1/3VCC 注: ADC_CR0.Buf 必须为1 10100: 内置温度传感器输出电压 注: ADC_CR0.Buf 必须为1 10101: 内部基准1.2V输出电压 注: ADC_CR0.Buf 必须为1
3:2	CkDiv	ADC时钟选择 00: PCLK时钟 01: PCLK时钟2分频 10: PCLK时钟4分频 11: PCLK时钟8分频
1	Reserved	保留
0	En	ADC使能控制 1: 使能ADC 0: 禁止ADC

40.11.2 ADC 基本配置寄存器 1 (ADC_CR1)

偏移地址：0x004

复位值：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15		14		13		12		11		10		9		8	
RAccClr		RegCmp		HtCmp		LtCmp		RAccEn		Mode		DmaJqr		DmaSqr	
WO		RW		RW		RW		RW		RW		RW		RW	
7		6		5		4		3		2		1		0	
ThCh										Align		Reserved			
RW										RW					

位	标记	功能描述
31:16	Reserved	保留
15	RAccClr	ADC转换结果累加寄存器清零 1: 无作用 0: ADC转换结果累加寄存器 (ADC_ResultAcc) 清零
14	RegCmp	ADC区间比较控制 1: 使能区间比较 0: 禁止区间比较
13	HtCmp	ADC高阈值比较控制 1: 使能高阈值比较 0: 禁止高阈值比较
12	LtCmp	ADC低阈值比较控制 1: 使能低阈值比较 0: 禁止低阈值比较
11	RAccEn	ADC转换结果自动累加控制 1: 使能ADC转换结果自动累加功能 0: 禁止ADC转换结果自动累加功能
10	Mode	ADC转换模式选择 1: 扫描转换模式 0: 单次转换模式
9	DmaJqr	插队扫描触发DMA读取控制 1: 使能插队扫描转换触发DMA读取 0: 禁止插队扫描转换触发DMA读取
8	DmaSqr	顺序扫描触发DMA读取控制

		1: 使能顺序扫描转换触发DMA读取 0: 禁止顺序扫描转换触发DMA读取
7:3	ThCh	阈值比较通道选择 00000: 选择通道0进行阈值比较 00001: 选择通道1进行阈值比较 00010: 选择通道2进行阈值比较 10101: 选择通道 21 进行阈值比较
2	Align	转换结果对齐控制 1: 转换结果16Bits左对齐存储 0: 转换结果16Bits右对齐存储
1:0	Reserved	保留

40.11.3 ADC 顺序扫描转换通道配置寄存器 0 (ADC_SQR0)

偏移地址: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CH5Mux					CH4Mux					CH3Mux				
	RW					RW					RW				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3Mux	CH2Mux					CH1Mux					CH0Mux				
RW	RW					RW					RW				

位	标记	功能描述
31:30	Reserved	保留
29:25	CH5Mux	顺序扫描转换通道5选择, 设置参见ADC_CR0. SGLMux
24:20	CH4Mux	顺序扫描转换通道4选择, 设置参见ADC_CR0. SGLMux
19:15	CH3Mux	顺序扫描转换通道3选择, 设置参见ADC_CR0. SGLMux
14:10	CH2Mux	顺序扫描转换通道2选择, 设置参见ADC_CR0. SGLMux
9:5	CH1Mux	顺序扫描转换通道1选择, 设置参见ADC_CR0. SGLMux
4:0	CH0Mux	顺序扫描转换通道0选择, 设置参见ADC_CR0. SGLMux

40.11.4 ADC 顺序扫描转换通道配置寄存器 1 (ADC_SQR1)

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CH11Mux				CH10Mux				CH9Mux			
				RW				RW				RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH9Mux		CH8Mux				CH7Mux				CH6Mux					
RW		RW				RW				RW					

位	标记	功能描述
31:30	Reserved	保留
29:25	CH11Mux	顺序扫描转换通道11选择, 设置参见ADC_CR0. SGLMux
24:20	CH10Mux	顺序扫描转换通道10选择, 设置参见ADC_CR0. SGLMux
19:15	CH9Mux	顺序扫描转换通道9选择, 设置参见ADC_CR0. SGLMux
14:10	CH8Mux	顺序扫描转换通道8选择, 设置参见ADC_CR0. SGLMux
9:5	CH7Mux	顺序扫描转换通道7选择, 设置参见ADC_CR0. SGLMux
4:0	CH6Mux	顺序扫描转换通道6选择, 设置参见ADC_CR0. SGLMux

40.11.5 ADC 顺序扫描转换通道配置寄存器 2 (ADC_SQR2)

偏移地址: 0x018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CNT				CH15Mux			
								RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH15Mux	CH14Mux				CH13Mux				CH12Mux						
RW	RW				RW				RW				RW		

位	标记	功能描述
31:24	Reserved	保留
23:20	CNT	顺序扫描转换次数
19:15	CH15Mux	顺序扫描转换通道15选择, 设置参见ADC_CR0. SGLMux
14:10	CH14Mux	顺序扫描转换通道14选择, 设置参见ADC_CR0. SGLMux
9:5	CH13Mux	顺序扫描转换通道13选择, 设置参见ADC_CR0. SGLMux
4:0	CH12Mux	顺序扫描转换通道12选择, 设置参见ADC_CR0. SGLMux

40.11.6 ADC 插队扫描转换通道配置寄存器 (ADC_JQR)

偏移地址: 0x01C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CNT		CH3Mux			
										RW		RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3Mux		CH2Mux				CH1Mux				CH0Mux					
RW		RW				RW				RW					

位	标记	功能描述
31:24	Reserved	保留
21:20	CNT	插队扫描转换次数
19:15	CH3Mux	插队扫描转换通道3选择, 设置参见ADC_CR0. SGLMux
14:10	CH2Mux	插队扫描转换通道2选择, 设置参见ADC_CR0. SGLMux
9:5	CH1Mux	插队扫描转换通道1选择, 设置参见ADC_CR0. SGLMux
4:0	CH0Mux	插队扫描转换通道0选择, 设置参见ADC_CR0. SGLMux

40.11.7 ADC 顺序扫描转换通道 x 转换结果 (ADC_SqrResult0 - 15)

偏移地址: 0x020 ~ 0x5C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC 顺序扫描转换通道x转换结果

40.11.8 ADC 插队扫描转换通道 x 转换结果 (ADC_JqrResult0 - 3)

偏移地址: 0x060 ~ 0x6C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC 插队扫描转换通道x转换结果

40.11.9 ADC 转换结果（ADC_Result）

偏移地址：0x070

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	Result	ADC转换结果

40.11.10 ADC 转换结果累加值 (ADC_ResultAcc)

偏移地址: 0x074

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ResultAcc[19:16]			
												RO			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ResultAcc[15:0]															
RO															

位	标记	功能描述
31:20	Reserved	保留
19:0	ResultAcc	ADC转换累加值

40.11.11 ADC 比较上阈值 (ADC_HT)

偏移地址: 0x078

复位值: 0x0000 0FFF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HT											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	HT	ADC转换结果比较上阈值

40.11.12 ADC 比较下阈值 (ADC_LT)

偏移地址: 0x07C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	b	0
Reserved				LT											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	LT	ADC转换结果比较下阈值

40.11.13 ADC 中断标志寄存器 (ADC_IFR)

偏移地址: 0x080

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	
Reserved																								
7				6			5			4			3			2			1			0		
Reserved						JQRIF			SQRIF			REGIF			HTIF			LTIF			SGLIF			
						RO			RO			RO			RO			RO			RO			

位	标记	功能描述
31:6	Reserved	保留
5	JQRIF	ADC插队扫描转换完成标志 1: ADC插队扫描转换完成 0: ADC插队扫描转换未完成
4	SQRIF	ADC顺序扫描转换完成标志 1: ADC顺序扫描转换完成 0: ADC顺序扫描转换未完成
3	REGIF	ADC转换结果比较区间标志 1: ADC转换结果位于[ADC_LT , ADC_HT) 区间内 0: ADC转换结果位于[ADC_LT , ADC_HT) 区间外
2	HTIF	ADC转换结果比较上阈值标志 1: ADC转换结果位于[ADC_HT, 4095] 区间内

		0: ADC转换结果位于[ADC_HT, 4095]区间外
1	LTIF	ADC转换结果比较下阈值标志 1: ADC转换结果位于[0, ADC_LT)区间内 0: ADC转换结果位于[0, ADC_LT)区间外
0	SGLIF	ADC单次转换完成标志 1: ADC单次转换完成 0: ADC单次转换未完成

40.11.14 ADC 中断清除寄存器 (ADC_ICR)

偏移地址: 0x084

复位值: 0x0000 003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7		6		5		4		3		2		1		0									
Reserved				JQRIC		SQRIC		REGIC		HTIC		LTIC		SGLIC									
				R1W0		R1W0		R1W0		R1W0		R1W0		R1W0									

位	标记	功能描述
31:6	Reserved	保留
5	JQRIC	ADC插队扫描转换完成标志; 写0清除, 写1无作用。
4	SQRIC	ADC顺序扫描转换完成标志; 写0清除, 写1无作用。
3	REGIC	ADC转换结果比较区间标志; 写0清除, 写1无作用。
2	HTIC	ADC转换结果比较上阈值标志; 写0清除, 写1无作用。
1	LTIC	ADC转换结果比较下阈值标志; 写0清除, 写1无作用。
0	SGLIC	ADC单次转换完成标志; 写0清除, 写1无作用。

40.11.15 ADC 单次转换或顺序扫描转换外部中断触发源配置寄存器 (ADC_ExtTrigger0)

偏移地址：0x088

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15	PB15	PA15	PC11	PB11	PA11	PD07	PC07	PB07	PA07	PD03	PC03	PB03	PA03	DMA	SPI2
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8
SPI1	PCA	RTC	VC2	VC1	LPUART2	LPUART1	UART2
RW	RW	RW	RW	RW	RW	RW	RW

7	6	5	4	3	2	1	0
UART1	STIM	STIM	BTIM456	ATIM	GTIM3	GTIM2	GTIM1
RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	PC15	PC15中断触发ADC转换
30	PB15	PB15中断触发ADC转换
29	PA15	PA15中断触发ADC转换
28	PC11	PC11中断触发ADC转换
27	PB11	PB11中断触发ADC转换
26	PA11	PA11中断触发ADC转换
25	PD07	PD07中断触发ADC转换
24	PC07	PC07中断触发ADC转换
23	PB07	PB07中断触发ADC转换
22	PA07	PA07中断触发ADC转换
21	PD03	PD03中断触发ADC转换
20	PC03	PC03中断触发ADC转换
19	PB03	PB03中断触发ADC转换
18	PA03	PA03中断触发ADC转换
17	DMA	DMA中断触发ADC转换
16	SPI2	SPI2中断触发ADC转换
15	SPI1	SPI1中断触发ADC转换
14	PCA	PCA中断触发ADC转换
13	RTC	RTC中断触发ADC转换

12	VC2	VC2中断触发ADC转换
11	VC1	VC1中断触发ADC转换
10	LPUART2	LPUART2中断触发ADC转换
9	LPUART1	LPUART1中断触发ADC转换
8	UART2	UART2中断触发ADC转换
7	UART1	UART1中断触发ADC转换
6	STIM_CC4	STIM_CC4 触发ADC转换
5	STIM_TRG02	STIM_TRG02触发ADC转换
4	BTIM456	BTIM456触发ADC转换
3	ATIM_TRADC	ATimer 触发ADC转换
2	GTIM3	GTimer3触发ADC转换
1	GTIM2	GTimer2触发ADC转换
0	GTIM1	GTimer1触发ADC转换

Notes:

- 1) 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。
- 2) 使用 GTIM 触发 ADC，如不使能 NVIC，则由硬件自动清除中断标志。

40.11.16 ADC 插队扫描转换外部中断触发源配置寄存器（ADC_ExtTrigger1）

偏移地址：0x08C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PC15	PB15	PA15	PC11	PB11	PA11	PD07	PC07	PB07	PA07	PD03	PC03	PB03	PA03	DMA	SPI2
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8
SPI1	PCA	RTC	VC2	VC1	LPUART2	LPUART1	UART2
RW	RW	RW	RW	RW	RW	RW	RW

7	6	5	4	3	2	1	0
UART1	STIM	STIM	BTIM456	ATIM	GTIM3	GTIM2	GTIM1
RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31	PC15	PC15中断触发ADC转换
30	PB15	PB15中断触发ADC转换

29	PA15	PA15中断触发ADC转换
28	PC11	PC11中断触发ADC转换
27	PB11	PB11中断触发ADC转换
26	PA11	PA11中断触发ADC转换
25	PD07	PD07中断触发ADC转换
24	PC07	PC07中断触发ADC转换
23	PB07	PB07中断触发ADC转换
22	PA07	PA07中断触发ADC转换
21	PD03	PD03中断触发ADC转换
20	PC03	PC03中断触发ADC转换
19	PB03	PB03中断触发ADC转换
18	PA03	PA03中断触发ADC转换
17	DMA	DMA中断触发ADC转换
16	SPI2	SPI2中断触发ADC转换
15	SPI1	SPI1中断触发ADC转换
14	PCA	PCA中断触发ADC转换
13	RTC	RTC中断触发ADC转换
12	VC2	VC2中断触发ADC转换
11	VC1	VC1中断触发ADC转换
10	LPUART2	LPUART2中断触发ADC转换
9	LPUART1	LPUART1中断触发ADC转换
8	UART2	UART2中断触发ADC转换
7	UART1	UART1中断触发ADC转换
6	STIM_CC4	STIM_CC4 触发ADC转换
5	STIM_TRG02	STIM_TRG02触发ADC转换
4	BTIM456	BTIM456触发ADC转换
3	ATIM_TRADC	ATimer 触发ADC转换
2	GTIM3	GTimer3触发ADC转换
1	GTIM2	GTimer2触发ADC转换
0	GTIM1	GTimer1触发ADC转换

Notes:

- 1) 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。
- 2) 使用 GTIM 触发 ADC，如不使能 NVIC，则由硬件自动清除中断标志。

40.11.17ADC 单次转换启动控制寄存器（ADC_SglStart）

偏移地址：0x090

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														Start	
														RW	

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC单次转换启动控制 1: 启动ADC单次转换 0: 停止ADC单次转换

40.11.18 ADC 顺序扫描转换启动控制寄存器（ADC_SqrStart）

偏移地址：0x094

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														Start	
														RW	

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC顺序扫描转换启动控制 1: 启动ADC顺序扫描转换 0: 停止ADC顺序扫描转换

40.11.19 ADC 插队扫描转换启动控制寄存器（ADC_JqrStart）

偏移地址：0x098

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															Start
															RW

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC插队扫描转换启动控制 1: 启动ADC插队扫描转换 0: 停止ADC插队扫描转换

40.11.20 ADC 一直转换启动停止控制寄存器 (ADC_allStart)

偏移地址: 0x09C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														Start	
														RW	

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC插队扫描转换启动控制 1: 启动ADC一直转换 0: 停止ADC一直转换, 需要等待到当次转换结束 读0无转换; 1正在转换

40.11.21 BGR 配置寄存器 (BGR_CR)

偏移地址: 0x0B0

复位值 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TS_EN	BGR_EN
														RW	RW

位	标记	功能描述
31:2	Reserved	保留
1	TS_EN	内置温度传感器使能控制 1: 使能内部温度传感器, 必须使能BGR 0: 禁止内部温度传感器 注意: 温度传感器使能 20μ s 后, 才能输出稳定信号。
0	BGR_EN	BGR使能控制 1: 使能 BGR 0: 禁止 BGR 注意: 1) PERIO_CLKEN.ADC为1时, 才可以操作此寄存器。 2) BGR使能 20μ s 后, 才能输出稳定的高精度参考电压。BGR稳定后才可以被其它模块使用, 故用户操作中应加入等待BGR稳定的步骤。 3) 当使用ADC, VC, OPA时, 必须使能BGR

41. 数模转换器（DAC）

DAC 模块是 12 位电压输出数模转换器。DAC 可以按 8 位或 12 位模式进行配置，并且可与 DMA 控制器配合使用。在 12 位模式下，数据可以采用左对齐或右对齐。可使用输入参考电压 VREF+（与 ADC 共享）。可以有选择地缓冲输出以实现更高的电流驱动。

41.1 DAC 特性

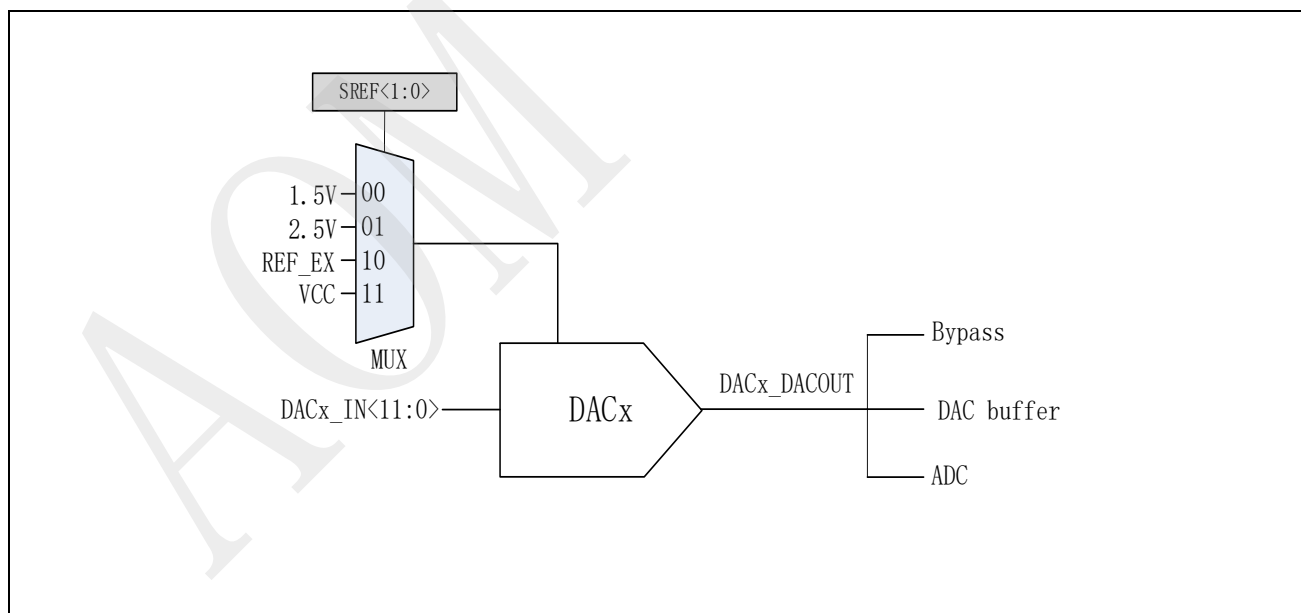
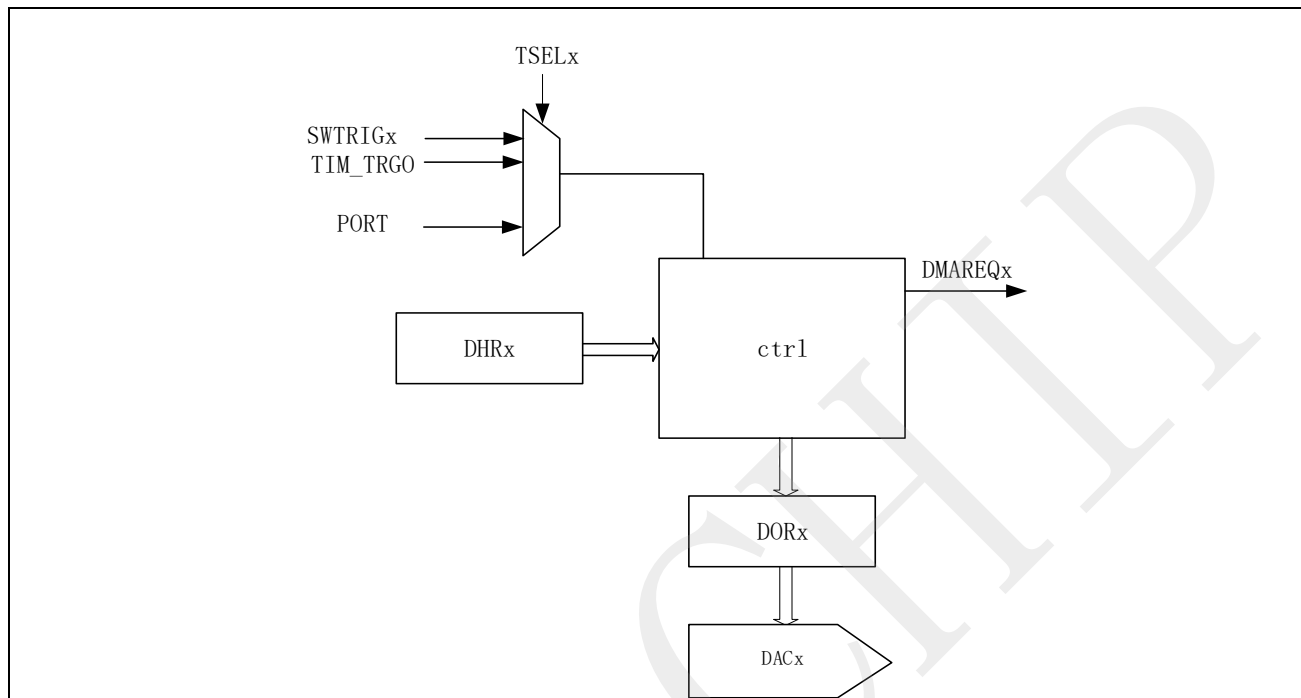
器件集成了 2 个 DAC 转换器，具有 2 个输出通道：DAC_OUT1/2

DAC 主要特性如下：

- 2 个数据保持寄存器
- 12 位模式下数据采用左对齐或右对齐
- 同步更新功能
- 生成噪声波
- 生成三角波
- DMA 功能（包括下溢检测）
- 通过外部触发信号进行转换
- 4 种参考源：AVCC 电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；

41.2 功能说明

41.2.1 框图



41.2.2 DAC 输出缓冲器使能

DAC 可以使用输出缓冲器，可用来降低输出阻抗并在不增加外部运算放大器的情况下直接驱动外部负载。关于输出缓存器的设置参考<运算放大器 OPA>章节。

通过 DAC_CR0 寄存器中的 BufEnx 位，可使能或禁止 DAC 通道输出缓冲器。

41.2.3 DAC 通道使能

将 DAC_CR0 寄存器中的相应 ENx 位置 1，即可使能对应 DAC 通道。经过一段启动时间 t_{WAKEUP} 后，各 DAC 通道被真正使能。

ENx 位只会使能模拟 DAC 通道。即使 ENx 位复位，DAC 通道数字接口仍处于使能状态。

41.2.4 DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 V_{REF+} 之间的输出电压。各 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$DAC_{out} = V_{ref} * DOR/4096$$

41.2.5 DAC 触发选择

如果 TENx 控制位置 1，可通过外部事件（定时计数器、外部中断线）或软件触发转换。TSELx[2:0]控制位将决定哪个可能事件将触发转换，如下表所示。

TSELx	000	001	010	011	100	101	110	111
触发源	GTIM1_	GTIM2_	GTIM3_	ATIM	保留	保留	SW	EXTI
	TRADC	TRADC	TRADC	TRADC			TRIG	

每当 DAC 所选定时器 TRGO 输出或所选外部端口触发上检测到上升沿时，DAC_DHRx 寄存器中存储的最后一个数据即会转移到 DAC_DORx 寄存器中。发生触发后再经过三个 APB1 周期，DAC_DORx 寄存器将会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，转换即会开始。DAC_DHRx 寄存器内容加载到 DAC_DORx 寄存器中后，SWTRIG 即由硬件复位。

注意：

- ENx 位置 1 时，无法更改 TSELx[2:0] 位。如果选择软件触发，DAC_DHRx 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DORx 寄存器。

41.2.6 单模式功能说明

41.2.6.1 DAC 数据格式

根据所选配置模式，数据必须按如下方式写入指定寄存器：

存在三种可能：

- 8 位右对齐：软件必须将数据加载到 DAC_DHR8Rx [7:0] 位（存储到 DHRx[11:4] 位）
- 12 位左对齐：软件必须将数据加载到 DAC_DOR12Lx [15:4] 位（存储到 DHRx[11:0] 位）
- 12 位右对齐：软件必须将数据加载到 DAC_DOR12Rx [11:0] 位（存储到 DHRx[11:0] 位）

根据加载的 DAC_DHRyyyx 寄存器，用户写入的数据将移位并存储到相应的 DHRx（数据保持寄存器 x，即内部非存储器映射寄存器）。之后，DHRx 寄存器将被自动加载，或者通过软件或外部事件触发加载到 DORx 寄存器。

31	24	15	7	0	
					8位右对齐
					12位左对齐
					12位右对齐

41.2.6.2 DAC 转换

DAC_DORx 无法直接写入，任何数据都必须通过加载 DAC_DHRx 寄存器（写入 DAC_DHR8Rx、DAC_DOR12Lx、DAC_DOR12Rx）才能传输到 DAC 通道 x。

如果未选择触发（DAC_CR0 寄存器中的 TENx 位复位），那么经过一个 APB1 时钟周期后，DAC_DHRx 寄存器中存储的数据将自动转移到 DAC_DORx 寄存器。但是，如果选择硬件触发（置位 DAC_CR0 寄存器中的 TENx 位）且触发条件到来，将在三个 PCLK1 时钟周期后进行转移。

当 DAC_DORx 加载了 DAC_DHRx 内容时，模拟输出电压将在一段时间

tSETTLING 后可用，具体时间取决于电源电压和模拟输出负载。

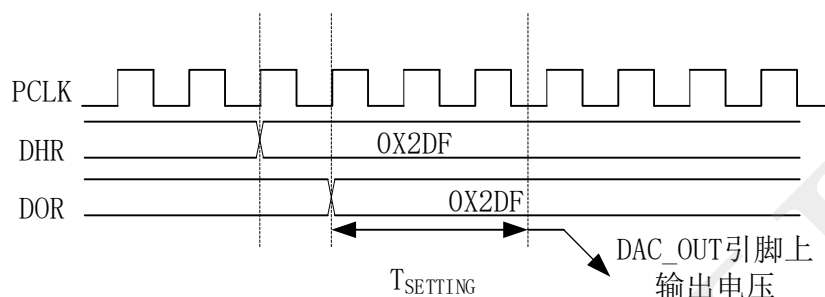


图 41-1 触发不使能时 DAC 转换时序图

生成单个 LFSR（线性反馈移位寄存器）的独立触发

要将 DAC 配置为此转换模式（请参见第 41.2.7 节：生成噪声），需要遵循以下顺序：

1. 将 DAC 通道触发使能位 TEN_x 置 1。
2. 通过设置 $TSEL_x[2:0]$ 位配置触发源。
3. 将 DAC 通道的 $WAVE_x[1:0]$ 位配置为“01”，并在 $MAMP_x[3:0]$ 位中配置相同的 LFSR 掩码值
4. 将 DAC 通道数据加载到所需 DAC_DHR_x 寄存器（ $DOR12RD$ 、 $DOR12LD$ 或 $DHR8RD$ ）。

DAC 通道 x 触发信号到达时， $LFSR_x$ 计数器内容（使用相同的掩码）与 DHR_x 寄存器内容相加，所得总和转移到 DAC_DOR_x 中（三个 APB 时钟周期之后）。 $LFSR_x$ 计数器随即更新。

生成单个三角波的独立触发

要将 DAC 配置为此转换模式（请参见第 41.2.8 节：生成三角波），需要遵循以下顺序：

1. 将 DAC 通道 x 触发使能位 TEN_x 置 1。
2. 通过设置 $TSEL_x[2:0]$ 位配置触发源。
3. 将 DAC 通道 x 的 $WAVE_x[1:0]$ 位配置为“1x”，并在 $MAMP_x[3:0]$ 位中配置相同的最大振幅值
4. 将 DAC 通道 x 数据加载到所需 DAC_DHR_x 寄存器（ $DOR12RD$ 、 $DOR12LD$ 或 $DHR8RD$ ）。

DAC 通道 x 触发信号到达时，DAC 通道 x 三角波计数器内容（使用相同的三角波振幅）与 DHR_x 寄存器内容相加，所得总和转移到 DAC_DOR_x 中（三个 APB 时钟周期之后）。

DAC 通道 x 三角波计数器随即更新。

41.2.7 噪音生成

为了生成可变振幅的伪噪声，可使用 LFSR（线性反馈移位寄存器）。将 $WAVEx[1:0]$ 置为“01”即可选择生成噪声。LFSR 中的预加载值为 $0xAAA$ 。在每次发生触发事件后，经过三个 APB 时钟周期，该寄存器会依照特定的计算算法完成更新。

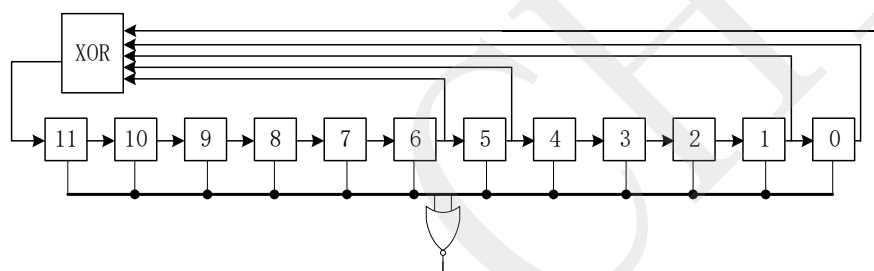


图 41-2 噪音生成

LFSR 值可以通过 DAC_CR0 寄存器中的 $MAMPx[3:0]$ 位来部分或完全屏蔽，在不发生溢出的情况下，该值将与 DAC_DHR_x 的内容相加，然后存储到 DAC_DOR_x 寄存器中。

如果 LFSR 为 $0x0000$ ，将向其注入“1”（防锁定机制）。

可以通过复位 $WAVEx[1:0]$ 位来将 LFSR 波形产生功能关闭。

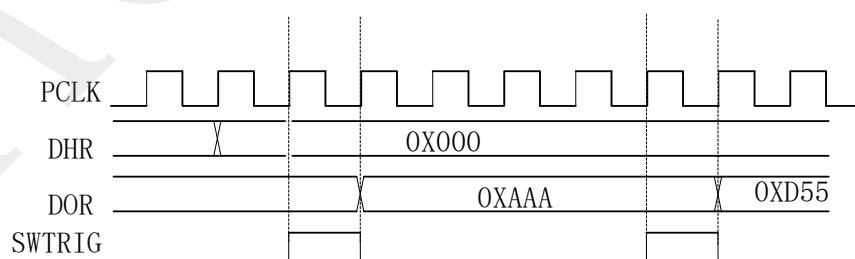


图 41-3 LFSR 产生波形的 DAC 转换（使用软件触发）

41.2.8 三角波生成

可以在直流电流或慢变信号上叠加一个小幅三角波。将 `WAVEx[1:0]` 置为“10”即可选择 DAC 生成三角波。振幅通过 `DAC_CR0` 寄存器中的 `MAMPx[3:0]` 位进行配置。每次发生触发事件后, 经过三个 APB 时钟周期, 内部三角波计数器将会递增。在不发生溢出的情况下, 该计数器的值将与 `DAC_DHRx` 寄存器内容相加, 所得总和将存储到 `DAC_DORx` 寄存器中。只要小于 `MAMPx[3:0]` 位定义的最大振幅, 三角波计数器就会一直递增。一旦达到配置的振幅, 计数器将递减至零, 然后再递增, 以此类推。

可以通过复位 `WAVEx[1:0]` 位来将三角波产生功能关闭。

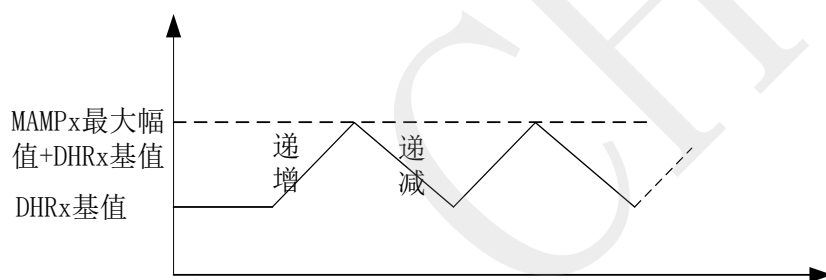


图 41-4 生成 DAC 三角波

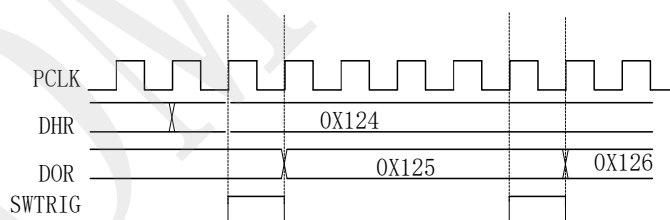


图 41-5 生成三角波波形的 DAC 转换 (使能软件触发)

41.2.9 DMA 请求

每个 DAC 通道都具有 DMA 功能。两个 DMA 通道用于处理 DAC 通道的 DMA 请求。

当 `DMAENx` 位置 1 时, 如果发生触发, 则将产生 DAC DMA 请求。DAC_DHRx

寄存器的值随后转移到 DAC_DORx 寄存器。

在双通道模式下，如果两个 DMAENx 位均置 1，则将产生两个 DMA 请求。如果只需要一个 DMA 请求，用户应仅将相应 DMAENx 位置 1。这样，应用程序可以在双通道模式下通过一个 DMA 请求和一个特定 DMA 通道来管理两个 DAC 通道。

DMA 下溢

DAC DMA 请求没有缓冲队列。这样，如果第二个外部触发到达时尚未收到第一个外部触发的应答，将不会发出新的请求，并且 DAC_SR 寄存器中的 DAM 通道下溢标志 DMAUDRx 将置 1，以报告这一错误状况。DMA 数据传输随即禁止，并且不再处理其他 DMA 请求。DAC 通道仍将继续转换旧有数据。

软件应通过写入“1”来将 DMAUDRx 标志清零，将所用 DMA 数据流的 DMAEN 位清零，并重新初始化 DMA 和 DAC 通道，以便正确地重新开始 DMA 传输。软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于各 DAC 通道，如果使能 DAC_CR0 寄存器中相应的 DMAUDRIEx 位，还将产生中断。

41.3 DAC 寄存器

基地址：0x40007400

寄存器	偏移地址	描述
DAC_CR0	0x000	DAC控制寄存器
DAC_SWTRIG	0x004	DAC软件触发寄存器
DAC_DHR12R1	0x008	DAC1 12位右对齐数据保持寄存器
DAC_DHR12L1	0x00C	DAC1 12位左对齐数据保持寄存器
DAC_DHR8R1	0x010	DAC1 8位右对齐数据保持寄存器
DAC_DHR12R2	0x014	DAC2 12位右对齐数据保持寄存器
DAC_DHR12L2	0x018	DAC2 12位左对齐数据保持寄存器
DAC_DHR8R2	0x01C	DAC2 8位右对齐数据保持寄存器
DAC_DHR12RD	0x020	DAC2 12位右对齐数据保持寄存器
DAC_DHR12LD	0x024	DAC2 12位左对齐数据保持寄存器
DAC_DHR8RD	0x028	DAC2 8位右对齐数据保持寄存器
DAC_DOR1	0x02C	DAC1数据输出寄存器

DAC_DOR2	0x030	DAC2数据输出寄存器
DAC_SR	0x034	DAC状态寄存器
DAC_CR1	0x038	DAC测试寄存器/外部触发选择寄存器

表 41-1 DAC 寄存器

41.3.1 DAC 控制寄存器 (DAC_CR0)

偏移地: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24
RSV	BufEn2	DMAUDRIE2	DMAEN2	MAMP2			
		RW	RW	RW			
23	22	21	20	19	18	17	16
WAVE2		TSEL2			TEN2	BypassEn2	EN2
RW		RW			RW	RW	RW
15	14	13	12	11	10	9	8
RSV	BufEn1	DMAUDRIE1	DMAEN1	MAMP1			
		RW	RW	RW			
7	6	5	4	3	2	1	0
WAVE1		TSEL1			TEN1	BypassEn1	EN1
RW		RW			RW	RW	RW

位	标记	功能描述
31	Reserved	保留
30	BufEn2	DAC2 通道输出缓冲器使能 0: 禁止DAC2 通道输出缓冲器 1: 使能DAC2 通道输出缓冲器 使能时需要OPA中的BUFEN无效
29	DMAUDRIE2	DAC2 DMA 下溢中断使能 0: 禁止DAC2通道DMA下溢中断 1: 使能DAC2通道DMA下溢中断
28	DMAEN2	DAC2 通道DMA使能 0: 禁止DAC2通道DMA模式 1: 使能DAC2通道DMA模式
27:24	MAMP2	DAC2通道掩码/振幅选择器 0000: 不屏蔽LFSR的位0/三角波振幅等于1 0001: 不屏蔽LFSR的位[1:0]/三角波振幅等于3 0010: 不屏蔽LFSR的位[2:0]/三角波振幅等于7 0011: 不屏蔽LFSR的位[3:0]/三角波振幅等于15 0100: 不屏蔽LFSR的位[4:0]/三角波振幅等于31 0101: 不屏蔽LFSR的位[5:0]/三角波振幅等于63 0110: 不屏蔽LFSR的位[6:0]/三角波振幅等于127

		0111: 不屏蔽LFSR的位[7:0]/三角波振幅等于255 1000: 不屏蔽LFSR的位[8:0]/三角波振幅等于511 1001: 不屏蔽LFSR的位[9:0]/三角波振幅等于1023 1010: 不屏蔽LFSR的位[10:0]/三角波振幅等于2047 ≥1011: 不屏蔽LFSR的位[11:0]/三角波振幅等于4095
23:22	WAVE2	DAC2 通道噪声/三角波生成使能 00: 禁止生成波 01: 使能生成噪声波 1x: 使能生成三角波
21:19	TSEL2	DAC2通道触发器选择 000: GTIM1 TRADC触发 001: GTIM2 TRADC触发 010: GTIM3 TRADC触发 011: ATIM TRADC触发 100: 保留 101: 保留 110: 软件触发 111: 外部端口触发
18	TEN2	DAC2通道触发使能 0: 禁止 DAC2 通道触发, 写入 DAC_DHR2 寄存器的数据在一个 APB 时钟周期之后转移到 DAC_DOR2 寄存器 1: 使能 DAC2 通道触发, DAC_DHR2寄存器的数据在三个 APB 时钟周期之后转移到DAC_DOR2 寄存器
17	BypassEn2	DAC2 通道bypass使能 0: 禁止bypass 1: 使能bypass
16	EN2	DAC2通道使能 0: 禁止 DAC2 通道 1: 使能 DAC2 通道
15	Reserved	保留
14	BufEn1	DAC1 通道输出缓冲器使能 0: 禁止DAC1 通道输出缓冲器 1: 使能DAC1 通道输出缓冲器 使能时需要OPA中的BUFEN无效
13	DMAUDRIE1	DAC1 DMA 下溢中断使能 0: 禁止DAC1通道DMA下溢中断 1: 使能DAC1通道DMA下溢中断
12	DMAEN1	DAC1 通道DMA使能 0: 禁止DAC1通道DMA模式 1: 使能DAC1通道DMA模式
11:8	MAMP1	DAC1通道掩码/振幅选择器 0000: 不屏蔽LFSR的位0/三角波振幅等于1

		0001: 不屏蔽LFSR的位[1:0]/三角波振幅等于3 0010: 不屏蔽LFSR的位[2:0]/三角波振幅等于7 0011: 不屏蔽LFSR的位[3:0]/三角波振幅等于15 0100: 不屏蔽LFSR的位[4:0]/三角波振幅等于31 0101: 不屏蔽LFSR的位[5:0]/三角波振幅等于63 0110: 不屏蔽LFSR的位[6:0]/三角波振幅等于127 0111: 不屏蔽LFSR的位[7:0]/三角波振幅等于255 1000: 不屏蔽LFSR的位[8:0]/三角波振幅等于511 1001: 不屏蔽LFSR的位[9:0]/三角波振幅等于1023 1010: 不屏蔽LFSR的位[10:0]/三角波振幅等于2047 ≥1011: 不屏蔽LFSR的位[11:0]/三角波振幅等于4095
7:6	WAVE1	DAC1 通道噪声/三角波生成使能 00: 禁止生成波 01: 使能生成噪声波 1x: 使能生成三角波
5:3	TSEL1	DAC1通道触发器选择 000: GTIM1 TRADC触发 001: GTIM2 TRADC触发 010: GTIM3 TRADC触发 011: ATIM TRADC触发 100: 保留 101: 保留 110: 软件触发 111: 外部端口触发
2	TEN1	DAC1通道触发使能 0: 禁止 DAC1 通道触发, 写入 DAC_DHR1 寄存器的数据在一个 APB 时钟周期之后转移到 DAC_DOR1 寄存器 1: 使能 DAC1 通道触发, DAC_DHR1 寄存器的数据在三个 APB 时钟周期之后转移到DAC_DOR1 寄存器
1	BypassEn1	DAC1 通道bypass使能 0: 禁止bypass 1: 使能bypass
0	EN1	DAC1 通道使能 0: 禁止 DAC1 通道 1: 使能 DAC1 通道

41.3.2 DAC 软件触发寄存器（DAC_SWTRIGR）

偏移地：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved							SWTRIG2	SWTRIG1															
							W1	W1															

位	标记	功能描述
31:2	Reserved	保留
1	SWTRIG2	DAC2软件触发 写1软件触发，自动清零，写0无动作。
0	SWTRIG1	DAC1 软件触发 写1软件触发，自动清零，写0无动作。

41.3.3 DAC1 12 位右对齐数据保持寄存器（DAC_DHR12R1）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR1											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	DHR1	DAC 通道1 12 位右对齐数据 这些位由软件写入，用于为 DAC 通道1指定 12 位数据。

41.3.4 DAC1 12 位左对齐数据保持寄存器（DAC_DHR12L1）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR1												Reserved			
RW															

位	标记	功能描述
31:16	Reserved	保留
15:4	DHR1	DAC 通道1 12 位左对齐数据 这些位由软件写入, 用于为 DAC 通道1指定 12 位数据。
3:0	Reserved	保留

41.3.5 DAC1 8 位右对齐数据保持寄存器 (DAC_DHR8R1)

偏移地址: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DHR1							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	DHR1	DAC 通道1 8 位右对齐数据 这些位由软件写入, 用于为 DAC 通道1指定 8 位数据。写入的为DHR的高八位。

41.3.6 DAC2 12 位右对齐数据保持寄存器 (DAC_DHR12R2)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR2											
				RW											

位	标记	功能描述
31:12	Reserved	保留
11:0	DHR2	DAC 通道2 12 位右对齐数据 这些位由软件写入，用于为 DAC 通道2指定 12 位数据。

41.3.7 DAC2 12 位左对齐数据保持寄存器（DAC_DHR12L2）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR2												Reserved			
RW															

位	标记	功能描述
31:16	Reserved	保留
15:4	DHR2	DAC通道2 12位左对齐数据 这些位由软件写入，用于为DAC通道2指定12位数据。
3:0	Reserved	保留

41.3.8 DAC2 8 位右对齐数据保持寄存器（DAC_DHR8R2）

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DHR2							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	DHR2	DAC 通道2 8 位右对齐数据 这些位由软件写入，用于为 DAC 通道2指定 8 位数据。写入的为DHR的高八位。

41.3.9 双 DAC 12 位右对齐数据保持寄存器 (DAC_DHR12RD)

偏移地址: 0x20

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DHR2											
				RW											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DHR1											
				RW											

位	标记	功能描述
31:28	Reserved	保留
27:16	DHR2	DAC 通道2 12 位右对齐数据 这些位由软件写入, 用于为 DAC 通道2指定 12 位数据。
31:12	Reserved	保留
11:0	DHR1	DAC 通道1 12 位右对齐数据 这些位由软件写入, 用于为 DAC 通道1指定 12 位数据。

41.3.10 双 DAC 12 位左对齐数据保持寄存器 (DAC_DHR12LD)

偏移地址: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DHR2												Reserved			
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR1												Reserved			
RW															

位	标记	功能描述
31:20	DHR2	DAC 通道2 12 位左对齐数据 这些位由软件写入, 用于为 DAC 通道2指定 12 位数据。
19:16	Reserved	保留
15:4	DHR1	DAC 通道1 12 位左对齐数据 这些位由软件写入, 用于为 DAC 通道1指定 12 位数据。
3:0	Reserved	保留

41.3.11 双 DAC 8 位右对齐数据保持寄存器 (DAC_DHR8RD)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHR2								DHR1							
RW								RW							

位	标记	功能描述
31:16	Reserved	保留
15:8	DHR2	DAC 通道2 8 位右对齐数据 这些位由软件写入, 用于为 DAC 通道2指定 8 位数据。写入的为DHR的高八位。
7:0	DHR1	DAC 通道1 8 位右对齐数据 这些位由软件写入, 用于为 DAC 通道1指定 8 位数据。写入的为DHR的高八位。

41.3.12 DAC1 数据输出寄存器 (DAC_DOR1)

偏移地址: 0x2C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DOR1							
								RO							

位	标记	功能描述
31:12	Reserved	保留
11:0	DOR	DAC 1 通道的数据输出

41.3.13 DAC2 数据输出寄存器 (DAC_DOR2)

偏移地址: 0x30

复位值: 0x0000 0000

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

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DOR2											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	DOR	DAC 2 通道的数据输出

41.3.14DAC 状态寄存器 (DAC_SR)

偏移地址: 0x34

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8																																
Reserved																																																							
7							6							5							4							3							2							1							0						
Reserved																		DMAUDR2							DMAUDR1																														
																		RW							RW																														

位	标记	功能描述
31:2	Reserved	保留
1	DMAUDR2	DAC 2 通道 DMA 下溢标志 0: DAC2 通道未发生 DMA 下溢错误状况 1: DAC2 通道发生 DMA 下溢错误状况 (当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 0 通道转换)
0	DMAUDR1	DAC 1 通道 DMA 下溢标志 0: DAC1 通道未发生 DMA 下溢错误状况 1: DAC1 通道发生 DMA 下溢错误状况 (当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 0 通道转换)

41.3.15DAC 控制寄存器 (DAC_CR1)

偏移地址: 0x038

复位值: 0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	PTIRGSEL	RSV	SREF
	RW		RW

位	标记	功能描述
31:7	Reserved	保留
6:4	PTRIGSEL	端口触发选择 000: PA9 001: PB9 010: PC9 011: PD9 100: PE9 101: PF9 11X: 保留
3:2	Reserved	保留，有实体寄存器，无功能
1:0	SREF	DAC参考电压选择 00: 内部1.5V 01: 内部2.5V 10: 外部参考电压ExRef (VREFP / VREFN) 11: AVCC电压

42. 模拟比较器 (VC)

42.1 模拟电压比较器 VC 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“+”输入端电压高于“-”输入端电压时，电压比较器输出为高电平；当“+”输入端电压低于“-”输入端电压时，电压比较器输出为低电平。本产品内部集成的模拟电压比较器 VC 具有以下特性：

- 支持电压比较功能；
- 支持内部 64 阶 VCC 分压（使用分压来源电压需要大于 1.8V）
- 支持 16 个外部输入端口和片内 BGR 输出的参考电压作为电压比较器的输入；
- 支持三种软件可配置的中断触发方式：高电平触发/上升沿触发/下降沿触发；
- 电压比较器的输出可以作为 通用定时器 和 低功耗定时器控制输入；
- 电压比较器的输出可以作为 高级定时器和通用定时器的刹车输入或者捕获输入；
- 支持在超低功耗模式下工作，电压比较器的中断输出可以将芯片从超低功耗模式下唤醒；
- 提供软件可配置的滤波时间以增强芯片的抗干扰能力。
- 注意：使用 VC 需要使能 BGR,参考 BGR 寄存器描述

42.2 电压比较器框架图

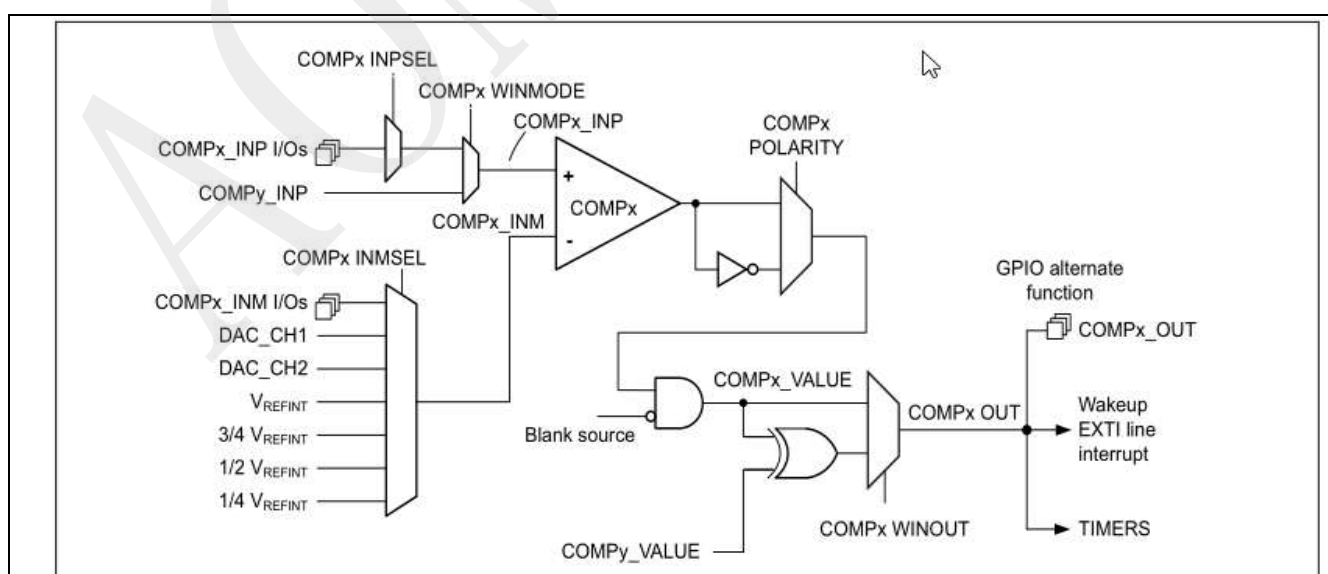


图 42-1 VC 框架图(在COMPx VALUE处插入滤波)

42.3 建立/响应时间

当使用电压比较器时,从 VC 使能或者 VC 的两端输入电压变化到输出正确的结果的时间由 VC 控制寄存器(VC_CR)中的 BIAS_SEL 控制位决定,电流越大 VC 响应越快,典型值从 200nS 到 20 μ s 四档可调。

如果需要内部 BGR 提供的参考电压(1.5V 或 2.5V)作为比较基准电压。内部 BGR 的启动时间大约为 20 μ s,电压比较器需要等待内部 BGR 稳定后才能正常输出。

42.4 滤波时间

在电压比较器固有的建立/响应时间之外,用户可以设置更长的滤波时间来过滤掉系统噪声,比如马达停止时的大电流噪声。

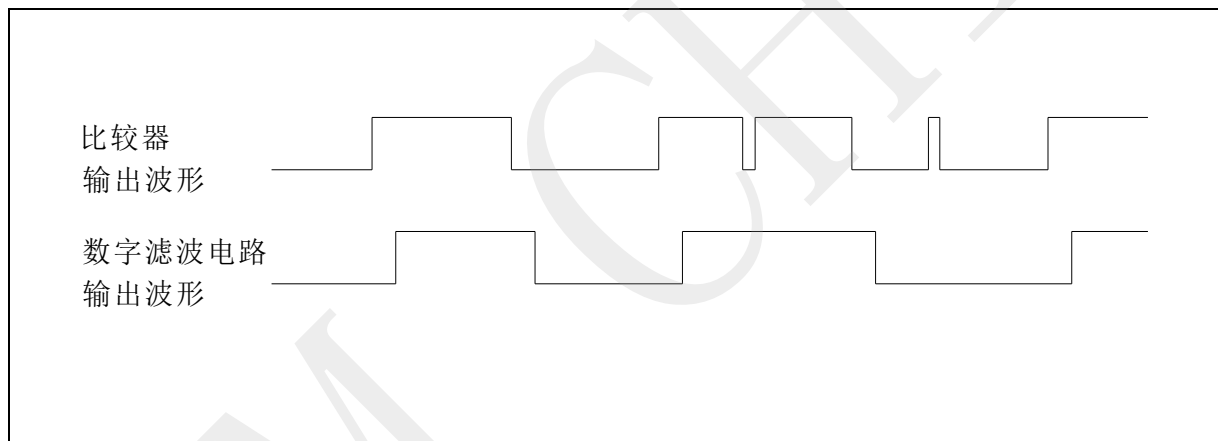


图 42-2 VC 滤波响应时间

42.5 迟滞功能

电压比较器可以选择迟滞功能,迟滞功能使能后的图示如下:

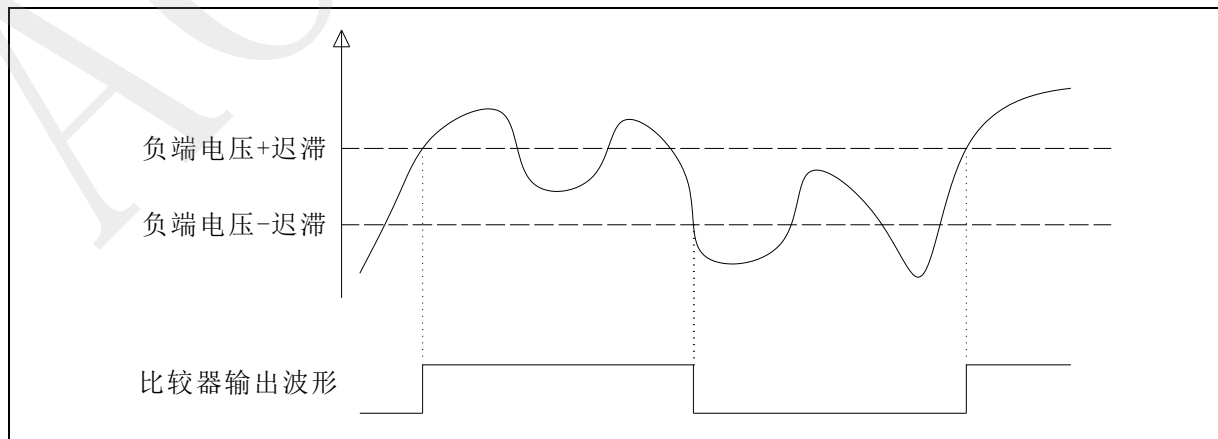


图 42-3 VC 迟滞功能

42.6 VC 寄存器

基地址 0x40012800

寄存器	偏移地址	描述
VC1_CFG	0x000	VC1 配置寄存器
VC1_CR	0x004	VC1 控制寄存器
VC1_FLT	0x008	VC1 滤波寄存器
VC1_OUT	0x00C	VC1 输出控制寄存器
VC1_IFR	0x010	VC1 中断标志寄存器
VC2_CFG	0x100	VC2 配置寄存器
VC2_CR	0x104	VC2 控制寄存器
VC2_FLT	0x108	VC2 滤波寄存器
VC2_OUT	0x10C	VC2 输出控制寄存器
VC2_IFR	0x110	VC2 中断标志寄存器

表 42-1 VC 寄存器

42.6.1 VC1 配置寄存器 (VC1_CFG)

偏移地址: 0x000

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved								VC_HYS_SEL		VC_BIAS_SEL					
								R/W		R/W					
7	6	5	4	3	2	1	0								
VC_REF2P5_SEL	VC_DIV_EN	VC_DIV													
R/W	R/W	R/W													

位	标记	功能描述
31:12	Reserved	保留
11:10	VC_HYS_SEL	VC1 迟滞选择: 00: 没有迟滞 01: 迟滞电压大约10mV 10: 迟滞电压大约20mV 11: 迟滞电压大约30mV
9:8	VC_BIAS_SEL	VC1 功耗选择 (功耗越大, 响应速度越快) 00: 300nA 01: 1.2uA 10: 10uA (需要开启BGR, BGR启动时间大约20μ s) 11: 20uA (需要开启BGR, BGR启动时间大约20μ s)
7	VC_REF2P5_SEL	VC_DIV 参考电压Vref选择 0: VCC 1: ADC_CRO. SREF所选择的参考电压
6	VC_DIV_EN	6位 DAC 使能 1: 使能
5:0	VC_DIV	6位 DAC 配置 000000: 1/64 Vref 000001: 2/64 Vref 000010: 3/64 Vref 000011: 4/64 Vref ... 111110: 63/64 Vref 111111: Vref

42.6.2 VC1 控制寄存器 (VC1_CR)

偏移地址: 0x004

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EN	N_SEL				P_SEL			
							RW	RW				RW			

位	标记	功能描述
31:9	Reserved	保留
8	EN	电压比较器使能 1: 使能电压比较器 0: 关闭电压比较器
7:4	N_SEL	电压比较器“-”端输入选择 0000: 选择 Vref1p2 0001: 选择 OPA1 输出 0010: 选择 OPA2 输出 0011: 选择 OPA3 输出 0100: 选择 OPA4 输出 0101: 选择 DAC1 输出 0110: 选择 DAC2 输出 0111: 选择 VC内部6bit DAC输出 1000: 选择 Vcore 1001: 选择 temp sensor 1010: 选择端口 PA0 1011: 选择端口 PA1 1100: 选择端口 PA2 1101: 选择端口 PA3 1110: 选择端口 PB4 1111: 选择端口 PB5
3:0	P_SEL	电压比较器“+”端输入选择 0000: NC 0001: 选择端口 PA1 0010: 选择端口 PA2 0011: 选择端口 PA3 0100: 选择端口 PB4 0101: 选择端口 PB5

		0110: 选择端口 PB6 0111: 选择端口 PB7 1000: 选择 OPA1 输出 1001: 选择 OPA2 输出 1010: 选择 OPA3 输出 1011: 选择 OPA4 输出 其他: 保留

42.6.3 VC1 滤波寄存器 (VC1_FLT)

偏移地址: 0x008

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved														IE	
Reserved														RW	
7	6	5	4	3	2	1	0								
FLTCK	level	rising	falling	FLTEN	debounce_time										
RW	RW	RW	RW	RW	RW										

位	标记	功能描述
31:8	Reserved	保留
8	IE	VC 中断使能 1: 使能 0: 禁止
7	FLTCK	滤波时钟选择, 0: 150k 1: pclk
6	level	VC 输出信号触发中断选择 1: 使能高电平触发 INT flag 0: 禁止高电平触发 INT flag
5	rising	VC 输出信号触发中断选择 1: 使能上升沿触发 INT flag 0: 禁止上升沿触发 INT flag
4	falling	VC 输出信号触发中断选择 1: 使能下降沿触发 INT flag

		0: 禁止下降沿触发 INT flag
3	FLTEN	1: 启动 VC 滤波 0: VC 无滤波
2:0	debounce_time	VC 输出滤波时间配置 111: 滤波时间大约为28.8ms 110: 滤波时间大约为7.2ms 101: 滤波时间大约为1.8ms 100: 滤波时间大约为450μs 011: 滤波时间大约为112μs 010: 滤波时间大约为28μs 001: 滤波时间大约为14μs 000: 滤波时间大约为7μs 注意: 滤波时间的配置只有在 FLTEN=1 时才有效。

42.6.4 VC1 输出寄存器 (VC1_OUT)

偏移地址: 0x00C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved						BLANKSEL[5:1]									
						RW									
7	6	5	4	3	2	1	0								
BLANKSEL[0]	WINMODE	TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	POL								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:13	Reserved	保留
12:7	BLANKSEL	xx_xxx1: STIM 0C4 xx_xx1x: STIM 0C5 xx_xlxx: ATIM3 0C4 xx_lxxx: ATIM3 0C5 xl_xxxx: GTIM1 0C2 lx_xxxx: 保留
6	WINMODE	窗口模式
5	TIMBK	VC 结果输出到Timer0/1/2/3 刹车控制

		1: 使能; 0: 禁止。
4	TIM3RCLR	VC 结果输出到 TIM3 REFCLR 使能控制 1: 使能; 0: 禁止。
3	TIM2RCLR	VC 结果输出到 TIM2 REFCLR 使能控制 1: 使能; 0: 禁止。
2	TIM1RCLR	VC 结果输出到 TIM1 REFCLR 使能控制 1: 使能; 0: 禁止。
1	TIM0RCLR	VC 结果输出到 TIM0 REFCLR 使能控制 1: 使能; 0: 禁止。
0	POL	VC 输出反向控制 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。

42.6.5 VC1 中断寄存器 (VC1_IFR)

偏移地址: 0x010

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved							VC_WinO	VC_Value	VC_INTF														
							RO	RO	RW0														

位	标记	功能描述
31:3	Reserved	保留
2	VC_WinO	VC 窗口输出状态
1	VC_Value	VC 输出状态
0	VC_INTF	VC 中断标志, 1发生 VC 中断; 0未发生中断; 写0清除中断标志, 写1无效

42.6.6 VC2 配置寄存器 (VC2_CFG)

偏移地址: 0x100

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved								VC_HYS_SEL		VC_BIAS_SEL					
								R/W		R/W					
7	6	5	4	3	2	1	0								
VC_REF2P5_SEL	VC_DIV_EN	VC_DIV													
R/W	R/W	R/W													

位	标记	功能描述
31:12	Reserved	保留
11:10	VC_HYS_SEL	VC1 迟滞选择: 00: 没有迟滞 01: 迟滞电压大约10mV 10: 迟滞电压大约20mV 11: 迟滞电压大约30mV
9:8	VC_BIAS_SEL	VC1 功耗选择 (功耗越大, 响应速度越快) 00: 300nA 01: 1.2uA 10: 10uA (需要开启BGR, BGR启动时间大约20μ s) 11: 20uA (需要开启BGR, BGR启动时间大约20μ s)
7	VC_REF2P5_SEL	VC_DIV 参考电压Vref选择 0: VCC 1: ADC_CR0.SREF所选择的参考电压
6	VC_DIV_EN	6位 DAC 使能 1: 使能
5:0	VC_DIV	6位 DAC 配置 000000: 1/64 Vref 000001: 2/64 Vref 000010: 3/64 Vref 000011: 4/64 Vref ... 111110: 63/64 Vref 111111: Vref

42.6.7 VC2 控制寄存器 (VC2_CR)

偏移地址: 0x004

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EN	N_SEL				P_SEL			
							RW	RW				RW			

位	标记	功能描述
31:9	Reserved	保留
8	EN	电压比较器使能 1: 使能电压比较器 0: 关闭电压比较器
7:4	N_SEL	电压比较器“-”端输入选择 0000: 选择 Vref1p2 0001: 选择 OPA1 输出 0010: 选择 OPA2 输出 0011: 选择 OPA3 输出 0100: 选择 OPA4 输出 0101: 选择 DAC1 输出 0110: 选择 DAC2 输出 0111: 选择 VC内部6bit DAC输出 1000: 选择 Vcore 1001: 选择 temp sensor 1010: 选择端口 PA0 1011: 选择端口 PA1 1100: 选择端口 PA2 1101: 选择端口 PA3 1110: 选择端口 PB4 1111: 选择端口 PB5
3:0	P_SEL	电压比较器“+”端输入选择 0000: NC 0001: 选择端口 PA1 0010: 选择端口 PA2 0011: 选择端口 PA3 0100: 选择端口 PB4 0101: 选择端口 PB5

		0110: 选择端口 PB6 0111: 选择端口 PB7 1000: 选择 OPA1 输出 1001: 选择 OPA2 输出 1010: 选择 OPA3 输出 1011: 选择 OPA4 输出 其他: 保留

42.6.8 VC2 滤波寄存器 (VC2_FLT)

偏移地址: 0x008

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved														IE	
Reserved														RW	
7	6	5	4	3	2	1	0								
FLTCK	level	rising	falling	FLTEN	debounce_time										
RW	RW	RW	RW	RW	RW										

位	标记	功能描述
31:9	Reserved	保留
8	IE	VC 中断使能 1: 使能; 0: 禁止
7	FLTCK	滤波时钟选择, 0: 150k 1: pclk
6	level	VC 输出信号触发中断选择 1: 使能高电平触发 INT flag 0: 禁止高电平触发 INT flag
5	rising	VC 输出信号触发中断选择 1: 使能上升沿触发 INT flag 0: 禁止上升沿触发 INT flag
4	falling	VC 输出信号触发中断选择 1: 使能下降沿触发 INT flag 0: 禁止下降沿触发 INT flag

3	FLTEN	1: 启动 VC 滤波 0: VC 无滤波
2:0	debounce_time	VC 输出滤波时间配置 111: 滤波时间大约为28.8ms 110: 滤波时间大约为7.2ms 101: 滤波时间大约为1.8ms 100: 滤波时间大约为450μs 011: 滤波时间大约为112μs 010: 滤波时间大约为28μs 001: 滤波时间大约为14μs 000: 滤波时间大约为7μs 注意: 滤波时间的配置只有在 FLTEN=1 时才有效。

42.6.9 VC2 输出寄存器 (VC2_OUT)

偏移地址: 0x108

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved						BLANKSEL[5:1]									
						RW									
7	6	5	4	3	2	1	0								
BLANKSEL[0]	WINMODE	TIMBK	TIM3RCLR	TIM2RCLR	TIM1RCLR	TIM0RCLR	POL								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:13	Reserved	保留
12:7	BLANKSEL	xx_xx1: STIM 0C4 xx_xx1x: STIM 0C5 xx_x1xx: ATIM3 0C4 xx_lxxx: ATIM3 0C5 x1_xxxx: GTIM1 0C2 lx_xxxx: 保留
6	WINMODE	窗口模式
5	TIMBK	VC1 filter 结果输出到Timer0/1/2/3 刹车控制 1: 使能; 0: 禁止。
4	TIM3RCLR	VC1 filter 结果输出到 TIM3 REFCLR 使能控制

		1: 使能; 0: 禁止。
3	TIM2RCLR	VC1 filter 结果输出到 TIM2 REFCLR 使能控制 1: 使能; 0: 禁止。
2	TIM1RCLR	VC1 filter 结果输出到 TIM1 REFCLR 使能控制 1: 使能; 0: 禁止。
1	TIM0RCLR	VC1 filter 结果输出到 TIM0 REFCLR 使能控制 1: 使能; 0: 禁止。
0	POL	VC1 filter 结果输出反向到各 TIM0/1/2/3/REFCLR 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。

42.6.10 VC2 中断状态寄存器 (VC2_IFR)

偏移地址: 0x10c

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved						VC_WINO		VC_Value		VC_INTF													
						RO		RO		RW0													

位	标记	功能描述
31:3	Reserved	保留
2	VC_WINO	VC 窗口输出状态
1	VC_VALUE	VC 输出状态
0	VC_INTF	VC 中断标志, 1发生 VC 中断; 0未发生中断; 写0清除中断标志, 写1无效

43. 低电压检测器（LVD）

43.1 LVD 简介

LVD 可用于监测 VCC 及芯片管脚的电压。当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号，用户可根据该信号执行一些紧急任务。

LVD 具有以下特性：

- 4 路监测源，AVCC、PA00、PC13、PE06；
- 16 阶阈值电压，1.8V~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

43.2 LVD 框图

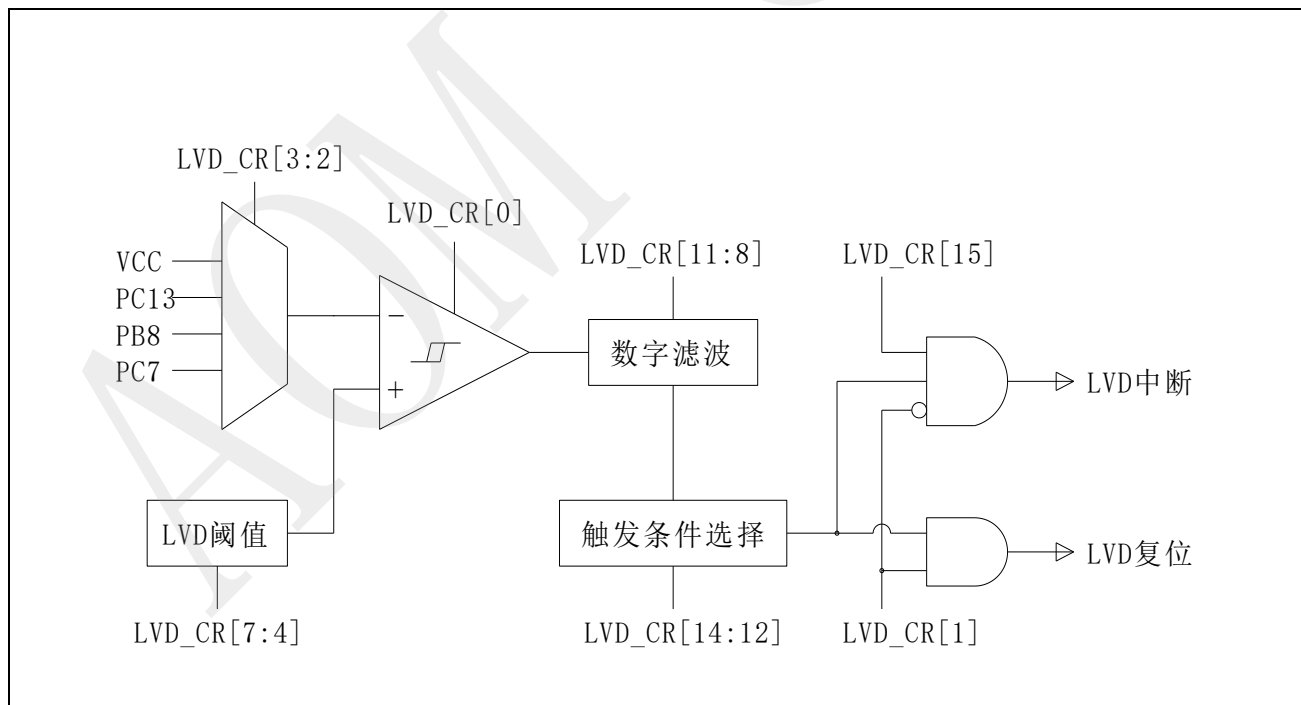


图 43-1 LVD 框图

43.3 迟滞功能

LVD 内置的电压比较器具有迟滞功能，可以增强芯片的抗干扰能力。LVD 的输出信号会等到输入信号高于或低于阈值电压 20mV 后才发生翻转，其输入输出信号示意如下：

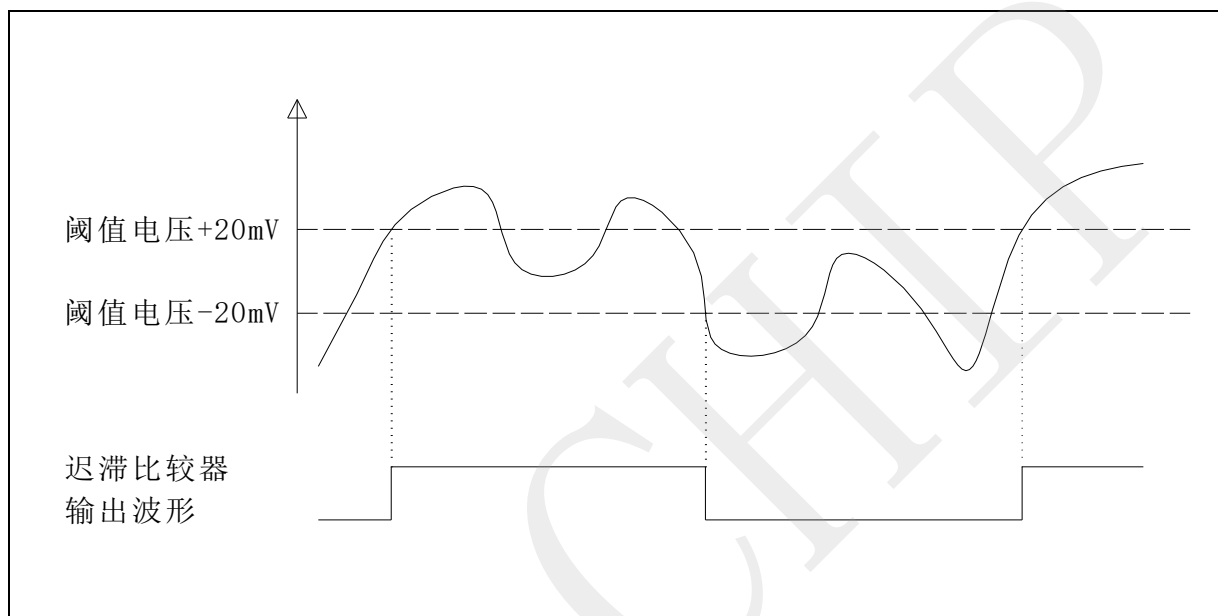


图 43-2 LVD迟滞响应

43.4 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.Debounce_time 的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。使能数字滤波模块，滤波示意如下所示：

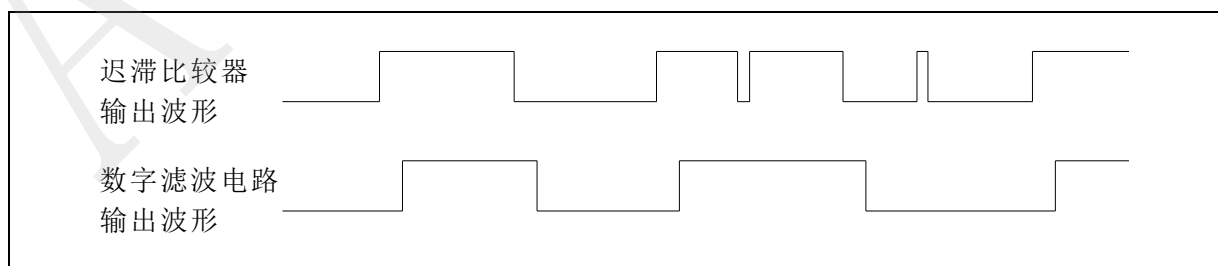


图 43-3 LVD 滤波输出

43.5 配置示例

43.5.1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。

配置方法如下所示：

Step1: 配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2: 配置 LVD_CR.VTDS，选择 LVD 的阈值电压。

Step3: 配置 LVD_CR.Debounce_time，选择 LVD 滤波时间。

Step4: 配置 LVD_CR.FLTEN，使能 LVD 滤波。

Step5: 设置 LVD_CR.HTEN 为 1，选择高电平触发 LVD 动作。

Step6: 设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位。

Step7: 设置 LVD_CR.LVDEN 为 1，使能 LVD。

43.5.2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。

配置方法如下所示：

Step1: 配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2: 配置 LVD_CR.VTDS，选择 LVD 的阈值电压。

Step3: 配置 LVD_CR.Debounce_time，选择 LVD 滤波时间。

Step4: 配置 LVD_CR.FLTEN，使能 LVD 滤波。

Step5: 设置 LVD_CR.RTEN 和 LVD_CR.FTEN 为 1，选择电平变化触发 LVD 动作。

Step6: 设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断。

Step7: 设置 LVD_CR.IE 为 1，使能 LVD 中断。

Step8: 使能 NVIC 中断向量表中的 LVD 中断。

Step9: 设置 LVD_CR.LVDEN 为 1，使能 LVD。

Step10: 在 LVD 的中断服务程序中执行用户需要进行的操作；退出中断服务程序前向 LVD_IFR 写入 0x00 以清除中断标志。

43.6 LVD 寄存器

基地址：0x40012000

寄存器	偏移地址	描述
LVD_CR	0x000	LVD 配置寄存器
LVD_FLT	0x004	LVD 滤波控制
LVD_IFR	0x008	LVD 中断标志寄存器

表 43-1 LVD 寄存器

43.6.1 LVD 配置寄存器 (LVD_CR)

偏移地址：0x000

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
VTDS								Source_sel				ACT		LVDEN									
RW								RW				RW		RW									

位	标记	功能描述
31:8	Reserved	保留
7:4	VTDS	LVD 阈值电压选择 1111: 3.3v 1110: 3.2v 1101: 3.1v 1100: 3.0v 1011: 2.9v 1010: 2.8v 1001: 2.7v 1000: 2.6v 0111: 2.5v 0110: 2.4v 0101: 2.3v 0100: 2.2v 0011: 2.1v 0010: 2.0v 0001: 1.9v 0000: 1.8v
3:2	Source_sel	LVD 监测来源选择

		11: PE06 端口输入电压 10: PC13 端口输入电压 01: PA00 端口输入电压 00: AVCC 电源电压
1	ACT	LVD触发动作选择 1: 系统复位 0: NVIC中断
0	LV DEN	LVD使能控制 1: 使能LVD 0: 禁止LVD

43.6.2 LVD 控制寄存器 (LVD_FLT)

偏移地址: 0x004

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved														IE	
Reserved														RW	
7	6	5	4	3	2	1	0								
RSV	level	rising	falling	FLTEN	debounce_time										
	RW	RW	RW	RW	RW										

位	标记	功能描述
31:9	Reserved	保留
8	IE	LVD 中断使能 1: 使能; 0: 禁止
7	Reserved	保留
6	level	LVD 输出信号触发中断选择 1: 使能高电平触发 INT flag 0: 禁止高电平触发 INT flag
5	rising	LVD 输出信号触发中断选择 1: 使能上升沿触发 INT flag 0: 禁止上升沿触发 INT flag
4	falling	LVD 输出信号触发中断选择 1: 使能下降沿触发 INT flag 0: 禁止下降沿触发 INT flag

3	FLTEN	1: 启动 LVD 滤波 0: LVD无滤波
2:0	debounce_time	LVD 输出滤波时间配置 111: 滤波时间大约为28.8ms 110: 滤波时间大约为7.2ms 101: 滤波时间大约为1.8ms 100: 滤波时间大约为450μ s 011: 滤波时间大约为112μ s 010: 滤波时间大约为28μ s 001: 滤波时间大约为14μ s 000: 滤波时间大约为7μ s 注意: 滤波时间的配置只有在 FLTEN=1 时才有效。

43.6.3LVD 中断寄存器 (LVD_IFR)

偏移地址: 0x008

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
Reserved							LVD_Filter		INTF														
							RO		RW0														

位	标记	功能描述
31:2	Reserved	保留
1	LVD_Filter	LVD Filter 后的状态
0	INTF	LVD中断标志: 1: 发生LVD中断; 0: 未发生中断; 写0清除中断标志, 写1无效。

44. 运算放大器（OPA）

OPA 模块适用于简易放大器和电压跟随器应用。OPA 的输入范围是 0V 到 AVCC，输出范围是 0.1V 到 AVCC-0.2V。

44.1 OPA 特性

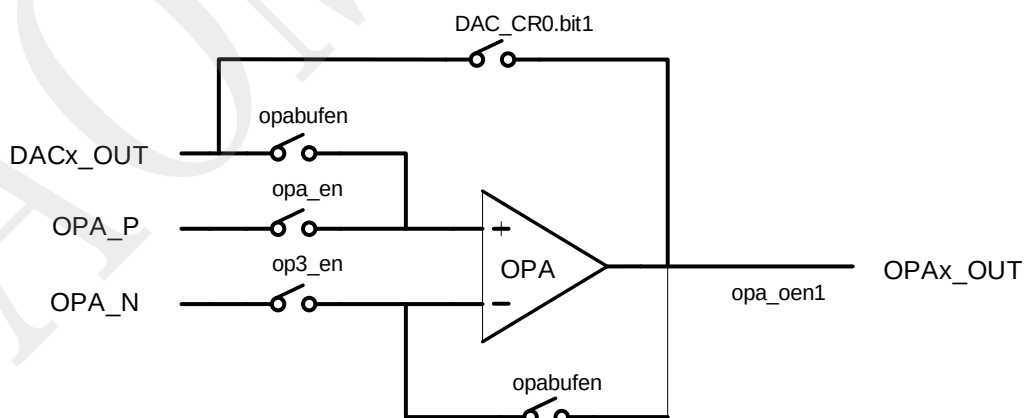
- 4 个独立配置运放
- OPA 的输入范围是 0 到 AVCC，输出范围是 0.1V 到 AVCC-0.2V 可编程增益
- 可配置为以下模式
 - 通用运放模式（general purpose OPA）
 - DAC 电压跟随器

44.2 OPA 功能描述

OPA 可以通过使用外部元件组成放大器来放大小信号模拟输入信号，输出为放大后的信号。

44.2.1 OPA (复用为 DAC buffer)

OPA 的示意图如下：



44.3 OPA 校零

OPA 支持不同模式的自动校零功能，可以软件自己控制，也可以硬件控制。高精度 OPA 应用，在每次使用 OPA 前,需要 OPA 校零。校零建议时间 10 μ s 以上。校零后请立即采样。

44.3.1 软件控制

- 使能需要校零的 OPA 的 OPA_CRx.AZEN 为 1
- 设置 OPA_CR.CLK_SW_SET 为 1
- 软件写 OPA_CR.AZ_PULSE 为 1
- 设定校零时间，等待校零时间结束，清除 OPA_CR.AZ_PULSE 为 0

44.3.2 软件触发控制

- 使能需要校零的 OPA 的 OPA_CRx.AZEN 为 1
- 选择需要校零的时间参数 OPA_CR.CLK_SEL
- 设置软件触发 OPA_CR.TRIGGER

44.3.3 ADC 触发控制

- 使能需要校零的 OPA 的 OPA_CRx.AZEN 为 1
- 选择需要校零的时间参数 OPA_CR.CLK_SEL
- 使能 ADC 触发 ADCTR_EN

44.4 OPA 寄存器

基地址：0x40007800

寄存器	偏移地址	描述
OPA_AZ	0x000	OPA 校准寄存器
OPA_CR	0x004	OPA 控制寄存器
OPA_BIAS	0x008	OPA 电流控制寄存器

表 44-1 OPA 寄存器

44.4.1 OPA 校准寄存器（OPA_AZ）

偏移地址：0x000

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Reserved																							
7	6	5	4	3	2	1	0																
CLK_SEL								CLK_SW_SET				AZ_PULSE		TRIGGER		ADCTR_EN							
RW								RW				RW		W1		RW							

位	标记	功能描述
31:8	Reserved	保留
7:4	CLK_SEL	自动校准脉冲宽度设置 0000:1个PCLK周期 0001:2个PCLK周期 0010:4个PCLK周期 0011:8个PCLK周期 0100:16个PCLK周期 0101:32个PCLK周期 0110:64个PCLK周期 0111:128个PCLK周期 1000:256个PCLK周期 1001:512个PCLK周期 1010:1024个PCLK周期 1011:2048个PCLK周期 11xx:4096个PCLK周期 建议自动校准时间为10μs
3	CLK_SW_SET	自动校零选择 1: 软件校准使能 0: 软件校准禁止, 软件触发校准使能
2	AZ_PULSE	软件校准 1: 校准 0: 无效
1	TRIGGER	软件触发校准设置, 自动清零 1: 软件触发校准 0: 无效
0	ADCTR_EN	ADC启动触发OPA自动校准使能 1: 使能 0: 禁止

		高精度OPA应用，在每次使用OPA前, 需要OPA校零。校零建议时间10μ s以上。校零后请立即采样。

44.4.2 OPA 控制寄存器（OPA_CR）

偏移地址：0x004

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Reserved				BUFEN4	BUFEN3	BUFEN2	BUFEN1								
				RW	RW	RW	RW								
7	6	5	4	3	2	1	0								
AZEN4	AZEN3	AZEN2	AZEN1	EN4	EN3	EN2	EN1								
RW	RW	RW	RW	RW	RW	RW	RW								

位	标记	功能描述
31:12	Reserved	保留
11	BUFEN4	OPA4单位增益功能使能
10	BUFEN3	OPA3单位增益功能使能
9	BUFEN2	OPA2单位增益功能使能
8	BUFEN1	OPA1单位增益功能使能
7	AZEN4	OPA4自动校零使能
6	AZEN3	OPA3自动校零使能
5	AZEN2	OPA2自动校零使能
4	AZEN1	OPA1自动校零使能
3	EN4	OPA4使能
2	EN3	OPA3使能
1	EN2	OPA2使能
0	EN1	OPA1使能

44.4.3 OPA 偏置寄存器（OPA_BIAS）

偏移地址：0x008

复位值: 0x000006DB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BIAS3			BIAS2			BIAS1		
							RW			RW			RW		

位	标记	功能描述
31:9	Reserved	保留
8:6	BIAS3	OPA3 偏置选择
5:3	BIAS2	OPA2 偏置选择
2:0	BIAS1	OPA1 偏置选择

45. SysTick 定时器

45.1 SysTick 定时器简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0 处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询计数标志。

45.2 设置 SysTick

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

Step1: 配置 SysTick->CTRL.ENABLE 为 0，禁止 SysTick。

Step2: 配置 SysTick->CTRL.CLKSOURCE，选择 SysTick 的时钟源。

Step3: 配置 SysTick->LOAD，选择 SysTick 的溢出周期。

Step4: 向 SysTick->VAL 写入任意值，清零 SysTick->VAL 及 SysTick->CTRL.COUNTFLAG。

Step5: 配置 SysTick->CTRL.TICKINT 为 1，使能 SysTick 中断。

Step6: 配置 SysTick->CTRL.ENABLE 为 1，使能 SysTick。

Step7: 在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。

注意：SysTick 溢出周期为 SysTick->LOAD+1，配置示例如下：

时钟源	SysTick->LOAD	溢出周期
HCLK 4MHz	3999	1ms

45.3 SysTick 寄存器

地址	名称	CMSIS 符号	全名
0xE000 E010	SYS_CSR	SysTick->CTRL	SysTick控制和状态寄存器
0xE000 E014	SYS_RVR	SysTick->LOAD	SysTick重装载寄存器
0xE000 E018	SYS_CVR	SysTick->VAL	SysTick当前值寄存器
0xE000 E01C	SYS_CALIR	SysTick->CALIB	SysTick校准值寄存器

45.3.1 SysTick 控制和状态寄存器（CTRL）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															COUNTFLAG
															RO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CLKSOURCE	TICKINT	ENABLE	
												RW	RW	RW	

位	符号	功能描述	类型	复位值
31:17	Reserved	—	—	—
16	COUNTFLAG	SysTick定时器溢出标志 1: SysTick定时器发生下溢出 0: SysTick定时器未发生溢出 读该寄存器，可清除COUNTFLAG标志	RO	0
15:3	Reserved	—	—	—
2	CLKSOURCE	SysTick时钟源选择 无参考时钟，始终以系统时钟作为SysTick时钟	RW	0
1	TICKINT	SysTick中断使能 1: 使能中断 0: 禁止中断	RW	0
0	ENABLE	SysTick定时器使能 1: 使能SysTick 0: 禁止SysTick	RW	0

45.3.2 SysTick 重载寄存器 (LOAD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RELOAD[23:16]							
								RW							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[15:0]															
RW															

位	符号	功能描述	类型	复位值
31:24	Reserved	—	—	—
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

45.3.3 SysTick 当前值寄存器 (VAL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CURRENT [23:16]							
								RW							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENT [15:0]															
RW															

位	符号	功能描述	类型	复位值
31:24	Reserved	—	—	—
23:0	CURRENT	读取该寄存器，获取SysTick定时器的当前计数值 写任意值到该寄存器，清零该寄存器及COUNTFLAG	RW	未定义

45.3.4 SysTick 校准值寄存器 (CALIB)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOREF	SKEW	Reserved						TENMS[23:16]							
RO	RO							RO							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TENMS[15:0]															
RO															

位	符号	功能描述	类型	复位值
31	NOREF	SysTick当前计数时钟标志 1: 当前计数时钟为内核时钟 0: 无功能	RO	-
30	SKEW	TENMS精度指示 1: TENMS值代表粗略的10ms 0: TENMS值代表精确的10ms	RO	
29:24	Reserved	-	-	
23:0	TENMS	10ms校准值	RO	-

46. 电气特性

46.1 测试条件

除非特别说明，所有电压的都以 VSS 为基准。

46.1.1 最小和最大数值

除非特别说明，在生产线上通过对 100% 的产品在环境温度 $T_A=25^{\circ}\text{C}$ 和 $T_A=T_{Amax}$ 下执行的测试(T_{Amax} 与选定的温度范围匹配)，所有最小和最大值将在最坏的环境温度、供电电压和时钟频率条件下得到保证。

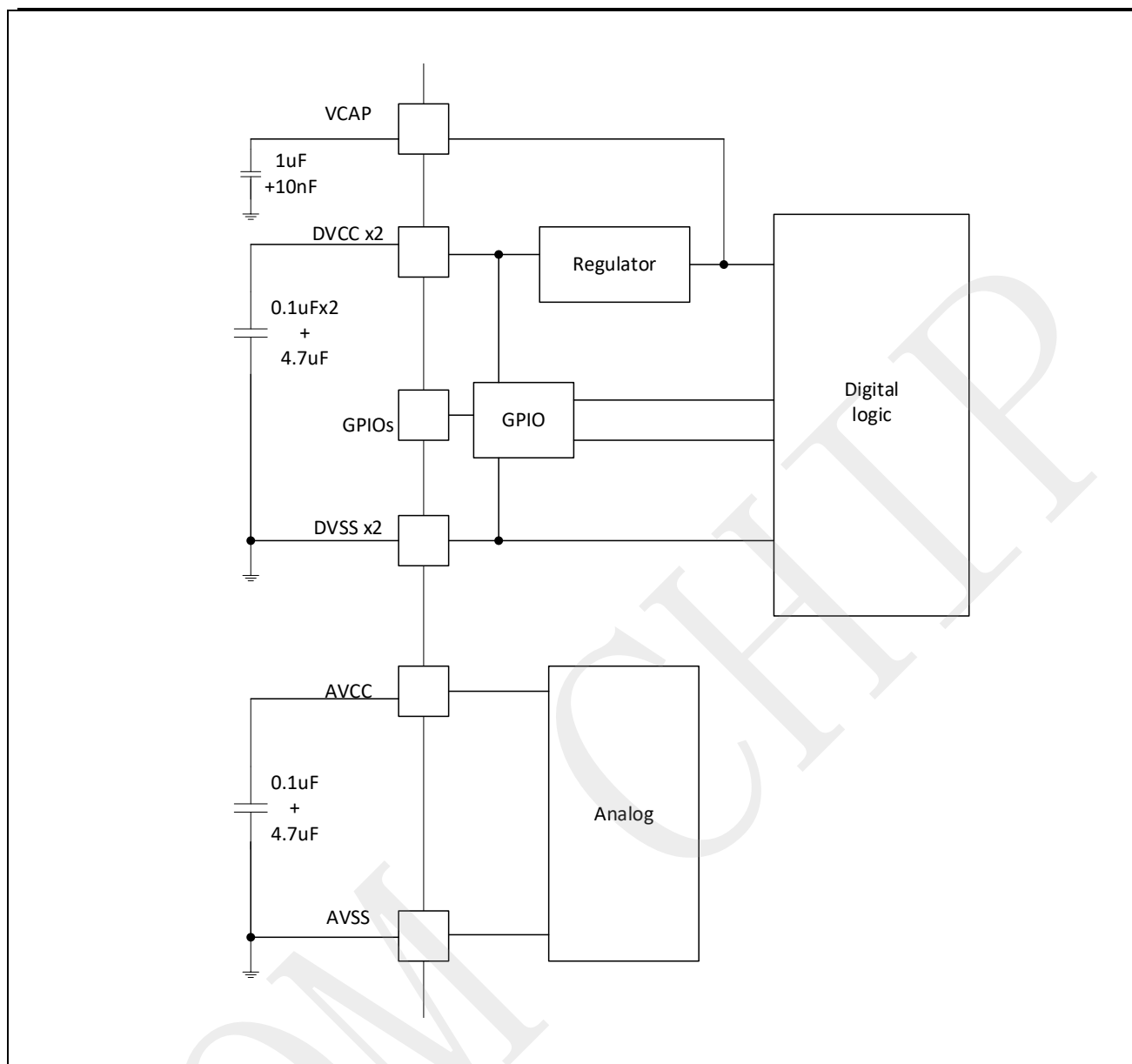
在每个表格下方的注解中说明为通过综合评估、设计模拟和/或工艺特性得到的数据，不会在生产线上进行测试；在综合评估的基础上，最小和最大数值是通过样本测试后，取其平均值再加减三倍的标准分布(平均 $\pm 3\Sigma$)得到。

46.1.2 典型数值

除非特别说明，典型数据是基于 $T_A=25^{\circ}\text{C}$ 和 $VCC=3.3\text{V}$ ($1.8\text{V} \leq VCC \leq 5.5\text{V}$ 电压范围)。
这些数据仅用于设计指导而未经测试。

典型的 ADC 精度数值是通过对一个标准的批次采样，在所有温度范围下测试得到，95% 产品的误差小于等于给出的数值(平均 $\pm 2\Sigma$)。

46.1.3 供电方案



注意：

- 每组电源都需要一个去耦电容，去耦电容尽量靠近相应电源管脚。

46.2 绝对最大额定值

加在器件上的载荷如果超过“绝对最大额定值”列表中给出的值，可能会导致器件永久性地损坏。这里只是给出能承受的最大载荷，并不意味着在此条件下器件的功能性操作无误。器件长期工作在最大值条件下会影响器件的可靠性。

符号	描述	最小值	最大值	单位
VCC - VSS	外部主供电电压（包含AVCC和DVCC） ⁽¹⁾	-0.3	5.5	V
V _{IN}	在其它引脚上的输入电压 ⁽²⁾	VSS-0.3	VCC + 0.3	V

$ \Delta V_{CCx} $	不同供电引脚之间的电压差		50	mV
$ V_{SSx} - V_{SS} $	不同接地引脚之间的电压差		50	mV
$V_{ESD}(HBM)$	ESD静电放电电压（人体模型）	参考绝对最大值电气参数		V

表 46-1 电压特性

- 所有的电源(DVCC,AVCC)和地(DVSS, AVSS)引脚必须始终连接到外部允许范围内的供电系统上。本芯片由两组电源（DVCC-DVSS、AVCC-AVSS）进行供电，上电时请确保 DVCC 和 AVCC 同时上电，且任何时候两组电源的压差不得超过 0.3V。
- $I_{INJ(PIN)}$ 绝对不可以超过它的极限，即保证 V_{IN} 不超过其最大值。如果不能保证 V_{IN} 不超过其最大值，也要保证在外部限制 $I_{INJ(PIN)}$ 不超过其最大值。当 $V_{IN}>V_{CC}$ 时，有一个正向注入电流；当 $V_{IN}<V_{SS}$ 时，有一个反向注入电流。

符号	描述	最大值 ⁽¹⁾	单位
I_{VCC}	经过DVCC/AVCC电源线的总电流(供应电流) ⁽¹⁾	300	mA
I_{VSS}	经过VSS地线的总电流(流出电流) ⁽¹⁾	300	mA
I_{IO}	任意I/O和控制引脚上的输出灌电流	25	mA
	任意I/O和控制引脚上的输出电流	-25	mA
$I_{INJ(PIN)}$ ⁽²⁾⁽³⁾	RESETB引脚的注入电流	+/-5	mA
	XTH的XTHI引脚和XTL的XTLI引脚的注入电流	+/-5	mA
	其他引脚的注入电流 ⁽⁴⁾	+/-5	mA
$\sum I_{INJ(PIN)}$ ⁽²⁾	所有I/O和控制引脚上的总注入电流 ⁽⁴⁾	+/-25	mA

表 46-2 电流特性

- 所有的电源(DVCC,AVCC)和地(DVSS,AVSS)引脚必须始终连接到外部允许范围内的供电系统上。
- $I_{INJ(PIN)}$ 绝对不可以超过它的极限，即保证 V_{IN} 不超过其最大值。如果不能保证 V_{IN} 不超过其最大值，也要保证在外部限制 $I_{INJ(PIN)}$ 不超过其最大值。当 $V_{IN}>V_{CC}$ 时，有一个正向注入电流；当 $V_{IN}<V_{SS}$ 时，有一个反向注入电流。
- 反向注入电流会干扰器件的模拟性能。
- 当几个 I/O 口同时有注入电流时， $\sum I_{INJ(PIN)}$ 的最大值为正向注入电流与反向注入电流的即时绝对值之和。该结果基于在器件 4 个 I/O 端口上 $\sum I_{INJ(PIN)}$ 最大值的特性。

符号	描述	数值	单位
T_{STG}	储存温度范围	-60 ~ +150	°C
T_J	最大结温度	105	°C

表 46-3 温度特性

46.3 工作条件

46.3.1 通用工作条件

符号	参数	条件	最小值	最大值	单位
f_{HCLK}	内部AHB时钟频率		0	48	MHz
f_{PCLK0}	内部APB0时钟频率		0	48	MHz
f_{PCLK1}	内部APB1时钟频率		0	48	MHz
DVCC	标准工作电压		1.8	5.5	V
AVCC ⁽¹⁾	模拟部分工作电压	必须与DVCC ⁽²⁾ 相同	1.8	5.5	V
P_D	功率耗散 $T_A=85^{\circ}\text{C}$	LQFP100		476	mW
	功率耗散 $T_A=85^{\circ}\text{C}$	LQFP64		455	mW
	功率耗散 $T_A=85^{\circ}\text{C}$	LQFP48		364	mW
	功率耗散 $T_A=85^{\circ}\text{C}$	LQFP32		357	mW
T_A	环境温度	最大功率消耗	-40	85	$^{\circ}\text{C}$
		低功率消耗 ⁽³⁾	-40	105	$^{\circ}\text{C}$
T_J	结温度范围		-40	105	$^{\circ}\text{C}$

表 46-4 通用工作条件

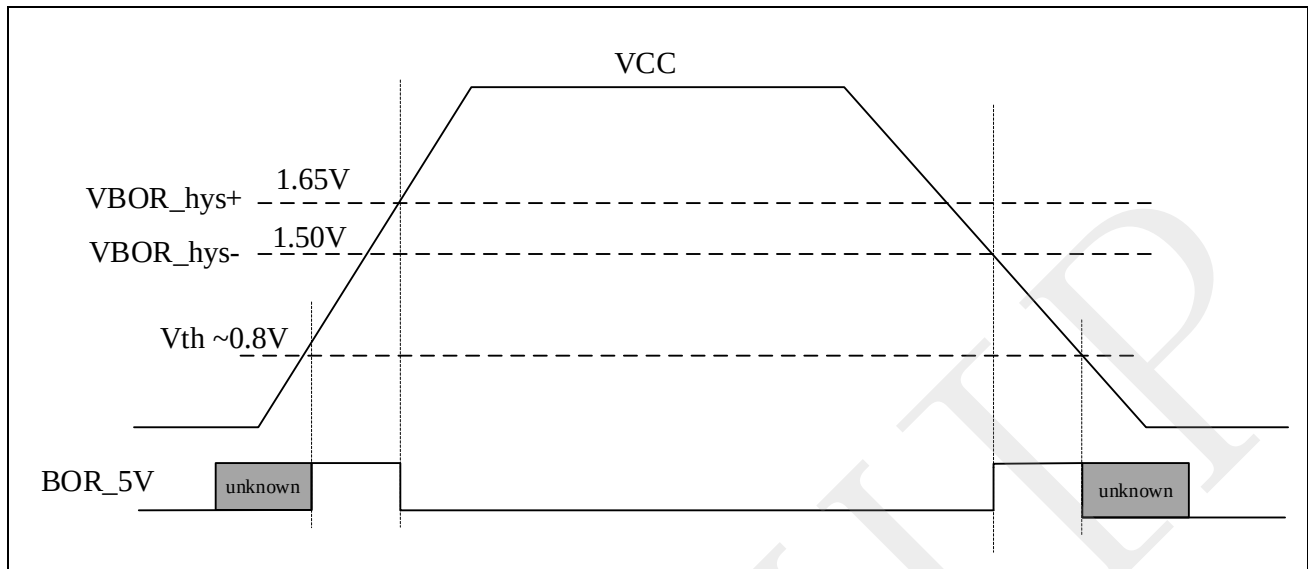
1. 当使用 ADC 时，参见 ADC 电气参数。
2. 建议使用相同的电源为 DVCC 和 AVCC 供电，在上电和正常操作期间，DVCC 和 AVCC 之间最多允许有 300mV 的差别。
3. 在较低的功率耗散的状态下，只要 T_J 不超过 T_{Jmax} ， T_A 可以扩展到这个范围。

46.3.2 上电和掉电时的工作条件

符号	参数	条件	最小值	最大值	单位
t_{VCC}	VCC上升速率		0	∞	$\mu\text{s/V}$
	VCC下降速率		10	∞	$\mu\text{s/V}$

表 46-5 上电和掉电的工作条件

46.3.3 内嵌复位和 LVD 模块特性



1. 设计保证，不在生产中测试。

图 46-1 POR/Brown Out 示意图

符号	参数	条件	最小值	典型值	最大值	单位
V_{por}	POR 释放电压（上电过程）		1.45	1.50	1.65	V
	BOR 检测电压（掉电过程）					

表 46-6 POR/Brown Out

符号	参数	条件	最小值	典型值	最大值	单位
V_{ex}	外部输入电压范围		0		VCC	V
V_{level}	检测阈值	LVD_CR.VTDS = 0x00	1.75	1.85	1.95	V
		LVD_CR.VTDS = 0x01	1.85	1.95	2.05	
		LVD_CR.VTDS = 0x02	1.95	2.05	2.15	
		LVD_CR.VTDS = 0x03	2.06	2.16	2.26	
		LVD_CR.VTDS = 0x04	2.16	2.26	2.36	
		LVD_CR.VTDS = 0x05	2.26	2.36	2.46	
		LVD_CR.VTDS = 0x06	2.36	2.46	2.56	
		LVD_CR.VTDS = 0x07	2.47	2.57	2.67	
		LVD_CR.VTDS = 0x08	2.57	2.67	2.77	
		LVD_CR.VTDS = 0x09	2.67	2.77	2.87	
		LVD_CR.VTDS = 0x0A	2.77	2.87	2.97	
		LVD_CR.VTDS = 0x0B	2.88	2.98	3.08	
		LVD_CR.VTDS = 0x0C	2.98	3.08	3.18	
		LVD_CR.VTDS = 0x0D	3.08	3.18	3.28	
		LVD_CR.VTDS = 0x0E	3.19	3.29	3.39	
		LVD_CR.VTDS = 0x0F	3.29	3.39	3.49	
I_{comp}	功耗			0.12		μA
$T_{response}$	响应时间			80		μs
T_{setup}	建立时间			400		μs
V_{hyste}	迟滞电压			40		mV
T_{filter}	滤波时间	LVD_debounce = 000		7		μs
		LVD_debounce = 001		14		
		LVD_debounce = 010		28		
		LVD_debounce = 011		112		
		LVD_debounce = 100		450		
		LVD_debounce = 101		1800		
		LVD_debounce = 110		7200		
		LVD_debounce = 111		28800		

表 46-7 LVD 模块特性

46.3.4 内置的参考电压

符号	参数	条件	最小值	典型值	最大值	单位
V_{REF25}	Internal 2.5v Reference Voltage	常温25°C 3.3V	2.475	2.5	2.525	V
V_{REF25}	Internal 2.5v Reference Voltage ⁽¹⁾	-40°C ~85°C; 2.8V~5.5V	2.463	2.5	2.525	V
V_{REF15}	Internal 1.5v Reference Voltage	常温25°C 3.3V	1.485	1.5	1.515	V
V_{REF15}	Internal 1.5v Reference Voltage ⁽¹⁾	-40°C ~85°C; 1.8V~5.5V	1.477	1.5	1.519	V
T_{Coeff}	Internal 2.5v 1.5v temperature coefficient	-40 ~ 85°C			120	ppm/°C

1. 数据基于考核结果，不在生产中测试

46.3.5 供电电流特性

电流消耗是多种参数和因素的综合指标，这些参数和因素包括工作电压、环境温度、I/O 引脚的负载、产品的软件配置、工作频率、I/O 脚的翻转速率、程序在存储器中的位置以及执行的代码等。

微控制器处于下列条件：

- 所有的 I/O 引脚都处于输入模式，并连接到一个静态电平上——VCC 或 VSS(无负载)。
- 所有的外设都处于关闭状态，除非特别说明。
- 闪存存储器的访问时间调整到 f_{HCLK} 的频率(0~24MHz 时为 0 个等待周期，24~48MHz 时为 1 个等待周期)。
- 当开启外设时： $f_{PCLK0} = f_{HCLK}$ ， $f_{PCLK1} = f_{HCLK}$ 。

Symbol	Parameter	Conditions			Typ ⁽¹⁾	Max ⁽²⁾	Unit
I_{DD} (Run in RAM)	All peripherals clock ON, Run while(1) in RAM	$V_{core}=1.5V$ $V_{CC}=3.3V$ $T_A=25^{\circ}C$	RCH clock source	4M	940		μA
				8M	1830		
				16M	3560		
				22.12M	4930		
				24M	5340		
			PLL RCH4M to xxM clock source	32M	7190		
				48M	10680		
	All peripherals clock OFF, Run while(1) in RAM	$V_{core}=1.5V$ $V_{CC}=3.3V$ $T_A=25^{\circ}C$	RCH clock source	4M	440		μA
				8M	830		
				16M	1560		
				22.12M	2140		

				24M	2310		
			PLL RCH4M to xxM clock source	32M	3200		
				48M	4710		
I_{DD} (Run CoreMark)	All peripherals clock OFF, Run CoreMark in Flash	$V_{core}=1.5V$ $V_{CC}=3.3V$ $T_A=25^{\circ}C$	RCH clock source	4M	990		μA
				8M	1840		
				16M	3480		
				22.12M	4650		
				24M	5000		
			PLL RCH4M to xxM	48M FlashWait=1	7600		
I_{DD} (Run mode)	All peripherals clock ON, Run while(1) in Flash	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$ $T_A=-40\sim 85^{\circ}C$	RCH clock source	4M	1250	1790	μA
				8M	2360	3390	
				16M	4640	6450	
				22.12M	6260	8750	
				24M	6750	9460	
		$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$ $T_A=-40\sim 85^{\circ}C$	PLL RCH4M to xxM clock source	16M	4910	6240	μA
				24M	6850	8850	
				32M FlashWait=1	8240	9550	
				40M FlashWait=1	10130	11840	
				48M FlashWait=1	12010	14000	
	All peripherals clock OFF, Run while(1) in Flash	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$ $T_A=-40\sim 85^{\circ}C$	PLL RCH8M to xxM clock source	16M	4990	6290	μA
				24M	6910	8910	
				32M FlashWait=1	8300	9600	
				40M FlashWait=1	10200	11840	
				48M FlashWait=1	12090	14050	
		$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$ $T_A=-40\sim 85^{\circ}C$	RCH clock source	4M	760	1240	μA
				8M	1360	2290	
				16M	2600	4190	
				22.12M	3460	5600	
				24M	3710	6010	
				24M	3710	6010	
		$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$ $T_A=-40\sim 85^{\circ}C$	PLL RCH4M to xxM clock source	16M	2860	3940	μA
				24M	3830	5450	
				32M FlashWait=1	4260	5030	

				40M FlashWait=1	5140	6190	
				48M FlashWait=1	6080	7340	
		V _{core} =1.5V V _{CC} =1.8~5.5V T _A =-40~85°C	PLL RCH8M to xxM clock source	16M	2930	4010	μA
				24M	3900	5510	
				32M FlashWait=1	4330	5090	
				40M FlashWait=1	5200	6230	
				48M FlashWait=1	6140	7390	
I _{DD} (Sleep mode)	All peripherals clock ON	V _{core} =1.5V V _{CC} =1.8~5.5V T _A =-40~85°C	RCH clock source	4M	690	780	μA
				8M	1330	1490	
				16M	2560	2850	
				22.12M	3540	3950	
				24M	3840	4280	
		V _{core} =1.5V V _{CC} =1.8~5.5V T _A =-40~85°C	PLL RCH4M to xxM clock source	16M	2860	3200	μA
				24M	4000	4490	
				32M FlashWait=1	5240	5890	
				40M FlashWait=1	6500	7310	
				48M FlashWait=1	7740	8730	
		V _{core} =1.5V V _{CC} =1.8~5.5V T _A =-40~85°C	PLL RCH8M to xxM clock source	16M	2490	3280	μA
				24M	4060	4560	
				32M FlashWait=1	5300	5950	
				40M FlashWait=1	6560	7360	
				48M FlashWait=1	7810	8760	
	All peripherals clock OFF	V _{core} =1.5V V _{CC} =1.8~5.5V T _A =-40~85°C	RCH clock source	4M	190	230	μA
				8M	330	390	
				16M	560	650	
				22.12M	760	880	
				24M	810	930	
		V _{core} =1.5V V _{CC} =1.8~5.5V T _A =-40~85°C	PLL RCH4M to xxM clock source	16M	860	980	μA
				24M	990	1110	
				32M	1240	1390	

				FlashWait=1			μA
				40M FlashWait=1	1500	1680	
				48M FlashWait=1	1760	1960	
				16M	930	1050	
				24M	1050	1190	
				32M FlashWait=1	1300	1460	
				40M FlashWait=1	1560	1750	
				48M FlashWait=1	1830	2040	
I_{DD} (LP Run)	All peripherals clock ON, Run while(1) in Flash	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K clock source Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	14	25	μA
				$T_A=50^{\circ}\text{C}$	15	24	
				$T_A=85^{\circ}\text{C}$	24	34	
	All peripherals clock OFF, Run while(1) in Flash	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K clock source Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	10	20	μA
				$T_A=50^{\circ}\text{C}$	11	19	
				$T_A=85^{\circ}\text{C}$	20	30	
I_{DD} (LP Sleep)	All peripherals clock ON	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K clock source Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	8	9	μA
				$T_A=50^{\circ}\text{C}$	9	10	
				$T_A=85^{\circ}\text{C}$	18	21	
	All peripherals clock OFF	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K clock source Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	4	4	μA
				$T_A=50^{\circ}\text{C}$	5	6	
				$T_A=85^{\circ}\text{C}$	14	18	
	LpTimer+RTC+32K clk ON, Other clk OFF	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K clock source Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	4	5	μA
				$T_A=50^{\circ}\text{C}$	5	6	
				$T_A=85^{\circ}\text{C}$	14	18	
I_{DD} (DeepSleep mode)	RTC+WDT+LPT+XTL32K +DeepSleep	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	1410	2110	nA
				$T_A=50^{\circ}\text{C}$	2330	3560	
				$T_A=85^{\circ}\text{C}$	8150	12910	
	LPT+XTL32K +DeepSleep	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	1290	1540	nA
				$T_A=50^{\circ}\text{C}$	2200	2690	
				$T_A=85^{\circ}\text{C}$	8080	9800	
	RTC+XTL32K +DeepSleep	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	1230	1410	nA
				$T_A=50^{\circ}\text{C}$	2150	2860	
				$T_A=85^{\circ}\text{C}$	7980	12100	
	XTL32K +DeepSleep	$V_{core}=1.5V$ $V_{CC}=1.8\sim 5.5V$	XTL32K Driver=0x0	$T_A=-40\sim 25^{\circ}\text{C}$	1230	1410	nA
				$T_A=50^{\circ}\text{C}$	2150	2840	
				$T_A=85^{\circ}\text{C}$	7980	12100	

	IRC32K +DeepSleep	Vcore=1.5V VCC=1.8~5.5V		T _A =-40~25°C	1150	1360	nA
				T _A =50°C	2070	2800	
				T _A =85°C	7910	12080	
	WDT +DeepSleep	Vcore=1.5V VCC=1.8~5.5V		T _A =-40~25°C	890	1050	nA
				T _A =50°C	1790	2180	
				T _A =85°C	7600	9380	
	DeepSleep	Vcore=1.5V VCC=1.8~5.5V		T _A =-40~25°C	760	920	nA
				T _A =50°C	1660	2330	
				T _A =85°C	7840	11630	

1. 若没有其他指定条件，该 Typ 的值是在 25 °C & V_{CC} = 3.3V 测得。
2. 若没有其他指定条件，该 Max 的值是 V_{CC} = 1.8~5.5V & Temperature = -40~85°C 范围内的最大值。
3. 数据基于考核结果，不在生产中测试。

表 46-8 工作电流特性

46.3.6 从低功耗模式唤醒的时间

唤醒时间是在 RCH 振荡器的唤醒阶段测量得到。唤醒时使用的时钟源依当前的操作模式而定：

- 休眠模式：时钟源是 RCH 振荡器
- 深度休眠模式：时钟源是进入深度休眠时所使用的时钟是 RCH 振荡器

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{wu}	休眠模式唤醒时间			1.8		μs
	深度休眠唤醒时间	$F_{MCLK} = 4MHz$		9.0		μs
		$F_{MCLK} = 8MHz$		6.0		μs
		$F_{MCLK} = 16MHz$		5.0		μs
		$F_{MCLK} = 24MHz$		4.0		μs

1. 唤醒时间的测量是从唤醒事件开始至用户程序读取第一条指令。

46.3.7 外部时钟源特性

外部输入高速时钟

符号	参数	条件	最小值	典型值	最大值	单位
f_{XTH_ext}	用户外部时钟频率 ⁽¹⁾		0	8	32	MHz
V_{XTHH}	输入引脚高电平电压		$0.7 \cdot V_{CC}$		V_{CC}	V
V_{XTHL}	输入引脚低电平电压		VSS		$0.3 \cdot V_{CC}$	V
$T_{r(XTH)}$	上升的时间 ⁽¹⁾				20	ns
$T_{f(XTH)}$	下降的时间 ⁽¹⁾				20	ns
$T_w(XTH)$	输入高或低的时间 ⁽¹⁾		16			ns
$C_{in(XTH)}$	输入容抗 ⁽¹⁾			5		pF
Duty	占空比		40		60	%
I_L	输入漏电流				± 1	μA

1. 由设计保证，不在生产中测试。

外部输入低速时钟

符号	参数	条件	最小值	典型值	最大值	单位
f_{XTH_ext}	用户外部时钟频率 ⁽¹⁾		0	32.768	1000	kHz
V_{XTHH}	输入引脚高电平电压		$0.7 \cdot V_{CC}$		V_{CC}	V
V_{XTHL}	输入引脚低电平电压		VSS		$0.3 \cdot V_{CC}$	V
$T_{r(XTH)}$	上升的时间 ⁽¹⁾				50	ns
$T_{f(XTH)}$	下降的时间 ⁽¹⁾				50	ns
$T_w(XTH)$	输入高或低的时间 ⁽¹⁾		450			ns
$C_{in(XTH)}$	输入容抗 ⁽¹⁾			5		pF

Duty	占空比		30		70	%
I _L	输入漏电流				±1	μA

1.由设计保证，不在生产中测试。

高速外部时钟 XTH

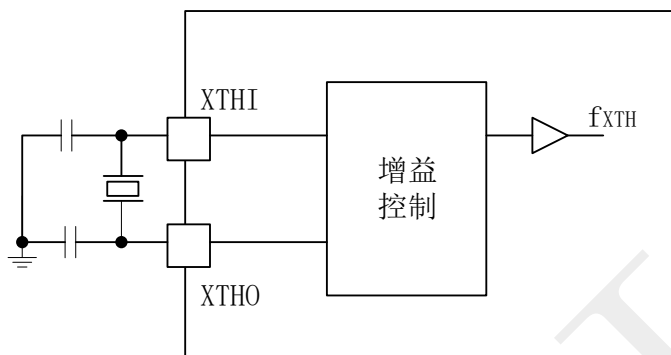
高速外部时钟(XTH)可以使用一个 4~32MHz 的晶体/陶瓷谐振器构成的振荡器产生。本节中所给出的信息是基于使用下表中列出的典型外部元器件,通过综合特性评估得到的结果。在应用中,谐振器和负载电容必须尽可能地靠近振荡器的引脚,以减小输出失真和启动时的稳定时间。有关晶体谐振器的详细参数(频率、封装、精度等),请咨询相应的生产厂商。

外部 XTH 晶振⁽¹⁾⁽²⁾

符号	参数	条件	最小值	典型值	最大值	单位
F _{CLK}	振荡频率		4		32	MHz
ESR _{CLK}	支持的晶振ESR范围	32M		30	60	Ω
		4M		400	1500	Ω
C _{LX} ⁽³⁾	负载电容	两个管脚都有负载电容		12		pF
Duty	占空比		40	50	60	%
I _{dd} ⁽⁴⁾	电流	32M Xtal, C _L =12pF, ESR=30Ω		600		μA
T _{start} ⁽⁵⁾	启动时间	32MHz @ XTH_CR.Driver=1111		300		μs
		4MHz @ XTH_CR.Driver=0011		2		ms

1. 谐振器的特性参数由晶体/陶瓷谐振器制造商给出。
2. 由综合评估得出，不在生产中测试。
3. C_{LX} 指 XTAL 的两个管脚负载电容 C_{L1} 和 C_{L2}。对于 C_{L1} 和 C_{L2}，建议使用高质量的、为高频应用而设计瓷介电容器，并挑选符合要求的晶体或谐振器。通常 C_{L1} 和 C_{L2} 具有相同参数。晶体制造商通常以 C_{L1} 和 C_{L2} 的串行组合给出负载电容的参数。在选择 C_{L1} 和 C_{L2} 时，应该根据晶振的频率和 ESR 等参数，并且将 PCB 和 MCU 引脚的容抗考虑在内。在晶振频率为 32M 时，C_{LX} 需要选择小的电容值，XTH_CR.Driver 为 1110 时，可以选择 C_{LX} 为 12pF。
4. 电流跟随频率变化而变化，测试条件：XTH_CR.Driver=1110
5. T_{start} 是启动时间，是从软件使能 XTH 开始测量，直至得到稳定的 32MHz/4MHz 振荡

这段时间。这个数值是在 XTH_CR.Startup=10 设置下，使用一个标准的晶体谐振器上测量得到，它可能因晶体制造商和型号的不同而变化较大。



低速外部时钟 XTL

低速外部时钟(XTL)可以使用一个 32.768kHz 的晶体/陶瓷谐振器构成的振荡器产生。本节中所给出的信息是基于典型外部元器件，通过综合特性评估得到的结果。在应用中，谐振器和负载电容必须尽可能地靠近振荡器的引脚，以减小输出失真和启动时的稳定时间。有关晶体谐振器的详细参数(频率、封装、精度等)，请咨询相应的生产厂商。

外部 XTL 晶振⁽¹⁾

符号	参数	条件	最小值	典型值	最大值	单位
F_{CLK}	振荡频率			32.768		kHz
ESR_{CLK}	支持的晶振ESR范围			65	85	k Ω
$C_{LX}^{(2)}$	负载电容	两个管脚都有负载电容		12		pF
DC_{ACLK}	占空比		30	50	70	%
$I_{dd}^{(3)}$	电流	$ESR=65\text{ k}\Omega$, $C_L=12\text{ pF}$		350	1000	nA
$T_{start}^{(4)}$	启动时间	$ESR=65\text{ k}\Omega$, $C_L=12\text{ pF}$, 40% - 60% duty cycle has been reached		500		ms

1. 由综合评估得出，不在生产中测试。
2. C_{LX} 指 XTAL 的两个管脚负载电容 C_{L1} 和 C_{L2} 。对于 C_{L1} 和 C_{L2} ，建议使用高质量的瓷

		$F_{MCLK} = 16\text{MHz}$		120		μA
		$F_{MCLK} = 24\text{MHz}$		140		μA
DC_{CLK}	占空比 ⁽¹⁾		45	50	55	%

1. 由综合评估得出，不在生产中测试。

内部 RCL 振荡器

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Dev	RCL振荡器精度	User-trimmed with clock_trim for given VCC and TA conditions		0.5		%
		VCC = 1.8V ~ 5.5V T _{AMB} = -40°C ~ 85°C T _{AMB} = -40°C ~ 85°C	-2.5		+2.5	%
		VCC = 1.8V ~ 5.5V T _{AMB} = -20°C ~ 50°C	-1.5		+1.5	%
F_{CLK}	振荡频率			38.4 32.768		kHz
T_{CLK}	启动时间			150		μs
DC_{CLK}	占空比 ⁽¹⁾		25	50	75	%
I_{CLK}	功耗			0.35		μA

1. 由综合评估得出，不在生产中测试。

46.3.9 PLL 特性

符号	参数	条件	最小值	典型值	最大值	单位
$F_{in}^{(1)}$	输入时钟		4	4	24	MHz
	输入时钟占空比		40		60	%
F_{out}	输出频率		8	—	48	MHz
Duty ⁽¹⁾	输出占空比		48%	—	52%	—
$T_{lock}^{(1)}$	锁定时间	输入频率4MHz	—	100	200	μs

1. 由综合评估得出，不在生产中测试。

46.3.10 存储器特性

符号	参数	条件	最小值	典型值	最大值	单位
EC _{FLASH}	擦写次数	Regulator Voltage=1.5V, T _{AMB} = 25℃	20K			cycles
RET _{FLASH}	数据保存期限	T _{AMB} = 85℃	20			Years
		常温	100			Years
Tw _{prog}	编程时间		6		7.5	μs
Tp _{erase}	页擦除时间		4		5	ms
Tm _{erase}	整片擦除时间		30		40	ms

46.3.11 EFT 特性

芯片复位可以使系统恢复正常操作。

符号	级别/类型
EFT to IO (IEC61000-4-4)	2kV Class:4
EFT to Power (IEC61000-4-4)	4kV Class:4

软件建议

软件的流程中必须包含程序跑飞的控制，如：

- 被破坏的程序计数器
- 意外的复位
- 关键数据被破坏(控制寄存器等)

在进行 ESD 测试时，可以把超出应用要求的电压直接施加在芯片上，当检测到意外动作的地方，软件部分需要加强以防止发生不可恢复的错。

46.3.12 绝对最大值(电气敏感性)

使用特定的测量方法，对芯片进行强度测试以决定它的电气敏感性方面的性能。

符号	参数	条件	最小值	典型值	最大值	单位
VESD _{HEM}	ESD @HumanBody Mode			8		kV
VESD _{CDM}	ESD @Charge DeviceMode			2		kV
VESD _{MM}	ESD @machineMode			200	400	V
I _{latchup}	Latch up current			200	400	mA

46.3.13 I/O 端口特性

输出特性——端口

符号	参数	条件	最小值	最大值	单位
V_{OH}	High level output voltage Source Current	Sourcing 5 mA, $V_{CC} = 3.3\text{ V}$ (see Note 1)	$V_{CC}-0.25$		V
		Sourcing 10 mA, $V_{CC} = 3.3\text{ V}$ (see Note 2)	$V_{CC}-0.6$		V
V_{OL}	Low level output voltage Sink Current	Sinking 6 mA, $V_{CC} = 3.3\text{ V}$ (see Note 1)		$V_{SS}+0.25$	V
		Sinking 15 mA, $V_{CC} = 3.3\text{ V}$ (see Note 2)		$V_{SS}+0.6$	V
V_{OHD}	High level output voltage Double source Current	Sourcing 10 mA, $V_{CC} = 3.3\text{ V}$ (see Note 1)	$V_{CC}-0.25$		V
		Sourcing 20 mA, $V_{CC} = 3.3\text{ V}$ (see Note 2)	$V_{CC}-0.6$		V
V_{OLD}	Low level output voltage Double Sink Current	Sinking 10 mA, $V_{CC} = 3.3\text{ V}$ (see Note 1)		$V_{SS}+0.25$	V
		Sinking 20 mA, $V_{CC} = 3.3\text{ V}$ (see Note 2)		$V_{SS}+0.6$	V

表 46-9 端口输出特性

NOTES: 1. The maximum total current, $I_{OH(max)}$ and $I_{OL(max)}$, for all outputs combined, should not exceed 40 mA to satisfy the maximum specified voltage drop.

2. The maximum total current, $I_{OH(max)}$ and $I_{OL(max)}$, for all outputs combined, should not exceed 100 mA to satisfy the maximum specified voltage drop.

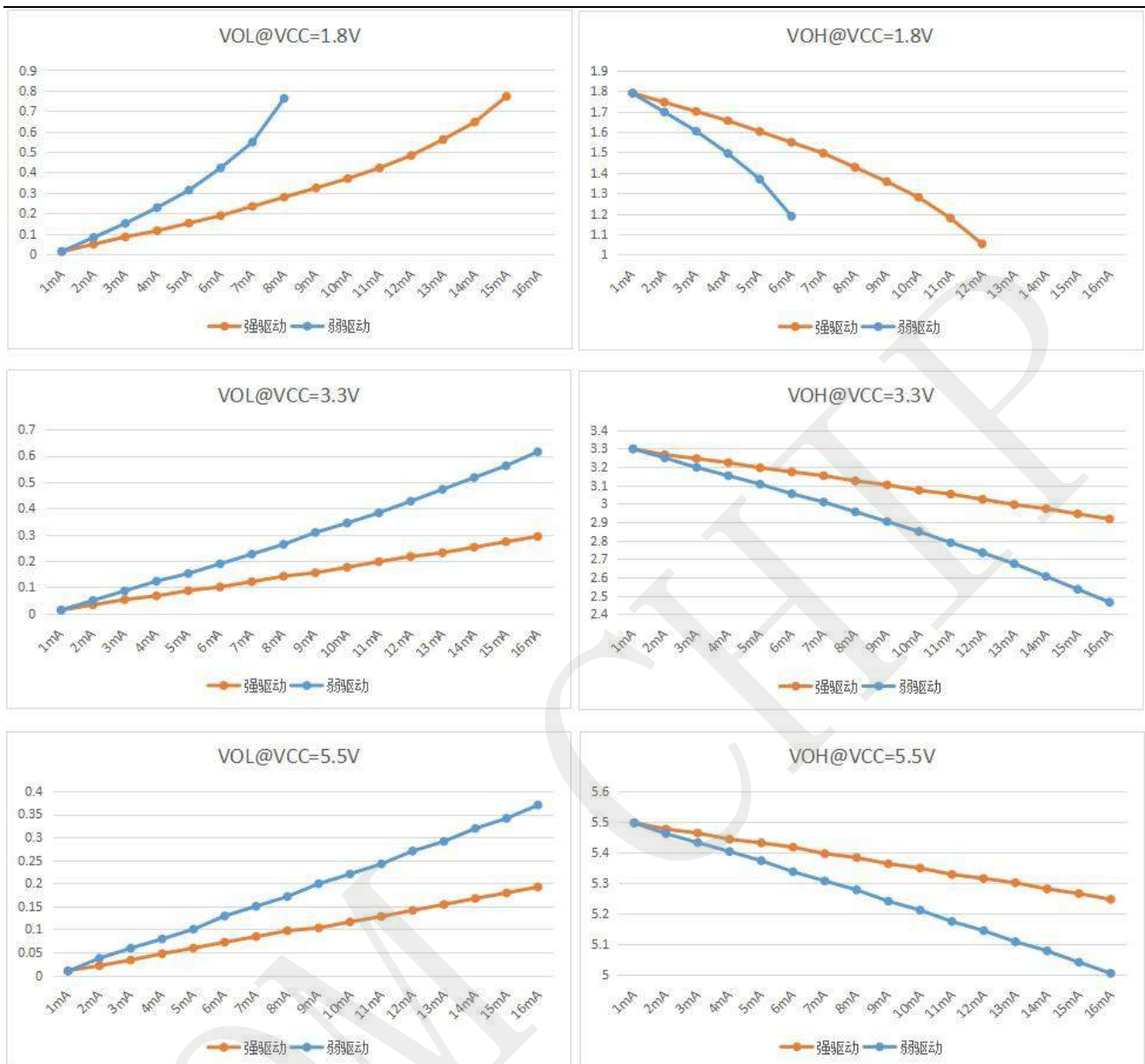


图 46-2 输出端口 VOH/VOL 实测曲线

输入特性——端口 PA,PB,PC,PD, PE,PF,RESET

符号	参数	条件	最小值	典型值	最大值	单位
V_{IH}	Positive-going input threshold voltage	VCC=1.8v	0.7*VCC			V
		VCC=3.3v	0.7*VCC			V
		VCC=5.5v	0.7*VCC			V
V_{IL}	Negative-going input threshold voltage	VCC=1.8v			0.3*VCC	V
		VCC=3.3v			0.3*VCC	V
		VCC=5.5v			0.3*VCC	V
$V_{hys}^{(1)}$	Input voltage hysteresis ($V_{IH} - V_{IL}$)	VCC=1.8v		0.3		V
		VCC=3.3v		0.4		V
		VCC=5.5v		0.6		V
$R_{pullhigh}$	Pullup resistor	Pullup enabled VCC=3.3V		80		k Ω
$R_{pulllow}$	Pulldown resistor	Pulldown enabled VCC=3.3V		40		k Ω
C_{input}	Input capacitance			5		pF

1. 由综合评估得出，不在生产中测试。

端口漏电特性——PA,PB,PC,PD,PE,PF

符号	参数	条件	最小值	典型值	最大值	单位
$I_{lk}(Px, y)$	Leakage current	V(Px, y) (see Note 1, 2)		± 50		nA

NOTES: 1. The leakage current is measured with VSS or VCC applied to the corresponding pin(s), unless otherwise noted.

2. The port pin must be selected as input.

46.3.14 RESETB 引脚特性

RESETB 引脚输入驱动使用 CMOS 工艺，它连接了一个不能断开的上拉电阻。

符号	参数	条件	最小值	典型值	最大值	单位
$V_{IL}^{(1)}$ (RESETB)	输入低电平电压		-0.3		0.3*VCC	
$V_{IH}^{(1)}$ (RESETB)	输入高电平电压		0.7*VCC		VCC+0.3	
$V_{hys}^{(1)}$ (RESETB)	施密特触发器电压迟滞			200		mV
R_{PU}	弱上拉等效电阻	$V_{IN}=VSS$		80		k Ω
$V_F^{(1)}$ (RESETB)	输入滤波脉冲				100	ns
$V_{NF}^{(1)}$ (RESETB)	输入非滤波脉冲		300			ns

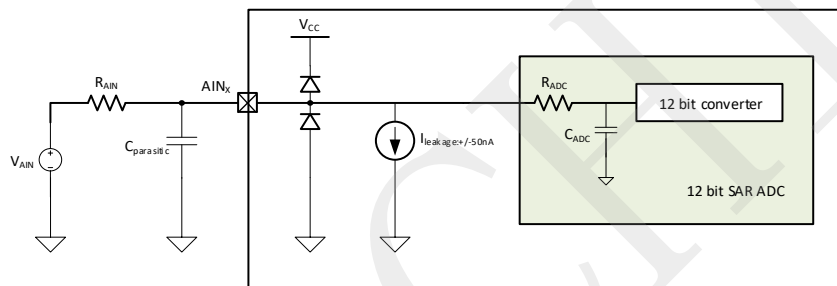
1. 由设计保证，不在生产中测试。

46.3.15 ADC 特性

符号	参数	条件	最小值	典型值	最大值	单位
V_{ADCIN}	Input voltage range	Single ended	0		V_{ADCREFIN}	V
V_{ADCREFIN}	Input range of external reference voltage	Single ended	0		AVCC	V
I_{ADC1}	Active current including reference generator and buffer	200kSPS + Internal Vref		2		mA
I_{ADC2}	Active current without reference generator and buffer	1MSPS + AVCC / Ext Vref		0.5		mA
C_{ADCIN}	ADC input capacitance			16	19.2	pF
$R_{\text{ADC}}^{(1)}$	ADC sampling switch impedance			1.5		k Ω
$R_{\text{AIN}}^{(1)}$	ADC external input resistor (2)				100	k Ω
F_{ADCLK}	ADC clock Frequency				24M	Hz
T_{ADCSTART}	Startup time of reference generator and ADC core			30		μ s
T_{ADCONV}	Conversion time		20	24	28	cycles
ENOB	Effective Bits	1MSPS@VCC>=2.7v 500kSPS@VCC>=2.4v 200kSPS@VCC>=1.8v REF=EXREF		10.3		Bit
		1MSPS@VCC>=2.7v 500kSPS@VCC>=2.4v 200kSPS@VCC>=1.8v REF=VCC		10.3		Bit
		200kSPS@VCC>=1.8v REF=internal 1.5V		9.4		Bit
		200kSPS@VCC>=2.8v REF=internal 2.5V		9.4		Bit
SNR	Signal to Noise Ratio	1MSPS@VCC>=2.7v 500kSPS@VCC>=2.4v 200kSPS@VCC>=1.8v REF=EXREF		68.2		dB
		1MSPS@VCC>=2.7v 500kSPS@VCC>=2.4v 200kSPS@VCC>=1.8v REF=VCC		68.2		dB

		200kSPS@VCC>=1.8v REF=internal 1.5V		60		dB
		200kSPS@VCC>=2.8v REF=internal 2.5V		60		dB
DNL ⁽¹⁾	Differential non-linearity	200kSPS; VREF=EXREF/AVCC	-1		1	LSB
INL ⁽¹⁾	Integral non-linearity	200kSPS; VREF=EXREF/AVCC	-3		3	LSB
Eo	Offset error			0		LSB
Eg	Gain error			0		LSB

1. 由设计保证，不在生产中测试。
2. ADC 的典型应用如下图所示：



对于 0.5LSB 采样误差精度要求的条件下，外部输入阻抗的计算公式如下：

$$R_{AIN} = \frac{M}{F_{ADC} * C_{ADC} * (N + 1) * \ln(2)} - R_{ADC}$$

其中 F_{ADC} 为 ADC 时钟频率，寄存器 ADC_CR0<3:2>可设定其与 PCLK 的关系，如表

46-10 ADC 时钟频率 F_{ADC} 和 PCLK 分频比关系

ADC_CR0<3:2>	N
00	1
01	2
10	4
11	8

表 46-10 ADC 时钟频率 F_{ADC} 和 PCLK 分频比关系

M 为采样周期个数，由寄存器 ADC_CR0<13:12>设定。

ADC_CR0<13:12>	M
00	4
01	6
10	8
11	12

表 46-11 采样时间 t_{sa} 和 ADC 时钟频率 F_{ADC} 的关系

R_{AIN} (k Ω)	F_{ADC} (kHz)
10	5600
30	2100
50	1300
80	820
100	660
120	550
150	450

表 46-12 ADC 时钟频率 F_{ADC} 和外部电阻 R_{AIN} 的关系 (M=12,采样误差 0.5LSB 的条件下)

对于上述典型应用，应注意：

- (a) 尽量减小 ADC 输入端口 AIN_X 的寄生电容 $C_{PARACITIC}$ ；
- (b) 除了考虑 R_{AIN} 值外，如果信号源 V_{AIN} 的内阻较大时，也需要加入考虑。

46.3.16VC 特性

符号	参数	条件	最小值	典型值	最大值	单位
V_{in}	Input voltage range		0		AVCC	V
V_{incom}	Input common mode range		0		AVCC-0.2	V
V_{offset}	Input offset	常温25℃ 3.3V	-10		+10	mV
I_{comp}	Comparator's current	VCx_BIAS_SEL=00 VCx_BIAS_SEL=01 VCx_BIAS_SEL=10 VCx_BIAS_SEL=11		0.3 1.2 10 20		μA

T_{response}	Comparator' s response time when one input cross another	VCx_BIAS_SEL=00 VCx_BIAS_SEL=01 VCx_BIAS_SEL=10 VCx_BIAS_SEL=11		20 5 1 0.2		μs
T_{setup}	Comparator' s setup time when ENABLE. Input signals unchanged.	VCx_BIAS_SEL=00 VCx_BIAS_SEL=01 VCx_BIAS_SEL=10 VCx_BIAS_SEL=11		20 5 1 0.2		μs
T_{wakeup}	From main bandgap enable to 1.2V BGR reference、Temp sensor voltage、ADC internal 1.5V、2.5V reference stable			20		μs
T_{filter}	Digital filter time	VC_debounce = 000 VC_debounce = 001 VC_debounce = 010 VC_debounce = 011 VC_debounce = 100 VC_debounce = 101 VC_debounce = 110 VC_debounce = 111		7 14 28 112 450 1800 7200 28800		μs

46.3.17 OPA 特性

OPA: (AVCC=2.2V ~ 5.5 V, AVSS=0 V, Ta=- 40°C ~ +85°C)

符号	参数	条件	最小值	典型值	最大值	单位
V_{in}	输入电压		0	—	AVCC	V
V_{o}	输出电压 ⁽¹⁾		0.1	—	AVCC-0.2	V
I_{o}	输出电流 ⁽¹⁾				1	mA
R_{L}	负载电阻 ⁽¹⁾		5k			Ω
T_{start}	初始化时间 ⁽²⁾				20	μs
V_{io}	输入失调电压	$V_{\text{ic}}=\text{AVCC}/2$, $V_{\text{o}}=\text{AVCC}/2$, $R_{\text{L}}=5\text{k}\Omega$, $R_{\text{s}}=50\text{ pF}$		± 6		mV
PM	相位裕度 ⁽¹⁾	$V_{\text{ic}}=\text{AVCC}/2$, $V_{\text{o}}=\text{AVCC}/2$ $R_{\text{L}}=5\text{k}\Omega$, $C_{\text{L}}=50\text{pF}$		80	—	deg
UGBW	单位增益带宽 ⁽¹⁾	$V_{\text{ic}}=\text{AVCC}/2$, $V_{\text{o}}=\text{AVCC}/2$ $R_{\text{L}}=5\text{k}\Omega$, $C_{\text{L}}=50\text{pF}$		9.3		MHz
SR	压摆率 ⁽¹⁾	$R_{\text{L}}=5\text{k}\Omega$, $C_{\text{L}}=50\text{pF}$		8		V/ μs

由设计保证，不在生产中测试。

1. 需要同时设置 BGR_CR<0>=1

46.3.18 LCD 控制器

符号	参数	工作条件	最小	典型	最大	单位
I_{LCD}	电流	$V_{CC}=3.3V$		3.3		μA
R_H	低驱动电阻			1M		Ω
R_L	高驱动电阻			360k		Ω
V_{LCDH}	LCD 可调最高电压				V_{CC}	V
V_{LCD3}	LCD 最高电压				V_{LCDH}	V
V_{LCD2}	LCD 2/3电压				$2/3 V_{LCDH}$	V
V_{LCD1}	LCD 1/3电压				$1/3 V_{LCDH}$	V
V_{LCD0}	LCD最低电压		0			V
ΔV_{XX}	LCD 电压偏差	$T_A=-40\sim 85^{\circ}C$			± 50	mV

46.3.19 DAC 特性

符号	参数	工作条件	最小	典型	最大	单位
V_{DACOUT}	Output voltage range	AVDD voltage reference, single ended	0		V_{CC}	V
V_{DACCM}	Output common mode voltage range		0		V_{CC}	V
I_{DAC}	Active current	500kSPS		15		μA
SR_{DAC}	Sample rate				500	kSPS
$t_{DACCONV}$	Conversion time		2			μs
$t_{DACSETTLE}$	Setting time			5		μs
SNR_{DAC}	Signal to Noise Ratio			59		dB
$SNDR_{DAC}$	Signal to Noise and Distortion Ratio			57		dB
$SFDR_{DAC}$	Spurious Free Dynamic Range			56		dB
$V_{DACOFFSET}$	Offset voltage	w/o buffer		2		mV
DNL_{DAC}	Differential non-linearity			± 1		LSB
INL_{DAC}	Integral non-linearity			± 5		LSB

46.4 唯一设备 ID 号

唯一设备标识符最适合：

- 用作序列号
- 在对内部 Flash 进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥以提高 Flash 中代码的安全性
- 激活安全自举过程等

80 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。

80 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

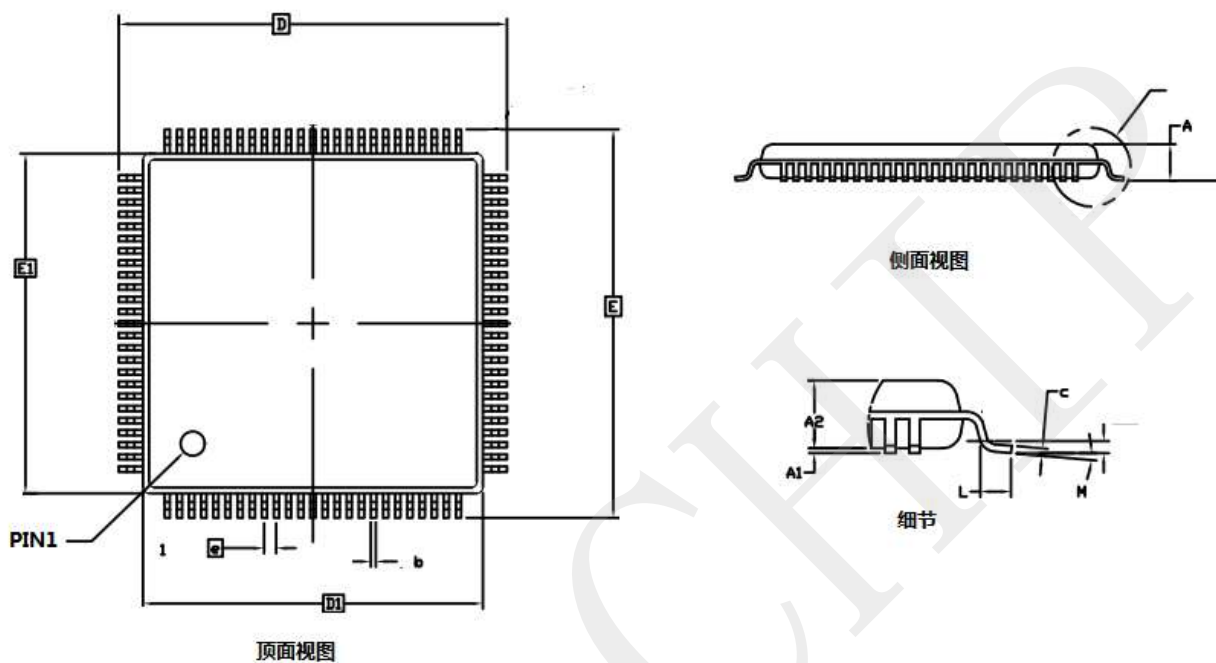
基址：0x0010 0E74

UID 规则

偏移地址	描述	UID Bits (80 bits)							
		7	6	5	4	3	2	1	0
0	Lot Number	UID[7:0]							
1		UID[15:8]							
2		UID[23:16]							
3		UID[31:24]							
4		UID[39:32]							
5		UID[47:40]							
6	X Coordinate on the wafer	UID[55:48]							
7	Y Coordinate on the wafer	UID[63:56]							
8	Wafer Number	UID[71:64]							
9	Rev ID	UID[79:72]							

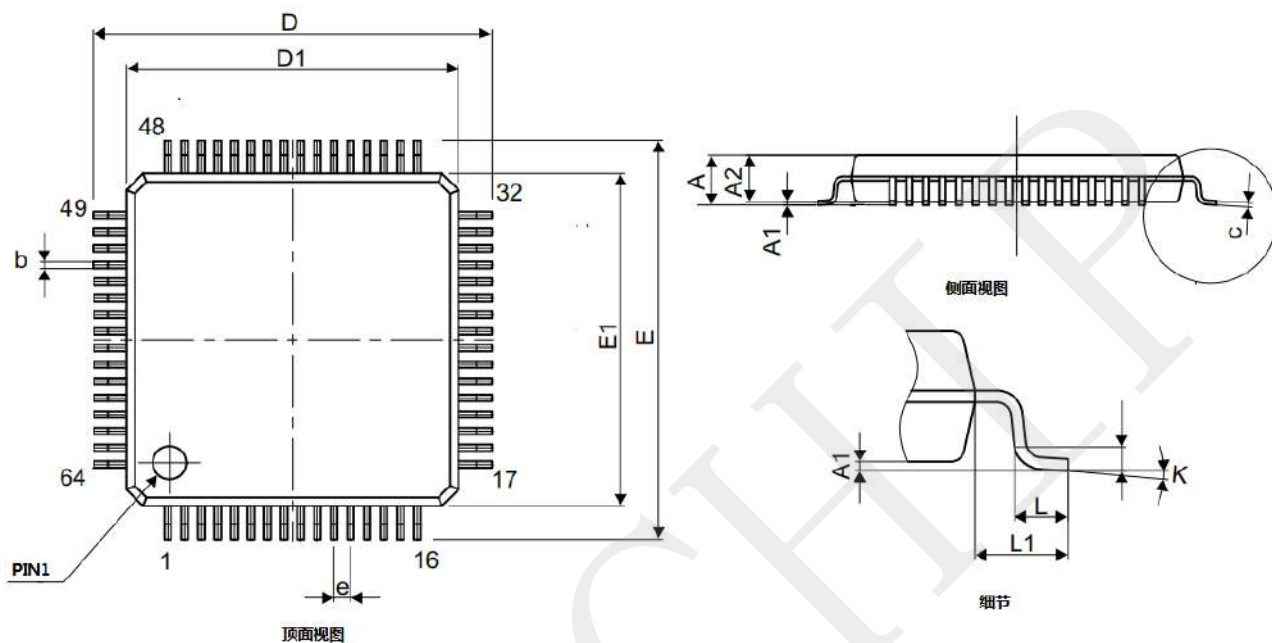
47. 器件封装

LQFP100



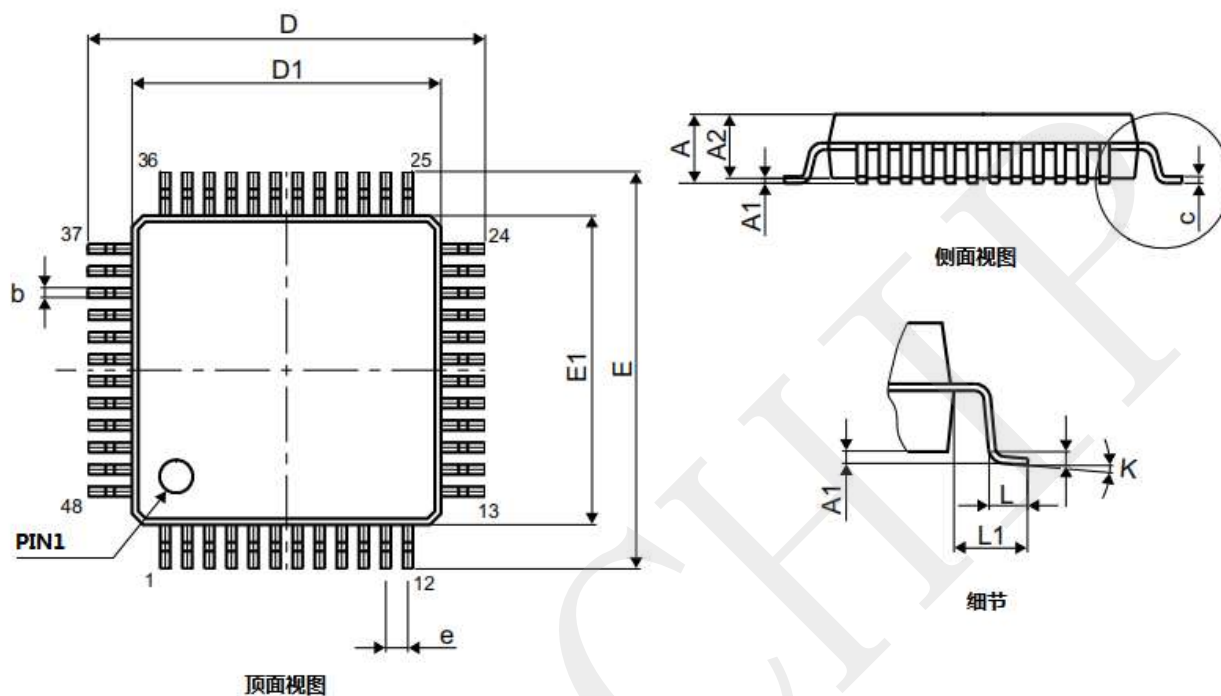
符号	尺寸(mm)		
	最小值	典型值	最大值
D	15.8	16.0	16.2
D1	13.8	14.0	14.2
E	15.8	16.0	16.2
E1	13.8	14.0	14.2
A	-	-	1.60
A1	0.05	0.10	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.09	0.15	0.20
e		0.5	
L	0.45	0.60	0.75
H	0°	3.5°	7°

LQFP64



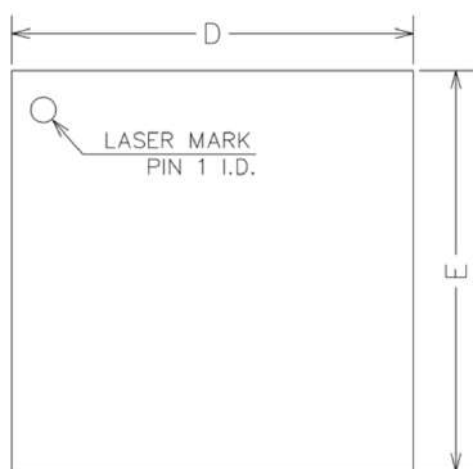
符号	尺寸(mm)		
	最小值	典型值	最大值
D	11.8	12.0	12.2
D1	9.8	10.0	10.2
E	11.8	12	12.2
E1	9.8	10	10.2
A	-	-	1.60
A1	0.05	0.10	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.09	0.15	0.20
e	-	0.5	-
L	0.45	0.60	0.75
L1	-	1.00	-
K	0°	3.5°	7°

LQFP48

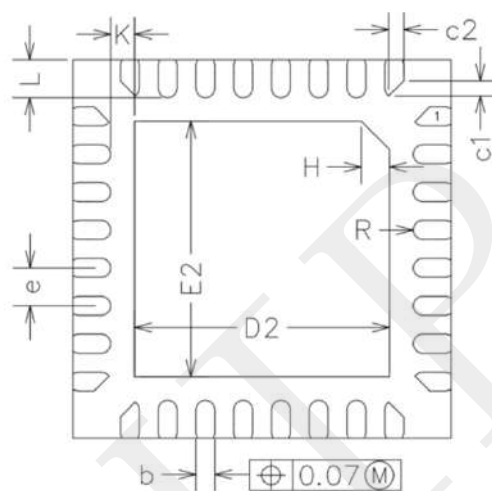


符号	尺寸(mm)		
	最小值	典型值	最大值
D	8.8	9.0	9.2
D1	6.8	7.0	7.2
E	8.8	9.0	9.2
E1	6.8	7.0	7.2
A	-	-	1.60
A1	0.05	0.10	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.09	0.15	0.20
e	-	0.5	-
L	0.45	0.60	0.75
L1	-	1.00	-
K	0°	3.5°	7°

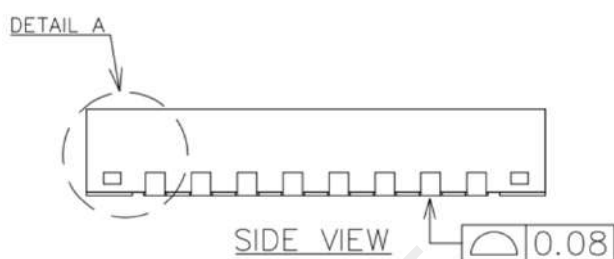
QFN32



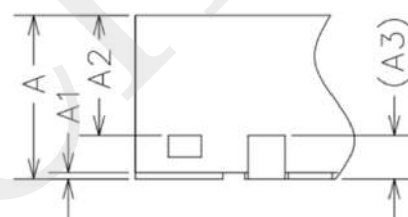
顶视图



底视图



侧视图



细节图

符号	尺寸(mm)		
	最小值	典型值	最大值
A	0.70	0.75	0.80
A1	0	0.02	0.05
A2	0.50	0.55	0.60
A3	0.20REF		
b	0.15	0.20	0.25
D	3.90	4.00	4.10
E	3.90	4.00	4.10
D2	2.60	2.70	2.80
E2	2.60	2.70	2.80
e	0.30	0.40	0.50
H	0.30REF		
K	0.25REF		
L	0.35	0.40	0.45
R	0.09	-	-
c1	-	0.16	-
c2	-	0.16	-

重要提示

未经成都声光微科技有限公司允许，任何单位和个人不得以任何方式和任何形式对其规格/数据表转载。

成都声光微科技有限公司及其子公司有权更改其公司产品说明书和/或产品，或停止任何产品或服务，恕不另行通知，并告知客户以获取有关信息的最新版本的验证，在下订单前，该产品说明书作为当前最新最完整解释依据。所有产品均以订单确认时提供的销售条件为准，包括有关保修、专利侵权和责任限制的条款。

成都声光微科技有限公司保证其产品性能规格适用在按照本公司标准保修时间范围内，本公司的测试和质量控制提供这项保证支持。非产品说明书定义的每个器件所有参数测试不全部执行测试，除了官方要求授权。

客户须知，成都声光微科技有限公司的产品不得被设计、制造用于掺入生命支持或其他危险的活动或环境中，对其产品的失败可能导致死亡，人身伤害的任何系统或产品，或财产或环境损害（“高风险应用”）中。成都声光微科技有限公司特此声明，本公司没有责任对客户或任何第三方，将本公司产品用于涉及任何高风险活动的产品使用中。

成都声光微科技有限公司

COPYRIGHT © 2016-2025, AOM technology limited